**# Exercise 2 - practical part: RNN -** **predicting the next word**

This code provides an implementation of LSTM and GRU by Zaremba architecture for predicting the next word from the Penn Tree Bank data set.

**## Dataset**

The Penn Tree bank is a widely used data set in natural language processing that consists of annotated sentences from the Wall Street Journal.

**## Model**

The LSTM by Zaremba architecture is used for predicting the next word in the Penn Tree Bank data set. It contains an embedding layer, then 2 LSTM blocks and then FC layers.

**## Run**

First set your relevant paths:

* models dir.
* plots dir. (optional if you would like to see convergence graphs we saved).
* Data dir.

Then choose to train/load model :

* Load Trained Model and Test it (flag = 1).
* Train new model (flag = 0).

Finally pick the method from:

- 'LSTM\_without\_dropout'

- 'LSTM\_with\_dropout'

- 'GRU\_without\_dropout'

- 'GRU\_with\_dropout'

All changes are made in the 'For the Instructor' cell. Then click on 'Run all' or ctrl F9.

**## Results**

The model final accuracy on the train and test dataset is printed below each graph.

* LSTM without dropout - model perplexity train: 39.25 | validation: 119.85 | test: 115.73
* LSTM with dropout - model perplexity train: 43.80 | validation: 96.59 | test: 92.67
* GRU without dropout - model perplexity train: 34.40 | validation: 134.39 | test: 129.95
* GRU with dropout - model perplexity train: 42.22 | validation: 101.53 | test: 97.452

**## Data & Inference**

At the end of the code we added cells that display the data and the transition to the tensor. And an example performed on the trained model, when we enter a sentence and it predicts the next word in the sentence.