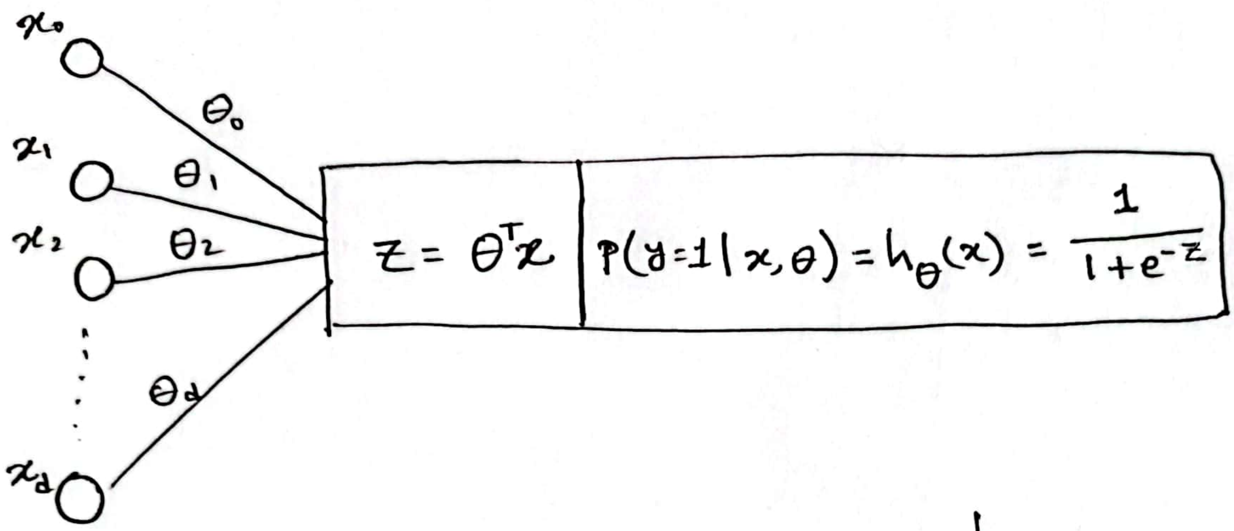
Logistic Regression



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

$$\hat{y} = \begin{cases} 1 & , \text{if } h_\theta(x) \geq 0.5 \\ 0 & , \text{if } h_\theta(x) < 0.5 \end{cases}$$

$$\hat{y} = \begin{cases} 1 & , \text{if } z \geq 0 \\ 0 & , \text{if } z < 0 \end{cases}$$

$\boxed{**}$ cost function for i^{th} sample: (Binary cross entropy loss)

$$J_i(\theta) = \begin{cases} -\log_e\left(h_\theta(x)\right), & \text{if } y = 1 \\ -\log_e\left(1 - h_\theta(x)\right), & \text{if } y = 0 \end{cases}$$

$$\hookrightarrow J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[y^i \times \log_e\left(h_\theta(x^i)\right) + (1-y^i)\log_e\left(1 - h_\theta(x^i)\right)\right]$$

Lecture - 11

P1

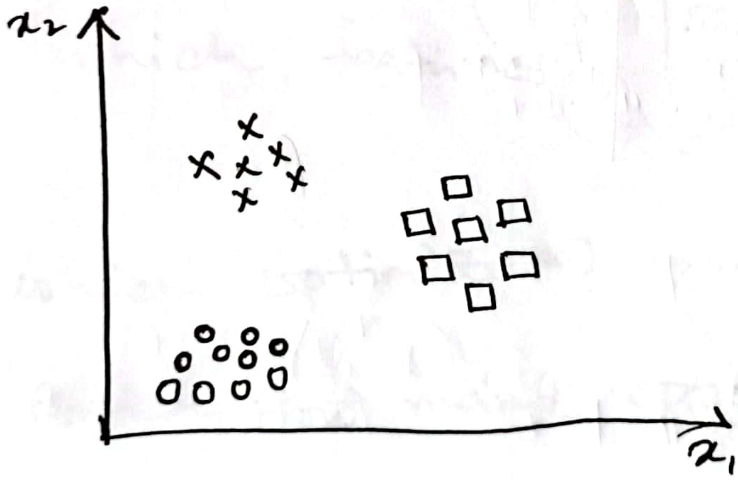
Rev.

P2

Multiclass classification

Email : Friends (0) / Colisgue (1) / Spam (2)

image : human (0) / horse (1) / cat (2) / dog (3)



$$y = \{0, 1, 2\}$$

o x □

LR এর extention

$y = \{0, 1, 2\}$

** Algo-1   One vs all

idea: Train a logistic regression classifier $\boxed{h_{\theta_i}^{(i)}}$ for each class i to predict the probability of $\boxed{y = i}$

3 টো θ create করবো।

$(h_{\theta_0}^{(0)}) = \theta_1^T x$ $(h_{\theta_1}^{(1)}) = \theta_1^T x$ $(h_{\theta_2}^{(2)}) = \theta_2^T x$

$\{\theta_{00}, \theta_{01}, \ldots, \theta_{0d}\}, \{\theta_{10}, \theta_{11}, \ldots, \theta_{1d}\}, \{\theta_{20}, \theta_{21}, \ldots, \theta_{2d}\}$

- $\{\theta_{00}, \ldots, \theta_{0d}\}$: এটা input দিলে output দিবে যে ~~[illegible]~~ একটা data x '0' হওয়ার probability কত
- $\{\theta_{10}, \ldots, \theta_{1d}\}$: এটা input দিলে আমাদের Data x '1' হবার probability কত
- $\{\theta_{20}, \ldots, \theta_{2d}\}$: Data x এর '2' হবার probability কত

So,

$$h_{\theta_0}^{(0)} = 0.2$$
$$h_{\theta_1}^{(1)} = 0.1$$
$$h_{\theta_2}^{(2)} = 0.9$$

যেটার probability বেশি Data x ঐ class এ পড়বে,

$$\boxed{y_{pred} = \arg\max h^{(i)}_{\theta_i}(x)}$$

$$h^{(0)}_{\theta_0}(x) = g(\theta_0^T x) = \frac{1}{1+e^{-\theta_0^T x}} = P(y=0 \mid x; \theta_0)$$

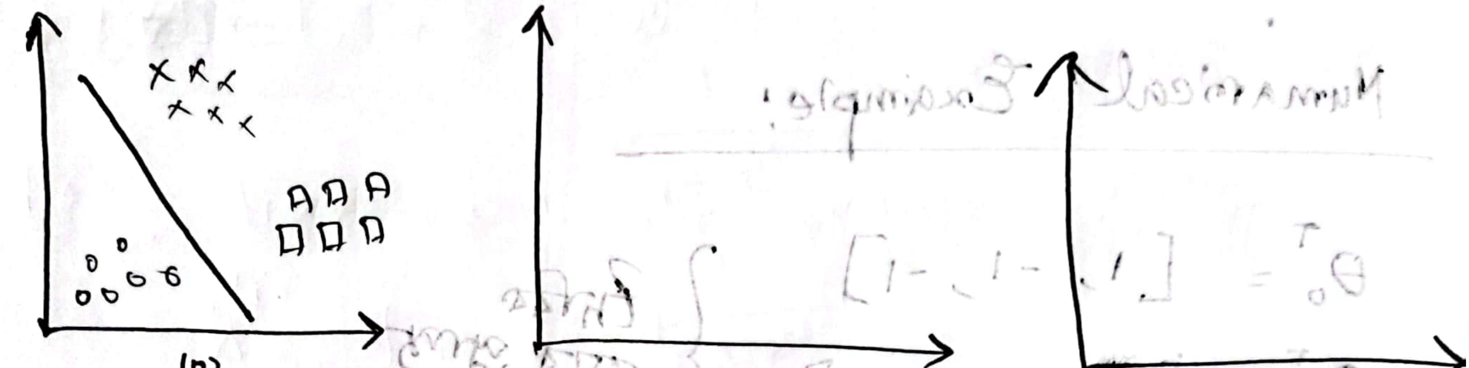
$[\theta_{00}, \theta_{01}, \theta_{02}, \ldots, \theta_{0d}]$

$$h^{(1)}_{\theta_1}(x) = g(\theta_1^T x) = \frac{1}{1+e^{-\theta_1^T x}} = P(y=1 \mid x, \theta_1)$$

$[\theta_{10}, \theta_{11}, \theta_{12}, \ldots, \theta_{1d}]$

$$h^{(2)}_{\theta_2}(x) = g(\theta_2^T x) = \frac{1}{1+e^{\theta_2^T x}} = P(y=2 \mid x, \theta_2)$$

$[\theta_{20}, \theta_{21}, \theta_{22}, \ldots, \theta_{2d}]$

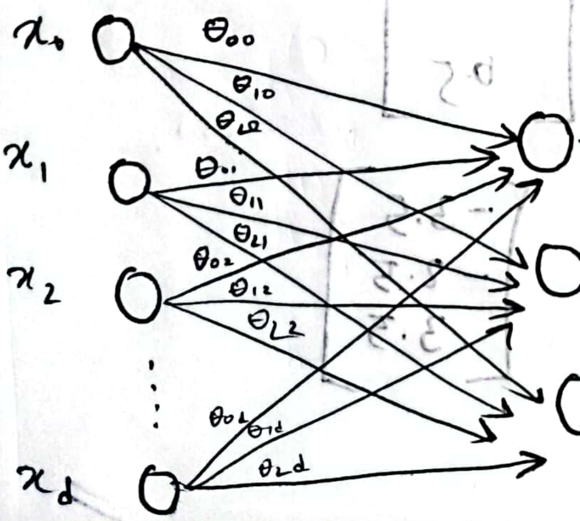


$h^{(0)}_{\theta_0}$

vectorized:

$$W = \begin{bmatrix} \theta_0^T \\ \theta_1^T \\ \theta_2^T \end{bmatrix} = \begin{bmatrix} \theta_{00}, \theta_{01}, \dots, \theta_{0d} \\ \theta_{10}, \theta_{11}, \dots, \theta_{1d} \\ \theta_{20}, \theta_{22}, \dots, \theta_{2d} \end{bmatrix}$$

$$h_\theta(x) = g(Wx) = g\left(\begin{bmatrix} \theta_0^T x \\ \theta_1^T x \\ \theta_2^T x \end{bmatrix}\right) = \begin{bmatrix} g(\theta_0^T x) \\ g(\theta_1^T x) \\ g(\theta_2^T x) \end{bmatrix}.$$



$h^{(1)}_{\theta_1}(x) = [0, 1]$ ⇝ 0 থেকে 1 এর মধ্যে একটি value দিবে

$h^{(2)}_{\theta_2}(x) = [0, 1]$

$h^{(3)}_{\theta_3}(x) = [0, 1]$

Numarical Example:

$$\left.\begin{aligned} \theta_0^T &= [1, -1, -1] \\ \theta_1^T &= [-1, -0.5, 1] \\ \theta_2^T &= [-4, 2, -1] \end{aligned}\right\}$$ ট্রেনিং থেকে প্রাপ্ত

$$x = \begin{bmatrix} 1 \\ 4 \\ 0.5 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & -1 & -1 \\ -1 & -0.5 & 1 \\ -4 & 2 & -1 \end{bmatrix}$$

$$\therefore z_1 = Wx$$

$$= \begin{bmatrix} 1 & -1 & -1 \\ -1 & -0.5 & 1 \\ -4 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \\ 0.5 \end{bmatrix}$$

$$= \begin{bmatrix} 1-4-0.5 \\ -1-2+0.5 \\ -4+8-0.5 \end{bmatrix} = \begin{bmatrix} -3.5 \\ -2.5 \\ 3.5 \end{bmatrix}$$

$$g(z) = \begin{bmatrix} \dfrac{1}{1+e^{3.5}} \\ \dfrac{1}{1+e^{2.5}} \\ \dfrac{1}{1+e^{-3.5}} \end{bmatrix} = \begin{bmatrix} 0.03 \\ 0.08 \\ 0.97 \end{bmatrix}$$

$\therefore y_{pred} = 2$

[**] How to get $\theta_0, \theta_1, \theta_2$?

3 errorfne $= J^{(0)}(\theta_0)$, $J^{(1)}(\theta_1)$, $J^{(2)}(\theta_2)$

Then optimize by : G.D / Adam / ... (seperately)

** Problems of one vs all classifier:

① Have to train seperately

② No probabilistic interpetation for $h_\theta(x)$

[sum(h) > 1]

সবগুলোর sum = 1 হয় না,
যদি each class এর probability দিত
তাহলে sum = 1 হতো

** Solution:

① we ~~want~~ want sum(h) = 1

② we want end to end learning

| 3টা seperate classifier train করবো না, একটা করবো যেটা সবগুলোকে classify করবে।

Softmax classifier

⇒ modification to the Previous algorithm:

① Sigmoid function to softmax function

② change y to one-hot vector.

③ from $J(\theta) = \underbrace{\overset{\text{Binary}\uparrow}{BCE}}_{\text{Cross entropy}}$ to $\overset{\text{Cross Entropy}}{CE}$

$① \Rightarrow$

$\Rightarrow$ $Wx = [z_0, z_1, \ldots z_{k-1}]$

$(k \times d{+}1)((d{+}1) \times 1)$

$(k \times 1)$

$$\text{sigmoid}(Wx)^T = \left[\frac{1}{1+e^{-z_0}}, \frac{1}{1+e^{-z_1}}, \ldots, \right]$$

$\Rightarrow$ Softmax function, $f_i = \dfrac{e^{z_i}}{\sum_{i=0}^{k-1} e^{z_i}}$

$$\text{Softmax}(Wx)^T = [f_0, f_1, \ldots, f_{k-1}]$$

$$f_1 = \frac{e^{z_1}}{e^{z_0} + e^{z_1} + e^{z_2}}$$

Probability

$$\left(\sum_{i=0}^{k-1} f_i = 1\right)$$

2 => one hot ~~vector~~ encoding

$$y = \{0, 1, 2\} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

3 =>

cross entropy loss,

$$J(w) = -\frac{1}{n} \underbrace{\sum_{i=1}^{n}}_{\text{avg over all training samples}} \underbrace{\sum_{k=0}^{k-1} y_k^{(i)} \log f_k}_{J_i(w)}$$

Optimizer => G.D

w^0 = random

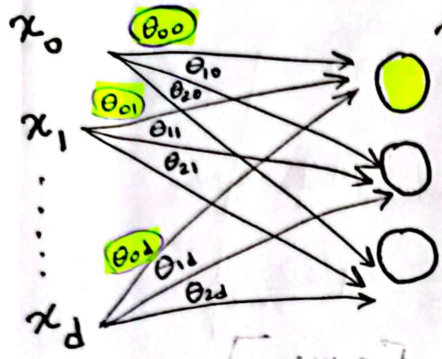
while not converged:

$w^{(k+1)} = w^k - \nabla_w J$

if $\nabla_w J \approx 0$:

converged == True

P5



$$\underbrace{\begin{bmatrix} \theta_{00} & \theta_{01} & \cdots & \theta_{0d} \\ \theta_{10} & \theta_{11} & \cdots & \theta_{1d} \\ \theta_{20} & \theta_{21} & \cdots & \theta_{2d} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}}_{Wx} = \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} \leftarrow g(z) \text{ [Softmax]}$$

$\theta_0 \uparrow \quad \theta_1 \uparrow \quad \theta_d \uparrow$

$(3, d) \times (d, 1) \qquad (3\times 1)$

$$\theta_0 = [0, 0.01, -0.05, 0.1, 0.05]$$
$$\theta_1 = [0.2, 0.7, 0.2, 0.05, 0.16]$$
$$\theta_2 = [-0.3, 0.0, -0.45, -0.2, 0.33]$$
$$x = [1, 15, 22, 44, 56], \quad y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Given

(Matrix দিয়ে করলে) $z = [Wx]$

$$z_0 = 0\times 1 - 0.01\times 15 - 0.05\times 22 - 0.1\times 44 + 0.05\times 56$$
$$= -2.58$$
$$z_1 = 0.02\times 1 - 0.7\times 15 + 0.2\times 22 - 0.05\times 44 + 0.16\times 56$$
$$= 0.86$$
$$z_2 = -0.3\times 1 - 0\times 15 - 0.45\times 22 + 0.2\times 44 + 0.33\times 56$$
$$= 0.28$$

$$\therefore \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -2.58 \\ 0.86 \\ 0.28 \end{bmatrix}$$

$$\therefore f\left(\begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix}\right) = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \dfrac{e^{-2.58}}{e^{-2.58} + e^{0.86} + e^{0.28}} \\ \dfrac{e^{0.86}}{e^{-2.58} + e^{0.86} + e^{0.28}} \\ \dfrac{e^{0.28}}{e^{-2.58} + e^{0.86} + e^{0.28}} \end{bmatrix} = \begin{bmatrix} 0.016 \\ 0.631 \\ 0.353 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} y_{pred}$$

Now, $y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, $y_{pred} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.016 \\ 0.631 \\ 0.323 \end{bmatrix}$

$$\therefore J_0(w) = -\sum_{k=0}^{K-1} y_k \log f_k$$

$$= -[y_0 \times \log f_0 + y_1 \times \log f_1 + y_2 \times \log f_2]$$

x_0 এর জন্য error

$$= -(0 \times \log(0.016) + 0 \times \log(0.631) + 1 \times \log(0.323))$$

$$= -\log(0.323) = 1.04$$

(Suppose)

$\therefore$ x_1 এর জন্য, $J_1(w) = 2.06$

$\therefore$ x_2 এর জন্য, $J_2(w) = 0.08$

$\therefore$ x_3 এর জন্য, $J_3(w) = 1.34$

$$\therefore J(w) = \frac{1}{n}\sum_{i=0}^{n-1}\left(-\sum_{k=0}^{K-1} y_k \log f_k\right)$$

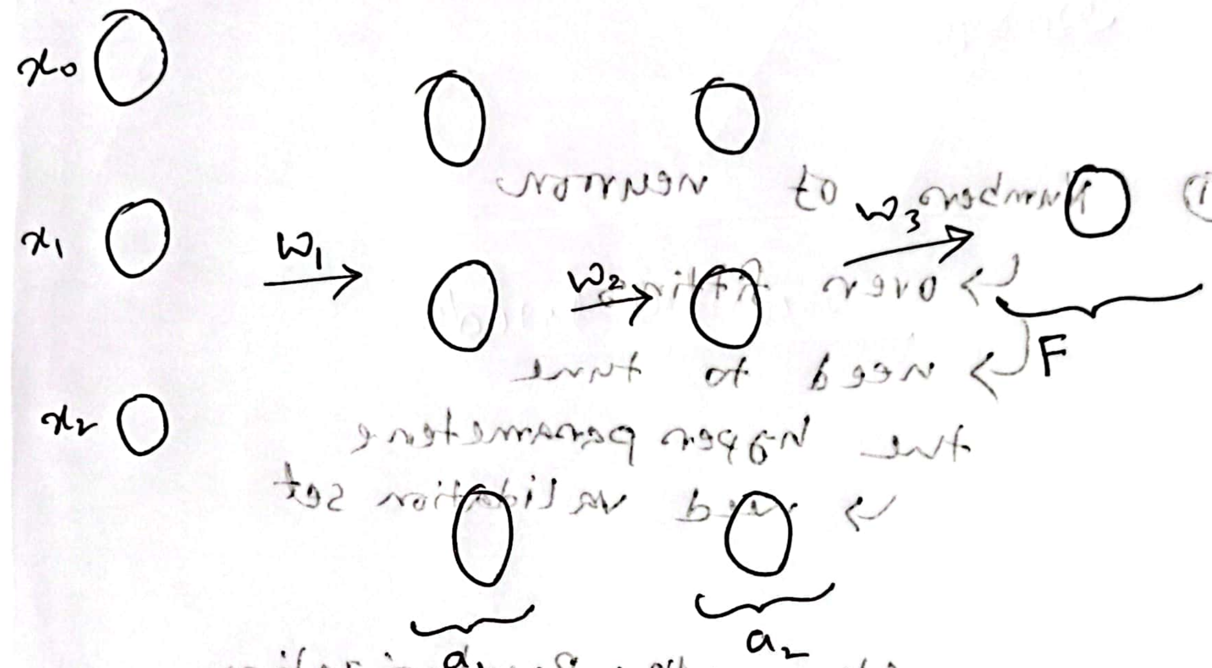
$n \rightarrow$ numbers of samples

$K \rightarrow$ number of class

$$= \frac{1}{4}\left[J_0(w) + J_1(w) + J_2(w) + J_3(w)\right]$$

$$= \frac{1}{4}(1.04 + 2.06 + 0.08 + 1.34) = 1.13$$

NN



$$a_1 = g(w_1 x)$$

$$a_2 = g(w_2 a_1)$$

$g(z)$ = some non-linear function
= sigmoid

$$F = f(\underbrace{w_3 a_2}_{z})$$

$f(z)$ = softmax

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[- \sum_{k=0}^{k-1} y_k^{(i)} \log f_k \right]$$

↳ minimize using GD

Practical consideration

① Numbers of neuron

↳ over fitting

↳ need to tune the hyper parametene

↳ need validation set

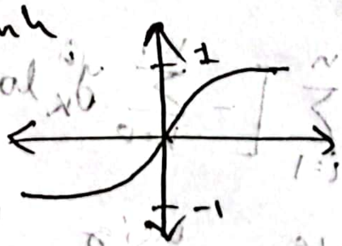
② over fiting solⁿ : Regularization

$J(\theta) + \lambda \sum |\theta|^2$

③ activation function :

sigmoid saturates → y is +ve

↳ use tanh



Relu(Z) → Z , if Z ⩾ 0
↳ 0 , otherwise