SC4001 Exam 2024-2025 Sem 1

1. (a) State whether each of the following statements is "TRUE" or "FALSE" with explanations. Each subquestion carries one mark.

Answer

F (i) The reason for using L1 or L2 regularization is for the model to converge toward the minimum training error more quickly.

T (ii) If two classes of samples are separable by a straight line, we can train a network without any non-linear activation in its hidden layers.

F (iii) Using Dropout always results in a lower evaluation error.

F (iv) Backpropagation is used to compute the optimal structure of a neural network, including the number of layers and neurons.

F (v) The *ReLU* (Rectified Linear Unit) activation function can introduce negative values in the output.

T (vi) Adding more layers in a neural network does not always result in a lower evaluation error.

T (vii) Using K-fold cross-validation is more often conducted when the dataset is small than when the dataset is large.

(7 marks)

1. (b) Give brief answers to the following. Each part carries 2 marks.

(i) State the shape of tensor [[[1], [0]], [[3], [4]], [[-2], [-1]]] .

Answer

Shape [3, 1, 2]

(ii) For Xavier/Glorat uniform weight initialization schemes, why are the gain values different for different activation functions?

Answer

The gain values are different for different activation functions because each activation function affects the variance of the outputs differently.

(iii) How would one decide to increase the number of layers (i.e. depth) versus the number of neurons per layer (i.e. width) in a neural network?

(6 marks)

Answer

- Depth for structured data (images, text, time-series).
- Width for small/low-dimensional datasets or when memory constraints favor fewer, wider layers.
- Balance depth and width based on task complexity, training stability, and generalization performance, validated empirically.

1. (c) Figure Q1 shows a dataset with each example belonging to one of the two classes, displayed in the space of its two features x_1 and x_2. You are to design a discrete perceptron network (i.e. with unit step activation functions) with a single hidden layer to classify the examples.

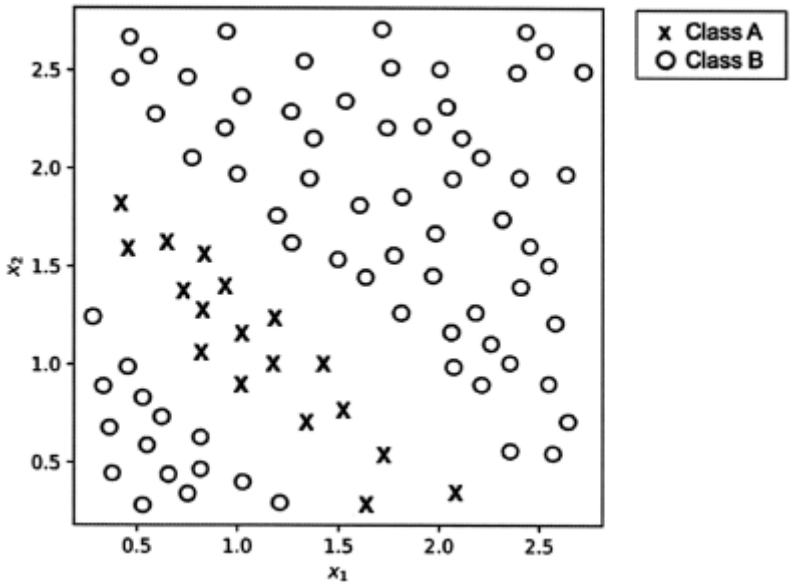
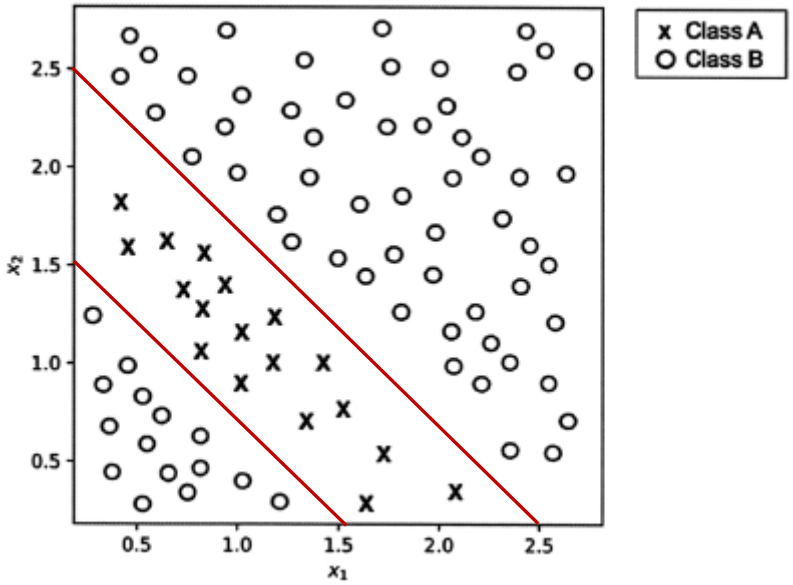


Figure Q1

(i) Draw a decision boundary separating the two classes to design the network.

(2 marks)

Answer



(ii) State the number of neurons in the hidden layer.

(2 marks)

Answer

Number of neurons = 2

1. (b) (iii) Draw the network indicating the values of all weights and biases.

(8 marks)

Answer

Line 1 (passing through (0, 1.5) and (1.5, 0))
Line 2 (passing through (0, 2.5) and (2.5, 0))

shaded above the line is +1 > 0
shaded below the line is -1 ≤ 0

Line 1 (passing through (0, 1.5) and (1.5, 0))

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0 - 1.5}{1.5 - 0} = -1$$

$$y = mx + c$$
$$1.5 = -1(0) + c$$
$$c = 1.5$$

$$x_2 = (-1)x_1 + 1.5$$
$$x_1 + x_2 - 1.5 = 0$$

since the shaded-region above the line,
$u_1 = -x_1 - x_2 + 1.5$

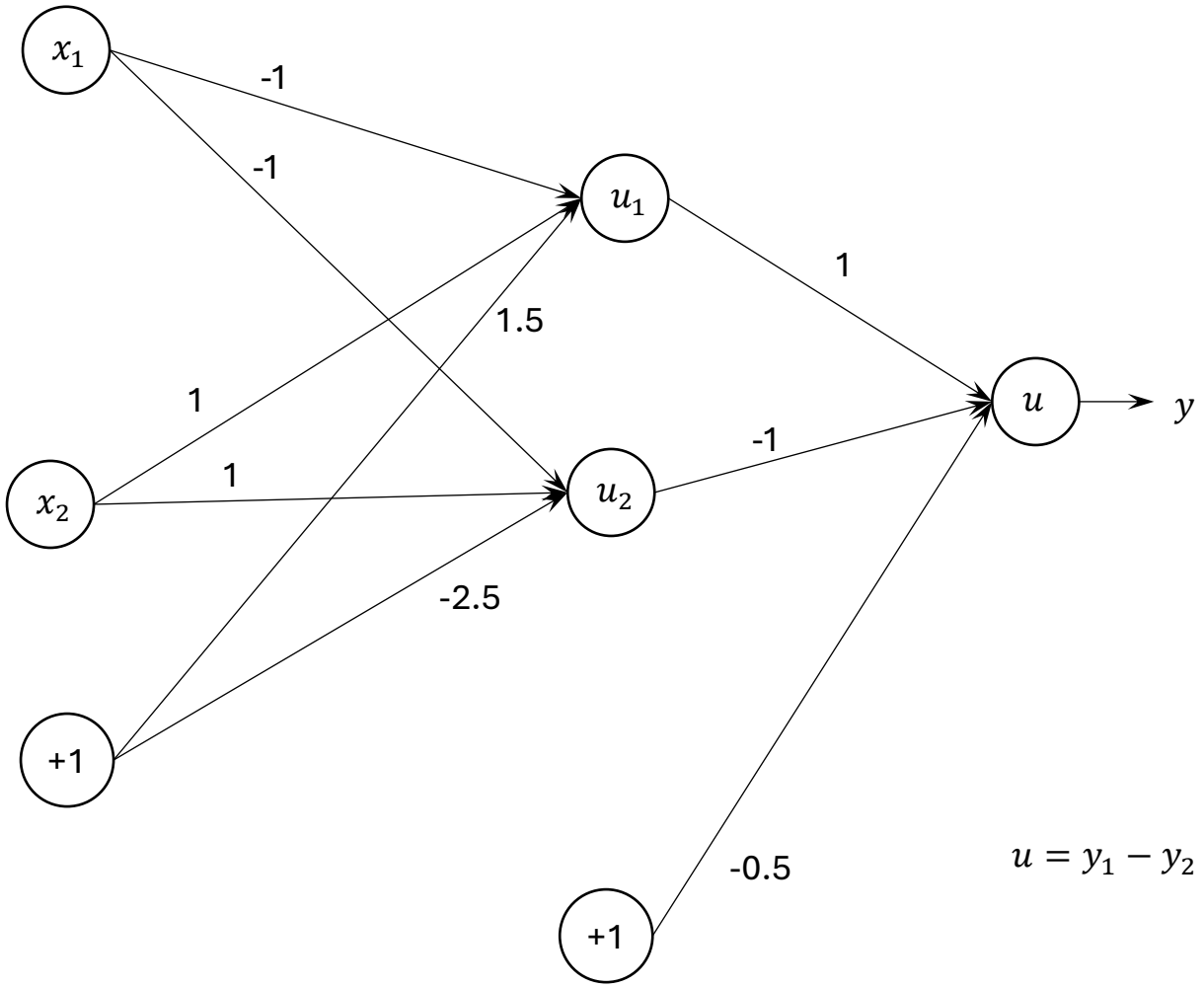
Line 2 (passing through (0, 2.5) and (2.5, 0))

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0 - 2.5}{2.5 - 0} = -1$$

$$y = mx + c$$
$$2.5 = -1(0) + c$$
$$c = 2.5$$

$$x_2 = (-1)x_1 + 2.5$$
$$x_2 = -x_1 + 2.5$$
$$x_1 + x_2 - 2.5 = 0$$

since the shaded-region below the line,
$u_2 = x_1 + x_2 - 2.5$



$u = y_1 - y_2 - 0.5$

2. (a) The 3-layer feedforward neural network shown in Figure Q2 receives two-dimensional inputs $(x_1, x_2) \in \boldsymbol{R}^2$ and produces a one-dimensional output y. The first hidden layer consists of three neurons and the second hidden layer consists of two neurons. All hidden neurons have *ReLU* activation functions and the output neuron is a logistic regression neuron. The weights of the network are initialized as indicated in Figure Q2 and all the biases are initialized to 0 (not shown).

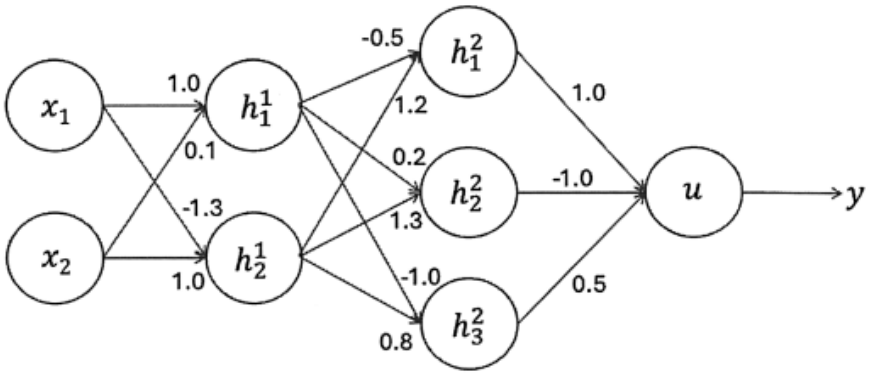


Figure Q2

The network is trained to produce a desired output $d = 0$ for an input $x = \begin{pmatrix} -1.0 \\ 1.5 \end{pmatrix}$. You are to perform one iteration of stochastic gradient descent learning with the example $(\boldsymbol{x}, d)$. Give your answers rounded up to three significant figures.

2. (a) Write initial weight matrices $\boldsymbol{W}$ and bias vectors $\boldsymbol{b}$, connected to the three layers. (3 marks)

Answer

$$W_1 = \begin{pmatrix} 1.0 & -1.3 \\ 0.1 & 1.0 \end{pmatrix} \qquad b_1 = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

$$W_2 = \begin{pmatrix} -0.5 & 0.2 & -1.0 \\ 1.2 & 1.3 & 0.8 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

$$W_3 = \begin{pmatrix} 1.0 \\ -1.0 \\ 0.5 \end{pmatrix} \qquad b_3 = 0.0$$

2. (b) Find the synaptic inputs $\boldsymbol{u}$ and outputs $\boldsymbol{h}$ of the two hidden layers.

(4 marks)

Answer

$$W_1 = \begin{pmatrix} 1.0 & -1.3 \\ 0.1 & 1.0 \end{pmatrix} \qquad b_1 = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

$$W_2 = \begin{pmatrix} -0.5 & 0.2 & -1.0 \\ 1.2 & 1.3 & 0.8 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

Input $x = \begin{pmatrix} -1.0 \\ 1.5 \end{pmatrix}$

Synaptic input to h_1,

$$u_1 = \boldsymbol{W}_1{}^T x + b_1 = \begin{pmatrix} 1.0 & 0.1 \\ -1.3 & 1.0 \end{pmatrix}\begin{pmatrix} -1.0 \\ 1.5 \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

$$= \begin{pmatrix} (1.0)(-1.0) + (0.1)(1.5) \\ (-1.3)(-1.0) + (1.0)(1.5) \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}$$

$$= \begin{pmatrix} -0.85 \\ 2.80 \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix} = \begin{pmatrix} -0.85 \\ 2.80 \end{pmatrix}$$

Output of h_1,

$$h_1 = relu(u_1) = \max\{0.0, u_1\} = \begin{pmatrix} \max\{0.0, -0.85\} \\ \max\{0.0, \quad 2.80\} \end{pmatrix} = \begin{pmatrix} 0.00 \\ 2.80 \end{pmatrix}$$

Synaptic input to h_2,

$$u_2 = \boldsymbol{W}_2{}^T h_1 + b_2 = \begin{pmatrix} -0.5 & 1.2 \\ 0.2 & 1.3 \\ -1.0 & 0.8 \end{pmatrix}\begin{pmatrix} 0.00 \\ 2.80 \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

$$= \begin{pmatrix} (-0.5)(0) + (1.2)(2.8) \\ (0.2)(0) + (1.3)(2.8) \\ (-1.0)(0) + (0.8)(2.8) \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix} + \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix}$$

Output of h_2,

$$h_2 = relu(u_2) = \max\{0.0, u_2\} = \begin{pmatrix} \max\{0.0, 3.36\} \\ \max\{0.0, 3.64\} \\ \max\{0.0, 2.24\} \end{pmatrix} = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix}$$

<u>Summary</u>

Synaptic input to h_1: $u_1 = \begin{pmatrix} -0.85 \\ 2.80 \end{pmatrix}$

Output of h_1: $h_1 = \begin{pmatrix} 0.00 \\ 2.80 \end{pmatrix}$

Synaptic input to h_2: $u_2 = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix}$

Output of h_2: $h_2 = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix}$

2. (c) Find the output $\boldsymbol{y}$ and the cross-entropy at the output layer.

(4 marks)

Answer

$$W_3 = \begin{pmatrix} 1.0 \\ -1.0 \\ 0.5 \end{pmatrix} \qquad b_3 = 0.0$$

Desired output $d = 0$

$$h_2 = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix}$$

Synaptic input to y,

$$u_3 = \boldsymbol{W}_3{}^T h_2 + b_3 = (1.0 \quad -1.0 \quad 0.5)\begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix} + 0.0$$

$$= (1)(3.36) + (-1)(3.64) + (0.5)(2.24) = 0.98 + 0.5 = 0.84$$

Output of y,

$$y = sigmoid(u_3) = \frac{1}{1 + e^{-u}} = \frac{1}{1 + e^{-0.84}} = 0.698$$

cross-entropy,

$$\mathrm{J} = -d\log(y) - (1-d)\log(1-y)$$
$$= -\log(1 - 0.698)$$
$$= -\log(0.302) = 1.20$$

<u>Summary</u>

output: $y = 0.698$

cross-entropy: $J = 1.20$

2. (d) Find the derivatives $f'(\boldsymbol{u})$ at the two hidden layers with respect to synaptic input $\boldsymbol{u}$, where f is the $ReLU$ activation function.

(2 marks)

Answer

$$y = f(u) = \max\{0, u\}$$

$$f'(u) = \begin{cases} 1, & if\ u > 0 \\ 0, & if\ u \leq 0 \end{cases}$$

Hidden layer h_1,

$$h_1 = f(u_1) = \max\{0.0, u_1\} = \begin{pmatrix} 0 \\ 2.80 \end{pmatrix}$$

$$f'(u_1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Hidden layer h_2,

$$h_2 = f(u_2) = \max\{0.0, u_2\} = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix}$$

$$f'(u_2) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Summary

derivatives $f'(u_1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$$f'(u_2) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

2. (e) Find gradients $\nabla_{\boldsymbol{u}} J$ of the cost $\boldsymbol{u}$ with respect to activations $\boldsymbol{u}$, at the three layers. (6 marks)

Answer

Backpropagation for FFN:

Desired output $d = 0$

$y = 0.698$

Outer layer u_3,

$$\nabla_{u_3} J = -(d - y) = -(0 - 0.698) = 0.698$$

Second hidden layer u_2,

$$W_3 = \begin{pmatrix} 1.0 \\ -1.0 \\ 0.5 \end{pmatrix} \qquad f'(u_2) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\nabla_{u_2} J = W_3(\nabla_{u_3} J) \cdot f'(u_2) = \begin{pmatrix} 1.0 \\ -1.0 \\ 0.5 \end{pmatrix} (0.698) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} (1.0)(0.698) \\ (-1.0)(0.698) \\ (0.5)(0.698) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix}$$

First hidden layer u_1,

$$W_2 = \begin{pmatrix} -0.5 & 0.2 & -1.0 \\ 1.2 & 1.3 & 0.8 \end{pmatrix} \qquad f'(u_1) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\nabla_{u_1} J = W_2(\nabla_{u_2} J) \cdot f'(u_1) = \begin{pmatrix} -0.5 & 0.2 & -1.0 \\ 1.2 & 1.3 & 0.8 \end{pmatrix} \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} (-0.5)(0.698) + (0.2)(-0.698) + (-1.0)(0.349) \\ (1.2)(0.698) + (1.3)(-0.698) + (0.8)(0.349) \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0.838 \\ 0.209 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.209 \end{pmatrix}$$

Summary

$$\nabla_{u_3} J = 0.698$$

$$\nabla_{u_2} J = \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix}$$

$$\nabla_{u_1} J = \begin{pmatrix} 0 \\ 0.209 \end{pmatrix}$$

2. (f) Find gradients $\nabla_{\boldsymbol{W}} J$, and $\nabla_{\boldsymbol{b}} J$ of the cost J with respect to weights $\boldsymbol{W}$, and biases $\boldsymbol{b}$, respectively, at three layers.

(6 marks)

Answer

Outer layer u_3,

$$h_2 = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix} \quad \nabla_{\boldsymbol{u_3}} J = 0.698$$

$$\nabla_{\boldsymbol{W_3}} J = h_2 \left(\nabla_{u_3} J\right)^T = \begin{pmatrix} 3.36 \\ 3.64 \\ 2.24 \end{pmatrix} 0.698 = \begin{pmatrix} (3.36)(0.698) \\ (3.64)(0.698) \\ (2.24)(0.698) \end{pmatrix} = \begin{pmatrix} 2.35 \\ 2.54 \\ 1.56 \end{pmatrix}$$

$$\nabla_{\boldsymbol{b_3}} J = \nabla_{\boldsymbol{u_3}} J = 0.698$$

Second hidden layer u_2,

$$h_1 = \begin{pmatrix} 0 \\ 2.80 \end{pmatrix} \qquad \nabla_{\boldsymbol{u_2}} J = \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix}$$

$$\begin{aligned} \nabla_{\boldsymbol{W_2}} J = h_1 \left(\nabla_{u_2} J\right)^T &= \begin{pmatrix} 0.00 \\ 2.80 \end{pmatrix} (0.698 \quad -0.698 \quad 0.349) \\ &= \begin{pmatrix} (0.0)(0.698) & (0.0)(-0.698) & (0.0)(0.349) \\ (2.8)(0.698) & (2.8)(-0.698) & (2.8)(0.349) \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 1.95 & -1.95 & 0.978 \end{pmatrix} \end{aligned}$$

$$\nabla_{\boldsymbol{b_2}} J = \nabla_{\boldsymbol{u_2}} J = \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix}$$

First hidden layer u_1,

$$x = \begin{pmatrix} -1.0 \\ 1.5 \end{pmatrix} \qquad \nabla_{\boldsymbol{u_1}} J = \begin{pmatrix} 0 \\ 0.209 \end{pmatrix}$$

$$\nabla_{\boldsymbol{W_1}} J = x \left(\nabla_{u_1} J\right)^T = \begin{pmatrix} -1.0 \\ 1.5 \end{pmatrix} (0 \quad 0.209) = \begin{pmatrix} (-1.0)(0) & (-1.0)(0.209) \\ (1.5)(0) & (1.5)(0.209) \end{pmatrix} = \begin{pmatrix} 0 & -0.209 \\ 0 & 0.314 \end{pmatrix}$$

$$\nabla_{\boldsymbol{b_1}} J = \nabla_{\boldsymbol{u_1}} J = \begin{pmatrix} 0 \\ 0.209 \end{pmatrix}$$

Summary

$$\nabla_{\boldsymbol{W_3}} J = \begin{pmatrix} 2.35 \\ 2.54 \\ 1.56 \end{pmatrix} \qquad \nabla_{\boldsymbol{b_3}} J = 0.698$$

$$\nabla_{\boldsymbol{W_2}} J = \begin{pmatrix} 0 & 0 & 0 \\ 1.95 & -1.95 & 0.978 \end{pmatrix} \qquad \nabla_{\boldsymbol{b_2}} J = \begin{pmatrix} 0.698 \\ -0.698 \\ 0.349 \end{pmatrix}$$

$$\nabla_{\boldsymbol{W_1}} J = \begin{pmatrix} 0 & -0.209 \\ 0 & 0.314 \end{pmatrix} \qquad \nabla_{\boldsymbol{b_1}} J = \begin{pmatrix} 0 \\ 0.209 \end{pmatrix}$$

3. Figure Q3 depicts a network that consists of two convolutional layers. The size of input or output volume is represented as *D x H x W*, where *D* is the number of channels, and *H x W* is the spatial size.

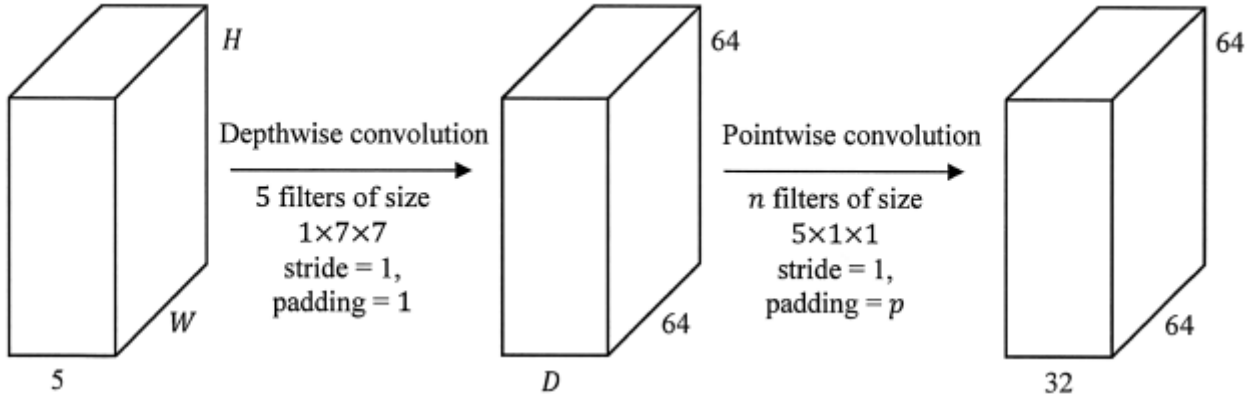


Figure Q3

3. (a) (i) Give the values of H, W, D, n and p.

(4 marks)

Answer

Depthwise convolution layer

- Output size: Dx64x64 ⇒ output = 64
- input size: 5xHxW ⇒ 5xNxN
- filter size: 1x7x7 ⇒ F = 7
- stride = 1: S = 1
- padding = 1: P = 1

$$\text{Output} = \frac{N - F + 2P}{S} + 1$$

$$64 = \frac{N - 7 + 2(1)}{1} + 1$$

$$64 = N - 4$$

$$N = 68$$

- H = W = N = 68

- Depthwise output channel = Pointwise input channel 5x1x1 ⇒ D = 5

Pointwise convolution layer

- Pointwise number of filters = output channel 32x64x64 ⇒ n = 32

- output size: 32x64x64 ⇒ output = 64
- input size: Dx64x64 ⇒ N = 64
- filter size: 5x1x1 ⇒ F = 1
- stride = 1: S = 1

$$\text{Output} = \frac{N - F + 2P}{S} + 1$$

$$64 = \frac{64 - 1 + 2(P)}{1} + 1$$

$$64 = 64 + 2P$$

$$P = 0$$

Summary

- H = 68
- W = 68
- D = 5
- N = 32
- P = 0

3. (a) (ii) Calculate the FLOPs of the depthwise convolution and pointwise convolution layers, respectively.

(4 marks)

Answer

Depthwise convolution layer

- Output size: Dx64x64 $\Rightarrow D_2 * H_2 * W_2$ $\Rightarrow H_2 = W_2 = 64$
- input size: 5xHxW $\Rightarrow D_1 * H_1 * W_1$ $\Rightarrow D_1 = 5$
- filter size: 1x7x7 $\Rightarrow$ F = 7

$$\text{FLOPs}_{\text{depthwise}} = (2 * F^2) * D_1 * H_2 * W_2 = (2 * 7^2) * 5 * 64 * 64 = 2{,}007{,}040$$

Pointwise convolution layer

- output size: 32x64x64 $\Rightarrow D_2 * H_2 * W_2$ $\Rightarrow D_2 = 32,\ H_2 = W_2 = 64$
- input size: 5x64x64 $\Rightarrow D_1 * H_1 * W_1$ $\Rightarrow D_1 = 5$
- filter size: 5x1x1 $\Rightarrow$ F = 1

$$\text{FLOPs}_{\text{pointwise}} = (2 * D_1) * D_2 * H_2 * W_2 = (2 * 5) * 32 * 64 * 64 = 1{,}310{,}720$$

3. (b) You are training a neural network and need to implement batch normalization on a given layer. For a mini-batch of size $N = 5$, you have the following input activations for a particular feature/dimension:

$$\mathbf{X} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \\ x^{(5)} \end{bmatrix} = \begin{bmatrix} 10 \\ 12 \\ 14 \\ 16 \\ 18 \end{bmatrix}$$

The batch normalization parameters are:

- Scale parameter (gamma): $\gamma = 1.5$
- Shift parameter (beta): $\beta = -2$
- Small constant for numerical stability: $\epsilon = 1 \times 10^{-5}$

Get the final output **y** of the batch normalization layer. For all calculations, round your answers to four decimal places where necessary. Show all your working steps for full credit.

(8 marks)

Answer

- mini-batch: **X** = [10, 12, 14, 16, 18]
- N = 5
- $\gamma = 1.5$
- $\beta = -2$
- $\epsilon = 1 \times 10^{-5} = 0.00001$
- Mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{5}(10 + 12 + 14 + 16 + 18) = 14$
- Variance

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$
$$= \frac{1}{5}[(10-14)^2 + (12-14)^2 + (14-14)^2 + (16-14)^2 + (18-14)^2]$$
$$= \frac{1}{5}[(-4)^2 + (-2)^2 + (0)^2 + (2)^2 + (4)^2] = 8$$

3. (b) cont

Answer

- Normalize the values

$$\hat{x}_1 = \frac{x_1 - \mu}{\sqrt{\sigma^2 + \epsilon}} = \frac{10 - 14}{\sqrt{8 + 1 \times 10^{-5}}} = -1.4142$$

$$\hat{x}_2 = \frac{x_2 - \mu}{\sqrt{\sigma^2 + \epsilon}} = \frac{12 - 14}{\sqrt{8 + 1 \times 10^{-5}}} = -0.7071$$

$$\hat{x}_3 = \frac{x_3 - \mu}{\sqrt{\sigma^2 + \epsilon}} = \frac{14 - 14}{\sqrt{8 + 1 \times 10^{-5}}} = 0$$

$$\hat{x}_4 = \frac{x_4 - \mu}{\sqrt{\sigma^2 + \epsilon}} = \frac{16 - 14}{\sqrt{8 + 1 \times 10^{-5}}} = 0.7071$$

$$\hat{x}_5 = \frac{x_5 - \mu}{\sqrt{\sigma^2 + \epsilon}} = \frac{18 - 14}{\sqrt{8 + 1 \times 10^{-5}}} = 1.4142$$

- Scale and shift the normalized values

$$\hat{y}_1 = \gamma\hat{x}_1 + \beta = (1.5)(-1.4142) - 2 = -4.1213$$

$$\hat{y}_2 = \gamma\hat{x}_2 + \beta = (1.5)(-0.7071) - 2 = -3.0607$$

$$\hat{y}_3 = \gamma\hat{x}_3 + \beta = (1.5)(0) - 2 = -2$$

$$\hat{y}_4 = \gamma\hat{x}_4 + \beta = (1.5)(0.7071) - 2 = -0.9394$$

$$\hat{y}_2 = \gamma\hat{x}_2 + \beta = (1.5)(1.4142) - 2 = 0.1213$$

<u>Summary</u>

- Mean $\mu = 14$
- Variance $\sigma^2 8 =$
- Normalize the values $\hat{x} = \begin{bmatrix} -1.4142 \\ -0.7071 \\ 0 \\ 0.7071 \\ 1.4142 \end{bmatrix}$
- Scale and shift the normalized values $\hat{y} = \begin{bmatrix} -4.1213 \\ -3.0607 \\ -2 \\ -0.9394 \\ 0.1213 \end{bmatrix}$

3. (c) Answer "TRUE" or "FALSE" to the following statements. Each sub-question carries 1 mark.

Answer

T (i) Backpropagation Through Time (BPTT) is an algorithm used to train RNNs by unrolling the network over time.

F (ii) Long Short-Term Memory (LSTM) networks are immune to the vanishing gradient problem due to their gating mechanisms.

F (iii) In a Recurrent Neural Network (RNN), the hidden state at time t is calculated by applying a nonlinear activation function to the weighted sum of the input at time t and the hidden state at time $t + 1$.

F (iv) In an RNN, using a *Tanh* activation function in the hidden layer helps mitigate the vanishing gradient problem.

T (v) The "cell state" in an LSTM can be considered as a conveyor belt that runs straight down the entire chain with only minor linear interactions.

(5 marks)

3. (c) Sparse autoencoders are designed to learn features that are robust and useful by encouraging sparsity in the hidden units. Given a dataset where each input **x** is a one-hot encoded vector, justify whether a sparse autoencoder would be appropriate for learning meaningful representations of such data. Support your answer with theoretical reasoning.

(4 marks)

Answer

- No, a sparse autoencoder is not appropriate for learning meaningful representations from one-hot encoded data.
- A sparse autoencoder encourages most hidden units to be inactive (near zero), pushing the network to learn compact and specialized features.
- However, when the input x is a one-hot encoded vector, each input already has exactly one non-zero feature, meaning the input itself is extremely sparse and distinctive.
- The goal of a sparse autoencoder is to discover hidden structure or patterns by activating only a small subset of hidden units per input.
- Theoretical reasoning:
  - One-hot encoded vectors represent categorical data with no internal structure or correlations between the categories (e.g., in a 5-class problem, category 1 and category 5 are not more "similar" than 1 and 2). Therefore, forcing sparsity in the hidden layer would not help discover meaningful features.

4. (a) Complete the following statements by filling in the blanks. These statements pertain to the self-attention mechanism in Transformers. Answers to be indicated in the answer sheet.

(i) The self-attention mechanism allows the model to weigh the significance of different tokens in the input sequence when encoding a particular token, effectively capturing long-range dependencies.

(1 mark)

(ii) In multi-head self-attention, the outputs from each attention head are concatenated and then passed through a final linear layer produce the combined output of the multi-head attention mechanism.

(2 marks)

(iii) The multi-head cross-attention in decoder takes the keys (K) and values (V) from the encoder, and takes queries (Q) from the previous layer of the decoder.

(3 marks)

(iv) The scaling factor 1/ $\sqrt{D_k}$ is essential for the effective computation of attention weights in Transformers. It prevents large dot product from causing the Softmax function to produce near-binary outputs, which would impede learning. By ensuring the attention scores are appropriately scaled, the model maintains stable gradients learns more effectively across different layers and attention heads.

(2 marks)

Answer

(i) long-range

(ii) concatenated
linear layer

(iii) keys (K)
values (V)
queries (Q)

(iv) large dot product
gradients

4. (b) Consider a scaled dot-product self-attention mechanism where you have a single query vector and multiple key and value vectors

$$\text{Query, } \mathbf{q} = [2 \quad 3], \quad \text{Key, } \mathbf{K} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \text{Value, } \mathbf{V} = \begin{bmatrix} 5 \\ 10 \\ 15 \end{bmatrix}$$

Determine the final output y = Attention(**q, K, V**). Assume $D_k = 2$. For all calculations, round your answers to four decimal places where necessary. Show all your working steps for full credit.

(14 marks)

Answer

$$\mathbf{q} = [2 \quad 3], \qquad \mathbf{K} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} 5 \\ 10 \\ 15 \end{bmatrix}$$

$$\text{Attentiion}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{QK}^{\mathrm{T}}}{\sqrt{\mathrm{D_k}}}\right)\mathbf{V}$$

$$\mathbf{QK}^{\mathrm{T}} = (2 \quad 3)\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} = ((2)(1)+(3)(0) \quad (2)(0)+(3)(1) \quad (2)(1)+(3)(1))$$
$$= (2 \quad 3 \quad 5)$$

$$\text{Softmax}\left(\frac{\mathbf{QK}^{\mathrm{T}}}{\sqrt{\mathrm{D_k}}}\right) = \text{Softmax}\left(\frac{(2 \quad 3 \quad 5)}{\sqrt{2}}\right)$$
$$= \text{Softmax}\left(\sqrt{2} \quad \frac{3}{2}\sqrt{2} \quad \frac{5}{2}\sqrt{2}\right)$$
$$= \left(\frac{e^{\sqrt{2}}}{e^{\sqrt{2}}+e^{1.5\sqrt{2}}+e^{2.5\sqrt{2}}} \quad \frac{e^{1.5\sqrt{2}}}{e^{\sqrt{2}}+e^{1.5\sqrt{2}}+e^{2.5\sqrt{2}}} \quad \frac{e^{2.5\sqrt{2}}}{e^{\sqrt{2}}+e^{1.5\sqrt{2}}+e^{2.5\sqrt{2}}}\right)$$
$$= (0.0879 \quad 0.1784 \quad 0.7337)$$

$$\text{Softmax}\left(\frac{\mathbf{QK}^{\mathrm{T}}}{\sqrt{\mathrm{D_k}}}\right)\mathbf{V} = (0.0879 \quad 0.1784 \quad 0.7337)\begin{pmatrix} 5 \\ 10 \\ 15 \end{pmatrix}$$
$$= (0.0819)(5)+(0.1784)(10)+(0.7337)(15)$$
$$= 13.199$$

4. (c) Explain the potential consequences when the discriminator in a Generative Adversarial Network (GAN) significantly outperforms the generator. How does this imbalance affect the training dynamics and the overall performance of the GAN?

(3 marks)

Answer

- When the discriminator significantly outperforms the generator in a GAN, several negative consequences arise:

- Vanishing Gradients
  - The discriminator becomes so good that it easily distinguishes real from fake data, assigning probabilities close to 0 for generated samples.
  - This causes the generator's gradients to vanish (become very small), making it difficult for the generator to learn and improve.

- Mode Collapse
  - The generator might start producing a limited variety of outputs, focusing on a few modes of the data distribution that the discriminator finds harder to distinguish.
  - This results in a lack of diversity in the generated samples.

- Training Instability:
  - The generator may struggle to learn effectively because the discriminator is too good at distinguishing real data from generated data.
  - This can lead to the generator producing poor-quality outputs, as it receives little useful feedback to improve.