Intelligent Information Management, 2020, 12, 36-42
https://www.scirp.org/journal/iim
ISSN Online: 2160-5920
ISSN Print: 2160-5912

An Improved Deep Learning Model for Predicting DNA Sequence Function

Dongfeng Li[1], Xiao Huang[2*]

[1]Shandong Experimental High School, Jinan, China
[2]Daqing Experimental High School, Daqing, China
Email: *1335671683@qq.com

How to cite this paper: Li, D.F. and Huang, X. (2020) An Improved Deep Learning Model for Predicting DNA Sequence Function. *Intelligent Information Management*, **12**, 36-42.
https://doi.org/10.4236/iim.2020.121003

Received: October 21, 2019
Accepted: December 31, 2019
Published: January 2, 2020

Abstract

Since a complete DNA chain contains a large data (usually billions of nucleotides), it's challenging to figure out the function of each sequence segment. Several powerful predictive models for the function of DNA sequence, including, CNN (convolutional neural network), RNN (recurrent neural network), and LSTM [1] (long short-term memory) have been proposed. However, all of them have some flaws. For example, the RNN can hardly have long-term memory. Here, we build on one of these models, DanQ, which uses CNN and LSTM together. We extend DanQ by developing an improved DanQ model and applying it to predict the function of DNA sequence more efficiently. In the most primitive DanQ model, the regulatory grammar is learned by the regulatory motifs captured by the convolution layer and the long-term dependencies between the motifs captured by the recurrent layer, so as to increase the prediction accuracy. Through the testing of some models, DanQ has greatly improved in some indicators. For the regulatory markers, DanQ achieves improvements above 50% of the area under the curve, via the measurement of the precision-recall curve.

Keywords

BLSTM, Convolutional Neural Network, DanQ Model, Random Dropout

1. Introduction

Previously, people raised some deep learning models to solve the prediction of DNA sequence's [2] function. They use particular deep learning algorithm to identify the large, feature-rich dataset. Convolutional neural network (CNN) is an efficient one, the variation of deep neural network (DNN) [3]. It divides into four parts: convolution layer, rectified linear units layer, pooling layer and loss

layer. Firstly, convolution kernel slides on the initial matrix and does some calculation, which will capture the vital feature of the data. In the ReLU layer, using some function (such as $f(x)=\tanh(x)$ or $f(x)=|\tanh(x)|$) can improve the training speed without altering the convolution layer. In the pooling layer, we will divide the convolution layer into several parts and the data's size will be minimized. Finally, the loss layer will punish the predicted result according to the difference between the predicted one and the real one. CNN can be used to do some researches in DNA, and the reason is that the convolution filters can arrest the sequence motifs which are short patterns that recur in DNA. From this, we can also deem that it has biological function.

Bi-directional long short-term memory (BLSTM) [2] is another variation of deep neural network. It contains two chains of long short-term memory (LSTM). LSTM is improved from recurrent neural network (RNN). This model consists of three gates: input gate, output gate and forget gate. Usually, data enters the input gate and the forget gate throws out some of noisy parts. That means forget gate can decide whether the data should be outputted and hold important information, so it's able to recollect a value for an arbitrary span of time. But it has not been used in prediction of DNA sequence function.

We use random dropout rate, which can make the trained neural network structure more flexible and converge the network quickly while ensuring the training accuracy. The iteration of neural network ensures the record of fast network with high upper limit, which greatly accelerates the training speed while keeping the accuracy unchanged.

Therefore, combining with the two models above, we chose DanQ. The works are elaborating the two main elements: DanQ model and random dropout, training the original DanQ and the improved one that the random dropout is added to, and analyzing the results, including its accuracy and speed. After that, we will evaluate the feasibility of the improvement, finding the optimal range of data's size, and finally get the conclusion.

In our research, we try to realize the following aims:

1) Be familiar with CNN and BLSTM, which were proposed earlier.

2) Try to combine the two models together to form "DanQ".

3) Train this new model proficiently and apply it to predict the function of DNA sequence.

4) Try to modify the DanQ model to make it more efficient.

DanQ

It's a hybrid framework combined with CNN and BLSTM. The first step is to convolve the inputted hot coding to simplify it and use the max pooling layer to learn it, and then input the result to the BLSTM layer, and after that, enter the last two layers that are a dense layer of rectified linear units and a multi-task sigmoid (like $f(x)=\left(1+e^{-x}\right)^{-1}$) output.

The DanQ model (see in **Figure 1**) is divided into six portions: first, the input sequence is thermally encoded into a four-row matrix. An output matrix with a

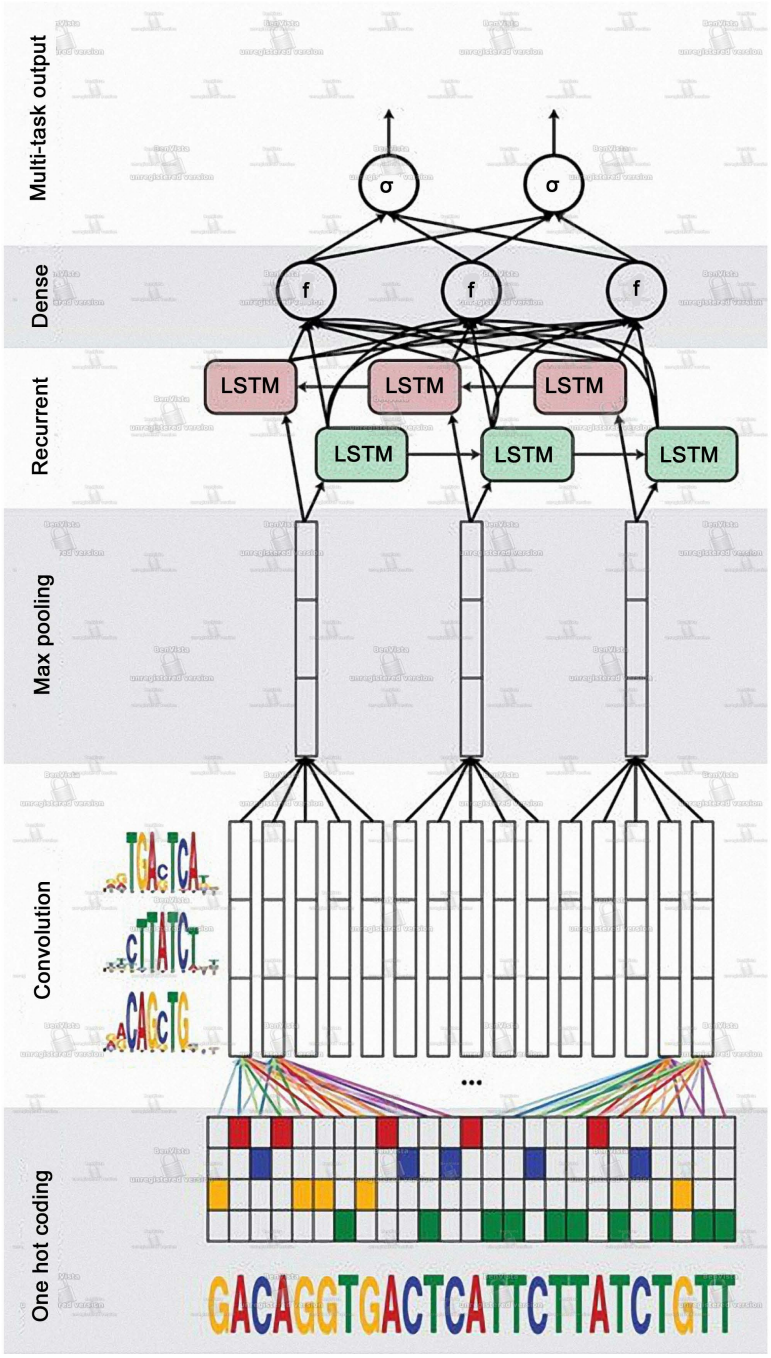

Figure 1. DanQ structure.

line for each convolution kernel and a column for each position in the input (minus the range of the kernel) was produced by a convolution layer with rectifier activation. The size of the output matrix though the dimensional axis was reduced by max pooling, retaining the count of channels. The orientations and dimensional distances between the motifs were deemed by consecutive BLSTM layer. The outputs of BLSTM were flattened into a layer as inputs to a completely connected layer of rectified linear units. A sigmoid non-linear alteration to a

vector was applied by the final layer, which is assists as probability predictions of the epigenetic marks to be contrasted via a loss occupation to the true target vector.

Since DanQ is just a combined model of ordinary neural network, it has the same problem as other deep learning models-overfitting. The large neural networks are slow to use, and overfitting makes the input information even more difficult to operate. Dropout can be added to solve this problem (see in **Figure 2**). Its mechanism is removing a neural unit randomly as well as its incoming and outcome connection. The probability of removing a unit is independent of other unit and it can be chosen artificially. Through multiple training, the optimal dropout rate could be obtained.

2. Methods

The functionality and data of DeepSEA framework also applies to DanQ. Namely，the reference genome of human grch37 was divided into 200 bp bin without overlapping. By intersecting the 919 CHIP-seq, DNase-seq peaks with the uniformly processed encode and roadmap epigenomics data releases, the targets are calculated, thus 919 binary target vectors are generated for each sample [3]. Each sample is matched with the carrier of its target to form a 1000 BP sequence overlapped on a 200 bp bin on at least one TF binding CHIP-seq peak. According to this information, we anticipated that each target vector would include at least one positive value; but, we detected that about 10% of all target vectors were negatives. Each 1000-bp DNA sequence is encoded into a 1000 × 4 binary matrix by a single bond, and its columns correspond to A, G, C and T. Training, validation and testing sets can be downloaded from DeepSEA website [4]. Samples were stratified via chromosomes into strictly non-overlapping training, validation and testing sets. The validation set was not used for training or testing. The set consists of reverse complements, effectively doubling the size of each dataset.

In order to evaluate the performance of the test set, we calculate the prediction probability of each sequence as the average of the probability prediction of the positive and negative complementary sequence pairs.

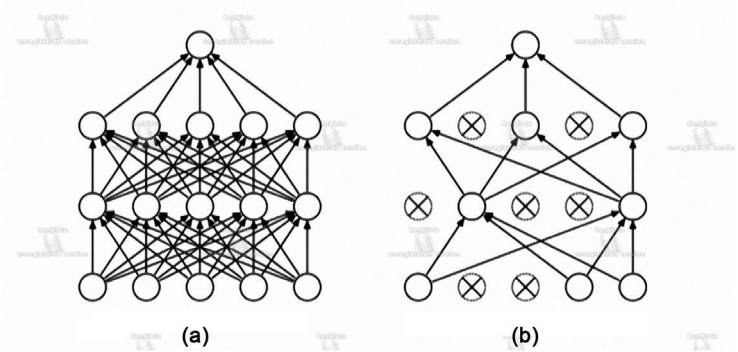

Figure 2. Dropout neural net model. (a) Standard neural net; (b) After applying dropout.

DanQ Model

For additional details on the architecture and related parameters used in this research, seeing Supplement. It includes discarding, which is used to randomly set the proportion of neuron activation in the maximum pool and BLSTM layer to 0 in each training step, so as to normalize the DanQ model. The dropout rate was set to be random so as to improve the velocity of convergence, in the altered algorithm.

We change the scale of dataset to get the training much faster. 4,400,000 data turns into 40,000. But we will also train the complete dataset after the program to avoid inaccuracy.

Improved Method

In the original DanQ, we set the dropout rate to 0.5 in LSTM (Long Short-term memory) and that of max-pooling layer is 0.2. They are changed into two random numbers between 0.1 and 0.3 of max-pooling layer and between 0.4 and 0.6 of LSTM. This change makes the neural network's structure more flexible.

The Website of Dataset

https://genome.cshlp.org/content/21/3/447/suppl/DC1.

3. Results

Through the training of original DanQ model, we find that using both convolution and recurrent, DanQ is a practical and effective model with high accuracy. But to improve it, we add random dropout in this model and compare it with the original one to figure out if this modification can really improve DanQ model. For the training of the neural network that has a larger batch size, the random dropout rate has a tremendous improvement on the training speed. To be specific, the epoch of the random dropout training is 20 in average of 14 times. By the contrast, the original DanQ method has average 27 epochs in average of 6 times with less than 0.3% loss smaller than that of random dropout. Based on these training facts, we can get the preliminary conclusion that random dropout rate improves the DanQ's training speed at large batch size.

4. Discussion

Meanwhile, the random dropout and original DanQ has little difference when applying to a smaller batch size. Both of them have an average 12 epochs with similar accuracy and loss. However, compared to larger batch size, each epoch that with smaller batch costs three times more than that of larger batch size to get similar loss and accuracy. Hence, the total training time is still larger. Maybe this size is not in suitable range of random dropout (see in **Figure 3** and **Figure 4**).

5. Conclusions

We have trained the DanQ model through the code implementation and proved that it's practical. Moreover, we use random dropout training and compare its

		epoch	acc	loss
Random	A	21	0.97913	0.08136
	B	18	0.97864	0.08439
	C	17	0.97958	0.08386
	D	13	0.97951	0.09647
	E	19	0.97978	0.08145
	F	32	0.98106	0.08013
	G	38	0.98202	0.07613
	H	19	0.97815	0.08013
	I	23	0.97803	0.07011
	J	11	0.97525	0.08101
	K	16	0.9796	0.08654
	L	20	0.97849	0.07462
	M	13	0.97431	0.08646
	N	21	0.97841	0.07202
average		20.07142857	0.978711	0.081049
DanQ	A	23	0.97869	0.08489
	B	28	0.97865	0.07914
	C	29	0.98035	0.07544
	D	32	0.97987	0.07914
	E	25	0.97972	0.08414
	F	27	0.97709	0.07343
average		27.33333333	0.979062	0.079363

Figure 3. The random dropout training and original DanQ.

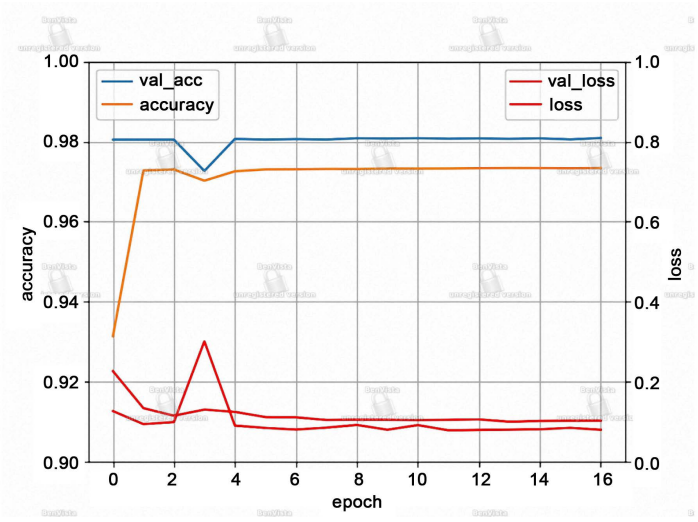

Figure 4. Two methods' accuracy at different epoch.

result with original DanQ's, finding that it can improve the DanQ's training speed. That means it's possible to make DanQ model analyze the DNA sequence more effectively. However, although the modified model is practical when the batch's size is small, it costs more when the batch's size is large. So we still need to find the most suitable range of random dropout to make it feasible. In addition, we will try to look for other better methods to modify DanQ model.

6. Future Goal

1) Although the average training time of random dropout algorithm is smaller than the original one, the range of it is still unpredictable. The random dropout that is trained longest has 38 epochs, which is much larger than the normal one. We are trying to improve this algorithm to make it more stable.

2) To enlarge the data to all 4,400,000.

3) To have some "tricks" on the network. For example, if one part of regions is too hard to the network, it can restudy it for more times than others.

4) To find the best range of the dropout.

Acknowledgements

The author wish to thank Professor Manolis Kellis in Massachusetts Institute of Technology, TA Ying Zhang in Chongqing and TA Zihao Zhang in Tongji University for providing dataset in some previous articles and guidance of code implementation and article's edition.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Graves, A. and Schmidhuber, J. (2005) Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, **18**, 602-610. https://doi.org/10.1016/j.neunet.2005.06.042

[2] Alipanahi, B., *et al.* (2015) Predicting the Sequence Specificities of DNA- and RNA-Binding Proteins by Deep Learning. *Nature Biotechnology*, **33**, 831-838. https://doi.org/10.1038/nbt.3300

[3] Quang, D., Chen, Y. and Xie, X. (2015) DANN: A Deep Learning Approach for Annotating the Pathogenicity of Genetic Variants. *Bioinformatics*, **31**, 761-763. https://doi.org/10.1093/bioinformatics/btu703

[4] Zhou, J. and Troyanskaya, O.G. (2015) Predicting Effects of Noncoding Variants with Deep Learning-Based Sequence Model. *Nature Methods*, **12**, 931-934. https://doi.org/10.1038/nmeth.3547