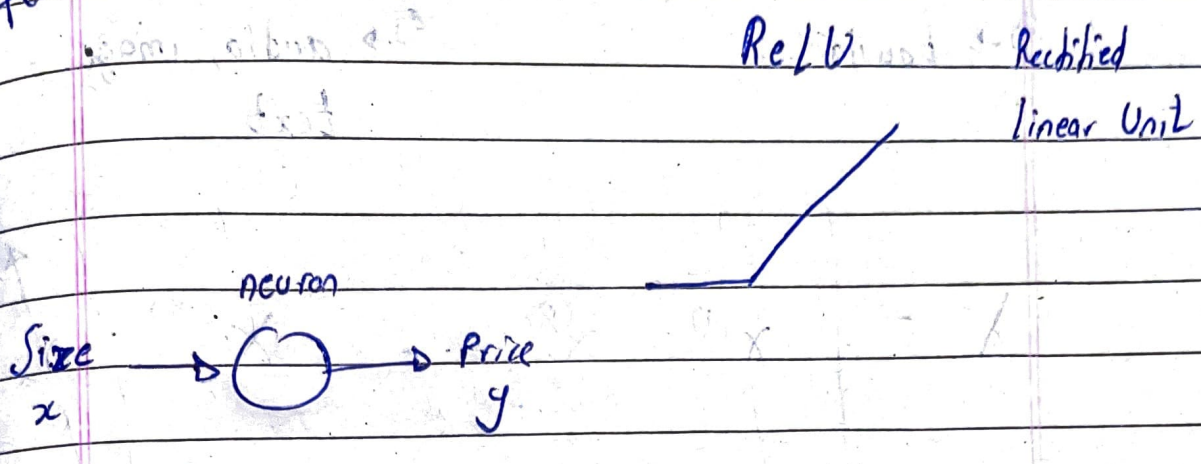
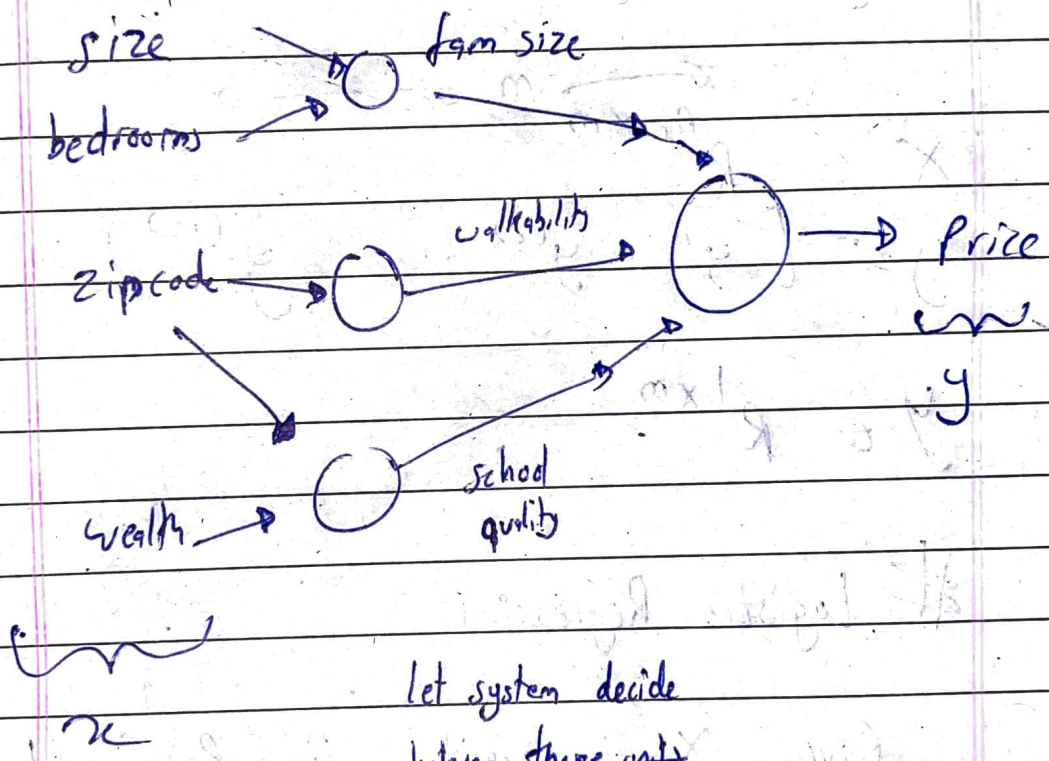
Neural Network





let system decide whatever these ~~units~~ hidden units are

Structered data
eg → tabular

Unstructured
eg → audio, image, text

$$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix} \quad n_x$$

m

$n_x \times m$

$X \in R^{n_x \times m}$

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

$y \in R^{1 \times m}$

Logistic Regression

Given X, want $\hat{y} = P(y=1|x)$

Parameters $w \in R^{n_x}$ $b \in R$

① Output $\hat{y} = \sigma(w^T x + b)$ sigmoid

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n_x} \end{bmatrix} \begin{matrix} \} b \\ \} w \end{matrix}$$

if $x_0 = 1$
$x \in R^{n_x+1}$

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

$\hat{y} = \sigma(w^T x + b)$

logistic error function

$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$ (concept)

$L(\hat{y}, y) = -(y \log \hat{y} + (1-y) \log(1-\hat{y}))$

- not using squared errors as it will minimize to 0 only
- if $y=1$ $L = -\log \hat{y}$ ← want $\log \hat{y}$ large
  want $\hat{y}$ large
  $\hat{y}$ can only be 1 max
  bc sigmoid
  $y=0$ $L = -\log(1-\hat{y})$ ← want $\hat{y}$ small
  $\hat{y}$ min → zero
  bc sigmoid

Cost Func

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

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

Gradient Descent Algo

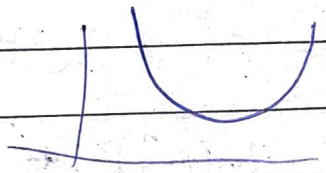
$$\hat{y} = \sigma(w^T x + b) \qquad \sigma(z) = \frac{1}{1+e^{-z}}$$

$$J(w,b) = \frac{1}{m} \sum L(\hat{y}^{(i)}, y^{(i)})$$

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

find w, b that min J

- Convex function → single bowl
  - → single local optima



- Doesn't usually matter where you ~~start~~

Repeat {

$$w := w - \alpha \frac{dJ(w,b)}{dw}$$

$$b := b - \alpha \frac{dJ(w,b)}{db}$$

}

learning rate

α → ~~fear~~ like step size

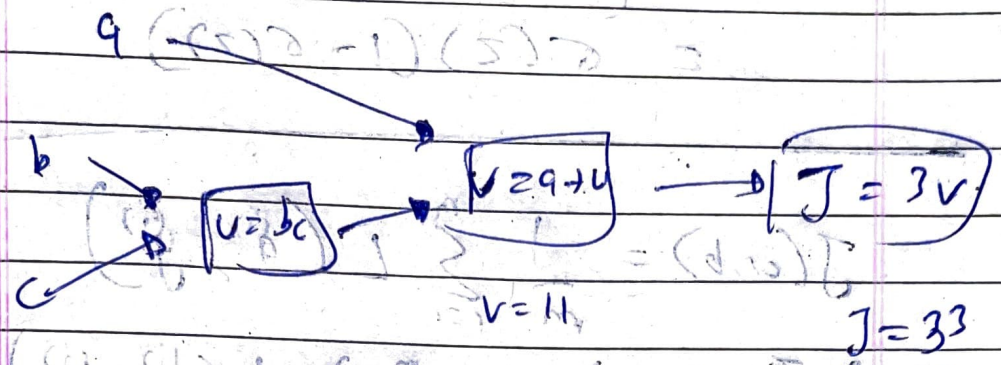
Computation Graph

$$J(a,b,c) = 3(a + bc)$$

$u = bc$

$v = a + u$

$J = 3v$

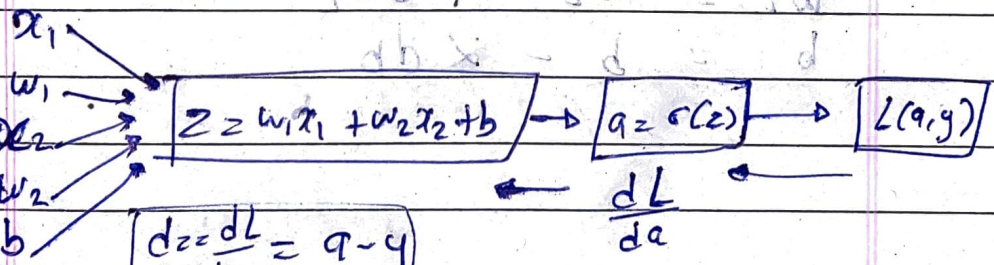


$$\frac{dJ}{dv} = 3$$

$$\frac{dJ}{da} = 3 = \frac{dJ}{dv}\frac{dv}{da}$$

In code $\frac{d(\text{Final Output Var})}{d\text{var}} =$ "dvar"

Logistic regression derivatives



$$= -\frac{y}{a} + \frac{1-y}{1-a} = \frac{a-y}{a(1-a)}$$

$$\frac{dL}{dw_1} = x_1\,dz$$

$$\frac{dL}{db} = dz$$

$$\frac{dL}{w_2} = x_2\,dz$$

$$\frac{d\sigma(z)}{dz} = \frac{e^{-z}}{(1+e^{-z})^2}$$

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

$$= \frac{1}{(1+e^{-z})}\left[\frac{1+e^{-z}}{1+e^{-z}} - \frac{1}{1+e^{-z}}\right]$$

$$= \sigma(z)(1-\sigma(z))$$

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(a^{(i)}, y^{(i)})$$

$$\frac{\partial}{\partial w_1} J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{\frac{\partial}{\partial w_1} L(a^{(i)}, y^{(i)})}_{dw_1^{(i)} = (x^{(i)}, y^{(i)})}$$

$$dw_1 = \frac{dJ}{dw_1}$$

$$w_1 := w_1 - \alpha\, dw_1$$

$$w_2 := w_2 - \alpha\, dw_2$$

$$b := b - \alpha\, db$$

Vectorization

$w \in R^{n_x}$ $X \in R^{n_x \times m}$

non vect

```
z = 0
for i in range
    z += w[i] * x[i]
z += b
```

vect

$z = np.dot(w, x) + b$

$w^T x$

better to built in func

$$Z = [z^{(1)}\ z^{(2)} \dots z^{(m)}]$$

$$= w^T X + \underbrace{[b, b, b \dots b]}_{1 \times m}$$

$$Z = np.dot(w.T, X) + b$$

Broadcasting

$$A = \sigma(Z)$$

$$dZ = A - Y$$

$$= [a^1 - y^1,\ a^2 - y^2 \dots]$$

$$db = \frac{1}{m} \sum_{i=1}^{m} dz^{(i)}$$

$$db = \frac{1}{m} np.sum(dZ)$$

$$dw = \frac{1}{m} X \cdot dZ^T$$

PAGE NO.
DATE ../../..

Broadcasting

(m, n) + (1, n) → (m, n)
matrix

×
/ (m, 1) → (m, n)

(m, 1) + R = [101, 102, 103]
[1, 2, 3]
100

- Don't use Rank 1 arrays
ie when
~~a.shape = (5,)~~

logistic Reg cost function

y=1 p(y|x) = ŷ
y=0 p(y/x) = 1−ŷ

$p(y|x) = \hat{y}^{y} \cdot (1-\hat{y})^{(1-y)}$

↑ log(P(y|x)) = −L(ŷ, y) ↓