Q1. What is a Multi-layer Perceptron (MLP) & how does it differ from a single-layer perceptron?

→ MLP is a type of artificial neural network that consists of multiple layers of neurons. Flow of MLP is first input layer, then one or more hidden layers depends on the dataset and an output layer.

Difference between MLP & SLP :-

MLP	SLP
(i) Contains hidden layers can solve non-linear problem.	No hidden layer is present. can solve only linear problem.
(ii) ~~Uses threshold (step) func~~	
(ii) Uses non-linear activation function like ReLU, Tanh, Sigmoid.	Uses threshold function.
(iii) Uses backpropogation & gradient descent for training	Uses perceptron learning rule.

Page No.
Date

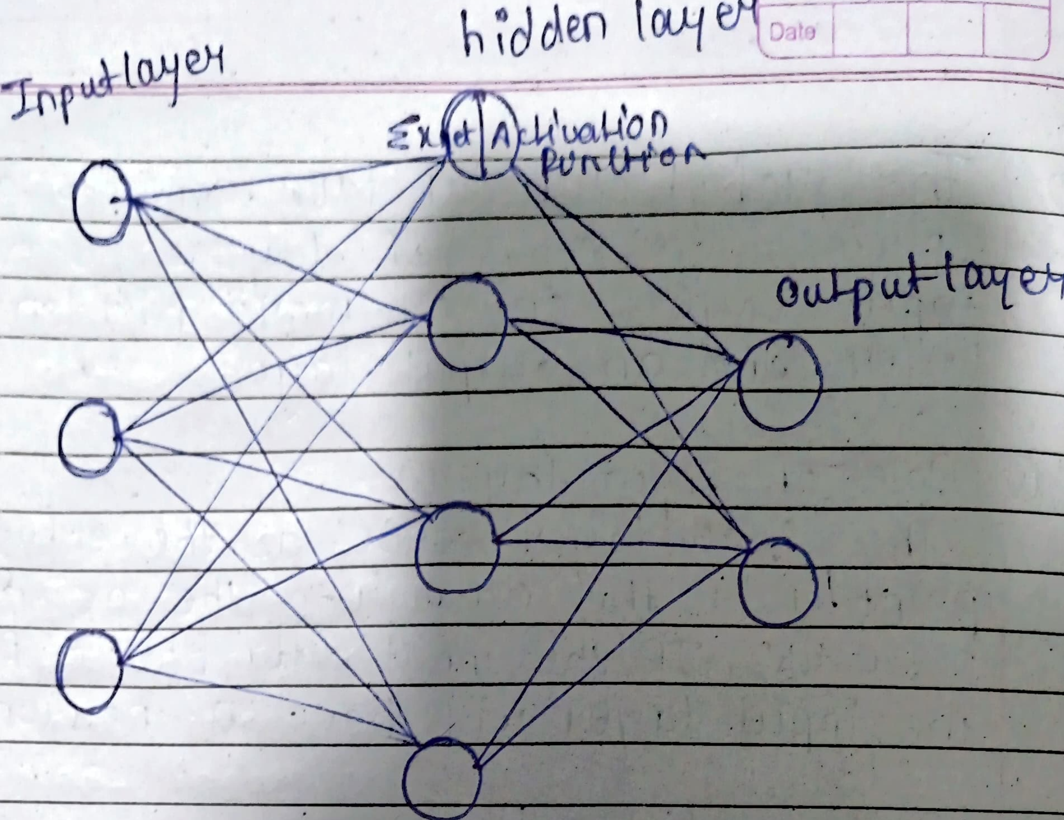
Q2. The architecture of MLP consists of three main types of layers such as an input layer, one or more hidden layer and an output layer.

(i) Role of input layer :-
The input layer serves as the entry point for the data into the neural network. If the input data has n features the input layer will have n neurons.

(ii) Role of hidden layer :-
Hidden layers are responsible for the training of the ~~data~~ model. They are the hyper parameters that can be tuned. In hidden layer weights & bias and activation function are applied during the training.

(iii) Role of output layer :-
The output layer produces the final output of the model. We can also calculate the loss with the help of output layer.

Loss = Actual value - Predicted value.



Q3. There are several ways to initialize weights:-

(1) Normal initialization:-

(a) Normal initialization:- All the weights in this format will be initialize from a normal initialization which lies b/w $(0, \sigma)$.

(b) Uniform ~~initialization~~ distribution:-

All the weights in uniform distribution will lie between $-\frac{1}{\sqrt{fanin}}$ to $\frac{1}{\sqrt{fanin}}$

(2) Xavier/ Glorot initialization:-

(i) Normal initialization:-

$$w_{ij}^{k} \sim N(0, \sigma)$$

$$\sigma = \frac{2}{\text{fan in} + \text{fan out}}$$

(ii) Uniform distribution:-
weights will lie between

$$\left[-\frac{\sqrt{6}}{\sqrt{\text{fan in} + \text{fan out}}} \quad + \frac{\sqrt{6}}{\sqrt{\text{fan in} + \text{fan out}}} \right]$$

(3) He initialization

(i) Normal initialization :-
weights lies between

$$(0, \sigma) \quad \sigma = \sqrt{\frac{2}{\text{fan in}}}$$

(ii) Uniform distribution
weights lies between

$$\sqrt{\frac{-6}{\text{fan in}}} \text{ to } \sqrt{\frac{6}{\text{fan out}}}$$

Page No.
Date

Weight initialization is important because if we initialize weight correctly then optimization of loss function will be achieved in the least time.

Q4. Purpose of Activation function :-

① Introduce non-linearity which allows the network to model complex relationship b/w input & output.

② Non-linear activation functions enable the stacking of multiple layer, each learning different features.

Commonly used Activation function

(i) Sigmoid

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

(ii) Tanh

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

(iii) ReLU

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

(iv) Leaky ReLU

$$f(x) = max(0.012, z)$$

Page No.
Date

(v) Parametrized ReLU

$$f(x) = \max(\alpha z, z)$$

Q5 Backpropogation is the process of adjusting the weights in a neural network to minize the loss function. It is the key part of training phase.

Steps in Backpropogation:-
(i) ~~For~~ Initialization of weights & bias randomly
(ii) Forward pass
(iii) Compute Loss
(iv) Calculate the gradient descent of loss w.r.t output layer weights.
(v) Propogate these gradient backwards through the network.
(vi) Update weights.

Q6. Choosing the numbers of hidden layers & neurons usually depends upon the problem statement or dataset.
If the problem is simple then usually 1 to 3 hidden layers are sufficient.