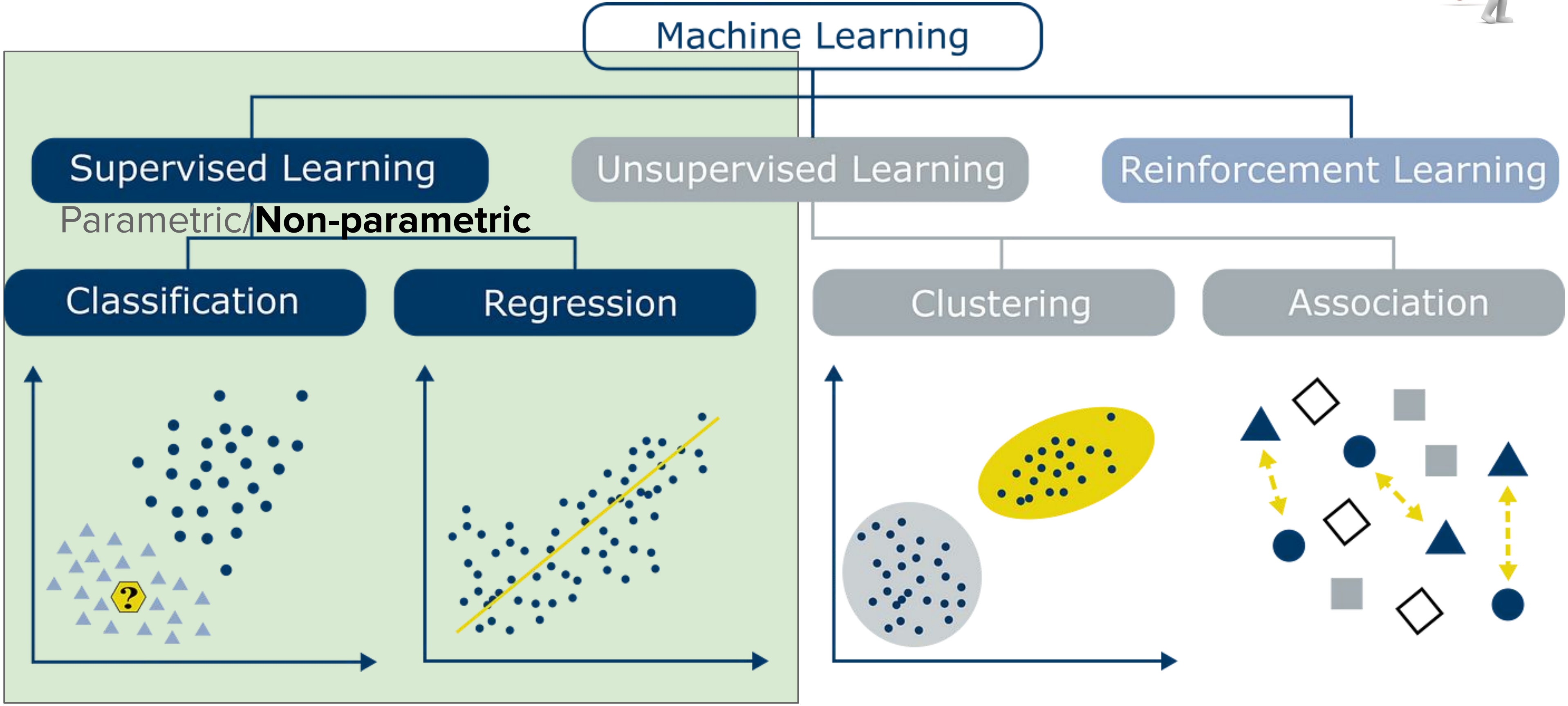
# Supervised Learning: Linear Models

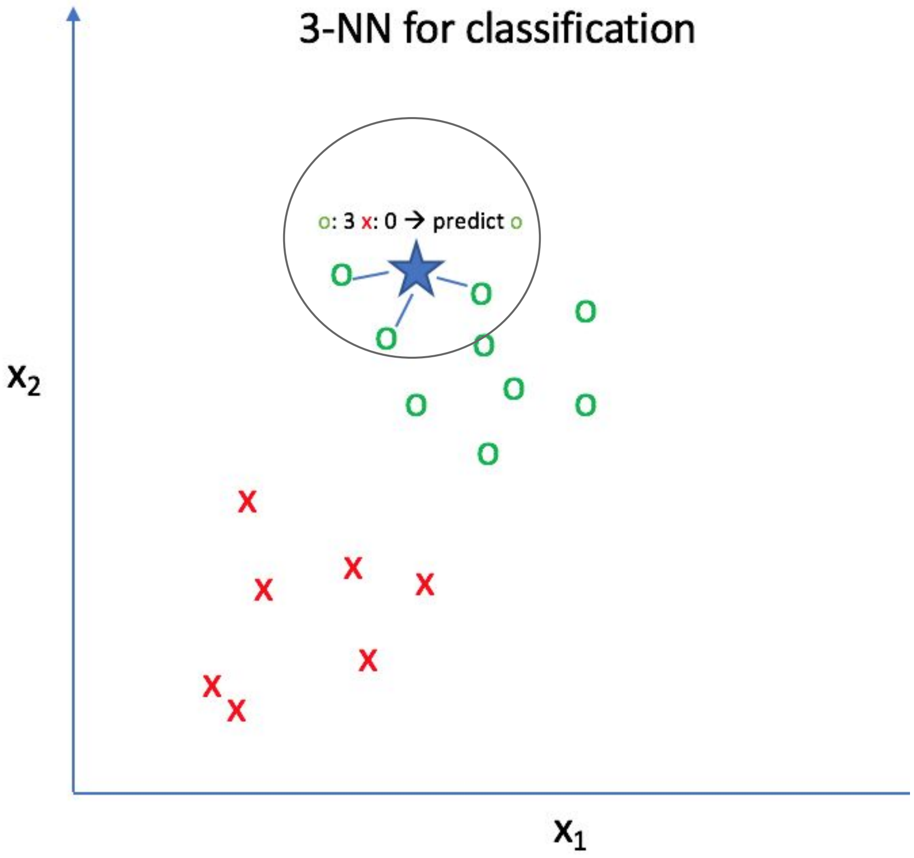
Mahesh Mohan M R
Centre of Excellence in AI
Indian Institute of Technology Kharagpur

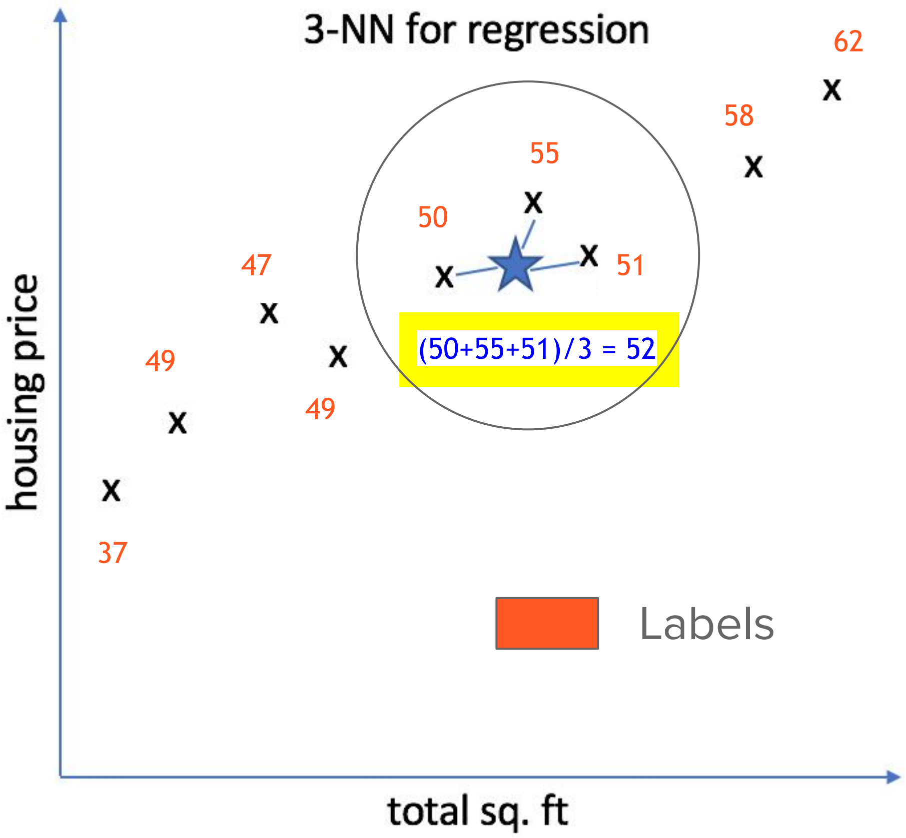
# Recap

# Summary of "Intro to Machine Learning"



# Non-parametric Method: K Nearest Neighb



# Multiplication:
# **Dot product (inner product)**

$$\vec{x} \cdot \vec{y} =$$

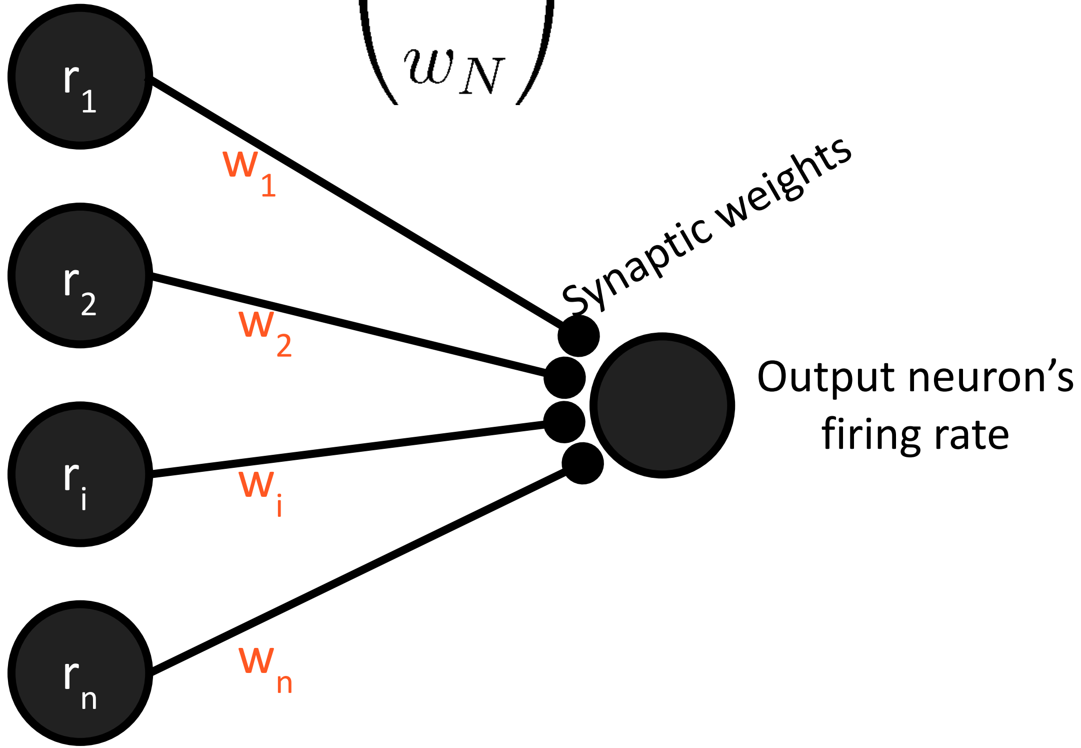
$$\begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$$

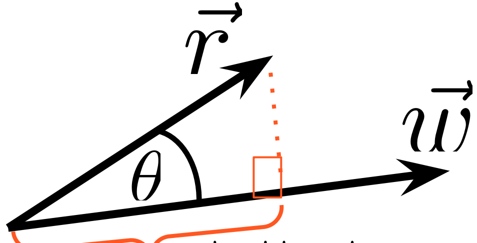
1 X N　　　　N X 1　　　　　　1 X 1

Outer dimensions give
size of resulting matrix

$$\begin{pmatrix} r_1 & r_2 & \cdots & r_N \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = r_1 w_1 + r_2 w_2 + \cdots + r_N w_N$$

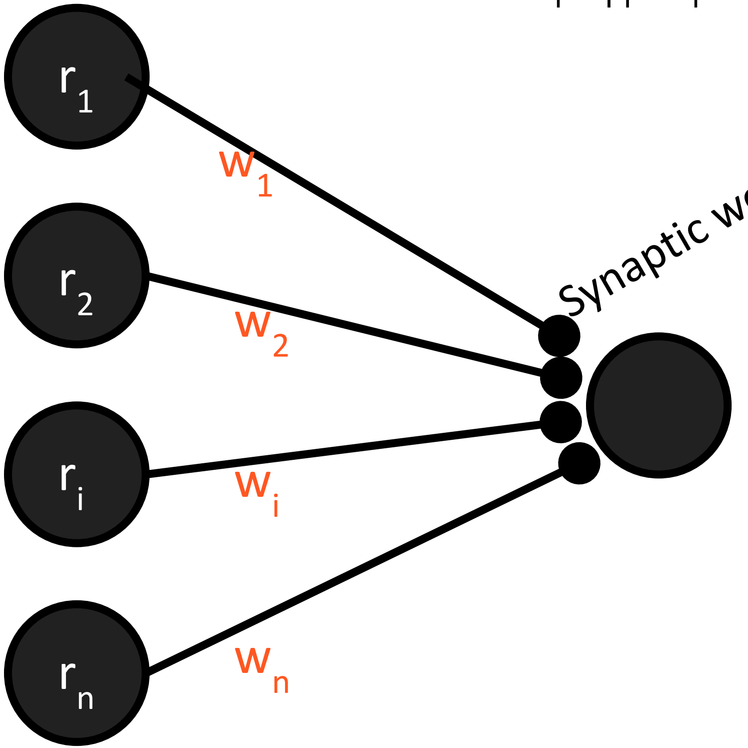
Input neurons'
Firing rates





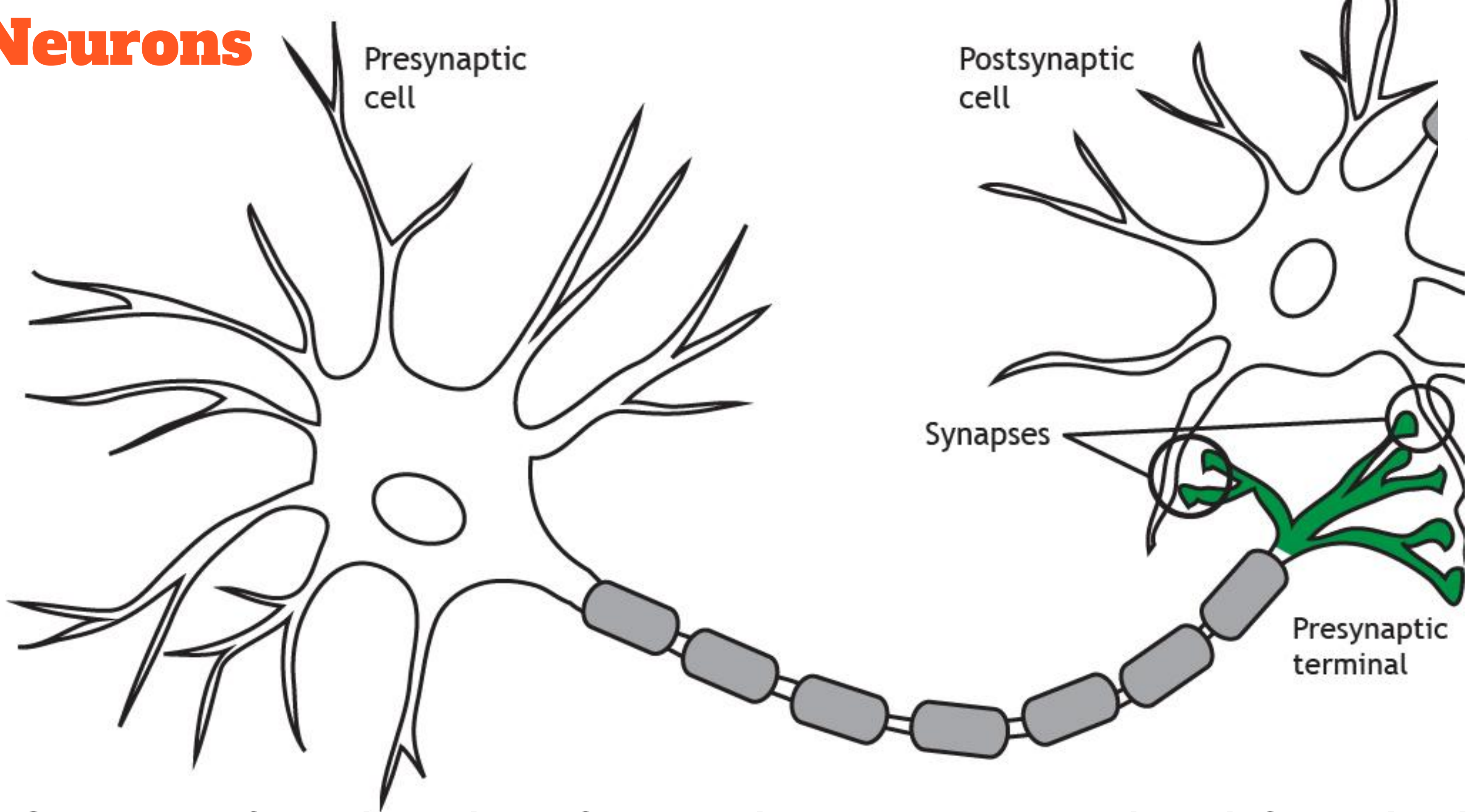

$$\vec{r} \cdot \vec{w} = |\vec{r}||\vec{w}| \cos(\theta)$$

Input neurons'
Firing rates




- <u>Insight</u>: for a given input (L2) magnitude, the response is maximized when the input is parallel to the weight vector
- Receptive fields also can be thought of this way

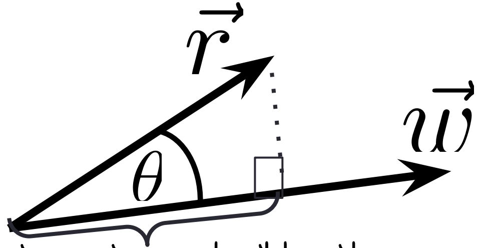
# Neurons



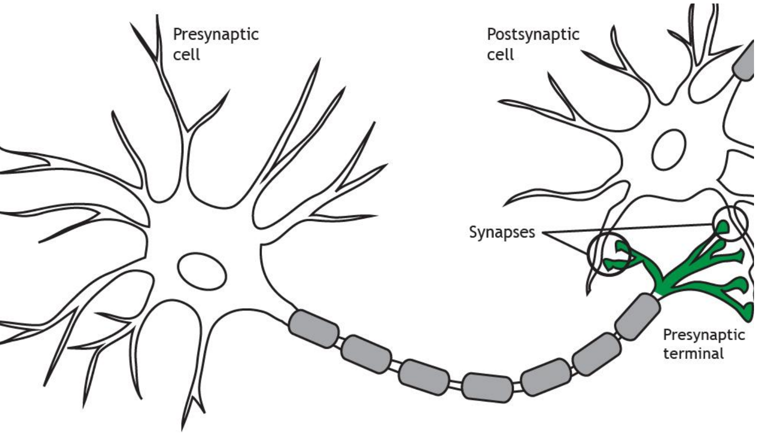

Synapses refer to **the points of contact between neurons where information is passed from one neuron to the next**.

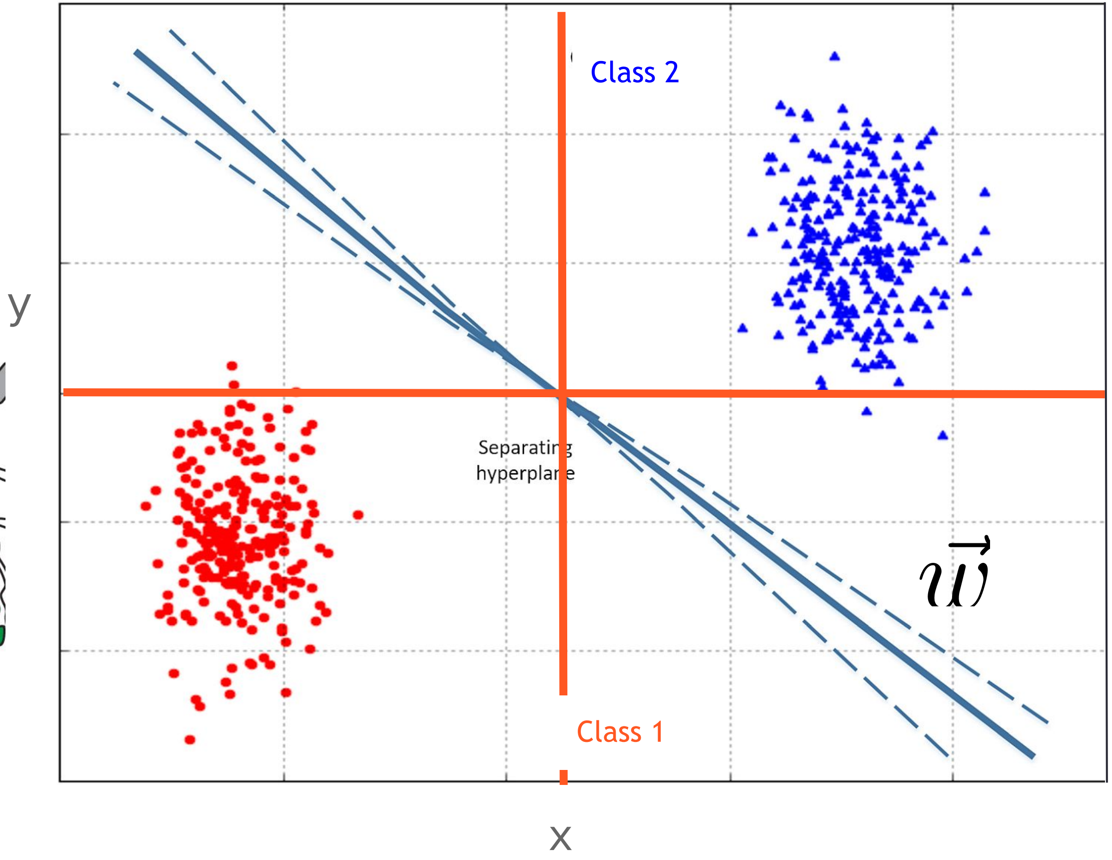
# Linear Models

# Parametric Method: Linear Models for Classification

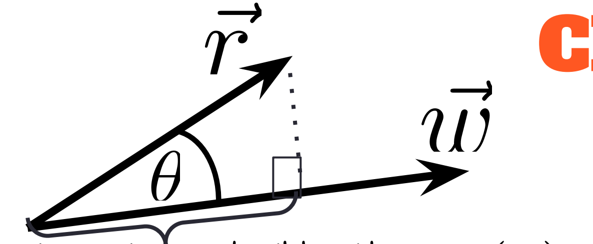


$$\vec{r} \cdot \vec{w} = |\vec{r}||\vec{w}| \cos(\theta)$$

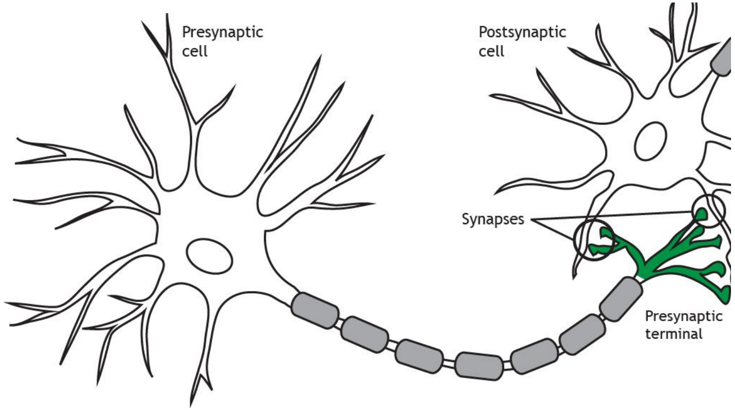


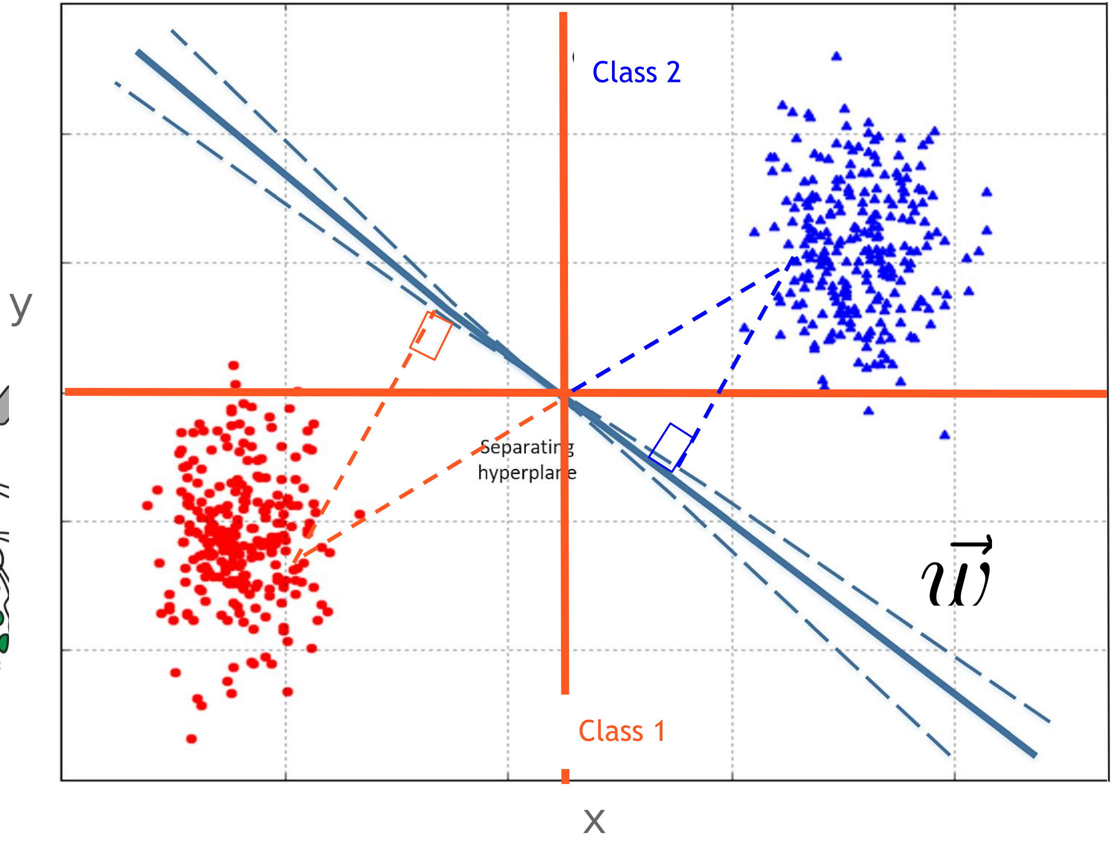



# Parametric Method: Linear Models for Classification

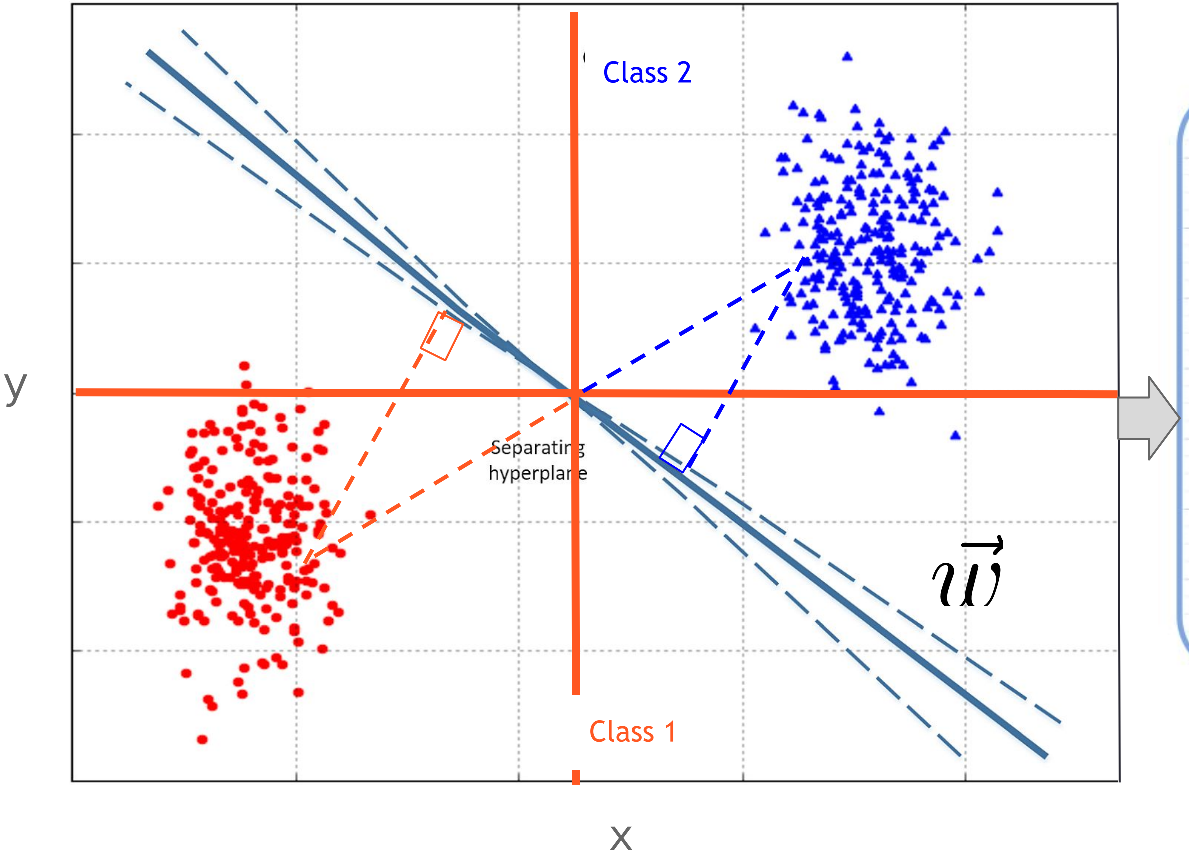


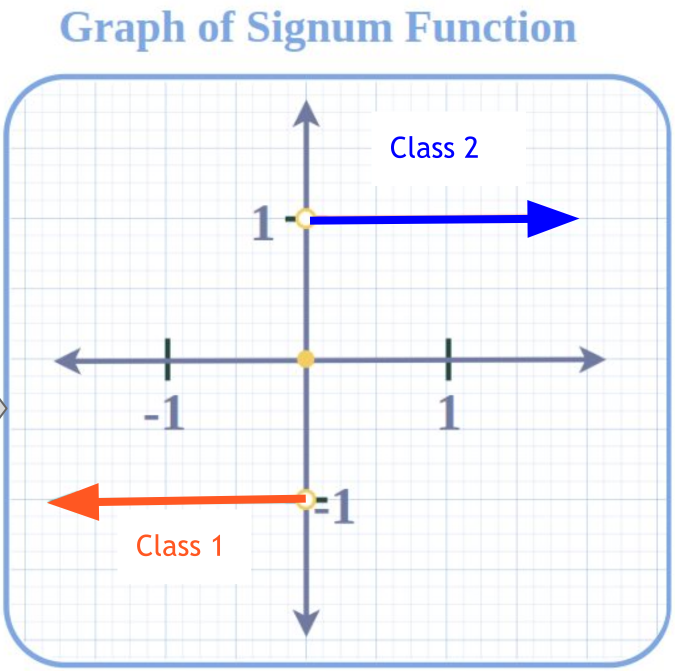
$$\vec{r} \cdot \vec{w} = |\vec{r}||\vec{w}| \cos(\theta)$$



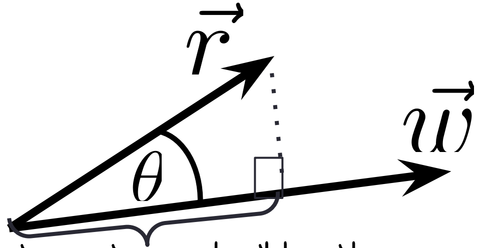



# Parametric Method: Linear Models for Classification

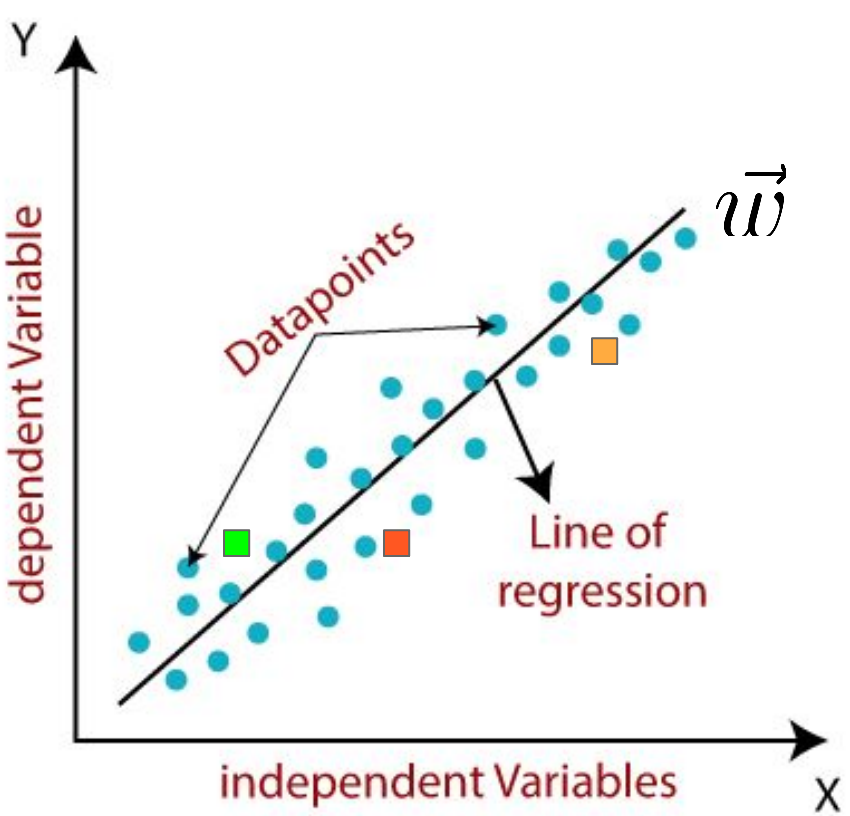


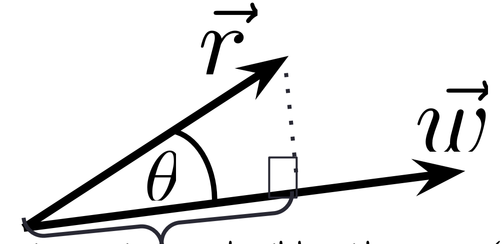
# Parametric Method: Linear Models for Regression



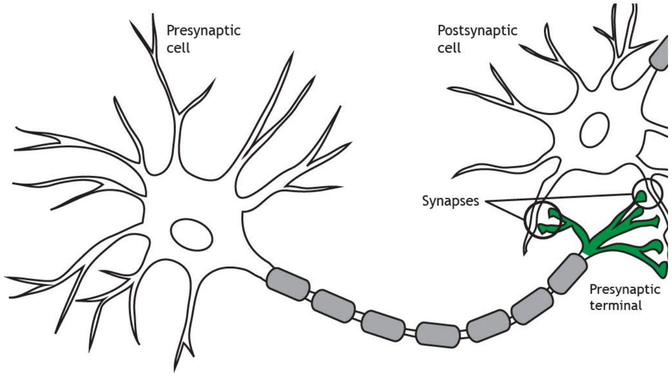

$$\vec{r} \cdot \vec{w} = |\vec{r}||\vec{w}| \cos(\theta)$$

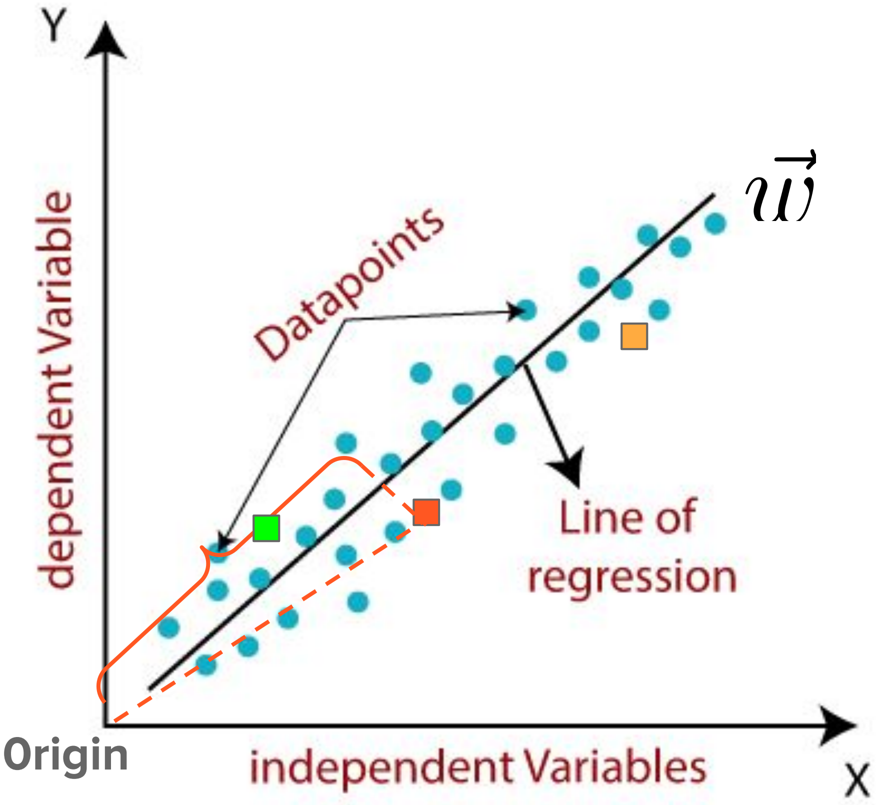


# Parametric Method: Linear Models for Regression

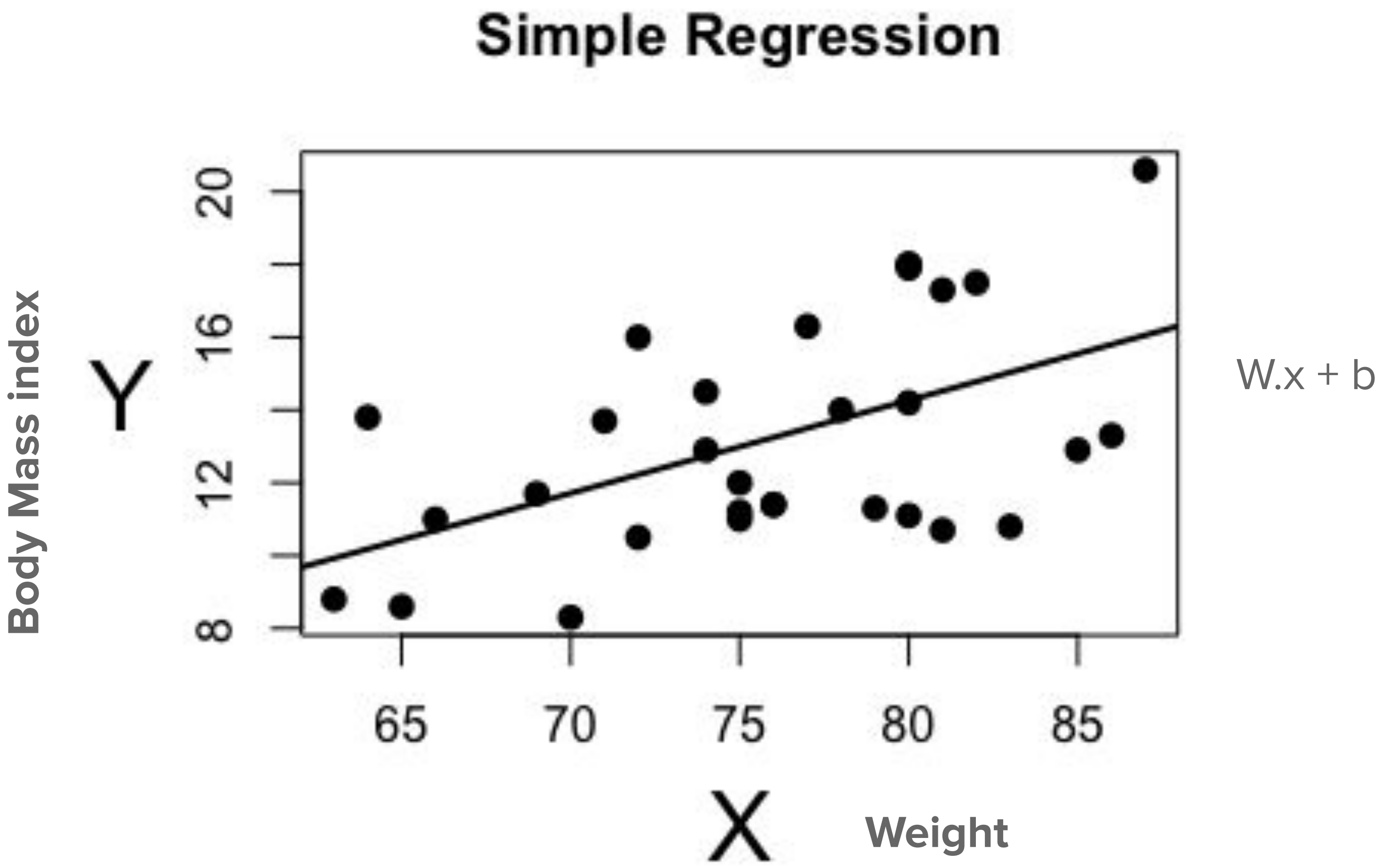


$$\vec{r} \cdot \vec{w} = |\vec{r}||\vec{w}| \cos(\theta)$$



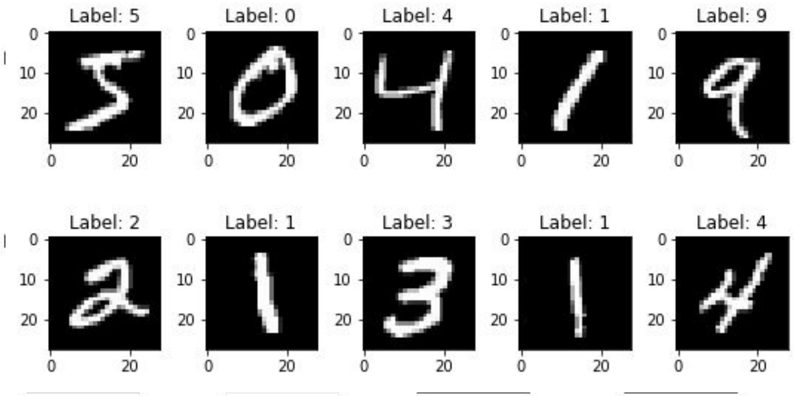
# Linear Models with Bias



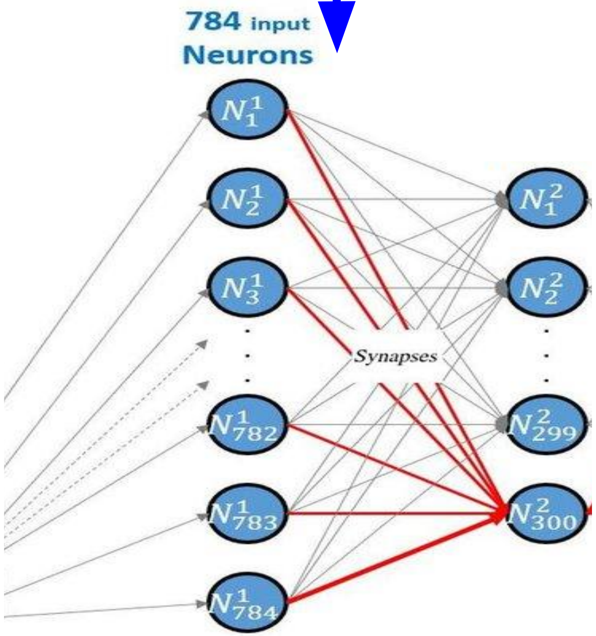
# Recap: AI: How to Solve it?

1. Collect Labelled Dataset
2. Design ANN Architecture
3. Define Loss Function
4. Optimize weights

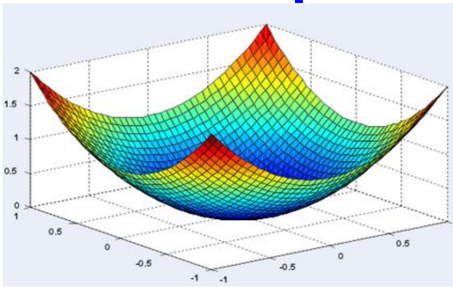
4. Optimize Weight




1. Collect Labelled dataset



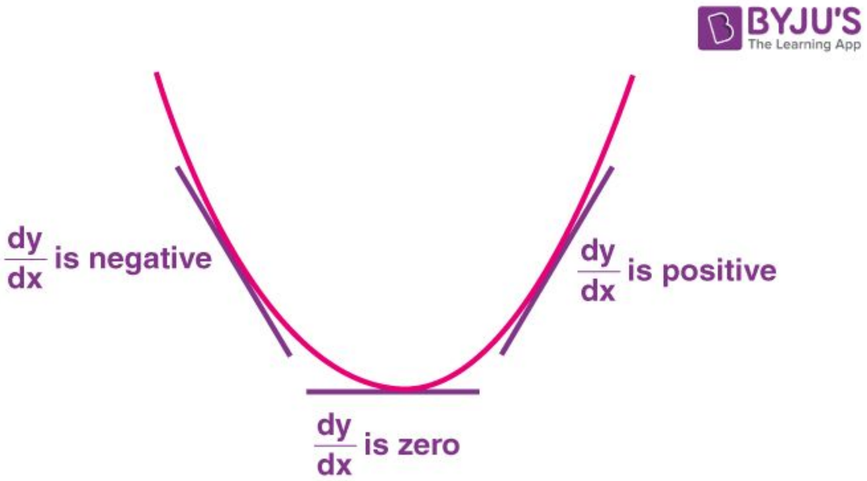

2. Design Artificial Neural Network

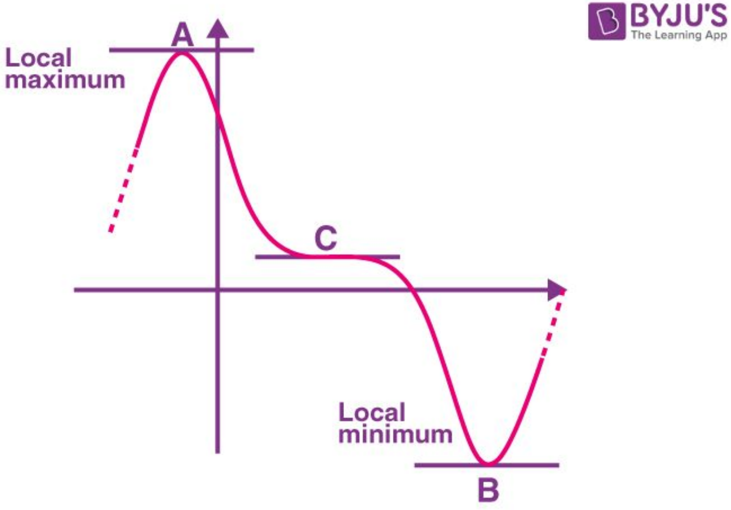


3. Loss Function

# Recap: Closed Form Expression

| | Maximum | Minimum |
|---|---|---|
| Necessary condition | $\frac{dy}{dx} = 0$ | $\frac{dy}{dx} = 0$ |
| Sufficient condition | $\frac{dy}{dx} = 0\ ;\ \frac{d^2y}{dx^2} < 0$ | $\frac{dy}{dx} = 0;\ \frac{d^2y}{dx^2} > 0$ |



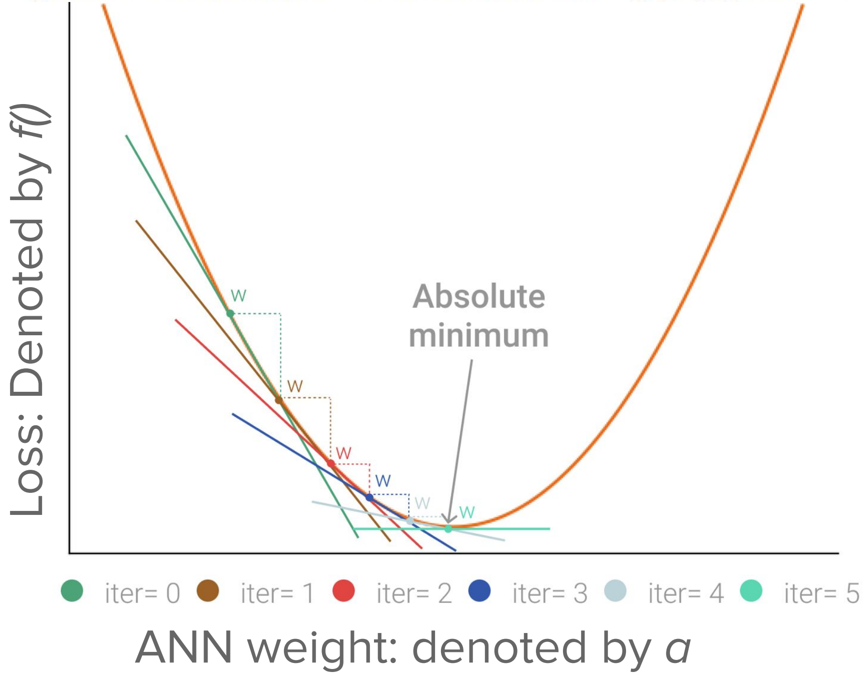



# Recap: Gradient Descent

**Definition 1** *Suppose f is a real valued function and a is a point in its domain of definition. The derivative of f at a is defined by*

$$\lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

*provided this limit exists. Derivative of $f(x)$ at a is denoted by $f'(a)$.*




What happens to Loss *f(a+h) - f(a)* if the weight update *h* is $-\lambda f'(a)$?
Note: $\lambda > 0$

Answer: $f(a+h) - f(a)$ is negative, which means $f(a+h) \le f(a)$.

Optimizing ANN: Update each ANN weight a as $a - \lambda f'(a)$, where $\lambda$ is the learning rate.

# Linear Regression

Labelled Data $\begin{bmatrix} x_{k1} \\ x_{k2} \\ \dots \\ x_{kn} \end{bmatrix} \Rightarrow y_k$

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{d1} & x_{d2} & \dots & x_{dn} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_d \end{bmatrix}$$

$$w = (X^T X)^{-1} X^T y$$

$$\nabla w = (X^T X) w - X^T y$$

$$L = \frac{1}{2} \sum_{n=1}^{N} (x_n^T w_0 + w_b - y_n)^2$$

where $w_0$ is the linear elements
$w_b$ is the bias.

$$= \frac{1}{2} \|Xw - y\|^2$$

$$= \frac{1}{2}(w^T X^T X w - 2 w^T X^T y + y^T y)$$

* $\frac{\delta}{\delta w} w^T a = a$
* $\delta/\delta w \; w^T A w = 2Aw$ → Exercise to be done.

$A$ is similar matrix
i.e; $A = A^T$

$$\frac{\delta L}{\delta w} = (X^T X)w - X^T y \quad \text{—} \; (1)$$

Closed form

$$X^T X w - X^T y = 0$$
$$w = (X^T X)^{-1} X^T y$$

Gradient descent

$$w_{i+1} = w_i - \lambda \frac{\delta L}{\delta w}$$

use (1) for $\frac{\delta L}{\delta w}$

18/11/2024