

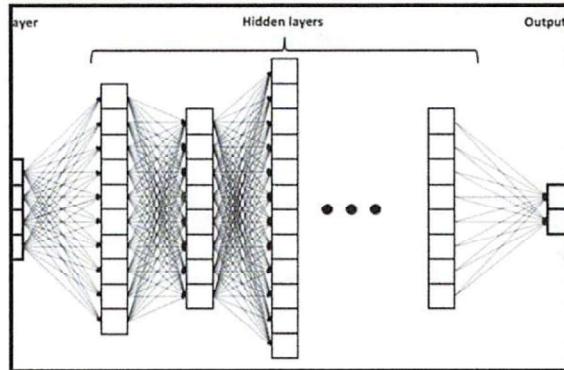
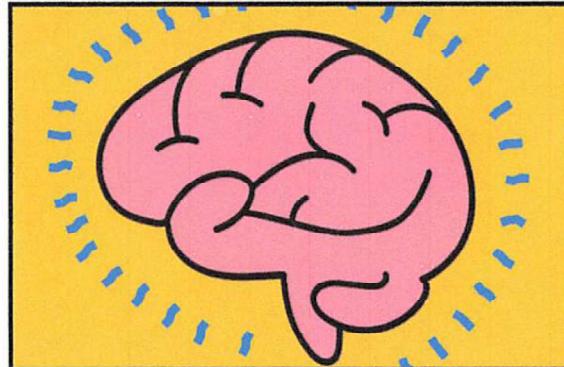
# Introduction to Artificial Neural Networks

This file is meant for personal use by bpidugu@hotmail.com only.  
Sharing or publishing the contents in part or full is liable for legal action.

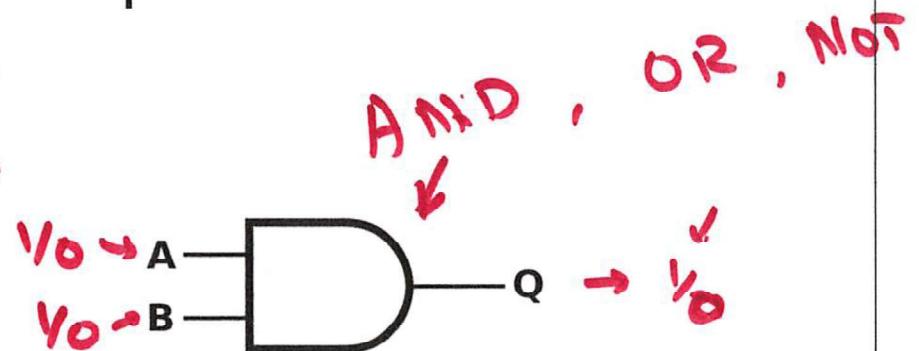
# Agenda

- Introduction
- Perceptron
- Multi Layer Perceptron 
- An Example - MNIST 
- Training
  - Gradient Descent
  - Local vs Global
  - Back Propagation
- Other
  - Activation functions   $\sigma =$
  - Loss functions   $L =$

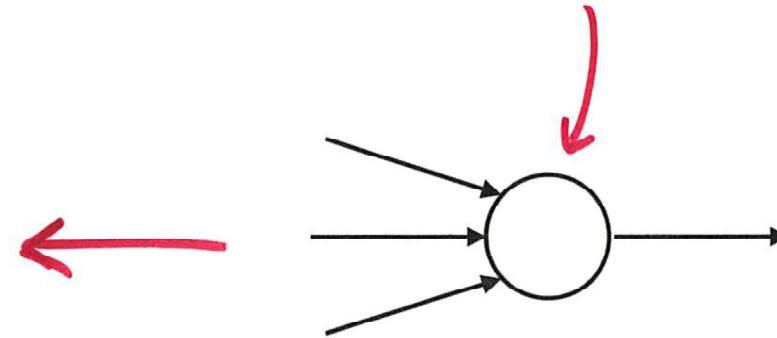
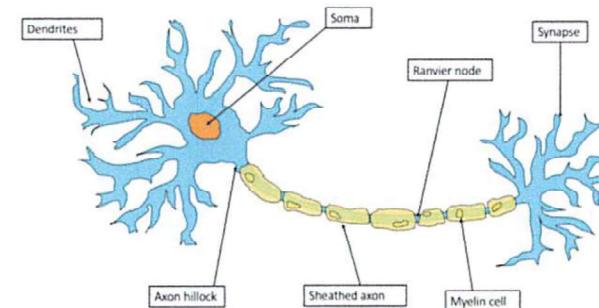
# Building towards a complex task!



$11 \rightarrow 1$   
 $\{0\} \rightarrow 0$   
 $01 \rightarrow 0$   
 $00 \rightarrow 0$



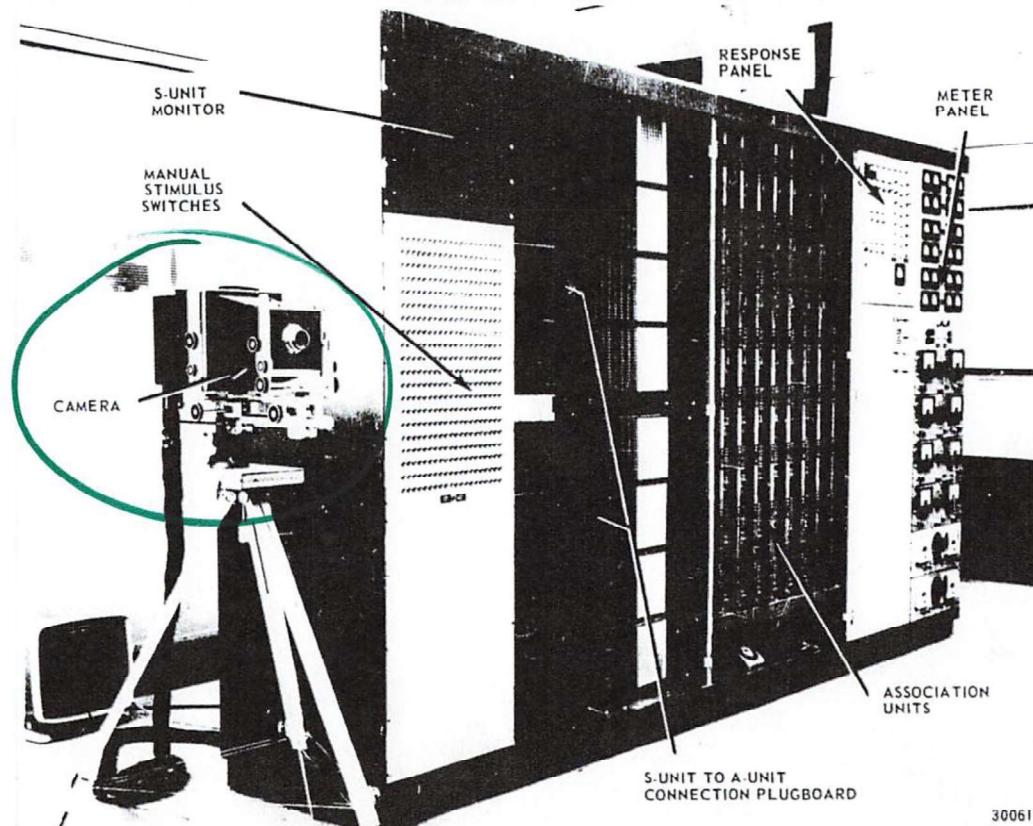
~~86 B~~



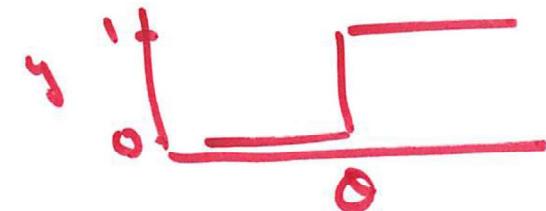
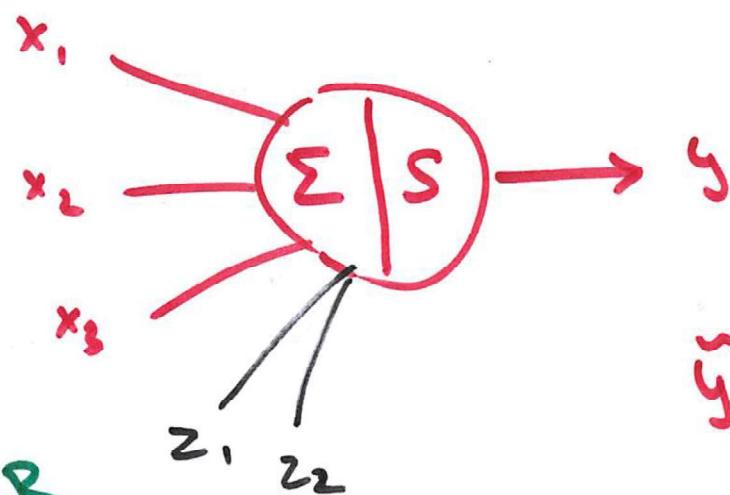
# History

- McCulloch-Pitts Neuron, 1940
- "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." - Rosenblatt, 1958

MARK I  
Perception machine



MP



~~AND, OR~~

$0=3 \rightarrow \text{'and'}$

$0=1 \rightarrow \text{'or'}$

$$\tilde{y} = \begin{cases} 1 & \text{if } \sum x_i \geq 0 \\ 0 & \text{if } \sum x_i < 0 \end{cases}$$

NOT

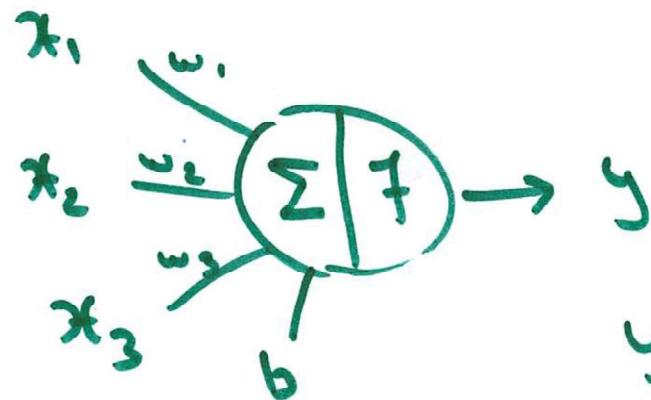
$$y = \begin{cases} \tilde{y} & \text{only if all } z_i = 0 \\ 0 & \text{if any } z_i = 1 \end{cases}$$

# Perceptron!

5

- What does it do?

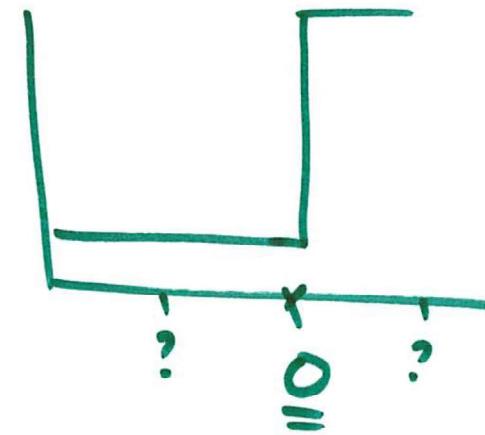
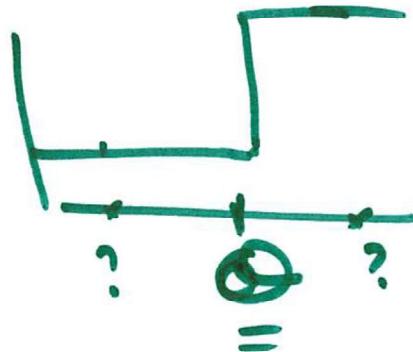
1958

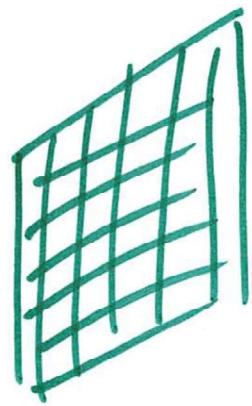


$$y = \text{f}(\sum x_i w_i)$$

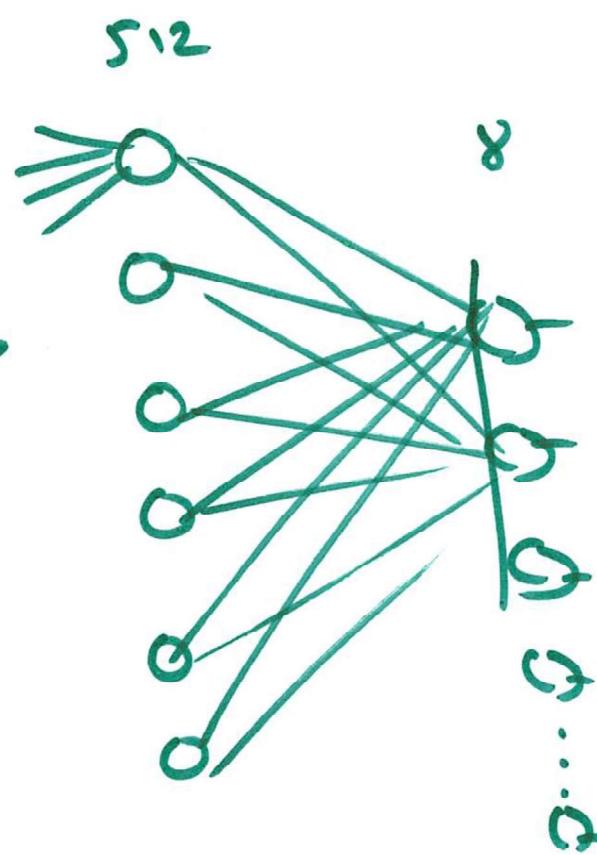
$$y = \text{f}(\sum x_i w_i + b)$$

- Bias

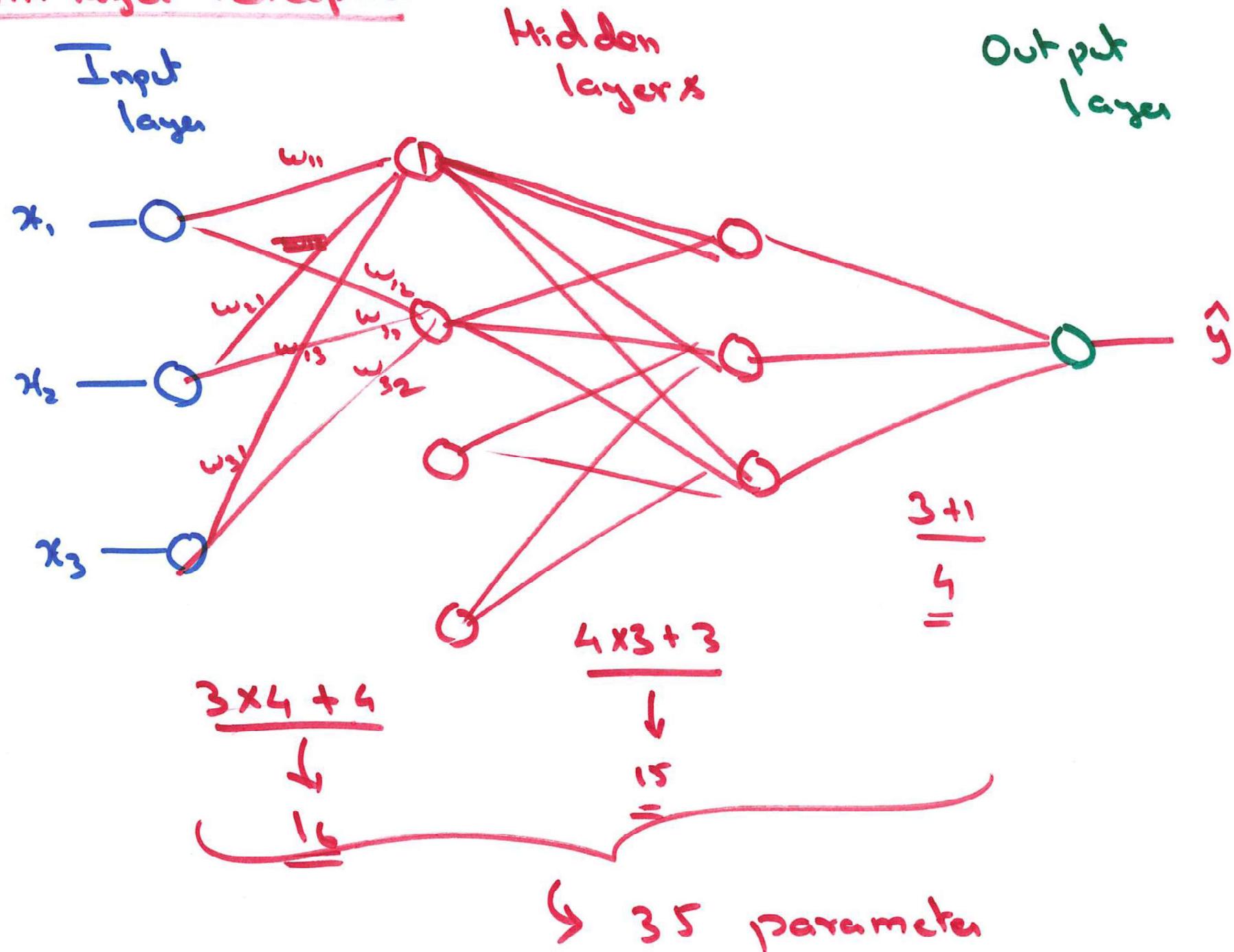


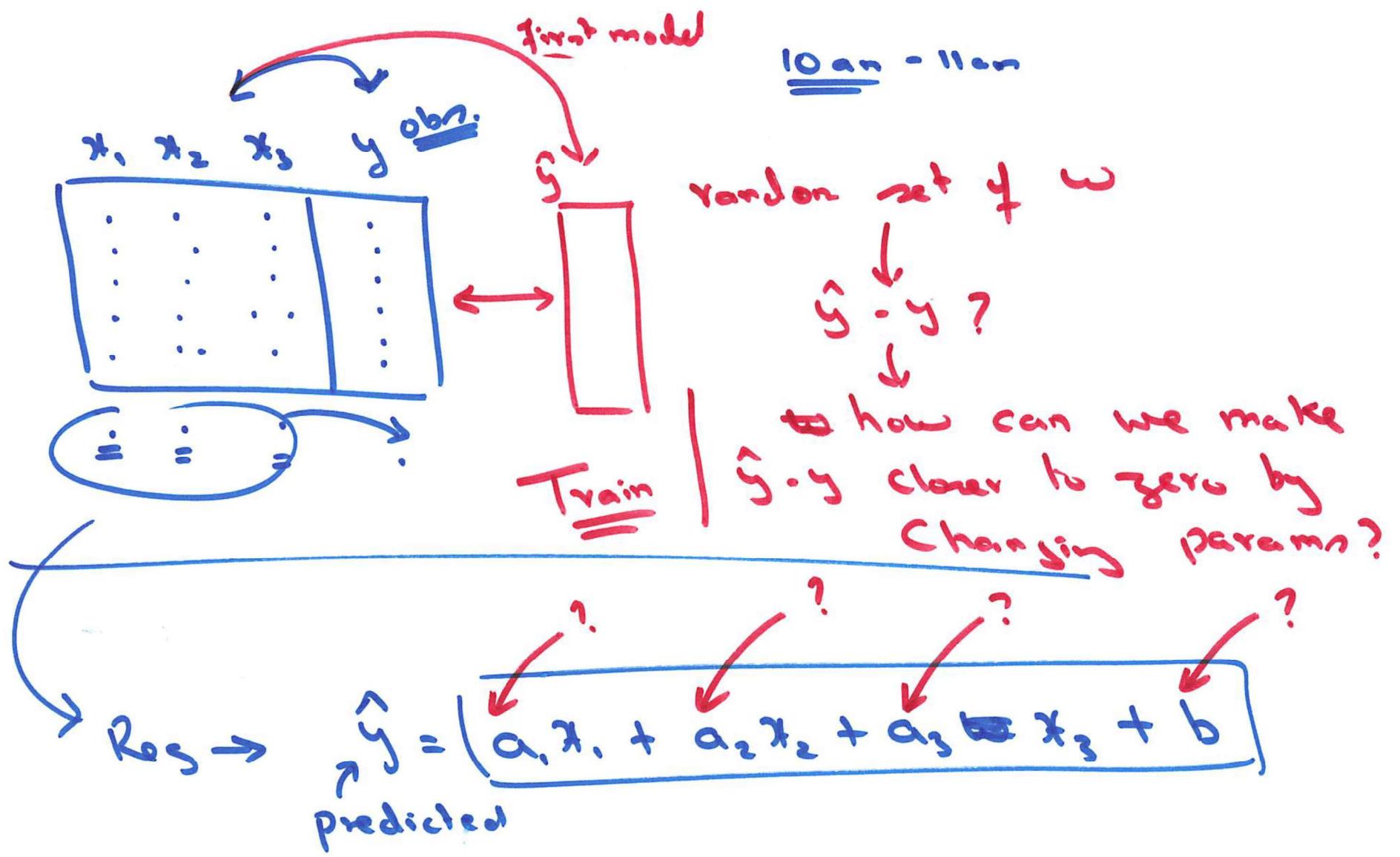


$20 \times 20$  →



## Multi layer Perceptron





## An Example

5 041921314

M N I S T

## database

70000

6

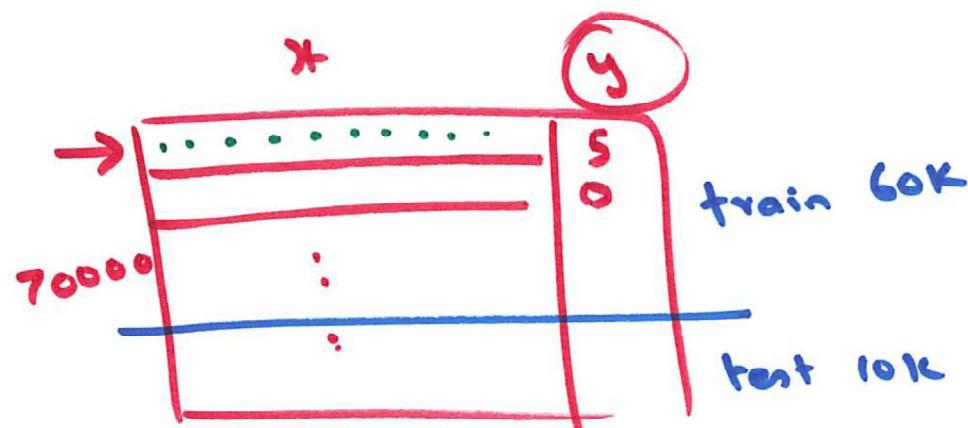
$$28 \times 28 = 784$$

卷之三

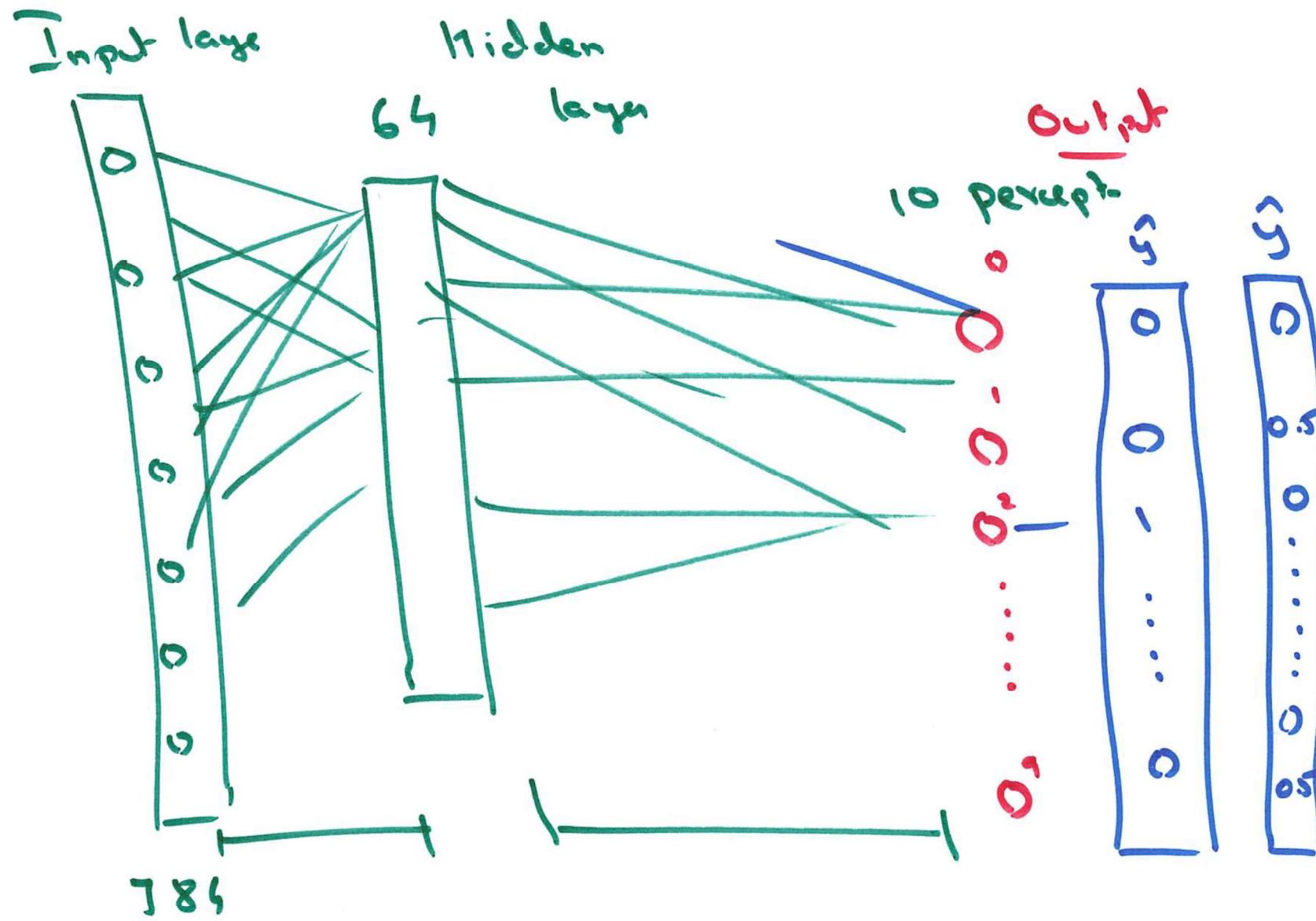
## Classification

卷之三

四

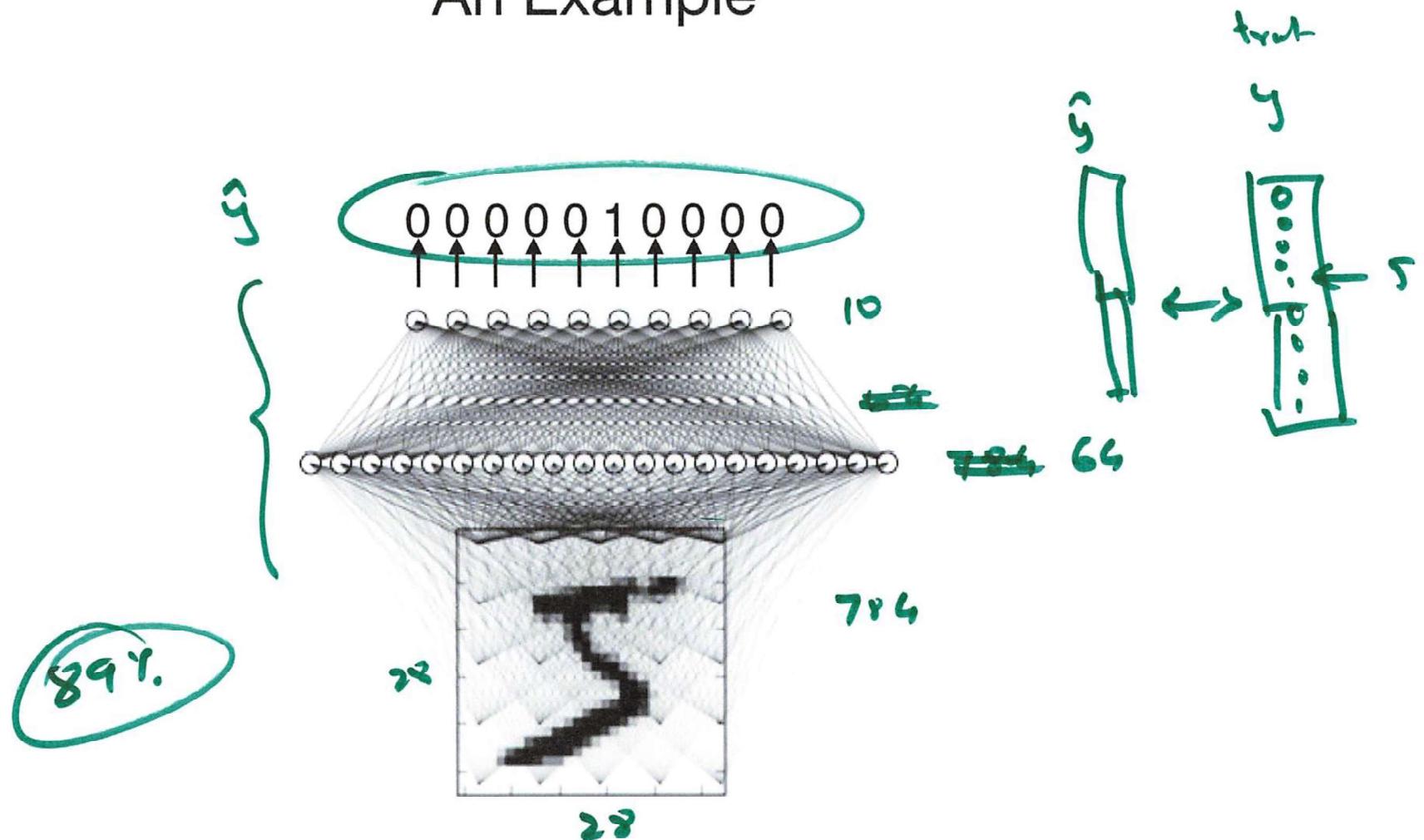


1



$$(784 \times 64 + 64) + ((64 \times 10) + 10) = \underline{\underline{50890}}$$

# An Example



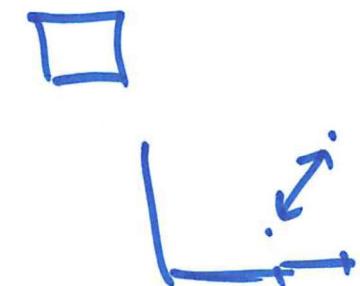
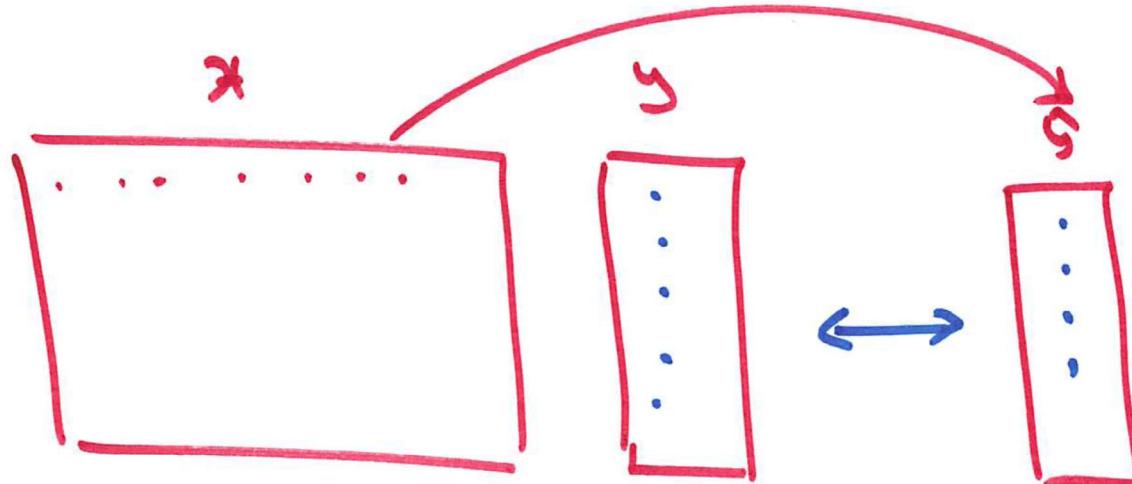
## Activation Function

$$\sigma()$$

$$f()$$

$$(z_i = \sigma(\sum w_i x_i + b))$$

$$f(f(f(\dots)))$$



$$\hat{y} = w_1x_1 + w_2x_2 + w_3x_3 + b$$

how far are  $\hat{y}$  &  $y$ ?

$$\frac{1}{N} \sum_i (y_i - \hat{y}_i)^2$$

Loss function

$$\frac{1}{N} \sum_i |\hat{y}_i - y_i|$$

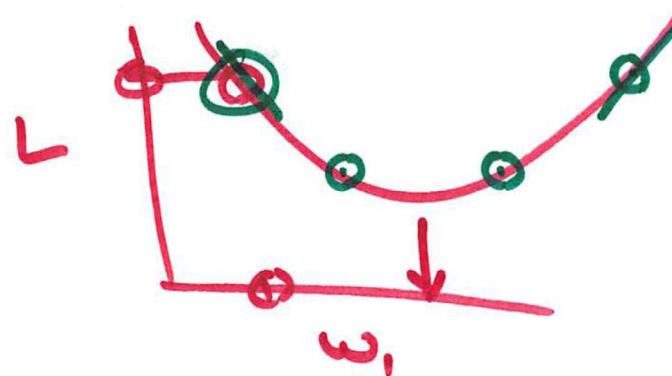


## Training

$$\min \sum (\hat{y}_i - y_i)^2 \text{ by changing } w_1, w_2, w_3, b$$



$$\min \sum_i ((w_1 x_{1i} + w_2 x_{2i} + w_3 x_{3i} + b) - y_i)^2 = L(w, b) \text{ by changing } w_i, b$$

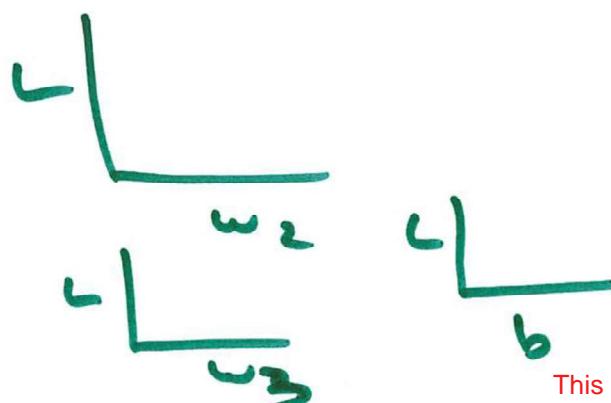


if Slope ~~Pos~~ Neg then more  $w_i \rightarrow$   
 if Slope ~~Neg~~ Pos then more  $w_i \leftarrow$

$$w_i^{\text{new}} = w_i^{\text{old}} - \eta \text{ (slope day } w_i)$$

↑ learning rate

Gradient Descent



$$\omega^{\text{new}} = \omega^{\text{old}} - \eta \frac{\nabla \omega}{\| \nabla \omega \|}$$

vector of slopes

