**Neural network system for prediction and filtering of novelty in text data**

This program implements the neural network of V.Y. Osipov (https://doi.org/10.1016/j.neucom.2018.05.009) and implements the functionality described in the articles (https://doi.org/10.1007/s00521-020-04843-5 , https://doi.org/10.1016/j.eswa.2020.114521 )

Author - Dmitriy Miloserdov (SPC RAS)

**How to use:**

1. Install Python 3.8

2. Install requirements (see requirements.txt)

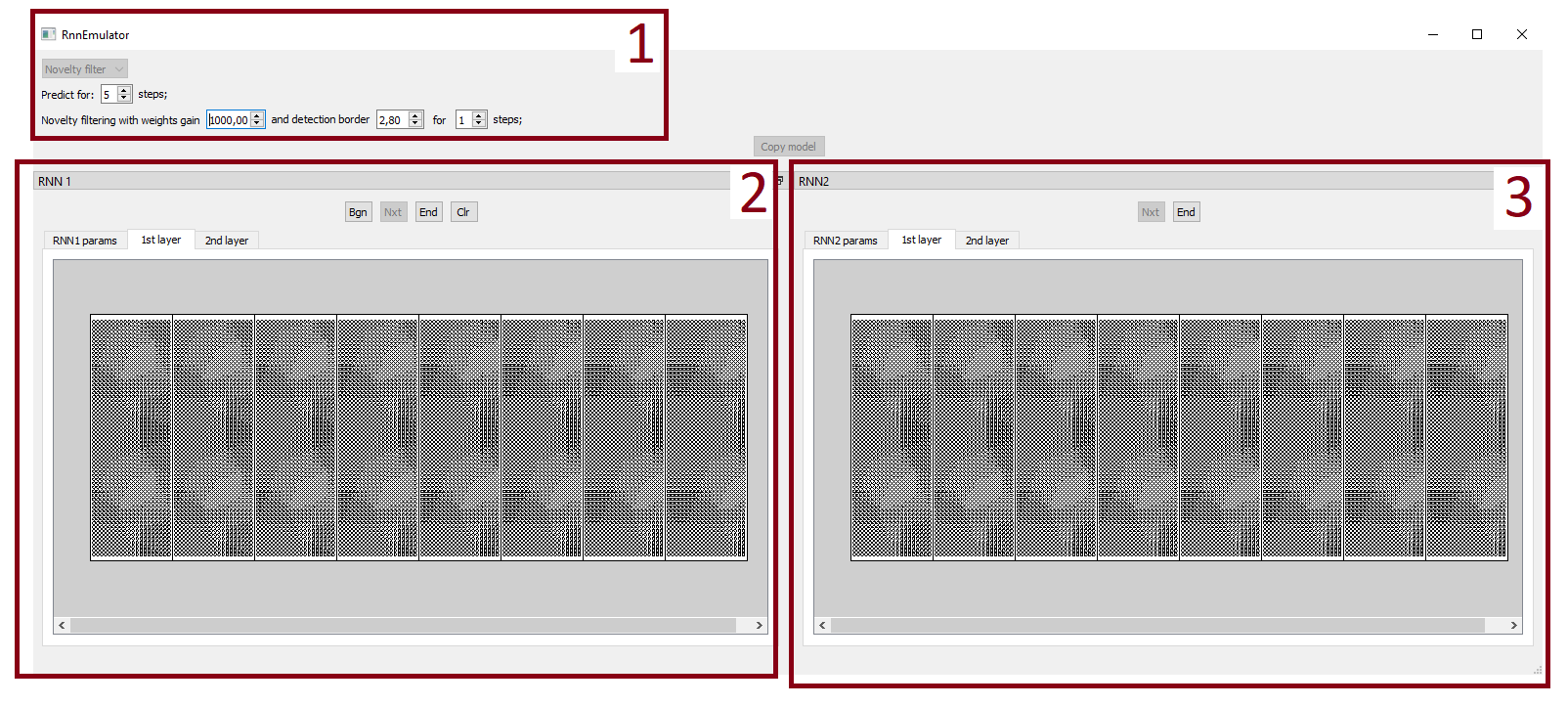
3. Run main.py

It is recommended to run the project in the PyCharm environment. Running in other environments has not been tested and may lead to the loss of paths and dependencies.

**Run Example:**

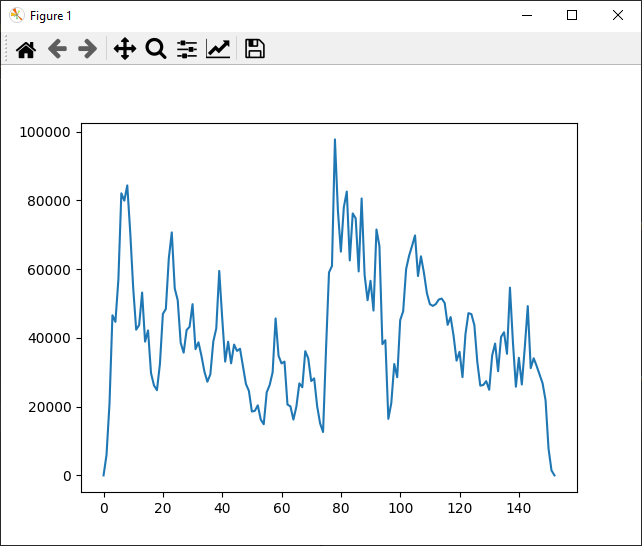
The system is already configured to run an example of novelty filtering in a text data stream.

The program is launched by executing a script main.py . A GUI window will open, consisting of 3 main areas: 1 – basic tinctures, 2 and 3 – visualization of layers of recurrent neural network RNN-1 and RNN-2, respectively.



In area 2, you need to press the “BGN” button, and the program execution will begin.

In the process of execution, text data in encoded form passes through RNN-1 and the dynamics of changes in weights is analyzed. When the novelty detection threshold is exceeded, data from RNN-1 is copied to RNN-2 and filtering is started. Filtered word associations representing novelty are output to the "results\_noveltyFiltering" folder in a text file “results\_xxxxxxxx.txt ”, where xxxxxxxx is the timestamp. At the end of processing the text stream, the graph with the history of the dynamics of changes in the weights of the synapses of the neural network system will also be displayed:



The detection threshold can be adjusted by the "detection border" parameter located in area 1 of the graphical interface (see figure).

To speed up processing, you can disable the visualization of layers. To do this, set the draw\_layers parameter to False in the settings\_common.ini file before starting the program.

**Initial data for processing:**

The source data attached to the program implements the example from the article. They are located in the example\_data\_noveltyFiltering folder. There are two files located there:

stream\_text\_data\_encoded\_to\_SSPs.txt – pre-prepared binary data containing encoded information about word connections for 150 matrices with a size of 1200 connections (for processing on a logical field of 60x20 neurons).

connections\_dictionary.txt – a list of links encoded in this example.

The ini files of the program settings used to filter novelty.

**Forecasting example**

To run the prediction example, you need to replace the files settings\_common.ini, settings\_rnn1.ini and settings\_rnn2.ini in the main directory with those in the example\_data\_forecasting folder, and then call main.py and click the “BGN” button in the RNN1 window. The processing of a sequence of 50 sets of words with a dictionary size of 1000 will begin, in which the prediction will be launched at 35, 40 and 45 steps. The prediction results will be output to the results\_forecasting folder in a text file “results\_xxxxxxxx.txt ”, where xxxxxxxx is the timestamp.

Metrics used to evaluate prediction accuracy include the percentage of misses (pe0) and the percentage of false positives (pe1).