CIS 519: Introduction to Machine Learning
Homework 4

Archith Shivanagere Muralinath(PennID: 82629708, PennKey: archith)

PART 1: Problem Set

Problem 3: TANH Neural Networks

Problem (3a):

The advantage of using the tanh function instead of the sigmoid in a neural network is that the sigmoid function saturates to '0' whereas the TANH function saturates to +/- 1. Suppose during training the activity in neural network is close to '0', then the sigmoid function's gradient may go to 0.

Problem (3b):

Given

$$y_k(x,\theta) = \sigma\left(\sum_{j=1}^{M} \theta_{jk}^{(2)} \sigma\Big(\sum_{i=1}^{d} \theta_{ij}^{(1)} x_i + \theta_{0j}^{(1)}\Big) + \theta_{0k}^{(2)}\right)$$

Let us find the relation between $\sigma(a)$ and $tanh(a)$.

$$tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} = \frac{1 - e^{-2a}}{1 + e^{-2a}} = 2\sigma(2a) - 1$$
$$\sigma(a) = \frac{1}{2} tanh\Big(\frac{a}{2}\Big) + \frac{1}{2}$$

Rewriting $y_k(x,\theta)$ in terms of $tanh$ we get,

$$y_k(x,\theta) = \sigma\left(\sum_{j=1}^{M} \theta_{jk}^{(2)} \Big(\frac{1}{2} tanh\Big(\frac{1}{2}\sum_{i=1}^{d} \theta_{ij}^{(1)} x_i + \frac{1}{2}\theta_{0j}^{(1)}\Big) + \frac{1}{2}\Big) + \theta_{0k}^{(2)}\right)$$
$$y_k(x,\theta) = \sigma\left(\sum_{j=1}^{M} \frac{1}{2}\theta_{jk}^{(2)} tanh\Big(\sum_{i=1}^{d} \frac{1}{2}\theta_{ij}^{(1)} x_i + \frac{1}{2}\theta_{0j}^{(1)}\Big) + \sum_{j=1}^{m} \frac{1}{2}\theta_{jk}^{(2)} + \theta_{0k}^{(2)}\right)$$
$$y_k(x,\theta) = \sigma\left(\sum_{j=1}^{M} w_{jk}^{(2)} tanh\Big(\sum_{i=1}^{d} w_{ij}^{(1)} x_i + w_{0j}^{(1)}\Big) + w_{0k}^{(2)}\right)$$

where w can be obtained from linear transformation as shown below

$$w_{jk}^{(2)} = \frac{1}{2}\theta_{jk}^{(2)}$$

$$w_{ij}^{(1)} = \frac{1}{2}\theta_{ij}^{(1)}$$

$$w_{0j}^{(1)} = \frac{1}{2}\theta_{0j}^{(1)}$$

$$w_{0k}^{(2)} = \sum \frac{1}{2}\theta_{jk}^{(2)} + \theta_{0k}^{(2)}$$

Therefore, the parameters of the two networks differ by linear transformations.

PART 2: Programming Exercises

Problem 1: Text Classification and ROC

Training Values:

Metric	Naive Bayes	SVM with cosine kernel
Time	0:00:02.931874	0:00:00.771261
Accuracy	0.972231744943	0.960918752143
Precision	0.858908341916	0.880020597322
Recall	0.972231744943	0.960918752143

Testing Values:

Metric	Naive Bayes	SVM with cosine kernel
Accuracy	0.849347189234	0.811439857093
Precision	0.791498573934	0.824983579848
Recall	0.849347189234	0.811439857093

The classifier I found to be better is Naive Bayes.This is only true for the selected parameters.The assumption of independence in Naive Bayes classifier is satisfied by the variables of the dataset and the degree of class overlapping is small.

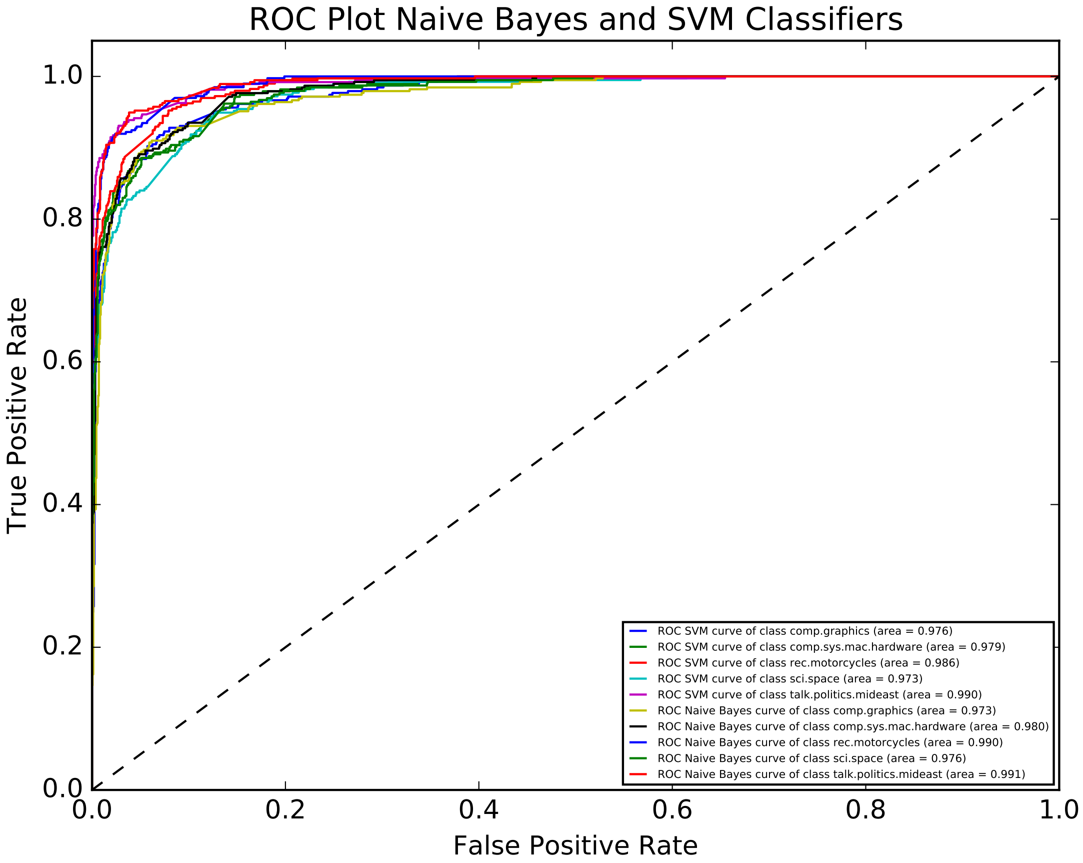
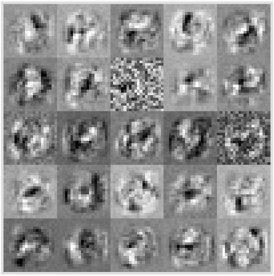


Figure 1: ROC Curves for Naive Bayes and SVM classifiers

Problem 2.5: Apply Your Neural Network to Digit Recognition

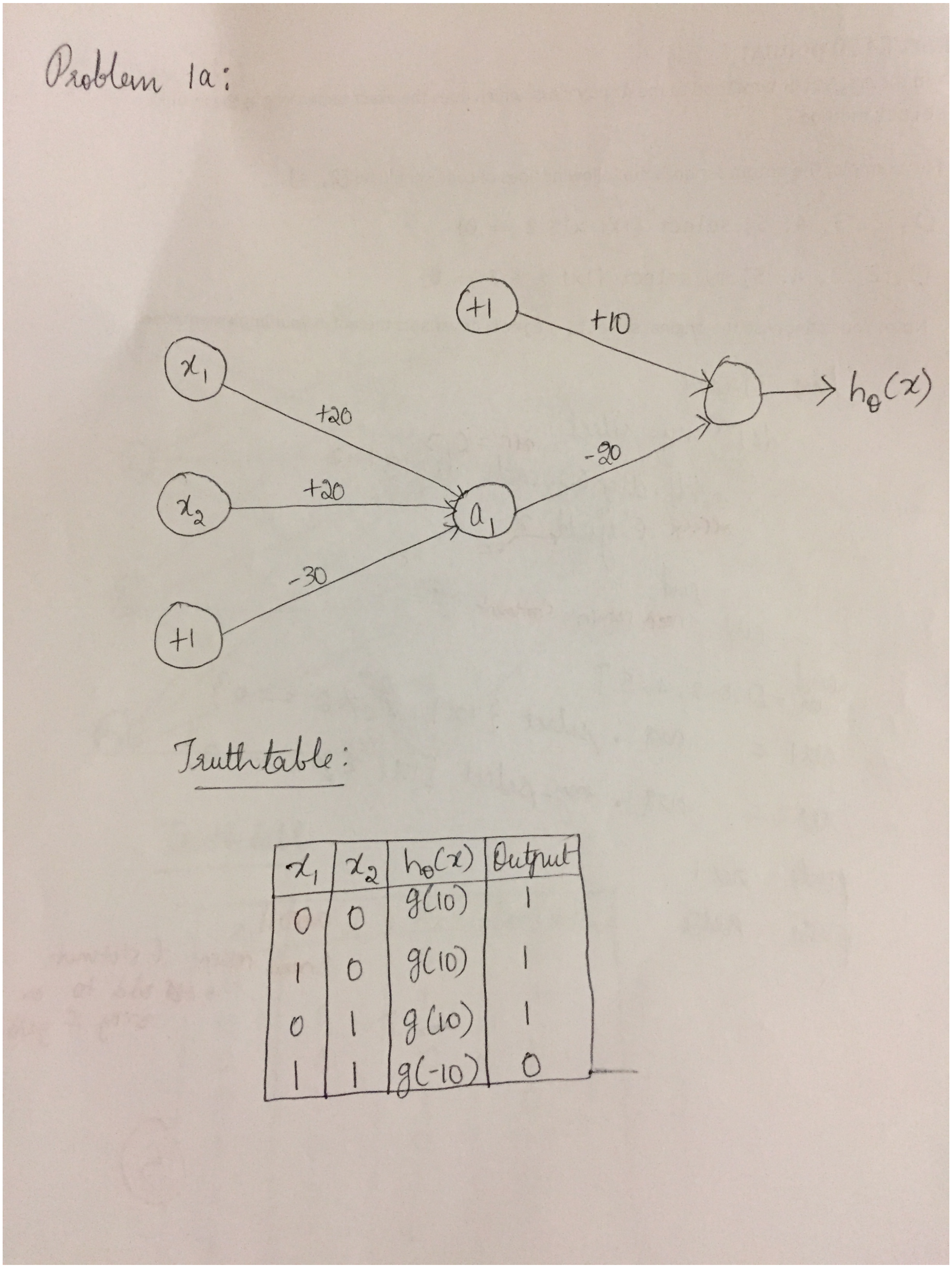
Optimal Learning Rate = 2.45
Regularization Parameter = 0.001
Maximum Training Performance = 0.957

Problem 2.6: Visualizing the Hidden Layers



PART 1: Problem Set

Problem 1: Logical Functions with Neural Networks

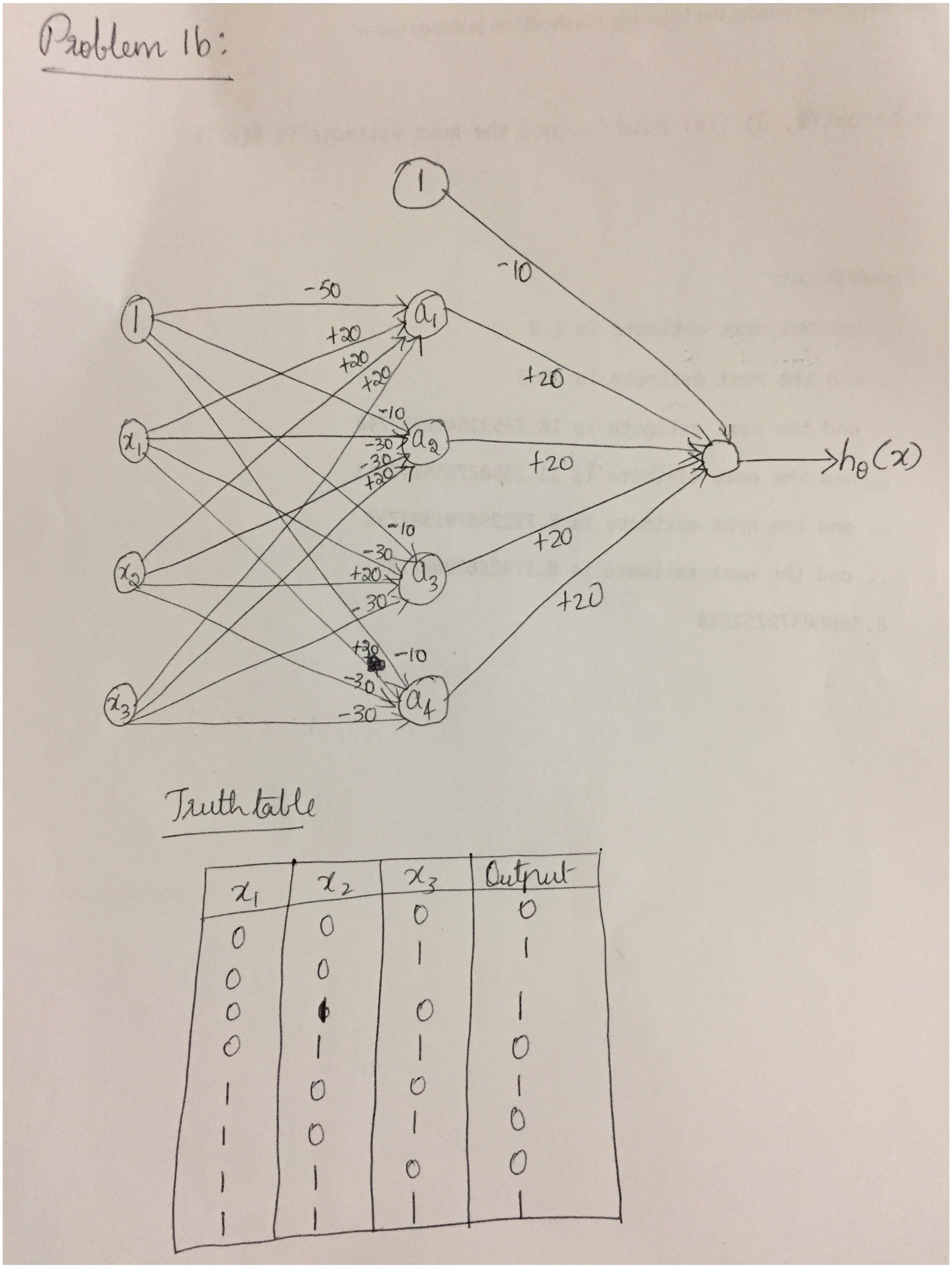


Problem 1a:

Truthtable:

x_1	x_2	$h_\theta(x)$	Output
0	0	g(10)	1
1	0	g(10)	1
0	1	g(10)	1
1	1	g(-10)	0

Figure 2: The NAND of two binary inputs



Problem 1b:

1
1
-10
-50
a_1
+20
+20
+20
+20
x_1
-10
-30
-30
+20
a_2
+20
$h_\theta(x)$
x_2
-10
-30
+20
-30
a_3
+20
+20
+20
-10
-30
-30
a_4
x_3

Truth table

x_1	x_2	x_3	Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 3: The parity of three binary inputs

Problem 2: Data Visualization and Feature Engineering

The two coordinate axes (features) along which you could plot the data to best visualize this dataset are Checking Status and Duration. These two were determined through the algorithms Maximum Corelation, Information Gain and best c val. By analyzing the scatter plot for these in the visualize tab, it is evident that these two are best features to visualize the dataset.