Diagnostic Methods

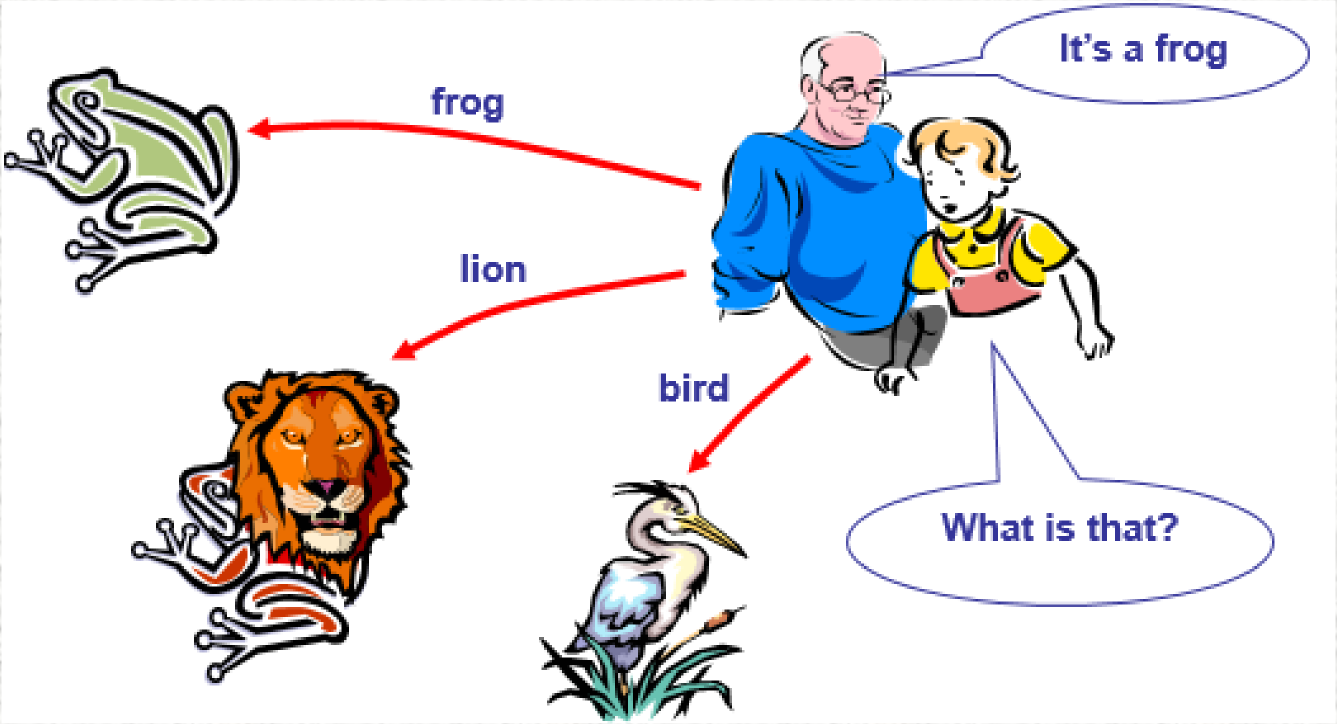
Non Model Diagnostic Methods
Artificial Neural Network
(ANN)

Outline

- Overview
- Artificial Neural Network (ANN)
  - Definition
  - Properties
  - Structure
  - Topologies
  - Applications
  - Learning Algorithms
  - Design Issues
  - Weakness/Strength
  - ANN in fault diagnosis
    - ANN applications

Overview

- Why and how are neural networks being used in solving problems?



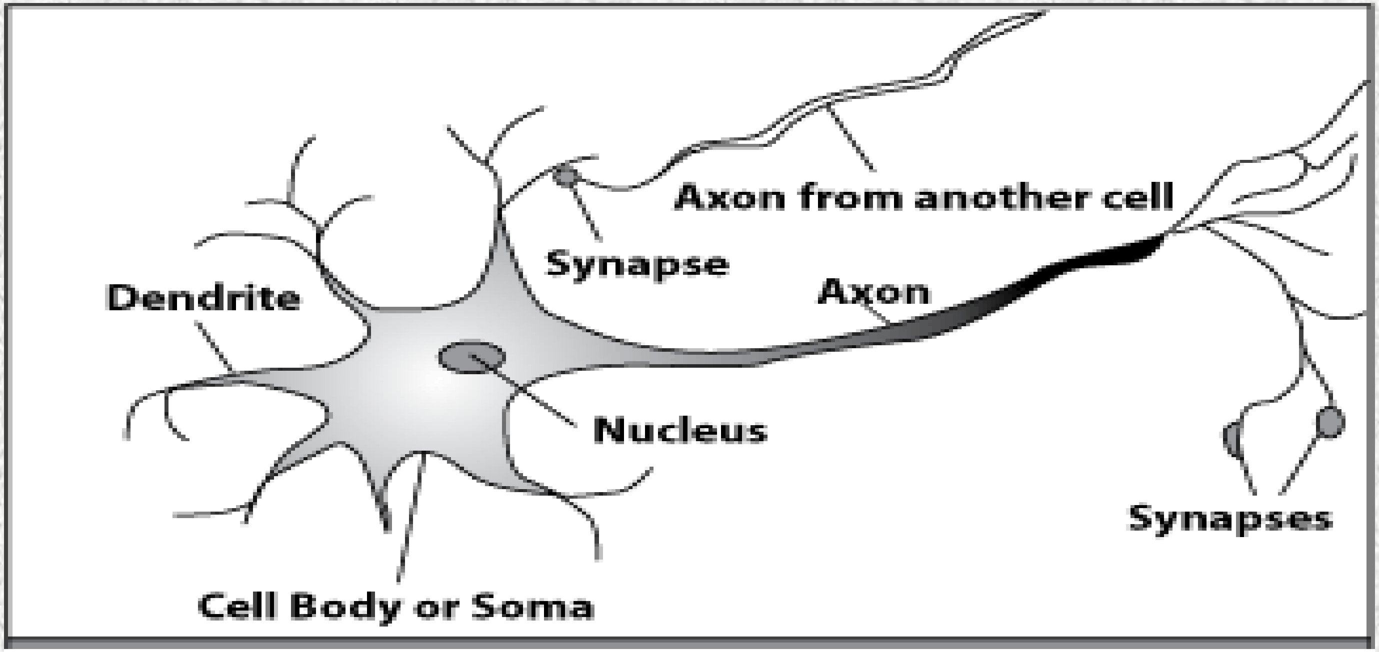
Overview

- “Data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain”.

(Tsoukalas & Uhrig, 1997)

Overview

- A processing element



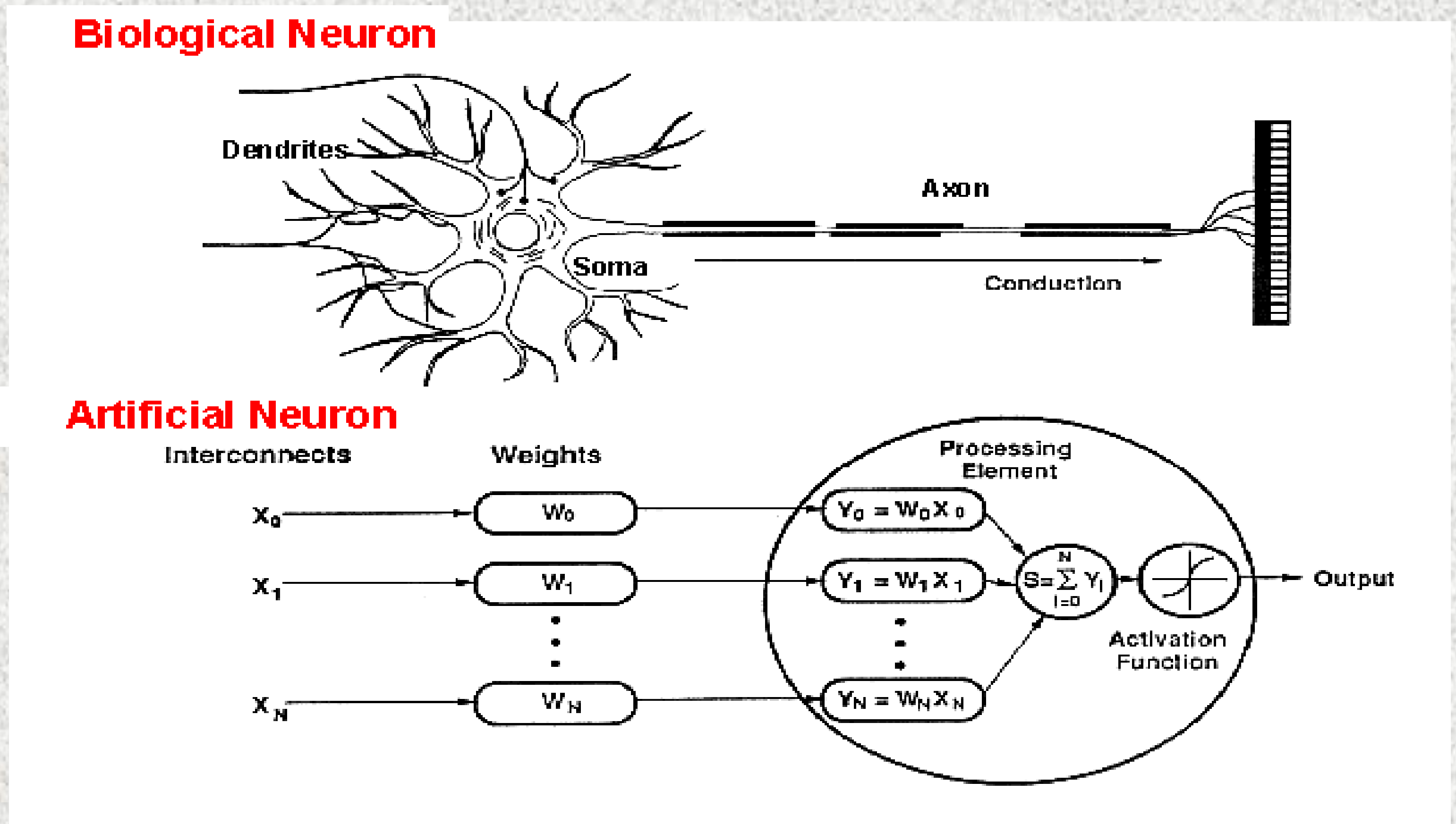
Dendrites: Input
Cell body: Processor
Synaptic: Link
Axon: Output

Artificial Neural Networks

- Humans perform complex tasks like vision, motor control, or language understanding very well.
- One way to build intelligent machines is to try to imitate the (organizational principles of) human brain.
- The brain is a highly complex and non-linear system, composed of some 10^{11} neurons that are densely connected ($\sim 10^4$ connection per neuron).

Overview

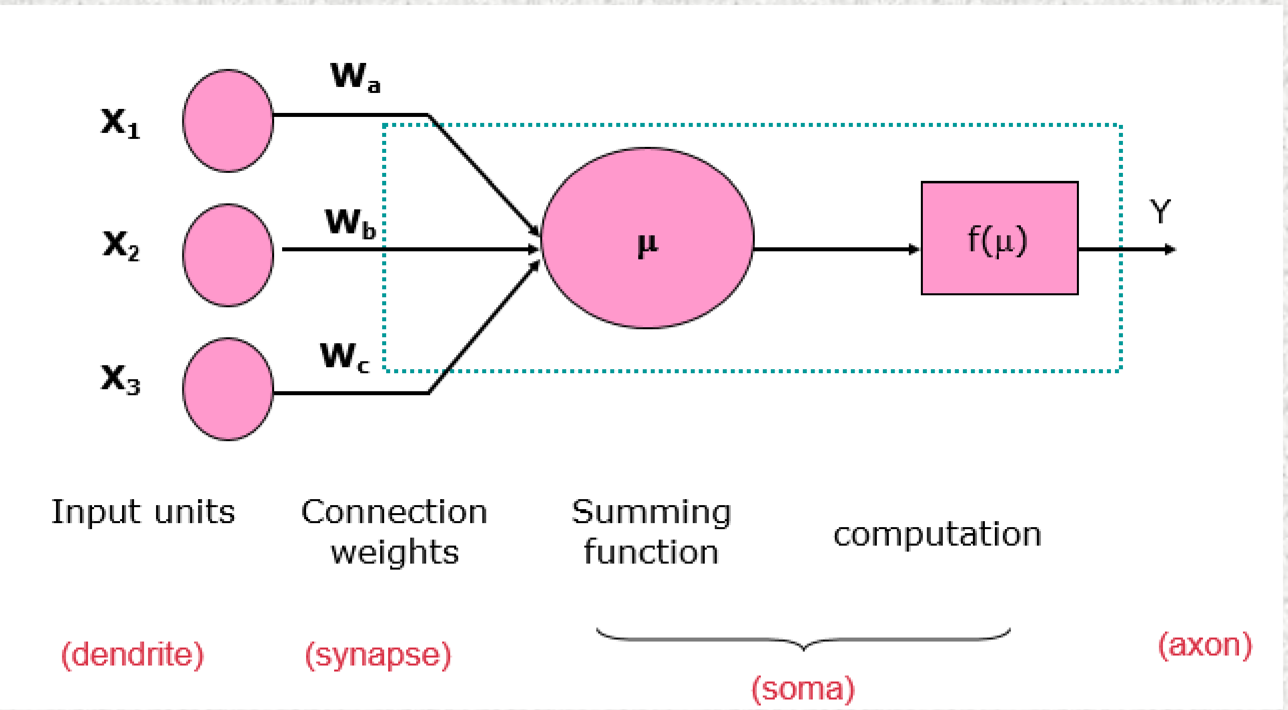
- An artificial neuron is an imitation of a human neuron.



Definitions

- A neural network is a massively parallel, distributed processor made up of simple processing units (artificial neurons).
- Distributed representation of knowledge over the connections. Knowledge is acquired by network through a learning process.
- The connections of the network and the strengths of the individual synapses establish the function of the network.

Definitions

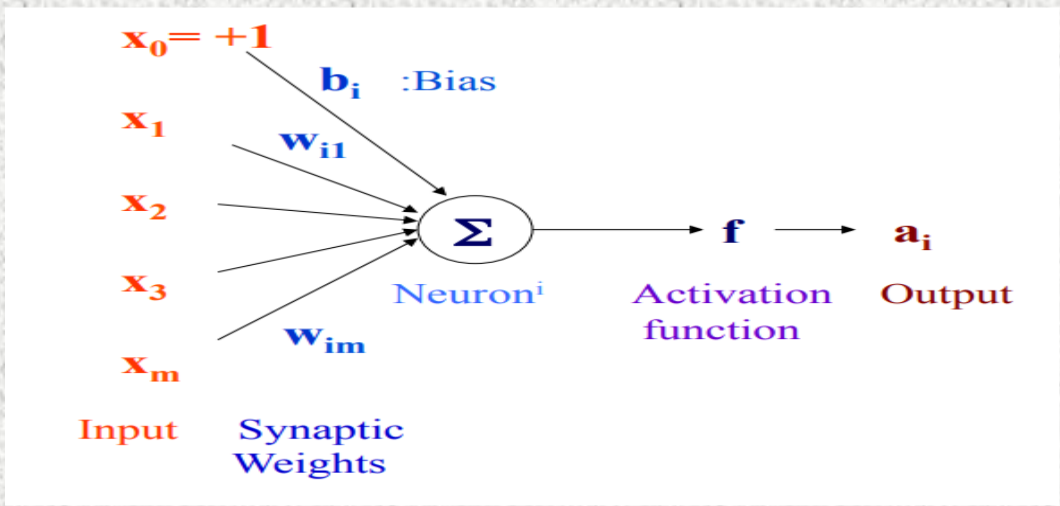


Properties

- Learning from examples:
  - labeled or unlabeled
- Adaptability:
  - changing the connection strengths to learn things
- Fault tolerance:
  - if one of the neurons or connections is damaged, the whole network still works quite well
- Thus, they might be better alternatives than classical solutions for problems characterized by:
  - high dimensionality, noisy, imprecise or imperfect data;
  - a lack of a clearly stated mathematical solution or algorithm

Structure

- Artificial neuron model consists of:



- The output of the neuron:

$$output = sum(weights * inputs) + bias$$

Structure

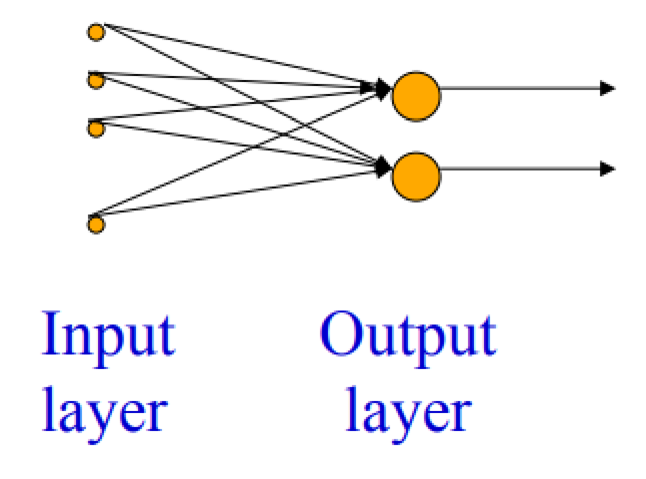
- An artificial neuron:
  - computes the weighted sum of its inputs (like synapses), called its net input.
  - adds its bias.
  - passes this value through an activation function.
- We say that the neuron “fires” (i.e. becomes active) if its output is above zero.
- Bias can be incorporated as another weight (strength of the connection), clamped to a fixed input of +1.0. This extra free variable (bias) makes the neuron more powerful.

Structure

- Activation function called the squashing function as it limits the amplitude (range) of the output of the neuron.
- Many types of activation functions are used:
  - linear: $a = f(n) = n$
  - threshold: $a = \begin{cases} 1 \text{ if } n >= 0 \\ 0 \text{ if } n < 0 \end{cases}$
  - sigmoid: $a = 1/(1+e^{-n})$

Topologies

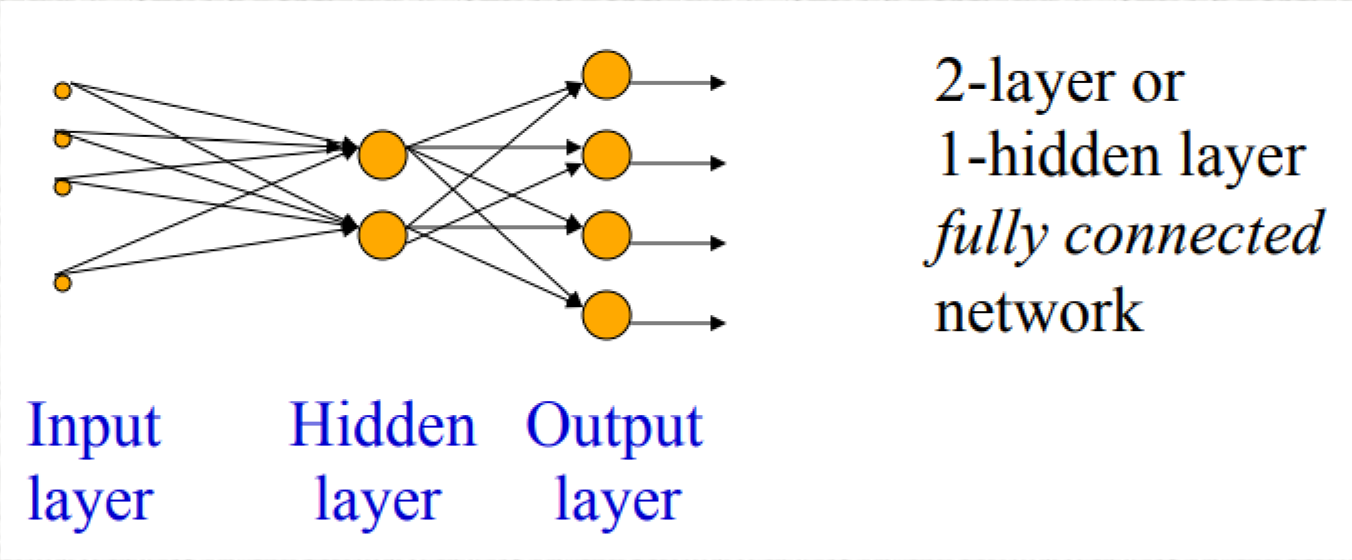
- Different network topologies:
  - Single layer feed-forward networks: Input layer projecting into the output layer



Topologies

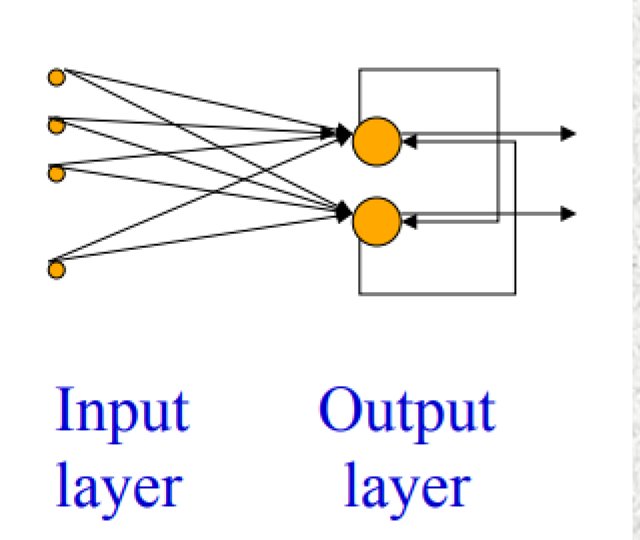
– Multi-layer feed-forward networks:

- One or more hidden layers
- Input projects only from one layer to the next



Topologies

– Recurrent/feedback networks: A network with feedback, where some of its inputs are connected to some of its outputs



Applications

- ANNs have been widely used in various domains for:
  - Classification, the aim is to predict the class of an input vector.
  - Optimization, the aim is to find the optimal values of parameters in an optimization problem.
  - Control, an appropriate action is suggested based on given an input vectors.
  - Data mining, with the aim of discovering hidden patterns from data (knowledge discovery).

Learning Algorithms

- Learn the connection weights from a set of training examples.
- Different network architectures required different learning algorithm. Depending on the weights, the computation of the neuron will be different.
- By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs.
- But when we have a large number of neurons, it would be quite complicated to find by hand all the necessary weights.
- However, we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called learning or training.

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.

Learning by trial-and-error

- Continuous process of:
  - Trial:
    - Processing an input to produce an output (In terms of ANN: Compute the output function of a given input).
  - Evaluate:
    - Evaluating this output by comparing the actual output with the expected output.
  - Adjust:
    - Adjust the weights.

Learning by trial-and-error

- Initialize the weights (w_0, w_1, …, w_k).
- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples.
  – Error function: $$E = \sum_i \left[Y_i - f(w_i, X_i)\right]^2$$
  – Find the weights w_i's that minimize the above error function:
    - e.g., gradient descent, backpropagation algorithm

Learning Algorithms

- 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.

Design Issues

- Initial weights (small random values $\in[-1,1]$)
- Transfer function (How the inputs and the weights are combined to produce output?)
- Error estimation
- Weights adjusting
- Number of neurons
- Data representation
- Size of training set

Design Issues

- Many neurons:
  - Higher accuracy
  - Slower
  - Risk of over-fitting
    - Memorizing, rather than understanding
    - The network will be useless with new problems.
- Few neurons:
  - Lower accuracy
  - Inability to learn at all
- Optimal number

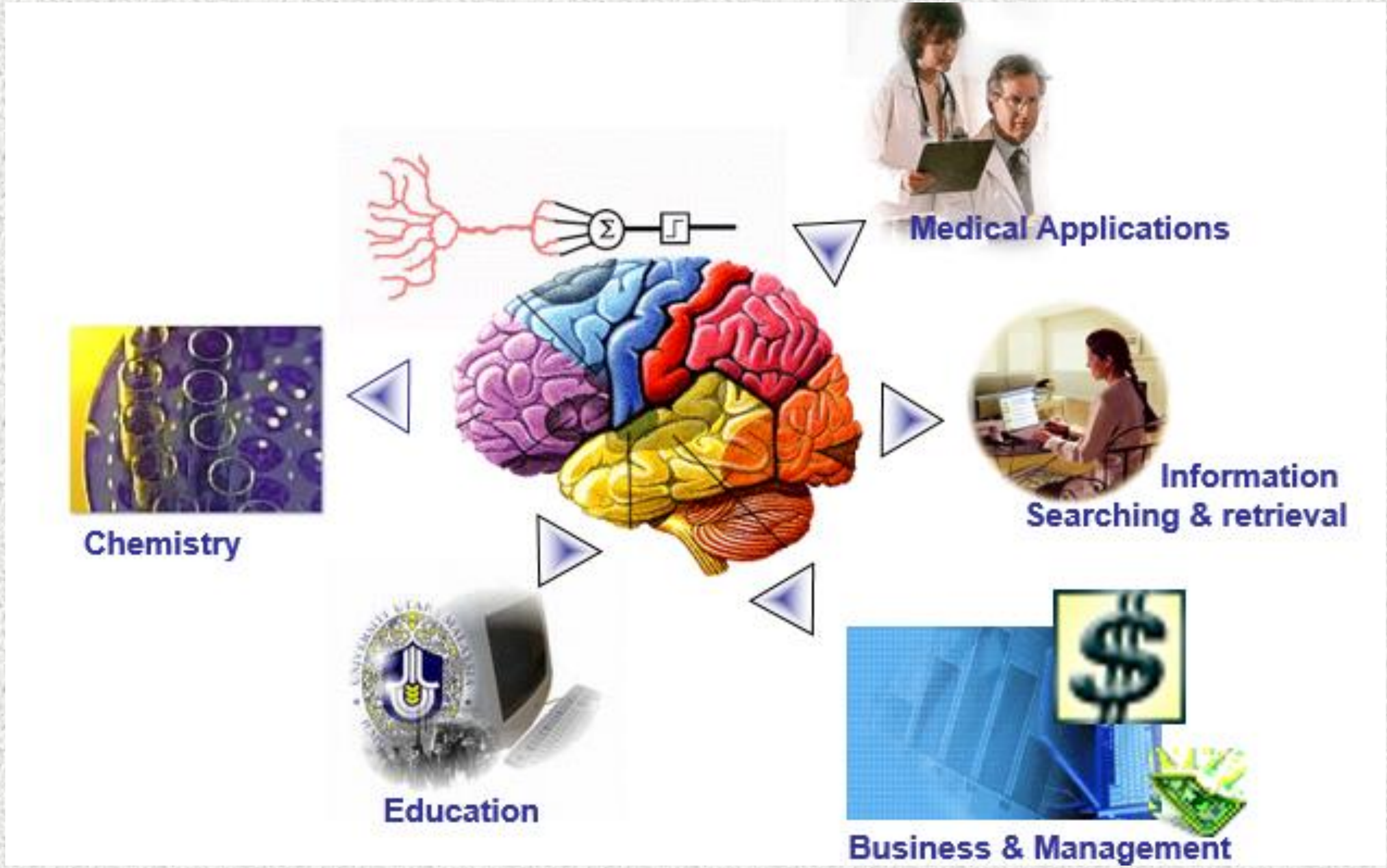
Design Issues

- Usually input/output data needs pre-processing:
  - Pictures: Pixel intensity
  - Text: A pattern
- Size of training set:
  - Over fitting can occur if a “good” training set is not chosen
  - What constitutes a “good” training set?
    - Samples must represent the general population.
    - Samples must contain members of each class.
    - Samples in each class must contain a wide range of variations or noise effect.
- The size of the training set is related to the number of hidden neurons.

Weakness/Strength

- Weakness
  - Long training time
  - Require a number of parameters typically best determined empirically, e.g., the network topology or "structure."
  - Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of "hidden units" in the network
- Strength
  - High tolerance to noisy data
  - Ability to classify untrained patterns
  - Well-suited for continuous-valued inputs and outputs
  - Successful on a wide array of real-world data
  - Techniques have recently been developed for the extraction of rules from trained neural networks

ANN Applications



References

- Craig Heller, and David Sadava, *Life: The Science of Biology, fifth edition*, Sinauer Associates, INC, USA, 1998.
- Introduction to Artificial Neural Networks, Nicolas Galoppo von Borries
- Tom M. Mitchell, *Machine Learning*, *WCB McGraw-Hill*, *Boston*, *1997*.