PVP 20

Code: 20AM3602, 20DS3603

III B.Tech - II Semester – Regular Examinations - APRIL 2025

DEEP LEARNING
(Common for AIML & DS)

Duration: 3 hours Max. Marks: 70

Note: 1. This paper contains questions from 5 units of Syllabus. Each unit carries 14 marks and have an internal choice of Questions.
2. All parts of Question must be answered in one place.

BL – Blooms Level CO – Course Outcome

			BL	CO	Max. Marks
		UNIT-I			
1	a)	Explain sigmoid, tanh, relu and softmax activation functions.	L2	CO1	7 M
	b)	What is regularization in Deep Learning? Explain the Dropout technique and its importance.	L2	CO1	7 M
		OR			
2	a)	Explain the differences between binary cross-entropy, categorical cross-entropy and sparse categorical cross-entropy.	L2	CO1	7 M
	b)	Differentiate between batch normalization and layer normalization.	L2	CO1	7 M
		UNIT-II			
3	a)	Illustrate key components of a Convolutional Neural Networks (CNNs) with an example.	L3	CO2	7 M

	b)	Explain how normalization improves the performance of convolutional neural networks with an example.	L2	CO2	7 M
		OR			
4	a)	Illustrate hyper-parameter tuning in convolutional neural networks.	L3	CO2	7 M
	b)	Demonstrate how stride affects the feature extraction in convolutional neural networks with an example.	L3	CO2	7 M
		UNIT-III			
5	a)	Explain the AlexNet architecture with an example.	L2	CO2	7 M
	b)	Analyze the merits and demerits of the ResNet architecture.	L4	CO4	7 M
		OR			
6	a)	Explain the Inception architecture with an example.	L2	CO2	7 M
	b)	Analyze the merits and demerits of the VGGNet architecture.	L4	CO4	7 M
		UNIT-IV			
7	a)	Demonstrate the trade-offs of using a pre-trained model instead of training a model from scratch.	L3	CO3	7 M
	b)	Illustrate how a Bidirectional Recurrent Neural Networks improves context understanding in natural language processing tasks with an example.	L3	CO3	7 M

		OR			
8	a)	Explain Gated Recurrent Unit (GRU) architecture.	L2	CO3	7 M
	b)	Demonstrate the role of decoder in encoder-decoder architecture and the limitations of encoder-decoder architecture.	L3	CO3	7 M
		UNIT-V			
9	a)	Compare and contrast the architectures of the generator and discriminator in Generative Adversarial Networks (GANs).	L4	CO4	7 M
	b)	Demonstrate the key challenges in training Generative Adversarial Networks (GANs).	L3	CO3	7 M
		OR			
10	a)	Discuss the applications of Generative Adversarial Networks (GANs).	L2	CO1	7 M
	b)	Compare Deep Convolutional GANs (DCGANs) and Cycle-Consistent GANs (CycleGANs).	L4	CO4	7 M

PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

III B.Tech. – I Sem- Regular Examinations

APRIL 2025

DEEP LEARNING – SCHEME OF EVALUATION

(Common for AIML and DS)

Subject code: 20AM3602,20DS3603

I. Short Scheme

Q.No	Question		CO - BL	Total Marks
	UNIT-I			
1.(a)	Explain sigmoid, tanh, relu and softmax activation functions.		CO1-L2	7M
	sigmoid ,tanh,relu and softmax activation functions explanation	7M		
1.(b)	What is regularization in Deep Learning? Explain the Dropout technique and its importance.		CO1-L2	7M
	Regularization explanation	4M		
	Dropout technique explanation	3M		
2(a)	Explain the differences between binary cross-entropy, categorical cross-entropy and sparse categorical cross-entropy.		CO1-L2	7M
	Binary cross-entropy explanation	3M		
	Categorical cross-entropy and Sparse Categorical cross-entropy types explanation	4M		
2(b)	Differentiate between batch normalization and layer normalization		CO1-L2	7M
	Batch normalization and Layer normalization explanation	5M		
	Any 2 differences	2M		

UNIT-II				
3(a)	**Illustrate key components of a Convolution Neural Network(CNNs) with an example.**		**CO2-L3**	**7M**
	Explanation of any 3 components of CNN	**6M**		
	Example	**1M**		
3(b)	**Explain how normalization improves the performance of convolutional neural networks with an example.**		**CO2-L2**	**7M**
	Normalization Definition	**2M**		
	Explanation	**5M**		
4(a)	**Illustrate hyper-parameter tuning in convolutional neural networks.**		**CO2-L3**	**7M**
	Hyper Parameters definition	**2M**		
	Explanation of types of hyper parameters and ways of tuning	**5M**		
4(b)	**Demonstrate how stride affects the feature extraction in convolutional neural networks with an example.**		**CO2-L3**	**7M**
	Stride definition	**2M**		
	Explanation of how stride affects the feature extraction with example	**5M**		
UNIT - III				
5(a)	**Explain the AlexNet architecture with an example.**		**CO2-L2**	**7M**
	AlexNet architecture explanation	**5M**		
	Diagram	**2M**		
5(b)	**Analyze the merits and demerits of ResNet architecture**		**CO4-L4**	**7M**
	ResNet explanation	**4M**		
	Any 3 merits and demerits	**3M**		

6(a)	Explain the Inception architecture with an example.		CO2-L2	7M
	Inception architecture explanation	5M		
	Diagram	2M		
6(b)	Analyze the merits and demerits of VGGNet architecture		CO4-L4	7M
	VGGNet explanation	4M		
	Any 2 merits and demerits	3M		
	UNIT-IV			
7(a)	Demonstrate the trade-offs of using a pre-trained model instead of training a model from scratch		CO3-L3	7M
	Pre-trained models definition	2M		
	Explanation of trade-offs	5M		
7(b)	Illustrate how a Bi directional recurrent neural network improves context understanding in natural language processing tasks with an example.		CO3-L3	7M
	Bi directional RNN explanation	6M		
	Example	1M		
8(a)	Explain Gated Recurrent Unit(GRU) architecture		CO3-L2	7M
	Gated Recurrent Unit(GRU) architecture explanation	6M		
	Diagram	1M		
8(b)	Demonstrate the role of decoder in encoder – decoder architecture and the limitations of encoder – decoder architecture		CO3-L3	7M
	Decoder explanation in encoder – decoder architecture	5M		
	Any 2 limitation	2M		
	UNIT - V			
9(a)	Compare and Contrast the architecture of the generator and discriminator in Generative Adversarial Networks(GANs)		CO4-L4	7M

	Generator architecture explanation	4M		
	Discriminator architecture explanation	3M		
9(b)	Demonstrate the key challenges in training Generative Adversarial Networks(GANs)		CO3-L3	7M
	Any 3 challenges in training Generative Adversarial Networks(GANs) explanation	7M		
10(a)	Discuss the applications of Generative Adversarial Networks(GANs)		CO1-L2	7M
	Any 3 applications in training Generative Adversarial Networks(GANs) explanation	7M		
10(b)	Compare Deep Convolutional GANs(DCGANs) and Cycle Consistent GANs(Cycle GANs)		CO4-L4	7M
	DCGANs explanation	4M		
	Cycle GANs explanation	3M		

II. Detailed Scheme

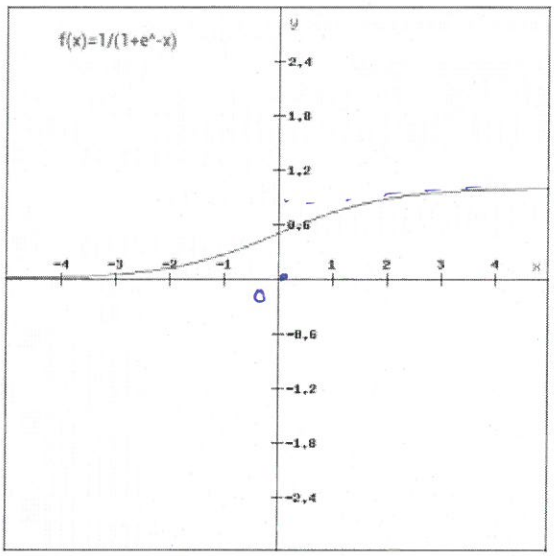
UNIT - I

1(a) Explain sigmoid, tanh, relu and softmax activation functions. **7M**

sigmoid ,tanh,relu and softmax activation functions **explanation**	**7M**

Ans: **Sigmoid Activation Function**

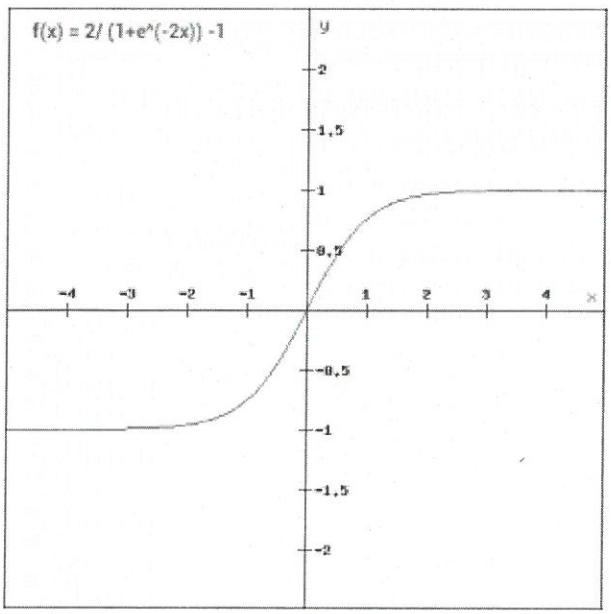
The next activation function in deep learning that we are going to look at is the Sigmoid activation function. It is one of the most widely used non-linear activation function. Sigmoid transforms the values between the range 0 and 1. Here is the mathematical expression for sigmoid- f(x) = 1/(1+e^-x)



$$f(x) = \frac{1}{1+e^{-x}}$$

Tanh

The tanh function is very similar to the sigmoid function. The only difference is that it is symmetric around the origin. The range of values in this case is from -1 to 1. Thus the inputs to the next layers will not always be of the same sign.

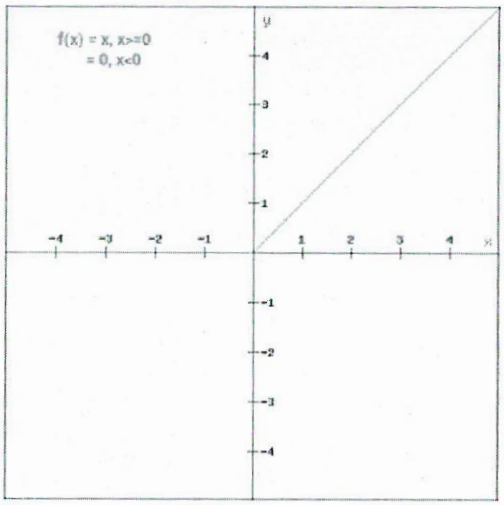


$$f(x) = \frac{e^{x}+e^{-x}}{e^{x}+e^{-x}}$$

ReLU Activation Function

The ReLU Activation function is another non-linear activation function that has gained popularity in the deep learning domain. ReLU stands for Rectified Linear Unit. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

$f(x) = \max(0,x)$



$f(x) = x, \quad x > 0$

$= 0, \quad x \leq 0$

Softmax

Softmax function is often described as a combination of multiple sigmoids. We know that sigmoid returns values between 0 and 1, which can be treated as probabilities of a data point belonging to a particular class. Thus sigmoid is widely used for binary classification problems.

The softmax function can be used for multiclass classification problems.

1(b) What is regularization in Deep Learning? Explain the Dropout technique and its importance. **7M**

Regularization explanation	**4M**
Dropout technique explanation	**3M**

Ans: Overfitting is the biggest issue in training any model. Regularization is a technique used in machine learning and deep learning to prevent overfitting and improve a model's generalization performance. It involves adding a penalty term to the **loss function** during training. Regularization in deep learning methods includes L1 and L2 regularization, dropout, early stopping, and more. By applying regularization for deep learning, models become more robust and better at making accurate predictions on unseen data.L1 and L2 are the most common types of regularization deep learning. L1 regularization is also called lasso regression. L2, This is known as ridge regression

DropOut:

Dropout is a regularization technique which involves randomly ignoring or "dropping out" some layer outputs during training, used in deep neural networks to prevent overfitting. Dropout is implemented per-layer in various types of layers like dense fully connected, convolutional, and recurrent layers, excluding the output layer. During training, dropout randomly deactivates a chosen proportion of neurons (and their connections) within a layer. This essentially temporarily removes them from the network.The deactivated neurons are chosen at random for each training iteration. This randomness is crucial for preventing overfitting.

OR

2(a) Explain the differences between binary cross-entropy, categorical cross-entropy and sparse categorical cross-entropy. 7M

Binary cross-entropy explanation	**3M**
Categorical cross-entropy and Sparse Categorical cross-entropy types explanation	**4M**

Ans:

1. Binary Cross Entropy:

It is used in binary classification problems like two classes. example a person has covid or not or my article gets popular or not. Binary cross entropy compares each of the predicted probabilities to the actual class output which can be either 0 or 1 .

2.Categorical Cross Entropy:

The Categorical crossentropy loss function is used to compute loss between true labels and predicted labels.It's mainly used for multiclass classification problems. For example Image classification of animal-like cat, dog, elephant, horse, and human.

3.Sparse Categorical Crossentropy:

It is used when there are two or more classes present in our classification task. similarly to categorical crossentropy. But there is one minor difference, between categorical crossentropy and sparse categorical crossentropy that's in sparse categorical cross-entropy labels are expected to be provided in integers.

2(b) Differentiate between batch normalization and layer normalization 7M

Batch normalization and Layer normalization explanation	**5M**
Any 2 differences	**2M**

Ans:

1. **Batch Normalisation (BN):**
   Batch Normalization is a technique used to normalize the activations of a layer within a mini-batch during training.It is commonly applied to intermediate layers of a neural network to help stabilize training, reduce internal covariate shift, and improve the convergence of the model.In essence, BN acts on the batch dimension, hence the name "Batch Normalization."

2. **Layer Normalization (LN):**
   Layer Normalization, on the other hand, is a technique used to normalize the activations of a layer across the entire layer, independently for each sample in the batch. It is often used in recurrent neural networks (RNNs) or transformers (including attention architectures) where the concept of "batch" may not be well-defined due to varying sequence lengths or when the batch size is limited to. The normalization is done per training example.

UNIT – II

3(a) Illustrate key components of a Convolution Neural Network(CNNs) with an example. 7M

Explanation of any 3 components of CNN	**6M**
Example	**1M**

Ans: Convolution Neural Network

Convolutional Neural Network (CNN) is the extended version of artificial neural networks

CNN Architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

Input Layers: It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image

Convolutional Layers: This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices. It slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch.

Pooling layer: This layer is used to reduce the spatial dimensions of feature map and also prevents overfitting.

Stride: Stride refers to the number of pixels by which a kernel moves across the input image.

Padding: padding is a technique used to preserve the spatial dimensions of the input image after convolution operations on a feature map. Padding is the process of adding extra pixels around the border of the input feature map

Fully Connected Layers: It takes the input from the previous layer and computes the final classification or regression task.

Output Layer: The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax

3(b) Explain how normalization improves the performance of convolutional neural networks with an example. 7M

Normalization Definition	**2M**
Explanation	**5M**

Ans: Normalization in CNNs

Normalization refers to techniques that adjust the inputs to layers within the neural network to make training more stable and efficient. The most common normalization techniques in CNNs are Batch Normalization and Layer Normalization.

Batch Normalization (BN) : Batch Normalization (BN) was introduced to address issues like vanishing/exploding gradients and to speed up training. It normalizes the activations of each layer so that the network doesn't face drastic changes in the distribution of activations across layers.

Layer Normalization: While Batch Normalization normalizes across the batch dimension, Layer Normalization normalizes across the feature dimension for each individual input sample. This process transforms the activations to have a mean of 0 and variance of 1. Normalization techniques like Batch Normalization stabilize training, reduce internal covariate shift, and allow for faster convergence by normalizing activations.

Example: Without Normalization the Pixel values are in the range **[0, 255]**. The network starts with random weights. With Normalization the pixel values by scaling them to **[0, 1]** (e.g., dividing by 255) or standardizing them to zero mean and unit variance.

OR

4(a) Illustrate hyper-parameter tuning in convolutional neural networks. 7M

Hyper Parameters definition	**2M**
Explanation of types of hyper parameters and ways of tuning	**5M**

Ans: Hyperparameter tuning in Convolutional Neural Networks (CNNs) is a crucial step in the model development process. Hyperparameters are parameters whose values are set before the learning process begins, and their optimal choice can significantly affect the performance of the model. In CNNs, hyperparameters include architecture-specificparameters (like filter size, number of layers), training parameters (like learning rate,batch size), and regularization parameters (like dropout rate).

Types of Hyperparameters in CNNs

1. Learning rate
2. Number of layers
3. Number of Epochs
4. Activation Functions
5. Number of filters (kernels)
6. Stride
7. Padding
8. Batch Size
9. Optimizer

Methods for Hyperparameter Tuning in CNNs

1. Grid Search
2. Random Search
3. Bayesian Optimization

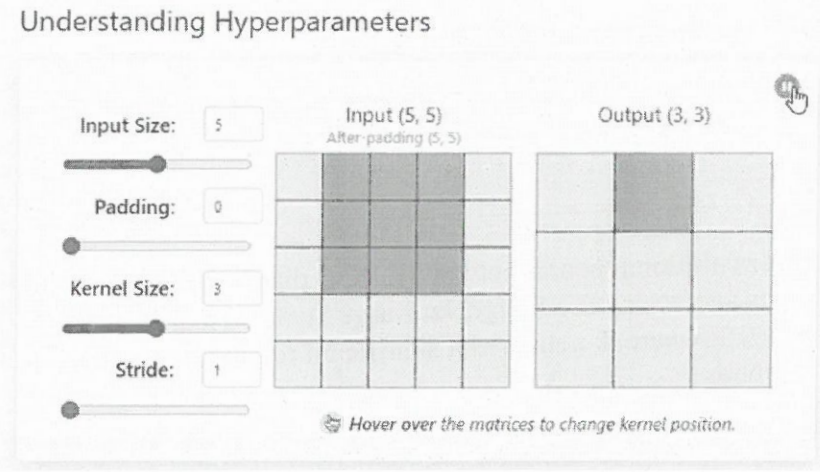
4(b) Demonstrate how stride affects the feature extraction in convolutional neural networks with an example. **7M**

Stride definition	**2M**
Explanation of how stride affects the feature extraction with example	**5M**

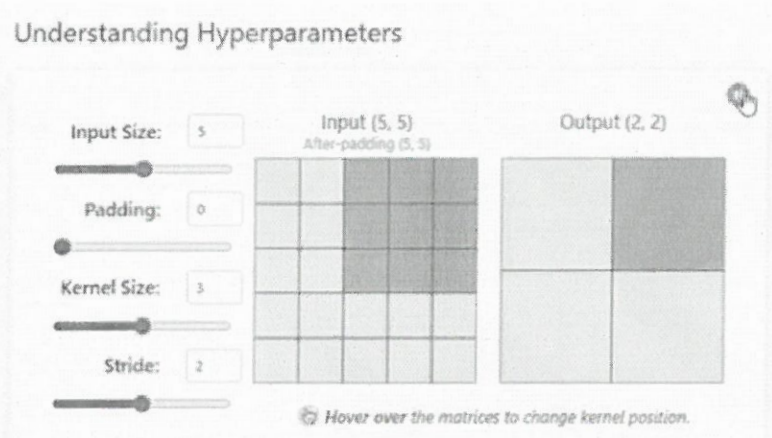
Ans: Stride:

Stride refers to the number of pixels by which a kernel moves across the input image.

For Example when Stride = 1 (**Students can explain with any example of their choice this is just sample one**)



When Stride = 2



A stride of 2 not only changes the way the convolution iterates over the input size but also the output by making it smaller (2 × 2). A larger stride will produce smaller output dimensions (as it covers the input image faster), whereas a smaller stride results in a larger output dimension.

Increasing the stride will allow the filter to cover a larger area of the input image, which can be useful for capturing more global features. In contrast, lowering the stride will capture finer and more local details. In addition, increasing the stride will control overfitting and reduce computational efficiency as it will reduce the spatial dimensions of the feature map.

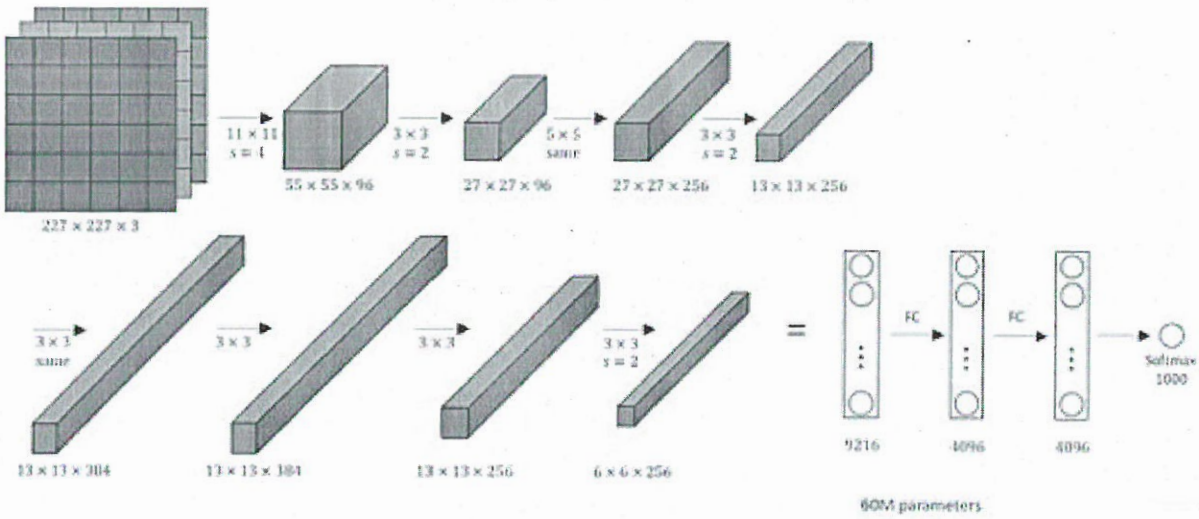
UNIT - III

5(a) Explain the AlexNet architecture with an example. **7M**

AlexNet architecture explanation	**5M**
Diagram	**2M**

Ans: AlexNet is a deep convolutional neural network (CNN) that was introduced in 2012 . It is widely recognized for its success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, where it achieved a significant reduction in error rate compared to previous methods.

AlexNet consists of 5 convolution layers, 3 max-pooling layers, 2 Normalized layers, 2 fully connected layers and 1 SoftMax layer. Each convolution layer consists of a convolution filter and a non-linear activation function called "ReLU". The pooling layers are used to perform the max-pooling function and the input size is fixed due to the presence of fully connected layers. The input size is mentioned at most of the places as 224x224x3 but due to some padding which happens it works out to be 227x227x3. Above all this AlexNet has over 60 million parameters



Key properties:

GPU Use: AlexNet's use of GPUs for parallel training revolutionized deep learning.

ReLU Activation: The use of ReLU gives better performance.
Data Augmentation: Techniques like image flipping and cropping elped in preventing overfitting.
Dropout Regularization: Dropout was a novel technique at the time for reducing overfitting in large networks.

(Any Example of their choice)

5(b) Analyze the merits and demerits of ResNet architecture **7M**

ResNet explanation	**4M**
Any 3 merits and demerits	**3M**

Ans: ResNet (Residual Networks) is a type of deep neural network architecture that introduces the concept of residual learning to address the problem of vanishing/exploding gradients and degradation in deeper networks. Below are some key features and properties of ResNet:

Properties of ResNet:

ResNet (Residual Networks) is a type of deep neural network architecture that introduces the concept of residual learning to address the problem of vanishing/exploding gradients and degradation in deeper networks.
Below are some key features and properties of ResNet:

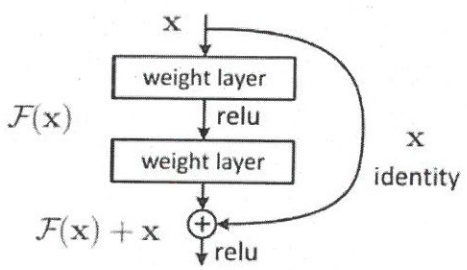
1. Residual Learning (Skip Connections):

2. Deep Network Capability:

3. Identity Mapping:

4. **Residual Network:** In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called **Residual Blocks**. In this network, we use a technique called ***skip connections***. The skip connection connects activations of a layer to further layers by skipping some layers in between. This forms a residual block. Resnets are made by stacking these residual blocks together.

$H(x) = F(x) + x.$



Skip (Shortcut) connection

Merits:

- Improved Accuracy
- Skip connection
- Residual Blocks
- Performance
- Vanishing Gradient Problem Mitigation
- Versatility

Demerits:

- Complex architecture
- Increased Computational and Memory Requirements
- Overfitting in Small Datasets
- Residual Connections May Not Always Be Effectiv e
- Gradient Explosion or Vanishing Problems
- Lack of Interpretability
- Over-Complexity in Some Applications
- Difficulty in Training Extremely Deep Models

OR

6(a) Explain the Inception architecture with an example. **7M**

Inception architecture explanation	**5M**
Diagram	**2M**

Ans: Inception V1 also known as GoogleNet was proposed by research at Google (with the collaboration of various universities) in 2014 This architecture was the winner at the ILSVRC 2014 image classification challenge.

Properties:

Factorized Convolutions, Global Average Pooling , Concept of Inception Module, Auxiliary Classifier for Training

High-Level Architecture of Inception :

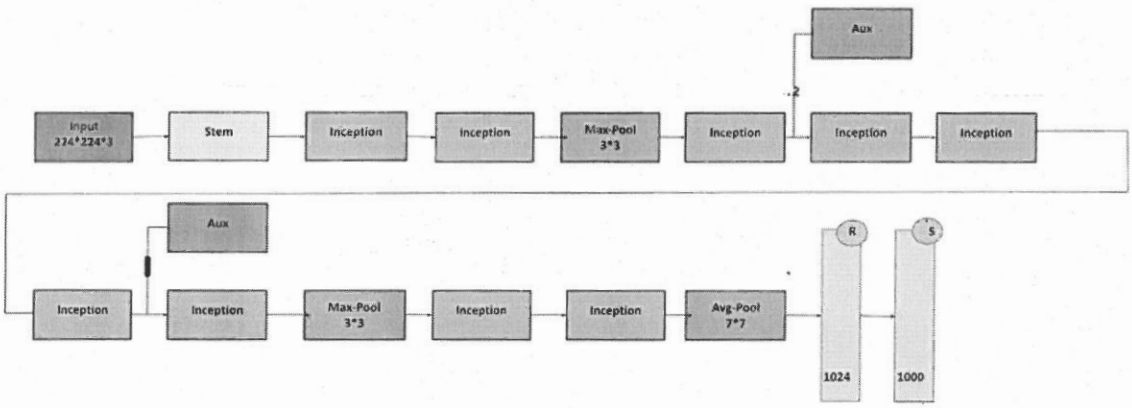
Input Layer: Image input, typically resized to 224x224x3 (for RGB images). •

Multiple Inception Modules: A series of Inception modules that learn hierarchical features.

Auxiliary Classifiers: Additional classifiers at intermediate stages to aid in training.

Global Average Pooling: After the convolutional layers, global average pooling is applied.

Final Softmax Layer: This layer outputs the final class probabilities



6(b)Analyze the merits and demerits of VGGNet architecture **7M**

VGGNet explanation	**4M**
Any 2 merits and demerits	**3M**

Ans:**VGG-16**

The VGG-16 model is a convolutional neural network (CNN) architecture that was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is characterized by its depth, consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers.

VGG Architecture:

The VGG-16 architecture is a deep convolutional neural network (CNN) designed for image classification tasks. It was introduced by the Visual Geometry Group at the University of Oxford. VGG-16 is characterized by its simplicity and uniform architecture, making it easy to understand and implement.

The VGG-16 configuration typically consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. These layers are organized into blocks, with each block containing multiple convolutional layers followed by a max-pooling layer for downsampling.

VGG-19 Architecture

VGG-19 is a deep convolutional neural network with 19 weight layers, comprising 16 convolutional layers and 3 fully connected layers. The architecture follows a straightforward and repetitive pattern, making it easier to understand and implement

Merits:

- Improved Accuracy
- Simplicity and Uniform Architecture
- Small Receptive Fields with Deep Stacking
- Transfer Learning Friendly

Demerits:

- It is very slow to train (the original VGG model was trained on Nvidia Titan GPU for 2- 3 weeks).
- The size of VGG-16 trained imageNet weights is *528* MB. So, it takes quite a lot of disk space and bandwidth which makes it inefficient.
- 138 million parameters lead to exploding gradients problem.

UNIT-IV

7(a) Demonstrate the trade-offs of using a pre-trained model instead of training a model from scratch. **7M**

Pre-trained models definition	**2M**
Explanation of trade-offs	**5M**

Ans:

Using a pre-trained model :

This offers the advantage of saving time and resources since the model has already learned general features from large datasets like ImageNet. It requires less data, is quicker to fine-tune, and performs well on tasks similar to its original training.

Training a model from scratch :

This gives you complete control over the architecture and allows the model to fully adapt to your specific data, which is ideal for unique problems or domains. The trade-off is that it demands a large labeled dataset, longer training time, and powerful hardware resources to achieve good performance.

Scenario 1 – Size of the Data set is small while the Data similarity is very high

In this case, since the data similarity is very high, we do not need to retrain the model. All we need to do is to customize and modify the output layers according to our problem statement. We use the pretrained model as a feature extractor.

Scenario 2 – Size of the data is small as well as data similarity is very low

The small size of the data set is compensated by the fact that the initial layers are kept pretrained(which have been trained on a large dataset previously) and the weights for those layers are frozen.

Scenario 3 – Size of the data set is large however the Data similarity is very low

In this case, since we have a large dataset, our neural network training would be effective. Hence, its best to train the neural network from scratch according to your data.

Scenario 4 – Size of the data is large as well as there is high data similarity

This is the ideal situation. In this case the pretrained model should be most effective.

7(b) Illustrate how a Bi directional recurrent neural network improves context understanding in natural language processing tasks with an example. 7M

Bi directional RNN explanation	**6M**
Example	**1M**

Ans:

Bidirectional Recurrent Neural Networks (Bi-RNNs):

Bidirectional RNNs (Bi-RNNs) are an extension of standard RNNs that process data in both forward and backward directions. This means that, for each time step, the network has access to both past and future information, which can be useful for tasks where future context is important (e.g., in natural language processing for sentence understanding).

Structure of Bi-RNNs:

A bidirectional RNN consists of two RNNs:

1. **Forward RNN**: This processes the sequence from the beginning to the end (as in a standard RNN).
2. **Backward RNN**: This processes the sequence from the end to the beginning.

At each time step, the output of the Bi-RNN is a combination of the outputs from both the forward and backward RNNs. This allows the network to utilize information from both past and future states to make predictions. After processing the sentence in both directions, the Bi-RNN combines the hidden states from the forward and backward passes. This gives the model access to both past and future context for each word.

For example: (any sentence example of their choice)

Sample ex: Apple is my favorite _____.

Apple is my favourite _____, and I work there.

Apple is my favorite _____, and I am going to

buy one.

OR

8(a) Explain Gated Recurrent Unit(GRU) architecture **7M**

Gated Recurrent Unit(GRU) architecture explanation	**6M**
Diagram	**1M**

Ans: GRU stands for Gated Recurrent Unit, which is a type of recurrent neural network (RNN) architecture that is similar to LSTM (Long Short-Term Memory).Like LSTM, GRU is designed to model sequential data by allowing information to be selectively remembered or forgotten over time. However, GRU has a simpler architecture than LSTM, with fewer parameters, which can make it easier to train and more computationally efficient.The main difference between GRU and LSTM is the way they handle the memory cell state. In LSTM, the memory cell state is maintained separately from the hidden state and is updated using three gates: the input gate, output gate, and forget gate. In GRU, the memory cell state is replaced with a "candidate activation vector," which is updated using two gates: the reset gate and update gate.

The reset gate determines how much of the previous hidden state to forget, while the update gate determines how much of the candidate activation vector to incorporate into the new hidden state.

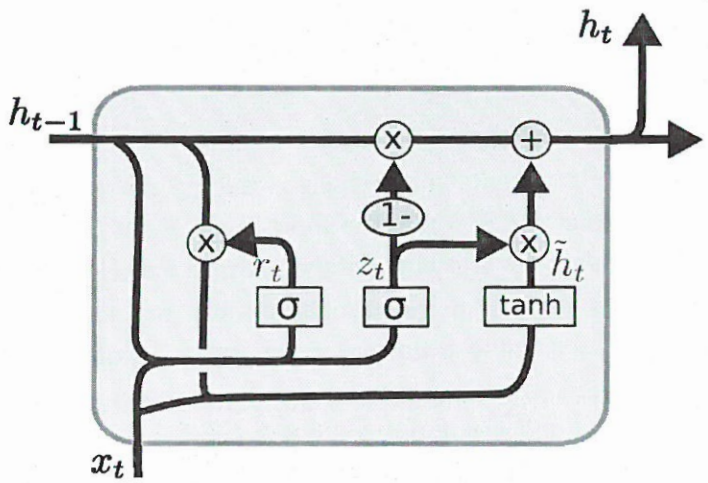
Overall, GRU is a popular alternative to LSTM for modeling sequential data, especially in cases where computational resources are limited or where a simpler architecture is desired.

The GRU consists of **two main gates**:

1. **Update Gate** (zt): This gate decides how much information from previous hidden state should be retained for the next time step.
2. **Reset Gate** (rt): This gate determines how much of the past hidden state should be forgotten.

These gates allow GRU to control the flow of information in a more efficient manner compared to traditional

Candidate activation vector: The candidate activation vector is a modified version of the previous hidden state that is "reset" by the reset gate and combined with the current input.



8(b) Demonstrate the role of decoder in encoder – decoder architecture and the limitations of encoder – decoder architecture **7M**

Decoder explanation in encoder – decoder architecture	**5M**
Any 2 limitation	**2M**

Ans: **Encoder**

- Multiple RNN cells can be stacked together to form the encoder. RNN reads each inputs sequentially
- For every timestep (each input) t, the hidden state (hidden vector) h is updated according to the input at that timestep X[i].
- After all the inputs are read by encoder model, the final hidden state of the model represents the context/summary of the whole input sequence.

Decoder

- The Decoder generates the output sequence by predicting the next output Yt given the hidden state ht.
- The input for the decoder is the final hidden vector obtained at the end of encoder model.
- Each layer will have three inputs, hidden vector from previous layer ht-1 and the previous layer output yt-1, original hidden vector h.
- The outputs occurred at each timestep of decoder is the actual output. The model will predict the output until the END symbol occurs.

9(a) Compare and Contrast the architecture of the generator and discriminator in Generative Adversarial Networks(GANs) **7M**

Generator architecture explanation	**4M**
Discriminator architecture explanation	**3M**

Ans: **Generator:**

The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real.vs fake.

Generator training requires tighter integration between the generator and the discriminator than discriminator training requires.

Discriminator:

The discriminator acts as a binary classifier, distinguishing between real and generated data. It learns to improve its classification ability through training, refining its parameters to detect fake samples more accurately.When dealing with image data, the discriminator often employs convolutional layers or other relevant architectures suited to the data type. These layers help extract features and enhance the model's ability to differentiate between real and generated samples.

9(b) Demonstrate the key challenges in training Generative Adversarial Networks(GANs) **7M**

Any 3 challenges in training Generative Adversarial Networks(GANs) explanation	**7M**

Ans: Challenges in GAN Training

Training GANs is sometimes difficult due to several challenges:

1. Mode Collapse: Mode collapse occurs when the generator starts producing a limited variety of outputs, rather than diverse examples that cover the entire data distribution.

2. Non Convergence

3. Gradient Instability: The gradient of the loss function with respect to the generator or discriminator can sometimes be very small

4. Hyperparameter Tuning: GANs are very sensitive to hyperparameters

5. Resource Intensive

6. Challenging for few datasets

7. Overfitting

8.Ethical and Legal Concerns

OR

10(a) Discuss the applications of Generative Adversarial Networks(GANs) **7M**

Any 3 applications in training Generative Adversarial Networks(GANs) explanation	**7M**

Ans: Applications of GAN

1.Image Synthesis & Generation: GANs generate realistic images, avatars, and high-resolution visuals by learning patterns from training data.

2.Image-to-Image Translation: GANs can transform images between domains while preserving key features.

3.Text-to-Image Synthesis: GANs create visuals from textual descriptions, enabling applications in AI-generated art, automated design, and content creation.

4.Data Augmentation: GANs generate synthetic data to improve machine learning models,making them more robust and generalizable, especially in fields with limited labeled data.

5.High-Resolution Image Enhancement: GANs upscale low-resolution images, improving clarity for applications like medical imaging, satellite imagery, and video enhancement.

6.Fake image generation

10(b) Compare Deep Convolutional GANs(DCGANs) and Cycle Consistent GANs(Cycle GANs) **7M**

DCGANs explanation	**4M**
Cycle GANs explanation	**3M**

Ans:

Deep Convolutional GANs: DCGANs apply specific architectural guidelines to standard GANs, particularly using convolutional layers to improve the generation of high-quality images. The generator network in a DCGAN consists of a series of transposed convolutional layers, with each layer progressively increasing the spatial dimensions of the input noise vector while decreasing the number of feature maps. The final layer outputs a synthetic image with the same dimensions as the real images in the dataset. The discriminator network in a DCGAN consists of a series ofconvolutional layers, with each layer progressively decreasing the spatial dimensions of the input image while increasing the number of feature maps. The final layer outputs a single value representing the probability that the input image is real.

CycleGAN (Cycle-Consistent Generative Adversarial Network):

This is a type of GAN designed for unpaired image-to-image translation. Unlike other models that require paired training data, CycleGAN can learn to translate images from one domain to another using unpaired datasets. This ability is particularly useful when paired datasets are unavailable or difficult to obtain.The core idea behind CycleGAN is cycle consistency. This

concept ensures that an image translated from domain X to domain Y and then back to domain X should be the same as the original image. This cycle consistency loss helps the model learn mappings that are more accurate and preserve the content of the images.