☆ WEEK 2

Binary classification.

- we're doing cat classification
- final answer will be whether input image is of cat or no

Notation.

~~input consists of~~
given data consisting of (x, y)

$(x, y) \rightarrow x \in R^{n_x}$, $y \{0, 1\}$ no yes

x ↓ vector in n dimensions

m training examples : $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots (x^{(m)}, y^{(m)})\}$
↓ m_{train} testing data = m_{test}

we combine all the input data (x) into a single matrix

$$X = \begin{bmatrix} \vdots & \vdots & & \vdots \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ \vdots & \vdots & & \vdots \end{bmatrix}$$
↓
$n_x m$ matrix

how each input is represented as a coloumn
At some places it can also be represented as a row of matrix
But coloumn representation is better.

similarly

$$Y = [y^{(1)}, y^{(2)} \cdots y^{(m)}]$$
↓
$1 \times m$ matrix

Logistic Regression. – (used for binary classification problems)

Given x, we want $\hat{y} = P(y=1/x)$

given x find probability of cat.

$x \in R^n$

parameters : $w \in R^n$, $b \in R$

output $\hat{y} = \sigma(\underbrace{w^T x + b}_{z})$ (sigmoid bcz $w^T x + b$ can be outside of [0,1])

$w_i = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$ → parameter coming from $x_1, x_2 \ldots x_n$ to a_1

centre of universe

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Logistic Regression, Cost function

we can do Loss $f^n = \frac{1}{2}(\hat{y} - y)^2$

predicted actual

Gradient descent doesn't work well

↳ but in logistic regression we usually dont do this bcz it has multiple local minimas so Logistic (graph is non-convex) regression me Gradient descent usually doont find the principal minima.

$$\text{Loss } f^n = -\left(y \log \hat{y} + (1-y)\log(1-\hat{y})\right)$$

↓

we want to minimize

eg:- $y = 1$ loss $f^n = -\log \hat{y}$

minimize ⟶ maximize $\log \hat{y}$

↓

maximize $\hat{y}$

↓

since we take sigmoid max $\hat{y}$ can be 1 only

If $y=0$ $\quad$ loss $= -\log(1-\hat{y})$

$\downarrow$

minimize $\rightarrow$ maximize $(1-\hat{y})$

$\downarrow$

minimize $\hat{y}$ $\rightarrow$ min value of y is 0 bz sigmoid

Loss f^n $\rightarrow$ measures how well you're doing on one training example

Cost f^n $\rightarrow$ measures how well you're doing on entire training set

$$\text{Cost } f^n = J(w,b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)})$$

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