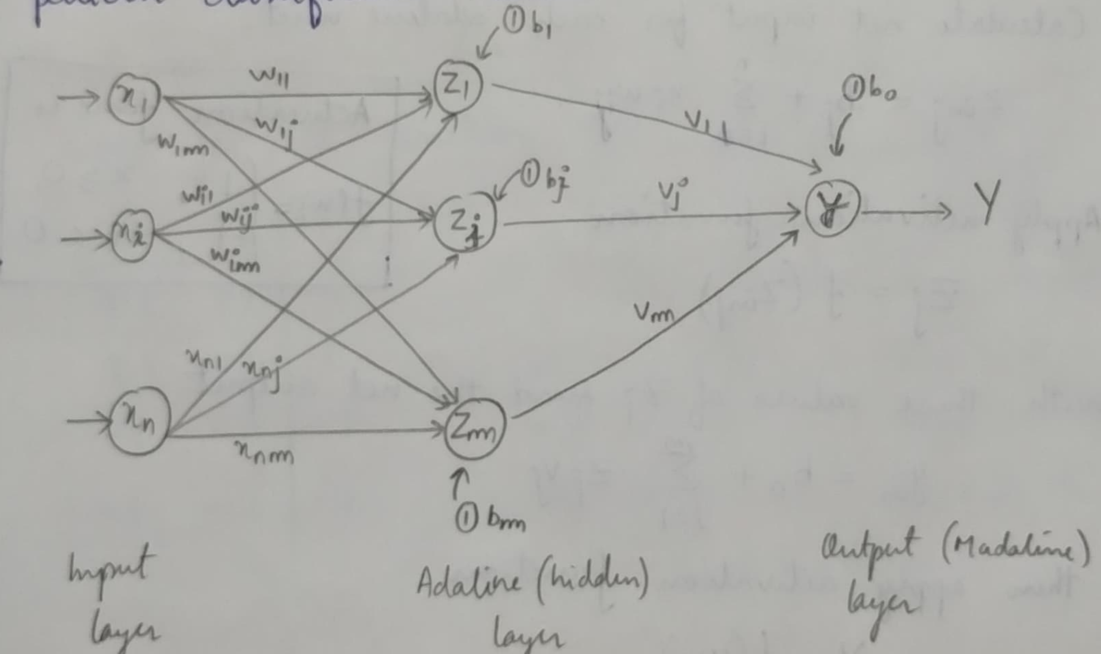
ASSIGNMENT - I

PERUMALLA DHARAN
AP21110010201

(Q1) Explain about MADALINE network, give its training algorithm and implement XOR function using it.

(Ans) MADALINE - Multiple Adaptive Linear Neurons model consists of many ADALINES in parallel with a single output unit whose value is based on certain selection rules.

One of the earliest neural networks, developed by Bernard Widrow and Ted Hoff in 1960. It is particularly suited for pattern classification tasks.



- The weights b/w the hidden layer and output layer are all same and kept as it is (it is not changed) while weights b/w input layer and hidden layer are changed (adjusted) in the training.

Training Algorithm –

PERUMALLA DHARAN
AP21110010201

Step 0 – Initialize the weights and bias randomly small values but for weights b/w adaline layer and output layer and bias is taken as 0.5 to get $Y=1$.
Also set learning rate α.

Step 1 – When stopping condition is false, perform steps 2–3

Step 2 – For each bipolar input $s=(x_1, x_2)$ and output (t), perform steps 3–7.

Step 3 – Activate input layer units from 1 to n.

$$x_i = s_i$$

Step 4 – Calculate net input for each adaline unit.

$$z_{inj} = b_j + \sum_{i=1}^{n} x_i w_{ij}$$

Activation func is

$$f(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

Step 5 – Apply activation function

$$Z_j = f(z_{inj})$$

Step 6 – With these values of Z_j find the net output

$$y_{in} = b_0 + \sum_{j=1}^{m} Z_j v_j$$

then apply activation function

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

Step 7 – Calculate the errors and update the weight.

① If $t=y$, no weight updation is required.

② If $t \neq y$ and $t=+1$,

update weights of z_j, where net input is closest to 0.

PERUMALLA DHARAN
AP21110010201

$$b_j(new) = b_j(old) + \alpha(1 - Z_{inj})$$

$$w_{ij}(new) = w_{ij}(old) + \alpha(1 - Z_{inj})\, x_i$$

(3) If $t \neq y$ and $t = -1$,
update weights on Z_k, whose net input is positive

$$w_{ik}(new) = w_{ik}(old) + \alpha(-1 - Z_{ink})\, x_i$$

$$b_k(new) = b_k(old) + \alpha(-1 - Z_{ink})$$

Step 8 - Stopping Condition - if there is no change in weight or no of iterations has been performed then stop else continue.

Implementing XOR function

Table for XOR function

x_1	x_2	t
+1	+1	-1
+1	-1	+1
-1	+1	+1
-1	-1	-1

Let us assume initial weights, bias and learning rates

$w_{11} = 0.05$ $\quad w_{12} = 0.1$

$w_{21} = 0.2$ $\quad w_{22} = 0.2$

$b_1 = 0.3$ $\quad b_2 = 0.15$

$v_1 = v_2 = b_3 = 0.5$

$\alpha = 0.5$

Now for input $x_1 = 1, x_2 = 1$ and output $t = -1$
Net input for hidden units

$$Z_{in1} = b_1 + x_1 w_{11} + x_2 w_{21} = 0.3 + 1 \times 0.05 + 1 \times 0.2 = 0.55$$

$$Z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22} = 0.15 + 1 \times 0.1 + 1 \times 0.2 = 0.45$$

PERUMALLA DHARAN
AP21110010201

Now applying activation function

$Z_1 = f(Z_{in1}) = f(0.55) = 1$

$Z_2 = f(Z_{in2}) = f(0.45) = 1$

Now,

Finding net input for output unit

$y_{in} = b_3 + Z_1 v_1 + Z_2 v_2$

$= 0.5 + 1 \times 0.5 + 1 \times 0.5 = 1.5$

Applying activation function

$Y = f(y_{in}) = f(1.5) = 1$

Since, $t \neq y$, weight updation needed and $t = -1$ and Z_{in1} and Z_{in2} are +ve so updating the weights and bias on both units

$w_{11}(new) = w_{11}(old) + \alpha(t - Z_{in1}) x_1$

$= 0.05 + 0.5(-1 - 0.55)(1) = -0.725$

$w_{12}(new) = w_{12}(old) + \alpha(t - Z_{in2}) x_1$

$= 0.1 + 0.5(-1 - 0.45)(1) = -0.625$

$w_{21}(new) = w_{21}(old) + \alpha(t - Z_{in1}) x_2$

$= 0.2 + 0.5(-1 - 0.55)(1) = -0.575$

$w_{22}(new) = w_{22}(old) + \alpha(t - Z_{in2}) x_2$

$= 0.2 + 0.5(-1 - 0.45)(1) = -0.525$

$b_1(new) = b_1(old) + \alpha(t - Z_{in1}) = 0.3 + 0.5(-1 - 0.55) = -0.475$

$b_2(new) = b_2(old) + \alpha(t - Z_{in2}) = 0.15 + 0.5(-1 - 0.45) = -0.575$

PERUMALLA DHARAN
AP21110010201

for input $x_1 = 1, x_2 = -1$, output $t = 1$

$Z_{in1} = -0.475 + (1)(-0.725) + (-1)(-0.58)$

$= -0.625$

$Z_{in2} = -0.575 + (1)(-0.625) + (-1)(-0.525) = -0.675$

$Z_1 = f(Z_{in1}) = f(-0.625) = -1$

$Z_2 = f(Z_{in2}) = f(-0.675) = -1$

$y_{in} = 0.5 + (-1)(0.5) + (-1)(0.5) = -0.5$

$y = f(y_{in}) = f(-0.5) = -1$

$t \neq y$ and $t = 1$

So, $W_{11}(new) = -0.725 + 0.5 \times (1 + 0.625)(1) = 0.0875$

$W_{21}(new) = -0.575 + 0.5 \times (1 + 0.625)(-1) = -1.39$

$b_1(new) = -0.475 + 0.5(1 + 0.625) = 0.34$

for input $x_1 = -1, x_2 = +1$, and output $t = 1$

$Z_{in1} = 0.34 + (-1)(0.0875) + (1)(-1.39) = -1.1375$

$Z_{in2} = -0.575 + (-1)(-0.625) + (1)(-0.525) = -0.475$

$Z_1 = f(Z_{in1}) = f(-1.1375) = -1$

$Z_2 = f(Z_{in2}) = f(-0.475) = -1$

$y_{in} = 0.5 + (-1)(0.5) + (-1)(0.5) = (-0.5)$

$y = f(y_{in}) = f(-0.5) = -1$

$t \neq y$ and $t = 1$

PERUMALLA DHARAN
AP21110010201

so,

$w_{12}(new) = -0.625 + 0.5(1+0.475)(-1)$

$= -1.3625$

$w_{22}(new) = -0.525 + 0.5(1+0.475)(+1) = 0.2125$

$b_2(new) = -0.575 + 0.5(1+0.475) = 0.1625$

for input $x_1 = -1$, $x_2 = -1$ and output $t = -1$

$z_{in1} = 0.34 + (-1)(0.0875) + (-1)(-1.39) = 1.6375$

$z_{in2} = 0.1625 + (-1)(-1.3625) + (-1)(0.2125) = 1.3125$

$z_1 = f(z_{in1}) = f(1.6375) = 1$

$z_2 = f(z_{in2}) = f(1.3125) = 1$

$y_{in} = b_3 + z_1 v_1 + z_2 v_2 = 0.5 + 0.5 + 0.5 = 1.5$

$y = f(y_{in}) = f(1.5) = 1$

$t \neq y$ and $t = -1$

$w_{11}(new) = 0.0875 + 0.5(-1-1.6375)(-1) = 1.4065$

$w_{12}(new) = -1.3625 + 0.5(-1-1.3125)(-1) = -0.207$

$w_{21}(new) = -1.39 + 0.5(-1-1.6375)(-1) = -0.069$

$w_{22}(new) = 0.2125 + 0.5(-1-1.3125)(-1) = 1.369$

$b_1(new) = 0.34 + 0.5(-1-1.6375) = -0.98$

$b_2(new) = 0.1625 + 0.5(-1-1.3125) = -0.994$

Now our first epoch is completed.

PERUMALLA DHARAN
AP21110010201

EPOCH - 2

$x_1 = 1, x_2 = 1, t = -1$

$z_{in1} = b_1 + x_1 w_{11} + x_2 w_{21}$

$= -0.98 + (1)(1.4065) - 0.069 = 0.3565$

$z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22}$

$= -0.994 - 0.207 + 1.369 = 0.168$

$z_1 = f(z_{in1}) = f(0.3565) = 1$

$z_2 = f(z_{in2}) = f(0.168) = 1$

$y_{in} = b_3 + z_1 v_1 + z_2 v_2 = 1.5$

$y = f(y_{in}) = f(1.5) = 1$

$t \neq y$, and $t = -1$

$w_{11}(new) = w_{11}(old) + \alpha(t - z_{in1}) x_i$

$= 1.4065 + 0.5(-1 - 0.3565)(1) = 0.7285$

$w_{12}(new) = w_{12}(old) + \alpha(t - z_{in2}) x_1$

$= -0.207 + 0.5(-1 - 1.68)(1) = -0.791$

$b_1(new) = b_1(old) + \alpha(t - z_{in1}) = -0.98 + 0.5(-1 - 0.3565) = -1.66$

$w_{21}(new) = w_{21}(old) + \alpha(t - z_{in1}) x_2$

$= -0.069 + 0.5(-1 - 0.3565)(1) = -0.75$

$w_{22}(new) = w_{22}(old) + \alpha(t - z_{in2}) x_2$

$= 1.369 + 0.5(-1 - 1.68)(1) = -0.207$

$b_2(new) = b_2(old) + \alpha(t - z_{in2}) = -0.994 + 0.5(-1 - 1.68) = -1.58$

PERUMALLA DHARAN
AP21110010201

Now, $x_1 = 1, x_2 = -1, t = 1$

$z_{in1} = b_1 + x_1 w_{11} + x_2 w_{21}$

$= -1.66 + (1)(0.7285) + (-1)(-0.75) = -0.1845$

$z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22}$

$= -1.58 + (1)(-0.791) + (-1)(-0.207) = -3.154$

$z_1 = -1$

$z_2 = -1$

$y_{in} = -0.5$

$y = -1$

$t \neq y$ and $t = 1$

$w_{11}(new) = w_{11}(old) + \alpha(t - z_{in1}) x_1$

$= 0.7285 + 0.5(1 + 0.1845)(1) = 1.3205$

$w_{21}(new) = w_{21}(old) + \alpha(t - z_{in1}) x_2$

$= -0.75 + 0.5(1 + 0.1845)(-1) = -1.34$

$b_1(new) = b_1(old) + \alpha(t - z_{in1}) = -1.66 + (1 + 0.1845)0.5 = -1.068$

Now, $x_1 = -1, x_2 = 1, t = 1$

$z_{in1} = -1.068 + (-1)(1.3205) + (1)(-1.34)$

$= -3.728$

$z_{in2} = -1.58 + (-1)(-0.791) + (1)(0.785)$

$= -0.002$

$z_1 = -1$

$z_2 = -1$

$y_{in} = -0.5$

PERUMALLA DHARAN
AP21110010209

$Y = -1$

So, $t \neq y$ and $t = 1$

$w_{12}(new) = w_{12}(old) + \alpha(t - z_{in2})(x_1)$

$= -1.29$

$w_{22}(new) = w_{22}(old) + \alpha(t - z_{in2})\, x_2$

$= \cancel{0.785}\ 1.29$

$b_2(new) = b_2(old) + \alpha(t - z_{in2})$

$= \cancel{} -1.08$

Now, $x_1 = -1$, $x_2 = -1$, $t = -1$

$z_{in1} = -1.068 + (-1)(1.3205) + (-1)(-1.34) = -1.0495$

$z_{in2} = -1.08 + (-1)(-1.29) + (-1)(0.785) = -1.071$

$z_1 = -1$

$z_2 = -1$

$y_{in} = -0.5$

$Y = -1$

$t = y$

No updation required.

EPOCH - 3

PERUMALLA DHARAN
AP21110010201

$x_1 = 1,\ x_2 = 1,\ t = -1$

$z_{in1} = b_1 + x_1 w_{11} + x_2 w_{21} = -1.0865$

$z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22} = -1.083$

$z_1 = -1$

$z_2 = -1$

$y_{in} = -0.5$

$y = -1$

$t = y$, no updation

for, $x_1 = 1,\ x_2 = -1,\ t = 1$

$z_{in1} = b_1 + x_1 w_{11} + x_2 x_{21} = 1.5915$

$z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22} = -3.655$

$z_1 = 1$

$z_2 = -1$

$y_{in} = +0.5$

$y = +1$

$t = y$, no updation

for, $x_1 = -1,\ x_2 = 1,\ t = 1$

$z_{in1} = b_1 + w_{11} x_1 + x_2 w_{21} = -3.728$

$z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22} = +1.501$

$z_1 = -1$

$z_2 = +1$

$y_{in} = 0.5$

$y = 1$

PERUMALLA DHARAN
AP21110010201

$t = y$, no updation

for, $x_1 = -1$, $x_2 = -1$, $t = -1$

$Z_{in1} = b_1 + x_1 w_{11} + x_2 w_{21} = -1.0495$

$Z_{in2} = b_2 + x_1 w_{12} + x_2 w_{22} = -1.701$

$Z_1 = -1$

$Z_2 = -1$

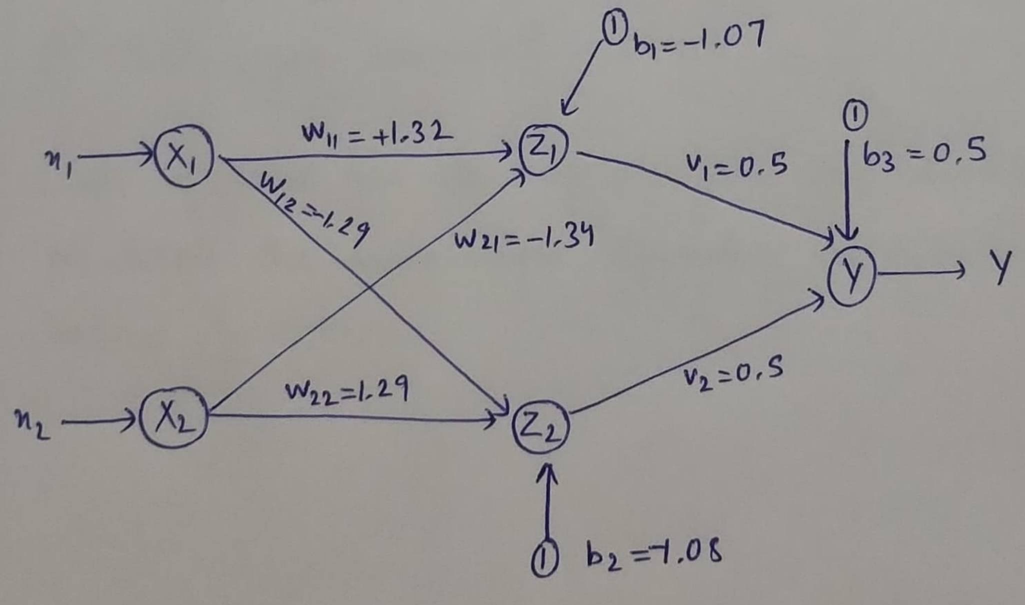
$y_{in} = -0.5$

$Y = -1$

$t = y$, no updation

Here all the weights converges at the end of epoch 3.

So, we stop here.



MADALINE network for XOR function

PERUMALLA DHARAN
AP21110010201

(Q2) With suitable diagrams, explain the basic models or basic architectures of Artificial Neural Networks.

(Ans) Artificial Neural Network is a type of machine learning model which is inspired by the structure and working of our human brain. It can perform various tasks such as classification, clustering, pattern matching, optimization and approximation.

The basic models of ANN are:

① Single layer feedforward network
② Multi layer feed forward network
③ Single node with its own feedback
④ Single layer recurrent network
⑤ Multi layer recurrent network

Each of these has its own strengths and weaknesses. We choose the architecture depending on our task and nature of data.

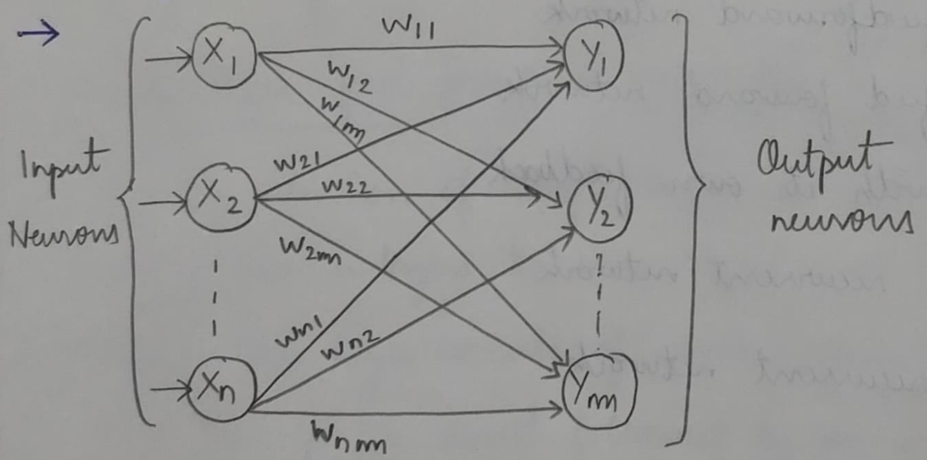
① Single layer feed forward network

PERUMALLA DHARAN
AP21110010201

→ In this network we have only two layers – input layer and output layer.

→ Input layer consists of input neurons each representing a feature of input data. No computation is done here and only input values are passed to the next layer.

→ The output layer may consist of single neuron or multiple neurons. This layer computes the weighted sum which is sum of product of input features and weight of edges along with bias.

→ 

→ Working –

① Each input x_i is multiplied with its corresponding weight w_i.

② Product of all the input neurons are then added along with bias to get weighted sum (let it be Z).

$$Z = x_1 w_1 + x_2 w_2 + \dots + x_n w_n + b$$

[where b is bias]

PERUMALLA DHARAN
AP21110010201

③ Now this weighted sum is passed through an activation function to get the final output.

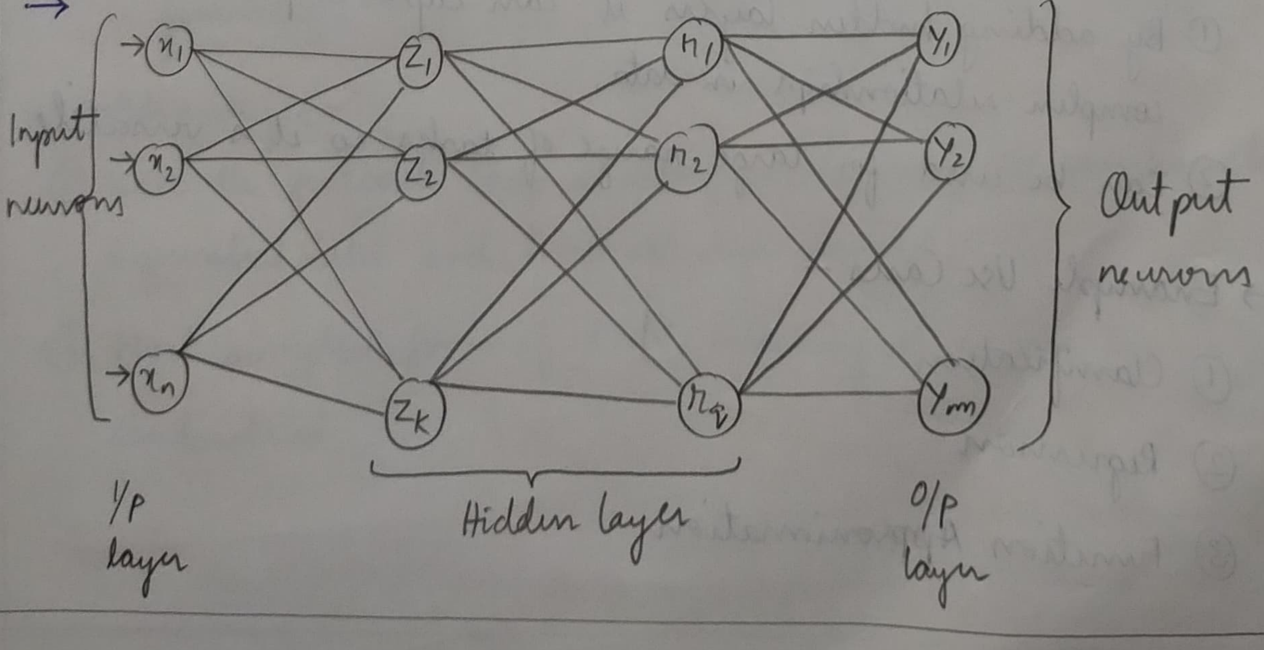
$$Y = f(Z)$$

→ Limitations –
① Can only solve problems which are linearly seperable.
② There are no hidden layers present.

→ Example Use Cases –
① Binary Classification
② Linear Regression

② Multi layer feed forward network

→ It is the extension of single layer feed forward network by adding hidden layers between input and output layers.
→ Each hidden layer consists of neurons that apply weights and bias to the input followed by an activation function.
→



PERUMALLA DHARAN
AP21110010201

→ Working –

① Each input x_i is multiplied with its corresponding weight w_i and are passed to the first hidden layer.

② Each hidden layer neuron computes the weighted sum of the input layer adds bias and then ~~applied~~ applies the activation function. The values obtained becomes the input to the next layer.

③ This is continued for all the hidden layers and then at output layer we get the final resulting output.

→ Limitations –

① Due to presence of many hidden layers it is computationally expensive.

② Risk of overfitting.

③ Internal workings of an learned model cannot be easily understood.

→ Advantages –

① By adding hidden layers it can capture patterns and complex relationships in data.

② Can be used for large range of tasks so it is versatile.

→ Example Use Cases –

① Classification

② Regression

③ Function Approximation

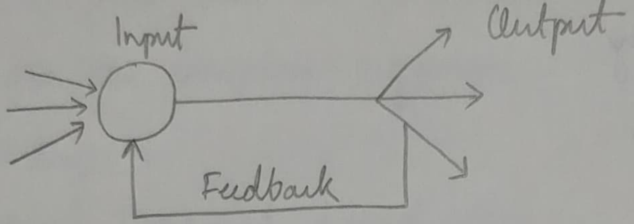
PERUMALLA DHARAN
AP21110010201

3 Single Node with its own feedback

→ It is a basic form of recurrent neural network (RNN) where a single neuron is connected to itself.

→ Recurrent neural networks are feedback networks with closed loops.

→



→ Working –

1. Firstly the neuron recieves the initial input and has an initial state (often initialized to 0 or small random value) and first output is generated and updates its state.

2. Now the neuron recieves the next input but now it combines the new input with the previous state and updates to a new state. (from feedback loop)

3. This is continued for all the inputs and atlast the output is generated.

→ Advantages –

1. Since the feedback loop is acting as memory, it can handle sequential data and temporal dependencies.

2. Most simplest form of RNN making it easier to understand.

PERUMALLA DHARAN
AP21110010201

→ Limitations –

① Limited Capacity

② Vanishing Gradient Problem may arise.

→ Example Use Case –

① Time Series Prediction

② Language Processing

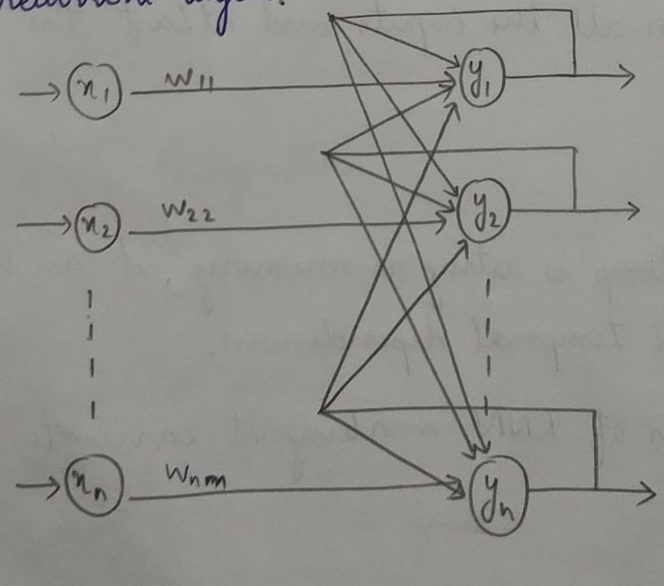
③ Signal Processing.

④ Single Layer Recurrent Network

→ This is a single layer network with a feedback connection in which the processing element output can be directed back to itself or any other element.

→ The layer where each neuron recieves input from the current input and output from previous time step is the recurrent layer.

→



PERUMALLA DHARAN
AP21110010201

→ Working –

① Initially, the initial states of the recurrent neurons are set to 0 or small random values.

② As input sequence is fed to the neuron, each neuron in the recurrent layer updates its state.

③ The outputs that are generated could be used immediately or after the complete sequence.

→ Advantages –

① Captures temporal dependencies

② Simple and effective to implement.

→ Limitations –

① Short term memory

② Computational Complexity

→ Example Use Cases –

① Language modeling

② Time Series Prediction

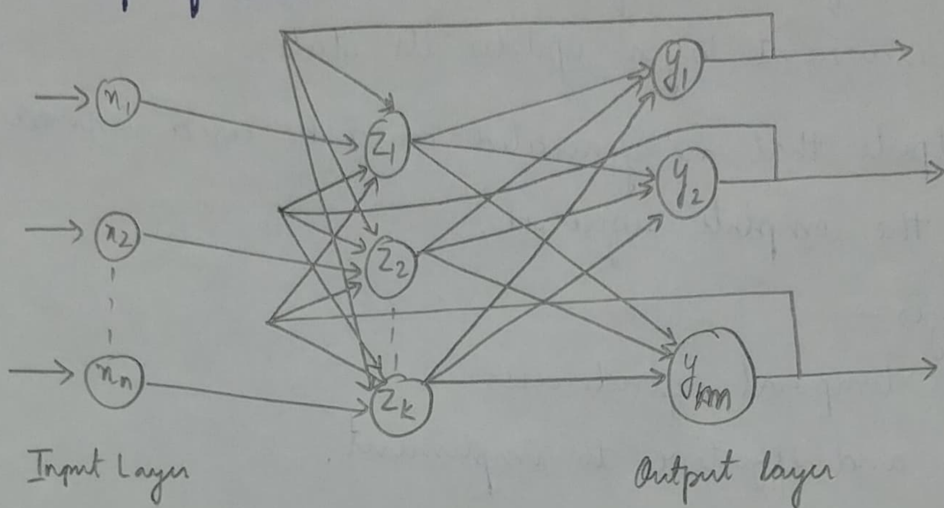
③ Speech Recognition.

⑤ Multi layer Recurrent Network

PERUMALLA DHARAN
AP21110010201

→ It is an advanced form of RNN where multiple layers of recurrent neurons are stacked on top of each other.

→



→ Working –

① Initially, the states of all recurrent neurons are typically initialized to zero or small random values

② As input is fed and is passed through each layer. The feedback connections of each layer of previous state is incoorporated in current processing.

③ The outputs at final layer could be used immediatly or later.

→ Advantages –

① Deeper Representation
② Improved temporal dynamics

→ Limitations –

① Training complexities
② Risk of Overfitting

→ Example Use Case –

① Language translation
② Speech Recognition