SC4001 Exam 2021-2022 Sem 1

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

(i) State the shape of the following tensor [[[[2, 0, 1]], [[2, 5, 3]]]] written in Python.

Answer

- The shape = (1, 2, 1, 3).

- The outermost list contains 1 element → first dimension is 1
- That element is a list with 2 elements → second dimension is 2
- Each of those 2 elements is a list with 1 element → third dimension is 1
- That element is a list with 3 numbers → fourth dimension is 3

(ii) Consider a trained logistic neuron. Would multiplying all its weights and the bias by a constant change its accuracy? State why?

Answer

- No, would not change its accuracy.

- Accuracy depends on class predictions, not waw probabilities, the decision boundary stays the same.

(iii) State how you could use a linear neuron to learn a given nonlinear unction.

Answer

- As long as the neuron is a linear combiner followed by a non-linear activation function, then regardless of the form of non-linearity used, the neuron can perform pattern classification only on linearly separable patterns.

(iv) State one advantage each for selecting mini-batch gradient descent method over batch gradient descent, and selecting mini-batch gradient descent method over stochastic gradient descent.

Answer

- Advantage over GD: Faster convergence (computational efficiency)

- Advantage over SGD: More stable updates (reduced variance)

(v) To train a neuron layer, all the weights were initialized to 0.6. Is this good idea? Justify your answer.

Answer

- No, it is not a good idea.

- If all weights are the same, every neuron in a layer will compute the same gradient during backpropagation.

1. (a) cont

(vi) A feedforward neural network receives two-dimensional inputs, has a hidden layer comprising of 5 neurons, and has an output layer comprising of 3 neurons. What is the total number of trainable parameters (i.e., weights and biases) in the network?

Answer

Given

• layer	= 2
• hidden neuron	= 5
• outer neuron	= 3
• input	= 2
• bias	= 1

hidden layer

- hidden neuron * (input + bias) = 5 * (2 + 1) = 5 * 3 = 15 parameters

outer layer

- outer neuron * (hidden neuron + bias)
  = 3 * (5 + 1) = 3 * 6 = 18 parameters

Total = 15 + 18 = 33 parameters

(vii) State why you would choose the three-way data split method over the cross-validation method and why you would sometimes have to use the cross-validation over the three-way data split method.

Answer

- 3-way over CV:
  It requires only one split of the data, making it computationally less intensive and easier to manage, especially with large datasets.

- CV over 3-way:
  Cross-validation gives better performance estimates by training and testing the model on multiple different data splits, reducing the risk of results depending on a single lucky/unlucky split.

(14 marks)

1. (b) Figure Q1(b) 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 with a single hidden layer to classify the examples.

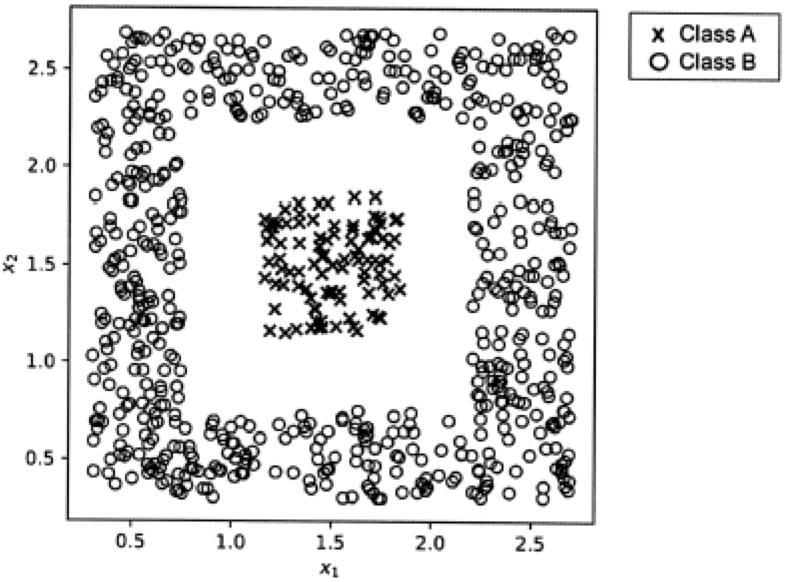
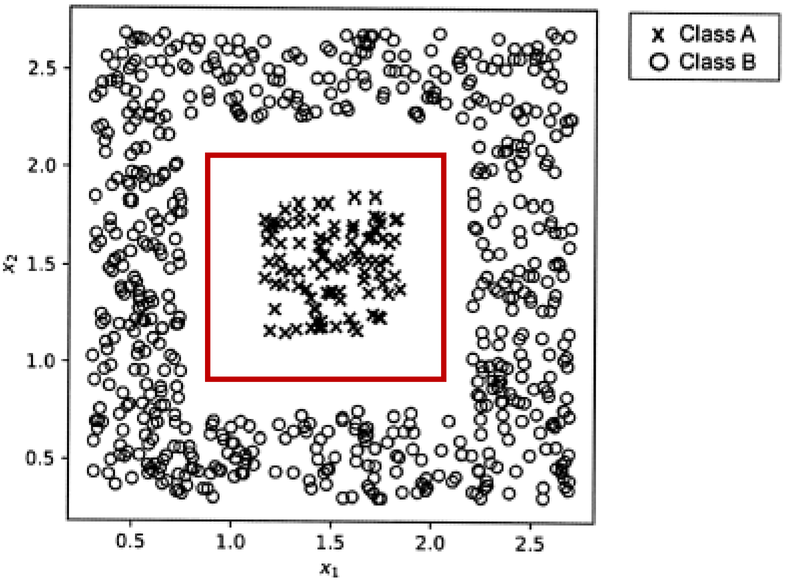


Figure Q1b

(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.

(1 marks)

Answer

Number of neurons = 4

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

(8 marks)

Answer

Line 1 (passing through (1, 2) and (2, 2))
Line 2 (passing through (2, 2) and (2, 1))
Line 3 (passing through (2, 1) and (1, 1))
Line 4 (passing through (1, 1) and (1, 2))

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

Line 1 (passing through (1, 2) and (2, 2))
$x_2 = 2$

since the shaded-region below the line,
$u_1 = x_2 - 2$

Line 2 (passing through (2, 2) and (2, 1))
$x_1 = 2$

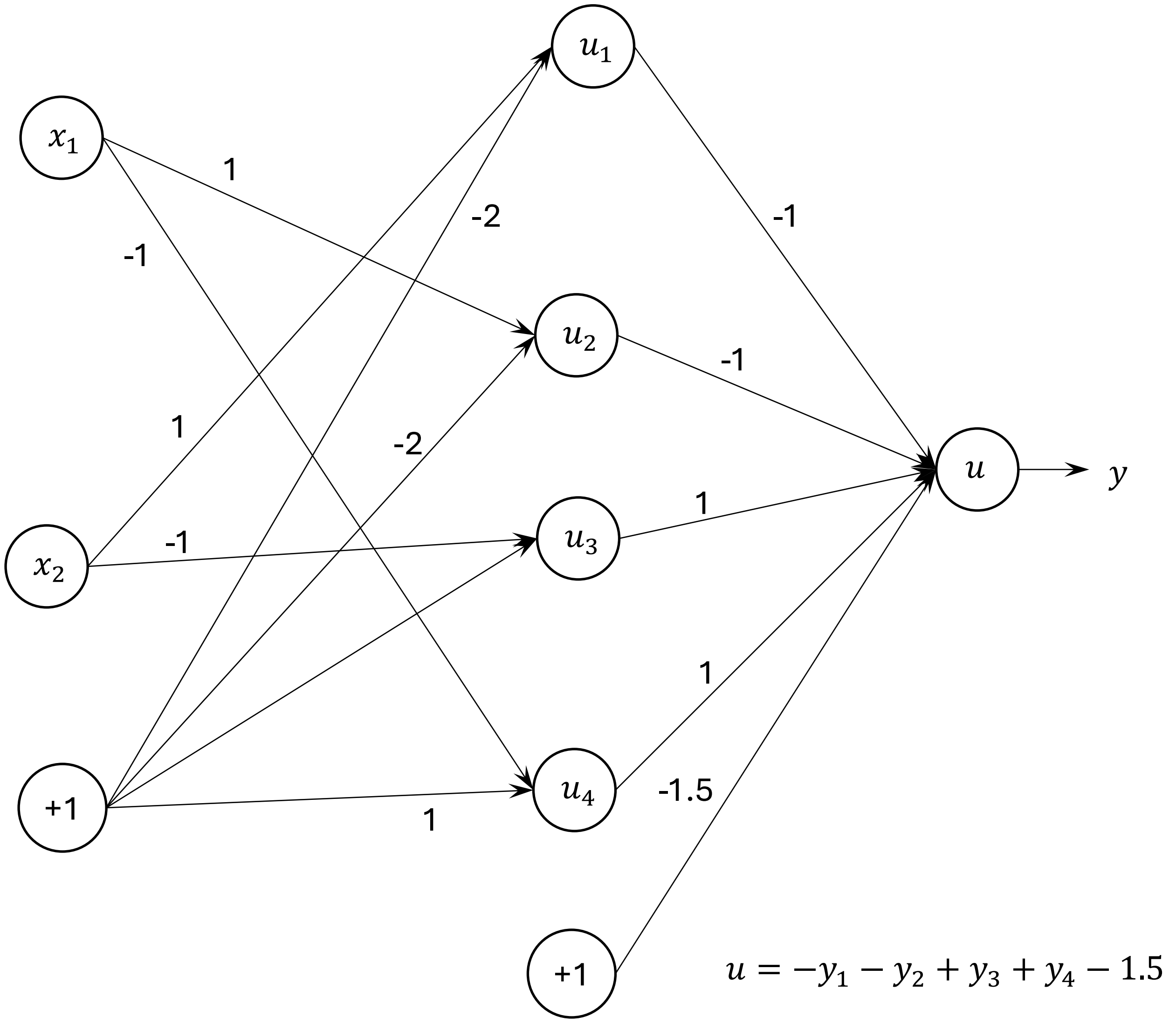
since the shaded-region below the line,
$u_2 = x_1 - 2$

Line 3 (passing through (2, 1) and (1, 1))
$x_2 = 1$

since the shaded-region below the line,
$u_3 = -x_2 + 1$

Line 4 (passing through (1, 1) and (1, 2))
$x_1 = 1$

since the shaded-region below the line,
$u_4 = -x_1 + 1$



$u = -y_1 - y_2 + y_3 + y_4 - 1.5$

2. The two-layer feedforward neural network shown in Figure Q2 receives two-dimensional inputs $(x_1, x_2) \in \boldsymbol{R}^2$ and produces an output label $y \in \{1, 2\}$. The hidden layer consists of three perceptrons and the outer layer is a *softmax* layer of two neurons. The weights of the networks are initialized as indicated in the figure and all the biases are initialized to 0.2.

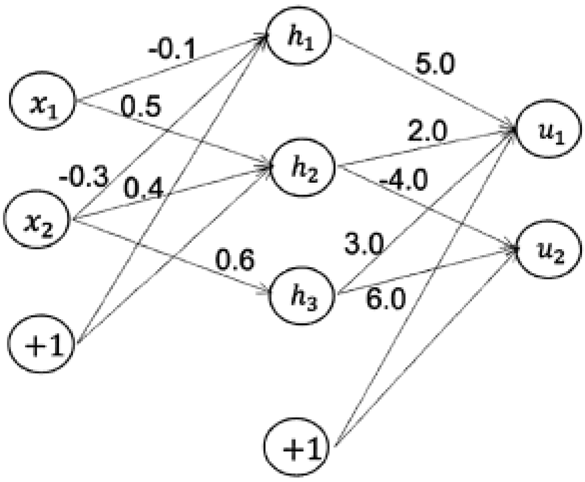


Figure Q2

The network is trained to produce a desired output $d = 2$ for an input $x = \begin{pmatrix} 0.5 \\ 2.0 \end{pmatrix}$. You are to perform one iteration of gradient descent learning with the example $(\boldsymbol{x}, d)$. The learning factor $\alpha = 0.4$. Give your answers rounded up to two decimal places.

2. (a) Write initial weight matrices $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$ of the hidden layer, and initial weight matrix $\boldsymbol{V}$ and bias vector $\boldsymbol{c}$ of the output layers.

(2 marks)

Answer

$$W = \begin{pmatrix} -0.1 & 0.5 & 0 \\ -0.3 & 0.4 & 0.6 \end{pmatrix}$$

$$b = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}$$

$$V = \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix}$$

$$c = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}$$

2. (b) Find the synaptic inputs $\boldsymbol{z}$ and output $\boldsymbol{h}$ of the hidden layer, and synaptic input $\boldsymbol{u}$ and output y of the output layer.

(5 marks)

Answer

$$W = \begin{pmatrix} -0.1 & 0.5 & 0.0 \\ -0.3 & 0.4 & 0.6 \end{pmatrix}, \quad b = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}, \quad V = \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix}, \quad c = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix} \qquad x = \begin{pmatrix} 0.5 \\ 2.0 \end{pmatrix}$$

Synaptic input to hidden-layer,

$$\boldsymbol{z} = \boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{x} + \boldsymbol{b} = \begin{pmatrix} -0.1 & -0.3 \\ 0.5 & 0.4 \\ 0.0 & 0.6 \end{pmatrix} \begin{pmatrix} 0.5 \\ 2.0 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \\ 0.0 \end{pmatrix} = \begin{pmatrix} (-0.1)(0.5) + (-0.3)(2) \\ (\ \ 0.5)(0.5) + (\ \ 0.4)(2) \\ (\ \ 0.0)(0.5) + (\ \ 0.6)(2) \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \end{pmatrix}$$

$$= \begin{pmatrix} -0.65 \\ 1.05 \\ 1.20 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \\ 0.2 \end{pmatrix} = \begin{pmatrix} -0.45 \\ 1.25 \\ 1.40 \end{pmatrix}$$

Output of the hidden layer,

$$\boldsymbol{h} = g(\boldsymbol{z}) = \frac{1}{1 + e^{-z}} = \begin{pmatrix} 0.39 \\ 0.78 \\ 0.80 \end{pmatrix}$$

Synaptic input to output-layer

$$\boldsymbol{u} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{h} + \boldsymbol{c} = \begin{pmatrix} 5.0 & 2.0 & 3.0 \\ 0.0 & -4.0 & 6.0 \end{pmatrix} \begin{pmatrix} 0.39 \\ 0.78 \\ 0.80 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix}$$

$$= \begin{pmatrix} (5)(0.39) + (\ \ 2)(0.78) + (3)(0.80) \\ (0)(0.39) + (-4)(0.78) + (6)(0.80) \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 5.91 \\ 1.68 \end{pmatrix} + \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 6.11 \\ 1.88 \end{pmatrix}$$

Output layer activation

$$y = \frac{e^{U}}{\sum_{k=1}^{K} e^{U_k}} = \begin{pmatrix} \dfrac{e^{6.11}}{e^{6.11} + e^{1.88}} \\ \dfrac{e^{1.88}}{e^{6.11} + e^{1.88}} \end{pmatrix}$$

$$= \begin{pmatrix} 0.99 \\ 0.01 \end{pmatrix} = \begin{pmatrix} \text{class 1} \\ \text{class 2} \end{pmatrix}$$

Only for softmax

Output

$$\boldsymbol{Y} = \arg\max_{k} f(\boldsymbol{U}) = (1)$$

Summary

synaptic input	$z = \begin{pmatrix} -0.45 \\ 1.25 \\ 1.20 \end{pmatrix}$
hidden layer	$h = \begin{pmatrix} 0.39 \\ 0.78 \\ 0.77 \end{pmatrix}$
synaptic input to output	$u = \begin{pmatrix} 6.02 \\ 1.70 \end{pmatrix}$
output layer	$y = \begin{pmatrix} 0.99 \\ 0.01 \end{pmatrix}$

2. (c) State the probabilities that input $\boldsymbol{x}$ belongs to each class.

(2 marks)

Answer

Since output $y = \begin{pmatrix} 0.99 \\ 0.01 \end{pmatrix} = \begin{pmatrix} \text{class 1} \\ \text{class 2} \end{pmatrix}$

- Probability that x belongs to class 1: 0.99
- Probability that x belongs to class 2: 0.01

2. (d) Find the cross-entropy cost J and classification error.

(3 marks)

Answer

Desired output $d = 2$

one-hot matrix $K = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

cross-entropy

$$\boxed{J = -\sum_{p=1}^{P} \left(\sum_{k=1}^{K} 1\left(d_p = k\right) \log\left(f\left(u_{pk}\right)\right) \right)}$$

$$\mathrm{J} = -\sum_{\mathrm{p}=1}^{8} \log\left(\mathrm{f}\left(\mathrm{u}_{\mathrm{pd}_{\mathrm{p}}}\right)\right)$$
$$= -(0 \log(y_1) + 1 \log(y_2)) = -\log(0.01) = 4.61$$

$$\boldsymbol{Y} = (1)$$

Classification error = $\sum 1(\boldsymbol{D} \neq \boldsymbol{Y}) = 1$

Summary

Cross-entropy $J \quad = 4.61$

Classification error = 1

2. (e) Find gradients $\nabla_{\boldsymbol{u}} J$ and $\nabla_{\boldsymbol{z}} J$ of the cost J with respect to $\boldsymbol{u}$ and $\boldsymbol{z}$, respectively. (7 marks)

Answer

Backpropagation for FFN:

$$d = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad y = \begin{pmatrix} 0.99 \\ 0.01 \end{pmatrix} \quad \boldsymbol{h} = \begin{pmatrix} 0.39 \\ 0.78 \\ 0.77 \end{pmatrix} \qquad V = \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix}$$

Gradient $\nabla_{\boldsymbol{u}} J$,

$$\nabla_{\boldsymbol{u}} J = -(d - y) = -\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.99 \\ 0.01 \end{pmatrix} \right) = -\begin{pmatrix} -0.99 \\ 0.99 \end{pmatrix} = \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix}$$

Gradient $\nabla_{\boldsymbol{z}} J$,

$$g'(\boldsymbol{z}) = \boldsymbol{h} \cdot (\boldsymbol{1} - \boldsymbol{h}) = \begin{pmatrix} 0.39 \\ 0.78 \\ 0.77 \end{pmatrix} \cdot \left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.39 \\ 0.78 \\ 0.80 \end{pmatrix} \right) = \begin{pmatrix} 0.39(1 - 0.39) \\ 0.78(1 - 0.78) \\ 0.80(1 - 0.80) \end{pmatrix} = \begin{pmatrix} 0.24 \\ 0.17 \\ 0.16 \end{pmatrix}$$

$$\begin{aligned} \nabla_{\boldsymbol{z}} J = V(\nabla_{\boldsymbol{u}} J) \cdot g'(\boldsymbol{z}) &= \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix} \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix} \cdot \begin{pmatrix} 0.24 \\ 0.17 \\ 0.16 \end{pmatrix} \\ &= \begin{pmatrix} (5)(0.99) + (\ \ 0)(-0.99) \\ (2)(0.99) + (-4)(-0.99) \\ (3)(0.99) + (\ \ 6)(-0.99) \end{pmatrix} \cdot \begin{pmatrix} 0.24 \\ 0.17 \\ 0.16 \end{pmatrix} \\ &= \begin{pmatrix} 4.95 \\ 5.94 \\ -2.97 \end{pmatrix} \cdot \begin{pmatrix} 0.24 \\ 0.17 \\ 0.16 \end{pmatrix} = \begin{pmatrix} 1.19 \\ 1.01 \\ -0.48 \end{pmatrix} \end{aligned}$$

Summary

Gradient $\nabla_{\boldsymbol{u}} J = \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix}$

Gradient $\nabla_{Z} J = \begin{pmatrix} 1.19 \\ 1.01 \\ -0.48 \end{pmatrix}$

2. (f) Find gradients $\nabla_V J$, $\nabla_c J$, $\nabla_W J$, and $\nabla_b J$ of the cost J with respect to $\boldsymbol{V}, \boldsymbol{c}, \boldsymbol{W}$, and $\boldsymbol{b}$, respectively.

(4 marks)

Answer

$$\nabla_{\boldsymbol{u}} J = \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix} \qquad \boldsymbol{h} = \begin{pmatrix} 0.39 \\ 0.78 \\ 0.77 \end{pmatrix}$$

$$\nabla_Z J = \begin{pmatrix} 1.19 \\ 1.01 \\ -0.48 \end{pmatrix}$$

Output layer:

$$\nabla_V J = h(\nabla_U J)^T = \begin{pmatrix} 0.39 \\ 0.78 \\ 0.80 \end{pmatrix} (0.99 \quad -0.99) = \begin{pmatrix} (0.39)(0.99) & (0.39)(-0.99) \\ (0.78)(0.99) & (0.78)(-0.99) \\ (0.80)(0.99) & (0.80)(-0.99) \end{pmatrix}$$

$$= \begin{pmatrix} 0.39 & -0.39 \\ 0.77 & -0.77 \\ 0.79 & -0.79 \end{pmatrix}$$

$$\nabla_c J = \nabla_{\boldsymbol{u}} J \quad = \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix}$$

Hidden layer:

$$x = \begin{pmatrix} 0.5 \\ 2.0 \end{pmatrix}$$

$$\nabla_W J = x(\nabla_Z J)^T = \begin{pmatrix} 0.5 \\ 2.0 \end{pmatrix} (1.19 \quad 1.01 \quad -0.53)$$

$$= \begin{pmatrix} (0.5)(1.19) & (0.5)(1.01) & (0.5)(-0.48) \\ (2.0)(1.19) & (2.0)(1.01) & (2.0)(-0.48) \end{pmatrix}$$

$$= \begin{pmatrix} 0.60 & 0.51 & -0.24 \\ 2.38 & 2.02 & -0.96 \end{pmatrix}$$

$$\nabla_b J = \nabla_Z J \quad = \begin{pmatrix} 1.19 \\ 1.01 \\ -0.48 \end{pmatrix}$$

Summary

$$\nabla_V J = \begin{pmatrix} 0.39 & -0.39 \\ 0.77 & -0.77 \\ 0.79 & -0.79 \end{pmatrix}$$

$$\nabla_c J = \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix}$$

$$\nabla_W J = \begin{pmatrix} 0.60 & 0.51 & -0.24 \\ 2.38 & 2.02 & -0.96 \end{pmatrix}$$

$$\nabla_b J \ = \begin{pmatrix} 1.19 \\ 1.01 \\ -0.48 \end{pmatrix}$$

2. (g) Find the updated values of $\boldsymbol{V}, \boldsymbol{c}, \boldsymbol{W}$, and $\boldsymbol{b}$.

(2 marks)

Answer

$$W = \begin{pmatrix} -0.1 & 0.5 & 0.0 \\ -0.3 & 0.4 & 0.6 \end{pmatrix}, \quad b = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.0 \end{pmatrix}, \quad V = \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix}, \quad c = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix} \quad \alpha = 0.4$$

$$\nabla_V J = \begin{pmatrix} 0.39 & -0.39 \\ 0.77 & -0.77 \\ 0.79 & -0.79 \end{pmatrix}$$

$$\nabla_c J = \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix}$$

$$\nabla_W J = \begin{pmatrix} 0.60 & 0.51 & -0.24 \\ 2.38 & 2.02 & -0.96 \end{pmatrix}$$

$$\nabla_b J = \begin{pmatrix} 1.19 \\ 1.01 \\ -0.48 \end{pmatrix}$$

$$\boldsymbol{V} = \boldsymbol{V} - \alpha \nabla_V J = \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix} - 0.4 \begin{pmatrix} 0.39 & -0.39 \\ 0.77 & -0.77 \\ 0.79 & -0.79 \end{pmatrix} = \begin{pmatrix} 5.0 & 0.0 \\ 2.0 & -4.0 \\ 3.0 & 6.0 \end{pmatrix} - \begin{pmatrix} 0.16 & -0.16 \\ 0.31 & -0.31 \\ 0.32 & -0.32 \end{pmatrix}$$

$$= \begin{pmatrix} 4.84 & 0.16 \\ 1.69 & -3.69 \\ 2.68 & 5.68 \end{pmatrix}$$

$$\boldsymbol{c} = \boldsymbol{c} - \alpha \nabla_c J = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix} - 0.4 \begin{pmatrix} 0.99 \\ -0.99 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.2 \end{pmatrix} - \begin{pmatrix} 0.40 \\ -0.40 \end{pmatrix} = \begin{pmatrix} -0.2 \\ 0.6 \end{pmatrix}$$

$$\boldsymbol{W} = W - \alpha \nabla_W J = \begin{pmatrix} -0.1 & 0.5 & 0.0 \\ -0.3 & 0.4 & 0.6 \end{pmatrix} - 0.4 \begin{pmatrix} 0.60 & 0.51 & -0.27 \\ 2.38 & 2.02 & -1.06 \end{pmatrix}$$

$$= \begin{pmatrix} -0.1 & 0.5 & 0.0 \\ -0.3 & 0.4 & 0.6 \end{pmatrix} - \begin{pmatrix} 0.24 & 0.20 & -0.11 \\ 0.95 & 0.81 & -0.42 \end{pmatrix} = \begin{pmatrix} -0.34 & 0.30 & 0.11 \\ -1.25 & -0.41 & 1.02 \end{pmatrix}$$

$$\boldsymbol{b} = \boldsymbol{b} - \alpha \nabla_b J = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.0 \end{pmatrix} - 0.4 \begin{pmatrix} 1.19 \\ 1.01 \\ -0.53 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.2 \\ 0.0 \end{pmatrix} - \begin{pmatrix} 0.48 \\ 0.40 \\ -0.21 \end{pmatrix} = \begin{pmatrix} -0.28 \\ -0.20 \\ 0.21 \end{pmatrix}$$

Summary

Updated $V = \begin{pmatrix} 4.84 & 0.16 \\ 1.69 & -3.69 \\ 2.68 & 5.68 \end{pmatrix}$

Updated $\boldsymbol{c} = \begin{pmatrix} -0.2 \\ 0.6 \end{pmatrix}$

Updated $W = \begin{pmatrix} -0.34 & 0.30 & 0.11 \\ -1.25 & -0.41 & 1.02 \end{pmatrix}$

Updated $\boldsymbol{b} = \begin{pmatrix} -0.28 \\ -0.20 \\ 0.21 \end{pmatrix}$

3. Given an input volume of size 3 x 225 x 225, we have 128 convolution filters each with a size of 3 x 3 x 3, a stride = 2, and no padding.

3. (a) (i) What is the output volume size?

(2 marks)

Answer

- N = 225
- F = 3
- S = 2
- P = 0

$$\text{Output } = \frac{N - F + 2P}{S} + 1 = \frac{225 - 3 + 2(0)}{2} + 1 = 111 + 1 = 112$$

- number of filters = 128

Output volume size = 128 x 112 x 112

3. (a) (ii) What is the number of parameters in this layer? Be reminded to account for the bias terms.

(2 marks)

Answer

- filter size = 3 x 3 x 3
- $d = 3$
- $F = 3$

- convolution filters = 128
- $n = 128$

Parameters per filer	= (d * F * F) + 1	= (3 * 3 * 3) + 1	= 28
Total parameters	= param * n	= 28 * 128	= 3584

Summary

Total parameters = 3584

3. (a) (iii) What is the size of output volume if depthwise convolution is performed? Assume the spatial size of a filter is 3 × 3, stride = 2, and no padding.

(2 marks)

Answer

Depthwise convolution layer

- N = 225
- F = 3
- S = 2
- P = 0

$$\text{Output } = \frac{N - F + 2P}{S} + 1 = \frac{225 - 3 + 2(0)}{2} + 1 = 111 + 1 = 112$$

- Input size 3 x 225 x 225 and filter 3 x 3 x 3, both having the same depth: 3

Depthwise convolution output volume size = 3 x 112 x 112

3. (b) (i) Give a reason why one would use a 1 × 1 convolution.

(2 marks)

Answer

- 1 × 1 convolution also known as pointwise convolution.

- Pointwise convolution can be used to change the size of channels. This can be used to achieve channel reduction and thus saving computational cost

3. (b) (ii) What would you set the padding of a two-dimensional convolutional layer to be (as a function of the filter width f) to ensure that the output has the same dimension as the input? Assume the stride is 1.

(2 marks)

Answer

In general, common to see CONV layers with stride 1, filters of size $f \times f$, and zero-padding with $\frac{f-1}{2}$. (will preserve size spatially)

e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

- S = 1

Set the padding $P = \frac{f-1}{2}$

$$\text{output } = \frac{N - f + 2P}{S} + 1$$

output = input

$$N = \frac{N - f + 2P}{1} + 1$$

$$N - 1 = N - f + 2P$$

$$2P = f - 1$$

$$P = \frac{f - 1}{2}$$

3. (b) (iii) You wish to train a convolutional neural network for classifying 12 different classes of flowers. You only have very limited training data, say 100 samples per class. Describe the steps to perform transfer learning to leverage a large-scale training dataset like ImageNet (which has 1000 classes).

(5 marks)

Answer

- Steps for transfer learning
  1. Choose a model pretrained on ImageNet (e.g., ResNet, VGG, EfficientNet).

  2. Load the pre-trained model without the final classification layer.

  3. Replace the final classification layer of the pre-trained model with a new fully connected layer from 1000 to 12 to match the flower classes.

  4. Freeze early layers (especially convolutional layers) to retain learned features and reduce overfitting.

  5. Train the model on your 12-class flower dataset, use data augmentation to increase diversity and reduce overfitting.

3. (c) An autoencoder has three neurons at the input layer and two neurons at the hidden layer. All the neurons have *sigmoid* activation functions. The weight matrix $\boldsymbol{W}$ of the hidden layer, the bias vector $\boldsymbol{b}$ of the hidden layer and the bias vector $\boldsymbol{c}$ of the output layer are given by

$$W = \begin{pmatrix} 0.2 & -0.2 \\ 0.0 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \ b = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}, \text{ and } \ c = \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix}$$

Reverse mapping from the hidden layer to the output is constrained to be the same as the input to hidden-layer mapping.

Consider the two inputs $\boldsymbol{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ and $\boldsymbol{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

3. (c) (i) Find the hidden layer activations.

(4 marks)

Answer

$$W = \begin{pmatrix} 0.2 & -0.2 \\ 0.0 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \ b = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix} \quad \boldsymbol{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \boldsymbol{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

For input $\boldsymbol{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$,

$$u_1 = W^T x_1 + b = \begin{pmatrix} 0.2 & 0.0 & 0.5 \\ -0.2 & 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} (\ \ 0.2)(1) + (\ \ 0)(0) + (0.5)(1) \\ (-0.2)(1) + (0.5)(0) + (0.5)(1) \end{pmatrix} + \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} 0.7 \\ 0.3 \end{pmatrix} + \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 1.2 \\ -0.2 \end{pmatrix}$$

$$h_1 = \sigma(u_1) = \text{sigmoid}(u_1) = \frac{1}{1 + e^{-u}} = \begin{pmatrix} 0.77 \\ 0.45 \end{pmatrix}$$

For input $\boldsymbol{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$,

$$u_2 = W^T x_2 + b = \begin{pmatrix} 0.2 & 0.0 & 0.5 \\ -0.2 & 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} (\ \ 0.2)(0) + (\ \ 0)(1) + (0.5)(1) \\ (-0.2)(0) + (0.5)(1) + (0.5)(1) \end{pmatrix} + \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} 0.5 \\ 1.0 \end{pmatrix} + \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}$$

$$h_2 = \sigma(u_2) = \text{sigmoid}(u_2) = \frac{1}{1 + e^{-u}} = \begin{pmatrix} 0.73 \\ 0.62 \end{pmatrix}$$

Summary

hidden activation for $\boldsymbol{x}_1$:

$$h_1 = \begin{pmatrix} 0.77 \\ 0.45 \end{pmatrix}$$

hidden activation for $\boldsymbol{x}_2$:

$$h_2 = \begin{pmatrix} 0.73 \\ 0.62 \end{pmatrix}$$

3. (c) (ii) Find the outputs of the autoencoder. Assume the decision threshold is 0.5.

(3 marks)

Answer

$$W = \begin{pmatrix} 0.2 & -0.2 \\ 0.0 & 0.5 \\ 0.5 & 0.5 \end{pmatrix},\ b = \begin{pmatrix} 0.5 \\ -0.5 \end{pmatrix},\ c = \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix} \quad \boldsymbol{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \boldsymbol{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

$$h_1 = \begin{pmatrix} 0.77 \\ 0.45 \end{pmatrix} \qquad h_2 = \begin{pmatrix} 0.73 \\ 0.62 \end{pmatrix}$$

For input $h_1 = \begin{pmatrix} 0.77 \\ 0.45 \end{pmatrix}$,

$$u_1 = Wh_1 + c = \begin{pmatrix} 0.2 & -0.2 \\ 0.0 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 0.77 \\ 0.45 \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} (0.2)(0.77) + (-0.2)(0.45) \\ (\ \ 0)(0.77) + (\ \ 0.5)(0.45) \\ (0.5)(0.77) + (\ \ 0.5)(0.45) \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} 0.06 \\ 0.22 \\ 0.61 \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix} = \begin{pmatrix} -0.44 \\ 0.42 \\ 0.11 \end{pmatrix}$$

$$y_1 = \sigma(u_1) = \text{sigmoid}(u) = \frac{1}{1 + e^{-u}} = \begin{pmatrix} 0.39 \\ 0.60 \\ 0.53 \end{pmatrix}$$

Apply threshold 0.5,

- $\leq$ 0.5 = 0
- > 0.5 = 1

$$y_1 = \begin{pmatrix} 0.39 \leq 0.5 \\ 0.60 > 0.5 \\ 0.53 > 0.5 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

For input $h_1 = \begin{pmatrix} 0.77 \\ 0.45 \end{pmatrix}$,

$$u_2 = Wh_2 + c = \begin{pmatrix} 0.2 & -0.2 \\ 0.0 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} 0.73 \\ 0.62 \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} (0.2)(0.73) + (-0.2)(0.62) \\ (\ \ 0)(0.73) + (\ \ 0.5)(0.62) \\ (0.5)(0.73) + (\ \ 0.5)(0.62) \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix}$$

$$= \begin{pmatrix} 0.02 \\ 0.31 \\ 0.68 \end{pmatrix} + \begin{pmatrix} -0.5 \\ 0.2 \\ -0.5 \end{pmatrix} = \begin{pmatrix} -0.48 \\ 0.51 \\ 0.18 \end{pmatrix}$$

$$y_2 = \sigma(u_2) = \text{sigmoid}(u) = \frac{1}{1 + e^{-u}} = \begin{pmatrix} 0.38 \\ 0.62 \\ 0.54 \end{pmatrix}$$

Apply threshold 0.5,

- $\leq$ 0.5 = 0
- > 0.5 = 1

$$y_2 = \begin{pmatrix} 0.38 \leq 0.5 \\ 0.62 > 0.5 \\ 0.54 > 0.5 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Summary

output for $\boldsymbol{x}_1$:

$$y_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

output for $\boldsymbol{x}_2$:

$$y_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

3. (c) (iii) Describe a way to encourage an autoencoder to learn sparse hidden structure. (3 marks)

Answer

To achieve sparse activations at the hidden-layer, the Kullback-Leibler (KL) divergence is used as the sparsity constraint:

$$D(\boldsymbol{h}) = \sum_{j=1}^{M} \rho \log \frac{\rho}{\rho_j} + (1-\rho) \log \frac{1-\rho}{1-\rho_j}$$

where M is the number of hidden neurons and ρ is the sparsity parameter.

KL divergence measures the deviation of the distribution $\{\rho_j\}$ of activations at the hidden-layer from the uniform distribution of ρ.

The KL divergence is minimum when $\rho_j = \rho$ for all j.

That is, when the average activations are uniform and equal to very low value ρ.

4. (a) Consider a Jordan-type recurrent neural network (RNN) that receives 2-dimensional input patterns $\boldsymbol{x} \in \boldsymbol{R}^2$ and has one hidden layer. The RNN has two neurons in the hidden layer (which are initialized to zeros) and one neuron in the output layer. The hidden layer neurons have $tanh$ activation functions and the output layer neurons use $sigmoid$ activation functions.

The weight matrices $\boldsymbol{U}$ connecting the input to the hidden layer, the top-down recurrence weight matrix $\boldsymbol{W}$, and $\boldsymbol{V}$ connecting the hidden output to the output layer are given by

$$\boldsymbol{U} = \begin{pmatrix} 0.4 & 0.1 \\ 0.5 & 0.4 \end{pmatrix}, \boldsymbol{W} = \begin{pmatrix} 1.0 & 0.5 \end{pmatrix} \text{ and } \boldsymbol{V} = \begin{pmatrix} 0.3 \\ -0.3 \end{pmatrix}$$

All bias connections to neurons are set to 0.1. The output layer is initialized to an output of 2.0.

Find the output of the network for an input sequence $(\boldsymbol{x}(1), \boldsymbol{x}(2), \boldsymbol{x}(3))$. Provide your answers rounded to four decimal places.

$$\boldsymbol{x}(1) = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}, x(2) = \begin{pmatrix} 0.5 \\ -1.0 \end{pmatrix} \text{ and } \boldsymbol{x}(3) = \begin{pmatrix} 2.0 \\ -2.0 \end{pmatrix}$$

(13 marks)

Answer

$$\boldsymbol{U} = \begin{pmatrix} 0.4 & 0.1 \\ 0.5 & 0.4 \end{pmatrix}, \boldsymbol{W} = \begin{pmatrix} 1.0 & 0.5 \end{pmatrix} \text{ and } \boldsymbol{V} = \begin{pmatrix} 0.3 \\ -0.3 \end{pmatrix}$$

$$\boldsymbol{x}(1) = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}, x(2) = \begin{pmatrix} 0.5 \\ -1.0 \end{pmatrix} \text{ and } \boldsymbol{x}(3) = \begin{pmatrix} 2.0 \\ -2.0 \end{pmatrix}$$

$$b = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}, y = 2.0 \text{ and } c = b = 0.1$$

$$\phi(u) = \tanh(u) \qquad = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

$$\sigma(u) = \text{sigmoid}(u) = \frac{1}{1 + e^{-u}}$$

4. (a) cont

Answer

$$\phi(u) = \tanh(u) \quad = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

$$\sigma(u) = \text{sigmoid}(u) = \frac{1}{1 + e^{-u}}$$

At $t = 1, \boldsymbol{x}(1) = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix}$,

$$\boldsymbol{h}(t) = \quad \phi(\boldsymbol{U}^T \boldsymbol{x}(t) + \boldsymbol{W}^T \boldsymbol{y}(t-1) + \boldsymbol{b})$$

$$\boldsymbol{h}(1) = tanh(\boldsymbol{U}^T \boldsymbol{x}(1) + \boldsymbol{W}^T \boldsymbol{y}(0) + \boldsymbol{b})$$

$$= \tanh\left(\begin{pmatrix} 0.4 & 0.5 \\ 0.1 & 0.4 \end{pmatrix}\begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix} + \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix} 2.0 + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right)$$

$$= \tanh\left(\begin{pmatrix} (0.4)(1) + (0.5)(0.5) \\ (0.1)(1) + (0.4)(0.5) \end{pmatrix} + \begin{pmatrix} (1.0)(2) \\ (0.5)(2) \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right)$$

$$= \tanh\left(\begin{pmatrix} 0.65 \\ 0.30 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right) = \tanh\begin{pmatrix} 2.75 \\ 1.40 \end{pmatrix} = \begin{pmatrix} 0.9919 \\ 0.8854 \end{pmatrix}$$

$$\boldsymbol{y}(t) = \qquad \sigma(\boldsymbol{V}^T \boldsymbol{h}(t) + c)$$

$$\boldsymbol{y}(1) = sigmoid(\boldsymbol{V}^T \boldsymbol{h}(1) + c)$$

$$= \text{sigmoid}\left((0.3 \quad -0.3)\begin{pmatrix} 0.9919 \\ 0.8854 \end{pmatrix} + 0.1\right)$$

$$= \text{sigmoid}((0.3)(0.9919) + (-0.3)(0.8854) + 0.1)$$

$$= \text{sigmoid}(0.1320) = 0.5330$$

At $t = 2, \boldsymbol{x}(2) = \begin{pmatrix} 0.5 \\ -1.0 \end{pmatrix}$,

$$\boldsymbol{h}(t) = \quad \phi(\boldsymbol{U}^T \boldsymbol{x}(t) + \boldsymbol{W}^T \boldsymbol{y}(t-1) + \boldsymbol{b})$$

$$\boldsymbol{h}(2) = tanh(\boldsymbol{U}^T \boldsymbol{x}(2) + \boldsymbol{W}^T \boldsymbol{y}(1) + \boldsymbol{b})$$

$$= \tanh\left(\begin{pmatrix} 0.4 & 0.5 \\ 0.1 & 0.4 \end{pmatrix}\begin{pmatrix} 0.5 \\ -1.0 \end{pmatrix} + \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix} 0.5330 + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right)$$

$$= \tanh\left(\begin{pmatrix} (0.4)(0.5) + (0.5)(-1) \\ (0.1)(0.5) + (0.4)(-1) \end{pmatrix} + \begin{pmatrix} (1.0)(0.5330) \\ (0.5)(0.5330) \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right)$$

$$= \tanh\left(\begin{pmatrix} -0.30 \\ -0.35 \end{pmatrix} + \begin{pmatrix} 0.5330 \\ 0.2665 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right) = \tanh\begin{pmatrix} 0.333 \\ 0.015 \end{pmatrix} = \begin{pmatrix} 0.3212 \\ 0.0150 \end{pmatrix}$$

$$\boldsymbol{y}(t) = \qquad \sigma(\boldsymbol{V}^T \boldsymbol{h}(t) + c)$$

$$\boldsymbol{y}(2) = sigmoid(\boldsymbol{V}^T \boldsymbol{h}(2) + c)$$

$$= \text{sigmoid}\left((0.3 \quad -0.3)\begin{pmatrix} 0.3212 \\ 0.0150 \end{pmatrix} + 0.1\right)$$

$$= \text{sigmoid}((0.3)(0.3212) + (-0.3)(0.015) + 0.1)$$

$$= \text{sigmoid}(0.1919) = 0.5478$$

4. (a) cont

Answer

$$\phi(u) = \tanh(u) \quad = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

$$\sigma(u) = \text{sigmoid}(u) = \frac{1}{1 + e^{-u}}$$

At $t = 3, \boldsymbol{x}(3) = \begin{pmatrix} 2.0 \\ -2.0 \end{pmatrix}$,

$$\boldsymbol{h}(t) = \quad \phi(\boldsymbol{U}^T\boldsymbol{x}(t) + \boldsymbol{W}^T\boldsymbol{y}(t-1) + \boldsymbol{b})$$

$$\boldsymbol{h}(2) = tanh(\boldsymbol{U}^T\boldsymbol{x}(3) + \boldsymbol{W}^T\boldsymbol{y}(2) + \boldsymbol{b})$$

$$= \tanh\left(\begin{pmatrix} 0.4 & 0.5 \\ 0.1 & 0.4 \end{pmatrix}\begin{pmatrix} 2.0 \\ -2.0 \end{pmatrix} + \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix} 0.5478 + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right)$$

$$= \tanh\left(\begin{pmatrix} (0.4)(2) + (0.5)(-2) \\ (0.1)(2) + (0.4)(-2) \end{pmatrix} + \begin{pmatrix} (1.0)(0.5478) \\ (0.5)(0.5478) \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right)$$

$$= \tanh\left(\begin{pmatrix} -0.2 \\ -0.6 \end{pmatrix} + \begin{pmatrix} 0.5478 \\ 0.2739 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}\right) = \tanh\begin{pmatrix} 0.4478 \\ -0.2261 \end{pmatrix} = \begin{pmatrix} 0.4201 \\ -0.2223 \end{pmatrix}$$

$$\boldsymbol{y}(t) = \quad \sigma(\boldsymbol{V}^T\boldsymbol{h}(t) + c)$$

$$\boldsymbol{y}(3) = sigmoid(\boldsymbol{V}^T\boldsymbol{h}(3) + c)$$

$$= \text{sigmoid}\left((0.3 \quad -0.3)\begin{pmatrix} 0.4201 \\ -0.2223 \end{pmatrix} + 0.1\right)$$

$$= \text{sigmoid}((0.3)(0.4201) + (-0.3)(-0.2223) + 0.1)$$

$$= \text{sigmoid}(0.2927) = 0.5727$$

Summary

- Output of the network:
  - y(1) = 0.5330
  - y(2) = 0.5478
  - y(3) = 0.5727

4. (b) Answer "TRUE" or "FALSE" to the following statements:

Answer

T (i) Layer normalization is usually used in Transformers.

F (ii) Positional encoding helps to stabilize the training of Transformers.

T (iii) Self-attention is performed in the decoder of Transformers.

F (iv) Transformer model pays attention to a single most important word in a sentence.

F (v) The feedforward network in the self-attention layer is not applied independently to each position.

T (vi) Positional encoding has the same dimension as the input embedding.

(6 marks)

4. (c) Explain an advantage of attention-based models over recurrent-based ones.

(3 marks)

Answer

- Attention-based models allow modelling of dependencies without regard to their distance in the input or output sequences

- Attention-based models can process all elements of the input sequence simultaneously, rather than sequentially.

- This allows for faster training and inference times, especially with long sequences.

4. (c) Describe a method to remedy mode collapse during training of Generative Adversarial Networks (GANs).

(3 marks)

Answer

- Mini-batch discrimination allows the discriminator to look at multiple samples in a batch simultaneously instead of evaluating each sample independently.

- This helps the discriminator detect lack of diversity in the generator’s outputs, which is a key symptom of mode collapse.