ECM3401: Literature Review

Attentive Neural Protein Structure Prediction

Prediction of Protein Tertiary Structure from Physiochemical Sequence Information
using Encoder-Decoder Architecture with Attention

Jack Marsh

21 November 2018

Abstract

Proteins are abundant in a virtually all biological processes within the cells of, not just humans, but all living organisms. Sequences of amino acids undergo a spontaneous disorder to order transition into their final 3-dimensional structure, called folding. Protein structure is important to determine as the structure dictates the function of the protein. This paper discusses current state-of-the-art methods of computationally determining structure before proposing a novel method that utilises the encoder-decoder neural network architecture with attention, which has had unrivalled success in various fields of natural language processing. The paper explores the underlying theory of the components that make up the encoder-decoder, displaying their suitability to resolving the protein folding problem. It hopes to transfer the success of encoder-decoders to the field of bioinformatics by providing a more general solution between molecular dynamics simulations and template-based modelling.

I certify that all material in this dissertation which is not my own work has been identified.

1 Introduction

Proteins are ubiquitous in a vast number of biological processes in the cells of all organic organisms. They have various functions ranging from antibodies that bind to foreign particles, such as viruses and bacteria, enzymes that acts as catalysts in chemical reactions within cells or messengers that transmits signals to coordinate biological processes. They also provide structural support for cells and can transport and store atoms and molecules within them [1]. Protein folding is important because the structure of a protein dictates its function. Determining protein structure is an extremely active field of research that remains elusive, captivating the minds of many researchers for over half a century. The fundamental principles of folding have practical applications in the exploitation of the advances in genome research, in the understanding of different pathologies and in the design of novel proteins with special functions, just to name a few. The expensive nature of experimental methods gives rise to the need for computational methods of discerning protein structure. Recurrent neural networks with long short-term memory units have had unprecedented success in the field of natural language processing, with their encoder-decoder derivative leading efforts in neural machine translation. This research will aim at applying the recent success of neural networks to the field of bioinformatics in an attempt to predict protein structure with an encoder-decoder architecture.

Section 2 covers the relevant literature around the biological advances of the past 70 years that have lead us to the position we are currently in whereby we can pose the question of whether tertiary structure can be accurately predicted by computational methods.

Section 3 focuses on the field of bioinformatics. It highlights the motivations towards computational methods over experimental ones while revealing the different approaches for determining protein structure *in silico* for the current state-of-the-art methods.

Section 4 investigates pioneering research into neural networks. Here we delve into the evolution of recurrent architectures, exploring the underlying theory in hope of shedding light on their suitability to the task at hand.

Section 5 outlines the project specification. It expresses the aims and objectives of the project, touching on related research and discussing the proposed method of implementation. It also indicates how the model error will be assessed.

2 Advances in Biology

2.1 The Folding Process

Under physiological conditions, a protein undergoes a spontaneous disorder to order transition called folding. Proteins are linear polymers built from a chain of 20 possible amino acids that consist of backbone atoms and a side chain, which differentiates them. The side chain determines the hydrophobicity, charge and polarity of the amino acid, which are all vital physiochemical features in the folding process. R is the general designation for an amino acid side chain, which is shown in Fig. 1 by the molecules coming off the C_α atom. There are 3 distinct stages to the folding process, identified by the structures they create.

2.1.1 Primary Structure

Protein molecules are made up of chains of amino acids that are linked by peptide bonds; referred to as polypeptides. The polypeptide backbone is a repeating sequence along the polypeptide chain. Peptide bonds are formed by the nucleophilic addition-elimination reaction between the carboxyl group of one amino acid and the amino group of another amino acid. The electron pair on the amino group of the second amino acid forms a covalent bond with the carbonyl carbon of the first amino acid, sharing an electron and giving off water in the process through condensation [2].

X-ray diffraction has found that these peptide bonds are rigid and planar. Thus, for a pair of amino acids linked by a peptide bond, six atoms lie in the same plane (Fig. 1), which, as the paper discusses later, provides a useful simplification to our tertiary structure prediction model [3]. The start of a polypeptide will always be a nitrogen atom, the N-terminus, and the end of a polypeptide will be a carbon atom, the C-terminus. Once in the chain, each amino acid is called a residue.

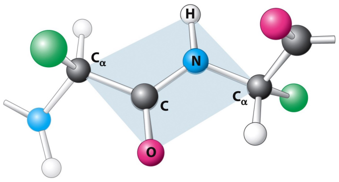


Figure 1: Peptide Bond

2.1.2 Secondary Structure

The next step in the folding process is typically towards two motifs, α-helices and β-sheets. In the spring of 1948, Pauling knew how primary structure formed as he had previously discovered, with Robert Corey, that peptide bonds are planar [4]. Yet there was no knowledge around how linear polymers folded into 3D structures, although Pauling correctly speculated hydrogen bonding was involved. Secondary structure is primarily determined by backbone interactions such as hydrogen bonding. Fig. 2 shows these secondary structures. α-helices are formed by the hydrogen bonding of the backbone into spiral shapes where the hydrogen bonds run up and down, stabilising the structure. There are two kinds of β-sheets. Parallel β-sheets and anti-parallel β-sheets. When the N-terminus and C-terminus are aligned in the same direction it is parallel and when they alternate it is anti-parallel. Rost authored a paper in which he proposed, for computational methods, the accuracy of prediction of protein secondary structure has a theoretical limit of 88% [5]. This was because experimental secondary structure assignment differed by 12% for structural homologs. Thus, he proposed this as the upper limit for computational methods, doubting even that *ab initio* methods will ever be more accurate.

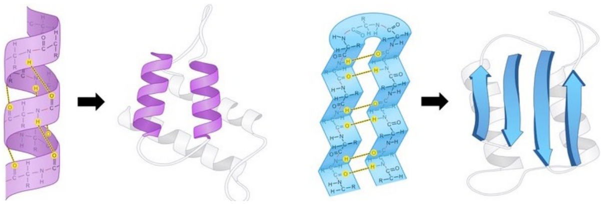


Figure 2: α-helix and β-sheet

2.1.3 Tertiary Structure

Finally, the tertiary structure of a protein is also stabilised by hydrogen bonds but new interactions are now introduced to the folding process. The major driving force is the polarity of a residue's side chain. Polarity determines how hydrophobic or hydrophilic the residue is. The "hydrophobic effect" dictates that in aqueous environments, such as cells, non-polar R groups are hydrophobic meaning they tend to cluster in the centre of globular proteins, away from the aqueous surroundings. On the other hand, polar R groups favour interactions with the surrounding solution and so tend towards the surface of a protein [6]. There are five polar amino acids that can carry a charge indicated by the residue's pK_a value that are split into two groups, basic and acidic. Basic bear a full positive charge while acidic bear a full negative charge, at the normal physiological pH. If an acidic residue comes into close proximity with a basic residue during the folding process, ionic bonds can form as they have opposing charges. The van der Waals force is an attractive force that balances the repulsion that increases when two non-polar atoms approach each other. As a result, there is a distance at which repulsive and attractive forces precisely balance, stabilising the hydrophobic core of proteins [7]. Disulphide bonds are sulphur-sulphur covalent bonds that form between the thiol groups of two cysteine residues. These are very strong intra-molecular bonds that play an important role in determining the structure of a protein.

2.2 Anfinsen's Dogma

In 1973, Anfinsen conducted a Nobel prize-winning experiment that lead to what is now referred to as the "Thermodynamic Hypothesis". The hypothesis states that "the three-dimensional structure of a native protein in its normal physiological milieu (solvent, pH, ionic strength, presence of other components such as metal ions or prosthetic groups, temperature and other) is the one in which the Gibbs free energy of the whole system is lowest; that is, that the native confirmation is determined by the totality of interatomic interactions and hence by the amino acid sequence, in a given environment" [8]. Throughout the 1950s, Anfinsen performed a series of "unscrambling" experiments in support of his eventual hypothesis, that showed primary structure determines the confirmation of a protein [9, 10]. Denaturing agents were mixed with Ribonuclease to create the denatured enzyme by breaking it's 4 disulphide bonds. When removing one of the denaturing agents the enzyme refolded to an inactive "scrambled" enzyme with incorrect disulphide bonds. Upon adding back trace amounts of the denaturing agent, the incorrect disulphide bonds broke causing it to eventually refold to its native state. This showed that for small globular proteins in their standard physiological environment, native structure is determined purely by the proteins amino acid sequence [11].

2.3 Ramachandran Plots

At the time that Anfinsen was conducting these experiments various types of polypeptide chain configurations had been proposed, notably α-helix, but there was no analytical method of writing the configurations. G. N. Ramachandran, was a physicist who devised the Ramachandran Plot in 1963 [12]. The peptide bond between the C_α and N only exists in two discrete configurations, trans and cis isomers. 99.9% of peptide bonds adopt the trans isomer with cis isomers reflecting trans isomers at exactly π radians, denoted by the ω angle. The Φ and Ψ angles are the angle of rotation about the single bonds $N - C_\alpha$ and $C_\alpha - C$ respectively. Plotting Φ against Ψ for all residues in a chain show the allowed confirmations of proteins while indicating secondary structures, displayed in Fig. 3 [13]. It is clear from the Ramachandran Plot that not all rotations are allowed, and this is due to steric hindrance.

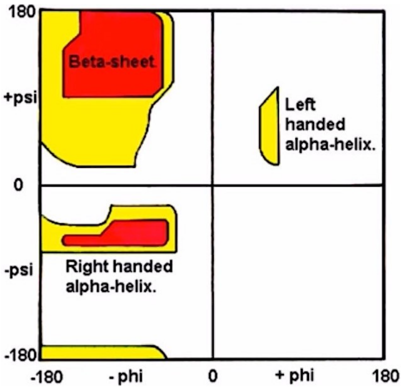


Figure 3: Ramachandran Plot

3 Advances in Bioinformatics

3.1 Experimental Methods

Several experimental methods are currently used to determine protein structure, including x-ray crystallography and NMR spectroscopy. In most cases, experimental data isn't sufficient and bond lengths and angles are often added manually resulting in structures comprised of a balanced mixture of experimental observation and knowledge-based modelling. As these experimental structures are used as the training data it is important to be critical of this.

Upon determining protein structure from x-ray crystallography, the protein is purified, crystallised and subjected to an intense beam of x-rays [14]. The crystal diffracts the beam into a pattern, showing the distribution of electrons in the protein. The electron density map can be interpreted to provide information about the location of each atom in the chain. Although x-ray crystallography shows detailed information about the protein, crystallisation can be difficult. Flexible proteins do not form useful electron density maps like rigid proteins because crystallography requires many molecules to be aligned in the same orientation [15]. The x-ray resolution measures the accuracy of crystallographic structures, with our training data having resolutions less than 2.0 Å.

For NMR spectroscopy, again the protein is purified but is then placed in a strong magnetic field and probed with radio waves [16]. A distinct set of observed resonances are used to build a model of the protein from local confirmations of bundled atoms. This method is favoured over x-ray crystallography for flexible proteins as they do not need to be locked in a crystal, instead information is provided in solution. Despite this, NMR spectroscopy is limited to small and medium proteins because large proteins have overlapping peaks. Both methods are expensive and time consuming which means great care must be taken when allocating resources to experimental methods. Considering this, determining structure *in silico* is highly favourable.

3.2 Computational Methods

The Protein Data Bank (PDB) is a computer-based archive of macromolecular structures that stores the atomic coordinates, derived from crystallographic studies, in a uniform format [17]. Many computational methods of determining structure source their training data from the PDB. Critical Assessment of Protein Structure Prediction (CASP) is a worldwide experiment for protein structure prediction that has taken place every two years since 1994 [18]. CASP lets researchers objectively test their structure prediction methods in a double-blind fashion.

3.2.1 Template-Based Modelling

A typical template-based modelling procedure involves, among others, two major steps: finding proteins with sequences similar to known structure(s), and building 3D models using the detected homologues as structural templates. There were notable improvements from CASP11 to CASP12 in the template-based modelling category with two servers from the Zhang group, Zhang-server and QUARK, outperforming the rest of the servers in a statistically significant manner [19].

I-TASSER (Zhang-server) has consistently ranked first in CASP competitions. Its name comes from its hierarchical approach of: Threading, ASSEmbly and Refinement. Threading refers to the bioinformatics procedure of identifying template proteins from structure databases. PSI-BLAST matches a query sequence to evolutionary relatives from the database. A sequence profile is created from Multiple-Sequence Alignments (MSA) and used to predict secondary structure using PSIPRED. This is then fed through 7 state-of-the-art threading programs with the top templates from each selected for assembly. Structural assembly involves continuous fragments being excised from the templates and used to build confirmations of well-aligned segments, with unaligned regions being built *ab initio.* The fragment assembly is performed by a Monte-Carlo simulation with this process being iteratively performed for refinement.

Template-based methods suffer when sequence similarity falls below 20% as the *ab initio* modelling is highly inaccurate.

3.2.2 Molecular Dynamics

Molecular Dynamics (MD) simulations can be used to model molecular systems such as proteins with an atomic level of precision. Many biological processes occur on millisecond timescales yet MD simulations on this scale lie beyond the reach of current general-purpose technology, by several orders of magnitude. D.E. Shaw are at the forefront of this computational method of determining structure.

In a 2008 paper they describe their specialised, massively parallel machine, named Anton, designed to accelerate MD simulations [20]. Anton simulates atomic motion over a period of time according to classical physics. At each discrete time step the force on each particle due to other particles is computed and the net force is used to update each particles position and velocity. The forces computed are bond, van der Waals and electrostatic forces. Their acceleration in simulation time comes from specialised hardware, which the report states outperforms the expected speedup predicted by Moore's Law.

Although Anton has enabled the longest MD simulations to date by a very large factor, it's performance is dependent on the size of the system being simulated. For instance, the protein BPTI took 18.2 μs/day to simulate, which has over 17,000 particles [21]. Despite these speed ups, Anton does not perform well on long sequences or on slow folding proteins. The Folding@home project found NTL9, the slowest-folding protein folded *ab initio* by MD simulation. NTL9 has a folding time of $\approx$1.5ms which means Anton would take nearly 3 months to simulate this singular protein [22].

4 Advances in Neural Network

4.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural network whose modern derivatives perform well on sequential data, notably speech recognition and machine translation [23, 24]. Fig. 4 shows the standard architecture of an RNN. For the purpose of intuition, it is common to imagine the RNN "unrolled" through time with each state being a time step (r.h.s. of diagram). From here on in, the "unrolled" RNN architecture is being referred to whenever mentioned. Like many deep learning frameworks, they were created in the 1980s but have only recently revealed their true potential with recent increases in computing power. The RNN scans through data from left to right, sharing parameters between each time step.

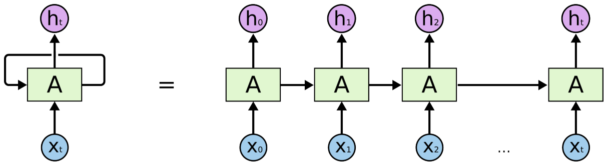


Figure 4: "Unrolled" RNN

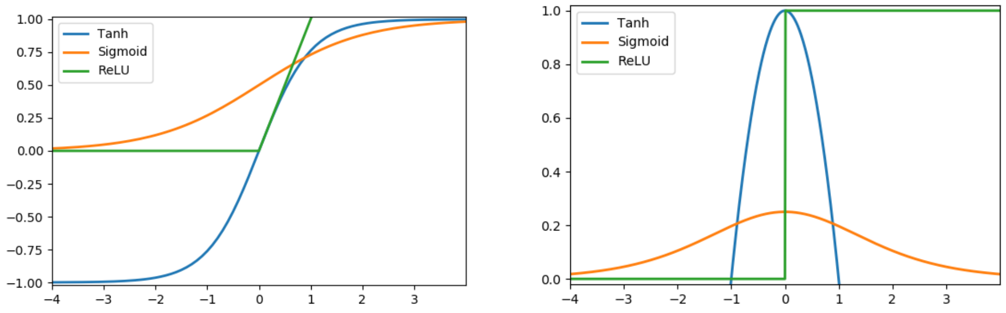
This ensures that predictions at an arbitrary time step $\hat{y}_t$ receives information from all previous states. The paper will discuss later how to incorporate information from future states at each $\hat{y}_t$. Forward propagation through an RNN is governed by the following equations, displaying how information is preserved through time:

$$h_t = \sigma_h(W_h[h_{t-1}, x_t] + b_h) \tag{1}$$

h_t is the hidden cell state at time t, which acts as the memory of the network, where W_a is a weight matrix, h_{t-1} is the hidden cell state at the previous time step, x_t is the feature vector at time t and b_h is a bias. σ_h is an activation function which is discussed later.

$$\hat{y}_t = \sigma_y(W_y h_t + b_y) \tag{2}$$

$\hat{y}_t$ is the prediction at time t, using the previously calculated h_t, its own weight matrix W_y and its own bias b_y. $\hat{y}_t$ also has its own activation function that depends on the task at hand. For the activation σ_y, when doing binary classification either sigmoid or softmax should be used but for multivariate classification softmax must be used because you cannot use a scalar function. For regression, a linear activation is best because the values are unbounded.



(a) Activations

(b) Derivatives

Figure 5: Activation Functions and their Derivatives

4.1.1 Backpropagation Through Time

To train a regular neural network we must learn the correct weight matrices via back-propagation. We need to define a loss function $J(\theta)$, that is minimised by adjusting the weights in the network, with θ being the parameters of the network. So, where N is the number of samples for a regular neural network and $\hat{y}$ a prediction of the actual output y, a loss function could look like:

$$J(\theta) = \frac{1}{N} \sum_{i}^{N} loss(\hat{y}_i, y_i) \tag{3}$$

Starting with a random initialisation of θ, Stochastic Gradient Descent (SGD) works by iteratively computing the gradient of $J(\theta)$ and moving in the opposite direction, to a new point, until convergence. The update rule calculates the new weights at each iteration with the following equation where η is a user-defined step:

$$\theta = \theta - \eta \frac{\partial J(\theta)}{\partial \theta} \tag{4}$$

With RNNs we use an adaption of this called Back-Propagation Through Time (BPTT). Again, we take the derivative of the loss with respect to each parameter and shift the parameters in the opposite direction to minimise the loss. However, since we are making a prediction at each time step, we have a loss at each time step too which must be summed to get the total loss, such as in Eq. 5. Since we have a loss at each of these time steps, to calculate the gradient, we must sum the gradients across time for each weight W, like in Eq. 6.

$$J(\theta) = \sum_{t}^{T} J_t(\theta) \tag{5}$$

$$\frac{\partial J}{\partial W} = \sum_{t}^{T} \frac{\partial J_t}{\partial W} \tag{6}$$

For a particular weight W, at a given time step t we apply the chain rule to calculate the loss. It is important to note that, for $k > 0$, the last term in the equation cannot be treated as a constant because according to Eq. (1) h_t depends on the previous cell state h_{t-1} and so on and so forth. This gives rise to one of the problems of RNNs called vanishing gradients. A generalisation for an arbitrary time step is shown by Eq. (7):

$$\frac{\partial J_t}{\partial W} = \sum_{k=0}^{t} \frac{\partial J_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W} \tag{7}$$

4.1.2 The problem of Vanishing Gradients

Although RNNs can "remember" past information basic RNNs have trouble learning long-range dependencies because of vanishing gradients. Let us assume t is a time step near the end of a long sequence. The loss at time t is calculated using Eq. (7). For illustrative purposes, let $k = 0$ so that:

$$\frac{\partial h_t}{\partial h_k} = \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial h_{t-2}} \cdots \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial h_0} \tag{8}$$

It is clear from the equation that as the gap between time steps increases this product gets longer, which is important because each term is the derivative of a cell state with respect to the previous cell state. As seen in Fig. 5, the σ in Eq. (1) is an activation function who's derivative is always less than 1. As a result, the chain rule product multiplies lots of small numbers together. This leads to errors from time steps distant in sequence having increasingly smaller gradients, often approaching 0, causing basic RNNs to be biased to short-ranged dependencies. Solutions to the vanishing gradient problem include choosing an appropriate activation function so that gradients do not become too small. Alternatively, initialising weight matrices to the identity matrix as opposed to a zerod matrix will help as well. Finally, as we will see in the following section, gated RNNs avoid the problem of vanishing gradients.

4.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber in 1997 as the remedy to the vanishing gradient problem of conventional BPTT [25]. They have since been refined and popularised by many researchers, most notably Felix Gers, Fred Cummins and Alex Graves [26, 27, 28]. The cell state C, is fundamental in avoiding vanishing gradients, running through the entire chain and taking only linear interactions. The LSTM can remove and add information to the cell state, which is regulated by three gates: the forget gate, the input gate and the output gate. Initially the LSTM decides which information to discard from the cell state through a multiplicative sigmoid layer f_t, or the forget gate. Eq. (9) looks at the previous hidden state h_{t-1} and the current feature vector x_t, outputting a vector of values between 0 and 1, with 0 meaning completely forget and 1 meaning completely remember. Next the LSTM decides what new information to store in the cell state. The input gate layer i_t in Eq. (10) decides which values to update and a tanh layer creates a vector of candidate values in Eq. (11), $\tilde{C}_t$. The product of these is added to the product of f_t and the previous cell state to give the current cell state C_t. Finally, we decide what to output at time t. The output, h_t is the product of a sigmoid layer on the previous hidden state and the current feature vector, and a tanh layer on the newly calculated cell state, displayed in Eq. (14). LSTMs enforce constant error flow through "constant error carrousels" by truncating the gradient where this does not do harm and multiplicative gate units learn to open and close access to this constant error flow, solving the problem of vanishing gradients [29, 30].

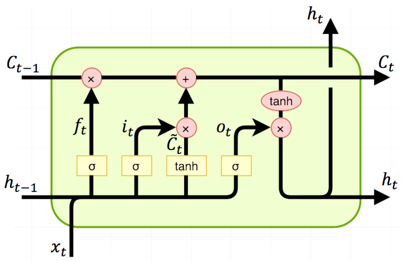


Figure 6: LSTM Cell

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{9}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{10}$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{11}$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \tag{12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{13}$$

$$h_t = o_t \tanh(C_t) \tag{14}$$

4.3 Bi-directional Recurrent Networks

In 1997, the same year that LSTMs were introduced, so too were Bi-directional RNNs (BRNNs) by Schuster and Paliwal [31]. As mentioned previously it would be useful if an RNN could also take into account future information. Prior to their publication, it was found that delaying output by M time steps meant the RNN could include $x_t + M$ steps of information. In theory, M could be made large enough to capture all future information. However, in practice prediction results dropped with large M. Small M did in fact improve performance but it's value had to be found via trial and error. To overcome the limitations of basic RNNs they proposed BRNNs, whose structure splits the state neurons of an RNN into two parts. The first part is responsible for the positive time direction and the second part responsible for the negative time direction, or forward and backward respectively. Outputs of forward states are not connected to the inputs of backwards states and vice versa. Graves concluded that bidirectional networks are significantly more effective than unidirectional ones, especially where context is vitally important, in a 2005 paper on the task of framewise phoneme classification [32].

4.4 Encoder-Decoder

Cho *et al.* proposed, more recently, a novel neural network model consisting of two RNNs that act as an encoder-decoder pair [33]. The encoder encodes a variable-length sequence into a fixed-length vector representation, and the decoder maps that representation back to a variable-length sequence. Both RNNs are trained jointly to maximise the conditional probability of a target sequence given a source sequence.

From a probabilistic perspective, the encoder-decoder model learns a conditional distribution over a variable-length sequence conditioned on another variable length sequence $p(\hat{y}_1, \ldots, \hat{y}_{T'} | x_1, \ldots, x_T)$ where T and T' may differ (although in our case they will not).

The encoder RNN reads each x_t sequentially, changing the hidden state h_t according to Eq. (1). Eq. (15) shows that once the RNN reaches the end of the sequence, as discussed previously, h_t is a summary of the whole input sequence. The decoder generates the output sequence by predicting the next symbol $\hat{y}_t$ given h_t. However, unlike regular RNNs, both $\hat{y}_t$ and h_t are conditioned on $\hat{y}_{t-1}$ and C. Thus, the hidden state of the decoder at time t is used to calculate the conditional distribution sequence using appropriate activation functions f and g. Therefore, to calculate the probability of a predicted sequence $\hat{Y}$, the decoder decomposes the joint probability into the ordered conditionals [34].

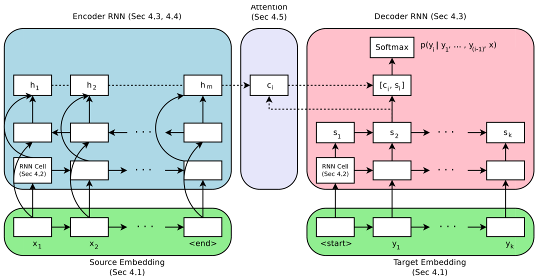


Figure 7: Attentive Encoder-Decoder

$$h_t = f(h_{t-1}, \hat{y}_{t-1}, C) \tag{15}$$

$$p(\hat{y}_t | \hat{y}_1, \ldots, \hat{y}_{t-1}, C) = g(h_t, \hat{y}_{t-1}, C) \tag{16}$$

$$p(\hat{Y}) = p(\hat{y}_1, \ldots, \hat{y}_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T} (\hat{y}_t | \hat{y}_1, \ldots, \hat{y}_{t-1}, C) \tag{17}$$

Cho *et al.* showed that encoder-decoder improved overall translation performance. However, upon investigation of the limitations of their model in a subsequent paper, they showed that neural machine translation performs relatively well on short sequences but performance degrades rapidly as the length of the sequence increases [35].

4.4.1 Attention Mechanism

Leading from this, Cho *et al.* went on to conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of the encoder-decoder architecture [36]. They proposed that by allowing the model to search for parts of the sequence that are relevant to predicting a target symbol, they can achieve state-of-the-art performance in neural machine translation. Attention has also shown promise in computer vision problems, such as image captioning [37]. Each time the proposed model generates a target symbol it searches, typically with a beam search, for a set of positions in the input sequence where the most relevant information is concentrated. The distinguishing feature of this model is that it no longer encodes the input sequence into a single fixed-length vector. Instead, the input is encoded to a sequence of vectors and chooses a subset adaptively while decoding the output. The new model defines each conditional probability from Eq. (17) as:

$$p(\hat{y}_i | \hat{y}_1, \ldots, \hat{y}_{i-1}, X) = g(\hat{y}_{i-1}, h_t, C_i) \tag{18}$$

Unlike Eq. (17) the probability is conditioned on a distinct context vector C_i for each target symbol $\hat{y}_i$. The context vector C_i depends on a sequence of annotations $(n_1, \ldots, n_T)$ where each n_i contains information about the whole input. The context vector C_i is computed as a weighted sum of the annotations where $e_{ij} = a(h_{i-1}, n_j)$ is an alignment model that scores how well the inputs around position j match outputs at position i. In 2016 Google Translate switched to the Google Neural Machine Translation system, which is based on an encoder-decoder with attention, exhibiting the power of this novel architecture [38].

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})} \tag{19}$$

$$C_i = \sum_{j=1}^{T} \alpha_{ij} n_j \tag{20}$$

5 Project Specification

5.1 Aims and Objectives

The main objective of the project is to transfer the success of the attentive encoder-decoder architecture to the field of bioinformatics, ultimately improving the accuracy of *ab initio* methods of protein tertiary structure prediction. Aside from this, the project will provide an in-depth analysis of the evolution of the model and how tuning of the hyperparameters affects performance. The models predictions will be compared to CASP candidates of previous years as an unbiased, objective measure of quality. A typical feed-forward through the network should take a matter of milliseconds, making it faster (assuming training is complete) than MD simulations. Moreover, due to training on proteins with low sequence similarity, the project aims to outperform template-based predictions on the same proteins.

5.2 Related Research

AlQuraishi released a paper earlier this year in which he utilises stacked BRNN gated with LSTMs to predict protein tertiary structure in a proposed model name a Recurrent Geometric Network (RGN) [39]. It takes as input a sequence of amino acids and a Position-Specific Scoring Matrix (PSSM) and outputs a 3D protein structure within 1-2 Å of template-based approaches, despite being template-free. The RGN is comprised of 3 stages: computation, geometry and assessment. The computation stage outputs from the inputted amino acids and PSSMs a vector of torsional angles at each unit. The geometric stage takes the torsional angles for a given residue and the partially completed backbone resulting from upstream geometric units and outputs a new backbone, extended by one residue, which is fed into the adjacent downstream geometric unit. Hence, the last unit outputs the full 3D structure of the protein. While training, the assessment stage computes the distance-based Root Mean Squared Deviation (dRMSD). It first computes pairwise distances between the atoms in the predicted structure and experimental structure and then the root mean square of the distance between these sets of distances. dRMSD is the loss function this is minimised in backpropagation. To assess the models error, AlQuraishi used it on the CASP7-12 candidates after training on CASP11 candidates as this provided the aforementioned benchmark. The paper states that the limitation of RGNs is its reliance on PSSMs, as they are much weaker than Multiple Sequence Alignments. The paper makes no mention of physio-chemical properties.

Ingraham attempted to learn an energy landscape and simulator simultaneously by designing a simulator as an RNN by unrolling it through time, where the weights are energetic restraints and the cell state the coordinates of the whole protein [40]. The simulator is initialised with a random set of coordinates for the protein governed by the energy restraints and learns via backpropogation how to manipulate the energy landscape to fold towards an energetically favourable confirmation. The paper uses Cartesian coordinates but notes that torsion angles could be used instead.

5.3 Design

The input data will be acquired from a search in the PDB for proteins with x-ray resolutions less than 2.0 Å, containing no DNA/RNA, with sequence identities less than 30%. The low x-ray resolution ensures that the training data is well resolved while the low sequence similarity ensures the model does not overfit on homologous training data. The input feature set will be made up of a range of physiochemical features of each amino acid. These include, but are not limited to: numerous hydrophobicty and polarity indices [41, 42, 43], the normalised number of hydrogen bond donors and charge parameters [44], and each amino acid's normalised preference of secondary structure [45, 46, 47]. BioPython is an open-source Python library for computational biology. It provides a tool for converting the Cartesian coordinates from the PDB file of a specified protein into the Φ and Ψ angles needed for the output data.

The input data will be bucketed based on sequence length with inputs being padded up to the maximum length of each bucket. The amino acids of each sequence will be one-hot encoded. The final model will consist of stacked, bi-directional, LSTM gated RNNs in an encoder-decoder architecture with attention. Stochastic gradient descent algorithms, such as ADAM and RMSProp, will be analysed to decide on the best gradient descent algorithm. As this is a regression problem, the mean-squared error will be used as the loss function between predicted and target torsional angles.

After training is complete, the predicted Φ and Ψ angles will be converted to Cartesian coordinates using a method adapted by AlQuraishi called parallelised Natural extension Reference Frame (pNeRF) [48]. From here structure can be reconstructed to compare the predicted structure to the target structure.

5.4 Limitations

The main issue to contend with is time. Training of the model, to a desirable accuracy, is likely to take several days to several weeks, depending on the architecture, feature set size and number of samples. As a result of this, development is already underway but it is crucial that the GPU speed ups provided by TensorFlow are taken advantage of.

Errors in angle prediction close to the N-terminus may cause rotational offsets in all subsequent residues upon conversion to Cartesian coordinates. This error may increase as the angular errors accumulate through the chain. A solution to this could be to devise an attention mechanism with a beam search whose beam width is proportional to the predicted Φ and Ψ angles deviation from their closest cluster on the Ramachandran plot. This ensures an increase in the searched space for the most inaccurate predictions. The model may not perform well on flexible proteins as the experimental training data was determined from x-ray crystallography where it is difficult to crystallise flexible proteins. Following from this, crystal structures may not be representative of *in vivo* because they have been forced to crystallise.

Although proteins can fold to their correct confirmations without outside help, it has been shown since Anfinsen's research that protein folding in a living cell is often assisted by *molecular chaperones*. Chaperones are special proteins that bind to the partly folded polypeptide and aid them along the most energetically favourable pathway. This is important to note as it suggests that amino acid sequence is not the sole determinant in tertiary structure.

5.5 Evaluation of Results

The Global Distance Test measure will be used to evaluate the predicted structure as this is the method used in CASP competitions, providing an easy comparison between the results of the model and the results of CASP candidates. For visual purposes, it will be interesting to superimpose the predicted structure on the reference structure to highlight the position of deviations.

6 Conclusion

The paper showed the advances in biology, bioinformatics and computer science that have made it possible to attempt protein structure prediction with computational methods. It revealed the accurate yet slow simulation time of MD methods and the effient yet less accurate predictions of template-based modelling. From here, the paper proposes a more general solution between the two based on the effective encoder-decoder neural network architecture with attention. It provides an in-depth analysis of the components that make up the encoder-decoder such as recurrent neural networks, long short-term memory cells and the attention mechanism. The predominant conclusion to take forward to the project itself, is that development of the model needs to continue moving forward at a steady pace to increase the available training time upon completion. Aside from this, supplementary research is needed around custom attention mechanisms. Furthermore, a strategy that can encode molecular chaperones into the model will be ideal. Finally, an exploration into stochasitc gradient descent algorithms will yield improved performance from backpropagation.

References

[1] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in Bioinformatics*, 19(3):482–494, 2018.

[2] Federation of European Biochemical Societies (FEBS) (Utrecht). Nomenclature and symbolism for amino acids and peptides. *European Journal of Biochemistry*, 138(1):9–37, 1984.

[3] Jeremy M. Berg, John L. Tymoczko, and Lubert Stryder. *Biochemistry.* W. H. Freeman and Company, seventh edition, 2012.

[4] Linus Pauling, Robert B. Corey, and H. R. Branson. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951.

[5] Burkhard Rost. Protein secondary structure prediction continues to rise. *Journal of Structural Biology*, 134:204–218, 2001.

[6] Alan Fersht. *Structure and Mechanism in Protein Science: Guide to Enzyme Catalysis and Protein Folding.* World Scientific, 1999.

[7] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell, 4th edition.* Garland Science, 2002.

[8] Christian B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[9] Christian B. Anfinsen, Robert R. Redfield, Warren L. Choate, Juanita Page, and William R. Carroll. Studies on the gross structure, cross-linkages, and terminal sequences in ribonuclease. *Journal of Biological Chemistry*, 2017:201–210, 1954.

[10] Christian. B. Anfinsen, E. Haber, M. Sela, and F. H. White. The kinettics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47(9):13091314, 1961.

[11] Frederick H. White Jr. Regeneration of native secondary and tertirary structures by air oxidation of reduced ribonuclease. *The Journal of Biological Chemistry*, 236:1353–1360, 1961.

[12] G. N. Ramachandran. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7:95–99, 1963.

[13] G. N. Ramachandran and V. Sasisekharan. Conformation of polypeptides and proteins. *Advances in Protein Chemistry*, 23:283–438, 1968.

[14] Andrea Ilari and Carmelinda Savino. Protein structure determination by x-ray crystallography. *Methods in molecular biology (Clifton, N.J.)*, 452:63–87, 02 2008.

[15] Elisabeth P Carpenter, Konstantinos Beis, Alexander D Cameron, and So Iwata. Overcoming the challenges of membrane protein crystallography. *Current opinion in structural biology*, 18(5):581–586, 10 2008.

[16] Kurt Wüthrich. The way to nmr structures of proteins. *Nature Structural Biology*, 8:923 EP –, 11 2001.

[17] FC Bernstein, TF Koetzle, GJ Williams, EF Meyer, MD Brice, JR Rodgers, O Kennard, T Shimanouchi, and M Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *Journal of molecular biology*, 112(3):535–542, May 1977.

[18] John Moult, Jan T. Pedersen, Richard Judson, and Krzysztof Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 23(3):ii–iv, 1995.

[19] Andriy Kryshtafovych, Bohdan Monastyrskyy, Krzysztof Fidelis, John Moult, Torsten Schwede, and Anna Tramontano. Evaluation of the template-based modeling in casp12. *Proteins: Structure, Function, and Bioinformatics*, 86(S1):321–334, 2018.

[20] David E. Shaw, Martin M. Deneroff, Ron O. Dror, Jeffrey S. Kuskin, Richard H. Larson, John K. Salmon, Cliff Young, Brannon Batson, Kevin J. Bowers, Jack C. Chao, Michael P. Eastwood, Joseph Gagliardo, J. P. Grossman, C. Richard Ho, and Ist Douglas J. Ierardi. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51:91–97, 2008.

[21] David E. Shaw. Millisecond-scale molecular dynamics simulations on anton. Technical report, D. E. Shaw Research, 2009.

[22] Vincent A. Voelz, Gregory R. Bowman, Kyle Beauchamp, and Vijay S. Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9(1-39). *Journal of the American Chemical Society*, 132(5):1526–1528, 2010.

[23] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013.

[24] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. Massive exploration of neural machine translation architectures. *CoRR*, abs/1703.03906, 2017.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short term memory. *Neural Computation*, 9:1735–1780, 1997.

[26] Felix A. Gers. Long short-term memory in recurrent neural networks. Technical report, École Polytechnique Fédérale de Lausanne, 2001.

[27] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. Technical report, IET Digital Library, 1991.

[28] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013.

[29] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[30] Yoshua Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994.

[31] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, 1997.

[32] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 4:2047–2052, 2005.

[33] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[35] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[36] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015.

[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[39] Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *bioRxiv*, 2018.

[40] Anonymous. Learning protein structure with a differentiable simulator. In *Submitted to International Conference on Learning Representations*, 2019. under review.

[41] Patrick Argos, J. K. Mohana Rao, and Paul A. Hargrave. Structural prediction of membrane-bound proteins. *European Journal of Biochemistry*, 128(2-3):565–575, 1982.

[42] "Daniel D. Jones". Amino acid properties and side-chain orientation in proteins: A cross correlation approach. *Journal of Theoretical Biology*, 50(1):167 – 183, 1975.

[43] J.M. Zimmerman, Naomi Eliezer, and R. Simha. The characterization of amino acid sequences in proteins by statistical methods. *Journal of Theoretical Biology*, 21(2):170 – 201, 1968.

[44] Jean-Luc Fauchère, Marvin Charton, Lemont B. Kier, Aarie Verloop, and Vladimir Pliska. Amino acid side chain parameters for correlation studies in biology and pharmacology. *International Journal of Peptide and Protein Research*, 32(4):269–278, 1988.

[45] Peter Y. Chou and Gerald D. Fasman. *Prediction of the Secondary Structure of Proteins from their Amino Acid Sequence*, pages 45–148. John Wiley & Sons, Ltd, 2006.

[46] Michael J. Geisow and Robin D.B. Roberts. Amino acid preferences for secondary structure vary with protein class. *International Journal of Biological Macromolecules*, 2(6):387 – 389, 1980.

[47] Minoru I. Kanehisa and Tian Yow Tsong. Local hydrophobicity stabilizes secondary structures in proteins. *Biopolymers*, 19(9):1617–1628, 1980.

[48] Mohammed AlQuraishi. pnerf: Parallelized conversion from internal to cartesian coordinates. *bioRxiv*, 2018.