National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2023** **Islamabad Campus**

CS-4045: Deep Learning for Perception

Serial No:

Final Exam

Total Time: 3 Hours

Total Marks: 100

Tuesday, 30th December, 2023

Course Instructors

Akhtar Jamil

____

Signature of Invigilator

______________________ ______________ ______________ ____________________

Student Name **Roll No.** **Course Section** **Student Signature**

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work.
3. Verify that you have **nineteen (19)** different printed pages including this title page. There are **six (6)** questions.
4. Calculator sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
6. Ensure that you do not have any electronic gadget (like mobile phone, smart watch, etc.) with you.

	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Total
Marks Obtained							
Total Marks	50	20	5	5	10	10	100

Question 1. **MCQ Answer Sheet**

Cross (X) the correct answer. Overwriting an MCQ will result in ZERO marks.

[1 x 50 = 50]

S. No	A	B	C	D	S. No	A	B	C	D
1.			x		26.	x			
2.		x			27.	x			
3.	x				28.				x
4.		x			29.	x			
5.		x			30.		x		
6.				x	31.				x
7.	x				32.		x		
8.			X		33.			x	
9.				x	34.				x
10.				x	35.				x
11.		x			36.	x			
12.				x	37.			x	
13.			x		38.			x	
14.				x	39.	x			
15.			x		40.			x	
16.				x	41.		x		
17.	x				42.		x		
18.		x			43.				x
19.		x			44.				x
20.		x			45.	x			
21.	x				46.				x
22.				x	47.			x	
23.			x		48.	x			
24.				x	49.		x		
25.				x	50.	x			

Question 1. MCQs based questions have only one possible answer. Only answers marked on the answer sheet will be considered.

1. Which of the following models are more suitable for image-to-image translation:
   a) ANNs
   b) Conventional Autoencoders
   c) Conditional GANs
   d) LSTMs

2. What is one challenge in training Conditional GANs?
   a) They require 3D input data
   b) **Ensuring balance between the generator and discriminator**
   c) They can only be trained on CPUs
   d) Requiring internet connectivity for training

3. Assume that for Yolo algorithm, an image is divided into 9 x 9 grids, each grid can predict 5 bounding boxes and there are total 10 classes. What will be the size of the final output tensor?
   a) **9 x 9 x 35**
   b) 9 x 9 x 55
   c) 9 x 9 x 75
   d) 9 x 9 x 95

4. Consider the input data $x = [0.5, 0.6]$ and its corresponding output obtained for an autoencoder network is $\hat{x} = [0.4, 0.7]$. The Mean Squared Error (MSE) Loss is:
   a) 0.10
   b) 0.01
   c) 0.25
   d) 0.45

5. Convolutional layers cannot be added to construct autoencoders.
   a) True
   b) False

6. What does 'position encoding' in Transformers help to achieve?
   a) It reduces the computational complexity of the model
   b) It encrypts the data for security purposes
   c) It increases the speed of the attention mechanism
   d) It helps the model in understanding the order of words in a sequence

7. What is the primary advantage of transformers over traditional RNNs?
   a) Faster training times due to parallelization
   b) Lower memory requirements
   c) Ability to handle 3D data
   d) Better performance on small datasets

8. What is the primary purpose of gradient clipping in neural network training?
   a) To prevent the vanishing gradient problem.
   b) To accelerate the training process.
   c) To mitigate the exploding gradient problem.
   d) To reduce the overall number of training parameters.

9. Which of the following is not correct about RNNs?
   a) RNNs can't capture long-range dependencies.
   b) RNNs can face gradient explosion or vanishing problems.
   c) RNNs require a large number of training steps.
   d) RNNs process data in parallel, thus making them highly efficient for sequential data.

10. How do DCGANs typically handle the vanishing gradient problem in GANs?
    a) By using high learning rates
    b) By reducing the number of layers in the network
    c) Implementing recurrent neural network structures
    d) **Through the use of batch normalization**

11. Why are Vision Transformers considered effective for large-scale image datasets?
    a) They require less computational power than CNNs
    b) They can capture long-range dependencies between image patches
    c) They are faster than traditional machine learning models
    d) They do not require GPU acceleration

12. In some LSTM architectures that use 'coupled forget and input gates', what is the primary function of this modification?
    a) Increases model complexity.
    b) Accelerates learning speed.
    c) Simplifies the model, reducing overfitting.
    d) Synchronizes forgetting and adding information.

13. In which of the following scenarios is a Recurrent Neural Network (RNN) generally NOT considered the most suitable model?
    a) Time-series forecasting, like stock price prediction
    b) Sequence generation, like text generation
    c) Processing fixed-size inputs with no sequential or time-dependent nature
    d) Language modeling, such as predicting the next word in a sentence

14. Which of the following solution(s) might help produce diverse outputs from conditional GANs?
    a) Adding Noise to input
    b) Adding regularization terms
    c) Changing the loss function
    d) All of the above.

15. In Vision Transformers, what is the purpose of the 'class token'?
    A) To classify the type of transformer model
    B) To act as a placeholder for missing image data
    C) To represent the entire image for classification purposes
    D) To encrypt the class labels for security

16. What is the primary function of a denoising autoencoder?
    a) To compress data efficiently.
    b) To speed up data processing.
    c) To classify data into categories.
    d) To reconstruct clean data from corrupted inputs.

17. In an LSTM network, the 'peephole connection' refers to:
    a) Connections that provide feedback from the cell to the gates.
    b) Connections from the previous hidden state to the gates in the current cell.
    c) An additional input layer to enhance memory capability.
    d) Connections bypassing the forget gate for faster computation.

18. In a GRU, consider the previous hidden state $h_{(t-1)} = 0.5$ and the current input $x_t = 0.6$. Assume the weight matrix for the reset gate is $W_r = [0.8, -0.5]$ and the bias is $b_r = 0.1$. Calculate the output of the reset gate r_t using the sigmoid function. What is approximated value of r_t?
    a) 0.45
    b) **0.52**
    c) 0.68
    d) 0.61

19. RNNs usually take variable length input and do not struggle to memorize with long sequences:
    a) True
    b) False

20. What technique is commonly used to mitigate the vanishing gradient problem in RNNs?
    a) Pooling layers
    b) Truncated Backpropagation Through Time
    c) Convolutional layers
    d) Fixed weight matrices

21. In Conditional GAN, the discriminator typically takes three types of inputs.
    a) True
    b) False

22. In the context of Transformers, what is 'self-attention'?
    a) The model's focus on its own parameters
    b) An external mechanism to adjust model weights
    c) The comparison of the model's output to a predefined standard
    d) An attention mechanism where the model relates different positions of a single sequence

23. The vanishing gradient problem in traditional RNNs is primarily addressed in LSTMs through:
    a) The use of ReLU activation functions.
    b) Decreasing the depth of the network.
    c) The introduction of gating mechanisms.
    d) Implementing dropout techniques.

24. Which of the following tasks is Recurrent Neural Networks (RNNs) NOT particularly suitable to solve?
    a) Sentiment Analysis
    b) Semantic Search
    c) Predicting the next word in a sentence based on the context of previous words.
    d) Identifying objects in a static image.

25. In the context of DCGANs, what does 'stability' in training refer to?
    a) The model's ability to generate consistent images over time
    b) The network's resistance to adversarial attacks
    c) The speed at which the network trains
    d) **The balance between the generator and discriminator during training**

26. Global average pooling may increase model stability, but it might hurt convergence speed.
    a) True
    b) False

27. The generator in DCGAN used a similar concept of strided convolutions that allow it to learn its own spatial upsampling.
    a) True
    b) False

28. Consider a simple RNN with a single neuron. The neuron's activation function is a hyperbolic tangent (tanh). Given the previous hidden state $h_{(t-1)} = 0.3$ and the current input $x_t = -0.5$, with a weight matrix $W = [0.6, -0.4]$ and a bias of $b = -0.1$. What is h_t?
    a) -0.47
    b) 0.35
    c) -0.30
    d) **0.27**

29. RNNs are generally computationally more expensive and are slow to converge.
    a) True
    b) False

30. Which of the following best describes the attention mechanism in Transformers?
    a) A tool to reduce computational complexity
    b) A method to focus on specific parts of the input sequence
    c) A technique to compress input data
    d) A way to increase the depth of the neural network

31. What type of activation function is commonly used in the discriminator's output layer of a DCGAN?
    a) Tanh
    b) ReLU
    c) Softmax
    d) Sigmoid

32. Why does Faster R-CNN utilize anchor boxes?
    a) To reduce computational complexity.
    b) To handle variations in objects' scale and aspect ratio.
    c) To increase the speed of training.
    d) To enhance color detection in images.

33. In sequence prediction, why is a BiLSTM often more effective?
    a) Faster training
    b) Less data needed
    c) Captures bidirectional dependencies
    d) Simpler architecture

34. Which loss function is commonly used in an autoencoder for reconstruction accuracy?
    a) Cross-entropy
    b) Hinge loss
    c) Binary Cross-entropy
    d) Mean Squared Error (MSE)

35. What happens in an autoencoder if the hidden layer has more neurons than the input layer?
    a) Increases accuracy
    b) Better feature extraction
    c) Reduces complexity
    d) Trivial encoding

36. In R-CNN, the region proposals are extracted first and then the CNN-based features are extracted.
    a) True
    b) False

37. What is a key feature of an undercomplete autoencoder?
    a) Have more layers than an overcomplete autoencoder.
    b) Have more neurons in the hidden layer than inputs.
    c) Have fewer neurons in the hidden layer than inputs.
    d) Have linear activation functions only.

38. What is the purpose of weight tying (weight sharing) in regularized autoencoders?
    a) To increase the model's training speed.
    b) To reduce the model's computational complexity.
    c) Acts as regularizer.
    d) To enhance the model's feature extraction capability.

39. In a Fast R-CNN model, an ROI pooling layer is applied to a feature map of size 8×8 pixels. If the ROI pooling size is set to 2×2, what is the size of the output size from the ROI pooling layer?
    a) $\mathbf{2 \times 2}$
    b) 4×4
    c) 3×3
    d) 2×4

40. How do Conditional GANs differ from traditional GANs?
    a) Conditional GANs have a simpler architecture
    b) Traditional GANs use labeled data, while Conditional GANs do not
    c) Conditional GANs use additional information to guide the generation process
    d) There is no difference; they are the same

41. In Faster R-CNN, the Region Proposal Network (RPN) is responsible for predicting the class of the object.
    a) True
    b) False

42. When is a BiLSTM less suitable compared to a standard LSTM?
    a) Large datasets
    b) Real-time processing
    c) Complex sequences
    d) Short sequences

43. Consider an RNN architecture for sentiment analysis problem. How is the loss typically calculated at the output layer?
    a) By summing the losses at each time step.
    b) By averaging the losses across all time steps.
    c) By using the loss from the first time step only.
    d) By computing the loss only at the final time step.

44. Using the nearest neighbor method, the following 2 × 2 matrix is upsampled to a 4x4 one. Which is the correct option?

$$\begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}$$

a)
$$\begin{bmatrix} 3 & 3 & 1 & 1 \\ 3 & 3 & 1 & 1 \\ 4 & 4 & 2 & 2 \\ 4 & 4 & 2 & 2 \end{bmatrix}$$

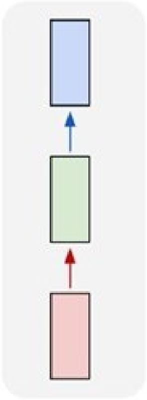
b)
$$\begin{bmatrix} 3 & 1 & 0 & 0 \\ 4 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

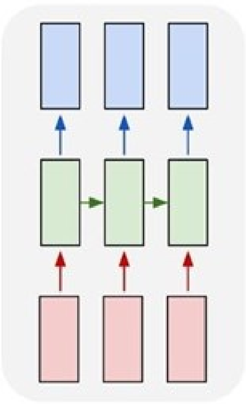
c)
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 2 \end{bmatrix}$$

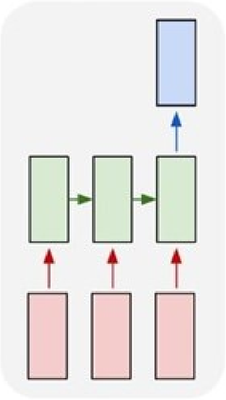
d)
$$\begin{bmatrix} 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

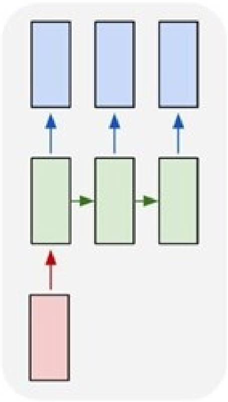
45. Generally two-shot detectors are more accurate than single shot detectors?
   a) True
   b) False

46. Which of the following RNN architecture is more suitable for image captioning problem?

a)


b)


c)


d)


47. Why is Non-Maximum Suppression (NMS) applied in Faster R-CNN?
   a) To accelerate model training.
   b) To reduce the feature map size.
   c) To resolve duplicate proposals on the same object.
   d) To enhance the resolution of the proposals.

48. Transformer models can process data in parallel, making them more efficient for tasks involving sequential data.
   a) True
   b) False

49. Selective search in computer vision combines pixels into regions using a top-down approach.
   a) True
   b) False

50. Fast-RCNN is a single network that can be trained for both classification and localization at the same time.
   a) True
   b) False

Question 2. Write short answers to the following questions. [2 x 10 = 20]

1) What is the model collapse problem in GANs? Write two solutions to deal with model collapse problem.

The model collapse problem in Generative Adversarial Networks (GANs) refers to a situation where the generator produces a limited variety of samples that are often very similar or nearly identical, making the generated data distribution lack diversity. This problem can occur when the generator learns to exploit weaknesses in the discriminator or fails to capture the entire data distribution properly.

Possible solutions:

- Regularization Techniques

- Architectural Variations or change loss function

- Adding Noise to Inputs

2) How GANs are trained?

The training process involves a continual back-and-forth between the generator and discriminator, with the generator trying to produce more realistic data and the discriminator trying to improve its ability to distinguish real from fake data. This adversarial training process continues until the generator produces data that is visually indistinguishable from real data, and the discriminator can't reliably differentiate between them. Achieving this equilibrium is the primary goal of GAN training.

3) How does Backpropagation Through Time (BPTT) differ from standard backpropagation?

The standard backpropagation is used for independent data points, whereas BPTT is employed when dealing with sequences. BPTT extends the traditional backpropagation by "unrolling" the RNN through time, treating the sequence as a series of connected feedforward networks, allowing it to capture temporal dependencies. It calculates gradients for each time step separately and accumulates them over the entire sequence, enabling the model to learn from the historical context inherent in sequential data.

4) Explain why KL-Divergence term in added in the objective function of VAE?

The regularization term in VAEs, often represented as the KL divergence loss, plays a crucial role in shaping the latent space and ensuring that it has the desired properties. It helps control the smoothness of the latent space, enforces a prior distribution, prevents overfitting, and simplifies the sampling process, ultimately improving the generative capabilities and interpretability of the VAE model.

5) Suppose you want the memory cell of an LSTM to sum all the inputs over time. What should be the possible output values for the input gate and forget gate? Explain.

To ensure that new inputs are always added to the cell state, both the input gate and output gate should ideally equal to 1 (or close to it).

6) Explain why do we need to perform reparametrize the sampling layer in VAE? Write its equation and explain the parameters used.

In Variational Autoencoders (VAEs), reparametrization of the sampling layer is a crucial step for enabling backpropagation through stochastic nodes. This process is essential because it allows the gradient of the loss function to be backpropagated through the random sampling process, which is inherently non-differentiable.

The reparametrization trick involves expressing the random variable Z (which is sampled from a distribution) as a deterministic function of a fixed distribution (like a standard Gaussian) and some parameters. The common approach in VAEs is:

$$Z = \mu + \sigma \odot \epsilon$$

Where:
- Z is the latent variable (sampled).
- μ is the mean of the distribution, learned by the network.
- σ is the standard deviation (or sometimes variance) of the distribution, also learned by the network.
- ϵ is a random noise sampled from a standard Gaussian distribution \(\mathcal{N}(0, 1) \).
- $\odot$ represents element-wise multiplication.

7) How does multi-head attention differ from single-head attention as used in Vision Transfer model?

Multi-head attention differs from single-head attention primarily in its ability to simultaneously focus on and capture multiple aspects or relationships within the data. In multi-head attention, multiple independent attention mechanisms, or "heads," operate in parallel, each potentially focusing on different types of information or patterns. This contrasts with single-head attention, where only one pattern or relationship is captured at a time. In the context of a Vision Transformer model, multi-head attention allows the network to attend to various features or parts of an image concurrently, offering a richer and more nuanced understanding of the visual input. This parallel processing of multiple attention pathways enables more complex and diverse data representations, making multi-head attention particularly effective for tasks involving intricate or multifaceted data, like those in vision processing.

8) Consider a single unit of a Recurrent Neural Network (RNN) with hyperbolic tangent activation function:

$$h_t = \tanh(W \cdot [x_t;\ h_{t-1}] + b_t)$$

The following data is feed to the network. Calculate the hidden state h_t.

$x_t = [0.5]$
$h_{t-1} = [0.1]$
$W = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}$
$b_t = [0.2]$

$h_t = \tanh(W \cdot [x_t;\ h_{t-1}] + b_t)$
$h_t = \tanh(W[0] \cdot x_t + W[1] \cdot h_{t-1} + b_t)$

$h_t = \tanh(\, 0.54\,)$
$h_t = \tanh(\, 0.6 \cdot 0.5 + 0.4 \cdot 0.1 + 0.2\,)$
$h_t \approx 0.493$

9) Explain Truncated Backpropagation through time.

Truncated Backpropagation Through Time (TBPTT) is a variant of the Backpropagation Through Time algorithm used for training Recurrent Neural Networks (RNNs), where the sequences are very long or continuous. In standard BPTT, gradients are computed by unrolling the entire sequence, which can lead to high computational costs and difficulties with vanishing and exploding gradients. TBPTT addresses these issues by breaking the long sequence into smaller segments or chunks. During training, the RNN is unrolled for a fixed number of timesteps (the truncation size), and the gradients are computed and propagated back only through this limited number of timesteps, not the entire sequence. This approach not only reduces the computational burden but also helps mitigate the vanishing gradient problem by limiting the number of steps over which gradients are accumulated. However, it introduces a trade-off between the computational efficiency and the ability to learn dependencies over longer time lags.

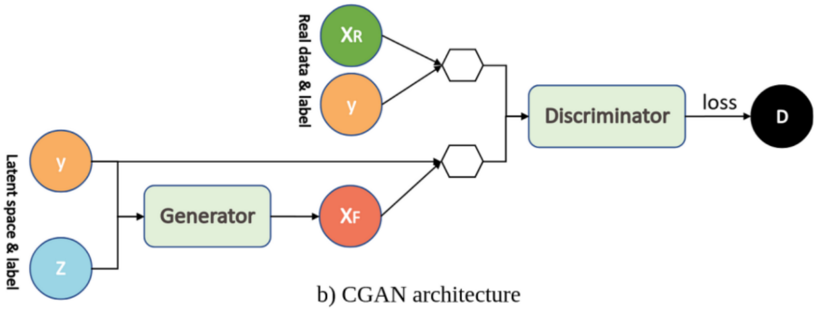
10) What are Regularized autoencoders? Write the objective function as used in them.

Regularized autoencoders are a type of autoencoder, used for unsupervised learning, which incorporates regularization techniques to achieve more meaningful and robust feature representations. The regularized autoencoders also impose additional constraints or penalties during training. These regularization techniques lead to more generalized models, preventing overfitting and often resulting in better feature extraction and representation. The objective function for a regularized autoencoder is as below:

$$\min_{\theta,w,w^*,\mathbf{b},\mathbf{c}} \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n}(\hat{x}_{ij}-x_{ij})^2+\lambda\|\theta\|^2$$

Question 3. [5+5]

a) With the help of a diagram, explain the working of conditional GANs. Also write its objective function.



b) CGAN architecture

Conditional Generative Adversarial Networks (cGANs) extend the basic GAN framework by introducing a conditional aspect, allowing the model to generate data conditioned on some additional information. This information could be anything like class labels, data from other modalities, or even textual descriptions. In a cGAN, both the generator and the discriminator are conditioned on this additional information, enhancing the control over the data generation process.

The working of a cGAN involves two main components: the generator and the discriminator. The generator in a cGAN receives both a random noise vector and the conditional information as inputs. It learns to generate data (like images) that not only appear realistic but also correspond to the given conditions. The discriminator, on the other hand, is tasked with distinguishing between the real data and the fake data produced by the generator. However, unlike standard GANs, the discriminator in a cGAN also takes the conditional information as an input, which it uses to make a more informed decision about whether the data is real or generated. This setup enables the model to generate data that is more specific and relevant to the given condition, leading to more targeted and controlled data generation.

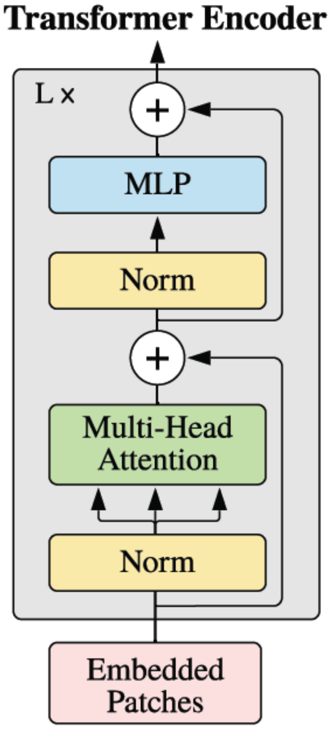
The objective function of a cGAN is an extension of the standard GAN objective, incorporating the conditional aspect. It can be represented as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{\mathrm{dt}}(x)}[\log D(x|y)] + E_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z|y)\right)\right)\right]$$

Here, G is the generator, D is the discriminator, x is the real data, z is the random noise, and y is the conditional information. The first term in the objective function represents the log-likelihood that the discriminator correctly identifies real data, while the second term represents the log-likelihood that the discriminator correctly identifies generated data as fake. The generator tries to minimize this function while the discriminator tries to maximize it, leading to a minimax game. This objective function ensures that the generator not only learns to produce realistic data but also aligns its outputs with the conditional inputs, making cGANs powerful tools for controlled and diverse data generation tasks.

b) With the help of a diagram explain working of transformer encoder. Explain each component involved.

The Transformer encoder, architecture is designed to process sequences in parallel, significantly speeding up training and handling long-range dependencies more effectively. The encoder consists of a stack of identical layers, each comprising two main subcomponents: a multi-head self-attention mechanism and a position-wise feed-forward network.



The multi-head self-attention mechanism is the heart of the Transformer encoder. It allows the model to weigh the importance of different parts of the input sequence differently, enabling it to capture various aspects of the sequence context at different positions. This mechanism involves three key vectors computed for each word: query (Q), key (K), and value (V). The attention weights are calculated using these vectors, determining how much focus each word in a sentence should have on every other word. This is done multiple times in parallel (hence 'multi-head'), allowing the model to jointly attend to information from different representation subspaces at different positions. After the attention calculation, the outputs are concatenated and linearly transformed. The second component, the position-wise feed-forward network, is a fully connected layer applied to each position separately and identically. It consists of two linear transformations with a ReLU activation in between. To facilitate training and enable the model to learn the order of the sequence, positional encodings are added to the input embeddings at the bottom of the encoder stack. These encodings have the same dimension as the embeddings, ensuring that the two can be summed. The Transformer encoder's design, notably its reliance on self-attention and absence of recurrence, allows for highly parallelized processing and a more nuanced understanding of sequence relationships, a significant advancement in handling sequential data.

Question 4. [3+3+4]

a) Explain why traditional RNNs struggle with to maintain long-term dependencies?

Traditional Recurrent Neural Networks (RNNs) struggle with long-term dependencies primarily due to the vanishing gradient problem, where gradients become exceedingly small during backpropagation, making it hard to learn connections between distant data points. Conversely, the exploding gradient problem can cause unstable learning with excessively large weight updates. RNNs also inherently have limited memory capacity, struggling to retain information from long sequences, and their sequential processing nature leads to computational inefficiencies, further hindering their ability to capture long-term dependencies effectively. Advanced architectures like LSTMs and GRUs have been developed to address these shortcomings by incorporating mechanisms for selective memory retention.

b) Explain the concept of RoIAlign as used in Mask R-CNN and describe how it differs from the RoIPool operation used in earlier models like Fast R-CNN.

RoIPool works by dividing the proposal area into a fixed number of equal-sized sections (bins), and then max-pooling is performed in each bin. This method, however, leads to quantization errors because it involves rounding off the coordinates of the region of interest (RoI) to the nearest integer before pooling, causing misalignments between the RoI and the extracted features. These errors particularly affect the performance in tasks requiring precise spatial locations, such as instance segmentation.

RoIAlign, introduced in Mask R-CNN, addresses this issue by avoiding any quantization of coordinates. Instead of rounding, RoIAlign uses bilinear interpolation to compute the exact values at fractional pixel locations, allowing for more precise extraction of features. This method preserves the spatial details, which is critical for generating high-quality object masks. By aligning the extracted features accurately with the input, RoIAlign significantly improves the performance of the model in instance segmentation tasks. The improved alignment leads to better localization of objects and more accurate segmentation masks, making RoIAlign a key advancement over the RoIPool operation used in earlier models.

c) Explain about working of Yolo V1 algorithm. Also write about the loss function(s) employed.

YOLO (You Only Look Once) version 1 is a groundbreaking approach in the field of object detection, known for its speed and efficiency. Unlike traditional methods that typically involve separate stages for generating region proposals and classifying them, YOLO integrates these two stages into a single convolutional neural network (CNN), enabling it to process images in a single pass – hence the name "You Only Look Once".

1. Single Convolutional Network: YOLO uses a single CNN to predict multiple bounding boxes and class probabilities for those boxes simultaneously. The network divides the input image into an $S \times S$ grid. Each grid cell is responsible for predicting a fixed number of bounding boxes.

2. Bounding Box Prediction: Each grid cell predicts bounding boxes and confidence scores for those boxes. The confidence score reflects the accuracy of the bounding box and whether the box contains an object.

3. Class Prediction: In addition to bounding boxes, each grid cell predicts the conditional class probabilities (given that there's an object) for each class.

4. Combining Predictions: For each bounding box, the conditional class probabilities are multiplied by the confidence score to get class-specific confidence scores, which encode both the probability of the class being present and the accuracy of the box.

The loss function in YOLO V1 is a critical component that handles the errors in the bounding box predictions and class predictions. It consists of several parts:

1. Bounding Box Regression Loss YOLO uses mean squared error for the bounding box coordinates (x, y, width, height). It gives more weight to the loss associated with the bounding box coordinates to emphasize accuracy in localization.

2. Object Confidence Loss This part of the loss penalizes confidence score errors for boxes containing objects. It emphasizes the accuracy of the confidence score when an object is present in the box.

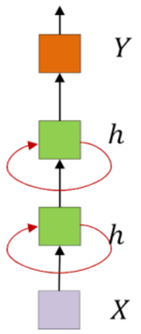
3. No-Object Confidence Loss YOLO also penalizes confidence scores for boxes that do not contain objects. Since most grid cells do not contain objects, this part helps in reducing false positives.

4. Classification Loss: This is the squared error for the class predictions (conditional class probability). It is only computed for the grid cells that actually contain an object.

The total loss function is a weighted sum of these components. YOLO places more weight on the localization loss (bounding box regression loss) and less on the confidence loss for boxes without objects, balancing the contribution of each part to the overall loss. This loss function enables YOLO to be trained end-to-end to perform object detection efficiently and accurately.

Question 5. [3+2+5]

a) Represent the following network in mathematical form.



$$h_i^{(1)}(t) = f_1\left(\sum_j w_{ji}^{(1)} X_j(t) + \sum_j w_{ji}^{(11)} h_i^{(1)}(t-1) + b_i^{(1)}\right)$$

$$h_i^{(2)}(t) = f_2\left(\sum_j w_{ji}^{(2)} h_j^{(1)}(t) + \sum_j w_{ji}^{(22)} h_i^{(2)}(t-1) + b_i^{(2)}\right)$$

$$Y(t) = f_3\left(\sum_j w_{jk}^{(3)} h_j^{(2)}(t) + b_k^{(3)}, k = 1..M\right)$$

b) Describe the use of non-max suppression algorithm in Yolo.

Discard all the boxes having probabilities less than or equal to a pre-defined threshold (say, 0.5)
For the remaining boxes:
Pick the box with the highest probability and take that as the output prediction
Discard any other box which has IoU greater than the threshold with the output box from the above step
Repeat step 2 until all the boxes are either taken as the output prediction or discarded

c) Given the following data, compute the output of forget gate, input gate, new hidden state and new cell state for the current input.

1. $Forgetgate: (f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f))$
2. $Inputgate: (i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i))$
3. $Candidatecellstate: (\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C))$
4. $Newcellstate: (C_t = f_t * C_{t-1} + i_t * \widetilde{C_t})$
5. $Outputgate: (o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o))$
6. $Newhiddenstate: (h_t = o_t * \tanh(C_t))$

Given Data:
$-Input\ at\ time(t), (x_t = 0.5)$
$-Previous\ hidden\ state, (h_{t-1} = -0.3)$
$-Previous\ cell\ state, (C_{t-1} = 0.6)$
$-Forgetgateweight(use\ same\ weights\ for\ input\ and\ previous\ hidden\ state),$
$(W_f = 0.9), bias(b_f = 0.1)$
$-Input\ gate\ weight, (W_i = 0.8), bias(b_i = -0.1)$
$-Candidate\ cell\ state\ weight, (W_C = -0.5), bias(b_C = 0.2)$
$-Output\ gate\ weight, (W_o = 0.3), bias(b_o = 0.1)$

$\mathbf{1}.\boldsymbol{ForgetGate}((\boldsymbol{f_t}))$:
$-Formula: (f_t = \sigma\big(W_{f_h}h_{t-1} + W_{f_x}x_t + b_f\big))$
$-GivenValues: (W_{f_h} = 0.9, W_{f_x} = 0.9, h_{t-1} = -0.3, x_t = 0.5, b_f = 0.1)$
$-Calculation: (f_t = \sigma(0.9\ \text{x}(-0.3) + 0.9\ \text{x}\ 0.5 + 0.1))$
$-Output: (f_t \approx 0.731)$

$\mathbf{2}.\boldsymbol{InputGate}((\boldsymbol{i_t}))$:
$-Formula: (i_t = \sigma\big(W_{i_h}h_{t-1} + W_{i_x}x_t + b_i\big))$
$-GivenValues: (W_{i_h} = 0.8, W_{i_x} = 0.8, h_{t-1} = -0.3, x_t = 0.5, b_i = -0.1)$
$-Calculation: (i_t = \sigma(0.8\ \text{x}\ (-0.3) + 0.8\ \text{x}\ 0.5 - 0.1))$
$-Output: (i_t \approx 0.524)$

$3. CandidateCellState\big((\widetilde{C_t})\big)$:
$-Formula: (\widetilde{C_t} = \tanh\big(W_{C_h}h_{t-1} + W_{C_x}x_t + b_C\big))$
$-GivenValues: (W_{C_h} = -0.5, W_{C_x} = -0.5, h_{t-1} = -0.3, x_t = 0.5, b_C = 0.2)$
$-Calculation: (\widetilde{C_t} = \tanh(-0.5 \cdot -0.3 + -0.5 \cdot 0.5 + 0.2))$
$-Output: (\widetilde{C_t} \approx -0.462)$

$4. NewCellState((C_t))$:
$-Formula: (C_t = f_t * C_{t-1} + i_t * \widetilde{C_t})$
$-GivenValues: (C_{t-1} = 0.6)$
$-Calculation: (C_t = 0.731 * 0.6 + 0.524 * -0.462)$
$-Output: (C_t \approx 0.393)$

$5. OutputGate((o_t))$:
$-Formula: (o_t = \sigma\big(W_{o_h}\ h_{t-1} + W_{o_x}x_t + b_o\big))$
$-GivenValues: (W_{o_h} = 0.3, W_{o_x} = 0.3, h_{t-1} = -0.3, x_t = 0.5, b_o = 0.1)$
$-Calculation: (o_t = \sigma(0.3 \cdot -0.3 + 0.3 \cdot 0.5 + 0.1))$
$-Output: (o_t \approx 0.539)$

$\mathbf{6}.\boldsymbol{NewHiddenState}((\boldsymbol{h_t}))$:
$-Formula: (h_t = o_t * \tanh(C_t))$
$-Calculation: (h_t = 0.539 * \tanh(0.393))$
$-Output: (h_t \approx 0.202)$