



Mini-Project: Topics in Deep Learning, TDL211

O. Azencot

Winter 2021

Assignment 2

**Due 4pm, Friday, December 4, 2020**

The goal of this assignment is to learn about computational layers in pyTorch, and to deepen your understanding of Multilayer Perceptrons (MLP), Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM) models by analyzing them. **Submit 4 files (NOT zipped):** MLP.py, RNN.py, LSTM.py and ex2.pdf.

Complete the following tasks.

1. Implement a simple linear layer that allows to change its input features and output features. Ignore the bias. Initialize with random weights. The documentation for the pyTorch linear layer appears here: <https://pytorch.org/docs/stable/generated/torch.nn.Linear.html>. Put your code in an MLP.py file.
2. Implement a basic recurrent layer (Elman) that allows to change its input features, output features and its nonlinearity. Ignore the bias. Initialize with random weights. The documentation for the pyTorch RNNCell layer appears here: <https://pytorch.org/docs/stable/generated/torch.nn.RNNCell.html>. Put your code in an RNN.py file.
3. Implement a basic long-short term memory layer that allows to change its input features and output features. Ignore the bias. Initialize with random weights. The documentation for the pyTorch LSTMCell layer appears here: <https://pytorch.org/docs/stable/generated/torch.nn.LSTMCell.html>. Put your code in an LSTM.py file.
4. Analyze qualitatively and quantitatively the MLP/RNN/LSTM models based on what we discussed in class and based on your code. You should include in your analysis at least 2 qualitative aspects and at least 1 quantitative aspect. Examples: time/space complexity (qualitative), training results on a simple task (quantitative), model features (qualitative), model advantages/disadvantages (qualitative), training time (quantitative), etc. Prepare an ex2.pdf document with your analysis.

**Tips.** For the coding tasks, you are not allowed to use `nn.Linear/nn.RNNCell/nn.LSTMCell`. For instance, here is an example of how to implement the `nn.Linear` layer: <https://hackernoon.com/how-to-build-your-own-pytorch-neural-network-layer-from-scratch-2x6136th>.

Good luck!