CS 725 (Autumn 2018): Assignment 2

This assignment worth **50 points** is due by **11.55 pm on Oct 26, 2018**. No extensions or delayed submissions will be allowed. The submission portal on Moodle will close at 11.55 pm on Oct 26.

Please read the following important instructions before getting started on the assignment.

1. This assignment should be completed individually.
2. This assignment is entirely programming-based and is hosted as a Kaggle task. Click here for further instructions on how to access Kaggle.
    1. Go to the Kaggle site.
    2. Create a new login using your roll number/GPO ID. **This is very important. We will use this to identify where you stand on the Kaggle leaderboard. So, do not use any alternate/fancy names.**
    3. Details of the task are available here.
    4. Please contact the TAs if you need any help. You can also initiate discussions on either Moodle or the discussion forum on the Kaggle page linked above.
3. Your final submission should be a .tgz bundle of a directory organized exactly as described here. Submissions that do not strictly adhere to this structure will be penalized. (Command-line arguments that the code should accept have been clearly listed in the assignment below).

    Name your submission directory using your roll number; it should be organized as follows and submitted as a .tgz file using the command:
    `tar -czf <your-roll-number>.tgz <your-roll-number>`

    ```
    <your-roll-number>/
        |
        |   +-nn.py/nn.cc/nn.m (depending on whether your code
        |                       is in Python, C++ or Matlab, respectively.)
        |   +-README
        |   +-report.pdf (contains plots/numbers asked for in the assignment)
    ```

4. Successful completion of the assignment would include: (A) Submitting `<your-roll-number>.tgz` on Moodle and (B) Having your roll number appear on the Kaggle leaderboard.

Implementing a feedforward neural network-based classifier from scratch

The data used in this assignment is a variant of the data used in Assignment 1. Given featurized Facebook posts, your task is to label them with one of "High", "Medium" or "Low" -- these labels signify the volume of comments the posts received after a certain amount of time has lapsed. For this task, you will implement a feedforward neural network classifier from scratch.

The task/data for this problem is hosted on Kaggle at the following link.

Submit your code as a single file, `nn.py` or `nn.cc` or `nn.m`. (If you are coding in Python, you are allowed to use all inbuilt libraries, `numpy` and `pandas`.) Within the `README` file, clearly mention how we should run your code. The following modules in your code / experiments will be graded:

1. Normalization (e.g. z-normalization) over the given data. **[2 points]**

2. Implement the backpropagation algorithm to compute gradients. Run gradient descent on a cross-entropy loss function with L2-norm regularization. Use exactly the following network configuration and **report the accuracy on `test.csv` within `report.pdf`**. **[25 points]**
   - 1 hidden layer with 100 units
   - Learning rate of 0.001
   - No regularization
   - Sigmoid activation function
   - Batch gradient descent with a batch size of 100

3. Start with an initial learning rate and decay it with time by evaluating performance on a validation set (randomly sampled from the training data, constituting 10% of the original training set). **Explain how you set your learning rate in `report.pdf`**. Add an L2-regularization term to the cross-entropy loss that is scaled by a regularization scaling factor λ. Vary the learning rate, number of hidden layers, number of nodes in each hidden layer and λ and record both the training/validation losses and training/validation accuracies. **Include a table structured as follows within report.pdf that records these numbers.** This table should have at least five rows and should include your most successful runs. **[3 + 10 points]**

Learning rate	Num/size of hidden layers	λ	Training loss	Validation loss	Training accuracy	Validation accuracy
0.001	2 (100, 100)	1e+1	--	--	--	--
0.001	3 (100, 100, 100)	1e+1	--	--	--	--
⋮	⋮	⋮	⋮	⋮	⋮	⋮

4. To learn more about how activation functions influence performance, implement at least two different activation functions for the nodes in the hidden layers. Describe how they behave by **reporting performance on `test.csv` within report.pdf** with using these different activation functions. **[5 points]**

5. For this part, you have free reign. Optimize the neural network to perform as well as possible on the classification task using any techniques you deem fit. **Please describe your innovations**

within report.pdf. The top-scoring students on the private leaderboard for the classification task could gain extra credit worth up to 5 points. **[5 points]**

For part 5, the following strategies might help improve your ranking on the Kaggle leaderboards:

1. **Regularization techniques (dropout, early stopping, etc.)**
2. **Optimization (weight initialization, learning rate schedule, momentum updates in gradient descent, etc.)**
3. **Activation functions (ReLU, tanh, etc.)**