C

2-10-2021 Tuesday

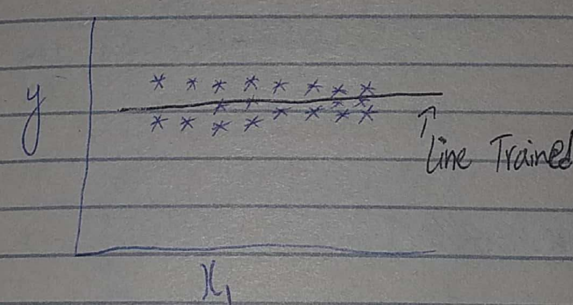
Week #7 Lecture #10

inear Regression:

Continous value as an output

g House Price Prediction

X_1, X_2 → no. of bedrooms , y
size Price
of House



$y = mx + c$ Learn values of ~~of~~ m & c

Hypothesis:

$$H_\theta(X_1, X_2) = W_2 X_2 + W_1 X_1 + \underbrace{W_0}_{c}$$

n no. of features = n no. of weights

W_3	W_2	W_1	W_0

1x4

$$\begin{bmatrix} X_3 \\ X_2 \\ X_1 \\ X_0 \end{bmatrix} = y \qquad y = WX$$

= 4x1

Loss function:

$$L = \frac{1}{2m} \sum_{i=1}^{m} \left(h(x_i) - \underbrace{y_i}_{\text{original label}} \right)^2$$

MS~~B~~E

Mean Squared Error

$$dw_2 = \frac{2}{2m} \sum^{m} \left(w_2 x_2 + w_1 x_1 + w_0 - y \right) \left(x_2^{m} \right)$$

$$dw_2 = \frac{1}{2m} \left[\frac{d}{dw_2} \left(w_2 x_2^{(0)} + w_1 x_1^{(0)} + w_0 - y^{(0)} \right)^2 \right] + \left(w_2 x_2^{(1)} + w_1 x_1^{(1)} + w_0 - y^{(1)} \right)^2 + \cdots + \left(w_2 x_2^{(m)} + w_1 x_1^{(m)} + w_0 - y^{(m)} \right)^2 \Big]$$

$$\cancel{dw_2 = \frac{1}{2m}[}$$

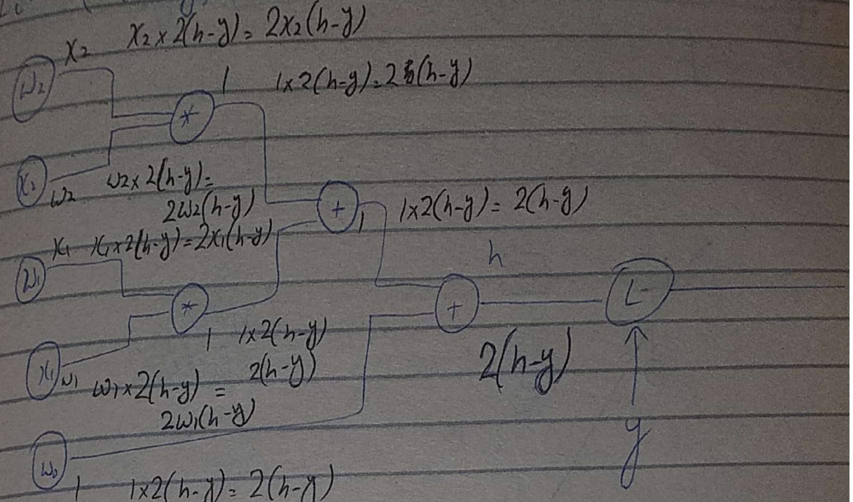
$$\cancel{dw_2} = \frac{1}{2m} \left[\cancel{2(w_2) + 2(w_2) + \cdots 2 + (w_2)} \right]$$

$$dw_2 = \frac{1}{2m} \Big[2 \left(w_2 x_2^{(0)} + w_1 x_1^{(0)} + w_0 - y \right) \left(x_2^{(0)} \right) + 2 \left(w_2 x_2^{(1)} + w_1 x_1^{(1)} + w_0 - y \right) \left(x_2^{(1)} \right) + \cdots + 2 \left(w_2 x_2^{(m)} + w_1 x_1^{(m)} + w_0 - y \right) \left(x_2^{(m)} \right) \Big]$$

$$dw_2 = \frac{1}{\cancel{2}m} \left[\cancel{2} \sum_{i=1}^{m} \left(w_2 x_2^{(i)} + w_1 x_1^{(i)} + w_0 - y^{(i)} \right) \left(x_2^{(i)} \right) \right]$$

$$dw_2 = \frac{1}{m} \sum_{i=1}^{m} \left(\left(w_2 x_2^{(i)} + w_1 x_1^{(i)} + w_0 - y^{(i)} \right) \left(x_2^{(i)} \right) \right)$$

$L_i = (h(x) - y)^2$

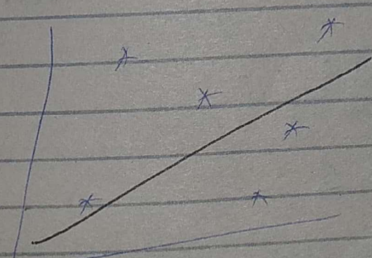


$$dw_2 = \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i)(x_2)$$

$$dw_1 = \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i)(x_1)$$

$$dw_0 = \frac{1}{m} \sum_{i=1}^{m} (h(x_i) - y_i) * 1$$

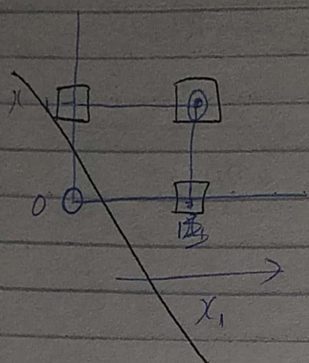
Linear regression is not good for complex data sets



Linear line can't make good Predictions in this.

Algorithm for Binary classification. / Logistic Regression

OR Gate



Linearly Separable

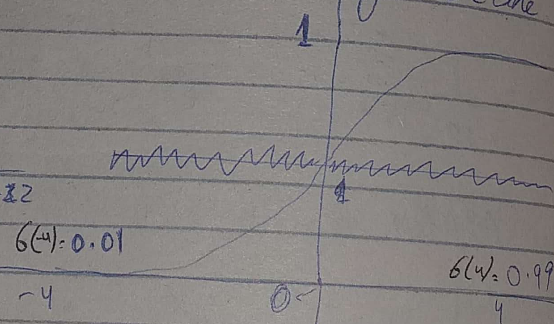
classes can be easily Separated in these classification.

only 2 classes.

$h(x) = \omega x$ //basically mx+c line

$Z = \omega x$

$h(z) = \dfrac{1}{1+e^{-z}}$



Another filter

After h(z)
if Probability greater than 0.5 or something, label predicted is 1, else label is 0.

answer between 0 & 1

It can be taken as Probability.

E

In Neural Network, 1 Neuron is a logistic unit.

Its a Loss function is

$$L = y\log(h(z)) - (1-y)(\log(1-h(z)))$$

in Positive example

$(1-y)$ term becomes Zero

For Positive example, label was 1

$$L = 1*\log(h(z))$$

$$L = \log(h(z))$$

True label was 0, ∞

True label was 1, 0

For Negative example, label was 0

$$L = 1*\log(1-h(z))$$

if $h(z)$ is 1. , true label was 0

$$L = 1*\log(1-1)$$

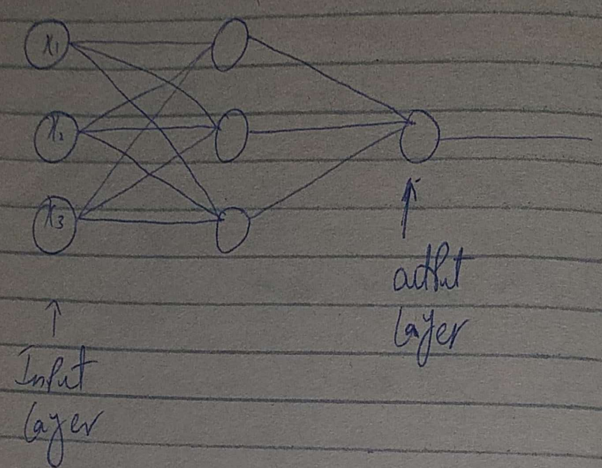
$$\log(0) = \infty$$

if $h(z)$ is 0 , true label was 0

$$L = 1*\log(1-0)$$

$$L = 0$$

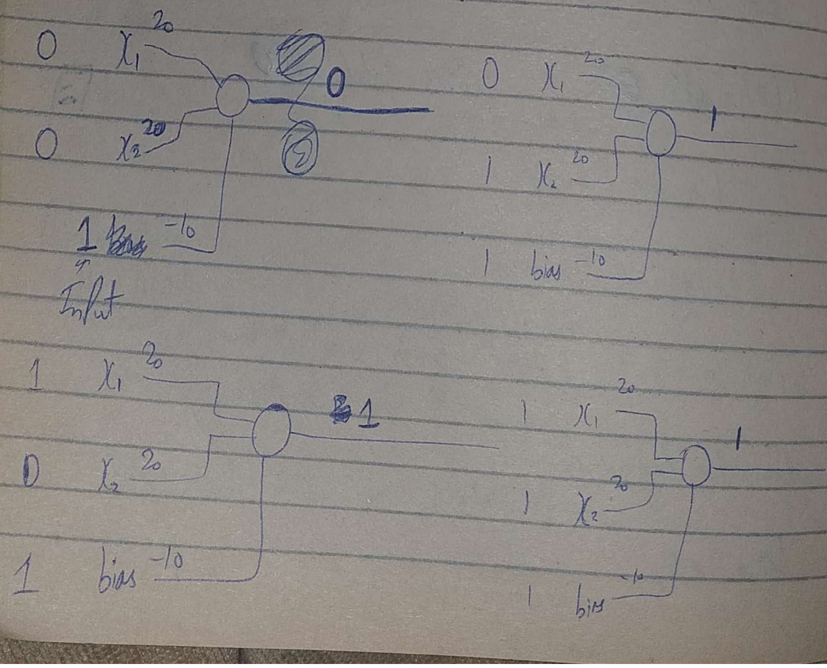
Neural Network:-



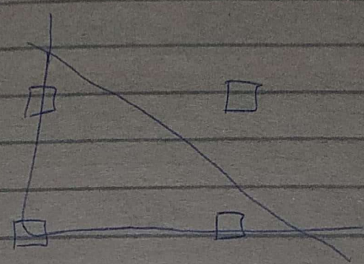
more Neurons means we can learn more complex functions. Polynomial function.

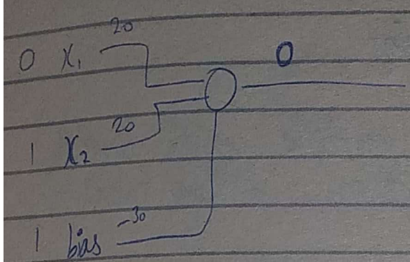
e.g ~~Linear~~ Logistic Regression.

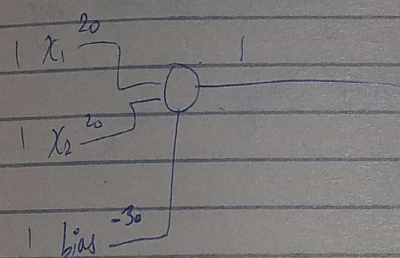
OR Gate



AND Gate:







XOR Gate:-

1 output when both input is same.

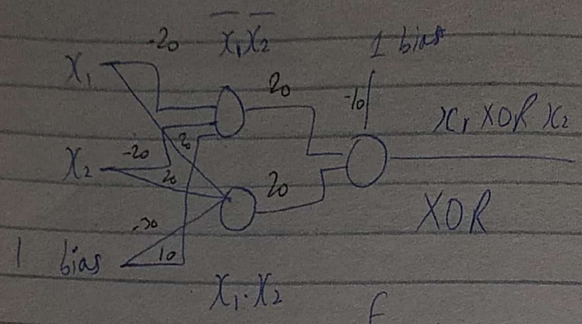
(not A . Not B)

+

(A . B)

3 logistic units.

XOR = $\bar{A}\bar{B} + AB$



a plane is learned After training.

$\boxed{\text{Polynomial Regression}}$ we use multiple logistic units to learn XNOR.

Problem:

As features increases, Polynomial becomes long.

NN = universal function Interpreter/Learning.

→ multiple logistics units