RESEARCH

Richard Meyes

Transparency and Interpretability for Learned Representations of Artificial Neural Networks

MOREMEDIA

Springer Vieweg

Transparency and Interpretability for Learned Representations of Artificial Neural Networks

Richard Meyes

Transparency and Interpretability for Learned Representations of Artificial Neural Networks

Springer Vieweg

Dr.-Ing. Richard Meyes, M.Sc.
Research Group Lead “Interpretable Learning Models”
Institute for Technologies and Management of Digital Transformation
University of Wuppertal
Wuppertal, Germany

The scientific studies and corresponding results presented in this book were conducted within the framework of a doctoral project to obtain the degree of Dr.-Ing. at the School of Electrical, Information and Media Engineering at the University of Wuppertal.

ISBN 978-3-658-40003-3 ISBN 978-3-658-40004-0 (eBook)
https://doi.org/10.1007/978-3-658-40004-0

This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH, part of Springer Nature.
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Men ought to know that from the brain, and from the brain only, arise our pleasures, joys, laughter, and jests, as well as our sorrows, pains, griefs, and tears. Through it, in particular, we think, see, hear, and distinguish the ugly from the beautiful, the bad from the good, the pleasant from the unpleasant, in some cases using custom as a test, in others perceiving them from their utility. It is the same thing which makes us mad or delirious, inspires us with dread or fear, whether by night or by day, brings sleeplessness, inopportune mistakes, aimless anxieties, absent-mindedness, and acts that are contrary to habit. These things that we suffer all come from the brain, when it is not healthy, but becomes abnormally hot, cold, moist, or dry, or

suffers any other unnatural affection to which it was not accustomed. Madness comes from its moistness. When the brain is abnormally moist, of necessity it moves, and when it moves neither sight nor hearing are still, but we see or hear now one thing and now another, and the tongue speaks in accordance with the things seen and heard on any occasion. But all the time the brain is still a man is intelligent.
— Hippocrates

The Sacred Disease, in Hippocrates, trans. W. H. S. Jones (1923), Vol. 2, 175

Contents

List of Figures

List of Tables

Introduction

1

*"I know that I am **intelligent** because I know that I know nothing."*

– Socrates, ancient Greek philosopher.

Artificial intelligence (AI) is a concept, whose meaning and perception has changed considerably over the last decades. Starting off with individual and purely theoretical research efforts in the 1950s, AI has grown into a fully developed research field of modern times and may arguably emerge as one of the most important technological advancements of mankind. Outside of the academic research domain, initially, AI has been predominantly portrayed in film and fiction before it gained significant importance for practical applications in industry and society. Amongst other things, this change can be greatly attributed to the digital transformation in the last years, during which the availability of computational resources to handle growing amounts of collected data has greatly increased and has made the application of AI in the form of artificial neural networks as black-box models, on large scales economically feasible. Despite the rapid technological advancements of these AI black-box models in the second decade of the twenty-first century, some key questions about the very nature of their decision-making remain unanswered. Why does an AI decide the way it does? What is the basis for its decision-making? Why was this basis learned to be important from the presented data? To what extent is this basis compatible with human concepts of morality and ethics and how can an AI's decision-making be deliberately influenced to assure such compatibility? All these questions revolve around the matter of transparency, interpretability and explainability of an AI's decision-making, a

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-658-40004-0_1.

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_1

young research field, coined with the general term XAI, which is emerging from increasingly strict requirements for AI to be used in safety critical or ethically sensitive domains.

The remainder of this introduction details the objects of investigation of this thesis and phrases four distinct research questions, the answer to which contribute towards the current state of research in the field of XAI with respect to two specific goals. The first goal is to bridge the two research fields of neuroscience and AI to utilize neuroscientific methods to facilitate transparency, interpretability and explainability of AI black-box models. The second goal is to transfer these methods to real-world scenarios in the industrial domain to exploit the increased degree of transparency, interpretability and explainability of AI black-box models to make them applicable and trustworthy for domain experts as well as to gain a deeper understanding of the industrial process that integrates these models. Finally, the structure of the thesis is illustrated as well as the structure of the research studies and their relation to the research questions.

Before continuing with the technical part of the conducted research, the philosophically keen reader may take a short detour to the first sub-chapter of the electronic supplementary material *"A Philosophical Viewpoint on the Concept of Intelligence"*, a brief philosophical excursion to explore some fundamental questions about the very nature of humanity and intelligence itself, to start with a non-technical introduction to the general topic of intelligence, whether human or artificial.

1.1 Object of Investigation

*"The mind that opens to a **new idea** never returns to its original size."*

– Albert Einstein, German Nobel laureate physicist.

The raised research questions, the conducted studies, and the reported results in this thesis are aimed to contribute to the broader goal of paving the way towards a **new perspective** on AI by bridging the two research fields of XAI and neuroscience. Following the advancements of AI research and the overall goal of understanding intelligence, this thesis contributes to said goal by investigating learned representations of artificial neural networks in the same spirit as neuroscientists have studied representations in the mammalian brain [1–6]. This spirit is largely driven by empirical approaches studying neural brain activity of mammals solving specific tasks in specific domains, such as vision, language generation, or motor control. Fortunately, all the necessary considerations regarding ethics

of experimental designs in neuroscientific studies are easily dismissed as they do not apply to the investigations of artificial neural networks. Thus, the transfer of such neuroscientific methods to the field of AI allows to retain the benefits, which is a highly advanced set of methods and approaches to deal with large neural systems, and resolve the challenges, the experimental and ethical considerations. In similar spirit to neuroscientific studies, the conducted studies in this thesis focus on the investigation of the learned representations of artificial neural networks that are trained to solve specific tasks in the domains of computer vision and continuous motor control to establish a foundation for a set of methods that are usable to facilitate transparency and interpretability of AI black-box models. Subsequently, these established methods used to characterize the learned representation of AI black-box models are transferred from the research domain, characterized by well-defined and manually constructed benchmark datasets and learning scenarios, to an application domain in the industrial field, to demonstrate the value of these methods to facilitate transparency and interpretability in a real-world data-driven scenario. The particular scenario chosen in this thesis constitutes a manufacturing process of car body parts, which serves as a designated proof-of-concept scenario to demonstrate the applicability of the developed approach towards facilitating transparency and interpretability of an artificial neural network's decision-making for real-world problems outside of the well-defined realms of fundamental research. It demonstrates how the improved degree of transparency and interpretability of the trained network's decision-making yields new insights and better understanding of the process itself, helping to further improve upon it.

1.2 Research Questions

*"**Research is formalized curiosity.** It is poking and prying with a purpose."*

– Zora Neale Hurston, American writer

The four **research questions** guiding the investigations presented in this thesis follow a bottom-up research approach addressing the investigability of learned representations of neural networks, the structure and organization of these learned representations, the relation between specific characteristics of these representations and individual network compartments, like individual neurons or layers, and the applicability of these methods and approaches for real-world data-driven scenarios in the industrial field. The first three research questions are aimed to

establish a methodical foundation for the investigation of learned representations while the fourth question aims to transfer these methods into the industrial domain. In the following, the research questions are explicitly stated, and additional explanations are given as to why these questions are sensible to ask with respect to the goal of this thesis.

RQ 1 How to determine the importance of individual neurons and groups of neurons for a network's learned ability to solve a specific task?

In bottom-up fashion, the first question starts with the fundamental building block of neural networks, the neuron. Specifically, it addresses the relation of individual neurons and functional populations of neurons of a trained neural network to the network's capabilities to solve a given task. The question is largely inspired by the nature of neuroscientific research and findings about the role of single neurons in the mammalian brain for specific tasks. For instance, individual neurons in the motor cortex show a high selectivity in their activity depending on the direction in which a subject's hand is moved. Specifically, individual neurons would only show strong activity when the hand is moved to the left side, while other neurons would only show strong activity when the hand is moved to the rights side. In similar fashion, the first question aims to uncover similar findings in artificial neural networks and to investigate the transferability of neuroscience inspired methods to artificial neural networks.

RQ 2 How to characterize the structure and organization of a neural network's learned representation qualitatively and quantitatively?

The second research question directly builds on the findings connected with answering the first research question. Given the ability to determine the importance of single neurons and groups of neurons for a network's task and corresponding sub-tasks, the question arises whether these neurons and groups of neurons can be structured, organized, and categorized. To investigate such structure and organization, the neurons of a neural network, which are the foundation of the network and its learned representation, must be characterized qualitatively and quantitatively in order to characterize the network's learned representation accordingly. Thus, the second question is aimed to provide a number of suitable methods to extract learned representations from trained neural networks based on their accessible parameters, i.e., their weights and activations. Besides the inherent relation of a trained network's weights to its representation of a given learning problem, its activations, which are the result of processing input data

with the learned parameterization, i.e., the learned set of weights, are closely related and equally interesting when investigating such representations. Similar to the first research question, the general idea of the second question is stems from a perspective from neuroscience research that is largely inspired by well-studied analogues organizations of the mammalian brain. One such organization, for example, can be found in the simple ability of the mammalian visual system to detect edges with different orientations, which are carefully mapped onto the surface of the neocortex. Thus, the learned representation of the external world that continuously stimulates the visual system is organized in a similar manner and strongly depends on the structure and organization of these external stimuli. In analogous fashion, the second research question addresses the existence of such structure and organization within learned representations of artificial neural networks. The key challenge in defining structure and organization is given by the absence of a given relation between individual neurons that provides a ground truth to which the activation of the neural system can be related to. More specifically, in case of biological neural systems, this ground truth is given by the anatomy of the system and the spatial relationship between individual areas and neurons. In such systems, neural activations can be analyzed with respect to these spatial relationships, for example, so that an organized way of how neural activation spreads spatially across the network can be analyzed in a meaningful way. However, due to the absence of such a straightforward ground truth for artificial neural networks, a major challenge to overcome answering the second research question is to determine such structure and organization in order to relate neural activity of the network to it.

RQ 3 How to determine the relation between the structure and organization of a neural network's learned representation and its emerging behavior?

The third question effectively builds upon the results of the first two questions as it presupposes a) the existence of specific roles of individual neurons and groups of neurons for a learned task and b) the organization of those neurons into distinct groups for sub-tasks of the overall task. Based on these two presuppositions, the third question aims to investigate the relation between the activity of these organized neurons and the exhibited behavior of an agent controlled by the network that possesses said structured and organized representation. The third question is again strongly inspired by findings of neuroscientific studies that related the activation of spatially organized neural subsystems, like the motor cortex, to the exhibition of specific behavioral routines, like moving individual

fingers or moving a hand into a specific direction. Answering this third question aims to investigate whether similar phenomena can be observed in artificial neural systems trained to perform motor control tasks.

RQ 4 How to utilize the investigation of structured and organized learned representations to facilitate transparency in industrial, data-driven real-world processes?

The fourth question unites the insights gained from the first three research questions and addresses the transfer of the previously employed methods and the obtained results for the application in industrial use cases and data driven manufacturing processes. Especially in the context of XAI and the growing requirement for artificial learning systems to be transparent and interpretable when applied in real world scenarios gives rise to the notion that neuroscientific methods can be used beneficially. Considering that neuroscience, isn't primarily concerned with building biological learning systems in the first place, since they are already given, but is rather concerned with explaining how these systems work and with uncovering underlying mechanisms for how they learn, it seems sensible to investigate whether methods and approaches that have been developed to reach that goal can be transferred to artificial neural systems. Answering the fourth question aims to investigate how understanding the learned representations of neural networks trained to perform predictions in data driven manufacturing processes helps to facilitate a better understanding of the process, complex relationships between data sources and important key performance indicators like product quality.

1.3 Structure of the Thesis

*"Good **order** is the foundation of all things."*

– Edmund Burke, Irish-British writer

The thesis consists of eight chapters structuring the common thread in sequential **order** as well as the electronic supplementary material with additional information regarding specific chapters. The additional sub-chapters in the electronic supplementary material are not absolutely necessary for the comprehension of this thesis, but nevertheless, provide interesting extensions and details to some specific points made throughout the main text. Furthermore, the electronic supplementary material contains additional figures, which in addition to the results

presented throughout the main text do not constitute new insights but rather reinforce the presented results and are thus moved into the electronic supplementary material to provide the required brevity of the main text and improve its clarity with respect to the objective of the thesis.

After chapter one, the introduction to this thesis, chapter two provides the necessary background on the topic on AI and the historic developments of AI research that lead up to the current questions on transparency, interpretability, and explainability of AI. Chapter three provides a brief overview of the relevant methods used throughout the presented research and transfer studies and ends with a further delimitation of the object of investigation. It serves as the backbone of foundations that leads up to the four research questions presented above. Chapter four presents the related work of the current field of research on explainable AI and previous contributions that are closely related to the general topic of the thesis. The presented related work is structured into three sub-chapters categorizing the recent advances on explainable AI into contributions on 1) the relationship between a network's input variables and its output, 2) visualizations of a network's activation during information processing, 3) investigations of a network's individual components and their role within the whole network. The last sub-chapter is closely related to the contributions of this thesis and ends with presenting the research gap of the current state of the art that is addressed in chapter four. Chapter five, the first part of the main contribution of the thesis, presents a series of five empirical research studies addressing the first three research questions. These studies are conducted in the realms of the two well-established research domains of computer vision and motor control. The results establish a methodical foundation that is aimed to be transferred into the industrial domain in the next chapter. Chapter six, the second part of the main contribution of the thesis, builds on the results of the research studies to address the fourth research question in two transfer studies to demonstrate the transferability of the established methods from the research domains of computer vision and motor control to the industrial domain aiming to investigate the usability of facilitating transparency in neural networks for a real-world application scenario. Chapter seven critically reflects on the presented results, achievements, and limits of the thesis and gives a brief outlook on possible future research directions building on the core results of the thesis. Finally, chapter eight concludes the thesis with a short summary.

Figure 1.1 illustrates the sequential structure of the thesis, gives an overview of the individual chapters and illustrates the study design and the relation of the individual studies to the research questions.

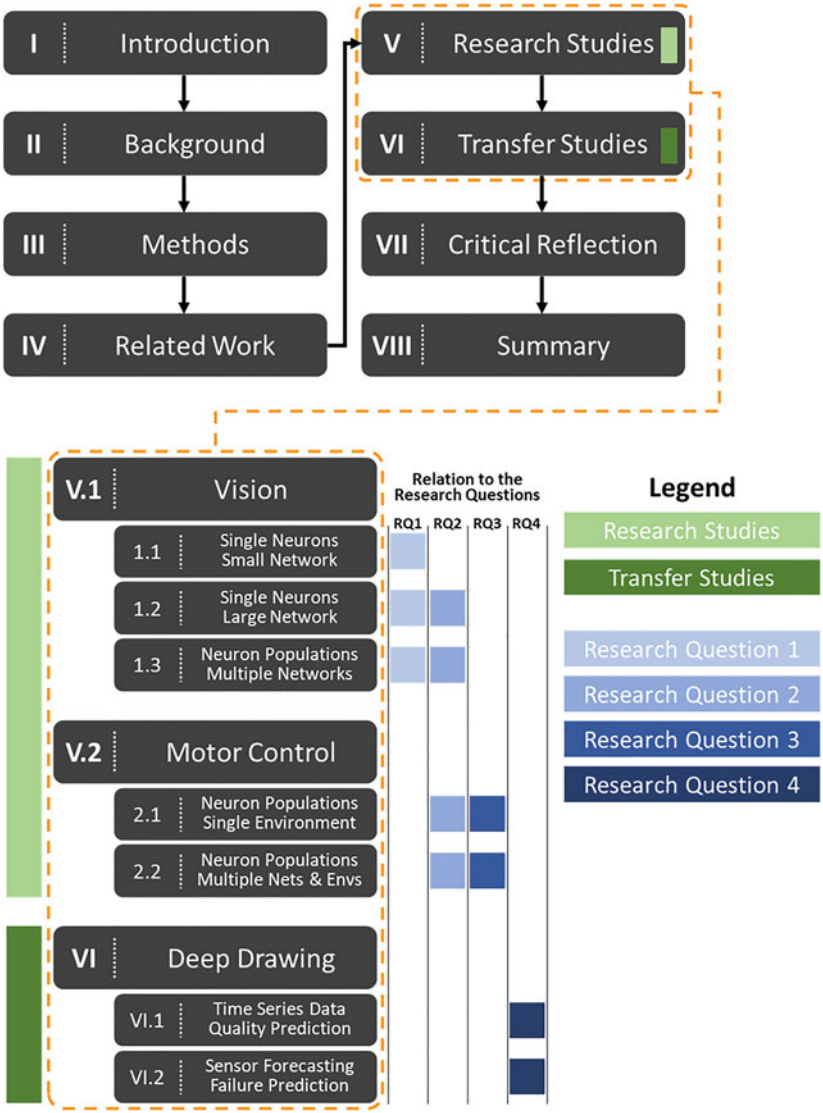


Figure 1.1 Schematic illustration of the sequential structure of the thesis, the study design, and the relation of the individual studies to the research questions

Background & Foundations

2

"One must have first of all a ***solid foundation.****"*

– Sri Aurobindo, Indian philosopher

The following chapter gives a brief introduction on the **background** of the past developments of the research field of AI and its major milestones that lead the field from early research in the depths of theoretical computer science into today's modern era of application-oriented computational engineering and thus its path towards the research branch of explainable AI. The discourse on the historical developments of the research field of AI is aimed to illustrate how the need for explainability, transparency and interpretability has emerged and became an increasingly important topic to address in light of the advancements and growing capabilities of artificial neural networks in various domains.

The chapter further provides the **methodological foundation** for the subsequent chapters, the current state of the art of explainable AI and the empirical studies conducted within the framework of this thesis. Building on the given foundation, it concludes with a further delimitation of the object of investigation.

2.1 A Short History of AI Research

"Human history in essence is the ***history of ideas****."*

– Herbert George Wells, English writer & sci-fi literature pioneer

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-658-40004-0_2.

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_2

The research field of AI has gone through a number of **historical developments** and important milestones starting in 1943 with the *early years* that lead up to the *golden ages*, which lasted until the beginning of the *AI winter* in 1969 and the resurrection of the research field in its *renaissance* in 1985, up until the beginning of modern AI research in 2009. The following sub-chapters guide the reader through these milestones and provide additional context and important details with respect to the topic of explainable AI. Figure 2.1 illustrates these historical developments and shows the three distinct eras and their respective milestones.

2.1.1 The Early Years

"A journey of a thousand miles ***begins with a single step.****"*

– Confucius, Chinese philosopher, poet, and politician

The foundation for building an artificial brain, which was essentially the original idea of AI research, was laid in 1943 by Warren McCulloch and Walter Pitts with their description of the infamous McCulloch-Pitts-Neuron [7]. This cell was **the very first and simplified model of an artificial neuron** that was designed to be mimicking real processes in neural structures in order to clarify whether the brain can really compute Turing computable functions. In 1949, Donald Hebb first addressed the interplay of several of such neurons combining to small networks coining the principle of Hebbian learning [8], which can be loosely phrased in terms of Hebb's famous quote "What fires together, wires together". Following this principle, in 1951, Marvin Minsky built the first neurocomputer called "Snarc", which was performing computations based on single cells that are connected to each other via automatically adjustable weights [12]. Although Snarc was designed to simulate the behavior of a mouse navigating through a maze, it did not serve any other practical purpose and could not be used for other practical applications.

Considering these early contributions to the research field of AI and the objective of establishing a new research field at that time, the topic of transparency, interpretability and explainability of AI was not yet a frequently discussed topic since the used methods were based on analytic approaches and mathematically, soundly formalized equations that did not require an additional degree of transparency to explain the results they produced.

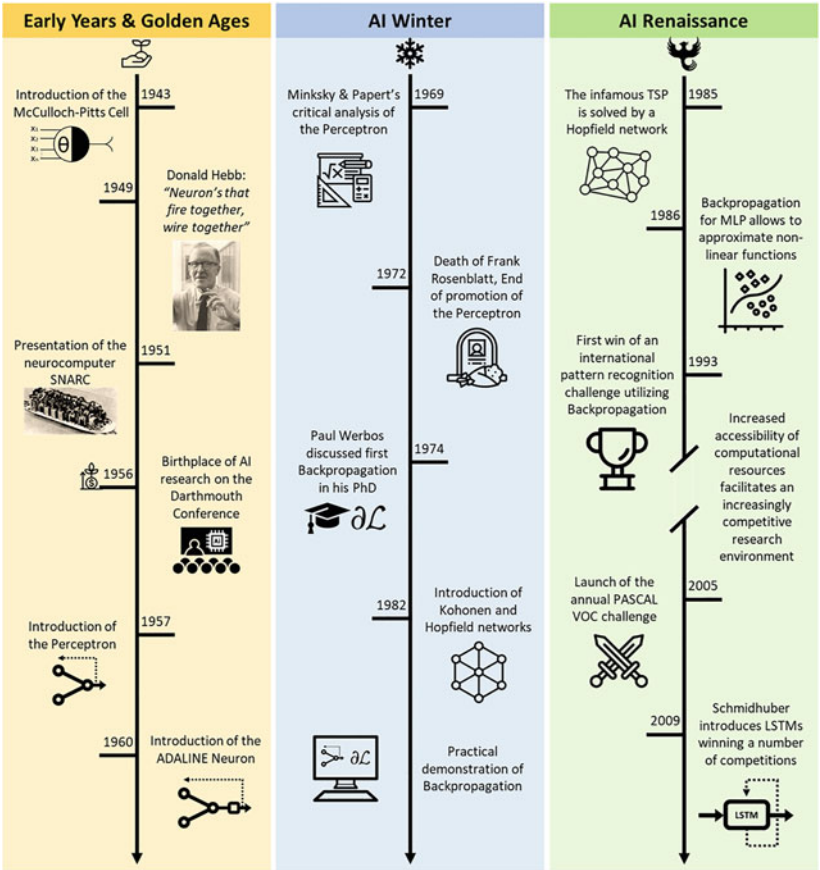


Figure 2.1 Illustration of the history of AI research starting in 1943 going through the golden ages until the beginning of the AI winter in 1969 and the resurrection of the research field in its renaissance in 1985 up until the beginning of modern AI research in 2009. Illustrations taken from [9–11]

2.1.2 The Golden Ages

"An intellectual golden age produces ***sages.****"*

– Jakub Bożydar Wiśniewski, Polish political scientist and diplomat

The early accomplishments of AI research opened up the possibility that fundamental computational tasks could be performed by machines mimicking the functions of the brain, promising a practical access to creating artificial intelligence. This perspective culminated in the Dartmouth Summer Research Project on Artificial Intelligence in 1956, which is widely regarded as **the birthplace of academic AI research** [13]. Inspired by the early successes of McCulloch, Pitts and Minsky, students and researchers gathered from around the world to discuss topics like "autonomous computers", "neural networks", "self-optimization", "abstraction" and "creativity". In 1957, Frank Rosenblatt and Charles Wightman built the Mark I Perceptron, the first neurocomputer with a practical application in the field of early computer vision, which was able to recognize simple digits with a 20×20 pixel sensor. In the following year, Rosenblatt formulated the perceptron convergence theorem, which states that the used learning rule to update the perceptron's weights is powerful enough for the perceptron to learn all possible solutions that it could theoretically represent [14]. In 1960, Bernard Widrow and Marcian E. Hoff presented the ADAptive LInear NEuron (ADALINE), the first neuron model that was used commercially in analog telephones for real-time echo filtering. Building upon the functions of the Perceptron, the ADALINE's learning rule, which is based on the least mean squares algorithm, adjusts weights not just based on the input of the neuron but also based on its output, which facilitates to build more stable knowledge representations preventing the ADALINE from forgetting previously learned knowledge to some extent.

The flourishing research spurred by the developments in the 1950s and 1960s came to a halt in 1969, when Marvin Minsky and Seymour Papert presented their detailed mathematical analysis of the perceptron [15, 16]. They showed that the original Perceptron by Rosenblatt is not able to learn the XOR-operation, one of the most fundamental computational operations of computer science, owing to its inability to approximate non-linear functions. Tragically, shortly after Minsky and Papert's work, Rosenblatt died in a boating accident and was unable to react to their critical publication about the perceptron. Only about 16 years later it should turn out that the perceptron can in fact approximate any non-linear function if it is extended by an additional layer, giving it the name Multi-Layer-Perceptron (MLP).

During the golden ages, the research field of AI was mostly concerned with proof-of-concept demonstrations of AI and artificial neural networks in order to demonstrate its capabilities to perform meaningful computation. At this point however, AI had not reached a level yet, on which its computations and results could not be comprehended by researchers, and most of its capabilities had comparable non-AI solutions that were fully understandable from a mathematical

model's point of view, such as simple logical computations or noise filtering in voice transmission signals of telephone connections.

2.1.3 The AI Winter

*"And **the dancing** were believed crazy by those who could not hear the music."*

– Friedrich Wilhelm Nietzsche, German philosopher and philologist

The fundamental doubts that were cast by Minsky and Paper's work toppled AI research in a deep winter leading to a loss of interest and significant cuts of US-government funding. However, **some more independent researchers continued to contribute valuable research bridging the period towards a brighter future.** One of the most vital contributions came from Paul Werbos in 1974, who first discussed the principle of using the backpropagation algorithm for adjusting the weights in a neural network in his dissertation [17], which, however, turned out to be one of the most important learning methods until today only more than ten years later. Subsequently, in 1982, Teuvo Kohonen and John Hopfield independently describe the Kohonen networks (self-organizing maps) and Hopfield networks, respectively nets named after them. Both networks presented new approaches to learning functions that were previously unexplored and untested. While Kohonen networks are based on learning structure in presented data in an unsupervised fashion, Hopfield networks introduced the first notion of recurrent connections allowing the network to process feedback information in addition to the widely used approach of only processing forward information. Finally, in the same year, Paul Werbos demonstrated the idea from his dissertation practically [18], paving the way out of the cold AI research winter.

2.1.4 The AI Renaissance

*"Bamboo, bent even to the ground, will **spring upright after the passage of the storm.**"*

– Japanese proverb

The **renaissance of AI research** started in 1985 when John Hopfield presented a solution of the infamous Traveling Salesman Problem, a fundamental combinatorial optimization problem of operations research and theoretical computer science, through a Hopfield network. Inspired by the results of Paul Werbos,

four years later in 1986, David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams showed by experiments that training neural networks via backpropagation can lead to useful internal representations of input data in deeper layers of neural networks [19]. They showed that backpropagation allows to train MLPs to approximate non-linear functions, refuting Minsky and Papert's assessment from 1969, which lead to the AI winter in the first place. With this finding, they laid the foundation of Deep Learning research as it is known today. In 1993, Eric A. Wan was the first to win an international pattern recognition competition utilizing backpropagation [20]. The resurrection of artificial neural networks in AI research and their proven usability by Eric Wan lead to the announcement of a series of international pattern recognition challenges further promoting competitive neural network research. One of the first annually organized challenges was the PASCAL VOC Challenge [21], held between 2005 and 2012, which addressed the task of visual object recognition in real world images. The main competitions consisted of an object classification task, answering the question of whether an image contains a specific object, and an object detection task, answering the questions where a specific object is localized in an image.

Although the period of the AI renaissance laid the foundation for the methods to train large neural networks in a way that makes them inherently untransparent and awards them their black-box character, AI researchers were too concerned with reviving the research field and focusing on various applications of the newly researched methods to optimize neural networks for various tasks and domains. Despite the inevitable conclusion that the approach of training neural networks via backpropagation creates system parameterizations, i.e., weight combinations of neural networks, that lead to meaningful and potent behavior, AI researchers were not particularly concerned with exploring the intricacies of these parameterizations. This lack of perspective is likely based on the critical nature of such intentions, that would have emphasized the limitations of the methods used at the time rather than embracing them as the salvation that led the field out of the cold AI winter.

2.2 The Modern Era of AI Research

"I do not fear ***computers.*** *I fear the lack of them."*

– Isaac Asimov, Russian-American biochemist

Starting at around 2008, the growing accessibility of **computational resources** and the development of computational frameworks and business models for

infrastructure as a service allowed a larger number of research facilities and single researchers to build and train neural networks with large amounts of data. Simultaneously, as modern pattern recognition challenges started to pose increasingly complex tasks, classical pattern recognition approaches based on pattern matching methods of handcrafted features were rendered infeasible due to the enormous amount of manual work that was required for manual feature extraction. Consequently, approaches based on utilizing neural networks, that were able to learn features from scratch, proved increasingly powerful to tackle these complex problems and turned out to be the preferred approach to these challenges.

A pioneer on this front was the research group at the Swiss AI laboratory IDSIA around Jürgen Schmidhuber, which won a series of eight international competitions in the field of machine learning and pattern recognition between 2009 and 2012 utilizing deep feed-forward and recurrent neural networks utilizing a special kind of neuron, the Long-Short-Term Memory (LSTM) Cell [22–24]. Due to the intuitive nature of image data, many of these competitions came from the field of computer vision, which mostly deals with pattern recognition problems in real-world images, and thus has emerged as the largest playground for neural networks as the most widely addressed domain in AI research.

The growing competitive nature of AI research has resulted in larger networks and novel network architectures that grew increasingly potent in solving increasingly complex tasks. However, despite spurring great progress due to the competitive environment amongst researchers, which resulted in the promotion of a strong focus on the optimization of model performances and an increase in computational efficiency, an equally strong increase of the attention for explainability, transparency and interpretability for these large and potent models remained absent.

2.2.1 Deep Learning for Computer Vision

*"An **image** is worth a thousand words."*

– English language adage

Spurred by the increasingly competitive nature of AI research, in 2010, the first ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was held, a competition evaluating algorithms for **image classification** and **object detection in images** at large scale, which besides enabling a big leap in the development of

neural networks for computer vision, played an important role to lay the foundation for the research of XAI. The high level motivation of the ILSVRC was to allow researchers to compare progress in a variety of recognition, localization, detection and masking tasks across a wide variety of object classes, taking advantage of the quite expensive labeling effort that has been made on the famous ImageNet dataset [25]. The ImageNet dataset contains more than 14 million images from 20.000 different, partly overlapping categories that have been manually labeled by Amazon's Mechanical Turk service. The ILSVRC uses a subset of the original ImageNet dataset, that contains about 1.2 million images from 1.000 non-overlapping categories.

Although Convolutional Neural Networks (CNNs), a specialized network architecture that deliberately exploits the spatial arrangement of information in images, are known since 1989 when first introduced by Yann LeCun [26], they caught attention of computer vision researchers only in 2012 when Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton introduced *AlexNet*, the first CNN that won the ILSVRC 2012 convincingly by a huge margin, achieving a top-5 error rate of 15.4% compared to the second placed 26.2% [27]. Its success was only possible by outsourcing the training procedure of the network to two GPUs, which required the development of a specialized GPU-accelerated version of the backpropagation algorithm and the split of the network into two separate processing streams, one for each GPU. Additionally, a novel training technique called Dropout was introduced, which helped to tackle the problem of overfitting *AlexNet's* parameters to the provided training dataset. *AlexNet* has started a new era of dominance of CNNs in the computer vision domain and since its first appearance, every winner of the subsequent ILSVRC challenges build upon CNNs and the principles introduced by *AlexNet*.

The ILSVRC has served AI researchers as an excellent playground to develop large and potent models for various computer vision tasks and has pushed the boundaries of neural networks beyond the capabilities of humans when it comes to the processing of large amounts of image data and performing classification tasks accurately. Although the topic of explainability, transparency and interpretability was not addressed within the competition of the ILSVRC, many of the developed models serve todays researchers pursuing that research branch as pre-trained objects of investigation to uncover intricacies and characteristics about their learned representations [28]. For more details on the historical milestones of the ILSVRC between 2013 and 2017, the interested reader may consult the second sub-chapter in the electronic supplementary material, "*Further Progress in*

the ILSVRC since AlexNet", which guides the reader through the continuous developments and iterative improvements on CNNs utilized to push the boundaries on the frontiers of computer vision.

2.2.2 Computer Vision Beyond the ILSVRC

"There is always ***a new challenge*** *to keep you motivated."*

– Sean Connery, British actor, and Oscar winner

Although the task of object detection (recognizing and localizing multiple objects in a single image) had been tackled in the PASCAL VOC challenge, the utilization of neural networks for this task still posed **a daunting challenge** and has not been researched extensively as most of the recent breakthroughs regarding novel and efficient network architectures happened in the course of the ILSVRC after the PASCAL VOC challenge ended in 2012. Up to this point, state-of-the-art object detection methods were "complex ensemble systems combining multiple low-level image features with high-level context", which had plateaued in their performance measured on the PASCAL VOC dataset [29].

The first neural network approach to these tasks was the Region-based CNN (R-CNN) proposed in 2014 by Ross Grishick at al., which improved upon the previous methods (measured on the PASCAL VOC dataset) by more than 30% [29]. In its core, the R-CNN is based on the application of a high-capacity backbone CNN trained for object recognition that is applied to bottom-up region proposals provided by an external method such as selective search [30]. Due to the increased interest in object detection performed by neural networks, Microsoft released the Common Objects in Context (COCO) dataset in the same year, which was specifically designed to "advance the state of the art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding" [31].

The resulting continuous improvement and incremental extension of CNN architectures for object detection tasks allowed to tackle more and more complex tasks facilitating their application in real-world scenarios such as facial recognition in crowded environments, sign detection for autonomous driving or tumor detection in medical diagnostics. However, contrary to the carefully engineered research tasks for which these architectures were developed and evaluated against, false positive and or false negative predictions of these models can have potentially devastating consequences in these safety-critical or ethically sensitive scenarios. For instance, the misclassification of a stop-sign by an autonomous

car, which can be mistakenly recognized to be a speed limit sign [32], may lead to a car crash with fatal consequences. Furthermore, the in-accurate segmentation of lesions or abnormalities in medical images for diagnostic purposes may lead to faulty conclusions for the treatment of a patient, which again may have potentially fatal consequences [33]. Such scenarios pose new requirements for Deep Learning models to be reliably utilized beyond academic research purposes. These key requirements consist of an increased degree of transparency and interpretability of their decision-making processes to be intuitively understandable for human domain experts that are using these models. Interestingly, these requirements are still maintained despite the superiority of Deep Learning models over human domain experts with respect to their performance on large datasets, e.g., when distinguishing between benign or malevolent tumors in medical images [34–36]. Specifically, the acceptance by domain experts of a Deep Learning model's assessment of a tumor being malevolent, resulting in terminal and uncurable illness, is always manually verified, despite the model's lower probability for misclassification compared to the human expert. In this case, trust in the assessment is not solely based on performance measures but on the transparency and interpretability of the model's assessment. The process of attributing varying degrees of importance to specific features in the analyzed data and the ability to explain the subsequent decision-making process is the key to facilitate acceptance and trust in the assessment. In the same way, Deep Learning models need to be transparent, interpretable, and explainable in order to facilitate trust in the technology.

The further advancements in the field of computer vision beyond the ILSVRC have increased the need for explainability, interpretability and transparency considering the increased capabilities of neural networks and the more complex nature of the tasks that could be solved. Especially given the suitability of the models for real-world applications such as the recognition of street signs for autonomous cars, the localization of pedestrians and cyclists in complex traffic scenarios, or the segmentation of tumors in medical image data recorded with MRIs or CTs, the ability for domain experts to fully comprehend the foundation based which these networks make their decisions is crucial for their applications in ethically loaded domains.

2.2.3 From Supervised Learning to Reinforcement Learning

"Only ***those who dare to fail*** *greatly* ***can*** *ever* ***achieve*** *greatly."*

– John F. Kennedy, 35th president of the USA

The Deep Learning breakthroughs in the domains of computer vision and natural language processing, where supervised learning has been the paradigm of choice in the last years, have inspired the transfer of successfully tested methods and ideas into the field of reinforcement learning, a learning paradigm inspired by **the trial-and-error principle**. Early research in the domain of reinforcement learning has focused on the simple environments of board games with complete information. Board games such as Go, Chess, Checkers, Othello, and Backgammon have been widely regarded as ideal testing grounds for exploring a variety of concepts and approaches in reinforcement learning. From a human perspective, these games offer the challenge of tremendous complexity with respect to how many different possibilities a game can develop, and sophistication required to play at expert level. At the same time, the relevant information of the games for a reinforcement learning agent to process and the performance measures to evaluate the agent's playing strength are well defined, and the game environments are readily automated, i.e., it is easy to simulate the board, and the rules of the games such as moves of legal play and conditions regarding when the game is over and ready to be scored.

One of the earliest demonstrations of utilizing neural networks in the reinforcement learning setting for such an environment was Gerald Tesauro's TD-Gammon introduced in 1995, a temporal difference learning model that learned to play the game of Backgammon [37]. TD-Gammon utilizes a neural network to approximate the evaluation function of the game by playing against itself and learning from the outcome by aiming to predict the outcome based on the current position of the game. The general challenge to overcome in such a learning setting is the so-called credit assignment problem, i.e., the challenge to handle the possibly large delay of the outcome for an observed board position, which, in general, makes it difficult for learning models to determine how important specific moves were for the outcome of the game. TD-Gammon's success showed that the general concept of temporal difference learning is a promising technique for learning with delayed rewards and spurred increased interest in researching neural networks to learn more complex games.

2.2.4 Deep Reinforcement Learning Breakthroughs for Video and Board Games

"We don't grow when things are easy; ***we grow when we face challenges.****"*

– Unknown author

Given the vast success the ILSVRC brought to the domain of computer vision by providing a playground for annually **challenges** with standardized learning tasks, it is no surprise the domain of deep reinforcement learning found its own equivalent in the Atari learning environment (ALE). The ALE was of the first environment suites with arguably the most impact for academic DRL research up to date, as it spurred continuous improvements of DRL agents to tackle increasingly difficult challenges. It consists of 57 games initially released for the Atari 2600 console and constitutes a rich learning environment for DRL agents with various games and unique challenges that are not encountered in board games with perfect information. The earliest success in the ALE was reported by Mnih et al. in 2013 [38], who demonstrated the successful application of a DRL to achieve superhuman performances in some of the 57 games. Similarly to the ILSVRC, it took seven years for the continuous improvements in the ALE to peak in the development of *Agent57* [39], a DRL agent capable to solve alle 57 games on a superhuman level. For more details on these developments that lead from the earliest attempts to tackle the ALE via a detour into the domain of physics simulations and finally back to the ALE and *Agent57*, the interested reader may consult the fifth sub-chapter of the electronic supplementary material "*Continuous Improvements of DRL Agents in the ALE*".

The successful application of DRL agents in the ALE raises the question, whether the superhuman performances achieved by the trained agents are merely a result of the agents being capable to perform more precise sequences of controller inputs than humans or whether they exhibit some higher-level strategic behavior, which possibly remained uncovered by humans, to approach the various games played. Since it is not straight forward to investigate such higher-level strategic behavior as it is not explicitly stored within the networks, the need to extract such behavior from the learned representations of the networks becomes increasingly desirable as DRL agents are able to tackle increasingly complex environments.

One of these complex environments is provided by the game Chess, which has become particularly meaningful with respect to its perceived suitability to reflect the capabilities of human intelligence and thus, has emerged as a preferred playground to push the limits of AI regarding to surpass human intellectual capabilities. Interestingly, the complexity of the game of Chess has not been tackled successfully by a learning algorithm until 2018 and classical Chess engines largely relied on static heuristics and brute force calculations to play the game on a superhuman level. Thus, the need for explainability, transparency, and interpretability in the reinforcement learning domain and specifically the deep reinforcement learning domain was basically non-existent due to the heuristic and

analytical nature of computer programs that tackled the games like Backgammon and Chess and has only emerged much later when learning systems like neural networks were used to tackle these complex board games. For more details on some of the important historical milestones that made the game of Chess one of the most important games to be taught to computers to advance the research on artificial intelligence, the interested reader may consult the fourth sub-chapter of the electronic supplementary material "*Chess as a Mirror for Human Intelligence and Benchmark for AI*".

Inspired by the successes of DRL agents for the ALE, a Google DeepMind research team tackled the challenges provided by zero-sum board games and focused on the ancient game of Go, which in comparison to Chess, is even more complex with respect to how many possibilities there are for a game to develop. The size of a Go board is $19 \cdot 19 = 384$ intersections and allows for many more pieces/stones to be placed than on a Chessboard with $8 \cdot 8 = 64$ squares. After a single move out the 361 and 360 possible moves of both players, there are $361 \cdot 360/4 = 32490$ different possible positions that can arise on a Go board (the reduction by a factor 4 is due to rotation invariance of the board), whereas there are only $20 \cdot 20 = 400$ different possible positions on a Chess board. Due to the larger size of the board, the estimated number of different possible positions on a Go board is of the order of 10^{174}, compared to only 10^{120} for a Chess board. Thus, the brute force approach in combination with good heuristics for efficient tree search that works so well for Chess engines is infeasible for the game of Go. For this reason, the game of Go was widely considered to be an unsurpassable hurdle for current methods and approaches used in game engines for another 10 to 15 years. However, in Mai 2016, DeepMind's AI *AlphaGo* [40] shocked the Go world by beating the Go world champion at the time, Lee Sedol, in a five-game match with 4:1 [41].

In contrast to classical Chess engines, *AlphaGo* is a deep reinforcement learning based system that is able to learn the game of Go from a huge number of example games played by human experts and by subsequent improvement through self-play. First, *AlphaGo* is trained on the human example games to predict the next move in any given position, which essentially forces *AlphaGo* to mimic human play as close as possible. However, in order to correct for the mistakes in human play, *AlphaGo* subsequently plays games against itself and deliberately diverges from its learned behavior to find improvements and learn to correct the erroneous human play. In the subsequent year, *AlphaGo*, the first successful application of a learning system to purposefully navigate through the vast number of possible variations of how a complex environment can unfold, was further improved by DeepMind by completely abandoning any human knowledge as a

knowledge basis from which the game is learned. The improved version called *AlphaGo Zero* [42], introduced in 2017, in contrast to its predecessor, learns the game of Go from scratch and solely via self-play. This way, *AlphaGo Zero* is not forced into a regime of local optimality and surpasses the strength of *AlphaGo* by several standard deviations on the Go ELO scale. Considering the complexity of Go in comparison to other board games, to no surprise DeepMind demonstrated the generalization capabilities of the learning approach behind *AlphaGo Zero* by transferring it to the Game of Chess and its Japanese variant Shogi a year later, in 2018. *AlphaZero* constitutes DeepMind's general approach to learning highly complex board games with complete information from scratch, surpassing every human and every computer engine that has ever played these games [43]. In a 1000-games match against the TCEC (Top Chess Engine Championship) 2016 world champion *Stockfish*, the strongest computer Chess engine at that time, *AlphaZero* won convincingly with 155 wins, 6 losses and 839 draws. Interestingly, while *Stockfish* searches through roughly 60 million positions per second, *AlphaZero* only searches through a fraction of those with roughly 60 thousand positions per second, demonstrating its superiority in terms of strategic planning, more accurate evaluation of board positions and a better selection of game variations, in which calculation time is to be invested. *AlphaZero's* play style was described by former Chess world champion Garry Kasparov as "risky, dynamic and open" just like his own, and very much non-perfectionist [44].

The successes of *AlphaZero* and its learning approach, which combines Monte Carlo tree search to navigate the complex unfolding of the board games with strong representative models for the accurate evaluation of the emerging board positions, have enjoyed huge success for the Games of Chess, Go and Shogi, where a perfect simulation of the environment is possible. More precisely, the rules of these board games do not change during the game and there is no noise or variance with respect to how these games are played. Thus, it is sufficient to provide *AlphaZero* with the fixed set of rules of the games for its self-play mechanism to unfold. However, in real-world problems such as robotic control tasks or other complex planning tasks, the dynamics that govern the environment are often complex and partially unknown, making it impossible to provide a fixed set of rules to an agent that interacts with the environment. In order to tackle these kind of dynamic environments, in 2019 DeepMind introduced *MuZero* [45], an extension of *AlphaZero's* learning approach that does not require to be provided with any set of rules that govern the dynamics of the environment in which the agent acts. *MuZero's* extension constitutes a learned model that is applied iteratively to predict the quantities most directly relevant to the planning task, i.e., the reward, the action-selection policy, and the evaluation function used for the observed

states of the environment. When evaluated on Go, Chess and Shogi, without any knowledge of the game rules, *MuZero* matches the superhuman performance of *AlphaZero*.

Given the capabilities of *AlphaGo*, *AlphaZero*, and *MuZero* to navigate through complex games like Go, Chess, and Shogi, employing strategic behavior that remained uncovered by humans until revealed by the AI suggests that there's a deeper understanding of the game to be learned from the AI's behavior. Especially due to the inherently aligning nature of the behavior, which is given by the possibility to simply observe it by having the AI play the game according to the rules that humans understand, makes it highly desirable to investigate the learned representations aiming to explicate the higher-level strategic behavior to learn from them. Thus, the need for explainability, transparency and interpretability of potent DRL agents is not only high given their superhuman capabilities, but also likely to be alignable with human concepts that have developed over time to understand the respective domain in which the agents are deployed.

2.2.5 Tackling Games with Imperfect Information

*"**Not knowing** is half the fun."*

– Rick Riordan, American author

During the times of continuous improvements in the ALE, most of the games had been successfully tackled by the year 2017 through combining the continual improvements since the initial success of Mnih et al. in 2013 [46]. These successes together with the milestone success of *AlphaGo* have inspired further research utilizing more complex video games with new and unprecedented challenges. Specifically, the successes in environments with perfect information, like boards games and most Atari games, has led DRL research to explore environments with **uncertainty and imperfect information.** Such circumstances require an agent to learn flexible strategies in order to react to unforeseen developments in the environment and to adapt to the various possibilities an opponent may or may not act during a game. Coincidentally, the development of modern e-sports has created a competitive environment for a number of different games in which humans have so far excelled, battling in teams for large amounts of prize money in professionally organized tournaments. This competitive environment results in the active and continuous development of these games over decades and the periodic changes of in-game mechanics constantly evolving the environment semantics, making it an ongoing challenge for top players and teams to stay

at the top of their respective games. In recent years, the two of the largest e-sports games, namely *DotA 2* [47] and *StarCraft II* [48], have been subject to research and provided playgrounds for further developments of DRL agents to tackle the challenges of complex strategic games with imperfect information. Until today, the two big players in the AI research field, OpenAI and DeepMind, have created DRL agents that exhibit superhuman capabilities in these games and beat the best human players in a competitive environment with corresponding constraints and rulesets. For more details on the developments regarding the games of *DotA 2* and *StarCraft II*, the interested reader may consult the sixth sub-chapter in the electronic supplementary material, "*Deep Reinforcement Learning for Games of Imperfect Information*".

The impressive capabilities of DRL agents to perform on superhuman level in zero-sum games with perfect information, like Chess and Go, and their ability to exhibit superhuman performances in cooperative games with imperfect information, like *DotA 2* and *StarCraft II*, intensifies the requirement for explainability, transparency and interpretability for their strategic decision-making and long-time horizon planning. Given the possibility to transfer such agents into real-world scenarios, which require cooperative interaction with humans, and possibly diverging strategic ideas of humans and DRL agents with respect to a shared goal, inevitably results in the need to address such divergences in case of critical consequences as a result of these divergences. Thus, the ability to investigate how behavior emerges from the learned representations of the underlying neural networks of the trained DRL agents is crucial to facilitate their application in real-world scenarios.

2.3 Towards Research on Transparency & Interpretability

> *"All truths are easy to understand once they are discovered.* ***The point is to discover them.****"*
>
> *– Galileo Galilei, Italian polymath*

Recently, AI research has developed a new branch that aims to **facilitate more transparency, interpretability and explainability** of machine learning and deep learning models specifically. Contrary to the optimization driven approach spurred by challenges like the ILSVRC, this new branch is concerned with reverse engineering large state-of-the-art networks aiming to investigate higher levels of organization and structure of the learned representations within the networks.

These investigations assume that meaningful patterns and organized paths of connections through the different layers emerge within the network during training. The assessment of whether patterns are meaningful is commonly based on how much they can be characterized by humanly interpretable concepts. Coincidently, this general reverse engineering approach to understand neural networks is not at all a new one. It has been widely followed in the research field of neuroscience, which has been tackling large and complex neural systems performing difficult and complex tasks for decades with the goal to understand how these systems work and what the underlying mechanisms of these systems are for their information processing and decision making. Contrary to AI research, right from the start, the field of neuroscience was confronted with the challenge to investigate large and complex biological nervous systems that are just as much if not even more of a black box than artificial neural networks, aiming to understand how these systems work and how knowledge is organized and represented. Before addressing some key developments of the neuroscientific research in the past, which provides inspiration for the research field of explainability, transparency and interpretability, the following treatise summarizes the reasoning behind the lack of research regarding these factors and addresses the relationship to neuroscientific approaches for investigating learned representations of complex neural systems.

2.3.1 A Shift of Paradigm: From Optimizing to Understanding

"Premature ***optimization is the root of all evil.****"*

– Donald Knuth, American computer scientist and Turing award winner

With respect to Chess, to put it in the words of former Chess world champion Garry Kasparov: "*AlphaZero* shows us that machines can be the experts, not merely expert tools. […] The knowledge it generates is information we can all learn from. *AlphaZero* is surpassing us in a profound and useful way, a model that may be duplicated on any other task or field where virtual knowledge can be generated." However, despite the undeniably high degree of learned expertise of *AlphaZero*, its predecessors, and successors, explainability of its decision-making is yet an unsolved problem. Understanding the choices of *AlphaZero*, i.e., the actions it takes and the moves it makes, is only possible for already very strong grandmasters, who developed a deep understanding about the game of Chess over the courses of their lives, however, to most humans the implicitly stored

knowledge that is represented by *AlphaZero* remains inexplicable and inaccessible. The same issue holds true for many of the previously presented milestones of deep learning research in the domains of computer vision and natural language processing as well as other deep reinforcement learning applications.

In general, the successes and breakthroughs of deep learning in the domains of computer vision, natural language processing and reinforcement learning have created a number of deep learning applications that solve the tasks they were developed for better than their human expert counterparts. Not only do these models work through heaps of data that are insurmountable for any single human being—as is the case in computer vision applications in the domain of medical diagnostics—but they also exhibit sound behavior and develop new and profound strategies to navigate their environments in ways that were previously undiscovered by humans—as for example regarding the Games of Chess and Go—despite hundreds of years of collective human experience in these environments [34–36, 39, 43, 49]. These successes can be largely attributed to the focus of AI research in the past decade. **AI research has been mostly optimization driven** and is marked by milestones that continuously improve the performance of models evaluated by some error metric on large and carefully constructed benchmark datasets. Once a learning task of such a benchmark dataset was sufficiently well solved and the minimization of the respective error metric reached a plateau, a new, more complex dataset was constructed to tackle new and more complex learning problems. This approach has been accompanied by or maybe even resulted in the increased availability and affordability of specialized computational resources for the training of deep learning models. Both developments, on the research side and on the hardware side, have resulted in increasingly large and complex network architectures that solve the tasks they were developed on a superhuman level. The result of these developments were increasingly more complex and more potent learning models that raised the question of how they represent the complex knowledge to perform on a superhuman level. Thus, the need for methods and approaches to investigate their learned representations has become more and more desirable. The lack of transparency of deep learning models and the lack of explainability and interpretability of their decision-making not only makes it difficult to extract learned knowledge and valuable insights from the model, but it also makes it difficult to trust its decision making as it is not comprehensible and traceable from a human point of view. Given the undoubtedly high potential of those models for various learning tasks in different domains, the transfer of their application seems obvious. However, due to their inherent lack of explainability, transparency and interpretability, their transfer and application to some domains, e.g., safety critical or ethically loaded domains, is problematic. Although these

models can be trained to perform better than most humans for a specific task, they are still not perfect and an accuracy of 100%, which would constitute a perfect model that makes no mistakes, is not realistic in light of dynamic factors and changing circumstances of the environments that these models are used in. Thus, these models are bound to make a mistake at some point, just as humans are, however, in contrast to humans, they cannot explain why they made a mistake. The inability to have a model explain its mistakes or to manually determine why it makes mistakes makes it difficult to fix these mistakes as the underlying mechanisms that cause them to remain hidden.

The general optimization driven research approach of the field of AI and the neglect of efforts to facilitate explainability, transparency and interpretability simultaneously to the fast developments of models and methods has added up to a mountain of research debt [50]. Research debt describes the phenomenon of a growing research field resulting in their researchers to have to climb an increasingly larger mountain of research knowledge before they can reach the peak and contribute to the research field at its frontiers. For the research field of AI, the development of increasingly larger and complex models has contributed to such a mountain of debt when it comes to the goal of facilitating explainability, transparency and interpretability, as researchers are required to facilitate these aspects from scratch for their individual investigations, their individual models and individual use cases and applications. Given the lack of well-established methods to facilitate these aspects, extensive research is necessary to lay a foundation for XAI researchers. This direly needed research may take great inspiration from the research field of neuroscience, as detailed in the following sub-chapter.

2.3.2 Inspirations from Neuroscience Research

"Any man could, if he were so inclined, ***be the sculptor of his own brain.****"*

– Santiago Ramon y Cajal, Spanish neuroscientist, pathologist, and histologist

The idea of large-scale **investigations of large and complex neural systems** is not new. The research field of neuroscience has been concerned with the investigation of learned representations in such systems for decades. Interestingly, many of today's new challenges encountered when addressing the lack of transparency and interpretability in large state-of-the-art artificial neural networks are typically encountered in neuroscientific studies when investigating the mammalian brain. Thus, neuroscience has developed a number of methods and approaches to tackle these challenges, while at the same time, had to deal with moral and ethical

concerns as well as experimental difficulties when conducting investigations of the living brain. Despite those additional challenges, neuroscience offers a unique perspective on large neural systems and the transfer to artificial neural networks seems promising, especially considering that artificial systems are not subject to experimental difficulties or ethical concerns.

The inspirations that the research field of XAI has taken from other research fields that are concerned with biological learning systems in the past have their earliest example from the late 19th century, which has influenced a whole branch of research that laid the foundation for the learning paradigm of reinforcement learning. Specifically, the utilization of using neural networks in the reinforcement learning setting, called deep reinforcement learning (DRL), is inspired by early experiments in the fields of cognitive and behavioral psychology [51]. In 1897, Ivan Pavlov, Nobel laureate and founding father of behavioral psychology, laid the cornerstone of what is known today as reinforcement learning with his famous experiments on classical conditioning [52]. Pavlov showed empirically that it is possible to condition dogs to manually trigger their reflex to an increased production of saliva simply by ringing a bell. The systematic conditioning was done by ringing a bell whenever the dogs were fed, leading them to make an internal connection between the auditory stimulus of the ring of the bell and the act of feeding. Although initially the ring of the bell had no effect on the dog and the reflex of increased production of saliva is only triggered during food intake, over time, due to the repeated coincidental events of the auditory stimulus and subsequent feeding, the ring of the bell alone was sufficient to trigger the reflex without actually feeding the dogs. The results of Pavlov experiments can be loosely described in a setting of the reinforcement learning paradigm, in which an agent adapts a specific behavior based on its observations of the environment and some external reward. In Pavlov's experiments, the agent (dog) observes its environment (bell ring) and changes its behavior (producing saliva) based on an external reward it receives (food provision). Although Pavlov demonstrated that an agent's behavior can be conditioned by the combined presentation of an external stimulus and a reward, his experiments were not aimed to incentivize an agent to adapt a specific behavior with the intention to learn how to solve a specific task. In computer science, classical conditioning has been adapted from behavioral psychology to achieve exactly that, to enable a program to learn how to solve specific tasks [53].

Besides the work of Pavlov, between 1959 and 1968, David H. Hubel and Torsten N. Wiesel laid the foundation that has inspired decades of neuroscientific research with their pioneering work on the processing of information in the visual system [1–6], which was honored with the Nobel Prize in 1981. They

investigated single neurons in the cat and monkey visual cortex and their activity in response to a carefully crafted set of stimuli. Specifically, they recorded the neuronal activity of single neurons, i.e., their firing frequency, in response to bar stimuli with different characteristics. The bar stimuli varied in orientation, length, and width and were presented as light bars on a projections screen, which was placed in the visual field of the animals. Hubel and Wiesel showed that neurons in the visual cortex possess a strong selectivity for the orientation of the bars as well as for the position of the bars within the visual field of the animals. More precisely, they showed that neurons are only active, when bars are presented at a specific location in the visual field, called the receptive field, and are presented with a specific orientation, i.e., a specific angle of the bar. Rather than exhibiting some kind of all-or-nothing activity in response to the bar stimuli, the activity of these neurons gradually emerges or fades away depending on the change orientation of the light bars. Thus, the activity of these specialized units is specific for small ranges of bar orientations rather than a single fixed orientation so that different units show overlaps in their activity in response to their specific ranges of orientations. [54]

Hubel's and Wiesel's results showed that individual neurons can be attributed with overtaking specific sub-tasks, i.e., they exhibit specific functionalities, during the overall activity of processing visual information. These kind of specialized units are not at all unique, like the infamous Jennifer Aniston neuron hypothesis states [55], but work together in groups, i.e., clusters of neurons exhibit joint and coordinated activity in response to stimuli with specific characteristics. Specifically, if one unit's activity starts to fade out as the orientation of the presented bar stimulus gradually changes, another unit's activity gradually rises. This kind of redundancy gives the brain of its most remarkable and most important feature, which is its robustness against the continuously progressing cell death of neurons, which is a natural side effect of aging. Inspired by Hubel's and Wiesel's results, later research found that this kind of specialization is organized and structured in specific ways throughout the neocortex. The scales on which characteristics of stimuli can be described, such as an angular scale to describe bar orientations or a frequency scale for sounds, is mapped onto the surface of the neocortex in a way so that smooth changes of stimulus characteristics on these scales correspond to smooth spatial transitions of activity on the surface of the neocortex. These kinds of organized structures have been found all over the neocortex. One of the very first and probably best studied of their kinds is the famous pinwheel structure in the visual cortex, which maps bar orientations onto its surface in a pinwheel-like fashion [56]. Another example can be found in the auditory cortex, where individual neurons are selectively activated for sound with specific frequencies

and the frequency scale of these sounds ranging between 125 Hz and 22,000 Hz is mapped onto the surface of the auditory cortex [57]. Lastly, the sensory and somatosensory motor cortex is characterized by a specific mapping of body parts onto its surface, called the Homunculus [58]. Interestingly, the size of the occupied area on the surface of the cortex does not correspond to the actual size of the body parts, but rather to the density of motor neurons responsible for sensory perception and motor control of these body parts, which results in a distortion of the representation of the actual body as a consequence of the mapping.

These representations show that low-level characteristics of external stimuli commonly found in the real-world are structured and organized by the brain. As this phenomenon is consistently found across mammals and in different areas of the mammalian brain processing different kinds of stimuli, this suggests that it is an inherent feature of complex neural systems to organize information processing in such a systematic way. Consequently, this raises the question, whether artificial neural networks trained on large and representative real-world datasets exhibit similar features of organizing the representations that they learn. Furthermore, an immediate follow-up question is whether neuroscientific methods, that have been used extensively to study large biological neural systems, can be transferred to artificial neural networks with the same success they yielded in neuroscience. Initially, this approach seems promising as neuroscientists are provided with trained systems that have developed over millions of years of evolution and thus, never had to go through the effort to create these systems in the first place. For this reason, neuroscientists have been concerned with reverse engineering these complex systems for decades aiming to understand their underlying mechanisms that make them such potent learning models. Consequently, neuroscience research has built decades of experience attempting to make large and complex neural systems transparent and interpretable.

These attempts have led to the development of various universally important methods to investigate neural systems *in vivo* and *in vitro*, which serve as a source of inspiration for the empirical investigations of artificial neural systems. For more details on these and their relation to the investigation of learned representations of neural systems, the interested reader may consult the seventh sub-chapter in the electronic supplementary material, "*Neuroscientific Methods to Investigate Large Neural Systems*".

Methods and Terminology

3

"Names don't constitute knowledge. ***Concepts*** *do." (paraphrased)*

– Richard Feynman, American Nobel laureate physicist

Throughout the remainder of this thesis, a number of terms and definitions are used repeatedly to refer to **important concepts** that are related to the transparency and interpretability of neural networks. Furthermore, in order to answer the previously phrased research questions, a number of different analysis methods are applied to various datasets that are used throughout this thesis. Some of these methods are of rather universal nature with respect to handling artificial neural networks and are well-established within the current state of the art of AI research, such as training and evaluating neural networks, while others are more specific to the concrete goal of this thesis. In case of standard methods such as backpropagation, regularization, the splitting of a dataset into training, test and validation data, and various other methods to prevent overfitting of neural networks, the reader is referred to corresponding literature the first time they are mentioned in the text for further details on the methods, which are beyond the scope of this thesis. In case of the more specific methods, the following section provides the necessary background on the relevant details of the analysis approaches to further delimitate the object of investigation.

3.1 Learned Representations

The concept of a learned representation generally refers to what a neural network has learned from its high dimensional training data. More specifically, a network is forced to learn some kind of internal representation of the training

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_3

data, oftentimes lower in dimensionality than the training data, to approximate a function that maps the presented inputs to the correspondingly presented targets (in the supervised learning case). A network's only ability to adapt itself to the characteristics of the presented training data is via its weights and bias, which therefore constitute the foundation of its learned representation. The highly individual parameterization of the network via these weights and biases that emerges from training is usually highly specific to the characteristics of the presented training data. However, despite their dependency on the presented training data, the weights and biases can be investigated independently from the data. In this case though, it is not clear how meaningful any interpretation of such an individual network parameterization is, as it deliberately discards the relationship to the training data from which it emerged. A part of the results of the studies presented with respect to answering the first research question will shed some light on this issue and show that an individual neuro's incoming weight distribution can indeed be interpreted in a meaningful way with respect to how important this individual neuron is for the network's learned task and what role it plays in the learned representation.

An alternative perspective on a network's learned representation, which directly considers the used training data, is provided by its activation. In contrast to its weights and biases, the network's activation can only be investigated upon processing of input data, giving it an inherent dependency on the used training data. This perspective is strongly inspired by neuroscientific studies of the mammalian brain, whose activity is a well-researched object of investigation for many decades. Specifically, the brain's activity has been recorded most famously via functional magnetic resonance imaging (fMRI), to visualize the activity of individual brain areas during the processing of information. These visualizations are typically obtained via blood oxygenation level dependent (BOLD) imaging within the brain, which is based on visualizing blood flow through the individual brain areas. The notion is that active brain areas require more oxygen to maintain its metabolism than inactive areas, which is provided by an increased amount of blood flow through these areas. fMRI has contributed greatly to insights about specialized areas of the brain responsible for the processing of different kinds of external stimuli, e.g., visual stimuli, auditory stimuli, or olfactory stimuli, as well as the generation of specific behavior, like speech or motion [59]. Thus, a brain area with a high blood flow is considered to be more active than brain areas with lower blood flow. For details about the fMRI method, how it allows to visualize brain activity and how these visualizations can be interpreted, the interested reader is referred to [60].

The concept of attributing meaning to specific network areas based on their activity can be directly transferred to artificial neural networks. In similar fashion to fMRI studies, artificial networks can be probed with data to invoke network activity, which can be recorded and used as a basis to determine the importance of individual neurons, groups of neurons or whole layers for processing the specific characteristics of the presented data. When probed with data that has not been presented during the network's training, the invoked activity is only meaningful if the network has learned to generalize from the training data to the data that is used to invoke the activity. In the worst case, the test data differs considerably from the training data, so that the invoked network activity cannot be interpreted meaningfully. For instance, a network that is trained to exclusively distinguish between cats and dogs in image data could be probed with images of cars and ships, however, it cannot reasonably be expected that the invoked activity is meaningful with respect to the distinction between ships and cars, as the networks was not forced to learn a representation that considers that distinction.

3.1.1 Investigating Learned Representations

Learned representations can be investigated on different spatial scales, i.e., locally, or globally. An investigation on the smallest possible scale, which is based on individual weight and bias values, does not provide any insights with respect to the individual meaning of individual neurons as it omits any comparison to other neurons. However, a comparison of these individual values can be made to themselves at different times during training allowing for comparisons between trained and untrained networks. This facilitates to characterize how these individual values change during network training, when the representation emerges and may yield insights about when specific characteristics of a learned representation emerge during training. The next larger scale is the scale of individual neurons. On that scale, a neuron's activation values can be investigated in similar fashion to individual weights and biases. Additionally, considering that an individual neuron usually possesses several incoming and outgoing connection with individual weights, a comparative perspective on its weights can be taken allowing to categorize these weighted connections according to some metrics, which in principle facilitates to attribute meaning to them based on such metrics. The next larger scale is the scale of individual network layers. On that scale, a layer's activation values can be investigated in similar fashion to individual neuron activations. In this case, considering that an individual layer usually possesses several neurons, the layer's neuron activations are considered jointly

and condensed into one single layer activation value. Additionally, a comparative perspective on the individual neuron's activation values can be taken in similar fashion to the incoming/outgoing weighted connections of individual neurons. Inspired by fMRI studies, subsets of a layer's set of neurons can be determined via joint activations in response to specific stimuli, revealing similarities or dissimilarities between neurons with respect to how they are activation for different stimuli. This way, functional neuron populations can be characterized via individual neuron's activation patterns. The largest scale is the scale of a whole network. On this scale, considering that a network usually possesses several layers, the network's layer activations are considered jointly and condensed into one single network activation value. Additionally, a comparative perspective on the individual layer's activation values can be taken in similar fashion to the individual neuron activations within a single layer. Furthermore, a comparison between different networks with respect to their network activations can be made irrespective of their specific network architecture, allowing for studies of network activity in similar fashion to neuroscientific studies, where different brains that are investigated are guaranteed to be different in their architecture and no control over aspects like network topology is given. The dimensionality of the investigated learned representations becomes high-dimensional when considered on larger scales and poses the inherent challenge to find and extract patterns within that high-dimensional space, which can be visualized and interpreted in a meaningful way. Number of metrics and methods are applied to these representations throughout the presentation of the results in this thesis, which are briefly dealt with in the following subsections.

3.1.1.1 Statistical Measures

The simplest and most straight forward way to investigate learned representations is via statistical descriptions of its individual parameters, i.e., their weights, biases, and activations. These values can be described via their distribution functions and their corresponding characterizing measures such as the mean, the median, the variance, and higher order statistical moments. In case of weights and biases, these distributions can be obtained at different stages during a network's training phase and then be compared with each other via statistical significance tests to determine whether the weights and biases show significant distinctions at different training stages, possibly revealing when they emerge during training. Additionally, in case of activations, these distributions can be obtained for specific subsets of inputs and then be compared with each other via statistical significance tests to determine whether the activations show significant distinctions when invoked by different inputs. Furthermore, the similarity of activations

in response to specific inputs by individual neurons, layers, or networks can be determined via their correlation to one another. Commonly, measures like the Pearson's correlation or Spearman's rank correlation are used to determine such similarities in the neuronal responses.

3.1.1.2 Transformations & Embeddings

In general, the interpretation of high-dimensional data via visual inspection is not straight forward because the visualization of more than three dimensions is inherently difficult to achieve due to our limited 3-dimensional perception of the environment. Even adding a fourth dimension to the three spatial dimensions, e.g., time, is already quite abstract and although comprehensible in principle, difficult to visualize. In case of learned representations, this general issue becomes quickly apparent when considering increasingly large numbers of neurons and layers of a trained network. Depending on the scale in which learned representations are investigated, each neuron's or layer's individual activation constitutes an individual dimension of the learned representation. Thus, these representations can have millions and billions of dimensions, e.g., when considering large state-of-the-art neural networks. The challenge of visualizing such high-dimensional representations with the purpose of facilitating interpretability is to find an embedding that reduces the representation's dimensionality to two dimensions without losing too much information in the process. In general, embedding techniques are aimed to find a projection of data from its original high-dimensional space to a lower dimensional space, typically two-dimensional, to be visualized in a scatter plot for example. This visualization can then be inspected to find any kind of structure or organization of the embedded learned representation. However, it must be kept in mind that the embedded representation is a somewhat simplified version of the original representation and that its structure may only be partly representative of the representation's structure in the original high-dimensional space. Furthermore, most state-of-the-art embedding techniques perform highly non-linear transformations, which makes the interpretation of the representation's structure in the embedded space difficult. A possible way around this issue would be to fall back to more classical embedding techniques that perform linear transformations, however, these techniques mostly perform worse than non-linear techniques with respect to the amount of information that can be preserved by the transformation. Thus, a universal problem of embedding learned representations is to find a good trade-off between preserving as much information of the original representation as possible and sacrificing interpretability due to non-linearity. Some examples of these kinds of embeddings presented later in this thesis were obtained via non-linear transformations, while other were deliberately obtained with linear

transformations. There are advantages and disadvantages for both approaches that must be considered for each analysis individually.

Three of the most widely used embedding techniques were used to obtain the results presented in this thesis, which are briefly outlined in the following subsections. Besides these three techniques, a number of other classical embedding techniques, both linear and non-linear, exist but are omitted here as they are beyond the scope of this thesis. Details about those techniques can be found in [61]. The three techniques were chosen specifically because they constitute the best representative of the two categories of embedding techniques, i.e., *Principal Component Analysis (PCA)* as a linear embedding, and *t-Stochastic Neighbor Embedding (t-SNE)* as well as *Uniform Manifold Approximation and Projection (UMAP)* as non-linear embeddings. The two non-linear techniques were used, because the t-SNE constituted the current state of the art at the time of use while UMAP replaced t-SNE during the work on this thesis.

Principal Component Analysis

A Principal Component Analysis (PCA) is a popular linear dimensionality reduction technique, i.e., it performs dimensionality reduction by embedding the data into a linear subspace of lower dimensionality [62]. When finding the embedding in that lower dimensional subspace, PCA aims to retain as much of the data's variance as possible. This is achieved by performing an orthogonal, linear transformation of the data's original space via rotating and translating the axis of its coordinate system. The principal components obtained by PCA constitute the directions of the newly obtained coordinate system. The principal components are ranked in descending order according to the fraction of the variance of the data that is measured along these components. Note that all components are orthogonal to each other. Figure 3.1 illustrates the transformation of the data. Loosely phrased, first, a set of orthogonal vectors is found, which is ranked according to the fraction of the variance of the data that can be measured along these vectors (cf. Figure 3.1, left). Second, each data point, itself being a vector in the original data space, is transformed, i.e., rotated and translated, so that it can be placed in the new space of the previously obtained set of orthogonal vectors (cf. Figure 3.1, middle). The data may be further reduced to an arbitrarily small number of dimensions by discarding individual principal components with lower variance, thus projecting the data onto components with higher variance (cf. Figure 3.1, right). Formally, the principal components are obtained by computing the eigendecomposition of the covariance matrix of the data. A linear mapping M for the covariance matrix $cov(\boldsymbol{X})$ of the dataset $\boldsymbol{X}$ must be found,

which maximizes $\boldsymbol{M}^T cov(\boldsymbol{X})\boldsymbol{M}$, which is equivalent to solving the eigenproblem $cov(\boldsymbol{X})\boldsymbol{M} = \lambda\boldsymbol{M}$, where λ are eigenvalues of the Matrix $\boldsymbol{M}$. The obtained principal components are the eigenvectors of the covariance matrix of the data $cov(\boldsymbol{X})$.

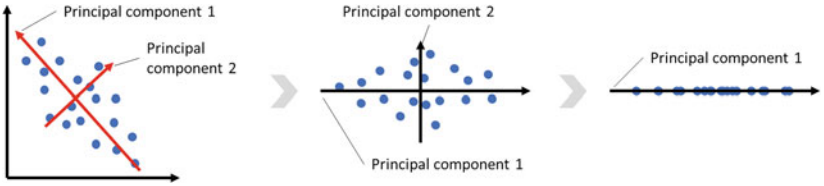


Figure 3.1 Schematic illustration of the data transformation process and subsequent dimensionality reduction step of PCA

PCA was found to outperform many non-linear dimensionality reduction techniques regarding the extraction of generalized information while offering better interpretability. Furthermore, in most cases the computational complexity is lower compared to non-linear techniques which is important for practical applicability [61, 63]. Due to its linear nature, PCA is inherently limited to embed highly non-linear data, and the significance of the transformation is strongly dependent on the numerical range of the variables, which commonly vary widely for real-world applications. Typically, this limitation is mitigated by scaling each feature to unit variance before computing the principal components.

t-Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) was first introduced in 2008 by Laurens van der Maaten [64] and has been the most prominent state-of-the-art dimensionality reduction technique for visualizing high dimensional data for over a decade. The main goal of t-SNE is to lay the focus on local structure within an n-dimensional dataset $\boldsymbol{X} = \{\boldsymbol{x}_i \in \mathbb{R}^n | i \in [1, 2, \ldots, N\}$ containing N data points, while also preserving global structure when reducing its dimensionality, which had usually been lost in other embedding methods that were aimed at preserving local structure. Loosely speaking, the structure of a dataset is determined by the pairwise distances of the individual points to each other. Upon reducing the dimensionality of a dataset, these distances cannot be preserved entirely so a tradeoff needs to be made between preserving local structure, i.e., pairwise distances between nearby points, and global structure, i.e., distances between clusters of points. The original t-SNE algorithm is based on two main

ideas: 1) Describing the pairwise similarity of point $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as the probability p_{ij} that $\boldsymbol{x}_j$ falls within a spatial probability distribution of $\boldsymbol{x}_i$ and 2) using the Kullback-Leibler divergence as a measure of similarity between the symmetric pairwise similarities of points in the original high-dimensional space p_{ij} and the projection in the low-dimensional space q_{ij}, and minimizing this divergence with gradient descent [65].

The following section describes the two ideas and the underlying motivation in more detail, as well as consequences for the interpretability of the projection.

Similarity Measure

In the original high-dimensional space of the dataset, pairwise similarities between points are computed using the symmetrized pairwise conditional probability in a Gaussian distribution:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

with

$$p_{i|j} = \frac{e^{\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)}}{\sum_{k \neq l} e^{\left(\frac{-\|x_k - x_l\|^2}{2\sigma^2}\right)}}$$

Using these probabilities, far-away points have an almost infinitesimal similarity. Close points, on the other hand, obtain a high similarity value. The symmetrization ensures that outlying points will contribute to the cost function, which results in more meaningful placements of such outliers.

In the low-dimensional embedding space, i.e., the target projection of the dataset $\boldsymbol{Y} = \{\boldsymbol{y}_i \in \mathbb{R}^m | i \in [1, 2, \ldots, N\}$, with $m < n$, similarities are computed using the Student's t-distribution [66] with one degree of freedom. Mathematically, the distribution is described by:

$$q_{i|j} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

Compared to the Gaussian distribution, the Student's t-distribution is wider at the bottom, a property also referred to as heavy-tailed. The qualitative difference between the two distributions is visualized in Figure 3.2. Since the heavy-tailed Student's t-distribution is used in the low-dimensional representation, the probability that more distant points y_j fall within the distribution of a specific point y_i is higher. Consequentially, a moderately distant point in the original dataset can be projected with a larger distance in the low-dimensional space. This resolves a problem known as the Crowding Problem. The Crowding problem often occurs in embedding methods with a focus on local structure, when small distances are heavily weighted in the objective function and large distances are weighted only marginally. That way, the weighted objective function results in projections, where all clusters are squeezed close to the origin of the new coordinate system.

Cost Function

The cost function used in t-SNE uses the pairwise similarities computed as described above and is a measure for the divergence between the two probabilistic representations. More specifically, the Kullback-Leibler divergence

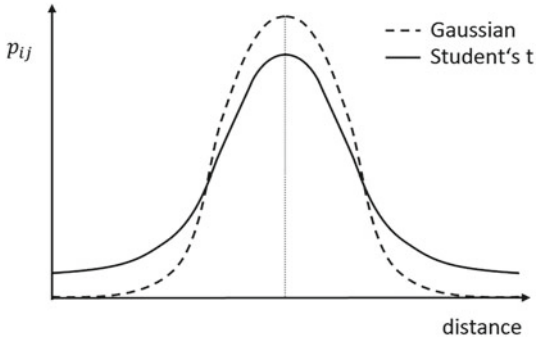


Figure 3.2 Illustration of the qualitative difference between the Gaussian and the Student's t-distribution

$$\sum_i \sum_j p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right)$$

is applied and minimized. From the above equation it is evident, that large distances p_{ij} in the data, which are modelled with a small distance in the low-dimensional representation q_{ij}, are heavily penalized. Modelling small distances

in the data far apart, however, contributes less to the loss function. In combination with the heavy-tailed distribution, volume differences of clusters are not represented in the projection. Originally, the t-SNE algorithm's computational complexity scaled quadratically with the number of points in the dataset. It was therefore computationally expensive to apply it to large high-dimensional datasets with more than 10,000 datapoints. In 2014 a solution to the scaling problem, using the Barnes-Hut approximation, was introduced [65], however, details regarding this approximation is not relevant for the further understanding of the presented results and beyond the scope of this thesis.

Interpretability

The principle of t-SNE has been successfully applied to both synthetic and real-world datasets, helping to visualize the inherent structure of these datasets [64, 67, 68]. Figure 3.3 shows a two dimensional t-SNE embedding of the 10,000 digits in the test set of the infamous MNIST dataset [69, 70]. In general, the individual classes are well separated from each other. Due to the focus on local structure, variations within the clusters are also represented in the mapping. For example, the class of digits containing hand-written ones (top left, red) shows a change of inclination when moving from left to right.

These and other comparative results on other datasets suggest that t-SNE provides a good qualitative insight into the structure within these datasets via mere visualization. However, there are some limitations to the algorithm's interpretability. t-SNE is governed by hyperparameters, depending on which the two-dimensional embedding can change. One of those hyperparameters is the perplexity, which serves as an initial guess of how many data points are considered when determining where to place individual data points in the embedded space and is related to the number of nearest neighbors that is used in other manifold learning algorithms. Larger datasets usually require a larger perplexity, as a larger number of neighboring points need to be considered. A small value for the perplexity usually results in a larger number of clustered data points, while larger values result in a smaller number of clusters containing more data points. Since the cost function does not penalize global dissimilarities between clusters severely, small perplexity values result in many small clusters, which are scattered across the embedded space without much recognizable structure. Large values, on the other hand, may result in blurred clusters, since dissimilar points still fall into the probability distribution of points belonging to another cluster. Moreover, because the cost function does not penalize global dissimilarities severely, it follows, that distances between clusters are less meaningful than distances within clusters. Therefore, while distances between clusters are an indication for a good

separation of these clusters on a global scale, the distances should not be interpreted relative to each other. For instance, the fact, that two clusters are twice as far apart from each other than two other clusters, does not mean that they are better separated, and it does not correspond to the degree of similarity of both clusters, meaning that the far apart clusters are not more dissimilar than the closer clusters. Similarly, interpreting the size of clusters must be done with caution. The scaling of local densities, i.e., the placement of data points within an individual cluster, which is performed within the algorithm, does not provide insights with respect to the similarity of the data points within the cluster. This means, that compressed clusters with a high density do not necessarily contain more similar data points than scattered clusters with a lower density. For this reason, the size of clusters in the projection, holds no information on original sizes and densities. [71]

Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) is a non-linear dimensionality reduction technique introduced in 2018, ten years later after the first appearance of t-SNE in, with the motivation to create a dimensionality reduction based on well-studied mathematical models [63]. The idea behind the algorithm can be divided into three parts: 1) Construction of a local topological manifold approximation on the original dataset using a fuzzy simplicial representation. 2) Construction of a simplified topological manifold approximation in the Euclidian space. 3) Minimizing the cross-entropy as a measure for the difference between both representations. The following section describes the three ideas and the underlying motivation as well as their consequences for the interpretability of the projection of the data in the resulting embedded space.

Local Topological Manifold Approximation of High-Dimensional Data

First, the original dataset is mathematically modelled as a topological space. Such topological models of data are commonly based on a sample of data points typically taken from a physical object of interest to be depicted that is described by the data points. These points are then connected to form edges, planes, and volumes of the topological model. UMAP uses the data points $\boldsymbol{x}_i$ of the dataset $\boldsymbol{X}$ directly and creates a topological model from them. However, the high-dimensional datapoints are only connected to form edges, as planes and volumes would be computationally expensive, and edges are sufficient to describe some kind of distance and similarity between these points. The resulting structure is a graph with edges connecting points which are close together. Subsequently, the

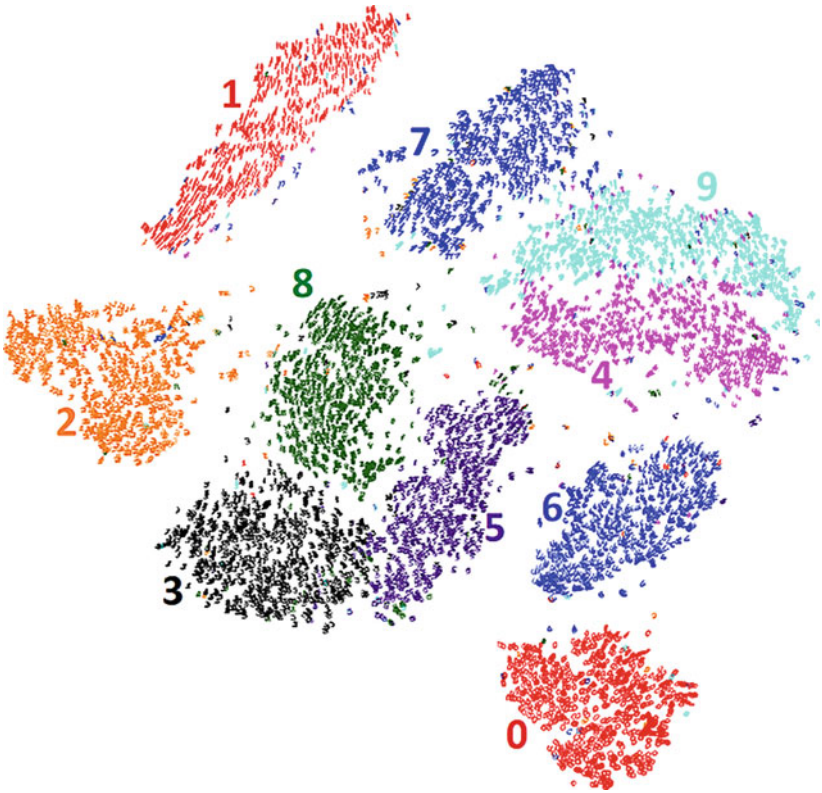


Figure 3.3 t-SNE embedding of the 10,000 digits in the MNIST test set. Taken from [172]

edges of the graph require a definition for their weights in order to represent distance or similarity between neighboring points. The definition is based on three key ideas:

1. The first idea is based on the assumption that the data points considered to build the graph are uniformly distributed. However, this is not necessarily true commonly not true when considering common Euclidean space. Thus, for this assumption to be justified, Riemannian geometry is applied to the manifold on which these data points are located, reshaping the original Euclidean space so that the considered data points are more evenly distributed. This allows

to assign an individual surrounding radius to each data point, in which k neighboring points lie. If each individual point on that reshaped manifold has k neighbors, the points are approximately uniformly distributed on that manifold.
2. The second idea is to define a similarity measure, which is based on assigning weights to the edges that connect individual data points and describes how similar individual points are to each other based on their distances from one another. The assigned similarity value is a fuzzy value between 1 for very close points and 0 for points outside of the defined radius.
3. The last idea is that each point is required have at least one connection to another point in order to avoid the extensive emergence of outliers. Outliers would represent points that have no connection to the rest of the data and thus don't contribute to the structure of the data. Therefore, an additional condition is considered, i.e., a minimum similarity value is assigned to an individual point and its nearest neighbor.

Figure 3.4 illustrates these three ideas on the example of a sinusoidal curve. Each data point is colored individually according to its positioning from left to right for visual purposes of distinction. The monochromatic circles correspond to the radii for the individual data points that reflect the distance to its nearest neighbor while the extensions of these circles and the fading color correspond to the similarity values assigned to the neighbors of an individual data point.

Applying these three ideas, a graph can be constructed. Since each point has an individual distance metric, the distance between two points might be described by multiple different values, depending on the effective radius around the involved points. Those values are summarized using the fuzzy set union, resulting in a graph with only single direct connections, where the edges represent the probability of the existence of the connection. Details about the mathematical background with respect to the topological construction of the graph that involves the three presented ideas are beyond the scope of this thesis but can be found in the original publication [63].

Low-dimensional Representation in Euclidian Space

Just like the original dataset, the low-dimensional UMAP projection of is also represented by a fuzzy topological structure, which makes both topological structures comparable, i.e., a similarity can be calculated via a cost function. To achieve this, two simplifications are used:

1. The first simplification regards the distance function. In the high-dimensional space, the uniform distribution was facilitated by assigning individual distance functions to each individual data point. In the projection space, a single distance metric is used to calculate the pairwise distances between all points to allow for easier comparison.
2. The second simplification regards the minimum of one connection per data point, which was enforced in the high-dimensional space by the circle around an individual data point with the radius equal to the distance to its nearest neighbor. This minimum distance is also accounted for in the projection space. However, the value of the minimum distance is manually passed to the algorithm as a hyperparameter, so it functions as a global criterion to determine whether two points are connected or not. Thus, the projection can manifest in different forms depending on the choice of this hyperparameter. [63]

Cost Function

To compare the two representations and minimize the difference effectively, a meaningful cost function F is introduced. Both representations of the dataset can be interpreted as pairs of points with probabilities describing the likelihood that a connection between two points exist. The result of both representations are probability vectors of size E, which are compared using cross-entropy,

$$F = \sum_{e \epsilon E} w_h(e) \cdot log\left(\frac{w_h(e)}{w_l(e)}\right) + (1 - w_h(e)) \cdot log\left(\frac{1 - w_h(e)}{1 - w_l(e)}\right)$$

where w_h and w_l correspond to high and low-dimensional weights, respectively. The first term of the equation ensures that large weights in the high-dimensional space w_h are represented by large weights in low-dimensional space w_l. On the contrary, the second term ensures that small weights w_h are represented by small weights in the projection space w_l. [63]

Interpretability

UMAP has been successfully applied to many large-scale datasets, providing results, which are comparable or better than those obtained with t-SNE [63, 72, 73]. Furthermore, UMAP has a higher computational efficiency and scalability than t-SNE [63]. Figure 3.5 shows the UMAP projection of four different datasets along with the projections obtained by using t-SNE. It shows that UMAP returns visually more accurate projections on the COIL20 [74] and the MNIST [69] dataset and captured more global structure in the Fashion-MNIST [75] and

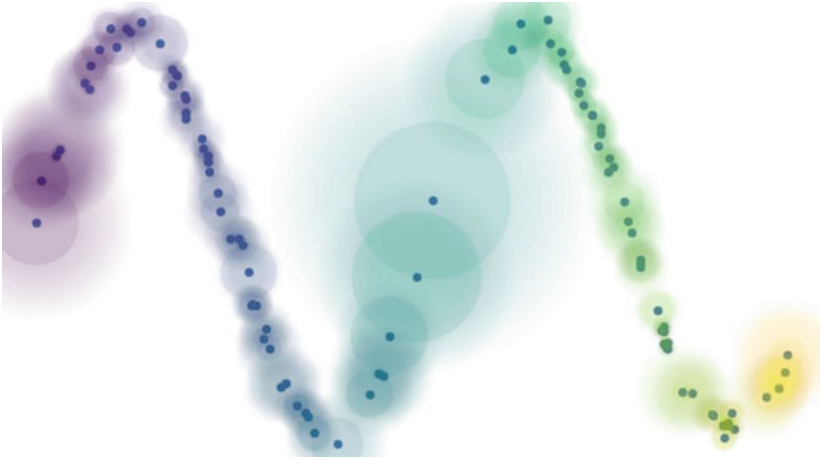


Figure 3.4 Qualitative illustration of the similarity measure in UMAP. Taken from [63]

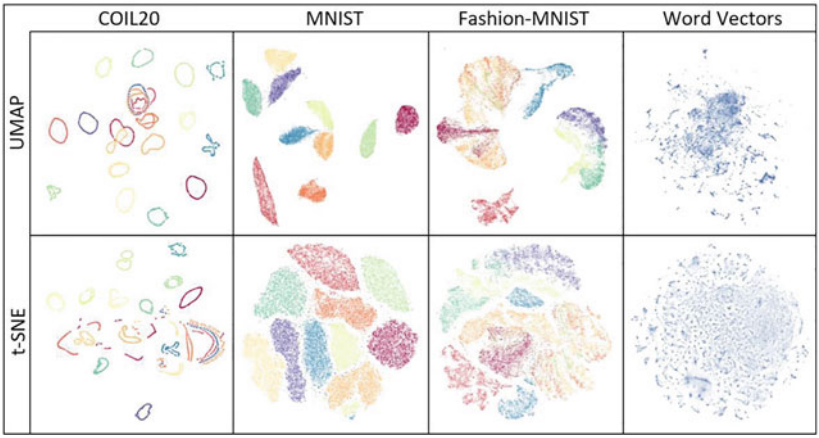


Figure 3.5 Comparison of projection results between t-SNE and UMAP on four different datasets. From left to right: COIL20, MNIST, Fashion MNIST, Word Vectors. Taken from [63]

Word Vectors [76] datasets, separating classes and categories further apart than t-SNE. Since UMAP preserves global structure in the cost function, the separation between clusters is more distinct than in t-SNE. Thus, the separation in a UMAP embedding is a direct indication of the actual separation in the original dataset. A limitation to the interpretability of UMAP projections is the depiction of density. The fuzzy topological structure, like the probability distribution in t-SNE, is adjusted to local densities in the dataset and does not represent differences in densities in the low-dimensional representation. [63]

3.1.2 Visualizing Structure of the Learned Representations

Embedding the high-dimensional learned representations in a two-dimensional projection space allows direct visual inspection of the structure of the representation and the relation between individual data points, which can either be weights, biases, or neuron activations. As the previously discussed embedding techniques aim to preserve the structure when the embedding is calculated based on similarity measures, the retained structure in the projection space gives insights into whether the parameter under consideration, i.e., either a network's weights, biases, or activations, shows any meaningful clustering. Such clustering, however, only reflects inherent structure of the learned representation and does not give any insights in how this structure may or may not be related to humanly interpretable concepts. Such concepts are, for example, given by the labels assigned to individual data points in a supervised learning scenario. For instance, the ten classes in the MNIST dataset correspond to such humanly interpretable concepts. Intuitively, one would expect that the distinction of these concepts should be reflected by, for example, the activations of individual neurons or individual layers of a supervised trained network, which means that these activation values are different depending on the processed input image. However, investigating such questions requires to incorporate information about such humanly interpretable concepts in the visualization of the learned representation in the two-dimensional projection space. This is usually easily achieved by different kinds of color coding the individual data points (cf. Figure 3.3 and Figure 3.5). Generally, an arbitrary color code can be applied, and the correct choice of a color code largely depends on the question that is aimed to be answered. In the presented results throughout this thesis, a number of different color codes are applied to different projection spaces and will be explained in conjunction with the presented results.

3.1.2.1 Magnitude of Neuron Activations

The most intuitive idea of whether a neuron within a neural network is important or not is to make this assessment based on the magnitude of the neuron's activation in response to a specific input. The notion is that if a neuron plays an important role within a network to correctly identify the presented input, e.g., to correctly classify an image in an object recognition task, its activation should reflect its importance for the task. Thus, color coding all data points in a two-dimensional projection space, obtained by PCA, t-SNE or UMAP, on a gradient color map allows to find clusters of neurons that have similar responses to the presented inputs. The color coding may be done according either to the neurons' individual activation values in response to a specific input or to their mean response value averaged over a set of inputs. When comparing activation values of different neurons to each other on an absolute scale, it may be the case that these values vary strongly over several orders of magnitude. This is not usually the case for neurons within the same layer of a network, due to mechanisms likes batch normalization and other regularization techniques frequently employed when training neural networks, however, it is possible to observe such strong variability across different network layers. Thus, an appropriate scaling of the individual activation values, so that all values range between 0 and 1, must be performed to facilitate comparability of those activation values. The specifics of the scaling need to be considered for each case individually depending on the question that is aimed to be answered via the visualization. For instance, in some cases it may be appropriate to scale all values relative to a shared maximum activation value, for example the maximum activation of a network layer, in other cases it may be more appropriate to scale each value separately relative to its individual maximum activation value. These specific are explicitly discussed in later chapters of this thesis when the corresponding results were presented.

3.1.2.2 Selectivity of Neuron Activations

An alternative to considering the mere magnitude of a neuron's activation is to consider its selectivity for individual inputs. The underlying notion is that a neuron, which is highly specialized on a specific input will exhibit a strong selectivity in its activation for that input. More specifically, such a highly selective neuron will respond with high activation values for some inputs or even just a single individual input and with low activation value for all other inputs. In case of a standard classification task, these specific inputs would correspond to images or time series with a specific label that belong to a specific class. Formally, a neuron's activation selectivity AS is given by

$$AS = \frac{\mu_{max} - \mu_{av.else}}{\mu_{max} + \mu_{av.else}}$$

where μ_{max} is the neuron's highest activation for a specific class, averaged over all examples of this class, and $\mu_{av.else}$ is the neuron's mean activation averaged across all examples of all other classes [77]. The AS can take values between 0 and 1, where a higher value denotes a stronger tendency of the neuron to only activate for input from on class. A value of 0 would correspond to a neuron whose activation is uniformly distributed over all classes, i.e., its activation is equal for all classes. In some rare cases, a neuron may be "dead" in the sense that it does not activate at all for any input, which means that the denominator of the AS would be equal to 0. In these cases, the AS itself is defined as 0, which is equal to the interpretation that the neuron shows no selectivity whatsoever.

3.1.2.3 Ablations of Individual Neurons

Besides considering measures to estimate a neuron's importance that are directly characterizing the neuron, an alternative perspective can be taken, which aims to estimate the importance of a neuron for the overall task indirectly. More specifically, the impact of the removal of a neuron on the network's performance is presumably a good measure for how important the neuron is. The notion is that if a neuron is important for the overall task, its removal would severely impact the network's learned capabilities. On the other hand, if a neuron is not important for the overall task, its removal should only have a negligible effect or even no effect at all on the network's capabilities to solve its learned task. Thus, systematically ablating individual neurons, i.e., temporarily disabling them to prohibit any flow of information through them, subsequently testing the network's performance and comparing it to its baseline performance, which is obtained from its healthy and undamaged state, allows to estimate the importance of individual neurons. This procedure is strongly inspired by lesion studies from the field of neuroscience. Typically, a lesion is caused intentionally and damages neural tissue in a specific brain location. Such lesions are typically the result of overpowering a measurement electrode with a strong electrical current, which causes its temperature to rise drastically so that the surrounding local neural tissue is irreversibly damaged in the process. Such lesions are utilized to determine the location of the electrode post-hoc after extended periods of measurement in long studies or to determine the role of the damaged tissue for a subject's ability to perform a cognitive task. The former is done only after the experiment while the latter might be done as part of an experiment. Due to the irreversibility of the caused damage, the procedure is ethically difficult and comes with a number of

ethical and practical problems best illustrated with a famous example from the literature [78]. In the famous case of Henry Molaison, known as Patient H.M., a man with severe epileptic seizures in the 1950s, had part of his hippocampus, the part of the brain, which was associated with epilepsy at that time, removed in order to cure him from his otherwise terminal illness. Although the severity of illness had been drastically reduced due to the lobotomy of the hippocampus, another curious and unexpected side effect had occurred. Molaison was not able to create short term memories anymore, which resulted in the inability to build a coherent understanding of a causal chain of events that unfolded over a period of several minutes or hours. He would constantly ask how he'd to his current location and why he'd gone there. The question arose whether he was rendered unable to learn anything at all. Curiously enough, after the lobotomy, he was still able to acquire complex motor skills and learned to play the piano [79]. Since most functions of playing the piano are purely motor control function, which are largely represented in a specific part of the neocortex of the brain, the motor cortex, the removal of the hippocampus did not affect this ability. However, Molaison was not able to build a memory of his newly acquired ability. As a result, playing the piano would constantly make him realize that there is something wrong with him as he couldn't remember why he was able to play the piano. The result this bizarre scenario of a brain lobotomy was a deeper understanding of the specific functions of individual parts of the brain [80]. The same principle can be transferred to artificial neural networks without all the ethical concerns of causing irreversible damage to the learning system under consideration.

Figure 3.6 illustrates the methodology of ablations in artificial neural networks. An ablation study of a neural network aiming to determine the importance of individual neurons or groups of neurons consists of five distinct steps:

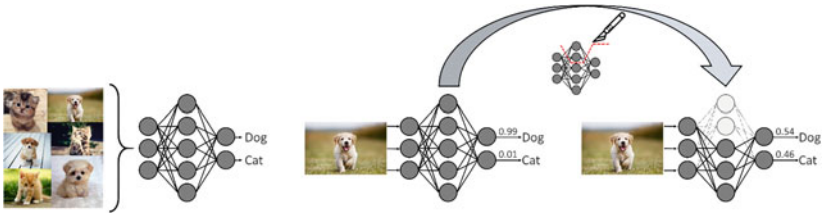


Figure 3.6 Schematic illustration of the principle of ablation studies. Left: A neural network is trained to classify images and distinguish cats and dogs. Right: The ablation process. Parts of the trained network are ablated and the impact on the network's performance is measured to determine the importance of the ablated part of the network for the learned task

1. Build and train a fully intact neural network and establish a baseline performance on some chosen dataset.
2. Systematically ablate individual neurons by setting their incomings weights and its bias to 0, which essentially prevents any flow of information through that neuron as every incoming input is zeroed out.
3. Test and evaluate the damaged network on the same data with which the base line performance was determined. Compare the performance of the damaged network with the baseline performance to determine the extent to which it changed.
4. Determine the role of the ablated neurons for the overall task. For instance, in case of a classification task, determine with respect to which class the network's performance changed, and with respect to which class it remained unaffected.
5. Investigate the network's learned representation for organization and structure with respect to the effects of the performed ablations.

The above procedure can be further extended to groups of neurons, e.g., whole filters in a CNN, or any arbitrary combination of neurons across a network. Such an extension of ablations is inherently complex due to the huge number of different combinatorial possibilities to ablate two or more neurons simultaneously. This issue can be mitigated somewhat by determining a criterion based on which neurons are ablated simultaneously, which requires different measures of similarities between neurons based on which such a decision can be made. There has been done some work aiming to address this issue [81], however, the details on the topic are beyond the scope of this thesis and will not be dealt with in depth. For the presented results, the specific methodology of how neurons were chosen for ablation are presented in the individual subchapters.

In analog fashion to the magnitude and the selectivity of a neuron's activation, the effect of its ablation can be determined either by its total change of accuracy (in a classification task) or by the selectivity of its accuracy change, which is formally described by the class-specific selectivity of the ablation effect SAE:

$$SAE = \frac{\Delta_{max} - \Delta_{av.else}}{\Delta_{max} + \Delta_{av.else}}$$

where Δ_{max} is the highest class-specific change in prediction accuracy caused by the ablation of that unit and $\Delta_{av.else}$ is the average change in prediction accuracy across the other classes caused by the ablation. Since both positive and negative changes of the prediction accuracy are possible, the absolute value is used for

Δ_{max} and $\Delta_{av.else}$ for the SAE to take values between 0 and 1. In cases, in which the denominator is equal to 0, the SAE is set to 0.

3.1.2.4 Gini Importance

An alternative to the previously discussed measures to estimate a neuron's importance for its network's learned task is via its assigned Gini importance when its activations are considered as a separating feature in a tree classification algorithm. Specifically, a tree classifier such as a simple decision tree or a random forest classifier can be trained to predict the class corresponding to a presented activation vector of a network layer or a whole network. This activation vector contains the activations of all neurons in response to a specific image with a given class. The calculated Gini impurity for all activations determines how well the activation of individual neurons allows to separate the set of all activation's vectors according to their class labels. The importance of a neuron is computed as the normalized total reduction of the impurity criterion brought by that feature, i.e., its activations [82]. The notion is that if a neuron's activation values have a high Gini impurity, thus allowing for a good separation of the activation vectors, this particular neuron exhibits a distinctive activation pattern is response to images of a specific class or of several specific classes. It is therefore important for the separation and thus the classification task.

3.2 Delimitation of the Object of Investigation

With the previously discussed methods that are repeatedly used in the studies presented, the object of consideration that was introduced on a general level is further delimited. Specifically, learned representations of neural networks applied in two different domains, namely computer vision and motion control, are investigated in five research studies. Subsequently, the developed methods and results of the research studies are transferred to an industrial use case in two transfer studies to demonstrate how the application of the previously presented methods to facilitate transparency and interpretability can be used to gain a better understanding of the industrial manufacturing process and interpret the decision making of a trained learning model.

3.2.1 Transparency for Computer Vision Models

In the field of computer vision, deep neural networks are widely used for standard tasks such as object recognition, object detection or instance segmentation. Over the years, starting with the success of *AlexNet* in the 2012 ILSVRC, these networks have grown both in complexity and potentiality, consuming huge amounts of image data to solve more and more challenging tasks. This development has reached a point at which these networks outperform human domain experts in applied scenarios such medical diagnostics where the identification of benign or malevolent tumors in medical image data is a routine task commonly performed by the well-trained human eye [34–36]. Despite their extraordinary capabilities, some fundamental issues with the way these highly capable neural networks reach their decisions have been uncovered. For instance, the almost perfect accuracy of a network trained to detect malignant instance of skin cancer tumors was found to strongly depend on the fact whether a ruler is present in the image or not [83]. This circumstance is due to the fact that in case of an identified tumor in an image, its size has to be determined, a measurement process typically performed with a ruler. Thus, the questions arises whether the trained model is nothing more than a good ruler detector or whether it has actually learned something about detecting tumors.

Examples like these have spurred research with respect to investigating the relationship between individual input features, like pixels in an image, and a network's decision. This way, a first level of transparency has been facilitated for why a network makes the decisions it does. In case of the previously discussed example, this level of transparency helps to verify whether the network bases its decisions on the pixels constituting the ruler or the actual tumor. Once a network is sure to have learned something meaningful about what a tumor looks like in fMRI images, the questions arise what it is precisely about the tumor that the network recognizes and how or where that information is represented within the network. Answering these questions would facilitate an increased level of transparency for the network's decision making as it allows to determine the important network structures that are responsible for recognizing a tumor for what it is. In case of a wrong decision of the network, this knowledge about what network parts play an important role for that decision can be further investigated w.r.t to why the network made a wrong decision and how the underlying mechanisms that led to this decision can be influenced to correct the network's error. This way of "debugging" neural networks is fundamentally different from the standard approach of just retraining the network repeatedly with all available data and slightly changed hyperparameters to optimize its capabilities.

Thus, with respect to the field of computer vision, the object of investigation of this thesis is further delimited aiming to lay out the groundwork for investigating the importance of specific network parts for a trained task in a bottom-up manner. To this end, a small shallow MLP network trained to perform an image recognition task on the well-known MNIST dataset is investigated with respect to its learned representation. Specifically, the importance of single neurons for the recognition task overall and for the recognition of individual classes is investigated based on an extensive ablation study of the network's individual neurons. Furthermore, a custom-made CNN trained on the MNIST dataset and two of its more complex variants, the KMNIST dataset [84] and the Fashion-MNIST dataset [75] is investigated with respect to the importance of individual neurons and groups of neurons for the overall task and for the recognition of individual classes. Additionally, the network is investigated with respect to how the learned representations evolve across the depth of the network, from its first layer processing the input images to its output layer producing the decision. Finally, the investigation is extended to the *VGG-19* pre-trained in the *ImageNet* dataset. As the *VGG-19* is much deeper than the previously investigated custom-made CNN, it allows for in depth investigation of the importance of individual network layers.

3.2.2 Transparency for Motion Control Models

Besides computer vision and natural language processing, the domain of motor control poses the third largely researched domain for the application of deep neural networks. Specifically, neural networks are trained to perform a wide range of control tasks of physical bodies, both in simulations and the real world. Such bodies typically consist of a collection of links connected via joints that can be controlled in order to move the body. The controlling agent, typically containing a neural network at its core that is trained to solve a specific task by controlling the agent's body, is able to apply torque to its joints in order to perform task specific and goal-oriented movements. In recent years, a number of examples of successfully trained agents to control robots for pick and place tasks or two-legged and four-legged bodies for locomotion tasks have been demonstrated [85]. However, these kinds of examples also produced agent policies (mostly in simulated environments), i.e., specific ways to control the bodies in order to achieve a given task, which did indeed achieve the given task, but their movements seemed completely unsensible and not at all intuitive [86]. In such cases, the investigation of the controlling network's learned representation promises interesting insights with respect to the question of why the agent produces specific movements, how

those movements originate from its internal neural activities and whether it can be influenced purposefully by manipulating the network's learned representation after training.

Just like in both previous domains, the field of learning how to control a body or individual parts of a body is closely related to neuroscientific studies of the motor cortex and neural activity of the brain when performing movements with the fingers, the hand, and the whole arm. Extensive research has been done on the characterization of individual neurons and groups of neurons in the motor cortex that are responsible to perform specific movements, such as moving the hand into either of the four directions, up, down, left, or right. Many studies have shown the finely tuned properties of such neurons that are selectively activated only when the hand moves into a specific direction [87]. These findings raise the question whether similar structure and organization of neural activities emerges in artificial neural networks that are trained to perform similar kinds of motor control tasks.

In order to investigate this question, the object of investigation of this thesis is further delimited to the investigation of learned representations of state-of-the-art deep reinforcement learning models trained to perform well-known motor control tasks. Specifically, In a first study, the learned representations of an actor network in an actor-critic architecture, which is responsible for learning an agent's policy, i.e., generating the agent's movement, is investigated with respect to its structure and organization in relationship to specific patterns of motion. Furthermore, network ablations are performed to investigate whether specific neurons or groups of neurons can be related to the agent's learned ability to execute specific motion patterns. In a second study, the universality of the results found for the individual agent is corroborated demonstrating that the learned behavior of motor control agents irrespective of the agent network's specific topology and the specific task at hand exhibits some universally occurring patterns with respect to its relationship to the underlying neural activity in the actor agent's actor network. Specifically, an empirical study of a number of agents with different network topologies trained in different control task environments has been conducted. All agent's learned representations were subject to the same analytical methodology to relate its learned behavior to its neural activity in order to find a universal relationship that is not necessarily unique to a specific network and a specific task.

3.2.3 Transfer to an Industrial Application Scenario

In addition to the three previously introduced research domains, which have their origins in the fundamental research of AI, a more application-oriented domain is considered that provides an exemplary use case for the transfer of the previously discussed methods to a real-world manufacturing scenario. For AI to be applicable in such scenarios, deep learning models, i.e., deep neural networks, are required to be transparent and interpretable for domain experts with respect to their decision-making processes. The increased degree of transparency and interpretability facilitates trust in the technology and allows domains experts to assess a model's decision-making in cases where it would lead to actions with high costs, such as the stop of a production line. The justification of such high-cost actions recommended by a deep learning model must be verifiable by human domain experts to ensure, that they are not executed unnecessarily. Additionally, transparency is also a vital criterion for industrial process certification and facilitates new standards with respect to the certification of learning models [88].

This thesis features such a scenario, in which the quality of manufactured car body parts is assessed predictively, i.e., a predictive quality scenario. The quality control process assesses whether the parts are undamaged and shaped correctly or whether they contain cracks that were possibly caused by the manufacturing process, leading to waste products. During the manufacturing process, data was acquired from the deep drawing tool in a press plant of a German car manufacturer. The tool was modified and enhanced by strain gauge sensors that were applied to the blank holder of the tool and by laser sensors at different flange retraction points. These sensors yield time series data with a sampling rate of 2 kHz, which provide insights into the intricacies of the process, i.e., they allow to assess the quality of the deformed metal sheet based on the process data rather than manual visual inspection. The application of deep neural networks for such a predictive quality scenario has been demonstrated many times, however, the transparency and interpretability of such a trained model's decision-making, i.e., in this case the decision to classify a manufactured car body part as either "OK" or "n-OK" is usually neglected.

In order to provide this highly required additional degree of transparency, the object of investigation of this thesis is further delimited to the investigation of the learned representation of custom-tailored CNN that was trained to predict the quality of the manufactured car body parts based on sensor data acquired during the manufacturing process. Specifically, a 1D-CNN was trained on the acquired time series data to classify the available stroke data into two classes, i.e., "OK" and "n-OK". The learned representation of the trained network was investigated

with respect to important time series motifs in the acquired process data that lead the network to the decision to classify a stroke as “n-OK”. Furthermore, network ablations were used to identify the important parts of the network that are responsible for the representation of these time series motifs. This way, the learned knowledge of the network, on which its decision-making is based, can be explicitly extracted, and provided to domain experts to give insights in how the trained network recognizes good and bad manufacturing strokes.

Related Work

4

*"All **progress is experimental.**"*

– John Jay Chapman, American author

The following chapter addresses the **most recent developments in the research branch of XAI** and provides an overview of the related work relevant to the presented results of this thesis. All these contributions relate to methods and approaches used in the empirical studies of this thesis in either one of two possible ways. 1) The presented contributions relate with respect to the general nature and the goal of the employed method to facilitate transparency and interpretability of a neural network or 2) they relate in the sense that they would constitute a direct alternative to the used methods in this thesis.

The related work is organized into three sub-chapters each containing contributions to the state of the art of a distinct nature. Classically, explanations for a trained network's decision have been derived based on the importance of individual input features of the network for the network's output. A variety of methods for different network architectures are presented in the first sub-chapter. The second sub-chapter addresses approaches aimed to facilitate transparency by visualizations of neural network activity with the help of graphical user interfaces that provide different plots and visualizations of a network's learned weights and activations as a result of processing specific inputs. The third sub-chapter addresses methods aimed to provide explanations for the importance and the role of individual structural network compartments, like individual neurons, networks layers or groups of neurons. Specifically, it bridges the research fields of neuroscience and XAI and addresses a recent approach to reverse-engineer trained neural networks [89] that has emerged from neuroscientific studies aiming to understand the mammalian brain. Rather than aiming to provide explanations

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_4

of a network's decision-making process based on the attempt to find underlying bottom-up mechanisms, these studies have investigated large neural networks in a more top-down fashion. The fundamental premise of such studies is that the brain as the object of investigation is so complex, that the classic scientific reductionist approach to understand its behavior is insufficient and more holistic approaches are required. Before diving into the highlighted sub-chapters, the following paragraph addresses some contributions that are unspecific to the four sub-chapter but still constitute important contributions to the general development of the research field.

A general introduction into concepts, taxonomies, opportunities and challenges as well as many of the most widely-used methods to facilitate transparency, interpretability, and explainability for trained models have been covered in comprehensive surveys [90–96]. Typically, research conducted in the field of XAI has been driven by different motives. The most pursued motive is the attempt to improve the structure of trained deep neural networks, which closely relates to the general goal of network optimization. This goal has been typically followed either by building an intuition for how knowledge is represented in these networks [97, 98], or by improving on the choice of hyperparameters and justifying the functionality of training methods, such as batch normalization and dropout [77, 99, 100]. Another highly researched topic is the process of pruning trained neural networks, which focusses on preserving as much of the network's learned capability as possible, while downscaling the networks computational size, commonly by removing parts of the network that don't contribute to the network's ability to perform its learned task [81, 101, 102]. Besides addressing the needs for transparency and interpretability of researchers that typically aim to improve networks during their training and development process, the aspect of interpretability is also important for non-expert users. For example, cases of obvious misclassification of real-world images caused by natural adversarial examples [103] or cases, in which minimal targeted changes to an image as subtle as single pixel alterations lead to misclassification [104] have caused a lack of trust in the decisions of trained neural networks. Such examples are particularly crucial in fields like automated medical diagnostics or autonomous driving, where a faulty decision can have possibly fatal consequences. In these fields, the lack of trust in an AI's decision, i.e., mostly a trained neural network's result, a major impediment to the widespread usage of the models in both the personal and professional domain [105]. A rather drastic example for the importance of interpretability can be taken from the field of medical technology, where the ability for transparent and logical reasoning that underlies certain decisions is a strict requirement for the legal certification of products [106, 107].

4.1 Relationship Between a Network's Input Features and its Output

The following presented related work addresses the issue of transparency and interpretability of a trained model's decisions aimed to uncover the relationship between its input and output. For instance, the importance of individual input features, like individual pixels in an image, for a network's result, i.e., the classification of images into different categories, reveals important and relevant parts of the image that lead to the network's recognition of a specific object, i.e., the class into which the image is categorized. The following paragraphs start with addressing example studies that highlight the general idea of facilitating transparency and interpretability by means of uncovering relations between network input and output in various applications and then presents a set of widely applied methods that facilitate that general idea. Although many of these methods neglect the inner mechanisms of the network or their learned representations completely or just consider them as means to an end utilizing them to uncover said relationships, some of those methods are employed in some studies of this thesis and are thus relevant to understand in the context of interpreting the presented results. Furthermore, the related work constitutes important background knowledge regarding the first research question as the presented methods in this thesis used to answer the first research question stand in direct competition with them.

Some concrete examples providing transparency of network decisions that rely on the relationship between input features and the network's output were reported in the past. For instance, Simonyan et al. proposed a method to generate an synthetic input image representing an individual class by maximizing the class's classification score output by the network in response to the generated image [108]. Papernot et al. as well as Su et al. have reported that marginal modifications of input examples, even as marginal as alterations to a single pixel of an image, can drastically change the prediction of a trained network [104, 109]. Faust et al. visualized a 2D t-SNE embedding of the output vectors of a trained neural networks and added points corresponding to new and unseen images into this embedded space to investigate their similarity based on the distance of the resulting point in the embedded space to other points [110]. In similar fashion, Fong et al. applied perturbations to input images and compared the perturbed images with respect to changes in the network's prediction based on saliency maps created for the images [111]. In equally similar fashion, Zintgraf et al. identified areas in input images with positive and negative influence on a network's prediction [112]. Arras et al. quantified the contribution of individual

words of a text input by deleting individual input features to investigate the influence of these deleted features on the classification result of a network trained to categorize text [113]. They found that deleting words that apparently play an important role for the classification of the text, resulted in a strong decrease of the network's performance. Zhou et al. found that CNNs trained to classify visual scenes learn representations of objects and elements typically found within those scenes during the training process [114]. They conducted systematic studies, simplifying images by removing certain elements from them until misclassification, to determine which input elements are most relevant for the network's prediction. Finally, one of the earliest approaches to facilitate transparency of vision networks has been conducted by Zeiler's and Fergus' famous deconvolution. They identified relevant parts in images by covering them up and analyzing the resulting feature representations of the famous *ZFNet* [115]. Greydanus et al. transferred the investigation of the relationship between model input and output to the deep reinforcement learning domain, specifically Atari agents. [116] Similarly, Hilton et al. applied these interpretability techniques to a model trained to play the video game CoinRun [117]. They build an interface for exploring the objects detected by the model, and how they influence its value function and policy [118].

On a more general and methodical level not directly connected to a particular application, the earliest contributions to uncover the relations between model input and output are given by two model-agnostic methods for indirectly explaining black box models, which are Bayesian Rule Lists (BRL) [119, 120] and Black Box Explanations through Transparent Approximations (BETA) [121]. Both methods aim to create predictive models that are interpretable by human experts. They are based on decision lists consisting of a set of "if–then" rules (e.g., if hypertension, then stroke) that discretize a high-dimensional, multivariate feature space into a set of simple, easily interpretable decision statements that are human interpretable. Another more general and model agnostic method for explaining the influences of individual input variables on the prediction of a learning model is Local Interpretable Model-Agnostic Explanations (LIME) [122]. LIME aims to understand the behavior of a trained model based on small perturbations of input variables and the study of the resulting changes in model predictions. Interpretability for humans is provided by the fact that the perturbations of the input variables are comprehensible to humans even though the model may have learned a significantly more complex representation of these input variables.

In contrast to model agnostic methods, a number of methods specific to neural networks have been developed. A prominent representative of these methods is Layer-wise Relevance Propagation (LRP), which makes the predictions of deep

neural networks interpretable based on the input variables of the network [123]. LRP computes scores for image pixels and image regions denoting the impact of the particular image region on the prediction of the classifier for one particular test image. Another method with a similar goal is Deep Taylor Decomposition (DTC) [124, 125]. DTC is based on propagating the network output back through the network using a predefined rule set. As a result, it produces a decomposition of the neural network output that is mapped on the input variables and allows the visualization of the relevance of individual input variables to the network prediction under investigation. A third prominent method with the same goal is Gradient-weighted Class Activation Mapping (Grad-CAM), which, similar to LRP and DTC, allows inferences about the relevance of a network's input variables to a particular network prediction [126]. In computer vision, Grad-CAM allows the creation of a coarse localization map for image data, indicating which sub-regions of an image led to a particular network prediction. A method for the study of temporally sequential events is provided by the Reversed Time Attention Model (RETAIN) [127]. RETAIN provides an attention mechanism that pays particular attention to specific events in the past and relates these events to the prediction of a network. For example, RETAIN helps interpret predictions of heart failure and allows inferences about the relevance of past symptoms or events, such as skin problems, skin disease, removal of skin lesions, cardiac arrhythmias, valvular disease, and coronary atherosclerosis, to the occurrence of heart failure.

4.2 Visualization of Network Properties and Graphical User Interfaces

The conducted studies in this thesis utilize a number of ideas to visualize learned representations of trained neural networks, which is addressed in the following presented related work. However, some of the visualizations employed in these studies expand on the employed visualizations in the related work with respect to their meaningfulness for the data and the environment to which they are related. More specifically, while almost all the visualizations in the framework of this thesis rely on the visualization of embedded network properties similar to the related work, they expand on their utilization for ablation studies and visualizing the effects of such ablations on the visualizations of a network's learned representations. Thus, these expanded visualizations directly contribute to answering the second research question as they help to uncover structure and organization of the learned representations of the neural networks in question. Furthermore, the use of such visualizations for the domain of deep reinforcement learning is a novel

approach to facilitate transparency and interpretability for the network's learned representations in that specific domain. Specifically, ablation studies conducted on the trained networks and the subsequent comparison of these visualization between the healthy and the damaged network allow to attribute meaning and purpose to individual neurons and groups of neurons for the learned control tasks. Furthermore, these visualizations are heavily utilized to investigate the link between an agent's learned behavior and its learned representations. Thus, the related work and the expanded utilization in the domain of deep reinforcement learning presented in this thesis is directly related to answering the third research question, as it uncovers the role of individual network elements for the emergence of learned behavior. The remainder of this sub-chapter presents the related work for different approaches of visualizations of network properties in various domains of deep learning research.

Humans are particularly visually driven creatures mostly relying on their sense of vision to understand their environment. Thus, it does not surprise that the approach of understanding neural networks based on interactive visualizations of network properties, like its weights and activations, has become a popular one spurring the research of facilitating their transparency and interpretability.

One of the most intuitive visualization approaches has been reported by Harley et al., who created an interactive tool to visualize the activations of individual neurons in a CNN that is trained on the MNIST dataset, which allows the user to manually draw digits onto a sketch map that are then classified by the trained network [128]. Smilkov et al. have introduce the TensorFlow Playground aiming to provide non-expert users an intuitive feel for the effect of different hyperparameters and structural variations of neural networks performing simple supervised classification or regression tasks [99]. Chung et al. have introduced a real-time visual analytics tool to visualize filters in a CNN, which can be ordered and ranked by similarity calculated via t-SNE, and visualize their weighted connections to filters in preceding and subsequent network layers [129]. Their tool allows to investigate a network at distinct stages during the training process providing potential insights into the emergence of the structure and organization of a network's learned representation. Liu et al. have exploited the visualization of ordered weighted connections between feature maps with a strong class-specific activation to investigate the structure of the classes' representations in a CNN and the location of the relevant knowledge within the network [130]. They further demonstrated how to apply these insights to explain a plateau of during the training process of the network, stopping it from further improving its performance. Kahng et al. addressed the unique design challenges for a visualization tool utilizing deep learning models that are deployed in industry and process the large-scale

datasets that are commonly found in the respective industrial use cases [131]. They visualized the deployed networks through a computational graph, whose interface includes the option to analyze specific network properties, such as the average neuron activations per class or the corresponding 2D projections of these activation. Investigating the projections of different network layers, they found that they became more distinct in deeper layers of the network and explained cases of misclassification.

Visualizing network input rather than its properties, Mahendran et al. conducted a direct analysis of the visual information contained in learned network representations. Specifically, they asked the question to which extent it is possible to reconstruct an image given the learned encoding of the image by a trained network. [132]. Following this inversion of learned representations approach, they showed that several layers in CNNs retain photographically accurate information about the reconstructed images with different degrees of geometric and photometric invariance. Similarly, Yosinski et al. found that learned knowledge about a class is stored more locally within deeper networks on a trained network and more distributed in upper layers [133]. The importance of the visual analysis of hidden layers rather than merely network input and output has been stressed by Olah et al., who proposed the combination of feature visualization [134] to demonstrate what input features would maximize a filters activation [135]. They formulated the task as a maximization task of an individual network filter's activation in response to the variation of a synthetically generated input image. Building up on that work and taking it a step further, Carter et al. as well as Li et al. not only visualized features as they are detected by different filters, but also arranged them in a two-dimensional UMAP projection called *Activation Atlases* to give them spatial meaning in relation to each [136, 137]. This projection was used to highlight features with a high influence on the prediction and to give an intuitive overview of where the knowledge is represented within the trained network. Consistent with previous findings, their results suggested that features become more specific along the layers of the network towards its output layer. Another interactive browser-based visualization tool was developed by Dibia [138]. The tool provides the visualization of several well-known and widely used neural network architectures pre-trained on different datasets. Amongst other things, users can interactively investigate the networks to look at visualizations of feature representations in specific layers, their similarities as well as UMAP embeddings of the different layer activations.

Using embeddings like t-SNE and UMAP as a means of investigating the complexity of the structure of learned representations of trained networks has been established as the go-to method for visual understanding. Elloumi et al.

analyzed a Deep Automatic Speech Recognition model by applying t-SNE to the activations recorded in different layers [139]. Their obtained projection visualizes differences in speech style and accent as well as where those differences were most prominent in the trained network. In similar fashion, Belinkov et al. have investigated the activations of end-to-end speech recognition systems based on CNN and RNN layers, which were trained to directly predict text from input acoustic features. They used a pre-trained model to generate frame-level features which are given to a network trained on frame classification into phones [140]. They found that activations from the first CNN layer produced better results than the original input, while the activations from subsequent RNN layers generally produced worse results. A different approach was taken by Aubry et al., who changed input features of three pretrained networks in a controlled manner in such a way that they'd vary individual scene factors that occur in natural images, such as object style, 3D viewpoint, color, and scene lighting configuration [141]. They reduced the dimensionality of the hidden layer activations via PCA to determine the importance of the changed feature or the affected scene factor for the network's prediction capabilities. They showed strong qualitative differences between the importance of changed features for the three investigated networks. Rauber et al. further analyzed the hidden layer representations of input data by applying t-SNE to the layer activations not only in different layers, but also in different training stages [142]. They showed that the distinctness of clusters depends on spatial and temporal characteristics of the learned representation, i.e., on the layer within the network and on the point in time during the training process. Specifically, the clusters became more distinct with increasing layer depth as well as with increasing training time. Additionally, they introduced *neuron projections*, in which they visualized the activation of neurons of an individual layer in two dimensions using the absolute metric MDS [143] on a matrix of pairwise similarities of neuron activations. Color-coding the individual points representing individual neuron activations through a measure for class selectivity, they found clusters of similar and class-specific neurons.

4.3 Investigating the Importance of Individual Network Components

Besides the relationship between a network's input and output as well as the mere visualization of its network properties, another way to facilitate transparency and interpretability for its decision making is based on the investigation of its individual components and the role they play for its learned task. To this end, a number

of example studies have been reported that investigated different networks trained for different tasks with respect to the importance of its individual network compartments such as individual neurons, groups of neurons or whole layers. The reported results in this thesis, generated over the past three years, largely complement the work related to this particular research branch, which is presented in the following sub-chapters. The main idea is to determine important network compartments and attribute individual meaning to them. Similar to the related work of OpenAI's clarity team (see below), i.e., to adopt a holistic perspective to investigate the learned representations of trained neural networks, the conducted studies in this thesis follow the same high-level goal but differ in some particular respects. Specifically, the neuroscience inspired methodology of ablations, that is absent in the works of OpenAI's clarity team, is thoroughly employed in the reported results and complements the state of the art in that regard. Thus, the related work presented is directly connected to the third and fourth research question, as it demonstrates how the current state of the art with respect to determining the roles of network compartments has emerged from the XAI research field and how some of the work has been inspired by approaches commonly employed in neuroscience. The following paragraphs and sub-chapters present the related work starting with general contributions to the investigation of network compartments, continues with related work regarding ablation studies and emphasizes the work of OpenAI's clarity team as one of the first large scale investigations inspired by the neuroscientific approach of reverse engineering large neural systems.

4.3.1 Miscellaneous Contributions

Bau et al. have reported their analysis of individual neurons of a neural machine translation model to determine whether learned knowledge is fully distributed across the network or if some of it can be attributed to individual neurons. [144]. They developed unsupervised methods to determine important neurons and ranked them by importance with respect to finding linguistic properties which they represented. In contrast, Molchanov et al. identified unimportant neurons in a network for pruning purposes [81]. They characterized neurons based on Taylor expansion that approximates the change in the cost function induced by these neurons estimating their importance based on the extent to which the cost function changes. Putting particular emphasis on the investigation of the representation of humanly interpretable concepts by individual network compartments, Bau et al. introduced network dissection, a method to quantify the degree of recognition of humanly interpretable concepts within data by individual filters of a trained CNN

[145]. They argued that the larger the number of neurons with a clear recognition of one or multiple concepts is, the higher the network's overall interpretability should be ranked. Similar investigations have been conducted for language recurrent language models. In search of neurons, which show a strong feature specific activation, Radford et al. identified a "sentiment" neuron, which discriminated between positive and negative sentiment of generated texts [146]. They showed that fixing the neurons activation value during the generative process of the model simply to be positive or negative generates samples with the corresponding positive or negative sentiment. Complementary, Agrawal et al. reported a neuron with a strong feature specific activation and, however, they additionally found that most neurons in the investigated CNN were multifaceted, rather than representing just a single distinct feature found in the data [147]. The universal nature of these findings was confirmed by Nguyen et al., who described that most neurons activate strongly not just for a single class in a classification task but rather for a number of different classes [148]. Been Kim et al. have introduced concept activation vectors (CAV), which provide an interpretation of a network's learned representation in terms of human-friendly concepts. They further introduced the process of testing with concept activation vectors (TCAV), which relies on directional derivatives to quantify the degree to which a humanly defined concept is important to a classification result of a trained network, e.g., how sensitive a prediction of wristwatch is to the presence of a circle [149].

Taking a slightly broader view on knowledge representation, Belinkov et al. analyzed the learned representations of individual layers in a recurrent language classification model performing part-of-speech and semantic tagging tasks [150]. They found that early layers were most relevant for the identification of semantic properties, whereas deeper layers represented part-of-speech and morphological features, thus confirming the localization of specific parts of the learned representations in the network. Durrani et al. conducted single neuron analysis using core linguistic tasks on pre-trained language models and found that small subsets of neurons predict these tasks, with lower-level tasks (such as morphology) localized in fewer neurons, compared to higher level task of predicting syntax. They further found differences to their finding across different language models. For example, they reported that neurons in *XLNet* [151] are more localized and disjoint when predicting these tasks compared to BERT and others, where they are more distributed and coupled [152]. Complementary to the investigation of single neurons, Filan et al. investigated the clusterability of groups of neurons in neural networks. Specifically, they investigated the weights of trained neural networks scrutable internal structure by answering the question how well a

network can be divided into groups of neurons with strong internal connectivity but weak external connectivity, i.e., clusters. They found that networks are typically more clusterable after training, and often clusterable relative to random networks with the same distribution of weights [153]. Similarly, Csordas et al. investigated the modularity of neural networks and investigated whether reusable functions emerge in these networks during training. Such functions offer numerous advantages, such as compositionality through efficient recombination of functional building blocks, better interpretability of the role of sub-areas of a network or preventing catastrophic interference. They used binary weight masks to identify individual weights and subnets responsible for such specific functions. They demonstrate how common NNs fail to reuse such submodules and offer new insights into the related issue of systematic generalization on language tasks [154]. Besides the general contributions detailed above, a number of more specific contributions have been conducted particularly employing methods inspired by neuroscientific studies. One of those methods are ablation studies, which have been used to determine the role of individual network compartments, i.e., mostly individual neurons.

4.3.2 Ablations Studies

In neuroscience, ablation studies were utilized to uncover structure and organization in the brain, i.e., a mapping of features inherent to external stimuli onto different areas of the neocortex [56–58]. Inspired by this neuroscience perspective, ablations have been transferred to artificial neural networks and applied in a number of different investigations. Dalvi et al. specifically created a toolkit to visualize neurons and layers of a trained network, which allows to manually ablate or manipulate individual neurons to study their effect on the networks performance [155]. Li et al. ablated individual neurons of a convolutional neural network aiming to identify neurons, which could be removed with little to no effect on the network's performance and found that the removal of whole filters may only result in a marginal decreases of network performance [101]. Similarly, the famous *AlexNet* [27] was subject to ablations and weight perturbations in a study conducted by Cheney et al., who found it to be robust against such manipulations [98]. In contrast, Bau et al. found that targeted ablations in a generative adversarial network [156, 157] trained to generate full images erased specific classes such as chairs or windows from the output images [157, 158]. They conclude that individual neurons play a significant role representing key features of the GAN's latent space, without which particular objects could

not be generated in the output images. Li et al. investigated a natural language processing model and erased parts of a word or whole words of the input vector and hidden units [159]. Comparing the logits output of the model before and after ablations, they observed negative and positive class-specific effects on the network's performance. These effects were generally more pronounced as a result of ablations in layers closer to the input. Moreover, they noticed less class-specific effects when the network was trained with regularization techniques like batch normalization. Dalvi et al. hypothesized in one of their studies that neurons with larger weights should be more important for the prediction accuracy [160]. They verified this hypothesis by ablating neurons with the largest weights and showed that the negative effect on the network's performance was greater than as compared to the effect of ablations of neurons with smaller weights. Furthermore, they identified neurons, for which the ablation caused class-specific changes in of the network's performance rather than an overall change. Instead of a neuron's weights, Morcos et al. focused on their activations and calculated the class-specific selectivity of each neuron based on its activation distribution for the individual classes [77]. They found that ablations of highly selective neurons were not more impactful than ablations of less selective neurons. However, they found a positive correlation between the network's reliance on single neurons and its capability to memorize, i.e., overfit the training data compared to its capability to generalization, i.e., transfer to the test data. Furthermore, they found that batch normalization used during training decreased the networks reliance on single neurons and made the network more robust against ablations. Consistent with previous findings, they confirmed that networks are more sensitive to ablations in earlier layers than in deeper layers.

The above contributions constitute a collection of the most impactful studies reported in the last few years that aimed to determine the role of individual network compartments. Importantly, the results of these studies are based on approaches individually tailored to the investigated networks. Their transfer to other domains, other networks and other learning tasks is not straight forward and would require a great amount of individual customization. Thus far, besides the shared goal and general idea of the used methods to investigate individual network compartments, there is no standard approach universally applicable to a wide range of network architectures, domains, and learning tasks.

4.3.3 Reverse Engineering of Neural Networks

A recent undertaking with respect to investigating individual network compartments has been conducted by OpenAI's clarity team, which took on the challenge to heavily reverse engineer state-of-the-art neural networks [89] aiming to study the connections between individual neurons and assuming to find meaningful algorithms in their weights. The approach is based on three speculative claims [161]:

1) "Features are the fundamental unit of neural networks. They correspond to directions […] [and] can be rigorously studied and understood."
2) "Features are connected by weights, forming circuits […, which] can also be rigorously studied and understood."
3) "Analogous features and circuits form across models and tasks."

Regarding the first claim, directions are linear combinations of neurons in a layer. Thus, a direction can be thought of as a vector in the vector space of neuron activations in a network layer. The concept of directions is not only useful but arguably an inevitable necessity. Considering that the number of neurons in the brain as well as in neural networks is limited, the idea that each meaningful concept hidden in a dataset – which can be, for example, all visual stimuli ever perceived by the brain's visual system, or a large image dataset processed by a large image recognition network – runs into the combinatorial problem that there's likely more concepts that need to be represented than there's neurons in the brain or the neural network. Thus, a combination of fundamental features of the dataset, i.e., individual pixels of an image, forming more complex features, e.g., a weighted combination of pixel values, that are represented jointly by a set of neurons seems like a possible way to address this issue. Regarding the second claim, circuits are subgraphs of a neural network. They consist of a set of features (cf. first claim) and the weighted edges that connecting them to the rest of the network that they are part of. The concept of a circuit is relying on the fundamental idea of deep learning, i.e., that learned representations emerge across the hierarchical structure of network layers. In the same way, circuits are hypothesized to emerge during training by connecting features in individual subsequent network layers. The third claim is a claim regarding the universality of the first two claims. It postulates the emergence of fundamental features and circuits across different models independent of their architectures and different tasks independent of their goals. A logical conclusion would be that the networks

learn representations of their training data based on fundamental characteristics of the data, rather than the specifics of the network or the learning task.

OpenAI's clarity team has investigated these three claims in a number of studies and demonstrated the emergence of basic features in the *InceptionV1-Net* [162]. Such features correspond to the representation of Gabor filter-like structures, colors, color contrasts, textures, i.e., spatial frequencies, or straight and curved edges with different angles [163]. They further demonstrated the emergence of circuits connecting these basic features to form more complex representations like curve detectors [164, 165] or high-low frequency detectors [166]. They investigated the universality of these features and circuits and showed that they emerge in different state-of-the-art networks trained in the *ImageNet* [25] dataset in different equivariant versions. They refer to that phenomenon as the emergence of equivariant features, i.e., analogue features as the result of simple transformations like rotations, translations, scaling or hue shift and hue rotation [167]. They further expanded on the perspective of circuits and found the emergence of larger structural phenomena called branch specialization [168]. Branch specialization refers to the splitting of network layers into distinct branches. Within the layers, neurons and circuits tend to self-organize to represent related functions into each branch to form larger functional units. They found evidence for the implicit existence of such structures in different neural networks that do not exhibit branches in their inherent architecture (like the original *AlexNet*), and that branches merely reifying these structures. Another universally observable phenomenon corresponding to a structural phenomenon on a larger scale, i.e., on the scale of whole network layers, is called weight banding [169]. Weight banding is consistently observed in the last convolutional layer of vision models relying on global average pooling, like the *InceptionV1-Net [162]*, the *ResNet-50 [170]* or the *VGG-19 [171]*, and describes the visualization of the spatial structure of weights in that layer, which forms horizontal stripes. It is unclear whether this consistently observed phenomenon is "good" or "bad", but it reflects a consistent link between network architecture decisions, like utilizing global average pooling, and the structure of the resulting trained weights. For all these universally occurring phenomena, it remains to be investigated whether knowledge about how they arise from a network's structural design is useful for future network designs.

Research Studies

5

The conducted studies in this thesis are strongly inspired by the general ideas of the previously presented milestones in the field of neuroscience. Their overarching goal is to demonstrate the transfer of the general notion of the neuroscientific approach to investigate neural systems with respect to how they represent knowledge. To this end, the previously presented methods are selectively transferred to artificial neural systems trained in two distinct domains to investigate the individual networks in relation to their respective learning tasks.

The presented studies are categorized in two sub-chapters corresponding to the two distinct domains. The first sub-chapter addresses the well-researched domain of computer vision and presents the results of three studies, which investigate the learned representations of neural networks trained on image data. Specifically, the neural networks are trained on supervised learning tasks and the results address the question of how the networks learned to represent the distinction between the individual classes of the data and how this distinction is related to the neural activity of the network. The second chapter addresses the equally well-researched domain of deep reinforcement learning, specifically in the sub-domain of motor control. Two studies present the results of the investigation of learned representations of neural networks trained on widely used benchmark motor control tasks to investigate how behavior of a trained agent is related to the neural activity of its control network.

The following listing gives a brief overview of the five research studies, their connection to each other and their relation to the four research questions. All the presented studies have been peer reviewed and published in various conference proceedings or are in print at the time of writing. For the most part, the results

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-658-40004-0_5.

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_5

of the studies rely on the methods outlined above, however, some studies utilize specific methods that will be briefly outlined in an appropriate section for reasons of self-consistency.

Computer Vision

1. The first study in the domain of computer vision introduces the general concept of ablations to investigate individual neurons in a small computer vision network trained to perform an image classification task. It directly addresses the first research question of how to determine the importance of individual neurons for a network's learned ability to solve a specific task. Its results constitute an important proof-of-concept for ablation studies to be a) transferrable from neuroscience studies to empirical studies of artificial learning systems and b) yield valuable insights about the role of individual neurons within a network. Furthermore, the results show that individual neurons can be categorized and grouped according to their specific roles, which addresses the second part of the first research question regarding groups of neurons and their relation to the learned task.
2. The second study directly builds upon the results of the first study and transfers the approach of ablations from a small and shallow MLP trained on a small dataset to a large state-of-the-art computer vision network trained on a large and more challenging dataset. In similar fashion, the study investigates individual neurons in different network layers and utilizes ablations to determine their importance of the learned task. Furthermore, it investigates to which extent the damage caused by ablations and the negative impact on the network's performance can be repaired by subsequent recovery training. The study solidifies the results of the first one and demonstrates the transferability of the approach to state-of-the-art networks and it's scalability to larger network architectures. The results of the study address the first research question and in addition to corroborating the previous results regarding the investigation of individual neurons and groups of neurons, they constitute the first step in describing structure and organization of learned representations in relation to the depth of the network. Specifically, the results show that different layers of the network are selectively important for specific sub-tasks and thus address the second research question regarding the structure and organization of learned representations qualitatively.
3. The third study extends the investigation of the previous study of groups of neurons regarding their role for the learned task of a network. Furthermore, extending the investigation of the first two studies on single individual

datasets, it addresses the universality of the previously found results and expands the investigation across different networks and different computer vision datasets. The results demonstrate the existence of functional neuron populations, i.e., groups of individual neurons jointly representing individual subsets of the processed data, thus, directly addressing the first research question. Furthermore, the study investigates the emergence of learned representations along the depth of the network and characterizes how it is structured and organized with respect to different network layers. Thus, the results solidify the initial notion of the second study and address the second research question in similar but more rigorous fashion.

Motor Control

4. The fourth study introduces the transfer of ablations to the paradigm of deep reinforcement learning. Specifically, it transfers the previous approaches to neural networks trained to perform motor control tasks, which require to represent structure and organization within their weights not directly connected to a predefined external structure given by labels as is the case for supervised learning tasks. Thus, the domain of deep reinforcement learning allows to investigate the relationship between an agent's learned representation and its emerging behavior that is exhibited to solve the given motor control task. Besides supporting the previous findings regarding the first and second research questions, the study addresses the third research question and demonstrates the relation between the structured and organization of the network's learned representation and its exhibited behavior. Specifically, ablations are used to demonstrate how damaging the learned representation affects its behavior in a specific manner and reveals the organization of that representation in terms of sub-dividing the overall control task into distinct sub-tasks.
5. The fifth study complements the previous one and addresses the generalization of the previously obtained results. The universality of the results is shown by investigating a large set of different networks trained to solve various control tasks. The results of the study address the third research question specifically and demonstrates how an agent's learned behavior emerges from its neural activity, which is shown to exhibit structure and organization in relation to the nature of the learning tasks.

The concept of the studies is deeply inspired by Hubel's and Wiesel's investigations of the activity of single units in the cat striate cortex in response to

visual stimuli and is transferred directly to neural networks performing image recognition tasks. However, contrary to Hubel and Wiesel, there is no notion of a receptive field of neurons established as such a local receptive field is already a fundamental part of the architecture of CNNs. Furthermore, the analysis of neural activity in groups of neurons that take on similar tasks and their relation to features of the input is directly transferred from neural tissue imaging studies to the neural networks investigated in the studies. The most important distinction here is the absence of any temporal aspects regarding the processing of visual information. Since the propagation of information through an artificial neural network is not time dependent, unless this temporal aspect is specifically given by the structure of the data, i.e., time series data, and by the according architecture of the neural network, i.e., a recurrent neural network, there is no equivalent to the temporal resolution of today's imaging methods, like fMRI. This temporal aspect is only considered in the studies six and seven, which deal with time series sensor data and investigate temporal aspects of the data and how they are represented by the trained networks.

Another direct transfer is the attempt to map the activity of neurons onto some inherent structure of the neural network. This structure is typically determined by a similarity between neurons that is computed by t-SNE or UMAP embeddings. Contrary to neuroscientific studies of the brain, studies on the structure of neural networks are not as straight forwards as they do not possess such an obvious structure that is comparable to the anatomy of the brain. Specifically, the neurons of a neural network with a given set of weights can be re-arranged in various ways, changing its 'anatomy' without affecting its computation. Thus, the results in this thesis deal with the key question of defining what neurons are close to each other and how to define an area of a network (cf. RQ2). Similar to the neuroscientific studies, once this structure is determined, the neurons' activation patterns are investigated in relation to this structure as well as to the network's inputs and outputs. The most extensively transferred method is inspired by neuroscientific lesion studies, i.e., ablation studies. Specifically, individual neurons, combinations of neurons and larger groups of neurons are ablated from various networks trained on various environments to investigate their roles to perform the learned tasks in the different domains of vision, and motor control.

5.1 Investigating Learned Representations in Computer Vision

In the context of this thesis, the conducted research in the domain of computer vision and the corresponding results have been published and peer-reviewed in a series of papers throughout 2019 until today [172–174]. The following subchapters largely repeat content of the papers and in part replicate some text passages with some minor changes or without any change. Additionally, the presented results are discussed in the context of the thesis and specifically in relation to the research questions.

5.1.1 Research Study 1: Characterizing Single Neurons in a Shallow MLP

The first research study [172, 173] introduces the use of network ablation to investigate the importance of single neurons to a network's learned ability to classify handwritten digits of the classical MNIST dataset. The network, a small and shallow MLP, was subject to single neuron ablations and pairwise neuron ablations to determine the impact of their removal on the network's overall classification performance as well as its class specific performance. Thus, the study directly addresses the first research question and demonstrates the feasibility of network ablations to determine the importance of individual neurons for a network's learned task.

5.1.1.1 Key Contributions of the Study

The first key finding of the study addresses the first research question demonstrating that neuron ablations are a feasible approach to characterize neurons within a trained network with respect to the role they play for solving a learned task. It is closely related to the work of Morcos et al. [77], who showed similar categorization of individual neurons based on the selectivity of their activations. Specifically in case of the presented study, the neurons in the investigated neuron were shown to fall into one of three distinct categories:

1. Neurons that are universally important for the performance of the network. The ablation of these neurons causes the network's performance to suffer greatly and across all classes or at least many of them. This suggests that these neurons learned to represent features that are universally important to recognize digits.

2. Neurons that are selectively important for the performance of the network. The ablation of these neurons causes the network's performance to suffer only regarding a specific class of digits. This suggests that these neurons learned to represent features that are unique to this class, as the network's ability to recognize all other classes remains largely unaffected.
3. Neurons that are negligibly important or not at all important for the performance of the network. The ablation of these neurons only negligibly, or on many cases not at all whatsoever, affect the network's performance. This suggests that these neurons didn't learn to represent any meaningful features necessary to recognize the digits.

The second key finding further addresses the first research question and gives an equally direct answer as the first key contribution detailing the characterization of individual neurons with respect to their importance of the network's learned task. Specifically, it demonstrates that importance of a single neuron for the network's performance correlates with the extent to which the neuron's weight distribution of incoming connections after training differs from the initial randomly initialized distribution. This result implies that the importance of a neuron for the network's learned task can be attributed to an inherent property of the neuron and/or the network and does not necessarily require a functional test, i.e., to perform thousands of inferences testing the network's change of performance as a result of ablations.

The third key finding of the study addresses the first research question and expands on the first finding showing that learned features by individual neurons are redundantly represented, i.e., that features are represented individually by more than just a single, individual neuron. This leads to a certain degree of robustness of the network to ablations as the ablation of either individual neuron does not affect the network's performance because of the redundant representation of the feature represented by the ablation neuron. Rather, all neurons representing that feature need to be ablated in order to affect the network's performance.

The fourth key finding addresses the first research question and further expands on the first and third finding showing that ablations, despite having a general negative effect on the overall classification performance of the network, consistently showed positive effects on the classification performance for specific classes. Specifically, the ablations of individual neurons lead to the improvement of the classification performance for individual classes. This suggests that the learned representation of a trained network may be purposefully manipulated to increase its classification performance beyond the local optimum that was reached during training.

5.1.1.2 Methods and Experimental Design

To answer the first research question and investigate the importance of single neurons within a network, a small, shallow MLP was subject to the investigation via ablations studies due to its simple architecture and the easy access to its individual neuron's properties, i.e., their weights, biases and activations. The simplicity of the network was deliberately chosen to emphasize the focus of the study on a proof of concept of the feasibility of ablation studies for investigating individual neurons of a neural network without confounding the results with possible effects resulting from complex architecture choices of, for example, large state of the art CNNs. However, an example of such a network is investigated in the second study, which complements the results of the first one. With the same argument in favor of simplicity over complexity to focus on the development of the methodical approach, a simple and well-researched dataset in the field of computer vision was used to train the network, the MNIST dataset. The properties of the dataset, i.e., it's well separable classes, it's balanced distribution of samples across classes and it's favorable ratio of number of samples per class to the total number of classes for the small and shallow network to reach a sufficiently high enough accuracy to conduct the ablation studies without extensive computational cost or manual effort for finetuning parameters.

The MLP investigated consists of an input layer comprising 784 neurons, which correspond to the 28×28 pixels of the input images of the MNIST dataset. Furthermore, the network has two hidden layers with 20 and 10 hidden neurons, respectively, all using standard ReLU activation [175]. The output layer contains 10 neurons with SoftMax activation [176], which correspond to the 10 classes of the dataset. The network was trained for 100 epochs on the $60,000$ images of the training set and reached an accuracy of 94.64% on the $10,000$ images of the test set. After training, ablations of single neurons were performed by manually setting the weights of all incoming connections to zero, essentially preventing any kind of information flow through this neuron. Since the network was trained without biases, zeroing a neuron's incoming weights is equivalent to removing the neuron from the network altogether. In order to investigate the effect of single neuron ablations, the performance of the damaged network was determined on the test set and compared to the performance of the original, undamaged network. A t-SNE embedding of the complete $10,000$ images of the test set were used to visualize the effects of the ablations and support the quantitative results qualitatively. Each single neuron of the 20 neurons in the first hidden layer as well as each single neuron of the 10 neurons in the second hidden layer was ablated individually and the resulting change in network performance was recorded. Additionally, neurons within a layer were ablated in pairs. Each

possible combination of pairs for the 20 neurons in the first hidden layer was ablated, i.e., 190 different combinations, and each possible combination of pairs for the 10 neurons in the second hidden layer was ablated, i.e., 45 different combinations. The impact of the pairwise ablations was recorded and compared to the summed impact of single neuron ablations of the corresponding neurons in order to investigate if neurons could possibly represent information redundantly.

5.1.1.3 Results

Figure 5.1 shows a t-SNE visualization of the 10,000 digits in the test set and serves as a basis for the visual evaluation of the effects of ablations. As t-SNE aims to preserve the local and global structure of the data when embedding the original 784-dimensional dataset into the 2-dimensional space, it allows to investigate whether this structure is represented in an organized manner in the network. The overall accuracy of the trained MLP on the test set was 94.6% with a slight variation across the classes ranging from 91.4% for class 8 to 98.4% for class 1. Figure 5.2 shows the overall classification accuracy of the MLP, its class-specific variation and the corresponding t-SNE plot. The black and red digits correspond to the correctly and incorrectly classified input images.

Single Neuron Ablations

The ablations of single neurons affected the accuracy in different ways. In general, the overall accuracy decreased, whereas the effect on single classes differed for specific ablations. Figure 5.4 shows the effects of the ablation of neuron 12 in the first hidden layer of the MLP, which resulted in the highest drop of overall accuracy of 44.5%p for a single ablated neuron. The heights of the black and red/green bars correspond to the amount of correctly and incorrectly classified digits after the ablation, respectively. However, the red colored digits do not contain the digits that were incorrectly classified by the undamaged network and only display the change of the classification performance as a result of the ablation. Green bars with a negative value correspond to the amount of correctly classified digits after the ablation, which were incorrectly classified by the undamaged network, thus representing an improvement of the classification performance. The network lost its ability to correctly classify most digits of the classes 1, 4, 7 and 9 with a drop in class-specific accuracy of more than 80%p. The effects on the classes 6 and 8 are less severe with a drop in class-specific accuracy of around 30%p, while the effect on all other classes is smaller than 10%p. The t-SNE plot suggests that this neuron represents certain features in the data that are shared across classes, as the majority of incorrectly classified digits are located close

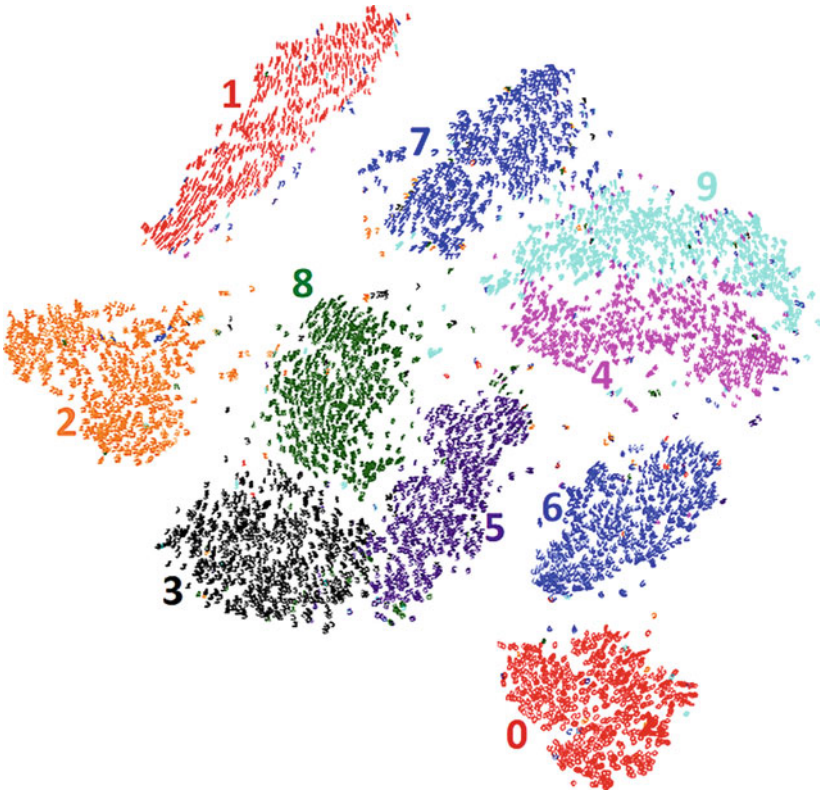


Figure 5.1 t-SNE embedding of the 10, 000 digits in the MNIST test set. Taken from [172]

to each other in the upper part of the plot. Figure ESM4 shows a similar representation for an ablation of neuron 16, where most of the incorrectly classified neurons are found in the bottom right part of the t-SNE plot.

Figure 5.3 shows the effects of the ablation of neuron 19 in the first hidden layer of the MLP, which resulted in a drop of overall accuracy of 11.6%p. In contrast to neuron 12, this neuron seems to represent features distinct to a single class, as the effect on the class-specific accuracy for class one is much stronger than for all other classes. Although this neuron is easy to interpret as it seems to represent features of a single class almost exclusively, it is not more important

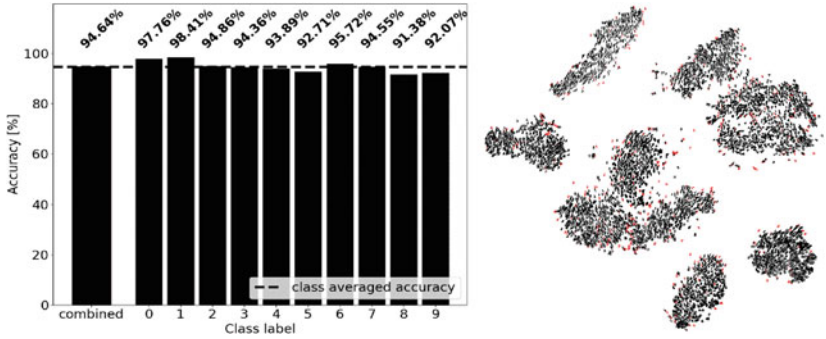


Figure 5.2 Overall accuracy, class-specific accuracy and t-SNE visualization of the trained MLP. Taken from [172]

for the classification task than other neurons, in terms of how strongly its ablation affects the overall classification performance. This result is consistent with previous investigations on the interpretability and importance of single neurons of an MLP classifier [77].

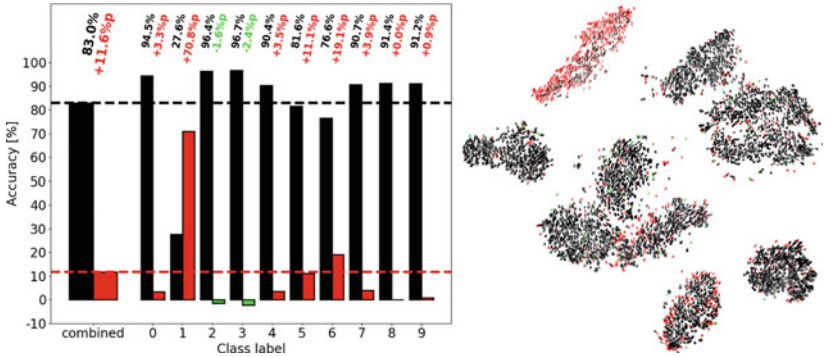


Figure 5.3 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neuron 19 in the first hidden layer. The neuron is an example for the selective representation of features distinct to a single class. Taken from [172]

Figure 5.5 shows the effects of the ablation of neuron 6 in the first hidden layer of the MLP, which resulted in a drop of overall accuracy of only 1.4%p. This neuron seems to play only a minor role in the classification task as the effect

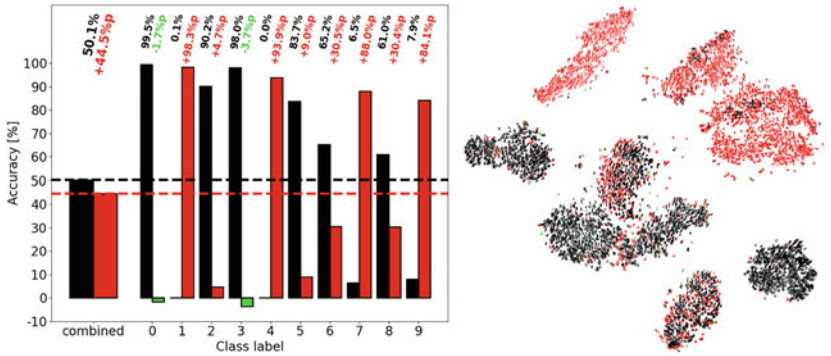


Figure 5.4 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neuron 12 in the first hidden layer. The neuron is an example for the representation of features corresponding to many different classes. Taken from [172]

of its ablation on the network's accuracy is small. In total, 4 out of the 20 neurons in the first hidden layer, neurons 6, 11, 13 and 18, showed similar effects, which makes them top candidates for pruning if one would want to optimize the size of the network (cf. Figure ESM5).

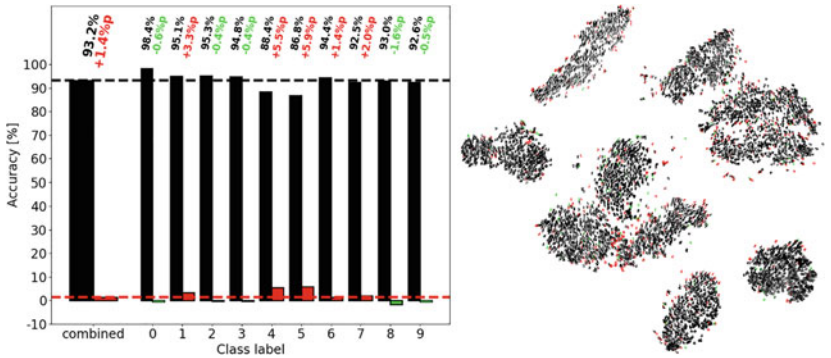


Figure 5.5 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neuron 6 in the first hidden layer. The neuron is an example for the negligible contribution to the classification task and could be pruned to optimize network size. Taken from [172]

Figure 5.6 shows the effects of the ablation of neuron 20 in the first hidden layer of the MLP, which resulted in a drop of overall accuracy of 14.6%p. This neuron seems to represent features corresponding to subtle and smoothly changing characteristics distinct to the classes 1, 6 and 9. The t-SNE visualization reveals that most of the incorrectly classified digits within a class can be found close to each other rather than evenly distributed across the whole class.

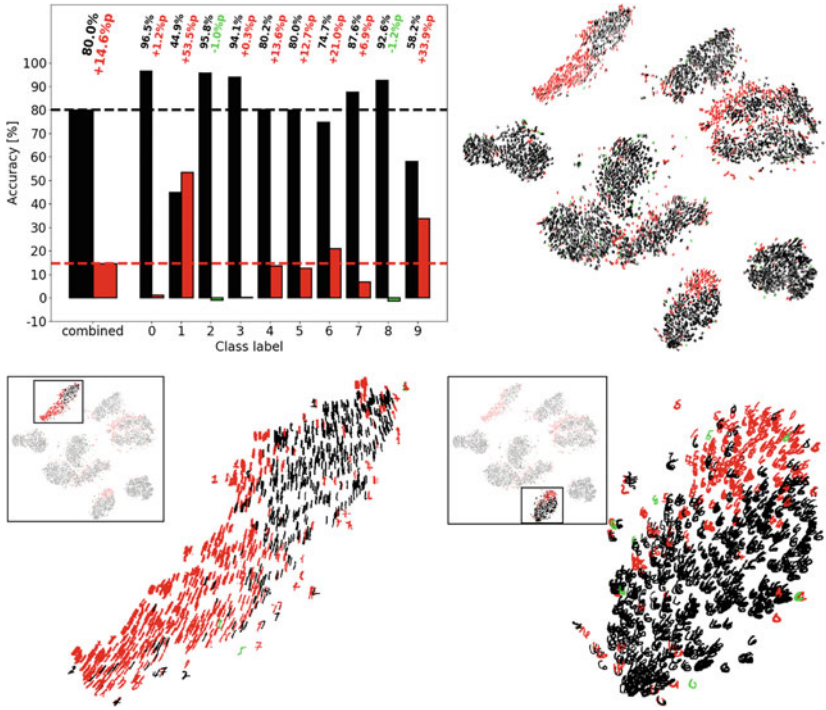


Figure 5.6 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neuron 20 in the first hidden layer. This neuron is an example for the representation of features that are distinct to a subset of digits within different classes. Taken from [172]

Figure 5.7 shows the effects of the ablation of neuron 3 in the first hidden layer of the MLP, which resulted in a drop of overall accuracy of 25.4%p but showed an increase of the class-specific accuracy of 5.7%p for class 5, which is

the strongest positive effect of all neurons in the first hidden layer. This observation is consistent across ablated single neurons, i.e., the damaged network would correctly classify some digits that were incorrectly classified by the undamaged network. These observations hint to a trade-off made while fitting the weights of the network via backpropagation, in which the recognition of a small number of digits is sacrificed for a much larger number of other digits. However, this raises the question whether the classification performance of a network can be increased beyond its trained capabilities by selectively ablating single connections to achieve the desired increase in accuracy without suffering from the negative effects.

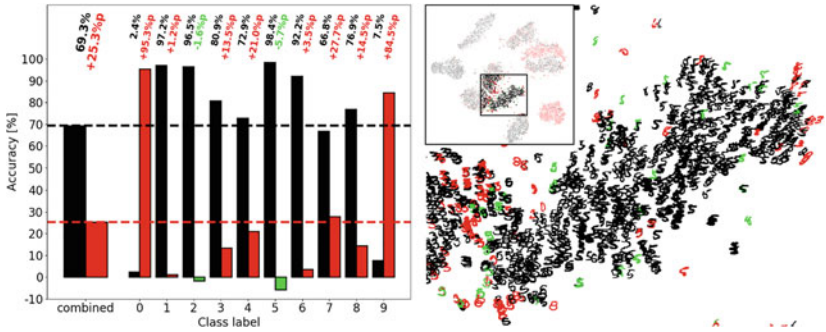


Figure 5.7 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neuron 3 in the first hidden layer. This neuron shows the strongest positive effect of an ablation, i.e., the increase of the class-specific accuracy of class 5. Taken from [172]

Following the observations of the ablations, the question arises whether the importance of neurons, i.e., the impact of their ablation on the network's performance, can be determined by a characteristic of these single neurons, which positively correlates with the change in the overall accuracy after their ablation. Such a characteristic would allow to describe the importance of single neurons for the classification task without the necessity to perform a functional test, i.e., run inference of the network for all $10,000$ images of the test set. It turns out that the degree to which the distribution of the incoming weights of a particular neuron after training differs from the randomly initialized normal distribution of weights before training is a good indication of the neuron's importance for the classification task. This difference is quantified by the p-value of the Mann-Whitney U test [177], a non-parametric statistical test, which determines whether

two independent observations were sampled from the same distribution. The p-value indicates the likelihood of both distributions to be the same ($p = 1$) or to be different from each other ($p \to 0$). Figure 5.8 shows a comparison of the weight distributions of the single neurons in the first hidden layer before and after training. Each distribution is visualized as a normalized 28×28 matrix, the same dimensions as the input images, with red and blue entries indicating high positive and negative weight values, respectively. The p-values indicate the difference of the distributions on the right-hand side compared to the left-hand side. Note that the distributions of neuron 6, 11, 13, and 18 did not change significantly during training (cf. Figure ESM5).

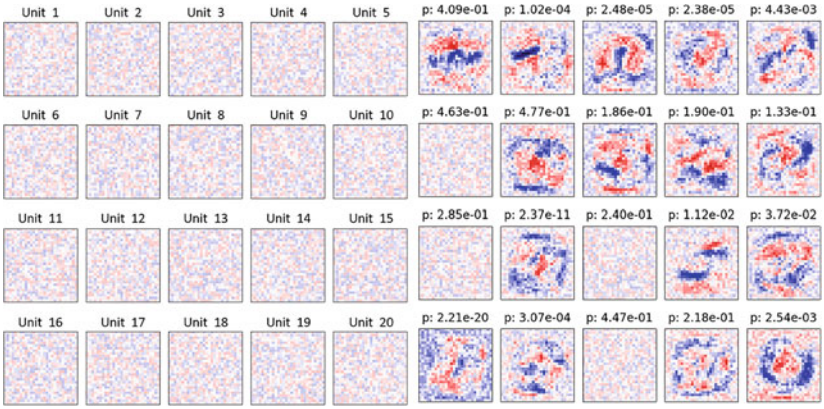


Figure 5.8 Comparison of the distributions of the incoming weights for the 20 single neurons in the first hidden layer before training (left) and after training (right). Taken from [172]

Figure 5.9 shows the Pearson [178] and Spearman [179] correlation of the Mann-Whitney U's p-value and the drop in accuracy after ablation. The left-hand side shows 20 samples corresponding to the 20 neurons in the first hidden layer of the network from which the previous results were generated. In order to verify that the observed correlation is not a result of the random initialization of the network, 20 more networks with different initializations were trained and the correlation coefficients for all 400 neurons within the first hidden layers of the 20 networks were calculated (cf. Figure 5.9, right-hand side). The

results suggest that, in general, the more a single neuron's distribution of incoming weights changes during training, the more important this neuron is for the overall classification performance.

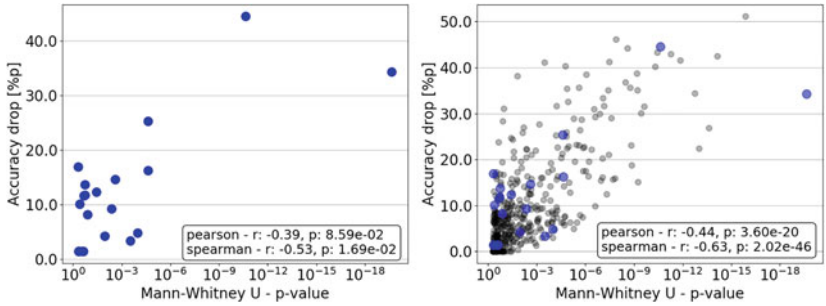


Figure 5.9 Correlation of the Mann-Whitney U's p-value with the drop in accuracy after ablation of a single neuron in the first hidden layer. Taken from [172]

Figure 5.10 shows a kernel density estimated distribution of the calculated Pearson and Spearman correlations from all 20 networks and, except for two Pearson coefficients, supports the average trend shown in Figure 5.9, right-hand side. This observation may prove useful for pruning neural networks. Neurons may be pruned based on the distributions of their incoming weights, thus, reducing the computational cost of repeatedly testing a pruned network on a large dataset.

The negative effects of ablations raise the question whether the representation of some classes within the networks is more specific to some neurons than for other classes. Aiming to answer this question, it was tested if the drop of the class-specific accuracy after an ablation is similar for all neurons within a network or if it shows a strong deviation. A high deviation would mean that some neurons within the network strongly represent a particular class while other neurons do not. This would suggest that this class is represented somewhat localized in the network rather than evenly distributed across all neurons. Therefore, for each of the 20 networks, the class-specific drops in accuracy for all 20 single neuron ablations in the first hidden layer were computed and the standard deviation was calculated. Furthermore, the mean of this class-specific accuracy deviation averaged across all 20 networks was computed in order to compare the deviations of the single networks to the population mean.

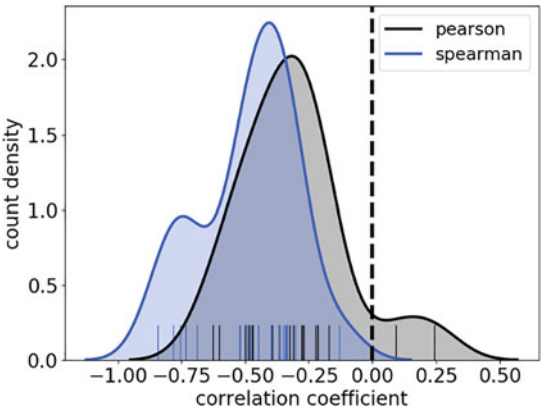


Figure 5.10 Distribution of the calculated Pearson and Spearman correlation coefficient for the 20 networks. Taken from [172]

Figure 5.11 shows the population averaged accuracy deviation and four examples of a single network accuracy deviation. The black line corresponding to the population averaged accuracy deviation shows that some classes are represented more selectively than other classes. For instance, the classes 1 and 4 have a much higher deviation than class 2, suggesting that, in general, class 2 is much more evenly represented across the first hidden layer than the classes 1 and 4. However, this trend is not universal for all 20 networks indicated by the accuracy deviations of the networks. The fact that the blue lines cross the population average suggests that, despite the general trend, the selectivity of the representation of the 10 classes is rather unique to each network. This means that some networks develop a more selective representation for some classes than others.

Pairwise Neuron Ablations

In addition to single neuron ablations, pairwise neuron ablations in the first hidden layer of the MLP were performed to investigate the feature representations for redundancies, i.e., whether the effects of pairwise neuron ablations are stronger than the sum of the corresponding single neuron ablations. In this case, the network retains its capability to correctly classify some specific classes after a single neuron ablation as another neuron still represents the corresponding features sufficiently well. However, the pairwise neuron ablation of both neurons causes the network to incorrectly classify those classes that were correctly classified in the case of the single neuron ablations, as there are no more neurons left that redundantly represent the necessary features.

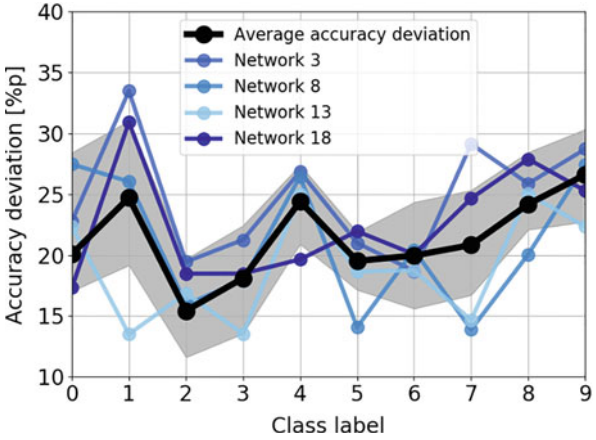


Figure 5.11 Class-specific averaged deviation across the 20 networks of the accuracy drops after ablations. Taken from [172]

Figure 5.12 shows the effects of the pairwise neuron ablation of neurons 4 and 16 in the first hidden layer of the MLP, causing the strongest observed effect to exceed the sum of the corresponding single neuron ablations. The height of the black, red/green and blue bars correspond to the number of digits correctly classified after the pairwise neuron ablation, the number of digits incorrectly classified after either corresponding single neuron ablation, and the number of digits incorrectly classified only after the pairwise neuron ablation, respectively. The digits in the t-SNE plot are colored accordingly. As a direct comparison to the single neuron ablations of neuron 12 (cf. Figure 5.4) and 19 (cf. Figure 5.3), Figure ESM6 shows the pairwise neuron ablation of neurons 12 and 19. The pairwise neuron ablation has a strong effect on class 6, for which more than 50% of the digits are incorrectly classified as a result of the pairwise neuron ablation but were correctly classified after either corresponding single neuron ablation. The t-SNE visualization shows that the digits corresponding to that redundant representation are more or less evenly distributed across the class. Note that the positive effect on class 3 is stronger than for either single neuron ablation, however this effect does not exceed the summed effects of both single neuron ablations. Contrary to the neurons 12 and 19, the t-SNE visualization for the pairwise neuron ablation of the neurons 5 and 10 suggests that the redundantly represented features correspond to the local structure of the data (cf. Figure 5.13). The blue colored digits within class 1 and 3 are clustered together, rather than being evenly distributed across

the whole class. Interestingly, the positive effect of the ablation can exceed the effects of the corresponding single neuron ablations. Even though the pairwise neuron ablation shows strong class-specific negative effects, the positive effect on class 5, improving the amount of correctly classified objects by $5.16\%p$, exceeds the summed effects of the corresponding single neuron ablations of $3.14\%p$ for neuron 5 and $0.45\%p$ for neuron 10 (cf. Figure ESM7).

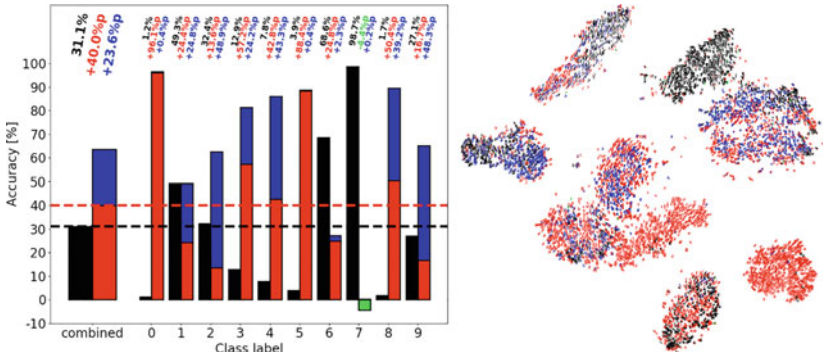


Figure 5.12 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neurons 4 and 16 in the first hidden layer. The pairwise ablation of these units had the strongest effect exceeding the summed effects of the corresponding single neuron ablations. Taken from [172]

5.1.1.4 Summary and Contribution of the Results to the Research Questions

The results of the first study address the first research question and demonstrate how ablations can be used to determine the importance of single neurons for a network's learned task. To this end, the effects of single and pairwise neuron ablations on the classification performance of a shallow MLP trained on the MNIST dataset were investigated. As expected, removing individual neurons affected the overall classification performance of the network negatively, implying that the removed neuron represents important features for the classification task. However, in some cases the class-specific performance for some classes increased despite the overall impairing effect. Considering the small gain in performance for a specific class compared to the much larger loss in performance for other

classes, the result suggests that a trade-off for a neuron's representation of features in favor of more general features relevant to a lot of classes rather than relevant to specific, individual classes emerges during training. A direct answer to the first research question is given by the finding of neurons falling into one of three distinct categories with respect to their importance for the learned task. Specifically, some neurons are universally important for the learned classification task and represent features distinct to many of the classes in the test set while other neurons are only selectively important for specific classes representing features only relevant for these classes. Additionally, a small number of neurons was shown to be not important for the classification task at all as their ablation did not impact the network's performance at all.

Furthermore, besides ablations as a method to be used to answer the first research question, the results of the study uncovered a characteristic of neurons following the ablation study based on which the first research question can be answered as well. Specifically, the distribution of a neuron's incoming weights indicates its importance for the classification task. The more a weight distribution has changed during training, the more important this neuron is for the classification task. This result may prove useful for pruning experiments, in which unimportant neurons are removed from the network without impairing its trained capabilities, as the importance of a neuron can be estimated with significantly reduced computational cost as compared to full functional tests of the network, as is required after ablations. The corresponding t-SNE visualizations of the ablation effects revealed that the features represented by single neurons mostly correspond to the global and local structure inherent to the data set. This suggests that information inherent to the stimuli, with which the network is trained, is mapped and locally represented in specific areas of the network, i.e., by a subset of neurons.

Further corroborating the answer to the first research question, pairwise ablations have shown redundancy in the representation of individual neurons. Specifically, pairwise neuron ablations revealed that the network exhibited robustness against structural alterations caused by ablations, as some features are represented redundantly in different single neurons. The pairwise ablations showed effects on the classification performance that exceed the combined effects of the corresponding single neuron ablations. Remarkably enough, this observation is true for the negative as well as for the positive effects on the performance. Curiously, the redundant representation of important features of external stimuli is one of the most important features exhibited by the mammalian brain. The resulting robustness to nerve cell damage is vital for its longevity, however, it is

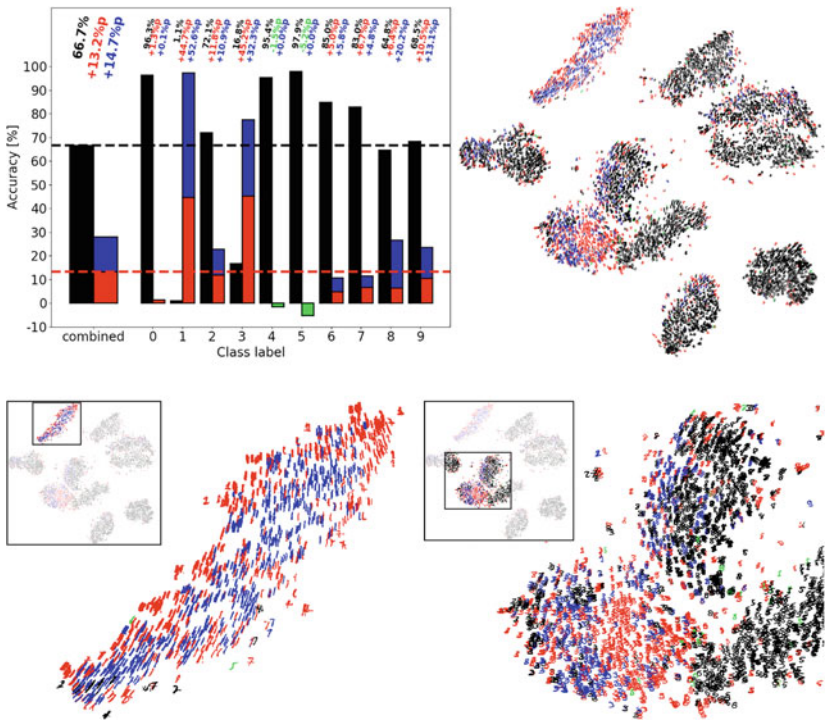


Figure 5.13 Overall accuracy, class-specific accuracy and t-SNE visualization of the damaged MLP after the ablation of neurons 5 and 10 in the first hidden layer. Note that the positive effect on class 5 is stronger after the pairwise neuron ablation than the summed effects after the corresponding single neuron ablations (cf. Figure A6). Taken from [172]

not clear why artificial neural network would develop this feature during training and how they benefit from it.

In summary, the results of the study answer the first research question and pave the way towards the second research question, which will be addressed in the following studies. Furthermore, some of the results obtained in this study will be solidified in the following studies to investigate their universality. For example, whether the positive effects of ablations are specific to the small network that was investigated, which wasn't heavily tuned to achieve the best possible results

in the dataset, or whether the phenomenon can be observed in larger, state-of-the-art networks, too, will be investigated in the second study. Furthermore, the specificity of individual neurons regarding the features they represent and their selective importance for individual classes and the specificity to the small network that was investigated or the transferability to larger state-of-the-art networks with a much larger number of neurons will be investigated in the second study. Finally, whether these selective neurons can be grouped into functional neuron clusters, i.e., a group of neurons that act jointly, and if so, how these clusters differ in size and how those differences can be related to the requirements for individual classes to be sufficiently represented within the network will be investigated in the third study.

5.1.2 Research Study 2: Network Ablations in a Deep Neural Network

The second study [172, 173] follows up on some of the results of the first study and extends its established methodology towards a large state-of-the-art computer vision neural network, the VGG-19. It corroborates the answers given to the first research question and addresses the aspects that remained unanswered after the results of the first study. To this end, the pre-trained network was subject to a systematic ablation study, in which different proportions of neurons of the network were ablated. Additionally, the extent to which the damages of ablations could be repaired by subsequent recovery training was investigated to shed light on a possible irreversibility of the damage caused by ablations. Contrary to the shallow MLP of the first study, the VGG-19 allows for an investigation of structure and organization along the depth of the network. Thus, the study addresses the second research question and demonstrates such structure and organization as different layers are shown to be important to different degrees for the classification task.

5.1.2.1 Key Contributions of the Study

The first key finding of the study corroborates the findings of the first study and addresses the first research question showing that the ablation of neurons influences the classification performance of the network negatively and positively and is a suitable approach to determine the importance of the neurons for the learned task. It further extends the previous findings as it demonstrates that the clear distinction of the neurons into different groups vanishes for a large network such as the VGG-19.

The second key finding of the study further details the answers given to the first research question and shows that that the harmful effects of ablations can be repaired by subsequent recovery training almost completely. It only takes a single epoch of recovery training to mitigate most of damage caused by ablations, even in cases of severe structural damage (up to 80% of ablated filters within a single convolutional layer) and a few more epochs to mitigate the last few percent points until the original performance of the undamaged network is recovered almost entirely. This result suggests that the important role that neurons being removed from the network have, considering the strong impact on network performance upon removal, can be taken over by other neurons via retraining.

The third key finding of the study addresses the second research question showing that the effects of ablations show a strong variance across the different layers of the network and revealed two layers that are much more important than the other layers suggesting that some layers are universally more important for the classification task than other layers. Considering that this observation varies across specific classes, the result hints towards class-specific structure and organization of the learned representations along the depth of the network.

5.1.2.2 Methods and Experimental Design

To address the second research question, a large and complex enough network was chosen that allows to investigate its learned representations with respect to its architectural features, like its depth. To this end, the VGG-19 with batch normalization was chosen as the network of interest, which was pre-trained on the ImageNet dataset as a representative of today's state-of-the-art CNNs for object recognition tasks. The VGG-19 has 19 layers with learnable weights, 16 convolutional and 3 fully connected layers. The ImageNet dataset used for this study consists of 1,000 categories with a total of 1.2 million images in the training set and 50 images per category in the validation set. Ablations of groups of similar filters with increasing proportions (1%, 5%, 10% and 25%) relative to the total number of filters in each of the convolutional layers of the network were performed. Due to the increasing sizes of the different convolutional layers, the same proportion may correspond to a different total number of ablated filters. The similarity between filters within a group was calculated based on the absolute Euclidean distance of the normalized filter weights. Similar to ablations of single units in the MLP in the first study, ablations were performed by manually setting the weights and biases of all incoming connections of a filter to 0, effectively eliminating any activation of that filter. The effect of ablations was evaluated by

testing the classification performance of the network on the validation dataset using the top-1 and top-5 accuracy.

Following the observations made in the ablation study, the capability of the network to recover its original classification performance was investigated by subsequent recovery training of the damaged network. For this purpose, ablations in the two most important layers for the classification task and retrained the damaged network were performed. All the weights in the layers above the one in which the ablation was performed were frozen, forcing the network to adapt to the change of information flow through the deeper layers of the network. First, it was investigated whether some filters are more important than others for recovering the classification performance. To this end, groups of filters with a proportion of 25% of a layer's total number of filters were ablated in multiple instances of the network. After the ablation, the network was retrained with the training set for 5 epochs during which the top-5 accuracy was computed. Second, the impact of the number of ablated filters within a layer on the network's recovery capabilities was investigated. For this purpose, ablations of 25% of the filters of one layer followed by recovery training were iteratively performed damaging the network further with each iteration. For each iteration, the filters to be ablated were chosen randomly and the recovery training was stopped after a minimum of 5 epochs when the top-5 accuracy did not improve by 0.05% over the course of 2 epochs. Note that the choice of ablated filters was performed as a selection with replacement, i.e., two consecutive ablations of 25% do not necessarily result in a total ablation of 50%. This allows to perform more than 4 iterations of ablation and subsequent recovery training, slowly and gradually decreasing the number of remaining filters in the damaged layer.

5.1.2.3 Results

The results of this study are separated into two sub-chapters. First, the results of network ablations of groups of filters is presented investigating the effects of ablations in the different layers of the VGG-19. Subsequently, the network's ability to recover from the inflicted damages of network ablations via subsequent recovery training is investigated.

Filter Ablations

Similar to the importance of single units in the MLP, some layers are more important for the classification task than other layers. Figure 5.15 top and bottom show the drop in top-1 and top-5 accuracy, respectively, for the ablation of 10% (left side) and 25% (right side) in all convolutional layers of the network.

The black curve shows the accuracy drop in each layer, averaged over all ablations performed in this layer. The number of ablations is equal to the number of filters in each particular layer, since each filter was chosen once as a reference for the choice of the 10% and 25% ablations based on filter similarity (cf. section "Methods and Experimental Design"). The red and green curves and shaded areas correspond to the lower and upper standard deviation of the average accuracy drop. Layer 33 and 46 showed a significantly higher drop in the top-1 and top-5 accuracy compared to other layers. This effect is more distinct for the smaller number of ablated filters (10%) and becomes less pronounced for the larger amount (25%). Concurrently, the effect of a larger number of ablated filters has a stronger impact on some layers than on others. For instance, layers 7, 17, and 20 show a significantly stronger drop in both, the top-1 and top-5 accuracy for 25% of ablated filters compared to 10% of ablated filters, while layer 40 is almost not affected at all. Additionally, the fact that some layers, e.g., layer 40, are largely unaffected by the increase of the proportion of ablated filters from 10% to 25% suggests that features represented in this layer may be redundantly represented in other layers or in other filters in the same layer, rendering ablations mostly harmless for the overall performance. Consistent with the observations of positive effects of ablations in the first study, the ablation of some filters in some layers of the VGG-19 showed an increase in top-1 accuracy indicated by the crossing of the zero-line of the red shaded area in Figure 5.15.

Similar to the first study, it was checked whether the importance of the layers for the overall classification performance shows class-specific variations. It was found that, despite the general class-average trend (c.f. Figure 5.14, black line), some layers are much more important for specific classes than for others. Figure 5.14 shows the class-specific drop in top-5 accuracy averaged over all ablations for 5 example classes in addition to the average drop in top-5 accuracy as Figure 5.15, bottom. In the case of the 10% ablations, class 50 shows a much higher drop in accuracy relative to the other classes after ablations in layer 7 and 20 and at the same time a lower drop in accuracy relative to the other classes after ablations in layer 33. Additionally, the drop in accuracy after the 25% ablation in layer 14 and 17 is much stronger and much weaker, respectively, than the average drop. This observation suggests that layers exhibit a certain degree of class-selectivity and therefore have different relative importance for the overall performance depending on the class. Based on this finding, it was further investigated how this selectivity is distributed across classes, i.e., to what extent a layer represents specific classes more than others.

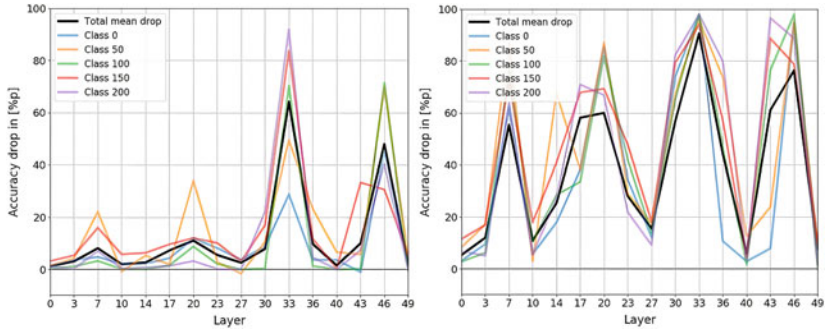


Figure 5.14 Examples for the variation of the class-specific effect of ablations of the top-5 accuracy for different amounts of ablated filters (left: 10%, right: 25%) in all convolutional layers. Taken from [172]

Figure 5.16 shows two extreme examples for the class-specific drop in top-5 accuracy after ablations of 10% (left) and 25% (right) in layer 46 (top) and layer 49 (bottom), respectively. Consistent with the observations of the first study, ablations had a negative effect on the classification performance for most classes. For some classes, however, the class-specific top-5 accuracy improved after the ablations. This effect was stronger for smaller ablations and in layers with a comparably small impact on the overall performance, such as layer 49 (c.f. Figure 5.16, bottom right).

Recovery Training

Subsequent to the ablations, it was investigated whether the negative effects on the classification performance could be recovered and if so, to what extent (cf. section "Methods and Experimental Design"). Figure 5.18 shows the top-5 accuracy after the ablation of 25% of filters in layer 33 and 46 and 5 epochs of subsequent recovery training in 5 instances of the VGG-19. The results show that the network recovered most of the lost classification ability after a single recovery epoch with a margin of less than $1\%p$ compared to its original top-5 accuracy (c.f. Figure 5.18, left side) with only marginal improvement for the epochs after the first one (c.f. Figure 5.18, right side). In general, the original accuracy was never exceeded after the recovery training. However, due to computational cost, recovery training was stopped after 6 epochs, even though the

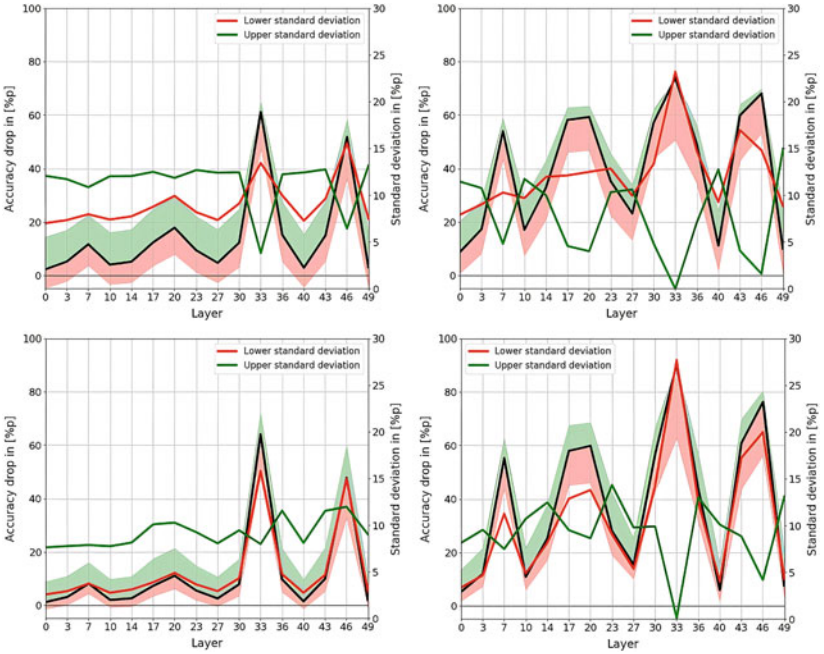


Figure 5.15 Effect on the top-1 accuracy (top) and top-5 accuracy (bottom) of ablations of different amounts (left: 10% of layers, right: 25% of layer filters) in all convolutional layers. Taken from [172]

accuracy was still increasing. In the case of layer 46, the extent of the drop in accuracy did not seem to impact the recovery process significantly. Although the top-5 accuracy after the ablations showed a strong variation of up to 30%p, the network was able to recover the damages regardless of the severity of the initial damage.

Figure 5.17 shows the top-5 accuracy for iteratively performed ablations of 25% of filters in layer 33 and 46 and subsequent recovery training in a single instance of VGG-19. Note that the filters to be ablated for each iteration were selected with replacement, resulting in a slow and gradual increase of the damage inflicted on the network with each iteration. After the last iteration, $\sim$ 80% of the filters were ablated in either layer. Remarkably, the network was able to

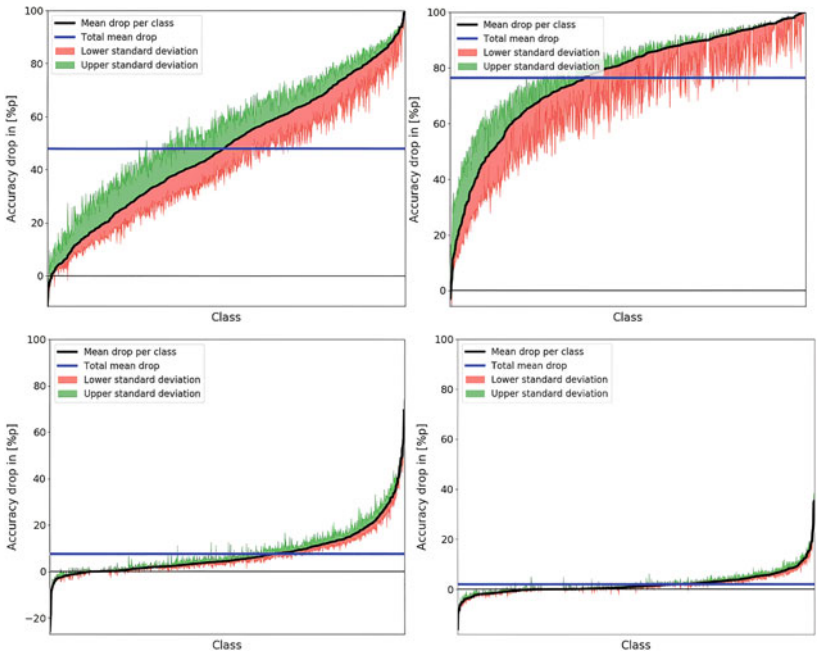


Figure 5.16 Top-5 class specific accuracy drop after ablation of 10% (left) and 25% (right) of filters in layer 46 (top) and 49 (bottom). Taken from [172]

recover almost completely from the damage caused by the ablations, despite the increasing number of ablated filters. Similar to the first recovery experiment, the performance rapidly increased during the first training epoch for each iteration and only improved marginally for the following epochs. The difference between the recovered top-5 accuracy and the original top-5 accuracy showed a slight increase with the number of iterations. The results suggest that with only $\sim 20\%$ of filters left in either one of the two most important layers, the network is still able to recover most of the damage and to represent the majority of the necessary information in the remaining network.

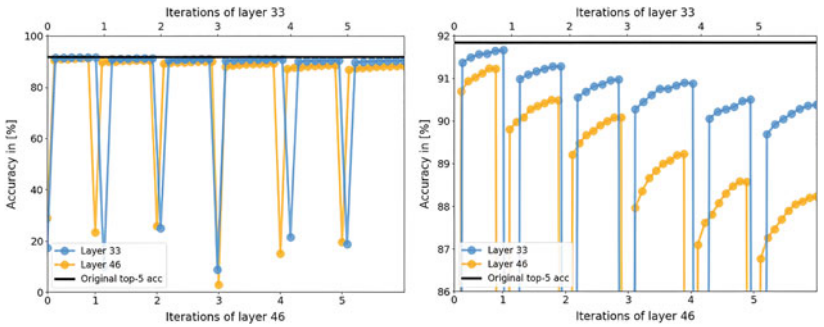


Figure 5.17 Iterative ablation of 25% of filters in layers 33 (Blue) and 46 (Orange) and subsequent recovery process of the top-5 accuracy of the VGG-19. Note that the filters ablated in each iteration were selected with replacement. Taken from [172]

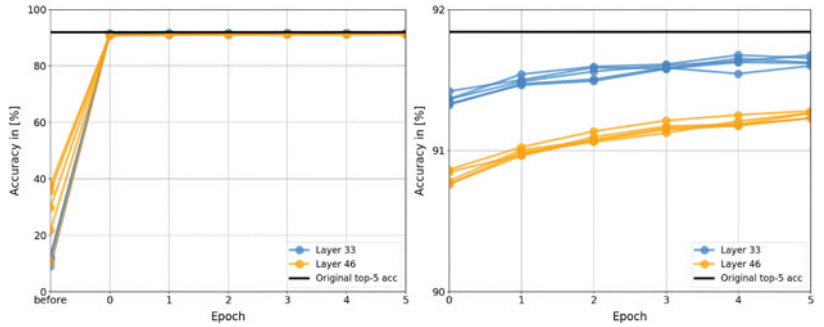


Figure 5.18 Recovery process of the top-5 accuracy if 5 instances of the VGG-19 after ablations of 25% of filters in layers 33 (Blue) and 46 (Orange). Taken from [172]

5.1.2.4 Summary and Contribution of the Results to the Research Questions

The results of the study corroborated the answers to the first research question given in the previous study by transferring the principle of ablations to the VGG-19 and gave a first answer to the second research question. Specifically addressing the first research question, the results supported, that ablations had negative as well as positive effects on the classification performance. In general, the higher the number of ablated filters was, the stronger the effect. Furthermore, the clear

distinction of neurons into three categories was not observed as clearly as in the first study and is therefore likely to be a result of the small and shallow network architecture in the first study. Addressing the second research question, the learned representation of individual classes was shown to be distributed across the layers of the network. Specifically, the different layers were shown not to be equally important for the classification performance. Despite a general trend, the importance of layers showed some class-specific variation. This indicates that the representation of features distinct to specific classes is somewhat localized in the network. More precisely, the effect of ablations turned out to be significantly stronger for two deeper layers (33 and 46) than for all other layers. At first, this may seem surprising as the upper layers are expected to be the most important layers as they are supposed to represent more general features common to many classes, whereas lower layers are supposed to represent more class-specific features [115]. However, a possible explanation may be that the features represented in these two layers are not redundantly represented in other layers, while the features in the upper layers are.

Furthermore, the results of the study investigated if and to what extent the inflicted ablation damage to the network could be recovered by subsequent recovery training. Irrespective of the location of the ablation and the severity of the inflicted damage, i.e., the magnitude of the drop in accuracy, most of the original classification performance could be recovered after a single epoch of training. Subsequent epochs only improved the performance marginally. However, the larger the number of ablated filters becomes, the harder it is for the network to recover its original performance. Remarkably though, even after an ablation of $\sim 80\%$ of filters in the most important layers (33 and 46), the performance could be recovered up to a difference of less than $4\%p$. This finding further details the answer to the first research qeustion and demonstrates that the role of individual neurons for the classification task neurons ablated from the network can be taken over by other neurons by subsequent retraining.

Building on the results of this study and in combination with the results of the first study, a key question arises for the third study in the domain of computer vision. So far, the results presented were extracted from networks trained on a single individual dataset. It remains unclear how specific those results are to the datasets and how they generalize across different datasets. The third study picks up that question and investigates a medium sized network trained on three different datasets with different complexities for a computer vision classification task.

5.1.3 Research Study 3: Functional Neuron Populations in Custom-made CNNs

The third study [174, 180] further corroborates the results addressing the first research question and investigates their universality. Specifically, this study investigates the effects of ablations on a custom-tailored CNN trained on three different datasets in order to validate that the previously observed results are not specific to a single dataset but can be observed across datasets. Furthermore, the study extends the methodical investigation of how to determine the importance of individual neurons beyond the use of ablations by other commonly used measures based on the neuron's activations. Additionally, the study addresses the second research question and investigates the evolvement of the learned representation along the hierarchically structured layers of the networks and how these representations are affected by partial ablations in different layers of the network.

5.1.3.1 Key Contributions of the Study

The first key finding expands on the answers of the first research question demonstrating that those different approaches to quantify the importance of individual neurons do not convincingly yield the same results. Specifically, attributing a neuron's importance to its magnitude of activation or its selectivity of activation yields different results than the attribution via the impact on the network's performance upon the neuron's ablation. Thus, the importance of a single neuron for the classification task cannot solely be attributed to the magnitude or the selectivity of its activity and the right method to determine a neuron's importance consistently remains unclear. Arguably, ablations are the most intuitive way to determine a neuron's importance as their removal and the corresponding impact on the network's performance are evidently describing the importance of their existence in the network. Thus, the results cast doubt on the widely accepted notion that a neuron's magnitude of activation determines it's importance for the learned task.

The second key finding addresses the second research question and demonstrates that the learned representations of the network evolve spatially along its hierarchically structured layers so that the distinction the network learned becomes most prominent in its second to last layer, i.e., the layer before the output layer. It describes the structure and organization of the learned representation with respect to the architectural features of the network and give a direct answer to the second research question. This finding is consistent with previously reported results and supports the key idea of deep learning, i.e., to represent data in hierarchically structured layers. Furthermore, these evolved representations are

impacted by ablations with respect to how distinct the individual classes are separated. Previously successfully separated classes in the activation-space show an overlap in the representation as a result of ablations, which leads to false classifications of inputs. Thus, the finding further answers the second research question and demonstrates how a distortion of the evolved representation as a result of ablations impacts the network's performance.

The third key finding addresses the first and second research question equally showing that the learned representation exhibits distinct patterns in the activation space of the network for different classes, a finding that has been observed consistently in the mammalian brain [56–58], where a distinct set of units shows a high activity in response to specific inputs and a low activity for other inputs. It describes how individual neurons form functional groups that act jointly in response to specific input stimuli giving some degree of organization of to the network's learned representation. These organized groups of neurons can be quantitatively investigated with respect to the number of neurons within those groups, which directly represents the amount of capacity used by a network to represent different classes. The size of these groups, i.e., the number of neurons within them, varies strongly across classes. Specifically, some classes only require a small number of neurons to be represented reliably, estimated by either of the three approaches to determine a neuron's importance, while other classes require a much larger number suggesting that the organization of the learned representation depends on how challenging it is to represent the individual classes.

5.1.3.2 Methods and Experimental Design

In order to solidify the answers to the first research question and further address the second research question, i.e., specifically to investigate the structure and organization of learned representations across different networks trained to perform image recognition tasks on different datasets, in this study, a custom-tailored network was investigated in two ways. First, in similar fashion to the first two studies, network ablations were conducted to estimate the importance of individual neurons in the different layers of the network, which supports the previously given answers to the first research question. Second, two ways of embedding the network activations were chosen to a) investigate the evolvement of the learned representation along the depth of the network and b) investigate for functional neuron populations, i.e., groups of neurons that activate jointly in response to specific inputs. Both ways provide insights in how the learned representations of the network are structured and organized and directly yield answers to the second research question.

Network Training and Ablations

The architecture of the network investigated in this study was custom-tailored to serve two purposes. The first purpose of the network is to be able to learn representations of three different datasets that are sufficiently accurate and yield close to state-of-the-art performances with respect to the respective classification tasks. The second purpose of the network is to be spatially complex enough to allow for investigations of structure and organization of the learned representation with respect to its architectural features, i.e., it's layer structure. Serving these purposes, the network functions as a generic representative for many commonly applied architecture choices in computer vision network. Specifically, convolutional and fully connected layers are the fundamental building blocks of most computer-vision models, making them the most interesting subjects of investigation. The number of layers and layer sizes were chosen with respect to the complexity of the datasets, so that an accuracy of at least 90% could be achieved, while still indicating the varying levels of difficulty between the three datasets. The MNIST-like datasets were chosen, because ten classes can still easily and comprehensively be captured and interpreted through visual inspection.

With these purposes and goals in mind, the network was built to contain of three convolutional layers, *"conv1"*, *"conv2"* and *"conv3"*, two fully connected layers, *"fc1"* and *"fc2",* and the output layer *"out"*. All convolutional layers have 64 two-dimensional kernels of size 5×5 with a stride of 1 and zero-padding of 2 and are followed by max-pooling layers with 2×2 kernels and stride 2. The fully connected layers are comprised of 512 neurons each, while the output layer is comprised of 10 neurons corresponding to the 10 classes of the datasets. ReLU activation is chosen for all layers except the output layer, which uses log-SoftMax activation. Separate instances of the network were trained on the normalized ($\mathcal{N}(\mu = 0.5, \sigma = 0.5)$) MNIST [70], Kuzushiji-MNIST [84] and Fashion-MNIST [75] dataset for 100 epochs with a learning rate of 0.001 and momentum of 0.9, optimizing the cross-entropy loss [181] with stochastic gradient descent for the ten target classes. The $60,000$ training images per dataset were processed with a batch size of 64. Testing was conducted using 9984 out of $10,000$ test images due to a test batch-size of 32. Henceforth, the three networks will be referred to as *"M-Net"*, *"K-Net"* and *"F-Net"*. All networks were implemented and trained with PyTorch v1.3 [182] and scored top-1 accuracies of 99.0%, 95.3% and 91.2%, on the MNIST, KMNIST and Fashion-MNIST dataset, respectively.

Ablations of single neurons in the fully connected layers were performed in the same fashion as in the first study, i.e., by manually setting their incoming weights and biases to zero, effectively preventing any flow of information through those

neurons. Concurrently, ablations in convolutional layers were performed in the same fashion as in the second study, i.e., by setting the weights and biases of all neurons of a kernel to zero, consequently ablating 5×5 neurons at once. For reasons of simplicity, ablated single units in the fully connected layers as well as ablated kernels are referred to as units throughout the remainder of this paper.

All computations were performed on a single end consumer machine containing an 8-core Ryzen 7 1800 X processor and a single NVIDIA GTX 1080 Ti GPU.

Embedding of Network Activations

Activations of each unit of the three networks in response to each image in the three test sets were stored in a matrix. Considering the number of test images, i.e., $9,984$, and the number of neurons in each network, i.e., $17,280$, the resulting activation matrices are $M_{X-Net} \in \mathbb{R}^{9984 \times 17280}$, where $X \in \{M, K, F\}$. The activation matrices were embedded using UMAP in two ways. Either dimension of the matrix was reduced, so that a point either represents the activation of the whole network or a single network layer in response to a single test image (horizontal reduction, $M \in \mathbb{R}^{9984 \times 2}$) or it represents the activation of a single unit in response to the whole test set (vertical reduction, $M \in \mathbb{R}^{2 \times 17,280}$). To calculate the embeddings, an open-source Python implementation of UMAP with default parameters was used throughout the analysis. Initial attempts for finding better values for the number of nearest neighbors or the minimum distance between data points yielded no significant visual improvement of the embeddings, nor did it substantially change observed phenomena qualitatively or quantitatively. Thus, the chosen approach turned out not to be sensitive to the particular form of embedding, which seems suitable as the goal is not to capture the ground truth behind the structure of the activation-space, but relative changes to it for reasons of comparison between different layers.

In order to make the activation embeddings after horizontal reduction of different network layers comparable to each other, embeddings were initialized by applying UMAP directly to the test set with loosened constraints ($min\_dist = 0.8$) so that the initial coordinates of the data points for each embedded layer activation were the same. Since linear shifts, scales and rotations are not accounted for by UMAP, Scipy's Procrustes transformation [183] was used to linearly scale, shift, reflect and rotate the embeddings with respect to the projection of the previous layer, which further improved the comparability between activation embeddings. The neighborhood hit (NH) in the activation embeddings was used as a quantitative measure of class separation. The NH-score is a measure for the percentage of points, for which the k nearest neighbors of a point belong to the

same class as the point itself. For the presented results, $k = 6$ was determined to yield reasonable results, which are consistent with the visual inspections. Notably, this was true for all used datasets, despite the differences in variance within single classes e.g., between MNIST and KMNIST. Aiming to investigate network activity for functional neuron populations, i.e., clusters of neurons with similar activations in response to the test images, different colors were assigned to each unit in the vertically reduced embeddings. Figure 5.19 shows the three neuron populations of *M-Net (*Figure 5.19c*, bottom right)*, *K-Net (*Figure 5.19b*, bottom left)* and *F-Net (*Figure 5.19a*, top)* with each neuron being colored according to its layer affiliation.

Alternative to coloring the neurons according to their corresponding layers in the network, they are also colored according to functional metrics characterizing a neuron's magnitude of activation or its class selectivity (neither shown in Figure 5.19). As a measure for how selectively a neuron activates for a specific class, the activation selectivity (AS), defined as

$$AS = \frac{\mu_{max} - \mu_{av.else}}{\mu_{max} + \mu_{av.else}} \in [0, 1]$$

was calculated for each neuron, where μ_{max} is the highest class-specific mean activity and $\mu_{av.else}$ is the mean activity across all other classes [77]. Higher values of AS denote a stronger tendency of a unit to only activate for a single class. In cases, in which the denominator was 0, the AS was manually set to 0. . As a measure of characterizing a neuron's importance for representing a class, the neurons were colored based on the change of network accuracy as a result of the ablation of that neuron. Similarly, the ablation effect selectivity (AES), is defined as

$$AES = \frac{\Delta_{max} - \Delta_{av.else}}{\Delta_{max} + \Delta_{av.else}}$$

where Δ_{max} is the highest class-specific change in accuracy and $\Delta_{av.else}$ is the average change in accuracy across the other classes. Since the AES can be positive or negative, the scale was separated into two, both of which were re-scaled according to their maximum positive and negative values to take values between 0 and 1. Analogously to the AS, in cases, in which the denominator was 0, the AES was set to 0.

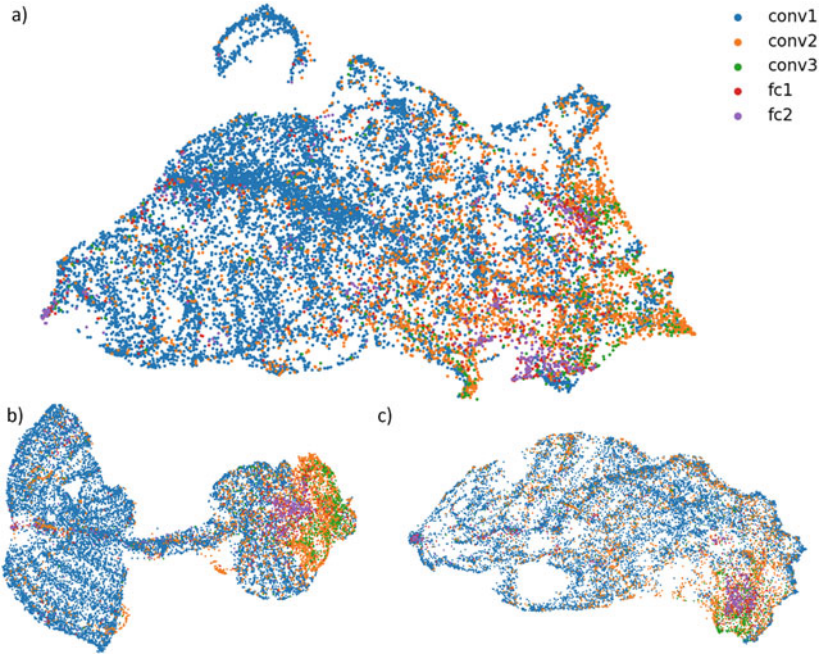


Figure 5.19 Neuron populations obtained from the vertical reduction of the activation space. a) F-Net, b) K-Net, c) M-Net. Taken from [174]

5.1.3.3 Results

The results of this study are separated into three sub-chapters. First, the results of network ablations of groups of individual neurons with varying group size is presented investigating the effects of ablations in the different layers of the network. Subsequently, the evolvement of the learned representation along the layers of the network and the effects of ablations on the representation representations in different layers is investigated. Finally, the embedded activation space of the individual neurons is investigated aiming to uncover functional neuron populations that contain individual neurons that act jointly in response to specific stimuli.

Network Ablations

Network ablations were performed in different layers to determine whether the representation of the different classes is equally distributed across the network layers or whether it shows a preference for some layers over the others. Initially performing ablations of 10%, 20%, 30%, 40% and 50% of neurons within a single layer revealed that all three networks are robust against ablations, showing only marginal changes in accuracy for smaller amounts. Thus, the following results exclusively report on the ablations of 50% of neuron within a layer. For the choice of neurons to be ablated, a random selection without replacement was conducted 100 times and the average change in accuracy for the specific classes was calculated.

Figure 5.20 shows the effects of ablations in the M-Net. Some classes are more severely affected by the ablations than other classes, suggesting that the amount of capacity used to represent the classes differs greatly. For instance, class 7 shows the largest change in accuracy while class 3 shows the smallest, indicating that the network used much more of its capacity to represent class 7 compared to class 3. This implies that the representation of class 7 is more complex than the representation of class 3. However, this is not directly explainable based on the mere separability of the classes. More precisely, although class 3 and class 7 are equally well separated in the embedded data space (cf. Figure ESM10), the amount of network capacity which is used for their accurate representations differs greatly. Another intuition is that class 3 is represented more redundantly than class 7, so that ablations have a less severe effect on class 3. Figure 5.20 further shows that the representations of single classes is distributed across the layers. For instance, the distribution of the classes 2 and 7 show the strongest localization in *conv3* compared to the other layers, which is consistent with the notion that the last convolutional layer functions as the feature extraction layer, which represents the most distinct features of the data. However, the classes 1, 8 and 9, for example, show a more equal distribution across the three convolutional layers.

The variance in capacity taken up by each class within the network raises the question whether the difficulty to predict a class correlates with the amount of capacity that is reserved by the network to represent that specific class. To answer this question, the Spearman rank-order correlation between the original prediction error and the change in accuracy for each class was calculated. A correlation coefficient of $r = 0.6$ and a p-value of $p = 0.07$ for *K-Net* and $r = 0.48$, $p = 0.16$ for *F-Net* suggests, that classes, which are more difficult to predict are also more sensitive to ablations. The corresponding bar plots can be found in Figure ESM8 and Figure ESM9. Note however, that the number of

samples of 10 is small limiting the descriptive statistical power of the test. In case of *M-Net* (cf. Figure 5.20), no significant correlation was found ($r = 0.26$, $p = 0.47$).

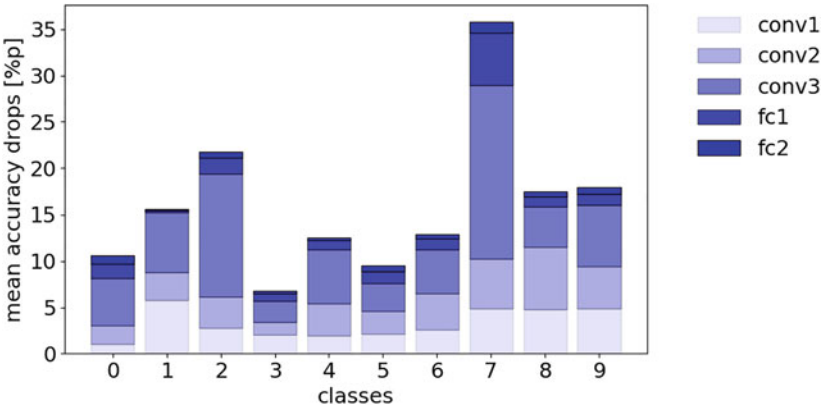


Figure 5.20 Stacked bar plot of the mean accuracy changes as a result of network ablations in M-Net. Taken from [174]

Evolvement of Representations

Aiming to answer the second research question, the study investigated how the learned representations evolve along the layers of the network to uncover a possible organization along its depth. Figure 5.21 shows the representations in the different network layers in the horizontally reduced activation-space of the undamaged *K-Net*. The scatter embedded neuron activations suggest that the separability of classes becomes more distinct further down the network as indicated by the increasing NH-score, which is consistent with previous findings [142]. In the representation in *conv2,* the NH-score is higher than the NH-score in the embedding of the original dataset, despite the higher dimensionality of the activation-space compared to the original feature space, suggesting that at least two convolutional layers are necessary to extract meaningful features to separate the classes. In general, class clusters become more dense and more distinct from *conv1* to *fc1* and are bundled together after *fc1* to *out*. For example, comparing the representations in *fc1* and *fc2*, class 3 (red) and class 9 (cyan) are mapped closer together. There are exceptions to this trend however, e.g., for class 5 (brown) and class 8 (yellow), which remain split-up even after the soft-max activation in *out*.

This shows that the representation of *out* is still able to represent more distinct classes than the number of labeled classes in the datasets.

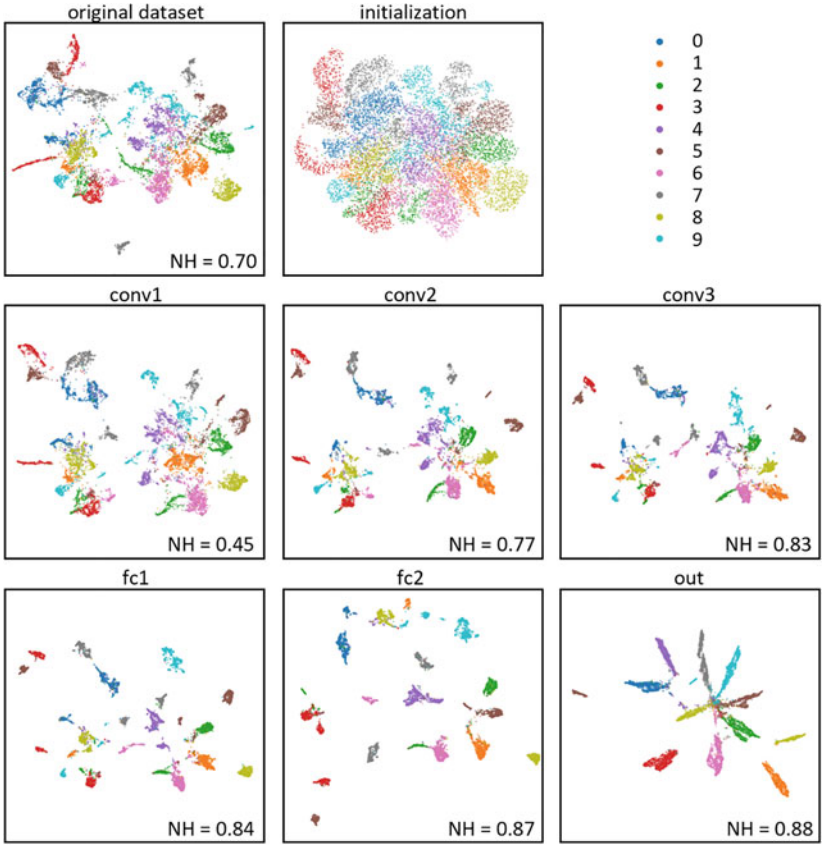


Figure 5.21 Evolvement of the learned representations along the layers of K-Net. Data points are colored according to their target class. The top middle panel shows the initialization used for all layer embeddings. Taken from [174]

Subsequently, it was investigated how the learned representations change after network ablations. Given the results of the first two studies, i.e., that ablations generally lead to worse classification performance for specific classes, an obvious hypothesis is that ablations would locally distort the activation-space so that

particularly heavily affected classes would be represented differently, e.g., more scattered, split or shifted to a different location, which leads to worse separability.

Figure 5.22 shows the layer representations of the ablated *K-Net*. Ablations were performed in $conv1$ resulting in the strongest accuracy change of $30.9\%p$ for class 1, and the weakest of $0.4\%p$ for class 2. The black dots represent all the images that are misclassified as a result of the network ablations. Ablations had no particular effect on the separability of classes in the layer representations, as indicated by the NH-scores except in the output layer, where the misclassified images are not distributed into separate clusters anymore. Most of the black points correspond to misclassified images of class 1, which showed the highest drop in accuracy of $30.9\%p$. This suggests that the ablations caused a distortion in the representations of class 1, which amplified along the network layers and led to a less distinguishable representation of class 1 in the output layer. Interestingly, the representation of class 2 was no longer unified into a single cluster, as was the case for the undamaged *K-Net*. Considering the high inter-class variance of the KMNIST dataset, this implies that ablations deprived the network of representing some kind of similarly that would aggregate the different characters of class 2. Yet, the network did not lose much of the prediction performance for this class ($0.4\%p$). These effects were consistently observed across the different datasets and the corresponding results are shown in the electronic supplementary material in Figure ESM10, Figure ESM11, Figure ESM12, and Figure ESM13 for *M-Net* and *F-Net*.

Functional Neuron Populations

Further addressing the second research question, the second part of the study aimed to identify functional neuron populations in the trained networks, i.e., groups of neurons that a) show covariant behavior in response to input stimuli or b) affect the network accuracy in a similar way. Such grouping of neurons would constitute an organized way of the learned representation with respect to specific activation patterns of neurons and the specificity of their responses to certain stimuli. Figure 5.23 shows the neuron population of *F-Net* with different color-codes. The neurons in Figure 5.23 a) and b) are colored according to their layer affiliations and their magnitude of activation in response to a single example image of class 1 and 2, respectively, where a strong/weak saturation corresponds to a strong/weak activation. The activations are normalized between values of 0 and 1 for each layer separately, due to large numerical differences of the absolute values across layers. Comparing both activation patterns with each other reveals that there are different clusters of neurons that jointly activate in response to the

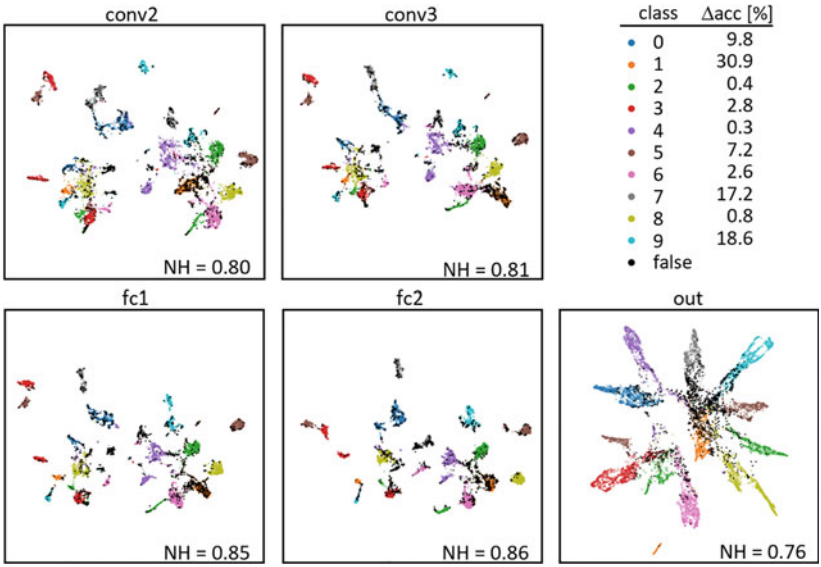


Figure 5.22 Effects of ablations in conv1 on the evolvement of the learned representations in subsequent layers in K-Net. Black points represent the misclassified images as a result of the ablations. Taken from [174]

different stimuli, indicating that the different classes are represented by different sets of neurons.

The neurons in Figure 5.23 c) and d) are colored according to their AS. Colored neurons are most selective for class 1 and 2, respectively, while grey neurons are most selective for other classes. A strong/weak saturation in color corresponds to a high/low value of selectivity. Consistent with Figure 5.23 5 a) and b), the selectivity patterns reveal different clusters of most selective neurons for the different classes. Interestingly, these clusters differ from the clusters in Figure 5.23 a) and b), suggesting that neurons, which are most selective for a specific class are not necessarily the most active in response to stimuli from that class. For both metrics, distinct clusters appear for all classes, but some regions are shared across classes. This implies that *F-Net* represents both, class specific and cross-class features in early layers. This is somewhat surprising, as early layers in CNNs are typically thought to represent general features shared by multiple classes [184]. Furthermore, the number of neurons that are most selective for a specific class

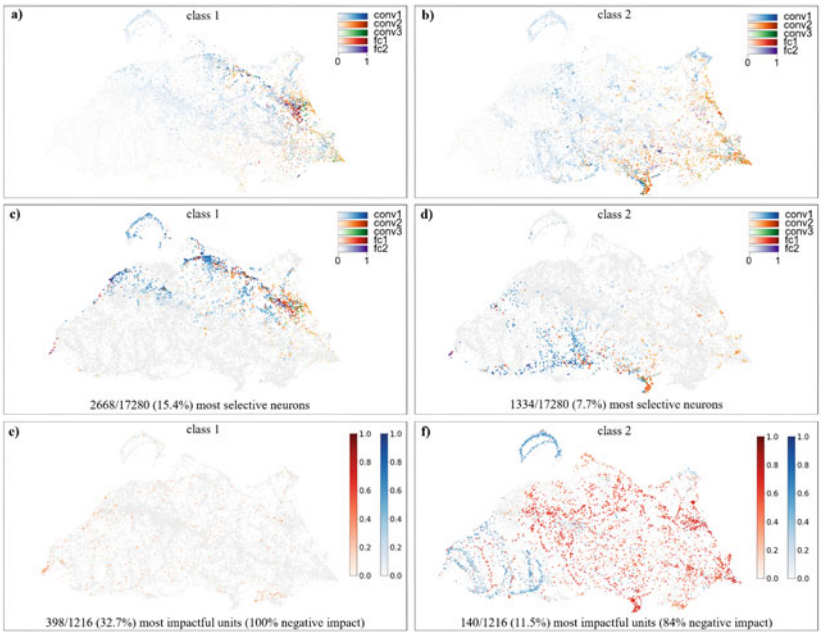


Figure 5.23 Neuron population of F-Net with different color-codes. Neurons are colored according to layer affiliation and magnitude of activation (top row), according to layer affiliation and selectivity of their activation (middle row) and according to their impact on network accuracy upon ablation (bottom row). Left column: neuron activation was measured in response to a single example of class 1 (trouser) while the impact of ablations was calculated for all images of class 1 in the test set. Right column: same as left column, but for class 2. Taken from [174]

varies greatly across the different classes. Specifically, more than twice as many neurons are most selective for class 1 compared to class 2.

The neurons in Figure 5.23 e) and f) are colored according to their AES. Neurons, which negatively/positively impact the networks accuracy upon ablation are red/blue, while the saturation corresponds to the severity of the impact. A strong/weak saturation corresponds to a high/low impact, which is scaled between 0 and 1 (cf. section "Embedding of Network Activations"). Note that for ablated units in the convolutional layers, 25 single neurons corresponding to the ablated kernel share the same AES value. Consistent with the previous results, the number of neurons that are most selective in their impact on network accuracy differs

greatly across the different classes. Specifically, almost three times as many neurons are most selectively impacting the accuracy of class 1 compared to class 2. However, all those neurons show a negative impact, while a couple of neurons show a positive impact on network accuracy of class 2 upon ablation. This finding is consistent with previously reported results in the first study and suggests that ablations may be used to fine-tune network structure to improve network performance beyond its initial training accuracies. Interestingly, similar classes, i.e., classes that are close to each other in the UMA P embedding-- of the data, did not necessarily invoke similar patterns of activation, selectivity, or ablation impact. For example, classes seem to arbitrarily share strongly activated areas or show exclusive patterns (cf. Figure 5.24 a) and b)). These effects were consistently observed across the different datasets and the corresponding results are shown in the electronic supplementary material in Figure ESM14 and Figure ESM15 for the neuron populations of *M-Net* and *K-Net*, even though for the latter, samples of the same class are largely different from each other.

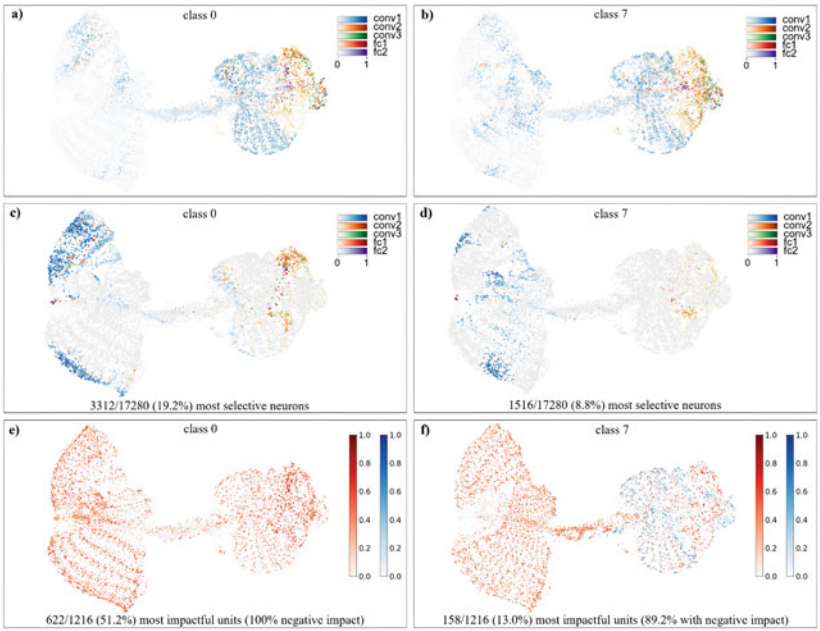


Figure 5.24 Neuron population of K-Net with coloring analogous to Figure 5.23. Taken from [174]

Assessing if classes with a high number of selective neurons are easier to separate than classes with a low number of selective neurons was done by calculating the Pearson correlation between the number of selective neurons per class and their NH-score calculated in the UMAP embeddings of the test dataset. A correlation coefficient of $r = 0.61$ and a p-value of $p = 0.06$ confirms a positive correlation, implying that classes that are well separable due to a higher number of class specific features are expectedly represented by a larger number of selective neurons in the network. This positive correlation was only found for *F-Net*, but not for *M-Net* ($r = 0.28$, $p = 0.44$) or *K-Net* ($r = 0.30$, $p = 0.40$). Additionally, *F-Net* showed a significant positive correlation between the number of units with the most class specific impact on network accuracy upon ablation and their NH-score ($r = 0.68$, $p = 0.03$). Again, such significance was not found for *M-Net* ($r = -0.31$, $p = 0.38$) nor *K-Net* ($r = 0.50$, $p = 0.14$). Note that in all cases the number of samples of 10 is small, limiting the descriptive statistical power of the test.

5.1.3.4 Summary and Contribution of the Results to the Research Questions

To solidify previous results addressing the first research question and further address the second research question, this study has taken an empirical approach to characterize the learned representations in three custom-tailored neural networks to identify structural key elements aiming to describe their role for the task of the network. The key results yield that class specific representations are not evenly distributed across the network but localized either in specific layers or groups of neurons and that these distributions vary greatly across classes. This implies that the extent of the localization of knowledge depends on class specific properties and raised two questions directly connected to the second research question, which are 1) How is the localization of a class specific representation affected by these class specific properties? 2) How does the robustness of these class specific representations against network ablations depend on these properties? Answering both questions shed light on the structure and organization of the learned representations, thus, partly answering the second research question.

To answer these questions, the evolvement of the learned representations along the layers of the unharmed network was visualized and shown to become more distinct along the depth of the network, facilitating better class separability in the network's activation-space. Subsequently, it was analyzed how the separability of these class representations is influenced through network ablations. Ablations in earlier layers only marginally affect the separability in subsequent layers but show

a strong effect in the output layer. This suggests that ablations do not selectively affect parts of the network but rather the whole network in a holistic manner as the relative positions of single units in the activation-space is mostly preserved. However, the distortion of the representation in the output layer implies that strongly class distinguishing features are still represented but more subtle features are not. This may be due to a redundant representation of such strongly class distinguishing features making their representation more robust against ablations than other features.

Further characterizing organization between individual neurons that constitute the learned representation, the finding of functional neuron populations revealed that the size of such populations differs greatly depending on their role to represent a specific class. The findings suggest that the required capacity to represent specific classes depends on the properties of that specific class. Furthermore, the lack of similarity between the functional neuron populations colored according to different metrics suggests that there is no single metric that sufficiently describes the role of single units within the whole network. This finding partly answers the first research question and casts doubt on the widely adopted approach to attributed importance of neurons to their magnitude and selectivity of activation in response to specific stimuli. In this context, the study only investigated the effect of single unit ablations on network performance, however, this does not allow to determine whether the ablated unit is single handedly important or whether this unit is part of an important path through the network that has been altered by the ablation. In a future research effort, it's planned to address this issue aiming to identify such important paths along the network.

5.2 Investigating Learned Representations in Motor Control

In the context of this thesis, the conducted research in the domain of motor control and the corresponding results have been published and peer-reviewed in two papers in 2021 [185, 186]. In the same fashion is for the computer vision domain, the following subchapters largely repeat content of the papers and in part replicate some text passages with some minor changes or without any change. Additionally, the presented results are discussed in the context of the thesis and specifically in relation to the research questions.

5.2.1 Research Study 4: Influence of Network Ablations on Activation Patterns

The fourth study [185, 187] builds upon the insights of the previous studies, supports previously given answers to the first two research questions and addresses the third research question investigating the lack transparency for deep reinforcement learning agents and their learned behavior. So far, the previous studies have investigated the role of single neurons and the organization of learned representations in terms of how neurons form functional groups or how they are distributed across network layers. However, the relation between these organized representations and the emerging behavior of the network hasn't been addressed in detail since the previous studies were conducted within the realms of supervised learning and simple image classification, which doesn't allow for the emergence of complex behavior other than a simple choice of a specific class. In the context of recent research and the way towards general-purpose AI, deep reinforcement learning (DRL) algorithms were utilized in environments with sparse rewards and complete information, like the game of Chess of Go, or in complex multi-agent environments with incomplete information, like the game of Dota 2 or StarCraft II, to learn complex behavior constituting a multitude of micro and macro decisions rather than a single decision for a class. However, the research path leading up to today's pinnacle of these applications is marked by a crisis of reproducibility and required intense manual trial-and-error efforts such as finding a good network initialization and subsequent hyper-parameter tuning, which can make all the difference between a working and a failing solution [86]. What complicates the problem even more is that many working solutions are interspersed with unwanted behavioral artifacts that manifest in the learned policy of agents, if the environment allows for such manifestation, e.g., in the domain of learning locomotion [188]. Such artifacts are commonly caused by incentivizing an agent to solely maximize a possibly richly shaped reward without any constraints on its policy. The usual approach of training agents to maximize their cumulative reward and quantitatively evaluating them solely based on this reward or any other performance metric, such as the ELO rating in Chess, makes it difficult to trust an agent without a clear understanding of how its behavior emerges from its internal processes and the complex interplay of its individual functional components?

This study addresses the third research question by investigating the behavior of DRL agents in three different classic control environments based on the learned representations of their policy networks, aiming to find a link between these representations and different behavioral stages during the execution of the trained policy. To this end, the agents' learned representations are characterized based on

their layer activations during the execution of the policy. Network ablations are used to intentionally damage agents, evoking malfunctioning behavior to compare the representations of the fully intact and damaged networks to each other.

5.2.1.1 Key Contributions of the Study

The first key result is coherent with the results of the previous studies and further solidifies the answers given to the first and second research question by investigating the impact of network ablations with different sizes in different layers on the agent's capability to solve its trained control task. It shows that the larger the fraction of ablated neurons, the stronger the impact on the agent's performance. Consistent with the results of the third study, which showed that the investigated networks were fairly robust against ablations, the agents exhibited a task specific robustness to these ablations depending on the size and location of the ablations.

The second key result addresses the second research question specifically and shows that the activations of single neurons contribute to solving the control task by coordinating and forming specific correlation patterns between these activations and the executed actions during an episode. Specifically, the healthy networks demonstrated specific correlation patterns that get distorted significantly as a result of ablations leading to low returns. Interestingly, some ablations do not lead to lower returns and do not distort the correlation pattern significantly. This result gives an answer to the second research question as it describes a way of organizing and coordinating neuron activations that emerges during training as a result of learning to solve the given control task. It further scratches the surface of the third research question as it relates this organized and coordinated neuron activations to the agent's observable behavior, i.e., the success of its executed policy with respect to reaching a high episodic reward and solving the control task.

The third key result directly addresses the third research question and demonstrates that the undamaged agents show a distinct pattern in the temporal evolvement of the actor's layer activation, i.e., the undamaged agent's learned representation contains distinct activation states that can be directly linked to the different behavioral stages of the policy that successfully solves the control task, ultimately providing a link between the agent's behavior and its internal processes. This pattern gets distorted as a result of network ablations and the comparison to the baseline pattern of the undamaged agent yields insights into how the malfunctioning agents' behavior is related to an alteration of the network's learned representation.

5.2.1.2 Methods and Experimental Design

In order to address the third research question and investigate the relation between a network's learned representation and its emerging behavior, this study investigated the behavior of DRL agents in three different classic control environments based on the learned representations of their policy networks, aiming to find a link between these representations and different behavioral stages during the execution of the trained policy. To this end, the agents' actor networks' learned representations are characterized based on their layer activations during the execution of the policy. Network ablations are used to intentionally damage agents, evoking malfunctioning behavior to compare the representations of the fully intact and damaged networks to each other.

Experimental Setup

The agents were trained in three different classic control environments, namely the cart-pole swing-up (CPSU) environment, the pendulum swing-up (PSU) environment and the cart-pole balance (CPB) environment (cf. Figure 5.25). Although each environment poses an individual challenge, they share the partial objectives of controlling a cart on a rail or balancing a pendulum/pole in an upright position, providing some degree of comparability of the observed agent's behavior across tasks. At this point, a more detailed explanation of the intricacies of these environments regarding their state space, action space and reward functions is left out as they are well-known benchmark environments for DRL research and have been extensively explained elsewhere [189, 190].

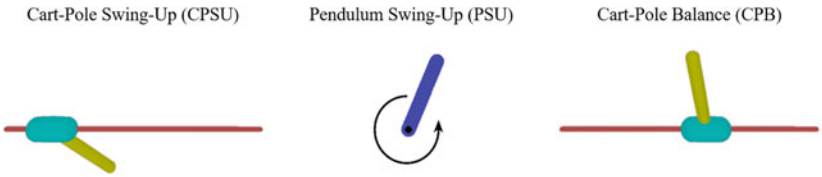


Figure 5.25 Three exemplary rendered images of the respective control environments. Taken from [185]

The object of investigation in this study is an actor-critic agent trained in the three described environments with the deep deterministic policy gradient algorithm as outlined in [85]. Both, the actor, and the critic network consist of two hidden layers with 400 neurons in the first layer and 300 neurons in the second

layer with both layers using ReLU activation and layer normalization [191]. The critic is supplied with the actor's chosen actions, which is superimposed by an Ornstein–Uhlenbeck noise process [192], only in the second hidden layer. Each agent was trained for 800, 000 time steps and optimized via Adam [193] with all other hyper-parameters being the same as in [85]. All computations were performed on a single machine containing two Intel Xeon Platinum 8168 processors with a total number of 48 physical cores and 8 NVIDIA Tesla V100 32G GPUs.

Characterization of Learned Representations

The actor's learned representations are characterized based on its layer activation during policy execution. Network ablations are used to intentionally damage the actor, evoking malfunctioning agent behavior to compare the representations of the fully intact and damaged networks. To this end, the activations of each single neuron within the fully intact actor and its predicted actions for each time step of an episode are recorded in addition to the cumulative episodic reward to establish a baseline recording. Additionally, the same data for each individual ablation case is recorded to compare it to the baseline recording.

Network Ablations

Partial network ablations were performed in a single layer with varying proportions of ablated neurons by manually clamping their activations to zero, effectively preventing any flow of information through the ablated neurons. The number of ablated neurons was selected in a range from 5% to 90% in steps of 5% until 30% and then in steps of 10% until 90%. In addition, it was deviated from this pattern once by ablating 33.33% of neurons within a layer. Thereby, the ablated neurons are selected in a sliding window manner that is shifted across the layer, similar to sliding a kernel over an image in a CNN while the window position is frozen during an episode. Note that the total number of ablations with the same proportion varies because they depend on the size of the layer, the size of the window and the stride of the window. For instance, in a layer with 300 neurons and a chosen window size of 5% with a stride of 10 neurons, 15 neurons are ablated at once resulting in 29 different network ablations in total. For all ablations, a constant stride value of 10 neurons were chosen to gather sufficient activation recordings for statistical analysis while at the same time keeping the computational efforts manageable.

Extraction of Activation Patterns

To determine how single neurons contribute to the control task, the Pearson correlation coefficient of its set of activations $A_{\{i,j\}} = \left\{a_{\{t\}} | t \in [0, T]\right\}$ and the outputs of the actor network $U = \left\{u_{\{t\}} | t \in [0, T]\right\}$ has been calculated for each time step within an episode, where t denotes the time step within the episode, T denotes the total number of time steps per episode, i denotes the i-th layer and j the j-th neuron within that layer. Furthermore, to characterize the learned representations within a layer of the actor, the activations of each single neuron in that specific layer for each time step of an episode are stored in a matrix $M^{\{T \times N\}}$, where T denotes the number of time steps per episode and N denotes the number of neurons per layer. The evolvement of the actor's activation during an episode is visualized using an open-source Python implementation of UMAP (cf. study three) to embed the stored activations into a two-dimensional space, i.e., $M \in \mathbb{R}^{\{T \times 2\}}$. Thus, each point in the embedded space represents the activation of a specific layer of the actor network for a single time step of an episode. Just as in study three, the default parameters for the UMAP embeddings were chosen after an initial attempt for finding better values for the number of nearest neighbors or the minimum distance between data points yielded no significant visual improvement of the embeddings.

5.2.1.3 Results

The results of this study are separated into three sub-chapters. First, the results of network ablations of groups of individual neurons with varying group size is presented investigating the effects of ablations in the two layers of the actor network on the agent's ability to achieve a high episodic reward. Subsequently, the impact of ablations on the activity of single neurons and the resulting characteristic activity pattern of the network, which leads to a high episodic reward, is investigated. Finally, the impact of network ablations on the network's whole layers' activity and the change of the resulting behavior is investigated.

Impact of Network Ablations on the Agent's Capability

To establish a baseline evaluation, the healthy agent was trained to achieve near state-of-the-art results in all three environments, i.e., a maximum total episodic reward of 886.4 for the CPSU task, −275.87 for the PSU task and 1,000 for the CPB task. For reasons of performance comparability across the three environments, the absolute return is normalized so that the minimum return value in each environment is 0 and the respective baseline return value is 1.

Figure 5.26 shows the normalized return for the baseline in comparison to all 29 network ablations in the first and second layer with a window size of 30% (120 neurons) for the three control tasks. For both swing-up tasks, most ablations in the first layer have a negative impact on the agent's capability to solve the tasks. Consistent with the observations of the previous studies, there are some ablations that have little to no impact or even a positive impact, thus increasing the return. In case of the CPB task, ablating 30% of the neurons in the first layer does not affect the agent's capability to solve the task at all. Contrary to the first layer, all ablations in the second layer have a strong negative impact for the CPSU task and the CPB task (except for two cases), however, only a few ablations have a comparably negative impact for the PSU task, where many ablations have little to no impact or even a positive impact. The negligible impact of ablations suggests that either the capacity of the network has not been exploited to its fullest extent so that some neurons do not contribute to solving the task and could be pruned or that the information represented by the ablated neurons is redundantly represented by other neurons making the agent robust against network ablations. The positive impact of ablations suggests that some neurons may play competing roles in the learned representation and that resolving this competition by targeted ablations improves the agent's capability to solve a task. Both observations are consistent with the results of the studies 1–3.

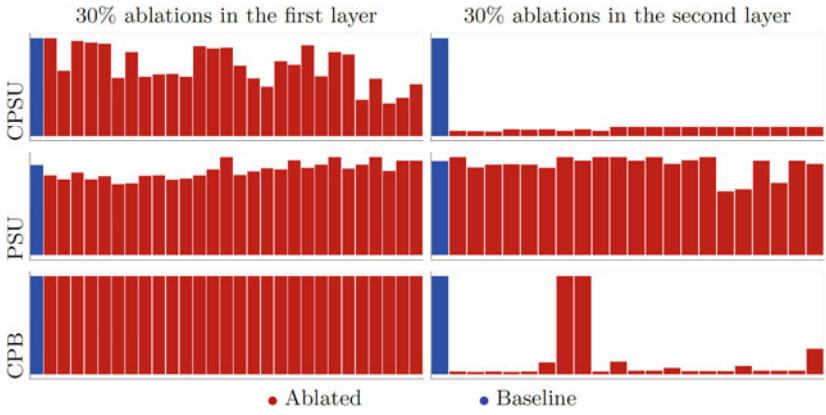


Figure 5.26 Comparison of the normalized returns achieved as a result of ablations of 30% of the neurons (red bars) to its respective baselines (blue bars). Taken from [185]

Figure 5.27 shows the distributions of the normalized returns resulting from the different network ablations in the first layer and second layer for the three control tasks. On average, the return decreases proportionally to the number of ablated neurons. Comparing the impacts in the first layer across the three tasks shows a similar trend for the CPSU and the PSU task, i.e., a slow but steady decrease of the achieved return with increasing sizes of ablations but a much more robust behavior for the CPB task, where ablations of up to 50% generally do not affect the agent's capability to solve the task. Further, comparing the impacts in the second layer shows a similar trend for the CPSU and the CPB task, i.e., a strong and sudden decrease in the achieved return for small ablation sizes, but a much more robust behavior for the PSU task, where ablations of up to 33.33% only marginally affect the agent's capability to solve the task. Interestingly, connecting the similarity of the ablation impacts with the similarity of the different tasks suggests that the first layer holds a representation of how to swing up the pole/pendulum while the second layer holds a representation of how to control the moving cart. More precisely, ablations in the first layer impact the agent in both tasks, in which a pole has to be swung up, while the representation for the task, which merely requires balancing the pole, is very robust against ablations in this layer. Analogously, ablations in the second layer strongly impact the agent in both tasks, in which a cart has to be controlled, while the representation for the task without a cart is fairly robust against ablations in this layer. These results suggest that interlinked learning objectives to solve the task such as controlling the cart, swinging up the pendulum and subsequently balancing it, are represented in different locations of the network. These observations are consistent with previously reported findings on the localized representations of specific classes in supervised trained neural networks on image classification tasks [135, 194, 195].

Impact of Ablations on Single Neuron Activity

Following the observations described above, the question arises what role the precise interplay of single neuron activity plays with respect to the agent's executed policy. More specifically, whether the contribution of single neuron activity to the executed actions during an episode shows a distinct pattern for the healthy agent and to what extent this pattern is distorted in case of ablations with a negative impact on the achieved return. To answer this question, this pattern is characterized via the set of Pearson correlation coefficients calculated for the activations of single neurons within a layer and the outputs of the actor network for each time step within an episode (cf. section "*Extraction of Activation Patterns*").

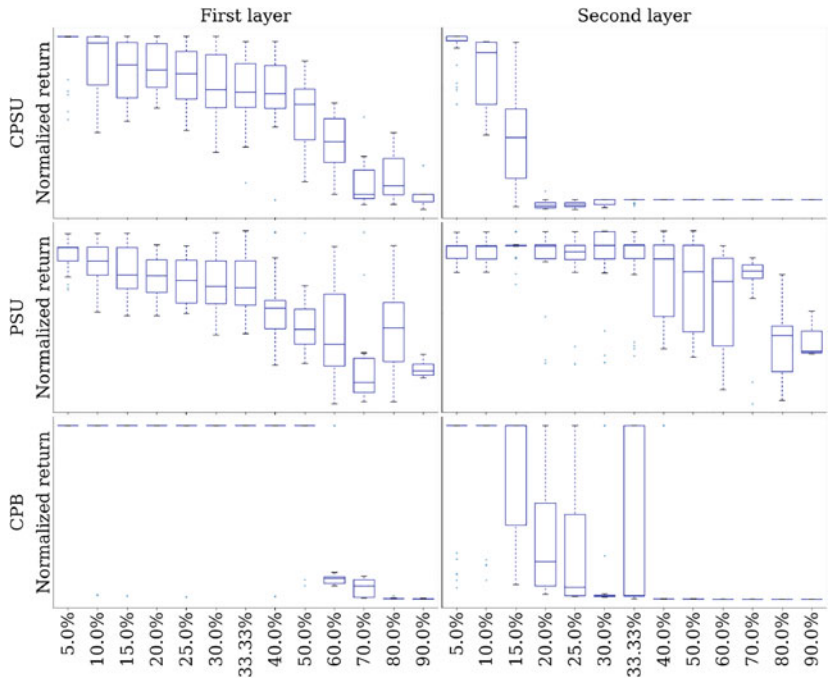


Figure 5.27 Distributions of the normalized returns for all ablations performed in the first layer (left side) and second layer (right side). Taken from [185]

Figure 5.28 shows this pattern for the baseline and four exemplary ablations of 5% of neurons in the first layer activated in the CPSU task. Each row contains 400 entries corresponding to the 400 neurons in the first layer. Each entry contains the correlation value and shows how the neuron's activation correlates with the actor's chosen action. The empty spaces in the rows show the ablated neurons, for which no correlation coefficient is calculated. The top row shows the baseline correlation pattern in comparison to the following four rows, which show the correlation patterns corresponding to the four exemplary ablations. The bottom four rows show to what extent the patterns resulting from the ablations change compared to the baseline pattern, specified by the difference between the baseline pattern and the ablation patterns. The ablations of neurons 100 to 119 and 270 to 289, resulting in the agent's failure to solve the task, show a general increase in correlation between the single neuron activity and the chosen

actions and the strongest difference of the pattern compared to the baseline. A high correlation value indicates a neuron's exclusive contribution to a specific control direction, i.e., whenever the cart is moved to either side, specific neurons are selectively active and contribute to the control in a specific direction. However, such distinct contributions of single neurons do not seem to resemble a robust representation as patterns with less distinct correlations between single neuron activations and the chosen actions generally lead to higher returns. This observation shows some similarity with the previously reported results about the importance of single neurons in supervised trained networks for image classification tasks in the studies 1–3. Specifically, networks that memorize well instead of generalizing are more reliant on neurons that show a high selectivity in their activation for specific classes, indicating that neurons, which selectively get activated for specific classes do not contribute as much to a robust and generalized representation as neurons with a less selective activation [77].

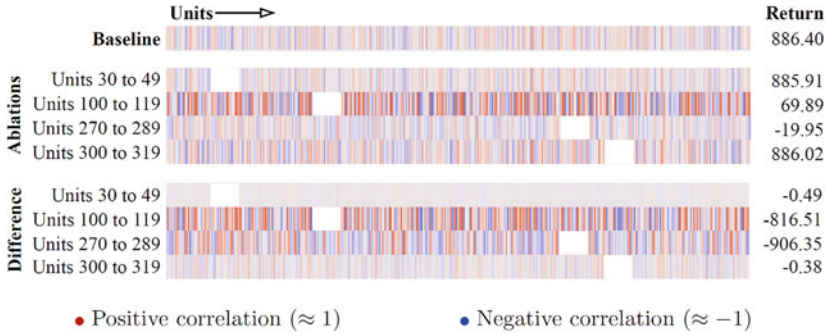


Figure 5.28 Correlation pattern of the activations of all 400 neurons in the first layer during the CPSU task for the healthy agent (baseline) and four exemplary ablations, as well as the change of these patterns compared to the baseline (bottom four rows). Taken from [185]

In order to further solidify that notion, the mean and the variance of the correlation patterns of all ablations are compared with the mean and the variance of the baseline pattern, hypothesizing that high values for the mean and the variance, corresponding to strong and distinct correlations, result in a low return. Figure 5.29 shows a scatter plot of the mean and the variance of the correlation patterns for the baseline and all 29 ablations of the size of 5% and their corresponding returns. Confirming the hypothesis, ablations of neurons resulting in large values for the mean and the variance, e.g., neurons 100 to 119 (marked

in the top right corner of the scatter plot) lead to low returns. Almost all other ablations with mean and variance values close to the baseline (points within the red ellipsis) do not result in task failures but achieve returns comparable to the baseline. Interestingly, the ablation of the neurons 270 to 289, which results in small values for the mean and the variance, also leads to a low return, suggesting that the hypothesis can be extended towards small values for the mean and the variance, corresponding to no clear contribution for most of the single neurons to the control task.

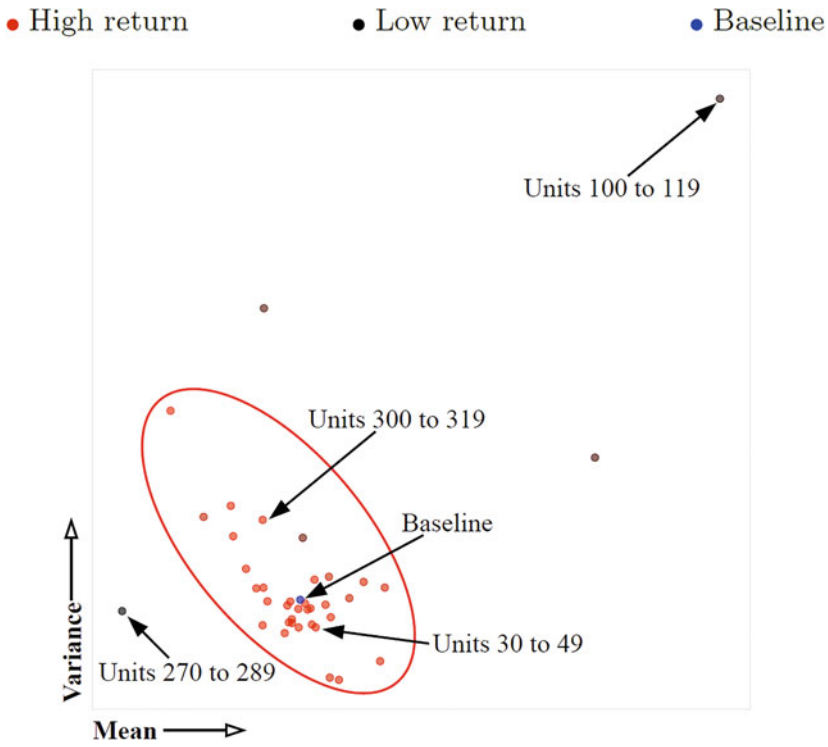


Figure 5.29 Scatter plot of the mean and the variance of the correlation patterns for the baseline and all 29 ablations of the size of 5% and their corresponding returns in the CPSU task. Taken from [185]

To further test the validity of the hypothesis across different sizes of ablations and across the three tasks, Figure 5.30 summarizes the effects of all ablations (5% to 90%) on the return and the dependency on the characteristics of the correlation patterns. Analogously to Figure 5.29, the x- and y-axis show the mean and the variance of the correlation patterns. For the CPSU task, the highest return is generally achieved for patterns with a low variance as ablations leading to larger variances show a decreased return. This suggests that the CPSU task requires single neurons to be generically involved in the control task and not to specialize too strongly on specific controls. On the contrary for the PSU task, higher returns are generally achieved for patterns with a high mean and high variance, suggesting a further refinement of our hypothesis with respect to task specific characteristics. Interestingly, ablations that increase both values beyond the baseline lead to even higher returns while patterns with low values lead to low returns. This suggests that the ability to swing-up the pendulum requires the neurons to contribute to the control in a very specific rather than generic way. Consistently, a very clear picture emerges for the CPB task, where no swing-up is required and only patterns with low values for mean and variance result in high returns, verifying our initial hypothesis. In combination with the CPSU task, this suggests that the ability to control the moving cart requires a generic involvement of single neurons in the control task rather than specific roles.

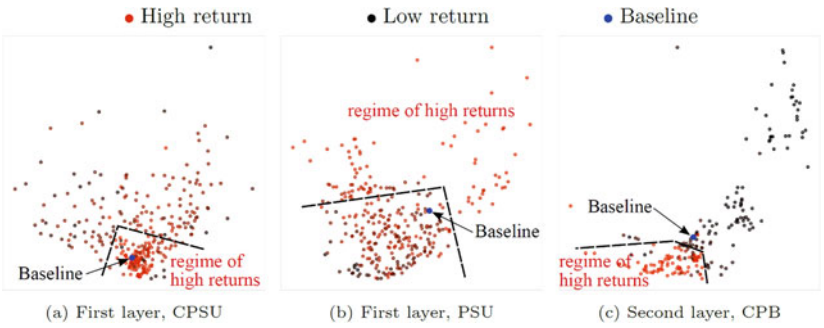


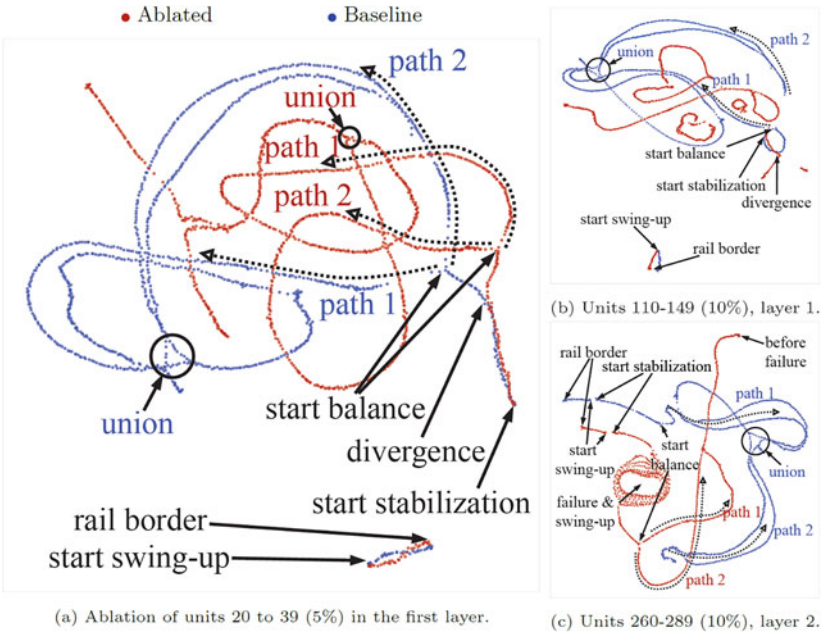
Figure 5.30 Scatterplot showing the mean (x-axis) and variance (y-axis) of the correlation coefficients for all ablations of the specified layer. Taken from [185]

Impact of Ablations on Layer Activation

Although the correlation patterns provide some insights on how the agent acts, they do not capture the temporal evolvement of the learned representations and do not answer questions with respect to such evolvements, e.g., at what point during the episode does the agent fail? When does it diverge from the baseline behavior and in what way? Does the agent go through different behavioral stages during an episode, and can these stages be linked to specific patterns in the learned representation? In order to answer these questions, the learned representations are characterized by embedding the layer activations recorded during an episode (cf. section "Characterization of Learned Representations") and compare the representations of the baseline to the representations resulting from the ablations.

Figure 5.31 shows this comparison for three exemplary ablation cases for the CPSU task. Each scatter plot contains 1,000 blue and 1,000 red points corresponding to the layer activation for each time step during an episode for the baseline and the ablation case, respectively. Note that even though the baselines in (a) and (b) show the exact same values, they are embedded slightly differently as the embeddings were calculated separately for all cases. The three cases correspond to ablations, which had no effect on the agent's capability to solve the task (cf. Figure 5.31 (a)) or which led to only half the return of the baseline (cf. Figure 5.31 (b) and (c)).

Figure 5.31 (a) shows the evolvement of the layer activation during an episode for the healthy and the damaged agent and how the different behavioral stages of the episode are linked to different sections of this evolvement. Both, the healthy and the damaged agent, start with moving the cart to the side, accelerating the pendulum to swing it up. After the initial swing-up (upon reaching the rail border), the agent is required to compensate for the excess momentum of the pole via corresponding cart movement to stabilize its upright position. This change in behavior results in a jump in the activation space from the initial activation path that corresponds to the initial swing-up behavior to another path that corresponds to the stabilization behavior. The difference in activations is likely due to the movement of the cart into the opposite direction upon reaching the rail border. Following the successful stabilization, the agent is required to balance the pole by rapidly switching directions of the cart to maintain an upright pole position. Interestingly, this behavior is represented in the activation space by two paths, along which the layer activation progresses as the agent acts throughout the episode. The layer activation repeatedly switches between these two paths suggesting that the network constantly changes between two distinct activation



(a) Ablation of units 20 to 39 (5%) in the first layer.

(b) Units 110-149 (10%), layer 1.

(c) Units 260-289 (10%), layer 2.

Figure 5.31 Comparison of the temporal evolvement of layer activations between the baseline and three exemplary ablation cases for the CPSU task. Taken from [185]

states corresponding to the balancing act of the pole. At some point during the episode, these two paths merge (union) as the balancing act leads to an almost static position of the cart and the pole. However, from a mechanical perspective, this constitutes an unstable equilibrium point for the pole, where small perturbations of the pole's angular position result in its downfall triggering a renewed balancing act that is resembled by a renewed separation of the merged paths. This observation suggests that the convergence of the actor's activation towards a single final activation state is not sufficient to solve the task. Rather, a stable and continuous transition between two distinct activation states is necessary to sufficiently represent the balancing act. This observation seems somewhat surprising considering the weak correlations of single neuron activity to the actor's chosen actions throughout an episode (cf. section "Impact of Ablations on Single Neuron Activity"). Although the single neuron activity does not correlate strongly with the network's executed actions, their combined activations lead to two distinct

activation states of the network, each of them corresponding to the movement of the cart in either one of the two possible directions during the balancing act. This suggests that single neurons do not contribute individually to the control task, but rather as part of a larger conglomerate of neurons that constitute the two different activation states.

Figure 5.31 b) shows an ablation case, for which the agent fails to balance the pole continuously after the initial swing-up and drops it after a short period of holding it in the upright position, reattempting the swing-up and balancing act. The layer activation diverges slightly from the baseline right from the start of the swing-up and further diverges completely after a short period of the stabilization phase. Consequently, due to this divergence, the layer activation of the damaged agent does not show the emergence of two distinct paths connected to the balancing act as the agent never succeeds in stabilizing the pole compensating its excess momentum after the initial swing-up.

Interestingly, the existence of two distinct activation states is not exclusive to the actor's first layer but also apparent in its second layer. Figure 5.31 c) shows an ablation case in layer two, in which the failure of the agent is caused by a drop of the pole after the initial swing-up and a short period of balancing, causing the pole to rotate at high speed until the end of the episode. The blue points resemble a similar pattern of the second layer's activation compared to the first layer including the divergence of the activation along two distinct paths, the attempt to merge these paths and the renewed separation. The failure of the agent, i.e., the continuous rotation of the pole at high speed, is visible in the activation space by the circularly arranged red points, from which the agent is not able to recover back onto the stabilization path and both connected paths corresponding to the balancing act.

5.2.1.4 Summary and Contribution of the Results to the Research Questions

This study corroborated previous results regarding the first two research questions and addressed the third research question aiming to understand how DRL agents act based on characterizing their learned representations of their policy networks. Specifically, the role of single neurons for the control task was investigated and it was shown that despite the absence of a strong correlation between their activations and the actor's chosen actions throughout an episode, agents, that solve their tasks successfully, show task specific patterns of weakly correlated single neuron activity that get distorted by network ablations leading to low returns. The importance of these patterns for a successful solution of the control task suggests

that the careful interplay between single neurons with respect to the executed policy is essential rather than their sole and isolated behavior. This finding partly answers the third research question as it uncovers a direct relation between the organized activity of neurons, i.e., structure and organization of the network's learned representation, and the emerging behavior.

Further detailing that relationship, the temporal evolvement of the actor's layer activations during an episode was investigated and showed that in case of the CPSU task, the consecutive steps executed during the episode to solve the task are precisely represented by the policy network and mapped onto its layer activations. It further showed that this mapping is essential for solving the task as its distortion as a result of network ablations leads to low returns and failed attempts to solve the task. The arrangement of the consecutive points in the embedded activation space revealed that the agent runs along specific paths in its activation space and that diverging from this path is fatal for its task performance. The most striking observation of these paths is given by the fact that the actor's layer activations can be very different for very similar states. The naive expectation is that the layer activation would converge to a single specific activation vector just as the consecutive states to be processed by the network become more and more similar to each other as the pole is balanced. However, it turned out that this is not the case, suggesting that the learned representations may contain some information that is encoded in the temporal dimension on which the states are ordered, i.e., that similar states evoke a different activation of the network depending on when it is presented to the network. These findings give answers to the third research question and demonstrate how temporal aspects that are important to the learning task are organized in the learned representation of the network and how they relate to the network's emerging behavior.

5.2.2 Research Study 5: Relation Between Neural Activations and Agent Behavior

The fifth study [186] further addresses the third research question and details the previously given answers by exploring universally observable relations between the structure and organization of learned representations that emerge in DRL agents trained to perform tasks requiring a specific type of movement, i.e., a particularly useful behavior to solve the given task. In neuroscience, movements are commonly distinguished into four categories: 1) involuntary and non-rhythmic,

i.e., reflexes like the Patellar tendon reflex or the swallowing reflex, 2) involuntary and rhythmic, i.e., subconsciously performed movements like breathing and the heartbeat, 3) voluntary and non-rhythmic, i.e., consciously performed movements like grabbing an object, and 4) voluntary and rhythmic movements, i.e., consciously performed repetitive movements like walking and running [196]. Due to the range of available RL environments, this study focusses on agents trained on continuous control tasks that require them to learn behaviors resembling the fourth category of motion, i.e., repetitive movements to progress in the environment and reach a high reward. The study investigates how the learned behavior of the agents is reflected in their neural activity and whether individual neurons can be accounted for the formation of such repetitive movements, thus, partly answering the third research question.

5.2.2.1 Key Contributions of the Study

The first key result builds on the insights gained from the previous study and addresses the third research question laying the foundation to connect a network's internal processes to its exhibited behavior. Specifically, the first key result is the characterization of the agents' learned behaviors via defined anchor points of their behavioral routines (behavioral anchors), i.e., individual parts of the complete task behavior that are formed repetitively during task performance.

The second key result further expands that foundation and describes the universality the behavioral anchors as different agents with different neural networks solving the same task share these behavioral anchors.

The third key result builds on that foundation and partly answers the third research question showing that the formation of these behavioral anchors can be attributed to a specific set of neurons in the controlling neural networks with similar activations and that these results are consistently found across agents despite their differently seeded weight initializations, network architectures and individual training phases.

The fourth key result further solidifies the answer to the third research question and addresses the universality of these findings by generalizing the analysis approach across different continuous control domains. The formation of behavioral anchors that are represented by a specific set of neurons within the network is consistently observed across different control domains and is not specific to individual tasks.

5.2.2.2 Methods and Experimental Design

In order to corroborate the results addressing the third research question and investigating the universality of the relation between structured and organized learned representations and the emerging behavior, the study investigates a number of different networks trained in six different motor control domains.

Domains and Tasks

The agents were trained in continuous control domains from the DeepMind Control Suite [197] that require a continuous stream of performed actions by the agents to receive high rewards. Additionally, the agents were trained in one domain, which does not require such a perpetual movement, to highlight the contrast of the learned behaviors. The most pronounced difference between both task types is given by the learned action sequences of the agents. In case of the perpetual movement tasks, which do not have a final state to be reached, the action sequences show repetitive patterns that are observed periodically throughout task performance. In case of the non-perpetual task, the agents can reach a final state, which continuously yields a high reward without further actions necessary, with a single distinct sequence of actions. For the perpetual movement tasks, five well-known and widely used continuous control domains were chosen, i.e., the cheetah domain with the run task, the walker domain with the run task, the quadruped domain with the run task, the finger domain with the spin task and the hopper domain with the hop task. For the non-perpetual movement task, the ball-in-cup domain with the catch task was chosen. Details about the state and action spaces of the individual domains can be found in the original publication [197] and on GitHub [198]. Figure 5.32 shows the different bodies controlled by the agents for each domain.

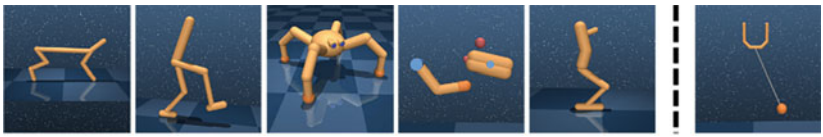


Figure 5.32 Overview of the used domains from left to right: cheetah, walker, quadruped, finger, and hopper (perpetual) vs. ball-in-cup (non-perpetual). Illustrations taken from [197]

Agents and Training

All agents were trained with a PyTorch implementation of the state-of-the-art Soft-Actor-Critic (SAC) algorithm [199]. To account for the obtained results being restricted to specific network design or weight initialization, twelve different agents that differ either in size of their individual neural network layers or seed that was used to initialize their network weights were trained for each domain. Inspired by the topologies commonly used in actor-critic approaches for continuous control tasks [85] all agents' actor and critic networks contained two fully connected layers with either 384, 512, 640 or 768 units per layer, and each network size was initialized with three different seeds before training, resulting in twelve different agents. An individual number of training time steps for each domain was chosen to achieve comparable performance levels to state-of-the-art results except for the hopper domain. To account for variances and instabilities during the training phases, the agents' performances were evaluated on 100 evaluation episodes after training. As a single evaluation episode lasts 1,000 timesteps and each timestep can yield a maximum reward of 1, the maximum possible reward of the whole episode is capped at 1,000 points. For reproducibility reasons, Table 5.1 summarizes the parameters used for the SAC algorithm and Table 5.2, Table 5.3, and Figure 5.33 summarize the training parameters of the individual agents and their respective performances. Despite the low rewards in the hopper domain reflected by the accumulation of the reward distribution on the left side, the agents still formed repetitive movements suitable for the analysis approach.

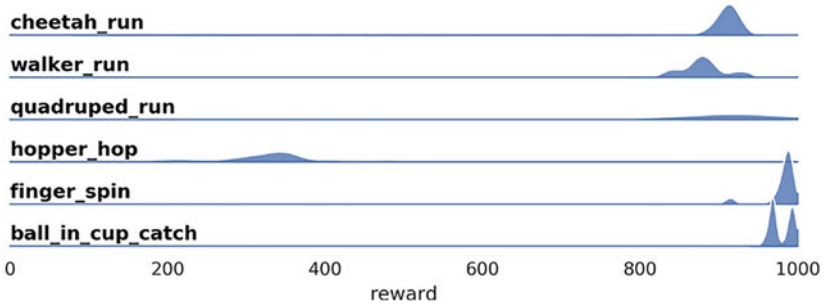


Figure 5.33 The reward distribution of the 100 evaluation episodes for each domain show that except for the hopper domain all other agents achieved state-of-the-art performance levels. Taken from [186]

Table 5.1 Summary of the SAC algorithm parameters

Parameter	Value
replay buffer capacity	2, 000, 000
optimizer	Adam
activation	ReLU
number of initial steps	100, 000
batch size	1, 024
discounting factor γ	0.99
temperature α	0.1
learning rate	10^{-4}
target smoothing coefficient	0.005
actor update frequency	1
target update frequency	2

Table 5.2 Summary of the agents' neural network parameters and performances

ID	#Neurons	Seed	Reward *"cheetah run"*		
			mean	min	max
1	384	1	892	847	898
2	384	2	913	906	915
3	384	3	903	847	908
4	512	11	901	814	915
5	512	12	916	913	918
6	512	13	917	908	919
7	640	21	878	116	917
8	640	22	897	176	919
9	640	23	918	915	920
10	768	31	914	907	917
11	768	32	915	831	918
12	768	33	915	911	916

Table 5.3 Summary of the number of training steps per domain

Domain	cheetah	Walker	quadruped	hopper	finger	ball in cup
#Steps	4, 000, 000	5, 000, 000	5, 000, 000	4, 000, 000	2, 000, 000	1, 000, 000

Data Generation

For the analysis of the agents' behaviors and corresponding neural activities, data from 100 evaluation episodes of each trained agent were acquired. For each time step of an evaluation episode, the observation and action values obtained from the respective domains as well as the single unit activations of the agents' actor networks were stored. The observation data consists of the agent's body's joint positions and joint velocities, the agent's center-of-mass velocity and possibly touch sensor readings, depending on the domain. The action data contains the applied forces and torques on the agents' limbs and joints. The neural activity data contains the activation values of all individual neurons of both layers of the actor networks after ReLU activation. The dimensionalities of the resulting data arrays vary depending on the domains and network architectures.

Analysis Approach

The analysis is motivated by the goal to link observable and repetitive agent behavior to its corresponding neural activities. Specifically, it's aimed to investigate whether this kind of behavior emerges from the activation of individual neurons or groups of neurons. To this end, the continuous and repetitive nature of the agents' observed behavior is characterized via visual inspection of the agents' trajectories through their respective observation spaces that are traversed during task execution. An agent's behavioral anchors are defined as specific locations in the observation space that are repetitively visited by the agent along its cyclic trajectory to fulfill its task. These specific locations are determined via clustering in the observation space and by counting the number of times the identified clusters are visited along the agent's trajectory during task execution. Furthermore, whether the repetitive nature of the agents' behaviors can be attributed to individual neurons or groups of neurons is investigated via two different metrics. First, the cluster specific selectivity of the activations of single neurons is calculated and determined a neuron to be important for the formation of a behavioral anchor when its activation is highly selective for the corresponding cluster. Second, random forest classifier is trained to predict the cluster based on the activation values of the single neurons and determined a neuron to be important for the formation of a behavioral anchor based on the corresponding feature importance value, i.e., the gini impurity (cf. section "Gini Importance") of that feature for the classifier. The assessments of single neurons' importance determined by both metrics is compared by calculating the Spearman's rank correlation coefficient between both importance rankings with respect to the different clusters and show that both metrics yield similar results. Figure 5.34 illustrates the analysis approach

with respect to the goal of relating behavior in the observation data to the agents' activation data.

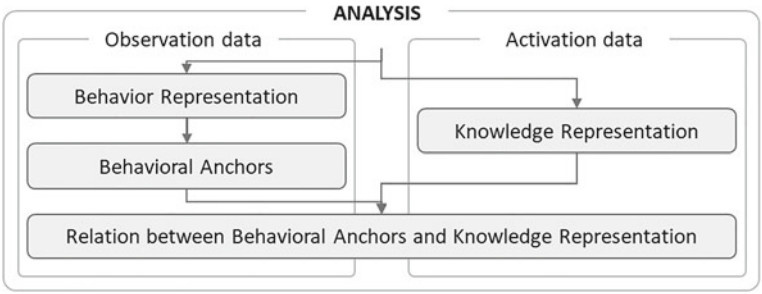


Figure 5.34 Illustration of the analysis approach aimed at relating learned agent behavior to its neural activity. Taken from [186]

5.2.2.3 Results

The results of this study are separated into four sub-chapters. First, the behavior of a single instance of an agent trained in the cheetah environment is determined to lay the methodical foundation for the characterization of agent behavior. Subsequently, reoccurring behavior shared across several agents with different network architectures, called behavioral anchors, is determined and provides the foundation of the third sub-chapter, which investigated the representation of these shared behavior anchors by the agents' neural network activations. Finally, universality of the results is investigated by transferring the approach to other motor control domains.

Single Instance Observations and Behavior

The first part of the study addresses the question of how agent behavior is represented within the observation data and how behavioral sequences can be determined and characterized. In the context of this study and in regard to reinforcement learning, an agent's behavior is defined as the trajectory through the observation and action spaces throughout one episode. The specific focus on the observation space seems sensible since the succession of observations represents the perceptible part of agent behavior. To investigate representations of behavior within the observation space, the observation data of one single agent is embedded in a latent space via principal component analysis (PCA) as a linear

projection method and uniform manifold approximation (UMAP) as a nonlinear projection method. For UMAP, an initial search for its parameters "effective minimum distance between embedded points" and "size of local neighborhood" was conducted that yielded visually appealing results in terms of the separation of individual points and groups of points. To make the different observations comparable to each other, each observation dimension was scaled to zero mean and unit variance.

Figure 5.35 shows the embedded observation data for PCA (left) and two different parameter settings of UMAP (right) as well as the agent's trajectory through the observation space. The cyclic path of the trajectory reflects the repetitiveness of the agent's movements throughout the episode. Each observation value is colored according to the reward received after its individual time step. The projection reveals a structural organization of the embedded points corresponding to the received reward with small rewards being placed in the center and large rewards being placed on the edges. At the start of the episode, the agent's trajectory starts in the center with no reward. Throughout the episode, the agent traverses along its trajectory, gains more and more speed and collects increasingly higher rewards.

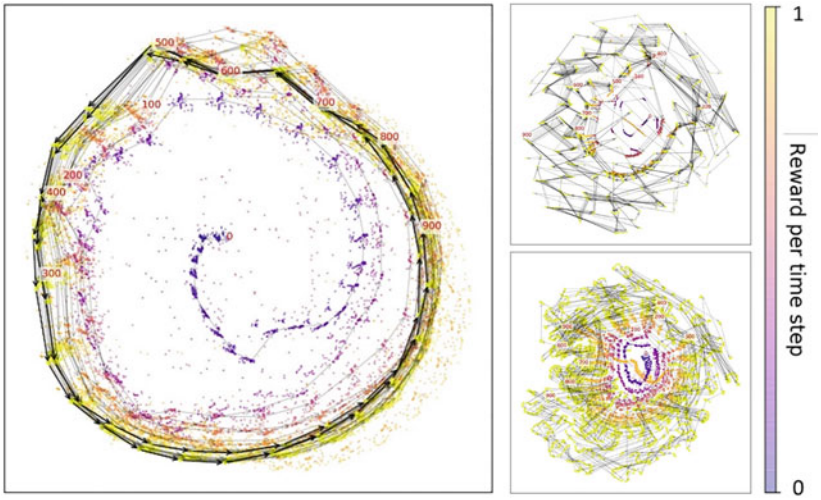


Figure 5.35 Example trajectory of one agent trained in the cheetah domain through the embedding space (PCA left, UMAP with different sets of parameters right) showing a circular path reflecting the repetitive nature of the learned behavior. Taken from [186]

Qualitatively, PCA produced more meaningful and more interpretable visualizations than UMAP, probably due to its linear projection since the points in the PCA embedding overlap more strongly and create a more evenly distributed circle. The better interpretability of linear projections compared to non-linear projections for high dimensional spaces, i.e., of neural network activations, has been discussed in more detail by Li et al. [200]. Quantitatively, both embeddings were evaluated via their trustworthiness [201]. While both, PCA and UMAP, attain high trustworthiness scores (> 0.97) for smaller numbers of considered neighboring points (~ 100), PCA outperforms UMAP (0.96 vs. < 0.90) with increasingly larger neighborhoods (> 5000). Henceforth, the presentation of the further results is limited to the PCA embeddings.

Shared Behavioral Anchors

Following the characterization of an individual agent's behavior, exploring whether such behavior is shared across different individually trained agents was the next step of the study. Such shared behavior, called behavioral anchors, is defined as locations in the collectively embedded observation space, which are repeatedly visited by the agents. It turned out that agents with different neural networks solved the same task in a similar manner by visiting similar behavioral anchors along their trajectories.

Figure 5.36 shows the collectively embedded observation data of all twelve agents trained in the cheetah domain distinguished by different colors. The depicted trajectory is an exemplary trajectory of one individual agent. The red numbers along the trajectory correspond to the index of the time step for the respective point. A single episode contains 1,000 time steps with the agent starting in the center of the plot and moves to the border of the embedding space until in cycles along the trajectory at the border. Except for a single agent (yellow/green points), all agents show a similar distribution of observation data along the depicted trajectory reflecting a similar behavior with respect to repetitive movements.

The locations reflecting the behavioral anchors were determined via k-means and DBSCAN clustering in the original and embedded observation space and by counting the number of times these clusters are visited along the agent's trajectory during task execution. An extensive parameter study to find the best parameters for the "number of clusters" (k-means), the "maximum distance between two neighboring points" and the "minimum number of samples per cluster" (DBSCAN) was conducted for different choices of dimensionalities of the observation space. The considerations ranged from the complete space with all its

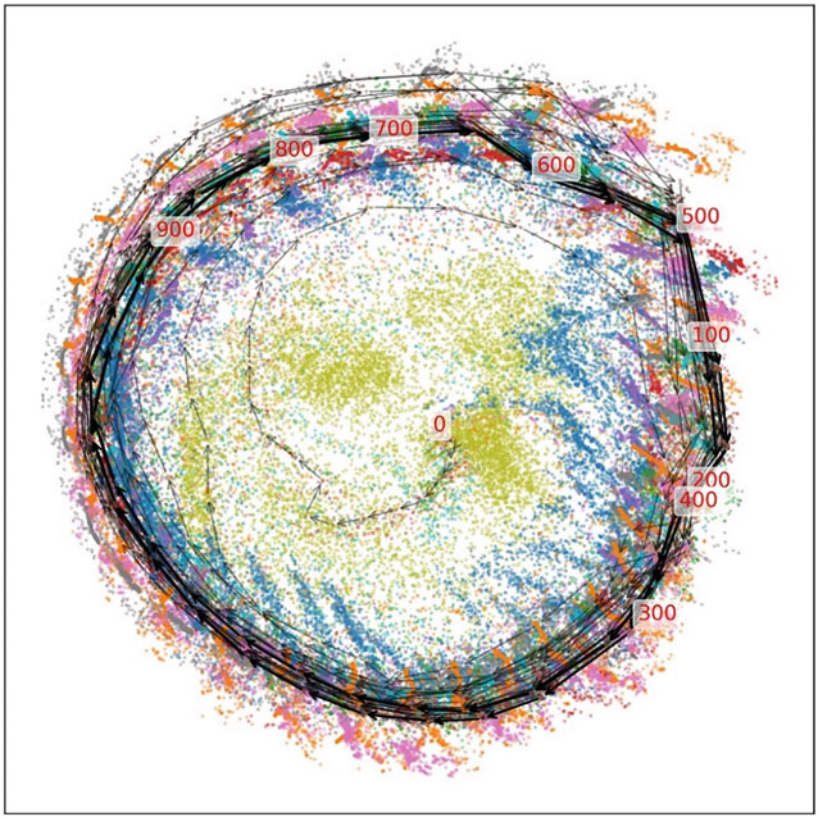


Figure 5.36 Example trajectory of one agent through the collective embedding space showing a circular path reflecting the repetitive nature of the learned behavior. Taken from [186]

original dimensions to only the two most important dimensions after dimensionality reduction via PCA or UMAP. The study showed that k-means clustering (with $k = 100$), performed on two dimensions in the PCA-embedded observation space, yields the most meaningful separations estimated based on visual inspection of the resulting scatterplots. The visual impressions are supported quantitatively by evaluating the clustering results with three metrics (Silhouette coefficient [202], Calinski-Harabasz index [203], Davies-Bouldin index [204]), which give different measures of the quality of a clustering and yielded consistent results with respect to their evaluations.

Figure 5.37 shows the clustering result in the collectively embedded observation data of all twelve agents trained in the cheetah domain with the individual clusters coded in different colors. Two subsets of cluster specific images showing the agent-controlled body are given below, demonstrating the similarities of the joint positions. In the first cluster, the cheetah's posture is characterized by the wide stretch of both its legs corresponding to jump mid-air to cover as much distance as possible moving forward. In the second cluster, the cheetah's posture is characterized by its front paw touching the ground preparing for the next jump forward. Note that some images do not seem to fit solely based on the joint positions, however, half of the observation space, i.e., the joint velocities, is not visible on the images but contribute to the calculation of the similarity between two observations in the embedded observation space. The separation of behavioral anchors via clustering each individual agent's observation space is more distinct compared to the collective embedding space. This indicates that each agent develops individual behavioral sequences that are distinct in some observation dimensions compared to other agents but ultimately produce a similar overall behavior, i.e., a repetitive forward movement.

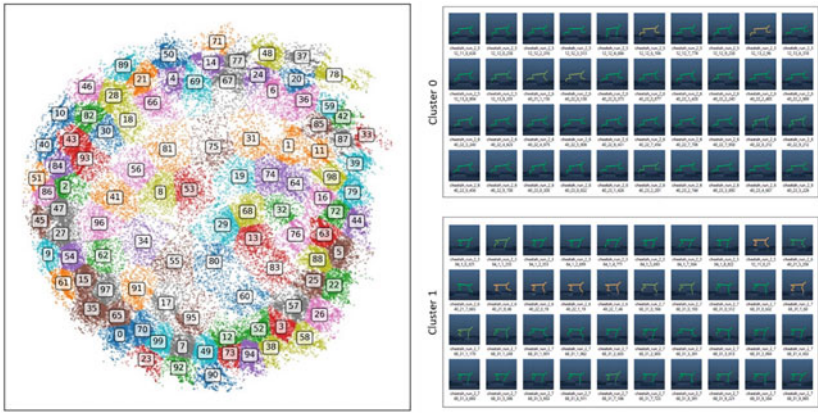


Figure 5.37 Clustering of collective embedding with images of two exemplary clusters demonstrates similarities between body positions of different agents within clusters. Taken from [186]

Representation of Behavioral Anchors via Network Activity

Considering that network activity revealed that populations of neurons in the mammalian motor cortex are selectively activated for specific motor control routines contributing to specific movements, the question arises whether this phenomenon can be observed in artificial neural networks, i.e., whether the identified behavioral anchors, which constitute a long sequence of repetitive behavior, can be related to the activity of individual neurons or groups of neurons in the agents' actor networks. To this end, the importance of single neurons was determined for their respective contributions to form behavioral anchors via two metrics.

First, similar to the third research study, the selectivity of each neuron's activation with respect to the individual behavioral anchors was calculated, which shows whether neurons are selectively activated for specific anchors or whether they are activated equally for all anchors. A neuron's activation selectivity was calculated with respect to the various behavioral anchors, i.e., the individual clusters in the embedded observation space, according to to

$$selectivity_{i,j,c} = \frac{a_{i,j,c} - a_{i,j,rest}}{a_{i,j,c} + a_{i,j,rest}}$$

where $a_{i,j,c}$ is the mean activity of the i-th neuron of layer j for cluster c and $a_{i,j,rest}$ is the mean activation of neuron i for the remaining clusters excluding c. A neuron's cluster selectivity can take values between -1 and $+1$. A positive/negative value means that the neuron's activity for a specific cluster is larger/smaller than its average activity for all the other clusters.

Second, for each cluster in the embedded observation space a binary random forest classifier was trained to predict whether a cluster's anchor behavior is formed by the agent based on its single neuron activations. The gini impurity was used to determine how well an individual neuron's activation can be used for the anchor prediction. Each random forest consisted of 100 estimators. The gini impurity can take values between 0 and 1, where 0 means that a feature is irrelevant for the classification and 1 means that a feature can perfectly classify the entire dataset.

Figure 5.38 shows the distributions of both metrics aggregated for all agents and all clusters in both layers for all domains (color coded). In these distributions, an individual neuron is activated multiple times in different evaluation runs and each activation value contributes to the distribution individually with no averaging or aggregation calculation performed. In general, most neurons are not highly important individually for the formation of specific behavioral anchors as they have low values for both their selectivity and feature importance for most of

the clusters (high bars to the left side of the scale of the x-axis). The second layer contains slightly more specialized neurons with high values of selectivity and feature importance compared to the first layer as indicated by a) the steady increase in bar heights towards higher selectivity values in layer 2 compared to the decrease after some point in layer 1, and b) the longer tail of the bar distribution for the feature importance for layer 2 as compared to layer 1. This observation is consistent with the hierarchical organization of neural networks, which forms more general representations in upper layers and more specialized generalizations in deeper layers [146].

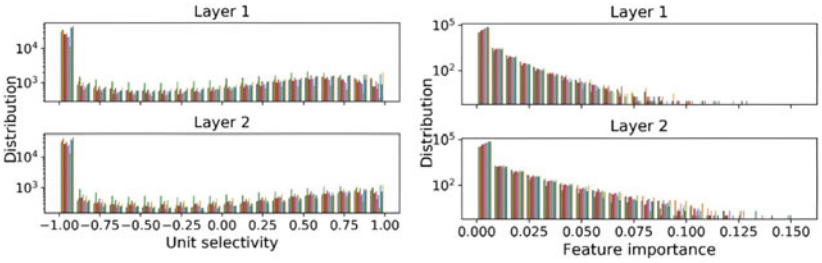


Figure 5.38 Distribution of unit selectivity (left) and feature importance (right) values for all instances. Taken from [186]

The above distributions raise the question whether neurons are consistently selective or important across multiple evaluation runs. Thus, an individual neuron's average selectivity and importance was calculated across the 100 evaluation runs and across all 100 clusters. Figure 5.39 shows the average selectivity and feature importance in descending order for all neurons in both layers of all agents trained in the cheetah domain. The plots reveal that only a small fraction of neurons is consistently selective (with values > 0) across different runs and clusters while most neurons are not.

Given the selectivity of the activation of some neurons for certain clusters, the question arises whether those clusters, for which neurons are most selective or most important, are behavioral anchors, i.e., clusters that are visited more frequently during task execution than clusters. Figure 5.40 shows the average selectivity and feature importance for the individual clusters in descending order as well as the number of times a cluster is visited by the agents along their respective trajectories. In general, the plots reveal no direct link between the degree of selectivity of a cluster and the number of times this cluster is visited

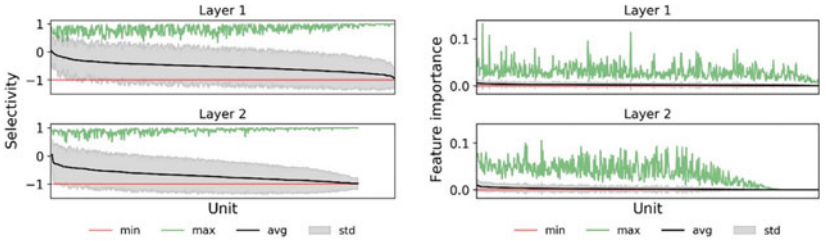


Figure 5.39 Average unit selectivity (left) and feature importance (right) sorted in descending order per neuron with minimum and maximum value. Taken from [186]

along the agent's trajectory exists. However, the clusters with the highest average neuron selectivity generally show a low number of occurrences. This suggests that some behaviors connected to these clusters are evoked by individual neurons, however, they do not seem to be too relevant for the overall solution of the task, as they are not visited frequently by the agent along its trajectory. This means, that behavioral anchors are not necessarily determined by the selectivity of neurons, nor their importance. It remains to be seen, how the behavioral anchors relate to neuron activity.

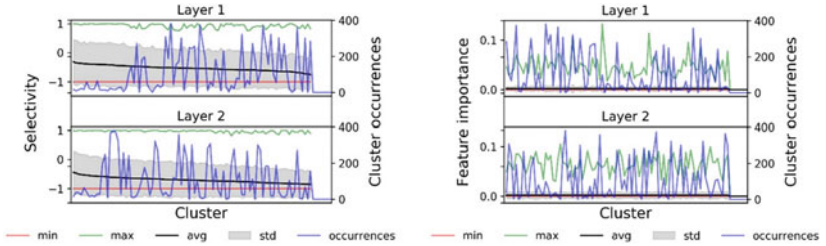


Figure 5.40 Average unit selectivity (left) and feature importance (right) sorted in descending order per cluster with minimum and maximum value as well as number of cluster occurrences. Taken from [186]

Considering that two metrics were used to determine a neuron's importance, the question arises whether both metrics attest to a neuron's importance in the same way. Figure 5.41 shows the spearman rank correlation between the neurons' selectivity and their feature importance. Despite their different distributions

(cf. Figure 5.38), both metrics are strongly correlated suggesting that estimating a ranking of importance for the individual neurons based on either metric yields comparable results. Considering that the unit selectivity is an activation amplitude-based metric, and that the calculation of the feature importance is also based on the individual neurons' activation amplitude, the result suggests that the activation amplitude is a valid indicator for a neuron's importance. This result is consistent with the well-established notion of assessing an individual motor cortex neuron's importance for specific movements in neuroscience studies.

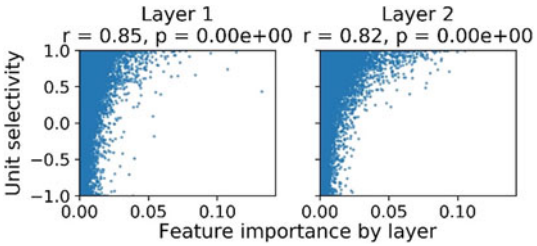


Figure 5.41 Spearman rank correlation between unit selectivity and feature importance for both layers of the actor network trained on the cheetah domain. Taken from [186]

The relation between both metrics and their coherent interpretation with respect to the importance of single neurons is supported by the visual inspection of the embedded activation space. For this purpose, a UMAP embedding is computed for the neurons' activations. A grid search across the parameters of the *local neighborhood size* and *minimum distance between neighbors* revealed $min_{dist} = 0.1$ and $n_{neighbors} = 15$ to yield good visual results of the embeddings. Figure 5.42 shows the embeddings of unit activations in both layers of the actor network for all 100 evaluation runs in the cheetah domain. Each dot in the scatter plot represents the activation value of an individual neuron where dots close to each other have similar activations and dots far away from each other have dissimilar activations during task execution. Each dot is color coded according to their individual cluster selectivity with red dots having high positive values and blue dots having high negative values. The four examples correspond to four distinct clusters in the observation embedding of the agent. The four examples reveal that the neuron's activations are clustered and organized in the activation embedding space with respect to the cluster selectivity. This means that depending on the agent's body posture, which is directly related to the clusters in the observation space, the neurons in its actor network exhibit distinct activations. More specifically, neurons that are determined to be most important for specific behavioral anchors have similar activations during task execution.

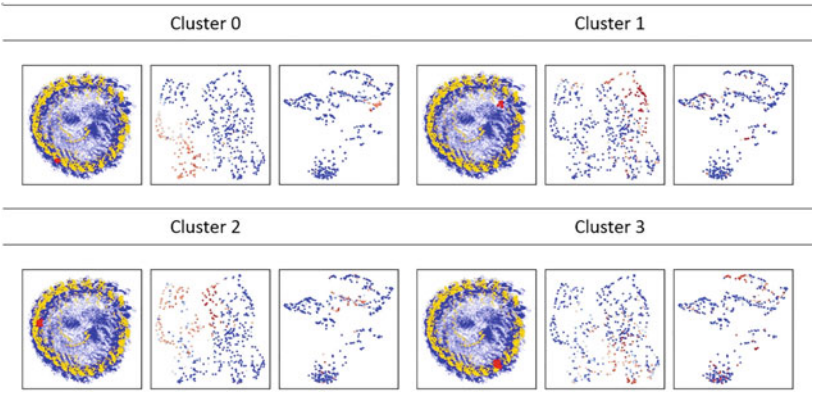


Figure 5.42 Four examples of the embedding of neuron activations colored by cluster selectivity. Taken from [186]

Figure 5.43 shows one exemplary case for cluster 0 (cf. Figure 5.42, top left), which compares the two metrics by which neuron activations are related to agent behavior. The right-hand side color codes the individual neurons according to their feature importance with high values being colors red and low positive values close to zero being blue. Similar to the cluster selectivity, the scatter plot reveals an organization of the most important neurons determined by the feature importance, which group together in a specific area of the embedded activation space. Consistent with the correlation result between both metrics shown in Figure 5.41, the color coding according to both metrics reveals the most important neurons in similar locations in the activation space. The overlap in layer two is much more distinct while some neurons that show a high selectivity do not necessarily show a high feature importance. The organization of important neurons characterized by their grouping in the activation space show some differences between layer 1 and layer 2. In general, the neurons in layer 2 are separated more distinctively than the neurons in layer 1, i.e., the neuron activations in layer 1 show smoother transitions from one neuron to another while the activations in layer two can be grouped more distinctively.

Transfer and Generalization to other Domains

To corroborate the methodical approach for analyzing agent behavior and investigating the connection to neural network activations as well as to validate the universality of the results presented so far, the analysis approach is transferred to

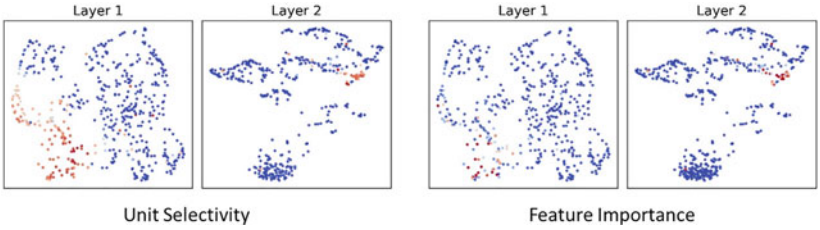


Figure 5.43 Embedding of units colored by neuron selectivity (left) and feature importance (right) of each neuron. Taken from [186]

other continuous control domains (cf. Figure 5.32). This allows to draw generalized conclusions that are consistent across domains and prevent finding results too specific to the agents' training environments. Note that the analysis methods and used parameters were not adapted for each domain specifically, but rather the approach as developed for the cheetah domain was transferred as is. Although domain specific parameter tuning would have likely yielded better trained agents, there is no reason to assume that the general gist of the observed results would change significantly.

Figure 5.44 shows example trajectories though the embedded observation spaces of trained agents in the domains *walker*, *quadruped*, *finger*, *hopper*, and *ball-in-cup* (from left to right). As the ball-in-cup domain does not require perpetual motion executed by the agents, it is not considered for further analysis and merely used as a contrast domain. Note that most of the dots in the scatter plot for the hopper domain (fourth panel from the left) are dark blue representing low rewards due to the poorly trained agents that did not learn good policies for the domain (cf. Figure 5.33). However, despite the low rewards, the agents nevertheless learned to form repetitive movements reflected by the closed loop trajectory, which largely contained movements connected to falling over and getting back up to hop forward. Just as in the cheetah domain, the repetitive behavior of the trained agents with different networks and differently initialized seeds is clearly reflected in the cyclic trajectories through the PCA embedded observation space. The embedded points are structured by reward, i.e., points corresponding to high rewards are close to each other and far away from points corresponding to low rewards. It turned out that depending on the domain the cyclic path has more pronounced edges in the trajectories, i.e., in the walker domain, indicating more distinct components within agent behavior.

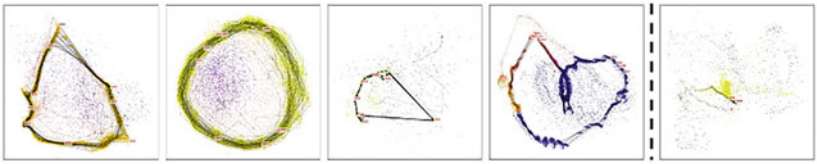


Figure 5.44 Example trajectories trough single individual agents' PCA embedded observation spaces in different domains. From left to right: walker, quadruped, finger, hopper, and ball-in-cup. Taken from [186]

Similar to the cheetah domain, behavioral anchors for each domain are determined by collectively embedding the observation space of all instances and clustering the embedded space using k-means clustering. Figure 5.45 shows example trajectories though the collective embedding spaces of trained agents in the domains *walker*, *quadruped*, *finger*, *hopper*, and *ball-in-cup* (from left to right). As in the cheetah domain, the circular characteristics of the trajectories confirms the repetitive nature of the agents' movements in these domains, supporting the universality of this observation across network topologies and training environments.

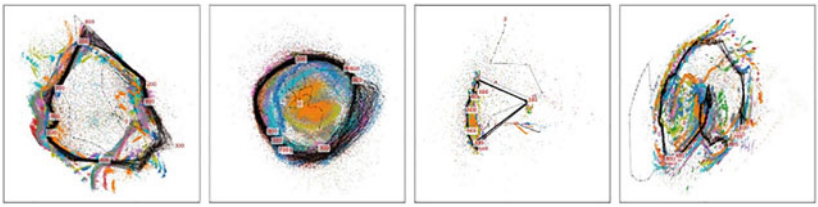


Figure 5.45 Example trajectories through the collective embedding spaces in different domains. From left to right: walker, quadruped, finger, and hopper. Taken from [186]

Behavioral anchors in the other domains were determined via k-means clustering in the collective embedding space of the trained agents in the respective domains. Depending on the domain as well as varying across clusters, the determined behavioral anchors showed different degrees of clear separation for the agents' body positions. Linking the behavioral anchors to the agents' neural network activations yields similar results as in the cheetah domain. Both importance

metrics, unit selectivity and feature importance, show a significant, positive correlation with variable strength ranging from 0.6 to above 0.8. Mapping the neuron importance values onto the neuron activation embeddings supports these correlations visually. Figure 5.46 demonstrates this finding exemplary for the two domains walker and quadruped.

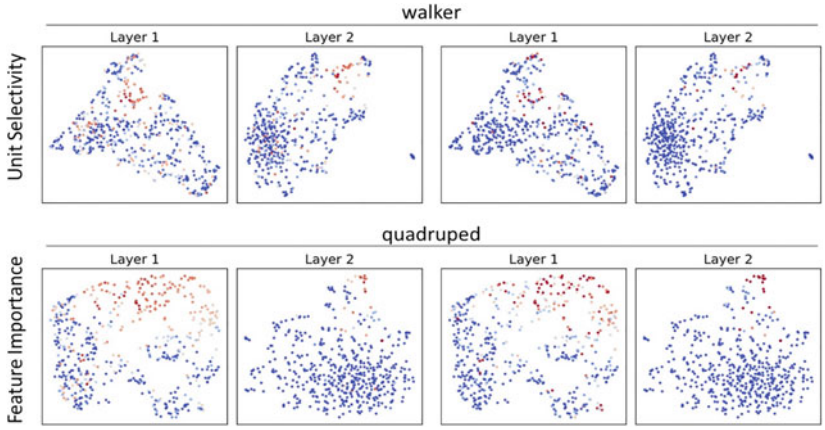


Figure 5.46 Embedding of units colored by unit selectivity (left) and feature importance (right) of each unit for walker and quadruped domain. Taken from [186]

5.2.2.4 Summary and Contribution of the Results to the Research Questions

Aiming to address the third research question, the study explored the learned behaviors of DRL agents trained to form repetitive movements in continuous control domains and explored how their behaviors can be characterized and how it relates to the agents' actor networks' neural activities. It showed that the repetitiveness of the behavior can be characterized via behavioral anchors, i.e., anchor points in the agents' observations spaces that are frequently visited along cyclic trajectories traversed during task execution. It found that agents with different neural networks solved the same task in a similar manner, visiting similar behavioral anchors along their trajectories. It further showed that a specific subset of neurons is important for the formation of these anchors and that these neurons show similar activations during task execution. The universality of the results was demonstrated by applying the analysis approach to different neural network

topologies with different weight initializations trained in different continuous control domains. Thus, the results give a direct answer to the third research question and describe the relation of the organization of the policy network's learned representation, characterized by the grouped selectivity of single neuron activations, to the agent's exhibited behavior.

The results in the domains other than the cheetah domain are less pronounced, e.g., the clustering evaluation (cf. section "Transfer and Generalization to other Domains") is worse, the images in the clusters are visually more dissimilar and the correlation between the neuron selectivity and the feature importance is smaller. This is likely caused by the fact that no parameter tuning has been conducted for the transfer to these domains and the methods have been used as developed in the cheetah domain. However, the results of the transfer of the approach are solid enough to support the universal nature of the observations across agent topologies and domains. The general gist of the observations made in the cheetah domain, i.e., the observation of cyclic trajectories, the selectivity of neuron activation regarding behavioral anchors, and the coherence between that neural activation and the computed feature importance to determine the importance of individual neurons for forming specific behavioral anchors can be found in the other domains, too. Thus, the previously given answer to the third research question is solidified and demonstrated to be universally valid across different networks, control tasks and learning environments.

Transfer Studies

6

The following chapter contains two transfer studies investigating the learned representation of neural networks trained on sensor time series data from an industrial use case as a direct representative of the typical nature of sequential data. At the same time, the two studies demonstrate the use of the presented methods not only for facilitating transparency and interpretability for the learned representations of the trained networks but also for gaining additional insights about the manufacturing process, which provides the sensor data, that is otherwise inaccessible to human domain experts.

Deep Drawing of Car Body Parts

1. The first transfer study transfers the previously employed methods to time series data, a type of data that is frequently encountered in various domains and is most importantly provided in industrial scenarios utilizing sensor data systems. Specifically, the approach of ablations is used to investigate the learned representations of a network trained in a predictive quality scenario in the automotive manufacturing industry. Specifically, a convolutional neural network trained on sensor time series data obtained from a deep drawing tool to predict process failures and detect waste products is investigated with respect to how it recognizes these failures based on the process data. The results of the study address the fourth research question and demonstrate how ablations can be utilized in combination with other methods to facilitate transparency for the industrial manufacturing process that provides the sensor data that serves as the basis for the investigated learned representation.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-658-40004-0_6.

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_6

2. The second transfer study complements the previous results and investigates the recently introduced transformer architecture and its suitability to facilitate insights into the same manufacturing process that was subject to the previous study. Specifically, its attention mechanism is investigated with respect to its potentials to yield interpretable and transparent insights into how the network predicts process failures. The study expands the complexity of the learning task as it considers simultaneously acquired sensor data and investigates spatial and temporal relationships between the individual sensor data. Thus, the study addresses the fourth research question and provides insights about important sensors and important time periods for the prediction of process failures.

6.1 Transfer Study 1: Network Ablations for Deep Drawing

The first transfer study [205, 206] addresses the fourth research question and transfers some of the previously employed methods to investigate the learned representations of neural networks from fundamental research relying on synthetically constructed datasets to a real world-application. It aims to demonstrate how the insights gained about the structure and organization of learned representations help to understand the underlying processes from which the training data is acquired in a comprehensible manner. In this study, the data is acquired from an industrial manufacturing process for car body parts. More specifically, the data comes from a deep drawing tool extended with a sensor systems of strain gauges that yield high-frequency time series data acquired over the course of the manufacturing process. The manufactured products are made from flat sheet metal, which is deformed by the deep drawing tool to take the shape of a specific car body part. During that process, the sheet metal may crack, which constitutes a process failure. A cracked sheet metal, although not usable for further assembly of the car body, is not of too much concern regarding production cost, however, manufacturing a cracked sheet metal may potentially damage the deep drawing tool, which would cause high financial costs. Thus, it is desirable to be able to predict the occurrence of process failures, i.e., the manufacturing of cracked sheet metal, in order to preemptively take action and protect the deep drawing tool from serious damage. The deep drawing manufacturing step is the second step out of six consecutively executed production steps before the manual quality control process at the end of the line, which is performed by two human workers. They investigate the metal sheets via visual inspection with the help of fluorescent powder and black light illumination in order to spot rough surfaces that are

caused by cracks. If a cracked sheet is identified, it is sorted out and recycled, however, no data about the quality control process is saved and the cause of the crack is not determined. Thus, in a work preceding this study, the manual quality control process at the end of the line was supported by an automated and data driven solution to identify cracks right after the deep drawing process [205]. This automated solution provided the foundation for the labeled data and the learning task used and addresses in this study.

In this study, a deep neural network was trained in a supervised manner predict the occurrence of these process failures based on the sensor data from the strain gauges. The investigation of the learned representation yields valuable insights into the relevant features in the time series data that hint towards the occurrence of process failures. Consecutively, domain experts may facilitate these insights and relate the particular temporal courses of the time series data that lead to process failures to specific mechanism of the manufacturing process in order to determine the a) reasons for such failure occurrences and b) prevent them from occurring in the future.

6.1.1 Key Contributions of the Study

The first key result of the study addresses the fourth research question utilizing a saliency approach to determine important features of the input time series data for the learned classification task. The saliency method of choice is Grad-CAM, which was originally developed for image data and computer vision tasks but is transferred in this study to one-dimensional sensor time series data and utilized to extract the relevant time windows for the trained network to perform the classification task.

The second key result, too, addresses the fourth research question and utilizes network ablations to identify the relevant neurons within the network that represent the most important time series motifs found within the previously identified time windows. Specifically, the extracted series motifs within the time windows are deemed most important by domain experts with an intuitive understand of the meaning of the sensor data and its relation to the manufacturing process.

6.1.2 Methods and Experimental Design

In this study, a 1D-convolutional neural network was trained to process the high-frequency sensory time series data recorded from the strain gauge sensor system

during the deep drawing of the sheet metals. Specifically, the network was trained in a supervised manner to predict the occurrence of process failures based on the temporal characteristics of the time series data. Subsequently, the relevant information was extracted from the network, based on which the failure prediction is made, i.e., what temporal course of the sensory time series data has led to the prediction and when it occurred during the manufacturing process. The network was trained on two distinct datasets. The first dataset contains the complete course of the sensor time series including possible cracks. The second dataset, which poses an arguably greater learning challenge, uses the same labels as the first one but a modified version of the time series. Specifically, the time series are cut at specific point in time so that no actual cracks are included in data that is processed by the network. This way, the network is forced to detect the process failures based on features in the time series that do not directly correspond to the cracks but rather hint towards the future occurrence of a crack.

After successful training, Gradient-weighted Class Activation Mappings (Grad-CAM) [126, 207] was applied to the network to extract the most important time windows of the time series for the decisions of the network. Furthermore, ablations were used to identify the most important time series motifs in the data by monitoring the network's performance after ablations of individual filters (cf. section "Methods and Experimental Design" of the first study). This way, domain experts at the manufacturing site were provided with the relevant time series motifs that are the basis for the decisions of the network with insights regarding the localization of the corresponding most important filters within the network that store this information, facilitating transparency for further inspection and fine tuning of the individual network filters.

Use Case & Data Acquisition

Deep drawing is a sheet metal forming process in which a sheet metal blank is radially drawn into a forming die by the mechanical action of a punch [208]. The first transfer study [205, 206] addresses the fourth research question and transfers some of the previously employed methods to investigate the learned representations of neural networks from fundamental research relying on synthetically constructed datasets to a real world-application. It aims to demonstrate how the insights gained about the structure and organization of learned representations help to understand the underlying processes from which the training data is acquired in a comprehensible manner. In this study, the data is acquired from an industrial manufacturing process for car body parts. More specifically, the data comes from a deep drawing tool extended with a sensor systems of strain gauges

that yield high-frequency time series data acquired over the course of the manufacturing process. The manufactured products are made from flat sheet metal, which is deformed by the deep drawing tool to take the shape of a specific car body part. During that process, the sheet metal may crack, which constitutes a process failure. A cracked sheet metal, although not usable for further assembly of the car body, is not of too much concern regarding production cost, however, manufacturing a cracked sheet metal may potentially damage the deep drawing tool, which would cause high financial costs. Thus, it is desirable to be able to predict the occurrence of process failures, i.e., the manufacturing of cracked sheet metal, in order to preemptively take action and protect the deep drawing tool from serious damage. shows a schematic illustration of the deep drawing metal forming process. The data was acquired from a deep drawing tool in a press plant of a German car manufacturer. The tool was modified and enhanced by eight strain gauge sensors operating at 2 kHz sampling rage that were applied to the blank holder of the tool and by laser sensors at different flange retraction points. Over the course of 11 months, data of 4,251 single deep drawing strokes was acquired during prototype manufacturing of a specific car body part. The strain gauge sensors measure an electrical resistance, which is proportional to the longitudinal extension or compression of the strain gauge, which in turn is proportional to the amount of mechanical force absorbed by the bottom part of the tool that is being exerted by the punch. The laser sensors measure the distance between the flange retraction points and the metal sheet.

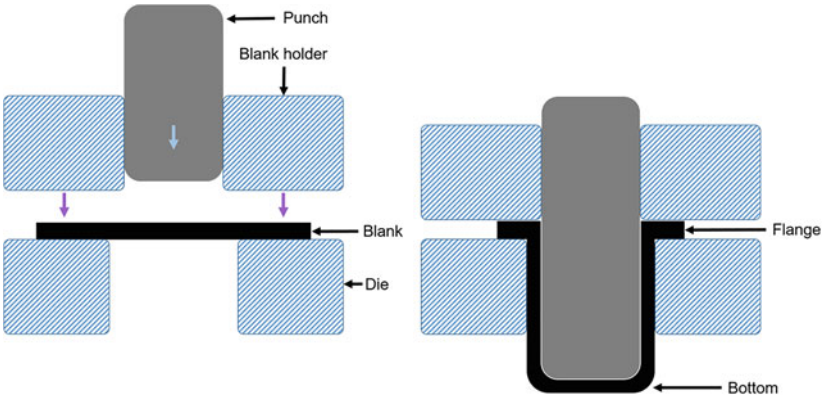


Figure 6.1 Schematic illustration of the deep drawing manufacturing process. Left: before the deforming process. Right: after the deforming process. Taken from [205]

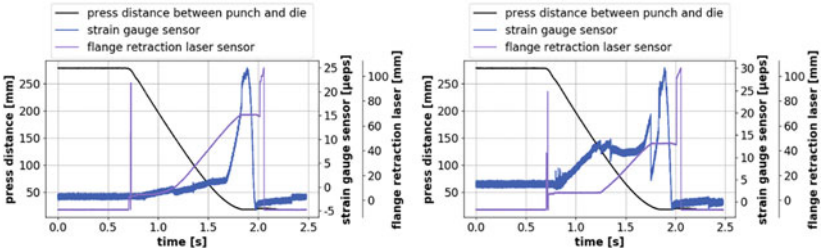


Figure 6.2 Left: exemplary time series data of a good stroke acquired from the deep drawing tool. Right: exemplary time series data of a bad stroke acquired from the deep drawing tool. Note that the sudden decrease of the strain gauge sensor data likely indicates a crack in the metal sheet. Additionally, the maximum distance measured by the flange retraction laser before the 2 seconds mark ($\sim$ 45 mm) is smaller as compared to the good stroke ($\sim$ 75 mm). Taken from [205]

Figure 6.2, left-hand side shows an example of the time series data acquired during a standard deep drawing stroke. The black curve shows the distance between the punch and the die. At the beginning of the process, the punch is at maximum distance of $\sim$ 250 millimeters and starts to move towards the die after $\sim$ 0.85 seconds. After $\sim$ 1.7 seconds, the die first contacts the metal sheet, and the mechanical force is transferred through the metal sheet to the bottom part of the press where the strain gauge sensors are placed. Upon contact, the strain gauge sensor signal (blue curve) shows the highest gradient.

The deforming process ends after $\sim$ 1.9 seconds, when the strain gauge sensor signal decreases strongly, and no more force is exerted on the bottom part of the press. As the metal sheet is being deformed, the flange retraction laser (purple curve) measures an increasing distance to the metal sheet. Figure 6.2, right-hand side shows an example of a bad deep drawing stroke. After $\sim$ 1.7 seconds, the metal sheet cracked during the deforming process and for a short period does not transfer the mechanical force to the bottom part of the press and thus to the strain gauge sensors. Therefore, the strain gauge sensor data shows a sudden drop of the signal as the strain gauge, which is continuously compressed during a standard deforming process, extends abruptly as the metal sheet cracks.

Learning Tasks

The goal of the study was to train two neural networks for two different tasks that differ in their level of difficulty. The first learning task is aimed to train a learning model to recognize cracks in the temporal course of one sensor signal.

Therefore, the input to the model is the whole strain gauge signal up until its peak value, i.e., potential cracks are contained in the temporal course of the data. Thus, the model is able to learn features in the temporal course of the sensor data that corresponds to the shape of cracks. The second learning task is aimed to predict the occurrence of cracks based in the temporal course of the strain gauge signal before they occur. To this end, the senor data signals are cut off at a point before the cracks occur, so that model cannot simply recognize cracks for what they are. Rather, the model is required to recognize patterns in the temporal course of the signal leading up to the critical deformation process, which may hint at future formation of cracks. The data from the flange retraction lasers is not used in the further analysis and discarded. Figure 6.3 shows two examples of a good stroke and a bad stroke as well as the distinction of the two learning tasks.

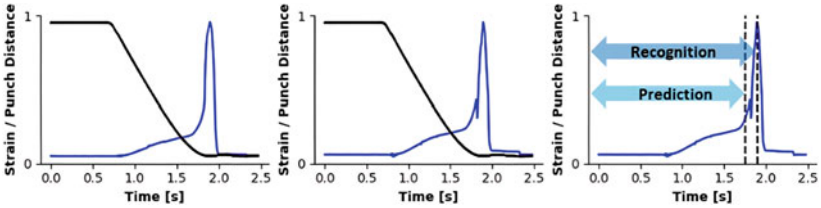


Figure 6.3 Examples of a good (left) and a bad (middle) deep drawing stroke as well as the signal usage for the two learning tasks. Taken from [206]

Data Preprocessing and Dataset Preparation

Since the acquired sensor data is not labeled on-line, an a-posteriori labeling process is required to prepare the data for the learning tasks. Before the automated labeling process, the data was cleaned from calibration strokes and highly irregular testing strokes with the help of a k-Means clustering approach separating the data into two clusters, one of which contains 926 strokes that were discarded for the further analysis of the data reducing the data to 3,325 strokes. k-means clustering was performed in a normalized two-dimensional space, where each stroke is characterized by its mean and its maximum value. Since the calibration and testing strokes typically have a low mean or are plain flat and have a low maximum value, the clustering in those two dimensions separates the strokes sufficiently. Figure 6.4 shows all recorded strokes and the k-means clustering result.

A crack can be identified in the strain gauge sensor data based on its development over time, which is the basis for the automated labeling procedure. The acquired data of a single stroke contains data points at the beginning period and the ending period of the time series during which the punch is not moving (cf. Figure 6.2, between 0.0 and $\sim$ 0.7 seconds and between $\sim$ 2.0 and $\sim$ 2.5 seconds). As the data contains no useful information, those periods are cut off from the time series. Before further processing, all 3,325 remaining strokes were interpolated to 2,320 data points so that all time series have the same number of datapoints, i.e., the input shape for the learning model is consistent for all strokes. Because real world sensory data is inherently corrupted with noise, before further analysis, the cut time series data was filtered with a digital forward-backward Butterworth filter [209] with a cutoff frequency of 120 Hz in order to clean the data from high frequency noise. Then, the gradient of the filtered time series data of the strain gauge sensors denoted by $grad(s(t))$ was analyzed to estimate whether the course of the time series resembles a crack or not. As the data is discretely sampled, the gradient at a specific sample point in time t_i is given by the difference quotient

$$grad(s(t_i)) = \frac{s(t_{i+1}) - s(t_i)}{t_{i+1} - t_i}.$$

Based on the experience of the domain experts, a crack was defined as the monotonic decrease of the strain gauge data, describing the duration and the severity of the crack. A minimum duration of 50 data points corresponding to 25 milliseconds with a minimum amplitude of 1.5 μeps, corresponding to a 5% drop of the signal with respect to its peak value was chosen as threshold parameters. This way, 120 cracks were identified post-hoc out of the 3,325 strokes resulting in a highly unbalanced ratio of 3.61% cracks to 96.39% non-cracks due to the well-optimized manufacturing process. Figure 6.5 shows a zoom in on the strain gauge sensor data of the last few millimeters of the process of two exemplary bad strokes and the result of the automated crack detection method. The sensory data is plotted against the travelled distance of the press, which corrects for different movement speeds of the press that result in a dilation or compression of the time series on the time scale. The black curves show the raw sensory signals of the strain gauges, and the red curves show the filtered data without noise. The grey areas correspond to the identified cracks based on the gradients of the strain gauge signals.

After preprocessing, the data was separated into training and test sets with an 80% to 20% ratio, which corresponds to a total number of 2,660 strokes in

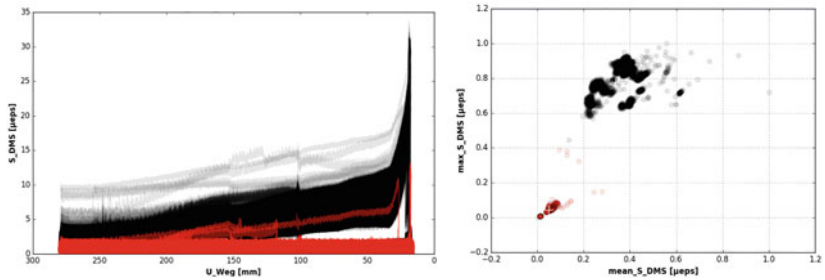


Figure 6.4 Left: Overview of all strain gauge signals plotted against the position of the movable punch. The closer the punch moves towards the die, the higher the strain gauge signals. The red signals correspond to calibration strokes and irregular strokes that are detected by the k-means clustering algorithm. Right: k-means clustering result of all strokes in two dimensions. The smaller, red colored cluster in the bottom left of the scatter plot corresponds to the red colored calibration and irregular test strokes. Taken from [205]

the training set and 665 strokes in the test set. The ratio of cracks and non-cracks in both sets were balanced equally so that the original ratio of cracks to non-cracks was preserved. Five individual splits of the training data for a subsequent evaluation of the network performances via five-fold cross-validation were prepared.

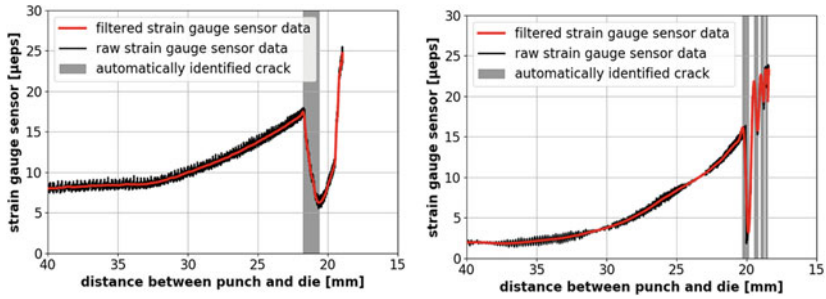


Figure 6.5 Zoom-in at the end of two exemplary strokes with cracked metal sheets. Left: a single large crack was identified at $\sim$ 23 mm. Right: several cracks are identified at different points during the process and with different degrees of severity. Taken from [205]

Model Design, Training and Evaluation

For both tasks, the learning model architecture employed is a shallow 1D-CNN containing four convolutional layers with intermittent max pooling layers after every second convolutional layer and three fully connected layers following the last pooling layer (cf. Table 6.1). All hidden layers use ReLU activation while the output layer utilizes SoftMax activation for the classification task.

Table 6.1 Neural Network architecture parameters

Name	Type	# Input channels	Input size	# Output channels	Kernel size	Stride
Layer 1	Conv	1	2320	16	200	4
Layer 2	Conv	16	531	16	40	2
Pooling	Max	16	246	16	2	2
Layer 3	Conv	16	123	32	20	1
Layer 4	Conv	32	104	32	10	1
Pooling	Max	32	47	32	2	2
Layer 5	FC	1504	–	128	–	–
Layer 6	FC	128	–	32	–	–
Layer 7	FC	32	–	2	–	–

Since the networks are trained to perform binary classification, they are evaluated accordingly. The four metrics used are 1) accuracy $\left(\frac{TP+TN}{N}\right)$, 2) True positive rate $\left(TPR = \frac{TP}{CP}\right)$, 3) False negative rate $\left(FNR = \frac{FP}{CP}\right)$, and 4) the F_1-score $\left(2 \cdot \frac{TPR \cdot Precision}{TPR \cdot Precision}\right)$, where Precision is $\left(\frac{TP}{TP+FP}\right)$. Here, $N = 3,325$ is the total number of strokes, TP is the total number true positives, i.e., network classifications that correctly predict a crack, TN is the total number of true negatives, i.e., network classifications that correctly predict the absence of a crack, CP is the total number of condition positives, i.e., cracks in the dataset, and FP is the total number of false positives, i.e., network classifications that wrongfully predict the occurrence of a crack. On both datasets, the trained network achieved an $F_1 score > 0.98$, correctly classifying cracks post-hoc as well as predicting the future occurrence of cracks, i.e., estimating if a crack is going to occur in the next second during the process.

The models were trained for 500 epochs on their respective training datasets utilizing Adam optimization and a learning rate of $\alpha = 0.003$ with a batch size of 200. Despite the highly unbalanced dataset, the model achieves good five-fold

cross validated, averaged scores of $F_1score = 0.9930$ for the simple recognition task and $F_1score = 0.9804$ for the more advanced prediction task. In the prediction case, the model's performance is worse with respect to detecting true negatives, i.e., correctly predicting the occurrence of cracks, than in the recognition task. Notably, both models exceed the statistical performance of a naive classifier that would classify every single stroke as a non-crack, which would achieve a high accuracy of 96.39%. The good true negative rates and the high $F_1scores$ indicate that neither model is naive in its decision-making, and both have successfully learned to solve its respective task. Table 6.2 summarizes the results of the trained models.

Table 6.2 Model performance for the recognition and prediction task evaluated via 5-fold cross-validation

Performance metric	Recognition	Prediction
Accuracy	98.90%	96.95%
True positives	99.46%	98.42%
True negatives	96.90%	91.64%
F1-score	0.9930	0.9804

6.1.3 Results

The results of this study are separated into two sub-chapters. First, relevant time windows for the classification of cracks are identified by applying Grad-CAM to the one-dimensional time series data. Subsequently, network ablations of individual filters are used to extract the corresponding time series motifs from the network, which are used to predict the occurrence of cracks.

Identification of Relevant Time Windows

To identify the relevant time windows, which contain the most important features for the networks to recognize/predict cracks, Grad-CAM was adapted for the 1D-CNN to calculate a one-dimensional saliency map, i.e., a saliency time series, for each time series signal. Subsequently, the time windows with the highest saliency values according to these saliency series were extracted. Figure 6.6 shows two exemplary strokes and their corresponding saliency series from the recognition test set. The blue curves show the strain gauge signals for a crack (left) and a non-crack (right). The red curves show the corresponding saliency series, which are scaled between 0 and 1 for each individual stroke to facilitate comparability across different strokes. In case of the crack, the highest saliency is placed on the

time window during which the actual crack occurs. This observation is consistent across all cracks. Naturally, the highest saliency is placed at the time window, which contains the strongest difference between cracks and non-cracks, which is the sudden decrease of the strain gauge signal. Similarly, in case of the non-crack, the highest saliency is placed at the end of the signal, however, a high saliency value is also placed at the very beginning.

Figure 6.7 shows the same strokes as Figure 6.6, but for the prediction task, i.e., the strain gauge signals are cut before the cracks occur. In case of the crack, the saliency increases with an increase of the strain gauge signal and reaches its highest value, when the strain gauge signal reaches its highest slope. This observation is consistent across all cracks for the prediction case and indicates, that the most important time window to predict cracks is the time window, in which the strain gauge signal increases the most. Similar to the recognition case, in case of the non-crack, the saliency decreases with the increase of the strain gauge signal, indicating, that the early stage of the signal is important to predict non-cracks.

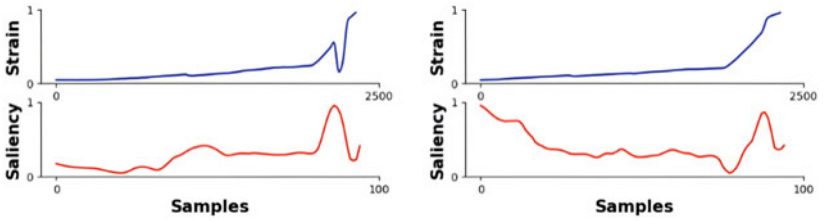


Figure 6.6 Two exemplary strokes and their corresponding saliency series. blue curves show the strain gauge signals for a crack (left) and a non-crack (right) and red curves the saliency series. Taken from [206]

Extraction of Relevant Time Series Motifs

After the identification of the important time windows for the networks' decision making via Grad-CAM, the question arises what characteristic time series motifs are hinting towards a crack. As Grad-CAM does not provide any information about what the network is looking for in these time windows, i.e., what characteristic time series motifs are important for the network's decision-making, they were subsequently extracted from the network. To this end, an ablation study was performed to remove individual trained filters from the network one-by-one with a subsequent checked the networks' performances on the test sets each time a

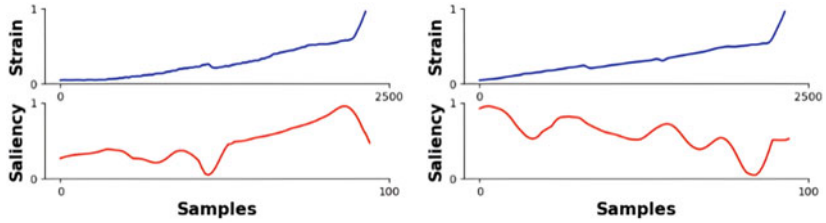


Figure 6.7 Same strokes as in Figure 6.6, but for the prediction task. Taken from [206]

filter was removed. Specifically, ablations were performed in each layer individually removing the filters in random order until the whole layer was removed, essentially reducing the network performance to 0%. These ablations were done 1,000 times and evaluated via the average effect on the networks' performance to account for effects that may be caused by specific ablation orders. The importance of a filter was based on the average decrease of the network's accuracy as a result of the performed ablation. Thus, the most important filters were identified to be the ones, which cause the strongest impairment of the networks' classification performance upon removal.

Figure 6.8 and Figure 6.9 show the accuracy differences averaged over 1,000 ablations of the 16 individual filters in the first layer of the recognition and prediction network, respectively. Consistent with the result of the previous studies, some filters do not seem to play an important role in the recognition task, as the difference in accuracy is negligible, e.g., for filters 4, and 13, or not even apparent at all, e.g., for filter 6. On the other hand, some filters show a strong difference in accuracy, e.g., filters 3, 5, 9, 10 and 11. This indicates that specific filters have emerged during training that are important for the recognition task. Similarly, in case of the prediction task, some filters are more important than others, however, unlike in the recognition task, two filters, i.e., 9 and 13, seem to be harmful for the correct prediction of cracks. On average, the network's accuracy increased after ablating these two filters, indicating, that the optimization process of the network's weights during training reached a local minimum and can be further improved. Furthermore, the averaged accuracy differences for the prediction network are generally larger than for the recognition network. This may be due to the prediction network relying on single filters more than the recognition network. Alternatively, the learned representation of the recognition network may be more robust than the learned representation of the prediction network due to redundant representations of individual features. Such redundancy is expressed by the fact,

that different filters represent similar features, i.e., time series motifs, so that the ablation of these filters does not necessarily impair the network's performance critically.

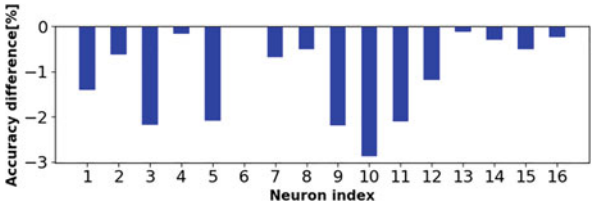


Figure 6.8 Accuracy differences upon ablations of single network filters in random order averaged across 1,000 trials for the **recognition** task. Taken from [206]

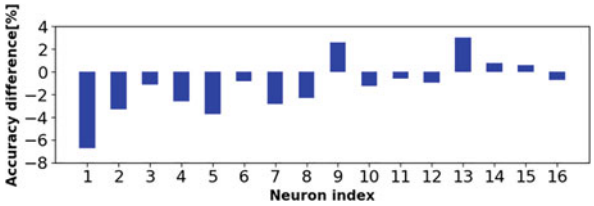


Figure 6.9 Accuracy differences upon ablations of single network filters in random order averaged across 1,000 trials for the **prediction** task. Taken from [206]

Figure 6.10 shows the seven most important time series motifs extracted from their corresponding filters of the first layer of the recognition network, from left to right and top to bottom, ranked by the average accuracy difference upon their ablation. All seven motifs share the same basic curve shape: a minor decrease at the beginning followed by a steep increase up to its peak value towards the end. Interestingly, this shape closely resembles the shape of a crack and is in accordance with what human domain experts would be searching for, when inspecting the time series data. Figure 6.11 shows two exemplary strokes containing cracks and a zoom in on temporal course of the sensor signal of the crack. Both zoom ins resemble the general shape that is found in the trained filters shown in Figure 6.10.

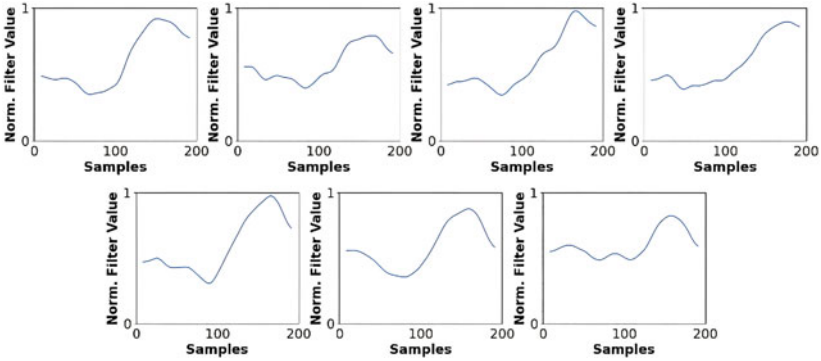


Figure 6.10 Top seven most important time series motifs extracted from the first layer of the recognition network in descending order from left to right, top to bottom. Taken from [206]

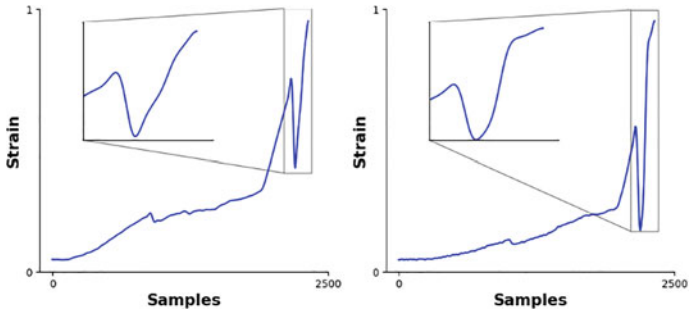


Figure 6.11 Examples of strokes containing most important time series motifs. Taken from [206]

Similar to Figure 6.10, Figure 6.12 shows the eight extracted most important time series motifs from their corresponding filters of the first layer of the prediction network, from left to right and top to bottom, ranked by the average accuracy difference upon their ablation. Unlike the recognition network, the prediction network has learned three distinct groups of curve shapes that seem to be important to predict cracks: 1) the three shapes highlighted in red show a decreasing trend of the signal with a slight decline, much smaller than the strong and sudden decline in case of a crack. These subtle declines of the strain gauge signal indicate the

formation of micro cracks in the metal sheet, causing the strain gauge signal to decrease, as the metal sheet does not transfer the mechanical force exerted by the punch onto the die. 2) The two shapes highlighted in green show strong fluctuations of the signal resembling a landscape with distinct peaks and valleys. These fluctuations are indicative for mechanical oscillations that emerge and build up in the tool during manufacturing. 3) The three shapes highlighted in blue are more similar to the shapes learned by the recognition network, whereas the one in the bottom right seems to constitute a transition to the first group of shapes (red).

The higher variance across the most important time series motifs of the prediction network in comparison with the extracted time series motifs of the recognition network suggest that the prediction task is more complex than the recognition task and requires the representation of a larger number of distinct features, i.e., different time series motifs. This is consistent with the intuition and expectation of the human domain experts, as the post-hoc identification of cracks in the time series data is much easier than a well-grounded estimation of what lead up to these cracks at the beginning of the stroke.

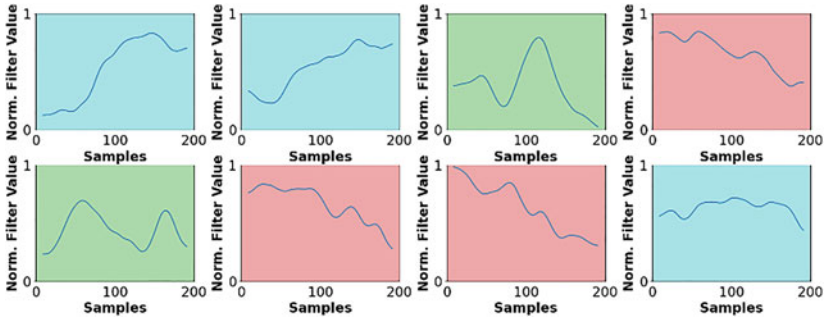


Figure 6.12 Top eight extracted most important time series motifs from the first layer of the prediction network in descending order from left to right, top to bottom. The three different filter categories are highlighted in blue, green and red. Taken from [206]

6.1.4 Summary and Contribution of the Results to the Research Questions

This study addressed the fourth research question and transferred network ablations in combination with a saliency approach to tackle the lack of transparency

of deep learning models for their application for failure prediction in an industrial manufacturing scenario. Specifically, a 1D-CNN was trained to recognize and predict cracks that occur in metal sheets during a deep drawing manufacturing process of car body parts. The most relevant time windows for the network to consider reaching its decision were identified by adapting Grad-CAM for the 1D-CNN architecture and the sensor time series data. Subsequently, the most important time series motifs were extracted from the network via ablations of individual network filters.

In order to recognize cracks in the strain gauge signal data, the trained network learned filters with a shape similar to the actual cracks. These learned shapes are in accordance with what human domain experts would look for in the data to identify cracks post-hoc. Furthermore, the network learned a broader set of different filters with different shapes to predict the occurrence of cracks rather than merely recognizing them. These shapes can be related to different faulty situations in the manufacturing process like the formation of micro cracks in the metal sheet or the emergence of mechanical oscillations in the deep drawing tool.

In regard to the goal of facilitating transparency and interpretability, in general, the extracted time motifs from the recognition network are intuitively understandable for domain experts and resemble shapes, which they would look for in the time series data to post-hoc identify cracks. However, no knowledge about characteristic time series motifs that would hint to the future occurrence of cracks was present before the study. Thus, the extraction of these important time series motifs from the prediction network followed by an investigation through the domain experts and the direct relation to the deep drawing process shed light on possible effects that would cause cracks in the metal sheets. These newly gained insights help to improve the manufacturing process with respect to control for those situations, that can result in cracks in the metal sheets. The integration of the approach into the manufacturing process provides the domain experts with a live prediction of the network and the visualization of the relevant time windows and time series motifs. This provides domain experts in-line insights into the process data a new and deeper level to monitor the integrity of their manufacturing process.

Concludingly, the insights gained in the first transfer study demonstrated the value of network ablations in combination with the more classic saliency approach, based on Grad-CAM, to facilitate transparency and interpretability for the trained network's decision-making process to provide domain experts with a data-driven approach to support predictive quality estimation.

6.2 Transfer Study 2: Attention Mechanisms for Deep Drawing

Overt and covert attention are two mechanisms famously exhibited by the visual system of the mammalian brain making it highly efficient in filtering the vast amounts of visual information that could possibly be processed [210]. The attention helps to focus on what is supposedly important and filter out what is not. Complementary to the first transfer study, which investigated the usability of network ablations to facilitate transparency for the manufacturing process of deep drawing, the second transfer study investigates the analogous attention mechanism of transformer networks that were originally introduced in 2017 to tackle problems in the domain of natural language processing [211] and their usability to gain insights about what sensors of the sensor system are most important to recognize and predict of process failures.

Contrary to the first transfer study, a multitude of sensor time series data acquired simultaneously is considered rather than just a single time series, to fully utilize the network's built-in attention layers and investigate which sensors are most strongly being attended to by the network. As cracks are not consistently measured by one out of the eight sensors or in all eight sensors at the same time but rather by particular sensors depending on the spatial location of the crack in the car body part, the learned attention of the network, which is part of its learned representation, provides insights into what sensors are most important for the prediction of cracks. Given the spatial distribution of the sensors across the deep drawing tool, the investigation aims to provide insights into the location of the predicted occurrence of cracks and allows to possibly identify specific measures to prevent these occurrences. Besides the spatial attention learned by the network, it also learns a temporal attention, providing insights into what parts of the individual sensor time series are important for the formation of cracks.

Further complementing the first transfer study, this study extends the original classification problem to three categories adding the most challenging category of "small cracks", which are similar to clean strokes but contain small decreases in the sensor time series signals that are corresponding to potentially harmful formation of micro cracks in the manufactured sheet metal. In addition to the classification task, the model is also trained to forecast the eight sensor time series to predict the occurrence of cracks before they actualize in the manufacturing process, which allows to intervene in case of the prediction of large cracks that could possibly damage the deep drawing tool.

The study extends the addressing of the fourth research question and expands on the approach of the previous study, i.e., the mere transfer of the established

research methods such as network ablations and activation embeddings. Specifically, analyzing the network's distribution of its attention weights as part of its learned representation, with respect to the importance of individual sensors and points in time for the prediction of process failures relates the network's learned representation to physical features of the manufacturing process. The presented modification of the network architecture originating from [211] demonstrates an exemplary scenario for the facilitation of transparency and interpretability based on a network's learned representation to the particular circumstances of a use case to address these particularities purposefully with the goal to gain new insights about the considered manufacturing process.

6.2.1 Key Contributions of the Study

The first key result addresses the fourth research question and investigated the relationship between the different input sensors and the occurrence of process failures, i.e., the formation of cracks in the metal sheets. Specifically, the learned weights of the spatial attention layer of the network as part its learned representation showed that two out of the eight input sensors turned out to be much more important for the prediction of cracks than the other sensors. From an engineering and domain expert perspective, this insight significantly advances the understanding about the formation of cracks on a spatial scale.

The second key result, too, addresses the fourth research question by analyzing the distribution of the learned weights of the temporal attention layer as part of the network's learned representation and showed that the eight different sensors have individually distinct time periods that are most important for the prediction of process failures. Specifically, the results show that the most important time periods to predict clean strokes are found in the beginning of the deformation process while the most important periods for the prediction of cracks are found towards the end for the deformation process, close to the predicted point in time of the occurrence of cracks.

6.2.2 Methods and Experimental Design

To investigate the relationship between the acquired sensor time series data and the formation of cracks, a dual stage attention neural network was trained to solve two learning tasks simultaneously. The network is trained to classify the sensor time series data into three categories, while forecasting its temporal course for

all three classes. The learned weights of both, the spatial and temporal attention layers are analyzed with respect to their specific characteristics in relation to the classification of the three classes and the prediction of their temporal courses.

Learning Tasks

For the classification task, the sensor time series data was labeled with the same procedure used in the first transfer study (cf. Transfer study 1, sub-chapter "*Data Preprocessing and Dataset Preparation*"), however, three classes were distinguished, i.e., *clean*, *small cracks* and *large cracks*. Figure 6.13 illustrates three example time series for each of the three classes. It further illustrates the forecasting learning task showing what part of the time series is the input to the network and what part is meant to be forecast.

Compared to the first transfer study, each stroke contains eight sensor time series, each of which is labeled independently from each other. Additionally, the stroke containing eight sensor time series is labelled according to the individual time series labels. If all eight time series are *clean*, the stroke receives the overall label *clean*. If at least one out the eight time series contains a *small crack* and the others are *clean*, the stroke receives the overall label *small crack*. If at least one out of the eight time series contain a *large crack*, the stroke receives the overall label *large crack*, independent of whether the other time series are *clean* or *small cracks*. This way, 700 out of the 3,329 strokes are clean, 2,527 contain small cracks and 102 contain large cracks. Considering the sensors individually, the dataset contains 15,763 clean time series, 10,521 small cracks and 348 large cracks.

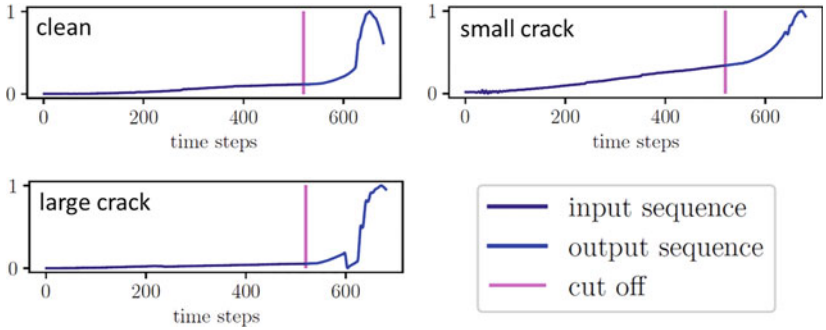


Figure 6.13 Three examples of the sensor time series for the three categories clean, small crack and large crack

Data Preprocessing and Dataset Preparation

The preprocessing of the data is done the same way as in the first transfer study (cf. transfer study 1, "*Data Preprocessing and Dataset Preparation")* with slight differences to account for the different network architecture and the different requirements for the data to achieve a sufficiently good classification and prediction performance. In this study, the time series were interpolated so that they all have the same length of 6.800 and subsequently down sampled with a ration of 1: 10, i.e., the resulting time series have a length of 680 points. This temporal resolution still contains enough information to detect the cracks, i.e., the temporal resolution is still high enough to resolve the temporal course of cracks allowing the network to learn to recognize its temporal course but reduced the computational effort to process the data compared to the original temporal resolution. Similar to the first transfer study, the time series are cut off and separated into model input and model target for the forecasting task. The input $\boldsymbol{X}$ and the target $\boldsymbol{Y'}$ for each stroke s is given by

$$\boldsymbol{X}_s = (\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}, \ldots, \boldsymbol{x}_{s,T_x})$$

and

$$\boldsymbol{Y}_s^{'} = (\boldsymbol{y}_{s,1}^{'}, \boldsymbol{y}_{s,2}^{'}, \ldots, \boldsymbol{y}_{s,T_y}^{'})$$

where $T_x = 520$ and $T_y = 160$. Both, input, and target as well as the individual vectors $\boldsymbol{x}_{s,i}$ and $\boldsymbol{y}_{s,i}^{'}$ are eight-dimensional and contain the sensor data of the eight strain gauge sensors.

For model training, the dataset was divided into a training set, a test set and a validation set with the ration of 60: 20: 20. Thus, the training set contains 1,996 strokes and the test and validation set contain 662 strokes. The data was split so that each of the three splits contains the same ratio of the three classes, which is 7: 25: 1 for the categories *clean*, *small cracks* and *large cracks*. Thus, the data sets are highly imbalanced, which is typically challenging for learning models, as they might learn to favor the characteristics of strokes encountered most frequently during training. In this use case, this would especially impact the prediction of large cracks, which constitute the rarest occurrence in the data. However, the prediction of large cracks is essential to the learning problem to detect failures in the deep drawing processes before they actualize. In order to mitigate the problem somewhat, the training and validation data set is augmented such that each of the three label categories contains the same number of strokes

simply via duplicating strokes labeled as *clean* and *large crack*. This results in a training set containing 4,623 strokes and a test and validation set containing 1,479 strokes.

Model Design, Training, and Evaluation

The neural network architecture employed is based on the dual stage attention recurrent neural network (DA-RNN) architecture proposed in [212]. Specifically, the combination of two attention levels, the spatial and the temporal level, is most fitting for the deep drawing use case that provides multimodal sensor time series data, i.e., data which supposedly differs in importance with respect to its temporal course and with respect to the spatial localization of the individual sensors. Compared to the original DA-RNN, a number of modifications have been made to address the peculiarities of the deep drawing use case. The original DA-RNN architecture was designed to process multiple driving time series and to predict the next successive data point of a single time series based on the processed information of the driving time series and the time course of the time series to be forecast.

Figure 6.14 shows a schematic illustration of the data processing and prediction procedure of the original DA-RNN (lefthand side) as well as the used model in this study (righthand side). Compared to the original DA-RNN, the modified learning model predicts several points creating a whole temporal course of the forecast and is able to do so for all eight sensors simultaneously.

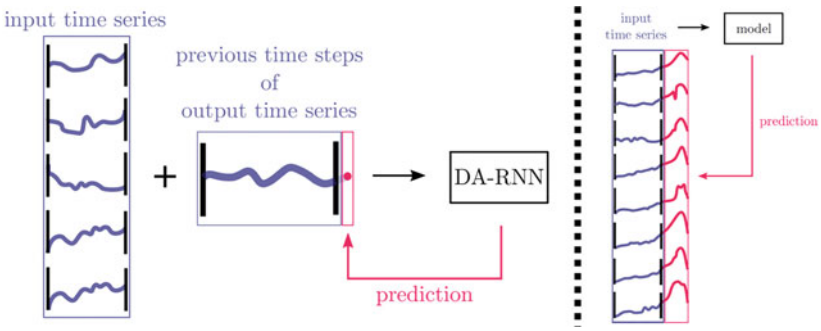


Figure 6.14 Left: Schematic illustration of the data processing and prediction procedure of the original DA-RNN. Right Schematic illustration of the data processing and prediction procedure of the modified model

The model architecture is illustrated in Figure 6.15. The time series of the eight sensors are fed into the two attention layers that compute the spatial attention across the eight sensors, i.e., the sensor attention, and the temporal attention. Both distributions are combined and fed into two fully connected layers that end in two heads, one to predict the time series forecast and one to predict the labels of the sensor time series. Combining both learning tasks into a single architecture forces the model to recognize the class of the time series that are forecast and makes it supposedly easier for the model to distinguish between the forecasts of a *clean* stroke compared to a *large crack*. The classification head uses the cross-entropy loss while the regression head uses the soft-DTW loss [213], a differentiable version of dynamic time warping, which is a metric to calculate the similarity between time series that considers the shape of the temporal course for the forecast, which would not be adequately considered by the MSE loss. For both heads and both loss functions, the strokes are weighted according to their class label to account for the increased importance to recognize large cracks despite their under representation in the datasets. Specifically, for the regression head using the soft-DTW loss, the weighting factors for each class are chosen according to the number of strokes available. Strokes that contain only clean sequences are weighted with a factor of 5, strokes that contain small cracks are weighted with a factor of 1 and sequences with big cracks are weighted with a factor of 15. For the classification head using the cross-entropy loss, the weighting factors are chosen according to the importance to recognize the individual classes. Specifically, it is more important to recognize large cracks than small cracks and clean strokes as they are the most dangerous for the manufacturing process. In the same way, it is more important to recognize small cracks than clean strokes. Thus, the weightings factors for the classification task were experimentally determined to be 1, 5 and 10 for the *clean* strokes, *small cracks,* and *large cracks*, respectively to yield a good performance. The model was trained on the balanced training dataset for 1.000 epochs with a batch size of 32 using the Adam optimizer. Table 6.3 shows the confusion matrix for the model's classification result on the validation dataset. The model reaches an overall accuracy of 84.60%. The precision for the three classes is 94.3%, 74.4%, and 70.0% for the labels *clean*, *small crack*, and *large crack*, respectively.

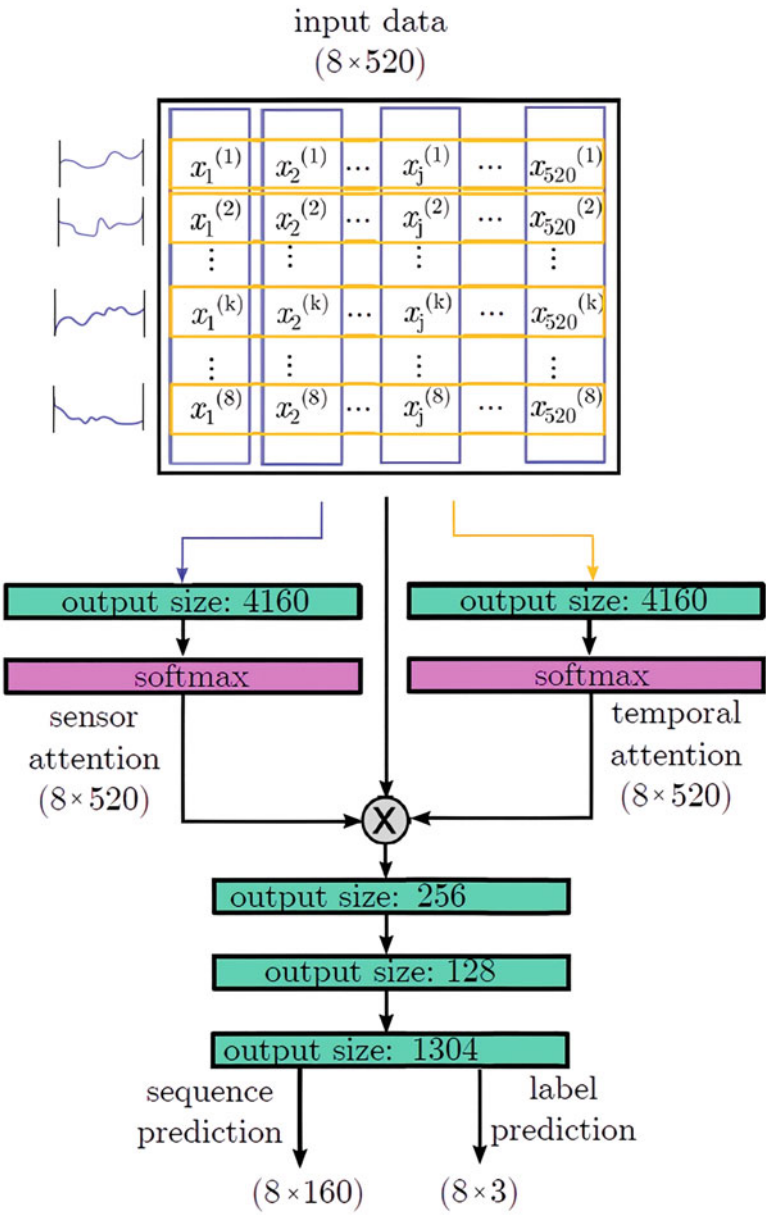


Figure 6.15 Schematic illustration of the architecture of the learning model

Table 6.3 Confusion matrix of the classification results of the learning model on the validation dataset

		ground truth		
		clean	*small crack*	*large crack*
predicted label	*clean*	2523	132	20
	small crack	623	1869	17
	large crack	9	3	28

Figure 6.16 shows two exemplary strokes containing time series of all three classes and the corresponding classification and forecasting results of the model. The model successfully forecasts the temporal courses of the sensor time series for all three classes. For reasons of brevity, only exemplary strokes are shown in this thesis. However, the trained model is available to use in an interactive web-based live demonstration [214] to investigate the different strokes of the three different classes In depth. The model can be used for the forecasting task on all strokes of all different classes to compare the forecast of different sensors within an individual stroke or to compare the forecast of different strokes to each other.

6.2.3 Results

To address the fourth research question and utilize the model's learned representation to facilitate interpretability and transparency for the deep drawing manufacturing process, the learned attention distributions are investigated with respect to their relation to the individual sensors and their time courses. In general, the interpretability of learned attention weights is not straight forward and their use to facilitate explainability is still a matter of current research [215–219]. Oftentimes though, a comparison of model learned attention with human attention is chosen as an approach to evaluate the interpretability of a model's learned attention [220–222], which, however, is an approach inapplicable to the deep drawing use case due to the absence of human attention and the lack of comparability. Thus, the model's learned attention distributions are analyzed with respect to different characteristics of the data, i.e., the learned weights of the *sensor attention* and the *temporal attention* layers, which are multiplied with the input data and thus function as a distribution of importance over the inputs, are investigated. The learned distributions are assumed to display the network's learned ability to attend to important points in time and to distinguish between important

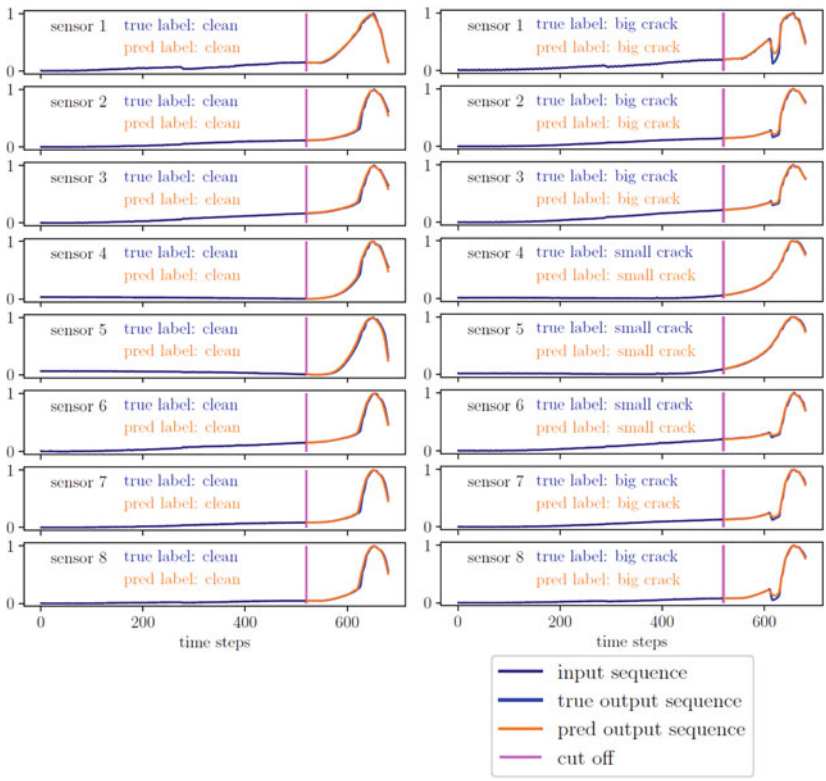


Figure 6.16 Two example strokes from the validation set with their respective sensor time series and the prediction of the model

sensors for the classification of the strokes as well as the forecast of the time series data in a phenomenologically similar way as the aforementioned overt and covert attention mechanisms found in the visual system [210].

Figure 6.17 shows two exemplary strokes and the corresponding attention distributions produced by the model during the forecast of the time series for the three different classes. The distributions are visualized as yellow and red histograms for the sensor attention and the temporal attention, respectively, with a bin width of five timesteps. Note that both attention distributions are computed along different dimensions. Specifically, the temporal attention is distributed across the 520 times steps of the time scale on which the time series data is

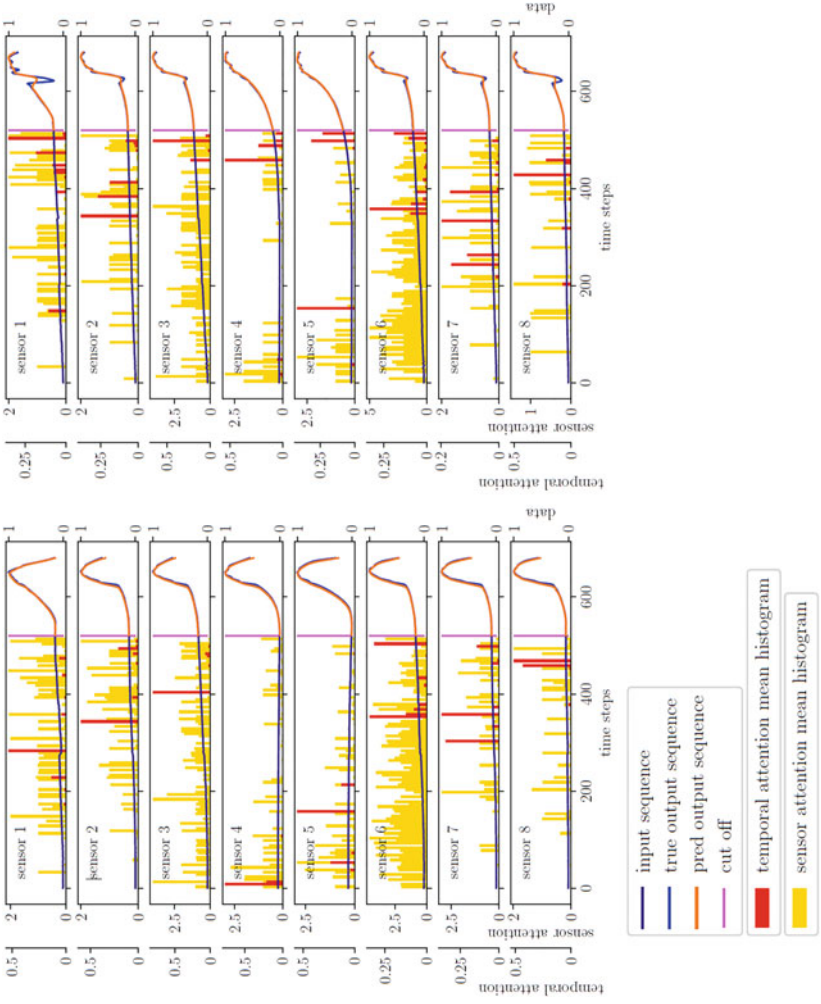


Figure 6.17 Two exemplary strokes and their corresponding eight sensor time series, the model forecast, and the corresponding learned attention distributions visualized as a histogram

plotted, while the sensor attention is distributed across the eight sensors for each time step. Regarding the sensor attention, this allows to investigate how the importance of a sensor compares to other sensors and how it changes over time. Regarding the temporal attention, this allows to investigate when a sensor signal is most important over its complete time course. The distributions of both attentions do not differ noticeably for the three different classes. Interestingly though, the sensor attention shows the strongest emphasis for sensor six, which has a higher density of high attention weights across time compared to other sensors which show a much sparser distribution of high weights. In this case, weights are considered high relative to the rest of the weights within the distribution and constitute obvious peaks in the distributions. The temporal attention distributions show some noticeable distinctions for the different classes. Specifically, the temporal attention of the fourth sensor is placed only at the very beginning of the time series for the *clean* stroke (cf. Figure 6.17, lefthand side) while it is placed at the very end of the sensor's time series for the *small crack* (cf. Figure 6.17, righthand side). In general, the temporal attention for is typically shifted towards the end of the input time series as well.

Figure 6.18 shows the mean temporal attention distributions for the three classes *clean*, *small crack* and *large crack*, which average over the individual temporal attention distributions of all strokes and all sensors within the respective class, to compare their universal characteristics to each other. The most obvious difference is that the distribution for the large cracks (red) is more dominant towards the end of the input time series and completely absent at the beginning. This observation makes sense in light of the nature of the deep drawing process. Near the end of the deep drawing process, the punch comes into contact with the metal sheet placed on the sheet holder exerting force onto it, which can potentially shift the position of the sheet so that it snaps out of place, which subsequently leads to the formation of cracks. Thus, the model's learned attention seems to be able to reflect that relation and provides interpretability for the model's decision-making process and provides insights about the important parts of the sensor data for that decision. The other two distributions are less pronounced towards the end of the input time series but exhibit their own individual characteristics. The average temporal attention distribution for small cracks (blue) shows a distinct peak at around the 150^{th} time step while the distribution for clean strokes (yellow) exhibits a distinct peak at the very beginning of the time series. The general observation to be made is that the attention distribution shifts from the beginning to the end of the input time series when cracks form in the metal sheet. Loosely phrased, the larger the crack, the more prominent the attention towards the end of the time series.

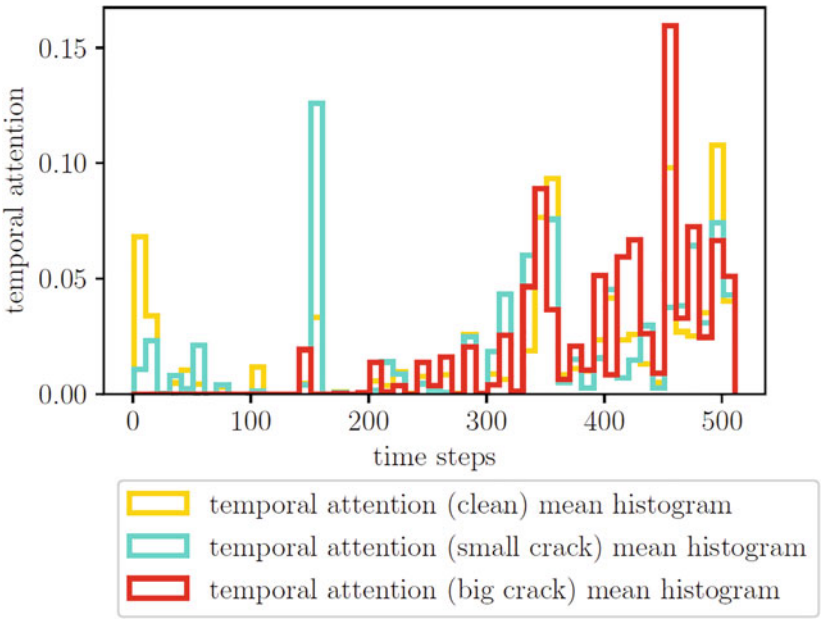


Figure 6.18 Comparison of the temporal attention distributions for the three classes clean, small crack and large crack

In order to support the qualitatively described differences between the three distributions with a quantitative measure, the standardized Euclidean distance, a special case of the Mahalanobis distance [223], between them is calculated. For pairwise comparisons of the three distributions, under the null hypothesis that two average attention distributions follow the same underlying distribution, the squared standardized Euclidean distance follows a distribution that can be approximated by a chi-square distribution [224]. Formally phrased: let a_1 and a_2 be vectors containing the values of two mean attention distributions, each with length T_x. Then, the squared standardized Euclidean distance D^2 between the two vectors is computed as

$$D^2 = \sum_{i=1}^{T_x} \frac{(a_{1,i} - a_{2,i})^2}{s_{1,i}^2 + s_{2,i}^2}$$

with

$$s_{1,i} = \frac{\sigma_{1,i}}{\sqrt{n_1}} \text{ and } s_{2,i} = \frac{\sigma_{2,i}}{\sqrt{n_2}}$$

where $\sigma_{1,i}$ and $\sigma_{2,i}$ are the standard deviations for the i-th element of the first and second mean attention vector, respectively, while n_1 and n_2 denote the total number of attention distributions in each set. This distance is comparable to the test statistic used in [225] for the comparison of two weighted histograms. To estimate the statistical significance of the difference between the distributions, the p-value is computed from the squared standardized Euclidean distance D^2. For all three pairwise comparisons, i.e., clean vs. small cracks, clean vs. large cracks, and small cracks vs. large cracks, the p-value is <0.1. This implies that the averaged attention distributions of the groups do not follow the same underlying distribution, and thus, are significantly different from one another.

Figure 6.19 shows the averages of all attention distributions regarding each sensor as well as the average input sequence data and its standard deviation. The average sensor attention distribution is similar to the distributions for individual strokes, of which two exemplary and representative ones are shown in Figure 6.17. Thus, the overall averaged sensor attention only varies minimally regarding different strokes. Furthermore, Figure 6.19 shows that the sequences of the third and sixth sensors are generally weighted with a higher sensor attention than the other sensors' sequences. Accumulating the averaged sensor attention weights of each sensor underlines that observation quantitatively. The accumulated average sensor attention of the third and sixth sensors amounts to 101.4 and 222.5, respectively, while that of the first, second, fourth, fifth, seventh, and eighth sensors amounts to 46.5, 34.2, 30.1, 31.1, 35.6, and 18.6, respectively. This observation is visualized in Figure 6.20.

Specifically, Figure 6.20 shows the average sensor attention distribution of each sensor and each category individually. The sensor attention is distributed almost equally for all three categories. This shows that independently from the characteristics of the stroke, the model establishes a similar sensor attention. It suggests that the model found that during training the most relevant information regarding the shape of the output sequences can be found in the third and sixth sensor. Additionally, the fourth and the fifth sensors are overall found to

be weighted less by the model, especially regarding the time window during the middle of the input time series. Interestingly, the corresponding average temporal course of input time series of the fourth and fifth sensor differs from those of the other sensors. They exhibit a comparatively very flat course up until the end of the time series, which is characterized by a comparatively strong increase of the sensor signal. This observation is quantitatively supported by calculating the average standardized Euclidean distance between the sequences of the fourth or fifth sensor and the sequences of the other sensors, which amounts to $D^2 = 2932942.1$ and is about four times higher than the average standardized Euclidean distance of $D^2 = 726187.8$ between all other sensors' input sequences. Additionally, the temporal attention is rather prominent at the beginning of the time series for the fourth and fifth sensors compared to the other sensors, which prominently show the temporal distribution peaks towards the end of the time series. This indicates that the model learned to compute a differently distributed attention to the fourth and fifth sensors' input time series. The visualization of the attention distributions in Figure 6.19 is extended by Figure ESM16, Figure ESM17, and Figure ESM18 in the electronic supplementary material, which display the average attention distributions and average input sequences for each sensor for all the time series exclusive to one of the three categories *clean*, *small crack*, and *large crack*, respectively. Figure A18 shows that the dataset does not possess strokes that include sequences with big cracks for the fourth or fifth sensor. This explains that the model learned to put less emphasis on the fourth and fifth sensors' input time series for the overall prediction of stroke labels and forecasts, which could be connected to the fact that no information for big cracks is contained in these input time series. This insight provides another perspective for the interpretation of the average temporal attention distribution illustrated in Figure 6.18. Specifically, none of the temporal attention distributions that belong to the fourth or fifth sensor are part of the computation of the average temporal attention distribution regarding large cracks. This might explain why the temporal attention distribution corresponding to data with large cracks is shifted more towards the end of the input time series, as only the fourth and fifth sensor generally exhibit peaks in the temporal attention distribution that are located rather at the beginning of the input sequence.

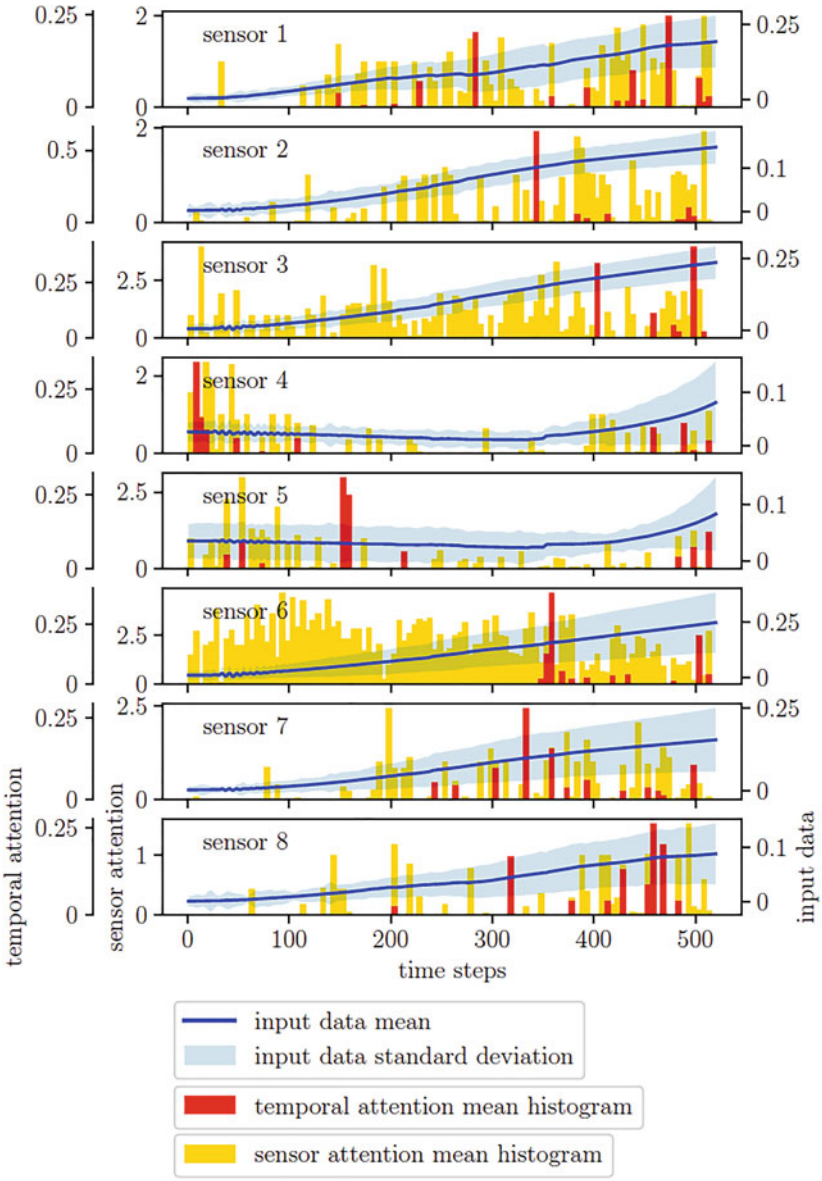


Figure 6.19 Mean sensor and temporal attention distributions averaged across all strokes regarding each sensor as well as the mean input time series data and its standard deviation

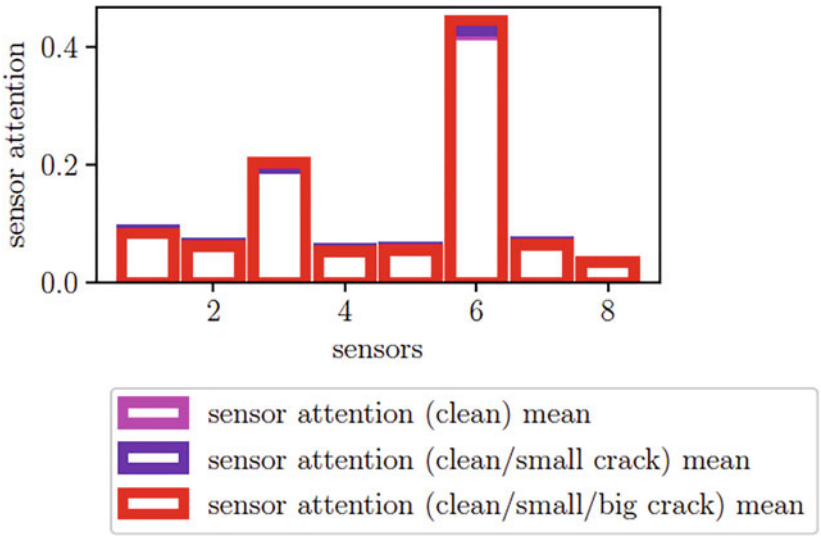


Figure 6.20 Mean sensor attention distribution across the eight sensors for each category individually

6.2.4 Summary and Contribution of the Results to the Research Questions

The second transfer study investigated the neuroscience inspired mechanism of overt and covert attention and the applicability of statistical analysis of its analogous attention mechanism originally introduced in transformer networks. The transfer lies in the analysis of the distributions of the learned weights as part of the network's learned representation to facilitate the transparency and interpretability for the industrial manufacturing process of deep drawing. To this end, the two weight distributions of the network's spatial attention layer as well as its temporal attention layer were investigated in relation to the learning task, i.e., to classify the sensor input data and predict its temporal course. The acquired deep drawing sensor time series data was pre-processed and categorized into three classes, i.e., *clean*, *small crack*, and *large crack*. The dataset is characteristically highly unbalanced for a manufacturing process containing only 348 *large cracks* compared to 15, 763 *clean* sensor time series, making the dataset challenging for learning models.

The model was trained on an augmented dataset with the same number of strokes for each category. As loss function, weighted version of the soft-DTW loss and the cross-entropy loss were applied for the respective regression and classification head of the model. The classification performance of the model reached an overall accuracy of 84.6%, and a precision of 94.3%, 74.4% and 70% for the classes *clean*, *small crack*, and *large crack*, respectively. The qualitative evaluation revealed that the model is capable of forecasting *small cracks* and *large cracks*, though qualitatively the prediction of *clean* time series was found to be significantly better.

The analysis of the weight distributions of both attention layers revealed that the input time series of the third and sixth sensor are weighted significantly higher than the other sensors. Furthermore, the learned weights of the temporal attention layer showed distinct distributions for all sensors that putting increased emphasis on different points in time for each sensor. In relation to the fourth research question, these results demonstrate the feasibility to investigate the learned representation of a trained model to gain insights about the real-world manufacturing process.

Critical Reflection & Outlook

7

"Accept both ***compliments and criticism****. It takes both* ***sun and rain*** *for a flower to grow."*

– English language adage

The following chapter summarizes the results of the conducted studies with respect to their contributions to the four research questions and puts them into perspective of the overall goal of the thesis, to investigate the feasibility of neuroscientific inspired approaches to facilitate transparency and interpretability for the decision-making of artificial neural networks. The first sub-chapter critically reflects on the **achievements and limits** of the results leading to the second sub-chapter, which addresses the limits, phrases open questions and discusses possible research efforts to answer these open questions in the future.

7.1 Reflection of Results & Contribution to Research Questions

This sub-chapter picks up the four research questions separately and relates the results of the conducted studies to them accordingly. The collected consideration of the results gives the answers to these four questions that were elaborated within the framework of this thesis. It further critically discusses the limits of the study results and the resulting gaps in answering the research questions.

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_7

7.1.1 Research Question 1

How to determine the importance of individual neurons and groups of neurons for a network's learned ability to solve a specific task?

The first research question was addressed in the first, second and third research study. In general, the results of the studies showed that neuroscience inspired approach of network ablations is a potent tool to determine the importance of single neurons and groups of neurons for the learned task of the network. Specifically, the first research study showed how network ablations were used to categorize neurons into three distinct roles with different meaning for the learned classification task. These roles include neurons that are globally important for task, i.e., for all tasks, neurons that are selectively important for specific classes, and neurons that are not important at all. It further revealed that the importance of a neuron, i.e., the severity of its ablation with respect to the impact on the network's performance, can be determined by the degree of change of the distribution of the neuron's incoming weights during training. Extending the ablation approach from individual neurons to pairwise neuron ablations revealed that some neurons are not unique in the role they play for the learned task but that their learned features are represented redundantly within the network by other neurons.

The second study corroborated the results and demonstrated the selectivity of neurons, more specifically filters in a CNN, which are small groups of single neurons, for individual classes. Thus, it bridges the gap between investigating the role of individual neurons and groups of neurons that have similar roles for the learned task. However, in case of filters of a CNN, the grouping is predetermined by the kernel size and does not account for any individual characteristic of the neurons within that group. The results further showed that the role of trained neurons is flexible and can be quickly changed by additional training, if required, for example in the case of network ablations causing severe impact on the network's performance due to the removal of important neurons.

The third study addressed this issue and investigated groups of individual neurons with no architectural grouping involved. Besides utilizing networks ablations, the importance of neurons was determined by two activation-based metrics, the magnitude, and the class selectivity. A comparison of the three metrics showed that they don't necessarily yield the same results with respect to determining the importance of neurons. Thus, it remains unclear how to determine this importance conclusively. Ablations are arguably the most intuitive way to determine a neurons importance, however, it must be considered that removing an individual neuron affects the network not only by removing what the neuron learned to

represent but also by changing the flow of information through the network, as several subsequent neurons receive an altered input. Thus, the ablation of a single neuron possibly affects the network to an extent that has not been investigated within the framework of this study. Hypothetically, removing a neuron in an early network layer influences all subsequent neurons, which receive an altered input. Subsequently, the altered inputs of these individual neuron neurons result in an altered output which again influences all the neurons in the subsequent layer. This effect not only repeats throughout all layers of the network but may intensify in deeper layers given the connectivity of network and the added effects of altered inputs and outputs towards the end of the network.

7.1.2 Research Question 2

How to characterize the structure and organization of a neural network's learned representation qualitatively and quantitatively?

The second research question was addressed in the second, third, fourth and fifth research study. In general, the results of the studies showed that locally targeted network ablations, i.e., ablations of neurons in specifically targeted locations of the network, like individual layers, are a suitable method to determine the importance of these layers for the network's learned task.

The second research study demonstrated the strong variance across network layers with respect to their importance for the network's learned task via spatially targeted network ablation within the individual layers. Furthermore, the results revealed that these differences between layers vary across classes suggesting that different classes are represented in different network layers. Thus, the learned representation is organized and spread across network layers universally, i.e., for all classes, and specifically, i.e., for individual classes. It remains unclear, which of those two effects is dominating in structuring the learned representation with respect to the distribution across layers and was not further investigated within the framework of this thesis. Furthermore, the observation was not investigated with respect to the dependence of the number of classes and the complexity of the learning task. Additionally, the results were not investigated regarding to their transfer to other state-of-the-art networks and remain specific for the network investigated, the VGG-19. The fourth research study corroborates the results and showed that the two different layers of the investigated actor network are differently affected by network ablations. Specifically, the first layer was shown to be much more robust against network ablations, which suggests that the learned

representation was more redundantly stored in the first layer as compared to the second layer. However, similar to the second research study, this result was neither investigated for different learning tasks nor was it transferred to other network architectures. Thus, the result remains specific to the investigated network and the chosen learning task.

Besides targeted network ablations, embedding methods like PCA and UMAP were shown to provide suitable approaches to define structure of a network's learned representation. Specifically, the embedding methods define the similarity between individual neurons or the similarity between layer or network activations for different states. This similarity allows to structure the individual components, i.e., individual neurons or network activation states, and relate an arbitrary measure to this given structure, for example the importance of the individual components for the correct classification of a specific class. The third research study utilized UMAP embeddings of the activations of individual neurons within the individual layers of the network to investigate the evolvement of the learned representation along the layers of the network. The approach showed that the learned representation becomes more distinct towards the deeper layers as the layer activations become more distinct for the individual classes. Network ablations were used to validate the relevance of this distinction and showed that the degree to which the learned representation is distinct with respect to the separability of the individual classes diminishes as a result of ablations. Thus, network ablations in combination with the embedding of network layer activations were demonstrated to provide valuable insights into the interplay of the individual layers with respect to learning task. The evolvement of the learned representation along the layers of a network is a phenomenon, which is also found in the visual cortex of the mammalian brain. Specifically, the visual system is organized in hierarchical layers which are known to represent distinct features of external visual stimuli, such as bars and contours that are represented in the earlier layers while concepts of whole objects are represented in later layers.

The third research study further showed that the learned representation of the network is organized in relation to the difficulty to predict the individual classes. Specifically, the number of most important neurons within the network varies strongly across the different classes suggesting that the learned representation is organized in relation to the capacity of the network required to represent the necessary features of the individual classes. Although this result was shown for three different datasets, and thus, bears some universality to it, it remains unclear whether the number of neurons determined most important for a specific class reflects the network's used capacity at all. This issue is further reinforced by the previously discussed circumstance that the three used metrics to determine

a neuron's importance for the network's learned task were not consistent across all three datasets. These further challenges any conclusion derived from the high variance of the number of most important neurons in relation to the individual classes as a definitive way to determine the most important neurons could not be established to begin with.

The fifth research study corroborates the results of the previous studies utilizing activation embeddings to reveal an organization of neurons and groups of neurons of the trained agent's actor network in relation to the executed actions. Specifically, the embeddings revealed that distinct groups of neurons are involved for specific movements of the body controlled by the agent.

7.1.3 Research Question 3

How to determine the relation between the structure and organization of a neural network's learned representation and its emerging behavior?

The third research question was addressed in the fourth and fifth research study. In general, the results of the studies showed that the structure and organization of the networks' learned representations can be related to their emerging behavior. Specifically, the activity of individual neurons and the organized activity jointly activated neurons working together in functional groups is correlated to the taken actions of the trained agent and its emerging behavior. This general phenomenon was consistently observed for a number of different DRL agents trained to perform a variety of motor control tasks and thus, demonstrated to be universal across network architectures and environments.

The fourth research study demonstrated that a specific organization of the activity of neurons within the first layer of the actor network could be related to the agent's capability to achieve a high reward in the chosen learning environment. Specifically, that pattern was established via computing the correlation coefficients of the individual neurons' activations and the agent's chosen actions. The relevance of that particular pattern was shown when this pattern was changed and distorted as a result of network ablations. Specifically, the more this pattern changed as a result of network ablations, the stronger the impact on the agent's capability to achieve a high episodic reward. This result, however, hinges strongly on the definition of the pattern via computing the correlation coefficients between neuron activations and the agent's performed actions. No alternative method was chosen to define a pattern and it remains unclear whether the chosen methods is best for the pursued research goal and to what extent the results are specific to that

method. In addition to the relation of activation patterns to the agent's actions, the UMAP embedding of the network's layer activations for each time step during an episode showed a particular organization which could be directly related to the different phases of the learned control policy that achieves a high episodic reward. Network ablations showed how this particular organization was distorted, which subsequently influences the agent's behavior and prevented it from achieving a high episodic reward. This observation is consistent with the previous result and further solidifies the result that the organized activity within the network relates to its emerging behavior. The results of the fourth research study are limited to a single agent trained to perform three similar control tasks, however, the results previously discussed addressing the third research question were only shown for a single exemplary learning environment, the cart-pole swing-up task. Thus, the presented results are possibly specific to the network architecture of the agent, or the chosen learning task and their universality remains uninvestigated.

The fifth research study addresses this lack of universality and investigated the relation between organized network activity and emerging agent behavior for a number of different network architectures and different learning environments. In order to make the relation to agent behavior across different environments with different learning objectives possible, the study defined behavior based on the agent's observation space and demonstrated the emergence of behavioral anchors in the observation spaces of the respective learning environments. The emergence of these anchors was consistently observed across different network architectures and relates to particular poses of the agents' controlled bodies. Furthermore, the organized activity of individual neurons in the UMAP embedded layer activation space was shown to relate to specific behavioral anchors. Thus, the cyclic behavior of the agents was observed to be the result of particular neurons and groups of neurons being jointly activated repeatedly throughout the episodes. The results of the fifth research study largely hinge on the definition of behavioral anchors and the method they are extracted from the observation space. Specifically, the clustering method of choice strongly influences the resulting behavioral anchors and is subject to the experimenters choice. In this study, a simple k-Means clustering approach with a fixed number of $k = 100$ clusters was chosen, which results in 100 distinct behavioral anchors. However, the true number of behavioral anchors is not known, and it remains unclear how different numbers of clusters or different clustering algorithms influence the resulting behavioral anchors.

Although two different approaches to determine a neuron's importance for the formation of specific behavioral anchors were investigated that yielded similar results according to the correlation of the resulting rankings with each other,

one of the methods, the feature importance based on the gini impurity, is potentially somewhat problematic and may obscure the resulting importance ranking. Specifically, the gini impurity may mask the importance of some neurons whose activations are highly correlated with other important neurons. For example, if the activations of neuron A led to the best separation of the classes, neuron B, whose activations are highly correlated to neuron A, would have a low value for the gini impurity, as it does not help to separate the classes further. Thus, the sequential nature of choosing the most important features via the gini impurity may yield vastly different assessments of neurons' importance despite their similar activations. Particularly, it yields a smaller number of most important neurons due to the hierarchically discarding nature of the method. Thus, while consistent with the importance ranking yielded by activation selectivity, it remains unclear how much information is lost using the gini impurity to determine a neuron importance.

7.1.4 Research Question 4

How to utilize the investigation of structured and organized learned representations to facilitate transparency in industrial, data-driven real-world processes?

The fourth research question was addressed in the first and second transfer study. In general, the results of both studies showed that the investigation of the networks' learned representations in relation to the intricacies of the deep drawing manufacturing process yielded interpretable insights with respect to the prediction of process failures. Specifically, the results of the studies demonstrated the use of network ablations to determine important time series motifs in the sensor data that hint to the formation of cracks in the manufactured metal sheets. Furthermore, the analysis of attention weights provided insights about the different sensors spatially distributed across the deep drawing tool with respect to their importance for the formation of cracks.

The first transfer study transferred the saliency approach of Grad-CAM to the one-dimensional sensor data of the deep drawing process and identified the most important time windows for the trained network to classify the sensor data and predict the occurrence of cracks. Network ablations were shown to provide insights about what particular time series motifs are the most important for the trained network to predict the occurrence of cracks. Specifically, these time motifs turned out to be similar to time courses that domain experts would look for during post-hoc data-driven quality estimation of the manufactured metal sheets.

However, domain experts only possess knowledge about the time series motifs corresponding to actual cracks, while time series motifs hinting towards the formation of cracks in the future were unknown. These time series motifs could be extracted from the trained network and constitute valuable insights for domain experts about possible reasons leading to the formation of cracks. The results of the study are solely derived from the investigation of the learned representation in the first layer of the network, as the temporal courses of the extracted time series motifs in the first layer is directly interpretable as they only constitute mere cutouts of the original time series data. The time series motifs in deeper layers of the network were not investigated and it remains unclear how much value they possess to facilitate increased transparency and interpretability for the network's decision process due to their not-straight-forward relation to the original sensor data.

The second transfer study investigated an analogous mechanism to overt and covert attention found in the mammalian brain in artificial neural networks to determine the importance of individual inputs on spatial and temporal scale. Specifically, the results demonstrated the feasibility to analyze the distribution of weights of attention layers, as part of the network's learned representation, with respect to the learned ability to classify the sensor time series data as well as to predict their temporal course. The adaptation of the attention mechanism from the transformer architecture and the customization of a network architecture for the deep drawing use case was demonstrated to yield valuable insights about a) the relations between the individual sensors to each other with respect to their importance for the learning task as well as about b) the most important points in time for the individual sensors, similarly to the first transfer study, but with higher temporal resolution. Specifically, the distribution of the sensor attention weights showed that two out of the eight sensors were significantly more important for the network to classify and predict process failures than the other sensors. This insight can be interpreted with respect to the spatial distribution of the sensors across the deep drawing tool. It suggests that most cracks form close to these sensors and are predominantly detected in their time series data. Thus, a possible measure could be to extend the tool with further sensors at the appropriate locations of the tool to provide a higher spatial resolution in the most important locations to detect process failures even more reliably.

The results of both studies are based on data from manufacturing a single specific car body part. This part is manufactured with a single specific deep drawing tool that is designed for the exact shape of the particular body part. No data acquired from other deep drawing tools manufacturing other body parts was analyzed in the framework of this thesis and it remains unknown to what

degree the presented results are specific to the peculiarities of the tool and the shape of the manufactured part. However, contrary to the results, the utilized approaches are not at all specific to the use case of deep drawing and can be transferred to any industrial use case providing sensor time series data with the goal to classify the data intro categories and predict their temporal course. The transfer merely requires the adaption of the network architecture, for example to account for a different number of sensors or a different number of classes, and to train the network with the process data. After the initial training and evaluation of the network, the approach for extracting relevant time windows and time series motifs remains the same for both approaches presented in the two transfer studies.

7.2 Future Research Directions

"The future is not a gift. It is an ***achievement.*** *"*

– Albert Einstein, German Nobel laureate physicist

The limitations of the conducted studies and the presented results suggest a number of **possible research directions to follow in the future** that build upon these results. Given the presented results of utilizing network ablations to characterize the role of individual neurons for the network's learned task, the most direct way to expand on the results of this thesis is to clarify the degree of coherence and/or incoherence with other metrics to determine a neuron's importance with respect to the learned task. An immediate follow up study to the presented results would be to investigate a large variety of different network architectures trained to perform different classification tasks on different datasets and characterize the importance of individual neurons with different metrics to compare the results with each other. A further possible direction for future research lies in the extension of the investigation of effects of network ablations and the identification of functional paths and sub-networks within the trained network. Specifically, since the ablation of single neurons does not only remove the single neuron in isolation, but also influences all subsequent neurons in hierarchically lower layers, which do not receive the same input as before the removal of the individual neuron, an interesting direction to pursue would be to investigate the extent to which the removal of single neurons affects the subsequent neurons. This way, functional paths with specific importance for the learning tasks might be uncovered, and even the combination of several paths possibly branching into each other would constitute functional sub-networks within the whole trained network. To this end, a subsequent study could investigate the effects of ablations of single neurons

along distinct paths along the layers of a network and determine paths depending on the strongest effect of the subsequent ablations along those paths. Characterizing all paths with respect to their importance for individual classes, they can be combined to sub-networks, which constitute a collection of paths that are most important for individual classes.

The results presented regarding the structure and organization of learned representations via activation embeddings investigated these representations only after training the networks to reach near state-of-the-art performances for their respective learning tasks. Thus, a possible direction for future research lies in the investigation of how these representations emerge during training. Specifically, an interesting question to pursue would be whether the representations emerge at different times during training in different layers or whether they emerge simultaneously in all layers. Possibly, the representation emerges in earlier layers before it emerges in later layers, since the later layers are somewhat dependent on the information they receive from the preceding layers. To this end, a subsequent study would repeat the investigations of this thesis for different checkpoints of a network undergoing training up until it reaches state of the art performance. The comparison of the results of the different network checkpoints would yield insights about the emergence of the structure and organization shown to be present in the final network.

In the same sense, the reported activation patterns of trained actor networks connected to achieving high rewards in motor control tasks as well as the behavioral anchors have only been investigated in the trained network. Similarly, the emergence of these patterns and anchors during training would be an interesting direction to pursue. Specifically, given the observation of network ablations distorting the activation pattern that leads to high rewards, it would be interesting to investigate whether different patterns, all leading to high rewards, are achievable and whether some patterns are more robust against ablations than others. To this end, in the same fashion as the previously proposed follow-up study, a subsequent study would repeat the investigations of this thesis for different checkpoints, extract the correlation patterns leading to the highest rewards for the respective checkpoints and investigate the emergence of the final pattern, which is connected to the highest reward achieved, over time.

These previously proposed lines of investigation of the effects of network ablations is closely connected to researching the broader aspect of robustness of neural networks against unintended network alterations. Specifically, the applications of neural networks as part of technical software systems in critical domains makes them, in principle, a possible target for malicious attacks. The alteration of network weights or the removal of critical neurons and pathways could have

potentially fatal consequences, for example in autonomous driving scenarios, in which traffic lights or road signs are misclassified as a result of a malicious alteration of the network. Thus, an interesting research direction to pursue would be the investigation of methods and approaches to make neural networks aware of unintended alterations and facilitate robustness of their learned representations regarding their intended task. To this end, a subsequent study would aim to combine the insights gained from the recovery training investigation of the second research study and the general insights from all the conducted network ablations to automatically trigger a network's recovery training upon a detected change of the distributions of layer weights. Specifically, the study would combine different kinds of network alterations, e.g., weight manipulation of neuron ablations, and the network would trigger a recovery training process to maintain its original performance as reliably as possible.

On a more general note, the results of the conducted studies, and more specifically, the chosen methods and approaches and the neuroscience inspired perspective on artificial neural networks and their learned representations demonstrated the feasibility of an empirical access to understand these learning models rather than a strictly analytical one, which has been taken many times in the past and lead to extensive optimization driven research largely neglecting the field of transparency, interpretability and explainability. An interesting and urgently necessary research direction for future work is the establishment of a scientific standard for the falsifiability of empirical studies conducted in the field of understanding learned representation of artificial neural networks. Uncovering parallels between the structure and organization of represented knowledge in artificial and biological neural systems opens up measures and possibilities for initial large-scale studies in artificial systems before transferring them to biological systems, which would yield a great benefit for both research fields and paves the way into an era of interdisciplinary research on general intelligence. Ultimately, to pave the way towards a new perspective of neuroscience inspired empirical studies on artificial neural networks, a currently emerging idea [226], aiming to exploit them as a test bed for neuroscientific research bridges the gap between both fields and addresses one of the most critical and prevalent issues in modern neuroscience, i.e., the issue of reproducibility, which stems from the large differences between brains and the commonly small sample sizes in neuroscientific studies [227].

Summary

8

"Had the ***melody*** *not* ***reached its end*** *it would not have* ***reached its goal*** *either."*

– Friedrich Wilhelm Nietzsche, German philosopher and philologist

This dissertation thesis addressed the topic of facilitating transparency and interpretability for learned representations of artificial neural networks. **The goal of this thesis** was to investigate a neuroscience inspired approach to tackle artificial neural systems. Specifically, this approach is characterized by its empirical nature to investigate a neural network's learned representation in the same spirit as neuroscientific studies of the brain, i.e., treating the network under consideration as a complex and nebulous neural system, which exhibits a distinctive learned representation that emerged as a result of its training with respect to solve a specific task. To achieve this goal, four distinct research questions provided a guiding framework for the bottom-up fashioned scientific investigation of the object of investigation, i.e., learned representations of artificial neural networks. Specifically, the research questions addressed the characterization of single neurons within trained networks with respect to their importance for the network's learned task as well as the formation of groups of neurons with similar roles that cluster into functional neuron populations. They further addressed the structure and organization of these neurons' activations as a result of the network performing its learned task during inference as well as the relation between their organized activations and the networks decision-making, i.e., either its classification result in the context of supervised learning or its exhibited behavior policy in the context of deep reinforcement learning. Finally, they addressed the transferability of the approach to the industrial domain to facilitate transparency of a real-world manufacturing process via the interpretability of the trained network's learned representations. The four research questions were answered in a series

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0_8

of five research studies, which established the methodological foundation, and two subsequent transfer studies, which demonstrated the transferability of these methods to an exemplary manufacturing scenario, i.e., a predictive quality use case of deep drawing of car body parts. The results of the studies demonstrated the feasibility of network ablations to determine the importance of individual neurons and groups of neurons for the learned task of the investigated networks. In combination with embeddings of the activations of neurons, layers, and the whole network, they proved suitable to investigate the structure and organization of the learned representations within the network and relate them to the networks' emerging behavior utilizing statistical analysis methods like statistical significance tests for weight distributions or correlation analyses for activation patterns. Furthermore, the transferability of the established methods was shown to be able to facilitate transparency for the deep drawing manufacturing process of car body parts by providing insights about the importance of the different sensors built in the deep drawing tool with respect to the prediction and forecast of process failures. The results of the studies give way to a number of interesting future research directions all under the umbrella of bridging the gap between the two related research fields of neuroscience and XAI, ultimately to establish a scientific standard for the falsifiability of empirical studies conducted in the field of understanding learned representation of artificial neural networks.

References

1. Hubel DH, Wiesel TN (1959) Receptive fields of single neurons in the cat's striate cortex. The Journal of physiology 148:574
2. Hubel DH, Wiesel TN (1960) Receptive fields of optic nerve fibres in the spider monkey. The Journal of physiology 154:572
3. Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology 160:106
4. Hubel DH, Wiesel TN (1963) Receptive fields of cells in striate cortex of very young, visually inexperienced kittens. Journal of neurophysiology 26:994–1002
5. Hubel DH, Wiesel TN (1965) Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. Journal of neurophysiology 28:229–289
6. Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. The Journal of physiology 195:215–243
7. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5:115–133
8. Hebb DO (1949) The organization of behavior: a neuropsychological theory. J. Wiley; Chapman & Hall
9. Wikipedia McCulloch-Pitts-Zelle—https://de.wikipedia.org/wiki/McCulloch-Pitts-Zelle. Accessed 25 Oct 2020
10. Markram H, Gerstner W, Sjöström PJ (2011) A history of spike-timing-dependent plasticity. Frontiers in synaptic neuroscience 3:4
11. OpenDataScience https://opendatascience.com/the-history-of-neural-networks-and-ai-part-i/. Accessed 25 Oct 2020
12. Russell S, Norvig P (2002) Artificial intelligence: a modern approach
13. Mirowski P (2003) McCorduck's Machines Who Think after Twenty-Five Years Revisiting the Origins of AI. AI Magazine 24:135
14. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65:386
15. Minsky M, Papert S (1969) An introduction to computational geometry. Cambridge tiass., HIT
16. Minsky M, Papert SA (2017) Perceptrons: An introduction to computational geometry. MIT press

R. Meyes, *Transparency and Interpretability for Learned Representations of Artificial Neural Networks*, https://doi.org/10.1007/978-3-658-40004-0

17. Werbos P (1974) Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph. D. dissertation, Harvard University
18. Werbos PJ (1982) Applications of advances in nonlinear sensitivity analysis. In: System modeling and optimization. Springer, pp 762–770
19. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. nature 323:533–536
20. Wan EA (1993) Time series prediction by using a connectionist network with internal delay lines. In: SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-, vol 15, p 195
21. Everingham M, van Gool L, Williams CKI et al. (2010) The pascal visual object classes (voc) challenge. International journal of computer vision 88:303–338
22. Graves A, Schmidhuber J (2008) Offline handwriting recognition with multidimensional recurrent neural networks. Advances in neural information processing systems 21:545–552
23. Graves A, Liwicki M, Fernández S et al. (2008) A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence 31:855–868
24. Kurzweil AI https://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions. Accessed 05 Nov 2020
25. Deng J, Dong W, Socher R et al. (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255
26. LeCun Y, Boser B, Denker JS et al. (1989) Backpropagation applied to handwritten zip code recognition. Neural computation 1:541–551
27. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25:1097–1105
28. OpenAI https://openai.com/blog/microscope/. Accessed 07 Jan 2022
29. Girshick R, Donahue J, Darrell T et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
30. van de Sande KEA, Uijlings JRR, Gevers T et al. (2011) Segmentation as selective search for object recognition. In: 2011 International Conference on Computer Vision, pp 1879–1886
31. Lin T-Y, Maire M, Belongie S et al. (2014) Microsoft coco: Common objects in context. In: European conference on computer vision, pp 740–755
32. Eykholt K, Evtimov I, Fernandes E et al. (2018) Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1625–1634
33. Zhou Z, Siddiquee MMR, Tajbakhsh N et al. (2018) Unet++: A nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp 3–11
34. Fujisawa Y, Otomo Y, Ogata Y et al. (2019) Deep-learning-based, computer-aided classifier developed with a small dataset of clinical images surpasses board-certified dermatologists in skin tumour diagnosis. British Journal of Dermatology 180:373–381

35. Gerazov B, Conceicao RC (2017) Deep learning for tumour classification in homogeneous breast tissue in medical microwave imaging. In: IEEE EUROCON 2017-17th International Conference on Smart Technologies, pp 564–569
36. Iizuka O, Kanavati F, Kato K et al. (2020) Deep learning models for histopathological classification of gastric and colonic epithelial tumours. Scientific Reports 10:1–11
37. Tesauro G (1995) Temporal difference learning and TD-Gammon. Communications of the ACM 38:58–68
38. Mnih V, Kavukcuoglu K, Silver D et al. (2013) Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602
39. Badia AP, Piot B, Kapturowski S et al. (2020) Agent57: Outperforming the atari human benchmark. arXiv preprint arXiv:2003.13350
40. Silver D, Huang A, Maddison CJ et al. (2016) Mastering the game of Go with deep neural networks and tree search. nature 529:484–489
41. DeepMind https://deepmind.com/research/case-studies/alphago-the-story-so-far. Accessed 05 Nov 2020
42. Silver D, Schrittwieser J, Simonyan K et al. (2017) Mastering the game of go without human knowledge. nature 550:354–359
43. Silver D, Hubert T, Schrittwieser J et al. (2018) A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362:1140–1144
44. Kasparov G (2018) Chess, a Drosophila of reasoning
45. Schrittwieser J, Antonoglou I, Hubert T et al. (2019) Mastering atari, go, chess and shogi by planning with a learned model. arXiv preprint arXiv:1911.08265
46. Hessel M, Modayil J, van Hasselt H et al. (2017) Rainbow: Combining improvements in deep reinforcement learning. arXiv preprint arXiv:1710.02298
47. Valve https://blog.dota2.com/. Accessed 10 Nov 2020
48. Blizzard https://starcraft2.com/en-us/. Accessed 11 Nov 2020
49. Brown TB, Mann B, Ryder N et al. (2020) Language models are few-shot learners. arXiv preprint arXiv:2005.14165
50. Olah C, Carter S (2017) Research Debt. Distill. https://doi.org/10.23915/distill.00005
51. Universität Linz http://paedpsych.jk.uni-linz.ac.at/4711/LEHRTEXTE/LERNEN/klassi.htm). Accessed 05 Nov 2020
52. Babkin BP (1949) Pavlov. A biography
53. Tesauro G (1986) Simple neural models of classical conditioning. Biological cybernetics 55:187–200
54. YouTube https://www.youtube.com/watch?v=8VdFf3egwfg. Accessed 18 Nov 2020
55. Quiroga RQ, Reddy L, Kreiman G et al. (2005) Invariant visual representation by single neurons in the human brain. nature 435:1102–1107
56. Kaschube M, Schnabel M, Wolf F (2008) Self-organization and the selection of pinwheel density in visual cortical development. New journal of physics 10:15009
57. Da Costa S, van der Zwaag W, Marques JP et al. (2011) Human primary auditory cortex follows the shape of Heschl's gyrus. Journal of Neuroscience 31:14067–14075
58. Nakamura A, Yamada T, Goto A et al. (1998) Somatosensory homunculus as drawn by MEG. Neuroimage 7:377–386
59. Wikipedia Functional magnetic resonance imaging—https://en.wikipedia.org/wiki/Functional_magnetic_resonance_imaging. Accessed 14 Jan 2022

60. Kandel ER, Schwartz JH, Jessell TM et al. (2000) Principles of neural science, vol 4. McGraw-hill New York
61. van der Maaten L, Postma E, van den Herik J et al. (2009) Dimensionality reduction: a comparative. J Mach Learn Res 10:13
62. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. Journal of educational psychology 24:417
63. McInnes L, Healy J, Melville J (2018) Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426
64. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. Journal of machine learning research 9
65. van der Maaten L (2014) Accelerating t-SNE using tree-based algorithms. The Journal of Machine Learning Research 15:3221–3245
66. Student (1908) The probable error of a mean. Biometrika:1–25
67. Dimitriadis G, Neto JP, Kampff AR (2018) T-SNE visualization of large-scale neural recordings. Neural computation 30:1750–1774
68. Abdelmoula WM, Balluff B, Englert S et al. (2016) Data-driven identification of prognostic tumor subpopulations using spatially mapped t-SNE of mass spectrometry imaging data. Proceedings of the National Academy of Sciences 113:12244–12249
69. LeCun Y, Jackel LD, Bottou L et al. (1995) Comparison of learning algorithms for handwritten digit recognition. In: International conference on artificial neural networks, vol 60, pp 53–60
70. Li Deng (2012) The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine 29:141–142
71. Wattenberg M, Viégas F, Johnson I (2016) How to use t-SNE effectively. Distill 1:e2
72. Becht E, McInnes L, Healy J et al. (2019) Dimensionality reduction for visualizing single-cell data using UMAP. Nature biotechnology 37:38–44
73. Li X, Dyck OE, Oxley MP et al. (2019) Manifold learning of four-dimensional scanning transmission electron microscopy. npj Computational Materials 5:1–8
74. Nene SA, Nayar SK, Murase H et al. (1996) Columbia object image library (coil-100)
75. Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747
76. Mikolov T, Sutskever I, Chen K et al. (2013) Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 26:3111–3119
77. Morcos AS, Barrett DGT, Rabinowitz NC et al. (2018) On the importance of single directions for generalization. arXiv preprint arXiv:1803.06959
78. Squire LR (2009) The legacy of patient HM for neuroscience. Neuron 61:6–9
79. Corkin S (1968) Acquisition of motor skill after bilateral medial temporal-lobe excision. Neuropsychologia 6:255–265
80. Milner B, Corkin S, Teuber H-L (1968) Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of HM. Neuropsychologia 6:215–234
81. Molchanov P, Tyree S, Karras T et al. (2016) Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440
82. Pedregosa F, Varoquaux G, Gramfort A et al. (2011) Scikit-learn: Machine Learning in Python. Journal of machine learning research 12:2825–2830

83. Esteva A, Kuprel B, Novoa RA et al. (2017) Dermatologist-level classification of skin cancer with deep neural networks. nature 542:115–118
84. Clanuwat T, Bober-Irizar M, Kitamoto A et al. (2018) Deep learning for classical japanese literature. arXiv preprint arXiv:1812.01718
85. Lillicrap TP, Hunt JJ, Pritzel A et al. (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971
86. Irpan A (2018) Deep Reinforcement Learning Doesn't Work Yet
87. Amirikian B, Georgopulos AP (2000) Directional tuning profiles of motor cortical cells. Neuroscience research 36:73–79
88. Krauß J, Pacheco BM, Zang HM et al. (2020) Automated machine learning for predictive quality in production. Procedia CIRP 93:443–448
89. Cammarata N, Carter S, Goh G et al. (2020) Thread: Circuits. Distill. https://doi.org/10.23915/distill.00024
90. Arrieta AB, D\'\ia S, Gil-López S et al. (2020) Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58:82–115
91. Tjoa E, Guan C (2020) A survey on explainable artificial intelligence (xai): Toward medical xai. IEEE Transactions on Neural Networks and Learning Systems
92. Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access 6:52138–52160
93. Das A, Rad P (2020) Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv preprint arXiv:2006.11371
94. Doran D, Schulz S, Besold TR (2017) What does explainable AI really mean? A new conceptualization of perspectives. arXiv preprint arXiv:1710.00794
95. Puiutta E, Veith EM (2020) Explainable reinforcement learning: A survey. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, pp 77–95
96. Doshi-Velez F, Kim B (2017) Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608
97. Springenberg JT, Dosovitskiy A, Brox T et al. (2014) Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806
98. Cheney N, Schrimpf M, Kreiman G (2017) On the robustness of convolutional neural networks to internal architecture and weight perturbations. arXiv preprint arXiv:1703.08245
99. Smilkov D, Carter S, Sculley D et al. (2017) Direct-manipulation visualization of deep networks. arXiv preprint arXiv:1708.03788
100. Srivastava N, Hinton G, Krizhevsky A et al. (2014) Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15:1929–1958
101. Li H, Kadav A, Durdanovic I et al. (2016) Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710
102. Anwar S, Hwang K, Sung W (2017) Structured pruning of deep convolutional neural networks. ACM Journal on Emerging Technologies in Computing Systems (JETC) 13:1–18

103. Hendrycks D, Zhao K, Basart S et al. (2021) Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 15262–15271
104. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation 23:828–841
105. Sorzano COS, Vargas J, Montano AP (2014) A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877
106. Goodman B, Flaxman S (2017) European Union regulations on algorithmic decision-making and a "right to explanation". AI Magazine 38:50–57
107. Lipton ZC (2018) The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue 16:31–57
108. Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034
109. Papernot N, McDaniel P, Goodfellow I et al. (2017) Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp 506–519
110. Faust K, Xie Q, Han D et al. (2018) Visualizing histopathologic deep learning classification and anomaly detection using nonlinear feature space dimensionality reduction. BMC bioinformatics 19:1–15
111. Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. In: Proceedings of the IEEE international conference on computer vision, pp 3429–3437
112. Zintgraf LM, Cohen TS, Adel T et al. (2017) Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:1702.04595
113. Arras L, Horn F, Montavon G et al. (2017) What is relevant in a text document?: An interpretable machine learning approach. PloS one 12:e0181142
114. Zhou B, Khosla A, Lapedriza A et al. (2014) Object detectors emerge in deep scene cnns. arXiv preprint arXiv:1412.6856
115. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision, pp 818–833
116. Greydanus S, Koul A, Dodge J et al. (2018) Visualizing and understanding atari agents. In: International Conference on Machine Learning, pp 1792–1801
117. Cobbe K, Klimov O, Hesse C et al. (2019) Quantifying generalization in reinforcement learning. In: International Conference on Machine Learning, pp 1282–1289
118. Hilton J, Cammarata N, Carter S et al. (2020) Understanding RL Vision. Distill. https://doi.org/10.23915/distill.00029
119. Wang T, Rudin C, Doshi-Velez F et al. (2017) A bayesian framework for learning rule sets for interpretable classification. The Journal of Machine Learning Research 18:2357–2393
120. Letham B, Rudin C, McCormick TH et al. (2015) Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics 9:1350–1371
121. Lakkaraju H, Kamar E, Caruana R et al. (2017) Interpretable & explorable approximations of black box models. arXiv preprint arXiv:1707.01154

122. Ribeiro MT, Singh S, Guestrin C (2016) Why should i trust you? Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
123. Binder A, Bach S, Montavon G et al. (2016) Layer-wise relevance propagation for deep neural network architectures. In: Information science and applications (ICISA) 2016. Springer, pp 913–922
124. Montavon G, Bach S, Binder A et al. (2016) Deep taylor decomposition of neural networks. In: Proceedings of the ICML 2016 Workshop on Visualization for Deep Learning
125. Montavon G, Samek W, Müller K-R (2018) Methods for interpreting and understanding deep neural networks. Digital Signal Processing 73:1–15
126. Selvaraju RR, Cogswell M, Das A et al. (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
127. Choi E, Bahadori MT, Kulas JA et al. (2016) Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. arXiv preprint arXiv:1608.05745
128. Harley AW (2015) An interactive node-link visualization of convolutional neural networks. In: International Symposium on Visual Computing, pp 867–877
129. Chung S, Suh S, Park C et al. (2016) ReVACNN: Real-time visual analytics for convolutional neural network. In: ACM SIGKDD Workshop on Interactive Data Exploration and Analytics (IDEA), p 7
130. Liu M, Shi J, Li Z et al. (2016) Towards better analysis of deep convolutional neural networks. IEEE transactions on visualization and computer graphics 23:91–100
131. Kahng M, Andrews PY, Kalro A et al. (2017) A cti v is: Visual exploration of industry-scale deep neural network models. IEEE transactions on visualization and computer graphics 24:88–97
132. Mahendran A, Vedaldi A (2015) Understanding deep image representations by inverting them. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5188–5196
133. Yosinski J, Clune J, Nguyen A et al. (2015) Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579
134. Olah C, Mordvintsev A, Schubert L (2017) Feature visualization. Distill 2:e7
135. Olah C, Satyanarayan A, Johnson I et al. (2018) The building blocks of interpretability. Distill 3:e10
136. Carter S, Armstrong Z, Schubert L et al. (2019) Activation atlas. Distill 4:e15
137. Li M, Scheidegger C Toward Comparing DNNs with UMAP Tour—https://tiga1231.github.io/umap-tour/. Accessed 15 Nov 2021
138. Dibia V ConvNet Playground, https://convnetplayground.fastforwardlabs.com. Accessed 17 Oct 2021
139. Elloumi Z, Besacier L, Galibert O et al. (2018) Analyzing learned representations of a deep asr performance prediction model. arXiv preprint arXiv:1808.08573
140. Belinkov Y, Glass J (2017) Analyzing hidden representations in end-to-end automatic speech recognition systems. arXiv preprint arXiv:1709.04482

141. Aubry M, Russell BC (2015) Understanding deep features with computer-generated imagery. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2875–2883
142. Rauber PE, Fadel SG, Falcao AX et al. (2016) Visualizing the hidden activity of artificial neural networks. IEEE transactions on visualization and computer graphics 23:101–110
143. Kruskal JB (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29:1–27
144. Bau A, Belinkov Y, Sajjad H et al. (2018) Identifying and controlling important neurons in neural machine translation. arXiv preprint arXiv:1811.01157
145. Bau D, Zhou B, Khosla A et al. (2017) Network dissection: Quantifying interpretability of deep visual representations. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6541–6549
146. Radford A, Jozefowicz R, Sutskever I (2017) Learning to generate reviews and discovering sentiment. arXiv preprint arXiv:1704.01444
147. Agrawal P, Girshick R, Malik J (2014) Analyzing the performance of multilayer neural networks for object recognition. In: European conference on computer vision, pp 329–344
148. Nguyen A, Yosinski J, Clune J (2016) Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616
149. Kim B, Wattenberg M, Gilmer J et al. (2018) Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International conference on machine learning, pp 2668–2677
150. Belinkov Y, Màrquez L, Sajjad H et al. (2018) Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. arXiv preprint arXiv:1801.07772
151. Yang Z, Dai Z, Yang Y et al. (2019) Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems 32
152. Durrani N, Sajjad H, Dalvi F et al. (2020) Analyzing individual neurons in pre-trained language models. arXiv preprint arXiv:2010.02695
153. Filan D, Casper S, Hod S et al. (2021) Clusterability in Neural Networks. arXiv preprint arXiv:2103.03386
154. Csordás R, van Steenkiste S, Schmidhuber J (2020) Are neural nets modular? inspecting functional modularity through differentiable weight masks. arXiv preprint arXiv:2010.02066
155. Dalvi F, Nortonsmith A, Bau A et al. (2019) NeuroX: A toolkit for analyzing individual neurons in neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 9851–9852
156. Goodfellow I, Pouget-Abadie J, Mirza M et al. (2020) Generative adversarial networks. Communications of the ACM 63:139–144
157. Bau D, Zhu J-Y, Strobelt H et al. (2020) Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences 117:30071–30078
158. Bau D, Zhu J-Y, Strobelt H et al. (2018) Gan dissection: Visualizing and understanding generative adversarial networks. arXiv preprint arXiv:1811.10597

159. Li J, Monroe W, Jurafsky D (2016) Understanding neural networks through representation erasure. arXiv preprint arXiv:1612.08220
160. Dalvi F, Durrani N, Sajjad H et al. (2019) What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 6309–6317
161. Olah C, Cammarata N, Schubert L et al. (2020) Zoom In: An Introduction to Circuits. Distill. https://doi.org/10.23915/distill.00024.001
162. Szegedy C, Liu W, Jia Y et al. (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
163. Olah C, Cammarata N, Schubert L et al. (2020) An Overview of Early Vision in InceptionV1. Distill. https://doi.org/10.23915/distill.00024.002
164. Cammarata N, Goh G, Carter S et al. (2020) Curve Detectors. Distill. https://doi.org/10.23915/distill.00024.003
165. Cammarata N, Goh G, Carter S et al. (2021) Curve Circuits. Distill. https://doi.org/10.23915/distill.00024.006
166. Schubert L, Voss C, Cammarata N et al. (2021) High-Low Frequency Detectors. Distill. https://doi.org/10.23915/distill.00024.005
167. Olah C, Cammarata N, Voss C et al. (2020) Naturally Occurring Equivariance in Neural Networks. Distill. https://doi.org/10.23915/distill.00024.004
168. Voss C, Goh G, Cammarata N et al. (2021) Branch Specialization. Distill. https://doi.org/10.23915/distill.00024.008
169. Petrov M, Voss C, Schubert L et al. (2021) Weight Banding. Distill. https://doi.org/10.23915/distill.00024.009
170. He K, Zhang X, Ren S et al. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
171. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
172. Meyes R, Lu M, Puiseau CW de et al. (2019) Ablation studies in artificial neural networks. arXiv preprint arXiv:1901.08644
173. Meyes R, Lu M, Puiseau CW de et al. (2019) Ablation Studies to Uncover Structure of Learned Representations in Artificial Neural Networks. In: Proceedings on the International Conference on Artificial Intelligence (ICAI), pp 185–191
174. Meyes R, Puiseau CW de, Posada-Moreno A et al. (2020) Under the Hood of Neural Networks: Characterizing Learned Representations by Functional Neuron Populations and Network Ablations. arXiv preprint arXiv:2004.01254
175. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Icml
176. Bishop CM (2006) Pattern recognition and machine learning: springer New York
177. Hettmansperger TP, McKean JW (2010) Robust nonparametric statistical methods. CRC Press
178. SPSS SPSS TUTORIALS: PEARSON CORRELATION—https://libguides.library.kent.edu/SPSS/PearsonCorr. Accessed 08 Dec 2021
179. Corder GW, Foreman DI (2014) Nonparametric statistics: A step-by-step approach. John Wiley & Sons

180. Meyes R, Puiseau CW de, Posada-Moreno A et al. Under the Hood of Neural Networks: Characterizing Learned Representations by Functional Neuron Populations and Network Ablations. In: The Steering Committee of The World Congress in Computer Science, Computer … 2021—Proceedings on the International Conference, in print
181. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press
182. Paszke A, Gross S, Massa F et al. (2019) Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32:8026–8037
183. Virtanen P, Gommers R, Oliphant TE et al. (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature methods 17:261–272
184. Gatys LA, Ecker AS, Bethge M (2015) A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576
185. Meyes R, Schneider M, Meisen T (2020) How Do You Act? An Empirical Study to Understand Behavior of Deep Reinforcement Learning Agents. arXiv preprint arXiv:2004.03237
186. Lu M, Meyes, R, Posada, A., Meisen T Exploring Behavior via Neural Network Activations in Deep Reinforcement Learning Agents. In: The Steering Committee of The World Congress in Computer Science, Computer … 2021—Proceedings on the International Conference, in print
187. Meyes R, Schneider M, Meisen T How Do You Act? An Empirical Study to Understand Behavior of Deep Reinforcement Learning Agents. In: The Steering Committee of The World Congress in Computer Science, Computer … 2021—Proceedings on the International Conference, in print
188. Popov I, Heess N, Lillicrap T et al. (2017) Data-efficient deep reinforcement learning for dexterous manipulation. arXiv preprint arXiv:1704.03073
189. Schulman J, Wolski F, Dhariwal P et al. (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347
190. Brockman G, Cheung V, Pettersson L et al. (2016) Openai gym. arXiv preprint arXiv:1606.01540
191. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:1607.06450
192. Uhlenbeck GE, Ornstein LS (1930) On the theory of the Brownian motion. Physical review 36:823
193. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
194. Veit A, Wilber MJ, Belongie S (2016) Residual networks behave like ensembles of relatively shallow networks. Advances in neural information processing systems 29:550–558
195. Rafegas I, Vanrell M, Alexandre LA et al. (2020) Understanding trained CNNs by indexing neuron selectivity. Pattern Recognition Letters 136:318–325
196. Deliagina TG (2008) Overview of motor systems. types of movements: reflexes, rhythmical and voluntary movements. In: Dynamical Systems, Wave-Based Computation and Neuro-Inspired Robots. Springer, pp 3–14
197. Tassa Y, Doron Y, Muldal A et al. (2018) Deepmind control suite. arXiv preprint arXiv:1801.00690

198. Tunyasuvunakool S, Muldal A, Doron Y et al. (2020) dm_control: Software and tasks for continuous control. Software Impacts 6:100022
199. Yarats D, Kostrikov I (2020) Soft actor-critic (sac) implementation in pytorch
200. Li M, Zhao Z, Scheidegger C (2020) Visualizing neural networks with the grand tour. Distill 5:e25
201. van der Maaten L (2009) Learning a parametric embedding by preserving local structure. In: Artificial Intelligence and Statistics, pp 384–391
202. Scikit-Learn Documentaiton Silhouette Coefficient—https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. Accessed 23 Dec 2021
203. Scikit-Learn Documentaiton Calinski and Harabasz Score—https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score.html. Accessed 23 Dec 2021
204. Scikit-Learn Documentaiton Davies-Bouldin Score—https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html. Accessed 23 Dec 2021
205. Meyes R, Donauer J, Schmeing A et al. (2019) A recurrent neural network architecture for failure prediction in deep drawing sensory time series data. Procedia Manufacturing 34:789–797
206. Meyes R, Hütten N, Meisen T (2021) Transparent and Interpretable Failure Prediction of Sensor Time Series Data with Convolutional Neural Networks. Procedia CIRP 104:1446–1451
207. Chattopadhay A, Sarkar A, Howlader P et al. (2018) Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV), pp 839–847
208. DIN 8584-3 (2003) 2003–09—Manufacturing processes forming under combination of tensile and compressive conditions—Part 3: Deep drawing; Classification, subdivision, terms and definitions
209. SciPy Documentation Butterworth Filter—https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html. Accessed 02 Jan 2022
210. Itti L, Koch C (2000) A saliency-based search mechanism for overt and covert shifts of visual attention. Vision research 40:1489–1506
211. Vaswani A, Shazeer N, Parmar N et al. (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
212. Qin Y, Song D, Chen H et al. (2017) A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971
213. Cuturi M, Blondel M (2017) Soft-dtw: a differentiable loss function for time-series. In: International conference on machine learning, pp 894–903
214. Meyes R Deep Drawing Sensor Signal Forecasting—http://demo.tmdt.uni-wuppertal.de/demonstrators/sensor_data_analytics_showroom/sensor_signal_forecasting. Accessed 24 Mar 2022
215. Jain S, Wallace BC (2019) Attention is not explanation. arXiv preprint arXiv:1902.10186
216. Wiegreffe S, Pinter Y (2019) Attention is not not explanation. arXiv preprint arXiv:1908.04626
217. Grimsley C, Mayfield E, Bursten J (2020) Why attention is not explanation: Surgical intervention and causal reasoning about neural models

218. Tutek M, Šnajder J (2020) Staying True to Your Word:(How) Can Attention Become Explanation? arXiv preprint arXiv:2005.09379
219. Bastings J, Filippova K (2020) The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? arXiv preprint arXiv:2010.05607
220. Mullenbach J, Wiegreffe S, Duke J et al. (2018) Explainable prediction of medical codes from clinical text. arXiv preprint arXiv:1802.05695
221. Sen C, Hartvigsen T, Yin B et al. (2020) Human attention maps for text classification: Do humans and neural networks focus on the same words? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 4596–4608
222. Sood E, Tannert S, Frassinelli D et al. (2020) Interpreting attention models with human visual attention in machine reading comprehension. arXiv preprint arXiv:2010.06396
223. Mahalanobis PC (1936) On the generalized distance in statistics. In:
224. McLachlan GJ (1999) Mahalanobis distance. Resonance 4:20–26
225. Gagunashvili ND (2006) Comparison of weighted and unweighted histograms. arXiv preprint physics/0605123
226. Botvinick M, Wang JX, Dabney W et al. (2020) Deep reinforcement learning and its neuroscientific implications. Neuron 107:603–616
227. Marek S, Tervo-Clemmens B, Calabro FJ et al. (2022) Reproducible brain-wide association studies require thousands of individuals. nature:1–7
228. Colbeck R, Renner R (2011) No extension of quantum theory can have improved predictive power. Nature communications 2:1–5
229. Ghirardi G, Romano R (2013) Ontological models predictively inequivalent to quantum theory. Physical review letters 110:170404
230. Hossenfelder S (2011) Testing super-deterministic hidden variables theories. Foundations of Physics 41:1521–1531
231. Piran N, Teall T (2012) The developmental theory of embodiment. Preventing eating-related and weight-related disorders: Collaborative research, advocacy, and policy change:169–198
232. Wikipedia Double-slit experiment—https://en.wikipedia.org/wiki/Double-slit_experiment. Accessed 25 Oct 2020
233. Herculano-Houzel S (2009) The human brain in numbers: a linearly scaled-up primate brain. Frontiers in human neuroscience 3:31
234. Halstead WC (1947) Brain and intelligence; a quantitative study of the frontal lobes
235. Jerison H (2012) Evolution of the brain and intelligence. Elsevier
236. Passingham RE (1975) The brain and intelligence. Brain, Behavior and Evolution 11:1–15
237. Roth G, Dicke U (2005) Evolution of the brain and intelligence. Trends in cognitive sciences 9:250–257
238. Hu J, Li Shen, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
239. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
240. Ren S, He K, Girshick R et al. (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99

241. Redmon J, Divvala S, Girshick R et al. (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
242. He K, Gkioxari G, Dollár P et al. (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
243. Wikipedia Geschichte des Schachspiels—https://de.wikipedia.org/wiki/Geschichte_des_Schachspiels. Accessed 05 Nov 2020
244. Wikipedia World Chess Championship—https://en.wikipedia.org/wiki/World_Chess_Championship. Accessed 08 Jan 2022
245. Wikipedia Deep Blue—https://de.wikipedia.org/wiki/Deep_Blue. Accessed 05 Nov 2020
246. Leela Chess Zero https://lczero.org/. Accessed 08 Jan 2022
247. Mnih V, Kavukcuoglu K, Silver D et al. (2015) Human-level control through deep reinforcement learning. nature 518:529–533
248. Greg Brockman, Vicki Cheung, Ludwig Pettersson et al. (2016) OpenAI Gym
249. OpenAI (2017) OpenAI Roboschool. https://github.com/openai/roboschool
250. Nair A, Srinivasan P, Blackwell S et al. (2015) Massively parallel methods for deep reinforcement learning. arXiv preprint arXiv:1507.04296
251. Horgan D, Quan J, Budden D et al. (2018) Distributed prioritized experience replay. arXiv preprint arXiv:1803.00933
252. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9:1735–1780
253. Hausknecht M, Stone P (2015) Deep recurrent q-learning for partially observable mdps. arXiv preprint arXiv:1507.06527
254. Kapturowski S, Ostrovski G, Quan J et al. (2018) Recurrent experience replay in distributed reinforcement learning. In: International conference on learning representations
255. Badia AP, Sprechmann P, Vitvitskyi A et al. (2020) Never Give Up: Learning Directed Exploration Strategies. arXiv preprint arXiv:2002.06038
256. Pathak D, Agrawal P, Efros AA et al. (2017) Curiosity-driven exploration by self-supervised prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 16–17
257. Burda Y, Edwards H, Storkey A et al. (2018) Exploration by random network distillation. arXiv preprint arXiv:1810.12894
258. Valve https://developer.valvesoftware.com/wiki/Dota_Bot_Scripting. Accessed 10 Nov 2020
259. OpenAI https://openai.com/blog/openai-five/. Accessed 10 Nov 2020
260. OpenAI https://openai.com/blog/dota-2/. Accessed 11 Nov 2020
261. OpenAI https://openai.com/blog/openai-five-finals/. Accessed 11 Nov 2020
262. OpenAI https://openai.com/projects/five/. Accessed 11 Nov 2020
263. OpenAI https://openai.com/blog/emergent-tool-use/. Accessed 27 Jan 2021
264. Baker B, Kanitscheider I, Markov T et al. (2019) Emergent tool use from multi-agent autocurricula. arXiv preprint arXiv:1909.07528
265. Vinyals O, Ewalds T, Bartunov S et al. (2017) Starcraft ii: A new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782

266. DeepMind https://deepmind.com/blog/announcements/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment. Accessed 11 Nov 2020
267. Vinyals O, Babuschkin I, Chung J et al. (2019) AlphaStar: Mastering the Real-Time Strategy Game StarCraft II
268. Vinyals O, Babuschkin I, Czarnecki WM et al. (2019) Grandmaster level in StarCraft II using multi-agent reinforcement learning. nature 575:350–354
269. Blizzard https://github.com/Blizzard/s2client-proto. Accessed 11 Nov 2020
270. Wikipedia AlphaGo—https://de.wikipedia.org/wiki/AlphaGo. Accessed 18 Nov 2020
271. Google https://9to5google.com/2017/10/19/alphago-zero/. Accessed 18 Nov 2020
272. Chess Internation https://www.chess-international.com/?p=21578. Accessed 18 Nov 2020
273. Wikipedia Backgammon—https://de.wikipedia.org/wiki/Backgammon. Accessed 18 Nov 2020
274. Wikipedia Iwan Petrowitsch Pawlow—https://de.wikipedia.org/wiki/Iwan_Petrowitsch_Pawlow. Accessed 18 Nov 2020
275. Wikipedia Garri Kimowitsch Kasparow—https://de.wikipedia.org/wiki/Garri_Kimowitsch_Kasparow. Accessed 18 Nov 2020
276. Mnih V, Badia AP, Mirza M et al. (2016) Asynchronous methods for deep reinforcement learning. In: International conference on machine learning, pp 1928–1937
277. Espeholt L, Soyer H, Munos R et al. (2018) Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In: International Conference on Machine Learning, pp 1407–1416
278. Glasser MF, Coalson TS, Robinson EC et al. (2016) A multi-modal parcellation of human cerebral cortex. nature 536:171–178
279. Wikipedia CT scan—https://en.wikipedia.org/wiki/CT_scan. Accessed 19 Jan 2022
280. Wikipedia Positron emission tomography—https://en.wikipedia.org/wiki/Positron_emission_tomography. Accessed 19 Jan 2022
281. Wikipedia Electroencephalography—https://en.wikipedia.org/wiki/Electroencephalography. Accessed 19 Jan 2022
282. Wikipedia Magnetoencephalography—https://en.wikipedia.org/wiki/Magnetoencephalography. Accessed 19 Jan 2022
283. Wikipedia Two-photon excitation microscopy—https://en.wikipedia.org/wiki/Two-photon_excitation_microscopy. Accessed 19 Jan 2022
284. Logothetis NK, Pfeuffer J (2004) On the nature of the BOLD fMRI contrast mechanism. Magnetic resonance imaging 22:1517–1531
285. Abel S, Weiller C, Huber W et al. (2015) Therapy-induced brain reorganization patterns in aphasia. Brain 138:1097–1112
286. Saur D, Lange R, Baumgaertner A et al. (2006) Dynamics of language reorganization after stroke. Brain 129:1371–1384
287. Ulm L, Copland D, Meinzer M (2018) A new era of systems neuroscience in aphasia? Aphasiology 32:742–764
288. Betzel RF, Bassett DS (2017) Multi-scale brain networks. Neuroimage 160:73–83
289. Yacoub E, Harel N, Uğurbil K (2008) High-field fMRI unveils orientation columns in humans. Proceedings of the National Academy of Sciences 105:10607–10612

290. Albers AM, Meindertsma T, Toni I et al. (2018) Decoupling of BOLD amplitude and pattern classification of orientation-selective activity in human visual cortex. Neuroimage 180:31–40
291. Wu SW, Pedapati EV (2016) Neuroplasticity Protocols: Inducing and Measuring Change. In: Pediatric Brain Stimulation. Elsevier, pp 45–70
292. Demitri M (2007) Types of brain imaging techniques. Retrieved April 23:2013
293. Li M, Liu F, Jiang H et al. (2017) Long-term two-photon imaging in awake macaque monkey. Neuron 93:1049–1057
294. Ikezoe K, Amano M, Nishimoto S et al. (2018) Mapping stimulus feature selectivity in macaque V1 by two-photon Ca2+ imaging: Encoding-model analysis of fluorescence responses to natural movies. Neuroimage 180:312–323
295. Cubelli R, Bastiani P de (2010) 150 years after Leborgne: why is Paul Broca so important in the history of neuropsychology? Cortex; a Journal Devoted to the Study of the Nervous System and Behavior 47:146–147
296. Baars B, Gage NM (2013) Fundamentals of cognitive neuroscience: a beginner's guide. Academic Press
297. University of St. Andrews Brain imaging techniques—https://www.st-andrews.ac.uk/psychology/research/brainimaging/. Accessed 22 Sep 2018
298. Adolphs R (2016) Human lesion studies in the 21st century. Neuron 90:1151–1153
299. Wurtz RH (2015) Using perturbations to identify the brain circuits underlying active vision. Philosophical Transactions of the Royal Society B: Biological Sciences 370:20140205
300. Kringelbach ML, Jenkinson N, Owen SLF et al. (2007) Translational principles of deep brain stimulation. Nature Reviews Neuroscience 8:623–635
301. Hallett M (2000) Transcranial magnetic stimulation and the human brain. nature 406:147–150
302. Wikipedia Patch clamp—https://en.wikipedia.org/wiki/Patch_clamp. Accessed 10 Nov 2021