DL VIVA

Deep Learning (University of Mumbai)

Biological neurons are the fundamental units of the nervous system and are responsible for transmitting information throughout the body.A biological neuron consists of three main parts: the dendrites (receive signals), the cell body (processes signals), and the axon (transmits signals).

The **McCulloch-Pitts** neural model- the earliest ANN model, has only two types of inputs — Excitatory and Inhibitory. The excitatory inputs have weights of positive magnitude and the inhibitory weights have weights of negative magnitude. The inputs of the McCulloch-Pitts neuron could be either 0 or 1. It has a threshold function as an activation function. So, the output signal yout is 1 if the input ysum is greater than or equal to a given threshold value, else 0.

A **perceptron** is a type of ANN that does certain computations to detect features.

Perceptron learning is a supervised learning algorithm used to train perceptrons. to adjust the weights of the perceptron based on training examples until it correctly classifies all the training data.updates the weights based on the error between the predicted output and the actual output for each training example.

Delta learning is a generalization of the perceptron learning algorithm that can be applied to networks with multiple layers.It is based on the idea of gradient descent, where the weights of the network are adjusted in the direction that minimizes a cost function, typically based on the difference between the predicted outputs and the true outputs.

A **multilayer perceptron** is a type of artificial neural network with multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer.

Multilayer Perceptron: Linearly separable, linearly non-separable classes

- classes are linearly separable if there exists a hyperplane that can separate the instances of one class from the instances of the other class in the feature space.
- Linearly non-separable classes cannot be separated by a single hyperplane in the feature space.

main differences between AI, Machine Learning, and Deep Learning?
Artificial Intelligence- enables machines to mimic human intelligence.
Machine Learning is a subset of AI which uses statistical methods to enable machines to improve with experiences.
Deep learning is a subset of machine learning that focuses on building and training neural networks with multiple layers

Three Classes of Deep Learning:

1. Supervised learning relies on labelled input and output training data
2. Unsupervised learning involves training a model on unlabeled data, where the algorithm must discover the underlying structure or patterns within the data.
3. Reinforcement learning involves an agent learning to interact with an environment to achieve a specific goal.

applications of deep learning:

1. pattern recognition
2. Image recognition
3. Machine translation
4. Sentiment analysis
5. Automatic Handwriting Generation
6. Automatic Text Generation.

Deep learning frameworks or tools:Tensorflow, Keras, Pytorch, Theano

Advantages of DL: high performance,scalability
Disadvantages of DL:longer time to execute the model,not good for small data sets

supervised learning algorithms in Deep learning?

1. Artificial neural network
2. Convolution neural network
3. Recurrent neural network

unsupervised learning algorithms in Deep learning?

1. Boltzmann Machine
2. Auto Encoders

Neural Networks are artificial systems that resemble biological neural networks in the human body.

Training, Optimization, and Regularization of Deep Neural Networks:

- Training deep neural networks involves presenting input data to the network, computing predictions, comparing them with actual output, and updating the network's parameters to minimize the loss function.
- Optimization algorithms like stochastic gradient descent (SGD) allow neural networks to be trained faster while achieving better performance.
- regularization techniques such as L1/L2 regularization are used to prevent overfitting.

types of deep neural networks:-

1. FeedForward Neural Network:- flow control starts at the input layer and moves to the output layer. These networks only have a single layer.
2. Radial Basis Function Neural Network:- This type of neural network usually has more than one layer, preferably two
3. A Multi-Layered Feedforward Neural Network (MLP) -consisting of multiple layers of neurons, with connections only going forward, from the input layer through one or more hidden layers to the output layer.

Learning factors- hyperparameters related to the training process

Activation functions introduce nonlinearity to the output of a neuron, allowing neural networks to learn complex patterns in the data.

1. The binary step function -based on a threshold.

2. Tanh- Outputs values between -1 and 1

3. Logistic(Sigmoid): S-shaped curve, squashes input values between 0 and 1

4. Linear-Identity function, returns the input as it is

5. Softmax-used to calculate the probability distribution of the event over 'n' different events ensuring that the sum of the probabilities equals 1

6. ReLU(Rectified Linear Unit): ReLU(x) = max(0, x), outputs the input if it's positive, zero otherwise

7. Leaky ReLU-addressing the issue of "dying ReLU" neurons by allowing a small, non-zero gradient for negative inputs

Loss Functions(cost functions)-calculate the difference between the predicted outputs of a model and the actual outputs.

1. Squared Error Loss (Mean Squared Error): Measures the average squared difference between the predicted and actual values, commonly used for regression tasks.
2. Cross Entropy Loss: Measures the difference between two probability distributions (predicted and actual)

Tensors-multidimensional arrays, which allows us to represent the data having higher dimensions

Backpropagation is used to train neural networks by computing gradients of the loss function with respect to the network's parameters. These gradients are then used to update the parameters in the direction that minimizes the loss function.

Learning Parameters:control the behavior of optimization algorithms during training

1. Gradient Descent (GD):Basic optimization algorithm that updates parameters in the opposite direction of the gradient of the loss function with respect to the parameters.

2. Batch gradient descent-used to calculate the gradients for the whole dataset and perform just one update at each iteration.

3. Stochastic Gradient Descent (SGD): used to calculate the gradient and update the parameters using only single training example.

4. Mini-Batch Gradient Descent: Instead of a single training example, mini-batch of samples is used.

5. Momentum-Based Gradient Descent:Optimization algorithm that adds momentum to the gradient updates

Overfitting model is trained with too much data capturing noise or irrelevant patterns in the training data

Bias- error that occurs in model due to incorrect assumptions
Variance- error introduced by the model's sensitivity to fluctuations in the training data.

Types of Biases:

1. Underfitting Bias: model is trained with very less data so that it cannot capture the underlying patterns in the data

2. Overfitting Bias: model is trained with too much data capturing noise or irrelevant patterns in the training data

3. Inductive Bias: Assumptions that guide the learning process towards certain types of models or solutions.

Bias variance trade-off:balance between bias and variance.
Increasing model complexity typically reduces bias but increases variance, and vice versa.
Low bias and low variance models are desirable, as they accurately capture the underlying patterns in the data and generalize well to new, unseen data.
High bias models underfit the data, while high variance models overfit the data.

Regularization is a set of techniques used to prevent overfitting and improve the generalization performance of models.

Regularization methods-

1. L1 Regularization (Lasso): Adds a penalty term to the loss function proportional to the L1 norm of the model weights

2. L2 Regularization (Ridge): Adds a penalty term to the loss function proportional to the squared L2 norm of the model weights

3. Parameter Sharing: certain parameters of the model to be shared across different parts of the network, reducing the number of parameters and preventing overfitting.

4. Dropout: Randomly drops a fraction of neurons during training

5. Weight Decay: Adds a regularization term to the loss function proportional to the sum of squared weights

6. Batch Normalization: Normalizes the activations of each layer to have zero mean and unit variance

7. Early Stopping: Stops training when the performance on a validation set stops improving

8. Data Augmentation: Increases the size of the training dataset by applying transformations such as rotation, translation to the input data

9. Adding Noise to Input and Output: Injects noise into the input data or the model's predictions during training, forcing the model to learn more robust features and reducing sensitivity to noise in the data.

Autoencoders are a type of artificial neural network used for unsupervised learning
It consists of an encoder that compresses input data to a lower-dimension and a decoder that reconstructs the original input.

1. Linear Autoencoder: autoencoder that can be trained with only single layer encoder and single layer decoder.

2. Undercomplete Autoencoder: smaller dimension for hidden layer compared to the input layer.

3. Overcomplete Autoencoders:larger dimension for hidden layer compared to the input layer.

4. Regularized autoencoders: use regularization techniques to prevent overfitting and improve generalization.

5. Denoising Autoencoders: take corrupted data like random noise as input and are trained to predict uncorrupted data as output

6. Sparse Autoencoders: Sparsity penalty is applied on the hidden layer to prevent overfitting.

7. Contractive Autoencoders:introduce an penalty that penalizes changes in the hidden layer when small changes in the input occur.

Application of Autoencoders: Image Compression

Convolution Operation:
In CNNs, convolutional layers use the convolution operation to apply filters to input data.
The filter slides over the input data, computing the dot product between the filter and the corresponding input values at each position, producing a feature map.

Padding is adding extra rows and columns of zeros to the input data before applying the convolution operation.

1. In valid padding, no padding is added to the input data
2. Same padding adds the necessary amount of padding to the input data so that the output size remains the same as the input size.
3. Full padding adds padding to the input data such that the output size is larger than the input size.

Stride- the number of positions the filter slides at each step while performing the convolution operation. A stride of 1 means the filter moves one position at a time

Relation between Input, Output, and Filter Size:

$$\text{Output Size} = \frac{\text{Input Size} - \text{Filter Size} + 2 \times \text{Padding}}{\text{Stride}} + 1$$

CNN Architecture:

- It consists of convolutional layers followed by pooling layers and fully connected layers.
- Convolutional layers apply filters to the input data to extract features.
- Pooling layers reduce the spatial dimensions of the feature maps
- Fully connected layers perform classification or regression tasks

Weight sharing - same set of filter weights is used across different spatial locations of the input data.

Fully Connected NN vs. CNN:

- In fully connected neural networks (NNs), every neuron in one layer is connected to every neuron in the next layer.
- CNNs use convolutional layers that apply filters to input data

Variants of Basic Convolution Function:

1. Full convolution - same set of filter weights is used across all spatial locations
2. Unshared convolution - separate sets of filter weights for different locations.
3. Tiled convolution - breaks down the input data into smaller tiles and applies convolution independently to each tile.

Modern Deep Learning Architectures:

1. LeNET: has seven layers: two convolutional layers, two average pooling layers, and three fully connected layers. designed for handwritten digit recognition

2. AlexNet: has eight layers: five convolutional layers and three fully connected layers, along with max-pooling layers and dropout for regularization.

Sequence Learning Problem: the output at any time step depends on previous input/output. Eg part of speech tagging

Unfolding Computational Graphs:repeated application of the same set of parameters over time

Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step.

1. A bidirectional RNN - processes input sequences in both forward and backward directions. It consists of two separate RNNs: one that processes the input sequence in the forward direction and another that processes it in the backward direction.

2. Backpropagation Through Time (BPTT) - trains RNNs by unfolding them over time and applying the backpropagation algorithm.

Vanishing gradients occur when the gradients of the loss function with respect to network parameters become very small as they are propagated back through time.
Exploding gradients occur when the gradients become very large

Truncated BPTT: forward and backward passes are run through chunks of sequences instead of the whole sequence.

Long Short-Term Memory (LSTM) is a type of RNN designed to overcome the vanishing gradient problem

- The cell state is a linear pathway that runs through the entire sequence of the data.
- The hidden state, or output, contains information that the LSTM decides to pass on to the next time step
- LSTM utilizes three types of gates:

1. Forget Gate: Determines which information from the cell state to forget or discard.
2. Input Gate (Selective Write): Determines which new information to add to the cell state.
3. Output Gate (Selective Read): Determines which information from the cell state to pass on to the output.

Gated Recurrent Unit (GRU):type of RNN similar to LSTM but with fewer parameters.
It combines the forget and input gates into a single update gate and merges the cell state and hidden state.

Generative Adversarial Networks (GANs) are a class of neural network which consist of two neural networks, the generator and the discriminator.

- The generator takes random noise as input and generates synthetic data samples.
- The discriminator distinguishes between real data samples and fake data samples generated by the generator.

Types of GAN:

1. Vanilla GAN :Consists of a generator and a discriminator trained simultaneously in a minimax game framework.
2. Conditional GAN:Extends the vanilla GAN by conditioning the generator and discriminator on additional information, such as class labels or attributes.
3. Deep Convolutional GAN :Utilizes convolutional neural networks (CNNs) in both the generator and discriminator networks.
4. Progressive GAN : A variant of GAN that gradually increases the resolution of generated images during training.

Applications of GAN: Image Generation, DeepFake, Image compression,brain tumor detection,fraud detection