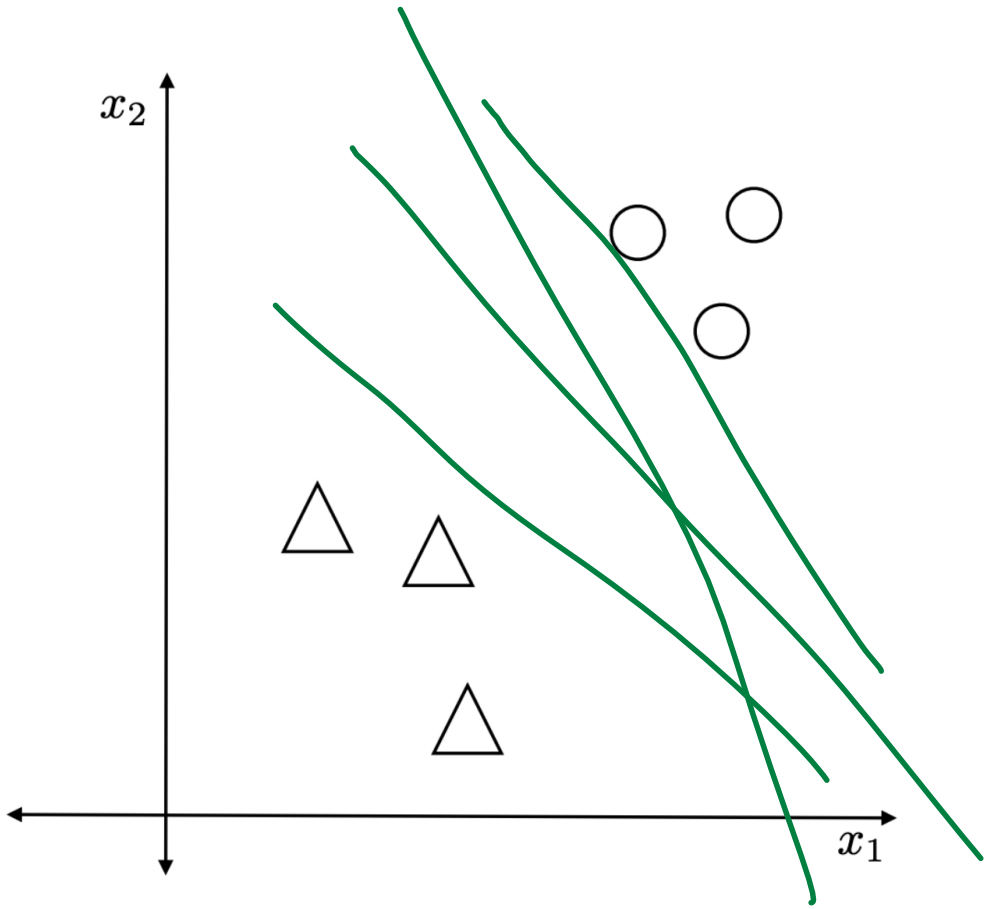
Tutorial Session 1: Linear Classification

Nora Koreuber

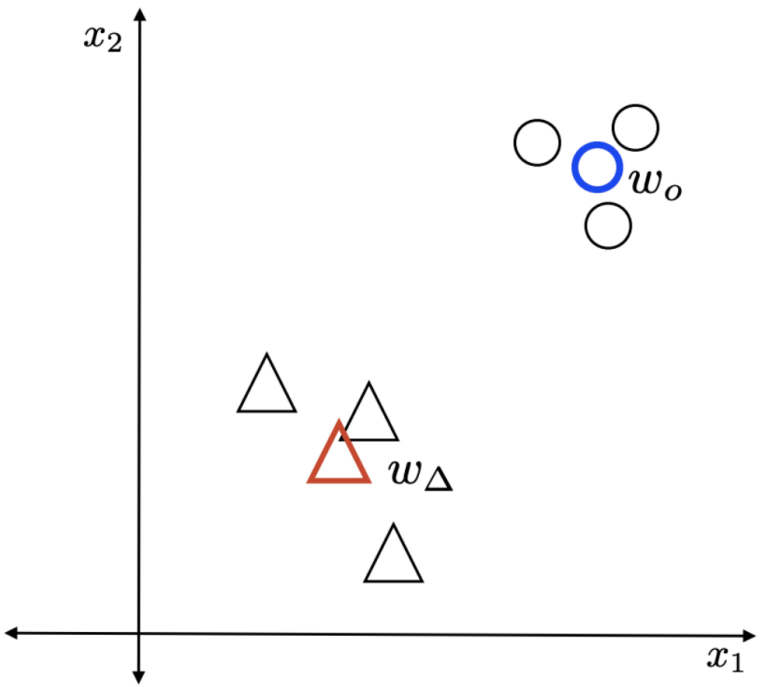
Linear Classification



possible decision boundaries

- 2 classes of data points that are linearly separable
- goal: a linear classifier learns to separate these 2 classes

Nearest Centroid Classifier



Find prototypes $\mathbf{w}_\Delta$ and $\mathbf{w}_o$ by estimating class means

$$\mathbf{w}_\Delta = \frac{1}{N_\Delta} \sum_{n=1}^{N_\Delta} \mathbf{x}_{\Delta,n}$$

$$\mathbf{w}_o = \frac{1}{N_o} \sum_{n}^{N_o} \mathbf{x}_{o,n}$$

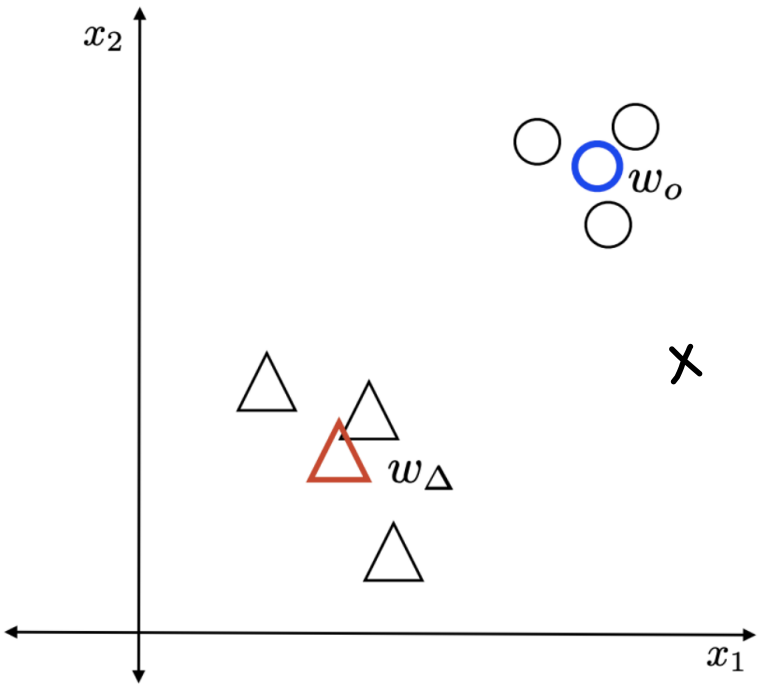
N_Δ, N_o are the number of data points

$\mathbf{x}_\Delta, \mathbf{x}_o$ are training data points

$\mathbf{w}_\Delta, \mathbf{w}_o$ are class prototypes

$\mathbf{x}_\Delta, \mathbf{x}_o, \mathbf{w}_\Delta, \mathbf{w}_o \in \mathbb{R}^D$

Nearest Centroid Classifier

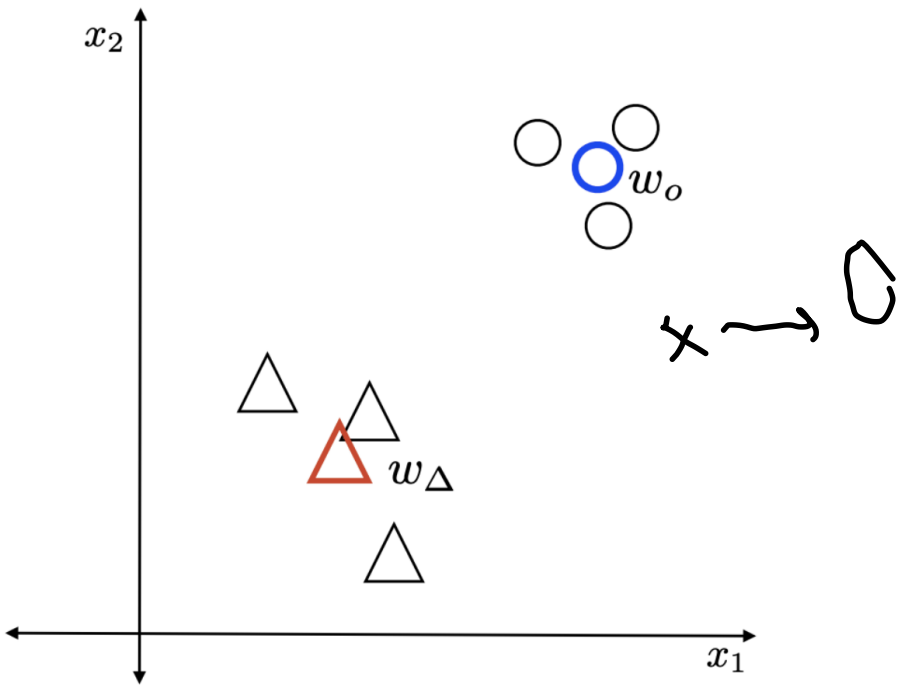


To decide class membership of a new data point $\mathbf{x} \in \mathbb{R}^D$, we can measure the distance to class prototypes $\mathbf{w}_\Delta, \mathbf{w}_o$

$$\|\mathbf{w}_\Delta - \mathbf{x}\| = \sqrt{\sum_{j=1}^{2} (\mathbf{w}_{\Delta j} - \mathbf{x}_j)^2}$$

$$\|\mathbf{w}_o - \mathbf{x}\| = \sqrt{\sum_{j=1}^{2} (\mathbf{w}_{oj} - \mathbf{x}_j)^2}$$

Nearest Centroid Classifier



Is $\mathbf{x}$ closer to $\mathbf{w}_o$?

$$\|\mathbf{w}_\Delta - \mathbf{x}\| > \|\mathbf{w}_o - \mathbf{x}\|$$

yes? $\rightarrow$ $\mathbf{x}$ belongs to o

no? $\rightarrow$ $\mathbf{x}$ belongs to Δ

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2
\end{aligned}
$$

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o)
\end{aligned}
$$

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o) \\
\mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_\Delta - \mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_o - \mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o
\end{aligned}
$$

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o) \\
\mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_\Delta - \mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_o - \mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
-2\mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> -2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o
\end{aligned}
$$

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o) \\
\mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_\Delta - \mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_o - \mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
-2\mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> -2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
0 &> 2\mathbf{w}_\Delta^\top \mathbf{x} - \mathbf{w}_\Delta^\top \mathbf{w}_\Delta - 2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o
\end{aligned}
$$

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o) \\
\mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_\Delta - \mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_o - \mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
-2\mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> -2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
0 &> 2\mathbf{w}_\Delta^\top \mathbf{x} - \mathbf{w}_\Delta^\top \mathbf{w}_\Delta - 2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
0 &> (\mathbf{w}_\Delta - \mathbf{w}_o)^\top \mathbf{x} - \frac{1}{2}(\mathbf{w}_\Delta^\top \mathbf{w}_\Delta - \mathbf{w}_o^\top \mathbf{w}_o)
\end{aligned}
$$

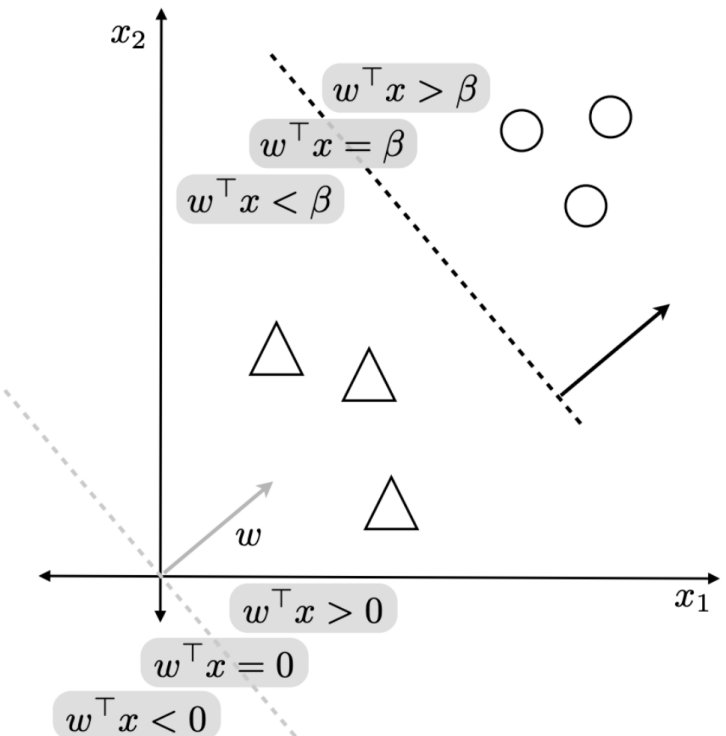
NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o) \\
\mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_\Delta - \mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{w}_o - \mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
-2\mathbf{w}_\Delta^\top \mathbf{x} + \mathbf{w}_\Delta^\top \mathbf{w}_\Delta &> -2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
0 &> 2\mathbf{w}_\Delta^\top \mathbf{x} - \mathbf{w}_\Delta^\top \mathbf{w}_\Delta - 2\mathbf{w}_o^\top \mathbf{x} + \mathbf{w}_o^\top \mathbf{w}_o \\
0 &> (\mathbf{w}_\Delta - \mathbf{w}_o)^\top \mathbf{x} - \frac{1}{2}(\mathbf{w}_\Delta^\top \mathbf{w}_\Delta - \mathbf{w}_o^\top \mathbf{w}_o) \\
0 &< \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)^\top}_{\mathbf{w}} \mathbf{x} - \underbrace{\frac{1}{2}(\mathbf{w}_o^\top \mathbf{w}_o - \mathbf{w}_\Delta^\top \mathbf{w}_\Delta)}_{\beta}
\end{aligned}
$$

NCC Discriminant Function

$$
\begin{aligned}
\text{distance}(\mathbf{x}, \mathbf{w}_\Delta) &> \text{distance}(\mathbf{x}, \mathbf{w}_o) \\
\|\mathbf{x} - \mathbf{w}_\Delta\| &> \|\mathbf{x} - \mathbf{w}_o\| \\
\|\mathbf{x} - \mathbf{w}_\Delta\|^2 &> \|\mathbf{x} - \mathbf{w}_o\|^2 \\
(\mathbf{x} - \mathbf{w}_\Delta)^\top (\mathbf{x} - \mathbf{w}_\Delta) &> (\mathbf{x} - \mathbf{w}_o)^\top (\mathbf{x} - \mathbf{w}_o) \\
\mathbf{x}^\top\mathbf{x} - \mathbf{x}^\top\mathbf{w}_\Delta - \mathbf{w}_\Delta^\top\mathbf{x} + \mathbf{w}_\Delta^\top\mathbf{w}_\Delta &> \mathbf{x}^\top\mathbf{x} - \mathbf{x}^\top\mathbf{w}_o - \mathbf{w}_o^\top\mathbf{x} + \mathbf{w}_o^\top\mathbf{w}_o \\
-2\mathbf{w}_\Delta^\top\mathbf{x} + \mathbf{w}_\Delta^\top\mathbf{w}_\Delta &> -2\mathbf{w}_o^\top\mathbf{x} + \mathbf{w}_o^\top\mathbf{w}_o \\
0 &> 2\mathbf{w}_\Delta^\top\mathbf{x} - \mathbf{w}_\Delta^\top\mathbf{w}_\Delta - 2\mathbf{w}_o^\top\mathbf{x} + \mathbf{w}_o^\top\mathbf{w}_o \\
0 &> (\mathbf{w}_\Delta - \mathbf{w}_o)^\top\mathbf{x} - \frac{1}{2}(\mathbf{w}_\Delta^\top\mathbf{w}_\Delta - \mathbf{w}_o^\top\mathbf{w}_o) \\
0 &< \underbrace{(\mathbf{w}_o - \mathbf{w}_\Delta)^\top}_{\mathbf{w}}\mathbf{x} - \underbrace{\frac{1}{2}(\mathbf{w}_o^\top\mathbf{w}_o - \mathbf{w}_\Delta^\top\mathbf{w}_\Delta)}_{\beta} \\
y(x) &= \mathbf{w}^\top\mathbf{x} - \beta
\end{aligned}
$$

Interpretation: The Decision Boundary



$\mathbf{x}, \mathbf{w} \in \mathbb{R}^D, \beta \in \mathbb{R}$

$$\mathbf{w}^\top \mathbf{x} - \beta = \begin{cases} > 0 & \text{if } \mathbf{x} \text{ belongs to o} \\ < 0 & \text{if } \mathbf{x} \text{ belongs to } \Delta \end{cases}$$

Points on the decision boundary satisfy $y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - \beta = 0$

Task 1 - Example Prototype Classifier

Consider the following data points:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$\mathbf{x}_1$ and $\mathbf{x}_2$ belong to class -1, while $\mathbf{x}_3$ and $\mathbf{x}_4$ belong to class $+1$.

1. Compute the class means $\mathbf{w}_{-1}$ and $\mathbf{w}_{+1}$.

$$w_{-1} = \frac{1}{2}\left(\begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} 1 \\ 4 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$w_{+1} = \frac{1}{2}\left(\begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Task 1 - Example Prototype Classifier

Consider the following data points:

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$\mathbf{x}_1$ and $\mathbf{x}_2$ belong to class -1, while $\mathbf{x}_3$ and $\mathbf{x}_4$ belong to class $+1$.

1. Compute the class means $\mathbf{w}_{-1}$ and $\mathbf{w}_{+1}$.
2. Compute the classification boundary $\mathbf{w}^\top \mathbf{x} - \beta = 0$ of the prototype classifier.

$$w = \begin{bmatrix} 3 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \beta = \frac{1}{2}\left(\begin{bmatrix} 3 & 2 \end{bmatrix}\begin{bmatrix} 3 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

$$= 5.5$$

Task 1 - Example Prototype Classifier

3. For each point, compute the assigned class label $\text{sign}(\mathbf{w}^\top \mathbf{x} - \beta)$. Are all points classified correctly?

$s\left(\begin{bmatrix}2 & 1\end{bmatrix}\begin{bmatrix}1\\-2\end{bmatrix} - 5.5\right) = -1$ ✓

$s\left(\begin{bmatrix}2 & 1\end{bmatrix}\begin{bmatrix}1\\4\end{bmatrix} - 5.5\right) = +1$ ✗

$s\left(\begin{bmatrix}2 & 1\end{bmatrix}\begin{bmatrix}4\\2\end{bmatrix} - 5.5\right) = +1$ ✓

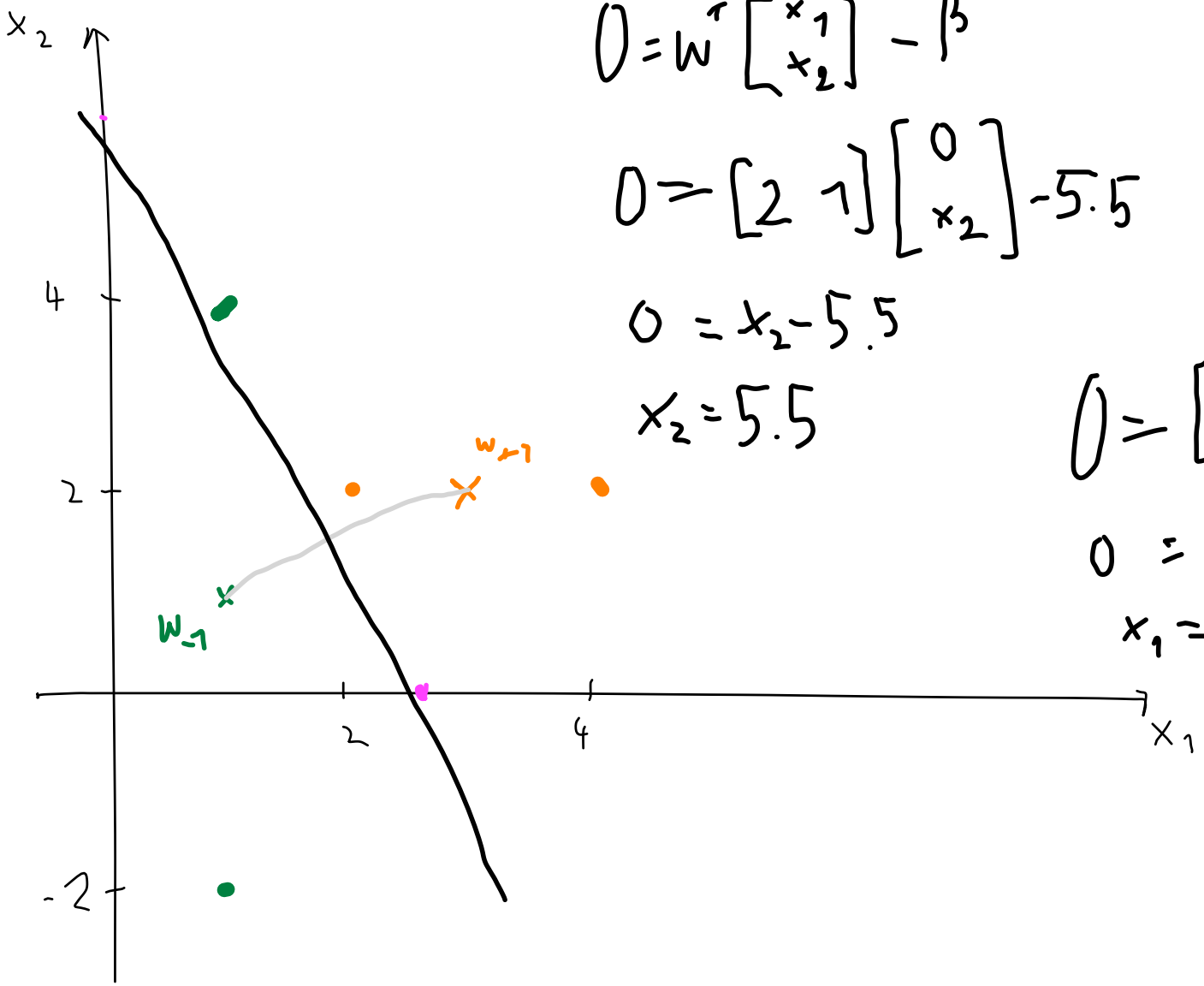
$s\left(\begin{bmatrix}2 & 1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix} - 5.5\right) = +1$ ✓

Task 1 - Example Prototype Classifier

3. For each point, compute the assigned class label $\text{sign}(\mathbf{w}^\top \mathbf{x} - \beta)$. Are all points classified correctly?
4. Sketch the data points, their class means $\mathbf{w}_{-1}$ and $\mathbf{w}_{+1}$, the normal vector $\mathbf{w}$, and the classification boundary.

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Task 1 - Example Prototype Classifier



$$0 = w^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \beta$$

$$0 = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \end{bmatrix} - 5.5$$

$$0 = x_2 - 5.5$$

$$x_2 = 5.5$$

$$0 = \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ 0 \end{bmatrix} - 5.5$$

$$0 = 2x_1 - 5.5$$

$$x_1 = 2.75$$

-1

+1

Task 2 - The linear classification boundary

Consider a linear classification boundary $\mathbf{w}^\top \mathbf{x} - \beta = 0$. Draw a sketch in 2D to visualize the classification boundary and answer the following questions:

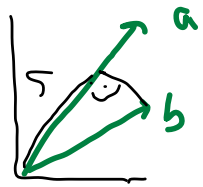
1. Suppose $\beta = 0$ and $\|\mathbf{w}\| = 1$. How large is the distance of a point $\mathbf{z}$ to the classification boundary?
2. How large is the distance of a point $\mathbf{z}$ to the classification boundary if $\|\mathbf{w}\| = 1$ but $\beta \neq 0$?
3. How large is the distance of a point $\mathbf{z}$ to the classification boundary for arbitrary β and $\mathbf{w}$?

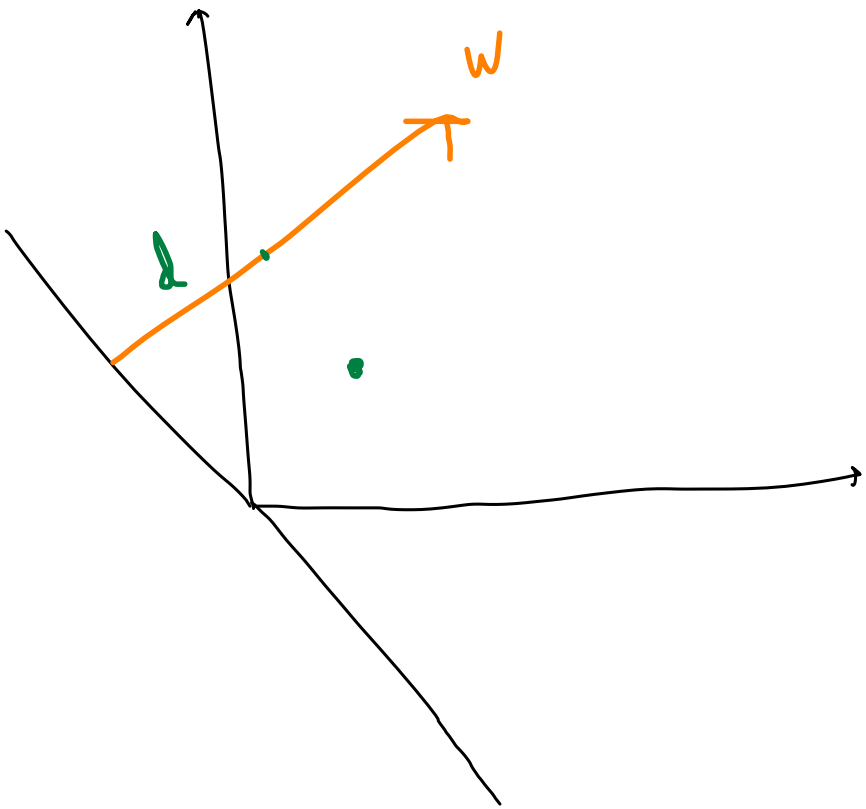
Task 2 - The linear classification boundary

1. $\beta = 0$ and $\|\mathbf{w}\| = 1$

scalar projection:

$$\|s\| = \frac{a^T b}{\|a\|}$$

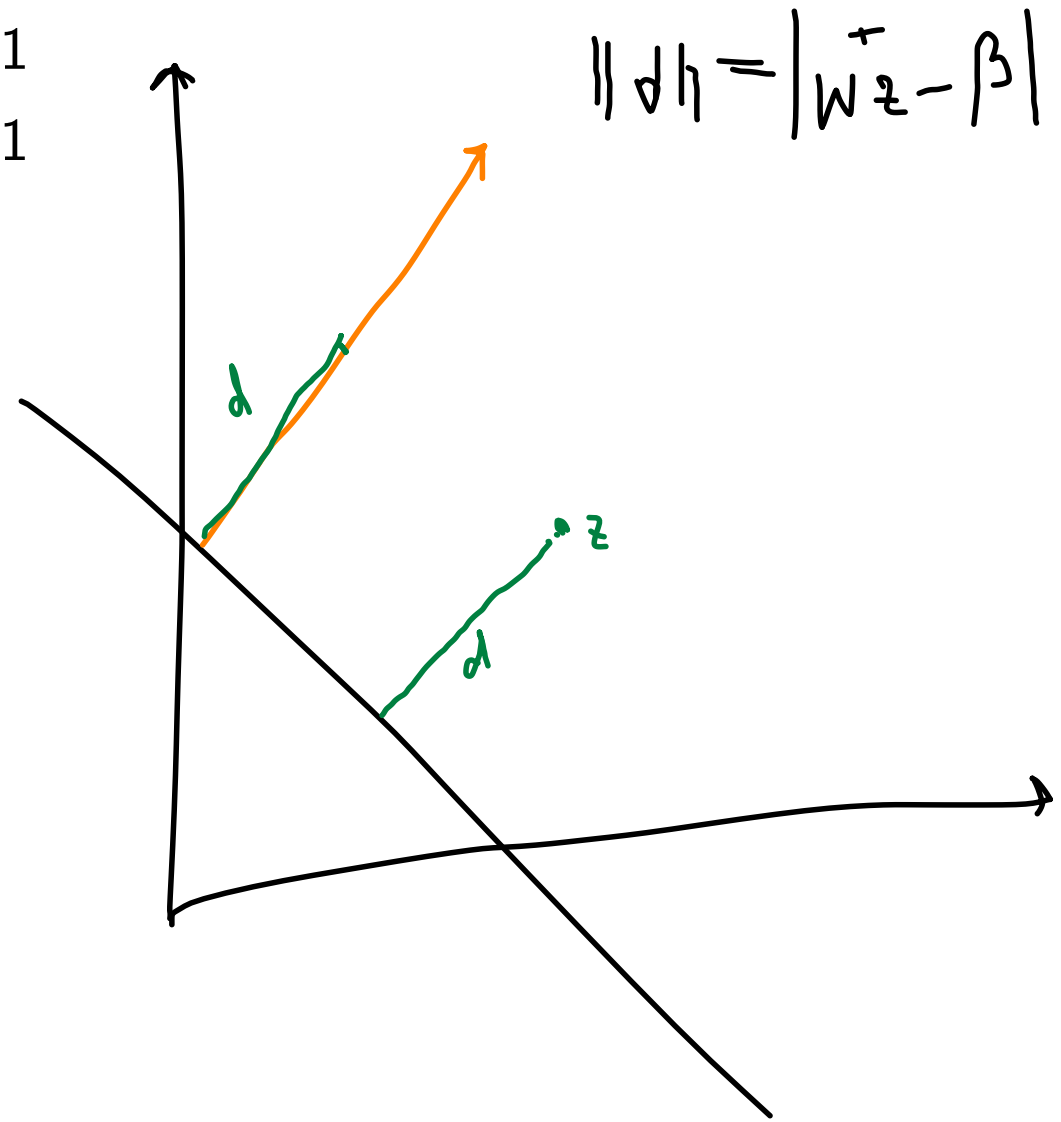




$$\|d\| = |w^T z|$$

Task 2 - The linear classification boundary

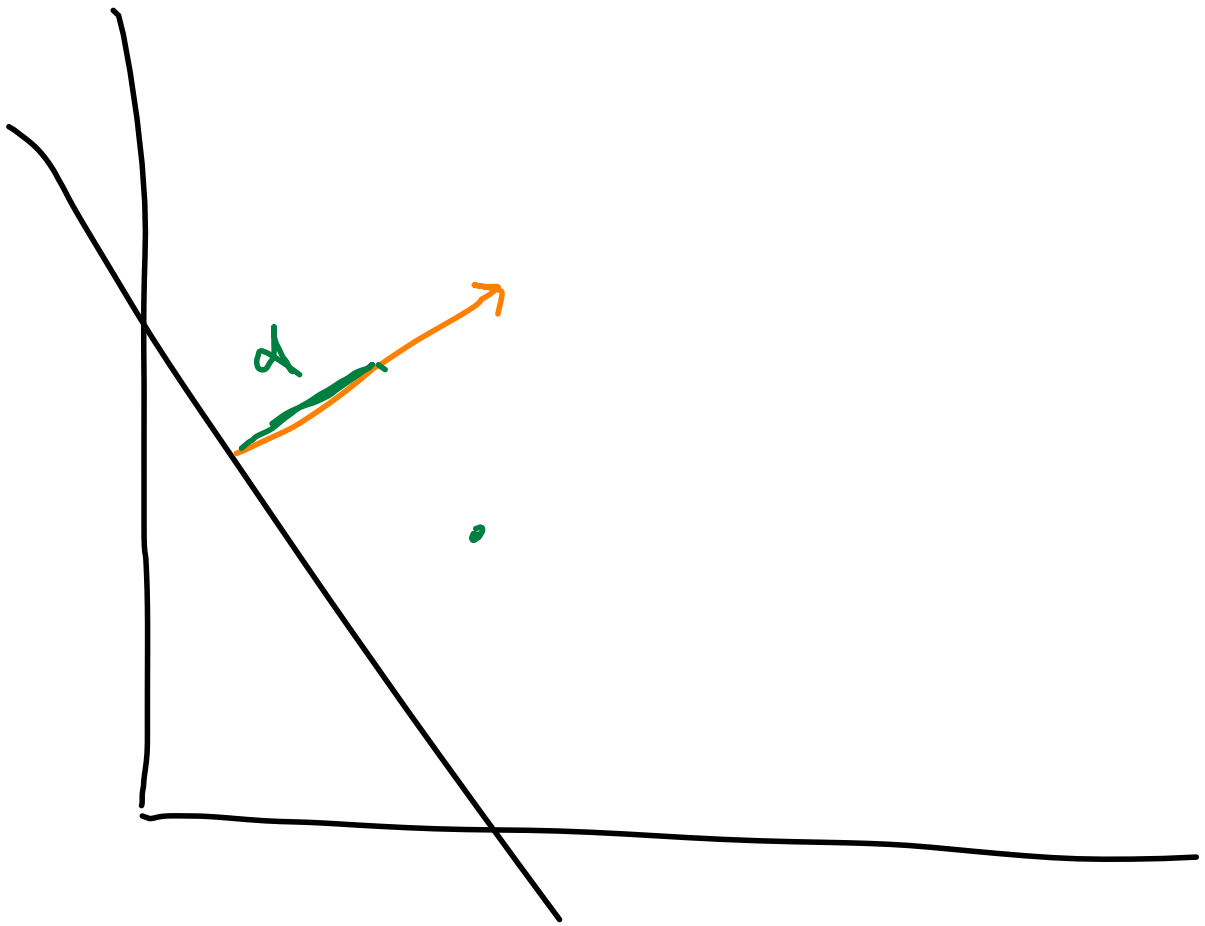
1. $\beta = 0$ and $\|\mathbf{w}\| = 1$
2. $\beta \neq 0$ and $\|\mathbf{w}\| = 1$



Task 2 - The linear classification boundary

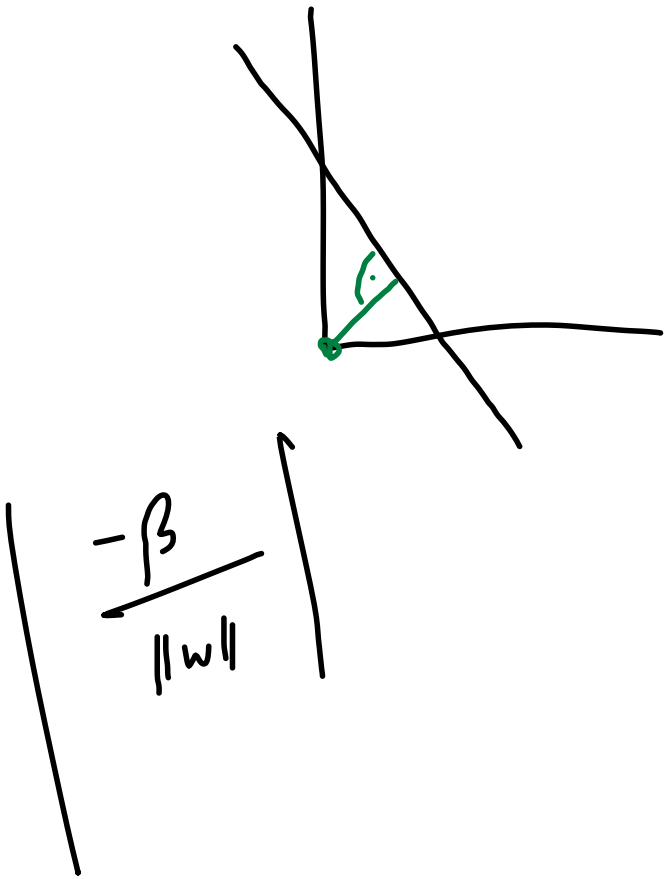
1. $\beta = 0$ and $\|\mathbf{w}\| = 1$
2. $\beta \neq 0$ and $\|\mathbf{w}\| = 1$
3. arbitrary β and $\mathbf{w}$

$$\|d\| = \frac{|w^T z - \beta|}{\|w\|}$$



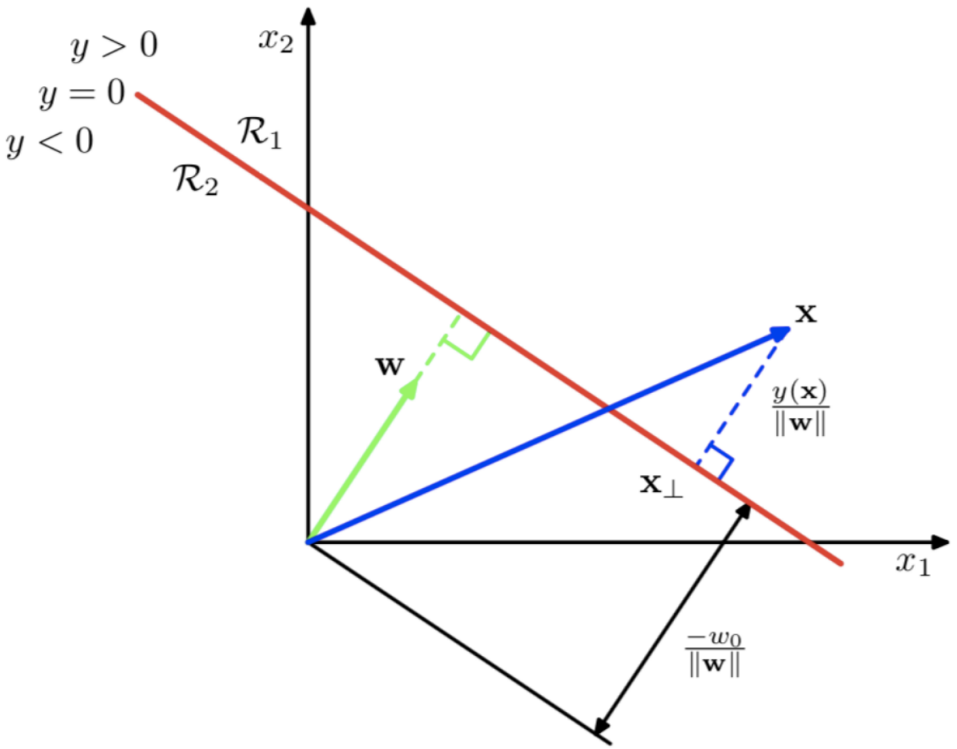
Task 2 - The linear classification boundary

1. $\beta = 0$ and $\|\mathbf{w}\| = 1$
2. $\beta \neq 0$ and $\|\mathbf{w}\| = 1$
3. arbitrary β and $\mathbf{w}$
4. distance between boundary and the origin for arbitrary β and $\mathbf{w}$



Task 2 - The linear classification boundary

Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.



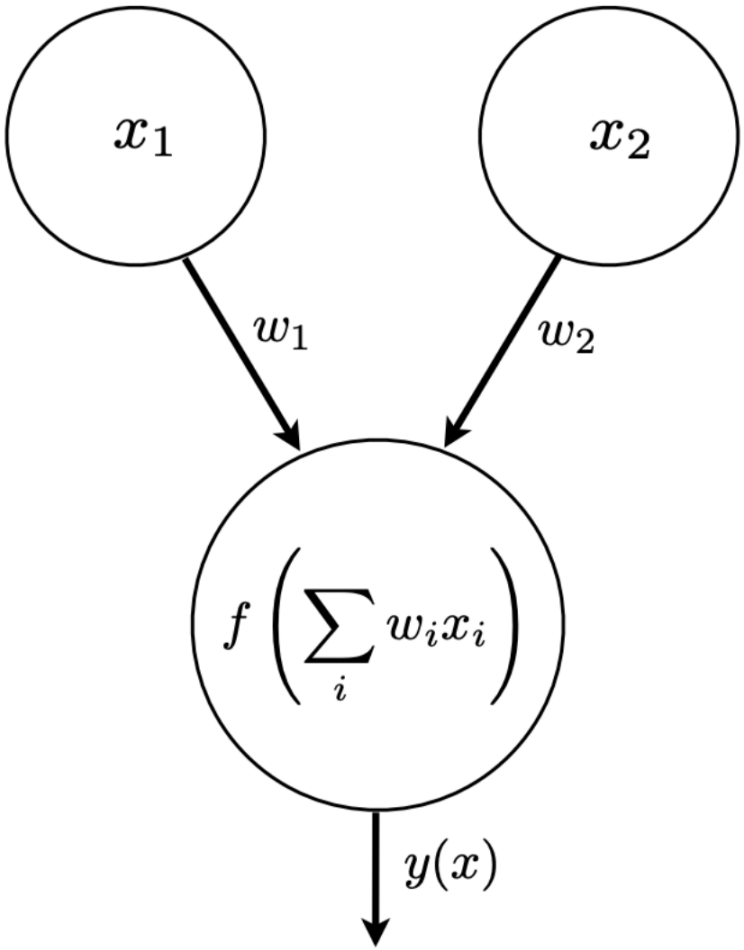
(Bishop, 2006, p. 182)

Perceptron

Figure 4.8 Illustration of the Mark 1 perceptron hardware. The photograph on the left shows how the inputs were obtained using a simple camera system in which an input scene, in this case a printed character, was illuminated by powerful lights, and an image focussed onto a 20×20 array of cadmium sulphide photocells, giving a primitive 400 pixel image. The perceptron also had a patch board, shown in the middle photograph, which allowed different configurations of input features to be tried. Often these were wired up at random to demonstrate the ability of the perceptron to learn without the need for precise wiring, in contrast to a modern digital computer. The photograph on the right shows one of the racks of adaptive weights. Each weight was implemented using a rotary variable resistor, also called a potentiometer, driven by an electric motor thereby allowing the value of the weight to be adjusted automatically by the learning algorithm.

(Bishop, 2006, p. 196)

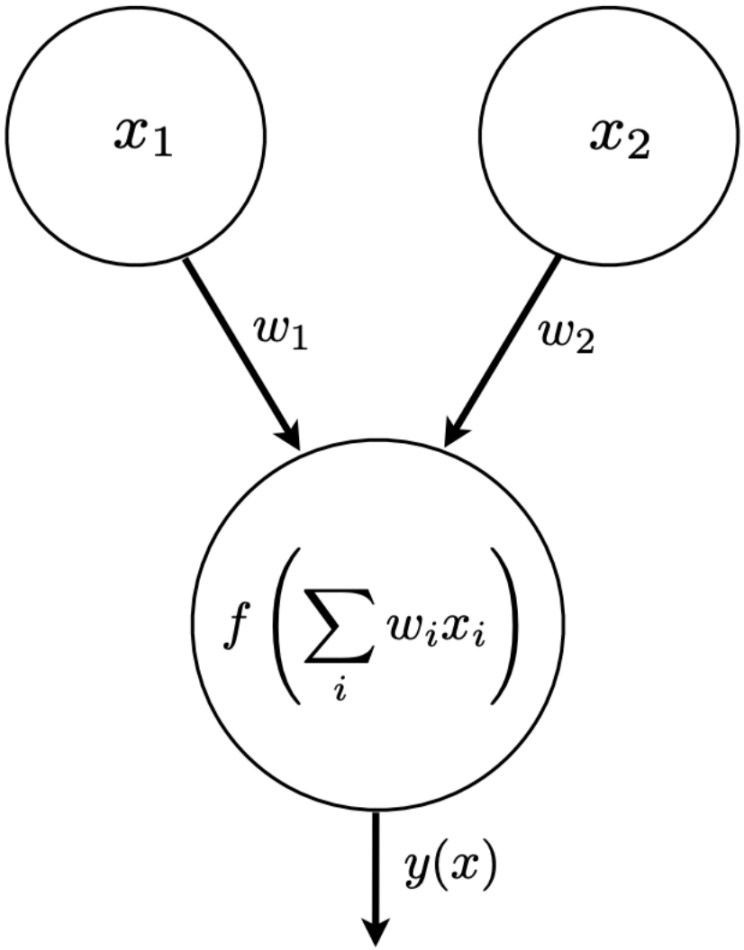
Perceptron



Input nodes $\mathbf{x}_i \in \mathbb{R}$ receive information

.

Perceptron

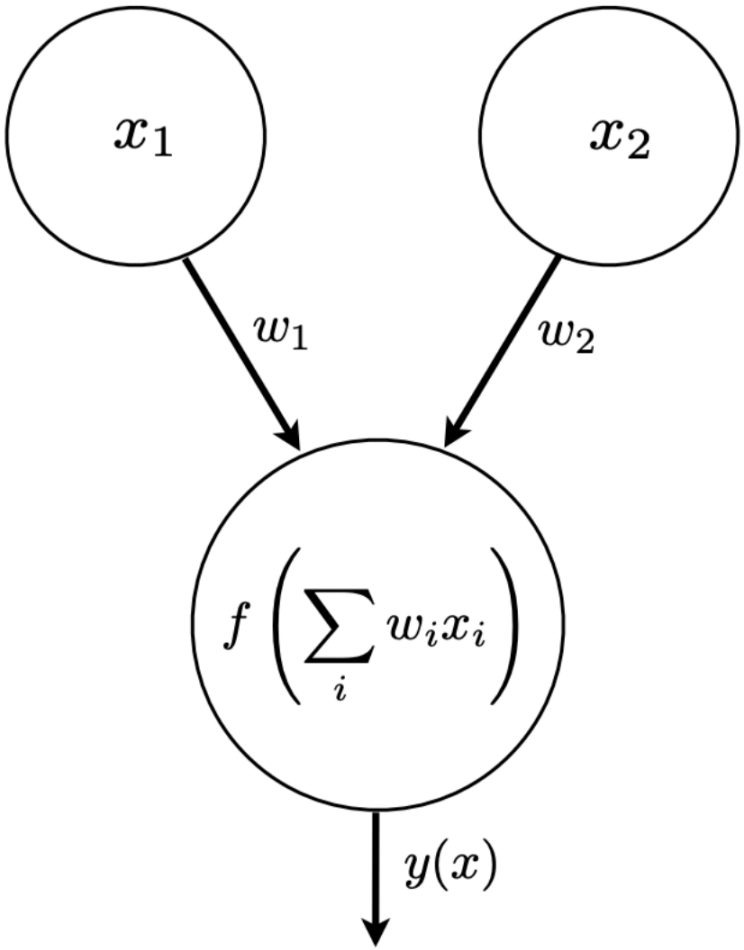


Input nodes $\mathbf{x}_i \in \mathbb{R}$ receive information

Inputs are multiplied with a weighting factor $\mathbf{w}_i \in \mathbb{R}$ and summed up:

$$a = \sum_{i=1}^{D} \mathbf{x}_i \mathbf{w}_i$$

Perceptron



Input nodes $\mathbf{x}_i \in \mathbb{R}$ receive information

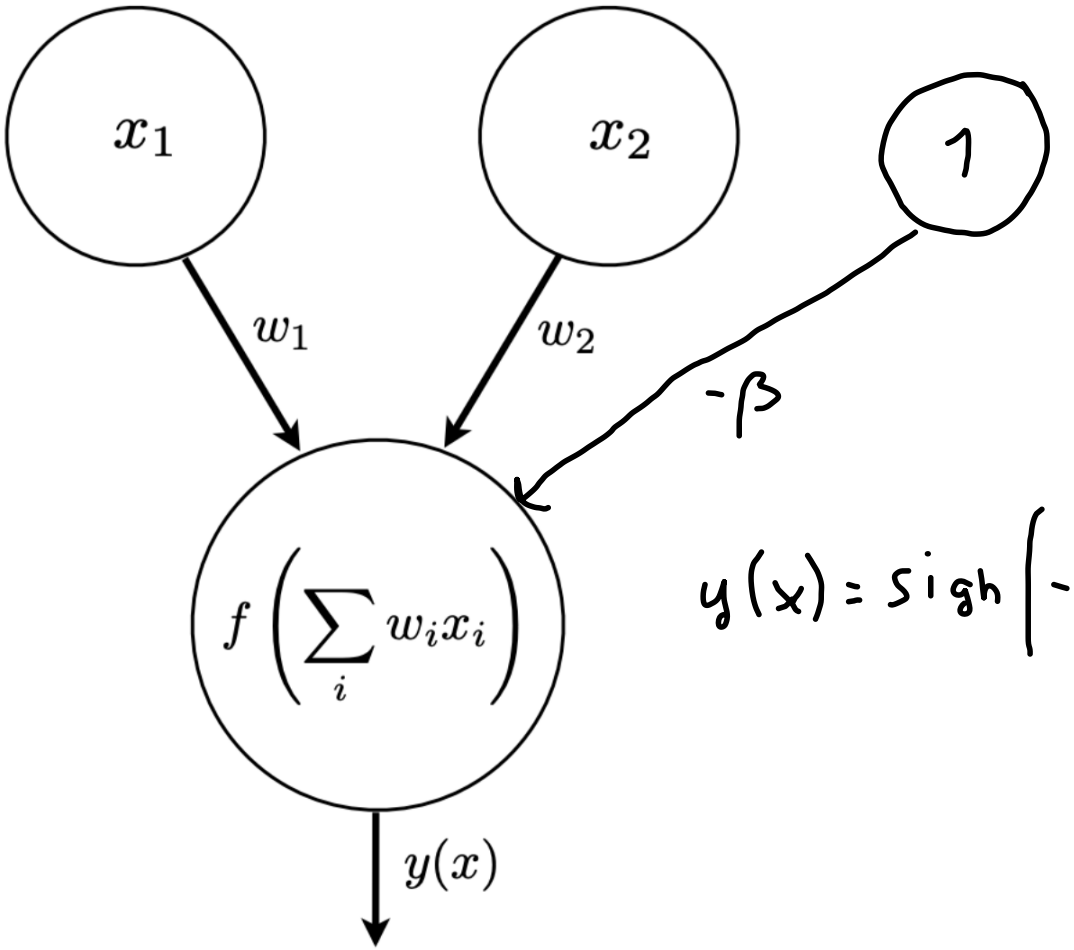
Inputs are multiplied with a weighting factor $\mathbf{w}_i \in \mathbb{R}$ and summed up:

$$a = \sum_{i=1}^{D} \mathbf{x}_i \mathbf{w}_i$$

The sum is mapped through a non-linear function $f(\cdot)$, i.e. sign-function

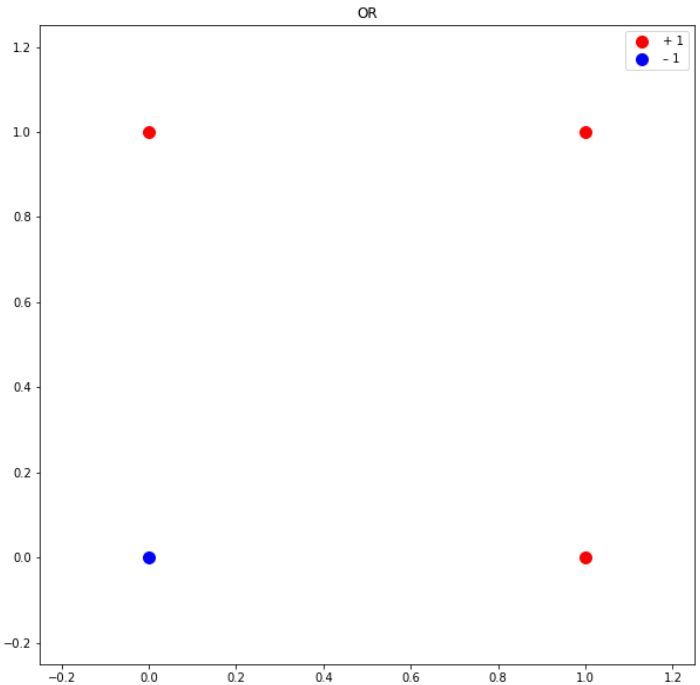
$$f(a) = \begin{cases} +1, & \text{if } a \geq 0 \\ -1, & \text{if } a < 0 \end{cases}$$

Including a Bias



$$y(x) = \mathrm{sign}\left(-\beta \cdot 1 + w_1 x_1 + w_2 \cdot x_2\right)$$

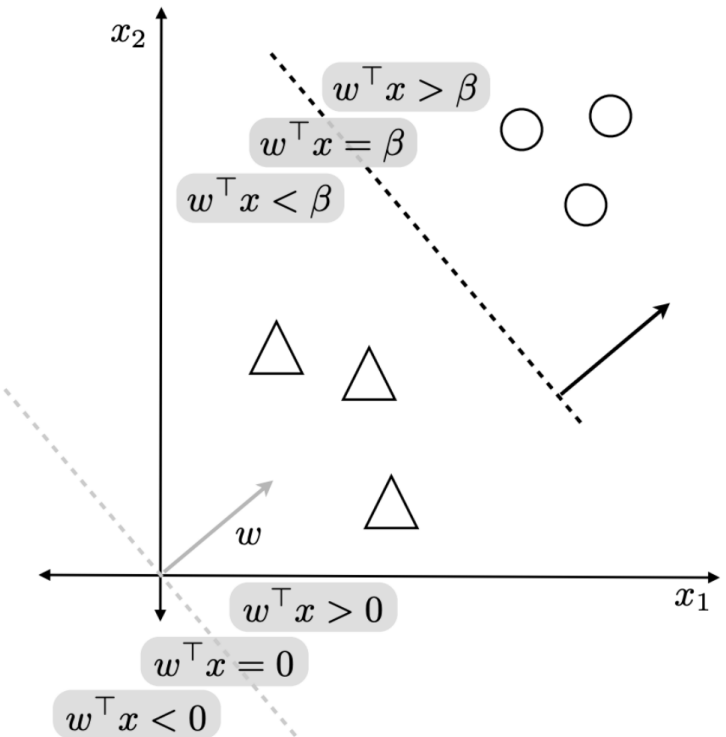
Why do we need a bias?



$$w^T \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \rightarrow +1$$

$$-\beta$$

Augmented Notation



$$\mathbf{w}^\top \mathbf{x} - \beta = \begin{cases} > 0 & \text{if } \mathbf{x} \text{ belongs to } o \\ < 0 & \text{if } \mathbf{x} \text{ belongs to } \Delta \end{cases}$$

The *offset* β can be included in $\mathbf{w}$

$$\tilde{\mathbf{x}} \leftarrow \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \qquad \tilde{\mathbf{w}} \leftarrow \begin{bmatrix} -\beta \\ \mathbf{w} \end{bmatrix}$$

such that

$$\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} - \beta.$$

From now on we will use the augmented notation but will neglect the tilde sign.

Error Function

What is a good $\mathbf{w}$?

The error function $\mathcal{E}_{\mathcal{P}}(\mathbf{w})$ tells us how well our current $\mathbf{w}$ is doing.

We note that a data point $\mathbf{x}_i$ is misclassified iff: $\mathbf{w}^\top \mathbf{x}_i y_i < 0$,
where y_i is the label of $\mathbf{x}_i$ ($y_i \in -1, 1$)

$$\mathbf{w}^\top \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -1$$

$-1 \cdot 1 = -1 < 0$ ✗

$1 \cdot 1 = 1 > 0$ ✓

Error Function

What is a good $\mathbf{w}$?

The error function $\mathcal{E}_{\mathcal{P}}(\mathbf{w})$ tells us how well our current $\mathbf{w}$ is doing.

We note that a data point $\mathbf{x}_i$ is misclassified iff: $\mathbf{w}^\top \mathbf{x}_i y_i < 0$,
where y_i is the label of $\mathbf{x}_i$ ($y_i \in -1, 1$)

Let $\mathcal{M}$ be the set of misclassified datapoints.
Then for a fixed $\mathbf{w}$

$$\mathcal{E}_{\mathcal{P}} = -\sum_{m \in \mathcal{M}} \mathbf{w}^\top \mathbf{x}_m y_m$$

The bigger $\mathcal{E}_{\mathcal{P}}$ is, the more data points are misclassified.

Optimization

We want to find a $\mathbf{w}$ that minimizes the error function:

$$\arg\min_{\mathbf{w}} \mathcal{E}_{\mathcal{P}}(\mathbf{w}) = -\sum_{m \in \mathcal{M}} \mathbf{w}^{\top} \mathbf{x}_m y_m$$

We solve the minimization problem by gradient descent.

Optimization

We want to find a $\mathbf{w}$ that minimizes the error function:

$$\arg\min_{\mathbf{w}} \mathcal{E}_{\mathcal{P}}(\mathbf{w}) = -\sum_{m \in \mathcal{M}} \mathbf{w}^{\top} \mathbf{x}_m y_m$$

We solve the minimization problem by gradient descent.

As the gradient $\nabla_{\mathbf{w}} \mathcal{E}_{\mathcal{P}}$ is a vector that points in the direction in which the error increases, we take a small step η in the opposite direction.

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \eta \nabla_{\mathbf{w}} \mathcal{E}_{\mathcal{P}}$$

Iterative application will give a solution, provided a solution exists.

Stochastic Gradient Descent

We do not need to compute the error w.r.t to all data points to do an update. Instead we can choose one data point randomly.

Stochastic Gradient Descent

We do not need to compute the error w.r.t to all data points to do an update. Instead we can choose one data point randomly.
Perceptron SGD:

1. Initialize $\mathbf{w}^{\text{old}}$ (randomly, $1/n$, ...)

Stochastic Gradient Descent

We do not need to compute the error w.r.t to all data points to do an update. Instead we can choose one data point randomly.
Perceptron SGD:

1. Initialize $\mathbf{w}^{\text{old}}$ (randomly, $1/n$, ...)
2. While there are misclassified data points

   Pick a random misclassified data point $\mathbf{x}_m$
   Descent in direction of the gradient at single data point $\mathbf{x}_m$

$$\mathcal{E}_m(\mathbf{w}) \quad = \quad -\mathbf{w}^\top \mathbf{x}_m y_m$$

Stochastic Gradient Descent

We do not need to compute the error w.r.t to all data points to do an update. Instead we can choose one data point randomly.
Perceptron SGD:

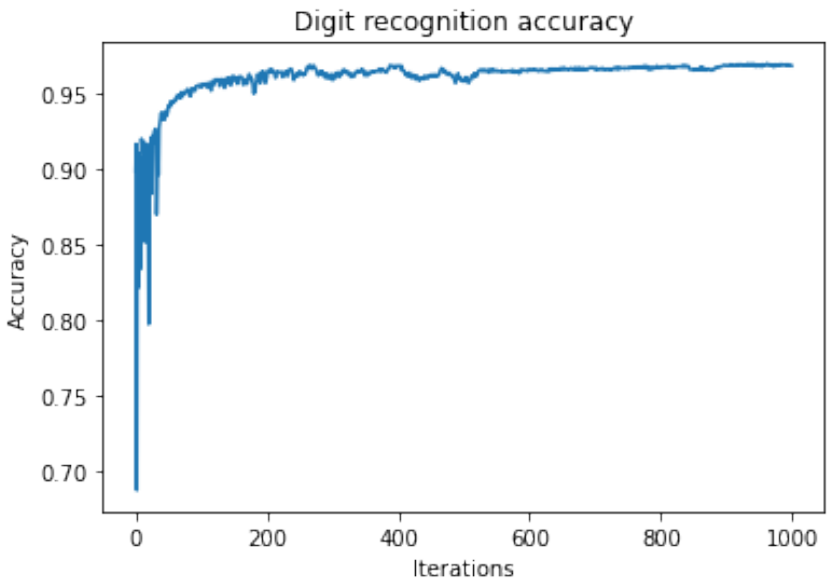
1. Initialize $\mathbf{w}^{\text{old}}$ (randomly, $1/n$, ...)
2. While there are misclassified data points

   Pick a random misclassified data point $\mathbf{x}_m$
   Descent in direction of the gradient at single data point $\mathbf{x}_m$

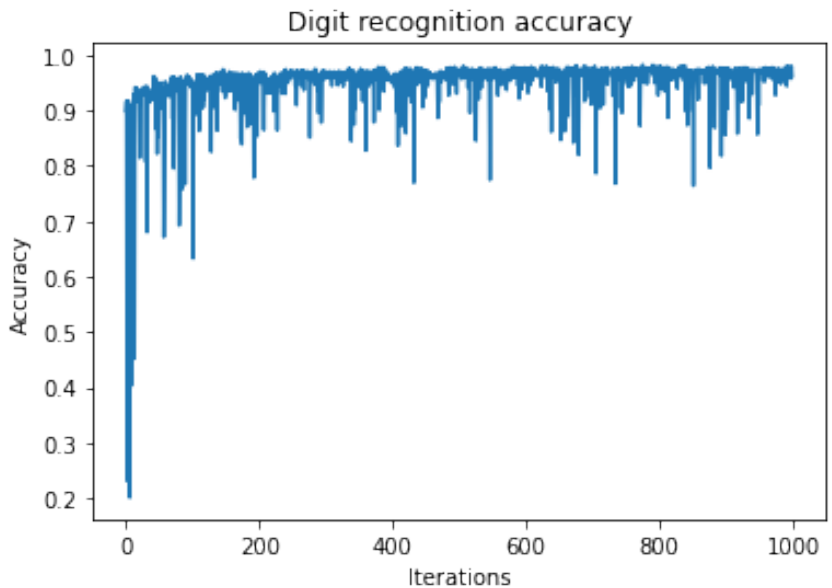
$$\begin{aligned} \mathcal{E}_m(\mathbf{w}) &= -\mathbf{w}^\top \mathbf{x}_m y_m \\ \nabla \mathcal{E}_m(\mathbf{w}) &= -\mathbf{x}_m y_m \\ \mathbf{w}^{\text{new}} &\leftarrow \mathbf{w}^{\text{old}} - \eta \nabla \mathcal{E}_m(\mathbf{w}^{\text{old}}) = \mathbf{w}^{\text{old}} + \eta \mathbf{x}_m y_m \end{aligned}$$

The Learning Rate η

Perceptron: A good learning rate is useful, but not always necessary to reach convergence.

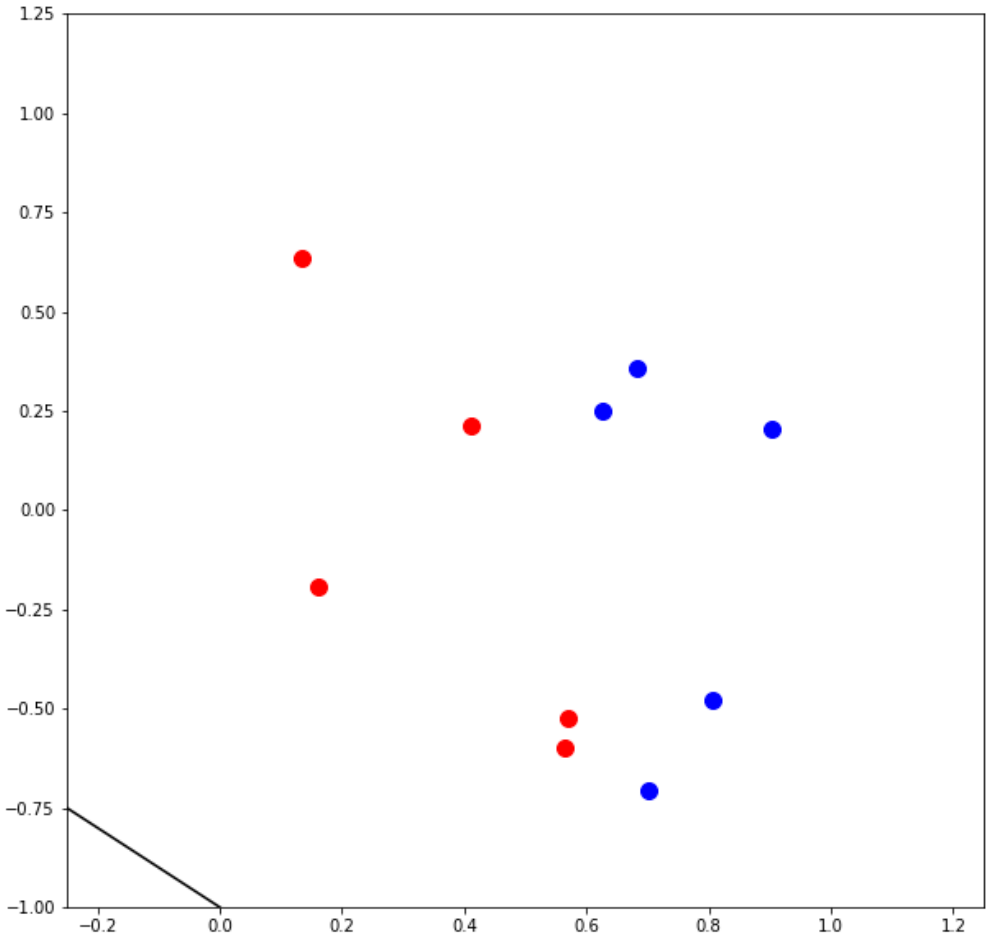


learning rate $\eta = \frac{1}{t}$



learning rate $\eta = 1$

The Perceptron Learning Algorithm in Action



Task 3 - Convergence of the Perceptron

Suppose we have N points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ with class labels $y_1, \ldots, y_N \in \{-1, +1\}$ and that the data set is linearly separable. Prove that the perceptron learning algorithm converges to a separating hyperplane in a finite number of steps.

.

Task 3 - Convergence of the Perceptron

Suppose we have N points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ with class labels $y_1, \ldots, y_N \in \{-1, +1\}$ and that the data set is linearly separable. Prove that the perceptron learning algorithm converges to a separating hyperplane in a finite number of steps.

1. We denote a hyperplane by $\mathbf{w}^\top \mathbf{x} = 0$. Let $\mathbf{z}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$. ($z_i$ lies on the hyperplane and is normalized.) Show that there exists a $\mathbf{w}_{\text{sep}}$ such that

$$\mathbf{w}_{\text{sep}}^T \mathbf{z}_i y_i \geq 1 \tag{1}$$

$\mathbf{w}_{\text{sep}}$ is a $\mathbf{w}$ which defines a decision boundary when all data points are classified correctly.

Task 3 - Convergence of the Perceptron

classified correctly —> $w_s^T z_i y_i \geq \varepsilon \quad \varepsilon > 0$

$$\frac{w_s^T z_i y_i}{\varepsilon} \geq 1$$

$$\tilde{w}_s^T z_i y_i \geq 1$$

Task 3 - Convergence of the Perceptron

Suppose we have N points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ with class labels $y_1, \ldots, y_N \in \{-1, +1\}$ and that the data set is linearly separable. Prove that the perceptron learning algorithm converges to a separating hyperplane in a finite number of steps.

2. Given a current $\mathbf{w}_{\text{old}} \in \mathbb{R}^D$, the perceptron algorithm identifies a point $\mathbf{z}_i$ that is misclassified and produces the update rule $\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \mathbf{z}_i y_i$. Using (1), show that

$$\|\mathbf{w}_{\text{new}} - \mathbf{w}_{\text{sep}}\|^2 \leq \|\mathbf{w}_{\text{old}} - \mathbf{w}_{\text{sep}}\|^2 - 1 \tag{2}$$

This means that with every update we get at least 1 closer to the separating hyperplane.

Task 3 - Convergence of the Perceptron

$$\| w_{new} - w_{sep} \|^2$$

$$= \| w_n \|^2 + \| w_s \|^2 - 2 w_s^T w_n$$

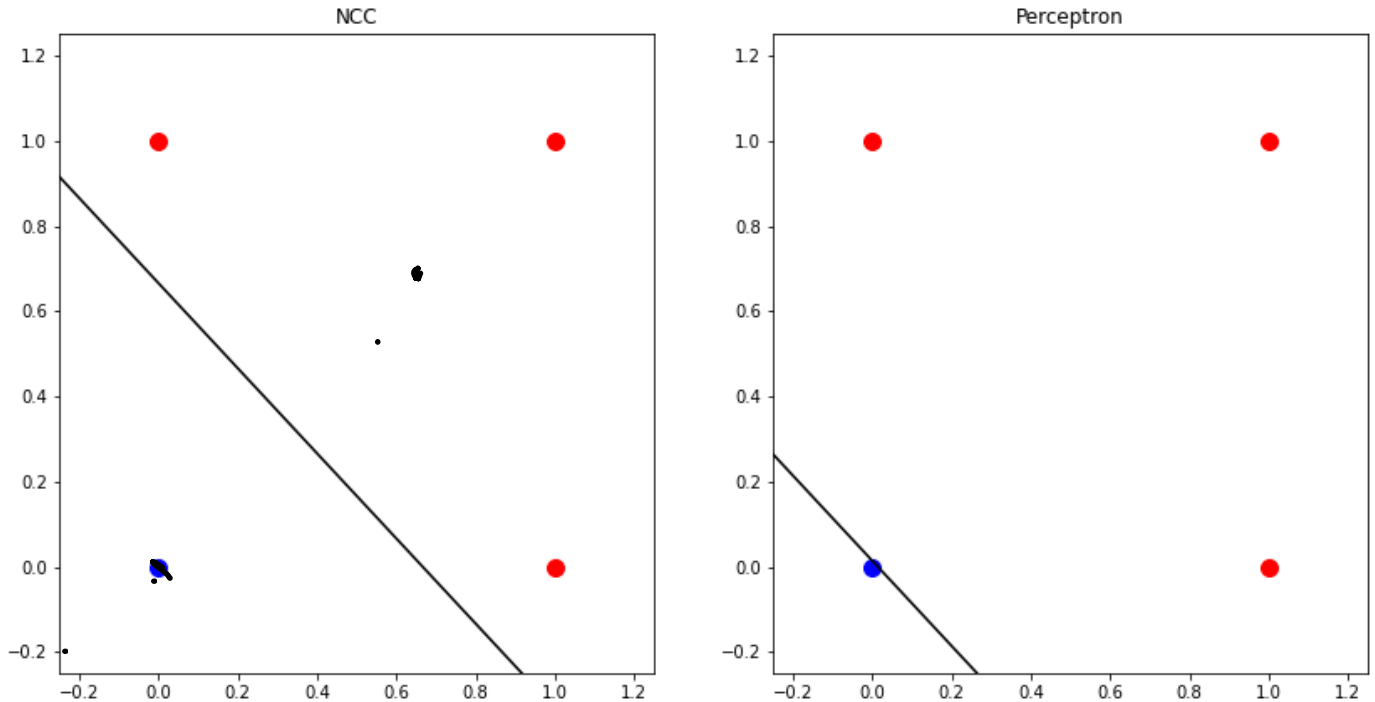
$$= \| w_0 + z_i y_i \|^2 + \| w_s \|^2 - 2 w_s^T (w_0 + z_i y_i)$$

$$= \| w_0 \|^2 + \underbrace{\| z_i y_i \|^2}_{=1} + 2 \underbrace{w_0^T z_i y_i}_{<0} + \| w_s \|^2 - 2 w_s^T w_0 - 2 \underbrace{w_s^T z_i y_i}_{\geq 1}$$

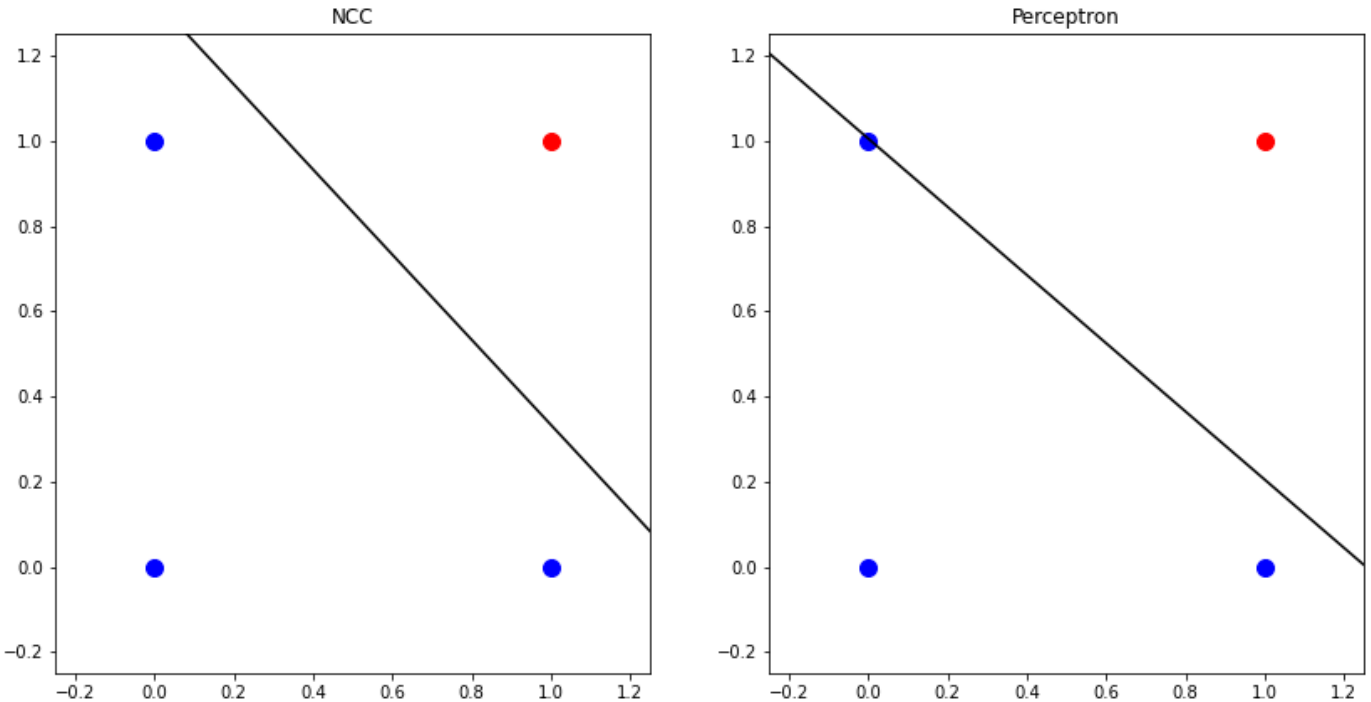
$$\leq \| w_0 \|^2 + \| w_{sep} \|^2 - 2 w_s^T w_0 + 1 - 2$$

$$= \| w_0 - w_s \|^2 - 1$$

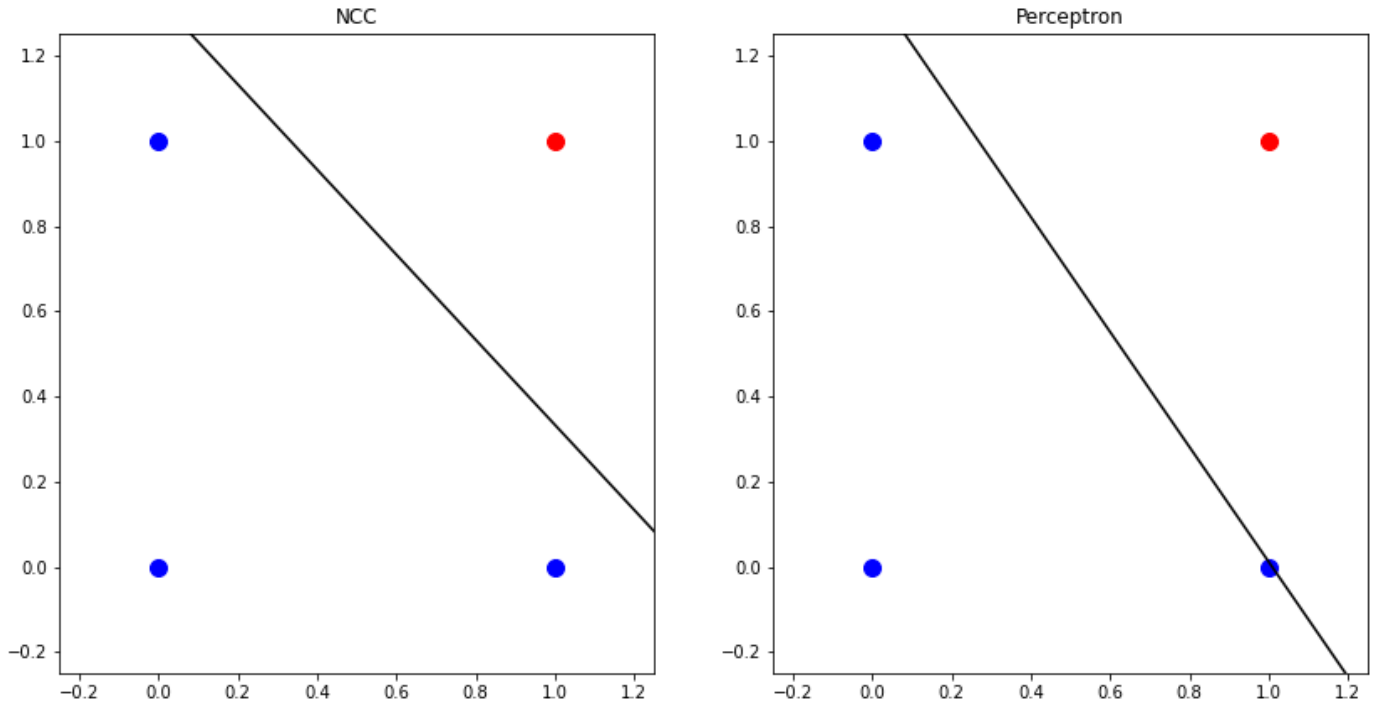
NCC and Perceptron: OR



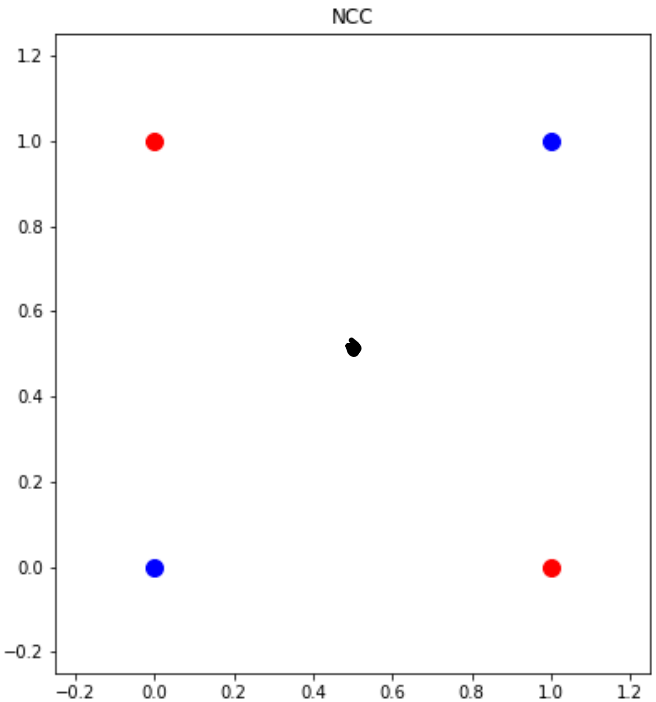
NCC and Perceptron: AND (1)

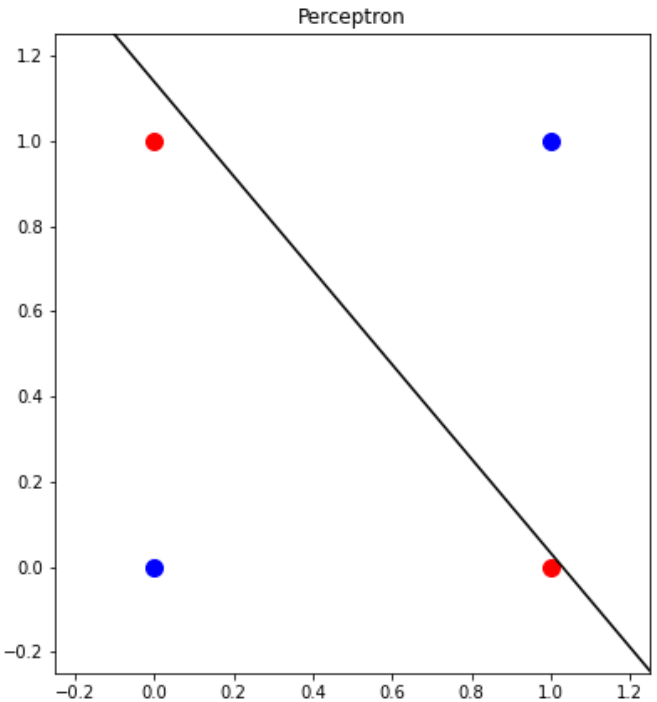


NCC and Perceptron: AND (2)

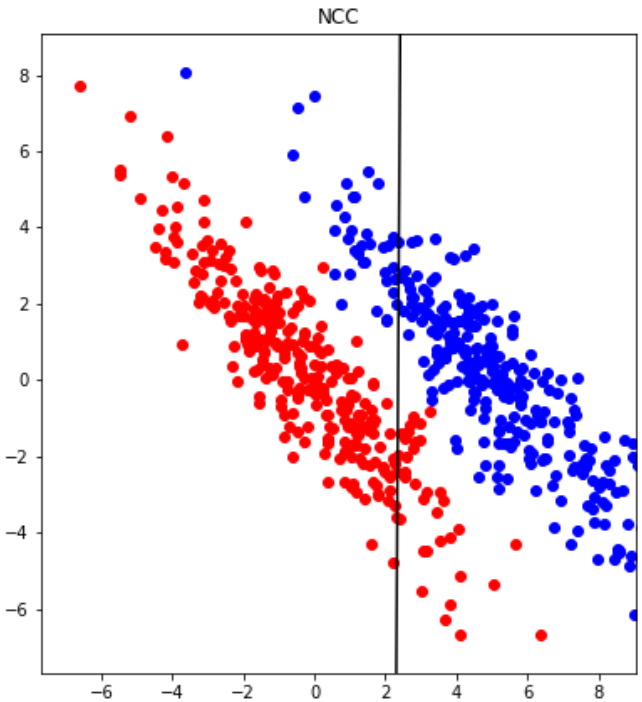


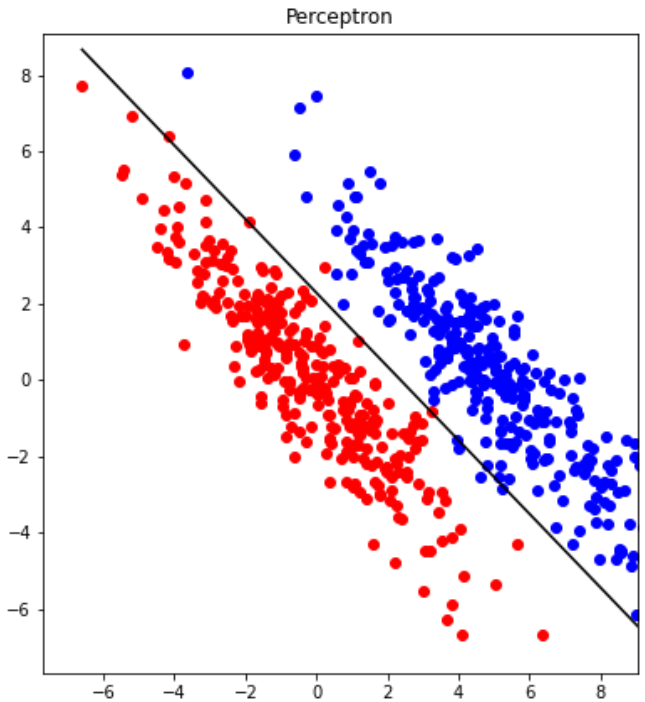
NCC and Perceptron: XOR





NCC and Perceptron





NCC and Perceptron

	NCC	**Perceptron**
Problem	Classification	Classification, (Regression)
Model	$y = \text{sign}(\mathbf{w}^T\mathbf{x})$	$y = f(\mathbf{w}^T\mathbf{x})$
Error	distance to $\mathbf{w}_{+1}, \mathbf{w}_{-1}$	$-\sum_{m \in M} \mathbf{w}^T\mathbf{x}_m y_m$
Optimization	closed form	SGD
Result	always the same	can differ
Application	Cancer Prediction[1]	NLP[2]

[1] (Tibshirani et al., 2002)
[2] (Collins, 2002)

References

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10): 6567–6572, 2002.