CMPE 462 Homework 3

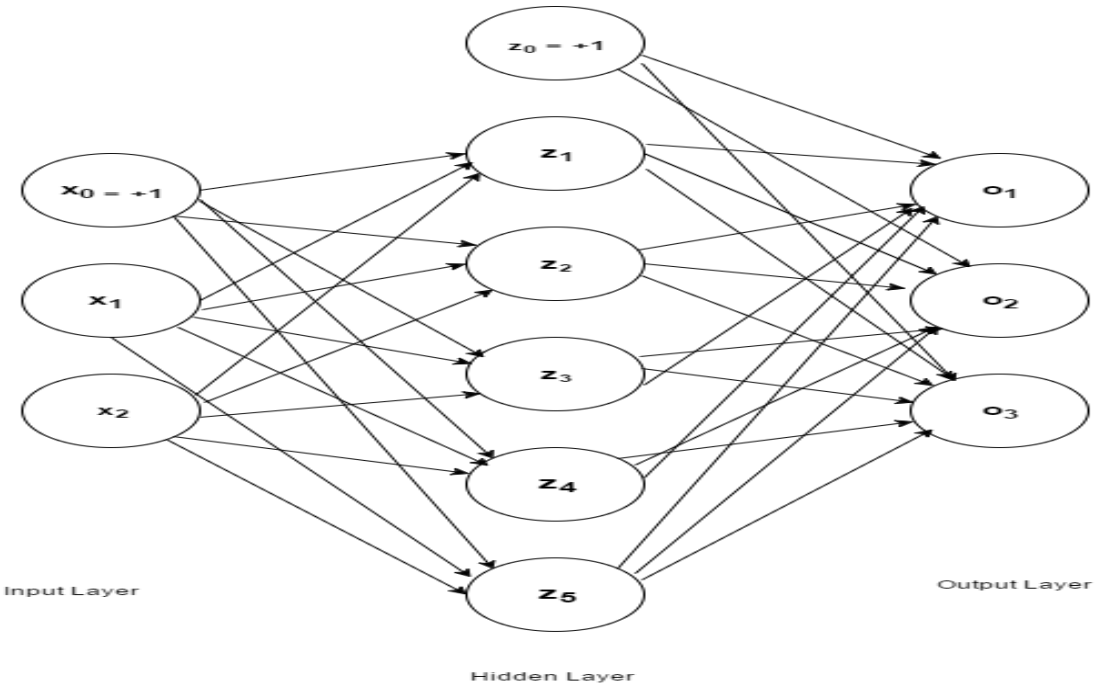
Enes Özipek
2012400162

April 26, 2017

Question 1

Since we are not working on many samples and also we are using only one hidden layer, I decided to split the data set as being 70% training set, 30% test set.

I used the following perceptron schema.



Input layer consists of X vector. x_0 is bias unit and $X_1 x_2$ are features.

Hidden layer consists of z vector. z_0 is bias unit and $z_{1...5}$ are calculated by using hyperbolic tangent function.

$$z = [1 \quad \tanh(x\theta_1)]$$

x is 1*3 vector while θ_1 is 3*5 matrix.

Output layer consists of 3 units. It is calculated as

$$o = z * \theta_2$$

z is 1*6 vector while θ_2 is 6*3 matrix.

PS: Since indexing starts with 1 in octave, I increment the class number by 1, that is, $\text{class}_{i=0,1,2}$ corresponds to $o_{i=1,2,3}$. I calculated probability for each class by considering 3 cases. For the most probable index, I declared its class by *index - 1*

Error function:

$$\frac{1}{2}\sum_t (r^t - o^t)^2$$

Question 2

This question was submitted as hard-copy.

Question 3

I divide the data set into test and training sets. Firstly, I started to train training set with a fixed learning rate. Then, after each epoch I updated the weights used between input-hidden layers and hidden-output layers.

Training

After each epoch, I saved the error rates for training and test sets.

I used the algorithm that we used in the class for the back propagation yet in a vectorized way. In other words,

Initialize all θ_1 and θ_2 to rand(-0.12,0.12)

Repeat For all $(x^t, r^t) \epsilon X$ in random order

$$z := tanh(x^t \theta_1^T)$$

$$o := z\theta_2^T$$

$$\Delta\theta_2 := \Delta\theta_2 + \eta z'(r^t - o^t)$$

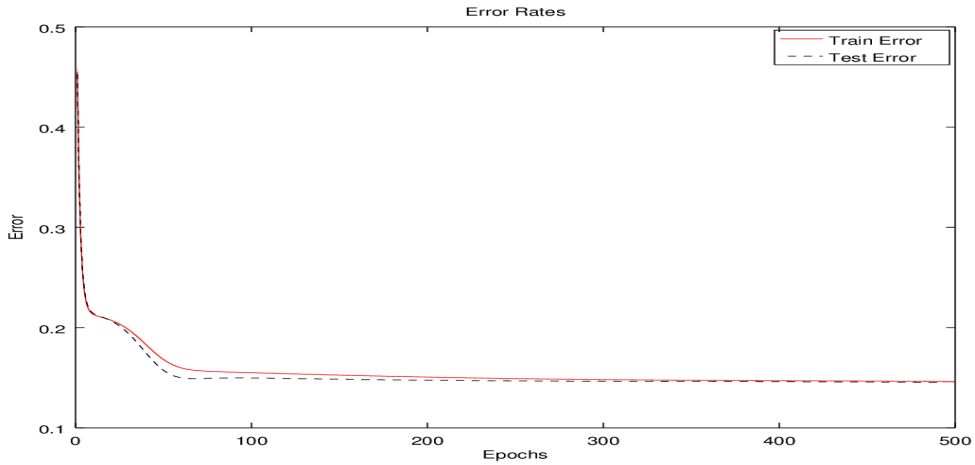
$$\Delta\theta_1 := \eta[(\sum(r^t - o^t)\theta_2)x^t]'(1 - z^2)$$

until convergence

Train & Test Error Plot

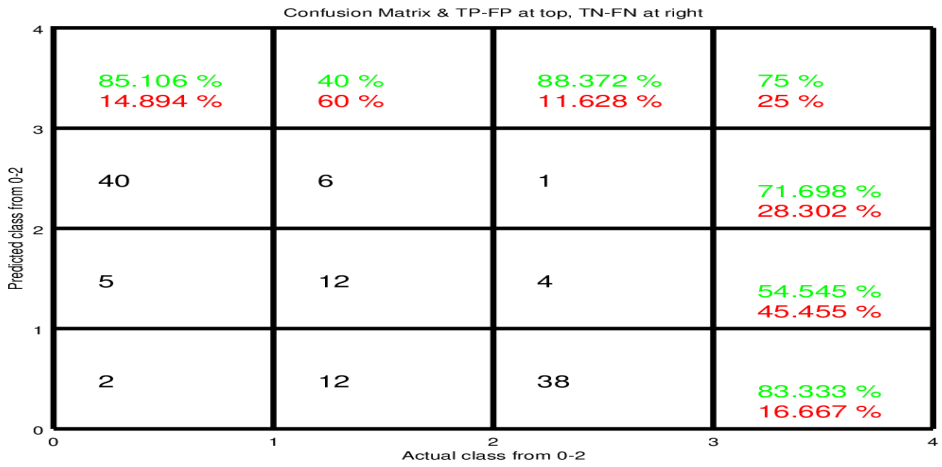
As I wrote error function before, I used mean square error to calculate error of each epoch. According to error and learning rate, I updated the weights between layers.

Following plot is obtained after training:



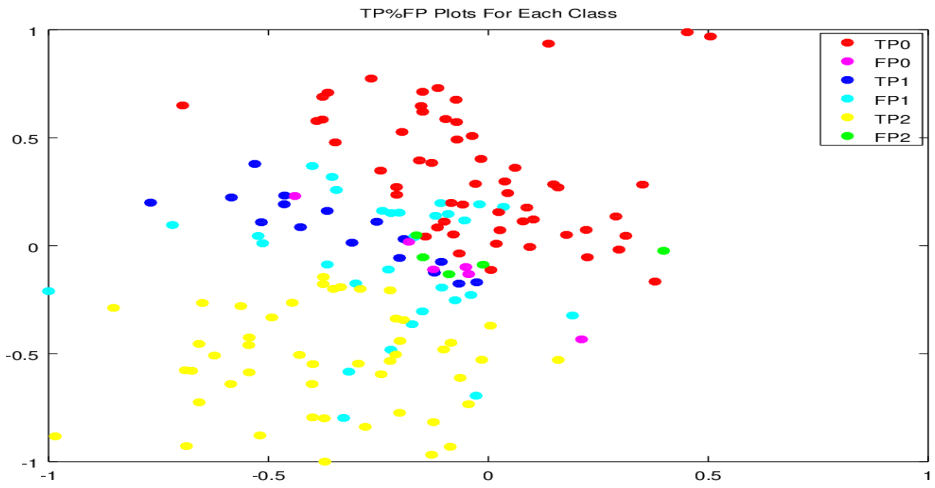
Testing

After plotting errors, I calculated the confusion matrix with the best Theta weights. Following matrix plot is my confusion matrix.



- Diagonals corresponds to True-Positive values. For instance 40 corresponds that my algorithm predicts samples belong to class 0 and, in truth, they are class 0.
- Prediction classes begins with class 0 and from row 2.
- First row contains values for True-Positive and False-Positive for each class.
- Last column, except first value, contains values for True-Negative and False-Negative for each class.
- Most right-top value denotes the rate for all test data

Plotting True-Positive & False-Positive



- For class 0 True-Positive values are in red and False-Positive values are in magenta
- For class 1 True-Positive values are in blue and False-Positive values are in cyan
- For class 2 True-Positive values are in yellow and False-Positive values are in green

To obtain this plot, I record class labels during the iteration of test set.

Bonus: Decision Boundaries

To obtain plot given below I followed steps itemized below:

- Generate random 20000 (x_1, x_2) points in range [-15,15].
- Run forward propagate with those points.
- Classify them with labels.
- Lastly, scatter plot them.