
Assignment 3

CSE 343/543 : Machine Learning

Due: 11:59PM, Oct. 27, 2017

Note: Please complete programming as well as theory component. Please start on time, this will be a time consuming assignment.

Submission (on Backpack) : Code + theory.pdf (a legible copy of scanned answers to theory questions) + report.pdf (a report explaining all your codes, plots and approaches)

Programming

Exploring data sets:

In this assignment, you need to use two datasets:

- MNIST: <http://yann.lecun.com/exdb/mnist/>
- An MNIST subset: **dataset_partA.h5**

Please explore both these datasets on your own, i.e, number/kind of features, number of data points, number of classes. Please ensure you look into this information since you will need to handle data accordingly in the section below.

Neural Networks:

1. **Objective #1 [80 marks]:** Implement the forward and backpropagation algorithms to train an artificial neural network from scratch. You are not permitted to use any external libraries in this part.
 - a. Implement a 2 layer neural network with 100 hidden units in the first layer and 50 in the next. Implement sigmoid activation function in each layer. Train/test your model on the MNIST subset. Report accuracy as the evaluation metric in this part. Save the weights of your best model in a file. **Please ensure that while your model is a 2 layer network, your code is generalisable and can be used out of the box for an n-layer network (i.e, do not hard code equations for the 2 layers).**
 - b. Implement a 2 hidden layer network with 100 hidden units in the first layer and 50 in the next. Implement softmax at the output layer and sigmoid in every other layer.

Train/test your model on the entire MNIST dataset. Report accuracy as the evaluation metric in this part. Save the weights of your best model in a file.

- c. Implement ReLU and Maxout, and repeat part A and B using these activations at every layer (except output).
-
2. **Objective #2 [20 marks]:** Use the Multi-Layer Perceptron model in sklearn to create an artificial neural network.
 - a. Implement a 2 layer neural network with 100 hidden units in the first layer and 50 in the next. Train/test your model on the MNIST subset, using sigmoid activation in each layer. Report the accuracy, **as compared** to what you received in 1A. Explain reasons for an observed difference in accuracies, if any.
 - b. Implement a 2 layer neural network with 100 hidden units in the first layer and 50 in the next. Train/test your model on the entire MNIST dataset, using softmax at the output layer and sigmoid activation in every other layer. Report the accuracy, **as compared** to what you received in 1B. Explain the difference in accuracies, if any.
 3. **Bonus [15 marks]:** Use the Multi-Layer Perceptron model in sklearn to create an artificial neural network.
 - a. Experiment with network structure and find one that outperforms the ones you tried above. Please experiment with at least **3** different network structures and provide your results. Try to provide some insight on the model structures you tried and why the best performing structure performs better than the others

Note:

- The assignment is very time consuming, so please do start on time. The training time for the models will be considerable, so starting a day before the deadline will not suffice.
- **The emphasis of this assignment is on implementing the neural network yourself (Objective #1).** The idea behind Objective #2 is just to evaluate how your self-written model performs; Objective #1 carries majority of the marks.
- Your code for Objective #1 should be executable. We will run it at the time of the evaluation, so please ensure both training and testing (including loading weights from the file) are in working condition

Theory Questions

1 Theory: Question 1

(5 marks) Can a neural net of arbitrary depth using just linear activation functions be used to model the XOR truth table? Can you mathematically prove this classifier equivalent to be same as another classifier discussed in class?

2 Theory: Question 2

(8 marks) Assume you have a given data-set with some labeling, which you try to train using a neural network of n layers. Each input datum is a m -dimensional array, with each value in the range $[0,1000]$. Your peer 'X' uses sigmoid activation in the network's layers, but is unable to train the model successfully. Assuming that there is no problem in the model's architecture, what could be the possible problem? Explain in terms of the activation function and back propagation. What would happen if X used ReLU instead? Would the problem worsen or get better? Suggest a data pre-processing technique (for both settings) to remedy this problem.

3 Theory: Question 3

(7 marks) The quadratic (squared error) cost function poses a "learning slow-down" problem (learning is slow when the error is large). Explain mathematically how the cross-entropy cost function solves this problem.