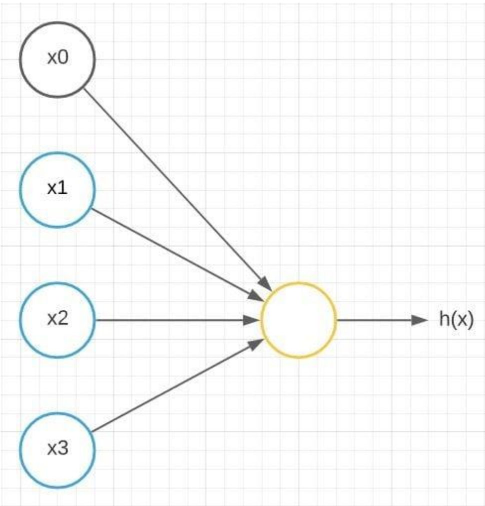
19EE701 – AI TECHNIQUES IN ELECTRICAL ENGINEERING

UNIT 2 – NEURAL NETWORKs

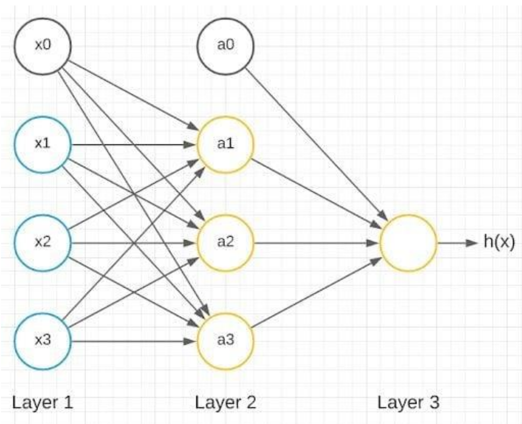
Topic : Neural Network Models

The NN that is implemented in the computer is called an artificial neural network (ANN), as they simulate the neurons present in the brain. The neurons are responsible for all of the actions voluntarily and involuntarily happening in our bodies. They transmit signals to and from the brain.



Model of a neuron

The above model represents a single neuron. A neural network is a group of these neurons strung together.



A network of neurons connected together creating a neural network

We have inputs x1, x2 and x3 as inputs and x0 as a bias unit. We also have three neurons in the next layer: a1, a2 and a3 with a0 as the bias unit. The last layer has only one neuron, which gives the output. Layer one is called the input layer, layer three is called the output layer and layer two is called the hidden layer. Any layer that isn't the input layer or the output layer is called the hidden layer.

How to Train and Model a Neural Network

The following are the steps involved in modeling and training a neural network.

1. Model the input layer according to the no. of input features.
2. Model the output layer according to the no. of classes in the output.
3. Model the number of hidden layers and the no. of neurons in the hidden layers optimally.
4. Randomly initialize weights.
5. Implement forward propagation to get h(x) for any x.
6. Implement code to compute cost function J.
7. Implement backpropagation to compute the gradient of cost with respect to weights.
8. Update the values of weights.
9. Perform steps five-through-nine recursively to minimize the cost J by modifying the weights after each epoch.

Different Types of Neural Networks Models

The nine types of neural network architectures are:

- Perceptron
- Feed Forward Neural Network
- Multilayer Perceptron
- Convolutional Neural Network
- Radial Basis Functional Neural Network
- Recurrent Neural Network
- LSTM – Long Short-Term Memory
- Sequence to Sequence Models
- Modular Neural Network

Definition:

- Neural networks are a class of machine learning models inspired by the structure and function of the human brain. They consist of interconnected nodes (neurons) organized in layers to process data and make predictions or decisions.

Components:

- **Neurons:** Basic units of computation, each applying a function to inputs and passing the result to the next layer.
- **Layers:** Neural networks are organized into layers:
    - **Input Layer:** Receives the raw data.
    - **Hidden Layers:** Intermediate layers where computations are performed. There can be multiple hidden layers.
    - **Output Layer:** Produces the final result or prediction.

2. Basic Architecture

Single-Layer Perceptron:

- **Structure:** Consists of a single input layer and a single output layer.
- **Function:** Can perform simple binary classification tasks.
- **Limitations:** Not suitable for more complex tasks like non-linear classification.

Multi-Layer Perceptron (MLP):

- **Structure:** Includes one or more hidden layers between the input and output layers.
- **Function:** Capable of learning non-linear relationships between inputs and outputs.
- **Activation Functions:** Introduce non-linearity to the model. Common activation functions include:
    - **Sigmoid:** σ(x)=11+e−x\sigma(x) = \frac{1}{1 + e^{-x}}σ(x)=1+e−x1
    - **ReLU (Rectified Linear Unit):** ReLU(x)=max(0,x)\text{ReLU}(x) = \max(0, x)ReLU(x)=max(0,x)
    - **Tanh (Hyperbolic Tangent):** tanh(x)=ex−e−xex+e−x\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}tanh(x)=ex+e−xex−e−x

3. Training Neural Networks

Forward Propagation:

- **Process:** Input data is passed through the network, layer by layer, using weights and activation functions to produce an output.

Loss Function:

- **Definition:** Measures the difference between the predicted output and the actual target values.
- **Common Loss Functions:**
    - **Mean Squared Error (MSE):** For regression tasks.
    - **Cross-Entropy Loss:** For classification tasks.

Backpropagation:

- **Process:** Calculates the gradient of the loss function with respect to each weight using the chain rule of calculus.
- **Purpose:** Updates weights to minimize the loss function, using optimization algorithms.

Optimization Algorithms:

- **Gradient Descent:** The most basic optimization algorithm that updates weights by moving in the direction of the negative gradient.
- **Variants:**
    - **Stochastic Gradient Descent (SGD):** Uses random subsets of data (mini-batches) to perform updates.
    - **Adam (Adaptive Moment Estimation):** Combines the benefits of RMSprop and SGD, using moving averages of gradients and squared gradients.

4. Advanced Neural Network Architectures

Convolutional Neural Networks (CNNs):

- **Purpose:** Specialized for image and spatial data processing.
- **Components:**
    - **Convolutional Layers:** Apply filters to input data to detect features.
    - **Pooling Layers:** Reduce the dimensionality of the data (e.g., Max Pooling).
    - **Fully Connected Layers:** Integrate features and perform classification or regression.

Recurrent Neural Networks (RNNs):

- **Purpose:** Designed for sequence data and time series.
- **Components:**
  - **Recurrent Layers:** Maintain a hidden state across time steps to capture temporal dependencies.
  - **Variants:**
    - **Long Short-Term Memory (LSTM):** Addresses the vanishing gradient problem and captures long-range dependencies.
    - **Gated Recurrent Unit (GRU):** A simplified version of LSTM with fewer parameters.

Generative Adversarial Networks (GANs):

- **Purpose:** Generate new data samples that resemble a training dataset.
- **Components:**
  - **Generator:** Creates fake data samples.
  - **Discriminator:** Distinguishes between real and fake data.
- **Training Process:** Involves a two-player game where the generator improves in creating realistic data, and the discriminator improves in identifying fakes.

Transformers:

- **Purpose:** Effective for sequence-to-sequence tasks like translation and text generation.
- **Components:**
  - **Attention Mechanism:** Allows the model to focus on different parts of the input sequence when making predictions.
  - **Self-Attention:** Helps in capturing relationships between different positions in the sequence.

5. Applications of Neural Networks

Image Classification:

- Example: Identifying objects in images (e.g., cats vs. dogs).

Natural Language Processing (NLP):

- Example: Sentiment analysis, language translation, and text generation.

Speech Recognition:

- Example: Converting spoken language into text.

Recommendation Systems:

- Example: Suggesting products or content based on user preferences.

6. Challenges and Considerations

Overfitting:

- **Definition:** When a model performs well on training data but poorly on unseen data.
- **Mitigation:** Use regularization techniques, dropout, and cross-validation.

Computational Resources:

- **Considerations:** Training deep neural networks can be resource-intensive, requiring powerful hardware (e.g., GPUs).

Interpretability:

- **Definition:** Understanding how neural networks make decisions can be challenging.
- **Techniques:** Use methods like saliency maps or feature importance analysis to gain insights.