CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

Optional content: Perceptron

- Perceptron Background
- Perceptron Algorithm
- Perceptron Intuition

Optional content: Perceptron

- Perceptron Background
- Perceptron Algorithm
- Perceptron Intuition

Hyperplane divides space into positive (+1) and negative (-1)

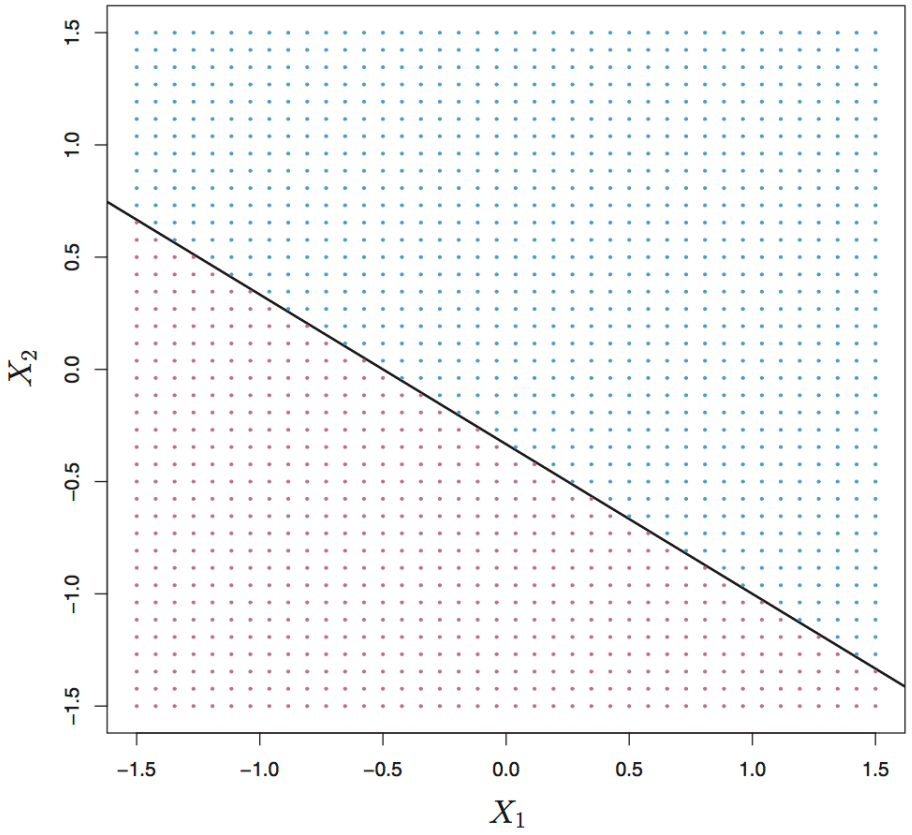


FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$*, and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$*.*

ISL: Figure 9.1

Goal: use training data to create a *separating* hyperplane

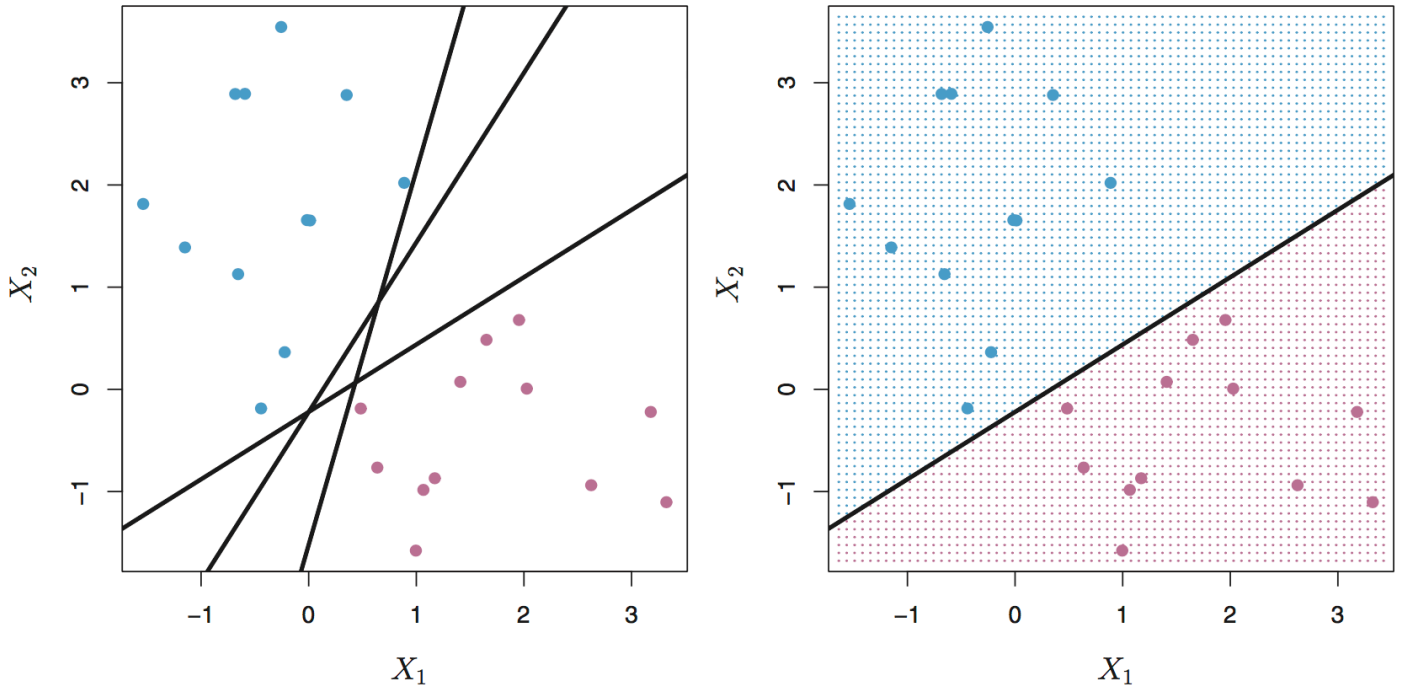


FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.*

ISL: Figure 9.2

These two hyperplanes would likely perform very differently on test data, but they both separate the training data

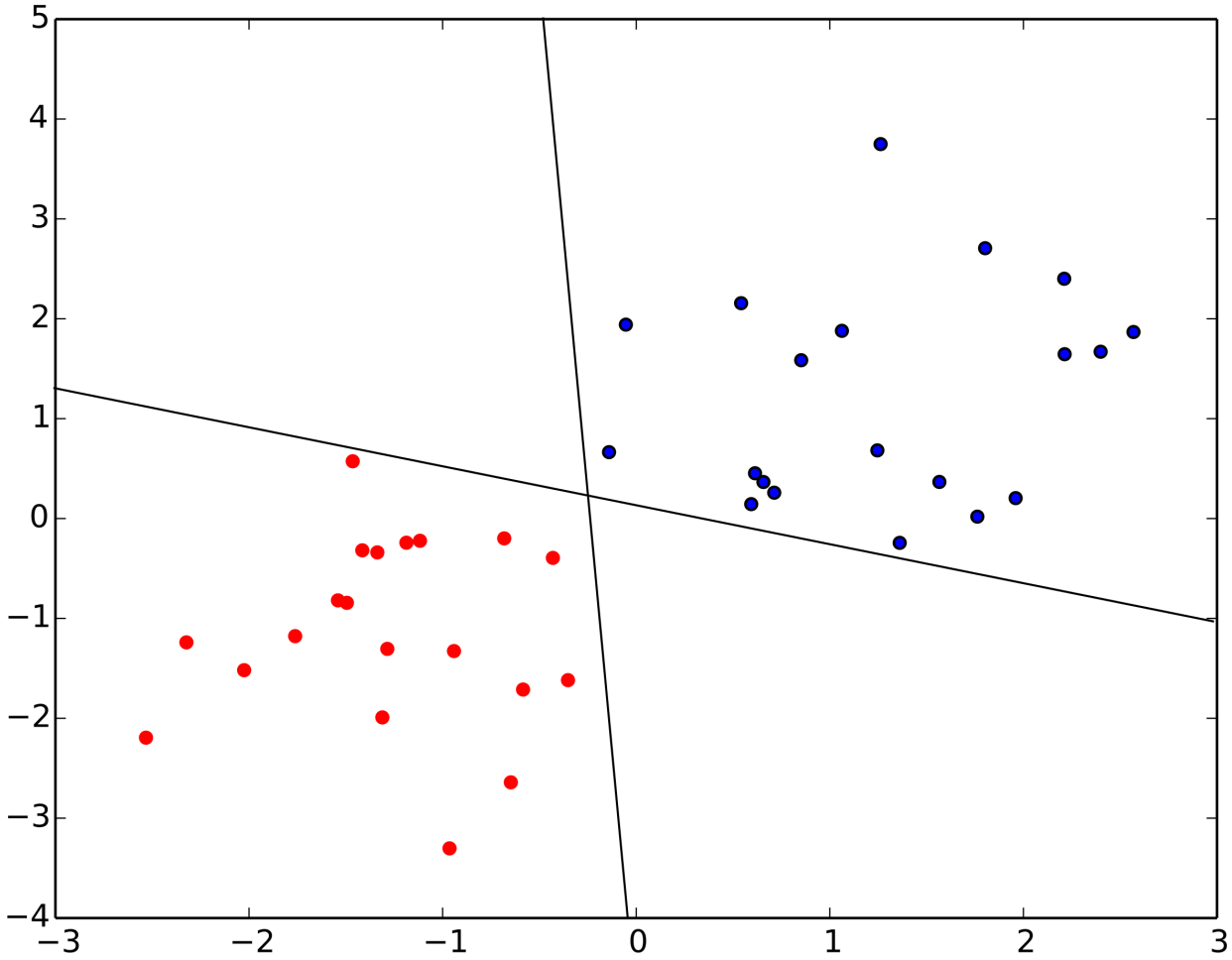


Image: Qwertyus, Wikipedia

Perceptron as a neural network

Biological model of a neuron

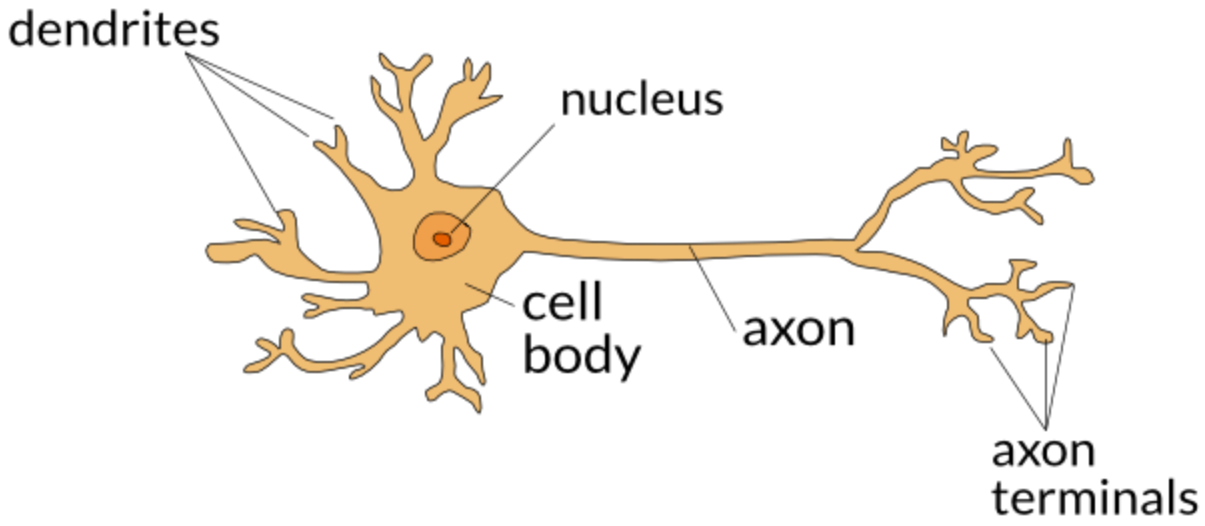


Image: modified from “Applied Go”

Perceptron as a neural network

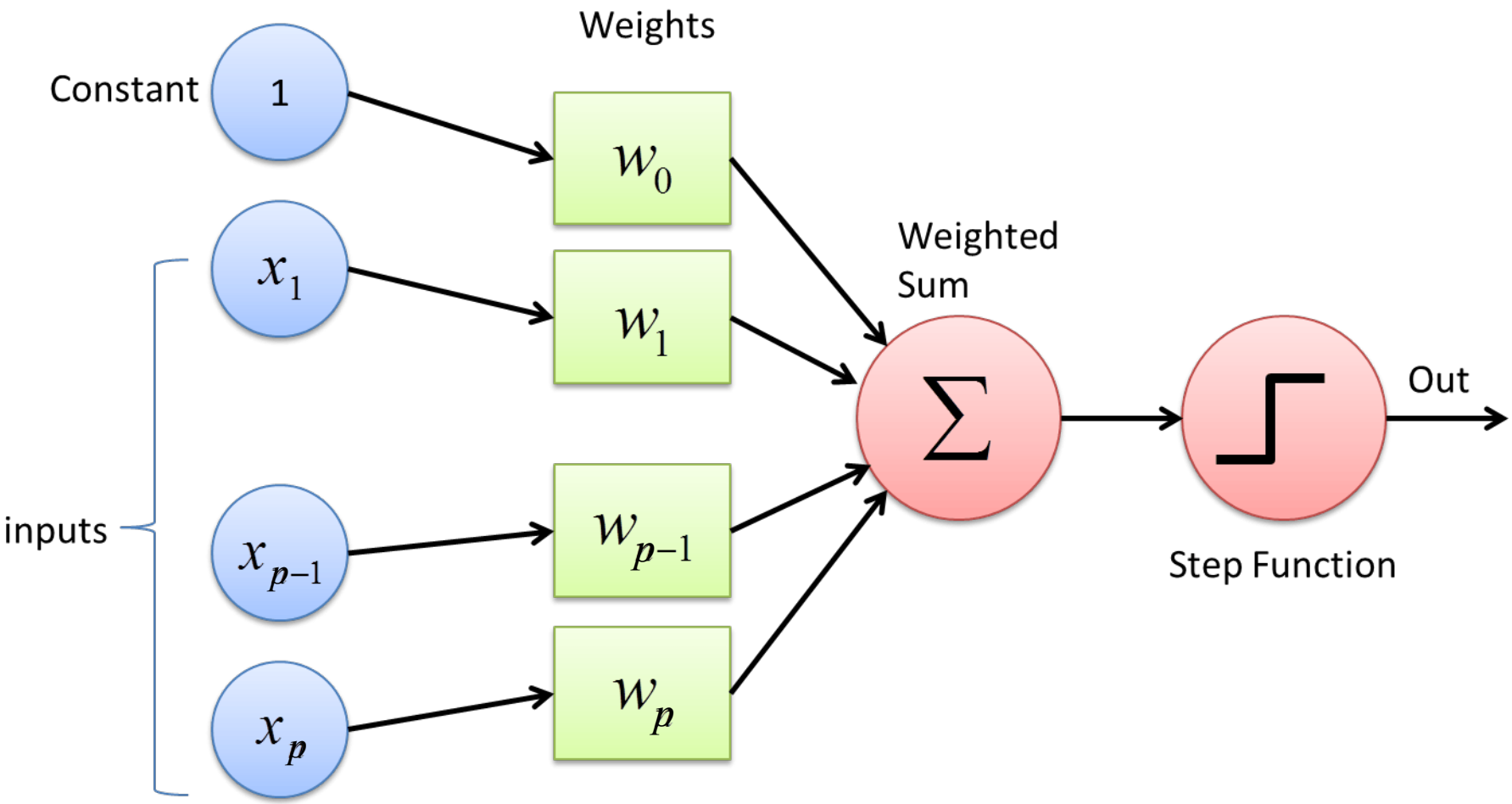


Image: modified from "Towards Data Science"

History of the Perceptron

- Invented in 1957 by Frank Rosenblatt

- Initially thought to be the "solution to AI"

  NYT said the perceptron was "*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*"

- Famous book "Perceptrons" by Marvin Minsky and Seymour Papert (1969)

- Confusion about the text contributed to first "AI winter"

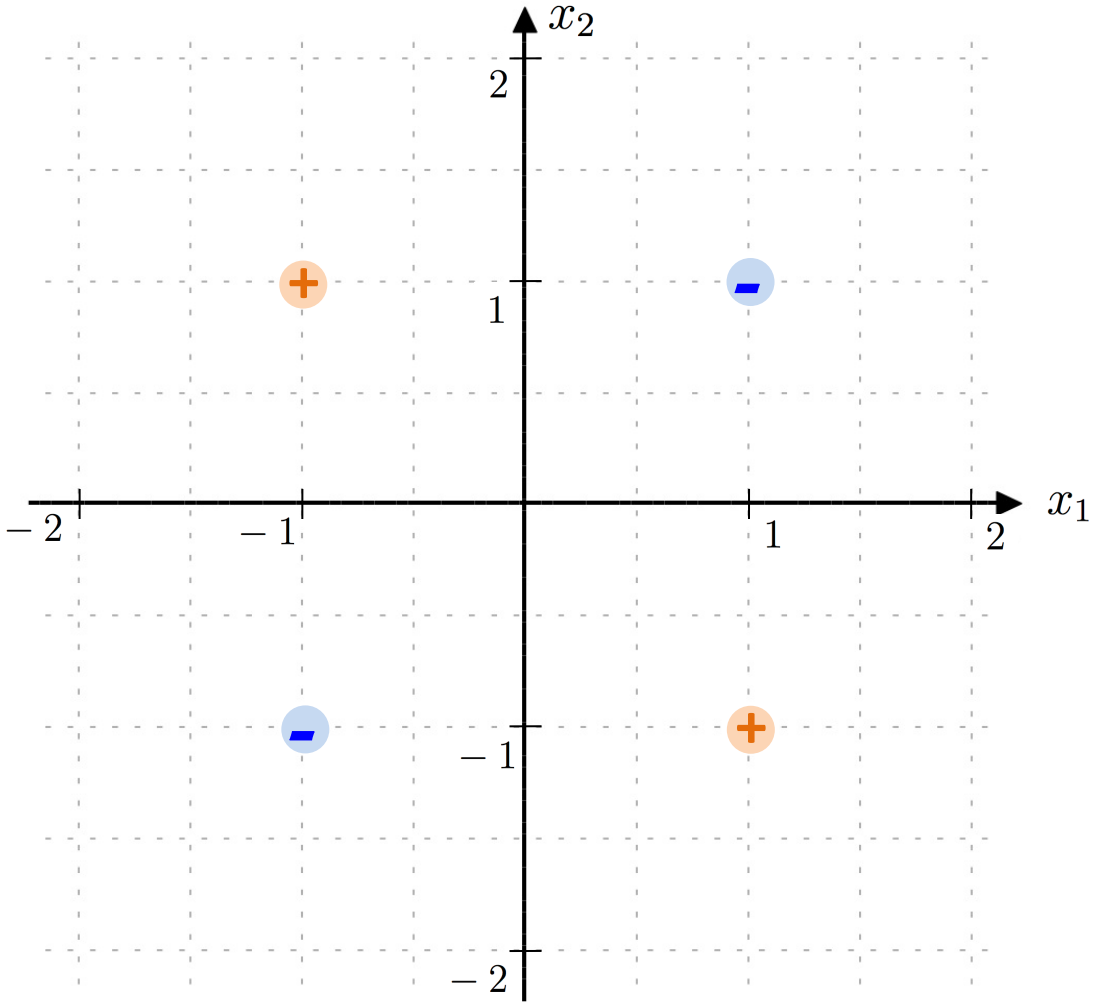
Convergence Guarantee

- Perceptron is guaranteed to converge to a solution if a separating hyperplane exists
- Not guaranteed to converge to a “good” solution
- No guarantees about behavior if a separating hyperplane does not exist!

Perceptron cannot learn XOR

(x_1 = 1 or x_2 = 1, but not both)

Why?

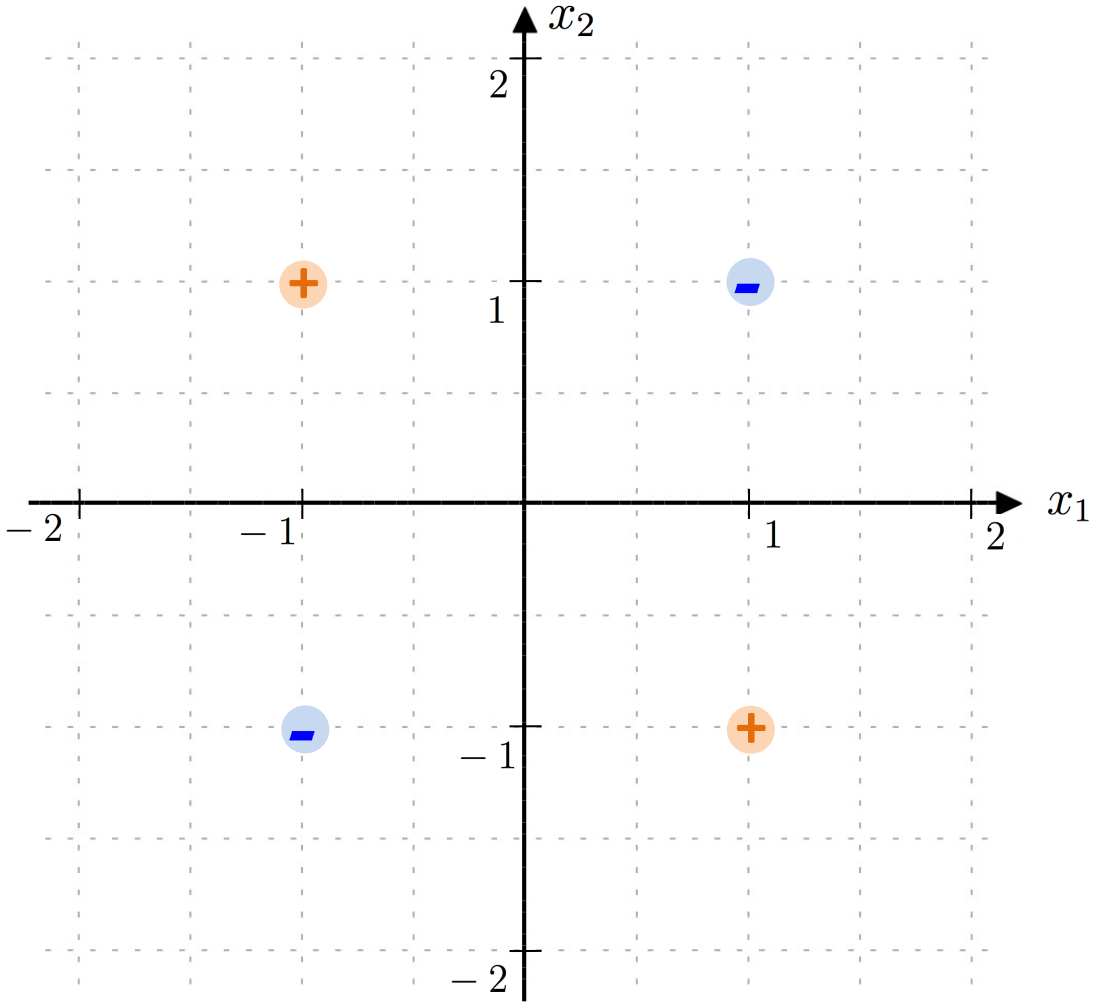


Perceptron cannot learn XOR

(x_1 = 1 or x_2 = 1, but not both)

Why?

Not linearly separable!



Optional content: Perceptron

- Perceptron Background
- Perceptron Algorithm
- Perceptron Intuition

Perceptron Algorithm

$y \in \{-1, 1\}$

goal

$h(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x})$ — same as logistic reg.

$\vec{w} \cdot \vec{x} > 0, \quad \hat{y} = +1$

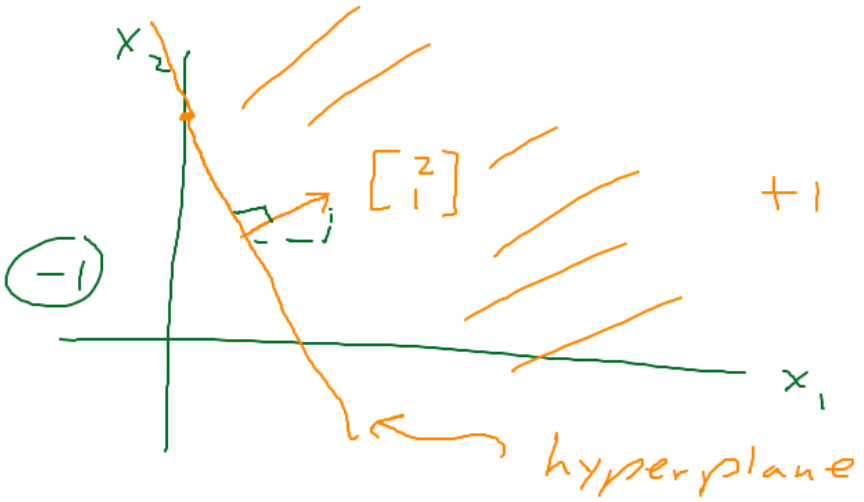
$\vec{w} \cdot \vec{x} \leq 0, \quad \hat{y} = -1$

example $\quad p = 2$

$-5 + 2x_1 + x_2 = 0$ } on boundary

$\vec{w} = \begin{bmatrix} -5 \\ 2 \\ 1 \end{bmatrix}$

$x_2 = -2x_1 + 5$



Perceptron Algorithm

- $\vec{w} = 0$ vector to start
- repeat until train set classified correctly

  select random data point $(\vec{x}_i, y_i)$

  if $\vec{x}_i$ classified correctly:

  do nothing

  else $\vec{w} \leftarrow \vec{w} + y_i \vec{x}_i$

Perceptron Algorithm

$\vec{w} \leftarrow \vec{w} + y_i \vec{x}_i$

don't match

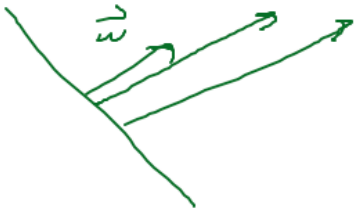
≈ lin. reg

log:

$h(\vec{x})$ was a prob.

① $\vec{w} \leftarrow \vec{w} - \frac{\alpha}{2}\left(h(\vec{x}_i) - y_i\right)\vec{x}_i$

$h(\vec{x}_i)$	y_i	$h(\vec{x}_i) - y_i$	$\cdot\, y_i$
1	-1	2	-2
-1	1	-2	-2



$\vec{w} \leftarrow \vec{w} - \frac{\alpha}{2}(\cancel{-2}) y_i \vec{x}_i$

$$\boxed{\vec{w} \leftarrow \vec{w} + \alpha y_i \vec{x}_i}$$

often: $\alpha = 1$

b/c I don't care about weight magnitude

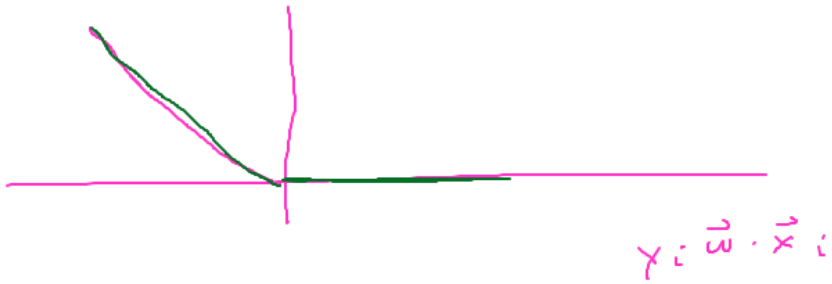
Perceptron Algorithm

Surrogate Loss ("cost" function)

$$J(\vec{w}) = \sum_{i=1}^{n} \max(0, -y_i \vec{w} \cdot \vec{x}_i)$$

true pred

if same sign
$\Rightarrow$ max = 0



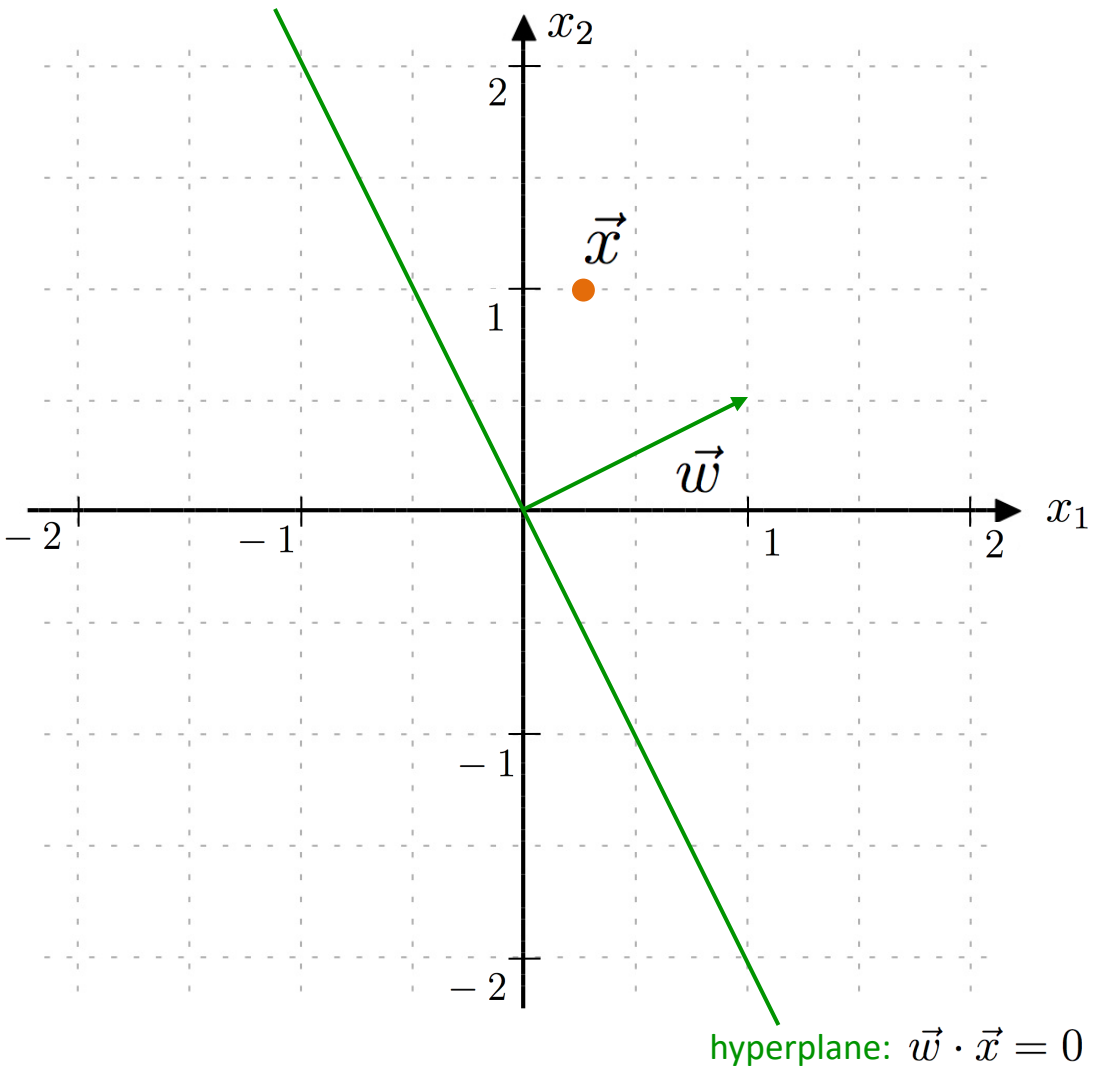
$$\nabla J(\vec{w}) = -y_i \vec{x}_i$$

$$\vec{w} \leftarrow \vec{w} + y_i \vec{x}_i$$

Optional content: Perceptron

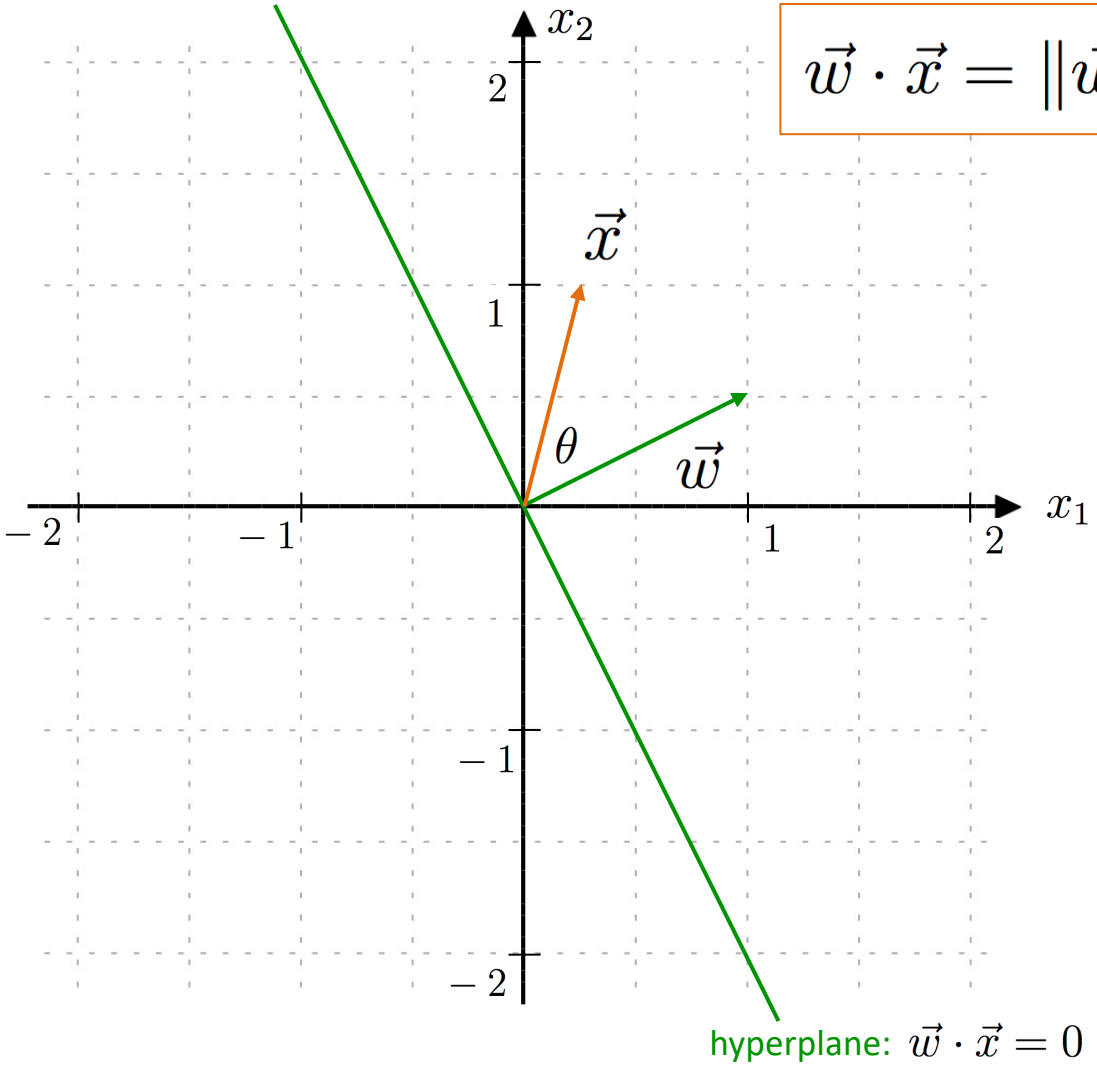
- Perceptron Background
- Perceptron Algorithm
- Perceptron Intuition

Intuition behind the dot product



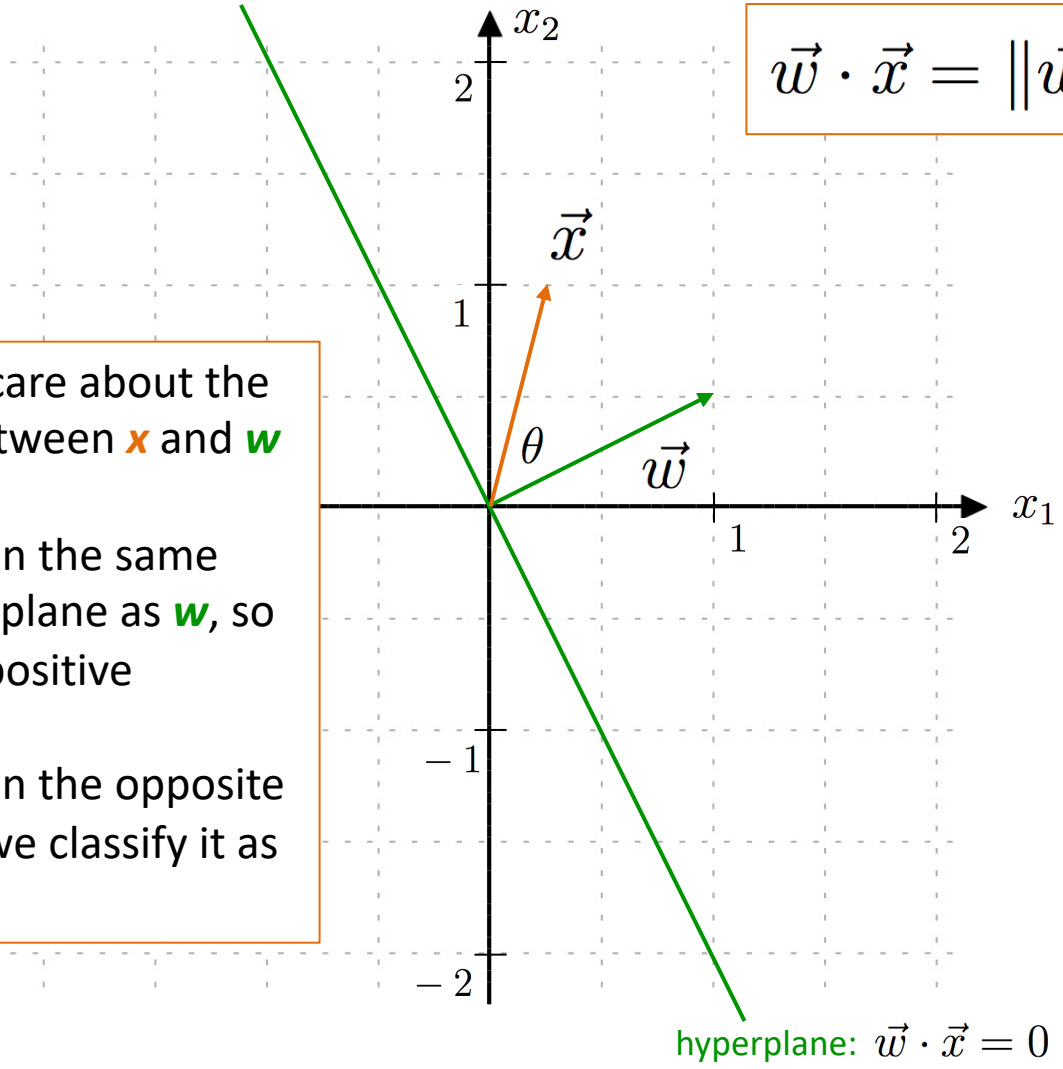
Intuition behind the dot product

$$\vec{w} \cdot \vec{x} = \|\vec{w}\| \|\vec{x}\| \cos\theta$$



Intuition behind the dot product

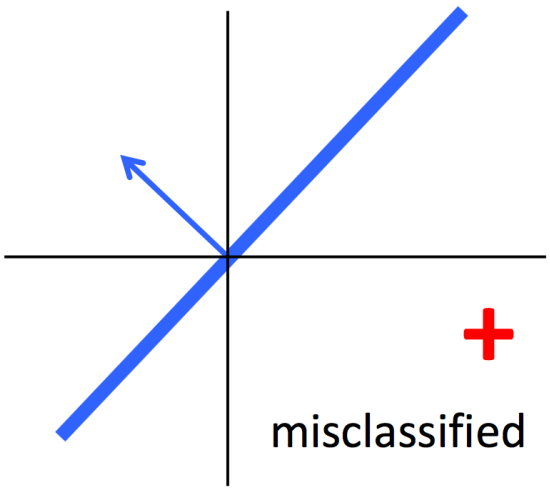
$$\vec{w} \cdot \vec{x} = \|\vec{w}\| \|\vec{x}\| \cos\theta$$

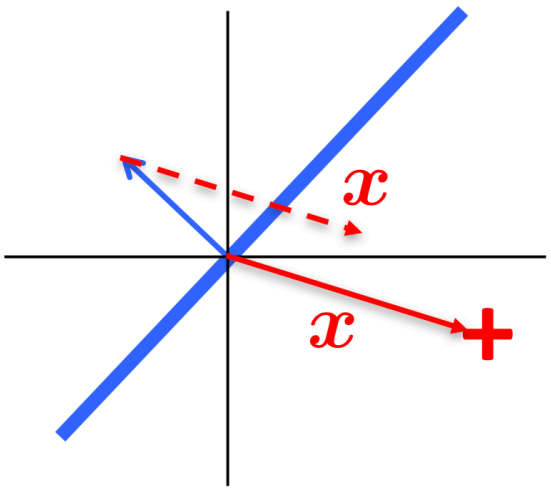


Takeaway: we only care about the sign of the angle between ***x*** and ***w***

- If cos θ > 0, ***x*** is on the same side of the hyperplane as ***w***, so we classify it as positive
- If cos θ < 0, ***x*** is on the opposite side from ***w***, so we classify it as negative

Perceptron algorithm and intuition





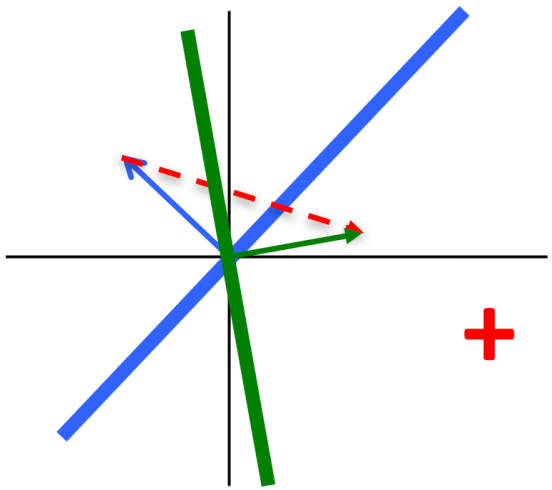
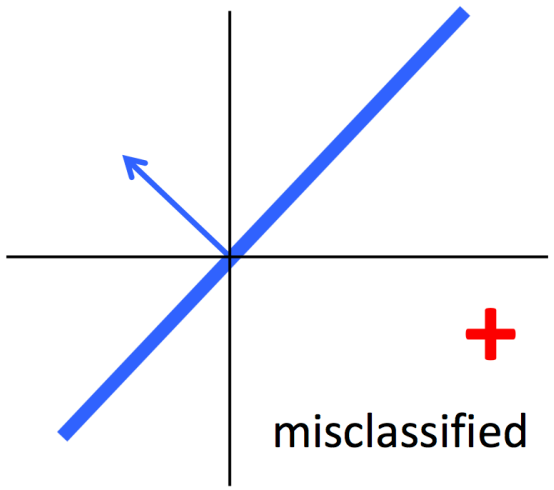
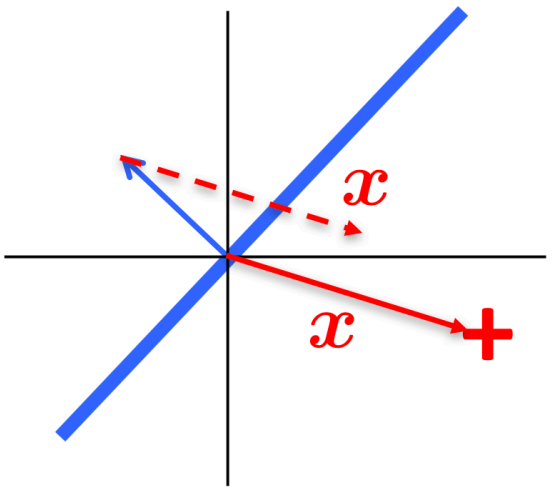
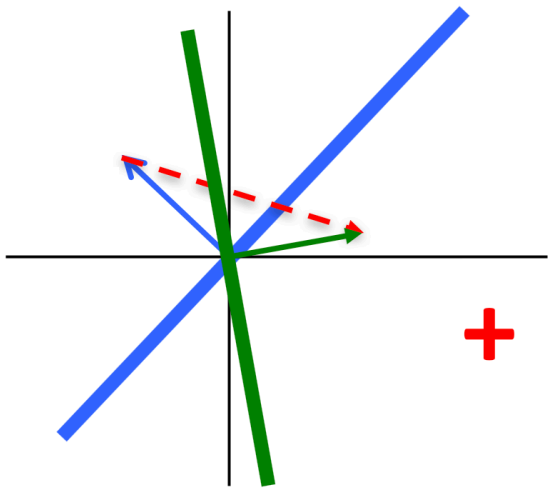


Image and Algorithm: modified from Eric Eaton

Perceptron algorithm and intuition







Let $\vec{w} = [0, 0, \cdots, 0]^T$

Repeat until convergence:

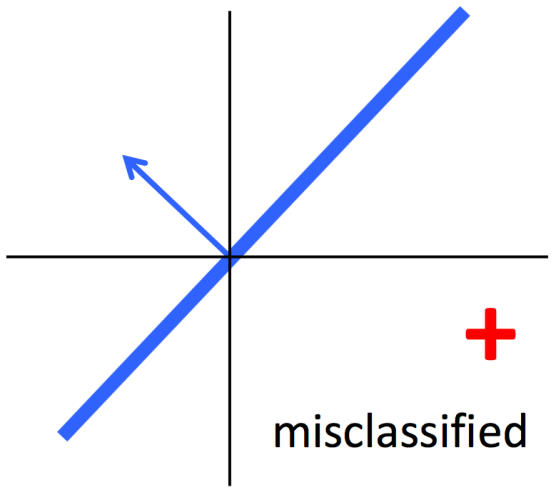
Receive training example $(\vec{x}_i, y_i)$

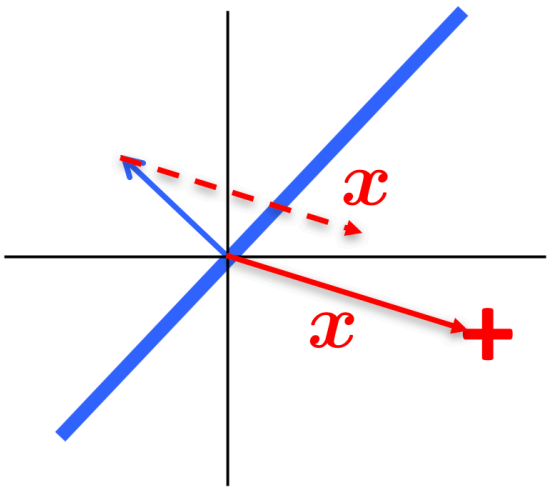
If $y_i(\vec{w}^T \vec{x}_i) \leq 0$ (incorrectly classified)

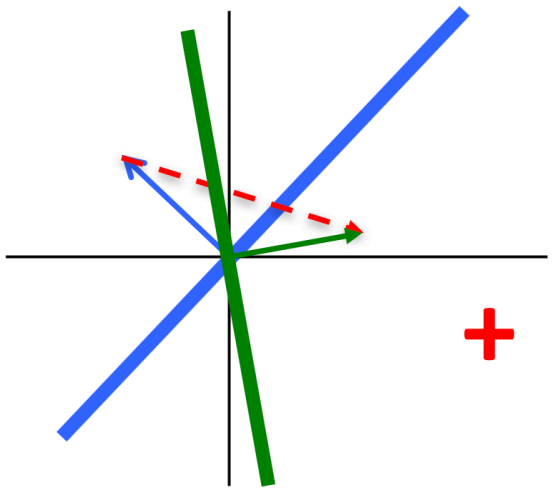
$$\vec{w} \leftarrow \vec{w} + \alpha y_i \vec{x}_i$$

Image and Algorithm: modified from Eric Eaton

Perceptron algorithm and intuition







Let $\vec{w} = [0, 0, \cdots, 0]^T$

Repeat until convergence:

Receive training example $(\vec{x}_i, y_i)$

If $y_i(\vec{w}^T \vec{x}_i) \leq 0$ (incorrectly classified)

$$\vec{w} \leftarrow \vec{w} + \alpha y_i \vec{x}_i$$

Convergence:

- All data points correctly classified
- Fixed number of iterations passed

Often: alpha = 1 (only changes magnitude of weight vector)

Image and Algorithm: modified from Eric Eaton