Homework 2

● Graded

Student

TIYA CHOKHANI

Total Points

55 / 55 pts

Question 1

Perceptron **15** / 15 pts

1.1 **(a)** **5** / 5 pts

✔ **– 0 pts** Correct

– 2.5 pts partially correct

– 5 pts Incorrect or did not attempt

1.2 **(b)** **5** / 5 pts

✔ **– 0 pts** Correct

– 3 pts Incorrect w

– 2 pts Incorrect prediction

– 5 pts Incorrect or did not attempt

– 1 pt Minor mistake on computation of prediction

1.3 **(c)** **5** / 5 pts

✔ **– 0 pts** Correct

– 2.5 pts Incomplete/explanation partly makes sense (missing log reg/perceptron)

– 5 pts Incorrect or did not attempt

Question 2

Neural Network **20** / 20 pts

2.1 (a) **4** / 4 pts

✔ **− 0 pts** Correct

− 2 pts No mention of specific activation functions

− 4 pts Incorrect or did not attempt

− 2 pts no explanation

− 2 pts Missing output layer part

2.2 (b) **3** / 3 pts

✔ **− 0 pts** Correct

− 1 pt Correct formulas with minor computation error

− 2 pts Some incorrect formulas

− 3 pts Incorrect or did not attempt

2.3 (c) **4** / 4 pts

✔ **− 0 pts** Correct

− 0.5 pts Sign error

− 1 pt Correct formula for loss with minor computation error, or wrong log base.

− 2 pts Incorrect / missing loss

− 1 pt Correct formula for derivative with minor computation error.

− 1 pt Partially incorrect formula for derivative

− 2 pts Incorrect / missing derivative

− 1 pt Incorrect answers due to wrong $\hat{y}$

− 4 pts Wrong / Did not attempt

2.4 **(d)** **6** / 6 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Missing/wrong final result for bias term
- **− 1 pt** Correct formula from chain rule, with one incorrect partial derivative or minor computation error.
- **− 2 pts** Correct formula from chain rule, with two incorrect partial derivatives.
- **− 3 pts** Correct formula from chain rule, with three incorrect partial derivative or minor computation error.
- **− 4 pts** Correct formula from chain rule but did not present the final result/no computation.
- **− 6 pts** Incorrect or did not attempt
- **− 5 pts** incorrect formula or missing formula
- **− 2.5 pts** partially correct formula

2.5 **(e)** **3** / 3 pts

- ✔ **− 0 pts** Correct
- **− 1 pt** Missing/extra bias term
- **− 3 pts** Incorrect / Did not attempt

Question 3

Multi-class Classification **10** / 10 pts

3.1 **(a)** **5** / 5 pts

- ✔ **− 0 pts** Correct
- **− 2 pts** Missing bias term
- **− 2 pts** Other minor mistake
- **− 5 pts** Incorrect or did not attempt

3.2 **(b)** **5** / 5 pts

- ✔ **− 0 pts** Correct
- **− 2.5 pts** Incorrect OvR
- **− 2.5 pts** Incorrect multinomial
- **− 5 pts** Incorrect or did not attempt

Question 4

Decision Boundary **10** / 10 pts

4.1 Neural Network (1 hidden layer with 10 ReLU) **5** / 5 pts

✔ **− 0 pts** Correct

− 1.5 pts No Explanation

− 2.5 pts Incorrect or missing answer

4.2 Neural Network (1 hidden layer with 10 tanh units) **5** / 5 pts

✔ **− 0 pts** Correct

− 1.5 pts No explanation

− 2.5 pts Incorrect or missing answer

Questions assigned to the following page: 1.1, 2.1, 1.3, and 1.2

Homework 2 Solutions
CS M148: Introduction to Data Science

Tiya Chokhani

March 10, 2025

Question 1

(a) After one epoch, the weight vector is

$$w = y_1x_1 + y_2x_2 + y_4x_4 + y_5x_5.$$

This indicates that the perceptron made an update for examples 1, 2, 4, and 5 (i.e., these examples were misclassified) and that example 3 was correctly classified since it did not contribute to the update.

(b) With $d = 3$ and the data

i	$[1,\, x_{i1},\, x_{i2}]$	y_i
1	$[1,\, 1,\, 0]$	$+1$
2	$[1,\, 2,\, -1]$	$+1$
3	$[1,\, 2,\, -3]$	-1
4	$[1,\, 3,\, -1]$	$+1$
5	$[1,\, 1,\, -1]$	$+1$,

the weight vector is updated only for examples 1, 2, 4, and 5. Hence,

$$w = (+1)[1,\, 1,\, 0] + (+1)[1,\, 2,\, -1] + (+1)[1,\, 3,\, -1] + (+1)[1,\, 1,\, -1] = [4,\, 7,\, -3].$$

For $x_2 = [1,\, 2,\, -1]$,

$$w^\top x_2 = 4 \cdot 1 + 7 \cdot 2 + (-3)(-1) = 4 + 14 + 3 = 21,$$

so the prediction is $+1$.

(c) If the data is not linearly separable, the perceptron algorithm will never converge because it keeps updating its weights indefinitely as there will always be misclassified points. In contrast, logistic regression minimizes a convex loss (such as binary cross-entropy), ensuring convergence and yielding probability outputs even when the classes overlap.

Question 2

(a) Hidden Layers: We can use any activation functions. We could use ReLU, ELU, Leaky ReLU and variants of these models. Activation functions in the hidden layers are meant to make our model sparse and address the gradient vanish or exploding issues. Output Layers: sigmoid, softmax, or tanh activations to generate probabilities of the input being in each class makes them suitable

Questions assigned to the following page: 2.2, 2.4, 2.1, and 2.3

options. We need specific activation functions to map the output of our neural network to the desired format.

(b) Consider the network with two inputs, two hidden neurons, and one output neuron. The given values are:

$$\begin{aligned} X_1 = 3.1, \quad X_2 = -9.8,& \\ \text{Hidden Neuron 1 (Sigmoid):} \quad & W_{11} = -0.8,\ W_{12} = -0.1,\ \text{bias} = 0, \\ \text{Hidden Neuron 2 (ReLU):} \quad & W_{21} = 3.8,\ W_{22} = 0.8,\ \text{bias} = 0, \\ \text{Output Neuron (Linear):} \quad & W_{31} = -2,\ W_{32} = 0.2,\ \text{bias} = 0. \end{aligned}$$

For Neuron 1:

$$\text{net}_1 = -0.8(3.1) - 0.1(-9.8) \approx -2.48 + 0.98 = -1.50,$$

and applying the sigmoid yields

$$a_1 \approx \frac{1}{1 + e^{1.50}} \approx 0.18.$$

For Neuron 2:

$$\text{net}_2 = 3.8(3.1) + 0.8(-9.8) \approx 11.78 - 7.84 = 3.94,$$

and with ReLU, $a_2 = 3.94$. The output neuron computes:

$$\text{net}_3 = -2(0.18) + 0.2(3.94) \approx -0.36 + 0.788 \approx 0.428,$$

so the output is approximately 0.43.

(c) The binary cross-entropy loss is defined as

$$L = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right].$$

For $y = 0$ and $\hat{y} = 0.43$,

$$L \approx -\log(0.57) \approx 0.562.$$

The derivative is:

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}} \approx \frac{1}{0.57} \approx 1.754.$$

(d) To compute $\frac{\partial L}{\partial W_{12}}$, we use the chain rule:

$$\frac{\partial L}{\partial W_{12}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial \text{net}_1} \cdot \frac{\partial \text{net}_1}{\partial W_{12}}.$$

Here, $\frac{\partial \hat{y}}{\partial a_1} = -2$ (since the output is a linear combination with weight -2), $\frac{\partial a_1}{\partial \text{net}_1} = a_1(1 - a_1) \approx 0.18 \times 0.82 \approx 0.1476$, and $\frac{\partial \text{net}_1}{\partial W_{12}} = X_2 = -9.8$. Thus,

$$\frac{\partial L}{\partial W_{12}} \approx 1.754 \times (-2) \times 0.1476 \times (-9.8) \approx 5.1.$$

Since the derivative with respect to the bias is negative, the bias should be increased during gradient descent.

Questions assigned to the following page: 2.5, 3.1, 3.2, 4.1, and 4.2

(e) In this network, the hidden layer has 2 neurons. Each neuron has 2 weights and 1 bias (total 3 parameters per neuron), and the output layer has 2 weights and 1 bias (3 parameters). The total number of parameters is:

$$2 \times 3 + 3 = 9.$$

Question 3

(a) For a multi-class problem with 4 classes and 25 features using One-vs-Rest logistic regression, each classifier has 25 weights and 1 bias, i.e., 26 parameters per classifier. For 4 classes, the total number of parameters is:

$$4 \times 26 = 104.$$

(b)

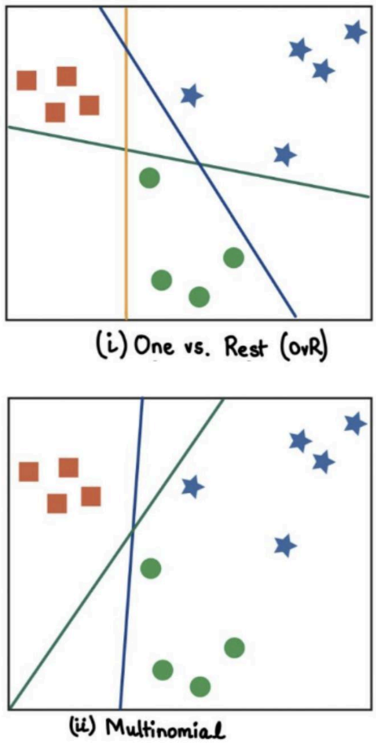

Figure 1:

Question 4

1. (b) The decision boundary for a network with 1 hidden layer of 10 ReLU units is piecewise linear with distinct segments because ReLU is linear in its active region.

Question assigned to the following page: 4.2

2. (a) In contrast, the network with 10 tanh units produces a smooth, continuously curved decision boundary due to the smooth, non-linear nature of the tanh activation.