



CS 540 Introduction to Artificial Intelligence
Neural Networks I: Perceptron
University of Wisconsin-Madison

Spring 2023

Announcements

- **Homeworks:**
 - Halfway through homework assignments!
- **Class roadmap:**

Last topic
included on
midterm!



Thursday, Mar 2	ML: Neural Networks I
Tuesday, Mar 7	ML: Neural Networks II
Thursday, Mar 9	Review Lecture; Midterm 5:45-7:15pm
Tuesday, Mar 21	ML: Neural Networks III



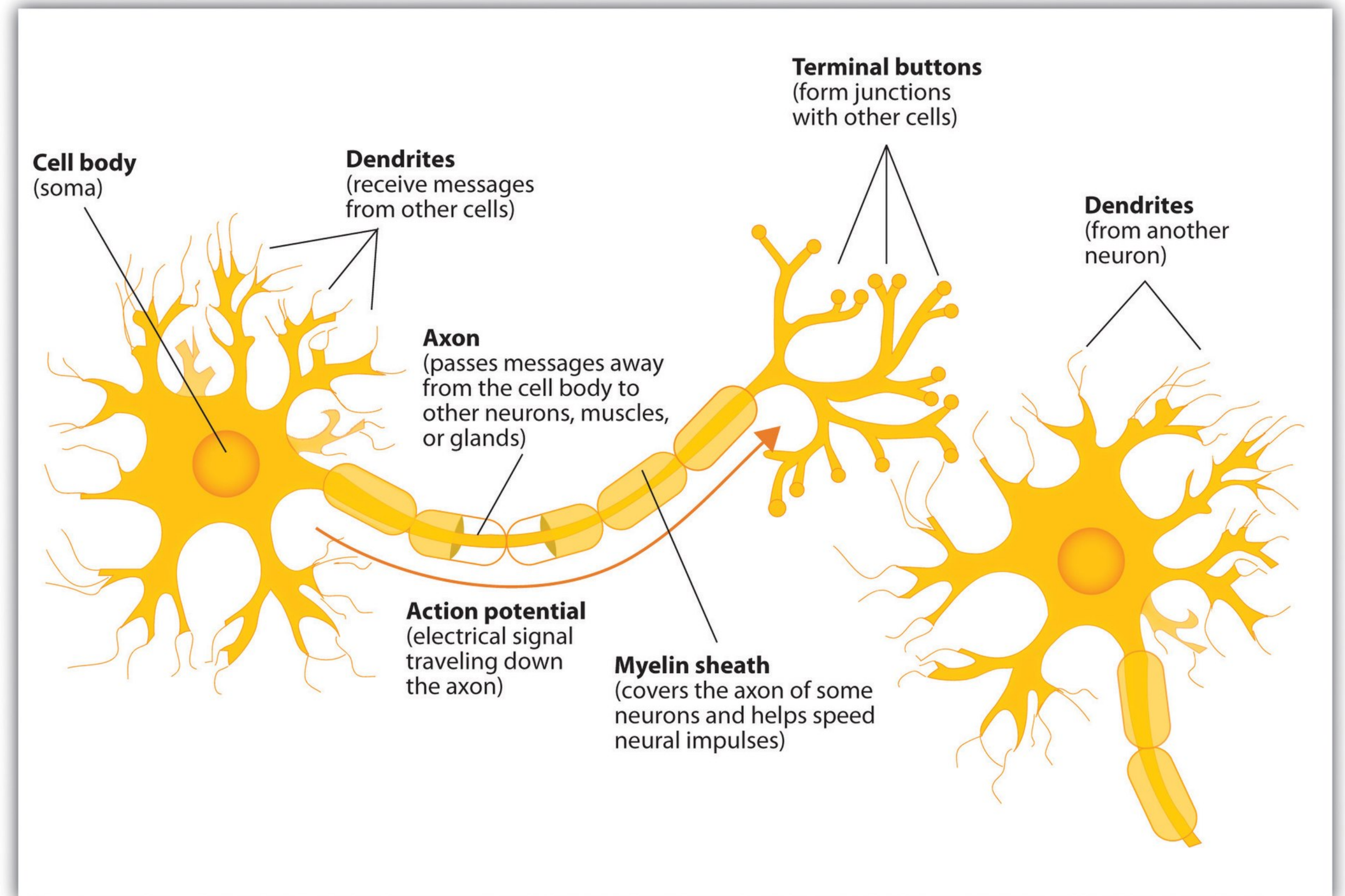
Part I: Single-layer Neural Network

Inspiration from neuroscience

- Inspirations from human brains
- Networks of **simple** and **homogenous** units



(wikipedia)



How to classify

Cats vs. dogs?

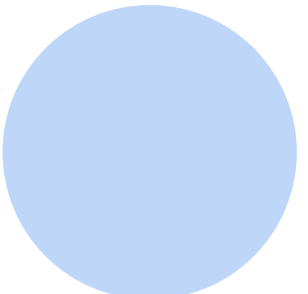
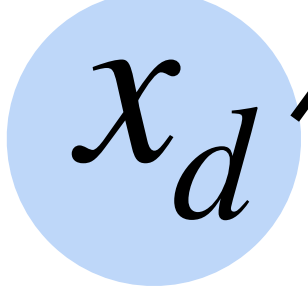
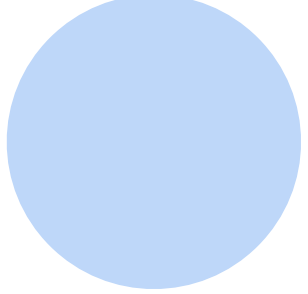
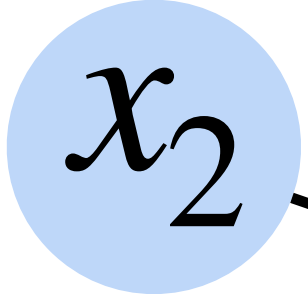
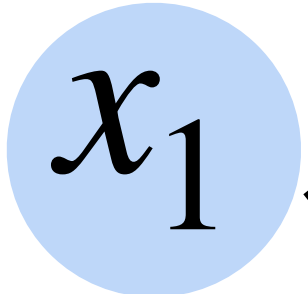


Perceptron

Cats vs. dogs?



Input



Output

Linear Perceptron

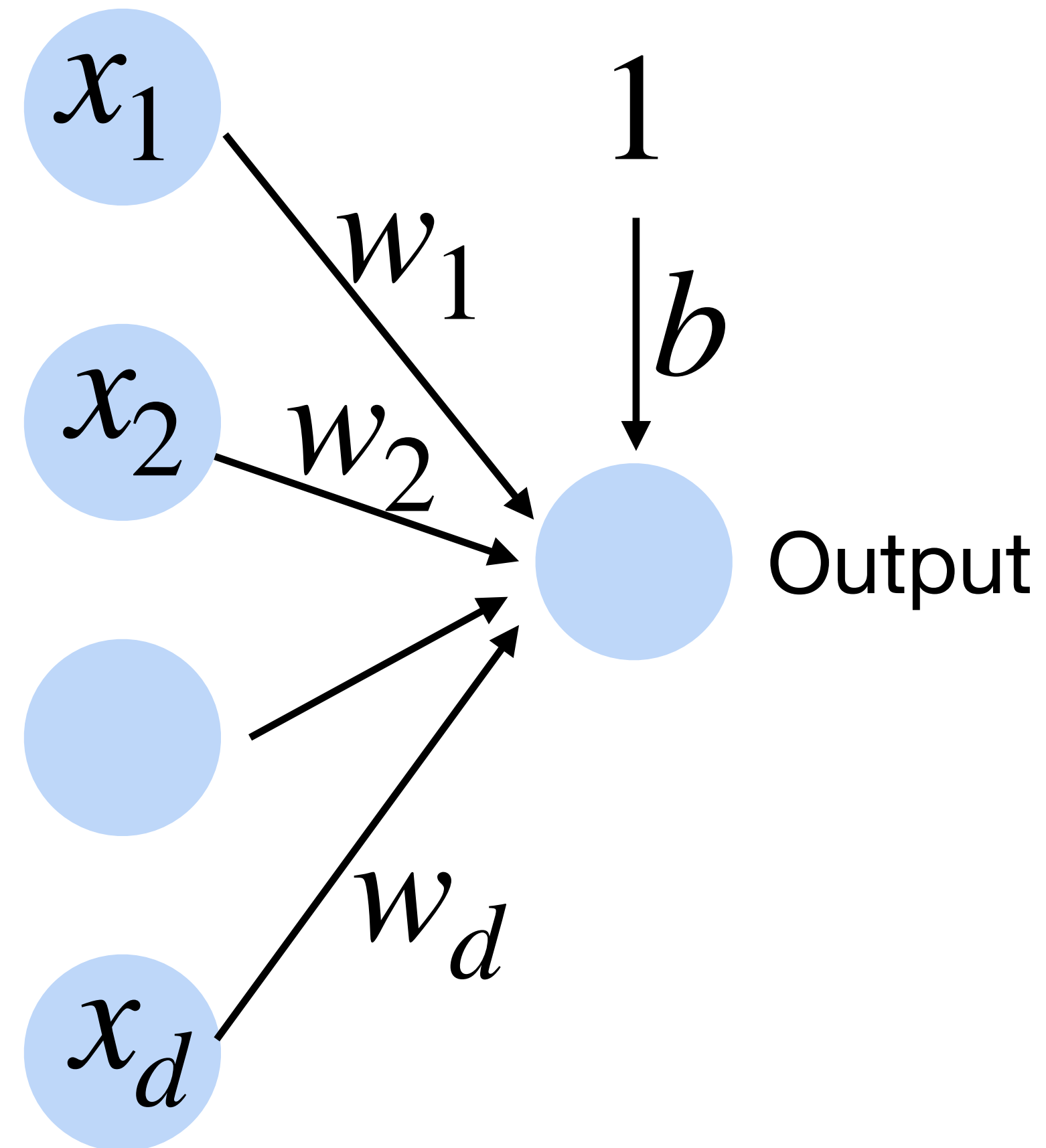
- Given input \mathbf{x} , weight \mathbf{w} and bias b , perceptron outputs:

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

Cats vs. dogs?



Input



Perceptron

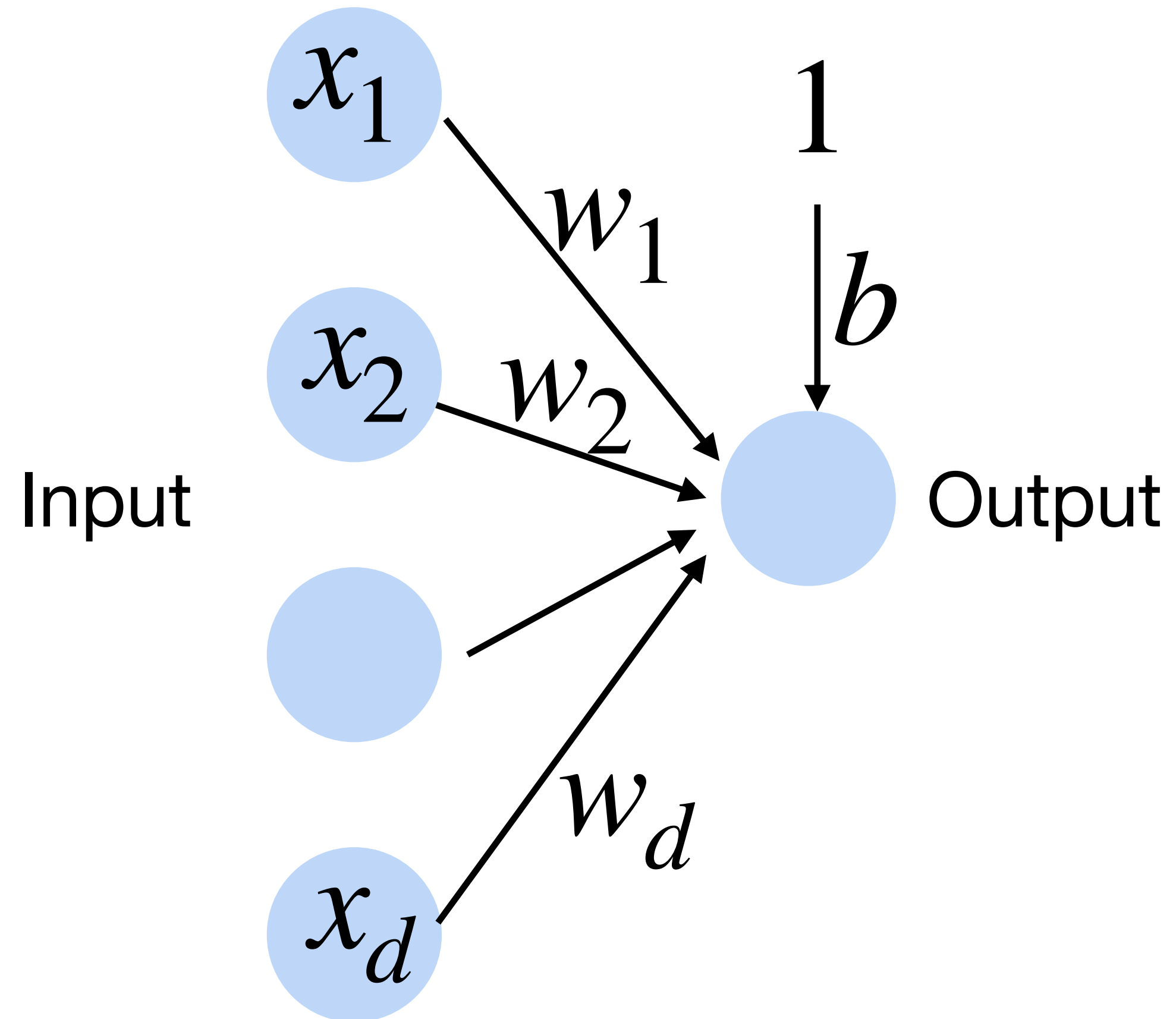
- Given input \mathbf{x} , weight \mathbf{w} and bias b , perceptron outputs:

$$o = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle + b)$$

$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Activation function

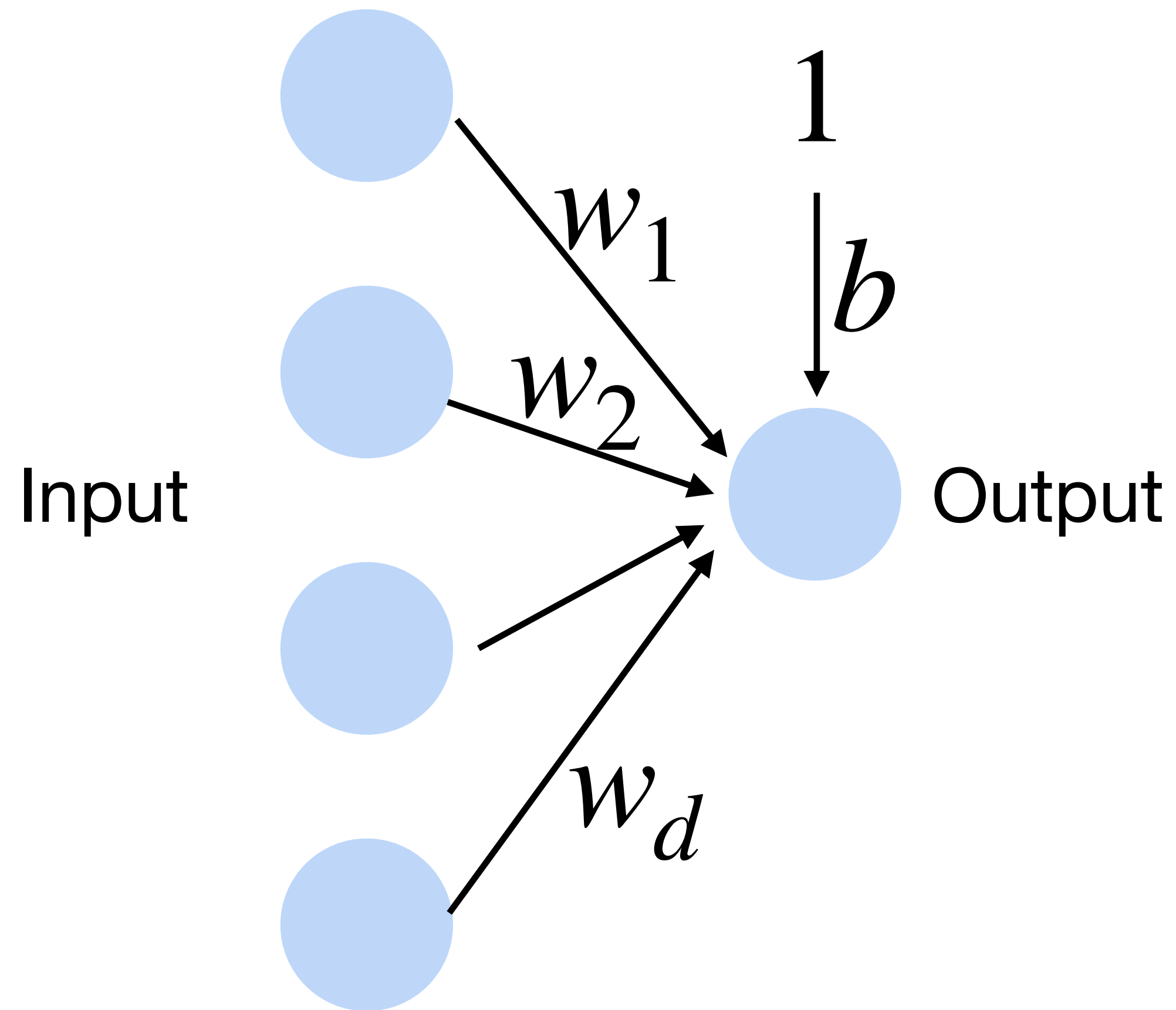
Cats vs. dogs?



Perceptron

- Goal: learn parameters $\mathbf{w} = \{w_1, w_2, \dots, w_d\}$ and b to minimize the classification error

Cats vs. dogs?



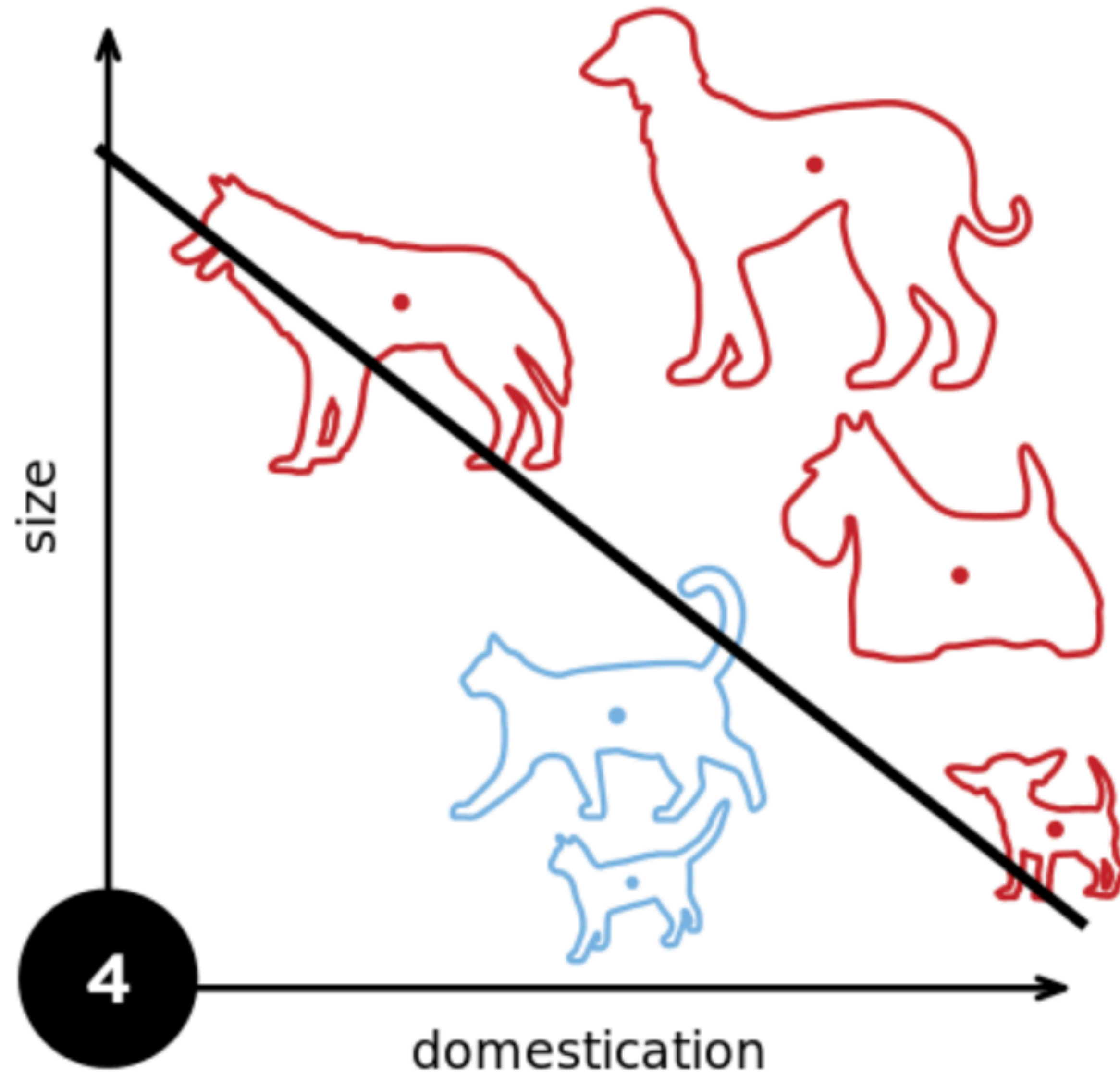
Training the Perceptron

x augmented with dimension of constant 1

Perceptron Algorithm

```
Initialize  $\vec{w} = \vec{0}$  // Initialize  $\vec{w}$ .  $\vec{w} = \vec{0}$  misclassifies everything.
while TRUE do // Keep looping
   $m = 0$  // Count the number of misclassifications,  $m$ 
  for  $(x_i, y_i) \in D$  do // Loop over each (data, label) pair in the dataset,  $D$ 
    if  $o_i \neq y_i$  then // If the pair  $(\vec{x}_i, y_i)$  is misclassified
       $\vec{w} \leftarrow \vec{w} + x_i$  if  $y_i = 1$ ,  $\vec{w} \leftarrow \vec{w} - x_i$  if  $y_i = 0$ 
       $m \leftarrow m + 1$  // Counter the number of misclassification
    end if
  end for
  if  $m = 0$  then // If the most recent  $\vec{w}$  gave 0 misclassifications
    break // Break out of the while-loop
  end if
end while // Otherwise, keep looping!
```


Perceptron



Example 2: Predict whether a user likes a song or not



model



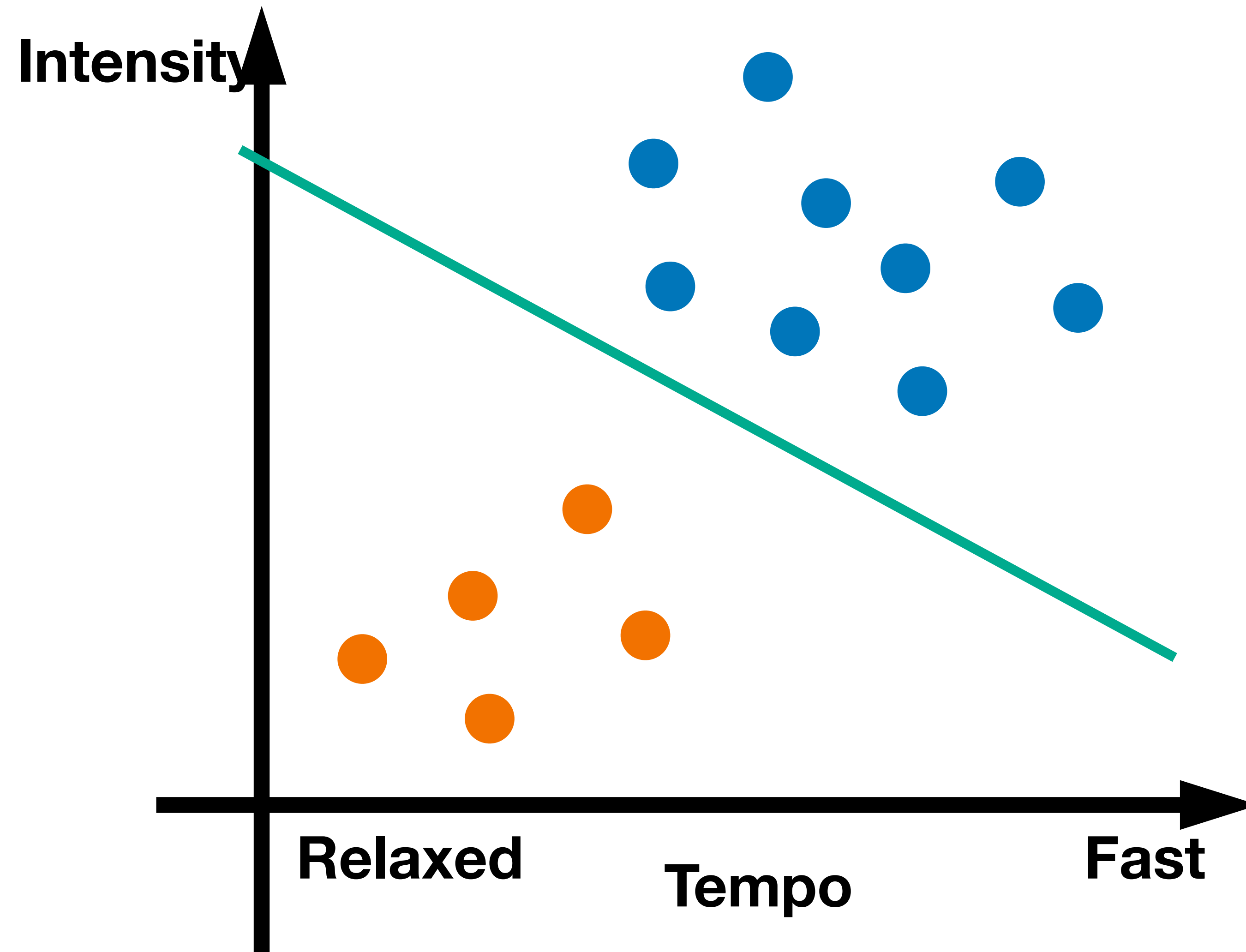
Example 2: Predict whether a user likes a song or not Using Perceptron



User Sharon

● DisLike

● Like



Learning AND function using perceptron

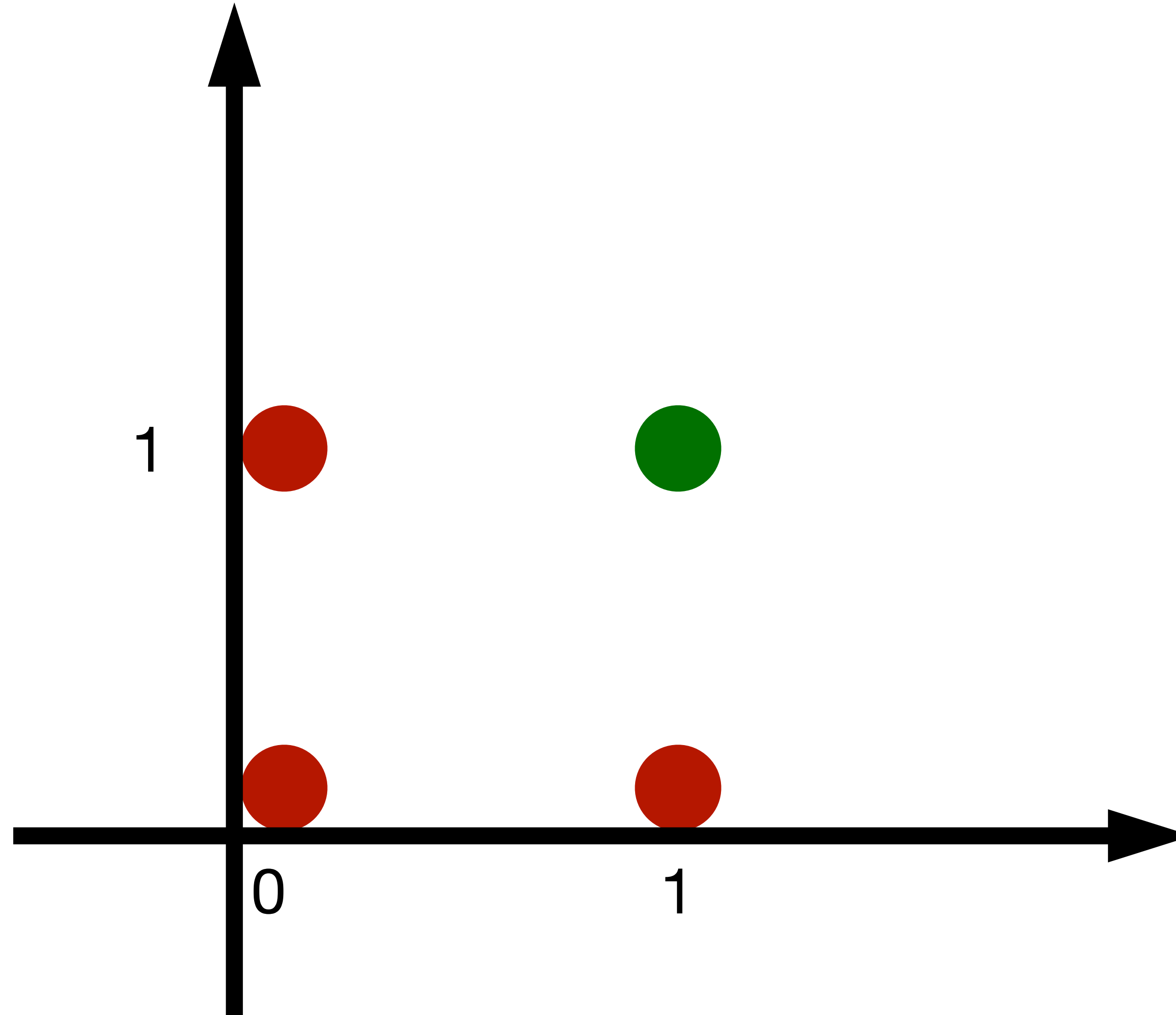
The perceptron can learn an AND function

$$x_1 = 1, x_2 = 1, y = 1$$

$$x_1 = 1, x_2 = 0, y = 0$$

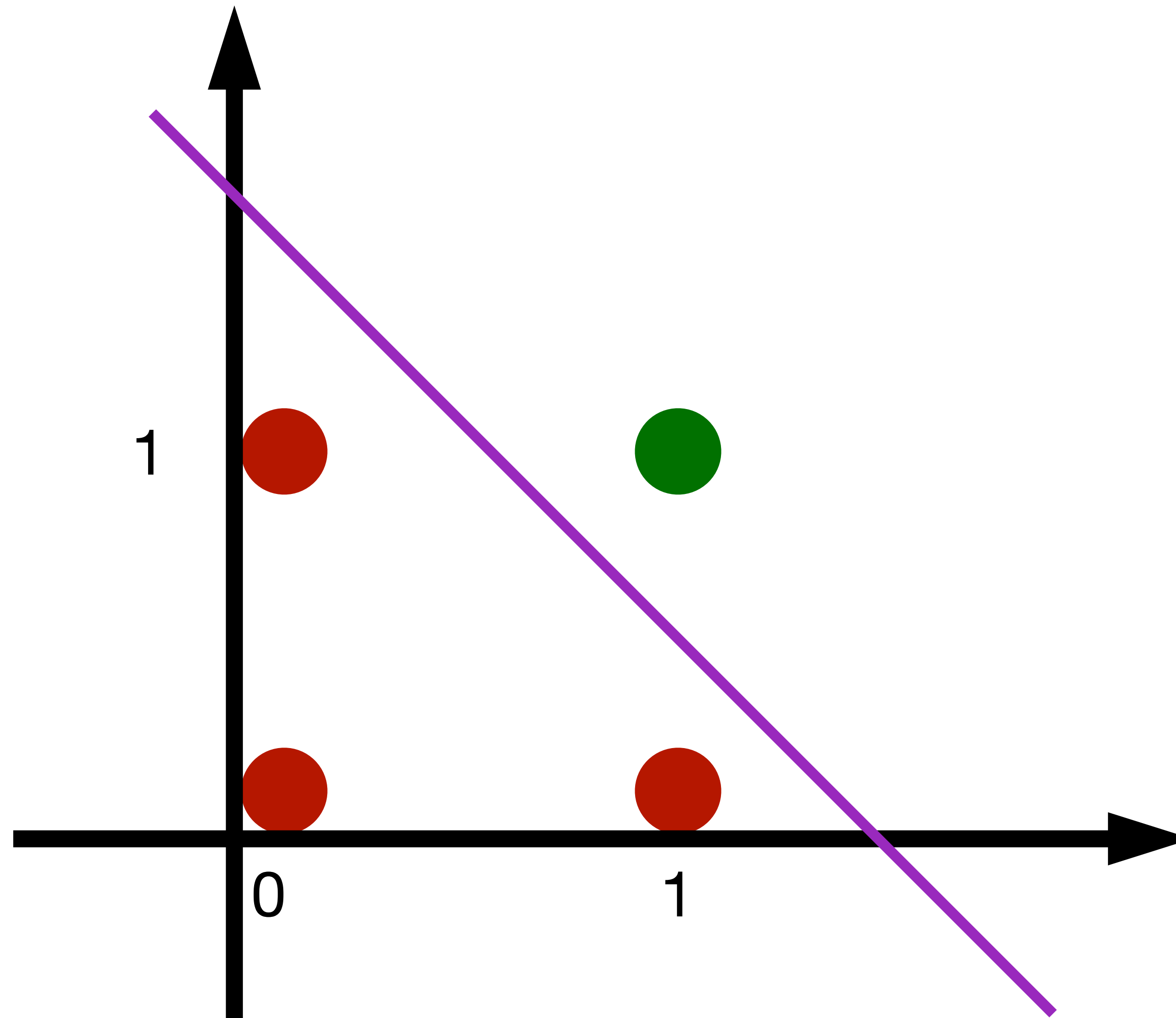
$$x_1 = 0, x_2 = 1, y = 0$$

$$x_1 = 0, x_2 = 0, y = 0$$



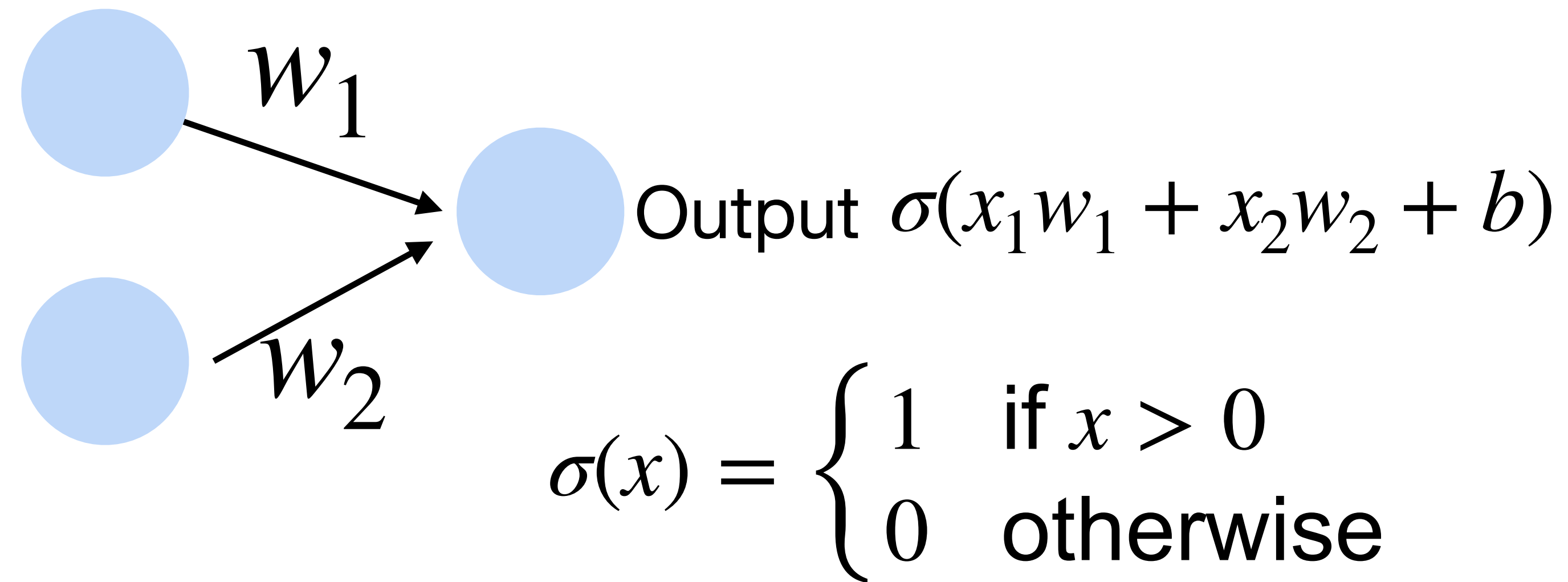
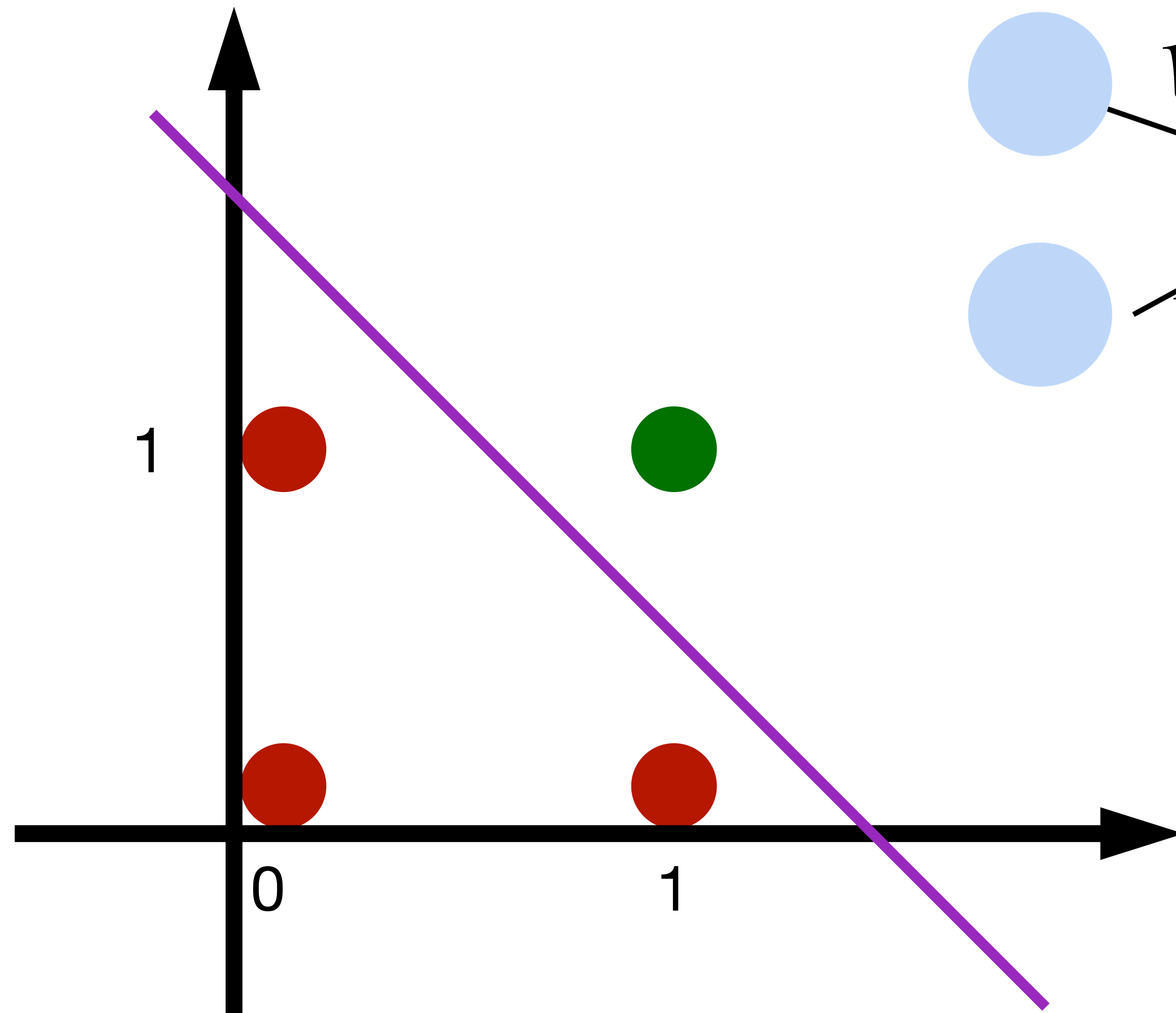
Learning AND function using perceptron

The perceptron can learn an AND function



Learning AND function using perceptron

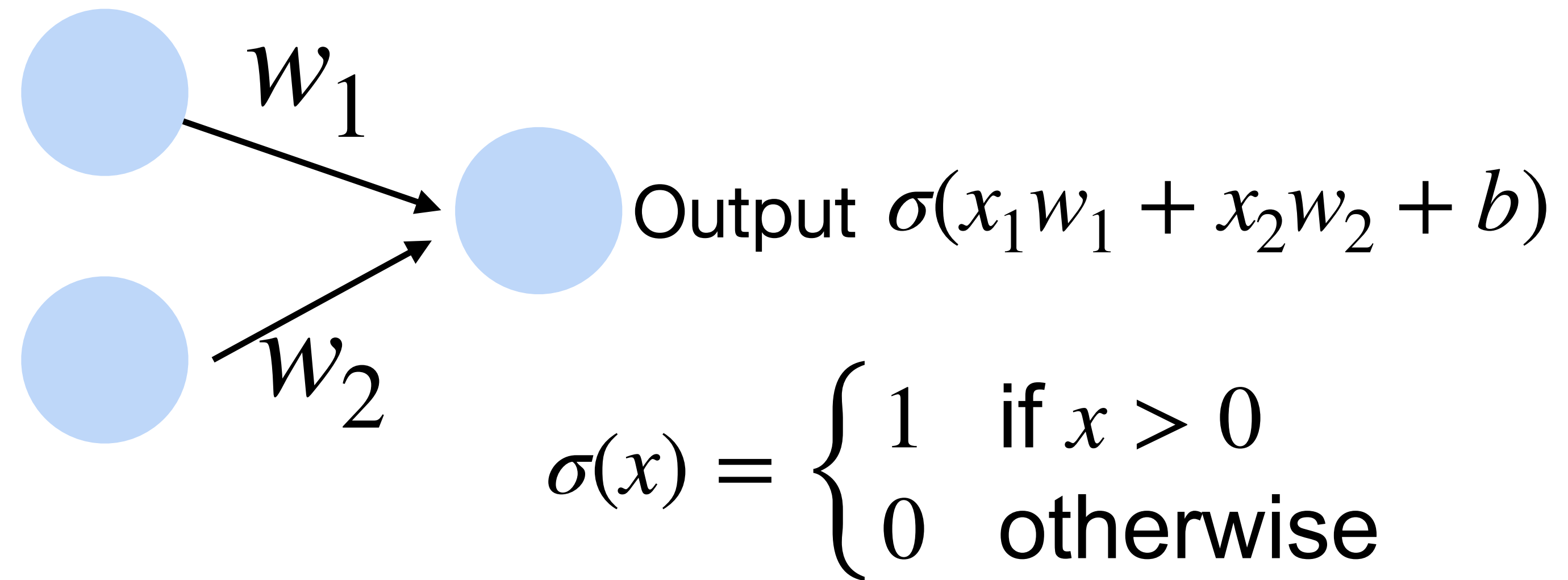
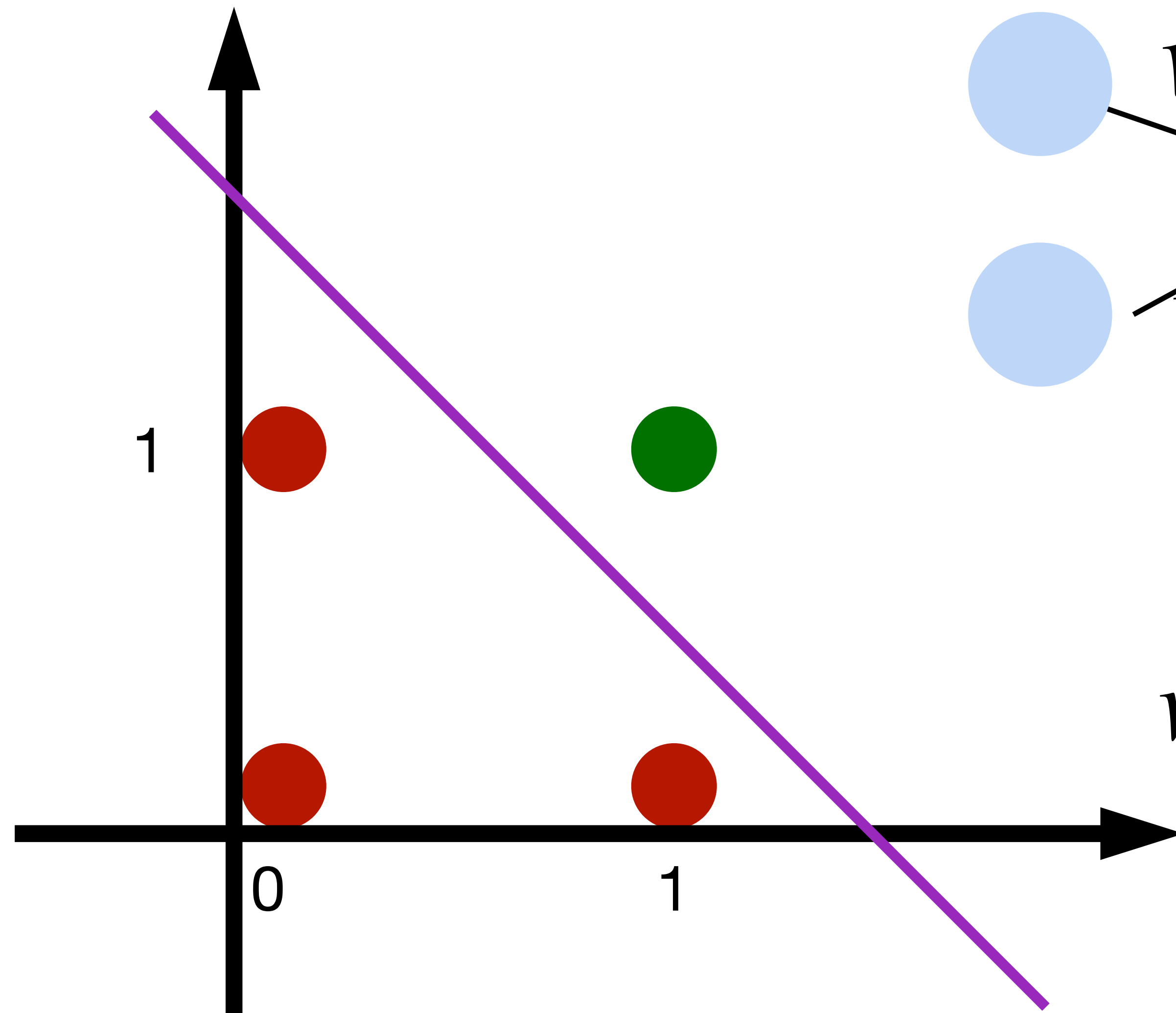
The perceptron can learn an AND function



What's w and b ?

Learning AND function using perceptron

The perceptron can learn an AND function



$$w_1 = 1, w_2 = 1, b = -1.5$$

Learning OR function using perceptron

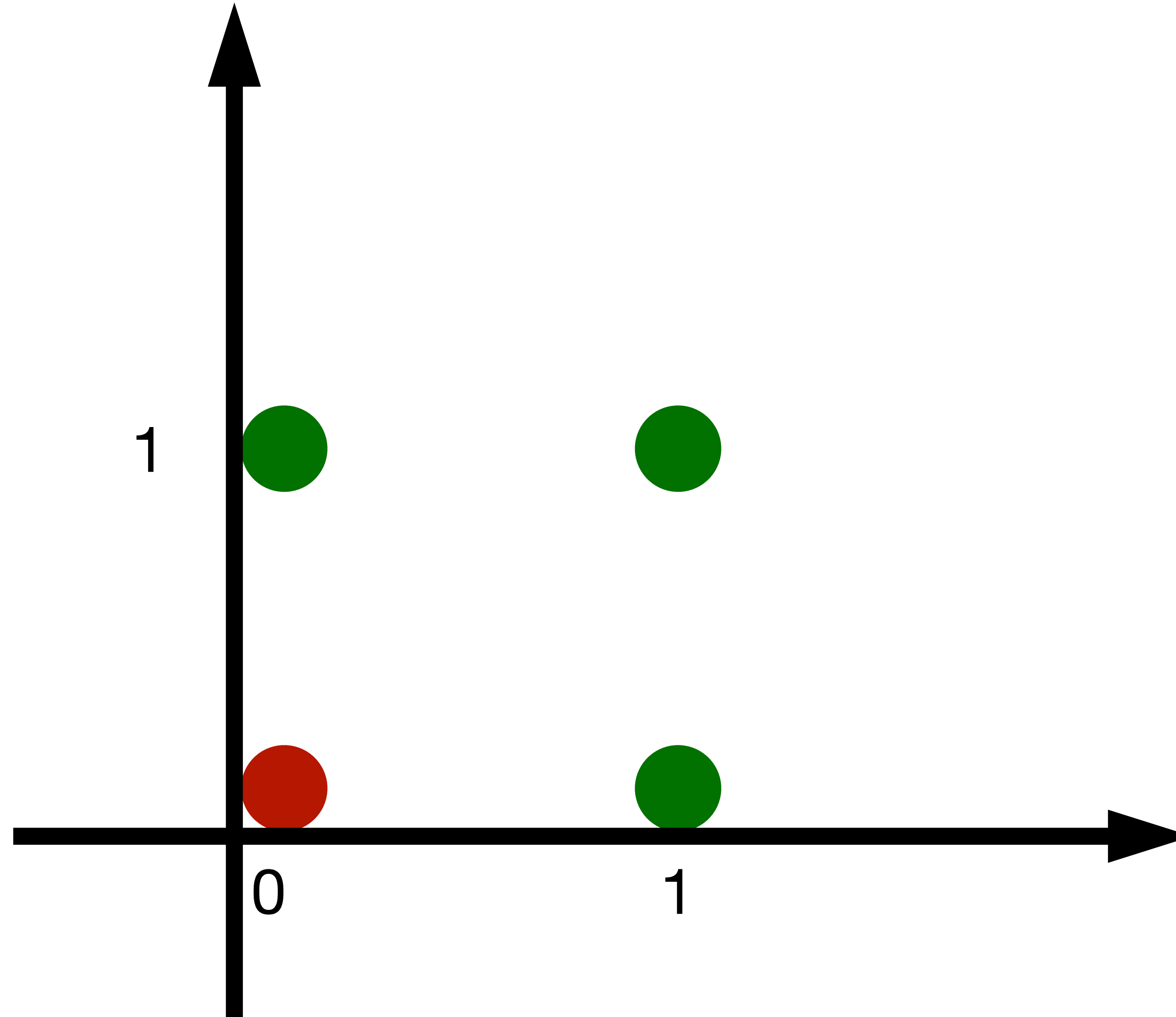
The perceptron can learn an OR function

$$x_1 = 1, x_2 = 1, y = 1$$

$$x_1 = 1, x_2 = 0, y = 1$$

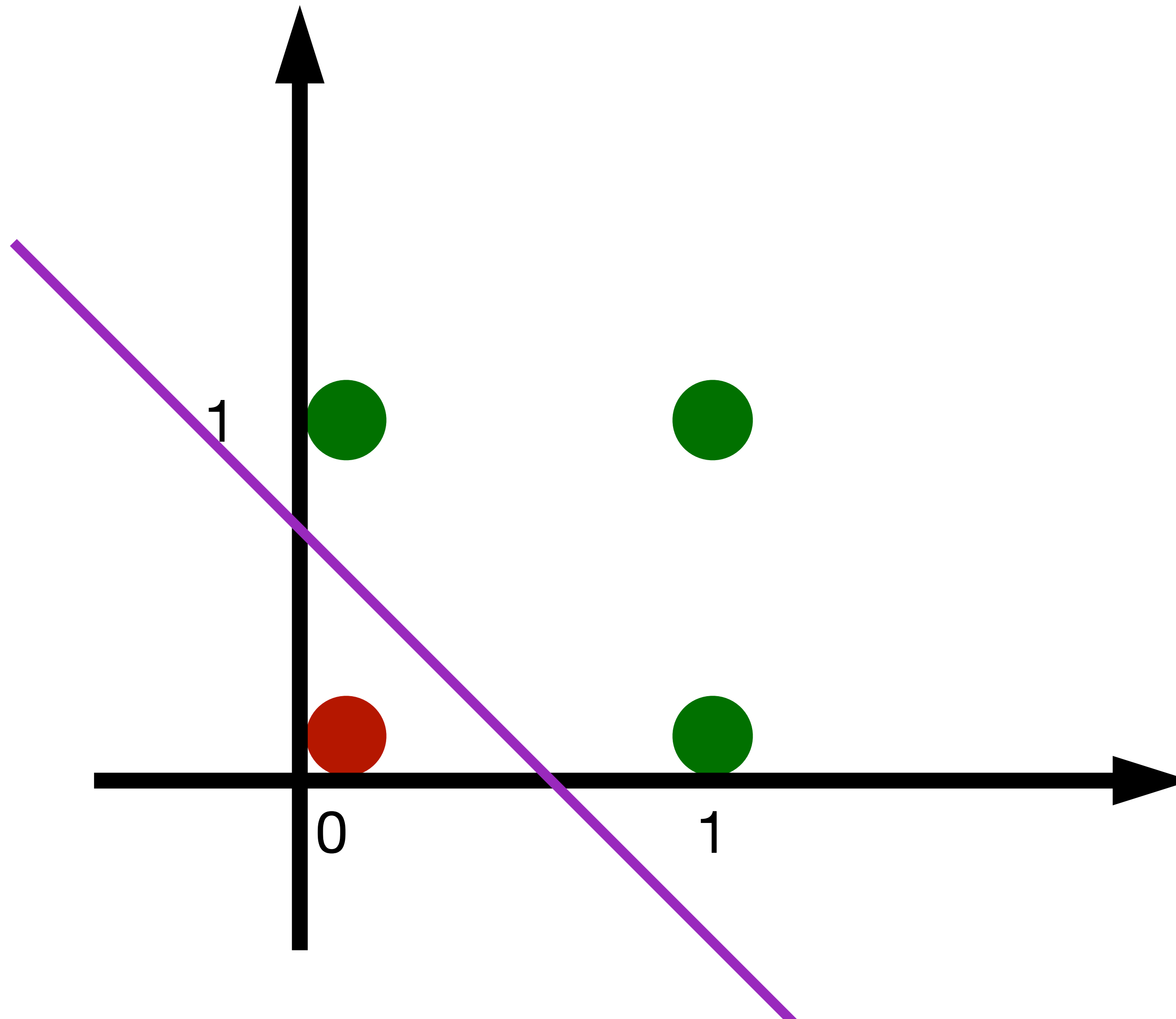
$$x_1 = 0, x_2 = 1, y = 1$$

$$x_1 = 0, x_2 = 0, y = 0$$



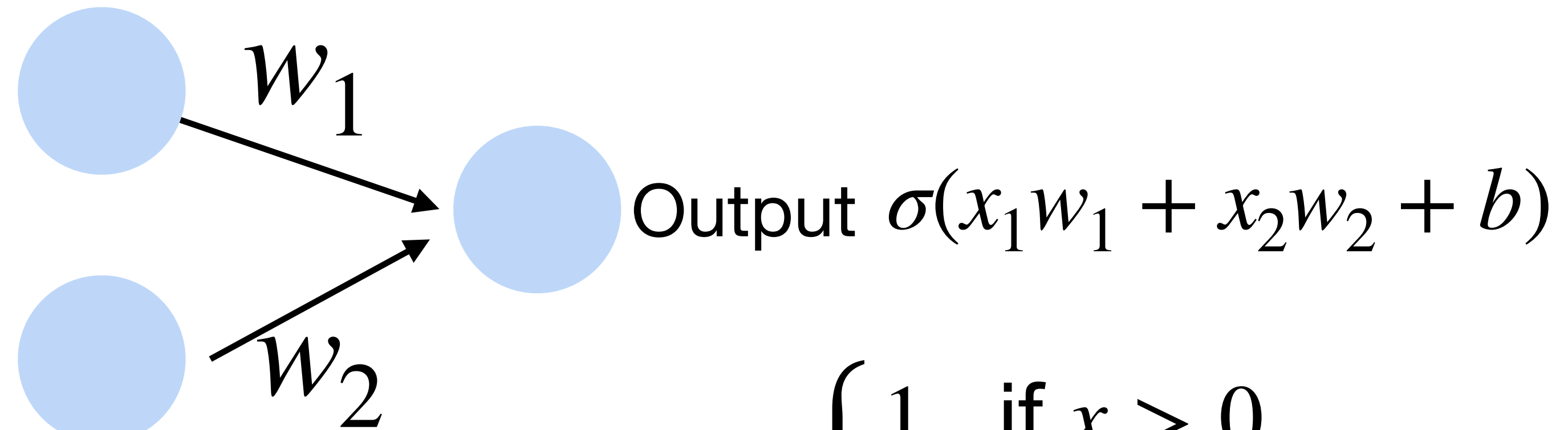
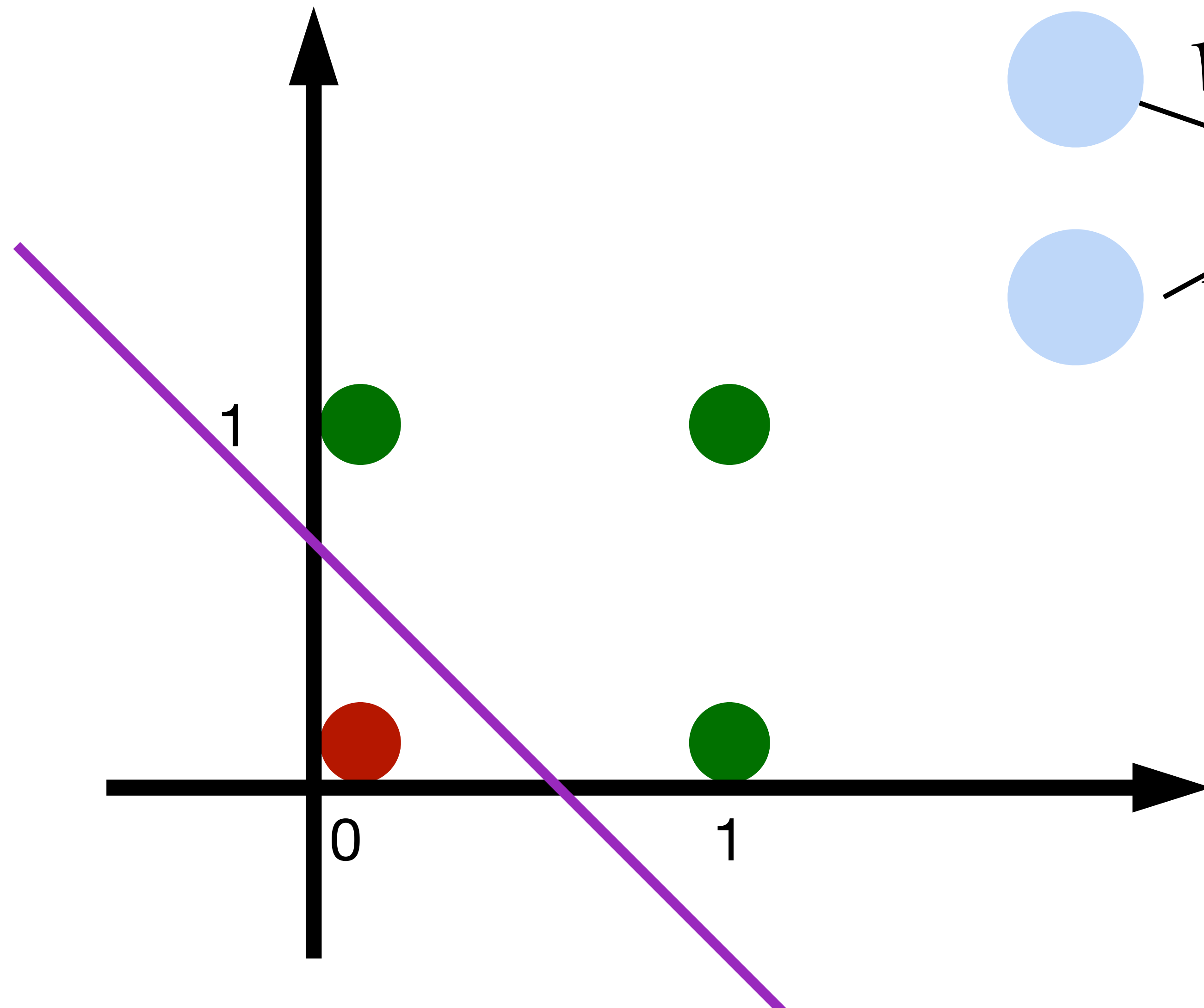
Learning OR function using perceptron

The perceptron can learn an OR function



Learning OR function using perceptron

The perceptron can learn an OR function

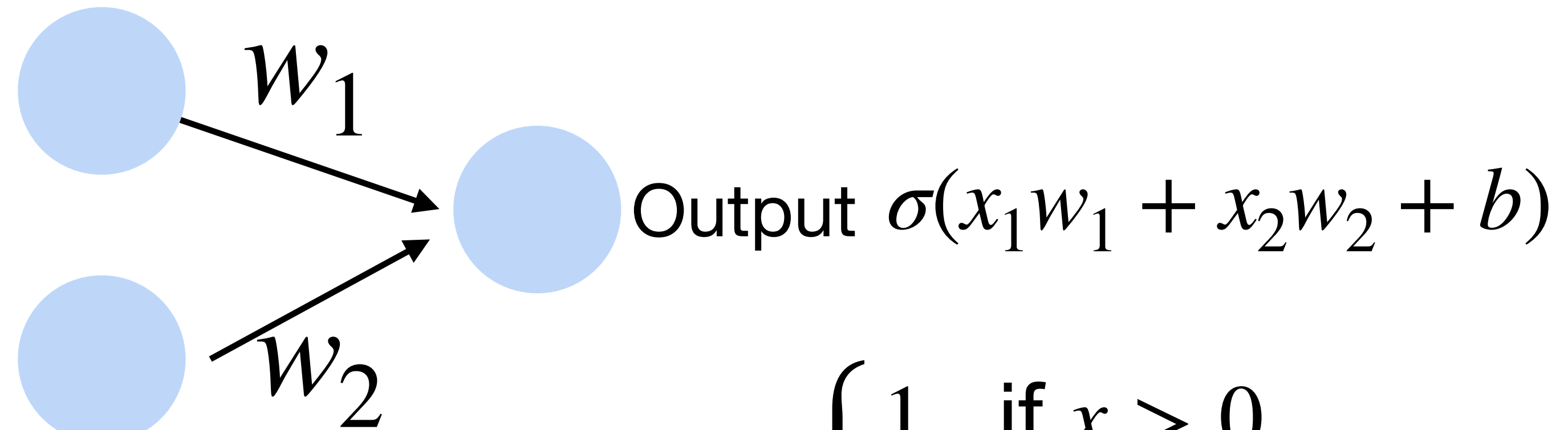
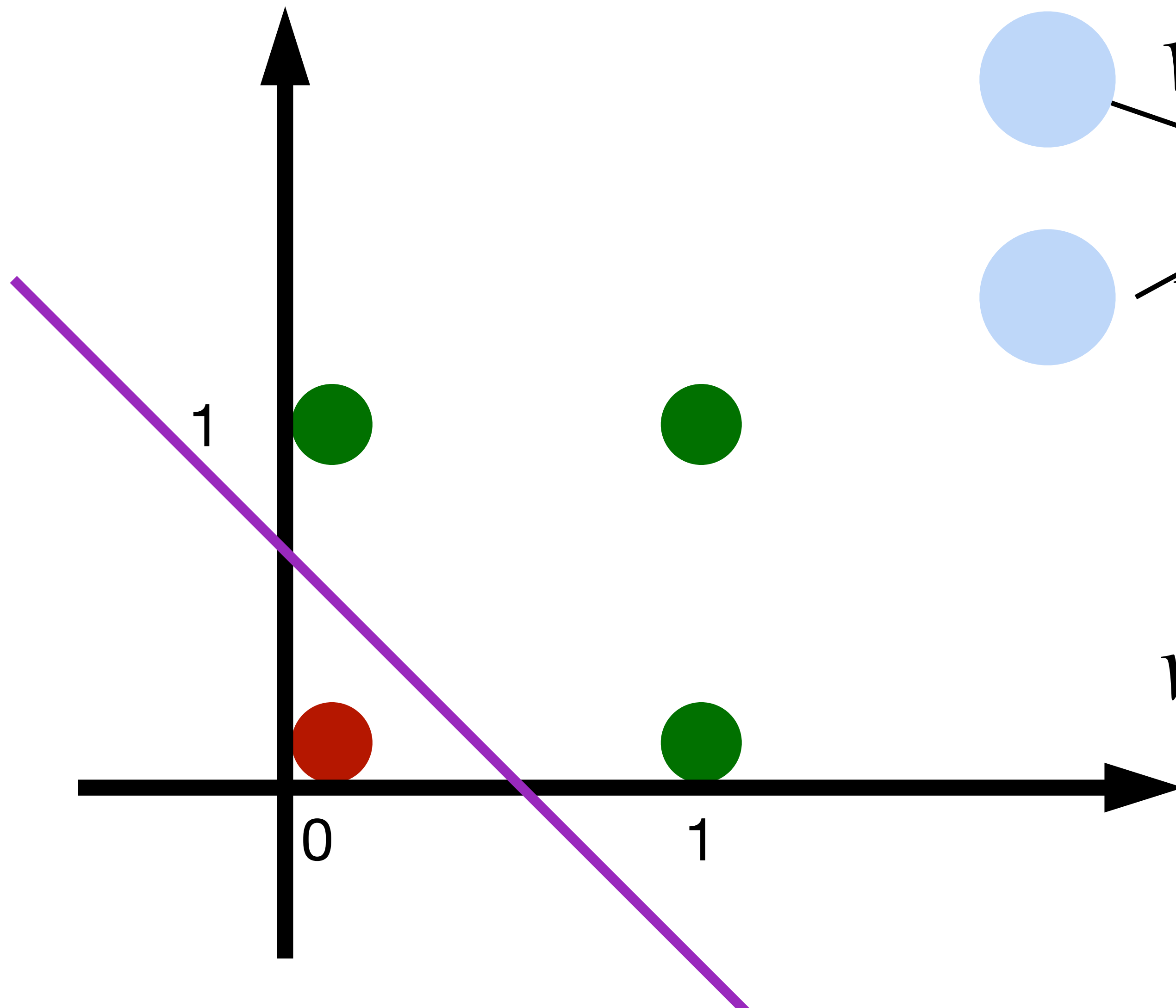


$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

What's w and b ?

Learning OR function using perceptron

The perceptron can learn an OR function



$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = 1, w_2 = 1, b = -0.5$$

Learning NOT function using perceptron

The perceptron can learn NOT function (single input)



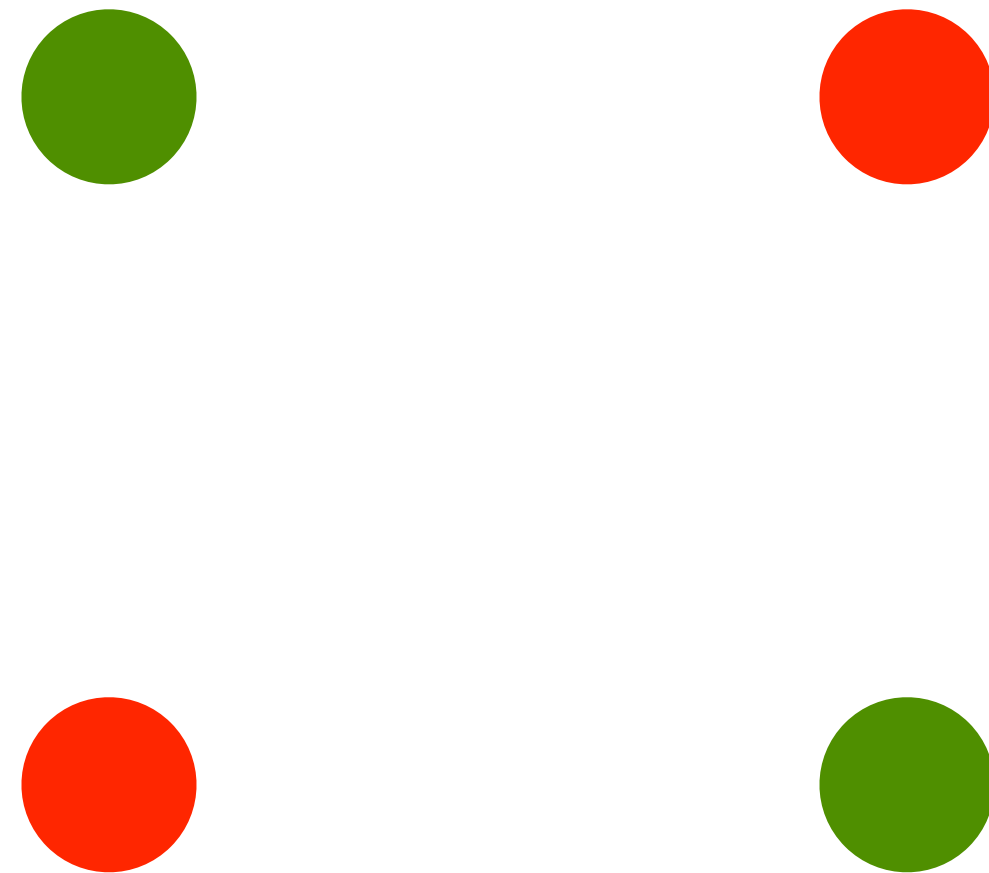
$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = -1, b = 0.5$$



XOR Problem (Minsky & Papert, 1969)

The perceptron cannot learn an XOR function
(it can only generate linear separators)



This contributed to the first AI winter

Quiz Break

Consider the linear perceptron with x as the input. Which function can the linear perceptron compute?

(1) $y = ax + b$

(2) $y = ax^2 + bx + c$

A. (1)

B. (2)

C. (1)(2)

D. None of the above

Quiz Break

Consider the linear perceptron with x as the input. Which function can the linear perceptron compute?

(1) $y = ax + b$

(2) $y = ax^2 + bx + c$

A. (1)

B. (2)

C. (1)(2)

D. None of the above

Answer: A. All units in a linear perceptron are linear. Thus, the model can not represent non-linear functions.

Quiz Break

Perceptron can be used for representing:

- A. AND function
- B. OR function
- C. XOR function
- D. Both AND and OR function

Quiz Break

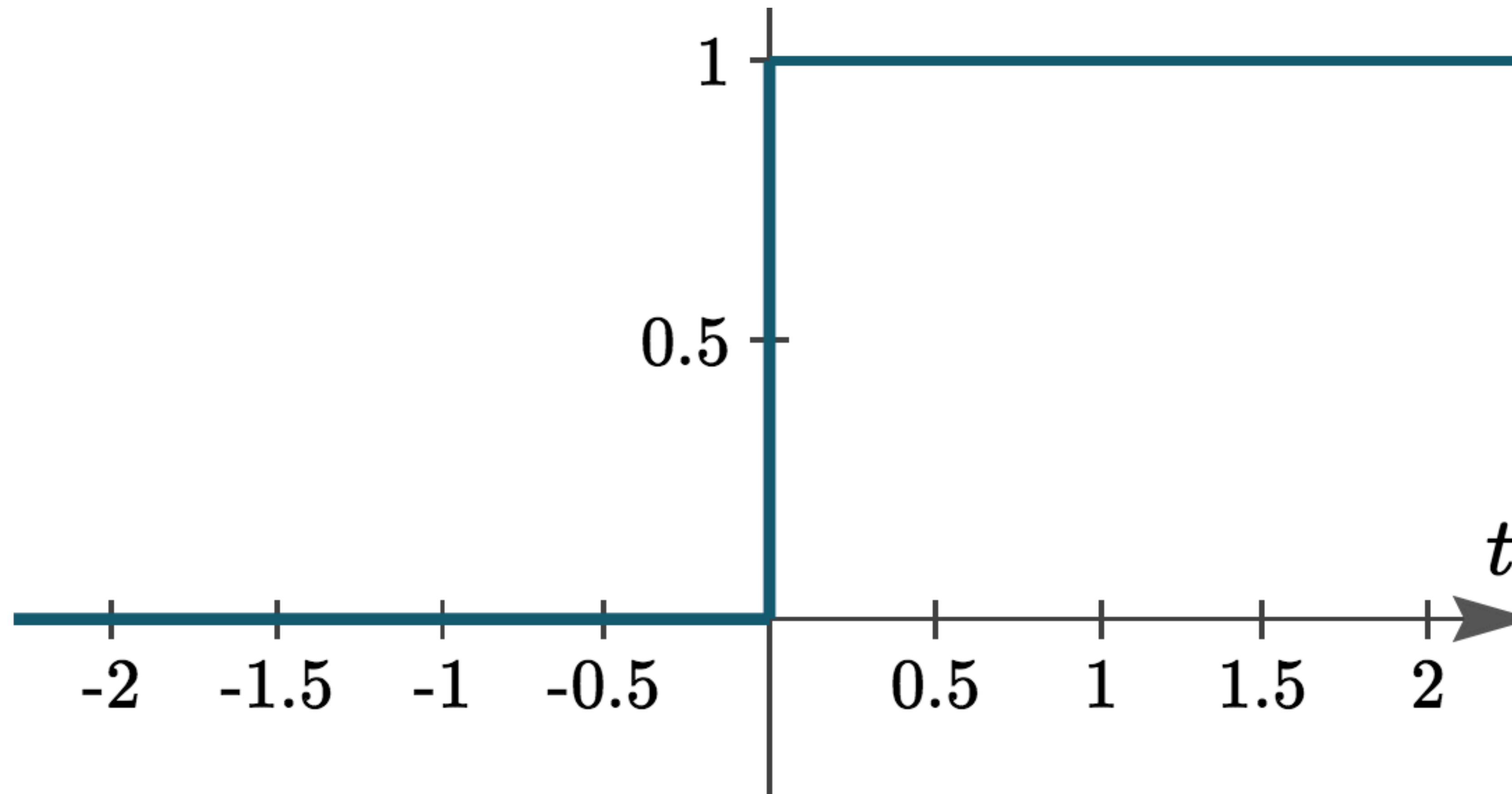
Perceptron can be used for representing:

- A. AND function
- B. OR function
- C. XOR function
- D. Both AND and OR function

Step Function activation

Step function is discontinuous, which cannot be used for gradient descent

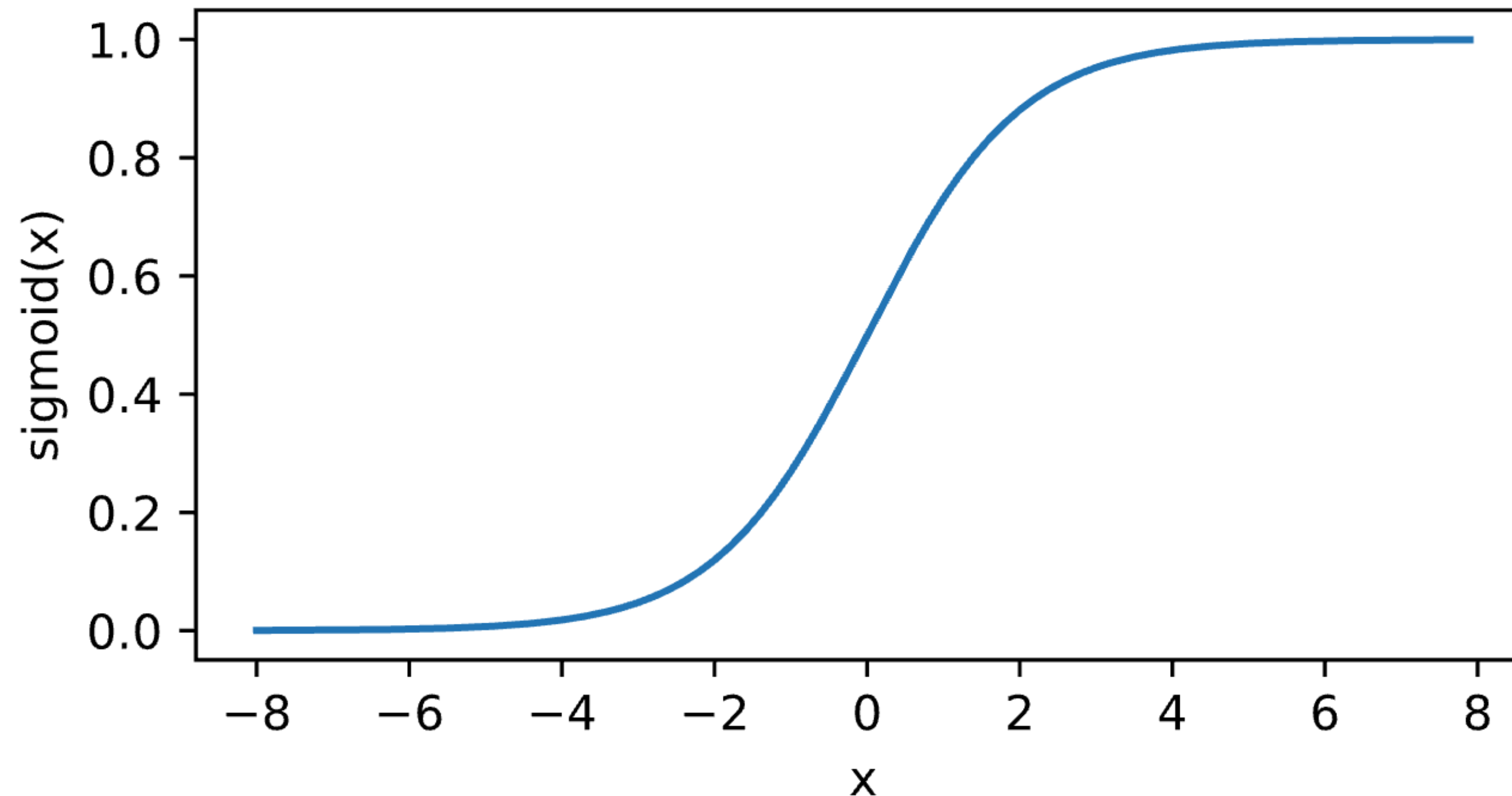
$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$



Sigmoid/Logistic Activation

Map input into $[0, 1]$, a **soft** version of $\sigma(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases}$

$$\sigma(z) = \text{sigmoid}(z) = \frac{1}{1 + \exp(-z)}$$

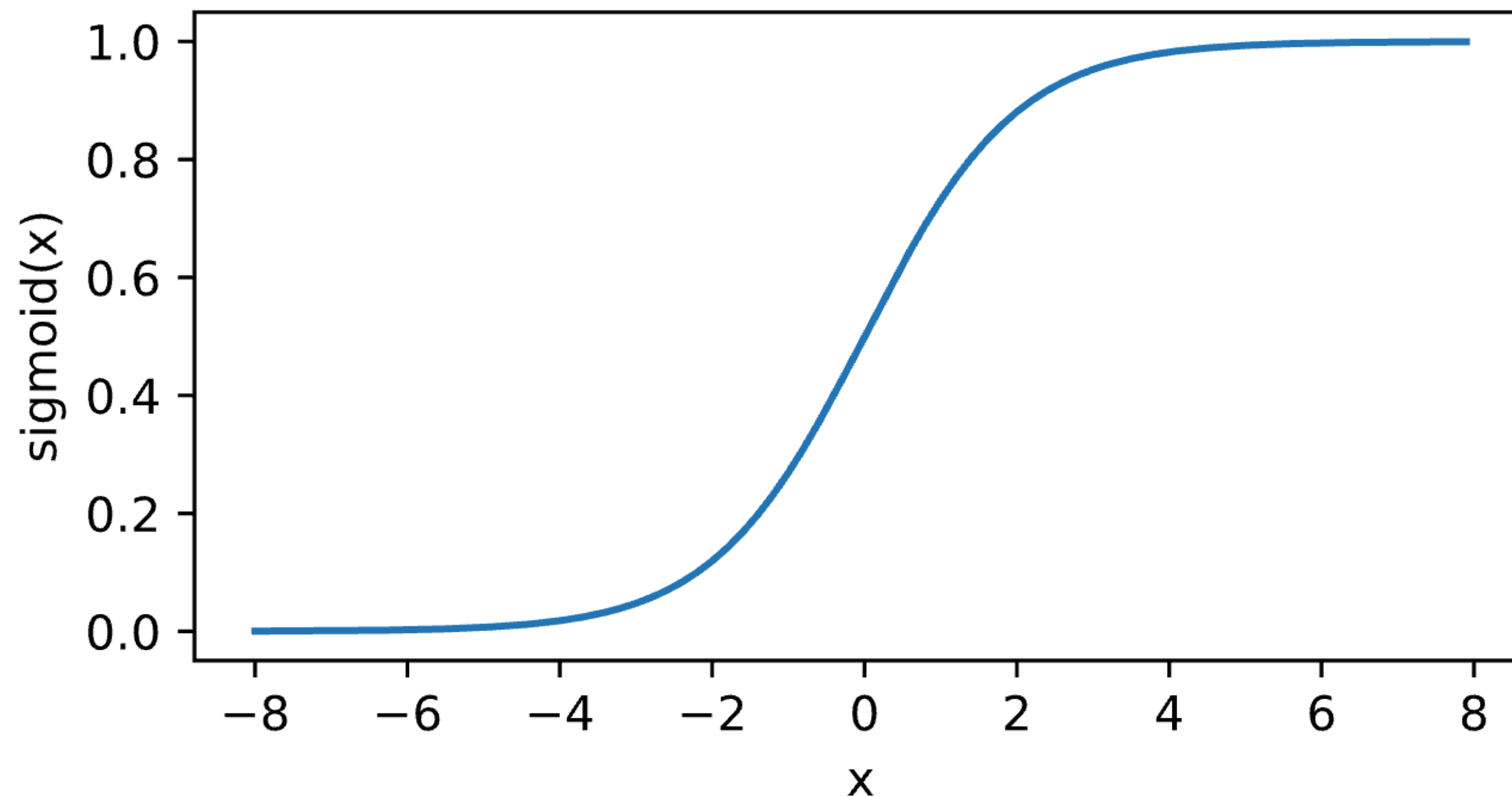


Logistic regression

$$\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$$

$$p(y = 1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$

$$p(y = -1 | \mathbf{x}) = 1 - \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^T \mathbf{x})}$$



Logistic regression

Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ $\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

Training: maximize likelihood estimate (on the conditional probability)

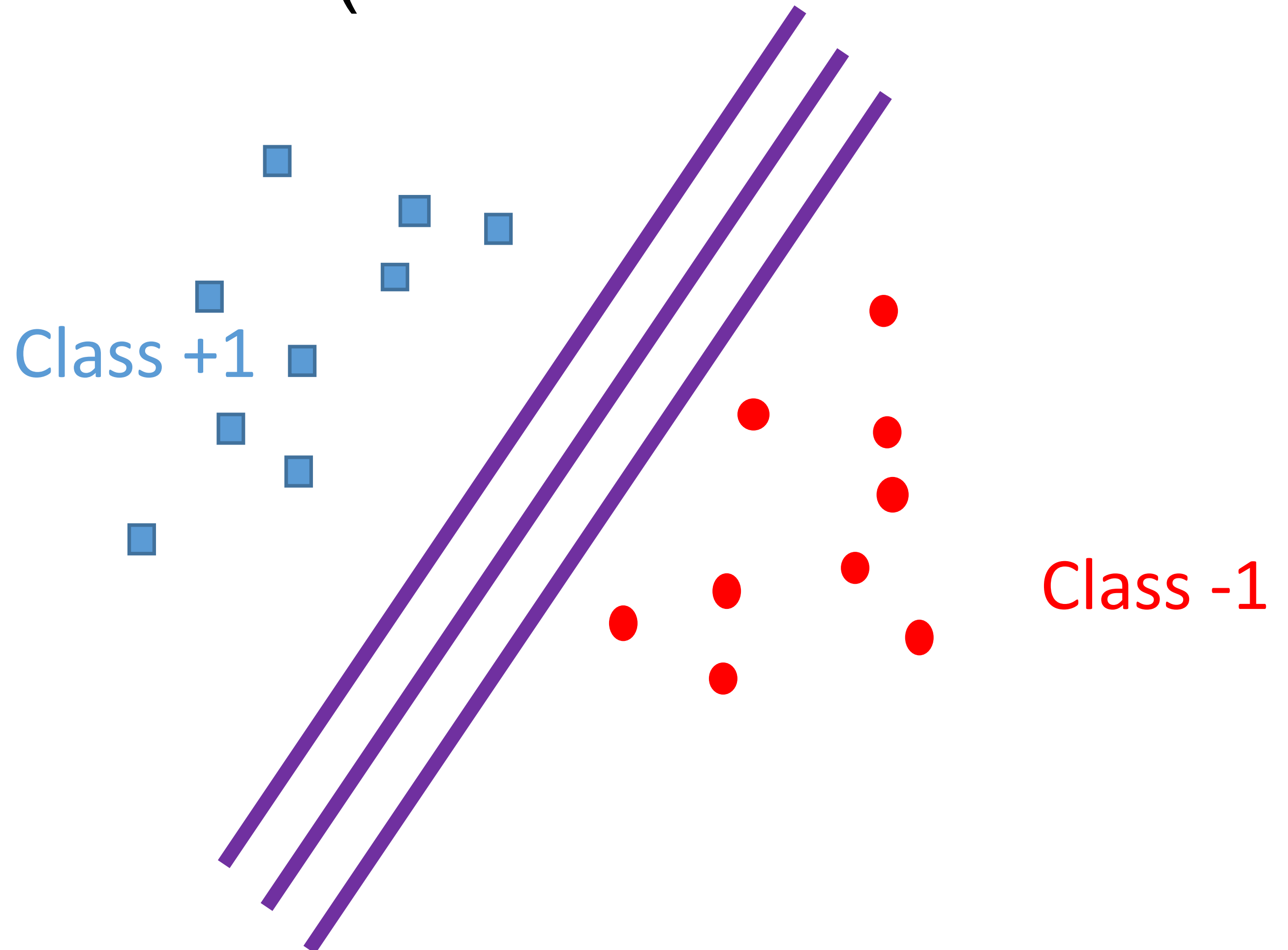
$$\max_{\mathbf{w}} \sum_i \log \frac{1}{1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)}$$

Logistic regression

Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ $\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

Training: maximize likelihood estimate (on the conditional probability)

When training data is linearly separable, many solutions



Logistic regression

Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ $\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

Training: maximum A posteriori (MAP)

$$\min_{\mathbf{w}} \sum_i -\log \frac{1}{1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)} + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

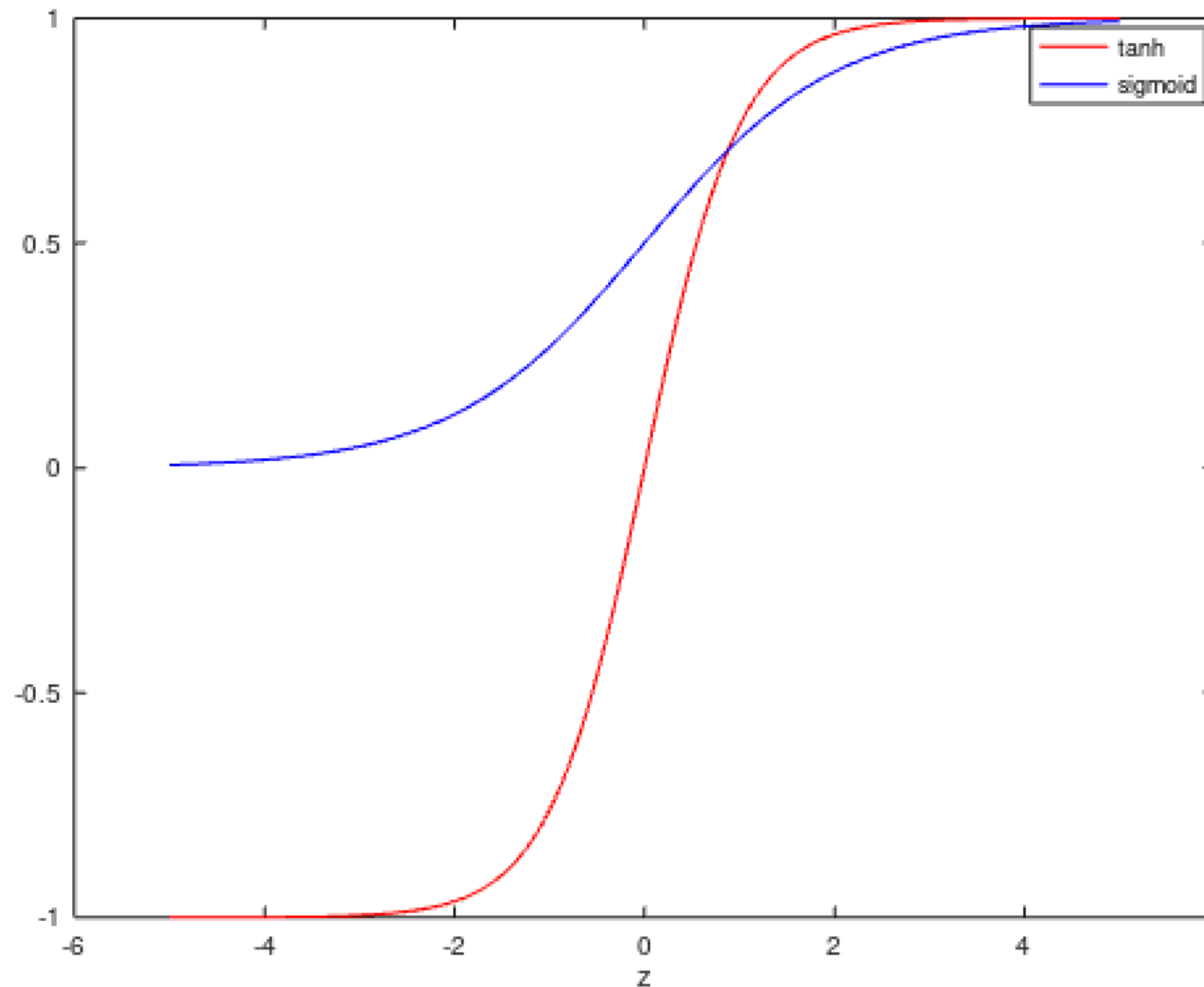
- Convex optimization
- Solve via (stochastic) gradient descent

Tanh Activation

Map inputs into $(-1, 1)$

$$\sigma(z) = \tanh(z) = \frac{1 - \exp(-2z)}{1 + \exp(-2z)}$$

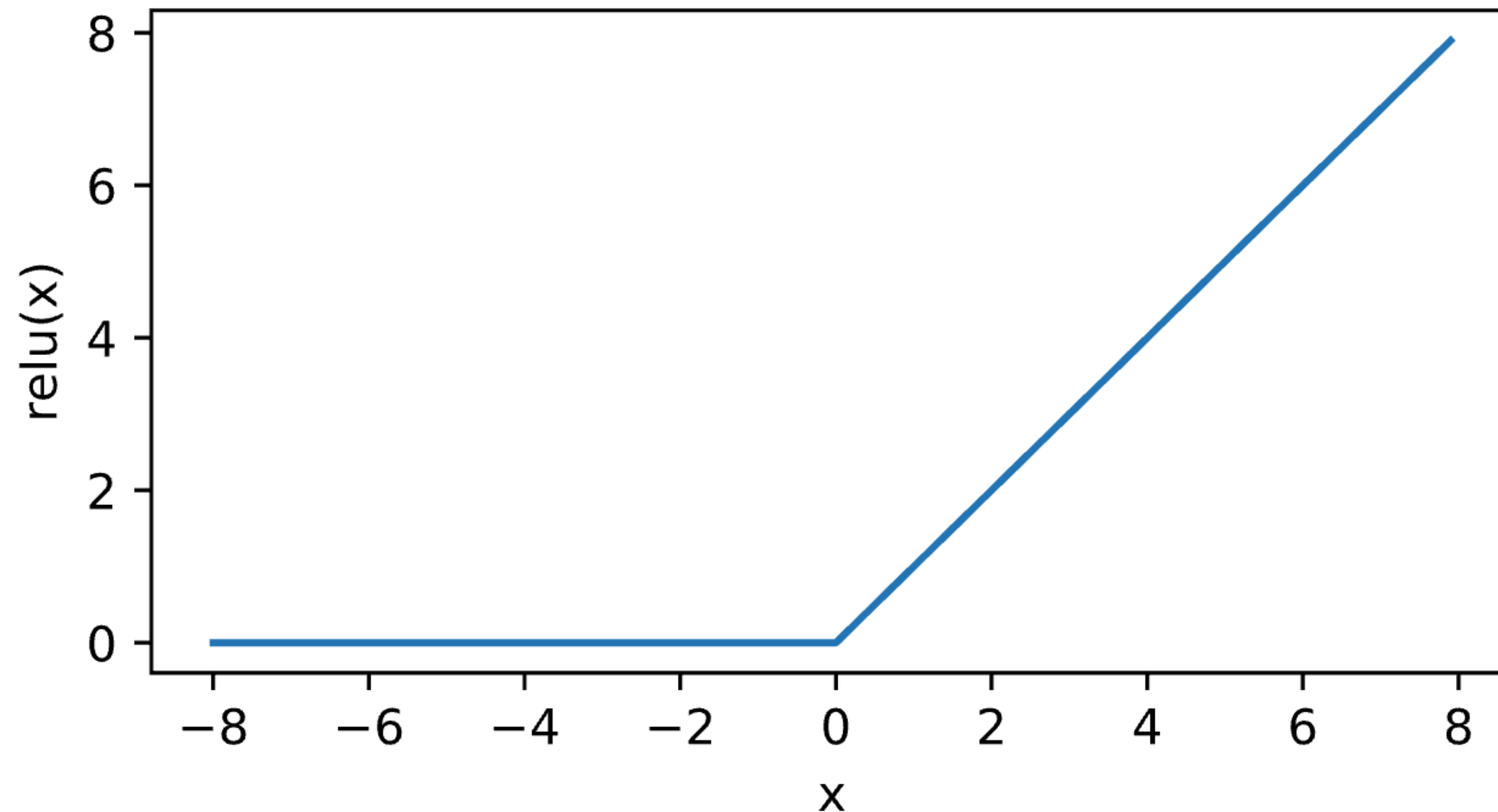
$$\tanh(z) = 2\text{sigmoid}(2z) - 1$$



ReLU Activation

ReLU: rectified linear unit (commonly used in modern neural networks)

$$\text{ReLU}(x) = \max(x, 0)$$



Quiz Break

Which one of the following is a valid activation function?

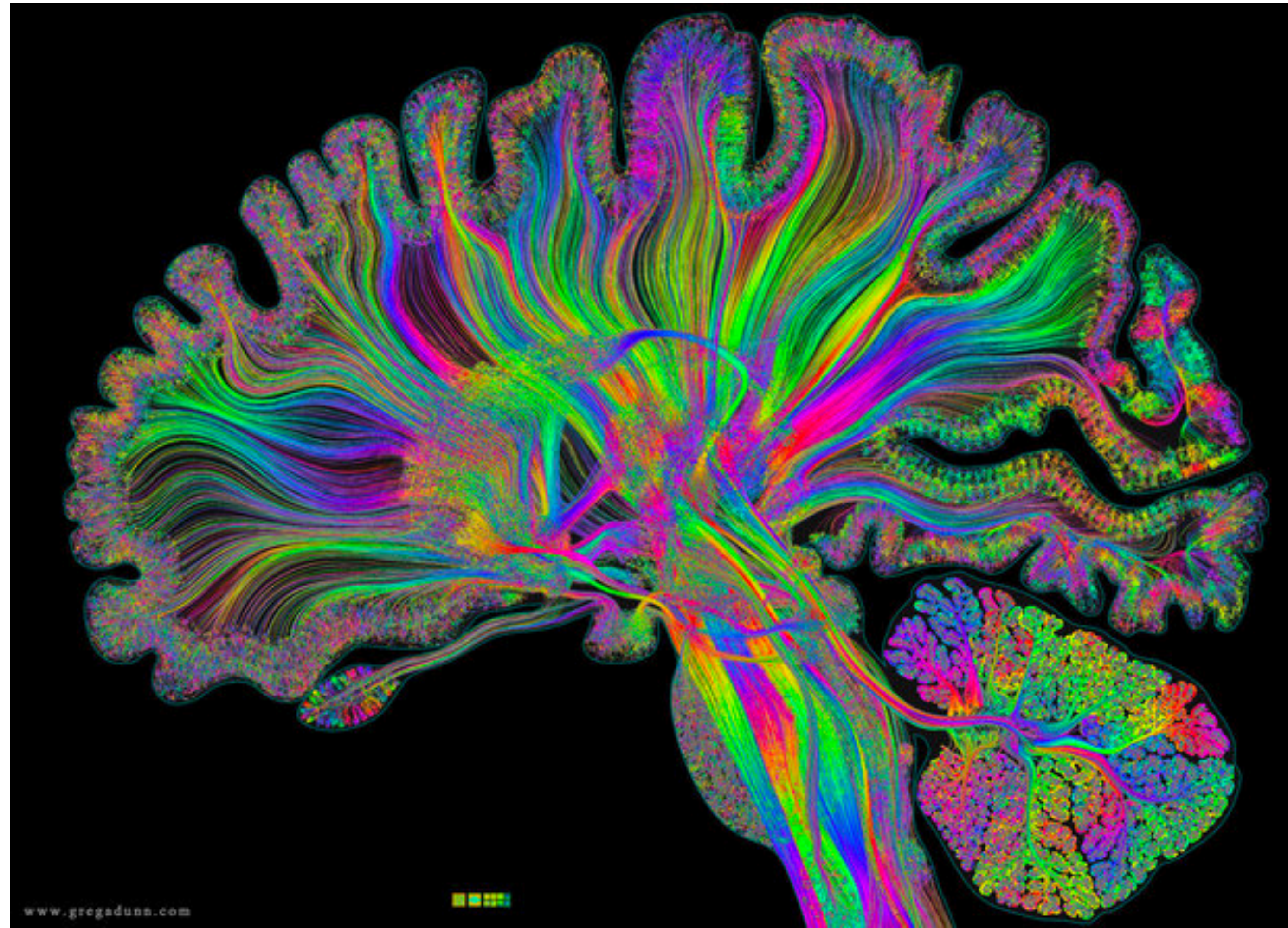
- a) Step function
- b) Sigmoid function
- c) ReLU function
- d) all of above

Quiz Break

Which one of the following is a valid activation function?

- a) Step function
- b) Sigmoid function
- c) ReLU function
- D) all of above**

Multilayer Perceptron



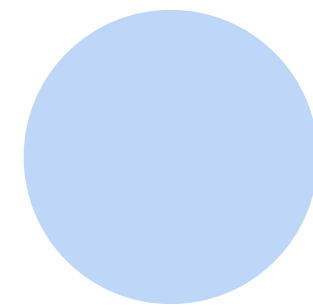
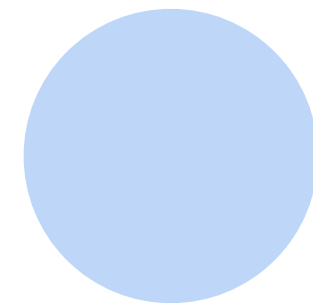
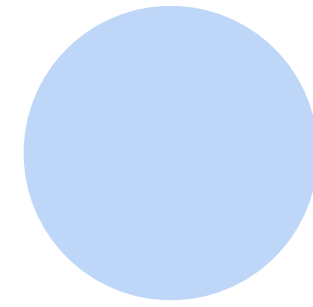
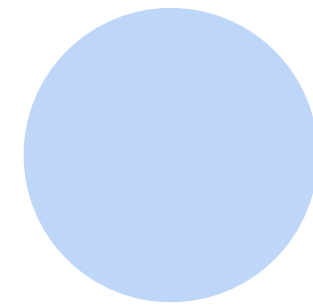
Single Hidden Layer

How to classify

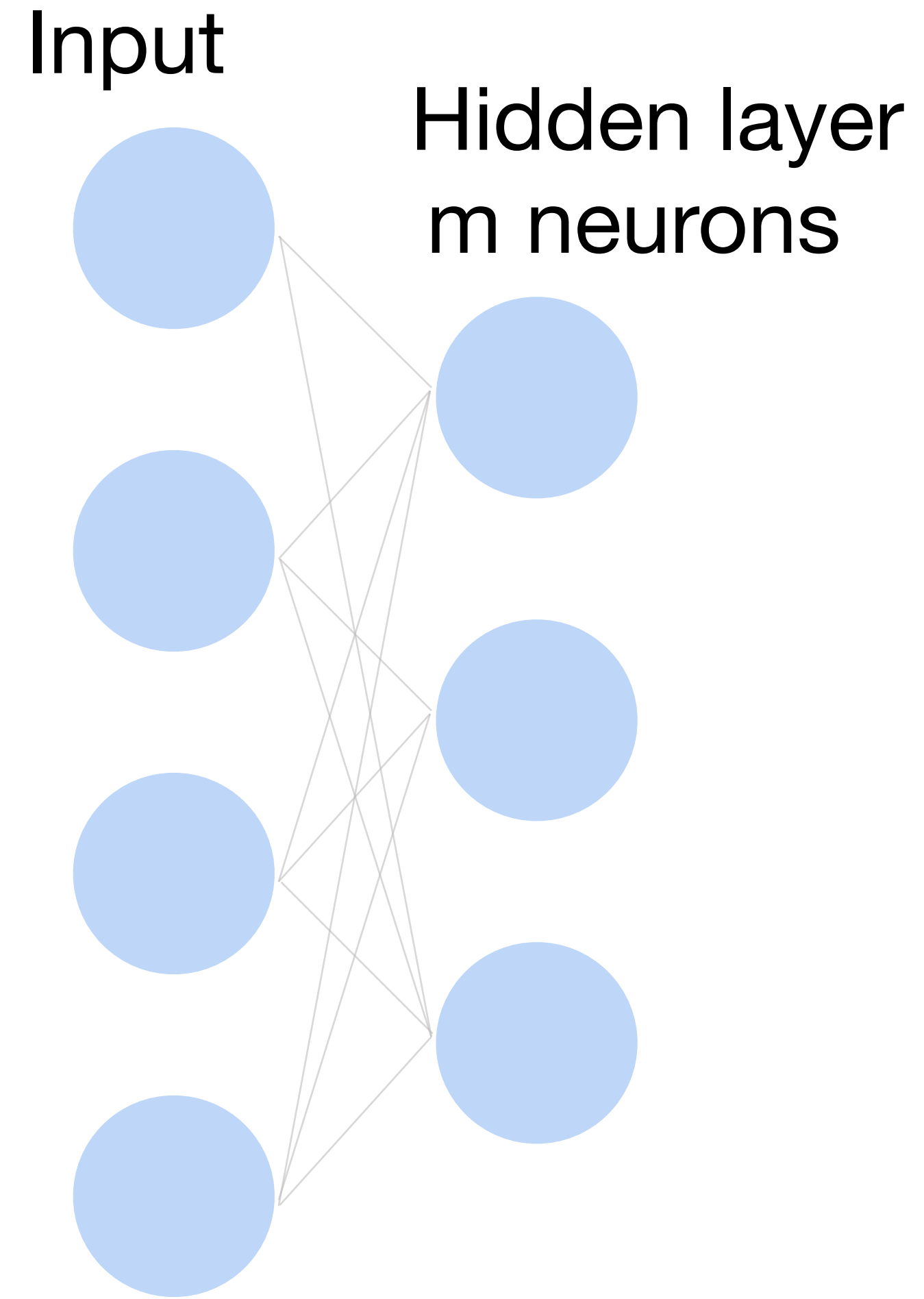
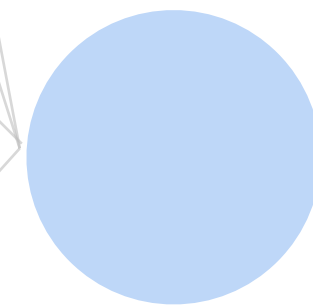
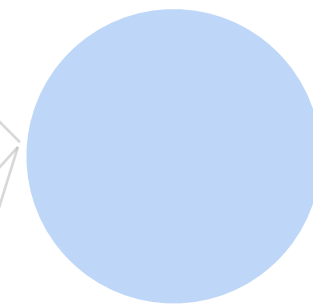
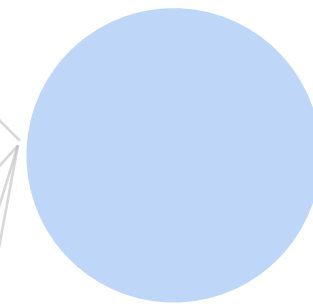
Cats vs. dogs?



Input



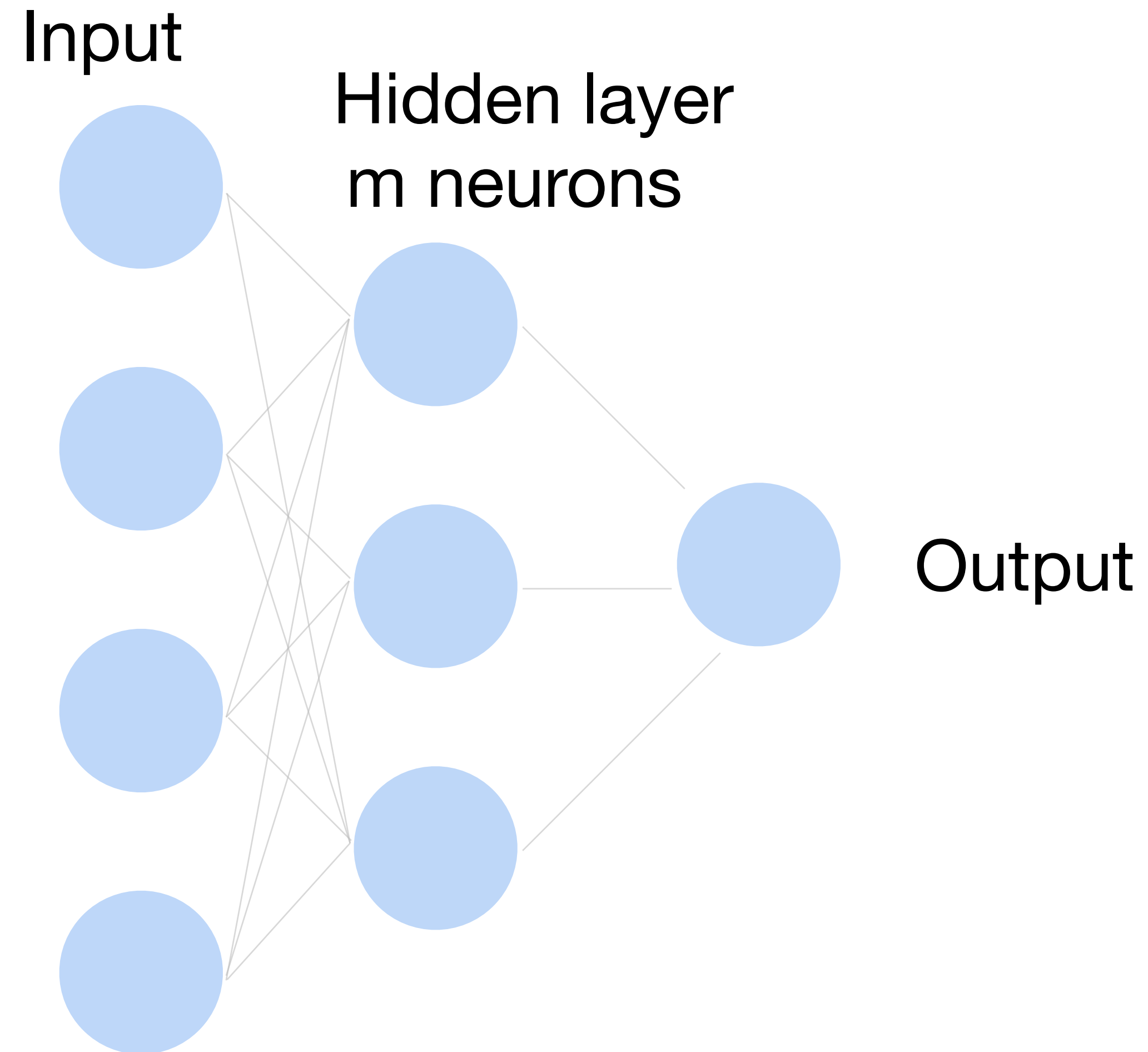
Hidden layer
m neurons



Single Hidden Layer

How to classify

Cats vs. dogs?

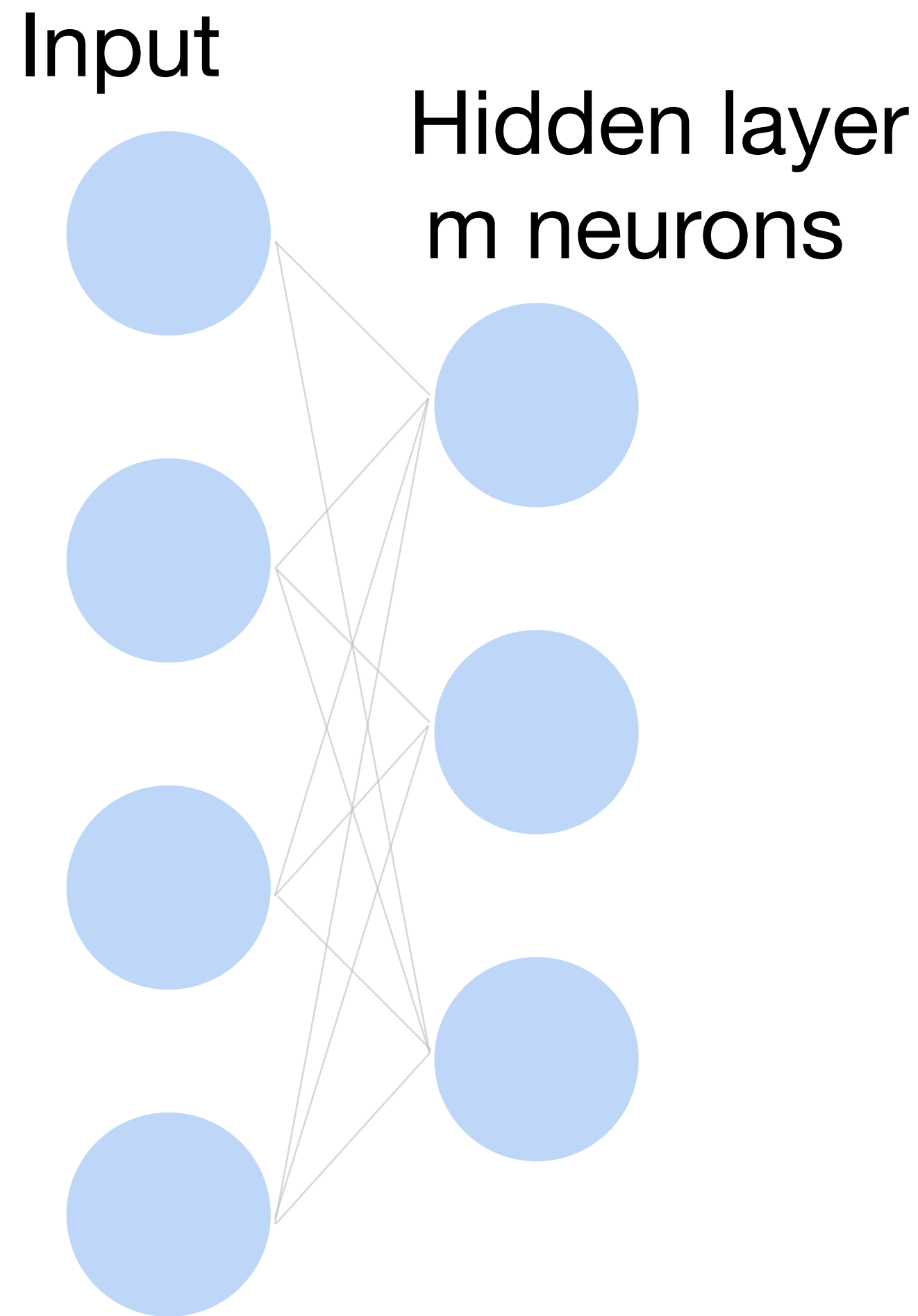


Single Hidden Layer

- Input $\mathbf{x} \in \mathbb{R}^d$
- Hidden $\mathbf{W} \in \mathbb{R}^{m \times d}$, $\mathbf{b} \in \mathbb{R}^m$
- Intermediate output

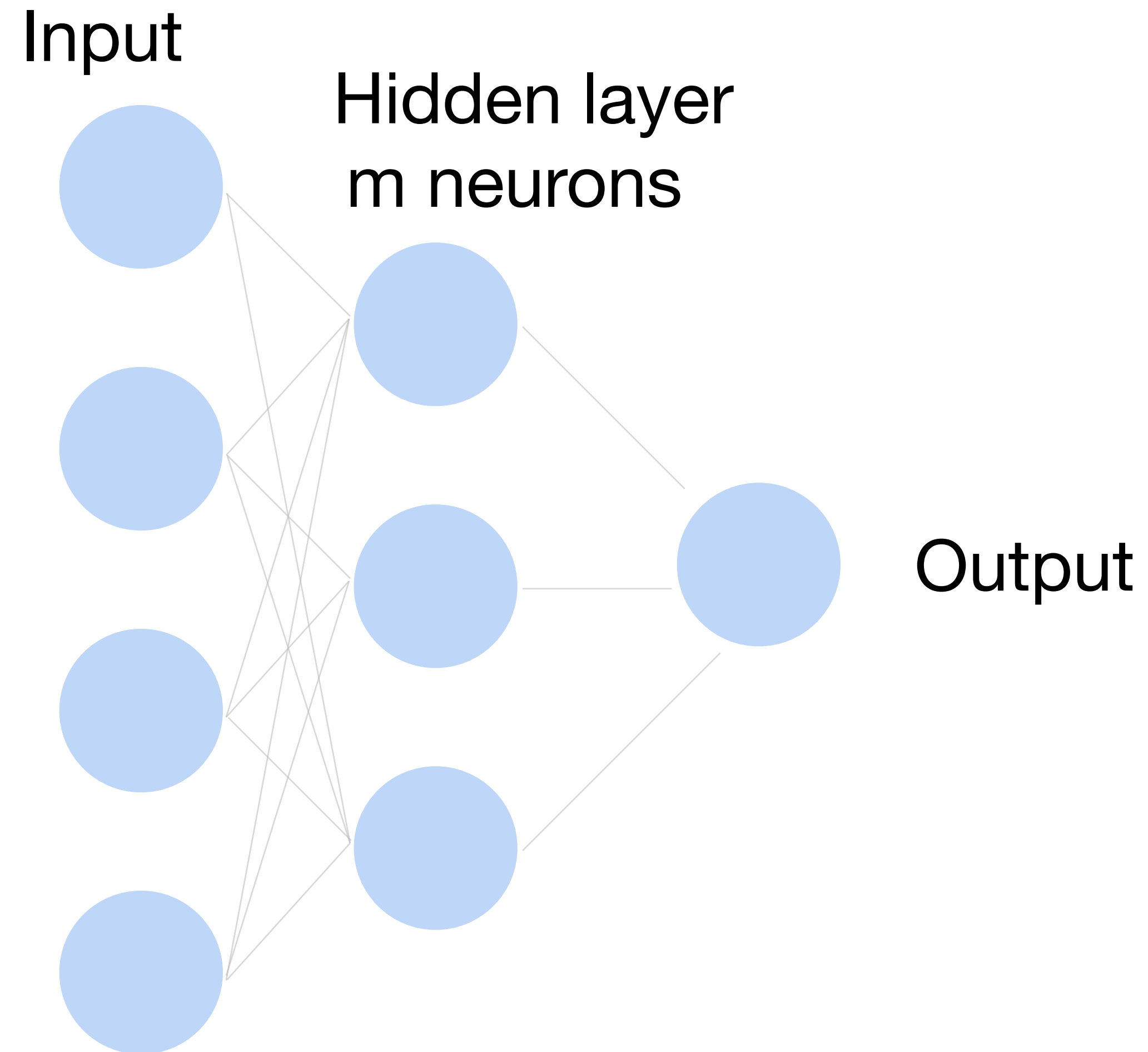
$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

σ is an element-wise
activation function



Single Hidden Layer

- Output $\mathbf{f} = \mathbf{w}_2^T \mathbf{h} + b_2$



Quiz Break

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Which of the following functions is NOT an element-wise operation that can be used as an activation function?

A $f(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

B $f(x) = \begin{bmatrix} \max(0, x_1) \\ \max(0, x_2) \end{bmatrix}$

C $f(x) = \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \end{bmatrix}$

D $f(x) = \begin{bmatrix} \exp(x_1 + x_2) \\ \exp(x_2) \end{bmatrix}$

Quiz Break

Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Which of the following functions is NOT an element-wise operation that can be used as an activation function?

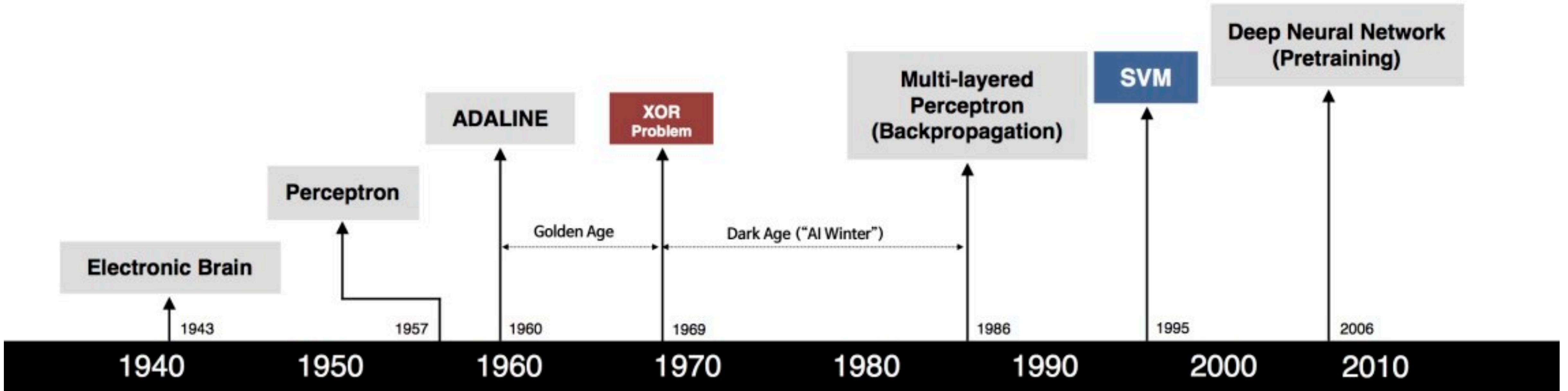
A $f(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

B $f(x) = \begin{bmatrix} \max(0, x_1) \\ \max(0, x_2) \end{bmatrix}$

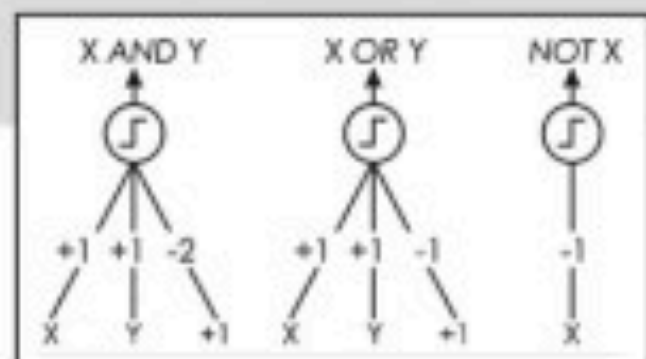
C $f(x) = \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \end{bmatrix}$

D $f(x) = \begin{bmatrix} \exp(x_1 + x_2) \\ \exp(x_2) \end{bmatrix}$

Brief history of neural networks



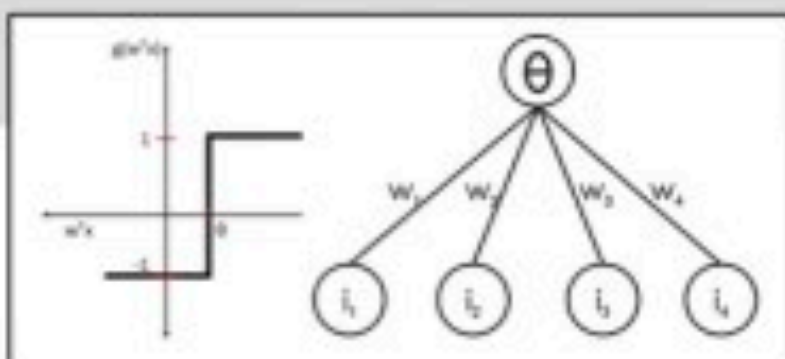
S. McCulloch - W. Pitts



- Adjustable Weights
- Weights are not Learned



F. Rosenblatt



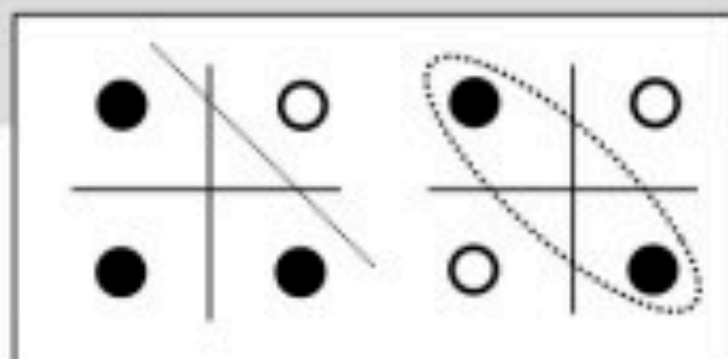
- Learnable Weights and Threshold



B. Widrow - M. Hoff



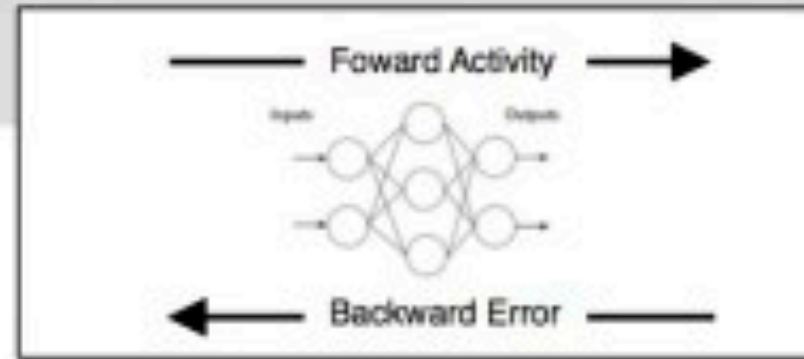
M. Minsky - S. Papert



- XOR Problem



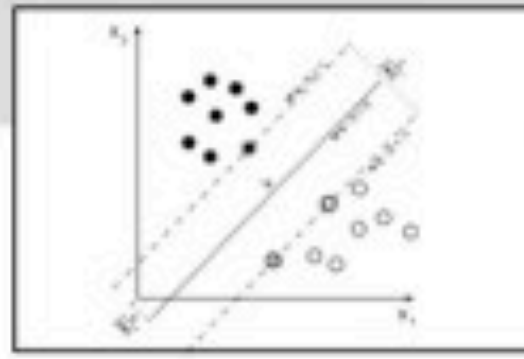
D. Rumelhart - G. Hinton - R. Williams



- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting



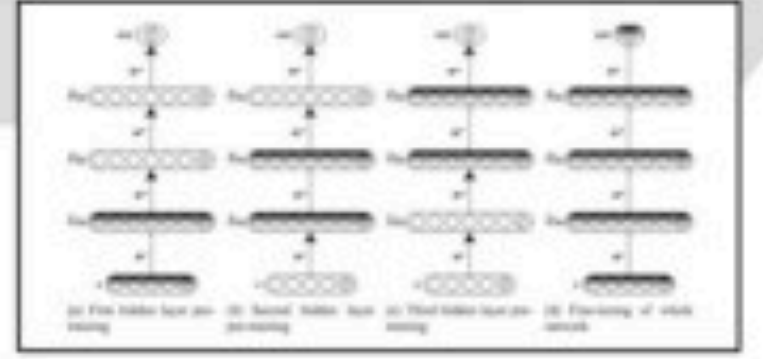
V. Vapnik - C. Cortes



- Limitations of learning prior knowledge
- Kernel function: Human Intervention



G. Hinton - S. Ruslan



- Hierarchical feature Learning

What we've learned today...

- Single-layer Perceptron
 - Motivation
 - Activation function
 - Representing AND, OR, NOT
- Brief history of neural networks