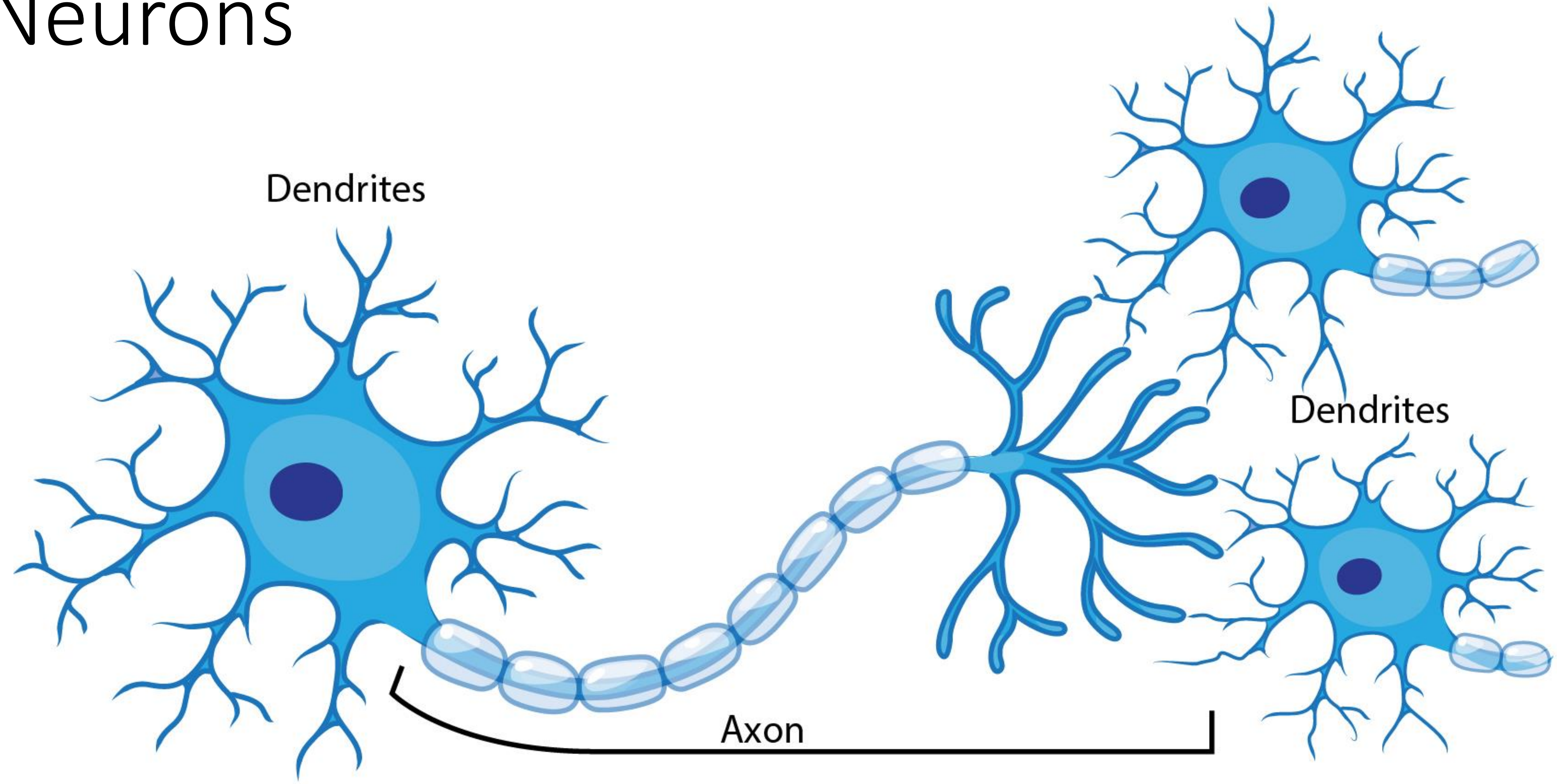
Neural Computation

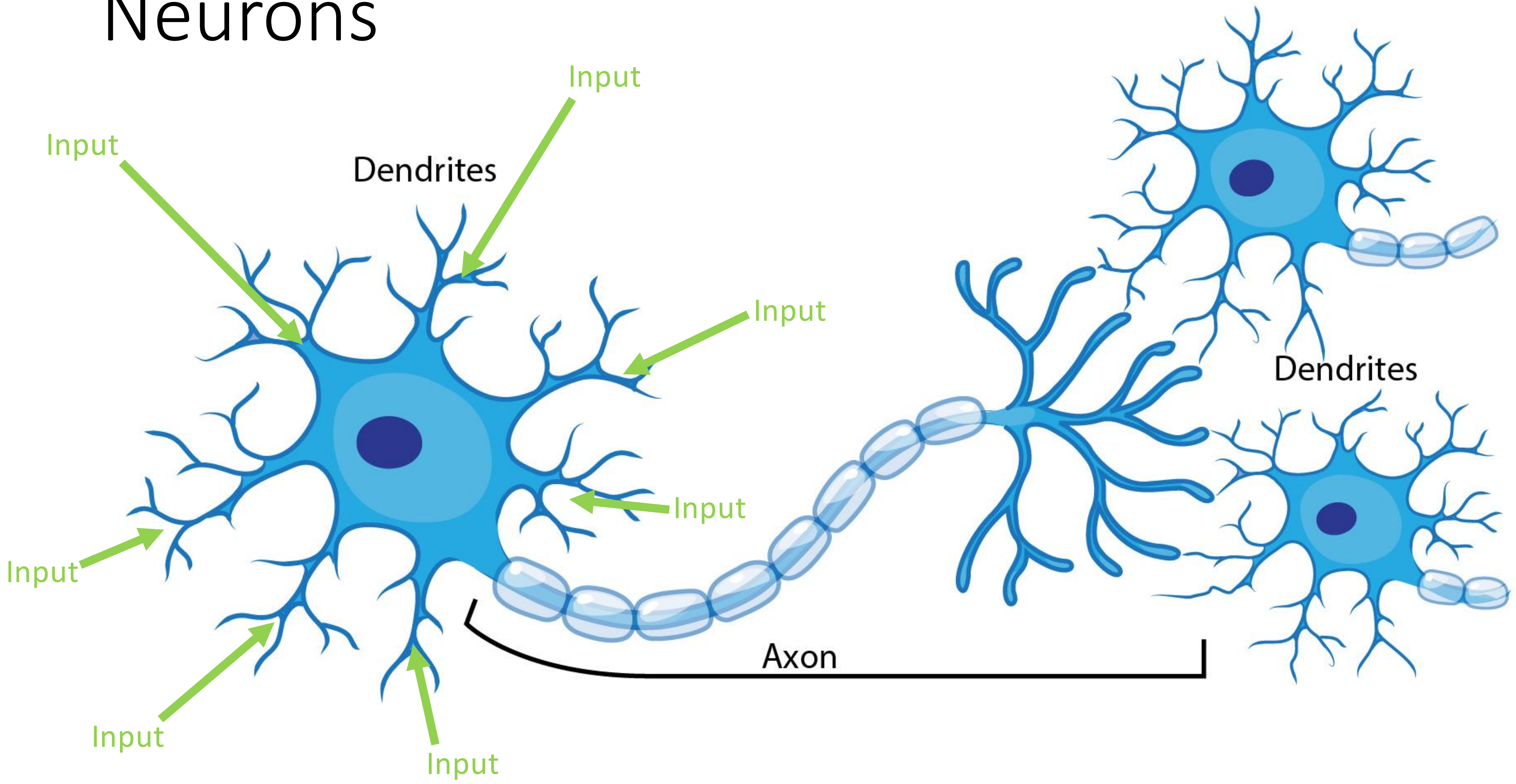
The Perceptron

Neurons



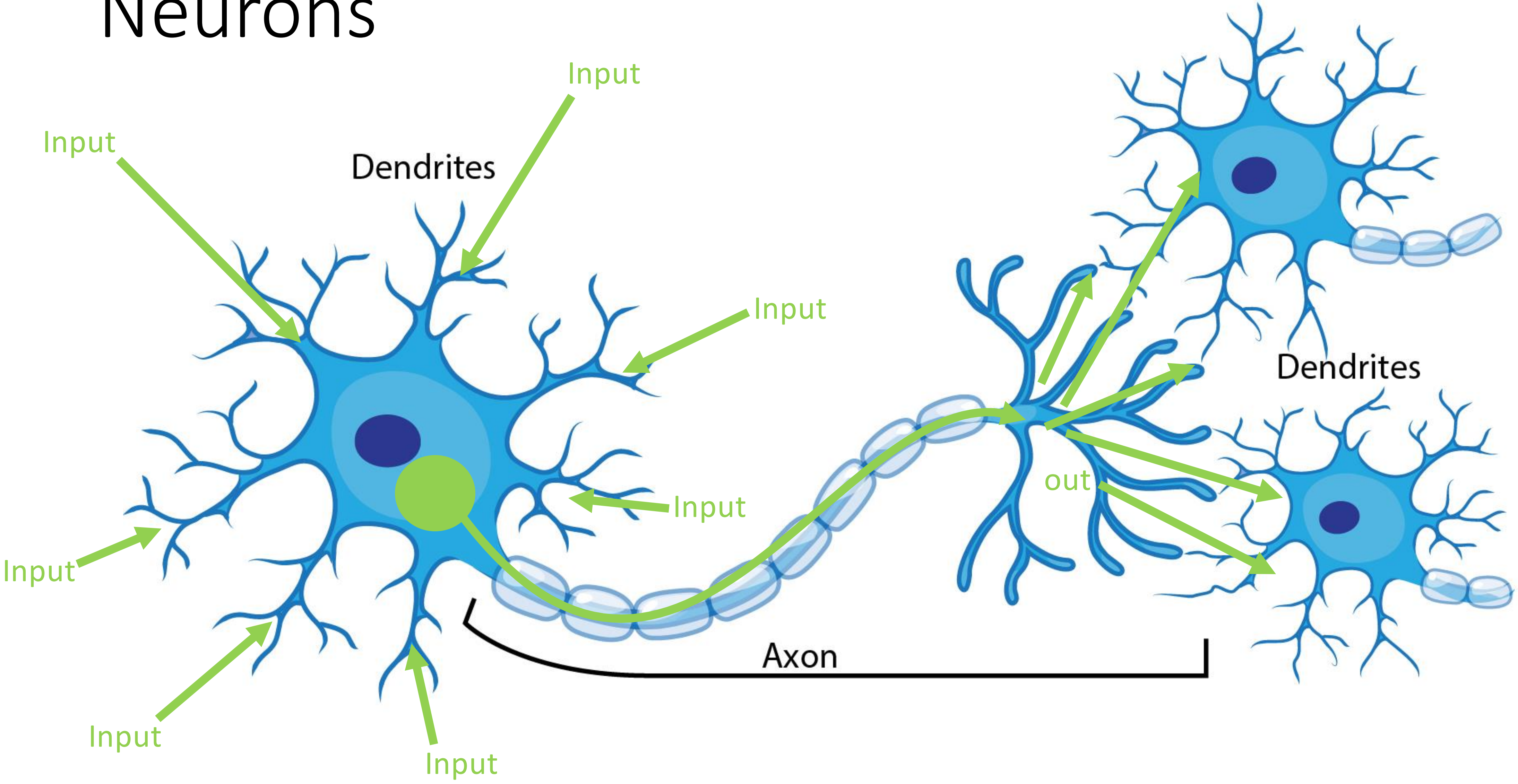
Devin K. Phillips. "Speed of the Human Brain". ASU - Ask A Biologist. 13 May, 2015.
https://askabiologist.asu.edu/plosable/speed-human-brain

Neurons



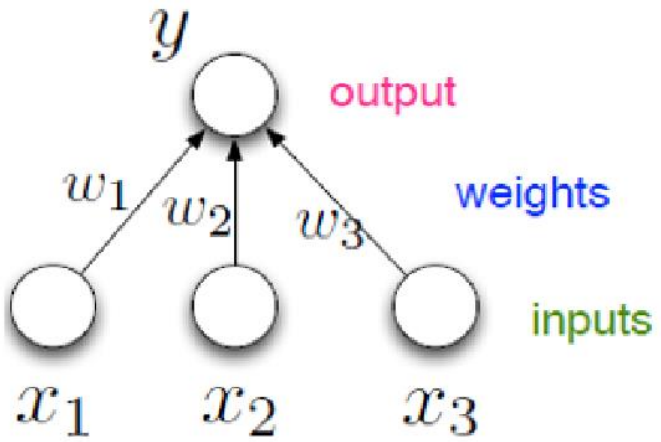
Devin K. Phillips. "Speed of the Human Brain". ASU - Ask A Biologist. 13 May, 2015.
https://askabiologist.asu.edu/plosable/speed-human-brain

Neurons



Devin K. Phillips. "Speed of the Human Brain". ASU - Ask A Biologist. 13 May, 2015.
https://askabiologist.asu.edu/plosable/speed-human-brain

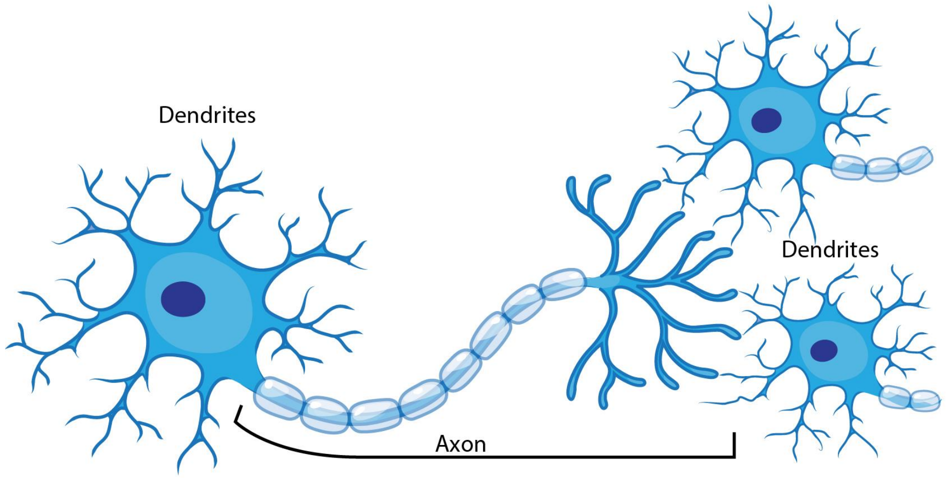
Perceptron aka McCulloch-Pitts Neuron



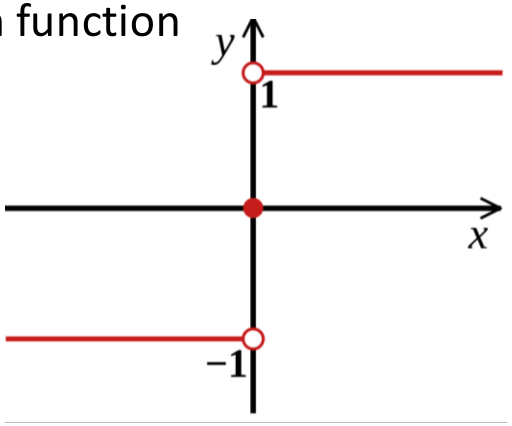
output
bias
i'th weight

$$y = \operatorname{sgn}\left(b + \sum_i x_i w_i\right)$$

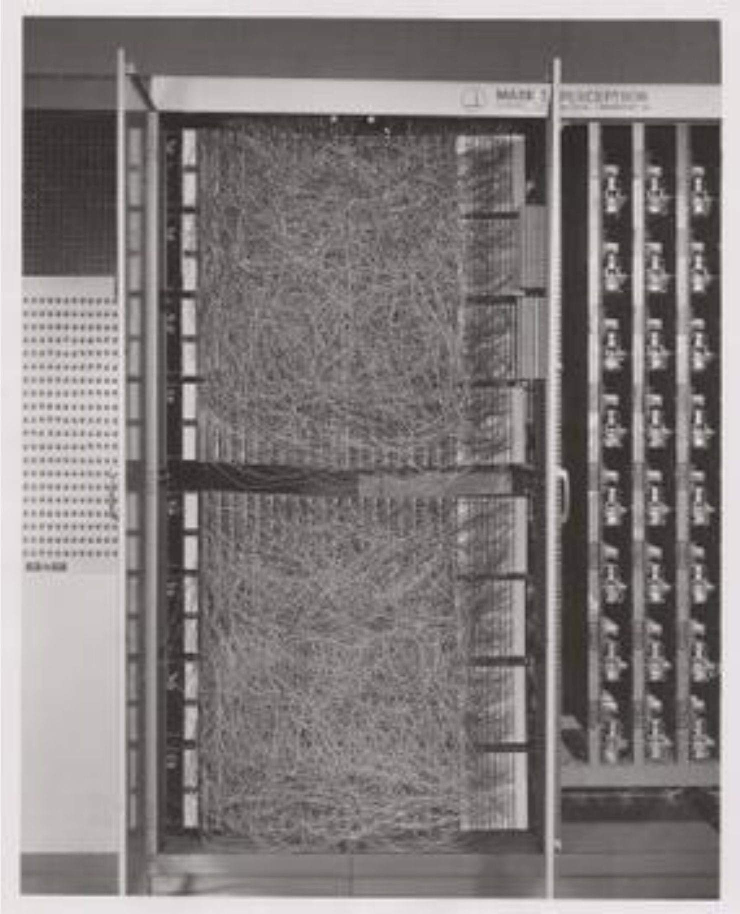
i'th input

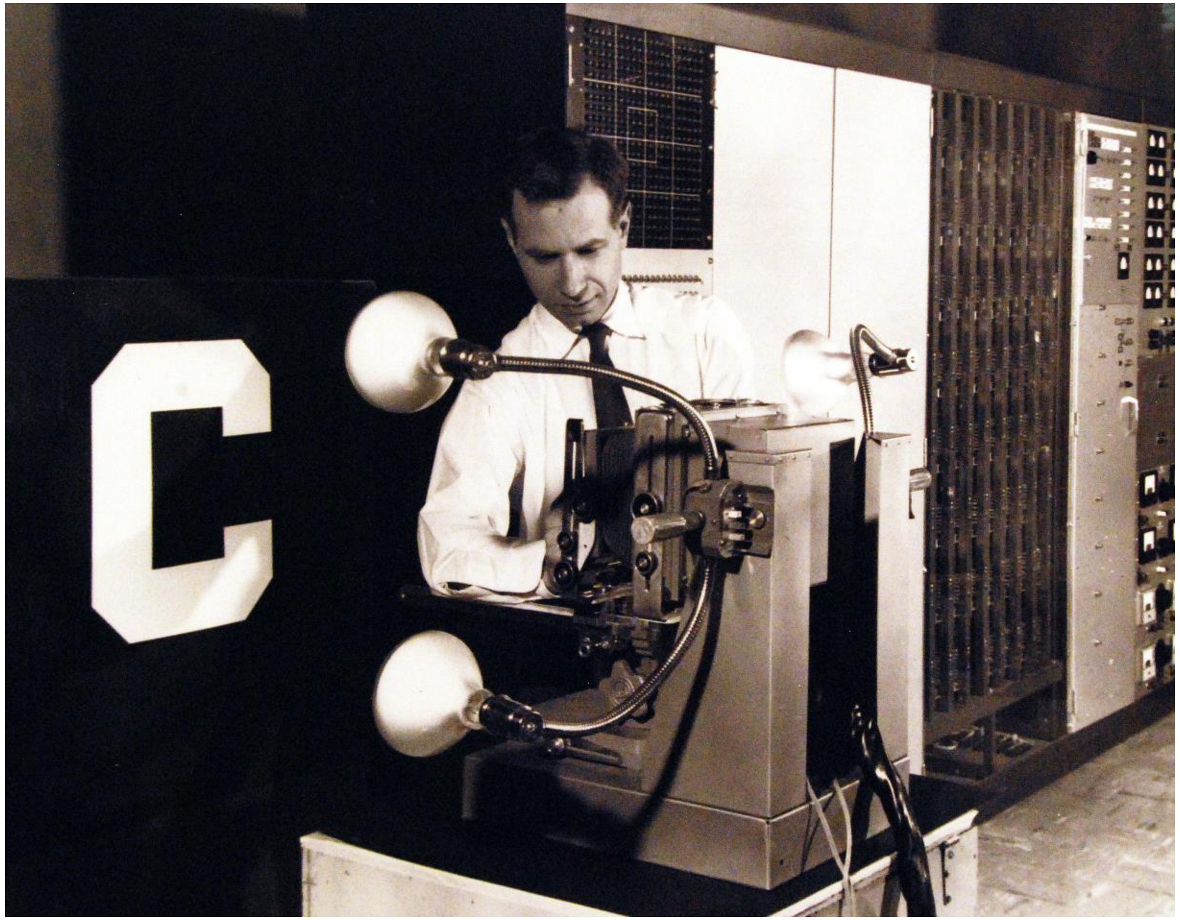


Sign function



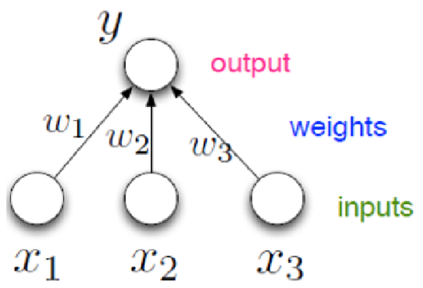
Frank Rosenblatt’s Perceptron





"the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."- NY times 1958

Perceptron Classifier



output

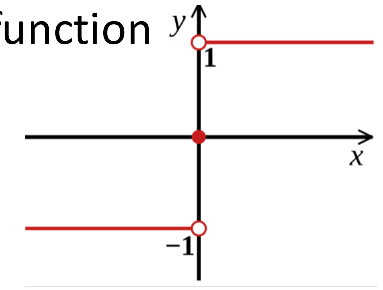
bias

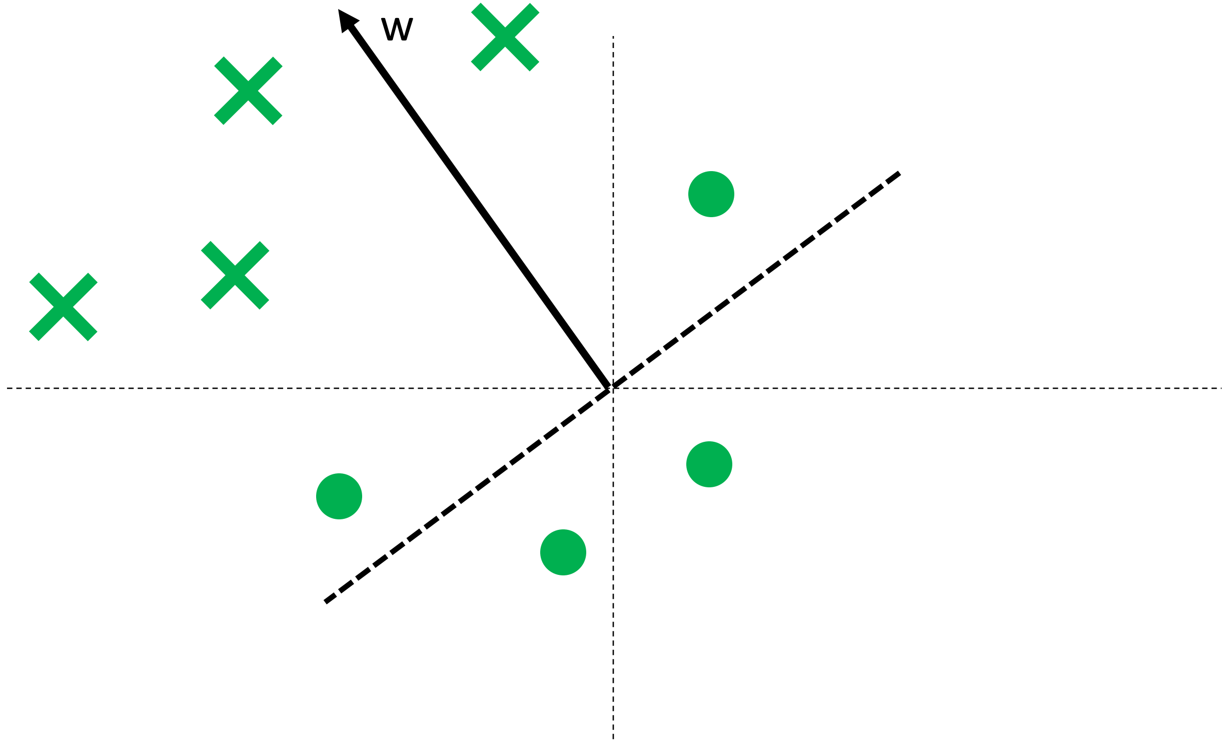
i'th weight

$$y = \mathrm{sgn}\left(b + \sum_i x_i w_i\right)$$

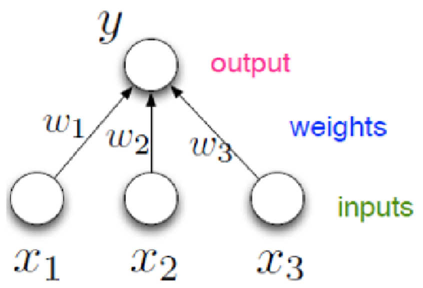
i'th input

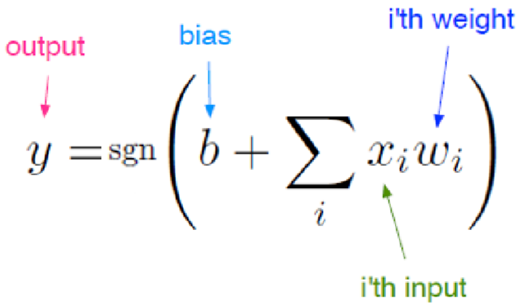
Sign function



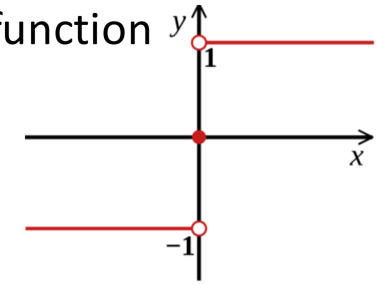


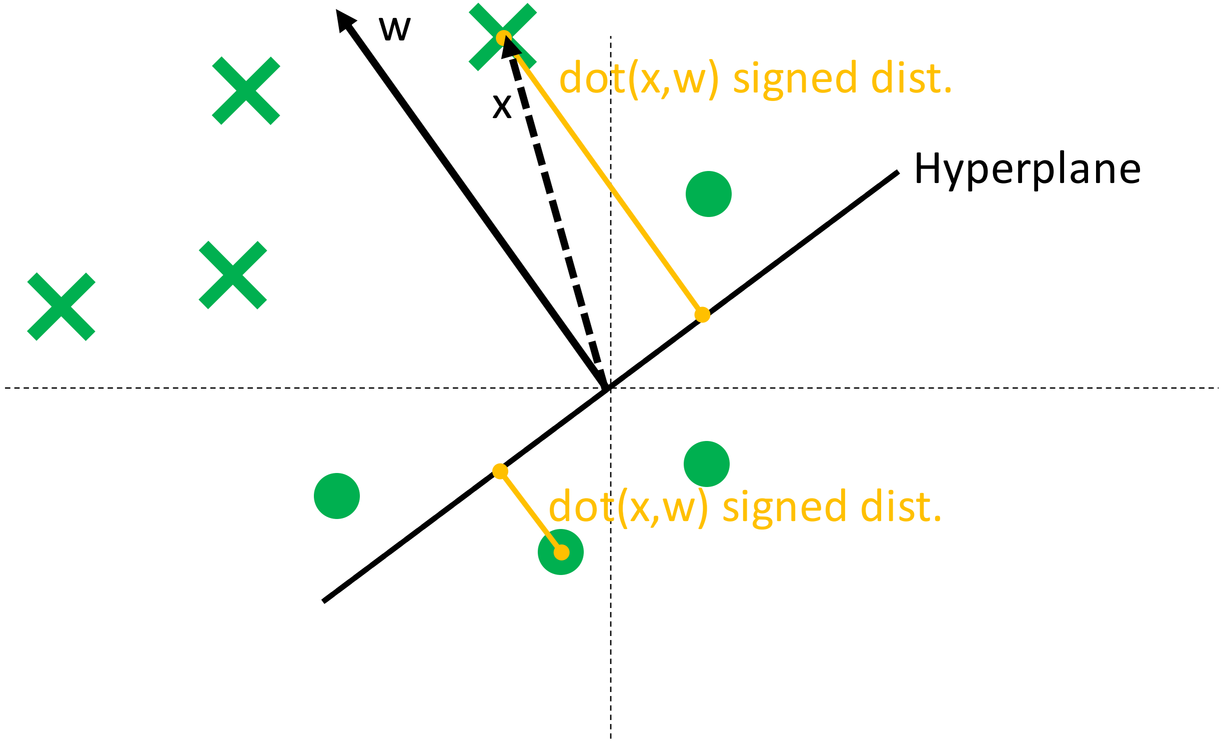
Perceptron Classifier



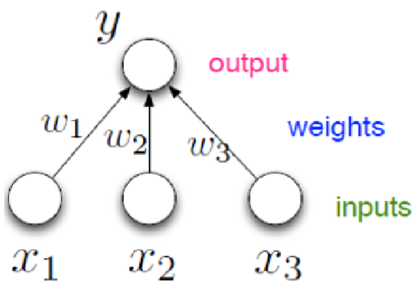


Sign function





Perceptron Classifier

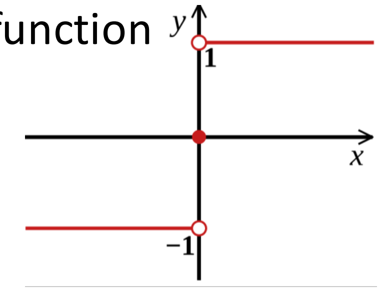


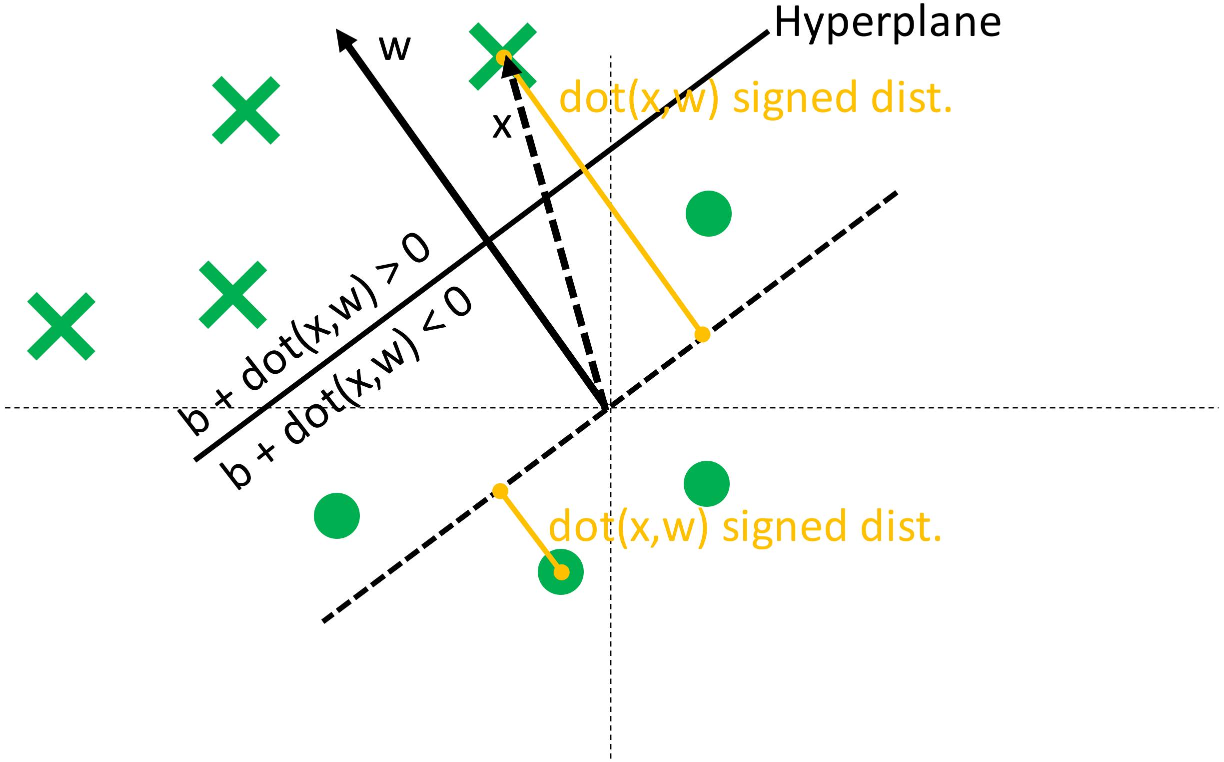
output
bias
i'th weight

$$y = \text{sgn}\left(b + \sum_i x_i w_i\right)$$

i'th input

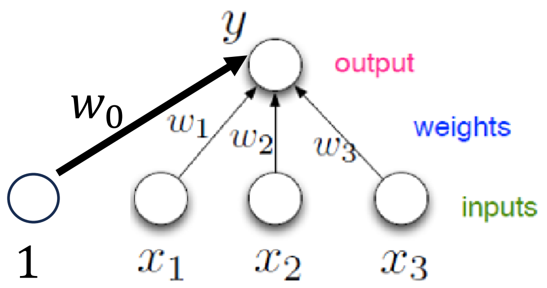
Sign function





Perceptron Classifier (without bias)

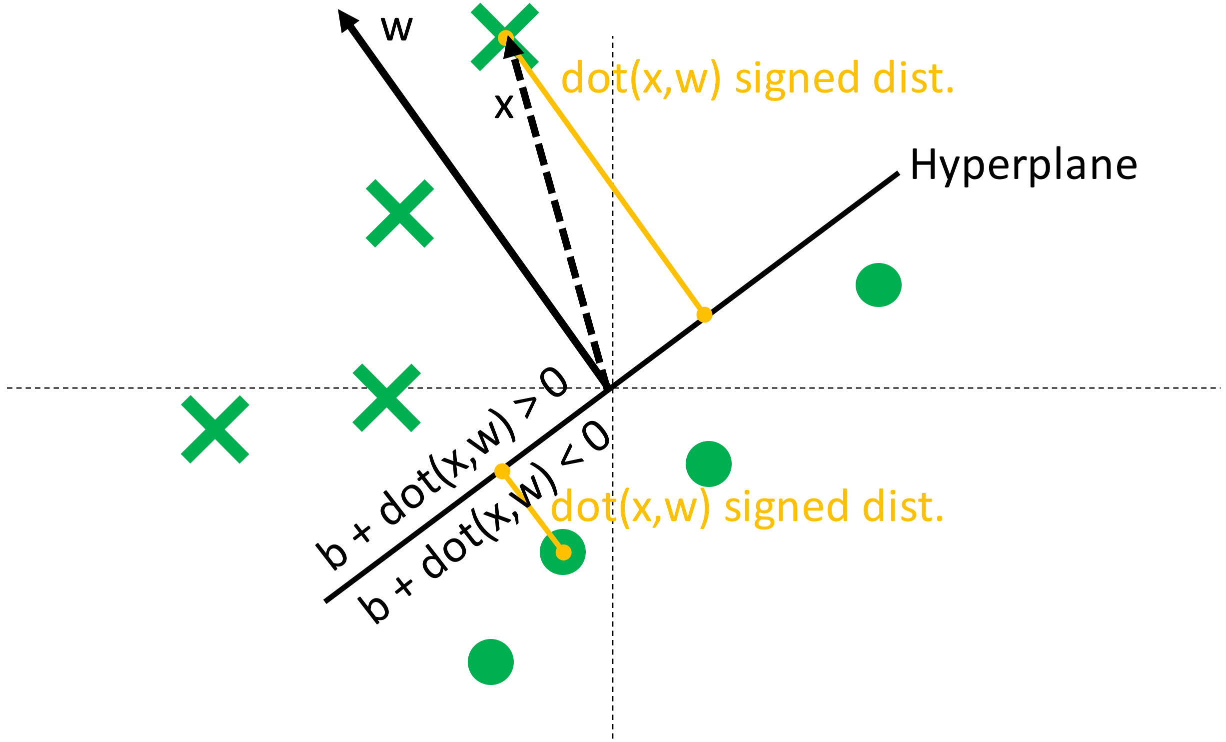
One	Dist (km)	Day	Commute time (min)
x_0	x_1	x_2	y
1	2.7	1	25
1	4.1	1	33
1	1.0	0	15
1	5.2	1	45
1	2.8	0	22



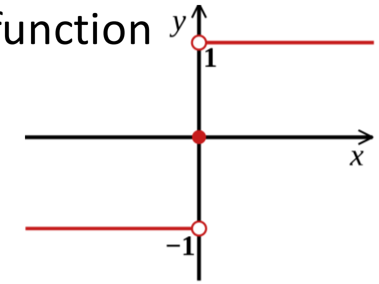
output bias i'th weight

$$y = \mathrm{sgn}\left(\cancel{b} + \sum_i x_i w_i\right)$$

i'th input



Sign function



Perceptron Algorithm

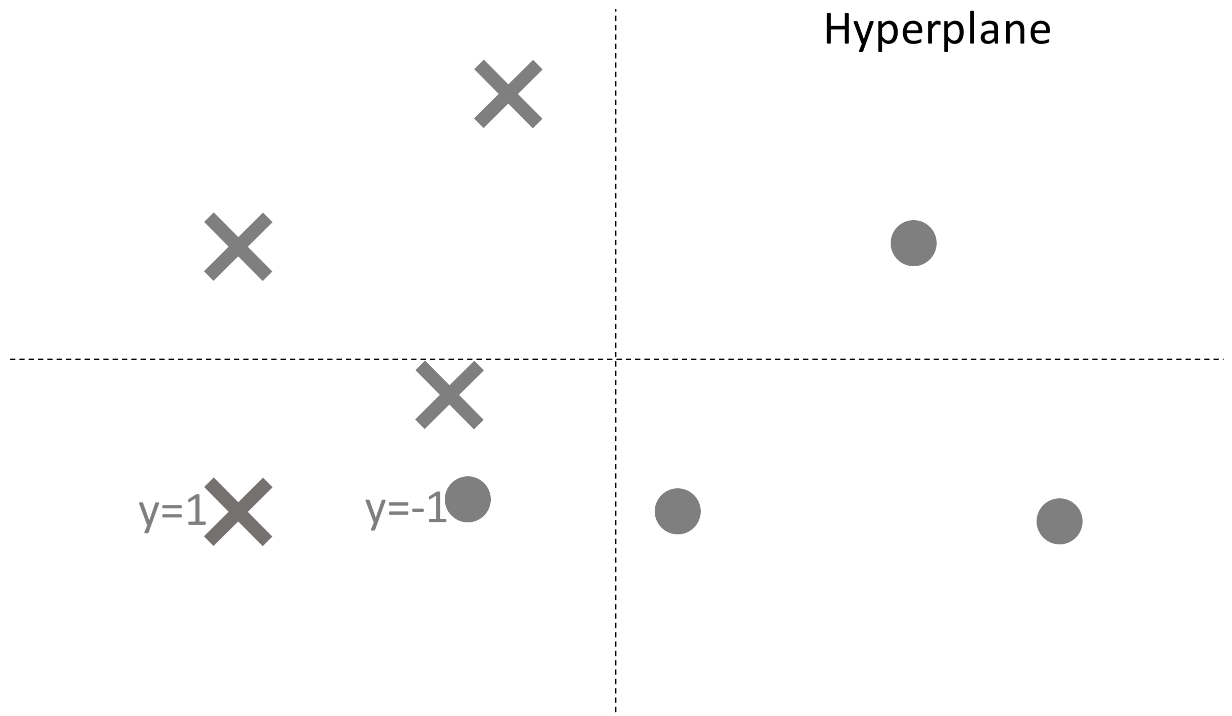
Initialize $\mathbf{w} = 0$ $\quad\triangleright$ we assume no bias for simplicity

while All training examples are not correctly classified **do**

  for $(\mathbf{x}, y) \in S$ **do** $\quad\triangleright$ Loop over each (feature, label) pair in the dataset

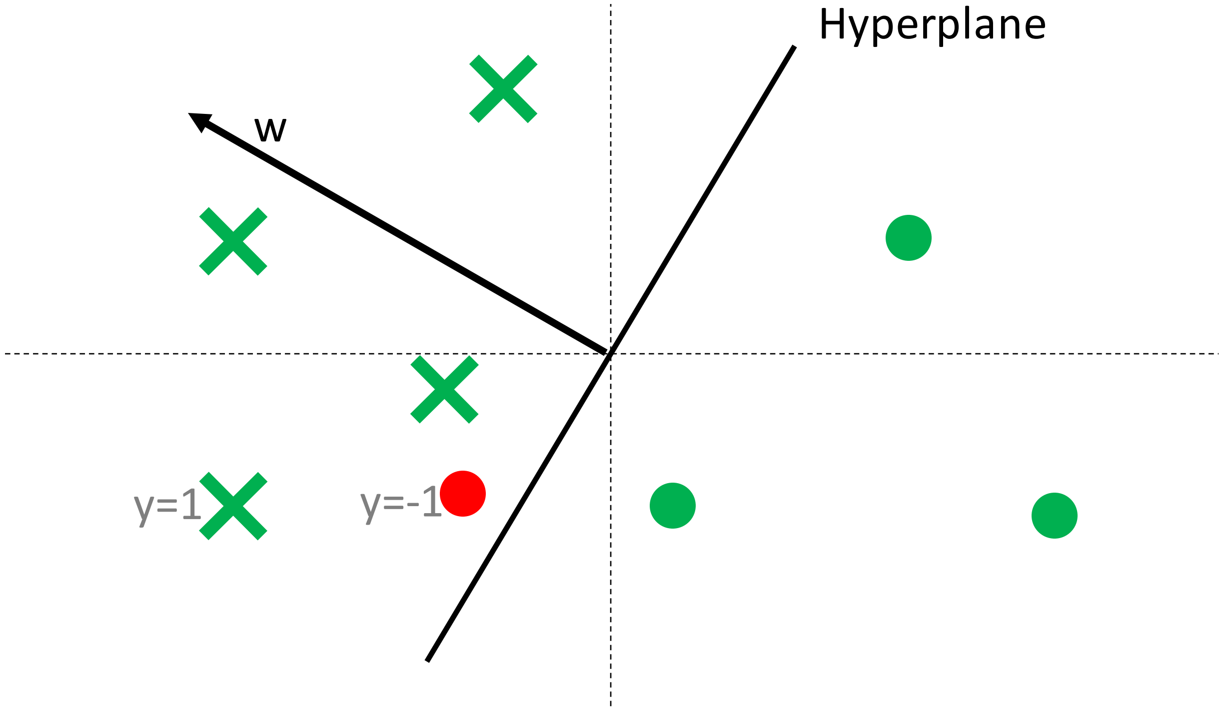
    if $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** $\quad\triangleright$ If the pair $(\mathbf{x}, y)$ is misclassified

      $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ $\quad\triangleright$ Update the weight vector $\mathbf{w}$



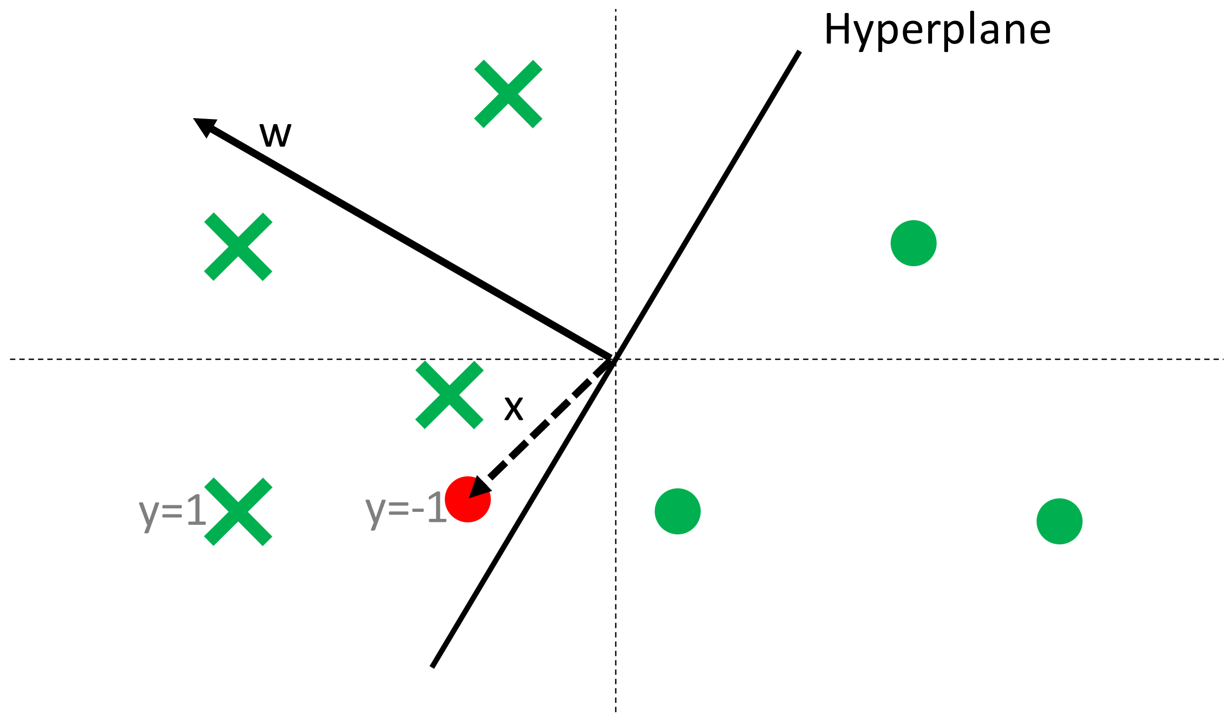
Perceptron Algorithm

Initialize $\mathbf{w} = 0$ $\triangleright$ we assume no bias for simplicity
while All training examples are not correctly classified **do**
  for $(\mathbf{x}, y) \in S$ **do** $\triangleright$ Loop over each (feature, label) pair in the dataset
    if $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** $\triangleright$ If the pair $(\mathbf{x}, y)$ is misclassified
      $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ $\triangleright$ Update the weight vector $\mathbf{w}$



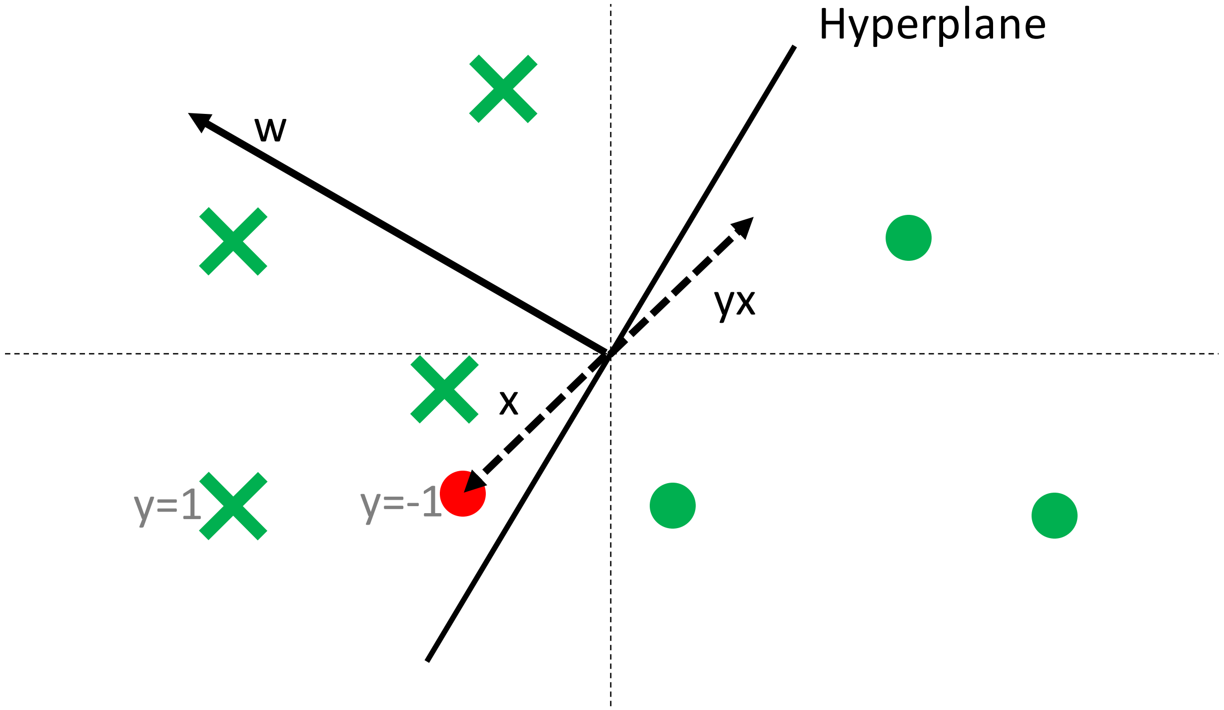
Perceptron Algorithm

Initialize $\mathbf{w} = 0$ ▷ we assume no bias for simplicity
while All training examples are not correctly classified **do**
  for $(\mathbf{x}, y) \in S$ **do** ▷ Loop over each (feature, label) pair in the dataset
    if $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** ▷ If the pair $(\mathbf{x}, y)$ is misclassified
      $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ ▷ Update the weight vector $\mathbf{w}$



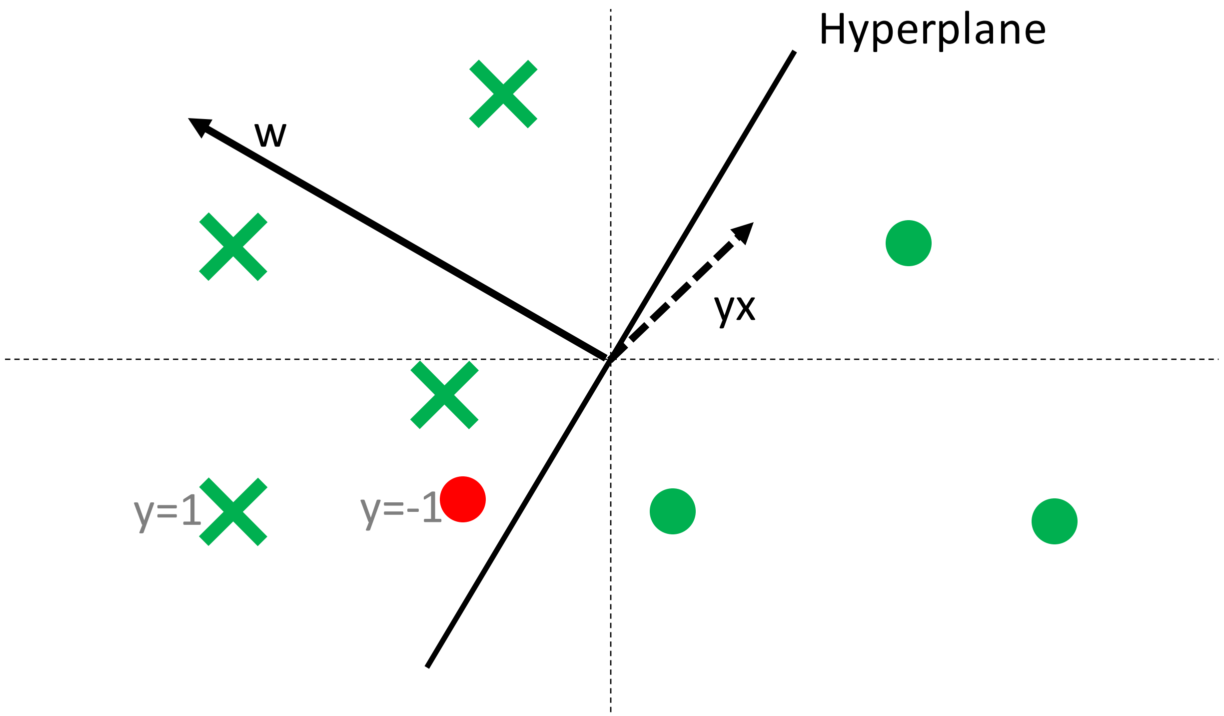
Perceptron Algorithm

Initialize $\mathbf{w} = 0$ ▷ we assume no bias for simplicity
while All training examples are not correctly classified **do**
    for $(\mathbf{x}, y) \in S$ **do** ▷ Loop over each (feature, label) pair in the dataset
        if $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** ▷ If the pair $(\mathbf{x}, y)$ is misclassified
            $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ ▷ Update the weight vector $\mathbf{w}$



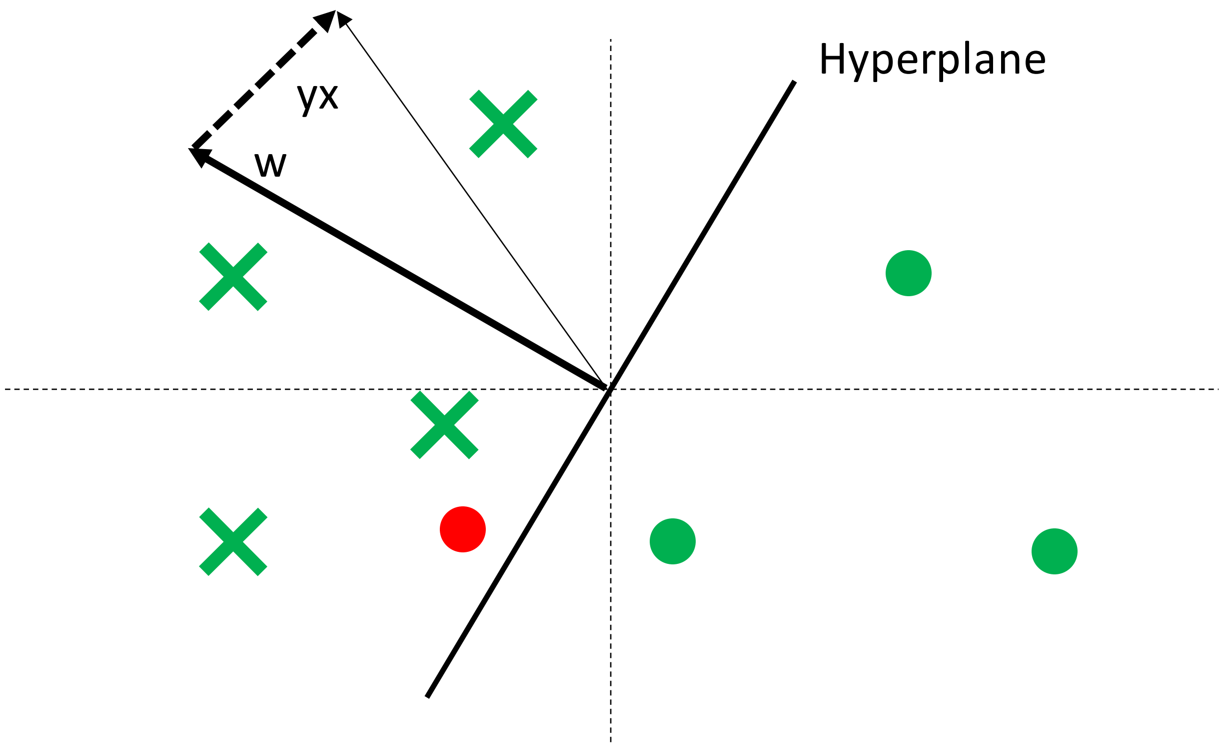
Perceptron Algorithm

Initialize $\mathbf{w} = 0$ ▷ we assume no bias for simplicity
while All training examples are not correctly classified **do**
  for $(\mathbf{x}, y) \in S$ **do** ▷ Loop over each (feature, label) pair in the dataset
    if $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** ▷ If the pair $(\mathbf{x}, y)$ is misclassified
      $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ ▷ Update the weight vector $\mathbf{w}$



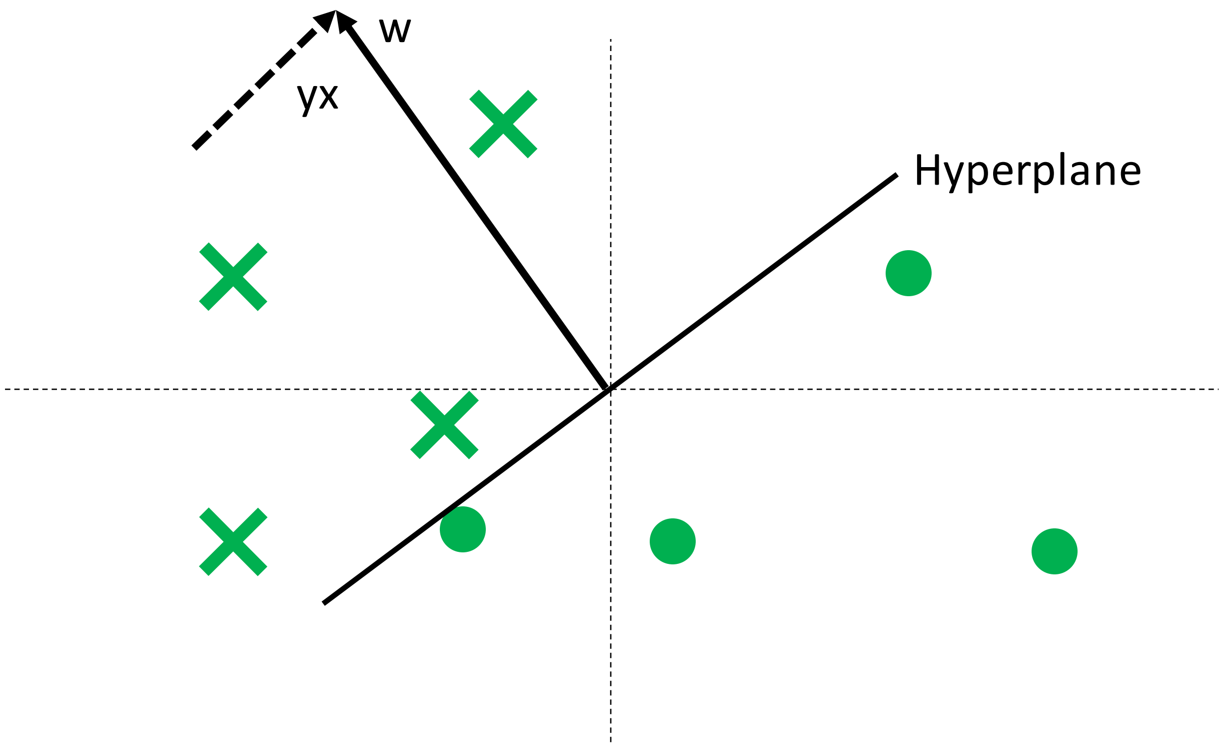
Perceptron Algorithm

1: Initialize $\mathbf{w} = 0$ ▷ we assume no bias for simplicity
2: **while** All training examples are not correctly classified **do**
3: **for** $(\mathbf{x}, y) \in S$ **do** ▷ Loop over each (feature, label) pair in the dataset
4: **if** $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** ▷ If the pair $(\mathbf{x}, y)$ is misclassified
5: $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ ▷ Update the weight vector $\mathbf{w}$

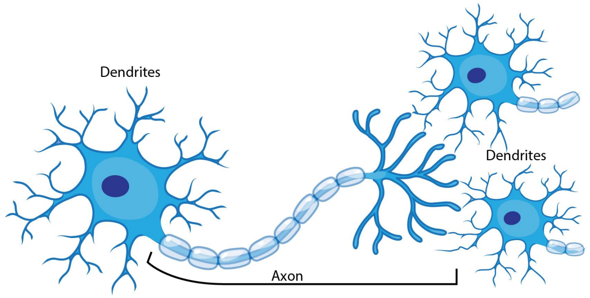


Perceptron Algorithm

Initialize $\mathbf{w} = 0$ ▷ we assume no bias for simplicity
while All training examples are not correctly classified **do**
  for $(\mathbf{x}, y) \in S$ **do** ▷ Loop over each (feature, label) pair in the dataset
    if $y \cdot \mathbf{w}^\top \mathbf{x} \leq 0$ **then** ▷ If the pair $(\mathbf{x}, y)$ is misclassified
      $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ ▷ Update the weight vector $\mathbf{w}$



Summary



Perceptron …

- is (very) simple model of biological neuron

- implements linear classifier

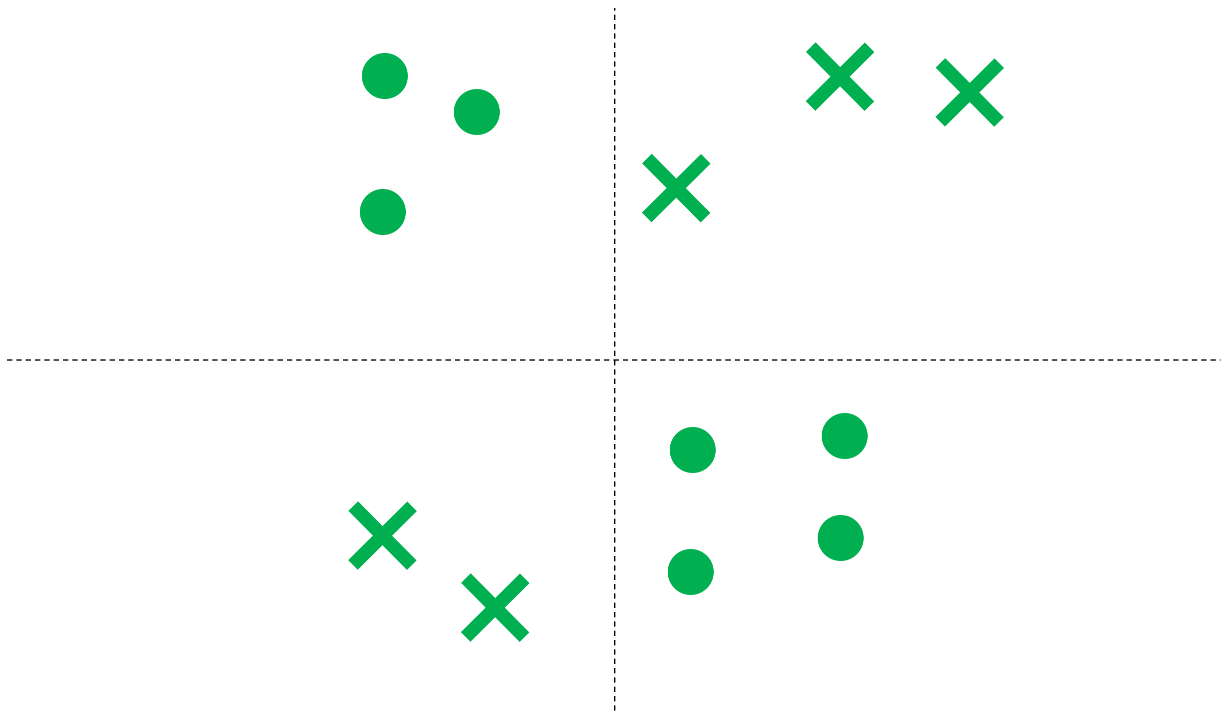
$$\overset{\text{output}}{y} = \mathrm{sgn}\left(\overset{\text{bias}}{b} + \sum_i \underbrace{x_i}_{\text{i'th input}} \overset{\text{i'th weight}}{w_i}\right)$$

- Algorithm finds weights and biases
  - Guaranteed to stop if linearly separable

Perceptron Algorithm

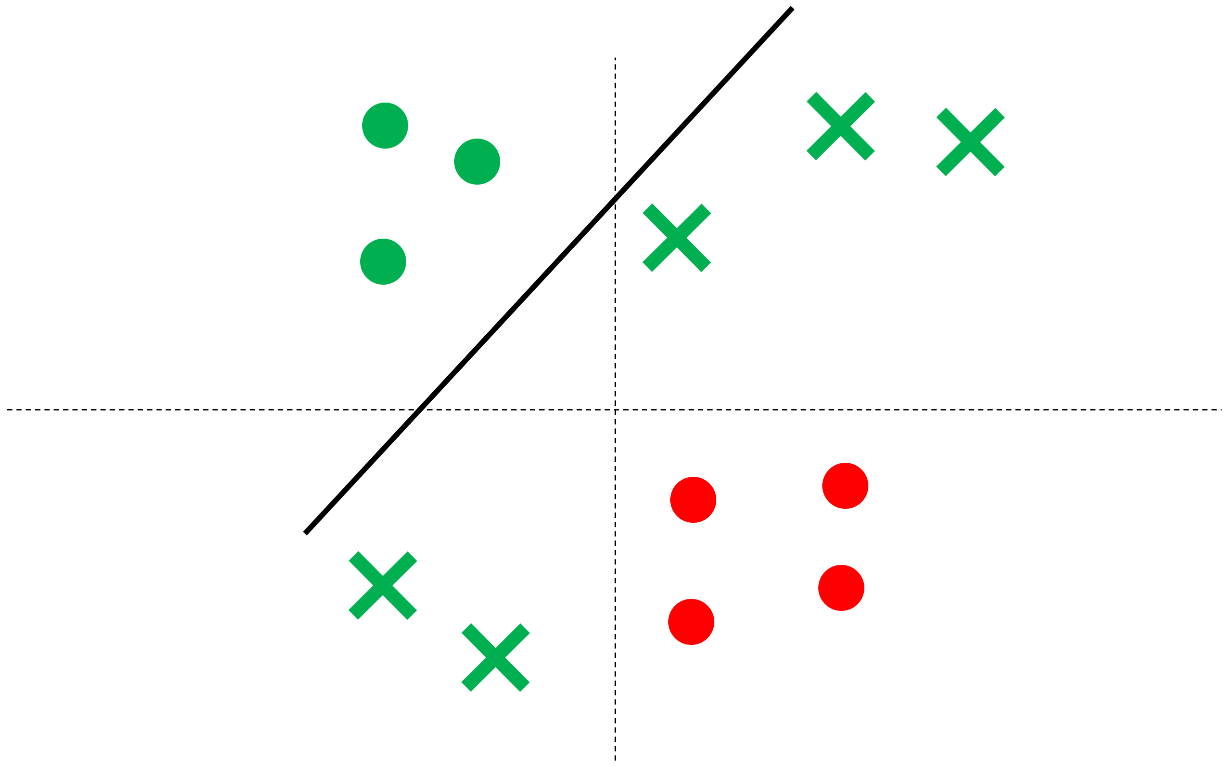
```
Initialize w = 0                                    ▷ we assume no bias for simplicity
while All training examples are not correctly classified do
    for (x, y) ∈ S do                   ▷ Loop over each (feature, label) pair in the dataset
        if y · w^⊤ x ≤ 0 then                         ▷ If the pair (x, y) is misclassified
            w ← w + yx                                  ▷ Update the weight vector w
```

Limitations



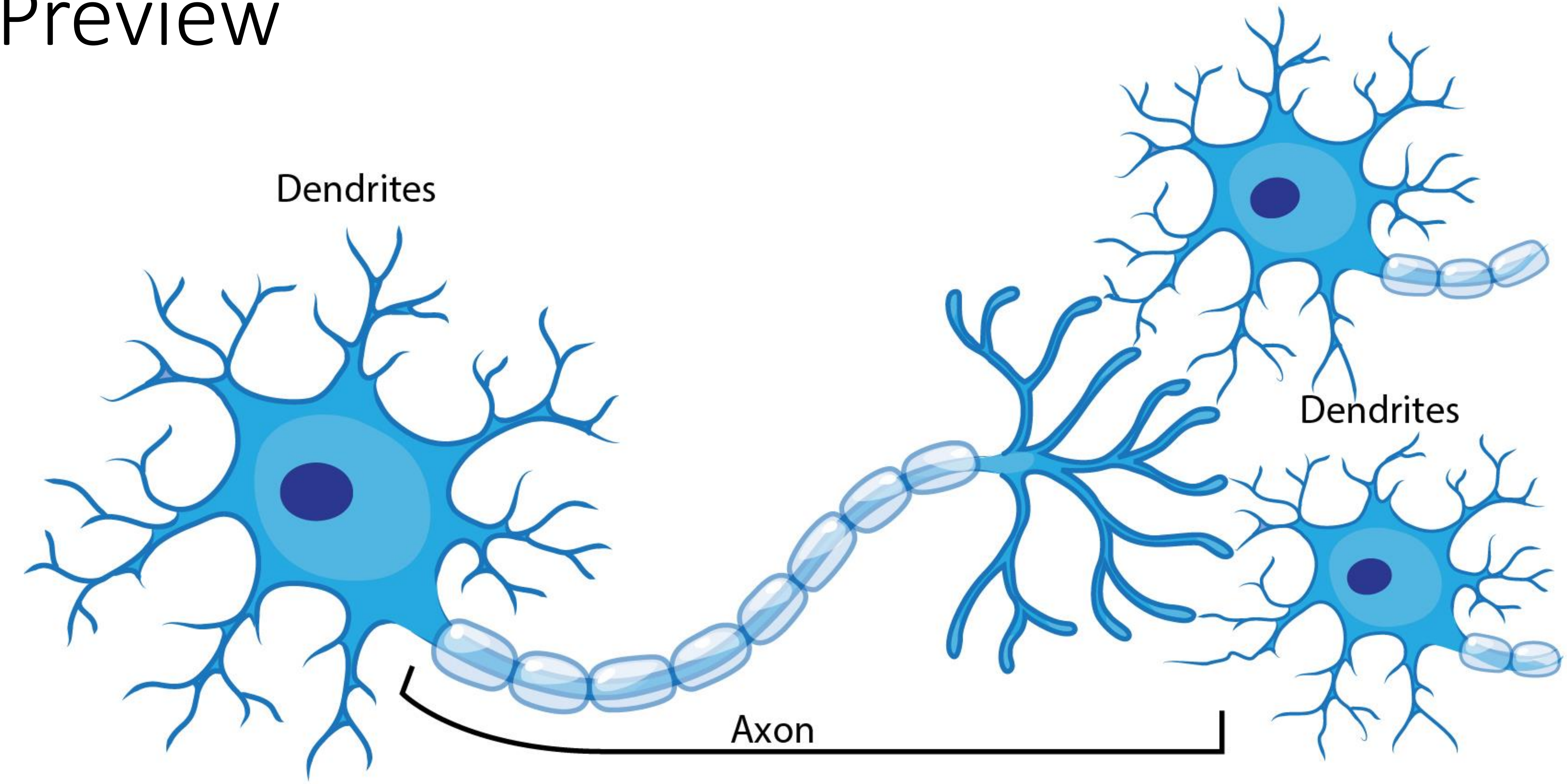
Some data is not linearly separable

Limitations



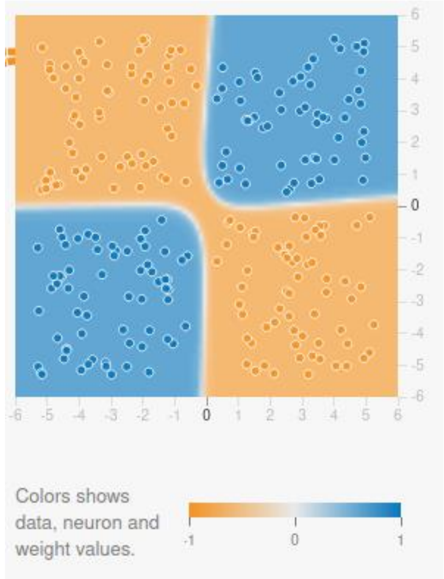
Some data is not linearly separable

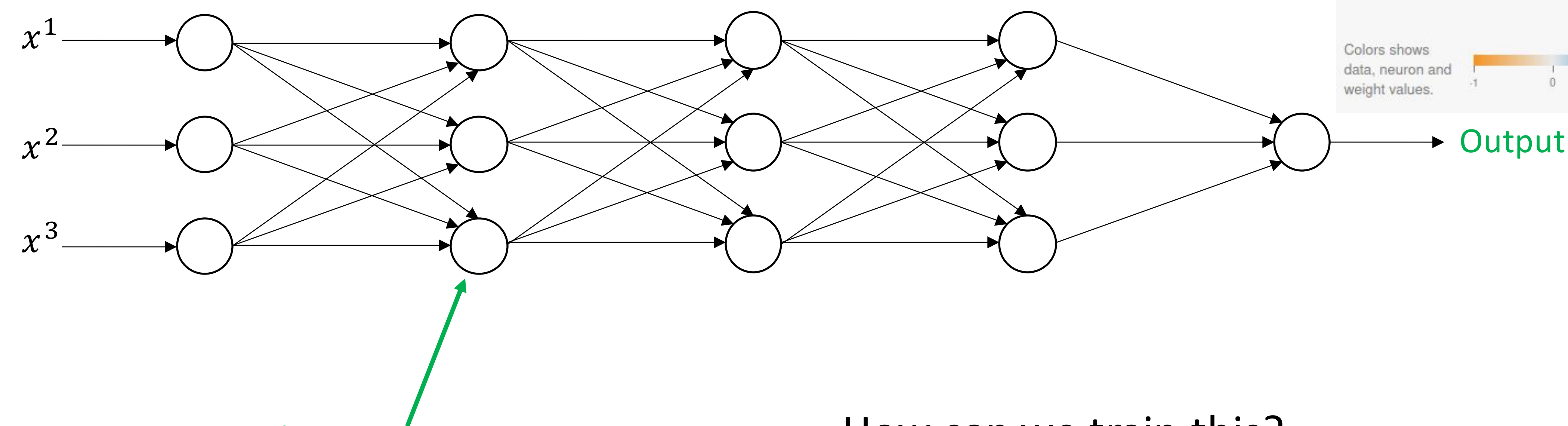
Preview



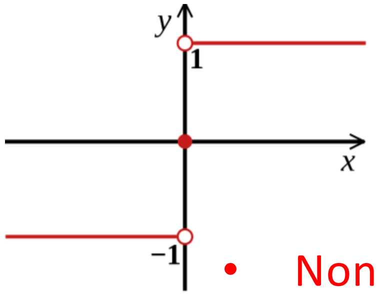
Devin K. Phillips. "Speed of the Human Brain". ASU - Ask A Biologist. 13 May, 2015.
https://askabiologist.asu.edu/plosable/speed-human-brain

Preview





This is a perceptron



output   bias   i'th weight

$$y = \text{sgn}\left(b + \sum_i x_i w_i\right)$$

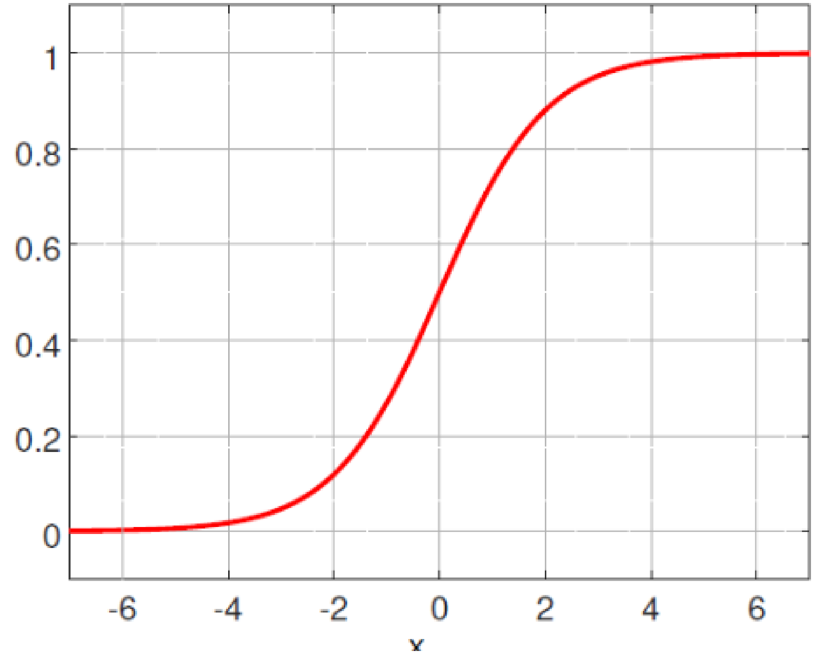
i'th input

- Non-differentiable
- Problem   Build soft perceptron

How can we train this?

- Perceptron algorithm no longer works
- Use gradient descent

The idea is to replace the sgn function with a differentiable non-linear function

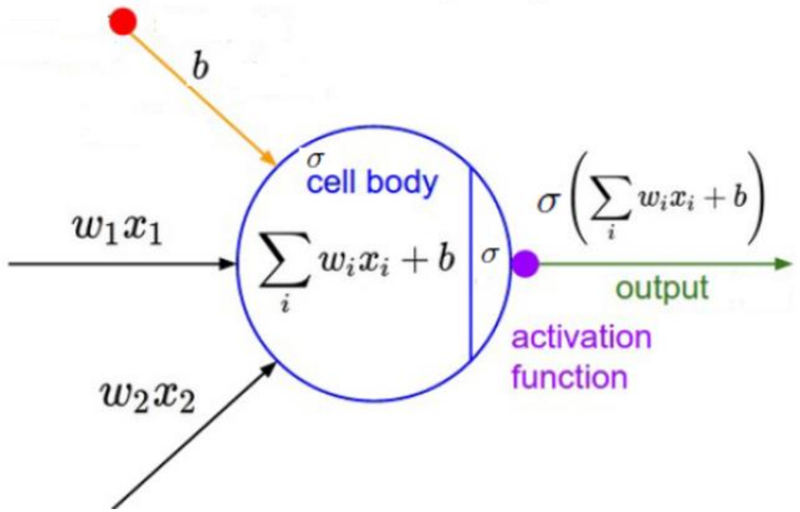


e.g., sigmoid function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Mapping: $(-\infty, +\infty) \mapsto (0, 1)$

$\sigma'(x) = \sigma(x)(1 - \sigma(x))$ (exercise)



$$f(\mathbf{x}) = \sigma\left(\sum_i w_i x_i + b\right)$$

Interpret as probability

Training

Dataset: n input/output pairs $S = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})\}$

Mean-Square Error

$$C(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} \Big(\underbrace{\sigma(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}_{\text{predicted output}} - \underbrace{y^{(i)}}_{\text{output}} \Big)^2.$$

How to compute?

$$\boxed{\frac{\partial(\sigma(\mathbf{w}^\top \mathbf{x}+b)-y)^2}{\partial w_i}}$$

In the next video

No closed form solution for the minimizer of $C(\mathbf{w})$

Use Gradient Descent to train a $\mathbf{w} \in \mathbb{R}^d$ with a small $C(\mathbf{w})$

$$\mathbf{w}^{\text{new}} = \mathbf{w} - \eta \nabla C(\mathbf{w}) \quad \Longleftrightarrow \quad w_i^{\text{new}} = w_i - \eta \frac{\partial C(\mathbf{w})}{\partial w_i}.$$