IE643: Deep Learning : Theory and Practice **Jul-Nov 2019**

Assignment 3: Due On 7^{th} October 2019 (11:55 PM IST)

1 Instructions

Answer all questions. Write your answers clearly; explain the assumptions you make and indicate the reasoning behind each statement of your answer. You can score a maximum of 34 marks in this assignment.

Please make sure that all your answers are present in a single pdf document. Upload on moodle, the python code, plots and pdf document as a single zip file named as `IE643_YOURROLLNO_assignment3.zip`. All your files within the zip file should follow similar naming convention. There will be no extensions to the submission deadline.

2 Questions

1. **[Use only Python]** Take the incomplete feed-forward code `Feed_forward(Exercise).ipynb` attached along with this assignment (recall that this code was posted earlier in moodle for you to try some experiments yourself). Consider the dataset in `Assignment3_train_data.zip` attached along with assignment.

   (a) Modify the code so that you use the and Mean Square Error (MSE) and cross-entropy (CE) loss functions. [**4 marks**]

   (b) Use the code to construct a network with appropriate input and output layers based on the train data. In addition, add 3 hidden layers with 300, 500 and 300 neurons. Use sigmoid activation functions in all hidden layers. [**2 marks**]

   (c) Normalize the train data so that all features have values in the range $[0, 1]$. [**2 marks**]

   (d) Use this normalized data for all your experiments. Now, record the following observations:

   (e) Consider the simple stochastic gradient descent algorithm with 100 epochs and mini-batch size 50 for both CE and MSE loss functions. Fix the learning rate to be 0.0001 for both CE and MSE loss functions. Plot the loss obtained on train data against the epoch for MSE and CE loss functions. Distinguish using different colors. Explain your obervations. [**4 marks**]

   (f) Fix the loss function to be MSE. Run the stochastic gradient descent algorithm with 50 epochs and mini-batch size 50 for different learning rates $\{0.1, 0.01, 0.001, 10^{-5}, 10^{-6}\}$. For each learning rate, plot the training loss against epoch. Explain your obervations.[**4 marks**]

   (g) Fix the loss function to be CE. Run the stochastic gradient descent algorithm with 50 epochs and mini-batch size 50 for different learning rates $\{0.1, 0.01, 0.001, 10^{-5}, 10^{-6}\}$. For each learning rate, plot the training loss against epoch. Explain your obervations.[**4 marks**]

   (h) Note down the learning rates for MSE and CE loss function which achieve the best training loss in each case. Now for the best learning rates noted, perform the stochastic gradient descent algorithm with 50 epochs with mini-batch size from the set $\{100, 200, 300, 400, 500\}$. Plot the train loss against epoch for each mini-batch size. Explain your observations. [**4 marks**]

2. **[Use only Python]** Consider the same network and the normalized train data which was used in Question 1. We will now consider the tuning of learning rate using the validation error as a possible criterion.

   (a) Write code to split the training data into two sets called S_1 and S_2 such that S_1 contains 80% of the training data and S_2 contains the remaining 20% of the training data. (We will call the set S_2 as the validation set). [**2 marks**]

   (b) Fix the loss function to be MSE. Choose the learning rates from the set $\{0.1, 0.01, 0.001, 10^{-4}, 10^{-5}, 10^{-6}\}$. For each learning rate, run the stochastic gradient descent algorithm on S_1, with 50 epochs and a mini-batch size of 200. For every 5 epochs, record the error achieved on the sets S_1 and S_2. Now plot these errors for every 5 epochs for each learning rate. Can you come up with a suitable selection procedure for the best learning rate using the experiments conducted? Explain your selection procedure and justify. [**3 marks**]

   (c) Fix the loss function to be CE. Choose the learning rates from the set $\{0.1, 0.01, 0.001, 10^{-4}, 10^{-5}, 10^{-6}\}$. For each learning rate, run the stochastic gradient descent algorithm on S_1, with 50 epochs and a mini-batch size of 200. For every 5 epochs, record the error achieved on the sets S_1 and S_2. Now plot these errors for every 5 epochs for each learning rate. Can you come up with a suitable selection procedure for the best learning rate using the experiments conducted? Explain your selection procedure and justify. [**3 marks**]

   (d) Compare the best learning rate achieved in the previous two scenarios with those obtained in Question 1. Are they same? If not, can you provide appropriate reasons? [**2 marks**]

3 Optional Challenge Question (Must be submitted by October 31, 2019 for consideration)

1. Use only `Pytorch` to construct a suitable feed-forward network of arbitrary number of layers, arbitrary number of neurons in each layer and suitable activation functions so that you get the best possible learned model on the train data. Use an appropriate loss function and a suitable optimization algorithm and learning rates. Also write code where you can accept a test data set and write code to compute and display the accuracy on the test data.

2. Submit your code and model in a single zip file named `IE643_rollno_assignment3_challenge.zip` by October 31, 2019.

3. If the accuracy obtained by your model on a private test set (which will be revealed after October 31, 2019) is ranked among top 10% of all submissions, your submission will be awarded bonus marks which would be considered for your final grades.
