Protein Secondary Structure Prediction with Gated Recurrent Neural Networks

R.Thendral, AN.Sigappi

Abstract: ***In computational biology, the protein structure from its amino acid sequence is difficult to predict, which impact the design of drug and molecular biology. Improving the accuracy of predicting acceptable protein structure is the main problem of predicting structure problem. The deep learning method is suitable for high level relation feature from the target protein sequence. Recurrent Neural Network(RNN) handle sequence data in effective manner. Experiment conducted on a well-known standard data set of the RCSB[12] shows that our model is extensively better than the state-of-the-art methods in different statistical measurement. This study makes clear and carry out the deep learning method can increase the protein properties and achieve a Q3 accuracy of 86 percentages .***

Keywords- Protein Structure prediction, Recurrent neural network, Long short term memory.

I. INTRODUCTION

Proteins play a main role in the human body's biological systems. Proteins are large, difficult molecules that play a number of major tasks in the body.

There is a difficult dependence between the sequence of a protein and its structure and one of the main computational biomedical challenges is to specify the structure for protein sequence [1], protein structure is the amino acid sequence ordered in the chain of polypeptides. The secondary structures of proteins are called repeated regular conformations on the polypeptide chain. Classically, secondary protein structures are described as three general conditions: helix (H), strand (E), and coil (C).

Predictions of the three state protein sequences known as Q3 prediction issues have been researched intensively using a variety of machine learning algorithms for decades, including the probability graph models [2,3], support vector machines [4, 5], hidden Markov models [6,7], artificial neural network [8].

Some early attempts to predict secondary structure have used quantitative methods, using properties gather from proteins with already define secondary structure, the ideas achieve accuracy more than 65% [9].

Machine learning method have used support vector machines [9] and neural networks for protein prediction and it has proved [9]. The technique of machine learning algorithms with protein sequence profiles on homology was a major step forward in secondary structure prediction, increasing accuracy to 70-79% [9].

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author

Thendral R*, Research Scholar, Computer Science and Engineering, Annamalai University, India. thendralrengarajanp@gmail.com
Dr. Sigappi.AN, Professor, Computer Science Engineering, Annamalai University, India. aucse_sigappi@yahoo.com

Protein sequence profiles were utilized in the SPINE neural network method reach 79.5% accuracy [9]. The deep learning methods can extract the features of amino acid with hidden structure in protein sequences compare with traditional machine learning methods. Biological derived data is often sequential for example amino acid sequence and DNA. The Neural network architectures can handle sequential data by naturally. Based on the Neural Network, the proposed system implements the LSTM network to predict protein structure for better performance.

In this work, the deep learning model is designed by long short time memory with and Recurrent neural network(LSTM-RNN). The LSTM system combines the learning of the features of the different protein properties and the estimation of the secondary protein structure to a high level.

II. THE DEEP LEARNING MODEL

The type of neural network designed to handle sequencing dependence is called a recurrent neural network. The LSTM RNN Network is used in deep learning as it is possible to successfully train very large architectures.

1.Simple RNN

The efficient method for handling sequence data is Recurrent Neural Network(RNN),.The process of RNN is shown in fig(1).

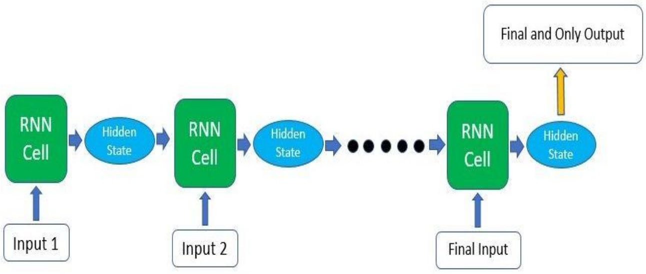


Figure 1 Simple RNN

A hidden state is formed as a matrix of zeros in the first step, so that it can be delivered to the RNN cell along with the first input in the sequenceIn basic RNNs, output data and hidden state are multiplied by weight matrices activated by Xavier or Kaiming. The multiplication result will be moved through the activation function (such as the tanh function) to introduce non-linearity. he hidden state that we have generated will then be fed back into the RNN cell together with the next data, and this process continues until we run out of input or the system is designed to stop producing outputs.

$$hidden_t = f(hidden_t\_input_t) \quad (1)$$

$$output_t = (weight_{output} * hidden_t) \quad (2)$$

2. Long Short-Term memory Network

German researchers Hochreiter and Schmidhuber proposed the key idea for Long Short-Term Memory (LSTM) in 1990s., It would allow the RNN to retain information over a longer period of time, not just between two stages in time. The LSTM cell allows the network to collect and store such relevant information and, if necessary, to inject it back into the model

Long short-term memory (LSTM) is a system which can learn task by using deep learning and avoids vanishing gradient problem [10]. LSTM is strengthened by repeated gates called "forget" gates and the memory of events that have occurred in history is required to learn tasks.

3. Working principle of LSTM

Long-short term memory network is powerful algorithm that can classify, group and make predictions about data, particularly time series and text. This feature address the problem of "short-term memory" of RNNs.

The differences between RNN and LSTM model are shown in fig2.

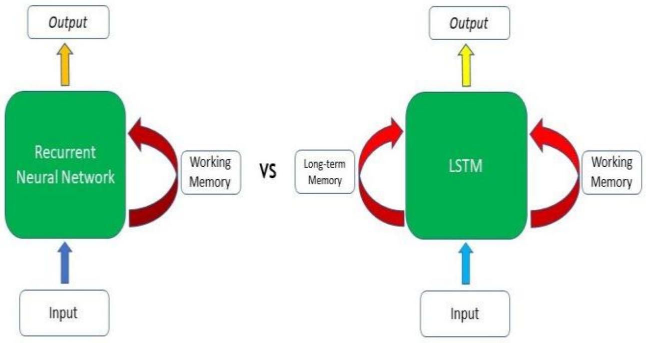


Figure 2: Simple RNN vs LSTM

The LSTM cell keeps three different part of data the current input data, the previous cell's short-term memory and the long-term memory at last.

The short-term memory is hidden state cell is shown fig 3.

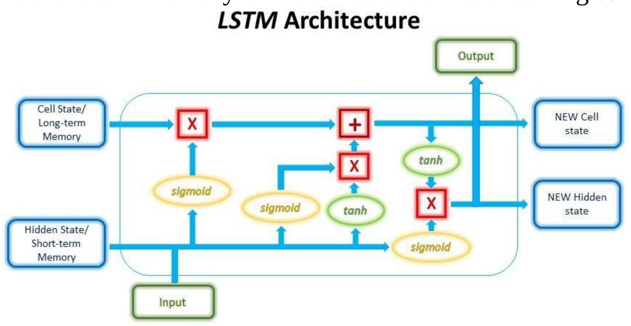


Figure 3 Working Priciple of the LSTM cell

The cell is uses as gates to standardize the data to be processed or discarded at each stage . These gates are called the Input Gate, the Forget Gate, and the Output Gate.

i)Input Gate

The input gate select key data from long- term memory that is the information from the function of previous stages short term memory and sequence of input. This information sent to the sigmoid function which is LSTM cell,and it create the layer sigmoid function sends information as 0's and 1's. The 0 indicates not related information and 1 indicates useful information.

$$i_1 = \alpha(W_{i1} . (H_{t-1}, x_t) + bias_{i1} \quad (3)$$

The next layer is short-term memory and the current input pass through an activation function to manage the network, commonly refer tanh function.

$$i_2 = \tanh\left(W_{i2} . (H_{t-1}, x_t) + bias_{i2}\right. \quad (4)$$

$$i_{input} = i_1 * i_2 \quad (5)$$

ii)Forget Gate

The gate forgets that data should be preserved or discarded from the long-term memory.

The forget vector is selected filter layer from the input layer. The short-term memory and output is send to sigmoid activation function to obtain the forget variable, similar to the first layer in the above Output Gate, but with different weights. The vector value is 0s and 1s and select which long-term memory to maintain.

$$f = \alpha(W_{forget} . (H_{t-1}, x_t) + bias_{forget}) \quad (6)$$

The outputs of Input gate and the forget gate is give point wise addition to new key of the long-term memory, which will be pass on to the next cell

$$C_t = C_{t-1} * f + i_{input} \quad (7)$$

iii) Output Gate

The output of the current stage can be derived from hidden state and it creates the third filter, the previous short-term memory and current input will be transfer to a sigmoid function again with different weights.

$$O_1 = \alpha(W_{output1} \cdot (H_{t-1}, x_t) + bias_{output1} \quad (8)$$

$$O_2 = \tanh(W_{output2} \cdot C_t + bias_{output2}) \quad (9)$$

$$H_t\ , O_t = O_1 * O_2 \quad (10)$$

The short-term and long-term memory designed by gates and it carried over to upcoming cell for the process to be recurring.

III. EXPERIMENTS AND RESULTS

A.Data Source and Representation

Protein sequence information is available in online databases. The main dataset lists peptide sequences and their corresponding secondary structures. It is a transformation of https://cdn.rcsb.org/etl/kabschSander/ss.txt.gz download from RCSB PDB [12] into a tabular structure.

1.The dataset consists of a 58673 sequence of amino acids. There is no redundancy in the list of protein sequences and the results obtained from experiments are evaluated. We have divided the obtained protein sequences into 46863 sequences for train data and 5923 for testing and the rest5 887 for validation model.

2.The datasets to sequences no longer than 128.

B. Experimental Results

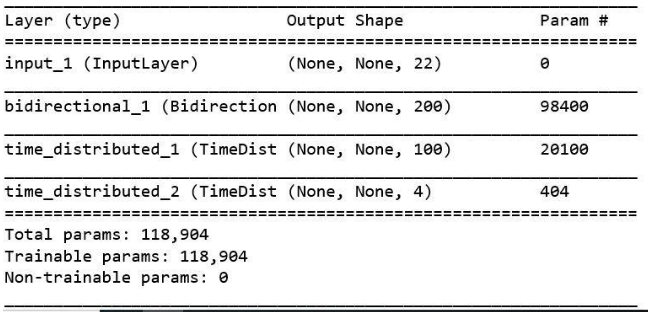
The result of Protein secondary structure prediction accuracy as shown in Table 1 and training and validation data curve is shown in fig (5,6,7).

C. Implementation

For this proposed work, Jupyter Notebook with python 3.0 is used for implementing the model. Scikit-learn libraries are mostly used for predictions with high performance. Scikit-learn is a machine learning library and it is an extension to Keras Library is used for the deep learning neural network implemented in this work.

Input the pre-processed sequence, pass it to an LSTM layer that returns a 100 dim sequence. This then goes through a Dense layer and finally a softmax is applied in order to predict the probability of each of the 3 states at every time step.

The architecture of the LSTM Model developed is as shown in Fig 4.



```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, None, 22)          0
_________________________________________________________________
bidirectional_1 (Bidirection (None, None, 200)         98400
_________________________________________________________________
time_distributed_1 (TimeDist (None, None, 100)         20100
_________________________________________________________________
time_distributed_2 (TimeDist (None, None, 4)           404
=================================================================
Total params: 118,904
Trainable params: 118,904
Non-trainable params: 0
_________________________________________________________________
```

Figure 4 Architecture of the model

In particular, repeated experiments with lower node quantity variation between secret layers would almost certainly result in a more successful model and various values for parameters, such as batch size and number of epochs, could well help the prediction process [9].

The Model is trained with different iteration and learning rate (Table 1 and 2).

Table 1 LSTM and Keras embedding Model with Epoch

LSTM	Epoch	Accuracy (%)	Training time
Model1	10	0.804	52 mins.
Model2	20	0.849	130 mins.
Model3	40	0.856	253 mins.

Table 2 LSTM and Keras embedding Model with Learning |Rate

LSTM	Epoch	Learning rate	Accuracy (%)
Model4	40	.001	85.6
Model4	**40**	**.002**	**86**

D. Learning curve For Model

Training system performance curves on the train and validation datasets used to test the under-fit, over-fit or well-fit model..
The learning curves for this approach are shown in figure (5,6,7 .) for the model1 and model2, model3.

The Learning curve LSTM shown in fig(5,6,7) for the model 1 and model2,model3.

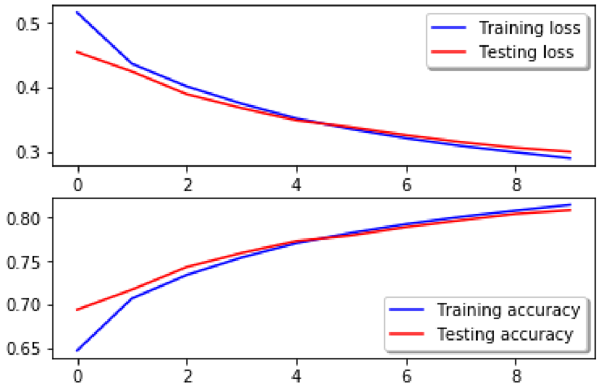


Figure 5

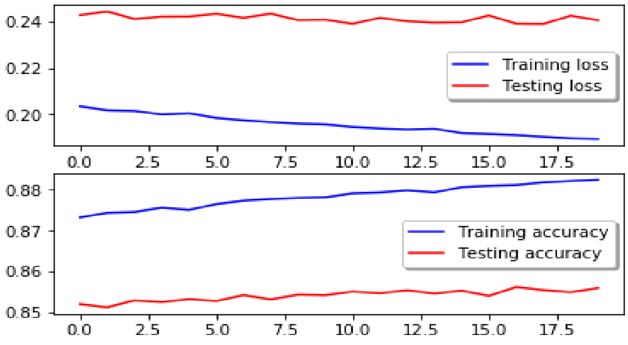


Figure 6

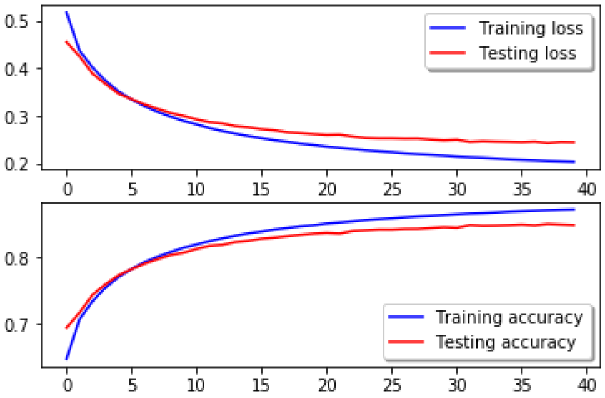
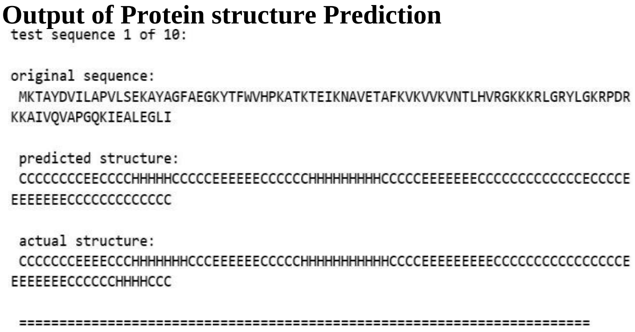


Figure 7

Output of Protein structure Prediction



```
test sequence 1 of 10:

original sequence:
 MKTAYDVILAPVLSEKAYAGFAEGKYTFWVHPKATKTEIKNAVETAFKVKVVKVNTLHVRGKKKRLGRYLGKRPDR
KKAIVQVAPGQKIEALEGLI

 predicted structure:
 CCCCCCCCEECCCCHHHHHCCCCCEEEEEEECCCCCHHHHHHHHHCCCCCEEEEEEECCCCCCCCCCCCCECCCCE
EEEEEEECCCCCCCCCCCCC

 actual structure:
 CCCCCCCEEEECCCHHHHHHHCCCEEEEEECCCCCHHHHHHHHHHHCCCCEEEEEEEEECCCCCCCCCCCCCCCCE
EEEEEEECCCCCCHHHHCCC

 =======================================================================
```

IV. DISCUSSION AND CONCLUSION

The LSTM has been trained with the RSCB dataset. Number of epochs and Learning rates of different models with LSTM are observed and compared in Table 1 and 2. The observation finds the model4 of LSTM reach high accuracy is 86% (Q3 Accuracy) with 40 epochs by learning rate of 0.002. In future work, The system aims to consider a bigger and deeper network to increase the accuracy of protein structure prediction. The Experimental obtained prediction accuracy of the proposed model is 86% and results are shown in the graphs.

REFERENCES

1. Cheng, Jianlin, Allison N. Tegge, and Pierre Baldi. "Machine learning methods for protein structure prediction." IEEE reviews in biomedical engineering 1 (2008): 41-49.
2. Schmidler, Scott C., Jun S. Liu, and Douglas L. Brutlag. "Bayesian segmentation of protein secondary structure." Journal of computational biology 7.1-2 (2000): 233-248.
3. Wang, Sheng, et al. "Protein secondary structure prediction using deep convolutional neural fields." Scientific reports 6 (2016): 18962.
4. Hua, Sujun, and Zhirong Sun. "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach." Journal of molecular biology 308.2 (2001): 397-407.
5. Guo, Jian, et al. "A novel method for protein secondary structure prediction using dual-layer SVM and profiles." PROTEINS: Structure, Function, and Bioinformatics 54.4 (2004): 738-743.
6. Asai, Kiyoshi, Satoru Hayamizu, and Ken'ichi Handa. "Prediction of protein secondary structure by the hidden Markov model." Bioinformatics 9.2 (1993): 141-146.
7. Aydin, Zafer, Yucel Altunbasak, and Mark Borodovsky. "Protein secondary structure prediction for a single-sequence using hidden semi-Markov models." BMC bioinformatics 7.1 (2006): 178.
8. Qian, Ning, and Terrence J. Sejnowski. "Predicting the secondary structure of globular proteins using neural network models." Journal of molecular biology 202.4 (1988): 865-884.
9. Spencer, Matt, Jesse Eickholt, and Jianlin Cheng. "A deep learning network approach to ab initio protein secondary structure prediction." IEEE/ACM transactions on computational biology and bioinformatics (TCBB) 12.1 (2015): 103-112.
10. Gers, Felix A., Nicol N. Schraudolph, and Jürgen Schmidhuber. "Learning precise timing with LSTM recurrent networks." Journal of machine learning research 3.Aug (2002): 115-143.
11. Graves, Alex, and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM networks." Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.. Vol. 4. IEEE, 2005.
12. Universal protein resource database. https://cdn.rcsb.org downloads. Accessed: 2018-01-05.

AUTHORS PROFILE

Thendral R, Research Scholar, Computer Science and Engineering, Annamalai University, India. She finished Master Engineering (CSE) in sathiyabama University. She has teaching experience 15 years. Her interested areas are Artificial Intelligence, Computer Graphics, Data Mining.

Dr. AN. SIGAPPI, received her Ph.D in Computer Science and Engineering from Annamalai University in 2013. She did her Master Degree in Computer science and engineering from Anna University. Currently she is serving as a Professor in the Department of Computer Science and Engineering, Annamalai University, India. Her areas of interest include Image Processing, Machine Learning, and Data Analytics. She has published more than 25 research articles in international journals and conferences.