Zurich University of Applied Sciences

School of Engineering

Centre for Artificial Intelligence

# Self-Organisation in a Biologically Inspired Learning Framework Based on Bernoulli Neurons

Incorporating Findings from Neuroscience about Natural Intelligence into a Computational Framework

A thesis by

Pascal Sager

for the degree of

Master of Science

Supervised by

Prof. Dr. Thilo Stadelmann

Dr. Jan Milan Deriu

Prof. Dr. Christoph von der Malsburg

August 21, 2023

*"Max Planck said, 'Science progresses one funeral at a time.'*
*The future depends on some graduate student who is deeply*
*suspicious of everything I have said."*

– Geoffrey E. Hinton, University of Toronto, 2017.

# Zusammenfassung

Deep Learning hat sich im letzten Jahrzehnt im Bereich der automatischen Bildanalyse als Standard-Technologie etabliert. Trotz beeindruckender Ergebnisse weist diese Technologie diverse Schwächen auf, wie begrenzte Robustheit gegenüber Störsignalen, eingeschränkte Transformationsinvarianz bei der Objekterkennung sowie den Bedarf an umfangreichen Trainingsdaten. Im Gegensatz dazu sind diese Schwächen im menschlichen Gehirn kaum vorhanden. Dies resultiert aus der nicht-sequenziellen Verarbeitung extrahierter Bildmerkmale im Gehirn und der Fähigkeit, eine visuelle Szene als mehr als die Summe ihrer Teile zu interpretieren, wie es die Gestalt-Psychologie beschreibt. Diese Fähigkeit ist darauf zurückzuführen, dass das Gehirn interne Konsistenz zwischen allen verbundenen Zellen mittels Selbst-Organisation und lokalem Lernen bildet, d.h. es wird durch gegenseitigen Zellsupport ein Konsens zwischen allen Merkmalen erreicht. Dadurch kann das "Early Commitment"-Problem gelöst werden, welches inhärent in tiefen neuronalen Netzwerken vorhanden ist. Neuronale Netzwerke sind von diesem Problem betroffen, weil sie Konsistenz nur an einer Stelle zwischen einer Vorhersage und einem Lernsignal mittels globalem Fehlerkorrekturalgorithmus bilden.

Basierend auf diesen Erkenntnissen wird in dieser Thesis ein neues Bildverarbeitungs-Framework vorgeschlagen, das sich stark an der Funktionsweise des menschlichen Gehirns orientiert. Entsprechend widmet sich ein bedeutender Teil dieser Thesis der Identifizierung und Interpretation von neurowissenschaftlichen Erkenntnissen. Diese Erkenntnisse werden analysiert und in ein Computerframework übertragen, wobei jeweils die Relation der einzelnen Komponenten des Frameworks zum biologischen Lernen verdeutlicht wird.

Das Framework besteht aus drei Komponenten: Dem Sensorsystem *S0*, welches Low-Level Merkmale aus den Bildern extrahiert; der Feature-Building Stage *S1*, welche mithilfe lateralen (intra-layer) Verbindungen Neuronengruppen, sogenannte Netzfragmente, bildet, die sich gegenseitig stützen und dadurch bekannte Muster stabilisieren; der Prototyp Stage *S2*, welche die gebildeten Netzfragmente mittels Projektionsphasern zu Objekt-Prototypen mappt sowie Feedback an *S1* gibt. Der Projektionsprozess zwischen *S1* und *S2* ist iterativ und dauert bis eine Konsistenz an jedem Punkt im Netzwerk erreicht wird, d.h. bis Zellen und Synapsen einen stabilen Zustand erreicht haben.

Während frühere Forschung bereits die Effizienz von Projektionsphasen gezeigt hat, ist die Implementierung von Netzfragmenten in *S1* mehrheitlich unerforscht. Folglich wird in dieser Arbeit die Implementierung dieser Komponente im Detail untersucht und anhand von Experimenten auf einem einfachen Datensatz mit geraden Linien diskutiert. Die Ergebnisse der Experimente zeigen, dass laterale Verbindungen, trainiert mit Hebbian Learning, tatsächlich zum Zellsupport genutzt werden können. Mithilfe des Zellsupports weist das Netzwerk eine deutlich höhere Robustheit gegenüber Rauschsignalen auf und kann bis zu 91.7% der unerwünscht durch Störsignale aktivierten Zellen deaktivieren. Zudem können unterbrochene Linien aufgrund der lateralen Unterstützung wiederhergestellt werden. Mit einer Reichweite der lateralen Verbindungen von 11 Pixeln können Unterbrechungen von bis zu 8 Pixeln rekonstruiert werden, mit zusätzlichem Feedback von *S2* sogar Unterbrüche bis zu 20 Pixel. Eine Ausarbeitung des vorgeschlagenen Frameworks könnte Schwächen von neuronalen Netzwerken reduzieren und wird als vielversprechende alternative Forschungsrichtung angesehen.

# Abstract

In the past decade, deep learning has established itself as state-of-the-art technology in various automatic image analysis tasks. Despite impressive results, this technology has several limitations, notably its limited robustness to noise, constrained transformation invariance during object recognition and reliance on a substantial amount of training data. Conversely, the human brain does not suffer from these limitations due to its non-sequential processing of extracted image features and its ability to perceive visual scenes holistically, i.e. interpret it as more than the sum of its part, as outlined by Gestalt psychology. This capability stems from the brain's ability to establish internal consistency between each connected cell pair through self-organisation and localised learning, i.e. a consensus is achieved across all features through mutual cell support. This mechanism solves the problem of "early commitment" inherent in deep networks as they rely on a global error correction algorithm to establish consistency at a single point between prediction and teaching signal.

This thesis builds upon these insights and proposes a novel image-processing framework inspired by the human brain's functionality. Accordingly, a significant part of this thesis is devoted to identifying and interpreting neuroscientific findings. These findings are analysed and translated into a computational framework, thereby linking each model component to the corresponding biological mechanism.

The framework consists of three components: The sensor system *S0*, responsible for extracting low-level features from the images; the feature-building stage *S1*, which uses lateral (intra-layer) connections to form neuron groups, so-called net fragments, fostering mutual support to stabilise known patterns; the prototype stage *S2*, which maps the formed net fragments to object prototypes using projection fibres and provides feedback to *S1*. The iterative projection process between *S1* and *S2* lasts until consistency is achieved at every point in the network, i.e. until cells and synapses have reached a stable attractor state.

While prior research has demonstrated the efficiency of projection fibres, implementing net fragments still needs to be explored. Consequently, this thesis analyses the implementation of this component in detail and discusses it by conducting experiments with a simple dataset based on straight lines. The experimental findings demonstrate that lateral connections trained with Hebbian learning can facilitate cell support effectively. The network exhibits significant robustness using cell support and can deactivate up to 91.7% of unwanted cell activity triggered by noise signals. Furthermore, lateral support can restore discontinuous lines, demonstrating the network's ability to deal with occluded objects. With a range of lateral connections of 11 pixels, interruptions of up to 8 pixels can be reconstructed, and with additional feedback from *S2*, even interruptions of up to 20 pixels can be restored. Improving the proposed framework can potentially reduce several weaknesses of conventional neural networks in the future and is considered a promising alternative research direction.

# Preface

I begin this preface by providing readers with insights about the background of this Master's thesis. It is important to understand this thesis as a first step towards a broader endeavour - as preparatory work for a potential dissertation. Hence, this thesis should not be seen as self-contained work but rather as a (hopefully exciting) conceptual foundation that will be further developed and refined in the upcoming years. Many aspects of the framework proposed in this thesis still require clarification or validation through experimental evidence. I kindly ask you, the reader of this thesis, to understand if certain concepts have not yet been fully explored. I hope you can see the value of these ideas and that they will arouse your curiosity.

Upon successful assessment of this thesis and achieving a satisfactory grade, I am entitled to use the title "Master of Science". Thus, I should "master" science or at least be able to work scientifically. Science can be defined as the systematic analysis of the real or virtual world through observations and experiments and the further development of existing technology. While this definition sounds straightforward, working successfully in science requires a lot of experience and commitment. Throughout my Master's studies, I have had the privilege of delving deeper into science and applying the obtained knowledge in research projects at the Centre for Artificial Intelligence (CAI) of the Zurich University of Applied Sciences (ZHAW). This daily engagement with scientific work has proven extremely valuable in writing this thesis and will undoubtedly help me in the future.

However, I still faced challenges when writing this thesis: As someone with an industry and engineering background, I tend to approach tasks with a "do-it" mentality and intuitively refine my ideas with hands-on experimentation. In this Master's thesis, it took me some time to transition from this do-it mindset to that of a researcher who is also strong in theoretical foundations. Thanks to this Master's thesis, I now better balance implementation and studying theory.

Completing this thesis spanned an entire year, the maximum duration allowed by my university. The process of formulating and refining the theory was punctuated by setbacks and experiments that did not produce the desired results. After approximately seven months of work, I developed two biologically inspired models that showed promising performance. In fact, I had even written an entire thesis about these models, which, with some improvements, could have been submitted. However, as these concepts did not entirely convince me as a foundation for a dissertation, I decided to discard my work and pursue a new approach only five months before the submission deadline. Consequently, this thesis contains only a fraction of the conducted experiments but builds on the experience gained from earlier failed attempts. Nevertheless, I am convinced that this decision was the right one and that the quality and consistency of a thesis should outweigh the quantity. I hope you, dear reader, agree with this perspective and understand these constraints that led to incomplete experimentation.

Throughout my Master's studies and my work at the CAI, I have had the privilege to be supported and mentored by Prof. Dr. Thilo Stadelmann, Head of the CAI. He emphasises the idea that (also in accordance with his blog-post) "Great methodology delivers great theses". While it is undoubtedly desirable to achieve an exceptional result in a thesis, it is equally, if not more, important to articulate the rationale behind the methodology, justify choices and demonstrate the limitations. Moreover, scientific breakthroughs always require courage to try something completely new, even if this means that the work may not lead to outstanding results worth publishing. These principles have guided me in writing this thesis, and I hope that readers will be able to understand my thought process.

This thesis spans the fields of computer science and neuroscience and attempts to make clear connections between these areas so that readers from both disciplines can understand my arguments. However, this also means that some aspects are described rather extensively. So if you as a reader

consider yourself an expert in one of these fields, feel free to skip (parts of) the fundamentals in Chapter 2.

At this point, I also want to thank colleagues and friends for supporting this thesis. Foremost, I think of my mentor Prof. Dr. Thilo Stadelmann, who got me excited about AI years ago and later introduced me to research. He always encourages creative ideas, thinking outside the box, and striving for greatness. Thank you for your support, help, and guidance; I have grown personally and professionally. Further thanks go to Dr. Jan Deriu. He has always helped to translate abstract ideas into concrete algorithms and to get them running. It's impressive how your understanding of deep learning, algorithms, and math can make complex problems look so simple. To Prof. Dr. Christoph von der Malsburg for his seemingly endless patience in introducing me to neuroscience. You have inspired me regularly with ideas and opened up a new way of thinking about biological and artificial learning (this was also the inspiration for using Geoffrey E. Hinton's quote at the beginning of this thesis, although I wouldn't presume to say that I will change the future). Even though we couldn't implement all of your ideas in this thesis, I learned a lot in our discussions and hope that I will be able to tackle more of your thoughts in the future.

The most profound thanks, however, goes to my family, who made this journey possible for me: To my parents, who supported and encouraged me in every way. To my younger brother, who inspired me to study. To my wife and son, who have been understanding and supportive and have always been the perfect counterbalance to the daily routine. Without the support of my family, I would never have been able to embark on this academic path.

# Contents

# List of Figures

# Mathematical Terms & Definitions

## 1 Notation

Depending on the context, a variable can be a scalar value, a vector, or a matrix. The following formatting is used:

| Formatting | Example | Meaning |
|---|---|---|
| No formatting, lower case | $a$ | A scalar value. |
| Bold, lower case | $\boldsymbol{a}$ | A vector. |
| Bold, upper case | $\boldsymbol{A}$ | A matrix. |
| Curved brackets | $a(\cdot)$ | A function, where $(\cdot)$ is a placeholder for a variable. |
| Rectangular brackets | $a[t]$ | Variable $a$ at time $t$. |
| Superscript rectangular brackets | $a^{[l]}$ | A variable within a neuronal layer $l$, e.g. $\boldsymbol{W}^{[l]}$ is the weight matrix of layer $l$. |

## 2 Variables

| Variable | Meaning |
|---|---|
| $\eta$ | The learning rate of an optimisation algorithm. |
| $\kappa$ | The number of alternative cells. |
| $\psi$ | Estimated pre- or post-synaptic activity of a cell, required for Hebbian learning (c.f. equation (2.16)). |
| $\rho$ | Upper limit for cell support before the support is decreased. |
| $\boldsymbol{a} = (a_1, ..., a_k)$ | The output of an intermediate layer of a multi-layer network with $k$ neurons, the output of the last layer is typically denoted as $\hat{\boldsymbol{y}}$. $a_i$ is the output of an activation function of a single neuron. |
| $\boldsymbol{b} = (b_1, ..., b_k)$ | The bias of a network layer with $k$ neurons. $b_i$ is the bias of a single neuron. |
| $C$ | Either the capacity of a Hopfield network (the number of patterns that can be stored) or the number of channels from an input matrix. |
| $\boldsymbol{h} = (h_1, ..., h_k)$ | Hidden state of the memory storing object prototypes. |
| $H$ | The height of an image. |
| $k$ | The number of neurons within a layer or kernel. |
| $L$ | The number of layers of a network. |
| $n$ | The length (size) of a vector, for example, an input of length $n$ is defined as $\boldsymbol{x} = (x_1, ..., x_n)$. $n_l$ is also used for the support distance of lateral connections. |
| $o$ | A single neuronal cell. |

| Variable | Meaning |
|---|---|
| $s$ | A power factor to push most activation probabilities towards 0 and only permit high activations to remain high (i.e. $\boldsymbol{a} := \boldsymbol{a}^s$). |
| $t$ | The current timestep. |
| $T$ | The total number of timesteps. |
| $w_{ij}$ | The weight between neuron $i$ and neuron $j$. |
| $\boldsymbol{w} = (w_1, ..., w_n)$ | A weight vector of a neuron, mapping an input of length $n$ to a scalar value. |
| $W$ | The width of an image. |
| $\boldsymbol{W} = (\boldsymbol{w}_1, ..., \boldsymbol{w}_k)$ | A weight matrix of a network layer. |
| $\boldsymbol{x} = (x_1, ..., x_n)$ | An input sample that is fed into a model or a network layer, typically a vector of length $n$. |
| $\boldsymbol{y}$ | The expected output of a model or a network layer (i.e. the ground truth), typically a vector ($\boldsymbol{y}$) for a multi-class classification task or a scalar value ($y$) for a regression or single-class classification task. |
| $\hat{\boldsymbol{y}}$ | The actual output of a model or a network layer (i.e. the prediction), typically a vector ($\hat{\boldsymbol{y}}$) for a multi-class classification task or a scalar value ($\hat{y}$) for a regression or single-class classification task. |
| $\boldsymbol{z} = (z_1, ..., z_k)$ | The output of the aggregation functions $g_1(\cdot), ..., g_n(\cdot)$ of a network layer of length $k$. $z_i$ is the output of the aggregation function of a single neuron. |
| $Z$ | Capacity of the memory. |

# 3 Functions

| Function | Meaning |
|---|---|
| $B(\cdot)$ | A Bernoulli probability distribution. |
| $f(\cdot)$ | An activation function such as $\sigma(\cdot)$, $(\cdot)^+$, or $\tanh(\cdot)$ (c.f. equation (2.6)). |
| $F(\cdot)$ | The free energy function. |
| $g(\cdot)$ | The aggregation function that calculates the scalar value that is fed into the activation function $f(\cdot)$ (c.f. equation (2.1), equation (2.5)). |
| $J(\cdot)$ | The Jaccard similarity between two binary vectors (c.f. equation (5.6)). |
| $\mathcal{L}(\cdot)$ | A loss function to calculate the quality of the model output, typically comparing a prediction $\hat{\boldsymbol{y}}$ with a teaching signal $\boldsymbol{y}$, i.e. $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ |
| $P(\cdot)$ | The probability of observing a variable. |

# 1 Introduction



Deep learning systems have been employed for decades [1] and pervasive influence our daily lives. This technology is utilised in diverse applications, such as machine translation [2], image analysis [3], natural language processing [4], and speech synthesis [5], among others. Deep learning methodologies' remarkable success and widespread adoption have revolutionised various industries [6], enabling enhanced performance and efficiency in numerous tasks and services. Recently, ChatGPT [7] was introduced, marking a pivotal moment in the widespread application of AI systems.

Deep learning systems are trained to optimise a specific target function, such as predicting the subsequent word token in the case of language models [8]. Therefore, such systems rely on statistical patterns rather than genuine understanding and still lack cognitive capabilities, reasoning ability, self-awareness, and intentionality [9], [10]. Furthermore, deep networks suffer from various issues of statistical learning, such as missing robustness [11], catastrophic forgetting [12], or requiring vast amounts of data [13].

Geoffrey E. Hinton, recipient of the Turing Award[1] and a prominent figure in the field, is considered one of the pioneers of deep learning. His remarkable contributions, including improving deep learning's error correction algorithm called "backpropagation of error" [14], [15] (c.f. Section 2.2), have laid the foundation for today's deep learning systems. After working over three decades in the field, Hinton expresses deep scepticism about end-to-end backpropagation of errors and even suggests to "throw it all away and start again" to improve current systems fundamentally [16]. While this view may seem extreme, it shows that the learning algorithm of such systems has significant shortcomings. Some of the most crucial limitations of deep learning systems are discussed in Section 2.3. To overcome these challenges, researchers have proposed diverse methods and approaches [17]–[19]. Despite these efforts, progress has been moderate, primarily focusing on mitigating the symptoms rather than addressing the core issues.

In this thesis, a novel learning framework based on neuroscientific findings is introduced to address the core issues inherent in deep learning systems. Inspired by the human brain's remarkable learning capabilities, this research aims to integrate neuroscientific insights into an image-processing framework. It seeks to overcome limitations and bridge the gap between artificial and biological intelligence. Significant inspiration for this thesis is drawn from the "Theory of Natural Intelligence" proposed by von der Malsburg *et al.* [20]. By anchoring the theoretical concepts of this theory in a concrete framework and exemplifying its learning capabilities through practical implementation, this thesis contributes to a comprehensive understanding of the theory's principles.

[6]: Bertolini, Mezzogori, Neroni, *et al.* (2021), 'Machine Learning for industrial applications'

[7]: Liu, Han, Ma, *et al.* (2023), *Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models*

[8]: Radford, Narasimhan, Salimans, *et al.* (2018), *Improving language understanding by generative pre-training*

[9]: Rosenbloom (2023), *Defining and Explorting the Intelligence Space*

[10]: Mitchell and Krakauer (2023), 'The debate over understanding in AI's large language models'

[11]: Akhtar and Mian (2018), 'Threat of Adversarial Attacks on Deep Learning in Computer Vision'

[12]: Kirkpatrick, Pascanu, Rabinowitz, *et al.* (2017), 'Overcoming catastrophic forgetting in neural networks'

[13]: Smith, Patwary, Norick, *et al.* (2022), *Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model*

1: The Turing Award is recognised as the highest academic award in computer science and sometimes also called the "Nobel Prize of Computing".

[14]: Rosenblatt (1962), *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

[15]: Linnainmaa (1976), 'Taylor expansion of the accumulated rounding error'

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

## 1.1 Motivation

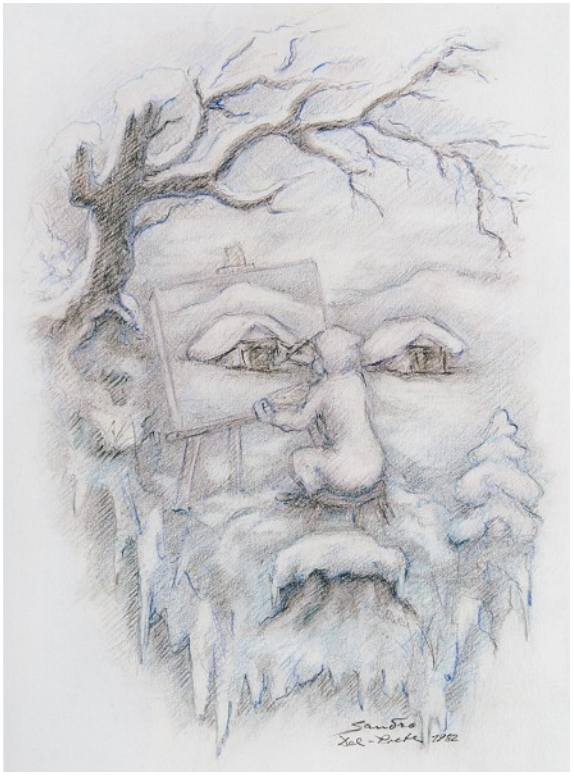

**Figure 1.1:** "Mountain Spirit in Winter" by del Prete [21] demonstrating that humans immediately can observe the overall pattern (a man's face) even though local features depict a painter drawing a house.

One of the insights from the *Gestalt* psychology [22]–[25] is that humans can recognise the "Gestalt" (the entire structure) of an object within a very short time; The brain can immediately recognise global patterns - arrays of local features that consistently conform to a known large-scale pattern - even if those local features are buried in noise or would, based on local context, be interpreted very differently. Thus, local decisions are made based on plausibility considering overall patterns, while the overall patterns can only be defined based on local features.

[22]: Wertheimer, Köhler, Fuchs, *et al.* (1938), *A source book of Gestalt psychology.*

[23]: Köhler (1992), *Gestalt psychology*

[24]: Wagemans, Feldman, Gepshtein, *et al.* (2012), 'A century of Gestalt psychology in visual perception'

[25]: Hamlyn (2017), *The Psychology of Perception*

For example, when looking at Figure 1.1, local and global patterns are not aligned. A painter drawing a picture from a house in a snowy landscape can be observed when looking at local features. However, a man's face is visible when looking at the overarching pattern. This example illustrates that avoiding the "fallacy of early commitment" is essential, as David Marr put it [26]. Otherwise, when focusing on local features only, systems would commit to objects like trees, a painter, an image, a house, and snow and be unable to recognise the global pattern, as a men's face typically comprises eyes, a nose, etc. and not the aforementioned objects.

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

The theory of natural intelligence [20] and work based on self-organising projection fibres [27]–[32] (c.f. Section 3.2.1) consider the principle of preventing early commitment as a core mechanism for the effectiveness of the human's visual system. Preventing early commitment allows a system to leave multiple options open simultaneously: The system does not take decisions early in the learning process as typical deep learning models do but considers local and global features simultaneously, continuously ruling out implausible hypotheses.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

The most used architectures for image processing are based on convolutional neural networks [33]–[35] (CNNs). This type of network cannot prevent early commitment by design: The first layers extract local features from images, only having access to small patches [35]. The extracted low-level features are combined into higher-level features in later layers [36]. Thus, the first layers do not consider global features but steer the decision process during training and inference toward specific directions based on local features. Therefore, CNNs take local decisions without consolidating global information[2].

Transformer-based [37] or fully connected [38] vision architectures, on the other hand, might not have this design limitation since early layers can access the entire input. However, they have a fallacy of early commitment because they process the input layer-wise. Typically, the input of a vision architecture is specific (e.g. an image) and mapped to more general information (e.g. a class label). However, general information is not used to confirm or validate specific information and therefore, high-level decisions can be misled by the wrong and inconsistent early decisions. Thus, the first layers make decisions on lower-level features and steer the decision process towards a specific direction without considering higher-level features, thereby being prone to early commitment as well.

The fact that early layers in neural networks make decisions without incorporating higher-level features arises from characteristics of the learning principle. Deep networks establish consistency at a specific point between a prediction and a teaching signal [39]. They employ a global error correction algorithm such as backpropagation of error [14], [15] to update all cells, aiming to enhance consistency at a specific point and thereby construct feature processing chains, i.e. the output of one neuron is used as input of a subsequent neuron. The first neurons in these chains must take decisions based on local features that are later used by neurons extracting higher-level features.

In contrast, consistency in the brain is built at every single point, i.e. between each connected cell pair [40]. The cells represent features that contribute to hypotheses rather than a single prediction. Active cells support one or multiple hypotheses while suppressing hypotheses inconsistent with the feature they represent. In a fast alignment process, increasing inhibition [41] deactivates hypotheses that are not supported by sufficient cells, resulting in a consensus among cells regarding perceptual interpretations. Thus, cells achieve consistency based on local interaction and self-organisation [42] without having an external teaching signal or global error correction [43], [44]. This fundamental different principle prevents early commitment, as discussed in Section 4.2.3.

Besides having the problem of early commitment, deep networks also lack dealing with ambiguity and object-independent transformation invariance [45], [46]. Deep networks can learn to generate transformation-invariant features if they have seen enough samples during training. However, they lack the ability to transfer the concept of a transformation, such as rotation, from one object to another.

In this thesis, a biologically inspired vision framework is proposed to mitigate these disadvantages. The framework builds consistency between each connected cell pair to prevent early commitment and deal with ambiguity. Furthermore, the applied self-organising process is decoupled

[33]: Fukushima (1980), 'Neocognitron'

[34]: Waibel, Hanazawa, Hinton, *et al.* (1987), 'Phoneme Recognition Using Time-Delay Neural Networks'

[35]: LeCun, Boser, Denker, *et al.* (1989), 'Backpropagation Applied to Handwritten Zip Code Recognition'

[36]: Prince (2023), *Understanding Deep Learning*

2: CNNs can be trained to make diverse decisions in high-level layers when appropriate labels are given. However, these decisions are still based on already taken local decisions.

[37]: Dosovitskiy, Beyer, Kolesnikov, *et al.* (2021), *An Image is Worth 16x16 Words*

[38]: Tolstikhin, Houlsby, Kolesnikov, *et al.* (2021), 'MLP-Mixer: An all-MLP Architecture for Vision'

[39]: Wang, Ma, Zhao, *et al.* (2022), 'A Comprehensive Survey of Loss Functions in Machine Learning'

[14]: Rosenblatt (1962), *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

[15]: Linnainmaa (1976), 'Taylor expansion of the accumulated rounding error'

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

[41]: Coombs, Eccles, and Fatt (1955), 'The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential'

[42]: Morris, Tarassenko, and Kenward (2006), *Cognitive Systems - Information Processing Meets Brain Science*

[43]: Grossberg (1987), 'Competitive Learning'

[44]: Crick (1989), 'The recent excitement about neural networks'

[45]: Mouton, Myburgh, and Davel (2020), 'Stride and Translation Invariance in CNNs'

from the object, allowing the model to learn the concept of transformation-invariance independent of objects. The thesis lies at the intersection of deep learning and neurocomputing[3]. It adopts many learning paradigms from neurocomputing, leveraging the capabilities of biologically inspired algorithms. At the same time, the computational efficiency of deep learning algorithms is being exploited. By merging the strengths of both fields, this work aims to develop a hybrid approach that combines the biological plausibility of neurocomputing with the computational efficiency of deep learning to foster advances in learning algorithms.

3: Neurocomputing is a subfield of neuroscience that focuses on implementing biologically plausible learning algorithms.

## 1.2 Contribution

1. The basics of deep learning and neurocomputing are summarised. Together with the related work, this provides a survey of the most important research dealing with alternative learning algorithms compared to conventional deep learning methods.
2. Many neuroscientific findings that are highly relevant for visual object recognition are identified and summarised, and a vocabulary compatible with the current deep learning framework is introduced.
3. A framework implementing the identified neuroscientific concepts is introduced based on a novel Bernoulli neuron firing a binary spike based on a probability distribution and activity received through self-organising synaptic connections.
4. The "Theory of Natural Intelligence" proposed by von der Malsburg *et al.* [20] is discussed and put into the context of the theory of self-organising projection fibres [27]–[32] and the proposed vision framework.
5. The feasibility of the proposed framework is demonstrated with experiments, the strengths and weaknesses are discussed, and directions for future research are given to further improve the proposed vision framework.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[27]: Bienenstock and von der Malsburg (1987), 'A Neural Network for Invariant Pattern Recognition'

[28]: Lades, Vorbruggen, Buhmann, *et al.* (1993), 'Distortion invariant object recognition in the dynamic link architecture'

[29]: Wiskott and von der Malsburg (1996), *Face Recognition by Dynamic Link Matching*

[30]: Wiskott, Fellous, Kuiger, *et al.* (1997), 'Face recognition by elastic bunch graph matching'

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

## 1.3 Organisation of Thesis

The remainder of the thesis is organised as follows: In Chapter 2, deep learning and neurocomputing fundamentals related to this thesis are outlined. In Chapter 3, the related work is introduced. In Chapter 4, promising biological concepts for visual object recognition are identified. Afterwards, in Chapter 5, a novel implementation of the identified biological concepts is proposed. In Chapter 6, conducted experiments with the proposed framework are described, and the obtained results are presented in Chapter 7. Finally, in Chapter 8, the thesis is concluded by discussing the advantages and disadvantages of the proposed learning framework and suggesting potential directions for future research. Thus, in Chapters 2-4, existing work is surveyed and put into context, while in Chapters 5-8, a novel vision framework is introduced and discussed.

# Fundamentals 2

The biological neuronal system not only inspired neural networks but also this thesis. Accordingly, in Section 2.1, it is explained how biological neurons work and how they are related to their artificial counterparts. Next, in Section 2.2, artificial neural networks are introduced. In Section 2.3, the limitations of such artificial neural networks are pointed out. Finally, in Section 2.4, biologically more plausible learning methods related to this thesis are explored.



## 2.1 Human Brain

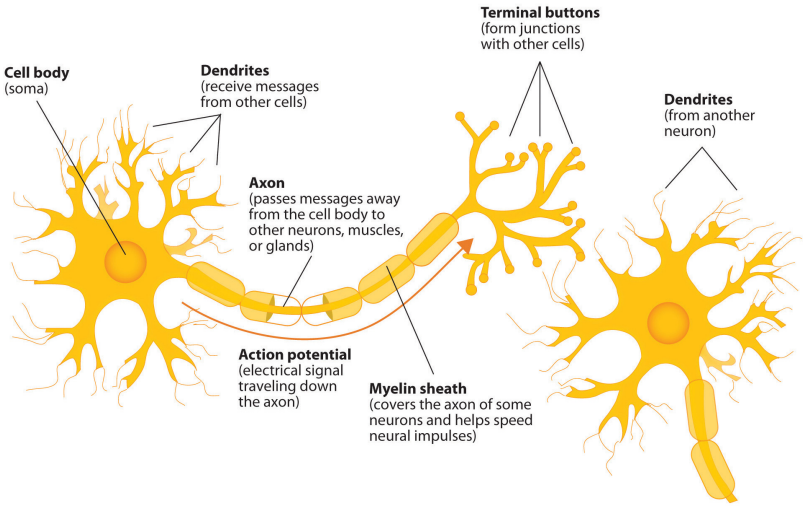


**Figure 2.1:** A diagram of the components of a biological neuron. The image is from Wikipedia [47].

A biological neuron is a cell that communicates with other neurons through precisely timed electrical pulses called spikes or action potential. Biological neurons are electrically excitable by voltage changes across their membranes. The neuron generates an action potential if the changes are significant enough within a short interval. This action potential propagates along the axon to the terminal buttons, activating synaptic connections to the dendrites of other neurons [48]. These components of a neuron are illustrated in Figure 2.1. The synaptic signal can be excitatory [49] or inhibitory [41], making the postsynaptic neuron more or less likely to fire an action potential itself. However, biological neurons do not follow strict rules; they adapt their firing rate to constant inputs, may continue firing after an input signal disappears and can even fire when no input is active [48], [50].

Biological neurons can be classified into sensory neurons, motor neurons, and interneurons. Sensory neurons respond to external stimuli such as light or sound and send signals to the spinal cord or the brain. Motor neurons receive brain and spinal cord signals to control muscles or organs. Interneurons establish connections between neurons within the same

[48]: Diamond (2019), 'Identifying what makes a neuron fire'

[49]: Takagi (2000), 'Roles of ion channels in EPSP integration at neuronal dendrites'

[41]: Coombs, Eccles, and Fatt (1955), 'The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential'

[50]: Wilson and Groves (1981), 'Spontaneous firing patterns of identified spiny neurons in the rat neostriatum'

brain or spinal cord region. However, this classification is a simplification, as the human brain consists of approximately 100 billion neurons [51] with diverse molecular, morphological, connectional, and functional properties [52].

Inspired by biological neurons, several variants of artificial neurons [53]–[55] have been proposed and are discussed in Section 2.2. Compared to their biological counterparts, artificial neurons are simplified models, ignoring much complexity. Like biological neurons, artificial neurons are usually connected to other neurons, forming an artificial neural network (ANN) [36]. Although the neurons in the first layer of an artificial network could be considered sensory neurons, the neurons in the last layer could be considered motor neurons, and the neurons in the middle layer could be considered interneurons, this distinction is less meaningful because artificial neurons always have the same inner structure regardless of their position [56], except for variations in their activation function [55].

Furthermore, artificial neural networks have a simpler organisational structure than the human brain. ANNs typically consist of one or a few network parts, such as encoders, which map data to a latent space, and decoders, which convert data from the latent space into a target vector [56]. ANNs are considered monolithic because they are hierarchical and typically trained in an end-to-end fashion with a single error correction signal [57]. In contrast, the human brain consists of many independent and interconnected organisational units, each responsible for a specific function [58]. For example, in the cerebral cortex, numerous small subunits exist [58], dedicated to specific tasks as illustrated in Figure 2.2. Furthermore, the human brain does not comprise an organisational hierarchy as ANNs [59], [60]. In the human brain, each unit applies similar deterministic functions to the information it receives [59], [60]. Furthermore, the biological neural network in the human brain is dynamic and subject to constant change through growth and reorganisation, known as neuroplasticity or neuronal plasticity [61].

Besides structural and functional differences, biological and artificial networks differ in their learning strategies. An artificial learning system requires a feedback signal from which it can learn. This is called the *credit assignment problem*. Backpropagation of error [14], [15] is the state-of-the-art algorithm that solves this problem by propagating the error signals back through the network [62]. However, information in the brain flows only in one direction, from presynaptic to postsynaptic neurons. Therefore, backpropagation of error is not biologically plausible [43], [44]. The brain relies on localised learning [63], where each unit adapts its behaviour based on the information it receives. Evidence suggests that the brain learns by connecting cells that are active simultaneously, known as Hebbian plasticity [40], thereby not relying on an external teaching signal.

Thus, among the most important differences between biological and artificial neurons is the time-dependent and asynchronous firing of biological neurons compared to the synchronous firing of artificial neurons. In addition, biological networks have different types of neurons and rely on local learning, while artificial networks usually comprise identical neurons and utilise a global error correction algorithm. Finally, biological networks are organised in a complex way comprising multiple

[51]: Herculano-Houzel (2009), 'The human brain in numbers'

[52]: Zeng and Sanes (2017), 'Neuronal cell-type classification'

[53]: McCulloch and Pitts (1943), 'A logical calculus of the ideas immanent in nervous activity'

[54]: Rosenblatt (1958), 'The perceptron'

[55]: Fukushima (1969), 'Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements'

[36]: Prince (2023), *Understanding Deep Learning*

[56]: Goodfellow, Bengio, and Courville (2016), *Deep Learning*

[57]: Glasmachers (2017), 'Limits of End-to-End Learning'

[58]: Felleman and Van Essen (1991), 'Distributed Hierarchical Processing in the Primate Cerebral Cortex'

[59]: Mountcastle (1978), 'An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System'

[60]: Mountcastle (1997), 'The columnar organization of the neocortex'

[61]: Costandi (2016), *Neuroplasticity*

[14]: Rosenblatt (1962), *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

[15]: Linnainmaa (1976), 'Taylor expansion of the accumulated rounding error'

[62]: Rumelhart, Hinton, and Williams (1986), 'Learning representations by back-propagating errors'

[43]: Grossberg (1987), 'Competitive Learning'

[44]: Crick (1989), 'The recent excitement about neural networks'

[63]: Lillicrap, Santoro, Marris, *et al.* (2020), 'Backpropagation and the brain'

units [58], while artificial networks are typically structured in layers [56].

[58]: Felleman and Van Essen (1991), 'Distributed Hierarchical Processing in the Primate Cerebral Cortex'

[56]: Goodfellow, Bengio, and Courville (2016), *Deep Learning*

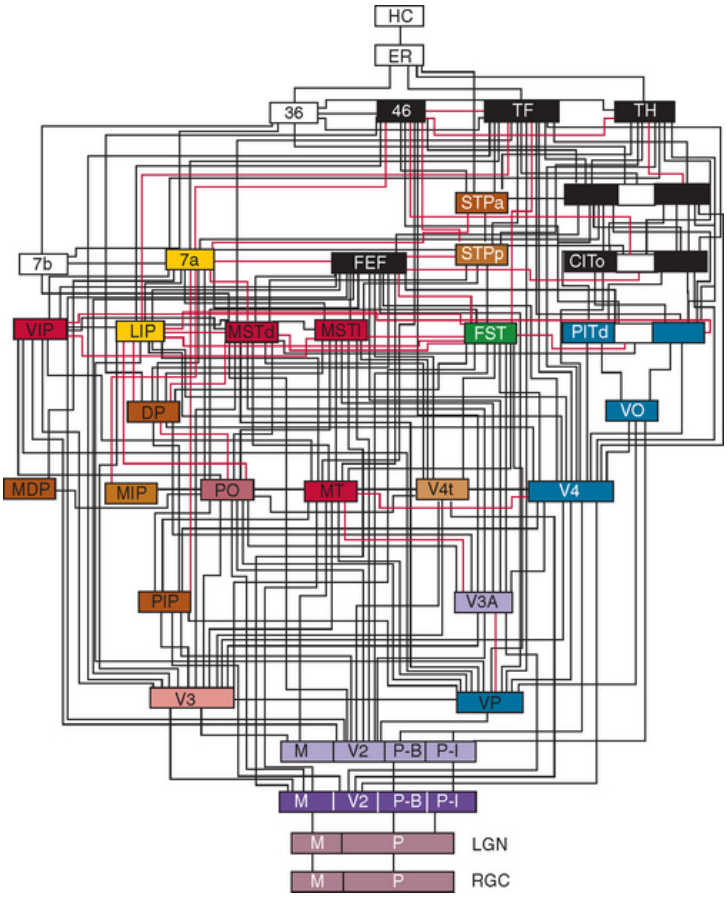


**Figure 2.2:** The organisation of the visual system in the cerebral cortex. The image is from Felleman *et al.* [58].

Biological findings strongly inspire the proposed vision framework. Since biological systems still have many advantages compared to their artificial counterpart, the differences between these systems are of interest and could be promising in developing novel architectures. Therefore, in Section 4.1, biological principles related to the human's vision system are examined in more detail.

## 2.2 Artificial Neural Networks

McCulloch and Pitts [53] proposed the first model of a neuron that can be connected to other neurons to form a network. Like the biological neuron, the artificial neuron of McCulloch and Pitts receives multiple input signals and transforms them into an output signal. Their neuron takes a binary input vector $\boldsymbol{x} = (x_1, ..., x_n)$ where $x_i \in \{0, 1\}$ and maps it to an output $\hat{y} \in \{0, 1\}$. The mapping from the input to the output is done by using an aggregation function $z = g(\boldsymbol{x})$ that sums up the input vector $\boldsymbol{x}$ and an activation function $f(z)$ that outputs 1 if $z$ is bigger than

[53]: McCulloch and Pitts (1943), 'A logical calculus of the ideas immanent in nervous activity'

a threshold $\theta$ and 0 otherwise:

$$z = g(\boldsymbol{x}) = g(x_1, ..., x_n) = \sum_{i=1}^{n} x_i \tag{2.1}$$

$$\hat{y} = f(z) = \begin{cases} 1, & \text{if } z \geq \theta \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

The formula is often rewritten by using a bias $b$ instead of the threshold $\theta$:

$$z = g(\boldsymbol{x}) = g(x_1, ..., x_n) = \sum_{i=1}^{n} x_i + b \tag{2.3}$$

$$\hat{y} = f(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

By adjusting the bias $b$, neurons can learn to fire under different conditions. In 1958, Rosenblatt [54] proposed the perceptron, which extends the neuron with additional learnable weights: The input vector $\boldsymbol{x}$ of length $n$ is multiplied with a weight vector $\boldsymbol{w} \in \mathbb{R}^n$ of the same length.

[54]: Rosenblatt (1958), 'The perceptron'

$$z = g(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b = \left( \sum_{i=1}^{n} w_i \cdot x_i \right) + b \tag{2.5}$$

The weight vector $\boldsymbol{w}$ remains identical if the output $\hat{y}$ corresponds to the desired output $y$ or is adjusted otherwise (c.f. Section 2.2.3). Later, the activation function $f(z)$ was replaced with other functions so that the output can be a real number $\hat{y} \in \mathbb{R}$ [55]. Often-used activation functions are

[55]: Fukushima (1969), 'Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements'

$$\text{Sigmoid: } f(z) = \sigma(z) = \frac{1}{1 + \mathrm{e}^{-z}} \tag{2.6}$$

$$\text{Rectified linear unit (ReLU): } f(z) = (z)^+ = \max\{0, z\} \tag{2.7}$$

$$\text{Hyperbolic tangent (tanh): } f(z) = \tanh(z) = \frac{\mathrm{e}^{z} - \mathrm{e}^{-z}}{\mathrm{e}^{z} + \mathrm{e}^{-z}} \tag{2.8}$$

To summarise, artificial neurons multiply incoming signals with learned weights and add a bias before applying a non-linear activation function. Unlike their biological counterparts, artificial neurons exhibit non-temporal behaviour and output continuous values instead of discrete binary spikes. Neurons with time-dependent spike patterns are discussed in Section 2.4.3.

### 2.2.1 Fully Connected Layer

Artificial neural networks consist of several neurons organised in a network. These neurons are typically arranged in layers [36]. In a basic configuration, each neuron in one layer is connected to each neuron in the following layer, forming a so-called fully connected layer [36], [56].

[36]: Prince (2023), *Understanding Deep Learning*

[56]: Goodfellow, Bengio, and Courville (2016), *Deep Learning*

In a fully connected layer with $k$ neurons, the input $\boldsymbol{x} \in \mathbb{R}^n$ is multiplied with a weight matrix $\boldsymbol{W} \in \mathbb{R}^{n \times k}$ and a bias $\boldsymbol{b} \in \mathbb{R}^k$ is added to obtain the

layer's output $\hat{\boldsymbol{y}} \in \mathbb{R}^k$:

$$\boldsymbol{z} = \boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b} \tag{2.9}$$

$$\hat{\boldsymbol{y}} = f(\boldsymbol{z}) \tag{2.10}$$

The universal approximation theorem [64] states that a shallow network with one hidden layer (i.e. one layer between input and output layer) and enough neurons can approximate any mapping function between inputs and outputs. However, a sequential arrangement of multiple layers is more efficient for complex functions [36]. This hierarchical approach allows the network to learn a hierarchy of features and capture intricate patterns [65].

[64]: Cybenko (1989), 'Approximation by superpositions of a sigmoidal function'

[36]: Prince (2023), *Understanding Deep Learning*

[65]: Bengio (2012), 'Deep Learning of Representations for Unsupervised and Transfer Learning'

In a multi-layer perceptron (MLP) with $L$ layers, the input $\boldsymbol{x}$ is passed through each layer, whereby each subsequent layer $l$ uses the output of the previous layer $l-1$ as input. In the following, the weights of layer $l$ are denoted as $\boldsymbol{W}^{[l]}$, the bias as $\boldsymbol{b}^{[l]}$, the output of the aggregation function as $\boldsymbol{z}^{[l]}$, and the output of the activation function as $\boldsymbol{a}^{[l]}$. The input in the first layer is the input data itself, i.e. $\boldsymbol{a}^{[0]} = \boldsymbol{x}$, and the output of the last layer corresponds to the model's prediction, denoted as $\boldsymbol{a}^{[L]} = \hat{\boldsymbol{y}}$. With this notation, the mathematical formulation of an MLP can be defined as follows:

$$\boldsymbol{z}^{[l]} = \boldsymbol{W}^{[l]}\boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]} \tag{2.11}$$

$$\boldsymbol{a}^{[l]} = f(\boldsymbol{z}^{[l]}) \tag{2.12}$$

A significant drawback of this approach is that layer-wise data processing leads to early commitment [26], as outlined in Section 1.1. The problem is that the first layers already commit to representations that are further processed by subsequent layers, thereby steering the learning process in specific directions without considering higher-level features. The framework proposed in this thesis introduces a novel layer that builds hierarchical representations within the same layer (c.f. Section 5.1.1), efficiently preventing the fallacy of early commitment.

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

### 2.2.2 Convolutional Networks

A problem of fully connected layers is that they are not position invariant, meaning they cannot recognise patterns regardless of their location in the input. Convolutional neural networks (CNNs) [33]–[35] are inspired by biological processes [55], [66] and can overcome this limitation: CNNs exhibit position equivariance by applying the same weights at all input positions [45]. A typical CNN consists of subsequently connected convolutional layers and pooling layers. Usually, an activation function is applied after each convolutional layer, while no activation function is used after pooling layers [56].

[33]: Fukushima (1980), 'Neocognitron'

[34]: Waibel, Hanazawa, Hinton, *et al.* (1987), 'Phoneme Recognition Using Time-Delay Neural Networks'

[35]: LeCun, Boser, Denker, *et al.* (1989), 'Backpropagation Applied to Handwritten Zip Code Recognition'

[55]: Fukushima (1969), 'Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements'

Convolutional layers employ convolution filters or kernels that slide along the input, generating translation-equivariant [45] responses known as feature maps [67]. Translation-equivariant means that the relative

placement of objects remains consistent between the layer's input and output, as the same filter is applied at all image positions. During the filtering process, the dot product is calculated between the filter and an input area (of the same size as the filter), resulting in a scalar value that is assigned to one position of the output matrix (i.e. the feature map) [56]. This process is repeated by shifting the filter by a specified stride until the entire input is processed and all values in the output matrix are calculated. Convolutional layers require much fewer parameters than fully connected layers of the same size because only the kernels need to be learned. This process of reusing the same weights at different input locations is known as parameter sharing [45]. By stacking multiple layers, CNNs become hierarchical: The convolutional operation squeezes information from surrounding pixels into a single output cell. Thus, using multiple layers sequentially continuously enlarges the receptive field - the area of input pixels that can influence a single value in a layer's feature map [65].

[56]: Goodfellow, Bengio, and Courville (2016), *Deep Learning*

[45]: Mouton, Myburgh, and Davel (2020), 'Stride and Translation Invariance in CNNs'

[65]: Bengio (2012), 'Deep Learning of Representations for Unsupervised and Transfer Learning'

Pooling layers downsize the input rather than extracting features [68]. Similar to convolutional layers, they slide a filter along the input. However, unlike convolutional layers, pooling filters do not have learned parameters but use an aggregation function. Usually, the filter selects the pixel with the highest value (max pooling) or calculates the average (average pooling) within the considered input area and uses this value as output [68]. The filter is then shifted by its size, ensuring that non-overlapping image patches are processed. Pooling layers discard a considerable amount of information but effectively reduce complexity and improve the model's robustness [68]. Nevertheless, discarding valuable information can also be the reason for misclassification [69].

[68]: Cireşan, Meier, Masci, *et al.* (2011), 'Flexible, High Performance Convolutional Neural Networks for Image Classification'

[69]: Sharma and Mehra (2019), 'Implications of Pooling Strategies in Convolutional Neural Networks'

Besides CNNs, alternative architectures for image processing have emerged that are characterised by position invariance, such as the vision transformer (ViT) [37] or MLP mixer [38]. However, providing an in-depth description of these architectures would exceed the scope of this introduction to deep learning.

[37]: Dosovitskiy, Beyer, Kolesnikov, *et al.* (2021), *An Image is Worth 16x16 Words*

[38]: Tolstikhin, Houlsby, Kolesnikov, *et al.* (2021), 'MLP-Mixer: An all-MLP Architecture for Vision'

### 2.2.3 Learning Algorithm

The model's prediction $\hat{\boldsymbol{y}}$ will only be close to the target output $\boldsymbol{y}$ if the weights $\boldsymbol{W}^{[l]}$ and biases $\boldsymbol{b}^{[l]}$ are properly defined. These parameters are learned through training, typically using backpropagation of errors [14], [15]. Training can take place in a supervised, unsupervised, semi-supervised or reinforcement learning setting [70], [71].

[70]: Russell and Norvig (2021), *Artificial intelligence*

[71]: Simmler, Sager, Andermatt, *et al.* (2021), 'A Survey of Un-, Weakly-, and Semi-Supervised Learning Methods for Noisy, Missing and Partial Labels in Industrial Vision Applications'

All these learning principles have in common that a loss function [39] (also called an objective function) $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is used to calculate the quality of the model output $\hat{\boldsymbol{y}}$ relative to the target output $\boldsymbol{y}$. The selected loss function is minimised iteratively using an error correction algorithm such as stochastic gradient descent (SGD)[1] until the network reaches a (local) minimum. Stochastic gradient descent is based on the insight that the negative gradient of the loss value indicates the direction of the steepest descent, i.e. the direction in which the loss decreases the most. Consequently, SGD updates the parameters of the network by taking steps of size $\eta$ (the learning rate) in the direction of the negative

[39]: Wang, Ma, Zhao, *et al.* (2022), 'A Comprehensive Survey of Loss Functions in Machine Learning'

1: There also exist other optimisation algorithms such as SGD with momentum, RMSprop or Adam [72].

gradient:

$$\begin{aligned} \Delta \boldsymbol{W}^{[l]} &= -\eta \cdot (\nabla_{\boldsymbol{W}^{[l]}} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})) \\ \boldsymbol{W}^{[l]} &:= \boldsymbol{W}^{[l]} + \Delta \boldsymbol{W}^{[l]} \end{aligned} \tag{2.13}$$

and

$$\begin{aligned} \Delta \boldsymbol{b}^{[l]} &= -\eta \cdot (\nabla_{\boldsymbol{b}^{[l]}} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})) \\ \boldsymbol{b}^{[l]} &:= \boldsymbol{b}^{[l]} + \Delta \boldsymbol{b}^{[l]} \end{aligned} \tag{2.14}$$

The term $(\nabla_{\boldsymbol{W}^{[l]}} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}))$ is the gradient of the weights $\boldsymbol{W}^{[l]}$ with respect to the loss $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ and the term $(\nabla_{\boldsymbol{b}^{[l]}} \mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}))$ is the gradient of the bias $\boldsymbol{b}^{[l]}$ with respect to $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y})$. The gradients of the weights can efficiently be calculated with backpropagation of error [14], [15], which is, in fact, just an intelligent implementation of the chain rule[2].

However, from a biological perspective, backpropagation of error seems not plausible [43], [44]. Furthermore, it has technical shortcomings, such as vanishing or exploding gradients when propagating through too many layers [73] or non-optimal loss landscapes [74]. Despite these deficits, it is still considered the best error correction algorithm to optimise neural networks on a specific task and can even outperform human experts [6], [75].

## 2.3 Limitations

Deep learning systems have proven themselves as excellent feature extractors and are used to fulfil various tasks in our daily lives. Nevertheless, they have several limitations, with some of the most pressing ones elucidated below.

**Computing Resources.** Deep learning models are typically trained on modern computing infrastructure. They benefit from Moore's law [76], stating that the number of transistors in a dense integrated circuit doubles every two years, allowing models to consume exponentially more computing resources. However, the physical limits of transistor size will most likely stop this exponential growth soon [77], and the future progress of hardware remains uncertain. Furthermore, the size of modern deep learning models exhibits an even faster growth, as demonstrated by state-of-the-art large language models: ELMo from 2018 uses around 94 million parameters [78], GPT-3 from 2020 uses around 175 billion parameters [79], and Megatron-Turing NLG from 2022 has 530 billion parameters [13]. An analysis by Open AI [80] shows exponential growth in computational usage by AI models, with a doubling time of about 3.4 months, outpacing the rate of hardware progress, which has a doubling time of 2 years. Moreover, the increasing size of deep networks poses a challenge for inference on low-budget hardware such as smartphones or embedded systems [81]. Although techniques such as quantisation [82], model pruning [83], and model distillation [84] exist to reduce model size after training, the question is whether increasing model size is the best way to develop more advanced systems regarding feasibility but also regarding energy consumption [85].

[14]: Rosenblatt (1962), *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

[15]: Linnainmaa (1976), 'Taylor expansion of the accumulated rounding error'

2: While a detailed discussion of backpropagation is out of scope for this thesis, we refer interested readers to the book "Understanding Deep Learning" by Prince [36].

[43]: Grossberg (1987), 'Competitive Learning'

[44]: Crick (1989), 'The recent excitement about neural networks'

[73]: Zhang, He, Sra, *et al.* (2020), *Why gradient clipping accelerates training*

[74]: Ioffe and Szegedy (2015), 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'

[6]: Bertolini, Mezzogori, Neroni, *et al.* (2021), 'Machine Learning for industrial applications'

[75]: Buetti-Dinh, Galli, Bellenberg, *et al.* (2019), 'Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition'

[76]: Moore (1965), 'Cramming More Components onto Integrated Circuits'

[77]: Kumar (2015), *Fundamental Limits to Moore's Law*

[78]: Peters, Neumann, Iyyer, *et al.* (2018), 'Deep Contextualized Word Representations'

[79]: Brown, Mann, Ryder, *et al.* (2020), 'Language Models are Few-Shot Learners'

**Catastrophic Forgetting.** Another major issue of deep learning systems is that they suffer from catastrophic forgetting [12], [86]. If a model is trained on a specific task and afterwards trained (or fine-tuned) on another task, the model suffers a "catastrophic" drop in performance over the first task [12]. The reason for this effect is that during training on the second task, the model adjusts the parameters learned during the first task and, therefore, "forgets" the learned input-output mapping functions. Mixing all datasets or learning all tasks in parallel in a multi-task setting [87] does not seem feasible to prevent catastrophic forgetting. Instead, models should remember previously learned knowledge even if a new task is learned. Online learning [88] and lifelong learning [89] do not solve catastrophic forgetting as they only allow models to adapt better to changing conditions.

[12]: Kirkpatrick, Pascanu, Rabinowitz, *et al.* (2017), 'Overcoming catastrophic forgetting in neural networks'

[86]: Liu, Yang, and Wang (2021), 'Overcoming Catastrophic Forgetting in Graph Neural Networks'

[87]: Zhang and Yang (2022), 'A Survey on Multi-Task Learning'

[88]: Sahoo, Pham, Lu, *et al.* (2018), 'Online Deep Learning'

[89]: Parisi, Kemker, Part, *et al.* (2019), 'Continual lifelong learning with neural networks'

**Extrapolate Data Distribution.** It is questionable if deep learning models can achieve *real* generalization[3] in the current learning framework. With enough data, deep learning can achieve generalisation in the sense that the model can interpolate within the known data distribution. However, deep learning models fail to extrapolate. For example, convolutional neural networks (CNNs) do not generalise to different viewpoints unless added to the training data [46].

3: Generalisation refers to the ability of the model to adapt appropriately to previously unseen data from the same distribution.

[46]: Madan, Henry, Dozier, *et al.* (2022), 'When and how convolutional neural networks generalize to out-of-distribution category–viewpoint combinations'

**Data Hunger.** Deep learning cannot learn abstract relationships in a few trials but requires many samples and is thus data-hungry. Gary Marcus [90] showcased this problem with an example: He defines the new word "schmister" as a sister over the age of 10 but under the age of 21. He found that humans can immediately infer whether they or their best friends have any "schmister". However, modern deep learning systems lack a mechanism for learning abstractions through explicit, verbal definitions and require thousands or even more training samples[4].

[90]: Marcus (2018), *Deep Learning: A Critical Appraisal*

4: LLMs can deal with such definitions when they are put into the context of a conversation during inference but not when the example is only shown once during training.

**Casual Reasoning.** No deep learning model has been able to demonstrate causal reasoning generically. Deep learning models find correlations between input and output data but not causation [36]. Other AI approaches, such as hierarchical Bayesian computing [91] or probabilistic graphical models [92] are better at causal reasoning but do not work well for processing high-dimensional data.

[36]: Prince (2023), *Understanding Deep Learning*

[91]: Allenby, Rossi, and McCulloch (2005), 'Hierarchical Bayes Models'

[92]: Koller and Friedman (2009), *Probabilistic graphical models: principles and techniques*

**Embodiment.** Deep learning models are, to some extent, isolated since they have no embodiment and cannot interact with the world. The human body provides needs, goals, emotions, and gut feelings [93]. One could argue that the body is, therefore, a co-processor of the brain. In current deep learning systems, emotions are absent, and the goals are set externally [39]. Deep reinforcement learning [94] is a step toward dissolving this isolation as the models interact with a virtual environment. However, AI systems interacting with the real world have not worked well so far. Moravec's paradox from 1995 [95] states that "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility". This statement still seems true almost 30 years later.

[93]: Mayer (2011), 'Gut feelings'

[39]: Wang, Ma, Zhao, *et al.* (2022), 'A Comprehensive Survey of Loss Functions in Machine Learning'

[94]: (2020), *Deep Reinforcement Learning*

[95]: Moravec (1995), *Mind Children: The Future of Robot and Human Intelligence*

These drawbacks are likely due to statistical learning based on a global error correction algorithm. In contrast, the human brain does not suffer from these drawbacks, indicating that different learning principles could resolve these issues. The framework proposed in this thesis introduces new learning principles that aim to reduce computational requirements, catastrophic forgetting, and data hunger and allow better data extrapolation as outlined in Section 8.1.

## 2.4 Neurocomputing

Neurocomputing is a subfield of neuroscience that focuses on implementing learning algorithms that adhere to biological plausibility. Thus, researchers are addressing the discrepancy described in Section 2.1 and strive to develop alternative learning principles.

### 2.4.1 Hebbian Learning

Hebbian learning, as proposed by Hebb [40], implements the adaptive nature of the connections between cells in the nervous system. Hebb's description states that when an axon of cell $A$ is in close proximity to cell $B$ and consistently contributes to its firing, growth processes or metabolic changes occur in one or both cells that increase the effectiveness of cell $A$ in firing cell $B$. This description is often summarised as "neurons that fire together wire together".

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

Hebbian learning implements the update of the synaptic weight $w_{ij}$ connecting neuron $i$ to neuron $j$ according to the aforementioned principle. The weight change depends on the presynaptic activity $a_i$ of neuron $i$ and postsynaptic activity $a_j$ of neuron $j$[5]:

$$\Delta w_{ij} = \eta a_i a_j \tag{2.15}$$

5: The presynaptic and postsynaptic activity corresponds to the output of a neuron's activation function $f(\cdot)$ in the preceding and subsequent layers.

where $\eta$ is the learning rate. Hebbian learning is used in the proposed framework to increase the weights between frequently co-activated binary neurons (c.f. Section 5.5.2). This rule causes two cells that are constantly active together to form an "association" where the activation of one cell promotes the activation of the other. Therefore, a network trained according to Hebb's rule is able to "auto-associate" a pattern [40]. In other words: When some cells representing a pattern are activated, they encourage all cells associated with this pattern to activate as well. These learned patterns are called engrams [96], often related to memories in a biological context [97].

[96]: Newman (1985), *Current perspectives in dysphasia*

[97]: Liu, Ramirez, Pang, *et al.* (2012), 'Optogenetic stimulation of a hippocampal engram activates fear memory recall'

The aforementioned Hebbian rule increases the weight between two binary cells that are active together and does not change when only one or none of the cells fires. Thus, the connection can only grow stronger. However, synapses can not only grow between cells but also disintegrate [61]. This process can be implemented by leveraging the covariance of neuronal activity [98]. The covariance is positive if two neurons fire often together and negative if they do not often fire together. The following

[61]: Costandi (2016), *Neuroplasticity*

[98]: Oja (1982), 'Simplified neuron model as a principal component analyzer'

equation changes the weight relative to the covariance:

$$\Delta w_{ij} = \eta(a_i - \psi_i) \cdot (a_j - \psi_j) \tag{2.16}$$

where $\psi_i$ and $\psi_j$ are estimates of the expected pre- and postsynaptic activity[6].

The formulas above lack two important constraints. First, the weight growth has no upper or lower bound: With long training on the same patterns, a synaptic connection $w_{ij}$ constantly increases or decreases. In practice, boundaries are defined to mitigate this issue. This can be implemented by normalising the length of the weight vector [98] or by using rate-based threshold adaption [99], [100]. Second, breaking the symmetry within the network is necessary: After initialisation, many cells tend to fire simultaneously, resulting in many similar updates of the synaptic connections. However, independent neurons can encode more information and work better than dependent neurons [101]. Thus, competition between neurons is needed to encourage differentiation, allowing only a subset of connections to be updated. Well-known approaches are winner-take-all competition, using a recurrent circuit that provides a competitive signal, anti-Hebbian learning [102] (a method that adds a penalty for similarly active neurons), or adapting the activation function of the neurons to enforce a specific activity distribution [103], [104].

Hebbian learning can be considered an alternative to backpropagation of error [14], [15]. Hebbian learning does not propagate error signals backwards but relies on a self-organisation process. In this thesis, Hebbian learning is leveraged as it allows to prevent early commitment, even though self-organising processes are more complex to train than global error correction algorithms [105].

### 2.4.2 Hopfield Networks

The Hebbian rule can be used to train Hopfield networks. Hopfield networks serve as associative (i.e. content-addressable) memory systems [106], similar to the nearest neighbour algorithm [107] or memory networks [108]. In a Hopfield network, all neurons are binary and connected without self-connections, i.e. $w_{ii} = 0$. Additionally, the synaptic weights in a Hopfield network are symmetrical, meaning $w_{ij} = w_{ji}$.

An input is fed into the network by setting the neuronal activity $\boldsymbol{a}[t = 0]$ at time $t = 0$ to a specific configuration. Hopfield networks have temporal dynamics, and the output evolves over time: After the initial input is set as the network's state, the cells influence each other, and the network's state is updated until a stable attractor state is reached. The state of a neuron at time $t + 1$ depends on the state of all other neurons at time $t$ within the network:

$$z_i[t+1] = \sum_{i \neq j} w_{ij} a_j[t] + b_i \tag{2.17}$$

$$a_i[t+1] = \begin{cases} 1, & \text{if } z_i[t+1] > 0 \\ -1, & \text{otherwise} \end{cases} \tag{2.18}$$

6: The expected activity can be estimated, for example, by calculating a moving average.

[98]: Oja (1982), 'Simplified neuron model as a principal component analyzer'

[99]: Bienenstock, Cooper, and Munro (1982), 'Theory for the development of neuron selectivity'

[100]: Intrator and Cooper (1992), 'Objective function formulation of the BCM theory of visual cortical plasticity'

[101]: Simoncelli and Olshausen (2001), 'Natural Image Statistics and Neural Representation'

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

[103]: Joshi and Triesch (2009), 'Rules for information maximization in spiking neurons using intrinsic plasticity'

[104]: Teichmann and Hamker (2015), 'Intrinsic Plasticity: A Simple Mechanism to Stabilize Hebbian Learning in Multilayer Neural Networks'

[105]: Risi (2021), *The Future of Artificial Intelligence is Self-Organizing and Self-Assembling*

[106]: Hopfield (1982), 'Neural networks and physical systems with emergent collective computational abilities.'

[107]: Fix and Hodges (1989), 'Discriminatory Analysis. Nonparametric Discrimination'

[108]: Weston, Chopra, and Bordes (2015), *Memory Networks*

When the aggregation value $z_i$ is bigger than 0, the cell turns on or off otherwise. Thus, a cell can either keep its state ($a_i[t+1] = a_i[t]$) or flip ($a_i[t+1] \neq a_i[t]$). A flipping state influences all other neurons and may encourage them to flip as well. Formal proof exists that after a finite number of timesteps, an attractor state is reached [106]. Thereby, an input pattern is attracted to the closest stable pattern. Hebbian learning [40] can be used to define what the stable patterns are: With a single iteration over the training patterns, the weights $w_{ij}$ and biases $b_i$ are updated so that these patterns become attractor states [109]. This allows using the network as an associative memory, i.e. to map an input pattern to the most similar predefined stable pattern.

For a long time, Hopfield networks had two limiting factors: First, the capacity $C$, i.e. the number of patterns that can be stored, was for a network with $k$ neurons limited to $C = 0.138k$ [110]. However, this limitation could be resolved more than three decades after the introduction of the Hopfield networks; Krotov *et al.* [111] first increased the capacity to a polynomial capacity w.r.t. $k$ and Demircigil *et al.* [112] later to exponential capacity w.r.t. $k$. The second limiting factor of Hopfield networks is that only binary patterns can be stored. Recently, Hopfield networks have been extended to continuous patterns [113].

Hopfield networks remain a niche, as they perform worse than retrieval systems [114] and memory networks [108]. They also lack hierarchical pattern recognition and may require additional models to store higher-level patterns than just the input data. In the context of the proposed framework, a Hopfield network could be used as a biologically plausible memory, storing concrete instances of objects. However, such a memory is not investigated within the context of this thesis and remains an open question for future research.

[106]: Hopfield (1982), 'Neural networks and physical systems with emergent collective computational abilities.'

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

[109]: Hopfield, Feinstein, and Palmer (1983), ''Unlearning' has a stabilizing effect in collective memories'

[110]: McEliece, Posner, Rodemich, *et al.* (1987), 'The capacity of the Hopfield associative memory'

[111]: Krotov and Hopfield (2016), 'Dense Associative Memory for Pattern Recognition'

[112]: Demircigil, Heusel, Löwe, *et al.* (2017), 'On a Model of Associative Memory with Huge Storage Capacity'

[113]: Ramsauer, Schäfl, Lehner, *et al.* (2021), *Hopfield Networks is All You Need*

[114]: Kowalski (1997), *Information Retrieval Systems*

[108]: Weston, Chopra, and Bordes (2015), *Memory Networks*

### 2.4.3 Spiking Neural Networks

Biological neurons emit time-dependent spikes (c.f. Section 2.1). To transmit information, especially the firing rate (i.e. the number of spikes per second) and precise timing of the spikes are relevant [115]. The amplitude and duration of the spike matter less. So-called spiking neural networks (SNNs) incorporate the concept of time into a computational model [116]. SNNs do not transmit information in each forward pass but rather send a signal when the membrane potential reaches a threshold value[7]. The most prominent models of spiking neurons are different integrate-and-fire (IF) neurons [117]–[119]. While each model has different mathematical properties, the concept remains the same: Each neuron has a membrane potential that is increased or decreased through spikes from other neurons and decays over time or is reset when the cell emits a spike after reaching a predefined threshold.

The synaptic plasticity can be learned with an adapted version of Hebbian learning, called the spike-timing-dependent (STDP) plasticity rule [120]. This rule strengthens connections if the presynaptic neurons fire before the postsynaptic neuron and weakens the connection otherwise.

For a long time, SNN only worked for very shallow networks. In 2018, Kheradpisheh *et al.* [121] proposed a deep spiking convolutional network inspired by CNNs to overcome this limitation. This network uses

[115]: Thorpe (1990), 'Spike arrival times: A highly efficient coding scheme for neural networks'

[116]: Maass (1997), 'Networks of spiking neurons'

7: The membrane potential is related to the electrical charge of the membrane of a biological neuron.

[120]: Bi and Poo (2001), 'Synaptic Modification by Correlated Activity'

[121]: Kheradpisheh, Ganjtabesh, Thorpe, *et al.* (2018), 'STDP-based spiking deep convolutional neural networks for object recognition'

convolutional and pooling layers with IF neurons and is trained with STDP. Despite these remarkable advances in SNNs, their performance is still inferior compared to equivalent artificial neural networks [122]. Several factors contribute to this discrepancy. First, SNNs require converting inputs like images into spike representations. This process results in losing important information, including colour and texture details. Furthermore, SNNs use non-differentiable activation functions, which makes them unsuitable for training by backpropagation of error [122].

[122]: Nunes, Carvalho, Carneiro, *et al.* (2022), 'Spiking Neural Networks'

Even though spiking neurons are biologically more plausible than neurons without time dependency, they are not used in the proposed framework. The reasons are that they do not provide an obvious advantage despite their plausibility and do not seem very suitable for processing a static input (e.g. an image) on a clocked machine (e.g. a CPU).

# Related Work 3

In this chapter, the related work is summarised. In Section 3.1, diverse learning frameworks are presented that deviate from the mainstream but might be promising for building more effective learning principles. Afterwards, in Section 3.2, an introduction to a publication titled "A Theory of Natural Intelligence" by von der Malsburg *et al.* [20] is provided. This work is the main source of inspiration for this thesis and is summarised in detail. This theory can be implemented by combining projection fibres with lateral connections in a self-organising manner. Therefore, projection fibres are summarised in Section 3.2.1 and work associated with self-organisation is reviewed in Section 3.3.



## 3.1 Alternative AI Approaches

Arguably, the only existing systems that, in non-experts' eyes, behave intelligently[1] are foundation models such as large language models (LLMs) [79], [123], and their fine-tuned versions [124]. However, like other deep learning models, also foundation models suffer from the typical drawbacks of statistical learning (c.f. Section 2.3). Therefore, a lot of research has been conducted to reduce these well-known problems: For example, current research aims to reduce hallucinations [125], [126], to implement an ongoing learning framework [88], [127], to transfer knowledge between tasks and data domains [18], [128], or to learn what should be learned [129], [130]. However, the underlying framework remains identical: Data is statistically mapped to manually or automatically generated labels, making it difficult to solve problems such as lack of robustness [11], [17], out-of-distribution generalisation [46], data efficiency [13], [90], energy consumption [85], catastrophic forgetting [12], causal understanding or common sense reasoning [9], [10]. In the following, alternative approaches to training deep networks are summarised that attempt to alleviate these problems with new principles that do not rely solely on label prediction or data reconstruction.

1: Which does not mean that such systems exhibit actual intelligence.

[79]: Brown, Mann, Ryder, *et al.* (2020), 'Language Models are Few-Shot Learners'

[123]: Touvron, Lavril, Izacard, *et al.* (2023), *LLaMA*

[124]: Ouyang, Wu, Jiang, *et al.* (2022), *Training language models to follow instructions with human feedback*

[125]: Feldman, Foulds, and Pan (2023), *Trapping LLM Hallucinations Using Tagged Context Prompts*

[126]: Manakul, Liusie, and Gales (2023), *SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models*

[88]: Sahoo, Pham, Lu, *et al.* (2018), 'Online Deep Learning'

[127]: Hoi, Sahoo, Lu, *et al.* (2021), 'Online learning'

[131]: Hawkins, Lewis, Klukas, *et al.* (2019), 'A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex'

[132]: Lewis, Purdy, Ahmad, *et al.* (2019), 'Locations in the Neocortex'

**1000 Brains.** Jeff Hawking and his research group use the brain as the single source of inspiration and build, following their interpretation, a biologically highly plausible system that implements the brain's learning algorithm [131]. The learning algorithm is an adapted version of unsupervised Hebbian learning [40]. Their theory is based on Vernon Mountcastle's proposal that the neocortex comprises many cortical columns, all having a similar architecture and performing similar functions [59], [60]. Thus, they argue that the brain does not comprise a single learning model like current deep learning systems but many similar but independent models for each object [132]. Each model uses different inputs from different sensory system parts. These models make predictions based on the input they receive and vote to reach a consensus on the

sensory observations. This theory is also known as the 1000 brains theory, as many models work in parallel. The principle of many parallel models is closely related to ensemble approaches in deep learning [133], with the constraint, similar to random forest [134], that only a portion of the data is available to each model. Current results are difficult to interpret as the proposed model has been trained and tested on the same dataset. In its current form, the model can recognise several hundred objects and is robust to noise. However, the number of available input signals severely limits the network's capacity, and the model cannot yet scale to recognise more objects. Hidden units cannot just be added to increase the model's capacity, as increasing capacity requires additional input signals.

[133]: Yang, Lv, and Chen (2023), 'A Survey on ensemble learning under the era of deep learning'

[134]: Ho (1995), 'Random decision forests'

Strictly enforcing biological plausibility limits performance significantly, as the biologically inspired algorithm cannot fully exploit the mathematical capabilities of computational hardware. Intuitively, however, several proposed principles appear promising: *Sparse coding* encourages compact and efficient representation of complex data [135], *self-organisation* of a large number of parallel models increases robustness [133], and *predicting future cell activation* allows learning in an unsupervised manner and does not restrict the model to specific predefined tasks.

[135]: Ferrier (2014), *Toward a Universal Cortical Algorithm*

**Recursive Cortical Network.** A second type of network that has evolved from the 1000 brains theory is called the recursive cortical network (RCN) [136][2]. It is based on the insight that a human's vision system processes shape and appearance differently [137] and that a familiar object with an unexpected colour can still be easily recognised. The RCN uses separate mechanisms to process contours and appearances and uses lateral connections [138] for internal consistency. When the features cannot explain an image at a certain level $l$, the active features from the level below $l-1$ are combined to create a new feature at level $l$. Afterwards, features are pruned using a cost function considering reconstruction and compression errors. A significant advantage of RCN is that it can learn from a few examples and has one-shot and few-shot learning capabilities. However, in its current form, RCNs cannot be applied to natural images, as contour hierarchies only can be implemented if the object is clearly separated from its background. Therefore, the network, in its current form, is limited to MNIST [139] and text-based CAPTCHAs.

[136]: George, Lehrach, Kansky, *et al.* (2017), 'A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs'

2: The driving force behind it, Dileep George, worked in Jeff Hawkings' group and was instrumental in developing the fundamentals of the 1000 brains theory.

[137]: Garg, Li, Rashid, *et al.* (2019), 'Color and orientation are jointly coded and spatially organized in primate primary visual cortex'

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

[139]: LeCun, Bottou, Bengio, *et al.* (1998), 'Gradient-based learning applied to document recognition'

Similar to the 1000 brains theory, this model has drawbacks that prevent its practical application but introduces many promising concepts: The *separation of shape and appearance* seems promising, as this limits the feature space, and these concepts can be learned independently. Furthermore, in the case of multiple classes, the network does not make a single prediction but creates hypotheses that are evaluated by an outer loop. Thus, the model loops between *generating* and *evaluating* hypotheses.

**Capsule Networks.** Neuronal capsule networks (CapsNet) [140] mimic biological neural organisation and explicitly model hierarchical relationships. These networks group neurons into "capsules"; each capsule represents a property such as position, size, orientation, deformation, texture, colour or movement. Several of these capsules form more stable representations for higher-level capsules. *Dynamic routing* [141] matches

[140]: Sabour, Frosst, and Hinton (2017), 'Dynamic Routing between Capsules'

[141]: Ash, Cardwell, and Murray (1981), 'Design and Optimization of Networks With Dynamic Routing'

bottom-up activations and top-down concepts generated based on available evidence over multiple iterations. It allows the network to learn viewpoint invariant knowledge and can deal with highly overlapping objects by detecting that one object is in front of another. However, CapsNets are still an active area of research and are not yet adopted in real-world scenarios. In particular, dynamic routing makes the algorithm slow, and so far, this approach has only worked on small datasets such as MNIST [139] or CIFAR-10 [142].

[139]: LeCun, Bottou, Bengio, *et al.* (1998), 'Gradient-based learning applied to document recognition'

[142]: Krizhevsky (2009), 'Learning Multiple Layers of Features from Tiny Images'

Some concepts of CapsNets are closely related to those of the 1000 brains theory and RCNs. However, CapsNets have their origins in computer science and are oriented towards deep learning principles. In contrast, the other frameworks have originated in neuroscience and focus on biological plausibility. CapsNets *divide the features* into capsules which is helpful for generalisation. However, a severe problem is the accumulation of spatial information, which prevents the model from scaling to larger datasets. The dynamic process of *iterating* between low-level and high-level features to analyse hypotheses is promising to prevent early commitment [26] and is highly relevant for this thesis.

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

**Parsimony and Self-Consistency.** Ma *et al.* [143] suggest that appropriate data structuring can lead to the emergence of intelligence, especially if the data is structured according to the principles of parsimony and self-consistency. The two principles describe *what* and *how* should be learned. Parsimony implies that an intelligent learning system should recognise low-dimensional structures in observed high-dimensional data and organise them in the most compact and structured way[3]. The second principle of self-consistency states that an intelligent learning system minimises the internal discrepancy between observed and regenerated observations. Such internal representations can be learned through self-criticism [144]. The described framework is promising as it incorporates biological concepts into a framework that unifies and clarifies many practical and empirical findings of deep learning. Unfortunately, the authors do not present any metrics, so estimating the framework's performance is difficult. The framework's challenge is implementing the parsimony principle, i.e., defining the constraints responsible for forming the latent space and scaling it to larger datasets.

[143]: Ma, Tsao, and Shum (2022), 'On the principles of Parsimony and Self-consistency for the emergence of intelligence'

3: The goal is not to achieve the best possible compression but to obtain compact and structured representations in a computationally efficient way.

[144]: Rennie, Marcheret, Mroueh, *et al.* (2017), 'Self-Critical Sequence Training for Image Captioning'

**Actionable Representations.** Another frequently used principle inspired by biological learning is the combination of representations with actions [145], [146]. It is known that animals integrate their actions (i.e., movements) with incoming sensory signals [147]. Keurti *et al.* [148] argue that such efference copies[4] help to learn useful latent representations of input the visual system perceives. They allow an agent to perform actions by transforming objects and ensure that real-world transformations can also be applied to latent representations, i.e., mental objects and real-world objects remain consistent when similar transformations are applied. Allowing an agent to interact with the world to understand it better and improve representations seems essential not only from a neuroscientific point of view but is also in line with theories from psychology: Piaget [149] argues that perceiving an object is more about understanding how it changes and behaves than creating a mental copy of the object. The

[145]: Knoblich and Sebanz (2006), 'The Social Nature of Perception and Action'

[146]: Zhou, Krähenbühl, and Koltun (2019), 'Does computer vision matter for action?'

[147]: Keller, Bonhoeffer, and Hübener (2012), 'Sensorimotor Mismatch Signals in Primary Visual Cortex of the Behaving Mouse'

4: The internal copy of an outgoing motion-generating signal.

combination of perception and action for autonomous machine intelligence is also postulated by LeCun [150]. He proposes creating a world model whose future state can be predicted based on planned actions. He emphasises the importance of self-supervised learning in combination with energy functions [151].

[150]: LeCun (2022), 'A Path Towards Autonomous Machine Intelligence'

[151]: Hinton (2002), 'Training Products of Experts by Minimizing Contrastive Divergence'

Actionable representations not only represent the system's input data but also capture their behaviour when undergoing actions applied to them. However, increasing the representation's information content comes with the costs of providing richer input data:

- Learning visual representations of an object requires images only.
- Learning an object's appearance from different viewpoints requires sequences of images of the same object.
- Learning actionable representations requires a simulation in which actions can be applied to objects.

**Active Inference.** LeCun [150] describes that energy functions are well suited to make predictions of the world state based on actions because they can shape the latent space well due to their regulating properties. In the context of active inference [152], [153], perception presents itself as a process of minimising the free energy [151] of variation with respect to beliefs about hidden variables. This process enables planning and inference by modelling generative processes $p(s, o)$. For example, if it rained at night ($p(s)$), one can infer that the grass is wet ($p(o)$). The model tries to model the chances of different hidden situations $p(s|o)$ based on prior beliefs ($p(s)$) and the likelihood of what it already observed ($p(o|s)$). Thus, active inference minimises the variational free energy by using past experiences to learn how the world works [152]. This helps to predict what might happen in the future by minimising the probability of surprising or unexpected situations. This principle is also actively researched by other well-known research groups. For example, Bengio's research group works on GFlowNets (Generative Flow Networks) [154], [155], making it possible to disentangle the explanatory causal factors and the mechanisms that connect them. Furthermore, energy-based models can generate new samples that resemble the training data by sampling from the energy function [156]. This allows for predicting future world states.

[152]: Friston, FitzGerald, Rigoli, *et al.* (2016), 'Active inference and learning'

[153]: Parr, Pezzulo, and Friston (2022), *Active Inference*

[154]: Bengio, Jain, Korablyov, *et al.* (2021), 'Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation'

[155]: Bengio, Lahlou, Deleu, *et al.* (2022), *GFlowNet Foundations*

[156]: Du and Mordatch (2020), *Implicit Generation and Generalization in Energy-Based Models*

**Conculsion.** There exist various alternative learning approaches with interesting principles. Many of these principles are adopted in this thesis; Similar to the 1000 brains theory, local self-organisation is used to build consistency between small models, i.e. between cells representing features. Also, sparse coding is used in the thesis as it enhances interpretability and robustness [157]. The iterative process of generating and evaluating hypotheses, as applied in RCNs [136] and CapsNets [140], is considered essential to prevent early commitment [26]. In the proposed framework, such a loop is implemented between cells of the same layer: A cell can only remain active when consistent with the global pattern (evaluation loop) while simultaneously defining the global pattern (generation loop). Lastly, the proposed framework leverages an energy function to implement a memory component.

[157]: Ahmad and Hawkins (2015), *Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory*

[136]: George, Lehrach, Kansky, *et al.* (2017), 'A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs'

Besides these adopted principles, other methodologies seem essential for future work. Especially the organisation of internal representations should be investigated further. Promising approaches for organising the latent space are actionable representations that allow an agent to better understand objects' behaviour under certain transformations and the principles of parsimony and self-consistency.

## 3.2 Natural Intelligence

In this section, the theory of natural intelligence proposed by von der Malsburg *et al.* [20] is discussed, as it serves as the foundation of this thesis. The authors point out the massive gap between the amount of information needed to describe the brain's structure (1 PB) and the amount needed to generate it (few GB) [158]. Therefore, the brain must be highly structured [159]–[161] The theory's authors argue that the "Kolmogorov algorithm [162] of the brain"[5] builds the neuronal structure by selecting from a set of pre-structured patterns. This aligns with other research that argues that the brain is dominated by similar cell patterns [59], [60]. Self-organisation is the only mechanism that experiments have not yet disproved as the brains Kolmogorov algorithm [163]–[167].

This mechanism loops between activity and connectivity, with activity acting back on connectivity through synaptic plasticity until a steady state, called an attractor network, is reached. The consistency property of an attractor network implies that a network has many alternative signal pathways between pairs of neurons [168]. Thus, the brain develops as an overlay of attractor networks called net fragments [169]. Net fragments consist of sets of neurons, whereby each neuron can be part of several net fragments. In the case of visual processing, net fragments can be considered filters that detect previously seen patterns in the visual input signal. An object is represented by multiple net fragments, where each fragment responds to the surface of that object and has shared neurons and connections with other net fragments representing that object. Thus, net fragments render the topological structure of the surfaces that dominate the environment. Nested net fragments of different sizes may represent a hierarchy of features. Complex objects, such as mental constructs, can thus be seen as larger net fragments composed of mergers of pre-existing smaller net fragments.

von der Malsburg *et al.* [20] have not addressed how their theoretical concepts can be translated into a computational model. A concrete implementation is proposed by Lehmann [170]. He proposes a new layer called the laterally connected layer (LCL) that forms lateral intra-layer connections based on the Hebbian learning rule [40]. Such lateral connections are synapses between cells in the same layer [138]. However, the proposed layer has only improved performance for small networks and datasets, and its effectiveness for larger models processing more complex has yet to be demonstrated. Furthermore, lateral connections are only used at the end of the network, thereby not preventing early commitment since features are still processed hierarchically. Moreover, Lehmann does not build net fragments and thus ignores some of the aspects of the theory.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[158]: McPherson, Marra, Hillier, *et al.* (2001), 'A Physical Map of the Human Genome'

[159]: Gazzaniga (1989), 'Organization of the Human Brain'

[160]: Ackerman (1992), *Discovering the brain*

[161]: Bassett and Gazzaniga (2011), 'Understanding complexity in the human brain'

[162]: Kolmogorov (1998), 'On tables of random numbers'

5: The Kolmogorov complexity describes the number of bits required by the shortest algorithm that can generate a given structure.

[59]: Mountcastle (1978), 'An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System'

[60]: Mountcastle (1997), 'The columnar organization of the neocortex'

[168]: von der Malsburg and Bienenstock (1987), 'A Neural Network for the Retrieval of Superimposed Connection Patterns'

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[170]: Lehmann (2022), 'Leveraging Neuroscience for Deep Learning Based Object Recognition'

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

In this thesis, lateral connections are used to build net fragments to overcome the limitations of the LCL layer. Furthermore, it analyses the theory of natural intelligence by conducting experiments and contributes to making it more concrete.

### 3.2.1 Projection Fibres

An important principle of the theory of natural intelligence is the rendering of topological structures. This means that image features extracted by a filter are mapped to a corresponding reference image, whereby neighbouring features in the source image are mapped to neighbouring regions in the reference image. This mapping is inspired by the human brain that maps cell activity in the primary visual cortex [171], [172] to transformation- and position-invariant reference objects stored in the temporal cortex [173], [174]. The mapping between corresponding cells is done with a particular type of axon called projection fibre [175]. In this section, different implementations of projection fibres are presented, while biological aspects are discussed in Section 4.1.4.

[171]: Tong (2003), 'Primary visual cortex and visual awareness'

[172]: Grill-Spector and Malach (2004), 'The human visual cortex'

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

[174]: Conway (2018), 'The Organization and Operation of Inferior Temporal Cortex'

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

Projection fibres map images to reference objects and can be defined as matching subgraphs [27]–[29]: An image can be described as a graph of image features, whereby the connections of the graph represent the spatial relationships between the features in the image. However, not all pixels usually belong to the same object, and thus, a subgraph represents an object. In addition, an idealised version of this object's subgraph is stored as a prototype. By matching the subgraphs of an image and the stored subgraph prototypes, projection fibres implement subgraph matching.

[28]: Lades, Vorbruggen, Buhmann, *et al.* (1993), 'Distortion invariant object recognition in the dynamic link architecture'

Lades *et al.* [28] divide the training into two phases: During a storage phase, sparse graphs labelled with Gabor-type wavelets are formed and stored as model prototypes. In the recognition phase, the Gabor wavelets of the perceived image are matched with the previously stored graph prototypes. Therefore, a sparse graph of the image features is adaptively formed to best match a given prototype graph. The matching process is based on adjusting one-to-one links between vertices in the model and image graphs; the model's prototype graphs are moved over the input graph and locally slightly deformed. A cost function controls the graph adaptation by favouring similarities and penalising metric deformations. This matching process between image features and stored models is repeated for each stored model prototype. Finally, the prototype with the lowest cost is selected as the recognised model. This process is straightforward, but the sequential matching of models prevents the model from scaling to large datasets requiring many prototypes.

[27]: Bienenstock and von der Malsburg (1987), 'A Neural Network for Invariant Pattern Recognition'

[151]: Hinton (2002), 'Training Products of Experts by Minimizing Contrastive Divergence'

An alternative approach is described by Bienenstock *et al.* [27]. They connect all prototype graphs to the input graph randomly, and a self-organising learning process iteratively improves these connections in a coarse-to-fine manner. The matching process from prototypes to feature sub-graphs is implemented by minimising an energy function [151]. However, this approach could not scale to larger datasets or photorealistic images.

[29]: Wiskott and von der Malsburg (1996), *Face Recognition by Dynamic Link Matching*

Wiskott *et al.* [29] proposed self-organising projection fibres for face recognition. The mapping between the image array and the stored prototype

array is based on synaptic plasticity and the constraint of preservation of topography to find matches. The preservation of topography is achieved with lateral connections, which are excitatory over a short range and inhibitory over a long range. The connectivity matrix is initialised using the similarities between the features of the connected neurons. Afterwards, the connectivity matrix is adjusted by moving the prototype sub-graph across the image until a good match to the feature graph is found. However, one problem with this model is that matching takes a long time and is, therefore, impractical.

Wolfrum *et al.* [31] utilise three layers, an input layer to extract image features, an assembly layer for recurrent information integration, and a gallery layer for storing face prototypes. The input layer is organised in a rectangular grid, while the assembly and gallery layers have face graph topology. Projection fibres map the features from the input layer to the assembly layer, from where the face graph is compared with prototypes in the gallery. This architecture is biologically plausible, works fast, and obtains good results. However, the main drawback is that the cells are organised according to a face graph topology. Thus, it does not work for different objects.

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

Projection fibres map the features found in the image to stored object prototypes based on their local similarity. This creates an explicit one-to-one mapping between features and prototypes. Therefore, unlike CNNs, no position and transformation information is lost, and the mapping is better interpretable. Current implementations cannot map images to prototypes if the features do not match well. This problem is alleviated in this thesis by mapping net fragments [169] instead of features. Net fragments are based on image features but remove noise, reconstruct missing parts, and locally generalise features. Thus, net fragments transform features into suitable representations occurring similarly in reference frames, enabling a more robust mapping.

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

## 3.3 Self-Organisation

Self-organisation is the process by which systems consisting of many units spontaneously acquire their structure or function without interference from an external agent or system [42]. They organise their global behaviour through local interactions amongst themselves. The absence of a central control unit allows self-organising systems to adjust to new environmental conditions quickly. Additionally, such systems have built-in redundancy with a high degree of robustness as they consist of many simpler individual units [176]. These individual units can even fail without the overall system breaking down.

[42]: Morris, Tarassenko, and Kenward (2006), *Cognitive Systems - Information Processing Meets Brain Science*

[176]: Wagner (2013), 'Robustness in Natural Systems and Self-Organization'

Self-organisation is highly relevant in this thesis: First, the theory of natural intelligence (c.f. Section 3.2) argues the brain's Kolmogorov algorithm is self-organising and a key to natural intelligence. Second, implementing a system that prevents early commitment cannot rely on a global error correction signal that controls the entire process. Instead, taking local decisions based on the feasibility of overarching patterns requires a self-organising approach.

**Growing Patterns.** Cellular automata contain a grid of similar cells[6] with an internal state updated periodically. Update rules define the transition from a given state to a subsequent state. During an update, cells can only communicate with the neighbouring cells. Thus, self-organisation is enforced by the definition of the update rules [177], [178]. Neural cellular automata [179], [180] use neural networks to learn the update rule between cells. The input in such a neural network is the state of a given cell and its neighbours, and the output is the subsequent cell state.

Cells in NCAs can be trained with gradient descent to grow learned 2D patterns such as images [181], [182]. These images are grown through self-organisation (i.e. the pixels pick a colour based on the colour of neighbouring pixels) and are surprisingly resistant to damage. For example, large parts of the images can be removed, and the system can still rebuild the entire image. However, the aforementioned approaches can only grow the pattern they are trained on. A recent method called Variational Neural Cellular Automata [183] uses an NCA as the decoder of a variational autoencoder [184]. This probabilistic generative model can grow images based on a vector sampled from a Gaussian distribution. However, there is still a significant performance gap compared to state-of-the-art generative models. Besides growing 2D patterns, NCAs can also create 3D patterns [185], simulate robots [186], or generalise to graph structures [187].

**Classify Images.** The process of growing images from cells of an NCA can also be inverted: Randazzo *et al.* [188] propose to use NCAs to classify given structures such as images. They apply the same network to each pixel of an image. In an iterative process based on local communication with neighbouring pixels, the image fragments agree on which object they represent. Intuitively, each pixel has a hypothesis about which object it might represent. By communicating with neighbours, pixels vote for and agree on one hypothesis over time. However, this approach is limited to local interaction and only works for simple images such as MNIST.

**Learning.** Self-organisation can also be used to optimise the weights of neural networks. Hebbian learning is considered a learning rule that follows self-organising principles [189], [190] as it updates weights between two cells based on the cells' state and does not require a global teaching signal (c.f. Section 2.4.1).

Kirsch *et al.* [191] use multiple tiny recurrent neural networks (RNNs) that have the same weight parameters but different internal states[7]. By using self-organisation and Hebbian learning, they show that it is possible to learn powerful learning algorithms such as backpropagation while running the network in forward mode only. However, it works only for small-scale problems as it can quickly get stuck in local optima.

**Network Architectures.** Unsupervised learning techniques usually map high-dimensional input data to a lower-dimensional representation [70]. Such mappings can also be implemented in a self-organising manner, for example, based on self-organising maps (SOMs) [192], [193]. SOMs

6: An image or features of an image can be interpreted as a 2D grid of cells.

[177]: Wolfram (1984), 'Cellular automata as models of complexity'

[178]: Vichniac (1984), 'Simulating physics with cellular automata'

[179]: Wulff and Hertz (1992), 'Learning Cellular Automaton Dynamics with Neural Networks'

[180]: Gilpin (2019), 'Cellular automata as convolutional neural networks'

[181]: Mordvintsev, Randazzo, Niklasson, *et al.* (2020), 'Growing Neural Cellular Automata'

[182]: Mordvintsev, Randazzo, and Fouts (2022), 'Growing Isotropic Neural Cellular Automata'

[183]: Palm, González-Duque, Sudhakaran, *et al.* (2022), *Variational Neural Cellular Automata*

[184]: Kingma and Welling (2022), *Auto-Encoding Variational Bayes*

[188]: Randazzo, Mordvintsev, Niklasson, *et al.* (2020), 'Self-classifying MNIST Digits'

[189]: Najarro and Risi (2020), 'Meta-Learning through Hebbian Plasticity in Random Networks'

[190]: Pedersen and Risi (2021), 'Evolving and Merging Hebbian Learning Rules'

7: Intuitively, these tiny RNNs can be interpreted as more complex neurons.

[70]: Russell and Norvig (2021), *Artificial intelligence*

map the input data to a discrete representation space of the training samples called a map. Unlike ANNs, they use competitive learning [194] instead of error correction learning algorithms such as backpropagation of error [14], [15]. Local competitive learning ensures that samples are close in the input space are also closed in the resulting maps. Thus, the data space is self-organised by local rearrangements.

However, SOMs have two significant limitations; First, the network structure must be predefined, limiting the mapping accuracy. Second, the map's capacity is predefined through the number of nodes. Growing networks overcome these limitations and add nodes or whole layers of nodes into the network structure at the positions of the map where the error is highest [195]–[197].

**Relevance.** The proposed framework is highly related to the aforementioned self-organising principles: First, net fragments can reconstruct local patterns when occluded, resembling methodologies that can grow patterns. Second, projection fibres implement a mapping from scenes to reference frames, related to approaches that implement classification based on self-organising principles. Finally, the proposed framework relies on a self-organising Hebbian learning algorithm. In contrast, the architecture of the network is predefined and has not yet adopted a self-organising approach. Nevertheless, it is essential for future research efforts to explore the dynamic incorporation of patterns in reference frames, thereby increasing the relevance of introduced concepts such as self-organising maps (SOMs).

[194]: Grossberg and Schmajuk (1989), 'Neural dynamics of adaptive timing and temporal discrimination during associative learning'

[14]: Rosenblatt (1962), *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*

[15]: Linnainmaa (1976), 'Taylor expansion of the accumulated rounding error'

[195]: Reilly, Cooper, and Elbaum (1982), 'A neural model for category learning'

[196]: Fritzke (1994), 'Growing cell structures: A self-organizing network for unsupervised and supervised learning'

[197]: Marsland, Shapiro, and Nehmzow (2002), 'A self-organising network that grows when required'

# Biological Inspiration 4



As described in Section 1.1, deep networks suffer from early commitment. However, the *Gestalt* psychology [22]–[25], as well as the theory of natural intelligence [20] and work based on self-organising projection fibres [27]–[29], [32] considers the principle of preventing early commitment as a core mechanism for the effectiveness of the human visual system.

The human brain can prevent early commitment [26] while still being an excellent image-processing system [22]–[25]. Consequently, the neuroscientific literature is studied, and promising findings to implement a vision framework preventing early commitment are identified. Thus, this chapter can be considered a survey of existing neuroscientific findings. However, compared to the related work section, it contains more interpretations and introduces a unified vocabulary compatible with both fields, neuroscience and deep learning.

In this chapter, important neuroscientific findings are presented in Section 4.1, describing lateral connections (Section 4.1.2), net fragments (Section 4.1.3), the local learning principle (Section 4.1.6), and projection fibres (Section 4.1.4). Afterwards, in Section 4.2, it is described how these biological findings could improve current systems.

## 4.1 Neuroscientific Findings

### 4.1.1 The Brain's Visual System

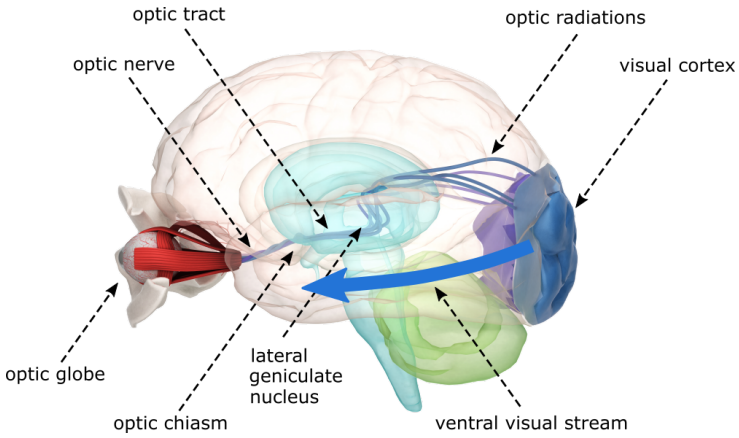


**Figure 4.1:** Visualisation of the human's visual system. The image is from Fasoli [198].

The visual system of humans is illustrated in Figure 4.1. The eyes are sensors that capture light waves and translate them into electrical pulses. These electrical signals travel through the human brain to the primary visual cortex [171], [172], located at the back of the head. Cells within the visual cortex fire spikes when specific visual stimuli appear within their receptive fields [172]. Thus, these cells can be considered filters that are excited if a known pattern is detected in the input data.

[171]: Tong (2003), 'Primary visual cortex and visual awareness'

[172]: Grill-Spector and Malach (2004), 'The human visual cortex'

The visual cortex detects patterns in visual data but does not draw conclusions from it. Instead, it forwards it as an information stream to other brain areas. In this thesis, especially the ventral visual stream [199] is of importance, which forwards the detected patterns to the temporal cortex [173], [174], a brain region located behind the ears. According to the two-stream hypothesis [199], the temporal cortex is responsible for object identification and recognition. Thus, the visual cortex detects patterns that are compared with object prototypes stored in the temporal cortex. A second stream, the dorsal stream [199], forwards the same data to the parietal cortex, a region on top of the head that predicts the object's spatial location relative to the viewer [200]. However, this stream is of less importance in this thesis.

[199]: Goodale and Milner (1992), 'Separate visual pathways for perception and action'

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

[174]: Conway (2018), 'The Organization and Operation of Inferior Temporal Cortex'

[200]: Colby and Goldberg (1999), 'Space and attention in parietal cortex'

The brain leverages net fragments [169], lateral connections [138], [201], [202], and projection fibres [203] to process visual data. These fundamental elements of the brain serve as inspiration for the proposed framework and are introduced in the following.

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

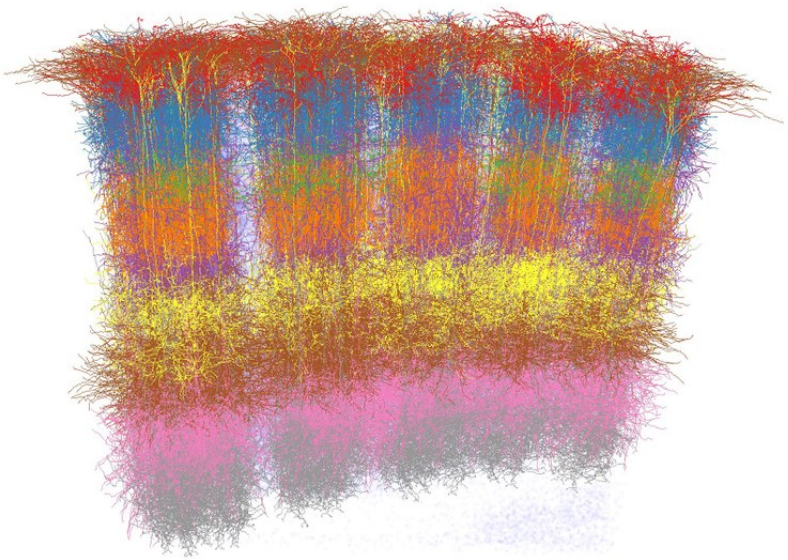

**Figure 4.2:** 3D reconstruction of five neighboring cortical columns of a rat. The image is from Oberlaender *et al.* [204].

### 4.1.2 Lateral Connections

The cerebral cortex forms the outer hull of the brain [205] and encompasses several regions, including the previously mentioned visual and temporal cortex, as well as the ventral visual stream. The cerebral cortex consists of many cylindrical arrangements of neurons called cortical columns [60]. A 3D reconstruction of five cortical columns is shown in Figure 4.2, with different layers visualised by different colours.

[205]: Narr, Woods, Thompson, *et al.* (2007), 'Relationships between IQ and Regional Cortical Gray Matter Thickness in Healthy Adults'

[60]: Mountcastle (1997), 'The columnar organization of the neocortex'

Information in the cerebral cortex is propagated forward from one layer to the next and has inspired layer-wise processing in deep learning architectures [36]. However, a closer look at the human brain reveals that there are also connections between neurons within the same layer that process information locally [138]. These intra-layer connections are called *lateral connections* [201], [202] and are visualised in a simplified manner in Figure 4.3 for a single neuron. The red neuron not only establishes

[36]: Prince (2023), *Understanding Deep Learning*

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

connections to neurons in the preceding and subsequent layers (marked in orange and green) but also lateral connections to neurons within its own layer (marked in red).

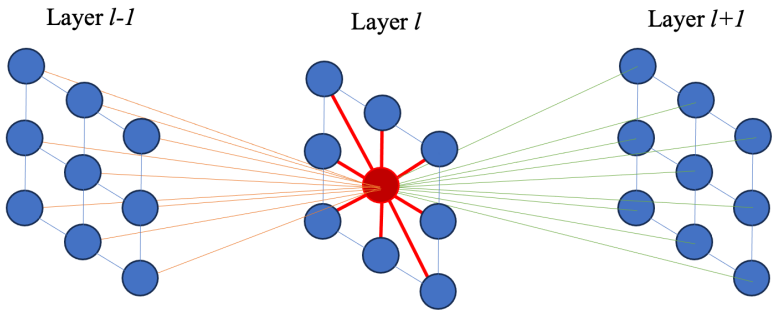


**Figure 4.3:** Visualisation of the connections of a single cell. The cell is connected to the previous layer (orange), the subsequent layer (green), and to neurons within the same layer (lateral connections, red).

According to von der Malsburg *et al.* [20], these lateral connections are used for *lateral support*. "Lateral support" means that neurons from the same layer support each other's activity: Neurons from the preceding layer can activate the red cell depicted in Figure 4.3 through the orange connections. However, inhibitory signals can suppress the activity [41] of the red neuron before it can fire a spike to the subsequent layer via the green connections. The neuron can only transmit a spike to the subsequent layers if it "survives" an inhibition phase [102], which is only possible if it receives sufficient lateral support [202] from laterally connected neurons. For instance, suppose the preceding layer activates several neurons within the same layer as the red neuron, and these activated neurons exhibit lateral connections. In that case, they can send spikes to each other, thereby providing mutual support. This allows them to maintain their action potential and remain active during the inhibition phase.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[41]: Coombs, Eccles, and Fatt (1955), 'The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential'

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

### 4.1.3 Net Fragments

Lateral connections grow between cells that are often active together [40]. Since cells are often simultaneously active when representing the same pattern, lateral support is increased between groups of neurons representing frequently occurring patterns [202]. Such groups are called *net fragments* [169]. All neurons within a net fragment support each other to remain active during an inhibition phase [102]. Thus, a layer with multiple net fragments can be considered a filter: While the previous layer might activate numerous cells, only the cells with sufficient lateral support remain active. Therefore, only learned patterns survive and send a spike to the next layer [169].

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

#### Local Neighbourhood

Net fragments represent patterns that can be distinguished between local patterns, which are spatially limited to a local area, and global patterns, which extend over larger regions of the image and might encompass the entire image. The number of possible patterns increases exponentially with the considered pattern size. Thus, local patterns occur more frequently. To capture frequently occurring patterns, the visual cortex limits the range of lateral connections to a *local neighbourhood*

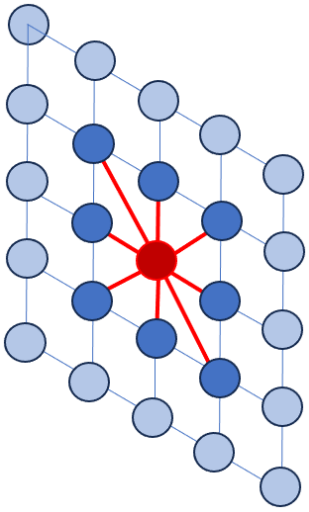

**Figure 4.4:** The lateral connections limited to a local neighbourhood.

[201], [202]. Capturing global patterns would require exponentially more cells, and therefore, limiting the range of lateral connections to local neighbourhoods is crucial. In Figure 4.4, such limited lateral connections are illustrated: The lateral connections from the red cell do not encompass the entire image but are only connected to cells in close proximity.

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

The size of the local neighbourhood in the human brain varies [206]. The primary visual cortex captures the input signal with a high variance and therefore has a strongly limited local neighbourhood size [169]. The temporal cortex contains transformation-independent object representations and can afford larger neighbourhood sizes as fewer distinct global patterns exist [169].

[206]: Pessoa (2014), 'Understanding brain networks and brain organization'

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

### Hierarchy of Net Fragments

A single cell is supported by its neighbouring cells, which, in turn, are supported by their neighbouring cells. Therefore, the support reaches much further than only the local neighbourhood [169], [20]. As the processing progresses, increasing inhibition causes cells without sufficient support to be turned off. Turning off one cell can trigger a chain reaction of further turn-offs. Therefore, lateral support occurs not only between individual cells but also between many overlapping net fragments [20].

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

Thus, the network consists of an overlay of net fragments, which can be interpreted as a larger net fragment, i.e. a multitude of cells supporting each other [20]. The size of a net fragment cannot be defined; the smallest possible net fragment is a single cell with its local neighbourhood, while the largest net fragment can span all active cells that are laterally connected. Furthermore, a single layer represents local and global features at the same time, whereby local features are stored in smaller net fragments and global features in larger net fragments [20]. Thus, net fragments form a feature hierarchy within a layer [20].

#### Alternative Cells and Pathways

At a given spatial location, different patterns can occur. However, the capacity of a net fragment is limited to represent a single pattern [20]. For example, cell $A$ may often fire with cells $B$ and $C$ in close proximity, exhibiting a high mutual lateral support with these cells. However, cells $B$ and $C$ might not fire together. Consequently, cell $A$ is involved in two distinct and mutually exclusive net fragments, once with cell $B$ and once with cell $C$. To facilitate such coexistence between net fragments, *alternative cells* are required [169]. A copy of cell $A$ must exist that behaves similarly but has different synaptic connections, i.e. exhibits *alternative pathways*. The precise biological mechanism of how this is implemented is unclear; One hypothesis is that within a group of cells that initially have similar afferent connections, cells undergo divergent connectivity changes during training [43], resulting in cells specialising in different patterns. Thus, multiple similar feature cells could exist at the same spatial location and enable the formation of alternative and mutually exclusive net fragments.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[43]: Grossberg (1987), 'Competitive Learning'

### 4.1.4 Projection Fibres

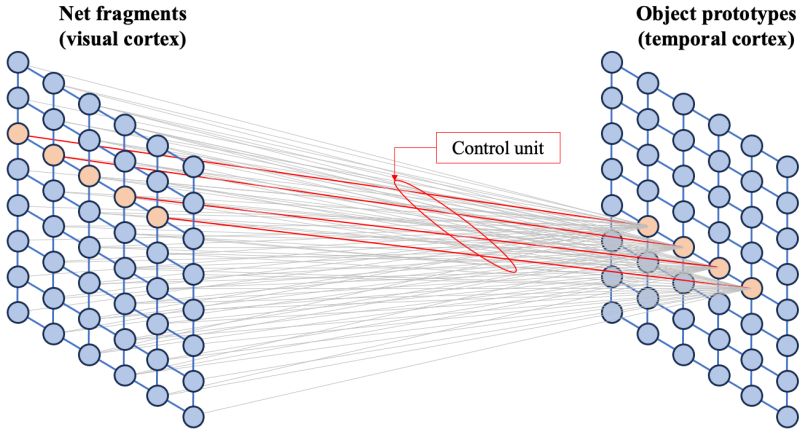


**Figure 4.5:** Net fragments (on the left) are projected to an object (on the right). Many projection fibres (grey) run between the net fragments and objects, but only a few belonging to the same maplet (red) have been turned on.

As described at the beginning of this chapter, the visual cortex [171] extracts patterns from visual information [172]. This process is implemented with the aforementioned building blocks, such as lateral connections [138], [201], [202] and net fragments [169]. In theory, this is sufficient to implement the principles from the Gestalt psychology [22]–[25] and allows building feature hierarchies without early commitment [26]. However, it is not sufficient for efficient visual object detection.

In the human brain, object detection occurs in the temporal cortex [173], [174], a region spatially distant from the visual cortex. The brain's solution to transmit information over such long distances is utilising *projection fibres* (a type of axons) [175], [203]. Projection fibres are links between neurons in the visual cortex and object prototypes (so-called reference frames) in the temporal cortex [199].

Object prototypes in the temporal cortex are net fragments similar to those in the visual cortex. However, the visual cortex captures an overlay of fragments that describe a captured visual scene [20], whereby it is unclear which fragments represent distinct objects and how they are related. On

[171]: Tong (2003), 'Primary visual cortex and visual awareness'

[172]: Grill-Spector and Malach (2004), 'The human visual cortex'

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

the other hand, the fragments in the temporal cortex depict one object, whereby these objects are position and transformation-invariant.

Projection fibres map neurons within the captured scene (in the primary visual cortex) to object prototypes (in the temporal cortex) with one-to-one connections [207], where pairs of neurons connected in the visual cortex project to pairs of neurons connected in the temporal cortex topologically. The projection fibres have some flexibility, allowing for local distortions and enabling transformation- and position-invariant mapping [29], [31]. This mapping allows the recognition of one or more objects within a scene and their relationships to each other, facilitating object recognition and scene interpretation.

[207]: Anderson and Essen (1987), 'Shifter circuits'

[29]: Wiskott and von der Malsburg (1996), *Face Recognition by Dynamic Link Matching*

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

#### Maplets and Control Units

The ventral stream [199] connects feature cells in the visual cortex to cells representing different object prototypes in the temporal cortex. Consequently, there is a multitude of projection fibres [175], [203], but only a fraction are active at any given time [207], [208]. Typically, the same set of projection fibres is activated by similar patterns. Such sets of projection fibres that are frequently activated simultaneously are grouped into *maplets* [209]. *Control units* decide which maplets are activated and thus initiate the mapping between the visual and temporal cortex [209]. A control unit in the human brain is a unipolar neuron, a kind of neuron with extensions (so-called processes) that end in synapses and can conduct signals in both directions - from the synapse to the neuron and from the neuron to the synapse [210]. Control units trigger a mapping when a net fragment in the visual cortex matches another fragment in the temporal cortex, i.e. when these two fragments have a high correlation [209]. However, they only remain active if numerous other projection fibres confirm the decision and map their respective fragments in the visual cortex to the same object prototype in a topological manner [169]. By doing so, the human brain generates numerous hypotheses about observations, but inhibitory signals quickly deactivate most projections, leaving only the plausible ones active.

[199]: Goodale and Milner (1992), 'Separate visual pathways for perception and action'

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

[207]: Anderson and Essen (1987), 'Shifter circuits'

[208]: Olshausen, Anderson, and Van Essen (1993), 'A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information.'

[209]: Zhu and von der Malsburg (2004), 'Maplets for correspondence-based object recognition'

[210]: (2019), *The Oxford handbook of invertebrate neurobiology*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

Such a projection between net fragments and an object prototype of a line is visualised in Figure 4.5. A vast amount of projection fibres (grey) run between these two areas, but only the most suitable ones are activated by the control unit of a maplet (red). Please note that the line on the left is translated and stretched. Nevertheless, projection fibres still map such a transformed object to an idealised prototype. In Figure 4.5, a direct mapping is shown for better clarity. In contrast, the mapping in the brain is done over several hierarchical levels, saving many projection fibres [207].

Projection fibres thus provide generalisation, i.e. different, transformed versions of an object are recognised and mapped to a reference frame. This explains the ability of humans to see a new object once and immediately recognise it in a transformed version.

### 4.1.5 Dynamic Mapping

Neuroscientific findings suggest that net fragments [20], [169] are present

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

in the visual and temporal cortex and that projection fibres map corresponding fragments topologically to implement object recognition [175], [201], [202]. Further experiments demonstrate that the recognition time for humans depends on the size [211] and orientation [212], [213] of objects. Thus, it takes time to align the external world with internal representations. This suggests that the brain implements an active dynamic process for correspondence finding rather than having a single forward pass. Furthermore, physiological evidence exists that connections in the visual system are not static and suggest that receptive fields in the visual system change from one instance to the next to route the current neuronal activations to representations [214], [215]. These findings suggest that projection fibres self-organise within a short time interval to initiate the corresponding mapping.

[211]: Bundesen and Larsen (1975), 'Visual transformation of size.'

[212]: Jolicoeur (1985), 'The time to name disoriented natural objects'

[213]: Lawson and Jolicoeur (1999), 'The effect of prior experience on recognition thresholds for plane-disoriented pictures of familiar objects'

[214]: Kusunoki and Goldberg (2003), 'The Time Course of Perisaccadic Receptive Field Shifts in the Lateral Intraparietal Area of the Monkey'

[215]: Womelsdorf, Anton-Erxleben, Pieper, *et al.* (2006), 'Dynamic shifts of visual receptive fields in cortical area MT by spatial attention'

### 4.1.6 Local Learning Principle

In the brain, consistency is evaluated at the level of synapses between connected cell pairs [40]. Each synapse is established if the firing of its source and target neurons is consistent. This process is crucial for establishing net structures, where each neuron within a net fragment can predict the firing of other neurons with a high probability [216].

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

[216]: Widrow, Kim, Park, *et al.* (2019), 'Nature's Learning Rule'

Such consistency is built between cells within the same layer connected by lateral connections [202] and between cells in different brain regions connected by projection fibres [175], [201]. Building consistency means that the cells reach a consensus on what they represent [20]: Features trigger cells belonging to specific net fragments and only remain active if other laterally connected cells contribute to the same fragment and provide mutual support [102]. Cells that do not receive sufficient support are deactivated by inhibition [41]. Thus, cells agree on which net fragments (which features) are present in the input in a self-organising manner. Consistency is achieved similarly between cells connected through projection fibres: Initially, many hypotheses (mappings) are activated, but after increasing inhibition, only the mapping receiving the most cell support remains active [20], [102]. Thus, every active cell votes for net fragments and either remains active or is deactivated by inhibition until the entire network becomes consistent by agreeing on what features and objects are observed in the input [20], [169].

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

[41]: Coombs, Eccles, and Fatt (1955), 'The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential'

This local learning is a key difference between natural (animal or human) learning and the frequently used backpropagation of error [14], [15]. With backpropagation, consistency is optimised at a single point [39], [44], specifically between the system's output and a teacher signal. All synapses, including those that are distantly connected (the "deep" connections), are guided by the consistency of this single point. Therefore, the human brain is dominated by local learning and self-organisation, while deep networks typically use a global learning rule that guides the learning process of the entire network. Since deep networks outperform humans on specific tasks [6], [75], I speculate that optimising consistency at a specific point, as done by deep learning, works exceptionally well for highly specialised tasks while optimising consistency between each neuron, as done in the human brain, improves knowledge transferability, object understanding and data efficiency (c.f. Section 8.1).

[75]: Buetti-Dinh, Galli, Bellenberg, *et al.* (2019), 'Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition'

## 4.2 Long-Term Vision

The aforementioned neuroscientific findings serve as the foundation for building a novel image-processing framework. The core behind this framework is based on two stages (representing the primary visual [171], [172] and temporal cortex [173], [174]) and projection fibres connecting them [175], [203]. The first stage builds net fragments [169] using lateral connections [138], [201], [202], reassembling a visual scene captured with a sensory system (the eyes). The second stage contains reference frames representing specific objects. These objects are centred and transformation-invariant and can thus afford longer-reaching lateral connections. Multiple 2D reference frames must exist for each object to represent an object from various viewpoints. Projection fibres connect these two stages [175], [203], mapping objects within a visual scene to object prototypes. This mapping serves as a scene interpretation layer by describing which objects are located where in the observed scene. In Chapter 5, a framework is proposed that implements these two stages linked with projection fibres. It is considered a computational implementation of the fundamentals of the biological visual system. However, in the long term, this framework can be further extended and is not limited to these two stages, enabling highly efficient scene processing. These extensions are discussed in the following to provide a long-term vision for the framework.

### 4.2.1 Object Classification

A classification layer maps an object to a specific instance [217], for example, a person, to a person's name. Projection fibres do not provide such a classification as these fibres map the pattern to a more generalised view, i.e. transforming a person to a reference person. Instead, a subsequent memory stage (related to the brain's inferior cortex [173]) is needed to map a reference object to an actual label. This memory stage contains multiple instances for each object and provides distinction between objects. The memory stage implements an $n$-to-$n$ mapping to the reference frame: Each reference frame has multiple instances (e.g. multiple persons exist), and each instance can belong to multiple reference frames (e.g. a face and a body could be different reference frames but belong to the same instance).

The human brain has single cells representing specific instances of objects [218]. Therefore, memory is assumed to consist of one or a few cells representing a "label", while reference frames consist of many cells describing an object's appearance [20] [169]. This allows storing and distinguishing many object instances while not requiring a vast number of cells.

### 4.2.2 Scene Interpretation

Projection fibres map all objects within an observed scene to prototypes. Thus, the projection fibres implement object segmentation and identification. In addition, the position of each object is known, as well as the relative differences in position between the objects. Such a mapping

[171]: Tong (2003), 'Primary visual cortex and visual awareness'

[172]: Grill-Spector and Malach (2004), 'The human visual cortex'

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

[174]: Conway (2018), 'The Organization and Operation of Inferior Temporal Cortex'

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[217]: Schmarje, Santarossa, Schroder, *et al.* (2021), 'A Survey on Semi-, Self- and Unsupervised Learning for Image Classification'

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

[218]: Gross (2002), 'Genealogy of the "Grandmother Cell"'

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

provides a mental description of a perceived scene and answers the question of which object is where. However, having a description is not sufficient to interpret a scene. For scene interpretation, all objects must be put into context.

As described in Section 4.1.6, consistency is built between the cells within the two stages to build net fragments and between the projection fibres connecting net fragments [20], [201], [202]. In the long term, more components must be added, allowing to build consistency between memories and reference frames as well. For instance, projection fibres could map one object in the scene to a person while mapping another object in close spatial proximity of this person's foot to a ball. By building consistency between these two objects, one could conclude that this person is most likely playing football (soccer). Synaptic connections between these references could be formed if such scenes are observed several times, allowing for scene interpretation. Thus, observing people touching a ball with their feet corresponds to a pattern the system frequently observes, characterised by the simultaneous activity of the corresponding net fragments and their relative position. By integrating this activity with relative positions and applying Hebbian updates, the network learns this pattern corresponding to football. After the pattern has been learned, cells can vote for it by providing lateral support to this scene interpretation, and the consistency property describes if an observed scene corresponds to playing football or not. Furthermore, memories [97], [173] could be integrated to confirm or reject this hypothesis. If, for example, the person is identified as a soccer player such as Ronaldo, it would further strengthen this hypothesis.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[97]: Liu, Ramirez, Pang, *et al.* (2012), 'Optogenetic stimulation of a hippocampal engram activates fear memory recall'

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

A system building consistency between object prototypes, relative object positions, and memories could learn interpretations of visual scenes and might have the potential to "understand" how objects are related to each other. A pivotal difference to deep networks is that the model can build such consistency on its own, while deep learning systems require a teaching signal to learn consistency. I speculate that having a system that can build consistency of completely unseen scenes without teaching signals is an important step towards emergence.

### 4.2.3 Avoiding Early Commitment

As described in the Section 1.1, deep learning models are prone to early commitment [26]. The brain's solution to prevent early commitment is building net fragments. A single layer with lateral connections that forms net fragments can represent local and global features at the same time[1]. Each set of active and laterally connected neurons can be considered a net fragment. Net fragments with few cells depict local features, while fragments containing many features represent global features.

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

1: This way of thinking about features can be hard to comprehend, especially for computer scientists familiar with deep learning.

A system that avoids the fallacy of early commitment should solve the conundrum that local decisions are taken based on plausibility in the light of high-level patterns, while high-level patterns can only be defined based on low-level features. The most fundamental local decision is whether a single cell should be active. Since a single cell requires support from all laterally connected cells, a high-level pattern (i.e. the activity of all directly or indirectly connected cells) is used to make this decision.

Therefore, local decisions are taken based on high-level patterns. The high-level pattern used to make this decision is defined by the sum of individual cells (i.e. low-level features). Hence, the human brain solves the conundrum within every single layer.

**Hierarchical Features**

Neuroscientific findings indicate that the human brain does not rely on a sequence of layers to build feature hierarchies as typically done in deep networks [36]. In the biological context, layers have other connotations, and evidence suggests that layers in cortical columns deal, for example, with depth rotations [219]. In the proposed framework, feature hierarchies are implicitly stored in net fragments whereby a larger fragment represents a global feature, and smaller fragments represent local features. Consequently, sequences of layers are not necessary to represent feature hierarchies.

[36]: Prince (2023), *Understanding Deep Learning*

[219]: Iamshchinina, Kaiser, Yakupov, *et al.* (2021), 'Perceived and mentally rotated contents are differentially represented in cortical depth of V1'

However, multiple stages are required to build abstractions of input signals, i.e. to generalise features. In the human brain, several regions interact to generate abstractions [58]. In fact, the visual cortex builds net fragments based on signals from the eyes, projection fibres map the object to prototypes, and other fibres map the prototypes to memories and interpret them. Thus, various stages are involved in processing images. Furthermore, processing occurs quickly until all stages reach a consensus [32]. Thus, no layer-wise processing is required as in deep learning, but multiple stages interact with each other to build feature abstractions. As described in Section 4.1.5, these interactions are a dynamic, iterative process rather than a single forward pass.

[58]: Felleman and Van Essen (1991), 'Distributed Hierarchical Processing in the Primate Cerebral Cortex'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

# Biologically Plausible Vision Framework

# 5

The introduction in Section 1.1 describes that preventing early commitment [26] is considered a fundamental property of the human visual system that is absent in current deep learning frameworks. The previous chapter introduced neuroscientific findings that could explain why the human brain has this ability and that are considered the principles implementing the findings of the Gestalt psychology [22]–[25].

In the following, a novel framework incorporating these identified principles is proposed. The focus is on translating the two stages described in Section 4.1 into a computational framework. Incorporating further stages required for object classification and scene interpretation as described in the long-term vision in Section 4.2 remains an open task for future research. While the inspiration and principles are based on previous research, a novel computational framework is described in this chapter.



## 5.1 3-Staged Model

Some of the neuroscientific principles described in Section 4.1 have been explored in the theory of self-organising projection fibres [29], [30], [32] (c.f. Section 3.2.1). However, these approaches do not yet scale to natural images except for human faces [31]. This is because most work in this area has neglected to model the learning and the dynamics of rich sets of net fragments in the visual cortex, which are fundamental according to the theory of natural intelligence [20].

This chapter describes a novel system based on binary neurons with the potential to scale to natural images since it extends the projection fibres with net fragments. The proposed system comprises three main components: A first stage *S0* that extracts features from the image, a stage *S1* that builds an overlay of net fragments, and a stage *S2* that uses projection fibres to map them to object prototypes. I call the stage *S0* the sensory system, *S1* the feature building stage, and *S2* the prototype stage. In the context of biology, the sensory stage *S0* could stand for the eyes translating visual information into neuronal activity [172], *S1* could stand for the primary visual cortex [171], and *S2* for the ventral stream [199] and an area in the temporal cortex [173]. In the following, an overview of these three building blocks is given from a computational perspective, and the advantages of the proposed framework are described. Afterwards, more implementation details are provided, making the framework more concrete.

### 5.1.1 Building Blocks

In Figure 5.1, an overview of the building blocks of the proposed framework is provided. After the sensory stage extracted features and some

[29]: Wiskott and von der Malsburg (1996), *Face Recognition by Dynamic Link Matching*

[30]: Wiskott, Fellous, Kuiger, *et al.* (1997), 'Face recognition by elastic bunch graph matching'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[172]: Grill-Spector and Malach (2004), 'The human visual cortex'

[171]: Tong (2003), 'Primary visual cortex and visual awareness'

[199]: Goodale and Milner (1992), 'Separate visual pathways for perception and action'

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

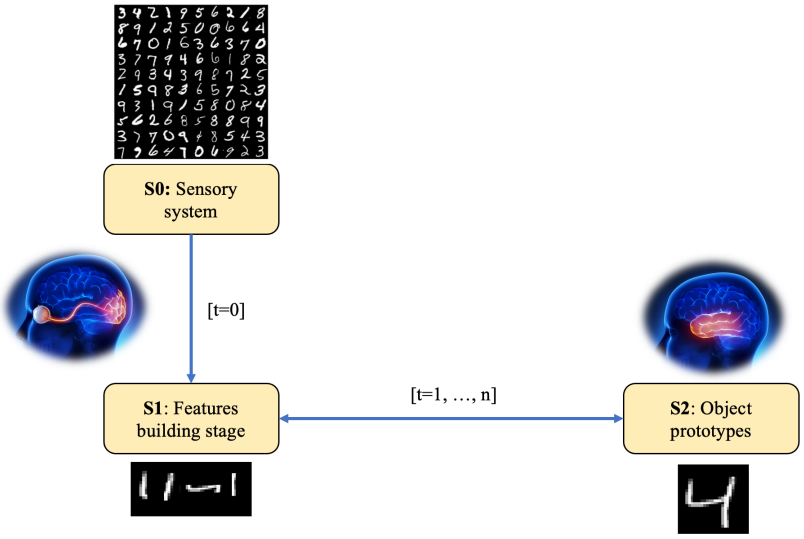


**Figure 5.1:** Overview of the framework. *S0* extracts features from the image at timestep $t = 0$, *S1* builds net fragments, and *S2* maps them to object prototypes using projection fibres. The network refines the features over multiple timesteps in which inhibition is increased, and cells without sufficient support are turned off.

initial net fragments have been built in *S1*, an inhibition phase [41], [102] lasting multiple timesteps turns off cells that do not receive sufficient lateral support. Furthermore, the project fibres [175] between *S1* and *S2* run in both directions, not only mapping net fragments to reference frames but also providing feedback to *S1* in the form of additional support for well-known objects. All these building blocks utilise a binary neuron, which I call the Bernoulli neuron, as its output state is sampled from a Bernoulli distribution. Utilising Bernoulli neurons lead to sparse and distributed binary network activities, which possess properties that enhance robustness to noise within a network [157].

**Sensors System *S0*.** A typical input to the sensory stage is an image having one (grey-scale) or three (RGB) colour channels. Therefore, such an image can be interpreted as having one or three features at every spatial location. The sensory system extracts multiple features by considering a spatial neighbourhood, thereby increasing the number of features at each location. For example, a sensory system can be a set of hand-crafted or Gabor filters [220], [221] applied at all image positions or the first layer of a pre-trained CNN [35].

**Feature Building Stage *S1*.** The feature building stage is a single layer with lateral connections [138] that builds an overlay of net fragments. Input from the sensory system activates some feature neurons in *S1*. However, their continued firing relies on receiving support from a sufficient number of other activated neurons that are laterally connected to them [169]. Initially activated neurons that do not receive enough lateral support deactivate after a short period due to inhibition [102].

The lateral connections are learned through self-organisation [40]. Patterns occurring repeatedly in the training data will constantly activate the same cells simultaneously. Using Hebbian learning [40] strengthens the connection between these cells, and the pattern is “stored” in a net fragment [169]. The inhibition strength increases during training to account for the increasing numbers of lateral connections so that cells with fewer active connections can be suppressed in favour of those

[41]: Coombs, Eccles, and Fatt (1955), ‘The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential’

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), ‘Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks’

[175]: Greig, Woodworth, Galazo, *et al.* (2013), ‘Molecular logic of neocortical projection neuron specification, development and diversity’

[157]: Ahmad and Hawkins (2015), *Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory*

[220]: Gabor (1946), ‘Theory of communication’

[221]: Granlund (1978), ‘In search of a general picture processing operator’

[35]: LeCun, Boser, Denker, *et al.* (1989), ‘Backpropagation Applied to Handwritten Zip Code Recognition’

[138]: Gilbert, Hirsch, and Wiesel (1990), ‘Lateral Interactions in Visual Cortex’

[169]: von der Malsburg (2018), ‘Concerning the Neuronal Code’

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

with more connections, allowing the latter to acquire even more lateral connections.

Consequently, many neurons might be activated by the sensory stage, but only the ones supporting each other remain active. Therefore, it is essential to view the process from the perspective that active neurons are integral parts of consistent global net fragments, and only the support from neurons within the same net fragments enables them to persist.

To facilitate various patterns and the coherent networks underlying them, neurons require many excitatory connections. To prevent cross-talk between net fragments (where a neuron receives support from a network it does not belong to), a sustained-firing condition is required in the form of a minimum number of connections that must be activated.

**Object Prototype Stage *S2*.** Stage *S2* has the same structure as *S1*. However, it has a smaller coverage area, focusing on object-centred representations rather than encompassing the entire visual field. It allows lateral connections with a greater range [202], [206], essential to represent larger-scale structures like objects. Thus, this stage contains isolated net fragments that can be considered object prototypes invariant to translation, scale, and orientation.

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[206]: Pessoa (2014), 'Understanding brain networks and brain organization'

In the object recognition process, corresponding net fragments in *S1* are mapped to object prototypes in *S2* through active projection fibres. Here, "corresponding" refers to neurons relating to the same point on the object's surface. The projection fibres between *S1* and *S2* are grouped in maplets, whereby a maplet comprises a collection of fibres that establish one-to-one connections between all neurons in a small patch of *S1* and all neurons in a small patch of *S2* in a topological manner. These topological connections link neighbouring neurons in *S1* to neighbouring neurons in *S2*. Both *S1* and *S2* are divided into overlapping patches, and for each pair of patches - one in *S1* and one in *S2* - a corresponding maplet exists.

Control units initiate activation of a maplet when they observe a high pattern correlation between the signals carried by its fibres from *S1* and the signals on their target neurons in *S2*. They inhibit competing control units: Many projection fibres can initially be activated, but only those activated S2 neurons with sufficient lateral support can remain active. Consequently, the activated projection achieves a homeomorphism, where neurons of a particular feature type in *S1* are connected to neurons of the same type in *S2* and neurons connected in *S1* activate neurons connected in *S2*.

### 5.1.2 Advantages

The proposed framework not only prevents early commitment [26] but is expected to have various additional advantages compared to the typical deep learning framework. These are described in the following.

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

**Ambiguity.** The proposed framework permits the persistence of multiple net fragments, enabling the system to handle ambiguity effectively. For example, when presented with a face comprising distinct objects (c.f. Figure 1.1), both the net fragments responsible for abstract faces and those associated with individual objects become concurrently active. Consequently, the model can simultaneously attend to these net fragments, utilising attention in its original sense rather than the conventional deep neural network (DNN) interpretation [222]. I speculate that this represents a fundamental distinction from neural networks that are typically compelled to represent the entire scene within a single high-dimensional dense vector.

[222]: Niu, Zhong, and Yu (2021), 'A review on the attention mechanism of deep learning'

**Robustness.** An input of a neural network is usually represented with a floating-point vector which is sequentially processed by mathematical functions (e.g. with neural layers). Artificial networks, in particular, are not robust to noise and are susceptible to adversarial attacks [11]. A binary vector, on the other hand, has different mathematical properties and is more robust against noise and adversarial attacks, especially if they are sparse and distributed[1] [157]. Subsampled or noisy vectors are still semantically similar and are close to the original vectors when compared, for example, by counting the overlap of bits between two vectors. Furthermore, the model builds consistency at every point in the network, making it more robust than when consistency is built at a single point [176].

[11]: Akhtar and Mian (2018), 'Threat of Adversarial Attacks on Deep Learning in Computer Vision'

1: Only a small portion of the bits are "on", and representations differ by multiple binary bits.

[157]: Ahmad and Hawkins (2015), *Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory*

[176]: Wagner (2013), 'Robustness in Natural Systems and Self-Organization'

**Object-Independent Transformations.** The same projection fibres are applied to all object prototypes, allowing the model to learn object-independent transformations. For example, an object might be slightly stretched, rotated, or deformed compared to the stored prototypes. The projection fibres learn to ignore slight deformations independent of the object type. This allows the architecture to learn transformation invariance and to transfer this capability to new objects that have not been transformed in the training data. Furthermore, object-independent projection fibres allow adding objects dynamically to static reference frames, implementing lifelong learning [89] without the risk of catastrophic forgetting [12], [86].

[89]: Parisi, Kemker, Part, *et al.* (2019), 'Continual lifelong learning with neural networks'

[12]: Kirkpatrick, Pascanu, Rabinowitz, *et al.* (2017), 'Overcoming catastrophic forgetting in neural networks'

[86]: Liu, Yang, and Wang (2021), 'Overcoming Catastrophic Forgetting in Graph Neural Networks'

## 5.2 Bernoulli Neuron

The previous section provided an overview of the proposed framework's inspiration, building blocks, and advantages. In the following, the building blocks are described in more detail, starting with the Bernoulli neuron. In traditional neural networks, neurons exhibit dense activity, meaning that even when applying the rectified linear unit (ReLU) activation function (c.f. equation (2.6)), many neurons remain active (above zero) [223]. However, biological neurons have different characteristics than artificial neural networks [224].

[223]: Rhu, O'Connor, Chatterjee, *et al.* (2018), 'Compressing DMA Engine'

[224]: Kandel (2013), *Principles of neural science*

Preliminary experiments have indicated that using non-binary artificial neurons is not well suited for learning net fragments. For instance, dealing

with weak or strong positive activations poses challenges when employing Hebbian updates. This suggests that implementing net fragments [169], similar to the human brain, requires using different principles than the ones used in classical neural networks. Inspired by neuroscientific findings, a probabilistic neuron that samples its activation from a Bernoulli distribution is introduced. Such a neuron is a binary neuron that does not fire when a certain threshold is reached but uses its internal state as a firing probability. A Bernoulli neuron $a_i$ in the context of net fragments is modelled as a probability density function of the form:

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

$$p = P(a_i = \text{active} | \text{activity of neighborhood}, \text{environment}) \tag{5.1}$$

Thus, the probability of a neuron being active depends on the activity pattern of the neurons in its local neighbourhood and factors of the environment (e.g., inhibition or presence of neurotransmitters). The output is sampled form a Bernoulli distribution, i.e., $B(p) = P(X = 1) = p = 1 - P(X = 0)$. Having a neuron whose firing probability $p$ is governed by the neighbourhood activity and the environment allows the implementation of the behaviour of net fragments [20], [169]: After receiving an input, the neurons get excited and fire with a higher probability. However, their firing probability decreases quickly if not supported by neighbouring neurons. Thus, uncertainty and potential net fragments govern timestep 0, while the network reaches an attractor state shortly after.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

### 5.2.1 Properties

The proposed neuron implements a stochastic process that allows it to fire even when the probability of it firing is low or, conversely, not to fire when the probability is high. I call this property "flipping". Flipping neurons lead to noise in the network's activations. However, this noise can be considered a normalisation mechanism within the network, similar to dropout layers [225]. The presence of neurons that can flip encourages the network to learn multiple parallel paths and to ignore the noise in its activations. Furthermore, using many binary neurons and sparse network activations increases the robustness of the network [157]. During inference, the stochastic process can be disabled using a fixed threshold of 0.5, resulting in a more stable network.

[225]: Hinton, Srivastava, Krizhevsky, *et al.* (2012), *Improving neural networks by preventing co-adaptation of feature detectors*

[157]: Ahmad and Hawkins (2015), *Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory*

Moreover, flipping neurons help to implement alternative pathways and cells required when a single cell is part of two mutually exclusive net fragments [168]. Alternative cells are a set of cells with afferent connections (c.f. Section 4.1.3). The stochastic property of Bernoulli neurons helps to solve the symmetry problem by encouraging each cell to undergo divergent connectivity changes during training so that it becomes part of mutually exclusive net fragments.

[168]: von der Malsburg and Bienenstock (1987), 'A Neural Network for the Retrieval of Superimposed Connection Patterns'

### 5.2.2 Practical Considerations

To prevent the network from being dominated by noise, it is crucial that most of the activation probabilities do not cluster around a mean value of $\mu = 0.5$. Otherwise, a significant proportion of neurons would have high

uncertainty about whether they should fire, leading to random firing patterns. This uncertainty problem occurs mainly in the initial phase of training when the network is not yet trained and therefore dominated by uncertainty.

This issue can be mitigated by pushing the activation probabilities towards 0 or 1. A simple approach is to apply the softmax function or to adjust the activation probabilities by a power factor $s$, i.e. $\boldsymbol{a} := \boldsymbol{a}^s$. The softmax function shifts the probabilities uniformly towards 0 or 1 while using a large factor $s$ drives most activations predominantly towards 0, and only high probabilities can remain high.

The advantage of using a factor $s$ is its adaptability: It can be set to a high value in the initial training phase and gradually lowered towards one as training progresses. This allows scoping with the network's uncertainty that reduces during training.

## 5.3 Processing Loops

The network requires different kinds of data processing loops that are divided as follows: The slow loop iterates through the images in the dataset; the medium loop iterates through different views of an image; and during the fast loop, inhibition takes place over multiple steps, whereby neurons that do not receive enough support are turned off. At each timestep, the innermost loop executes a step. Once the innermost loop is completed, a step is executed in an outer loop, and the process repeats. The specifics of these loops are described in the following.

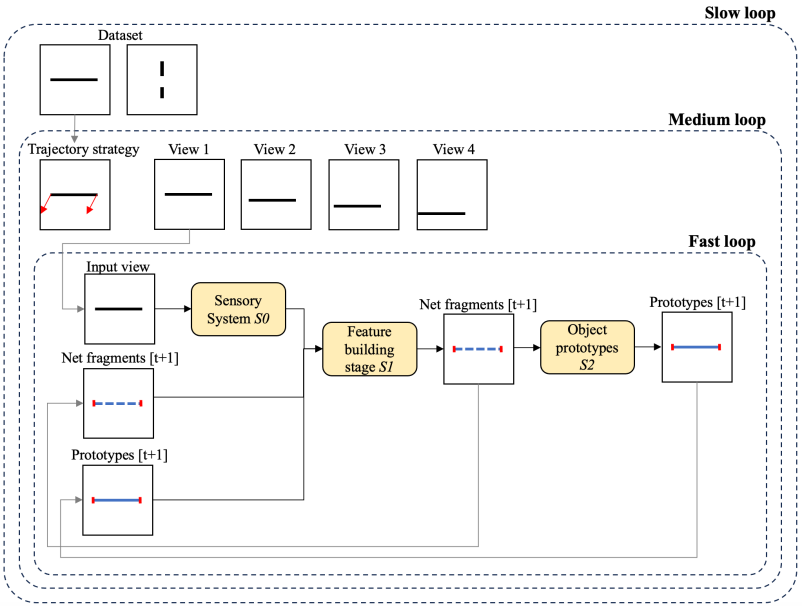


**Figure 5.2:** Processing loops of the network. From each sample in the dataset (slow loop), multiple views are generated (medium loop), and each view is processed over multiple timesteps by the model (fast loop).

**Slow Loop.** The dataset comprises multiple images. Each image in the dataset is processed one after the other, building the outermost loop. In Figure 5.2, an outer loop with a dataset containing vertical and horizontal lines is depicted, similar to data used in the conducted experiments (c.f. Section 6.1). However, depending on the application, different datasets could be used.

**Medium Loop.** For each image in the dataset, different views are sampled using data augmentation. In the experiments conducted in this thesis (c.f. Section 6.1), a trajectory strategy is implemented that moves the line continuously from an initial position to a target position. However, continuous movement is not mandatory, and random data augmentation can also be applied. It is important to inform the network that the same object with an identical inner structure is shown multiple times during the median loop so that it can learn to map it to the same object prototype despite different transformations and viewpoints.

**Fast Loop.** From each view, the sensory system creates neural activity that the network processes for $T \geq 1$ timesteps. During these timesteps, inhibition increases, and active cells that do not get lateral support from other cells are turned off. The fast loop aims to iteratively improve the net fragments in *S1* and the corresponding object prototypes over time. Therefore, the previous net fragments and object prototypes are fed into *S1* at every timestep, together with the sensory signal.

## 5.4 Sensory System *S0*

The goal of the sensory system is to perceive an input and extract multiple different features based on spatial neighbourhoods that can be used to build net fragments [20], [169] in the next stage. The input shape is $[C_{\text{in}} \times W \times H]$ where $C_{\text{in}}$ is the number of input channels, $W$ the image width and $H$ the image height. The output of this stage is of shape $[C_{\text{sensor}} \times W \times H]$, where $C_{\text{sensor}}$ is the number of output channels (i.e. the number of extracted features). Typically, $C_{\text{sensor}}$ is much larger than $C_{\text{in}}$, i.e. $C_{\text{sensor}} \gg C_{\text{in}}$. Thus, the sensory system extracts multiple features based on a local neighbourhood at each pixel location.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

As described in Section 5.1.1, a sensory system can be implemented with hand-crafted filters, Gabor filters [220], [221], or by using the first layer of a pre-trained convolutional network [35]. The advantage of using learned filters over fixed Gabor filters is that they can be optimised for the source data. However, this comes with the cost of additional filter training.

[220]: Gabor (1946), 'Theory of communication'

[221]: Granlund (1978), 'In search of a general picture processing operator'

[35]: LeCun, Boser, Denker, *et al.* (1989), 'Backpropagation Applied to Handwritten Zip Code Recognition'

A difficulty is that filter outputs are typically in continuous space and must be converted into a binary activation potential. One option is to normalise the sensor signal in the range $(0, .., 1)$ and to use this value as a probability to sample from a Bernoulli distribution, similar to the proposed Bernoulli neurons (c.f. Section 5.2). Another approach is to set all activations above a pre-defined threshold to 1 and to assign 0 to the values below the threshold. Alternatively, a third option is to use quantisation networks such as VQ-VAEs [226] as feature extractors. Such networks can map local features to a discrete value that can be translated into a binary activation pattern.

[226]: Oord, Vinyals, and (2017), 'Neural Discrete Representation Learning'

## 5.5 Feature Extracting Stage *S1*

The objective of *S1* is to build net fragments based on three types of input: The sensory signal, net fragments from the previous timestep,

and feedback from the object prototype stage. These three types of input serve the following purposes:

- **Sensory signal**: The sensory system extracts features from the image to provide initial cell activations. Theoretically, the sensory signal could be used only once at timestep $t = 0$. However, feeding the sensory signal into the network at every timestep stabilises the net fragments [2].
- **Previous net fragments**: Net fragments are the output of the stage *S1*, improved over multiple timesteps. To access information from the previous timestep, a recurrent connection is required. This connection is implemented by reusing the previous output (previous net fragments) as input.
- **Object prototypes**: The net fragments are mapped to object prototypes in *S2* using maplets. An inverse function is employed afterwards to map the object prototypes back to *S1* to provide information about detected objects.

2: This can be likened to closing our eyes: After they are closed, we can estimate the location of all objects but with uncertainty. Thus, the net fragments in our brain become unstable, and we can only estimate the object's position based on memories and previous net fragments.

There exist various ways of combining these three types of input signals. For example, the three arrays can be stacked[3] to hold all arrays available. However, stacking all three arrays would lead to a very high-dimensional input. A more sophisticated approach is to aggregate (some of) the matrices. In the experiments conducted in this thesis (c.f. Section 6.3), a straightforward approach is used: The previous net fragments can be overridden by the feedback from the object prototypes if the feedback is plausible. Only one of these two arrays is used, as these arrays are typically very similar and provide redundant information. The mapping from fragments to object prototypes and back reduces noise, as the feedback of *S2* is not a reconstructed version of the input but rather a reconstruction of an optimised object prototype. Consequently, the feedback of *S2* is typically less noisy and, therefore, preferred.

3: Stacking arrays refers to the process of combining multiple arrays along a specified axis to create a new multidimensional array, e.g. create an array of size $[2 \times 2]$ out of two arrays with size $[1 \times 2]$.

Nevertheless, the feedback from *S2* is not always correct and only incorporated if it is highly similar to the net fragments formed in *S1*. Especially at the beginning of training or when observing unknown objects, the feedback of *S2* should not be incorporated as it can be wrong. Such invalid feedback can be detected by measuring the similarity between net fragments and feedback, for example, using the Jaccard similarity (c.f. equation (5.6)). Thus, the net fragments are only overridden if the error is below a pre-defined threshold and overriding most likely improves results.

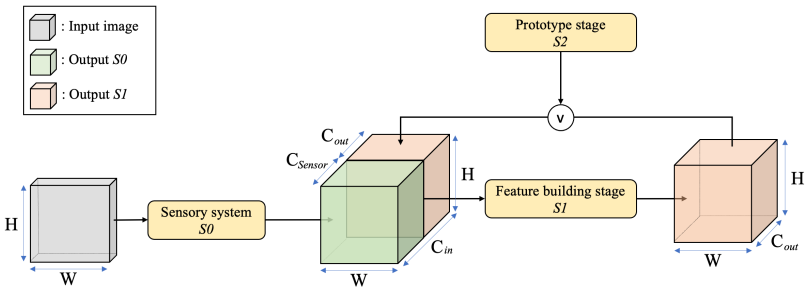


**Figure 5.3:** Visualisation of the input and output arrays of *S1*. The input into *S1* is the output of the sensory system and the previous output (recurrent connection). The feedback form *S2* can override the previous output.

In the following, the number of input channels is denoted as $C_{\text{in}}$ and the number of output channels is denoted as $C_{\text{out}}$. Note that the feedback matrix from *S2* has the same shape as the net fragments formed in *S1*.

Since the previous output (net fragments) or the feedback from *S2* is combined with the sensory signal and used as input, the number of input channels is larger than the number of output channels and defined as $C_{\text{in}} = C_{\text{sensor}} + C_{\text{out}}$. An overview about the input and output of *S1* is depicted in Figure 5.3: The output of *S1* is optionally replaced by the feedback of *S2*. In both cases, this array has a size of $[C_{\text{out}} \times W \times H]$ and is stacked with the output from the sensory system of size $[C_{\text{Sensor}} \times W \times H]$, resulting in an input of size $[C_{\text{in}} \times W \times H]$.

### 5.5.1 Lateral Support

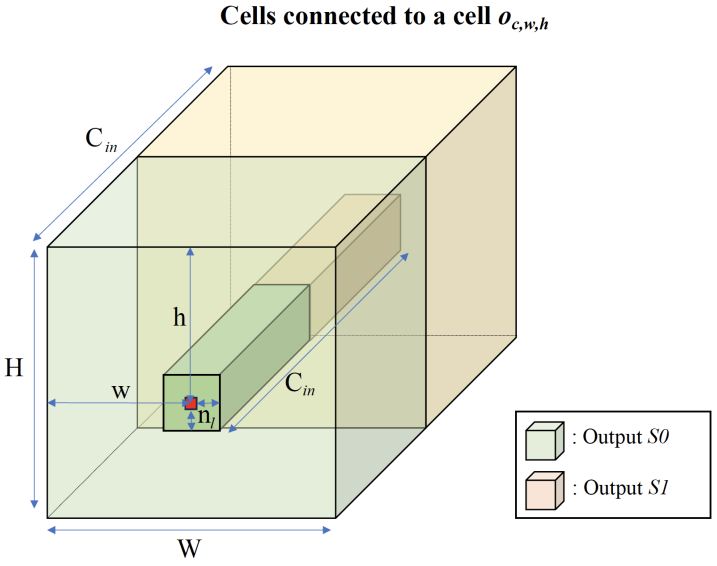


**Figure 5.4:** The local neighbourhood of a cell $o_{c,w,h}$: The outer cuboid represents the entire input, the inner cuboid the source cells that are connected to the target cell $o_{c,w,h}$.

In this section, it is described how lateral support can be implemented. A single output cell in *S1* is denoted as $o_{c,w,h}$, where $c \in \{0, \ldots, C_{\text{out}}\}$ is the output feature channel, and $w \in \{0, \ldots, W\}$ and $h \in \{0, \ldots, H\}$ denote the spatial location of the cell.

The lateral connections are limited to a distance of $n_l$ cells along the vertical and horizontal axes but are not limited along the input feature channels. Therefore, an output cell $o_{c,w,h}$ has lateral connections to all inputs within the range $[(0, \ldots, C_{\text{in}}) \times (w - n_l, \ldots, w + n_l) \times (h - n_l, \ldots, h + n_l)]$. Such a local neighbourhood is depicted in Figure 5.4 for a cell $o_{c,w,h}$. An input channel contains either features extracted by the sensory system or the previous net fragments. Thus, a cell can access all features detected by the sensory system or the previous cell states (recurrent connection) in its spatial neighbourhood. Furthermore, according to this definition, the cell is also connected to its own previous cell state at timestep $t - 1$. This type of connection is called self-support and allows a cell to remain active over time by supporting itself. Self-support is crucial to ensure the network is stable at the beginning of training (c.f. Section 5.5.3). However, after training for a short time, lateral connections are built, the inhibition strength increases so that self-support is not sufficient anymore, and lateral support from other cells is required to remain active.

Patterns can appear at different positions within an image, and the network should be able to recognise it independent of its spatial location [33], [34]. Therefore, lateral support must be position equivariant.

Convolutional architectures solve this problem with convolutional filters shifting over each pixel location [35]. This mechanism can also be used to implement the lateral connections: When using a convolutional kernel $\boldsymbol{W}$ with size $[C_{\text{out}} \times C_{\text{in}} \times (2n_l + 1) \times (2n_l + 1)]$, each output cell has a connection to its local neighbourhood as defined above. Since this kernel is applied at all cell positions, a cell is supported by neighbouring cells that represent the same pattern regardless of the spatial position. The weights within a kernel correspond to the support strength, indicating how much a neighbouring cell supports another cell. The support strength of a cell $o_{c,w,h}$ (i.e. how much support a cell receives from its neighbours) can be calculated with the convolutional operation:

[35]: LeCun, Boser, Denker, *et al.* (1989), 'Backpropagation Applied to Handwritten Zip Code Recognition'

$$o_{c,w,h} = \sum_{c'=0}^{C_{\text{in}}} \sum_{w_i=0}^{2n+1} \sum_{h_i=0}^{2n+1} \boldsymbol{W}_{c,c',w_i,h_i} \cdot o_{c',w-n+w_i,h-n+h_i} \tag{5.2}$$

Thus, the same weight $\boldsymbol{W}$ is applied at all input locations, defining the support strength of a cell based on its neighbourhood and independent of its position. This operation corresponds precisely to the output of a convolutional layer without bias term. Please note that the target cell represents a cell's state at time $t$ (denoted by $c \in \{0, \ldots, C_{\text{out}}\}$) and can access the state of all source cells (denoted by $c' \in \{0, \ldots, C_{\text{in}}\}$), i.e. all sensory cells or recurrently connected cell states at $t - 1$.

### 5.5.2 Hebbian Updates

The previous section introduces how a convolutional kernel $\boldsymbol{W}$ can be used to model the lateral support of neighbouring cells. In this section, it is described how the support strength, i.e. the weights of $\boldsymbol{W}$, can be learned. The human brain's learning algorithm is based on local self-organisation and unsupervised (or self-supervised) learning [20]. The biologically most plausible learning algorithm is Hebbian learning [40].

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

Hebbian learning evaluates consistency at each synapse and is well suited for learning lateral connections: If two cells are active together ("fire together"), their weight increases ("wire together") - this plasticity principle corresponds to the definition of lateral support. During training, the cells are activated in a specific pattern based on the sensory input. Hebbian learning strengthens the connections between simultaneously active cells. Thus, the connection strength between cells associated with the same net fragment is increased, leading to stronger mutual support.

Hebbian learning is introduced in Section 2.4.1, and it is described how the weight between two cells changes if they fire together (c.f. equation (2.15)). However, in this thesis, lateral support is implemented as a convolutional kernel that is applied at all input positions. Since the same weight is applied at different positions, the update of a connection depends not only on two cells but on multiple cells:

$$\Delta w_{c,c',k_w,k_h} = \eta \sum_{w=0}^{W-2n_l-1} \sum_{h=0}^{H-2n_l-1} o_{c,w+k_w,h+k_h} \cdot o_{c',w+k_w,h+k_h} \tag{5.3}$$

In this formula, $\eta$ is the learning rate, and $k_w \in \{0, ..., 2n_l + 1\}$ and $k_h \in \{0, ..., 2n_l + 1\}$ is the kernel index along the horizontal and vertical axis[4]. Please note that the weight between two laterally connected cells is increased if both are active simultaneously, whereby one cell is considered as the source cell (denoted as input channel $c'$) of the target cell (denoted as output channel $c$).

4: For example, $w_{1,2,3,4}$ represents the weight between the first output channel and the second input channel, located in the kernel's third column and fourth row.

During training, the network may encounter similar patterns multiple times, leading to strong connections between specific cells. Therefore, weight must be normalised so that these lateral connection strengths and the post-synaptic activity cannot grow towards infinite. After calculating the weight update and adding it to the weight matrix $\boldsymbol{W}$, the weights are normalised per output channel by dividing each channel by its Euclidean norm. This ensures that the weights are roughly in the range $(0, ..., +1)$. Therefore, one cell can only provide limited support to another cell, and multiple cells must support a cell to remain active.

### 5.5.3 Initialisation

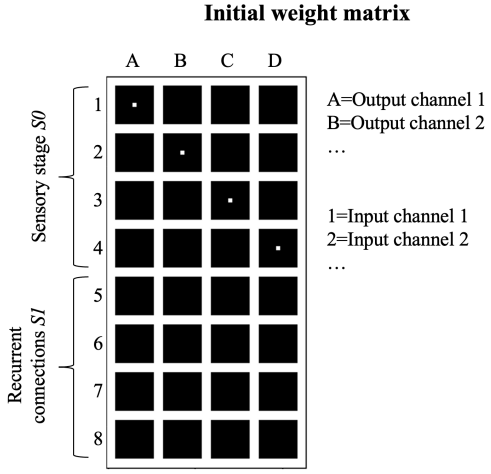


**Figure 5.5:** Initialisation of the lateral weight matrix. The weight at the middle of a kernel, whose input and output channel have the same index, is set to 1.

The previous sections explain that lateral support can be implemented with convolutional kernels and that the weights are learned using the Hebbian rule. In this section, it is discussed how these kernels should be initialised. If the kernels are not properly initialised, the activations are unstable and typically converge to weights that activate all or no cells. A proper initialisation should fulfil the following criteria:

- **Diversity:** A proper initialisation should break the symmetry. If, for example, all kernels are initialised identically, each kernel learns the identical pattern when applying Hebbian updates. This makes all output channels identical, and effectively the model learns only one feature.

- **Stability:** The Hebbian update is calculated after a timestep of the inner loop; therefore, the cells must be stable over timesteps. For example, if the lateral support is only initialised with 0 values, all cells turn off immediately. In that case, no cell is active after the first timestep, and no support can be learned.

An initialisation strategy that fulfils these criteria is when the weights are initialised with self-support. Self-support means that each cell supports itself to remain active over time, i.e. the recurrent connection to the same cell at the previous timestep must be set to 1. Self-support can be implemented by setting all weights $w_{c,c',w,h}$ of a kernel to 1 at the indexes that fulfil $c' = c$, $w = n+1$, and $h = n+1$. Thus, the weight at the middle of a kernel with the same input and output channel index is set to 1 while the other weights are set to 0. This initialisation strategy also works for kernels with a different number of input and output channels and is shown in Figure 5.5. Such a weight matrix copies the input activations to the output and ensures that the cell's activations at time $t$ and $t+1$ are identical. Therefore, initially, active cells receive a support of 1 (i.e. one other cell with a lateral weight of 1 is active). However, after applying the Hebbian learning rule, the weights are updated to capture the data's statistics so that active cells receive more support.

### 5.5.4 Inhibition

A continuous value representing the support strength is obtained after applying the convolutional operation to the binary input signal. The support strength has to be normalised into the range of $(0, \ldots, 1)$ to be used as an activation probability of a Bernoulli neuron. In the context of neuroscience, this normalisation is interpreted as the brain's inhibitory signal [41], [102]: After initialisation of the weight matrix $\boldsymbol{W}$, the highest possible support strength per cell is 1 as only self-support exists. During training, the maximum support strength increases significantly (up to $2n_l + 1$) as other cells start supporting a given cell. Consequently, the support required to remain active increases during training. However, the support strength does not only change during training. It also varies over timesteps, executed in the inner loop: At time $t = 0$, only the sensory system provides input. At timestep $t = 1$, both the sensory input and the recurrent connections can be active, typically leading to increased support. Therefore, the support strength is highly time-dependent.

[41]: Coombs, Eccles, and Fatt (1955), 'The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential'

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

Besides being time-dependent, the support strength also changes depending on the data: Some images generally contain more features, leading to more active cells and higher lateral support. Furthermore, support varies across different spatial locations, as some image regions typically contain more features than others. Therefore, cells in such regions typically receive more support than those in regions with fewer features and fewer activated cells.

The inhibition strength must be highly adaptive to cope with such dynamic support strength. Variation due to the training progress is taken into account by dividing the support strength through the highest possible support. This is implemented by dividing the activations in each channel $c$ through the sum of weights in the same feature channel $c$. Thus, if an output channel has many synapses that could provide lateral support, inhibition is stronger:

$$o_{c,w,h} := \frac{o_{c,w,h}}{\sum_{c'=0}^{C_{\text{in}}} \sum_{w_i=0}^{W} \sum_{h_i=0}^{H} w_{c,c',w_i,h_i}} \tag{5.4}$$

This formula also accounts for different levels of support between channels, i.e. the problem of one feature channel receiving more support than others is mitigated.

A solution to deal with varying support strength due to the timestep, input data, and spatial location can be found in the human brain. Neuroscience findings suggest an upper limit of concurrently active incoming synapses for active cells [224]. Thus, each cell has not only a lower limit of lateral support but also an upper limit. The support is reduced if a cell's support is above a pre-defined threshold $\rho$. Preliminary experiments suggest that values in the range $\rho = (1.2n_l, ..., 1.5n_l)$ work well. The support strength of a cell $o_{c,w,h}$ is modified as follows:

[224]: Kandel (2013), *Principles of neural science*

$$o_{c,w,h} := \begin{cases} o_{c,w,h} & \text{, if } o_{c,w,h} < \rho \\ \rho - \frac{1}{2}(o_{c,w,h} - \rho), & \text{otherwise} \end{cases} \tag{5.5}$$

This function is visualised in Figure 5.6. No normalisation is applied if a

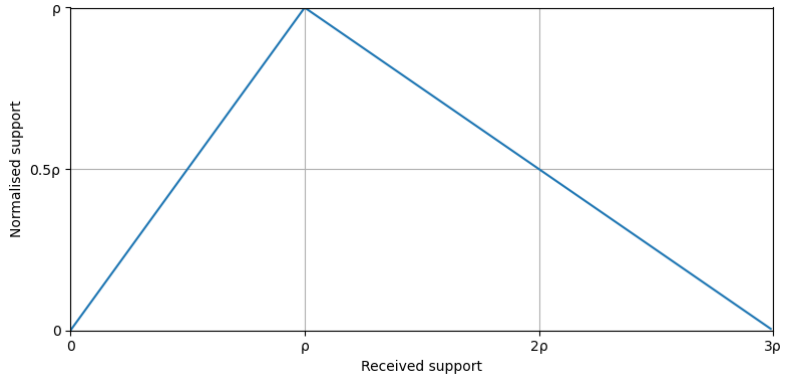


**Figure 5.6:** The visualisation of formula equation (5.5). The x-axis shows the received support, and the y-axis shows the support after normalisation: As soon as the received support is bigger than $\rho$, it is reduced with a slope of $-0.5$.

cell's activation is below $\rho$. However, if the activation is above $\rho$, the cell's lateral support strength is decreased with a slope of $-0.5$. Experimental findings suggest that this upper support limit helps overcome varying support strengths during training.

### 5.5.5 Alternative Cells

As described in Section 4.1.3, alternative cells and pathways are necessary to deal with different patterns that activate similar feature cells. "Alternative" in this context means that only one cell among the alternatives is active, i.e. these cells are mutually exclusive. At the beginning of training, alternative cells are copies of an initial cell. However, after training, these cells contribute to different patterns and have different behaviour.

Alternative cells can be implemented by adding additional (alternative) output channels. The duplication factor $\kappa$ defines how many alternative channels should be added. Thus, the weight $\boldsymbol{W}$ with alternative cells is of shape $[(\kappa \cdot C_{\text{out}}) \times C_{\text{in}} \times (2n+1) \times (2n+1)]$.

In Figure 5.7, it is visualised how the weight $\boldsymbol{W}$ with a duplication factor of $\kappa = 2$ are initialised. In that case, $\kappa = 2$ output channels are mutually exclusive, leading to activation in only one of these channels. Implementing mutually exclusive cells requires competition between

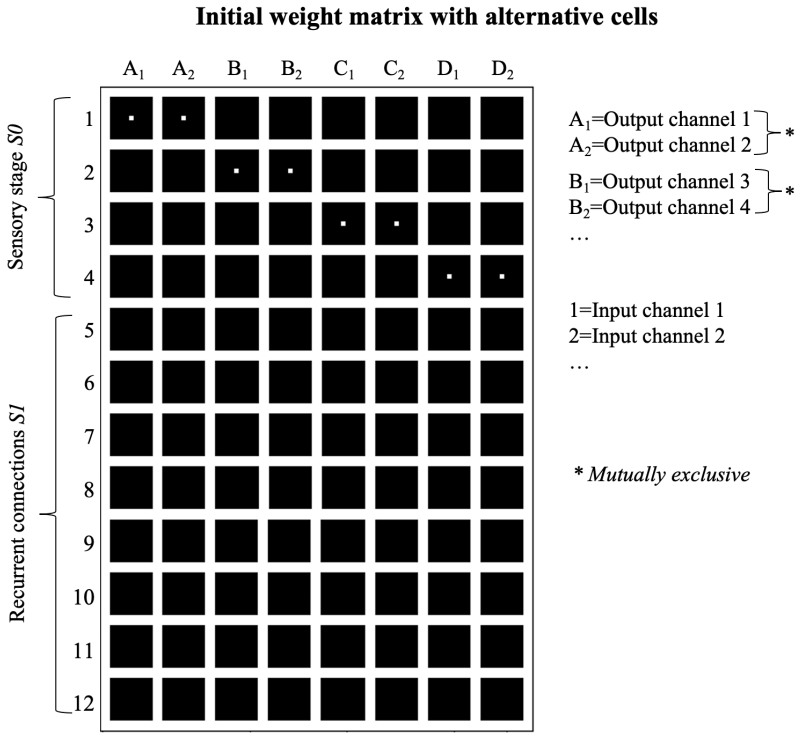


**Figure 5.7:** Initialisation of the lateral weight matrix with alternative cells. The initialisation is similar to the one shown in Figure 5.5, except each channel is duplicated $\kappa = 2$ times.

alternative cells so that only the most suitable cell can remain active if multiple exclusive cells are active [168]. For example, such competition can be implemented by comparing how well a specific activation pattern fits the already learned lateral support [102]–[104]. Another option is implementing inhibition between the competing channels so that the dominant channel turns off the other channels.

[168]: von der Malsburg and Bienenstock (1987), 'A Neural Network for the Retrieval of Superimposed Connection Patterns'

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

[103]: Joshi and Triesch (2009), 'Rules for information maximization in spiking neurons using intrinsic plasticity'

[104]: Teichmann and Hamker (2015), 'Intrinsic Plasticity: A Simple Mechanism to Stabilize Hebbian Learning in Multilayer Neural Networks'

### 5.5.6 Measuring Support Quality

After learning net fragments, it is crucial to evaluate their quality. A simple approach is to measure the support needed to remain active and the average support active and inactive cells receive. At the beginning of training, self-support is used, and therefore, the average support of active cells is 1, and the average support of inactive cells is 0. However, during training, lateral connections are learned that support cells to remain active. This leads to higher activation in general, which, in turn, increases the threshold to remain active. Thus, the average activation of the cell increases as well as the threshold to remain active. These statistics can be measured over the training process to evaluate the quality of the net fragments.

As a second metric, we can measure the robustness against noise. Net fragments should only support cells that are activated by a learned pattern. Thus, cells activated by noise should not receive enough support and be turned off after a few cycles. Therefore, as a second metric, we can add noise to the input image and measure how many cells in the sensory system are activated due to the noise and what ratio of them remains active after applying lateral connections.

## 5.6 Prototype Stage *S2*

In stage *S2*, the net fragments are mapped to idealised reference objects. This mapping is implemented by projection fibres [175], [203], which are grouped into maplets [209]. As soon as a control unit of a maplet detects a strong correlation between the neurons it connects in *S1* and *S2*, it initialises the mapping by turning on its fibres.

This mapping from fragments to reference representations seems to be one of the core algorithms of the biological visual system as it can solve the binding problem [227], [228], i.e. answers the questions of how visually perceived objects are bound together based on their properties such as shape, texture, colour, contour, or motion.

Implementing such a mapping poses various challenges. In the following, some simplifications are assumed that are ignored in the proposed framework and have to be solved in future work.

- **Storing object prototypes in *S2***: It is assumed that the reference representations are already stored in *S2*. Consequently, the mapping process is reduced to finding the most suitable projection, disregarding that the object prototype might not be stored yet. Nevertheless, it would be desirable that the network can, for example, when encountering high uncertainty, autonomously recognise the absence of a proper object prototype and create one.

- **Enhancing object prototypes in *S2***: It is assumed that the object prototypes are idealised, can remain static, and do not require further updates. Findings from psychology suggest that our brain is highly structured and might contain such prototypes from birth [229]. However, these prototypes are also known to be optimised with increasing experience over time [229]. Moreover, updating prototypes is important when novel prototypes are stored, as an optimised form of such a reference object typically cannot be derived from a single sample.

- **Object-centered input**: The input images are assumed to contain exactly one object rather than complex scenes containing multiple objects. This allows to map a part of an image, i.e. the region where the object is located, to exactly one object reference frame. In real-world scenarios, visual scenes often comprise multiple objects, requiring the model to map a single input to multiple prototypes. Consequently, an attention mechanism becomes essential to identify object boundaries before comparing them to suitable references.

- **Pre-defined projection fibres**: It is assumed that the projection fibres already exist from the beginning and remain unchanged throughout the learning process. Thus, the learning process focuses on the activation of control units. This assumption requires predefining many fibres, some of which may be unnecessary, making the system less efficient. Dynamically growing or pruning fibres could make the system more efficient and robust.

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

[209]: Zhu and von der Malsburg (2004), 'Maplets for correspondence-based object recognition'

[227]: Revonsuo and Newman (1999), 'Binding and Consciousness'

[228]: Feldman (2013), 'The neural binding problem(s)'

[229]: Simion and Di Giorgio (2015), 'Face perception and processing in early infancy'

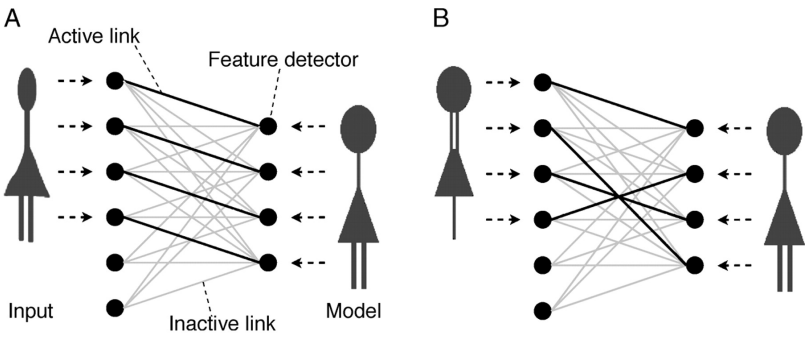


**Figure 5.8:** Visualisation of the correspondence problem: Corresponding points between an input and a model must be linked. All potential correspondences are visualised as grey lines, while the correct correspondences are shown as black lines. Example ***A*** shows a correct correspondence, while example ***B*** visualises a wrong correspondence, although similar features are connected. The image is from [31].

### 5.6.1 Correspondence-Mapping

So far, the problem is described as mapping entire net fragments to object prototypes. However, this is a somewhat simplified view, as many similar objects differ slightly. Therefore, one cannot just compare objects but rather multiple features that define an object. This problem of mapping features extracted from an input image to features from a reference object is known as the correspondence problem. Wolfrum *et al.* [31] motivate the correspondence problem based on Figure 5.8. This figure depicts two stick figures where the input's features, such as the head or neck, must be linked to the corresponding model's features. There exist various mappings, symbolised as grey lines. The correspondence problem is to find a subset of these links that are the correct correspondences (visualised as black lines in example ***A***).

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

Unfortunately, it is not sufficient to calculate the similarity between features as illustrated in Figure 5.8 example ***B***. Different images of the same object can vary significantly, resulting in a high degree of similarity between non-corresponding features [230]. The reason is that not only the features define an object but also their spatial arrangement. Therefore, correspondence-based systems must take both into account.

[230]: Wiskott (1999), 'The role of topographical constraints in face recognition'

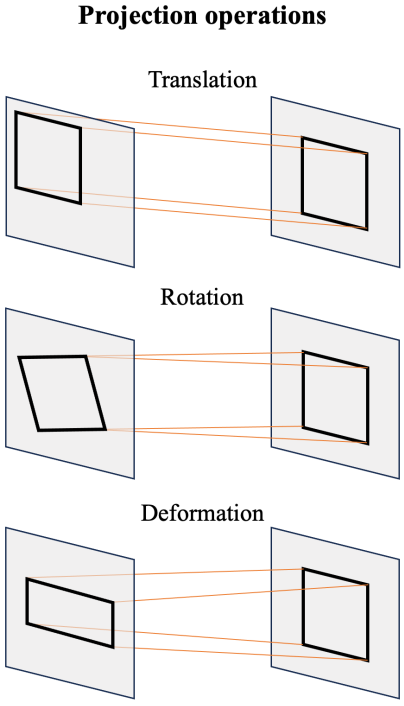


**Figure 5.9:** Different projection operations that must be implemented with projection fibres. Projection fibres should make the mapping process invariant to translation, rotation, and deformation.

In Figure 5.9, the operations needed to deal with different spatial arrangements are visualised. Projection fibres must be invariant to translation, rotation, and deformation. Such transformation can occur globally, as depicted in Figure 5.9, and locally. In Section 5.6.2, it is described how the similarity between local areas connected by a projection fibre can be measured. This statistic is needed to decide whether a control unit should switch on. Subsequently, in Section 5.6.3, it is described how the projection fibres can be wired.

### 5.6.2 Measuring Similarity

Correspondence-based mapping requires measuring feature similarity. Since binary neurons are used, similarity can be calculated by comparing the neurons' activity. They are similar if both neurons are on or off; if one neuron is on while the other is not, they are dissimilar. However, multiple neurons exist at the same spatial location, representing various features. The mapping process compares how similar the spatial locations between a net fragment and a corresponding location of the reference object are. Therefore, all neurons at the same location are compared.

Both the net fragments in $S1$ and the prototypes in $S2$ are of shape $[C_{\text{out}} \times W \times H]$. Thus, per spatial location $(x, y)$ (where $x \in \{0, \ldots, W\}$ and $y \in \{0, \ldots, H\}$), exists a one-dimensional feature vector of length $C_{\text{out}}$. This vector is denoted as $\boldsymbol{a}_{S1,(x,y)}$ for $S1$ and $\boldsymbol{a}_{S2,(x,y)}$ for $S2$. These two vectors can be compared using the Jaccard similarity $J$, which is defined as:

$$J_{x,y} = J(\boldsymbol{a}_{S1,(x,y)}, \boldsymbol{a}_{S2,(x,y)}) = \frac{|\boldsymbol{a}_{S1,(x,y)} \cap \boldsymbol{a}_{S2,(x,y)}|}{|\boldsymbol{a}_{S1,(x,y)} \cup \boldsymbol{a}_{S2,(x,y)}|} \tag{5.6}$$

The term $|\boldsymbol{a}_{S1,(x,y)} \cap \boldsymbol{a}_{S2,(x,y)}|$ describes the number of corresponding cells that are activated in both $S1$ and $S2$, while the second term $|\boldsymbol{a}_{S1,(x,y)} \cup \boldsymbol{a}_{S2,(x,y)}|$ is the number of cells that are activated in either $S1$ or $S2$. Features that are deactivated in $S1$ and $S2$ are not taken into account. The Jaccard similarity $J$ is in the range of $(0, \ldots, 1)$, where 1 is the highest possible similarity.

As motivated in Figure 5.8 example ***B***, having a high similarity between two feature vectors is not enough to describe the quality of a projection, as also the spatial arrangement must be considered. Therefore, not only the similarity between two spatial locations $(x, y)$ is compared but all neurons in a local neighbourhood. Such a comparison is visualised in Figure 5.10: The similarity between the two vectors (depicted as dark blue circles) also depends on their local neighbours (light blue circles). The size of the local neighbourhoods is defined as $n_J$, and the

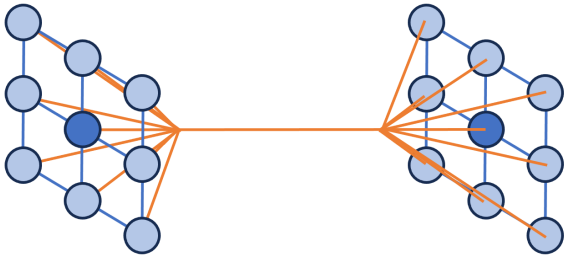

**Figure 5.10:** The Jaccard similarity is calculated between two vectors (coloured in dark blue) and their spatial neighbours (coloured in light blue).

local neighbourhoods as $\boldsymbol{A}_{S1,(x,y)} = (\boldsymbol{a}_{S1,(x-n_J,y-n_J)}, \ldots, \boldsymbol{a}_{S1,(x+n_J,y+n_J)})$ and $\boldsymbol{A}_{S2,(x,y)} = (\boldsymbol{a}_{S2,(x-n_J,y-n_J)}, \ldots, \boldsymbol{a}_{S2,(x+n_J,y+n_J)})$, respectively. The Jaccard similarity between two vectors that consider the local neighbourhood is thus defined as:

$$J_{x,y} = J\left(\boldsymbol{A}_{S1,(x,y)}, \boldsymbol{A}_{S2,(x,y)}\right) \tag{5.7}$$

Considering a local neighbourhood can be likened to having local support, as is the case for net fragments. Multiple cells must support the mapping to be activated by the corresponding control unit. For example, the similarity between two dissimilar vectors can still be high if their context is similar, which helps deal with noise in the data. On the other hand, the similarity between two identical vectors is low if their context is dissimilar and therefore does not match. Such local support thus increases robustness and provides higher similarity for spatially correctly arranged features.

### 5.6.3 Mapping Process

In the previous section, it is discussed how the similarity between activity patterns of neurons connected by projection fibres can be calculated. In the following, it is described how projection fibres can be wired.

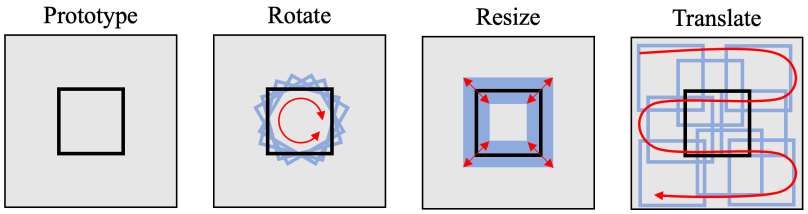


**Figure 5.11:** Different operations applied to a prototype.

A straightforward approach is to apply different operations such as translation, rotation, and deformation to a reference frame as shown in Figure 5.11. These operations can be used individually or in combination, resulting in multiple augmented prototype versions. Thereby, the shift of each neuron is tracked, and a projection fibre is used to map a neuron from the prototype to the corresponding neuron in the augmented version.

However, such an object-dependent mapping would not scale as each object requires a multitude of fibres. Instead, the mapping must be object-independent and focus on local features. In this context, "object-independent" still means that the feature similarity has to be calculated per object but that not each different object requires a unique set of projection fibres. Shifter circuits [207], [208] are a kind of hierarchical mapping that implement such an object-independent mapping focusing on local features. Shifter circuits define different hierarchical levels, whereby each level is responsible for a specific operation, such as a rotation, shift, or translation. During the fast processing loop when the input image is static, very fast Hebbian plasticity [40] can be used to create a topographical projection, i.e. to connect neighbouring cells in *S1* with neighbouring cells in *S2* as shown by Fernandes *et al.* [32]. Furthermore, Fernandes *et al.* [32] also demonstrates that the medium processing loop providing different image views can be leveraged to teach the model what the same object from different viewpoints looks like. This helps to improve the mapping processing and the prototypes.

[207]: Anderson and Essen (1987), 'Shifter circuits'

[208]: Olshausen, Anderson, and Van Essen (1993), 'A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information.'

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

Each projection fibre calculates the similarity between the neurons it connects when processing an image. Afterwards, the average similarity is calculated per maplet, and this similarity is used as the activation probability of a Bernoulli neuron that can turn on the corresponding control unit. Similar to *S1*, this leads to many activations at the beginning and inhibition is used to turn some of the maplets off. Maplets support each other locally and can only remain active if neighbouring features in *S1* remain neighbouring features in *S2*. Thus, the better a set of maplets can map a prototype to an object reference, the higher its probability of remaining active. For more details on the implementation, please refer to the work by Anderson *et al.* [207], Olshausen *et al.* [208], and Fernandes *et al.* [32].

[207]: Anderson and Essen (1987), 'Shifter circuits'

[208]: Olshausen, Anderson, and Van Essen (1993), 'A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information.'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

### 5.6.4 Feedback to *S1*

*S2* serves two purposes: It not only maps an input to a reference frame to obtain a transformation-invariant representation but also provides feedback to *S1* by mapping the reference object back to *S1*. To provide feedback, the most plausible prototype is selected. As described in Section 5.5, this prototype from *S2* can overwrite the representations in *S1* and is thus incorporated into the learning process in *S1*.

The representations in *S1* and *S2* might slightly vary as the net fragments in *S1* represent a particular instance of an object. In contrast, *S2* represents a generalised (optimised) version of the same object. However, these representations should still be similar (otherwise, a proper prototype in *S2* is missing). Therefore, *S1* receives an optimised version of the net fragments as input. This can be considered a bias towards a better representation. By applying Hebbian updates between the input in *S1* and the prototypes from *S2*, *S1* learns to convert its features to an optimised version. Thus, the feedback from *S2* provides additional support in *S1* and can help to build better representations and fragments.

#### Measuring *S2* Quality

*S2* can be considered an associative memory, mapping net fragments to the most suitable reference frame. A strength of such a system is that it is highly robust and can deal well with noisy data or occluded objects [113]. Therefore, *S2* can be evaluated by occluding data and letting it reconstruct it as it is done for many deep learning systems [231], [232].

[113]: Ramsauer, Schäfl, Lehner, *et al.* (2021), *Hopfield Networks is All You Need*

[231]: Han, Laga, and Bennamoun (2021), 'Image-Based 3D Object Reconstruction'

[232]: Qu, Liu, Wang, *et al.* (2022), 'TransMEF'

Furthermore, *S2* maps net fragments to object prototypes. Thus, another way to evaluate the quality of *S2* is to measure if the input is mapped to the corresponding object prototype. The mapping accuracy can be measured on an object or feature level with typical classification metrics [217] such as accuracy or F1-score.

[217]: Schmarje, Santarossa, Schroder, *et al.* (2021), 'A Survey on Semi-, Self- and Unsupervised Learning for Image Classification'

Please note that *S2* highly depends on *S1*. To evaluate the performance *S2* independently, it has to be encapsulated from *S1*. This can be achieved using strongly augmented reference objects from *S2* as input instead of actual net fragments from *S1*.

# 6 Experiments



The following sections describe preliminary experiments conducted to assess the feasibility of the proposed vision framework. The experiments' primary focus is on the first stage *S1*, as it is the foundation to develop the second stage *S2*. Furthermore, no work exists about implementing stage *S1*, in contrast to self-organising projection fibres, which are the central element of *S2*. Therefore, the most pressing issue is demonstrating the effectiveness of *S1* according to the proposed principles. To demonstrate the effectiveness of incorporating feedback from the second into the first stage, a simple mockup is used to simulate *S2*.

In the remainder of this chapter, a simple dataset is introduced in Section 6.1. Next, the implementation of the sensory system is presented in Section 6.2, the implementation of the feature extracting stage in Section 6.3, and the implementation of the prototype stage in Section 6.4. The results obtained from these experiments are presented in Chapter 7.

## 6.1 Dataset

The objective of the experiments conducted is to demonstrate that building net fragments can be implemented using Hebbian learning [40] (c.f. Section 2.4.1). Thereby, the focus is on researching novel principles and comprehending and analysing the networks' output rather than scaling the model to large datasets or pushing benchmarks. Consequently, a straightforward dataset is introduced that comprises straight lines only.

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

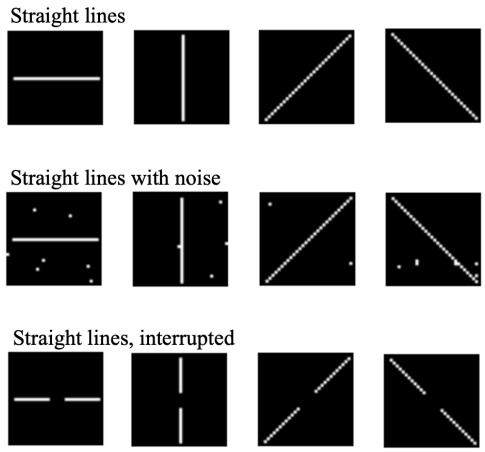


**Figure 6.1:** Sample images from the straight line dataset. The first row shows the images used for training, the second row shows images with noise, and the third row shows discontinuous lines. The images in the second and third rows are only used for evaluation.

The dataset is generated online, meaning images are created when required and not stored on the disk. This provides high flexibility and allows dynamically generating different images, which is helpful, especially during evaluation. For the conducted experiments, black and white images with a dimensionality of $[1 \times 32 \times 32]$ are used, whereby 1 is the number of colour channels and $[32 \times 32]$ is the image's width and height,

respectively. The dataset is binary and depicts different types of lines. All background pixels are set to 0, while the pixels representing a line are set to 1. No normalisation is used as the data distribution of the binary dataset is already well-aligned, and normalisation is unnecessary.

During *training*, four fixed images are used, depicting vertical, horizontal, and two diagonal lines (one with a positive and one with a negative slope). The starting and ending coordinates $(x, y)$ of these lines are as follows: A horizontal line from $(2, 16)$ to $(30, 16)$, a vertical line from $(16, 2)$ to $(16, 30)$, a diagonal line with positive slop from $(2, 2)$ to $(30, 30)$, and a diagonal line with negative slop from $(2, 30)$ to $(30, 2)$. These lines are shown in the first row of Figure 6.1. The training dataset consists of 500 images, randomly sampled in each epoch.

During *testing*, different kinds of images are generated. First, an optional noise parameter is introduced. In this thesis, this parameter is set to 0.005, letting each neuron with a probability of 0.5% switch its activation from 0 to 1, or vice versa. Thus, on average, 5.12 pixels in the image change their activation. Such images with noise are shown in the second row of Figure 6.1. Second, the continuous line is interrupted in the middle, resulting in a discontinuous line. The length of the break is a hyperparameter and within a range of 0 to 20 pixels. Lines with a break of 5 pixels are shown in the last row of Figure 6.1. Third, a trajectory strategy generates different views of an image, as described in Section 5.3. This trajectory strategy allows the specification of starting and ending coordinates and generates a set of lines encompassing all trajectories between these coordinates. The result of such a trajectory strategy is shown in Figure 6.2, where a horizontal line is converted to a diagonal line with a positive slope. Note that these strategies are only utilised during testing to evaluate the behaviour of the network when encountering lines not seen during training.

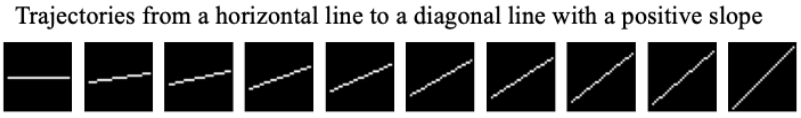


**Figure 6.2:** A sample trajectory strategy is applied to the horizontal line so that it becomes, over several steps, a diagonal line.

## 6.2 Sensory System *S0*

The dataset used in this thesis is straightforward and does not require learning highly specialised filters. Furthermore, it is expected that learning proper filters is a simple task that poses not many challenges as deep learning networks have proven themselves as excellent pattern recognition systems able to learn filters that are well tuned to the data domain [3], [6], [233].

Hand-crafted filters are sufficient and preferred for the conducted experiments as they do not require additional training and can be highly interpretable. Interpretability facilitates a better understanding of the extracted features and better comprehension of the net fragments built in the subsequent stage based on these features.

The hand-crafted filters used for the experiments are illustrated in Figure 6.3. Four filters are applied, each specialising in a different type

[3]: Bhatt, Patel, Talsania, *et al.* (2021), 'CNN Variants for Computer Vision'

[6]: Bertolini, Mezzogori, Neroni, *et al.* (2021), 'Machine Learning for industrial applications'

[233]: Zou, Chen, Shi, *et al.* (2023), 'Object Detection in 20 Years'

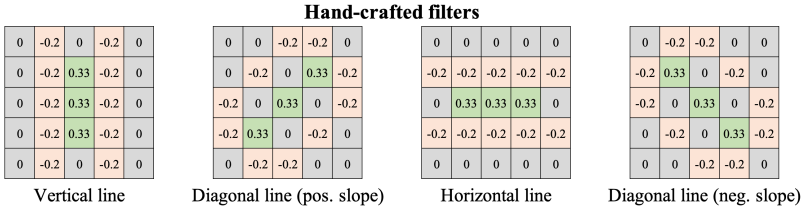


**Figure 6.3:** The hand-crafted filters of the sensory system are used to extract features from the images. The filters are optimised for horizontal, vertical, and diagonal lines.

of line. These filters function similarly to a convolutional layer with frozen (non-trainable) weights and no bias term. During processing, the filters are shifted with a stride of 1 across the input image of size $[1 \times 32 \times 32]$ so that they are applied at all input positions. The borders are padded with 0 values to keep the input and output sizes identical. After applying the four filters, the output has a shape of $[4 \times 32 \times 32]$. However, the activations can be in the range of $(-1, \ldots, +1)$. In order to obtain binary activations, these activations are used as the firing probability of a Bernoulli neuron, whereby values $\leq 0$ are mapped to an activation probability of 0%. Thus, the network's binary activation is sampled from a Bernoulli distribution.

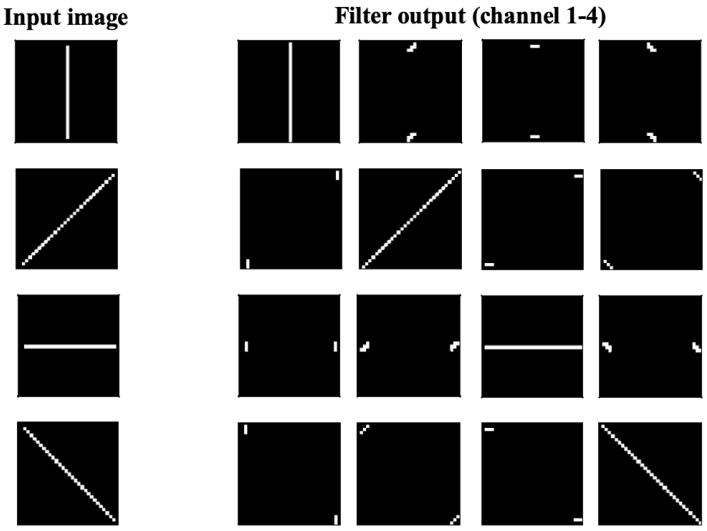


**Figure 6.4:** Output of hand-crafted filters for the straight lines used during training. Each row shows the input image (on the left) and the responses of the four filters (on the right).

The filter response of these four filters applied to the images from the training dataset is shown in Figure 6.4. Please note that a fixed threshold of 0.5 instead of a Bernoulli neuron is used to make this figure appear visually less noisy. For each type of line, one filter specialising in this line has a strong response, almost reassembling the input image (except extending the line by a few pixels). The other filters primarily activate around the endpoints of the lines.

## 6.3 Feature Extracting Stage *S1*

The feature-extracting stage further processes the sensory signal of shape $[4 \times 32 \times 32]$ to build net fragments. In the conducted experiments, no alternative cells are used. As described in Section 5.5, the input of *S1* consists not only of the signal from the sensory system but also of the previous net fragments that can be overridden by the feedback from the

object prototypes. These inputs are stacked along the first dimension, resulting in an input matrix of shape $[8 \times 32 \times 32]$ and an output of shape $[4 \times 32 \times 32]$. Thereby, the first four channels (i.e. the input with index $[(1, ..., 4) \times H \times W]$) are the output of the sensory system, and the last four channels (i.e. input with index $[(5, ..., 8) \times H \times W]$) are the previous net fragments. The previous net fragments are overridden by the feedback of $S2$ if the Jaccard similarity between the net fragments and the $S2$ feedback is above a threshold value of 0.85, and thus, the feedback is considered credible.

The lateral support distance of a single cell is defined as $n_l = 5$. Consequently, a cell can get lateral support by all cells not further away than 5 cells in each direction or from $2n_l + 1 = 11$ cells per input channel. The lateral support is implemented with a convolutional kernel $\boldsymbol{W}$ of size $[4 \times 8 \times 11 \times 11]$, that maps the 8 input channels to 4 output channels. The kernel is initialised as described in Section 5.5.3 and updated with Hebbian updates as described in Section 5.5.2. However, implementing Hebbian updates for a convolutional kernel is challenging, as in an efficient implementation, the kernel is not shifted over the image, but a circulant matrix is used, which cannot provide information about which cells connected through a synapse were active simultaneously [234]. This is solved by using two convolutional kernels as described in Section 6.3.1. However, the principle remains the same, and this two-step procedure is only used for higher computational efficiency.

[234]: Miconi (2021), *Hebbian learning with gradients*

The convolutional operation is applied between the input and the weight matrix $\boldsymbol{W}$ to obtain the lateral support strength. This support strength is normalised as described in Section 5.5.4 using an upper cell-support limit of $\rho = 1.3 \cdot n_l$. However, at the end of the normalisation procedure, each output channel is divided by its highest value to ensure that the support strength is in the range $(0, ..., 1)$ and the highest activation has an activation probability of 1.

$$x_{c,w,h} := \frac{x_{c,w,h}}{\max_{w' \in \{0,...,W\}, h' \in \{0,...,H\}} x_{c,w',h'}} \tag{6.1}$$

The input is processed over $T = 6$ timesteps, and the Hebbian update is calculated between the input and the median activation during these timesteps to increase training stability. The learning rate is set to 0.1, the mini-batch size is 256, and the model is trained for 10 epochs.

### 6.3.1 Implementation Details

In an efficient implementation of a convolutional layer, a circulant matrix is used so that all kernel updates can be calculated in parallel [234]. However, by applying this operation, information about the cell's lateral influence is lost.

This problem is solved using two convolutional operations, producing a slight memory overhead but increasing computation drastically compared to shifting kernels in a loop over the image. The first operation is a fixed, binary convolution that restructures each input patch into a single-column vector. This is followed by a $[1 \times 1]$ convolution containing the actual weights. Specifically, the input is passed through a fixed convolution with an input size of $[C_{in} \times (2n_l + 1) \times (2n_l + 1)]$ and

$C_{in}2(2n_l + 1)$ output channels. The weight vector for this convolution is set to 1 for the connections linking input $c_i, w, h$ to output $c_i wh$ (where $c_i$ $w$, and $h$ range from 1 to $C_{in}$, $2n + 1$, and $2n + 1$, respectively), and it is set to 0 everywhere else. This process reorganises the values of each input patch from the original convolution into non-overlapping column vectors, effectively duplicating them. Next, the actual weights of the original convolution can be applied using a simple $[1 \times 1]$ convolution. This can be achieved by performing a tensor product with appropriate broadcasting. Thus, the proposed method allows calculating Hebbian updates [40] while fully leveraging the computational capabilities of current deep learning hardware.

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

## 6.4 Prototype Stage *S2*

The conducted experiments focus on *S1*. However, since *S2* is needed to provide feedback to *S1*, a mockup simulates such a feedback signal. The mockup described in the following is inspired by the brain's memory system [173], located in the prefrontal cortex [235]. In the context of our framework, such a memory system would be located after *S2* and map the reference object to memories. A single memory can be considered a label, i.e., the projection fibres could initiate a mapping to a face while the memory is the person's name.

[173]: Miyashita (1993), 'Inferior Temporal Cortex'

[235]: Tomita, Ohbayashi, Nakahara, *et al.* (1999), 'Top-down signal from prefrontal cortex in executive control of memory retrieval'

Many neurons are active when we see a person for the first time, indicating that it has not yet been stored by neuronal plasticity. However, after a short learning period, our brain is rewired to remember a frequently occurring object with one or a few single cells [218]. The proposed mockup implements this behaviour: It maps a 2D activation pattern to one or a few cells in a self-organising manner and vice versa. This memory mapping is directly applied to *S1*, mapping net fragments to memory cells and returning feedback, thereby ignoring projection fibres. Thus, a feedback signal is provided without implementing *S2*. Therefore, all aspects from *S1* can be evaluated, including incorporating feedback.

[218]: Gross (2002), 'Genealogy of the "Grandmother Cell"'

### 6.4.1 Implementation

The memory mockup should map net fragments in *S1* denoted as $\boldsymbol{a}$ to a hidden memory state $\boldsymbol{h}$ and vice versa. Thereby, $\boldsymbol{h}$ represents the self-organised memory cells. The mapping processes can be described as conditional probabilities $P(\boldsymbol{h}|\boldsymbol{a})$ and $P(\boldsymbol{a}|\boldsymbol{h})$. Restricted Boltzmann machines (RBMs) [151], [236] are generative stochastic networks that can learn such a probability distribution. They use a linear transformation to map from $\boldsymbol{a}$ to $\boldsymbol{h}$ and the inverse linear transformation to map backwards from $\boldsymbol{h}$ to $\boldsymbol{a}$.

[151]: Hinton (2002), 'Training Products of Experts by Minimizing Contrastive Divergence'

[236]: Smolensky (1986), 'Information Processing in Dynamical Systems: Foundations of Harmony Theory'

First, the 3D activation map containing the net fragments is flattened and multiplied with a weight matrix $\boldsymbol{W}_{S2}$ of size $[(C_{out} \cdot W \cdot H) \times Z]$. This results in a one-dimensional binary vector $\boldsymbol{h}$ of length $Z$, whereby $Z$ is a hyperparameter and defines the memory's capacity. In the conducted experiments, the capacity is set to $Z = 16$. For both mappings $P(\boldsymbol{h}|\boldsymbol{a})$ and $P(\boldsymbol{a}|\boldsymbol{h})$, a bias term $\boldsymbol{b}_a$ and $\boldsymbol{b}_h$, and the sigmoid function to squeeze the activations in the range $(0, \ldots, 1)$ are used. The weight $\boldsymbol{W}_{S2}$ can be

interpreted as fully connected projection fibres, mapping the cells in *S1* ($\boldsymbol{a}$) to memory cells ($\boldsymbol{h}$).

$$P(\boldsymbol{h}_j = 1|\boldsymbol{a}) = \text{sigmoid}(\boldsymbol{W}_{S2} \cdot \boldsymbol{a} + \boldsymbol{b}_a) = \frac{1}{1 + e^{\boldsymbol{W}_{S2} \cdot \boldsymbol{a} + \boldsymbol{b}_a}} \tag{6.2}$$

$$P(\boldsymbol{a}_i = 1|\boldsymbol{h}) = \text{sigmoid}(\boldsymbol{W}_{S2}^\top \cdot \boldsymbol{h} + \boldsymbol{b}_h) = \frac{1}{1 + e^{\boldsymbol{W}_{S2}^\top \cdot \boldsymbol{h} + \boldsymbol{b}_h}} \tag{6.3}$$

Similar to *S1*, the activity of these neurons is used as probability, and the binary output is sampled from a Bernoulli distribution.

$$\boldsymbol{h}_{out} \sim \text{Bernoulli}(P(\boldsymbol{h}|\boldsymbol{a})) \tag{6.4}$$

$$\boldsymbol{a}_{out} \sim \text{Bernoulli}(P(\boldsymbol{a}|\boldsymbol{h})) \tag{6.5}$$

The parameters $\boldsymbol{W}_{S2}$, $\boldsymbol{b}_a$, and $\boldsymbol{b}_h$ are updated by minimising the difference of the free energy function $F(\cdot)$ [151] between $\boldsymbol{a}_{in}$ and $\boldsymbol{a}_{out}$ (i.e. $F(\boldsymbol{a}_{in}) - F(\boldsymbol{a}_{out})$). Recent findings suggest that the brain implements algorithms similar to minimising energy functions as well [237]. Since no backpropagation of error is used, this mapping seems biologically plausible as the rest of the framework. For more details about the implementation, interested readers are referred to the publication by Hinton [151] or a well-written blog post by Hui [238]. The free energy function is minimised using the Adam [72] optimiser with a learning rate of 0.05, and the parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \cdot 10^{-8}$. During training, the learning rate is reduced by a factor of 0.1 if the free energy does not reduce further for more than 2 epochs until a final learning rate of $1 \cdot 10^{-6}$ is reached. The batch size is set to 256, and the model is trained for 10 epochs (similar to *S1*).

By minimising the free energy function, the memory decides by itself which representation should be stored. After observing cell activity $\boldsymbol{a}$ in *S1*, $\boldsymbol{a}$ is mapped to $\boldsymbol{a}_{out}$, whereby $P(\boldsymbol{h}|\boldsymbol{a})$ defines the probability for the cell activity in the memory. Since we sample $\boldsymbol{h}_{out} \sim \text{Bernoulli}(P(\boldsymbol{h}|\boldsymbol{a}))$, $\boldsymbol{h}_{out}$ becomes binary and cells with a low probability tend to be turned off, while cells with a high probability tend to fire. This can be interpreted as a filter against noise and slight deformations: A cell activity $\boldsymbol{a}$ is mapped to the closest known configuration of $\boldsymbol{h}$. To provide feedback to *S1*, the returned vector $\boldsymbol{a}_{out}$ is calculated in a similar fashion and sampled from $P(\boldsymbol{a}|\boldsymbol{h})$. Thus, the memory is a probabilistic associative memory, mapping an input to a hidden state from which an output is sampled. This sampled output is not a reconstructed input but rather an optimised prototype stored in the memory and used to provide feedback to *S1*.

[151]: Hinton (2002), 'Training Products of Experts by Minimizing Contrastive Divergence'

[237]: Isomura, Kotani, Jimbo, *et al.* (2023), 'Experimental validation of the free-energy principle with in vitro neural networks'

[238]: Hui (2017), *Machine learning - Restricted Boltzmann Machines*

[72]: Kingma and Ba (2017), *Adam: A Method for Stochastic Optimization*

# Results 7



This chapter presents the results of the conducted experiments. First, an overview of the results achieved with the entire system is provided. Subsequently, individual components of the system are examined in more detail. The code and documentation are made publicly available. Further information can be found in the appendix Chapter A.

## 7.1 Entire System

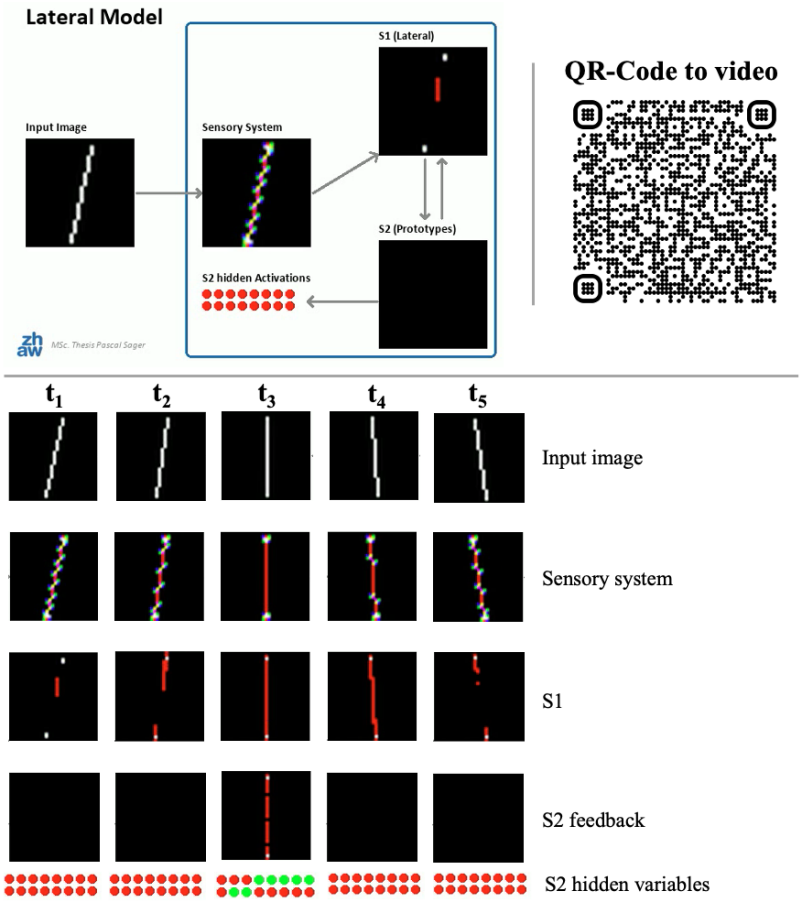


**Figure 7.1:** Frames of a video visualising the model's activations. At the top of the image, an actual video frame and a QR code linking to the video are shown. At the bottom of the image, screenshots of the video are shown, depicting the changing network activations over time.

An overview of the entire system is provided in Figure 7.1. This image is derived from a video available online at sagerpascal.github.io/lateral-connections or with the QR code on the top right of the figure. For a detailed explanation of the components shown in the video, please refer to appendix Section A.1. The video shows the network's activations for a straight line rotated counterclockwise around its centre. In the lower part of Figure 7.1, a series of five network states is shown shortly before and after a vertical line is reached.

The network has only been trained on vertical, horizontal, and diagonal lines. Therefore, many lines fed into the network in this video represent unknown objects[1]. Nevertheless, *S1* still detects local patterns in most images, such as multiple pixels aligned vertically, horizontally, or diagonally. Therefore, it can provide lateral support between local pixel groups representing such a local pattern. The closer the input becomes to a learned pattern, the bigger the lateral support. For instance, at time $t_3$, the input corresponds to a vertical line as observed during training. In that case, all pixels receive enough lateral support to remain active. Thus, lines observed during training get lateral support at all positions and are captured in net fragments. In contrast, lines not observed during training have local features corresponding to learned patterns and are supported by smaller local net fragments.

1: With projection fibres, such transformed objects instances should be mapped to the same prototype. However, the memory used as a mockup cannot deal with such transformations.

For the conducted experiments, *S2* is simulated by a self-organising memory, storing patterns observed during training. As long as the net fragments represent an unknown pattern, no latent cells in *S2* are activated and no feedback to *S1* is provided. However, when the net fragments in *S1* correspond to a learned pattern, *S2* provides feedback and further increases certainty in *S1*. When replacing the memory used to simulate *S2* with projection fibres, the feedback should be even better, as *S2* will be able to detect transformed objects.

The network's behaviour is as expected: *S1* builds net fragments based on well-known patterns observed during training. All input features observed during training receive full lateral support. Moreover, images not seen during training also contain local patterns similar to those from the training data and, therefore, still receive local support. *S2* responds to patterns stored in its memory, only providing support to *S1* for objects seen during training. Furthermore, all samples seen during training have automatically been saved in *S2*. The system is able to produce net fragments that roughly reassemble the input from the sensory system. However, the net fragments are more robust than the sensory system, as shown in the following sections.

### 7.1.1 Effect of Noise

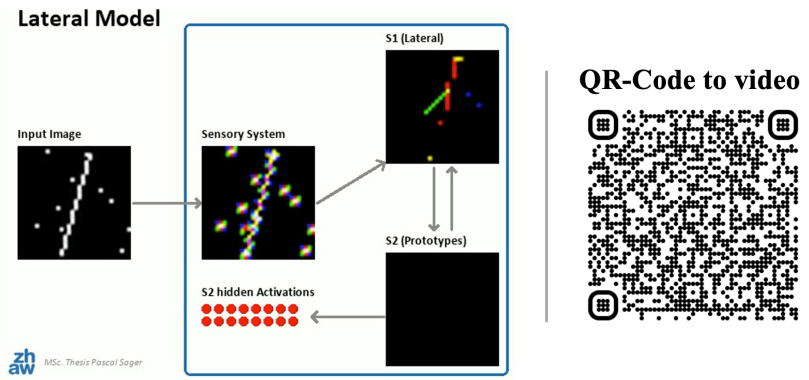


**Figure 7.2:** A frame of a video visualising the network's behaviour if noise is added to the data. The QR code on the right links to the corresponding video.

Figure 7.2 refers to a video demonstrating the network's behaviour when noise is added to the input data. Additionally, the effect of adding noise to the four training images is visualised in Figure 7.3. The noise is generated

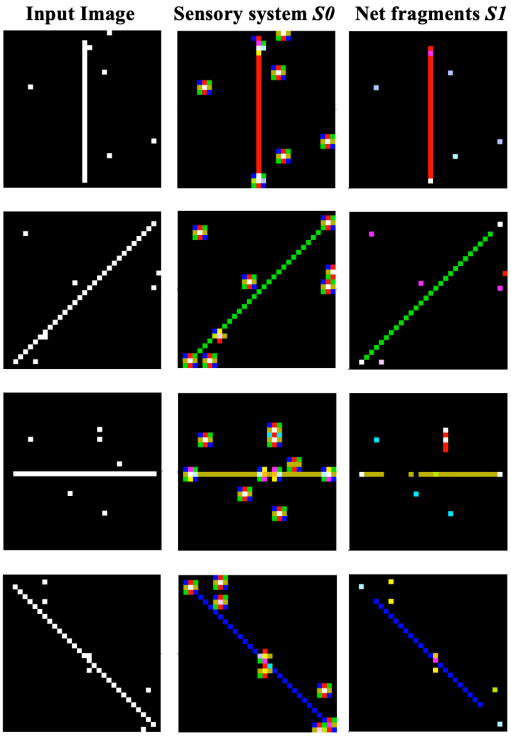


**Figure 7.3:** Effect of adding noise to the four images seen during training.

by randomly flipping a pixel in the input data from 0 to 1 or vice versa with a probability of 0.5%.

To assess the network's ability to deal with noise, the same input is fed into the model twice: once with and once without noise. The activations of *S1* for these two versions of the input image are compared, and the percentage of feature cells that are initially triggered by the introduced noise but subsequently deactivated due to insufficient support is measured.

This analysis shows that the system can remove about 68.2% of the noise from the input data. However, this effectiveness is mainly because a single noise pixel triggers 3 cells in each feature channel of the sensory system, resulting in 12 active cells. After building net fragments, only the cells at the centre of the four feature channels remain active, giving the impression that the noise has been removed while, in reality, it only has been reduced.

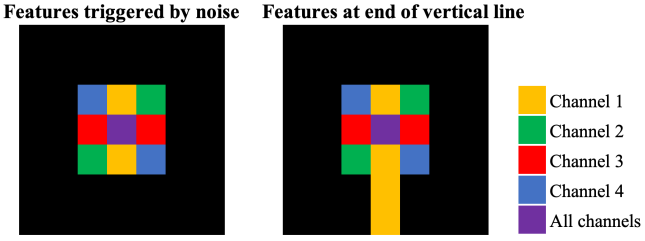


**Figure 7.4:** The features triggered by noise (on the left) compared with the features triggered by a line end (on the right). The network input is shown on the left, the output of the sensory system in the middle, and the net fragments built by *S1* on the right. The different colours indicate different feature channels.

In fact, only in 5.8% of the cases is noise in the input completely removed. Several reasons contribute to the difficulty of obliterating noise: First, noise can be located close to the line or other noise and thus receives lateral support from other cells. Second, when observing noise in the input, the sensory system triggers activations in all feature channels similar to activations found at line ends as visualised in Figure 7.4. As a

result, activations triggered by noise are very similar to learned patterns. Therefore, these cells support each other and cannot be adequately filtered by the system. Overall, this behaviour is to be expected, and noise should only be filtered out if it is significantly different from learned patterns.

Although the noise cannot be completely filtered out, the net fragments are still accurately mapped to the correct prototype in *S2*. Thus, the system still correctly interprets the input despite the noise.

**Noise per Channel**

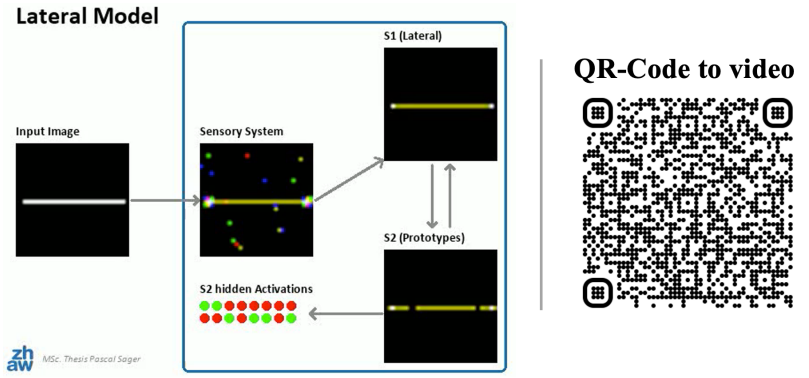


**Figure 7.5:** A frame of a video visualising the network's behaviour if noise is added to the feature channels. The QR code on the right links to the corresponding video.

In this section, it is investigated whether noise can be filtered when there is no correlation between the locations of the noise within the feature channels. Thus, the noise does not correspond to learned patterns anymore. Therefore, noise is not added to the input data but to each feature channel of the sensory systems' output separately. This experiment is considered more relevant for real-world scenarios, as future systems that deal with real-world data will have a much larger number of input channels and more diverse patterns, making it unlikely that noise resembles a learned pattern that the network considers valid.

Figure 7.5 refers to a video demonstrating the networks' behaviour when noise is added to each feature cell with a probability of 0.5%. Additionally, the effect of adding noise to the feature channels is visualised in Figure 7.6 for the four training images. As can be observed in this figure, net fragments can remove noise very well when it does not correspond to learned patterns. In fact, about 91.7% of the noise is removed, demonstrating the network's high robustness to such perturbations. The noise that is not removed is the noise that is very close to the actual line and therefore receives support from a valid object.

### 7.1.2 Discontinuous Line

Net fragments [20], [169] can not only reduce noise but also recreate occluded objects, as inactive cells receiving enough lateral support from their neighbours can turn on. This phenomenon is demonstrated by analysing the network's behaviour if a discontinuous line is fed into the network. In Figure 7.7, a QR code is contained linking to a video demonstrating that *S1* is able to reconstruct lines that are interrupted by up to 8 pixels.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

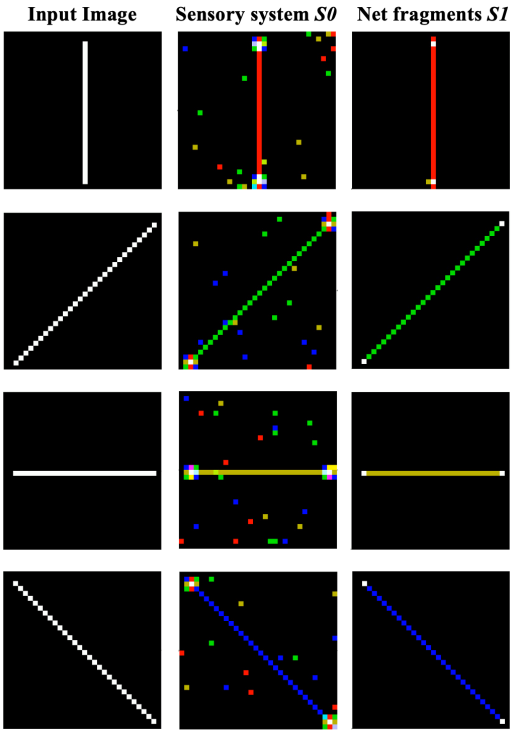


**Figure 7.6:** Effect of adding noise to the four feature channels (i.e. the output of the sensory system) of the images seen during training.The network input is shown on the left, the output of the sensory system with added noise in the middle, and the net fragments built by *S1* on the right. The different colours indicate different feature channels.

In the experimental setup, the centre of the line is detected, and a varying number of pixels starting from the centre are intentionally switched off. The feedback from *S2* is switched off so that only the reconstruction based on net fragments within *S1* is tested. Remarkably, the *S1* consistently succeeds in reconstructing the original training input when up to 6 pixels are removed. Furthermore, in many cases, it can reconstruct lines with up to 8 pixels missing, although it fails with more than 8 pixels removed.

The extent to which the network can reconstruct discontinuous lines depends on the range of lateral connections $n_l$. As expected, increasing the value of $n_l$ allows *S1* to reconstruct lines with more missing pixels, improving its performance in recovering occluded or destroyed objects. However, $n_l$ should not be too large so that *S1* build net fragments based on local features (c.f. Section 4.1.3).

In the conducted experiments, the lateral range is set to $n_l = 11$. The impressive ability to reconstruct up to 8 missing pixels with this setting suggests that recreating occluded patterns works effectively.

### 7.1.3 S2 Feedback

In the following, feedback from *S2* is additionally incorporated in *S1*, and it is analysed if *S1* can efficiently utilise it to reconstruct lines.

In Figure 7.8, the activation probabilities for all cells across the four output channels for a vertical line with 20 missing pixels are visualised. The top row shows the probabilities when no feedback form *S2* is incorporated, and the bottom row the probabilities after the feedback is incorporated.

As visible in the top row of Figure 7.8, without incorporating feedback from *S2*, the line cannot be fully reconstructed using lateral connections

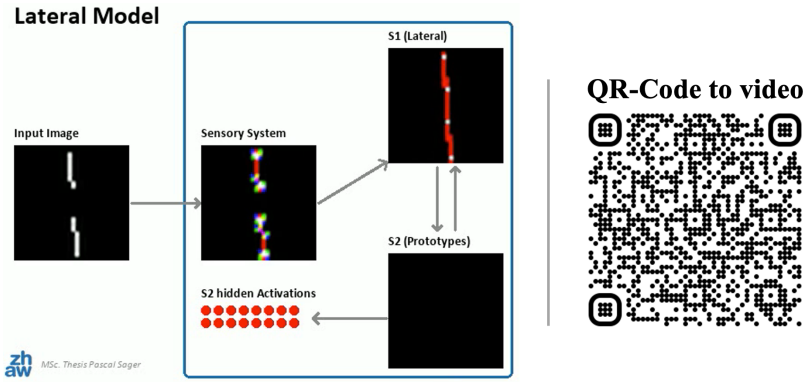


**Figure 7.7:** A frame of a video visualising the network's behaviour for discontinuous lines. The QR code on the right links to the corresponding video.

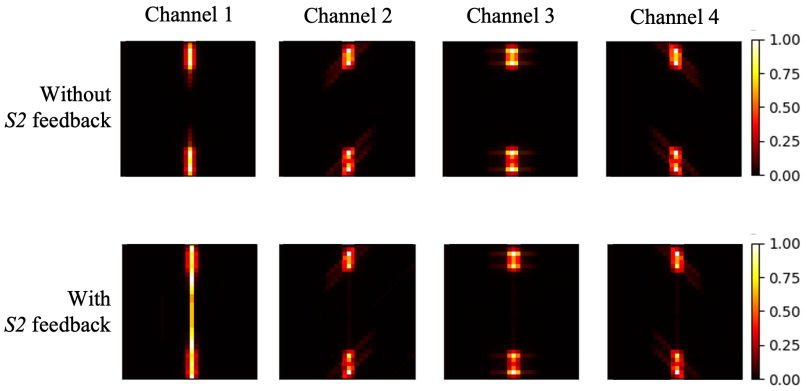


**Figure 7.8:** The activation probabilities per cell across all four channels with and without feedback from *S2*. The input is a discontinuous vertical line with 20 pixels missing.

in *S1*, as 20 of missing pixels exceeds the range of lateral connections. However, *S2* is still able to map the line with missing pixels to the correct prototype and provide appropriate feedback. After the feedback is incorporated, the activation probabilities for the entire vertical line increase significantly. Especially in the middle section of the first channel, the activation probabilities increase from 0% to approximately 65%.

When a continuous vertical line is fed into the system, the activation probability in the middle section of the line is above 90%. This high probability aligns with the fact that the sensory signal and the memory's feedback are consistent. However, when the memory expects activations that are not detected by the sensory system (e.g. due to the missing pixels), the activation probability decreases. This behaviour reflects meaningful modelling of the network's uncertainty when integrating feedback from *S2* while encountering occluded objects.

In conclusion, the feedback from *S2* can be effectively incorporated into *S1*. It helps to deal with occluded objects and creates stability in net fragments, even when the sensory system does not detect (occluded) parts of the object.

## 7.2 Model Weights

The weight matrix of *S1*, which contains the strength of the learned lateral connections, is discussed in the following. In Figure 7.9, the learned weights are visualised. The input of *S1* consists of four channels from the sensory stage (rows labelled as 1-4) and four channels from recurrent connections (labelled as 5-8). *S1* produces four output channels,

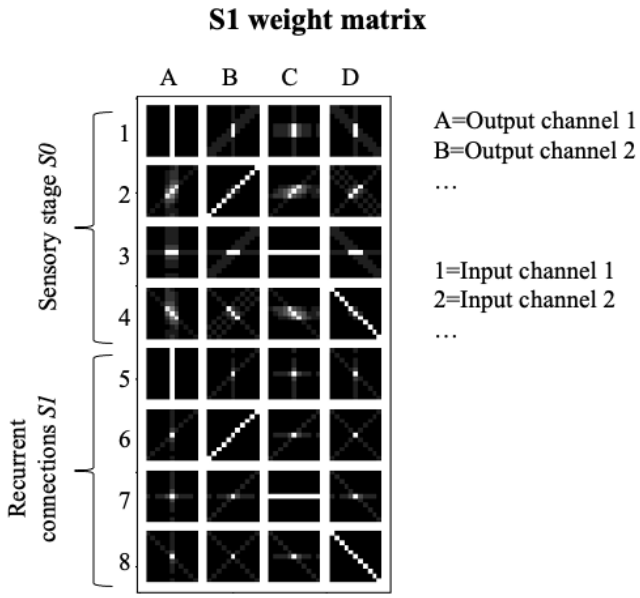


**Figure 7.9:** The weight matrix of *S1* after training.

and the kernels contributing to each output channel are depicted in columns labelled from *A* to *D*. Each output channel specialises in a different type of line: Output channel *A* focuses on vertical lines, channel *B* on diagonal lines with a positive slope, channel *C* on horizontal lines, and channel *D* on diagonal lines with a negative slope.

An analysis of output channel *A*, which focuses on horizontal lines, is presented in the following. However, it is important to note that the four output channels have similar characteristics, with the main difference being that the filters are rotated by 45°. Consequently, insights from channel *A* are also transferable to all other channels.

In Figure 7.10, the features processed when a horizontal line is fed into the system are visualised. First, the sensory system extracts four features from the input (visualised in the box named "output sensory system"). Channel 1 contains "vertical-line features", spanning the entire vertical length of the image. The channels 2-4 contain features of diagonal and horizontal lines. However, the sensory system recognises these features only at the ends of the lines. Thus, at the ends of the vertical line, about three neurons respond for each channel 2-4 to represent these features. These features extracted by the sensory system are fed into the channels 1-4 of *S1*. As expected, these features have been incorporated into the weight matrix accordingly (see *A*1-*A*4).

Based on these features, output channel *A* generates a response roughly corresponding to the vertical line initially fed into the system. Thus, channel *A* fulfils its purpose and represents vertical lines. Besides channel *A*, also the channels *B*-*D* become active. However, these channels specialise in different lines and only activate exactly one pixel at the line ends, where the sensory system produces a very high activity across all channels.

The output of *S1* is reused as an input signal in the next timestep $t+1$. This is implemented as a recurrent connection between the output channels *A*-*D* and the input channels 5-8. As expected, the filters processing the recurrent input used for output channel *A* specialise in the activity that

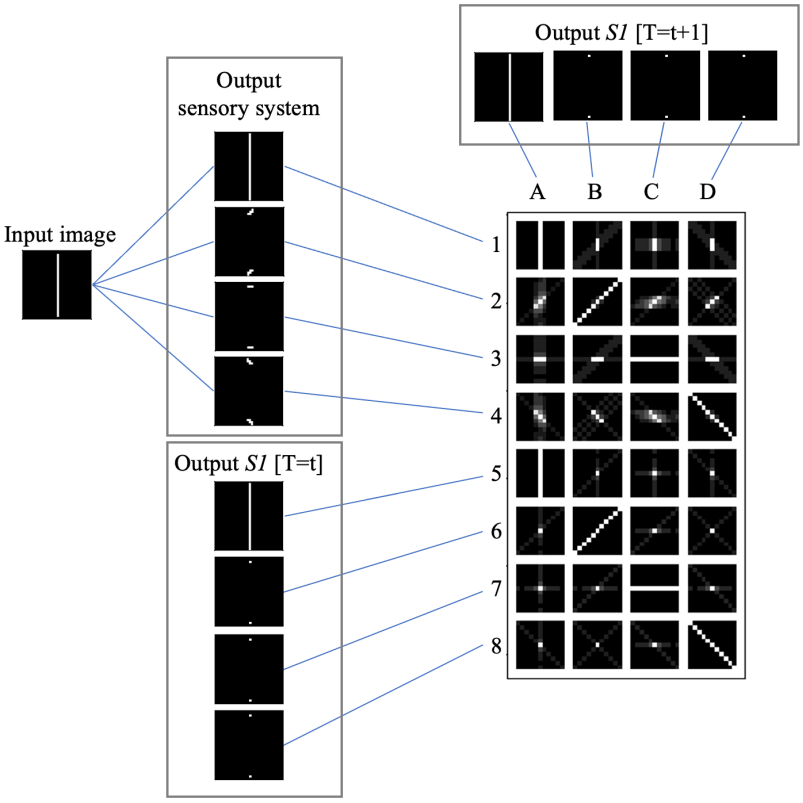


**Figure 7.10:** An overview of the data processed by the weight matrix of *S1*.

is produced by *S1* for vertical lines: When a vertical line is processed, the output channel *A* outputs a vertical line and the channels *B*-*D* a single-cell activation, corresponding to the filters *A*5-*A*8.

### 7.2.1 Weight Normalisation

As described in Section 5.5.4, the weight is normalised in the range $(0, \ldots, 1)$. Normalising the weights is crucial for the proper functioning of the network. Without weight normalisation, lateral support could be dominated by a single cell, resulting in infinite lateral support if trained long enough. In the human brain, there is no such dominance of single cells, and neighbouring cells play an equally important role in providing support [224].

[224]: Kandel (2013), *Principles of neural science*

After 10 epochs without weight normalisation, some lateral connections reach a weight above 74 and dominate the decision process of whether neighbouring cells should remain active. This leads to undesired activations and weight updates. In Figure 7.11, the weight matrix after training for 10 epochs without weight normalisation is depicted. No clear structure is visible within the weights, and the support provided within the network appears somewhat random. Thus, normalisation is not only biologically more plausible but also a necessity to obtain meaningful lateral weights.

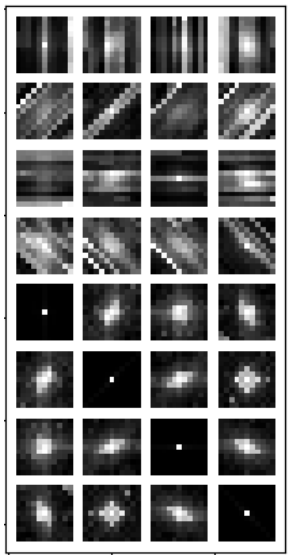

**Figure 7.11:** Weight matrix of *S1* after training without weight normalisation.

### 7.2.2 Initialisation

In this section, the crucial aspect of weight initialisation is discussed. In Section 5.5.3, it is described that initialising the weight with self-support is essential for the proper functioning of the network. Two different approaches exist to initialise weights with self-support, as shown in Figure 7.12. Regardless of the strategy chosen, both approaches lead to identical weight matrices after the training process.

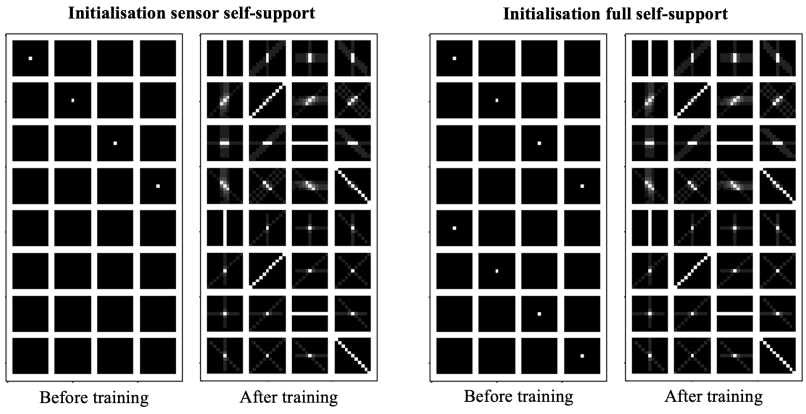


**Figure 7.12:** Two different ways of initialising the weights of *S1* with self support. For both initialisation strategies, the initial weights are shown on the left and the weights after training on the right.

However, it is important to note that not all weight initialisation strategies lead to good results. In Figure 7.13, two other strategies and the resulting weight matrix after training are shown: Initialising the weights randomly leads to support between cells that should not support each other. Consequently, this results in unwanted network activations, and the network converges towards weight parameters where the weights are almost identical across all output channels and do not provide lateral support in the desired manner. If, on the other hand, the weights are initialised with zeros, the network has no active outputs. Consequently, all cells are immediately deactivated, and the weights remain unchanged during Hebbian learning.

In conclusion, using one of the two self-support weight initialisation strategies shown in Figure 7.12 is crucial. These methods ensure proper

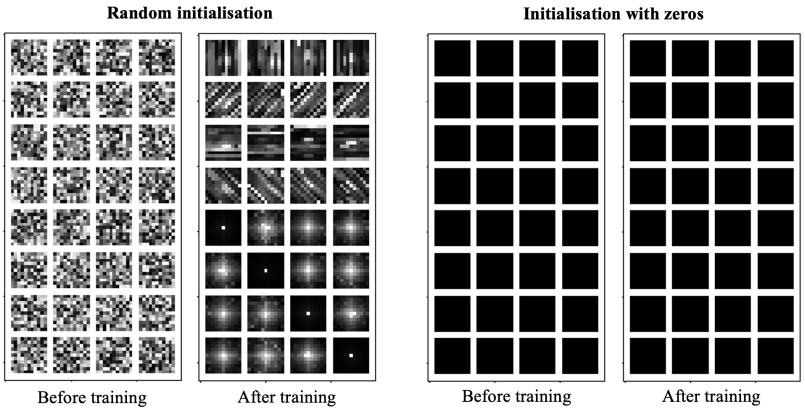


**Figure 7.13:** Two different ways of initialising the weights of *S1*. The random weight initialisation strategy is shown on the left side of the image, and the zero initialisation strategy is shown on the right side. For both initialisation strategies, the initial weights are shown on the left and the weights after training on the right.

functioning and effective learning, unlike the random or zero weight initialisation strategies depicted in Figure 7.13.

## 7.3 Support Quality

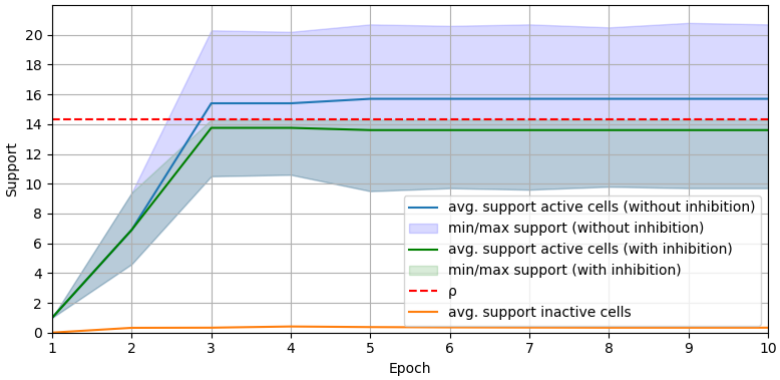


**Figure 7.14:** The average lateral support received by active and inactive cells during training. The y-axis shows the support, and the x-axis the training epoch. The blue line is the average, and the light blue interval is the min./max. support an active cell receives during training when no inhibition is used; the green interval represents the average/min./max. support an active cell receives support when inhibition is used; the orange line is the average support inactive cells receive. The dotted red line is the inhibition limit $\rho$, marking the threshold where the support is reduced.

In Figure 7.14, the support strength received by active and inactive cells during training is presented before the support strength is normalised and translated into an activation probability. The green interval depicts the support with inhibition and the blue interval without inhibition.

Before training, only self-support exists, i.e. the received support for active cells is 1. After training for 3 epochs, the average support active cells receive increases to 13.8 with inhibition and 15.7 without inhibition. At this point in training, most lateral connections have converged to a synaptic weight strength of 1. Thus, on average, a single cell is supported by approximately 14 neighbouring cells (with inhibition limit) or 16 cells (without inhibition limit). On the other hand, inactive cells receive, on average, a lateral support of 0.3, significantly less than active cells. Thus, active cells receive much more lateral support after training, while inactive cells still do not receive significant support. The support difference between active and inactive cells increases from 1 at the beginning of training to 13.5 after training when inhibition is used. This increase in support implies that it becomes much more difficult for individual cells to become active and explains why noise can be filtered efficiently.

The support active cells receive is limited by the inhibitory strength $\rho = 1.3 \cdot n_l = 14.3$. When a cell exceeds this threshold $\rho$, its activation

probability is linearly reduced. This effect is visible when comparing the blue interval (without inhibition) with the green interval (with inhibition). With inhibition, the support strength for each cell is pushed below $\rho$, depicted as the red dotted line.

In Figure 7.14, it is shown that cells can receive lateral support strengths of up to 21 when no inhibitory signals are present in the network (see the max. values of the blue interval). However, such strong support leads to undesired effects, as discussed in the next section.

The results depicted in Figure 7.14 demonstrate that *S1* builds net fragments as expected. The gap between the support strength of active and inactive cells becomes bigger during training, ensuring that only cells representing known patterns remain active.

### 7.3.1 Inhibition

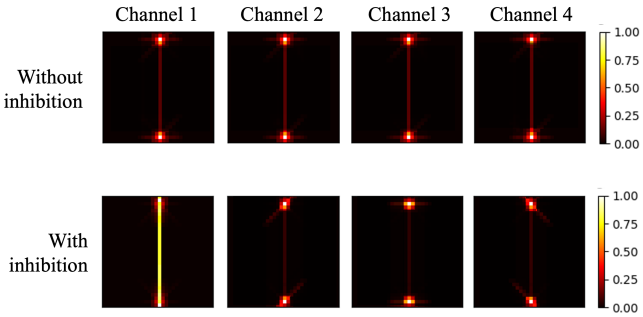


**Figure 7.15:** The activation heatmap indicates the activation probability for all cells across the four channels. The upper row shows the probability without inhibition and the lower row with inhibition.

The impact of the inhibition threshold $\rho$ is discussed in the following. The activation heatmap shown in Figure 7.15 displays the activation probabilities for cells across four channels when a vertical line is fed into a model. The top row shows the probabilities of a model trained without inhibition, while the bottom row shows the probabilities of a model trained with inhibition.

As discussed, all filters are active at the line ends, leading to many active cells and high lateral support in that region. Specifically, the cell located at the line end receives lateral support from up to 21 neighbouring cells if no inhibition is used. When an activation strength of 21 is mapped to an activation probability of 1.0 and an activation strength of 0 to 0.0, the cells receive activation probabilities as visualised in the upper row of Figure 7.15. The pixels at the line ends are dominant and have an activation probability of 1.0, while the other pixels on the line have an activation probability of approximately 0.37. Therefore, only the cells at the line ends have a high activation probability but not the other cells depicting the middle of the line.

Introducing inhibition effectively addresses this issue. With inhibition, the lateral support is limited to $\rho$. The effect of inhibition on the activation probability per cell is shown in the second row of Figure 7.15. Notably, the activation probabilities of cells at the line ends remain high. However, the activation probabilities of the other pixels depicting the vertical line increase significantly, especially in the first channel, which represents vertical line features. Thus, inhibition introduces an upper activity threshold so that regions with many active features are limited

in providing mutual support and are not dominant compared to other regions.

Please note that the two heatmaps in Figure 7.15 stem from different models. Inhibition strongly influences the training process, and "turning on" inhibition would not convert the activation probabilities in the first into the ones shown in the second row. Instead, inhibition normalises the activation probabilities throughout the training process, influencing weight updates. Without inhibition, the activations are dominated by line ends, causing all channels to learn similar features. With inhibition, the channels specialise more on distinct features as no feature dominates the learning process. Therefore, the weights (and thus the activation probabilities shown on the second row of Figure 7.15) are more diverse.

## 7.4 Conclusion

The previous sections discuss the obtained results, thereby focusing on specific aspects. In conclusion, *S1* can build net fragments [20], [169], and it is demonstrated that these fragments associate input patterns with learned patterns, thereby removing noise or reconstructing occluded parts of objects. Thus, the experiments demonstrate that net fragments can be implemented according to the proposed principles. Since removing noise and reconstructing objects improves representations over multiple timesteps, it is considered a hierarchical processing of features without being subject to early commitment [26].

This is considered an essential step towards implementing the proposed framework. Implementing projection fibres [175], [203] is based on the principle of comparing local features in *S1* and *S2* and initiating a mapping between them if neighbouring fibres agree [29]. Such a mapping has already been implemented [27], [28], [30]–[32] but only works well if patterns in the two stages are highly similar. As demonstrated in this chapter, making local patterns more similar can be achieved with net fragments. Net fragments either turn off cells representing unknown patterns that are not present in *S2* and thus cannot be mapped, or they support and reconstruct (i.e. optimise) existing patterns to become more similar to patterns stored in *S2*. Thus, the conducted experiments lay the foundation for future research and implementing projection fibres.

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

[29]: Wiskott and von der Malsburg (1996), *Face Recognition by Dynamic Link Matching*

[27]: Bienenstock and von der Malsburg (1987), 'A Neural Network for Invariant Pattern Recognition'

[28]: Lades, Vorbruggen, Buhmann, *et al.* (1993), 'Distortion invariant object recognition in the dynamic link architecture'

[30]: Wiskott, Fellous, Kuiger, *et al.* (1997), 'Face recognition by elastic bunch graph matching'

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

# 8 Future Work & Conclusion

## 8.1 Discussion



In Section 1.1, it is discussed that the human brain can recognise the "Gestalt" (the entire structure) of an object within a very short time [22]–[25] because it prevents early commitment [26]. This capability is absent in current deep learning systems. Since the biological system implements this behaviour, important neuroscientific findings responsible for the effectiveness of the biological vision system are identified in Section 4.1. It is described that the human brain uses lateral connections [138], [201], [202] to build net fragments [20], [169] that are mapped to reference frames using projection fibres [175], [203]. Such fibres allow object-independent mapping [31], [32], making the system highly efficient.

Based on these findings, a novel computational framework is proposed. It encompasses three building blocks, all using novel binary neurons called Bernoulli neurons. These biologically inspired neurons allow to implement net fragments that improve robustness. The sensory stage *S0* corresponds in the biological context to the eyes and extracts features from images. The feature building stage *S1*, inspired by the primary visual cortex [171], [172], leverages lateral connections to form net fragments, groups of neurons that support each other's activity. This stage is well examined in this thesis and iteratively refined based on empirical investigations. The experiments confirm the usefulness of lateral connections in tasks such as occluded object reconstruction and noise reduction. The prototype stage *S2* takes inspiration from the ventral visual stream [199] and the temporal cortex [173], [174]. It uses projection fibres to map network fragments onto object prototypes.

At its core, the entire network is based on self-organisation, locality and cell consistency principles. Net fragments arise from cells communicating with their spatial neighbours, while projection fibres connect neighbouring cells in *S1* and *S2* and seek consistency with neighbouring fibres. The iterative process of generating net fragments and mapping them to object prototypes leads to efficient transformation-invariant feature processing independent of specific objects.

A significant difference between the proposed system and deep networks lies in the mechanism of building consistency: Deep networks optimise consistency at a single point in the network by comparing its prediction with a teaching signal. A global error correction algorithm such as backpropagation adjusts all components in the network to minimise inconsistencies at this specific point. In contrast, the proposed framework implements a model that optimises consistency between each neuron, akin to the human brain. Furthermore, consistency is built by the network itself in a self-organising manner without requiring an external teaching signal. Propagating the error layer-wise backwards makes the learning algorithm biologically implausible [43], [44] and leads to early

[138]: Gilbert, Hirsch, and Wiesel (1990), 'Lateral Interactions in Visual Cortex'

[201]: Liang, Gong, Chen, *et al.* (2017), 'Interactions between feedback and lateral connections in the primary visual cortex'

[202]: Stettler, Das, Bennett, *et al.* (2002), 'Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex'

[20]: von der Malsburg, Stadelmann, and Grewe (2022), *A Theory of Natural Intelligence*

[169]: von der Malsburg (2018), 'Concerning the Neuronal Code'

[175]: Greig, Woodworth, Galazo, *et al.* (2013), 'Molecular logic of neocortical projection neuron specification, development and diversity'

[203]: Tanigawa, Wang, and Fujita (2005), 'Organization of Horizontal Axons in the Inferior Temporal Cortex and Primary Visual Cortex of the Macaque Monkey'

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

[43]: Grossberg (1987), 'Competitive Learning'

[44]: Crick (1989), 'The recent excitement about neural networks'

commitment [26] (c.f. Section 1.1). Furthermore, such networks have intrinsic limitations [9]–[13], [17], [46], [85], [90]. Nevertheless, training all neurons in a way to optimise consistency at a specific point works exceptionally well for very specific tasks and can outperform human experts [75]. Therefore, such systems probably also outperform models implementing a brain-like algorithm on such specialised tasks.

[26]: Marr (2010), *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*

[75]: Buetti-Dinh, Galli, Bellenberg, *et al.* (2019), 'Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition'

However, building a system that optimises consistency between each connected cell pair has a different set of advantages. Its self-organising nature allows to prevent early commitment since it does not build feature processing chains (c.f. Section 1.1). Properly implemented projection fibres can map net fragments to object prototypes even when transformed. Projection fibres allow the transfer of knowledge [31], [32] much more efficiently between objects, could increase computational efficiency and allow better extrapolation of the data distribution. Furthermore, prototypes are stored in *S2* in separate reference frames, thereby reducing the risk of catastrophic forgetting [12] as newly acquired knowledge cannot overwrite previously learned knowledge. As outlined in the vision presented in Section 4.2, the proposed framework could interpret entire visual scenes meaningfully without requiring a teaching signal. For instance, each cell in the network represents a specific part of a visual scene and can predict the activity of neighbouring cells. Thus, each cell in the network contributes directly to coherent representations in the decision-making processes. Consequently, each signal and cell votes for a particular course of action and seek consensus without an external source providing a global teaching signal. I argue that this is a different way of thinking about artificial learning and potentially opens up new paths for how intelligent systems could be trained.

[31]: Wolfrum, Wolff, Lücke, *et al.* (2008), 'A recurrent dynamic model for correspondence-based face recognition'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

[12]: Kirkpatrick, Pascanu, Rabinowitz, *et al.* (2017), 'Overcoming catastrophic forgetting in neural networks'

## 8.2 Future Work

In this thesis, a novel vision framework is proposed. However, numerous avenues exist for future work to further improve the framework. These improvements can focus on various dimensions, such as

- extending the theoretical foundations of the framework with a memory layer or scene interpretation according to the vision outlined in Section 4.2,
- refining and confirming the existing theoretical foundations by conducting further experiments,
- scaling the framework to different datasets,
- introducing multi-modality similar to the human brain,
- and identifying metrics to evaluate the system better.

In the following section, these improvements are discussed, and concrete next steps are suggested.

### 8.2.1 Extending Theory

The current theoretical foundations are limited to the stages *S0*, *S1*, and *S2*. While *S0* and *S1* are well developed, *S2* should be further refined by conducting experiments based on existing work by [32] and avoiding the presumed simplifications (c.f. Section 5.6). In particular, novel prototypes from unseen objects should be stored automatically, the prototypes should be iteratively improved throughout training, projection fibres should be learned dynamically, and the network should be extended beyond the limits of object-centric images. With these extensions, the framework should be able to map various objects to prototypes and perform well on different datasets.

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

Furthermore, two building blocks are missing compared to the vision described in Section 4.2. First, an additional memory stage *S3*, storing specific instances of objects, should be added. While projection fibres should map the perceived objects to reference representations, this stage could assign labels to the objects and distinguish different object instances. A simple version of a memory is proposed in the conducted experiments as a mockup of *S2* (c.f. Section 6.4.1). However, it has not yet been combined with projection fibres. Second, a scene interpretation stage needs to be included. While the projection fibres answer the question of "what?" is visible within an image, an additional stage should analyse the relation between objects and allow the interpretation of visual scenes.

### 8.2.2 Refining Theory

The stage *S1* is well developed, and it has been demonstrated in experiments that Bernoulli neurons trained with Hebbian learning can form net fragments. Nevertheless, adding alternative cells and negative Hebbian learning seems crucial to scale to different datasets.

The stage *S2* has only been explored from a theoretical viewpoint within this thesis. Conducting experiments to refine the proposed theoretical foundations and demonstrate their efficiency is important to measure the overall framework performance. These tasks are discussed in the following.

#### Alternative Cells in *S1*

As outlined in Section 4.1.3, cells can contribute to mutually exclusive net fragments. For example, cell $A$ may participate in a fragment with cell $B$ and another fragment with cell $C$, while cell $B$ and $C$ avoid simultaneous activation. This exceeds the functional capacity of cell $A$, and a copy of $A$ is needed to establish separate lateral connections with cell $B$ and $C$.

In Section 5.5.5, it is described that alternative cells could be implemented by duplicating the output channels of the weight matrix $\boldsymbol{W}$ of *S1*. Alternative cells contribute to different net fragments and are mutually exclusive. Consequently, competition between these alternative cells is required to ensure that only a winning cell can become active and that activity in alternative cells is suppressed[1]. Well-known competition strategies

1: Activity can be suppressed by inhibition, as it is already implemented in *S1*.

are winner-take-all competition, providing an external competitive signal, anti-Hebbian learning [102], or adapting the activation function of the neurons to enforce a specific activity distribution [103], [104] (c.f. Section 2.4.1).

[102]: Vogels, Sprekeler, Zenke, *et al.* (2011), 'Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks'

[103]: Joshi and Triesch (2009), 'Rules for information maximization in spiking neurons using intrinsic plasticity'

[104]: Teichmann and Hamker (2015), 'Intrinsic Plasticity: A Simple Mechanism to Stabilize Hebbian Learning in Multilayer Neural Networks'

Furthermore, Hebbian learning [40] must be extended to enable forgetting previously learned patterns. Currently, each update only increases the weights, which strengthens lateral support. However, some updates could inadvertently create incorrect connections between different (alternative) cells. In the following section, negative Hebbian learning is described, which allows forgetting learned connections and seems crucial to implementing alternative cells. This mechanism eliminates the need to carefully prevent false updates during the initial training phase, as erroneous updates can be corrected as soon as the alternative cells are separated enough.

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

### Negative Hebbian Learning within *S1*

While conventional Hebbian learning [40] increases the synaptic weights between simultaneously active neurons, negative Hebbian learning introduces a complementary process by decreasing the synaptic weight between cells that fire disjoint. These negative updates are not only crucial in the formation of alternative cells but also in gradually eliminating less significant patterns that have been imprinted during the training phase.

Implementing negative Hebbian updates is challenging, especially when the data is dominated by negative correlations[2], as shown in appendix Chapter B. One possible strategy to overcome this problem is using a significantly lower learning rate for negative updates than for positive ones. This asymmetry ensures that the process of forgetting is slower than the process of learning, preventing the abrupt erasure of acquired patterns. Another solution is using alternative cells: If two patterns have a positive correlation at one point and a negative correlation at another point (as in the example shown in Chapter B), these patterns can be processed differently by employing alternative cells, effectively maintaining their distinct representations.

2: The dataset contains more samples where two feature cells are active separately than simultaneously.

### Refine *S2*

An important task for future work is the empirical improvement of *S2*, which has only been theoretically developed based on identified neuroscientific findings and existing work. The current blueprint describing its implementation has to be further refined by conducting experiments.

In the first phase, the integration and evaluation of projection fibres based on shifter circuits [207], [208] should be done and explored within the proposed framework. Afterwards, different object views should be explored (provided by the medium processing loop) as done by Fernandes *et al.* [32]. Currently, these views are only used during evaluation, but once *S2* is implemented, they can be crucial in learning to associate different object views to the same prototype, encouraging transformation-invariant mappings.

[207]: Anderson and Essen (1987), 'Shifter circuits'

[208]: Olshausen, Anderson, and Van Essen (1993), 'A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information.'

[32]: Fernandes and von der Malsburg (2015), 'Self-Organization of Control Circuits for Invariant Fiber Projections'

### 8.2.3 Scaling to Different Datasets

In the experiments, a dataset comprising straight lines is used, effectively illustrating the principles and enhancing understanding of the proposed framework. Nevertheless, assessing the models' scalability to larger and more diverse datasets is important. One possible avenue is to use traditional classification datasets such as MNIST [139], CIFAR-10 [142], or ImageNet [239]. However, it is important to note that the primary goal is not to push benchmarks for image classification. Instead, the goal is to obtain high-quality object representations.

This endeavour may require building new datasets generated by an image rendering engine capable of simulating 3D objects and generating data in real-time. Using such an engine allows to generate visualisation of objects undergoing realistic-looking transformations and depth rotations. This method allows the evaluation of the model's ability to process complex and diverse visual data that more closely resemble real-world scenarios. Moreover, these transformations are an integral part of the proposed processing loop and even allow interaction with objects.

[139]: LeCun, Bottou, Bengio, *et al.* (1998), 'Gradient-based learning applied to document recognition'

[142]: Krizhevsky (2009), 'Learning Multiple Layers of Features from Tiny Images'

[239]: Russakovsky, Deng, Su, *et al.* (2015), 'ImageNet Large Scale Visual Recognition Challenge'

### 8.2.4 Multi-Modality

This work focuses on a framework for computer vision. However, the architecture has broader applicability and can be used for processing different sensor signals in multimodal settings [240]–[242]. Having similar cell architectures processing different signals is also in line with findings from neuroscience [59], [60].

In the case of images, net fragments in *S1* represent learned visual patterns that are part of an object's surface and are mapped with protection fibres to object prototypes that describe the visual appearance of objects. The same architectural structure can be applied to other types of signals. For example, an alternative sensory system could perceive audio signals. In this scenario, the local support in *S1* would extend over nearby frequency ranges and time intervals. Consequently, phonemes or syllables could correspond to frequently occurring patterns captured and supported by net fragments. In the second stage (*S2*), a sequence of phonemes or syllables could be mapped onto word prototypes.

Different sensory systems could even have separate domain-specific *S1* stages in a multimodal setting, while the prototypes in *S2* could be shared across modalities. This arrangement would allow the integration of various sensor signals and facilitates the creation of internal object representations with multiple modalities.

[240]: Ngiam, Khosla, Kim, *et al.* (2011), 'Multimodal Deep Learning'

[241]: Liu, Li, Xu, *et al.* (2018), *Learn to Combine Modalities in Multimodal Deep Learning*

[242]: Baltrusaitis, Ahuja, and Morency (2019), 'Multimodal Machine Learning'

[59]: Mountcastle (1978), 'An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System'

[60]: Mountcastle (1997), 'The columnar organization of the neocortex'

### 8.2.5 Framework Evaluation

Lastly, suitable metrics should be identified to evaluate the entire framework. Many of the results are analysed qualitatively by visual inspection but cannot be properly measured with corresponding metrics. While some metrics are proposed in Section 5.5.6 and Section 5.6.4, they need to be revised to analyse the entire framework and describe its performance from different perspectives.

# Appendix

# Results: Online Sources

# A


**Figure A.1:** QR-Codes with links to sources. The first code directs to the GitHub repository containing the source code, the second code to the repository containing the LaTeX-files to build this documentation, and the third code points to video visualisations of the results.

The codebase and results from this thesis have been released as open source on GitHub: The code is available at github.com/sagerpascal/lateral-connections, and the documentation is available at github.com/sagerpascal/msc-thesis. Furthermore, a GitHub webpage provides video visualisations from some of the results is available at sagerpascal.github.io/lateral-connections/results/final_results. QR codes linking to these URLs are provided in Figure A.1 for convenient access using electronic devices.

## A.1 Video

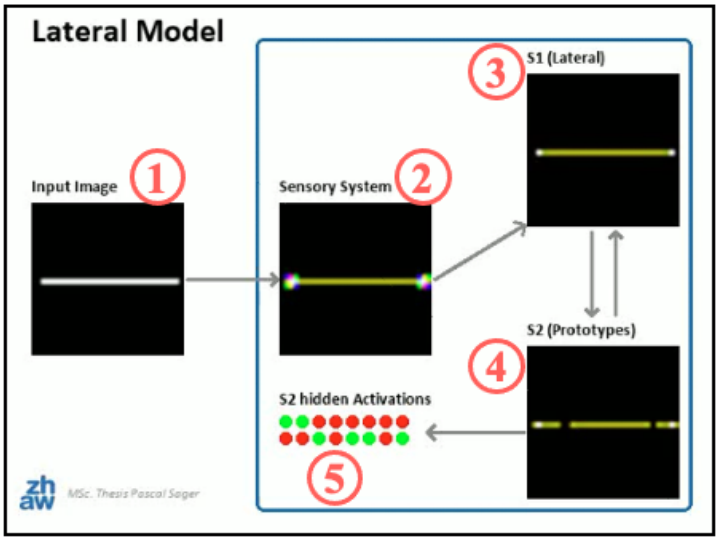


**Figure A.2:** An overview of the components visualised in the videos.

In the following, the video visualisations accessible at sagerpascal.github.io/lateral-connections/results/final_results are explained. These explanations are limited to an overview of which components are shown in each video. An interpretation of the video contents is provided in the corresponding result section.

Two video versions are shown for each experiment, both produced by the same model using the same parameter weights. In the first video version, the Bernoulli neuron is replaced by a neuron using a fixed threshold. This provides a video output that is more stable and has no flickering activations caused by sampling from a probability distribution. For the

*S0* and *S1* stages, a threshold of 0.5 is used. Consequently, neurons with a probability of $\geq 0.5$ are set to 1, while the other neurons are set to 0. A threshold of 0.9 is used for the *S2* stage, visualising only activities with high certainty that roughly correspond to those accepted by *S1* as feedback signals. The second video shows the network activities when the Bernoulli neurons are used.

Each video visualises the processing of the input over time. The first six video frames show how the video is processed over $T = 6$ timesteps of the fast loop, followed by five additional frames depicting the final prediction after the fast loop. By doing so, viewers have time to analyse the network's activations during this short interruption before the next input is fed into the model, and the process repeats.

In Figure A.2, a single video frame is shown, providing an overview of the components displayed in each video:

1. The left part of the video displays the input image fed into the sensory system. It is a binary image with one colour channel, whereby active pixels are depicted in white and inactive pixels are depicted in black.
2. The activations of the sensory system *S0* are shown in the middle of the video. The sensory system extracts 4 features at each location. Each feature combination is visualised in a different colour, and areas without activations are depicted in black.
3. *S1* is visualised in the top right corner. It uses the same colours as the sensory system. However, the activations might differ since neurons with insufficient lateral support are turned off, or other neurons might switch on.
4. *S2* is shown in the bottom right corner. It uses the same colours as the sensory system and *L1*. The visualisation depicts the returned prototype, i.e., the feedback provided to *S1* after mapping *S1'* activities to the latent variables.
5. The latent variables of *S2* are shown at the center bottom of the video as 16 circles. Each circle represents a cell, with green indicating an active cell and red indicating an inactive one.

For a detailed explanation of the content and observations in each video, please refer to the results chapter of this thesis.

# Negative Hebbian Learning

# B

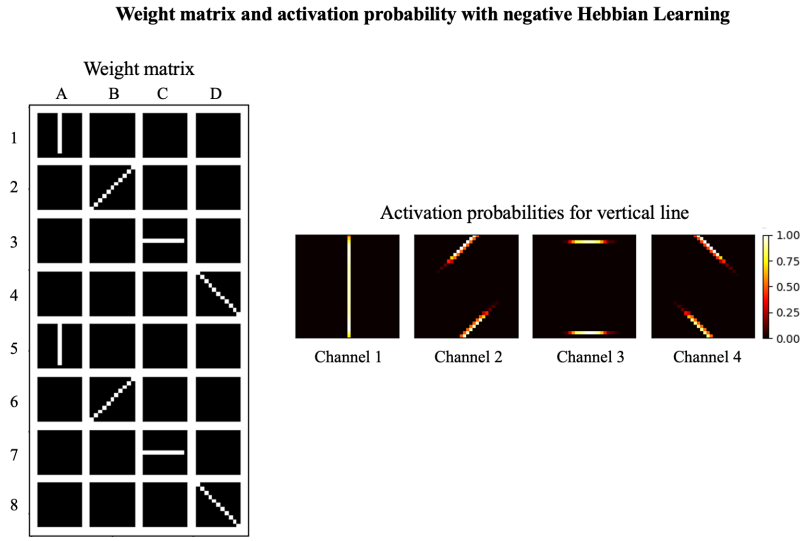


**Figure B.1:** The weight matrix and the corresponding activation probabilities for a vertical line of a model trained with negative Hebbian learning.

In this section, the concept of negative Hebbian Learning in *S1* is examined, a learning paradigm designed to allow neural networks to forget unimportant or inconsistent features. In the conducted experiments, only positive Hebbian learning [40] is used to strengthen connections between active cells. Conversely, negative Hebbian learning additionally reduces the connection strength between cells that fire disjointly[1]. While negative Hebbian learning facilitates eliminating previously learned but inconsistent connections, it also poses challenges in maintaining desired lateral connections that are needed to provide support between feature cells.

[40]: Hebb (1949), *The Organization of Behavior; A Neuropsychological Theory*

1: I.e. one cell is active while the other is inactive.

In Figure B.1, the weight matrix of a model trained with negative Hebbian updates and the activation probabilities of a vertical line fed into the model is visualised. Despite the divergence of the activation probabilities compared to those of positive Hebbian learning (c.f. Figure 7.10), these activations are still considered valid representations of lines. However, the major issue is that the output channels do not rely on multiple distinct features.

For instance, the output channel *A* representing vertical lines only considers the input channels 1 and 5, whereby channel 1 contains the "vertical lines features" from the sensory system, and channel 5 is its own recurrent connection. Regrettably, input channels 2-4, which contain additional sensory signals, are disregarded by output channel *A*. Therefore, only cells representing vertical line features support this output channel. In contrast, positive Hebbian learning considers all input channels, resulting in different features supporting each other.

Consequently, negative Hebbian learning leads to lower lateral support within the network. This might not be an issue for the used line dataset but is crucial for real-world scenarios[2]. Negative Hebbian learning, while facilitating the filtering out of irrelevant features, also tends to make

2: For example, one channel could represent eyes and another channel eyelashes. These features should support each other.

features mutually exclusive, which prevents learning adequate support between them.

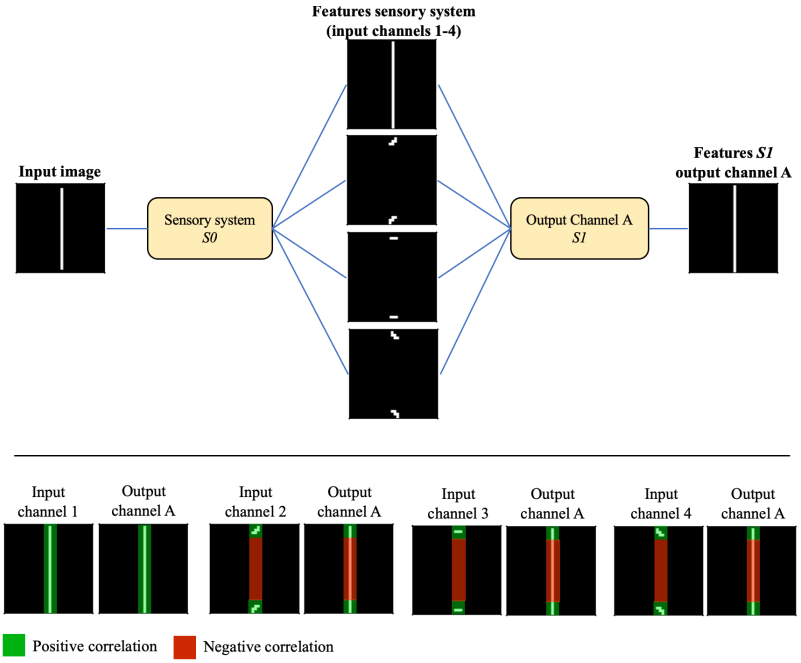


**Figure B.2:** Correlation between features from sensory input channels and the output channel $A$. The upper part of the images visualises the sensory features fed into output channel $A$ and the expected result. The lower part of the image indicates where positive and negative correlation occurs between sensory channels and output channel $A$.

The primary issue is that, except for one input channel, there are more negative correlations between the input channels and a single output channel, as illustrated in Figure B.2. In this context, "negative correlation" refers to disjointly active cells, while "positive correlation" refers to cells that fire together. In the case of the vertical line, the output channel $A$ is expected to reassemble the line roughly.

Hebbian learning compares this output with the input channels, strengthening the weights for positively correlated input-output pairs and weakening them for negatively correlated pairs. These correlations are visualised in the lower part of Figure B.2. Input channel 1 and output channel $A$ have high similarity. Therefore, the activations between input and output have a positive correlation, and the corresponding lateral connections defined by kernel $A1$ undergo a positive Hebbian update. However, all the other channels have a positive correlation only at the line ends and a negative correlation in the middle section of the line. Consequently, there is more negative than positive correlation at the kernels $A2$-$A4$, and these features undergo more negative than positive updates. This causes the lateral support learned at the line ends to be suppressed by the negative updates.

The resolution to this issue remains unclear and requires additional experiments. One potential approach is to use a significantly lower learning rate for negative updates than for positive updates; another solution could be to introduce alternative cells (c.f. Section 8.2.2)

# Bibliography


[1] A. G. Ivakhnenko and V. G. Lapa, *Cybernetic Predicting Devices*. New York, NY, USA: CCM Information Corporation, 1965 (cited on page 1).

[2] R. Dabre, C. Chu, and A. Kunchukuttan, 'A survey of multilingual neural machine translation,' *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–38, Sep. 30, 2021, ISSN: 0360-0300. DOI: 10.1145/3406095 (cited on page 1).

[3] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, 'CNN variants for computer vision: History, architecture, application, challenges and future scope,' *Electronics*, vol. 10, no. 20, pp. 2470–2498, Oct. 11, 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10202470 (cited on pages 1, 58).

[4] D. W. Otter, J. R. Medina, and J. K. Kalita, 'A survey of the usages of deep learning for natural language processing,' *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Feb. 2021, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2020.2979670 (cited on page 1).

[5] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, 'A review of deep learning based speech synthesis,' *Applied Sciences*, vol. 9, no. 19, p. 4050, Sep. 27, 2019, ISSN: 2076-3417. DOI: 10.3390/app9194050 (cited on page 1).

[6] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, 'Machine learning for industrial applications: A comprehensive literature review,' *Expert Systems with Applications*, vol. 175, no. 3, pp. 114 820–114 849, Aug. 2021, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2021.114820 (cited on pages 1, 11, 33, 58).

[7] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, *et al.*, *Summary of ChatGPT/GPT-4 research and perspective towards the future of large language models*, May 10, 2023. DOI: 10.48550/arXiv.2304.01852. [Online]. Available: http://arxiv.org/abs/2304.01852 (visited on 06/24/2023) (cited on page 1).

[8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, *Improving language understanding by generative pre-training*, 2018. [Online]. Available: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (visited on 06/24/2023) (cited on page 1).

[9] P. S. Rosenbloom, *Defining and exploring the intelligence space*, Jun. 16, 2023. DOI: 10.48550/arXiv.2306.06499. [Online]. Available: http://arxiv.org/abs/2306.06499 (visited on 06/24/2023) (cited on pages 1, 17, 76).

[10] M. Mitchell and D. C. Krakauer, 'The debate over understanding in AI's large language models,' *Proceedings of the National Academy of Sciences*, vol. 120, no. 13, e2215907120, Mar. 28, 2023, ISSN: 0027-8424. DOI: 10.1073/pnas.2215907120 (cited on pages 1, 17, 76).

[11] N. Akhtar and A. Mian, 'Threat of adversarial attacks on deep learning in computer vision: A survey,' *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2807385 (cited on pages 1, 17, 40, 76).

[12] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, *et al.*, 'Overcoming catastrophic forgetting in neural networks,' *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, Mar. 28, 2017, ISSN: 0027-8424. DOI: 10.1073/pnas.1611835114 (cited on pages 1, 12, 17, 40, 76).

[13] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, *et al.*, *Using DeepSpeed and megatron to train megatron-turing NLG 530b, a large-scale generative language model*, Feb. 4, 2022. DOI: 10.48550/arXiv.2201.11990. [Online]. Available: http://arxiv.org/abs/2201.11990 (visited on 06/28/2023) (cited on pages 1, 11, 17, 76).

[14] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC, USA: Spartan Books, 1962 (cited on pages 1, 3, 6, 10, 11, 14, 25, 33).

[15] S. Linnainmaa, 'Taylor expansion of the accumulated rounding error,' *BIT*, vol. 16, no. 2, pp. 146–160, Jun. 1976, ISSN: 0006-3835. DOI: 10.1007/BF01931367 (cited on pages 1, 3, 6, 10, 11, 14, 25, 33).

[16] Axios Media Inc. 'Artificial intelligence pioneer says we need to start over.' (2017), [Online]. Available: https://www.axios.com/2017/12/15/artificial-intelligence-pioneer-says-we-need-to-start-over-1513305524 (visited on 09/04/2022) (cited on page 1).

[17] T. Long, Q. Gao, L. Xu, and Z. Zhou, 'A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions,' *Computers & Security*, vol. 121, p. 102 847, Oct. 2022, ISSN: 0167-4048. DOI: 10.1016/j.cose.2022.102847 (cited on pages 1, 17, 76).

[18] P. Sager, S. Salzmann, F. Burn, and T. Stadelmann, 'Unsupervised domain adaptation for vertebrae detection and identification in 3d CT volumes using a domain sanity loss,' *Journal of Imaging*, vol. 8, no. 8, p. 222, Aug. 19, 2022, ISSN: 2313-433X. DOI: 10.3390/jimaging8080222 (cited on pages 1, 17).

[19] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, 'Improving sample efficiency in model-free reinforcement learning from images,' in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, ser. AAAI'21, vol. 35, Online: AAAI Press, May 18, 2021, pp. 10 674–10 681 (cited on page 1).

[20] C. von der Malsburg, T. Stadelmann, and B. F. Grewe, *A theory of natural intelligence*, Apr. 22, 2022. DOI: 10.48550/arXiv.2205.00002. [Online]. Available: http://arxiv.org/abs/2205.00002 (visited on 07/11/2022) (cited on pages 1, 2, 4, 17, 21, 27, 29–35, 37, 41, 43, 46, 66, 74, 75).

[21] S. del Prete, *Mountain spirit in winter*, 1982 (cited on page 2).

[22] M. Wertheimer, W. Köhler, W. Fuchs, W. Benary, K. Gottschaldt, F. Wulft, J. Ternus, and K. Duncker, *A source book of Gestalt psychology*. London, England: Kegan Paul, Trench, Trubner & Company, 1938 (cited on pages 2, 27, 31, 37, 75).

[23] W. Köhler, *Gestalt psychology: an introduction to new concepts in modern psychology*. New York, NY, USA: Liveright, 1992, ISBN: 978-0-87140-218-9 (cited on pages 2, 27, 31, 37, 75).

[24] J. Wagemans, J. Feldman, S. Gepshtein, R. Kimchi, J. R. Pomerantz, P. A. Van Der Helm, and C. Van Leeuwen, 'A century of gestalt psychology in visual perception: Conceptual and theoretical foundations.,' *Psychological Bulletin*, vol. 138, no. 6, pp. 1218–1252, 2012, ISSN: 0033-2909. DOI: 10.1037/a0029334 (cited on pages 2, 27, 31, 37, 75).

[25] D. W. Hamlyn, *The Psychology of Perception: A Philosophical Examination of Gestalt Theory and Derivative Theories of Perception*. London, England: Routledge, Mar. 27, 2017, ISBN: 978-1-315-47329-1 (cited on pages 2, 27, 31, 37, 75).

[26] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Cambridge, MA, USA: MIT Press, 2010, ISBN: 978-0-262-28961-0 (cited on pages 2, 9, 19, 20, 27, 31, 35, 37, 39, 74–76).

[27] E. Bienenstock and C. von der Malsburg, 'A neural network for invariant pattern recognition,' *Europhysics Letters (EPL)*, vol. 4, no. 1, pp. 121–126, Jul. 1, 1987, ISSN: 0295-5075. DOI: 10.1209/0295-5075/4/1/020 (cited on pages 2, 4, 22, 27, 74).

[28] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, 'Distortion invariant object recognition in the dynamic link architecture,' *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, Mar. 1993, ISSN: 0018-9340. DOI: 10.1109/12.210173 (cited on pages 2, 4, 22, 27, 74).

[29] L. Wiskott and C. von der Malsburg, 'Face recognition by dynamic link matching,' Ruhr-Univ., Inst. für Neuroinformatik, Bochum, Germany, IR-INI 96-05, 1996, p. 16 (cited on pages 2, 4, 22, 27, 32, 37, 74).

[30] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, 'Face recognition by elastic bunch graph matching,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, Jul. 1997, ISSN: 0162-8828. DOI: 10.1109/34.598235 (cited on pages 2, 4, 37, 74).

[31] P. Wolfrum, C. Wolff, J. Lücke, and C. von der Malsburg, 'A recurrent dynamic model for correspondence-based face recognition,' *Journal of Vision*, vol. 8, no. 7, p. 34, Dec. 29, 2008, ISSN: 1534-7362. DOI: 10.1167/8.7.34 (cited on pages 2, 4, 23, 32, 37, 52, 74–76).

[32] T. Fernandes and C. von der Malsburg, 'Self-organization of control circuits for invariant fiber projections,' *Neural Computation*, vol. 27, no. 5, pp. 1005–1032, May 2015, ISSN: 0899-7667. DOI: 10.1162/NECO_a_00725 (cited on pages 2, 4, 27, 36, 37, 54, 55, 74–78).

[33] K. Fukushima, 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,' *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, ISSN: 1432-0770. DOI: 10.1007/BF00344251 (cited on pages 3, 9, 45).

[34] A. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, 'Phoneme recognition using time-delay neural networks,' in *Meeting of the Institute of Electrical*, ser. IEICE'87, vol. 37, Tokyo, Japan: IEEE, 1987, pp. 329–339 (cited on pages 3, 9, 45).

[35] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, 'Backpropagation applied to handwritten zip code recognition,' *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541 (cited on pages 3, 9, 38, 43, 46).

[36] S. J. D. Prince, *Understanding Deep Learning*. Cambridge, MA, USA: MIT Press, 2023, ISBN: 978-0-262-04864-4 (cited on pages 3, 6, 8, 9, 11, 12, 28, 36).

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, Jun. 3, 2021. DOI: 10.48550/arXiv.2010.11929. [Online]. Available: http://arxiv.org/abs/2010.11929 (visited on 08/09/2023) (cited on pages 3, 10).

[38] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, *et al.*, 'MLP-mixer: An all-MLP architecture for vision,' in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NeurIPS'21, vol. 34, Online: Curran Associates, Inc., 2021, pp. 24 261–24 272 (cited on pages 3, 10).

[39] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, 'A comprehensive survey of loss functions in machine learning,' *Annals of Data Science*, vol. 9, no. 2, pp. 187–212, Apr. 2022, ISSN: 2198-5804. DOI: 10.1007/s40745-020-00253-5 (cited on pages 3, 10, 12, 33).

[40] D. O. Hebb, *The Organization of Behavior; A Neuropsychological Theory*. Oxford, England: Psychology Press, 1949 (cited on pages 3, 6, 13, 15, 17, 21, 29, 33, 38, 46, 54, 57, 61, 78, 85).

[41] J. S. Coombs, J. C. Eccles, and P. Fatt, 'The specific ionic conductances and the ionic movements across the motoneuronal membrane that produce the inhibitory post-synaptic potential,' *Journal of Physiology*, vol. 130, no. 2, pp. 326–373, Nov. 28, 1955, ISSN: 0022-3751. DOI: 10.1113/jphysiol.1955.sp005412 (cited on pages 3, 5, 29, 33, 38, 48).

[42] R. Morris, L. Tarassenko, and M. Kenward, *Cognitive Systems - Information Processing Meets Brain Science*. Edinburgh, Scotland: Elsevier, 2006, ISBN: 978-0-12-088566-4 (cited on pages 3, 23).

[43] S. Grossberg, 'Competitive learning: From interactive activation to adaptive resonance,' *Cognitive Science*, vol. 11, no. 1, pp. 23–63, Jan. 3, 1987, ISSN: 0364-0213. DOI: 10.1111/j.1551-6708.1987.tb00862.x (cited on pages 3, 6, 11, 31, 75).

[44] F. Crick, 'The recent excitement about neural networks,' *Nature*, vol. 337, no. 6203, pp. 129–132, Jan. 1989, ISSN: 0028-0836. DOI: 10.1038/337129a0 (cited on pages 3, 6, 11, 33, 75).

[45] C. Mouton, J. C. Myburgh, and M. H. Davel, 'Stride and translation invariance in CNNs,' in *Artificial Intelligence Research*, vol. 1342, Cham, Switzerland: Springer, 2020, pp. 267–281, ISBN: 978-3-030-66150-2. DOI: 10.1007/978-3-030-66151-9_17 (cited on pages 3, 9, 10).

[46] S. Madan, T. Henry, J. Dozier, H. Ho, N. Bhandari, T. Sasaki, F. Durand, H. Pfister, and X. Boix, 'When and how convolutional neural networks generalize to out-of-distribution category–viewpoint combinations,' *Nature Machine Intelligence*, vol. 4, no. 2, pp. 146–153, Feb. 21, 2022, ISSN: 2522-5839. DOI: 10.1038/s42256-021-00437-5 (cited on pages 3, 12, 17, 76).

[47] Wikipedia. 'Neuron.' (2023), [Online]. Available: https://en.wikipedia.org/wiki/Neuron (visited on 02/19/2023) (cited on page 5).

[48] M. E. Diamond, 'Identifying what makes a neuron fire,' *Journal of Physiology*, vol. 597, no. 10, pp. 2607–2608, May 2019, ISSN: 1469-7793. DOI: 10.1113/JP278049 (cited on page 5).

[49] H. Takagi, 'Roles of ion channels in EPSP integration at neuronal dendrites,' *Neuroscience Research*, vol. 37, no. 3, pp. 167–171, Jul. 2000, ISSN: 0168-0102. DOI: 10.1016/S0168-0102(00)00120-6 (cited on page 5).

[50] C. J. Wilson and P. M. Groves, 'Spontaneous firing patterns of identified spiny neurons in the rat neostriatum,' *Brain Research*, vol. 220, no. 1, pp. 67–80, Sep. 1981, ISSN: 0006-8993. DOI: 10.1016/0006-8993(81)90211-0 (cited on page 5).

[51] S. Herculano-Houzel, 'The human brain in numbers: A linearly scaled-up primate brain,' *Frontiers in Human Neuroscience*, vol. 3, p. 31, 2009, ISSN: 1662-5161. DOI: 10.3389/neuro.09.031.2009 (cited on page 6).

[52] H. Zeng and J. R. Sanes, 'Neuronal cell-type classification: Challenges, opportunities and the path forward,' *Nature Reviews Neuroscience*, vol. 18, no. 9, pp. 530–546, Sep. 2017, ISSN: 1471-003X. DOI: 10.1038/nrn.2017.85 (cited on page 6).

[53] W. S. McCulloch and W. Pitts, 'A logical calculus of the ideas immanent in nervous activity,' *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943, ISSN: 0007-4985. DOI: 10.1007/BF02478259 (cited on pages 6, 7).

[54] F. Rosenblatt, 'The perceptron: A probabilistic model for information storage and organization in the brain.,' *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, ISSN: 0033-295X. DOI: 10.1037/h0042519 (cited on pages 6, 8).

[55] K. Fukushima, 'Visual feature extraction by a multilayered network of analog threshold elements,' *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 4, pp. 322–333, 1969, ISSN: 0536-1567. DOI: 10.1109/TSSC.1969.300225 (cited on pages 6, 8, 9).

[56] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016 (cited on pages 6–10).

[57] T. Glasmachers, 'Limits of end-to-end learning,' in *Proceedings of the 9th Asian Conference on Machine Learning*, ser. ACML'17, vol. 77, Seoul, South Korea: PMLR, Nov. 15, 2017, pp. 17–32 (cited on page 6).

[58] D. J. Felleman and D. C. Van Essen, 'Distributed hierarchical processing in the primate cerebral cortex,' *Cerebral Cortex*, vol. 1, no. 1, pp. 1–47, Jan. 1, 1991, ISSN: 1460-2199. DOI: 10.1093/cercor/1.1.1 (cited on pages 6, 7, 36).

[59] V. Mountcastle, 'An organizing principle for cerebral function: The unit model and the distributed system,' in *The Mindful Brain*, Cambridge, MA, USA: MIT Press, 1978, pp. 7–50 (cited on pages 6, 17, 21, 79).

[60] V. Mountcastle, 'The columnar organization of the neocortex,' *Brain*, vol. 120, no. 4, pp. 701–722, Apr. 1, 1997, ISSN: 1460-2156. DOI: 10.1093/brain/120.4.701 (cited on pages 6, 17, 21, 28, 79).

[61] M. Costandi, *Neuroplasticity*. Cambridge, MA, USA: MIT Press, 2016, ISBN: 978-0-262-52933-4 (cited on pages 6, 13).

[62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, 'Learning representations by back-propagating errors,' *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, ISSN: 0028-0836. DOI: 10.1038/323533a0 (cited on page 6).

[63] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. E. Hinton, 'Backpropagation and the brain,' *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, Jun. 2020, ISSN: 1471-003X. DOI: 10.1038/s41583-020-0277-3 (cited on page 6).

[64] G. Cybenko, 'Approximation by superpositions of a sigmoidal function,' *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989, ISSN: 0932-4194. DOI: 10.1007/BF02551274 (cited on page 9).

[65] Y. Bengio, 'Deep learning of representations for unsupervised and transfer learning,' in *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning*, ser. ICML'12, vol. 27, Bellevue, DC, USA: PMLR, Jul. 2, 2012, pp. 17–36 (cited on pages 9, 10).

[66] D. H. Hubel and T. N. Wiesel, 'Receptive fields and functional architecture of monkey striate cortex,' *Journal of Physiology*, vol. 195, no. 1, pp. 215–243, Mar. 1, 1968, ISSN: 0022-3751. DOI: 10.1113/jphysiol.1968.sp008455 (cited on page 9).

[67] W. Zhang, K. Itoh, J. Tanida, and Y. Ichioka, 'Parallel distributed processing model with local space-invariant interconnections and its optical architecture,' *Applied Optics*, vol. 29, no. 32, p. 4790, Nov. 10, 1990, ISSN: 0003-6935. DOI: 10.1364/AO.29.004790 (cited on page 9).

[68] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, 'Flexible, high performance convolutional neural networks for image classification,' in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, ser. IJCAI'11, vol. 2, Barcelona, Spain: AAAI Press, 2011, pp. 1237–1242, ISBN: 978-1-57735-514-4 (cited on page 10).

[69] S. Sharma and R. Mehra, 'Implications of pooling strategies in convolutional neural networks: A deep insight,' *Foundations of Computing and Decision Sciences*, vol. 44, no. 3, pp. 303–330, Sep. 1, 2019, ISSN: 2300-3405. DOI: 10.2478/fcds-2019-0016 (cited on page 10).

[70] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*, 4th. Hoboken, NY, USA: Pearson, 2021, ISBN: 978-0-13-461099-3 (cited on pages 10, 24).

[71] N. Simmler, P. Sager, P. Andermatt, R. Chavarriaga, F.-P. Schilling, M. Rosenthal, and T. Stadelmann, 'A survey of un-, weakly-, and semi-supervised learning methods for noisy, missing and partial labels in industrial vision applications,' in *Proceedings of the 8th Swiss Conference on Data Science*, ser. SDS'21, Lucerne, Switzerland: IEEE, Jun. 2021, pp. 26–31, ISBN: 978-1-66543-874-2. DOI: 10.1109/SDS51136.2021.00012 (cited on page 10).

[72] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, Jan. 29, 2017. DOI: 10.48550/arXiv.1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 06/27/2023) (cited on pages 10, 62).

[73] J. Zhang, T. He, S. Sra, and A. Jadbabaie, *Why gradient clipping accelerates training: A theoretical justification for adaptivity*, Feb. 10, 2020. DOI: 10.48550/arXiv.1905.11881. [Online]. Available: http://arxiv.org/abs/1905.11881 (visited on 07/03/2023) (cited on page 11).

[74] S. Ioffe and C. Szegedy, 'Batch normalization: Accelerating deep network training by reducing internal covariate shift,' in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, ser. ICML'15, vol. 37, Lille, France: PMLR, 2015, pp. 448–456 (cited on page 11).

[75] A. Buetti-Dinh, V. Galli, S. Bellenberg, O. Ilie, M. Herold, *et al.*, 'Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition,' *Biotechnology Reports*, vol. 22, e00321, Jun. 2019, ISSN: 2215-017X. DOI: 10.1016/j.btre.2019.e00321 (cited on pages 11, 33, 76).

[76] G. E. Moore, 'Cramming more components onto integrated circuits,' *Electronics*, vol. 38, no. 8, pp. 114–1116, Apr. 1965 (cited on page 11).

[77] S. Kumar, *Fundamental limits to moore's law*, Nov. 17, 2015. DOI: 10.48550/arXiv.1511.05956. [Online]. Available: http://arxiv.org/abs/1511.05956 (visited on 06/27/2023) (cited on page 11).

[78] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, 'Deep contextualized word representations,' in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, ser. NAACL'18, vol. 1, New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202 (cited on page 11).

[79] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, *et al.*, 'Language models are few-shot learners,' in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NeurIPS'20, vol. 33, Vancouver, Canada: Curran Associates, Inc., 2020, pp. 1877–1901 (cited on pages 11, 17).

[80] Open AI. 'AI and compute.' (2018), [Online]. Available: https://openai.com/blog/ai-and-compute/ (visited on 08/19/2022) (cited on page 11).

[81] A. Berthelier, T. Chateau, S. Duffner, C. Garcia, and C. Blanc, 'Deep model compression and architecture optimization for embedded systems: A survey,' *Journal of Signal Processing Systems*, vol. 93, no. 8, pp. 863–878, Aug. 2021, ISSN: 1939-8115. DOI: 10.1007/s11265-020-01596-1 (cited on page 11).

[82] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, *Integer quantization for deep learning inference: Principles and empirical evaluation*, Apr. 20, 2020. DOI: 10.48550/arXiv.2004.09602. [Online]. Available: http://arxiv.org/abs/2004.09602 (visited on 06/28/2023) (cited on page 11).

[83] T. Choudhary, V. Mishra, A. Goswami, and J. Sarangapani, 'A comprehensive survey on model compression and acceleration,' *Artificial Intelligence Review*, vol. 53, no. 7, pp. 5113–5155, Oct. 2020, ISSN: 1573-7462. DOI: 10.1007/s10462-020-09816-7 (cited on page 11).

[84] S. Zhou, Y. Wang, D. Chen, J. Chen, X. Wang, C. Wang, and J. Bu, 'Distilling holistic knowledge with graph neural networks,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ser. ICCV'21, Montreal, Canada: IEEE, Oct. 2021, pp. 10 387–10 396 (cited on page 11).

[85] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, 'Estimation of energy consumption in machine learning,' *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, Dec. 2019, ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2019.07.007 (cited on pages 11, 17, 76).

[86] H. Liu, Y. Yang, and X. Wang, 'Overcoming catastrophic forgetting in graph neural networks,' in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, ser. AAAI'21, vol. 35, Online: AAAI Press, May 18, 2021, pp. 8653–8661 (cited on pages 12, 40).

[87] Y. Zhang and Q. Yang, 'A survey on multi-task learning,' *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, Dec. 1, 2022, ISSN: 1041-4347. DOI: 10.1109/TKDE.2021.3070203 (cited on page 12).

[88] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, 'Online deep learning: Learning deep neural networks on the fly,' in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, ser. IJCAI'18, vol. 4, Stockholm, Sweden: AAAI Press, Jul. 2018, pp. 2660–2666, ISBN: 978-0-9992411-2-7. DOI: 10.24963/ijcai.2018/369 (cited on pages 12, 17).

[89] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, 'Continual lifelong learning with neural networks: A review,' *Neural Networks*, vol. 113, pp. 54–71, May 2019, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2019.01.012 (cited on pages 12, 40).

[90] G. Marcus, *Deep learning: A critical appraisal*, Jan. 2, 2018. DOI: 10.48550/arXiv.1801.00631. [Online]. Available: http://arxiv.org/abs/1801.00631 (visited on 06/28/2023) (cited on pages 12, 17, 76).

[91] G. M. Allenby, P. E. Rossi, and R. E. McCulloch, 'Hierarchical bayes models: A practitioners guide,' *Journal of Bayesian Applications in Marketing*, vol. 11, pp. 418–440, Jan. 2005, ISSN: 1556-5068. DOI: 10.2139/ssrn.655541 (cited on page 12).

[92] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. Cambridge, MA, USA: MIT Press, 2009, ISBN: 978-0-262-01319-2 (cited on page 12).

[93] E. A. Mayer, 'Gut feelings: The emerging biology of gut–brain communication,' *Nature Reviews Neuroscience*, vol. 12, no. 8, pp. 453–466, Aug. 2011, ISSN: 1471-003X. DOI: 10.1038/nrn3071 (cited on page 12).

[94] *Deep Reinforcement Learning: Fundamentals, Research and Applications*, 1st ed., Singapore, Republic of Singapore: Springer Singapore, 2020, ISBN: 978-9-811-54095-0. DOI: 10.1007/978-981-15-4095-0 (cited on page 12).

[95] H. Moravec, *Mind Children: The Future of Robot and Human Intelligence*. Cambridge, MA, USA: Harvard University Press, 1995, ISBN: 978-0-674-57618-6 (cited on page 12).

[96] S. P. Newman, *Current perspectives in dysphasia*. Edinburgh, Scotland: Churchill Livingstone, 1985, ISBN: 978-0-443-03039-0 (cited on page 13).

[97] X. Liu, S. Ramirez, P. T. Pang, C. B. Puryear, A. Govindarajan, K. Deisseroth, and S. Tonegawa, 'Optogenetic stimulation of a hippocampal engram activates fear memory recall,' *Nature*, vol. 484, no. 7394, pp. 381–385, Apr. 2012, ISSN: 0028-0836. DOI: 10.1038/nature11028 (cited on pages 13, 35).

[98] E. Oja, 'Simplified neuron model as a principal component analyzer,' *Journal of Mathematical Biology*, vol. 15, no. 3, pp. 267–273, Nov. 1982, ISSN: 0303-6812. DOI: 10.1007/BF00275687 (cited on pages 13, 14).

[99] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, 'Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex,' *Journal of Neuroscience*, vol. 2, no. 1, pp. 32–48, Jan. 1, 1982, ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.02-01-00032.1982 (cited on page 14).

[100] N. Intrator and L. N. Cooper, 'Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions,' *Neural Networks*, vol. 5, no. 1, pp. 3–17, Jan. 1992, ISSN: 0893-6080. DOI: 10.1016/S0893-6080(05)80003-6 (cited on page 14).

[101] E. P. Simoncelli and B. A. Olshausen, 'Natural image statistics and neural representation,' *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, Mar. 2001, ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.24.1.1193 (cited on page 14).

[102] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner, 'Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks,' *Science*, vol. 334, no. 6062, pp. 1569–1573, Dec. 16, 2011, ISSN: 0036-8075. DOI: 10.1126/science.1211095 (cited on pages 14, 29, 33, 38, 48, 50, 78).

[103] P. Joshi and J. Triesch, 'Rules for information maximization in spiking neurons using intrinsic plasticity,' in *International Joint Conference on Neural Networks*, ser. IJCNN'09, Atlanta, GA, USA: IEEE, Jun. 2009, pp. 1456–1461, ISBN: 978-1-4244-3548-7. DOI: 10.1109/IJCNN.2009.5178625 (cited on pages 14, 50, 78).

[104] M. Teichmann and F. Hamker, 'Intrinsic plasticity: A simple mechanism to stabilize hebbian learning in multilayer neural networks,' in *Proceedings Workshop New Challenges in Neural Computation*, ser. NC2, Aachen, Germany: Machine Learning Reports, Mar. 2015, pp. 103–111 (cited on pages 14, 50, 78).

[105] S. Risi. 'The future of artificial intelligence is self-organizing and self-assembling,' sebastianrisi.com. (2021), [Online]. Available: https://sebastianrisi.com/self%5C%5Fassembling%5C%5Fai (cited on page 14).

[106] J. J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities.,' *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, ISSN: 0027-8424. DOI: 10.1073/pnas.79.8.2554 (cited on pages 14, 15).

[107] E. Fix and J. L. Hodges, 'Discriminatory analysis. nonparametric discrimination: Consistency properties,' *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, p. 238, Dec. 1989, ISSN: 0306-7734. DOI: 10.2307/1403797 (cited on page 14).

[108] J. Weston, S. Chopra, and A. Bordes, *Memory networks*, Nov. 29, 2015. DOI: 10.48550/arXiv.1410.3916. [Online]. Available: http://arxiv.org/abs/1410.3916 (visited on 06/29/2023) (cited on pages 14, 15).

[109] J. J. Hopfield, D. I. Feinstein, and R. G. Palmer, ''unlearning' has a stabilizing effect in collective memories,' *Nature*, vol. 304, no. 5922, pp. 158–159, Jul. 1983, ISSN: 0028-0836. DOI: 10.1038/304158a0 (cited on page 15).

[110] R. J. McEliece, E. C. Posner, E. Rodemich, and S. Venkatesh, 'The capacity of the hopfield associative memory,' *IEEE Transactions on Information Theory*, vol. 33, no. 4, pp. 461–482, Jul. 1987, ISSN: 0018-9448. DOI: 10.1109/TIT.1987.1057328 (cited on page 15).

[111] D. Krotov and J. J. Hopfield, 'Dense associative memory for pattern recognition,' in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16, vol. 29, Red Hook, NY, USA: Curran Associates, Inc., 2016, pp. 1180–1188, ISBN: 978-1-5108-3881-9 (cited on page 15).

[112] M. Demircigil, J. Heusel, M. Löwe, S. Upgang, and F. Vermet, 'On a model of associative memory with huge storage capacity,' *Journal of Statistical Physics*, vol. 168, no. 2, pp. 288–299, Jul. 2017, ISSN: 1572-9613. DOI: 10.1007/s10955-017-1806-y (cited on page 15).

[113] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, *et al.*, *Hopfield networks is all you need*, Apr. 28, 2021. DOI: 10.48550/arXiv.2008.02217. [Online]. Available: http://arxiv.org/abs/2008.02217 (visited on 06/29/2023) (cited on pages 15, 55).

[114] G. Kowalski, *Information Retrieval Systems*, 1st. New York, NY, USA: Springer, 1997, vol. 1, ISBN: 978-0-7923-9926-1 (cited on page 15).

[115] S. J. Thorpe, 'Spike arrival times: A highly efficient coding scheme for neural networks,' in *Parallel processing in neural systems and computers*, New York, NY, USA: Elsevier, 1990, pp. 91–94, ISBN: 978-0-444-88390-2 (cited on page 15).

[116] W. Maass, 'Networks of spiking neurons: The third generation of neural network models,' *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997, ISSN: 0893-6080. DOI: 10.1016/S0893-6080(97)00011-7 (cited on page 15).

[117] L. F. Abbott, 'Lapicque's introduction of the integrate-and-fire model neuron,' *Brain Research Bulletin*, vol. 50, no. 5, pp. 303–304, Nov. 1999, ISSN: 0361-9230. DOI: 10.1016/S0361-9230(99)00161-6 (cited on page 15).

[118] E. M. Izhikevich, 'Simple model of spiking neurons,' *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003, ISSN: 1045-9227. DOI: 10.1109/TNN.2003.820440 (cited on page 15).

[119] R. Brette and W. Gerstner, 'Adaptive exponential integrate-and-fire model as an effective description of neuronal activity,' *Journal of Neurophysiology*, vol. 94, no. 5, pp. 3637–3642, Nov. 2005, ISSN: 1522-1598. DOI: 10.1152/jn.00686.2005 (cited on page 15).

[120] G.-q. Bi and M.-m. Poo, 'Synaptic modification by correlated activity: Hebb's postulate revisited,' *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 139–166, Mar. 2001, ISSN: 1545-4126. DOI: 10.1146/annurev.neuro.24.1.139 (cited on page 15).

[121] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, 'STDP-based spiking deep convolutional neural networks for object recognition,' *Neural Networks*, vol. 99, pp. 56–67, Mar. 2018, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2017.12.005 (cited on page 15).

[122] J. D. Nunes, M. Carvalho, D. Carneiro, and J. S. Cardoso, 'Spiking neural networks: A survey,' *IEEE Access*, vol. 10, pp. 60 738–60 764, 2022, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3179968 (cited on page 16).

[123] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, *et al.*, *LLaMA: Open and efficient foundation language models*, Feb. 27, 2023. DOI: 10.48550/arXiv.2302.13971. [Online]. Available: http://arxiv.org/abs/2302.13971 (visited on 07/01/2023) (cited on page 17).

[124] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, *et al.*, *Training language models to follow instructions with human feedback*, Mar. 4, 2022. DOI: 10.48550/arXiv.2203.02155. [Online]. Available: http://arxiv.org/abs/2203.02155 (visited on 07/01/2023) (cited on page 17).

[125] P. Feldman, J. R. Foulds, and S. Pan, *Trapping LLM hallucinations using tagged context prompts*, Jun. 9, 2023. DOI: 10.48550/arXiv.2306.06085. [Online]. Available: http://arxiv.org/abs/2306.06085 (visited on 07/01/2023) (cited on page 17).

[126] P. Manakul, A. Liusie, and M. J. F. Gales, *SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models*, May 7, 2023. DOI: 10.48550/arXiv.2303.08896. [Online]. Available: http://arxiv.org/abs/2303.08896 (visited on 07/01/2023) (cited on page 17).

[127] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, 'Online learning: A comprehensive survey,' *Neurocomputing*, vol. 459, pp. 249–289, Oct. 2021, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.04.112 (cited on page 17).

[128] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, 'A comprehensive survey on transfer learning,' *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, ISSN: 0018-9219. DOI: 10.1109/JPROC.2020.3004555 (cited on page 17).

[129] S. Thrun and L. Pratt, 'Introduction and overview,' in *Learning to Learn*, Boston, MA, USA: Springer, 1998, pp. 3–17, ISBN: 978-1-4615-5529-2. DOI: 10.1007/978-1-4615-5529-2_1 (cited on page 17).

[130] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, 'Meta-learning in neural networks: A survey,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 5149–5169, Sep. 2022, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2021.3079209 (cited on page 17).

[131] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, 'A framework for intelligence and cortical function based on grid cells in the neocortex,' *Frontiers in Neural Circuits*, vol. 12, p. 121, Jan. 11, 2019, ISSN: 1662-5110. DOI: 10.3389/fncir.2018.00121 (cited on page 17).

[132] M. Lewis, S. Purdy, S. Ahmad, and J. Hawkins, 'Locations in the neocortex: A theory of sensorimotor object recognition using cortical grid cells,' *Frontiers in Neural Circuits*, vol. 13, p. 22, Apr. 24, 2019, ISSN: 1662-5110. DOI: 10.3389/fncir.2019.00022 (cited on page 17).

[133] Y. Yang, H. Lv, and N. Chen, 'A survey on ensemble learning under the era of deep learning,' *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5545–5589, Jun. 2023, ISSN: 0269-2821. DOI: 10.1007/s10462-022-10283-5 (cited on page 18).

[134] T. K. Ho, 'Random decision forests,' in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, ser. ICDAR'95, vol. 1, Montreal, Canada: IEEE, 1995, pp. 278–282, ISBN: 978-0-8186-7128-9. DOI: 10.1109/ICDAR.1995.598994 (cited on page 18).

[135] M. R. Ferrier, *Toward a universal cortical algorithm: Examining hierarchical temporal memory in light of frontal cortical function*, Nov. 17, 2014. DOI: 10.48550/arXiv.1411.4702. [Online]. Available: http://arxiv.org/abs/1411.4702 (visited on 07/01/2023) (cited on page 18).

[136] D. George, W. Lehrach, K. Kansky, M. Lázaro-Gredilla, C. Laan, *et al.*, 'A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs,' *Science*, vol. 358, no. 6368, eaag2612, Dec. 8, 2017, ISSN: 0036-8075. DOI: 10.1126/science.aag2612 (cited on pages 18, 20).

[137] A. K. Garg, P. Li, M. S. Rashid, and E. M. Callaway, 'Color and orientation are jointly coded and spatially organized in primate primary visual cortex,' *Science*, vol. 364, no. 6447, pp. 1275–1279, Jun. 28, 2019, ISSN: 0036-8075. DOI: 10.1126/science.aaw5868 (cited on page 18).

[138] C. Gilbert, J. Hirsch, and T. Wiesel, 'Lateral interactions in visual cortex,' *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 55, no. 1, pp. 663–677, Jan. 1, 1990, ISSN: 0091-7451. DOI: 10.1101/SQB.1990.055.01.063 (cited on pages 18, 21, 28, 31, 34, 38, 75).

[139] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, 'Gradient-based learning applied to document recognition,' *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 0018-9219. DOI: 10.1109/5.726791 (cited on pages 18, 19, 79).

[140] S. Sabour, N. Frosst, and G. E. Hinton, 'Dynamic routing between capsules,' in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, vol. 30, Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 3859–3869, ISBN: 978-1-5108-6096-4 (cited on pages 18, 20).

[141] G. R. Ash, R. H. Cardwell, and R. P. Murray, 'Design and optimization of networks with dynamic routing,' *Bell System Technical Journal*, vol. 60, no. 8, pp. 1787–1820, Oct. 1981, ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1981.tb00297.x (cited on page 18).

[142] A. Krizhevsky, 'Learning multiple layers of features from tiny images,' Master's Thesis, Canadian Institute for Advanced Research, Toronto, Canada, 2009 (cited on pages 19, 79).

[143] Y. Ma, D. Tsao, and H.-Y. Shum, 'On the principles of parsimony and self-consistency for the emergence of intelligence,' *Frontiers of Information Technology & Electronic Engineering*, vol. 23, no. 9, pp. 1298–1323, Sep. 2022, ISSN: 2095-9184. DOI: 10.1631/FITEE.2200297 (cited on page 19).

[144] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, 'Self-critical sequence training for image captioning,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR'17, Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1179–1195, ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.131 (cited on page 19).

[145] G. Knoblich and N. Sebanz, 'The social nature of perception and action,' *Current Directions in Psychological Science*, vol. 15, no. 3, pp. 99–104, Jun. 2006, ISSN: 0963-7214. DOI: 10.1111/j.0963-7214.2006.00415.x (cited on page 19).

[146] B. Zhou, P. Krähenbühl, and V. Koltun, 'Does computer vision matter for action?' *Science Robotics*, vol. 4, no. 30, eaaw6661, May 22, 2019, ISSN: 2470-9476. DOI: 10.1126/scirobotics.aaw6661 (cited on page 19).

[147] G. B. Keller, T. Bonhoeffer, and M. Hübener, 'Sensorimotor mismatch signals in primary visual cortex of the behaving mouse,' *Neuron*, vol. 74, no. 5, pp. 809–815, Jun. 2012, ISSN: 0896-6273. DOI: 10.1016/j.neuron.2012.03.040 (cited on page 19).

[148] H. Keurti, H.-R. Pan, M. Besserve, B. F. Grewe, and B. Schölkopf, *Homomorphism autoencoder – learning group structured representations from observed transitions*, Jun. 6, 2023. DOI: 10.48550/arXiv.2207.12067. [Online]. Available: http://arxiv.org/abs/2207.12067 (visited on 07/01/2023) (cited on page 19).

[149] J. Piaget, 'Cognitive development in children: Piaget development and learning,' *Journal of Research in Science Teaching*, vol. 2, no. 3, pp. 176–186, Sep. 1964, ISSN: 0022-4308. DOI: 10.1002/tea.3660020306 (cited on page 19).

[150] Y. LeCun, 'A path towards autonomous machine intelligence,' *Open Review*, vol. 62, Jun. 2022 (cited on page 20).

[151] G. E. Hinton, 'Training products of experts by minimizing contrastive divergence,' *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, Aug. 1, 2002, ISSN: 0899-7667. DOI: 10.1162/089976602760128018 (cited on pages 20, 22, 61, 62).

[152] K. Friston, T. FitzGerald, F. Rigoli, P. Schwartenbeck, J. O'Doherty, and G. Pezzulo, 'Active inference and learning,' *Neuroscience & Biobehavioral Reviews*, vol. 68, pp. 862–879, Sep. 2016, ISSN: 0149-7634. DOI: 10.1016/j.neubiorev.2016.06.022 (cited on page 20).

[153] T. Parr, G. Pezzulo, and K. J. Friston, *Active Inference: The Free Energy Principle in Mind, Brain, and Behavior*. Cambridge, MA, USA: MIT Press, Mar. 29, 2022, ISBN: 978-0-262-36997-8 (cited on page 20).

[154] E. Bengio, M. Jain, M. Korablyov, D. Precup, and Y. Bengio, 'Flow network based generative models for non-iterative diverse candidate generation,' in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NeurIPS'21, vol. 34, Online: Curran Associates, Inc., 2021, pp. 27 381–27 394 (cited on page 20).

[155] Y. Bengio, S. Lahlou, T. Deleu, E. J. Hu, M. Tiwari, and E. Bengio, *GFlowNet foundations*, Aug. 15, 2022. DOI: 10.48550/arXiv.2111.09266. [Online]. Available: http://arxiv.org/abs/2111.09266 (visited on 07/01/2023) (cited on page 20).

[156] Y. Du and I. Mordatch, *Implicit generation and generalization in energy-based models*, Jun. 29, 2020. DOI: 10.48550/arXiv.1903.08689. [Online]. Available: http://arxiv.org/abs/1903.08689 (visited on 07/01/2023) (cited on page 20).

[157] S. Ahmad and J. Hawkins, *Properties of sparse distributed representations and their application to hierarchical temporal memory*, Mar. 25, 2015. DOI: 10.48550/arXiv.1503.07469. [Online]. Available: http://arxiv.org/abs/1503.07469 (visited on 07/02/2023) (cited on pages 20, 38, 40, 41).

[158] J. D. McPherson, M. Marra, L. Hillier, R. H. Waterston, A. Chinwalla, *et al.*, 'A physical map of the human genome,' *Nature*, vol. 409, no. 6822, pp. 934–941, Feb. 1, 2001, ISSN: 1476-4687. DOI: 10.1038/35057157 (cited on page 21).

[159] M. S. Gazzaniga, 'Organization of the human brain,' *Science*, vol. 245, no. 4921, pp. 947–952, Sep. 1989, ISSN: 0036-8075. DOI: 10.1126/science.2672334 (cited on page 21).

[160] S. Ackerman, *Discovering the brain*. Washington, DC, USA: National Academy Press, 1992, ISBN: 978-0-309-04529-2 (cited on page 21).

[161] D. S. Bassett and M. S. Gazzaniga, 'Understanding complexity in the human brain,' *Trends in Cognitive Sciences*, vol. 15, no. 5, pp. 200–209, May 2011, ISSN: 1364-6613. DOI: 10.1016/j.tics.2011.03.006 (cited on page 21).

[162] A. N. Kolmogorov, 'On tables of random numbers,' *Theoretical Computer Science*, vol. 207, no. 2, pp. 387–395, Nov. 1998, ISSN: 0304-3975. DOI: 10.1016/S0304-3975(98)00075-9 (cited on page 21).

[163] D. J. Willshaw and C. von der Malsburg, 'How patterned neural connections can be set up by self-organization,' *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 194, no. 1117, pp. 431–445, Nov. 12, 1976, ISSN: 0080-4649. DOI: 10.1098/rspb.1976.0087 (cited on page 21).

[164] D. J. Willshaw and C. von der Malsburg, 'A marker induction mechanism for the establishment of ordered neural mappings: Its application to the retinotectal problem,' *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, vol. 287, no. 1021, pp. 203–243, Nov. 1979, ISSN: 0080-4622. DOI: 10.1098/rstb.1979.0056 (cited on page 21).

[165] W. Singer, 'The brain as a self-organizing system,' *European Archives of Psychiatry and Neurological Sciences*, vol. 236, no. 1, pp. 4–9, 1986, ISSN: 0175-758X. DOI: 10.1007/BF00641050 (cited on page 21).

[166] J. A. S. Kelso and A. Fuchs, 'Self-organizing dynamics of the human brain: Critical instabilities and šil'nikov chaos,' *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 5, no. 1, pp. 64–69, Mar. 1995, ISSN: 1054-1500. DOI: 10.1063/1.166087 (cited on page 21).

[167] S. J. A. Kelso, *Dynamic patterns: the self-organization of brain and behavior*, 3rd. Cambridge, MA, USA: MIT Press, 1999, ISBN: 978-0-262-61131-2 (cited on page 21).

[168] C. von der Malsburg and E. L. Bienenstock, 'A neural network for the retrieval of superimposed connection patterns,' *Europhysics Letters (EPL)*, vol. 3, no. 11, pp. 1243–1249, Jun. 1, 1987, ISSN: 0295-5075. DOI: 10.1209/0295-5075/3/11/015 (cited on pages 21, 41, 50).

[169] C. von der Malsburg, 'Concerning the neuronal code,' *Journal of Cognitive Science*, vol. 19, no. 4, pp. 511–550, Dec. 2018. DOI: 10.17791/JCS.2018.19.4.511 (cited on pages 21, 23, 28–34, 38, 41, 43, 66, 74, 75).

[170] C. Lehmann, 'Leveraging neuroscience for deep learning based object recognition,' Master's Thesis, Zurich University of Applied Sciences, Winterthur, Switzerland, 2022 (cited on page 21).

[171] F. Tong, 'Primary visual cortex and visual awareness,' *Nature Reviews Neuroscience*, vol. 4, no. 3, pp. 219–229, Mar. 2003, ISSN: 1471-003X. DOI: 10.1038/nrn1055 (cited on pages 22, 27, 31, 34, 37, 75).

[172] K. Grill-Spector and R. Malach, 'The human visual cortex,' *Annual Review of Neuroscience*, vol. 27, no. 1, pp. 649–677, Jul. 21, 2004, ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.27.070203.144220 (cited on pages 22, 27, 31, 34, 37, 75).

[173] Y. Miyashita, 'Inferior temporal cortex: Where visual perception meets memory,' *Annual Review of Neuroscience*, vol. 16, no. 1, pp. 245–263, Mar. 1993, ISSN: 0147-006X. DOI: 10.1146/annurev.ne.16.030193.001333 (cited on pages 22, 28, 31, 34, 35, 37, 61, 75).

[174] B. R. Conway, 'The organization and operation of inferior temporal cortex,' *Annual Review of Vision Science*, vol. 4, no. 1, pp. 381–402, Sep. 15, 2018, ISSN: 2374-4642. DOI: 10.1146/annurev-vision-091517-034202 (cited on pages 22, 28, 31, 34, 75).

[175] L. C. Greig, M. B. Woodworth, M. J. Galazo, H. Padmanabhan, and J. D. Macklis, 'Molecular logic of neocortical projection neuron specification, development and diversity,' *Nature Reviews Neuroscience*, vol. 14, no. 11, pp. 755–769, Nov. 2013, ISSN: 1471-003X. DOI: 10.1038/nrn3586 (cited on pages 22, 31–34, 38, 51, 74, 75).

[176] A. Wagner, 'Robustness in natural systems and self-organization,' in *Robustness and Evolvability in Living Systems*, Princeton, NJ, USA: Princeton University Press, Dec. 31, 2013, pp. 297–309, ISBN: 978-1-4008-4938-3. DOI: 10.1515/9781400849383.297 (cited on pages 23, 40).

[177] S. Wolfram, 'Cellular automata as models of complexity,' *Nature*, vol. 311, no. 5985, pp. 419–424, Oct. 1984, ISSN: 0028-0836. DOI: 10.1038/311419a0 (cited on page 24).

[178] G. Y. Vichniac, 'Simulating physics with cellular automata,' *Physica D: Nonlinear Phenomena*, vol. 10, no. 1, pp. 96–116, Jan. 1984, ISSN: 0167-2789. DOI: 10.1016/0167-2789(84)90253-7 (cited on page 24).

[179] N. H. Wulff and J. A. Hertz, 'Learning cellular automaton dynamics with neural networks,' in *Proceedings of the 5th International Conference on Neural Information Processing Systems*, ser. NIPS'92, vol. 5, Denver, CO, USA: Morgan-Kaufmann, 1992, pp. 631–638 (cited on page 24).

[180] W. Gilpin, 'Cellular automata as convolutional neural networks,' *Physical Review*, vol. 100, no. 3, p. 032 402, Sep. 4, 2019, ISSN: 2470-0045. DOI: 10.1103/PhysRevE.100.032402 (cited on page 24).

[181] A. Mordvintsev, E. Randazzo, E. Niklasson, and M. Levin, 'Growing neural cellular automata,' *Distill*, vol. 5, no. 2, Feb. 11, 2020, ISSN: 2476-0757. DOI: 10.23915/distill.00023 (cited on page 24).

[182] A. Mordvintsev, E. Randazzo, and C. Fouts, 'Growing isotropic neural cellular automata,' in *The 2022 Conference on Artificial Life*, ser. ALIFE'22, Online: MIT Press, 2022, p. 65. DOI: 10.1162/isal_a_00552 (cited on page 24).

[183] R. B. Palm, M. González-Duque, S. Sudhakaran, and S. Risi, *Variational neural cellular automata*, Feb. 2, 2022. DOI: 10.48550/arXiv.2201.12360. [Online]. Available: http://arxiv.org/abs/2201.12360 (visited on 07/03/2023) (cited on page 24).

[184] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, Dec. 10, 2022. DOI: 10.48550/arXiv.1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114 (visited on 07/03/2023) (cited on page 24).

[185] S. Sudhakaran, D. Grbic, S. Li, A. Katona, E. Najarro, C. Glanois, and S. Risi, *Growing 3d artefacts and functional machines with neural cellular automata*, Jun. 4, 2021. DOI: 10.48550/arXiv.2103.08737. [Online]. Available: http://arxiv.org/abs/2103.08737 (visited on 07/03/2023) (cited on page 24).

[186] K. Horibe, K. Walker, and S. Risi, 'Regenerating soft robots through neural cellular automata,' in *Genetic Programming*, vol. 12691, Cham, Switzerland: Springer, 2021, pp. 36–50, ISBN: 978-3-030-72811-3. DOI: 10.1007/978-3-030-72812-0_3 (cited on page 24).

[187] D. Grattarola, L. Livi, and C. Alippi, 'Learning graph cellular automata,' in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NeurIPS'21, vol. 34, Online: Curran Associates, Inc., 2021, pp. 20 983–20 994 (cited on page 24).

[188] E. Randazzo, A. Mordvintsev, E. Niklasson, M. Levin, and S. Greydanus, 'Self-classifying MNIST digits,' *Distill*, vol. 5, no. 8, Aug. 27, 2020, ISSN: 2476-0757. DOI: 10.23915/distill.00027.002 (cited on page 24).

[189] E. Najarro and S. Risi, 'Meta-learning through hebbian plasticity in random networks,' in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NeurIPS'20, vol. 33, Vancouver, Canada: Curran Associates, Inc., 2020, pp. 20 719–20 731 (cited on page 24).

[190] J. W. Pedersen and S. Risi, 'Evolving and merging hebbian learning rules: Increasing generalization by decreasing the number of rules,' in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO'21, Lille, France: ACM, Jun. 26, 2021, pp. 892–900, ISBN: 978-1-4503-8350-9. DOI: 10.1145/3449639.3459317 (cited on page 24).

[191] L. Kirsch and J. Schmidhuber, 'Meta learning backpropagation and improving it,' in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NeurIPS'21, vol. 34, Online: Curran Associates, Inc., 2021, pp. 14 122–14 134 (cited on page 24).

[192] T. Kohonen, 'Self-organized formation of topologically correct feature maps,' *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982, ISSN: 0340-1200. DOI: 10.1007/BF00337288 (cited on page 24).

[193] T. Kohonen, *Self-Organization and Associative Memory*, 3rd. Heidelberg, Germany: Springer Berlin Heidelberg, 1989, ISBN: 978-3-642-88163-3 (cited on page 24).

[194] S. Grossberg and N. A. Schmajuk, 'Neural dynamics of adaptive timing and temporal discrimination during associative learning,' *Neural Networks*, vol. 2, no. 2, pp. 79–102, Jan. 1989, ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90026-9 (cited on page 25).

[195] D. L. Reilly, L. N. Cooper, and C. Elbaum, 'A neural model for category learning,' *Biological Cybernetics*, vol. 45, no. 1, pp. 35–41, Aug. 1982, ISSN: 0340-1200. DOI: 10.1007/BF00387211 (cited on page 25).

[196] B. Fritzke, 'Growing cell structures: A self-organizing network for unsupervised and supervised learning,' *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, Jan. 1994, ISSN: 0893-6080. DOI: 10.1016/0893-6080(94)90091-4 (cited on page 25).

[197] S. Marsland, J. Shapiro, and U. Nehmzow, 'A self-organising network that grows when required,' *Neural Networks*, vol. 15, no. 8, pp. 1041–1058, Oct. 2002, ISSN: 0893-6080. DOI: 10.1016/S0893-6080(02)00078-3 (cited on page 25).

[198] D. Fasoli. 'The human visual system,' Artificial Intelligence. (2023), [Online]. Available: https://www.neuroinformatics.it/ai/ (visited on 07/14/2023) (cited on page 27).

[199] M. A. Goodale and A. Milner, 'Separate visual pathways for perception and action,' *Trends in Neurosciences*, vol. 15, no. 1, pp. 20–25, Jan. 1992, ISSN: 0166-2236. DOI: 10.1016/0166-2236(92)90344-8 (cited on pages 28, 31, 32, 37, 75).

[200] C. L. Colby and M. E. Goldberg, 'Space and attention in parietal cortex,' *Annual Review of Neuroscience*, vol. 22, no. 1, pp. 319–349, Mar. 1999, ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.22.1.319 (cited on page 28).

[201] H. Liang, X. Gong, M. Chen, Y. Yan, W. Li, and C. D. Gilbert, 'Interactions between feedback and lateral connections in the primary visual cortex,' *Proceedings of the National Academy of Sciences*, vol. 114, no. 32, pp. 8637–8642, Aug. 8, 2017, ISSN: 0027-8424. DOI: 10.1073/pnas.1706183114 (cited on pages 28, 30, 31, 33–35, 75).

[202] D. D. Stettler, A. Das, J. Bennett, and C. D. Gilbert, 'Lateral connectivity and contextual interactions in macaque primary visual cortex,' *Neuron*, vol. 36, no. 4, pp. 739–750, Nov. 2002, ISSN: 0896-6273. DOI: 10.1016/S0896-6273(02)01029-2 (cited on pages 28–31, 33–35, 39, 75).

[203] H. Tanigawa, Q. Wang, and I. Fujita, 'Organization of horizontal axons in the inferior temporal cortex and primary visual cortex of the macaque monkey,' *Cerebral Cortex*, vol. 15, no. 12, pp. 1887–1899, Dec. 1, 2005, ISSN: 1047-3211. DOI: 10.1093/cercor/bhi067 (cited on pages 28, 31, 32, 34, 51, 74, 75).

[204] M. Oberlaender, R. T. Narayanan, R. Egger, H. Meyer, L. Baltruschat, V. Dercksen, R. Bruno, C. P. J. De Kock, and B. Sakmann, 'Beyond the cortical column: Structural organization principles in rat vibrissal cortex,' in *Proceedings of the 5th Congress of Neuroinformatics*, ser. INCF'12, Munich, Germany: Frontiers Research Foundation, 2012 (cited on page 28).

[205] K. L. Narr, R. P. Woods, P. M. Thompson, P. Szeszko, D. Robinson, T. Dimtcheva, M. Gurbani, A. W. Toga, and R. M. Bilder, 'Relationships between IQ and regional cortical gray matter thickness in healthy adults,' *Cerebral Cortex*, vol. 17, no. 9, pp. 2163–2171, Sep. 1, 2007, ISSN: 1047-3211. DOI: 10.1093/cercor/bhl125 (cited on page 28).

[206] L. Pessoa, 'Understanding brain networks and brain organization,' *Physics of Life Reviews*, vol. 11, no. 3, pp. 400–435, Sep. 2014, ISSN: 1571-0645. DOI: 10.1016/j.plrev.2014.03.005 (cited on pages 30, 39).

[207] C. H. Anderson and D. C. van Essen, 'Shifter circuits: A computational strategy for dynamic aspects of visual processing.,' *Proceedings of the National Academy of Sciences*, vol. 84, no. 17, pp. 6297–6301, Sep. 1987, ISSN: 0027-8424. DOI: 10.1073/pnas.84.17.6297 (cited on pages 32, 54, 55, 78).

[208] B. A. Olshausen, C. H. Anderson, and D. C. Van Essen, 'A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information.,' *Journal of Neuroscience*, vol. 13, no. 11, pp. 4700–4719, 1993, ISSN: 0270-6474 (cited on pages 32, 54, 55, 78).

[209] J. Zhu and C. von der Malsburg, 'Maplets for correspondence-based object recognition,' *Neural Networks*, vol. 17, no. 8, pp. 1311–1326, Oct. 2004, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2004.06.010 (cited on pages 32, 51).

[210] *The Oxford handbook of invertebrate neurobiology*, New York, NY, USA: Oxford University Press, 2019, ISBN: 978-0-19-045675-7 (cited on page 32).

[211] C. Bundesen and A. Larsen, 'Visual transformation of size.,' *Journal of Experimental Psychology: Human Perception and Performance*, vol. 1, no. 3, pp. 214–220, 1975, ISSN: 1939-1277. DOI: 10.1037/0096-1523.1.3.214 (cited on page 33).

[212] P. Jolicoeur, 'The time to name disoriented natural objects,' *Memory & Cognition*, vol. 13, no. 4, pp. 289–303, Jul. 1985, ISSN: 0090-502X. DOI: 10.3758/BF03202498 (cited on page 33).

[213] R. Lawson and P. Jolicoeur, 'The effect of prior experience on recognition thresholds for plane-disoriented pictures of familiar objects,' *Memory & Cognition*, vol. 27, no. 4, pp. 751–758, Jul. 1999, ISSN: 0090-502X. DOI: 10.3758/BF03211567 (cited on page 33).

[214] M. Kusunoki and M. E. Goldberg, 'The time course of perisaccadic receptive field shifts in the lateral intraparietal area of the monkey,' *Journal of Neurophysiology*, vol. 89, no. 3, pp. 1519–1527, Mar. 1, 2003, ISSN: 0022-3077. DOI: 10.1152/jn.00519.2002 (cited on page 33).

[215] T. Womelsdorf, K. Anton-Erxleben, F. Pieper, and S. Treue, 'Dynamic shifts of visual receptive fields in cortical area MT by spatial attention,' *Nature Neuroscience*, vol. 9, no. 9, pp. 1156–1160, Sep. 2006, ISSN: 1097-6256. DOI: 10.1038/nn1748 (cited on page 33).

[216] B. Widrow, Y. Kim, D. Park, and J. K. Perin, 'Nature's learning rule,' in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, New York, NY, USA: Elsevier, 2019, pp. 1–30, ISBN: 978-0-12-815480-9. DOI: 10.1016/B978-0-12-815480-9.00001-3 (cited on page 33).

[217] L. Schmarje, M. Santarossa, S.-M. Schroder, and R. Koch, 'A survey on semi-, self- and unsupervised learning for image classification,' *IEEE Access*, vol. 9, pp. 82 146–82 168, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3084358 (cited on pages 34, 55).

[218] C. G. Gross, 'Genealogy of the "grandmother cell",' *The Neuroscientist*, vol. 8, no. 5, pp. 512–518, Oct. 2002, ISSN: 1073-8584. DOI: 10.1177/107385802237175 (cited on pages 34, 61).

[219] P. Iamshchinina, D. Kaiser, R. Yakupov, D. Haenelt, A. Sciarra, *et al.*, 'Perceived and mentally rotated contents are differentially represented in cortical depth of v1,' *Communications Biology*, vol. 4, no. 1, p. 1069, Sep. 14, 2021, ISSN: 2399-3642. DOI: 10.1038/s42003-021-02582-4 (cited on page 36).

[220] D. Gabor, 'Theory of communication: The analysis of information,' *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, Nov. 1946, ISSN: 2054-0604. DOI: 10.1049/ji-3-2.1946.0074 (cited on pages 38, 43).

[221] G. H. Granlund, 'In search of a general picture processing operator,' *Computer Graphics and Image Processing*, vol. 8, no. 2, pp. 155–173, Oct. 1978, ISSN: 0146-664X. DOI: 10.1016/0146-664X(78)90047-3 (cited on pages 38, 43).

[222] Z. Niu, G. Zhong, and H. Yu, 'A review on the attention mechanism of deep learning,' *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2021.03.091 (cited on page 40).

[223] M. Rhu, M. O'Connor, N. Chatterjee, J. Pool, Y. Kwon, and S. W. Keckler, 'Compressing DMA engine: Leveraging activation sparsity for training deep neural networks,' in *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, ser. HPCA'18, Vienna, Austria: IEEE, Feb. 2018, pp. 78–91, ISBN: 978-1-5386-3659-6. DOI: 10.1109/HPCA.2018.00017 (cited on page 40).

[224] E. R. Kandel, *Principles of neural science*, 5th. New York, NY, USA: McGraw-Hill, 2013, ISBN: 978-0-07-139011-8 (cited on pages 40, 49, 70).

[225] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, Jul. 3, 2012. DOI: 10.48550/arXiv.1207.0580. [Online]. Available: http://arxiv.org/abs/1207.0580 (visited on 07/10/2023) (cited on page 41).

[226] A. van den Oord, O. Vinyals, and k. kavukcuoglu koray, 'Neural discrete representation learning,' in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, vol. 30, Long Beach, CA, USA: Curran Associates, Inc., 2017, pp. 6309–6318 (cited on page 43).

[227] A. Revonsuo and J. Newman, 'Binding and consciousness,' *Consciousness and Cognition*, vol. 8, no. 2, pp. 123–127, Jun. 1999, ISSN: 1053-8100. DOI: 10.1006/ccog.1999.0393 (cited on page 51).

[228] J. Feldman, 'The neural binding problem(s),' *Cognitive Neurodynamics*, vol. 7, no. 1, pp. 1–11, Feb. 2013, ISSN: 1871-4080. DOI: 10.1007/s11571-012-9219-8 (cited on page 51).

[229] F. Simion and E. Di Giorgio, 'Face perception and processing in early infancy: Inborn predispositions and developmental changes,' *Frontiers in Psychology*, vol. 6, p. 969, Jul. 9, 2015, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2015.00969 (cited on page 51).

[230] L. Wiskott, 'The role of topographical constraints in face recognition,' *Pattern Recognition Letters*, vol. 20, no. 1, pp. 89–96, Jan. 1999, ISSN: 0167-8655. DOI: 10.1016/S0167-8655(98)00122-6 (cited on page 52).

[231] X.-F. Han, H. Laga, and M. Bennamoun, 'Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1578–1604, May 1, 2021, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2954885 (cited on page 55).

[232] L. Qu, S. Liu, M. Wang, and Z. Song, 'TransMEF: A transformer-based multi-exposure image fusion framework using self-supervised multi-task learning,' in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, ser. AAAI'22, vol. 36, Online: AAAI Press, Jun. 28, 2022, pp. 2126–2134 (cited on page 55).

[233] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, 'Object detection in 20 years: A survey,' *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023, ISSN: 0018-9219. DOI: 10.1109/JPROC.2023.3238524 (cited on page 58).

[234] T. Miconi, *Hebbian learning with gradients: Hebbian convolutional neural networks with modern deep learning frameworks*, Nov. 1, 2021. [Online]. Available: http://arxiv.org/abs/2107.01729 (visited on 07/11/2023) (cited on page 60).

[235] H. Tomita, M. Ohbayashi, K. Nakahara, I. Hasegawa, and Y. Miyashita, 'Top-down signal from prefrontal cortex in executive control of memory retrieval,' *Nature*, vol. 401, no. 6754, pp. 699–703, Oct. 1999, ISSN: 0028-0836. DOI: 10.1038/44372 (cited on page 61).

[236] P. Smolensky, 'Information processing in dynamical systems: Foundations of harmony theory,' in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, Cambridge, MA, USA: MIT Press, 1986, pp. 194–281, ISBN: 0-262-68053-X (cited on page 61).

[237] T. Isomura, K. Kotani, Y. Jimbo, and K. J. Friston, 'Experimental validation of the free-energy principle with in vitro neural networks,' *Nature Communications*, vol. 14, no. 1, p. 4547, Aug. 7, 2023, ISSN: 2041-1723. DOI: 10.1038/s41467-023-40141-z (cited on page 62).

[238] J. Hui. 'Machine learning - restricted boltzmann machines,' Jonathan Hui blog. (Jan. 15, 2017), [Online]. Available: https://jhui.github.io/2017/01/15/Machine-learning-Boltzmann-machines/ (visited on 07/25/2023) (cited on page 62).

[239] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, *et al.*, 'ImageNet large scale visual recognition challenge,' *International Journal of Computer Vision*, IJCV'15, vol. 115, no. 3, pp. 211–252, Dec. 2015, ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y (cited on page 79).

[240] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, 'Multimodal deep learning,' in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11, vol. 25, Bellevue, DC, USA: PMLR, 2011, pp. 689–696, ISBN: 978-1-4503-0619-5 (cited on page 79).

[241] K. Liu, Y. Li, N. Xu, and P. Natarajan, *Learn to combine modalities in multimodal deep learning*, May 29, 2018. DOI: 10.48550/arXiv.1805.11730. [Online]. Available: http://arxiv.org/abs/1805.11730 (visited on 07/07/2023) (cited on page 79).

[242] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, 'Multimodal machine learning: A survey and taxonomy,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, Feb. 1, 2019, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2798607 (cited on page 79).