Chapter 6: Sequence-To-Sequence Modeling for Protein Secondary Structure Prediction

6.1 Introduction

Proteins are biomolecules vital for developing the cell of all living organisms. Proteins perform as catalysts, hormones, receptors, and transporter within all living organisms [81]. Prediction of protein secondary structure helps in estimating protein functions. Protein secondary structure prediction from their primary sequences playing an essential role for several applications such as drug designing. There are twenty amino acids those makeup protein primary sequences. Each amino acid is abbreviated to one or three letters called codons. Protein structure is described mainly in three levels primary structure, secondary structure, and tertiary structure [12]. Primary sequences are a linear chain of twenty amino acids by linking the amino group of one amino acid with the carboxyl group of another amino acid with a peptide bond. The three categories for secondary structure prediction are alpha-helix, Beta Sheet, and coil. The tertiary structure is the three-dimensional structure of the protein. Protein sequences are dependent on biophysical and biochemical properties to determine secondary structure like sentences in natural language dependent on the grammatical rule. Due to this likeness, protein sequences are input to the sequence-to-sequence method for secondary structure prediction. Sequence-to-sequence learning is

about training models to convert sequences from one language to sequences in another language.

Natural language processing has a wide range of character based research work for classification and prediction. Character-level information is classified into three classes based on the inclusion of character-level information in their computational models. The three types are Tokenization based models, Bag-of-n-gram models, and End-to-end models. However, some blends are also possible, such as the token-based bag-of-n-gram model and the bag-of-n-gram model trained end-to-end. Machine translation utilizes character-based methods to avoid problems such as rare words or out-of-vocabulary words. Neural machine translation has become popular recently with character-level and sub-word units. Sequence-to-sequence models with character-based neural networks are applied to several tasks like machine translation, question answering, and speech recognition.

The neural machine translation performed well in several applications of language translation. A recurrent neural network-based language model was proposed, which performs natural language processing using neural machine translation [41]. A deep long short term memory based sequence model proposed for machine translation. This recurrent neural network-based model has an eight-layer in the encoder and an eight-layer in the decoder [82]. The neural machine translation models are needed less memory compared to traditional machine translation models. The neural machine translation model utilizes the capability of deep learning and representation learning.

Furthermore, unlike conventional translation systems, all parts of the neural translation model are trained jointly (end-to-end) to maximize the translation

performance[82]. Character-based models have overcome the problem of neural machine translation, such as rare or out-of-vocabulary words. Due to higher performance of character-based models many variants are proposed [83–85]. Neural machine translation with character and character n-gram words having better performance [86–89]. Some attention-based neural translation models are proposed, which makes joint learning of alignment and translation [67]. Deep learning-based models such as hierarchical recurrent neural network [67,87] and combinations of recurrent neural network and convolutional neural network have been proposed for neural machine translation[86,89].

Protein sequences are governed by their biophysical and biochemical principles, similar to the grammar of a natural language determining sentences' structure. This analogy motivates treating protein sequences as the output of a specific language and develops natural language processing sequence-to-sequence method to discover functions encoded within protein sequences. Several methods were proposed for predicting the secondary structure of proteins in the last six decades. Protein secondary structure prediction methods combine protein sequence database homology searches [90–92] with features, such as physicochemical properties of the primary sequence [93], backbone torsion angles [8]. These combined features fed into neural networks [94, 95] or deep neural networks [5,7] produce the secondary structure. Recently, the most used tools for secondary structure prediction are PSIPRED [96] and JPRED [90], which have an average Q3 accuracy of approx 80–85% for benchmark datasets. PSIPRED was the first secondary structure prediction tool that uses the PSI-BLAST search to improve prediction accuracy. PSIPRED uses the

UNIREF90 protein database to obtain a sequence profile and then fed into a two-layer neural network. JPRED integrates the PSI-BLAST sequence profile and HMMer [97] sequence profiles as inputs to the neural network. The currently available state-of-the-art tools that achieve the highest accuracy Q3 accuracy around 85–90% include SSpro[36], DeepCNF[98], PORTER [94], and PSRSM [99]. SSpro is used template data by utilizing sequence homology sequences. If homology sequences are not available, then the neural network predicts their secondary structure element. DeepCNF and PORTER use a combination of deep convolutional neural networks and conditional neural fields with sequence profiles. PSRSM uses a support vector machine variation. This chapter main contribution is to (1) propose a model, which learns to map a protein primary sequence to a secondary structure sequence as language translation. (2) One LSTM layer is used for the encoder to encode the primary sequences and returns its internal state to input in the decoder. Another LSTM layer works as a decoder to predict the secondary structure sequence. (3) The proposed model is evaluated for the three-class secondary structure predictions on two publicly available datasets cullpdb and data1199. Experiments demonstrate -that the combination of LSTM based encoder and decoder captures better features to improve the secondary structure prediction.

6.2 Datasets

To train the model protein data was obtained by downloading from cullpdb[43], with a resolution of less than 2.5 Å, and R-factor less than 1.0, 30% non-redundant sequences in February 2017. We removed sequences with incomplete information with a length of less than 30 residues, a similarity greater than 30% according to

BlastClust [100]. We used publicly available dataset data1199 as testing set. The test set came from [5]including 1199 non-redundant sequences.

Table 6.1: Amino acid one-hot Encoding Representation

A	1	0	0	0	-------	0	0
R	0	1	0	0	-------	0	0
C	0	0	1	0	-------	0	0
D	0	0	0	1	-------	0	0
T	0	0	0	0	-------	1	0
N	0	0	0	0	-------	0	1

One-hot vector encoding is an instance of one-to-one representation as shown in table 6.1. One-to-one representation is known as local representation. Given a set of M characters we can represent each character w_i using a vector in the size of M, where the vector has zeros everywhere except at the index of word *i* having the value of 1, hence calling this representation "one-hot vector representation".

6.4 Neural Machine Translation

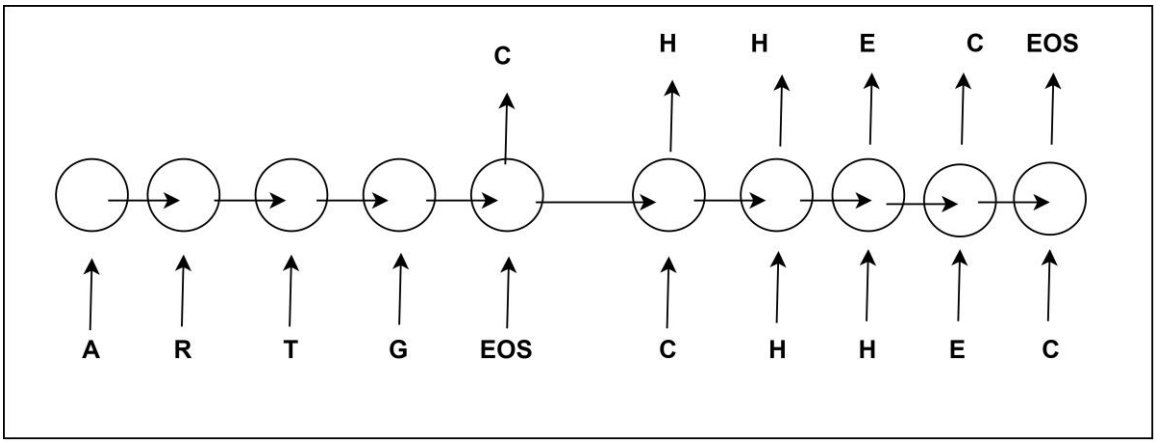

Figure: 6.1 Sequence to Sequence using LSTM [8]

Neural machine translation is a machine translation system that uses an artificial neural network to increase fluency and accuracy the process of machine translation. Neural machine translation is based on a simple encoder decoder based network. The type of neural networks used in neural machine translation is recurrent neural networks [8]. The reason for selecting recurrent neural network for the task is the basic architecture of the recurrent neural network. Recurrent neural network involves cyclic structure which enables the learning of repeated sequences much easier as compared to other neural network architectures. Recurrent neural network can be unrolled to store the sentences as a sequence in both sources as well as target languages. A typical structure for recurrent neural networks is described in Fig. 6.2. The figure explains how a single layer of recurrent neural networks can be unrolled into multiple layers, and information of the previous time period can be stored in a single cell as well. Recurrent neural network architecture can be modified to provide better solutions for a specific task. Chopra et al. [9] have used attentive recurrent neural network architecture for abstractive sentence summarization. Luong et al. [10] have developed attention-based neural machine translation using modified recurrent neural network architectures.

In this work, we explore a simplified and shallow network which learns to map a protein primary sequence to a secondary structure sequence as language translation.One LSTM layer is used for the encoder to encode the primary sequences and returns its internal state to input in decoder. Another LSTM layer works as a decoder to predict the secondary structure sequence. The proposed model is evaluated for the three-class secondary structure predictions on two publicly available datasets

cullpdb and data1199. Experiments demonstrate that the combination of LSTM based encoder and decoder captures better features to improve the secondary structure prediction.

Let X and Y be the source and target as protein primary sequences and secondary structure, respectively. The neural machine translator decoder is using conditional probability

$$P(y \mid x) = P(y \mid x_1, x_2, x_3 ....., x_m) \qquad \textbf{(6.1)}$$

where $x_1, x_2, x_3, \ldots\ldots x_m$ are the fixed size vectors encoded by the encoder. Using the chain rule, the above expression then becomes

$$P(y \mid x) = P(y \mid y_1, y_2, y_3 ....., y_{i-1}, x_1, x_2, x_3 ....x_m) \qquad \textbf{(6.2)}$$

Thus because of the multiplicative effect, the output in longer sentences is very low and results in inaccuracy. In practice, it is difficult for RNNs to learn these dependencies. Since the typical sequences have such complex context-dependent cases, RNN should not be used for encoder and decoder design. To overcome the shortcomings of the RNNs, we use LSTM models for encoding and decoding.

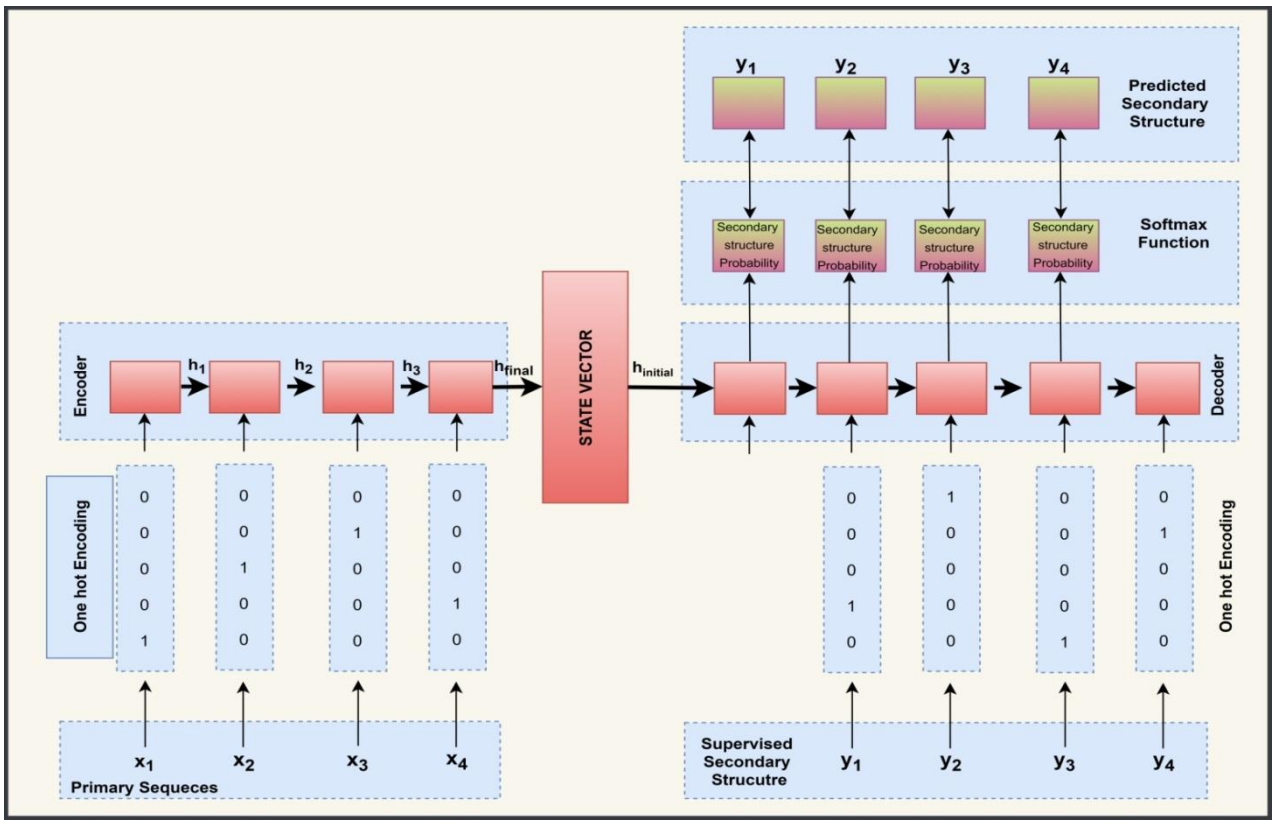

Figure 6.2: Sequence to sequence model for protein secondary structure prediction

6.3 Proposed Model

The protein sequences are split into twenty amino acid characters. Each amino acid numbered with an integer value in the range of 1-20. The amino acid characters are represented as one-hot vector. The protein sequences are of varying length, but the deep learning model accepts the fixed-length. If any sequence exceeds in size, then the remaining character is discarded. We padded with zeros in the shorter sequence.

The Proposed protein secondary structure prediction model is long short-term memory network-based sequence-to-sequence architecture. The input is the primary sequence, and the output is the protein secondary structure. Naturally, both the input

protein primary sequences and output secondary structure sequences are of equal lengths. Since each amino acid, character has a corresponding secondary structure character. In the proposed model, the conditional probability of a secondary structure sequence ($y_1, \ldots, y_m$) estimated given an input amino acid sequence ($x_1, \ldots, x_n$).

$$p\left(y_1, y_2 \ldots\ldots\ldots y_n \mid x_1, x_2, \ldots\ldots\ldots x_n\right) \tag{6.4}$$

The proposed sequence-to-sequence model first encodes the input protein primary sequence one amino acid character at a time, representing the primary sequence using a latent vector representation. Then, it decodes that representation to a secondary structure sequence.

Long short term memory (LSTM) [39] uses the capability of input gate (i_t), forget gate (f_t), and output gate (o_t) to control the flow of information to operate selective read, selective forget, and selective write. To utilize the information efficiently and discarding the unnecessary information, the three gates in LSTM uses the current input, previous state, and output selectively. The activation function depends on the gates used in LSTM. The logistic and sigmoid function is used as an activation function. The flow of information depends on the memory blocks used in the hidden layer. The Sequence to Sequence for primary Sequence to the secondary structure is depicted in Figure 6.3. The proposed model uses two LSTMs with 256 units each. The state vector representation obtained from the first LSTM provided as the input to the second LSTM. The first LSTM is used to represent the protein primary sequence, and the second LSTM is used to model the secondary structure sequence. The first LSTM layer receives a sequence of amino acid character and encodes to state vectors while the second LSTM layer receives the state vectors. After the whole Sequence is

exhausted, the second LSTM layer is fed the secondary structure character, which prompts it to decode its current state vectors into a sequence of characters. During training, the decoding stage maximizes the predicted secondary structure sequence's log-likelihood given the state vectors of the primary Sequence and the previous secondary structure character it has seen.

6.4 Results and Discussion

We have focused on the implementation of our NMT system in two phases. In the first phase, we have focused only on cullpdb[20] sequence translation. The reason for selecting cullpdb[20] is the availability of more resources for the language pair. Once the NMT system is operational on this language pair, we have tested the same on data1199 language pairs. The Q3 accuracy for protein secondary structure prediction is defined as the percentage of total residues for which the secondary structure predicted correctly.

Python version 3.6.7 was used for implementing the proposed sequence-to-sequence model. Keras[47] used at the front end and Tensorflow [48] version 1.9 used as the back-end. Keras is an open-source, high-level machine learning API implemented using Python programming language. Tensorflow has excellent computational ability.

The dataset is divided into a ratio of 70 to 30, using a statistical sampling method for training and testing. This splitting of data between training and testing is used to ensure the result's accuracy and produce a more generalized model. The recurrent neural network model requires a significant amount of processing time for large datasets due to complex hidden layers with many processing cells. To avoid

overfitting, a dropout of 0.1 was adopted in the proposed model. The large mini-batch size results in high computation time for each iteration. If a mini-batch size is too small, then the process never converges. The RmsProp optimization [49] is used with a mini batch size of 64 to process the model faster. The categorical cross-entropy is used to update the weights and bias. The categorical cross-entropy calculated by the negative loglikelihood loss between the supervised training data and model probability distribution shows a loss between actual and predicted values for the given training data.

The proposed sequence to sequence model performance for protein secondary structure prediction is compared with other single sequence-based methods SPIDER3-Single[4] and PSIpred-Single[50] cullpdb dataset listed in Table 6.1. We find that the proposed model performance with single LSTM is higher than other sequence-based methods, i.e., SPIDER3-Single[4] and PSIpred-Single[50]. SPIDER3-Single combines the one hot feature vector with Bidirectional LSTM for protein secondary structure prediction, and PSIpred-Single considers statics of significant amino acids by calculating their correlation at each segment.

Table 6.2: Comparison of the performance of various single-sequence based prediction on Cullpdb

Methods	**Q3(%)**
Seq2Seq(LSTM)	84.87
SPIDER3-Single	73.24
PSIpred-Single	70.21

Table 6.3: Comparison of various single-sequence based prediction on data1199

Methods	**Q3(%)**
Seq2Seq Model	87.39
SPIDER3	83.3
JPred4	79.3
RaptorX	81.5

To show the effectiveness of the proposed sequence-to-sequence model performance compared with single sequence based methods SPIDER3 [4], JPred4[90], and RaptorX [77] to predict protein secondary structure for dataset data1199 shown in table 6.2. Spider3 methods used bidirectional long short term memory deep learning neural network. JPred4 used the JNet procedure, and RaptorX used a deep neural network of convolutional neural fields for protein secondary structure prediction. The protein sequences in the testing sets and training sets have low similarity to estimate our model's performance. The Q3 accuracy of our sequence-to-sequence model is 87.39%.

6.5 Conclusion

Statistical phrase-based MT systems have been facing the problem of accuracy and requirement of large datasets for a long time, and in this work, we have investigated the possibility of using a shallow RNN and LSTM-based neural machine translator for solving the issue of protein secondary structure prediction. In the proposed work, protein primary sequences are translated to their secondary structure using sequence-

to-sequence model. Protein primary sequences represented as one hot encoding, directly feed to LSTM based sequence-to-sequence model. The proposed model achieves state-of-the-art performance on two publicly available dataset CullPDB and data1199. In spite of its simplicity, the LSTM based sequence-to-sequence model easily captures the complex relation between amino acids and their secondary structure.