Logistic regression as neural network

Logistic regression is a binary classification algorithm.

Let, we have a image. we have to classify if it's a cat or not.

(x, y) , $x \in \mathbb{R}^{n_x}$, $y \in \{0, 1\}$.

m = Numer of train example = M_{train} = $\{(x^{(1)}, y^{(1)}), (x^{(11)}, y^{(11)}) \ldots\}$

Numer of test example = M_{test}

← grouped train example

X = contains all the train data.

$$= \underbrace{\left[\begin{array}{cccc} | & | & & | \\ x^{(1)} & x^{(11)} & \cdots & x^{(m)} \\ | & | & & | \end{array}\right]}_{m} \Big\updownarrow n_x \quad \in R^{n_x \times m}$$

$$\begin{bmatrix} - - x^{(1)^T} - - \\ - - x^{(11)^T} - - \\ \vdots \\ - - x^{m^T} - - \end{bmatrix}$$ ⇒ not feasible in NN.

$Y = [y^{(1)} \quad y^{(2)} \quad y^{(3)} \quad \ldots \quad y^{(m)}] \in \mathbb{R}^{1 \times m}$

Logistic regression:-

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

$x \in \mathbb{R}^{n_x}$

Parameter $w \in R^{n_x}$, $b \in R$

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

$\sigma_z = \dfrac{1}{1 + e^{-z}}$

↑ sigmoid function

Given,

$((x^1, y^1), (x^2, y^2) \ldots\ldots (x^m, y^m))$ we want $\hat{y}^i \approx y^i$

Loss function (error function):-

$\mathcal{L}(\hat{y}, y) = \underbrace{\frac{1}{2}(\hat{y} - y)^2}_{\text{as small as possible}}$ → this doesn't work well in logistic regression

$\mathcal{L}(\hat{y}, y) = -\left(y \log \hat{y} + (1-y) \log(1-\hat{y})\right)$ → we use this as Logistic regression cost function convex

if $y = 1$ $\mathcal{L}(\hat{y}, 1) = -\log \hat{y}$ — we want this small so $\log \hat{y}$ should be large so $\hat{y}$ should be large

$y = 0$ $\mathcal{L}(\hat{y}, y) = -\log(1-\hat{y})$

Loss function is for single train example.

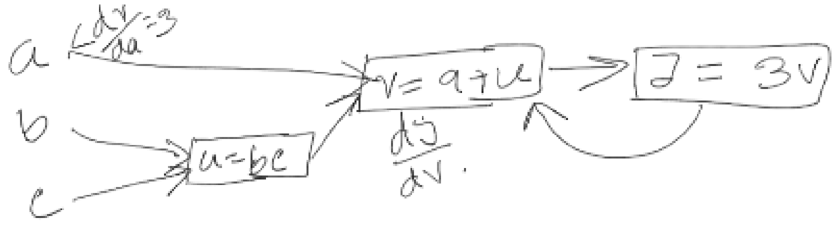
cost function is for whole train example

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

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

Computation graph:-

$J = 3(a+bc)$



$\frac{dJ}{dv} = ?\ 3 \quad (J = 3v) = $ "dv" ⇒ denotion in code.

$\frac{dJ}{da} = \frac{dJ}{dv} \cdot \frac{dv}{da} = 3 \times 1$.