Unit - 2

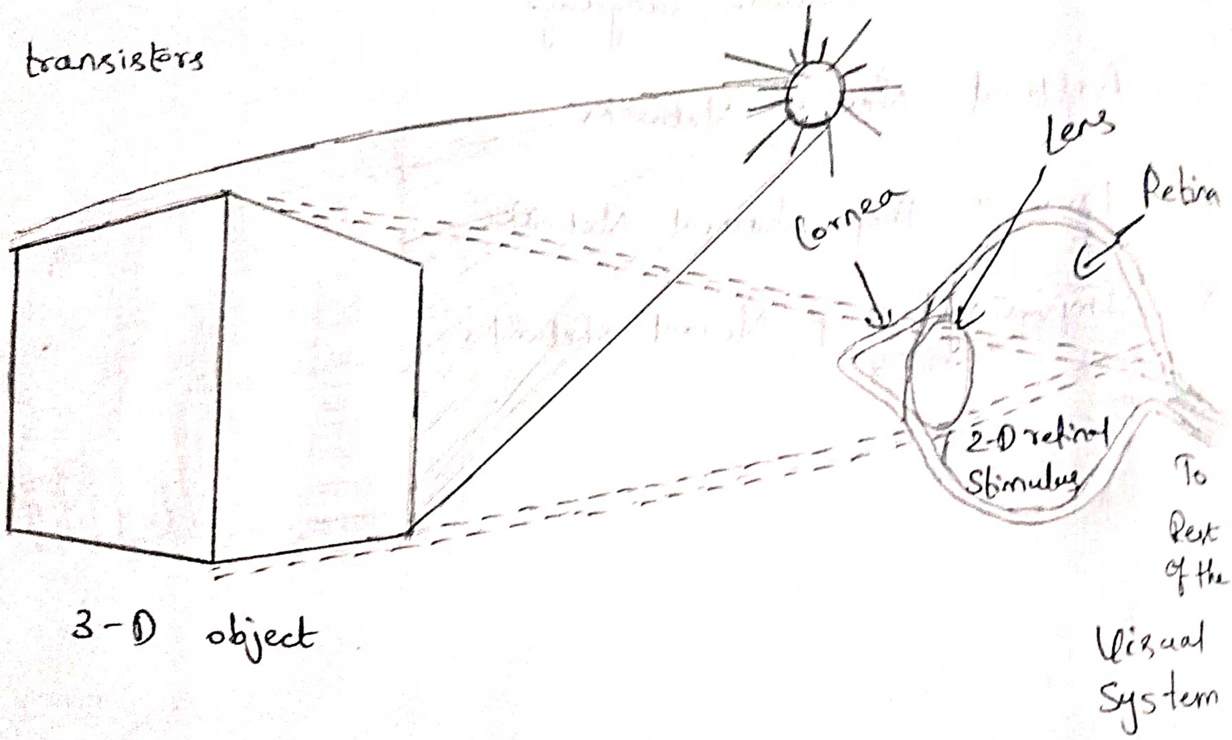
Introduction to Deep Learning

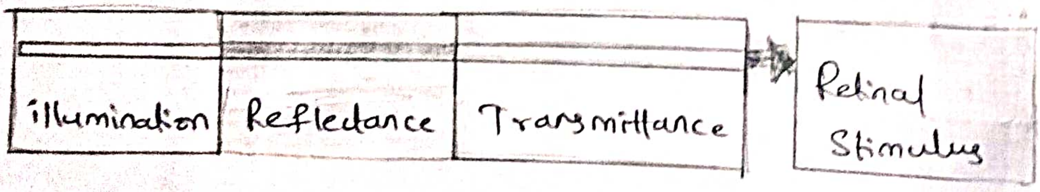
Syllabus

1. Biological and Machine Vision
2. Human and Machine Language
3. Artificial Neural Networks
4. Training Deep Neural Networks
5. Improving Deep Neural Networks.

(1.) Biological and Machine Vision :-

⇒ **Biological Vision** :- runs on an interconnected network of cortical cells and organic neurons. Computer Vision, on the other hand, runs on electronic chips composed of transistors

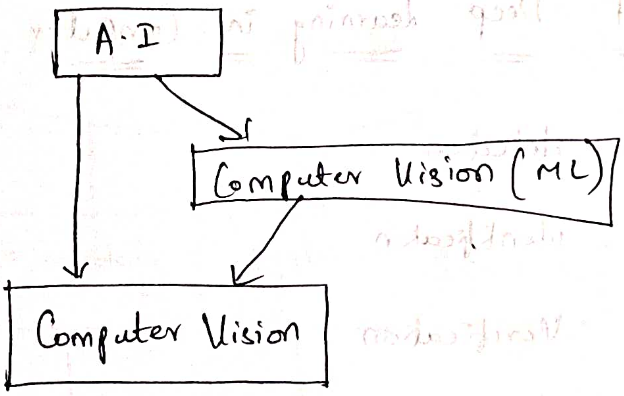


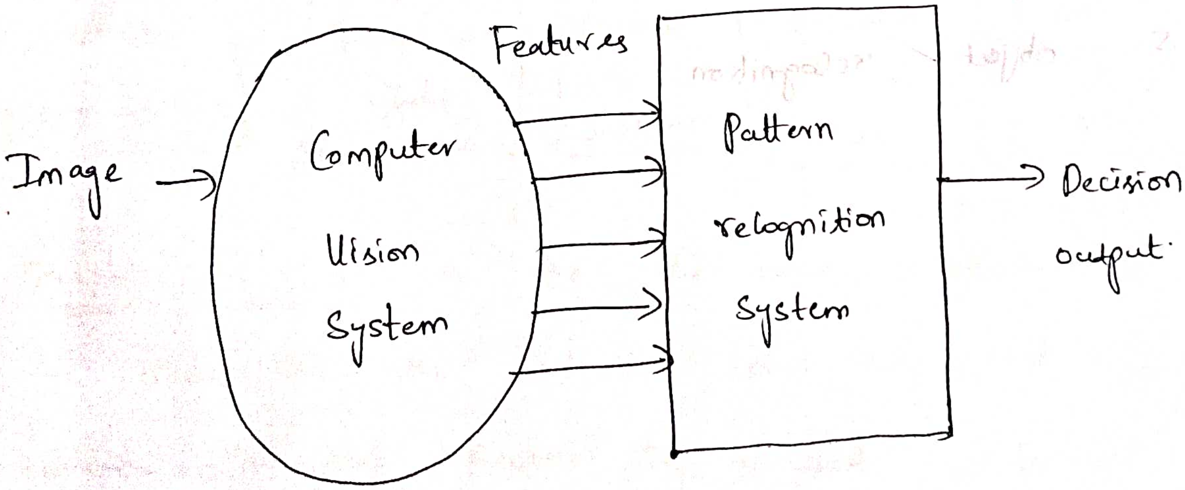


⇒ **Computer Vision (CV) (or) Machine Vision** :- is the Scientific field which defines how machines interpret the Meaning of images and Videos.

⇒ Computer Vision algorithms analyze certain criteria in images and Videos and then apply interpretations to predictive (or) decision making tasks.

⇒ Today, deep learning techniques are most commonly used for Computer Vision mostly we will use (CNN's) Convolutional Neural Networks which are used to provide a Multi layered architecture that allows Neural networks to Focus on the most relevant Features in the image which identifies the image by dividing into Patterns.





⇒ Convolutional Neural Networks : The Foundation of Modern Computer Vision :-

⇒ Modern Computer Vision algorithms are based on Convolutional neural networks (CNN's) which provide a dramatic improvement in performance compared to traditional image Processing algorithms.

⇒ CNN's are typically Used for Computer Vision tasks although text Analytics and audio analytics Can also be Performed.

⇒ One of the First CNN architecture was AlexNet which won the ImageNet Visual recognition challenge in 2012.

⇒ **Uses of Deep learning in Computer Vision :-**

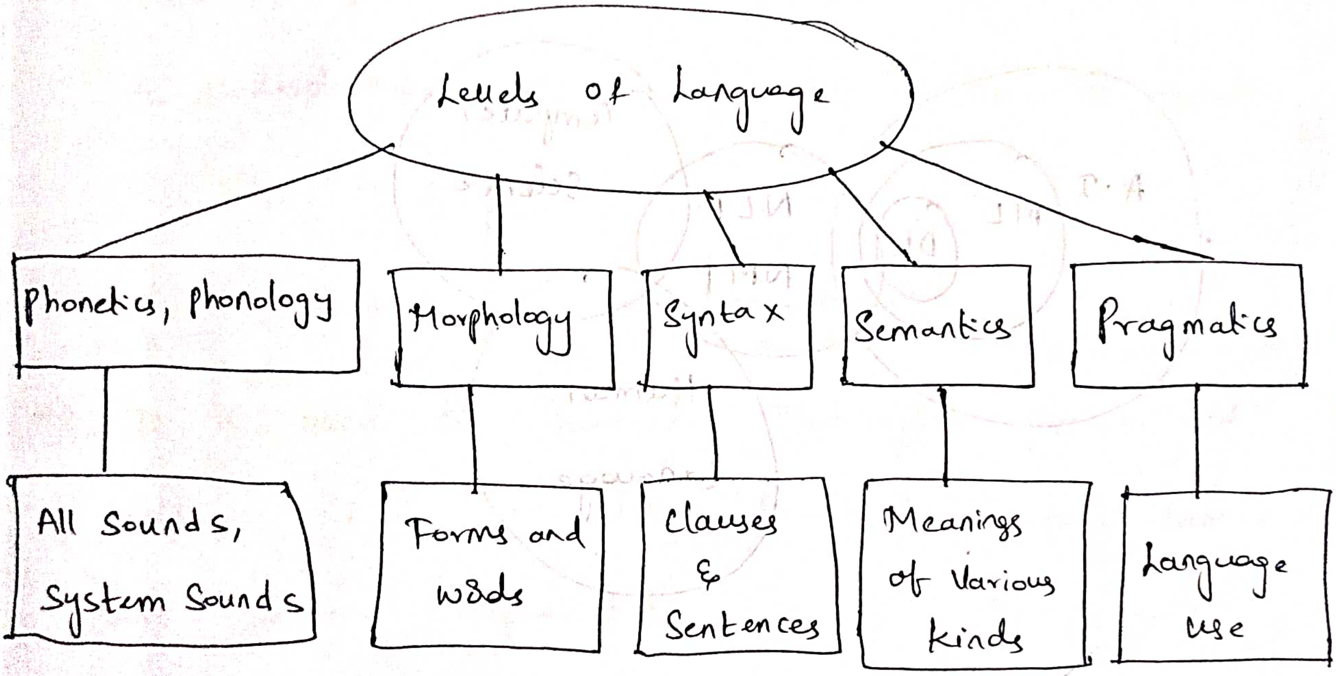
1. object detection
2. object identification
3. object Verification
4. object location
5. object recognition

② **Human and Machine Language :-**

⇒ Human Language allows Speakers to Express thoughts in Sentences which is used for Communication between Each other

⇒ Human Language forms are three types

1. Speech 2. Writing 3. Gestures.

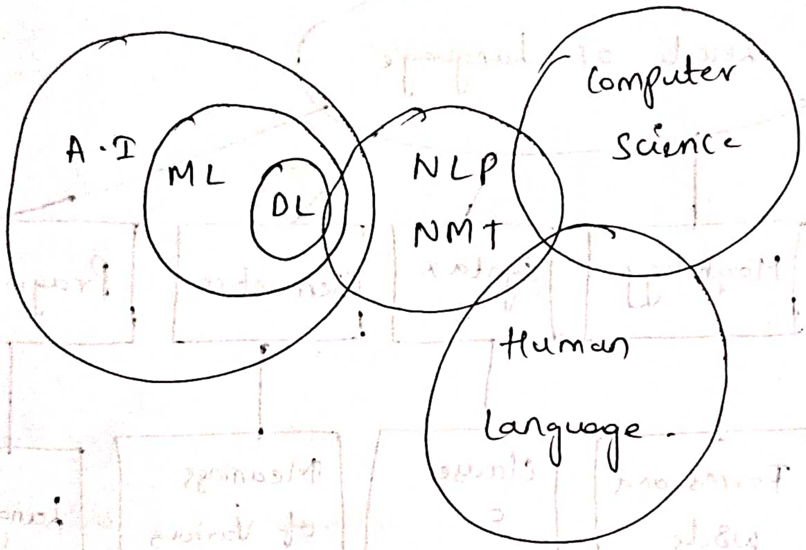


⇒ Machine Language on the other hand which is used to Understand the instructions which are passed by Humans for that purpose. it is used to convert human Language to Machine by Binary Language (0's & 1's) but due to we are using advanced technologies like A.I, M.L, & D.L we have to Perform Complex tasks So we used the Languages Known as NLP & NMT for translation

NLP - Natural Language Processing

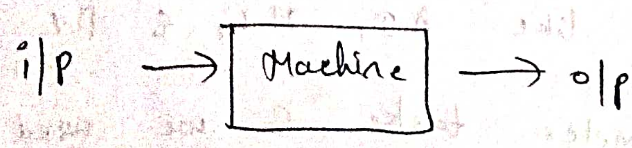
NMT - Nueral Machine Translation.

⇒ NLP (Natural Language Processing) and NMT (Neural Machine translation) which connects to these technologies like A.I, M.L, D.L we can see in below diagram



1.) NLP (Natural Language Processing) :-

⇒ NLP refers to the branch of Computer Science which is used to Understand text and Spoken words in much the Same way human beings can.



⇒ Components of NLP :-

⇒ Basically there are two Components of NLP System

1. NLU (Natural language Understanding)
2. NLG (Natural language Generation)

1. NLU (Natural Language Understanding) :-

⇒ It is used to mapping the given input in Natural Language into Useful representation to Understand and analyzing different aspects of the language.

2. NLG (Natural language Generation) :-

⇒ It is used to Generate Meaningful phrases and Sentences i.e in the form of Natural Language from Internal representation

1. Text Planning
2. Sentence planning
3. Text Realization

Example :-

⇒ In this Example we used to Convert the Source Code to Machine language by following the Phases which involved in the Natural Language Processing.

Source Code $\quad a = b + c * d$

↓

$\boxed{\text{Lexical Analyzer}}$

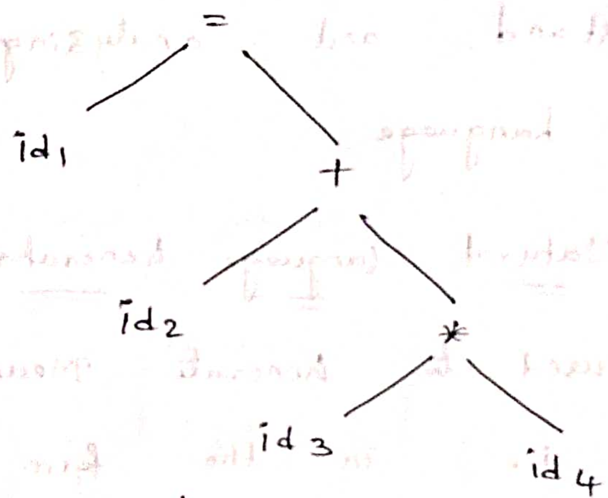
↓

tokens $\quad id_1 = id_2 + id_3 * id_4$

↓

$\boxed{\text{Syntax Analyzer}}$

Syntax tree



↓

$\boxed{\text{Code generator}}$

↓

Back end

Load	id_3
Mul	id_4
add	id_2
Store	id_1

Assembly language

2nd Example :-

=> Parts of Speech (Pos) tagging :-

=> A Pos is a grammatical classification that commonly includes Verbs, adjectives, adverbs, nouns etc.

=> Pos tagging is an important Natural Language Processing application used in Machine translation

Ex :-

John hit the ball.

S → NP VP

VP → V NP

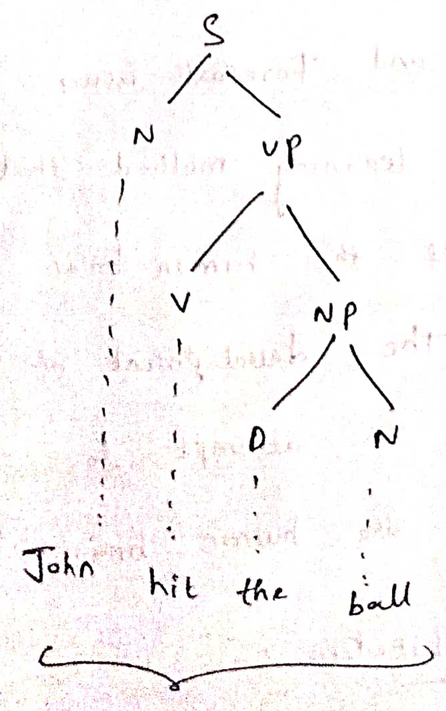
NP → N

NP → DN

NP - Noun phrase

VP - Verb phrase

DN - determinent (the, this, that)



CFG (Context Free grammer)

2) NMT (Neural Machine Translation) :-

⇒ NMT is a language which is used to convert automatically a text language into another form of text language

Examples :-

Telugu → English

English → Hindi

Hindi → Spanish, French Etc.

---

(3) Artificial Nueral Networks (ANN) :-

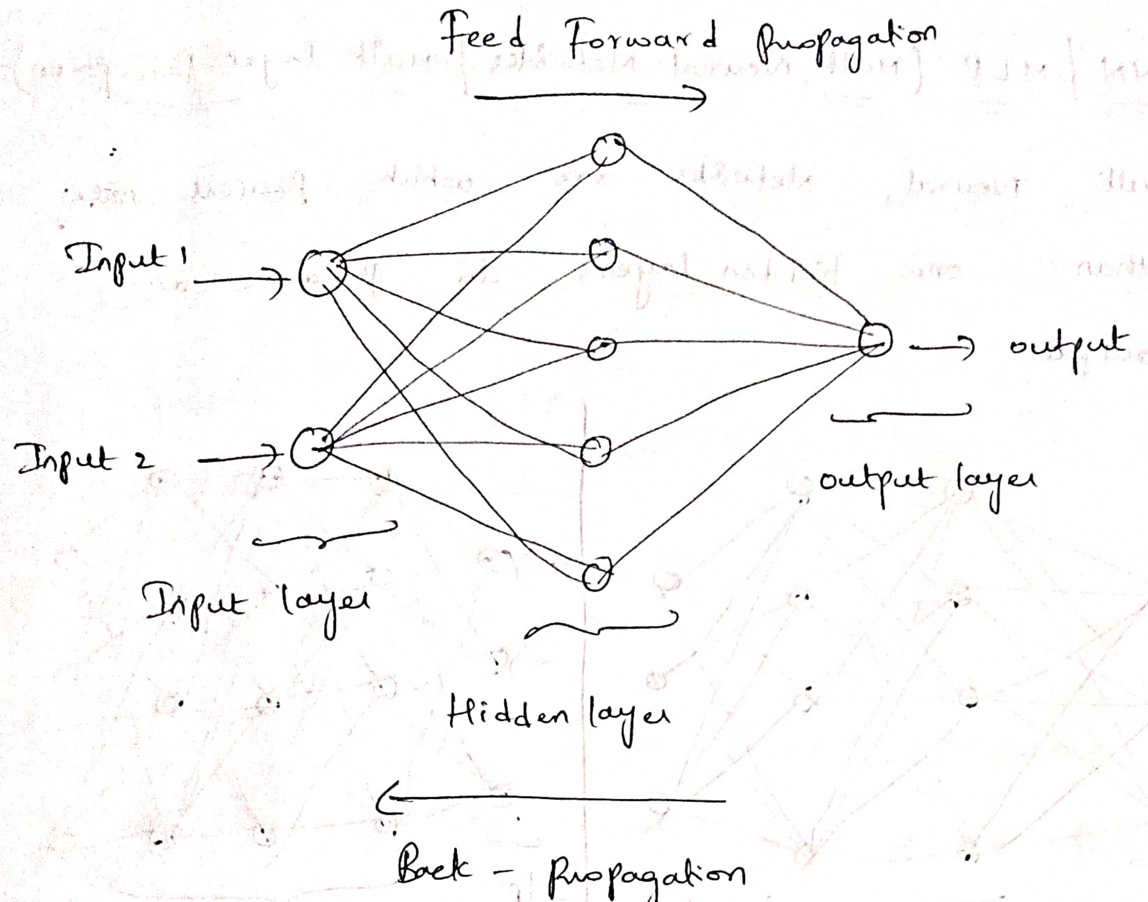
⇒ Artificial Nueral Networks (ANN) are algorithms based on brain function and are used to model complicated patterns and Forecast issues

⇒ The ANN is a deep learning method that arose from the concept of the human brain Biological Neural Networks. The development of ANN was the result of an attempt to replicate the workings of the human brain

⇒ Artificial Neural Networks Architecture :-

⇒ There are three layers in the Network architecture

1. The Input layer
2. The Hidden layer (it can be more than one)
3. The output layer

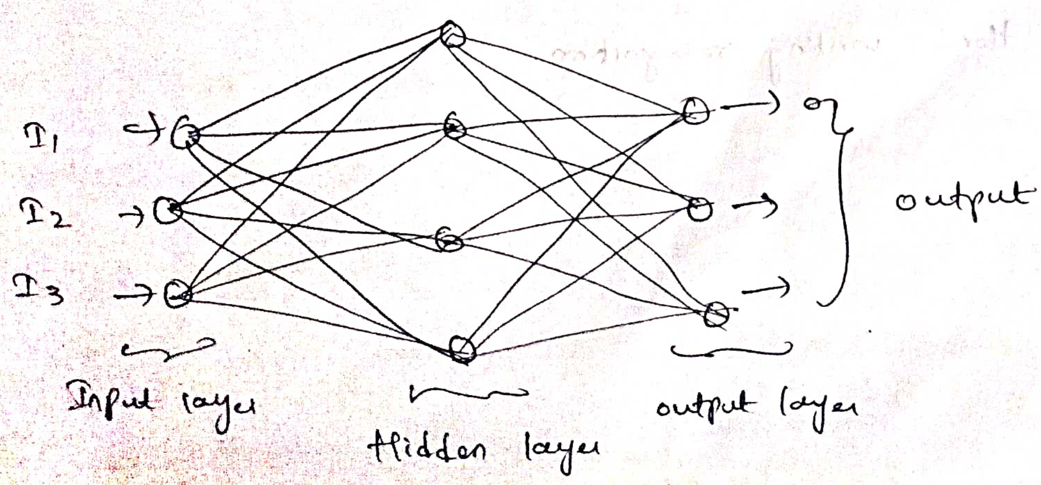


⇒ In ANN Specially there are two types of Networks which we have been used basically

1. SNN (Single Neural Networks)
2. MNN (Multi Neural Networks)

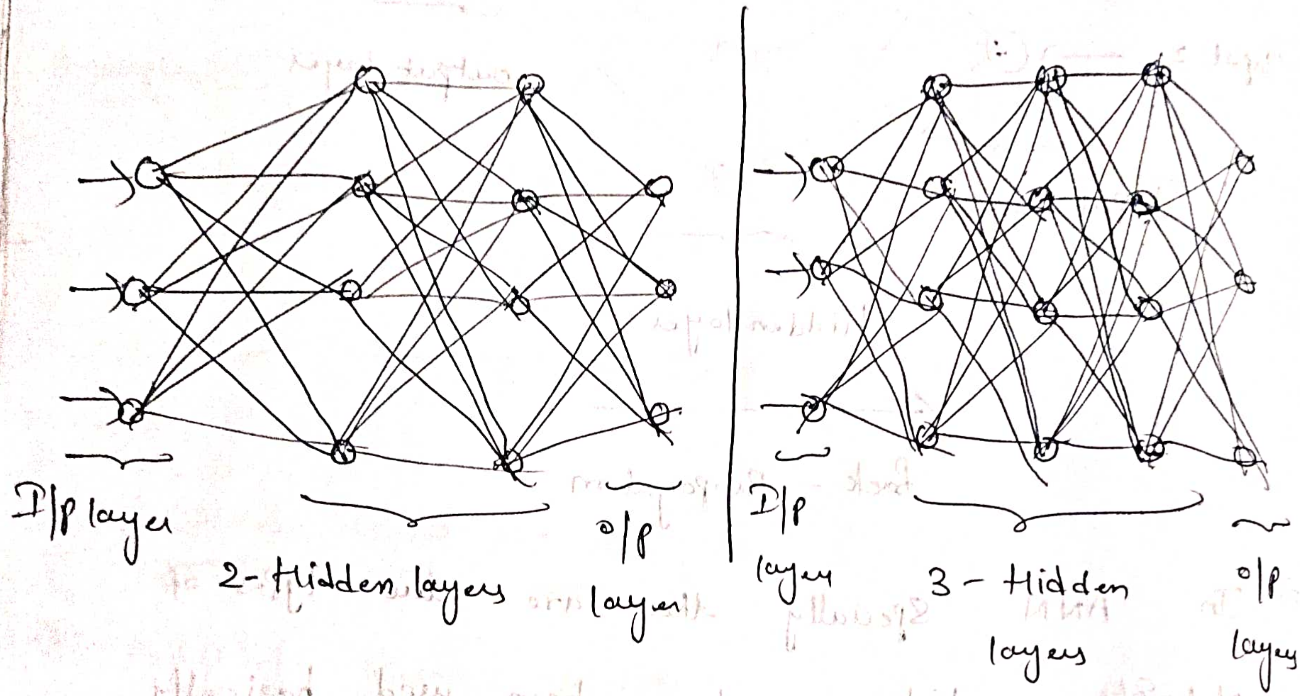
⇒ <u>SNN (Single Neural Networks)</u> :-

⇒ In this Neural networks we will be used only one ~~Neural~~ hidden layer to process the output



→ MNN / MLP (Multi Neural Networks / Multi layer Perceptron).

⇒ Multi Neural Networks are which process more than one hidden layers to produce an output.



⇒ Applications of ANN :-

1. Image Processing
2. character recognition
3. Forecasting
4. Facial recognition
5. Stock market prediction
6. Hand writing recognition

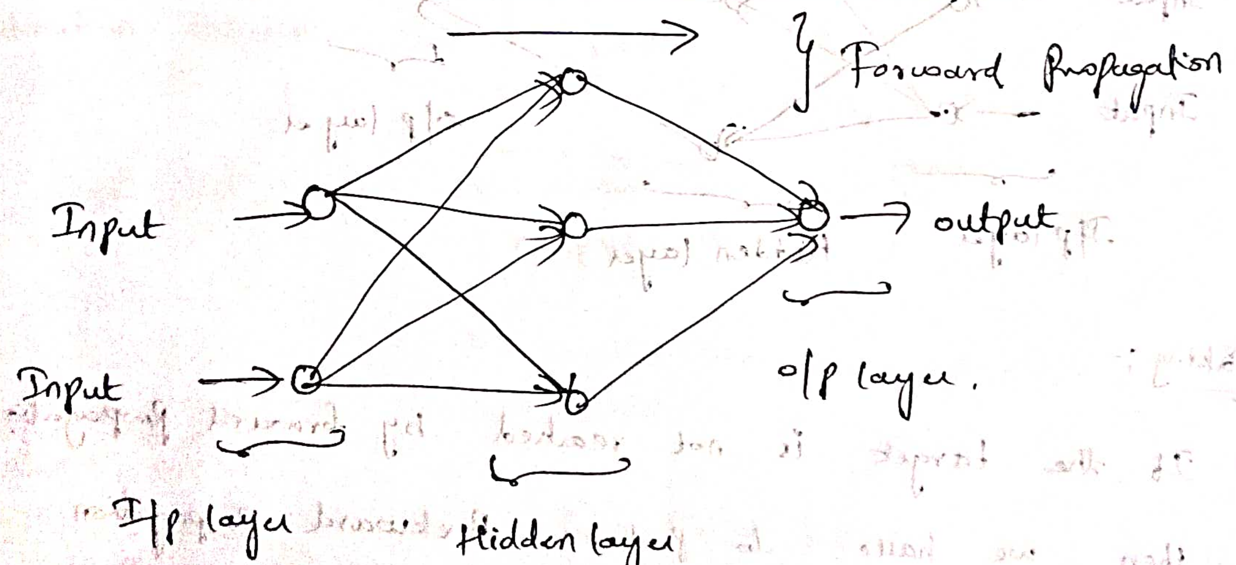
4) Training Deep Neural Networks :-

⇒ For Training a deep Neural Networks we often use two techniques those are.

1. Forward Propagation
2. Backward Propagation.

1) Forward Propagation :-

⇒ In the name it self forward Propagation is used to do the Process with in the flow of the layers like input layer, hidden layer and output layer according to the Target values.



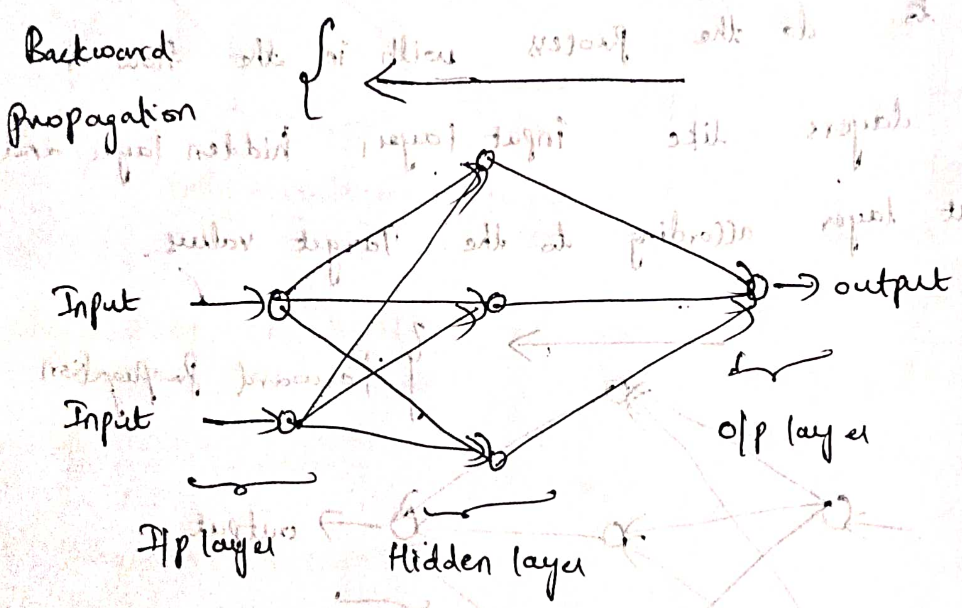
⇒ Working :-

⇒ First we have to take input values then we have to process that to hidden layer from that it will Process to output layer if the value reaches output of Target then our model will success otherwise we have to Perform Backward Propagation

2) <u>Backward Propagation</u> :-

=) The Core of Neural Network training is back Propagation its a technique for fine-tuning the weights of a neural networks based on the Previous Epoch's Error rate (i.e) iteration.

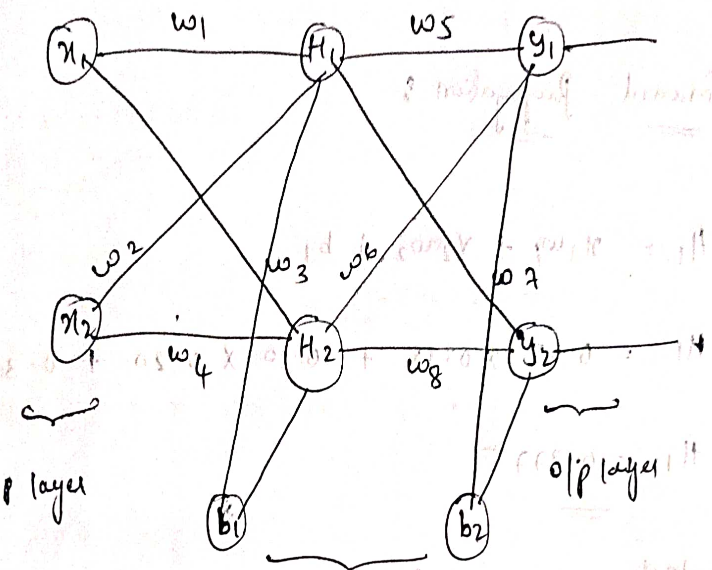
=) By fine-tuning the weights, you may low Error rates and improve the model's generalization, making it more dependable.



<u>Working</u> :-

=) If the target is not reached by forward Propagation then we have to perform Backward Propagation By updating Each weights we have to do from Backward i.e. from output layer to hidden layer and then to input layer and after reaching i/p layer again we have to perform Forward Propagation these iterations have to be done Until the target value reached.

Example :-



$$\boxed{H_1 = x_1 w_1 + x_2 w_2 + b_1}$$

Activation function is $\boxed{\text{Sigmoid} = \frac{1}{1+e^{-x}}}$

$$\boxed{\text{output } H_1 = \frac{1}{1+e^{-H_1}}}$$

Input values :-

$x_1 = 0.05$ $\quad b_1 = 0.35$

$x_2 = 0.10$ $\quad b_2 = 0.60$

Initial weights :-

$w_1 = 0.15$ $\quad w_4 = 0.30$ $\quad w_7 = 0.50$

$w_2 = 0.20$ $\quad w_5 = 0.40$ $\quad w_8 = 0.55$

$w_3 = 0.25$ $\quad w_6 = 0.45$

Target values :-

$T_1 = 0.01$ $T_2 = 0.99$.

1. Forward Propagation :-

$H_1 = x_1 w_1 + x_2 w_2 + b_1$

$H_1 = 0.05 \times 0.15 + 0.10 \times 0.20 + 0.35$

$H_1 = 0.3775$

output $H_1 = \dfrac{1}{1+e^{-H_1}}$ $\dfrac{1}{1+e^{-0.3775}}$

$= 0.593269992$

In the same way H_2 also

$H_2 = x_1 w_3 + x_2 w_4 + b_1$

$H_2 = 0.05 \times 0.25 + 0.10 \times 0.30 + 0.35$

$H_2 = 0.0125 + 0.4 + 0.35$

$H_2 = 0.7625$

output $H_2 = \dfrac{1}{1+e^{-H_2}}$

$= \dfrac{1}{1+e^{-0.7625}}$

$= 0.596884378$

Now, for calculating 'y_1'

$$y_1 = out\ H_1 \times w_5 + out\ H_2 \times w_6 + b_2$$

$$= 0.593269992 \times 0.4 + 0.596884378 \times 0.45 + 0.6$$

$$= 1.105905967$$

$$out\ y_1 = \frac{1}{1+e^{-y_1}} = \frac{1}{1+e^{-1.105905967}}$$

$$out\ y_1 = \underline{\underline{0.75136507}}$$

In the same way

$$y_2 = out\ H_2 \times w_8 + out\ H_1 \times w_7 + b_2$$

$$y_2 = 0.596884378 \times 0.55 + 0.593269992 \times 0.50 + 0.60$$

$$y_2 = 0.3282864079 + 0.296634996 + 0.60$$

$$y_2 = \underline{\underline{1.2249214039}}$$

$$output\ y_2 = \frac{1}{1+e^{-y_2}} = \frac{1}{1+e^{-1.2249214039}}$$

$$output\ y_2 = \underline{\underline{0.772928465}}$$

⇒ The outputs are not matched to target values

So we have to calculate Error for that

$$E_{Total} = \Sigma \frac{1}{2} (target - output)^2$$

1) Calculate Total Error :-

$$\varepsilon_{Total} = \sum \frac{1}{2} (\text{target} - \text{output})^2$$

$$= \underbrace{\frac{1}{2}(T_1 - \text{out } y_1)^2}_{O1} + \underbrace{\frac{1}{2}(T_2 - \text{out } y_2)^2}_{O2}$$

$$= \frac{1}{2}(0.01 - 0.7513)^2 + \frac{1}{2}(0.99 - 0.7729)^2$$

$$= 0.274811083 + 0.023560026$$

$$= 0.298371109$$

$$\varepsilon_{Total} = 0.298371109 \quad (\text{Error rate})$$

2) Back Propagation :-

To update new weights from old weights

Consider w_5 :-

$$\text{Error at } w_5 = \frac{\partial E_{total}}{\partial w_5}$$

Partial differention

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial \text{out } y_1} \times \frac{\partial \text{out } y_1}{\partial y_1} \times \frac{\partial y_1}{\partial w_5}$$

$$E_{total} = \frac{1}{2}\left(T_1 - out\ y_1\right)^2 + \frac{1}{2}\left(T_2 - out\ y_2\right)^2$$

$$\frac{dE_{total}}{d\,out\ y_1} = \cancel{2} \times \frac{1}{\cancel{2}}\left(T_1 - out\ y_1\right)^{2-1} * -1 + 0$$

$$= -\left(T_1 - out\ y_1\right)$$

$$= -\left(0.01 - 0.75136507\right)$$

$$= 0.74136507$$

$$out\ y_1 = \frac{1}{1+e^{-y_1}}$$

$$\frac{d\,out\ y_1}{d y_1} = out\ y_1\left(1 - out\ y_1\right)$$

$$= 0.75136507\left(1 - 0.75136507\right)$$

$$= 0.186815602$$

$$\frac{dy_1}{dw_5} = 1 \times out\ H_1 \times w_5^{(1-1)} + 0 + 0$$

$$= out\ H_1$$

$$= 0.593269992$$

$$\frac{dE_{total}}{dw_5} = \frac{dE_{total}}{d\,out\ y_1} \times \frac{d\,out\ y_1}{dy_1} \times \frac{dy_1}{dw_5}$$

$$= 0.74136507 \times 0.186815602 \times 0.593269992$$

$$= 0.082167041 \text{ (new } w_5 \text{ value)}$$

⇒ Similarly we have to update all weights i.e, $w_6, w_7, w_8, w_1, w_2, w_3, w_4$.

(5) <u>Improving deep Neural Networks</u> :-

⇒ To improve the Performance of the deep Neural networks we have to gone through three approaches they are.

1. Hyper parameter tuning.
2. optimization
3. Regularization

⇒ 1) <u>Hyper parameter tuning</u> :-

⇒ Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set.

⇒ The Combination of hyperparameters maximizes the models Performance, minimizing a predefined loss function to produce better results with fewer Errors

Parameters

- Model Parameters (Internal)
  1. weights
  2. Bias
- Hyper Parameters (External)
  1. Learning rate
  2. no. of Epochs.

1. Weights - which is used to pass the data between neural network to produce Accurate output

2. Bias - which is a constant that supports the weights to produce output

3. Learning rate - which will be taken Externally to produce optimal value for accurate o/p

Ex :- $\boxed{\eta = 0.8}$

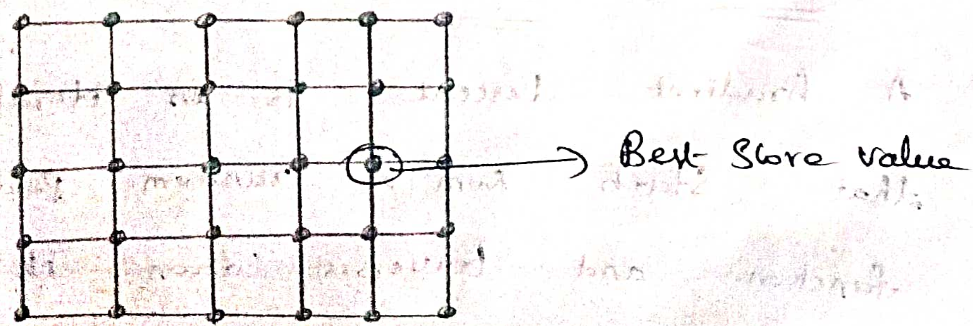
4. no. of Epochs - It is nothing but iterations how deeply the neural networks are trained to produce accurate o/p.

⇒ Methods of Hyper Parameter tuning :-

1. Grid Search CV (CV = Cross Validation.)
2. Randomized CV.

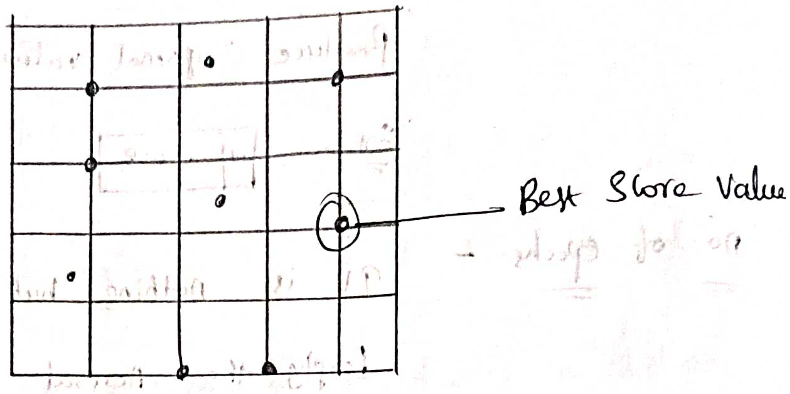
1. Grid Search CV :-

⇒ This is the method which we have cross validate all combinations of the sets in the grid values and then pick the value that gives the best score.



2. Randomized CV :-

=) Randomized Search CV is a method which is used to only cross validate random combination of hyper-parameters i.e random samples.



2) optimization technique :-

=) optimization algorithms are responsible for reducing losses and provide most accurate results possible.

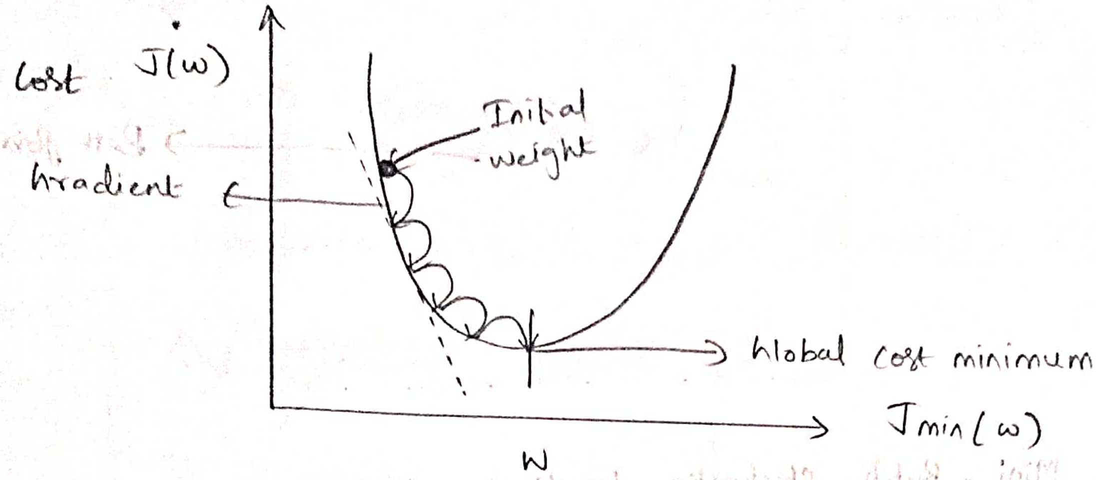
=) Techniques used in this optimization are :-

1. Gradient Descent
2. Stochastic Gradient Descent (SGD)
3. Mini-Batch Stochastic Gradient Descent (MB-SGD) :

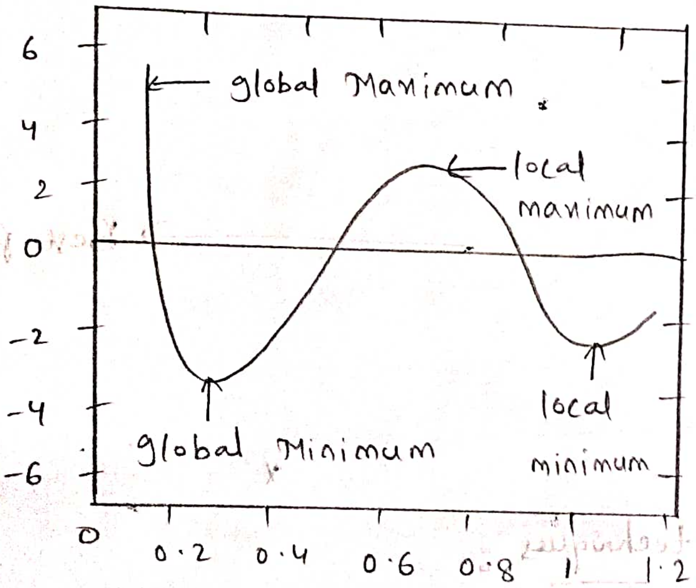
=) 1. Gradient Descent :-

=) A Gradient descent is an iterative algorithm, that starts from a random point on the function and traverses down its slope in steps

Until it reaches lowest point of that function



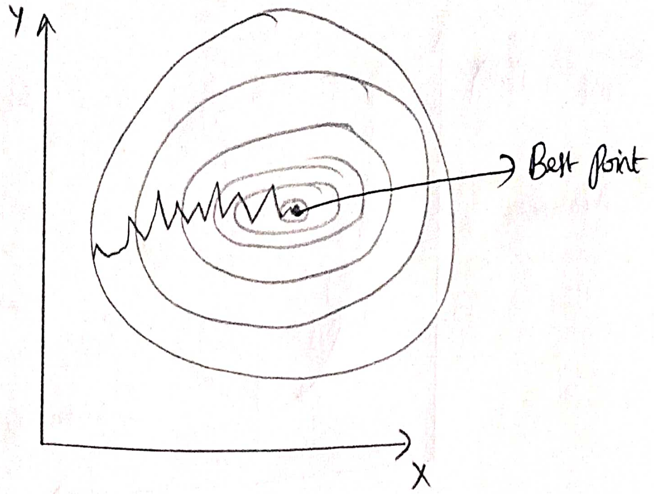
$$\boxed{\text{Cost Function :- } \quad \theta_j = \theta_j - \alpha \cdot \nabla J(\theta)}$$



⇒ To obtain the best solution we must reach global minimum point.

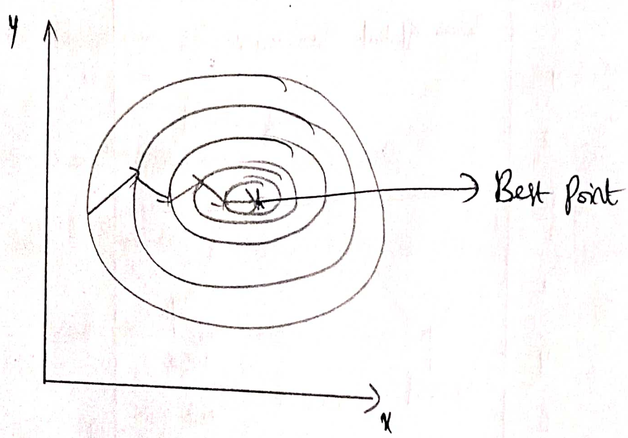
2. <u>Stochastic Gradient Descent (SGD)</u> :-

⇒ SGD is generally noisier than typical Gradient Descent, it usually took a higher number of iterations to reach the minima, because of its randomness in its descent.



3. Mini - Batch Stochastic Gradient descent :-

⇒ It is considered to be the cross - over between GD and SGD. it reaches the point much faster than the SGD



3) Regularization techniques :-

⇒ Regularization is used to minimize the adjusted Loss function and Prevent Overfitting (or) Underfitting.

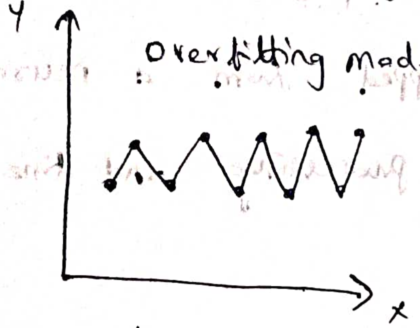
⇒ Techniques used in Regularization are

1. L1 regularization.
2. L2 regularization
3. Dropout & Drop Connect regularization

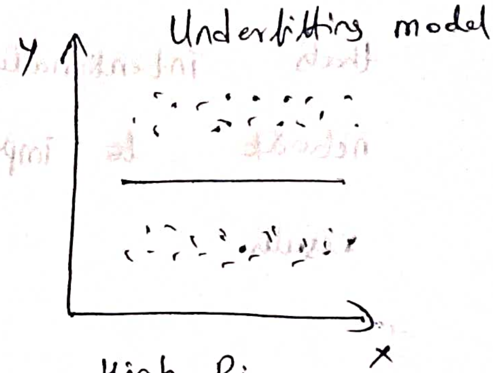
1. L_1 Regularization :- (or) Lasso Regression :-

=> It is used to find a Good model without overfitting (or) Underfitting

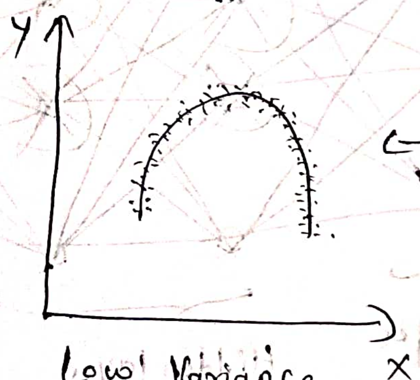
Ex :-



High Variance



High Bias



← Good Balance model

Low Variance

Low Bias

2. L_2 Regularization (or) Ridge Regression :-

=> It is used to find the loss of the data which we have taken in a Squared form we have to take it as a cost function. (Mean Squared Error)

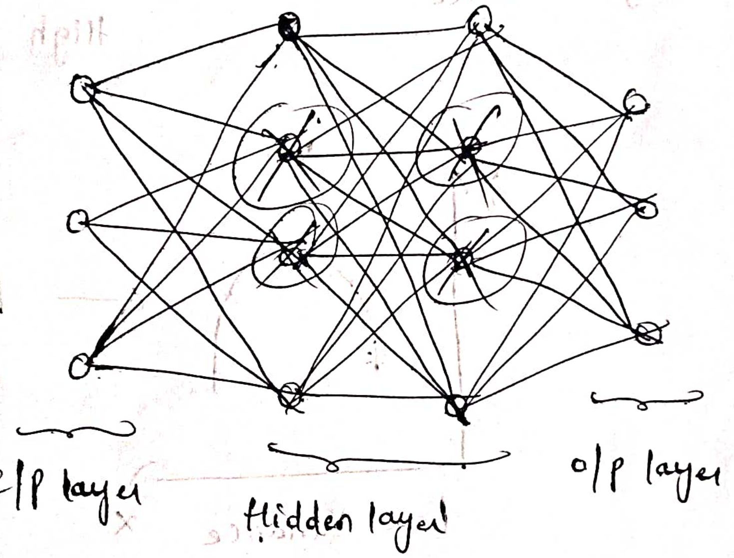
$$MSE = \frac{1}{N}\sum_{x \in d}\left(\text{Original value} - \text{Prediction value}(x)\right)^2$$

$$MSE = \sum_{x \in d} \frac{(y - y')^2}{N}$$

3. Drop out and Drop Connect Regularization :-

=) Dropout regularization :-

=) Dropout refers to data, (or) noise Error data that's intentionally dropped from a neural network to improve processing and time to results.



=) Drop Connect regularization :-

=) After dropout unneccessary data nodes we have to remove all connections of that nodes and reconnect network for remaining nodes