Machine Learning (part II)

Biological and Artificial Neural Networks

Angelo Ciaramella

Introduction

- Artifical Neural Networks (ANNs) are at the very core of Deep Learning
  - powerful
  - scalable

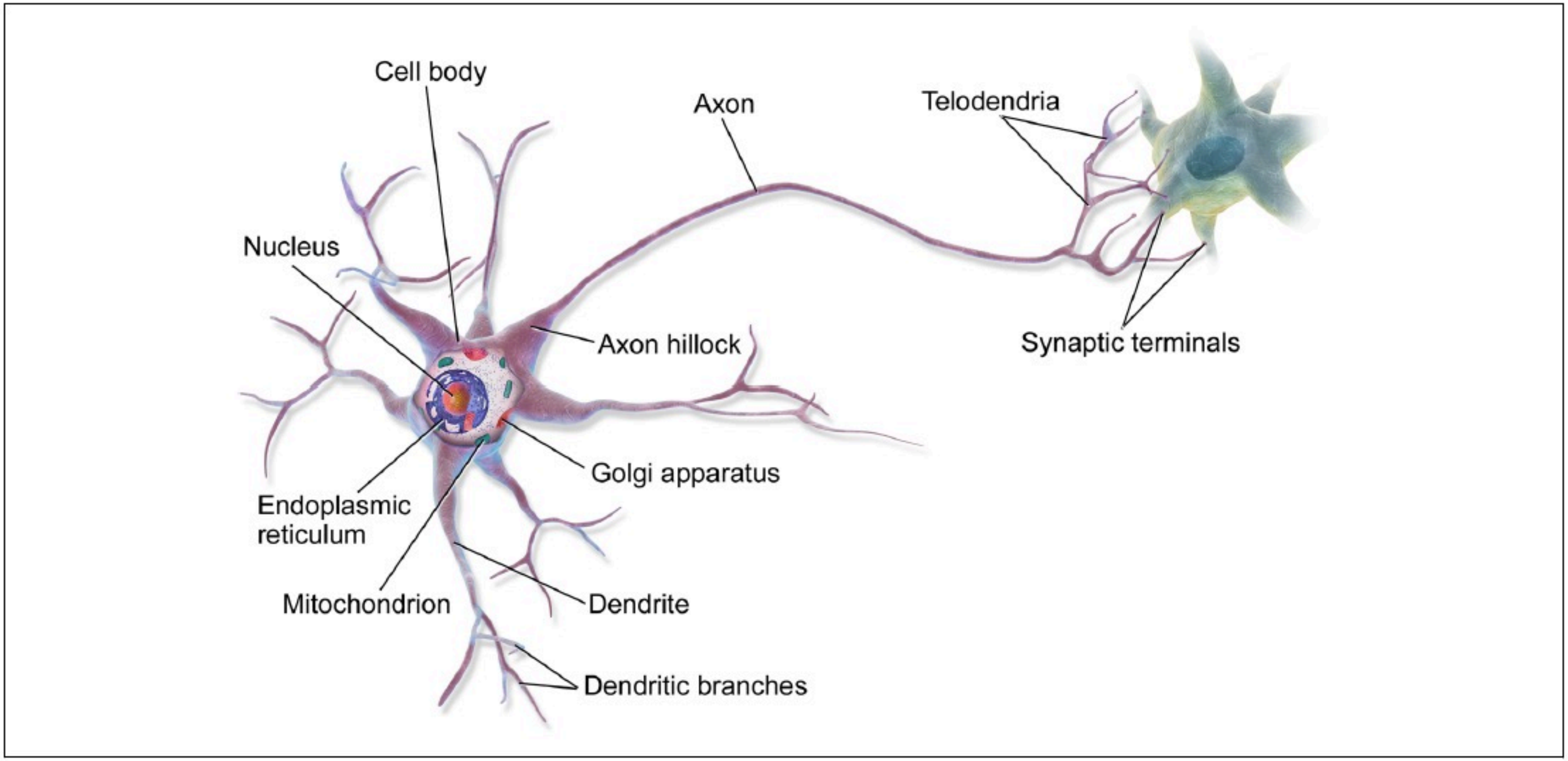
- Complex ML tasks
  - classifying billions of images (e.g., Google Images)
  - powering speech recognition services (e.g., Apple's Siri),
  - recommending the best videos to watch to hundreds of millions of users every day (e.g., YouTube)
  - learning to beat the world champion at the game of Go by playing millions of games against itself (DeepMind's Alpha-Zero)

Pattern Recognition and Machine Learning, J. C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Biological neuron

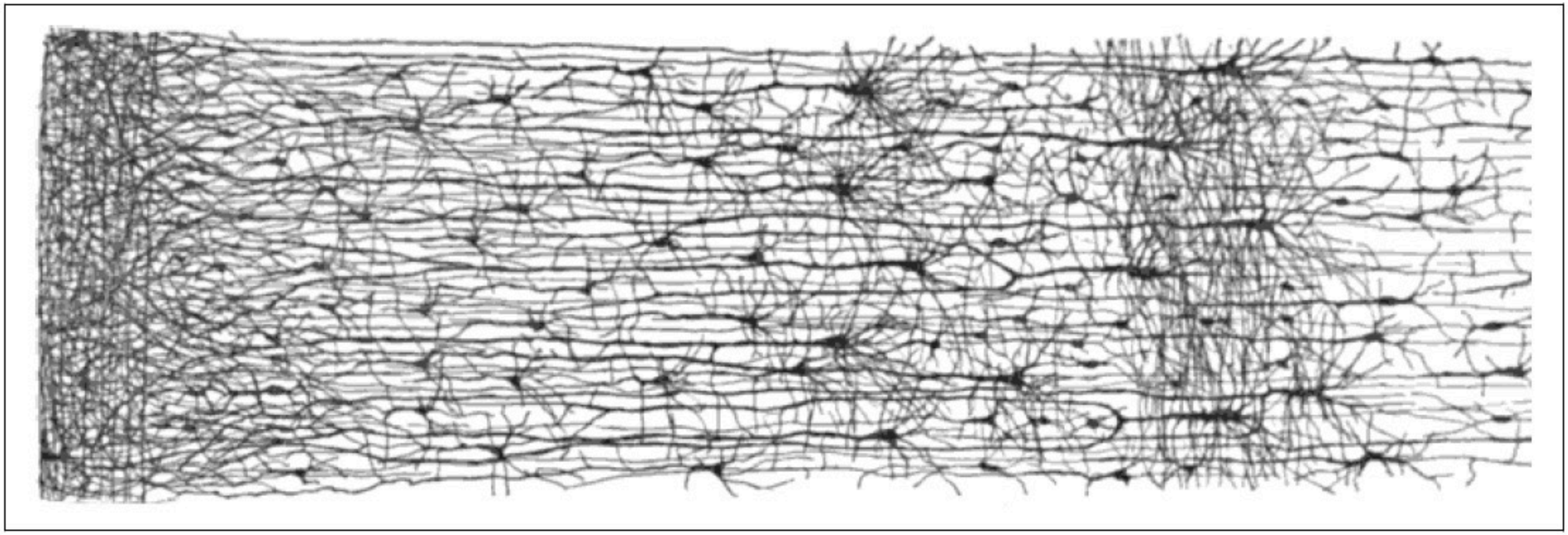
- Biological neurons
  - behave in a rather simple way
  - are organized in a vast network of billions of neurons
  - each neuron typically connected to thousands of other neurons
- Neural networks
  - Highly complex computations can be performed by a vast network of fairly simple neurons

Biological neuron



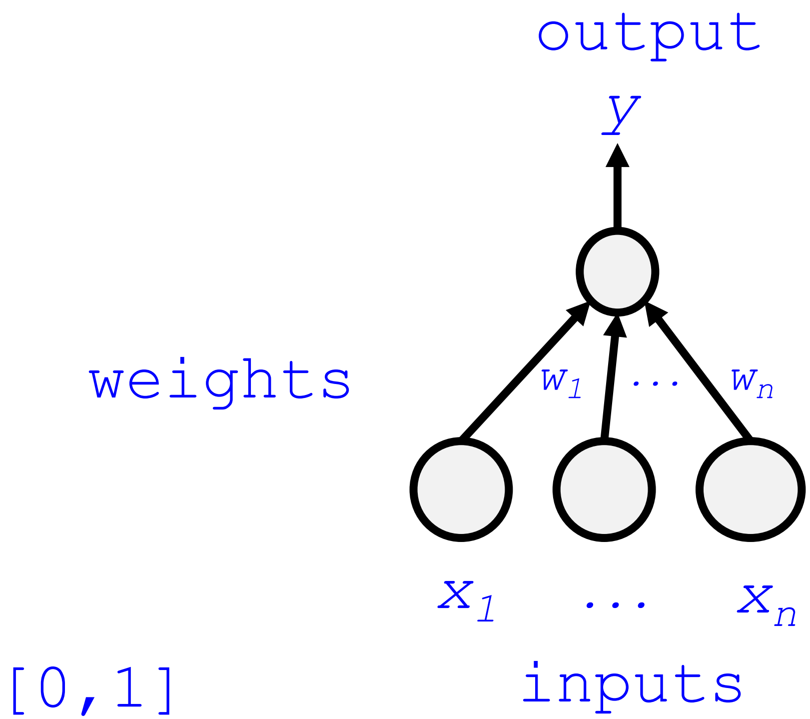
Biological Neuron

Biological Neural Network



Multiple layers in a biological neural network (human cortex)

Artificial neuron



$$\mathbf{w} = \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}$$

Artificial neuron

- sum

$$z = \sum_{i=1}^{n} w_i \, x_i = \mathbf{w}^{\mathrm{T}} \mathbf{x}$$

- output

$$y = f(\mathbf{w}, \mathbf{x}) = \theta(z)$$

- activation functions

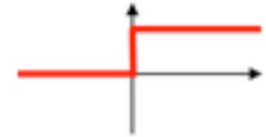
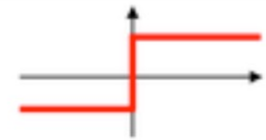
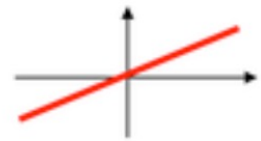
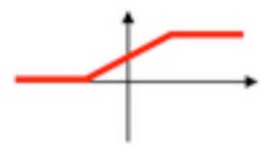
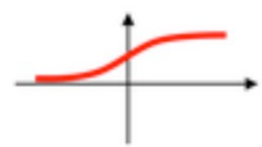
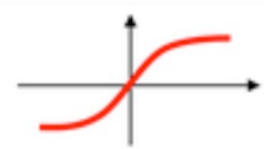
$$\theta = \begin{cases} 1 & if\ z \geq 0 \\ 0 & if\ z < 0 \end{cases}$$

Heaviside

$$\theta = \begin{cases} -1\ if\ z < 0 \\ 0\ if\ z = 0 \\ +1\ if\ z > 0 \end{cases}$$

Signum

Activation functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

Linear models for regression

- Linear regression

$$y = f(\mathbf{w}, \mathbf{x}) = w_0 + w_1 x_1 + \cdots + w_n x_n$$

- Basis functions extension

$$y = w_0 + \sum_{j=1}^{n-1} w_j \, \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\Phi}(\mathbf{x})$$

$$\Phi = (\phi_0, \dots, \phi_{M-1})^T$$

Linear models for regression

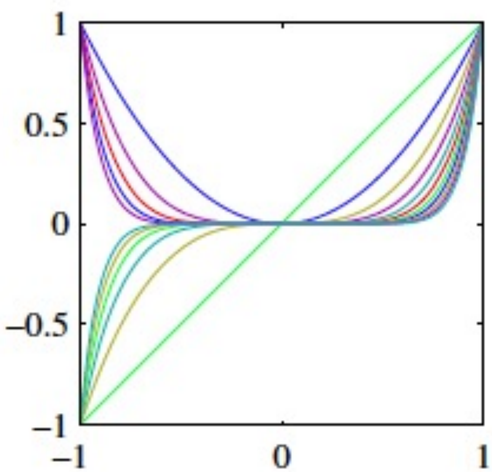
- non-linear regression basis

$$\phi_j(x) = x^j$$

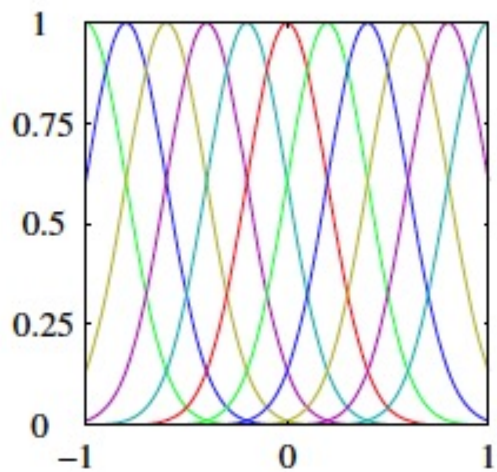
$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2\sigma^2}\right\}$$

$$\phi_j(x) = \vartheta\left(\frac{x-\mu_j}{\sigma}\right)$$

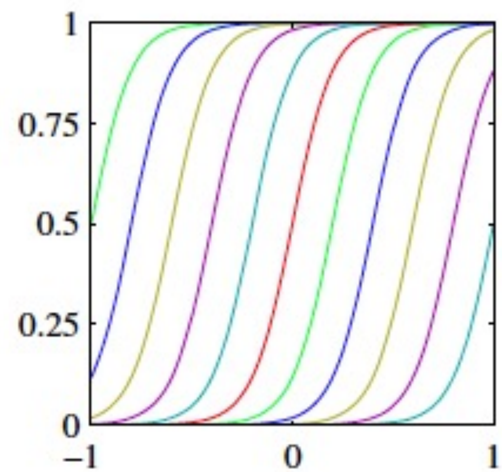
$$\vartheta(a) = \frac{1}{1+\exp(-a)}$$



Polinomial



Gaussians



Sigmoidal

Linear models for classification

- Classification

$$\mathbf{x} \rightarrow C_k \qquad\qquad k = 1, .., K$$

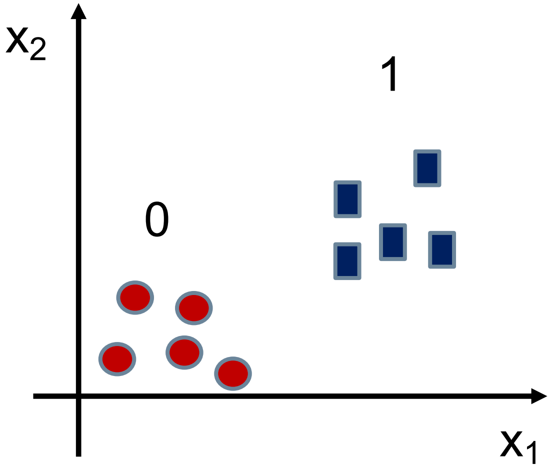
- The classes are taken to be disjoint
  - the input space is divided into decision regions whose boundaries are called decision boundaries
  - for linear models
    - (D-1)-dimensional hyperplanes within the D-dimensional input space

Target values

- K = 2 classes

$$t \in \{0,1\}$$

$$C_1 \rightarrow 1$$
$$C_2 \rightarrow 0$$

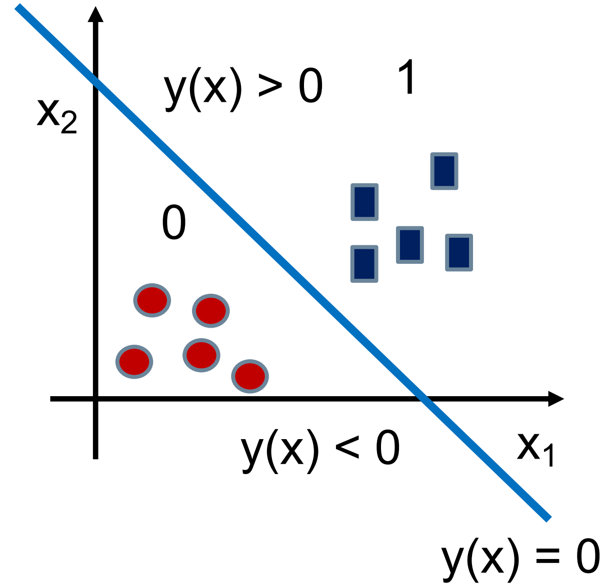


- K > 2 classes

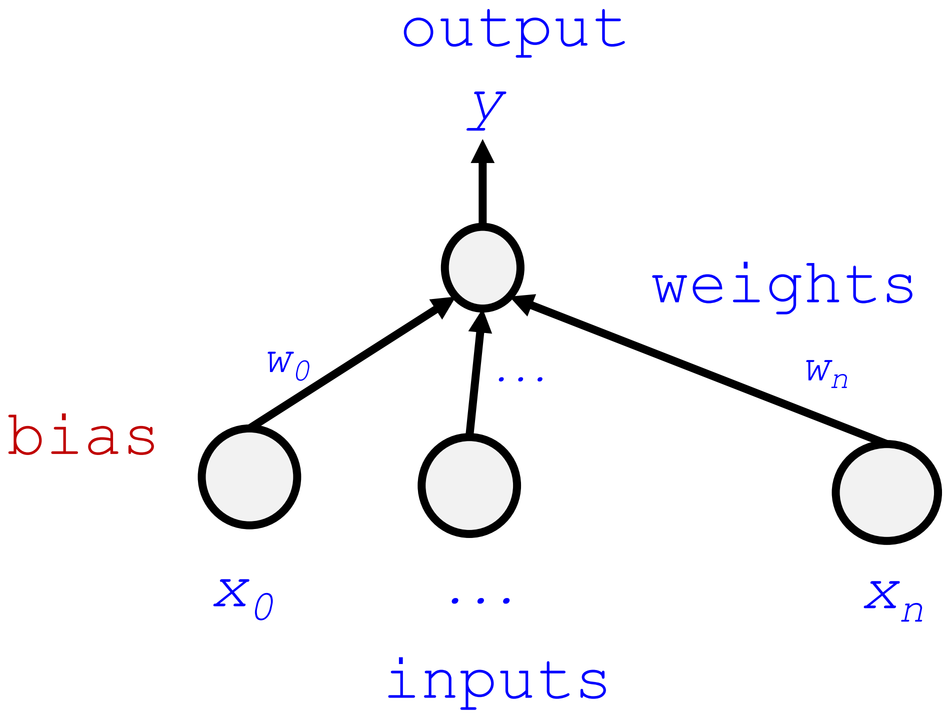
$$\mathbf{t} = (0, 1, 0, 0, 0)^T$$

1-of K coding (K = 5)

Linear discriminant function



$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x} + w_0$$

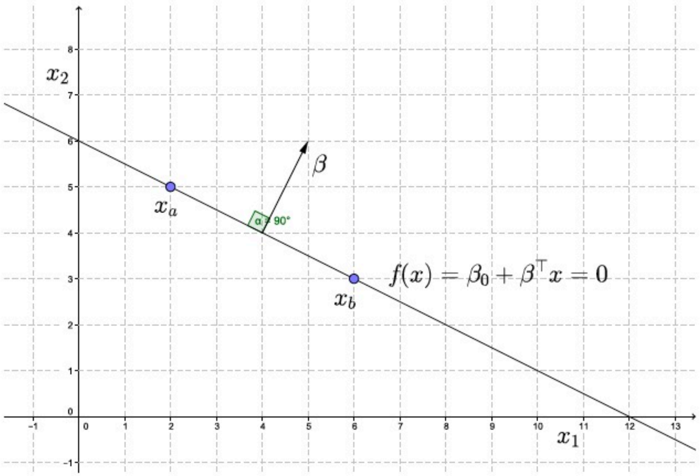


- Learning of the parameters w and w_0

Decision surface orientation

- Decision boundary

$$y(\mathbf{x}, \mathbf{w}) = 0$$



- Two points

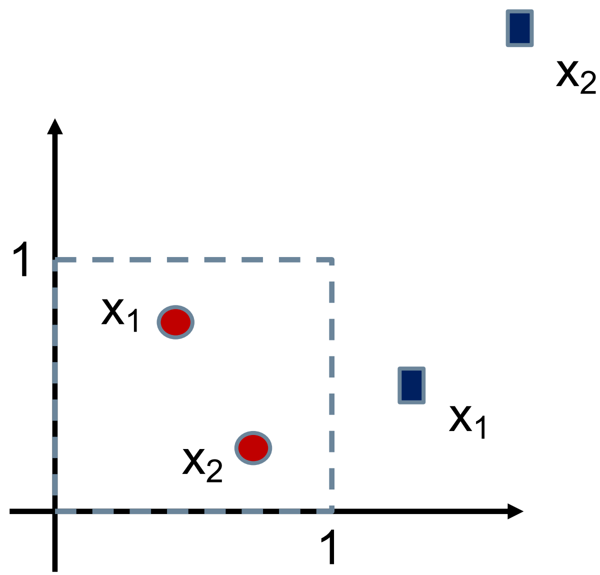
$$y(\mathbf{x}_\mathrm{A}, \mathbf{w}) = 0$$
$$y(\mathbf{x}_\mathrm{B}, \mathbf{w}) = 0$$

$$\Rightarrow \quad \mathbf{w}^T(\mathbf{x}_\mathrm{A} - \mathbf{x}_\mathrm{B}) = 0$$

w is orthogonal to every vector within the decision surface determining the orientation of the decision surface

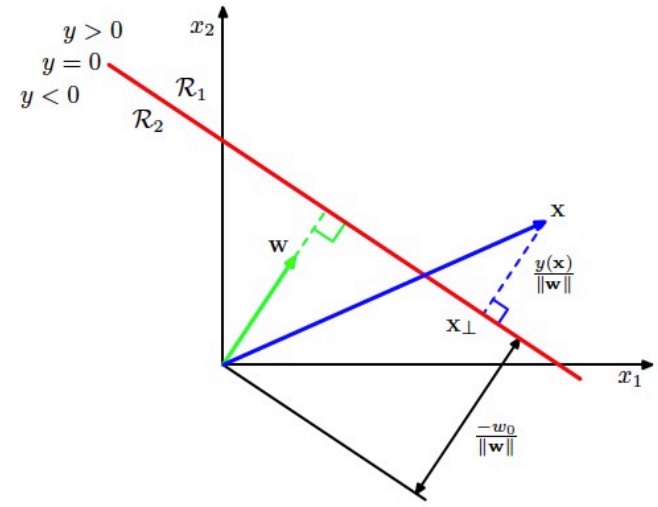
Decision surface distance

- normalization



$$\tilde{x}_i = \frac{x_i}{\|\mathbf{x}\|} = \frac{x_i}{\sqrt{\mathbf{x}\mathbf{x}^T}} = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}}$$

Decision surface distance



- point $\mathbf{x}$ on

$$y(\mathbf{x},\mathbf{w}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$$

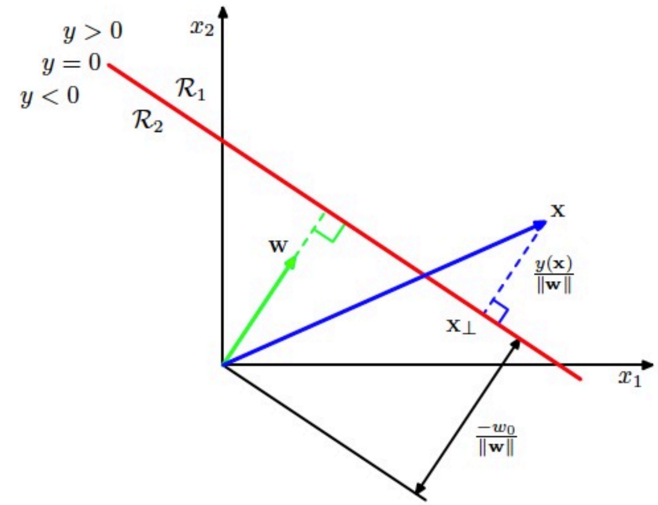
$$\frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$ normal distance from the origin to decision surface

norm

$$\|\mathbf{w}\| = \sqrt{\mathbf{w}\mathbf{w}^T} = \sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$$

w_0 determines the location of the decision surface

DS point distance



- arbitrary point **x**

orthogonal projection onto the decison surface

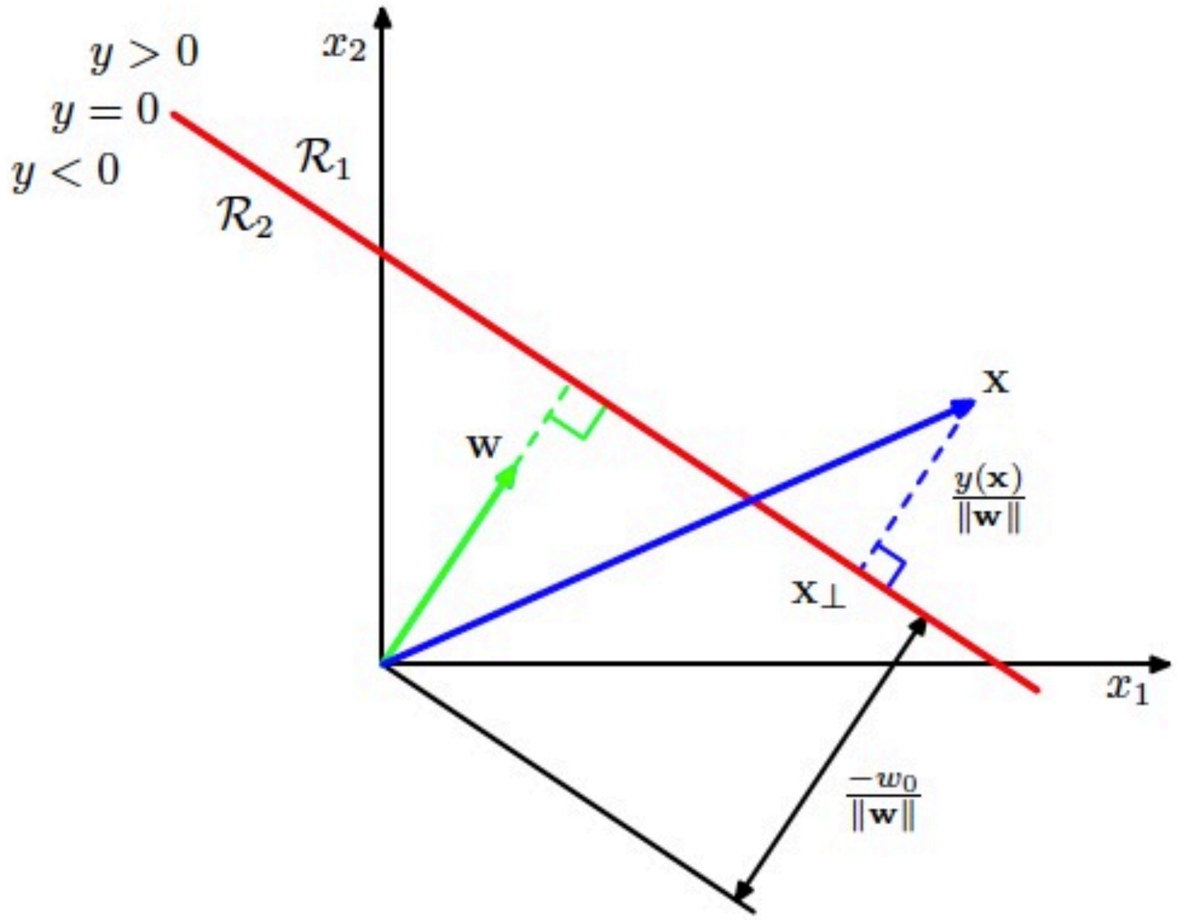
$$\mathbf{x} = \mathbf{x}_{\perp} + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

Multiplying both sides by $\mathbf{w}^T$ and adding w_0

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|}$$

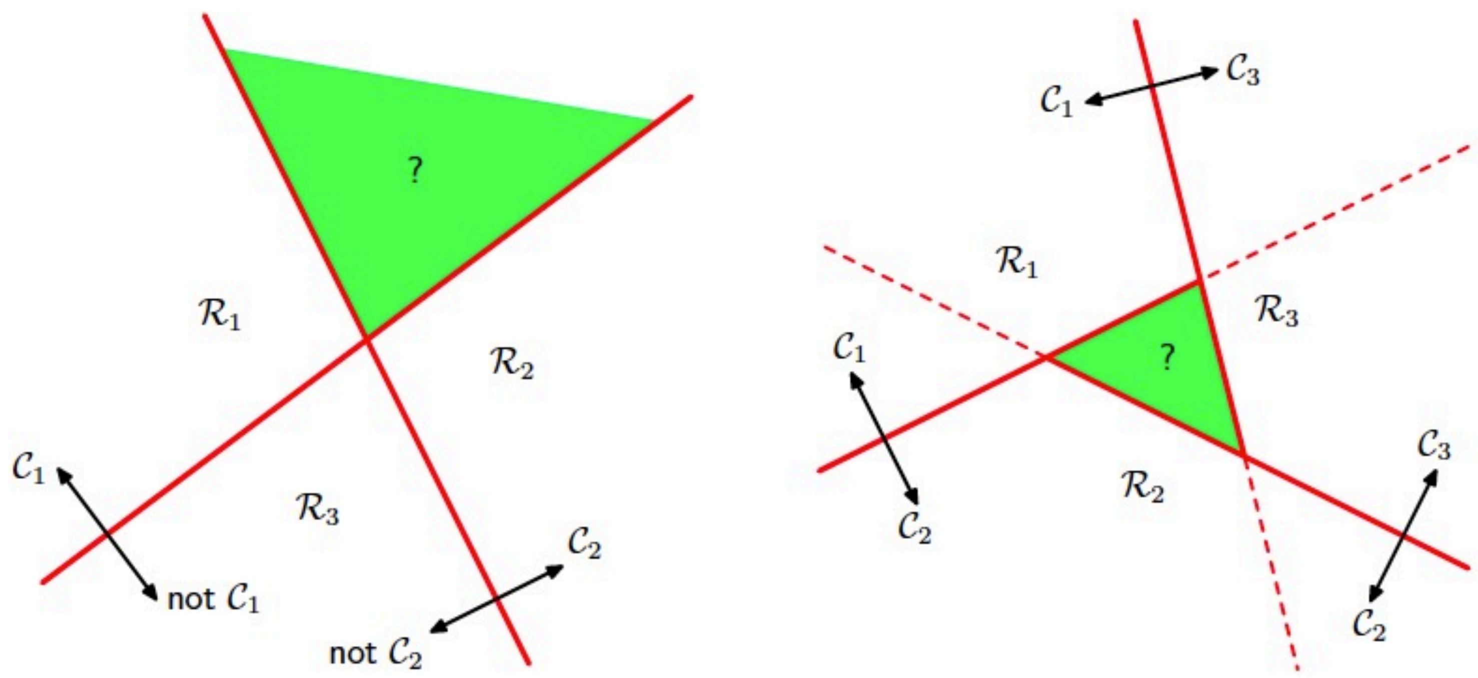
r perpendicular distance of the point **x** from the decision surface

Linear discriminant function



Multiple classes

- Approaches
  - one-versus-the rest
  - one-versus-one

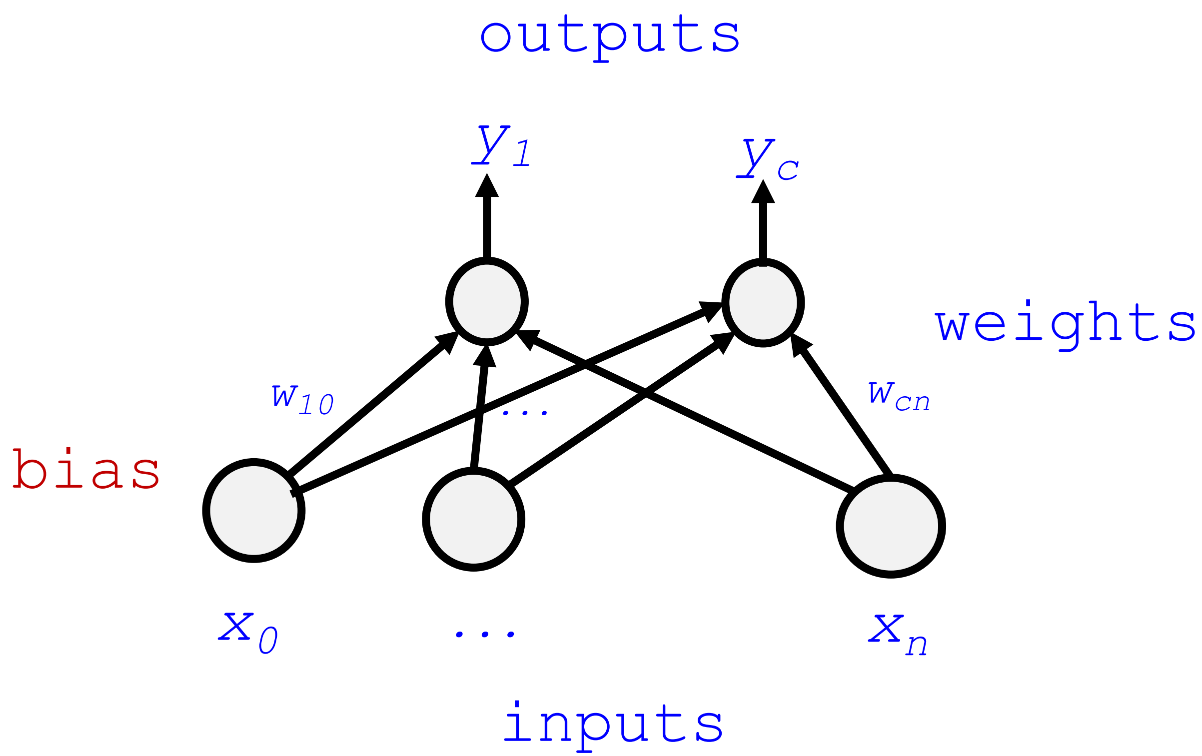


Multiple linear discriminant

- K > 2 classes

$$y_k(\mathbf{x}, \mathbf{w}_k) = \mathbf{w}_k{}^T \mathbf{x} + w_{k0}$$

$$y_k(\mathbf{x}, \mathbf{w}_k) > y_j(\mathbf{x}, \mathbf{w}_j)$$
$$for\ all\ j \neq k \ \rightarrow C_k$$



- bias is a threshold

Multiple linear discriminant

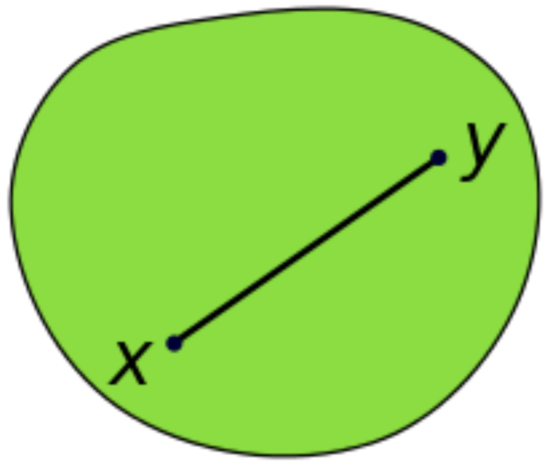
- K > 2 classes

$x_0 = 1$

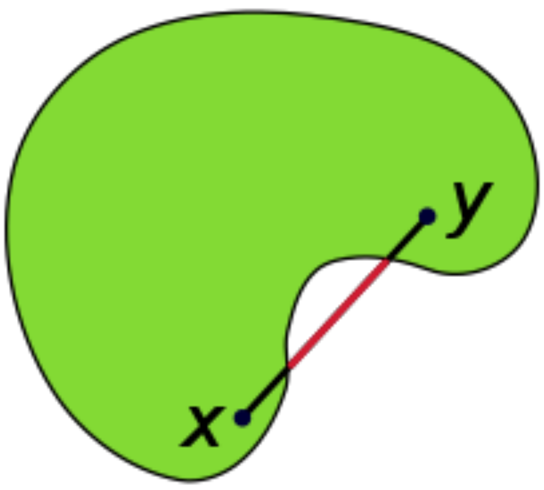
$$y_k(\mathbf{x}, \mathbf{w}_k) = \sum_{i=1}^{n} w_{ki} x_i + w_{k0} = \sum_{i=0}^{n} w_{ki} x_i$$

- Output unit has the largest activation
  - Set of decision regions which are always simply connected and convex

Multiple linear discriminant



$\mathcal{R}_k$ is convex



$\mathcal{R}_k$ is no convex

Multiple linear discriminant

- The decision regions

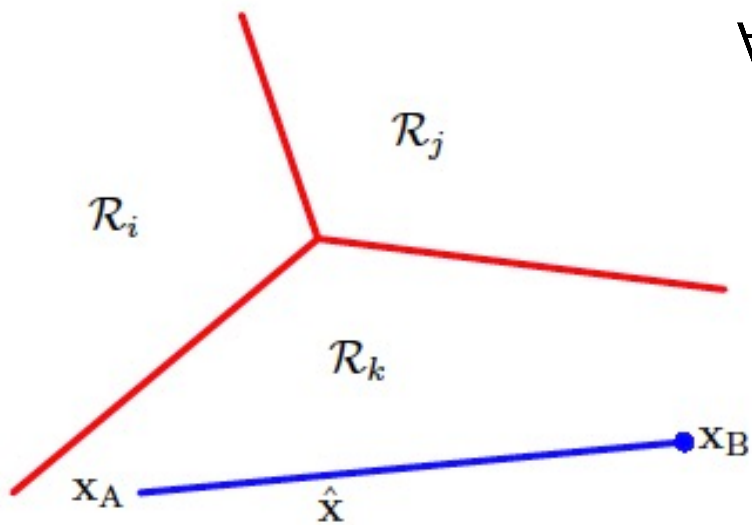
$$\hat{\mathrm{x}} = \lambda \mathbf{x}^A + (1-\lambda)\mathbf{x}^B$$

from the linearity

$$y_k(\mathbf{x}^A) > y_j(\mathbf{x}^A)$$
$$y_k(\mathbf{x}^B) > y_j(\mathbf{x}^B)$$
$$\forall\, j \neq k$$

$$y_k(\widehat{\mathrm{x}}) = \lambda y_k(\mathbf{x}^A) + (1-\lambda) y_k(\mathbf{x}^B)$$

$$y_k(\hat{\mathrm{x}}) > y_j(\hat{\mathrm{x}})$$
$$\forall\, j \neq k$$

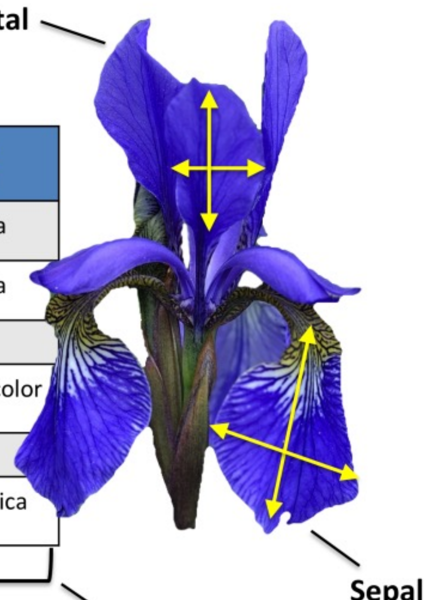


All the points on the line also line in $\mathcal{R}_k$ so the region must be simply connected and convex

Dataset - iris

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica



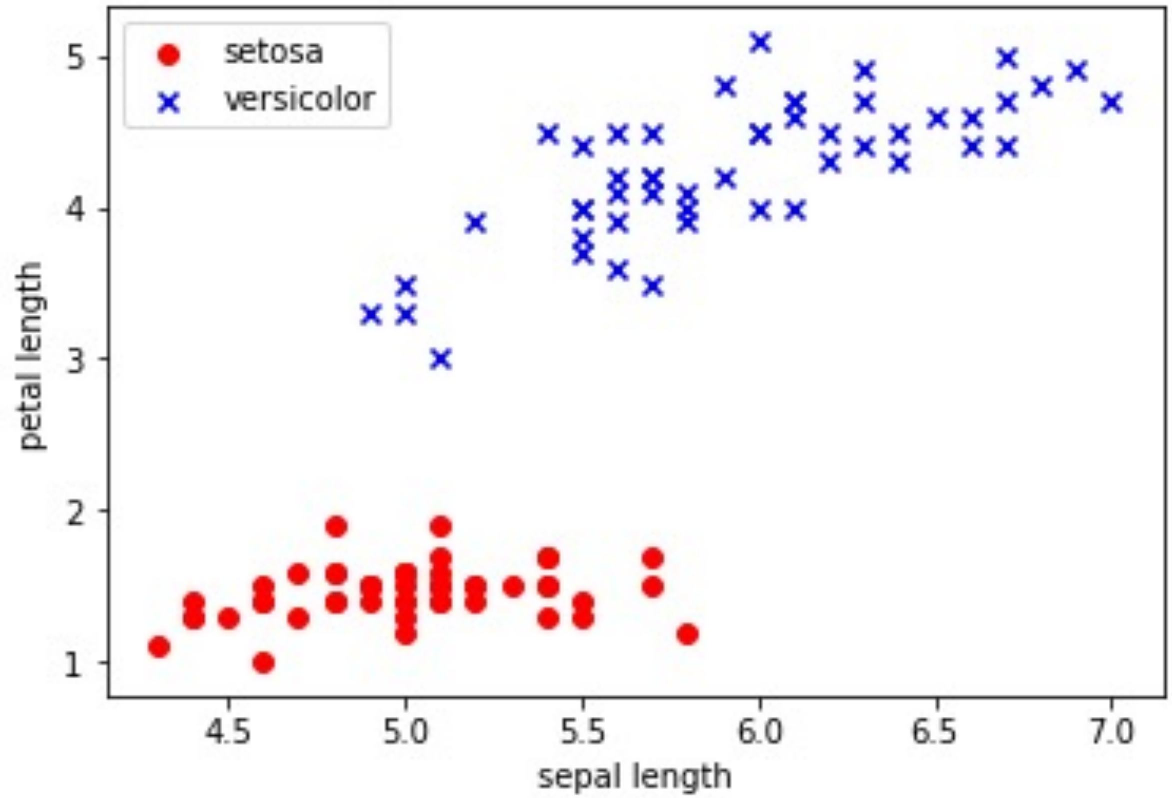
Features
(attributes, measurements, dimensions)

Class labels
(targets)

x_i

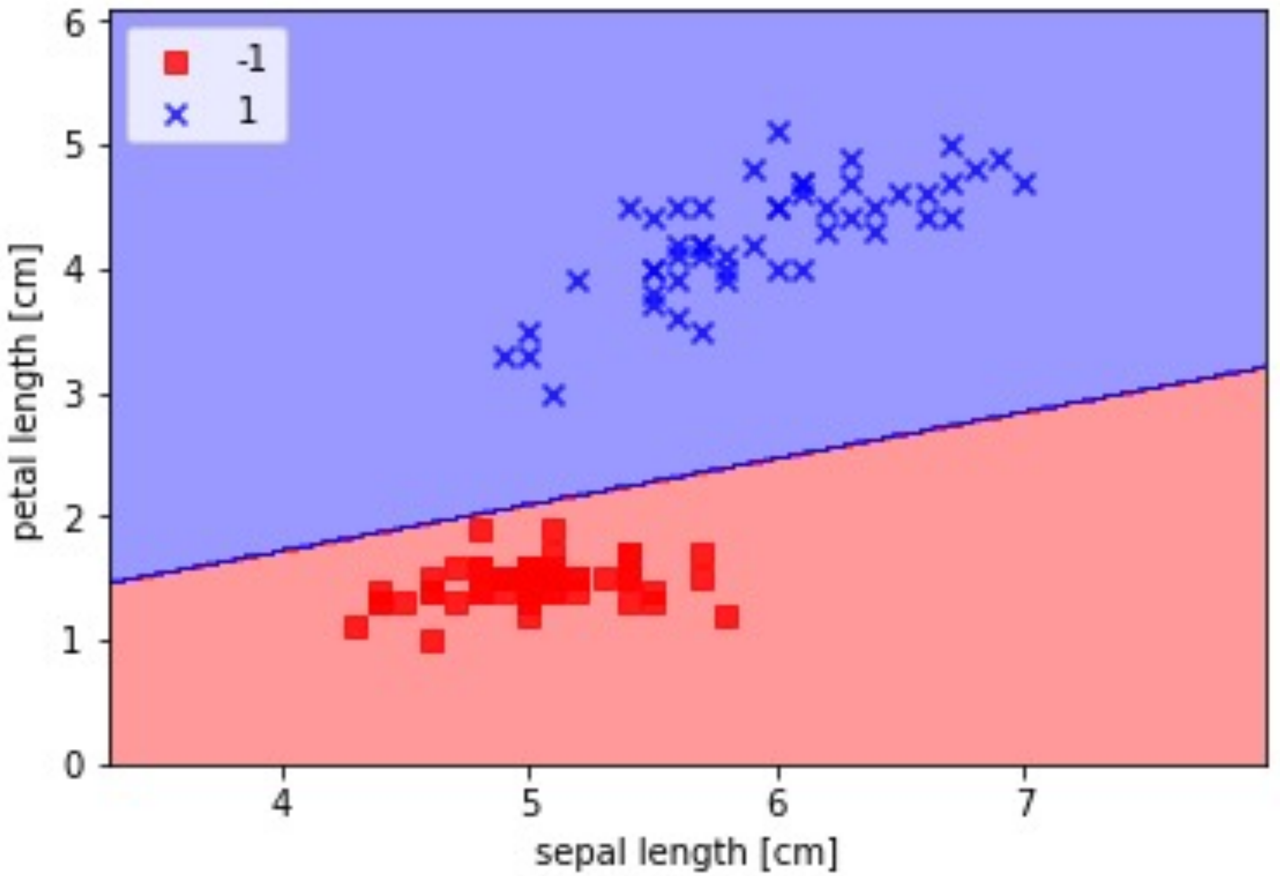
t_i

Iris visualization



2D Plot of IRIS data

Decision boudary



Decision boundary after learning