Perceptron

Agenda

- Review of models
- Parts of a perceptron
- Making decisions with a learned model
- Learning a perceptron representation
- Batch and Online learning
- Sparse and Dense reprsentations

Learning from examples

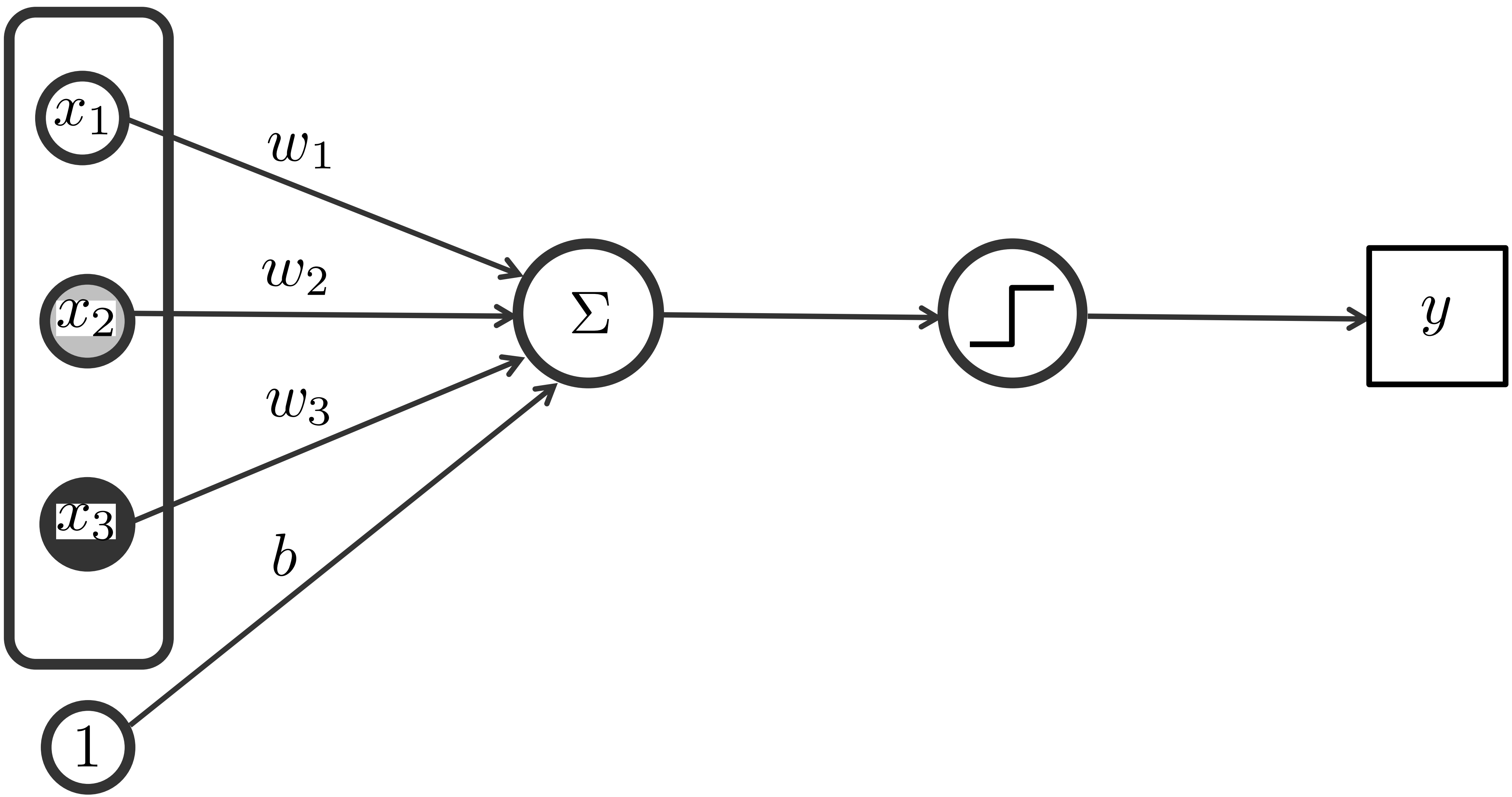
- Decision Trees
  - learn nested **if-then-else** rules
- kNN
  - memorize examples
- Linear classifiers
  - find simple boundaries in space

Frank Rosenblatt 1928 – 1971. Psychologist, inventor of the perceptron algorithm.

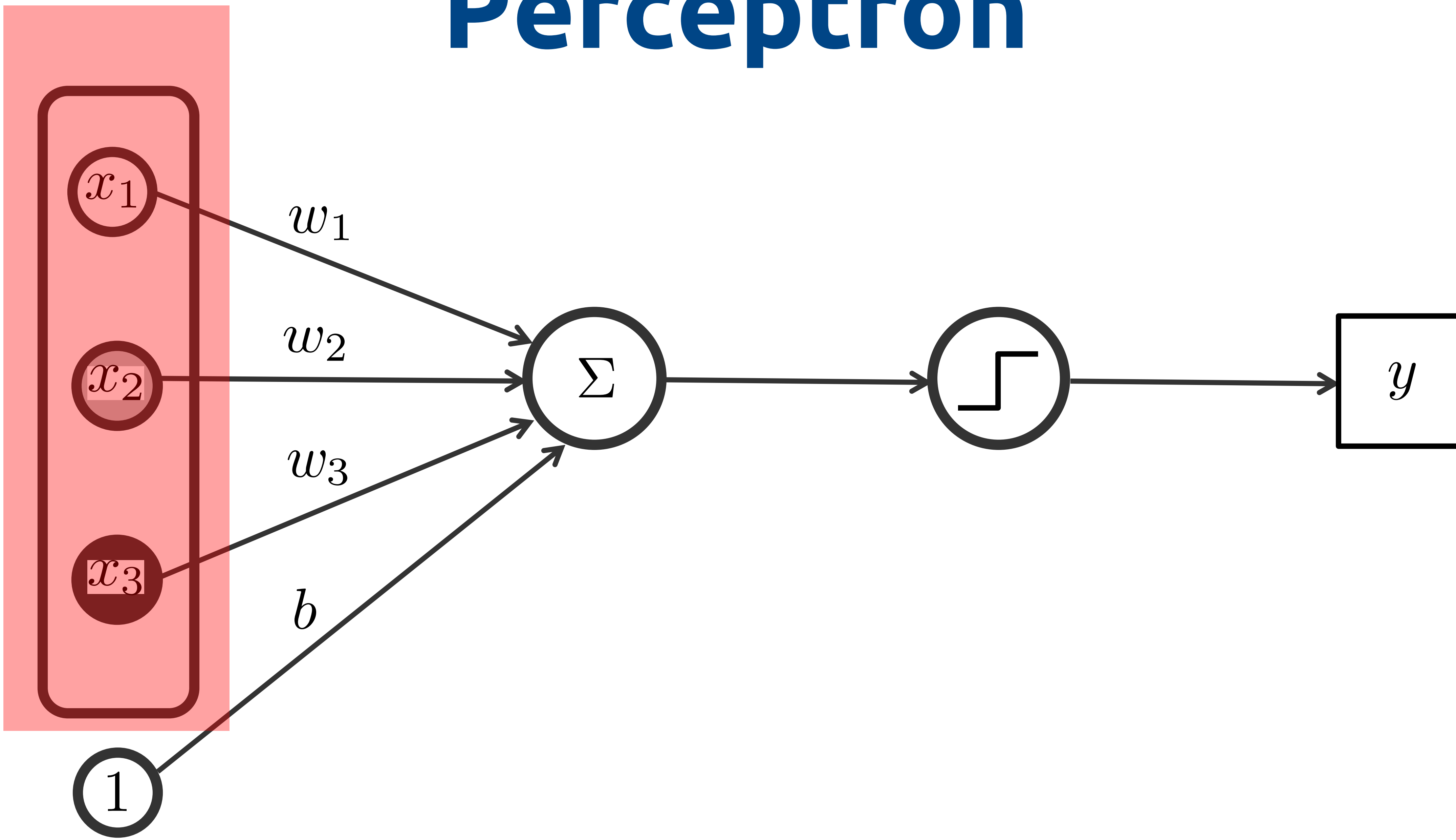
Agenda

- Review of models
- Parts of a perceptron
- Making decisions with a learned model
- Learning a perceptron representation
- Batch and Online learning
- Sparse and Dense representations

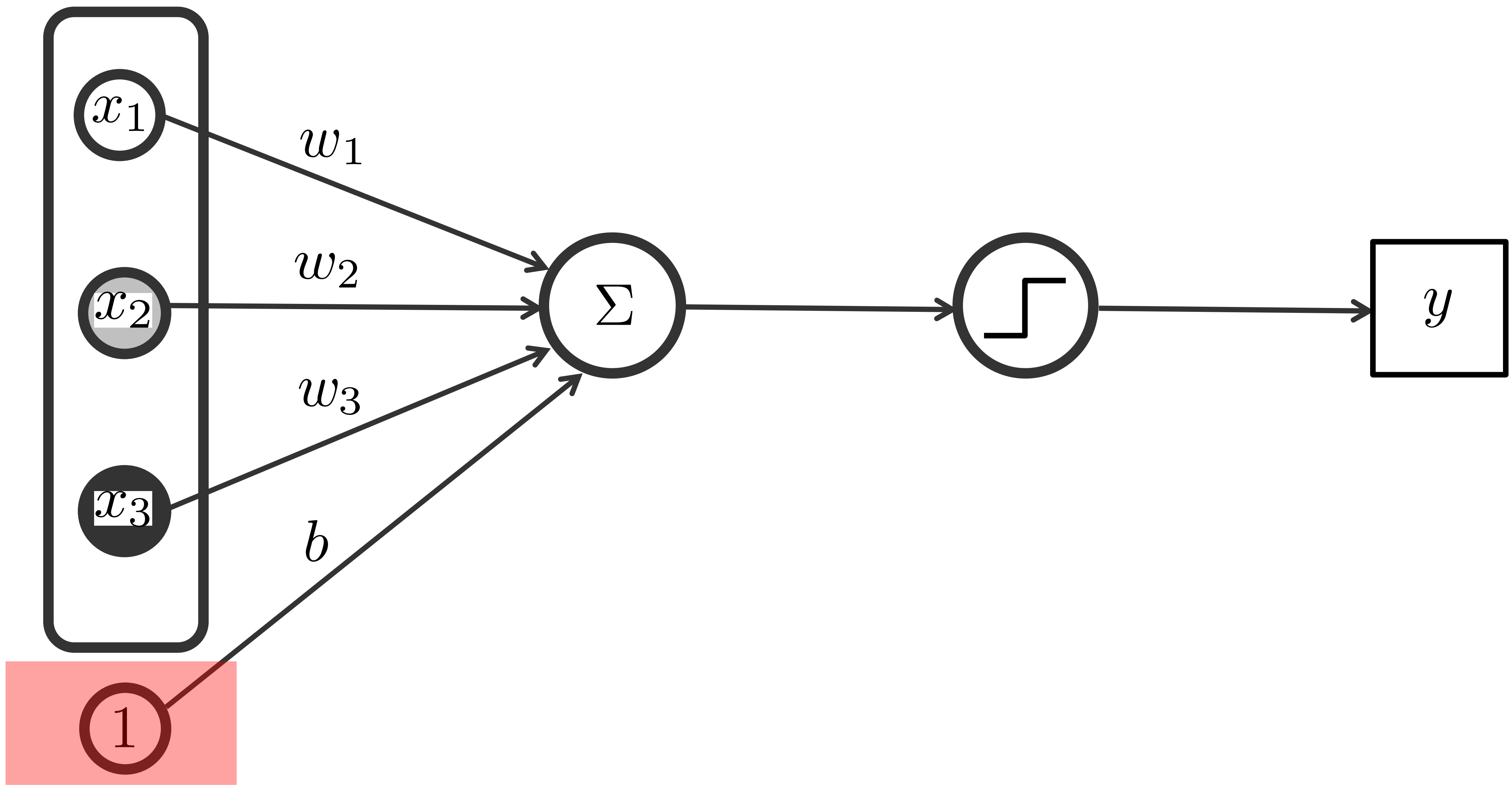
Perceptron



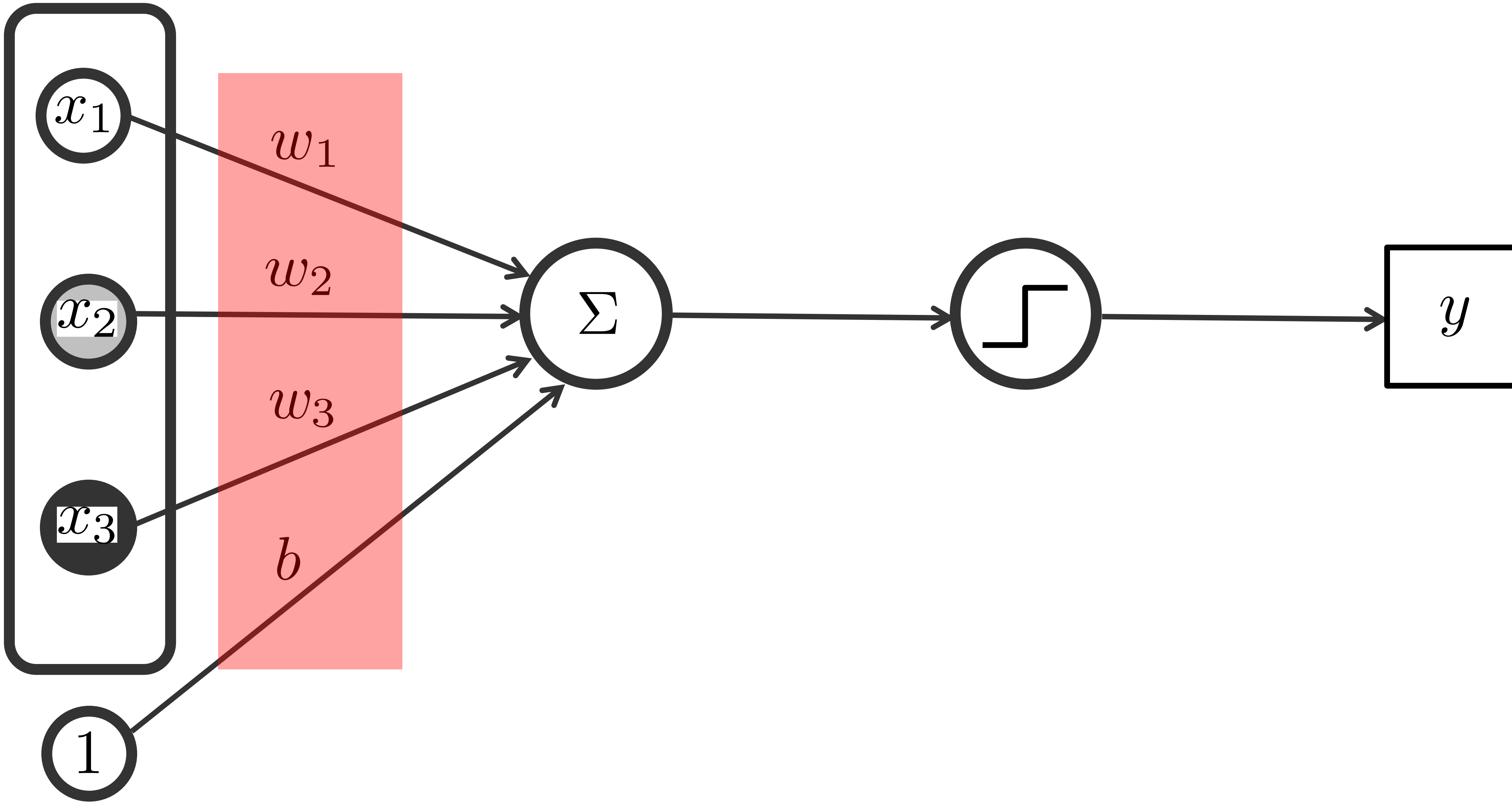
Perceptron



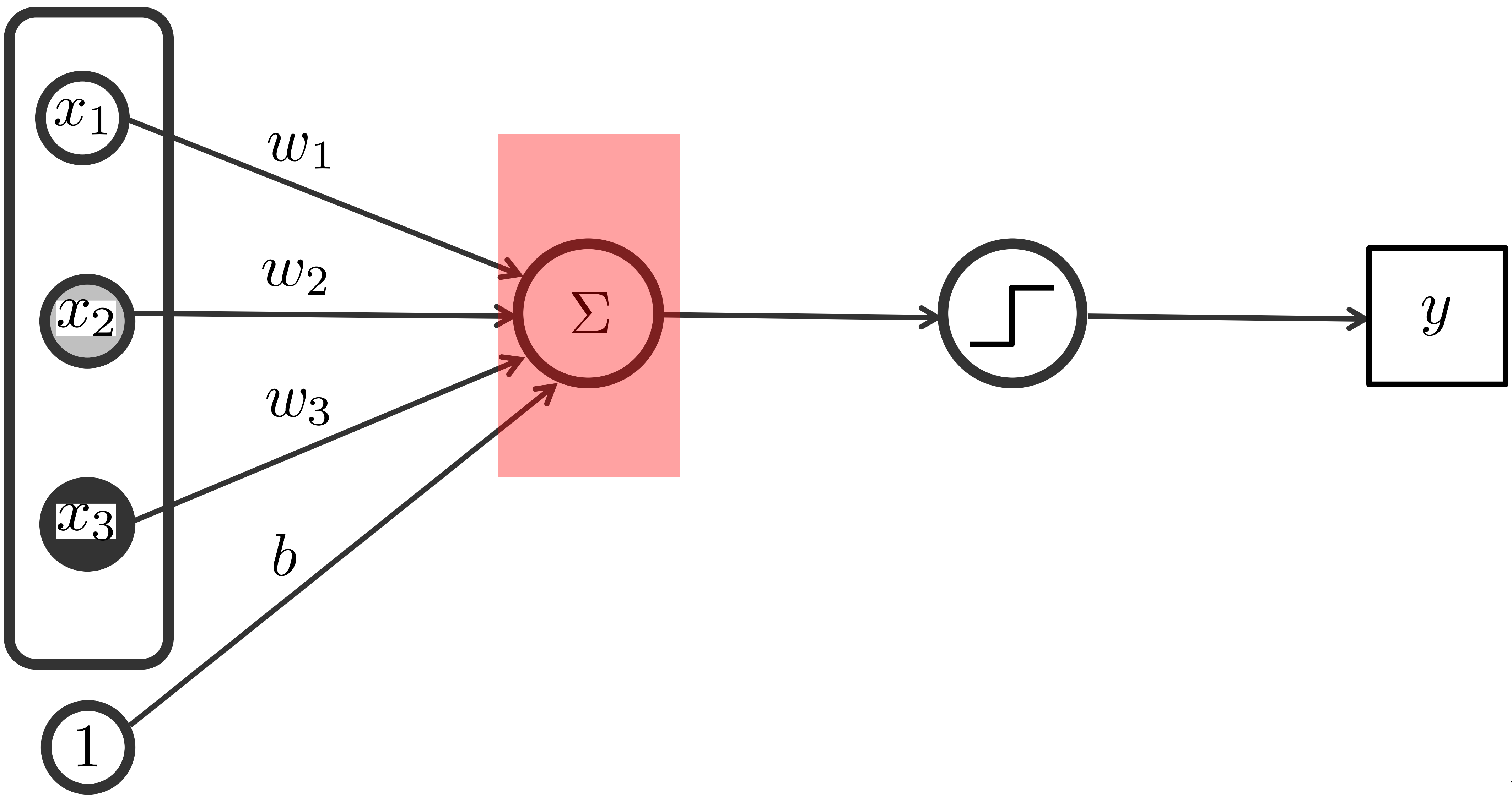
Perceptron



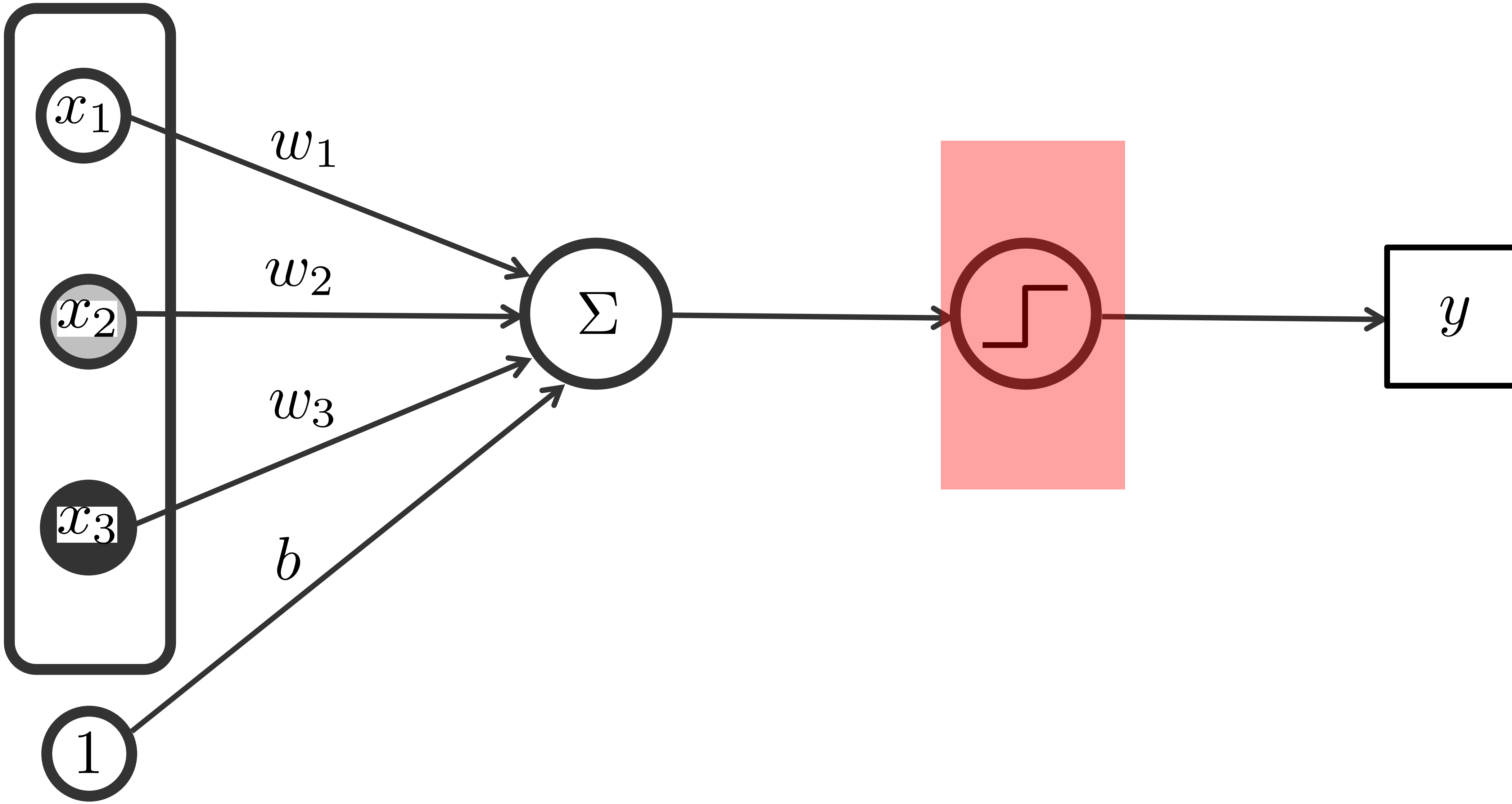
Perceptron



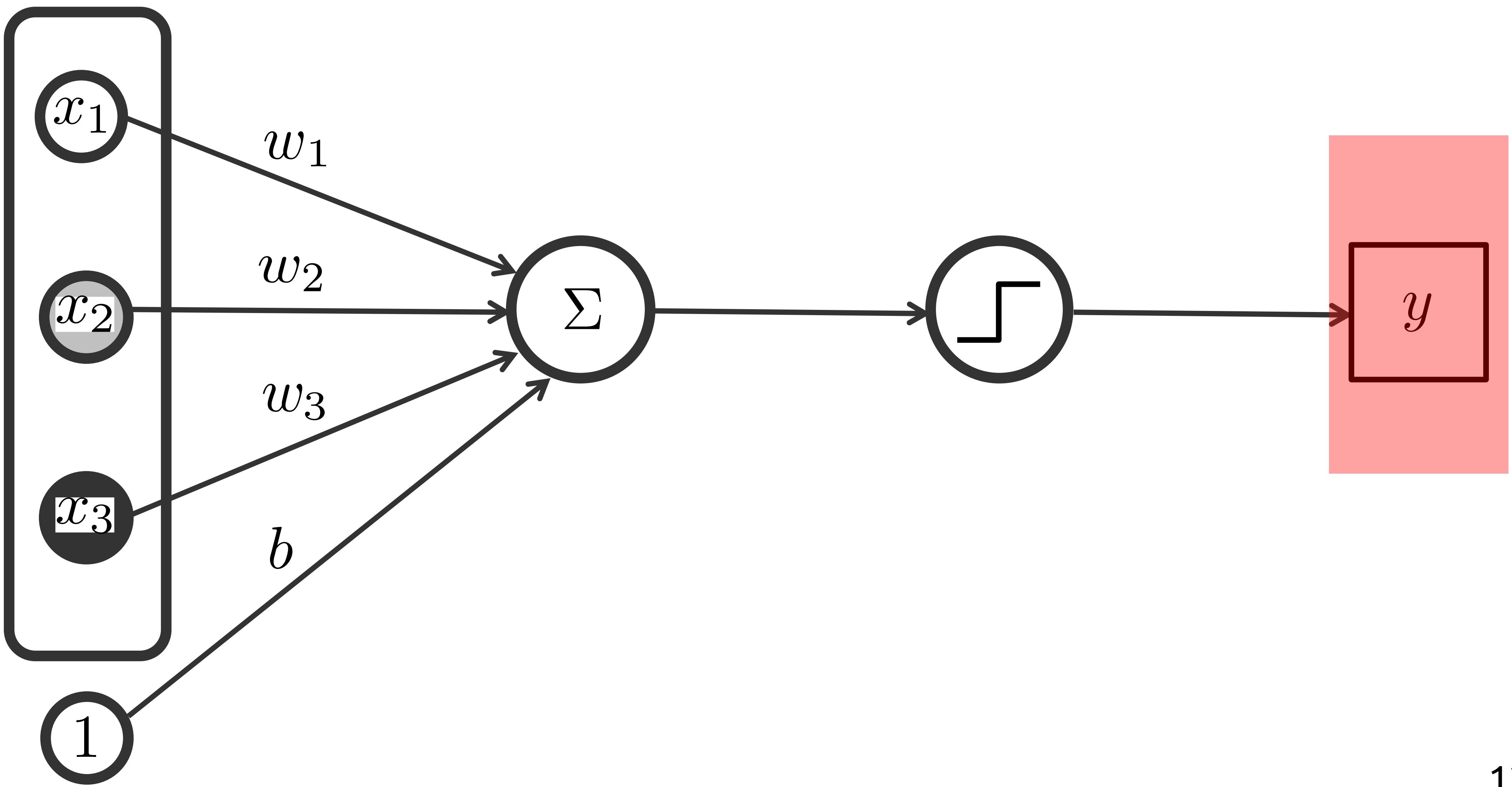
Perceptron



Perceptron



Perceptron



Agenda

- Review of models
- Parts of a perceptron
- Making decisions with a learned model
- Learning a perceptron representation
- Batch and Online learning
- Sparse and Dense representations

Perceptron classification rule

- Perceptron uses a simple rule to classify objects
- It computes the weighted sum of the input features (plus **bias**)
- If this sum is greater than or equal to **0**, it outputs positive class **+1**
- Otherwise it outputs negative class **-1**

Discriminant function

$$f(\mathbf{x}) = \left(\sum_{i=1}^{N} w_i x_i \right) + b$$

$$y = \begin{cases} +1 & \text{if } f(\mathbf{x}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Example: movie reviews

	#good	#dark	#mediocre	#the
$\mathbf{x}^1$ = (	2,	0,	0,	5)
$\mathbf{w}$ = (	2.5,	0.5,	-4.0,	0.0)

b = 0.5

score f($\mathbf{x}^1$) =

Role of bias

- When $\mathbf{w} \cdot \mathbf{x} \approx 0,$ bias decides which class to predict
- Makes the default decision
- Biases the classifier towards positive or negative class

Geometric interpretation in 2D

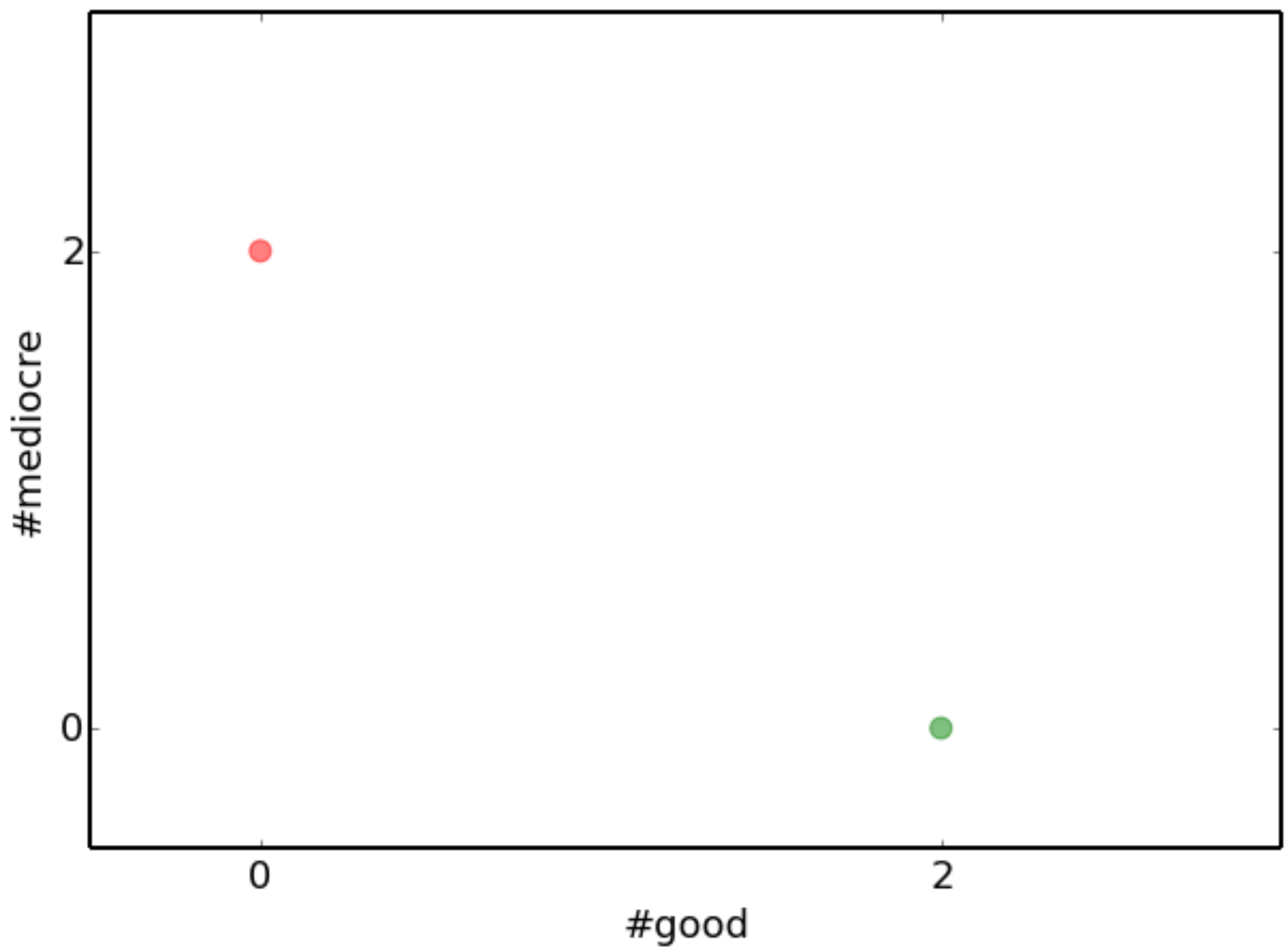
#good #mediocre

$\mathbf{x}^1 = (\quad 2, \quad 0 \quad)$

$\mathbf{x}^2 = (\quad 0, \quad 2 \quad)$

$\mathbf{w} = (\quad 2.5, \; -4.0 \quad)$

$b = 0.5$


#mediocre
2
0
0
2
#good

Decision boundary

$$w_1 x_1 + w_2 x_2 + b = 0$$

Solve for x_2

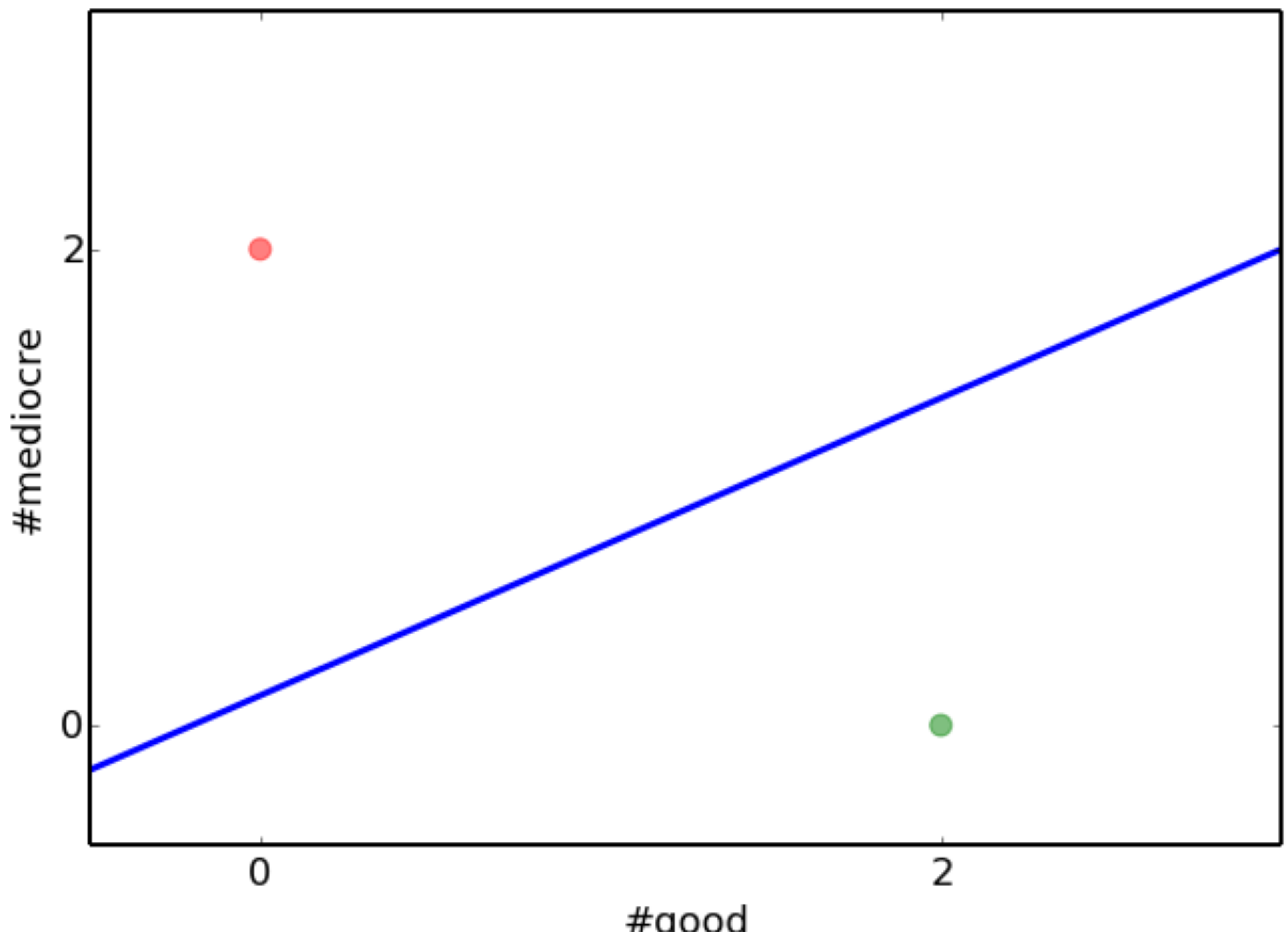
$$x_2 = \underbrace{-\frac{w_1}{w_2}}_{\text{slope}} x_1 - \underbrace{\frac{b}{w_2}}_{\text{intercept}}$$

Aside: Multiple dimensions

- Examples with only 2 dimensions are not very realistic
    - What about domains with more than two features?
- 2 dimensions are “split” by a 1 dimensional shape: a line
- 3 dimensional space is “split” by a 2 dimensional shape: a plane
- N dimensional space is “split” by an (N-1) dimensional shape: a hyperplane

Agenda

- Review of models
- Parts of a perceptron
- Making decisions with a learned model
- Learning a perceptron representation
- Batch and Online learning
- Sparse and Dense representations


2
0
#mediocre
0
2
#good

How can we find good (w,b) ?

- Go through examples one by one
- Try classifying current example with current (**w**,b)
- If correct, keep going
- If not correct, adjust (**w**,b)

- New item: ($\mathbf{x}$, +1)
- With current ($\mathbf{w}$, b), the score $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ is less than 0

How to adjust (w,b)?

- New item: ($\mathbf{x}$, +1)
- With current ($\mathbf{w}$, b), the score $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ is less than 0

How to adjust (w,b)?

- New item: ($\mathbf{x}$, +1)
- With current ($\mathbf{w}$, b), the score $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ is less than 0
- How do we change b to make it higher?
- How do we change **w** to make it higher?

A Concrete Example

$\mathbf{x}^1 = (\quad 2, \quad 0, \quad 0, \quad 5 \quad)$

$\mathbf{w} = (-0.5, \quad 1.0, \quad -2.0, \quad 0.0 \quad)$

$\mathbf{b} = 0$

$f(\mathbf{x}^1) = \mathbf{w} \cdot \mathbf{x}^1 + b = -1.0$

Change b to increase $f(\mathbf{x}^1)$

Change $\mathbf{w}$ to increase $f(\mathbf{x}^1)$

Vector addition and subtraction

$$\mathbf{c} = \mathbf{a} + \mathbf{b}$$

$$\text{for all } i,\ c_i = a_i + b_i$$

$\mathbf{x}^1$	= (	2,	0,	0,	5)
$\mathbf{w}$	= (	-0.5,	1.0,	-2.0,	0.0)
$\mathbf{w}+\mathbf{x}^1$	= (	1.5,	1.0,	-2.0,	5.0)

Reminder: Discriminant function

$$f(\mathbf{x}) = \left(\sum_{i=1}^{N} w_i x_i \right) + b$$

$$y = \begin{cases} +1 & \text{if } f(\mathbf{x}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Update rule: example (x,y), model (w,b)

$$1{:}\quad y_{\text{pred}} = \text{predict}((\mathbf{w}, b), \mathbf{x})$$

Update rule: example (x,y), model (w,b)

$y_{\text{pred}} = \text{predict}((\mathbf{w}, b), \mathbf{x})$

if $y = +1$ and $y_{\text{pred}} = -1$ **then**

  $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}$

  $b \leftarrow b + 1$

else if $y = -1$ and $y_{\text{pred}} = +1$ **then**

  $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{x}$

  $b \leftarrow b - 1$

Aside: Dot product notation

$$\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{N} w_i x_i$$

$$\boxed{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b}$$

Update rule: example (x,y), model (w,b)

$y_{\text{pred}} = \text{predict}((\mathbf{w}, b), \mathbf{x})$
if $y = +1$ and $y_{\text{pred}} = -1$ **then**
    $\mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}$
    $b \leftarrow b + 1$
else if $y = -1$ and $y_{\text{pred}} = +1$ **then**
    $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{x}$
    $b \leftarrow b - 1$

Proof Step 3 increases predicted score

New weight vector

$$(\mathbf{w} + \mathbf{x}) \cdot \mathbf{x} = \mathbf{w} \cdot \mathbf{x} + \mathbf{x} \cdot \mathbf{x}$$

$$\text{Since } \mathbf{x} \cdot \mathbf{x} \geq 0$$

$$\text{Thus } \mathbf{w} \cdot \mathbf{x} + \mathbf{x} \cdot \mathbf{x} \geq \mathbf{w} \cdot \mathbf{x}$$

One learning iteration over *N* examples

$\mathbf{w} \leftarrow \mathbf{0}$

$b \leftarrow 0$

for $n = 1..N$ **do**

$\quad y^n_{\text{pred}} = \text{predict}((\mathbf{w}, b), \mathbf{x}^n)$

$\quad$ **if** $y^n = +1$ and $y^n_{\text{pred}} = -1$ **then**

$\quad\quad \mathbf{w} \leftarrow \mathbf{w} + \mathbf{x}^n$

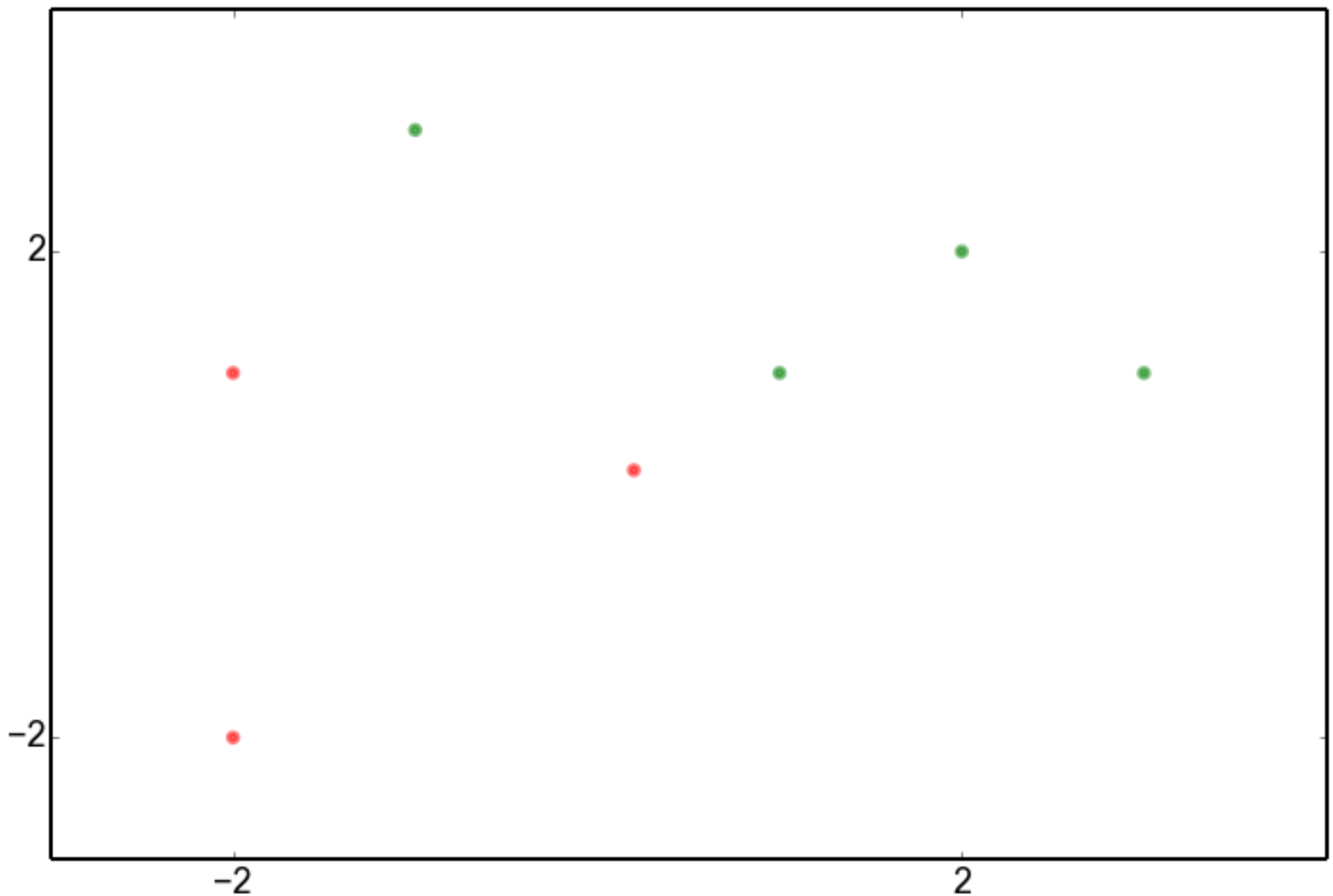
$\quad\quad b \leftarrow b + 1$

$\quad$ **else if** $y^n = -1$ and $y^n_{\text{pred}} = +1$ **then**

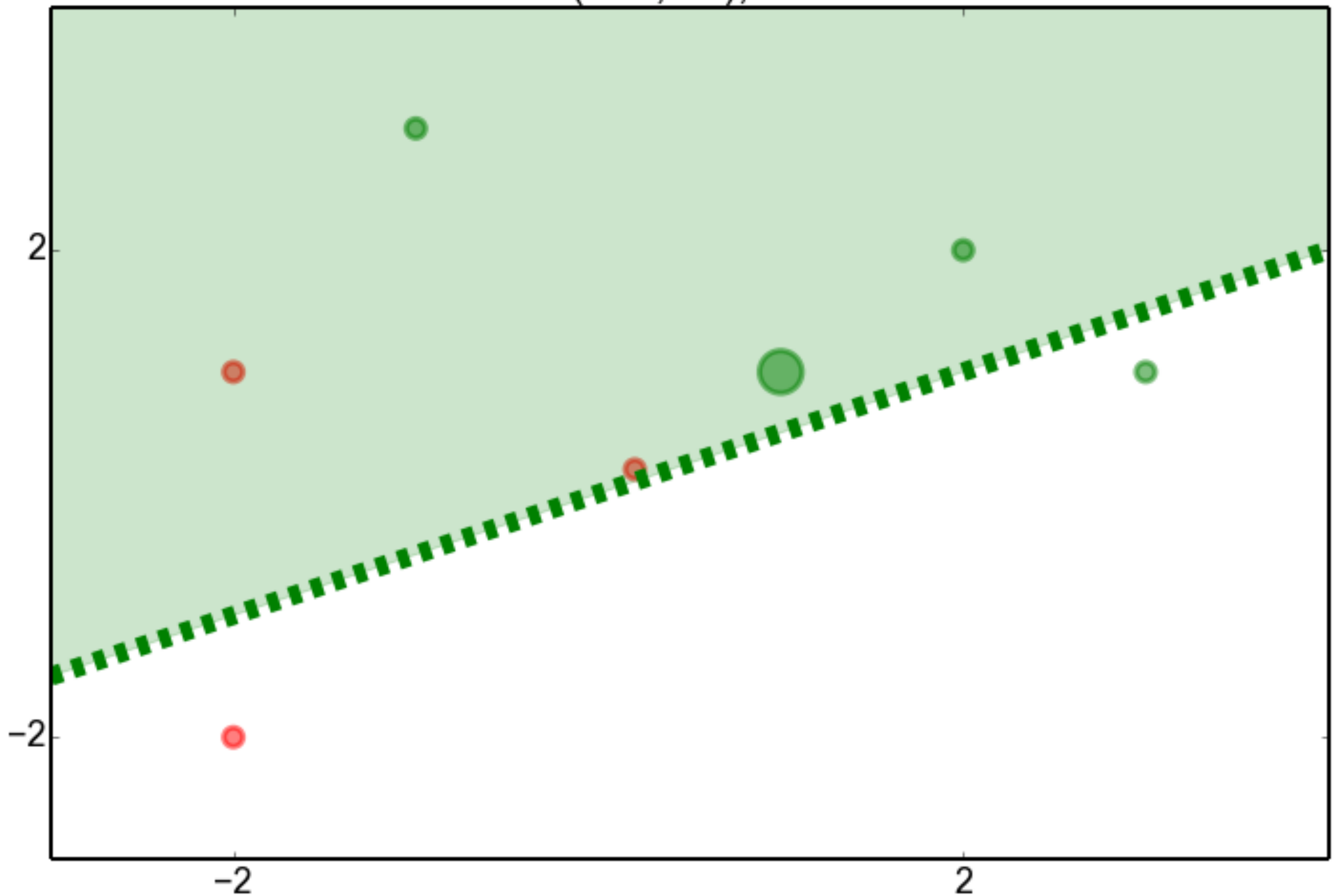
$\quad\quad \mathbf{w} \leftarrow \mathbf{w} - \mathbf{x}^n$

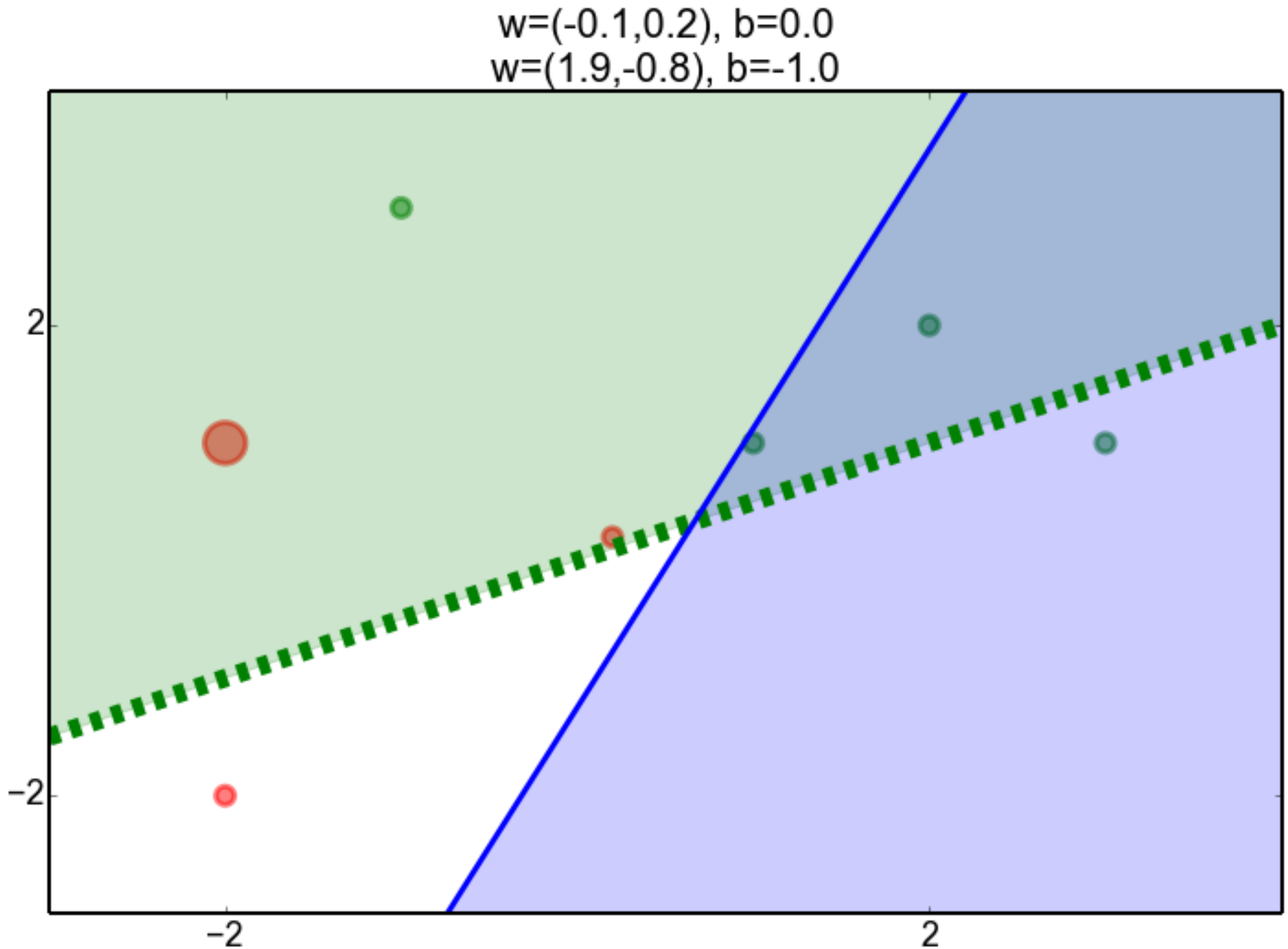
$\quad\quad b \leftarrow b - 1$

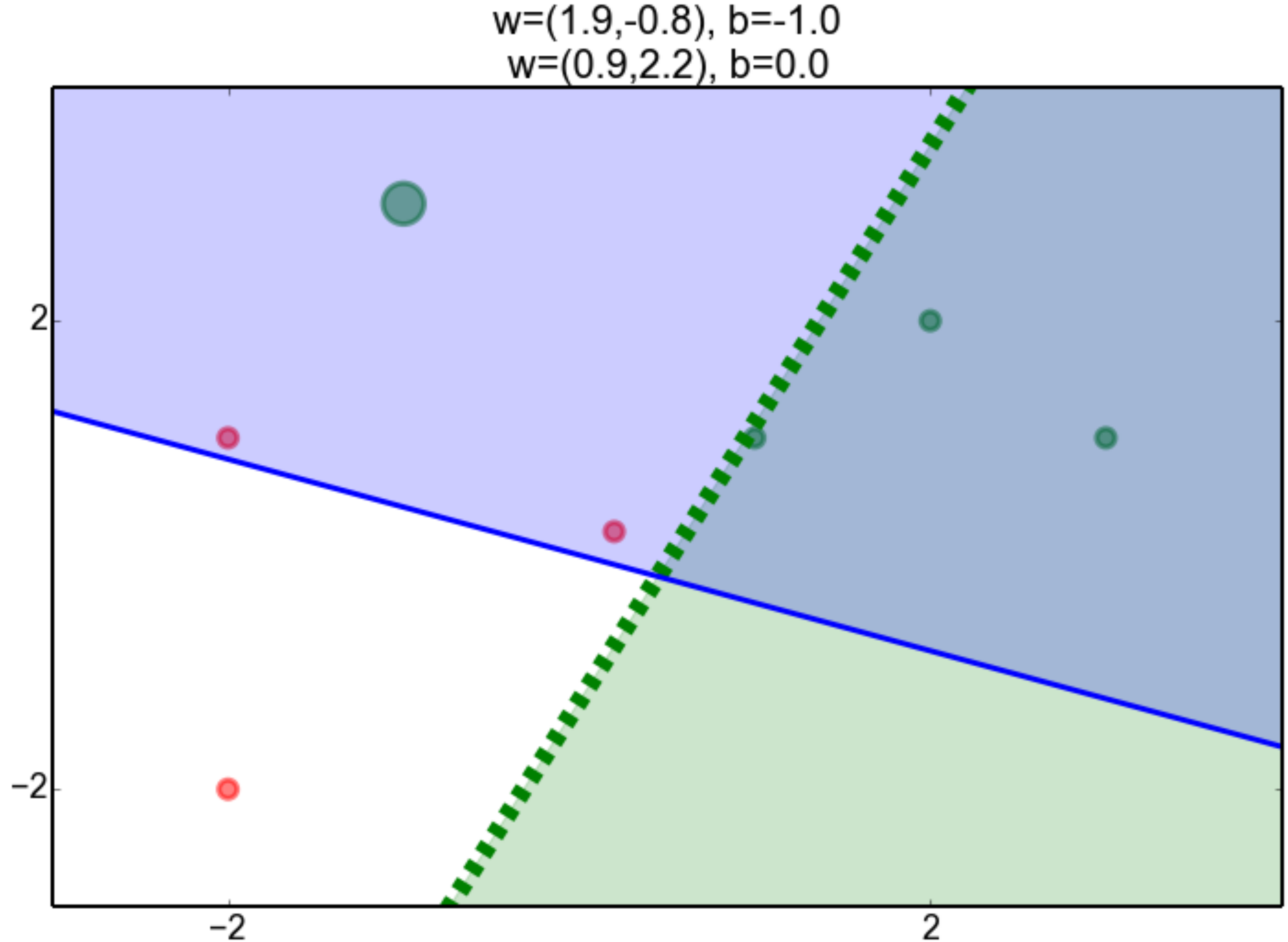
Example


2
−2
−2
2

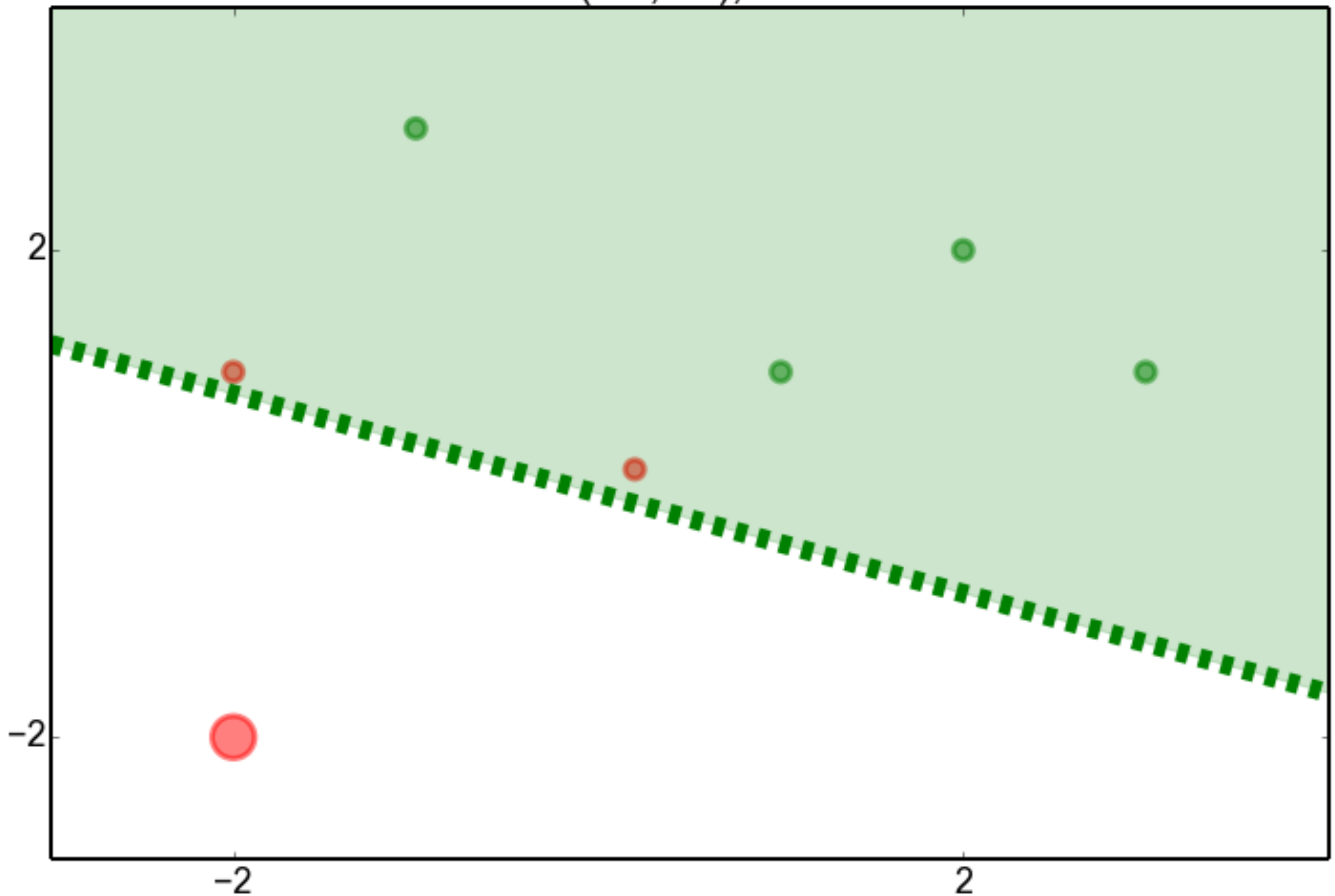
w=(-0.1,0.2), b=0.0
w=(-0.1,0.2), b=0.0




w=(-0.1,0.2), b=0.0
w=(1.9,-0.8), b=-1.0
2
−2
−2
2

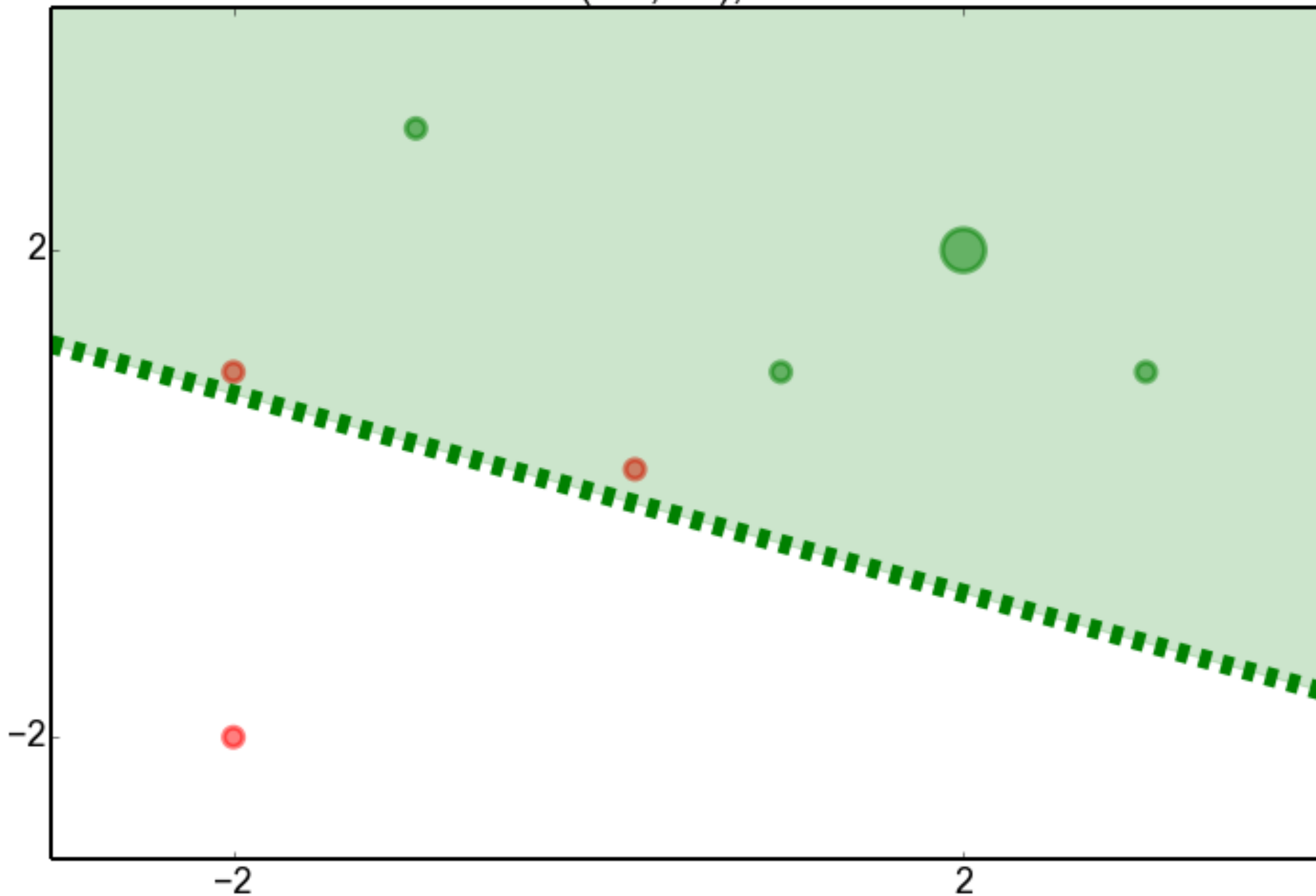

w=(1.9,-0.8), b=-1.0
w=(0.9,2.2), b=0.0
2
−2
−2
2

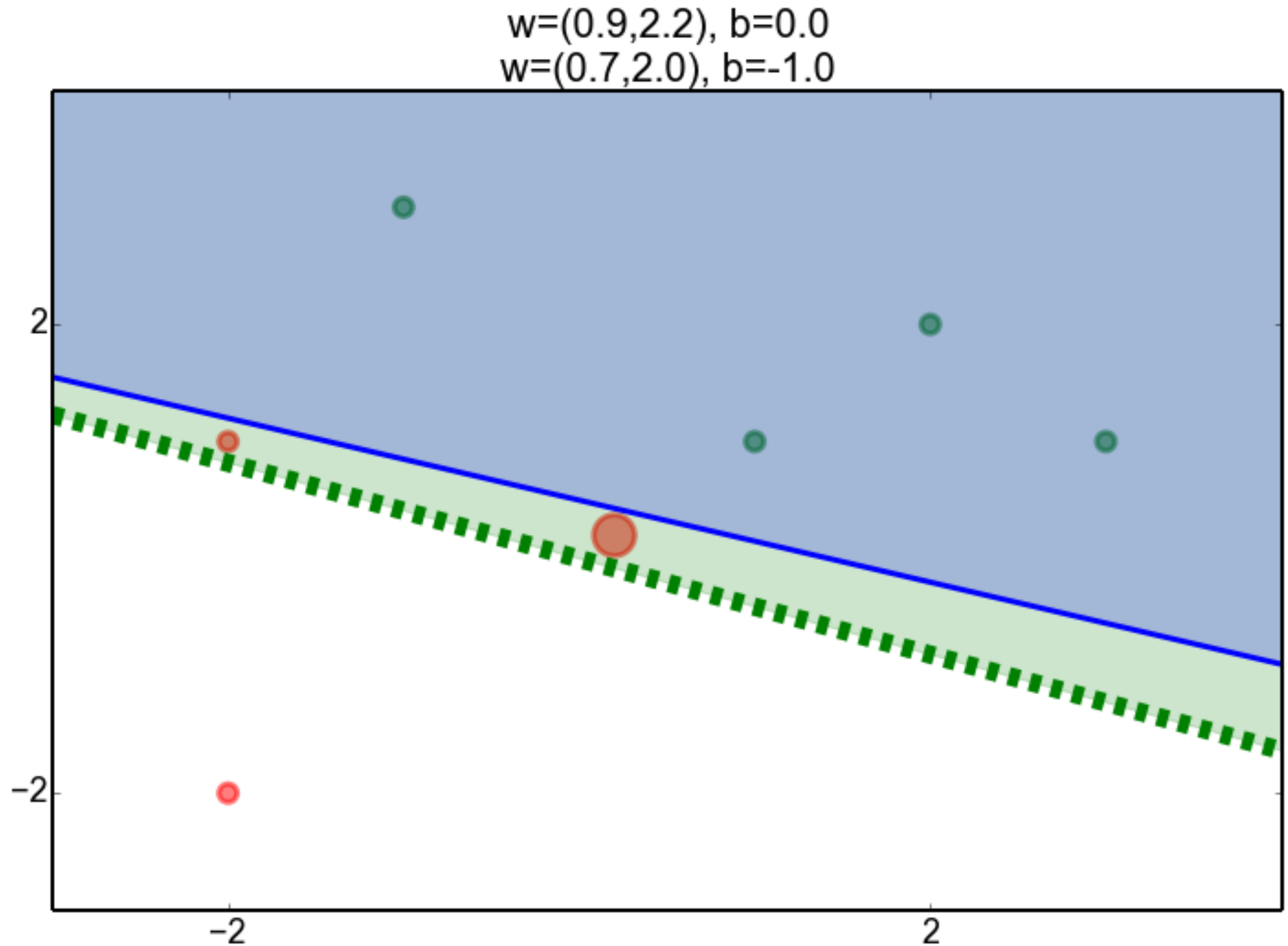
w=(0.9,2.2), b=0.0
w=(0.9,2.2), b=0.0



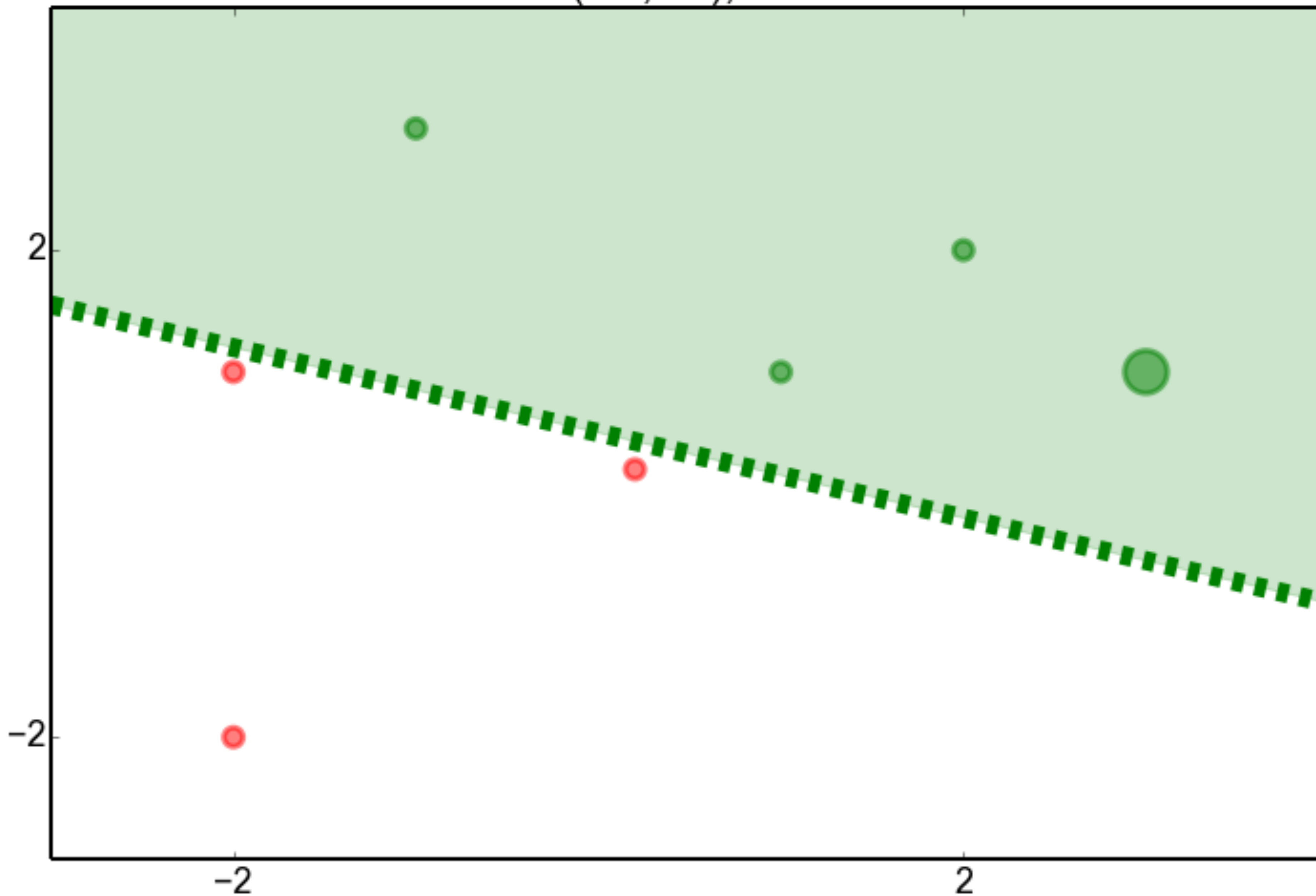
w=(0.9,2.2), b=0.0

w=(0.9,2.2), b=0.0




w=(0.9,2.2), b=0.0
w=(0.7,2.0), b=-1.0
2
−2
−2
2

w=(0.7,2.0), b=-1.0
w=(0.7,2.0), b=-1.0



Termination

It can be proven that if there is a linear boundary separating **+1** from **-1**, the perceptron algorithm will find it.

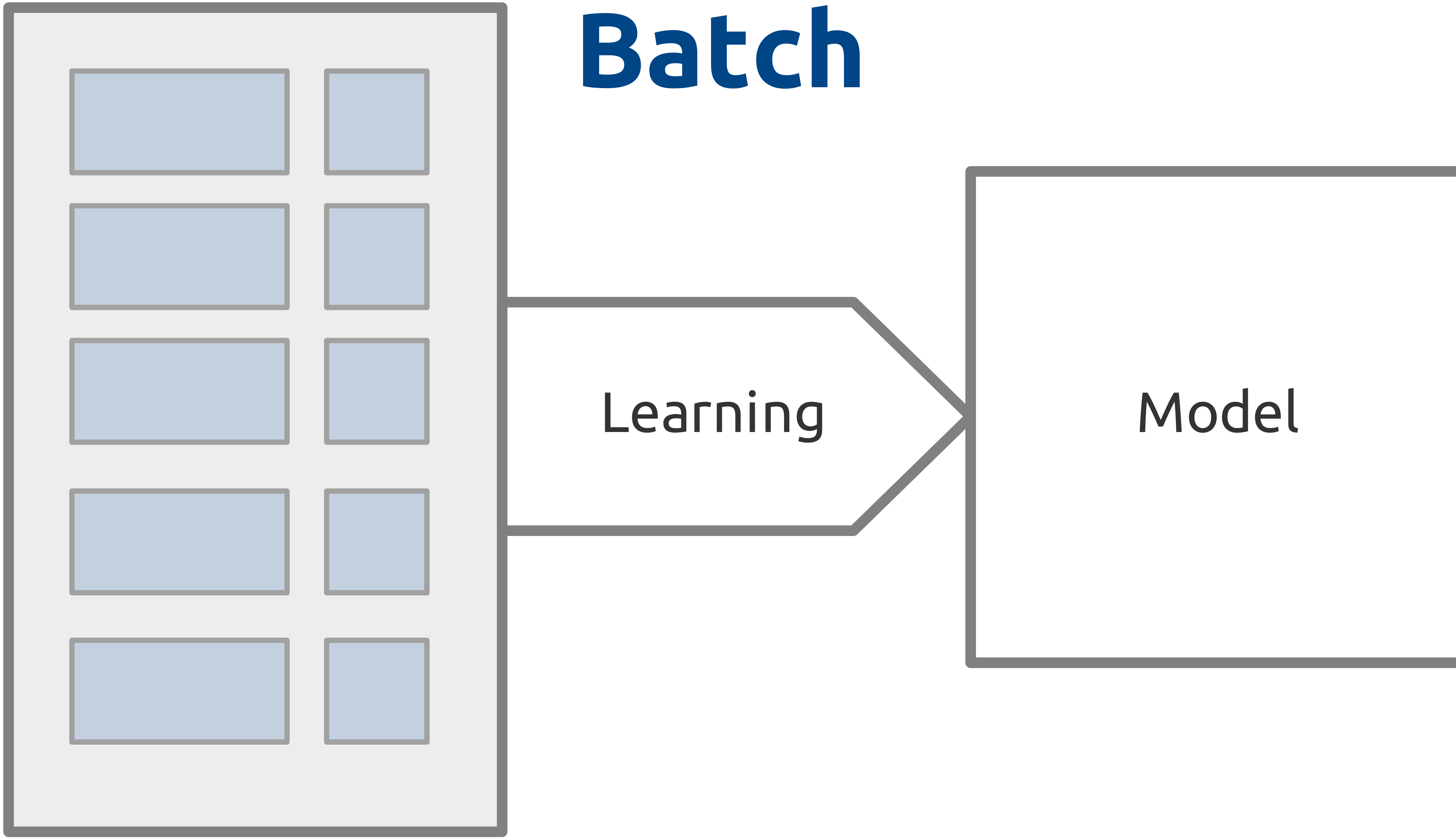
Agenda

- Review of models
- Parts of a perceptron
- Making decisions with a learned model
- Learning a perceptron representation
- Batch and Online learning
- Sparse and Dense representations

Streaming data

- Data which doesn't stop coming
  - Recordings from sensors
  - Posts to social media
  - News articles

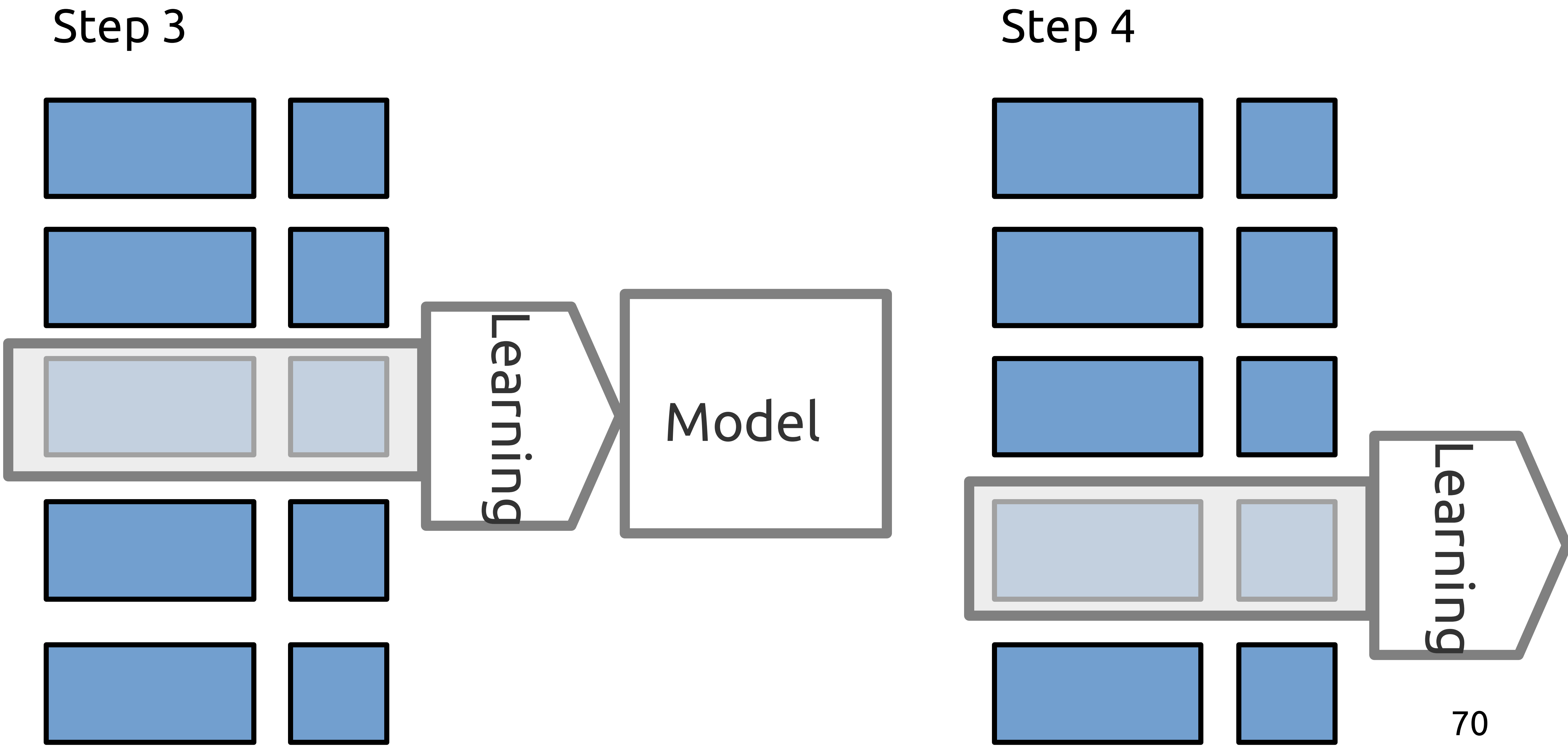
Batch



Online learning

- Perceptron looks at one example at a time
- Online learners are good for streams of data
  - Social media posts
  - Photo uploads
  - User queries on a search engine

Online



Online vs Batch

- **Batch** algorithm has to remember whole dataset
- **Online** algorithm only remembers current example
- Perceptron can imitate batch learning by iterating over data several times

Evaluation in pure online learning

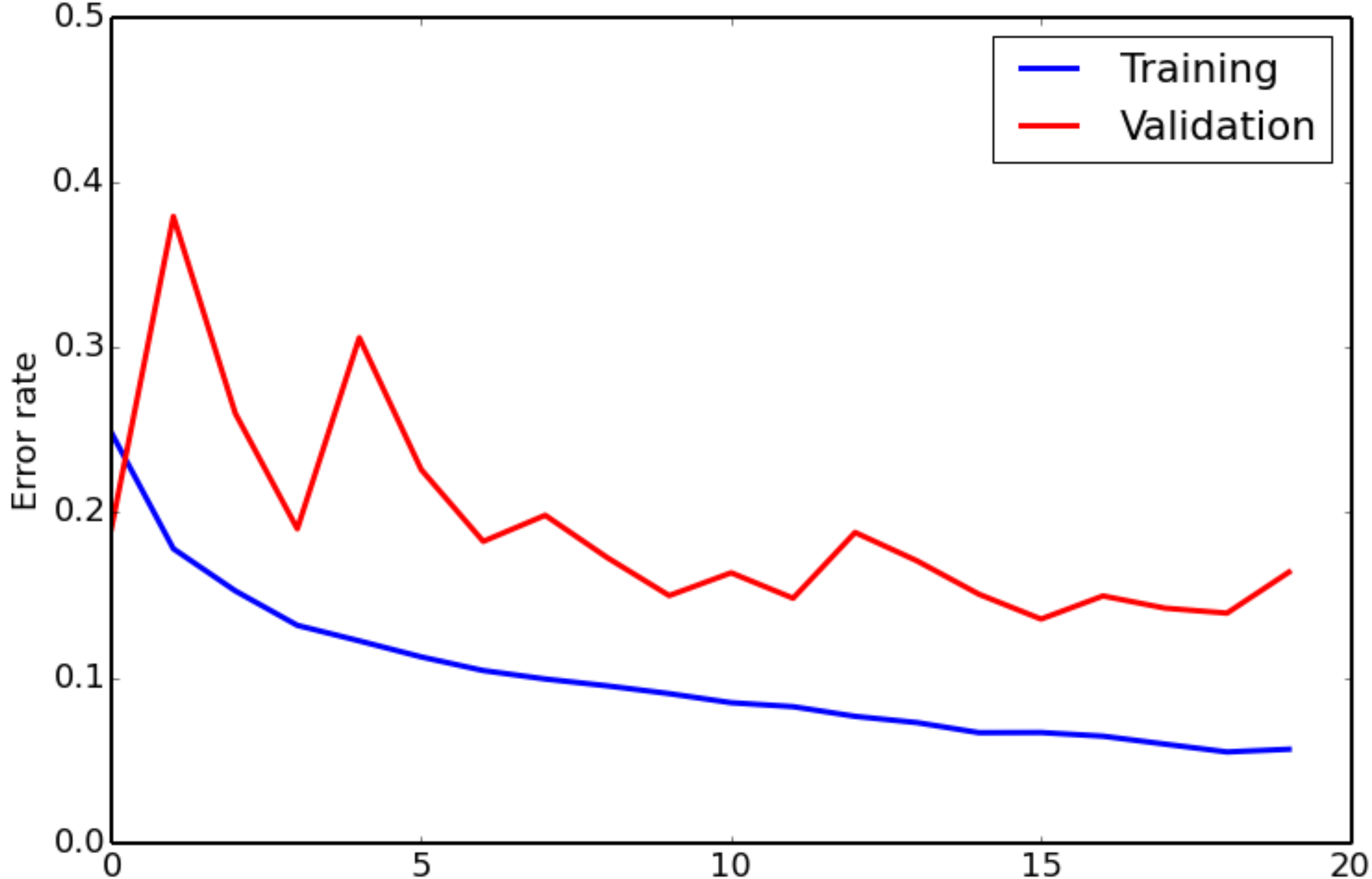
- Make prediction for current example
- Record if correct or not
- (Update model), go to next example
- At each point in time: error rate = proportion of mistakes made so far relative to total examples seen so far

Evaluation with multiple iterations

- When simulating batch learning by using **multiple iterations**, we would be evaluating on previously seen examples
- Use separate **development** set!

Learning ratings of movies in the sentiment dataset

- 25,000 movie reviews, positive and negative
- Use 5,000 for validation, 20,000 for training
- 20 iterations



Training
Validation
Error rate
0.0
0.1
0.2
0.3
0.4
0.5
0
5
10
15
20

Early stopping

- Number of iterations is a kind of hyper-parameter of the “batch” Perceptron
- Stop training when error on validation data stops dropping
- When training error goes down, but validation goes up, we're **overfitting**

Weight averaging

- We can remember the weights from each iteration, and average them
  - Creates an ensemble of perceptrons!
- In practice, such weights generalize better than the “best” single batch

The meaning of Percetron's most important features

- Bottom 10
  - waste worst poorly mess awful disappointment fails lacks annoying worse
- Top 10
  - subtitles captures enjoyable subtle noir surprisingly today excellent wonderfully perfect
- Around 0:
  - very character since during you're second stories particularly yourself hit

Agenda

- Review of models
- Parts of a perceptron
- Making decisions with a learned model
- Learning a perceptron representation
- Batch and Online learning
- Sparse and Dense representations

Sparsity

$$
\begin{pmatrix}
0 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 7 \\
0 & 0 & 0 & 5 & 0 & 0 & 0 & 8 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 4 & 0 \\
0 & 1 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\
0 & 0 & 7 & 0 & 0 & 0 & 0 & 5 & 0 & 0
\end{pmatrix}
$$

Representing text

- Text document often represented with word counts
- How many elements in the vector?
- Many tens or hundreds of thousands
- Use a **sparse representation** which omits zero values

Dense representation

	#good	#dark	#mediocre	#the
$\mathbf{x}^1 = ($	2,	0,	0,	5 $)$
$\mathbf{x}^2 = ($	0,	1,	2,	7 $)$

Sparse representation

V = (#good #dark #mediocre #the)

- $\mathbf{X}^1$ = { 0:2, 3:5 }
- $\mathbf{X}^2$ = { 1:1, 2:2, 3:7 }

- **All absent values are implicitly zero.**

Sparse vectors in Python

- Python dictionaries
- Sparse matrices in **scipy**
- Some ML algorithms, in scikit-learn and elsewhere, can be optimized to work even more efficently with sparse data

Summary

- Perceptron as a linear model for classification
- Perceptron algorithm learns from feedback on mistakes
- Online vs batch learning

Image credits

- Frank Rosenblatt http://www.rutherfordjournal.org/images/TAHC_rosenblatt-sepia.jpg