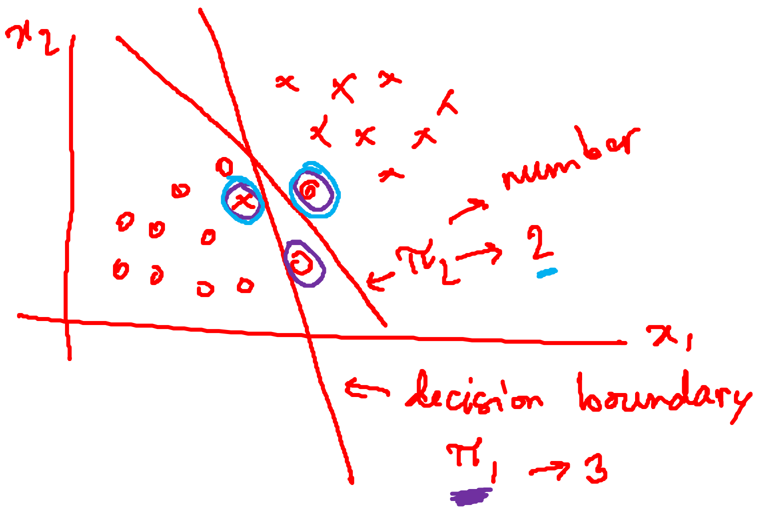
Lecture 2

Perceptron (lecture notes)

CSE465: Pattern Recognition and Neural Network
Sec: 3
Faculty: Silvia Ahmed (SvA)
Spring 2025

Perceptron Loss Function



→ Cannot quantify the success of the decision boundary.

Q: Which one is the better decision boundary?

Sol^n 1: ↓↓ # of misclassified points.

π_2 is the winner.

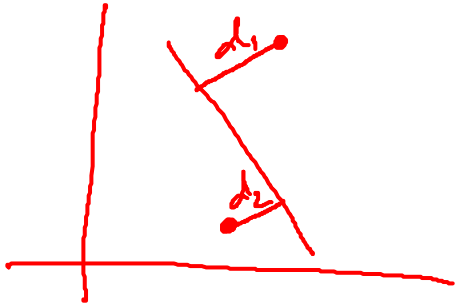
Sol^n 2: $f(w_1, w_2, b) \rightarrow$ number ($\equiv$ error).

Loss function

Linear regression → MSE

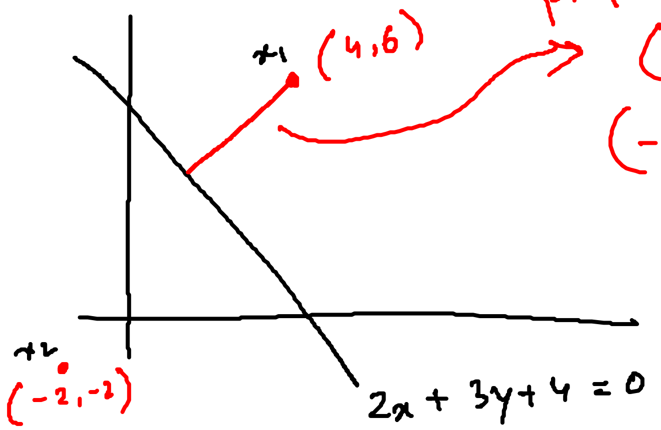
Logistic " → log loss

SVM → Hinge.



$$\sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \dots + (x_n - x_n')^2}$$

proportion to the perpendicular distance.



$(4,6) \Rightarrow 2(4) + 3(6) + 4 = 0 = 30$ $\quad f(x_1)$

$(-2,-2) \Rightarrow 2(-2) + 3(-2) + 4 = -6 \rightsquigarrow |-6| = 6$ $\quad f(x_2)$

Matrix Dot product

$\boxed{30+6} \rightarrow$ Error.

Regularization Term

SGD

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

$$L(y_i, \underset{\hat{y}_i}{f(x_i)}) = \max(0, -y_i f(x_i)), \quad n = \text{\# of rows in data.}$$

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

Breakdown

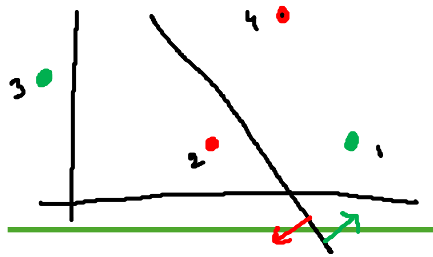
	x_1	x_2	y
1	x_{11}	x_{12}	y_1
2	x_{21}	x_{22}	y_2
$\vdots$	$\vdots$		
i	x_{i1}	x_{i2}	y_i
$\vdots$			
n	x_{n1}	x_{n2}	y_n

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

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

$$\max(0, -y_i f(x_i)) \rightarrow \begin{cases} -y_i f(x_i) \geq 0 \Rightarrow -y_i f(x_i) \\ -y_i f(x_i) < 0 \Rightarrow 0 \end{cases}$$

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



	x_1	x_2	y
1.	7	8	1
2.	6	8	-1
3.	4	2	1
4.	1	1	-1

4 possible cases:

y_i	$\hat{y}_i$
1	1
-1	-1
1	-1
-1	1

For point 1:

$y_1 = 1$: +ve

$f(X_1) = +ve$

$-y_1 f(X_1)$: $-ve$.

$\max(0, -y_1 f(x_1)) = 0$

For point 2:

$y_2 = -1$

$f(X_2) = -ve$

$-y_2 f(X_2)$: $-ve$

$\max(0, -y_2 f(x_2)) = 0$

For point 3:

$\left.\begin{matrix} y_3 = 1 \\ f(X_3) = -ve \end{matrix}\right\}$ -ve

$-y_3 f(X_3) = +ve$

$\max(0, -y_3 f(X_3)) = -y_3 f(X_3)$

For point 4:

$\left.\begin{matrix} y_4 = -1 \\ f(X_4) = +ve \end{matrix}\right\}$ -ve

$-y_4 f(X_4) = +ve$.

$\max(0, -y_4 f(x_4)) = -y_4 f(x_4)$.

$L = \frac{1}{n} \sum \text{errors}.$ $\rightarrow$ Total error of one decision boundary.

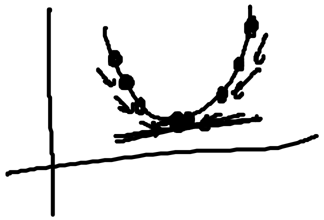
$\underbrace{w_1, w_2, b}$ $\rightarrow$ optimize $\rightarrow$ Gradient Descent

for i in epochs:

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

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

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



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

$$f(x_i) = x_{i1} w_1 + x_{i2} w_2 + b_i$$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f(x_i)} \cdot \frac{\partial f(x_i)}{\partial w_1}$$

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

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

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

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

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

- Flexible by design:

Perceptron → mathematical function

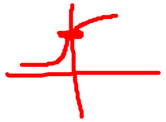
Activation Function

① Step function (⎍) : 1, -1

We want probabilities.

↓

② Activation : Sigmoid function



(used in logistic regression)

$$a(z) = \frac{1}{1+e^{-z}}$$ → Loss function

↓

Log loss/ binary cross entropy

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

③ Multiclass classification:

Activation : softmax function → Loss function : Categorical cross entropy

$$f = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

$$L = \sum_{j=1}^{M} y_j \log (\hat{y}_j).$$

④ Regression:

No Activation function

z → z loss function → MSE

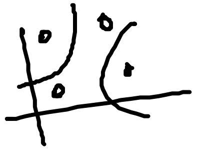
Linear Regression.

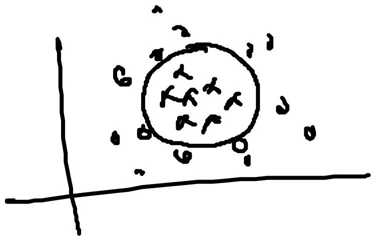
Problems with Perceptron

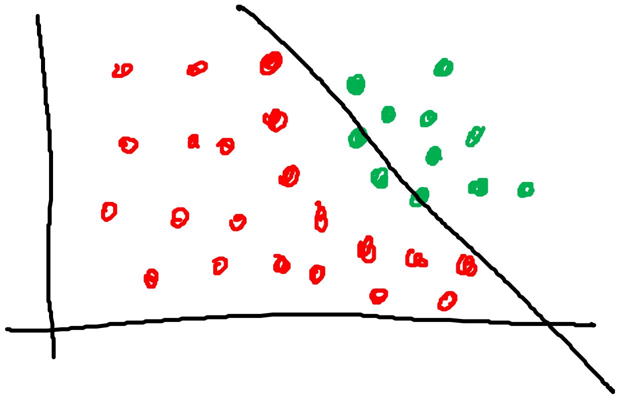
Works only with linearly separable data

AND, OR

XOR



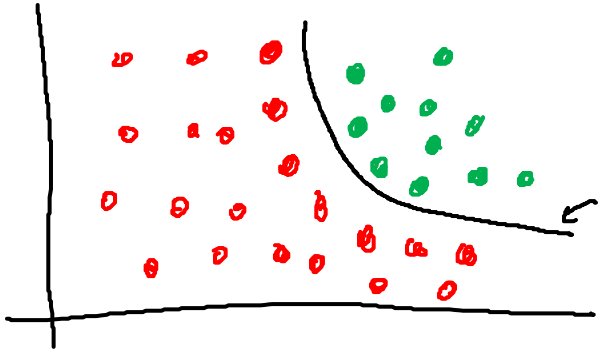




2D → line
↓
3D → Plane
4D → Hyperplane

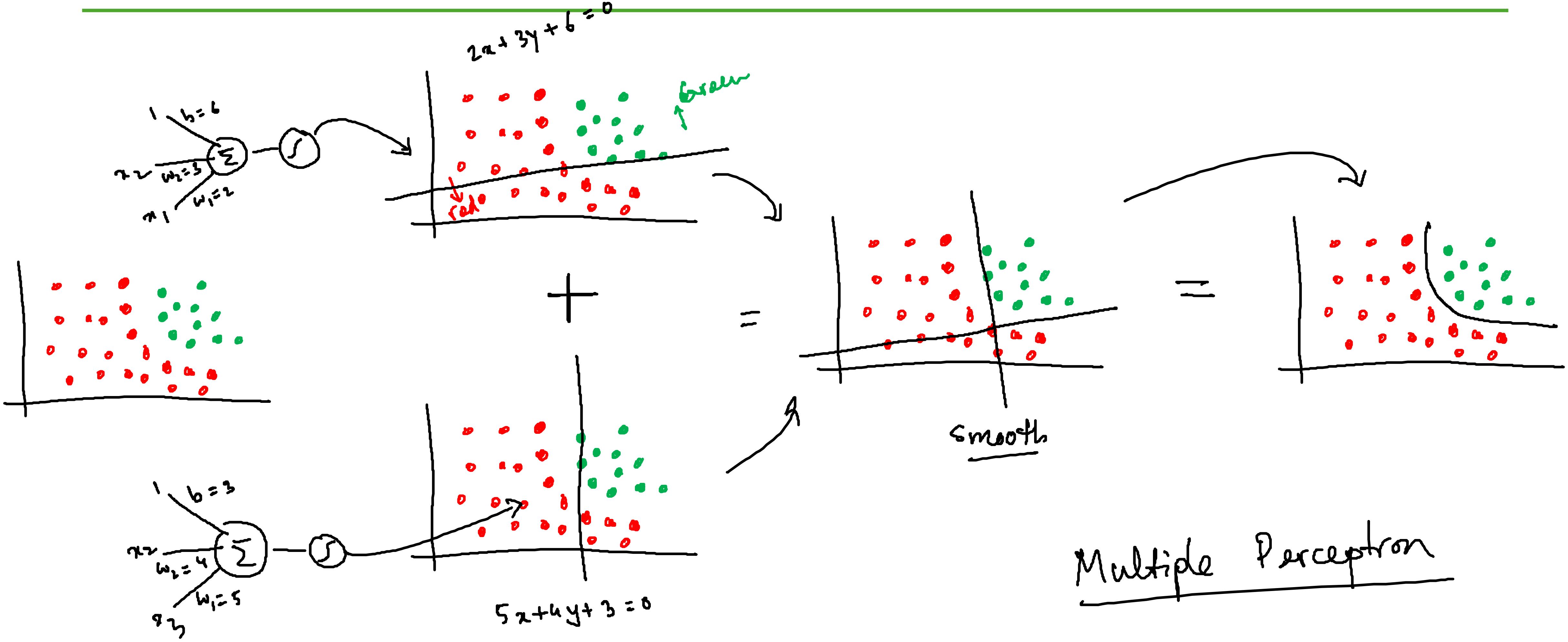
Perceptron

does n't work



decision boundary

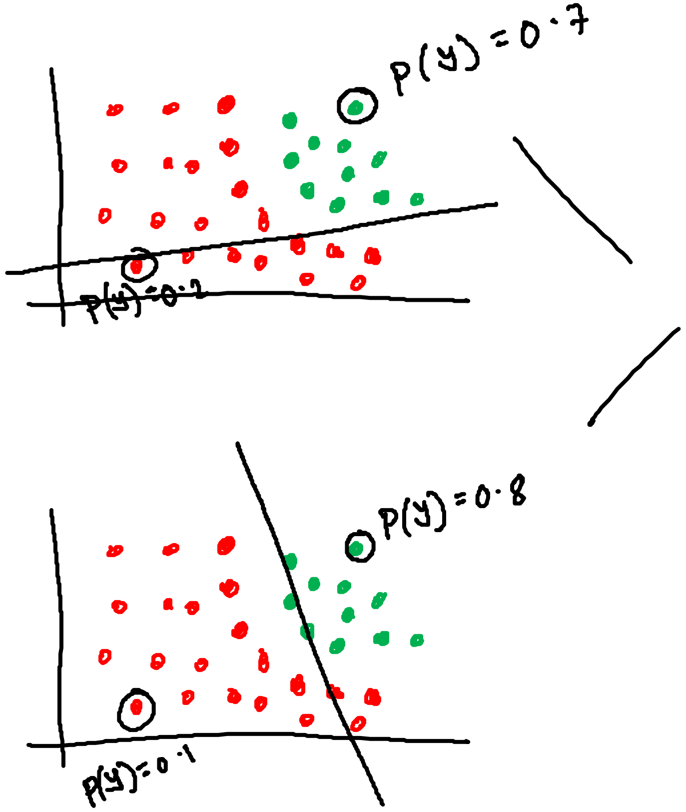
→ need curve.



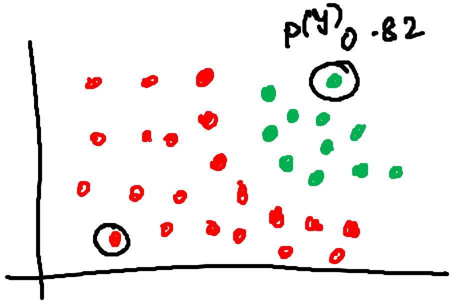
2x + 3y + 6 = 0
b = 6
w2 = 3
w1 = 2
x2
x1
Green
red
b = 3
w2 = 4
w1 = 5
5x + 4y + 3 = 0
smooth
Multiple Perceptron

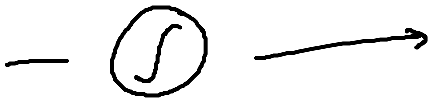
Mathematical Reasoning

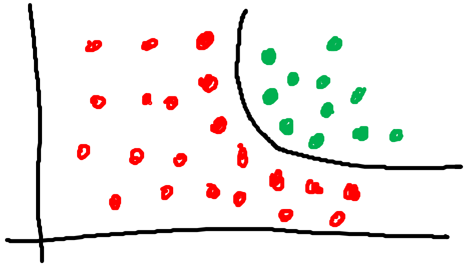
$\int$: Sigmoid



$0.1 + 0.2 = 0.3$





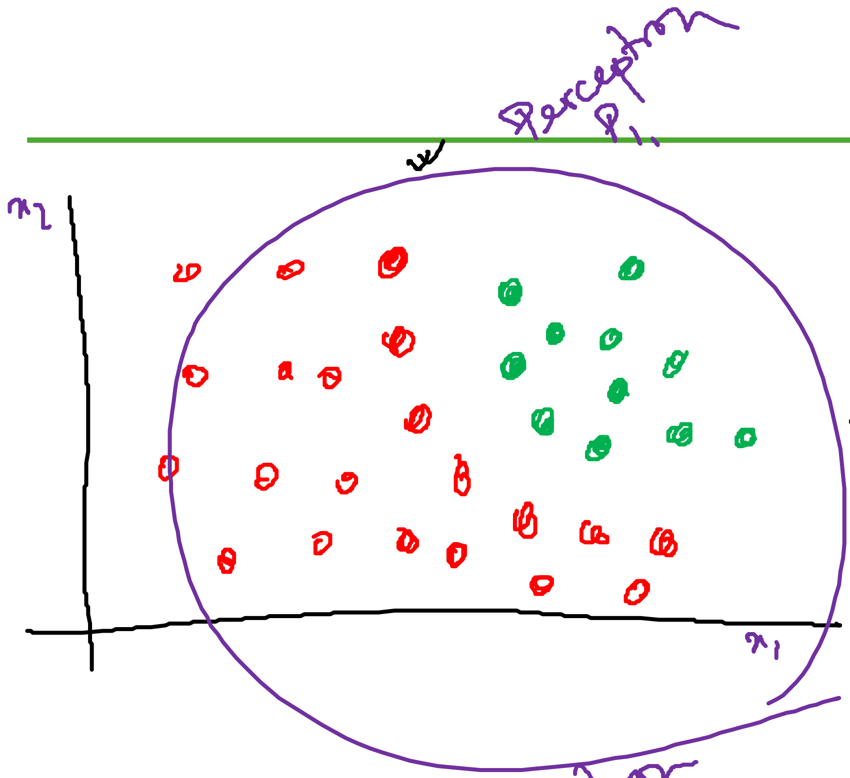


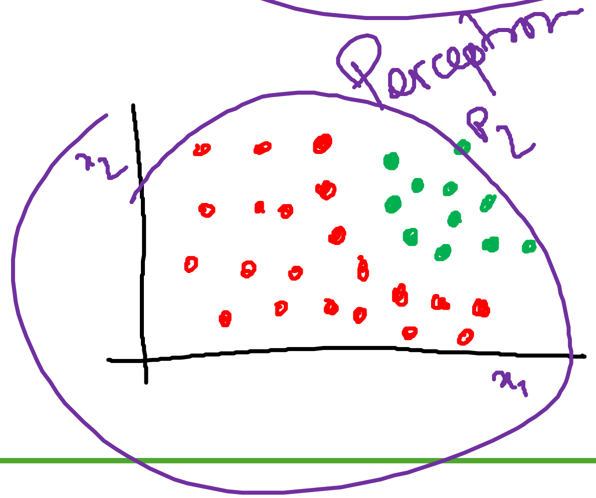
$\rightarrow 0.7 + 0.8 = 1.5$

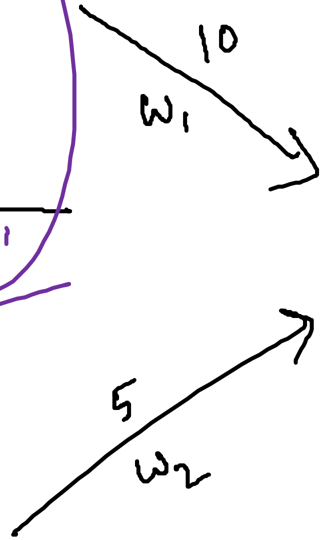
$\frac{1}{1+e^{-1.5}} \rightarrow 0.82$

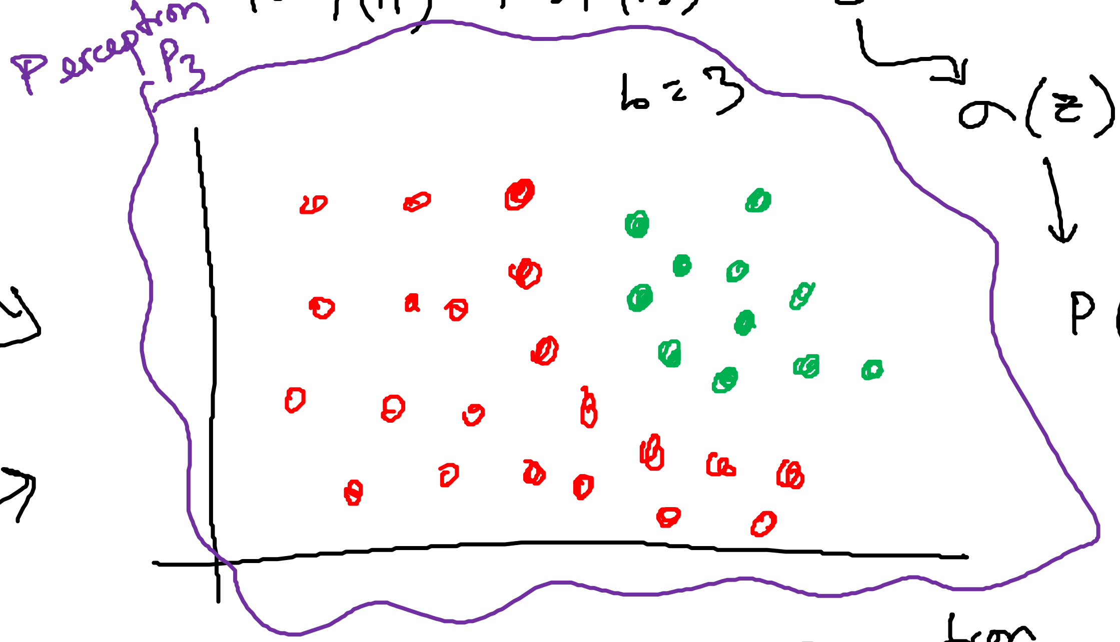
$$10\,P(p_1) + 5\,P(p_2) + b = Z$$

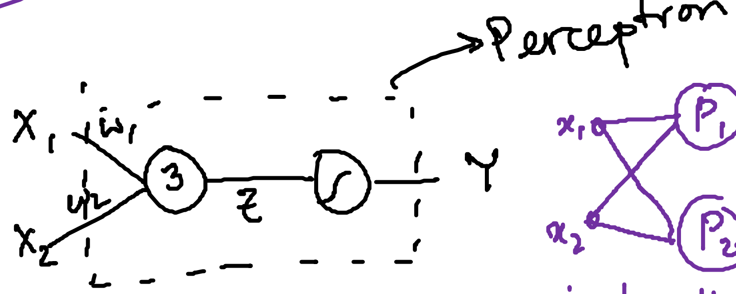
$$Z \rightarrow \sigma(Z) \rightarrow P(Y).$$

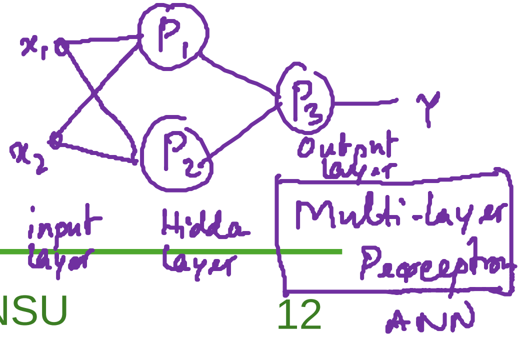


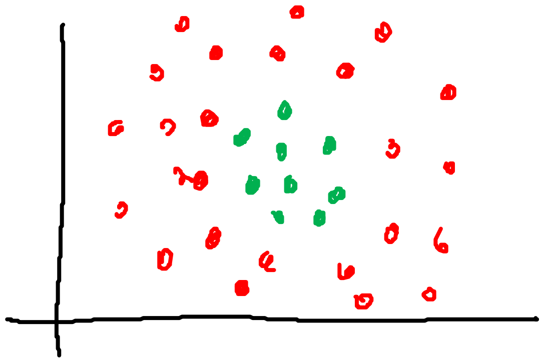








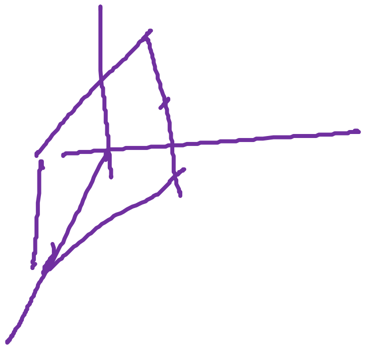




Adding nodes in hidden layer

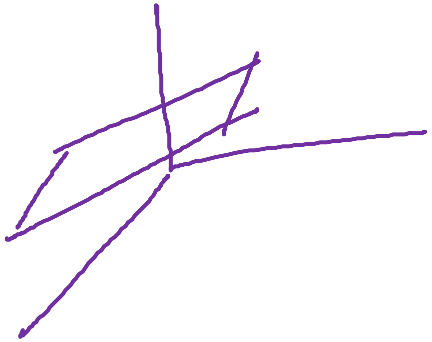
Adding input features

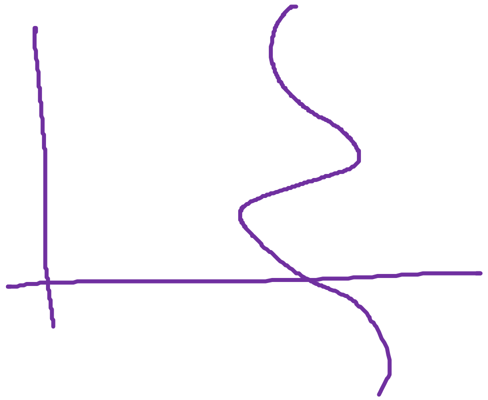
x_1 :



x_2 :

x_3 :





MLP → Universal Function Approximator.