Neural Networks - Part I (continued)

Background Definitions

A neural network is a computing system which consists of a number of simple, highly interconnected processing elements which process information by determining the value of an output signal based on the values of several input signals.
After knowledge has been learned, it is stored in the way the neurons are connected and in the way the individual neurons adjust their weights as the network learns.

Unlike expert systems, neural networks do not require the user to specify a number of "if-then" rules. The network only requires specific examples of input values along with the corresponding output values. One essentially teaches the network how to respond to a set of specific examples. The network determines the rules that work for the specific examples.

The first layer of a network is called the input layer and the last layer is called the output layer. The middle layer or layers are referred to as hidden layers. Each processing unit has a number of input signals and one output signal that usually fans out to several neurons in the next layer. Figure 1 illustrates a neural network with three neurons.

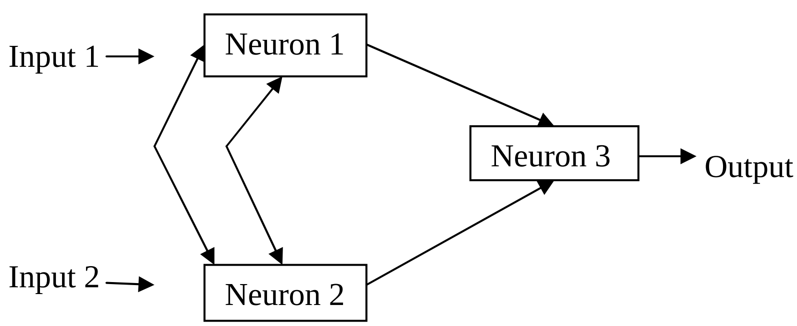


Figure 1. Network with three neurons

Neuron Weights

The output of a neuron is usually determined from the weighted sum of the input values. Each connection to the neuron has a different effect on the input depending upon the weight of the connection. The weights change as the network learns.

Figure 2 illustrates a neuron with input values of 3 and 6, and with corresponding weights of 0.2 and 0.1. For the example in Figure 2, the output value of Y1 is 1.2.

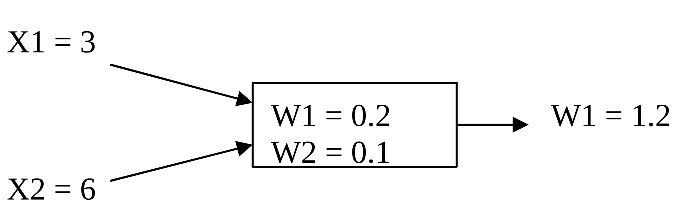


Figure 2. Neuron showing weighted inputs

The output of 1.2 is obtained as follows:

Y1 = (W1 x X1) + (W2 x X2)
Y1 = (3 x 0.2) + (6 x 0.1)
Y1 = 1.2.

The Transfer Function

The weighted sum of the inputs to a neuron is usually modified by a mathematical transfer function. The transfer function modified the weighted sum of the input values to a reasonable value before passing the signal onto the next layer. If the input sum were not modified, the output could become very large, especially with many input connections. For our example, a function is used that transforms any real number into a number between 0 and 1:

$$f(X) = \frac{1}{(1 + e^{-x})}$$

Where, e = 2.71828.

In the example, f(1.2) becomes 0.768

Why Artificial Neural Networks?

The long course of evolution has given the human brain many desirable characteristics not present in von Newmann modern parallel computers. These include:

- Massive parallelism,
- Distributive representation and computation,
- Learning ability,
- Generalization ability,
- Adaptivity,
- Inherent contextual information processing,
- Fault tolerance,
- Low energy consumption

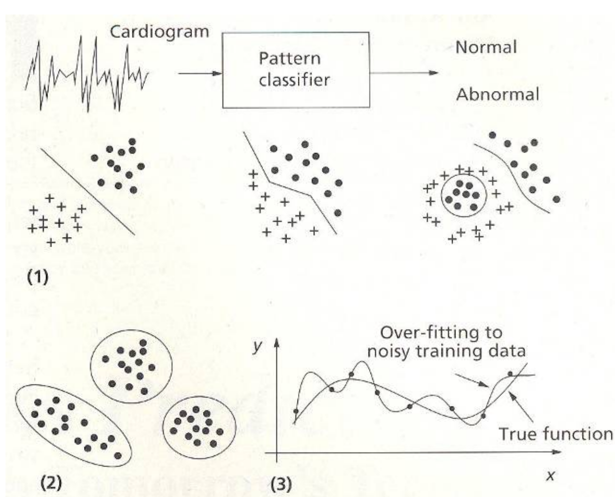
Table I shows the capabilities of the Von Newmann computer versus those of the biological neural system.

Table 1. Von Neumann computer versus biological neural system.

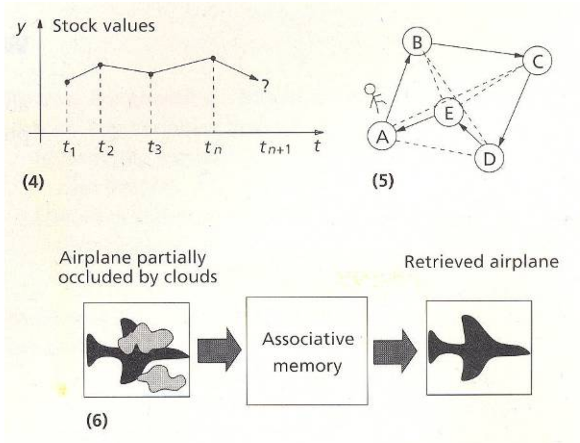
	Von Neumann computer	Biological neural system
Processor	Complex High speed One or a few	Simple Low speed A large number
Memory	Separate from a processor Localized Noncontent addressable	Integrated into processor Distributed Content addressable
Computing	Centralized Sequential Stored programs	Distributed Parallel Self-learning
Reliability	Very vulnerable	Robust
Expertise	Numerical and symbolic manipulations	Perceptual problems
Operating environment	Well-defined, well-constrained	Poorly defined, unconstrained

Thus, the biological neural system architecture is completely different from the von Newman architecture. The difference significantly affects the type of functions each computational model can best perform.

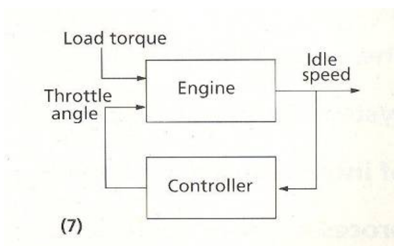
Inspired by biological neural networks, artificial neural networks (ANN) are massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. ANN models attempt to use some “organizational” principles believed to be used in the human. **Figure** 3 (1-7) portrays typical tasks of special interest to computer scientists and engineers that ANN can perform.



(1) Pattern classification; (2) clustering/categorization; (3) function approximation;



(4) prediction/forecasting; (5) optimization; (6) retrieval by content;



(7) control (engine idle speed). (Adapted from DARPA Neural Network Study).

II. NEURAL NETWORK APPLICATIONS

This section will illustrate some of the most important applications of ANNs.

A. *Application to control*.

Here, the ANN makes use of nonlinearities, learning, parallel processing, and generalization capabilities for application to advanced intelligent control.

In supervised control, a ANN learns the mapping from sensor input to desired actions by adapting to a training set of examples of what it should do, Figure 4.

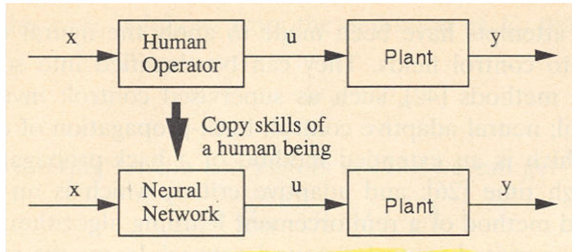


Figure 4. Supervised Control.

For example, for modeling and acquiring human skills and transmitting them to robots and telerobots, the neural network uses teaching data as shown by Asada [1].

In inverse control, a neural network learns the inverse dynamics of a system Figure 5.

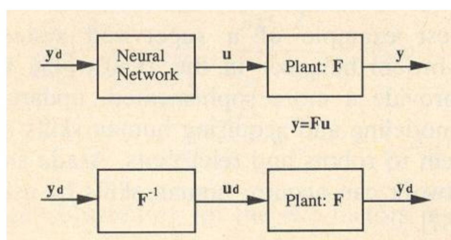


Figure5. Inverse Control

The ANN is used in the control loop. For trajectory control of a robotic manipulator some researchers [2] have used inverse control indirectly with optimal control as shown in Figure 6.

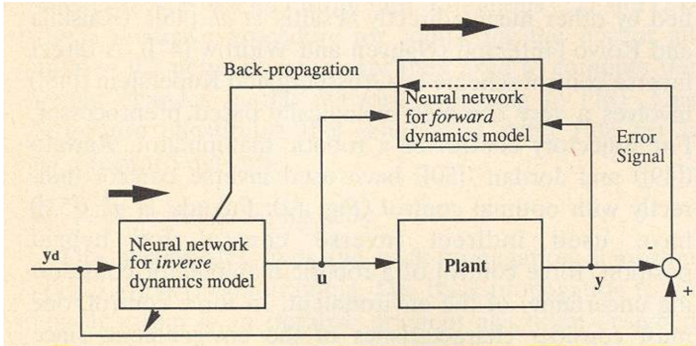


Figure 6. Indirect Inverse Control (forward and inverse modeling)

In force control, one must consider characteristics of the environment, since they affect the control loop. However, it is difficult to take account of uncertainties in the environment. Therefore, a ANN-based controller is effective in force control because of its adaptability.