Written Assignment 5

Quiz 5

Question 1 Not yet answered Marked out of 4.00 Flag question

Given a set of raw scores (logits) z = (2.0, 1.0, 0.1) compute the **Softmax probabilities** for each class using the Softmax function. Based on your calculations select the correct answer by considering only the first 3 decimals.

Select one:

- a. (0.551, 0.334, 0.128)
- b. None of the alternatives.
- c. (0.659, 0.242, 0.098)
- d. (0.712, 0.213, 0.081)

ΡS33 ΜΗΔ5
p. 95 (ή 25)
διαφάνεια

$$z = [2, 1, 0.1] \qquad \text{Softmax}(z) = \frac{e^{z}}{\sum_j e^{z_j}}$$

$- \max z_i$

$$z' = [2-2, 1-2, 0.1-2] = [0, -1, -1.9]$$

$$e^{0} = 1,\ e^{-1} = \frac{1}{e} \simeq 0.3679,\ e^{-1.9} \simeq 0.1496 \qquad \sum_3 e^{z_3} = 1 + 0.3679 + 0.1496 = 1.5175$$

$$\text{Softmax}_1 = \frac{1}{1.5175} \simeq 0.659$$

$$\text{Softmax}_2 = \frac{0.3679}{1.5175} \simeq 0.242$$

$$\text{Softmax}_3 = \frac{0.1496}{1.5175} \simeq 0.098$$

Question 2 Not yet answered Marked out of 4.00 Flag question

Using a linear classifier specified by $w = (-1, 3)$ and $b = 1$, select the correct option that answers the following questions. Consider that points are in the form $x = (x_1, x_2)$ and that the classification is given by the sign of $f(x) = w \cdot x + b$

1. At what point does the decision boundary intersect with the x-axis?
2. At what point does the decision boundary intersect with the y-axis?
3. What is the label assigned to the data point 1: $p_1 = (-1, 0.5)$?
4. What is the label assigned to the data point 2: $p_2 = (2, -4)$?

Select one:

- a. x_1 = 1.0, x_2 = -0.33, +1, -1
- b. x_1 = 2.0, x_2 = 0.1, -1, -1
- c. x_1 = 1.0, x_2 = 0.33, +1, +1
- d. None of the alternatives.

$w = (-1, 3)$, $x = (x_1, x_2)$

$b = 1$

$f(x) = w \cdot x + b = -x_1 + 3x_2 + 1$

1. $f(x) = 0 \Rightarrow -x_1 + 3x_2 + 1 = 0 \overset{x_2=0}{\Longrightarrow} -x_1 + 1 = 0 \Rightarrow \boxed{x_1 = 1}$

2. $f(x) = 0 \Rightarrow -x_1 + 3x_2 + 1 = 0 \overset{x_1=0}{\Longrightarrow} 3x_2 + 1 = 0 \Rightarrow x_2 = -1/3 = \boxed{-0.33}$

3. $f(p_1) = 1 + 3 \times 0.5 + 1 = 3.5 \overset{f(p_1)>0}{\Longrightarrow} \boxed{+1}$

4. $f(p_2) = -2 + 3 \times (-4) + 1 = -13 \overset{f(p_2)<0}{\Longrightarrow} \boxed{-1}$

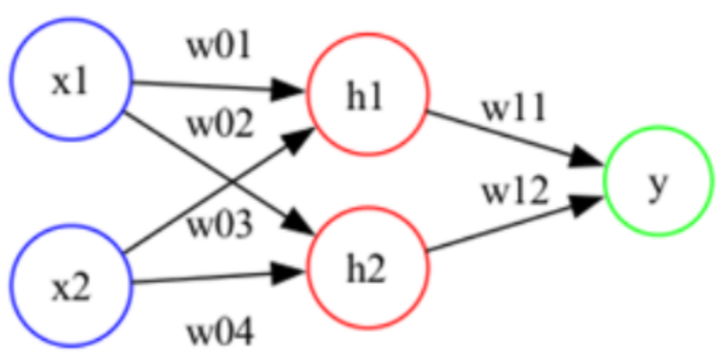
Question 3

Not yet answered   Marked out of 4.00   Flag question

You are given a Neural Network with one hidden layer and a single output node depicted in the following figure with linear activation functions for both the hidden layer and sigmoid activation function for the output layer. The initial weights are given in the table below, the input $p = (x_1, x_2) = (0.5, 1.5), y = 1$. The cost function used is the Binary Cross Entropy loss $L(y, \hat{y}) = -[y \cdot log(\hat{y}) + (1 - y) \cdot log(1 - \hat{y})]$, where the natural logarithm is used.

Given a single forward pass select the option that correctly answers the following: *network output; BCE loss value*

Initial weights					
w01	w02	w03	w04	w11	w12
0.5	-0.3	0.8	0.2	0.7	-0.5



Select one:

- a. 0.719; 0.329
- b. None of the alternatives.
- c. 0.3791; 0.9678
- d. 0.2122; 0.4559

compute hidden layer activations

$(x_1, x_2) = (0.5, 1.5)$

$y = 1$

$h_1 = w_{01}x_1 + w_{03}x_2 = \frac{1}{2}\cdot\frac{1}{2} + \frac{4}{5}\cdot\frac{3}{2} = \frac{1}{4} + \frac{6}{5} = \frac{29}{20}$

$h_2 = w_{02}x_1 + w_{04}x_2 = -\frac{3}{10}\cdot\frac{1}{2} + \frac{1}{5}\cdot\frac{3}{2} = -\frac{3}{20} + \frac{3}{10} = \frac{3}{20}$

$\left.\begin{array}{l} h_1 = 1.45 \\ h_2 = 1.5 \end{array}\right\}$

compute the output layer neuron

$z = w_{11}\cdot h_1 + w_{12}\cdot h_2 = \frac{7}{10}\cdot\frac{29}{20} - \frac{1}{2}\cdot\frac{3}{20} = \frac{203}{200} - \frac{3}{40} = \frac{203-15}{200} = \frac{188}{200}$

$z = 0.94$

Apply sigmoid activation to convert the real-valued output to a probability

$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-0.94}} \simeq 0.719$

Compute the BCE loss

$L(y,\hat{y}) = -[y\cdot\log(\hat{y}) + (1-y)\cdot\log(1-\hat{y})] =$

$= -1\cdot\log(0.719) + 0 = 0.329$

Question 4

Not yet answered   Marked out of 4.00   Flag question

In a Support Vector Machine (SVM) binary classification problem (in two dimensions), we are given the separating hyperplane:

$y = -\frac{3}{4} \cdot x + 3$

What is the distance (margin) between the two margin hyperplanes?

Select one:

- a. 2
- b. 5
- c. 0.4
- d. None of the alternatives.

a) decision boundary   decision function

$y = -\frac{3}{4}x + 3 \Rightarrow 4y = -3x + 12 \Rightarrow 3x + 4y - 12 = 0$ where $f(x) = 3x + 4y - 12$

Thus $w = [3, 4]$

Margin $= \frac{2}{\|w\|} = \frac{2}{\sqrt{3^2+4^2}} = \frac{2}{\sqrt{25}} = \frac{2}{5} = \boxed{0.4}$

b) $\frac{3}{4}x + y - 3 = 0$

$w = [\frac{3}{4}, 1]$

Margin $= \frac{2}{\sqrt{\frac{9}{16}+1}} = \frac{2}{\sqrt{\frac{25}{16}}} = \frac{2\cdot 4}{5} = \boxed{1.6}$

Question 5

Not yet answered   Marked out of 4.00   Flag question

Given the following points $p_1 = (7, 15)$, $p_2 = (11, 8)$, $p_3 = (7, 19)$, $p_4 = (11, 11)$, $p_5 = (4, 8)$, find the principal components and select the proper answer.

What is the percentage of the variance explained by the second eigenvector?

Select one:

- a. $\approx 25\%$
- b. $\approx 72\%$
- c. $\approx 5\%$
- d. None of the alternatives.

calculate mean of coord

$\bar{x} = \frac{7+11+7+11+4}{5} = 8$

$\bar{y} = \frac{15+8+19+11+8}{5} = \frac{61}{5} = 12.2$

Center the data

$P_1' = (-1, 2.8)$

$P_2' = (3, -4.2)$

$P_3' = (-1, 6.8)$

$P_4' = (3, -1.2)$

$P_5' = (-4, -4.2)$

Compute Covariance Matrix

centered matrix $X = \begin{bmatrix} -1 & 2.8 \\ 3 & -4.2 \\ -1 & 6.8 \\ 3 & -1.2 \\ -4 & -4.2 \end{bmatrix}$

$Cov(X) = \frac{1}{n-1} X^T \cdot X$

$Var(x) = \frac{1}{4}(1+9+1+9+16) = \frac{36}{4} = 9$

$Var(y) = \frac{1}{4}(7.84+17.64+46.24+1.44+17.64) = \frac{90.8}{4} = 22.7$

$Cov(x,y) = \frac{1}{4}(-2.8-12.6-6.8-3.6+16.8) = \frac{-9}{4} = -2.25$

$\Sigma = \begin{bmatrix} 9 & -2.25 \\ -2.25 & 2.27 \end{bmatrix} \xrightarrow{\text{eigenvalues}} \begin{array}{l} \lambda_1 \simeq 23.06 \\ \lambda_2 \simeq 8.64 \end{array}$

Total Variance

$\lambda_1 + \lambda_2 = 31.7$ so variance explained

$\frac{23.06}{31.7} = 72.7\%$

$\frac{8.64}{31.7} = 27.3\%$

Question 6

Not yet answered Marked out of 4.00 Flag question

You are given the points and labels presented in the table below, which result to a hard margin SVM with the following hyperplane: $-2 \cdot x_1 + 3 \cdot x_2 - 5 = 0$

Point number	x_1	x_2	label
1	1	2	-1
2	4	4	-1
3	1	1	-1
4	2	1	-1
5	0	2	1
6	3	4	1
7	2	4	1
8	4	5	1

margin hyperplanes: $-2x_1 + 3x_2 - 5 = \pm 1$

$f(x) = -2x_1 + 3x_2 - 5$

$f(P_1) = -2+6-5 = -1$ ✓
$f(P_2) = -8+12-5 = -1$ ✓
$f(P_3) = -2+3-5 = -4$
$f(P_4) = -4+3-5 = -6$
$f(P_5) = 0+6-5 = 1$ ✓

$f(P_6) = -6+12-5 = 1$ ✓
$f(P_7) = -4+12-5 = 3$
$f(P_8) = -8+15-5 = 2$

Select the option that has the points that lie on the separating hyperplanes.

P_1, P_2 on negative margin hyperplane
P_5, P_6 on positive margin hyperplane

Select one:

- a. None of the alternatives.
- b. Points 1, 2, 5, 6
- c. All points
- d. Points 1, 5

Question 7

Not yet answered Marked out of 4.00 Flag question

Suppose you have a perceptron with initial weights **w = [0.2, -0.1, 0.3, 0.0]**, learning rate **η=0.5**, and input **x= [1,0,1,1]** with true label **y=-1**. After **one incorrect prediction**, what are the updated weights?

Select one:

- a. [0.2, −0.1, 0.3, 0.0]
- b. [0.7, −0.1, 0.8, 0.5]
- c. [−0.3, −0.1, −0.2, −0.5]
- d. [−0.5, −0.6, −0.7, −1.0]
- e. [0.0, 0.0, 0.0, 0.0]
- f. None of the alternatives.

$\Delta w = \eta \cdot y \cdot x = \frac{1}{2} \cdot (-1) \cdot [1,0,1,1] = [-\frac{1}{2}, 0, -\frac{1}{2}, -\frac{1}{2}]$

$w_{new} = w + \Delta w = [-0.3, -0.1, -0.2, -0.5]$

Question 8

Not yet answered Marked out of 4.00 Flag question

What condition causes the Perceptron algorithm to enter an infinite loop?

Select one:

- a. Negative input values. no problem
- b. Perfect linear separability. this causes convergence
- c. None of the alternatives.
- d. Non-linear separability of data.
- e. High-dimensional feature vectors. no problem as long as they're linearly separable
- f. Too large learning rate. can cause overshooting, not infinite loops.

Question 9

Not yet answered Marked out of 4.00 Flag question

If a ReLU neuron's input is -0.75, what is its gradient during backpropagation?

Select one:

- a. 0.75
- b. 0
- c. 1
- d. Undefined
- e. -0.75
- f. None of the alternatives.

$ReLU(x) = \max(0, x) \xrightarrow[\text{pass}]{\text{forward}} ReLU(-0.75) = 0$

$\frac{d}{dx} ReLU(x) = \begin{cases} 1 & , x > 0 \\ 0 & , x < 0 \\ \text{undefined} & , x = 0 \end{cases}$

$\frac{d}{dx} ReLU(-0.75) = 0$

Question 10

Not yet answered Marked out of 4.00 Flag question

A neural network has weights updated as $w_{new} = w_{old} - \eta \cdot \frac{\partial Loss}{\partial w}$. If the learning rate η is too high, what happens during training?

Select one:

- a. Immediate optimal solution. too optimistic
- b. Possible non-convergence.
- c. Gradient becomes zero permanently.
- d. Stable convergence. opposite of what happens
- e. Slow convergence. happens with small learning rate
- f. None of the alternatives.

Written Assignment 5

Topic 2: Neural Networks

(15 total points) Perform the necessary calculations and answer the following three questions regarding the full learning cycle of the neural network shown in Figure 1. This network consists of one hidden layer and is used for a binary classification task.

The activation function for the hidden layer is ReLU:

$$ReLU(x) = \max(0, x)$$

The output neuron uses the sigmoid activation function:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

The loss function is Binary Cross-Entropy (BCE):

$$L(y, \hat{y}) = -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$$

where $\hat{y}$ is the prediction/output of the neural network, y is the true label, and log refers to the natural logarithm.

The given neural network has its weights and biases randomly initialized as follows:

w_{01}	w_{02}	w_{03}	w_{04}	w_{11}	w_{12}	b_{01}	b_{02}	b_{10}
0.15	0.25	0.30	0.10	0.45	0.55	0.1	0.2	0.3

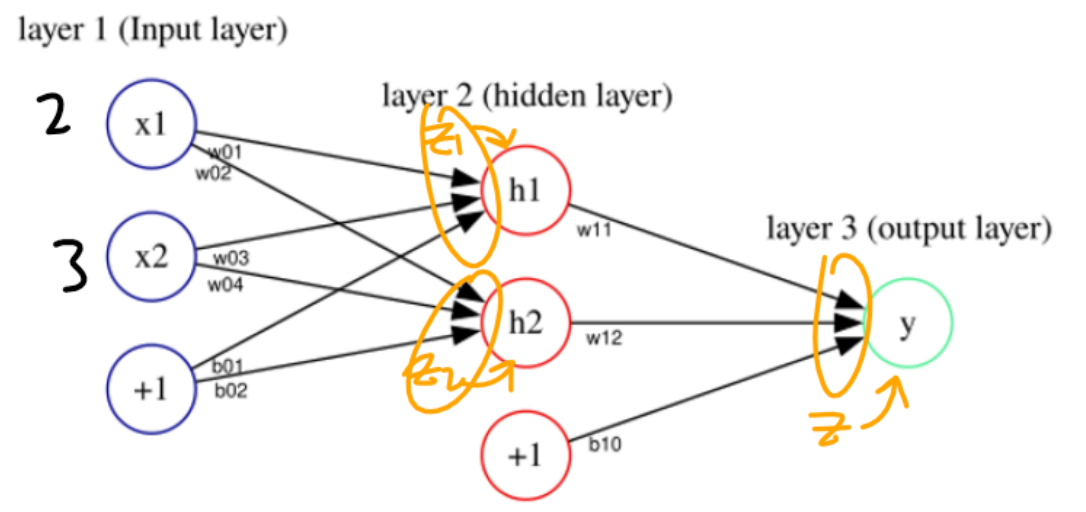


Figure 1: Architecture of examined neural net.

(a) (**5 points**) Let's assume that the input of the provided neural net is $(x_1, x_2) = (2, 3)$, whose ground truth value y is equal to 1. Execute only one forward pass for that input and compute and report the value of the network's Loss function by filling in the place indicated by the numbered question mark. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step). Assume the weights as shown in the table above.

activation

$$ReLU(x) = \max(0, x)$$

output

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \text{Sigmoid}$$

loss function

$$BCE \quad L(y, \hat{y}) = -(y\log\hat{y} + (1-y)\log(1-\hat{y}))$$

$$(x_1, x_2) = (2, 3), \quad \text{ground truth } y = 1$$

- 1 hidden layer
- binary classification

a) Forward Pass

hidden layer activations

$$h_1 = ReLU(\overbrace{x_1 \cdot w_{01} + x_2 \cdot w_{03} + b_{01}}^{z_1}) = ReLU(2\times 0.15 + 3\times 0.3 + 0.1) = ReLU(1.3) = 1.3$$

$$h_2 = ReLU(\underbrace{x_1 \cdot w_{02} + x_2 \cdot w_{04} + b_{02}}_{z_2}) = ReLU(2\times 0.25 + 3\times 0.1 + 0.2) = ReLU(1) = 1$$

output neuron input

$$z = h_1 \cdot w_{11} + h_2 \cdot w_{12} + b_{10} = 1.3\times 0.45 + 0.55 + 0.3 = 1.435$$

Output neuron (sigmoid activation)

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-1.435}} = 0.807$$

BCE loss

$$L(y, \hat{y}) = -[1\cdot \log(0.807) + (1-1)\cdot \cancel{\log(0.293)}^{0}] \Rightarrow$$

$$L(1, 0.807) = -\log(0.807) = \boxed{0.213}$$

(b) (**5 points**) Using the same input, perform **one backpropagation pass** through the network and update the weights and biases using the **gradient descent algorithm**. Assume a learning rate of $\eta = 0.1$. Provide the updated value for each of the weights. Fill in the positions indicated by numbered question marks in the table below with the correct new weight values. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

Answer:

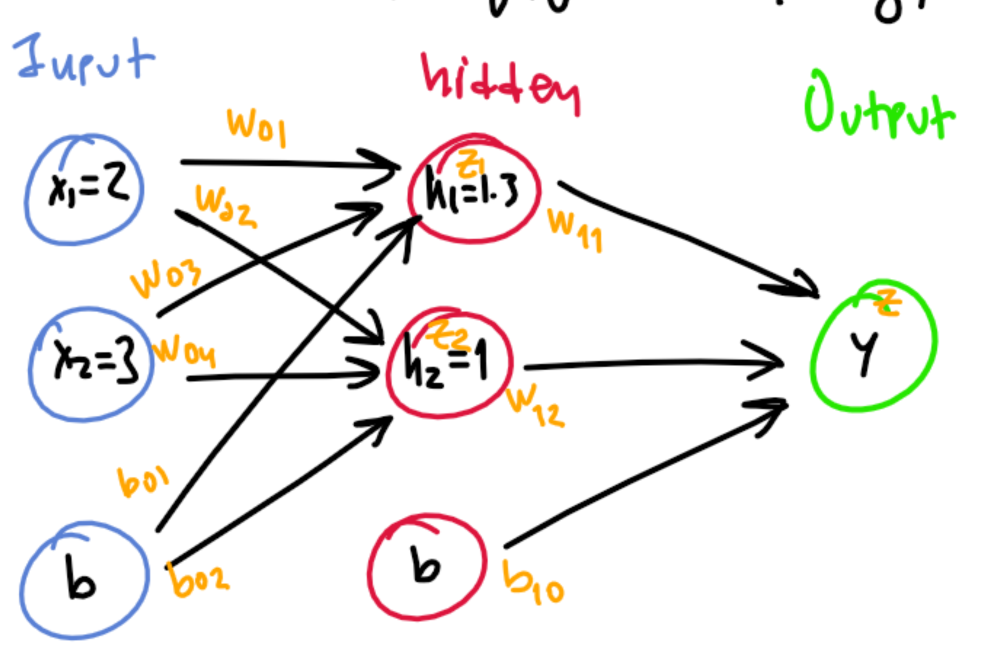
w_{01}	w_{02}	w_{03}	w_{04}	w_{11}	w_{12}	b_{01}	b_{02}	b_{10}
?1	?2	?3	?4	0.475	?5	?6	?7	?8

$x_1 = 2$, $x_2 = 3$, $y = 1$, $\eta = 0.1$

$h_1 = 1.3$, $h_2 = 1$, $z = 1.435$, $\hat{y} = \sigma(z) = 0.807$

$L(y, \hat{y}) = 0.213$

$$L(y, \hat{y}) = -(y\log\hat{y} + (1-y)\log(1-\hat{y}))$$



$z = h_1 w_{11} + h_2 w_{12} + b_{10}$

$z_1 = x_1 w_{01} + x_2 \cdot w_{03} + b_{01}$

$z_2 = x_1 w_{02} + x_2 w_{04} + b_{02}$

Gradient @ Output Layer

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}} + (-1)\cdot\frac{1-y}{1-\hat{y}} = -\frac{1}{0.807} = -1.239$$

$$\frac{\partial \hat{y}}{\partial z} = -\frac{1}{(1+e^{-z})^2}\cdot e^{-z}\cdot(-1) = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1+e^{-z}-1}{(1+e^{-z})^2} = \frac{\cancel{1+e^{-z}}}{(1+e^{-z})^{\cancel{2}}} - \frac{1}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} - \frac{1}{(1+e^{-z})^2} = \hat{y} - (\hat{y})^2 = \hat{y}(1-\hat{y}) = 0.807(1-0.807) =$$

$$= 0.807\times 0.193 = 0.155$$

$$\frac{\partial L}{\partial z} = \frac{\partial L}{\partial \hat{y}}\cdot\frac{\partial \hat{y}}{\partial z} = -1.239\times 0.155 = -0.192$$

calculations were repeated with no truncation before the final result.

Gradients for weights & bias to output neuron

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial z}\cdot\frac{\partial z}{\partial w_{11}} = -0.192\times h_1 = -0.192\times 1.3 = -0.249 \Rightarrow w_{11}' = w_{11} - 0.1\times(-0.249) = 0.45 + 0.024 = 0.474$$

$$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial z}\cdot h_2 = -0.192\times 1 = -0.192 \Rightarrow w_{12}' = 0.55 - 0.1\times(-0.192) = 0.55 + 0.019 = 0.569$$

$$\frac{\partial L}{\partial b_{10}} = \frac{\partial L}{\partial z}\cdot 1 = -0.192 \Rightarrow b_{10}' = 0.3 - 0.1\times(-0.192) = 0.3 + 0.019 = 0.319$$

Backpropagate to hidden layer

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial z}\cdot\frac{\partial z}{\partial h_1} = \frac{\partial L}{\partial z}\cdot W_{11} = -0.192 \times 0.45 = -0.086$$

$$\frac{\partial L}{\partial h_2} = -0.192 \times 0.55 = -0.105$$

$$ReLU'(x) = \begin{cases} 0, & x<0 \\ \text{undefined}, & x=0 \\ 1, & x>0 \end{cases}$$ therefore $ReLU'(z_1) = ReLU'(z_2) = 1$ since $z_1>0, z_2>0$

Gradients for hidden layer weights & biases

$$\frac{\partial L}{\partial w_{01}} = \frac{\partial L}{\partial h_1}\cdot\frac{\partial h_1}{\partial w_{01}} = \frac{\partial L}{\partial h_1}\cdot\left(ReLU'(z_1)\cdot\frac{\partial z_1}{\partial w_{01}}\right) = \frac{\partial L}{\partial h_1}\cdot x_1 = -0.086\times 2 = -0.172 \implies w_{01}' = w_{01} - 0.1\times(-0.172) = 0.15 + 0.017 = \boxed{0.167}$$

$$\frac{\partial L}{\partial w_{03}} = \frac{\partial L}{\partial h_1}\cdot\frac{\partial h_1}{\partial w_{03}} = \frac{\partial L}{\partial h_1}\cdot\left(ReLU'(z_1)\cdot\frac{\partial z_1}{\partial w_{03}}\right) = \frac{\partial L}{\partial h_1}\cdot x_2 = -0.086\times 3 = -0.258 \implies w_{03}' = 0.3 - 0.1\times(-0.258) = 0.3 + 0.025 = \boxed{0.325}$$

$$\frac{\partial L}{\partial b_{01}} = \frac{\partial L}{\partial h_1}\cdot\frac{\partial h_1}{\partial b_{01}} = \frac{\partial L}{\partial h_1}\cdot\left(ReLU'(z_1)\cdot\frac{\partial z_1}{\partial b_{01}}\right) = (-0.086)\times 1\times 1 = -0.086 \implies b_{01}' = 0.1 - 0.1\times(-0.086) = 0.1 + 0.008 = \boxed{0.108}$$

$$\frac{\partial L}{\partial w_{02}} = \frac{\partial L}{\partial h_2}\cdot x_1 = -0.105\times 2 = -0.21 \implies w_{02}' = 0.25 - 0.1\times(-0.21) = 0.25 + 0.021 = \boxed{0.271}$$

$$\frac{\partial L}{\partial w_{04}} = \frac{\partial L}{\partial h_2}\cdot x_2 = -0.105\times 3 = -0.315 \implies w_{04}' = 0.1 - 0.1\times(-0.315) = 0.1 + 0.031 = \boxed{0.131}$$

$$\frac{\partial L}{\partial b_{02}} = -0.105 \implies b_{02}' = 0.2 - 0.1(-0.105) = 0.2 + 0.01 = \boxed{0.210}$$

calculations were repeated with no truncation before the final result.

Answer:

w_{01}	w_{02}	w_{03}	w_{04}	w_{11}	w_{12}	b_{01}	b_{02}	b_{10}
0.167	0.271	0.325	0.131	0.474	0.569	0.108	0.210	0.319

gradients: −0.172 −0.21 −0.258 −0.315 −0.249 −0.192 −0.086 −0.105 −0.192

(c) (**5 points**) Using the same input again, perform **one backpropagation pass** through the network and update the weights and biases using the **Adam algorithm**. Assume the following hyper-parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$. Provide the updated value for each of the weights. Assume that you start with the weights as shown in Table 1. Fill in the positions indicated by numbered question marks in the table below with the correct new weight values. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

The Adam optimizer operates as follows (g_t is the gradient wrt the weights at iteration t):

1. $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
2. $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
3. $\mu_t = \frac{m_t}{1-\beta_1^t}$
4. $\alpha_t = \frac{v_t}{1-\beta_2^t}$
5. $w_t = w_{t-1} - \frac{\eta}{\sqrt{\alpha_t}+\varepsilon}\mu_t$

Answer:

w_{01}	w_{02}	w_{03}	w_{04}	w_{11}	w_{12}	b_{01}	b_{02}	b_{10}
?1	?2	?3	?4	?5	0.649	?6	?7	?8

$\beta_1 = 0.9$

$\beta_2 = 0.999$

$\varepsilon = 10^{-8}$

$\eta = 0.1$

for $t=1$: $m_0 = 0$, $v_0 = 0$

- **w_{01}** $\quad w_{01}^{(0)} = 0.15 \quad g_1 = -0.172$

1. $m_1 = 0.9\times \cancel{m_0}^{0} + (1-0.9)\times g_1 = 0.1\times g_1$
2. $v_1 = 0.999\times \cancel{v_0}^{0} + (1-0.999)\times g_1^2 = 0.001\times g_1^2$
3. $\mu_1 = \frac{m_1}{1-\beta_1} = \frac{0.1\times g_1}{0.1} = g_1$
4. $\alpha_1 = \frac{v_1}{1-\beta_2} = \frac{0.001\times g_1^2}{0.001} = g_1^2$
5. $w_{01}^{(1)} = w_{01}^{(0)} - \frac{\eta}{\sqrt{\alpha_1}+\varepsilon}\times\mu_1 = w_{01}^{(0)} - \frac{\eta}{\sqrt{g_1^2}+\varepsilon}\times g_1 =$

$= 0.15 - \frac{0.1}{|g_1| + 10^{-8}}\times g_1 = 0.15 - \frac{0.1\times(-0.172)}{0.172 + 10^{-8}} =$

$= 0.15 + \frac{0.0172}{0.172000001} = \boxed{0.249}$

Similarly......

Answer:

w_{01}	w_{02}	w_{03}	w_{04}	w_{11}	w_{12}	b_{01}	b_{02}	b_{10}
0.249	0.349	0.399	0.199	0.549	0.649	0.199	0.299	0.399

Topic 3: Support Vector Machines (SVM)

(15 total points) This topic concerns training of Support Vector Machine classifiers using the gradient descent method, in two scenarios: (a) a linearly separable dataset and (b) a non-linearly separable dataset. You will base your solution on concepts and methods described in Chapter 12, Section 12.3 of "J. Leskovec, A. Rajaraman & J.D. Ullman (2020). Mining of Massive Datasets (3rd edition). Cambridge University Press (MMDS).

The first dataset consists of two classes with three instances each. The coordinates and labels of these six instances are provided in the table shown below.

Point	x-axis	y-axis	Label
A	4	2	+1
B	2	1	+1
C	3	3	+1
D	0.5	3	-1
E	2	5	-1
F	1	1	-1

Apply a Support Vector Machine classifier using the gradient descent method for 2 iterations, using batch learning and answer the next sub-questions. The objective loss function to minimize is Equation 12.4 from Section 12.3.3 of the MMDS book, which includes regularization and hinge loss terms.

Assume the following parameters whenever a SVM classifier is to be trained throughout all sub-questions of Topic 3:

- Regularization parameter C = 2
- Learning rate η = ¼
- Initial weight vector w = [u, v] = [-1, 0]
- Initial bias b = -1

(a) (8 points) Train the SVM classifier for 2 iterations using batch learning. Fill in the numbered question marks in the table shown below which represents the first iterations of gradient descent. Decision Pattern stands for column 'Bad' in Figure 12.20 in the course book MMDS. Furthermore, the status of each condition is represented by a sequence of x's and o's, with 'x' representing a condition that does not hold (false) and 'o' representing one that does (true). Do all calculations with the greatest accuracy. Report the final results with 4 decimal places of accuracy (do not round up or down the numbers at any calculation step).

$$f(\mathbf{w}, b) = \frac{1}{2}\sum_{j=1}^{d} w_j^2 + C\sum_{i=1}^{n} \max\Big\{0,\ \overbrace{\underbrace{1 - y_i\Big(\sum_{j=1}^{d} w_j x_{ij} + b\Big)}_{\text{hinge loss}}}^{\text{margin}}\Big\} \qquad (12.4)$$

where $d=2$ (dimensionality)

$w_j = j$-th component of weight vector $w = [u, v] = [-1, 0]$

$x_{ij} = j$-th feature of example i

$\sum_{j=1}^{d} w_j x_{ij} = w_1 x_{i1} + w_2 x_{i2} = w \cdot x_i$

$b = -1$ (bias term)

$C = 2$, $\eta = \frac{1}{4}$

regularization: $\frac{1}{2}\|w\|^2$

	$\mathbf{w} = [u, v]$	b	Bad	$\partial/\partial u$	$\partial/\partial v$	$\partial/\partial b$
(1)	[0.000, 1.000]	−2.000	oxoooo	−0.200	0.800	−2.100
(2)	[0.040, 0.840]	−1.580	oxoxxx	0.440	0.940	−1.380
(3)	[−0.048, 0.652]	−1.304	oxoxxx	0.352	0.752	−1.104
(4)	[−0.118, 0.502]	−1.083	xxxxxx	−0.118	−0.198	−1.083
(5)	[−0.094, 0.542]	−0.866	oxoxxx	0.306	0.642	−0.666
(6)	[−0.155, 0.414]	−0.733	xxxxxx			

x: false o: true (Bad)

Figure 12.20: Beginning of the process of gradient descent

batch gradient descent for 2 iter. using hinge loss+regularization.

$$f(w,b) = \frac{1}{2}\|w\|^2 + 2\cdot\sum_{i=1}^{n}\max(0, 1 - y_i(wx_i + b))$$

$m_i = \text{decision} = y_i(wx_i + b)$ → decision ≥ 1, hinge loss $= 0$ ⇒ o : good
→ decision < 1, hinge loss $= 1 - m_i > 0$ ⇒ x : bad

We begin by incorporating b as the third component of $\mathbf{w}$, and for notational convenience, we shall use u and v as the first two components, rather than the customary w_1 and w_2. That is, we take $\mathbf{w} = [u, v, b]$. We also expand the two-dimensional points of the training set with a third component that is always 1. That is, the training set becomes

Iteration 1 $(w_0 = [-1, 0], b_0 = -1)$

Iterate until convergence:
- For $j = 1, \dots, d$
  - **Evaluate:** $\nabla(j) = \frac{\partial f(w,b)}{\partial w_j} = w_j + C\sum_{i=1}^{n}\frac{\partial L(x_i, y_i)}{\partial w_j}$
  - **Update:** $w_j \leftarrow w_j - \eta\nabla(j)$

$y_i(wx_i+b)$

$([4,2,1],1)$ d → $1\cdot(-1\cdot 4 + 0\cdot 2 - 1) = -5 < 1$ x
$([2,1,1],1)$ e → $1\cdot(-1\cdot 2 + 0 - 1) = -3 < 1$ x
$([3,3,1],1)$ c → $1\cdot(-1\cdot 3 + 0 - 1) = -4 < 1$ x
$([0.5,3,1],-1)$ i → $-1\cdot(-1\cdot 0.5 + 0 - 1) = 1.5 > -1$ o
$([2,5,1],-1)$ s → $-1\cdot(-1\cdot 2 + 0 - 1) = 3 > -1$ o
$([1,1,1],-1)$ i → $-1\cdot(-1\cdot 1 + 0 - 1) = 2 > -1$ o
o
n

Bad Points (A, B, C)

$\nabla_w = w_0 - C\cdot\sum y_i x_i = [-1,0] - 2\cdot[(4,2)+(2,1)+(3,3)] = [-1,0] - 2\cdot[9,6]$
$= [-19, -12]$

$\nabla_b = -C\cdot\sum y_i = -2(1+1+1) = -6$

Update the weights

$w_1 = w_0 - \eta\cdot\nabla_w = [-1,0] - 0.25\cdot[-19,-12] = [-1,0] + [4.75, 3] = [3.75, 3]$

$b_1 = b_0 - \eta\cdot\nabla_b = -1 - 0.25\cdot(-6) = -1 + 1.5 = 0.5$

Iteration 2 $(w_1 = [3.75, 3], b_1 = 0.5)$

$y_i(wx_i+b)$

$([4,2,1],1)$ d → $1\times(3.75\times 4 + 3\cdot 2 + 0.5) = 15 + 6 + 0.5 = 21.5 > 1$ o
$([2,1,1],1)$ e → $1\times(3.75\times 2 + 3\cdot 1 + 0.5) = 11 > 1$ o
$([3,3,1],1)$ c → $1\times(3.75\times 3 + 3\cdot 3 + 0.5) = 20.75 > 1$ o
$([0.5,3,1],-1)$ i → $-1\times(3.75\times 0.5 + 3\cdot 3 + 0.5) = -11.375 < -1$ x
$([2,5,1],-1)$ s → $-1\times(3.75\times 2 + 3\cdot 5 + 0.5) = -23 < -1$ x
$([1,1,1],-1)$ i → $-1\times(3.75\times 1 + 3\times 1 + 0.5) = -7.25 < -1$ x
o
n

Bad Points (D, E, F)

$\nabla_w = w_1 - C\cdot\sum y_i x_i = [3.75, 3] - 2[-1\times(0.5,3) + (-1)\times(2,5) + (-1)\times(1,1)] = [3.75, 3] + 2[3.5, 9] = [3.75, 3] + [7, 18] = [10.75, 21]$

$\nabla_b = -C\cdot\sum y_i = -2\cdot(-1-1-1) = 6$

Update the weights

$w_2 = w_1 - \eta\nabla_w = [3.75, 3] - 0.25[10.75, 21] = [1.0625, -2.25]$

$b_2 = b_1 - \eta\nabla_b = 0.5 - 0.25\times 6 = -1$

Answer:

Iteration	Decision Pattern (Bad)	u w[0]	v w[1]	b
0 (initial)	-	-1	0	-1
1	x x x o o o	3.75	3	0.5
2	o o o x x x	1.0625	-2.25	-1

(b) (1 point) Using the final weight vector and bias after the second iteration, calculate the margin. Replace the numbered question mark below with the correct value. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

$$\gamma = \frac{1}{\|w\|} = \frac{1}{\sqrt{u^2+v^2}} = \frac{1}{\sqrt{1.0625^2+(-2.25)^2}} = \frac{1}{\sqrt{6.19140625}} = 0.401$$

Answer:

Margin = **?1**

(c) (1 point) Using the same final weight vector and bias, calculate the confidence scores *(a confidence measure in SVM classification, based on its distance from the margin boundary*, MMDS Section 13.3.3, Equation 12.4) for the points by replacing the respective numbered question marks below with the correct values. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

$d(x,y) = ux + vy + b$

After iteration 2

$u = 1.0625$

$v = -2.25$

$b = -1$

Point	x-axis	y-axis	Label
A	4	2	+1
B	2	1	+1
C	3	3	+1
D	0.5	3	-1
E	2	5	-1
F	1	1	-1

Answer:

Point A = **?1**

Point C = **?2**

Point E = **?3**

Point F = **?4**

$(u,v,b) = (1.0625, -2.25, -1)$

$\underline{ux+vy+b}$

Point A: $4\times 1.0625 + 2\times(-2.25) + (-1) = -1.25$

Point C: $3\times 1.0625 + 3\times(-2.25) + (-1) = -4.5625$

Point E: $2\times 1.0625 + 5\times(-2.25) + (-1) = -10.125$

Point F: $1\times 1.0625 + 1\times(-2.25) + (-1) = -2.1875$

(d) (5 points) Assume that the dataset is extended with a new point V (see table below), resulting in a non-linearly separable scenario. The presence of slack variables ξ_i is now required as part of the hinge loss in the objective function.

Point	x-axis	y-axis	Label
A	4	2	+1
B	2	1	+1
C	3	3	+1
D	0.5	3	-1
E	2	5	-1
F	1	1	-1
V	3.6	4.8	-1

Repeat the training of the SVM classifier for 2 iterations using batch learning. Fill in the numbered question marks in the table shown below which represents the first iterations of gradient descent. Do all calculations with the greatest accuracy. Report the final results with 2 decimal places of accuracy (do not round up or down the numbers at any calculation step).

Iteration 1 $(w_0 = [-1,0], b_0 = -1)$

$A = -5 < 1$ x

$B = -3 < 1$ x

$C = -4 < 1$ x

Bad Points (A,B,C)

$D = 1.5 > -1$ o

$E = 3 > -1$ o

$F = 2 > -1$ o

$V = ([3.6, 4.8, 1], -1) \rightarrow -1\times(-1\times 3.6 + 0\times 4.8 - 1) = -(-3.6-1) = 4.6 > -1$ x

$\nabla w = [-19, -12]$ $\nabla_b = -6$

Updated weights

$w_1 = [3.75, 3]$

$b_1 = 0.5$

same as before

Iteration 2 $(w_1 = [3.75, 3], b_1 = 0.5)$ Again A, B, ..., F are the same as before

$A = 21.5 > 1$ o

$B = 11 > 1$ o

$C = 20.75 > 1$ o

$D = -11.375 < -1$ x

$E = -23 < -1$ x

$F = -7.25 < -1$ x

$V = -1\times(3.75\times 3.6 + 3\times 4.8 + 0.5) = -(13.5 + 14.4 + 0.5) = -28.4 < -1$ x

Bad Points (D,E,F,V)

$\nabla w = w_1 - C\cdot\sum y_i x_i = [3.75, 3] - 2[-1\times(0.5,3) + (-1)\times(2,5) + (-1)\times(1,1) + (-1)(3.6,4.8)] = [3.75,3] + 2\times[(3.5,9) + (3.6,4.8)] = [3.75,3] + 2\times[7.1, 13.8] =$
$= [3.75, 3] + [14.2, 27.6] = [17.95, 30.6]$

$\nabla b = -C\cdot\sum y_i = -2(-1-1-1-1) = 8$

Update the weights

$w_2 = w_1 - \eta\nabla w = [3.75, 3] - 0.25[17.95, 30.6] = [3.75, 3] + [-4.4875, -7.65] = [-0.7375, -4.65]$

$b_2 = b_1 - \eta\nabla b = 0.5 - 0.25\times 8 = -1.5$

Answer:

Iteration	Decision Pattern (Bad)	u	v	b
0 (initial)	-	-1	0	-1
1	xxxoooo	3.75	3	0.5
2	ooo xxxx	-0.73	-4.65	-1.50

Topic 4: Principal Component Analysis (PCA)

(15 total points) Autonomous robots often navigate through rough terrain by analyzing point cloud data to detect the ground plane. One way to achieve this is by using **Principal Component Analysis** (PCA) to fit a plane to a set of 3D points.

In this exercise, you will use PCA to **analyze a small point cloud dataset and determine the best-fitting plane**. The principal components will reveal the dominant directions of variation in the data, allowing us to extract the normal vector of the plane.

Problem Context

A quadruped robot is navigating uneven terrain. Its onboard **LiDAR sensor** captures 3D points from the **ground surface**. To estimate the robot's orientation relative to the ground, we need to extract the ground plane from a set of noisy 3D points.

Your task is to analyze the given 3D point cloud and **extract the plane equation using PCA**.

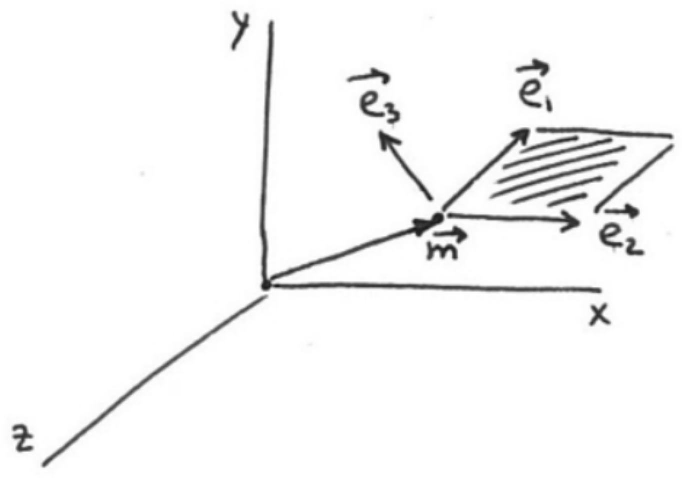


Figure 1: PCA for plane extraction demonstration.

Answer:

m = (?1, ?2, 9.27)

ID	X	Y	Z
1	?3	?4	?5
2	?6	-7.16	?7
3	?8	?9	?10
4	?11	?12	1.6
5	3.976	?13	?14

Data

ID	X	Y	Z
1	2.02	9.55	32.52
2	-5.66	-3.37	2.62
3	4.81	7.93	18.24
4	7.91	7.48	10.87
5	7.24	-2.64	-17.9

The points in the table above **lie approximately on a plane**, which we want to extract.

$m = (\bar{x}, \bar{y}, 9.27)$

$\bullet\ \bar{x} = \frac{1}{5} \cdot \sum x_i = \frac{16.32}{5} = 3.264$

$\bullet\ \bar{y} = \frac{1}{5} \sum y_i = \frac{18.95}{5} = 3.79$

$m = (3.264, 3.79, 9.27)$

The process for extracting the plane is as follows:

1) we compute the mean of the data,
2) we compute the centered data points (i.e. subtracting the mean from the points),
3) we compute the sample covariance matrix of the centered data points,
4) we compute the eigenvalues and eigenvectors of the covariance matrix,
5) we sort the eigenvectors from largest to smallest eigenvalue, and the eigenvector with the smallest eigenvalue is the plane normal (as shown in Figure 1).

Answer the questions below.

(a) (1 point) The first step in PCA is to compute the mean point (m) and center the data by subtracting the mean ($X_{centered} = X - m$). Fill in the numbered question marks for the mean point m and centered data. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

Data

				$X - \bar{x}$	$Y - \bar{y}$	$Z - \bar{z}$			
ID	X	Y	Z	X_{cent}	Y_{cent}	Z_{cent}	X_{cent}^2	Y_{cent}^2	Z_{cent}^2
1	2.02	9.55	32.52	-1.244	5.76	23.25	1.548	33.178	540.563
2	-5.66	-3.37	2.62	-8.924	-7.16	-6.65	79.640	51.266	44.223
3	4.81	7.93	18.24	1.546	4.14	8.97	2.391	17.140	80.451
4	7.91	7.48	10.87	4.646	3.69	1.6	21.591	13.616	2.56
5	7.24	-2.64	-17.9	3.976	-6.43	-27.170	15.808	41.345	738.179
							120.978	39.13625	357.494

Same for XY, XZ, YZ

(b) (6 points) The next step is to compute the sample 3x3 covariance matrix. Fill in the numbered question mark in the covariance matrix below with the correct values. Do all calculations with the greatest accuracy. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

$$Cov = \begin{bmatrix} cov(X,X) & cov(X,Y) & cov(X,Z) \\ cov(Y,X) & cov(Y,Y) & cov(Y,Z) \\ cov(Z,X) & cov(Z,Y) & cov(Z,Z) \end{bmatrix}$$

$$= \begin{bmatrix} 30.242 & 13.677 & -14.076 \\ 13.677 & 39.135 & 99.819 \\ -14.076 & 99.819 & 351.503 \end{bmatrix}$$

where $cov(X,Y) = cov(Y,X)$ etc

$Cov(X,X) = \frac{1}{5-1} \sum X_i^2$

$Cov(X,Y) = cov(Y,X) = \frac{1}{5-1} \sum (X_i \times Y_i)$

⋮ etc

Answer:

30.242	13.677	-14.076
13.677	39.135	99.819
-14.076	99.819	351.503

(c) (8 points) The final step is to compute the eigenvalues and eigenvectors, and find the parameters for the equation of the plane. You can assume that the mean point lies on the plane. We remind you that the equation of the plane is:

$$p_1 x + p_2 y + p_3 z = p_4$$

Fill in the numbered question marks in the table below with the correct values for the parameters p_i. Report the final results with 3 decimal places of accuracy (do not round up or down the numbers at any calculation step).

Answer:

p_1	p_2	p_3	p_4
0.496	-0.829	0.255	0.847

```
[26]: eigenvalues, eigenvectors = np.linalg.eigh(cov_matrix)
print(eigenvalues)
print(eigenvectors)
```

min eigenval

```
[1.81671260e-01 3.97558466e+01 3.80944462e+02]
[[ 0.49697294 -0.86732584 -0.02763675]
 [-0.82928918 -0.48407459  0.2791975 ]
 [ 0.25553345  0.11583474  0.95983591]]
```

The smallest eigenvalue is $\lambda_1 = 1.816$

So, the normal vector: $[0.496, -0.829, 0.255] = \vec{v}$

The mean point has to lie on the plane

$p_1 x + p_2 y + p_3 z = p_4$

$\Leftrightarrow [p_1, p_2, p_3] \begin{bmatrix} x \\ y \\ z \end{bmatrix} = p_4$, therefore $p_4 = \vec{v} \cdot \vec{m}$ where $\vec{m} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix}$

$p_4 =$

```
[28]: p1, p2, p3 = normal_vector
p4 = np.dot(normal_vector, mean)
print(p4)
```

$= 0.847$

```
0.8479087834136436
```