|  |
| --- |
| CSE 440 |
| Assignment 7 |
| Coursera – Week 4 & 5 |

Submitted by-

Mahbuba Tasmin

1610064042

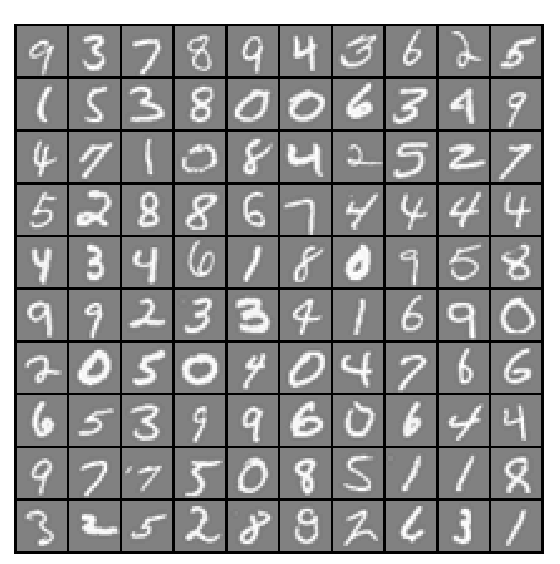
Submitted to

M Ehsanul Karim

**Week 4 : Multi class classification and Neural networks**

Objective: In this exercise, one-vs-all logistic regression and neural networks will be implemented to recognize hand written digits(from 0 to 9).

Visualizing the data :



**Vectorized Logistic Regression :**

In this part of the exercise, logistic regression will be re used with regularization that will be vectorized implementation.

Testing lrCostFunction() with regularization brings outputs as:

Cost: 2.534819 Expected cost: 2.534819

Gradients: 0.146561 -0.548558 0.724722 1.398003

**One-vs-all Classification :**

In this part of the exercise, one-vs-all classification will be implemented by  
training multiple regularized logistic regression classifiers, one for each of  
the *K* classes in our dataset (Figure 1).

After running 50 iteration on each of k (=10) classes in the given dataset, the accuracy from the associated costs of each class comes as 94.980000.

**Neural Networks**  
In this part of the exercise, an already trained neural network will be implemented to recognize handwritten digits using the same training set as before. The goal is to implement the feedforward propagation algorithm to use our weights for prediction.

**Feedforward Propagation and Prediction**  
In the feedforward computation, similar to the one-vs-all classification strategy, the prediction from the neural network will be the label that has the largest output (*hθ*(*x*))*k* .

In this case, the accuracy comes as 97.5% , already trained data sets are recognized with a higher accuracy when implemented with neural network. After that, an interactive sequence will launch displaying images from the training set one at a time, while the console prints out the predicted label for the displayed image.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Week 5 – neural network learning

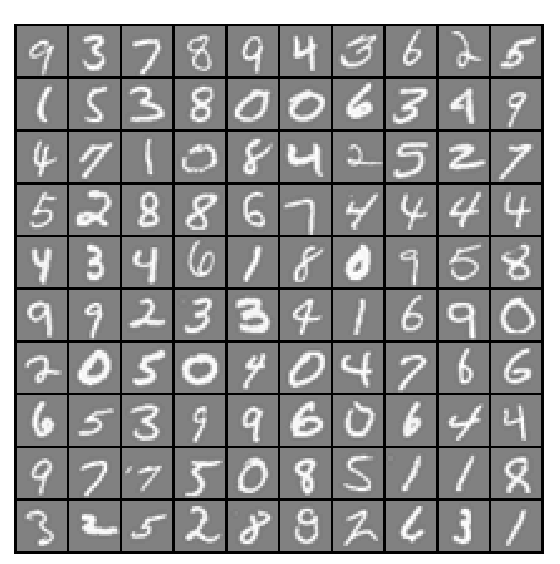
Objective:

Implement the backpropagation algorithm for neural networks and apply it to the task of hand-written digit recognition, along with learning the parameters for the neural network.

Intuition:

The intuition behind the backpropagation algorithm is as follows. Given a training example (*x*,*y*), we will first run a "forward propagation" to compute all the activations throughout the network, including the output value of the hypothesis *hƟ*(*x*). Then, for each node *i* in layer *l*, we would like to compute an "error term" \delta^{(l)}_i that measures how much that node was "responsible" for any errors in our output. For an output node, we can directly measure the difference between the network's activation and the true target value, and use that to define \delta^{(n_l)}_i (where layer *nl* is the output layer). For hidden units, we will compute \delta^{(l)}_i based on a weighted average of the error terms of the nodes that uses a^{(l)}_i as an input.

Visualizing the dataset :



Feed forward and cost function :

We will see what the neural network currently predicts given the weights and biases in the dataset; first, without regularization and second,with regularization.

|  |  |
| --- | --- |
| Lambda (λ) | Cost at parameters |
| 0 (without regularization) | 0.287629 |
| 1(with regularization) | 0.383770 |

**Initializing Neural Network Parameters and checking backpropagation :**

Difference between two columns (Left- Numerical Gradient, Right-Analytical Gradient) with and without regularization is as follows :

|  |  |  |
| --- | --- | --- |
| Regularization | Standard result | Obtained result |
| without | less than 1e-9 | 2.33553e-011 |
| with | less than 1e-9 | 2.25401e-011 |

**Training accuracy and visualizing hidden layer :**

Cost at (fixed) debugging parameters (w/ lambda = 3.000000): 0.576051

**Training neural network :**

For different set of weight decay terms(λ) and number of iterations , accuracy level is clearly different i.e. the performance of the neural network varies. With the right learning settings, it is possible to get the neural network to perfectly fit the training set. But this process won’t be as accurate when predicting new set of data that it hasn’t seen before.

Epsilon is set to 0.12 for all cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | Lambda | Cost | Accuracy |
| 50 | 3 | 6.664829e-001 | 95.140000 |
| 400 | 3 | 5.671986e-001 | 97.540000 |
| 400 | 1 | 3.257341e-001 | 99.5 |

**Visualizing Neural Network :**

For λ = 3, iteration = 50 For λ = 3, iteration = 400 For λ = 1 iteration = 400

|  |  |  |
| --- | --- | --- |
|  |  |  |