Universitat de les Illes Balears

Machine Learning

Lesson 3: Supervised Learning

Linear Models (LMS, Logistic Regression, Perceptron)

Linear models

Linear regression

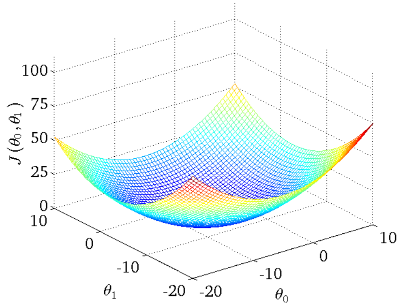
Hypothesis:

the model is linear

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j$$

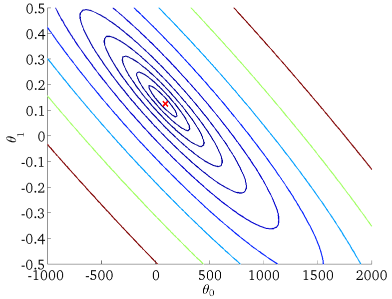
Cost function

How much costs me to be wrong ?



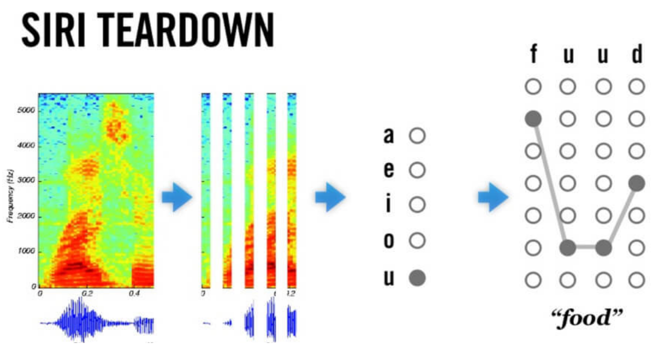
Gradient descent

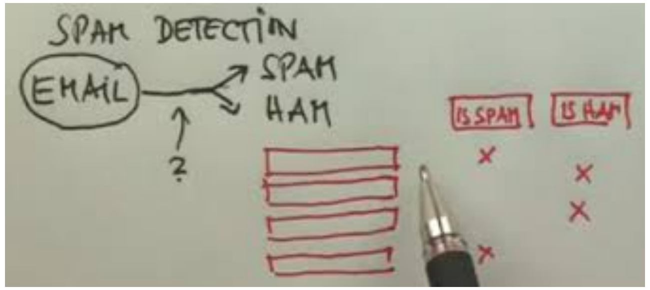
Using the gradient to find the minimum (batch & incremental) and cost

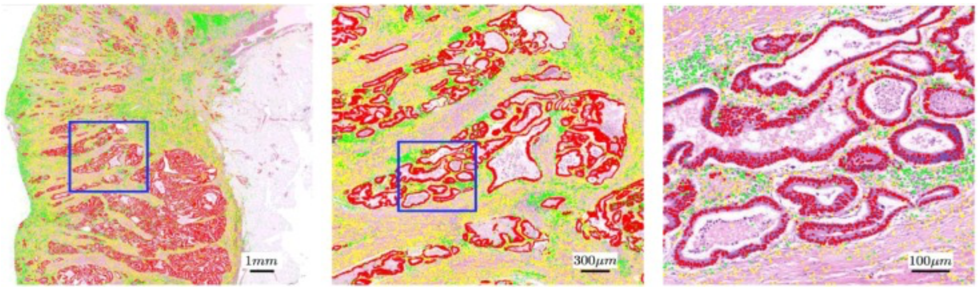


ML Model: classification

In machine learning, **classification** is the problem of identifying to which of a set of <u>categories</u> a new observation belongs, on the basis of a training set of data containing observations whose category membership is known.







Binary classification

There are only two categories

$$y = \{0,1\},$$

where

0: negative class (-)

1: positive class (+)

Task: assign a class for a new input.

Email: Spam / Not Spam
Fraudulent online transactions: Yes / No
Tumor: Malignant / Benign

Dataset = $\{(x^{(i)}, y^{(i)}); i = 1, \dots m\}$

$(x^{(i)}, y^{(i)})$ = Training example

$x^{(i)}$ = "input" variable (*features*), $x \in \mathcal{X}$

$y^{(i)}$ = "output" variable (*target*), $y \in \mathcal{Y}$

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Binary classification – Linear model

Hypothesis: the model is linear

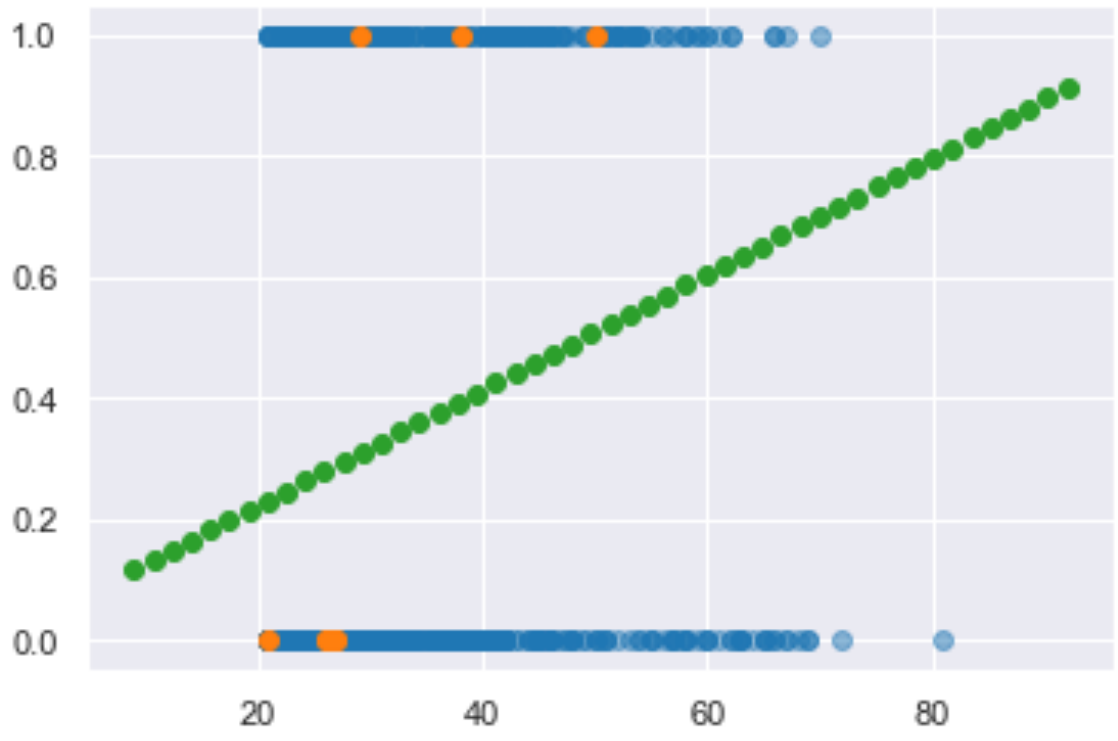
$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} \longrightarrow h_\theta(x) = g\left(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}\right) = \begin{cases} 1, & \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} \geq 0 \\ 0, & otherwise \end{cases}$$

Binary classification – Linear model

Pr(Y = 1 | X = x)

However… It does no sense for h_θ to take values larger than 1 or smaller than 0, therefore we must redefine our hypothesis

- Predicted value is continuous, not probabilistic



Logistic regression

We need a function similar to the threshold, but it has to be **continuous** and **derivable**. We redefine our hypothesis:

$$h_\theta(x) = g\left(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}\right) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}}}$$

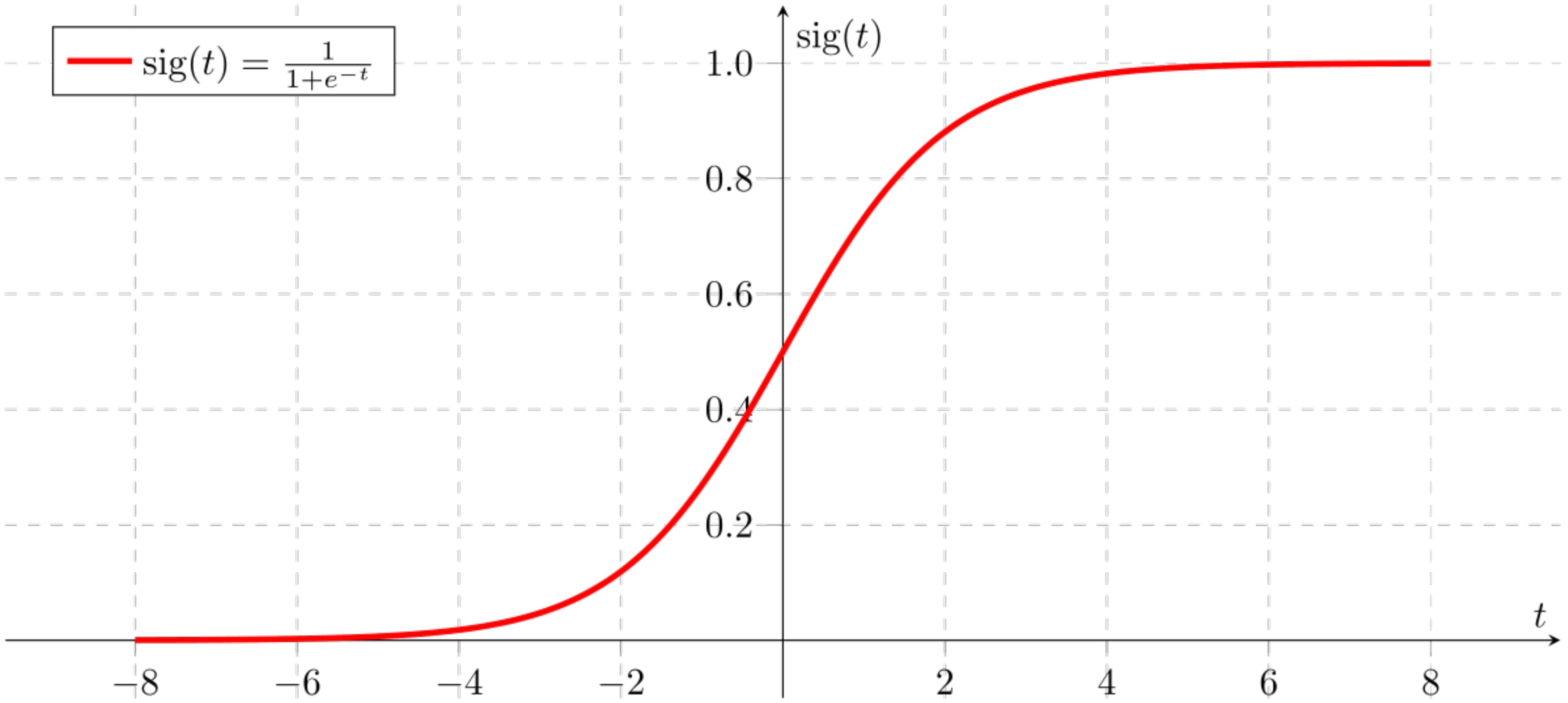
The **logistic function**…

$$g(t) = \frac{1}{1 + e^{-t}}$$

…derived

$$\frac{dg(t)}{dt} = g(t) \cdot (1 - g(t))$$

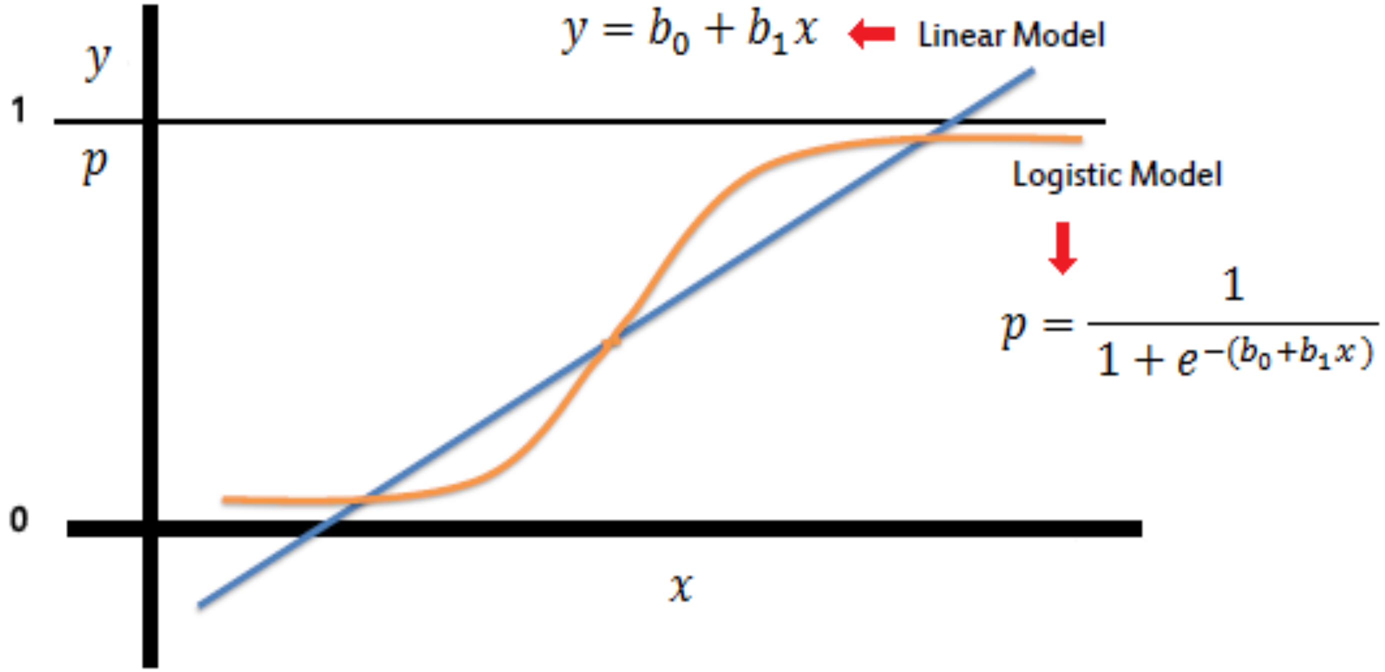
The logistic function



Graphical interpretation

$$h_\theta(x) = g(z) = g\left(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}\right) = \frac{1}{1 + e^{-\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}}}$$

$$\mathbf{z} = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}$$



Probabilistic approach

$h_\theta(\mathbf{x})$, can be interpreted as the estimated probability that $y = 1$ on input $\mathbf{x}$

Odds of success:

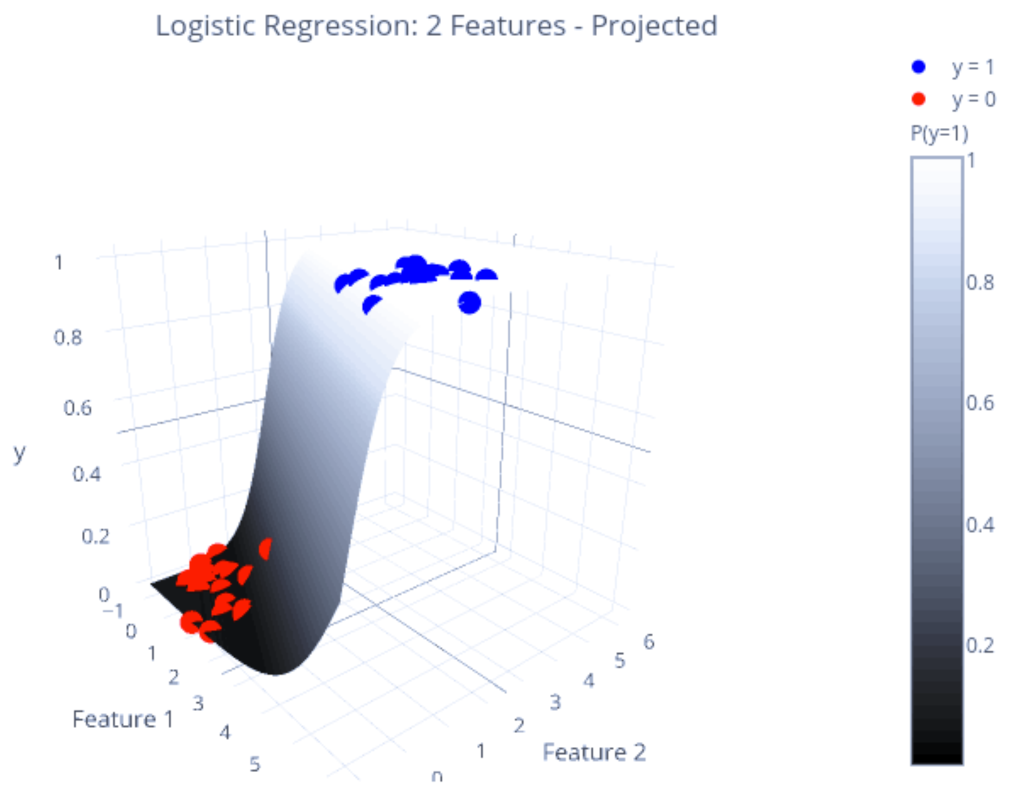
Odds = p / (1 – p)   0 <= odds <= inf

Logit(p) = log(p / (1 – p))   -inf < log(odds) < inf

log(p / (1 – p)) = $\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}$

Exercise:

Manipulate the equation above to get the logistic regression expression



How we define a Loss function for classification ?

loss function is a function

$Loss: (\mathrm{h}_\theta(\mathrm{x}), y) \in \mathbb{R} \times Y \rightarrow Loss(\mathrm{h}_\theta(\mathrm{x}), y) \in \mathbb{R}$,

that takes as inputs the predicted value, $\mathrm{h}_\theta(\mathrm{x})$, corresponding to the real data value, y, and outputs how different they are.

Basic idea: 1-sample cost function

Quadratic Loss function

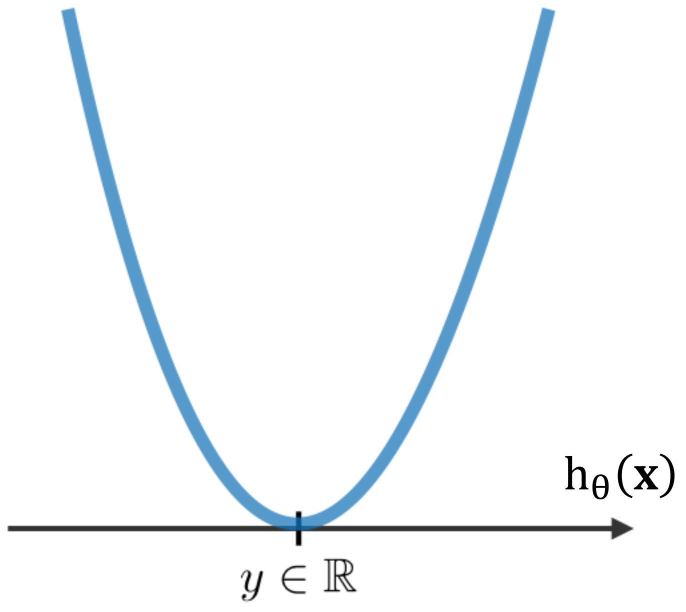
If a **loss function** is a function

$Loss: (\mathrm{h}_\theta(\mathbf{x}), y) \in \mathbb{R} \times Y \rightarrow Loss(\mathrm{h}_\theta(\mathbf{x}), y) \in \mathbb{R}$,

that takes as inputs the predicted value, $\mathrm{h}_\theta(\mathbf{x})$, corresponding to the real data value, y, and outputs how different they are.

<u>Basic idea</u>: 1-sample cost function

Q: $Loss(\mathrm{h}_\theta(\mathbf{x}), y) = (\mathrm{h}_\theta(\mathbf{x}) - y)^2$

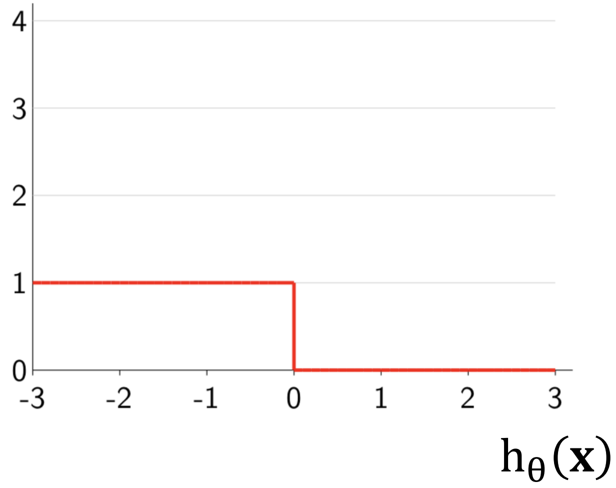


Binary Loss function

Gradient of Loss 0-1 is 0 everywhere, therefore the stochastic **gradient descent is not applicable**.

In addition, Loss 0-1 is insensitive to how badly model messed up.

$L_{0\text{-}1}$: $Loss(\mathrm{h}_\theta(\mathbf{x}), y) = \mathbf{1}[\mathrm{h}_\theta(\mathbf{x}) \cdot y \leq 0]$

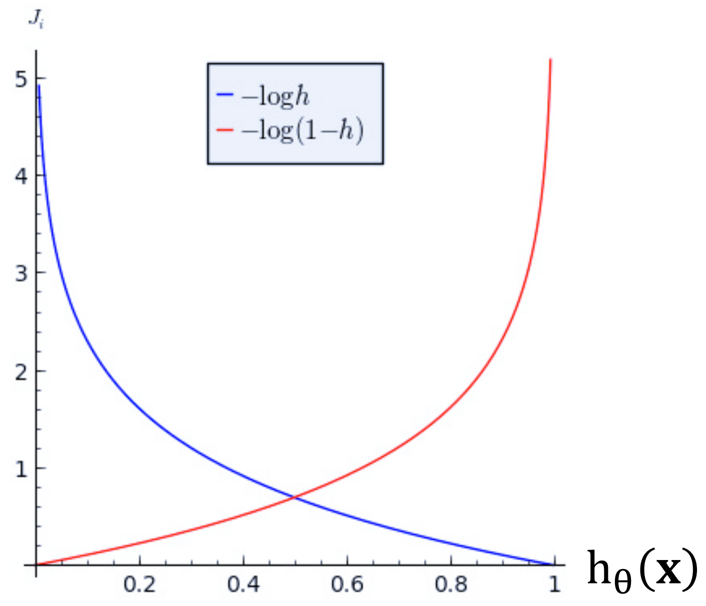


Logistic Loss function

Given a training set, how do we learn the parameters?

<u>Basic idea</u>: to make h_θ close to y and penalize misclassifications

$$Loss(h_\theta(\mathbf{x}), y) = \begin{cases} -\log h_\theta(\mathbf{x}), & y = 1 \\ -\log(1 - h_\theta(\mathbf{x})), & y = 0 \end{cases}$$



Logistic loss vs. MSE for classification

Why not use MSE for classification…

- MSE for classification is not convex → cannot always find local minimum
- Penalization of errors is poor !

Ex. Calculate loss for worst case scenario in a binary classification using:

MSE: LogLoss :

Logistic Regression update rule

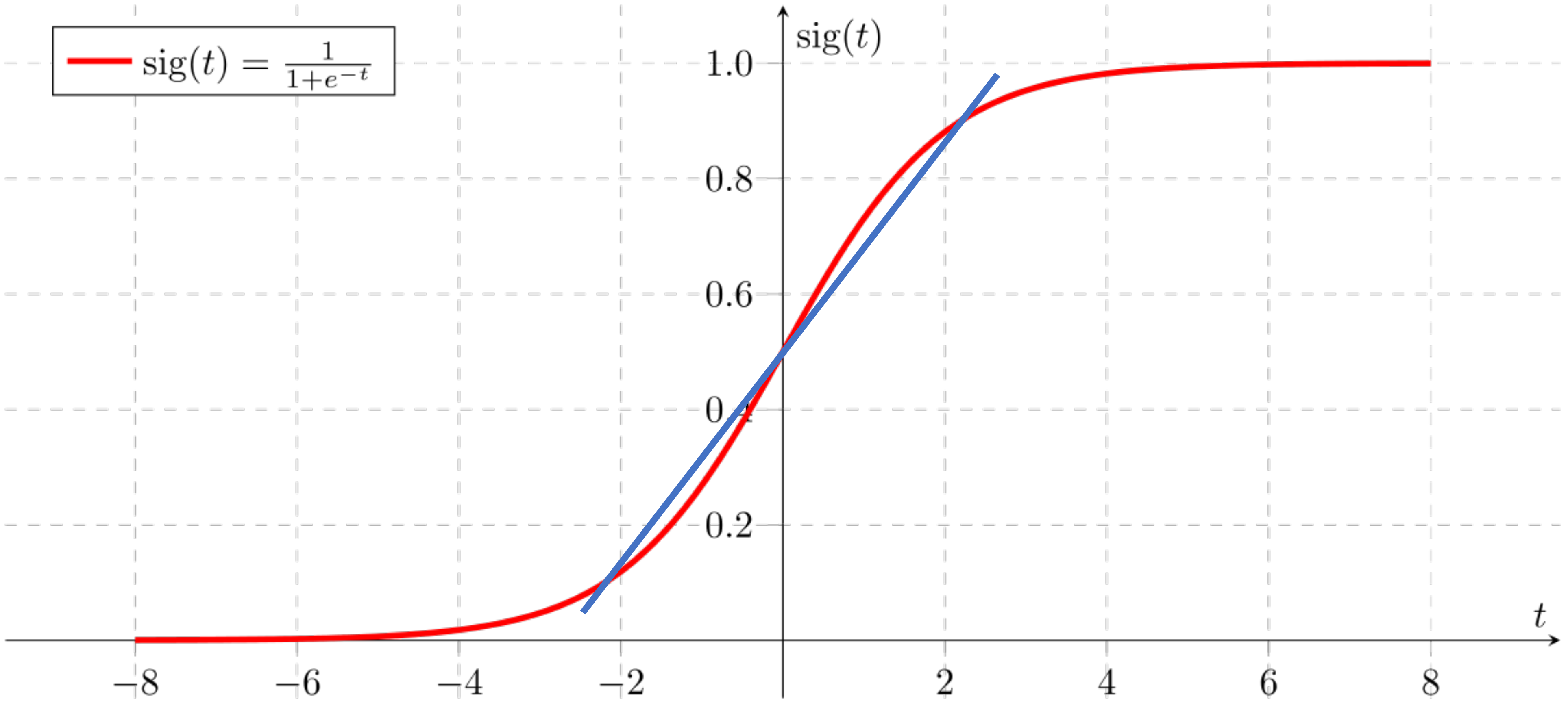
$$J(\theta) = -y \cdot \log h_\theta(\mathbf{x}) - (1-y) \cdot \log\left(1 - h_\theta(\mathbf{x})\right)$$

$$\theta_j := \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta) \longrightarrow \boxed{\theta_j := \theta_j + \alpha \cdot \left(y - h_\theta(x)\right) \cdot x_j}$$

Is it the same update rule as LMS?

http://sambfok.blogspot.com.es/2012/08/partial-derivative-logistic-regression.html

Logistic Regression update rule



Exercise: probabilistic interpretation

0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
0	0	0	0	0	1	1	0	1	0	1	0	1	0	1	1	1	1	1	1

This Table shows the number of hours each student spent studying ML, and whether they passed (1) or failed (0).

If we use a Logistic Regression model with parameters $\theta_0 = -4.0777$ and $\theta_1 = 1.5046$ as a learning result, what would be the probability function of passing conditioned to the number of study hours? What would be the minimum number of hours to have more likely to pass the exam?

Exercise: probabilistic interpretation

From the Dataset, the number of the input data dimensions is *n=1*, then:

$$h_\theta(\mathbf{x}) = g(\theta_0 + \theta_1 x_1) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1)}} = \frac{1}{1 + e^{-(-4.07 + 1.5 x_1)}}$$

that could be interpreted as a function of the probability to pass the exam conditioned to the student number of study hours, $P(\text{“}pass\ the\ exam\text{”} | \mathbf{x}, \theta_0, \theta_1)$.

Exercise: probabilistic interpretation

For the second question we need to compute:

$$h_\theta(\mathbf{x}) > 0.5$$

$$\frac{1}{1+e^{-(-4.07+1.5x_1)}} > \frac{1}{2}$$

Then:

$$e^{-(-4.07+1.5x_1)} = 1$$

$$\log\left(e^{-(-4.07+1.5x_1)}\right) = \log(1)$$

$$4.07 - 1.5x_1 = 0$$

$$x_1 = 2.71$$

Odds of passing exam

hours	odds
1	1:13
2	1:3
3	3:2
4	7:1
5	31:1

2.71 is the minimum number of hours to have more likely to pass the exam.

Neural networks

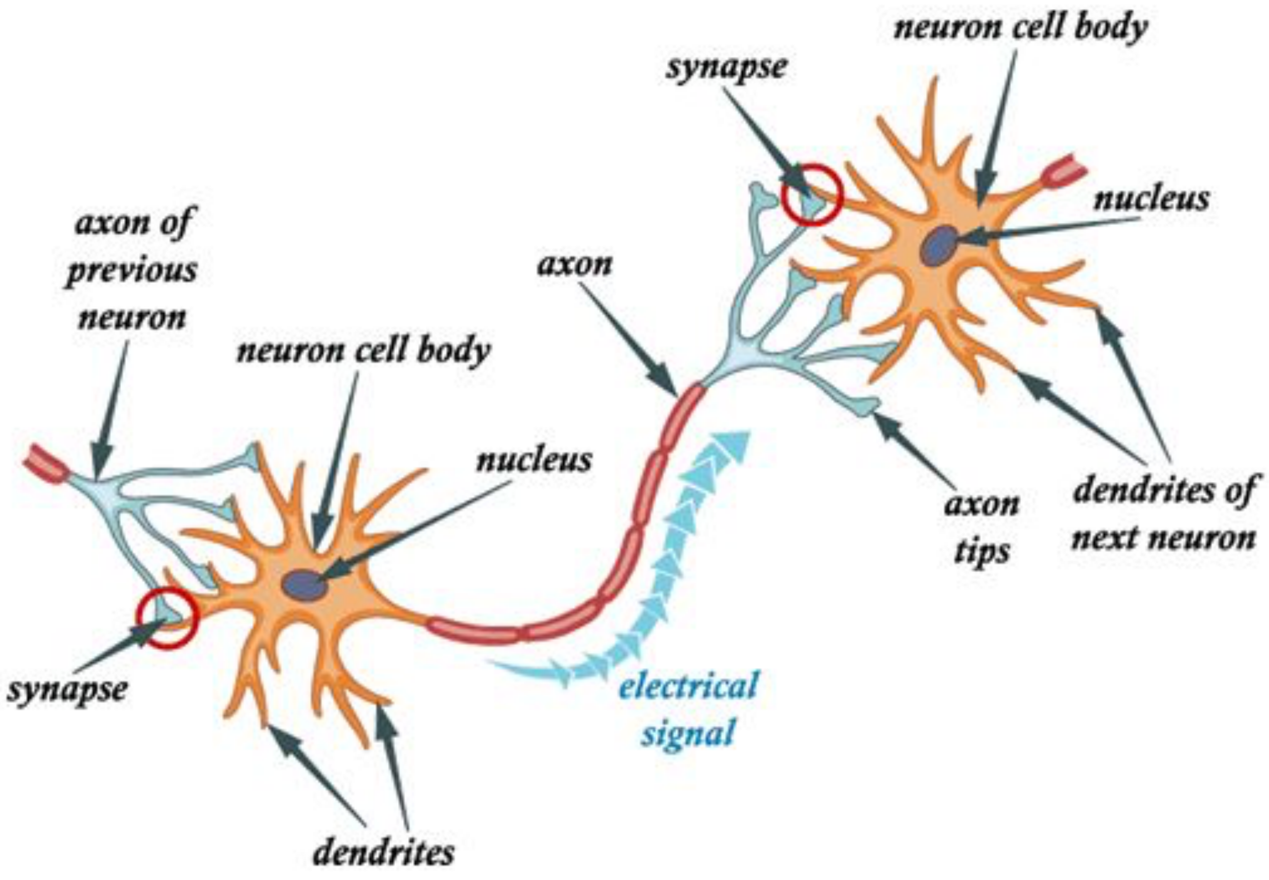
Brain communication is due to a set of nerve fibers that are like wires.

The brain is made up of a gray substance that contains 100,000 million neurons.

Nerve cells or neurons send, receive, store signals, form data and transmit messages.

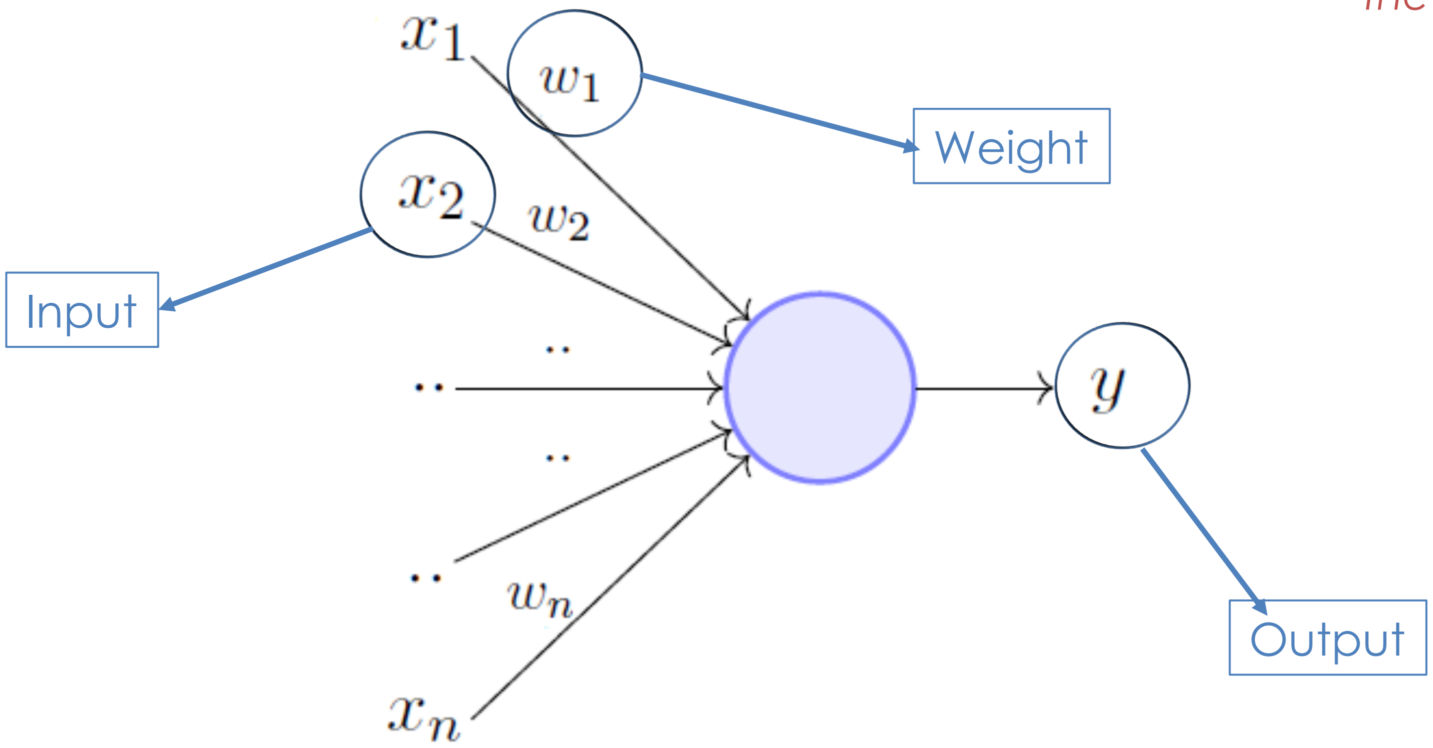
Each neuron has hundreds of connections with other cells

Neuron



neuron cell body
synapse
nucleus
axon of
previous
neuron
axon
neuron cell body
nucleus
axon
tips
dendrites of
next neuron
synapse
electrical
signal
dendrites

Perceptron

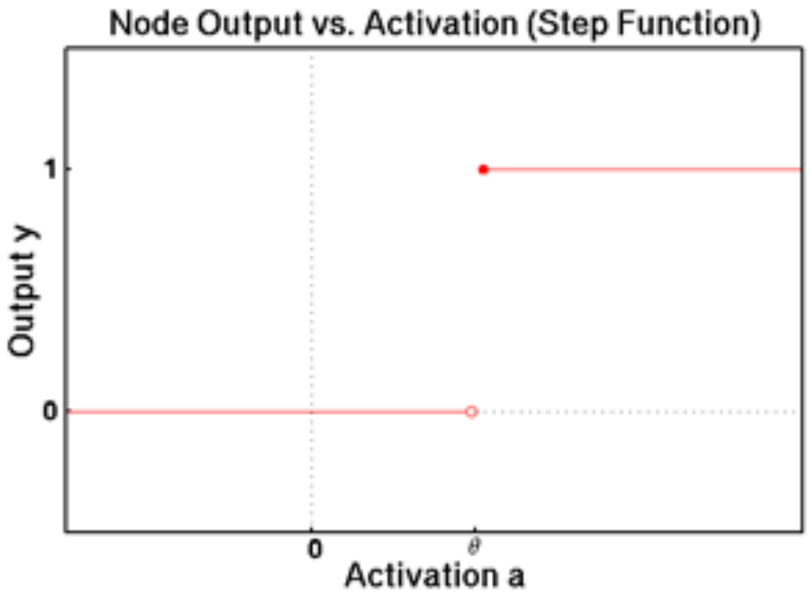


The perceptron was invented in 1958 at the Cornell Aeronautical Laboratory by Frank Rosenblatt.

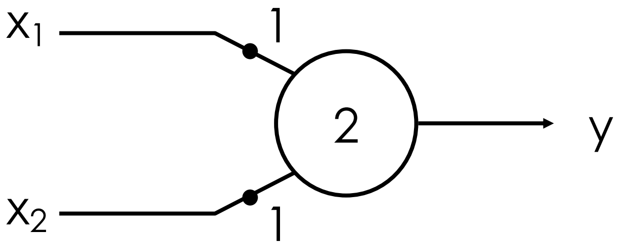
Perceptron model

$$net = \sum_{j=1}^{n} x_j\, w_j$$

$$y = \begin{cases} 1 & if \sum_{j=1}^{n} x_j\, w_j \geq \theta \\ 0 & if \sum_{j=1}^{n} x_j\, w_j < \theta \end{cases}$$



Perceptron example: AND



x_1, x_2	AND	$\sum x_i w_i$	θ	y
0 0	0	0	2	0
0 1	0	1	2	0
1 0	0	1	2	0
1 1	1	2	2	1

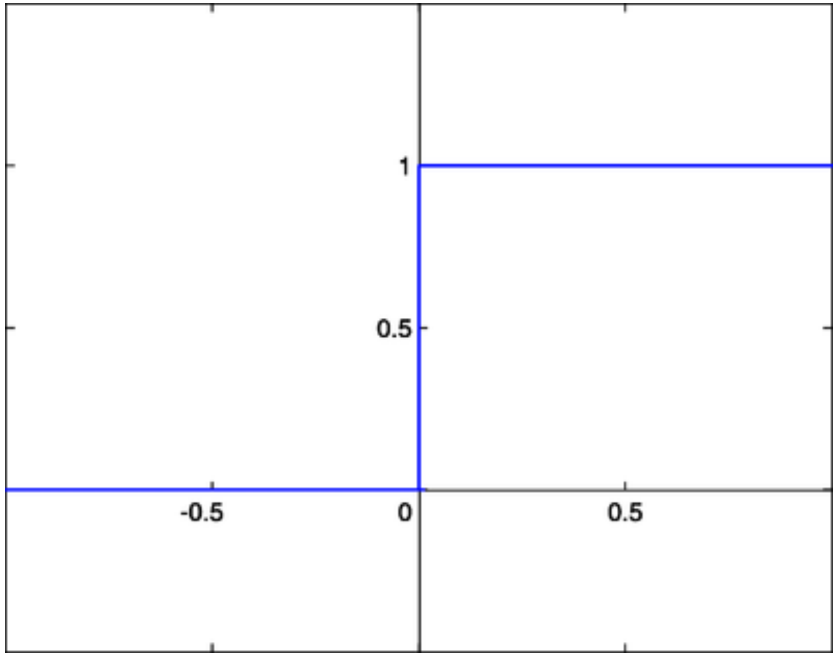
Perceptron model: convention

$$net = \sum_{j=0}^{n} x_j\, w_j$$

$$y = \begin{cases} 1 & if \sum_{j=0}^{n} x_j\, w_j \geq 0 \\ 0 & if \sum_{j=0}^{n} x_j\, w_j < 0 \end{cases}$$

$w_0 = -\theta; \quad x_0 = 1;$

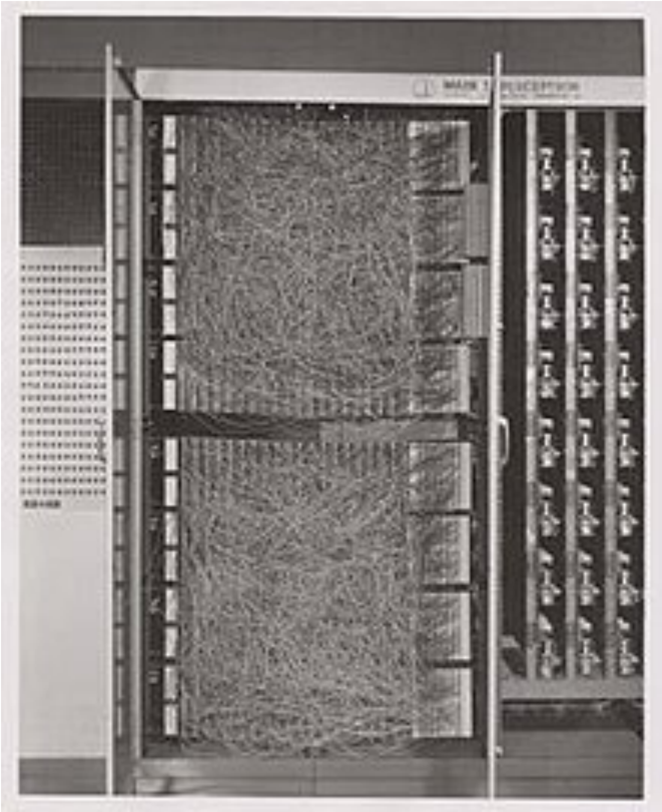
is called the bias because it represents the prior (prejudice)



Perceptron machine

"Mark 1 perceptron": this machine was designed for image recognition: it had an array of 400 photocells, randomly connected to the "neurons". Weights were encoded in potentiometers, and weight updates during learning were performed by electric motors.

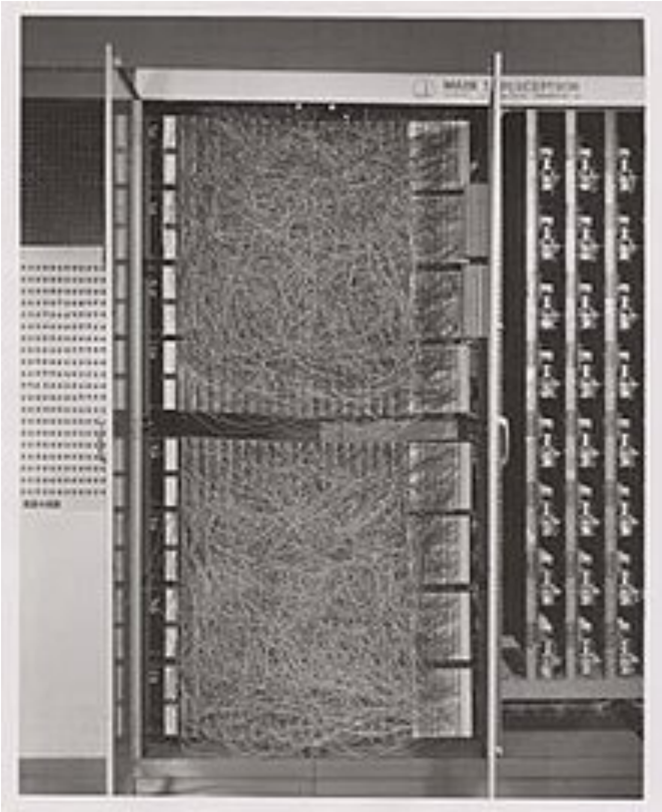
- Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning. Springer.



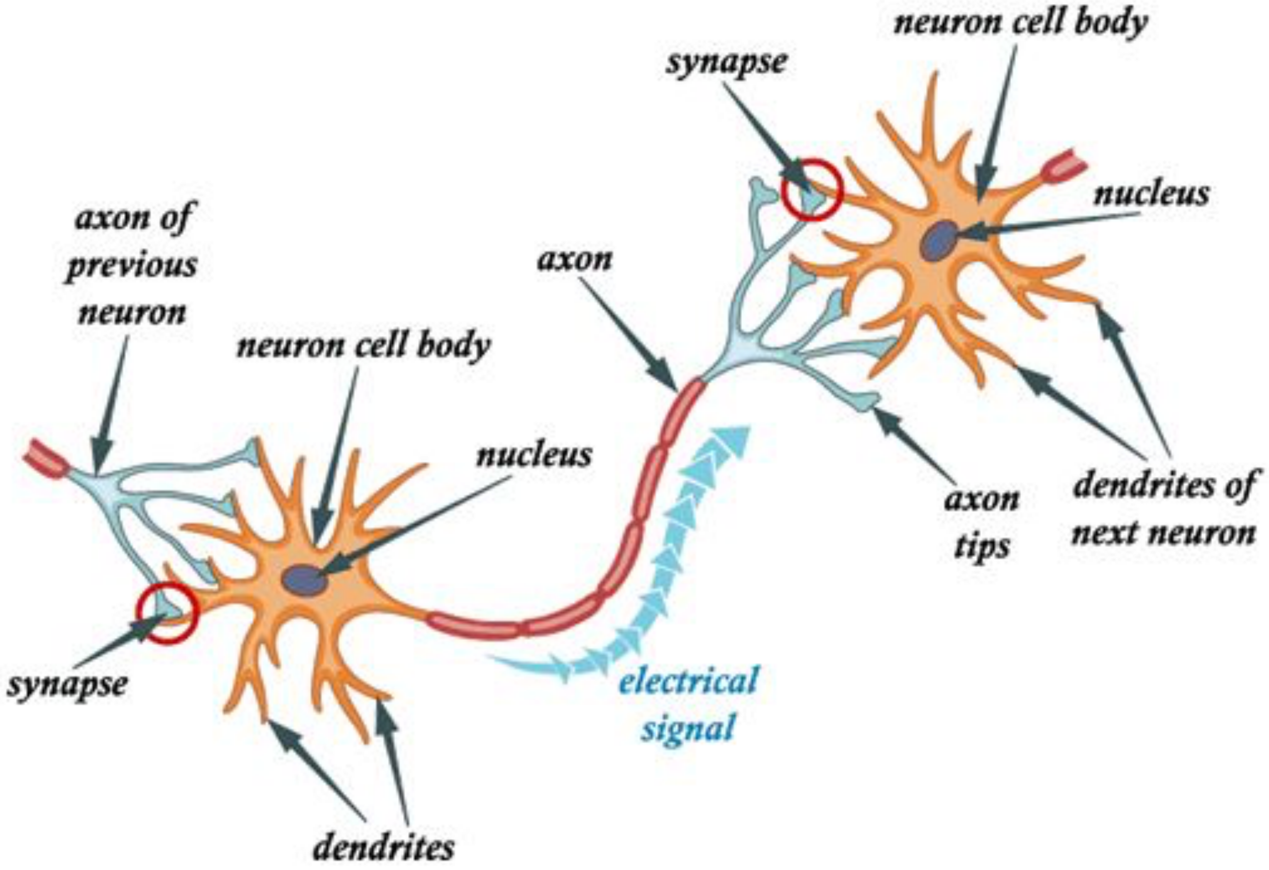
Perceptron machine

The New York Times reported the perceptron to be "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."

- Mikel Olazaran (1996). "A Sociological Study of the Official History of the Perceptrons Controversy". Social Studies of Science. 26 (3): 611–659



Remember: Neuron



Deep learning