J. Biomedical Science and Engineering, 2016, 9, 280-286
Published Online April 2016 in SciRes. http://www.scirp.org/journal/jbise
http://dx.doi.org/10.4236/jbise.2016.95021

DNA Sequence Classification by Convolutional Neural Network

Ngoc Giang Nguyen[1], Vu Anh Tran[1], Duc Luu Ngo[1], Dau Phan[1], Favorisen Rosyking Lumbanraja[1], Mohammad Reza Faisal[1], Bahriddin Abapihi[1], Mamoru Kubo[2], Kenji Satou[2]

[1]Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Japan
[2]Institute of Science and Engineering, Kanazawa University, Kanazawa, Japan
Email: giangnn.bkace@gmail.com, tvatva2002@gmail.com, ndluu@blu.edu.vn, pdaukg@gmai.com, favorisen@gmail.com, reza.faisal@gmail.com, bahriddinabapihi@yahoo.com, mkubo@t.kanazawa-u.ac.jp, ken@t.kanazawa-u.ac.jp

Received 25 February 2016; accepted 24 April 2016; published 27 April 2016

Abstract

In recent years, a deep learning model called convolutional neural network with an ability of extracting features of high-level abstraction from minimum preprocessing data has been widely used. In this research, we proposed a new approach in classifying DNA sequences using the convolutional neural network while considering these sequences as text data. We used one-hot vectors to represent sequences as input to the model; therefore, it conserves the essential position information of each nucleotide in sequences. Using 12 DNA sequence datasets, we evaluated our proposed model and achieved significant improvements in all of these datasets. This result has shown a potential of using convolutional neural network for DNA sequence to solve other sequence problems in bioinformatics.

Keywords

DNA Sequence Classification, Deep Learning, Convolutional Neural Network

1. Introduction

In life science, study of DNA is an important factor of understanding about organisms. DNA carries most of the genetic instructions of development, functioning, and reproduction in all organisms. Nowadays, with development of sequencing technologies, we can easily read a DNA sequence. According to data from NHGRI Genome Sequencing Program, cost of reading one million base pairs rapidly reduced from more than $5000 in September

How to cite this paper: Nguyen, N.G., Tran, V.A., Ngo, D.L., Phan, D., Lumbanraja, F.R., Faisal, M.R., Abapihi, B., Kubo, M. and Satou, K. (2016) DNA Sequence Classification by Convolutional Neural Network. *J. Biomedical Science and Engineering*, **9**, 280-286. http://dx.doi.org/10.4236/jbise.2016.95021

2001 to only $0.014 in October 2015. The amount of data about DNA sequences is also exponentially increasing. For example, the size of GenBank, a popular database of DNA sequences, has grown up to more than 2 billion base pairs in December 2015. It is excellent if we could use these huge data with the power of modern computer to help us understand about DNA.

By using thoroughly understood sequence to train machine learning models, we could use the trained models to predict profile of unknown sequences. In recent years, a new branch of machine learning models called deep learning was introduced. It is a group of models which have multiple non-linear transforming layers used for representing data at successively high-level abstractions. With many layers, these models are expected to be able to solve complicated problems. There are several researches which applied deep learning models for studying DNA sequences. In 2013, Eichholt and Cheng [1] have studied of using boosting and deep networks to predict protein disorder. In this research, the inputs of the predictor are several sequence based features including values from a position specific scoring matrix, predicted solvent accessibility and secondary structure, and statistical characterizations. They have achieved accuracy of 0.82 and an area under the ROC curve of 0.9. In another research of Leung *et al*. in 2014 [2], a model based on deep neural network was trained to predict splicing patterns in individual tissues and differences in splicing patterns across tissues. They used 1393 genomic features extracted from sequences to train the model and achieved significantly better performance in comparison with previous researches' models. One common thing in these researches is that they used expertized features to represent sequences. By doing that, essential information about position of each nucleotide in the sequence was lost which led to the reduction of models' performance. Why don't we use the sequence itself as input for deep learning model? There is a problem that deep learning models receive a numerical vector or matrix, not a sequence of successive letters as input.

In this research, we have proposed a deep learning model using one-hot vector to represent sequences, which conserves the position information of each nucleotide in sequences. This model was inspired from a deep learning model for text classification. With this model, we have achieved improvement of performance in several validation datasets.

In Section 2, we present in detail about the deep learning model we have proposed and how nucleotide sequences are represented to input into the model. In Section 3, we show experiments and results of evaluating the model with 12 validation datasets. In Section 4, we give some conclusions and discussions.

2. Model

2.1. Convolutional Neural Network

The convolutional neural network is a deep learning model with a key idea of using convolutional layers to extract features from input data. It was inspired by visual mechanism of living organisms. In convolutional neural network model, neurons in a convolutional layer are able to extract higher-level abstraction features from extracted features of previous layer.

The most intuitive example which shows abilities of convolutional neural network is faces recognition application presented in the paper of Lee *et al*. [3]. The model used in [3] includes four convolutional layers; each convolutional layer is followed by a sub-sampling layer, then attached to three fully connected neural network layers followed by an output layer. Architecture of the model is shown in **Figure 1**. In this application, neurons

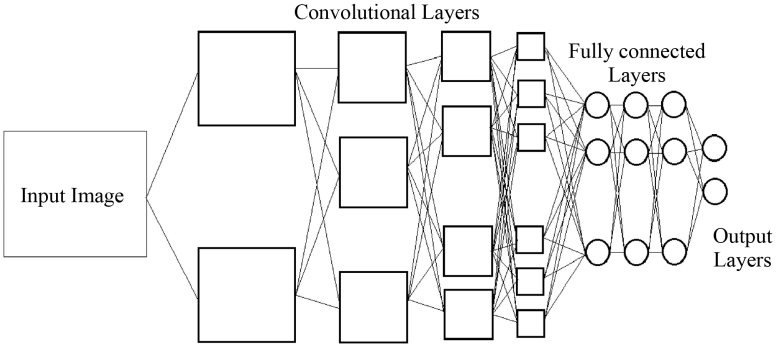

Figure 1. Architecture of convolutional neural network for face recognition.

in first convolutional layer of the model are able to recognize simple shapes such as vertical line, horizontal line or diagonals from pixel input. In second layer, neurons can recognize more complex shapes like curves or end point of lines. In third layer, some parts of a human face such as eye, nose or mouth were recognized. In last convolutional layer, it is able to recognize a whole human face.

With the advantage of extracting features with high level abstraction from data to get better performance, convolutional neural network has been applied in many applications such as image recognition, video analysis, natural language processing.

2.2. Convolutional Neural Network for Text

The convolutional neural network also has been applied in text data problems such as topic categorization, spam detection, and sentiment classification. Unlike digital image data which are two-dimensional numerical matrixes, text data are one-dimensional sequences of successive letters. Therefore, we need to represent it into numerical values so that it could be used as the input for the convolutional neural network. In many researches, a lookup table is used to match each word in vocabulary with a represented vector called word vector. The size of word vectors are fixed for each model and the values of them are either trained as a part of the convolutional neural network or fixed by some other method such as word2vec [4]. However, in a research of Johnson and Zhang [5], they claim that using lookup table is likely using unigrams information, while bi-grams or n-grams information is more discriminating in classifying samples. Therefore, they proposed a different approach that using one-hot vectors to represent words. Then by concatenating word vectors of some nearby words they included the n-gram information into the representation of text.

In **Figure 2** below, we show an example of using one-hot vector to represent text into two-dimensional numerical matrix. Assume that we have a small dictionary D which contains 5 words: “I”, “cat”, “dog”, “have”, and “a”. Each word in the dictionary will be represented by a one-hot vector. Then to represent the sentence “I have a dog”, for each region of successive words (in this case is 2 words) we concatenate one-hot vector of each word in the region to generate a vector representing this region. After representing all regions by this mechanism, we will have a 2D numerical matrix representing the sentence.

The representation matrixes are then directly input to the convolutional neural network model to do the classification. They called the model seq-CNN (a convolutional neural network for sequences). This model has achieved out performances in sentiment classification and topic classification problems.

2.3. Convolutional Neural Network for Sequence

Unlike text data, DNA sequences are sequences of successive letters without space. There is no term of word in DNA sequence. Therefore, we propose a method to translate DNA sequences to sequence of words in order to apply the same representation technique for text data without losing position information of each nucleotide in sequences.

In **Figure 3**, we show an example of translating a DNA sequence into a sequence of words. We use a window with fixed size and slide it through the given sequence with a fixed step’s stride. As in the example, window size equals 3 and step’s stride equals 1. In each step, a segment of nucleotides is read from the window. This read

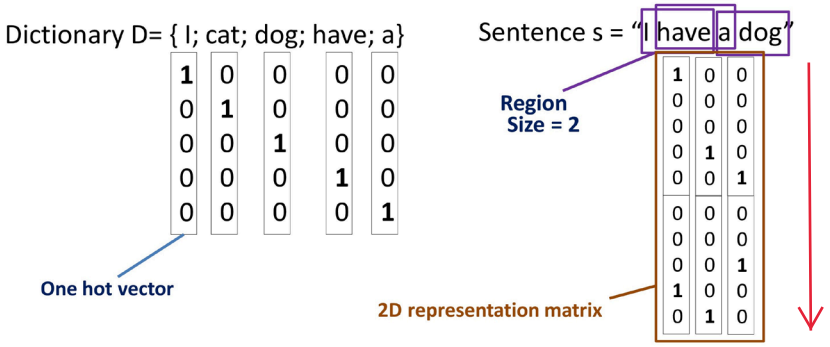

Figure 2. Example of text representation by one-hot vector.

segment is considered as a word and added to the destination sequence. After the final step, we will have a sequence of words which was derived from the given DNA sequence. Then we use the same technique for representing text to represent this generated sequence.

As in **Figure 4**, by using a word size value of 3 nucleotides, we have a dictionary with 64 different words. Each word is then represented by a one-hot vector of size 64. Now using the translating method, we generate a sequence of words from the give DNA sequence. And applying the same representation technique for text we have a two-dimensional numerical matrix which contains information of specific position of each nucleotide in the sequence. This matrix is then used as input for the convolutional neural network model to do classifier.

3. Experiments and Results

3.1. Datasets

In order to evaluate the performance of the proposed model in solving DNA sequence classification problem, we have used 12 datasets. In **Table 1**, we show the detail of these datasets.

Among them, 10 datasets were derived from a research of Pokholok *et al*. [6]. These datasets are about DNA sequences wrapping around histone proteins. It is a mechanism used to pack and store long DNA sequence into a cell's nucleus. Name convention in these datasets is constructed as follows: "H3" or "H4" indicates the histone type, "K" and a succeeding number indicate a modified amino acid (e.g. "K14" denotes the 14th amino acid "K" has been modified), and "ac" or "me" indicate the type of modification (acetylation or methylation) and a number after "me" indicates times of methylation. In each dataset, samples are sequences with length of 500 base pairs and belong to "Positive" or "Negative" class. Samples in "Positive" class contain regions wrapping around histone proteins. In contrast, samples in "Negative" class do not contain them. With these datasets, if we could predict histone profiles from sequences with a certain level of accuracy we might help to understand about expression pattern of genes.

Other datasets are Splice and Promoter datasets, which are benchmark datasets from UCI machine learning repository. The Splice dataset is about the splice junctions. In genes, there are regions which are removed during

Figure 3. Translating DNA sequence into sequence of words.

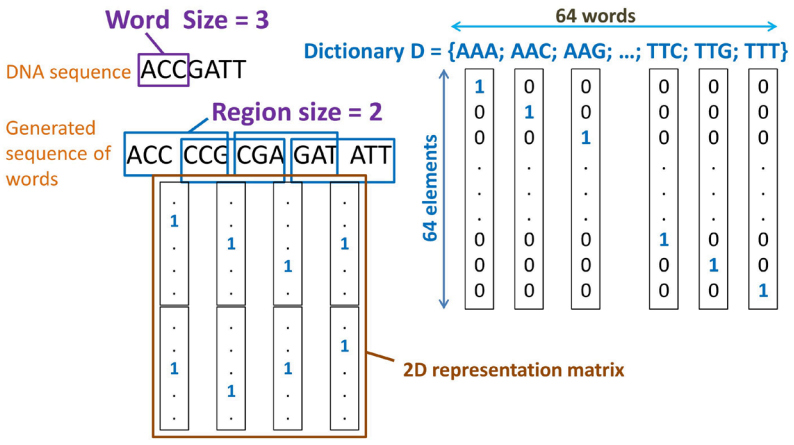

Figure 4. One-hot vector representation of DNA sequence.

Table 1. Datesets description.

No	Dataset	Description	# Classes	# Sample	Sequence Length (base)
1	H3	H3 occupancy	2	7667; 7298	500
2	H4	H4 occupancy	2	6480; 8121	500
3	H3K9ac	H3K9 acetylation relative to H3	2	15,415; 12,367	500
4	H3K14ac	H3K14 acetylation relative to H3	2	18,771; 14,277	500
5	H4ac	H4 acetylation relative to H4	2	18,410; 15,686	500
6	H3K4me1	H3K4 mono-methylation relative to H3	2	17,266; 14,411	500
7	H3K4me2	H3K4 di-methylation relative to H3	2	18,143; 12,540	500
8	H3K4me3	H3K4 tri-methylation relative to H3	2	19,604; 17,195	500
9	H3K36me3	H3K36 tri-methylation relative to H3	2	18,892; 15,988	500
10	H3K79me3	H3K79 tri-methylation relative to H3	2	15,337; 13,500	500
11	Splice	Primate splice-junction gene sequences with associated imperfect domain theory	3	762; 765; 1648	60
12	Promoter	*E. coli* promoter gene sequences with partial domain theory	2	53;53	57

RNA transcription process called introns, and regions which are used to generate mRNA called exons. Junctions between them called splice junctions. There are two kinds of splice junction that is exon-intron junction and intron-exon junction. In Splice datasets, samples are sequences of 60 base pairs length and belong to one of three classes: "EI" (Exon-Intron junction) which contains exon-intron junction; "IE" (Intron-Exon junction) which contains intron-exon junction; and "N" (Neither EI or IE) which does not contain any splice junction. The Promoter dataset is about promoters which are regions near transcription start sites of genes. In RNA transcription, RNA polymerases will bind to these regions to start transcription processes. This dataset contains sequences in length of 57 base pairs belonging to two classes: "Positive" which includes nucleotides from promoter; and "Negative" which does not include nucleotides from promoter. For these two datasets, if we could give a good prediction about profiles of sequences we also help the understanding of gene expression.

To compare performance of our proposed model with traditional approach, we chose 10 datasets used in the research of Higashihara *et al*. [7] in which they used n-mer features as representation of sequences and Support Vector Machine as classifier. In addition, we used two benchmark datasets from UCI machine learning repository to show the performance of our model in dealing with general sequence data.

3.2. Model Configurations

We used CONTEXT, a C++ software package which provides an implementation of convolutional neural network for text categorization on GPU, to implement our model. The model contains 2 convolutional layers. Each of these layers is followed by a sub-sampling layer. These layers are used to extract features from representation matrixes of sequences. The extracted features are then transformed by using a fully connected neural network layer which contains 100 neurons. In this layer, we used a dropout value of 0.5 to reduce the effect overfitting. Finally, a soft max output layer is used to predict labels of input sequences.

Other hyper parameters of the model were chosen based on its performances in datasets for tuning which is held out from validation datasets.

3.3. Evaluation

To evaluate performance of our proposed model in validation datasets, we mainly conducted 10-fold cross validation three times, and calculated the average accuracies. Only for Promoter dataset, we used leave one out method to compare with previous research in which they also used this method.

For comparison, we used the results by previous researches in these validation datasets, which is known as best performance before this research. The previous best results of classifying 10 Histone datasets were taken from [7]. In this paper, they used 4-mer frequency features to represent sequences and used a support vector

machine with RBF kernel as classification model. They also used a feature selection technique to improve performance of their classification. The best previous result on Splice dataset was taken from a paper of Li and Wong [8]. They used C4.5 classifier to achieve this performance. Knowledge-based neural network model was used in a research of Towell *et al*. [9] to achieve the best previous results in Promoter datasets.

3.4. Results

As shown in **Table 2**, our proposed model has significant improvements in all validation datasets. The lowest improvement is nearly 1% of accuracy and the highest improvement is over 6% of accuracy. These improvements are quite high in comparison with other approaches such as finding good representations for sequences or feature selection which were applied before. It showed that features extracted by convolutional layers of the convolutional neural network are very useful for the classifier to classify sequences into true categories.

4. Discussion

In this research, we achieved improvements by using convolutional neural network, a deep learning model with the high power of representing complicated problems. We also used one-hot vectors to represent sequences, so we could preserve specific position information of each individual nucleotide in sequences. With easy-to-classify datasets such as Splice dataset and Promoter dataset, our model achieved nearly perfect classification. It could be used as a reliable tool to support study about these kinds of data. For Histone datasets, there are improvements, but the performances are still low. So for study about these kinds of data, predictions of our model should just be used as references. Since other sequence data in bioinformatics such as amino acid sequence data are also data of sequences of successive letters, our model could be applied in these data.

One limitation of this research is that we empirically chose hyper-parameters such as word size, region size, and configuration of network architecture. Through the results of several experiments, we recognized that these hyper-parameters significantly affected the prediction performance of our model. Therefore, further study about this problem is needed.

5. Conclusions

The convolutional neural network has shown its excellent performance in many study fields. In this research, it also worked well in dealing with A, C, T and G nucleotides of DNA data. By using one-hot vectors to represent DNA sequences and applying a convolutional neural network model, we have achieved significant performance improvements in all evaluation datasets and reached to the state of the art in benchmark datasets.

Since the convolutional neural network has an ability of extracting high-level abstraction features, we will

Table 2. Comparison in performance of proposed model and previous researches' models.

No	Dataset	Previous best accuracy (%)	Accuracy by seq-CNN (%)			Improvement in average (%)
			Minimum	Maximum	Average	
1	H3	86.47	88.79	89.23	88.99	2.52
2	H4	87.32	87.82	88.25	88.09	0.77
3	H3K9ac	75.08	78.59	79.29	78.84	3.76
4	H3K14ac	73.28	77.96	78.34	78.09	4.81
5	H4ac	72.06	77.29	77.52	77.40	5.34
6	H3K4me1	69.71	73.80	74.42	74.20	4.49
7	H3K4me2	68.97	71.12	71.77	71.50	2.53
8	H3K4me3	68.57	74.60	74.76	74.69	6.12
9	H3K36me3	75.19	79.15	79.34	79.26	4.07
10	H3K79me3	80.58	82.59	83.28	83.00	2.42
11	Splice	94.70	95.87	96.73	96.18	1.48
12	Promoter	96.23	99.06	99.06	99.06	2.83

study about extracted features from the model's convolutional layers to see if there are any interesting and meaningful features extracted. By doing that, we could deeply understand how the convolutional neural network worked in dealing with sequences.

With the promising results in this DNA classification, we will try to apply the model in other sequence data of bioinformatics.

Acknowledgements

In this research, the super-computing resource was provided by Human Genome Center, the Institute of Medical Science, the University of Tokyo. Additional computation time was provided by the super computer system in Research Organization of Information and Systems (ROIS), National Institute of Genetics (NIG). This work was supported by JSPS KAKENHI Grant Number 26330328.

References

[1] Eickholt, J. and Cheng, J. (2013) DNdisorder: Predicting Protein Disorder Using Boosting and Deep Networks. *BMC Bioinformatics*, **14**, 88-98. http://dx.doi.org/10.1186/1471-2105-14-88

[2] Leung, M.K.K., Xiong, H.Y., Lee, L.J. and Frey, B.J. (2014) Deep Learning of the Tissue-Regulated Splicing Code. *Bioinformatics*, **30**, i121-i129. http://dx.doi.org/10.1093/bioinformatics/btu277

[3] Lee, H., Grosse, R., Ranganath, R. and Ng, Y.A. (2009) Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical. *Proceeding of the* 26*th Annual International Conference on Machine Learning*, Montreal, 14-18 June 2009, 609-616. http://dx.doi.org/10.1145/1553374.1553453

[4] Mikolov, T., Chen, K., Greg, C. and Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[5] Johnson, R. and Zhang, T. (2015) Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. *Proceeding of Human Language Technologies*: *The* 2015 *Annual Conference of the North American*, Denver Colorado, 31 May-5 June 2015, 103-112. http://dx.doi.org/10.3115/v1/n15-1011

[6] Pokholok, D.K., Harbison, C.T., Levine, S., Cole, M., Hannett, N.M., Lee, T.I., Bell, G.W., Walker, K., Rolfe, P.A., Herbolsheimer, E., Zeitlinger, J., Lewitter, F., Gifford, D.K. and Young, R.A. (2005) Genome-Wide Map of Nucleosome Acetylation and Methylation in Yeast. *Cell*, **122**, 517-527. http://dx.doi.org/10.1016/j.cell.2005.06.026

[7] Higashihara, M., Rebolledo-Mendez, J.D., Yamada, Y. and Satou, K. (2008) Application of a Feature Selection Method to Nucleosome Data: Accuracy Improvement and Comparison with Other Methods. *WSEAS Transactions on Biology and Biomedicine*, **5**, 153-162.

[8] Li, J. and Wong, L. (2003) Using Rules to Analyse Bio-Medical Data: A Comparison between C4.5 and PCL. *Proceedings of Advances in Web-Age Information Management* 4*th International Conference*, Chengdu, 17-19 August 2003, 254-265. http://dx.doi.org/10.1007/978-3-540-45160-0_25

[9] Towell, G., Shavlik, J. and Noordewier, M. (1990) Refinement of Approximate Domain Theories by Knowledge-Based Artificial Neural Networks. *Proceedings of the* 8*th National Conference on Artificial Intelligence*, Boston, 29 July-3 August 1990, 861-866.