Today's Agenda

1) Perceptron loss Function

2) MLP Intuition

12marks | CGPA | Placement

$$L(w_1, w_2, b) = \boxed{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))} + \boxed{\alpha R(w_1, w_2)}$$

$$L(\quad = \max(0, -y_i f(x_i))$$

$f(x_i) \rightarrow w_1x_1 + w_2x_2 + b$

$$L = \frac{1}{n}\sum_{i=1}^{n} \max(0, -y_i f(x_i))$$

no of rows $\leftarrow n$

Imagine $\rightarrow$ [(non-zero) / 0 + n2/0 + n2/0]

Loss function $\longrightarrow$ w_1, w_2, b if the value changes

line equation

Minimize the loss

Gradient Descent

minimize loss function

argmin $\rightarrow$ least value index

$$\boxed{L = \underset{w_1, w_2, b}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{2} \max(0, -y_i f(x_i))}$$

Geometry of loss Function

X_1	X_2	Y
X_{11}	X_{12}	Y_1
X_{12}	X_{21}	Y_2

X_{ij}: $i \rightarrow$ row, $j \rightarrow$ column

$$L = \frac{1}{n}\sum_{i=1}^{n} \max(0, -yf(x_i))$$

$$\max(0, \underbrace{-y_i f(x_i)}_{x})$$

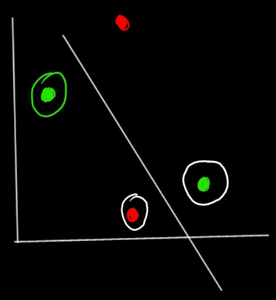
$$\max(0, -y_i \underline{x})$$

$x \geq 0$

$x < 0$

$$L = \frac{1}{2}\left[\max(0, -y_1 f(x_1)) + \max(0, -y_2 f(x_2))\right]$$

Original	Predicted
Y	$\hat{Y}$
✓ 1	1
✓ -1	-1
1	-1
-1	1



$w \cdot x_i$ Positive

Case I +ve +ve = +ve

Predict Region

$= \max(0, -(+ve))$

$= \max(0, -ve)$

$= 0$

✓ Correct = 0

Incorrect = Non-Zero Component

Case II

y_i $f(x_i)$

-1 $-ve$ = +ve

$= \max(0, -(+ve))$

$= \max(0, -) = 0$

✓ underfitting { Training Accuracy

✓ overfitting { Prediction Accuracy

data points | dataset

2000 rows

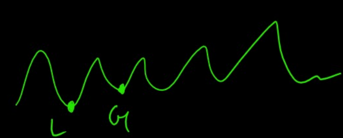
loss term for a single point = 0

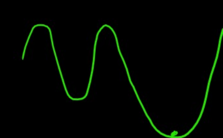
loss for the entire dataset cannot be 0 when there are misclassifications

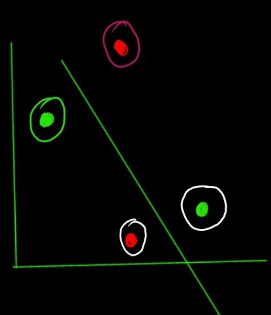
0 loss

overfitted

underfitted







y_i $f(x_i)$

1 $-ve$ = $-ve$

$\max(0, -y_i f(x_i))$

$\max(0, -(-ve))$

$(-ve)$ {NZ}

$\begin{cases} \geq 0 \\ < 0 \end{cases}$

Case IV

y_i $f(x_i)$

-1 $+ve$ = $-ve$

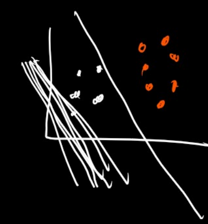
$(-ve)$ {NZ}

Loss Function

Formula Calculation for different data points

$$= [0 + N2 + N2 + 0]$$

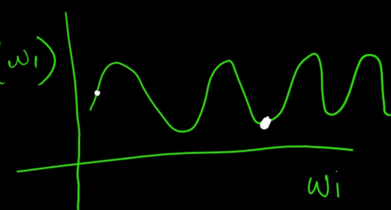
$$= N2$$



$$L = \underset{w_1, w_2, b}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \max(0, -y_i f(x_i))$$

$L(w_1, w_2 b)$

$L(w_1)$



α

for i in iterations:

Weight Update

$w_1 x_{i1} + w_2 x_2 + b$

$$w_1 = w_1 + \eta \frac{\partial L}{\partial w_1}$$

$\eta \rightarrow$ Learning Rate $= 0.01$

$$w_2 = w_2 + \eta \frac{\partial L}{\partial w_2}$$

$$b = b + \eta \frac{\partial L}{\partial b}$$

$$\left[\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial b}\right]$$

$$f(x) = w_1 x_{i1} + w_2 x_{i2} + b$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f(x_i)} \times \frac{\partial f(x_i)}{\partial w_1} = x_{i1}$$

Conditional for $\partial f(x_i)$

$\max(0, -y_i f(x_i))$

$$\frac{\partial L}{\partial f(x_i)} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i & \text{if } y_i f(x_i) < 0 \end{cases}$$

Partial Derivatives

$$\frac{\partial L}{\partial w_1} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i1} & \text{if } y_i f(x_i) < 0 \end{cases}$$

$$\frac{\partial L}{\partial w_2} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i x_{i2} & \text{if } y_i f(x_i) < 0 \end{cases}$$

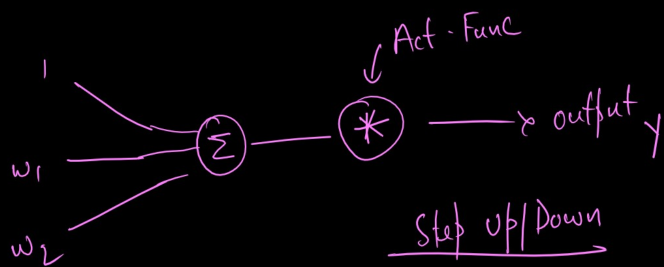
$$\frac{\partial L}{\partial b} = \begin{cases} 0 & \text{if } y_i f(x_i) \geq 0 \\ -y_i & \text{if } y_i f(x_i) < 0 \end{cases}$$

Code
- Pytorch
- Tensorflow

SGD → Loss Functions, Activations, Optimizers → SGD
Step up/down

Perceptron → Flexibility

class
multiclass
Regression



Step Up/Down Functions

Output
↳ Format

$\sigma(z) = \frac{1}{1+e^{-z}}$ Sigmoid → (0, 1) → Probabilities

Binary Classification → Binary Cross Entropy

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

Multiclass Classification → categorical cross Entropy

$$L = \sum_{j=1}^{M} y_i \log(\hat{y}_i)$$

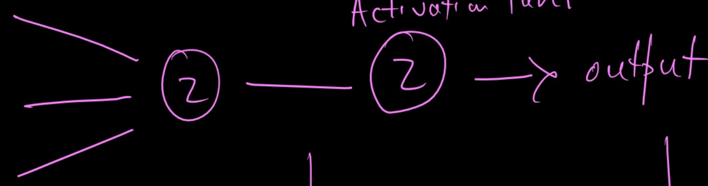
$$\text{Softmax} = \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}} = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

Softmax Regression

0.6 dog
0.3 cat
0.1 house

Loss ~~Activation~~	Act ~~loss~~ Function	Output
Perceptron loss	Step	Perceptron → binary classifier
log loss / binary entropy loss	Sigmoid	logististic → B.C
categorical cross entropy	Softmax	Softmax Regression → Multiclass classifier

Regression → linear



MSE | Linear | linear regression → numbers

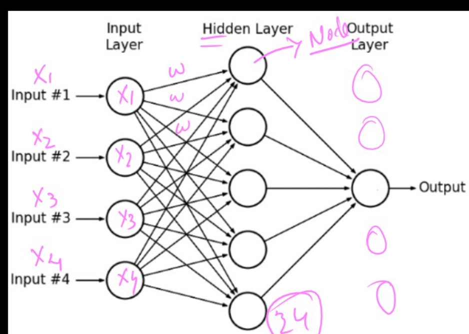
MLP → Multi layer Perceptron

M.C.C

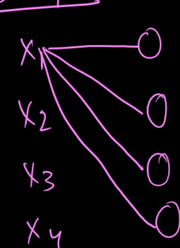
Perceptron → Non-Linear Data

Perceptron | Node

Multiple Perceptrons



I.P



Hidden layer
- → Fully Connected
- → Dense
- → Linear

$4 \times 5 = 20$ connections

$= 20 + 4$ bias

$= 24$

$5 \times 1 = 5$

$5 + 5$

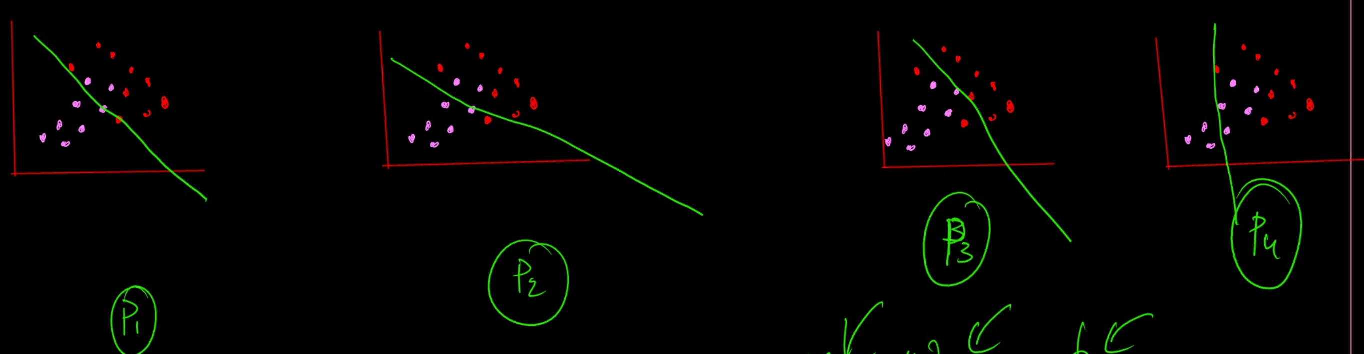
$= 10$

$= 24 + 10 = 34$ (Trainable Parameters)

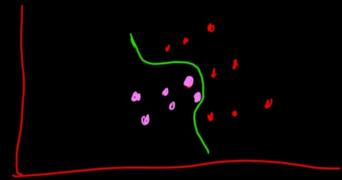
w, w, w, w, b, b, b,

Decision Boundary

Perceptron

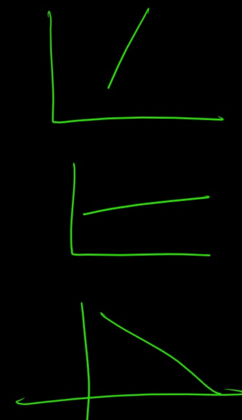


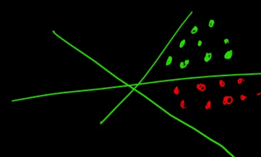
w_1, w_2, b



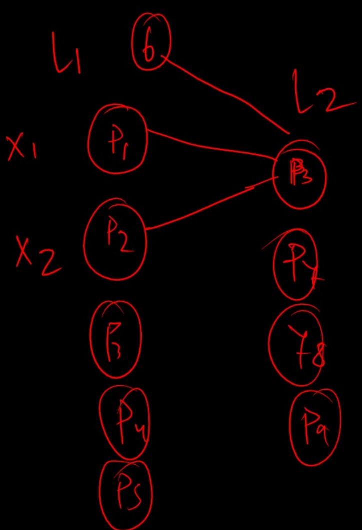
* More Perceptrons *

More Decision Boundary





Input



O.P

Bigger level Perception

Hyperparameters

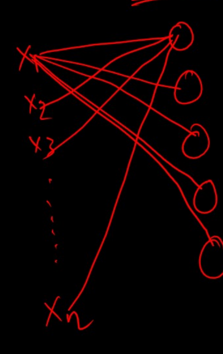
Loss Functions

1) Problem

1) L.R
2) A.F
3) Hidden layers
4) Hidden layer Nodes

Time complexity
Computation complexity

Hidden



Trade off

Selection | computat

150 → GPU
201

MLP

↓

Deep Neural Networks

Layers

Impactful → hyperparameter

Mathematical Calculation of a Network

1) Forward Propagation

F.P

↓

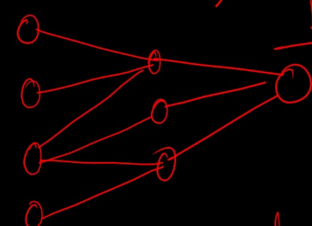
output

↓

BP

L → R

Dot Product



$$\left\{\frac{\partial L}{\partial w_1}\right\} \left\{\frac{\partial L}{\partial w_1}\right\}$$

2) Backward Propagation

R - L

→ Universal Approximators

UAF

1) Differentiation

Update the weights