2022 DNN EndSem Regular

1. (a) Consider a CNN architecture with an input image of size $128 \times 128 \times 3$. The architecture consists of two convolutional layers with 32 and 64 filters of size 3×3 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 64×64, how will it affect the number of trainable parameters? [1]

   (c) Compute the trainable parameters are there in the first convolutional layer [1]

   (d) Write Python code to implement this architecture using TensorFlow Keras. [3]

   (e) Given a 3x3 input matrix I and and a 2x2 filter F

$$I = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 2 & 3 & 4 \end{pmatrix} \qquad F = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

   Compute the output matrix after applying the convolution operation. If your BITS ID is odd use a stride of 1 otherwise use a stride of 2. Assume zero-padding. [2]
   (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 (32 filters): [1 mark]
  Output dimensions: (128 - 3 + 1) x (128 - 3 + 1) x 32 = 126x126x32
- Second Convolutional Layer (64 filters): [1 mark]
  Output dimensions: (126 - 3 + 1) x (126 - 3 + 1) x 64 = 124x124x64
- Max-Pooling Layer (2x2): [1 mark]
  Output dimensions: 124 / 2 x 124 / 2 x 64 = 62x62x64

(b)
- Number of Parameters per Filter will remain the same, since the filter size (3x3) and the number of input channels (3) remain the same. [0.5 mark]
- The number of filters in each layer (32 and 64) remains the same. [0.5 mark]

(c) Number of Parameters = (Filter Size * Number of Input Channels+1) * Number of Filters = $(3 \times 3 \times 3 + 1) \times 32 = 896$ parameters. [1 mark]

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

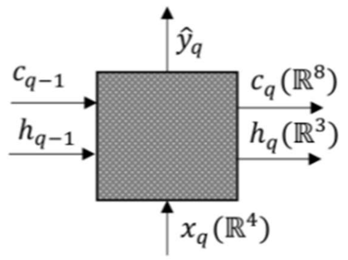
```
# Define the CNN model
model = models.Sequential()
# First Convolutional Layer
model.add(layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(64, 64, 3)))
# Second Convolutional Layer
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
# Max pooling Layer
model.add(layers.MaxPooling2D((2, 2)))
```

(e) 2 marks for correct answer. No step marking.

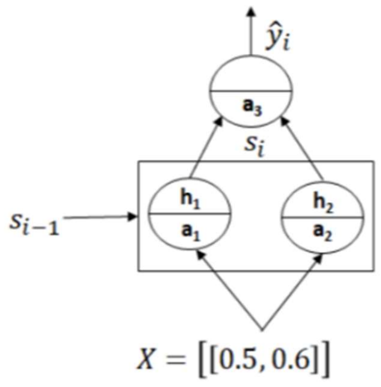
$$O-(odd\,id) = \begin{pmatrix} 0 & 2 \\ -3 & -3 \end{pmatrix} \qquad O-(even\,id) = \begin{pmatrix} 0 \end{pmatrix}$$

2022 DNN EndSem Regular

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. [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 0.12 and previous state as [0.1, 0.2]. The arrows shown in the diagram are representative. [5]



2022 DNN EndSem Regular

(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 Gate (i_t): [1 mark]
  Weights for $h_t(R^3)$: 3 (for each component in h_t) $\times$ 8 (for each component in c_t) = 24 weights. Biases: 8 (for each component in c_t) = 8 biases.
- Forget Gate (f_t): [1 mark]
  Weights for $h_t(R^3)$: 3 (for each component in h_t) $\times$ 8 (for each component in c_t) = 24 weights. Biases: 8 (for each component in c_t) = 8 biases.
- Output Gate (o_t): [1 mark]
  Weights for $h_t(R^3)$: 3 (for each component in h_t) $\times$ 8 (for each component in c_t) = 24 weights. Biases: 8 (for each component in c_t) = 8 biases.
- Candidate Activation ($\tilde{c}_t$): [1 mark]
  Weights for $h_t(R^3)$: 3 (for each component in h_t) $\times$ 8 (for each component in c_t) = 24 weights.
  Weights for $c_t(R^8)$: 8 (for each component in c_t) $\times$ 8 (for each component in c_t) = 64 weights.
  Biases: 8 (for each component in c_t) = 8 biases.
- Total [1 mark]
  = 24+24+24+24+64 weights + 8+8+8+8 biases = 160 weights + 32 biases

(b)

$$
\begin{aligned}
Weighted\,sum &= (u_1 * i_1) + (u_2 * i_2) + (w_1 * h_0) + b \\
\text{Neuron 1: } a_1 &= (0.12 * 0.5) + (0.12 * 0.6) + (0.12 * 0.1) + 0.12 \\
&= 0.06 + 0.072 + 0.012 + 0.12 = 0.264 && [0.5] \\
h_1 &= \tanh(0.264) \approx 0.255 && [1] \\
\text{Neuron 2: } a_2 &= (0.12 * 0.5) + (0.12 * 0.6) + (0.12 * 0.2) + 0.12 \\
&= 0.06 + 0.072 + 0.024 + 0.12 = 0.276 && [0.5] \\
h_2 &= \tanh(0.276) \approx 0.269 && [1]
\end{aligned}
$$

Output Neuron:

$$
\begin{aligned}
Weighted\,sum &= (v_1 * h_1) + (v_2 * h_2) + c \\
a_3 &= (0.12 * 0.255) + (0.12 * 0.269) + 0.12 \\
&= 0.0306 + 0.03228 + 0.12 = 0.18288 && [1] \\
\hat{y} &= sigmoid(0.18288) \approx 0.545 && [1]
\end{aligned}
$$

3. (a) 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? [2]

   (b) A healthcare startup wants to analyze MRI scans to detect tumors automatically. Each MRI scan is a sequence of 50 grayscale images of size 256 $\times$ 256. Describe the neural network architecture you would recommend for this task. Justify your choice. [2]

   (c) You are building a model for detecting fake news articles. Describe the neural network architecture you would recommend for this task. Justify your choice. [2]

2022 DNN EndSem Regular

(d) A tech startup aims to build a chatbot that can understand the context of a conversation, remember past interactions, and respond appropriately. Describe the neural network architecture you would recommend for this task. Justify your choice. Discuss how you would handle varying lengths of input sequences in the model. [2]

(e) An automotive company is developing an autonomous vehicle. One of the challenges is to recognize traffic signs and signals accurately in different lighting conditions. Describe the neural network architecture you would recommend for this task, and justify your choice. Discuss how you would ensure the model generalizes well to different lighting conditions. [2]

(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 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 $4 * [(D * H) + (H * H) + H]$

$$\begin{aligned}
4 * [(D * H) + (H * H) + H] &= 180 \\
4 * [(3 * H) + (H * H) + H] &= 180 \\
H^2 + 4H - 45 &= 0 \\
(H + 9)(H - 5) &= 0 \\
H &= -9 \\
H &= 5
\end{aligned}$$

Dimensionality of the hidden state (H) is approximately 5.

(b) Network recommendation - 1 marks + Justification - 1 mark
Convolutional LSTM (ConvLSTM or CNN-LSTM) would be suitable as it can handle both spatial and temporal dependencies. Sigmoid in last dense layer.
Convolutional Neural Network (CNN) for Image Feature Extraction and Long Short-Term Memory (LSTM) Layers for Temporal Modeling.

(c) Network recommendation - 1 marks + Justification - 1 mark
Pre-trained word embeddings like Word2Vec, GloVe, or fastText can be used to capture semantic information. Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) Layer or Bidirectional RNN/LSTM or Transformer-based model like BERT. Sigmoid in last dense layer.

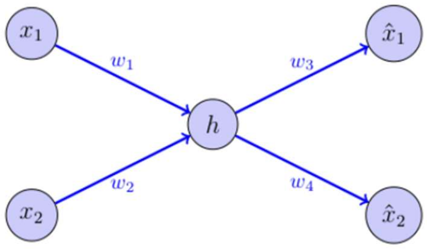
2022 DNN EndSem Regular

(d) Network recommendation - 1 marks + Justification - 1 mark
Pre-trained embeddings like Word2Vec, GloVe, or fastText can be used to capture semantic information. Recurrent Neural Network (RNN) or Transformer Encoder. Seq2Seq Decoder or pre-trained models like GPT-3 or BERT. Output layer depends on the chatbot's specific task.

(e) Network recommendation - 1 marks + Justification - 1 mark
Image Enhancement to enhance the visibility of traffic signs in different lighting conditions (optional). Convolutional Neural Networks (CNN) would be ideal for image recognition tasks. CNN with batch norm, global average pooling with softmax at output layer.
Strategies for Generalizing to Different Lighting Conditions include Data augmentation, transfer learning, ensemble learning, Huber loss. (any one to be written)

4. (a) The training data $(x_1, y_1) = (3.5, 0.5)$ for a single Sigmoid neuron and initial values of $w = -2.0, b = -2.0, \eta = 0.10, \beta = 0.90, epsilon = 1e-8, s_W = 0$ and $s_B = 0$ are provided. Showing all the calculations, find out the values of w, b, s_W and s_B after one iteration of RMSProp. [5]

(b) The training data $(x_1, y_1) = (3.5, 0.5)$ is fed the network given below. The initial values of $w_1 = 0.7, w_2 = 0.2, w_3 = -0.2, w_4 = 0.5, \eta = 0.10, \beta = 0.90$. Showing all the calculations, compute the values of $h, \hat{x}_1, \hat{x}_2, w_4, w_2$ after one iteration of SGD with momentum. Assume RMSE loss. [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.)

2022 DNN EndSem Regular

(a) No marks for equations

- Compute the Gradient [1]

$$\begin{aligned}\hat{y} &= \sigma(wx+b) = \sigma(-2.0\cdot 3.5 - 2.0) \approx 0.0180\\ \nabla_w L &= (\hat{y}-y)\cdot x = (0.0180-0.5)\cdot 3.5 \approx -0.493\\ \nabla_b L &= \hat{y}-y = 0.0180 - 0.5 \approx -0.482\end{aligned}$$

- Update Moving Averages [2]

$$\begin{aligned}s_W &= \beta\cdot s_W + (1-\beta)\cdot(\nabla_w L)^2 = 0.90\cdot 0 + 0.10\cdot(-0.493)^2 \approx 0.0243\\ s_B &= \beta\cdot s_B + (1-\beta)\cdot(\nabla_b L)^2 = 0.90\cdot 0 + 0.10\cdot(-0.482)^2 \approx 0.0233\end{aligned}$$

- Update Weights and Biases [2]

$$\begin{aligned}w &= w + \frac{\eta}{\sqrt{s_W+\epsilon}}\cdot\nabla_w L = -2.0 + \frac{0.10}{\sqrt{0.0243+1e-8}}\cdot(-0.493) \approx -2.0482\\ b &= b + \frac{\eta}{\sqrt{s_B+\epsilon}}\cdot\nabla_b L = -2.0 + \frac{0.10}{\sqrt{0.0233+1e-8}}\cdot(-0.482) \approx -2.0471\end{aligned}$$

(b) No marks for equations

- Forward Pass [1]

$$h = \sigma(w_1x_1 + w_2x_2 + b) = \sigma(0.7 * 3.5 + 0.2 * 0.5 + 0) \approx 0.927$$

- Compute the reconstructed outputs [1]

$$\begin{aligned}\hat{x}_1 &= h\cdot w_3 + b = 0.927 * (-0.2) = (-0.1854)\\ \hat{x}_2 &= h\cdot w_4 + b = 0.927 * 0.5 = 0.4635\end{aligned}$$