Chapter 1
Overview : Artificial Neural Network

Dinesh Mainali

Outlines

- Neural computing
- Neural computing applications
- Overview of neural computing
- Engineering approaches to neural computing
- ANNs: The mapping viewpoint
- The structure viewpoint
- learning approaches
- Relationship of ANN to other technologies
- Historical efforts

Neural computing

- Traditionally computing refers to the programmed computing where algorithm for a given task is designed and then implemented in any of the programming languages
- Alternative perspective of computing called soft computing is suitable when it comes to model the biological system like human brains. Soft computing differs from hard computing in following sense:
  - Parallel and distributive computation
  - Learning replaces priori program development

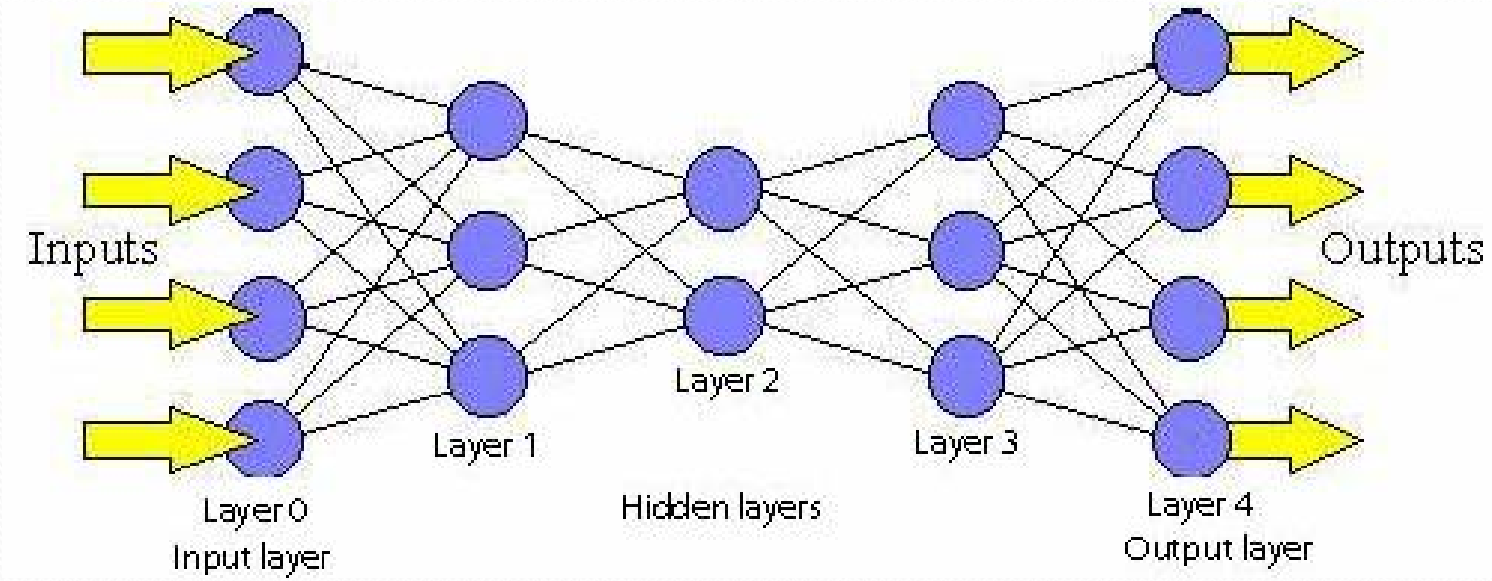
- Biologically motivated computing paradigm
- Can be dominant computing architecture and ANN may become ultimate RISC building block
- No technology fit for every type of problem. It's the system designer's responsibility to trade off conventional solution against the ANN alternative
- Three fundamental questions to be addressed before implementing ANN
  - Is ANN really new and how can it be used?
  - How it is similar or different from other technology?
  - Does the Architecture fits in the comman ANN architecture?

Definition of ANN

- A structure or Network composed of a number of interconnected units (artificial neurons). Each unit has I/O characteristics and implement a local computation or function. The output of any unit is determined by its I/O characteristics , its interconnections to other units and possibly external inputs
- The manual design of the network is possible(but tedious job) but they are so designed that after training via different sample data they acquire overall functionality
- The overall functionality is determined by the network topology, individual neuron characteristics , training methods and training data

Fundamental neural network concept

- The ANN model consists of reconfigurable interconnection of simple elements or units

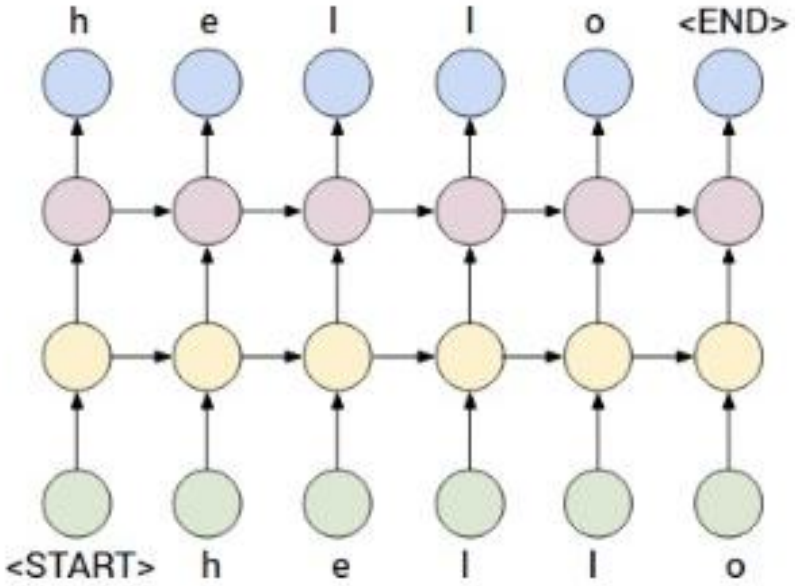


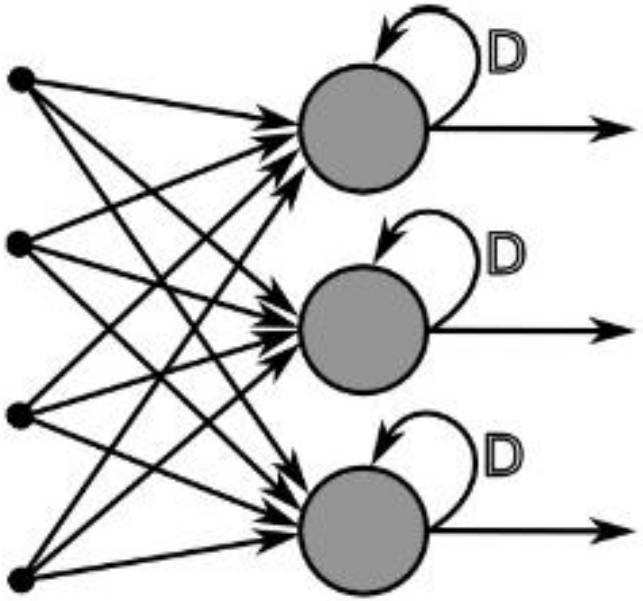
Feed forward neural network

What are Recurrent Neural Networks

1. In their simplest form (RNNs), they are just Neural Networks with a feedback loop

2. The previous time step's hidden layer and final outputs are fed back into the network as part of the input to the next time step's hidden layers.





- In the above figures it is worth to identify
  - Nodes
  - Different Layers
  - Open loop
  - Closed loop
  - Hidden units and hidden layers
- Closed loop or feedback allows the network to exhibit complex temporal dynamics compared to open loop
- Hidden units form hidden layers and they are not interfaced with external world
- Nodes and links determines the network topology

- The Individual units implement the local function and the overall network of interconnected units display a corresponding functionality
- The analysis of the whole network can merely done by training and testing with data
- An ANN designer designs a network with the help of specification provided to him or her and it is his/her duty to determine the network parameters as per specification
- In the learning process weight adjustment is the crucial factor and system knowledge or experience or training is stored in the form of weight
  - Weight describes the strength of inter-element connectivity

Neural Computing Applications

- Image processing and computer vision
- Signal Processing
- Pattern recognition
- Medicine and medical research
- Military system
- Financial system
- Planning Control and Search
- Power system
- Human interfacing

Brief overview of Neural Computing

- Biological system functionality
  - Explained in previous lecture
  - Motivation: Adaptability, context sensitive nature, error tolerance large memory capacity, and real time capability of biological system
  - Basis of biological computation is a small number of serial steps, each occurring on a massively parallel scale
  - Each unit of parallel architecture is simple processing body and locally connected

Neuromorphic Computing

- A new discipline based on design and fabrication of artificial neural system
  - E.g.: vision system, speech recognition system
- Has a wide range of application varying from non linear adaptive control of complex system to design of autonomous functional body
- Most of all fundamental principles of this engineering are inspired from biological system

Human brain computattional properties

- Although we have complete knowledge of neuron's physiological architecture still the characteristics of fundamental high level computation is a mystery
- Huge number of processing units and complex interconnection among them
- Human brain are evolved and still in the process optimizing the brain for survival

Characteristic of human brain

- integrates and stores experiences(self–organize experiences)
- Consider new experience in context of stored experience
- Able to make accurate prediction about the new situations
- Does not require perfect information
- Represent a fault tolerant architecture
- Seems to have available, perhaps unused, neurons ready for use- continuous learning
- Does not provide much useful information regarding its operation at high level
- Tends to cause behavior that is homeostatic

Neural approach to computation

- Training Vs Programming
  - young children are not programmed but learn by example and adaptation
  - For training approach to be feasible computer must be trainable and sufficient training data must be available
- Neural system
  - Trainable, adaptive and even self-organizing information system
  - Develop a functionality based on training
  - May provide computational architecture through training rather than specific design

- Mathematical models and simulation
  - Mathematical models : A set of non-linear , n-dimensional equations that characterize the overall network operation, as well as structure, unit , dynamics, and training
  - generally difference and differential equations are employed
- Simulation: performance analysis of ANN is done via simulation in digital computer
  - Often requires tremendous computation
  - May require modification of actual structure

- "Connectionist" model and computing
  - Connectionist model: Based on the notion that many human computational processes are naturally carried out in highly parallel fashion with significant interaction between processes
  - If architecture exist such computation fashion can be emulated in machine learning
  - Overall computation is distributive: each single unit provides the computational effort
  - Connectionist computing model may be shown to have a number of abstract properties relevant to their application
  - In some way it is a generalization of neural network, an unit may be more complex than a neuron

Advantages of ANN

- Massively parallel computation
- Fault tolerant
- Adaptive
- Little need for extensive characterization of problem other than through the training set

Disadvantages

- No clear design rules for any arbitrary problem
- No generic rule to asses the internal operation of the network
- Training in most cases is difficult and sometime impossible
- Difficult to predict future network performance

Engineering approach to neural computing

- Before trying to solve a problem using neural computing approach, we must address the three questions

1. Are ANN techniques suitable or even applicable? Does the problem have one or more solutions
2. Can we develop or modify useful ANN architecture for the situation and train the ANN ?
3. are there formal tools and heuristic that may be applied to asses the ANN solution procedure?

Procedure for ANN System Engineering

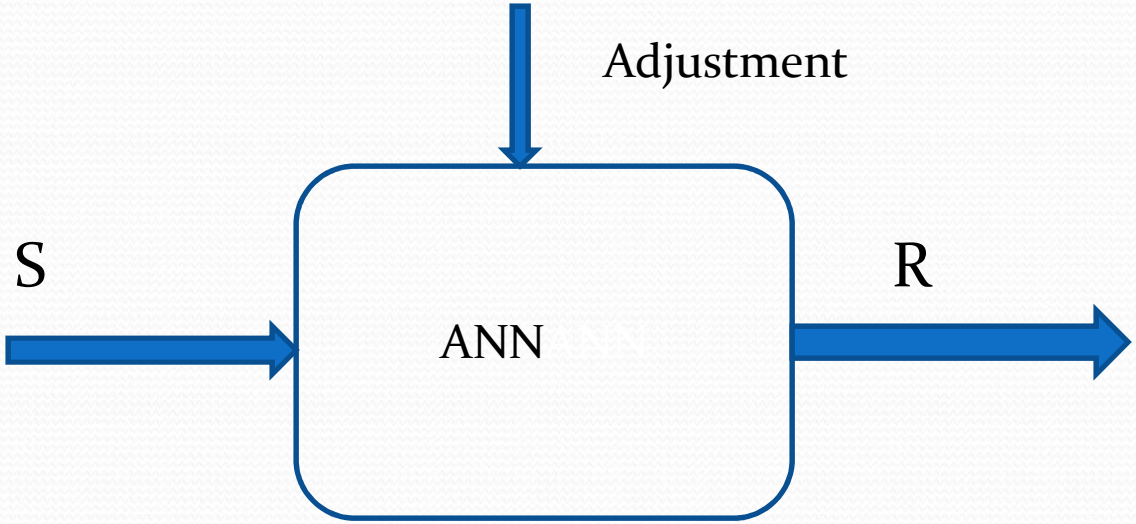
- Study the classes of measurements and patterns under considerations to develop possible characterization
- Determine the availability of measurement(inputs) or features
- Consideration of constraints like system performance and computational resources on desired system
- Consider the availability and quality of training and test data
- Consider the availability of suitable and known ANN structure
- Develop an ANN simulation
- Train ANN
- Simulate ANN system performance using test sets
- Repeat the above processes until desired performance is achieved

ANN: The Mapping Viewpoint

- Mapping: mapping of states of conceptual problem into ANN state OR a specific I/O or perhaps time varying state mapping

Basic perceptual system and stimulus-response approaches

- Human perceptual system
  - Includes memory and sensory information processing
  - Elemental SR characterization of perceptual system as in fig below represent a black box system



- The internal computation is irrelevant, not understood or not quantifiable
- Train the black box to learn the correct response for each of the training sample
- Priori knowledge and detail understanding of internal architecture is minimal
- After training, it is hope that the internal architecture is self-organized to extrapolate on the basis of prior experience when new data of similar pattern is fed
- Structure of ANN may be hierarchical- a box of boxes where internal boxes may have different topology
- Main idea of black box approach is developing mapping between input and output
- The success of this approach highly depends upon learning algorithm and training samples
- Also it is not guaranteed that provided algorithm and samples wou;d train an ANN for specific application

Network Inputs and Outputs

Input selection:

- Process of choosing inputs to ANN
- Requires considerable judgment
- Inputs may be pre-processed
- Mathematical tools and simulation may help to identify the appropriate inputs
- Restriction in measurement systems for a given application may restrict the set of possible inputs

Input distortion

- ANN mapping invariance to certain deviation in input pattern is desired
- ANN exhibit invariance to some extent of input perturbations
- Same response is desired for a class of input stimuli and may be the deviation in input is due to undesired causes like noise
- Humans are good at it
-

Output selection

- Application dependent
- Eg in digit recognition example the output always be on eof ten symbols

Vector Representation for SR characteristics

- Usually the ANN stimulus(input of ANN or state of neuron) is represented as vector x and the desired response as x_d
  - the desired mapping can be expressed as

$f_{D:}\, s \longrightarrow r \quad or \quad r=f_D(s)$

  - linear mapping for above type equation can be performed which can be implemented using mapping matrix : $r = Ms$
    - *Matrix M dimension obeys the matrix multiplication rule. This allows input and output to be in different vector space*

Parameters, weight and constraints

- For a single unit *p* considering its parameters '*a*' mapping can be stated as:

$$r = f_p(s, a_p)$$

- Specifications of single desired I/O of the form (s,r) places a constraints on f_p and a_p
- Moreover, constraints for other (s,r) pairs can be added
- Solution of this constraint satisfaction problem may or may not exist or may have multiple solutions
- Now we can have a combined characteristics of whole network as

$$r_i = f(s_i, a_c, w)$$

w represents the weight

- The desired network behavior as (s_i , r_i)put constraints on f, a_c , w
- Choose network structure with some constraints on w but initially the value is random
- Individual unit characteristics f, a_c are chosen
- Training now becomes finding the one or more solutions of w

ANN: The structure viewpoint

- ANN structure depends upon
  - Unit characteristics
  - Learning/training paradigm
  - Network topology
  - Network function

ANN functions

- What an ANN supposed to do- this can also determines the network type
- Some popular function of an ANN are
  - Pattern Association
  - Content addressable memory or associative Memory model (CAM) or (AM) : Hopfield net
  - Self organizing networks

Neural network structure

- Connectivity determines structure
- Group of neurons can be locally interconnected to form clusters /layers that are loosely or indirectly connected to other clusters/ layers
- ANN application requires an assessment of neural net architecture,
- Possible architecture includes:
  - Designing an application dependent network structure that perform desired computation
  - Selecting a common pre-existing structure for which training algorithm is available : feed forward network
  - Adapting a pre-existing structure to suit a specific application
- Recurrent and non recurrent neural net are two generic structure of neural net

Network topology

- Recurrent
- Non recurrent
- Layered, hierarchical
- Competative

ANN learning approach

- Learning , training and understanding are almost same
- Training : the use of information to refine current mapping behavior f_A toward f_D
- For applications like pattern associator H may be used iteratively to refine f_D by comparing the actual response r_a of the untrained system with the desired or target response r_d moving f_A closer to f_D through training
- Learning may be based on deterministic approach like back-propagation or hebbian OR simulated annealing (beyond our course)

Training sets(H) and Test sets

- Training set: Already available prior information of I/O mapping like samples of input defining desired system behavior
- A training set H can be described in the form of ordered pairs

$$H=\{(s_i,r_i)\} \quad i=1,2,3.....n \qquad (A)$$

The above equation is a specification for mapping in supervised learning. It defines only a limited number (n) of the possible infinite number of such mapping

- For unsupervised learning H are not mappings but input states ,the ANN must determine the clusters of the sample data

Generalization

- Any solution to equation A in the form $x_d = f(x)$ must satisfy the equation at *n* points in R^d
- What would be the behavior of points other than the n x
- We need generalization here
- Mapping constraints solely provided by the training set allows the either the generalized solution or the memorized solution
- Addition test set would be used to test and refine the mapping

Learning Curves

- A learning system may adapt its internal architecture to achieve a better response perhaps based on previously quantified performance
- Performance metric may be difference or error between desired and actual system output
- General learning techniques are based upon error correction e.g. GDR(generalized delta rule)
- GDR is a typical gradient descent technique: the system is modified following each iteration which may lead to typical learning curve
- Let p(n) is the probability of the subject making correct response in the n th trial of learning experiment

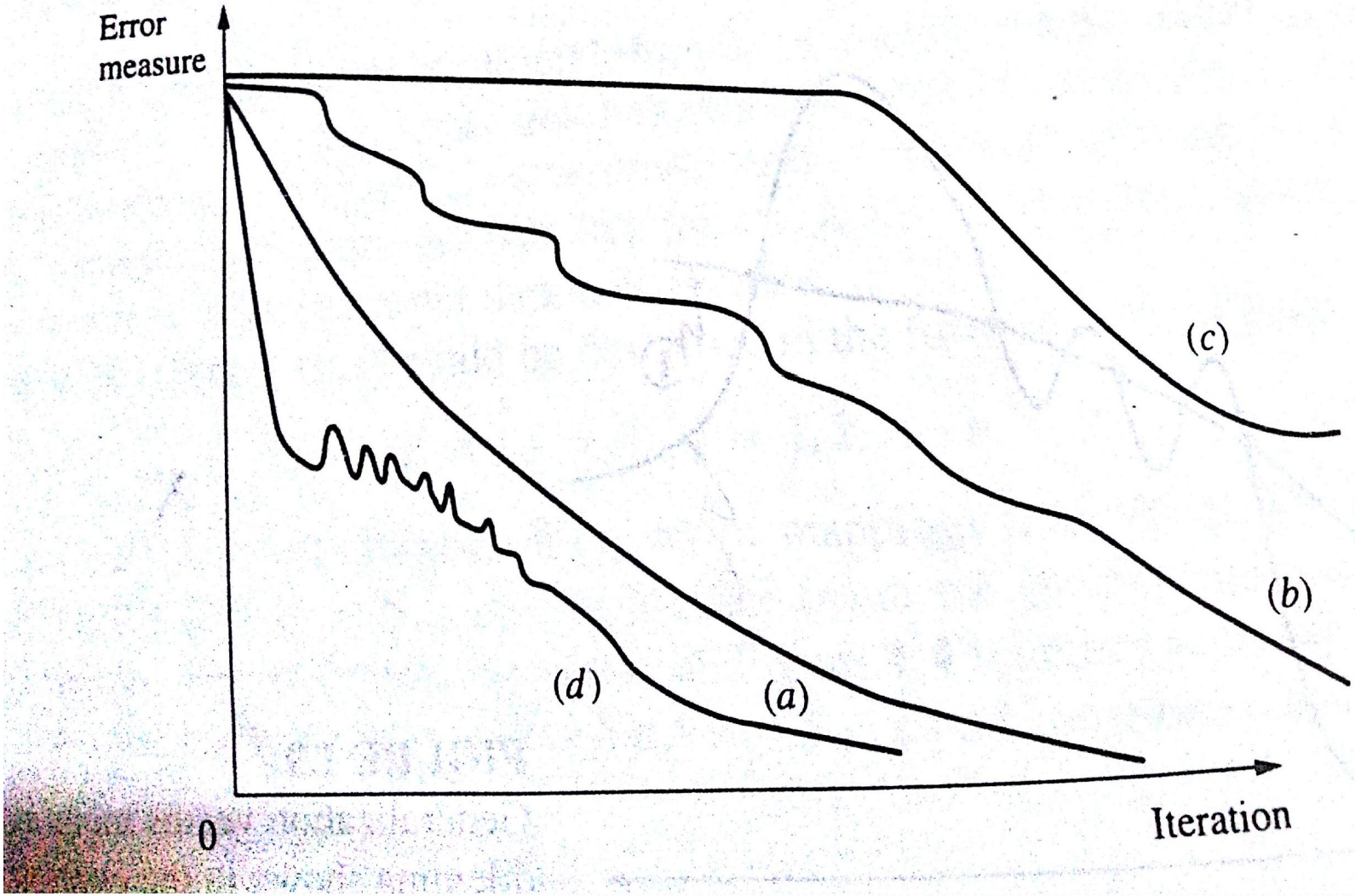
$$P(n)= 1-(1-P(1))\ (1 - \theta)^{n-1}$$

Here $n \geq 1$, and $\theta \in [0 , 1]$ is a learning parameter

- The initial error is reduced in subsequent trials in monotonically decreasing manner thereby increasing performance
- However the monotonically increasing performance is often difficult to attain in practical ANN system
- Another performance metric may be speed of learning and it is often confused with speed of the network in the actual application
    - Learning is always offline

Error measurement aand error trajectories

- Used to guide and asses training
- Various error measures exist ans we will be discussing the next chapters
- A typical of it is shown in fig below


Error
measure
(c)
(b)
(d)
(a)
0
Iteration

Relationship of ANNs to other techniques

ANN approach overlap with other technical areas. It is an imporatant component of soft computing which includes inter-related areas of

- Pattern recognition
- AI
- Fuzzy system
- Genetic computing: offer potentially important mechanism for the training of ANNs – genetic algorithm

- Pattern recognition
  - in some cases of pattern recognition there is underlying and quantifiable statistical basis for pattern generation
  - In other cases the underlying structure of the patterns provide the information fundamental for PR
  - In other neither of the above cases holds. For such cases we are able to develop and train the computational architecture to correctly associate input patterns with desired responses
  - Statistical PR assumes that there is a statistical basis for the classification of algorithms. Features are extracted from input data and each feature vector is assigned to one of the classes.
  - Features are assumed to be generated by a state of nature , and thus the underlying model is of a state of nature or class conditioned set of probabilities
  - Many example if neural implementation of statistical PR algorithms for classification

- Artificial intelligence
  - Some of the human intelligence are successfully emulated by the computers
  - Scientific and engineering approach is being expended in capturing the architectural and functional aspests of intelligent behavior
  - Till this date , emerging technology in ANN or neural computing has played a major role

Historical efforts

Perceptron and earlier (1940- 1960) : first generation of ANN

- **1943** McCulloch and Pitts proposed the McCulloch-Pitts neuron model
- **1949** Hebb published his book*The Organization of Behaviour*, in which the Hebbian learning rule was introduced
- **1958** Rosenblatt introduced the simple single layer networks called Perceptrons
- 1960 widrow: perceptron work was paralleled by widrow's LMS training algorithm for adaline/madaline devices. Both of them were single layer machine strongly influenced by pattern classification needs
- **1969** Minsky and Papert's book Perceptrons demonstrated the limitation of single layer perceptrons

Post-Perceptron : Second generation

Shortcomings of perceptron were addressed by general feed forward nets and associative learning. More architecture training algorithms arose

This era began with the realization that adding hidden layers to the network could yield significant computation versatality

Key events:

- **1980** Grossberg introduced his Adaptive Resonance Theory (ART) for adaptive network
- **1982** Hopfield published a series of papers on Hopfield networks(recurrent nets)
- **1982** Kohonen developed the Self-Organizing Feature Maps
- **1986** Back-propagation learning algorithm for multi-layer perceptrons was rediscovered, and the whole field took off again
- **1990s** ART-variant networks were developed
- **1990s** Radial Basis Functions were developed
- **2000s** Support Vector Machines were developed

- Key points
  - Mapping network sugmented with one ore more hidden layers became able to solve many problems which were not solvable for perceptron
  - Other interconnection strategies were considered-concept of recurrent neural net became popular
  - Unsupervised learning (self –organizing)concept began to emerge. This include Grossberg and Kohonen influnced net
  - Significant increase in available computing resourcefollowing Rosenblatt's work and positively influenced ANN work

Third generation

Performance and practical issues are paramount is now upon us

- Assessment of the limitation of networks
- Assessment of the generalization ability of ANN
- Fusion of ANN with other technologies like Genetic algorithm and fuzzy approach
- Implementation of ANN using dedicated hardware

Future Direction and open issues

- What is the complexity of the overall network?
- What is the complexity of training algorithm?
- How does the ANN solution scale with problem dimension?
- Does the ANN solution generalize?
- What does the network learn?
- Are parameters other than weights adjusted as part of training?
  - These questions leads us to the development of ANN and we strive for
    - Better understanding of ANN
    - Better design methodologies and tool
    - Incorporation of multiple networks
    - Exploration of comparing with alternative solution