CS 540 Introduction to Artificial Intelligence
Perceptron

University of Wisconsin-Madison
Spring 2026 Sections 1 & 2

Announcements

- **Homework**:
  - HW5 due on **Wednesday March 4 at 11:59 PM**

- TA Discussion/Review Session on **Thursdays at 5:30 PM in Morgridge Hall 3610**.

- Class roadmap:

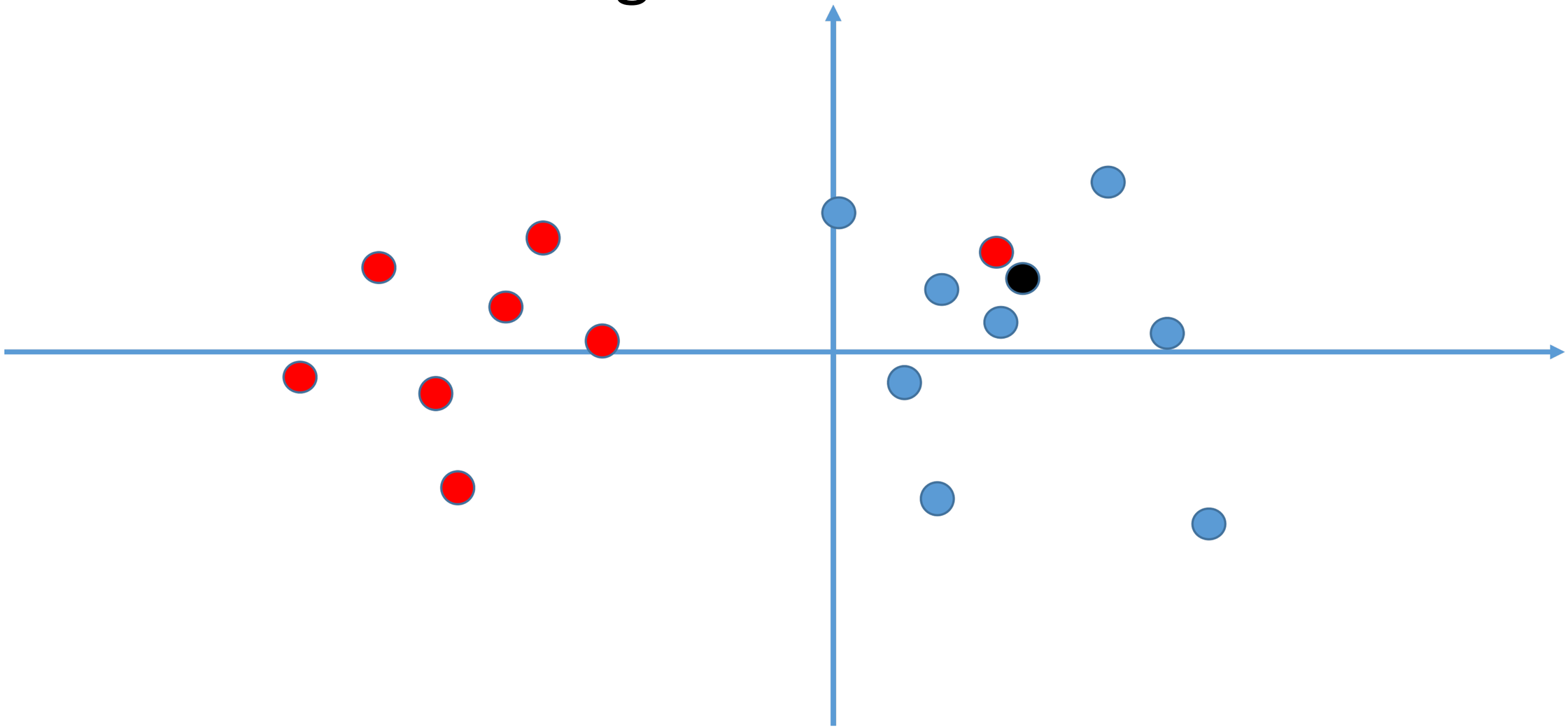
Machine Learning: Neural Networks I (Perceptron)
Machine Learning: Neural Networks II
Machine Learning: Neural Networks III

Supervised Learning

Outline

- Review KNN, MLE, Naïve Bayes
- Single-layer Neural Network
- Activation Functions
- Multi-layer Neural Network

Review: k-Nearest Neighbors

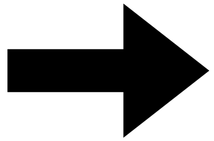


Review: Supervised Machine Learning

Parametric Supervised Machine Learning

$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

Given some labeled training data drawn **independently** from a fixed underlying distribution, also called the i.i.d. assumption



select $\hat{f}(\theta)$ from a pool of models $\mathcal{F}$ that **best describe the data observed**

Maximum likelihood: best fits the data

Review: Bayesian Classification via MLE

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

(Prediction) (Posterior)

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

(by Bayes' rule)

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

Using labelled training data, learn **class priors** and **class conditionals**

Review: Bayesian classification

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

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

(Prediction) (Posterior)

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

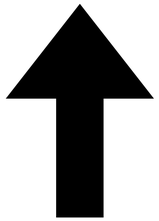
(by Bayes' rule)

Independent of y

Review: Naïve Bayes Assumption

Conditional independence of feature attributes

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



Easier to estimate

(using MLE!)

Quiz break

Q3-2: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other.
We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

- A Pass
- B Fail

Quiz break

Q3-2: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other.
We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

- **A Pass**
- B Fail

Quiz break

We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

- A Pass
- B Fail

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

$$P(y = P|x_1 = Y, x_2 = Y, x_3 = N)$$
$$= \frac{P(x_1 = Y|Y = P)P(x_2 = Y|Y = P)P(x_3 = N|Y = P)P(y = P)}{P(x_1 = Y, x_2 = Y, x_3 = N)}$$
$$= \frac{2}{3} * \frac{2}{3} * \frac{1}{3} * \frac{3}{5} / P(x_1 = Y, x_2 = Y, x_3 = N)$$

$$\propto \frac{4}{9*5}$$ Larger!

$$P(y = F|x_1 = Y, x_2 = Y, x_3 = N)$$
$$= \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{2}{5} / P(x_1 = Y, x_2 = Y, x_3 = N)$$

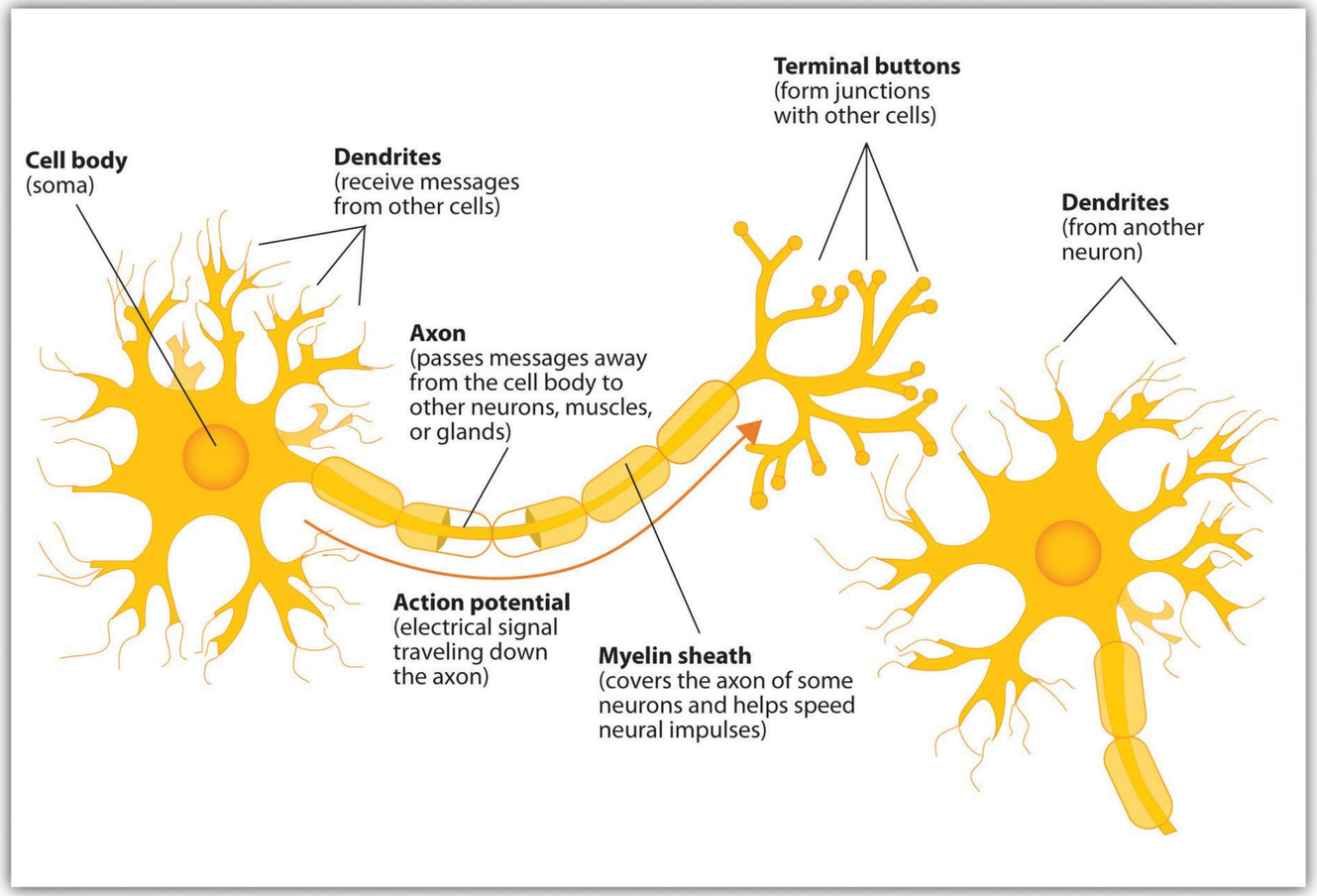
$$\propto \frac{1}{4 * 5}$$

Single-layer Neural Network

Inspiration from neuroscience

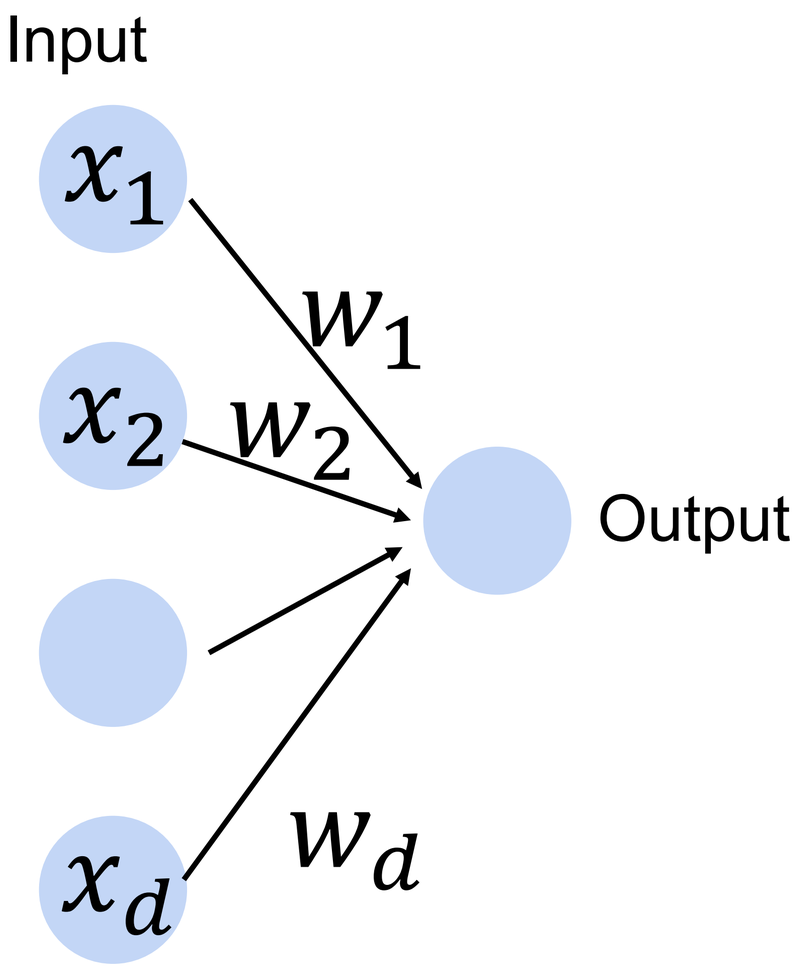
- Inspirations from human brains
- Networks of simple and homogenous units

(wikipedia)



Perceptron

Cats vs. dogs?

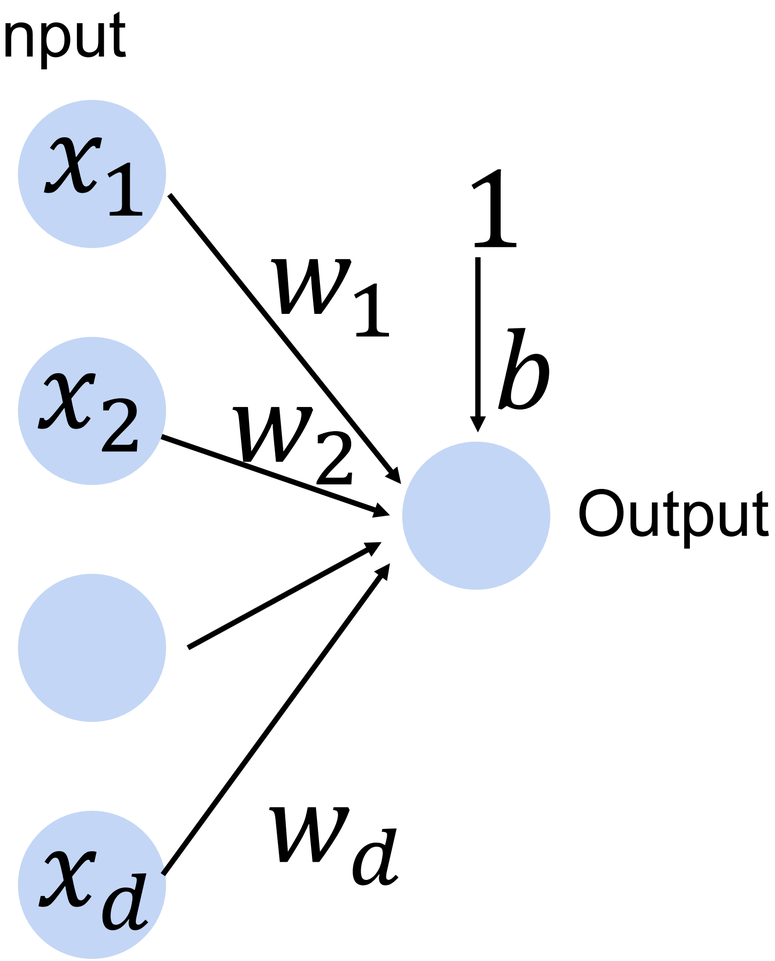


Linear Perceptron

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

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

Cats vs. dogs?



Perceptron

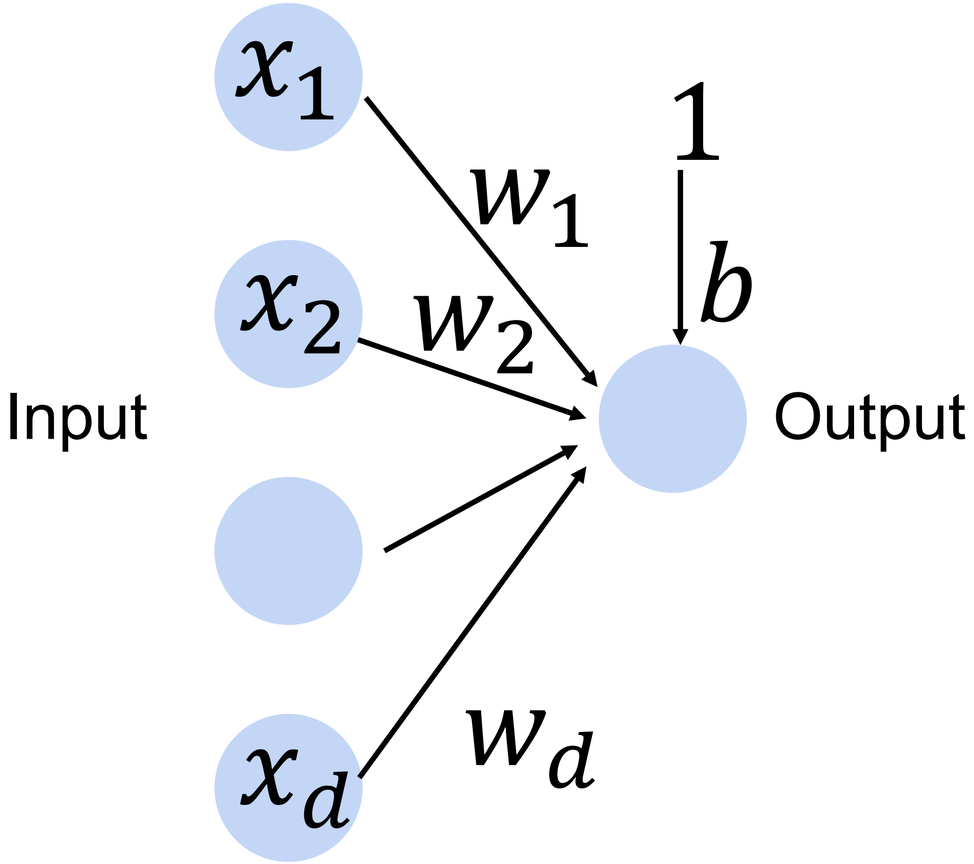
- Given input $\mathbf{x}$, weight $\mathbf{w}$ and bias b, perceptron outputs:

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

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

Activation function

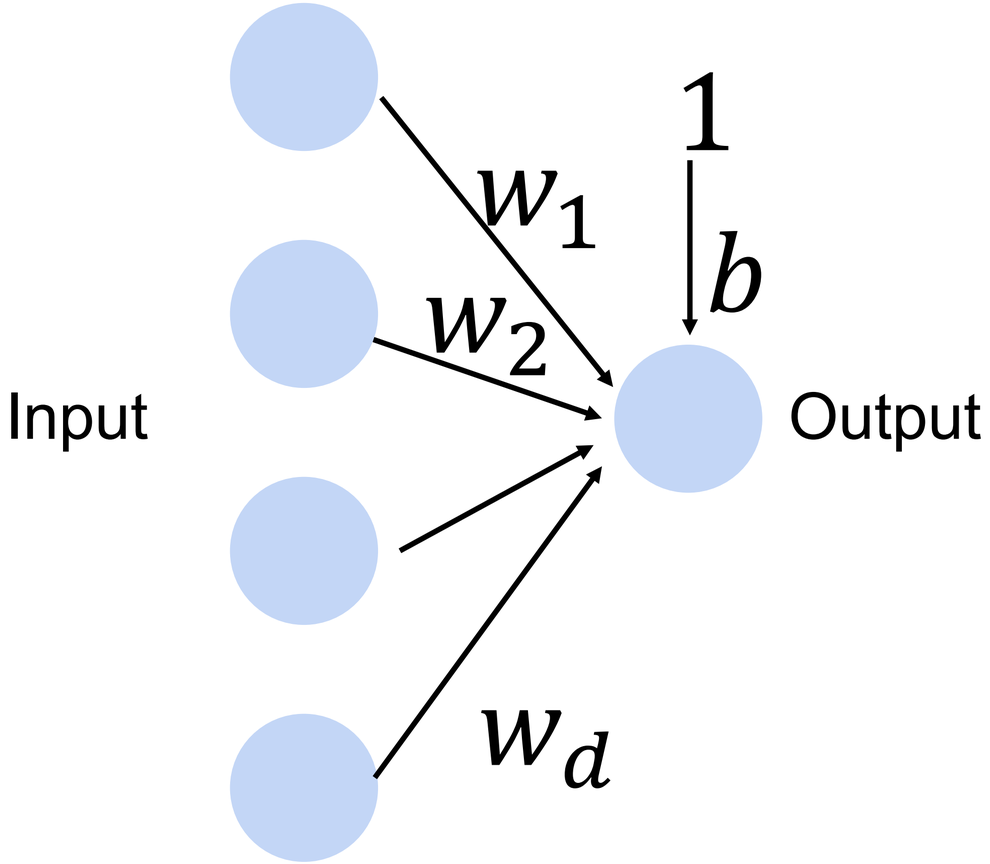
Cats vs. dogs?



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

- Iterative algorithm to decrease loss on training data
  - Start with initial weights w_0, b_0
  - When perceptron misclassifies a point, update the weights

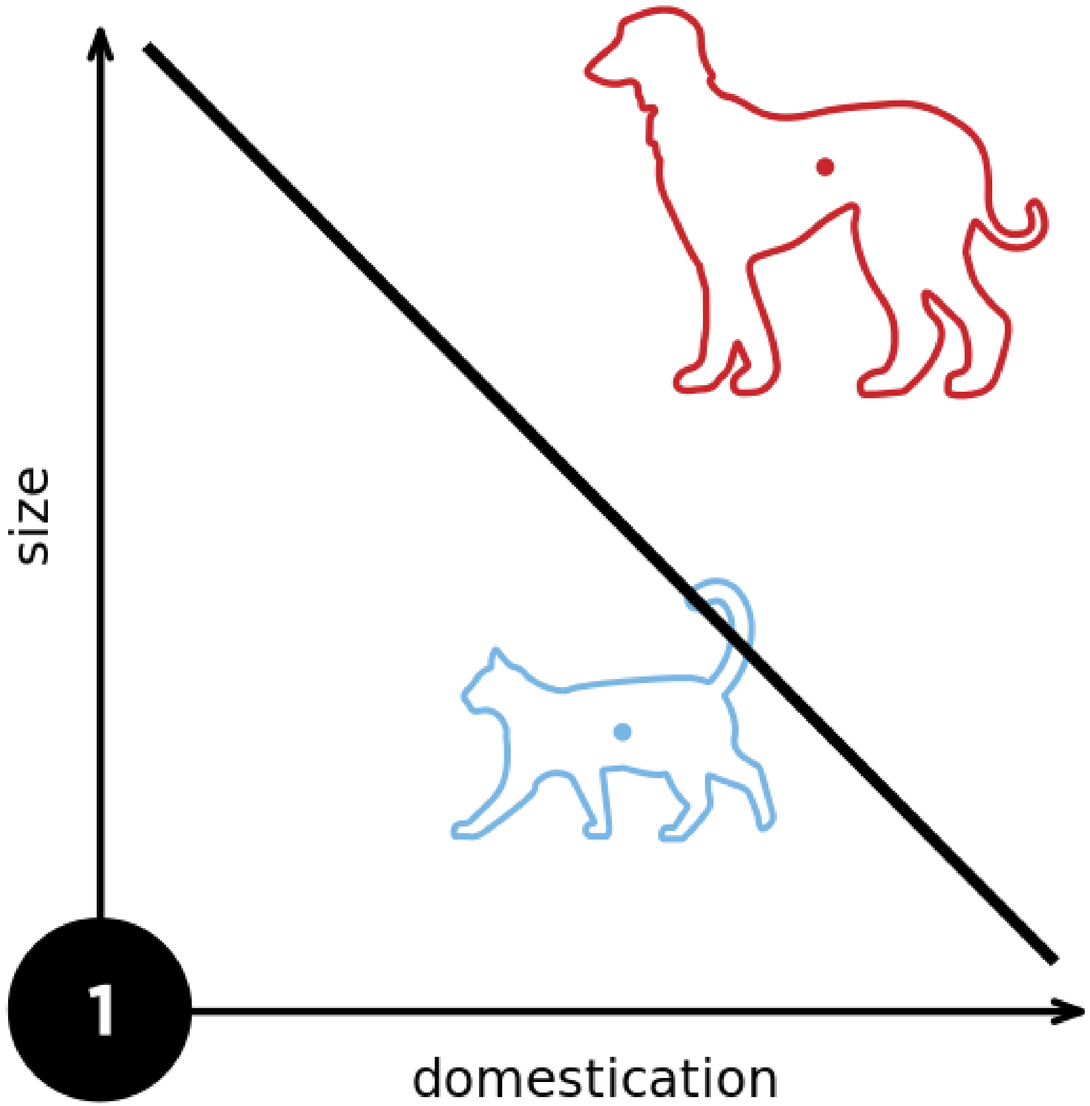
- Randomly pick next training example
  - It is a **stochastic** training algorithm
  - Similar to stochastic gradient descent (SGD)

The Perceptron Learning Rule

Perceptron Learning Algorithm
Input: dataset (X, y) number of steps T, step size η
1. Initialize w_0, b_0 2. For $t = 1, 2, \ldots, T$ 3. Pick random (x_i, y_i) 4. Predict $\hat{y}_i \leftarrow \sigma(\langle w_{t-1}, x_i \rangle + b_{t-1})$ 5. If $\hat{y}_i \neq y_{\mathrm{i}}$: 6. $w_t \leftarrow w_{t-1} + \eta(y_i - \hat{y}_i)x_i$ 7. $b_t \leftarrow b_{t-1} + \eta(y_i - \hat{y}_i)$ 8. Return w_T

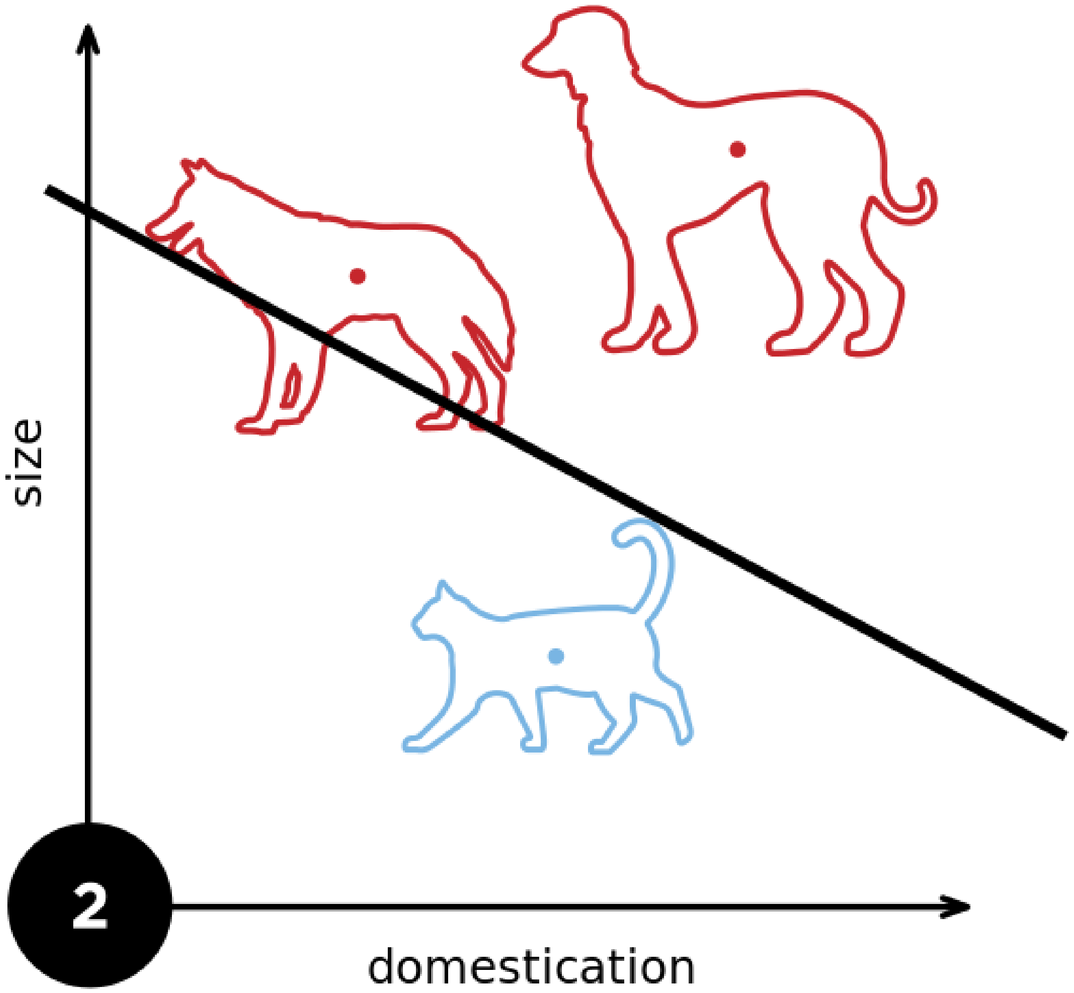
Gradient Descent
Input: dataset (X, y), loss function L, number of steps T, step size η
1. Initialize w_0 2. For $t = 1, 2, \ldots, T$ 3. Calculate $g_t = \nabla L(w_{t-1}; X, y)$ 4. Update $w_t \leftarrow w_{t-1} - \eta g_t$ 5. Return w_T

Perceptron



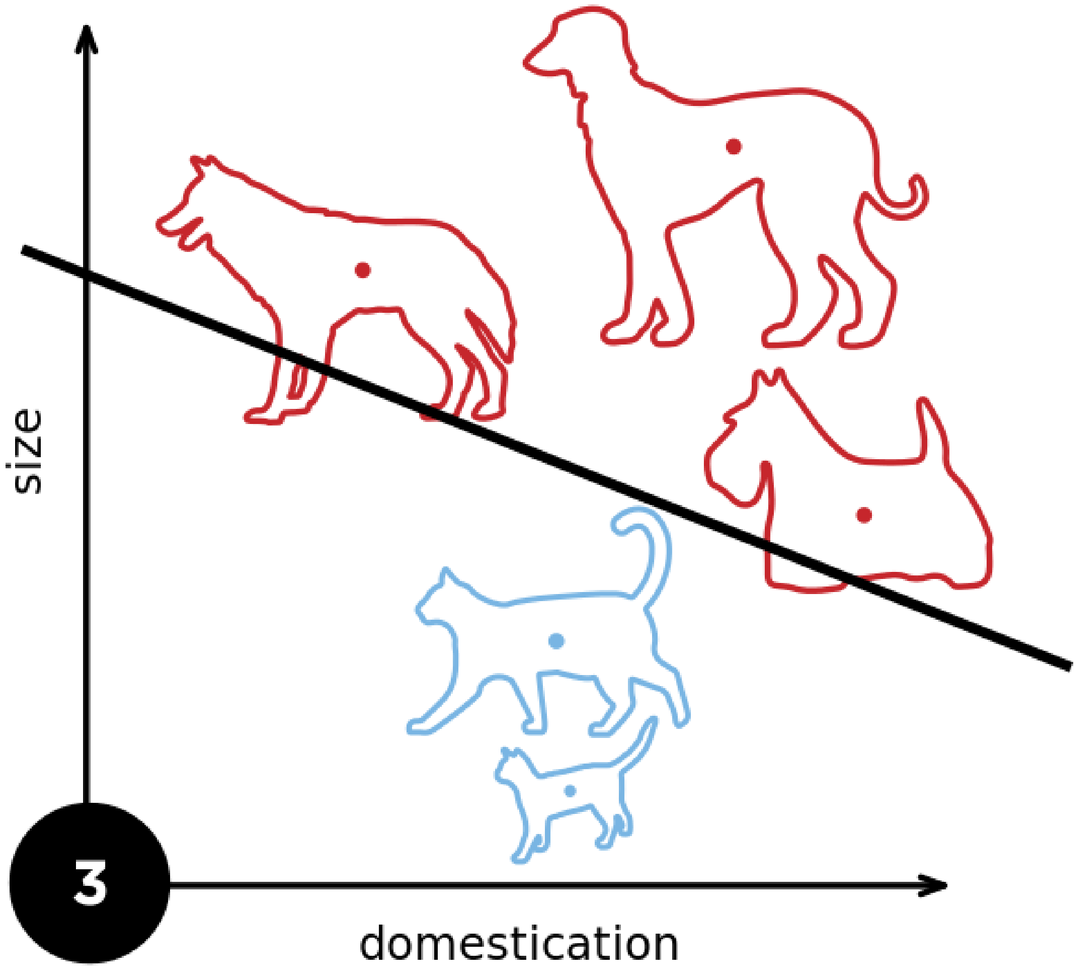
From wikipedia

Perceptron



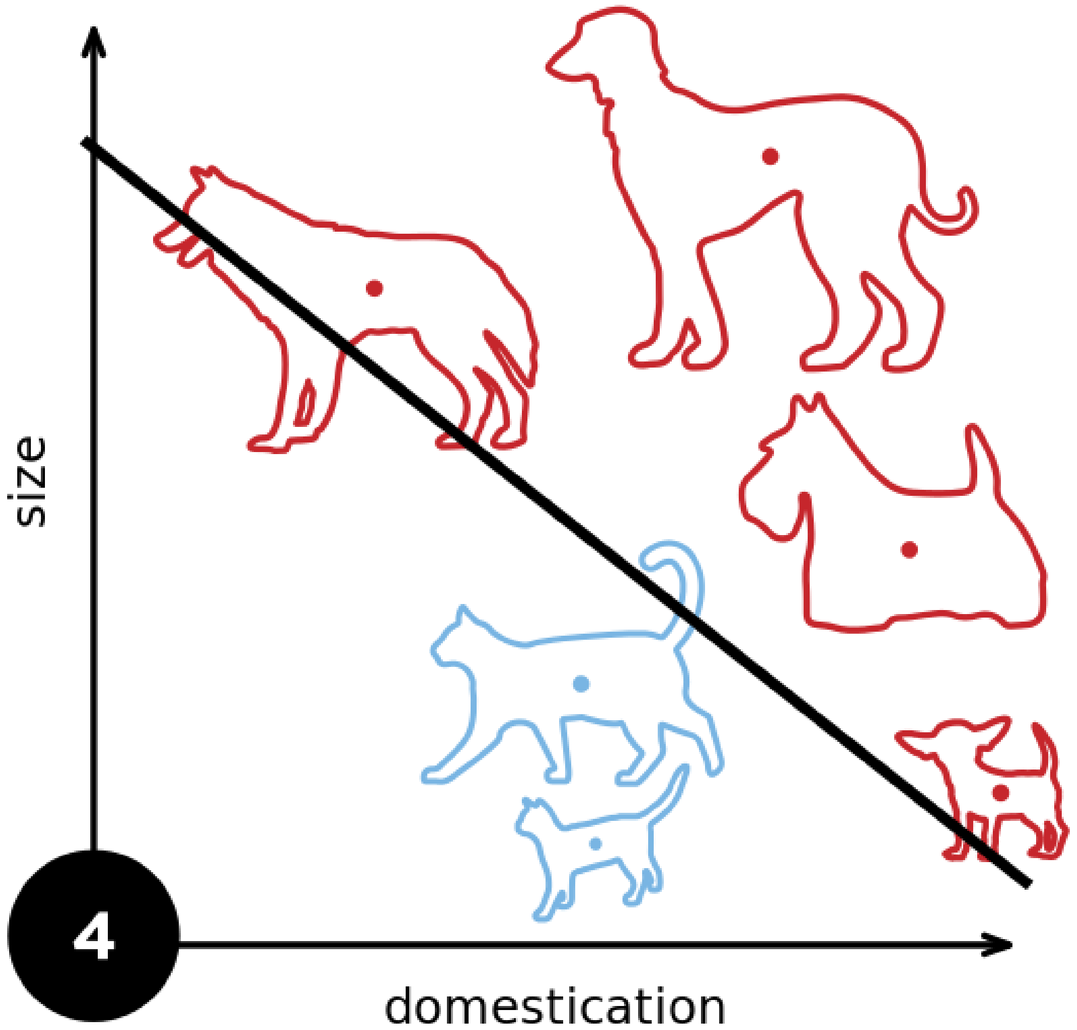
From wikipedia

Perceptron



From wikipedia

Perceptron



From wikipedia

Learning AND function using perceptron

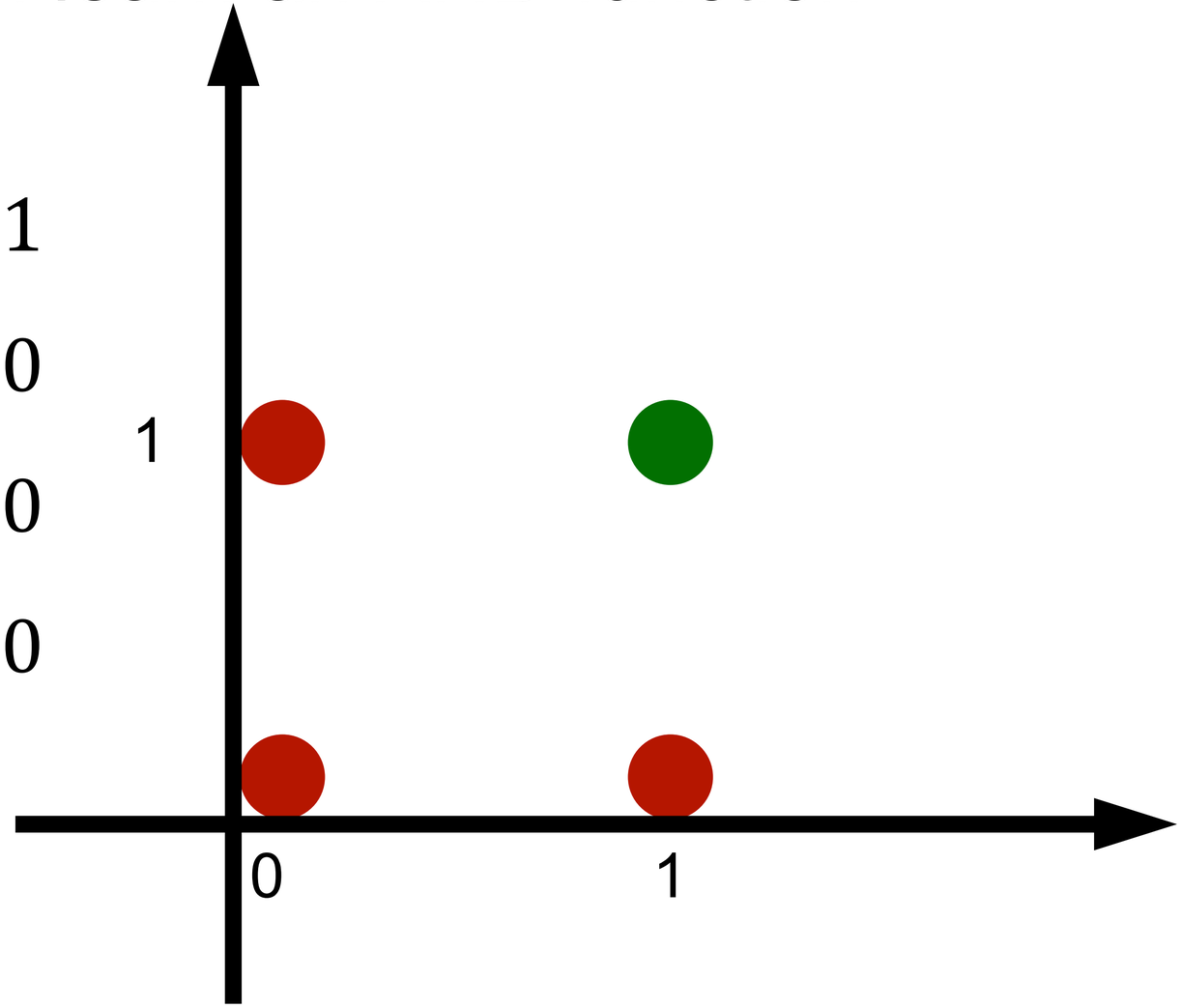
The perceptron can learn an AND function

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

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

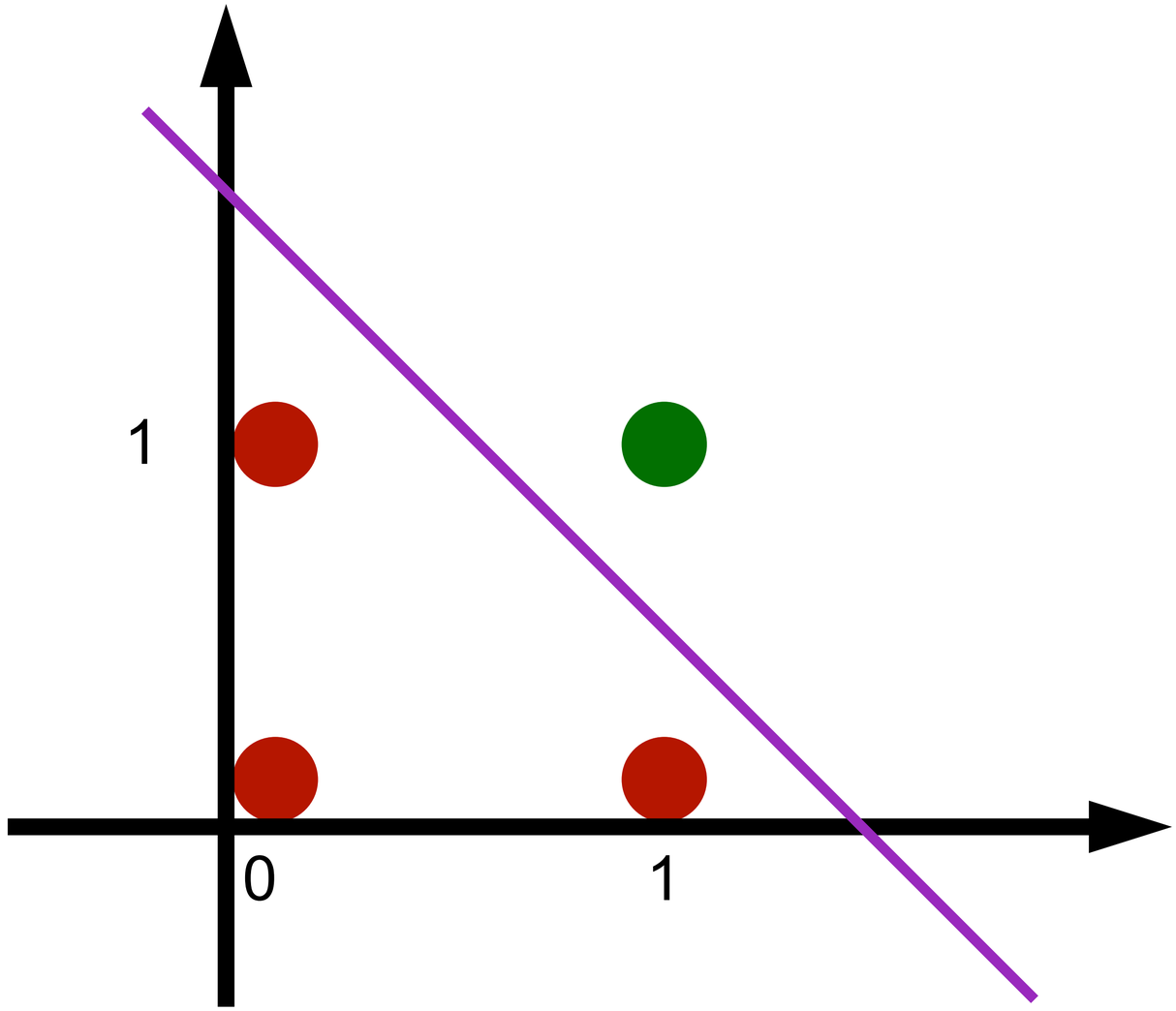
$x_1 = 0, x_2 = 1, y = 0$

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



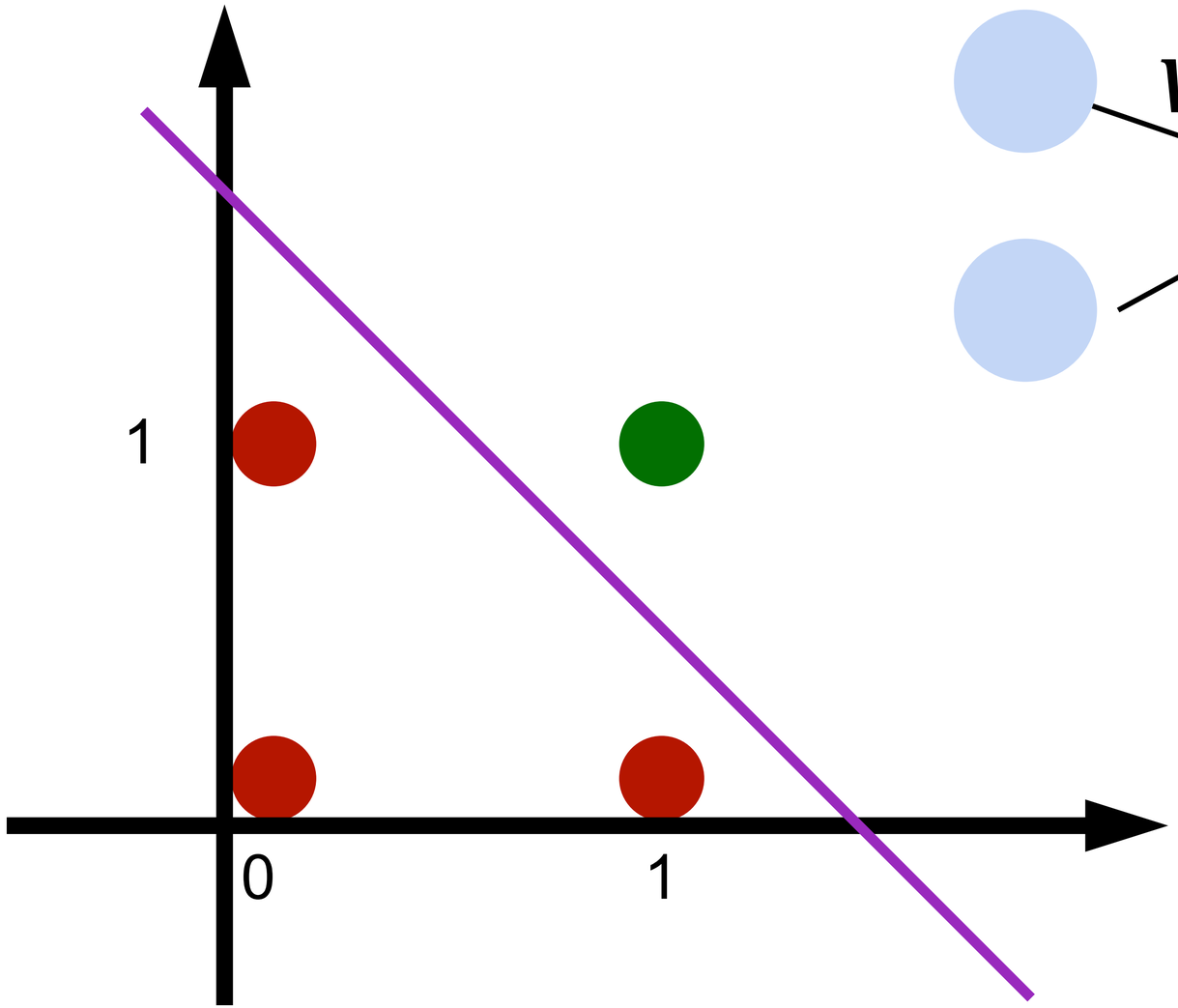
Learning AND function using perceptron

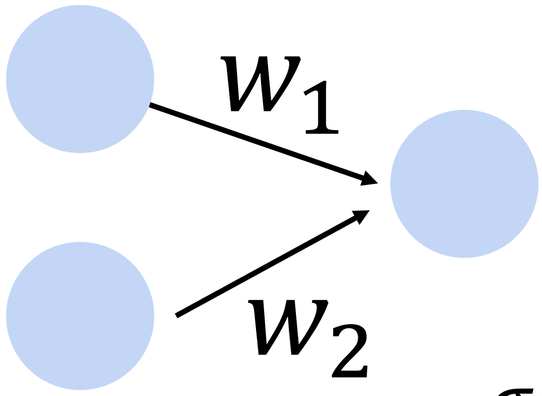
The perceptron can learn an AND function



Learning AND function using perceptron

The perceptron can learn an AND function





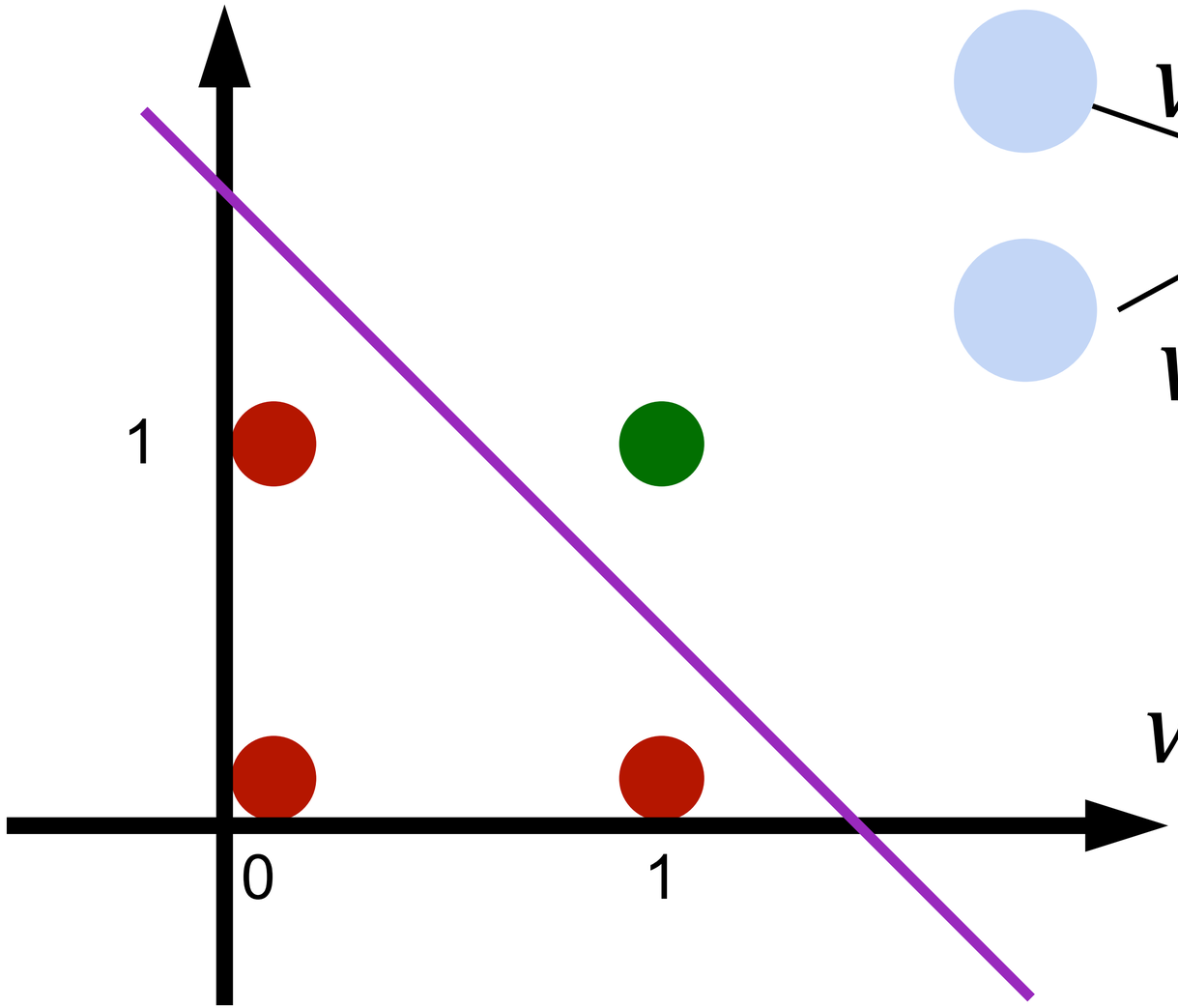
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

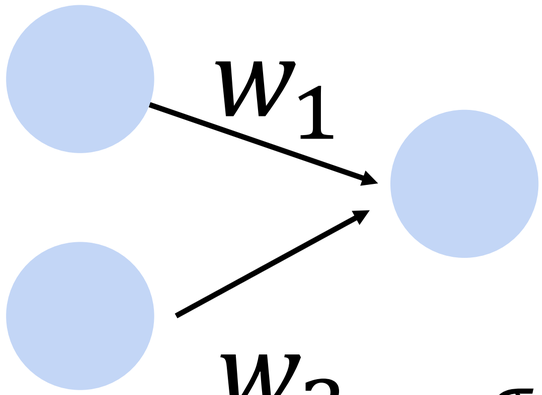
$$\sigma(x) = \begin{cases} 1 & if\ x > 0 \\ 0 & otherwise \end{cases}$$

What's w and b?

Learning AND function using perceptron

The perceptron can learn an AND function





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

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

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

Learning OR function using perceptron

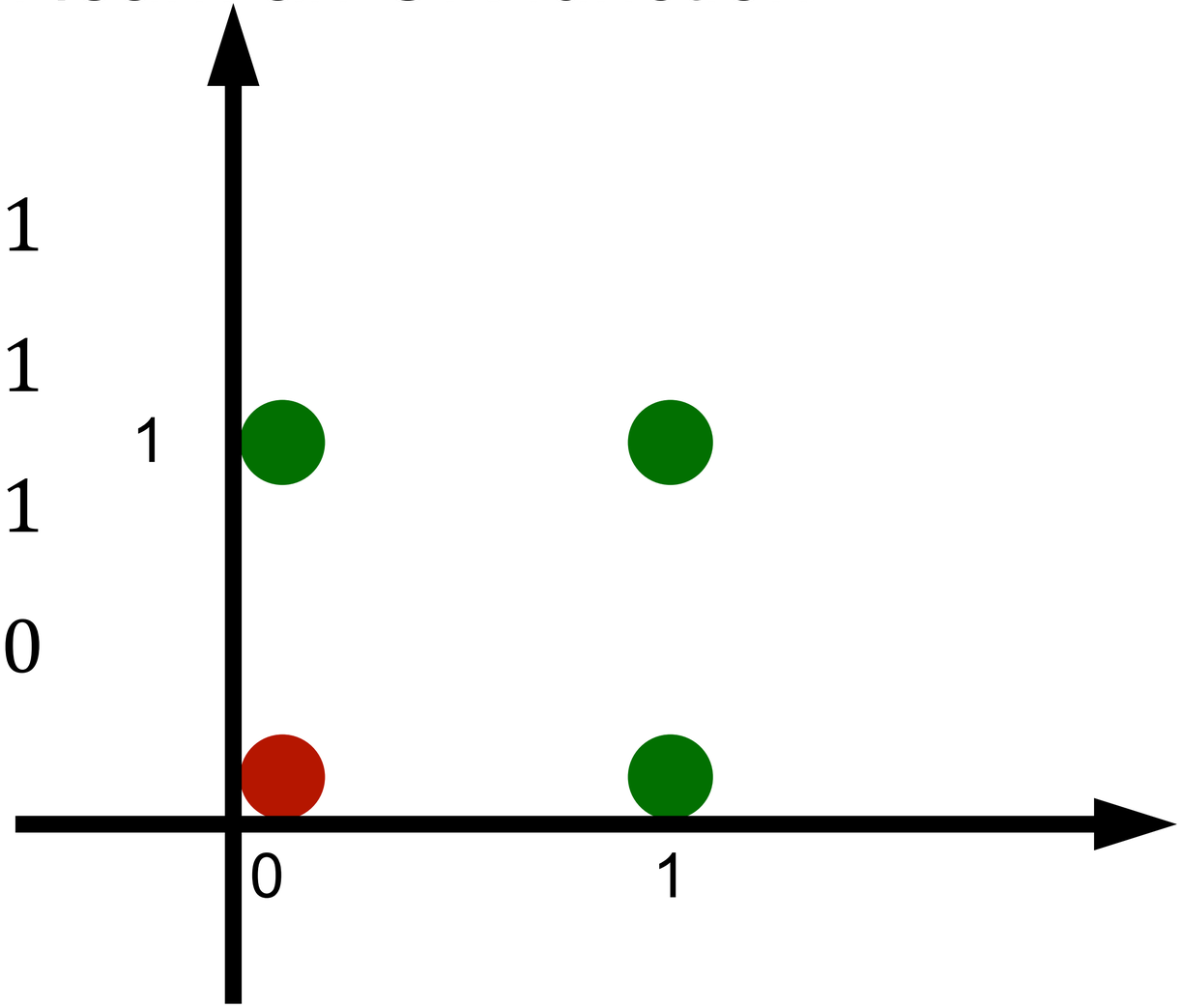
The perceptron can learn an OR function

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

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

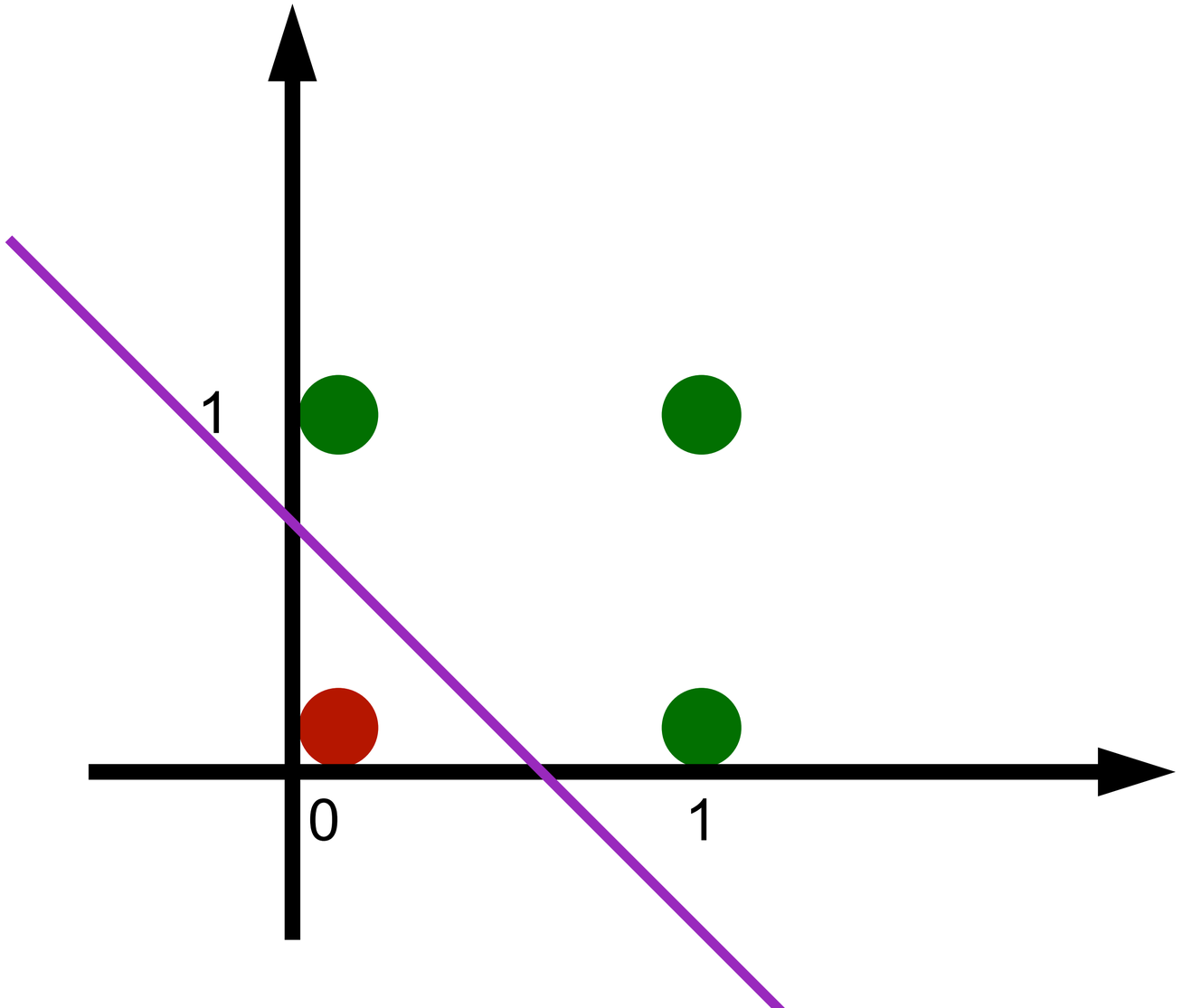
$x_1 = 0, x_2 = 1, y = 1$

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



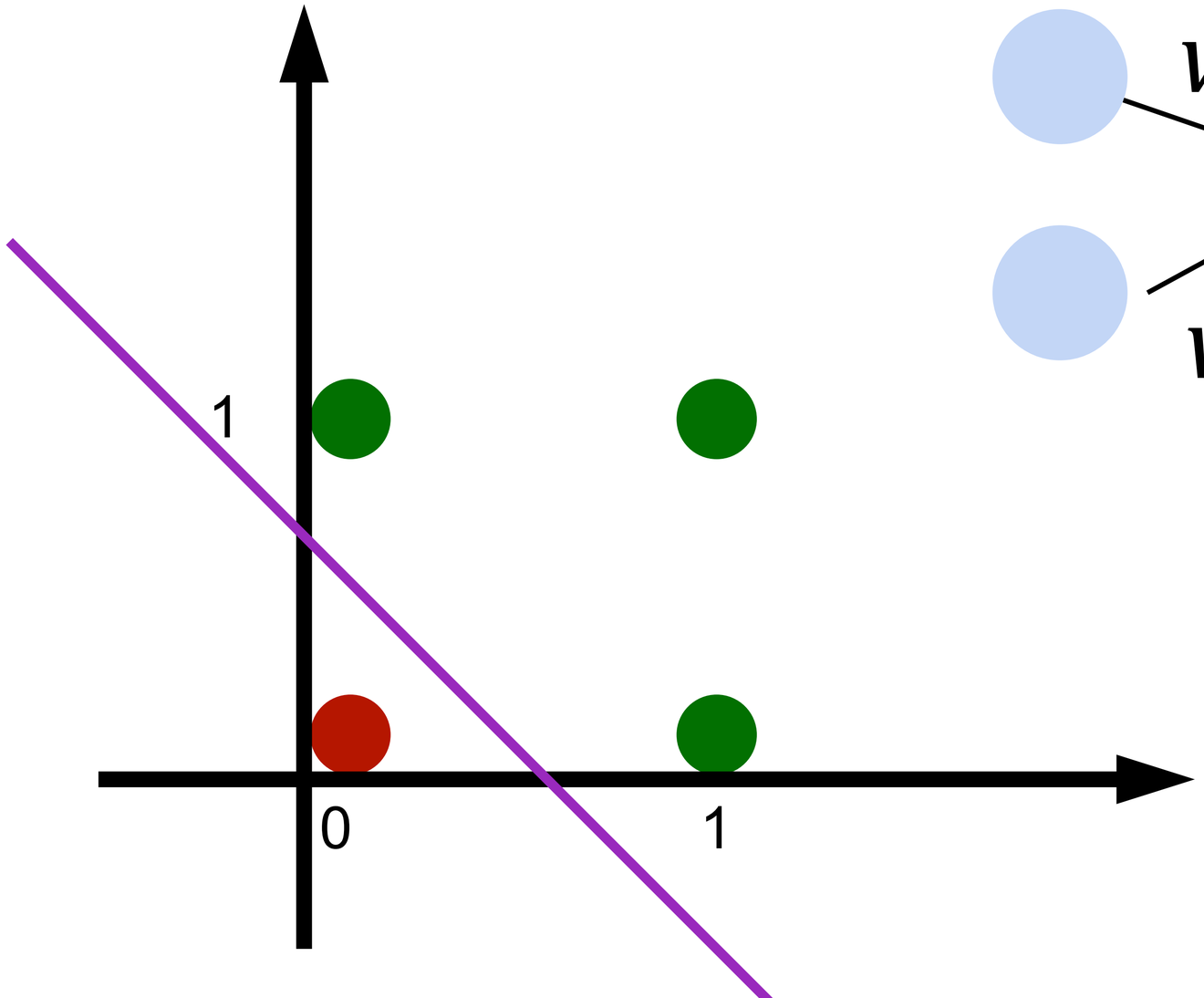
Learning OR function using perceptron

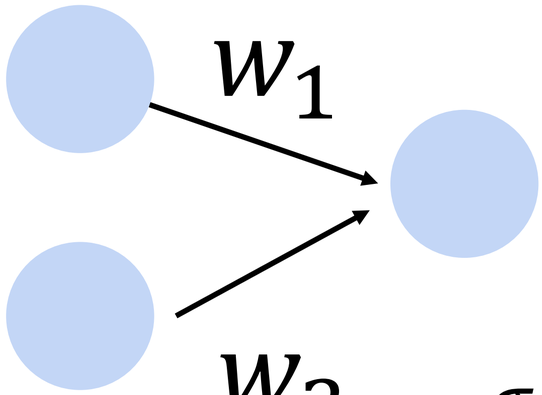
The perceptron can learn an OR function



Learning OR function using perceptron

The perceptron can learn an OR function





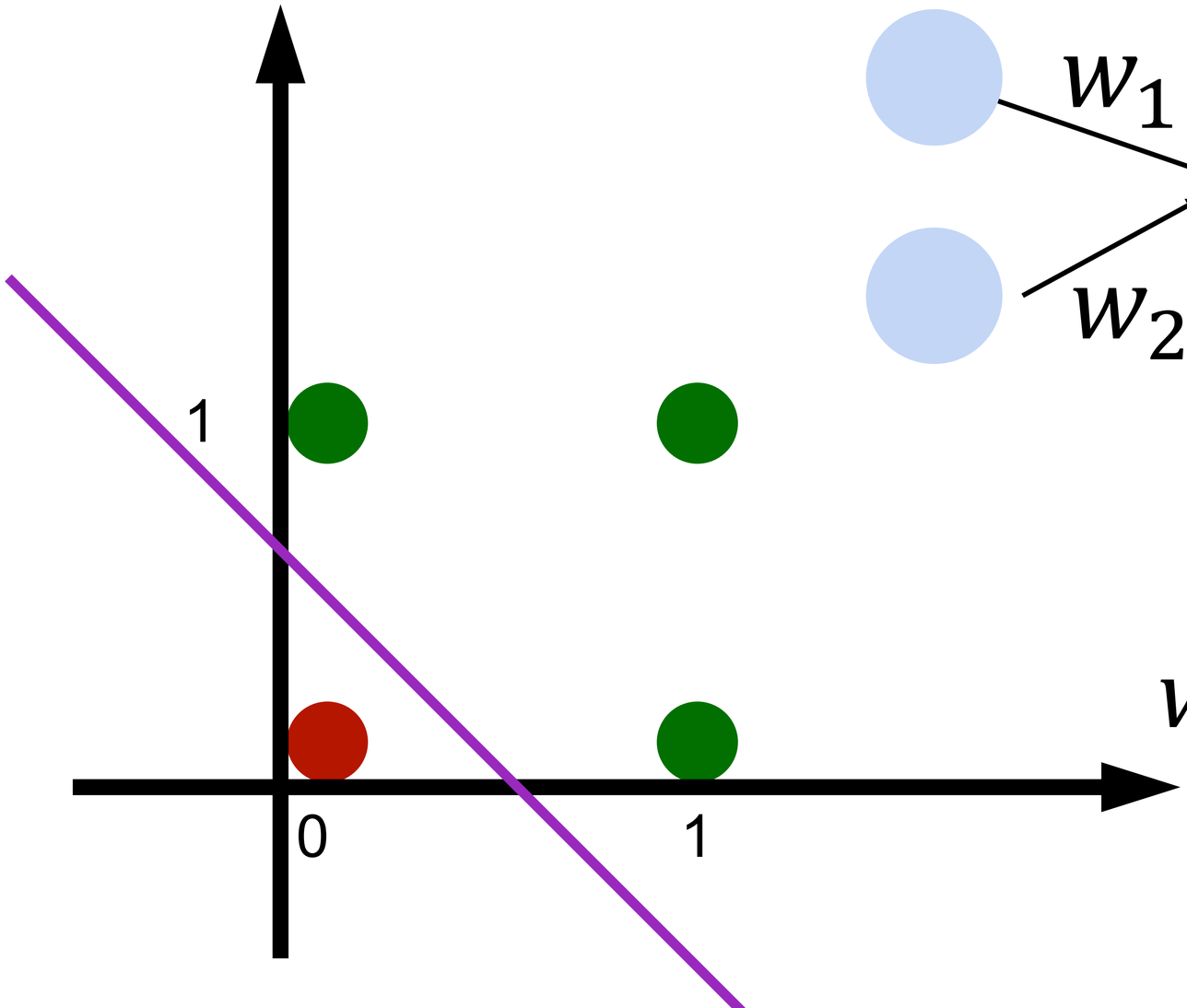
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

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

What's w and b?

Learning OR function using perceptron

The perceptron can learn an OR function



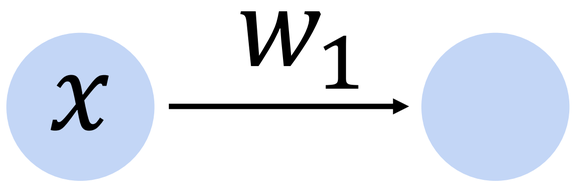
Output $\sigma(x_1 w_1 + x_2 w_2 + b)$

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

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

Learning NOT function using perceptron

The perceptron can learn NOT function (single input)

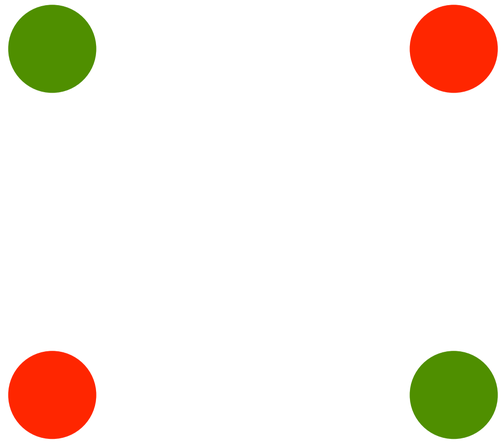


$$\sigma(x) = \begin{cases} 1 & if\ x > 0 \\ 0 & otherwise \end{cases}$$

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

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 present 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

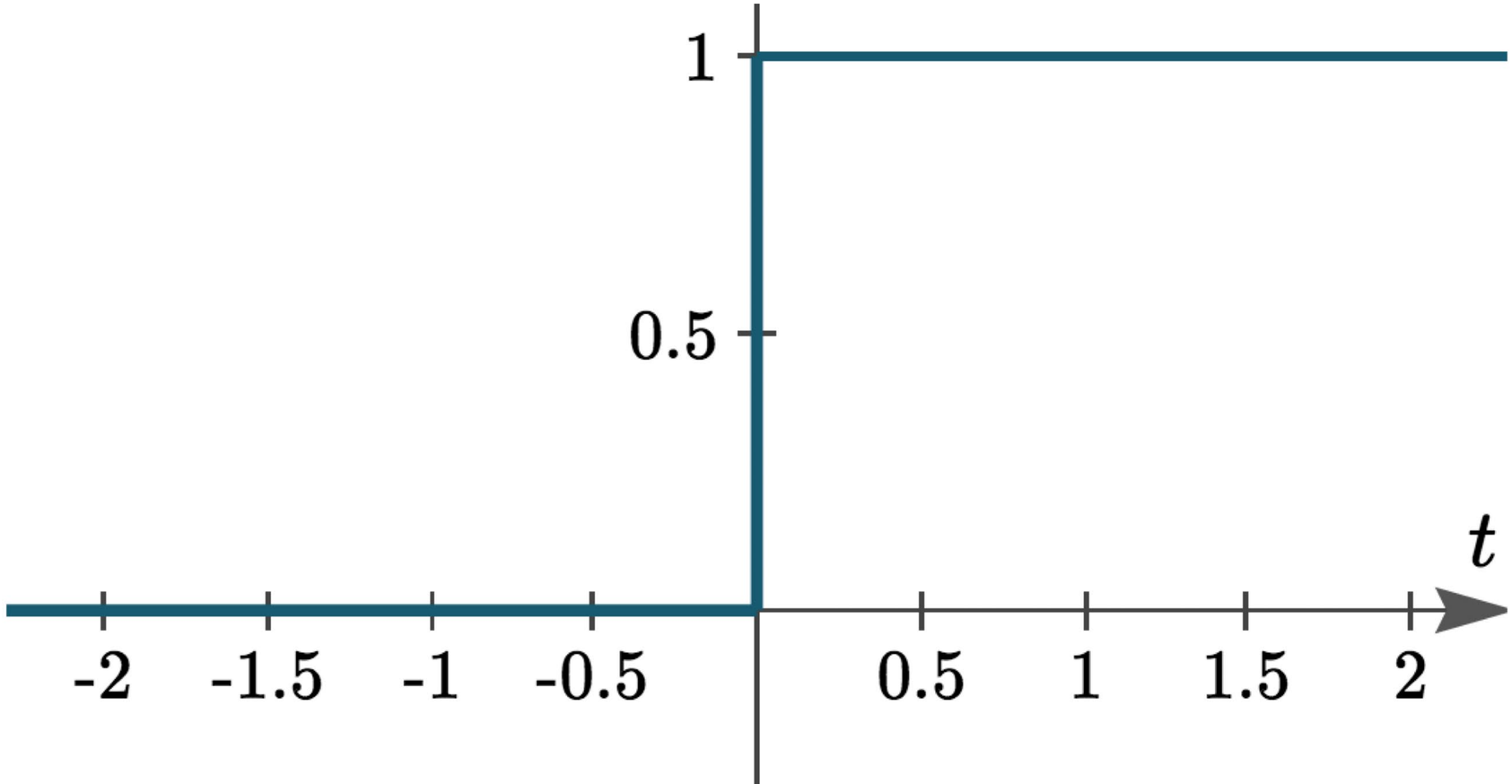
D. Both AND and OR function

Activation Functions

Step Function activation

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

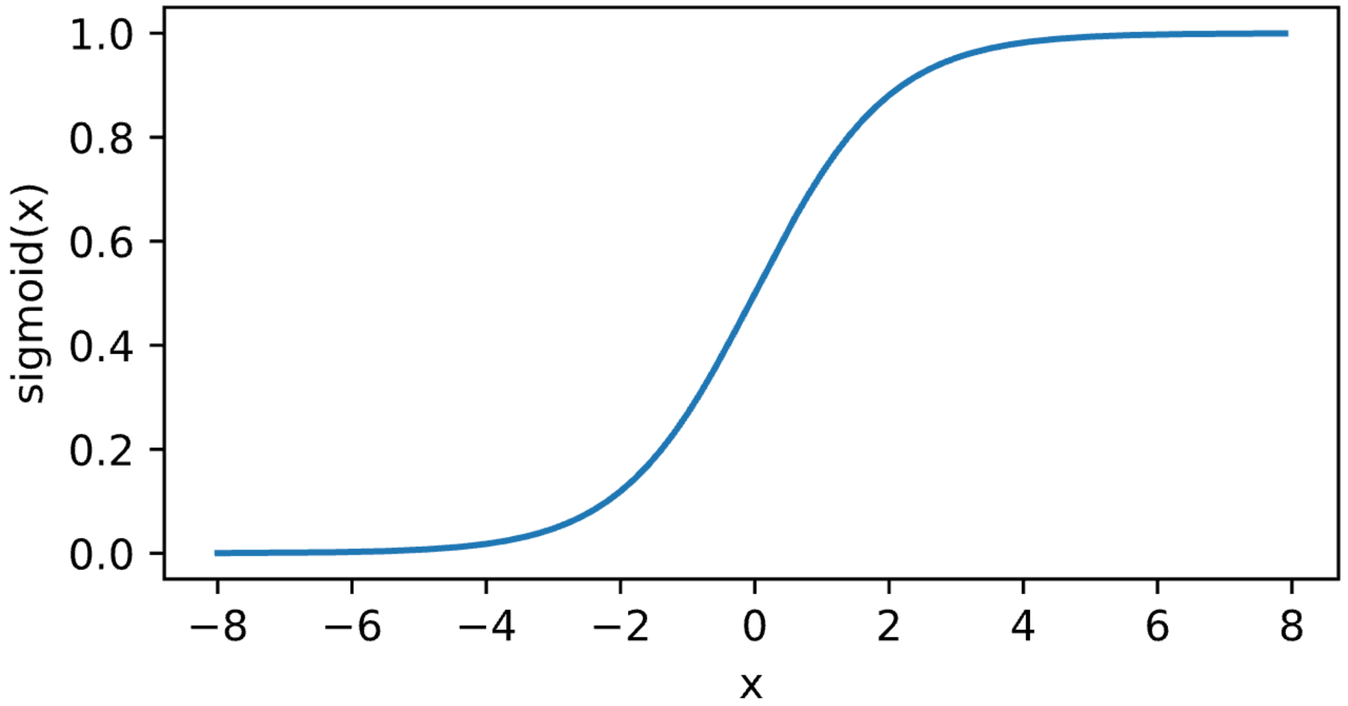
$$\sigma(x) = \begin{cases} 1 & if\ x > 0 \\ 0 & otherwise \end{cases}$$



Sigmoid/Logistic Activation

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

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

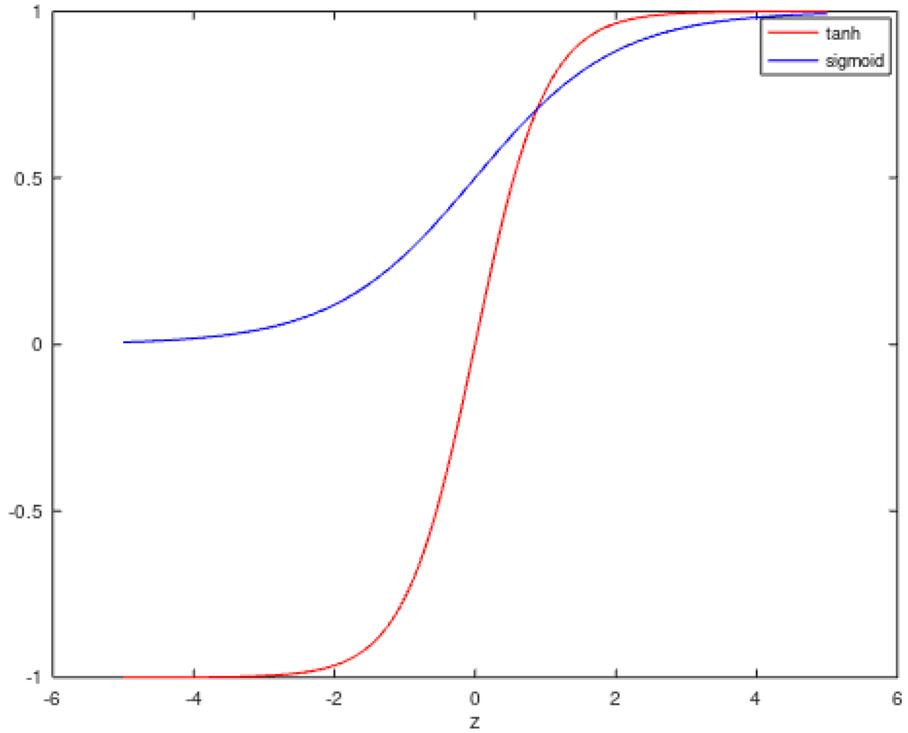


Tanh Activation

Map inputs into (-1, 1)

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

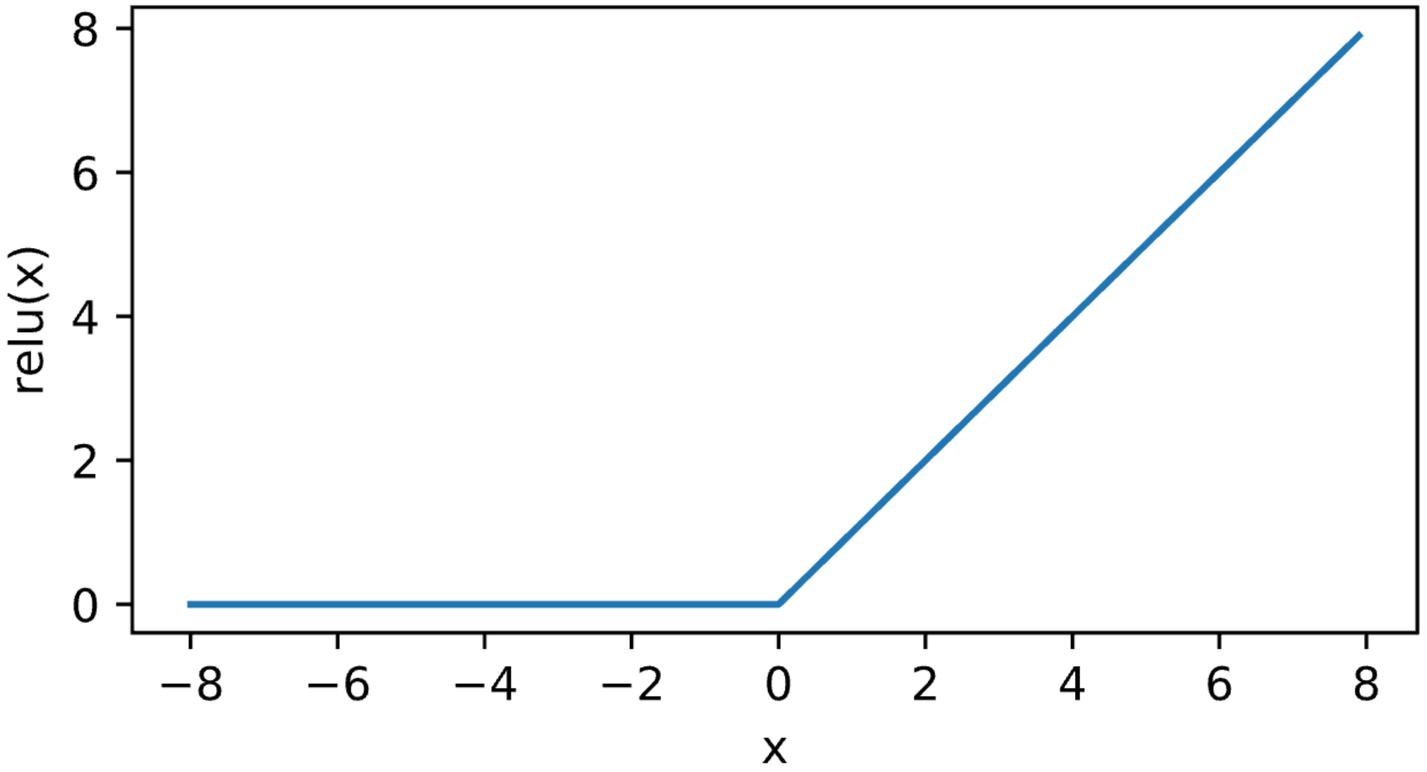
$$tanh(z) = 2sigmoid(2z) - 1$$



ReLU Activation

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

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



Quiz Break

Which one of the following is valid activation function

a)Step function
b) Sigmoid function
C) ReLU function
D) all of above

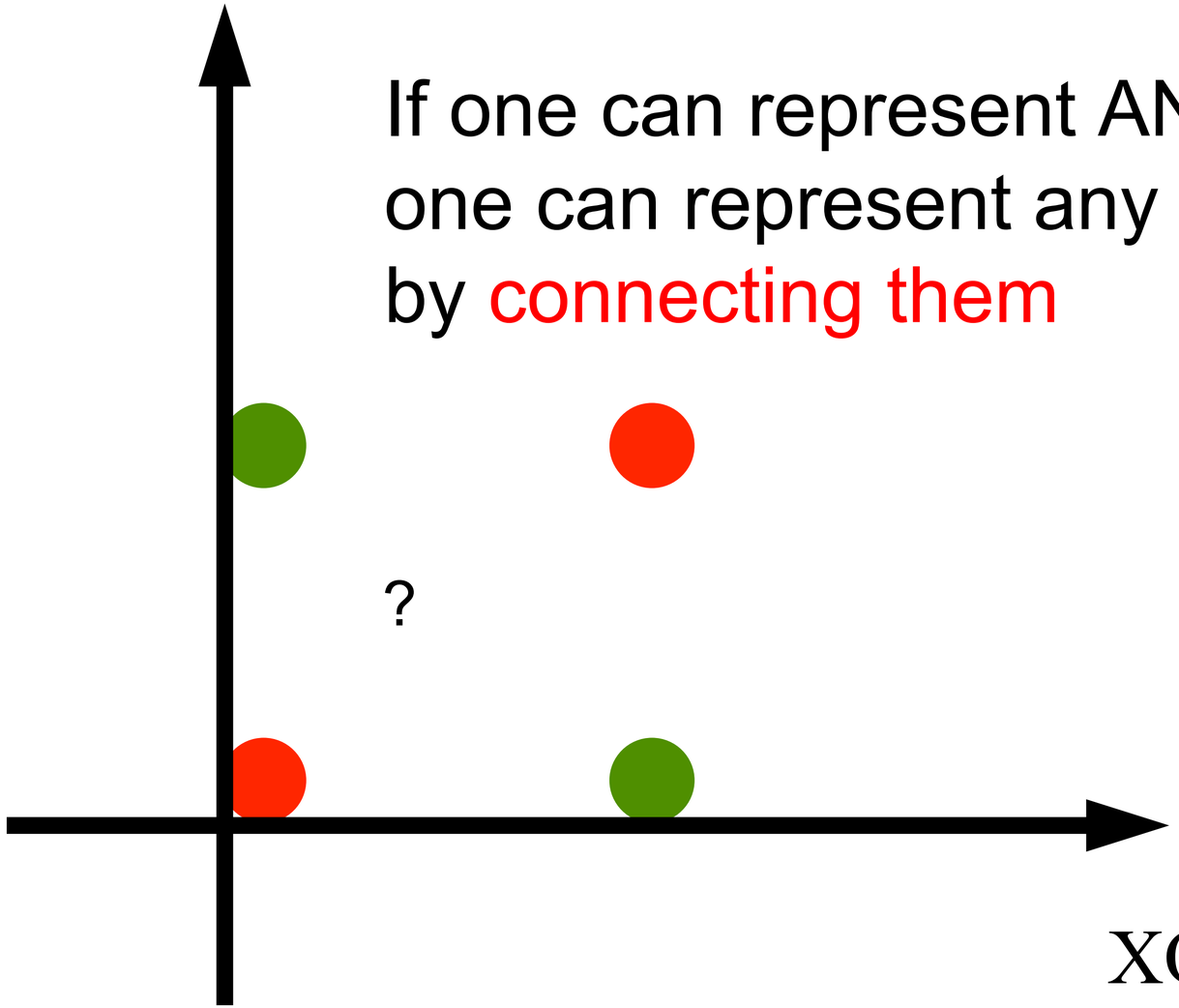
Quiz Break

Which one of the following is valid activation function

a)Step function
b) Sigmoid function
C) ReLU function
D) all of above

The limited power of a single neuron

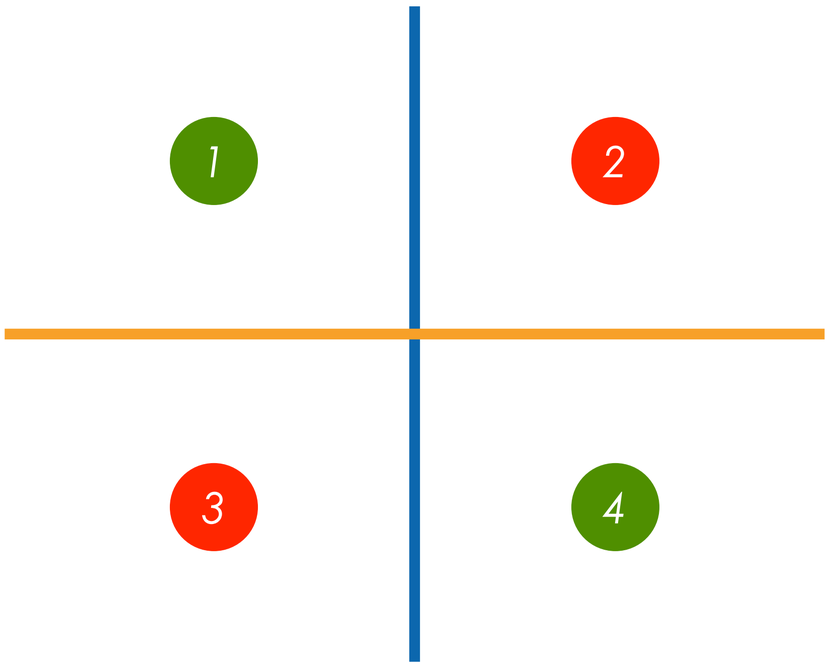
XOR problem

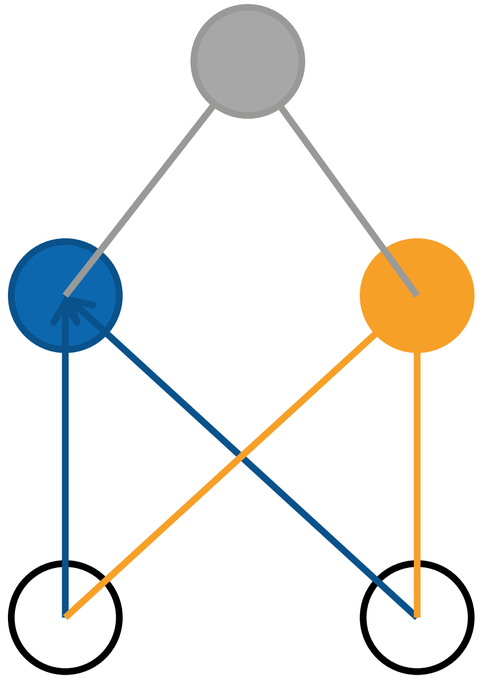


If one can represent AND, OR, NOT,
one can represent any logic circuit (including XOR),
by connecting them

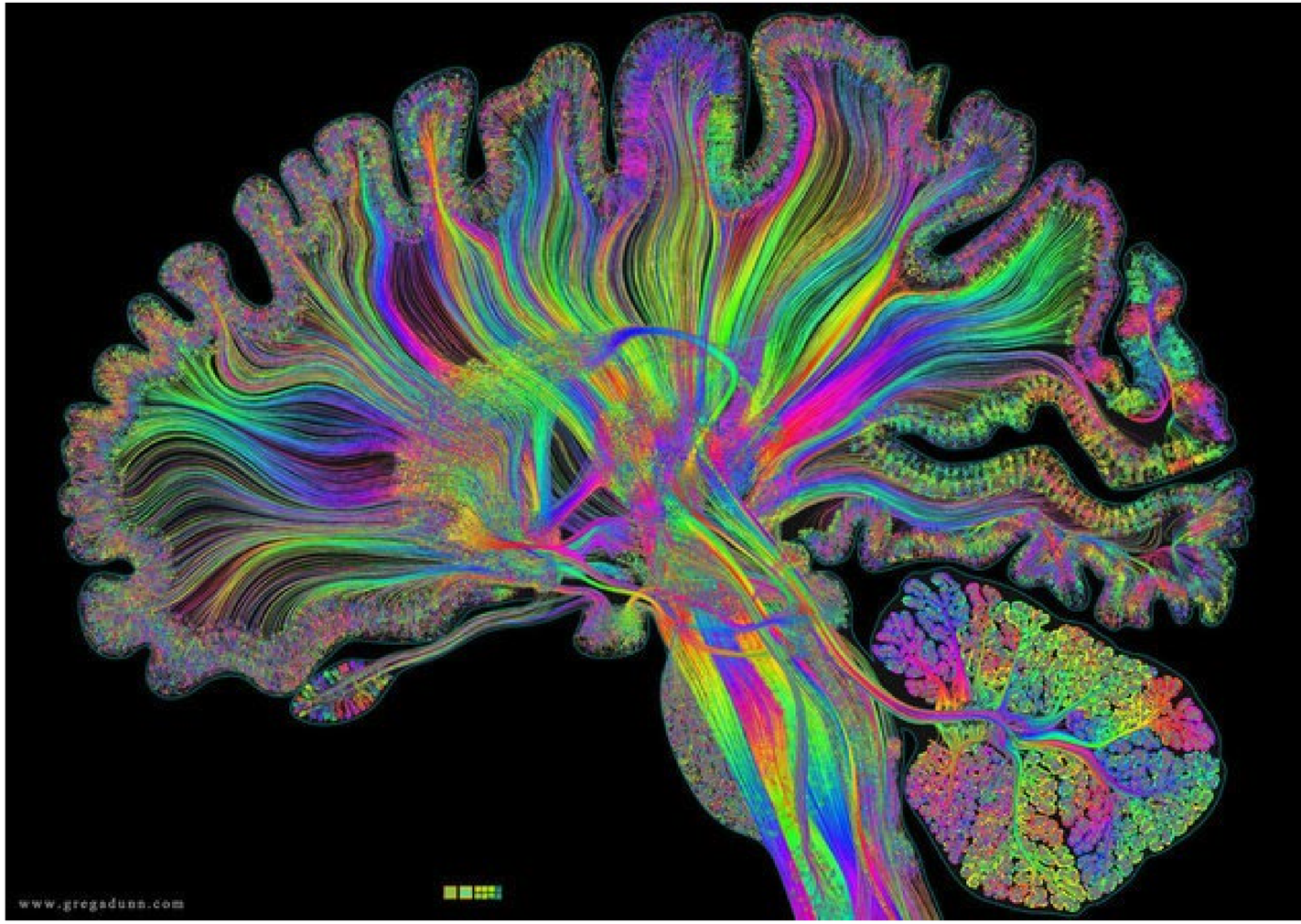
$$XOR(x1,x2\ ) = (x1 \wedge \neg x2) \vee (\neg x1 \wedge x2)$$

Learning XOR





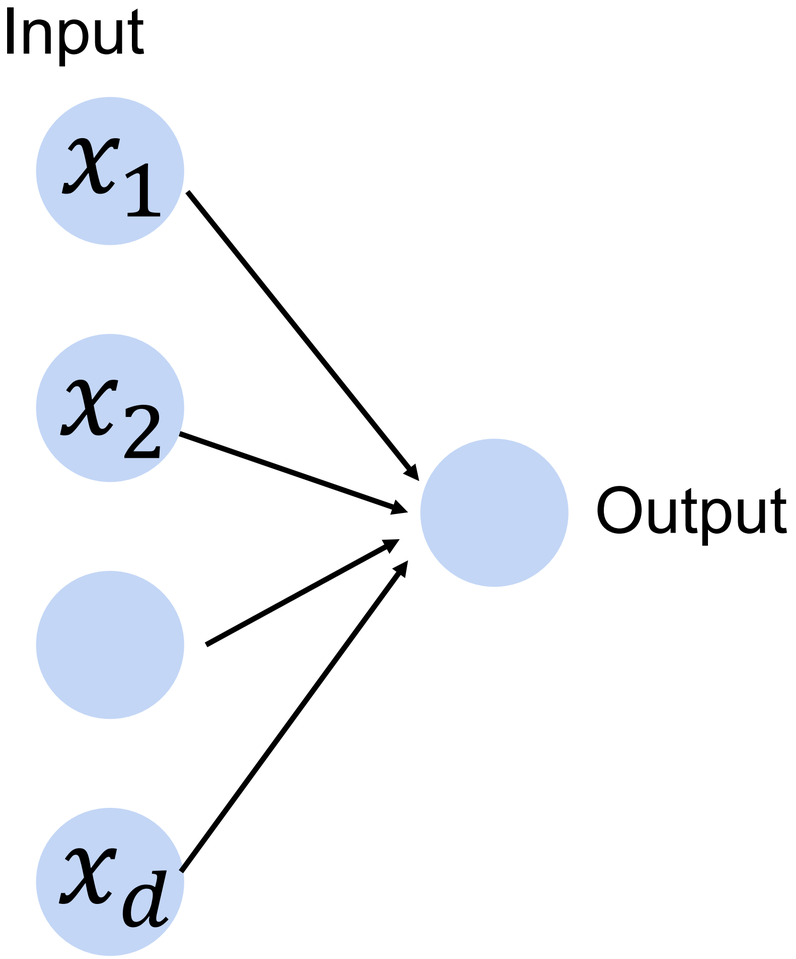
Multilayer Perceptron



The perceptron has one layer of weights

Each input node receives one scalar feature (e.g., one pixel)

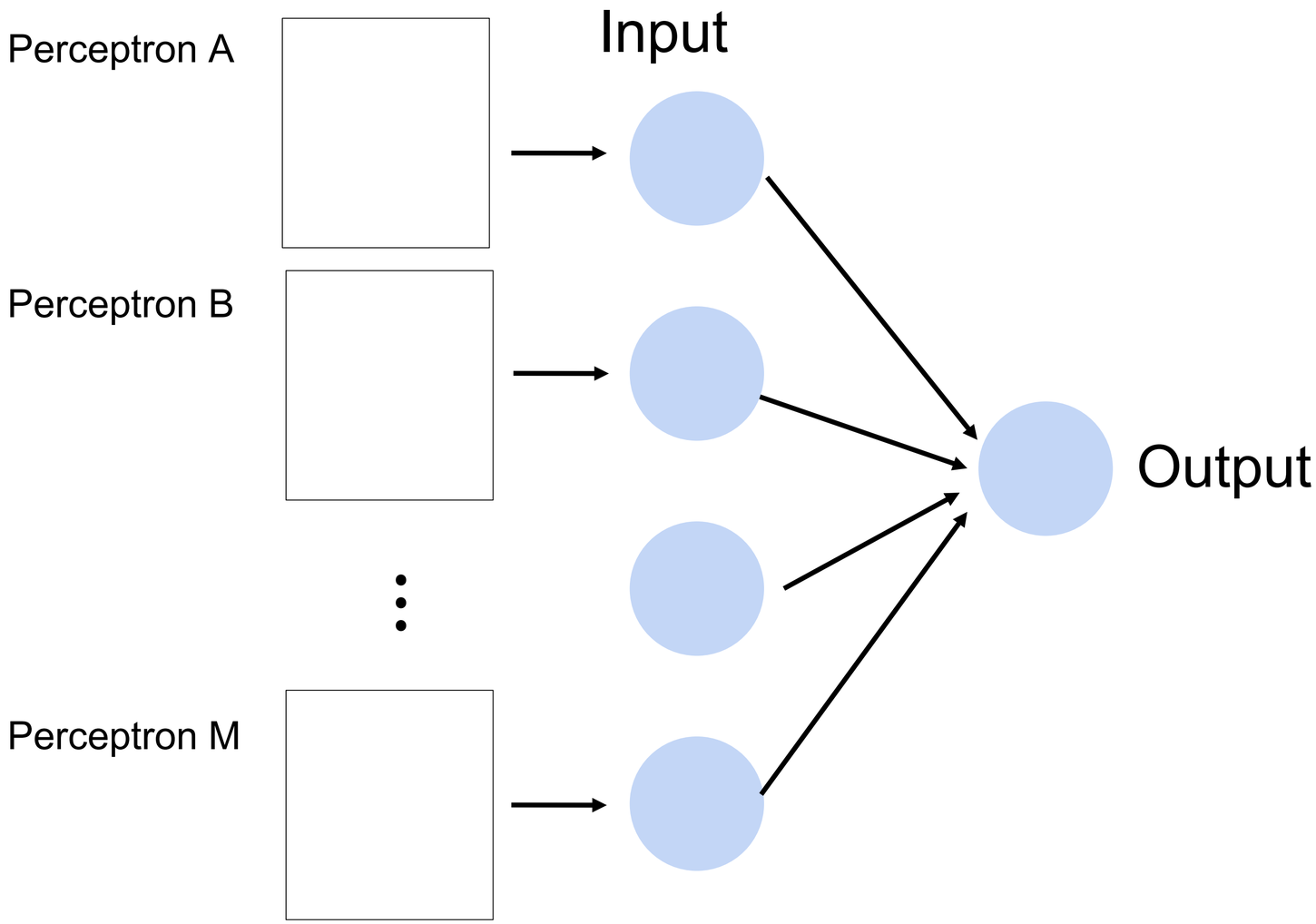
Cats vs. dogs?



Beyond one layer

Big Idea: take our inputs from other perceptrons!

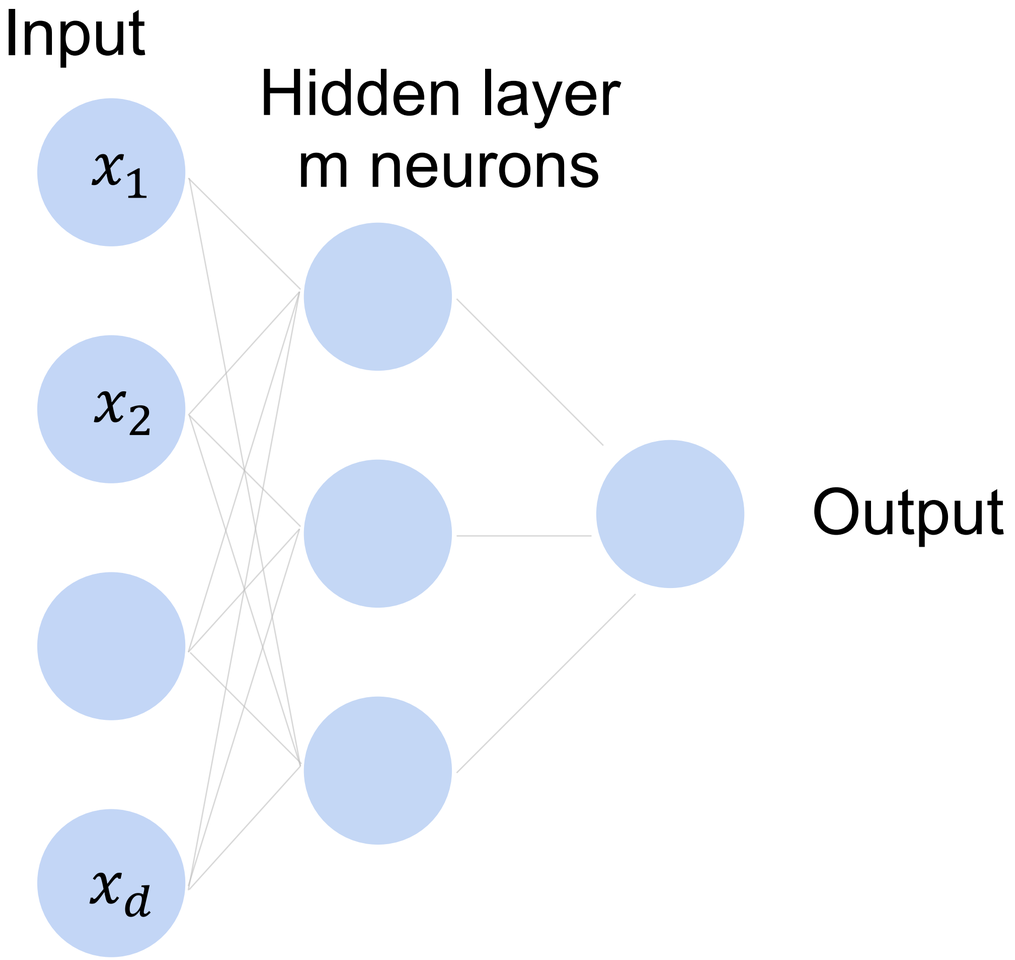
Cats vs. dogs?



Multilayer Perceptron with One "Hidden" Layer

How to classify

Cats vs. dogs?



Single Hidden Layer

Input $x \in \mathbb{R}^d$

Hidden $W_h \in \mathbb{R}^{m \times d}, b_h \in \mathbb{R}^m$

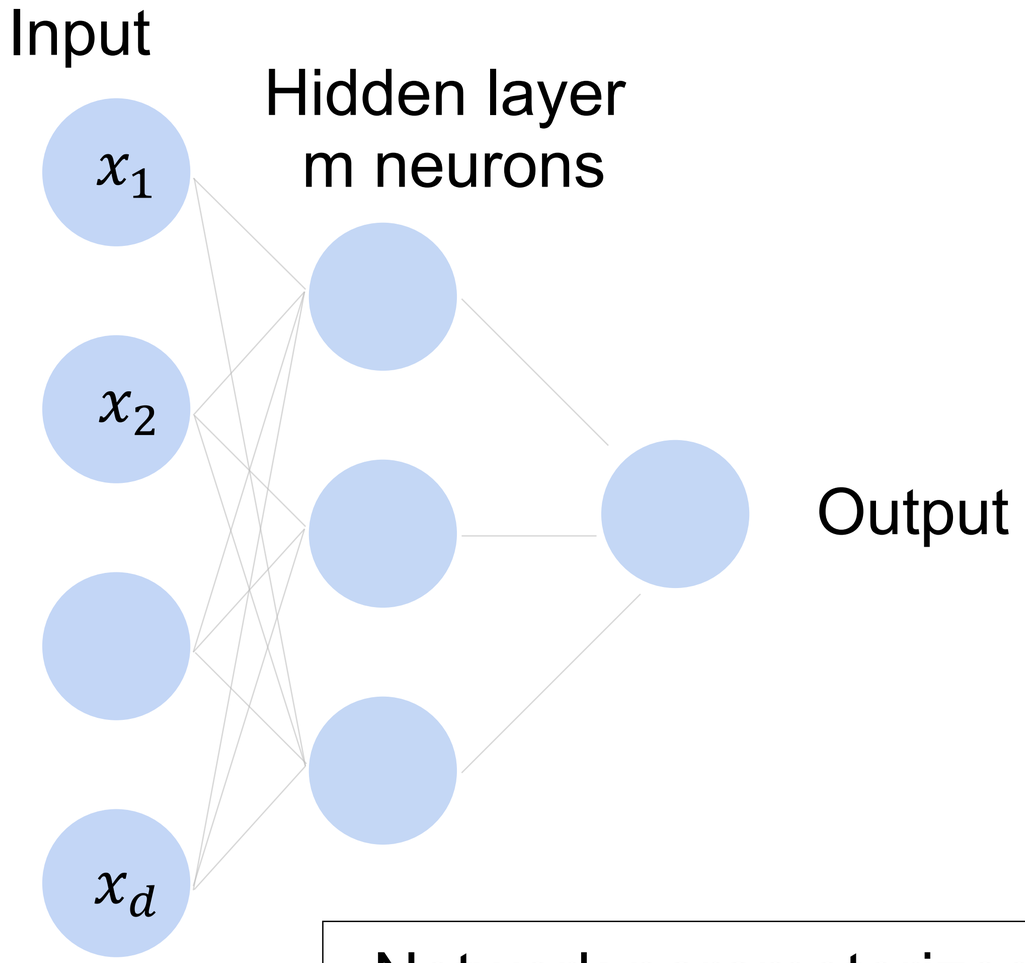
Intermediate output

$$h = \sigma(W_h x + b_h)$$

σ is an element-wise activation function

- Final output/final perceptron

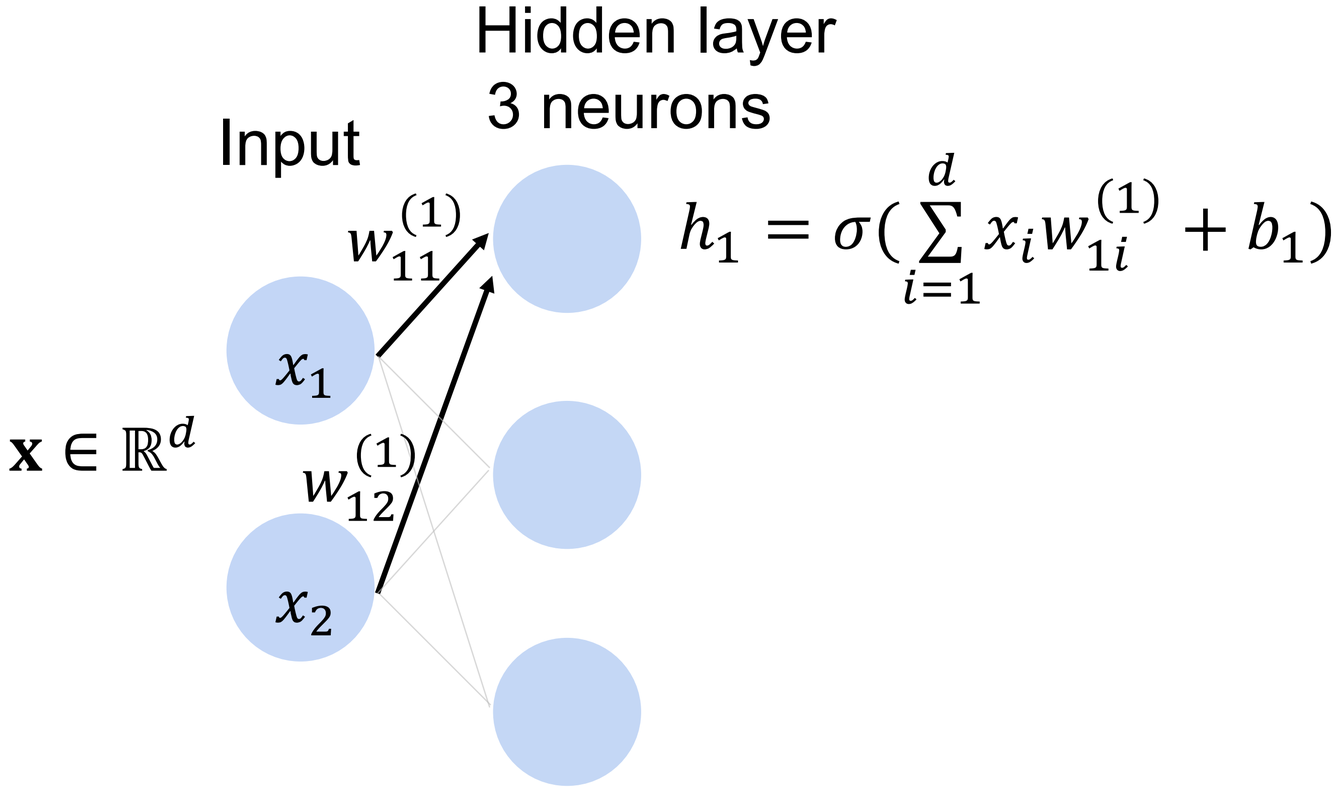
$$\sigma(\langle w_f, h \rangle + b_f)$$



Network parameterized by W_h, b_h, w_f, b_f

Multi-layer perceptron: Example

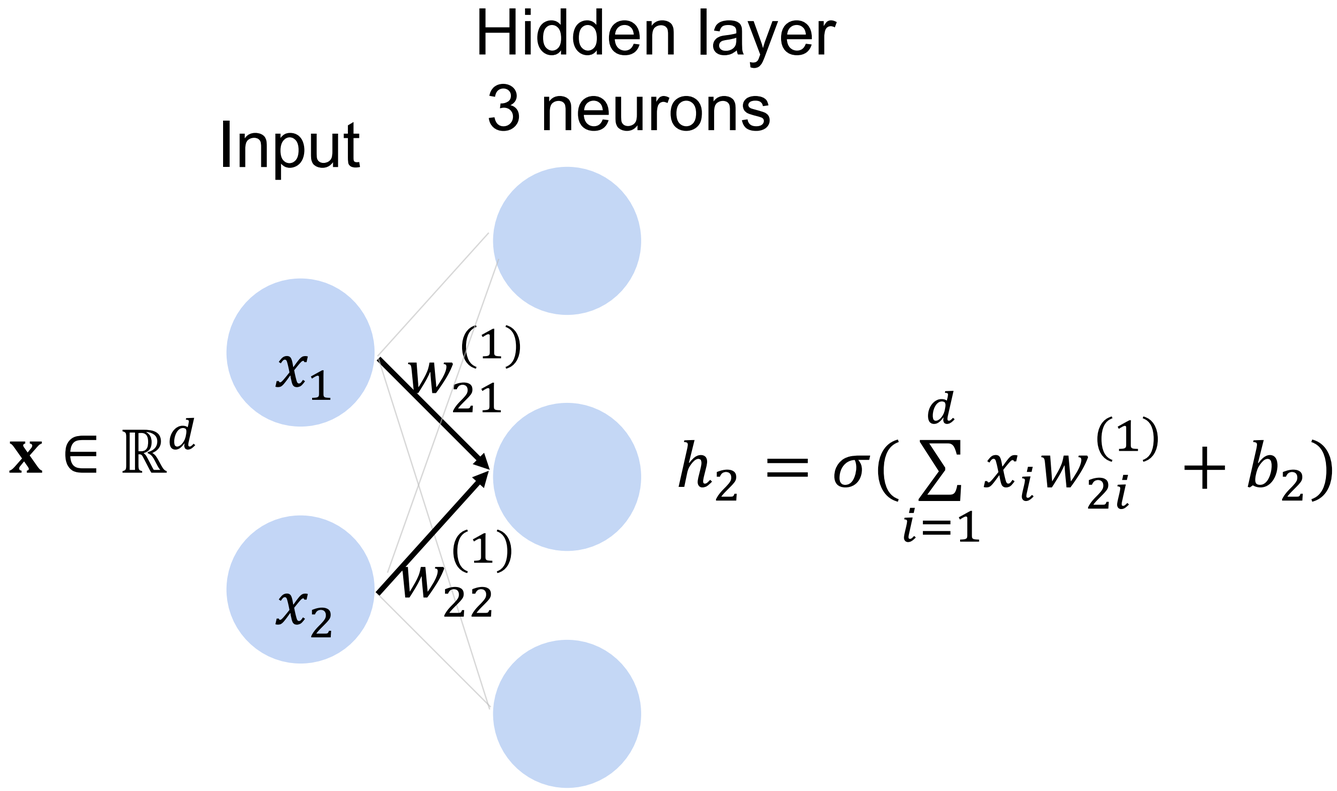
- Standard way to connect Perceptrons
- Example: 1 hidden layer, 1 output layer, depth = 2



$$h_1 = \sigma\left(\sum_{i=1}^{d} x_i w_{1i}^{(1)} + b_1\right)$$

Multi-layer perceptron: Example

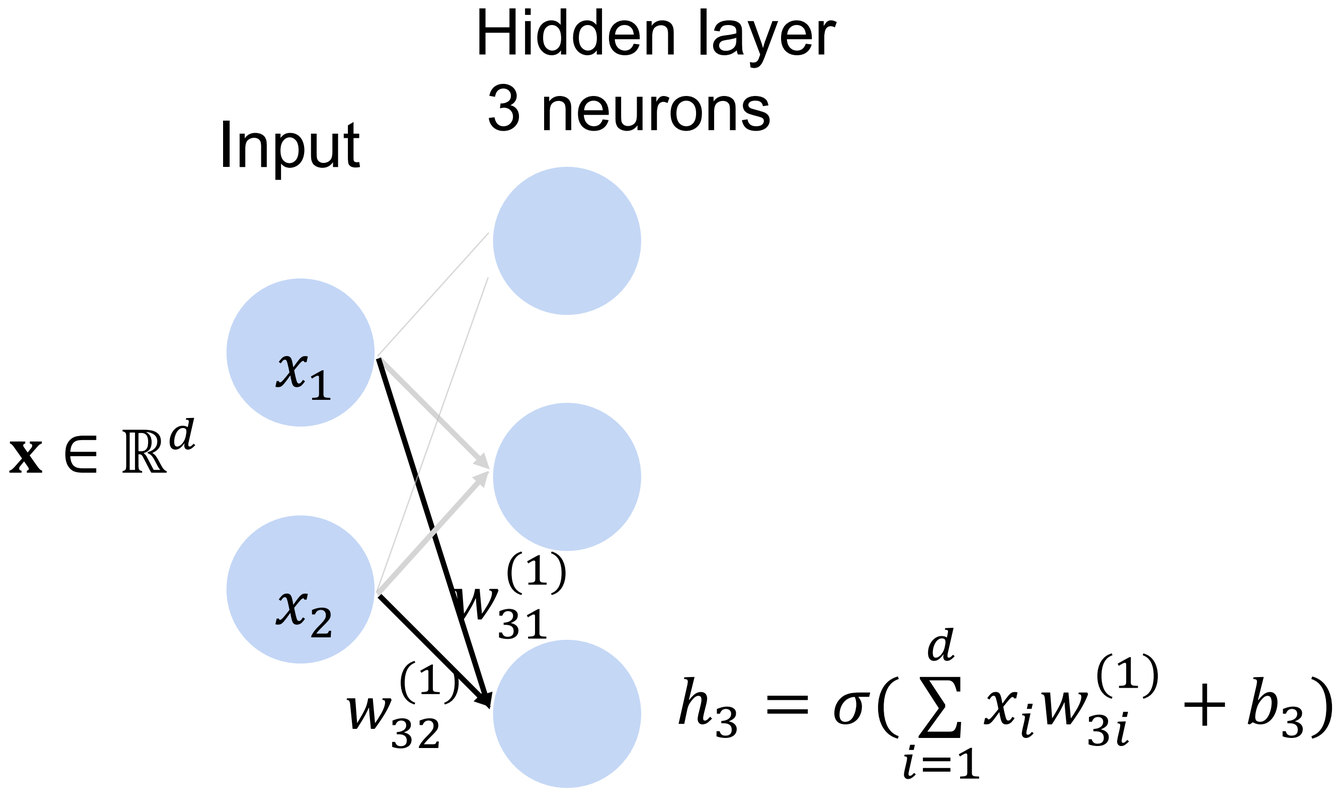
- Standard way to connect Perceptrons
- Example: 1 hidden layer, 1 output layer, depth = 2



$$h_2 = \sigma\left(\sum_{i=1}^{d} x_i w_{2i}^{(1)} + b_2\right)$$

Multi-layer perceptron: Example

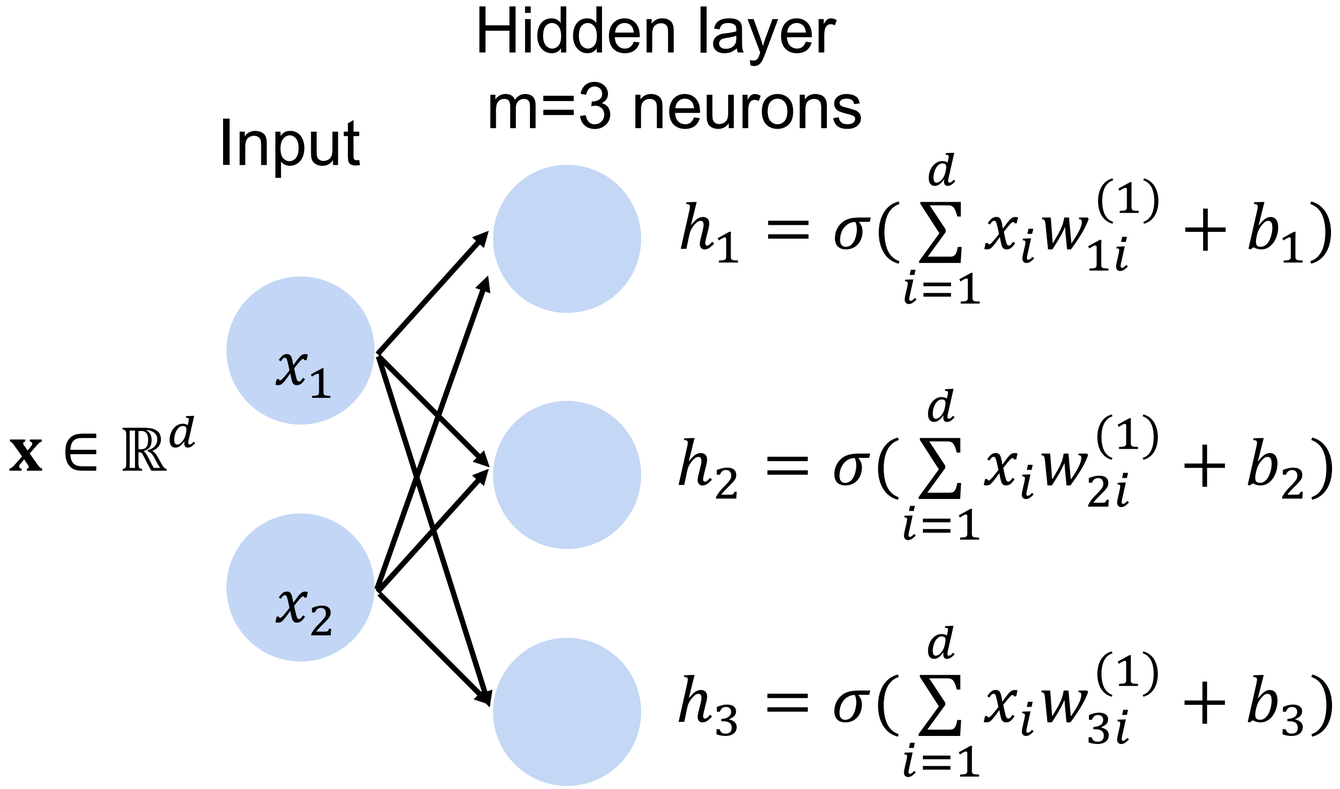
- Standard way to connect Perceptrons
- Example: 1 hidden layer, 1 output layer, depth = 2



$$h_3 = \sigma(\sum_{i=1}^{d} x_i w_{3i}^{(1)} + b_3)$$

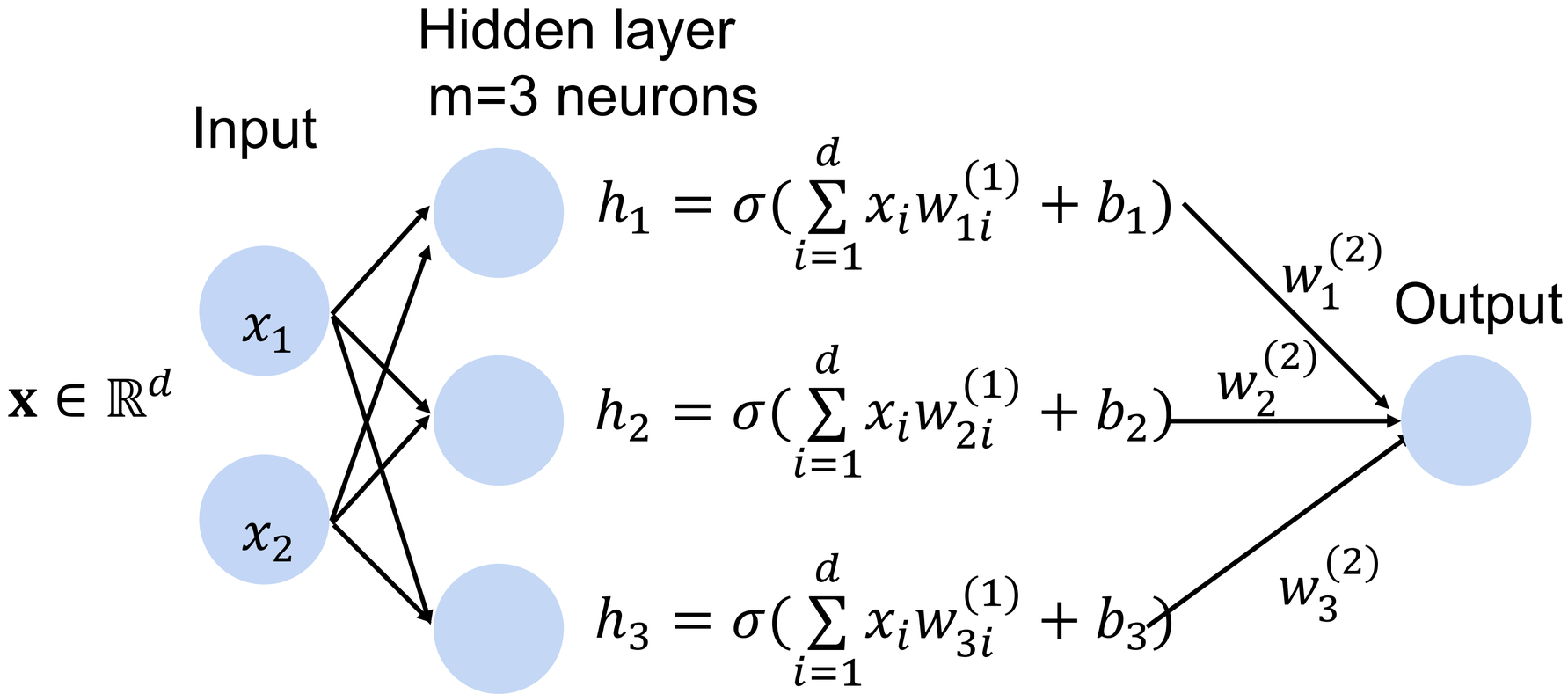
Multi-layer perceptron: Example

- Standard way to connect Perceptrons
- Example: 1 hidden layer, 1 output layer, depth = 2



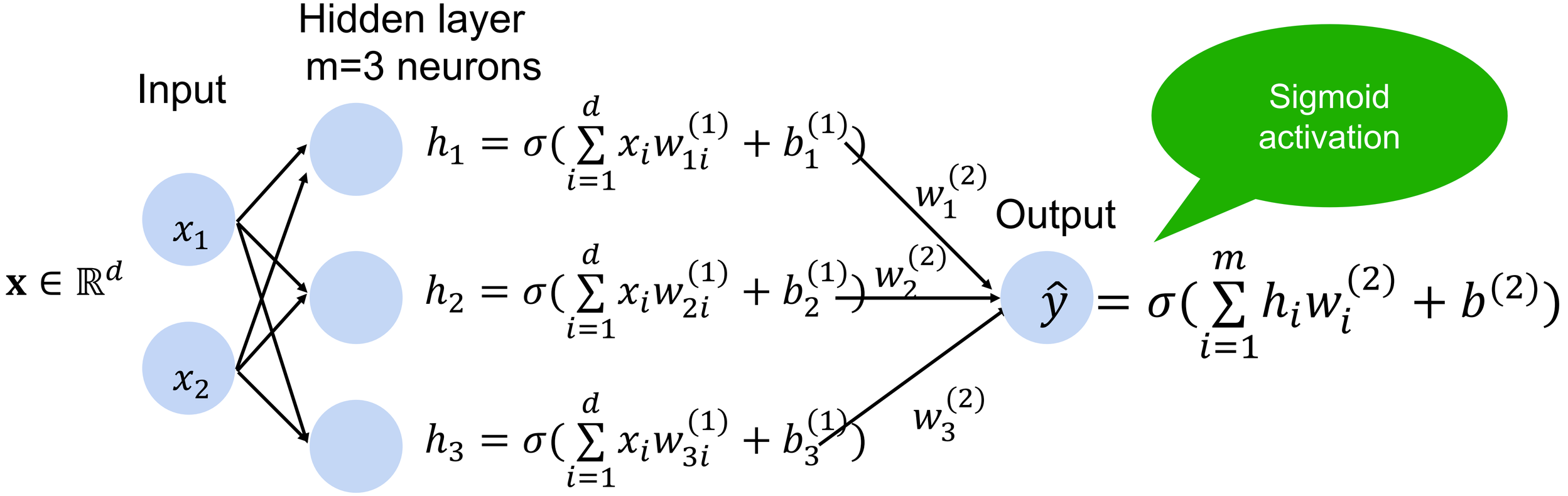
Multi-layer perceptron: Example

- Standard way to connect Perceptrons
- Example: 1 hidden layer, 1 output layer, depth = 2



Multi-layer perceptron: Example

- Standard way to connect Perceptrons
- Example: 1 hidden layer, 1 output layer, depth = 2

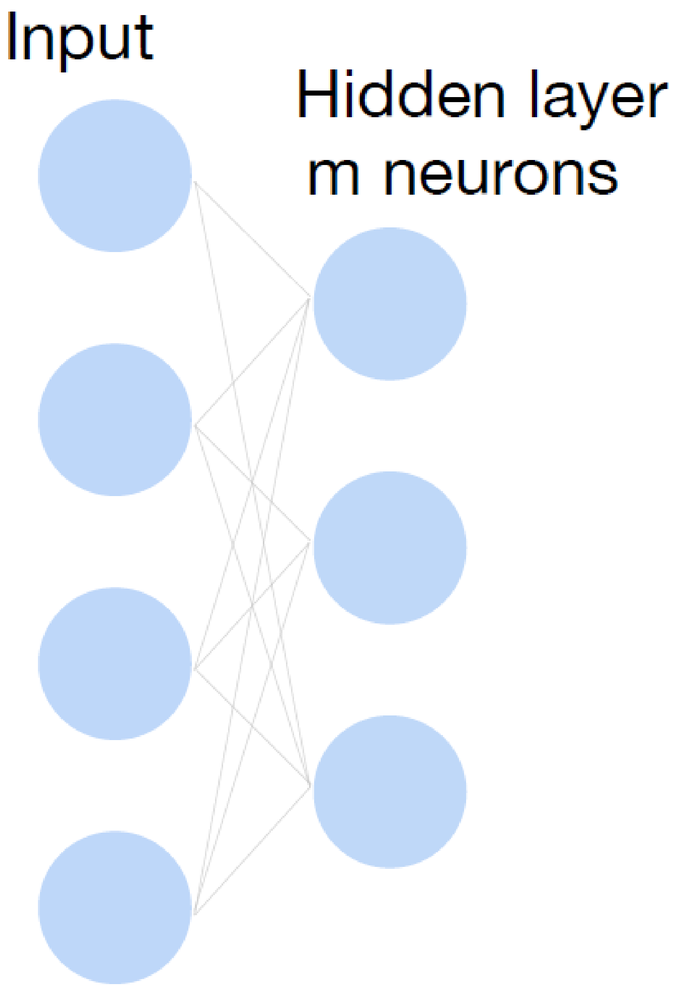


Multi-layer perceptron: Matrix Notation

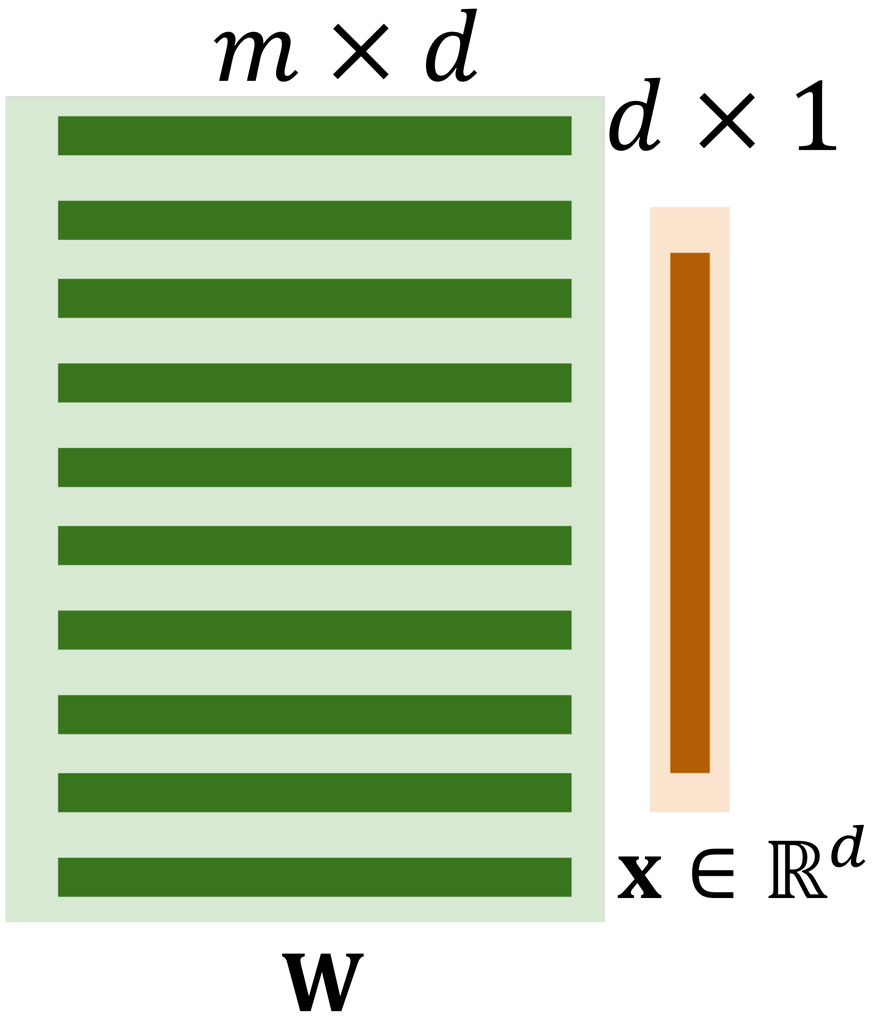
- Input $\mathbf{x} \in \mathbb{R}^d$
- Hidden $\mathbf{W}^{(1)} \in \mathbb{R}^{m \times d}, \mathbf{b}^{(1)} \in \mathbb{R}^m$
- Intermediate output

$$\mathbf{h} = \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$$

$$\mathbf{h} \in \mathbb{R}^m$$

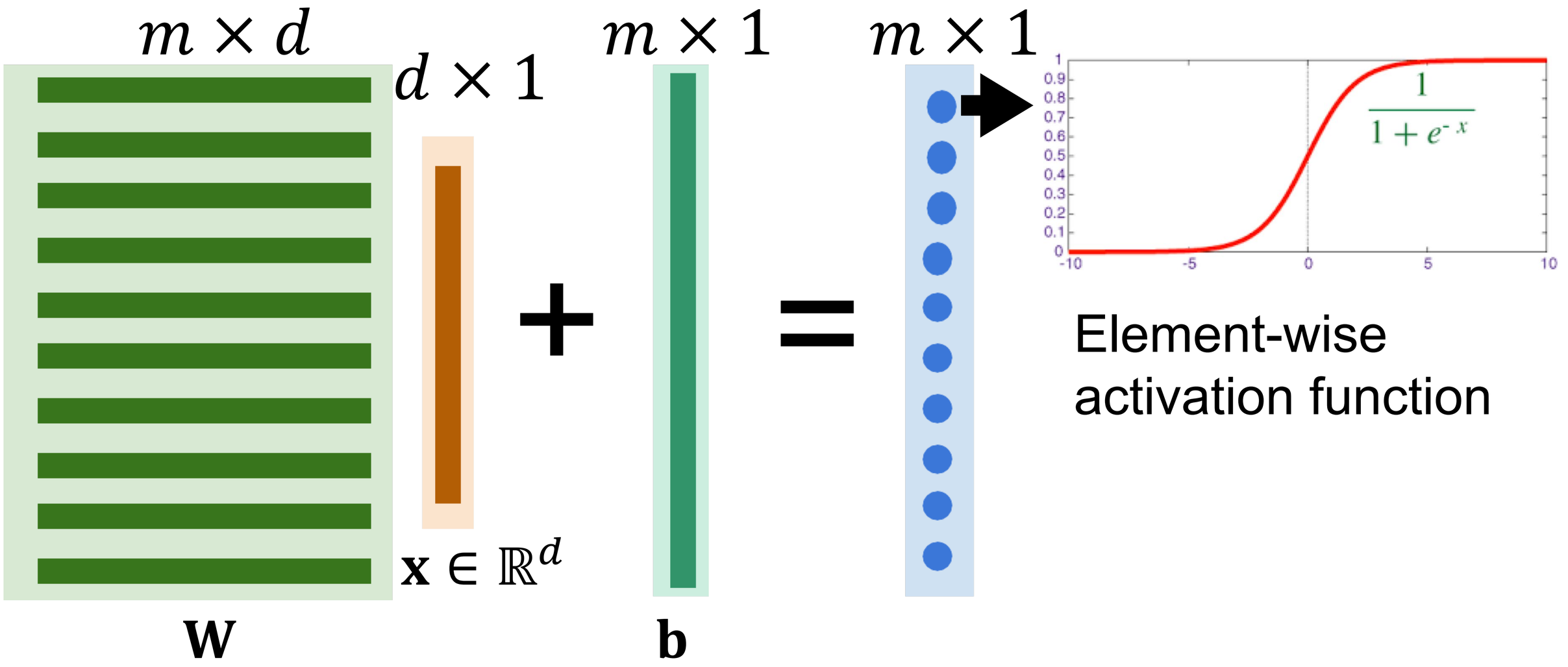


Multi-layer perceptron: Matrix Notation



Multi-layer perceptron: Matrix Notation

Key elements: linear operations + Nonlinear activations

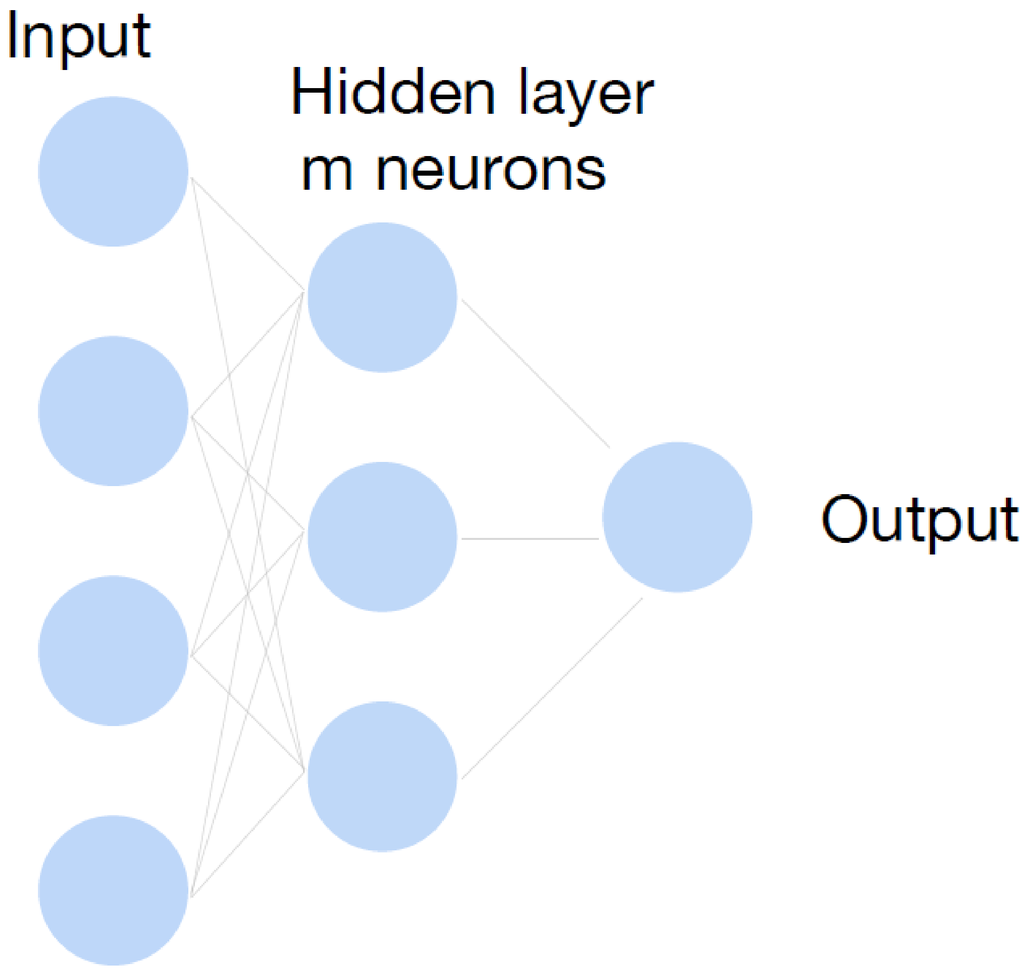


Multi-layer perceptron: Matrix Notation

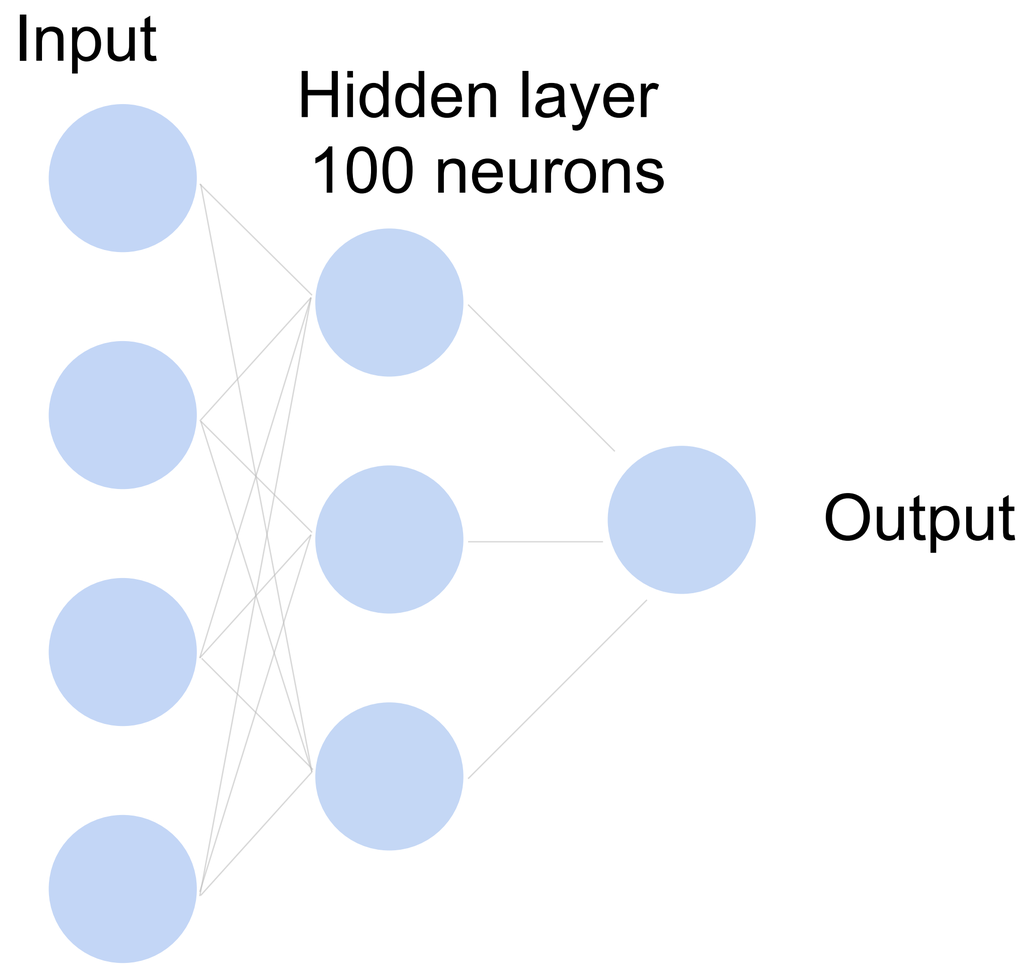
- Input $\mathbf{x} \in \mathbb{R}^d$
- Hidden $\mathbf{W}^{(1)} \in \mathbb{R}^{m \times d}, \mathbf{b}^{(1)} \in \mathbb{R}^m$
- Intermediate output

$$\mathbf{h} = \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$$

$$\hat{y} = \sigma(\mathbf{w}^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$$



Classify cats vs. dogs



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

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

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

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

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?

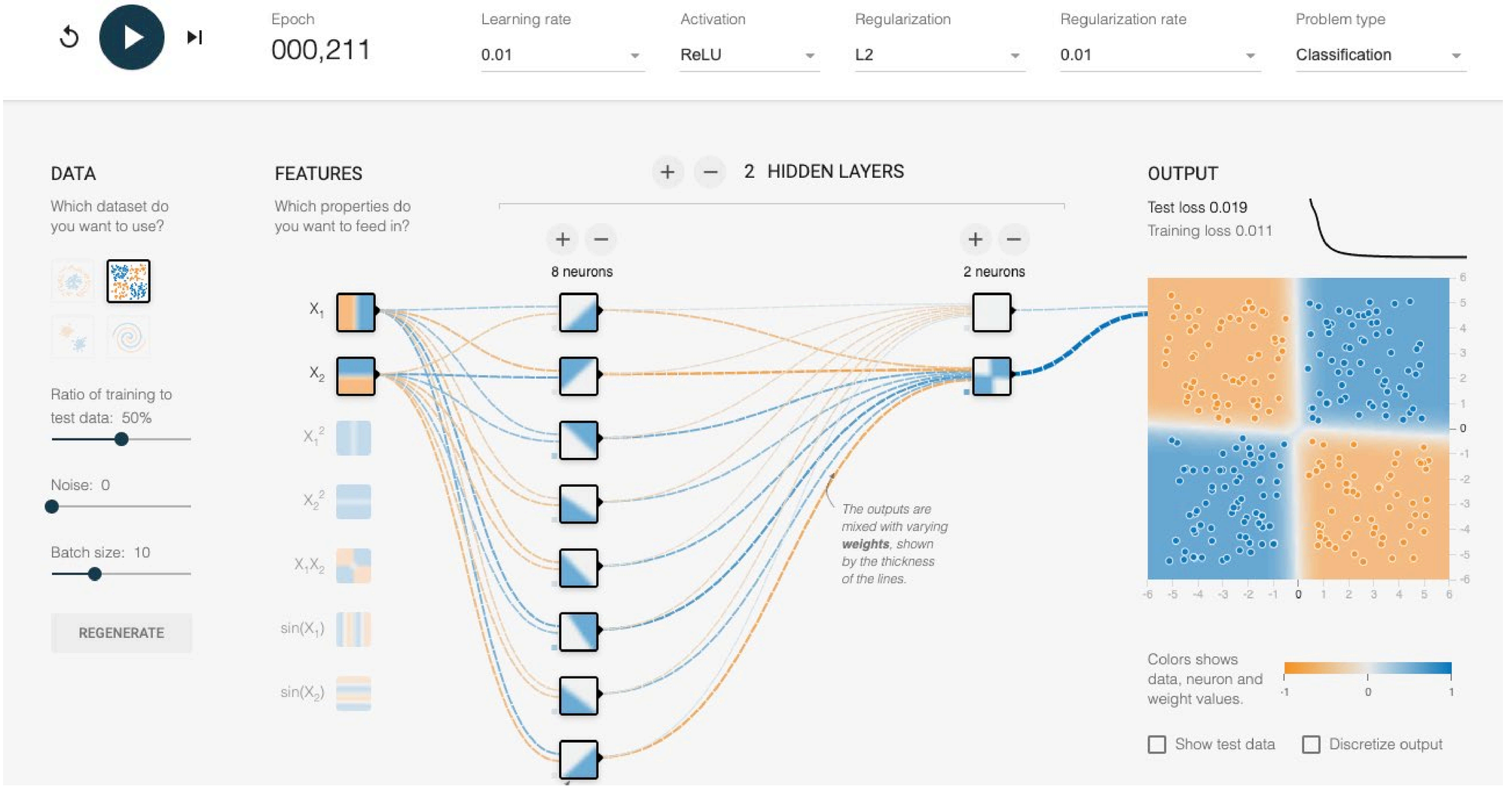
A $\mathrm{f}(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

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

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

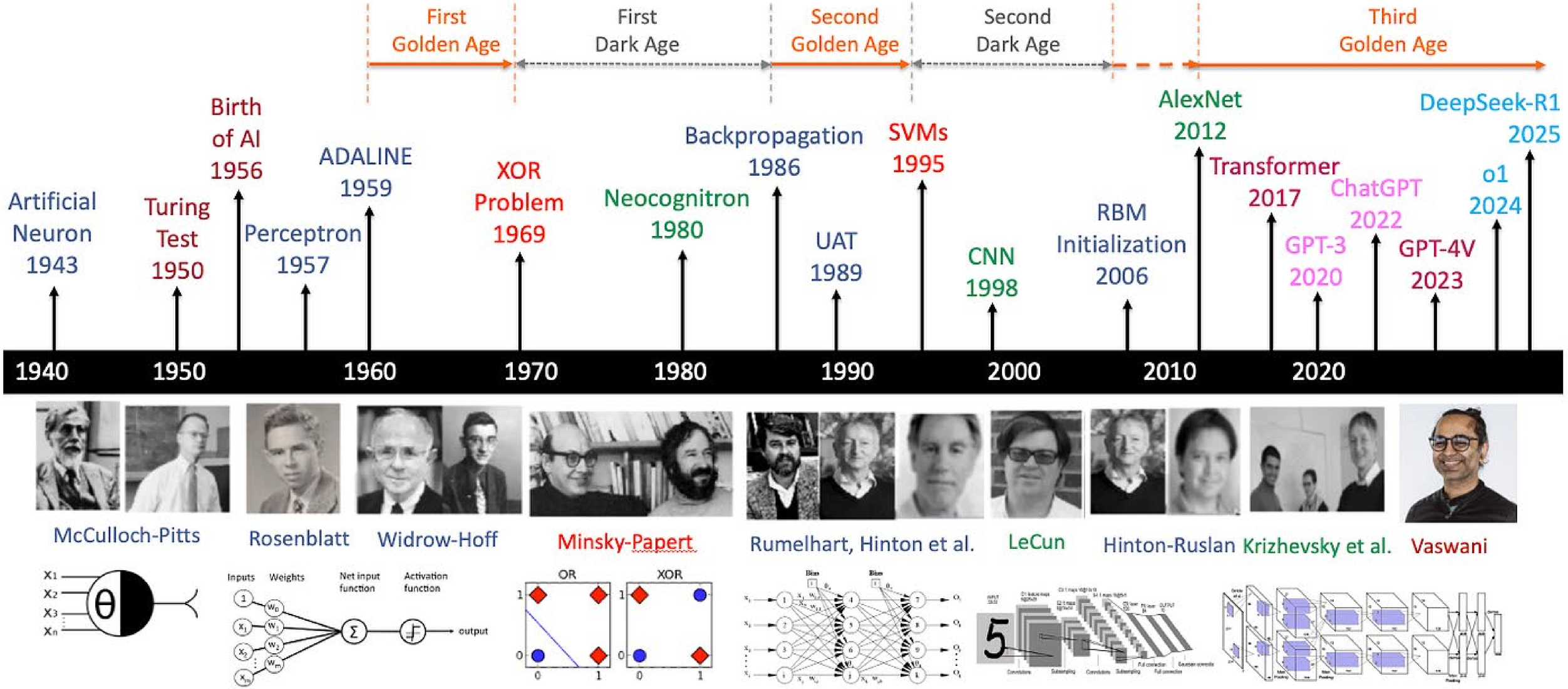
D $\mathrm{f}(x) = \begin{bmatrix} \exp(x_1 + x_2) \\ \exp(x_2) \end{bmatrix}$

Demo: Learning XOR using neural net



https://playground.tensorflow.org/

A Brief History of AI with Deep Learning



https://medium.com/@lmpo/a-brief-history-of-ai-with-deep-learning-26f7948bc87b

What we've learned today…

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

Suggested Readings

- Textbook: Artificial Intelligence: A Modern Approach (4th edition). Stuart Russell and Peter Norvig. Pearson, 2020.
  - Sections 19.6.4, 21.1