CS-497: Deep Learning for Natural Language Processing
Group Homework #2

(16.0 + 2.0 bonus points)

Winter 2022

Description: For this assignment, your group will implement your choice of two neural network language models in Python using either PyTorch (preferred) or Keras deep learning platforms. You will train our model on the Wikitext-2 corpora for 20 epochs. For each model, please present:

(i) learning curves of perplexity over the training, validation and test sets,
(ii) final perplexities for the trainings, validation and test sets,
(iii) a list of all hyper-parameters used to train your models, including values selected and a brief description describing you selection method,
(iv) a short description (one or two paragraphs) describing each model architecture, and
(v) observations regarding your results.

There are many codebases available for each of these language models. You may consult these codebases, but you must write your own unique implementation. Similarly, you may confer with other groups, but each group must make a unique submission. You may use the cross-entropy loss functions available in the deep learning platforms. For an extra one-point bonus for each model, you are encouraged to implement these yourself by computing your own logits, softmax and negative log-probabilities. It is okay to use stochastic gradient descent or any of the optimizers provided in the deep learning platforms. To avoid doubt, you should use the built-in functionality of the deep learning platforms to calculate and apply the gradients and updates, respectively.

Listed below are three different neural network language models with a set of hyper-parameter specifications. Select you own values for hyper-parameters not otherwise specified, and explain your choices in the write-ups. Please select two from the following three choices for this assignment:

1. Feed-Forward Neural Network with a sliding window of length 5, a 100-dimensional embedding space, a single hidden layer, tied embeddings and 20 batches during training. (8.0 points)
2. Recurrent Neural Network with 30 time steps, a 100-dimensional embedding space, two hidden layers using a tanh() activation function, tied embeddings and 20 batches during training. (8.0 points)
3. LSTM Neural Network with 30 time steps, a 100-dimensional embedding space, two hidden layers using an LSTM cell provided by the deep learning platform, tied embeddings and 20 batches during training. You should also apply drop-out and gradient clipping using the functionally provided by the deep learning platform. Parameter values for each of these are up to you, and should be described in your write-up. (8.0 points)

Your models should be initialized using a uniform distribution in the range [-0.1 to 0.1]. These models should take less than one hour on a commodity-grade GPU. You are encouraged to follow the architecture presented in a Neural Probabilistic Language Model (https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf), Recurrent Neural Network Regularization (https://arxiv.org/pdf/1409.2329.pdf) or the implementation provided in `zaremba.py` (attached), however you are free to use any architecture. Note: the `zaremba.py` example is implemented in TensorFlow 1.0 and should only be used as a guide. Hyper-parameters can be hard-coded (please group this in a single location in the code) or obtained by parsing from the command line using `argparse` (or something similar).

What to Turn In: Your submission should include (i) the Python code for you language models, preferably a single source file per model and (ii) a PDF of your write-up. All of these files should be submitted as a single zip file.