TÉCNICO LISBOA

Planning, Learning and Decision Making

Lecture 15. Learning from examples - Neural approaches

Approaches to learning

- "Symbolic" approach → Learn rules (DT)
- Probabilistic approach → Learn probabilities (LR, NB)
- Similarity-based approach → Learn by similarity (kNN, SVM)

Neural approach

Neural approach

- Inspired by how our brain works
- Idea has been around since 1940s (age of “cybernetics”)

Linear Threshold Unit

- McCulloch & Pitts (1943)

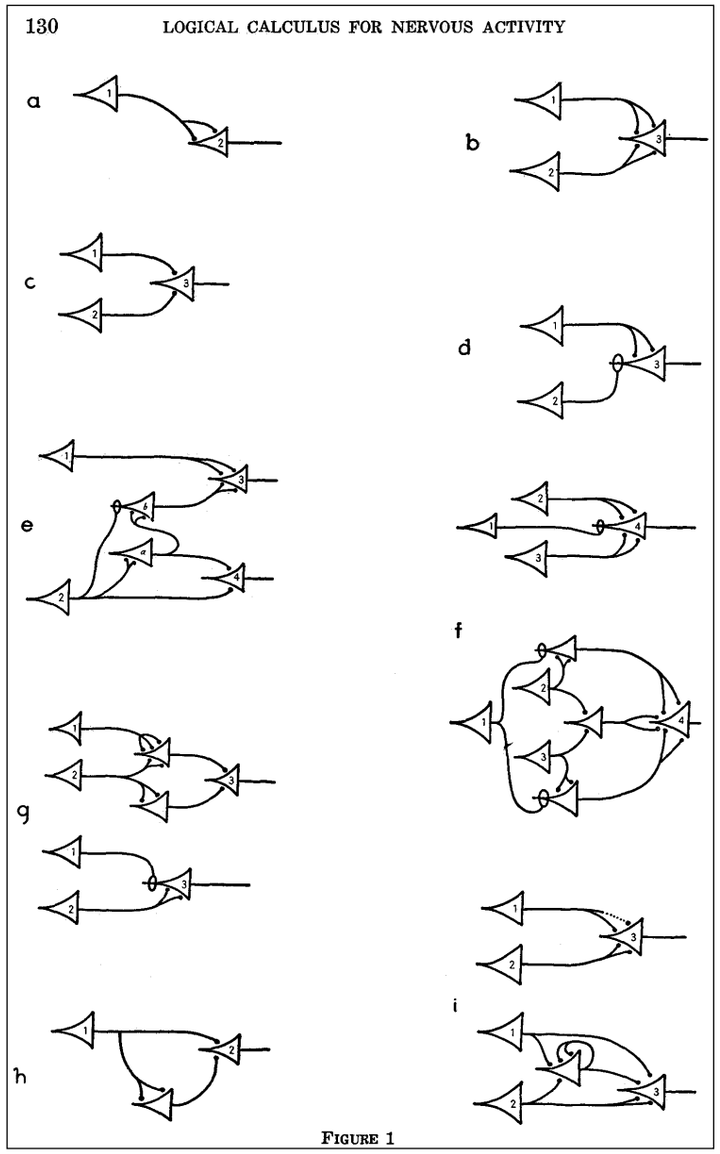
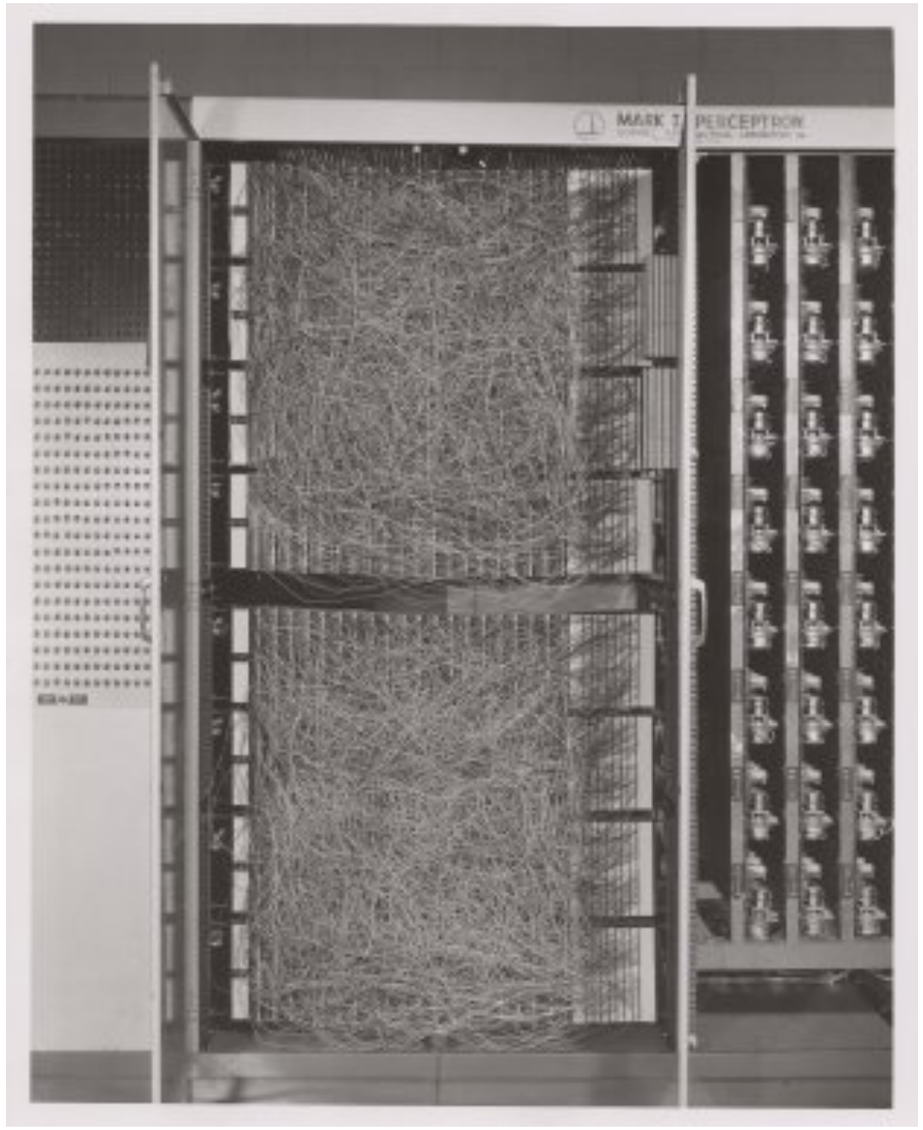


FIGURE 1

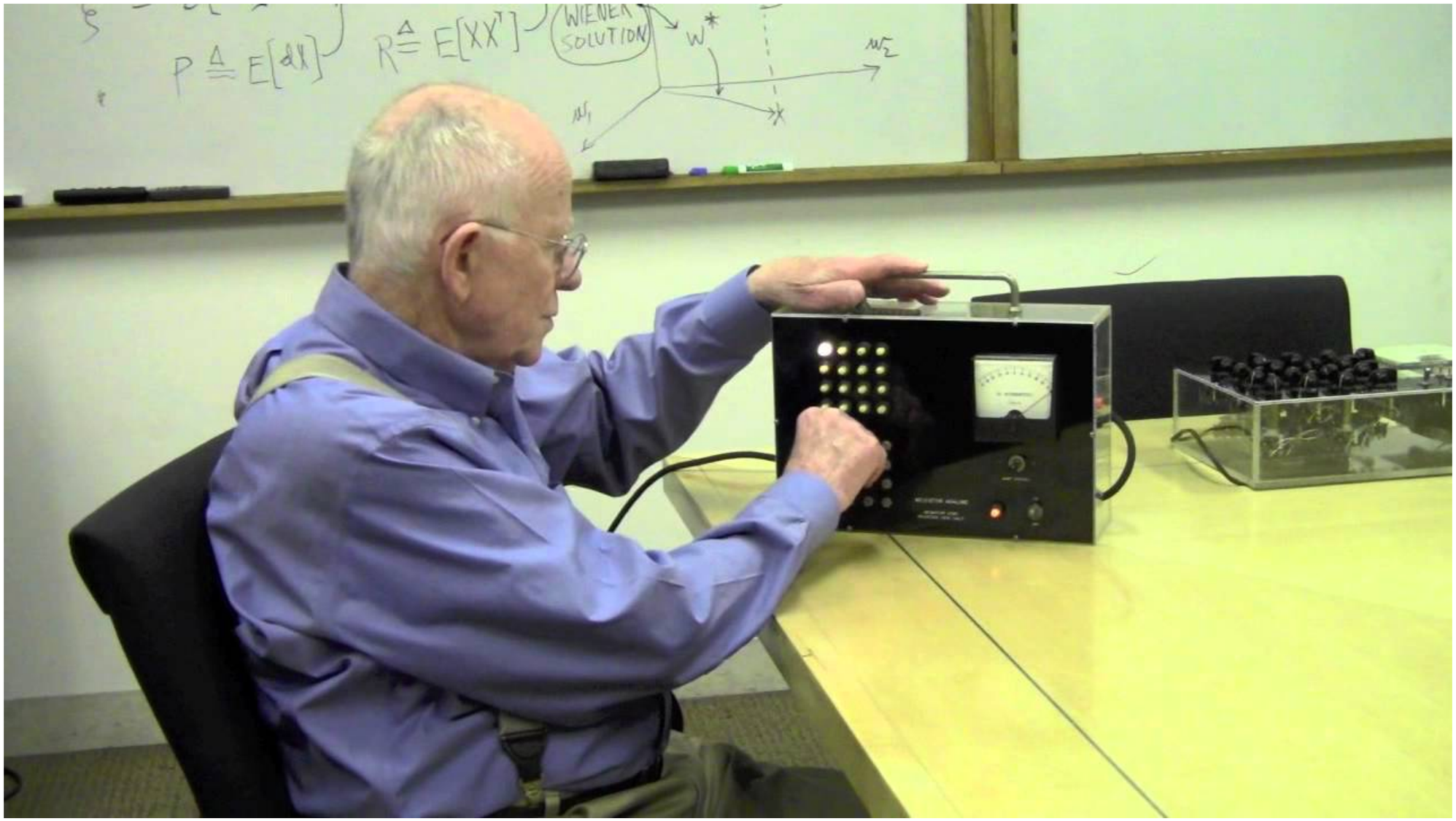
Perceptron

- Rosenblatt, 1962



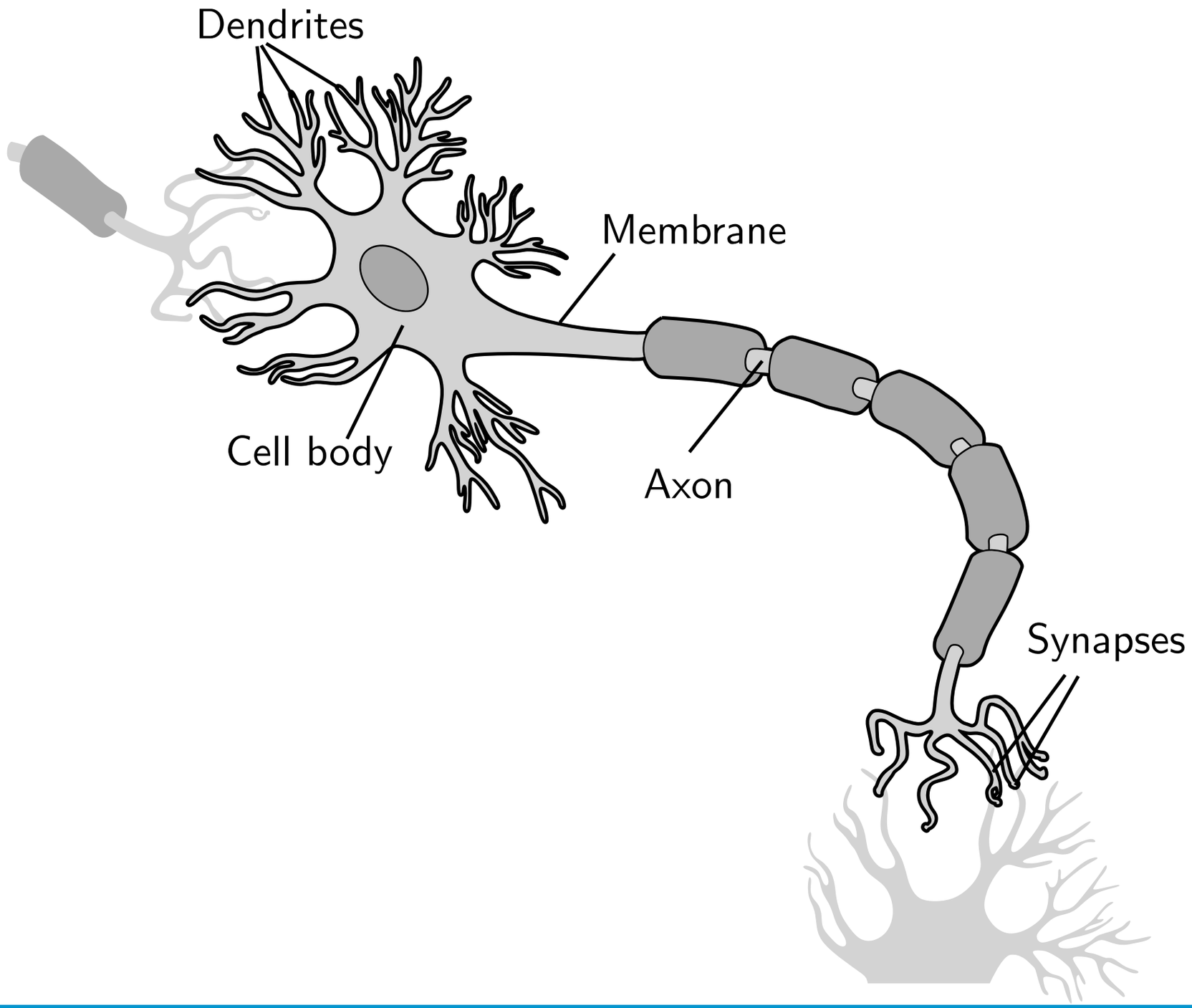
Adaline

- Widrow & Hoff (1960s)

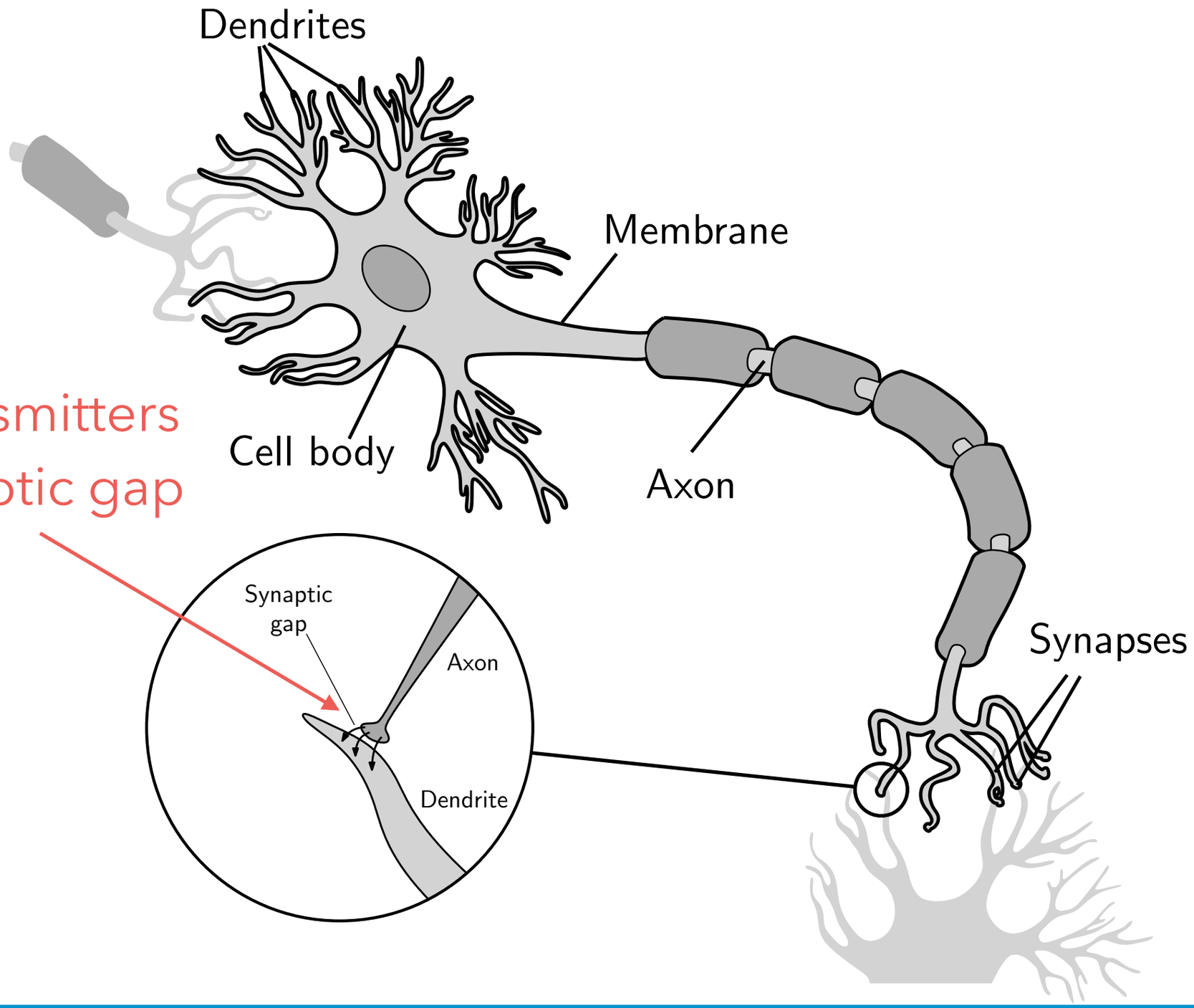


What's this all about?

Actual neuron

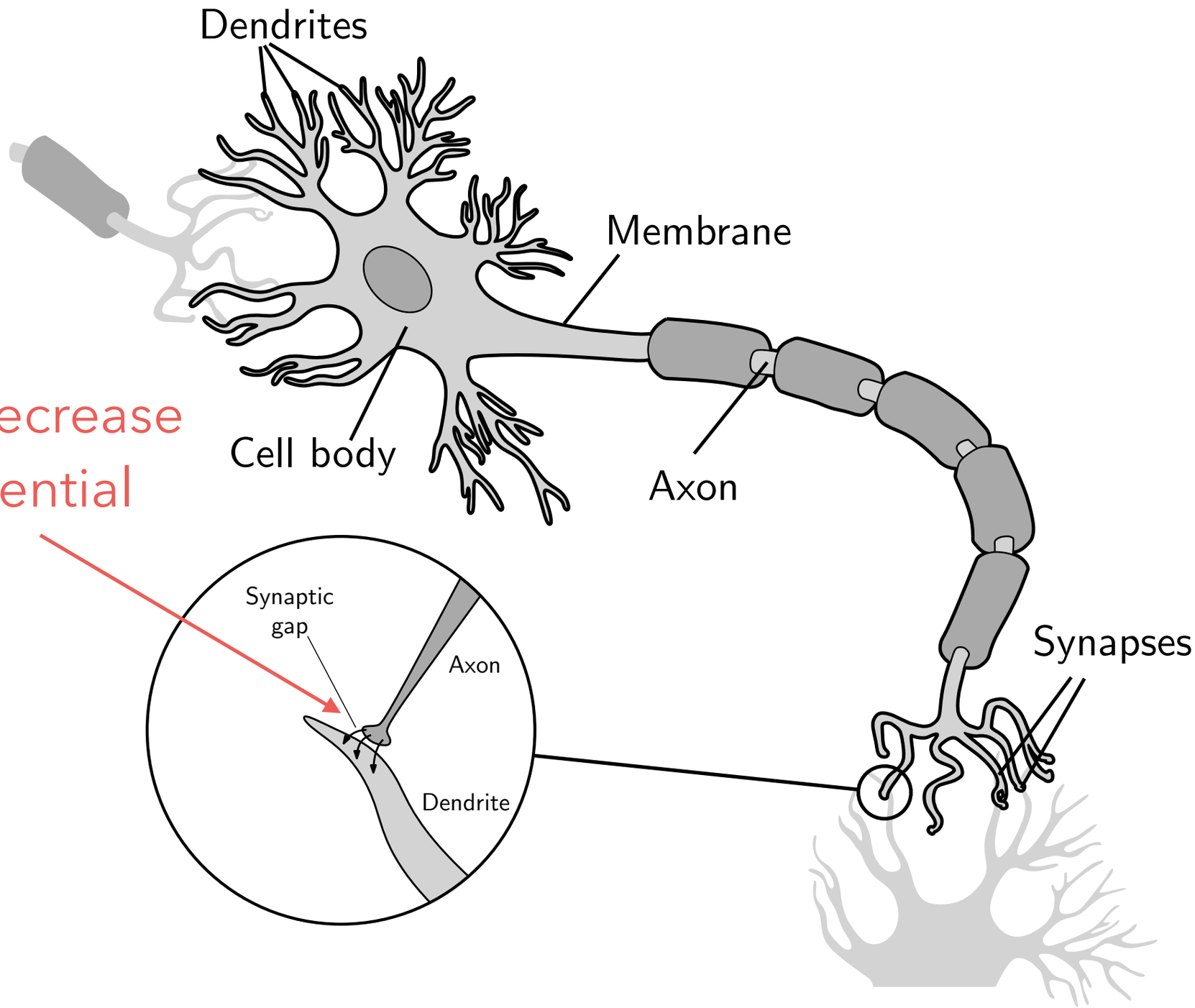


Actual neuron

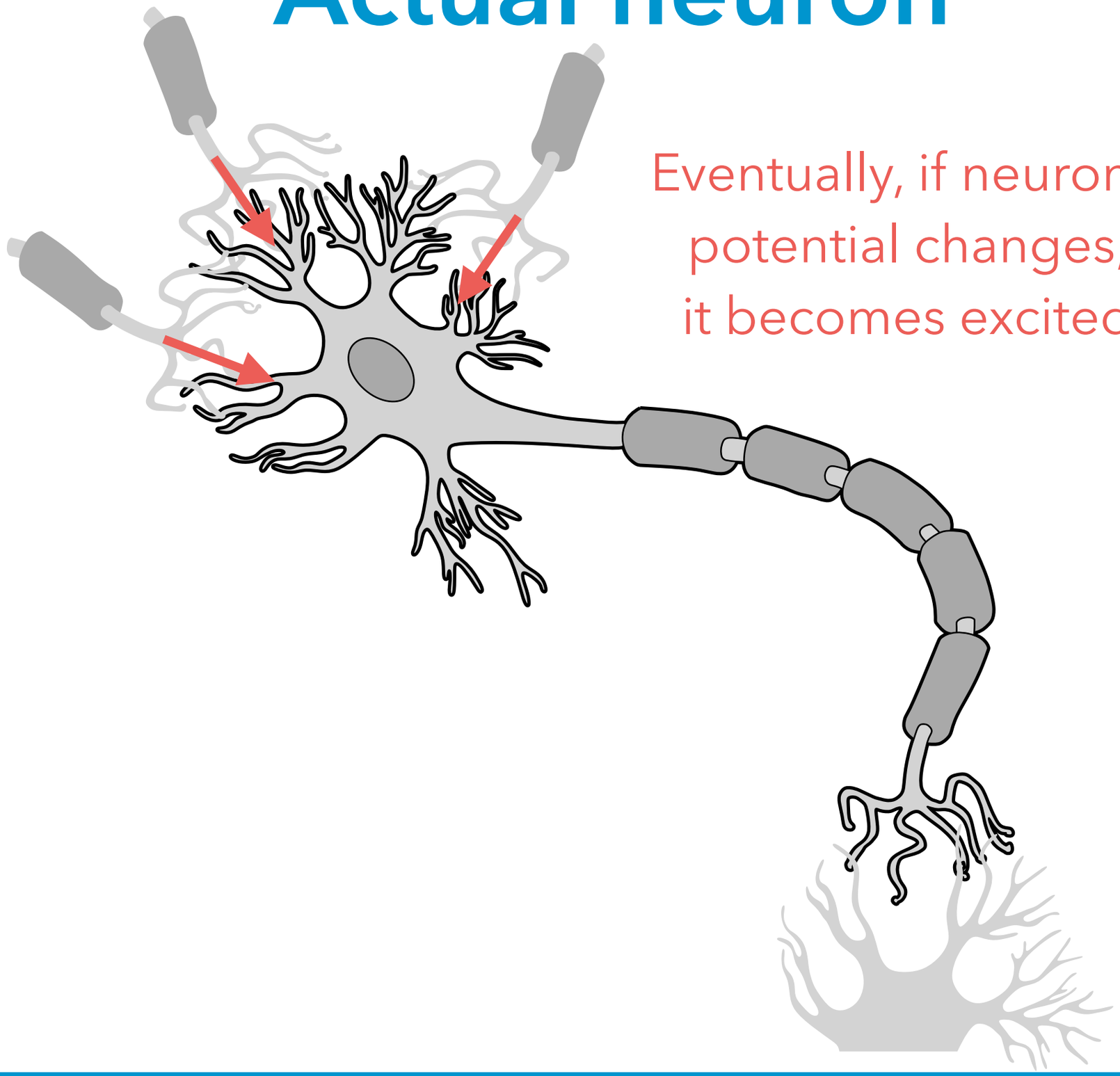


Neurotransmitters cross synaptic gap

Actual neuron

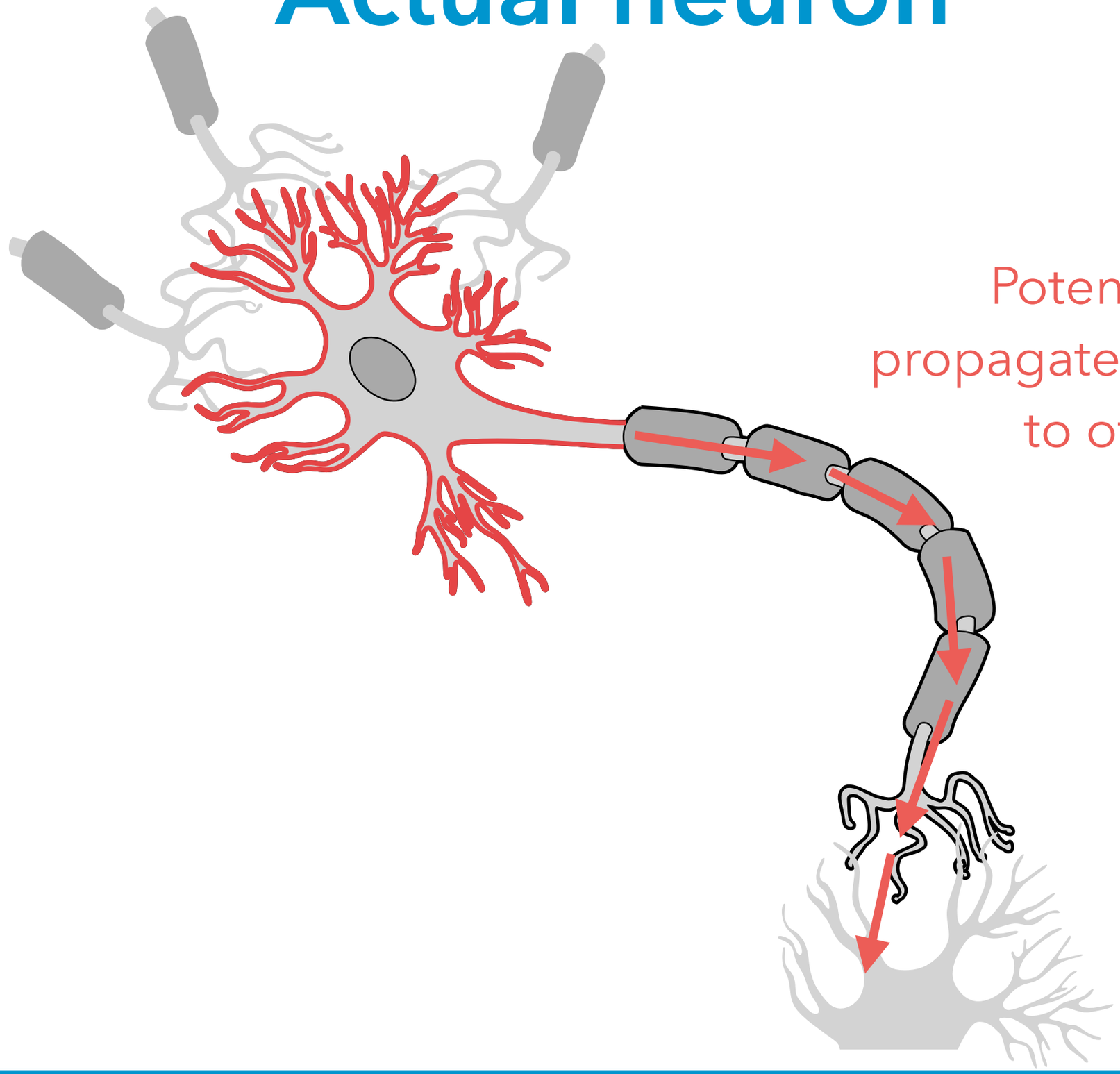


Actual neuron



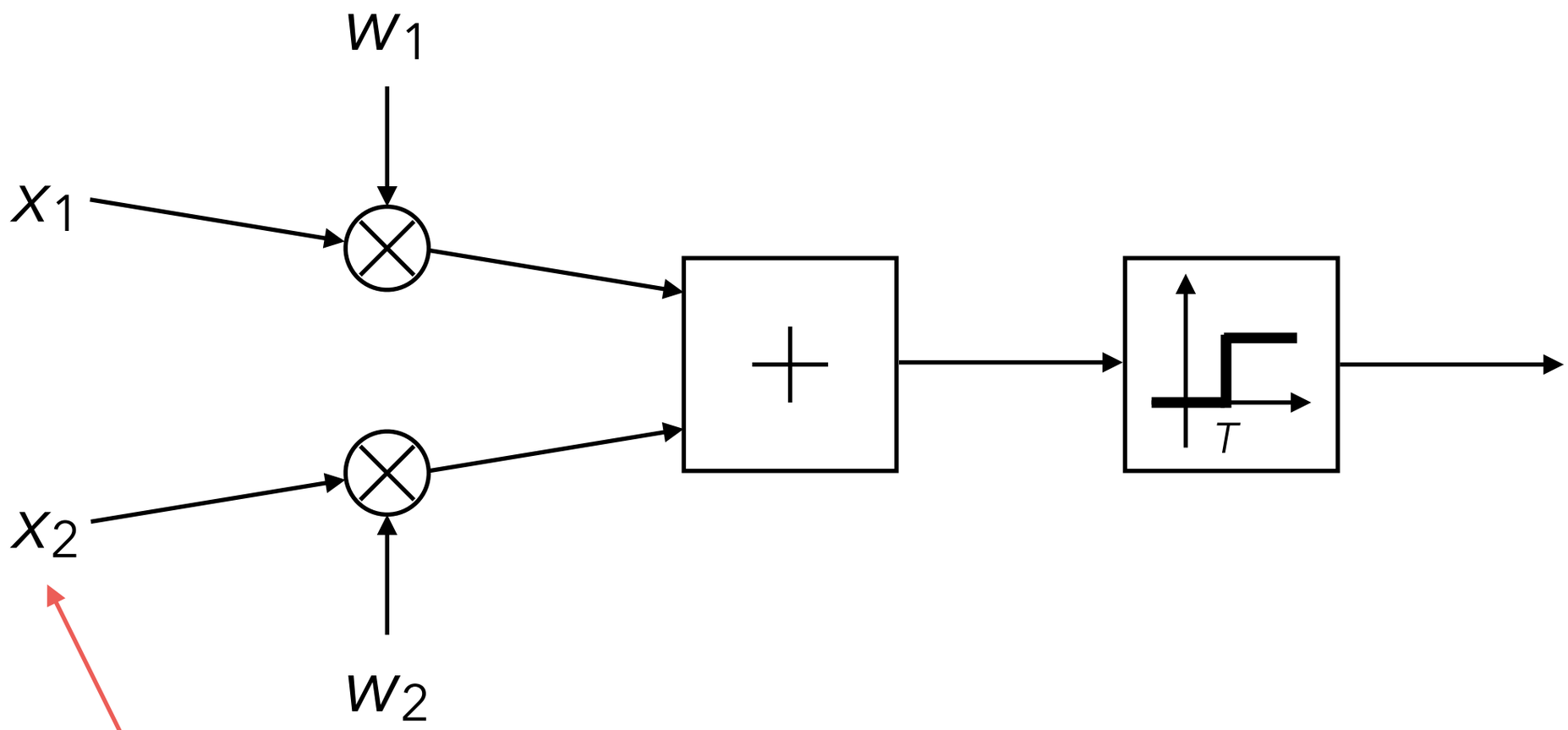
Eventually, if neuron's potential changes, it becomes excited

Actual neuron



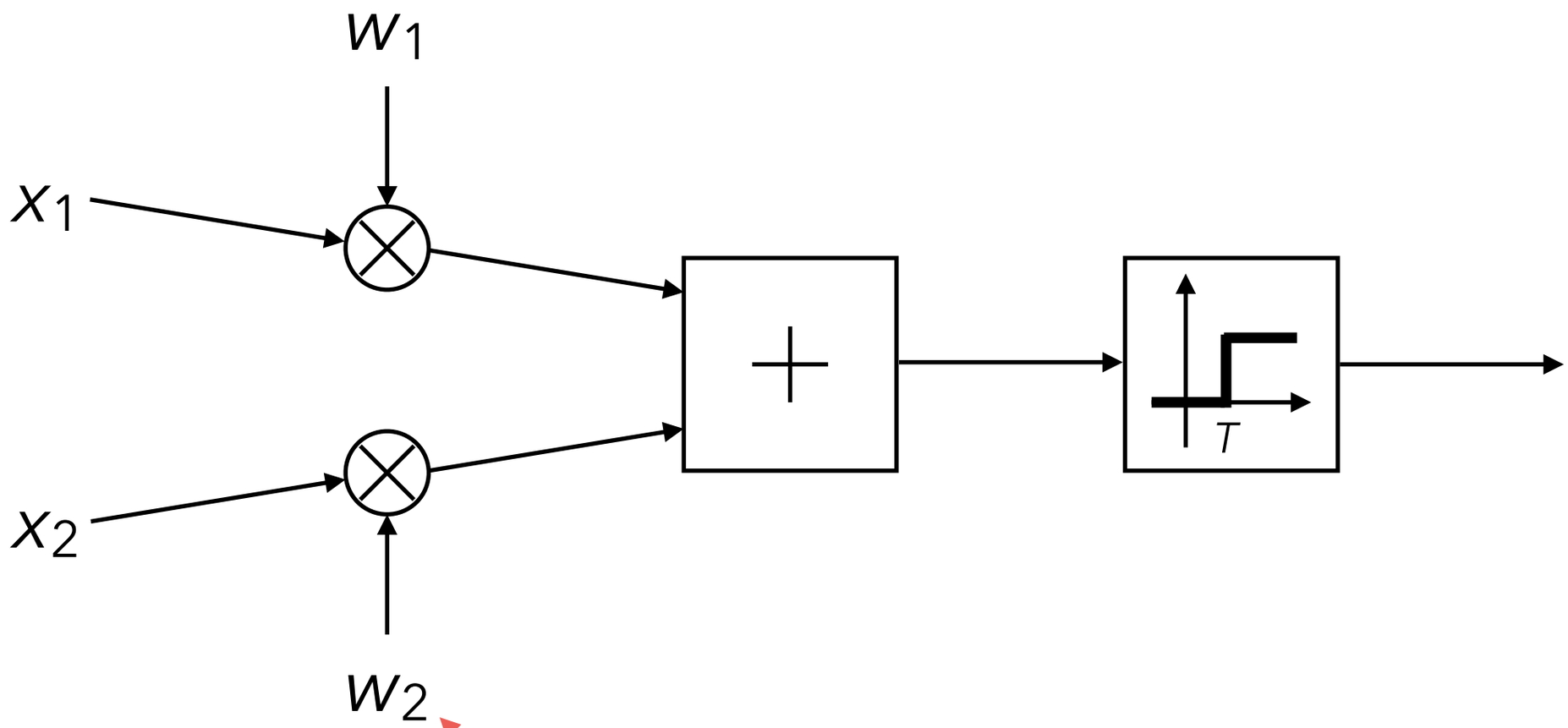
Potential is then propagated through axon to other cells

Artificial neuron



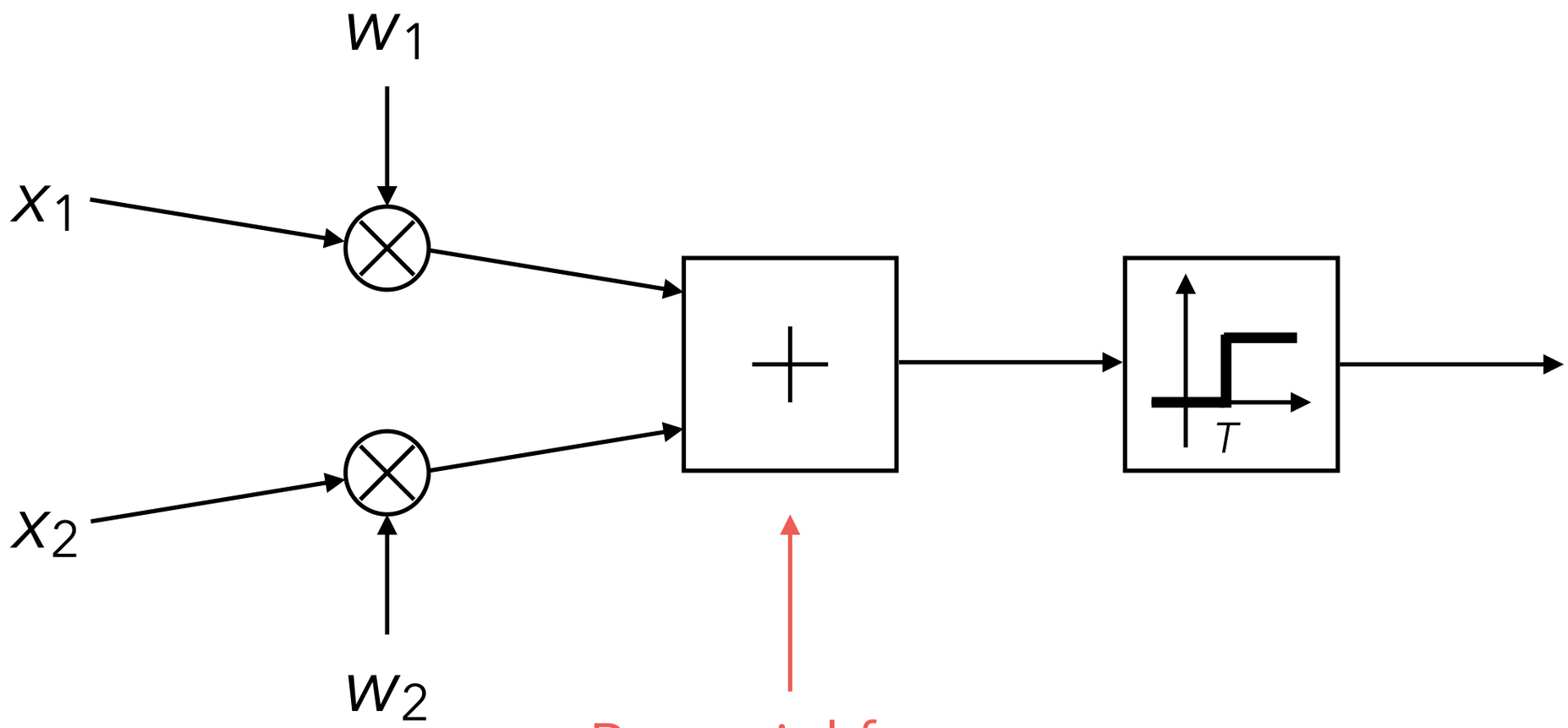
Input 0 - 1
(neurotransmitters)

Artificial neuron



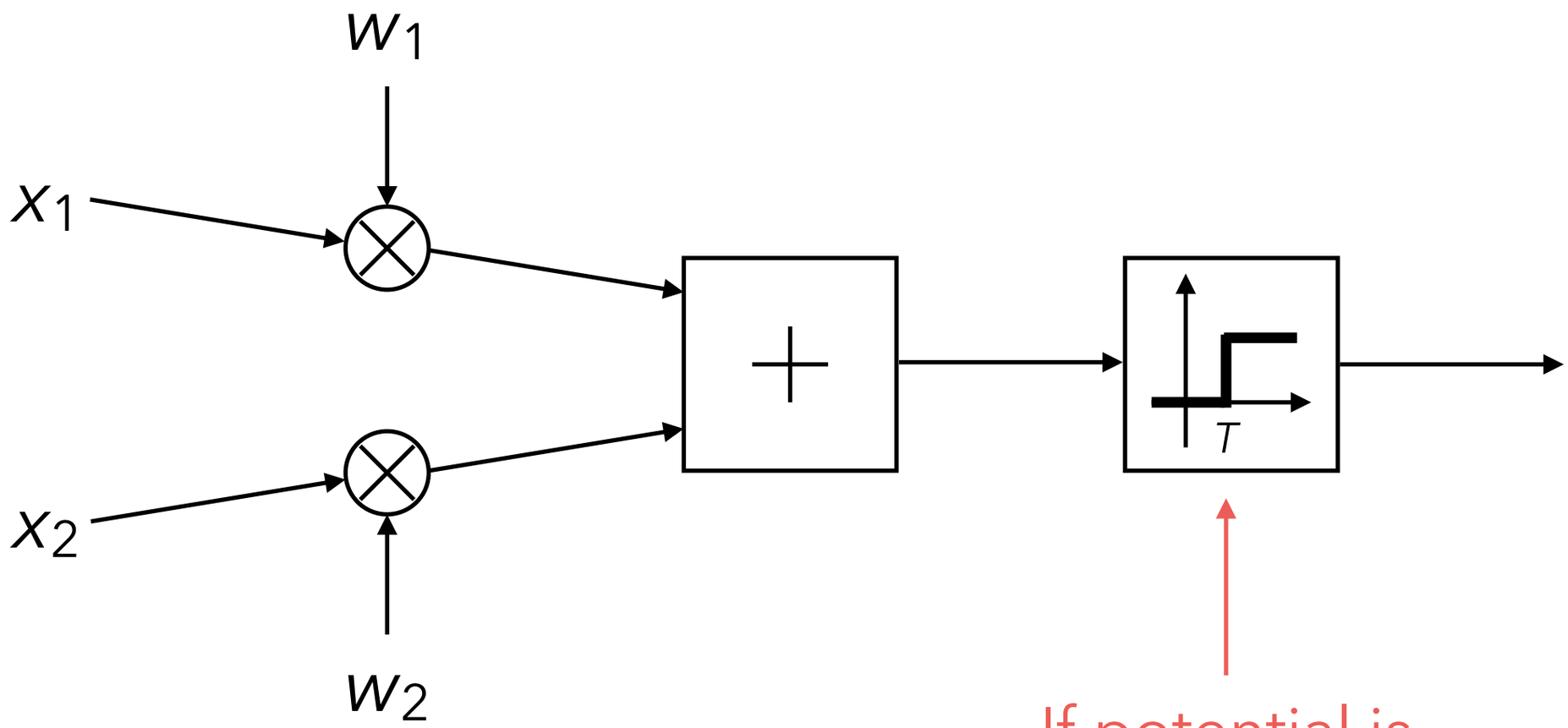
Weights
(intensity of
neurotransmitters)

Artificial neuron



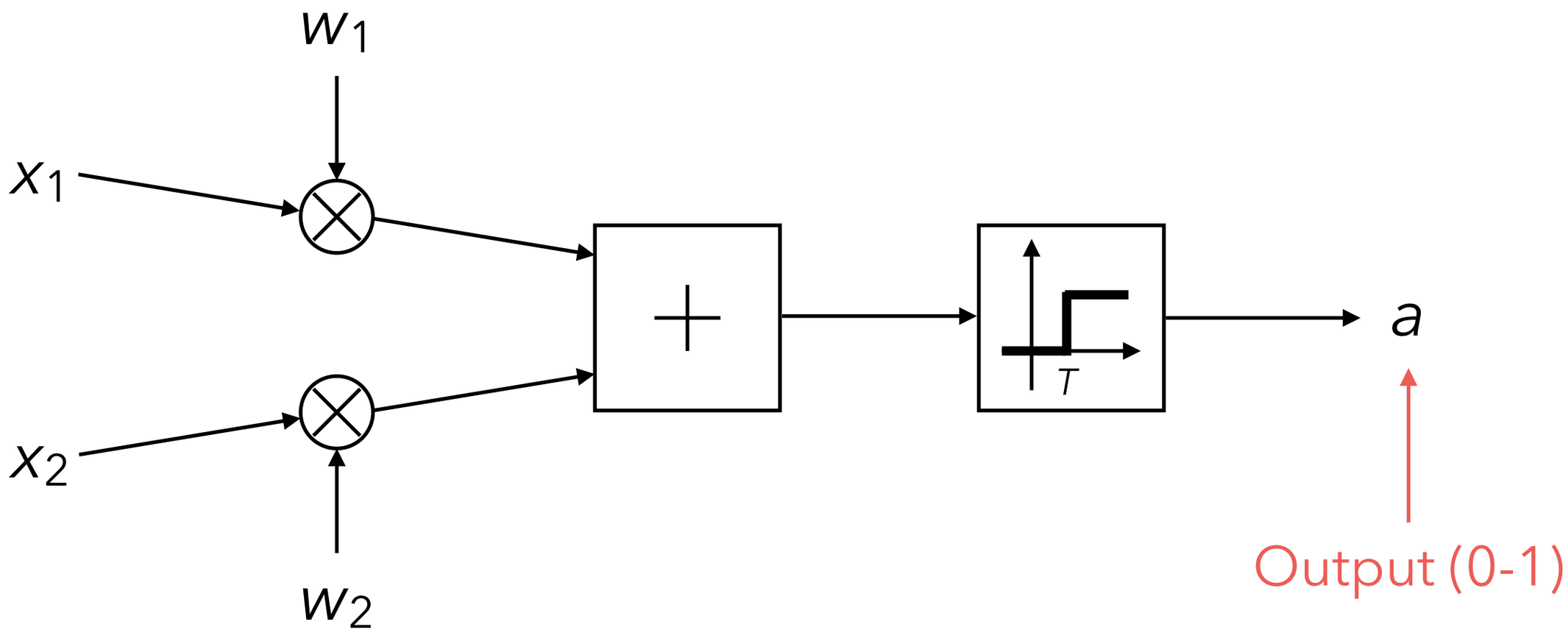
Potential from
all inputs
is added up

Artificial neuron

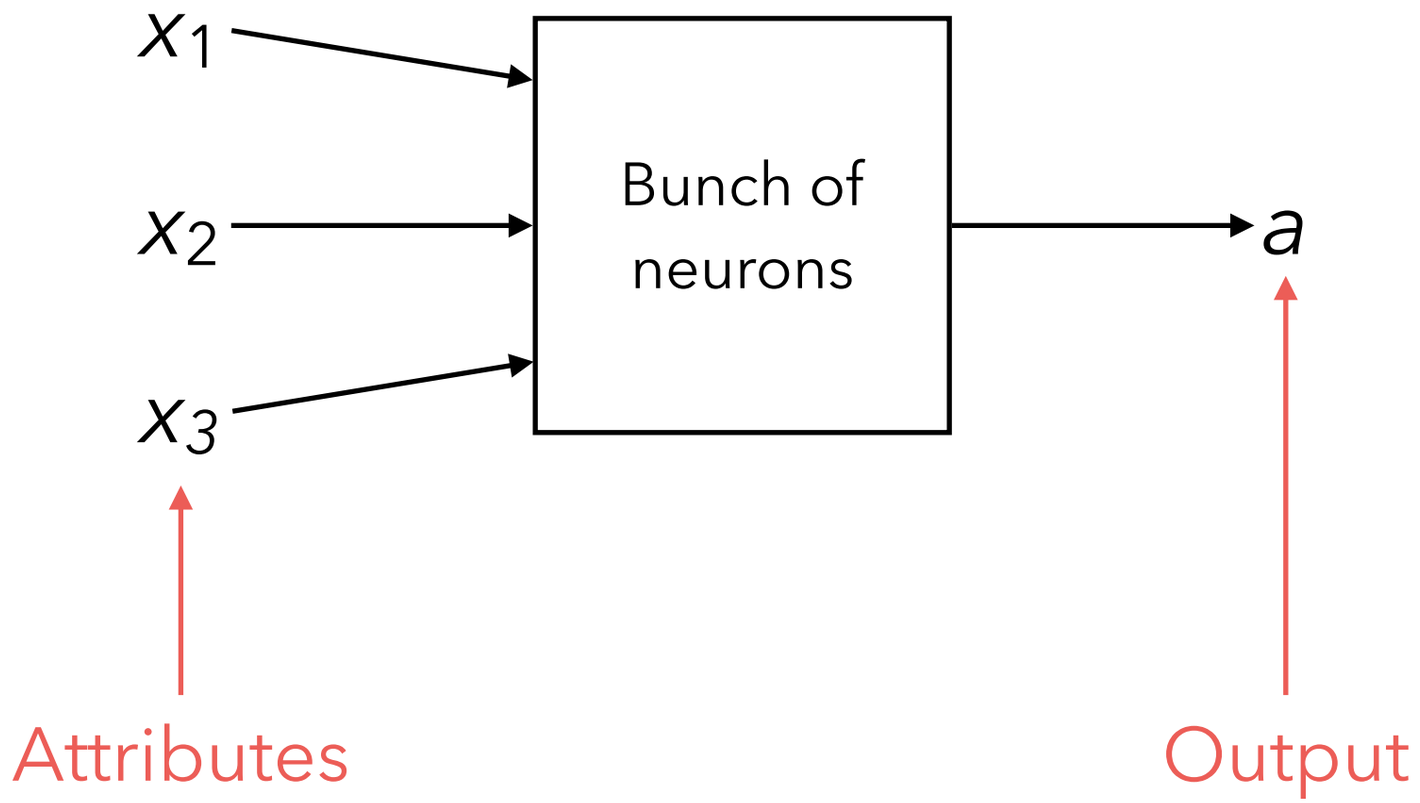


If potential is large enough, neuron “fires”

Artificial neuron

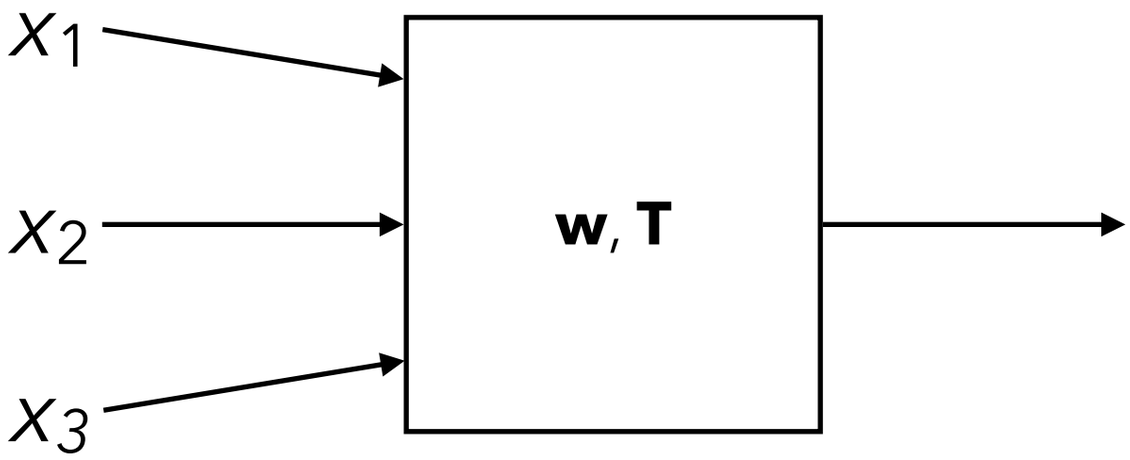


Neural network



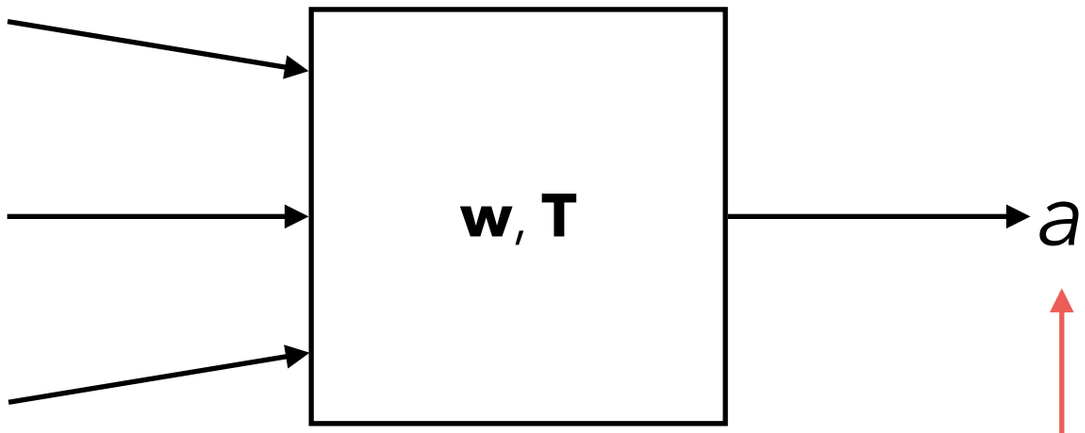
How do we learn with this?

Neural network



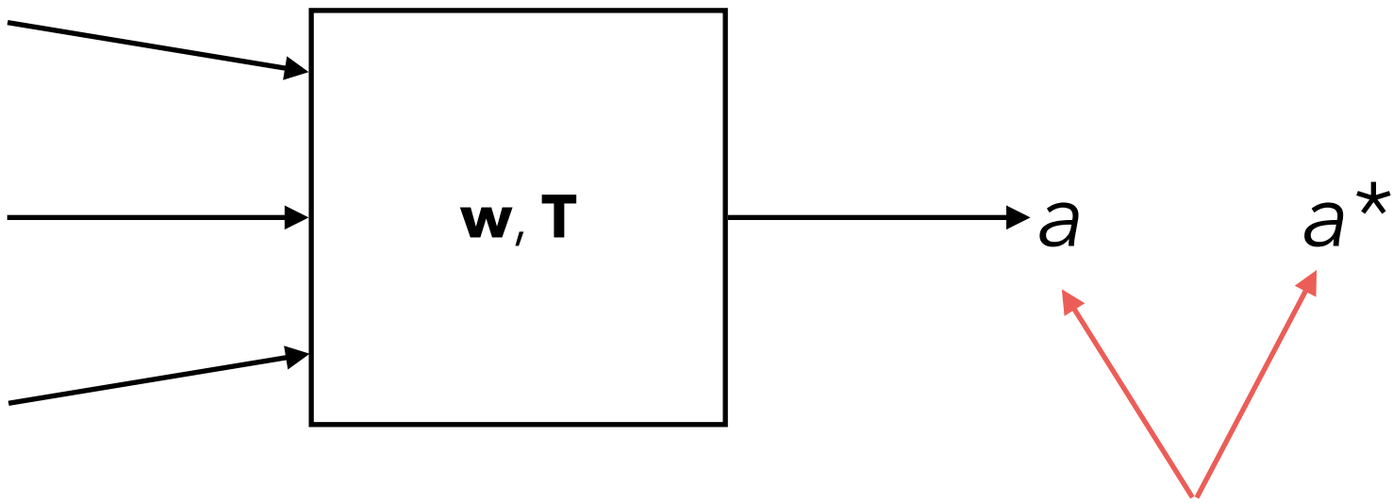
We provide example **x**

Neural network



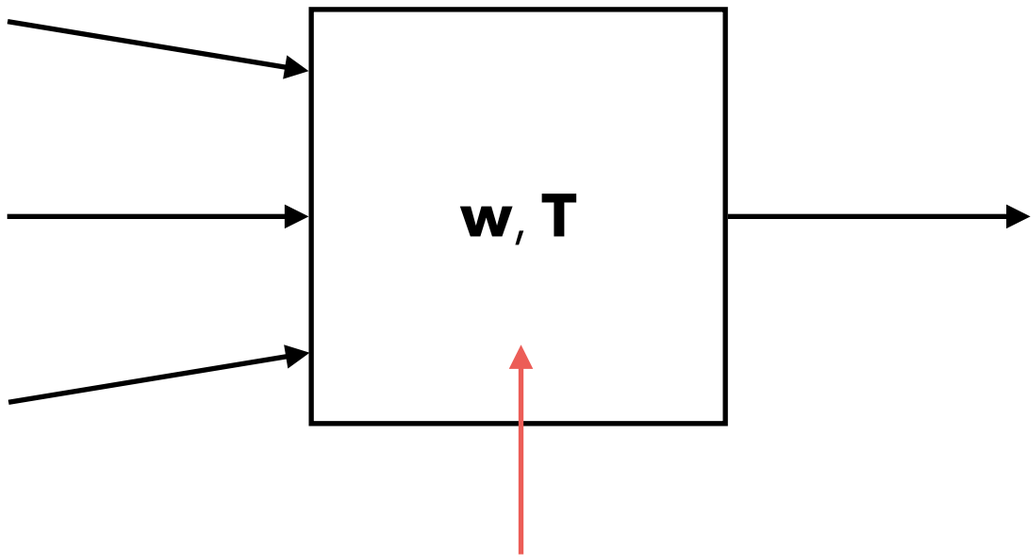
We compute the output, $a = f(\mathbf{x}, \mathbf{w}, \mathbf{T})$

Neural network



We compare the output with the desired value, *a**

Neural network



We adjust
the weights
and thresholds

Training neural networks

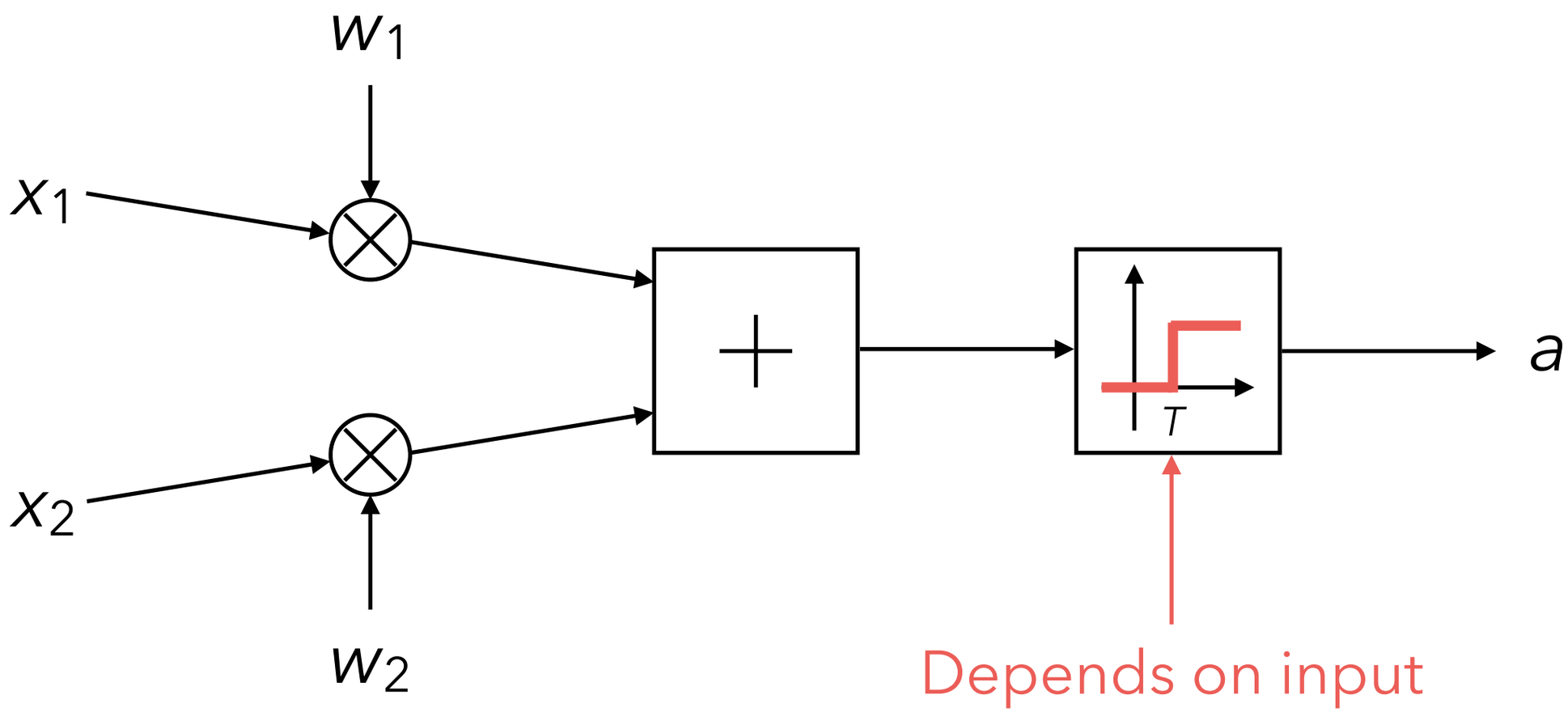
- How do we measure the performance?
  - For example, we can use the **negative log likelihood**:

$$\hat{L}_N(f) = -\frac{1}{N}\sum_{n=1}^{N} \log f(a_n \mid \boldsymbol{x}_n, \boldsymbol{w}, \boldsymbol{T})$$

- How do we adjust **w** and **T**?

... not easy. Why?

Discontinuity!



Depends on input is not continuous!

$$a = \mathbb{I}(\boldsymbol{w}^\top \boldsymbol{x} > T)$$

Step 1. Get rid of *T*

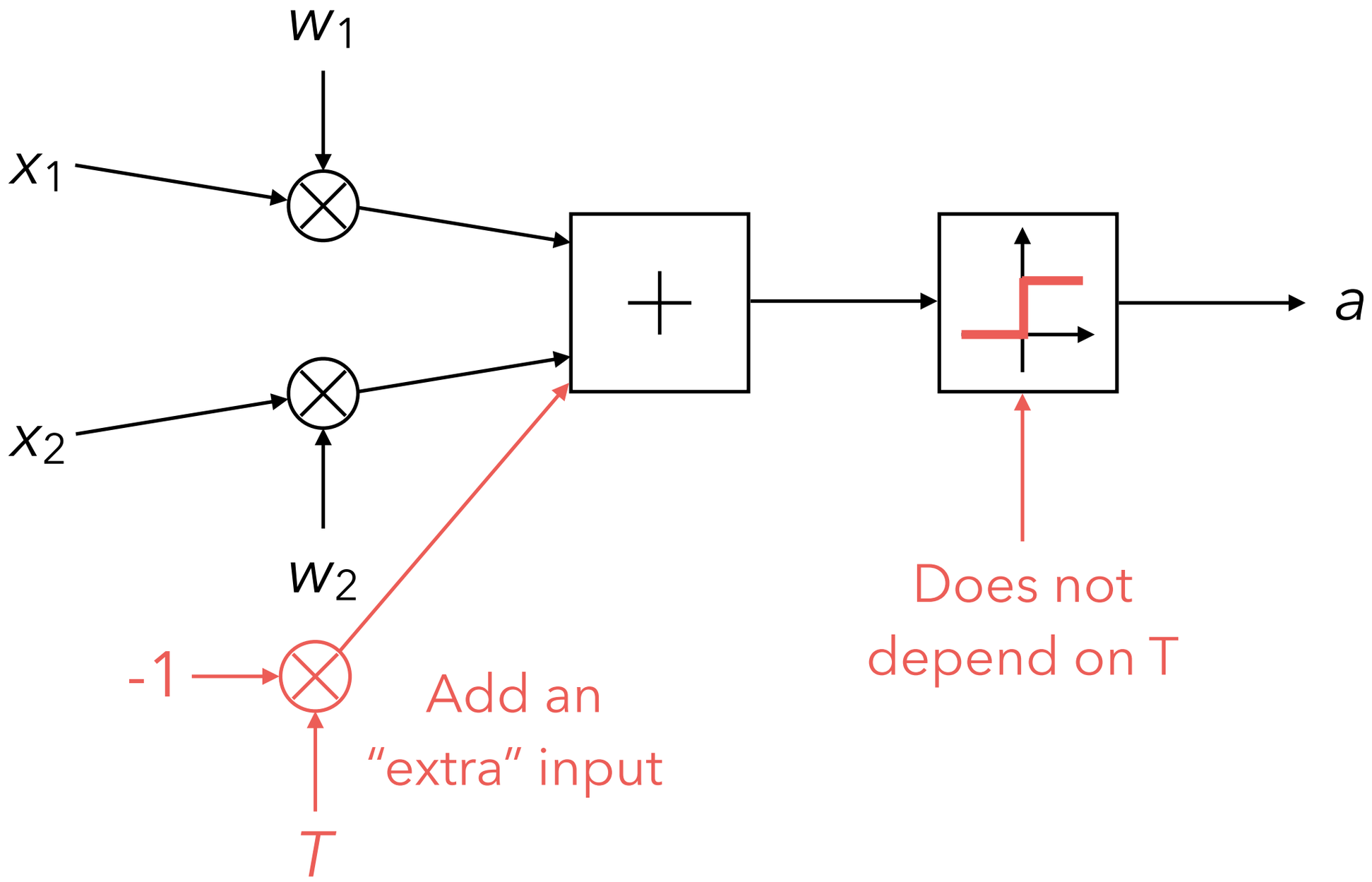
- How?
- The expression

$$a = \mathbb{I}(\boldsymbol{w}^\top \boldsymbol{x} > T)$$

is equivalent to

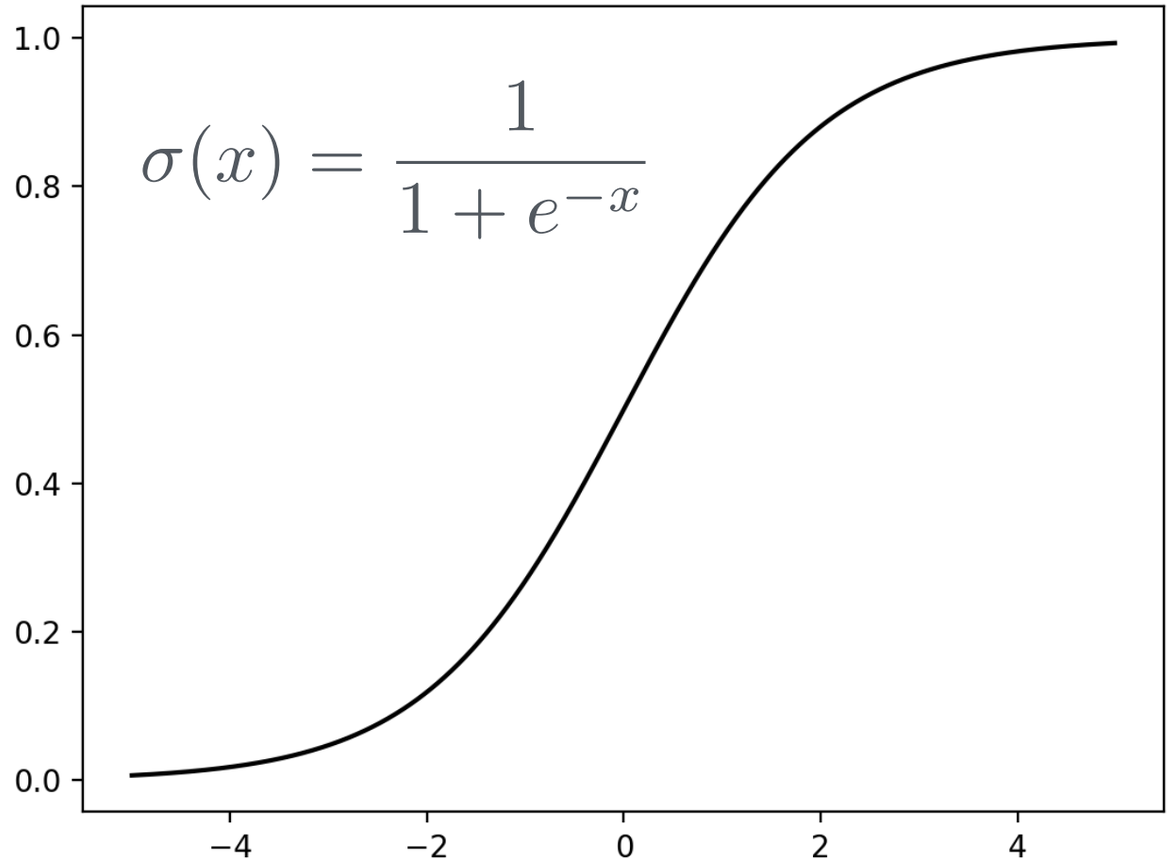
$$a = \mathbb{I}(\boldsymbol{w}^\top \boldsymbol{x} - T > 0)$$

Step 1. Get rid of *T*



Step 2. Smooth the threshold

- Instead of using the function ▯, we use a smooth version of it
- For example,



Training neural networks

- Now we can take the negative log-likelihood

$$\hat{L}_N(f) = -\frac{1}{N}\sum_{n=1}^{N} \log f(a_n \mid \boldsymbol{x}_n, \boldsymbol{w}, \boldsymbol{T})$$

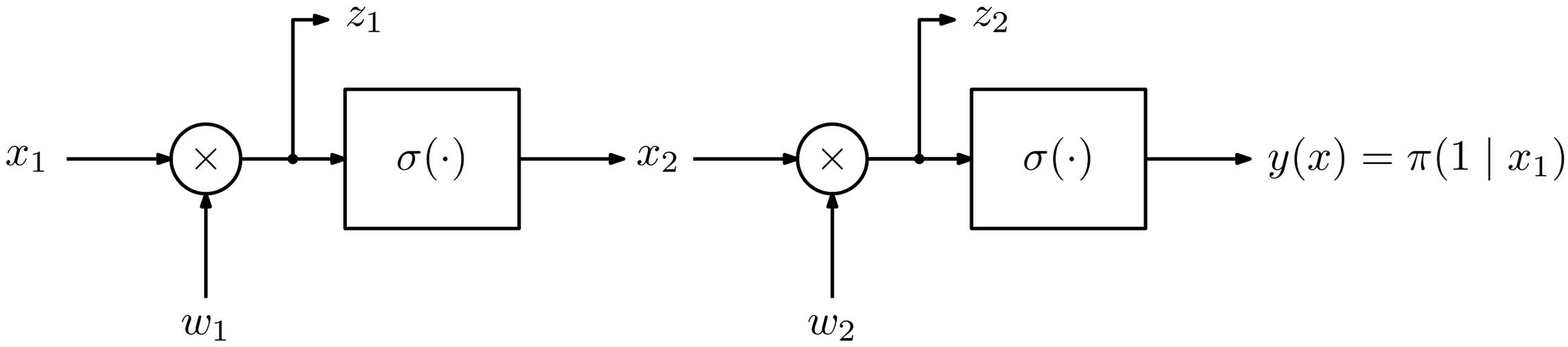
  and differentiate with respect to **w** and **T** (which is just another weight now)

- We adjust **w** and **T** using **gradient descent**

$$\boldsymbol{w} = \boldsymbol{w} - \alpha \nabla_{\boldsymbol{w}} \hat{L}_N(f)$$

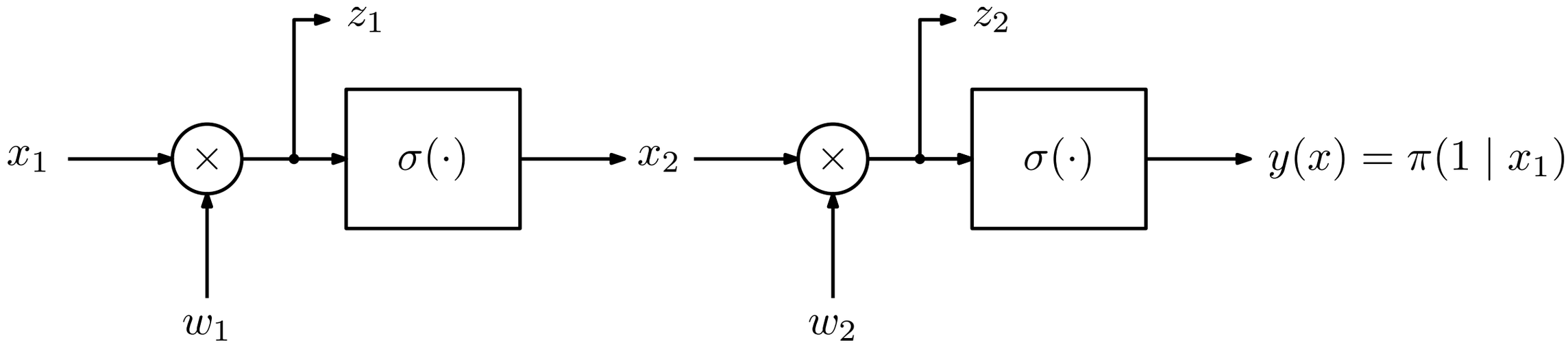
World's simplest net

World's simplest net



- How can we compute the dependence of the negative log-likelihood on w_1 and w_2?
  - We use the chain rule (of derivatives)

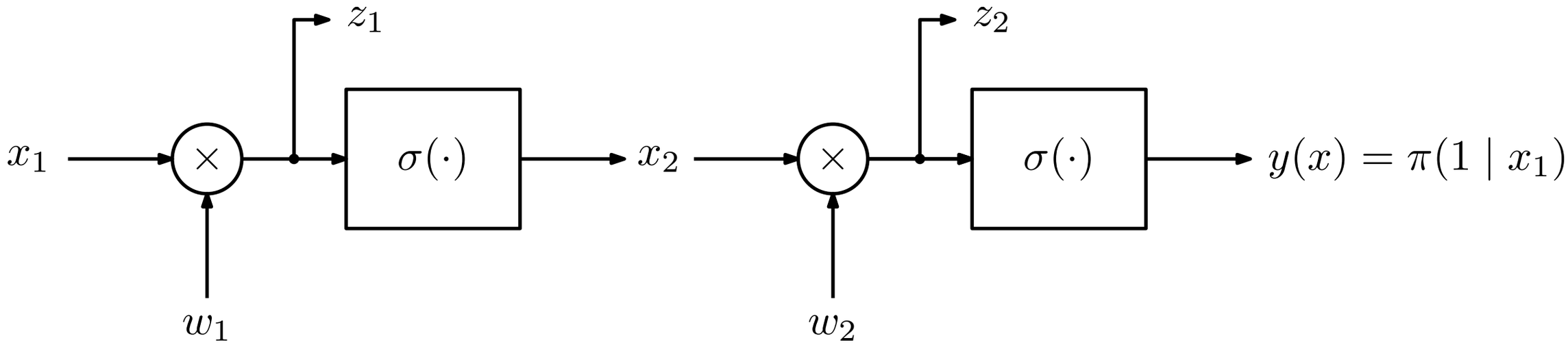
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial w_2}$$

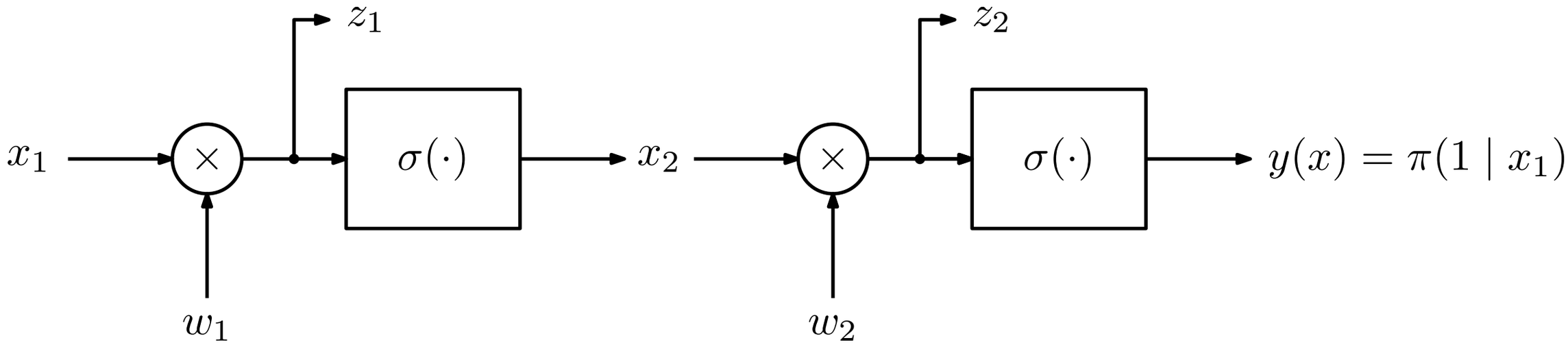
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}$$ ✓

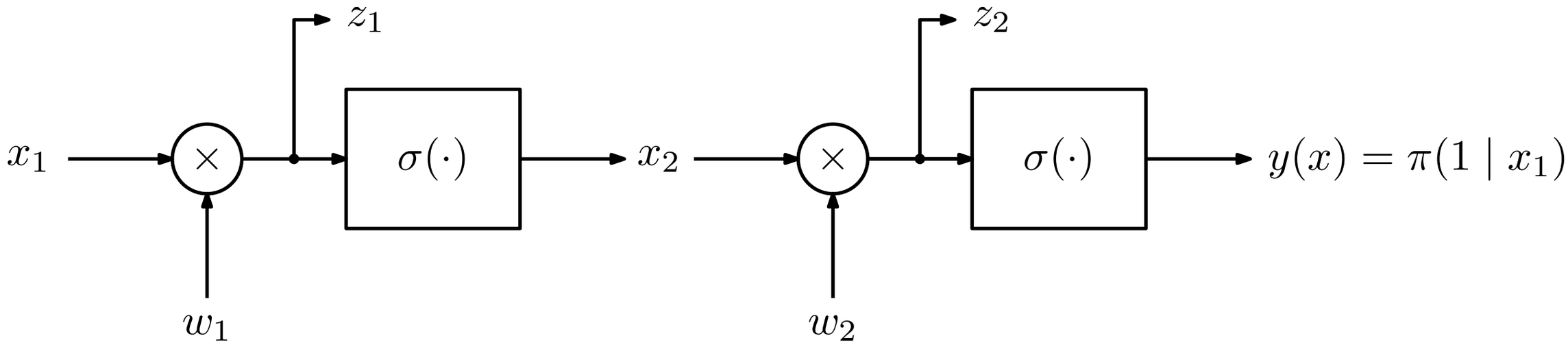
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}$$

World's simplest net

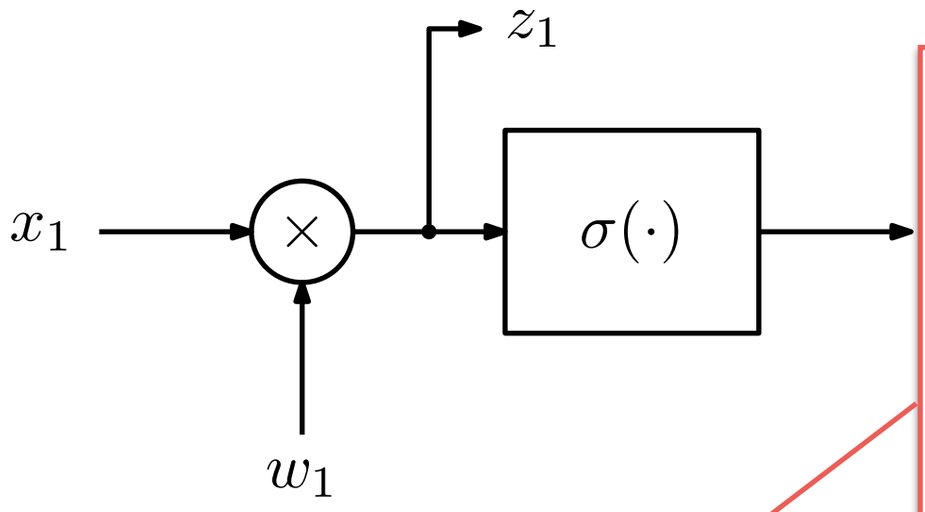


$$\frac{\partial \hat{L}_N}{\partial w_1} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \frac{\partial \hat{L}_N}{\partial y} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}$$

Now let's break these down!

World's simplest net



z_2

For a single sample (x, a),

$$\hat{L} = -a\log(y) - (1-a)\log(1-y)$$

so:

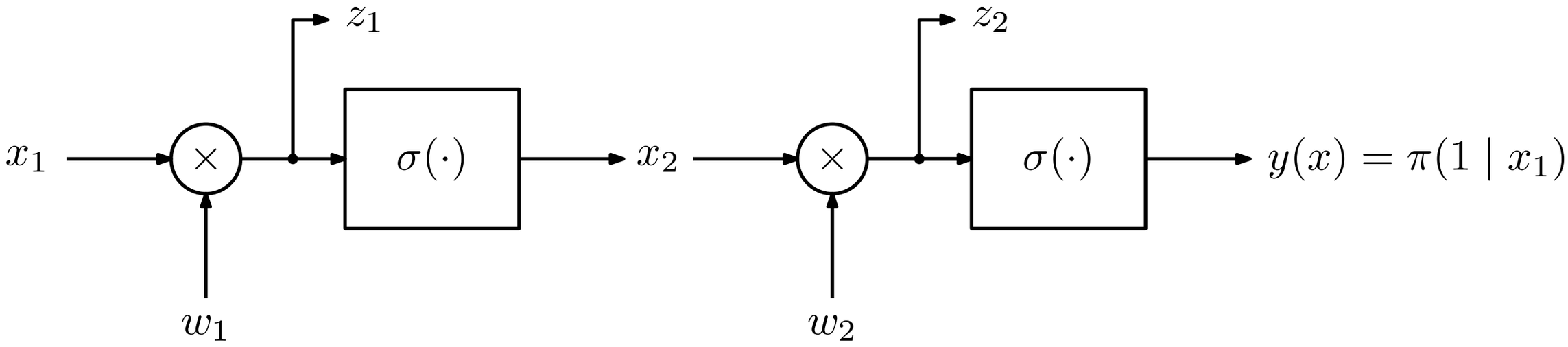
$$\frac{\partial \hat{L}}{\partial y} = \frac{-a}{y} + \frac{1-a}{1-y}$$

$(1 \mid x_1)$

$$\frac{\partial \hat{L}_N}{\partial w_1} = \boxed{\frac{\partial \hat{L}_N}{\partial y}} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \boxed{\frac{\partial \hat{L}_N}{\partial y}} \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}$$

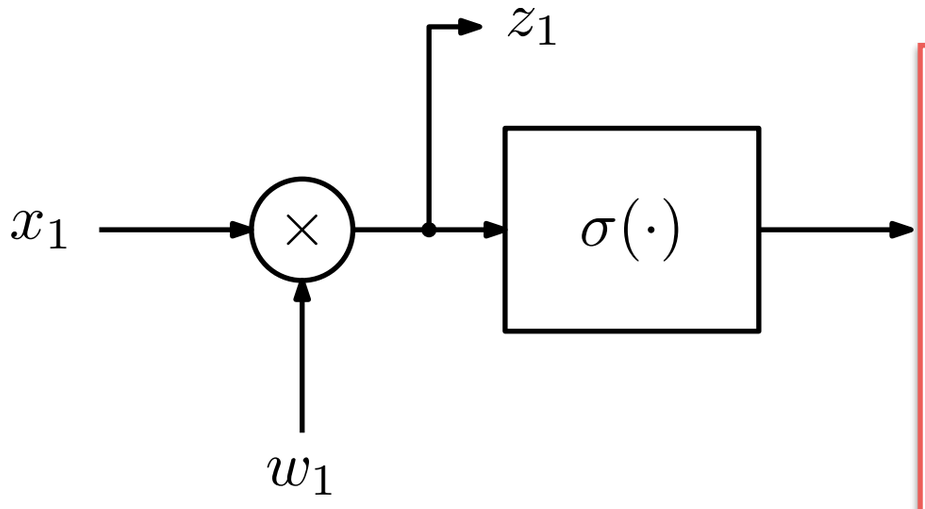
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = \left(\frac{-a}{y} + \frac{1-a}{1-y} \right) \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \left(\frac{-a}{y} + \frac{1-a}{1-y} \right) \times \frac{\partial y}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}$$

World's simplest net



z_2

We have

$$y = \sigma(z_2) = \frac{1}{1 + \exp(-z_2)}$$

so:

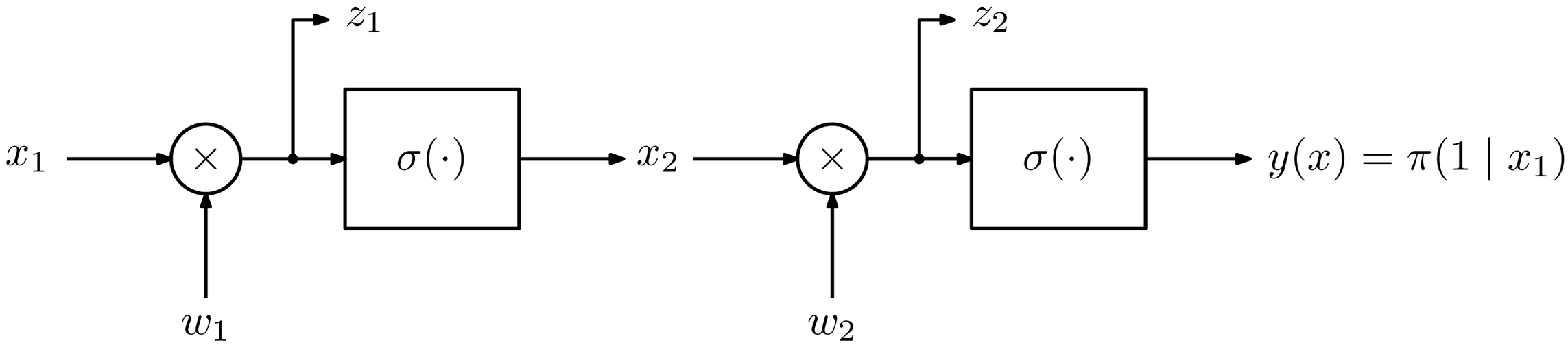
$$\frac{\partial y}{\partial z_2} = y(1 - y)$$

$(1 \mid x_1)$

$$\frac{\partial \hat{L}_N}{\partial w_1} = \left(\frac{-a}{y} + \frac{1-a}{1-y} \right) \times \boxed{\frac{\partial y}{\partial z_2}} \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \left(\frac{-a}{y} + \frac{1-a}{1-y} \right) \times \boxed{\frac{\partial y}{\partial z_2}} \times \frac{\partial z_2}{\partial w_2}$$

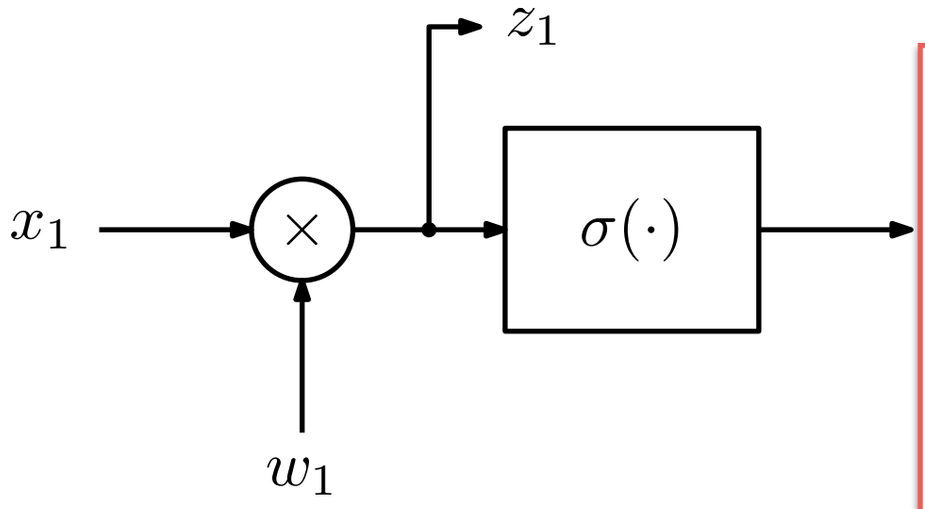
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) \times \frac{\partial z_2}{\partial w_2}$$

World's simplest net



z_2

This one is easy:

$$z_2 = w_2 x_2$$

so:

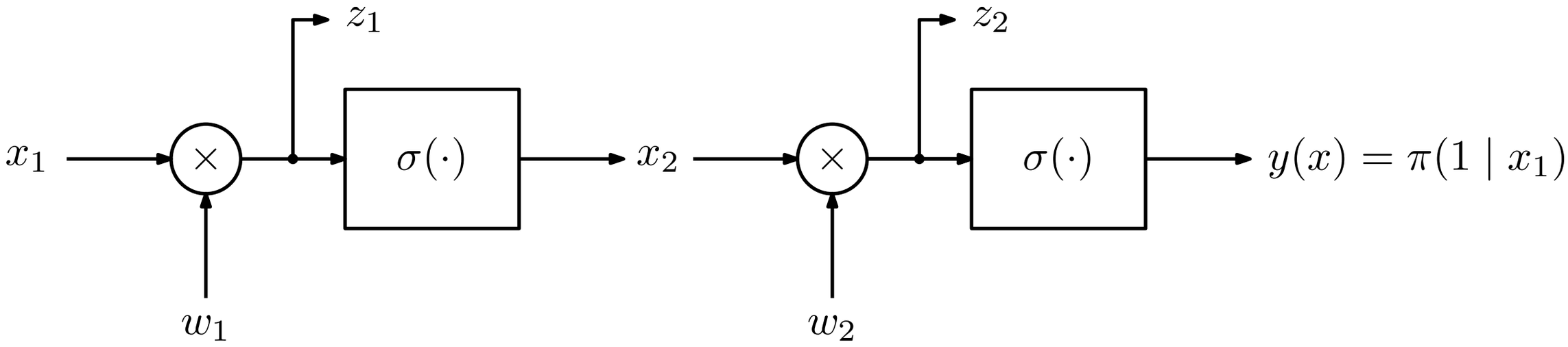
$$\frac{\partial z_2}{\partial w_2} = x_2$$

$(1 \mid x_1)$

$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) \times \boxed{\frac{\partial z_2}{\partial w_2}}$$

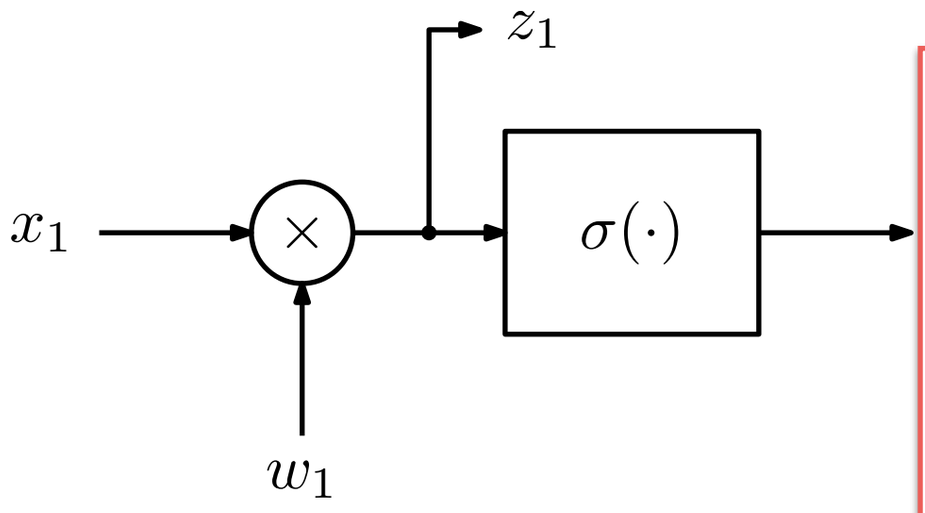
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) \times \frac{\partial z_2}{\partial x_2} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

World's simplest net



z_2

This one is also easy:

$$z_2 = w_2 x_2$$

so:

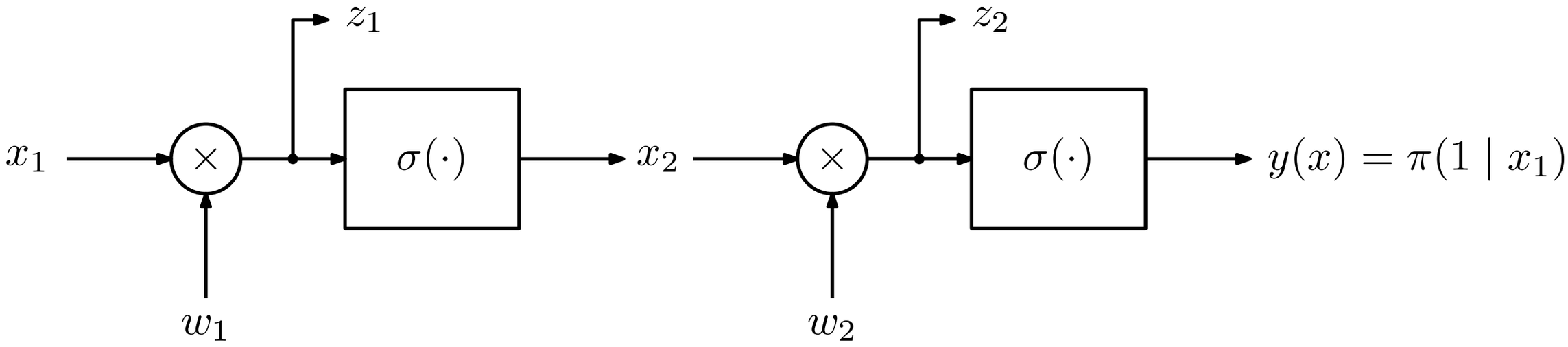
$$\frac{\partial z_2}{\partial x_2} = w_2$$

$(1 \mid x_1)$

$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) \times \boxed{\frac{\partial z_2}{\partial x_2}} \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

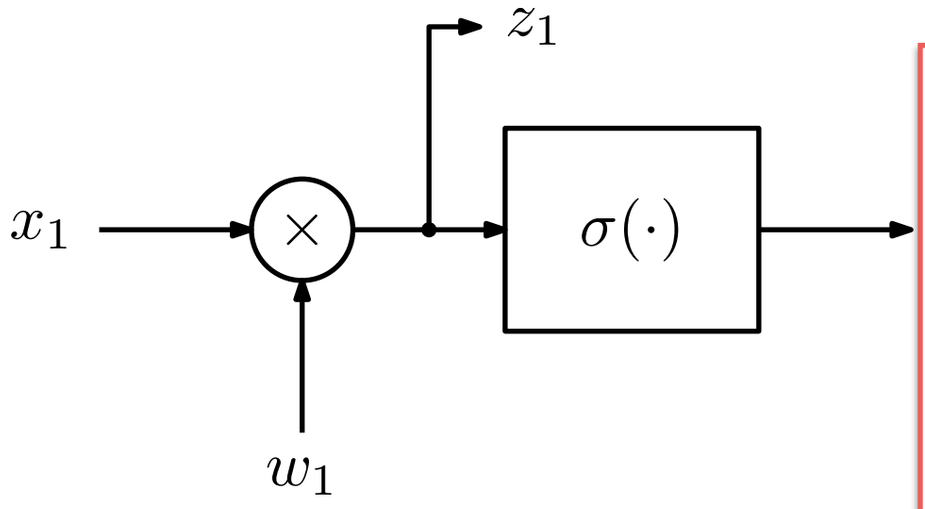
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) w_2 \times \frac{\partial x_2}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

World's simplest net



z_2

This is similar to a previous one:

$$x_2 = \sigma(z_1) = \frac{1}{1 + \exp(-z_1)}$$

so:

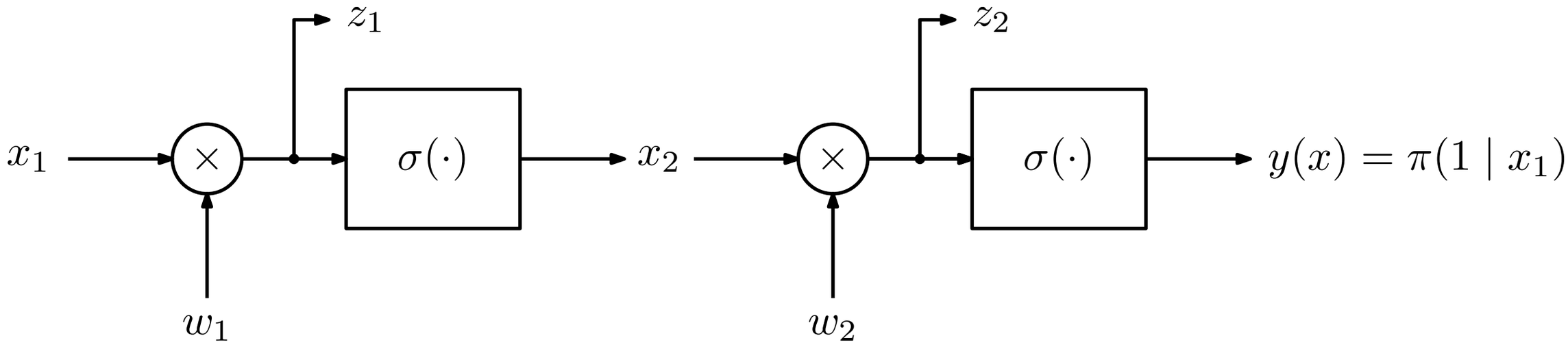
$$\frac{\partial x_2}{\partial z_1} = x_2(1 - x_2)$$

$(1 \mid x_1)$

$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a)w_2 \times \boxed{\frac{\partial x_2}{\partial z_1}} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a)x_2$$

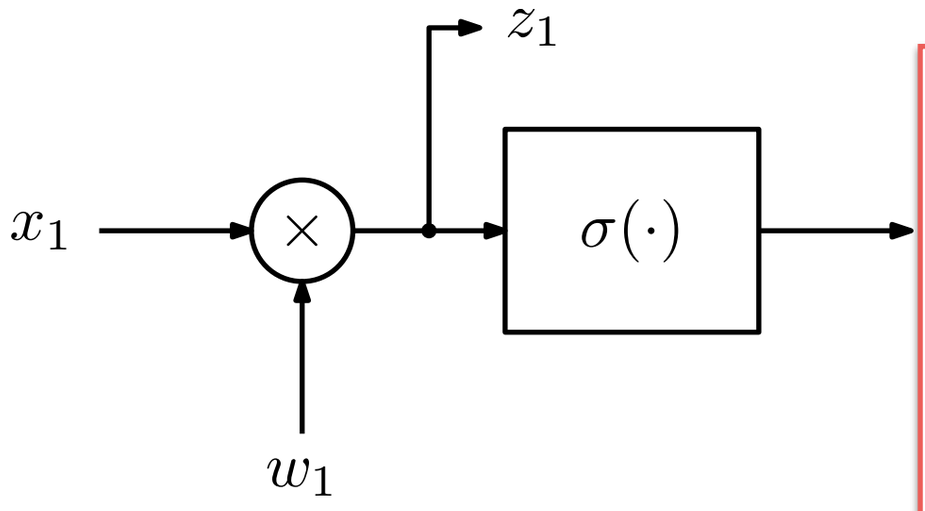
World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) w_2 x_2 (1 - x_2) \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

World's simplest net



z_2

And finally

$$z_1 = w_1 x_1$$

so:

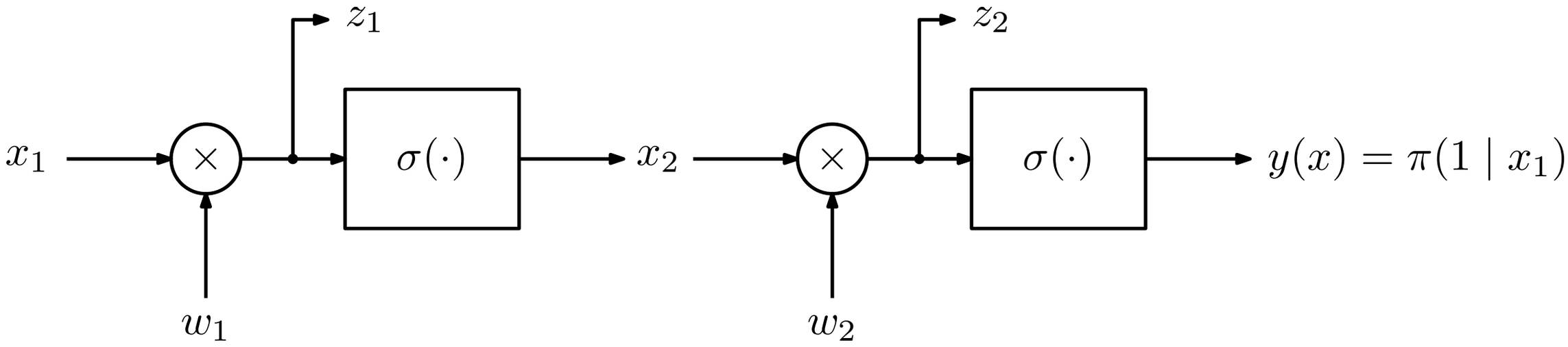
$$\frac{\partial z_1}{\partial w_1} = x_1$$

$(1 \mid x_1)$

$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) w_2 x_2 (1 - x_2) \times \boxed{\frac{\partial z_1}{\partial w_1}}$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

World's simplest net

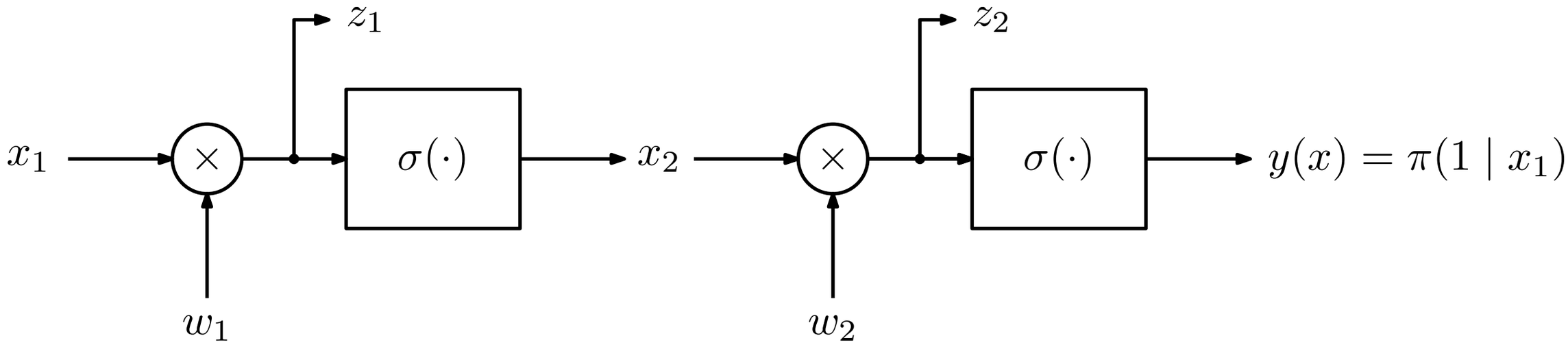


$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) w_2 x_2 (1 - x_2) x_1$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

Looking more carefully...

World's simplest net

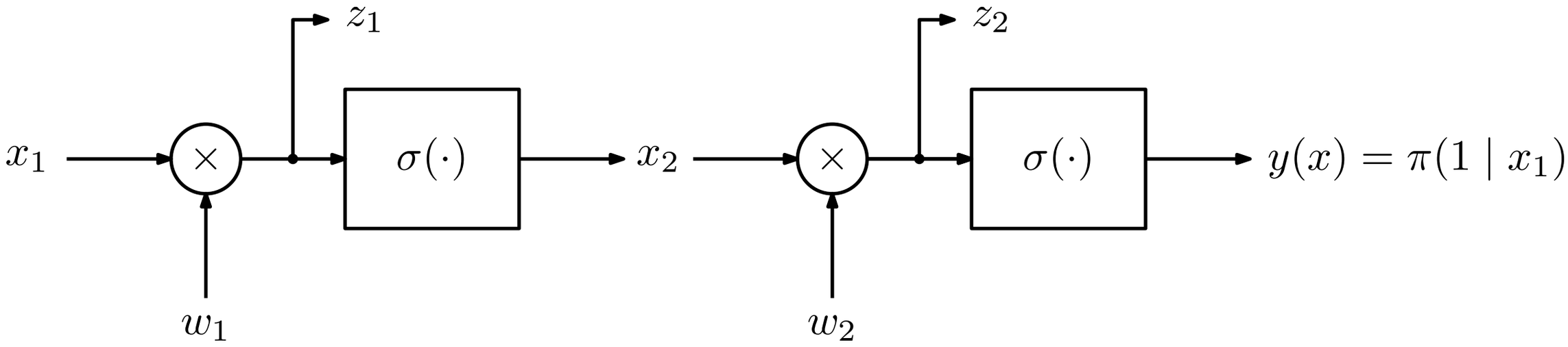


$$\frac{\partial \hat{L}_N}{\partial w_1} = (y - a) w_2 x_2 (1 - x_2) x_1$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y - a) x_2$$

The derivative with respect to w_i is *something* × x_i

World's simplest net

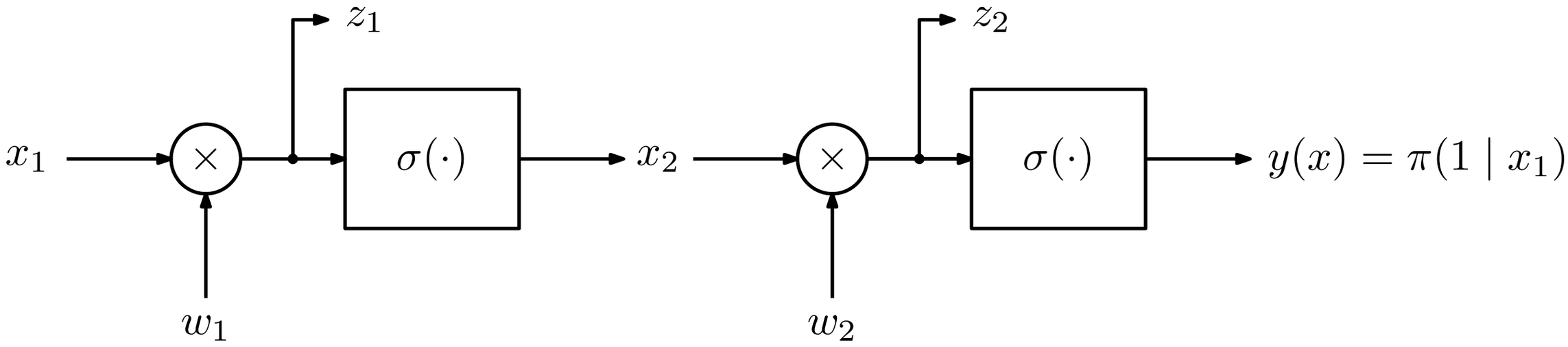


$$\frac{\partial \hat{L}_N}{\partial w_1} = \underbrace{(y-a)w_2x_2(1-x_2)}_{\delta_1}x_1$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \underbrace{(y-a)}_{\delta_2}x_2$$

We will call that something δ_i

World's simplest net

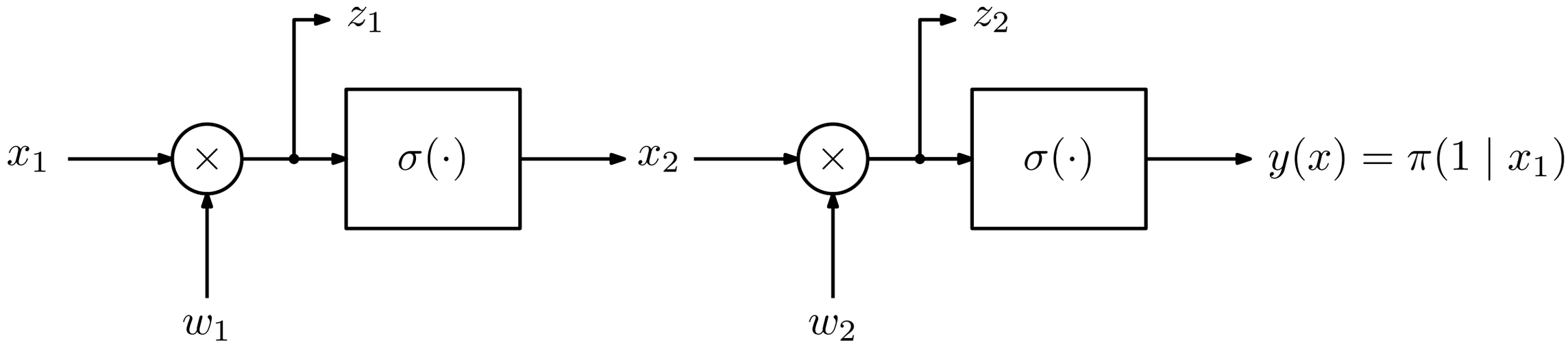


$$\frac{\partial \hat{L}_N}{\partial w_1} = \underbrace{(y-a)}_{\delta_2} w_2 x_2 (1-x_2) x_1$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = (y-a) x_2$$

We can compute δ_1 from δ_2

World's simplest net



$$\frac{\partial \hat{L}_N}{\partial w_1} = \delta_1 x_1$$

$$\frac{\partial \hat{L}_N}{\partial w_2} = \delta_2 x_2$$

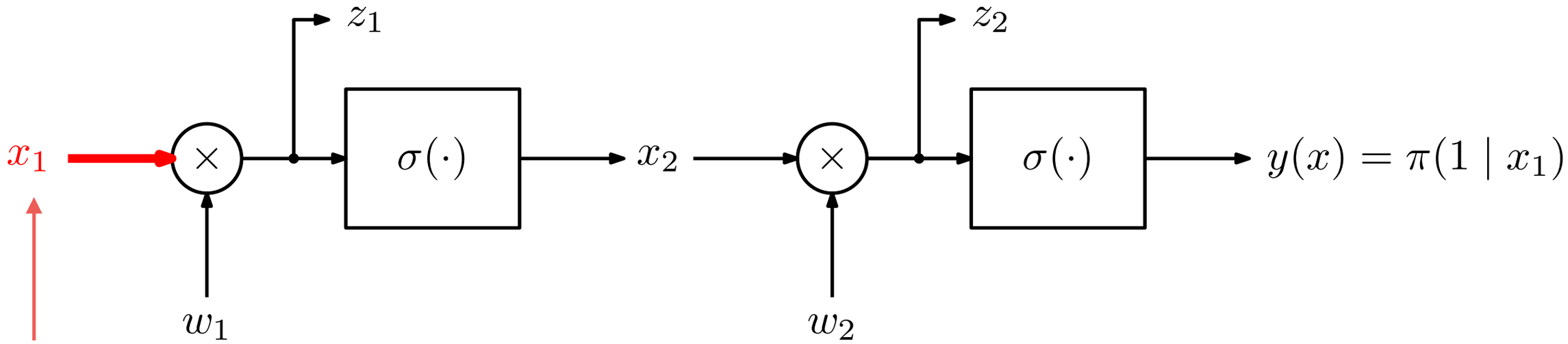
$$\delta_i = \frac{\partial \hat{L}_N}{\partial z_i}$$

$$\delta_{i-1} = \delta_i w_i \sigma'(x_i)$$

TÉCNICO LISBOA

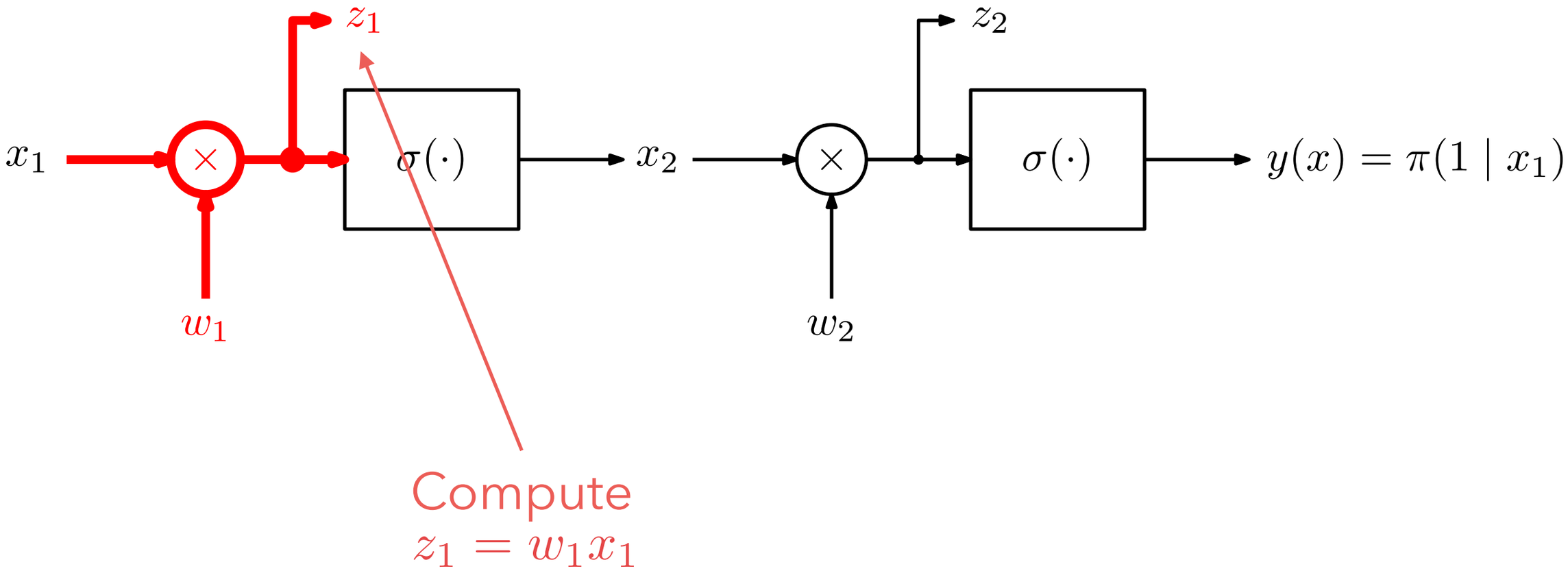
Let’s repeat, now with pictures…

Forward propagation

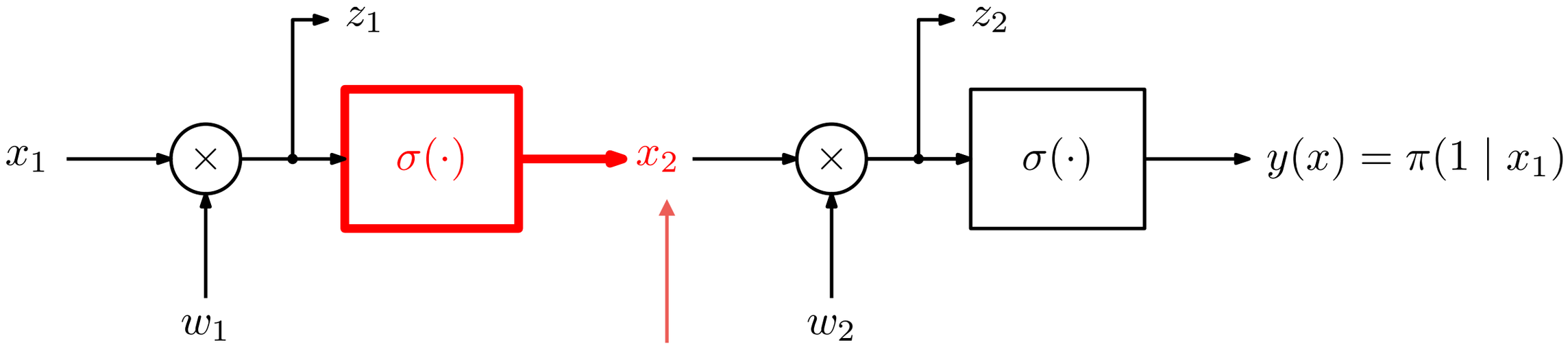


Given

Forward propagation



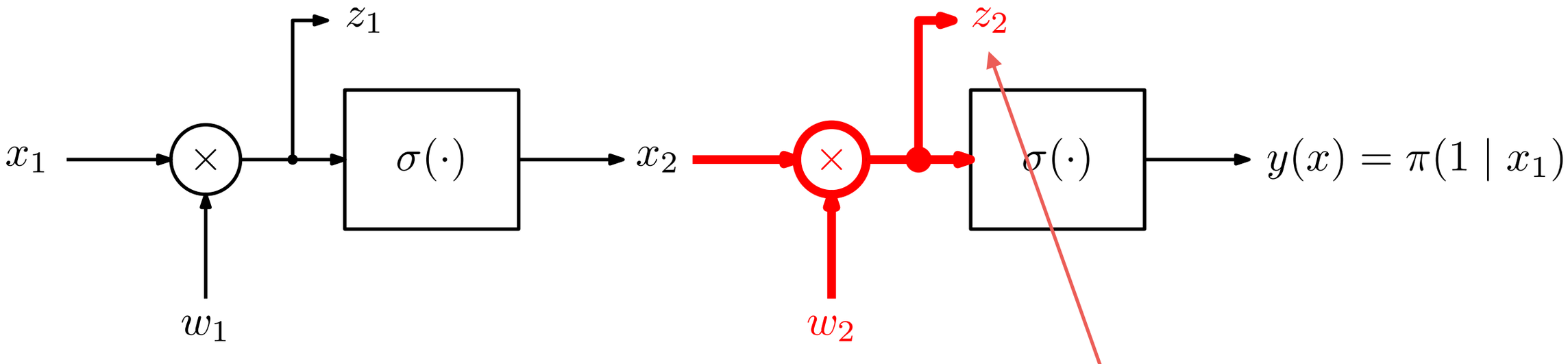
Forward propagation



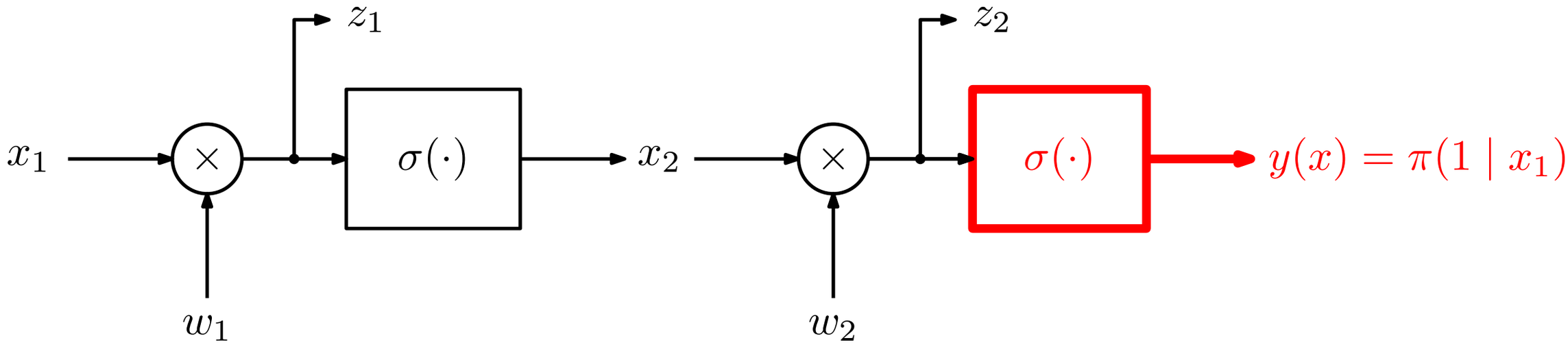
Compute

$$x_2 = \sigma(z_1)$$

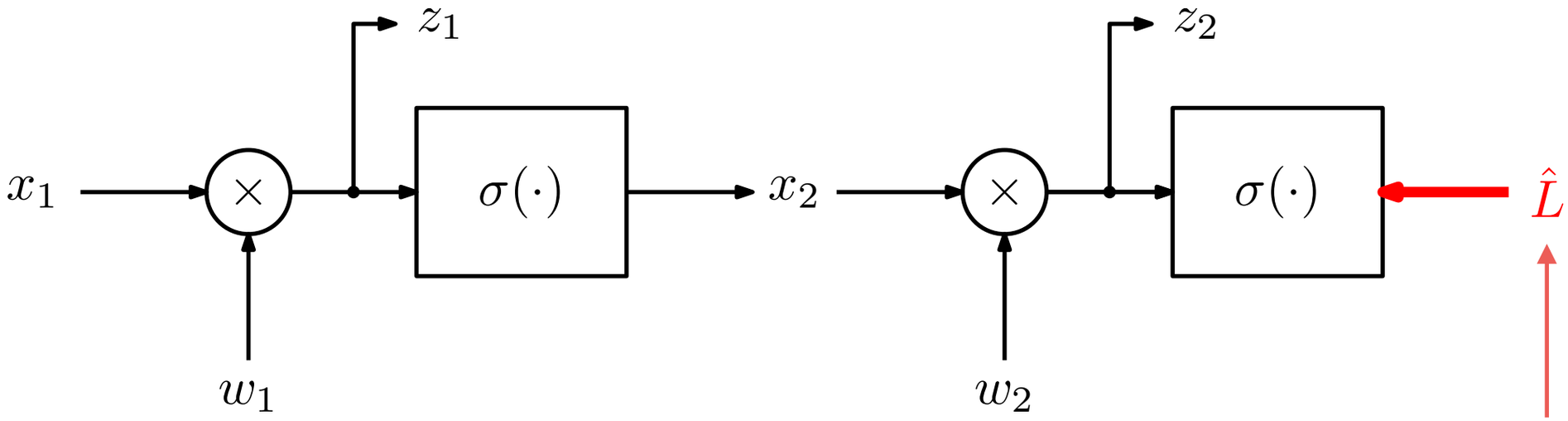
Forward propagation



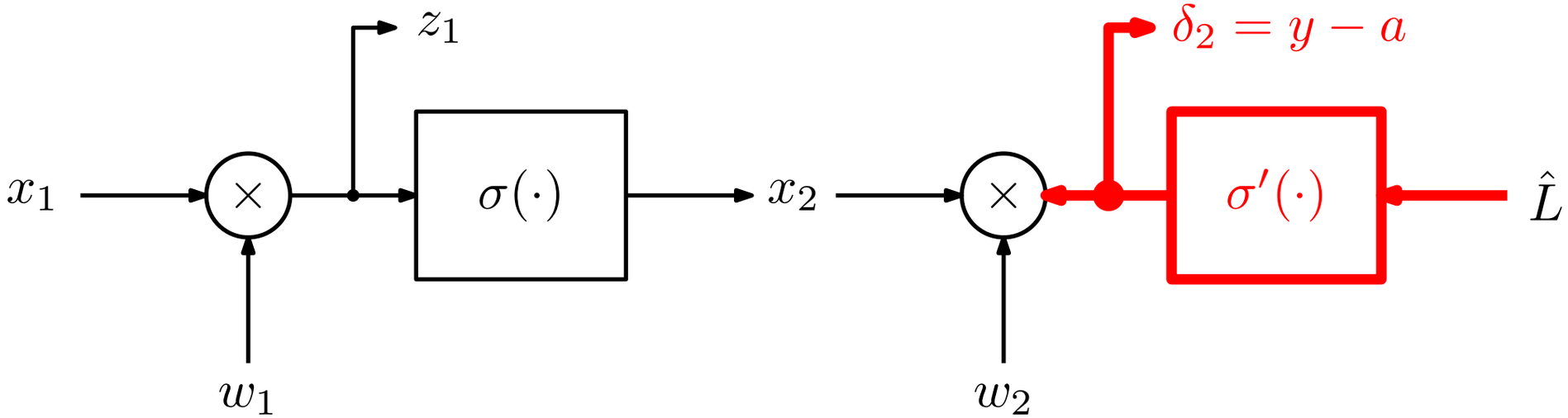
Forward propagation



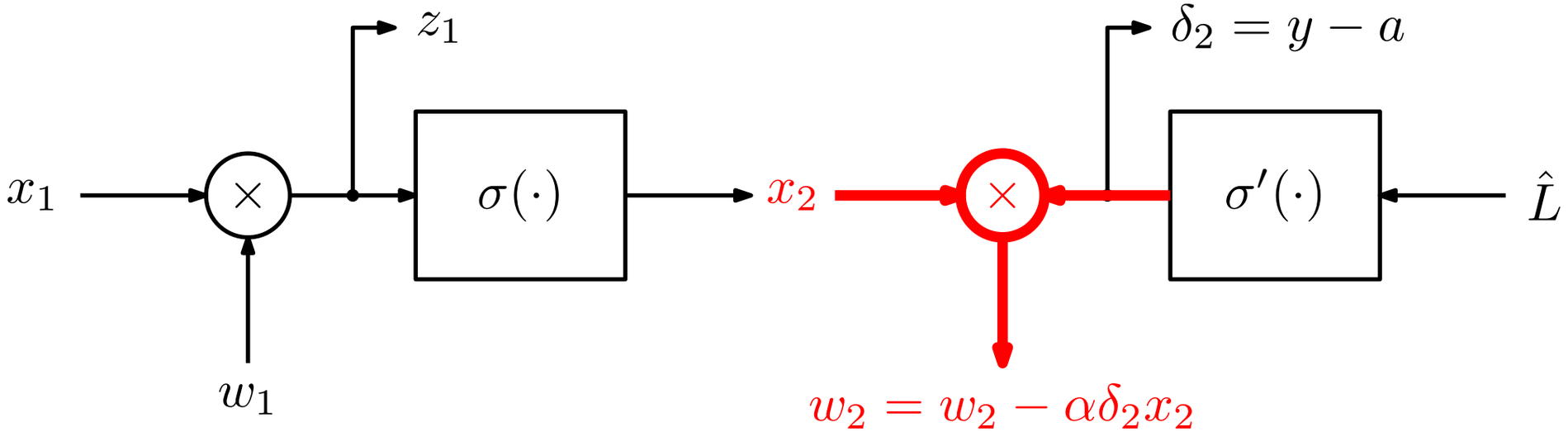
Backward propagation



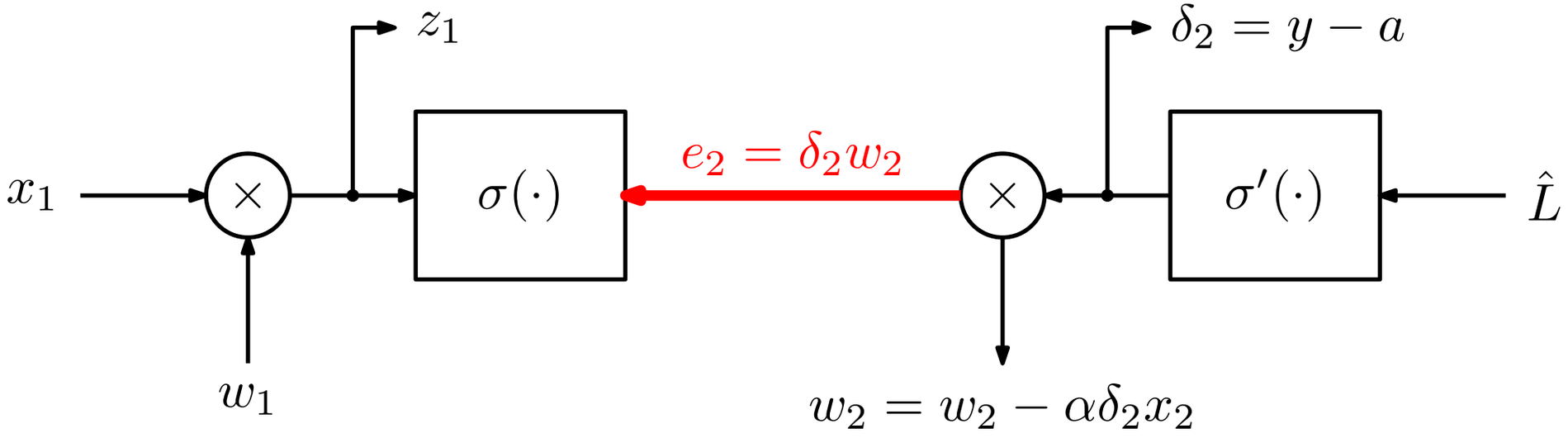
Backward propagation



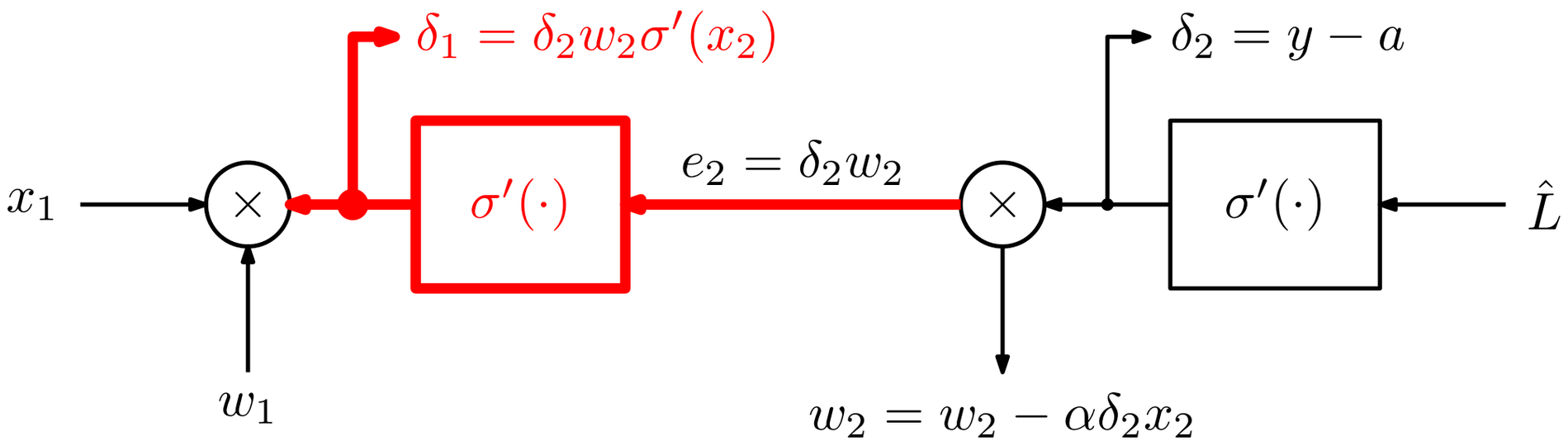
Backward propagation



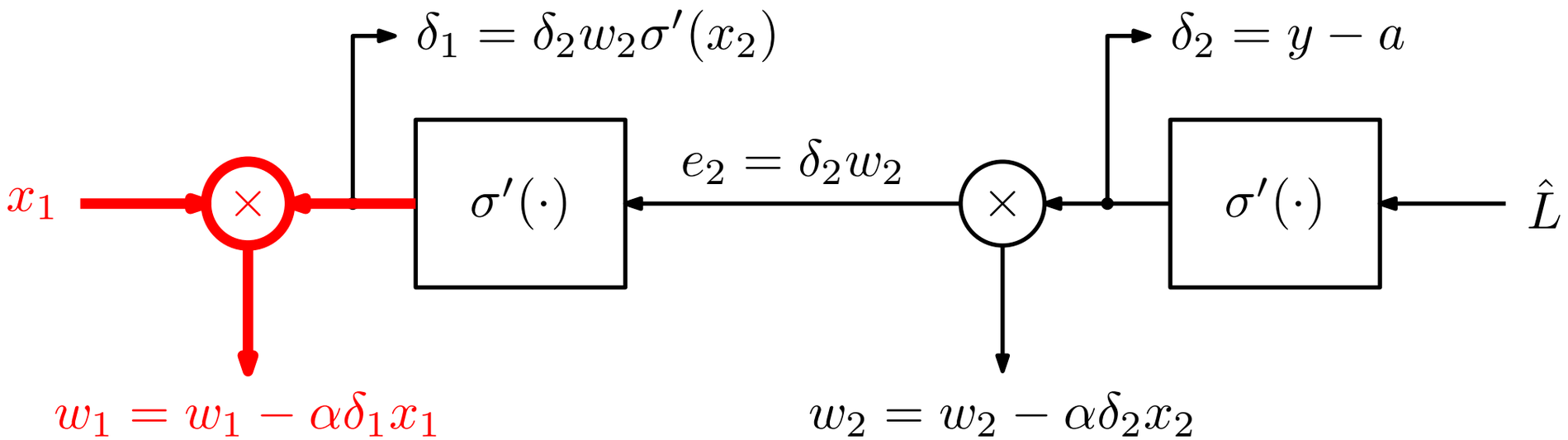
Backward propagation



Backward propagation



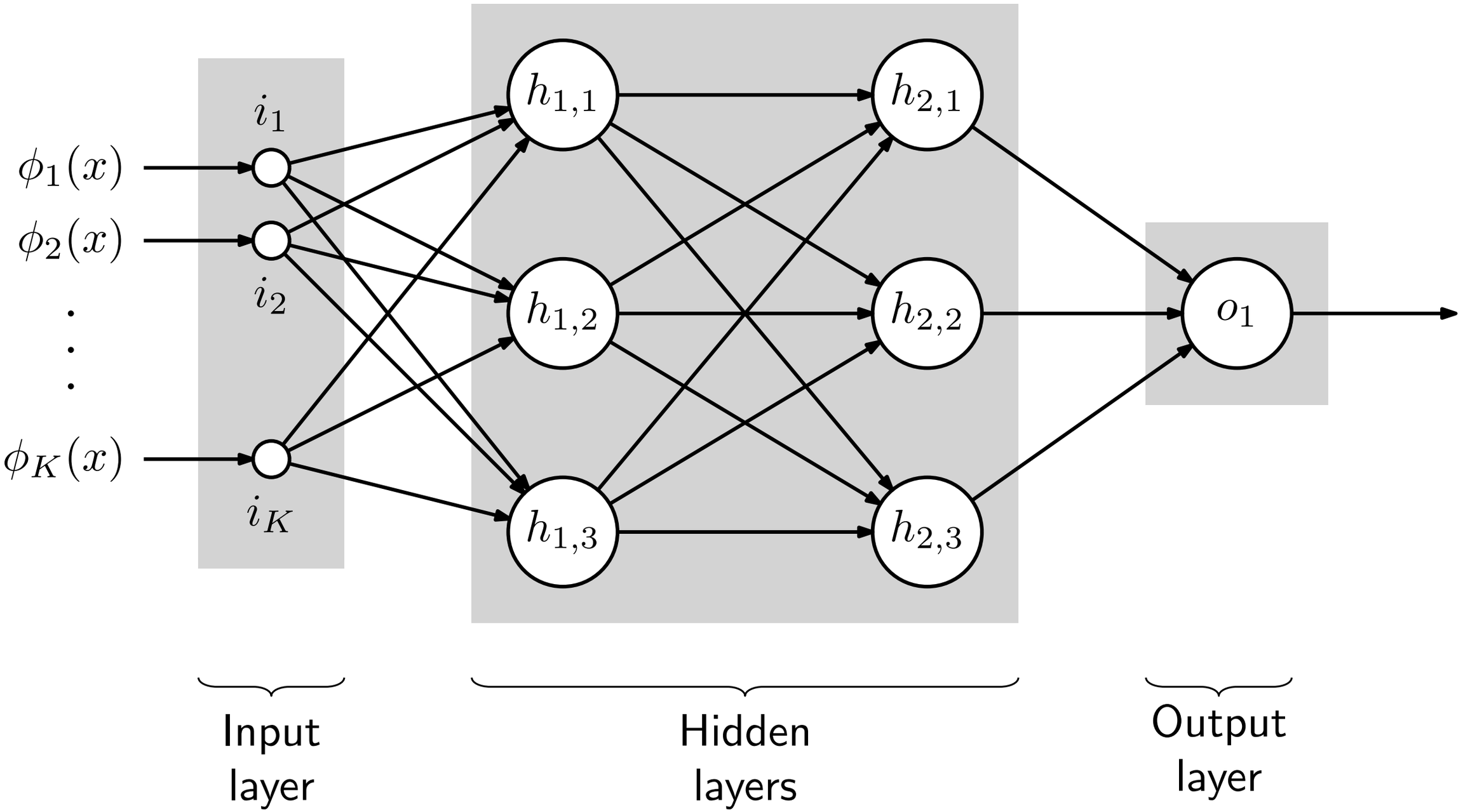
Backward propagation



Back-propagation

- Back-propagation is the standard approach to compute the gradient used in the training of neural networks
- Generalizes in a simple manner for more complicated networks

Multilayer perceptron



Multi-layer perceptron

- MLP can have multiple hidden layers
- Hidden layers and output layers can have varying number of neurons
- Layers are fully connected (all neurons in a layer connect to all neurons in the next layer)

Back propagation for MLP

1. Initialize weights w_{ij} (connecting output of neuron i to neuron j) to small random values
2. Apply a data-point $\mathbf{x}^*$ to the input layer
3. Propagate the signal through the network

$$y_i = \sigma(z_i) = \sigma(\boldsymbol{w}_{ji}\boldsymbol{y}_j)$$

4. Back-propagate δ_i back through the network

$$\delta_i = \begin{cases} (a_i - a) & \text{if } i \text{ is an output unit} \\ \sigma'(z_i)\sum_k w_{ij}\delta_j & \text{otherwise} \end{cases}$$

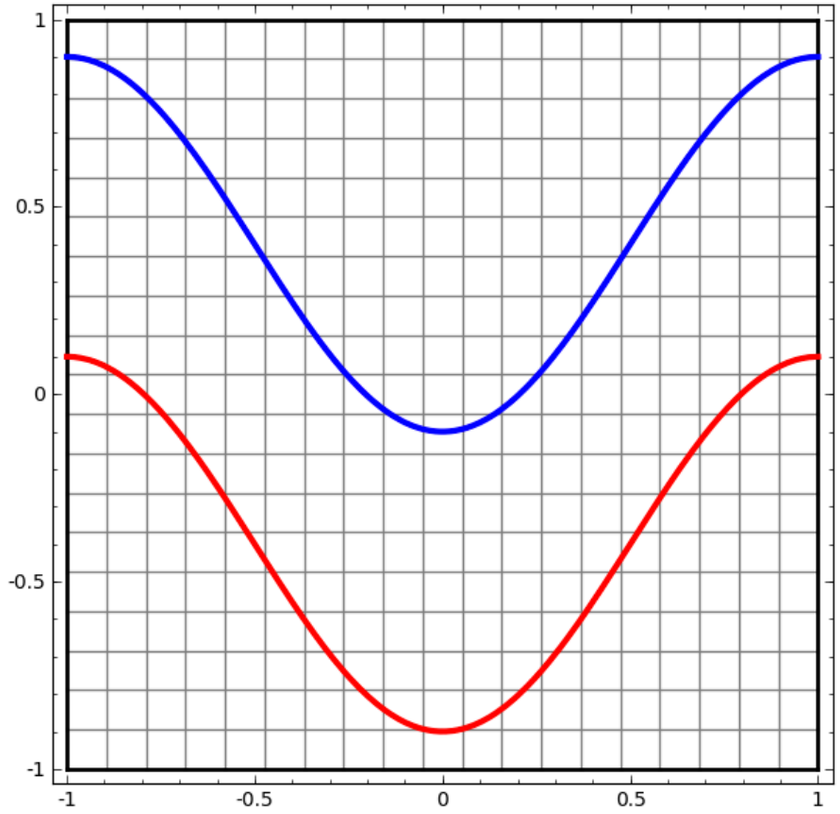
5. Update weights

How do NN work?

- Output is like logistic regression
  - What do hidden layers do?

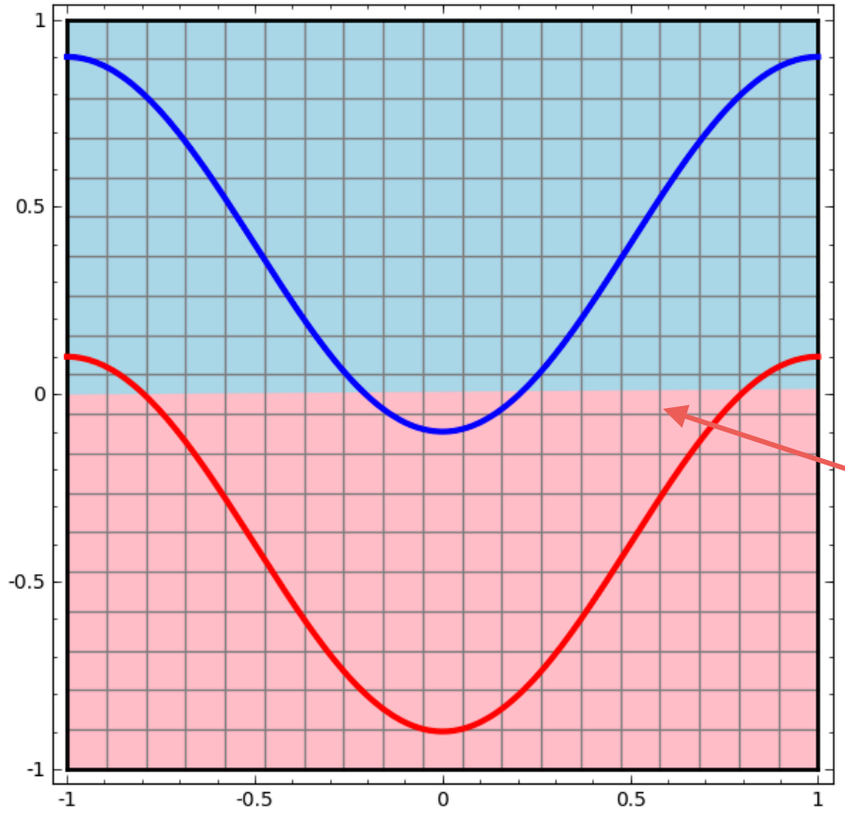
How do NN work?

- Simple example:
  - Classify points as belonging to one of the two curves



How do NN work?

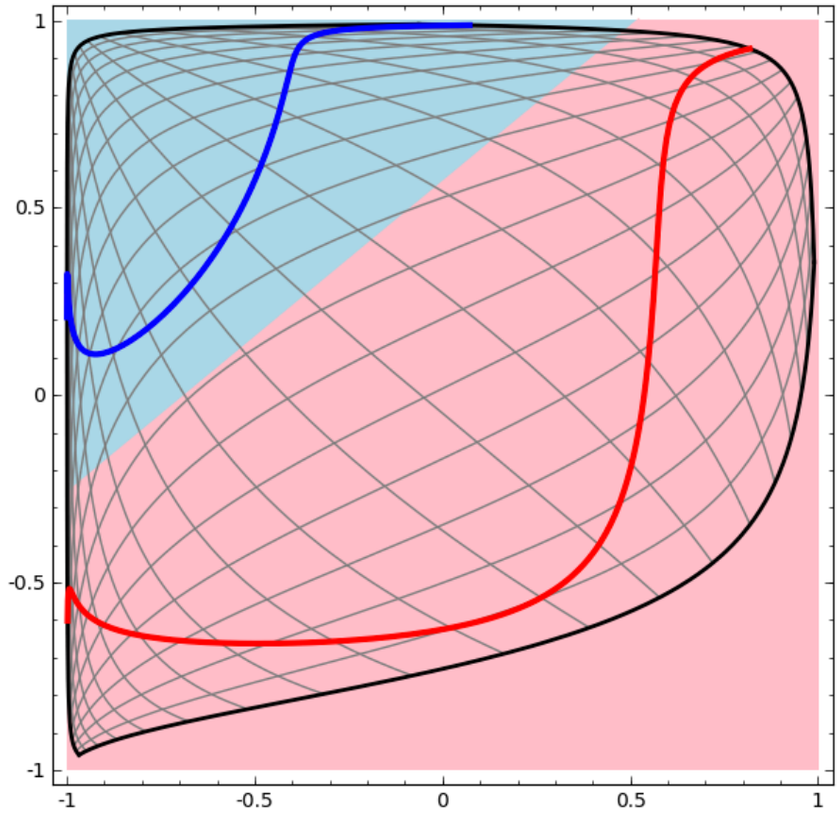
- Simple example:
  - Classify points as belonging to one of the two curves



A linear classifier (LR) does not work…

How do NN work?

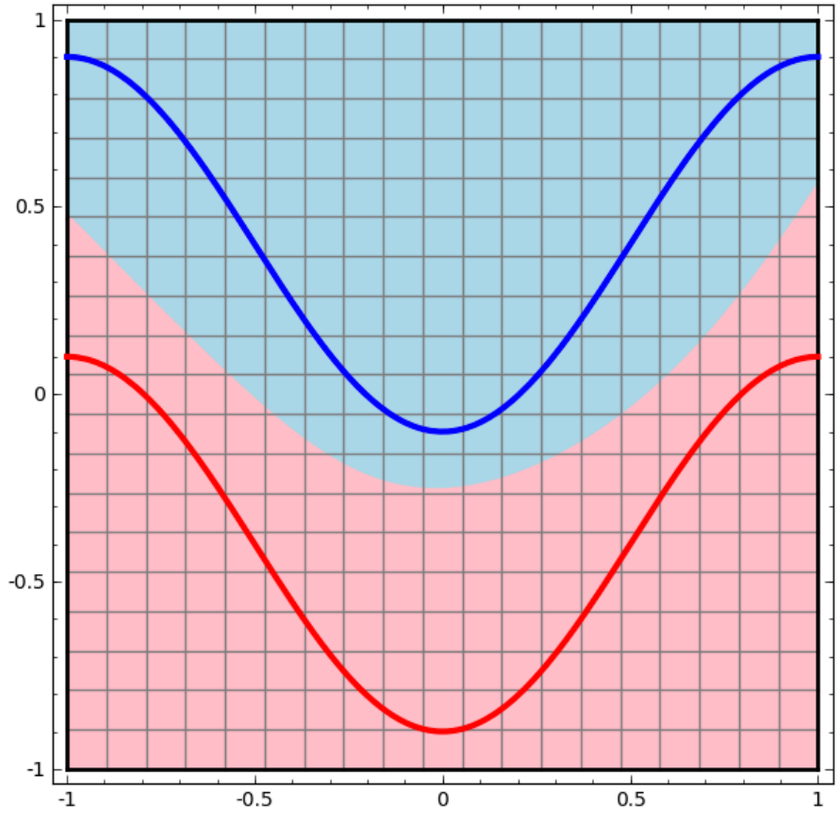
- Simple example:
  - Classify points as belonging to one of the two curves



Hidden layers “tweak” the space making the data linearly separable

How do NN work?

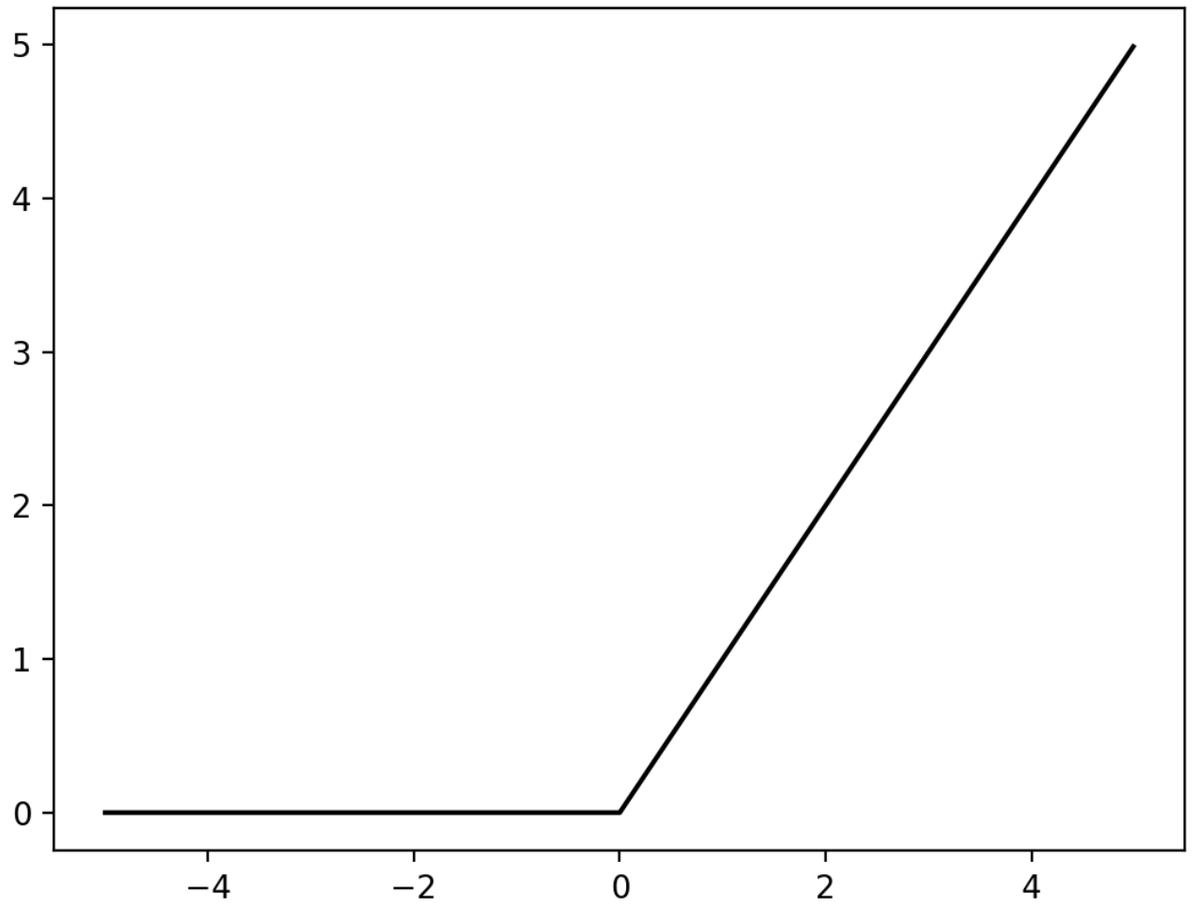
- Simple example:
  - Classify points as belonging to one of the two curves



In the original space, this is what the decision boundary looks like

Final remarks

- Current practice has replaced the sigmoid function by a rectified linear unit (ReLU)



Final remarks

- Current practice has replaced the sigmoid function by a rectified linear unit (ReLU)
- Neural networks are at the core of **deep learning**
- Neural networks used in deep learning are usually not MLP, although the structure near the output layers resembles it
- The training method is still back-propagation