Deep Learning with PyTorch Workshop

CHAPTER 1: INTRODUCTION TO NEURAL NETWORKS AND DEEP LEARNING

BY MUHAMMAD RAJABINASAB

Fundamental concept

NN are constructed and implemented to model the human brain.

Performs various tasks such as pattern-matching, classification, optimization function, approximation, vector quantization and data clustering.

These tasks are difficult for traditional computers

Artificial Neural Networks (ANN)

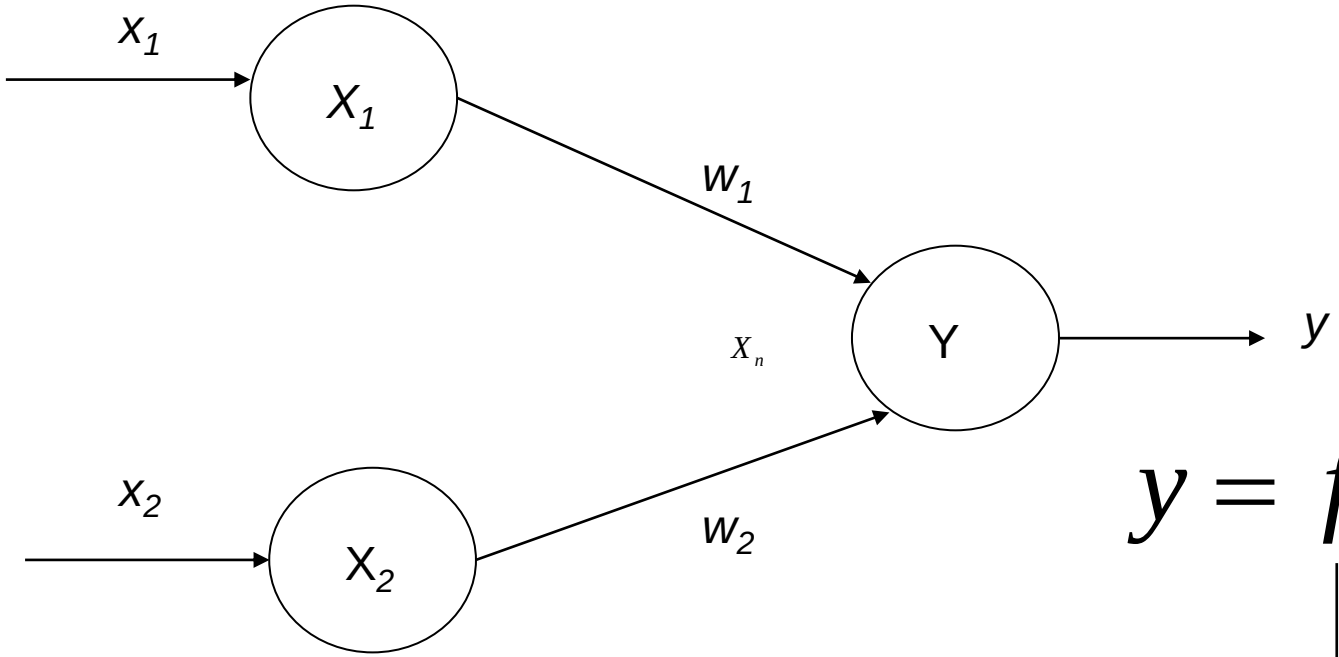
ANN possess a large number of processing elements called nodes/neurons which operate in parallel.

Neurons are connected with others by connection link.

Each link is associated with weights which contain information about the input signal.

Each neuron has an internal state of its own which is a function of the inputs that neuron receives.

Artificial Neural Networks

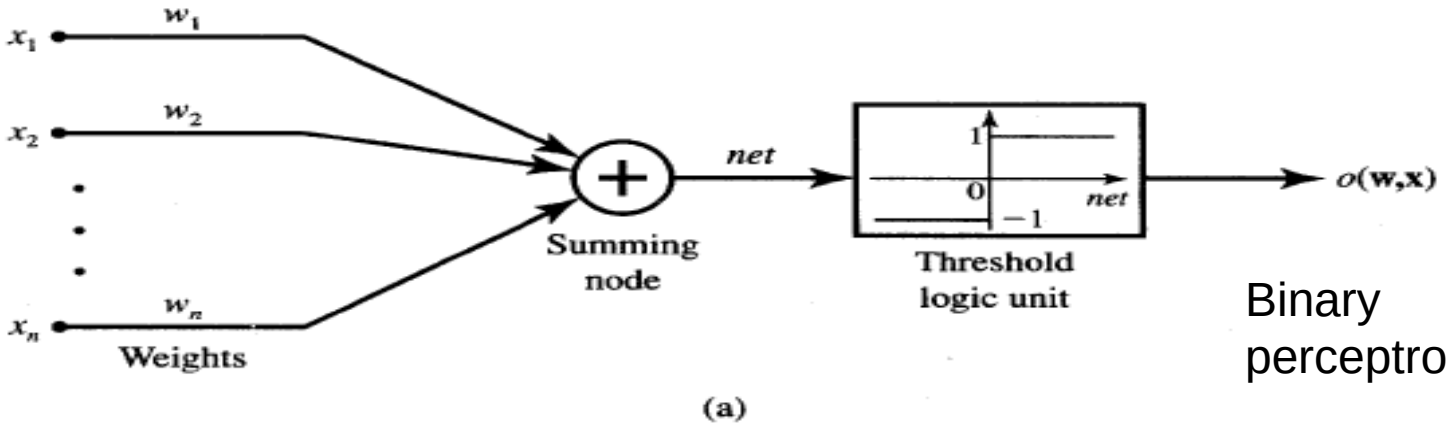


$$y = f(y_{in})$$

Activation Function

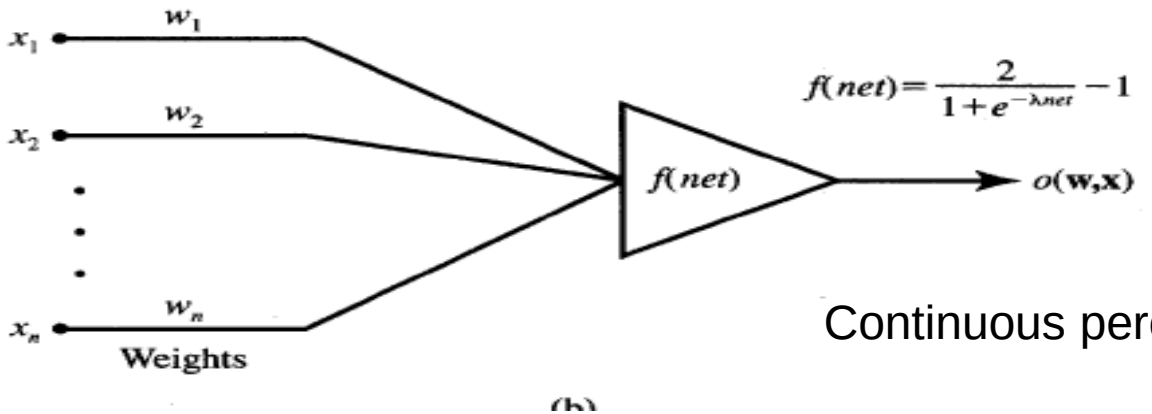
$$y_{in} = x_1 w_1 + x_2 w_2$$

Common models of neurons



(a)

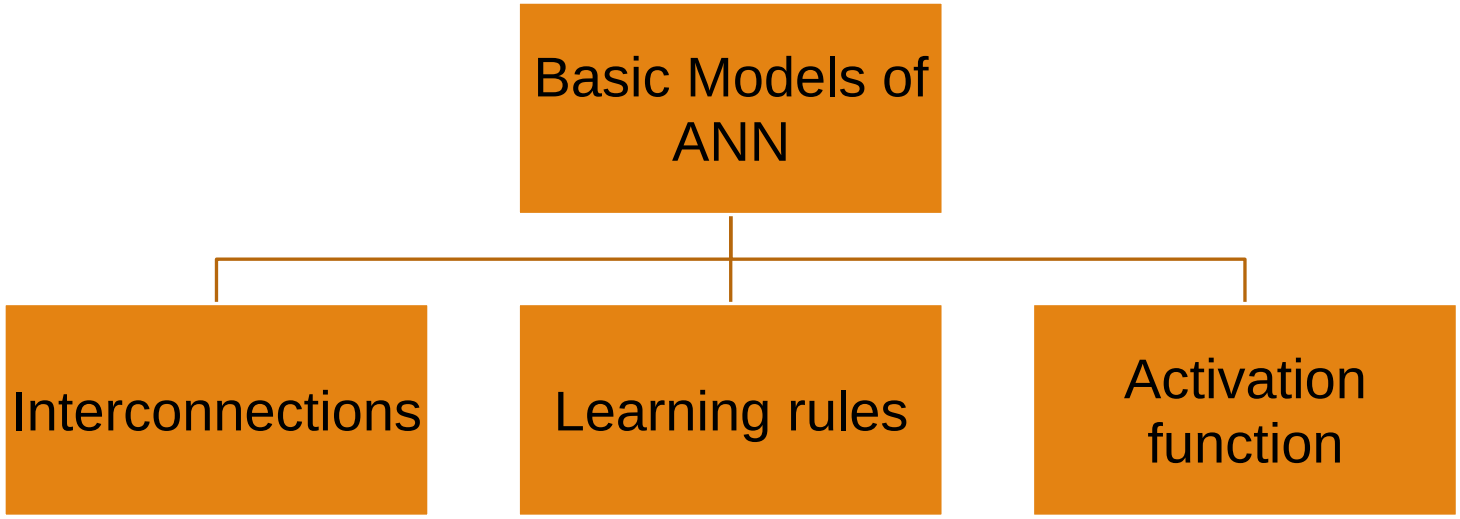
Binary perceptrons



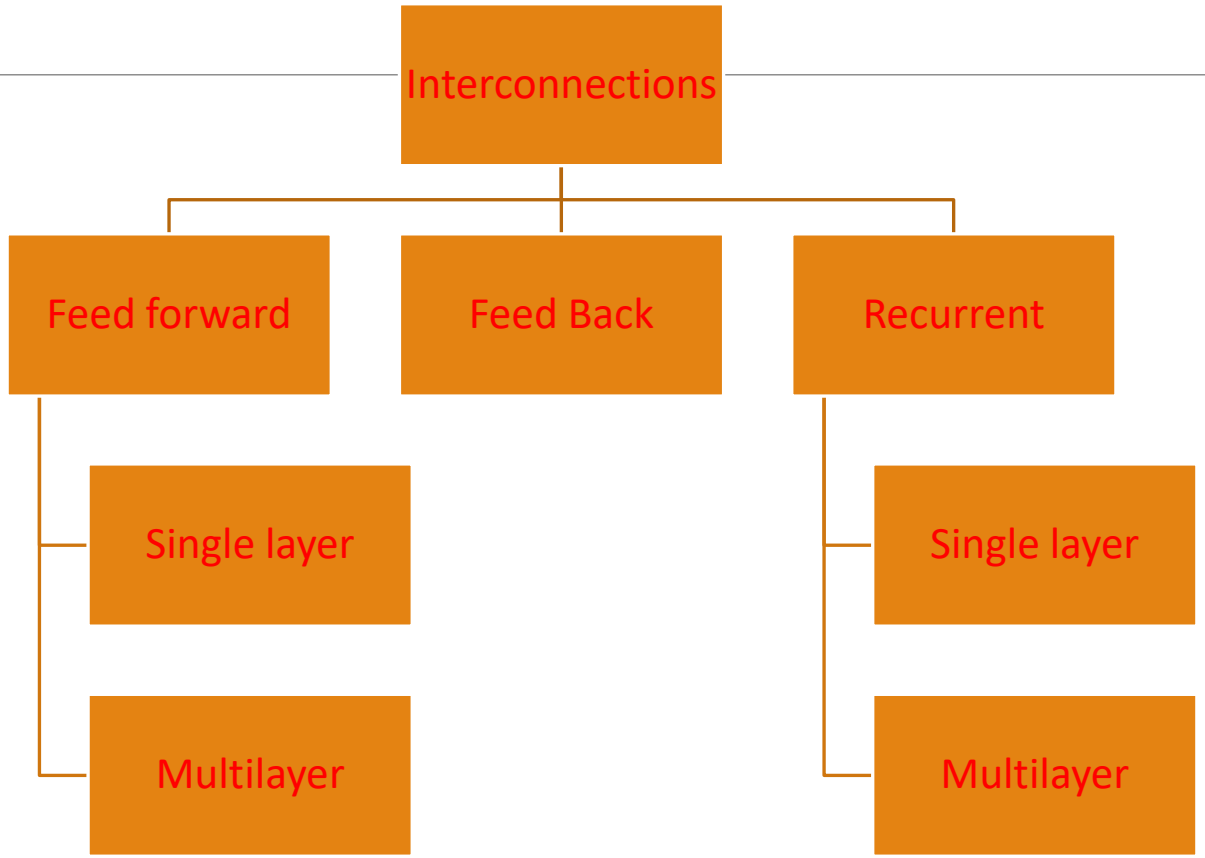
(b)

Continuous perceptrons

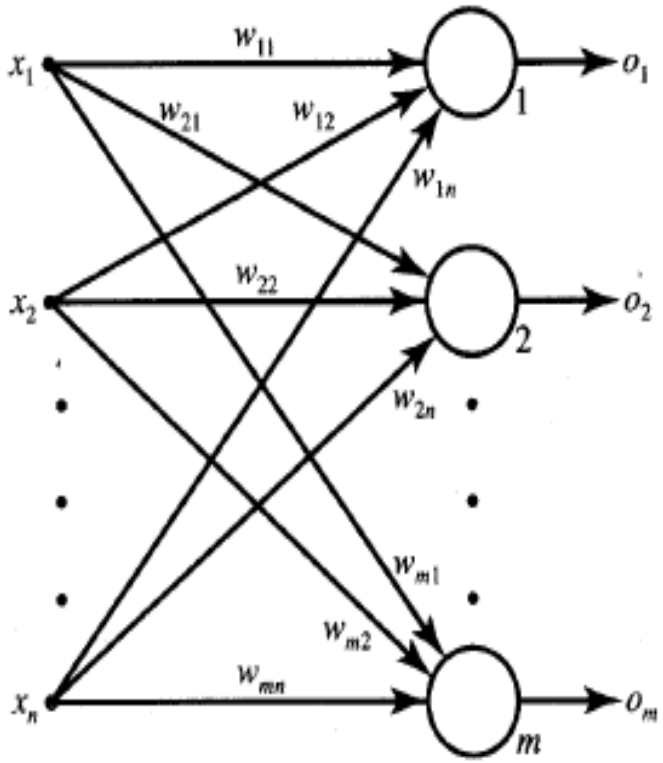
Basic models of ANN

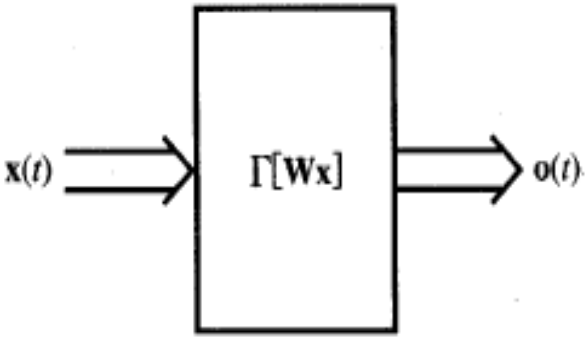


Classification based on interconnections

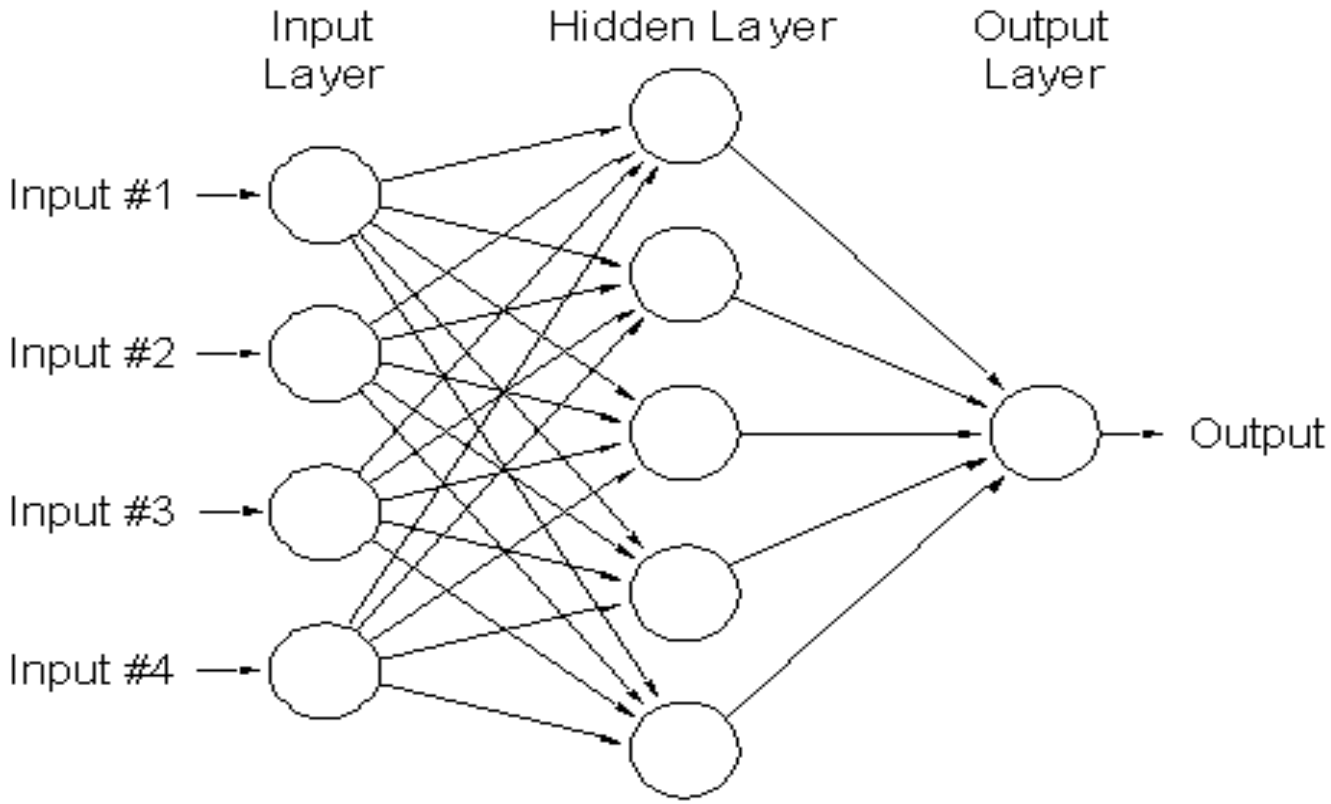


Single layer Feedforward Network





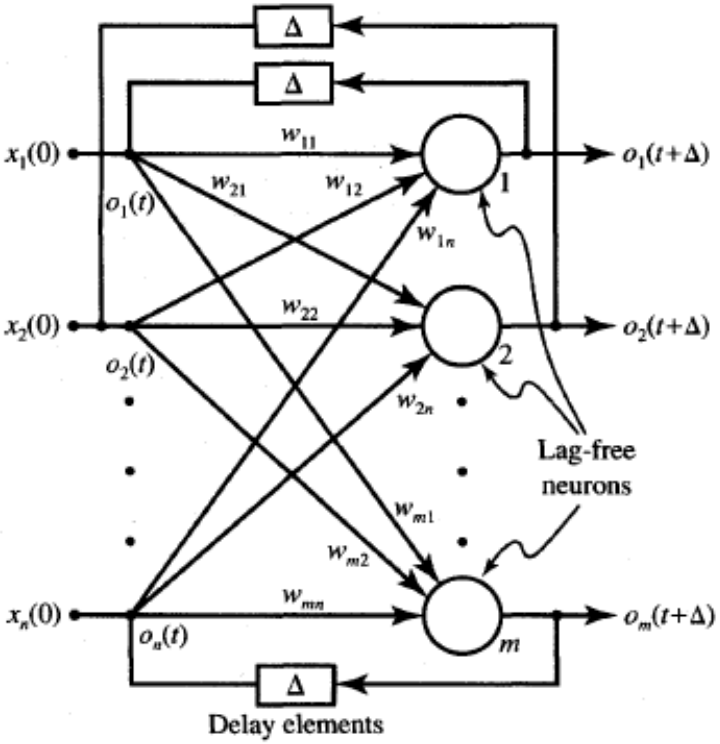
Multilayer feed forward network

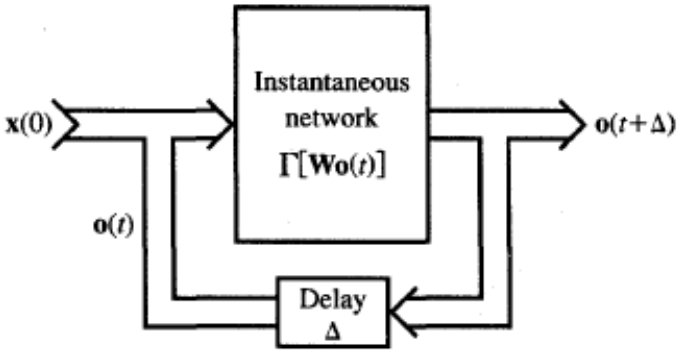


Can be used to solve complicated problems

Feedback network

When outputs are directed back as inputs to same or preceding layer nodes it results in the formation of feedback networks

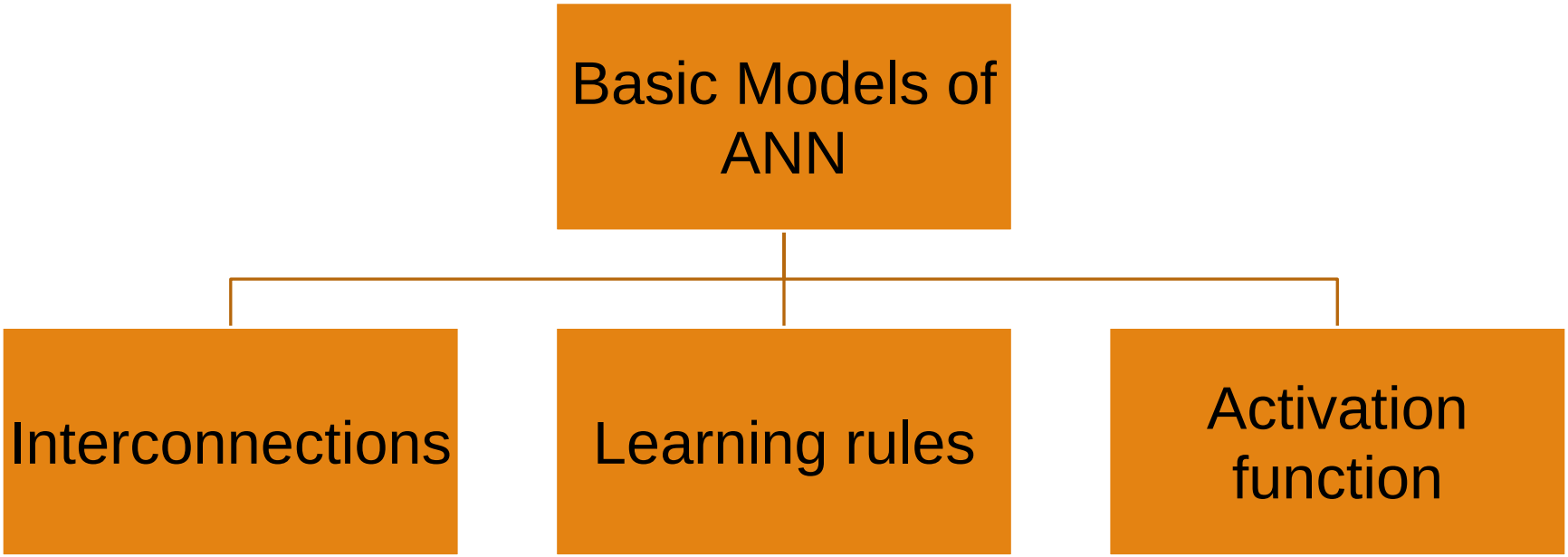




Recurrent Networks

Feedback networks with closed loop are called **Recurrent Networks**. The response at the k+1'th instant depends on the entire history of the network starting at k=0.

Basic models of ANN



Learning

It’s a process by which a NN adapts itself to a stimulus by making proper parameter adjustments, resulting in the production of desired response

Two kinds of learning

- Parameter learning:- connection weights are updated
- Structure Learning:- change in network structure

Training

The process of modifying the weights in the connections between network layers with the objective of achieving the expected output is called training a network.

This is achieved through

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Classification of learning

Supervised learning

Unsupervised learning

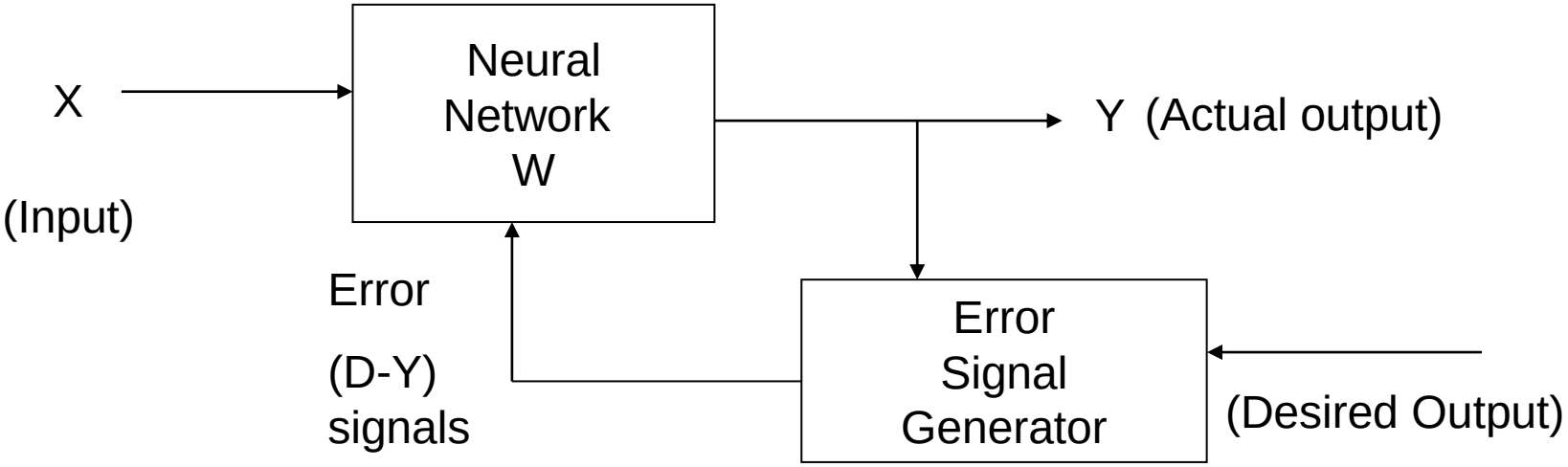
Reinforcement learning

Supervised Learning

Child learns from a teacher

Each input vector requires a corresponding target vector.

Training pair=[input vector, target vector]

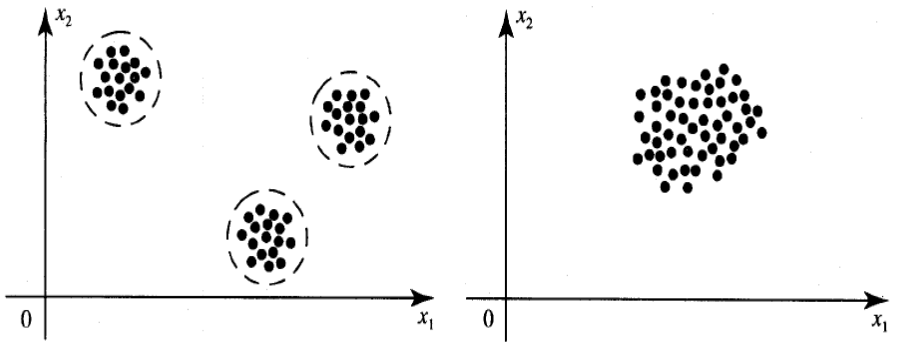


Unsupervised Learning

How a fish or tadpole learns

All similar input patterns are grouped together as clusters.

If a matching input pattern is not found a new cluster is formed



Self-organizing

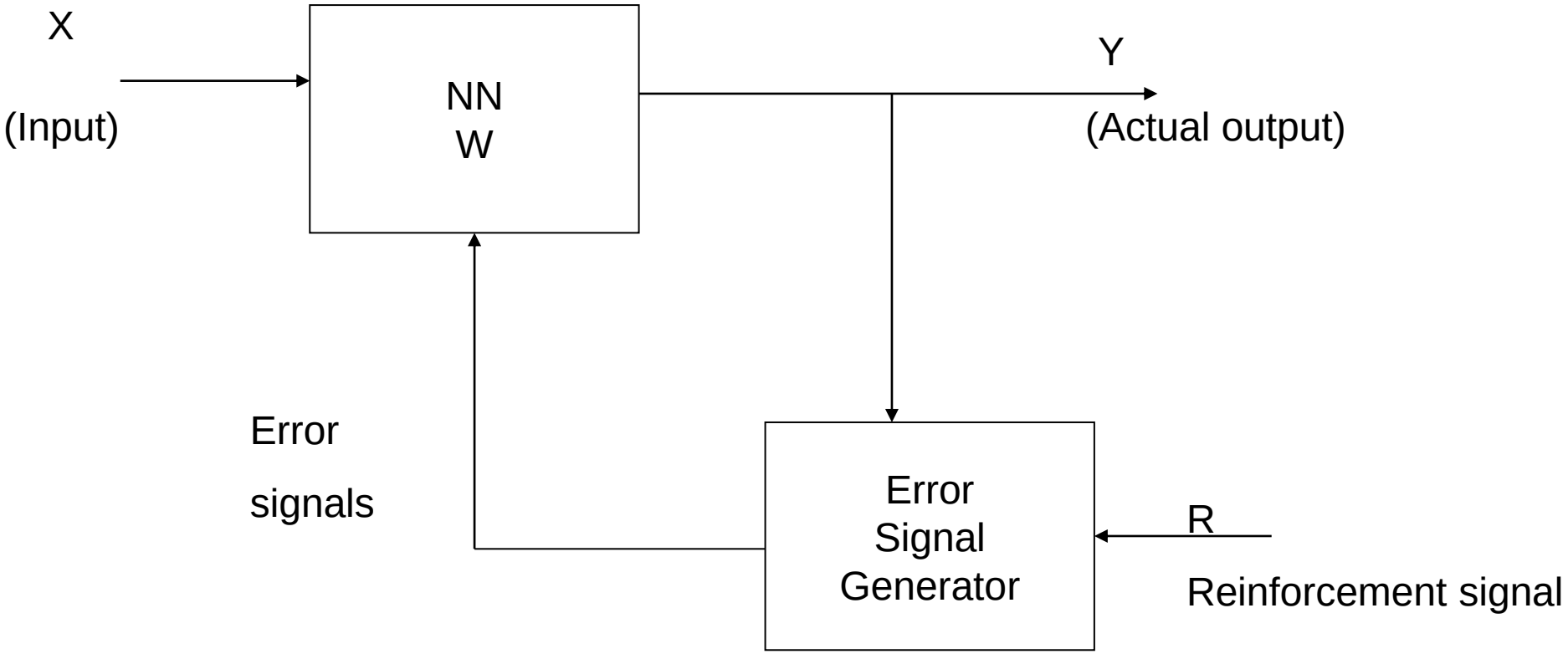
In unsupervised learning there is no feedback

Network must discover patterns, regularities, features for the input data over the output

While doing so the network might change in parameters

This process is called self-organizing

Reinforcement Learning



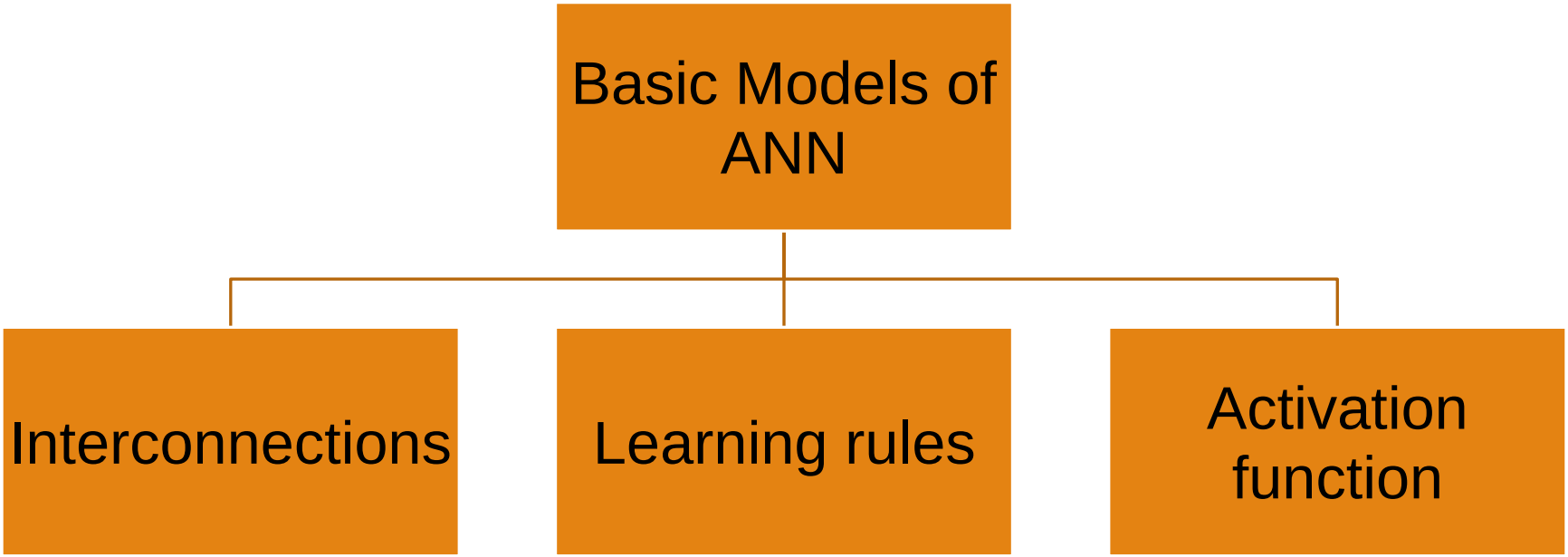
When Reinforcement learning is used?

If less information is available about the target output values (critic information)

Learning based on this critic information is called reinforcement learning and the feedback sent is called reinforcement signal

Feedback in this case is only evaluative and not instructive

Basic models of ANN



Activation Function

1. Identity Function
   f(x)=x for all x
2. Binary Step function

$$f(x) = \begin{cases} 1 if x \geq \theta \\ 0 if x < \theta \end{cases}$$

3. Bipolar Step function

$$f(x) = \begin{cases} 1 if x \geq \theta \\ -1 if x < \theta \end{cases}$$

4. Sigmoidal Functions:- Continuous functions
5. Ramp functions:-

$$f(x) = \begin{cases} 1 if x > 1 \\ x if\ 0 \leq x \leq 1 \\ 0 if x < 0 \end{cases}$$

Important terminologies of ANNs

Weights

Bias

Threshold

Learning rate

Momentum factor

Vigilance parameter

Notations used in ANN

Weights

Each neuron is connected to other neurons by means of directed links

Links are associated with weights

Weights contain information about the input signal and is represented as a matrix

Weight matrix also called connection matrix

Weight matrix

$$W = \begin{bmatrix} w_1^T \\ w_2^T \\ w_3^T \\ . \\ . \\ . \\ . \\ w_n^T \end{bmatrix} = \begin{bmatrix} w_{11} w_{12} w_{13} \cdots w_{1m} \\ w_{21} w_{22} w_{23} \cdots w_{2m} \\ \cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots \\ w_{n1} w_{n2} w_{n3} \cdots w_{nm} \end{bmatrix}$$

Activation Functions

Used to calculate the output response of a neuron.

Sum of the weighted input signal is applied with an activation to obtain the response.

Activation functions can be linear or non linear

Already dealt

- Identity function
- Single/binary step function
- Discrete/continuous sigmoidal function.

Bias

Bias is like another weight. Its included by adding a component $x_0=1$ to the input vector X.

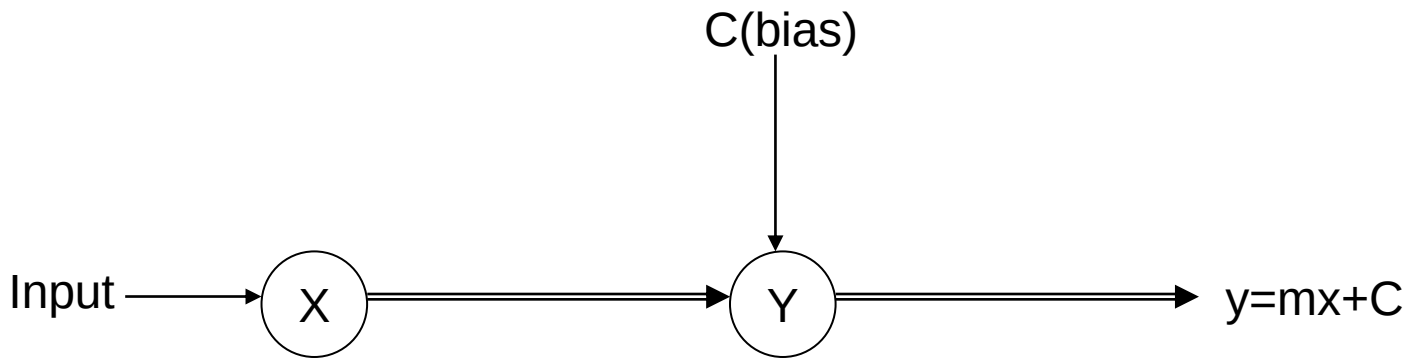
$X=(1,X_1,X_2 \ldots X_i, \ldots X_n)$

Bias is of two types

- Positive bias: increase the net input
- Negative bias: decrease the net input

Why Bias is required?

The relationship between input and output given by the equation of straight line y=mx+c



Threshold

Set value based upon which the final output of the network may be calculated

Used in activation function

The activation function using threshold can be defined as

$$f(net) = \begin{cases} 1 if net \geq \theta \\ -1 if net < \theta \end{cases}$$

Learning rate

Denoted by α.

Used to control the amount of weight adjustment at each step of training

Learning rate ranging from 0 to 1 determines the rate of learning in each time step.

Training Neural Networks

We use data samples to train a neural network.

The main algorithm responsible for training is the backpropagation algorithm.

Back Propagation learns by iteratively processing a set of training data (samples).

Back Propagation

STEP ONE: initialize the weights and biases.

The weights in the network are initialized to random numbers from the interval [-1,1].

Each unit has a BIAS associated with it

The biases are similarly initialized to random numbers from the interval [-1,1].

STEP TWO: feed the training sample.

Back Propagation

STEP THREE: Propagate the inputs forward; we compute the net input and output of each unit in the hidden and output layers.

STEP FOUR: back propagate the error.

STEP FIVE: update weights and biases to reflect the propagated errors.

STEP SIX: terminating conditions.

Deep Learning

Neural networks with more than one hidden layers are called “Deep Neural Networks”.

Performing learning tasks with deep neural networks is called “Deep Learning”.

In this workshop we are going to learn doing deep learning tasks using PyTorch.

PyTorch is an optimized Deep Learning framework based on Python and Torch and is mainly used for applications using GPUs and CPUs.

In this chapter we discussed some basics about neural networks and deep learning so you will be ready to work with this framework.

Thanks For Your Attention!

Feel free to ask any questions