A Deep Learning Approach to DNA Sequence Classification:

Conference Paper · July 2016

DOI: 10.1007/978-3-319-44332-4_10

4 authors:

Riccardo Rizzo
Italian National Research Council
121 PUBLICATIONS **1,136** CITATIONS
SEE PROFILE

Antonino Fiannaca
Italian National Research Council
63 PUBLICATIONS **580** CITATIONS
SEE PROFILE

Massimo La Rosa
Italian National Research Council
61 PUBLICATIONS **570** CITATIONS
SEE PROFILE

Alfonso Urso
Italian National Research Council
99 PUBLICATIONS **823** CITATIONS
SEE PROFILE

A DEEP LEARNING APPROACH TO DNA SEQUENCE CLASSIFICATION: FIRST RESULTS

First Author(1), Second Author(2)

(1) First Institute
Affiliation, address, email

(2) Second Institute
Affiliation, address, email

Keywords: Convolutional Network, Deep Learning, Artificial Neural Network, Spectral Sequence Representation, K-mers representation.

Abstract. Deep learning neural networks are capable to extract significant features from raw data, and to use these features for classification tasks. In this work we present a deep learning neural network for DNA sequence classification based on spectral sequence representation. The framework is tested on a dataset of 3000 16S genes and compared to the GRNN that we tested outperform most of the classification algorithm.

1 Scientific Background

Classification tasks are strongly based on the features that represent the objects to classify. In order to build a good representation it is necessary to recognize and measure important detail of the object, but in some cases it is quite difficult to understand which features use, and this affects the performances of the classification model.

Recently neural deep learning architectures or deep learning models, were proved to be able to extract useful features from input patterns. These architecture are mainly applied to image processing and are capable to identify objects on natural images.

The term ”deep” refers intuitively to the number of layers that are used in these networks, and, more precisely, is related to the path from an input node to the output node in the network (considering the network as a directed graph) [3].

Among the deep learning architecture it is usually comprised the LeNet-5 network, or convolutional network, a neural network that is inspired by the visual system’s structure [1]. This network was used for character recognition in the original paper, and for image processing [2] and speech detection [4].

The application of these techniques to gene classification requires a fixed dimension representation of the sequences like the spectral representation based on k–mers occurrences. This representation was used for sequence classification in many works [7, 8, 9]. In particular in our work [9] is noticed that some k–mers are much more important than the other for sequence representation, this means that in the representing vectors there are details that should be taken into account. This observation resembles the problem of feature extraction from image and this idea is at the core of the present work.

In this work we want to understand if the convolutional network is capable to identify and to use these features for sequence classification, outperforming the classifier proposed in the past.

2 Materials and Methods

2.1 *Convolutional Neural Network*

The Convolutional Neural Networks (CNN) are made by a very large number of connections and layers. The one used in this work is a modified version of the LeNet-5

network introduced by LeCun et al. in [1] and is implemented using the python Theano package for deep learning [5, 6].

The LeNet-5 is a network made by two lower layers of convolutional and max-pooling processing elements, followed by two "traditional" fully connected Multi Layer Perceptron (MLP) processing layers, so that there are 6 processing layers.

The convolutional layers calculate L 1-D convolutions between the kernel vectors w^l and the input signal x:

$$q^l(k) = \sum_{u=-n}^{n} w^l(u)x(k-u) \tag{1}$$

In eq. 1 $q^l(k)$ is the component k of the l-th output vector and $w^l(u)$ is the component u of the l-th kernel vector. After a bias term b^l is added and a non–linear function is applied:

$$h^l(k) = \tanh(q^l(k) + b^l) \tag{2}$$

The vector h^l is the output of the convolutional layer. The max-pooling is a non-linear down-sampling layer. In these processing layers the input vector is partitioned into a set of non-overlapping regions (of 2 elements in this implementation) and, for each sub-region, the maximum value is considered as output. This processing layer reduces the complexity for the higher layers and operates a sort of translational invariance. Convolution and max-pooling are usually considered together and are represented in Fig.1 as two highly connected blocks.

In the proposed architecture the first convolutional layer has $L = 10$ filters of 5 elements ($n = 2$), followed by a max-pooling layer of dimension 2, while the second layer has $L = 20$ filters of the same dimension, and the same max-pooling layer.

The two upper level layers corresponds to a traditional fully-connected MLP: the first layer of the MLP operates on the total number of output from the lower level (the output is flattened to a 1-D vector) and the total number hidden units is 500. The output layer has one unit for each class.

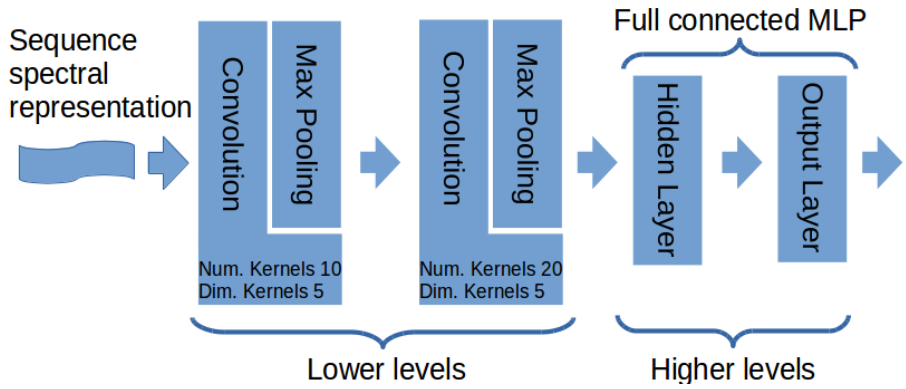

Figure 1: The architecture of the network used.

2.2 *Spectral Representation*

The spectral representation has been used in many bioinformatics works [7, 8, 9] in order to obtain a fixed-size vector representation of genomic sequences. Given a fixed value k, a spectral representation is a vector of size 4^k. Its components are computed by counting the occurrences of small DNA snippets of length k, called k–mers, which are extracted from the genomic sequences by means of a sliding window, with step = 1 and length = k. In case of k–mers containing one or more undefined nucleotides, for example the "N" character, they are discarded. The spectral representation adopts the so

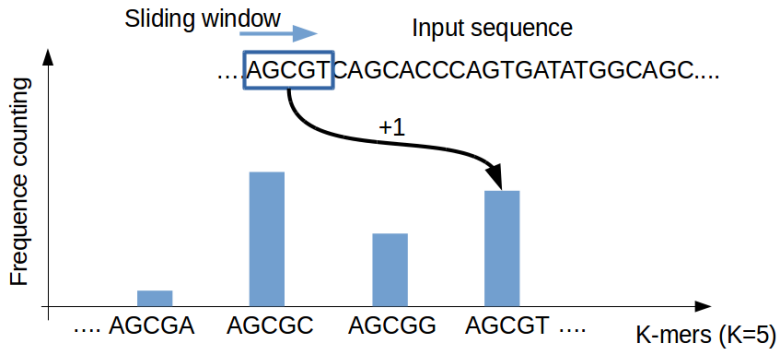

Figure 2: *k*–mers spectral representation.

called "bag–of–words" model, which does not take into account the position of k–mers in the original sequence. This procedure is summarized in Figure 2

2.3 *Dataset of 16S sequences*

The 16S rRNA sequences have been downloaded from the RDP Ribosomal Database Project II repository [10], release 10.27. We randomly selected 1000 sequences from each of the three most common bacteria phyla, Actinobacteria, Firmicutes, Proteobacteria, collecting in total 3000 sequences. All the sequences have length greater than 1200 bp, are classified as type strain, i.e. they are the best representative of their own species, and are certified as "of good quality" by the RDP database.

3 **Results**

Experimental tests have been carried out using the algorithm and the dataset described in Section 2. Two kinds of experiments have been made. In the first case, using a ten fold cross validation scheme, the prediction performances of the CNN have been tested at each taxonomic rank (from phylum to genus) and considering full length sequences. In the second case, the ten-fold cross validation scheme was repeated considering as test set the sequence fragments of shorter size, 500 bp long, obtained randomly extracting 500 consecutive nucleotides from the original full length sequences. This way, we wanted to asses if the network is able to correctly predict the taxonomic rank of the test sequences even if they also contain only a small part (500 bp) of the original information content. In our experiments, we set the k-mers size to k = 5, as done in other works adopting the spectral representation [7, 8, 9]. The CNN has been run considering two different kernels sizes: kernel_0 = kernel_1 = 5 in the first run; kernel_0 = 25, kernel_1 = 15 in the second run. From here on, the first kernels configuration will be named *kern_1*, whereas the second one will be named *kern_2*.

Classification scores, in terms of accuracy, precision and recall, have been compared with another classifier, based on the General Regression Neural Network (GRNN) algorithm, presented in our previous work [9]. The GRNN is a one-pass training neural network, usually adopted for regression purposes, that we adapted for the classification of barcode sequences of animal species, taking into account the COI gene. Moreover we developed three different versions of the GRNN, each one implementing a different distance model: euclidean distance, city-block (Manhattan) distance, Jaccard distance.

In our experiments, the CNN network with *kern_1* configuration always provided better results with respect to the CNN with kern_2 configuration. For this reason, in the following we will only discuss the results obtained with *kern_1* configuration.

All the classification scores have been summarized in the charts of Figures 3, 4, 5. Considering the full length sequences, it is evident that our approach based on the CNN network, with *kern_1* configuration, reaches almost identical scores, with variance lesser than 1%, with regards with the GRNN classifiers based on the euclidean and the city block distance models. Otherwise the GRNN with Jaccard distance model produced

lower results.

Classification scores considering 500 bp sequences showed very interesting results. Our CNN approach, with *kern_1* configuration, clearly outperforms all the other classifiers in terms of accuracy at all taxonomic levels. Only at genus level, accuracy score does not reach the 50%: this behaviour can be explained considering the great number of different genera (393) of the dataset. With regards to the precision chart (Figure 4), the CNN with *kern_1* configuration outperforms the other classifiers at phylum and class level; while at order, family and genus level the GRNN with city block distance model reached better results. This behaviour is, however, balanced if we look at the recall scores (Figure 5). There once again the CNN with *kern_1* configuration always reaches the highest scores, demonstrating that our approach has a better true positive rate, that is the percentage of retrieving correctly classified samples.

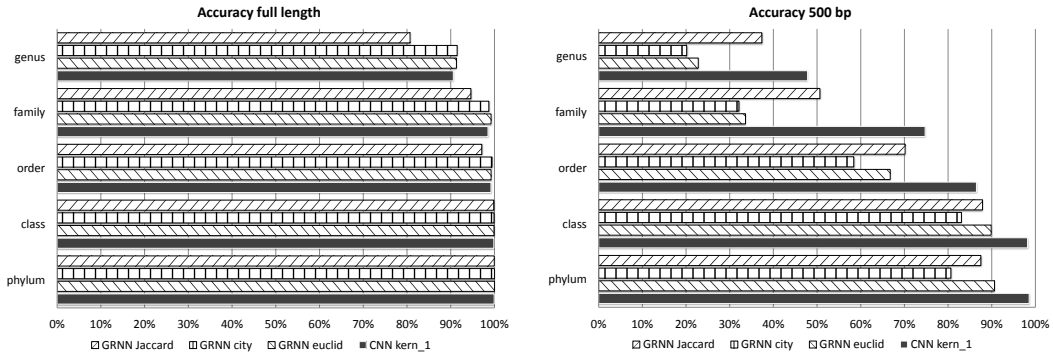

Figure 3: Accuracy scores for full length sequences (left chart) and 500 bp sequences (right chart).

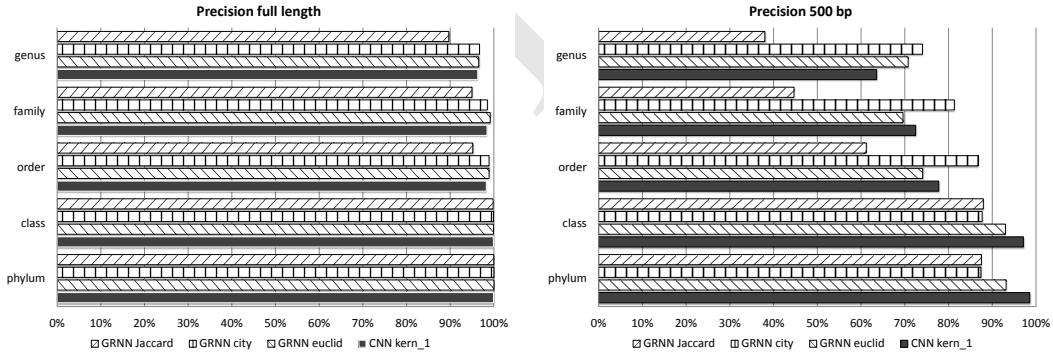

Figure 4: Precision scores for full length sequences (left chart) and 500 bp sequences (right chart).

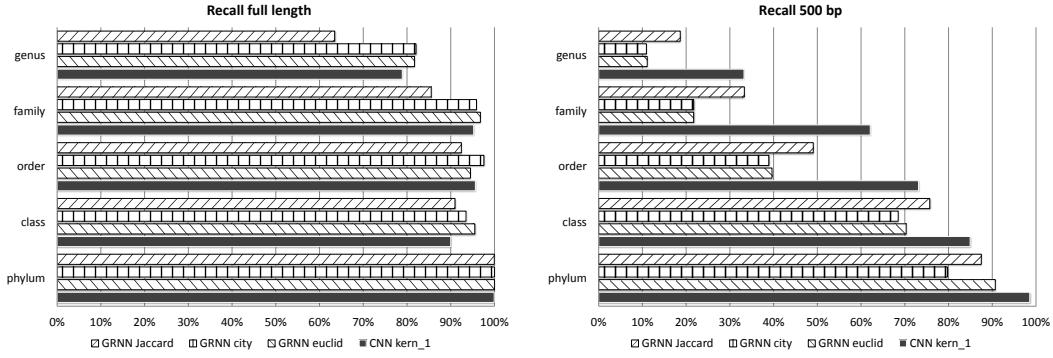

Figure 5: Recall scores for full length sequences (left chart) and 500 bp sequences (right chart).

4 Conclusion

These first experiments confirms that the approach is worth of attention and future work. There are at least two things that need much more investigations: the surprisingly not so good recall results with full length sequences, compared with the 500 bp, that carry much less information and more noise, and the precision results.

It is also strange that a network with a larger kernels is not able to give better results, and we plan to investigate also this problem is future works.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* vol.86, n.11, pp. 2278-2324, 1998.

[2] C. Farabet, and C. Couprie, L. Najman,Y. LeCun, "Learning hierarchical features for scene labeling". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, n.8, pp. 1915–1929

[3] Y. Bengio, "Learning deep architectures for AI." *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.

[4] S. Somsak, A. C. Surendran, J. C. Platt, and C. J.C. Burges. "Convolutional networks for speech detection." *Interspeech*. 2004.

[5] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley and Y. Bengio. "Theano: new features and speed improvements". *NIPS 2012 deep learning workshop*.

[6] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. "Theano: A CPU and GPU Math Expression Compiler". *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 30 - July 3, Austin, TX, 2010.

[7] P. Kuksa, V. Pavlovic. "Efficient alignment-free DNA barcode analytics". *BMC Bioinformatics*, vol.10, Suppl.14, pp.S9, 2009.

[8] A. Fiannaca, M. La Rosa, R. Rizzo, A. Urso. "Analysis of DNA Barcode Sequences Using Neural Gas and Spectral Representation". *Engineering Applications of Neural Networks (EANN)* . Communications in Computer and Information Science, vol.384, pp.212–221, 2013

[9] R. Rizzo, A. Fiannaca, M. La Rosa, A.Urso. "The General Regression Neural Network to Classify Barcode and mini-barcode DNA". *Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB)*. Lecture Notes in Computer Science, in press.

[10] J.R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R.J. Farris, A.S. Kulam-Syed-Mohideen, D.M., McGarrell, T. Marsh, G.M. Garrity, J.M. Tiedje. "The Ribosomal Database Project: improved alignments and new tools for rRNA analysis". *Nucleic acids research*, vol.37(Database Issue), pp.D141–145