Homework 3 (100 points)

Deep Learning

The homework will be due in 2 weeks from the day of release.

In this homework you will implement a 2-layer neural network using any Deep Learning Framework (e.g., TensorFlow, PyTorch etc.). Upload a .txt file with a link to your file as your submission on Submitty.

You need to perform the following tasks for this homework:

In your project, you will pick a dataset (not the same as in the previous homeworks) and describe the problem you would like to solve. Include a link to the dataset source. It is highly recommended that you pick a dataset with at least 10,000 (or more observations). There are many ways of describing a big dataset and one way to describe it is – a big dataset is more complex. Complexity can refer to the number of observations, features, or the type of data. For this project, there is no restriction to the number of features your dataset has. However, having more features gives you greater ability to apply the techniques discussed in class.

Next, you should pick a Deep Learning Framework that you would like to use to implement your 2-layer Neural Network.

Task 1 (10 points): Assuming you are not familiar with the framework, in this part of the homework you will present your research describing the resources you used to learn the framework (must include links to all resources). Clearly explain why you needed a particular resource for implementing a 2-layer Neural Network (NN). (Consider how you will keep track of all the computations in a NN i.e., what libraries/tools do you need within this framework.)

For example, some of the known resources for TensorFlow are:

https://www.tensorflow.org/guide/autodiff

https://www.tensorflow.org/api_docs/python/tf/GradientTape

Hint: You need to figure out the APIs/packages used to implement forward propagation and backward propagation.

Task 2 (60 points): Once you have figured the resources you need for the project, design, and implement your project. The project must include the following steps (it's not limited to these steps):

      1. Exploratory Data Analysis (Can include data cleaning, visualization etc.)

      2. Perform a train-dev-test split.

      3. Implement forward propagation (clearly describe the activation functions and other hyper-parameters you are using).

4. Compute the final cost function.

5. Implement gradient descent (any variant of gradient descent depending upon your data and project can be used) to train your model. In this step it is up to you as someone in charge of their project to improvise using optimization algorithms (Adams, RMSProp etc.) and/or regularization.

6. Present the results using the test set.

NOTE: In this step, once you have implemented your 2-layer network you may increase and/or decrease the number of layers as part of the hyperparameter tuning process.

Task 3 (10 points): In task 2 describe how you selected the hyperparameters. What was the rationale behind the technique you used? Did you use regularization? Why, or why not? Did you use an optimization algorithm? Why or why not?

Task 4 (20 points): Create another baseline model (can be any model we covered so far except a deep learning model). Using the same training data (as above) train your model and evaluate results using the test set. Compare the results of both models (the Neural Network and the baseline model). What are the reasons for one model performing better (or not) than the other? Explain.