

CS 419M :Assignment 2

March 2021

This assignment is entirely programming-based. The Assignment based contest is hosted on kaggle [here](#)

1. Go to the Kaggle site.
2. Create a new login using your roll number. It is important that you use your roll number and nothing else.
3. Please contact the TAs if you need any help setting this up.

This assignment will familiarize you with training and evaluating feed-forward neural networks. You will work on a classification task where you will have to predict an english capital alphabet given some set of its features. (More details about this dataset are available [here](#).)

Part 1

In this part you will implement a feed-forward neural network using just numpy , train it and report its performance on validation set.

Part 1.A (25 points)

Implement the functions definitions given in **nn.py** to build and train a neural network. Run mini-batch gradient descent on cross-entropy loss function.

For both Part 1.A and Part 1.B, use fixed values of following hyper-parameters:

- Number of epochs: 100
- Batch size for training: 128
- Seed for numpy: 42
- Initialize Weights and Biases using uniform distribution in the range $[-1, 1]$

What to submit in Part 1.A?

For Part 1.A, only code needs to be submitted in the file **nn_1.py**.

Part 1.B (15 Points)

Report Cross-entropy loss on training and validation data using different hyper-parameter configurations given in **part_1b.csv**.

What to submit in Part 1.B?

Fill the table in **part_1b.csv** file given in the repository.

Part 2 (10 Points)

In Part 2, you will evaluate performance of your neural network that you built in part 1 on test data given in test.csv .

In this part, there is no restriction on any hyper-parameter values. You are also allowed to explore various hyper-parameter tuning and cross-validation techniques.

You are also free to create any wrapper functions over given functions in nn.py

Submit your predictions on test data in the Kaggle Competition.

What to submit in Part 2?

Create a two-column csv file **part_2.csv** and write the name of hyper-parameter in first column and value in the second column. **Also submit your code for part 2 in nn_2.py** . This code may contain all the enhancements you did for part 2.

Part 3 (30 points)

In this part you are required to implement a neural network using one of the very popular libraries Tensorflow or Pytorch . Choose one of the library of your liking , learn the basics of that library from the references given below or any other tutorial from internet. And then implement the neural network to predict the Alphabet. In this part also you have full freedom to use any library functions of Pytorch or Tensorflow. You can do this part on [colab](#) or locally . Although colab is recommended.

What to submit in part 3 ?

In this part you are required to submit nn_3.ipynb file.(A Jupyter notebook).The last line of your notebook should print your network's performance on test data.

Some references you might find helpful.

<http://neuralnetworksanddeeplearning.com/>

[Tutorials for Pytorch](#)

[Tutorials for Tensorflow](#)

[nn1](#)

[nn2](#)

[Dropout](#)

[Batch Normalization](#)

[Early Stopping](#)

Submission

- Your submission should contain five files: (i) nn_1.py, (iii) nn_2.py, (iv) part_1b.csv, (v) part_2.csv, (vi) nn_3.ipynb and (vii) readme.txt.
- Use readme.txt to describe any other information needed to run your code successfully.
- Add these files to directory <your_roll_number>.
- Compress the directory <your_roll_number> in .tgz format using following command:

```
tar -czf <your_roll_number>.tar.gz <your_roll_number>
```

- Submit the `<your_roll_number>.tar.gz` file.