Assignment 2: Recurrent Neural Networks

*The deadline for this assignment is **May 20, 2022, 22:00***

You must submit your solution electronically via the Absalon home page.

Important rules and notes:

- *All assignments in the course are individual:*
- *You are not allowed to collaborate with anyone on the assignment and you are not allowed to communicate your solutions to other students.*
- *You must not ask for help from anyone except the teachers and TAs on the course.*
- *On the other hand, we encourage you to use the exercise classes and the Absalon forum to get help. The exercise sessions exist to help you with the assignments, and you are welcome to ask any questions related to the teaching material and the assignments on the forum.*
- *If your solution contains material from other sources than the assignment text, you must cite the source of the material and any changes you have made. This also applies to material from textbooks, Absalon, etc.*
- *If your solution uses methods or notation which are not used in the course material, you must specify where you have found the method or notation.*
- *If you are in doubt about plagiarism or citation rules, please ask the teachers or TAs.*

Please be very observant of these rules. We do not want any plagiarism cases, both for your and our sake.

This assignment has two parts.

Part A. Fill out the Google Form multiple-choice test at:

https://docs.google.com/forms/d/e/1FAIpQLSdzXlMGqOaRKJ5_GHnSKORj9XQRlCovtwCU41zzKOh9Ked69Q/viewform?usp=sf_link

Part B. Write a one-page research report following the instructions in Appendix.

Appendix. You will study the capacity of RNNs and LSTMs by training them on fractions of formal languages and evaluating how well they generalize. The formal languages are:

$$a^n b^n \text{ and } a^n b^n c^n$$

You will train RNNs and LSTMs on randomly sampled members and non-members of these languages **of length 20 or less**. In the case of the first language, this could include:

<aaab,0>
<aaabbb,1>
<aaaaaaabbb,0>

where 0 means the string is *not* a member of the language, 1 means it is. Note that if you find it hard to learn from this very skewed distribution, you are more than welcome to balance the sampling somewhat, sampling more balanced subsets of random members of each class.

It is important that you are as fair to the two architectures as possible. You should do hyper-parameter search - in some way - by splitting your training data - in some way. It is important that your report describes what you did in both cases, e.g., how you searched for

what hyper-parameters, and how you split the data (and why). You subsequently evaluate the selected RNN and LSTM models on members of these languages **of length 21 or longer**.

When evaluating the models you can of course produce a classification accuracy or F1 score. The main result, however, should be a graph, plotting performance (accuracy or F1) over sentence length. This should take the form of a coordinate system with lengths 21 and upwards (say to 50) along the *x* axis and performance values (between 0 and 1) along the *y* axis.

Note: You can implement the RNNs and LSTMs on your own or use existing libraries. If you have additional bandwidth, look into the trade-off between dimensionality and depth of the architectures and their performance.