Homework VI

STAT 5685/6685 - Fall Semester 2023

Please upload all relevant files & solutions into Canvas. Include your answers (minus code) in a single PDF titled `<lastname and initals>_HW6.pdf` and any Python scripts specified in the form of .ipynb notebooks. Any requested plots should be sufficiently labeled for full points. **PLEASE DO NOT UPLOAD COMPRESSED FILES (.zip,.7z, etc.)**

Unless otherwise stated, programming assignments should use built-in functions in Python, Tensorflow, and PyTorch. In general, you may use the `scipy` stack [1]; however, exercises are designed to emphasize the nuances of machine learning and deep learning algorithms - if a function exists that trivially solves an entire problem, please consult with the TA before using it.

By design some responses have been kept as open ended. This means that different students will explore the problems differently and come up with different results by the end of their exploration process. It is not expected for two students to have identical solutions to a given problem.

Google Colab is the recommended environment for this assignment.

Problem 1 - 10 points

1. In class, we learned that neural networks are universal in the sense that they can be used to approximate any continuous function to an arbitrary degree of accuracy. Yet we also learned that feature engineering can give a big boost to learning performance, even when using neural networks. How is it possible for both of these statements to be true? Couldn't a neural network just learn the relevant features for a problem, eliminating the need for feature engineering?

2. In your own words, describe how attention in neural networks works. Make sure you use equations as appropriate.

Problem 2 - 30 points

In this problem we will implement a transformer model called RoBERTa. RoBERTa is an optimized BERT that runs much faster than traditional BERT. You can read the following paper for more details: **https://arxiv.org/abs/1907.11692**.

We are developing an intent classification model. That is, we have seven different intents and our task is to be able to take any sentence in English and classify the sentence into one of these intents. Think of an MNIST like dataset where the output, rather than being one of the numeric labels 1 through 10, is actually one of the seven possible intents. Starter code with prompts is available at:

`https://github.com/KevinMoonLab/STAT-6685/blob/master/HW-6/Roberta.ipynb`

1. Make sure the GPU is enabled in Google Colab. Use Google Colab for this problem.
2. Follow the instructions in the notebook to mount the drive
3. Launch google drive (drive.google.com) in another tab.
4. Download the data folder ***2017-06-custom-intent-engines*** from github. Upload the unzipped folder to your google drive right under **My Drive**. **My Drive** is your default home location in **Google Drive**.
5. Use ***Roberta.ipynb***
6. Throughout the notebook there are prompts that read ***Explain***. Respond to each prompt in your PDF.
7. Once you finish training the model, input the sentences at the end of the notebook to the model and report the output.
8. Try with 5 additional sentences of your own, reporting the sentences and the output.
9. Describe how this simple intent classifier could be used as a component of a Chat Bot. This is an open ended question with different approaches admissible.

Problem 3 - 60 points

(5685) Do two out of the three problems from 3.1, 3.2, 3.3.
(6685) Do all three problems.

Problem 3.1 - (6685-20pts) - (5685-30pts)

In this problem we will implement a LSTM Network. LSTMs have been successfully used for time series forecasting in various applications. Forecasting is a useful skill and we will explore how we can forecast the

number of airline passengers using an LSTM. For this problem all the code is provided, we are just going to experiment with different parameters and discuss what the code is doing. Starter code with prompts is available at:

`https://github.com/KevinMoonLab/STAT-6685/blob/master/HW-6/LSTM.ipynb`

1. (5pts/8pts) Provide a clear explanation for every ***Explain*** prompt in the code.
2. (5pts/8pts) Implement a function that computes the test error (MSE between the test data and the value predicted by the model).
3. (10pts/14pts) Try 3 different values for `seq_length` and 3 different values for `hidden_size`. For each possible combination of these parameters (a grid of 9 combinations in total), include a plot of the time series prediction in your main **pdf** with their corresponding test error. Comment on the influence of these parameters on the results.

Problem 3.2 - (6685-20pts) - (5685-30pts)

In this problem we will look at code that implements MAGAN to generate images using the **MNIST** dataset. You can read the following paper for more details: **https://arxiv.org/abs/1803.00385**. For this problem all the code is provided, we are just going to understand what the code is doing. Provide a clear explanation for every ***Explain*** prompt in the code. Code and prompt are available at:

`https://github.com/KevinMoonLab/STAT-6685/blob/master/HW-6/MAGAN.ipynb`

Problem 3.3 - (6685-20pts) - (5685-30pts)

In this problem we will implement Graph Neural Networks (GNNs). GNNs have recently gained increasing popularity in both applications and research, including domains such as social networks, knowledge graphs, recommendation systems, and bioinformatics. We will implement basic network layers of a GNN, namely graph convolutions, and attention layers. Finally, we will apply a GNN on a node-level task. We will use the Cora dataset, which is a citation network where nodes represent documents. Each node is described by a 1433-dimensional bag-of-words feature vector. Two documents are connected if there exists a citation link between them. The task is to infer the category of each document (7 in total). Starter code with prompts is available at:

`https://github.com/KevinMoonLab/STAT-6685/blob/master/HW-6/GNN.ipynb`

1. (5pts/7.5pts) Read through the tutorial, follow the instruction code, and implement the GCN layer based on the mathematical expression in the class of GCNLayer.

2. (5pts/7.5pts) Read the math and code for the graph attention (GAT) layer. Make sure you understand what a GAT layer does. After that, apply the graph GAT with the same input as a GCN layer. Use the same weights and bias. Use two heads to set to be a vector of arbitrary numbers to obtain different attention values. Print out the the output features. Briefly comment on the difference to the output from the GCN layer.

3. (5pts/7.5pts) Implement the GNNs via PyTorch Geometric. Here we have a MLP as the baseline model. Complete the part marked with *** in *class GCN()*, i.e. replace the linear layers by the GCN layers and complete the forward pass. Compare the accuracy to MLP, and run the code of tSNE visualization. Briefly comment on the results.

4. (5pts/7.5pts) Complete the part marked with *** in *class GAT()*, i.e.implement the GAT layers, complete the forward pass and define the optimizer and criterion. Generate the same tSNE plot after you train the model. Did you obtain a better result than the GCN model?

References

[1] "The scipy stack specification." [Online]. Available: https://www.scipy.org/stackspec.html