CS181 Artifitial Intelligence Project: Interaction And Structure of Aptamers Prediction Based On Neural Network

FANG Zhiyu, WANG Xiyuan, LI Boyi, ZHAO Jianhao, CHEN Siyun
2018520146 2018533177 2018533187 2018533157 2018533243
fangzhiy1@ wangxy7@ liby@ zhaojh@ chensu@

Abstract

Aptamers are short single-strand DNA sequences that can bind to their specific targets with high affinity and specificity. Generally, the computational prediction approaches of aptamer have been proposed to carry out in two main categories: interaction-based prediction and structure-based predictions. However in this report, we use neural network to predict whether it will bind to the target. Using these computational methods and tools, biologists might take advantage of these computational techniques to design more accurate and more sensitive aptamers.

1. Introduction

Aptamers were first introduced in the last decade of the twentieth century. An aptamer interacts with its target through different intermolecular interactions such as Van der Waal's forces, electrostatic interactions, 3D shape, stacking, and hydrogen bonds. Moreover, aptamers can fold into a variety of secondary and tertiary structure which be able to recognize multiple target binding sites. Aptamers are analogous to antibodies for a wide range of targets and applications (Chandola et al., 2016). However, they have some major advantages over antibodies (Lakhin et al., 2013). The most prominent advantage of aptamers in comparison with protein antibodies is their stability at high temperatures.

Recently utilization of computational methods in aptamer prediction approaches drawn much attention. the computational methods are simple, time and cost-effective, and does not require sophisticated instruments (Ahirwar et al., 2016b). Several compu- tational methods are applied in aptamer approaches e.g. designing efficient aptamers (Hamada, 2018), simulating aptamer selection (Spill et al., 2016), in-silico performing SELEX (Wondergem et al., 2017), interaction prediction (Li et al., 2014), and structure predic- tion (Boniecki et al., 2016). All these methods require the utiliza- tion of aptamer databases (DB) (Lee et al., 2004a). What's more, machine learning methods have been widely used in the prediction of protein-aptamer interactions, some computational models have been developed, for example, Li et al. developed a random forest-based protein-aptamer interaction prediction model, Zhang et al. presented a novel model based on the ensemble method in 2016. This report is an implement of predicting using neural network.

2. Background theorey

2.1. dataset

The protein-aptamer pair required for the experiment comes from PPAI[1], which is a web server for predicting protein-aptamer interactions. After processing, there are 1375 items remaining (mainly due to the category imbalance problem, many negative cases have been deleted). After processing by pse-AAC[2] and pse-KNC[3], the data are in the files feature_balance.npy and label_balance.npy.

2.2. feature extraction

For a pair of protein sequence and aptamer sequence, it is obviously not a good choice to directly use their sequence as a feature, and it will lose many of the properties of the protein and the aptamer itself. Therefore, in this model, pse-AAC and pse-KNC, which are widely used in bioinformatics, are used to convert protein sequences and aptamer sequences into 70 and 20-dimensional vectors according to the physical and chemical properties of amino acids and nucleotides, respectively. The vectors are spliced together to generate a 90-dimensional vector as a fixed-length input, the protein-aptamer pair is marked as combinable or non-combinable. The classes are coded with one-hot, where the positive one is marked as [1, 0], the negative one is marked as [0,1].

When processing protein by pse-AAC, we selected five physical and chemical properties of amino acid polarity,

Sencondary structure, Molecular volume, Codon diversity, and Electrostatic charge as indicators, and the λ was set to 50 (that is, 50 connected amino acid sequences are considered at most)

When processing the aptamer sequence by pse-KNC, the open-sourced pseKNC-general are chosen, the homepage is http://lin-group.cn/server/pseknc, and the relevant programs are attached in the attachment. For both rna and dna, their six physical and chemical properties of shift, slide, rise, tilt, twist, roll are selected as the index of the generated sequence, λ is set to 4, and K is set to 2 (that is, only the properties of the two-tuple in the sequence are considered)

Then, we use Min-Max Normalization to do linear transformation.

$$x^* = \frac{x - \min}{\max - \min} \tag{1}$$

2.3. method

Artificial neural network is also referred to as neural network or connection model for short. It is an algorithmic mathematical model that imitates the behavioral characteristics of animal neural network and performs distributed parallel information processing. This kind of network relies on the complexity of the system and achieves the purpose of processing information by adjusting the interconnection between a large number of internal nodes. In engineering and academia, it is often simply referred to as "neural network" or quasi-neural network.

2.3.1 Input Layer

The input layer has two node. One is the feature of protein, another is the feature of Aptamer. They're vectors of length 70 and 20.

In our model, we have trained a neural network to use it to predict whether the protein and aptamer will bind or not. From 1347 training data, we randomly split it to two parts: one is the training set, which has 1024 example, another is validation set, which has 323 example.

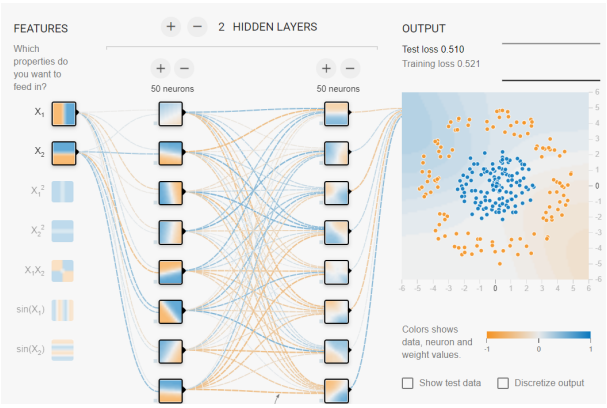


Figure 1. Neural Network

2.3.2 Hidden Layer

We have two hidden layers. Each of them has 50 nodes. Use dot multiply, we have

$$H = X * W1 + b1 \tag{2}$$

and

$$Y = H * W2 + b2 \tag{3}$$

then we can get the output.

Through the calculation of the above two linear equations, we can get the final output Y due to the fact that a series of linear equations can eventually be expressed by a linear equation. In other words, the above two formulas can be expressed by a linear equation after being combined. This is true for two neural networks, even if the network depth is increased to 100 layers, it is still the same. In this case, the neural network loses its meaning.

So here we need to add an activation layer to the network.

2.3.3 Activation Layer

We use function tanh() in tensorflow as active function. It introduces nonlinear characteristics into neural networks.

2.3.4 Regularation

The problem of overfitting usually occurs when there are too many variables (features). In this case, the trained equation always fits the training data well, that is, our cost function may be very close to 0 or just 0.

However, such a curve does everything possible to fit the training data, which will cause it to be unable to generalize the new data sample, so that the price of the new sample cannot be predicted. Here, the term "generalization" refers to the ability of a hypothetical model to be applied to new samples. New sample data refers to data that does not appear in the training set.

The L2 regularation makes the parameter smaller, which makes the model simpler.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right] \tag{4}$$

2.3.5 Backpropagation and parameter optimization

We simply use the gradient descent to optimize the parameter in network. The theroem are below:

Gradient descent:

$$W_{ij} = W_{ij} - \alpha\frac{\partial L(w,b)}{\partial W_{ij}} \quad b_i = b_i - \alpha\frac{\partial L(w,b)}{\partial b_i} \tag{5}$$

Backpropagation use Chain rule to calculate loss function difference to every parameter.

$$v_i^l = \sum_{j=0}^{n} w_{ij}^l y_j^{l-1} + b^l \tag{6}$$

v_i^l represents the value of i^{th} node in l^{th} layer.

Set residual as:

$$\delta_i^l = \frac{\partial L(w,b)}{\partial v_i^l} \tag{7}$$

$$w_{ij}^l = w_{ij}^l - \alpha \frac{\partial L(w,b)}{\partial w_{ij}^l} \tag{8}$$

$$= w_{ij}^l - \alpha \frac{\partial L(w,b)}{\partial v_i^l} \times \frac{\partial v_i^l}{\partial w_{ij}^l} \tag{9}$$

$$= \delta_i^l \times \frac{\partial}{\partial w_{ij}^l} \left[\sum_{j=0}^{n^{l-1}} w_{ij}^l a_j^{l-1} + b_i^l \right] \tag{10}$$

$$= \delta_i^l \times a_j^{l-1} \tag{11}$$

$$b_i^l = b_i^l - \alpha \frac{\partial L(w,b)}{\partial b_i^l} \tag{12}$$

$$= b_i^l - \alpha \frac{\partial L(w,b)}{\partial v_i^l} \times \frac{\partial v_i^l}{\partial b_i^l} \tag{13}$$

$$= \delta_i^l \times \frac{\partial}{\partial b_i^l} \left[\sum_{j=0}^{n^{l-1}} w_{ij}^l a_j^{l-1} + b_i^l \right] \tag{14}$$

$$= \delta_i^l \tag{15}$$

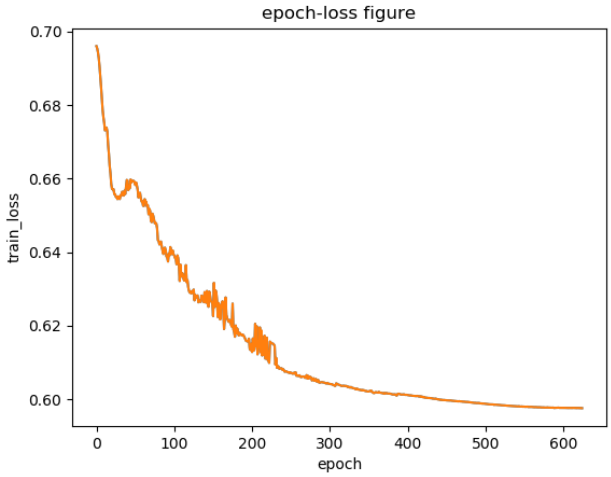
$a_j^{(l-1)}$ represents the active value at j^{th} value in $(l-1)^{th}$ layer.

$$\delta_i^{K-1} == \sum^{n^K} \left[\delta_j^K \times w_{ji}^{K-1} \right] \times f'_{K-1} \left(v_i^{K-1} \right) \tag{16}$$

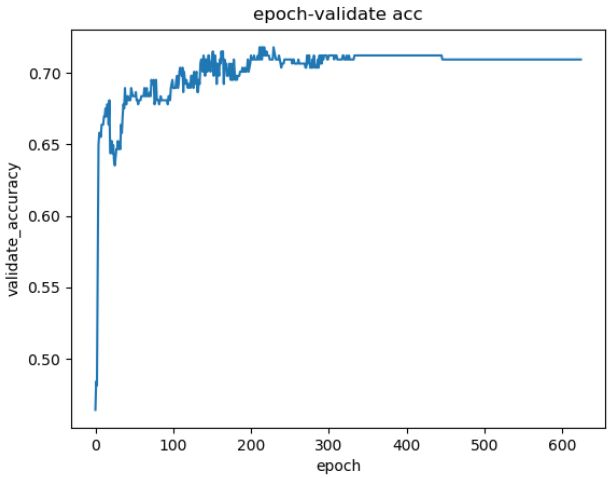
We can calculate the residual in every layer. Using gradient descent, we can get the updated W and b.

3. Result Display

After 1000 epochs, we have finished the network training.



The epoch-loss figure clearly depicts that as the number of epoch increases, the training loss decreases by 10.0% generally and finally converges at 60.0%.



The epoch-validate accuracy figure shows that as the number of epoch increases, the validate accuracy has a rapid growth first,which increases to 67.0%, then it fluctuates slightly and finally converges at 72.0%.

4. Thought and Conclusion

In this study, we developed a neural network for predicting and analyzing whether an aptamer will bind to its target.With the 1024 optimal aptamer-target pairs selected, our approach achieved an overall accuracy of 72.0% on an independent dataset. It's delightful to see that we have reached a high accuracy to some extent, however, there's still a lot to improve. First, although the substitution of activation function from ReLu to tanh slightly enhances the prediction ability (from 70.0% to 72.0% on validation set), the loss fluctuates even when the accuray converges. The possible reason for this fluctuation may due to lack of shuffling of

training samples or the nature of tanh. Next, the dataset is too small to train more accurately, which mainly due to the imbalance of category of the original one. Furthermore, using a more complex network to train maybe is a better choice, such as Deep Neural Network and Convolutional Neural Network... Additionaly, we can not only extract five physical and chemical properties, but take other features into consideration as well to get a more accurate result. In conclusion, using the computational methods and tools,we hope biologists might take ad-vantage of these computational techniques to design more accurate and more sensitive aptamers.

5. External material

Pse-AAC: Extract feature from protein.
Pse-KNC: Extract feature from aptamer.
PPAI dataset: Dataset we use.
Python library:
numpy: Load the feature and label.
Tensorflow: Construction of neural network, training.
Pandas: Save the output to csv.
os: Save model.

References

[1] https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03574-7.

[2] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3899287/#pone.0086729.s001

[3] https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1087-5

[4] Li, J., Ma, X., Li, X. et al. PPAI: a web server for predicting protein-aptamer interactions. BMC Bioinformatics 21, 236 (2020). https://doi.org/10.1186/s12859-020-03574-7

[5] Neda Emami, Parvin Samadi Pakchin, Reza Ferdousi, Computational predictive approaches for interaction and structure of aptamers, Journal of Theoretical Biology, Volume 497, 2020, 110268, ISSN 0022-5193,