ANN Architectures

For 7th sem

Fundamental Concepts of Network Architectures

- **What is an Architecture?**
  - Refers to the arrangement of neurons, their connections, and the flow of information.
  - Determines how the network processes data and learns.
- **Key Elements of Architecture:**
  - **Layers:** Input, Hidden, Output.
  - **Connectivity:** How neurons in one layer connect to neurons in the next.
  - **Direction of Flow:** Feedforward vs. Recurrent.
  - **Number of Layers/Neurons:** Depth and Width of the network.
  - **Activation Functions:** Applied within neurons (covered last time, but important architectural choice).

Why Different Architectures?

- No single architecture fits all problems.
- Different data types (images, text, tables) and problem types (classification, regression, generation) require specialized structures for optimal performance.
- Optimization for specific tasks and computational efficiency.

Architecture 1: The Perceptron

- **The Simplest Neural Network:** The foundational model.
- **Structure:**
  - **Input Layer:** Receives features.
  - **Output Layer:** A single neuron with a step (or threshold) activation function.
  - **No Hidden Layers.**
- **Operation:**
  1. Calculates a weighted sum of inputs + bias.
  2. Applies a step function: Outputs 1 if the sum exceeds a threshold, 0 otherwise.
- **Capabilities:** Can only solve **linearly separable** problems (e.g., AND, OR gates).
- **Limitations:** Cannot solve non-linearly separable problems (e.g., XOR gate), which led to the "AI Winter" of the 1970s.

Architecture 2: Multi-Layer Perceptron (MLP)

- **Overcoming Perceptron Limitations:** MLPs introduced hidden layers.
- **Structure:**
  - **Input Layer:** Receives raw data.
  - **One or More Hidden Layers:** Where complex computations and feature transformations occur. Each hidden neuron connects to all neurons in the previous and next layers (fully connected/dense layers).
  - **Output Layer:** Produces the final prediction.
- **Key Characteristics:**
  - **Feedforward:** Information flows in one direction, from input to output, with no loops.
  - **Fully Connected (Dense) Layers:** Every neuron in one layer is connected to every neuron in the subsequent layer.
  - **Non-linear Activation Functions:** Crucial in hidden layers to enable learning of complex patterns
- **Capabilities:**
  - **Universal Function Approximators:** With enough hidden neurons and layers, MLPs can approximate any continuous function.
  - Can solve non-linearly separable problems (like XOR).
- **Training:** Uses Backpropagation to adjust weights and biases.
- **Usage:** Tabular data classification/regression, simple image classification (though CNNs excel here). Forms the basis for many other architectures.
- **Image:** A clear diagram of an MLP with input, hidden, and output layers, showing dense connections.

Architecture 3: Convolutional Neural Networks (CNNs)

- **Specialized for Grid-like Data**: Primarily used for images, but also audio, video.
- **The Problem with MLPs for Images:**
  - Too many parameters (weights) for large images (e.g., 1000x1000 image has 1 million pixels, connecting to even one neuron in the next layer would be massive).
  - Lose spatial information (pixels are treated independently).
- **Key CNN Layers/Concepts:**
  - **Convolutional Layer:**
    - Uses small "filters" (kernels) that slide over the input image.
    - Each filter detects specific features (e.g., edges, textures, corners).
    - **Parameter Sharing:** The same filter is applied across the entire image, drastically reducing parameters.
    - **Sparse Connectivity:** Each output neuron only connects to a small region of the input.
    - **Local Receptive Fields:** Neurons only "see" a small part of the input.
    - **Output:** Feature maps (activated features detected by filters).

Architecture 3: Convolutional Neural Networks (CNNs)

- **Pooling Layer (Downsampling):**
  - Reduces the spatial dimensions of the feature maps.
  - **Max Pooling:** Takes the maximum value in a window.
  - **Average Pooling:** Takes the average value in a window.
  - Reduces computational cost and helps with translation invariance.
- **Fully Connected Layer (Dense Layer):** After several Conv and Pooling layers, the data is flattened and
  fed into standard MLP layers for classification/regression.
- **Usage:** Image classification, object detection, facial recognition, medical image analysis, video analysis.

Architecture 4: Recurrent Neural Networks (RNNs)

- **Specialized for Sequential Data:** Text, speech, time series data, video sequences.
- **The Problem with MLPs/CNNs for Sequences:**
  - They treat inputs as independent.
  - Cannot remember past information in a sequence.
  - Order matters! "Dog bites man" vs. "Man bites dog."
- **Key Concept: Memory (Hidden State):**
  - RNNs have "loops" or "recurrent connections" that allow information to persist across time steps.
  - A neuron's output at time t depends not only on its input at time t but also on its hidden state from time t−1.
- **Structure:** A simple RNN cell that takes an input, combines it with the previous hidden state, and produces an output and a new hidden state.

Architecture 4: Recurrent Neural Networks (RNNs)

- **Types/Variants (Brief Mention):**
  - **Simple RNNs:** Suffer from vanishing/exploding gradients over long sequences.
  - **Long Short-Term Memory (LSTM) Networks:**
    - Designed to combat vanishing gradients.
    - Use "gates" (input, forget, output) to control the flow of information, allowing them to remember or forget information over long periods.
  - **Gated Recurrent Units (GRUs):** Simpler version of LSTMs with fewer gates.
- **Usage:** Natural Language Processing (NLP - machine translation, text generation, sentiment analysis), speech recognition, time series forecasting.

Choosing the Right Architecture

- **Data Type is Key:**
  - **Tabular Data:** MLPs are a good starting point.
  - **Images:** CNNs are dominant.
  - **Sequences (Text, Time Series, Audio):** RNNs (especially LSTMs/GRUs) or Transformers.
- **Problem Type:** Classification, Regression, Generation, Clustering.
- **Complexity of Data:** More complex patterns often require deeper networks or specialized layers.
- **Resources:** Computational power, amount of data available.

What is Machine Learning?

- Subset of Artificial Intelligence
- Systems learn from data to make decisions or predictions
- Three main categories:
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning

Overview of Learning Types

Type	Data	Goal	Learning Signal
Supervised	Labeled	Predict Output	Error/Accuracy
Unsupervised	Unlabeled	Find Patterns	Similarity/Distance
Reinforcement	Reward Feedback	Maximize Rewards	Rewards/Penalties

Supervised Learning

- Input → Output mapping using labeled data
- Learn a function f(x) ≈ y
- Applications:
  - Email spam detection
  - Disease diagnosis
  - Stock price prediction

Supervised Learning: Algorithms

- Linear Regression
- Decision Trees
- Support Vector Machines
- Neural Networks

Unsupervised Learning

- Data has no labels
- Goal: Discover hidden patterns or structure
- Applications:
  - Customer segmentation
  - Market basket analysis
  - Topic modeling

Unsupervised Learning: Algorithms

- K-Means Clustering
- Hierarchical Clustering
- DBSCAN
- Principal Component Analysis (PCA)

Reinforcement Learning

- Learn by interacting with the environment
- Agent learns by trial-and-error
- Key elements:
  - State, Action, Reward, Policy, Value

Reinforcement Learning: Examples

- Game playing (Chess, Go)
- Self-driving cars
- Robotics and control systems

Comparative Summary

Feature	Supervised	Unsupervised	Reinforcement
Data	Labeled	Unlabeled	Reward-based
Goal	Predict	Discover	Optimize actions
Signal	Error	Structure	Reward
Use Case	Classification	Clustering	Decision-making

Real-World Applications

- Supervised: Face recognition, spam filters
- Unsupervised: Customer insights, anomaly detection
- Reinforcement: AlphaGo, autonomous vehicles

Summary

Supervised: Learn from labeled examples

Unsupervised: Uncover hidden structures

Reinforcement: Learn from actions and consequences