CS 540 Introduction to Artificial Intelligence

Perceptron

Yingyu Liang
University of Wisconsin-Madison

Oct 19, 2021

Slides created by Sharon Li [modified by Yingyu Liang]

Today's outline

- Naive Bayes (cont.)
- Single-layer Neural Network (Perceptron)

Part I: Naïve Bayes (cont.)

Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

Posterior probability p(Yes | ☀) vs. p(No | ☀)

Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

Posterior probability p(Yes | ) vs. p(No | )

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

Posterior probability p(Yes | ☀) vs. p(No | ☀)

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

$$p(\text{Play} \mid ☀) = \frac{p(☀ \mid \text{Play})\, p(\text{Play})}{p(☀)}$$

Bayes rule

Example 1: Play outside or not?

- **Step 1**: Convert the data to a frequency table of Weather and Play

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Example 1: Play outside or not?

Step 1: Convert the data to a frequency table of Weather and Play

Step 2: Based on the frequency table, calculate **likelihoods** and **priors**

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

$$p(\text{Play} = \text{Yes}) = 0.64$$

$$p(☀ \mid \text{Yes}) = 3/9 = 0.33$$

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Example 1: Play outside or not?

Step 3: Based on the likelihoods and priors, calculate posteriors

P(Yes| ☀)
=P(☀ |Yes)*P(Yes)/P(☀)

?

P(No| ☀)
=P(☀ |No)*P(No)/P(☀)

?

Example 1: Play outside or not?

Step 3: Based on the likelihoods and priors, calculate posteriors

P(Yes| ☀)
=P(☀ |Yes)*P(Yes)/P(☀)
=0.33*0.64/0.36
=0.6

P(No| ☀)
=P(☀ |No)*P(No)/P(☀)
=0.4*0.36/0.36
=0.4

P(Yes| ☀) > P(No| ☀) go outside and play!

Bayesian classification

$$\hat{y} = \arg\max_y p(y \mid \mathbf{x})$$

(Prediction)

(Posterior)

$$= \arg\max_y \frac{p(\mathbf{x} \mid y) \cdot p(y)}{p(\mathbf{x})}$$

(by Bayes' rule)

$$= \arg\max_y \; p(\mathbf{x} \mid y) p(y)$$

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\boxed{\hat{y}} = \arg\max_y \boxed{p(y \mid X_1, \ldots, X_k)} \quad \text{(Posterior)}$$

(Prediction)

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes' rule)}$$

Independent of y

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes' rule)}$$

Independent of y

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k)$$ (Posterior)

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)}$$ (by Bayes' rule)

$$= \arg\max_y \boxed{p(X_1, \ldots, X_k \mid y)\ p(y)}$$

Bayesian classification

What if $\mathbf{x}$ has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k)$$ (Posterior)

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)}$$ (by Bayes' rule)

$$= \arg\max_y \boxed{\underbrace{p(X_1, \ldots, X_k \mid y)}_{\text{Class conditional likelihood}} \; p(y)}$$

Class prior

Naïve Bayes Assumption

Conditional independence of feature attributes

$$p(X_1, \ldots, X_k \mid y)p(y) = \Pi_{i=1}^{k} \boxed{p(X_i \mid y)} p(y)$$

Easier to estimate

(using MLE!)

Part I: Single-layer Neural Networks

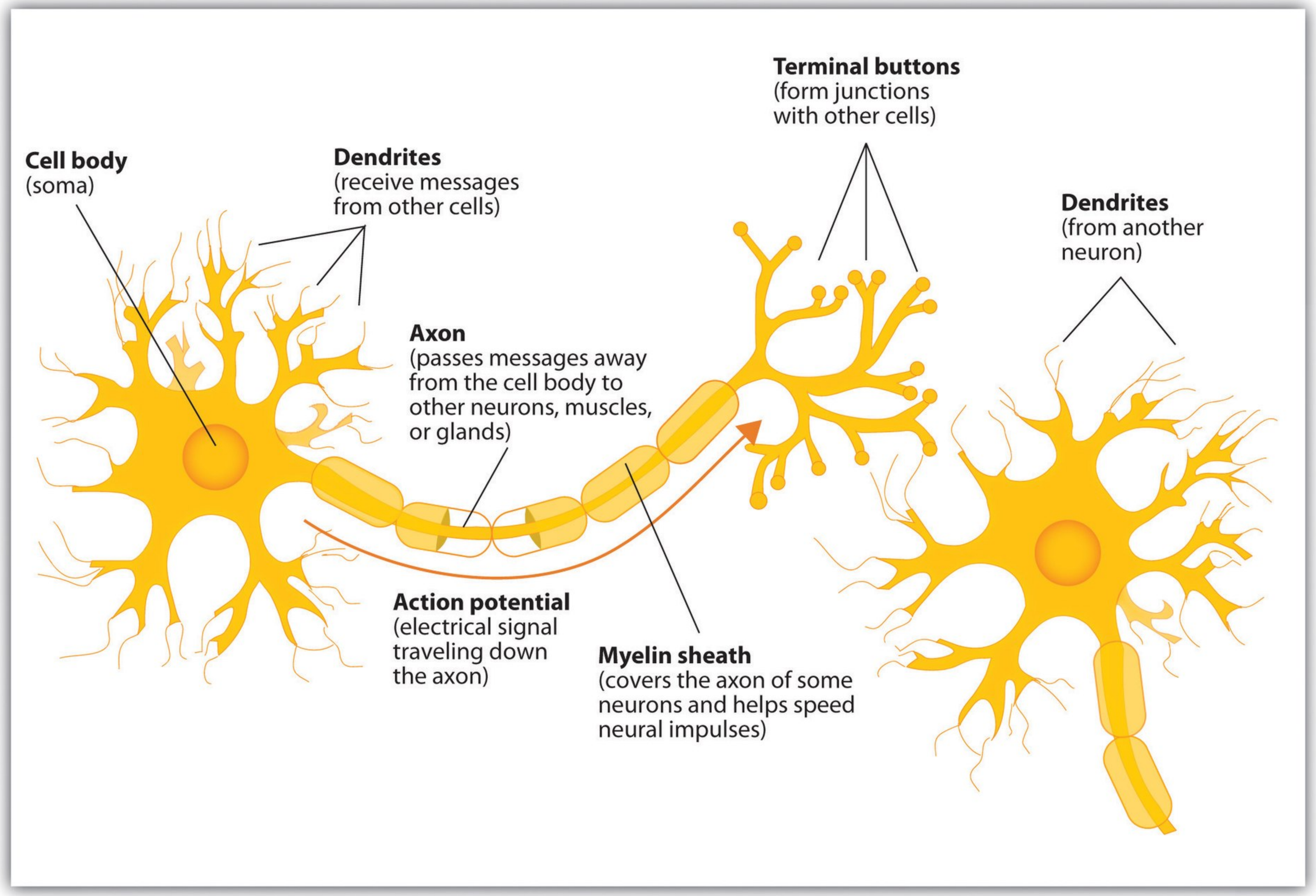
How to classify

Cats vs. dogs?

Inspiration from neuroscience

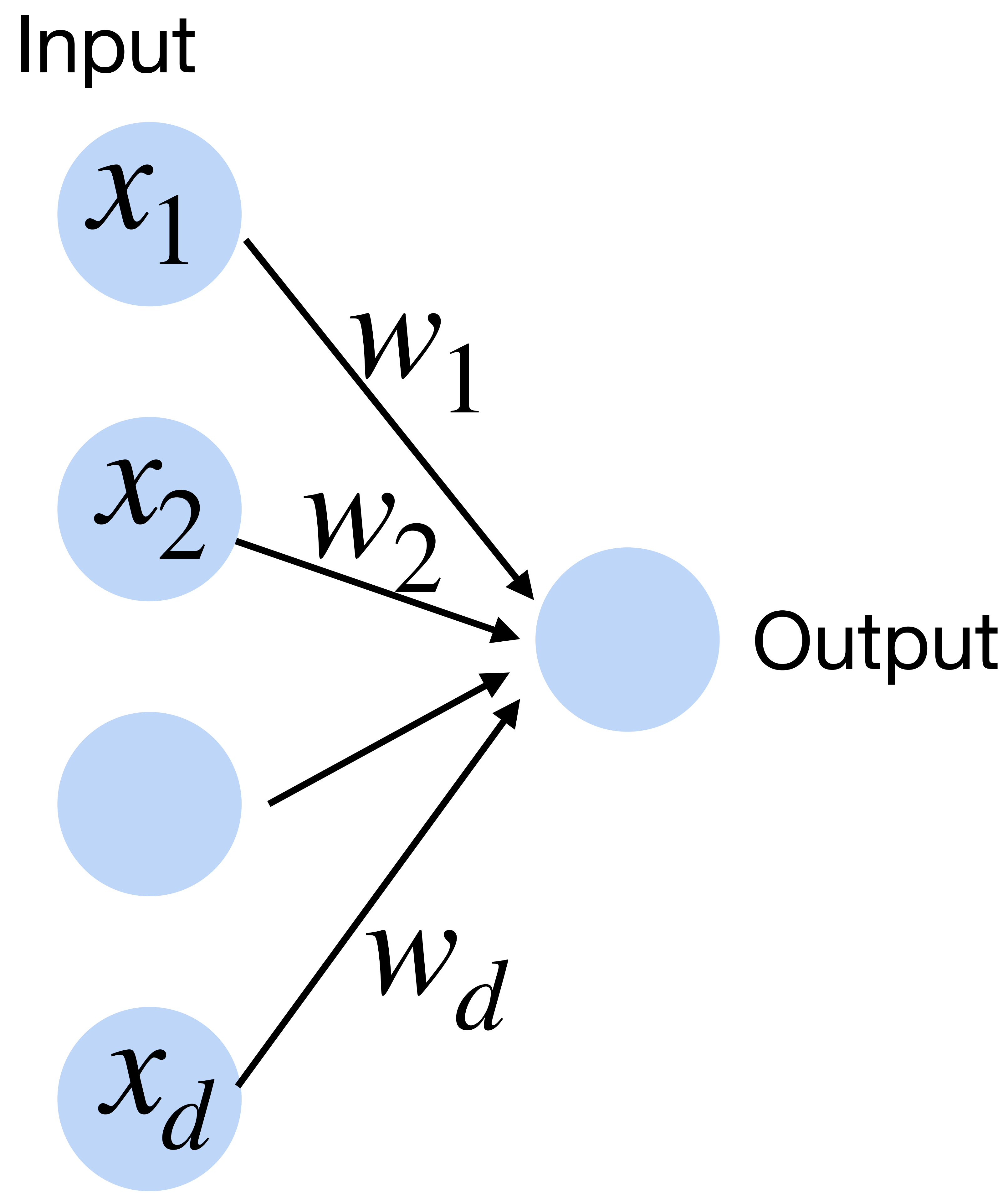
- Inspirations from human brains
- Networks of simple and homogenous units

(wikipedia)



Perceptron

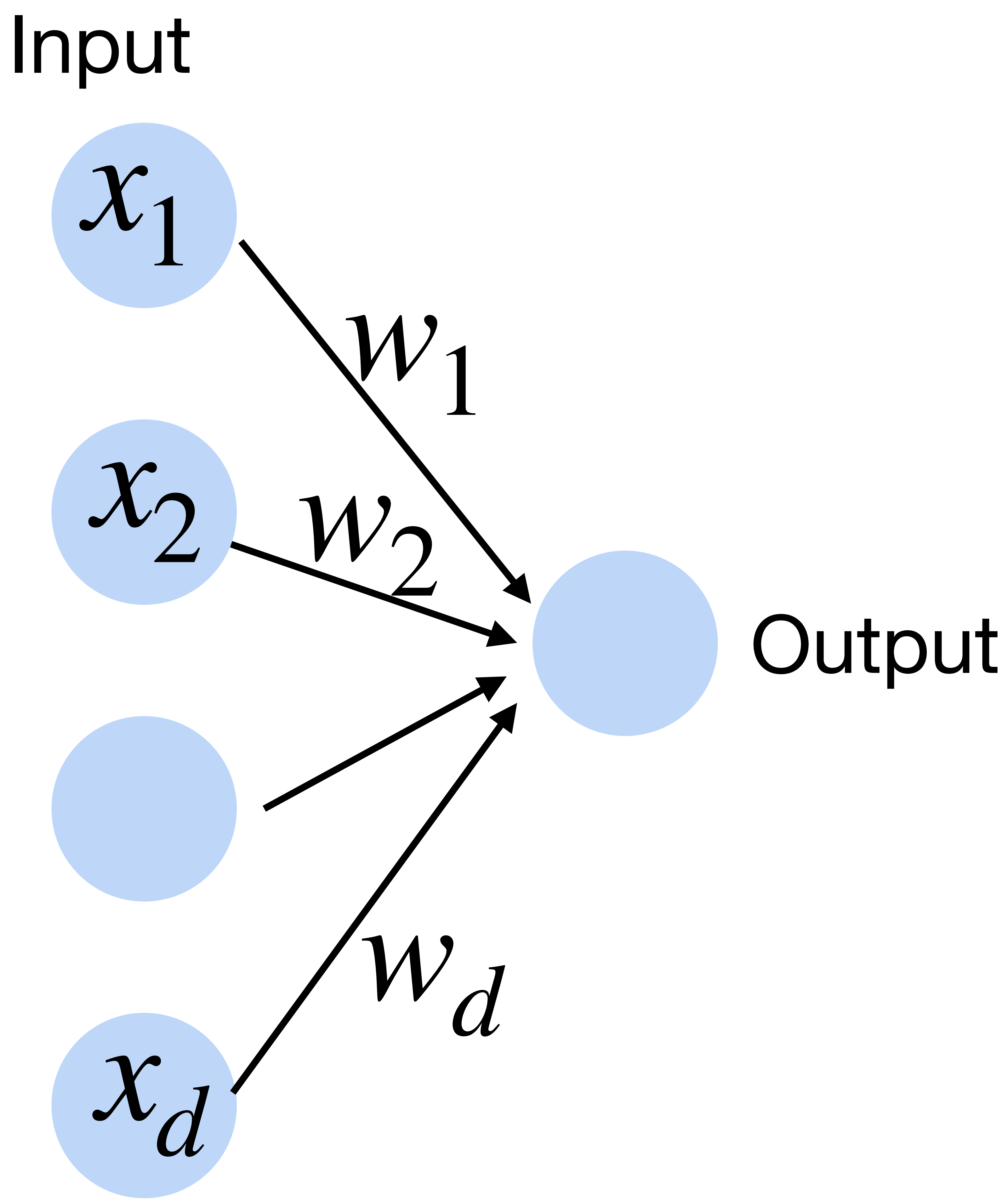
Cats vs. dogs?



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?

Perceptron

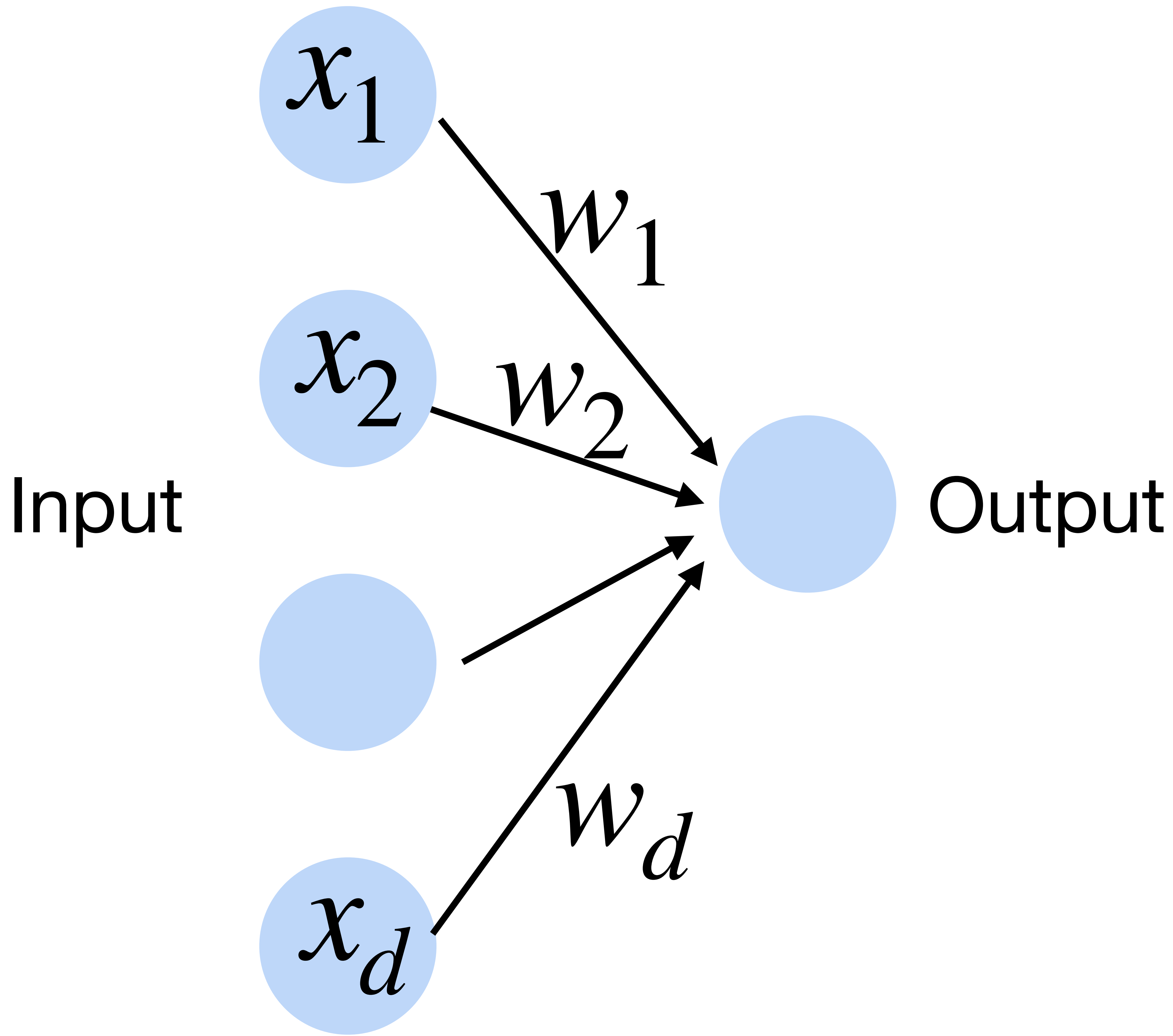
- Given input $\mathbf{x}$, weight $\mathbf{w}$ and bias b, perceptron outputs:

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

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

Activation function

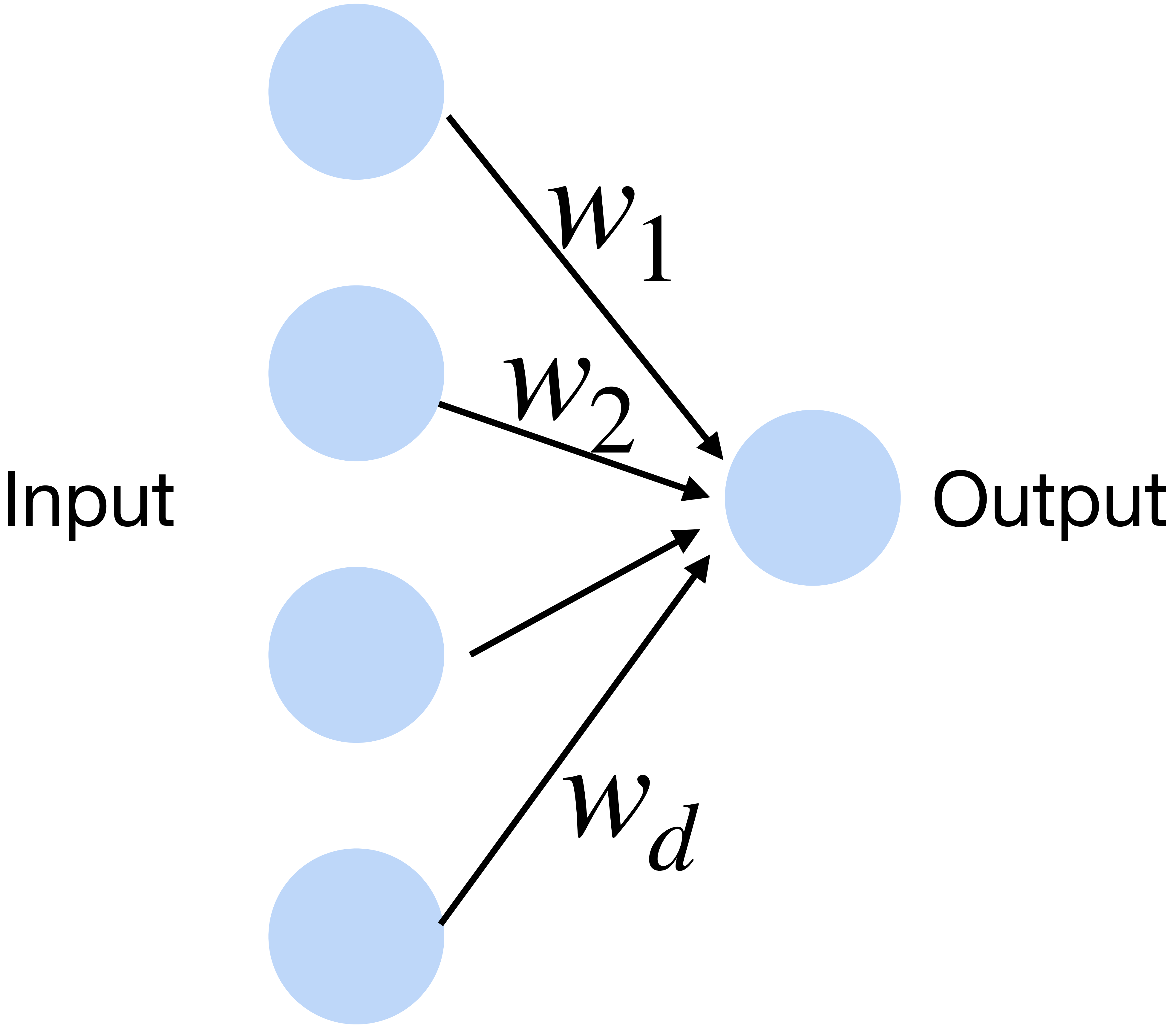
Cats vs. dogs?



Perceptron

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

Cats vs. dogs?



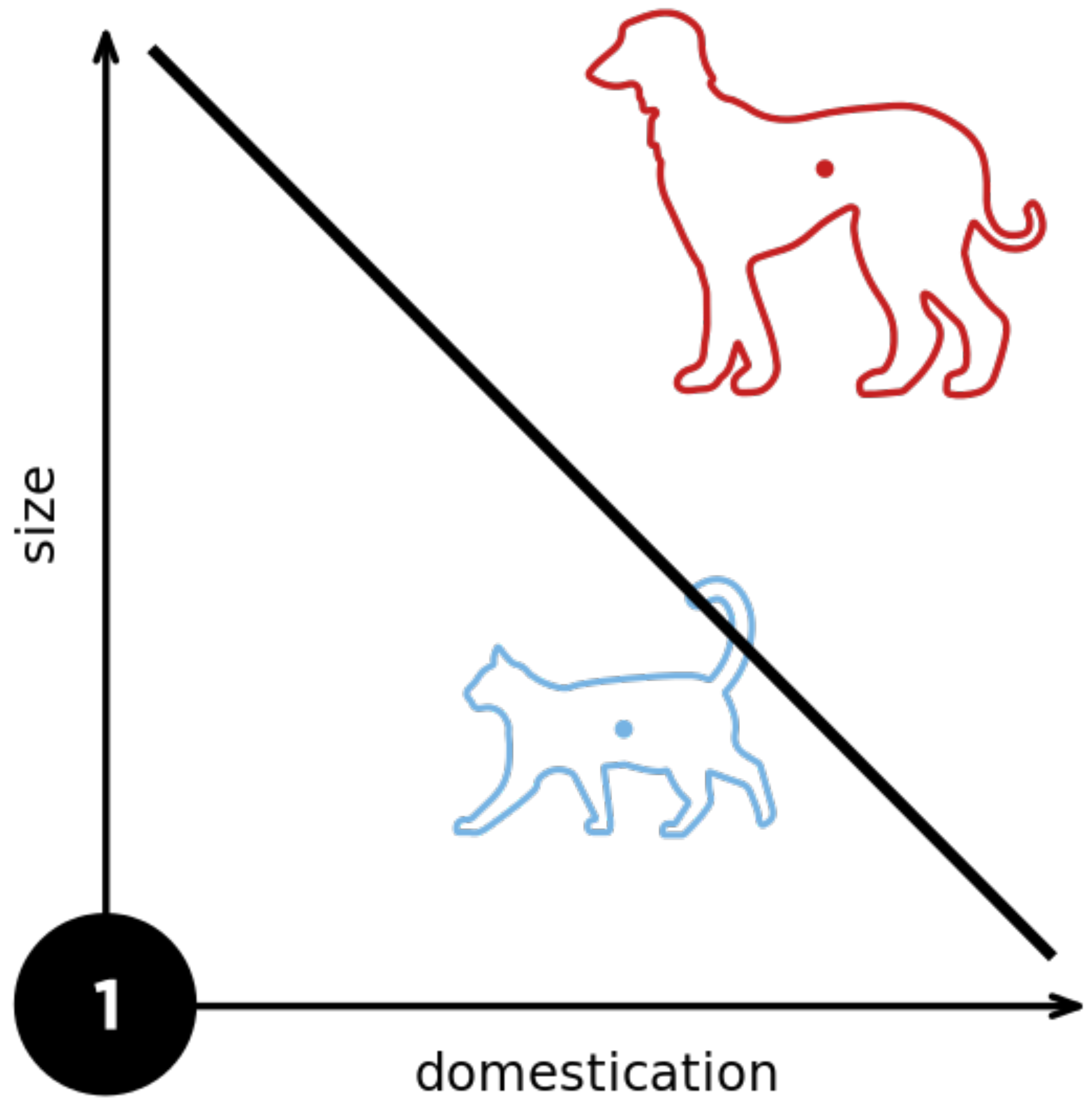
Training the Perceptron

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!

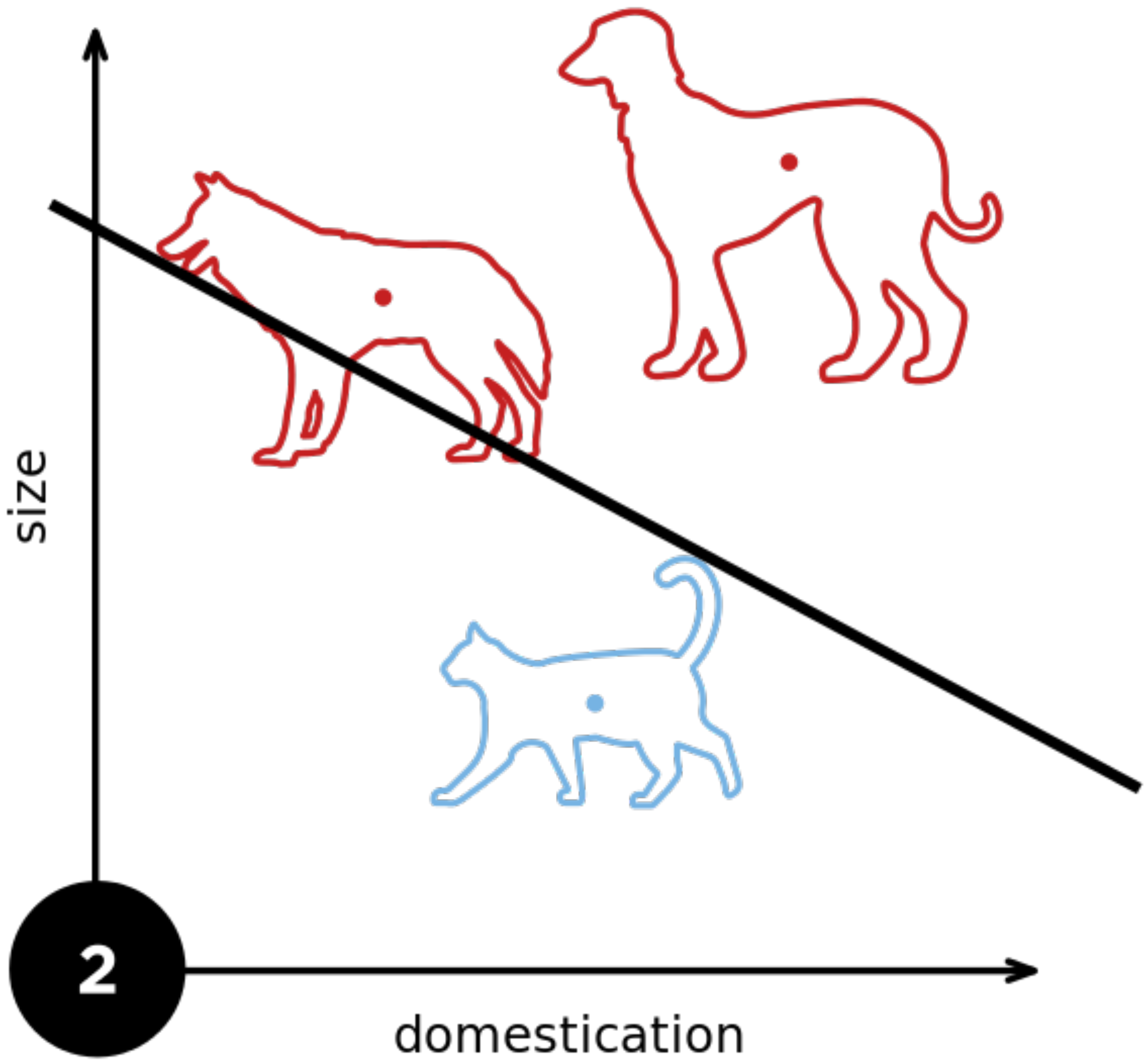
For simplicity, the weight vector and input vector are extended vectors (including the bias or the constant 1).

Perceptron



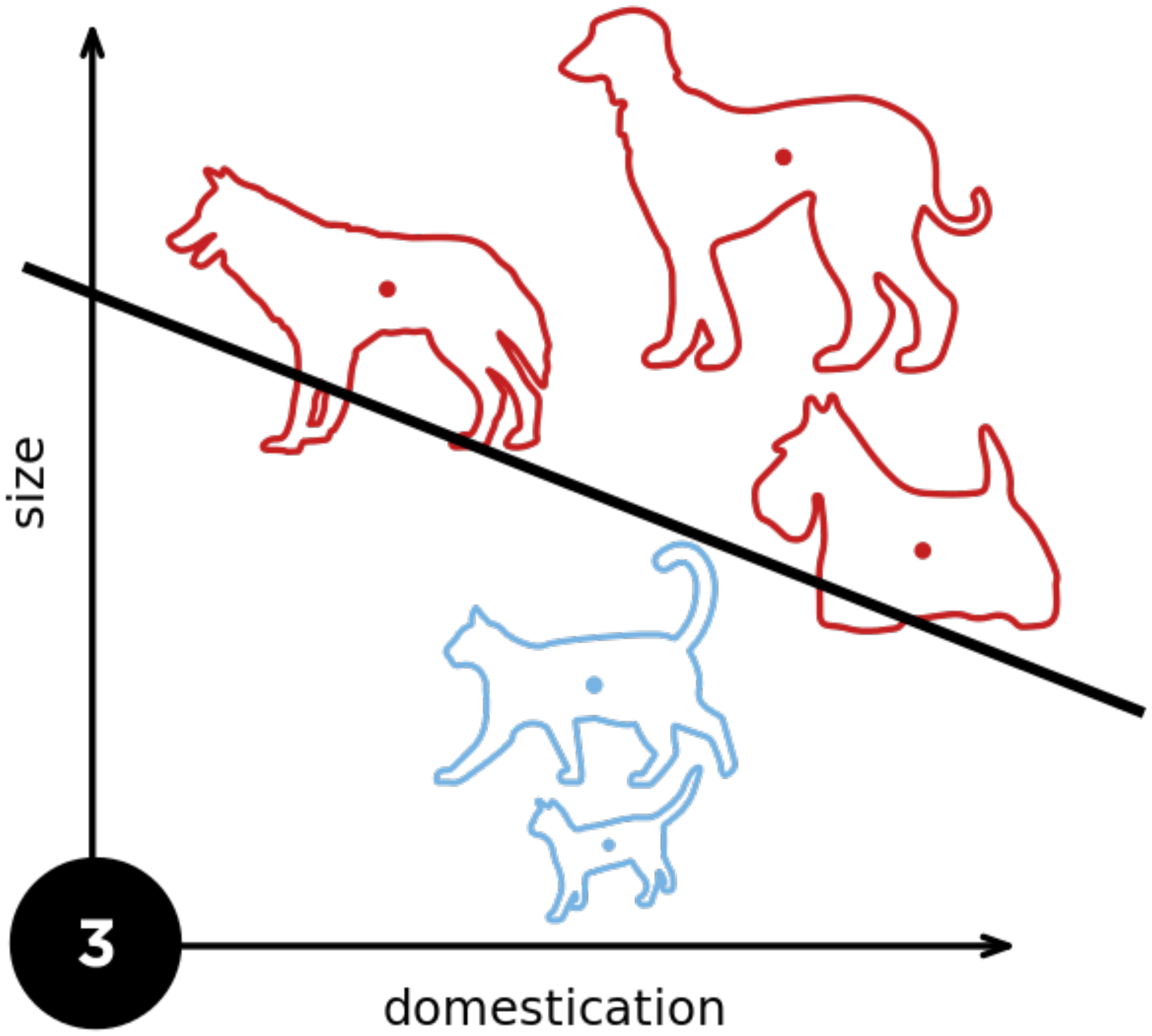
From wikipedia

Perceptron



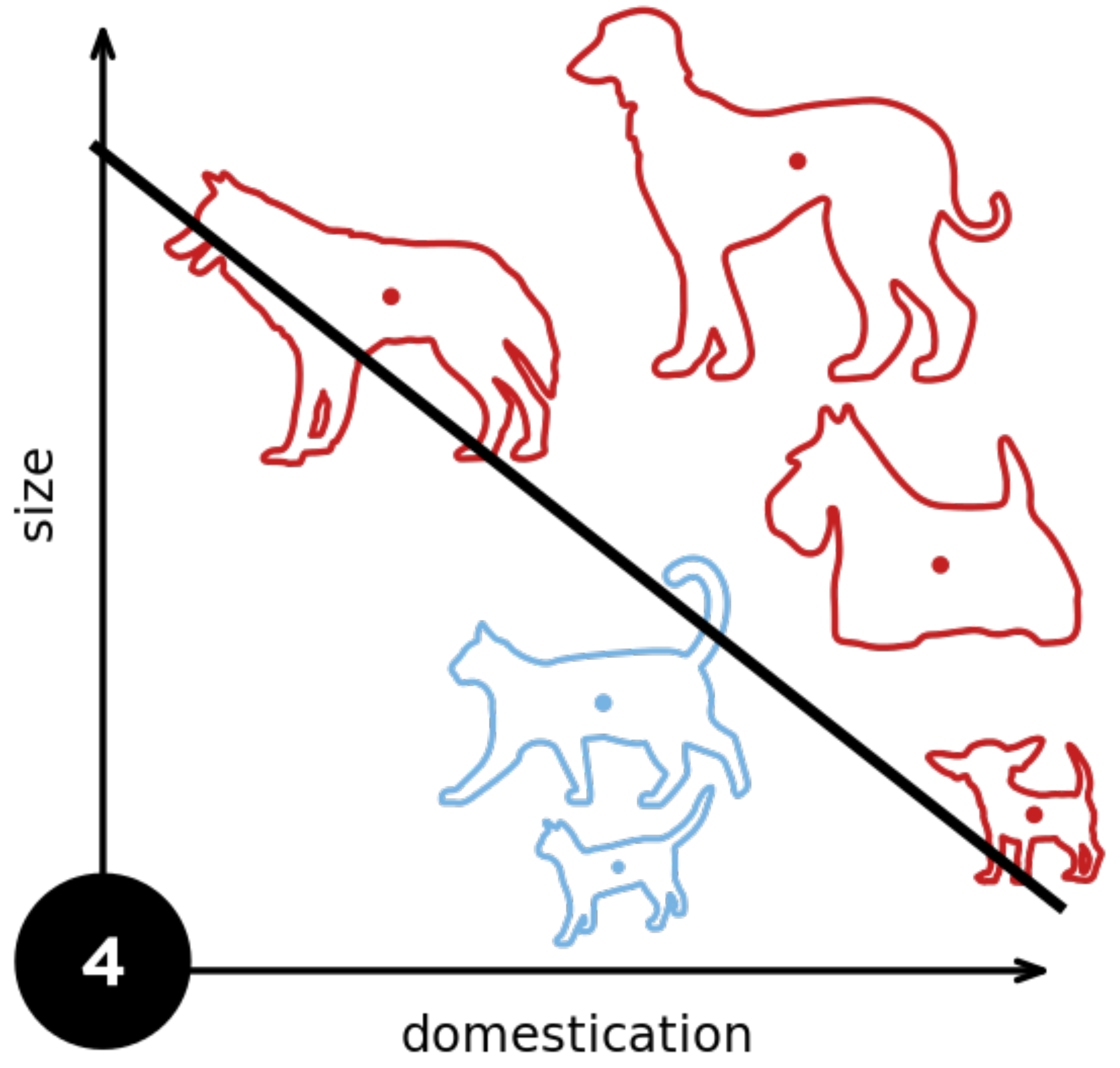
From wikipedia

Perceptron



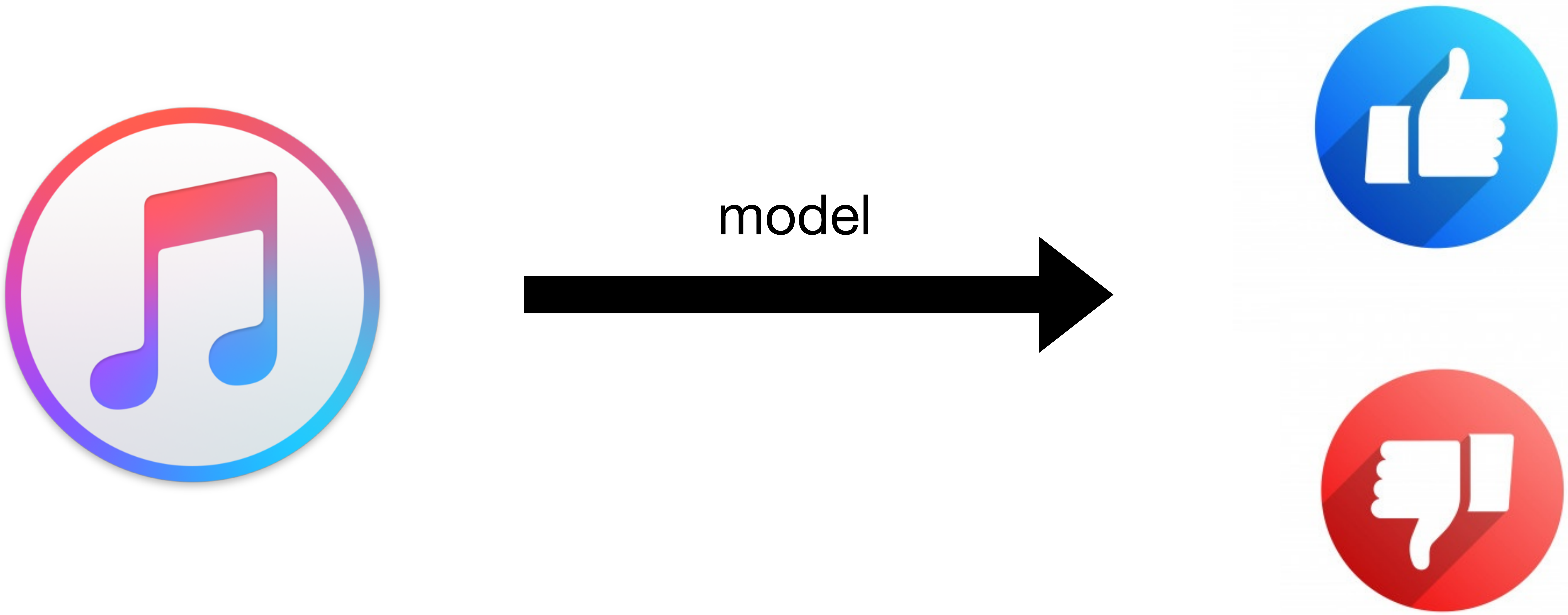
From wikipedia

Perceptron

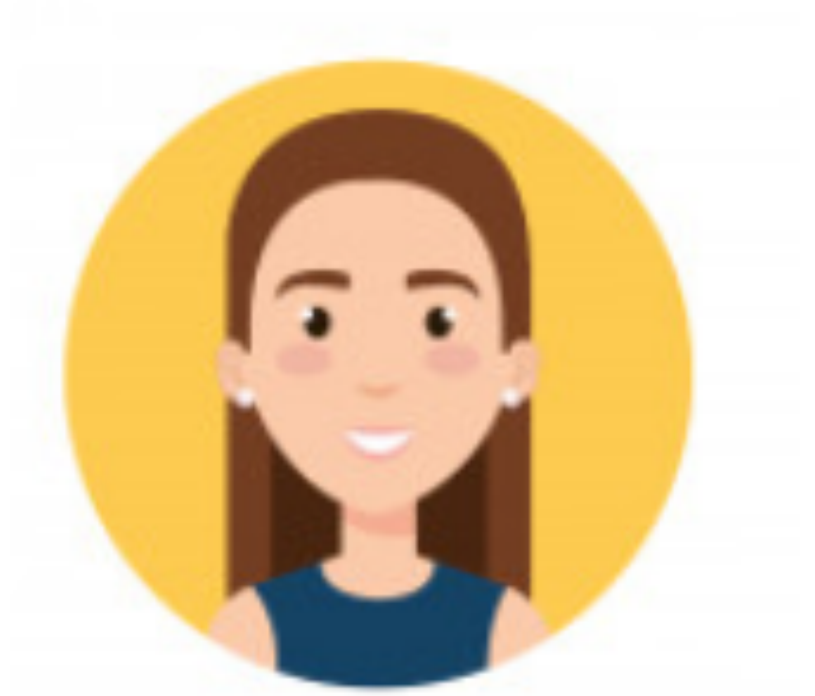


From wikipedia

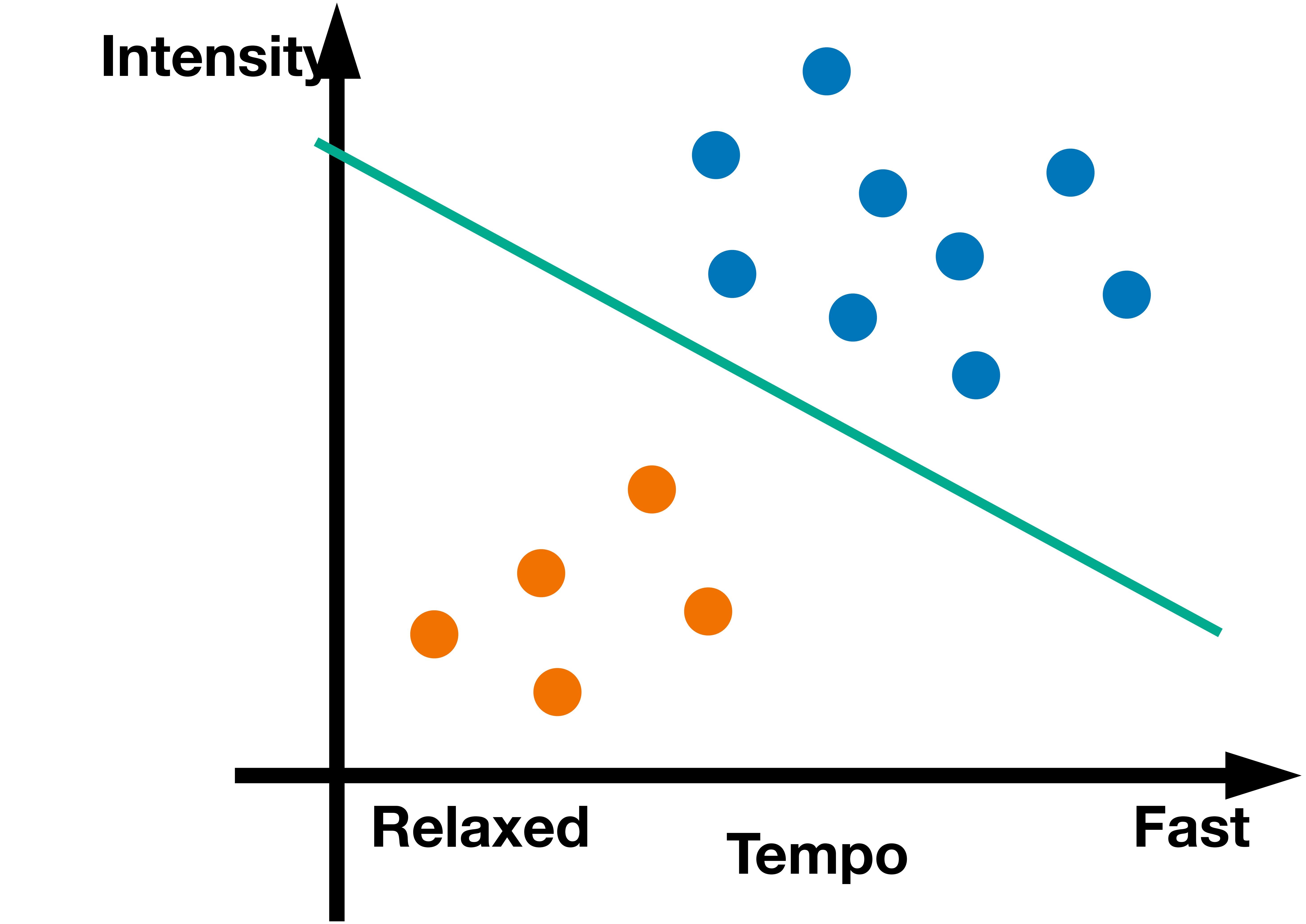
Example 2: Predict whether a user likes a song or not



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



User Sharon



Learning AND function using perceptron

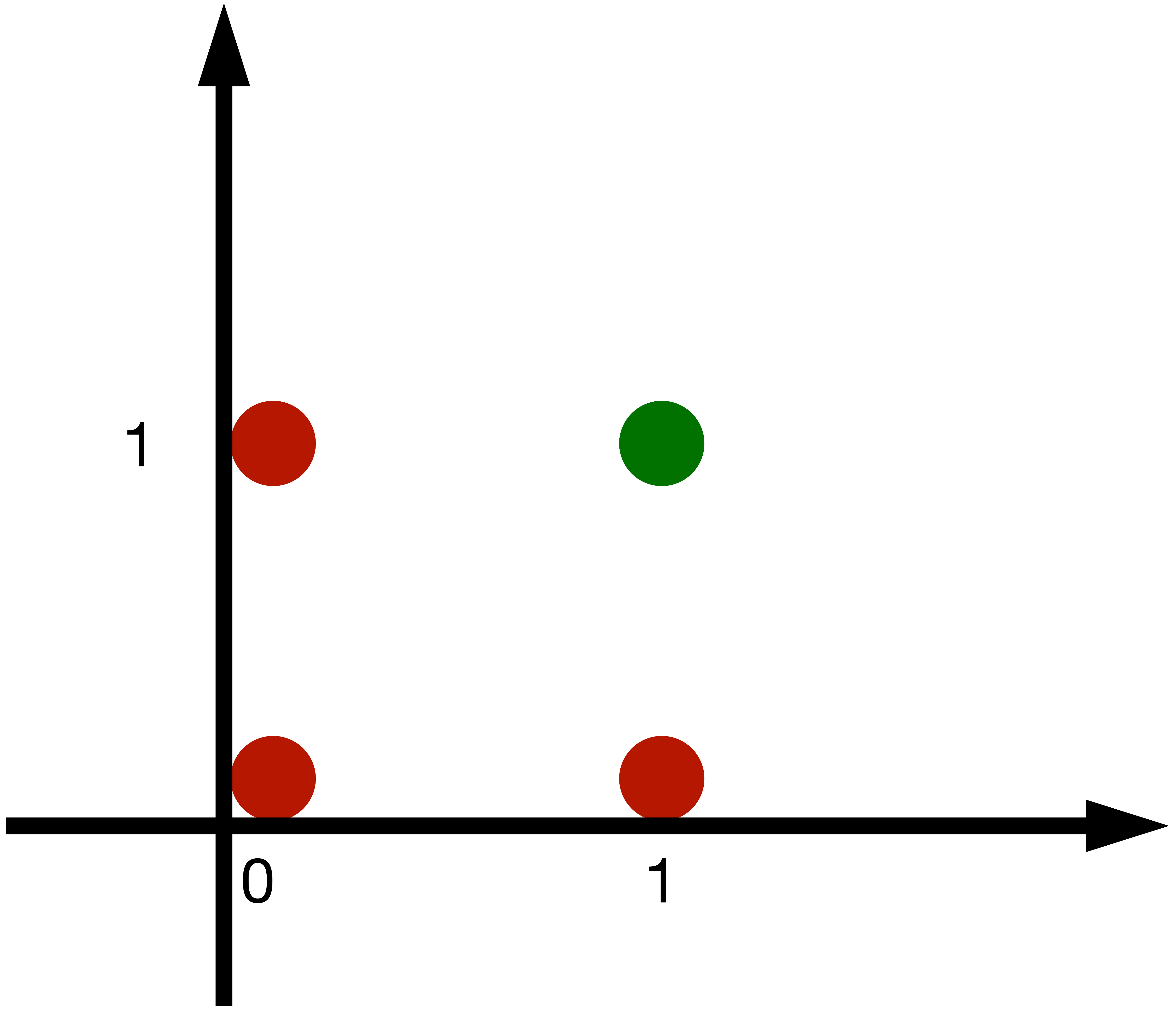
The perceptron can learn an AND function

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

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

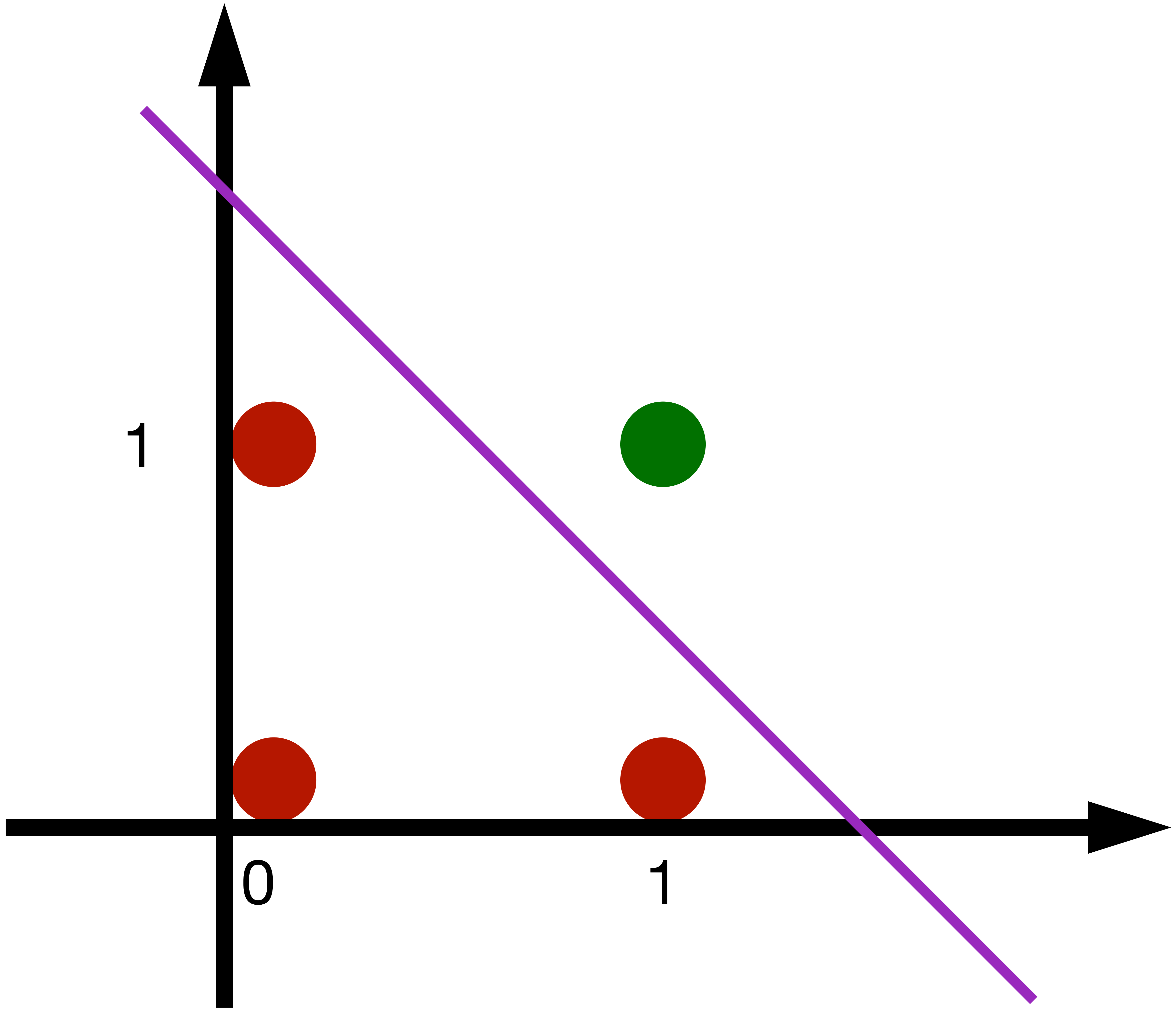
$$x_1 = 0, x_2 = 1, y = 0$$

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



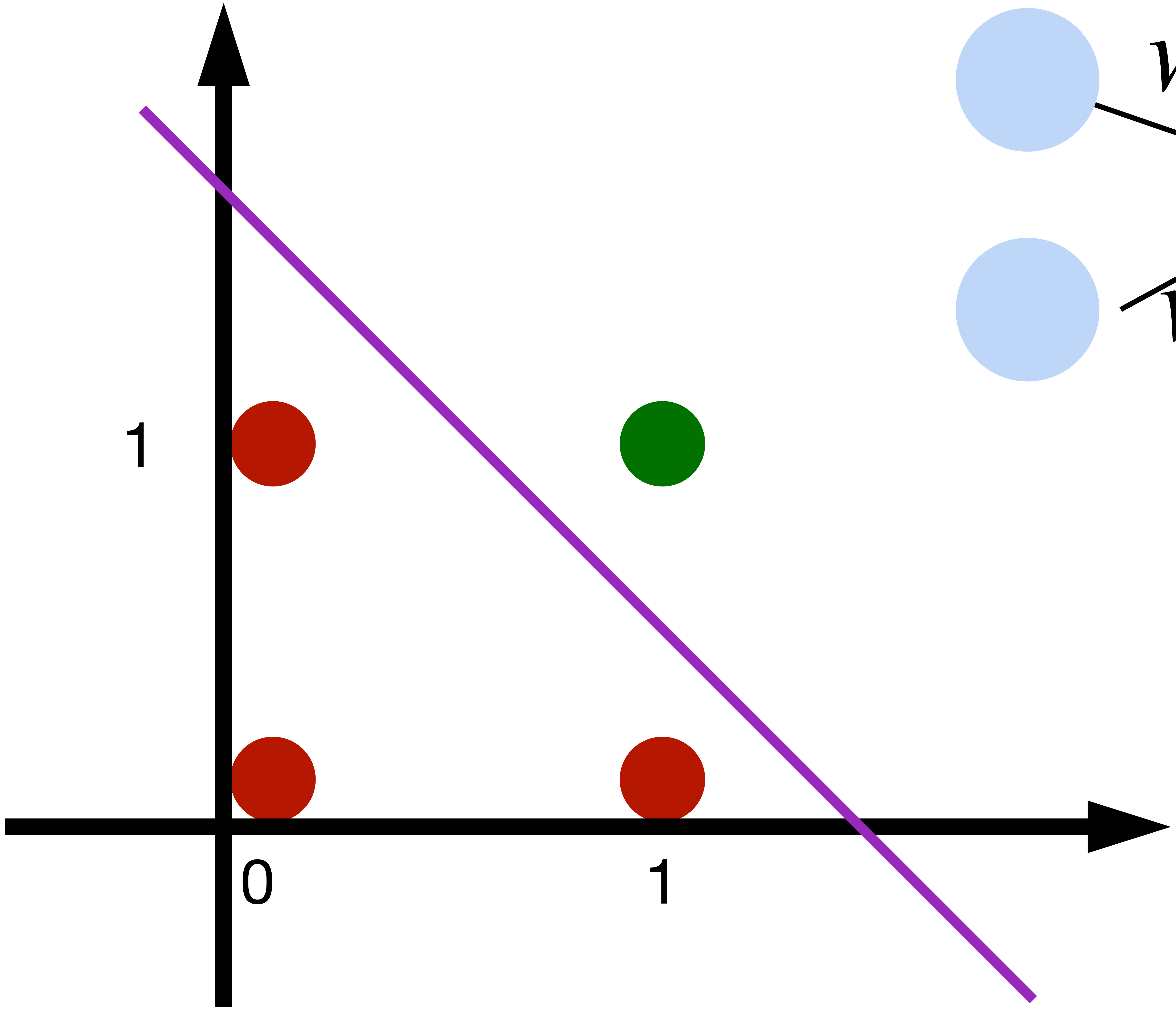
Learning AND function using perceptron

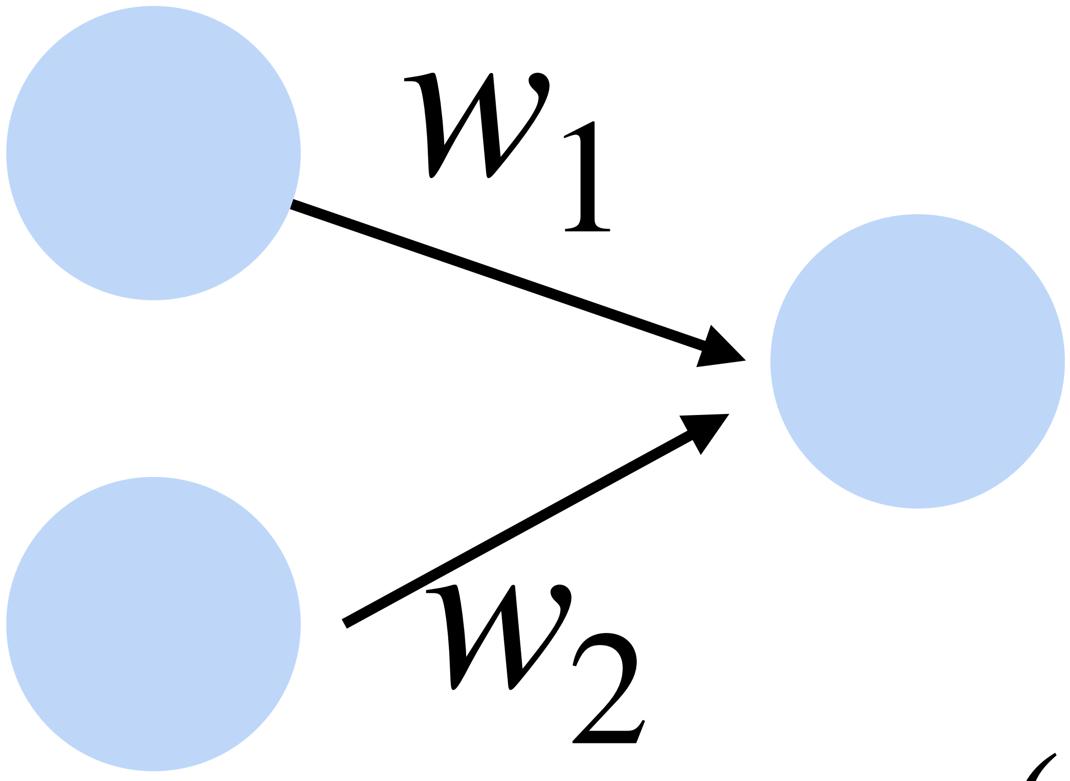
The perceptron can learn an AND function



Learning AND function using perceptron

The perceptron can learn an AND function



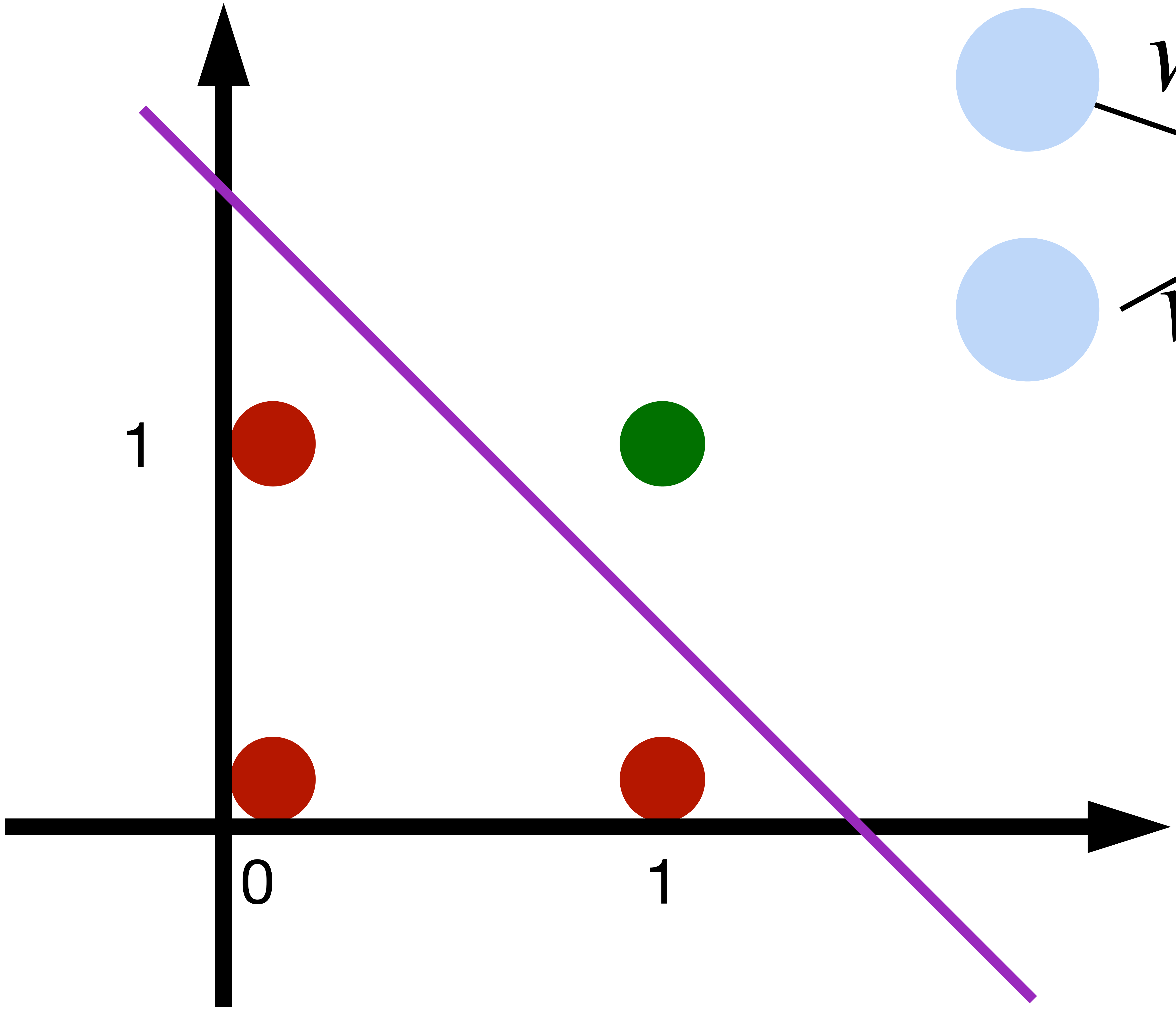


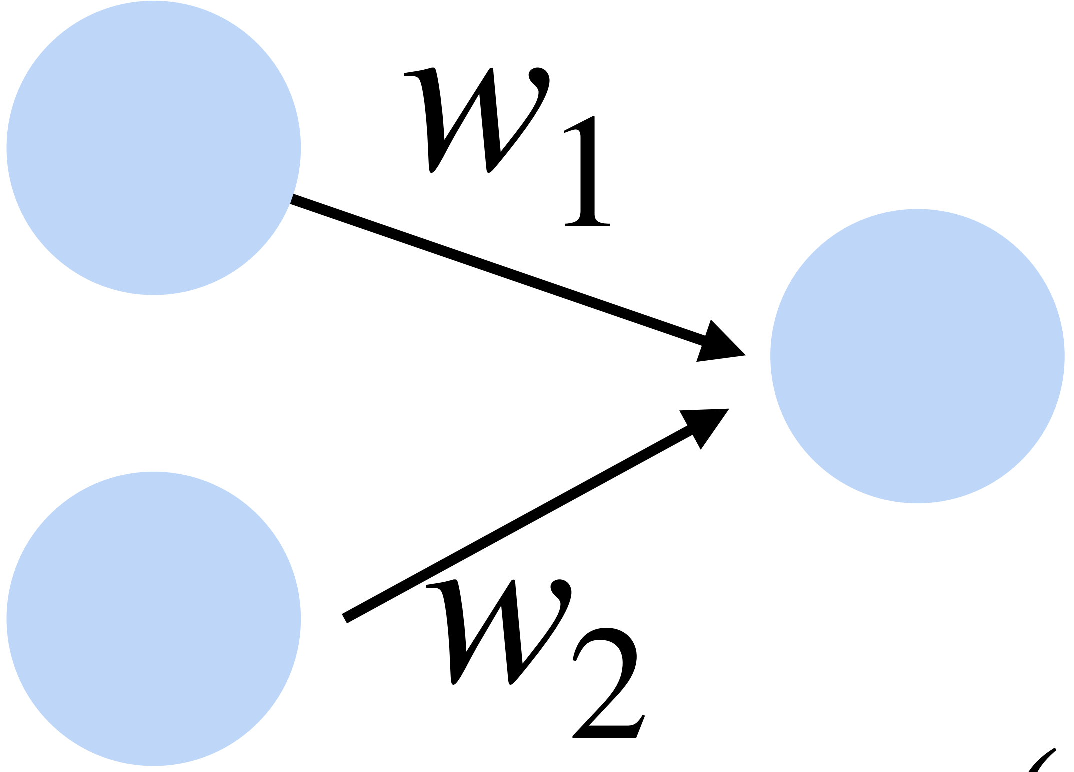
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

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

Learning AND function using perceptron

The perceptron can learn an AND function





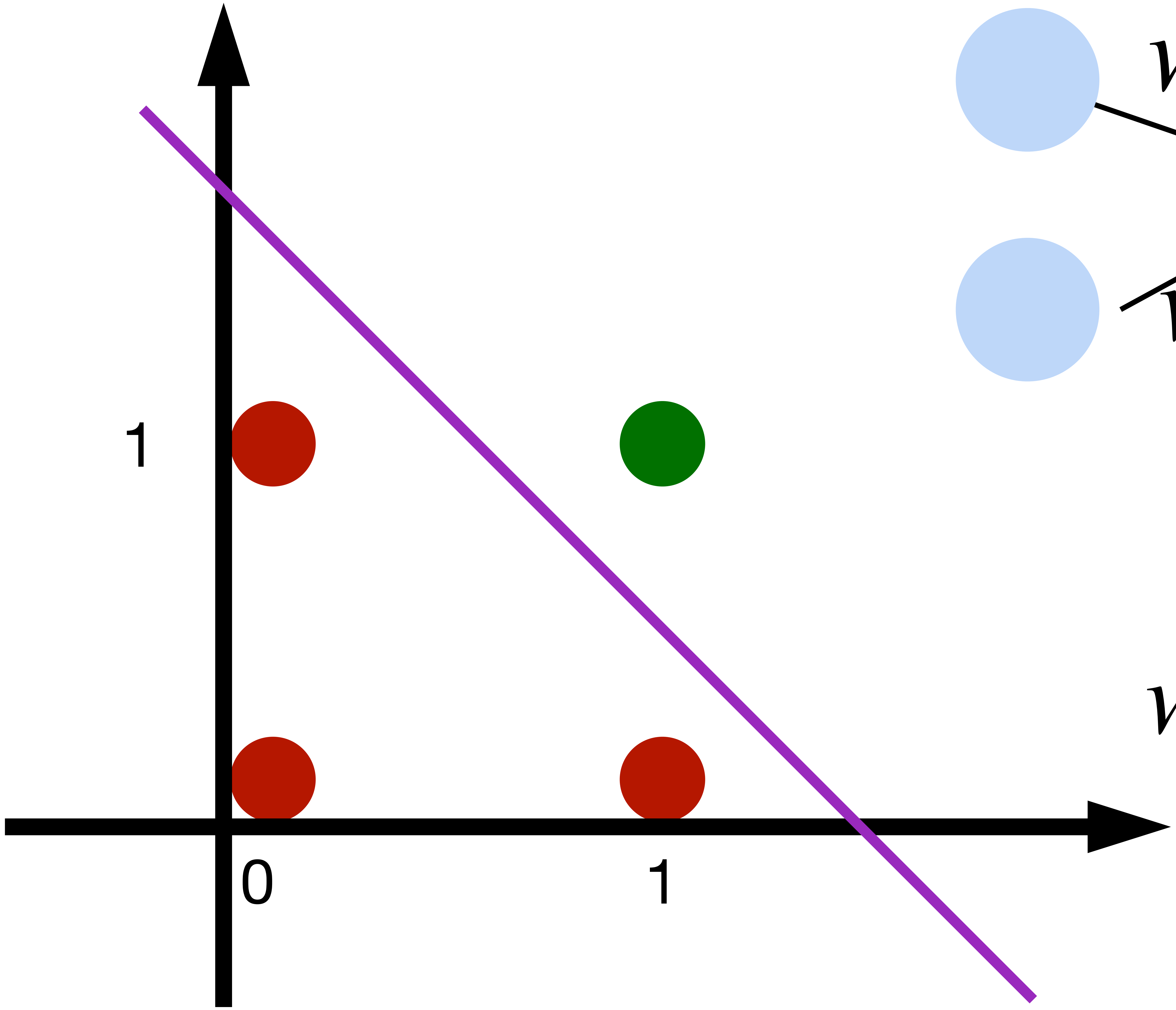
Output $\sigma(x_1w_1 + x_2w_2 + b)$

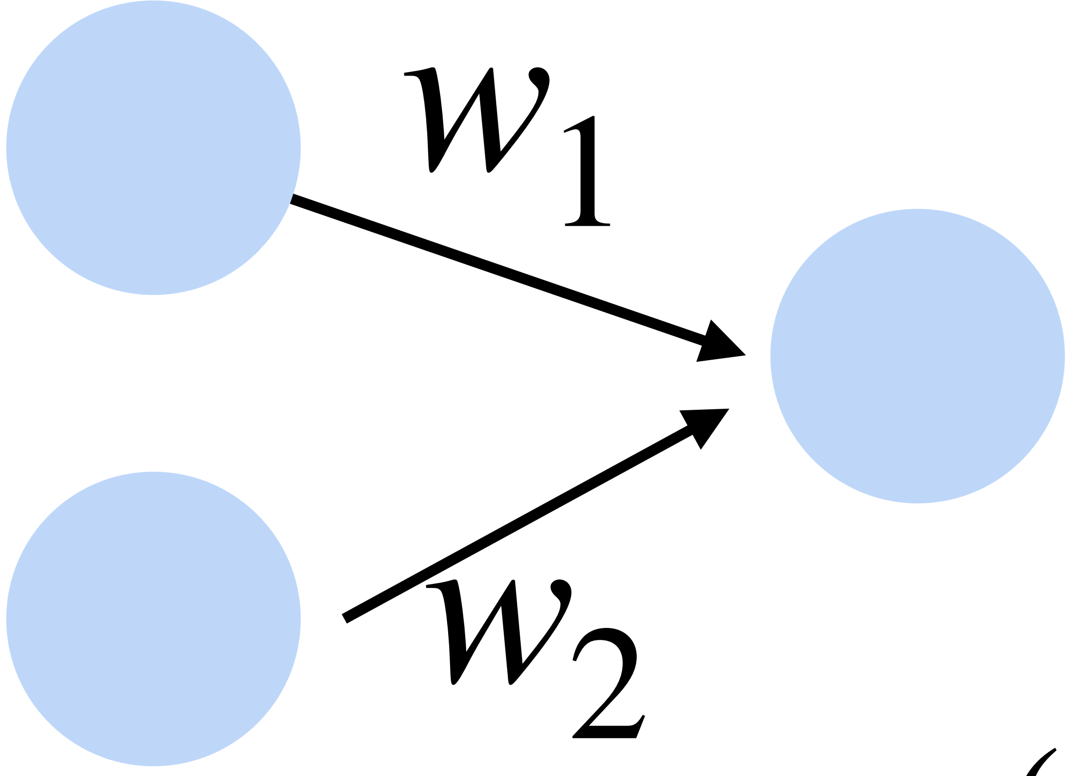
$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

What's w and b?

Learning AND function using perceptron

The perceptron can learn an AND function





Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

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

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

Learning OR function using perceptron

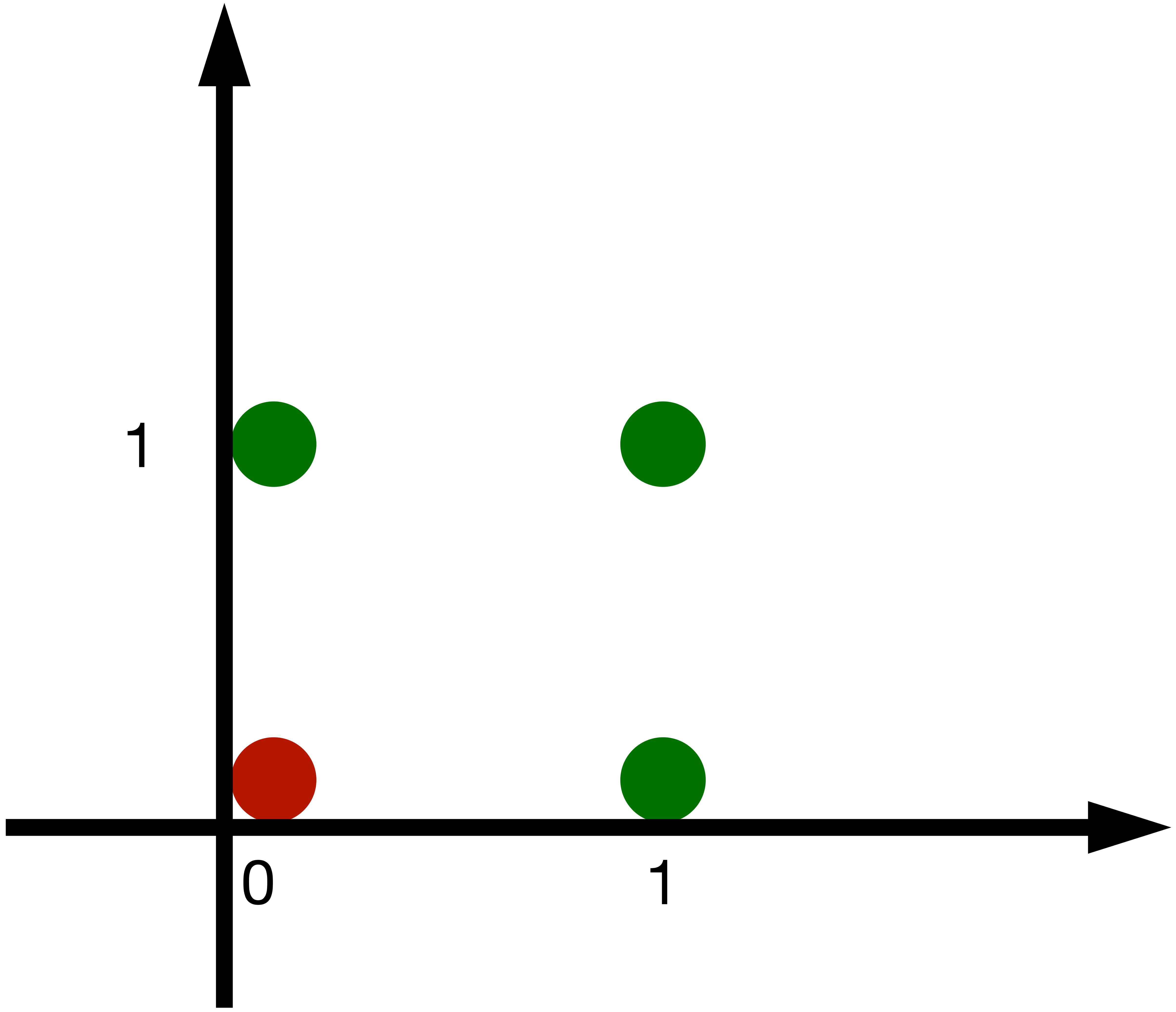
The perceptron can learn an OR function

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

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

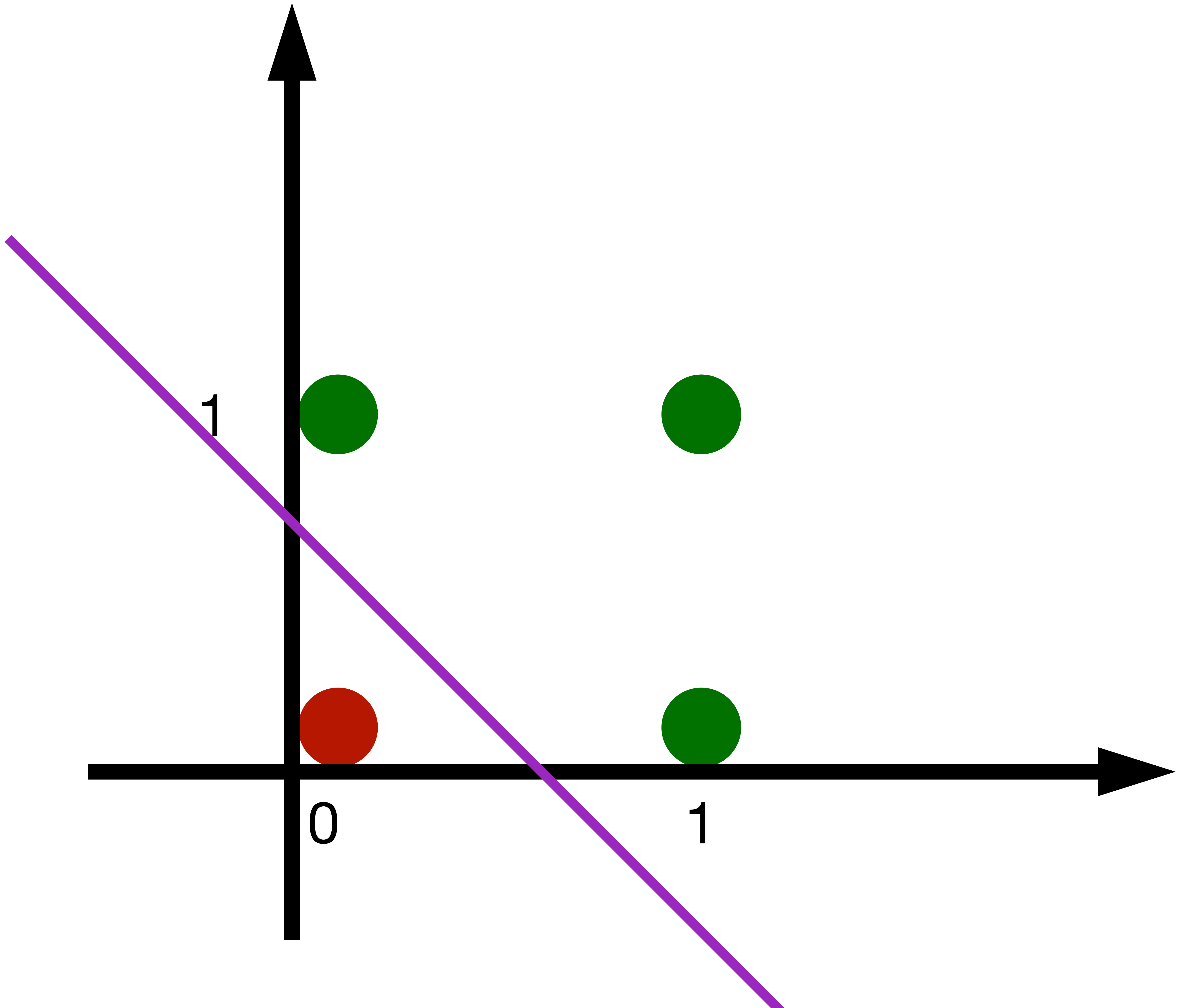
$$x_1 = 0, x_2 = 1, y = 1$$

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



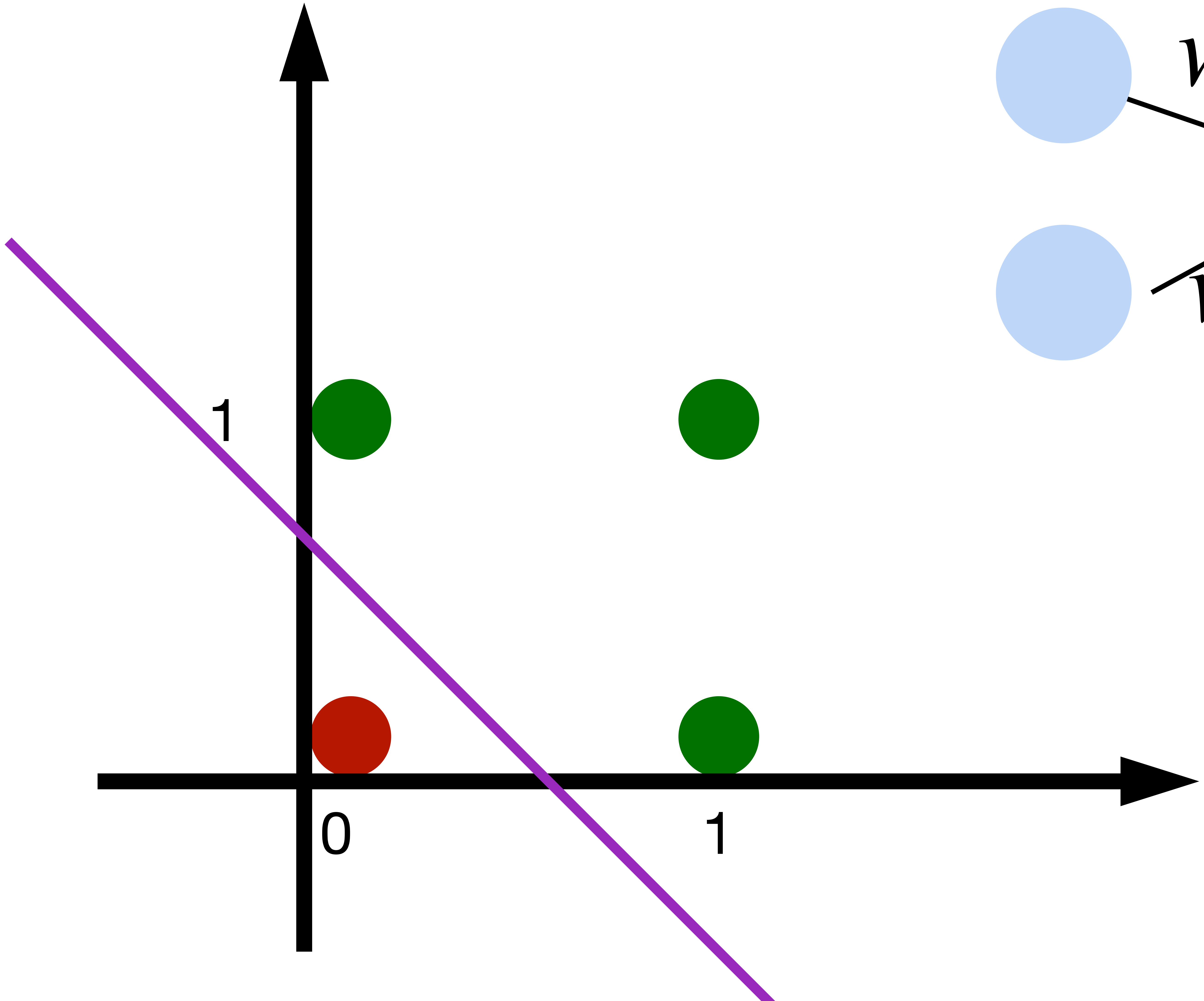
Learning OR function using perceptron

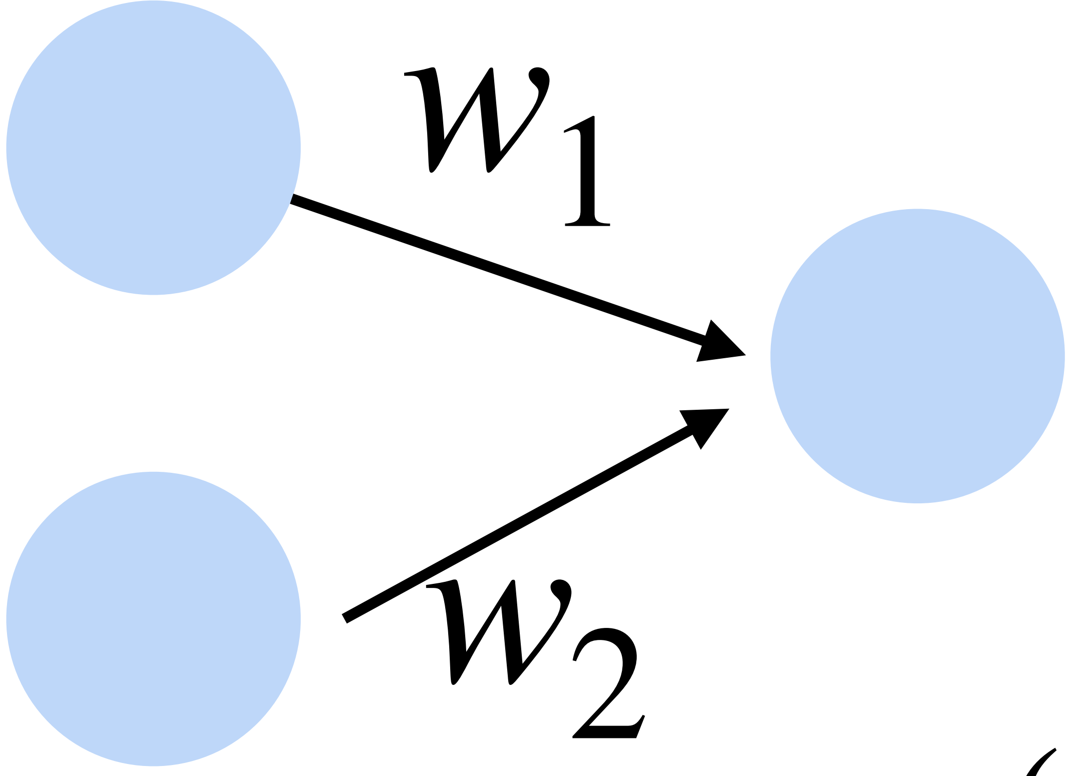
The perceptron can learn an OR function



Learning OR function using perceptron

The perceptron can learn an OR function



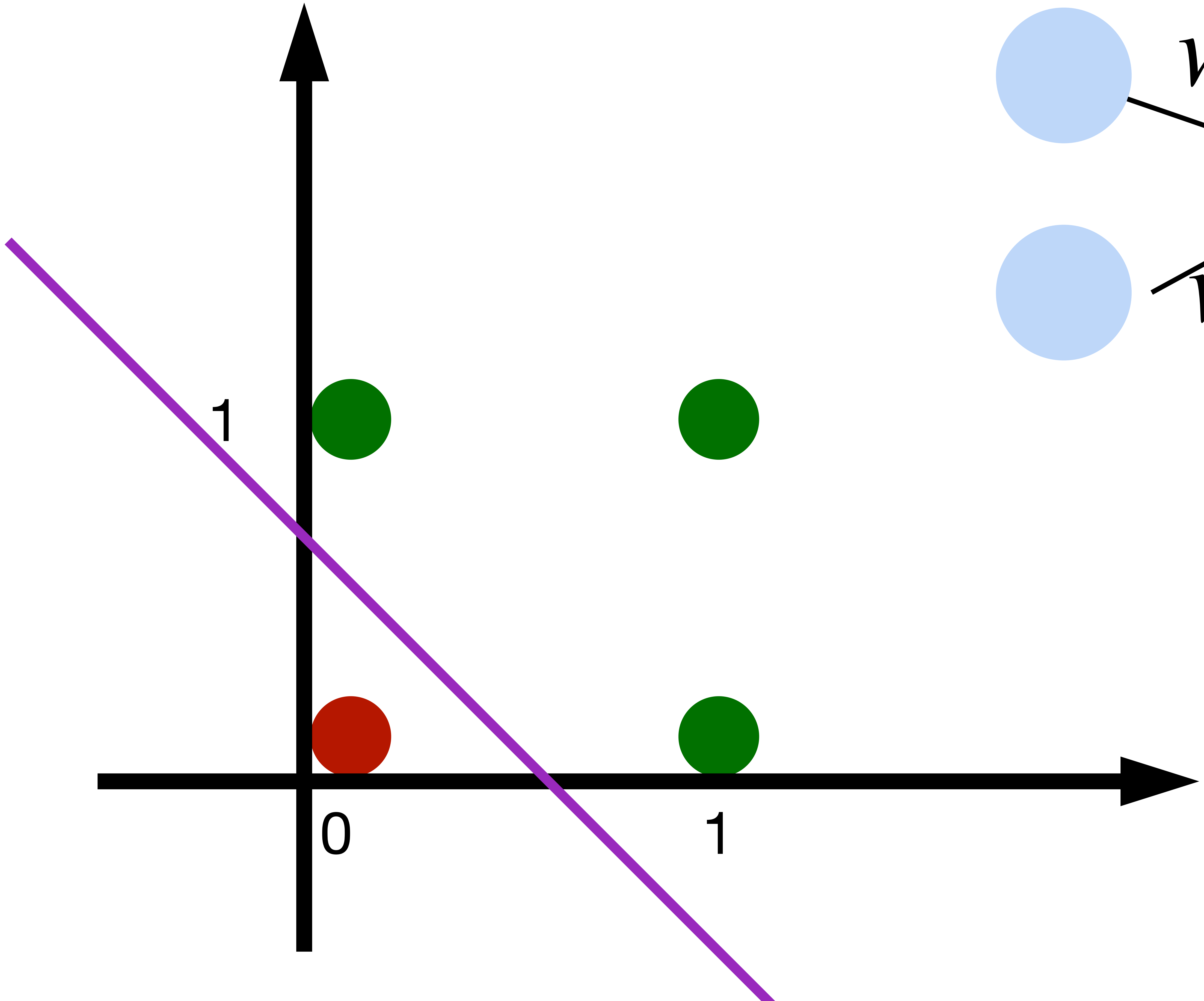


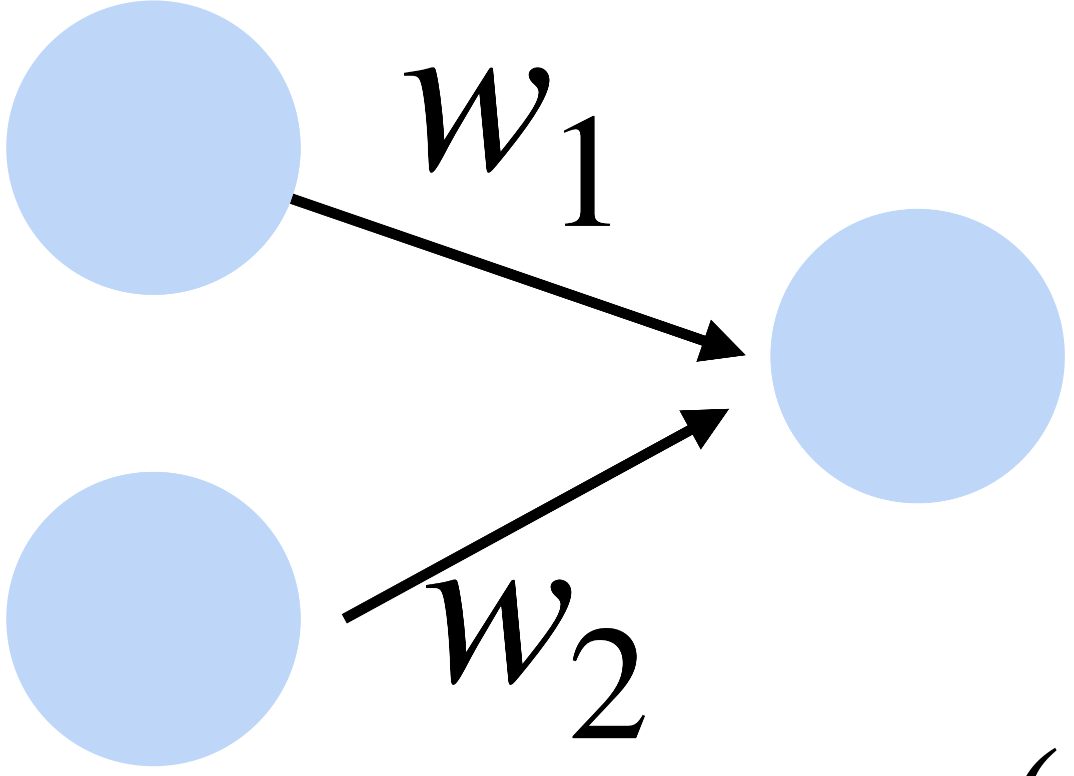
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

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

Learning OR function using perceptron

The perceptron can learn an OR function





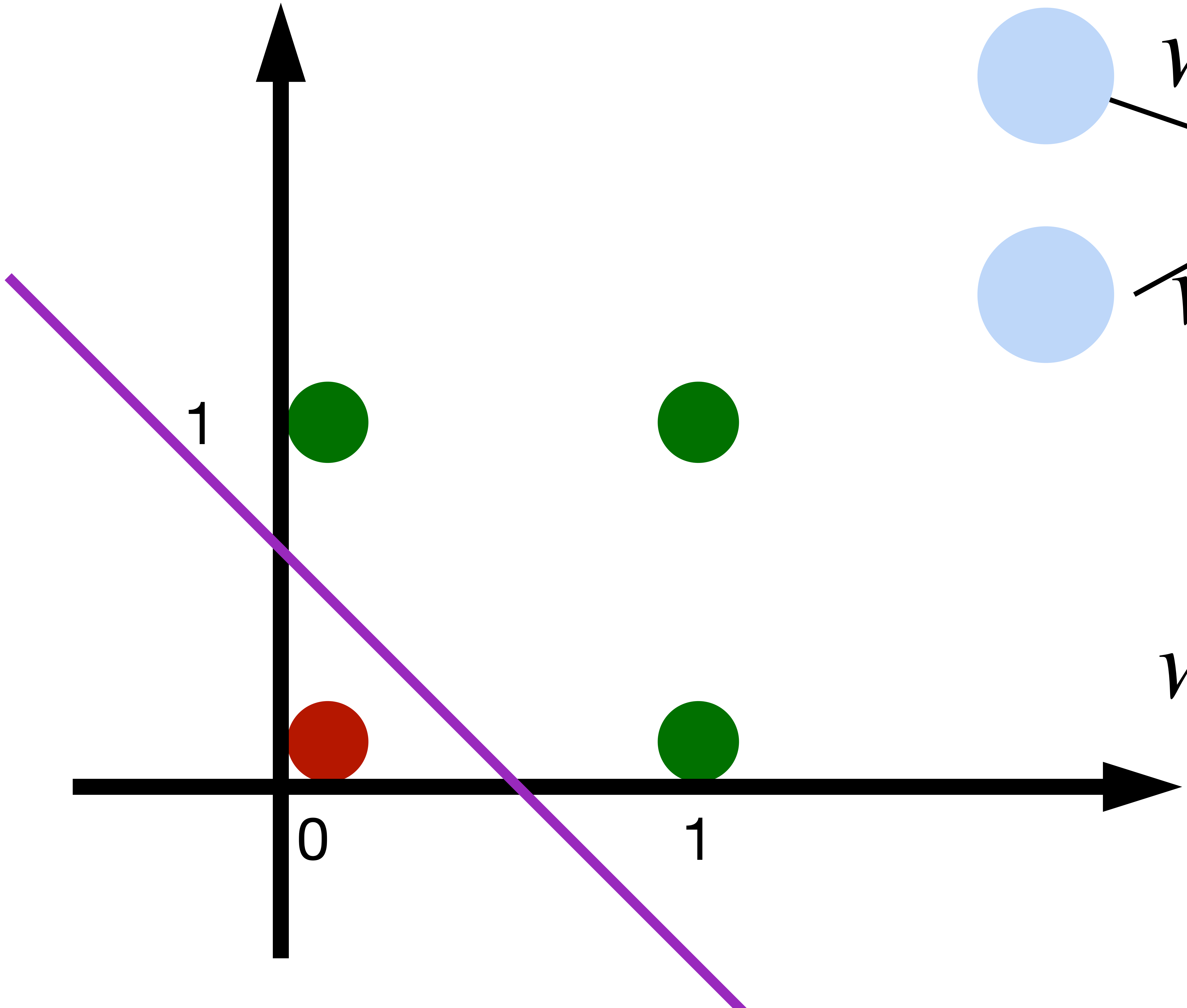
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

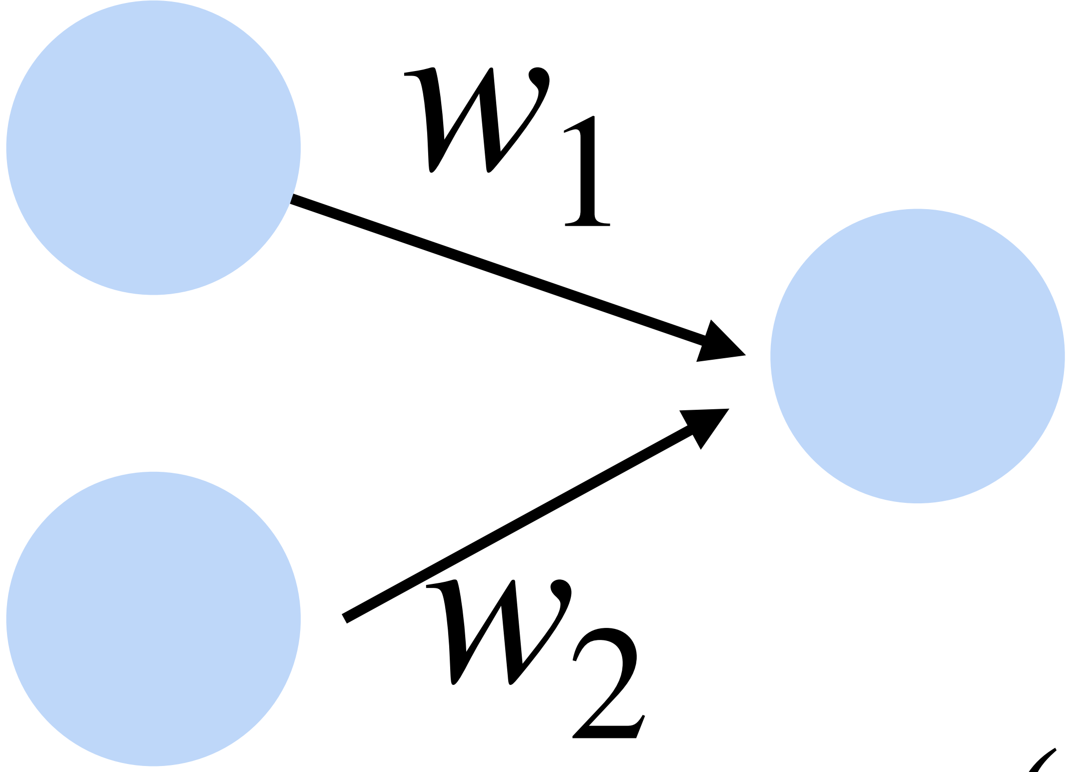
$$\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

What's w and b?

Learning OR function using perceptron

The perceptron can learn an OR function





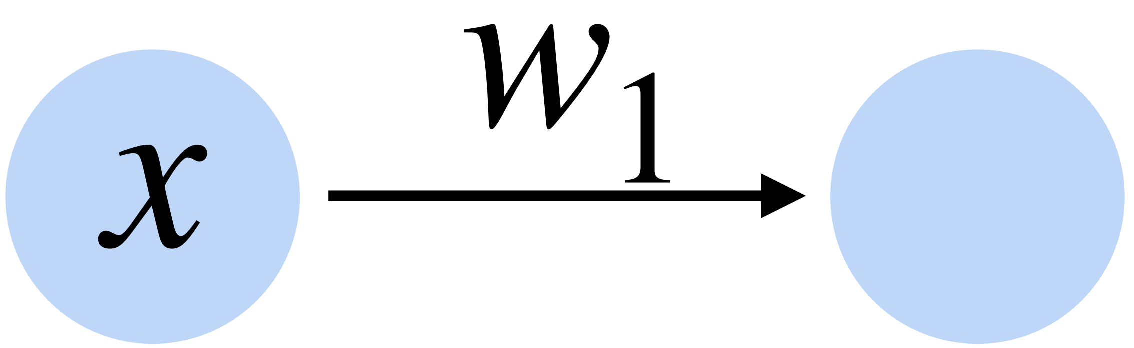
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

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

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

Learning NOT function using perceptron

The perceptron can learn NOT function (single input)

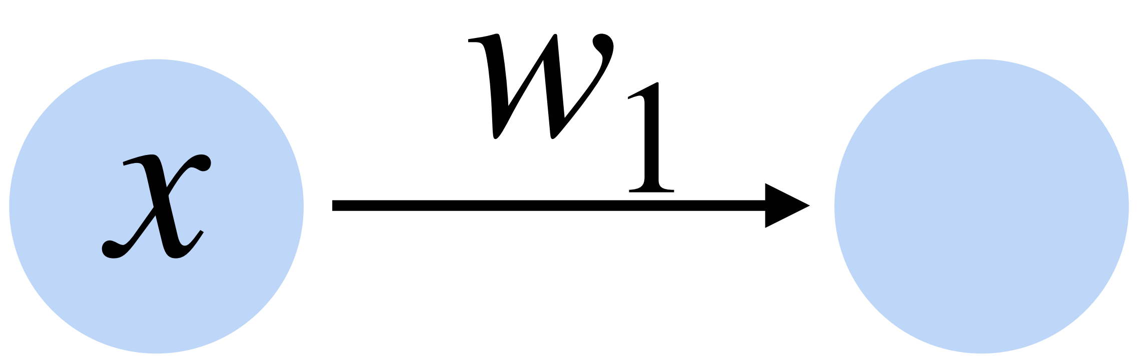


Output $\sigma(xw_1 + b)$

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

Learning NOT function using perceptron

The perceptron can learn NOT function (single input)



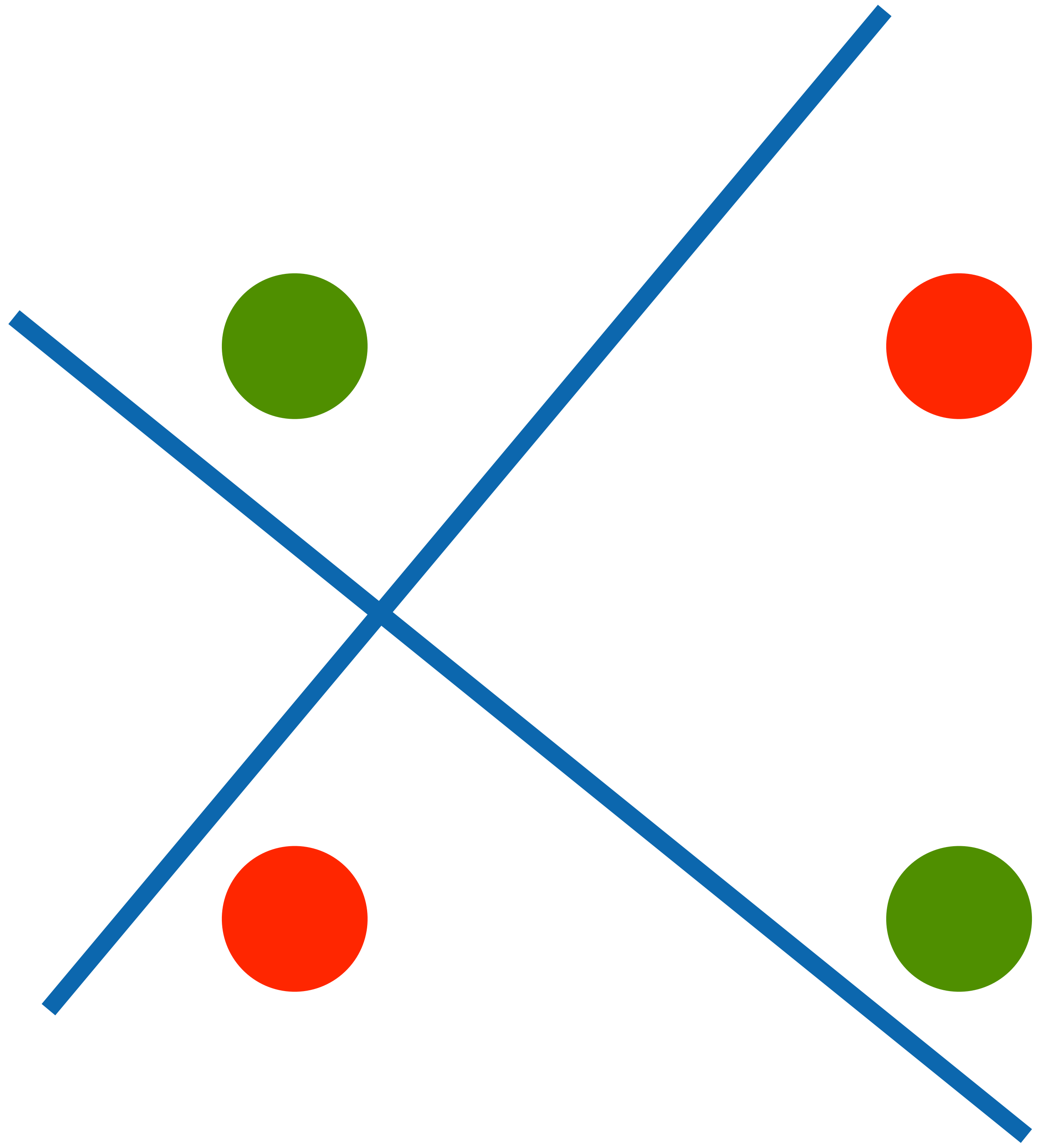
Output $\sigma(xw_1 + b)$

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

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

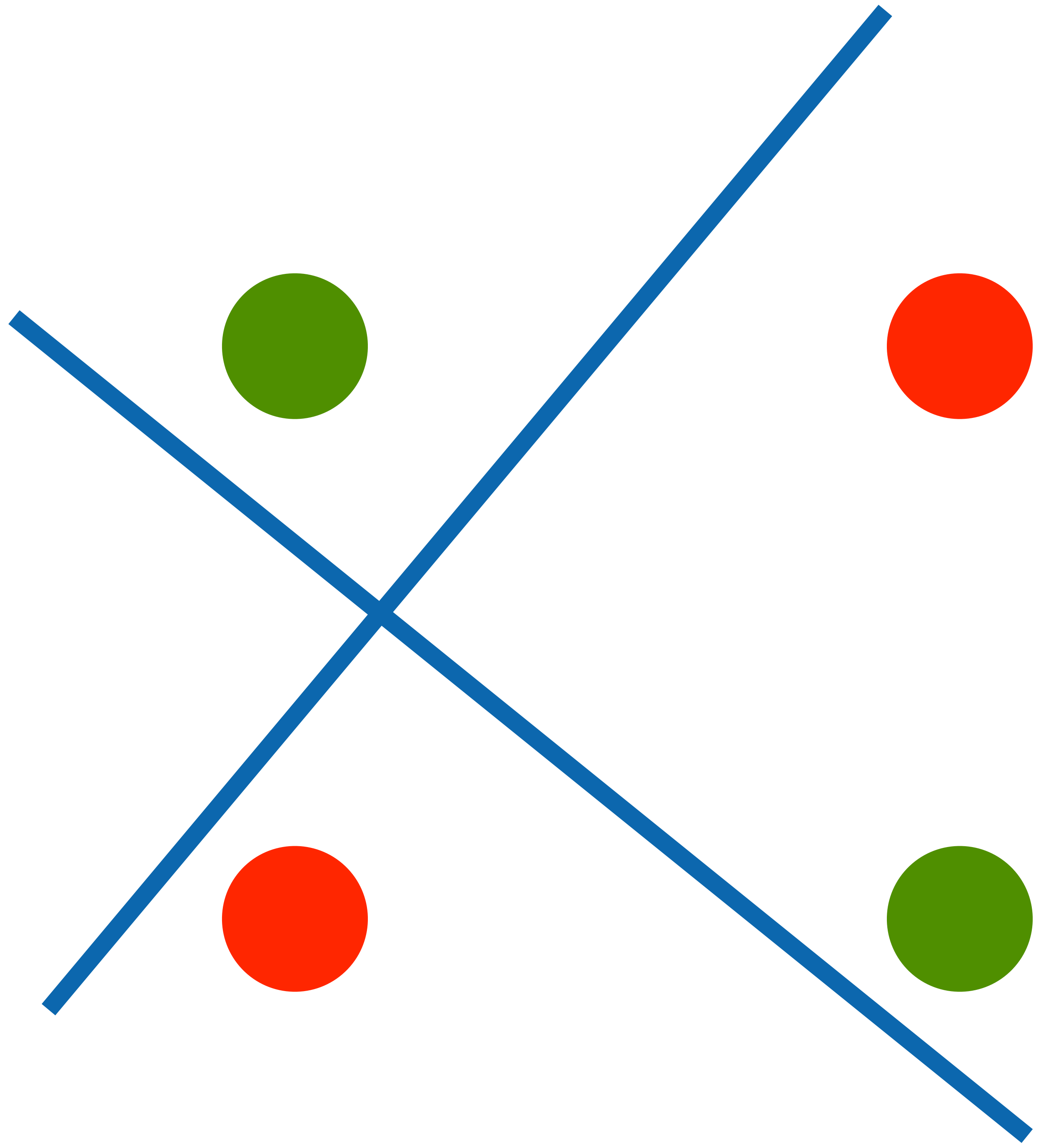
XOR Problem (Minsky & Papert, 1969)

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



XOR Problem (Minsky & Papert, 1969)

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



This contributed to the first AI winter

Thanks!