Performance analysis of Deep Learning based Enzyme Classifiers for Selecting Optimum Design Space

Sakib Ferdous*
Department of Biology
Iowa State University
ferdous@iastate.edu

Ibne Farabi Shihab*
Department of Computer Science
Iowa State University
ishihab@iastate.edu

***Abstract*— Classifying a particular protein class has turned out to be a extreme important factor nowadays, in the field bio-science . In this circumstance, We are proposing deep learning based protein sequence classifier. Here, the classifier will be trained on a dataset consisting 37000 annotated protein sequence with 4 types of deep neural nets – first with only embedding layer, then CNN, LSTM and lastly, with a combination of LSTM and embedding layer. The performance of each type of algorithm is presented with validation score and confusion matrix. In this work we found greatest success with LSTM based classifier. However, the CNN based classifier seems to over fit and lots of fluctuation in validation score is being observed. We have also baselined the two well established EC classification tool for length, EC type and composition.**

Keywords Protein classification, CNN, sequence classification

I. Introduction

Proteins are chains of amino acids arranging themselves along the backbone alpha carbons and folds to give 3D structures like alpha helix of beta sheets. The function and efficacy of proteins largely depends on the arrangement of amino acids which results in three dimensional structures. There are different zones in the structure with the help of which proteins are able to carry out many functions, like breaking bonds, forming bonds, carrying out reactions etc. Proteins are classified based on the type of task they perform. Classifying proteins based on their attributes is called the protein classification problem.

Protein is produced in cells by a series of complex mechanism. Generally speaking, ribosome produces the single stranded template called RNA from DNA inside chromosome. From this RNA Protein is produced. The different nature of produced proteins defines the diversification between and beyond organisms. So it can be said that, the type of proteins that are produced depends solely on DNA.Once proteins are formed, they take 3D structure. This 3D structure depends on the charges and different interplaying forces on the amino acid residues. There are different regions in the structure that are active by which proteins perform certain task.

Recently there has been a very rapid development in the field of Natural Language Processing in Artificial Intelligence. Using technique like embedding layer or n-gram NLP algorithms can be accommodated to classify texts. One of the most common examples includes classifying movie reviews in IMDB dataset. Taking the concept from the example These NLP algorithms can also be used to the protein classification problem, where the sequence of proteins is input as text. The idea here is that after training from a large pool of labelled data the model will learns to classify proteins solely based on sequence data.

In this work, we have proposed a CNN based model classify proteins. The rest of the paper will be arranged as experimental setup, literature review and will be be concluded by results followed by conclusion.

II. Experimental Setups

A. Data Acquisition

In this work we are using the Kaggle 'Structural Protein Sequence' dataset. The dataset consists of one hundred forty thousand labelled protein sequence data. Then the data should be divided into two categories, - train set and test set. Train set is also divided into two sets – model building set and validation set.

B. Choosing of model and problem formulation

Going further into the problem, for sequential text classification problems RNN or recurrent neural networks is one of most widely used algorithms. However, no wonder that convolution neural nets are also being used alongside RNN to boost up the efficiency.

Traditionally there are few approaches to deal with string type data. One way is to, converting string type of data to 'one-hot encoding' or array of binary numbers before entering the network. In addition to, utilize the

neighboring effect of the amino acids 'k-mers' conversion is also a very fruitful approach where words are converted into k set of words.

Scikit Learn' is an excellent tool for separating and processing the data which is being used here. For neural network based models two deep learning libraries were formerly used – Tensorflow and Keras. In this work, we are using the latest one named Pytorch. Instead of sticking with one model, we came with the idea of using few models and figure out which one works the best for serving our purpose.

First we will attempt only word embedding with sequential model. Next we will bring several reformations on the model in the following order - 1. Convolution layer addition CNN 2. Long short term memory (embedding + LSTM) layer 3. LSTM + Dense layer 4. LSTM + CNN, CNN + LSTM

In the protein classification problem our goal is to train a neural network so that it can classify proteins based on the sequence data only. In the Kaggle dataset competition highest accuracy obtained was 60% using convolutional neural net. From the different combination of the models (described in previous section), and tuning hyperparameters we expect to get validation set accuracy greater than 70%.

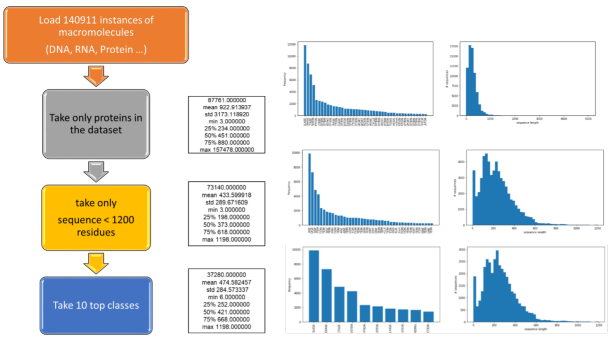

Fig. 1. Pictorial view of data pre-processing

C. Data Preprocessing

The first step in building a model is data preprocessing.Firstly, we trim the dataset to extract the only protein sequences. The raw data are input and their properties are observed with the help of 'Pandas' and other statistical visualization tools matplotlib and seborn[10]. The most important criteria in classification tasks is even distribution of the data classes. If the data properties are skewed, some data are dropped or resampled to get a well distributed dataset[Figure 1].

TABLE I
RESULT OF DIFFERENT MODEL

Model Structure	Training accuracy	Testing accuracy
WE with SM	81.5%	∼51.8%
CNN	91%	∼60%
CNN+LSTM	10.2%	10.3%
LSTM+embedding	73.1%	70.3%
LSTM+embedding+dense	37.5%	36.9%

III. LITERATURE REVIEW

There are dedicated search tools called Blast already available for the purpose of protein classification based on sequence similarity. [1] But this techniques become inadequate as the search space is increasing day by day for huge amount of data produced by Next generation sequencing tools. The reason for attempting machine learning techniques on this dataset is due to recent interest of applying AI in biological sequence data. Specifically, text based natural language processing (NLP) showed great promise in analyzing biological sequence. [2]Google research team recently solved the classification problem with 17929 protein classes (here we used only 10) which outperformed conventional search tools. [3]

Jing and coworkers recently came up with biological sequence classification toolkit called AutobioSeqPy which works by CNN and bi directional LSTM layer. [4] The UDSMProt framework took this one step further by adding Enzyme class prediction, homology detection and fold detection on Swissprot Database. [5] DeepFam is another protein family modelling methods.

It is of great interest to observe how development in NLP techniques contributes to the bioinformatics field. There are several notebooks in Kaggle where the classification problem was solved with the help of Tensorflow and Keras with CNN layers. In most cases the highest accuracy they get on the validation set is about 60%. Here in this project we are going to use Pytorch models and will make the model increasingly complex to observe the effect on accuracy. After this we will observe the effect of different string manipulations. This will offer us valuable insights on which regions in a protein sequence is most important in determining its class. Afterwards we will apply GPT, a generative neural network for producing novel protein sequences.

A. Experimental Methodology

Initially we started with only an embedding layer followed by batch normalization. It did perform really well on training phase but clearly it was overfitting it we look at the Figure 2. Then, we moved to a start Alex net with 5 convolution layer as CNN model. In

terms of training, it did really well though it showed the same issue of overffitng like the previous one[Figure 4,5]. Next idea was to try and see how a LSTM works with CNN. For this case, we added a LSTM layer after the final Convulation layer. This model performed really poor which can be found Figure 6. Later we tried to use LSTM layer with an embedding layer which we found out to be the best model as it did not show any kind of overfitting[Figure 8]. We also tried to use dense layer with our best one but it performed poorly[Figure 10].

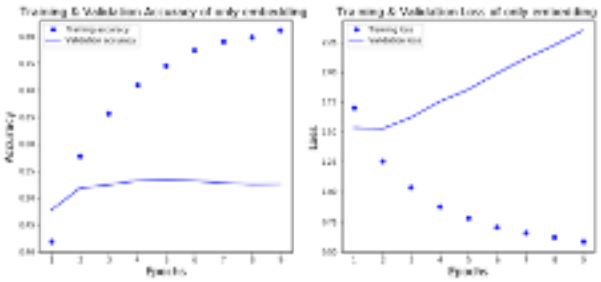

Fig. 2. Learning curve and loss of embedding

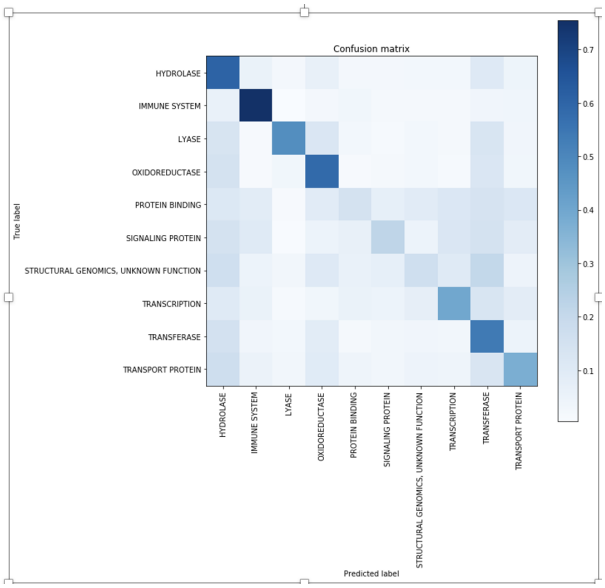

Fig. 3. confusion matrix of embedding

IV. Results

We put confusion matrix and pictorial view of our experimental models above to have an idea what we did[Figure 3-11]. In the experimental section we just described the models with verdict but here in section, we will try put reasoning behind those. In addition to that the classification result will be described here. We first

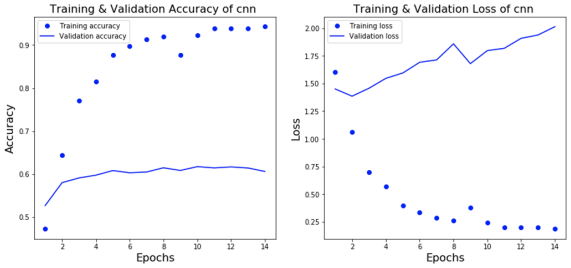

Fig. 4. Learning curve and loss of CNN

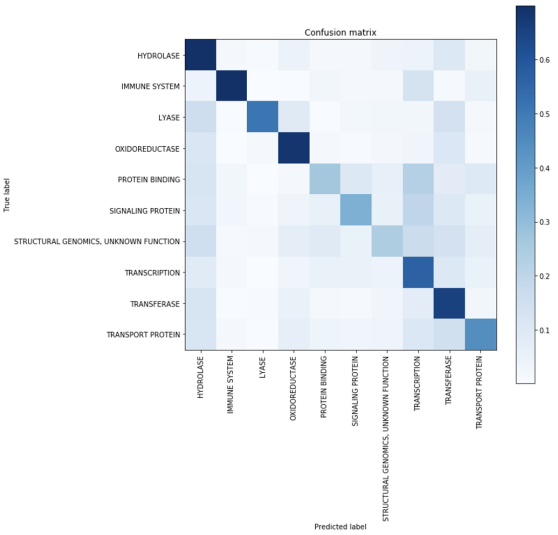

Fig. 5. confusion matrix of CNN

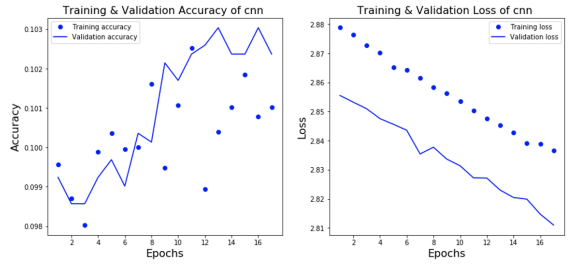

Fig. 6. Learning curve and loss of CNN+LSTM

followed the traditional way of sequence classification which is embedding. We got the good training accuracy result though the testing accuracy was on the lower side. This indicated that our model got overfitted and too long sequence can be one of the reasons for this as the model might have hard time to generalize. Then we tried a simple 5 layered CNN where we get a very good training accuracy though the testing is poor as

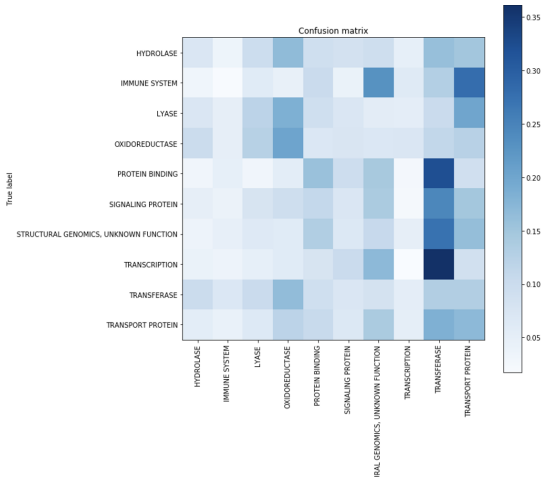

Fig. 7. confusion matrix of CNN+LSTM

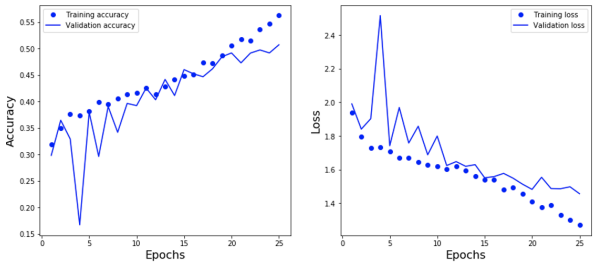

Fig. 8. Learning curve and loss of LSTM+embedding

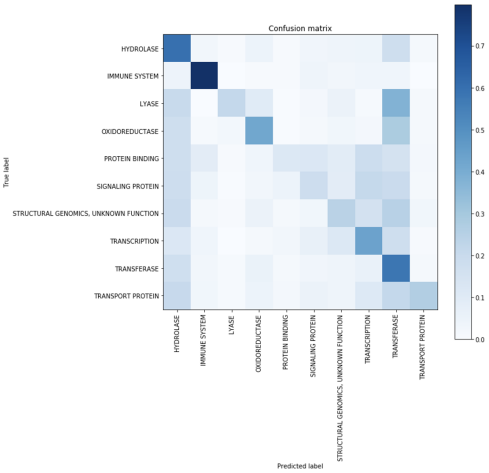

Fig. 9. confusion matrix of LSTM+embedding

before. We assumed the same reason for this model to as we did not go deeper for this and also the CNN had hard time to have an idea about the relation among the sequence.Then, we tried different variation

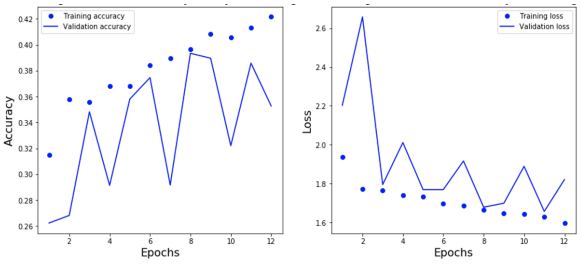

Fig. 10. Learning curve and loss of LSTM+embedding+dense

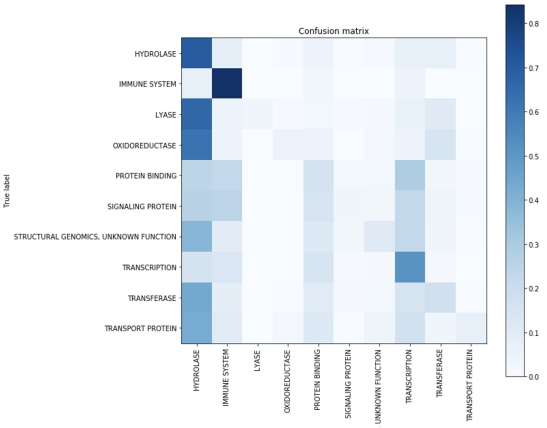

Fig. 11. confusion matrix of LSTM+embedding+dense

Layer (type)	Output Shape	Param #
embedding_29 (Embedding)	(None, 400, 64)	1664
lstm_6 (LSTM)	(None, 128)	98816
batch_normalization_40 (Batc	(None, 128)	512
dense_43 (Dense)	(None, 10)	1290

Total params: 102,282
Trainable params: 102,026
Non-trainable params: 256
None

Fig. 12. layerwise diagram of LSTM+embedding

and mixture of CNN and LSTM and from there we found that LSTM+embedding did a really good job. The idea behind this was that as we know sequence has a relation and they do follow some certain patterns, LSTM learned that when we passed that using embedding. In addition to, the batch normalization helped us to reduce the overfitting. Thus, from this experiment we found that LSTM+embedding is the best among all with around 70% accuracy. More detailed result has been shown in

[Table I].

V. Conclusion

In this work, we presented a deep learning based classification model for protein classification. We showed that a LSTM model with embedding layer worked best for serving our purpose with testing accuracy of 70.3%. In case of future plan,we can try k-mers or bag of words technique to modify the sequence. ON top of that, to take it to more advanced level, we will use Generative Adversarial Network to produce novel proteins of a specific type. The novel proteins will be compared with the test set to see their match with the specified type with local alignment score. They will be also searched through protein data bases to see their match with specific types.

References

[1] Z. Wu, "Data-Driven Protein Engineering Thesis by," 2021.

[2] M. Tsubaki, K. Tomii, and J. Sese, "Compound-protein interaction prediction with end-toend learning of neural networks for graphs and sequences," Bioinformatics, vol. 35, no. 2,pp. 309–318, 2019, doi: 10.1093/bioinformatics/bty535.

[3] M. L. Bileschi et al., "Using Deep Learning to Annotate the Protein Universe," bioRxiv, pp.1–29, 2019, doi: 10.1101/626507.

[4] R. Jing, Y. Li, L. Xue, F. Liu, M. Li, and J. Luo, "AutoBioSeqpy: A Deep Learning Tool for the Classification of Biological Sequences," J. Chem. Inf. Model., vol. 60, no. 8, pp. 3755–3764, 2020, doi: 10.1021/acs.jcim.0c00409.

[5] N. Strodthoff, P. Wagner, M. Wenzel, and W. Samek, "UDSM-Prot: Universal deep sequence models for protein classification," Bioinformatics, vol. 36, no. 8, pp. 2401– 2409, 2020, doi: 10.1093/bioinformatics/btaa003.

[6] N. Anand, P. Huang, "Generative modeling for protein structures", NIPS,2018.

[7] (Yu et al., 2017)Yu, L., Zhang, W., Wang, J., Yu, Y. (2017). SeqGAN: Sequence generative adversarial nets with policy gradient. 31st AAAI Conference on Artificial Intelligence,AAAI 2017, 2852–2858.

[8] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley, and C. Whitley. 2002. A Comparison of Genetic Sequencing Operators. (August 2002).

[9] A. E. Eiben and James E. Smith. 2015. Introduction to Evolutionary Computing (2nd ed.). Springer Publishing Company, Incorporated.

[10] Matplotlib: Visualization with Python,accessed 9 December 2021, ¡https://matplotlib.org/¿