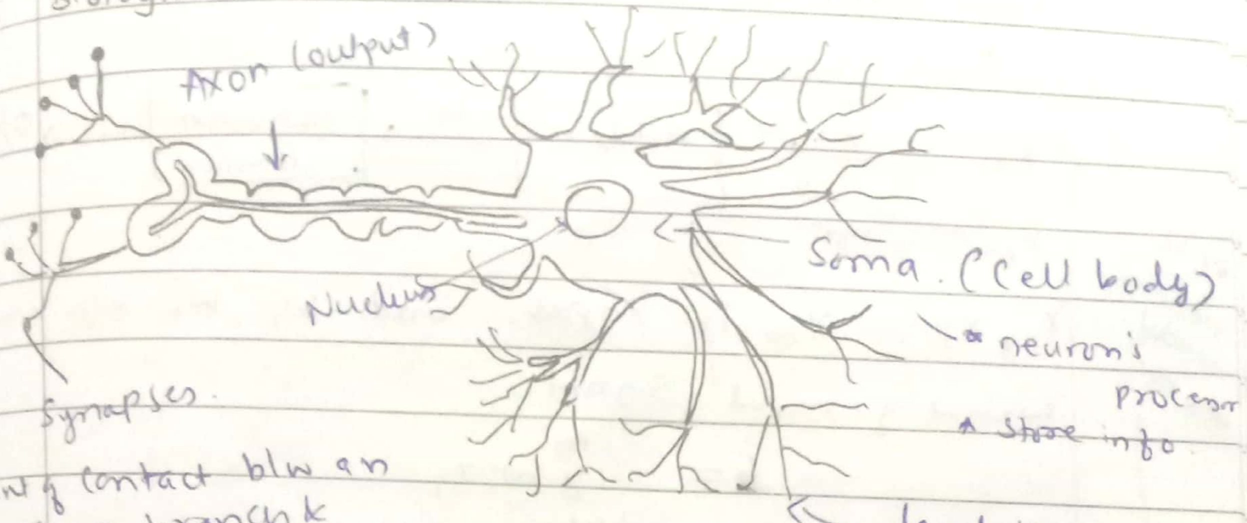
ML unit- 6

Biological Neuron ☆☆



point of contact b/w an axon branch receiving a neuron.

Stimuli - Electric impulse

Hebb's Rule

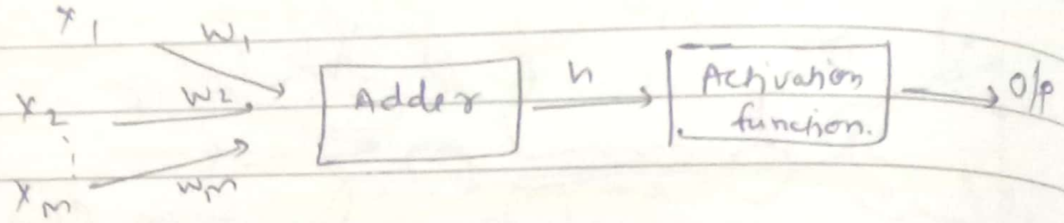
* If two cells or system of cells that are repeatedly active at same time will tend to become associated. so that activity in one facilitates activity in the other.

* When a cell persistently activates another nearby cell the connection b/w the two cell become stronger.

Artificial Neural Network (ANN)

- Artificial Neural Network is a parallel computingal system consist of many simple processing element connected to perform a particular task.

→ Neural network is a network of artificial neurons for solving AI problem.



mcp neuron model **

$x_1, x_2 .. x_m$ is input, and $w_1, w_2, - w_n$ are weight of input signal.

$$h = \sum_{i=1}^{m} W_i x_i$$

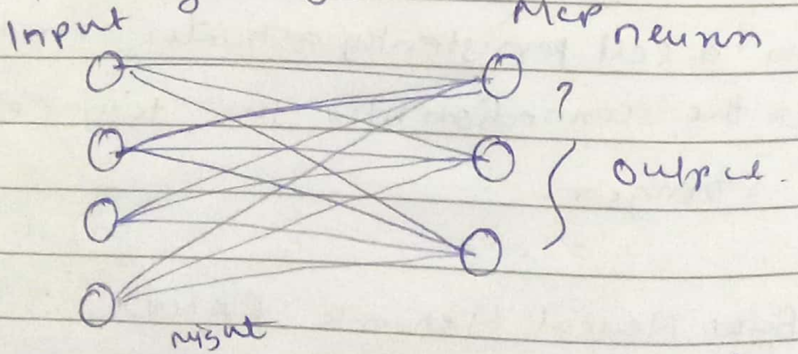
and threshold value θ.

So, if $h > \theta$ neuron fire output 1

$h \leq \theta$ not fire. output 0

Perceptron (Single layer Neural Network)

* perceptron is a Collection of Mcp neuron put together with a set of input and corresponding weight.



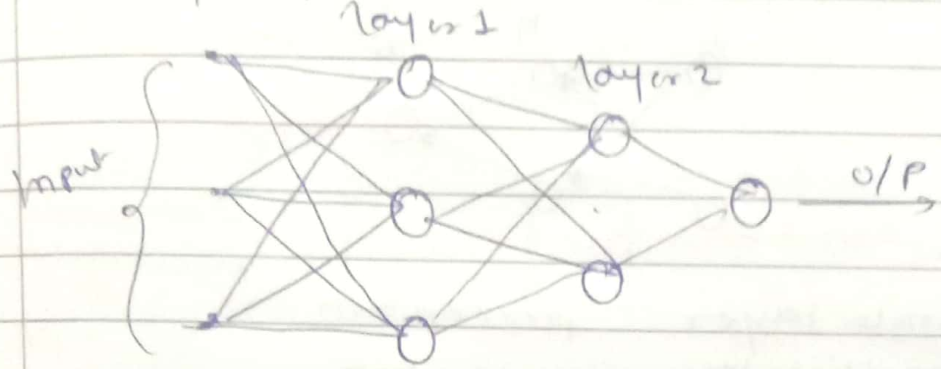
* The neuron might get same input but diff. weight attached to them.

* ~~If threshold~~ If threshold, then fire neuron.

* Bias Input

It is Set to fixed random arbitary value Such that it is non zero. (-1)

Multilayer perceptron.



+ Creating multiple layer of perceptron network to solve complex problem.

Perceptron consist of 4 parts.

1. ~~AA~~ Input value, layer
2. Weight and bias
3. Net sum
4. Activation function.

Shallow Neural Network - less than 3 layer

Deep Neural Network - more than 3 layer

Neural Network (NN) Architecture

① Perceptron.

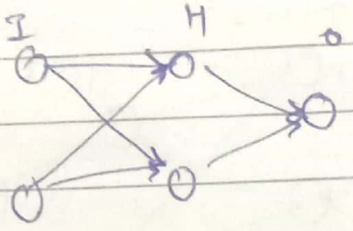
→ No hidden layer, for classification.

(2) Feed forward

→ data flow in one direction.

→ neuron of same layer are independent.

→ Backpropagation is used to minimise error.



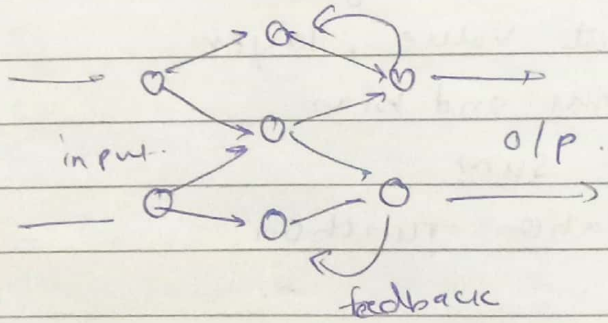
① Single layer perceptron

② multiple layer perceptron.

~~③ Recurrent Neural Network~~.

(3) Feedback.

→ Signal can travel in both direction in feedback

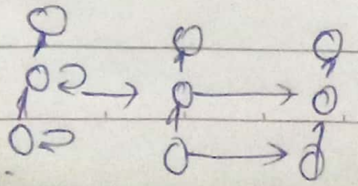


Recurrent neural network (RNN)

→ output can be directed back to itself or to another processing element or both.

① singlayer recurrent network

② multilayer recurrent network.

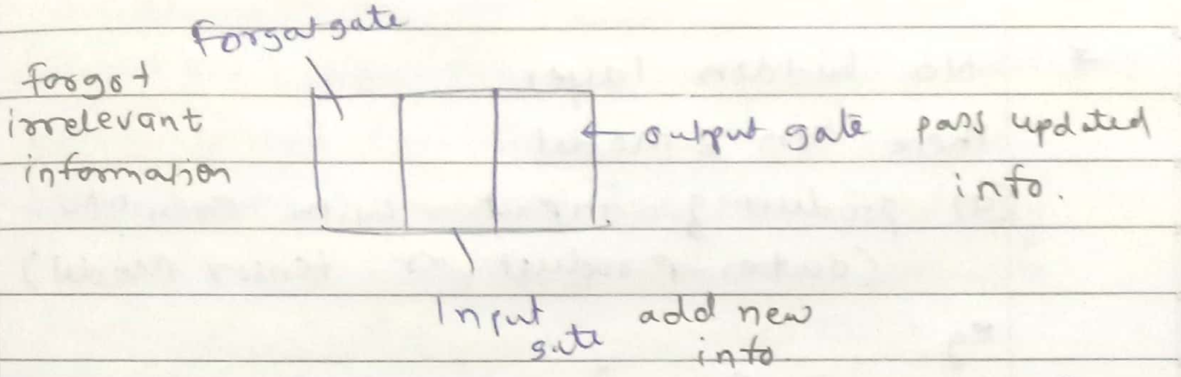


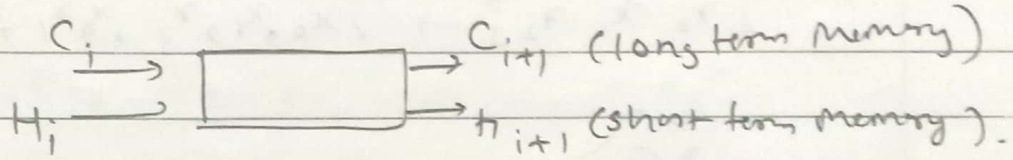
Long / short Term Memory (LSTM) Neural Network

+ Same as RNN But with memory

+ LSTM network treat neuron as a memory cell

* RNN cannot remeber data from a long time ago but LSTM can.



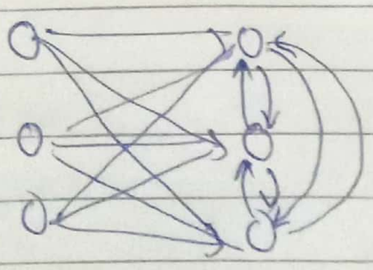


Competitive Neural Network.

⇒ output neuron Connected also

→ all output neuron participate in competition for getting fired or activated.

→ The neuron have Maximum net input will be winner and set 1 and other 0.



- used in unsupervised learning

Page No.

Date

Functional link Artificial Neural Network (FLAN

- Generating additional inputs for feedforward network by applying functions to the original raw inputs.

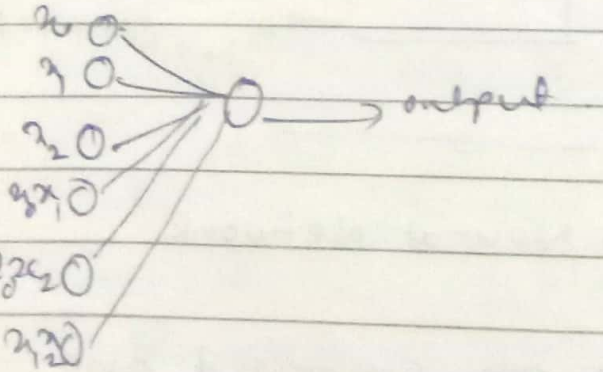
* No hidden layer.

There can 2 Model

(a) product of input with each other
(outer-product or tensor Model)

eg.

$\{x_0\ x_1\ x_2\} \rightarrow \{x_0, x_1, x_2, x_0x_1, x_1x_2, x_2x_0\}$



(b) using univariate function.
(Functional Expansion model)

* It apply nonlinear activation function.

eg

$\{x_0\ x_1\ x_2\} \rightarrow \{x_0, x_1, \sin(x_1), x, \cos(x_1, x_1 ...$

Activation function.

1)

→ Activation function decides whether the artificial neuron would fire or not for a given set of input.

→ It help to solve the complex non linear model.

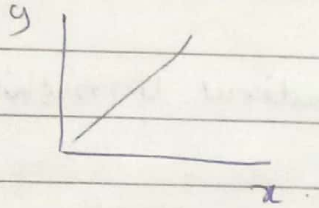
→ without activation function output signal will just be a linear function.

→ Non-linear complex function mapping between the inputs and required variable

Types

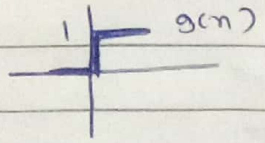
(a) Identity function.

$g(x) = x$.



(b) Binary step function.

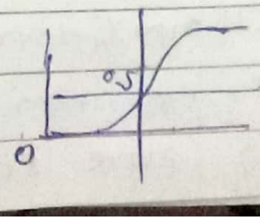
$g(x) = 1$ when $x > \theta$ $(\theta = 0)$
otherwise 0.



(c) Sigmoid /logistic function.

→ Range of output 0 to 1.

$$S(x) = \frac{1}{1+e^{-x}}$$



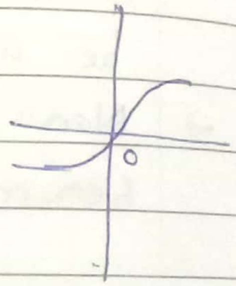
Adv. (1) Non linear
(2) fixed output range

disadv. (1) vanishing gradient → towards end g the cuve. Very less changegy wrt
(2) output is not zero Centred.

(d) TanH | Hyperbolic Tangent function.

Range -1 to 1

Adv. (1) Non linear
(2) fixed output range
(3) zero centred output.

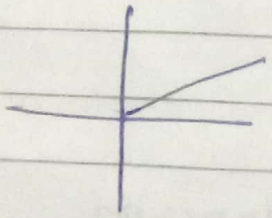


$$\text{~~Tab~~ TanH}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}.$$

disadv.
(1) gradient vanishing.

(e) ReLu (Rectified linear unit).

$$f(x) = \begin{cases} 0 & \text{for } x \le 0 \\ x & \text{for } x > 0 \end{cases} = \max(0, x)$$



adv
(1) non linear
(2) Not vanishing gradient
(3) More efficient

disadv
(1) not zero centred

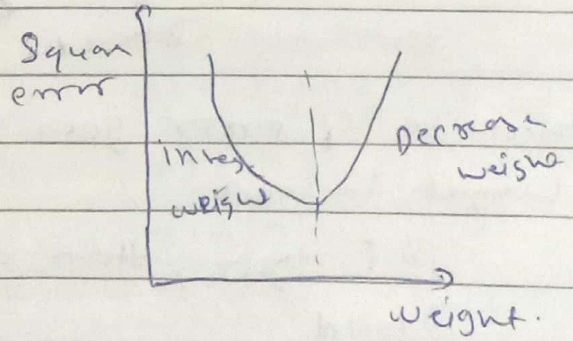
learning process

(1) The number of layers in the Network
(2) The direction of Signal flow
(3) The number of nodes in each layer
no. of input = no. of features
(4) The value of weight attached with each interconnection between neurons.

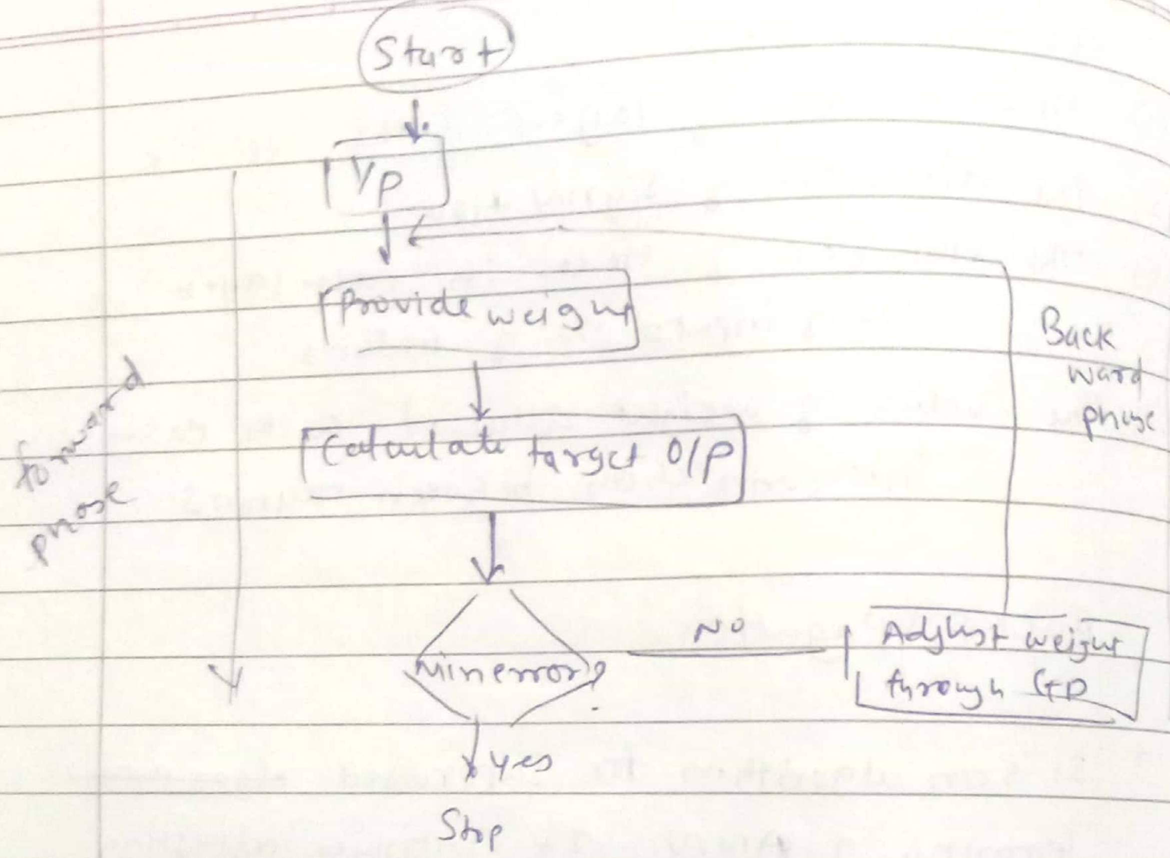
Backpropagation.

- It is an algorithm for supervised ~~algorithm~~, learning of ANN. It Continues adjusting the weight of the connected neuron with an objective to reduce the deviation of the output signal from the target output.

- It Consist of iteration known as epochs.
- We need to read Global loss minimum



- gradient descent is used.



$$E = \frac{1}{2}(y_{out} - y_{target})^2.$$

$$w_{new} = w_{old} - \eta \times \frac{\partial E}{\partial w_{old}}$$

η = Learning rate.

* error goes down when weigh increase

$\frac{\partial E}{\partial w_{old}} < 0$, then increase weight

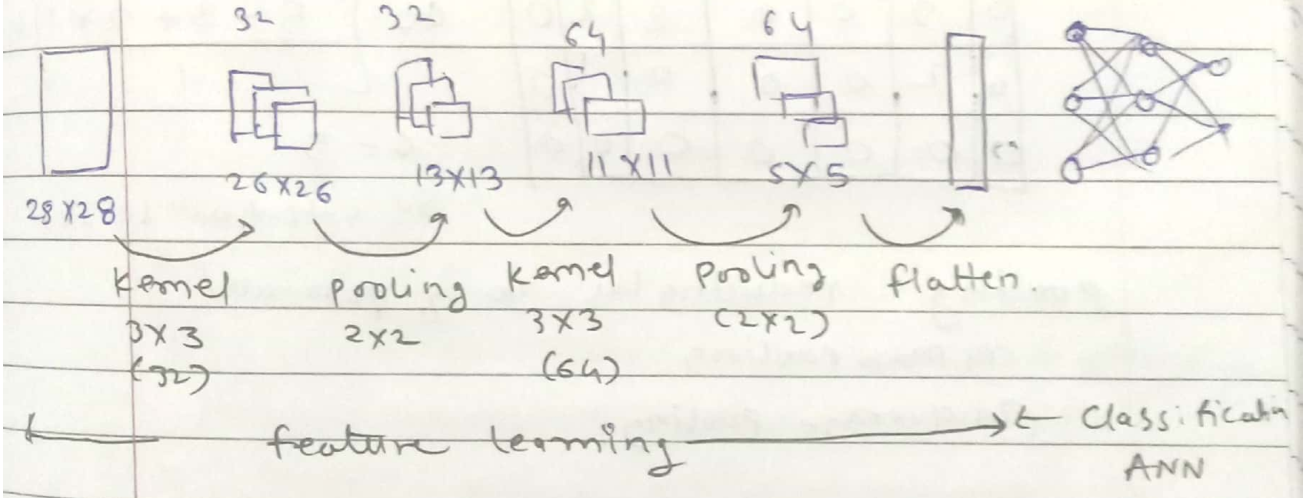
* Otherwise if error goes up when increase weight increase.

$\frac{\partial E}{\partial w_{old}} > 0$, then decrease weight

Convolution Neural Network. (CNN)

→ used for More Complex images.

→ each neuron in the next layer is connected only to a group of closely located neurons of the preceding layer, called local receptive field (or a patch)



Kernel (filter) :- used to extract the feature from images. It move over the input data and perform the dot product.

3×0 + 3×1 + 2×2 + 0×2 + 0×2 + 1×0 + 3×0 + 1×1 + 2×2

$$\underbrace{\begin{bmatrix} 3 & 3 & 2 & 1 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ 3 & 1 & 2 & 2 & 3 \\ 2 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 1 \end{bmatrix}}_{5\times5} \cdot \underbrace{\begin{bmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}}_{\text{Kernel } 3\times3} = \overset{\text{(output) feature map}}{\begin{bmatrix} 12 & 12 & 17 \\ 10 & 17 & 19 \\ 9 & 6 & 14 \end{bmatrix}}$$

Stride (S) : no of pixel that the kernel moves over the input matrix.

Padding:

When filter does not fit the Input Image So, pixel is added on the border of Input image data.

0	0	0	0	0	0	0
0	3	3	2	1	0	0
0	0	0	1	3	1	0
0	3	1	2	2	3	0
0	2	0	0	2	2	0
0	2	0	0	0	1	0
0	0	0	0	0	0	0

$$O = \left[\frac{i-k+2p}{s}\right]+1$$

$$O = \left[\frac{5-3+2\times1}{1}\right]+1$$

$$O = 5$$

So, output will be 5x5

pooling : reducing the no. of parameter

(1) Max pooling:

(2) average pooling.

flattening.

- It is transforming the entirepooled feature map matrix into a single column which then fed to the neural network for processing

fully-connected layer:

each node in output layer connect directly to previous node.

+ Here ANN is used

Activation function..

Page No.
Date

Radial Basis function (RBF)

* It is popular Kernel trick to classify non-linear data.

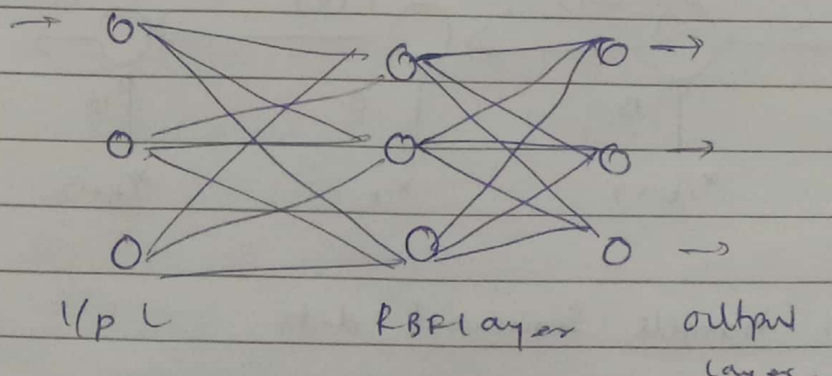
$$K(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$$

$$\frac{1}{2\sigma} = \gamma$$

$$K(x,y) = e^{-\gamma (x-y)^2}$$

RBF Network.

- It is used for approximate function & recognition Pattern (classification)
- It is multilayer feedforward network. It consists of n no. of input neuron & m no. of o/p neuron.



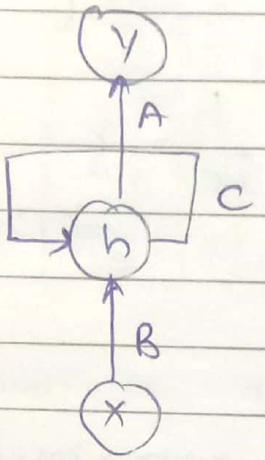
* RBF network never have more than one layer of non linear neuron.

In RBF layer ~~bin~~ non linear data is ~~be~~ converted to linear data. 2D → 3D

Scanned with CamScanner

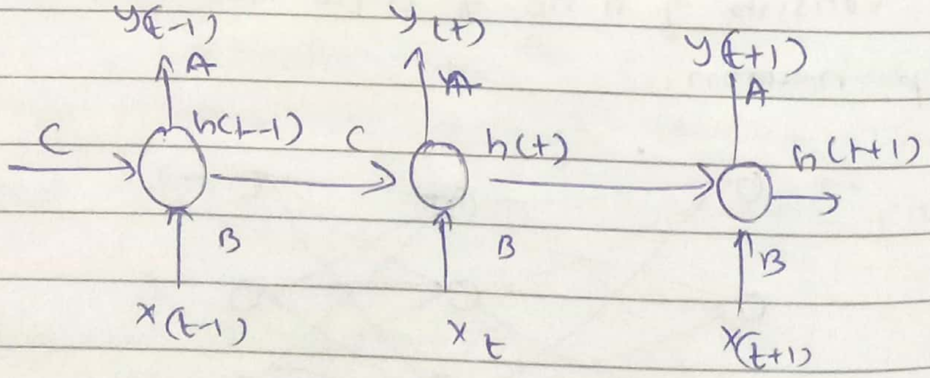
Recurrent Neural Network.

→ Saving output of a particular layer and feeding this back to the input in order to predict out of the layer.



A, B, C are parameters of network.

At any given time t, current input is combination of x_t and x_{t-1}



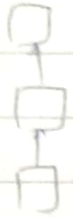
* Can handle Sequential data.

Application

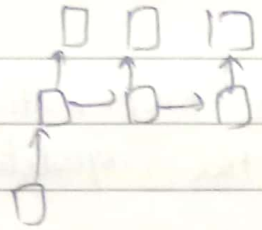
① NLP

② Image captioning

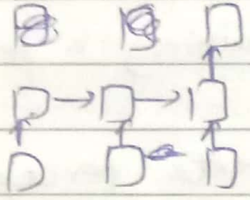
Types. (1) one to one



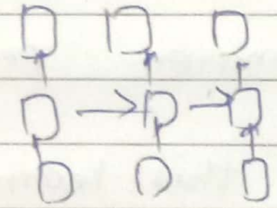
(2) one to many



(3) Many to one

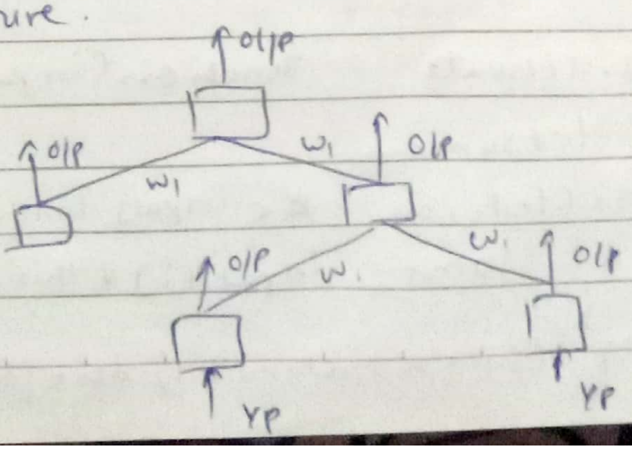


(4) Many to many.



Recursive neural Network.

- -> Deep learning
- -> Applying some set of weights recursively over a structured input, to produce a structured prediction over variable size input structure.
- -> hierarchical data.
- -> tree structure.



Limitation of MLP

(a) Training time: take lots of time as full connected. So input size is large

(b) ~~To~~ The proper functioning of the model depend on the quality of the training.

Training a perceptron

(a) create perceptron Object

```
function Perceptron {no, learningRate=0.000
{
    this. learnc = learningRate;
    this. bias =1;
    this. weights =[ ];
    for (let i=0; i<=no; i++){
        this.weight[i] = Math.random()*2-1;
    }
}
```

(b) Add Activation function

```
this.activate = function(inputs){
    letsum=0
  for(let i=0 ; i< inputs.length; i++){
        sum+= input[i] * this.weight[i];
    }
  if (sum>0){return 1} else{return 0}
```

(c) Creating a Training function.

```
this.train = function(inputs, desired) {
        input.push(this.bias);
        let guess = this.activate (inputs);
        let error = desired - guess;
        if (error != 0){
            for (let i = 0 ; i<inputs.length ; i++){
                this.weights[i] += this.learnc * error *
                                                input[i];
            }
        }
    }
```

Delta learning Rule.

→ This rule state that the modification in weight is the multiplication of error and input.

$$\Delta w = \eta \cdot error.\ input$$

$$w_{new} = w_{old} + \Delta w$$

→ It is independent of the activation function.

→ working same as Gradient descent.

Step.

① initialize weight with random value.

② Apply perceptron. if error then modify weight

$$w_j \leftarrow w_i + \eta (y_t - y_a) \cdot x_{ij}$$

③ Continue until error minimim.