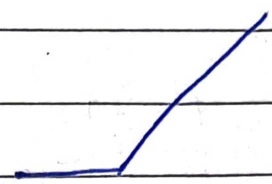
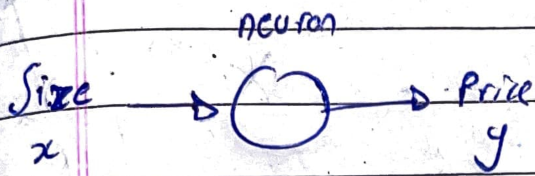
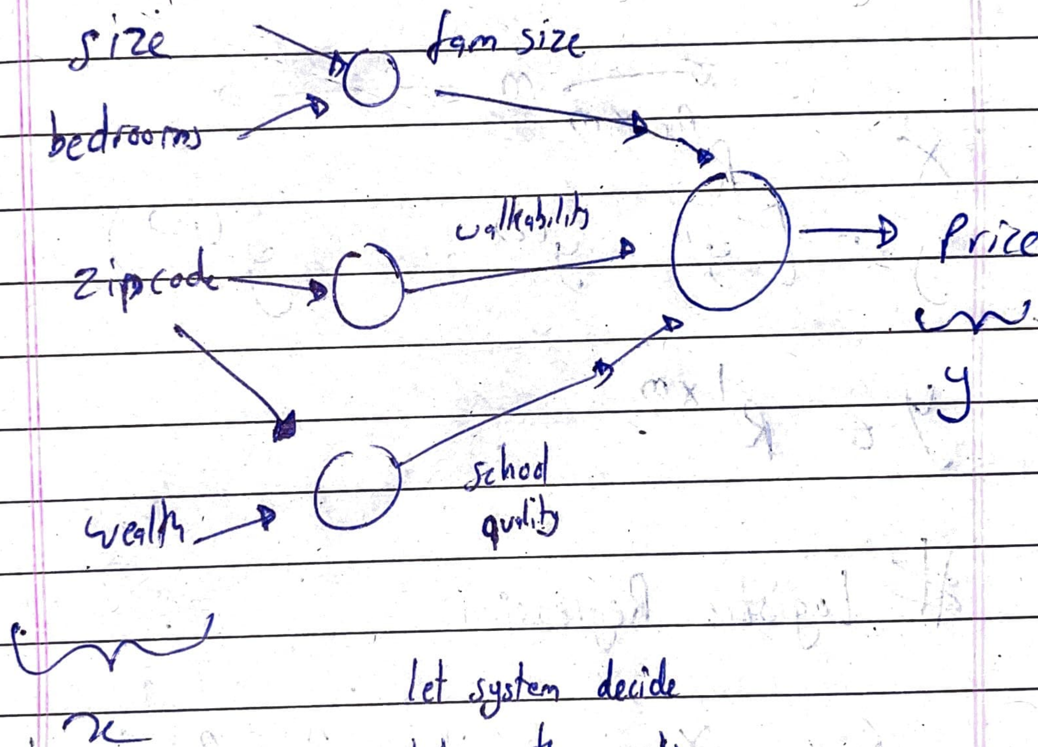
Neural Network

ReLU — Rectified linear Unit







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

Structered data

eg → tabular

Unstructured

eg → audio, image, text

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

$$\longleftarrow m \longrightarrow$$

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

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

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

Logistic Regression

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

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

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

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n_x} \end{bmatrix} \begin{matrix} \}b \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\}\omega \end{matrix}$$

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 $y = 1$ $\quad L = -\log \hat{y}$ ← want $\log \hat{y}$ large
  want $\hat{y}$ large
  $\hat{y}$ can ~~only~~ only be 1 max
  bc sigmoid

  $y = 0$ $\quad L = -\log(1-\hat{y})$ ← 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

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

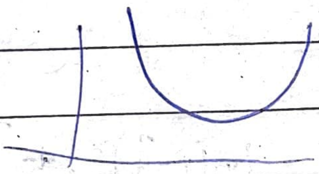
$J(w,b) = \frac{1}{m} \sum L(\hat{y}^{(i)}, y^{(i)})$

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

find w, b that min J

- Convex function → single bowl

→ single local optima



- Doesn't usually matter where you start

Repeat {

learning rate

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

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

}

tear

α → like step size

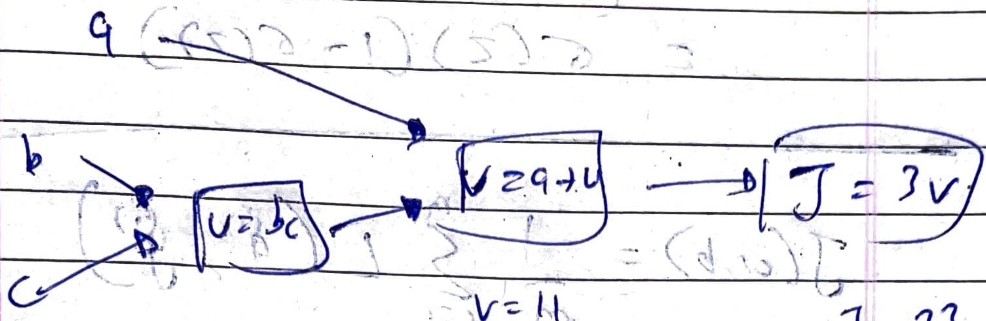
Computation Graph

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

$u = bc$

$V = a+u$

$J = 3v$



$v = 11$ $\qquad J = 33$

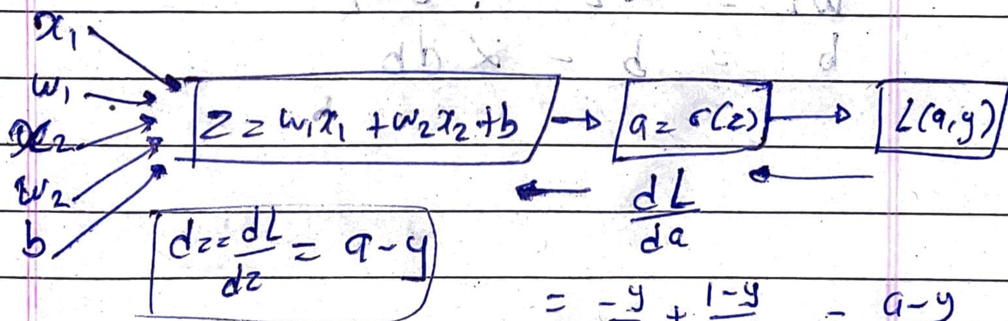
$V = 11.001$ $\qquad J = 33.003$

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

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

In code $\quad \dfrac{d\,\text{Final Output Var}}{d\text{var}} = $ "dvar"

Logistic regression derivatives



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

$$\frac{dL}{dw_1} = x_1 dz \qquad \frac{dL}{db} = dz$$

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

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

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

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

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

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

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

$$\partial w_1 = \frac{\partial J}{\partial w_1}$$

$$w_1 := w_1 - \alpha\, dw_1$$
$$w_2 := w_2 - \alpha\, dw_2$$
$$b := b - \alpha\, db$$

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

$z = \underbrace{np.dot(w, x)}_{w^T x} + b$

better to built in func