Introduction to Artificial Neural Networks

• What is an Artificial Neural Network?
  - It is a computational system inspired by the
    Structure
    Processing Method
    Learning Ability
  of a biological brain

- Characteristics of Artificial Neural Networks
  A large number of very simple processing neuron-like processing elements
  A large number of weighted connections between the elements
  Distributed representation of knowledge over the connections
  Knowledge is acquired by network through a learning process
• **Why Artificial Neural Networks?**

  - Massive Parallelism
  - Distributed representation
  - Learning ability
  - Generalization ability
  - Fault tolerance

• **Elements of Artificial Neural Networks**

  - Processing Units
  - Topology
  - Learning Algorithm

• **Processing Units**
Node input: \( net_i = \sum_j w_{ij} I_j \)

Node Output: \( O_i = f(net_i) \)

• Activation Function

- An example
• **Topology**

![Topology diagram](image)

• **Learning**

- Learn the connection weights from a set of training examples

- Different network architectures required different learning algorithms

**Supervised Learning**

The network is provided with a correct answer (output) for every input pattern

Weights are determined to allow the network to produce answers as close as possible to the known correct answers

The *back-propagation* algorithm belongs into this category
Unsupervised Learning

Does not require a correct answer associated with each input pattern in the training set

Explores the underlying structure in the data, or correlations between patterns in the data, and organizes patterns into categories from these correlations

The *Kohonen algorithm* belongs into this category

Hybrid Learning

Combines supervised and unsupervised learning

Part of the weights are determined through supervised learning and the others are obtained through unsupervised learning

- **Computational Properties**

A single hidden layer feed-forward network with arbitrary sigmoid hidden layer activation functions can approximate arbitrarily well an arbitrary mapping from one finite dimensional space to another
• **Practical Issues**

  - Generalization vs Memorization

  ![Diagram of Good fit vs Bad fit](image)

  How to choose the network size (free parameters)

  How many training examples

  When to stop training

• **Applications**

  - Pattern Classification
  - Clustering/Categorization
  - Function approximation
  - Prediction/Forecasting
  - Optimization
  - Content-addressable Memory
  - Control
• Two Successful Applications

- Zipcode Recognition

- Text to voice translation (NeTtalk)