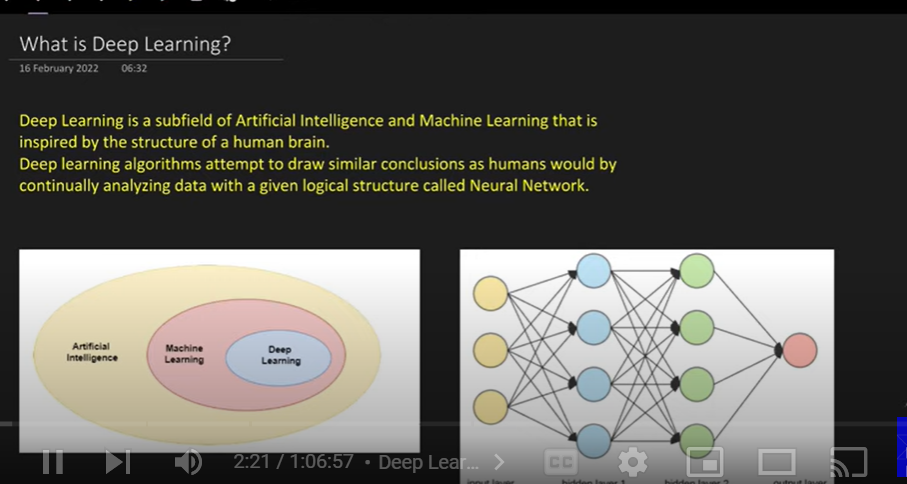
Deep learning:

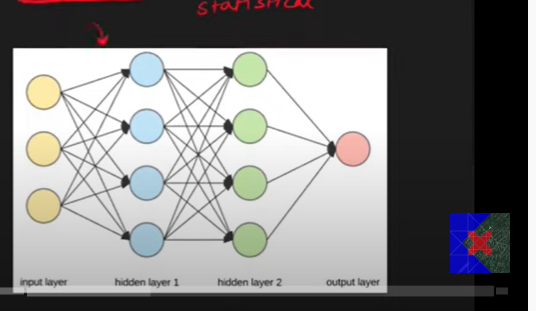


Sub field of ai and machine learning

We learn from data

Machine learning: depends on statistical techniques

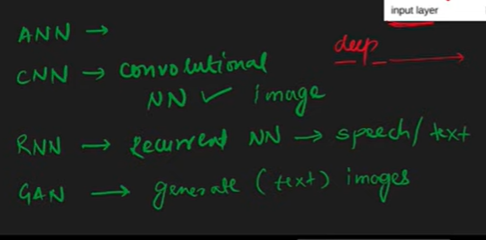
Deep learning: depends on logical structure , neural networks( human brain)

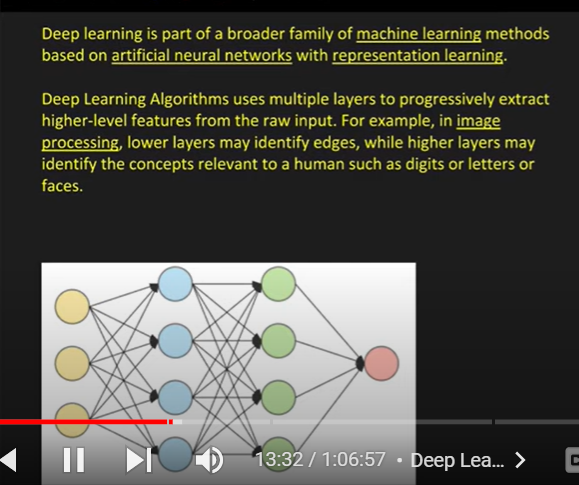


It’s a logical structure

Fundamental unit is perceptron

Deep – as number of hidden layer increases





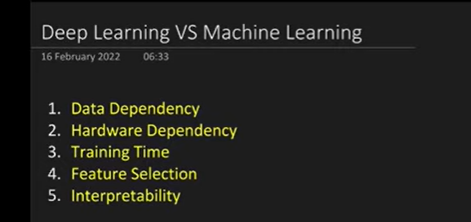
Part of machine learning

Deep learning feature engineering not require, it automatically does that

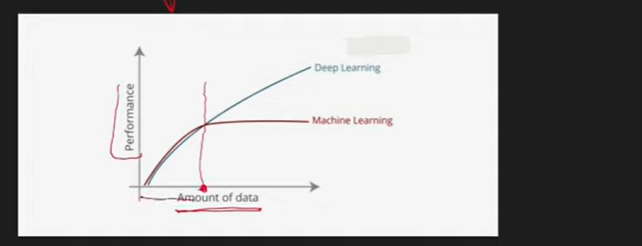
As the deepness increases complexity increases

Which uses: representation learning(it does feature extraction by itself)

Difference between deep learning and machine learning:



Data learning require large quantity of data



For less amount of data deep learning

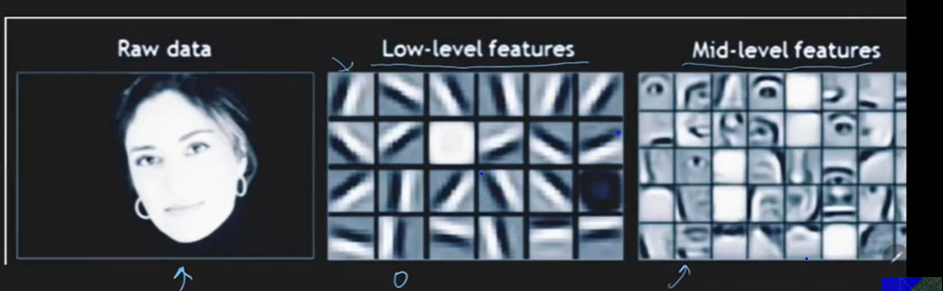
Data increases – performance increases

Deep learning : complex multiplication – to handle that we need powerful gpu with more memory

Training time is high as deep learning calculation are high

During prediction time: high speed

Feature selection: representation learning, it extract relevant features

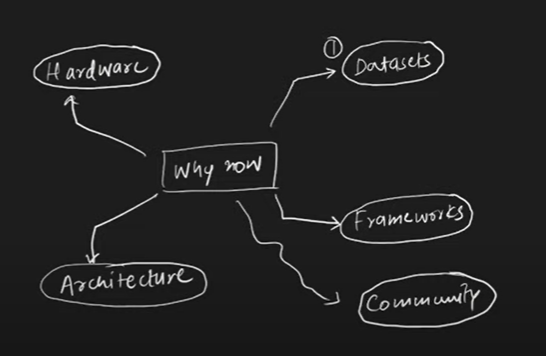


For first hidden layer extracts low level features

2nd – mid level features

Problem with interpretability :

We don’t know what’s running inside mode to decide cat or dog—so called black box



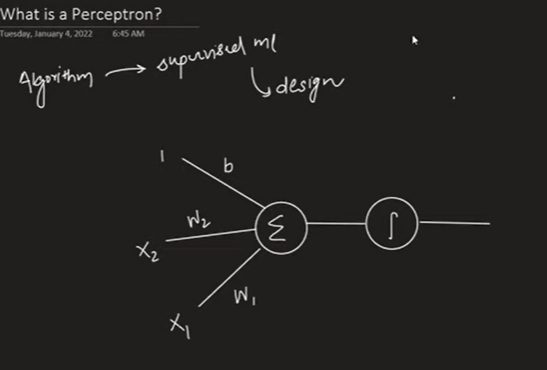
Require huge volume of data

**What is a Perceptron? Perceptron Vs Neuron | Perceptron Geometric Intuition**

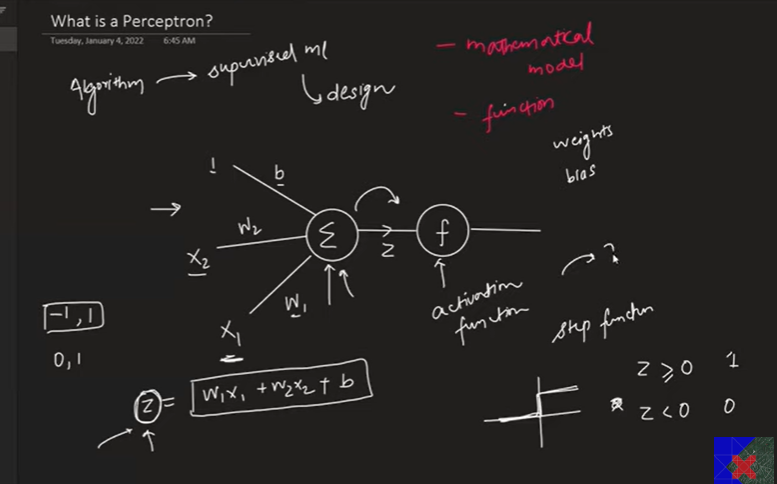
Perceptron is building block of neural network

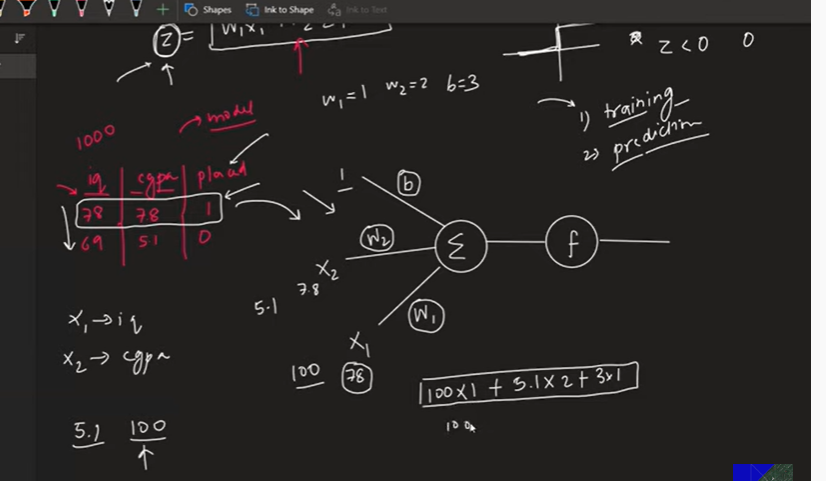
Perceptron is algorithm used for supervised machine learning

Building block

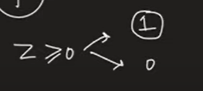


Mathematical model or function

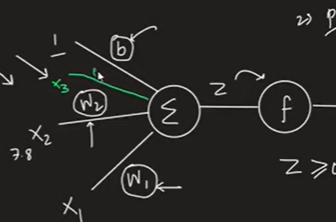


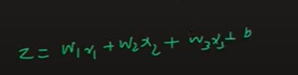


Send to activation function- find z value

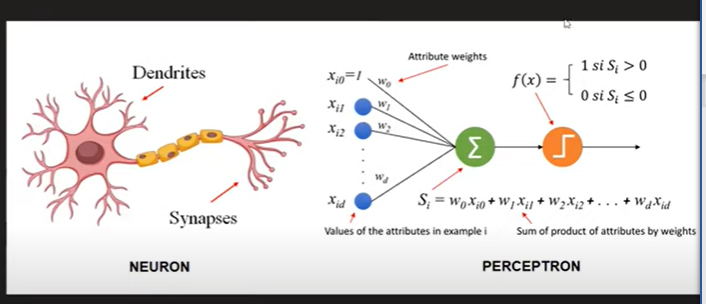


If I have more than 2 columns





Deep learning heavily inspired by our own nervous system



Dendroid – input

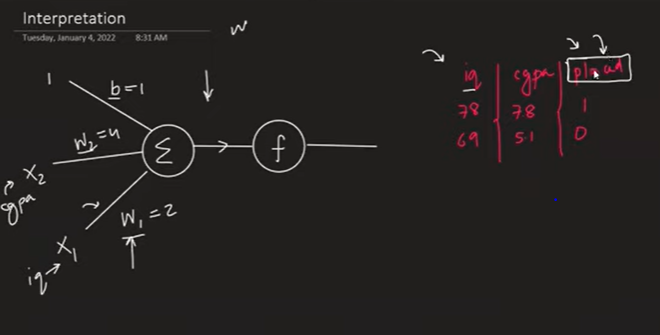
Center- activation

Axon- output

Diff between neuron and perceptron:

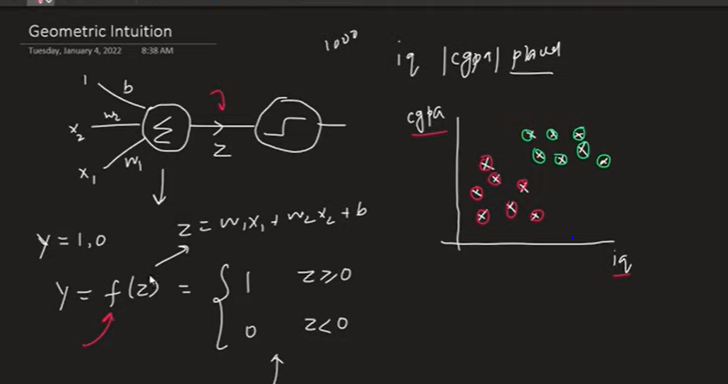
Neuron is very complex where as perceptron is simple

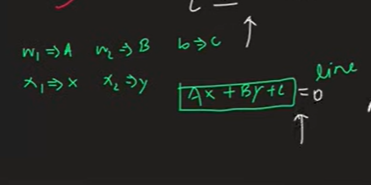
Summation function and step (activation), where as neuron complex chemical function



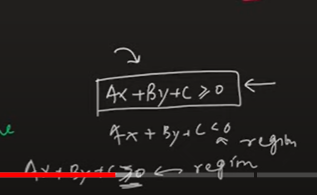
Since cgpa weight is 4 compare to iq weight 2: so cgpa is much important

Weights tells us about feature importance





Below shows region



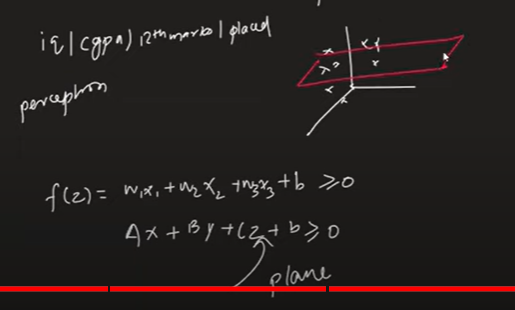
Perceptron is basically line,

By using that we find regions

With this region we divide 2 class , perceptron is binary classifier

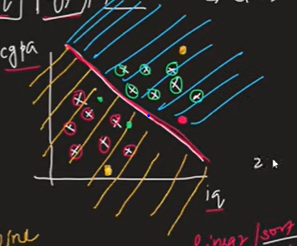
More than 2 variables- 3d – perceptron acts line as plan

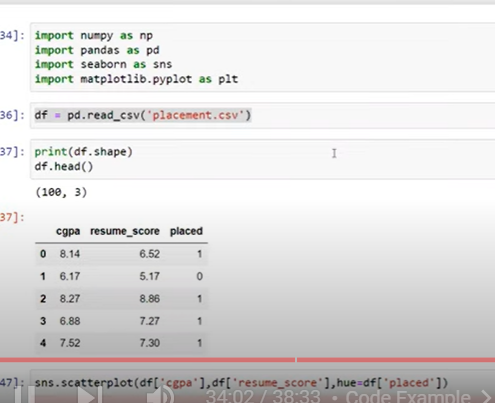
4D – onwards perceptron acts like hyperplane



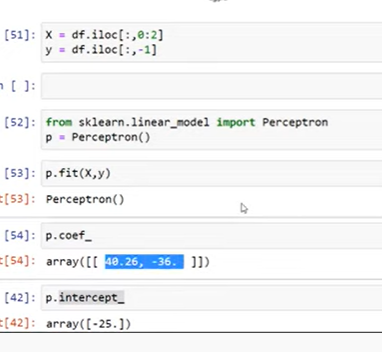
Perceptron classify line or sort of line- data should be almost linear

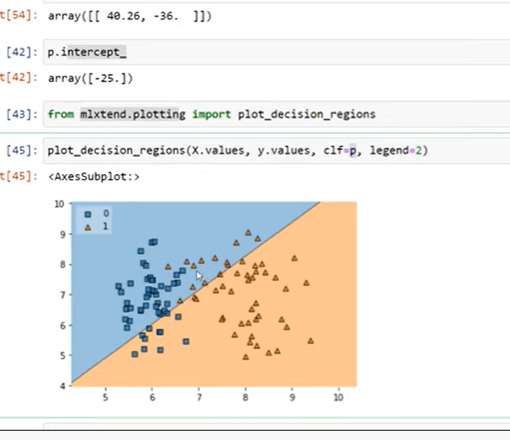
Perceptron fails if data is not linear



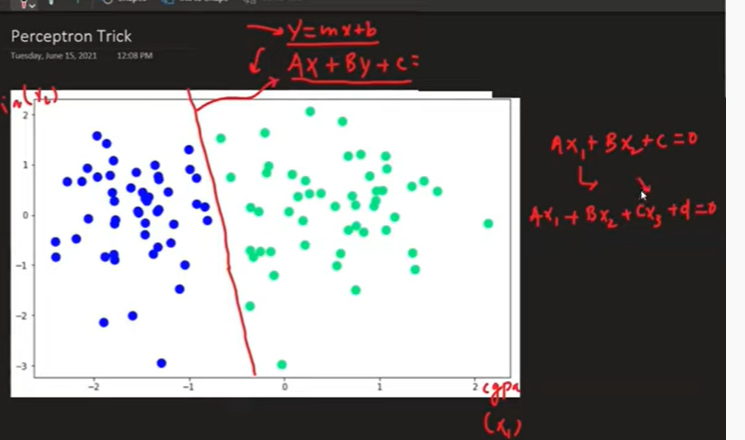


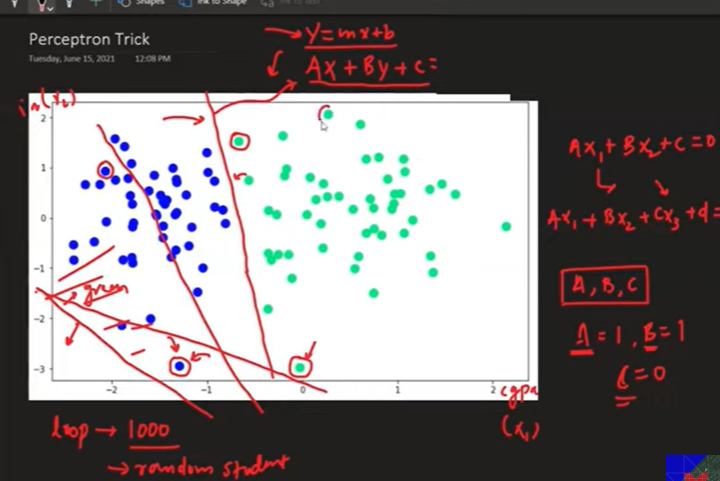






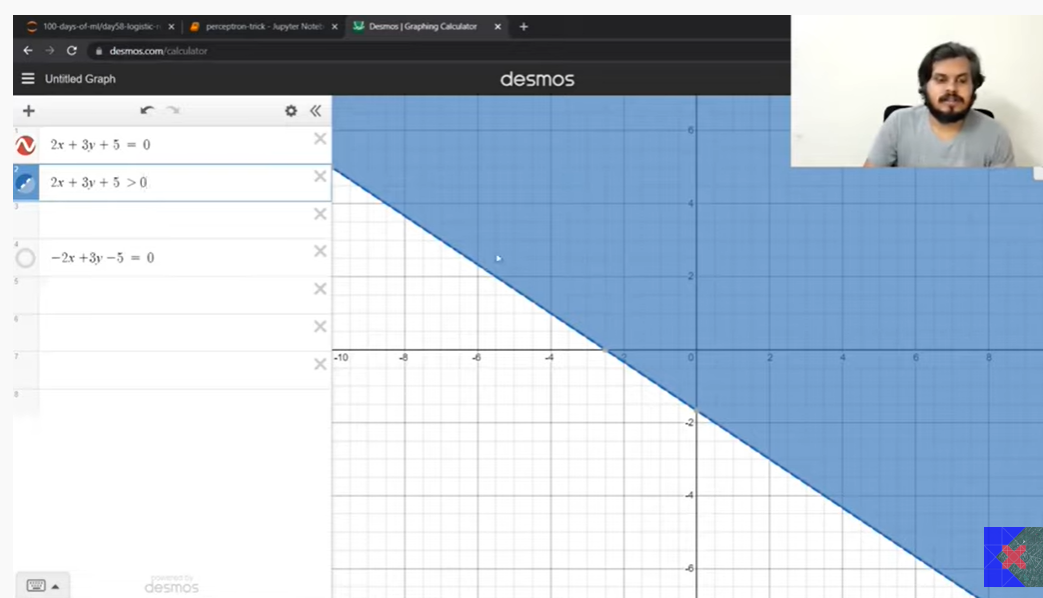
**Perceptron Trick | How to train a Perceptron | Perceptron Part 2 | Deep Learning Full Course**

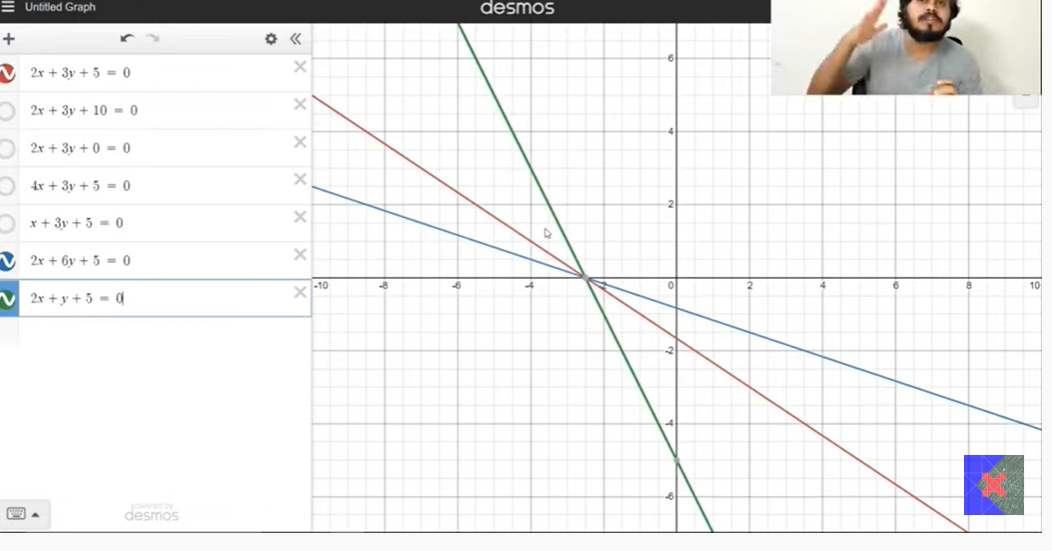




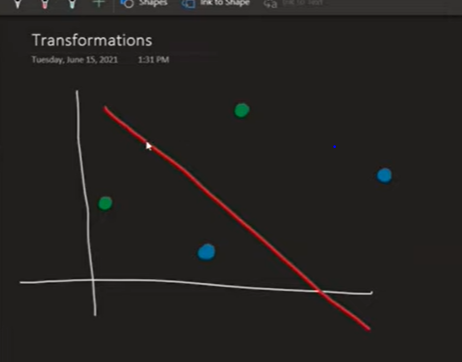
A, b, c value changes based on points position if they are correctly classified

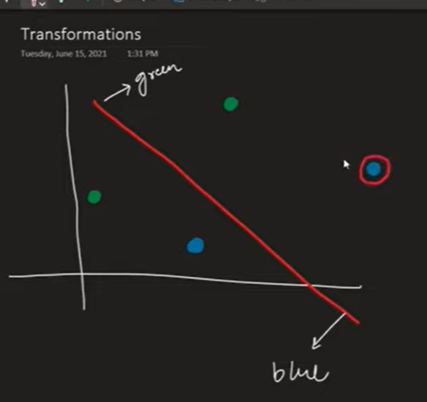
This convergence will be continued till nothing is miss classified





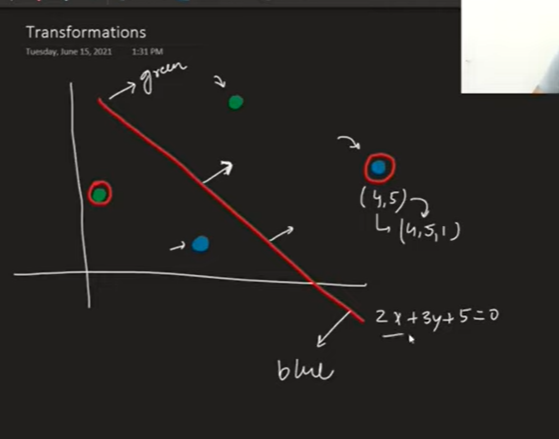
Transformation

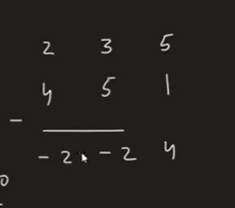




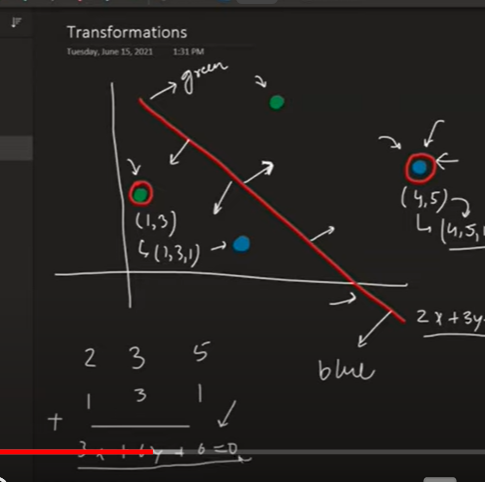
2 points miss classified

Transformation should be based on miss classified



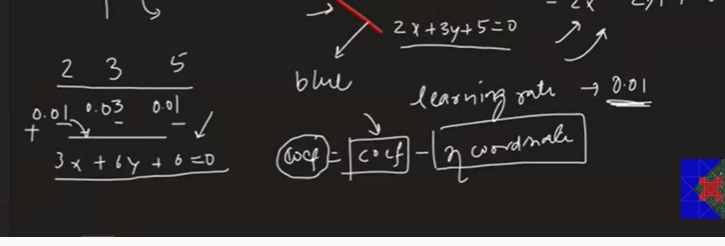


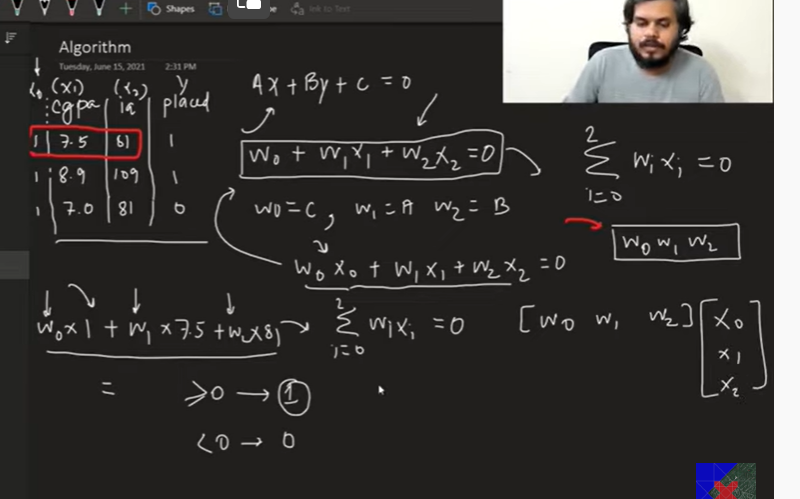
New line =-2x-2y+4

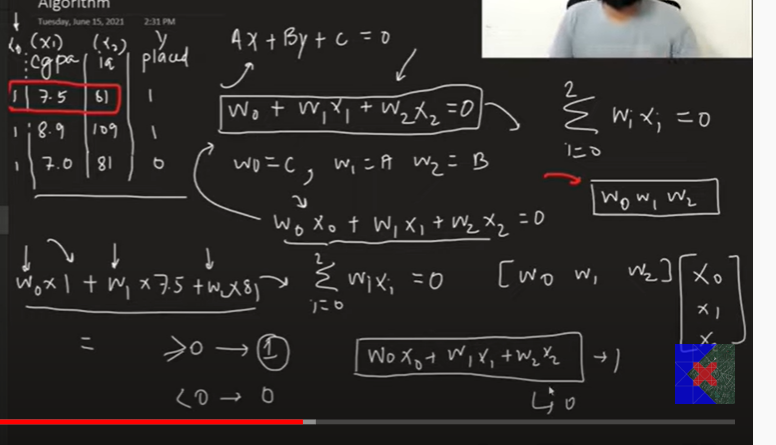


We don’t directly deduct

But it will be based on learning rate

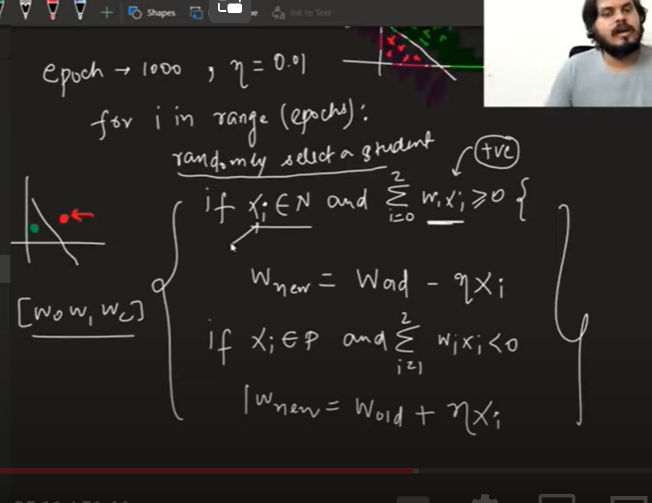






We decide epoch value =1000

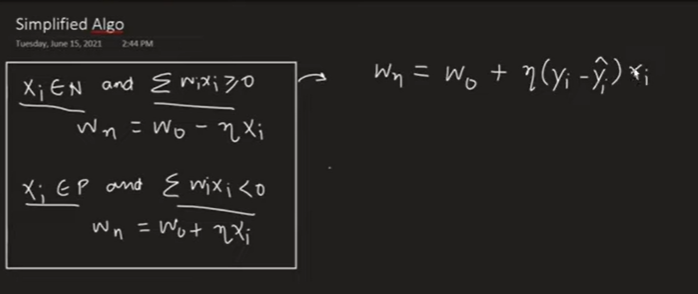
Learning rate eata: 0.01

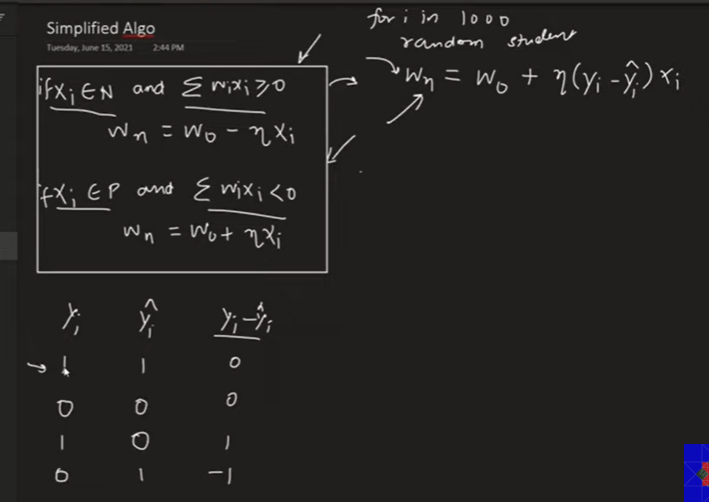


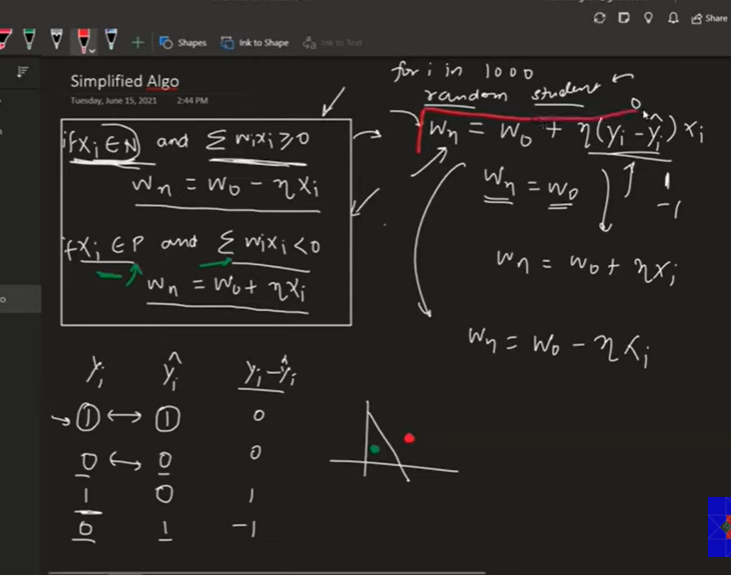
Wnew will be based on if ur wrongly classifies data in positive or negative region

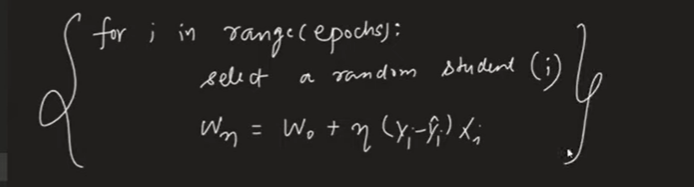
Negative region +add

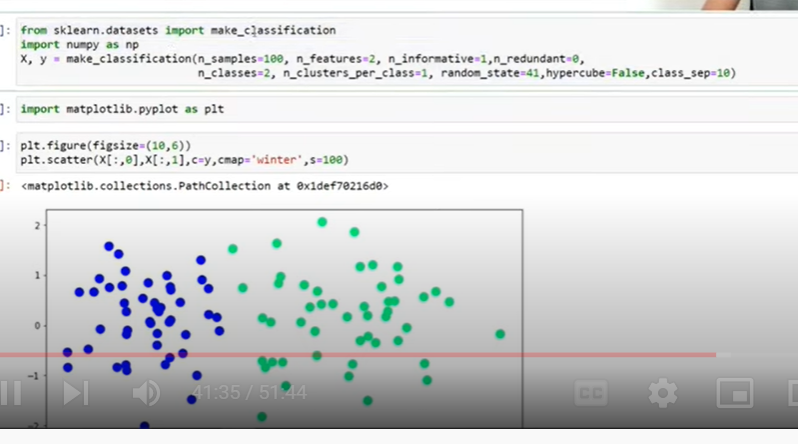
Positive region –

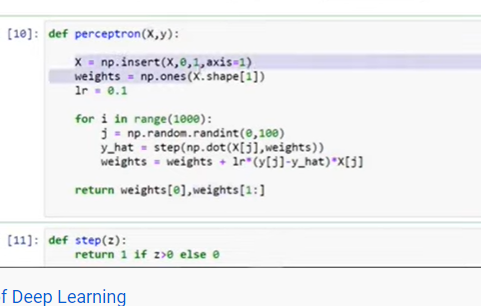


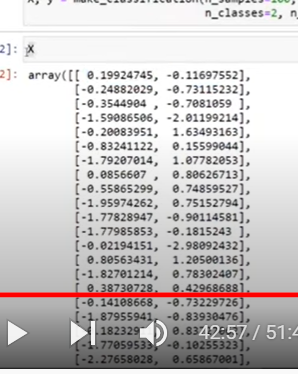




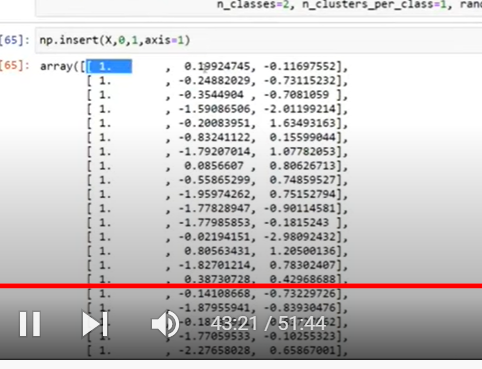


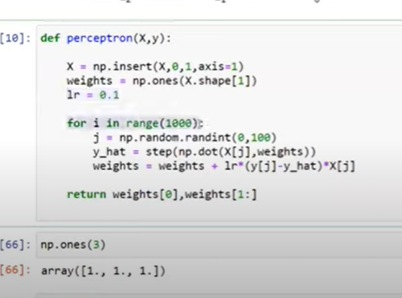




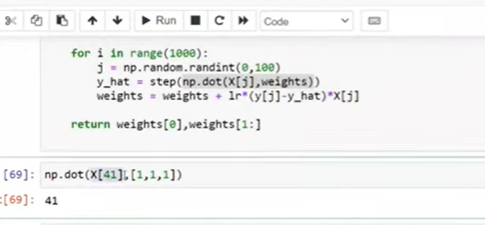


X lo add one more column as 1





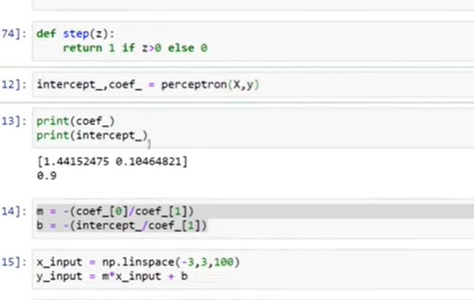
Np.once creates coefficients 1 for bias and variables

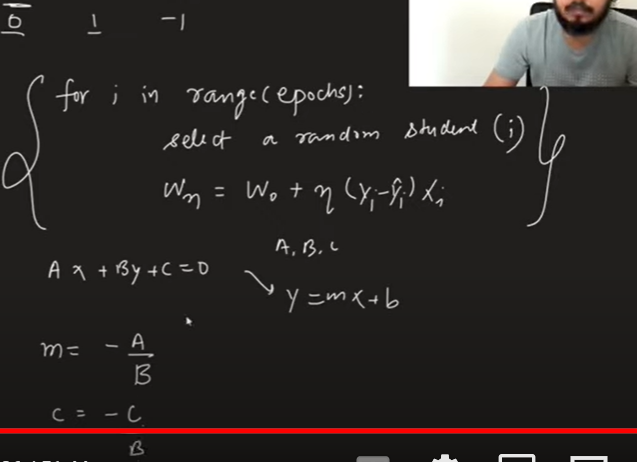


If Y-hat is greater than 1 will be placed

Weights update

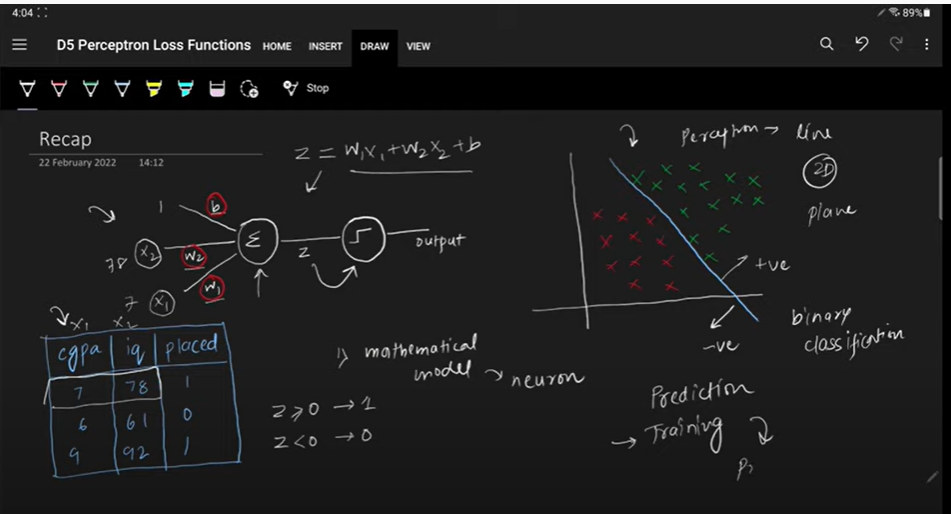
Weight[0]: intercept





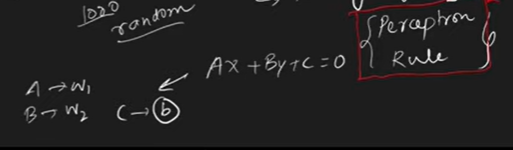
**Perceptron Loss Function | Hinge Loss | Binary Cross Entropy | Sigmoid Function**

Perceptron is mathematical function based on neurons



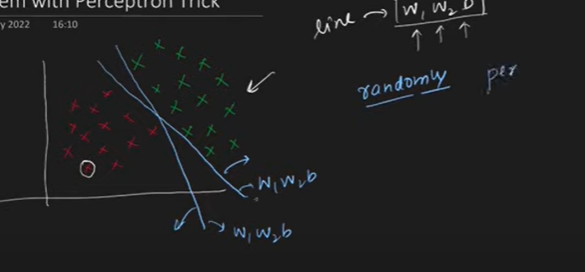
In training we will find weights bias and coefficients

Perceptron trick



Perceptron trick ; we cant say 100 percent that the line is correct and weights are correct

If point is not miss classified, line will not be moved

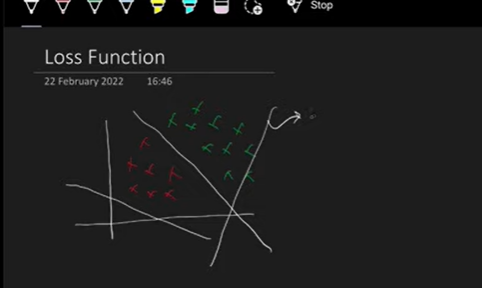


Perceptron trick will not say how good the model

Perceptron model will not converge as it takes random points

If same point selected every time : line will not be moved

LOSS FUNCTION: quantify your model

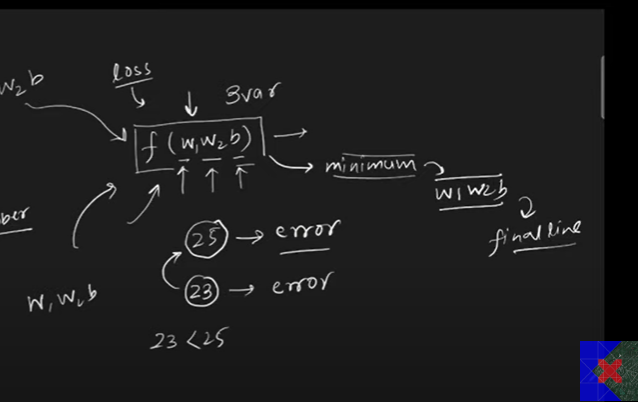


We need such a line which classify both perfectly

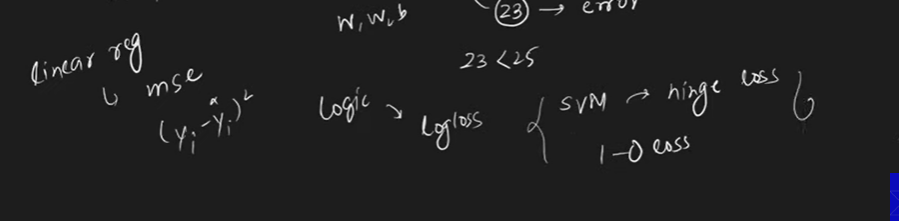
Line is a equation

Loss function is a finction of W1,w2,b

Any changes in w1,w2.b will change equation



We need to find such w, b where the error will be less



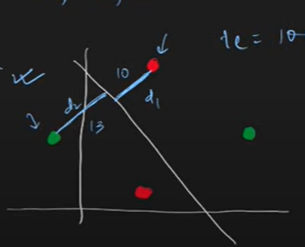
Loss function for perceptron:



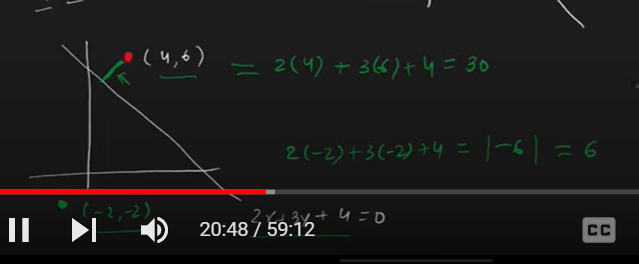
Number of mis classified points also can be error

If mis classified points near to line is high error

If miss classified point near to line is less error

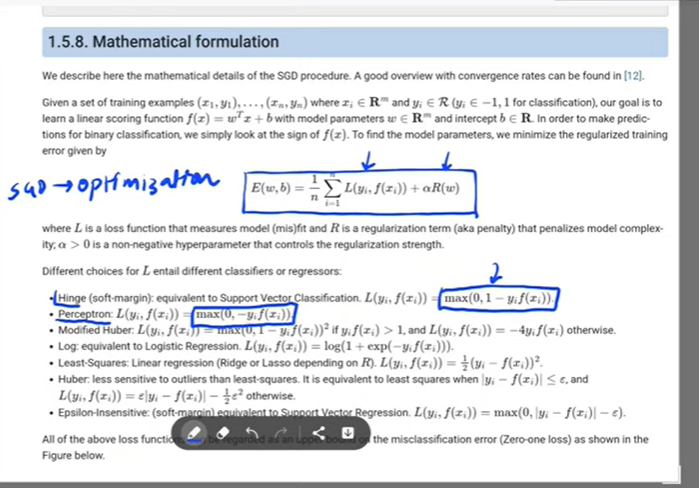


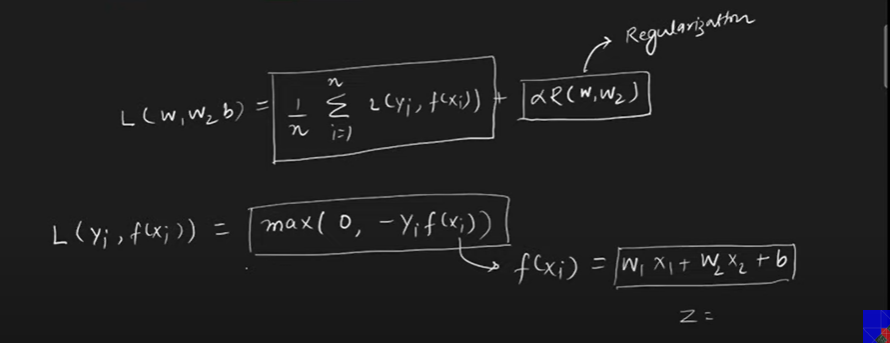
But perceptron function error don’t depend on error



Total loss=36

This above function is called dot product

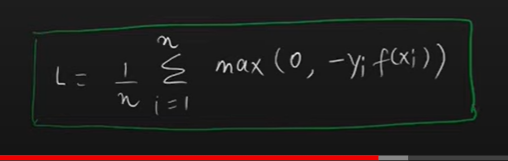




It has both loss and regularization

First we need to calculate loss

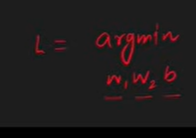
Perceptron loss function



Loss function depends on w1,w2,b

Because remaining things constant

We need to find such w,b where



Above equation is min

