# Artificial Neural Networks

Research & Academic Collaboration Program – 2018
University of Kelaniya

Dr. C. D. Tilakaratne
Senior Lecturer/Department of Statistics
University of Colombo

# Overview

1. Introduction to Data Mining
2. What is Artificial Neural Network
3. Feedforward NN
    1. Architecture
    2. How does FNN work
    3. Generalization ability
    4. FNN Validation
4. Dynamic NN

# Introduction to Data Mining

- What is DM?
- Data Mining (DM) is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data – Fayyad
  - An essential part of knowledge discovery process

# Introduction to Data Mining

- Misbelief
  - DM tools automatically find the patterns you're looking for, without being told what to do
  - DM techniques are so sophisticated that they can substitute for contextual knowledge or for experience in analysis and model building
  - The methods used in DM are fundamentally different from traditional quantitative data analysis techniques

# Introduction to Data Mining

- What DM IS
  - A ''tool kit" for extracting the knowledge
  - An interdisciplinary exercise
  - Not replacing or anti-traditional statistics,
    - it is a valuable extension

# Introduction to Data Mining



- **DM vs. Statistics**

| Aspect | Statistics | DM |
|---|---|---|
| Size of data sets | Data set with a few thousand data points would be large | Can handle databases with million of records |
| Cleanliness of data | Clean data is necessary for most statistical analysis | Can handle contaminated data |
| Assumptions | Results depend on the validity of assumptions | No assumptions required |
| Patterns in data | Search for pre-determined pattern | Does not know precisely what sort of structure is seeking |
| Significance of results | Look for statistical significance | Statistical significance is not important |
| Nonnumeric data | Limitations in handing nonnumeric data (eg. image, audio, text data) | No such limitations |
| Spurious Relationships | Controlled in statistics (eg. control of overall experimental error) | Uncontrollable |

# Introduction to Data Mining

- Objectives of DM
  - Classification
    - Examining a feature of a newly presented object & assigning it to one of a predetermined set of classes
  - Estimation
    - Given some input data, coming up with a value for some unknown continuous variables such as income, height, or credit-card balance
  - Forecasting
    - Same as classification & estimation except that the records are classified according to some predicted future behaviour or estimated future value
  - Affinity grouping or association rules
    - Determine which things go together, also known as dependency modelling. eg. market basket analysis
  - Clustering
    - Segmentation a population into a number of subgroups or clusters
  - Description & visualization
    - Exploratory or visual data mining

# Artificial Neural Networks

- References
  - Elements of Artificial Neural Networks (Mehrotra, Mohan & Ranka)
  - Matlab User's Guide (Neural Network Toolbox)

# Introduction

- An artificial neural network is a computer program or hardwired machine that is designed to learn in a manner similar to human brain
  - Neuron → Node/Unit/Cell/Neuron
  - Synapse → Connection/Edge/Link
  - Synaptic Efficiency → Connection Strength/Weight
  - Firing Frequency → Node Output

# History



- The history of artificial neural networks began with the invention of the first artificial model for biological neurons by McCulloch & Pitts (1943).
- This model was utilised in the development of the first artificial neural network by Rosenblatt (1958).
- This network was based on a unit named a perceptron. It is also called single layer feedforward network. The perceptron could correctly classify linearly separable pattern class data.
- However, Minsky & Papert (1969), showed that the perceptron could not learn to classify linearly inseparable functions.
- It did show, however that these limitations could be resolved if networks contained more than one layer, but at that time no effective training algorithm for multi-layer networks was available.
- Rumelhart, Hinton, & Williams (1986) revived interest in neural networks by introducing the generalized delta rule for learning by backpropagation, which is the most commonly used training algorithm for multi-layer networks used today

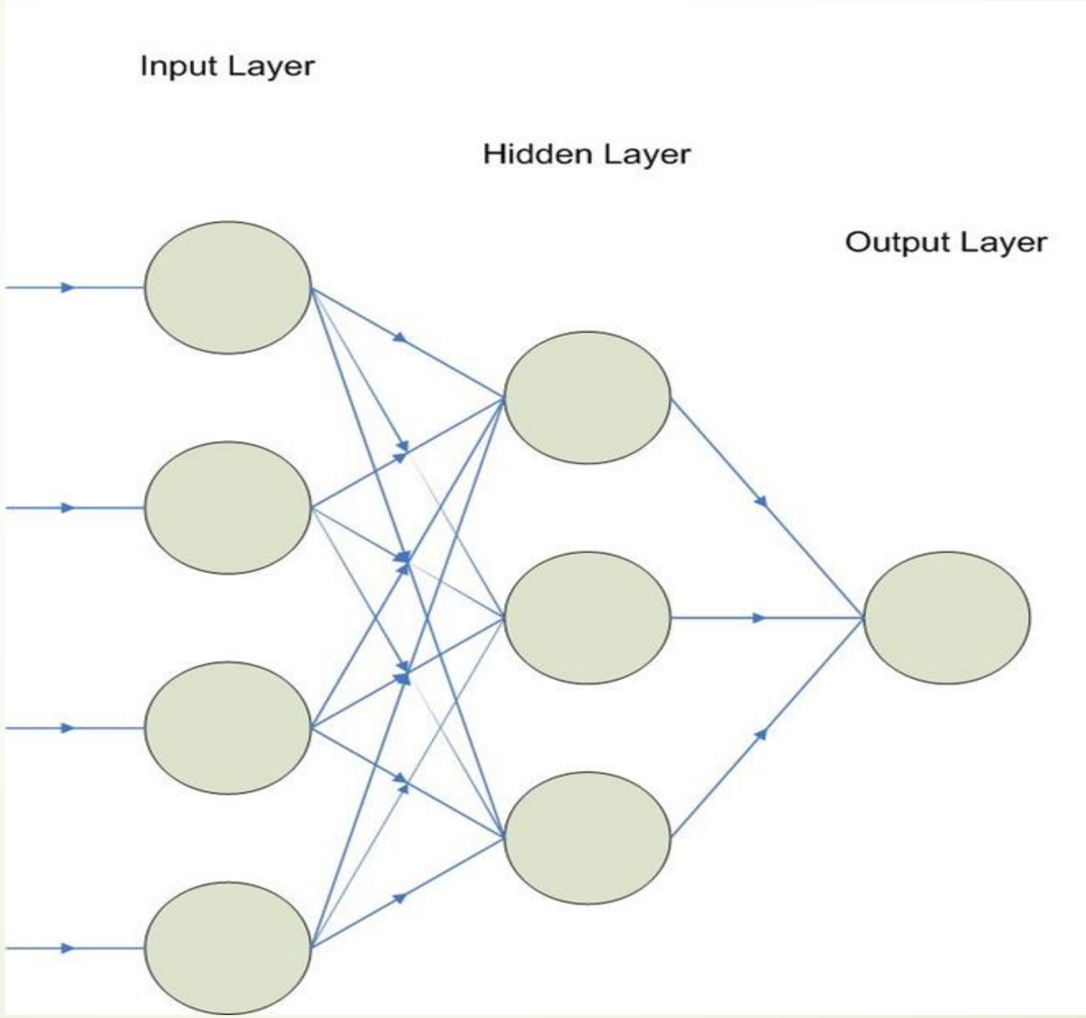
# NN Architectures

- Several different artificial neural network architectures have been developed in the past.
- Common to all these architectures is the linking together of many neurons, or nodes, into numerous interconnected pathways of inputs and outputs.
  - Fully connected networks
  - Layered networks
  - Acyclic networks
  - **Feedforward networks**
  - Modular neural networks

# Multilayer Feedforward Neural Networks

- Multilayer feedforward networks with non-linear node functions can overcome the limitations related to the perceptron
- When these networks are trained with backpropagation algorithm they are referred to as 'Multilayer perceptrons' (MLPs).

# Multilayer Feedforward Neural Networks



# Multilayer Feedforward Neural Networks

- Above figure depicts an example of a multi-layer feedforward neural network.
- A multilayer feedforward neural network can have any number of layers and any number of units (neurons) per layer.
- The first layer is called the Input Layer and the last layer is called the Output Layer. The middle layers are called Hidden layers.

# Multilayer Feedforward Neural Networks

- Each neuron-to-neuron connection is modified by a weight (or connection strength).
- In addition, each neuron (in processing layers)has an extra input that is assumed to have a constant value of one, and the weight that modifies this extra input is called the bias.
- All the information propagates along the connections in the direction of network inputs to network outputs, hence the term feedforward.

# How does a FNN work?

- Input layer nodes simply transmit input values to the hidden layer nodes, and do not perform any computation
- The no. of input nodes equals to the dimensionality of input patters (i.e. no. of input variables/ features)
- The no. of output nodes is decided according to the problem under consideration

# How does a FNN work?

- The input neurons simply pass on the input vector
  - For example *i*-th input vector (*i*=1, …*n*):

$$I_i = \{X_{ij}: j = 1, 2, \cdots, m\}$$

- The neurons in the hidden layer(s) and the output layer are processing units.
- Each processing neuron is associated with an activation function.

# How does a FNN work?



- The following equation gives the net output from the *r*-th neuron of the hidden layer:

  - $net_r = f_1\left(\sum_{j=1}^{m} w_{jr} X_{ij} + b_r\right)$

- where $X_{ij}$ is the *i*-th input to the *j*-th neuron of the input layer, $w_{jr}$ is the weight of the link connecting *j*-th neuron of the input layer to *r*-th neuron of the hidden layer, and $b_r$ is the bias associated with the *r*-th neuron of the hidden layer. $f_1$ is the transformation function associated with the hidden layer.

# How does a FNN work?

- Output of the *s*-th neuron of the output layer is given by:

$$net'_s = f_2\left(\sum_r w_{rs} net_r + b'_s\right)$$

- where $f_2$ is the transformation function associated with the output layer. $w_{rs}$ is the weight of the link connecting *r*-th neuron of the hidden layer to *s*-th neuron of the output layer, & $b'_s$ is the bias associated with the *s*-th neuron of the output layer. $net_r$ is defined as given in the previous slide.

# How does a FNN work?

- A network is fed with inputs as well as outputs (training data or sample data). Then the network learns the mapping from inputs to corresponding outputs. This is called **supervised learning**.
- In supervised learning FNN minimizes the error sum of squares (i.e. $\sum_{i=1}^{n}(a_i - e_i)^2$ , where $a_i$ and $e_i$ are the actual and estimated values of the *i*-th output)
- In order for the network to learn the patterns of the data, a learning algorithm is needed. Backpropagation is the learning algorithm most commonly used for the feedforward neural networks.
  - There are several backpropagation algorithms
  - In backpropagation learning weights and bias are adjusted in a way the error sum of squares is minimized

# Transfer functions

- Linear function
  - The linear transfer function calculates the neuron's output by simply returning the value passed to it.
    - purelin
- Sigmoid functions
  - These transfer functions are commonly used in backpropagation networks, because they are:
    - Continuous and differentiable everywhere
    - Rotationally symmetric about some point
    - Asymptotically approach their saturation values
  - Multilayer networks often use the log-sigmoid transfer function
    - logsig
  - Alternatively, multilayer networks can use the tan-sigmoid transfer function
    - tansig

# Generalization ability of FNN

- A network must perform well on unseen data distinct from data used for learning
- Therefore, in network development, available data is separated into 2 parts: training data and test data
- Excessive training on the training data sometimes decreases performance on the test data . One way to avoid this danger of ‘overtraining’ is by constant evaluation of the system using test data as learning proceeds

# Generalization ability of FNN

- A NN is considered to be efficient in prediction, only if it can predict the out of sample (test) data well.
- A NN said to be well generalized when the input-output mapping computed by the network is approximately correct for the test data.
- In backpropagation learning, it is expected that a network become well trained.

# Generalization ability of FNN

- A NN with large no. of neurons may predict the sample data well, by over-fitting the input patterns, but may poorly predict the out of sample data.
- Because of this, network of smaller sizes (with smaller no. of neurons) are empirically preferred over large networks.
- However, a network with few neurons may not be powerful enough to learn the input patterns well.
- Therefore, it is advisable to choose the appropriate network size, in order to obtain accurate prediction on out of sample data.

# Generalization ability of FNN

- A network that produces a high level of forecasting errors on the test data, but a low level of errors on training data, is said to be overfit to the training data.
- This happens when testing of network is done after completing the training.
- One solution for this problem is to constantly monitor the performance of the network on the validation data.

# Generalization ability of FNN

- The weights are adjusted only on the basis of training set, but the error is monitored on the validation set.
- Training continues as long as the error on the validation set continues to decrease, and is terminated if the error on the validation set start to increase.

# Generalization ability of FNN

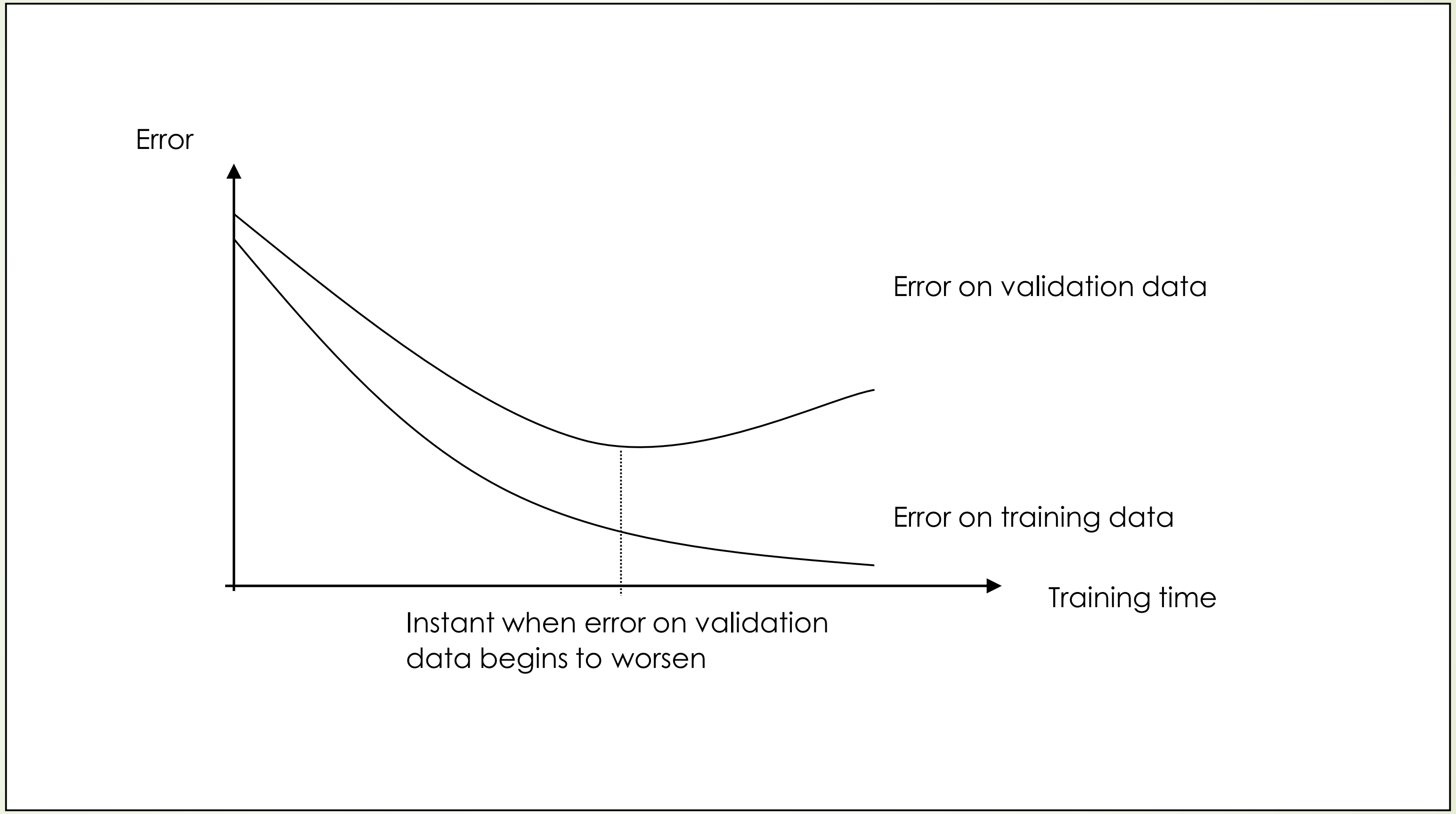



Figure 1: Change in error with training time, on training set and validation set

# Generalization ability of FNN

- Performance over the validation set is monitored over several iterations to eliminate random fluctuation of error.
- In this method the validation data is not used for training; weight changes are computed only on the basis of the network's performance on training data.
- With the stopping criterion, final weights depend on the validation data in an indirect manner.
- Since the weights are not obtained from the current validation data, it is expected that the network will continue to perform well on future validation data.

# Prediction ability of FNN

- The size of the training set affects the predictability of a network.
  - If the size of the training set is too small, the network tends to overfit the data
- Predictability also depends on the no. of epochs over which a network is trained.
  - If the network trained for a large no. of epochs, it may be over-trained; hence the generalization ability becomes poor.
  - Alternatively, a network trained for a small no. of epochs, has poor predictive power, as it has not learnt the data properly.

# Prediction ability of FNN

- A practical method to determine a reasonable value for the maximum no. of epochs is to plot an appropriate measurement of error such as sum of squared errors at predetermined intervals (eg. at each 100th epoch). After plotting the error measure for a number of randomly selected starting weights the researcher can choose the maximum no. of epochs based on the point where the error measure stops decreasing quickly and flattens.

# FNN Validation

- Once a NN is trained, its prediction ability should be validated.
- There are 2 commonly used methods:
  - Cross validation
  - Test-set validation
- Out of these two methods, the most common is test-set validation.

# Cross Validation

- The whole data set is randomly allocated to *k* equal-sized samples. One sample is separated for testing and all other samples are used for training. The performance of the trained network is estimated by suitable error/accuracy function. This procedure is repeated for all *k* samples, thus producing *k* models with associated error/accuracy estimates. The mean of these estimates is used as a total estimate of the prediction error/accuracy of the fitted NN model.
- However, this method is too expensive in terms of computation when *k* is large. This problem can be overcome by using the weights from previous training as the starting values for the next model.

# Test-set Validation

- The standard method of neural network validation is to allocate the majority of data into a training set and balance into a test set.
- Sometimes the training set is further divided into a validation set.
- The training set is used to build the model while the validation set is used to test the training as it proceeds to determine the best point for early stopping or the best network architecture (no. of hidden layers and no. of neurons in a hidden layer).
- The test set is only used for final estimation of the generalization performance.
- The disadvantage of this method is wastage of data which are used for testing, because this data set is not used for training the NN model.

# Test-set Validation

- Common ways of selecting the test set are either to select the most recent data for the test set or to select some percentage of data randomly from the entire data set.
- The advantage of a randomly selected test set is that the risk of using data characterised by one type of pattern is largely avoided.
- However, researcher must be careful not to include the test data into the training set.
- Selecting the most recent data generated after that of training set is possibly the most ideal approach.

# Test-set Validation

- Another method of selecting test set is the deterministic method.
- Select every *n*-th (*n*< size of the whole data set) data point for the test set
- However, this method is not recommended since it can result in cycles in the sampled data.

# Dynamic NN

- NN can be classified into dynamic & static categories.
- Static networks (eg. feedforward) have no feedback elements & contain no delays.
- In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network.
- Dynamic networks can be divided further into two categories:
  - those that have only feedforward connections (nonrecurrent network), and
  - those that have feedback, or recurrent, connections (recurrent networks).

# Nonrecurrent Dynamic Networks

- When a network contains delays, the input to the network would normally be a sequence of input vectors that occur in a certain time order.
  - A simple network that contains one delay

# Recurrent NN

- A recurrent neural network (RNN) is a class of NN where connections between units form a directed cycle.
- This creates an internal state of the network which allows it to exhibit dynamic temporal behaviour.
- RNN can also propagate data from later processing stages to earlier stages.

# Advantages/Disadvantages of Dynamic networks

- Advantages
  - Dynamic networks are generally more powerful than static networks.
  - Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns.
  - Can be applied to many areas that static networks fails.
- Disadvantages
  - More difficult to train

# Applications of Dynamic Networks

- Applications in many disparate areas, such as:
  - Prediction in financial markets
  - Channel equalization in communication systems
  - Phase detection in power systems
  - Sorting
  - Fault detection
  - Speech recognition
  - Prediction of protein structure in genetics

# NARX Network

- The Nonlinear AutoRegressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosing several layers of the network.
- a nonlinear autoregressive exogenous model (NARX) is a nonlinear autoregressive model which has exogenous inputs. This means that the model relates the present value of the time series to both:
- past values of the same series; and
- present and past values of the driving (exogenous) series
- The NARX model is based on the linear AutoRegressive model, which is commonly used in time-series modeling.

# Applications of NARX Network

- There are many applications for the NARX network.
- can be used as a predictor, to predict the next value of the input signal.
- can also be used for nonlinear filtering, in which the target output is a noise-free version of the input signal.
- can be used for modeling of nonlinear dynamic systems.