Machine Learning

COMP3611

Coursework: Multi-layer Perceptron

This coursework comprises 10% of the marks for this module. Deadline: **9/11/2020** at 10:00 am, by electronic submission in Minerva.

Implement the following specification in Python and using only NumPy. This coursework is individual and must not be completed in group or by using code snippets from the Internet. The code will be checked for plagiarism.

Introduction

The MNIST database of handwritten digits has a training set of 50,000 examples, validation set of 10,000 examples and a test set of 10,000 examples. The digits have been size-normalized and centred in a fixed-size image. Each image is 28x28 pixels and labelled into one of 10 categories [0-9]. We will only use a subset of this dataset for the coursework: the first 10,000 examples of the training set for training and 1,000 examples of the testing set for testing.

Prerequisite

In this Coursework you will design and train a multi-layer neural network to perform image classifications on the MNIST dataset.

In order to run the code necessary for this coursework, you need to install the following packages inside an anaconda environment or you can work on Google Colab.

```
pip install numpy
pip install jupyter
pip install matplotlib
```

This setting is the exact same setting we used for the weekly labs.

Design of a multi-layer NN with NumPy

You are going to use only the NumPy library to design a multi-layer perceptron. The network will have two hidden layers and an output layer. Each hidden layer will have a sigmoid function as its activation function. The output layer will have a softmax function as its activation function, so that we will be able to interpret the output of the network as a probability distribution.

Exercises

Implement Exercise 1 and 2 in the **'mlp.py'** file and test your implementation in the Jupyter Notebook.

Exercise 1

Start by opening the given **'mlp.py'** file and implement the forward phase of the network training procedure. This is similar to the lab implementation where we went through implementing the MLP algorithm with one hidden layer and linear output activation function. However, here, you are asked to do it with two hidden layers and the output layer activation function is softmax. Implement this inside the 'forwardPass' method of the MLP class in the given **'mlp.py'** Python file

[6 marks]

Exercise 2

Implement the backward phase of the network training procedure. Implement this inside the for loop in the 'train' method of the MLP class in the given **'mlp.py'** Python file. Test your implementation in the notebook. You can train the network with the default parameters setting in the notebook. Once the network is trained for 1000 iterations, evaluate the network by testing its performance on the testing dataset. The testing accuracy will likely be less than %50.

[10 marks]

Exercise 3

We can improve the performance of the network by experimenting with different parameter setting. Find the parameter setting that achieves more than 90% testing accuracy. You can change the number of iterations, the learning rate value, and the number of neurons in each hidden layer. You can also experiment with training the network with gradient descent with momentum. The goal is to get the testing accuracy to be more than %90. Once you have found the best parameter setting that achieved the desired accuracy, please run the cell in the notebook that saves the parameters and the network weights in a pickle file. This exercise gives you a chance to experiment with architecture and parameter optimization, a fundamental step in real-world use of neural networks.

[4 marks]

Submission Instructions

Submit a single zip file named after your email **username** (use only **zip** extension: no gz, and especially no rar), containing exactly three files, the final Python file 'mlp.py', the Notebook file and the pickle file that has the weights and the parameter setting of your best network.

Uploads to Minerva can fail, or you can unintentionally forget to include a required file. It is your responsibility to **download your submission** and check that everything is correct. I will only accept submissions through Minerva.