TIPR Assignment - II

TA: Anirban

February 14, 2019

Instructions

- This is a coding assignment. The code has to be written in Python. You can use Python version 2 or 3 to solve all the problems. Please mention the version you have used in the report.
- The assignment deliverable is your python code, results, plots and your report of observations.
- Send the link of your github repo to **tipr-e1313@outlook.com** with the following format for the subject name of email: `TIPR_2_<last 5 digits of serial number>`.
- You are encouraged to discuss among yourselves, but **DO NOT COPY** solutions or code. The consequences will be severe.
- Read the complete assignment carefully, before attempting to solve it.
- Follow the instructions provided for how your code is to be run, the formats for input and output files and the naming conventions. — Detailed Instructions
- The submission deadline is **March 4, 2019**.

Best of luck for the assignment. HAPPY CODING!

Taare Zameen Par : Every child is special!

Part-One (40 + 20 Points)

Ishaaan Awasti is a very special child of very special ability. He is *Dyslexic*. He has difficulty with reading, writing and speaking. His school teacher Ram Shankar Nikumbh has observed that he has got a problem with distinguishing numerals. So, he had an idea that if some system can be developed which can help Ishaan identifying numbers, that can be of great help for him. So he seeks out for help! Can you, being a Machine Learning enthusiast, build such a system? Challenge yourself!

The task here is simple yet subtle. You need to build a **Optical Numeral Recognition** (ONR) system using neural network. You have to design a Multi-Layer Neural Network (MLNN) from scratch - everything starting from initialization, forward propagation and backward propagation(commonly coined as back-prop) - everything by yourself. **NO LIBRARY IS ALLOWED**.

Step-1 (5 Points) : Initialize parameters.
Step-1 (15 Points) : Implement forward propagation.
Step-3 (15 Points) : Code out Backward propagation.
Step-4 (5 Points) : Design the cost function.

You need to write four different function for the above four steps and then connect accordingly. Write your code in such a way that the architecture of neural network can be plugged in from outside. Your code should be able to take the number of layers, activation function and number of neurons in each layer as external input (format mentioned later). Remember that the final layer of your neural-net should be a *softmax* layer. You should use *Acccuracy* and *F1-score* as evaluation metrics for you model.

Task - I (2.5 Points)

Experiment with different number of layers of your neural network and give a plot of accuracy vs. layer count, macro-F1 vs. layer count and micro-F1 vs. layer count.

Task - II (2.5 Points)

Try out various configurations of number of neurons in each layer and find the best one. Provide a suitable plot of the same.

Task - III (2.5 Points)

Fix the number of layers and neurons in each layer and feed the hidden layer with the following activation functions - *sigmoid*, *tanh*, *relu* and *swish*. Evaluate the performance of your model. Do it for different configurations of your neural net. Provide a suitable plot of the results obtained.

Task - IV (2.5 Points)

Try out various initialization techniques and observe the results.

Task - V (10 Points)

Check with **keras** library and its MLP implementation and compare the results.

Part - Two (20 Points)

Ishaan's brother Kishaan is also dyslexic, but he has got another problem - he has difficulty recognising animals. You have to design a simplistic tool which can differentiate between cats and dogs. Repeat all the tasks mentioned in the earlier scenario. Also, do not forget to compare your results with the keras library.

Part - Three (20 Points)

For all the three datasets provided in the first assignment, check if the MLNN is performing better than what you obtained using Bayes classfier or NN-classifier.

Datasets Description

1. MNIST

It is an *image dataset* consisting of total 42000 images of size 28×28. It contains images of numerals 0 to 9.

2. Cat-Dog

It consists around 25000 images of cats and dogs. The size of the individual image is 200×200.

How to run

Far each of the two datasets mentioned above, there will be hidden test files. The performance of your code will be tested on these test data. I should be able to run the code in the following format:

To test the model :
`python main.py --test-data <test_data_directory_path> --dataset <dataset_name>`

To train and then test the neural-net :
`python main.py --train-data <train_data_directory_path> --test-data <test_data_directory_path> --dataset <dataset_name> --configuration <layer_config>`

Assignment Deliverable

- Prepare a report and name it `TIPR_Report_2.pdf`. In your report, you need to briefly describe what you have done, present the results (in a form you think is good) and provide a brief discussion of the results you

obtained. It should contain the answers of the tasks, plots and output data for the given datasets. Please make it concise and to-the-point.

- The name of the plots asked for should follow the following naming convention: `part_i_task_j.png`. For example, the filename for part-one task-I should be `part_1_task_1.png`. Put these plots into the `output_plots/` directory.
- The name of output file for various tasks needs to be `part_i_task_j.txt` or `part_i_task_j.pdf`. Put them into the `output_data/` directory.

Sample Input and Output

The sample format for running the code when I want to just test your neural network is as follows:
`python main.py --test-data <test_data_directory_path> --dataset MNIST`

The sample format for running the code when I want to both train and test your neural network with a pluggable configuration is as follows:
`python main.py --train-data <train_data_directory_path> --test-data <test_data_directory_path> --dataset MNIST --configuration [3 5 7]`

In the previous comman, the value corresponding to the configuration parameter signifies that there will be three hidden layers having 3, 5 and 7 neurons respectively. The input and output layer configurations you will set yourself.

Here is how the sample output would look like in the terminal when I would run the code with test data for, e.g., *MNIST* dataset.

```
Test accuracy ::  78.32
Test Macro F1-score ::  47.35
Test Micro F1-score ::  53.23
```