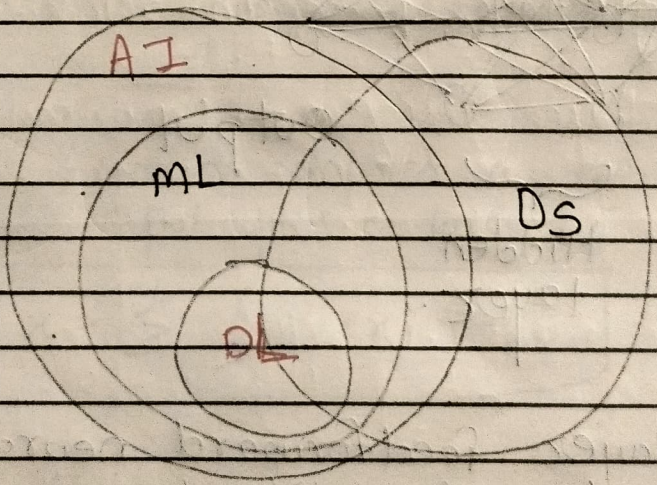
UNIT-I

Fundamentals of Deep Learning...

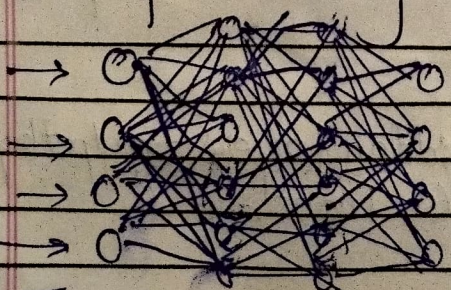
Page No.
Date

Q Explain Deep learning with its architecture.

- The term deep usually refers to the number of hidden layers in the neural network.
- Deep learning is the subset of machine learing.
- In machine learning algorithm predict outputs & discover patterns in dataset whereas deep learning have algorithms based on complex Neural networks that mimic the way a human brain works. to detect pattern in large unstructure dataset
- Deep learning handles both structured & unstructured data and is capable of processing high-dimensional dataset such as image, audio & text.



Deep learning architecture.

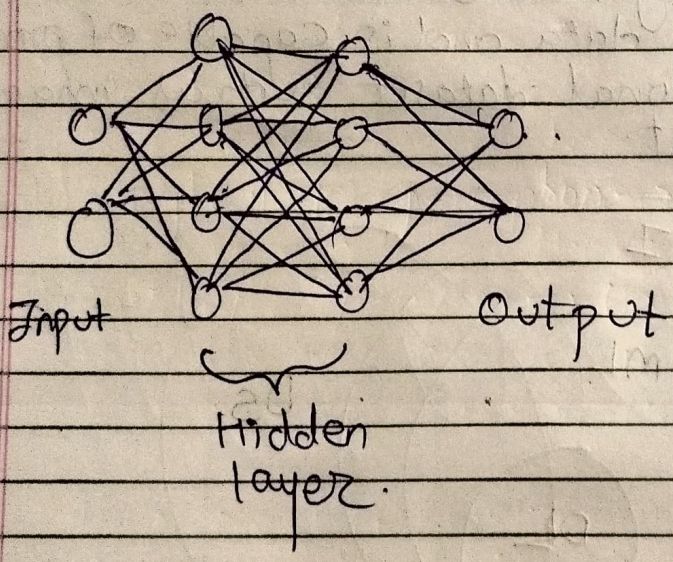


Applications

1) Image classification
2) Speech recognition
3) Fraud detection.
4) Sentiment analysis
5) Healthcare
6) Automation - Self driving Car.

Q] Draw and explain the architecture of multilayered feed forward neural network.

→



Multilayer feedforward neural Network is a type of artifical neural network in which information flows strictly in one direction.

from input layer → hidden layers → output layer without any feedback loops.

1) Input layer →

- In this layer raw data is received.
- Each neuron here represents one input feature.
- passes the value directly to the first hidden layer without applying activation function.

2) Hidden layer →

- It is one or more intermediate layers that transform input data into more abstract representation.
- Each neuron in hidden layer receives input from all neurons in the previous layer (fully connected)
- Each neuron computes a weighted sum of its input plus a bias

$$x = \sum (w_i \cdot x_i) + b$$

3) Output layer

- Produces the final prediction of the network.
- for regression → Linear activation ; for classification → Sigmoid.

forward propagation →

- Data moves layer by layer in forward direction only
- Each layer applies transformation & passes o/p to next layer.

If question asked about FFN

Feedforward Neural Network is an artificial nn in which information flows only in one direction – from the input layer to the output layer.

Two types

1) Single layer perceptron → No hidden layer
2) Multilayer perceptron → One or more hidden layer

Q. Diff between Single layer FFN & Multi layer FFN

Single layer FFN	multi layer FFN.
1) No hidden layer contain only one layer of weight	Two or more layers & have hidden layer
2) Simple Structure	Complex structure
3) Can solve only linear seperable problem	Can solve linear & non-linear problems.
4) Learning Capability is low	Learning Capability is high
5) Faster to train	Slower to train
6) less computation required	more computation required.
7) Ex → Single layer perceptron	Ex → Multilayer perceptron

Q] Explain the concept of gradient based learning.

→ ~~Grad~~ Many of the ML can have optimization problem like loss functions.

- The efficient way to train models is with Stochastic Gradient Decent.
- Gradient based learning helps a computer learn by fixing its mistake step by step.
- It checks the error by using the weights
- Gradient decend helps in telling which way to balance weights to minimise/reduce the errors.
- weights are changed in opp side of GD to reduce weight.
- How much weight can be change each time is decided by learning rate.
- The steps are repeated until the error is minimized.
- This process of fixing the weights step by step is called gradient ~~desent~~ descent.
- The stochastic gradient ~~d~~ has some

limitations. It is not possible to set single learning rate & train the frequent & infrequent feature at the same time.

ex→ from sklearn.linear-model import sgd classifier.

Q] Explain Vanishing Gradient problem? Describe various solutions to this problem.

→

- when backpropagation is used the earlier stages/layers will receive very small updates compared to the later ~~years~~ layers. This problem is called as vanishing gradient problem.

(error signals)

- In vanishing gradients problem, the gradients become very small as they move toward the earlier layers.
- Because of this the weights of earlier stages hearely change.
- This cause network learn very slow sometimes stop learning entirely.

Solutions to the vanishing Gradient problems →

1] Weight initialization → Start weights carefully with special techniques to keep gradients at a healthy scale during training

2] Long short-term memory Network.
- It is the best way to handle this problem
- It have special gates to maintain gradients better over long sequences.

3) Relu activation fn

- Replace activation fn like sigmoid or tanh with Relu because derivative is 1 for +ve inputs, avoiding gradients from shrinking too much.

4) Batch Normalization

- Normalize layer inputs during training to stabilize and maintain gradient flow

5) ResNets:

- Residual Neural Networks uses shortcut connections that allow gradients to flow directly to earlier layers.

Q. Explain the working of an artificial neuron. What is activation function. explain various types

→ An artificial neuron is the fundamental building block of neural network mimicking a biological neuron.

- Its function is to take multiple inputs, process them & produce a single output.
- This output then becomes an input for another neuron in the network.

The working of artificial Neuron:

1) Weighted sum: Each input to the neuron is assigned with weight. The weight determines the importance of input. The neuron multiplies each input value (x_i) by its corresponding weight (w_i) and then adds all these weighted input together.

2) Bias: A bias value is added to this weighted sum. The bias allows the neuron to shift the activation function it gives more flexibility to data

$$Z = (w_1x_1 + w_2x_2 \dots + w_nx_n) + b$$

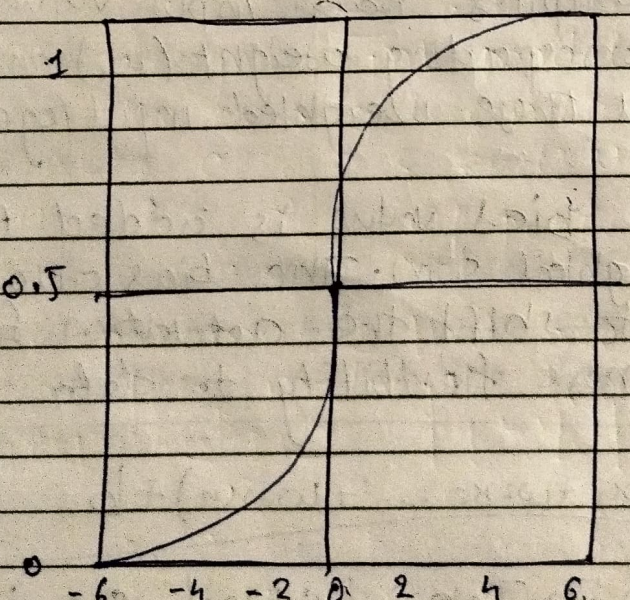
3) Activation → The weighted sum is then passed to activation function. allowing it to learn more complex patterns.

- activation function is mathematical function that determines the output of neuron.
- activation function decides whether or not a neuron will be activated or not & transfer to next layer
- activation function helps in solving complex problem that are beyond simple linear relationships.

1] Sigmoid

It produce a curve with s shape.

$$Sig(t) = \frac{1}{1+e^{-t}}$$



It transforms the input between 0 & 1

2) ReLU
Rectified linear unit.

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

o/p ranges to 0 to ∞

- Relu solves the vanishing gradient problem.
- Relu is most commonly used in CNN or MLP.

3) LRelu & ERELU.

- To address the dying ReLU problem, Leaky Relu introduces a small, non-zero gradient for negative input values.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ ax & \text{if } x \leq 0. \end{cases}$$

where a is usually 0.001. This means neuron will still receive some gradient & can update even when the input is negative.

4) ERelu → Elastic Relu is an activation function designed to combine the benefits of Relu with an elastic property to improve learning flexibility.

- for positive o/p it behaves like ReLu for -ve instead of outputing zero or small fixed slope. it uses learnable parameter.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \text{elastic function of } x & \text{if } x \leq 0 \end{cases}$$

Q] What is hyper parameter? explain its types

→

Hyperparameters are parameters whose value control the learning process & determine the value of model parameters.

- while designing ML model one has many choices & gets confused which design to choose & ends up doing trials for defining perfect ML model.

- The parameters that are used to define this model are known as hyperparameters

- Choosing right hyperparameters is critical for building effective models that train efficiently & perform well.

1] Layer size

- These decides the size & shape of the ~~model~~ input & output layers.
- This refers to the no. of neuron within given layer of neural network.
- Insufficient no. of neurons prevent network from modeling complex data, while too many can lead to long training time & over fitting

2] Magnitude : Learning Rate.

- The amount that the weights are updated during traning is called as learning rate.
- It uses values typically ~~pos~~ in positive range between 0.0 & 1.0.
- Large learning rate() make the model faster but at same time it may cause us to miss the minimum loss fn. & only reach the surrounding of it.
- On other hand, choosing lower learning gives better chance of finding local minima but end up with long time.

Q Explain loss function for Regression
Explain loss function for classification.

→ The loss function measures how well a ML model prediction match the true value.

• It calculates the diff between predicted & actual output.

Loss function for regression.

1) MSE (Mean square error).
measures average of square of error

$$mse = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

2) MAE

measure avg of absolute difference.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Loss f^n for Classification.

1) Cross Entropy loss (log loss).

$$L = \frac{-1}{n} \cdot \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1-y_i) \log(1-\hat{y}_i)]$$

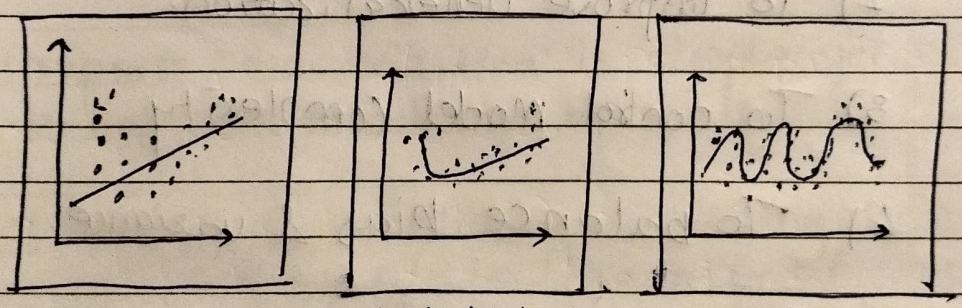
2) Hinge loss.

used mainly in svm.

$$L = \max(0, 1 - y_i \cdot \hat{y}_i).$$

8] What is Regularization? Explain the need for regularization. Explain Droupout Regularization

→



- Regularization is a technique used to reduce overfitting in machine learning model.

- Overfitting occurs when model tries to extract every single feature but cost with irregularities, noise in the output.

- It works by adding a penalty term to the loss function to control the complexity of the model.
- This penalty prevents the model from fitting noise in the training data.
- Regularization helps improve the generalization of the model, so it performs better on unseen data.

Why do we need Regularization?

1) To prevent overfitting
2) To improve Generalization.
3) To control model complexity.
4) To balance bias & variance.
5) To make model more robust.
6) It helps in finding the right balance between underfitting & overfitting.

Page No.
Date

Types of Regularization →

1] L1 Regularization (Lasso) :-

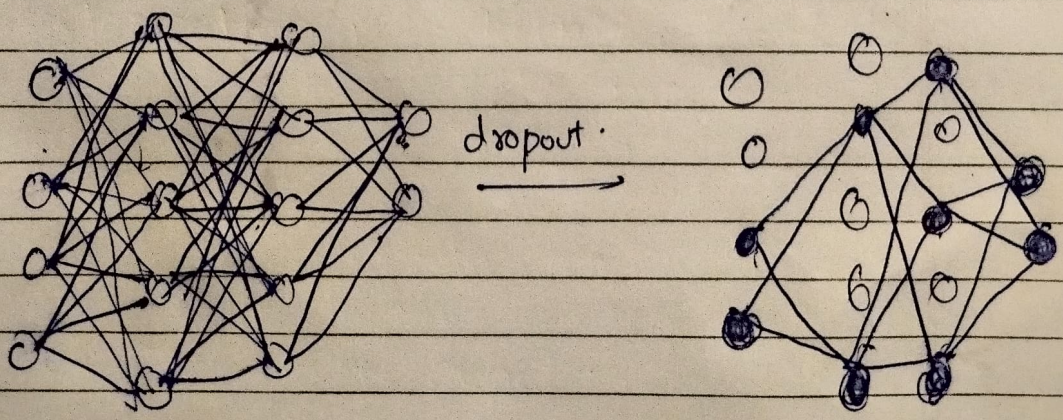
- Laso stands for least absolute & selection operator
- It adds the sum of absolute weights as penalty. Here many weights become zero.

2] L2 Regression. (Ridge).

- It is the technique of regularization to avoid the overfitting in training data set.
- It adds sum of square weight as penalty. Encourages smaller weight but not zero.

Dropout Regularization

- Dropout Regularization is simple yet effective technique introduced by Hinton.
- It helps in reducing overfitting by reducing noise & setting o/p to zero for of each hidden layer with some probability



- This means some neurons are "dropped" ignored in that training pass.

- It forces the model to learn features because it cannot rely on specific neurons

- During ~~dro~~ testing, dropout is turned off and all neurons are active. Output is calculated according to balance the training effect.

- Dropout is inexpensive but powerful method of regularization.