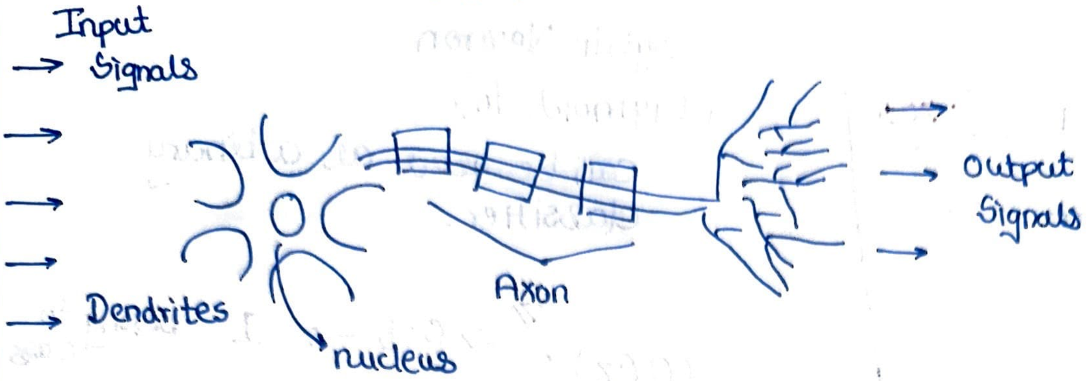
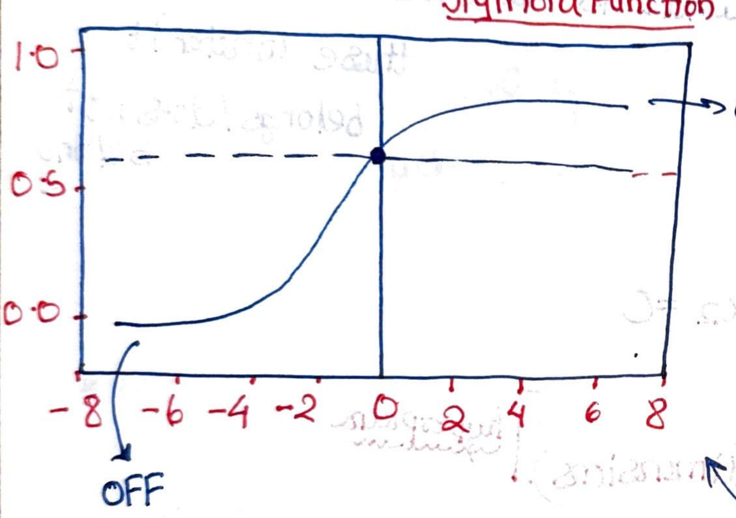
Week 3 - Pt2

Logistic Regression



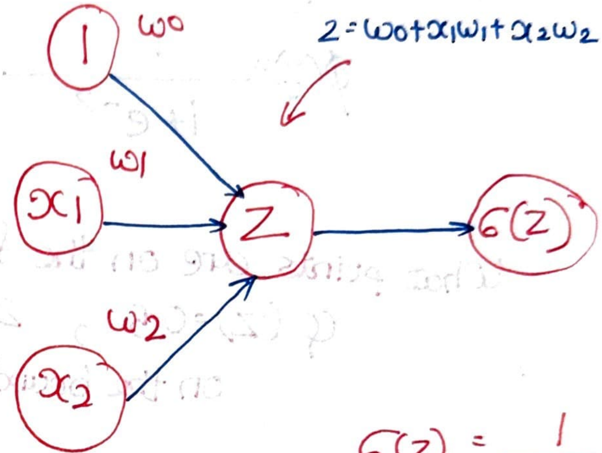
$$\phi(z) = \frac{1}{1+e^{-z}}$$

Sigmoid Function



net input

$z = w_0 + x_1 w_1 + x_2 w_2$



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

in reality biological neurons dont just have 0 or 1

They have continuous like

Sigmoid function

it imitates the on/off values

→ Logistic Neuron can be used as a binary Classifier

Activation.

$\varphi(z) > 0.5 \rightarrow 1$

$\varphi(z) < 0.5 \rightarrow 0$

Probability → The input has a Property

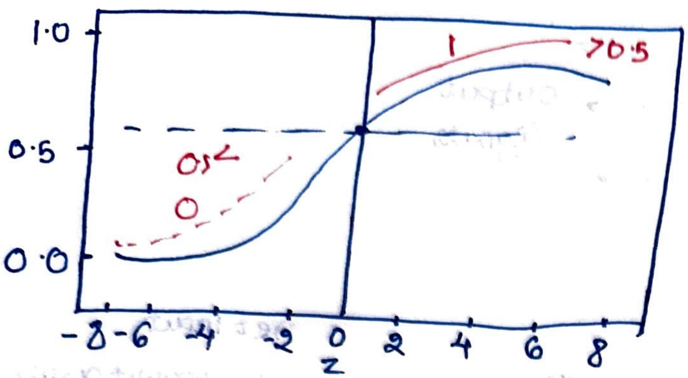
The input is a dog/not

human/dog

Input has a property

Properties of Logistic Neuron :-

Binary Classifier / Linear Classifier



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

Logistic Neuron
(Sigmoid fn)
can be used as a binary classifier.

$\varphi(z) \xrightarrow{\geq} 0.5 \rightarrow 1$ belong to class

$\xrightarrow{<} 0.5 \rightarrow 0$ not belongs to class

What is the decision boundary b/w these whether it belongs / does not belong

What points are on the fence?

$\varphi(z) = 0.5$, $z = 0$

on the boundary.

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

$$\rightarrow w_0 + x w^T = 0$$

(in higher dimensions)] hyperplane

(2) Why we look into the output of logistic neuron / sigmoid function and interpret it as probability?

$P \rightarrow \text{odds}(P) = \frac{P}{1-P} \rightarrow \text{logit}(P)$

P ← probability

$1-P$ (complement)

$[0,1]$ $[0,\infty]$

$$= \log\left(\frac{P}{1-P}\right)$$

$[-\infty, \infty]$

z : logit of some P

$$\log \frac{P}{1-P} = z \Rightarrow P = \frac{1}{1+\exp(-z)}$$

The loss function (negative log likelihood)

$$L(x, y, w)$$ Data, label, weights

→ Find w that minimizes the Loss function.

likelihood x

$X \rightarrow Z \rightarrow \varphi(z)$ = p : probability that X is in class 1

$z = w_0 + w_1 x_1 \ldots$

activation func

$\varphi(z) = \hat{y}$

$X \rightarrow$ true label y

→ predicted → $\hat{y}$

probability = 0.8

80% good

but 100% would be better.

* $y \rightarrow$ true label
* $\hat{y} \rightarrow p \rightarrow$ predicted (probability)

$y_T = 1, \; p = 0.8$

$y = 0, \; p = 0.2$ I am 80% good as per the complement

$$\text{likelihood}(x) = \hat{y}^{\,y}(1-\hat{y})^{(1-y)}$$

when $y = 1$ $\hat{y} = 0.8$

$$\text{likelihood}(x) = \hat{y}^{\,1}(1-\hat{y})^{(1-1)}$$ $\hat{y} = 0.8$

$$= 0.8^1$$

$y = 0, \; \hat{y} = 0.2$

$$\text{likeli} = \hat{y}^{\,0}(1-\hat{y})^{(1-0)}$$

$$= 0.2^0(1-0.2)^{(1-0)} = 0.8$$

$X = \{ x_1, \ldots\ldots, x_n \}$

likelihood (X) = product of individual likelihood

$$= \prod_{j=1}^{n} \text{likelihood}(x_j)$$

measure of goodness → higher the likelihood → better it is.

View this as loss function → minimize

likelihood → maximize

$$Lh(X) \xrightarrow{max} \log(Lh(x)) = \sum_{j=1}^{n} \log(\text{likelihood}(x_j))$$

maximize the log (Lh) insted of Lh directly.

Convert to minimize

* loss function :- $nll(X) = -\sum_{j=1}^{n} \log(\text{likelihood}(x_j))$

So if we minimize loss we maximize likelihood.

Logistic Regression and Regularization

labels
$\rightarrow \{0,1\}$

function w = Logistic Regression – Stochastic Fit (X, y)
Initialize randomly a d-dimensional vector w and a scalar b
Shuffle the rows (points) of X → shuffle
for $j=1$ to nepochs
  for $t=1$ to n/k → hyperparameter k (size of mini batch) one update per batch.

$$b = b - \rho \sum_{j=(t-1)k+1}^{tk} \left(\sigma(b + X_j w^T) - y_j \right)$$

sigmoid fn

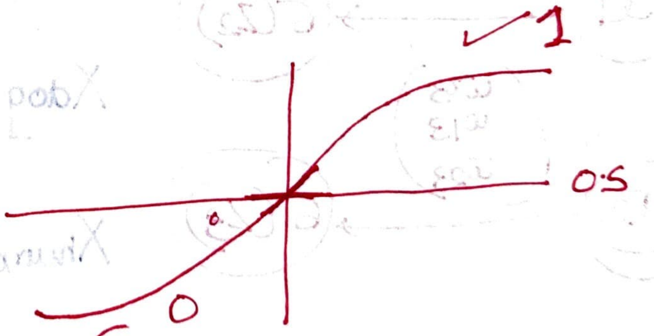
$\hat{y}_j \quad - y_j$

  for $k=1$ to d:

$$w_k = w_k - \rho \sum_{j=(t-1)k+1}^{tk} \left(\sigma(b + X_j w^T) - y_j \right) X_{jk}$$

return w, b

$L(X, y, w)$: w that minimizes loss
gradient descent



when $k=1$: identical to perceptron

difference is $\sigma \rightarrow$ sigmoid is used for calculation

SGD → Logistic neuron

0, 1   0 ...
0.5 ...
1

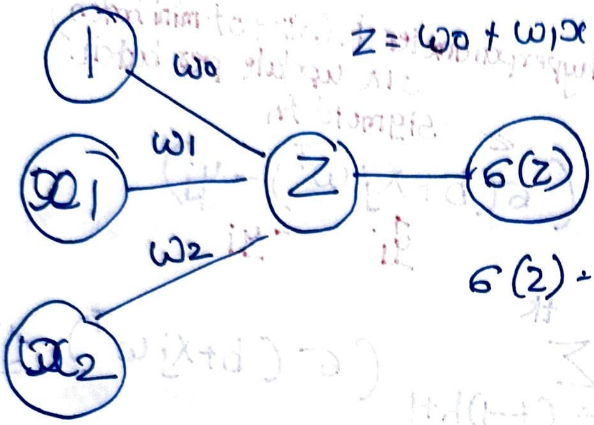
$\underbrace{Xw^T}_{z} + \underbrace{b}_{z}$

$\phi \quad \pounds \quad \underline{\underline{+1}}_{-1}$

*From one vs rest (From binary to multilabel classification) :-

~~$\mathcal{L}(X, y, \omega)$: ω that minimizes the loss function~~

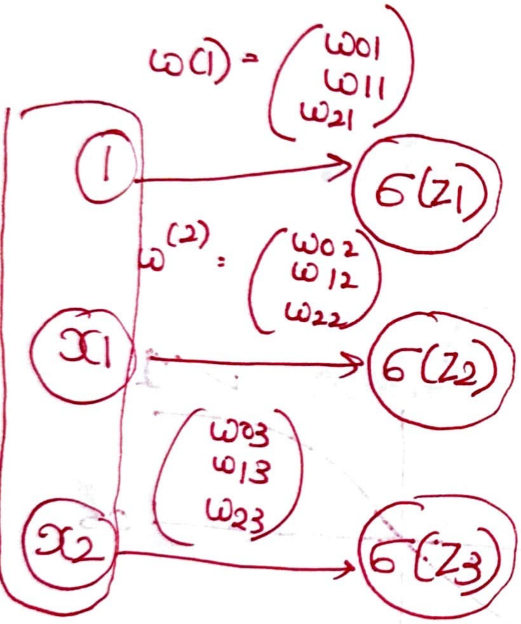
$z = \omega_0 + \omega_1 x_1 + \omega_2 x_2$



logistic neuron

$\sigma(z) = \frac{1}{1+e^{-z}}$ } Prob X is in class 1

CAT / DOG / HUMANS : $X = [X_{CAT}, X_{DOG}, X_{CAT}]$



Train 3 classifiers

X_{cat} : $[X_{cat}, \{X_{dog}, X_{human}\}]$ 0·3 cat
1 0

X_{dog} : $[X_{dog}, \{X_{human}, X_{cat}\}]$ 0·6 dog
1 0

X_{human} $[X_{human}, \{X_{cat}, X_{dog}\}]$ 0·8 human
1 0

$\frac{0.3}{1.7} + \frac{0.6}{1.7} + \frac{0.8}{1.7}$ → output prob distribution

Final Classification = max {0·3, 0·6, 0·8}

0·8 → human
(highest probability)

(0·3 + 0·6 + 0·8 don't sum up to 1) but its fine } $\frac{\sigma(z_i)}{\sum_i \sigma(z_i)}$
= 1·7

if we have k labels : k classifiers (X)

every classifier uses entire dataset but with different labels.

~~[illegible]~~ $\frac{k(k-1)}{2}$ $\{X_{CAT}, X_{dog}\}$