Classification of B-Cell Epitopes through Deep Proteomics

John Kevin Lopez Cava

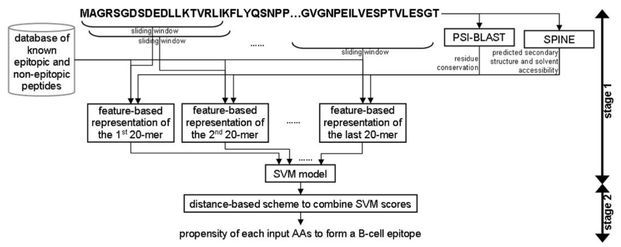
Introduction (Problem Statement)

The immune system is a powerful tool that an organism uses in order to recognize foreign bodies - also known as antigens - through the use of antibodies. Once such recognition occurs, the immune system can signal immune cells in order to eliminate the antigen from the organism. Then if the organism gets in contact with the antigen again, antibodies can then recognize the antigen and give an appropriate response.

However, there are limitations with the immune system. In order for the immune system to recognize a pathogen, the organism has to be infected. Moreover, even if the immune system recognizes a pathogen, the pathogen may be able to mutate and then escape detection from the immune system. It would be beneficial if a pathogen can be isolated, and a vaccine can be synthesized just from the antigen sequence.

Previous Work

Previous work done utilizes a a sliding window of 20-mers of protein sequence that is then fed into PSI-BLAST and SPINE. PSI-BLAST provides the residue conservation, while SPINE provides the secondary structure of the sequence. The features of the 20-mers are then processed into a Support Vector Machine that then gives scores to the amino acids. These scores then determine if this amino acid in the sequence is a B-Cell epitope or not.



Methods

This poster proposes utilizing a method called Continuous Distributed Representation in order to create “deep embeddings of protein sequences that turns them into vectors which then can be used in deep learning. Another method that is used is called a “one hot encoding”, in which each amino acid in the protein sequence is encoded uniquely by an array of numbers. Both encoding methods will then be inputted into a different deep learning model. The former method will utilize a convolutional neural network, while the latter method will utilize a Long Short Term Memory neural networks (LSTMs).

The model that implements Continuous Distributed Representation focuses on the idea of proteomics in order to vectorize protein sequences. By using proteomic data, a neural network model is able to vectorize 3-mer protein sequences into 100 element vector. This vector is then an embedded representation of the sequence.

The model with the one-hot encoding utilizes a LSTM because the model takes in each amino acid one by one. A LSTM has the ability to take into consideration previous input, which then provides an output for each following amino acid in the sequence. For this model, the last output after reading in the last amino acid would determine if the sequence is an epitope or not. In addition, two different LSTM models are utilized - one with 1 layer, and the other with 3 layers. This is in order to determine if more layers will provide accurate results.

In addition, the training parameters - in particular, the learning rate - are varied in order to also determine if any change on those parameters will improve or decrease the accuracy of the model.

Results

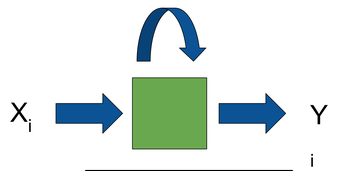


Figure A. LSTM 1 Layer

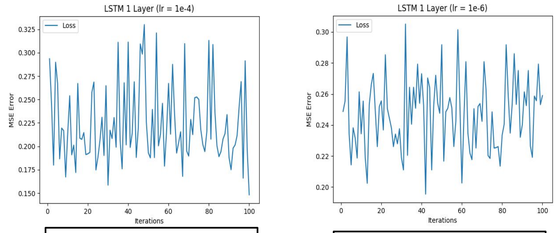


Figure D. Loss of LSTM 1 Layer with lr = 1e-4

Figure E. Loss of LSTM 1 Layer with lr = 1e-6

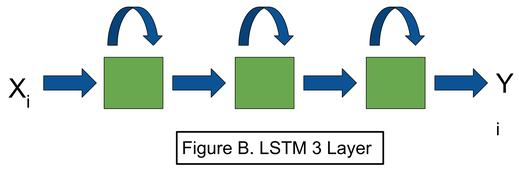


Figure B. LSTM 3 Layer

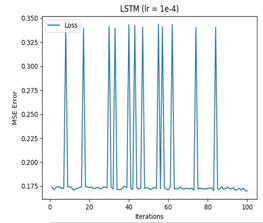


Figure F. Loss of LSTM 3 Layer with lr = 1e-4

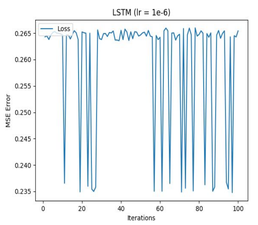


Figure G. Loss of LSTM 3 Layer with lr = 1e-6

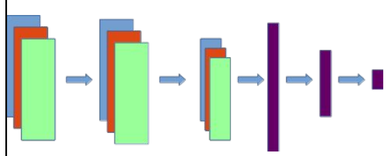


Figure C. Convolutional Neural Network

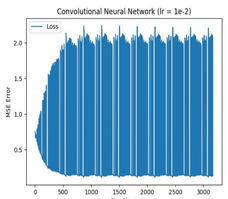


Figure H. Loss of CNN with lr = 1e-2

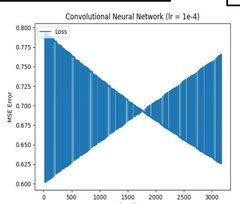


Figure I. Loss of CNN with lr = 1e-4

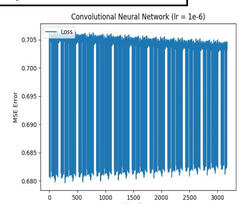


Figure J. Loss of CNN with lr = 1e-6

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
LSTM 1 Layer lr=1e-4	0.14	0.11	0.13	0.16	0.14	0.14
LSTM 1 Layer lr=1e-6	0.12	0.14	0.12	0.12	0.12	0.12
LSTM 3 Layer lr=1e-4	0.15	0.13	0.14	0.12	0.12	0.13
LSTM 3 Layer lr=1e-6	0.16	0.16	0.11	0.14	0.11	0.13
CNN lr=1e-2	**0.84**	**0.85**	**0.84**	**0.88**	**0.84**	**0.85**
CNN lr=1e-4	0.87	0.88	0.14	0.86	0.88	0.73
CNN lr=1e-6	0.14	0.86	0.12	0.12	0.85	0.42

Table 1. Different variations of neural networks (LSTM vs. CNN, different learning rate) were tested on a test dataset. Each neural network model was tested 5 times, which are then averaged. The convolutional neural network with learning rate of 1e-2 had the most accurate prediction compared to the different models. Moreover, there is a significant difference in accuracy between LSTM and CNN models.

SEQ194 Test Dataset Evaluation

Specificity = 0.0
Sensitivity = 0.99

References

Gao, Jianzhao, et al. "BEST: improved prediction of B-cell epitopes from antigen sequences." *PloS one* 7.6 (2012): e40104.

Asgari, Ehsaneddin, and Mohammad RK Mofrad. "Continuous distributed representation of biological sequences for deep proteomics and genomics." *PloS one* 10.11 (2015): e0141287.

Conclusion

Both methods proposed in this poster have significant limitations towards its intended use. The LSTM model that utilized the one-hot encoding embedding representation had poor results. For each different LSTM used, the overall accuracy of the LSTMs were around 12-14%. This indicates that either one-hot encoding is not an effective embedding strategy or that the LSTM is not able to learn the recurrent structure of the sequence. It can also be a combination of the two. However, it can be argued that the number of iterations that the LSTMs were trained on were indeed signficantly less than the amount of iterations that the convolutional neural networks were trained on (100 iterations vs. 3000 iterations) The main issue with training LSTMs is that these models are much more computationally expensive, and as such training a convolutional neural network is much faster than an LSTM. Though, looking at the training loss in the figures for the LSTMs, it can be safe to conclude that the models weren’t learning the data as the loss isn’t converging.

In many aspects, it can be also concluded that all models had a difficulty in learning the data, as the loss as a function of iterations are eratic and are not converging. This includes the convolutional neural network. While the CNN was able to adequately classify which 20-kmer protein sequences are epitopes, it completely failed to provide any useful information in regards to the SEQ194 dataset. This is evident with the ROC results and how sensitivity was zero. Thus, the model only predicted one label for all the amino acids in the sequence, and did not adequately provide any distinction between B-Cell epitope or non B-Cell epitope.

Overall, in terms of determining if a 20-kmer protein sequence is a B-Cell epitope, the results showcase that a convolutional neural network with continuous distributed representation outperforms a one-hot encoding representation with a LSTM. However, a convolutional neural network can’t provide any information in predicting from a variable sequence. It can be considered that the model overfited the data, and thus didn’t distinguish any B-Cell epitopes fro the SEQ194 dataset. As such, there should be an emphasis on designing other neural network architectures that can adequately classify these sequences.

Future Work

One consideration is to avoid using convolution as a neural network strategy because a convolution learns from spatial data; however, the continuous distributed representation has already destroyed the structured data from protein sequence. Thus future work should consider utilizing continuous distributed representation, but with a neural network without convolutions. Or another consideration is utilizing continuous distributed representation with LSTMs, and with more computational power to train with more iterations.

From the data collected, it can be concluded that at this time, more future work should be focused not in sequential models, but rather one models that consider spatial features of the sequence. We can consider instead replacing the SPINE model utilized by previous works, and try to build deep learning models to predict secondary or even tertiary structures that can then be utilized further for prediction.

Once a section of a protein sequence is classified, future work can then revolve around learning models to create antibodies that can then recognize recently isolated antigens. Current work only utilize classification of B-Cell epitopes in order to help scientists create vaccines; however, with more data and better models, hopefully future work can be made in automatically using proteomics to make more effective vaccines.

College of Liberal Arts & Sciences
ARIZONA STATE UNIVERSITY