Dr. George Karraz, Ph. D.

Neural Networks
Lectures I,II

Dr. George Anwar Karraz, Ph. D.

Contents:

1. Introduction
2. The Artificial Neuron
3. Biological Neural Networks
4. Where are Neural Nets Being Used
5. Neural Networks Architectures
6. Training Methods.
7. Common Activation Functions
8. The Development of Neural Networks

1. Introduction

An artificial neural network is an information- processing system that has certain performance characteristics in common with biological neural networks.

Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology based on the assumptions that:

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links.
3. Each connection link has an associated weight, which, in a typical neural net , multiplies the signal transmitted.
4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

1. Introduction

- A neural network is characterized by its:

1. **Architecture:** its pattern of connections between the neurons.
2. **Learning algorithm**: its method of determining the weights on the connection.
3. **Activation function**.

1. Introduction

- Some of the artificial neural networks applications are:

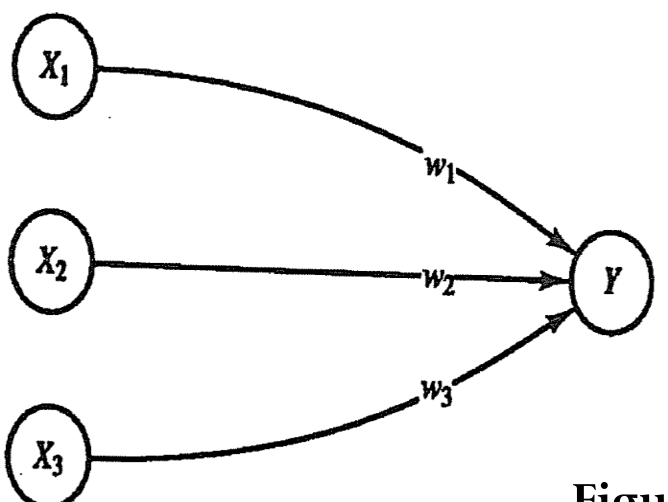
1. Storing and recalling data or patterns.
2. Classifying patterns.
3. Performing general mappings from input patterns to output patterns.
4. Grouping similar patterns.
5. Finding solutions to constrained optimization problems

2. The Neuron

- Each neuron has an internal state, called its activation or activity level, which is a function of the inputs it has received.
- A neuron sends its activation as a signal to several other neurons.
- <u>Its important to note that a neuron can send only one signal at a time, although that signal is broadcast to several other neurons.</u>

2. The Neuron

- Example: consider a neuron Y, illustrated in the Figure 1.1 that receives inputs from neurons X1, X2, and X3. The activations (output signals) of these neurons are x1, x2, and x3, respectively. The weights on the connections are w1, w2, and w3 respectively. The net input, y-in to neuron Y is the sum of the weighted signals from neurons X1, X2, and X3.



$$y\_in = w_1x_1 + w_2x_2 + w_3x_3.$$

the activation of neuron Y is given by:

$$y = f(y\_in)$$

Figure 1.1 A simple artificial neuron

2. The Neuron

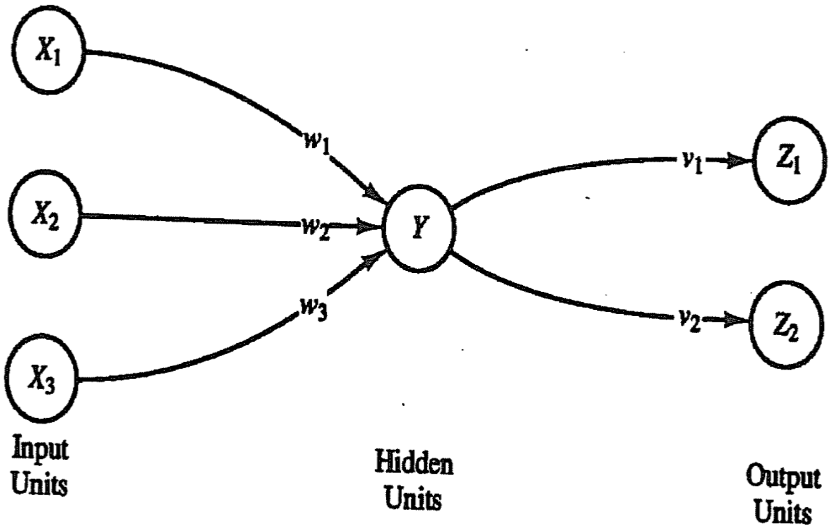


Figure 1.2 A simple neural network

- Now suppose further that neuron Y connected to neurons Z1 and Z2, with weights v1, and v2 respectively, as shown in Figure 1.2. neuron Y sends its signal y to each of these units. However, in general the values received by neurons Z1 and Z2, will be different, because each signal is scaled by the appropriate weight v1, and v2. In a typical net, the activations z1, and z2 of neurons Z1, and Z2 would depend on inputs from several or even many neurons, not just one, as shown in this simple example.

2. The Neuron

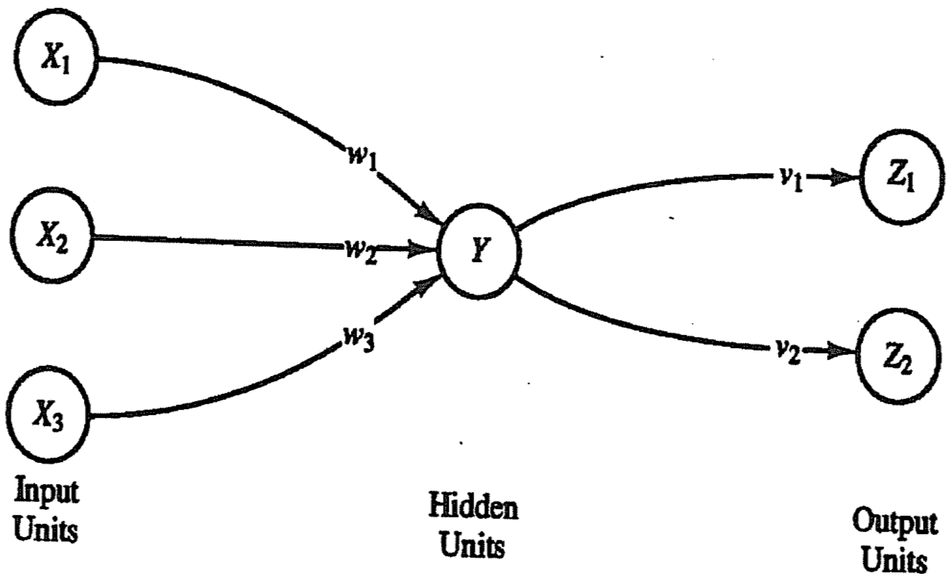


Figure 1.2 A simple neural network

- Although the neural network in Figure 1.2 is very simple, the presence of a hidden units, together with a nonlinear activation function, gives it the ability to solve many more problems than can be solved by a net with only input and output units. On the other hand, it is more difficult to train a net with hidden units (finding optimal values for the weights)

3. Biological Neural Networks

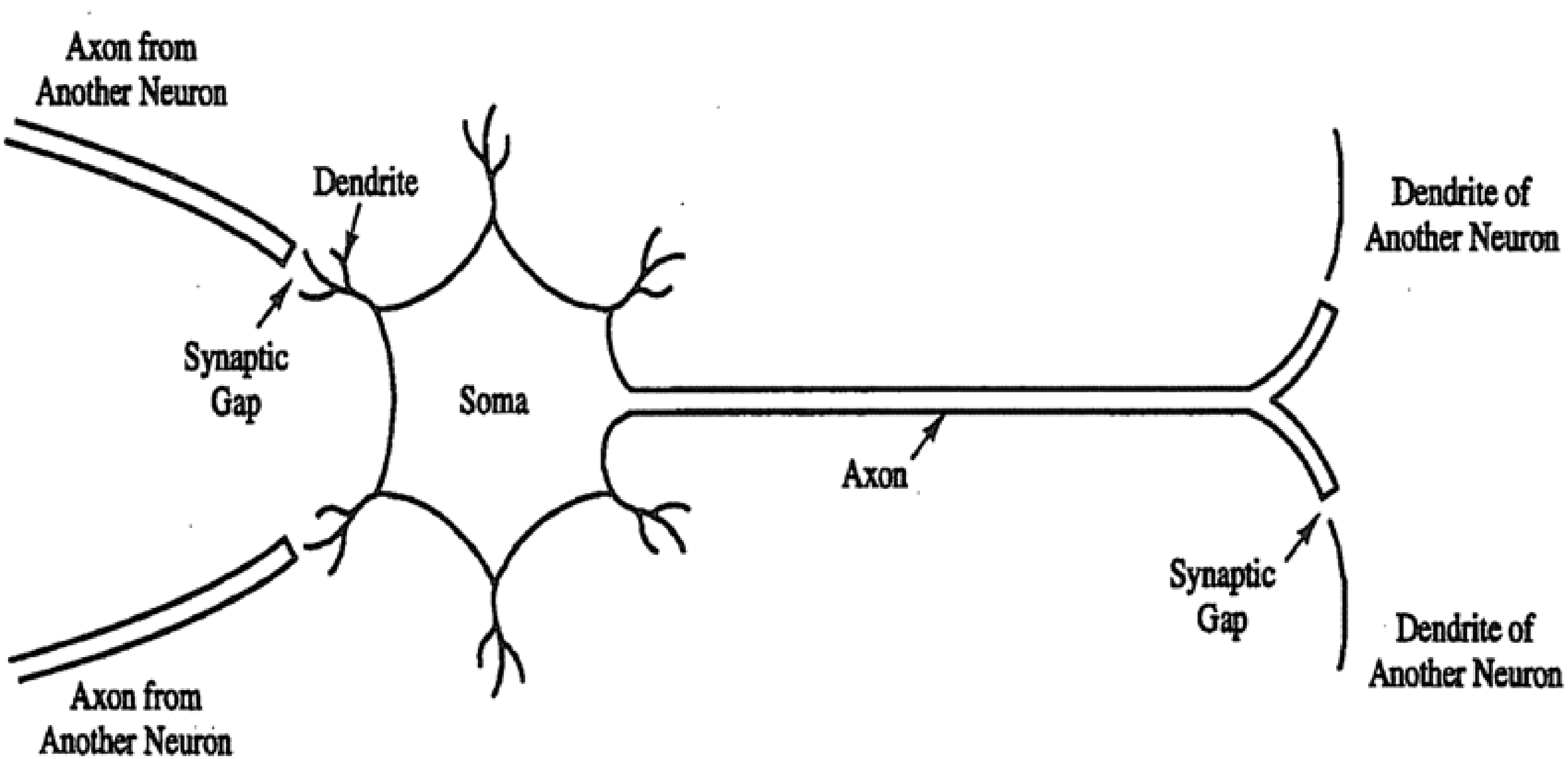


Figure 1.3 A Biological Neuron

3. Biological Neural Networks

- A biological neuron has three types of components that are of particular interest in understanding an artificial neuron its:

  Dendrites, **Soma**, and **Axon**.
- The many dendrites receive signals from other neurons. The signals are electric impulses that are transmitted across a synaptic gap by means of chemical process.
- The action of chemical transmitter modifies the incoming signal (typically, by scaling the frequency of the signals that are received) in a manner similar to the action of the weight in an artificial neural network.
- The cell body sums the incoming signals. When sufficient input is received, then it transmits a signal over its axon to other cells.
- The transmission of the signal from a particular neuron is accomplished by an action potential resulting from differential concentration of ions on either side of the neuron's axon sheath (the brain's white matter) the ions most directly involved are potassium, sodium, and chloride.
- A generic biological neuron is illustrated in Figure 1.3.

3. Biological Neural Networks

- Several key features of the processing elements of artificial neural networks are suggested by the properties of biological neurons:

1. The processing element receives many signals
2. Signals may be modified by a weight at the receiving synapse.
3. The processing element sums the weighted inputs.
4. The output from a particular neuron may go to many other neurons (the axon branches).
5. A synapse's strength may be modified by experience.
6. Memory is distributed: Long-term memory resides in the neurons' synapses or weights, and Short-term memory corresponds to the signals sent by the neurons.

4. Where are Neural Nets Being Used

- Signal Processing
- Control
- Pattern Recognition
- Medicine
- Speech Production
- Speech Recognition
- Business

5. Neural Networks Architectures
5.1. Single-Layer net

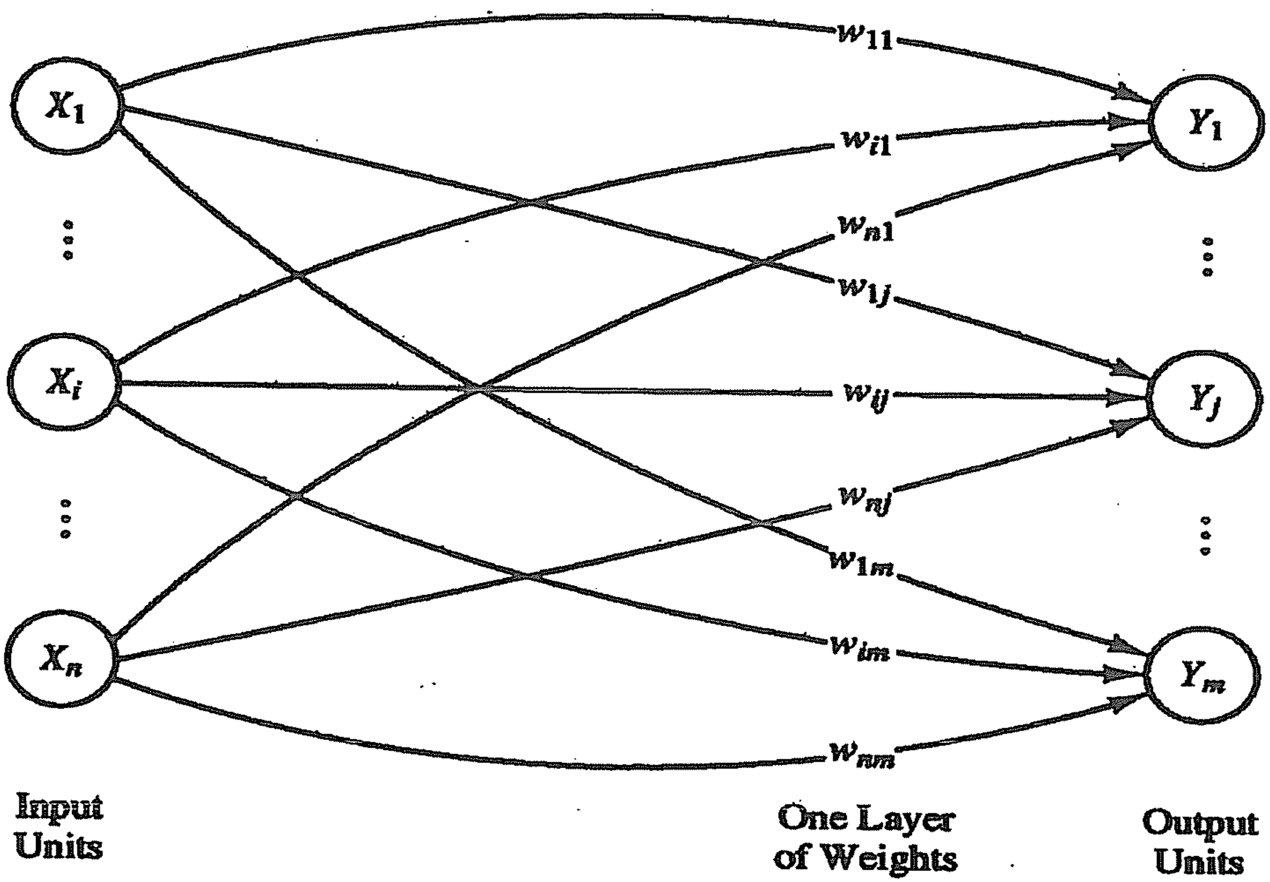


Figure 1.4 A single-layer net

5. Neural Networks Architectures

5.1. Single-Layer net

1. **Single Layer Net:** A single layer of connection weights. Often, the units can be distinguished as input units, which receive signals from the outside world, and output units, from which the response of the net can be read. In the typical single-layer net shown in Figure 1.4 the input units are fully connected to output units. An example of a single layer net is **Hopfield** net.

5.Neural Networks Architectures

5.1. Single-Layer net

- Note:

1. For pattern classification, each output unit corresponds to a particular category to which an input vector may or may not belong.
2. The weights for one output unit do not influence the weights for other output units.
3. We can use this architecture for pattern association, but now the overall pattern of output signals gives the response pattern associated with the input signal that caused it be produced.

5. Neural Networks Architectures

5.2. Multilayer net

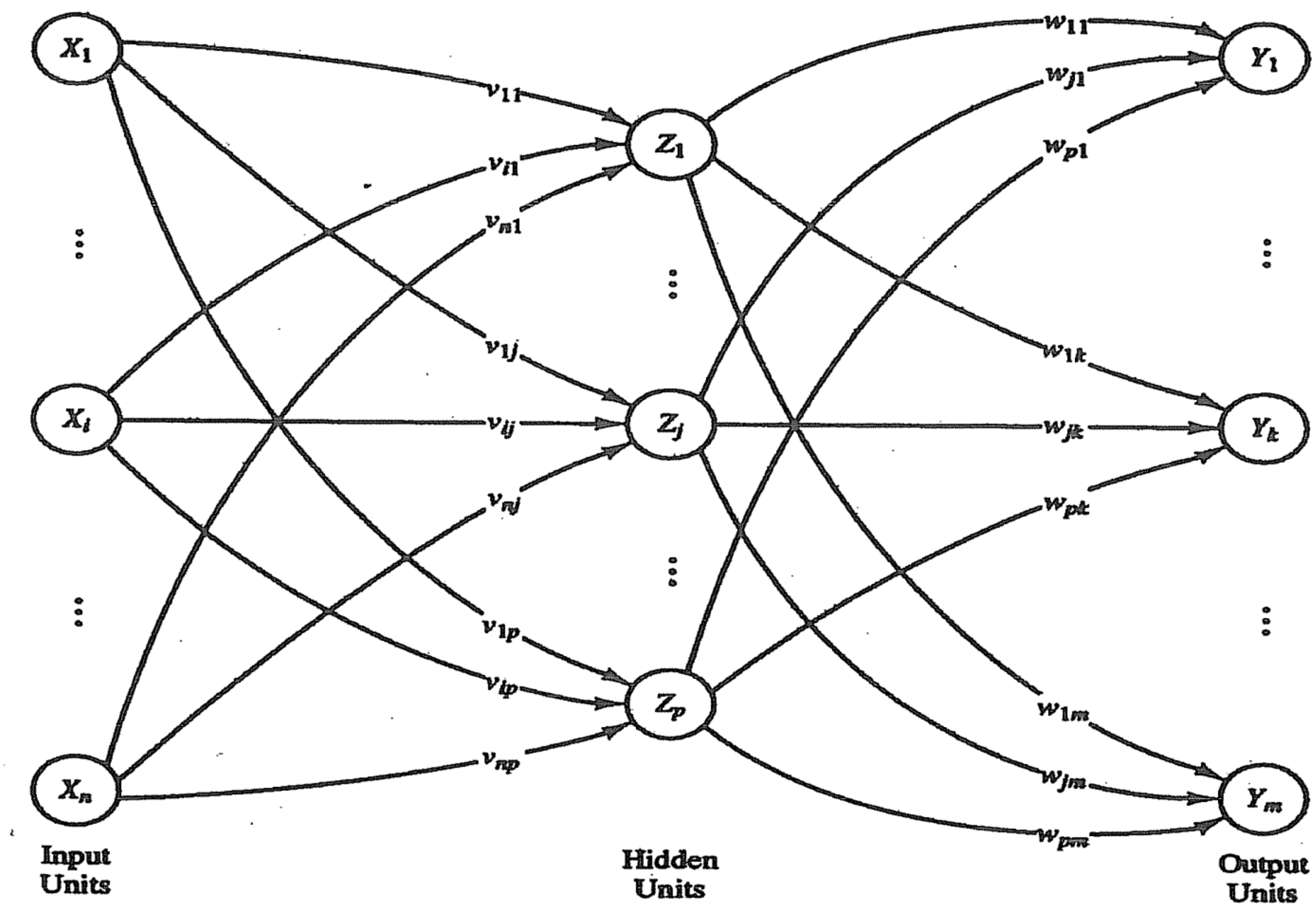


Figure 1.5 A Multilayer net

5. Neural Networks Architectures

5.2. Multilayer net

- Multilayer net is a net with one or more layers (or levels) of nodes (the so-called hidden units) between the input and the output units. Typically, there is a layer of weights between two adjacent levels of units (input, hidden, or output)

Multilayer nets can solve more complicated problems than can single-layer nets, but training may be more difficult. However, in some cases, training may be more successful, because it is possible to solve a problem that a single- layer net cannot be trained to perform correctly. See Figure 1.5.

5. Neural Networks Architectures

5.3. Competitive layer

- A competitive layer forms a part of a large number of neural networks.
- Typically, the inter-connections between neurons in the competitive are not shown in the architecture diagrams for such net.
- An example of the architecture for a competitive layer is given in Figure 1.6.

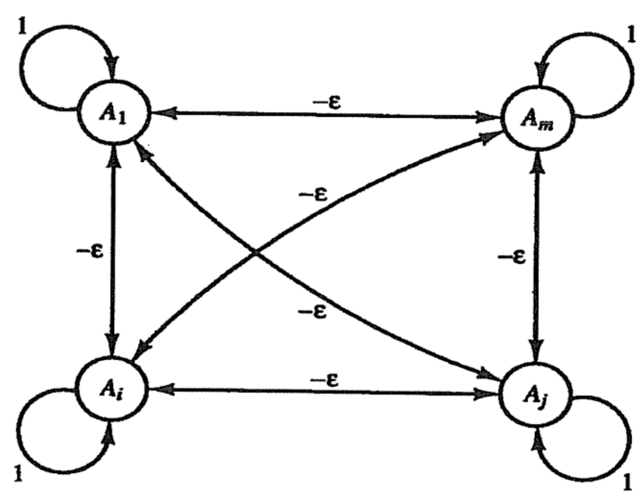


Figure 1.6 A competitive layer

6. Training Methods

- The method of setting the values of the weights (training) is an important distinguishing characteristic of different neural networks.
- We shall distinguish two types of training: supervised and unsupervised for a neural network, in addition, there are nets whose weights are fixed without an iterative training process.
- There is some ambiguity in the labeling of training methods as supervised or unsupervised, and some authors fine a third category, self-supervised training, useful.
- In general there is a useful correspondence between the type of training that is appropriate and the type of problem we wish to solve.

6. Training Methods

6. 1. Supervised Training

- In perhaps the most typical neural net setting, training is accomplished by presenting a sequence of training vectors, or patterns, each with an associated target output vector. The weights are then adjusted according to a learning algorithm. This process is known as **supervised training.**
- Some of the simplest neural nets are designed to perform pattern classification, i.e., to classify an input vector as either belonging or not belonging to a given category.
- In this types of neural net, the output is bivalent element, say 1 (if the input vector belongs to the category) or -1 (if it does not belong).
- Pattern association is another special form of mapping problem, one in which the desired output is not just a (yes) or (no) , but rather a pattern. A neural network that is trained to associate a set of input vectors with corresponding set of output vectors is called an ***associative memory***. If the desired output vector is the same as the input vector, the net is an ***auto associative memory***; If the output vector is different from the input vector, the net is an ***heteroassociative memory***

6. Training Methods
6.2. Unsupervised Training

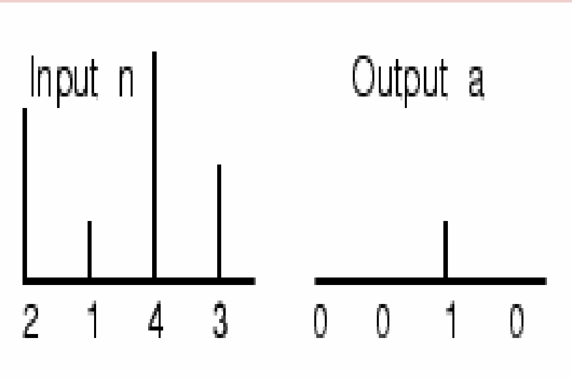
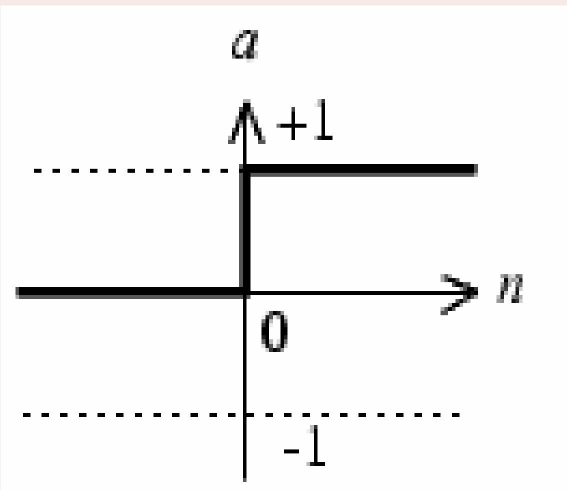
- Self organizing neural nets group similar vectors together without the use of training data to specify what a typical member of each group looks like or to which group each vector belongs.
- A sequence of input vectors is provided , but no target vectors are specified.
- The net modifies the weights so that the most similar input vectors are assigned to the same output (or cluster)unit.
- The neural network will produce an exemplar (representative) vector for each cluster formed self organizing nets called (**Kohonen** self-organizing maps) and it is described by the adaptive resonance theory.

6. Training Methods

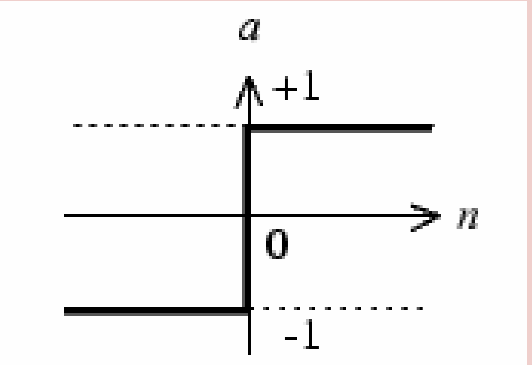
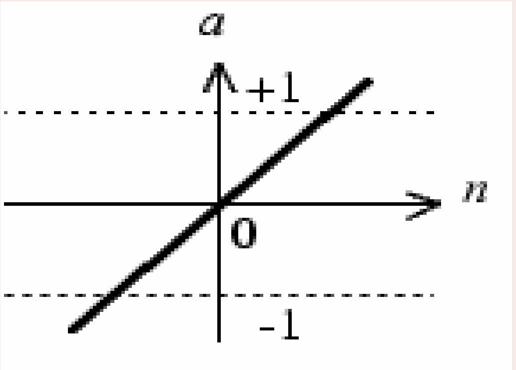
6. 3. Fixed weight net

- Still other types of neural networks can solve constrained optimization problems. Such nets may work for problems that can cause difficulty for traditional techniques, such as problems with conflicting constraints (i.e., not all constraints can be satisfied simultaneously).
- When these nets are designed, the weights are set to represent the constraints and the quantity to be maximized or minimized.
- **Boltzmann** machine (without learning) and the continuous **Hopfield** net can be used for constrained optimization problems.
- Fixed weights are also used in contrast enhancing nets.

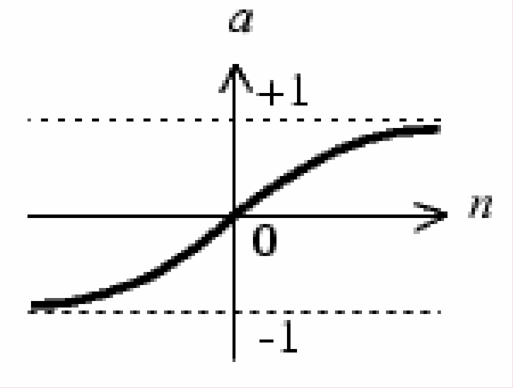
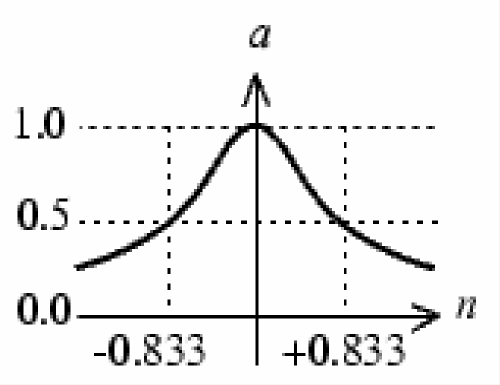
7. Common Activation Functions

Activation Function	Description	Representation
Softmax	**Returns output vectors with elements between 0 and 1, but with their size relations intact. Softmax does not has a derivative function.**	
Hard-limit	**Limits neuron output to either 0, if the net input argument n is less than 0, or 1, if n is greater than or equal to 0. This function is used to create neurons that make classification decisions.**	

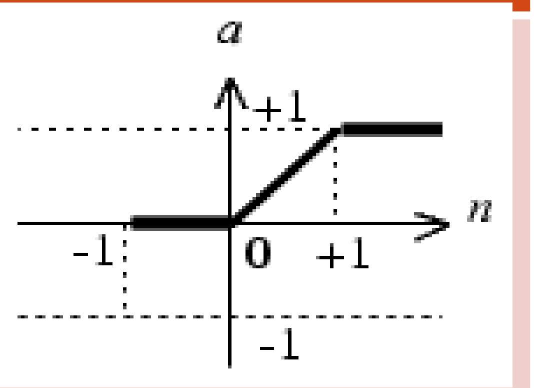
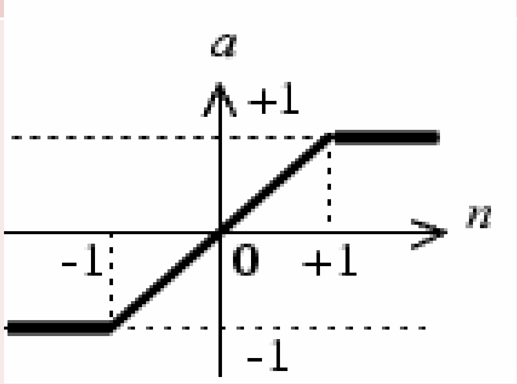
7. Common Activation Functions

Activation Function	Description	Representation
Symmetric hard-limit	$g = \text{Symmetric hard-limit}(v)$ $= \begin{cases} (1) \ldots \text{ if } \ldots v \geq 0 \\ (-1) \ldots \text{ otherwise} \end{cases}$	
Linear	**Applies a linear relation. Neurons of this type are used as linear approximations in linear filters.**	

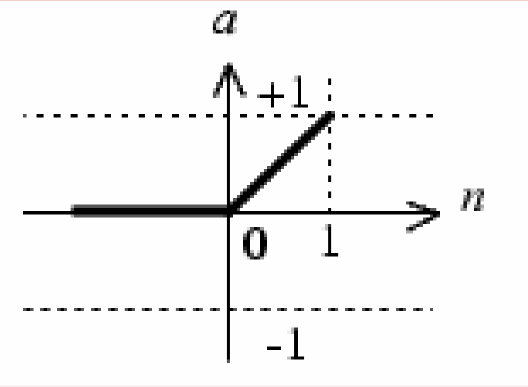
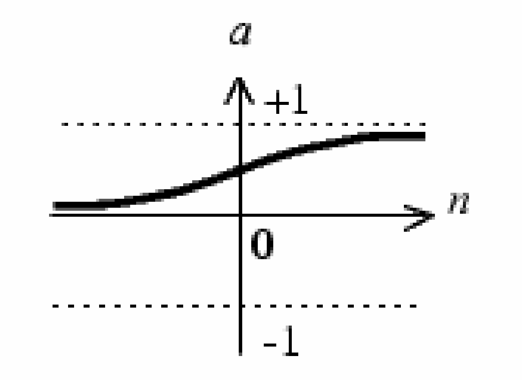
7. Common Activation Functions

Activation Function	Description	Representation
Tangent Sigmoid	Takes an input, which may has any value between plus and minus infinity, and squashes the output into the range -1 to 1.	
Radial Basis	The radial basis function has a maximum of 1 when its input is 0. A radial basis neuron acts as a detector that produces 1 whenever the input p is identical to its weight vector p.	

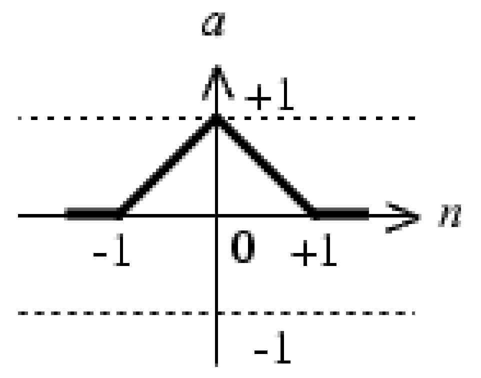
7. Common Activation Functions

Activation Function	Description	Representation
Satlin	Takes an input N, and returns values of N truncated into the interval [0, 1].	
Symmetrical Satlin	Takes an input N, and returns values of N truncated into the interval [-1, 1].	

7. Common Activation Functions

Activation Function	Description	Representation
Positive Linear	**Applies a linear positive relation.**	
Log-sigmoid	**Takes an input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1.**	

7. Common Activation Functions

Activation Function	Description	Representation
Triangular Basis	Takes one input N, and returns each element of N passed through a radial basis function.	

8. The Development of Neural Networks

8.1. McCulloch-Pitts Neuron 1943: the neurons can be arranged into a net to produce any output that can be presented as a **combination of logic functions**. The flow of information through the net assumes a unit time step for signal to travel from one neuron to the next. This time delay allows the net to model some physiological processes, such as the perception of hot and cold. **The idea of a threshold such that if the net input to a neuron is greater than the threshold then the unit fires is one of the feature of McCulloch-Pitts Neuron that is used in many artificial neurons today. McCulloch-Pitts are used most widely as logic circuits** [Anderson & Rosenfeld , 1988]

8. The Development of Neural Networks

8.2. Hebb Learning 1949: Donald Hebb designed the first learning law for artificial neural networks. His premise was that if two neurons were **active simultaneously**, then the **strength** of the connection between them should be **increased**. Refinements were subsequently made to this rather general statement to allow computer simulations [Rochester, Holland, Haibt & Duda, 1956].

The idea is closely related to the **correlation matrix** learning developed by Kohonen & Anderson 1972 among others. An expanded form of Hebb learning [McClelland & Rumelhart, 1988] in which units that are simultaneously off also reinforce the weight on the connection between them will be presented.

8. The Development of Neural Networks

8.3. The 1950s and 1960s : The First Golden Age of Neural Networks

John von Neumann the " father of modern computing" was keenly interested in **modeling the brain** [von Neumann 1958]. Johnson, Brown, Anderson, & Rosenfeld 1988 discuss the interaction between von Neumann and early neural network researchers as McCulloch, and present further indication of von Neumann 's views of the directions in which computers would develop.

- ***Perceptrons*** **:** Frank Rosenblatt (1958,1959,1962), perceptron consists of an input layer connected by paths with fixed weights to associator neurons; the **weights on the connection paths where adjustable**. The perceptron learning rule uses an iterative weight adjustment that is more poweful than Hebb rule. Perceptron learning can be proved to converge to the correct weights if there are weights that will solve the problem at hand.

8. The Development of Neural Networks

- Adaline[Widrow&Hoff, 1960]: the **least mean squares** or **delta rule**, it is closely related to perceptron learning rule. The delta rule adjusts the weights to reduce the difference between the net input to the output unit and the desired output.

- **8.4. The 1970s, 1980s :**

- **Kohonen:** delta with associative memory neural network. His more recent work [Kohonen, 1982] has been the development of **self-organizing** feature maps that use topological structure for the **cluster unit**. These nets have been applied to **speech recognition**, then [Kohonen, 1988] the solution of the " Traveling Salesman Problems" , and **Musical Composition** [Kohonen, 1989].

- **Andesrson:** started his researches in associative memory neural networks. He developed these ideas into his " Brain-State-in-a-Box", which **truncates the linear output** of earlier models to prevent the output from becoming too large as the net iterates to find a stable solution. One of the most important applications of this net is the **medical diagnosis**.

- **Grossberg & Carpenter:** developed a thory of self organizing neural networks called adaptive resonance theory with bineary input patterns and for continuously valued inputs

8. The Development of Neural Networks

8.4. The 1970s, 1980s :

- Backpropagation Parker(1985), LeCun (1986): Able to solve the **mapping problems** and it is a general method to training a multilayer net. This method propagates information about errors at the output units back to the hidden units.
- Neocognitron: A set of specialized neural nets for **character recognition**
- Boltzmann Machine: It is nets in which weights or activations are changed on the basis of **probability density function**. These nets incorporate such classical ideas as simulated annealing and Bayesian decision theory.

The End