

# Deep Learning Course – Assignment 3

## Sequence models

Submission date: 30.06.21, 23:55

Please submit the assignment via Moodle

### General Instruction:

- Solitary submissions or in pairs.
- The code must be written in Python 3.7.1 and run on GPU in google colab
- You are allowed to use only Pytorch, numpy, pandas and matplotlib as external libraries
- The code must be reasonably documented

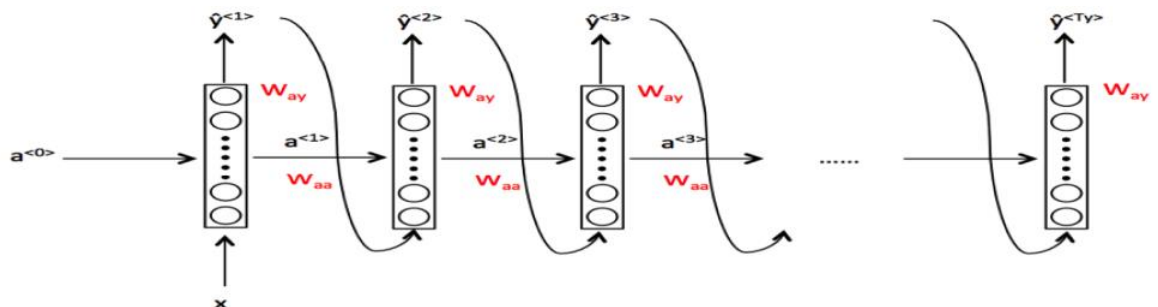
In this assignment you'll implement next word prediction model using RNNs. Then, you'll use this model for text generation in several ways.

### Dataset:

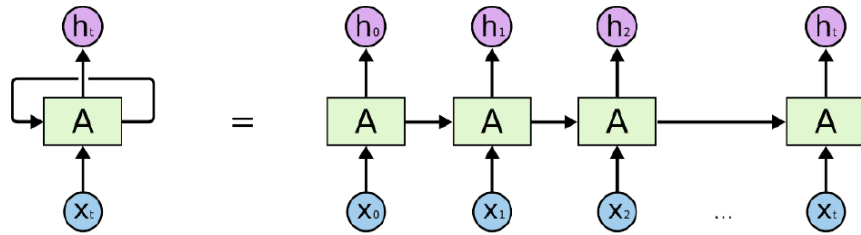
- Use Penn Treebank Dataset (word level). You can find it here: <https://deepai.org/dataset/penn-treebank>
- Load the data and build Dataset and DataLoader for the training set, validation set and test set.
- Select the batch size that works best for you

### Tasks:

1. Write and train an RNN network on the next word prediction task.
  - a. You should tokenize the texts, build a vocabulary and convert the words to ids
  - b. You can use LSTM/GRU/RNN in your model
  - c. You should add an embeddings layer
  - d. Try to achieve a high accuracy as much as you can (on the validation set)
  - e. You can add regularizations and improve the basic model
2. Using your trained model from task 1, implement the following 3 ways to generate a text of 50 words:
  - a. Given an initial vector and the first word. Each time the network generates the next word based on the previous cell output. The input in cell "t" is the max probability prediction of cell "t-1".



- b. Given an initial vector and the first word. Each time the network generates the next word based on the previous one in the real text. The input in cell "t" is the ground truth output in cell "t-1" (the real text).



- c. Beam search, with beam width of 5. (as we learned in class, see <https://medium.com/@dhartidhami/beam-search-in-seq2seq-model-7606d55b21a5> )

Submission instructions:

Your pdf report should contain:

- Model architecture description, training procedure (regularizations, optimization details etc).
- Report the accuracy for task 1 on training set, validation set and test set.
- Write and discuss which way of generating text:
  - Generates a more similar text to the real word
  - Generates the most "non-sense" text.
  - Generates a new text that makes sense

In addition, you should also supply:

- Code able to reproduce your results.
- The main should include a demo of running all the 3 generation ways.
- The best model and its code also should be submitted