* Machine learning :-

Learning means = improving with experience at some task.

- Improve over task T,
- With respect to performance measure, P
- Based on experience, E. (data)

⇒ Traditional Programming :-

Normal Computer v/s ML.

Data → [computer] → output.
Program →

Machine learning :-

Data → [computer] → Program.
output →

⇒ Types of Machine learning :-

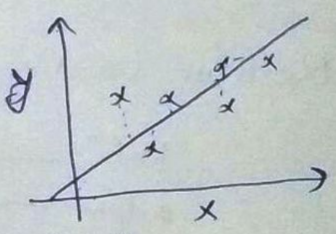
- Supervised learning.
  - providing right answers for every data.
  - classification → Discrete values
  - Regression = continuous values. (infinite features.
- Unsupervised learning - clustering the data with different arrangements.
- Reinforcement learning :- trail and error method in finding the best outcome based on experience.

⇒ Regression :- It can be used in cases to determine casual relations between the independent and dependent variables.

Training set
↓
learning Algorithm.
↓
~~Size of~~ i/p (x) ⟶ $\boxed{h}$ ⟶ y O/p.
hypothesis.

So, how do we represent this hypothesis 'h', which will maps the input to the output.

① Linear Regression :-



$$h_\theta(x) = \theta_0 + \theta_1 x.$$

So to minimize $h_\theta(x)$, we have to minimize θ_0, θ_1 & for that we can define a function such as cost function. as :-

$$\min_{\theta_0,\theta_1} J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$h_\theta(x^{(i)})$ = predicted value

$y^{(i)}$ = actual value

$h_\theta x^{(i)} - y^{(i)}$ = error of the given data.

This cost function with two parameters can be very complex while computation, so we need some software to minimize the cost function -

Gradient Descent.

③

Gradient Descent

for minimizing the cost function $J(\theta_0, \theta_1)$

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$\alpha \rightarrow$ learning parameter. (choosen smartly)

Gradient means $\rightarrow$ slope of the line we draw on the graph.
$\llcorner$ automatically take smaller steps, so no need to decrease α (learning rate) over & over again.

Linear Regression with multiple variables. (multiple features)

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots \theta_n x_n .$$

$$\therefore h_\theta(x) = \theta^T x .$$

$\theta \rightarrow \mathbb{R}^{n+1} - (n+1)$ variables.

cost function:-

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

& Gradient Descent function $\rightarrow$.

$$\theta_j = \theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

simultaneously update for every $j = 0, 1, 2 \dots n$.

for $n = 1$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta x^{(i)} - y^{(i)} \right) x_0^{(i)}$$

$$\theta_1 = \theta_1 - \alpha \frac{1}{m} \sum_{j=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_1^{(i)}$$

$\rightarrow x^{(i)}$ for variable x_1.

for $n \geq 1$

$$\theta_n = \theta_n - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_n^{(i)}$$

⇒ Features scaling:-

L Reducing the number of features or range of features

L To reach the minima within less steps or to converge easily.

⇒ Scaling range $-1 \leq x \leq 1$ - Scaled using average methods.

⇒ Features can be scaled using Mean Normalization.

$$x_1 = \frac{x_1 - \mu_1}{S_1}$$ $\mu_1 \rightarrow$ avg value of S_1

S_1 - range of values

* Normal Equations ::

L method to solve θ analytically. by using $\boxed{\theta = (X^T X)^{-1} X^T y}$

For 1D. ($\theta \in \mathbb{R}$) & $J(\theta) = a\theta^2 + b\theta + c$, to calculate the minima.

$$\frac{d}{d\theta} J(\theta) = 0 \quad \text{\& solve for } \theta.$$

For $\theta \in \mathbb{R}^{n+1}$ → multiple variables.

$$J(\theta_0, \theta_1, \theta_2 \ldots \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \ldots = 0$$ for every j, solve for $\theta_0, \theta_1, \theta_2 \ldots \theta_n$.

⑤

* Logistic Regression :- for classification problems

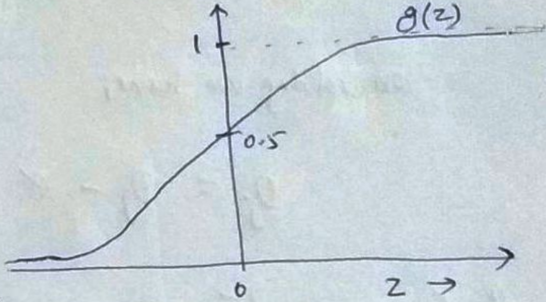
$$0 \leq h_\theta(x) \leq 1$$

So logistic regression function can be defined as :-

$$h_\theta(x) = g(z) \quad , \text{where} \quad g(z) = \frac{1}{1+e^{-z}}$$

if we have $z = \theta^T x$ for linear regression, we conclude that.

$$h_\theta(x) = \frac{1}{1+e^{-(\theta^T x)}}$$



$h_\theta(x)$ = estimated probability that $y=1$ on i/p x.

$h_\theta(x) = P(y=1 \mid x;\theta)$ Probability that $y=1$, given x parameterized by θ.

$$P(y=0 \mid x;\theta) + P(y=1 \mid x;\theta) = 1.$$

* Decision Boundary :-

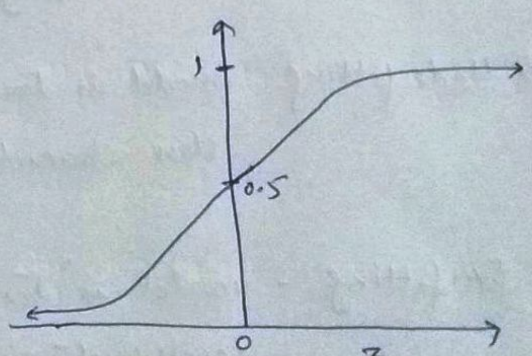
$h_\theta(x) = g(\theta^T x)$; $P(y=1 \mid x;\theta)$

for $y=1$ if $h_\theta(x) \geq 0.5$

$$\Rightarrow \boxed{\theta^T x \geq 0}$$

for $y=0$ if $h_\theta(x) < 0.5$

$$\Rightarrow \boxed{\theta^T x < 0}$$



* Cost function of Logistic Regression :-

$$\text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

6

for multiple values :-

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]$$

To fit parameter θ, minimize $J(\theta)$ over θ

⇒ Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

on solving we have;

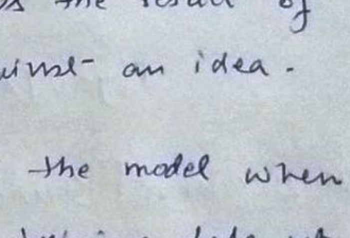
$$\theta_j = \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

* Components of generalization error :-

- Bias :- It is a phenomenon that skews the result of an algorithm in favour or against an idea.
- Variance :- It refers to the changes in the model when using different portions of training data set.

⇒ Underfitting : model is too "simple" to represent all the relevant class characteristics.

⇒ Overfitting :- model is "too complex" and fits irrelevant characteristics (i.e. noise) in the data



* Regularization :-

∟ keep all the parameters or features but reduce the magnitude or values of parameters θ.

∟ works well when we have lot of features, each of which contributed a bit to predicting y.

* Regularized cost functions:-

- Linear Regression:-

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{m} \theta_j^2\right]$$

- Gradient Descent:-

$$\theta_j = \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)} + \lambda/_m\, \theta_j\right]$$

- Normal Equation:-

$$\theta = \left(XX^T + \lambda \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{3\times 3}\right)^{-1} X^T y.$$

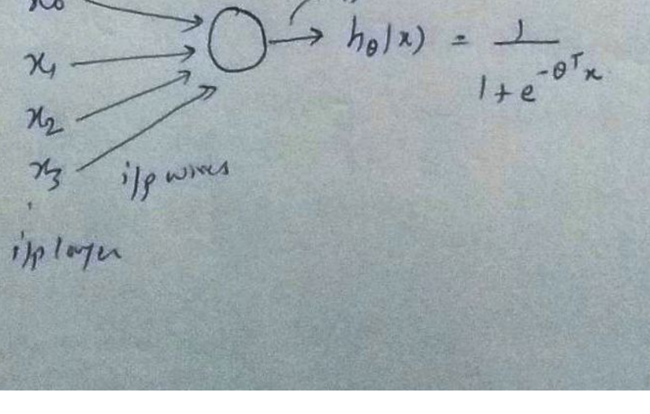
for $n=2$, matrix will be $(n+1)(n+1)$

- logistic Regression:-

$$J(\theta) = \left[-\frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log\left(h_\theta(x^{(i)})\right) + \left(1-y^{(i)}\right)\log\left(1-h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

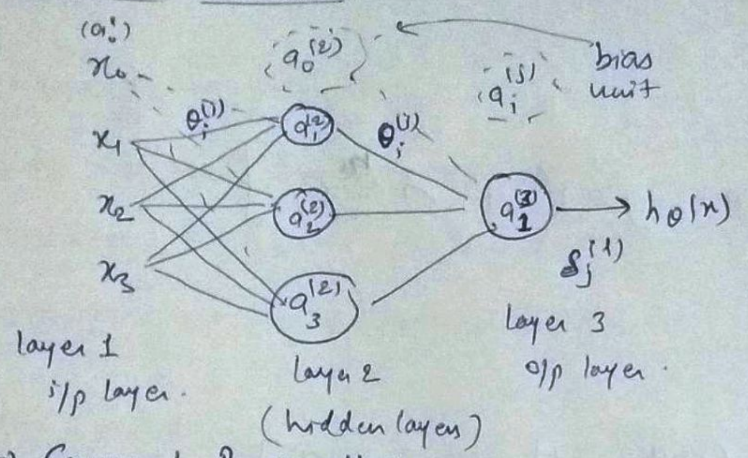
* Neural Network Representation

L Model Representation : Logistic unit



⑧

⇒ Neural Networks :-



⇒ Forward Propagation.

$a^{(1)} = x \quad ; \quad z^2 = \theta' x$

$z^2 = \theta' a^{(1)} \quad , \quad a^{(2)} = g(z^2) \Rightarrow a^{(2)} = \theta' a^1$

$a_1^{(2)} = g(\theta_{10}^1 x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^1 x_3)$

& so on . . . . .

$h_\theta(x) = a_1^{(3)} = g(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)})$

⇒ Back propagation :- minimizing error or gradient computing :-

$\delta_j^{(l)}$ = error of node j in layer l

eg. $\delta_j^{(3)} = a_j^{(3)} - y_j$ or $\delta^{(3)} = a^{(3)} - y$.

$\delta^{(2)} = (\theta^{(2)})^T \delta^{(3)} * g'(z^{(2)})$