CS M148: Introduction to Data Science
Released: Feb. 28th, 2025

Homework 2

Submit to Gradescope
Due: March 10th, 2PM PST

1 Perceptron [15 points]

Consider training a Perceptron model $y = \text{sgn}(\mathbf{w}^\top \mathbf{x})$, $\mathbf{w} \in \mathbb{R}^d$ on a dataset $D = \{(\mathbf{x}_i, y_i)\}, i = 1 \dots 5$. Both $\mathbf{w}$ and $\mathbf{x}_i$ are vectors of dimension d, and $y \in \{+1, -1\}$ is binary. Assume that the bias term is already augmented in $\mathbf{x}_i$: $\mathbf{x}_i = [1, x_1, \dots, x_{d-1}]$. The activation function is a sign function where $\text{sgn}(x) = 1$ for all $x > 0$ and $\text{sgn}(x) = -1$ otherwise. The Perceptron algorithm is given below,

Algorithm 1 Perceptron

```
Initialize $\mathbf{w} = \mathbf{0}$
for i = 1 ... N do
    if $y_i \neq \text{sgn}(\mathbf{w}^\top \mathbf{x}_i)$ then
        $\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$
    end if
end for
return w
```

(a) **(5 points)** The Perceptron model is trained for 1 epoch, i.e. iterated over the entire dataset once, and the weight vector after this training epoch, $\mathbf{w}$, is expressed as $\mathbf{w} = y_1\mathbf{x}_1 + y_2\mathbf{x}_2 + y_4\mathbf{x}_4 + y_5\mathbf{x}_5$. What can we infer about the training process from this?

Solution: x_1, x_2, x_4, x_5 are all misclassified.

(b) **(5 points)** Let $d = 3$ and the data points be given as follows

i	$\mathbf{x}_{i,1}$	$\mathbf{x}_{i,2}$	$\mathbf{x}_{i,3}$	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

Following the formulation of your answer in (a), what is of $\mathbf{w}$ given the values of the data points? Express with a vector of numbers this time. Furthermore, if we iterate through the dataset again, what will be the model's prediction on x_2?

Solution: $\mathbf{w} = [4, 7, -3]$. It will not make mistake again and will predict $+1$.

(c) **(5 points)** How does the Perceptron handle cases where the data is not linearly separable? How does this compare to logistic regression?

Solution: The Perceptron cannot handle non-linearly separable data because it only updates weights based on misclassified examples and does not minimize any continuous loss. If the data is not separable, it will oscillate or fail to converge. Logistic regression uses probabilistic approach, so the loss can converge even when the data is not separable.

.

2 Neural Networks [15 points]

(a) **(4 points)** Refer to Lecture 13 for the activation functions of neural networks. Considering a binary classification problem, what are possible activations choices for the hidden and output layers repectively? Explain why.

Solution: Hidden layers: Any activations are ok. ReLU, Leaky ReLU, ELU and variants are popular choices. 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. We need specific activation functions to map the output of our neural network to the desired format.

(b) **(3 points)** Consider a binary classification problem where $y \in \{0, 1\}$. We consider the neural network in Figure 2 with 2 inputs, 2 hidden neurons, and 1 output. We let neuron 1 use **Sigmoid** activation and neuron 2 use **ReLU** activation, respectively. The other layers use the linear activation function. Suppose we have an input $X_1 = 3.1$ and $X_2 = -9.8$, with label $y = 0$. And the weights are initialized as $W_{11} = -0.8, W_{12} = -0.1, W_{21} = 3.8, W_{22} = 0.8, W_{31} = -2, W_{32} = 0.2$, and the bias term W_{10}, W_{20}, W_{30} are all initialized to be 0. Compute the output of the network rounded to two decimal places.

Solution:
$z_1 = W_1^T X = W_{11}X_1 + W_{12}X_2 + W_{10} = 1.2 \times 3.1 + 0.3 \times (-9.8) + 0 = -2.48 + 0.98 = -1.5$
$h_1 = \text{Sigmoid}(z_1) = 0.182425$
$z_2 = W_2^T X = W_{21}X_2 + W_{22}X_2 + W_{20} = (3.8) \times 3.1 + (0.8) \times (-9.8) + 0 = 11.78 - 7.84 = 3.94$
$h_2 = \text{ReLU}(z_2) = 3.94$
$\hat{y} = W_{31}h_1 + W_{32}h_2 + W_{30} = (-2) \times 0.182425 + 0.2 \times 3.94 = -0.36485 + 0.788 \approx 0.42.$

(c) **(4 points)** We consider the binary cross entropy loss function. What is the loss of the network on the given data point in (b)? What is $\frac{\partial \mathcal{L}}{\partial \hat{y}}$? Report both rounded to three decimal places. (**Hint.** Refer to the lecture slides for defining a binary cross-entropy loss function. For simplicity, use your rounded answer from part b.)

Solution:
$\mathcal{L} = -y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y}) = -1 \times ln(1 - 0.42) = -1 \times ln(0.58) = 0.544.$
$\frac{\partial \mathcal{L}}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}}(-y \ln(\hat{y}) - (1 - y) \ln(1 - \hat{y})) = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}} = \frac{1}{0.58} = 1.724.$

(d) **(6 points)** We now consider the backward pass. Given the same initialized weights and input as in (b), write the formula and calculate the derivative of the loss w.r.t W_{12} to the nearest decimal point, i.e. $\frac{\partial \mathcal{L}}{\partial W_{12}}$. Should the bias at Neuron 1 be increased or decreased?

Solution:
$\frac{\partial \mathcal{L}}{\partial W_{12}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_{12}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial W_{12}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} W_{31} \cdot \sigma(z_1) \cdot (1 - \sigma(z_1)) \cdot X_2 = 1.724 \times -2 \times 0.149146 \times -9.8 = 5.0.$
The gradient of the bias can be calculated as:
$\frac{\partial \mathcal{L}}{\partial W_{10}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_{10}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial W_{10}} = \frac{\partial \mathcal{L}}{\partial \hat{y}} W_{31} \cdot \sigma(z_1) \cdot (1 - \sigma(z_1)) \cdot 1 = 1.724 \times -2 \times 0.149146 \times 1 = -0.514.$
We can save time by reusing the intermediate values we got while calculating $\frac{\partial \mathcal{L}}{\partial W_{12}}$.
The gradient at the bias is negative, so we should increase it.

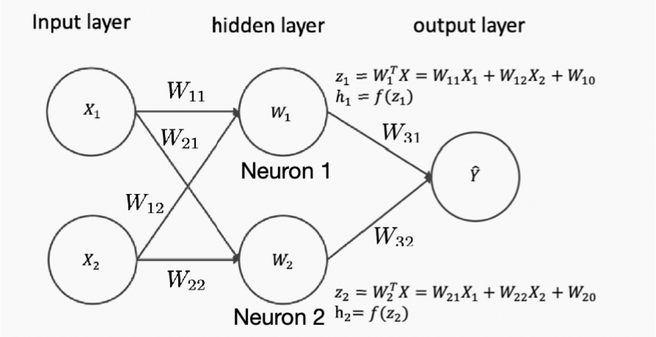


Figure 1: Neural Network

(e) **(4 points)** Given the neural network as in (b), how many parameters does the network have? (**Hint**. Each weight unit counts as a parameter, and we also consider the bias terms $(W_{10}, \ldots)$ as parameters.)

Solution: $3 \times 2 + 3 = 9$ parameters

3 Multi-class Classification [10 points]

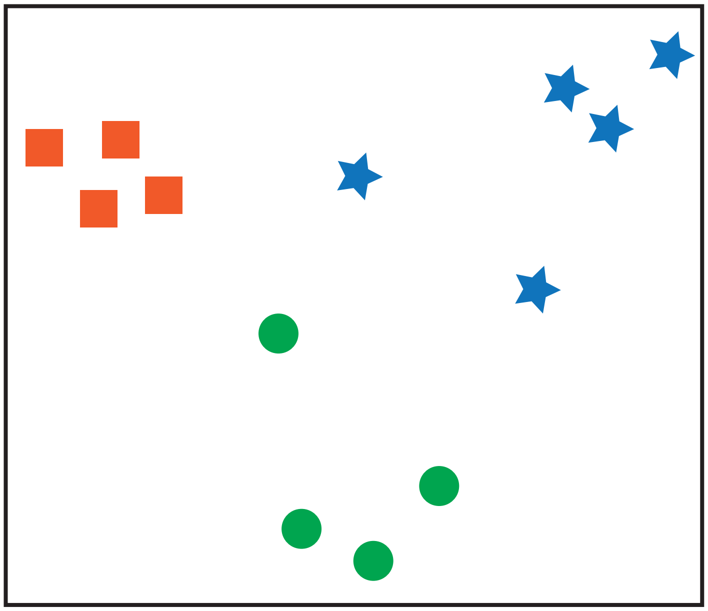


Figure 2: Multiclass Logistic Regression

(a) **(5 points)** Consider a multi-class classification problem with 4 classes and 25 features. We will use the logistic regression model to first build our binary classifier. Then, what will be the total number of parameters for using **One vs. Rest (ovr)** strategies for the multi-class classification task using logistic regression? (**Hint**. Refer to lecture 11 for multi-class logistic regression model.)

Solution: For each classes we need to train a classification model, and each model will have $25 + 1$ parameters, so in total is $4 \times 26 = 104$

(b) **(5 points)** Consider a new multi-class classification problem with 3 classes. The distribution of the points is shown in the figure (Square - Class 1, Circle - Class 2, Star - Class 3). Draw the linear classifiers used for classifying the three classes, using (i) One vs. Rest (OvR), and (ii) Multinomial approaches.

Solution: 3 lines for OvR where each line separates a class with all others. 2 lines for Multinomial, where each line separates a class with the reference class; any reference class is fine.

4 Decision Boundary [10 points]

Consider the classification problems with two classes, which are illustrated by circles and crosses in the plots below. In each of the plots, one of the following classification methods has been used, and the resulting decision boundary is shown:

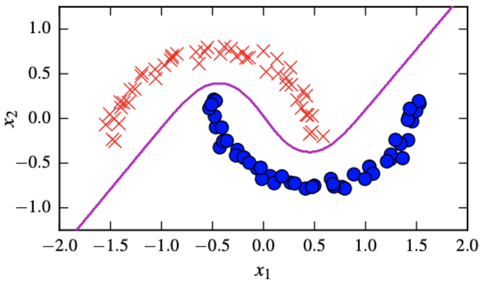
(1) Neural Network (1 hidden layer with 10 ReLU)

Solution: (b) It should be piecewise linear due to the ReLU activiation function

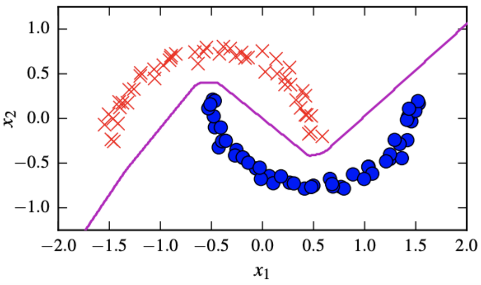
(2) Neural Network (1 hidden layer with 10 tanh units)

Solution: (a) It should be curved due to the tanh activiation function

Assign each of the previous methods to exactly one of the following plots (in a one to one correspondence) by annotating the plots with the respective letters, and **explain briefly** why did you make each assignment.



(a)



(b)