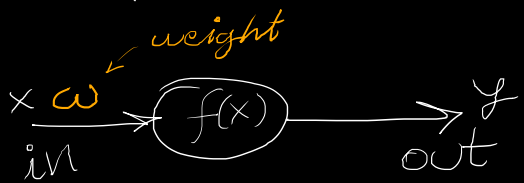
Ann Basic

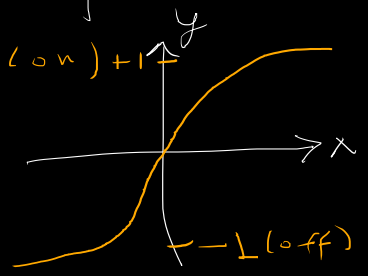
* artificial Neural Network

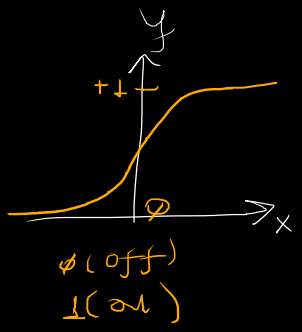
① * An: artificial Neuron
Perceptron



* Neuron's math model
- in = x (numbers) ← known data
- out = y (numbers) ←
- w = weight (number) ← unknown data!
- f(x): activation function

② * f(x):





$$y = f(x \cdot w)$$

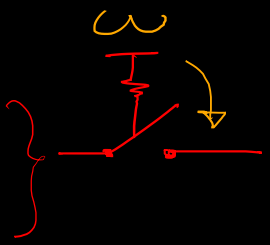
$$y(x) = f(x \cdot w)$$

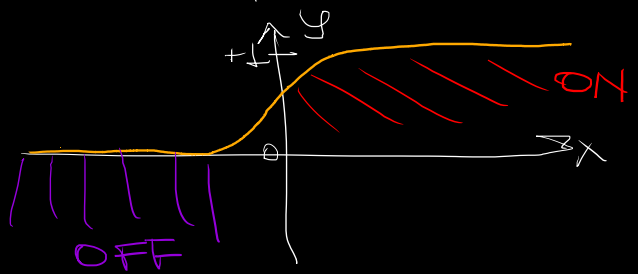
---

③ f(x) Example

Sigmoid:

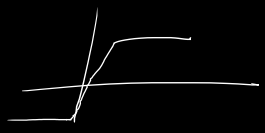
$$f(x) = \frac{1}{1 - e^{-x}}$$

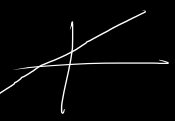




④ What is else?

* Sigmoide

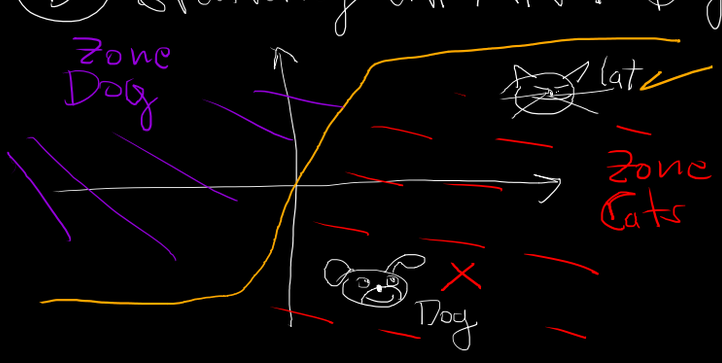
* tanh 

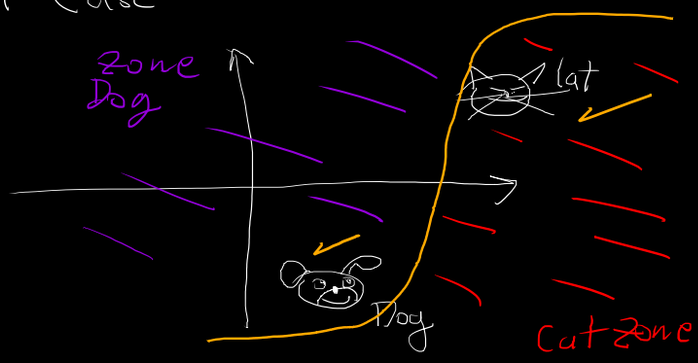
* ReLU 

* maxout ....

etc !!

⑤ Training an AN : Dog vs Cat Case





non-trained AN
"w" is random

trained AN
"w" was computed

What is "train" ? = Solve w !!

⑥ How to Solve ?

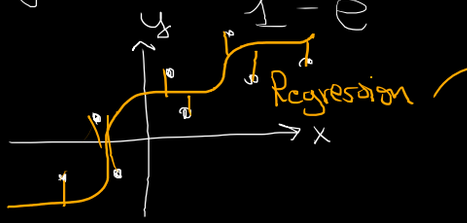
out in

$$y = f(x \cdot w) = \frac{1}{1 - e^{-x \cdot w}}$$

Input(x)	Output(y)
1 (1.0)	cat(0,5)
2	Dog
3	Cat
4	Dog
5	:
6	:

$$y = \frac{1}{1 - e^{-x \cdot w}} \qquad y(x,w) = \frac{1}{1 - e^{-x w}}$$

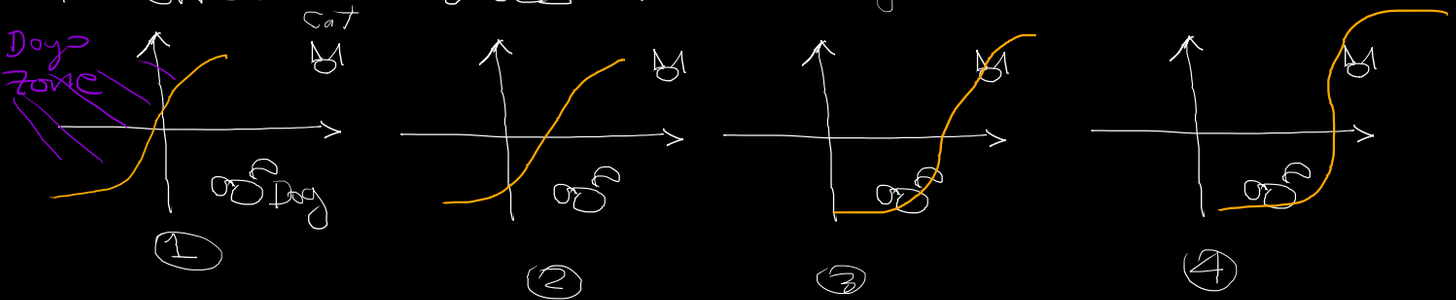
$$x = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$
$$y = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

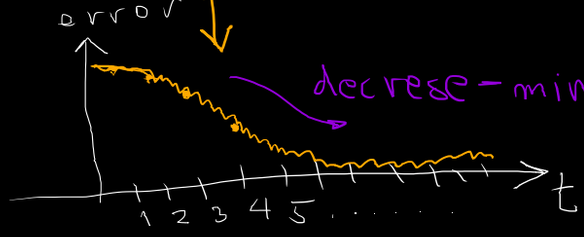


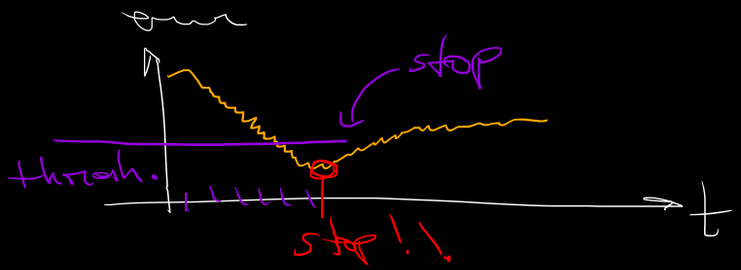
$$\text{Error} = \| \underbrace{y(x,w)}_{\text{real}} - \underbrace{f(x \cdot w)}_{\text{prediction}} \|$$

$$(y - f)^2$$

* Error → Loss ⇨ Training is an iterative process







* Training process :

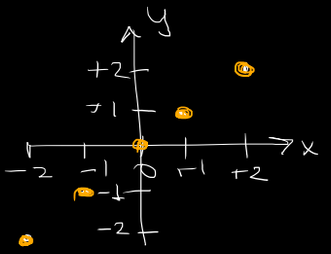
$$\underbrace{\| Loss(w) \|}_{\min} = \underbrace{\| error(w) \|}_{\min}$$

$\min$ $\min$

① Example

$x \rightarrow f(x) \rightarrow y$

x	y
-1	-1
-2	-2
0	0
+1	1
+2	+2



$$y = \frac{1}{1 - e^{-xw}} = y(x, w)$$ ← sigmoid

$$-1 = \frac{1}{1 - e^{-1w}} \qquad -2 = \frac{1}{1 - e^{-2w}} \qquad +2 = \frac{1}{1 - e^{2w}}$$

$$0 = \frac{1}{1 - e^{0w}} \qquad +1 = \frac{1}{1 - e^{1w}}$$

$\forall w$, all eqs are true

$w = 0$ $\quad w = 1$ $\quad w = 2$ $\quad w = -1$ $\quad w = \ldots\ldots$

$y = f(x)$ / suppose $w = 1$

$$-1 = \frac{1}{1 - e^{-1w}} = \frac{1}{1 - e^{-1}} = -0,8$$

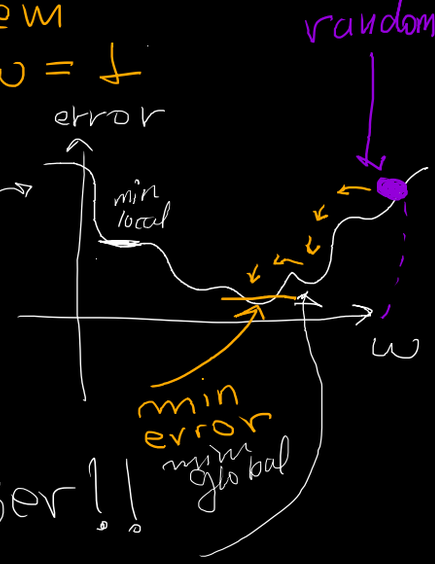
-1 ← i want it; -0,8 ← predicted

$$\text{error} = \|-1 - 0,8\| = 0,2$$

) when $w = 1$

For any "w" $\text{error}(w) = \left\| y - \frac{1}{1 - e^{xw}} \right\|$

$= \|y - f(x \cdot w)\|$



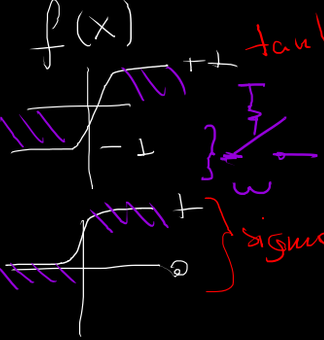
How to find min error? Optimizer!!

* Summary:

: perceptron

variables:

- x, y, f(x) } data known
- w — data known

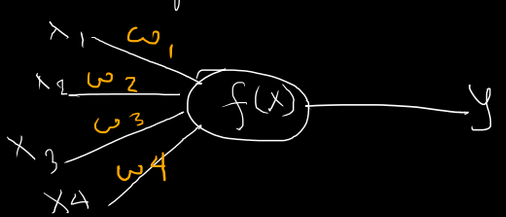


Train: loop iterative

Search w optimo

loss / error } min ↓ decreases error !!

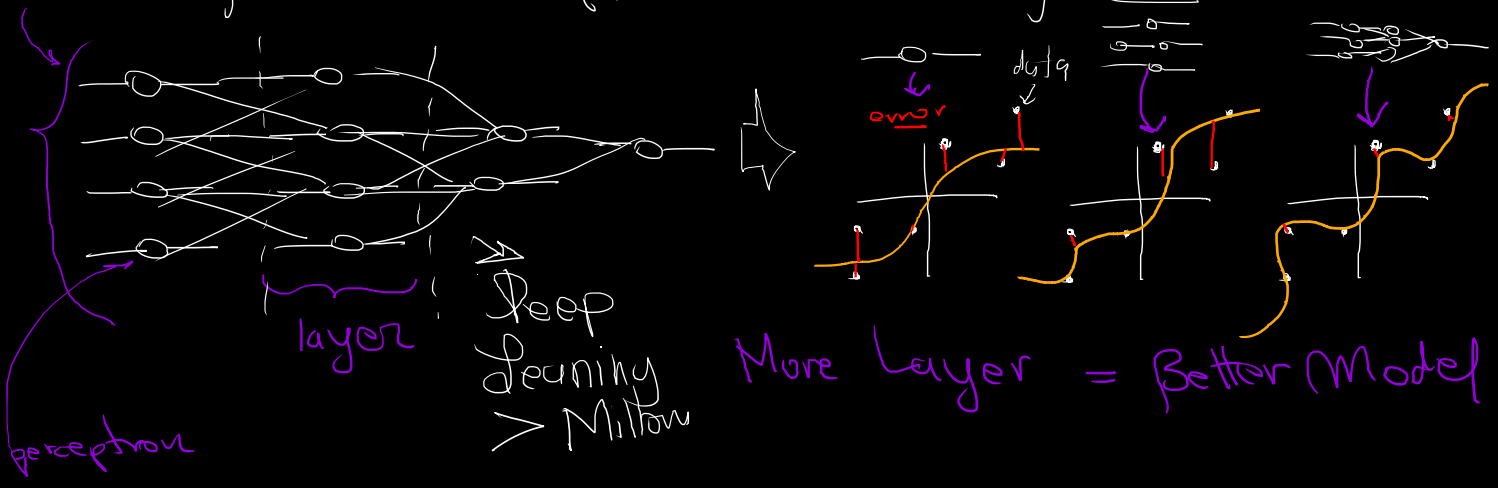
Model General



$$y = f\left(\sum_i x_i w_i\right)$$

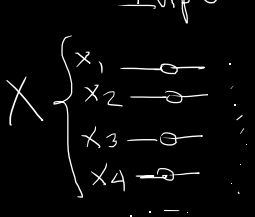
MLP

① Multi Layer Perceptron (Math model of Dense Brain)

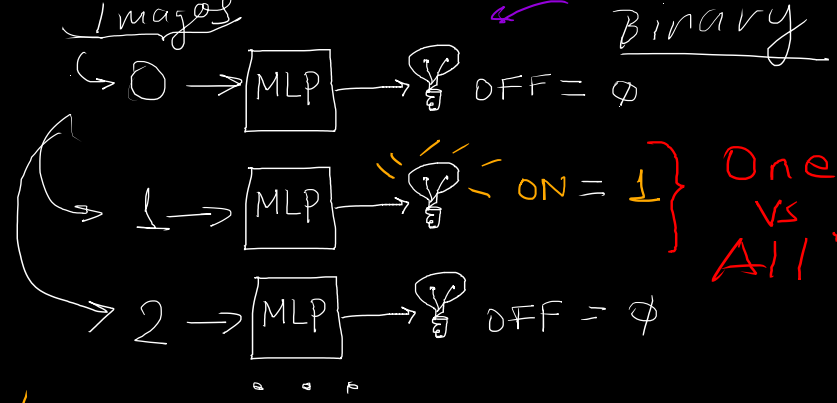


② Practical Example: Numbers Classifier

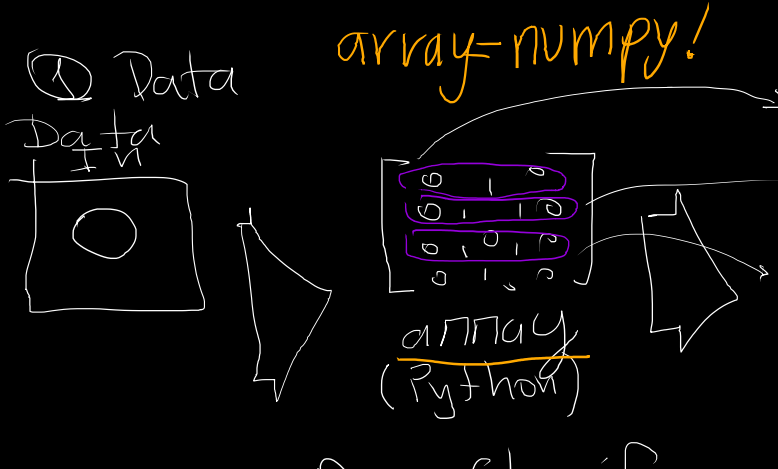
Input Output



Images

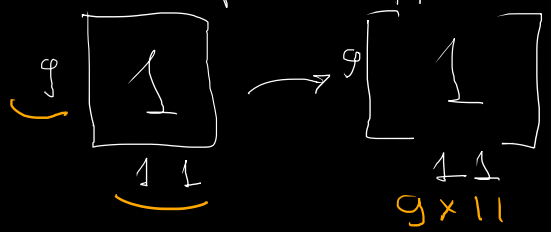


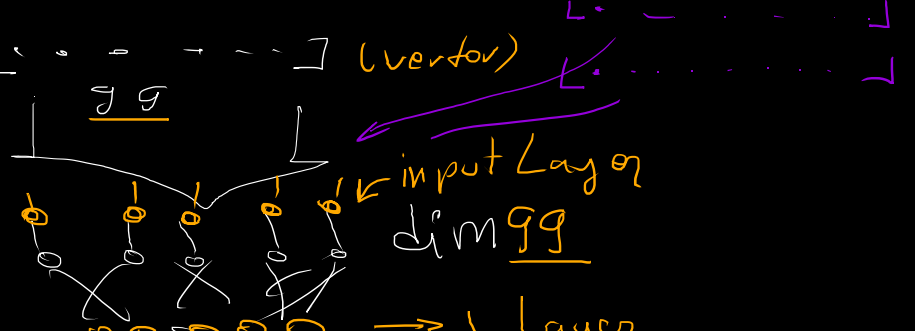
① Data



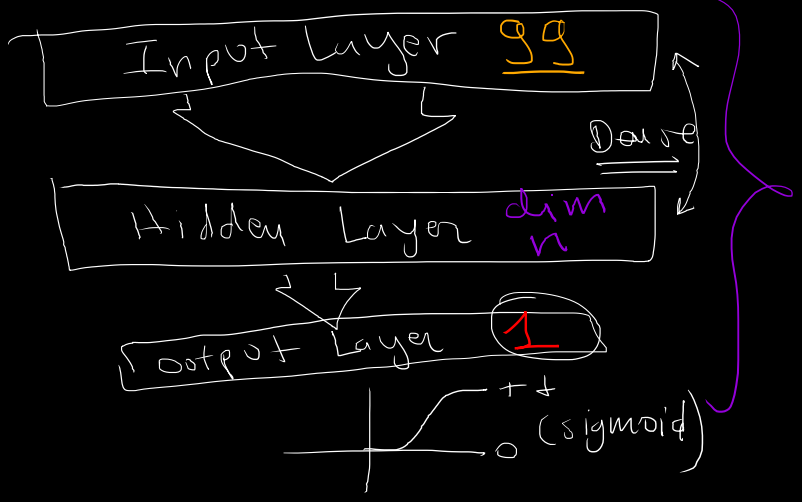
One Classif.

* Implementation:



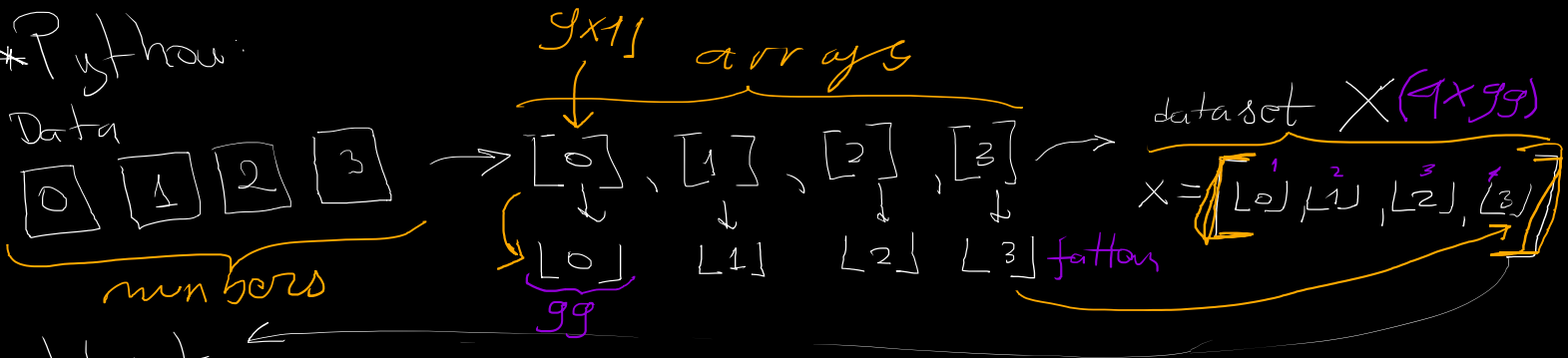


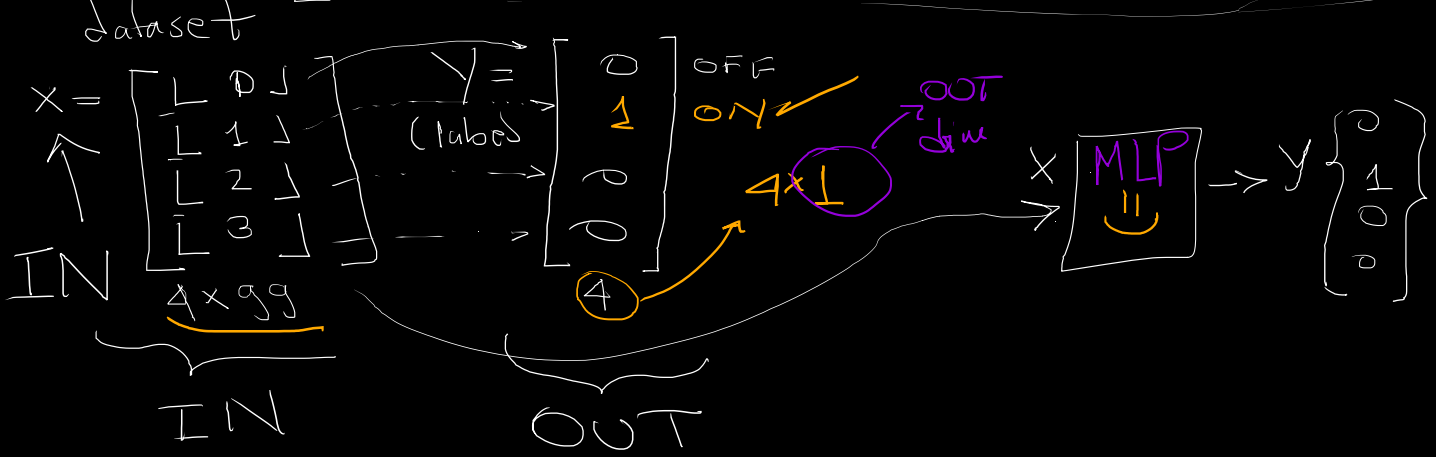
* Build Model



* Model
* Data In/out
* Optimizer (Gradient)
* iterative process ....
0, 1, 2, 3, ...
loop

* Python:

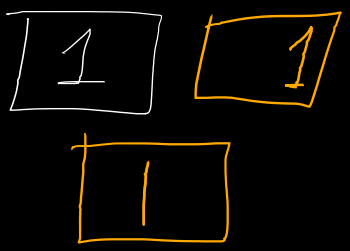




Binary Classifier:

One vs All

Class "One"



Class "Whatever"

Size 1 + 2 = 3 → 3

* How improve?