# Cambridge AI+

## Lecture 6: Perceptron

Thomas Sauerwald

University of Cambridge, Department of Computer Science and Technology

Feb 2021

# Outline

# Landscape of Machine Learning Algorithms

Training Set provided initially

**Supervised Learning**

Nearest Neighbour, Regression, Logistic Regression, Decision Trees and Random Forests, Perceptron, Naive Bayes, Boosting, Support Vector Machines, Neural Networks

Predict unseen data

# Landscape of Machine Learning Algorithms

Training Set provided initially

No Training Set

**Supervised Learning**

Nearest Neighbour, Regression, Logistic Regression, Decision Trees and Random Forests, Perceptron, Naive Bayes, Boosting, Support Vector Machines, Neural Networks

**Unsupervised Learning**

Density Estimation (Maximum Likelihood), Feature Extraction: Dimensionality Reduction, Principal Component Analysis, Singular Value Decomposition, Clustering

Predict unseen data

Extract Knowledge

# Landscape of Machine Learning Algorithms

Training Set provided initially

Feedback after Decisions

No Training Set

**Supervised Learning**

Nearest Neighbour, Regression, Logistic Regression, Decision Trees and Random Forests, Perceptron, Naive Bayes, Boosting, Support Vector Machines, Neural Networks

**Online/Reinforcement Learning**

Online Perceptron, Weighted-Majority, Markov Chains, Hidden-Markov Models, Markov Decision Processes: Temporal Difference, Q-Learning, SARSA

**Unsupervised Learning**

Density Estimation (Maximum Likelihood), Feature Extraction: Dimensionality Reduction, Principal Component Analysis, Singular Value Decomposition, Clustering

Predict unseen data

Maximise Reward

Extract Knowledge

# Outline

Introduction

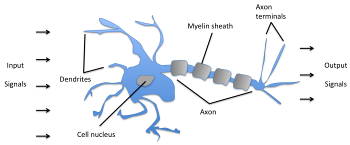
Perceptron

Conclusion, Problems and Solutions

Additional Material: Why Perceptron Works

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold

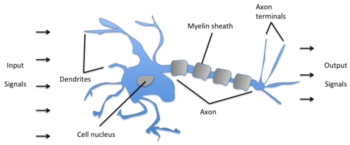



Schematic of a biological neuron.

Source: https://sebastianraschka.com

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines

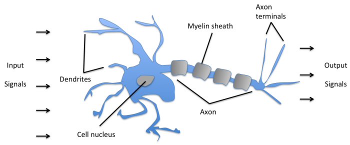



Schematic of a biological neuron.

Source: https://sebastianraschka.com

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks

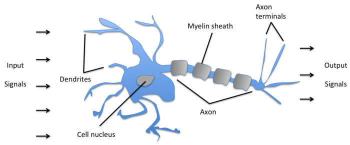



Schematic of a biological neuron.

Source: https://sebastianraschka.com

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks




Schematic of a biological neuron.

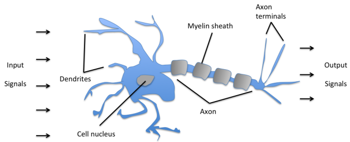
Source: https://sebastianraschka.com

**The Set-Up**

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks




Schematic of a biological neuron.

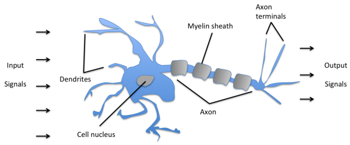
Source: https://sebastianraschka.com

**The Set-Up**

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks



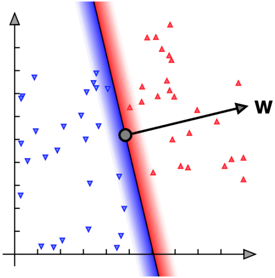

Schematic of a biological neuron.

Source: https://sebastianraschka.com

**The Set-Up**

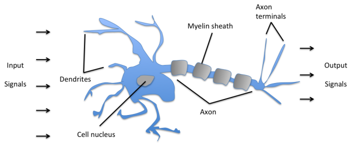
- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$



## Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks




Schematic of a biological neuron.

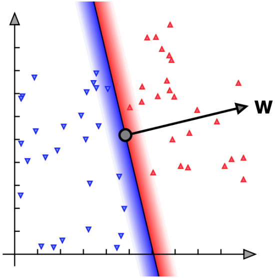
Source: https://sebastianraschka.com

**The Set-Up**

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

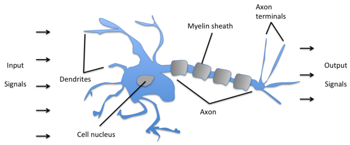
$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

Each predictor assigns every point to $+1$ or $-1$.



# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks




Schematic of a biological neuron.

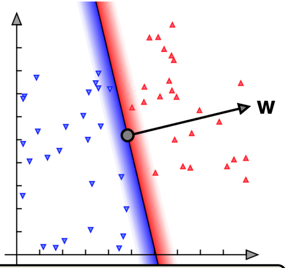
Source: https://sebastianraschka.com

**The Set-Up**

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

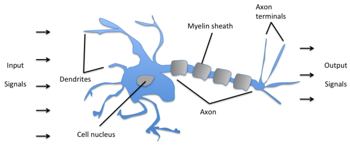
Each predictor assigns every point to $+1$ or $-1$.




Can classify all points correctly only if they are linear separable.

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\leadsto$ Support Vector Machines
- Combining and cascading perceptrons
  $\leadsto$ Neural Networks




Schematic of a biological neuron.

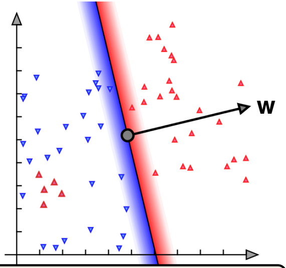
Source: https://sebastianraschka.com

The Set-Up

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

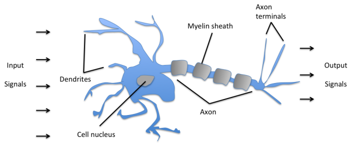
Each predictor assigns every point to $+1$ or $-1$.




Can classify all points correctly only if they are linear separable.

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks



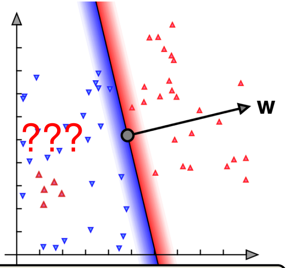

**Schematic of a biological neuron.**

Source: https://sebastianraschka.com

### The Set-Up

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

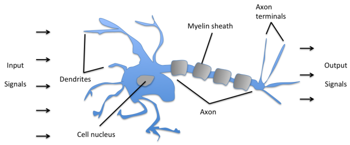



Each predictor assigns every point to $+1$ or $-1$.

Can classify all points correctly only if they are linear separable.

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks



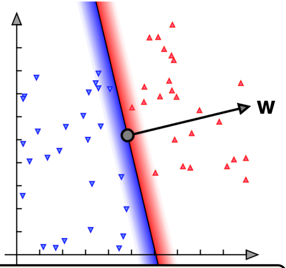

Schematic of a biological neuron.

Source: https://sebastianraschka.com

**The Set-Up**

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

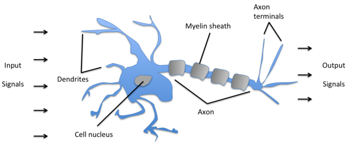



Each predictor assigns every point to $+1$ or $-1$.

Can classify all points correctly only if they are linear separable.

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks




Schematic of a biological neuron.

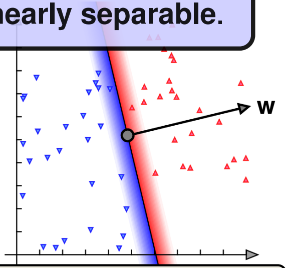
Source: https://sebastianraschka.com

We assume that the input to Perceptron is **linearly separable.**

### The Set-Up

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

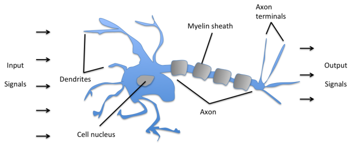



Each predictor assigns every point to $+1$ or $-1$.

Can classify all points correctly only if they are linear separable.

# Introduction to Perceptron

- The Perceptron algorithm was invented 1958 by Frank Rosenblatt
- inspired by a biological neuron: output 1 only if input is above a certain threshold
- Relaxing and smoothing threshold
  $\rightsquigarrow$ Support Vector Machines
- Combining and cascading perceptrons
  $\rightsquigarrow$ Neural Networks



Schematic of a biological neuron.

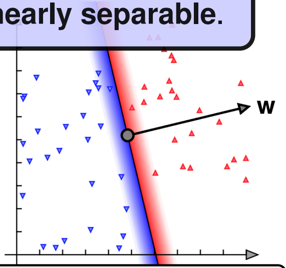
Source: https://sebastianraschka.com

We assume that the input to Perceptron is **linearly separable.**

The Set-Up

- Let $\mathcal{X} = \mathbb{R}^d$ be the feature space
- Let $\mathcal{Y} = \{-1, +1\}$ be the label space
- A predictor is $h : \mathcal{X} \to \{-1, +1\}$
- In Perceptron, a predictor is a halfspace, defined by $\mathbf{w}$ and bias $b$:

$$\mathbf{x} \mapsto \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$




Each predictor assigns every point to $+1$ or $-1$.

Can classify all points correctly only if they are linear separable.

# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is

# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

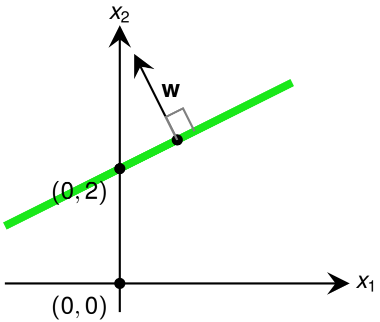
- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

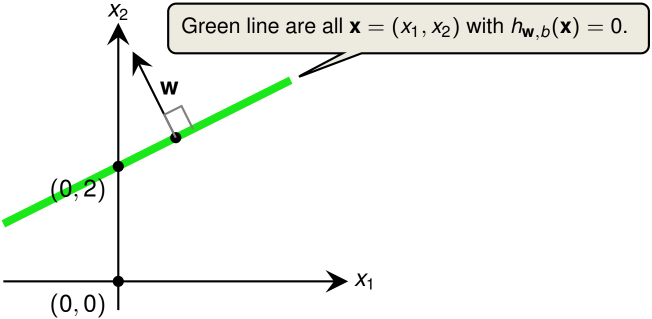


## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

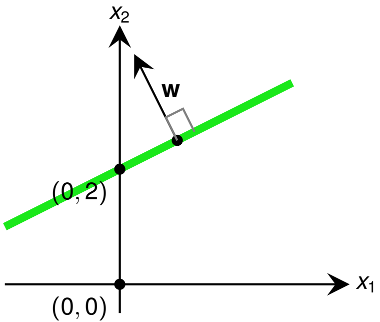


## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

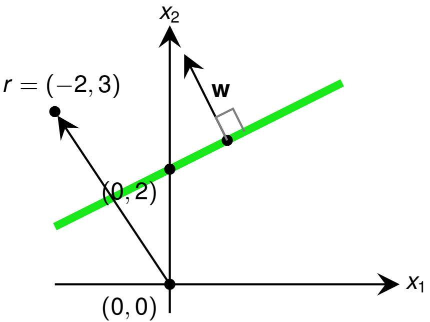


## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

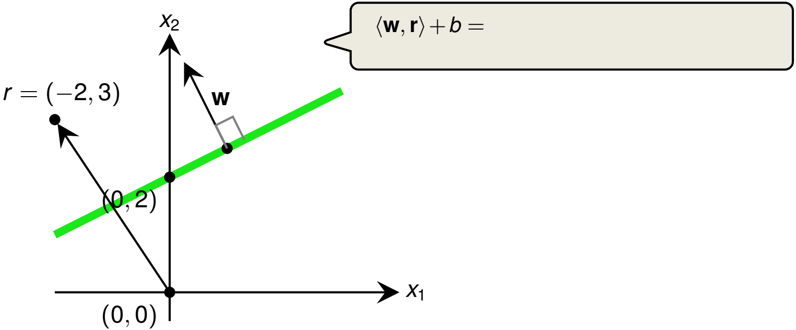


## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

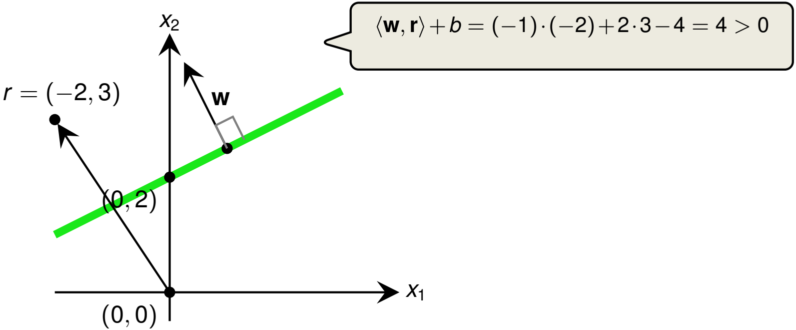


# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

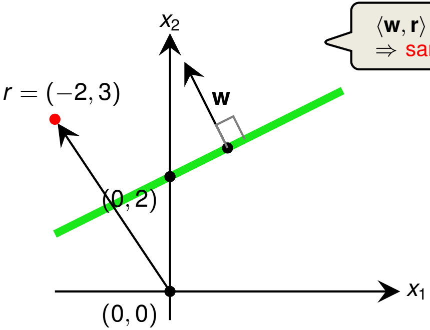


## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$



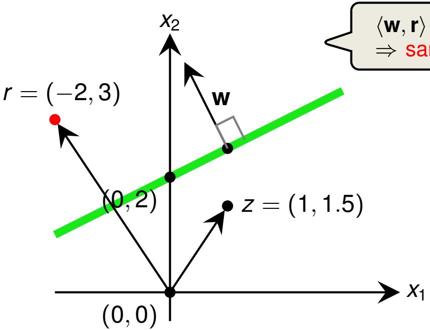

$\langle \mathbf{w}, \mathbf{r} \rangle + b = (-1) \cdot (-2) + 2 \cdot 3 - 4 = 4 > 0$
$\Rightarrow$ same side as $\mathbf{w}$

# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$



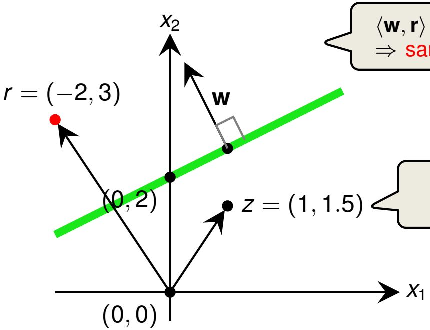

$\langle \mathbf{w}, \mathbf{r} \rangle + b = (-1) \cdot (-2) + 2 \cdot 3 - 4 = 4 > 0$
$\Rightarrow$ same side as $\mathbf{w}$

# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$




$\langle \mathbf{w}, \mathbf{r} \rangle + b = (-1) \cdot (-2) + 2 \cdot 3 - 4 = 4 > 0$
$\Rightarrow$ same side as $\mathbf{w}$

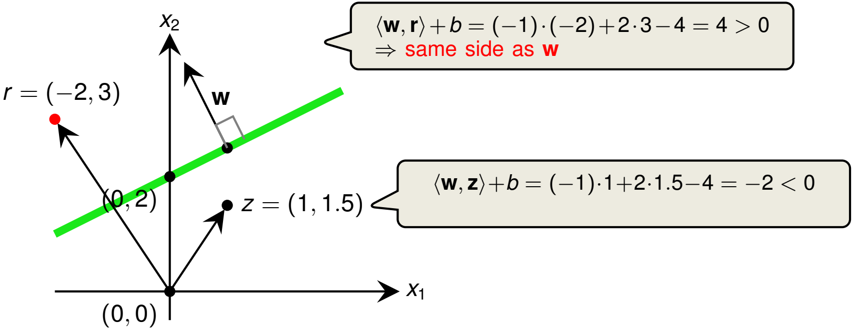
$\langle \mathbf{w}, \mathbf{z} \rangle + b =$

# Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$

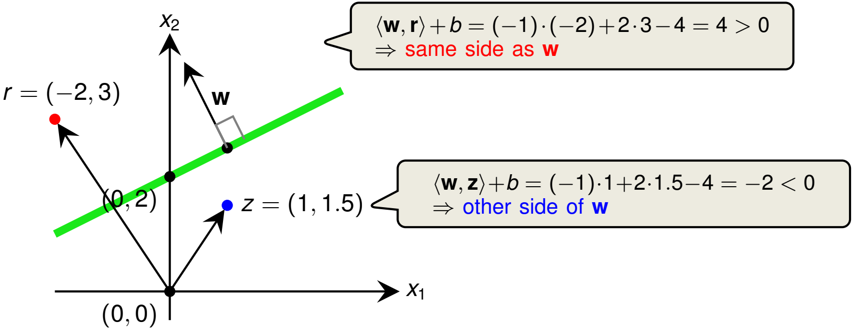


## Linear Functions, Inner Products and Classifications

- A linear function is parameterised by a normal vector $\mathbf{w} \in \mathbb{R}^d$ and bias (scalar) $b$ so that for every $\mathbf{x} \in \mathbb{R}^d$,

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left( \sum_{i=1}^{d} w_i \cdot x_i \right) + b.$$

- Note: sign of $\langle \mathbf{w}, \mathbf{x} \rangle + b$ corresponds to which side $\mathbf{x}$ is
- Example: $d = 2$, $b = -4$, $w_1 = -1$, $w_2 = 2$



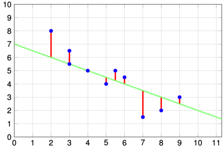
# Linear Regression versus Perceptron

Linear Regression

# Linear Regression versus Perceptron

**Linear Regression**

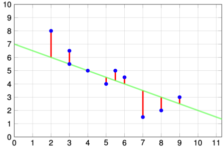
- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data



# Linear Regression versus Perceptron

Linear Regression

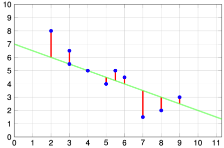
- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent



## Linear Regression versus Perceptron

**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent

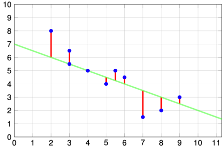



**Perceptron**

## Linear Regression versus Perceptron

**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent



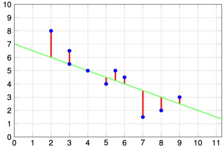

**Perceptron**

- Input: Data set with $d$ features and one **class**

# Linear Regression versus Perceptron

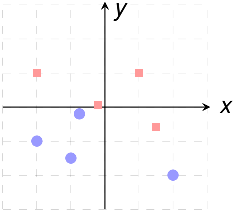
**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent




**Perceptron**

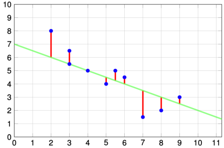
- Input: Data set with $d$ features and one **class**



# Linear Regression versus Perceptron

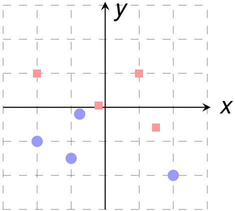
**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent




**Perceptron**

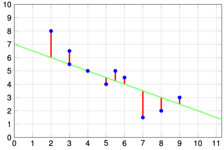
- Input: Data set with $d$ features and one **class**
- Find: $y = \text{sign}(\langle w, x \rangle + b)$ to separate classes



# Linear Regression versus Perceptron

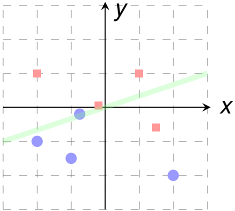
Linear Regression

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent




Perceptron

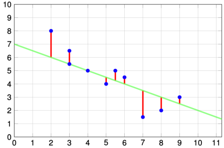
- Input: Data set with $d$ features and one **class**
- Find: $y = \mathrm{sign}(\langle w, x \rangle + b)$ to separate classes



# Linear Regression versus Perceptron

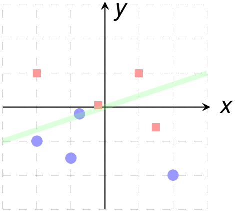
**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent




**Perceptron**

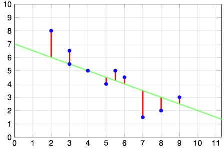
- Input: Data set with $d$ features and one **class**
- Find: $y = \mathrm{sign}(\langle w, x \rangle + b)$ to separate classes
- Condition: No point should be misclassified



# Linear Regression versus Perceptron

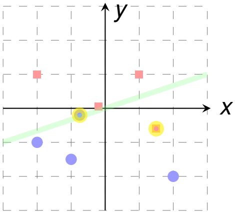
**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent




**Perceptron**

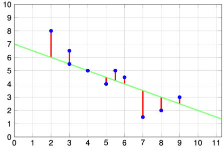
- Input: Data set with $d$ features and one **class**
- Find: $y = \mathrm{sign}(\langle w, x \rangle + b)$ to separate classes
- Condition: No point should be misclassified



# Linear Regression versus Perceptron

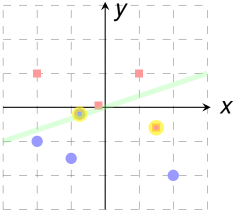
**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent




**Perceptron**

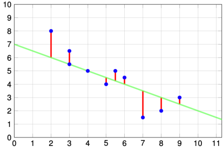
- Input: Data set with $d$ features and one **class**
- Find: $y = \mathrm{sign}(\langle w, x \rangle + b)$ to separate classes
- Condition: No point should be misclassified
- Solution: Stochastic Gradient Descent (or LPs)



# Linear Regression versus Perceptron

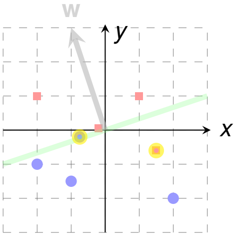
**Linear Regression**

- Input: Data set with $d$ features and one **outcome**
- Find: $y = \langle w, x \rangle + b$ to fit data
- Optimisation: Minimise Squared Error
- Solution: (Stochastic) Gradient Descent

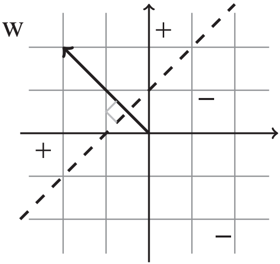



**Perceptron**

- Input: Data set with $d$ features and one **class**
- Find: $y = \text{sign}(\langle w, x \rangle + b)$ to separate classes
- Condition: No point should be misclassified
- Solution: Stochastic Gradient Descent (or LPs)

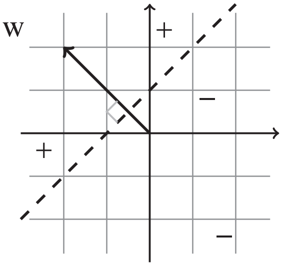


# From Affine Linear Functions to homogenous linear Functions



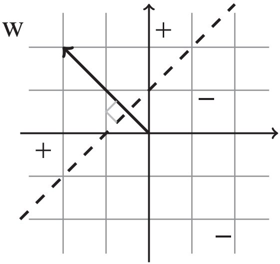
# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$



# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \dots, w_d) \in \mathbb{R}^{d+1}$
- Let $\mathbf{x}' = (1, x_1, x_2, \dots, x_d) \in \mathbb{R}^{d+1}$

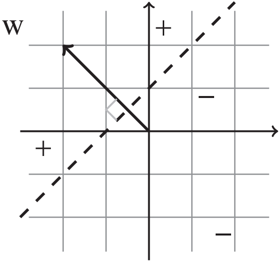


# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- Let $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

$\Rightarrow$ Then

$$h_{\mathbf{w},b}(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle$$

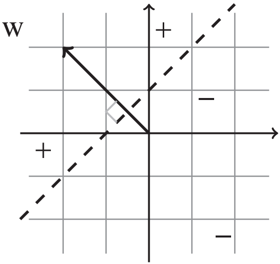


# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- Let $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

$\Rightarrow$ Then

$$h_{\mathbf{w},b}(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle$$



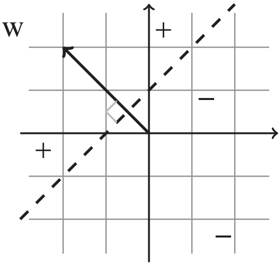
# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- Let $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

$\Rightarrow$ Then

$$h_{\mathbf{w},b}(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle$$

- Hence we need to learn a correct $\mathbf{w}' \in \mathbb{R}^{d+1}$



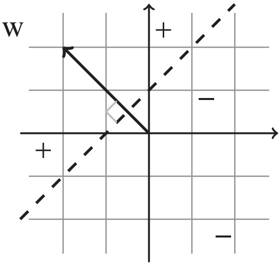
# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- Let $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

$\Rightarrow$ Then

$$h_{\mathbf{w},b}(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle$$

- Hence we need to learn a correct $\mathbf{w}' \in \mathbb{R}^{d+1}$
- A common trick in Machine Learning so that all parameters are unified.



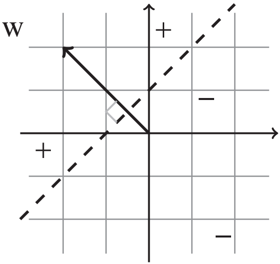
# From Affine Linear Functions to homogenous linear Functions

- Let $\mathbf{w}' = (b, w_1, w_2, \ldots, w_d) \in \mathbb{R}^{d+1}$
- Let $\mathbf{x}' = (1, x_1, x_2, \ldots, x_d) \in \mathbb{R}^{d+1}$

$\Rightarrow$ Then

$$h_{\mathbf{w},b}(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle$$

- Hence we need to learn a correct $\mathbf{w}' \in \mathbb{R}^{d+1}$
- A common trick in Machine Learning so that all parameters are unified.



## The Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
  **for** $t = 1, 2, \ldots$
    **if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
    **else**
      **output** $\mathbf{w}^{(t)}$

# The Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
  **for** $t = 1, 2, \ldots$
    **if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
    **else**
      **output** $\mathbf{w}^{(t)}$

Check whether there is a point $\mathbf{x}_i$ which is misclassified by $\mathbf{w}^{(t)}$!

# The Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
**else**
**output** $\mathbf{w}^{(t)}$

Check whether there is a point $\mathbf{x}_i$ which is misclassified by $\mathbf{w}^{(t)}$!

- Main component is the **additive** update rule within the for-loop

## The Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
  **if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
  **else**
    **output** $\mathbf{w}^{(t)}$

Check whether there is a point $\mathbf{x}_i$ which is misclassified by $\mathbf{w}^{(t)}$!

- Main component is the **additive** update rule within the for-loop
- There is also an online-version of Perceptron for reinforcement learning

# The Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
  **for** $t = 1, 2, \ldots$
    **if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
    **else**
      **output** $\mathbf{w}^{(t)}$

Check whether there is a point $\mathbf{x}_i$ which is misclassified by $\mathbf{w}^{(t)}$!

- Main component is the **additive** update rule within the for-loop
- There is also an online-version of Perceptron for reinforcement learning
- It is possible to find a hyperplane directly using Linear Programming

# The Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
**else**
**output** $\mathbf{w}^{(t)}$

Check whether there is a point $\mathbf{x}_i$ which is misclassified by $\mathbf{w}^{(t)}$!

- Main component is the **additive** update rule within the for-loop
- There is also an online-version of Perceptron for reinforcement learning
- It is possible to find a hyperplane directly using Linear Programming
- We will now see an illustration of this algorithm on a small data set

# Illustration of Perceptron (1/2)

## Illustration of Perceptron (1/2)

Original Input

- dimension $d = 2$, $m = 8$ points

# Illustration of Perceptron (1/2)

Original Input

- dimension $d = 2$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (-2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (-0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (-2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (-1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (-0.2, 0.05)$, $y_8 = 1$ pos.

# Illustration of Perceptron (1/2)

Original Input

- dimension $d = 2$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (-2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (-0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (-2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (-1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (-0.2, 0.05)$, $y_8 = 1$ pos.

Output

Find normal vector $(w_1, w_2)$ and bias $b$ correctly classifying all points.

# Illustration of Perceptron (1/2)

Original Input

- dimension $d = 2$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (-2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (-0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (-2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (-1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (-0.2, 0.05)$, $y_8 = 1$ pos.

New Input

- dimension $d = 3$, $m = 8$ points

Output

Find normal vector $(w_1, w_2)$ and bias $b$ correctly classifying all points.

# Illustration of Perceptron (1/2)

**Original Input**

- dimension $d = 2$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (-2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (-0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (-2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (-1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (-0.2, 0.05)$, $y_8 = 1$ pos.

**New Input**

- dimension $d = 3$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (1, -2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (1, -0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (1, -2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (1, -1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (1, 2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1, 1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (1, -0.2, 0.05)$, $y_8 = 1$ pos.

**Output**

Find normal vector $(w_1, w_2)$ and bias $b$ correctly classifying all points.

# Illustration of Perceptron (1/2)

**Original Input**

- dimension $d = 2$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (-2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (-0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (-2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (-1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (-0.2, 0.05)$, $y_8 = 1$ pos.

**New Input**

- dimension $d = 3$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (1, -2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (1, -0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (1, -2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (1, -1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (1, 2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1, 1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (1, -0.2, 0.05)$, $y_8 = 1$ pos.

**Output**

Find normal vector $(w_1, w_2)$ and bias $b$ correctly classifying all points.

**Output**

Find normal vector $(w'_1, w'_2, w'_3)$ correctly classifying all points.

# Illustration of Perceptron (1/2)

**Original Input**

- dimension $d = 2$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (-2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (-0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (-2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (-1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (-0.2, 0.05)$, $y_8 = 1$ pos.

**New Input**

- dimension $d = 3$, $m = 8$ points
- $\mathbf{x}_1 = (1, 1, 1)$, $y_1 = 1$ pos.
- $\mathbf{x}_2 = (1, -2, -1)$, $y_2 = -1$ neg.
- $\mathbf{x}_3 = (1, -0.75, -0.2)$, $y_3 = -1$ neg.
- $\mathbf{x}_4 = (1, -2, 1)$, $y_4 = 1$ pos.
- $\mathbf{x}_5 = (1, -1, -1.5)$, $y_5 = -1$ neg.
- $\mathbf{x}_6 = (1, 2, -2)$, $y_6 = -1$ neg.
- $\mathbf{x}_7 = (1, 1.5, -0.6)$, $y_7 = 1$ pos.
- $\mathbf{x}_8 = (1, -0.2, 0.05)$, $y_8 = 1$ pos.

**Output**

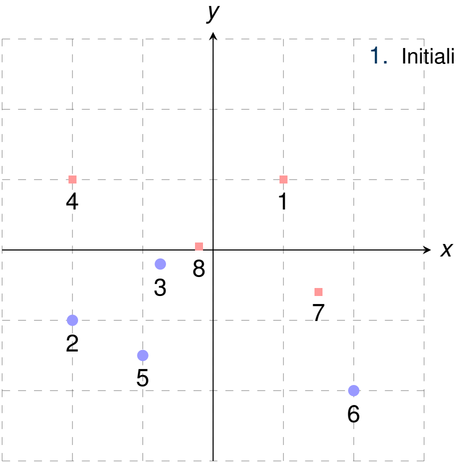
Find normal vector $(w_1, w_2)$ and bias $b$ correctly classifying all points.

**Output**

Find normal vector $(w_1', w_2', w_3')$ correctly classifying all points.

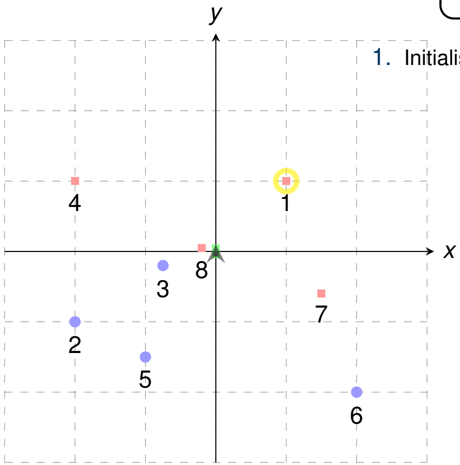
$b = w_1'$, $w_1 = w_2'$ and $w_2 = w_3'$!

## Illustration of Perceptron (2/2)




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$.

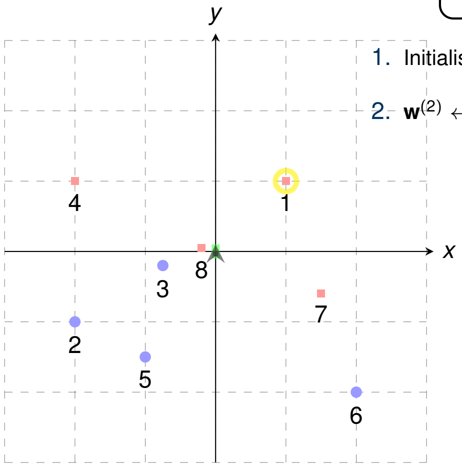
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$

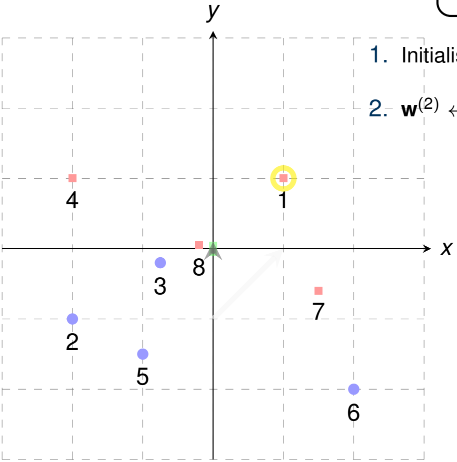
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$.

# Illustration of Perceptron (2/2)

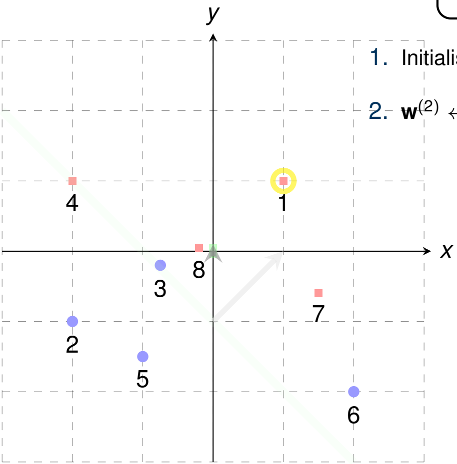



> **if** $(\exists\, i$ s.t. $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
> $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$

2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$.

# Illustration of Perceptron (2/2)

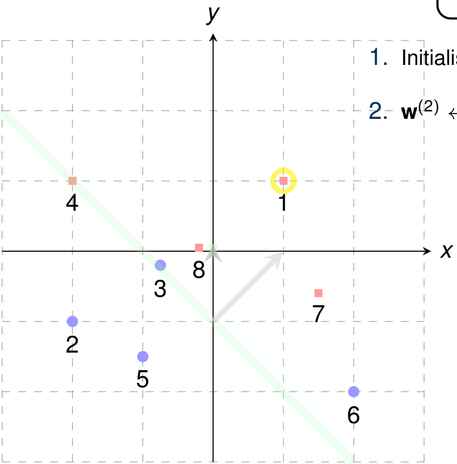



$$\begin{aligned}&\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}\\ &\quad \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i\end{aligned}$$

1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$.

# Illustration of Perceptron (2/2)

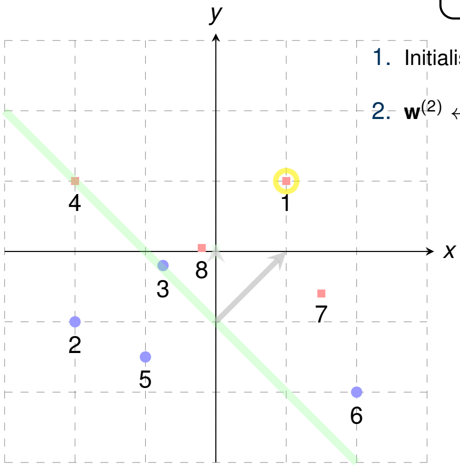
$$\textbf{if } (\exists\, i \;\; \text{s.t.} \;\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$.

# Illustration of Perceptron (2/2)

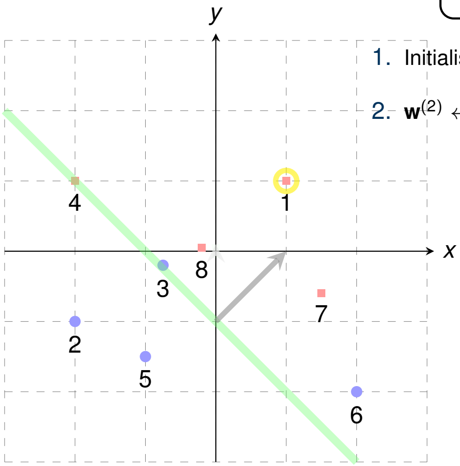
**if** $(\exists\, i \;\; \text{s.t.} \;\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$.

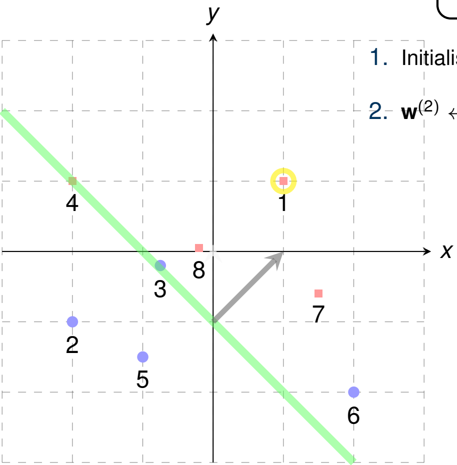
# Illustration of Perceptron (2/2)

**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$.

# Illustration of Perceptron (2/2)

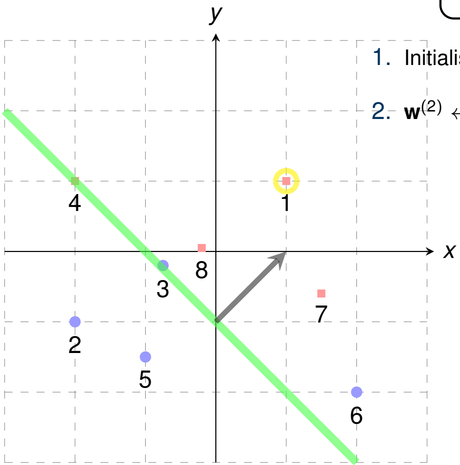



**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$.

# Illustration of Perceptron (2/2)

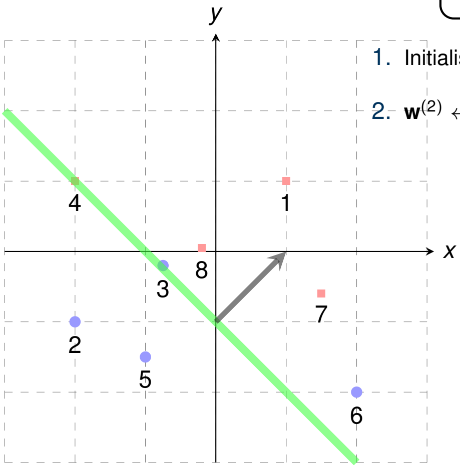
**if** $(\exists\, i \;\; \text{s.t.} \;\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$.

# Illustration of Perceptron (2/2)

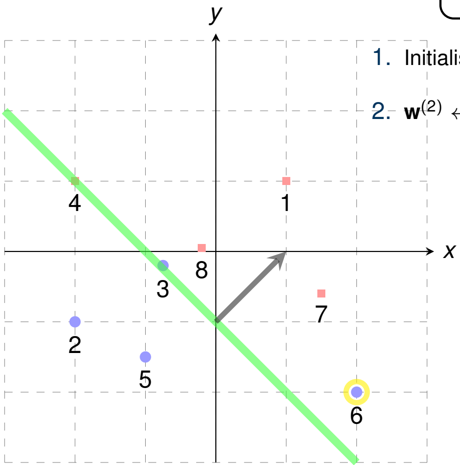
**if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$.

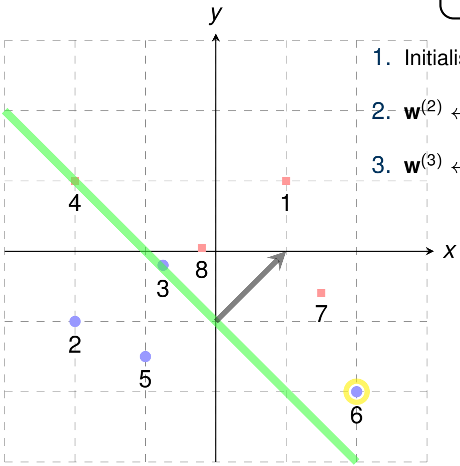
# Illustration of Perceptron (2/2)

$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.

# Illustration of Perceptron (2/2)

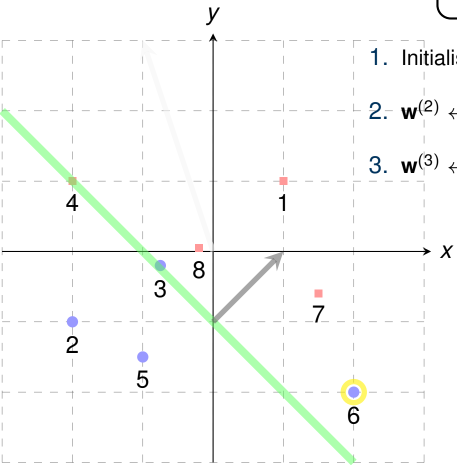



**if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$.

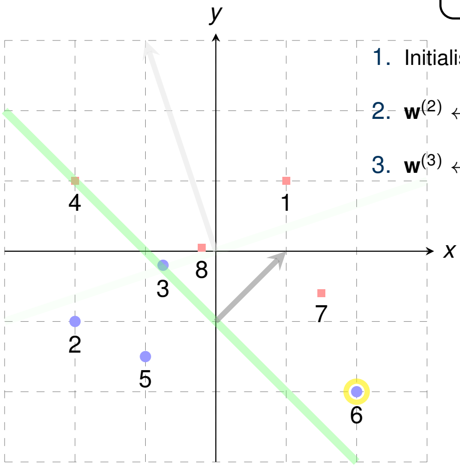
# Illustration of Perceptron (2/2)

**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$.

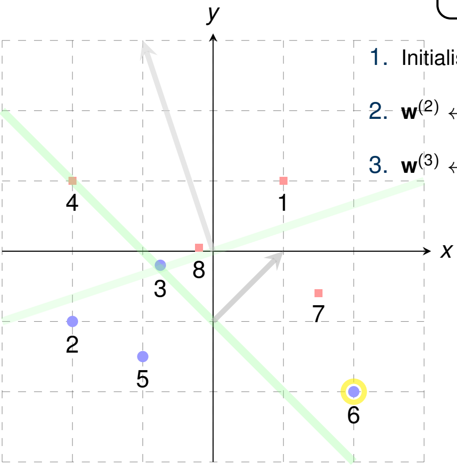
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \;\; \text{s.t.} \;\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$.

# Illustration of Perceptron (2/2)

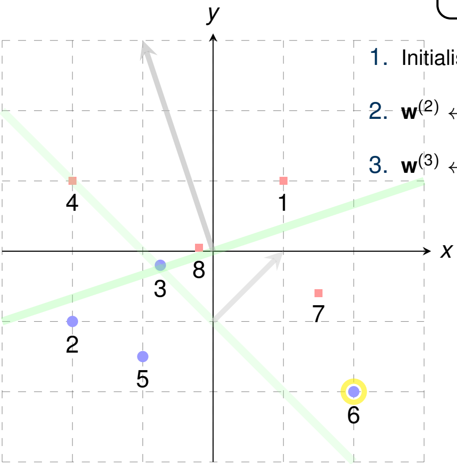



$$\textbf{if } (\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$.

# Illustration of Perceptron (2/2)

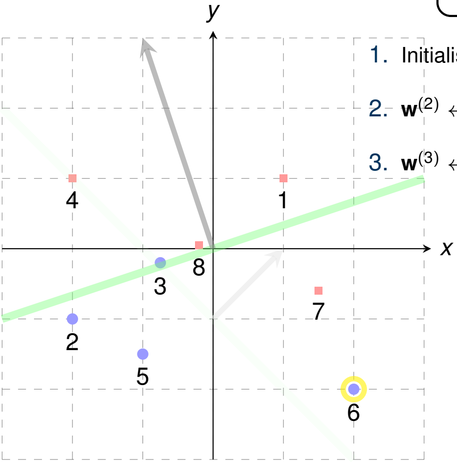
**if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$.

# Illustration of Perceptron (2/2)

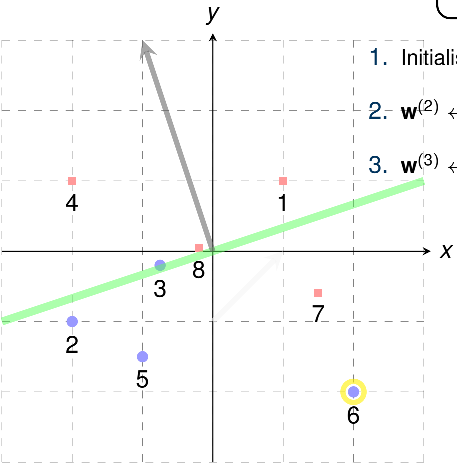
**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$.

# Illustration of Perceptron (2/2)

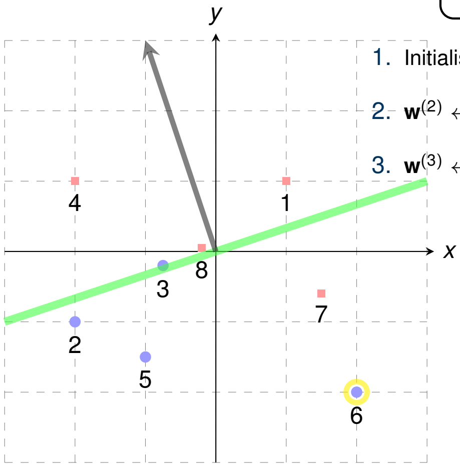
**if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$.

# Illustration of Perceptron (2/2)

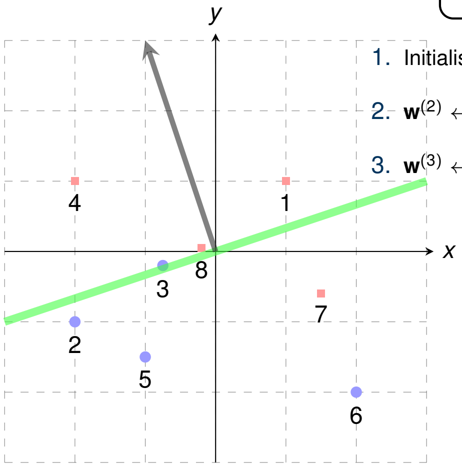
$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$.

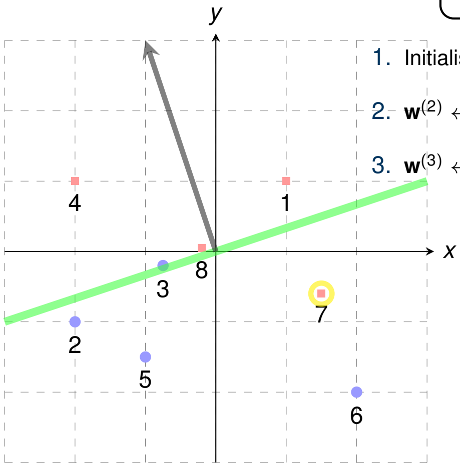
# Illustration of Perceptron (2/2)

if $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$.

# Illustration of Perceptron (2/2)

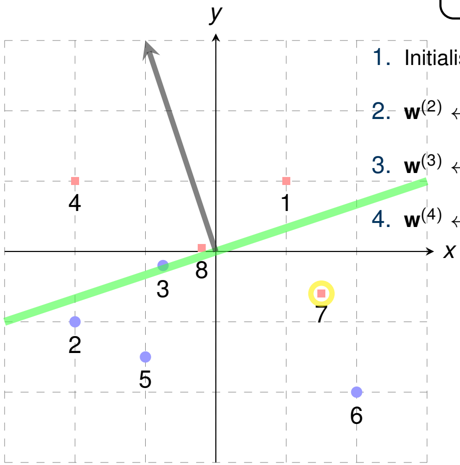



**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.

**Quiz 1:** What is $w^{(4)}$?

# Illustration of Perceptron (2/2)

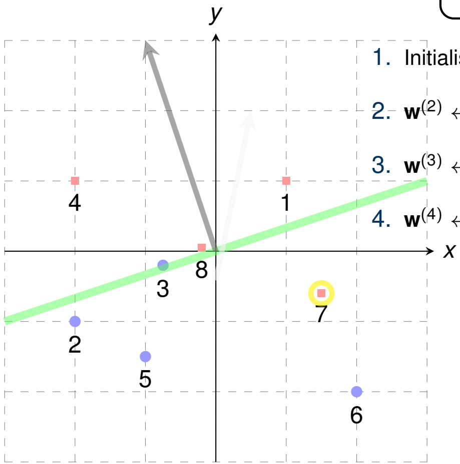



$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

**Quiz 1:** What is $w^{(4)}$?

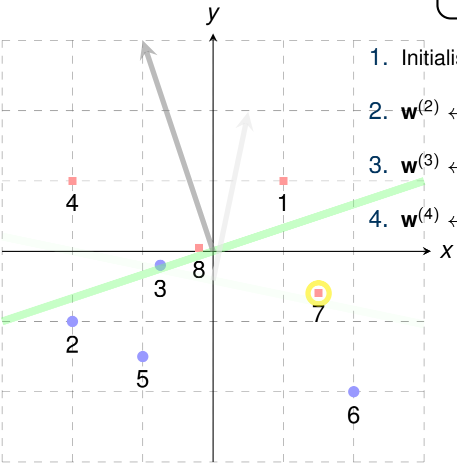
## Illustration of Perceptron (2/2)




> **if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
> $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

# Illustration of Perceptron (2/2)

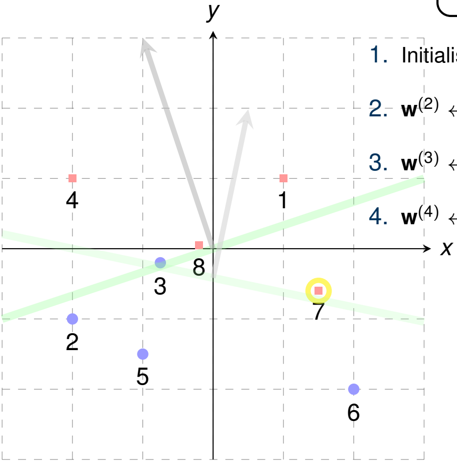



**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

# Illustration of Perceptron (2/2)

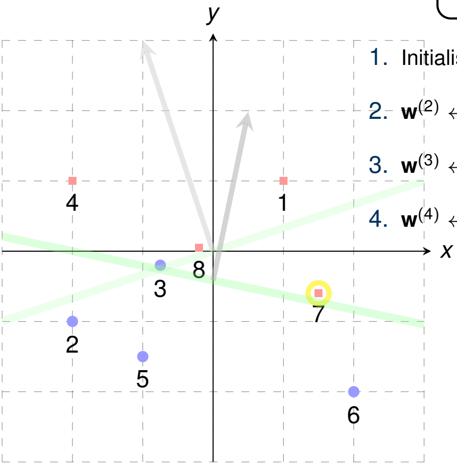
**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$.

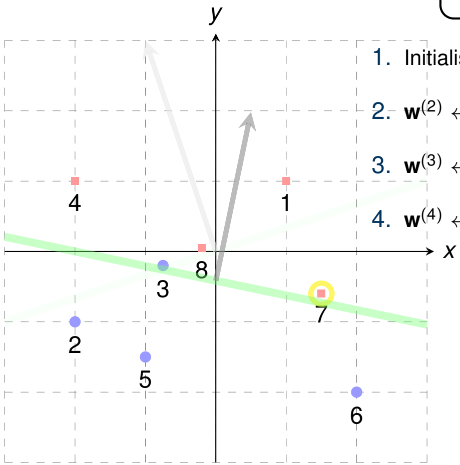
# Illustration of Perceptron (2/2)

$$\textbf{if } (\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

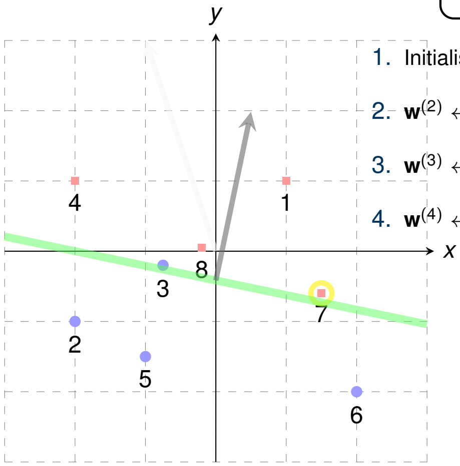
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

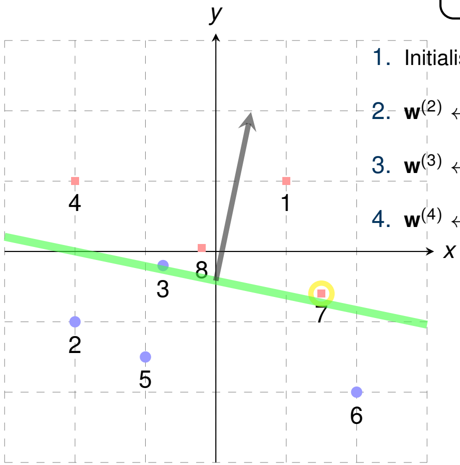
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

# Illustration of Perceptron (2/2)

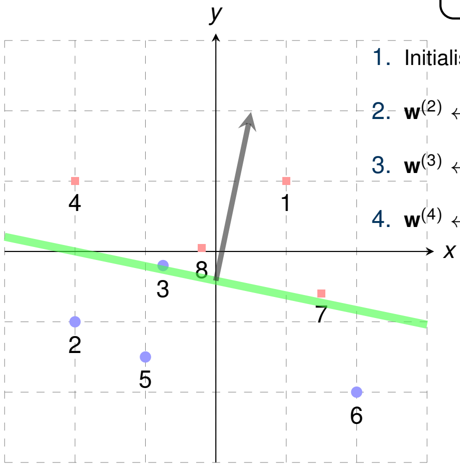



$$\textbf{if } (\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

# Illustration of Perceptron (2/2)

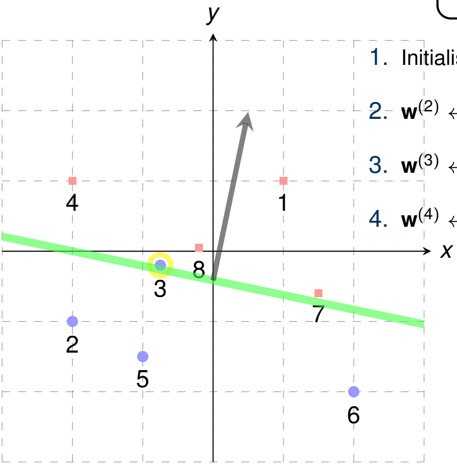
**if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$.

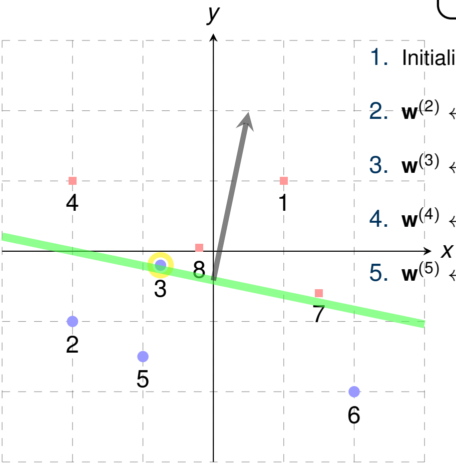
## Illustration of Perceptron (2/2)

$$\begin{aligned}&\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}\\ &\quad \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i\end{aligned}$$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.

# Illustration of Perceptron (2/2)

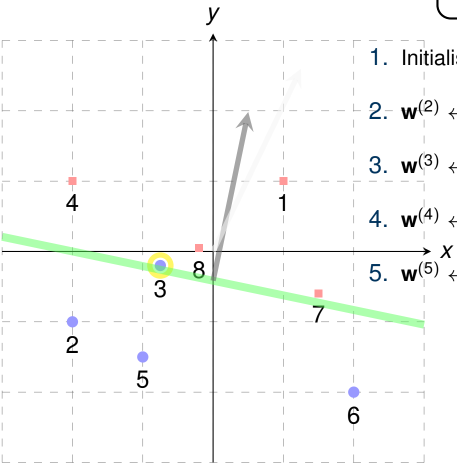



$$\begin{aligned}&\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}\\ &\quad \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i\end{aligned}$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$.

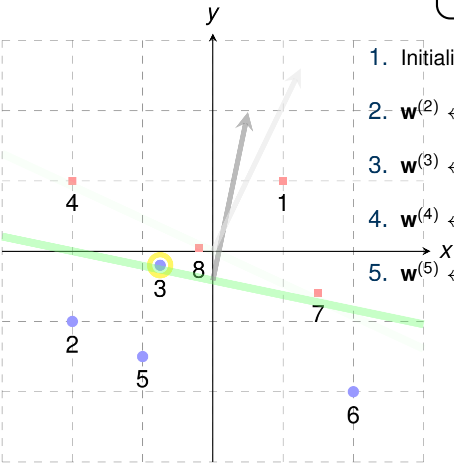
# Illustration of Perceptron (2/2)

**if** $(\exists\, i \;\; \text{s.t.} \;\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$.

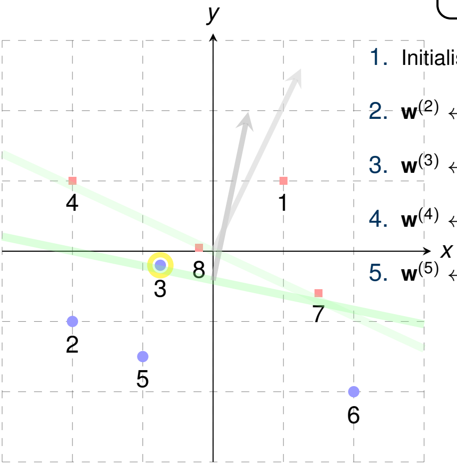
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$.

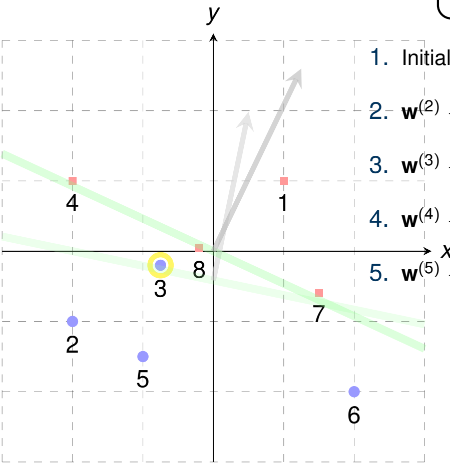
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$.

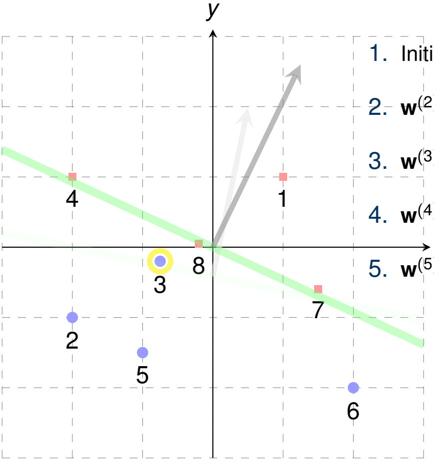
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \le 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$.

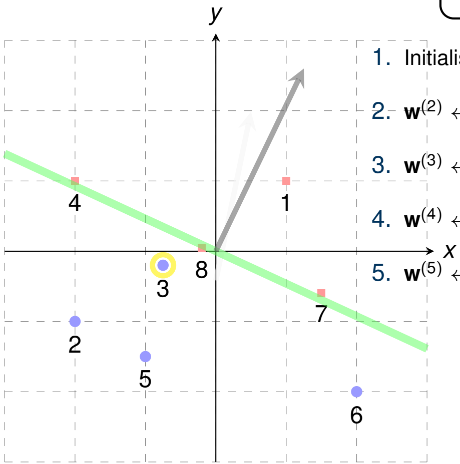
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$.

# Illustration of Perceptron (2/2)

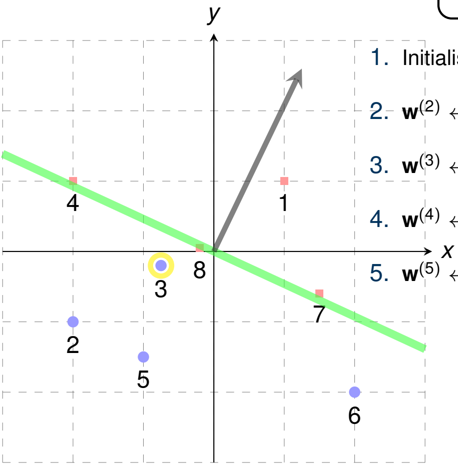



**if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$.

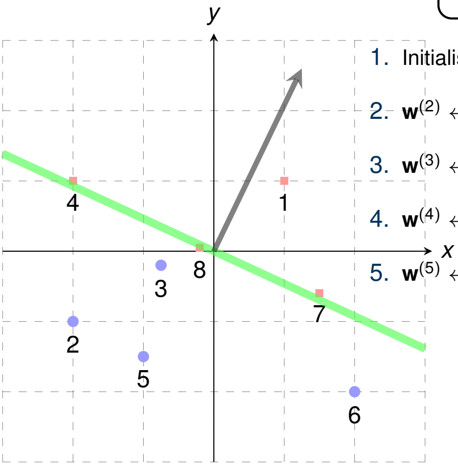
# Illustration of Perceptron (2/2)

$$\textbf{if } (\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \le 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$.

# Illustration of Perceptron (2/2)

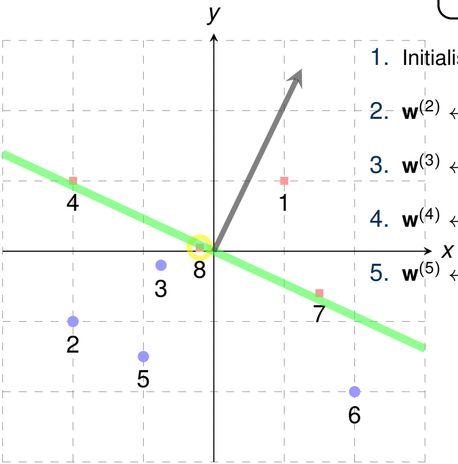



**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$.

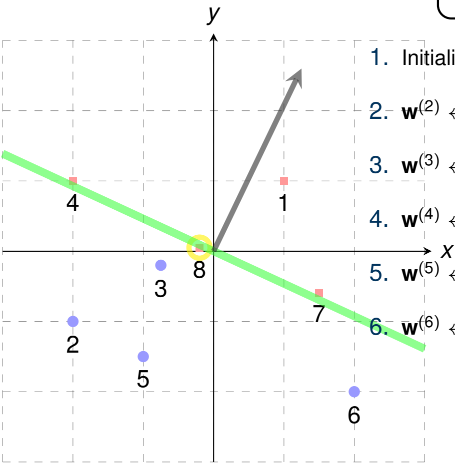
# Illustration of Perceptron (2/2)

**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.

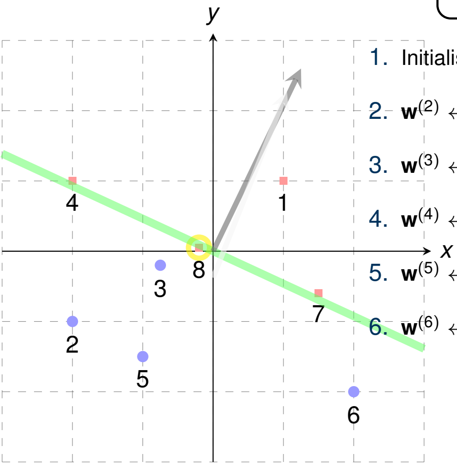
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \;\text{ s.t. }\; y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i\mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

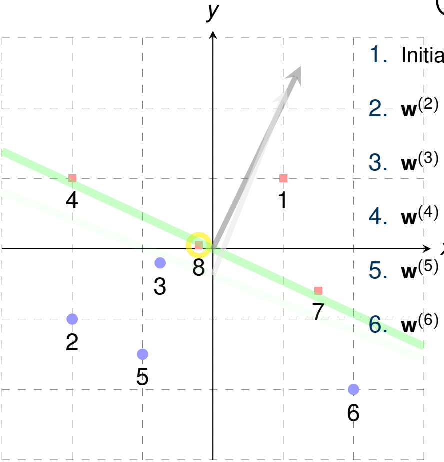
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then} \quad \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

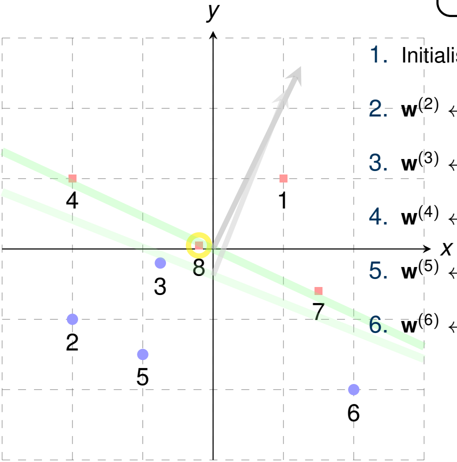
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

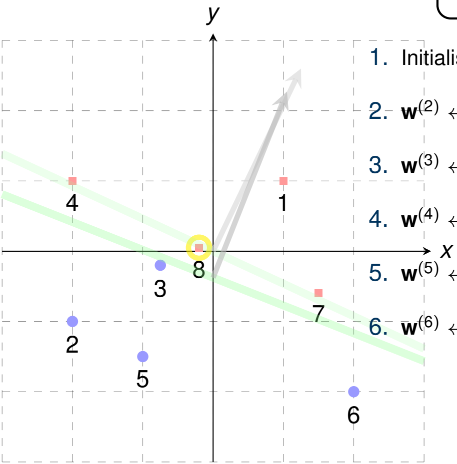
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

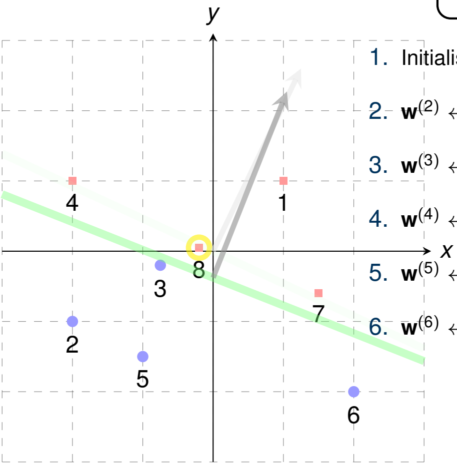
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

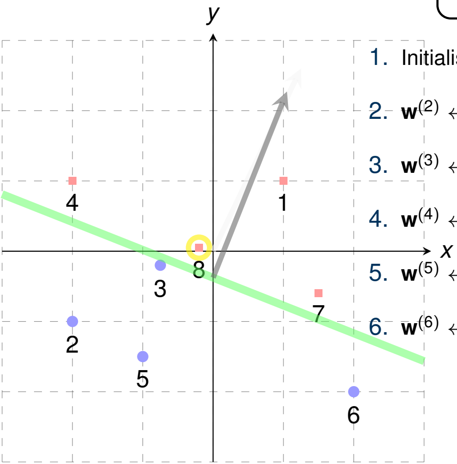
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

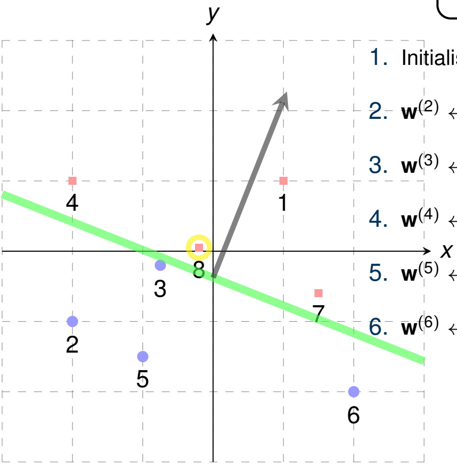
# Illustration of Perceptron (2/2)




**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

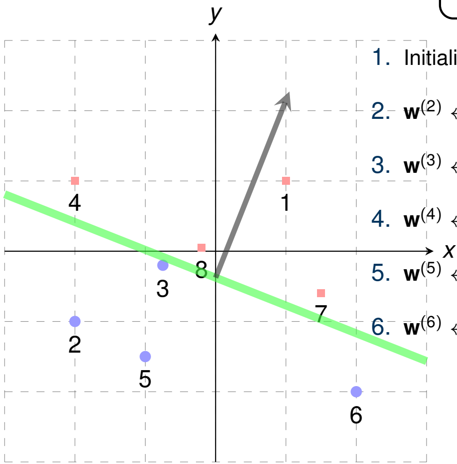
# Illustration of Perceptron (2/2)




$$\textbf{if } (\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \le 0) \text{ then}$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$$

1. Initialise $\mathbf{w}^{(1)} = (0, 0, 0)$. Pick $\mathbf{x}_1 = (1, 1, 1)$
2. $\mathbf{w}^{(2)} \leftarrow (1, 1, 1)$. Pick $\mathbf{x}_6 = (1, 2, -2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0, -1, 3)$. Pick $\mathbf{x}_7 = (1, 1.5, -0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1, 0.5, 2.4)$. $\mathbf{x}_3 = (1, -0.75, -0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0, 1.25, 2.6)$. $\mathbf{x}_8 = (1, -0.2, -0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1, 1.05, 2.65)$.

# Illustration of Perceptron (2/2)

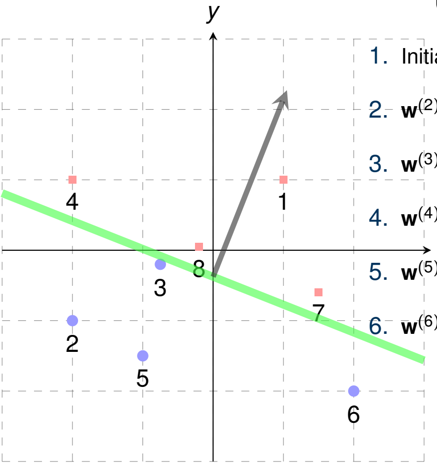



**if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$. $\mathbf{x}_8 = (1,-0.2,-0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1,1.05,2.65)$.

# Illustration of Perceptron (2/2)

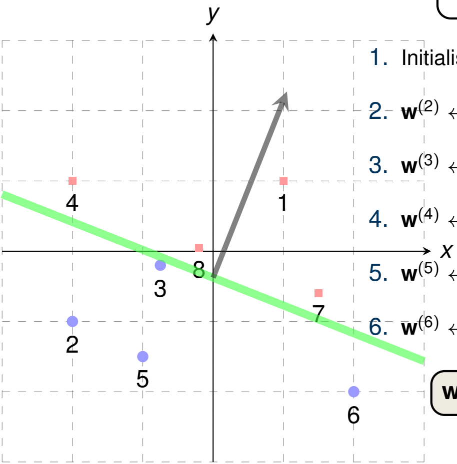
**if** $(\exists\, i$ s.t. $y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$




1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$. $\mathbf{x}_8 = (1,-0.2,-0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1,1.05,2.65)$. **Finished!**

# Illustration of Perceptron (2/2)

**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \le 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

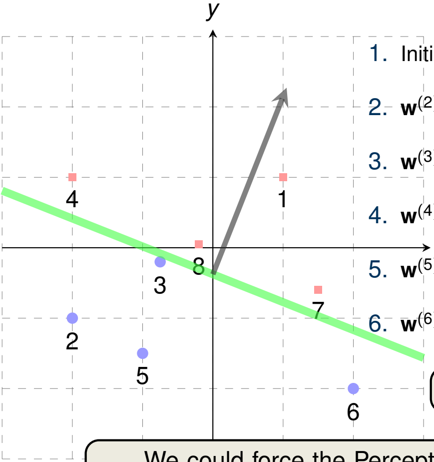



1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$. $\mathbf{x}_8 = (1,-0.2,-0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1,1.05,2.65)$. **Finished!**

**w** sums over misclassified points!

# Illustration of Perceptron (2/2)

**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$



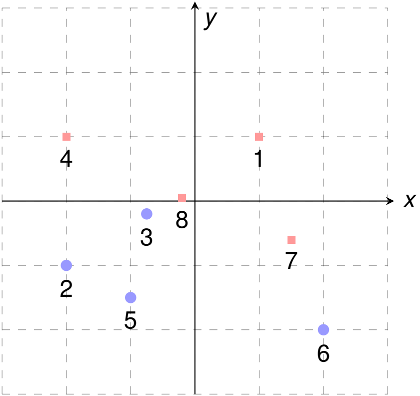

1. Initialise $\mathbf{w}^{(1)} = (0,0,0)$. Pick $\mathbf{x}_1 = (1,1,1)$
2. $\mathbf{w}^{(2)} \leftarrow (1,1,1)$. Pick $\mathbf{x}_6 = (1,2,-2)$.
3. $\mathbf{w}^{(3)} \leftarrow (0,-1,3)$. Pick $\mathbf{x}_7 = (1,1.5,-0.6)$.
4. $\mathbf{w}^{(4)} \leftarrow (1,0.5,2.4)$. $\mathbf{x}_3 = (1,-0.75,-0.2)$.
5. $\mathbf{w}^{(5)} \leftarrow (0,1.25,2.6)$. $\mathbf{x}_8 = (1,-0.2,-0.05)$.
6. $\mathbf{w}^{(6)} \leftarrow (1,1.05,2.65)$. **Finished!**

$\mathbf{w}$ sums over misclassified points!

We could force the Perceptron to run forever by adding more and more points near the green line!

# Intuition of the Perceptron Algorithm

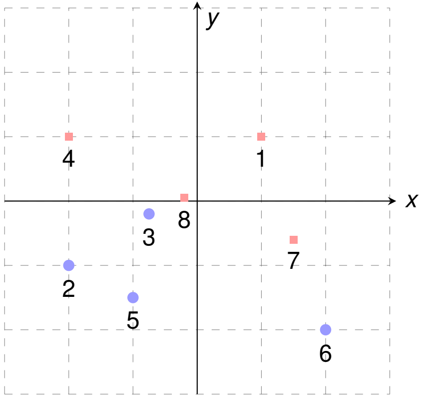
Why does the Perceptron Algorithm make progress?



# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

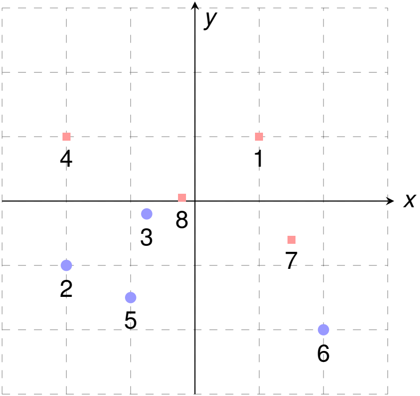


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle =$$

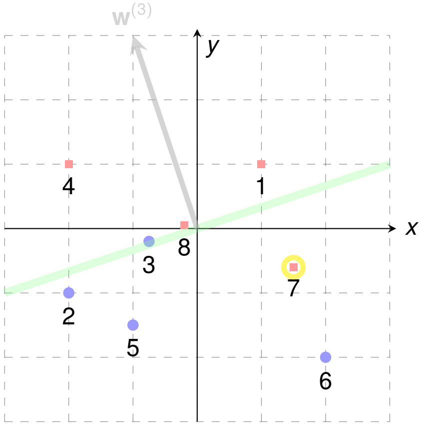


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle =$$

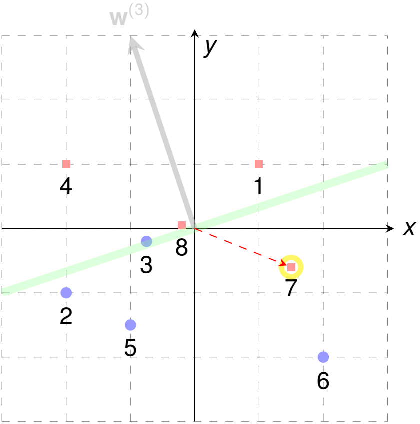


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle =$$

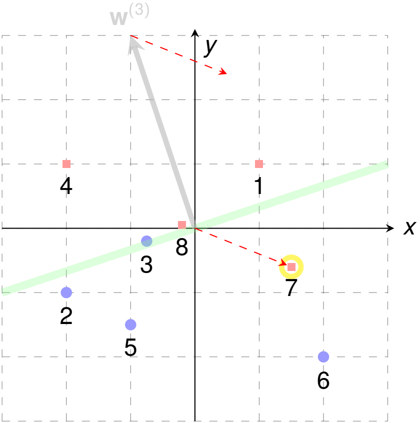


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle =$$

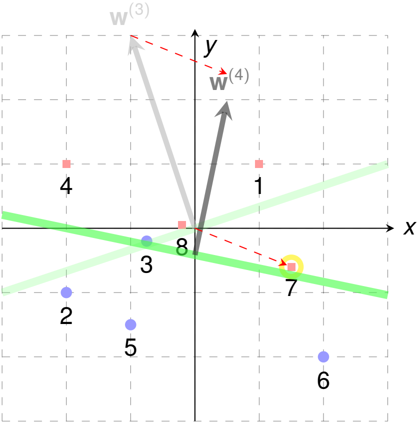


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle = y_i \langle \mathbf{w}^{(t)} + y_i \mathbf{x}_i, \mathbf{x}_i \rangle =$$

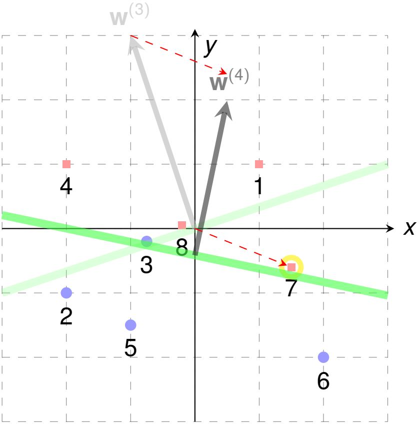


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle = y_i \langle \mathbf{w}^{(t)} + y_i \mathbf{x}_i, \mathbf{x}_i \rangle = y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 .$$

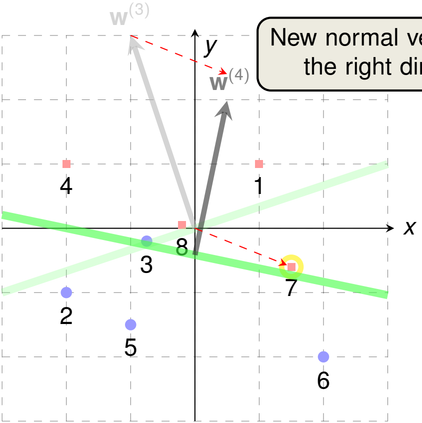


# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i \langle \mathbf{w}^{(t+1)}, \mathbf{x}_i \rangle = y_i \langle \mathbf{w}^{(t)} + y_i \mathbf{x}_i, \mathbf{x}_i \rangle = y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 .$$



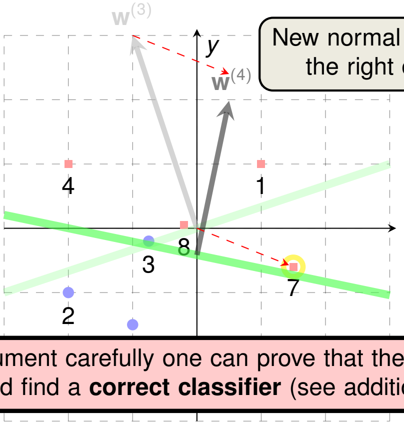

New normal vector is guided towards the right direction regarding $\mathbf{x}_i$!

# Intuition of the Perceptron Algorithm

Why does the Perceptron Algorithm make progress?

The Perceptron guides the solution to be "more correct" on the *i*-th example:

$$y_i\langle \mathbf{w}^{(t+1)}, \mathbf{x}_i\rangle = y_i\langle \mathbf{w}^{(t)} + y_i\mathbf{x}_i, \mathbf{x}_i\rangle = y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle + \|\mathbf{x}_i\|^2 .$$




New normal vector is guided towards the right direction regarding $\mathbf{x}_i$!

Using this argument carefully one can prove that the Perceptron will **terminate** and find a **correct classifier** (see additional material)!

# Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman “Mining of Massive Data Sets”):

# Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman “Mining of Massive Data Sets”):

| $i$ | “and” | “offer” | “the” | “of” | “sale” | $y_i$ |
|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 1 | 1 | $+1$ pos. |
| $\mathbf{x}_2$ | 0 | 0 | 1 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_3$ | 0 | 1 | 1 | 0 | 0 | $+1$ pos. |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 | $+1$ pos. |
| $\mathbf{x}_6$ | 1 | 0 | 1 | 1 | 0 | $-1$ neg. |

# Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman "Mining of Massive Data Sets"):

$y_i = +1$ means SPAM and
$y_i = -1$ means NON-SPAM.

| $i$ | "and" | "offer" | "the" | "of" | "sale" | $y_i$ |
|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 1 | 1 | $+1$ pos. |
| $\mathbf{x}_2$ | 0 | 0 | 1 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_3$ | 0 | 1 | 1 | 0 | 0 | $+1$ pos. |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 | $+1$ pos. |
| $\mathbf{x}_6$ | 1 | 0 | 1 | 1 | 0 | $-1$ neg. |

# Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman "Mining of Massive Data Sets"):

$y_i = +1$ means SPAM and
$y_i = -1$ means NON-SPAM.

| $i$ | "and" | "offer" | "the" | "of" | "sale" | $y_i$ |
|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 1 | 1 | $+1$ pos. |
| $\mathbf{x}_2$ | 0 | 0 | 1 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_3$ | 0 | 1 | 1 | 0 | 0 | $+1$ pos. |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 | $+1$ pos. |
| $\mathbf{x}_6$ | 1 | 0 | 1 | 1 | 0 | $-1$ neg. |

- Run Perceptron and update normal vector using data points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ in that order

# Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman "Mining of Massive Data Sets"):

$y_i = +1$ means SPAM and
$y_i = -1$ means NON-SPAM.

| $i$ | "and" | "offer" | "the" | "of" | "sale" | $y_i$ |
|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 1 | 1 | $+1$ pos. |
| $\mathbf{x}_2$ | 0 | 0 | 1 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_3$ | 0 | 1 | 1 | 0 | 0 | $+1$ pos. |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 | $+1$ pos. |
| $\mathbf{x}_6$ | 1 | 0 | 1 | 1 | 0 | $-1$ neg. |

- Run Perceptron and update normal vector using data points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ in that order
- Final normal vector of hyperplane is:

$$\mathbf{w}^{(5)} = (0, 2, 0, -1, 1),$$

and bias is 0

## Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman "Mining of Massive Data Sets"):

$y_i = +1$ means SPAM and
$y_i = -1$ means NON-SPAM.

| $i$ | "and" | "offer" | "the" | "of" | "sale" | $y_i$ |
|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 1 | 1 | $+1$ pos. |
| $\mathbf{x}_2$ | 0 | 0 | 1 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_3$ | 0 | 1 | 1 | 0 | 0 | $+1$ pos. |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 | $+1$ pos. |
| $\mathbf{x}_6$ | 1 | 0 | 1 | 1 | 0 | $-1$ neg. |

- Run Perceptron and update normal vector using data points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ in that order
- Final normal vector of hyperplane is:

$$\mathbf{w}^{(5)} = (0, 2, 0, -1, 1),$$

and bias is 0

Interpretation: The words "offer" and "sale" are indicative of SPAM, while the word "of" is indicative of non-SPAM. The other words are neutral.

# Example: Perceptron for Text Classification

Consider the following example of a training set for spam classification

(based on a set taken from Leskovec, Rajaraman, Ullman "Mining of Massive Data Sets"):

$y_i = +1$ means SPAM and
$y_i = -1$ means NON-SPAM.

| $i$ | "and" | "offer" | "the" | "of" | "sale" | $y_i$ |
|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 1 | 1 | $+1$ pos. |
| $\mathbf{x}_2$ | 0 | 0 | 1 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_3$ | 0 | 1 | 1 | 0 | 0 | $+1$ pos. |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | $-1$ neg. |
| $\mathbf{x}_5$ | 1 | 0 | 1 | 0 | 1 | $+1$ pos. |
| $\mathbf{x}_6$ | 1 | 0 | 1 | 1 | 0 | $-1$ neg. |

- Run Perceptron and update normal vector using data points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ in that order
- Final normal vector of hyperplane is:

**Quiz 2:** Classify email that contains words "buy", "offer", "the" and "of"

$$\mathbf{w}^{(5)} = (0, 2, 0, -1, 1),$$

and bias is 0

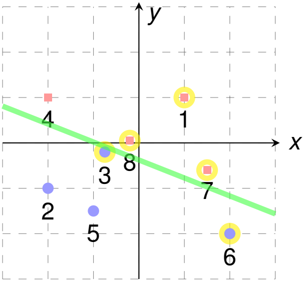
Interpretation: The words "offer" and "sale" are indicative of SPAM, while the word "of" is indicative of non-SPAM. The other words are neutral.

# Outline

# Problems of the Perceptron

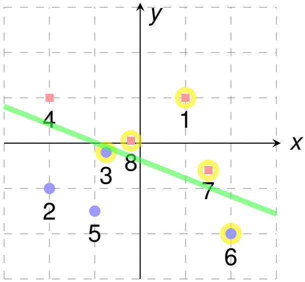
- Convergence may be slow (depends on geometry of point set)

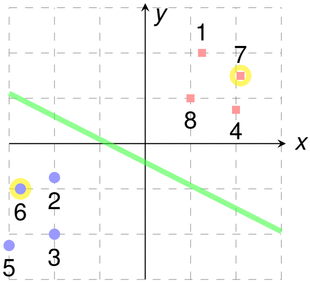
# Problems of the Perceptron




- Convergence may be slow (depends on geometry of point set)
  - small margin $\rightsquigarrow$ potentially slow convergence

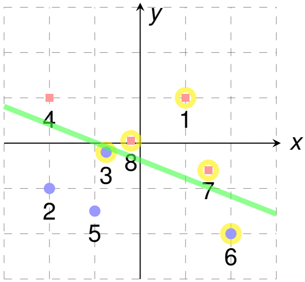
# Problems of the Perceptron

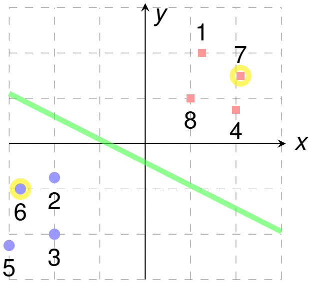



- Convergence may be slow (depends on geometry of point set)
  - small margin $\rightsquigarrow$ potentially slow convergence
  - large margin $\rightsquigarrow$ guaranteed quick convergence

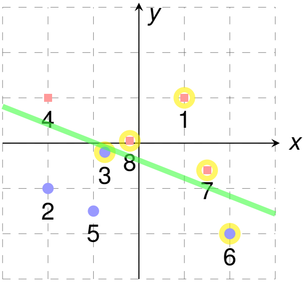
# Problems of the Perceptron

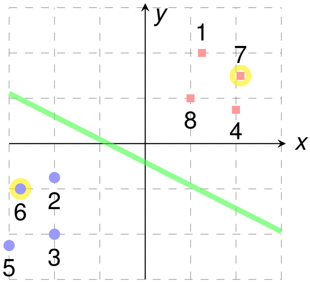



- Convergence may be slow (depends on geometry of point set)
  - small margin $\rightsquigarrow$ potentially slow convergence
  - large margin $\rightsquigarrow$ guaranteed quick convergence
- Heuristic: Update with a random misclassified point (computing furthest or closest points not clear whether it helps and takes too much computation!)

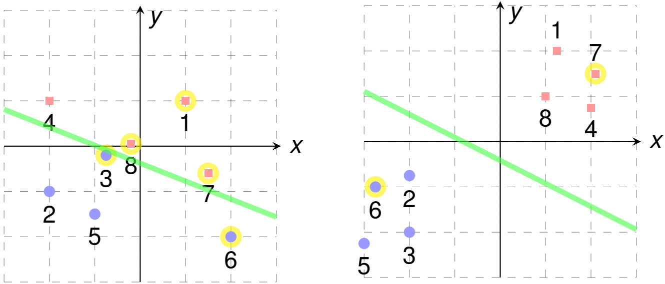
# Problems of the Perceptron



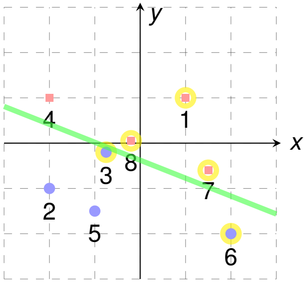

- Convergence may be slow (depends on geometry of point set)
  - small margin $\rightsquigarrow$ potentially slow convergence
  - large margin $\rightsquigarrow$ guaranteed quick convergence
- Heuristic: Update with a random misclassified point (computing furthest or closest points not clear whether it helps and takes too much computation!)
- **Big Problem:** Perceptron cannot handle data that is not linearly separable

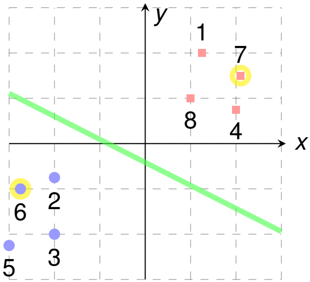
# Problems of the Perceptron




- Convergence may be slow (depends on geometry of point set)
  - small margin $\rightsquigarrow$ potentially slow convergence
  - large margin $\rightsquigarrow$ guaranteed quick convergence
- Heuristic: Update with a random misclassified point (computing furthest or closest points not clear whether it helps and takes too much computation!)
- **Big Problem:** Perceptron cannot handle data that is not linearly separable
  - if points are almost linearly separable, could use Average or Voted Perceptron (idea: store successful classifier like in Stochastic Gradient Descent)

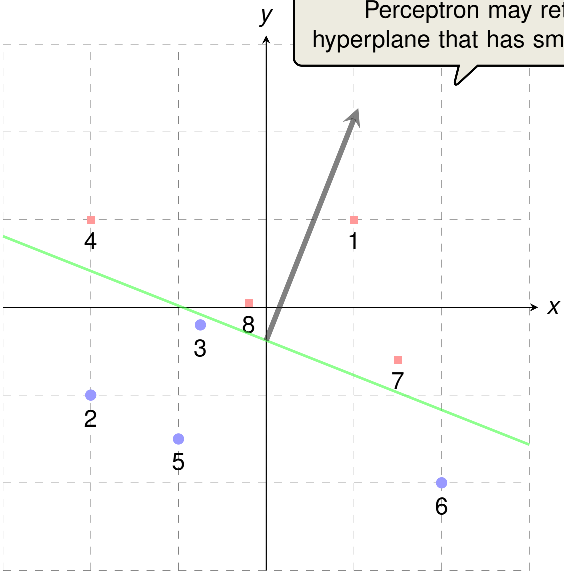
# Problems of the Perceptron



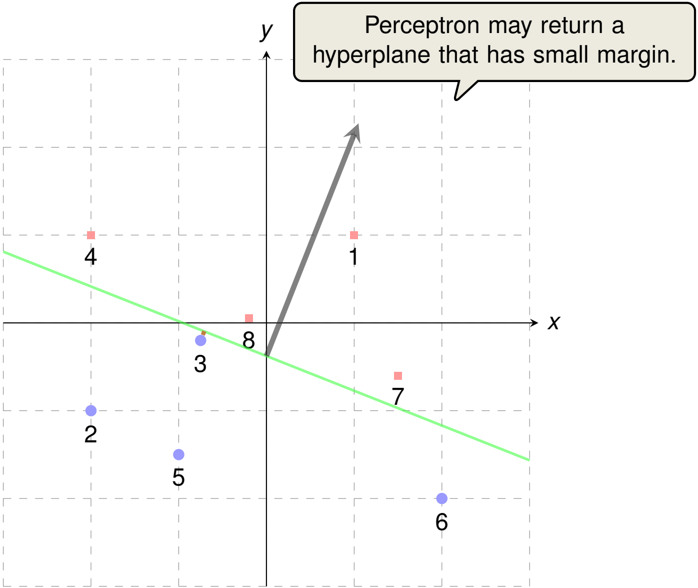

- Convergence may be slow (depends on geometry of point set)
  - small margin $\rightsquigarrow$ potentially slow convergence
  - large margin $\rightsquigarrow$ guaranteed quick convergence
- Heuristic: Update with a random misclassified point (computing furthest or closest points not clear whether it helps and takes too much computation!)
- **Big Problem:** Perceptron cannot handle data that is not linearly separable
  - if points are almost linearly separable, could use Average or Voted Perceptron (idea: store successful classifier like in Stochastic Gradient Descent)
  - if points are far from linearly separable, better use other methods like SVM, SVM with Kernels or Neutral Networks

# Problem 1: Small Margin

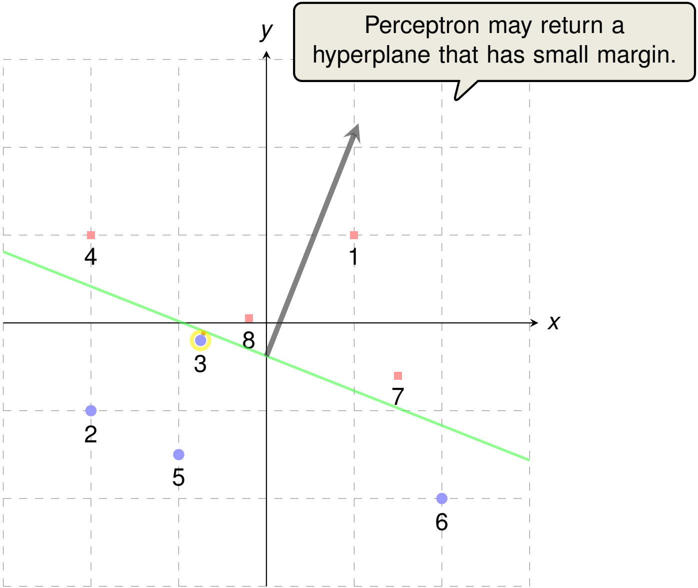



Perceptron may return a hyperplane that has small margin.

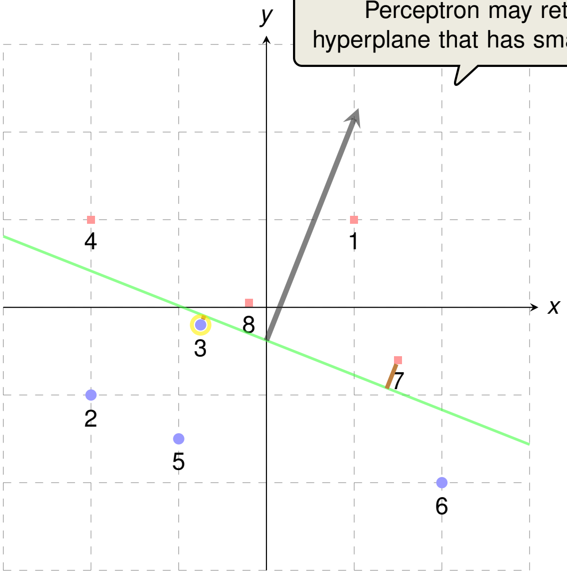
# Problem 1: Small Margin



# Problem 1: Small Margin

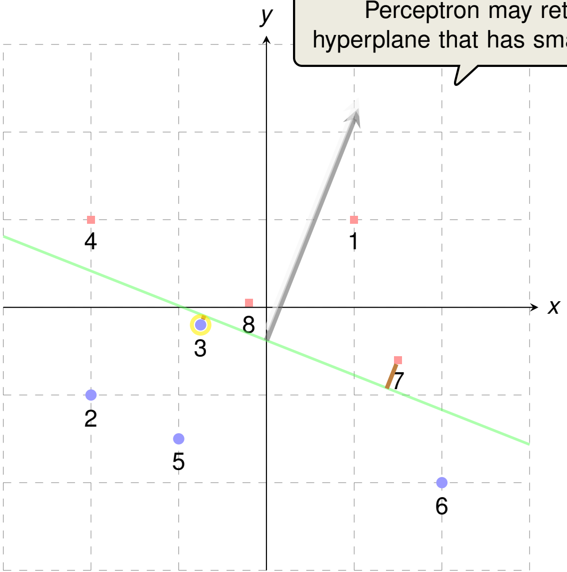


## Problem 1: Small Margin



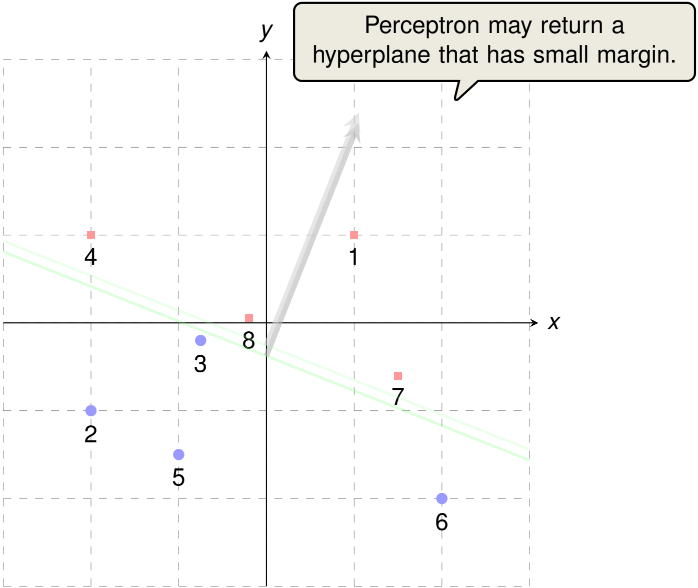

Perceptron may return a hyperplane that has small margin.

# Problem 1: Small Margin

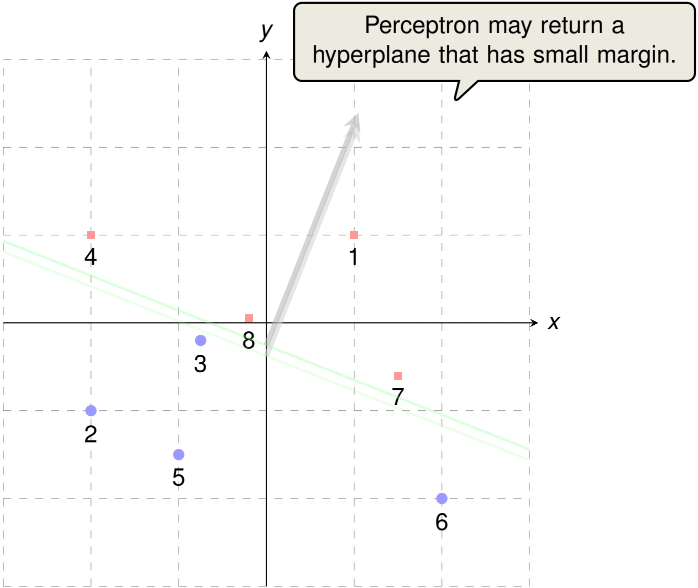



Perceptron may return a hyperplane that has small margin.

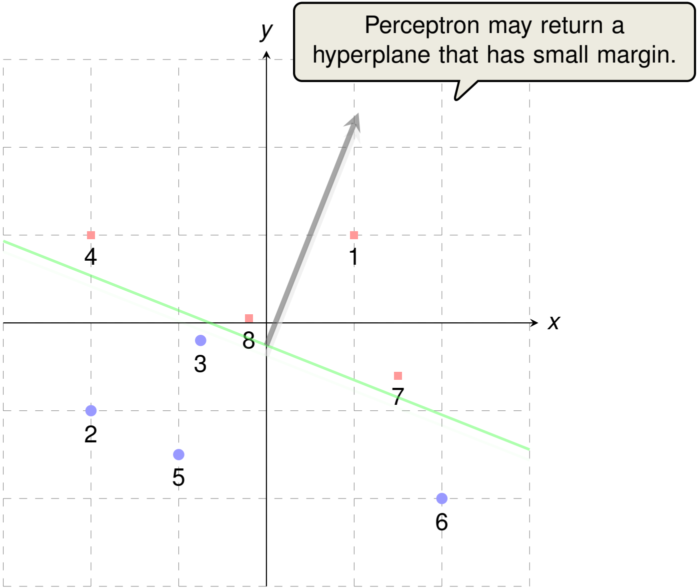
# Problem 1: Small Margin



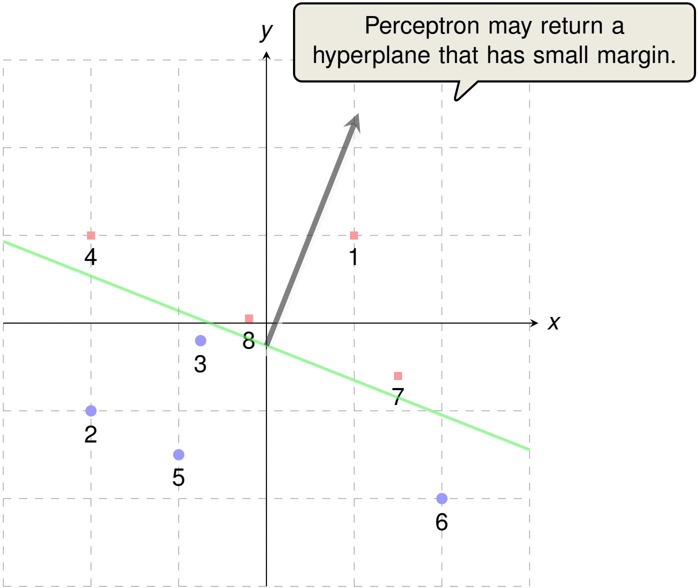
# Problem 1: Small Margin



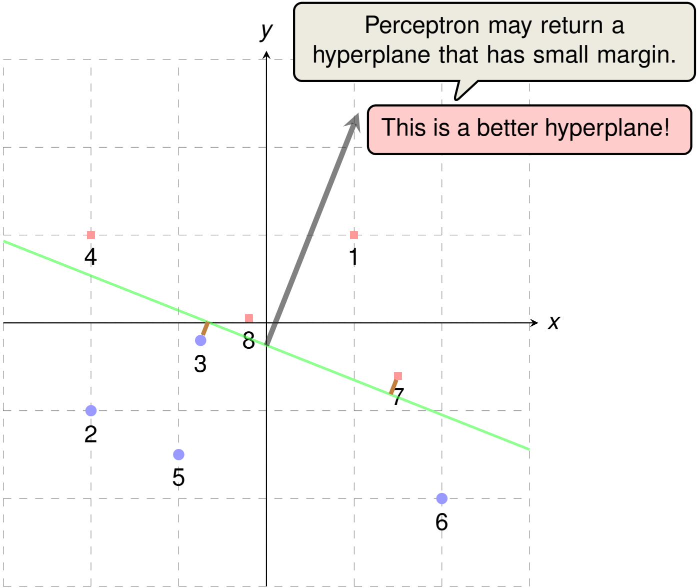
# Problem 1: Small Margin



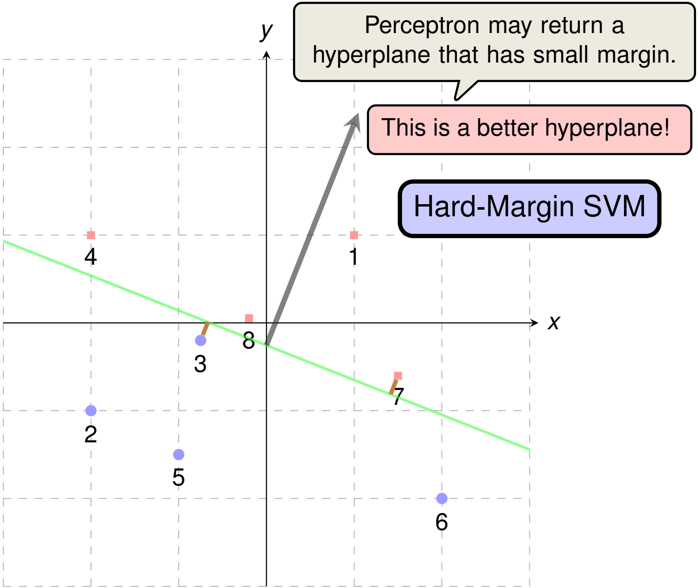
## Problem 1: Small Margin



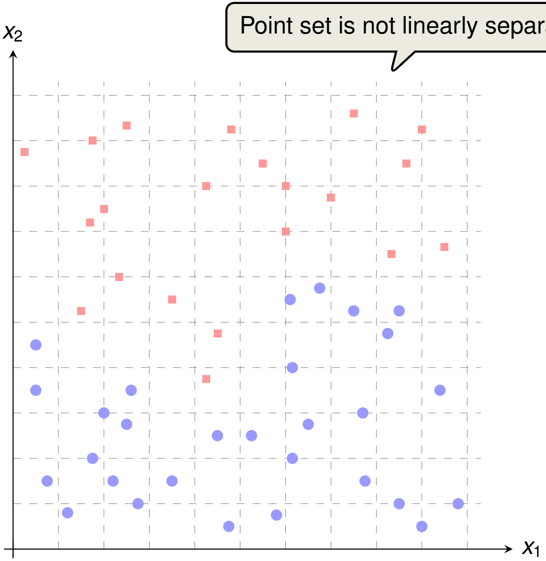
## Problem 1: Small Margin



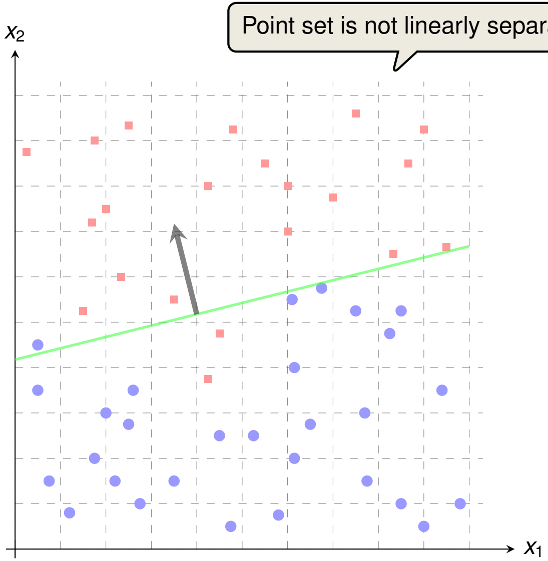
## Problem 1: Small Margin



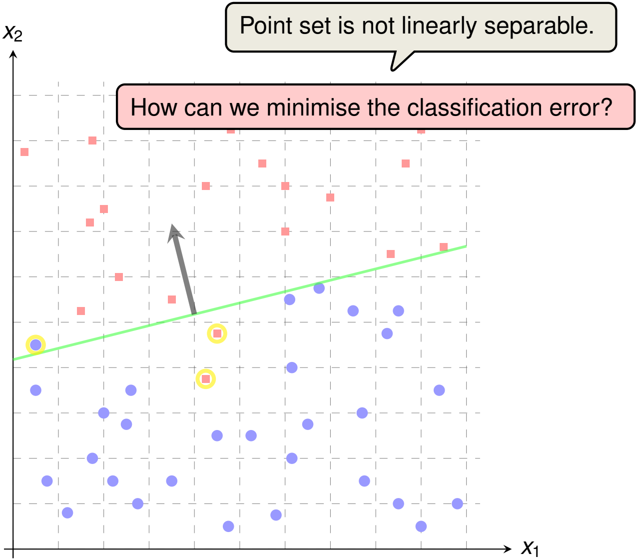
# Problem 2: Linear Separation not (completely) possible



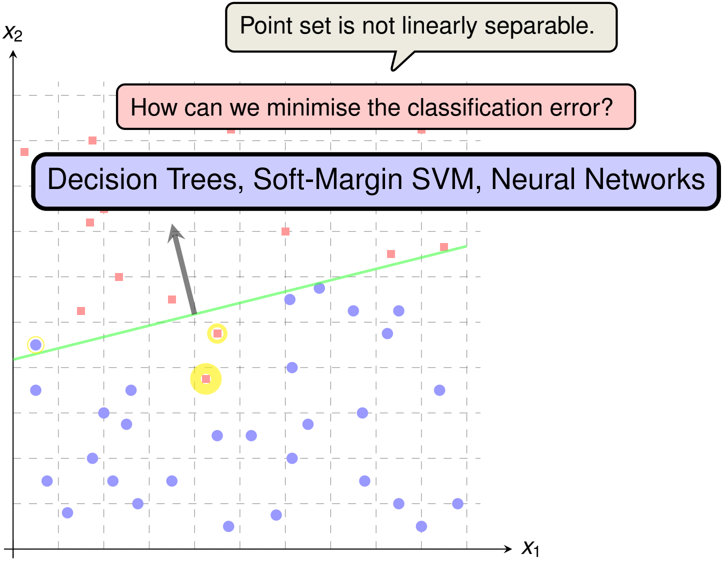
## Problem 2: Linear Separation not (completely) possible



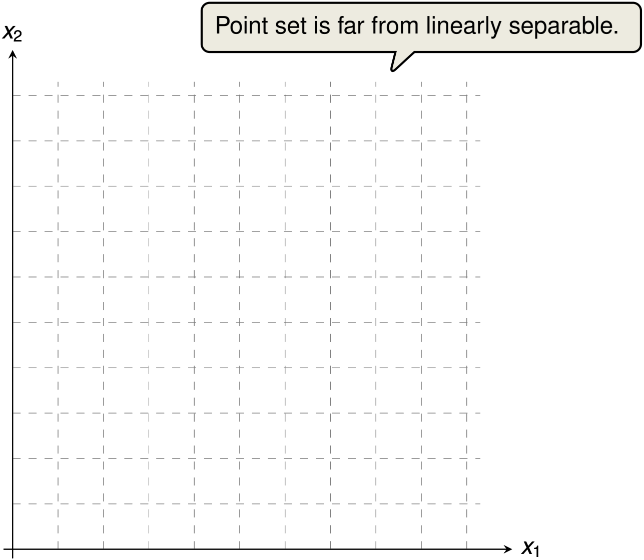
# Problem 2: Linear Separation not (completely) possible



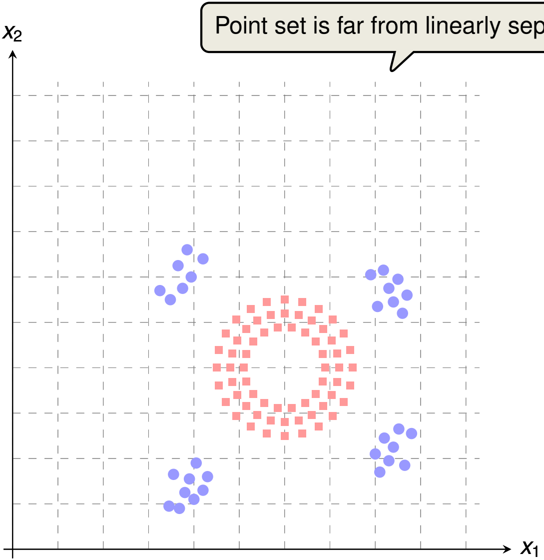
# Problem 2: Linear Separation not (completely) possible



# Problem 3: Linear Separation not possible at all!



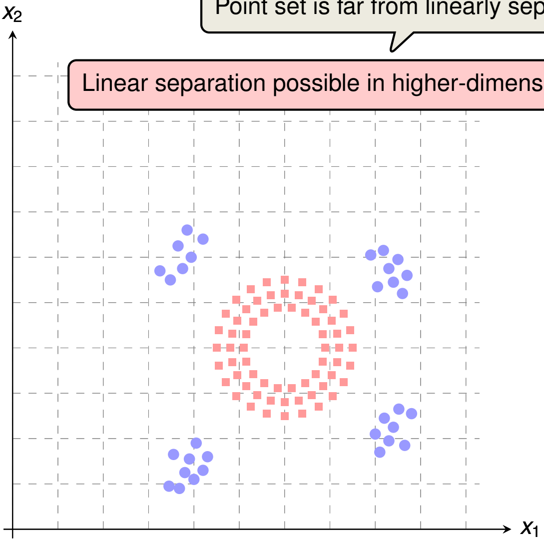
## Problem 3: Linear Separation not possible at all!



# Problem 3: Linear Separation not possible at all!

Point set is far from linearly separable.

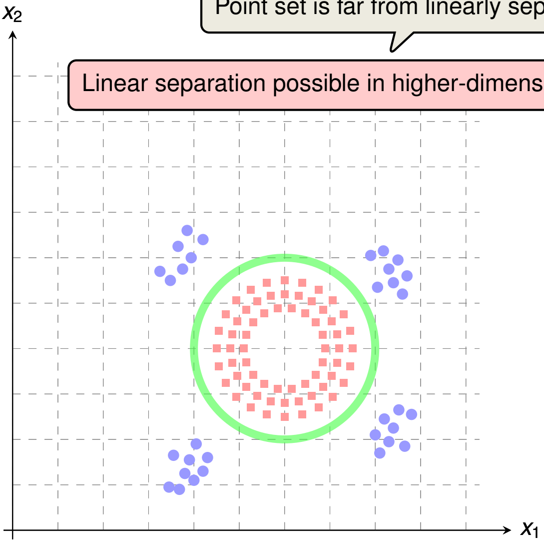
Linear separation possible in higher-dimensional feature space!



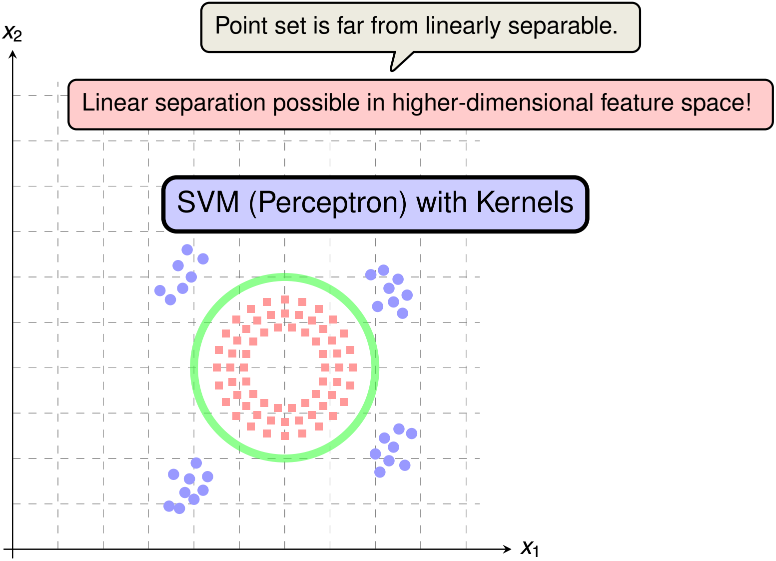
# Problem 3: Linear Separation not possible at all!

Point set is far from linearly separable.

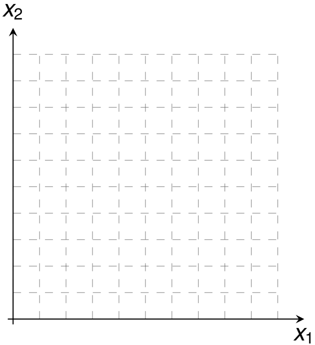
Linear separation possible in higher-dimensional feature space!



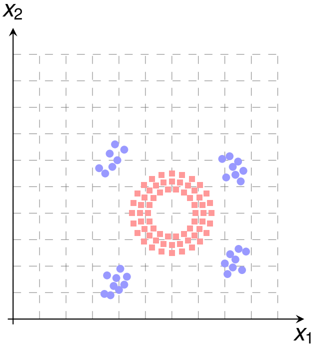
# Problem 3: Linear Separation not possible at all!



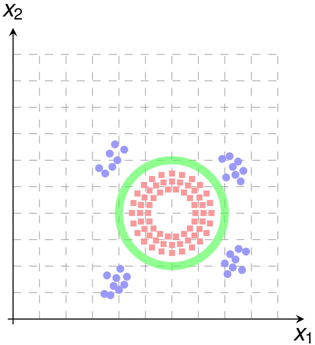
# A Glimpse at the The Kernel Method



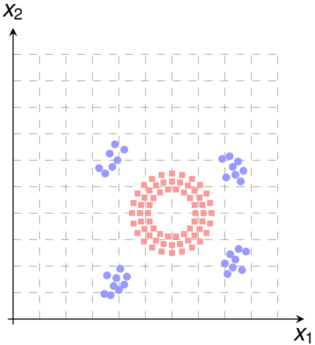
# A Glimpse at the The Kernel Method



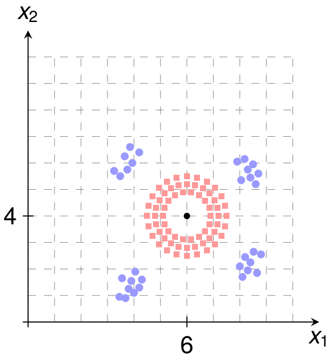
# A Glimpse at the The Kernel Method



# A Glimpse at the The Kernel Method

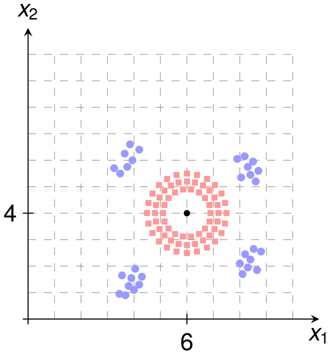


# A Glimpse at the The Kernel Method



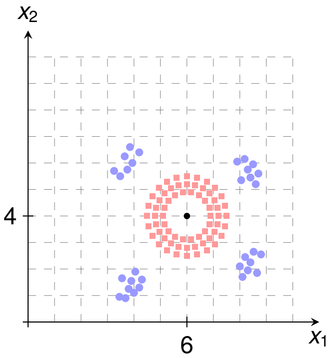
# A Glimpse at the The Kernel Method

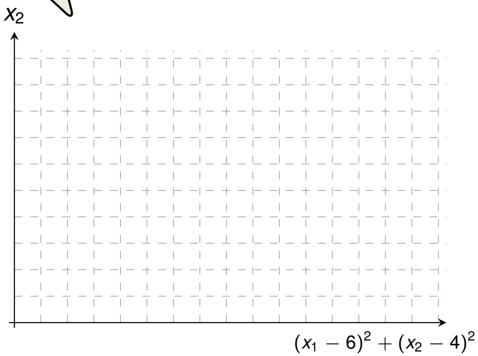
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

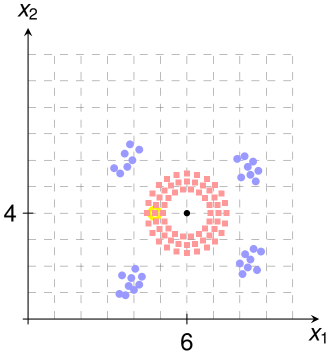
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

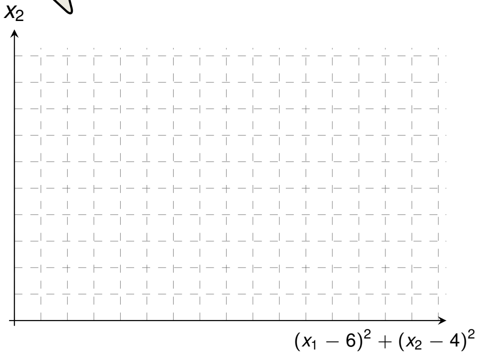


# A Glimpse at the The Kernel Method

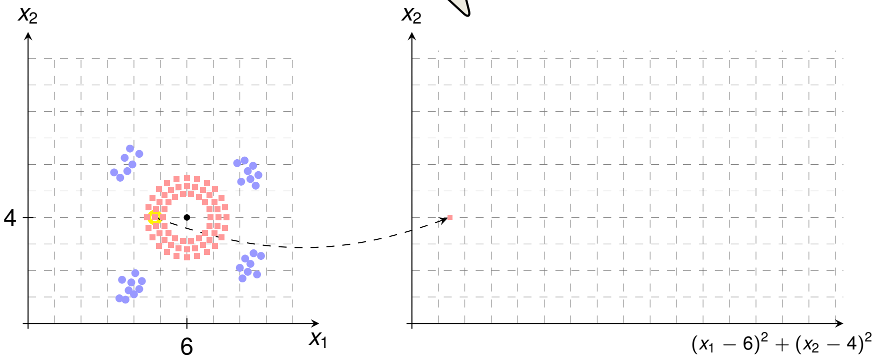
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



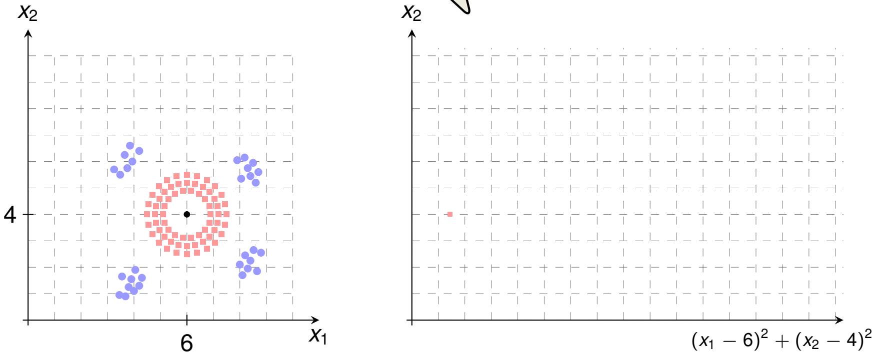
# A Glimpse at the The Kernel Method

new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

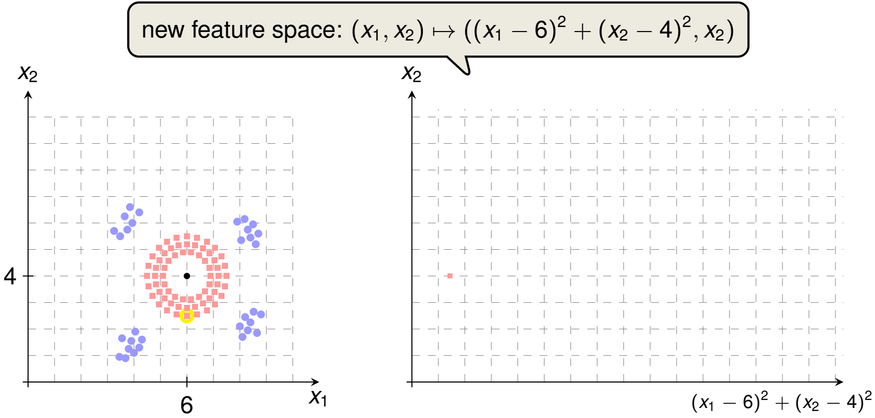


# A Glimpse at the The Kernel Method

new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

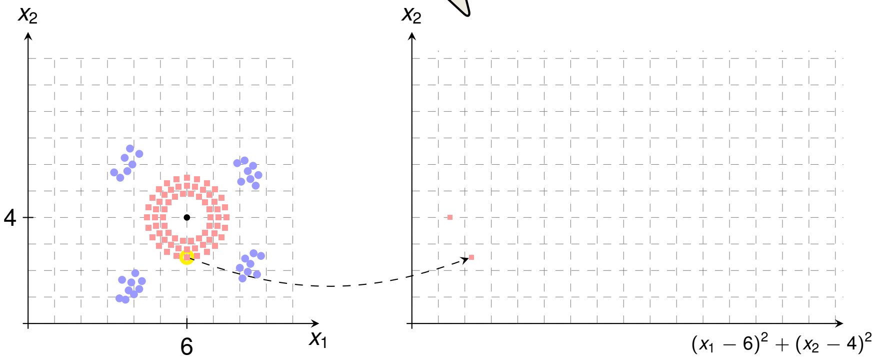


# A Glimpse at the The Kernel Method



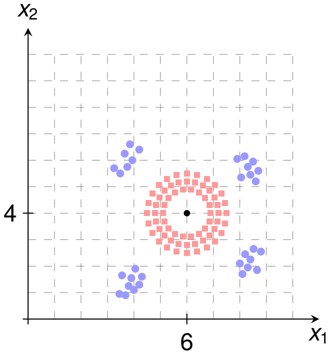
# A Glimpse at the The Kernel Method

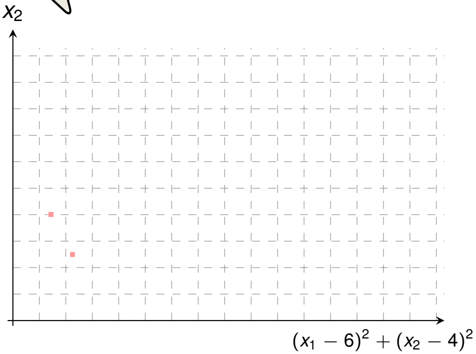
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

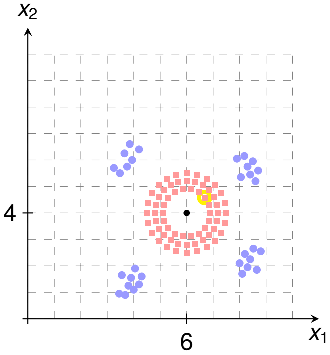
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

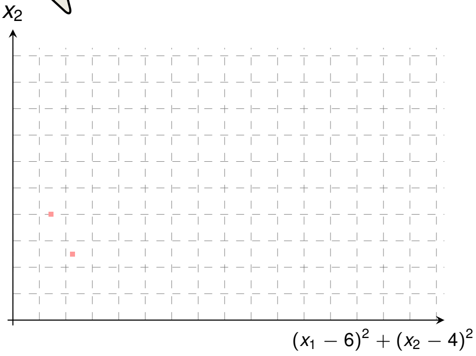


# A Glimpse at the The Kernel Method

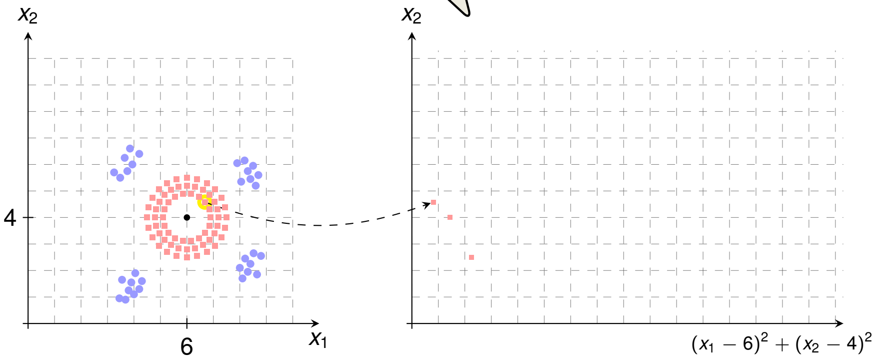
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



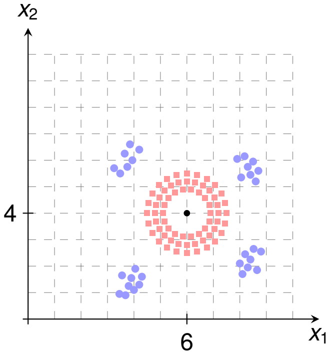
# A Glimpse at the The Kernel Method

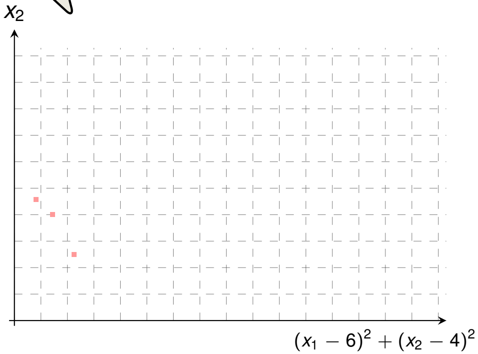
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

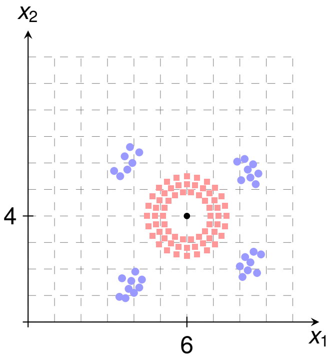
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

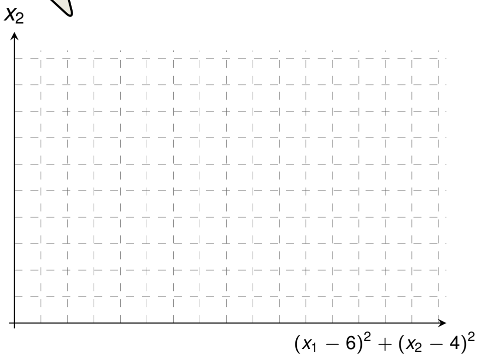


# A Glimpse at the The Kernel Method

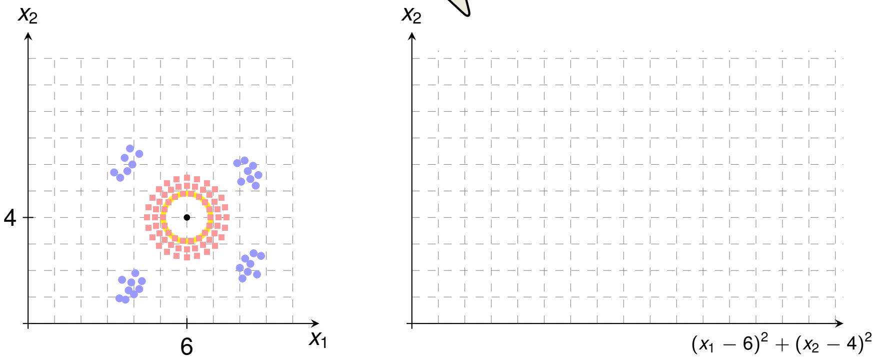
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



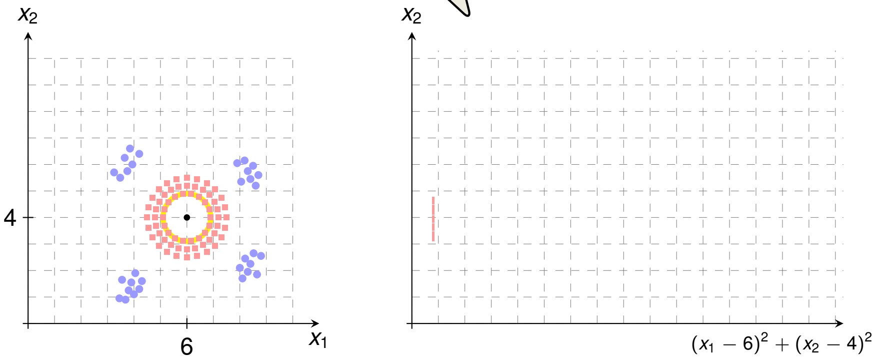
# A Glimpse at the The Kernel Method

new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



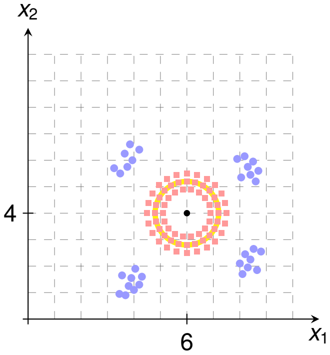
# A Glimpse at the The Kernel Method

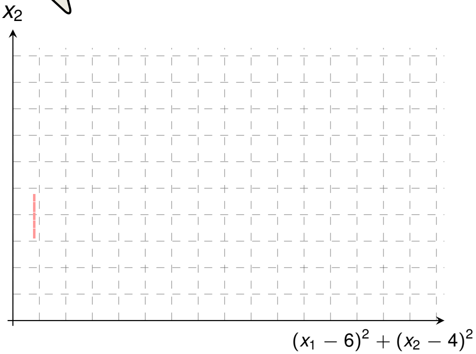
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

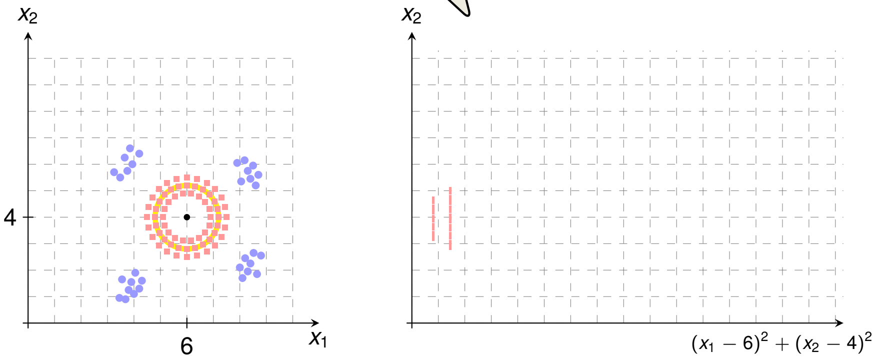
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



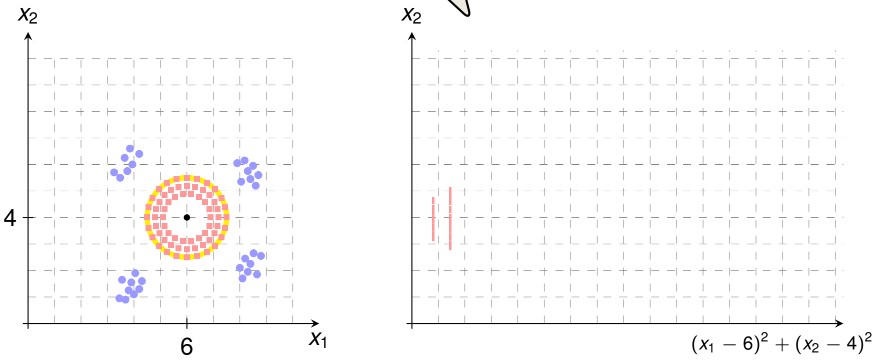
# A Glimpse at the The Kernel Method

new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



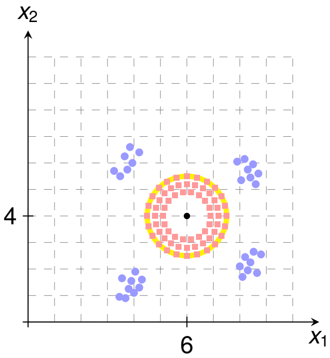
# A Glimpse at the The Kernel Method

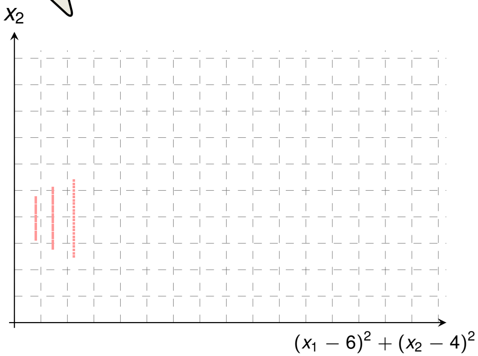
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

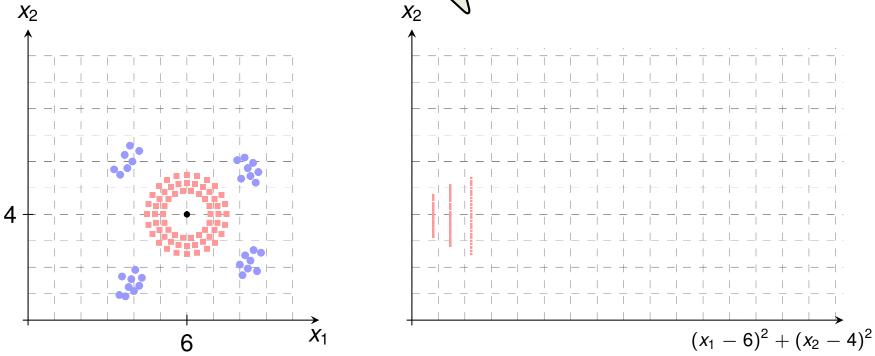
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



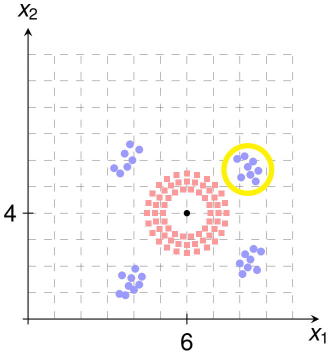
# A Glimpse at the The Kernel Method

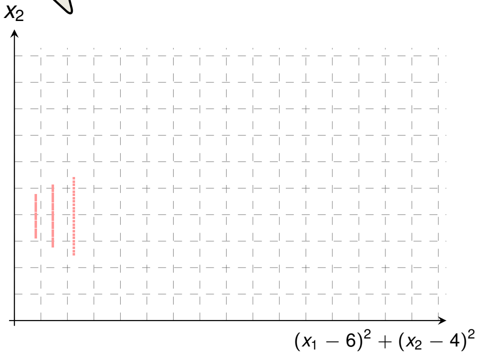
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

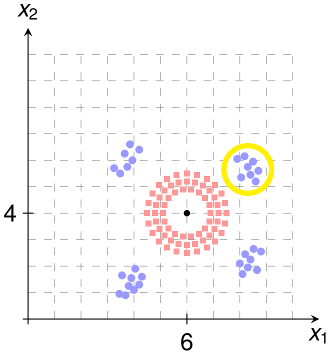
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

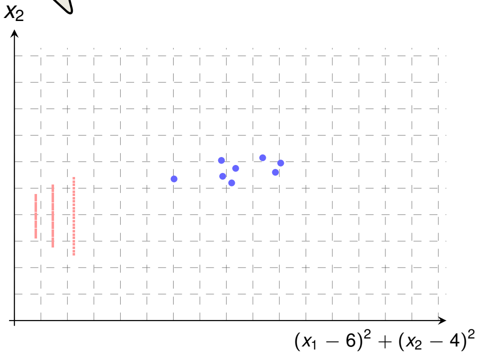


# A Glimpse at the The Kernel Method

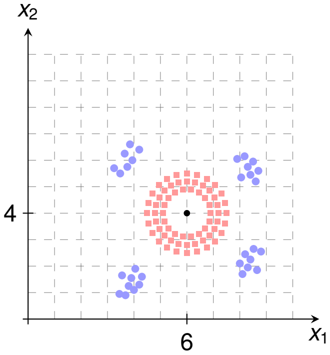
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

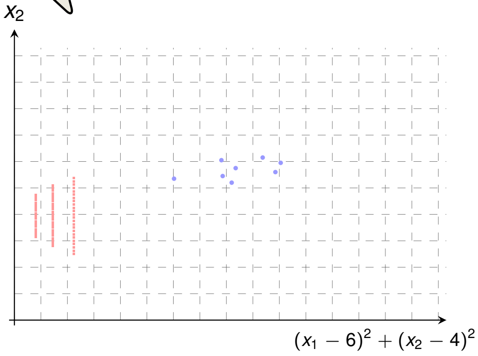


# A Glimpse at the The Kernel Method

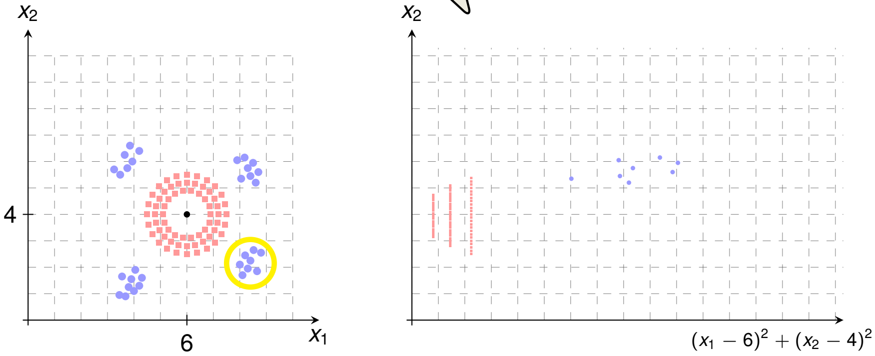
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



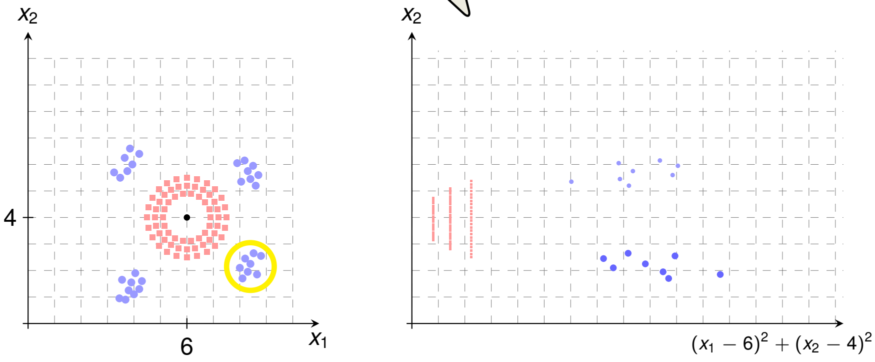
# A Glimpse at the The Kernel Method

new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



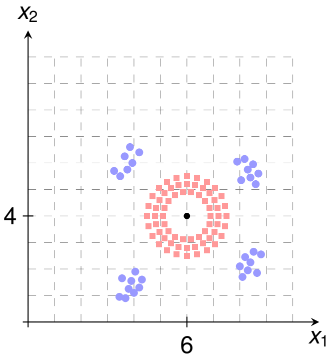
## A Glimpse at the The Kernel Method

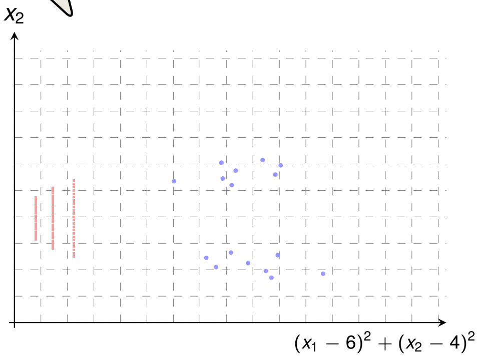
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

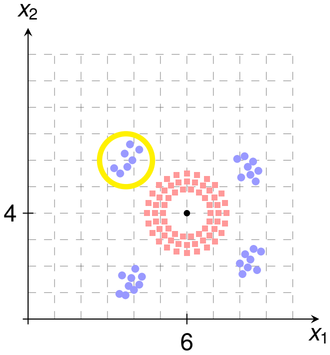
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

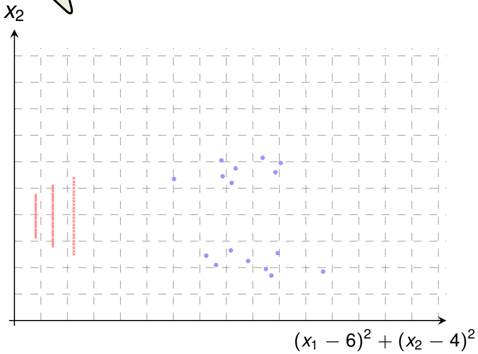


# A Glimpse at the The Kernel Method

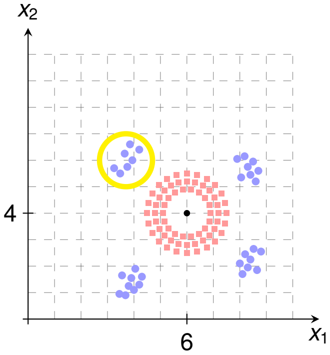
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

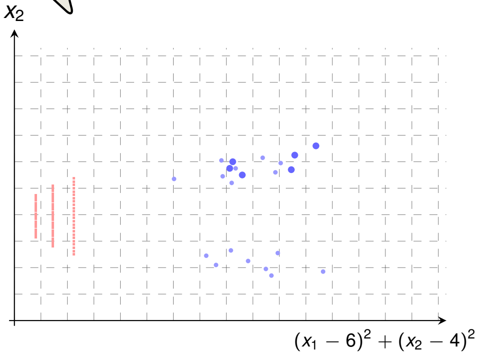


# A Glimpse at the The Kernel Method

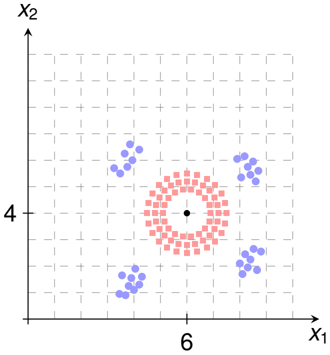
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

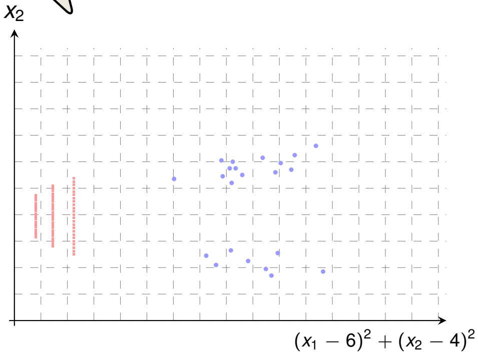


# A Glimpse at the The Kernel Method

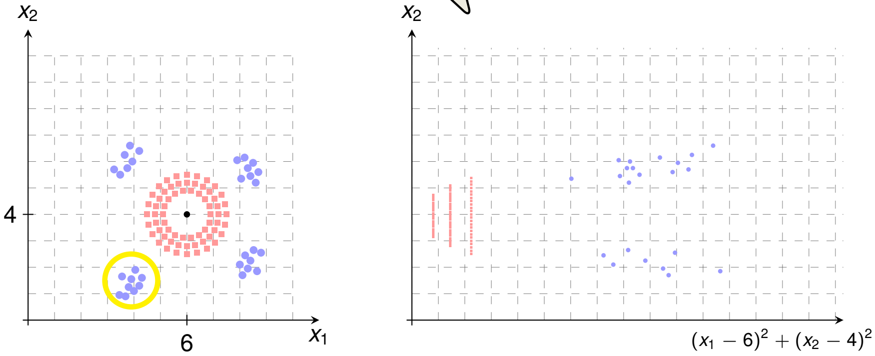
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



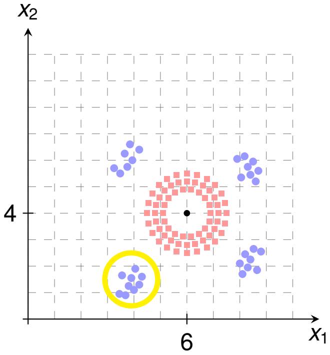
# A Glimpse at the The Kernel Method

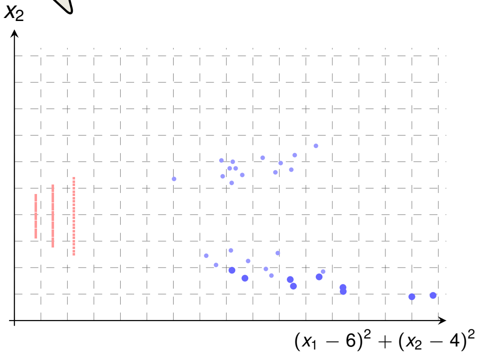
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

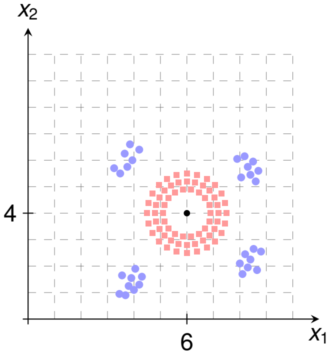
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

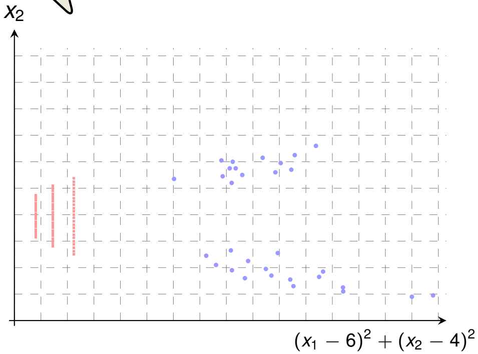


# A Glimpse at the The Kernel Method

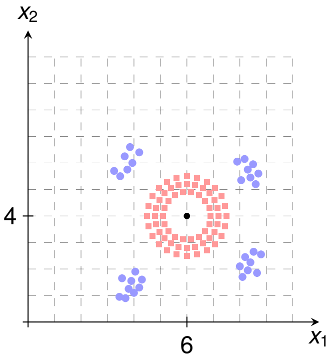
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$

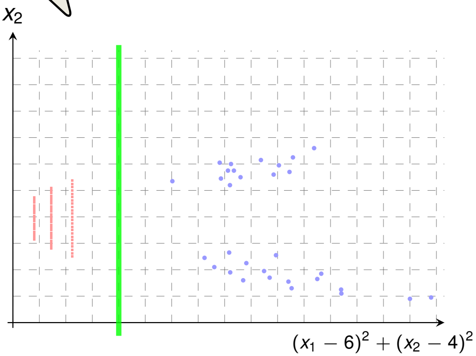


# A Glimpse at the The Kernel Method

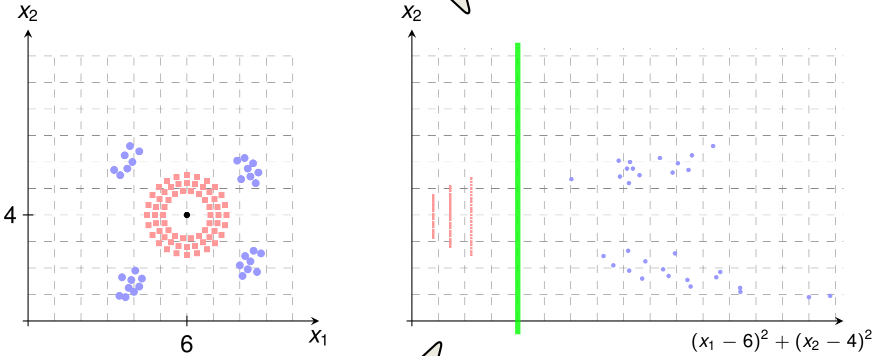
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$



# A Glimpse at the The Kernel Method

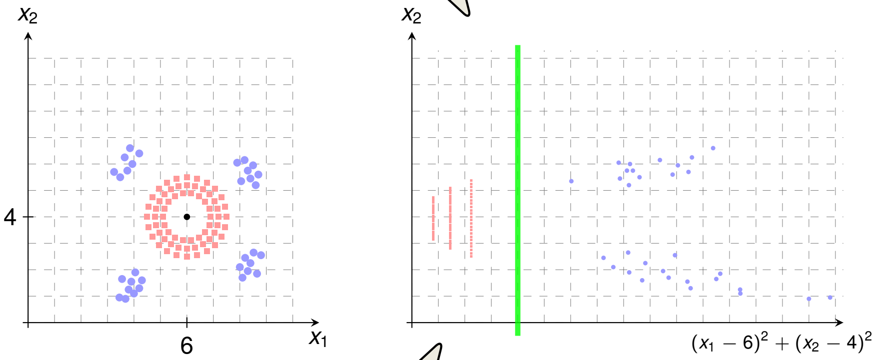
new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$




in new feature space, a linear separation is possible!

# A Glimpse at the The Kernel Method

new feature space: $(x_1, x_2) \mapsto ((x_1 - 6)^2 + (x_2 - 4)^2, x_2)$




in new feature space, a linear separation is possible!

- How do we choose a proper feature space?
- How do we compute the linear classifier efficiently?
- How do we obtain a non-linear classifier in the original space?

# Outline

Introduction

Perceptron

Conclusion, Problems and Solutions

Additional Material: Why Perceptron Works

# Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
  **for** $t = 1, 2, \ldots$
    **if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
    **else**
      **output** $\mathbf{w}^{(t)}$

# Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
  **if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
  **else**
    **output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$.

# Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
  **for** $t = 1, 2, \ldots$
    **if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
    **else**
      **output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$. Then Batch Perceptron stops after at most $(R/\gamma)^2$ iterations, and when it stops it holds that for all $i \in [m]$, $y_i \langle \mathbf{w}^{(t)}, x_i \rangle > 0$.

# Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
  **if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
  **else**
    **output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$. Then Batch Perceptron stops after at most $(R/\gamma)^2$ iterations, and when it stops it holds that for all $i \in [m]$, $y_i \langle \mathbf{w}^{(t)}, x_i \rangle > 0$.

# Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
  **if** $(\exists\, i \ \text{ s.t. } \ y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
  **else**
    **output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$. Then Batch Perceptron stops after at most $(R/\gamma)^2$ iterations, and when it stops it holds that for all $i \in [m]$, $y_i \langle \mathbf{w}^{(t)}, x_i \rangle > 0$.

- Correctness: Clear that when it terminates, all points are correctly classified ✓

# Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
**if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
**else**
**output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$. Then Batch Perceptron stops after at most $(R/\gamma)^2$ iterations, and when it stops it holds that for all $i \in [m]$, $y_i \langle \mathbf{w}^{(t)}, x_i \rangle > 0$.

- Correctness: Clear that when it terminates, all points are correctly classified ✓
- Running Time: Let $\mathbf{w}^*$ be a vector that achieves min. $B$

## Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
 **for** $t = 1, 2, \ldots$
  **if** $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
   $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
  **else**
   **output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$. Then Batch Perceptron stops after at most $(R/\gamma)^2$ iterations, and when it stops it holds that for all $i \in [m]$, $y_i \langle \mathbf{w}^{(t)}, x_i \rangle > 0$.

- Correctness: Clear that when it terminates, all points are correctly classified ✓
- Running Time: Let $\mathbf{w}^*$ be a vector that achieves min. $B$
  - Geometric Perspective: The cosine between $\mathbf{w}^*$ and $\mathbf{w}^{(t+1)}$ increases each step

## Analysis of the Perceptron Algorithm

**Batch Perceptron**

**input:** A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
**initialize:** $\mathbf{w}^{(1)} = (0, \ldots, 0)$
**for** $t = 1, 2, \ldots$
  **if** $(\exists\, i \text{ s.t. } y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ then
    $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$
  **else**
    **output** $\mathbf{w}^{(t)}$

**Analysis**

Assume $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ are separable by a hyperplane. Let $R := \max_{1 \leq i \leq m} \|\mathbf{x}_i\|$ and let $\mathbf{w}^*$ be a vector with $\|\mathbf{w}^*\| = 1$ such that $y_i \langle \mathbf{w}^*, x_i \rangle \geq \gamma$ for all $1 \leq i \leq m$. Then Batch Perceptron stops after at most $(R/\gamma)^2$ iterations, and when it stops it holds that for all $i \in [m]$, $y_i \langle \mathbf{w}^{(t)}, x_i \rangle > 0$.

- Correctness: Clear that when it terminates, all points are correctly classified ✓
- Running Time: Let $\mathbf{w}^*$ be a vector that achieves min. $B$
  - Geometric Perspective: The cosine between $\mathbf{w}^*$ and $\mathbf{w}^{(t+1)}$ increases each step
  - Since cosine is at most 1, this implies termination after $(R/\gamma)^2$ steps

# Convergence Analysis

Proof consists of 3 **Key Steps**:

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If *w* separates all points correctly, then $\alpha \cdot w$ does as well.

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle$$

.

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle$$

.

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle$$

.

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned}\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma.\end{aligned}$$

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned}\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma.\end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\|\mathbf{w}^{(t+1)}\|^2$$

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned}\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma.\end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\|\mathbf{w}^{(t+1)}\|^2 = \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2$$

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \end{aligned}$$

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \end{aligned}$$

Binomial Expansion

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned}\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma.\end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\begin{aligned}\|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R.\end{aligned}$$

Binomial Expansion

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned}\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma.\end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

Binomial Expansion

$$\begin{aligned}\|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R.\end{aligned}$$

4. **Combining**:

$$\|\mathbf{w}^{(t+1)}\| \geq$$

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R. \end{aligned}$$

Binomial Expansion

4. **Combining**:

$$\sqrt{t} \cdot R \geq \|\mathbf{w}^{(t+1)}\| \geq$$

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

Binomial Expansion

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R. \end{aligned}$$

4. **Combining**:

$$\sqrt{t} \cdot R \geq \|\mathbf{w}^{(t+1)}\| \geq \langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle$$

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

Binomial Expansion

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R. \end{aligned}$$

4. **Combining**:

$$\sqrt{t} \cdot R \geq \|\mathbf{w}^{(t+1)}\| \geq \langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t \cdot \gamma$$

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R. \end{aligned}$$

Binomial Expansion

4. **Combining**:

Since $\|\mathbf{w}^*\| = 1$

$$\sqrt{t} \cdot R \geq \|\mathbf{w}^{(t+1)}\| \geq \langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t \cdot \gamma$$

## Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:
   $$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$
3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:
   $$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R. \end{aligned}$$

   Binomial Expansion
4. **Combining**:

   Since $\|\mathbf{w}^*\| = 1$

   $$\sqrt{t} \cdot R \geq \|\mathbf{w}^{(t+1)}\| \geq \langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t \cdot \gamma \qquad \square$$

# Convergence Analysis

Proof consists of 3 **Key Steps**:

1. **Step 1**: We can always find a $\mathbf{w}^*$ with $\|\mathbf{w}^*\| = 1$ and $\langle \mathbf{w}^*, \mathbf{x} \rangle \geq \gamma > 0$.
   - Proof: If $w$ separates all points correctly, then $\alpha \cdot w$ does as well.
2. **Step 2**: The inner product increases: $\langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle \geq t \cdot \gamma$
   - Proof:

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t+1)} \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i \rangle = \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + y_i \cdot \langle \mathbf{w}^*, \mathbf{x}_i \rangle \\ \geq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle + \gamma. \end{aligned}$$

3. **Step 3**: The norm is at most $\|\mathbf{w}^{(t+1)}\|^2 \leq t \cdot R^2$:
   - Proof:

Binomial Expansion

$$\begin{aligned} \|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i \cdot \mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle + \|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R. \end{aligned}$$

4. **Combining**:

Since $\|\mathbf{w}^*\| = 1$

$$\sqrt{t} \cdot R \geq \|\mathbf{w}^{(t+1)}\| \geq \langle \mathbf{w}^{(t+1)}, \mathbf{w}^* \rangle \geq t \cdot \gamma \qquad \square$$

Runtime bound independent of number of points and dimensionality!