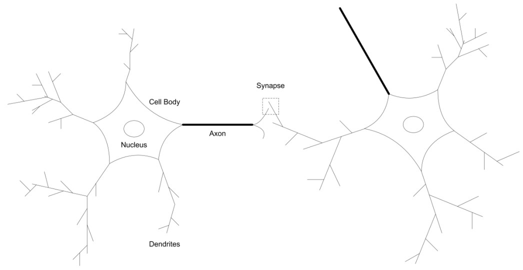
Knowledge Discovery and Data Mining

Unit # 7

Acknowledgement

- Most of the slides in this presentation are taken from course slides provided by
  - Han and Kimber (Data Mining Concepts and Techniques) and
  - Tan, Steinbach and Kumar (Introduction to Data Mining)
  - Several other online sources

Neurons Insider Our Body

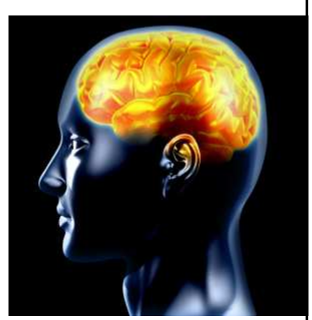


- We are born with about 100 billion neurons
- A neuron may connect to as many as 100,000 other neurons
- Signals "move" via electrochemical signals
- The synapses release a chemical transmitter – the sum of which can cause a threshold to be reached – causing the neuron to "fire"

Sajjad Haider Spring 2010 3

How Our Brain Works?

- While neural networks are modeled after our understanding of the way in which our brain works, surprisingly little is known about how our brains actually function.
- Through various types of inspection, we can see our brain in operation, but because of the massive number of neurons and interconnections between these neurons, how it works remains a mystery (though many theories exist).



Sajjad Haider Spring 2010 4

Artificial Neural Networks

- Neural networks are biologically motivated computing structures that are conceptually modeled after the brain.
- The neural network is made up of a highly connected network of individual computing elements (mimicking neurons) that collectively can be used to solve interesting and difficult problems.

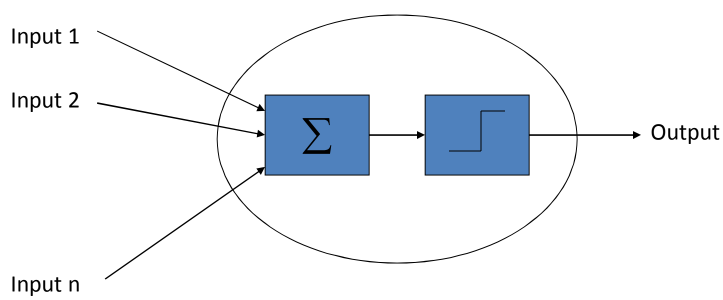
Artificial Neuron

- An artificial neuron is an information-processing unit that is fundamental to the operation of an ANN. It consists of three basic elements:
  - A set of connecting links from different inputs, each of which is characterized by a weight or strength. In general, the weights of an artificial neuron may lie in a range that includes negative as well as positive values.
  - An adder for summing the input signals weighted by the respective synaptic strengths.
  - An activation function for limiting the amplitude of the output of a neuron.

Birth of Artificial Neural Networks

- The story of artificial neural networks is interesting because, like AI itself, it's one of grand visions, eventual disappointment, and finally, silent adoption.
- In 1943, McCulloch and Pitts developed a neural network model based on their understanding of neurology, but the models were typically limited to formal logic simulations (simulating binary operations).

McCulloch-Pitts Neuron

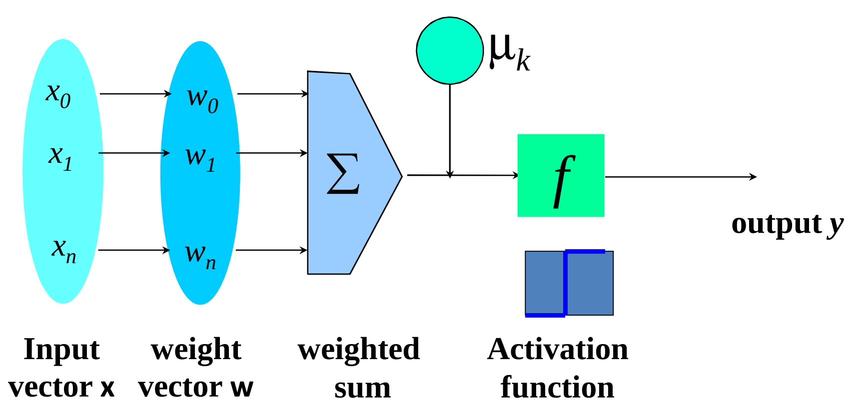


- The basic idea was proposed in 1943.
- A set of synapses (connections) brings in activations from other neurons.
- A processing unit sums the inputs, and then applies a non-linear activation function.
- An output line transmits the results to other neurons.

Bias

- A bias is also commonly applied to each neuron, which is added to the weighted sum of the inputs prior to passing through the transfer function.
- A weight is also commonly applied to the bias.

A Neuron (= a perceptron)

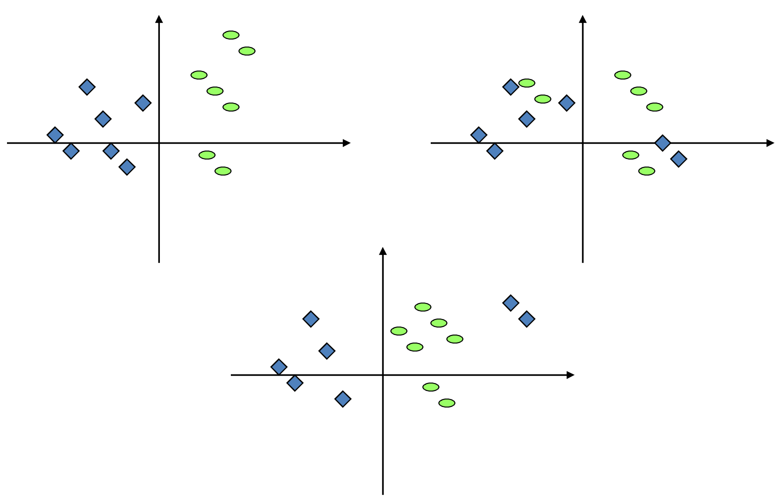


- The *n*-dimensional input vector **x** is mapped into variable y by means of the scalar product and a nonlinear function mapping

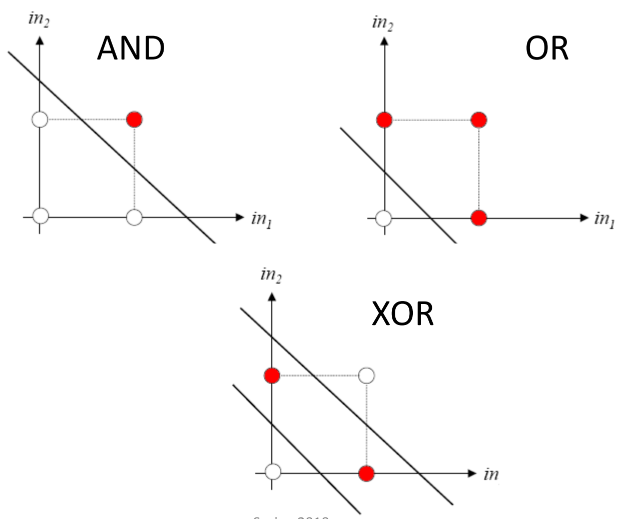
Dark Period

- In 1969, the growing popularity of neural networks was brought to a halt.
- Marvin Minsky and Seymour Papert wrote a book entitled "Perceptrons" in which limitations of single-layer perceptrons were discussed.
- The result was severe reductions in neural network research funding, and a corresponding reduction in the effort applied to the field.

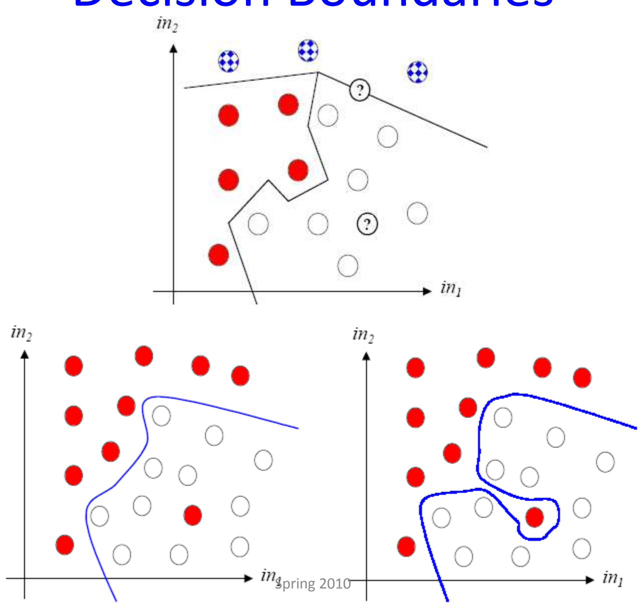
Linearly Separable Data



AND, OR, XOR



Decision Boundaries



Revival

- In 1974, Paul Werbos developed the **backpropagation** algorithm, which permitted successful learning in multilayer neural networks.
- Since the 1970s, research and successful results in neural network design have attracted scientists back to the field.

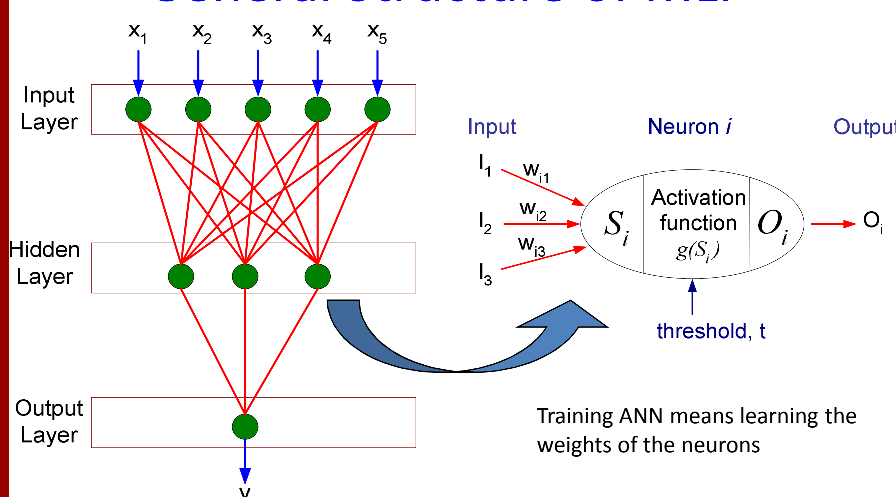
Multilayer Feed-Forward Networks

- Multilayer feed-forward networks are one of the most important and most popular classes of ANNs in real-world applications.
- They are commonly referred to as multilayer perceptrons, which represent a generalization of the simple perceptron.

Characteristics of MLP

- A multilayer perceptron has three distinctive characteristics
  - The model of each neuron in the network includes usually a nonlinear activation function, sigmoid or hyperbolic.
  - The network contains one or more layer of hidden neurons that are not a part of the input or output of the network. These hidden nodes enable the network to learn complex and highly nonlinear tasks by extracting progressively more meaningful features from the input patterns.
  - The network exhibits a high degree of connectivity from one layer to the next one.

General Structure of MLP



Training ANN means learning the weights of the neurons

Specification of ANN

- The number of input attributes found within individual instances determines the number of input layer nodes.
- The user specifies the number of hidden layers as well as the number of nodes within a specific hidden layer.

Input Format

- The input to individual neural network nodes should be numeric and fall in the closed interval range [0,1].
- We need a way to numerically represent categorical data.
  – Attribute Color: {Red, Green, Blue, Yellow}
- We also need a conversion method for numerical data falling outside the [0,1] range.
  – Values: 100, 200, 300, 400

Architecture of NN?

- How many neurons are required in the input layer?

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Architecture of NN?

- How many neurons are required in the input layer?

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Output Format

- The nodes of the input layer pass input attribute values to the hidden layer unchanged.
- A hidden or output layer node takes input from the connected nodes of the previous layer, combines the previous layer node values into a single value, and uses the new value as input to an evaluation function.
- The output of the evaluation function is a number in the closed interval [0, 1].

Sigmoid Function

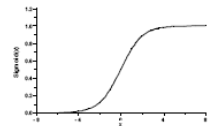
- The first criterion of an evaluation function is that the function must output values in the [0, 1] interval range.
- A second criterion is that the function should output a value close to 1 when sufficiently excited.
- The sigmoid function meets both criterion and is often used for node evaluation.
  - $f(x) = 1 / (1 + e^{-x})$

Transfer Functions

Sigmoid Functions These are smooth (differentiable) and monotonically increasing.

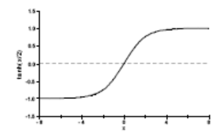
The logistic function

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$



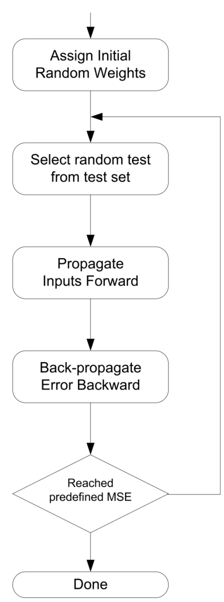
Hyperbolic tangent

$$\tanh\left(\tfrac{x}{2}\right) = \frac{1-e^{-x}}{1+e^{-x}}$$



Working of ANN

- Learning is accomplished by modifying network connection weights while a set of input instances is repeatedly passed through the network.
- Once trained, an unknown instance passing through the network is classified according to the value(s) seen at the output layer.



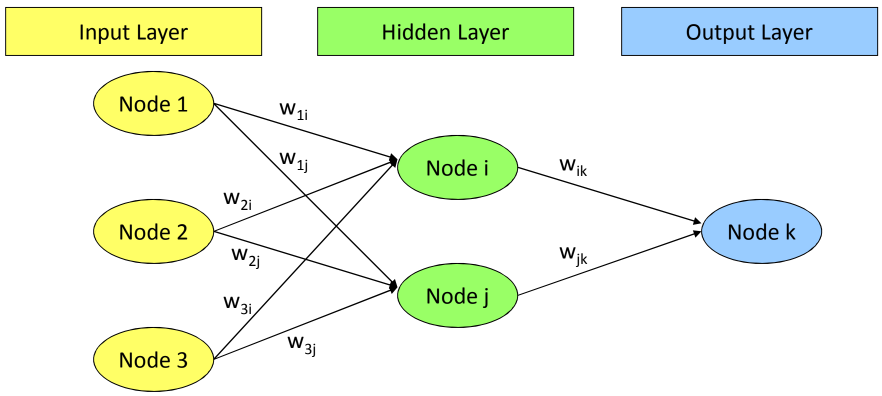
How A Multi-Layer Neural Network Works?

- The **inputs** to the network correspond to the attributes measured for each training tuple.
- Inputs are fed simultaneously into the units making up the **input layer.**
- They are then weighted and fed simultaneously to a **hidden layer.** The number of hidden layers is arbitrary, although usually only one
- The weighted outputs of the last hidden layer are input to units making up the **output layer**, which emits the network's prediction
- From a statistical point of view, networks perform **nonlinear regression**: Given enough hidden units and enough training samples, they can closely approximate any function.

Backpropagation

- Iteratively process a set of training tuples and compare the network's prediction with the actual known target value.
- For each training tuple, the weights are modified to **minimize the mean squared error** between the network's prediction and the actual target value
- Modifications are made in the "**backward**" direction: from the output layer, through each hidden layer down to the first hidden layer, hence "**backpropagation**".

A Fully Connected Feed-Forward Network



Explanation of the Backpropagation Algorithm

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50, T=0.65

- Input = {1.0, 0.4, 0.7}
- Input to node i = 0.2x1.0 + 0.3x0.4 - 0.1x0.7 = 0.25
- Now apply the sigmoid function: f(0.25) = 0.562
- Input to node j = ?
- Input to node k = ?
- Error(k) = $(T - O_k)\, O_k\, (1 - O_k)$
  - T = the target output
  - O_k = the computed output at node k
- Error(k) = ?

Explanation of the Backpropagation Algorithm

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50

- Error(i) = Error(k) w_{ik} O_i $(1 – O_i)$
  = ?
- Error(j) = ?
- The next step is to update the weights associated with the individual node connections.
- Weight adjustments are made using the delta rule
  - To minimize the sum of the square errors, where error is defined as the distance between computed and actual output

Explanation of the Backpropagation Algorithm

w_{1i}=0.20, w_{1j}=0.10, w_{2i}=0.30, w_{2j}=-0.10, w_{3i}=-0.10, w_{3j}=0.20, w_{ik}=0.10, w_{jk}=0.50

- $w_{ik} = w_{ik}$ (current) + Δw_{ik}
- Δw_{ik} = r x Error(k) x O_i
  - where r is learning rate parameter, $0 < r < 1$
- Compute: Δw_{ik} Δw_{1i} Δw_{2i} Δw_{3i}

Algorithm

- Initialize the network:
  - Create the network topology by choosing the number of nodes for the input, hidden, and output layers.
  - Initialize weights for all node connections to arbitrary values between -1.0 and 1.0.
  - Choose a value between 0 and 1 for the learning parameter.
  - Choose a terminating condition.
- For all the training instances:
  - Feed the training instance through the network.
  - Determine the output error.
  - Updated the network weights.
- If the terminating condition has not been met, repeat step 2.
- Test the accuracy of the network on a test dataset. If the accuracy is less than optimal, change one or more parameters of the network topology and start over.

Neural Network as a Classifier

- 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
  - Algorithms are inherently parallel
  - Techniques have recently been developed for the extraction of rules from trained neural networks

General Considerations

- What input attributes will be used to build the network?
- How will the network output be represented?
- How many hidden layers should the network contain?
- How many nodes should there be in each hidden layer?
- What conditions will terminate network training?

Design Issues in ANN Learning

- The number of nodes in the input layer should be determined. Assign an input node to each numerical or binary input variable. If the input variable is categorical, we could either create one node for each caterogical value or encode k-ary variable using $[\log_2 k]$ input node.
- The number of nodes in the output layer should be established. For a two-class problem, it is sufficient to use a single output node. For a k-class problem, there are k output nodes.

Design Issues in ANN Learning (Cont'd)

- The network topology must be selected. Finding the right topology is not an easy task. One way to do this is to start from a fully connected network with a sufficiently large number of nodes and hidden layers, and then repeat the model building procedure with a smaller number of nodes.
- The weights and biases need to be initialized. Random assignments are usually acceptable.
- Training examples with missing values should be removed or replaced with most likely values.

Weakness

- The biggest criticism of neural networks is that they lack the ability to explain their behavior.
- The algorithm is not guaranteed to converge to an optimal solution.
  - Manipulation of various learning parameter
- Neural networks can easily be over trained to the point of working well on the training data but poorly on test data.
  - Division of data into training and testing sets.