19EE701 – AI TECHNIQUES IN ELECTRICAL ENGINEERING

UNIT 2 – NEURAL NETWORKs

Topic : Neural Network Model Architecture

1. Introduction to Neural Networks

- **Definition:** Neural networks are computational models inspired by the human brain. They consist of interconnected layers of nodes (neurons) that process information in stages.
- **Basic Terminology:**
  - **Neuron:** Basic unit of a neural network.
  - **Layer:** A collection of neurons.
  - **Activation Function:** A function that determines the output of a neuron.

2. Feedforward Neural Networks (FNN)

- **Structure:**
  - **Input Layer:** Takes in input features.
  - **Hidden Layers:** Intermediate layers where computation occurs.
  - **Output Layer:** Produces the final output.
- **Activation Functions:**
  - **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:** tanh(x)=ex−e−xex+e−x\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}tanh(x)=ex+e−xex−e−x

3. Convolutional Neural Networks (CNN)

- **Purpose:** Specialized for processing grid-like data (e.g., images).
- **Layers:**
  - **Convolutional Layers:** Apply convolutional filters to the input.
  - **Pooling Layers:** Reduce the spatial dimensions (e.g., Max Pooling).
  - **Fully Connected Layers:** Classic dense layers used for final classification.
- **Activation Function:** Often ReLU.

4. Recurrent Neural Networks (RNN)

- **Purpose:** Designed for sequential data (e.g., time series, text).

- **Components:**
    - **Recurrent Layer:** Allows information to persist by using the output of one time step as input to the next.
    - **Vanishing Gradient Problem:** A challenge where gradients can become very small, hampering learning.
- **Variants:**
    - **Long Short-Term Memory (LSTM):** Addresses the vanishing gradient problem using gates (input, output, and forget gates).
    - **Gated Recurrent Unit (GRU):** A simplified version of LSTMs with fewer gates.

5. Generative Adversarial Networks (GANs)

- **Purpose:** Used for generating synthetic data that resembles real data.
- **Components:**
    - **Generator:** Creates new data samples.
    - **Discriminator:** Evaluates the authenticity of generated samples.
- **Training Process:** The generator and discriminator are trained in opposition, improving each other iteratively.

6. Autoencoders

- **Purpose:** Learn efficient representations (encodings) of data, typically for dimensionality reduction.
- **Components:**
    - **Encoder:** Maps input to a lower-dimensional space.
    - **Decoder:** Reconstructs the input from the encoded representation.
- **Variational Autoencoders (VAEs):** A probabilistic version that models the data distribution.

7. Transformers

- **Purpose:** Primarily used for NLP tasks and are effective for processing sequences of data.
- **Components:**
    - **Attention Mechanism:** Allows the model to focus on different parts of the input sequence.
    - **Self-Attention:** Computes the representation of a word based on the other words in the sequence.
- **Popular Architectures:**

    - **BERT (Bidirectional Encoder Representations from Transformers)**
    - **GPT (Generative Pre-trained Transformer)**

8. Neural Network Training

- **Backpropagation:** The algorithm used to compute the gradient of the loss function with respect to the weights.
- **Optimization Algorithms:**
    - **Stochastic Gradient Descent (SGD)**
    - **Adam (Adaptive Moment Estimation)**
- **Regularization Techniques:**
    - **Dropout:** Randomly dropping neurons during training to prevent overfitting.
    - **L2 Regularization:** Penalizing large weights to reduce overfitting.

9. Advanced Topics

- **Meta-Learning:** Learning how to learn, focusing on improving the learning process itself.
- **Neural Architecture Search (NAS):** Automated design of neural network architectures.

10. Applications

- **Computer Vision:** Image classification, object detection.
- **Natural Language Processing:** Machine translation, sentiment analysis.
- **Reinforcement Learning:** Training agents to make sequences of decisions.