studocu

QPS new - question paper

BITS WILP M Tech Data Science & Engineering (Birla Institute of Technology and Science, Pilani)

Scan to open on Studocu

Studocu is not sponsored or endorsed by any college or university
Downloaded by A S (shuklarchana001@gmail.com)

Birla Institute of Technology and Science, Pilani

Work Integrated Learning Programmes Division

M. Tech. in AI & ML

I Semester 2023-2024

End-Semester Test
(EC3 - Regular)

Course Number	AIMLCZG511
Course Name	DEEP NEURAL NETWORK
Nature of Exam	Closed Book
Weight-age for grading	30
Duration	2.5 hrs

# Pages	2
# Questions	5

Instructions

1. Read the question carefully and answer to the point. All answers must be directed to the question in bullet points.

2. Use the correct and exact keywords or nomenclature in your answer.

3. Assumptions made if any, should be stated clearly at the beginning of your answer.

1. Compute the dimensions of W^Q, W^K, W^V, W^O if the input dimensions and attention dimensions are given as $d = 2048$ and the query and key dimensions are $d_k = 512$ and value dimensions are $d_v = 1024$ and the number of head $h = 4$. Compute the dimensions of the attention vector. [5]

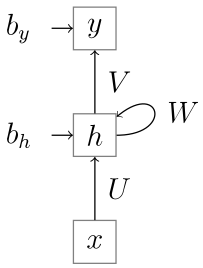
   One mark each

$$\begin{aligned}
\text{Dimensions of } W^Q &= (h \times d) \times d_k = (4 \times 2048) \times 512 \\
\text{Dimensions of } W^K &= (h \times d) \times d_k = (4 \times 2048) \times 512 \\
\text{Dimensions of } W^V &= (h \times d) \times d_v = (4 \times 2048) \times 1024 \\
\text{Dimension of attention vector} &= d_v = 4 \times 1024 \\
\text{Output Projection Matrix} W^O &= h \times d_v \times d = 4 \times 1024 \times 2048
\end{aligned}$$

2. Design a single hidden layer recurrent neural network that outputs the moving sum of difference of two input real sequences. For example, left-to-right input sequences $p = \{0.6, 0.4, 0.3, 1.2, -1.2\}$ and $q = \{-1.2, 1.2, 0.3, 0.4, 0.6\}$ will produce the left-to-right output sequence $\{1.8, 1.0, 1.0, 1.8, 0\}$. All nodes use linear activation function $g(x)$, i.e., $g(x) = x$. There are two input unit corresponding to two input sequences and one output unit. Hidden states are denoted by h, as shown in the following figure. Note, at time t,

$$h_t = g(Wh_{t-1} + Ux - b_h) \text{ and } y_t = g(vh_t - b_y)$$

Downloaded by A S (shuklarchana001@gmail.com)

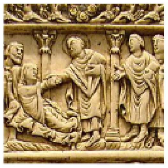


(a) What is the minimum number of hidden nodes required? Justify [1]
minimum number of hidden nodes = 1

(b) Compute W, U, V, b_h and b_y. Only integer values are allowed. Show all computations. [5]
One mark each
$b_h = 0; b_y = 0; U = [1 - 1]; W = 1; V = 1$

3. Two historians approach you for your deep learning expertise. They want to classify images of historical objects into 3 classes depending on the time they were created: Antiquity ($y = 0$), Middle Ages ($y = 1$) and Modern Era ($y = 2$)

(A) Class: Antiquity


(B) Class: Middle Ages

(C) Class: Modern Era

You come up with a CNN classifier. For each layer, calculate the number of weights, number of biases and the size of the associated feature maps. [8]
The notation follows the convention:

- CONV-K-N denotes a convolutional layer with N filters, each them of size KxK, Padding and stride parameters are always 0 and 1 respectively.
- POOL-K indicates a KxK pooling layer with stride K and padding 0.
- FC-N stands for a fully-connected layer with N neurons.

Layer	Activation map dimensions	Number of weights	Number of biases
INPUT	128 x 128 x 3	0	0
CONV-9-32			
POOL-2			
CONV-5-64			
POOL-2			
CONV-5-64			
POOL-2			
FC-64			
FC-3			

Downloaded by A S (shuklarchana001@gmail.com)

One mark each

Layer	Activation map dimensions	Number of weights	Number of biases
INPUT	128 x 128 x 3	0	0
CONV-9-32	120 x 120 x 32	7776	32
POOL-2	60 x 60 x 32	0	0
CONV-5-64	56 x 56 x 64	51200	64
POOL-2	28 x 28 x 64	0	0
CONV-5-64	24 x 24 x 64	102400	64
POOL-2	12 x 12 x 64	0	0
FC-64	64	589824	64
FC-3	3	192	3

4. A deep learning researcher is designing an LSTM-based neural network for a natural language processing task. The researcher plans to use a single-layer LSTM with specific architectural parameters. The LSTM layer is expected to have an input size of 2^{10} and a hidden size of 2^8. Additionally, the researcher decides not to use bidirectionality in the LSTM layer. The expected output is a 2^{12} vector. Based on these specifications, calculate the total number of parameters required for this LSTM layer. Show Gate wise necessary calculations. [5]
One mark each

$$\begin{aligned}
\text{Parameters for cell state} &= n_i \times n_h + n_h \times n_h + n_h = 2^{10} \times 2^8 + 2^8 \times 2^8 + 2^8 \\
\text{Parameters for input gate} &= n_i \times n_h + n_h \times n_h + n_h = 2^{10} \times 2^8 + 2^8 \times 2^8 + 2^8 \\
\text{Parameters for forget gate} &= n_i \times n_h + n_h \times n_h + n_h = 2^{10} \times 2^8 + 2^8 \times 2^8 + 2^8 \\
\text{Parameters for output gate} &= n_i \times n_h + n_h \times n_h + n_h = 2^{10} \times 2^8 + 2^8 \times 2^8 + 2^8 \\
\text{Parameters for output projection} &= n_h \times n_o + n_o = 2^8 \times 2^{12} + 2^{12}
\end{aligned}$$

5. Consider a simplified version of the NiN architecture with one NiN block followed by global average pooling and a softmax output layer. Each NiN block consists of the following layers: 1x1 convolutional layer with 20 filters, three separate 3x3 convolutional layers, each with 50 filters, two separate 5x5 convolutional layers with 20 filters, ReLU activation function applied after each convolutional layer. Assume that the input to the NiN architecture is a greyscale image with dimensions of 256x256 pixels. Calculate the following:

   (a) The number of parameters (weights and biases) in one NiN block. [4]

   (b) The dimensions of the output feature map after the NiN block. [1]

   (c) The total number of parameters in the entire NiN architecture. [1]

Rubric

- Number of parameters in one NiN block:

$$\begin{aligned}
\text{1x1 Convolutional layer } Weights + bias &= 1 \times 1 \times 1 \times 20 + 20 = 40 && [1] \\
\text{3x3 Convolutional layer } Weights + bias &= 3 \times 3 \times 20 \times 50 + 50 = 9050 \\
\text{For three 3x3 Convolutional layers } &= 3 \times 9050 = 27150 && [1] \\
\text{5x5 Convolutional layer } Weights + bias &= 5 \times 5 \times 20 \times 20 + 20 = 10020 \\
\text{For two 5x5 Convolutional layers } &= 2 \times 10020 = 20040 && [1] \\
\text{Total parameters in onw NiN block } &= 40 + 27150 + 20040 = 47230 && [1]
\end{aligned}$$

Downloaded by A S (shuklarchana001@gmail.com)

- Dimensions of the output feature map after the NiN block: Assume that the convolutional layers use "same" padding. Output feature map after the NiN block will be of size 256x256. [1]
- Since the NiN architecture consists of one NiN block followed by global average pooling and a softmax output layer, the NiN block contains 47230 parameters. [1]

Downloaded by A S (shuklarchana001@gmail.com)

Birla Institute of Technology and Science, Pilani

Work Integrated Learning Programmes Division

M. Tech. in AI & ML

II Semester 2022-2023

End-Semester Test
(EC3 - Regular)

Course Number	AIMLCZG51
Course Name	DEEP NEURAL NETWORK
Nature of Exam	Open Book
Weight-age for grading	40
Duration	2.5 hrs
Date of Exam	

* Pages	4
* Questions	4

1. (a) Consider a CNN architecture with an input image of size $256 \times 256 \times 3$. The architecture consists of two convolutional layers with 64 and 128 filters of size 5×5 respectively, followed by a max-pooling layer of size 2×2. Calculate the output dimensions after each layer. [3]

   (b) If the input image size is changed to $128 \times 128 \times 3$, how will it affect the number of trainable parameters? [1]

   (c) Compute the trainable parameters are there in the second convolutional layer for part (a). [1]

   (d) Write Python code to implement for part (a) architecture using TensorFlow Keras. Add one dense layer and an output layer for binary classification. [3]

   (e) I represents the top left corner pixel values of a much larger image (not shown in entirety). F is a filter that convolves over this image with stride 1. O is the result of this convolution, where only 3 resulting pixels at top left are shown. Importantly, there is no zero padding, which implies that each pixel in O is the dot product resulting from the filter being placed at the corresponding position on the image in I. Find the values of x, y and z. [3]

$$I = \begin{pmatrix} 3 & 0 & 3 & -3 & 0 & \dots \\ -3 & 2 & 0 & 3 & -2 & \dots \\ -5 & 0 & 3 & -2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \qquad F = \begin{pmatrix} x & y & z \\ 1 & 0 & 2 \\ -1 & 1 & 0 \end{pmatrix} \qquad O = \begin{pmatrix} 4 & 0 & -3 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

(Note: Your answer must be supported by the calculation / justification / reasoning for all parts of this question. There are no marks for the direct answer or the output of any tool or program or for direct screenshots or images of textbook pages.)

Rubrics and one solution:

(a)
- First Convolutional Layer (64 filters): [1 mark]
  Output dimensions: $(256 - 5 + 1)x(256 - 5 + 1)x64 = 252 \times 252 \times 64$
- Second Convolutional Layer (128 filters): [1 mark]
  Output dimensions: $(252 - 5 + 1)x(252 - 5 + 1)x64 = 248 \times 248 \times 128$

Downloaded by A S (shuklarchana001@gmail.com)

- Max-Pooling Layer (2x2): [1 mark]
  Output dimensions: $248/2x248/2x128 = 124 \times 124 \times 128$

(b)
- Number of Parameters per Filter will remain the same, since the filter size (5x5) and the number of input channels (3) remain the same. [0.5 mark]
- The number of filters in each layer (64 and 128) remains the same. [0.5 mark]

(c) Number of Parameters = (Filter Size * Number of Input Channels+1) * Number of Filters $= (5 \times 5 \times 64 + 1) \times 128 = 204,928$ parameters. [1 mark]

(d) 1 mark for each layer. If sequential model not declared and created give 0 marks for the entire answer.

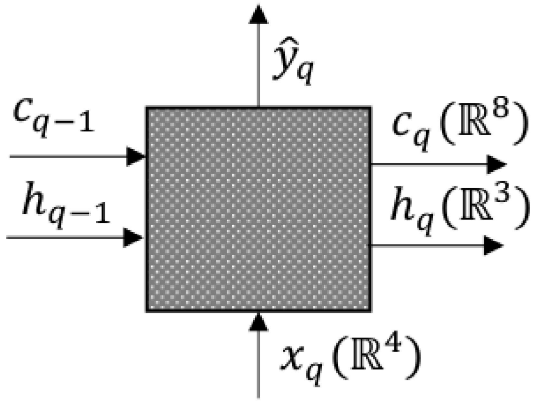
```
# Define the CNN model
model = models.Sequential()
# First Convolutional Layer
model.add(layers.Conv2D(64, (5, 5), activation='relu',
input_shape=(256, 256, 3)))
# Second Convolutional Layer
model.add(layers.Conv2D(128, (5, 5), activation='relu'))
# Max pooling Layer
model.add(layers.MaxPooling2D((2, 2)))
```

(e) 3 marks for correct equations and answers. No step marking.

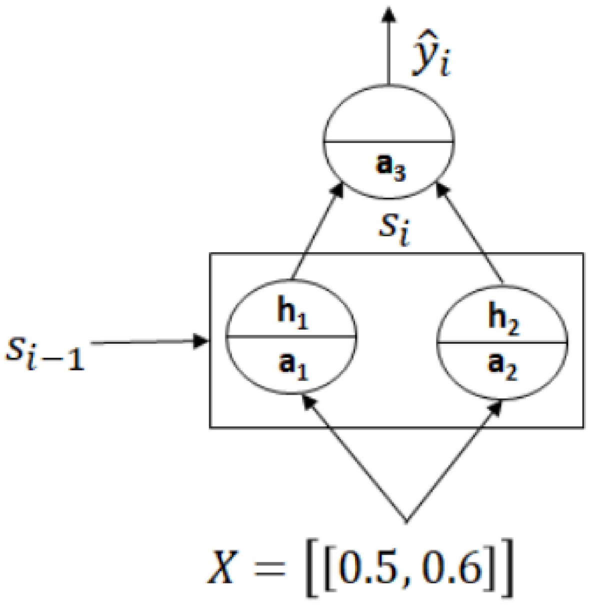
$$\begin{aligned} \text{equations:} \quad 3x + 3y &= 12 \\ 3y - 3z &= -11 \\ 3x - 3y &= 6 \\ \text{solving:} \quad x &= 3 \\ y &= 1 \\ z &= 14/3 \end{aligned}$$

Downloaded by A S (shuklarchana001@gmail.com)

2. (a) A data science engineer proposed an LSTM network (shown below) for a deep learning problem. Calculate the parameters to be learnt in it. The dimensions are also shown in the parentheses. Output is one dimensional. [5]



(b) A RNN is shown below with one RNN layer of two tanh neurons and a fully connected output layer with one sigmoid neuron. A record of input with one time-step and two input features is processed with it. Calculate the output of the network assuming all weights and biases are initialized with value 0.2 and previous state as [0.3, 0.4]. The arrows shown in the diagram are representative. [5]



(Note: Your answer must be supported by the calculation / justification / reasoning for all parts of this question. There are no marks for the direct answer or the output of any tool or program or for direct screenshots or images of textbook pages.)

(a)
- Input (x_t): R^4
- Hidden state (h_t): R^3
- Cell state (c_t): R^8
- Output (y_t): R^1
- Input Gate (i_t)

Downloaded by A S (shuklarchana001@gmail.com)

- Weight matrix for input gate (W_{xi}): $R^{4\times3}$
- Weight matrix for hidden state (W_{hi}): $R^{3\times3}$
- Bias for input gate (b_i): R^3

- Forget Gate (f_t)
  - Weight matrix for input gate (W_{xf}): $R^{4\times3}$
  - Weight matrix for hidden state (W_{hf}): $R^{3\times3}$
  - Bias for forget gate (b_f): R^3
- Output Gate (f_t)
  - Weight matrix for input gate (W_{xo}): $R^{4\times3}$
  - Weight matrix for hidden state (W_{ho}): $R^{3\times3}$
  - Bias for Output gate (b_o): R^3
- Cell State (c_t)
  - Weight matrix for input gate(W_{xc}): $R^{4\times8}$
  - Weight matrix for hidden state (W_{hc}): $R^{3\times8}$
  - Bias for cell state (b_c): R^8
- Output (y_t)
  - Weight matrix for output (W_{hy}): $R^{3\times1}$
  - Bias for output (b_y): R^1
- Total= $12+9+3+12+9+3+12+9+3+32+24+8+3+1 = 140$ parameters

(b) Two tanh neurons in the RNN layer. One sigmoid neuron in the output layer. Input features: [0.5, 0.6] Initial state: [0.3, 0.4]. All weights and biases are initialized with a value of 0.2. $W_{hh} = [0.2, 0.2]; (W_{xh} = [0.2, 0.2]; b_h = [0.2, 0.2]$

- Calculate the Hidden State (RNN Layer)

$$\begin{aligned}
h_t &= \tanh(W_{hh} \times h_{t-1} + W_{xh} \times x_t + b_h)\\
h_1 &= \tanh([0.2, 0.2] \times [0.3, 0.4] + [0.2, 0.2] \times [0.5, 0.6] + [0.2, 0.2])\\
h_1 &= \tanh([0.06, 0.08] + [0.1, 0.12] + [0.2, 0.2])\\
h_1 &= [\tanh(0.06), \tanh(0.08)]\\
h_1 &\approx [0.059964, 0.079995]
\end{aligned}$$

- Calculate the Output (Output Layer)

$$\begin{aligned}
y &= \sigma(W_{hy} \times h_t + b_y)\\
y &= \sigma([0.2, 0.2] \times [0.059964, 0.079995] + 0.2)\\
&= \sigma(0.025993 + 0.2) \approx 0.556275
\end{aligned}$$

Downloaded by A S (shuklarchana001@gmail.com)

3. (a) Consider a 1D CNN model for time series forecasting with a sequence length of 100. The model has a single convolutional layer with 64 filters, each of size 3, followed by a fully connected layer with 128 neurons. Batch normalization is applied after convolution layer. Calculate the total number of trainable parameters and non-trainable parameters in the model. [3]

(b) Consider a Deep Neural Network with 3 hidden layers for time series forecasting. The input sequence has a length of 50, and the hidden layers have 64, 128, and 256 neurons, respectively. Calculate the total number of trainable parameters in the model. Assume the output sequence length. [3]

(c) A data scientist wants to build a Deep Learning model using a single LSTM cell for forecasting a time-series. The training dataset has several records with 15 time steps each. Each time step consists of three-feature normalized numeric data. If total learnable parameters in this model are 180, what is the dimensionality of the short state? [3]

(Note: Your answer must be supported by the calculation / justification / reasoning for all parts of this question. There are no marks for the direct answer or the output of any tool or program or for direct screenshots or images of textbook pages.)

(a)
- Convolutional Layer: $64\, filters \times (3\, weights + 1\, bias) = 256$ parameters.
- Batch Normalization
  Trainable parameters: 64 filters × (2 parameters) = 128 trainable parameters
  Non-trainable parameters: 64 filters × (2 parameters) = 128 non-trainable
- Fully connected layer: $128 neurons \times (6400 weights + 1 bias) = 819,328$ trainable parameters.
- Total Trainable Parameters = 256 (Convolution) + 128 (Batch Norm) + 819, 328 (Fully Connected) = 819, 712
  Total Non-Trainable Parameters = 128 (Batch Norm)

(b)
- Input Layer: 0 trainable parameters
- First hidden layer: $50 \times 64 + 64 = 326450 \times 64 + 64 = 3264$ parameters
- Second hidden layer: $(64 \times 128) + 128 = 8320(64 \times 128) + 128 = 8320$ parameters
- Third hidden layer: $(128\times256)+256 = 33024(128\times256)+256 = 33024$ parameters
- Output Layer: Assume it has 1 neuron $(256 \times 1) + 1 = 257(256 \times 1) + 1 = 257$ parameters
- Total $= 3264 + 8320 + 33024 + 257 = 44705$ parameters

(c) Input dimension (D) is 3.
Weights for Input have the shape (D, H). Weights for Hidden State have the shape (H, H). Biases (b) has the shape (H,1).For each gate, $(D * H) + (H * H) + H$ parameters. For all gates and the cell state, there are $4 * [(D * H) + (H * H) + H]$

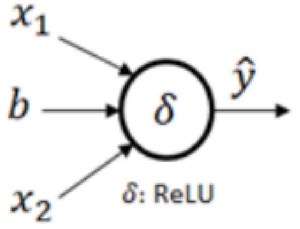
$$4 * [(D * H) + (H * H) + H] = 180$$
$$4 * [(3 * H) + (H * H) + H] = 180$$
$$3H^2 + 7H - 45 = 0$$
$$H = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = 2$$

Dimensionality of the hidden state (H) is approximately 2.

Downloaded by A S (shuklarchana001@gmail.com)

4. (a) The training data $(x_1, y_1) = (3.5, 0.5)$ for a single Sigmoid neuron and initial values of $w = -2.0, b = -2.0, \eta = 0.10, \beta_1 = 0.90, \beta_2 = 0.99, \epsilon = 1e-8, s_W = 0$ and $s_B = 0$ are provided. Showing all the calculations, find out the values of w, b, s_W, s_B, r_W, r_B after one iteration of Adam. Use BCE as loss function. Apply bias-correction. [8]

(b) For the training data $[x_1, x_2, y] = [0.65, 0.75, 2.50]$, an engineer selected an initial set of parameters for the single neuron as shown below. Assuming SSE loss function, evaluate his selection of parameters mathematically with proper calculations and reasoning. [3]



w₁	w₂	b	η
0.30	0.50	1.75	2.00

(Note: Your answer must be supported by the calculation / justification / reasoning for all parts of this question. There are no marks for the direct answer or the output of any tool or program or for direct screenshots or images of textbook pages.)

(a)
- Forward Pass and Calculate Loss

$$\hat{y} = \sigma(wx + b) = \sigma(-2.0 \times 3.5 - 2.0) \approx 0.0180$$
$$L = -[y \times \log(\hat{y}) + (1 - y) \times \log(1 - \hat{y})$$
$$L = -[0.5 \times \log(0.0180) + (1 - 0.5) \times \log(1 - 0.0180)] \approx 4.0076$$

- Calculate Gradients

$$\nabla_w L = (\hat{y} - y) \times x = (0.0180 - 0.5) \times 3.5 \approx -0.493$$
$$\nabla_b L = \hat{y} - y = 0.0180 - 0.5 \approx -0.482$$

- Update First Moments

$$s_W = \beta_1 \times s_W + (1 - \beta_1) \times \nabla_w L = 0.90 \times 0 - 0.10 \times (-0.493) \approx 0.04437$$
$$s_B = \beta_1 \times s_B + (1 - \beta_1) \times \nabla_b L = 0.90 \times 0 - 0.10 \times (-0.482) \approx 0.04382$$

- Update Second Moments

$$r_W = \beta_2 \times r_W + (1 - \beta_2) \times (\nabla_w L)^2 = 0.999 \times 0 - 0.001 \times (-0.493)^2 \approx 0.0001216$$
$$r_B = \beta_2 \times r_B + (1 - \beta_2) \times (\nabla_b L)^2 = 0.999 \times 0 - 0.001 \times (-0.482)^2 \approx 0.0001168$$

- Corrected First Moments

$$\hat{s}_W = \frac{s_W}{1 - \beta_1^1} \approx \frac{0.04437}{1 - 0.90^1} \approx 0.4437$$
$$\hat{s}_B = \frac{s_B}{1 - \beta_1^1} \approx \frac{0.04382}{1 - 0.90^1} \approx 0.4382$$

- Corrected Second Moments

$$\hat{r}_W = \frac{r_W}{1 - \beta_2^1} \approx \frac{0.0001216}{1 - 0.999^1} \approx 0.0608$$
$$\hat{r}_B = \frac{r_B}{1 - \beta_2^1} \approx \frac{0.0001168}{1 - 0.999^1} \approx 0.0584$$

Downloaded by A S (shuklarchana001@gmail.com)

- Update Weights and Biases

$$w = w - \frac{\eta}{\sqrt{\hat{r}_W} + \epsilon} \times \hat{s}_W = -2.0 - \frac{0.10}{\sqrt{0.0608} + 1e - 8} \times 0.4437 \approx -2.081$$
$$b = b - \frac{\eta}{\sqrt{\hat{r}_B} + \epsilon} \times \hat{s}_B = -2.0 - \frac{0.10}{\sqrt{0.0584} + 1e - 8} \times 0.4382 \approx -2.079$$

(b) Training data: $[x_1, x_2, y] = [0.65, 0.75, 2.50]$ Initial parameters: $w_1 = 0.3$, $w_2 = 0.5$, $b = 1.75$ Learning rate: $\eta = 2$

- Calculate the Predicted Output

$$\hat{y} = \sigma(w_1 x_1 + w_2 x_2 + b)$$
$$\hat{y} = \sigma(0.3 \times 0.65 + 0.5 \times 0.75 + 1.75) = \sigma(2.32) \approx 0.9106$$

- Calculate the Sum of Squared Errors (SSE) loss

$$\text{SSE} = \frac{1}{2}(\hat{y} - y)^2 = \frac{1}{2}(0.9106 - 2.50)^2 = \frac{1}{2}(-1.5894)^2 = \frac{1}{2} \times 2.5245 = 1.2623$$

To improve the model's performance, the engineer may need to adjust the weights and bias using a learning algorithm (e.g., gradient descent) to minimize the SSE loss.

Downloaded by A S (shuklarchana001@gmail.com)