SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

SCHOOL OF ELECTRICAL AND ELECTRONICS

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS

UNIT – IV – DEEP LEARNING – SECA4002

UNIT IV   DIMENTIONALITY REDUCTION

Optimization in deep learning– Non-convex optimization for deep networks- Stochastic Optimization Generalization in neural networks- Spatial Transformer Networks- Recurrent networks, LSTM Recurrent Neural Network Language Models- Word-Level RNNs & Deep Reinforcement Learning - Computational & Artificial Neuroscience.

4.1 Optimization in Deep Learning:

In Deep Learning, with the help of loss function, the performance of the model is estimated/ evaluated. This loss is used to train the network so that it performs better. Essentially, we try to minimize the Loss function. Lower Loss means the model performs better. The Process of minimizing any mathematical function is called Optimization.

Optimizers are algorithms or methods used to change the features of the neural network such as weights and learning rate so that the loss is reduced. Optimizers are used to solve optimization problems by minimizing the function

The Goal of an Optimizer is to minimize the Objective Function(Loss Function based on the Training Data set). Simply Optimization is to minimize the Training Error.

4.1.1 Need for Optimization:

- Prescence of Local Minima reduces the model performance
- Prescence of Saddle Points which creates Vanishing Gradients or Exploding Gradient Issues
- To select appropriate weight values and other associated model parameters
- To minimize the loss value (Training error)

4.2. Convex Optimization:

Convex optimization is a kind of optimization which deals with the study of problem of minimizing convex functions. Here the optimization function is convex function.

All Linear functions are convex, so linear programming problems are convex problems. When we have a convex objective and a convex feasible region, then there can be only one optimal solution, which is globally optimal.

Definition: A set $C \subseteq Rn$ is convex if for $x, y \in C$ and any $\alpha \in [0, 1]$,

$$\alpha x + (1 - \alpha)y \in C$$

Convexity plays a vital role in the design of optimization algorithms. This is largely due to the fact that it is much easier to analyze and test algorithms in such a context.

Consider the given Figure 4.1 given below, select any to points in the region and join them by a straight Line. If the line and the selected points all lie inside the region then we call that region as Convex Region (as Shown in the diagram Figure 4.1)

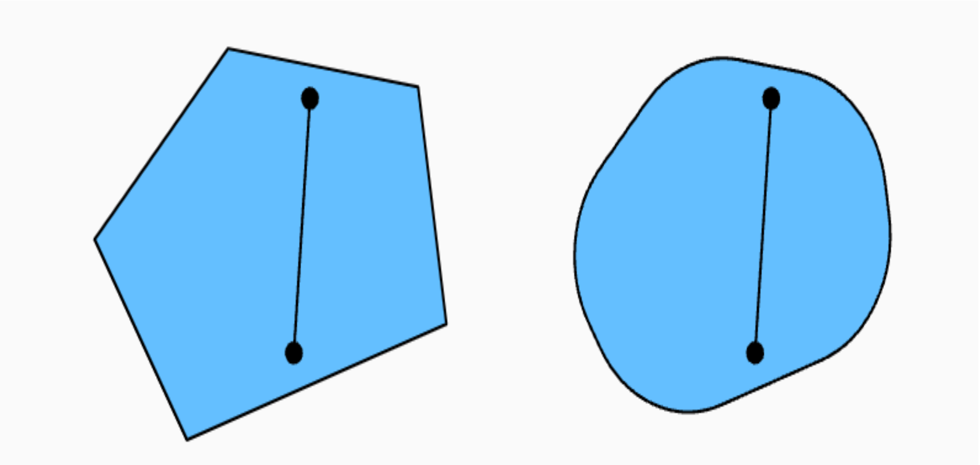

Figure 4.1: Convex Regions

A convex optimization problem is of the form:

$$\min_{x \in D} f(x)$$

subject to

$$g_i(x) \leq 0, \; i = 1, ..., m$$

$$h_j(x) = 0, \; j = 1, ..., r$$

where f and g_i are all convex , and h_j are affine. Any local minimizer of a convex optimization problem is a global minimizer.

4.3. Non-Convex Optimization:

- The Objective function is a non- convex function
- All non-linear problems can be modelled by using non convex functions. (Linear functions are convex)
- It has Multiple feasible regions and multiple locally optimal points.
- There can't be a general algorithm to solve it efficiently in all cases
- Neural networks are universal function approximators, to do this, they need to be able to approximate non-convex functions.

  Refer the figure 4.2 .It shows Non Convex Region

4.3.1. How to solve non-convex problems?

- ✶ Stochastic gradient descent
- ✶ Mini-batching
- ✶ SVRG
- ✶ Momentum

4.3.2. Reasons For Non-Convexity:

- Presence of many Local Minima
- Prescence of Saddle Points
- Very Flat Regions
- Varying Curvature

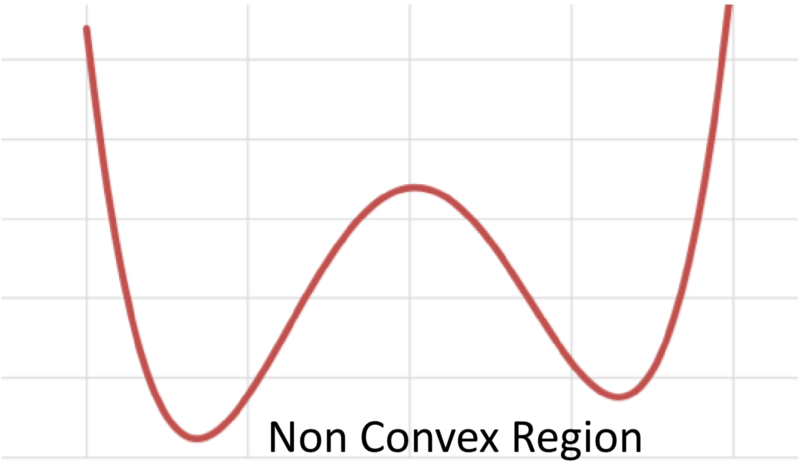

Figure 4.2: Convex Regions

4.4. Spatial Transform Network [STN]:

Spatial Transformer Network (SSTN) helps to crop out and scale-normalizes the appropriate region, which can simplify the subsequent classification task and lead to better classification performance. The Spatial Transformer Network contains three parts Namely, Localization, Grid Generator and Sampler. These Networks are used for performing Transformations such as Cropping, Rotation etc on the given input images.

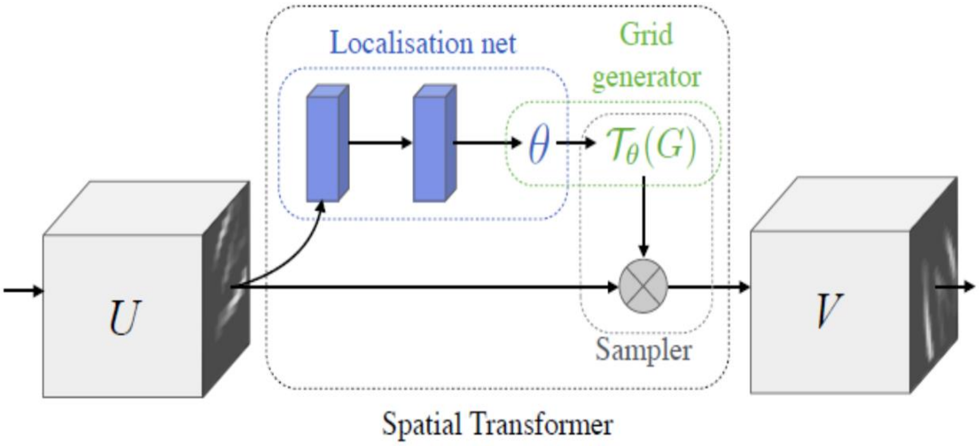

Figure 4.3: Convex Regions

Localisation Net:

With **input feature map *U***, with width *W*, height *H* and *C* channels, **outputs are *θ***, the parameters of transformation *Tθ*. It can be learnt as affine transform

Grid Generator:

Suppose we have a regular grid G, this G is a set of points with **target coordinates (xt_i, yt_i)**. Then we **apply transformation *T θ* on G**, i.e. *T θ*(*G*). After *Tθ*(*G*), a set of points with **destination coordinates (xt_i, yt_i) is outputted**. These points have been altered based on the transformation parameters. It can be Translation, Scale, Rotation or More Generic Warping depending on how we set *θ* as mentioned above.

Sampler:

Based on the new set of coordinates (xt_i, yt_i), we generate a transformed output feature map *V*. This *V* is translated, scaled, rotated, warped, projective transformed or affined, whatever. It is noted that STN can be applied to not only input image, but also intermediate feature maps.

➢ **STN is a mechanism that rotates or scales an input image or a feature map in order to focus on the target object and to remove rotational variance** .

❑ One of the most notable features of STNs is their modularity (the module can be injected into any part of the model) and their ability to be trained with a single backprop algorithm without modification of the initial model.

4.4.1. Advantages:

- Helps in learning explicit spatial transformations like translation, rotation, scaling, cropping, non-rigid deformations, etc. of features.
- Can be used in any networks and at any layer and learnt in an end-to-end trainable manner.
- Provides improvement in the performance of existing models.

4.5. Recurrent Neural Networks:

- RNNs are very powerful, because they combine two properties:
  - Distributed hidden state that allows them to store a lot of information about the past efficiently.
  - Non-linear dynamics that allows them to update their hidden state in complicated ways.
- With enough neurons and time, RNNs can compute anything that can be computed by your computer.

4.5.1. Need for RNN:

❑ Normal Networks cannot handle sequential data

- ❑ They considers only the current input
- ❑ Normal Neural networks cannot memorize previous inputs

The solution to these issues is the RNN

RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer. We can convert a Feed-Forward Neural Network into a Recurrent Neural Network as given below in figure 4.4.

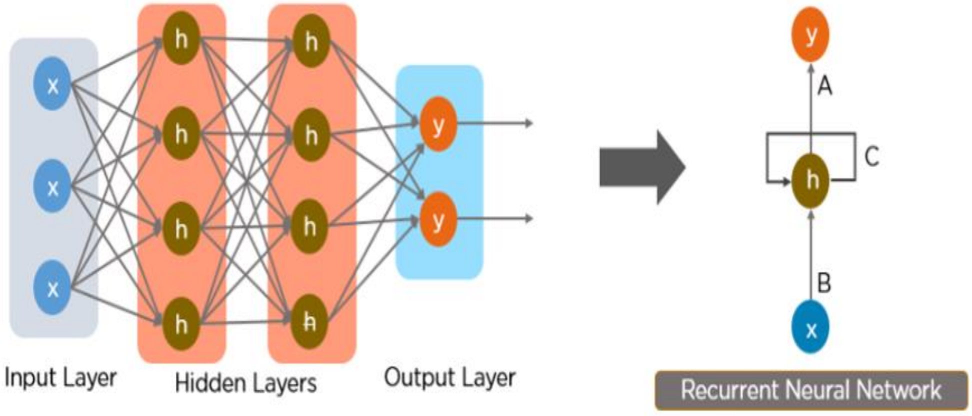

Figure 4.4: Converting a Full network into Recurrent Network

The nodes in different layers of the neural network are compressed to form a single layer of recurrent neural networks. A, B, and C are the parameters of the network. Here, “x” is the input layer, “h” is the hidden layer, and “y” is the output layer. A, B, and C are the network parameters used to improve the output of the model. At any given time t, the current input is a combination of input at x(t) and x(t-1). The output at any given time is fetched back to the network to improve on the output.(Refer Figures 5A and 5 B)

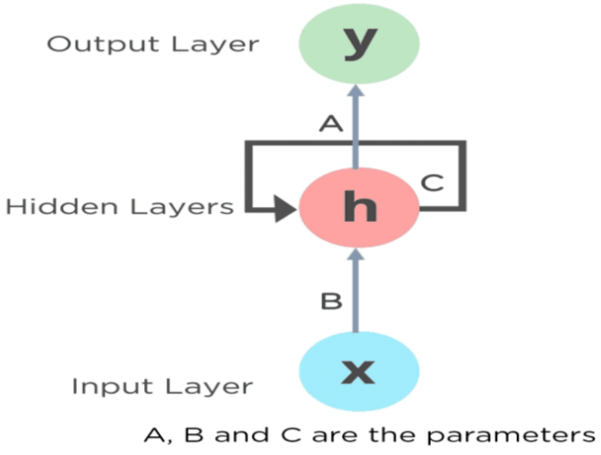

Figure 4.5 A: Recurrent Network

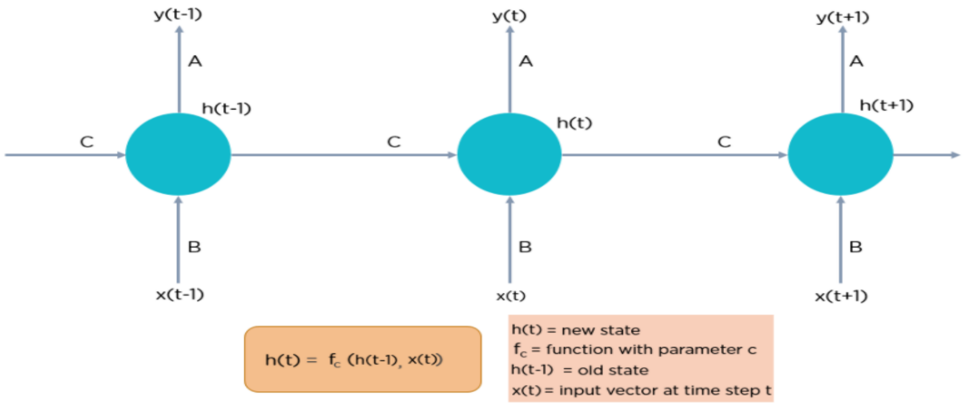

Fig: Fully connected Recurrent Neural Network

Figure 4.5 B: Fully Connected RNN

4.5.2. Providing Input to RNN:

We can specify inputs in several ways:

– Specify the initial states of all the units.

– Specify the initial states of a subset of the units.

– Specify the states of the same subset of the units at every time step.

4.5.3. providing Targets to RNN:

We can specify targets in several ways:

– Specify desired final activities of all the units

– Specify desired activities of all units for the last few steps

• Good for learning attractors

• It is easy to add in extra error derivatives as we backpropagate.

– Specify the desired activity of a subset of the units

4.6. Long Short Term Memory Network’s (LSTM):

LSTMs are a special kind of RNN — capable of learning long-term dependencies by remembering information for long periods is the default behavior. All RNN are in the form of a chain of repeating modules of a neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

LSTMs also have a chain-like structure, but the repeating module is a bit different structure. Instead of having a single neural network layer, four interacting layers are communicating extraordinarily.

Hochreiter & Schmidhuber (1997) solved the problem of getting an RNN to remember things for a long time (like hundreds of time steps). They designed a memory cell using logistic and linear units with multiplicative interactions. Information gets into the cell whenever its “write” gate is on. The information stays in the cell so long as its “keep” gate is on. Information can be read from the cell by turning on its “read” gate.(Refer Figure 4.6 – shown Below)

To preserve information for a long time in the activities of an RNN, we use a circuit that implements an analog memory cell.

- A linear unit that has a self-link with a weight of 1 will maintain its state.
- Information is stored in the cell by activating its write gate.
- Information is retrieved by activating the read gate.
- We can backpropagate through this circuit because logistics are had nice derivatives.

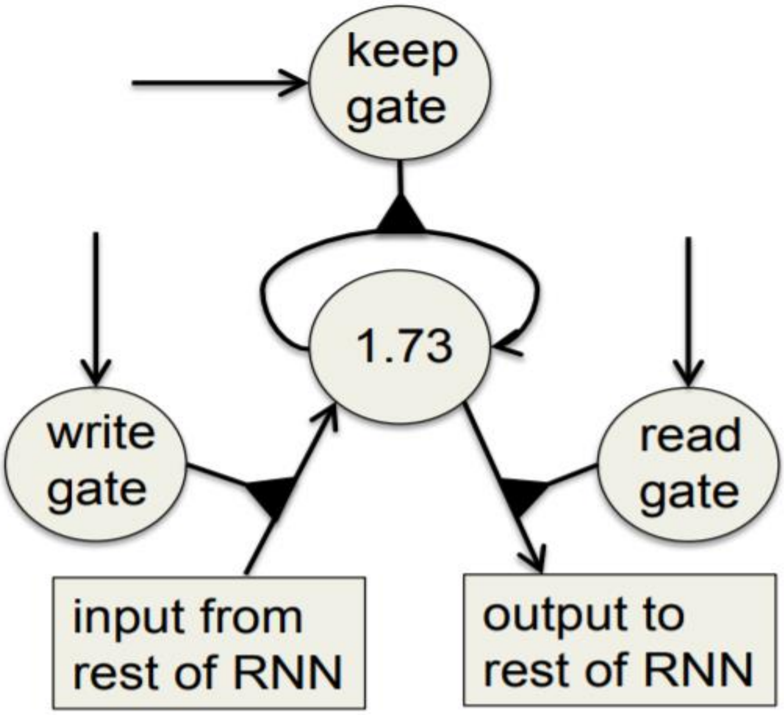

Figure 4.6 B: Read, Keep, Write gate of an LSTM

4.6.1. Steps Involved in LSTM Networks:

Step 1: Decide how much past data it should remember

The first step in the LSTM is to decide which information should be omitted from the cell in that particular time step. The sigmoid function determines this. It looks at the previous state (ht-1) along with the current input xt and computes the function.

Step 2: Decide how much this unit adds to the current state

In the second layer, there are two parts. One is the sigmoid function, and the other is the tanh function. In the sigmoid function, it decides which values to let through (0 or 1). tanh function gives weightage to the values which are passed, deciding their level of importance (-1 to 1).

Step 3: Decide what part of the current cell state makes it to the output

The third step is to decide what the output will be. First, we run a sigmoid layer, which decides what parts of the cell state make it to the output. Then, we put the cell state through tanh to push the values to be between -1 and 1 and multiply it by the output of the sigmoid gate.

4.6.2. Applications of LSTM include:

- Robot control
- Time series prediction
- Speech recognition
- Rhythm learning
- Music composition
- Grammar learning
- Handwriting recognition

4.7. Computational and Artificial Neuro-Science:

Computational neuroscience is the field of study in which mathematical tools and theories are used to investigate brain function.

The term “computational neuroscience” has two different definitions:

1. using a computer to study the brain
2. studying the brain as a computer

Computational and Artificial Neuroscience deals with the study or understanding of how signals are transmitted through and from the human brain. A better understanding of How decision is made in human brain by processing the data or signals will help us in developing Intelligent algorithms or programs to solve complex problems. Hence, we need to understand the basics of Biological Neural Networks (BNN).

4.7.1. The Biological Neurons:

The human brain consists of a large number, more than a billion of neural cells that process information. Each cell works like a simple processor. The massive interaction between all cells and their parallel processing only makes the brain’s abilities possible. Figure 1 represents a human biological nervous unit. Various parts of biological neural network(BNN) is marked in Figure 4.7.

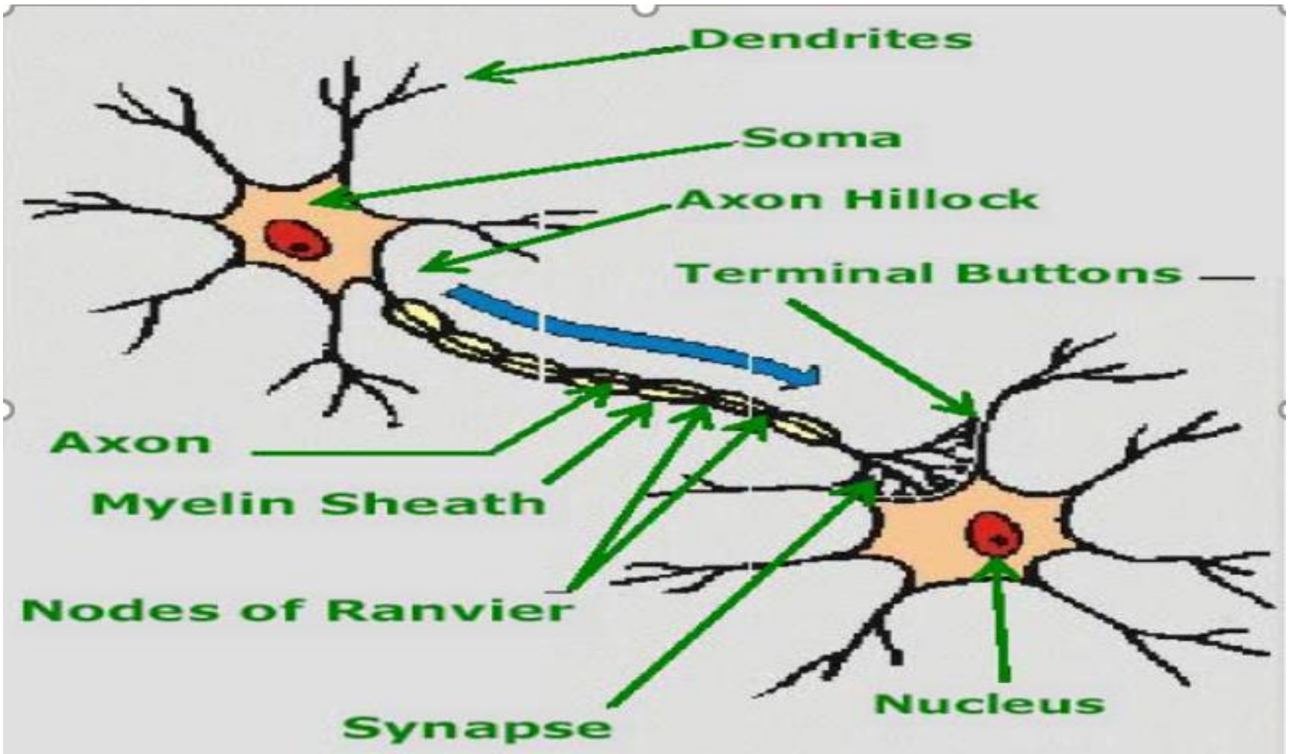

Figure 4.7: Biological Neural Network

Dendrites are branching fibres that extend from the cell body or soma.

Soma or cell body of a neuron contains the nucleus and other structures, support chemical processing and production of neurotransmitters.

Axon is a singular fiber carries information away from the soma to the synaptic sites of other neurons (dendrites ans somas), muscels, or glands.

Axon hillock is the site of summation for incoming information. At any moment, the collective influence of all neurons that conduct impulses to a given neuron will determine whether or n ot an action potential will be initiated at the axon hillock and propagated along the axon.

Myelin sheath consists of fat-containing cells that insulate the axon from electrical activity. This insulation acts to increase the rate of transmission of signals. A gap exists between each myelin sheath cell along the axon. Since fat inhibits the propagation of electricity, the signals jump from one gap to the next.

Nodes of Ranvier are the gaps (about 1 μm) between myelin sheath cells. Since fat serves as a good insulator, the myelin sheaths speed the rate of transmission of an electrical impulse along the axon.

Synapse is the point of connection between two neurons or a neuron and a muscle or a gland. Electrochemical communication between neurons take place at these junctions.

Terminal buttons of a neuron are the small knobs at the end of an axon that release chemicals called neurotransmitters.

Information flow in a neural cell

The input/output and the propagation of information are shown below.

4.7.2. Artificial neuron model

An artificial neuron is a mathematical function conceived as a simple model of a real (biological) neuron.

- The McCulloch-Pitts Neuron
  This is a simplified model of real neurons, known as a Threshold Logic Unit.
- A set of input connections brings in activations from other neuron.
- A processing unit sums the inputs, and then applies a non-linear activation function (i.e. squashing/transfer/threshold function).
- An output line transmits the result to other neurons.

4.7.3. Basic Elements of ANN:

Neuron consists of three basic components –weights, thresholds and a single activation function. An Artificial neural network(ANN) model based on the biological neural sytems is shown in figure 4.8.

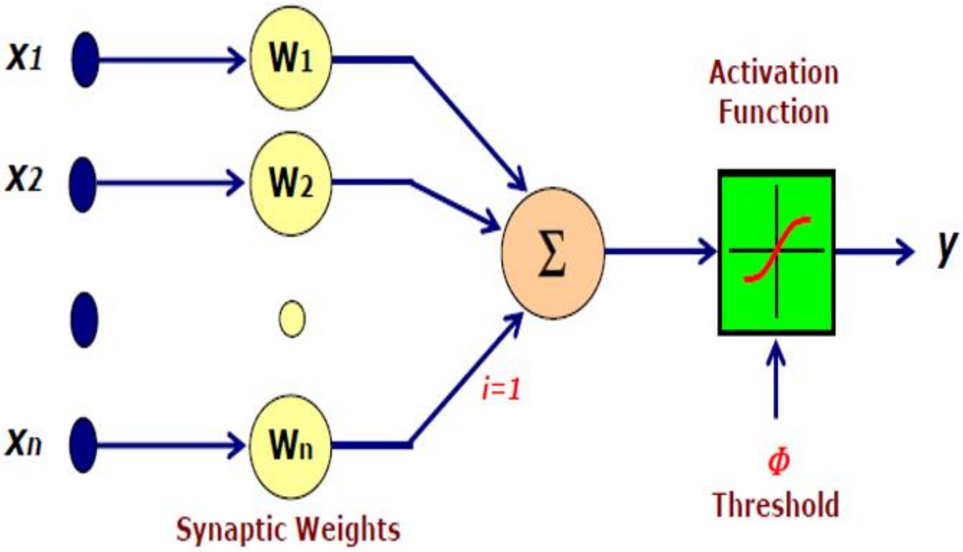

Figure 4.8: Basic Elements of Artificial Neural Network

The goal of computational neuroscience is to explain how electrical and chemical signals are used in the brain to represent and process information**.** It explains the biophysical mechanisms of computation in neurons, computer simulations of neural circuits, and models of learning.

4.7.4. Applications of Computational Neuro Science:

- Deep Learning, Artificial Intelligence and Machine Learning
- Human psychology
- Medical sciences
- Mental models
- Computational anatomy
- Information theory

Reference Books:

1. B. Yegnanarayana, “Artificial Neural Networks” Prentice Hall Publications.
2. Simon Haykin, “Artificial Neural Networks”, Second Edition, Pearson Education.
3. Laurene Fausett, “Fundamentals of Neural Networks, Architectures, Algorithms and Applications”, Prentice Hall publications.
4. Cosma Rohilla Shalizi, Advanced Data Analysis from an Elementary Point of View, 2015.
5. 2. Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
6. 3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
7. 4. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

Note: For further reference, kindly refer the class notes, PPTs, Video lectures available in the Learning Management System (Moodle)

**************************** ALL THE BEST **