

CS-5007 Deep Learning

Assignment 3 - FNNs

23 March, 2021

Instructions

1. Submit the assignment as a single notebook(*.ipynb).
2. Follow the following naming format strictly: $\langle name_roll \rangle_Assignment_3.ipynb$
3. Try to solve each sub question in a new cell wherever possible. This improves the notebook's readability.
4. Please note down your observations in text blocks below the corresponding code blocks.
5. Mention and justify your assumptions if any.

Introduction

In this assignment, we will perform two simple tasks using feed forward neural networks. The first task will be regression and the second one will be classification.

Task 1: (2 marks)

Dataset creation:

Sample 2000 points from the uniform distribution $\mathcal{U}(-100,+100)$. This will be your input X . Define y to be the square of X (i.e $y_i = x_i^2 \dots \forall x_i \in X$). This forms the output.

FNN construction and training:

1. Construct a single hidden layer neural network. Decide the best hyper-parameters for it.
2. Train this neural network using no more than 5000 epochs on the above generated dataset. This will be your baseline performance.
3. Add one another layer in the network. Train again with no more than 5000 epochs. Achieve an accuracy better than or equal to the baseline.
4. Experiment, if accuracy can be further improved by adding some more layers. Note your observations and analysis.
5. Plot the output of the model for all $x_i \in X$ and compare with that of true y .
6. [Ungraded] Want to see the power of activation functions? Try removing them from all the layers in the above network, train again and plot the outputs. No matter how long you train, no matter how many layers you add (making sure the network is not memorising each and every data-point), you will see similar output plot as with a single hidden layer network without activation. What does this suggest?

Task 2: (3 marks)

Dataset:

Download the MNIST hand-written digits image dataset from here: <http://yann.lecun.com/exdb/mnist/>. It is considered the *hello world* of image classification task. It consists of 60,000 train and 10,000 test images of hand written digits ranging from [0-9]. The images are of size 28×28 . Build a classifier for this dataset using a multi-layer perceptron(FNN).

Model construction:

1. Decide the hyper-parameters like *number of layers, number of neurons in each layer, batch size, learning rate, etc.*
2. Split the training data into train-validation sets. Appropriately choose a train-validation ratio.
3. Train and tune the network using this base structure on train-validation data. This will be your baseline performance.
4. Plot the train and validation accuracy and loss curves. Lets call them the *performance plots*.
5. Add batch normalization layers after each layer. See if the performance(accuracy) improves. If yes, continue with this model, else stick with the previous model. Note your observations. Note the changes in the plots if any.
6. Now, again train using dropout regularization(add a drop-out layer). Check the performance of the model. Note the observations in the performance plots.
7. Test the best model(among the 3 above), on the test dataset. report the results in a tabular format.

* * *