Deep Recurrent Neural Network for Predicting the Protein Function Family from Sequence

THOTA RAJESHWAR RAO #[1]

[#1] B.Tech Student,

Department of Computer Science & Engineering,
Sikkim Manipal Institute of Technology, Majitar, East Sikkim-737136, India.

ABSTRACT

Now a day it is very faster and cheaper to collect the biological sequence data and hence the need to extract useful information from sequences is becoming a problematic situation, often limited by low throughput experimental characterizations. If we take protein sequence, accurate prediction of their functions directly from their primary amino acid sequences has been a long standing challenge. All the primitive mechanisms use to follow manual approach for identifying the pattern from primary amino acid sequence and lot of alignment problems and feature missing problems occurred due to manual approach. Hence this motivated me to design this article in which we try to use machine learning using artificial recurrent neural networks (RNN) was applied towards classification of protein function directly from primary sequence without sequence alignment, heuristic scoring or feature engineering. One of the best RNN model which we try to apply on our current problem is long short term memory (LSTM) units trained on public, annotated datasets from UniProt achieved high performance for in class prediction of four important protein functions tested, particularly compared to other machine learning algorithms using sequence derived protein features. Here we try to collect more than 450 protein family data and then try to apply our LSTM model to train on that family data. Once the LSTM model is trained on that appropriate data now we can test the performance of our given model by taking sample input data and then test how effectively our proposed model is going to categorize the protein family based on the input data.

Index Terms:

Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), Amino Acid Sequence(AAS), Heuristic Scoring, Protein Sequence.

1. INTRODUCTION

As the cost of DNA sequencing is decreasing drastically over the last decade, the volume of biological sequences particularly for new proteins is also increasing rapidly. Discovering the functions of these new proteins not only could allow one to better understand their roles in their native contexts, but also utilize them in synthetic biology to assembled new biological circuits and pathways for useful applications such as production of valuable compounds or treating disease. However, the experimental characterization of proteins' properties such as structure and function can be slow and resource demanding using techniques such as x-ray crystallography, cryoTEM, or functional assays, significantly outpaced by sequencing. A predictive pipeline that can accurately translate primary sequence to function would allow filtering of the vast sequence dataset to an experimentally manageable subset of high confidence candidates of highest interest toward a particular function or application is greatly desired.

Currently there are several popular methods for extracting useful information from primary sequences and infer functional information based on comparison of new sequences to existing sequences of known function. For example, BLAST performs sequence alignment with heuristic scoring. Multiple sequence alignments can be used to build models that capture conservation patterns (i.e. profiles or motifs), such as Position Specific Score Matrices (PSSM)1 or Hidden Markov Models2. These profiles can be used to iteratively search in a database search (e.g. PSIBLAST, jackhammer) to detect remote homologies, allowing the discovery of protein clusters or families that are evolutionarily related. New query sequences may be aligned to existing profiles for identification of function. The alignment scores and "Evalue" can help indicate the degree of homology between the new sequence and existing sequences. Powerful and popular as these existing approaches are for protein function annotation directly from sequence, there may still be limited in classifying sequences coding for proteins with similar function or structure but are very distant in evolutionary scale or have come to adopt similar function via convergent evolution. For instance the proteases have independently evolved the "catalytic triad" active site in 23 protein super families. The "catalytic triad" consists an acid base nucleophile configuration of three amino acids arranged in spatial proximity but can be distant on the sequence. Given the difficulty in accurate prediction of three dimensional protein folding, the "catalytic triad" is difficult to predict based on sequence alignment. Here I present

the application of machine learning using recurrent neural networks, recently gaining popularity and successes for natural languages processing, to capture high dimensional, complex patterns in biological sequences in order to predict protein functions, potentially beyond the capability of current methods

2. LITERATURE SURVEY

In this section we will mainly discuss about the background work that is carried out in order to prove the performance of our proposed Method. Literature survey is the most important step in software development process. For any software or application development, this step plays a very crucial role by determining the several factors like time, money, effort, lines of code and company strength. Once all these several factors are satisfied, then we need to determine which operating system and language used for developing the application. Once the programmers start building the application, they will first observe what are the pre-defined inventions that are done on same concept and then they will try to design the task in some innovated manner.

MOTIVATION

Two well known authors, Robert Rentzsch , & Christine A Orengo ,et.al, proposed a paper" Protein function prediction using domain families," and described SVM algorithm has less performance compared with SIMO algorithm for synthetic information extraction from imbalanced datasets. The authors concentrated on how to extract useful features from medical records and how to classify the unbalanced dataset by using some well known classification algorithms. But the experimental results clearly state that SVM is not able to achieve more accuracy for medical data diagnosis.The comparative results clearly state that SVM is having very less accuracy compared with SIMO algorithm which is used in this current paper.

A well known author, Diana Ekman ,et.al, proposed a paper "Multi-domain Proteins in the Three Kingdoms of Life: Orphan Domains and Other Unassigned Regions" and described that ccomparative studies of the proteomes from different organisms have provided valuable information about protein domain distribution in the kingdoms of life. Earlier studies have been limited by the fact that only about 50% of the proteomes could be matched to a domain. Here, we have extended these studies by including less well-defined domain definitions, Pfam-B and clustered domains, MAS, in addition to Pfam-A and SCOP domains. It was found that a

significant fraction of these domain families are homologous to Pfam-A or SCOP domains. Further, we show that all regions that do not match a Pfam-A or SCOP domain contain a significantly higher fraction of disordered structure.

3. EXISTING SYSTEM & ITS LIMITATIONS

In the existing system there was no proper method to identify the protein sequences easily. Deciding Amino acid based on proteins sequence is becoming a predominant task in existed system there are lot of machine learning algorithms used to detect amino acid type based on protein sequence but Machine learning algorithms will have their own limitations hence they won't give much accurate results for complex data .

The following are the main limitations in the existing system.

1. More Time Delay in finding the protein sequence and its family.
2. There is a huge delay in the calculating the protein sequences.
3. There is no appropriate technique to identify the protein sequences and its families.

4. PROPOSED SYSTEM & ITS ADVANTAGES

In proposed system we are applying deep learning model for protein classification from DNR sequence. In this project we are building a LSTM architecture to perform amino acid detection based on protein sequence in this project we took the data set from kaggle and preprocessed the data by taking help of pandas module and we understood the data bit clearly by performing exploratory data analysis by taking help of seaborn and matplotlib libraries and to understand the protein sequences to the machine we took help of NLP and we preprocessed the text in the form of vectors and finally by applying designed LSTM architecture we are checking model performance and with the designed model we are predicting type of amino acid based on given custom protein sequence.

The following are the advantages of the proposed system. They are as follows:

1) By using LSTM model we can able to check the protein sequence structure and classify easily.

2) In this paper we use multiple CNN models and check the accuracy of best algorithm among many algorithms.

3) This is very accurate in classifying the protein sequences.

5. PROPOSED LSTM MODEL

The predicted function is eventually validated by experimental assay. Furthermore, RNN models could predict certain phylogenetically distinct “outofclass” protein families with similar function, albeit with worse sensitivity and selectivity. The recurrent neural network (RNN) model contains one or more sets of “bidirectional” recurrent layers with longshort term memory (LSTM) neurons processing the input sequence one residue or character at a time (Figure 1b). The forward layer scans the protein sequences from the N towards the C terminus and reversed for the backward layer, allowing the network to make use of context on both sides of each position rather than just what was seen before in a single direction. Each residue in the input protein sequence is converted into a “onehot” vector whose elements are all 0 except at the position of the amino acid it corresponds to, where it is set to 1.

Each LSTM neuron in each recurrent layer uses input “i”, output “o”, gate “g”, and forget “f” gates to modulate the input vector and update the neuron’s internal cell state “c” and hidden state “h”. The gates apply matrices, whose elements are adjustable parameters to be learned, on the input and hidden state vectors at each recurrent step/layer and subsequently normalize the results with nonlinear activation functions (Methods). Intuitively, given each new input vector (i.e. sequence residue), the gates control what and how much to add to and output from the hidden state memory, which encodes sequence patterns relevant toward particular protein function (Figure 1c). These features of the LSTM architecture allow the RNN to maintain, over many recurrent iterations, the magnitudes of both the relevant signals in feed forward propagation as well as the error gradients in backpropagation, thereby resolving the issues of loss of contextual memory and vanishing/exploding gradients that have limited the usefulness of traditional RNNs in processing long sequences (e.g. hundreds of units/iterations).

The outputs from the last LSTM neurons of the forward and reverse hidden layers are eventually fed to a fully connected layer of artificial neurons, where each neuron represents one functional class and outputs via the “softmax” activation function the probability that the input

sequence represents a particular functional class. The number of recurrent hidden layers, LSTM neurons in each layer, hidden units (i.e. hidden state vector dimension) in each LSTM neuron, and the architecture of each LSTM neuron (e.g. “peephole” connections”) are hyperparameters that can be optimized. For example, stacking several recurrent layers by feeding the output of each LSTM neuron in one recurrent layer as input into an adjacent recurrent layer, or increasing the number of neurons and hidden units, enable more complex or hierarchical representations at the risk of over fitting. Furthermore, the number of output neurons, which represents the number of functions to be simultaneously considered (i.e. multiplex), can be varied. In this work, a single set of bidirectional recurrent layers was utilized for inclass predictions, and up to three sets were used for training toward outofclass predictions.

As protein sequences vary widely in length, the number of LSTM neurons in the recurrent layer was capped, typically at 333 representing a maximum of 333 amino acids sequence or around 1 kilobase of DNA. For proteins smaller than 333 residues the sequence was prepadded with 0’s up to a 333digit sequence, where the digits 1 to 20 represents the 20 canonical amino acids. For functions with mostly large proteins such as the CRISPR associated nucleases, up to 800 N terminal amino acids were input for training, and subsequently the same RNN model was trained on up to 800 Cterminal amino acids. There are 128 hidden units in each LSTM neuron (i.e. the hidden state is represented by a 128 element vector). A high dimensional hidden state vector can encode more information to represent more complex function related sequence features. This can be an advantage compared to some sequential models (e.g. hidden Markov model) with limited number of internal states at the expense of interpretability.

Additionally, the multiple nonlinear operators of the LSTM (e.g. activation functions) allow complex updating of the hidden state memory. Adding to this flexibility, the probability of “Dropout”, the random severing of connections between layers, was consistently set to 0.5. Unlike previous artificial neural network based methods, the LSTM model here does not limit itself to learning short “profiles” or motifs of predefined length3–5 (e.g. 21 amino acid window3) but instead learns from the entire sequence up to a maximum length (e.g. 333, 500 or 800 from each terminus) in order to capture potentially longrange patterns.

6. IMPLEMENTATION PHASE

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. The front end of the application takes Google Collaboratory and as a Back-End Data base we took UCI Heart Patients Records as dataset. Here we are using Python as Programming Language to implement the current application. The application is divided mainly into following 5 modules. They are as follows:

1. Import Necessary Libraries
2. Load Dataset Module
3. Data Pre-Processing
4. Train the Model Using CNN Models.

1. IMPORT NECESSARY LIBRARIES

In this module initially we need to import all the necessary libraries which are required for building the model. Here we try to use all the libraries which are used to convert the data into meaningful manner. Here the data is divided into numerical values which are easily identified by the system, hence we try to import numpy module and for plotting the data in graphs and charts we used matplot library.

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

```
[ ] df=pd.read_json('News_Category_Dataset_v2.json',lines=True)
    df.head()
```

	category	headline	authors	link	short_description	date
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...	Melissa Jeltsen	https://www.huffingtonpost.com/entry/texas-ama...	She left her husband. He killed their children...	2018-05-26
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...	Andy McDonald	https://www.huffingtonpost.com/entry/will-smit...	Of course it has a song.	2018-05-26
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 57	Ron Dicker	https://www.huffingtonpost.com/entry/hugh-gran...	The actor and his longtime girlfriend Anna Ebe...	2018-05-26
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...	Ron Dicker	https://www.huffingtonpost.com/entry/jim-carre...	The actor gives Dems an ass-kicking for not fi...	2018-05-26

2. LOAD DATASET MODULE

In this module the we try to load the dataset which is downloaded or collected from Kaggle repository. Here we store the dataset names as `protein-dataset` file and this dataset contains the following information such as :

Each and every attribute contains some information which are tested and collected based on individual patient id.

3. DATA PRE-PROCESSING MODULE

Here in this section we try to pre-process the input dataset and find out if there are any missing values or in-complete data present in the dataset. If there is any such data present in the dataset, the application will ignore those values and load only valid rows which have all the valid inputs.

4. TRAIN THE MODEL USING CNN MODEL AND FIND OUT APPROPRAITE CLASS NAME

Here we try to train the current model on given dataset using Proposed LSTM Model and then try to find accuracy of that LSTM model and can able to predict the type of classname.

```
from tensorflow.keras.layers import Dense,Dropout,Embedding,LSTM,Bidirectional
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential

model=Sequential()
model.add(Embedding(len(list(unique_words)),300,input_length=len_max))
model.add((LSTM(128,dropout=0.5, recurrent_dropout=0.5,return_sequences=True)))
# model.add(Bidirectional(LSTM(128,dropout=0.5, recurrent_dropout=0.5,return_sequences=True)))
model.add((LSTM(64,dropout=0.5, recurrent_dropout=0.5,return_sequences=False)))
#model.add(LSTM(32,dropout=0.5, recurrent_dropout=0.5,return_sequences=False))
model.add(Dense(100,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.01),metrics=['accuracy'])
model.summary()
```

7. EXPERIMENTAL REPORTS

In this application we try to use Google collab as platform and take LSTM model as training model to predict the protein sequence family.

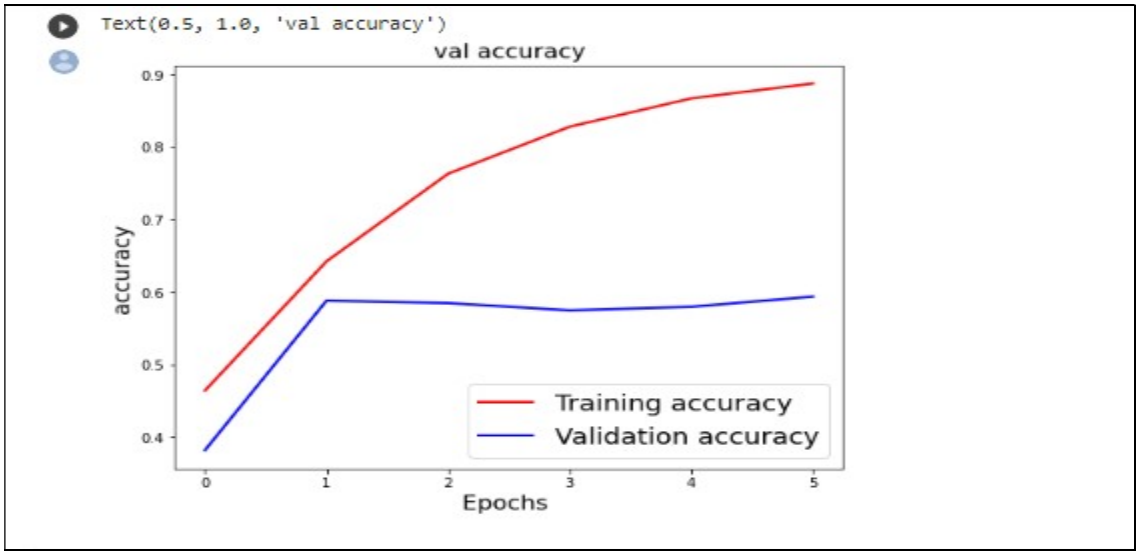
Apply LSTM Model

```
from tensorflow.keras.layers import Dense,Dropout,Embedding,LSTM,Bidirectional
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Sequential

model=Sequential()
model.add(Embedding(len(list(unique_words)),300,input_length=len_max))
model.add((LSTM(128,dropout=0.5, recurrent_dropout=0.5,return_sequences=True)))
# model.add(Bidirectional(LSTM(128,dropout=0.5, recurrent_dropout=0.5,return_sequences=True)))
model.add((LSTM(64,dropout=0.5, recurrent_dropout=0.5,return_sequences=False)))
#model.add(LSTM(32,dropout=0.5, recurrent_dropout=0.5,return_sequences=False))
model.add(Dense(100,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.01),metrics=['accuracy'])
model.summary()
```

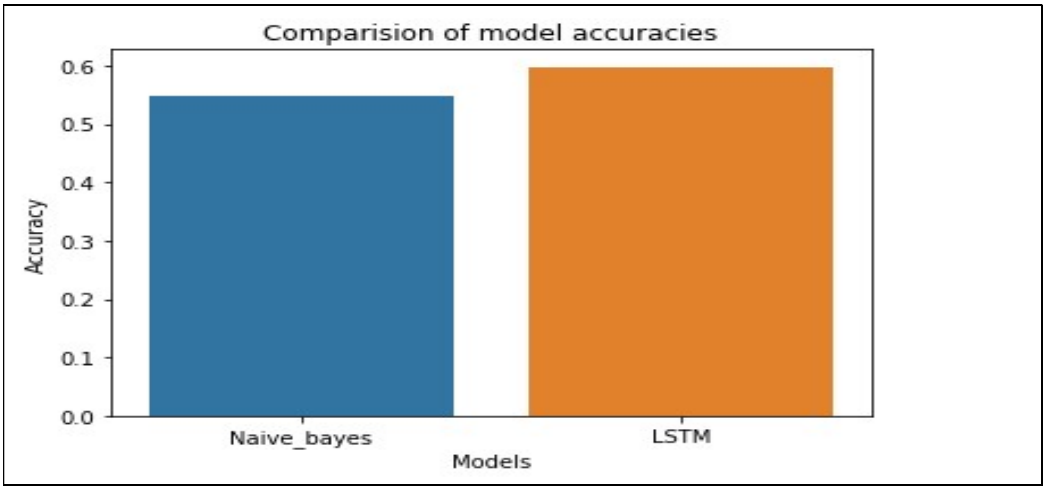
In the above window we can clearly say LSTM and Bi-LSTM are applied to the current application to train the sequences and find out sequence comes under which family group.

Validation GRaph



From the above window we can clearly see the validation graph for our current model which is trained and verified the accuracy. If we observe clearly we can see training accuracy is more than validation accuracy. Hence we can conclude that our proposed model achieved very good accuracy for identifying the protein sequence accurately.

Model Comparison Graph



From the above window we can clearly identify that LSTM is achieved more accuracy compared with Naïve bayes classification algorithm. Hence we can conclude that current application achieved good accuracy with LSTM.

8. CONCLUSION

In this proposed article we used recurrent neural network (RNN) based on LSTM can be trained to classify certain protein functions with high level of accuracy from input amino acid sequences alone. Experimental validation of the predicted iron sequestering or mineralizing proteins including some currently not easily identified by other bioinformatics methods confirm the accuracy and utility of the model for prediction. Compared against popular sequence prediction and analysis tools such as BLAST and HMMER, the RNN model currently has several potential benefits but also limitations. Hence we try to conclude that our proposed LSTM model can achieve good accuracy for predicting the type of protein based on sequence data collected from sample data. Once the type of protein is identified now we can able to observe the family type and can predict the family accurately. As a future work we can extend the same application by using some more advanced models to achieve more accuracy.

9. REFERENCES

1. Altschul, S. F. et al. Gapped BLAST and PSIBLAST: a new generation of protein database search programs. Nucleic Acids Res 25, 3389–3402 (1997).

2. Finn, R. D., Clements, J. & Eddy, S. R. HMMER web server: Interactive sequence similarity searching. Nucleic Acids Res. 39, 29–37 (2011).

3. Hochreiter, S., Heusel, M. & Obermayer, K. Fast model based protein homology detection without alignment. Bioinformatics 23, 1728–1736 (2007).

4. Wu, C., Berry, M., Shivakumar, S. & McLarty, J. Neural Networks for FullScale Protein Sequence Classification: Sequence Encoding with Singular Value Decomposition. Mach. Learn. 21, 177–193 (1995).

5. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA and RNA binding proteins by deep learning. Nat Biotechnol33, 831–838 (2015).

6. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG] 1–15 (2014).

7. Jutz, G., Van Rijn, P., Santos Miranda, B. & Boker, A. Ferritin: A versatile building block for bionanotechnology. Chem. Rev. 115, 1653–1701 (2015).

8. Liu, X. et al. Engineering Genetically Encoded Mineralization and Magnetism via Directed Evolution. Scientific Reports. 6, 38019 (2016). doi:10.1038/srep38019.

9. Roy, A., Kucukural, A. & Zhang, Y. ITASSER:a unified platform for automated protein structure and function prediction. Nat Protoc 5, 725–738 (2010).

10. Hunter, S. et al. InterPro: The integrative protein signature database. Nucleic Acids Res. 37, 211–215 (2009).

11. Zetsche, B. et al.Cpf1 is a single RNA guided endonuclease of a Class 2CRISPRCas system. Cell 163, 759–771 (2015).

12. Jinek, M. et al. A Programmable DualRNA–Guided DNA End nuclease in Adaptive Bacterial Immunity. Science. 337, 816–822 (2012).

13. Cong, L. et al. Multiplex Genome Engineering Using CRISPR/Cas Systems. Science (80-.). 339, 819–824 (2013).

14. Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R. & Schmidhuber, J.LSTM: A Search Space Odyssey. arXiv:1503.04069 (2015).

15. LeCun, Y. et al. Deep learning. Nature 521, 436–444 (2015).

16. Dzmitry Bahdana, Bahdanau, D., Cho, K. & Bengio, Y. Neural Machine Translation By Jointly Learning To Align and Translate. arXiv:1409.0473 (2014).

17. Wu, Y. et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv:1609.08144 (2016).

18. Sønderby, S. K., Sønderby, C. K., Nielsen, H. & Winther, O. Convolutional LSTM networks for subcellular localization of proteins. arXiv:1503.01919 (2015).