Introduction to Artificial Intelligence

Jaderick P. Pabico
Institute of Computer Science, College of Arts and Sciences
University of the Philippines Los Baños, College 4031, Laguna

CMSC 170 – Introduction to AI
2nd Semester 2009-2010

Neural Networks

- Artificial Neural Network
  - A new breed of computer system
- vs. the old breed/normal systems
  - Rule-based system: good at doing what the programmer wants it to do
  - Not so good at dealing with:
    - noisy data, data with missing parts, massive parallelism, fault tolerance, adaptation

Neural Networks

- Artificial Neural Networks (ANN)
  - information processing paradigm inspired by biological nervous systems, such as our brain
- ANN Structure
  - large number of highly interconnected processing elements (neurons) working together
- Like people, they learn from experience
  - by example

Neural Networks

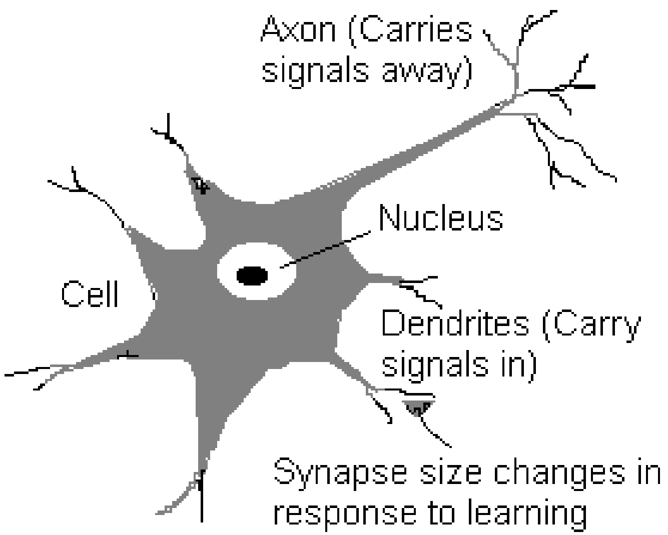
- Neural networks are configured for a specific application, such as pattern recognition or data classification, through a learning process
- In a biological system, learning involves adjustments to the synaptic connections between neurons
- same for ANN

Neural Networks

- When/where can neural network systems help?
  - when we can't formulate an algorithmic solution.
  - when we can get lots of examples of the behavior we require.
  - ‘learning from experience’
  - when we need to pick out the structure from existing data.

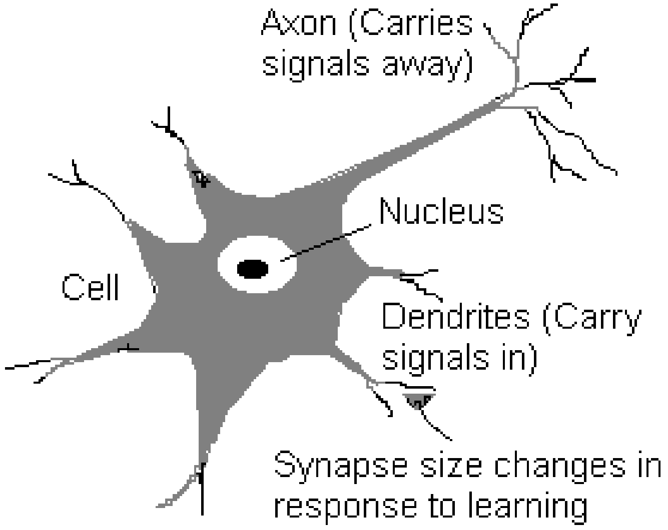
Neural Networks

- Inspiration from Neurobiology
  - A neuron: many-inputs / one-output unit
  - output can be excited or not excited
  - incoming signals from other neurons determine if the neuron shall excite ("fire")
  - Output subject to attenuation in the synapses, which are junction parts of the neuron



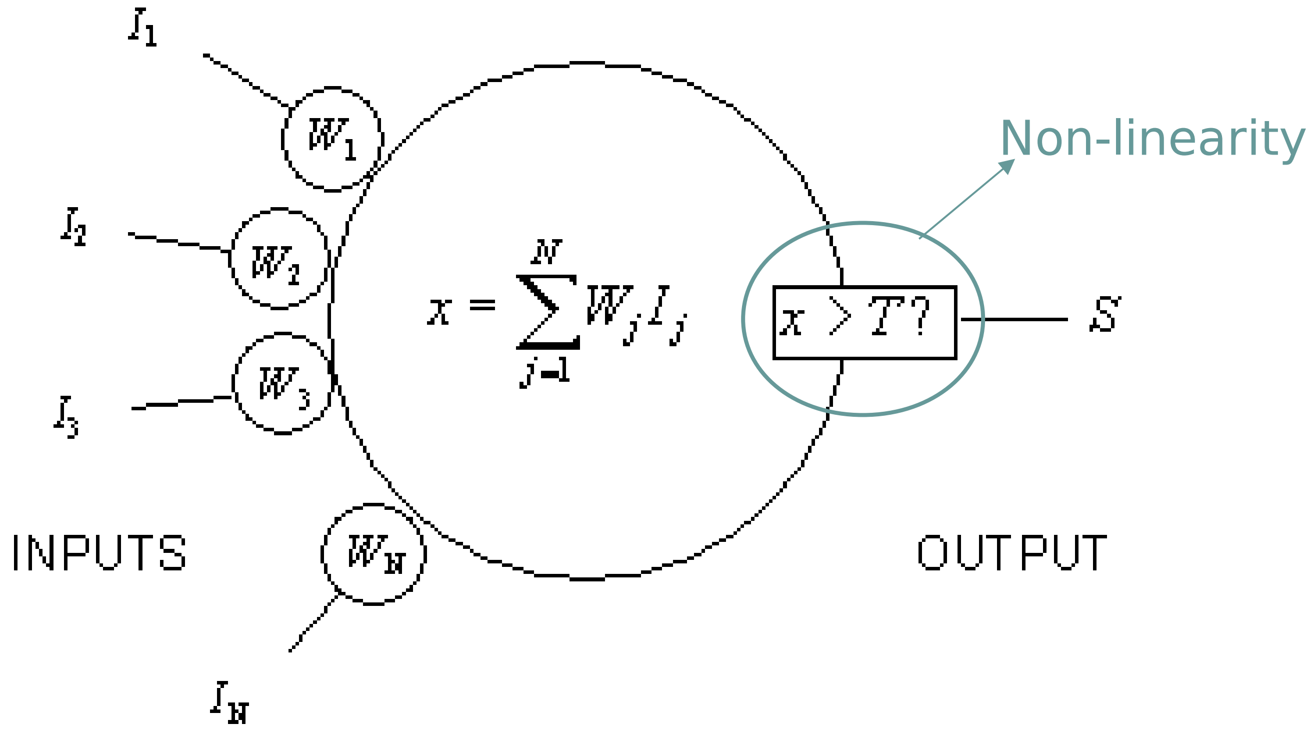
Neural Networks

- Synapse concept
  - The synapse resistance to the incoming signal can be changed during a "learning" process [1949]
  - **Hebb's Rule**
    - If an input of a neuron is repeatedly and persistently causing the neuron to fire, a metabolic change happens in the synapse of that particular input to reduce its resistance.



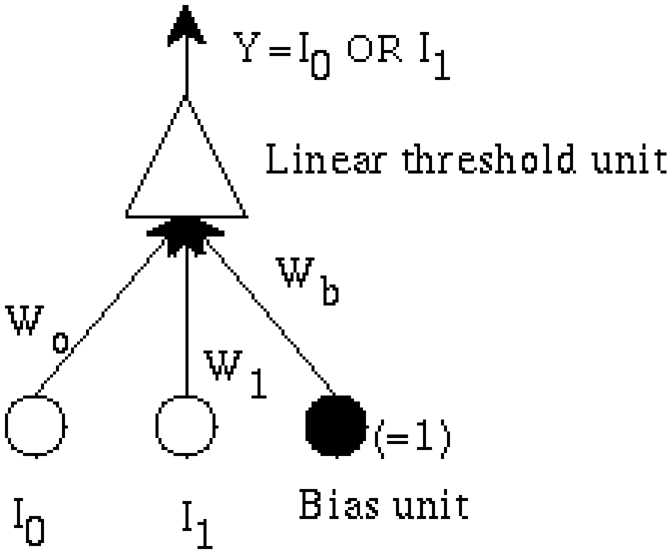
Neural Networks

- Mathematical representation
  - The neuron calculates a weighted sum of inputs and compares it to a threshold. If the sum is higher than the threshold, the output is set to 1, otherwise to -1.



Neural Networks

- A simple perceptron
  - It's a single-unit network



Neural Networks

- A simple perceptron
  - It’s a single-unit network
  - Change the weight by an amount proportional to the difference between the desired output and the actual output.

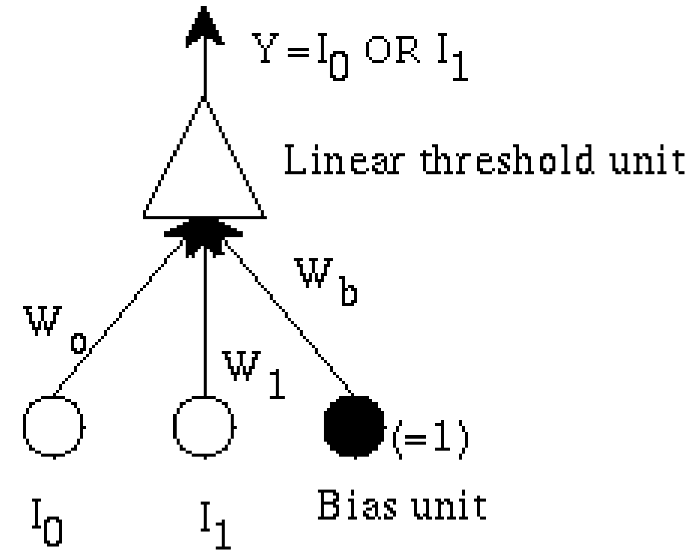
Input

$$\Delta W_i = \eta * (D-Y).I_i$$

Learning rate

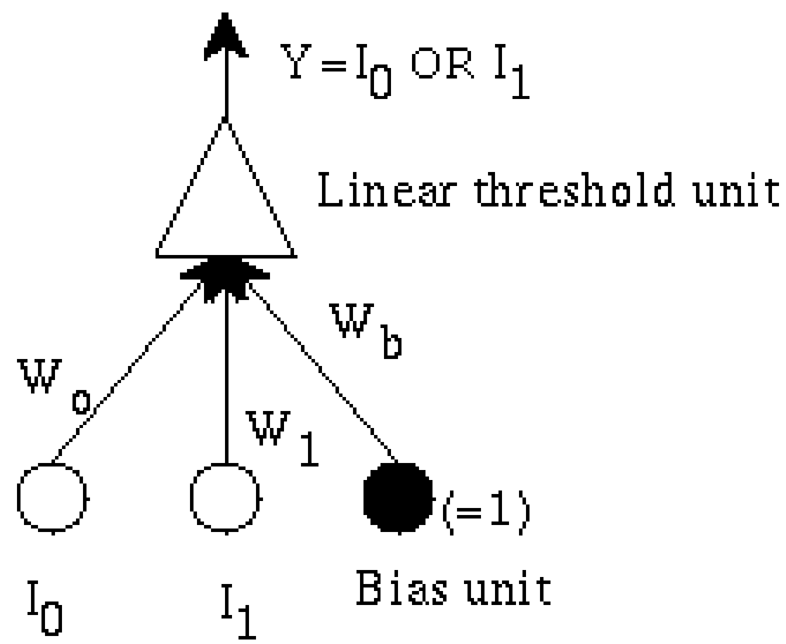
Desired output

Actual output



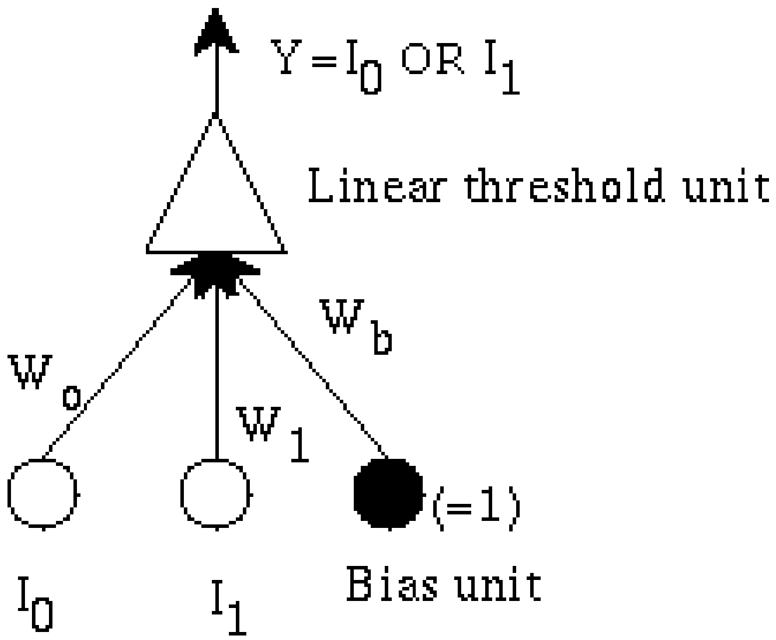
Neural Networks

- Example: A simple single unit adaptive network
  - The network has 2 inputs, and one output. All are binary.



Neural Networks

- Example: A simple single unit adaptive network
  - The network has 2 inputs, and one output. All are binary.
  - The output is
    - 1 if $W_0 I_0 + W_1 I_1 + W_b > 0$
    - 0 if $W_0 I_0 + W_1 I_1 + W_b \leq 0$



Neural Networks

- Example: A simple single unit adaptive network
  - The network has 2 inputs, and one output. All are binary.
  - The output is
    - 1 if $W_0 I_0 + W_1 I_1 + W_b > 0$
    - 0 if $W_0 I_0 + W_1 I_1 + W_b \leq 0$
  - We want it to learn simple OR: output a 1 if either I_0 or I_1 is 1.

Neural Networks

- Learning
  - From experience: examples / training data
  - Strength of connection between the neurons is stored as a weight-value for the specific connection
  - Learning the solution to a problem = changing the connection weights

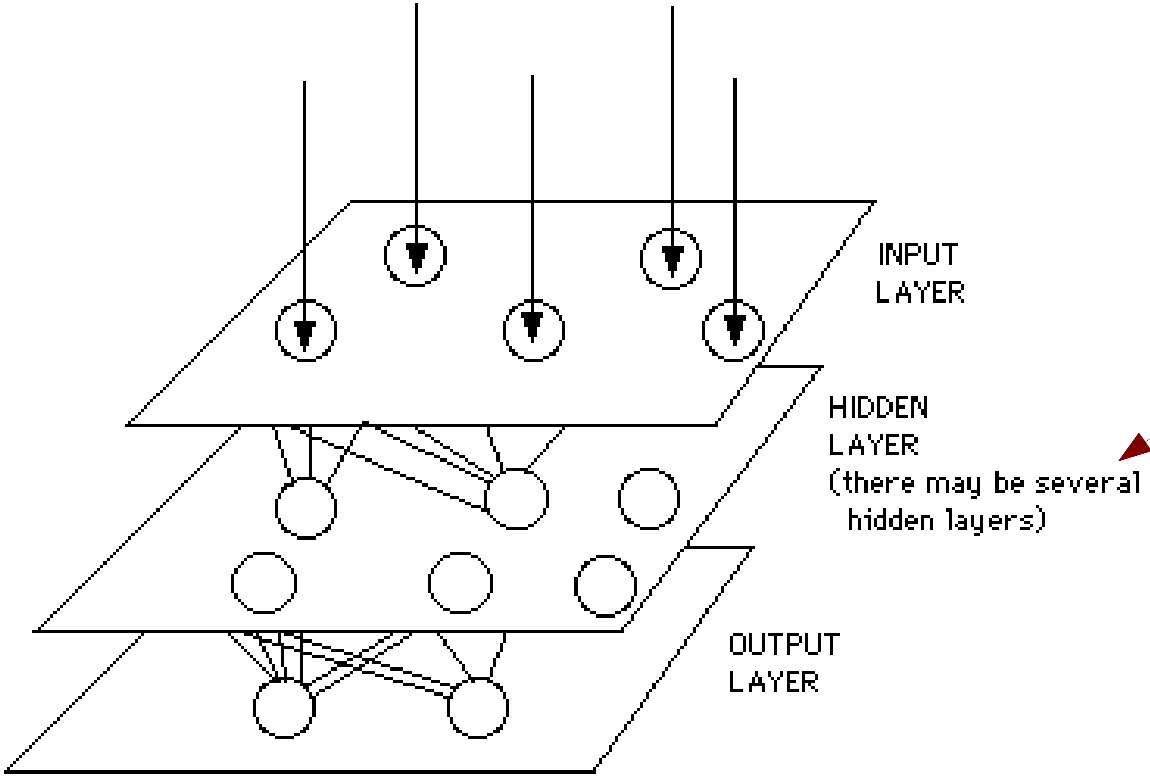
Neural Networks

- Operation mode
  - Fix weights (unless in online learning)
  - Network simulation = input signals flow through network to outputs
  - Output is often a binary decision
  - Inherently parallel
  - Simple operations and threshold:
    - fast decisions and real-time response

Neural Networks

- Adaptive interaction between individual neurons
- **Power**: collective behavior of interconnected neurons

The hidden layer learns to *recode* (or to *provide a representation* of) the inputs: associative mapping



Neural Networks

- Evolving networks
  - Continuous process of:
    - Evaluate output
    - Adapt weights
    - Take new inputs

LEARNING

  - ANN evolving causes stable state of the weights, but neurons continue working: network has ‘learned’ dealing with the problem

Neural Networks

- Learning performance
  - Network architecture
  - Learning method:
    - Unsupervised
    - Reinforcement learning
    - Backpropagation

Neural Networks

- Unsupervised learning
  - No help from the outside
  - No training data, no information available on the desired output
  - Learning by doing
  - Used to pick out structure in the input:
    - Clustering
    - Reduction of dimensionality or compression
  - Example: Kohonen's Learning Law

Neural Networks

- Competitive learning: example
  - Example: Kohonen network
    - Winner takes all
    - only update weights of winning neuron

    - Network topology
    - Training patterns
    - Activation rule
    - Neighbourhood
    - Learning

Neural Networks

- Reinforcement learning
  - Teacher: training data
  - The teacher scores the performance of the training examples
  - Use performance score to shuffle weights ‘randomly’
  - Relatively slow learning due to ‘randomness’

Neural Networks

- Back propagation
  - Desired output of the training examples
  - Error = difference between actual & desired output
  - Change weight relative to error size
  - Calculate output layer error , then propagate back to previous layer
  - Improved performance, very common!

Neural Networks

- Hopfield law
  - if the desired and actual output are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate.
  - One perceptron can only separate linearily!

Neural Networks

- Online / Offline
  - Offline
    - Weights fixed in operation mode
    - Most common
  - Online
    - System learns while in operation mode
    - Requires a more complex network architecture

Neural Networks

- Generalization vs. specialization
  - Optimal number of hidden neurons
  - Too many hidden neurons : you get an over fit, training set is memorized, thus making the network useless on new data sets
  - Not enough hidden neurons:
  - network is unable to learn problem concept
  - conceptually: the network's language isn't able to express the problem solution

Neural Networks

- Generalization vs. specialization
  - Overtraining:
    - Too much examples, the ANN memorizes the examples instead of the general idea
  - Generalization vs. specialization trade-off:
    - # hidden nodes & training samples

Neural Networks

- Where are ANN used?
  - Recognizing and matching complicated, vague, or incomplete patterns
  - Data is unreliable
  - Problems with noisy data
    - Prediction
    - Classification
    - Data association
    - Data conceptualization
    - Filtering
    - Planning

Neural Networks

- Prediction: learning from past experience
  - pick the best stocks in the market
  - predict weather
  - identify people with cancer risk
- Classification
  - Image processing
  - Predict bankruptcy for credit card companies
  - Risk assessment

Neural Networks

- Recognition
  - Pattern recognition: SNOOPE (bomb detector in U.S. airports)
  - Character recognition
  - Handwriting: processing checks
- Data association
  - Not only identify the characters that were scanned but identify when the scanner is not working properly

Neural Networks

- Data Conceptualization
  - infer grouping relationships (e.g. extract from a database the names of those most likely to buy a particular product.)
- Data Filtering
  - e.g. take the noise out of a telephone signal
- Planning
  - Unknown environments
  - Sensor data is noisy
  - Fairly new approach to planning

Neural Networks

- Strengths of ANN
  - Power: Model complex functions, nonlinearity built into the network
  - Ease of use:
    - Learn by example
    - Very little user domain-specific expertise needed
  - Intuitively appealing: based on model of biology, will it lead to genuinely intelligent computers/robots?
  - Neural networks cannot do anything that cannot be done using traditional computing techniques, BUT they can do some things which would otherwise be very difficult.

Neural Networks

- General Advantages
  - Advantages
    - Adapt to unknown situations
    - Robustness: fault tolerance due to network redundancy
    - Autonomous learning and generalization
  - Disadvantages
    - Not exact
    - Large complexity of the network structure
  - For motion planning?

Neural Networks

- Status of Neural Networks
  - Most of the reported applications are still in research stage
  - No formal proofs, but they seem to have useful applications that work