National University of Computer & Emerging Sciences, Karachi

Spring 2023 CS-Department

Final Exam

22nd May 2023, 8:30 am – 11:30 am

Course Code: CS4045		**Course Name:** Deep Learning
Instructor Name: Dr Muhammad Atif Tahir, Sumaiyah Zahid		
Student Roll No:		**Section No:**

Instructions:

- Return the question paper.
- Read each question completely before answering it. There are 9 **questions and 5 pages**
- In case of any ambiguity, you may make assumptions. But your assumption should not contradict any statement in the question paper.
- Show all steps clearly.

Time: 180 minutes. **Max Marks**: 50 points

Question 1 [5 Points]

(a) When should you pre-train a new model?

Sol: When there is no pre-trained model available for your specific language or When you have concerns about the bias of the pre-trained model you are using

(b) What is seq2seq model and how to achieve this?

Sol: seq2seq takes as input a sequence of words (sentence or sentences) and generates an output sequence of words. It does so by use of the recurrent neural network (RNN).

(c) What Is the Difference Between a Feedforward Neural Network and Recurrent Neural Network?

Sol: A Feedforward Neural Network signals travel in one direction from input to output. There are no feedback loops; the network considers only the current input. It cannot memorize previous inputs (e.g., CNN).

A Recurrent Neural Network's signals travel in both directions, creating a looped network. It considers the current input with the previously received inputs for generating the output of a layer and can memorize past data due to its internal memory.

(d) Write two main applications of a Recurrent Neural Network (RNN)?

The RNN can be used for sentiment analysis, text mining, and image captioning.

(e) What is Pooling on CNN, and How Does It Work?

Sol: Pooling is used to reduce the spatial dimensions of a CNN. It performs down-sampling operations to reduce the dimensionality and creates a pooled feature map by sliding a filter matrix over the input matrix.

(f) What Are Vanishing and Exploding Gradients?

Sol: While training an RNN, your slope can become either too small or too large; this makes the training difficult. When the slope is too small, the problem is known as a "Vanishing Gradient." When the slope tends to grow exponentially instead of decaying, it's referred to as an "Exploding Gradient." Gradient problems lead to long training times, poor performance, and low accuracy.

(g) Write down the two primary components of Generative Adversarial Network (GAN)

Sol: Generator
Discriminator

(h) What do you understand by transfer learning?
Sol: Transfer learning is the process of transferring the learning from a model to another model without having to train it from scratch. It takes critical parts of a pre-trained model and applies them to solve new but similar machine learning problems.

(i) Name four commonly used transfer learning models.
Sol: Some of the popular transfer learning models are: VGG-16, BERT, GTP-3, Inception V3, XCeption

(j) Random Erasing was specifically designed to combat image recognition challenges due to _____.
Sol: Occlusion

Question 2 [8 Points]:

(a) [1 Point] Consider a CNN architecture, where the input image is of size 200x200x64. The first convolutional layer has 96 kernels of size 10x10x3. The stride is 2 and padding is 1. What would be the size of the output image after the first convolutional layer?
Sol: output = (input + 2p –k)/s+1

output = (200 + 2(1) – 10) / 2 +1 = 192/2+1 = 83x83x96

(b) [1 Point] Calculate the number of parameters with RGB image as input, followed by 3×3 filter, and output with 9 channels.
Sol: *num_params*
= $[i \times (f \times f) \times o] + o$
= **[3 × (3×3) × 9] + 9**
= **252**

(c) Consider simple convolutional layer example with following input X and Filter F

```
X= [[2 3 1]        F= [[1 -1]
    [4 5 6]           [-1 1]]
    [4 7 8]]
```

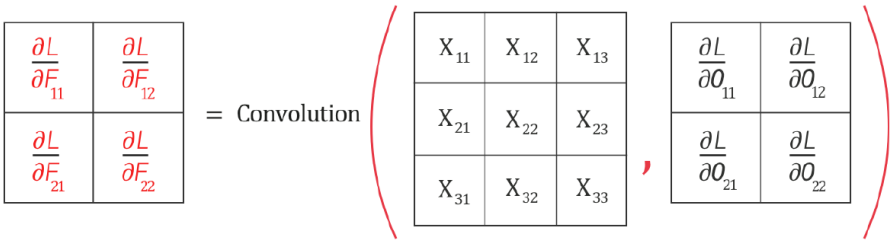
(i) [2 points] Compute the convolution operation between X and F which gives output O. Use stride = 1. Show all calculations.

Sol:

O11 = 2 * 1 + 3* -1 + 4 * -1 + 5 * 1 = 2-3-4+5 = 0
O12 = 3 * 1 + 1* -1 + 5 * -1 + 6 * 1 = 3-1-5+6 = 3
O21 = 4 * 1 + 5* -1 + 4 * -1 + 7 * 1 = 4-5-4+7 = 2
O22 = 5 * 1 + 6* -1 + 7 * -1 + 8 * 1 = 5-6-7+8 = 0

(ii) [2 points] Find the gradients for X and F i.e. ∂L/∂X and ∂L/∂F assuming ∂L/∂O = X= `[[2 3 ][4 5]`. For ∂L/∂X, just show the formula and no need to compute full convolution operation.

Sol:



$$\frac{\partial L}{\partial X} = \text{Full Convolution}\left(\text{180° rotated Filter F}, \text{Loss Gradient } \frac{\partial L}{\partial O}\right)$$

Question 3 [3 Points]:

(a) Explain why Deep Ensemble Learning would most likely work better than simple Deep Learning?
Sol: Students need to discuss Statistical, Computational, and Representational Problems

(b) "IBaggedFCNET: An Ensemble Framework for Anomaly Detection in Surveillance Videos" paper was discussed during lectures. Why does an ensemble work for this problem?
Sol: Due to Bagging. Bagging reduces variance by voting/ averaging, thus reducing the overall expected error

(c) What is the difference b/w Homogenous and Heterogeneous Deep Ensemble Learning?
Sol: (a) Homogeneous means the same deep learning method trained in a different way (b) Heterogeneous Deep Ensemble Learning means a combination of various deep learning methods

Question 4 [9 Points]: Consider the given four-layer autoencoder architecture(number of neurons in each layer : 2-3-1-3-2), where each node in the network applies ReLU non-linearity along with standard network operations, for simplicity the bias unit are not considered for the given network.

The details of the input and weights are given below.

$X = [[0.43],\ [0.98]]$

$weights\ of\ L^1$

$W^1 = \quad [[0.23,\ 0.24],\ [0.,\ 0.7\ ],[0.01,\ 0.45]]$

$weights\ of\ L^2$

$W^2 = \ [[0.42, 0.51,\ 0.89]]$

$weights\ of\ L^3$

$W^3 = \ [[0.25],\ [0.31],\ [0.59]]$

$weights\ of\ L^4$

$W^4 = \ [[0.97,\ 0.55,\ 0.97],\ [0.71,\ 0.7\ ,\ 0.22]]$

Learning Rate = 0.5.

The reconstruction error is given below.

$$J(\theta) = \quad - \sum_{d=1}^{|X|} X_d \, log(\hat{X}_d)$$

i. [3.5 Points] Compute the output of each node in the network for forward propagation. Nodes 1,2,3,4,5,6,7

ii. [1 Point] Compute the reconstruction error using network output $\hat{X}$. y=[0.23, 0.61]

iii. [3 Points] Perform backpropagation, compute the derivatives i.e. dW^4, dW^3

iv. [1.5 Points] Calculate the updated weights. Only W4.

For derivation of final layer:

$$dz^{[n]} = A^n - X$$
$$dw^{[n]} = dz^{[n]} * A^{[n-1]}$$

For derivation other layers:

$$dz^{[n-1]} = W^{[n]} * dz^{[n]} * g^{[n-1]'}(z^{[n-1]})$$
$$dw^{[n-1]} = dz^{n-1} * A^{[n-2]}$$

Inverse of ReLU

$$g'(x) = \{0 \quad if\ x < 0\ 1 \quad if\ x > 0$$

Solution:

Node 1 : $0.43 \times 0.23 + 0.98 \times 0.24 = 0.3341$

Node 2 : $0.43 \times 0 + 0.98 \times 0.07 = 0.686$

Node 3 : $0.43 \times 0.01 + 0.98 \times 0.45 = 0.4453$

Node 4 : $0.3341 \times 0.42 + 0.686 \times 0.51 + 0.4453 \times 0.89$

$= 0.886499$

Node 5 : $0.886499 \times 0.25 = 0.22162$

Node 6 : $0.886499 \times 0.31 = 0.2748$

Node 7 : $0.886499 \times 0.59 = 0.52303$

Node 8 : $0.22162 \times 0.97 + 0.2748 \times 0.55 + 0.52303 \times 0.97$

$= 0.87345$

Node 9 : $0.22162 \times 0.71 + 0.2748 \times 0.7 + 0.52303 \times 0.22$

$= 0.4647768$

Reconstruction loss :

$J(\theta) = -0.43 \log(0.87345) - 0.98 \log(0.4647768)$

$= 0.3486$

Some students have reconstructed the error from $y = [0.23, 0.61]$

$J(\theta) = -0.43 \log(0.23) - 0.98 \log(0.61)$

$J(\theta) = 0.484833$

Back propagation for final layer.

$d z_8^4 = A_8^4 - X$

$= 0.87345 - 0.43 = 0.44345$

$d z_9 = A_9 - X$

$= 0.464776 - 0.98 = -0.5152232$

$dw_{58} = d z_8 \times A_5$

$= (0.44345)(0.22162) = 0.098277$

$dw_{68} = d z_8 \times A_6$

$= (0.44345)(0.2748) = 0.12186$

$dw_{78} = d z_8 \times A_7$

$(0.44345)(0.52303) \Rightarrow 0.23193$

$dw_{59} = d z_9 \times A_5$

$= (-0.5152232) \times (0.22162) \Rightarrow -0.11418$

$dw_{69} = d z_9 \times A_6$

$= (-0.5152232) \times (0.2748) \Rightarrow -0.1415$

$dw_{79} = d_{29} \times A_7$

$= (-0.515232) \times (0.52303) = -0.2694$

Backpropagation for layer 3

$d_{23} = \sum w^4 \times d_2^4 \times g^3 \times z^3$

$d_{25} = 0.97 \times (0.44345)\ (1)\ (0.22162) +$
$0.71 \times (-0.515232)(1)(0.22162)$
$= 0.01425$

$dw_{45} = d_{25} \times A_4$
$= 0.01425 \times 0.8864 = 0.012636$

$d_{26} = 0.97 \times (0.44345)(1)(0.2748) +$
$0.71 \times (-0.515232)(1)(0.2748)$
$= -0.0321$

$dw_{46} = d_{26} \times A_4$
$= -0.0321 \times 0.8864 = -0.02844$

$d_{27} = 0.97 \times (0.44345)\ (1)\ (0.52303) +$
$0.22 \times (-0.515232)(1)(0.52303)$
$= 0.16569$

$dw_{47} = d_{27} \times A_4$
$= 0.16596 \times 0.8864 = 0.146886$

Question 5 [7 Points]:
RNN
[4 points] Recall that a recurrent neural network (RNN) takes in an input vector xt and a state vector ht−1 and returns a new state vector ht and an output vector yt :

$$h_t = f(w_1 x_t + w_2 h_{t-1} + b_2)$$
$$y_t = g(w_3 h_t + b_3),$$

where f and g are activation functions.
(a) The following diagram depicts a single RNN unit, where xt , ht−1, ht and yt are all scalars, as a state machine:

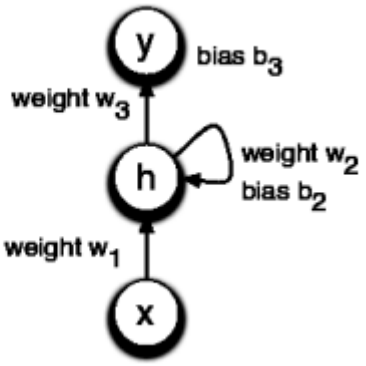
$w_1 =$ _____

$w_2 =$ _____

$b_2 =$ _____

$w_3 =$ _____

$b_3 =$ _____



Suppose that f is a binary threshold unit and g is a linear unit:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

$$g(x) = x.$$

Fill in weights (w1, w2, w3), biases (b1, b2) so that the RNN initially outputs 0, but as soon as it receives an input of 1, it switches to outputting 1 for all subsequent time steps. For instance, the input 0001010 produces the output 0001111. The hidden unit has an initial value of 0.
Hint: In one possible solution, the hidden unit has an activation ht = 0 until there's an input xt = 1, at which point it switches to maintaining an activation of 1 forever. The output unit always predicts the same as the hidden unit, i.e. yt = ht.

Solution:
There are many possible solutions, but only one main scenario: the output unit must predict the same as the hidden state.
This requires that:
b3 = 0
w3 + b3 = 1
b2 < 0
w2 + b2 ≥ 0
w1 + b2 ≥ 0,
so values like **w1 = 1, w2 = 1, b2 = −1, w3 = 1, b3 = 0** would be considered correct.

LSTM:
You build a sentiment analysis system that feeds a sentence into a LSTM, and then computes the sentiment class between 0 (very negative) and 4 (very positive), based only on the final hidden state of the LSTM.

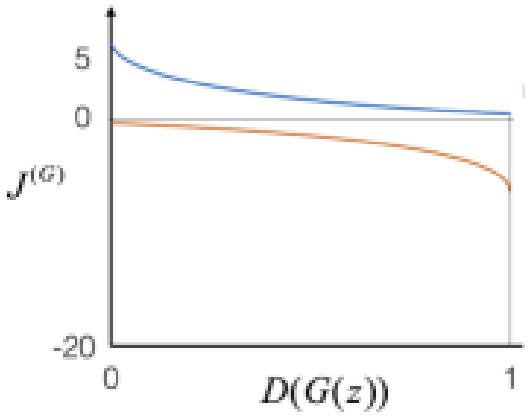
1. [1 point] Name at least one benefit of the LSTM model over the bag-of-vectors model.
   Solution: The LSTM model is able to integrate information from word ordering, e.g. "this was not an amazing fantastic movie" while the bag-of-vectors model can not.

2. [1 points] What is one advantage that a LSTM would have over a neural window-based model for this task?
   Solution: There are multiple answers: We can process arbitrary length inputs. It can encode temporal information.('Take ordering into consideration' is only partially correct, because theoretically window-based model also can, although hard to). Shared weights. Fewer parameters. The number of parameters would increase proportionally to the input size of the neural window-based network whereas it would stay constant for RNNs since weights are shared at every time-step.

3. [1 points]You observe that your model predicts very positive sentiment for the following passage:
   "Yesterday turned out to be a terrible day. I overslept my alarm clock, and to make matters worse, my dog ate my homework. At least my dog seems happy…"
   Why might the model misclassify the appropriate sentiment for this sentence**?**
   Solution: The final word in the sentence is 'happy' which has a very positive sentiment. Since we only use the final hidden state to compute the output, the final word would have too much impact in the classification. In addition, because the sentence is quite long, information from earlier time steps may not survive due to the vanishing gradient problem

Question 6 [3 Points]:
Consider training a GAN with generator G(z) and discriminator D(G(z)). The perfect discriminator outputs 1 on a real instance, 0 on a fake instance. Figure 1 shows the training losses for two generator loss functions. In detail, for the first m generated points, i = 1,…,m. Each point in the plot shows the value of D(G(zi)), J1(G) and J2(G), defined as follows.

1. J1(G) = -1/m ∑ i=1,..m ln(D(G(zi)). The curve above 0 axis.
2. J2(G) = 1/m ∑ i=1,..m ln(1-D(G(zi)). The curve below 0 axis.



1. [1 point] Early in the training, is the value of D(G(z)) closer to 0 or closer to 1? Explain why.
Early in the training, the value of D(G(z)) is closer to 0 detecting generator images as fake. The generator is still learning to create realistic samples that can fool the discriminator. When it will learn completely it would give 1, assuming them as a real instance.
2. [1 point] Which of the two cost functions would you choose to train your GAN? Justify your answer.
Both cost functions can contribute to training a successful GAN, but they emphasize different aspects:

- **If the primary goal is to focus on improving the quality and realism of the generated samples, the cost function moving from 5 to 0 for D(G(z)) may be more suitable. By encouraging the generator to produce samples that are increasingly difficult for the discriminator to differentiate, this cost function can lead to generating more realistic and high-quality outputs.**
- **On the other hand, if the objective is to prioritize the generator's ability to deceive the discriminator and generate samples that are hard to classify, the cost function going from 0 to -5 may be preferred. This cost function encourages the generator to focus on producing samples that are more likely to trick the discriminator, even if they may not necessarily be highly realistic.**

3. [1 point] A GAN is successfully trained when D(G(z)) is close to 1. True or False? Explain your answer.

False. If D(G(z)) is close to 1, it means that the discriminator is assigning high probabilities to the generated samples, considering them as real. This scenario suggests that the discriminator is ineffective in distinguishing between real and generated data. The ideal scenario is when the discriminator assigns probabilities close to 0.5 for both real and generated samples.

Question 7[5 Points]

Consider the word embeddings for the input sentence "DLP IS FUN" are X:

where X= `[[2 1 1]`
`[3 2 1]`
`[1 4 5]]`

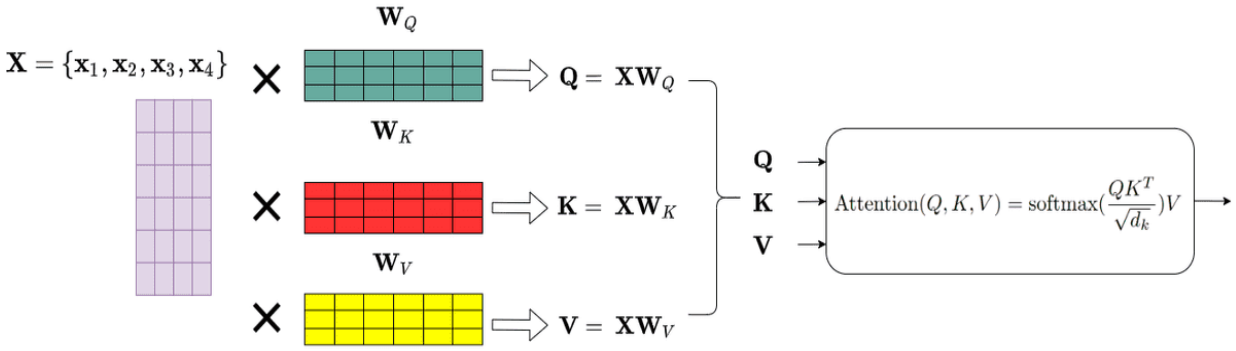
Calculate the *Outputs* of the Self-Attention A(Q,K,V), given that:

W_{Query} = `[[2 3]`
`[0 2]`
`[2 3]]`

W_{Key} = `[[0 0]`
`[2 1]`
`[2 2]]`

W_{Value} = `[[2 2]`
`[3 0]`
`[3 3]]`

and d_k = 1 and all biases = 0. You can look at the diagrams from lecture slides for A(Q,K,V) calculation in below



Solution

Q is [[6 11]
[8 16]
[12 26]]
K is [[4 3]
[6 4]
[18 14]]
V is [[10 7]
[15 9]
[29 17]]
[[29. 17.]
[29. 17.]
[29. 17.]]

Question 8 [5 Points]

a) [2 points] Write down the 3 applications of Embedded Deep AI.
Solution: HealthCare, Smart Cities, Autonomous Cars

b) [2 points] What are the two types of quantization techniques used in Embedded Deep AI.
Solution: Post-training quantization
Quantization-aware training

c) [1 point] Considering the low hardware cost and easy access of GPU nowadays, do you think embedded AI is important?
Solution: Yes

Question 9 [5 Points] Write down the name of the missing components of Transformers in the diagram below. You only need to write in this question paper. The list of components is below, you need to fit.

Input embedding, Output Embedding, Positional Encoding, Multi-Head Attention, Masked Multi Head Attention, Add & Norm, Feed Forward, Linear, Softmax, Sigmoid