University of South Bohemia
Faculty of Science
České Budějovice, Czech Republic

and
Johannes Kepler University
Faculty of Engineering and Natural Sciences
Linz, Austria

Modern approaches of protein classification

Bachelor's thesis

Roland Ackerl

Supervisor: Assist.-Prof. Mag. Dr. Günter Klambauer

Institute for machine learning,
Johannes Kepler University,
Austria
2023

Ackerl, R., 2023: Modern approaches of protein classification, Bc. Thesis, in English. 37 pages, University of South Bohemia, České Budějovice, Czech Republic and Johannes Kepler University, Linz, Austria.

Annotation

The primary goal of this bachelor thesis was to determine the architecture, strengths and weaknesses of AlphaFold and AlphaFold 2 and show a case example of how to classify proteins based on its primary sequence.

Affirmation

I declare that I am the author of this qualification thesis and that in writing it I have used the sources and literature displayed in the list of used sources only.

Linz, 29.05.2023,

Roland Ackerl
………………

Roland Ackerl

Table of Content

List of Figures

List of Tables

1. Abstract

Proteins are large biomolecules that play a fundamental role in every cell in an organism, performing a variety of functions, such as DNA replication, aiding cell movement, and catalyzing biochemical reactions. The process of protein synthesis is complex, and the amino acid sequence encoded in the DNA is transcribed into a mature mRNA that serves as a template for protein synthesis in the ribosomes. However, even with the amino acid sequence, fully understanding the protein's function and activity requires a comprehensive analysis of its structure. Proteins exhibit hierarchical organization of structure, with four distinct levels: primary, secondary, tertiary, and quaternary. Determining protein structure using experimental methods such as X-ray crystallography, fluorescence spectroscopy, and protein nuclear resonance is time-consuming and expensive. Therefore, computational methods have become essential for investigating the structure and function of proteins. Advanced algorithms and modeling techniques allow researchers to predict the spatial arrangement of amino acids in a protein sequence and simulate protein folding and unfolding. Computer-aided approaches are particularly useful for studying hypothetical or artificially designed proteins that may not be amenable to experimental techniques. While experimental and computational methods are critical for understanding protein structure and function, environmental factors such as pH, temperature, presence of ions, salt content, or osmolality significantly impact protein folding and stability. Understanding the relationship between protein structure and function requires a comprehensive analysis that takes into account both the primary amino acid sequence and the environmental factors that affect protein folding and stability.

2. Introduction

Proteins, which are composed of amino acids and linked together through peptide bonds to form a polymer, are among the largest biomolecules found in nature. They play a crucial role in maintaining the functionality of every cell in an organism, performing a diverse range of tasks such as facilitating the transport and detection of various substances, aiding in DNA replication, enabling cell movement, and catalyzing a wide array of biochemical reactions. Despite being the subject of intense scrutiny for decades, only approximately 144,000 protein structures have been determined to date, despite the fact that billions of different proteins are known to exist (wwPDB consortium, 2019).

The process of protein synthesis is a complex one that begins with the DNA sequence encoding the amino acids necessary to form the protein. This sequence is transcribed into a pre-mRNA molecule which is then modified to create a mature mRNA that serves as the template for protein synthesis in the ribosomes. Each base triplet in the mRNA corresponds to a specific amino acid or a start/stop signal, which in turn determines the linear sequence of amino acids that will be synthesized to form the primary structure of the protein.

However, simply knowing the amino acid sequence is not enough to fully understand the function and activity of the protein. When initially synthesized, the protein lacks a stable three-dimensional structure and may be partially folded, random coiled, or completely unfolded. While smaller proteins may spontaneously form their functional conformation, larger and more complex proteins often require support from chaperones (Beissinger M., et al., 1998) to prevent undesirable or even harmful manifestations such as prions (Beringue V., et al., 2008), toxins (Lybchenko Y.L., et al., 2010), toxins, loss of function (Hutt D.M., et al., 2010), or aggregation and accumulation (Bevan-Jones W.R., et al., 2020).

Moreover, even proteins with similar or identical amino acid sequences may have different conformations due to environmental factors such as pH, temperature, presence of ions, salt content, or osmolality (Millan S., et al., 2020). These environmental influences can significantly impact the folding and stability of the protein, leading to differences in function and activity. Thus, understanding the relationship between protein structure and function requires a comprehensive analysis that takes into account both the primary amino acid sequence and the environmental factors that affect protein folding and stability.

Proteins exhibit a hierarchical organization of structure, with four distinct levels. At the most basic level, the primary structure of a protein represents the linear sequence of amino acids that make up the protein. This level of structure is the most detailed and can be directly sequenced to determine the specific amino acid sequence.

The secondary structure of a protein describes the spatial arrangement of a local area, or protein domain, which can take on several configurations such as alpha-helices, beta-sheets, or random coils. The tertiary structure refers to the overall spatial arrangement of a single-chain protein, which incorporates all of the protein domains present. This level of structure is critical to protein function, as it defines the specific three-dimensional conformation of the protein.

The quaternary structure describes the arrangement of multiple macromolecules, which are held together by various intermolecular forces such as van der Waals forces, hydrogen bonds, or Coulomb forces (McNaught A.D., et al., 1997). This level of organization is essential for the function of many proteins, particularly those involved in complex enzymatic reactions or large macromolecular assemblies.

It's important to note that the definition of protein structure outlined above does not encompass more complex structural configurations such as capsomeres in the capsid of viruses or collagen in the collagen fibril. However, this hierarchical organization of protein structure provides a valuable framework for understanding the complex interplay between protein sequence, structure, and function.

There are several experimental techniques available for determining the spatial configuration of a protein, including X-ray crystallography, fluorescence spectroscopy, circular dichroism, protein nuclear resonance (NMR), and others (Kikhney A.G., et al., 2015, Singh B.R., 2000) . However, these methods are often time-consuming, expensive, and require specialized equipment. Some of them rely on mutations, gradual unfolding or folding, and the observation of conformational changes, while others are based on the crystallization of an isolated protein. Additionally, in some cases, it may not be possible to prepare the sample properly for analysis, making experimental approaches less feasible.

To overcome these limitations, computational methods have become an essential tool for investigating the structure and function of proteins. By using advanced algorithms and modeling techniques, researchers can predict the spatial arrangement of amino acids in a protein sequence, and simulate the folding and unfolding of proteins. Computer-aided approaches are particularly useful for studying hypothetical or artificially designed proteins that may not be amenable to experimental techniques.

Overall, both experimental and computational methods play critical roles in advancing our understanding of protein structure and function. By combining these approaches, researchers can gain a more comprehensive understanding of the complex interplay between protein sequence, structure, and function, which is essential for the development of new therapeutics and biomaterials.

Modeling protein folding is a significant challenge, as trying out every possible configuration sequentially would take longer than the age of the known universe to evaluate the true or even the most plausible 3D structure. This is known as Levinthal's Paradox (Levinthal C., 1969), as proposed by Cyrus Levinthal, who was aware that proteins fold spontaneously in a very short period of time in reality. Levinthal suggested that the secondary structure is a thermodynamic state, where the most stable and some metastable forms are located in local minima of the configuration energy. As a result, he proposed the existence of potential pathways that would make the inclusion of unstable states unnecessary, thereby significantly speeding up the computation of the protein folding model.

The many factors that are required to create such a model are difficult to define, too numerous, too complex, or even still unknown. Therefore, various machine learning methods are utilized to recognize and learn patterns and relationships themselves. These methods include common architectures such as attention mechanisms for evaluating context priorities, convolutional layers with pooling to identify peculiarities and reduce data size, and recurrent-based neural networks that can also incorporate unknown features. Other statistical methods, alignment algorithms, and generative adversarial networks (GAN) may also be employed for structural modeling (Wu J., et al., 2016). These methods are often used in combination to increase the accuracy of the predictions.

This bachelor thesis focuses on the theoretical description of solving the protein folding problem, the current state of research, as well as a classification of the protein data set SCOP 1.67 based on the given primary structure using a deep learning approach.

3. Background

3.1 Amino acids and proteins

The building blocks of proteins are amino acids, which are chemical compounds consisting of carbon, nitrogen and oxygen. Their general structure consists of a methylene group (-CH2- for glycine or -CHR- for any other) in the center, one amino group (-NH2 or in substituted form -NR2) on one side and one carboxy group (-COOH) on the other. The carbon of the methylene group adjacent to the carboxy is designated the Cα atom, while the following C atoms on the substituted residue chain are designated in Greek alphabet order up to delta δ. For the protein synthesis α-amino acids are needed, where the amino group is on the Cα. Peptides made of other forms than the α-configuration do not occur in nature, but the amino acids do and serve different purposes. For example, β-alanine is part of pantothenic acid (vitamin B5) and γ-Aminobutyric acid (GABA) acts as a neurotransmitter. Except of glycine, which is achiral, amino acids can have 2 isomers which are basically mirrored configurations of each other: the L- and the D- form. Naturally ocuring amino acids have the L-configuration, but some exceptions like D-Alanin or D-Valin can occur in bacteria, archaeea, fungi and other lower life forms (Wang H., et al., 2014).
For these reasons and for the sake of simplicity, the α- and L-form is omitted from here for amino acids.

Amino acids can be differentiated depending on the side chain (-R) on the Cα. Depending on the nature of these residues, 20 different standard amino acids can be classified. Each of these has a three-letter abbreviation and a one-letter code. In addition, four other abbreviations and codes are used as placeholders for undetermineable or undetectable amino acids in protein sequencing via X-ray cristallography or a chemical approach.

The negatively charged residues include aspartate (Asp) and glutamate (Glu), which have carboxylic acid groups in their side chains. The positively charged residues are arginine (Arg), lysine (Lys), and histidine (His), which have basic amino groups in their side chains.

The hydrophilic amino acids are those that have polar but uncharged side chains. These include serine (Ser), threonine (Thr), asparagine (Asn), glutamine (Gln), and tyrosine (Tyr).

The hydrophobic amino acids are those that have nonpolar side chains, which are typically involved in the formation of the protein's interior. These include alanine (Ala), valine (Val), leucine (Leu), isoleucine (Ile), phenylalanine (Phe), tryptophan (Trp), and methionine (Met).

There are also some amino acids with special properties. Cysteine (Cys) has a thiol (-SH) group in its side chain that can form disulfide bonds with other cysteine residues, stabilizing the protein's structure. Selenium is present in the side chain of selenocysteine (Sec), which is found in some proteins. Proline (Pro) has a cyclic structure that makes it less flexible than other amino acids, and glycine (Gly) is unique in that its side chain is just a hydrogen atom, allowing for more flexibility in certain regions of a protein.

Undetermineable or undetectable amino acids are asparagine or aspartate (Asx), glutamine or glutamate (Glx), leucine or isoleucine (Xle) and any unknowm (Xaa).

Selenocysteine (Sec) and pyrrolysine (Pyl) are used by some eukaryots and several microorganisms, resulting in a total of 26 possible codes to describe a protein.

Furthermore, there are five other Greek letters and the symbols + and -, which act as another placeholder to describe amino acids that are difficult to determine in more detail according to their chemical properties.

When the carboxyl group is joined to the amino residue of different amino acids, a chain connected by a peptide bond is formed. The length of these chains varies, with chains of up to 30 amino acids referred to as peptides and longer, more complex chains referred to as polypeptides or proteins. The sequence of amino acids in a protein is dictated by the genetic information stored in DNA, which is transcribed into RNA and then translated by ribosomes into a protein sequence.

After synthesis, the protein begins to fold or wrap itself into a specific shape, which is essential for its function. The process of protein folding is complex and can involve many post-translational modifications, such as the chemical modification of specific amino acid residues or the binding of cofactors or other proteins. Chaperones can also play a role in helping to fold proteins into their correct shape.

While the energetically optimal and most stable secondary structure of a protein under specific conditions is largely predetermined based on its sequence, there are often metastable states that occur. These states can significantly influence the function and stability of the protein. Therefore, understanding the folding and stability of proteins is critical for understanding their biological function and developing new treatments for diseases.

After some time or under certain circumstances, some proteins must be degraded again. If a protein is in the wrong place in an organism, or it is misfolded, or if an entire cell has to be broken down into its individual parts in the course of apoptosis, proteolysis is mostly unavoidable to maintain a proper function of the whole organism.

A peptide bond itself is considered to be relatively stable. Uncatalyzed hydrolysis in an aqueous environment would take several hundred years, so proteolysis mostly occurs under catalyzed conditions with the help of enzymes called proteases. Proteolysis not only takes place within the cell space or an organism during degradation in the course of protein turnover, but also during food intake, where digestive enzymes break down the proteins into amino acids or shorter polypeptides that are available to the body. There is also proteolytic processing after protein synthesis, a completed reaction with another protein (signaling protein sequences), or after passing through a membrane (target protein sequences), where only a part of the protein is separated and disposed of and the rest can then take its active form.

Conversely, an under- or over-expression of proteases in the wrong places can cause abnormalities in function, diseases or even death of an organism.

Proteins have been the subject of intense scientific scrutiny, but conducting research on them can be extremely challenging. The first hurdle researchers must overcome is locating the targeted protein or at least having an idea of its location within an organism. To facilitate research, genetically modified microorganisms are often created, which enable researchers to study either new proteins or the functions of known ones by slightly modifying a specific position. Once the protein has been located, it must be isolated. However, this process almost always involves the destruction of cells. There are several approaches to isolating proteins, but the most common involves a series of steps that include cytolysis or mechanical disruption of the cell, centrifugation, precipitation, and chromatography or gel electrophoresis. Each step must be precisely tailored to the desired protein and requires various tools to eliminate disruptive factors. For example, when destroying the membranes in the first step, protease inhibitors must be added to prevent active proteases inside the cells from decomposing the proteins to be examined. Centrifugation must also be adjusted precisely as some proteins tend to sink to the bottom while others remain in solution. Some proteins coagulate in an aqueous environment, while others only begin to precipitate above a certain salt content.

Chromatography and gel electrophoresis can be used efficiently only if various properties of a protein are already known to some extent. These methods rely on the separation of proteins based on properties such as size, charge, and hydrophobicity. As such, prior knowledge of these properties is required to select the appropriate chromatography column or gel matrix. Additionally, researchers may also use mass spectrometry to identify and analyze the isolated proteins.

Overall, isolating and studying proteins requires careful consideration of various factors. Successful protein isolation depends on precise tailoring of each step and understanding the properties of the protein being studied. Nonetheless, the insights gained from such research can provide invaluable information about the structure and function of proteins and aid in the development of treatments for various diseases.

Although various methods exist for analyzing the secondary, tertiary, or quaternary structure of proteins, they all require significant effort and resources. As a result, researchers have been striving for decades to develop an in silico solution to simulate proteins and their structures on a computer. With the advent of modern algorithms, researchers can now obtain quick and highly reliable results. Even older approaches, such as Clustal, have been continuously improved over the years. Despite the high accuracy of these results, the complexity of the technologies used has also increased. Purely mathematical approaches are no longer adequate for solving such a complex problem, so self-learning algorithms are being used more frequently. With advancements in computer hardware since the late 2000s, larger architectures are now able to model increasingly complex tasks, leading to significant progress in protein research. These developments have allowed researchers to gain a deeper understanding of the complex nature of proteins and their interactions, paving the way for new discoveries in fields such as medicine, biotechnology, and biochemistry.

3.2 AI: artificial neural network, deep learning and convolutional neural network

Artificial intelligence (AI) is a rapidly growing field within computer science that has achieved remarkable progress in recent years, resulting in significant advances in many areas and greatly accelerating research. The success of AI has led to the introduction of a variety of new terms, which are often used interchangeably despite their differences. The primary objective of AI is to simulate specific decision-making structures and to enable independent problem-solving. Although AI cannot be precisely defined, it is commonly used in research and development as an umbrella term for various subdomains, including machine learning, mathematical logic, statistics, knowledge-based systems, and search and optimization processes. (Bitkom e.V. und Deutsches Forschungszentrum für künstliche Intelligenz, 2017).

Machine learning is a subfield of AI that involves teaching machines to learn and improve from experience without explicit programming. It is based on the idea that a machine can learn from patterns and make decisions based on those patterns, instead of relying on explicit instructions from a programmer. Machine learning is used in a wide range of applications, from natural language processing to image and speech recognition.

Mathematical logic is another important subfield of AI that is focused on developing formal methods to represent and reason about knowledge and information. It provides a rigorous framework for developing logical systems, formal languages, and inference mechanisms that enable machines to reason and make decisions based on logical rules and principles.

Statistics is also an essential component of AI, providing tools and techniques for analyzing and modeling complex data sets. It is used in many areas, including machine learning, data mining, and natural language processing, to extract meaningful insights and make predictions based on statistical patterns and relationships.

Knowledge-based systems are AI systems that incorporate knowledge and expertise in a specific domain, allowing them to reason and make decisions in that domain. These systems are built on top of knowledge representation and reasoning techniques, which enable machines to understand and reason about complex information in a specific domain (Hitzler P., 2022)

One method of machine learning is the artificial neural network (ANN), which, like artificial neurons, is inspired by the biological design. In contrast, the highly complex connections of neurons in the nervous system are not directly simulated, but a simplified model is used to process information for a specific task. The core element is the artificial neuron as a logical threshold value element with several inputs and one output (McCulloch, et al., 1943). If the scalar product of the vector of real-valued weights and the input vector (as well as a bias) exceeds a certain threshold value, an output is generated according to its definition. In the case of the first neuron used in the so-called single layer perceptron, only the state 1 or 0 could be used as a Boolean variable output . The learning effect results from an adjustment of the weights, which is essentially based on the difference between the true and the predicted output (Rosenblatt F., 1958).

However, this simple architecture is only able to resolve linearly separable functions, such as the logical operators AND, OR and NOT, but not XOR. This restriction can be avoided with a multilayer perceptron in which at least one hidden layer of neurons with a subsequent non-linear activation function such as the hyperbolic tangent is used before the output layer. This architecture represents the simplest form of an ANN.

With increasing complexity such as a higher number of hidden layers, a change in the architecture (fully connected or short-cuts) or recurrent neural networks (RNN), other activation functions like rectifier linear unit (ReLU) and further modifications (dropout, attention mechanisms, layer skipping as used in residual neural networks), the models used are summarized under the term “deep learning”. The transition from an ANN to a deep neural network (DNN) is fluid and still not well defined. An ANN with only one hidden layer is basically able to approximate each function to a certain extent and it is conversely denoted to as “shallow”. According to this approach, all ANNs with two or more hidden layers can be referred to as DNN (Goodfellow I., et al., 2016). DNNs seems to prove to be a more accurate model type in most cases, but the number of layers does not always correlate with the prediction performance (Bianco S., et al., 2018).

Convolutional Neural Networks (CNNs) have become a powerful tool for image classification tasks due to their ability to extract relevant features from images. At a high level, a CNN accepts input in the form of a 1D array, 2D or 3D matrix, and applies one or more filter kernels, also known as convolutional filters, over the input. The convolution operation produces a feature map, which represents the presence of a certain feature in the input. The process of moving the filters over the input and computing the inner product at each position is known as convolution.

A CNN typically consists of several layers of convolutional and pooling operations. After the initial convolution operation, each subsequent layer applies a series of convolutional filters to the output of the previous layer. Each filter extracts a different feature from the input, which is used to create a new feature map. This allows the network to learn complex representations of the input, with each layer building on the features learned by the previous layer.

Pooling layers are used to reduce the size of the feature maps generated by the convolutional layers. The most common type of pooling is max pooling, where the maximum value in a sub-region of the feature map is retained and the rest of the values are discarded. This helps to reduce the amount of data that needs to be processed by the network, while still retaining the important features.

The alternating layers of convolution and pooling are repeated several times in a typical CNN, with each subsequent layer building on the features learned by the previous layers. The final layer of the CNN is typically a fully connected layer, where the features generated by the previous layers are flattened and fed into an artificial neural network (ANN) for classification.

Overall, CNNs have shown remarkable performance in image classification tasks and have been widely used in a variety of applications, including self-driving cars, medical image analysis, and facial recognition. (Nagi J., 2011)

3.3 CASP-challenge

Critical Assessment of Protein Structure Prediction (CASP) is a public experiment and competition that has been held biennially since 1994 and is hosted by the University of California, Davis. It is supported and documented by the National Institutes of Health (NIH) as well as the United States National Library of Medicine (NLM) (Protein Structure Prediction Center, US National Institute of General Medical Sciences (NIH/NIGMS, 2007-2020, https://predictioncenter.org/). The aim of this challenge is to compare the methods of different research groups for the structure elucidation of proteins on the basis of the primary structure. The data set to be tested consists of protein structures that have not yet been determined or published which ensures that every participant has the same opportunities.

In order to be able to compare different models, the so-called global distance test (GDT) is used, in which the distances of the calculated alpha carbon positions are compared with the experimentally determined positions at different cutoff values. The unit for the distance is Å r.m.s.d (Angstrom Root-Mean-Square Deviation of Atomic Positions). The respective percentage of the positions within the tolerances is used for further evaluations. Mostly these cutoffs are in the range between 0.5 Å and 10 Å (Zemla A., 2011). For the CASP challenges, the GDT total score (GDT_TS) is used, which is composed of the average of the cutoffs of 1, 2, 4 and 8 Å (Kryshtafovych A., et al., 2007). For comparison, the width of a carbon atom is at approximatly 1.4 Å, so models performing below this value can be considered to be very precise.

3.4 SCOP database

The Structural Classification of Proteins (SCOP) database is a freely accessible database for the classification of protein structures and was created by the Medical Research Council (MRC) Laboratory of Molecular Biology (LMB) in Cambridge, England, and was released 1994 (Andreeva A., et al., 2014). This database aims to establish structural and evolutionary relationships between various proteins, provided the structures are known. Different discrete units, the protein domains, are divided into families and superfamilies depending on their evolutionary divergence and, since the release of SCOP2 in January 2020, further divided into interrelationships and hyperfamilies (Andreeva A., et al., 2020). IUPRs (Intrinsically Unstructured Protein Region), which contain proteins that do not adopt a globular folded structure, have also been added. These either contain a mixture of different conformations or are unstructured until they bind to other macromolecules.

Table 1: Progress of the SCOP classification compared to SCOP version 1.75

Number	SCOP 2 29th June 2022	SCOP 2 January 2020	SCOP 1.75
Folds	1562	1388	1195
IUPRs	24	17	n.a.
Hyperfamilies	22	15	n.a.
Superfamilies	2816	2455	1962
Families	5936	5060	3902
Interrelationships	60	46	n.a.

The main purpose of SCOP was to support structural biologists in the analysis of similar protein structures, but the high number of reciprocal relationships also allowed it to be used to evaluate protein structure comparison and prediction methods.

3.5 Homology detection

Homology refers to the similarity between nucleotide or amino acid sequences of different organisms that have a common ancestor. This similarity is indicative of the fact that the sequences or the evolutionary changes are based on a common inherited sequence. Sequence alignment, which is a computational biology method, is used to identify homologues and classify the proteins being compared. Alignments of several sequences can also be used to display homologous regions (Simon C., 1994).

The most popular and effective methods for alignment include the Smith-Waterman algorithm (Smith T., et al., 1981), FASTA (Pearson W., et al., 1988), and BLAST/PSI-BLAST (Altschul SF., et al., 1990). Most approaches, such as support vector machines (SVMs) (Vapnik V.N., 2000) or position-specific scoring matrices (PSSM) (Stormo G.D., et al., 1982) are resource-intensive and time-consuming, especially when comparing a new sequence against a large database of sequences (Hochreiter S., et al., 2007). Traditional machine learning approaches also require a fixed number of features to represent a sequence, making them less adaptable to new data.

Current approaches, such as neural networks or LSTM (Hochreiter S., et al., 1997), can automatically learn representations from sequence data. These models can interpret unknown or hidden factors that are not considered in a sequence alignment, such as the chemical properties of amino acids and their sequence, as well as the effects of genetic recombination or genetic shuffling on sequence position. Such approaches are more flexible and adaptive to new data, making them useful tools for analyzing protein homology.

3.6 AlphaFold and AlphaFold 2

AlphaFold, a deep learning program developed by Google's DeepMind, has revolutionized protein structure prediction. Prior to its development, the most successful models were based on fragment assembly, where a structure of a short section was successively modified until a structure with low potential was obtained. However, AlphaFold has outperformed these models in every ranking at the

13th CASP challenge in December 2018, achieving a maximum average GDT_TS score of almost 58. Its successor, AlphaFold 2, has achieved an even higher score of 87 at the 14th CASP challenge in July 2020, demonstrating its remarkable accuracy in predicting protein structures. This represents a significant advancement in the field of computational biology and has the potential to revolutionize drug discovery and disease research (Protein Structure Prediction Center, US National Institute of General Medical Sciences (NIH/NIGMS, 2007-2020, https://predictioncenter.org/).

AlphaFold's folding process consists of various steps, starting with a convolutional neural network (CNN) (Schmidhuber J., 2015) that has been trained with the data from the PDB. Based on the distance between the $C_ß$ atoms and extracted features of the multiple sequence alignment (MSA) (EMBL-EBI, Hinxton) generated by an algorithm based on an Hidden Markov Model (HMM) approach called HH-suite3 (Steinegger M., et al., 2019) and PSI-BLAST, a discrete probability distribution from distances and torsion of each amino acid pair in a 64x64 residue region is evaluated. A distogram is generated from this data, in which the generated distances correspond to the real ones. In order to realize structures for the calculation of C_β coordinates, the protein geometry backbone torsion angles (φ, ψ) are used to create a differentiable model $\mathbf{x} = G(\varphi, \psi)$ that was iteratively optimized in 1200 steps by gradient descent (GD) (Ruder S., 2016).

For AlphaFold 2, some changes and extensions were introduced, such as a complete redesign of the neural network-based model (Jumper J., et al., 2021). Like AlphaFold, the network consists of two main steps: the processing of the input and the creation and manipulation of the 3D model. At the beginning, MSAs and paired features are converted into a new representation. For this purpose, an overall assigned loss function and the intermediate losses are iteratively minimized by means of a new equivariate attention architecture. This neural network block is called Evoformer. The Evoformer generates a sequence-by-residues matrix (an $N_{seq} \times N_{res}$ array) using attention-based and non-attention-based components. In the second part, the structure of the proteins is defined by rotation and translation of each residue, whereby the atomic chain structure of the protein is broken down using an equivalent transformer, which enables further refinement. Several similar layers that perform the same task are used throughout the network. This means that the same loss function can be used and the outputs are returned recursively to the same modules, which enables continuous communication between the blocks. Although this increases the training time slightly, the accuracy increases significantly. Both labeled and unlabeled data were used for the training. To do this, the structure of a subset of the data set was first predicted and filtered according to high confidence. This new data set was combined with the one previously used and this architecture was trained again with it. This procedure enables effective use of the unlabeled data with a simultaneous increase in accuracy. To predict the final structure, the paired representations of the 3D backbone structures of the MSA are used. The rotations and translations are limited to their steric possibilities, taking into account the side chains. The peptide bond geometry acts completely unrestricted. However, since this procedure often violates the actually possible arrangements, an additional violation loss term is used, which is minimized by gradient descent in order to reduce steriochemical violations without reducing the accuracy at the same time. Some residues of the MSA are randomly masked and predicted with a transformer similar to the BERT architecture (Devlin J., et al., 2019). This approach leads in the network anticipating phylogenetic and covariation relationships within the protein sequences without having to manually add additional information or hardcode additional statistics.

However, AlphaFold 2 has some limitations. The accuracy drops significantly when the mean alignment depth falls below 30 protein sequences, resulting in a wrong definition of the structure in the early stages of the network, since the information from the MSA is insufficient. To overcome this limitation, additional methods may be required to improve the prediction accuracy.

On the other hand, depths over 100 sequences no longer seem to have a significant influence on accuracy. It was also observed that the network performed better for proteins with fewer intra-chain or homotypic contacts. AlphaFold 2 won the CASP14 challenge in November 2020 with a median GDT score of 92.4 out of 100 and achieved 88% of the predictions with a GDT_TS score of over 80.

Despite this outstanding performance, which was undeniably observed in comparison with other approaches, there are still some limitations of the meaningfulness of the results. When using one of the traditional manual methods to measure a specific protein, the circumstances and conditions (context) at the time of measurement are known. These include, among other factors, the property of a protein not to fold into the next shape until it has bound to another protein or in the presence of specific metal ions, or after chemical modification or formation of a larger complex. When examining known protein sequences, most publications therefore explain in detail why a specific configuration forms or cannot form. However, for yet unknown sequences, AlphaFold 1 and 2 can predict the structure with outstanding accuracy, but this configuration has been stripped of its context.

An example of this is the human 60S ribosomal protein L19, which in nature only retains a specific configuration when bound to the ribosomal RNA complex. Because this structure can only arise during formation of the ribosome, it cannot be observed or replicated in aqueous solution. Since structures from PDB (protein data bank, https://www.rcsb.org/) were used to train AlphaFold 1 and 2, it can be observed in this example that the predicted structure of the protein L19 corresponds to that of the in bound to the ribosome, although it would never have formed on its own under this conditions (Vernon R., 2021).

In summary, AlphaFold 2 is undoubtedly a breakthrough in the field of structural bioinformatics, representing a significant leap in the accuracy and efficiency of protein structure prediction. This model outperforms all previous models and ranks among the best in the industry. One of the most significant changes in AlphaFold 2 is the complete redesign of the neural network-based model. The results obtained with AlphaFold 2 are impressive, and the accuracy achieved in predicting protein structures surpasses that of experimental techniques, which is remarkable. However, the authors of the study admit that there is still room for improvement, and the problem of protein folding is not yet considered entirely solved.

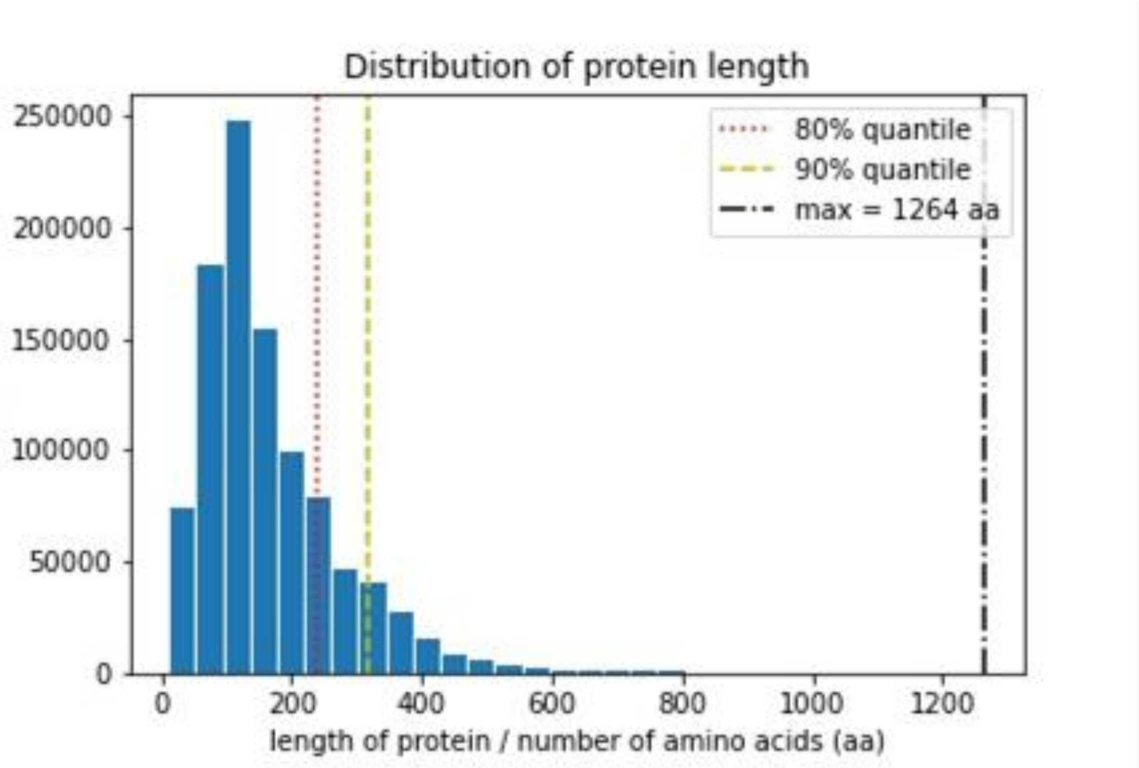
Despite these limitations, the impact of AlphaFold 2 cannot be overstated. The ability to predict protein structures accurately and efficiently opens up new possibilities for drug discovery, protein engineering, and the understanding of diseases at the molecular level. The results obtained with AlphaFold 2 have already paved the way for further research, and it is likely that we will see even more significant advances in the field of structural bioinformatics in the years to come.

4. Methods

4.1 Data preprocessing

The data record consists of one or more FASTA files in a folder with at least one protein sequence each. The name FASTA file contains a label “train” or “test” for use in training or evaluation of the model, as well as the class “pos” and “neg” as a label. Each file is read out and the number of amino acids in the longest sequence is determined as “maxlen”. Although there are only 20 proteinogenic amino acids (aa), other symbols are used as placeholders for two or more as well as unknown ones, resulting in a total of 26 amino acid labels (UPAC_IUB, 1984). Due to these values, a zero padded matrix of the form “maxlen × aa” is created as a template. Each sequence is one-hot coded and inserted in the center of this this template. A positive and a negative labeled set are combined into one for the respective data set for training or testing.

Figure 1: Distribution of protein length in the full dataset (Source: own illustration)



4.2 Model architecture

The model used for the classification of protein sequences is based on a CNN. A dataset generator with 64 sequences each act as input, which is then processed and classified at the end. As a loss function, the binary cross entropy is used as a criterion for the training and AdamW (Loshchilov I., et al., 2019) as an optimizer.

At the beginning, the input is cloned. One set is passed to a fully connected ANN, called the inner ANN, while the other set is passed to a parallel CNN pipeline. The outputs are concatenated, processed by an attention mechanism and passed to another ANN that acts as the final classifier. The inner ANN acts similar to an encoder that automatically recognizes the various properties of the sequences without having to enter them specifically. This information is thus reduced to an array whose length is defined by the number of output nodes. In contrast, the structural information of the positions of the amino acids is processed in the parallel CNN pipeline. The merging of this information allows the downstream attention mechanism to prioritize individual properties and thus make classification easier for the final ANN.

4.2.1 inner ANN

The inner ANN is a fully connected feed forward ANN with two hidden layers. Each layer consists of a linear function with 1024 nodes, a 1D-batch normalization, SELU as an activation function and an alpha dropout of 0.3 (Klambauer G., et al., 2017). The output layer also contains 1024 nodes, without normalization or dropout steps but sigmoid as an activation function.

4.2.2 Convolutions

In the next step of the network, the processed input is passed on to two parallel convolution pipelines, called gouped convolutions (Kirzhevsky A., et al., 2012). These each consist of three blocks with a 1-dimensional (1D) convolution layer, followed by a SELU activation function instead of the commonly used ReLU to avoid the dying ReLU problem (Lu L., et al., 2020), a alpha dropout of 0.3 and a pooling layer. The difference between the two paths lies in the pooling layers implemented: one uses 1D max pooling and the other uses 1D average pooling, both with a kernel size of 3 and a stride of 1.

The convolution layers are configured the same in both paths. The first column block starts with an input from 26 channels, a kernel size of 26 and 26 output channels. Furthermore, the hyperparameter "groups" is set to 26 at this layer only, which is equivalent to a division of the input analogous to 26 convolution steps running in parallel. Thus, a separate filter set is used for each channel and the output is subsequently concatenated. In the second block, 52 channels are output by a kernel of size 26, whereby the stride has been decreased to 1. Then one path is pooled with 1D-max and the other with 1D-average, both with a size of 3. Each output of these blocks is activated with a SELU function and an alpha dropout of 0.3 is applied.

Both paths are flattened and concatenated together with the output of the inner ANN to one array per sequence.

4.2.3 Attention mechanism

Each sequence is processed by a Hopfield network. (Ramsauer H., et al., 2021).

Hopfield networks are feedback networks, but there is only one layer in which every neuron is connected to each other except itself. This layer acts as an attention mechanism, in which patterns in the data set are recognized by setting calculated pooling weights. Since this is determined for each query and a softmax function is applied to the stored patterns, it can be used as a pooling over the sequence.

The input size is given by the length of the assembled arrays from the inner ANN and the grouped convolutions with 3208. The maximum update steps are fixed at 3 and a dropout of 0.2 is used.

4.2.3 Classification ANN

This data set is passed on to a fully connected feed forward ANN with an input layer, 2 hidden layers with 1024 nodes each and an output layer with one output node. In between there are a SELU activation function, a batch normalization layer and a dropout of 0.3. The sigmoid function was used for the output layer to obtain the output in a range between 0 and 1.

5. SCOP Experiments

The following experiments were performed on an ASUS Zenbook UX510U with an Intel Core i7-7500U, an NVIDIA GeForce GTM 960M and a memory of 24 GB. The OS was Ubuntu 20.04.2 LTS, coding was carried out with Python 3.8.5 and CUDA 11.1 in Jupyter 6.3.0. The used machine learning framework was PyTorch 1.8.1. Due to the hardware limitation and the size of the data set, each FASTA file had to be loaded and processed individually during the training. Thus, the data loader was set to a maximum batch size of 64.

5.1 Pre-testing and the final network

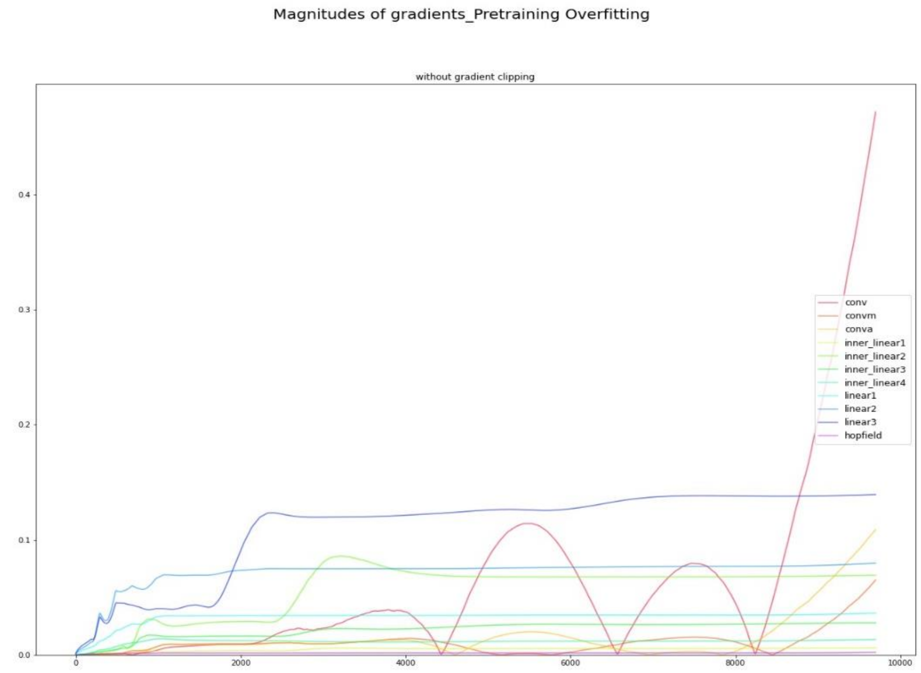
The training was initially carried out with a small sub-sample of the data for 100 epochs, an initial learning rate of 5e-3 and a weight decay of 1e-4 to get a first impression of the behavior of the network. Although overfitting was to be expected with these settings, this was considered irrelevant for the initial trials.

The training set consisted of 2848 positive and 3324 negative samples, the test set consisted of 15 positive and 665 negative samples. To counteract this imbalance, the datasets were duplicated in each run until there was an approximately 1:1 ratio in the dataset.

In each epoch, the current learning rate was multiplied by a factor of 0.99 (Bengio Y. 2019). The mean magnitudes of the gradients of each layer as well as the losses and the accuracies of one training and test set were tracked.

The magnitudes of the gradients were smoothed by a Savitzky–Golay filter (Savitzky A., et al., 1964) with a polynomial of the first degree and a window size of 25 data points in order to represent the long-term trends. All smoothed magnitudes except the last output layer were visualized, with a significant increase in the gradients of the linear layers of the classification ANN groups being recognizable at the beginning. After around 30 epochs, this trend reverses, with the linear layers remaining at a constant level. The magnitudes of the convolutional layers begin to alternately rise and fall, escalating sharply towards the end of the training.

Figure 2: Pre-testing stage: magnitude of gradients(Source: own illustration)



The collected losses and accuracies were also visualized. In the case of the losses, there is a strong fluctuation above and below the starting value of the beginning of the training, which usually indicates that the applied learning rate is too high. The effect is further enhanced by the unbalanced ratio of approximately 1:44 of the positive to negative test samples.

Figure 3: Pre-testing stage: Pretraining accuracy development without gradient clipping (Source: own illustration)

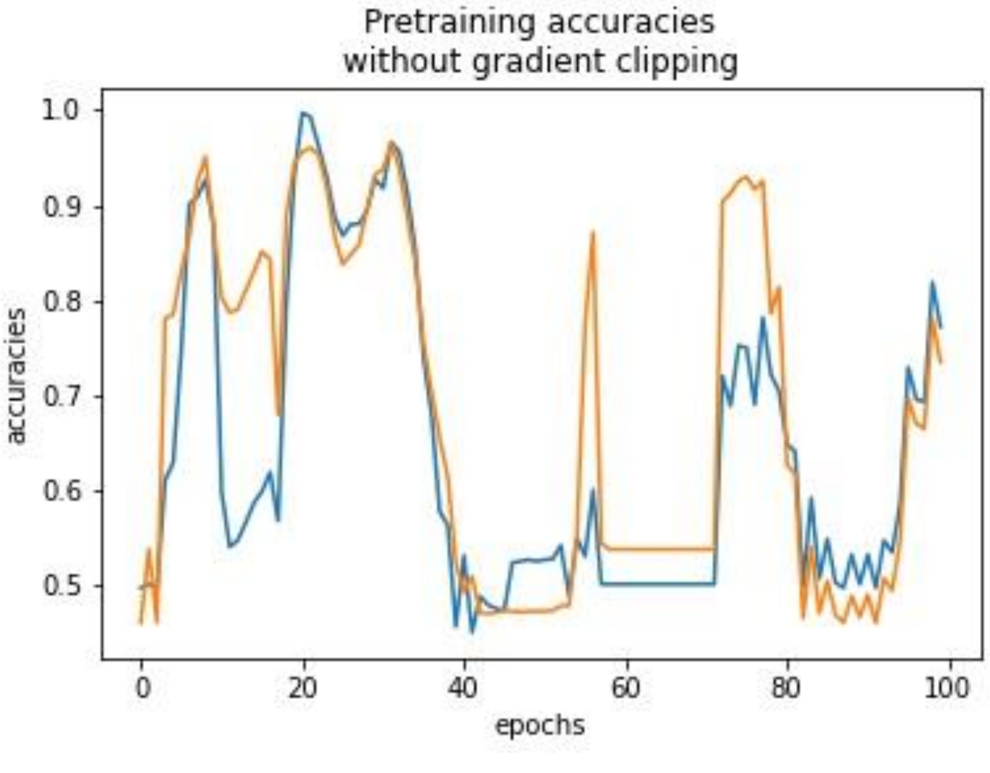
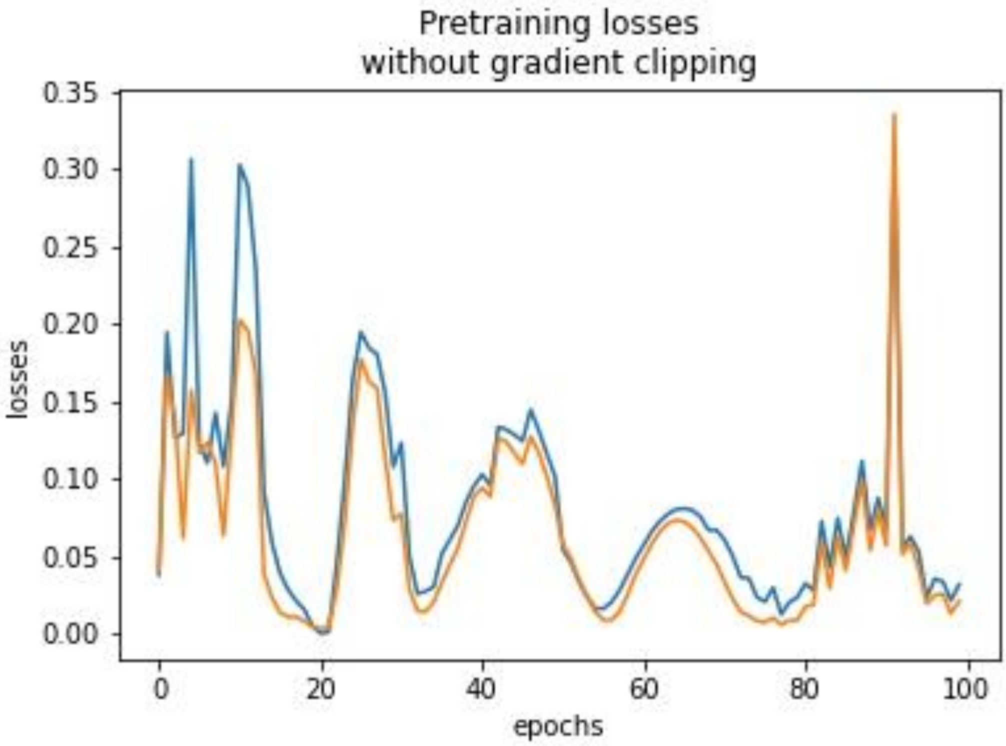


Figure 4: Pre-testing stage: Pretraining loss development without gradient clipping (Source: own illustration)



This pattern was identified as the exploding gradient effect and therefore a gradient clipping step was added in the training phase (Bengio Y., et al., 1994). The decision was made in favor of gradient clipping-by-norm, since with the gradient clipping-by-value the gradients are only clipped within a previously defined threshold and an optimal value can only be approximated carefully through several experiments. With norm clipping, the norm of all gradients together is determined and compared with the previously defined maximum norm threshold. If it is exceeded, the gradient is updated according to

$$g \leftarrow \text{threshold} * g / ||g||$$

where g is the gradient and ||g|| represents the norm of g. Thus, the gradients are always adapted to the extremes taking into account the other gradients. The maximum norm value was determined by testing at 1e-3 with a p-norm of 2.

This behavior is even more evident in the accuracies, which is mainly due to the unequal distribution of the two classes in the test samples. In the case of the losses, a weakening fluctuation can be seen as the learning rate in the last epoch was only around 37% of its value at the beginning.

Almost half of this value was adopted as the starting value, thus, the initial learning rate was reduced to 1e-3.

Again, the test was repeated with these modifications on a new network for 100 epochs to show their possible effect in the visualizations.

Figure 5: Testing stage: magnitude of gradients with gradient clipping (Source: own illustration)

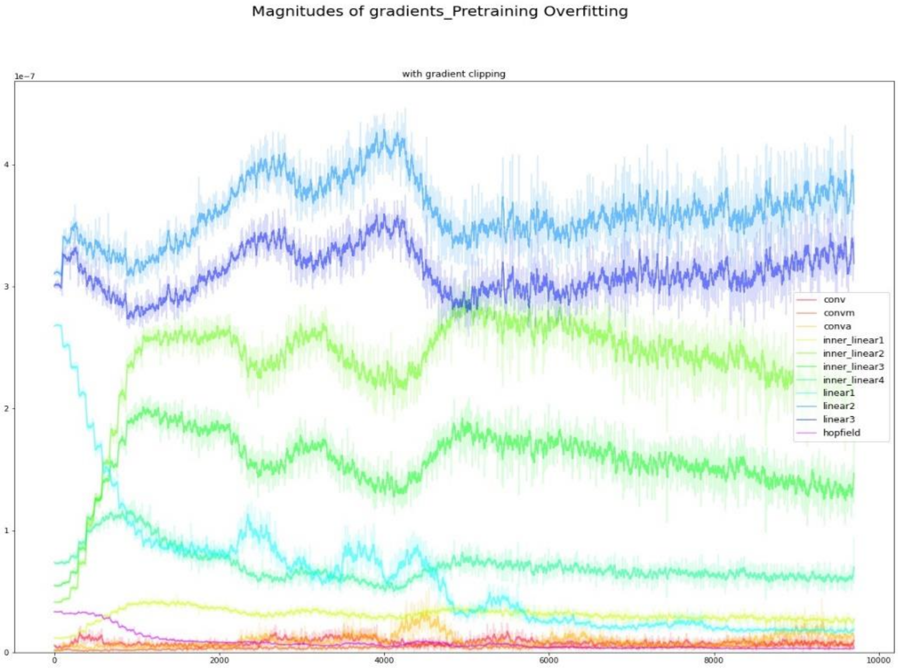


Figure 6: Testing stage: Pretraining accuracy development with gradient clipping (Source: own illustration)

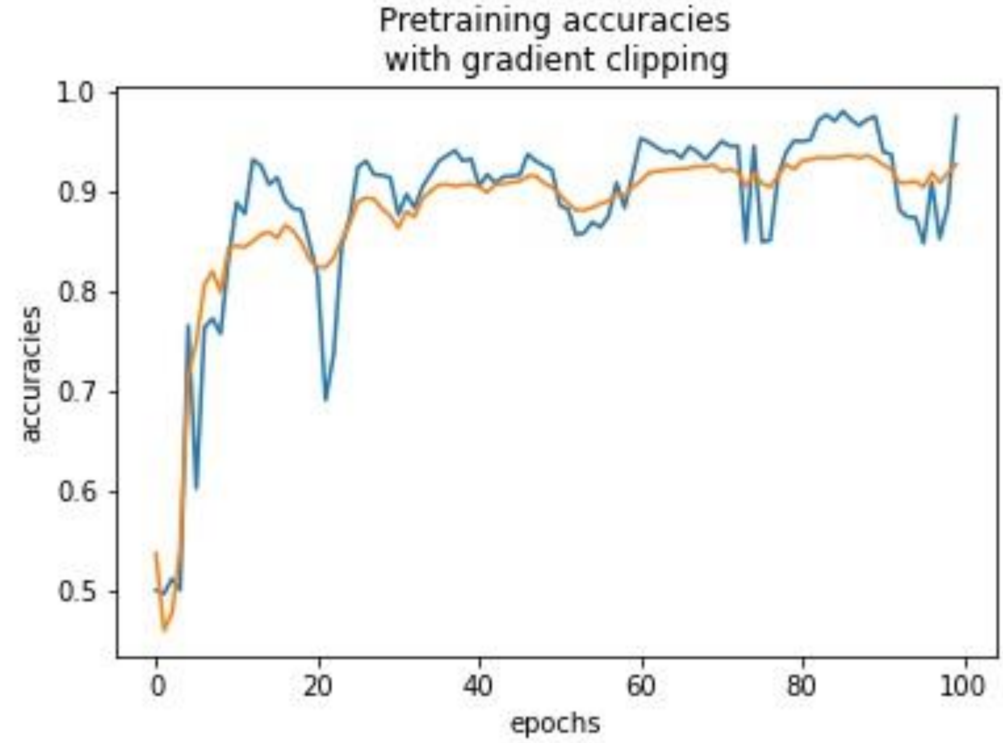
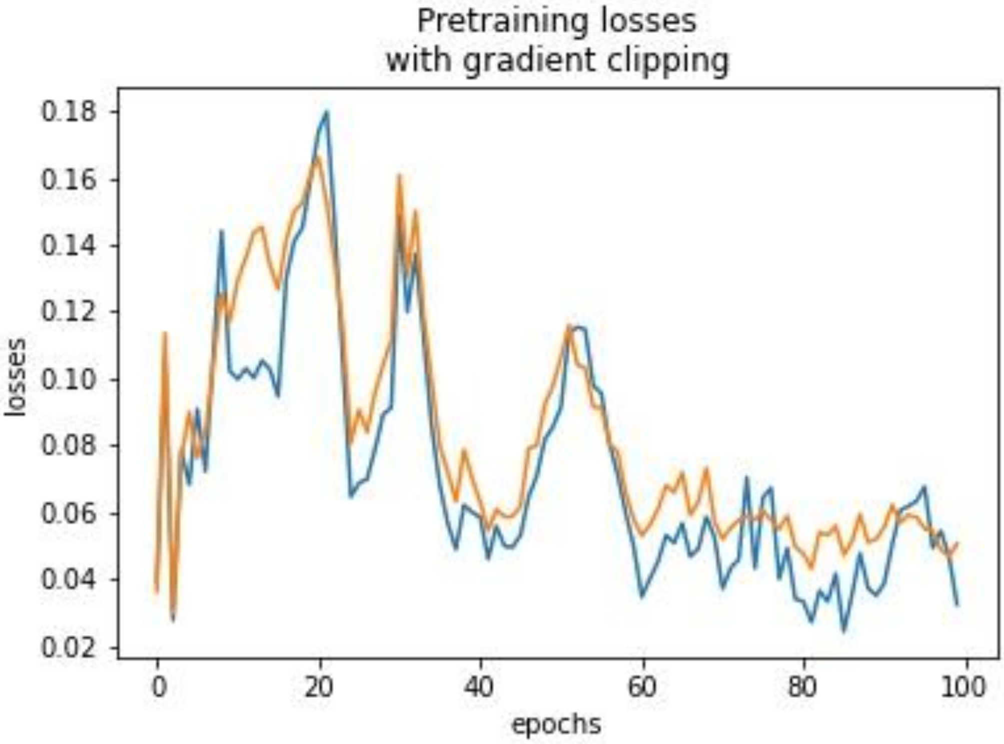


Figure 7: Testing stage: Pretraining loss development with gradient clipping (Source: own illustration)



In terms of losses and accuracies, a small fluctuation was observed above the 70th epoch, with the losses also starting to fluctuate. The magnitudes of gradients also fluctuated too much in the first half of the entire training period. This indicates that the learning rate is too high.

For this reason, the learning rate is multiplied by a factor of 0.99 every 3 epochs. The step decay of the learning rate prevents over-jumping a global minimum of the cost function in order to increase the chance of an ideal optimization (Bengio Y. 2019).
Because the positive and negative samples included in the dataset were not in an ideal 1:1 ratio, some of the minority class were randomly duplicated until balance was reached. This method is known as Random Oversampling (Fernández A., et al., 2018).

The final model is pre-trained with only one of the 409 datasets. 100 epochs were deliberately trained with an initial learning rate of 1e-3 and a weight decay of 1e-4 to an overfitting state with an approximated minimum of the loss value. The binary cross entropy is used as a criterion for calculating losses. After each epoch, the current learning rate was reduced by 1%, resulting in a final learning rate of almost 37% of the initial one. In addition, the model was saved after each epoch.

Figure 8: Training stage: accuracy developement (Source: own illustration)

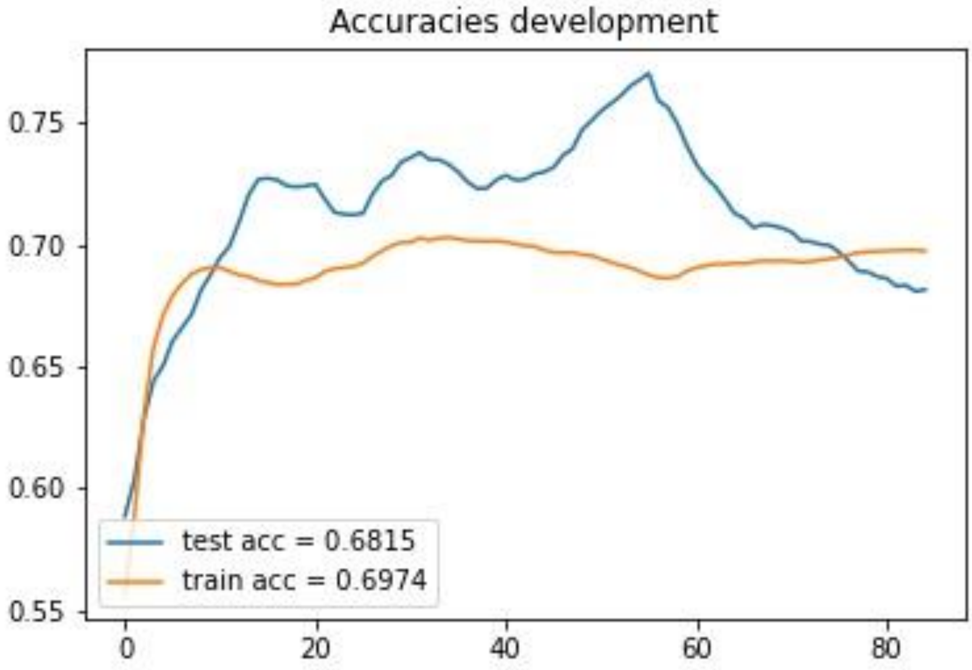
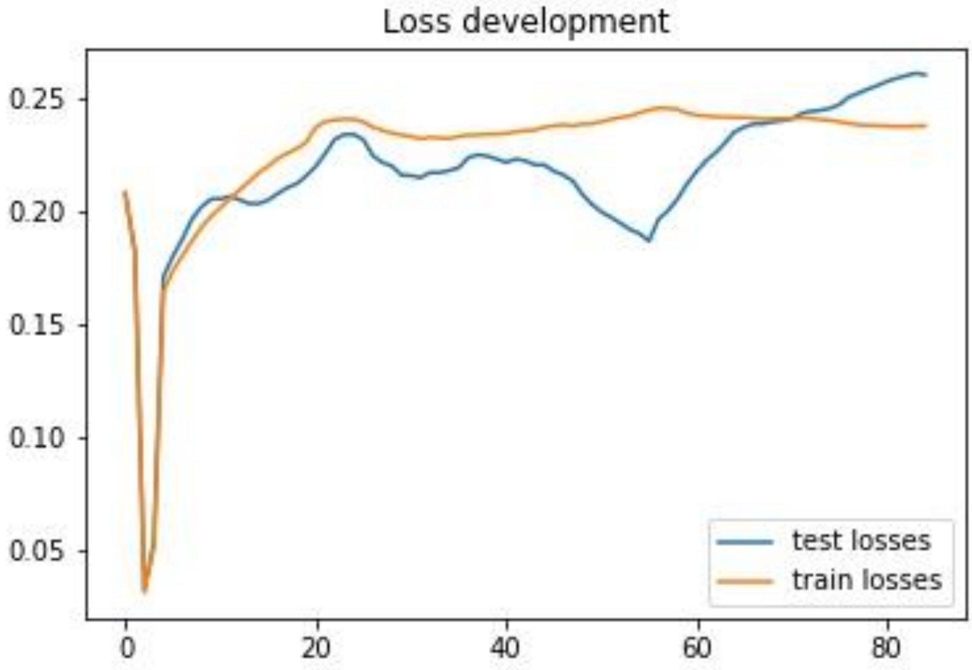


Figure 9: Training stage: loss developement (Source: own illustration)



Since minimal loss of the test set is seen at epoch 56 in the pre-trained model, it is chosen as the basis for the training of the full dataset.

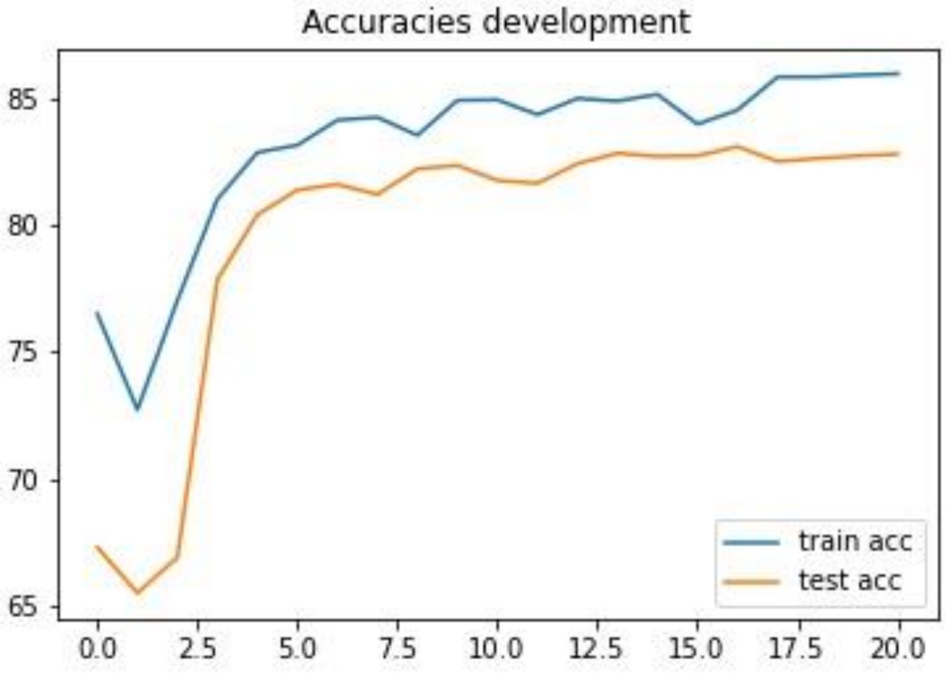
The final training takes place in 20 epochs across all data sets, with the same initial learning rate of 1e-3 and a weight decay of 1e-4. The actual learning rate of the optimizer being multiplied by the factor 0.65 after every 3 epochs. The size of this factor leads to a continuously smaller decreasing range from 1e-3 to 4.9e-5 of the learning rate, in which the learning process shows an acceptable behavior. In this network architecture, the gradients during the training phase were clipped by the maximum gradient norm of 1e-3 and its norm type of L2 (Euclidean norm) to prevent the effect of the exploding gradients (Philipp G., et al., 2018).

6. Results

As expected, the accuracies dropped from 76.5% to 72.7% in the training dataset and from 67.3% to 65.5% in the first epoch when training with the entire dataset.

After the second epoch, it rose sharply again and was able to steadily increase its accuracies to over 84.2% for the train set and 81.6% for the test set up to epoch 6. From there it could only be increased slowly with a few fluctuations before reaching the maximum of 86.0% for the training set and 82.8% for the test set towards the end of training in epoch 20.

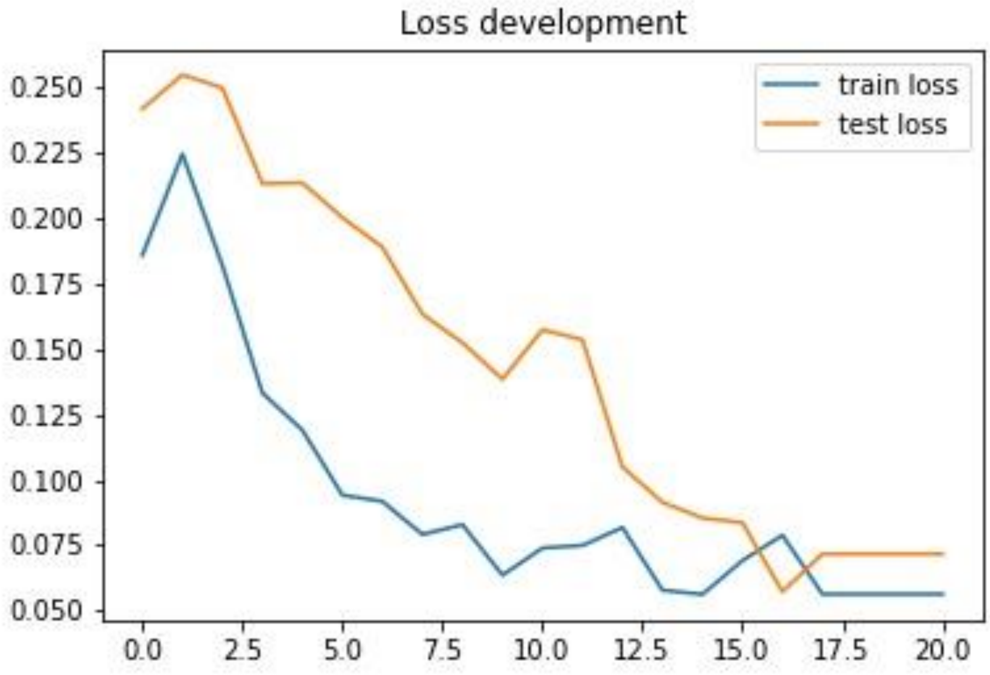
Figure 10: Post-training stage: accuracy developement (Source: own illustration)



Further epochs were omitted since the learning rate had already reached a very low value and, thus, no significant improvement was evident since the 17th epoch.

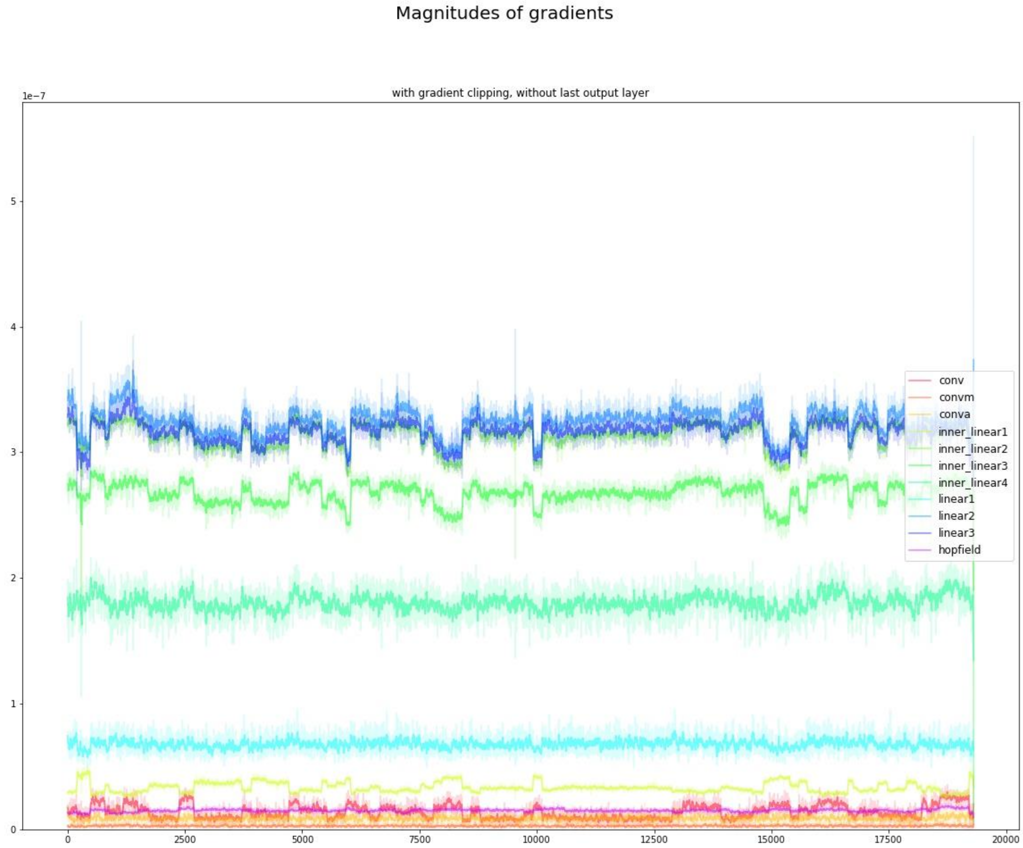
A similar behavior can also be seen in the case of losses. After the losses for the test set increased from 0.242 to 0.255 and for the train set from 0.186 to 0.225 in the first epoch, they then dropped again very quickly. The loss reduction momentum persists until about the 9th epoch, when values of 0.064 for the train set and 0.139 for the test set were reached. Except for a small peak in epoch 16, the losses leveled off at values of 0.056 for the train set and 0.072 for the test set.

Figure 11: Post-training stage: loss developement (Source: own illustration)



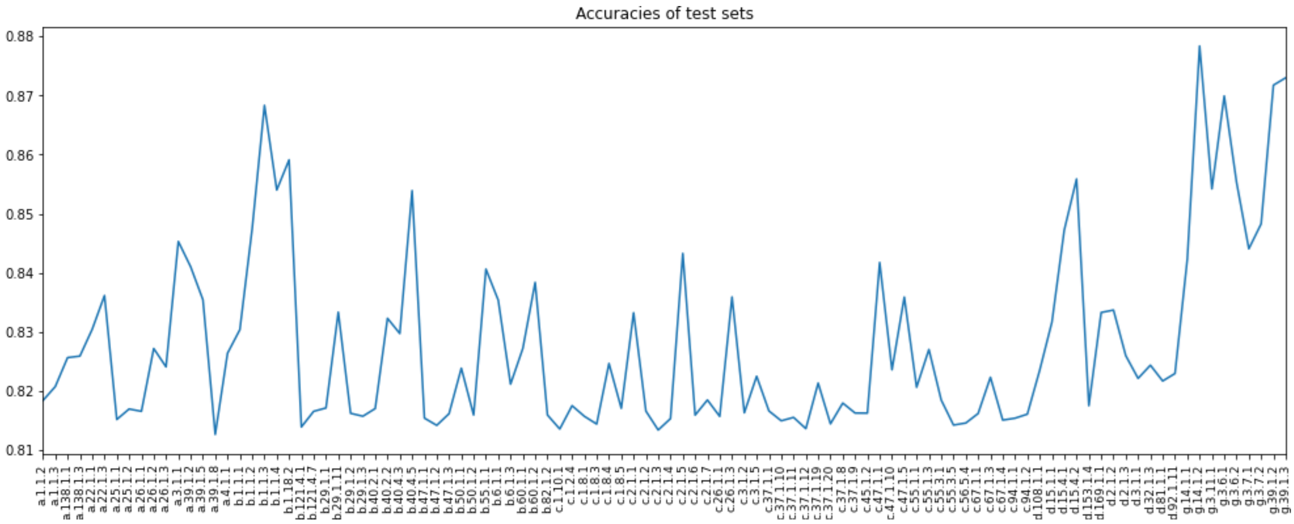
In addition, the magnitudes of the gradients were also visualized in order to observe the activity of the individual layers. Here it can be observed how the gradient clipping contributed significantly to a flattening of the gradients. Furthermore, no drifting of the gradients can be observed, which indicates a constant learning performance of each layer.

Figure 12: Training stage: magnitudes of gradients of final training (Source: own illustration)



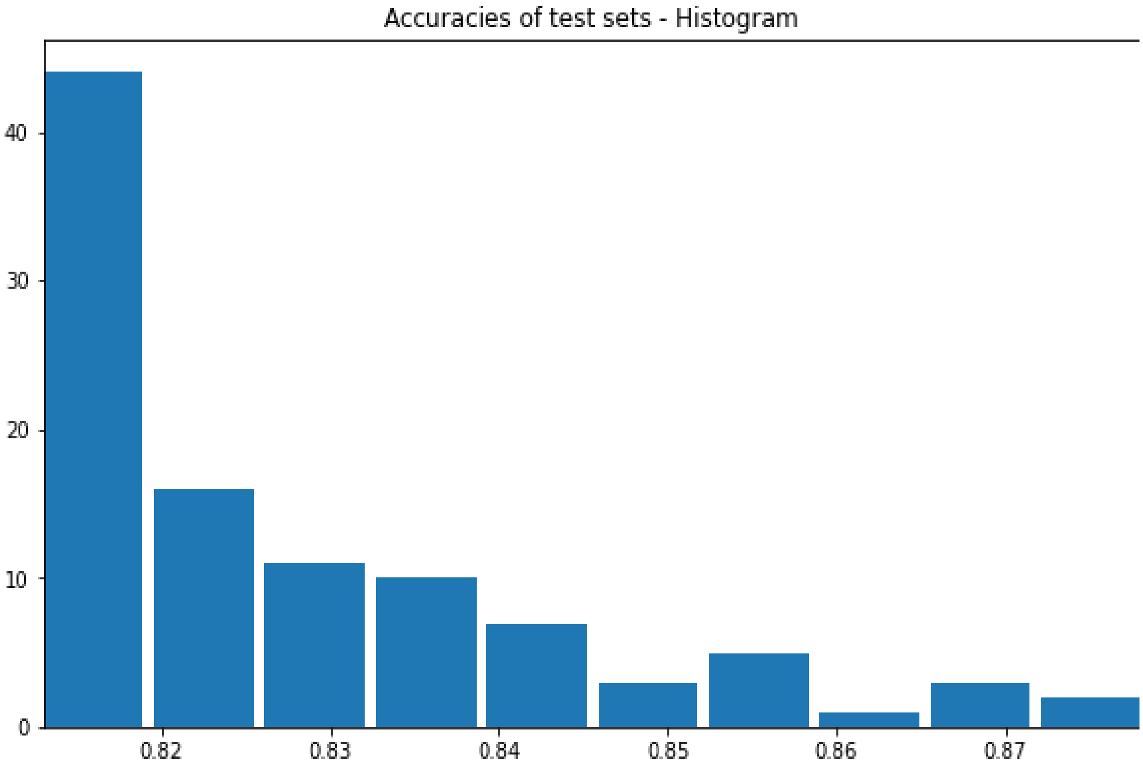
When looking at the data sets of the test sets individually, a fluctuation in accuracy between 81.3% and 87.8% can be seen.

Figure 13: Final accuracies of each test set, numerically ascending (Source: own illustration)



Presented as a histogram, it becomes clearer that the majority of the test sets are in the accuracy range below 82%.

Figure 14: Final accuracies of each test set, Histogram (Source: own illustriation)



7. Conclusion and Discussion

Random oversampling was used to obtain a balanced data set. This simplifies the interpretation of subsequent model statistics, reduces spontaneous accuracy fluctuations and normalizes the costs. A disadvantage of this method is that the model focuses more on the duplicated samples and a class is therefore partially overfitted. Hybrid methods of oversampling and undersampling as well as more sophisticated approaches such as Synthetic Minority Oversampling Technique (SMOTE) (Chawla N.V., et al., 2002) would be more suitable for better balancing.

Another drawback lies in the architecture of the model. Since an ANN was used parallel to the convolution pipeline to determine some hidden features, the number of nodes of its input layer is already defined in advance. Firstly, this leads to a limitation of the maximum length of the entered protein sequence. New samples that exceed this length cannot be processed by this model. Secondly, all sequences shorter than this length in the given dataset contain a high number of leading and tailing zeros, which can lead to a reduction in generalization since the information is placed in the center of the sequence. A workaround could be to place the shorter samples at a random position (data augmentation). If repeated several times, this can also be used to artificially expand the data set. In general, it would be ideal to take an approach that is capable of processing a sequence directly, regardless of length. For example, an LSTM or a Transofrmer (Vaswani A., et al., 2017) would be better suited for this.

While the grouped convolutions are able to learn a broader range of low- and high-level features (Krizhevsky A., et al., 2017), the model would benefit from an even further extension of this block due to the existing hardware limitation.

8. References

Abraham S., Golay M. J. E., Smoothing and Differentiation of Data by Simplified Least Squares Procedures, (1964), Anal. Chem., 36(8), pg. 1627–1639, doi: 10.1021/ac60214a047

Altschul S.F., et al. Basic local alignment search tool, (1990), J. Mol. Biol, vol. 215, pg. 403-410

Andreeva A., Howorth D., Chothia C., Kulesha E., Murzin A., SCOP2 prototype: a new approach to protein structure mining, (2014), Nucl. Acid Res., 42 (D1): D310-D314

Andreeva A., Kulesha E., Gough J., Murzin A., The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures, (2020), Nucl. Acid Res., 48 (D1): D376-D382

Beissinger M., Buchner J., How chaperones fold proteins, (1998), Biol Chem., 379(3):245-59. PMID: 9563819.

Bengio Y., Practical recommendations for gradient-based training of deep architectures, (2019), v2, doi: 10.48550/arXiv.1206.5533

Bengio, Y., Simard, P., and Frasconi, P., Learning long-term dependencies with gradient descent is difficult, (1994), IEEE Transactions on Neural Networks, 5(2), pg. 157–166.

Beringue V., Vilotte J.L., Laude H., Prion agent diversity and species barrier, (2008), Institut National de la Recherche Agronomique (INRA), UR892, Virologie et Immunologie Moléculaires, F-78350 Jouy-en-Josas, France, INRA, UR339, Génétique Biochimique et Cytogénétique, F-78350 Jouy-en-Josas, France, doi:10.1016/B978-0-12-801238-3.02648-9

Bevan-Jones W.R., Cope T.E., Jones P.S., Kaalund S.S., Passamonti L., Allinson K., Green O., Hong Y.T., Fryer T.D., Arnold R.D., Coles J.P., Aigbirhio F.I., Larner A.J., Patterson K., O'Brien

J.T., Rowe J.B., Neuroinflammation and protein aggregation co-localize across the frontotemporal dementia spectrum, (2020), Brain, Volume 143, Issue 3, pg. 1010–1026, doi: 10.1093/brain/awaa033

Bianco S., Cadene R., Celona L., Napoletano P., Benchmark Analysis of Representative Deep Neural Network Architectures, (2018), IEEE Access, vol. 6, pg. 64270-64277, doi: 10.1109/ACCESS.2018.2877890.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority oversampling technique, (2002), J. Artif. Intell. Res. 16, pg. 321–357

Devlin J., Chang M.W., Lee K., Toutanova K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, (2019), v2, arXiv:1810.04805

EMBL-EBI, Wellcome Genome Campus, Hinxton, Cambridgeshire, CB10 1SD, UK., https://www.ebi.ac.uk/Tools/msa/

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., Herrera, F., Learning from Imbalanced Data Sets, (2018). Page 82, doi:10.1007/978-3-319-98074-4

Goodfellow I., Bengio Y., Courville A., Deep Learning (2016), MIT Press, http://www.deeplearningbook.org

Hitzler P., Neuro-Symbolic Artificial Intelligence: The State of the Art, (2022), pg. 83-97, IOS Press BV, Amsterdam, Netherlands, ISBN 978-1-64368-244-0

Hochreiter S, Schmidhuber J. Long short-term memory, (1997), Neural Comput, vol. 9, pg. 1735-1780

Hochreiter S., Heusel M., Obermayer K., Fast model-based protein homology detection without alignment, (2007), Bioinformatics, Volume 23, Issue 14, pg. 1728–1736, doi: 10.1093/bioinformatics/btm247

Hutt D.M., Herman D., Rodrigues A.P.C., Noel S., Pilewski J.M., Matteson J., Hoch B., Kellner W., Kelly J.W., Schmidt A, Thomas P.J., Matsumura Y., Skach W.R., Gentzsch M., Riordan J.R., Sorscher E.J., Okiyoneda T., Yates III J.R., Lukacs G.L., Frizzell R.A., Manning G., Gottesfeld

J.M., Balch W.E., Reduced histone deacetylase 7 activity restores function to misfolded CFTR in cystic fibrosis, (2010), Nature Chemical Biology volume 6, pg. 25–33, doi: 10.1038/nchembio.275

IUPAC, McNaught A. D., Wilkinson A., Compendium of Chemical Terminology, 2nd ed. (the "Gold Book"), (1997), Blackwell Scientific Publications, Oxford, ISBN 0-9678550-9-8, Online version (2019-) created by S. J. Chalk, doi: 10.1351/goldbook

Jumper J., Evans R., Pritzel A., et al., Highly accurate protein structure prediction with AlphaFold, (2021), Nature 596, 583–589. doi:10.1038/s41586-021-03819-2

Kikhney A. G., Svergun D. I. (2015), A practical guide to small angle X-ray scattering (SAXS) of flexible and intrinsically disordered proteins, FEBS letters, 589(19), pg. 2570-2577.

Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S., Self-Normalizing Neural Networks, (2017), Advances in Neural Information Processing Systems (NIPS)

Krizhevsky A., Sutskever I., Hinton G. E., ImageNet classification with deep convolutional neural networks, (2017), Communications of the ACM, 60(6), pg. 84–90. doi: 10.1145/3065386

Krizhevsky A., Sutskever I., Hinton G.E., ImageNet Classification with Deep Convolutional Neural Networks, (2012), Advances in Neural Information Processing Systems 25 (NIPS 2012)

Kryshtafovych A., Prlic A., Dmytriv Z., Daniluk P., Milostan M., Eyrich V., Hubbard T., Fidelis K., New tools and expanded data analysis capabilities at the Protein Structure Prediction Center, (2007), Proteins. 69 Suppl 8: pg. 19–26. doi: 10.1002/prot.21653. PMC 2656758. PMID 17705273.

Künstliche Intelligenz, (2017), Bitkom e.V. und Deutsches Forschungszentrum für künstliche Intelligenz, pg. 28, retrieved 15. Mai 2022, https://www.dfki.de/fileadmin/user_upload/import/9744_171012-KI-Gipfelpapier-online.pdf

Levinthal C., Mossbauer Spectroscopy in Biological Systems: Proceedings of a meeting held at Allerton House, Monticello, Illinois, (1969), University of Illinois Press, pg. 22-24

Loshchilov I., Hutter F., Decoupled Weight Decay Regularization, (2019), v3, https://arxiv.org/abs/1711.05101

Lu L., et.al., Dying ReLU and Initialization: Theory and Numerical Examples, (2020), Communications in Computational Physics, Vol 28, pages 1671–1706, doi:10.4208/cicp.oa-2020-0165

Lyubchenko Y.L., Kim B.H., Krasnoslobodtsev A.V., Yu J., Nanoimaging for protein misfolding diseases, (2010), WIREs Nanomedicine and Nanobiotechnology, Volume 2, Issue 5, pg. 526-543, doi: 10.1002/wnan.102

McCulloch W.S., Pitts W., A logical calculus of the ideas immanent in nervous activity, (1943), Bull Math. Biophys. 5, pg. 115-133

Millan S., Swain B.C., Tripathy U., Mishra P.P., Sahoo H., Effect of micro-environment on protein conformation studied by fluorescence-based techniques, (2020), Journal of Molecular Liquids, Volume 320, Part B, doi: 10.1016/j.molliq.2020.114489

Nagi J., Ducatelle F., Di Caro G. A., Cireşan D., Meier U., Giusti A., Gambardella L. M., Max-pooling convolutional neural networks for vision-based hand gesture recognition, (2011), 2011 IEEE international conference on signal and image processing applications (ICSIPA), pg. 342-347

Pearson W., Lipman D., et al., Improved tools for biological sequence comparison, (1988), Proc. Natl Acad. Sci, vol. 85, pg. 2444-2448

Philipp G., Song D., Carbonell J.G., The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions, (2018), https://arxiv.org/abs/1712.05577

Protein Structure Prediction Center, US National Institute of General Medical Sciences (NIH/NIGMS), 2007-2020, https://predictioncenter.org/

Ramsauer H., Schäfl B., et al., Hopfield Networks is All You Need, (2021), arXiv:2008.02217, https://github.com/ml-jku/hopfield-layers

Rosenblatt, F., The perceptron: A probabilistic model for information storage and organization in the brain. (1958), Psychological Review, 65(6), pg. 386–408. doi: 10.1037/h0042519

Ruder, S., An overview of gradient descent optimization algorithms, (2016), ArXiv Preprint ArXiv:1609.04747.

Schmidhuber J., Deep learning in neural networks: An overview, (2015), Neural Networks, Volume 61, pg. 85-117, ISSN 0893-6080, doi: 10.1016/j.neunet.2014.09.003

Simon C., Frati F., Beckenbach A., Crespi B., Liu H., Flook P., (1994), Evolution, weighting, and phylogenetic utility of mitochondrial gene sequences and a compilation of conserved polymerase chain reaction primers. Annals of the entomological Society of America, 87(6), 651-701.

Singh B. R., Basic aspects of the technique and applications of infrared spectroscopy of peptides and proteins, (2000), ACS Symposium Series 750, pg. 2-37

Smith T., Waterman M., et al., Identification of common molecular subsequences, (1981), J. Mol. Biol., vol. 147, pg. 195-197

Steinegger M., Meier M., Mirdita M., Vöhringer H., Haunsberger S.J., Söding J., HH-suite3 for fast remote homology detection and deep protein annotation, (2019), BMC Bioinformatics, 473. doi: 10.1186/s12859-019-3019-7

Stormo G.D., Schneider T.D., Gold L., Ehrenfeucht A., Use of the ‘Perceptron’ algorithm to distinguish translational initiation sites in E. coli, Nucleic Acids Research, (1982), Volume 10, Issue 9, pg. 2997–3011, doi: 10.1093/nar/10.9.2997

UPAC-IUB Joint Commission on Biochemical Nomenclature, Nomenclature and Symbolism for Amino Acids and Peptides. (1984), Eur. J. Biochem. 138: pg. 9-37

Vapnik V.N., The Nature of Statistical Learning Theory, (2000), Statistics for Engineering and Information Science, 2nd edition, New York, Springer Verlag, ISBN: 0-387-98780-0

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I., Attention is All you Need, Advances in Neural Information Processing Systems, (2017), Volume 30, doi: 10.5555/3295222.3295349

Vernon R., AlphaFold 2: What No One Is Talking About, (2021), Cyclica, retireved on 16 January 2022, https://blog.cyclicarx.com/limitations-of-alphafold

Wang H., Fewer D.P., Holm L., Rouhiainen L., Sivonen K., Atlas of nonribosomal peptide and polyketide biosynthetic pathways reveals common occurrence of nonmodular enzymes, (2014), University of California, San Diego, La Jolla, CA, doi: 10.1073/pnas.1401734111

Wu J., Zhang C., Xue T., Freeman W.T., Tenenbaum J.B., Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling, (2016), 29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain

wwPDB consortium, Protein Data Bank: the single global archive for 3D macromolecular structure data, (2019), Nucleic Acids Research, Volume 47, Issue D1, pg. D520–D528, doi: 10.1093/nar/gky949

Zemla A., Local-Global Alignment for Finding 3D Similarities in Protein Structures, (2011), Lawrence Livermore National Security, LLC