|  |
| --- |
| **COGS 260: Assignment 3** |

**Apoorve Dave**

**A53103070**

**a1dave@ucsd.edu**

**Abstract**

A report on Homework 3 of COGS Image Recognition Assignment.

**1 Perceptron Learning**

* 1. **Linear Separability**

As seen from below plots, across all pairs of dimensions, the two classes are linearly separable.

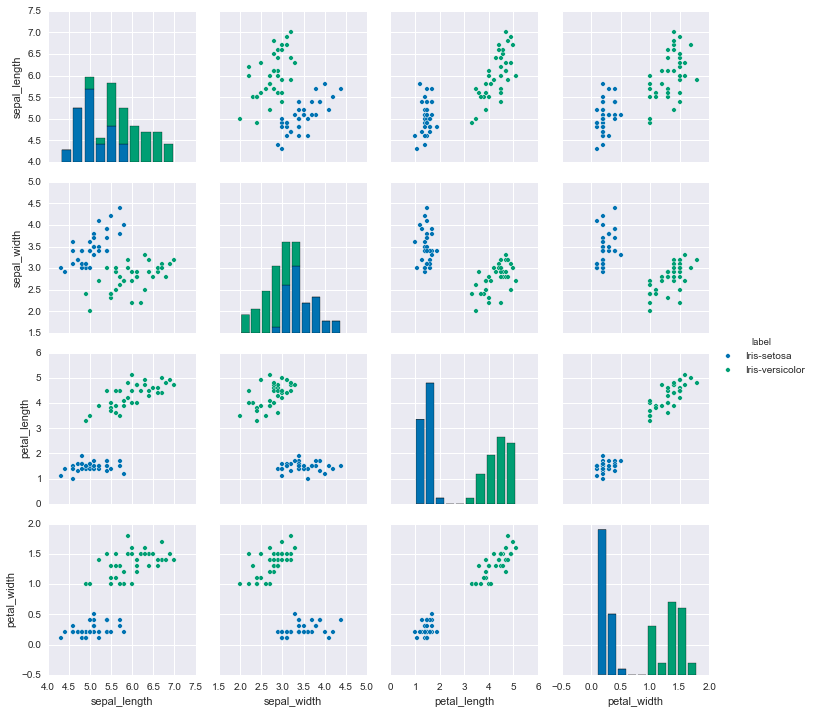
****

Figure 1: Linear separability

* 1. **Perceptron Learning without Z-scoring**

Without Z-scoring the data, it takes slightly more time(3 iterations over data set) for convergence.

Learning rate

Number of iterations = 3

Accuracy on test set = 100%

* 1. **Perceptron Learning with Z-scoring**

Observation: With Z-scoring, it takes just one iteration over data set to reach the 100% accuracy on test set.

Learning rate

Number of iterations = 1

Accuracy on test set = 100%

1. **Feed Forward Neural Network**
   1. **NN with 1 hidden layer**
      1. **Weight update rule:**
      2. **Hyperparameters:**

**Architecture:** 784 -> 30 -> 10

**Learning Rate:** 3/#samples

**#Epochs:** 30

**Plots:**



* 1. **NN with 2 hidden layers**

Performance was slow as the network complexity increased. Higher accuracy on training data as well as test data was observed because of network learning more complex features.

**Architecture:** 784 -> 30 -> 30 -> 10

**Learning Rate:** 3/#samples

**#Epochs:** 30

**Plots:**



* 1. **Regularization and Momentum**
     1. **Regularization Only:**

As expected, the model performed better after regularization as it was able to generalize better.

**Architecture:** 784 -> 30 -> 10

**Learning Rate:** 0.1/#samples

**Regularization(lambda)**: 5/#samples

**#Epochs:** 30

**Plots:**



* + 1. **Momentum Only:**

Achieved faster convergence with momentum value of 0.9. Below are the figures for the same.

**Architecture:** 784 -> 30 -> 10

**Learning Rate:** 0.1/#samples

**Momentum(mu)**: 0.9

**#Epochs:** 30

**Plots:**



* + 1. **Momentum And Regularization:**

**Architecture:** 784 -> 30 -> 10

**Learning Rate:** 0.1/#samples

**Momentum(mu)**: 0.9

**Regularization(lambda)**: 5/#samples

**#Epochs:** 30

**Plots:**

****

1. **CNN**

**Base Network Architecture:**

Model Layers in order:

Spatial**Conv**olution(Filter size 5x5)

Spatial**MaxPool**ing(Filter size 3x3)

**ReLU**

Spatial**Conv**olution(Filter size 5x5)

**ReLU**

Spatial**MaxPool**ing(Filter size 3x3)

Spatial**Conv**olution(Filter size 5x5)

**ReLU**

Spatial**MaxPool**ing(Filter size 3x3)

Linear(64\*3\*3 -> 64)) – **(Fully Connected)**

Linear(64 -> 10)) - **(Fully Connected)**

**LogSoftMax**

* 1. **SGD**

Iterations for convergence

Test Accuracy

Plots

* 1. **Batch Normalization**

Iterations for convergence

Test Accuracy

Plots

* 1. **Replace the fully connected layer by average pooling layer**

Iterations for convergence

Test Accuracy

Plots

* 1. **Adaptive Gradient (Extra credit)**

Iterations for convergence

Test Accuracy

Plots

* 1. **Nesterovs Accelerated Gradient**

Iterations for convergence

Test Accuracy

Plots

* 1. **RMSprop**

Iterations for convergence

Test Accuracy

Plots

**Reference:**

1. [**http://neuralnetworksanddeeplearning.com/**](http://neuralnetworksanddeeplearning.com/)
2. [**https://github.com/gcr/torch-residual-networks/blob/master/train-cifar.lua**](https://github.com/gcr/torch-residual-networks/blob/master/train-cifar.lua)