ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Shanghai, 19-21 June, 2020, pp. 13-18
doi: 10.18178/wcse.2020.06.003

Deep learning for Protein-Protein Interactions Predication

Shengyu Lu [1], Beizhan Wang [1 +] and Hongji Wang [1]

[1]School of Informatics, Xiamen University, Xiamen City, Fujian, China

Abstract. As the main component, proteins play an important role in the cell activities of organisms. Most organisms carry out the cell activities by protein-protein interactions (PPI), so deep researches about PPI is of great significance. Some machine learning methods have been applied to predict PPI by extracting features from massive protein data and then training the models to implement classification. However, these methods can only be used to deal with balanced datasets and their effects can be improved further. We proposed a deep learning method based on Bi-LSTM model to predict PPI. For protein sequences, our method automatically encoded the amino acids and represented the protein sequences, and then extracted the sequence features and implemented classification. Experimental results showed that our method can achieve higher accuracy than advanced methods, and can solve the problem of unbalanced datasets.

Keywords: protein-protein interactions (PPI), protein sequences, Bi-LSTM

1. Introduction

Since the generation of high-throughput technology, scientists can obtain huge amounts of biological data for statistics and analysis [1]. Calculation methods can overcome the defects of traditional biological experimental methods [2] such as high costs and time consumption. In recent years, some scholars have proposed many innovative methods for protein structure, protein-protein interactions (PPI) [3] and protein-protein interaction sites (PPIS) prediction [4]. These methods combined biological theoretical knowledge and mathematical models to train models [5] and achieve predication, and this promoted the development of protein research.

Bock et al. proposed an algorithm to predict PPI based on protein sequences [6]. They extracted the semantic features from protein sequences, combining the physical features of amino acids in proteins such as charge properties, water solubility, and then adopted the support vector machine (SVM) to implement classification. You et al. used multi-scale continuous and discontinuous local feature descriptors to encode amino acid sequences [7]. They assumed that consecutive amino acid fragments with different fragment lengths and used them to predict PPI. To select the best features, they used the minimum redundancy and maximum correlation criterion. This criterion can reduce the dimensions and computational complexity. Finally, they used the SVM classifier to predict PPI. Du et al. used the amphiphilic pseudo amino acid composition (APAAC) to extract features from the protein sequences, and then inputted the features of two proteins into two independent deep neural networks (DNN) to predict PPI [8]. Li et al proposed an algorithm to predict PPI based on the properties of the PPI network. They constructed a PPI network and used the improved network partition algorithm to split the network into sub networks, and then adopted the scoring function to predict PPI [9]. Li et al also proposed using the artificial neural network (ANN) paradigm to classify the protein structures. They divided a 3D protein structure into serval parts and then extracted statistical features from these parts to implement classification [10]. Ivanoska et al proposed a semantic clustering method to predict protein functions based on semantic similarity metrics and the whole network topology. They applied k-medoids clustering combined with several semantic similarity metrics as weights

+ Corresponding author. Tel.: + 18059204016
E-mail address: wangbz@xmu.edu.cn.

in the distance-clustering matrix [11]. Although these algorithms have made some progress in PPI predication, they only solved some certain problems and still had their own defects [12].

Currently, deep learning methods with powerful feature extraction capabilities have been applied to many fields such as computer vision [13] and natural language processing (NLP) [14]. They can extract depth features and latent features of some objects with deep neural networks and implement great effects[15]. Our method based on deep learning, we took full advantages of Bi-LSTM model [16] that considered the impacts of bidirectional time series of sequences and used it to extract the features of protein sequences and implement PPI. Compared with other advanced methods, our method can achieve greater effects [17].

2. Method

Based on protein sequences, we firstly automatically encode the amino acids and represent them as vectors, and train the model and update the vectors in the embedding layer. Then we extract the features of two proteins and combine them into a vector in the final layer, and finally we achieve predication by classification.

2.1. Protein Sequences Representation

We define a protein sequence as $P = \{a_1, a_2, \dots a_n\}$, where n represents the length of the protein sequence, and $a_i \in P$ represents the amino acid i of the protein. We define that there are m kinds of amino acids in the protein, so a_i has m kinds of different representations. Sequence representation is the basis for protein data analysis and PPI prediction. In traditional PPI prediction methods, the protein sequence is represented by a matrix, such as the position specificity scoring matrix (PSSM). It uses the statistical probability method to analyse the massive protein sequences. For each amino acid of a protein sequence, it calculates the mutation probability of other amino acids. Therefore, each amino acid can be represented as a $1 * (m-1)$ vector, and the protein sequence is represented as a matrix $n * (m-1)$. However, this method is complicated to calculate, and cannot represent the relation between amino acids. We refer to the embedding representation of words in NLP [18]. It divides the sequence into individual words, and uniquely encodes each word. The protein sequence is inputted into the neural network and each word which represents the amino acid in the embedding layer is initialized with a random vector. We train the model and optimize the loss function [19], and obtain the vector representation of each amino acid.

2.2. Bi-LSTM

Given two protein sequences $P_i = \{a_i^1, a_i^2 \dots a_i^n\}$, $P_j = \{a_j^1, a_j^2, \dots a_j^m\}$, n and m represent the length of the sequences, and we divide them into several words by amino acid and input them to Bi-LSTM model [20]. The Bi-LSTM model based on the traditional LSTM combines the forward LSTM and the backward LSTM. The traditional LSTM only can encode the backward sequence from the previous sequence while cannot encode the previous sequence from the backward sequence. However, the Bi-LSTM model can simultaneously capture the two-way contextual dependencies and describe the relationship between the amino acids of a protein. We can see the Bi-LSTM as Fig.1:

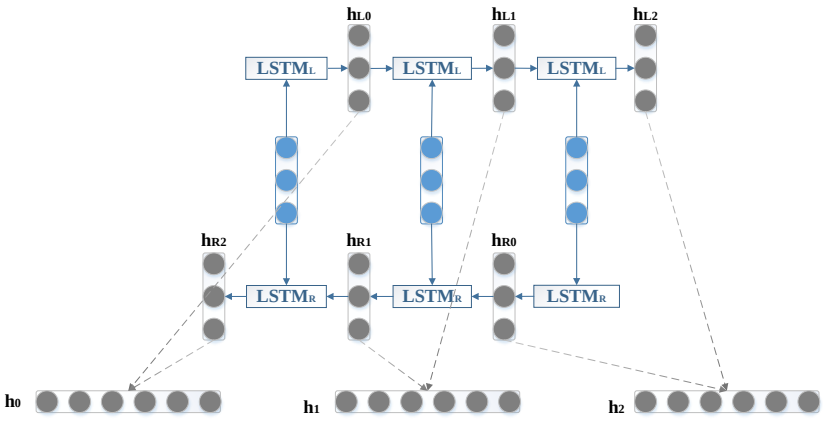

Fig. 1: The framework of Bi-LSTM

We let P_i as an example, we input the protein sequence $\{a_i^1, a_i^2 \ldots a_i^n\}$ into the forward model sequentially to obtain the forward latent vector $\{H_{L_i}^1, H_{L_i}^2 H_{L_i}^3, \ldots H_{L_i}^n\}$, and then input the sequence $\{a_n \ldots a_3, a_2, a_1\}$ into the backward model, and get the backward latent vector $\{H_{R_i}^1, H_{R_i}^2 H_{R_i}^3, \ldots H_{R_i}^n\}$. Then, we combine the latent vectors obtained by the forward model and the backward model and obtain the feature vector $\{H_i^1, H_i^2, H_i^3, \ldots H_i^n\}$. The calculation of the latent vectors is implemented by the hidden layer function *H*. The equations of *H* are as following:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$

$$h_t = O_t tanh(c_t) \quad (5)$$

Where σ represents the activation function, and x_t represents the inputted word. The *i, f, o* represent the input gate, forget gate, and output gate respectively, and *c* is the input activation function of memory cell. W_{ht} and W_{xo} represent the input gate matrix and the out gate matrix of the hidden layer respectively. They are both diagonal matrix. b_i is the bias.

In the final layer of the Bi-LSTM network, we use the softmax function to implement classification and calculate the probability of a category. The equation is as following:

$$p(C_k|x) = y_k = \frac{e^k}{\sum_{d=1}^{K} e^d} \quad (6)$$

Where, $p(C_k|x)$ represents the probability that the predicted category is *d* when input the *x*, and *K* is the number of categories. C_k represents the category. We minimize the loss function through training the model, the equation is as：

$$\text{Loss} = -\frac{\sum_{i=1}^{n} \ln \text{p}(z|x_i)}{n} = -\frac{\sum_{i=1}^{n}\sum_{d=1}^{K} z_k ln y_k}{n} \quad (7)$$

Where, x_i is a vector that represents an amino acid in the protein sequence. z_k and $y_k$represent the true category and predicted category respectively.

3. Experiment

In this section, we executed experiments and used three evaluation metrics to test the performance of our method, compared with several baseline methods.

3.1. Dataset

We used the public protein dataset to perform our experiments [21]. The dataset was from human protein references database (HPRD, 2007 version). It provided the positive samples that two protein sequences can interact. For the negative samples, we obtained them by pairing proteins at different subcellular locations and referenced to the Swiss-Prot database version 57.3.

3.2. Evaluation Metrics

We used the precision, recall and F1 to evaluate the performance of our method [22], compared with other methods. The equations are as following:

$$\text{precision} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{recall} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{F1} = \frac{2*precision*recall}{precision+recall} \quad (10)$$

Where, TP, FP and FN represent the true positive, false positive and false negative respectively.

3.3. Baseline Methods

We listed the baseline methods as following:

CNN: Convolutional neural network (CNN) has several layers and a great number of parameters, and it can extract the features of sequences by convolution and pooling operation [23]. We designed three convolutional layers in CNN and filter size was $3 * 3$. We used the Embedding method to represent the protein sequences, and then inputted them into the CNN to extract features and implement classification.

SVM: SVM algorithm extracts the features of sequences and uses a hyperplane to divide two different boundaries to implement classification [5].

Random forest: Random forest is an ensemble learning algorithm. It consists of multiple decision trees and obtains the final result through a voting mechanism from all decision trees [12].

3.4. Experimental Setting

We performed experiments using python and tensorflow 2.0. There were 32 units in the Bi-LSTM model, and the size of batch was set to be 32. For the embedding of amino acids, we set the dimensions to be 128. The number of epochs was 15.

3.5. Result Analysis

We selected 20000 samples randomly and set the same number of positive samples and negative samples to test the performance of our model.

Table.1: The evaluation metrics of all method

Method	Precision	Recall	F1
Bi-LSTM	**0.9782**	**0.9511**	**0.9644**
CNN	0.9424	0.9032	0.9135
SVM	0.6726	0.6965	0.6843
RF	0.9475	0.8452	0.8934

From Tab.1, we can see that the evaluation metrics of all methods in our dataset. Compared with other baseline methods, our model considered the bidirectional time series of protein sequences and obtained more protein information. The precision, recall, and F1 of our method were higher than those of other methods. These results showed that our method performed better than the baselines methods. When deal with protein datasets and other biological datasets, we often have to address the problem of unbalanced distribution. Therefore, we changed the positive samples ratio in our dataset to evaluate our method performance in unbalanced datasets.

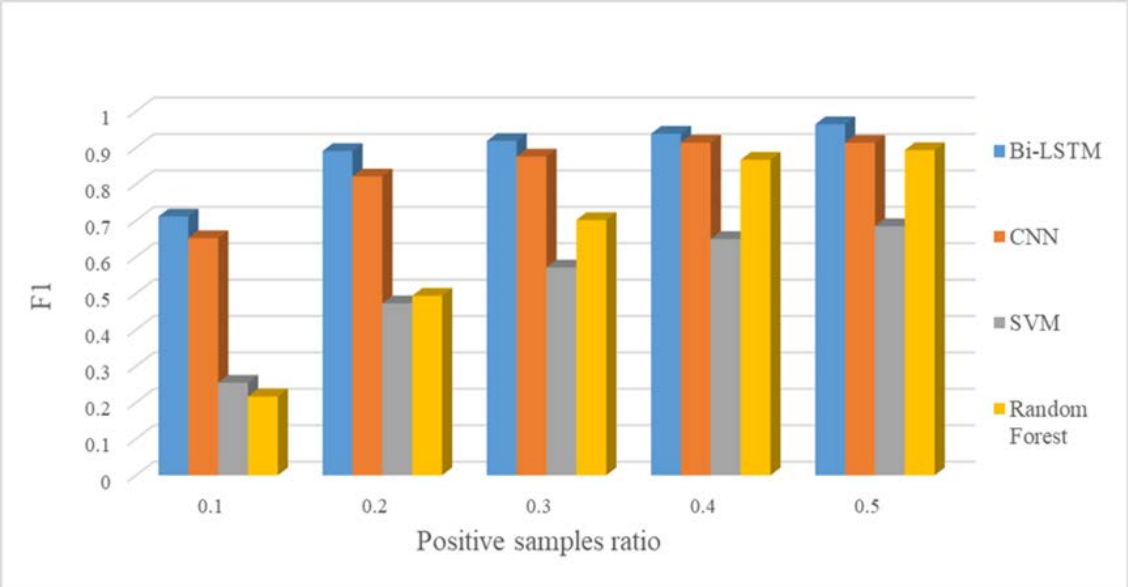

Fig. 2: The F1 in the dataset with different positive samples ratio

As shown in the Fig.2, in the case of unbalanced distribution of positive and negative samples, the effects of our model were always better than other baseline methods. Therefore, our method can address the problem of unbalanced protein datasets and provide an available reference for other biological datasets such as drug datasets.

4. Conclusion

In this paper, we proposed an effective method for PPI predication. Our method based on protein sequences, adopted the embedding method to represent the amino acids of proteins. We considered the time series of protein sequences and used the Bi-LSTM model to extract features and implement classification to predict PPI. Compared with advanced machine learning methods, our method can perform better. Besides, when the datasets were unbalanced distribution, our method can also remain high accuracy than other methods, so our method can help address the problem of unbalanced datasets in other fields.

5. Acknowledgements

The work was supported by the Natural Science Foundation of China (No. 61502402) and the fundamental research funds for the central universities (No. 207220180073).

6. References

[1] T. Sun, B. Zhou, L. Lai, and J. Pei, ‘Sequence-based prediction of protein protein interaction using a deep-learning algorithm’, BMC Bioinformatics, vol. 18, no. 1, Dec. 2017, doi: 10.1186/s12859-017-1700-2.

[2] Z.-H. You, X. Li, and K. C. Chan, ‘An improved sequence-based prediction protocol for protein-protein interactions using amino acids substitution matrix and rotation forest ensemble classifiers’, Neurocomputing, vol. 228, pp. 277–282, Mar. 2017, doi: 10.1016/j.neucom.2016.10.042.

[3] S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, ‘Predicting protein–protein interactions through sequence-based deep learning’, Bioinformatics, vol. 34, no. 17, pp. i802–i810, Sep. 2018, doi: 10.1093/bioinformatics/bty573.

[4] X. Du, S. Sun, C. Hu, X. Li, and J. Xia, ‘Prediction of protein–protein interaction sites by means of ensemble learning and weighted feature descriptor’, Journal of Biological Research-Thessaloniki, vol. 23, no. S1, May 2016, doi: 10.1186/s40709-016-0046-7.

[5] S. Lu, B. Wang, H. Wang, and Q. Hong, ‘A Hybrid Collaborative Filtering Algorithm Based on KNN and Gradient Boosting’, in 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, 2018, pp. 1–5, doi: 10.1109/ICCSE.2018.8468751.

[6] Y. E. Göktepe and H. Kodaz, ‘Prediction of Protein-Protein Interactions Using An Effective Sequence Based Combined Method’, Neurocomputing, vol. 303, pp. 68–74, Aug. 2018, doi: 10.1016/j.neucom.2018.03.062.

[7] T.-H. Kuo and K.-B. Li, ‘Predicting Protein–Protein Interaction Sites Using Sequence Descriptors and Site Propensity of Neighboring Amino Acids’, International Journal of Molecular Sciences, vol. 17, no. 11, p. 1788, Oct. 2016, doi: 10.3390/ijms17111788.

[8] L. Zhang, G. Yu, D. Xia, and J. Wang, ‘Protein–protein interactions prediction based on ensemble deep neural networks’, Neurocomputing, vol. 324, pp. 10–19, Jan. 2019, doi: 10.1016/j.neucom.2018.02.097.

[9] H. Li, C. Liu, and L. Burge, ‘Predicting Protein-Protein Interactions Based on PPI Networks’, IJMLC, pp. 794–797, 2012, doi: 10.7763/IJMLC.2012.V2.239.

[10] H. Li, C. Liu, L. Burge, and W. Southerland, ‘Classification of Protein 3D Structures Using Artificial Neural Network’, IJMLC, pp. 791–793, 2012, doi: 10.7763/IJMLC.2012.V2.238.

[11] I. Ivanoska, K. Trivodaliev, and S. Kalajdziski, ‘Protein Function Prediction Using Semantic Driven K-Medoids Clustering Algorithm’, IJMLC, pp. 52–56, Feb. 2014, doi: 10.7763/IJMLC.2014.V4.385.

[12] S. Lu, H. Chen, X. Zhou, B. Wang, H. Wang, and Q. Hong, ‘Graph-Based Collaborative Filtering with MLP’, Mathematical Problems in Engineering, vol. 2018, pp. 1–10, Dec. 2018, doi: 10.1155/2018/8314105.

[13] S. Lu, B. Wang, H. Wang, L. Chen, M. Linjian, and X. Zhang, ‘A real-time object detection algorithm for video’, Computers & Electrical Engineering, vol. 77, pp. 398–408, Jul. 2019, doi: 10.1016/j.compeleceng.2019.05.009.

[14] A. Bouaziz, C. Dartigues-Pallez, C. da Costa Pereira, F. Pre-cioso, P. Lloret, "Short text classification using semantic random forest", Proceedings of DaWaK'14, pp. 288-299, 2014.

[15] Z.-P. Liu and H. Miao, ‘Prediction of protein-RNA interactions using sequence and structure descriptors’, Neurocomputing, vol. 206, pp. 28–34, Sep. 2016, doi: 10.1016/j.neucom.2015.11.105.

[16] S. Yadav, A. Ekbal, S. Saha, A. Kumar, and P. Bhattacharyya, ‘Feature assisted stacked attentive shortest dependency path based Bi-LSTM model for protein–protein interaction’, Knowledge-Based Systems, vol. 166, pp. 18–29, Feb. 2019, doi: 10.1016/j.knosys.2018.11.020.

[17] S. Lu, ‘Deep learning for object detection in video’, Journal of Physics: Conference Series, vol. 1176, p. 042080, Mar. 2019, doi: 10.1088/1742-6596/1176/4/042080.

[18] H. A. Burkhardt, D. Subramanian, J. Mower, and T. Cohen, ‘Predicting Adverse Drug-Drug Interactions with Neural Embedding of Semantic Predications’, Bioinformatics, preprint, Sep. 2019.

[19] S. Lu and B. Wang, ‘An image retrieval algorithm based on improved color histogram’, Journal of Physics: Conference Series, vol. 1176, p. 022039, Mar. 2019, doi: 10.1088/1742-6596/1176/2/022039.

[20] B. Zhang, J. Li, L. Quan, Y. Chen, and Q. Lü, ‘Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network’, Neurocomputing, vol. 357, pp. 86–100, Sep. 2019, doi: 10.1016/j.neucom.2019.05.013.

[21] W. You, Z. Yang, G. Guo, X.-F. Wan, and G. Ji, ‘Prediction of DNA-binding proteins by interaction fusion feature representation and selective ensemble’, Knowledge-Based Systems, vol. 163, pp. 598–610, Jan. 2019, doi: 10.1016/j.knosys.2018.09.023.

[22] S. Lu, ‘An Image Retrieval Learning Platform with Authentication System’, in 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, 2018, pp. 1–5, doi: 10.1109/ICCSE.2018.8468711.

[23] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, ‘Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling’, arXiv:1611.06639 [cs], Nov. 2016.