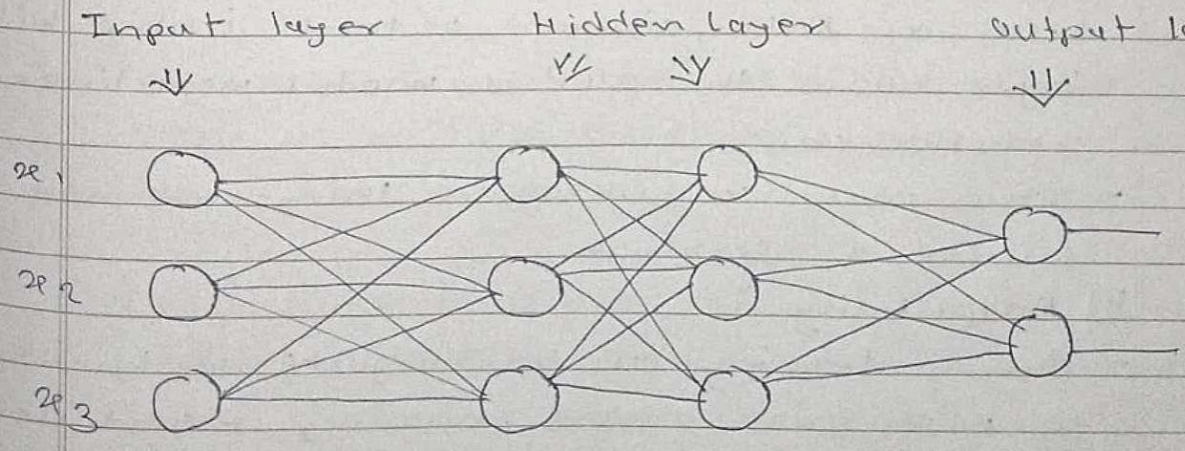
Deep Learning

M-1

Page No.
Date

Q. Multilayer Perceptron (MLP)

→ 1] Multilayer Perceptron was introduced to overcome the problems of single layer perceptron.
2] In single layer perceptron there is only one hidden layer that's why it was difficult for single layer perceptron to handle complex problems.
3] That's where the Multilayer Perceptron comes in place.
4] A fully connected Multilayer Neural Network is called as Multilayer Perceptron.
5] A Multilayer Neural Network has 3 layers including one input layer, one output layer and more than one hidden layer.
6] It is a typical example of Feed forward Network.
7] MLP is capable of handling the complex problems / relationship in data by adjusting the weights between the nodes during the training
8] Activation functions are applied to the output of nodes in each layer.



Q. Representation Power of Multilayer Perceptron.

→ 1) Universal Approximation
2) Hierarchy of Abstraction
3) Non-linear Activation function
4) Adaptibility
5)

1) Universal Approximation :
- An MLP, especially with multiple layers can approximate almost any type of relationship between input and output.
- It can capture wide range of complese relationship.

2) Hierarchy of Abstraction :
- The architecture of MLP with multiple hidden layer, enables them to learn hierarchical representation of data.
- Each layer can capture different levels of abstraction.

3) Non-linear Activation function :
- The use of non-linear activation functions such as Relu, sigmoid or tanh in the hidden layer allows the MLP to model non-linear relationship.
- It is crucial for tasks where things are not simple & straight

4) Adaptibility :
- During training an MLP adjusts it's knowledge to minimize errors, making it good at adapting new information.

Q. Gradient Descent

→ 1) Gradient Descent is nothing but a derivative that defines the effects on outputs of the function with the little bit of variation in input
2) It is widely used optimization algorithm in Deep learning that is used to minimize the cost function of Neural Network model during training.
3) It is used to optimize the weight & biases based on the cost function.
4) The cost function evaluates the difference between the actual & predicted output
5) The main aim of gradient descent is to minimize the cost function.
6) There are 3 Types of Gradient Descent
   i) Batch Gradient Descent
   ii) Stochastic Gradient Descent
   iii) Mini-Batch Gradient Descent.

i) Batch Gradient Descent : It uses the entire training Dataset to compute gradient of the loss function.

ii) Stochastic Gradient Descent : It updates parameters after each training sample, making it faster but noisy.

iii) Mini Batch Gradient Descent : It finds a balance between looking at the entire data and just one piece at a time.
- It compromise between Batch & stochastic Gradient descent.

8. Perceptron Learning Rule

→ 1) Perceptron is a basic building block used to build the ANN

2) It takes real valued input and calculates linear combination of these inputs and generates the output.

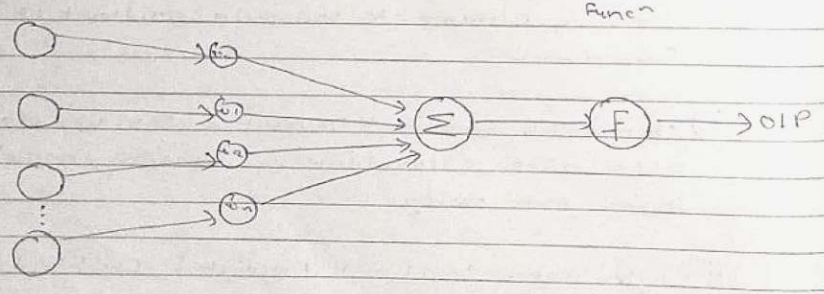
3) The output will be ~~one~~ equal to 1 if the result is greater than threshold else the output will be zero.

4) Perceptron learning is a simple algorithm used in supervised learning to train binary classifier

5) Perceptron learning rule works by updating weights in the neural network.

6) It works good only for the linearly separable data.

7) It has input layer, weight & Bias, net input ~~and~~, Activation function & output.



9. Feed forward Neural Network

→ 1) It is a type of a Artificial Neural Networks.

2) It was the first and the simplest type of ANN

3) In this ANN connections does not form a cycle.

4) In this ANN (Neural networks) information flows only in one direction that is forward direction.

5) When the data is transmitted through the input layer it goes through the hidden layer to the output layer

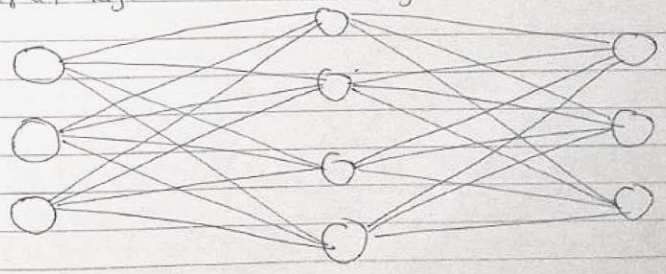
6) The job of hidden layer is to transform the input into something that output layer can use

7) The hidden layer uses activation function to map the resulting values between 0 to 1.

8) Hidden layer mostly uses Relu activation function

9) The output layer uses softmax activation function to map the value & to find the probabilities of output.

10) The inputs are fed by series of weights which is then computed by hidden layer.



Page No.
Date

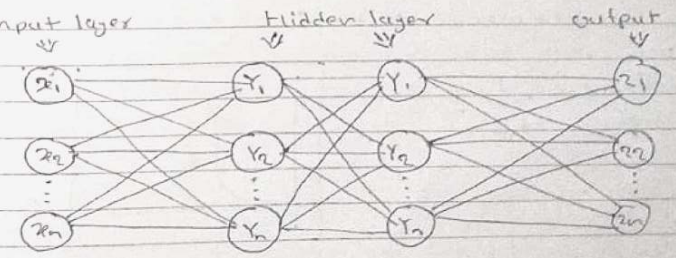
Q. Terminologies of Deep Learning

→ 1) Neural Network
2) ANN
3) DNN
4) Activation functions
5) Backpropagation
6) Gradient Decent
7) CNN
8) RNN
9) Transfer Learning
10) overfitting
11) Underfitting
12) Dropout
13) Batch Normalization
14) Loss function
15) Epoch
16) Supervised Learning
17) Unsupervised Learning
18) Auto encoder
19) GPU
20) LSTM
21) GAN

Page No.
Date

Q. Multilayer Feed forward Neural Network. (MFFNN).

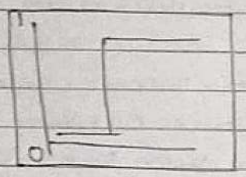
→ 1) A Multilayer Feed Forward Neural Networks is an interconnected ANN with multiple layers that has neurons with weight associated with them
2) ~~and~~ They compute the results using activation function.
3) The flow of Multilayer Feed Forward Neural Network is from input to output units.
4) The MFFNN does not have any loops, no feedback and no single moves in backward direction.
5) The MFFNN has multiple hidden layers that make makes it multilayer neural network.
6) And it is Feed forward because it is a network that follows top-down approach to train the network
7) ~~Multilayer~~ MFFNN includes Input layer, more than one hidden layer and output layer
8) The input & hidden layer uses sigmoid and linear activation function
9) The output layer uses step activation function.
10) Multiple hidden and output layer increases the accuracy of the output.



Q. Activation functions

→ 1) Step function
2) Sigmoid function
3) Tanh
4) Relu
5) Leaky Relu
6) Softmax function

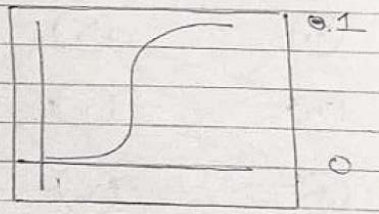
1) Step function :
- In this activation function the output will be either 0 or 1.
- If the input value is greater than or equal to threshold then the output will be 1, else output will be 0.



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

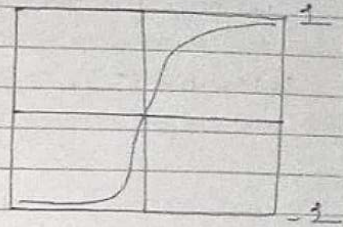
2) Sigmoid function :
- Sigmoid function is widely used in back propagation technique.
- The output range of this activation function is between 0 to 1.
-

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



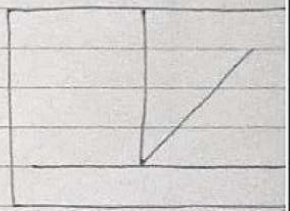
3) Tanh (Hyperbolic Tangent) :
- The output value range from -1 to 1
- F

$$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$$

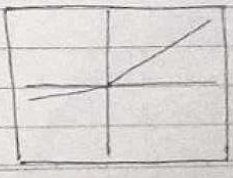


4) ReLu (Rectified linear unit) :
- It outputs the input value only if it is positive else it produces 0.
- It introduce the non-linearity to deep learning model.

$$f(x) = \begin{cases} 1 \text{ if } x \text{ is positive} \\ 0 \text{ if } x \text{ is negative} \end{cases}$$



5) Leaky ReLu :
- This is advance version of ReLu
- Unlike ReLu it outputs the value even if the input is negative but it gives very small value e.g. 0.001.



6) Softmax function :

- This activation function scales numbers into probabilities Distribution (Probabilities)

$$\text{Softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Q Explain Loss functions

→ 1) Loss function is a method of evaluating how well your algorithm is modeling your Dataset

2) It measures the error between actual value and predicted value.

3) It is used to optimize the model during training.

4) There are 3 Types of Loss functions

i) Mean Squared Error

ii) Mean Absolute Error

iii) cross Entropy Error

i) Mean squared Error :

- It is widely used loss function in Deep Learning and Statistics

- MSE measures the average difference between the predicted values & Actual values.

- It is commonly used in regression task.

- e.g. Temperature Prediction

ii) Mean Absolute Error :

- It was introduced to overcome the problems of MSE.

- It is commonly used loss function in regression task.

- It is used to quantify the absolute average difference between the actual values & Predicted values.

- It measures the accuracy of a regression model by calculating the average absolute difference between the actual & predicted valu

- e.g. House Price Prediction

iii) Cross Entropy Loss :-

- It is used for classification task
- It quantifies the difference between the Actual ~~probability~~ class probability and the

iii) Cross Entropy Loss :-

- It is used for classification task.
- It quantifies the difference between the predicted class probabilities and actual class labels.
- It is mainly used to train neural network for task like image classification & NLP
- e.g. Image classification.

Q. Back Propagation Algorithm

→ 1) The BackPropagation is a short term used for backward propagation of errors.

2) It is an algorithm that designed to test for error working back from output node to input nodes.

3) It is a important mathematical tool for improving the accuracy of predictions in data mining and machine learning.

4) Backpropagation is an algorithm used to quickly calculate derivatives in a neural networks.

5) There are 2 Types of Backpropagation

i) static Backpropagation

ii) Recurrent Backpropagation

i) static Backpropagation :- It is a network that is used to develope static input for static output

- static networks can solve the static classification problem, such as Optical character recognition (OCR).

ii) Recurrent Backpropagation :- This networks is used for fixed-point learning.

- This means that during neural networks training, the weights are numerical values that determine how much nodes -- also refered to as neurons influence output values.

6) The key difference here is that static networks offers instant mapping, while recurrent network does not.

7) ANN and CNN use backpropagation as a learning algorithm to compute gradient descent.
8) Backpropagation is a supervised Learning
9) Backpropagation algorithms are used extensively to train feed forward neural network such as CNN
10) It enables the use of gradient methods, such as gradient descent, stochastic GD, to train the multilayer ~~perceptron~~ networks and update weights to minimize the errors.

Applications
i) AI
ii) NLP
iii) OCR
iv) Image Processing etc.

Advantage
i) Highly Adaptable
ii) don't require prior knowledge about the n/w.
iii) User Friendly
iv) Fast & Easy

Disadvantage
i) Matrix based approach
ii) Sensitive to noisy data
iii) Performance is dependent on input data.

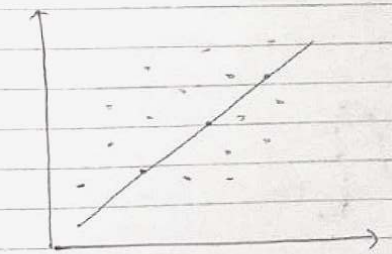
Q. Explain Underfitting & Overfitting

→ 1) Underfitting :
- Underfitting is a situation which occurs when a ML model is too simple to capture the underlying patterns in the training data.
- In other words Underfitting occurs when a model is too simple to capture the complexity.
- Underfitting causes poor performance of model.
- In simple term, an underfit models are inaccurate, especially when applied to new, unseen examples.

**Reasons for Underfitting

1) Unable to capture the complexity.
2) The size of training dataset used in not enough.
3) Features are not scaled

Techniques to Reduce Underfitting
1) Increase Model complexity
2) Increase the number of features
3) Remove the noisy data.
4) Increase the model training duration.



- It has low variance & high bias.

2) Overfitting :

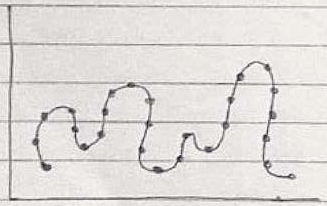
- Overfitting is a situation which occurs when your ML model tries to capture extra data-points ~~except from~~ or patterns given before the test.
- A model is said to be overfitted when the model does not make the accurate predictions on testing data, when gets trained with so much data.
- It has high variance & low bias
- The model is unable to categorise data correctly, because of too many details and noise.

Reasons for overfitting

- High variance & low bias
- The model is too complex
- The size of training data.

Technique to reduce Overfitting

- Reduce model complexity
- Early stopping during training phase
- Increase training data.



Q. Explain Types of Biases & variance

→ 1) Bias can be defined as the inability of the model because of that there is some difference or error occuring between the model's predicted value and the actual value.

2) These difference between the actual value & the predicted value are known as bias error.

3) Bias is a systematic error that occurs due to wrong assumptions in the ML process.

4) Bias $(\hat{Y}) = E(\hat{Y}) - Y$

where $\hat{Y}$ = Predicted value

Y = Actual value

$E(\hat{Y})$ = expected value of estimator $\hat{Y}$

5) There are 2 Types of Bias

i) Low Bias

ii) High Bias

i) Low Bias : - means fewer assumptions are taken to build the target function.

- In this case, the model will closely match the training dataset.

ii) High Bias : means more Assumptions are taken to build the targeted function.

- In this case, the model will not match the training dataset closely.

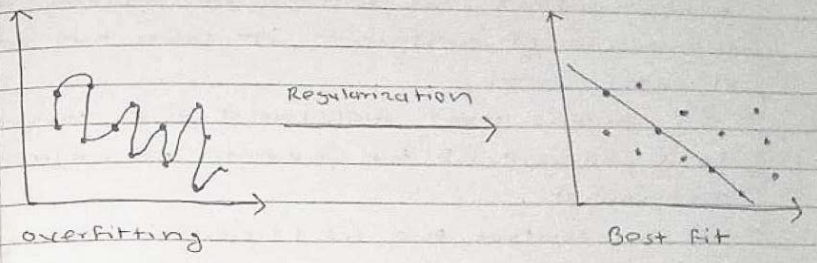
7) Bias refers to the gap between the predicted and actual value.

1] Variance can be defined as the measure of spread in data from its mean position.
2] A high variance model is overly complex.
3] Variance measures the model's sensitivity to ~~fact~~ fluctuations in the training dataset.
4] If a model has high variance it can lead to overfitting
5] If a model has high variance there is a chance that it can have noisy data.
6] variance = $E[(\hat{Y} - E(\hat{Y}))^2]$
where $E[\hat{Y}]$ = expected value of predicted values.
7] There are 2 Types of variance
i) Low variance
ii) High variance

i) Low variance ⇒ Means that the model is less sensitive to changes in training data.
- Low variance can lead to underfitting

ii) High variance ⇒ Means that the model is very sensitive to changes in training data.
- It fits the training data too closely that it fails on the new traing dataset.

Q. Explain Regularization
→ 1] Regularization refers to techniques that are used to calibrate machine learning models in order to ~~adjust the~~ minimize the adjusted loss function and prevent overfitting & underfitting.



2] Using Regularization we can fit our Model appropriately on a given test curve.
3] By using regularization we can remove the errors or we can say noisy data.
4] Regularization has 2 Techniques
i) L2 (Ridge Regularization)
ii) L1 (Lasso Regularization)

i) Ridge Regularization (L2) ⇒
- It is also known as ridge regression
- It modifies the over-fitted or under-fitted models by adding penalty equivalent to the sum of the squares of the magnitude of coefficient
- This means that the mathematical function representing our machine learning model is minimized & coefficients are calculated.
- Ridge Regression performs regularization by shrinking the coefficient present.

ii) Lasso Regression :

— It modifies the over-fitted or underfitted models by adding penalty equivalent to the sum of the absolute values of coefficients.

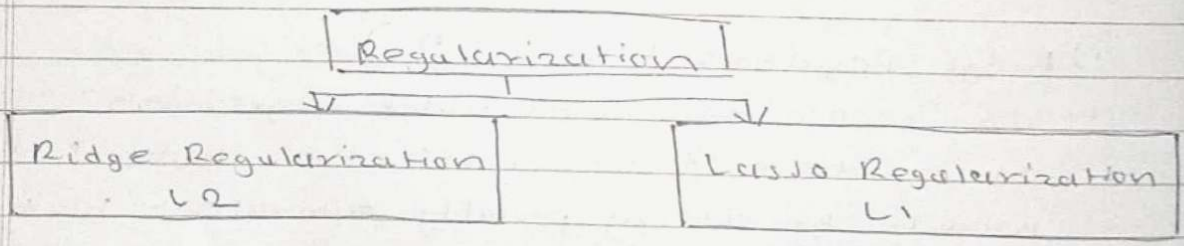
- Lasso Regression also performs coefficient minimization, but instead of squaring the magnitudes of coefficient, it takes true values of coefficient.

- This means that coefficient sum can also be 0, because of the presence of negative coefficients.

- We can control the coefficient values by by controlling the penalty terms, just like the ridge regression.

5] Regularization is used to reduce the complexity of model.

6] Regularization shrinks the magnitude of regression coefficient.



Page No.
Date

Q. Explain Autoencoders with it's properties.

→ 1) Autoencoder is a neural network that tries to reconstruct it's input.

2) Autoencoders are unsupervised learning algorithm that ~~helps to~~ reduces the number of dimensions.

3) Once the data is encoded through the algorithm it is then decoded to other side.

4) Autoencoders are consist of 3 parts.

i) Encoder

ii) Bottlenecks.

iii) Decoder.

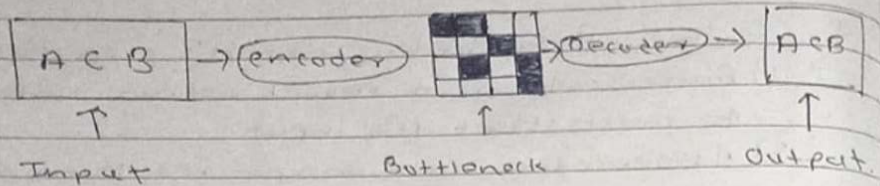
i) Encoder :- It is a module that compresses the input data into a encoded representation.

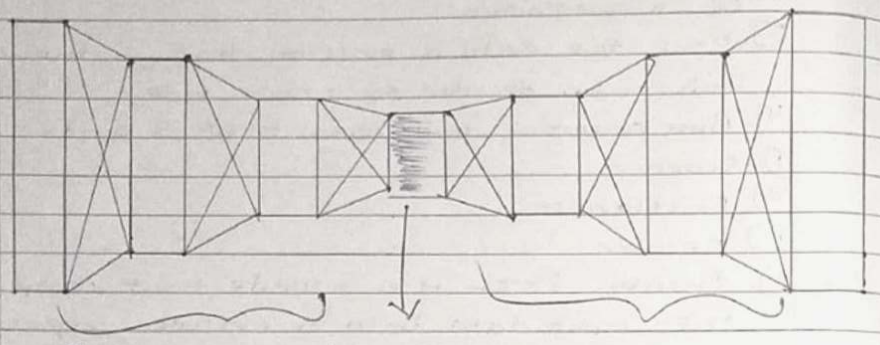
- After the compression the input is sent to the Bottleneck.

ii) Bottleneck :- It is a module that contains the compressed knowledge representation Therefore this is the most important step of autoencoder.

iii) Decoder :- It is a module that helps to decompress the knowledge representation and reconstruct the data back from ils ~~input~~ encoded input.

5) The encoders are trained with the decoders.





6) The encoder compresses the input & decoder attempts to recreate the input from the compressed version
7) The autoencoder compresses the data into lower dimension & then reconstruct the output from this representation.
8) After training the encoder model is saved & decoder is discarded.
9) The loss function is calculated between the actual & predicted input.
10) Applications
i) Data storage
ii) Feature detection
iii) Recommendation system.

11) Properties :
i) Data specific :
- Autoencoders are only able to meaningfully compress data similar to what they have been trained on.
- They are different from a standard compression technique like "gzip".

ii) Lossy :
- Autoencoders are lossy compression technique
- Once the data is encoded then the o/p will not be same as original data
- It will be close to original data but not exact as original one.

iii) Unsupervised :
- Autoencoder is a unsupervised learning algorit
- Auto encoder does not require a labels to trained but to be more precise they are self supervised.
- They are self supervised because they gener ate their own labels from the training data.

9. Explain Undercomplete, Overcomplete and linear autoencoders.

→ 1) Undercomplete Autoencoder:

i) It is a type of a neural network where the dimensionality of encoded representation is smaller than the input data.

ii) It enforces the model to learn a compressed representation of the input

iii) The goal is to force the network to learn more efficient and informative representation of data.

e.g. Think of it like a compact backpack, it only holds the most important stuff, forcing the model to learn essentials.

2) Overcomplete Autoencoder:

i) It is a type of a neural network where the dimensionality of a encoded representation is higher than the input data.

ii) Overcomplete Autoencoders can sometimes leads to the overfitting but they can sometimes be usefull in specific scenarios.

iii) This allows model to potentially learn redundant features.

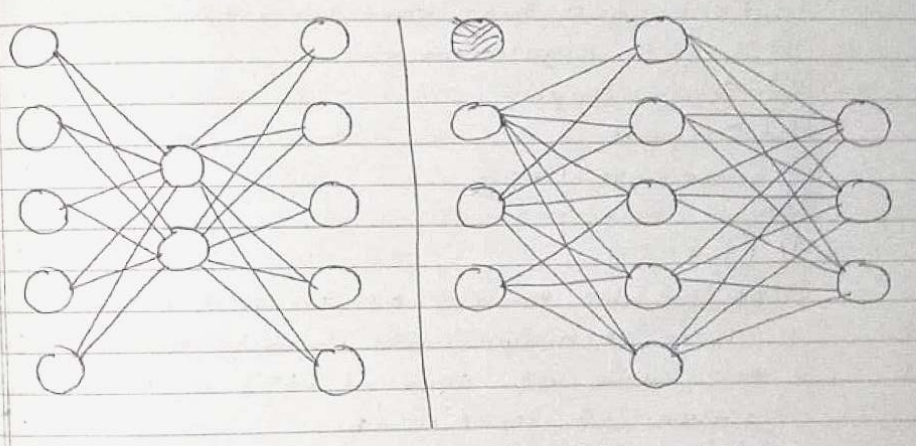
e.g. Imagine having a backpack that is larger than needed, it might end up carrying more than necessary, including potentially unnecessary things.

3) Linear Autoencoder:

i) It is a type of a autoencoder where both the encoder & decoder are composed of only linear transformations.

ii) linear autoencoders can capture linear relationships in the data, they might struggle with more complex & non-linear patterns

e.g. Imagine a simple box that can do only straightforward actions. It's like having a tool that can handle basic tasks but struggles with more complex job.



Undercomplete Autoencoder

Overcomplete Autoencoder.

Q. Regularization in Autoencoder

→ 1) We use regularization in autoencoders to prevent the overfitting and to improve the generalization.

2) Autoencoders aim to learn compressed representations of the input data and reconstruct the original input from those encodings.

3) However, without regularization, they can overfit the training data. This leads to a poor performance on unseen data.

4) There are some common regularization technique that can be used

i) L1 & L2 Regularization
ii) Early stoping
iii) Dropout
iv) Weight Decay

i) L1 & L2
- It is also known as Lasso & Ridge regularization
- These techniques add penalty equivalent to the sum of squared (L2) or & absolute (L1) values of coefficient.
- encouraging sparsity (Fewer active neurons) and prevent overfitting.

ii) Early stoping : Training is stopped before the model starts memorizing the training data, preventing overfitting

iii) Dropout : Neurons are randomly dropped out during the training.

iv) Weight Decay :- This is similar to the L2 regularization, but directly applied to the weights during training.
- This helps in reducing the overfitting.

5) Benifits of Regularization in Autoencoder

i) Improve Generalization : Model will perform better on unseen data.

ii) Enhanced feature Representation : Useful for task like dimensionality reduction or anomaly detection.

iii) Robustness to Noise : The model become less sensitive to noise in the input data.

6) There are 3 Types of Regularized Autoencoder

i) Sparse Autoencoder
ii) Contructive Autoencoder
iv) Variational Autoencoder

Q9 Denoising Autoencoder

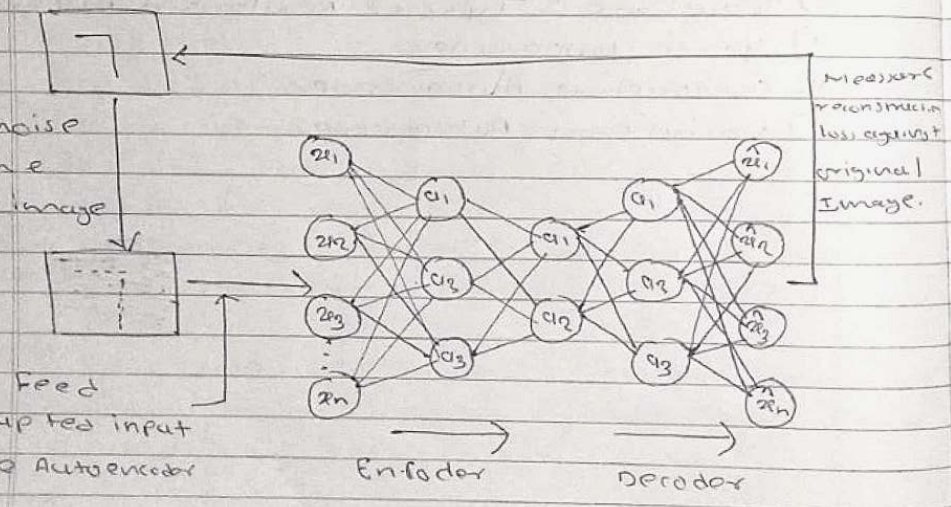
→ 1) A denoising Autoencoder simply corrupts the input data using probabilistic process before feeding it to the network.

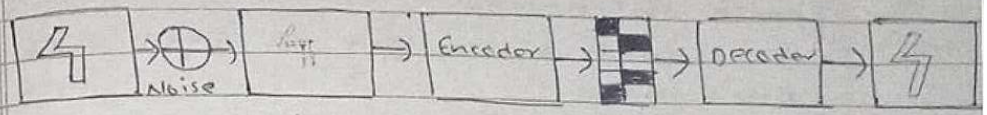
2) Denoising Autoencoder creates a corrupted copy of a input by introducing some noise.

3) Noise was randomly added to the input data and then autoencoder was trained to recover the original.

4) This helps to avoid the autoencoders to copy the input to the output without learning features about the data

5) One of the common applications of such autoencoder is to pre-process the image to improve the accuracy of a optical character recognition (OCR) algorithm.





original Image

Noisy Input

Applications :

1) Image Denoising
2) Fraud detection
3) Data compression
4) Anomaly Detection

Advantages

1) Improved Data Quality
2) Feature Learning
3) Unsupervised Learning

Q. Sparse Autoencoder

→ 1) It is one of the types of Regularized Autoencoder.

2) A sparse Autoencoder tries to ensure that the ~~neuron~~ neuron is inactive most of the time in network.

3) In other words we can say that majority of neurons remain inactive during encoding.

4) And this functionality provides 2 key benefits

i) Feature selection : Discarding ~~to~~ redundant or irrelevant details. This leads to more interpretable.

ii) Robustness to noise : with the fewer active neurons, sparse autoencoder are less sensitive to noise in the input data, achieving better performance in noisy environment.

5) Sparse Autoencoder uses following regularization techniques

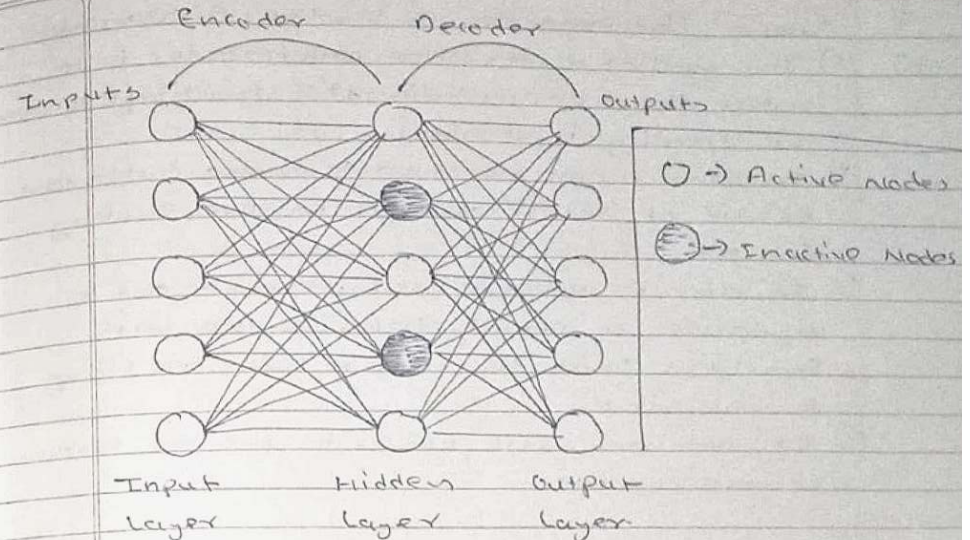
i) L1 Regularization

ii) Dropout

i) L1 Regularization :- Adds penalty equivalent to the sum of absolute values of coefficient/ hidden layer weights

- This encourage them to be close to zero, leading to fewer active neurons.

ii) Dropout : During Training, a random subset of a hidden layer is dropped out.



Advantages

1) Improved Generalization
2) Interpretability
3) Reduced computational cost.

Page No.
Date

Q. Sparse Autoencoder Contractive Autoencoder

→ 1) The goal of contractive autoencoder is to reduce the representation's sensitivity towards the training input data.

2) In other words we strive to make autoencoder robust of small changes in the training dataset.

3) In order to achieve this we must add a regularizer or penalty term to the cost function that the autoencoder is trying to minimize.

4) It adds an extra term in the loss function of autoencoder.

Advantages

1) Improved Generalization
2) Robustness to the noise & outliers.
3) Enhanced Feature Learning.

Applications

1) Image & Signal Processing
2) NLP
3) Bioinformatics.

Page No.
Date

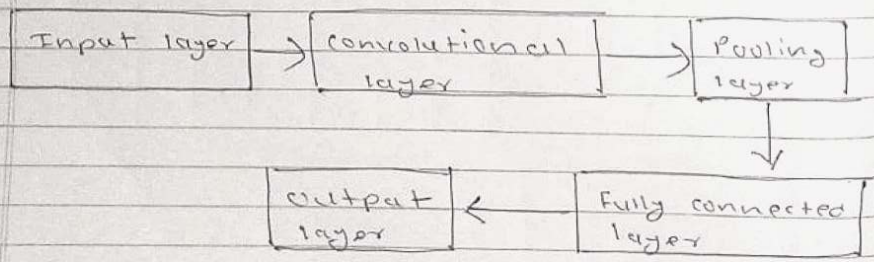
Q. Applications of Autoencoder

→ 1) Data compression & Denoising
2) Feature Extraction
3) Dimensionality Reduction
4) Anomaly Detection
5) Outlier Identification
6) Data Generation & Augmentation
7) Recommendation System
8) Music Generation
9) Robotics.

Date

9. Explain CNN Architecture

→ 1) CNN stands for convolutional Neural Network.
2) CNN is a extended & Advance version of ANN.
3) CNN is used for Image Recognition and Pattern classification.
4) It is made up of a multiple layers such as input layer, convolutional layer, Pooling layer, fully connected layer & output layer.
5) The convolutional layer is a key component in CNN where the filters are applied to the images to extract the features such as edge detection, textures & shapes.
6) CNN is a supervised learning Algorithm
7) CNN are trained with the large dataset of labeled images, where the network learns to recognize the patterns & features.
8) CNN can run direct without any preprocessing.
9) It is a feed forward Neural Network
10) It uses Relu & softmax Activation function.



Date

i) Input layer :- In this layer we feed the inputs to the CNN model.
- This is the first layer of CNN

ii) convolutional layer :
- This is one of the most important layer of CNN.
- In this layer filters are applied to the images to extract the features such as edge detection texture & shapes.
- Feature Extraction is performed in this layer.
- This layer uses Relu activation function.

iii) Pooling layer : It is a common pooling technique that is used to pool the maximum value within a specific region of the input
- This layer is also known as max pooling layer

iv) fully connected layer :
- Towards the end of the CNN this layer used for making predictions or classifications.
- These layer takes feature extracted by the convolutional layer & Pooling layer and combine them to produce output.
- It uses softmax Activation function.

Page No.
Date

9. Padding & stride

→ * Padding

1) After the convolutional operation the original size of the image starts shrinking.
2) To overcome this problem padding was introduced.
3) Padding is a process of adding extra pixels around the input image.
4) Padding prevents the reduction of the feature map size

2	4	9	1	4
2	1	4	4	6
1	1	2	9	2
7	3	5	1	3
2	3	4	8	5

↓

0	0	0	0	0	0	0
0	2	4	9	1	4	0
0	2	1	4	4	6	0
0	1	1	2	9	2	0
0	7	3	5	1	3	0
0	2	3	4	8	5	0
0	0	0	0	0	0	0

Image

X

1	2	3
-4	7	4
-2	-5	1

Filter / kernel

=

21	59	37	-19	2
30	51	66	20	43
-14	31	49	101	-19
59	15	53	-2	21
49	87	64	76	10

Features.

Page No.
Date

formula ∴ $[(n-f)/s]+1$

where,
n = input size
f = filter size
s = stride

∴ $[(7-3)/1]+1 = 5$

* stride.

1) stride is a parameter in a CNN that determines how convolutional filter moves across the input
2) when the number of stride is 1, it moves the filter to 1 pixel at a time
3) similarly when the stride is 2, it moves the filter to 2 pixel at a time & so on.

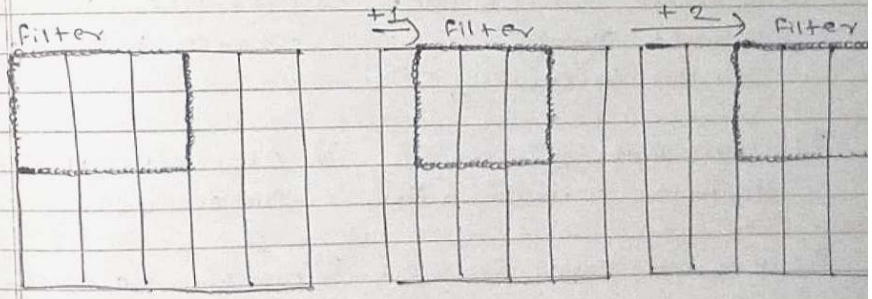


Image Image Image

$\left[\left(\frac{n+2p-f+1}{s}\right)+1\right]$ $\left[\left(\frac{n+2p-f+1}{s}\right)\right]$

Q. Fully connected NN vs convolutional NN.

→

	FCNN	CNN
1]	Fully connected layers	1] Convolutional Layers
2]	Flattened input Data	2] Grid like input data.
3]	All neurons are connected to each other	3] Local connectivity via convolutional filters.
4]	There is no parameter sharing	4] There is parameter sharing
5]	Can Perform General tasks	5] Can Perform Image Related tasks.
6]	Limited ability to learn ~~spatial, stability~~ spatial hierarchy	6] Effectively learns spatial hierarchies
7]	Uses ReLu, Tanh & softmax activation func	7] Uses ReLu Activation function
8]	Requires fixed size input	8] Can handle variable size input.
9]	Slower Training speed	9] Faster Training speed
10]	Simple Architecture	10] Complex Architecture

Q. Explain variants of basic convolutional function

→

1] Dilation convolution
2] Transposed convolution
3] Depthwise convolution
4] Grouped convolution
5] Atrous convolution

1] Dilation convolution :
i] Dilation convolution helps expand the area of the input image covered without pooling
ii] The objective is to cover more information from the output obtained with every convolution operation.
iii] Dilation convolution introduces gaps between the filter elements in a kernel.
iv] It is useful for tasks like edge detection & semantic segmentation.

2] Transposed convolution :
i] Transposed convolution is a opposite of standard convolution
ii] It upsamples the input feature map by learning filters that expand the spatial dimensions.
iii] This is useful for tasks like image generation, super resolution & autoencoders.

3] Depthwise convolution

i] Depthwise ~~sp~~ separable convolution factorizes the standard convolution operation into two separate steps

- Depthwise convolution &
- Pointwise convolution

ii] Depthwise convolution applies individual filters to each input channel separately.

iii] Pointwise convolution then combines the outputs of each channel using 1x1 filter.

4] Grouped convolution :

i] Group convolution divides the input channel into smaller groups and applies separate sets of filters to each group.

ii] This allows more diverse features for each group.

5] Atrous convolution :

i] Atrous convolution is a special case of dilation convolution where dilation rate is applied asymmetrically in different directions.

ii] This can be used in a tasks like medical image analysis & object detection.

8. Explain Multichannel convolutional operation.

→ 1] It is a workhorses of CNN's, the core architecture behind many modern AI Application

2] They are the key to extracting complex features from the data with multiple dimensions, like images and videos.

i] Multiple channels :

- Imagine an image as stack of layers, each representing a different color channel (RGB)
- These channels holds the distinct information and can be processed independently.

ii] convolutional kernel : The kernel is a small filter that slides across the input data.

iii] Channel-wise convolutional : In multichannel operations the kernel interacts with each input channel separately.

iv] Channel summation : After the individual convolution, the output from each channel are summed to create a single feature map.

Advantages

1] Feature Richness
2] Efficiency
3] Representational Power

Applications

1] Image Recognition
2] Object Detection
3] Semantic Analysis
4] Video Analysis
5] NLP

Q. Explain LeNet Architecture

→ 1] The LeNet Architecture was developed by Yann LeCun in late 1980's.

2] It is considered as the first convolutional Neural Networks.

3] LeNet was used for postal automation task such as reading zip codes on mail.

4] It was designed to recognize handwritten digits and was one of the first successfull application of CNN for this task.

5] The LeNet architecture consist of convolutional & Pooling layers and followed by fully connected layers

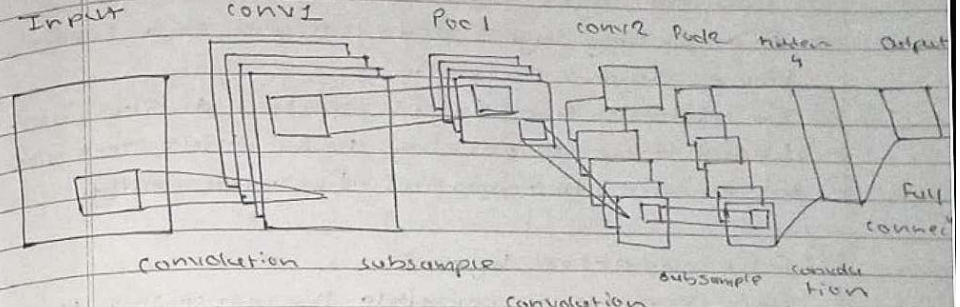
6] The convolutional layers are responsible for learning features from the input image, while the pooling layer reduces the spatial dimensionality and fully connected layer is responsible for classification.

7] It is a foundation of many further developments & improvements in CNN architectures.

8] LeNet follows feed forward approach means output of one layer is used as input in another layer.

9] The LeNet Architecture consist of following layers

i] Input layer
ii] convolutional layer
iii] Pooling layer
iv] fully connected layer
v] Output layer.



i] Input layer :

- In CNN the input layer are responsible for receiving input data & passing it to the next layers in the Network.
- In LeNet input layer is consist of a grid of neurons, with each neuron representing a pixel in the input image.
- This layer is forwarded by one or more convolutional layer, which apply filter to the input data & extract features from it.
- Basically input layer plays a vital role by providing the raw data that the network uses to learn & make predictions.

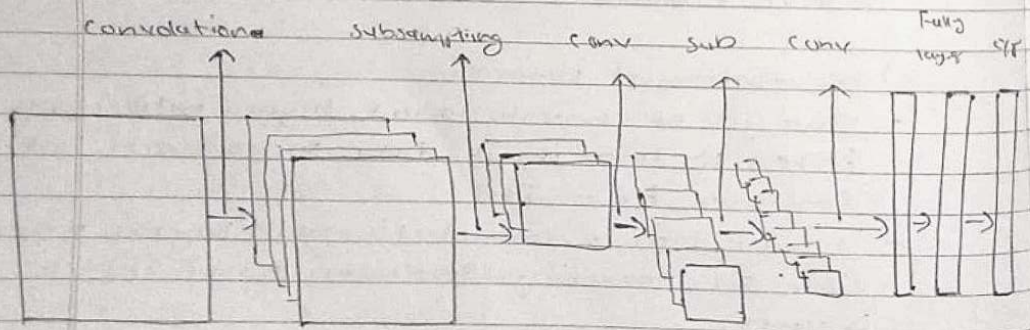
ii] convolutional layers :

- The job of convolutional layers is to apply filters to the input data to extract the features from it.
- LeNet has 2 convolutional layers responsible for learning features from input images.

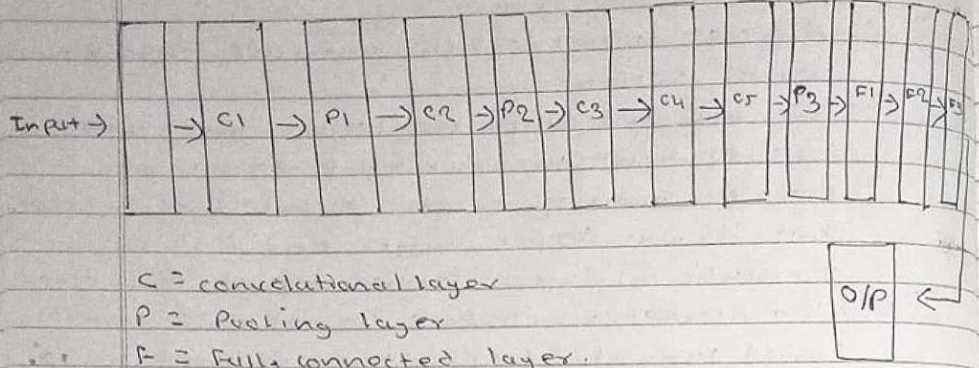
3) Pooling layer :
- The pooling layers downsample the convolutional layers output.
- It reduces the data's spatial dimension.
- LeNet has 2 pooling layer that are used to reduce the spatial dimensionality.

iv) Fully connected layer :
- This layer is responsible for processing the input received from the convolutional & Pooling layer.
- It makes the prediction based on the input received from convolutional & Pooling layer.
- LeNet has 3 fully connected layer that are used for classification.

v) Output layer :
- This layer produces the final prediction based on the output of the fully connected layers.
- It gives the final output of the model.



Q. AlexNet Architecture
→ 1) AlexNet was the first large scale convolutional Neural Network.
2) This was the first architecture that used GPU to boost the training performance.
3) AlexNet had very similar architecture to LeNet but it was deeper, bigger and featured convolutional layers stacked top of each other.
4) The AlexNet architecture consist of 5 convolutional layer & 3 pooling layer followed by 3 fully connected layer.
5) It also include a Normalization layer and Relu activation function on every layer except the output layer.
6) Here the convolution layers are responsible for learning features from the input image while the Pooling layer is responsible for reducing the spatial dimensionality and fully connected layer for classification.
7) Alexnet follows feed forward approach means the output of one layer is used as input in another layer.
8) The AlexNet Architecture consist of following layers.
i) Input layer
ii) convolutional layer (5 layer)
iii) Pooling layer (3 layer)
iv) Fully connected layer (3 layer)
v) Output layer



C = convolutional layer
P = Pooling layer
F = Fully connected layer.

i) Input layer :
- This layer is responsible to feed the input data to the network.
- Input layer plays vital role by providing the raw data to the network which can be processed later.

ii) convolutional layer :
- It takes input data from input layer & applies filter to the input data to extract the features.
- Alexnet has 5 convolutional layer for learning features from input data.

iii) Pooling layer :
- Each layer performs max pooling to reduce the spatial dimensionality of feature maps.
- Alexnet has 3 pooling layers.

iv) Fully connected layer :
- This layer is responsible for processing the input received from the pooling & convolutional layer.
- It makes the prediction based on the input received from the pooling & convolutional layer.
- Alexnet has 3 fully connected layer.

v) output layer :
- This layer produces the final ~~output~~ prediction based on the ~~fo~~ output of fully connected layer.
- It uses softmax activation function to generate the probability of prediction.

vi) Each layer uses ReLu activation function except the output layer.

Page No.
Date

Q. Explain ResNet Architecture.

→ 1) It is a one of the popular architecture of CNN

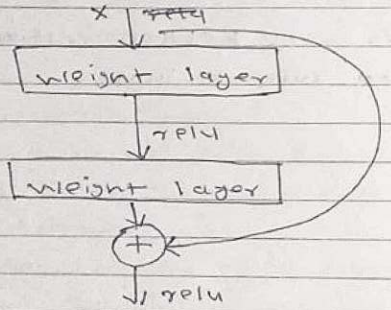
2) ResNet stands for Residual Network.

3) In order to solve the problem of vanishing gradient, this architecture introduced the concept called Residual Blocks.

4) In this architecture we use a technique called skip connections.

5) This technique connects activations of a layer to further layers by skipping some layers between, and this forms a residual block.

6) ResNet are made by stacking these residual block together



7) The advantage of adding this type of skip connections is that if any layer hurt the performance then it will be skipped by regularization.

8) That's how ResNet solve the problem of vanishing Gradient.

Page No.
Date

Advantages

1) Deeper Architecture
2) Improved Accuracy
3) Generalization
4) skip connections.

Q. Explain RNN

→ 1) RNN stands for Recurrent Neural network.

2) RNN was introduced to overcome the problem of Feed Forward Neural network.

3) We know that in feed forward NN the flow was only towards the forward direction and to solve this problem RNN allows us to loop, between previous and current state.

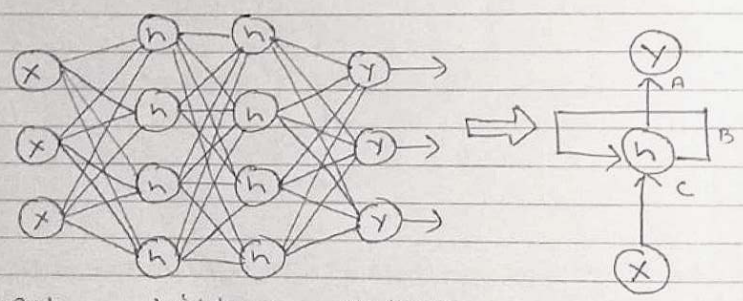
4) It tries to perform same function on every input iteration that's why it is recurrent.

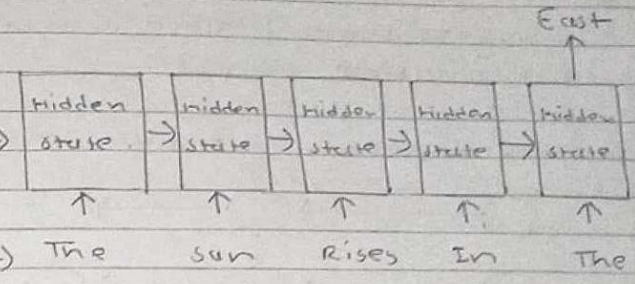
5) RNN use Tanh activation function.

6) RNN has a memory that remembers all the information about what has been calculated.

7) It reduces the complexity of parameters.

8) Below is the example of how you can convert Feed Forward NN to RNN





9) RNN has 4 Types

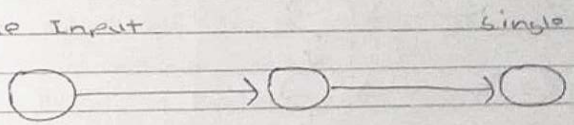
i) One to One

ii) One to Many

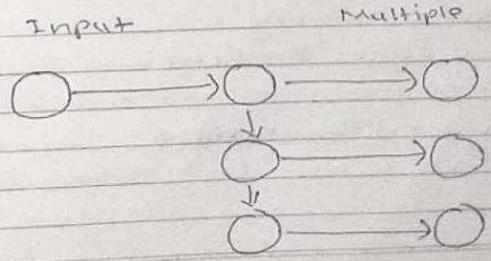
iii) Many to Many

iv) Many to One

i) One to one : There is only one input and output.

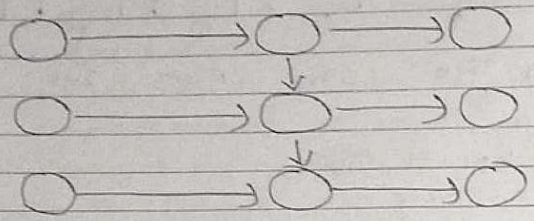


ii) One to Many : In this type there is only one input & multiple output.



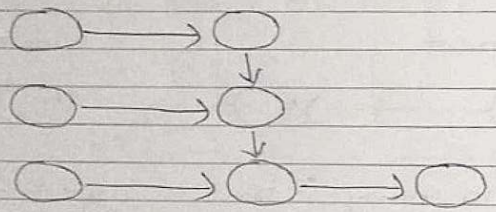
iii) Many to Many : There is multiple input and multiple output.

Multiple Input Multiple Output.



iv) Many to One : There is multiple input but only one output.

Multiple Input Single Output.



Application
1) Image captioning
2) Time series Prediction
3) NLP
4) Machine Translation

Q. Explain LSTM

→ 1) LSTM is a powerful type of RNN

2) LSTM was introduced to resolve the short term memory issue of RNN which is known as vanishing gradient.

3) LSTM is specifically designed to handle the sequencial data such as time series, text and speech.

4) LSTM models are capable of handling the long term sequencial data.

5) And this makes them suitable for tasks such as language translation, speech recognition & time series forecasting.

6) A traditional RNN has single hidden ~~layer~~ state which makes it difficult for network to understand long term dependencies.

7) To address this issue LSTM introduced memory cell, which is a container that holds the information for a period of time.

e.g. consider a line

"Teja lived in India for 13 years. He loves listening to good music. He is fan of Programming. He is fluent in ----."

8) In above example as a human we can easily predict that ans will be Hindi. but how did we predict that?

9) ~~we just reassemble the previous~~

9) we just remembered the previous words ~~and predict~~ that Teja lived in India for 13 years.

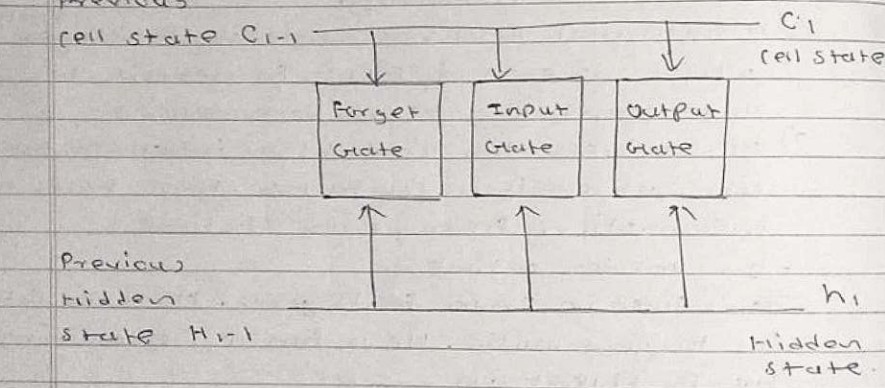
(Sigmoid & Tanh).

10] This led us to conclude that Tejas might be ~~Fth~~ fluent in Hindi.

11] And To train model to remember long term dependencies LSTM introduces Memory cell which holds all the previous data.

12] The memory cell is controlled by the 3 gates
i] Input Gate
ii] Output Gate
iii] Forget Gate

13] These Gates Decides what information to add, remove from the memory cell.



14] In LSTM, ~~LSTM~~ hidden ~~layer~~ state is used for predictions.

15] Hidden ~~layer~~ state is divided into 2 states, cell state & Hidden state.

16] The cell state is called as internal memory while the hidden state is used for computing output, for making predictions.

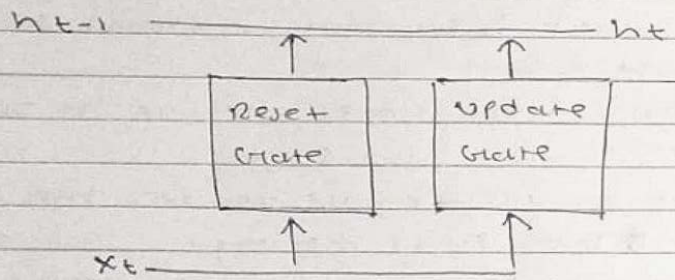
i] Forget Gate
- This Gate is responsible to decide what information to be removed from cell state
- e.g. Tejas is Good singer, He lives in India. TJS is also Good Singer
- As we start talking about TJS The nlw will understand that the subject is changed from Tejas to TJS
- And the information about the Tejas is no longer need so the network will forget the Tejas.

ii] Input Gate :
- The input gate is responsible for ~~adding~~ decide what information should store in cell state.
- e.g. Tejas is Good singer, He lives in India. TJS is also good singer.
- After the forget ~~get~~ Gate removes the information of Tejas from memory.
- Then the gate decides what info should keep in the memory so it stores the TJS.

iii] The Output Gate :
- The output Gate is responsible to decide what information to ~~be~~ taken out from the cell state
- e.g. TJS's debut film was a huge success. congrats...
- Here the Gate will decide TJS to ~~be~~ raise out.

Q. Explain GRU

→ 1] GRU stands for Gated Recurrent Units.
2] GRU is a advance version of LSTM with less parameters but with same functionality.
3] while backpropagating LSTM network, we need to update a lot of parameters in every iteration.
4] This increases our training time, so we introduced the GRU cell, which act as a simplified version of LSTM cell.
5] Unlike the LSTM, GRU cell has only 2 gates and one hidden state.
i] Reset Gate
ii] Update Gate



i] Update Gate :
- This gate is responsible to decide what information from the previous time step can be taken forward to the next time step.
- Basically it is a combination of input gate, and forget gate.
- The update gate is regulated by sigmoid function.

ii] Reset Gate :
- The reset gate is responsible how to add new information
- The Reset gate is regulated function.

Features
1] Simplified Architecture
2] Gating Mechanism
3] Adaptive Learning

Advantages
1] Efficiency
2] Performance
3] Interpretability

Applications
1] NLP
2] Speech Recognition
3] Time series Forecasting
4] Music Generation & Analy