Introduction to Artificial Neural Networks

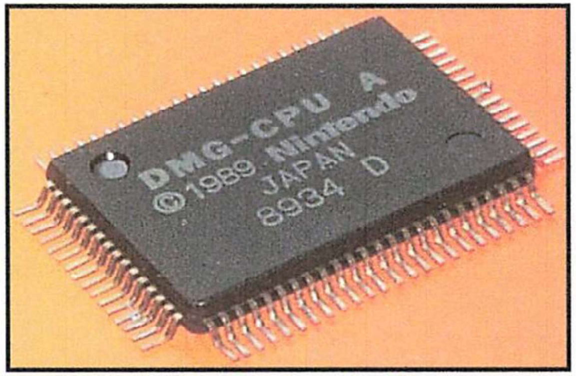
Agenda

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

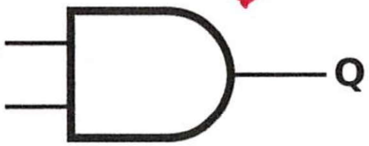
Building towards a complex task!

1 1 → 1
1 0
0 1
0 0 } → 0

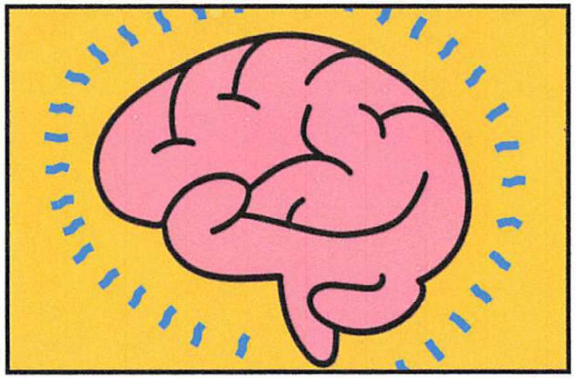
AND , OR , NOT



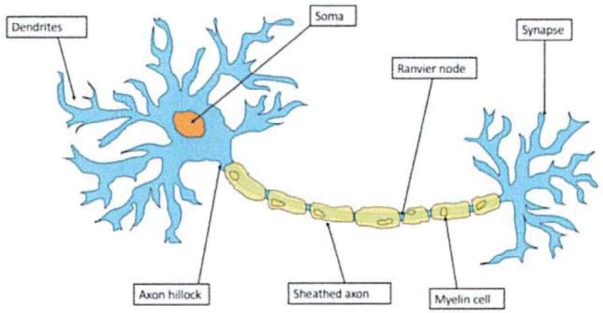
1/0 → A
1/0 → B

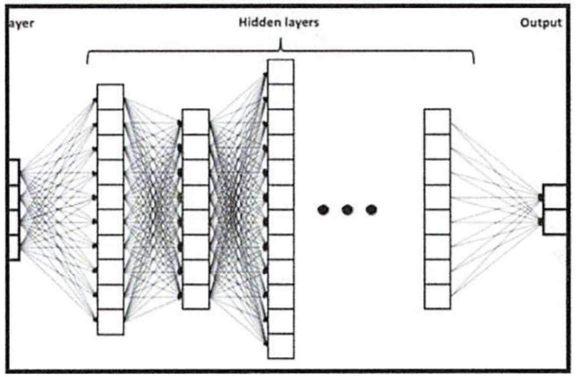


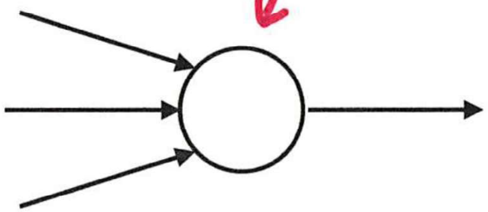
→ 1/0



86 B



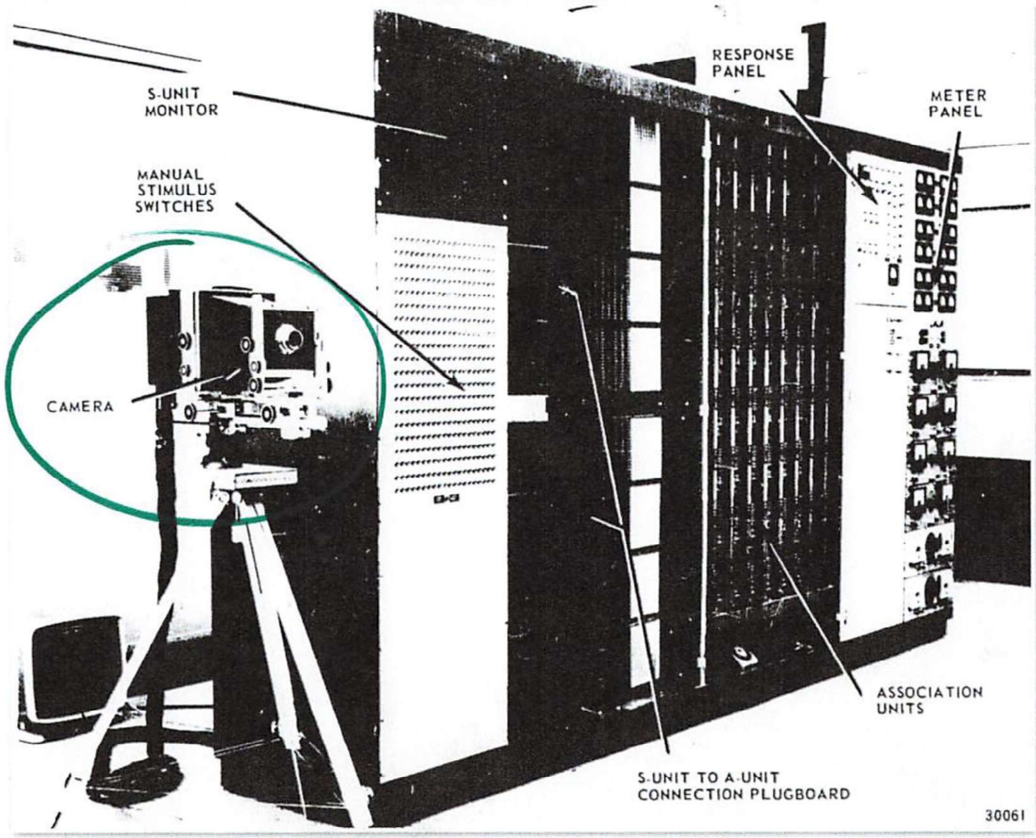




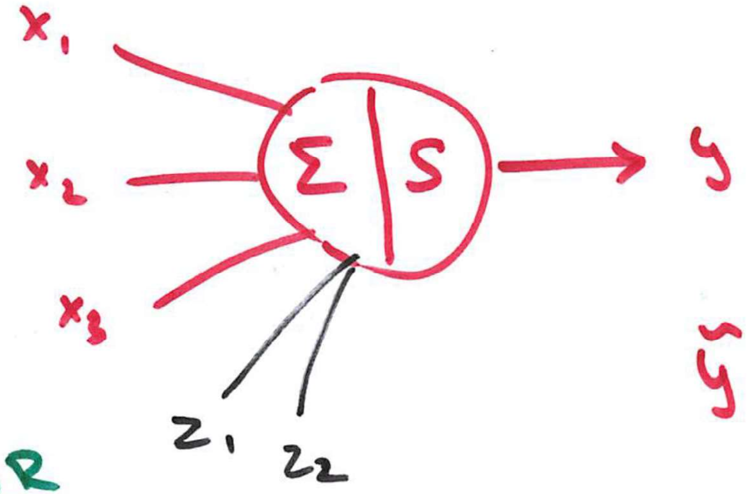
History

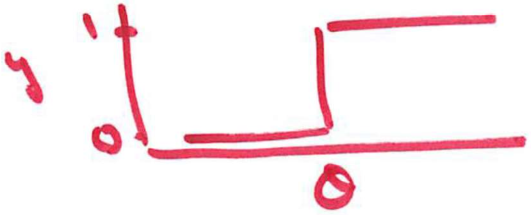
- McCulloch-Pitts Neuron, 1940
- "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." - Rosenblatt, 1958

MARK I
~~Perce~~ perceptron
machine



MP





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

AND, OR

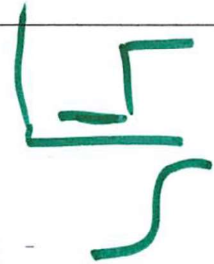
$\theta = 3 \rightarrow$ 'and'

$\theta = 1 \rightarrow$ 'or'

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

NOT

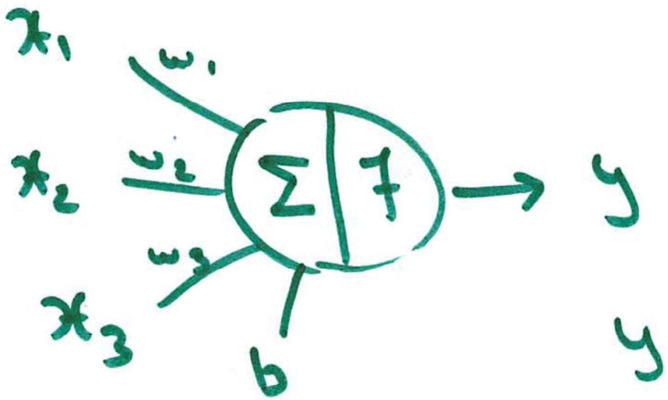
Perceptron!



$$y = f\left(\underline{\sum x_i w_i}\right)$$

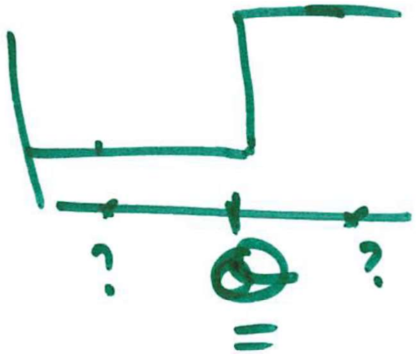
- What does it do?

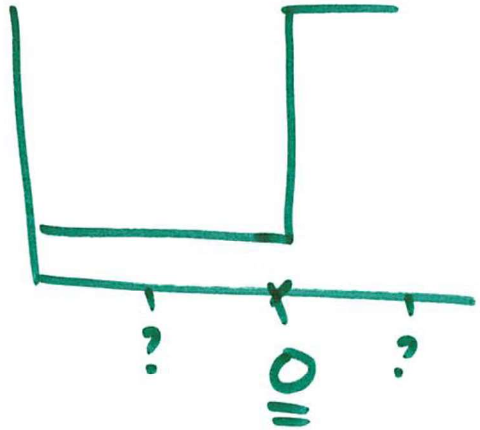
1958

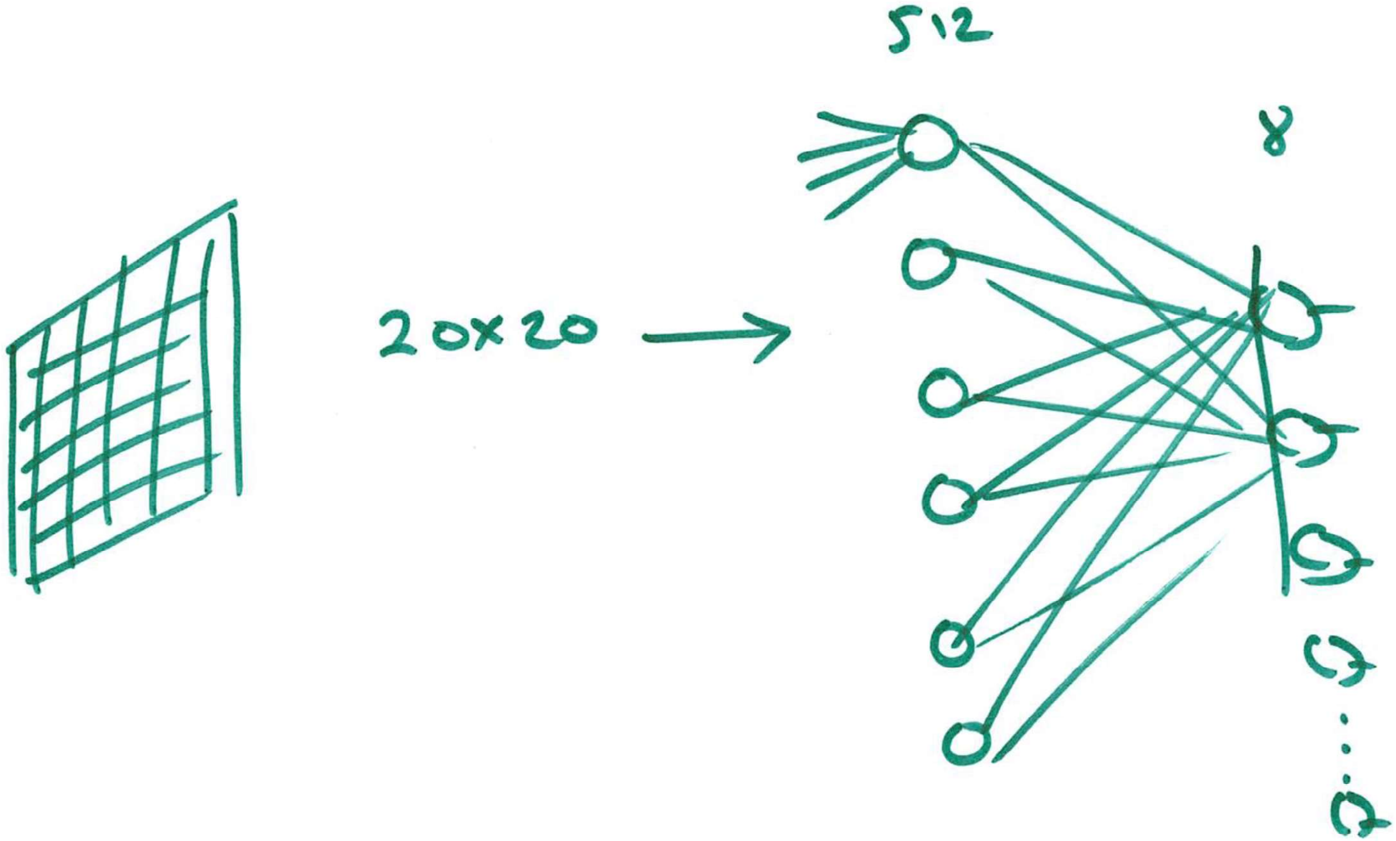


$$y = f\left(\sum x_i w_i + \underline{b}\right)$$

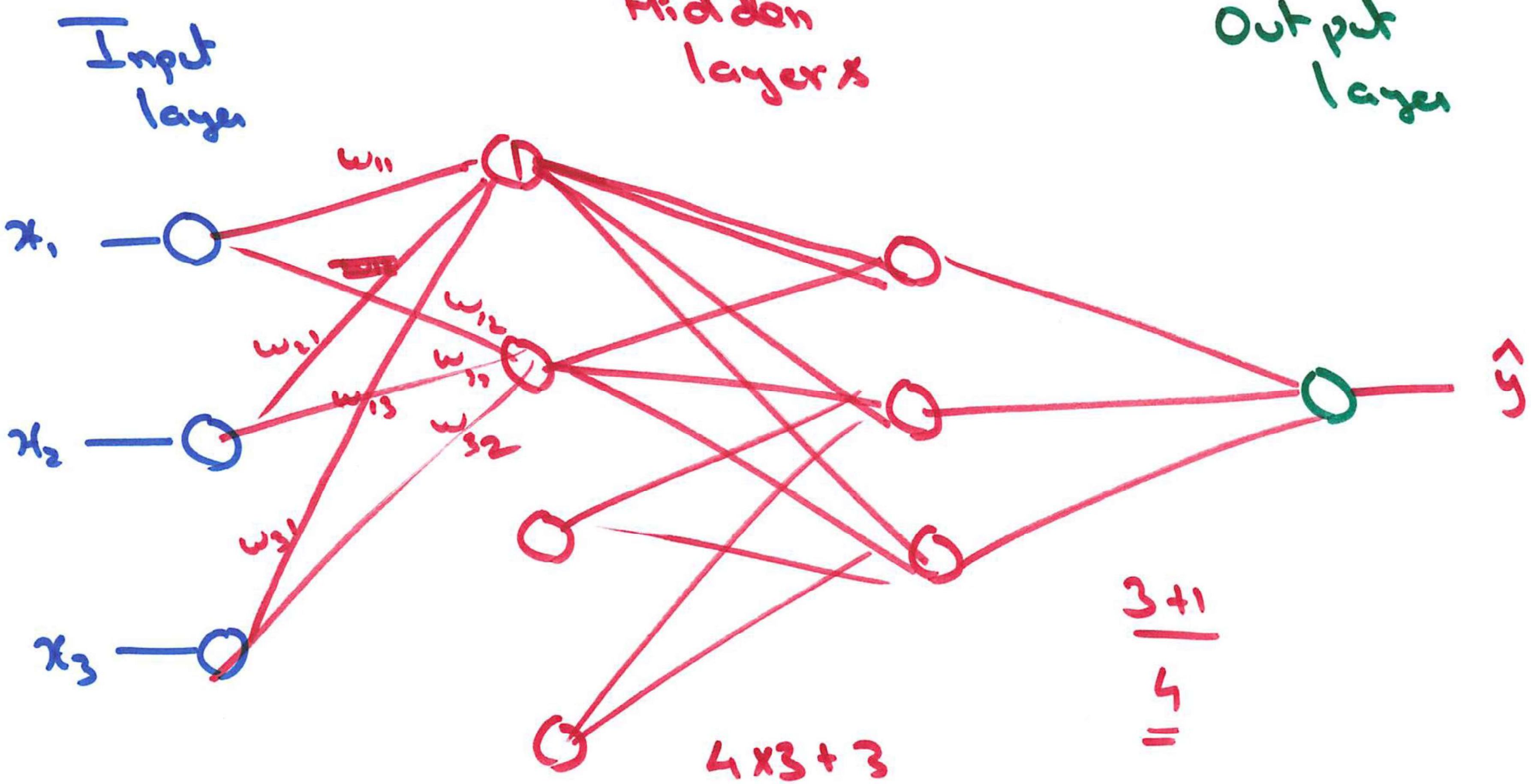
- Bias






512
8
20x20

Multi layer Perceptron



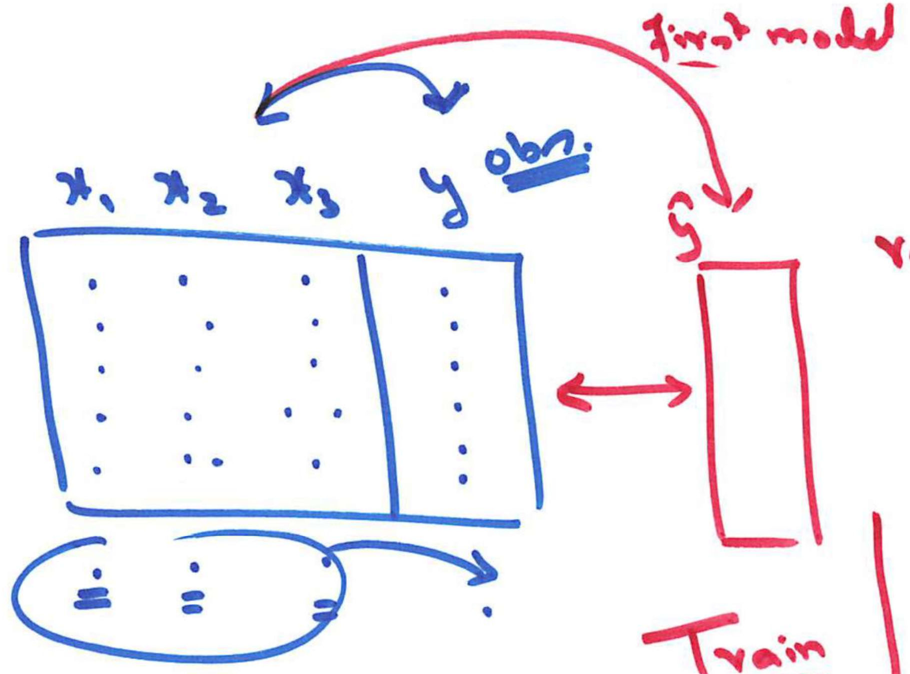
$\underline{3+1} \rightarrow \underline{\underline{4}}$

$\underline{3 \times 4 + 4} \rightarrow \underline{\underline{16}}$

$\underline{4 \times 3 + 3} \rightarrow \underline{\underline{15}}$

$\underbrace{16 + 15 + 4} \rightarrow$ 35 parameter

10 am - 11 am



Train

random set of w

↓

$\hat{y} - y$?

↓

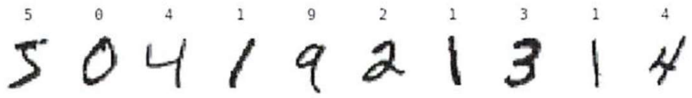
~~to~~ how can we make $\hat{y} - y$ closer to zero by changing params?

? ? ? ?

Reg → $\hat{y} = \boxed{a_1 x_1 + a_2 x_2 + a_3 x_3 + b}$

predicted

An Example



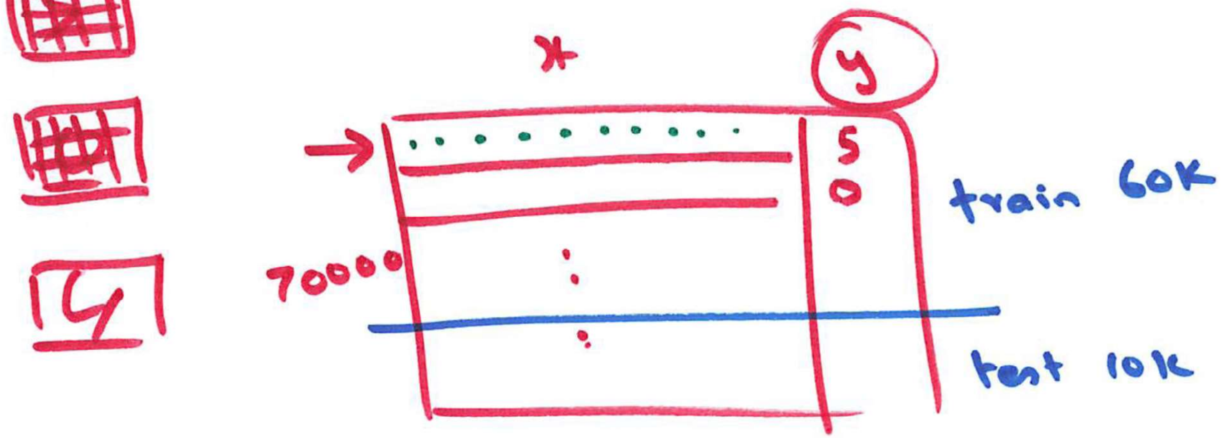
MNIST database

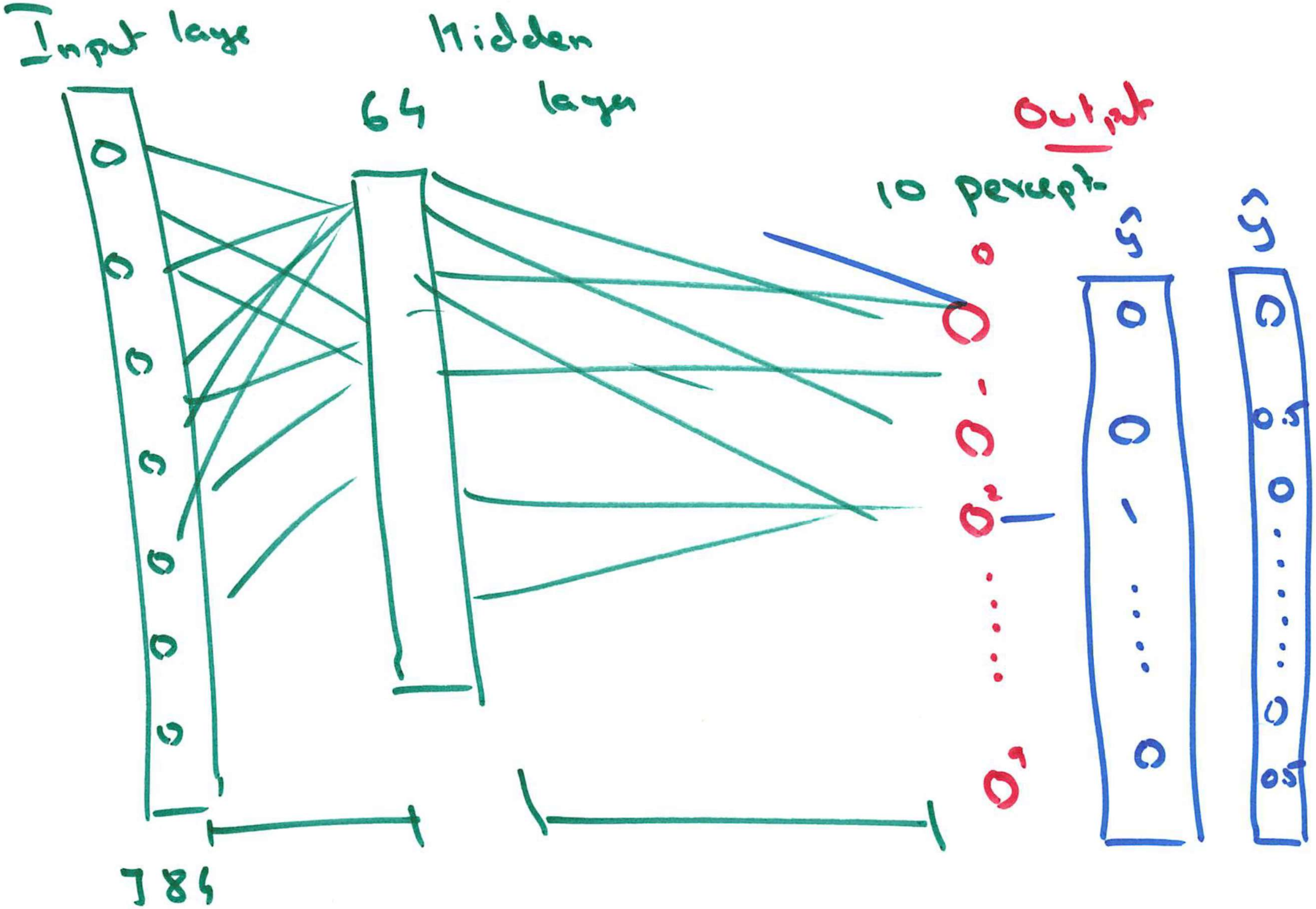
Classification

70000

*

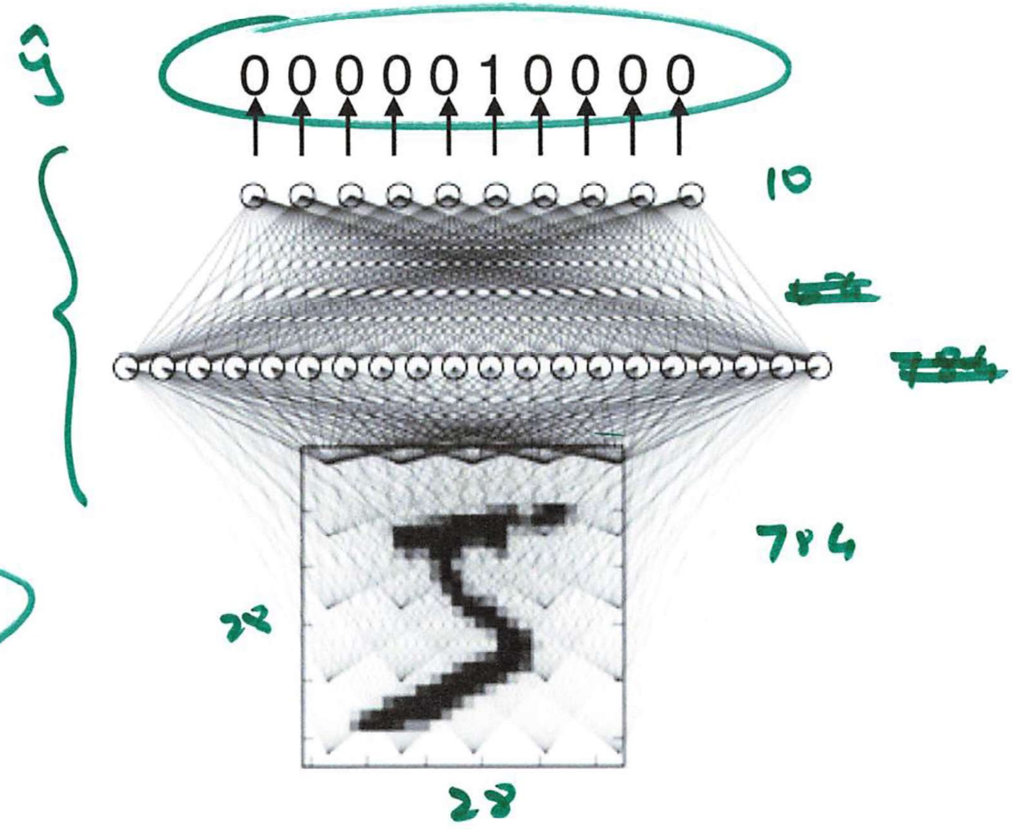
28x28 = 784



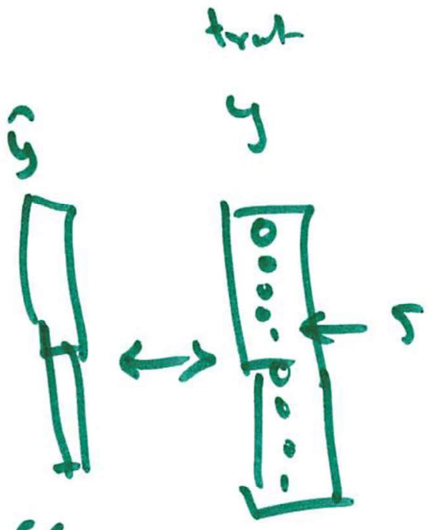


$$(784 \times 64 + 64) + ((64 \times 10) + 10) = \underline{\underline{50890}}$$

An Example



89%



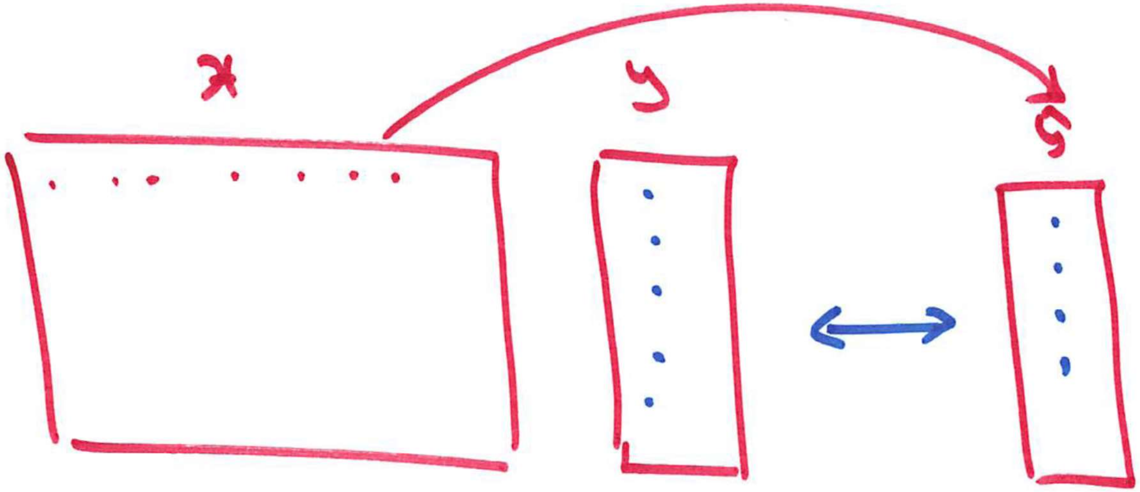
Activation func

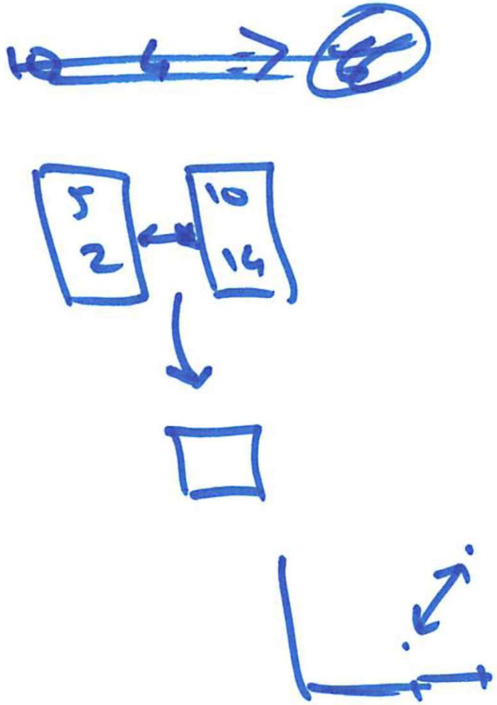
$\sigma(\ )$

$f(\ )$

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

$$f\left(f(f(\ \ldots\ \ldots\ ))\right)$$





$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

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

$$\boxed{\frac{1}{N}\sum_i \left(\hat{y}^i - y^i\right)^2}$$

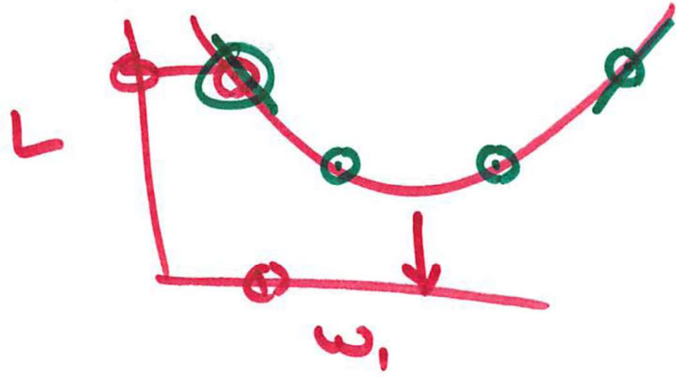
Loss function

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

Training

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

$$\min \sum_i \left((w_1 x_1^i + w_2 x_2^i + w_3 x_3^i + b) - y^i \right)^2 = L(\underline{w}, b) \text{ by changing } w_i, b$$



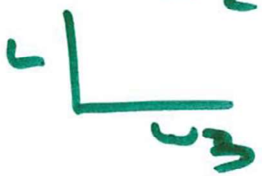
if Slope Neg ~~-ve~~ then move w_1 →

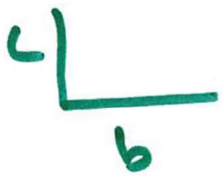
if Slop Pos ~~+ve~~ then move w_1 ←

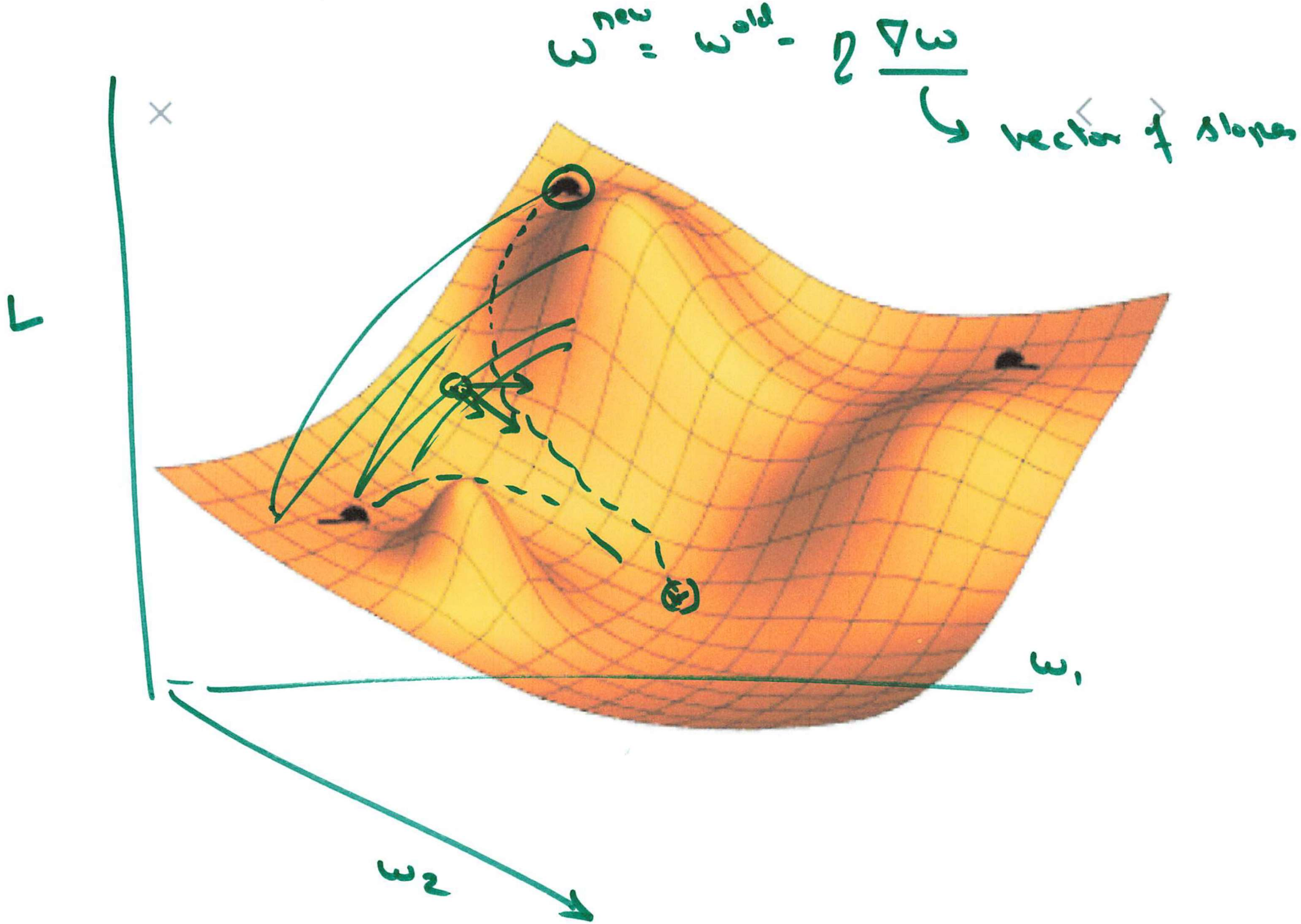
$$\boxed{w_i^{new} = w_i^{old} - \eta \,(\text{Slope along } w_i)}$$

↑ learning rate

Gradient Descent






$w^{new} = w^{old} - \eta \nabla w$
vector of slopes
L
w_1
w_2