CS 4641 Homework 4

Budi Ryan — GTID: 903266310

1 Problem 1

(a) The proposed neural network for this function:

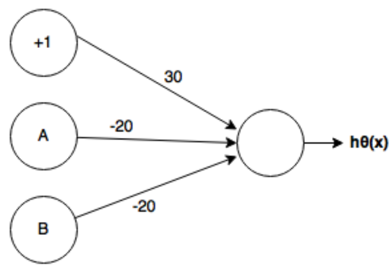


Figure 1: NAND NN

Not let's verify that the proposed neural network is correct. The formula for this neural network is:

$$h_\theta(X) = g(30 - 20A - 20B)$$

The truth table for the NAND function of 2 inputs:

A	B	$\neg(A \wedge B)$	$h_\theta(X)$
0	0	1	$g(30+0+0) = g(30) \approx 1$
0	1	1	$g(30-20+0) = g(10) \approx 1$
1	0	1	$g(30+0-20) = g(10) \approx 1$
1	1	0	$g(30-20-20) = g(-10) \approx 0$

Proofed!

(b) My logical function to solve the problem:

$$h_\theta(X) = (A \oplus B) \oplus C$$

Based on the above equation, my proposed neural network for this function is:

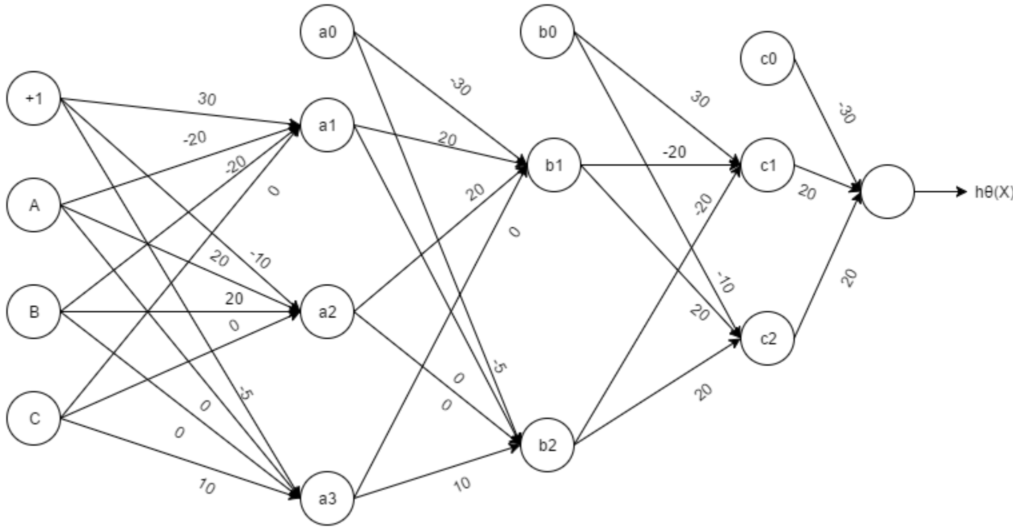


Figure 2: 3-bit Parity NN

The truth table for the even parity function of 3 inputs (The formula is only for the last layer):

A	B	C	$(A \oplus B) \oplus C$	$h_\theta(X)$
0	0	0	0	$g(-30) \approx 0$
0	0	1	1	$g(10) \approx 1$
0	1	0	1	$g(10) \approx 1$
0	1	1	0	$g(-10) \approx 0$
1	0	0	1	$g(10) \approx 1$
1	0	1	0	$g(-10) \approx 0$
1	1	0	0	$g(-10) \approx 0$
1	1	1	1	$g(10) \approx 1$

Proofed!

2 Problem 2

In this problem I will provide all calculation step by step until I calculated the weights after first 2 epochs:

$\theta = [0.1, 0.1, 0.1], [0.1, 0.1]$

FIRST EPOCH:

First training set:

$a_1 = [1, 1, 0]$

$a_2 = g(0.1 \times 1 + 0.1 \times 0 + 0.1 \times 1) = g(0.2) = 0.549$

$a_3 = g(0.1 \times 1 + 0.1 \times 0.549) = g(0.1549) = 0.538$

$\delta_3 = a_3 - y = 0.538 - 1 = -0.461$

$\delta_2 = \theta^T \times \delta_3. * g'(z_2)$

$\delta_2 = -0.0113$

$\Delta^{11} = a_1\delta_2 = [-0.0113, -0.0113, 0]$

$\Delta^{12} = a_2\delta_3 = [-0.461, -0.253]$

Second training set:

$a_1 = [1, 0, 1]$

$a_2 = g(0.1 \times 1 + 0.1 \times 0 + 0.1 \times 1) = g(0.2) = 0.549$

$a_3 = g(0.1 \times 1 + 0.1 \times 0.549) = g(0.1549) = 0.538$

$\delta_3 = a_3 - y = 0.538 - 0 = 0.538$

$\delta_2 = \theta^T \times \delta_3. * g'(z_2)$

$\delta_2 = -0.0113$

$\Delta^{21} = a_1\delta_2 = [0.0133, 0, 0.0133]$

$\Delta^{22} = a_2\delta_3 = [0.538, 0.295]$

Combining both Deltas from both training set:

$\Delta^1 = \Delta^{11} + \Delta^{21} = [0.002, -0.0113, 0.0133]$

$\Delta^2 = \Delta^{12} + \Delta^{22} = [0.077, 0.0424]$

$D_1 = 0.5 \times \Delta^1 = [0.001, 0.00565, 0.00665]$

$D_2 = 0.5 \times \Delta^2 = [0.0385, 0.0212]$

Update to the weight after first epoch:

$\theta_1 = [0.1, 0.1, 0.1] - 0.3[0.001, -0.0056, 0.0066] = [0.099, 0.1016, 0.098]$

$\theta_2 = [0.1, 0.1] - 0.3[0.038, 0.021] = [0.884, 0.093]$

SECOND EPOCH:

First training set:

$a_1 = [1, 1, 0]$

$a_2 = g(0.0997 \times 1 + 0.101695 \times 1) = g(0.2013) = 0.55017$

$a_3 = g(0.08845 \times 1 + 0.0936 \times 0.55017) = g(0.1399) = 0.5349$

$\delta_3 = a_3 - y = 0.5349 - 1 = -0.46506$

$\delta_2 = \theta^T \times \delta_3. * g'(z_2)$

$\delta_2 = -0.01077$

$\Delta^{11} = a_1\delta_2 = [-0.01077, -0.01077, 0]$

$\Delta^{12} = a_2\delta_3 = [-0.46506, -0.25586]$

Second training set:

$a_1 = [1, 0, 1]$

$a_2 = g(0.0997 \times 1 + 0.098 \times 1) = g(0.1977) = 0.54926$

$a_3 = g(0.08845 \times 1 + 0.0936 \times 0.549) = g(0.13988) = 0.534$

$\delta_3 = a_3 - y = 0.534 - 0 = 0.534$

$\delta_2 = \theta^T \times \delta_3. * g'(z_2)$

$\delta_2 = 0.0124$

$\Delta^{21} = a_1\delta_2 = [0.0124, 0, 0.0124]$

$\Delta^{22} = a_2\delta_3 = [0.534, 0.2938]$

Combining both Deltas from both training set:

$\Delta^1 = \Delta^{11} + \Delta^{21} = [0.00216307, -0.01077, 0.0124]$

$\Delta^2 = \Delta^{12} + \Delta^{22} = [0.06985, 0.0379]$

$D_1 = 0.5 \times \Delta^1 = [0.000815, -0.00538, 0.0062]$

$D_2 = 0.5 \times \Delta^2 = [0.0349, 0.0189]$

Update to the weight after first epoch:

$\theta_1 = \theta_1 - 0.3[0.000815, -0.00538, 0.0062] + 0.9[0.001, -0.00565, 0.00665] = [0.10035, 0.0982, 0.1021]$

$\theta_2 = \theta_2 - 0.3[0.0349, 0.0189] + 0.9[0.0335, 0.021] = [0.11262, 0.107]$

Theta after first epoch:

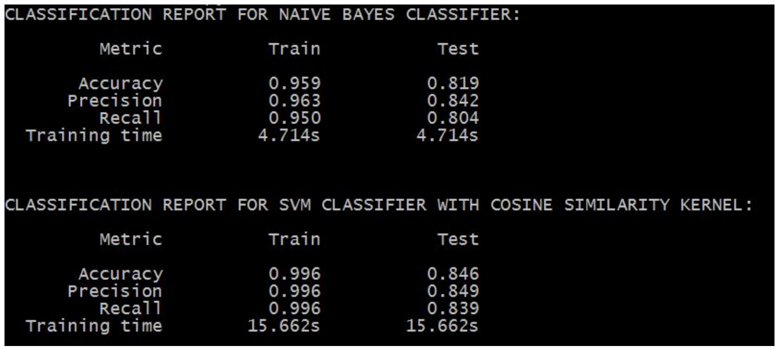
$$\theta = [0.1016, 0.098, 0.099, 0.093, 0.884]$$

Theta after second epoch:

$$\theta = [0.0982, 0.1021, 0.10035, 0.107, 0.11262]$$

3 Problem 3 (Text Classification Problem)

Now let's see the head-to-head performance comparison(the values here are all out of 1) between Naive Bayes and SVM with cosine similarity kernel:
(Note that for Naive Bayes, I used CountVectorizer combined with TF-IDF feature vectors, lowercased all terms and removed stop words, used log-scaled term counts for TF and inversed the document frequency over the training data only):


CLASSIFICATION REPORT FOR NAIVE BAYES CLASSIFIER:

Metric	Train	Test
Accuracy	0.959	0.819
Precision	0.963	0.842
Recall	0.950	0.804
Training time	4.714s	4.714s

CLASSIFICATION REPORT FOR SVM CLASSIFIER WITH COSINE SIMILARITY KERNEL:

Metric	Train	Test
Accuracy	0.996	0.846
Precision	0.996	0.849
Recall	0.996	0.839
Training time	15.662s	15.662s

Figure 3: ML showdown!

Analysis:
From the table I will say SVM is better than NB. Although it is true that SVM has a significantly longer training time compared to NB, the testing accuracy of SVM is higher than NB's.

4 Problem 4 (Neural Network + Challenge)

Here are the optimal parameters I got after running and tuning the Neural Network:

(1) Training performance: 0.9836 (out of 1)

(2) Regularization parameter: 0

(3) Learning rate: 2

(4) NumEpoch: 1500

Here is my visualization of the hidden layer with the specified number of epoch:

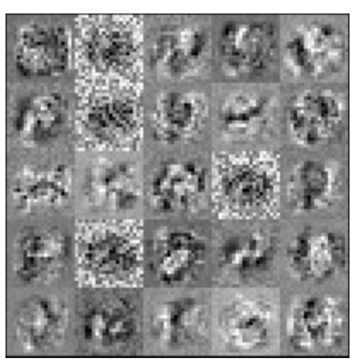


Figure 4: Hidden layer visualization