# Assignment 5: Sequence Models

CS480/680 – Spring 2019

Out: July 15, 2019
Due: July 30 (11:59pm), 2018

**Submit an electronic copy of your assignment via LEARN. Late submissions incur a 2% penalty for every rounded up hour past the deadline. For example, an assignment submitted 5 hours and 15 min late will receive a penalty of ceiling(5.25) * 2% = 12%.**

**Be sure to include your name and student number with your assignment.**

1. **[50 pts]** Recurrent neural network implementation

   In this question, you will experiment with various types of recurrent neural networks (RNNs) in PyTorch. PyTorch is a popular package for dynamic neural networks that can easily handle sequence data of varying length. For GPU acceleration, it is recommended that you perform your experiments in Google's Colaborary environment. This is a free cloud service where you can run Python code (including PyTorch) with GPU acceleration. A virtual machine with two CPUs and one GPU will run up to 12 hours after which it must be restarted. The following steps are recommended:

   - Create a Python notebook in Google Colab: `https://colab.research.google.com`
   - Click on `edit`, then `notebook settings` and select `GPU` for hardware acceleration.
   - PyTorch is already pre-installed in Google Colab. Get familiar with PyTorch (`http://pytorch.org/`) by going through the tutorial "Get familiar with PyTorch: a 60 minute blitz": `http://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html`

   Answer the following questions:

   (a) **[20 pts]** Encoder implementation

   - Download the file `cs480_char_rnn_classification_tutorial.ipynb` from the course website (associated with the PyTorch tutorial "Classifying Names with a Character-Level RNN".

   Compare the accuracy of the encoder when varying the type of hidden units: linear units, gated recurrent units (GRUs) and long short term memory (LSTM) units. For linear hidden units, just run the script of the Python notebook as it is. For GRUs and LSTMs, modify the code of the tutorial. Hand in the following material:

   - Electronic copy of your code
   - Graph that contains 3 curves (linear hidden units, GRUs and LSTM units). The y-axis is the test (validation) negative log likelihood and the x-axis is the number of thousands of iterations.
   - Explanation of the results (i.e., why some hidden units perform better or worse than other units).

   (b) **[20 pts]** Decoder implementation

   - Download the file `cs480_char_rnn_generation_tutorial.ipynb` from the course website (associated with the PyTorch tutorial "Generating Names with a Character-Level RNN").

Compare the accuracy of the decoder when varying the information fed as input to the hidden units at each time step: i) previous hidden unit, previous character and category; ii) previous hidden unit and previous character; iii) previous hidden unit and category; iv) previous hidden unit. For i), just run the Python notebook as it is. For ii) and iv) modify the code to feed the category as input to the hidden unit(s) of the first time steps only. For iii) and iv), modify the code to avoid feeding the previous character as input to each hidden unit. Hand in the following material:

- Electronic copy of your code
- Graph that contains 4 curves (i, ii, iii, iv). The y-axis is the test (validation) negative log likelihood and the x-axis is the number of thousands of iterations.
- Explanation of the results (i.e., how does the type of information fed to the hidden units affect the results).

(c) **[10 pts]** Seq2seq implementation

- Download the file `cs480_seq2seq_translation_tutorial.ipynb` from the course website (associated with the PyTorch tutorial "Translation with a Sequence to Sequence Model with Attention").

Compare the accuracy of the seq2seq model with and without attention. For the seq2seq model with attention, just run the script of the tutorial as it is. For the seq2seq model without attention, modify the code of the tutorial. Hand in the following material:

- Electronic copy of your code
- Graph that contains 2 curves (with attention and without attention). The y-axis is the test (validation) negative log likelihood and the x-axis is the number of thousands of iterations.
- Explanation of the results (i.e., how does attention affects the results).

2. **[50 pts]** Fake News Detection

- Sign up on the DataCup website: `https://www.defidonnees.com/` by creating an account.
- Join the competition "Leaders Prize: Fact or Fake News?".
- Download the dataset and read the documentation about the data.
- Further instructions will be posted on the course website to complete this part of the assignment.