Improved Architecture of Artificial Neural Network for Secondary Structure Analysis

Polina Lunina[1], Semyon Grigorev[1]

[1]*Saint Petersburg State University, JetBrains Reserach, St. Petersburg, Russia*

E-mail: *lunina_polina@mail.ru, semyon.grigorev@jetbrains.com*

Motivation

An approach for biological sequences processing by combination of ordinary formal grammars and neural networks is proposed in the work [1]. While classical way is to model secondary structure of the full sequence by using probabilistic grammar, the proposed approach utilizes ordinary grammar only for primitive secondary structure features description. These features can be extracted by parsing and processed by using artificial neural network. In this work we show how to use convolutional networks instead of dense networks, and how improve performance of solution by creation network which can handles sequences instead of parsing results.

Results

Two tasks for evaluation. The first is a classification of tRNA into 2 classes: eukaryotes and prokaryotes (EP). The second one is a classification into 4 classes: archaea, bacteria, fungi and plants (ABFP). We use sequences from open databases [2, 3]. Results are presented in the table. Base model means network which handles parsing result and extended model handles sequences and is based on the corresponding base model.

Classifier	EP		ABFP	
Approach	Vectors	Images	Vectors	Images
Base model accuracy	94.1%	96.2%	86.7%	93.3%
Extended model accuracy	97.5%	97.8%	96.2%	95.7%
Total samples (train:valid:test)	20000:5000:10000		8000:1000:3000	

Questions

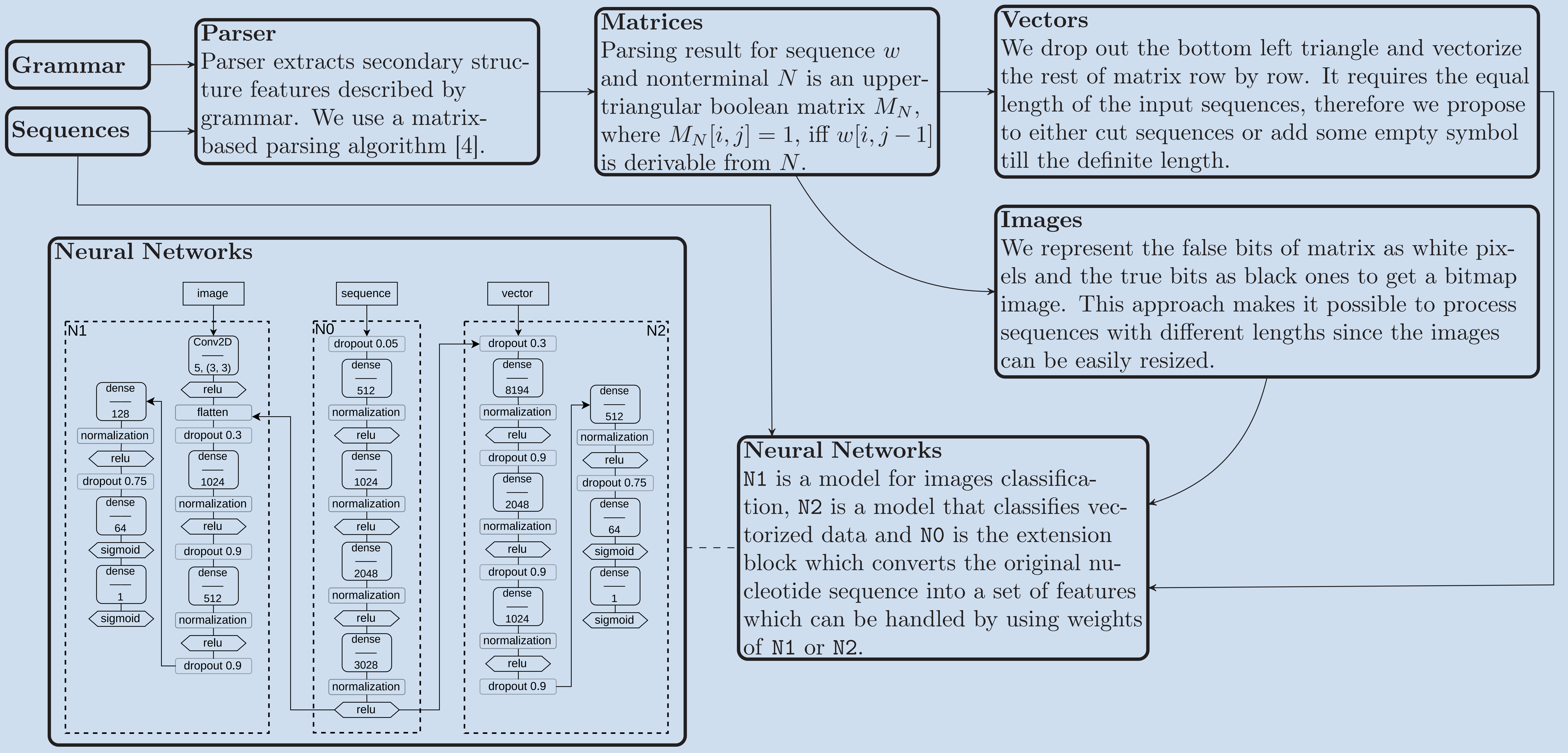
Is it possible to use convolutional neural networks for parsing result processing? The result of parsing algorithm is a set of upper triangular boolean matrices. The original idea is to vectorize these matrices row by row and use DNNs for these vectors processing. Matrices can be treated as bitmaps, where the false bits of matrix correspond to white pixels and the true bits to black ones. To handle these images we use network with convolutional layers followed by linearization and then the same structure as for vectorized data.

Is it possible to move parsing to network training step? Parsing is the most time-consuming operation of the proposed solution. We solve this problem by using two-staged learning. At the first step, we prepare a neural network (vector- or image-based) for our task which takes parsed data as an input. After that we extend trained network with a number of input layers that convert the original nucleotide sequence into parsing result. This way we create a network which can handle sequences, not parsing result. So, parsing is required only for the base network training.

Future Research

- Solve other genomic sequences analysis tasks, e.g. 16s rRNA processing and chimeric sequences filtration.
- Investigate applicability of the proposed approach for proteomic sequences processing, e.g. for proteins functions prediction.
- Utilize generative neural networks for sequences secondary structure prediction.

Solution Overview



Grammar

Sequences

Parser
Parser extracts secondary structure features described by grammar. We use a matrix-based parsing algorithm [4].

Matrices
Parsing result for sequence w and nonterminal N is an upper-triangular boolean matrix M_N, where $M_N[i,j]=1$, iff $w[i,j-1]$ is derivable from N.

Vectors
We drop out the bottom left triangle and vectorize the rest of matrix row by row. It requires the equal length of the input sequences, therefore we propose to either cut sequences or add some empty symbol till the definite length.

Images
We represent the false bits of matrix as white pixels and the true bits as black ones to get a bitmap image. This approach makes it possible to process sequences with different lengths since the images can be easily resized.

Neural Networks
`N1` is a model for images classification, `N2` is a model that classifies vectorized data and `N0` is the extension block which converts the original nucleotide sequence into a set of features which can be handled by using weights of `N1` or `N2`.

Neural Networks

Acknowledgments

The research was supported by the Russian Science Foundation grant 18-11-00100 and a grant from JetBrains Research.

Information

Trained mofels and other materials are pablished at GitHub: `https://github.com/LuninaPolina/SecondaryStructureAnalyzer`.

References

[1] Semyon Grigorev. and Polina Lunina. The composition of dense neural networks and formal grammars for secondary structure analysis. In *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 3: BIOINFORMATICS,*, pages 234–241. INSTICC, SciTePress, 2019.

[2] Genomic tRNA Database. Web page. URL: `http://gtrnadb.ucsc.edu/`. Last accessed 05.06.2019.

[3] tRNADB-CE. Web page. URL: `http://trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/index.cgi`. Last accessed 05.06.2019.

[4] Rustam Azimov and Semyon Grigorev. Context-free path querying by matrix multiplication. In *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, page 5. ACM, 2018.