Introduction to Data Mining Using Artificial Neural Networks

ISM 611

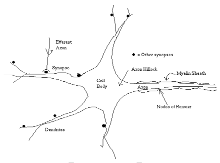
Dr. Hamid Nemati

SEMINAR OUTLINE

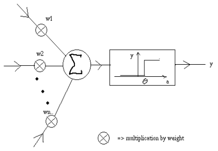
- Introduction to and Characteristics of Neural Networks
- Comparison of Neural Networks to traditional approaches to problem solving
- History of Neural Networks
- Model of a neuron
- Neural Networks
- How to develop Data mining Applications using Neural Networks
- Demo of PRW Artificial Neural Network Package

Artificial Neural Networks

- Is the type of information processing whose architecture is inspired by the structure of biological neural systems.



Biological



Artificial

Definitions of Neural Networks

- ... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

According to the DARPA Neural Network Study (1988, AFCEA International Press, p. 60):

Definitions of Neural Networks

- A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store the knowledge.

According to Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan, p. 2:

Definitions of Neural Networks

- A neural network is a circuit composed of a very large number of simple processing elements that are neurallybased. Each element operates only on local information. Furthermore each element operates asynchronously; thus there is no overall system clock.
  - (According to Nigrin, A. (1993), Neural Networks for Pattern Recognition, Cambridge, MA: The MIT Press, p. 11)

- Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge.
  - (According to Zurada, J.M. (1992), Introduction To Artificial Neural Systems, Boston: PWS Publishing Company, p. xv)

Artificial Neural Networks

Based on the following assumptions:

1. Information processing occurs at many simple processing elements called **neurons.**
2. Signals are passed between neurons over interconnection links.
3. Each interconnection link has an associated weight.
4. Each neuron applies an activation function to determine its output signal.

Artificial Neural Networks

The Operation of a Neural Network is controlled by three properties:

1. *The pattern of its interconnections,* ***architecture.***
2. *Method of determining and updating the weights on the interconnections,* ***training.***
3. *The function that determines the output of each individual neuron,* ***activation or transfer function.***

Characteristics of Neural Networks

1. Neural Network is composed of a large number of very simple processing elements called neurons.
2. Each neuron is connected to other neurons by means of interconnections or links with an associated weight.
3. Memories are stored or represented in a neural network in the pattern of interconnection strengths among the neurones.
4. Information is processed by changing the strengths of interconnections and/or changing the state of each neurones.
5. A neural network is trained rather than programmed.
6. A neural network acts as an associative memory. It stores information by associating it with other information in the memory. For example, a thesaurus is an associative memory.

Characteristics of Neural Networks

7. It can generalize; that is, it can detect similarities between new patterns and previously stored patterns. A neural network can learn the characteristics of a general category of objects on a series of specific examples from that category.
8. It is robust, the performance of a neural network does not degrade appreciably if some of its neurones or interconnections are lost. (distributed memory)
9. Neural networks may be able to recall information based on incomplete or noisy or partially incorrect inputs.
10. A neural network can be self-organizing. Some neural networks can be made to generalize from data patterns used in training without being provided with specific instructions on exactly what to learn.

Model of a Neuron

- A neuron has three basic elements:
  1. A set of *synapses or connecting links*, each with a weight or strength of it own. A positive weight is *excitatory* and a negative weight is *inhibitory*.
  2. An *adder* for summing the input signals.
  3. An *activation function* for limiting the range of output signals, usually between [-1, +1] or [0, 1].
- Some neurons may also include:
  - a ***threshold*** to lower the net input of the activation function.
  - a ***bias*** to increase the net input of the activation function

Traditional Approaches to Information Processing Vs. Neural Networks

1. ***Foundation: Logic vs. Brain:***
   - **TA**:Simulate and formalize human reasoning and logic process. TA treat the brain as a black box. TA focus on how the elements are related to each other and how to give the machine the same capabilities.
   - **NN**:Simulate the intelligence functions of the brain. NN focus on modeling the brain structure. NN attempts to create a system that functions like the brain because it has a structure similar to the structure of the brain.

Traditional Approaches to Information Processing Vs. Neural Networks

2. ***Processing Techniques: Sequential vs Parallel***
   – **TA:**The processing method of TA is inherently sequential.
   – **NN:**The processing method of NN is inherently parallel. Each neuron in a neural network system functions in parallel with others.

Traditional Approaches to Information Processing Vs. Neural Networks

3. Learning: Static and External vs. Dynamic and Internal
   – ***TA:***Learning takes place outside of the system. The knowledge is obtained outside the system and then coded into the system.
   – **NN**:Learning is an integral part of the system and its design. Knowledge is stored as the strength of the connections among the neurons and it is the job of NN to learn these weights from a data set presented to it.

Traditional Approaches to Information Processing Vs. Neural Networks

4. ***Reasoning Method: Deductive vs. Inductive***
   – **TA:** Is deductive in nature. The use of the system involves a deductive reasoning process, applying the generalized knowledge to a given case.
   – **NN**: Is inductive in nature. It constructs an internal knowledge base from the data presented to it. It generalizes from the data, such that when it is presented a new set of data, it can make a decision based on the generalized internal knowledge.

Traditional Approaches to Information Processing Vs. Neural Networks

5. ***Knowledge Representation: Explicit vs. Implicit***
   – **TA**:It represents knowledge in an explicit form. Rules and relationships can be inspected and altered.
   – **NN:**The knowledge is stored in the form of interconnections strengths among neurons. Nowhere in the system, one can pick up a piece of computer code or a numerical value as a discernible piece of knowledge.

Strengths of Neural Networks

- Generalization
- Self-organization
- Can recall information based on incomplete or noisy or partially incorrect inputs.
- Inadequate or volatile knowledge base
- Project development time is short and training time for the neural network is reasonable.
- Performs well in **data-intensive** applications
- Performs well where:
  - Standard technology is inadequate
  - Qualitative or complex quantitative reasoning is required
  - Data is intrinsically noisy and error-prone

Limitations of Neural Networks

- No explanation capabilities
- Still a “blackbox” approach to problem solving
- No established development guidelines
- Not appropriate for all types of problems

Applications of Neural Networks

- Economic Modeling
- Mortgage Application Assessments
- Sales lead assessments
- Disease Diagnosis
- Manufacturing Quality Control
- Sports forecasting
- Process Fault detection
- Bond Rating
- Credit Card Fraud
- Detection
- Oil Refinery Production Forecasting
- Foreign Exchange Analysis
- Market and Customer Behavior Analysis
- Optimal resource Allocation
- Financial Investment Analysis
- Optical Character Recognition
- Optimzation

NEURAL NETWORKS IN FINANCIAL MARKET ANALYSIS

- Karl Bergerson of Seattle’s Neural Trading Co. developed Neural$ Trading System using Brainmaker software.
- He used 9 years of financial data including price, and volume, to train his neural net.
- He ran it against a $10,000 investment. After 2 years, the financial account had grown to $76,034 - a 660% growth.
- When tested with new data his system was 89% accurate.

AI-Expert May 1997.

NEURAL NETWORKS IN BANKRUPTCY PREDICTION

- NeuralWare’s Application Development Services and Support (ADSS) group developed a backpropagation neural network using NeuralWorks Professional software to predict bankruptcies.
- The network had 11 inputs, one hidden layer and one output.
- The network was trained using a set of 1000 examples. The network was able to predict failed banks wih an accuracy of 99 percent.
- ADSS has also developed a neural network for a credit card company to identify those credit card holders who have 100% probability of going bankrupt.

Al Review, Summer 1996

NEURAL NETWORKS IN FORECASTING

- Pugh & Co. of NJ has used Brainmaker to forecast the next year’s corporate bond ratings for 115 companies.
- The Brainmaker network takes 23 financial-analysis factors and the previous year’s index rating as inputs and output the rating forecast for next year.
- Pugh & Co. claims that the network is 100% accurate in categorizing ratings among categories and 95% accurate among subcategories.

NEURAL NETWORKS IN QUALITY CONTROL:

- CTS Electronics of TX has used a NeuroShell neural network software package for classification of loudspeaker defects in its Mexico assembly line.
- 10 input nodes represent distortions and 4 output nodes represent speaker-defect classes.
- Networks require only a 40 minute training period.
- The efficiency of the network has exceeded CT’s original specifications.

History Of Neural Networks

- In 1943, McCulloch and Pitts, proposed the first model of an artificial neuron.
- In 1949, Donald Hebb in his book: *"The Organization of Behavior"*, suggested that the strength of connections among neurons in brain changes dynamically. He suggested a physiological learning rule for the synaptic modification.
- In 1954, Minsky wrote the first “neural network” dissertation at Princeton University titled: “*Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem.*”

History Of Neural Networks

- In 1958, Frank Rosenblatt, used McCulloch and Pitts model and Hebb's proposition to developed Perceptrons, the first neural network architecture that was capable of learning.
- In 1960, Widrow and Hoff introduced the Least Mean Square (LMS) algorithm and formulated ADALINE (adaptive linear neuron) used Delta rule to train the network.
- In 1943, McCulloch and Pitts, proposed the first model of an artificial neuron.

History Of Neural Networks

- In 1969, Minsky and Papert published Perceptrons, which mathematically proved the limitations of Perceptrons. They suggested that neural network can not be used to represent even some simple systems. Their book was so influential that neural network research was brought to a stand-still for over a decade.
- In early 1980's, the works of Hopfield, Merr, Kohonen and others rekindled the interest in neural network research.
- In 1987, Robert Hecht-Nelson used Kolmogorov's theorem to mathematically disprove Minsky and Papert's conjectures. He showed that any real mapping can be exactly implemented by a 3-layer neural network.

Neural Network Architecture

- In a Neural Network, neurons are grouped into layers or slabs.
- The neurons in each layer are the same type.
- There are different types of ***Layers.***
- The ***Input layer***, consists of neurons that receive input from the external environment.
- The ***Output layer,*** consists of neurons that communicate to the user or external environment.
- The ***Hidden layer***, consists of neurons that ONLY communicate with other layers of the network.

Input to Neurons

- A neuron receives inputs from other neurons.
- It does not matter whether the input to neuron comes form the neurons on the same layer or on another layer.
- The message that a neuron receives form another neuron is modified by the strength of the connection between the two.
- The net input to a neuron is the sum of all the messages it receives from all neurons it is connected to.
- There may also be ***External input*** to a neuron.

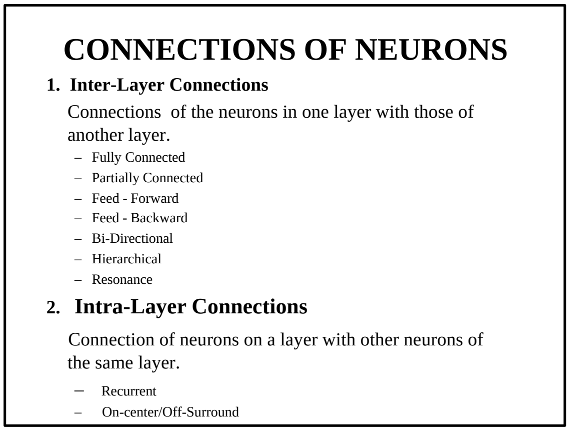
Input to Neurons

- The net input to a neuron creates an action potential.
- When the action potential reaches a given level, the neuron fires and sends out a message to the other neurons.

$$input_i = \sum_{all\ j\ neurons\ connected\ with\ i} w_{ij} \ * \ output_j$$

Output From Neurons

- For a neuron to send out output to the other neurons, the action potential or the net input to the neuron should go through a filter or transformation.
- This filter is called "***Activation Function***" or "***Transfer Function***".
- There are number of different activation functions:
  - Step Function
  - Signum Function
  - Sigmoidal Function
  - Hyperbolic Tangent Function
  - Linear Function
  - Threshold-linear Function



CONNECTIONS OF NEURONS

1. **Inter-Layer Connections**

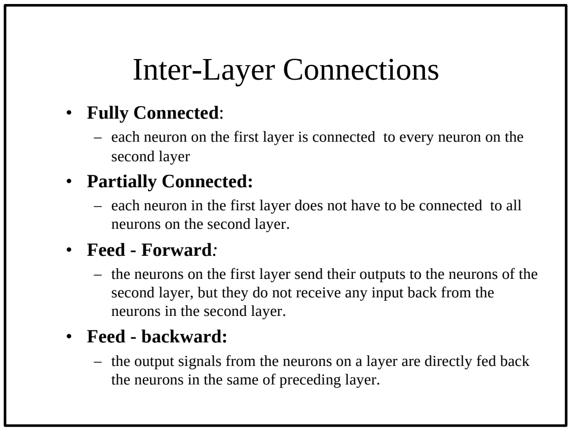
   Connections of the neurons in one layer with those of another layer.

   – Fully Connected
   – Partially Connected
   – Feed - Forward
   – Feed - Backward
   – Bi-Directional
   – Hierarchical
   – Resonance

2. **Intra-Layer Connections**

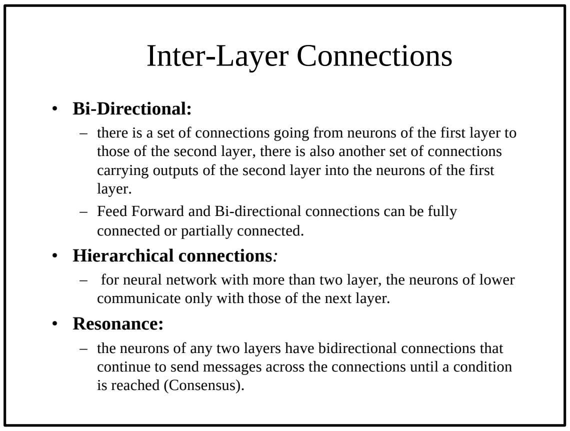
   Connection of neurons on a layer with other neurons of the same layer.

   – Recurrent
   – On-center/Off-Surround



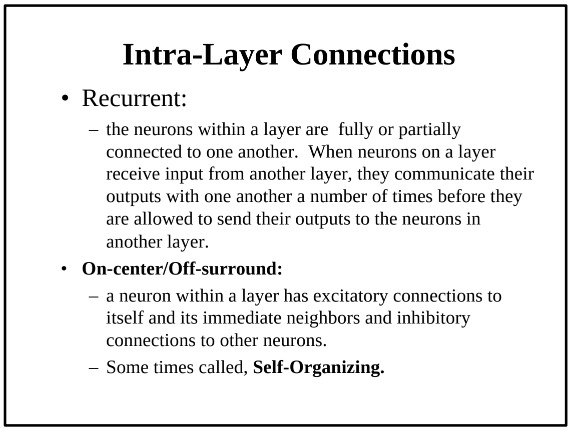
Inter-Layer Connections

- **Fully Connected**:
  – each neuron on the first layer is connected to every neuron on the second layer
- **Partially Connected:**
  – each neuron in the first layer does not have to be connected to all neurons on the second layer.
- **Feed - Forward***:*
  – the neurons on the first layer send their outputs to the neurons of the second layer, but they do not receive any input back from the neurons in the second layer.
- **Feed - backward:**
  – the output signals from the neurons on a layer are directly fed back the neurons in the same of preceding layer.



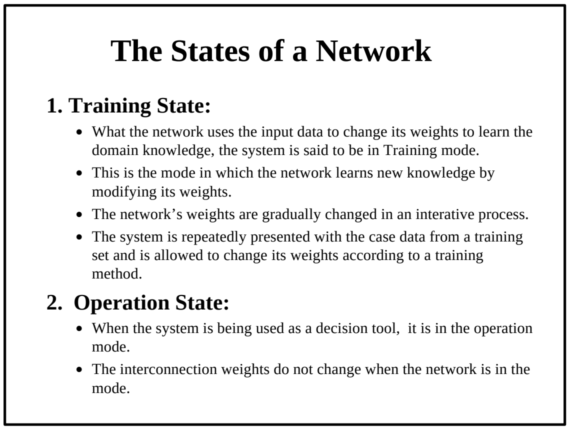
Inter-Layer Connections

- **Bi-Directional:**
  – there is a set of connections going from neurons of the first layer to those of the second layer, there is also another set of connections carrying outputs of the second layer into the neurons of the first layer.
  – Feed Forward and Bi-directional connections can be fully connected or partially connected.
- **Hierarchical connections***:*
  – for neural network with more than two layer, the neurons of lower communicate only with those of the next layer.
- **Resonance:**
  – the neurons of any two layers have bidirectional connections that continue to send messages across the connections until a condition is reached (Consensus).



Intra-Layer Connections

- Recurrent:
  – the neurons within a layer are fully or partially connected to one another. When neurons on a layer receive input from another layer, they communicate their outputs with one another a number of times before they are allowed to send their outputs to the neurons in another layer.
- **On-center/Off-surround:**
  – a neuron within a layer has excitatory connections to itself and its immediate neighbors and inhibitory connections to other neurons.
  – Some times called, **Self-Organizing.**



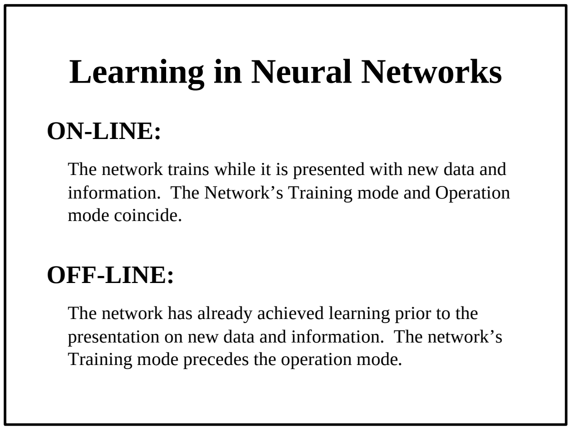
The States of a Network

1. Training State:

- What the network uses the input data to change its weights to learn the domain knowledge, the system is said to be in Training mode.
- This is the mode in which the network learns new knowledge by modifying its weights.
- The network's weights are gradually changed in an interative process.
- The system is repeatedly presented with the case data from a training set and is allowed to change its weights according to a training method.

2. Operation State:

- When the system is being used as a decision tool, it is in the operation mode.
- The interconnection weights do not change when the network is in the mode.



Learning in Neural Networks

ON-LINE:

The network trains while it is presented with new data and information. The Network's Training mode and Operation mode coincide.

OFF-LINE:

The network has already achieved learning prior to the presentation on new data and information. The network's Training mode precedes the operation mode.

Learning in Neural Networks

- The process neural network uses to compute the interconnection weights among neurons.
- For the neural networks to learn, it should be given a learning method to change the weights to the ideal values.
- Learning methods is the most important distinguishing factor in various neural networks.
- There are two distinct types of learning:
  - **Supervised Learning**
  - ***Unsupervised Learning***

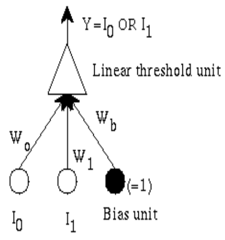
Supervised Learning

- In **supervised learning**,
  - the system developer tells the network what the correct answer is and the network determine the weights in such a way that once given the input, it would determine the output.
  - The idea in supervised learning is that the network is repeatedly given facts about the various cases, along with the expected outputs.
  - The network uses the learning method to adjust the weights in order to produce the outputs close to what is expected.

Unsupervised learning,

- The network receives only the inputs and no information about on the expected output.
- In these networks, the systems learns to produce the pattern of what it has been exposed to.
- These networks are sometimes referred to as Self Organizing Networks.

Example of A simple single unit adaptive network



The network has 2 inputs, and one output.
All are binary. The output is

0 if $W_0 * I_0 + W_1 * I_1 + W_b < 1.5$
1 if $W_0 * I_0 + W_1 * I_1 + W_b >= 1.5$

We want it to learn simple OR: Set
$W_0 = .5$,
$W_1 = .5$,
$W_b = 1$

Types of Learning Rules

Mathematical equation that governs the change in the weights in successive iterations.

- *Hebb Rule*
- *Delta Rule*
- *Generalized Delta Rule*
- *Kohonen Rule*

Hebb Rule

- Was introduced by Donald Hebb, a Canadian psychologist.
- There a number of different interpretations of Hebb rule:

  $W_{ij}(new) - W_{ij}(old) = a * Output_i * Output_j$

  $W_{ij}(new) - W_{ij}(old) = a * Output_i * Input_j$

  $W_{ij}(new) - W_{ij}(old) = a * Output_i * DesOutput_j$

- The higher the value of a , the faster the network learns, but will have a lower capability to generalize.
- In some implementations, the value of a is allowed to change as the network learns.

DELTA RULE

- It is sometime known as the Least Mean Square Rule (LMS).
- It is similar to the Steepest Descent method where the change in a weight is proportional to the negative of the gradient of the error surface with respect to that particular weight.
- The Delta rule minimizes the sum of the squared errors, where error is the defined to be the distance between an actual output of the system and the desired output, for a given input data.

DELTA RULE

$W_{ij\ (new)} - W_{ij\ (old)} = a \quad {}^{*}Output_{i} \ {}^{*}\ Error_{j}$

Where

$Error_{j} = Desired\ Output_{j} - Output_{j}\ (old)$

- The generalized Delta Rule, the derivative of the transfer function is added to the equation. The derivative to the transfer function is evaluated at input$_j$
- There are a number of variations to Delta rule that are intended to reduce training time and reduce the problem of trapping in local minimum.
- A momentum function may be added to the delta rule formula.
- Use Genetic algorithms, simulated annealing, Tabu search are some of the other methods to overcome this problem.

KOHONEN RULE

- Is an example of Unsupervised learning Rule.
- It does not require prior knowledge about the output.
- It is self organizing.
- The training can be done on-line.
- Is good for clustering applications where there is no prior knowledge of the out comes.
- It uses the following equation to modify the weights:

$W_{ij\ (new)} - W_{ij\ (old)} = a \quad {}^{*}(extinput_{i} - Wij\ _{(old)})$

Classifications Neural Network Architectures

Neural Networks can be classified according to their architecture with respect to:

1. *Corresponding of input and output data*
2. *Number of layers*
3. *Certainty of firing*
4. *Types of connectivity*
5. *Temporal features*

Correspondence of input and output data

- One important feature of the brain is its associative memory
- Neural Networks designed to be associative memory devices, they produce an associative output for any given input.
- For example, for a given set of attributes of a loan applicant, the network associate a decision about the loan.

Correspondence of input and output data

The associative nature of the networks can be classified into two groups.

- **Auto-Associative** *networks:*
  - In these networks, the input vector is the same as the output vector. They are useful for recognizing patterns from incomplete and erroneous data, example is to identify an airplane using its radar fingerprint, or to recognize a person from an old or faded picture.
- **Hetero-Associative** *networks:*
  - The input set and the output sets are not of the same type. In the loan application example, the input set is the attributes of the applicants and the output set is the decision for the loan.

NUMBER OF LAYERS

- ***Two Layer***
- ***Multi-Layer***

Certainty of Firing

Neural Networks can be grouped according to the certainty of the firing of their neurons.

- **Deterministic neural networks:**
  - Have a deterministic activitation function. Once a neuron reaches a certain level of activation dictated by the transfer function, it fires and sends out impulses to other neurons.
- **Stochastic neural networks:**
  - The firing of the neuron is not certain, but takes places according to a probability distribution. Example of these networks is Boltzmann machine network.

Types of Connectivity

- **Cross-bar connections:**
  - networks with this type of connections allow feed back among layers.
- **Feed-back/Feed-forward connections:**
  - networks with this type of connections allow the weights in one layer to change in response to an output from other layers.
- **Recurrent connections:**
  - networks with this type of connections allow feedback connections among neurons. A network with recurrent connections exhibits dynamical behavior. Given an initial state, the network evolves as time elapses. If the network is stable, a state equilibrium can eventually be reached.

Two Layer Neural Networks

- We define a Two-Layer neural network as a network in which the interconnections between only two layers of the network changes in the learning process. The network may have additional layers but only the one set of the inter-layer connections can be trained.
- The Two layer networks may include:
  - Preceptron
  - **ADALINE** and **MADALINE**
  - Kohonen’s Self-organizing Network
  - Hopfield
  - Brain-state-in-box
  - Instar and outstar

TEMPORAL FEATURES

Static:

- There is no element of time.
- The network accepts inputs one at a time, and produces output without any connection to the next input.
- The loan application example is such a network.

TEMPORAL FEATURES

DYNAMIC:

- A dynamic network goes through a series of inputs through time and generate output accordingly.
- This type of network is usually called a Spatio-temporal network.
- A network receives a series of patterns and recognizes that they represent a dynamic event.
- An aircraft recognition neural network is an example of such a system.

Multi-Layer Networks

- We define a Multi-Layer neural network as a network in which the interconnections between more than the two I/O layers of the network change in the learning process. The network has additional layers, hidden from the environment.
- The Multi-Layer networks may include
  - – Backpropagation Networks
  - – Counterpropagation Networks
  - – ART Networks
  - – Hopfield Networks
  - – BAM Networks

BACKPROPAGATION NETWORKS

- Were introduced by Rumelhart and Hinton in 1986.
- It is the most widely used network today.
- The basic idea in back propagation network is that the neurons of lower layer send up their outputs to the next layer.
- It is a fully connected, feed forward, hierarchical multi layer network, with hidden layers and no intra-connections.
- The weights between output layer and the hidden layer are updated based on the amount of error at the output layer, using generalized delta rule.
- The error of the output layer propagates back to the hidden layer via the backward connections.

BACKPROPAGATION NETWORKS

- The connection weights between the input layer and the hidden layer are updates based on the generalized delta rule.
- There is a forward flow of outputs and a backward update of the weights based on the error on the output layer.
- During the training process, the network seeks weights that minimize the sum of the square of output layer errors.
- There is a backward connection from the output layer to the hidden layer, from the hidden layer to the next lower hidden layer and from the lowest hidden layer to the input layer.

BACKPROPAGATION NETWORKS

- The backward connections between the each pair of layers have the same weight values as those of the forward connections and are updated together.
- Backpropagation Networks major limitation is the trap of local minimum. the network's trained weights may not be the ones that necessarily produce the best answer.
- Counterpropagation Networks is a generalization of the backpropagation networks that overcome this limitation.
- This network has five layers and receives external input data from both top and bottom ends of the network.

BACKPROPAGATION NETWORKS

- The output of the neurons move in both directions and the final output of the network is the pattern that matched a combination of the two inputs to the systems at both ends.
- Recurrent Backpropagation networks include element of time into the systems.
- It receives a series of input patterns in an interval of time, and produces the corresponding series of output patterns in the same interval.
- For example, the network may receive loan payment patterns of a person for six month and may recognize that the pattern matches a person who is going to default on the loan.

NEURAL NETWORKS TRAINING

- To **train** a neural network, the developer repeatedly present the network with data.
- During the training the network is allowed to change its interconnections weights using one of the learning rules.
- The training continues until a pre-specified condition is reached.
- The design and training a neural network is an ITERATIVE process. The developer goes through the designing system, training the systems and evaluating the system repeatedly .

Neural Networks Training

- The issue of training the neural networks can be divided into categories:

Training data set

Training strategies

Developing Neural Network Applications for Data Mining

1. Is Neural Network approach appropriate?
2. Select appropriate Paradigm.
3. Select input data and facts.
4. Prepare data.
5. Train and test network.
6. Run the network.
7. Use the Network for Data Mining

Is Neural Network Approach Appropriate?

- Inadequate knowledge base
- Volatile knowledge bases
- Data-intensive system
- Standard technology is inadequate
- Qualitative or complex quantitative reasoning is required
- Data is intrinsically noisy and error-prone
- Project development time is short and training time for the neural network is reasonable.

Select appropriate paradigm:

- Decide on network archiecture according to general problem area (e.g., Classification, filtering, pattern recognition, optimization, data compression, prediction)
- Decide on transfer function
- Decide on learning method
- Select network size. (e.g., How many inputs, output neurons? How many hidden layers and how many neurons per layer?)
- Decide on nature of input/output.
- Decide on type of training used.

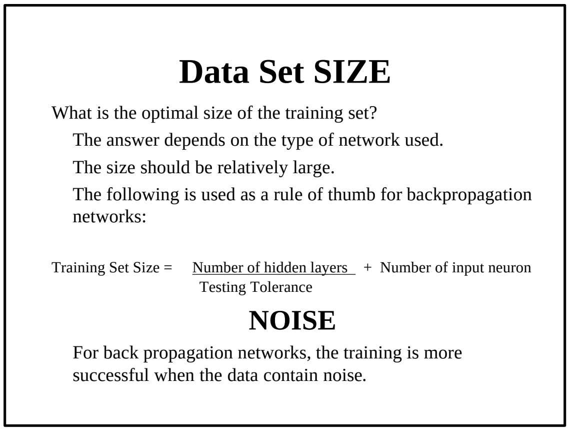
Select Input Data And Facts

- What is the problem domain?
- The training set should contain a good representation of the entire universe of the domain.
- What are the input sources.
- What is the optimal size of the training set?
  - The answer depends on the type of network used.
  - The size should be relatively large.
  - The use a rule of thumb
- For backpropagation networks, the training is more successful when the data contain noise.

Data Set Considerations

In selecting a data set, the following issues should be considered:

- *Size*
- *Noise*
- *Knowledge domain representation*
- *Training set and test set*
- *Insufficient data*
- *Coding the input data*



Data Set SIZE

What is the optimal size of the training set?

The answer depends on the type of network used.

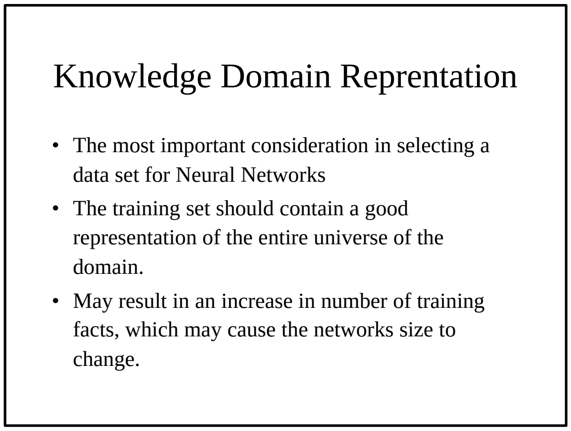
The size should be relatively large.

The following is used as a rule of thumb for backpropagation networks:

$$\text{Training Set Size} = \frac{\text{Number of hidden layers}}{\text{Testing Tolerance}} + \text{Number of input neuron}$$

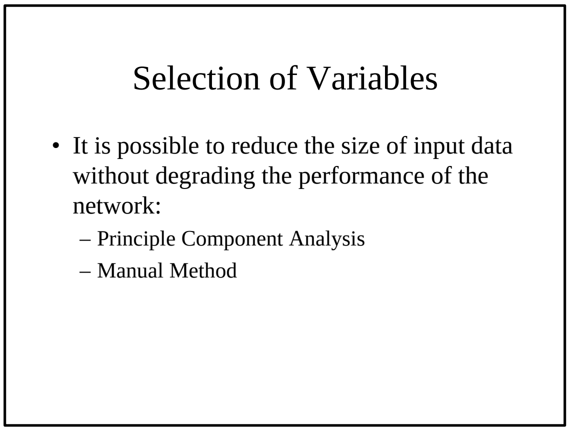
NOISE

For back propagation networks, the training is more successful when the data contain noise.



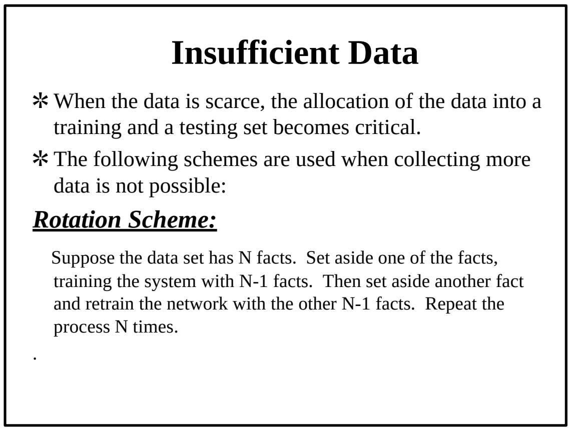
Knowledge Domain Reprentation

- The most important consideration in selecting a data set for Neural Networks
- The training set should contain a good representation of the entire universe of the domain.
- May result in an increase in number of training facts, which may cause the networks size to change.



Selection of Variables

- It is possible to reduce the size of input data without degrading the performance of the network:
  - Principle Component Analysis
  - Manual Method



Insufficient Data

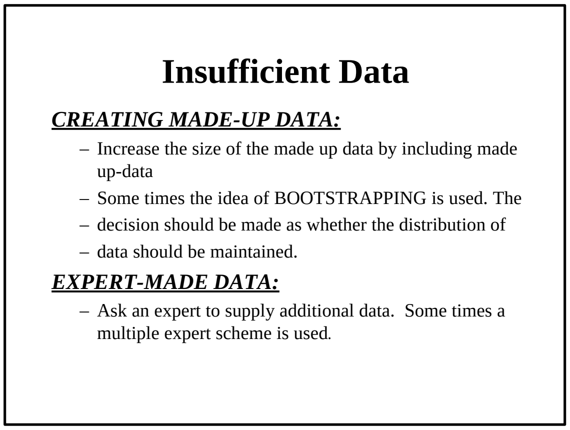
✻ When the data is scarce, the allocation of the data into a training and a testing set becomes critical.

✻ The following schemes are used when collecting more data is not possible:

<u>Rotation Scheme:</u>

Suppose the data set has N facts. Set aside one of the facts, training the system with N-1 facts. Then set aside another fact and retrain the network with the other N-1 facts. Repeat the process N times.

.



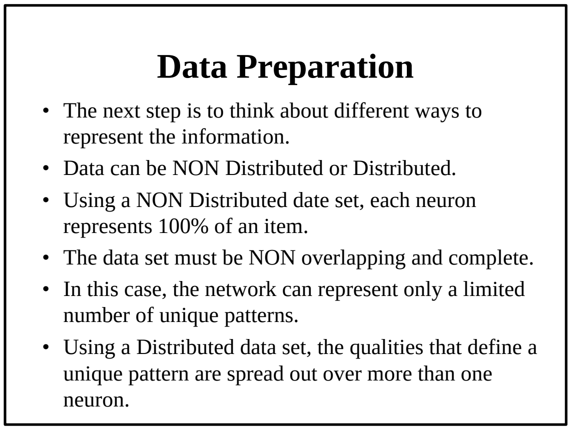
Insufficient Data

<u>CREATING MADE-UP DATA:</u>

- Increase the size of the made up data by including made up-data
- Some times the idea of BOOTSTRAPPING is used. The
- decision should be made as whether the distribution of
- data should be maintained.

<u>EXPERT-MADE DATA:</u>

- Ask an expert to supply additional data. Some times a multiple expert scheme is used.



Data Preparation

- The next step is to think about different ways to represent the information.
- Data can be NON Distributed or Distributed.
- Using a NON Distributed date set, each neuron represents 100% of an item.
- The data set must be NON overlapping and complete.
- In this case, the network can represent only a limited number of unique patterns.
- Using a Distributed data set, the qualities that define a unique pattern are spread out over more than one neuron.

Data Preparation

- For example, a purple object can be represented described as being half red and half blue.
- The two neurons assigned to red and blue can together define purple, eliminating the need to assign a third purple neuron.
- The problem with the Distributed system is that the input/output should be translated twice.
- The advantage of these networks is that fewer neurons are required to define the domain.

Continuous vs. Binary Data

- The developer should decide as whether a piece data is continuous of binary.
- If a continuous data is represented in a binary form, the network may NOT be able to train properly.
- The decision as whether a piece data is continuous of binary may not be simple.
- If the continuous data set is spread evenly within a data range, it may be reasonable to represent it as binary.

Actual Values Vs. Change In Values

- An important decision in representing continuous data is whether to use actual amounts or changes in amounts.
- Whenever possible, it is better to use changes in values.
- Using the changes in values may make it easier for the network to appreciate the meaning that the data represents.

Coding the Input Data

The training data set should be properly normalized.

The training data set should match the design of the network.

- Zero-mean-unit Variant (Zscore)
- Min-Max Cut off
- Sigmoidale

Training Strategies

✻ Training a network depends on the networks characteristics.

✻ Many networks require iterative training, that is the data is presented to the network repeatedly.

✻ For the training a system, the developer faces a number of training issues:

- Proportional Training
- Over-training and Under-training
- Updating
- Scheduling

Search for the best network

- Use exhaustive search
- Use Gradient search methods
- Use Genetic Algorithms
- Use Probabilistic search methods
- Use Deterministic search methods

Proportional Training

- In training a neural network, it is not uncommon that a fact is presented to the network many times. The question is how many times should a given training fact be presented to the network?
- For example, if the training set has 500 fact, it takes 500,000 iteration to train the network, each fact is presented to the network 1,000 times. Is that sufficient? If a training fact is repeated a number of times in the data set, it must receive a proportional representation in the number of iterations it is presented to the network.

OVER-TRAINING AND UNDER-TRAINING

✻ In an iterative training process, a higher number of iterations does NOT always mean a better training outcome.

✻ Too many iterations may lead to over training and too few iterations may lead to under training.

✻ The reason for overtraining is not quite clear yet, but it has been reported that in some networks, training beyond a certain point may lower the over performance of the network.

✻ The developer of the network should reach a balance between the two.

- The performance of the system is usually measured with the endogenous data (data used for training the system) and exogeneonous data (data used to test the system, it should represent the data that the system has not seen yet.)

How Do You Know When to Stop Training

✻Time

✻Number of Epochs

✻Percentage of Error

✻Root Mean Square Error (RMS)

✻R-Squared

✻Number of iterations since the best