Unit 1: Simple Neural Networks

5. Perceptrons

9/15/2020

Perceptrons

1. Simple neural networks generalize the Rescorla-Wagner model of associative learning

2. Perceptrons are general-purpose linear classifiers. They can solve lots of problems

3. But they can't solve all problems…

Classical conditioning

Unconditioned
Stimulus (US)

↓

Unconditioned
Response (UR)

Unconditioned
Stimulus (US)
+
Conditioned
Stimulus (CS)

↓

Unconditioned
Response (UR)

Conditioned
Stimulus (CS)

↓

Conditioned
Response (CR)

The Rescorla-Wagner model: Learning is prediction error

$$P\left(cheese\right) = V_{total}$$

$$\Delta V = \alpha \cdot \left(\lambda - V_{total}\right)$$

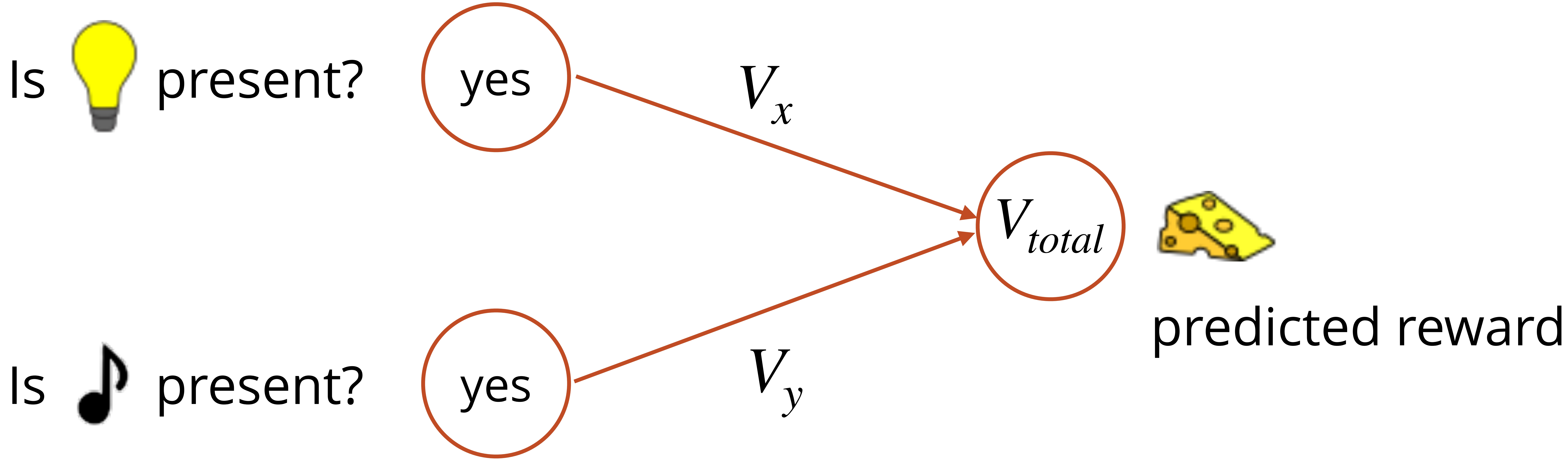
On each trial, the rat predicts whether or not it will get cheese

This prediction comes from the combination of all cues

After each trial, update predictions for each cue

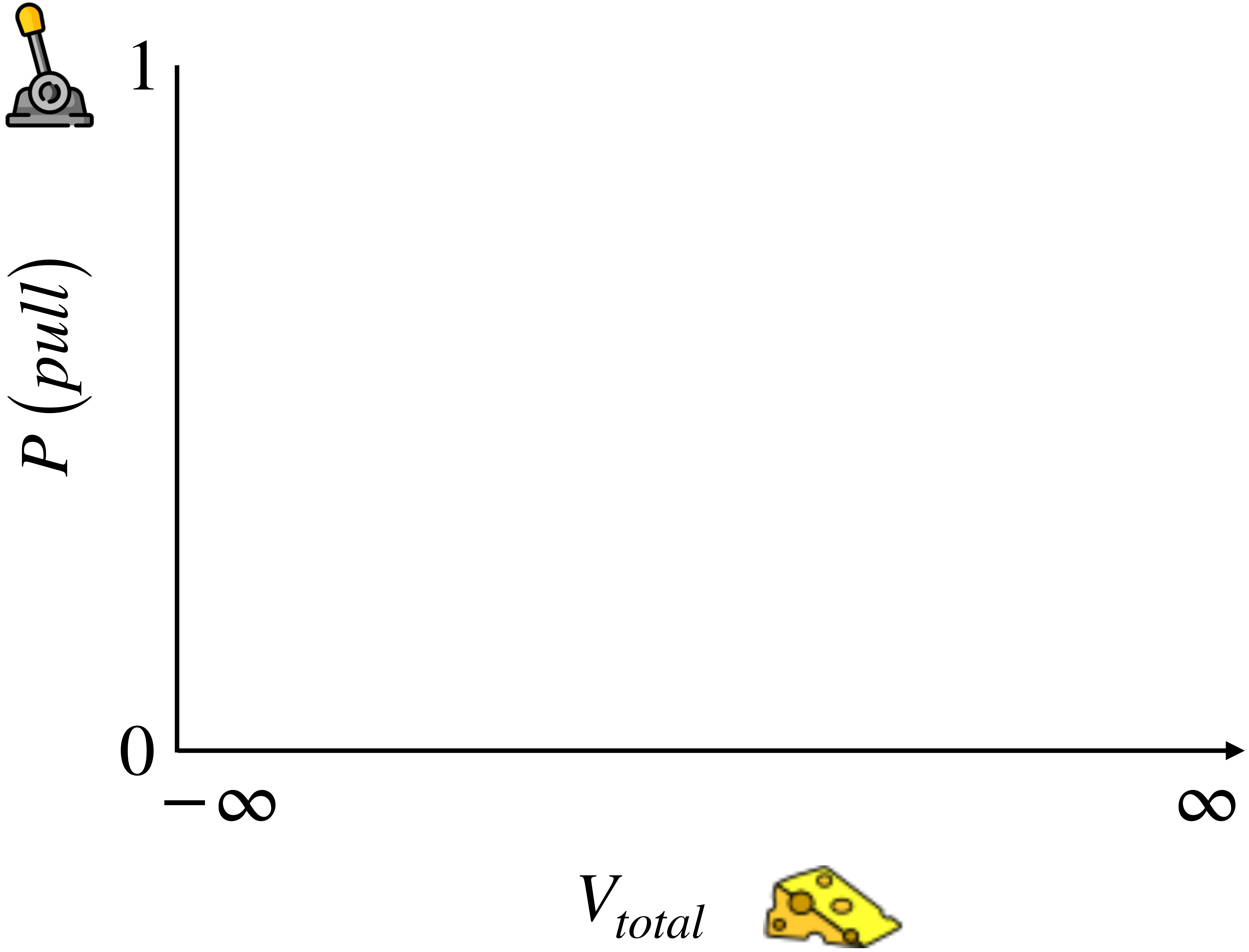
- If the rat gets cheese—but *didn't expect* cheese—increase prediction for each cue
- If the rat doesn't get cheese—but *expected* cheese—decrease prediction for each cue
- Otherwise, don't change anything

A network representation of Rescorla-Wagner

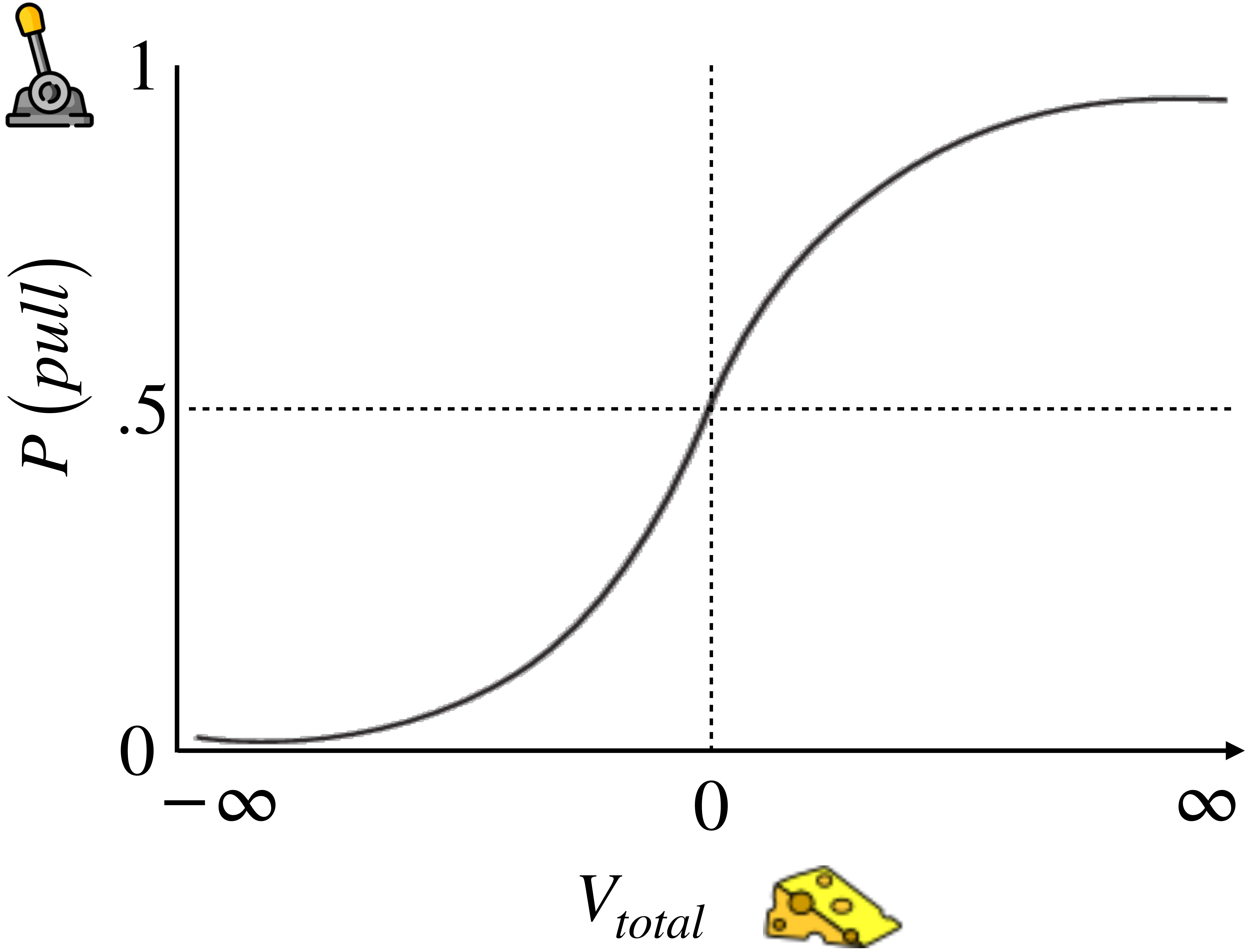


Like Ramscar et al. (2010)

Modification 1: Connecting prediction to action with a squashing function

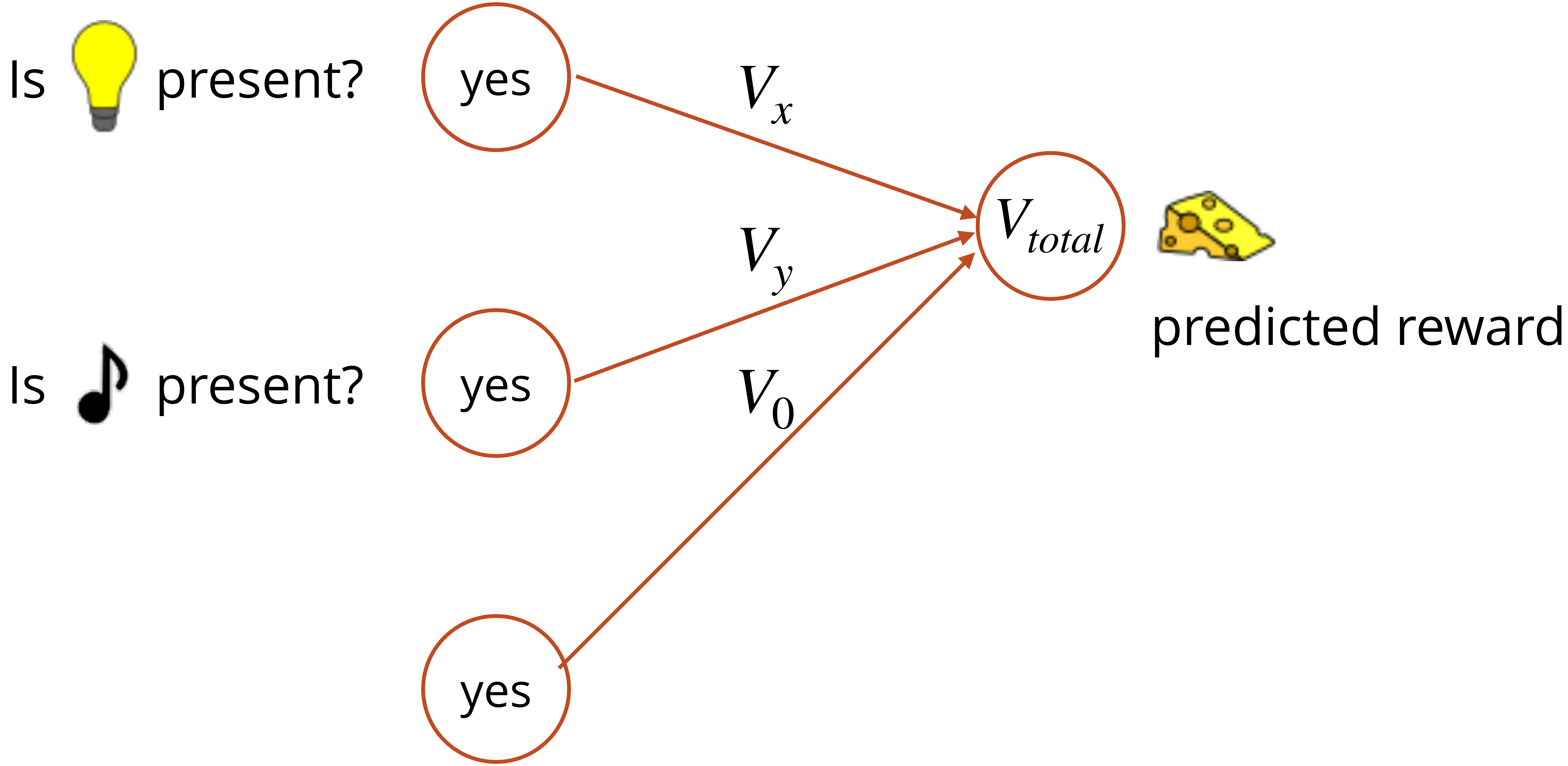


The sigmoid (logistic) function

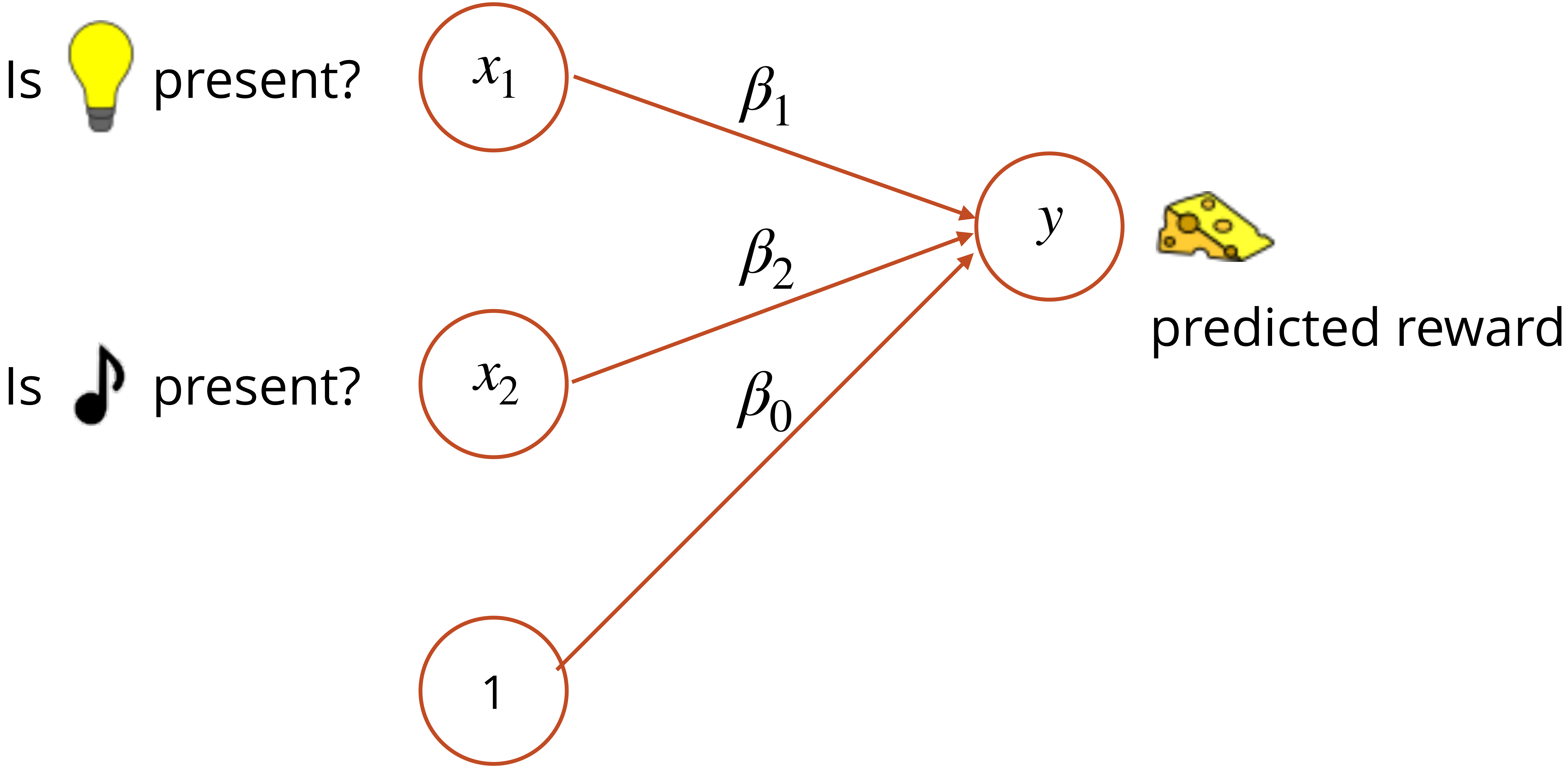


$$y = \frac{1}{1 + e^{-x}}$$

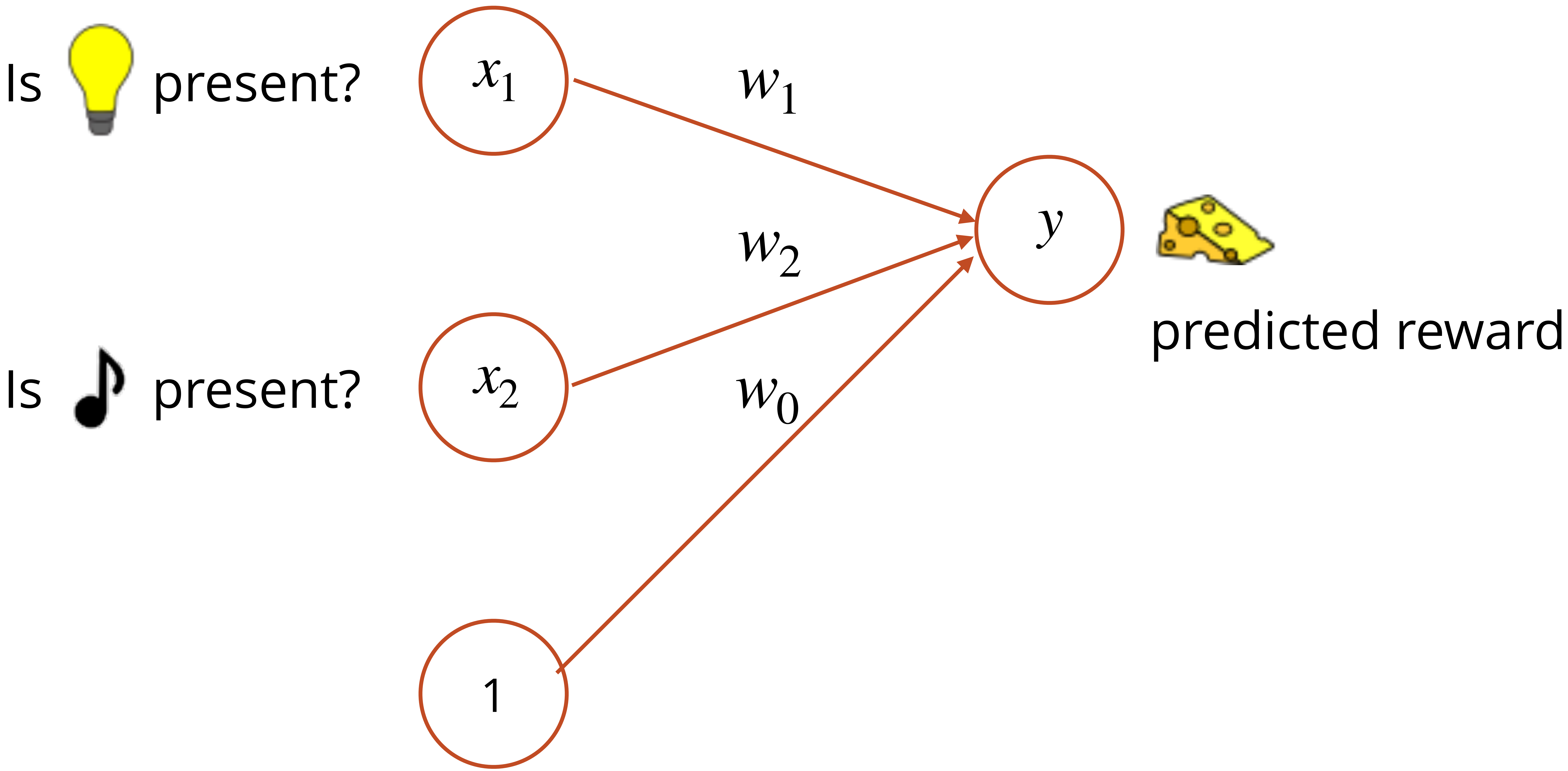
Modification 2: a bias term



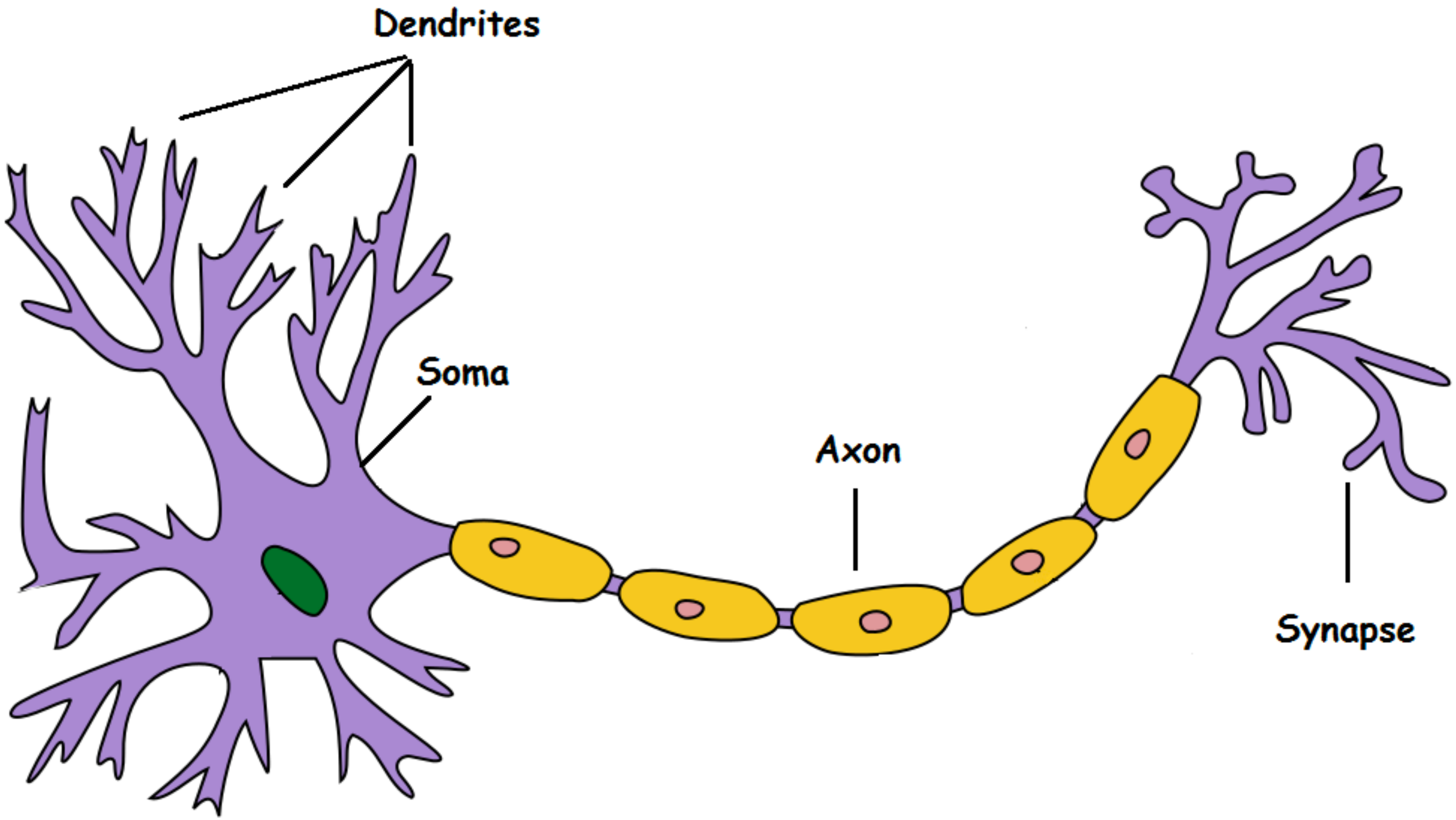
Aside: This is exactly logistic regression!



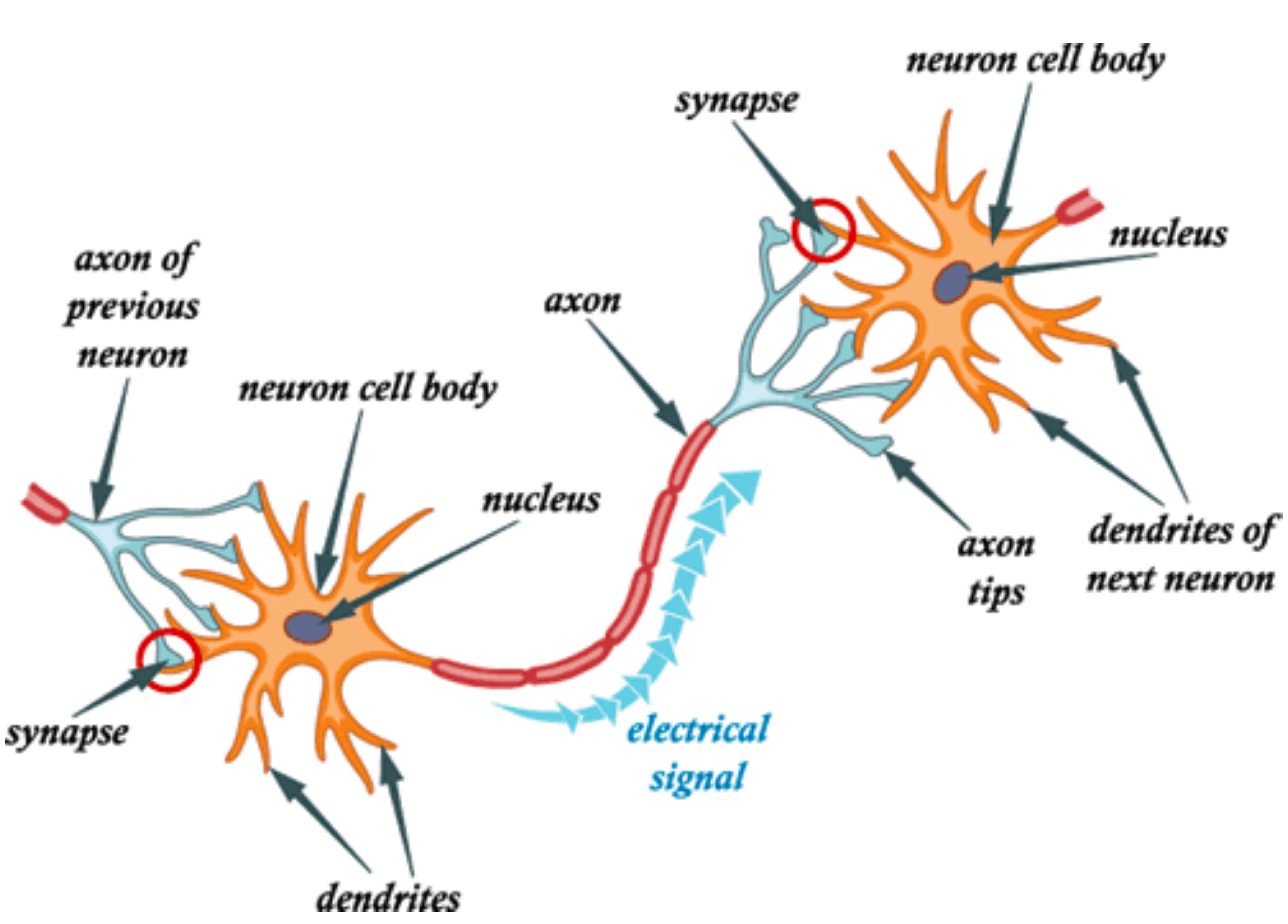
It's also a very simple model of a neuron

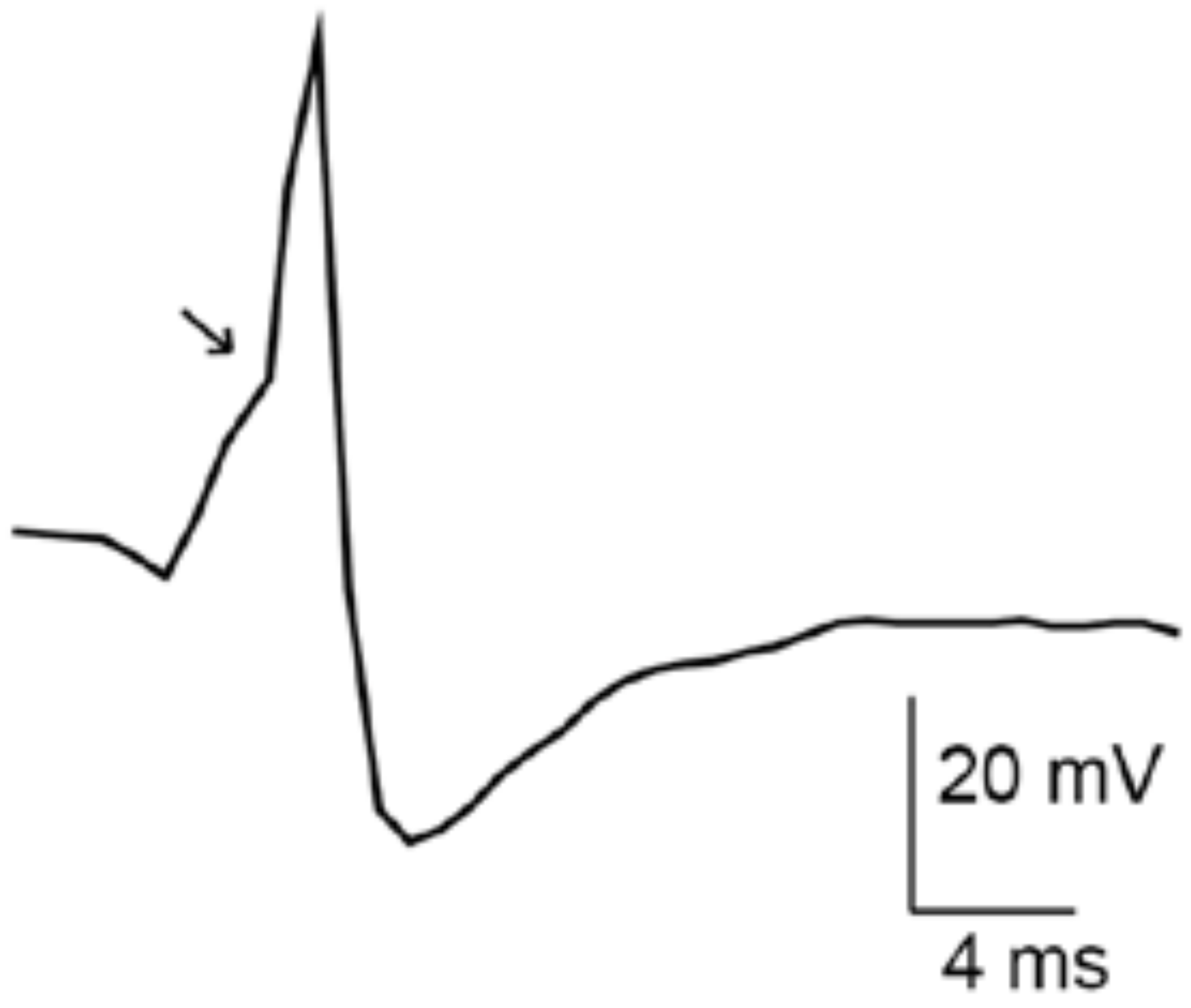


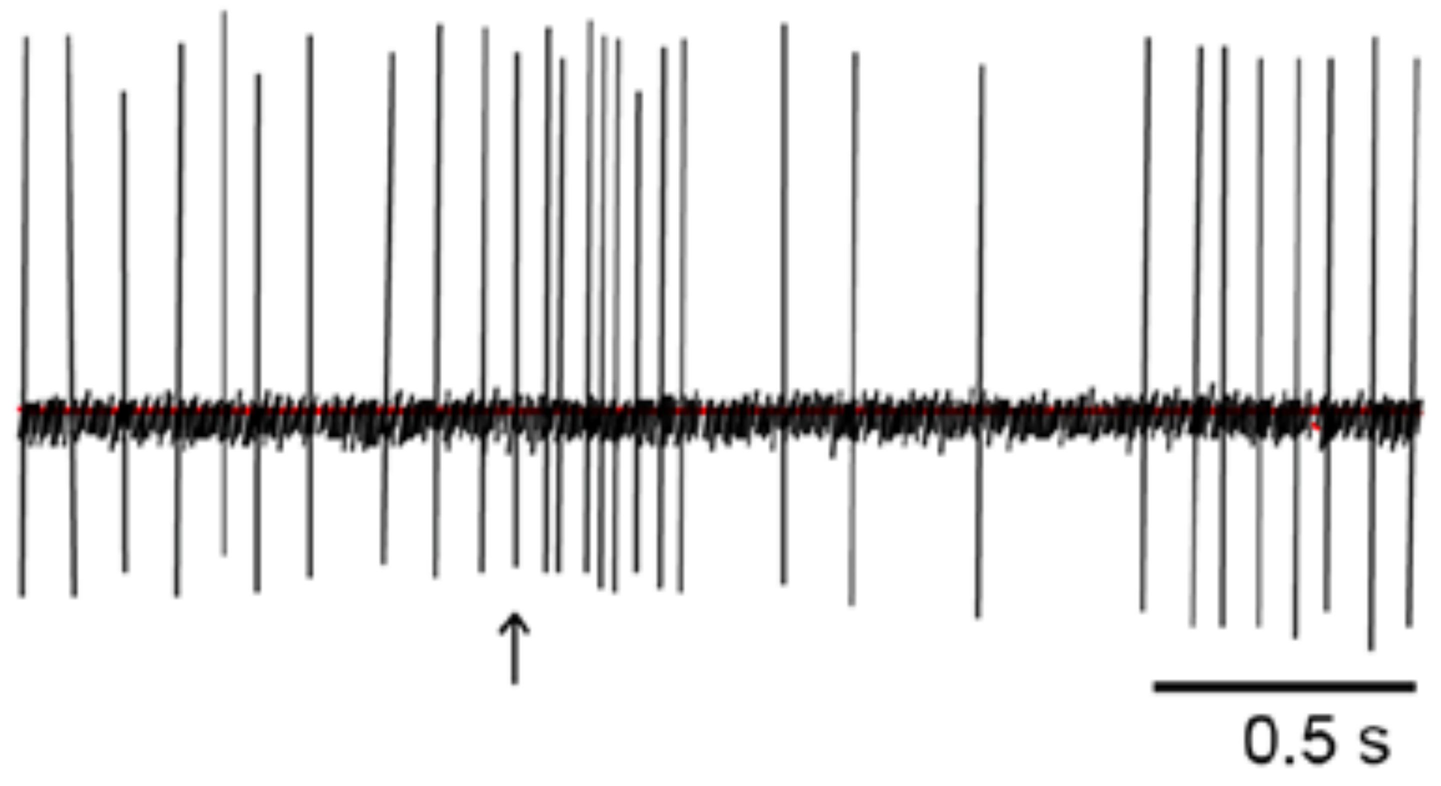
A biological neuron



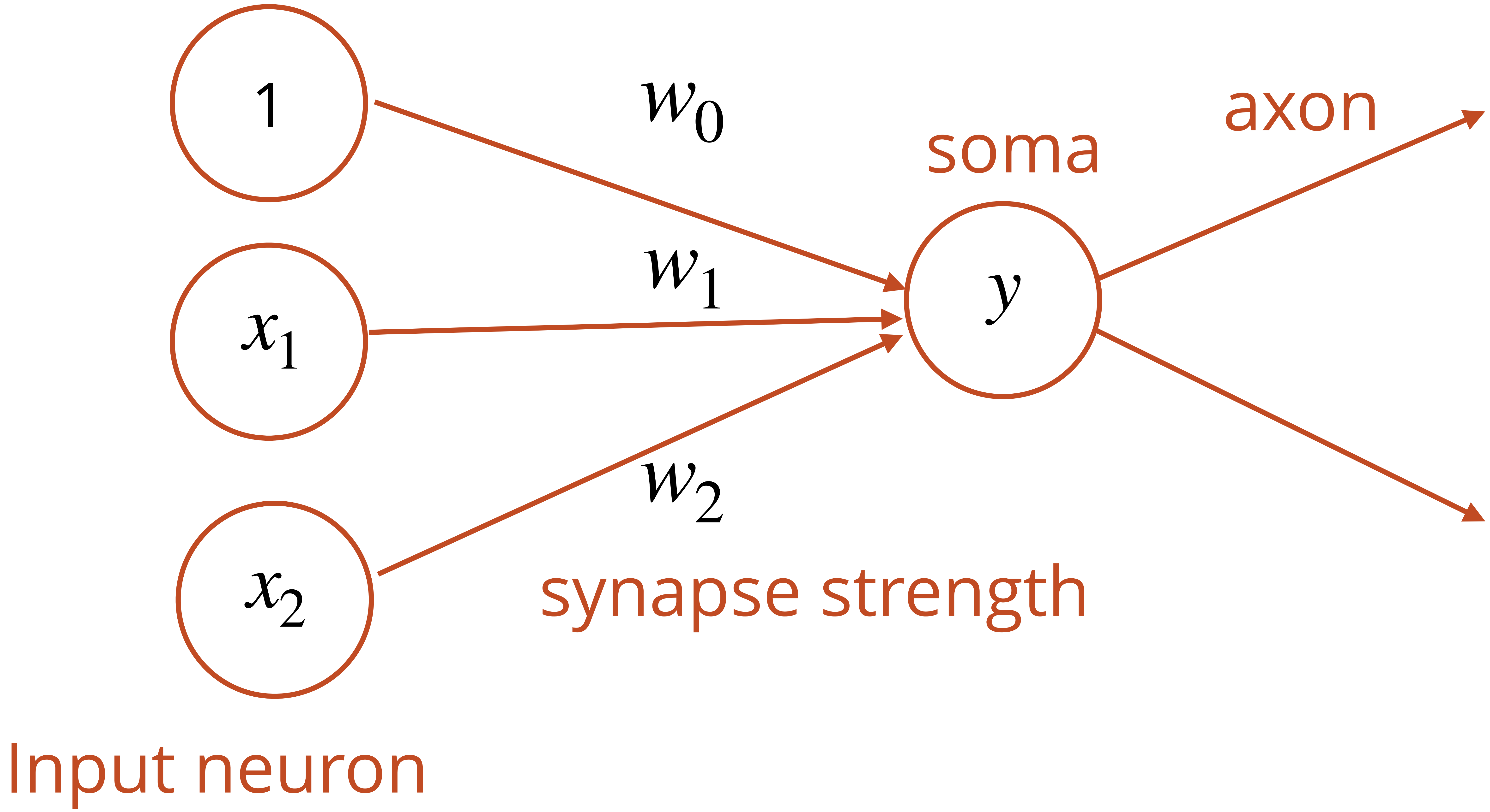
A biological neuron

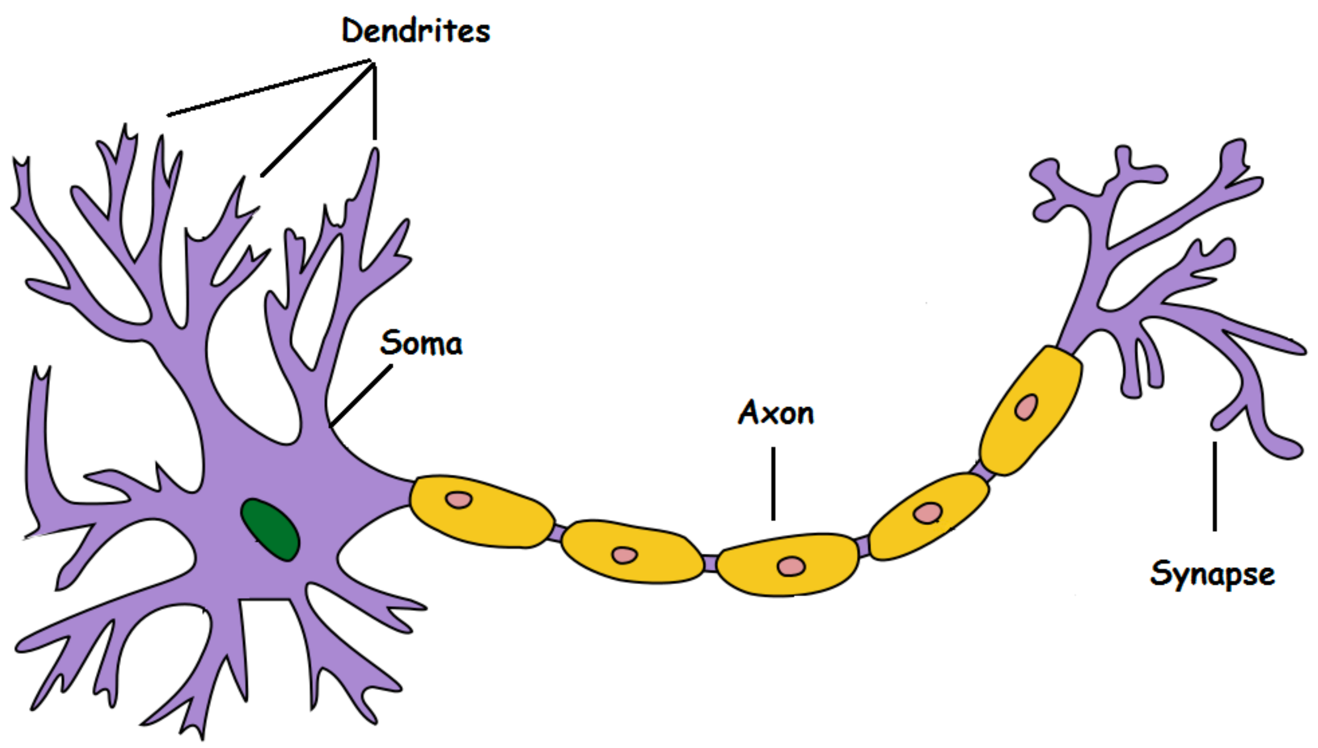




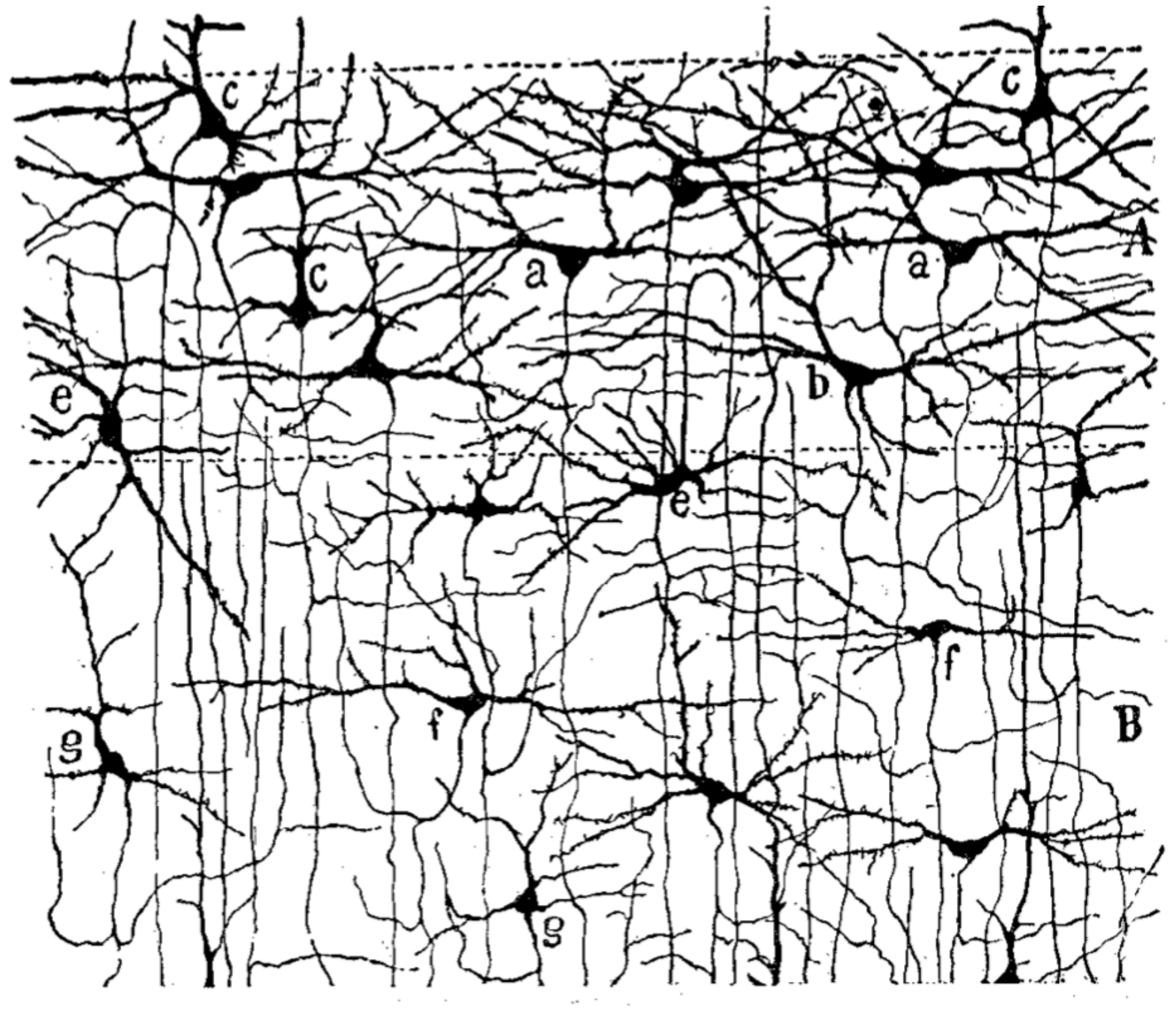


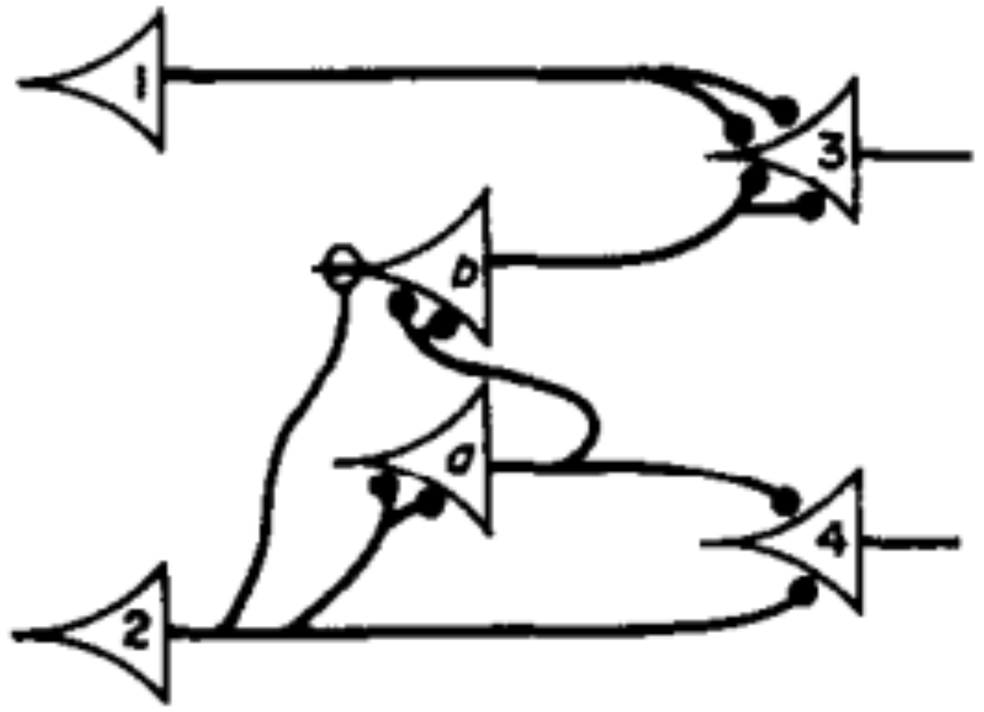
An artificial neuron



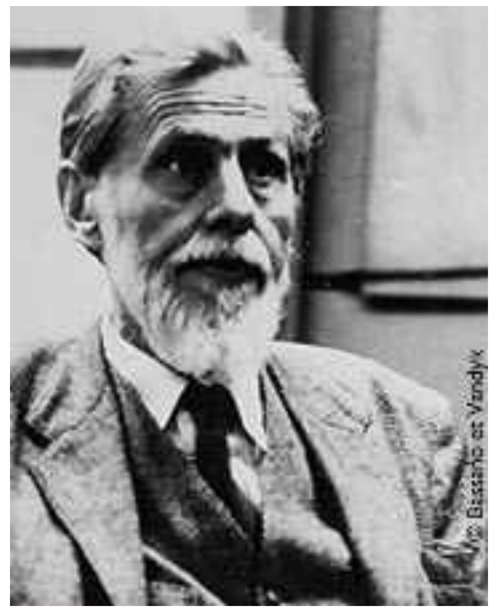


The first artificial neural networks





Because neuronal firing is discrete, neural networks can approximate boolean logic!

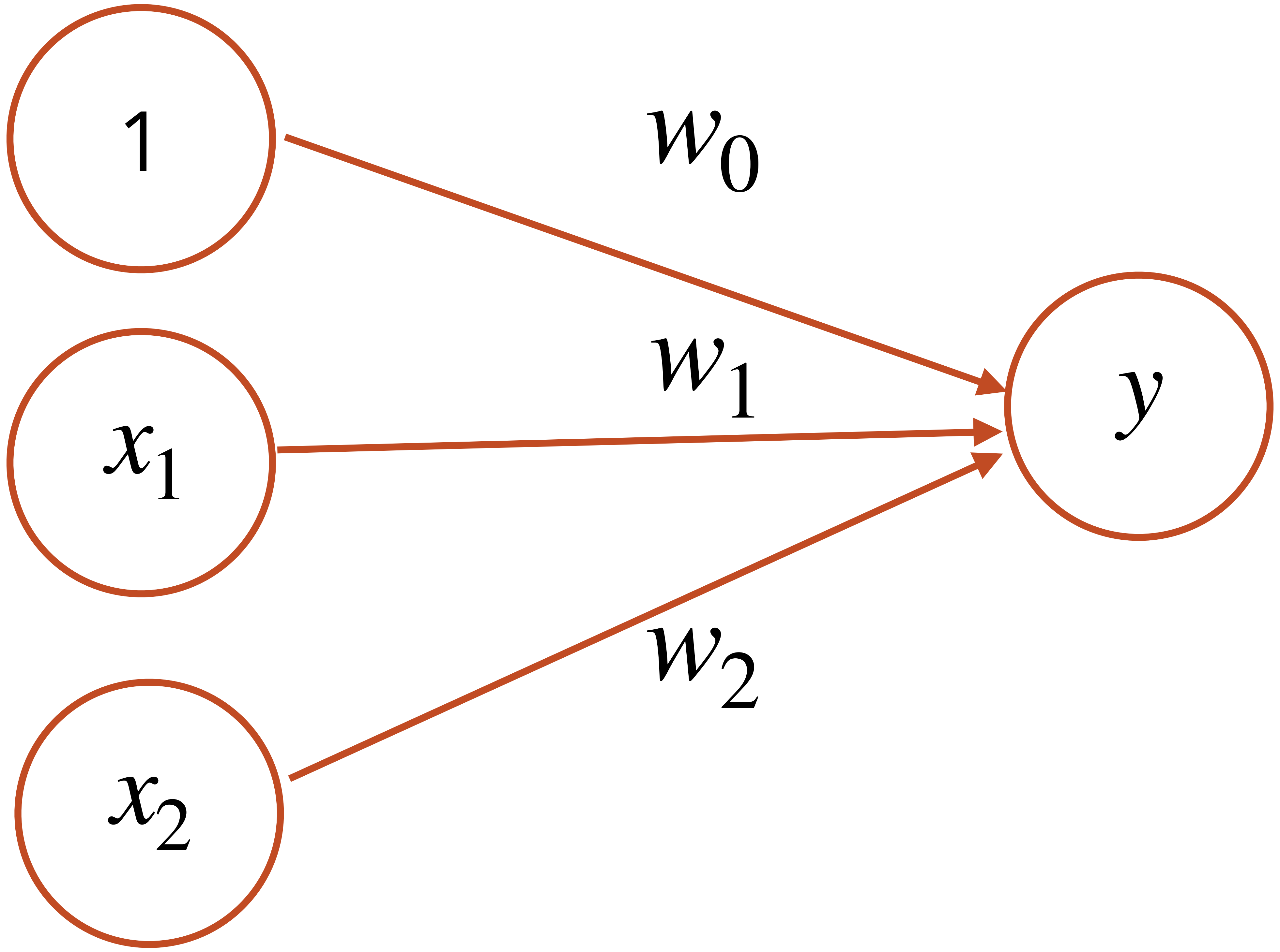


Warren McCulloch

Walter Pitts

What’s more, they can *learn* to do logic

The Perceptron learning rule (Rosenblatt, 1958)



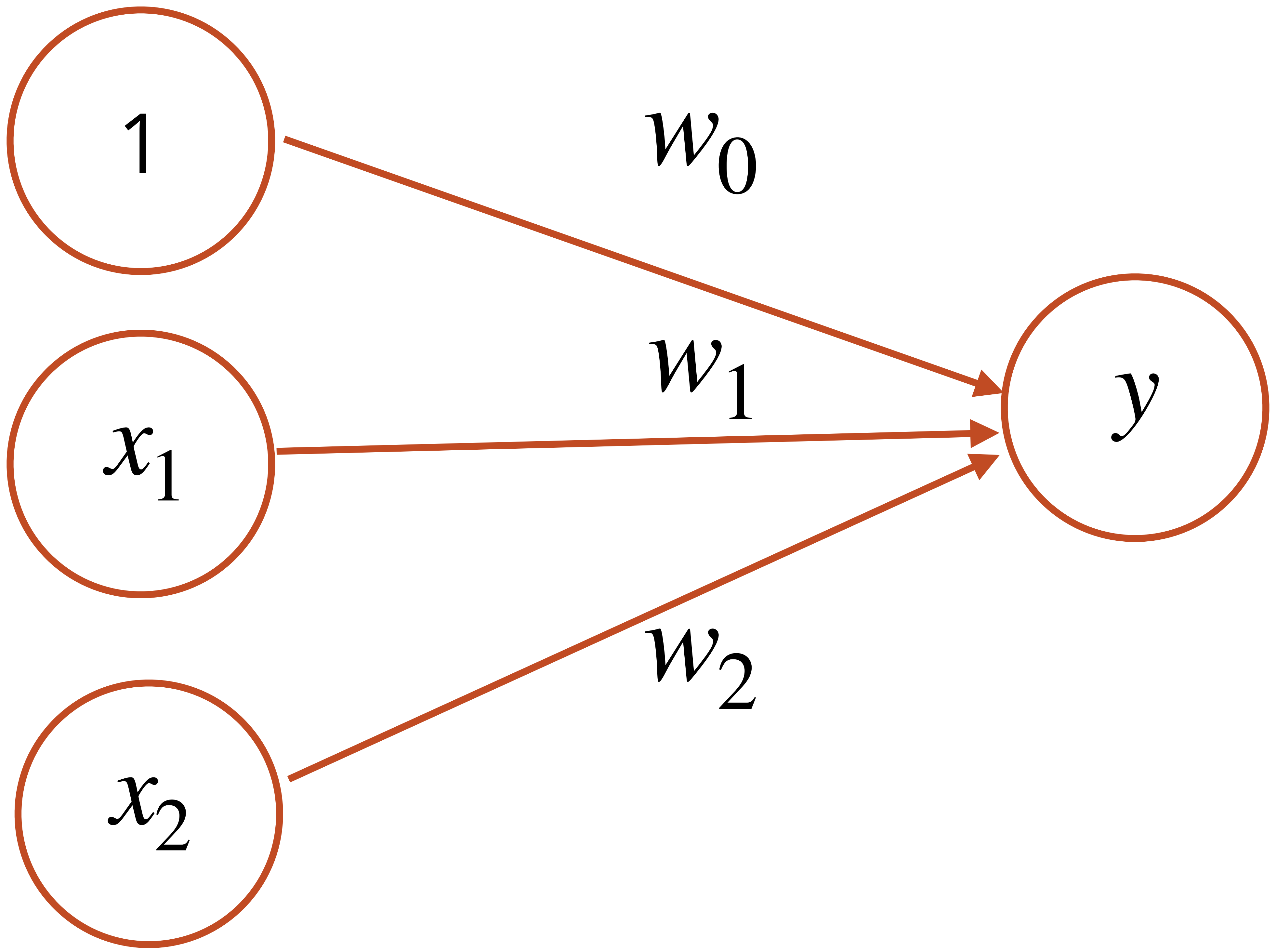
$$f(x) = \frac{1}{1 + e^{-x}}$$

$$y = f \sum_i w_i \cdot x_i$$

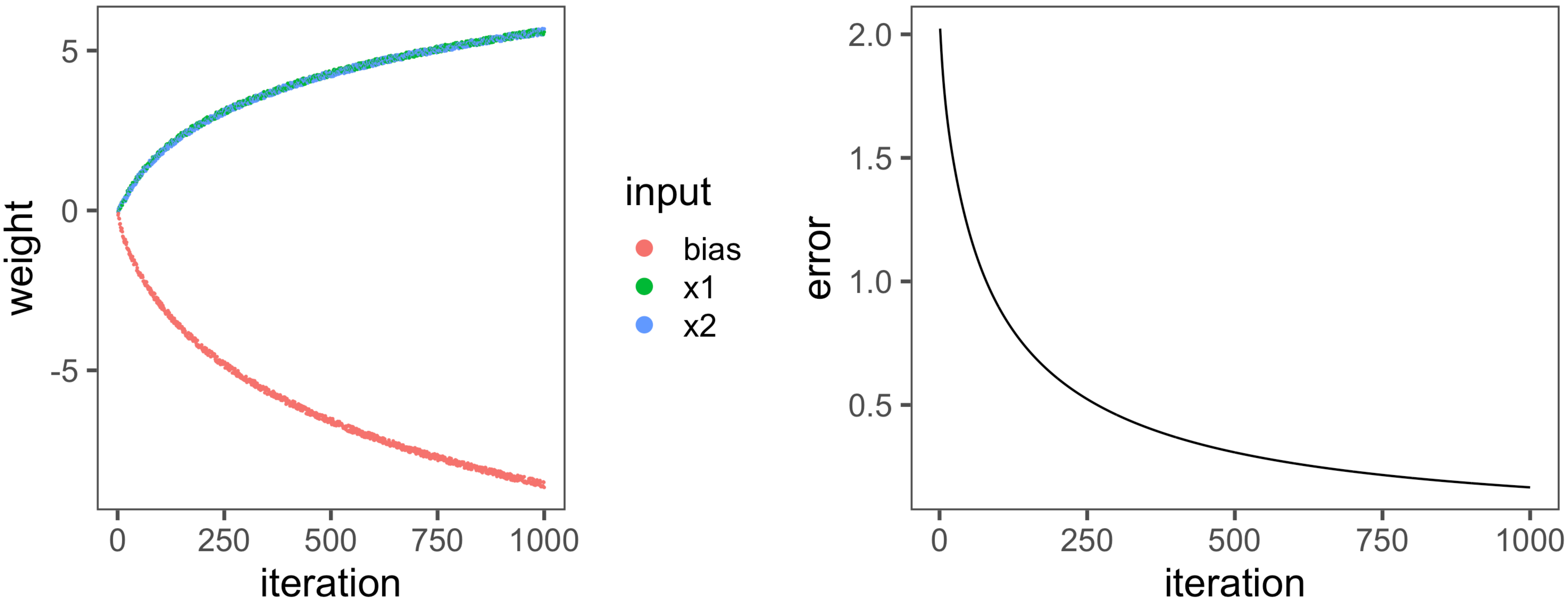
Perceptron learning rule $\Delta w_i = \alpha \cdot \left(y - \hat{y}\right) x_i$

Rescorla-Wagner learning rule $\Delta V = \alpha \cdot \left(\lambda - V_{total}\right)$

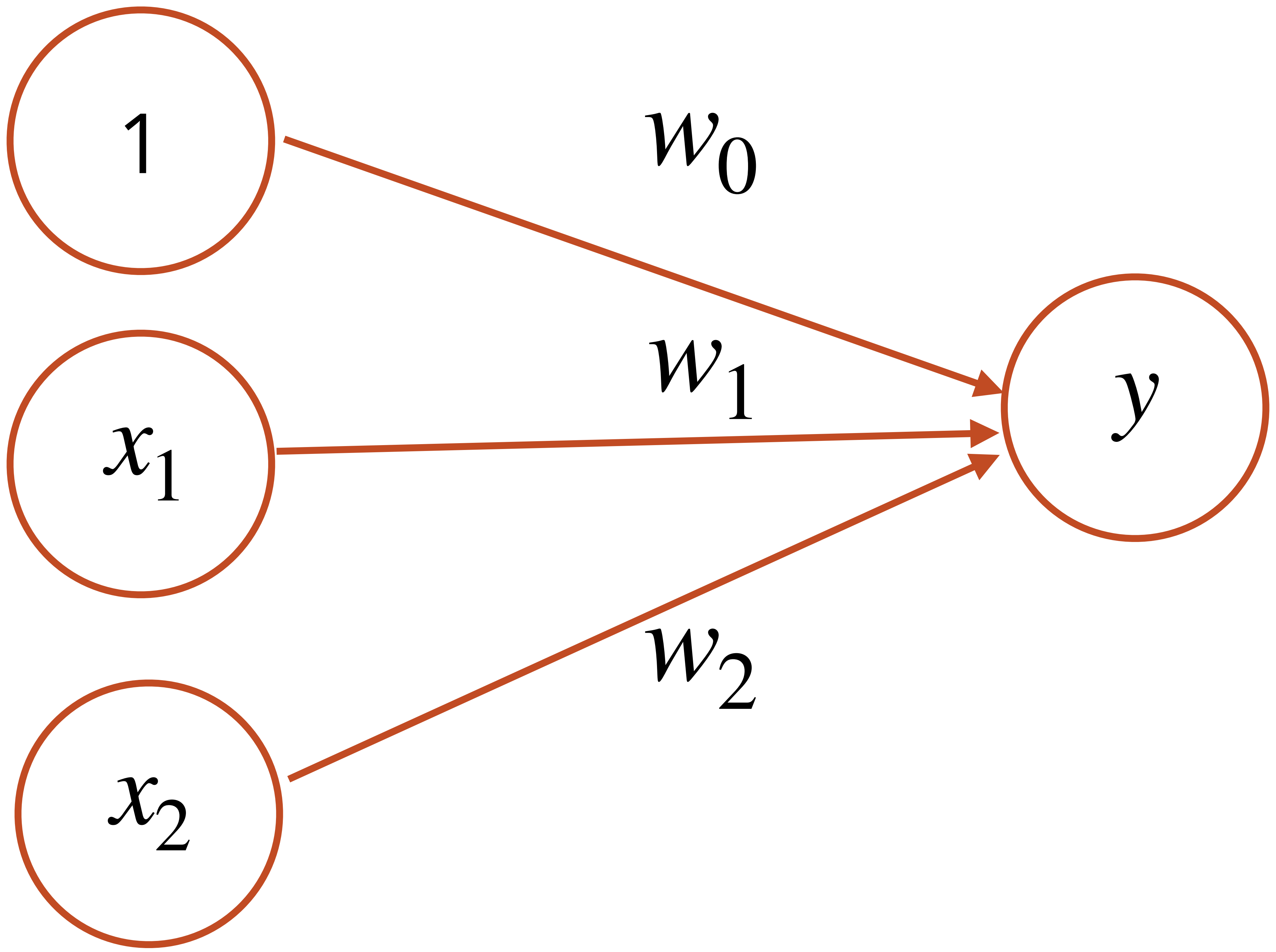
Building an AND network



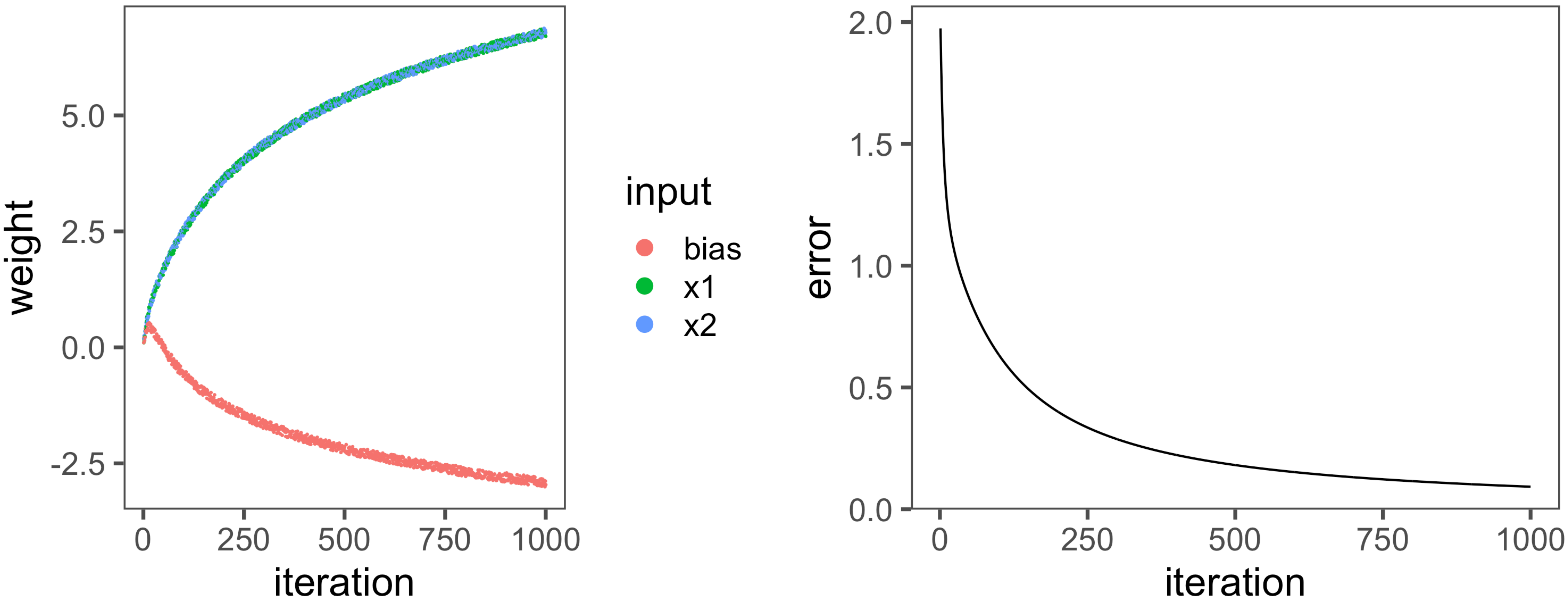
Learning the AND function



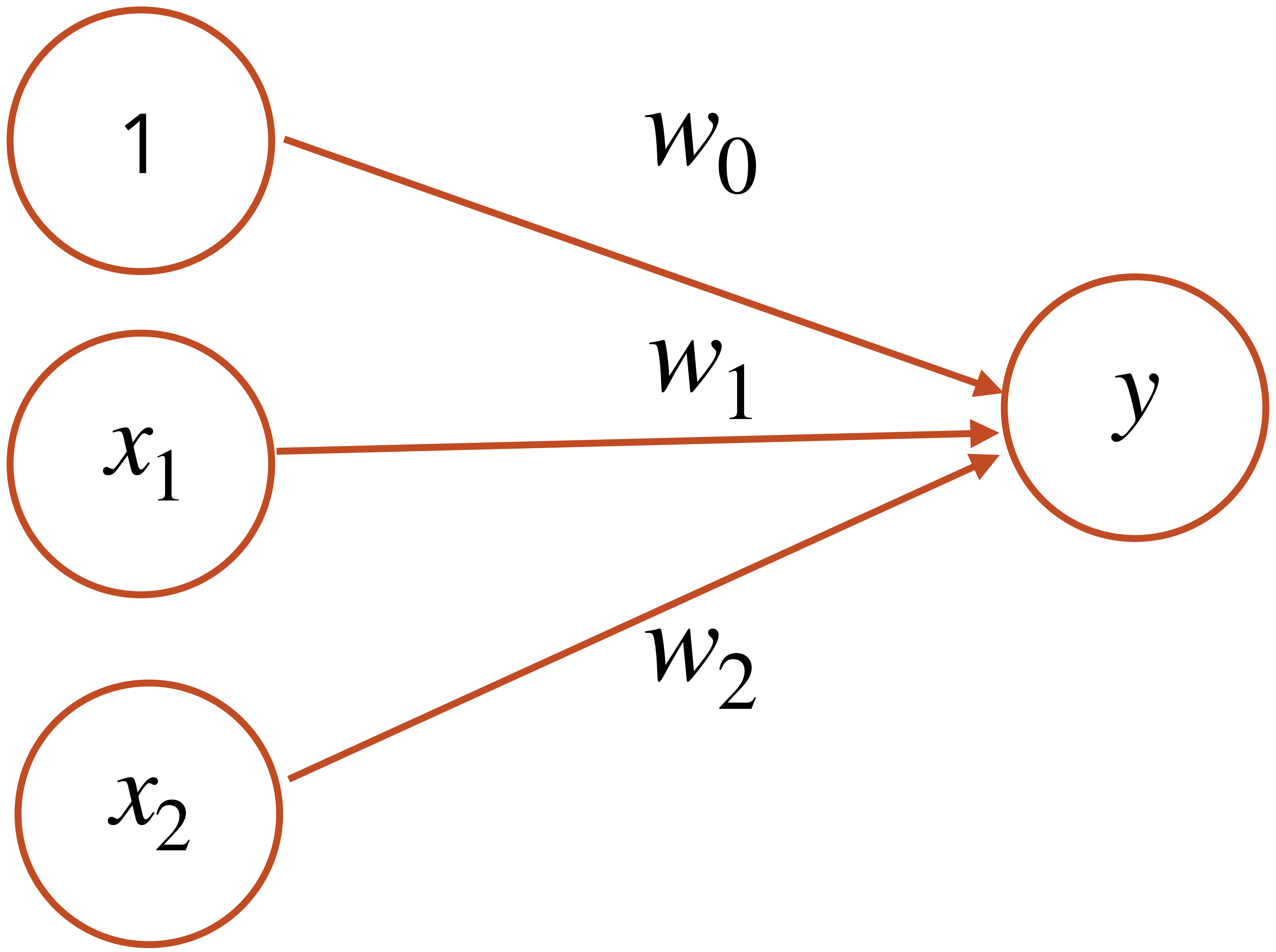
Building an OR network



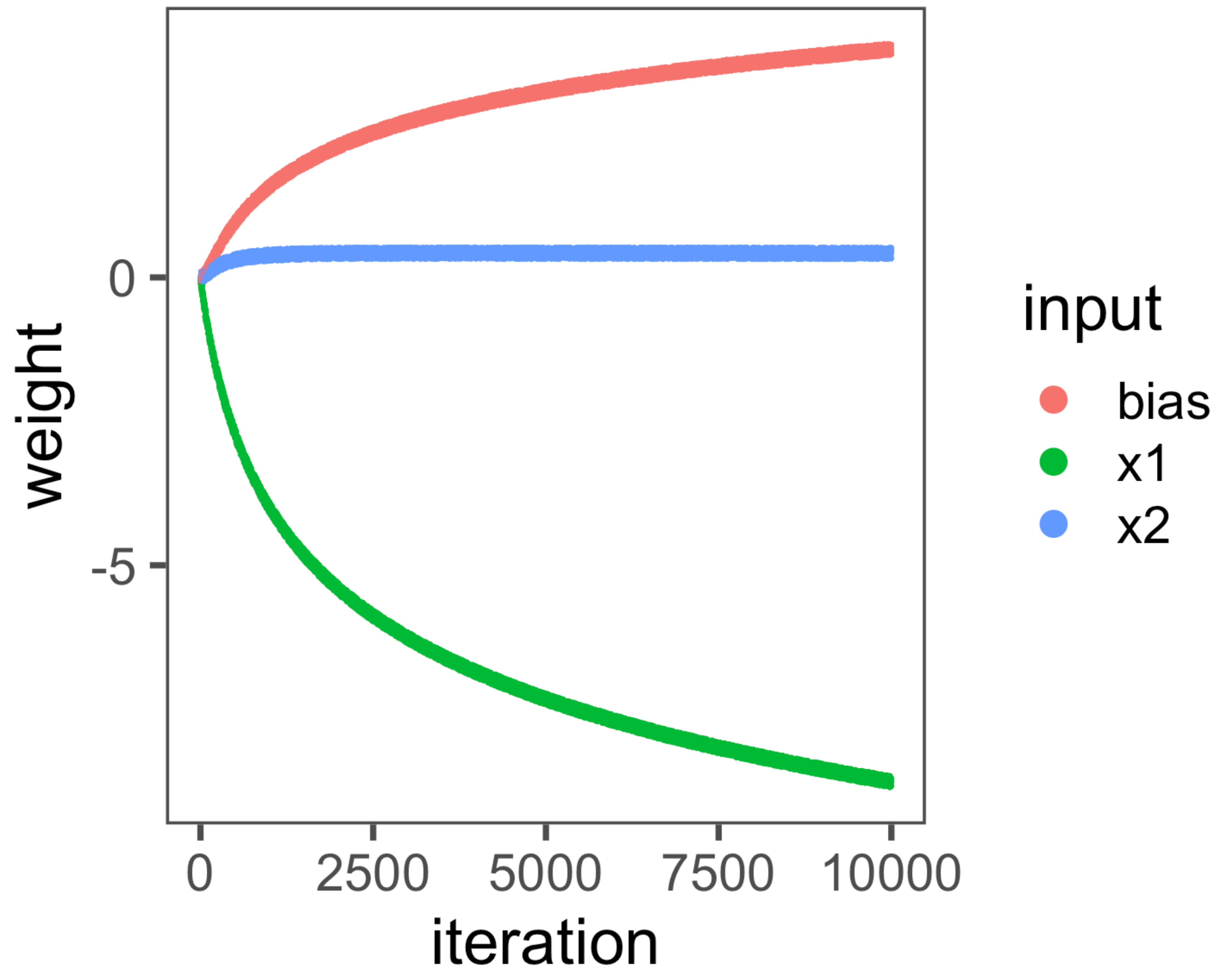
Learning the OR function

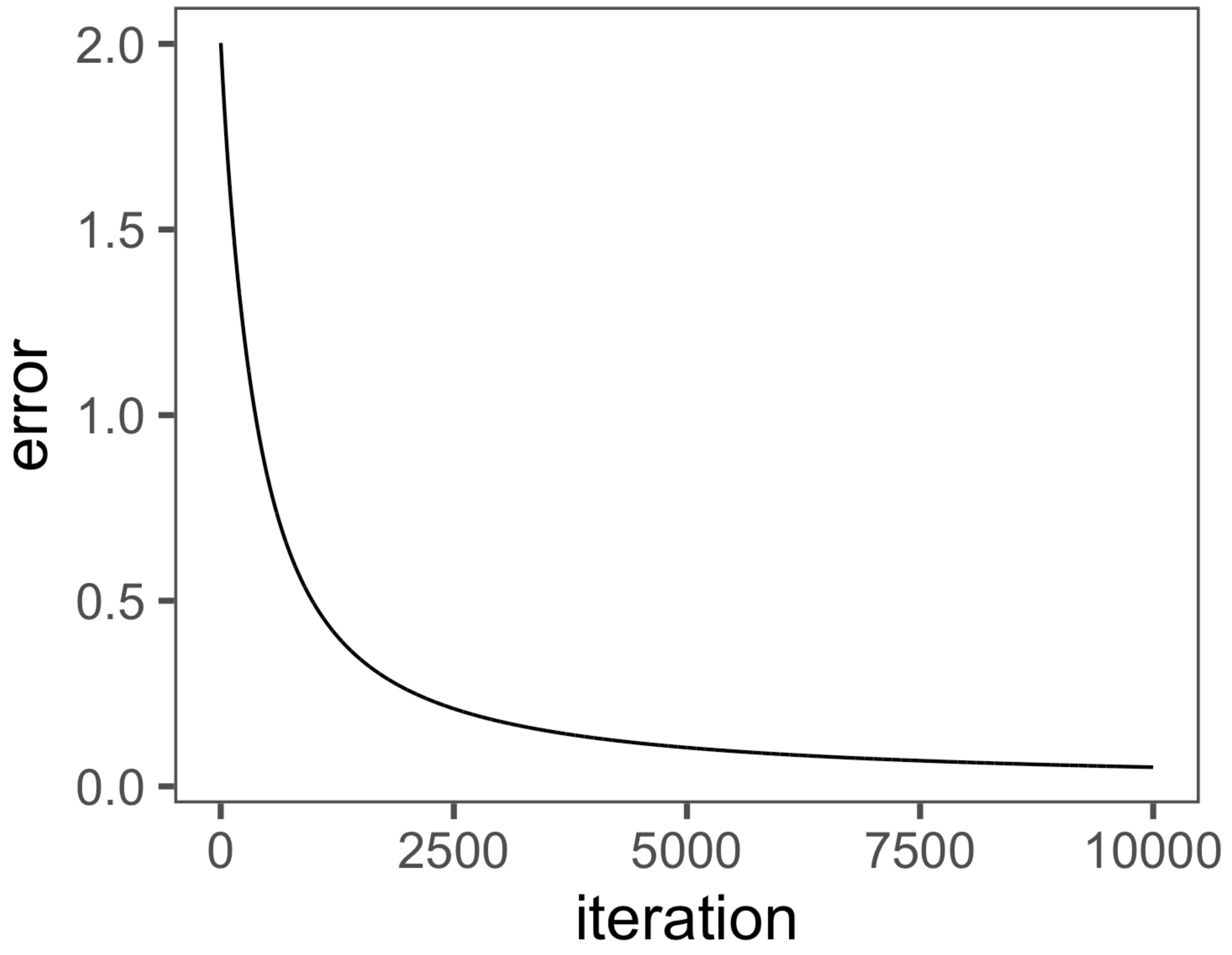


Building a NOT(x1) network



Learning the NOT(x) function





Perceptrons as general classifiers

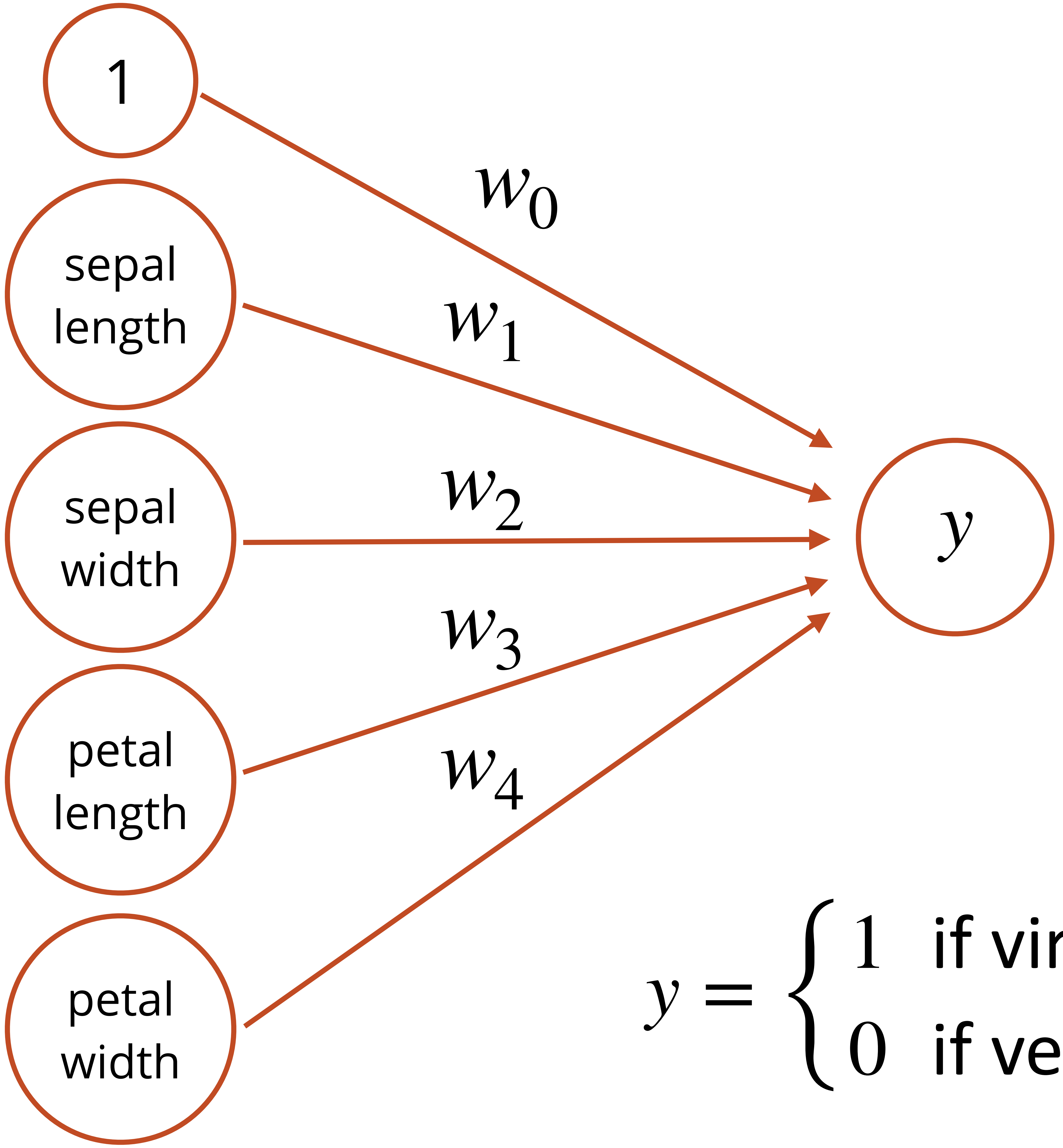
iris virginica

iris versicolor

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
7	3.2	4.7	1.4	0
6.4	3.2	4.5	1.5	0
6.9	3.1	4.9	1.5	0
5.5	2.3	4	1.3	0
6.5	2.8	4.6	1.5	0
5.7	2.8	4.5	1.3	0
6.3	3.3	4.7	1.6	0
4.9	2.4	3.3	1	0
6.6	2.9	4.6	1.3	0
5.2	2.7	3.9	1.4	0
5	2	3.5	1	0

Fisher (1936)

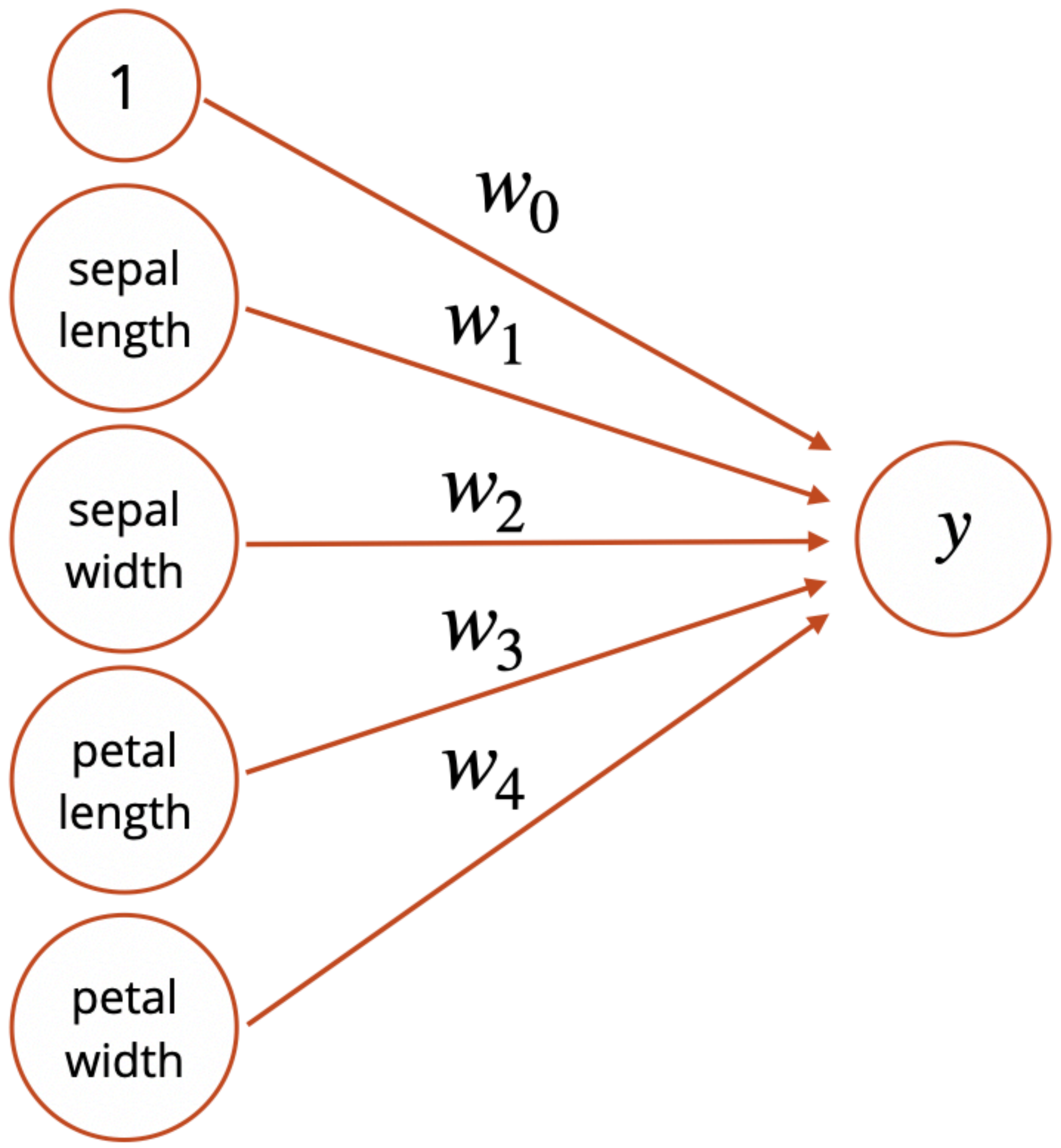
Building an iris classifier



$$y = \begin{cases} 1 & \text{if viriginica,} \\ 0 & \text{if versicolor} \end{cases}$$

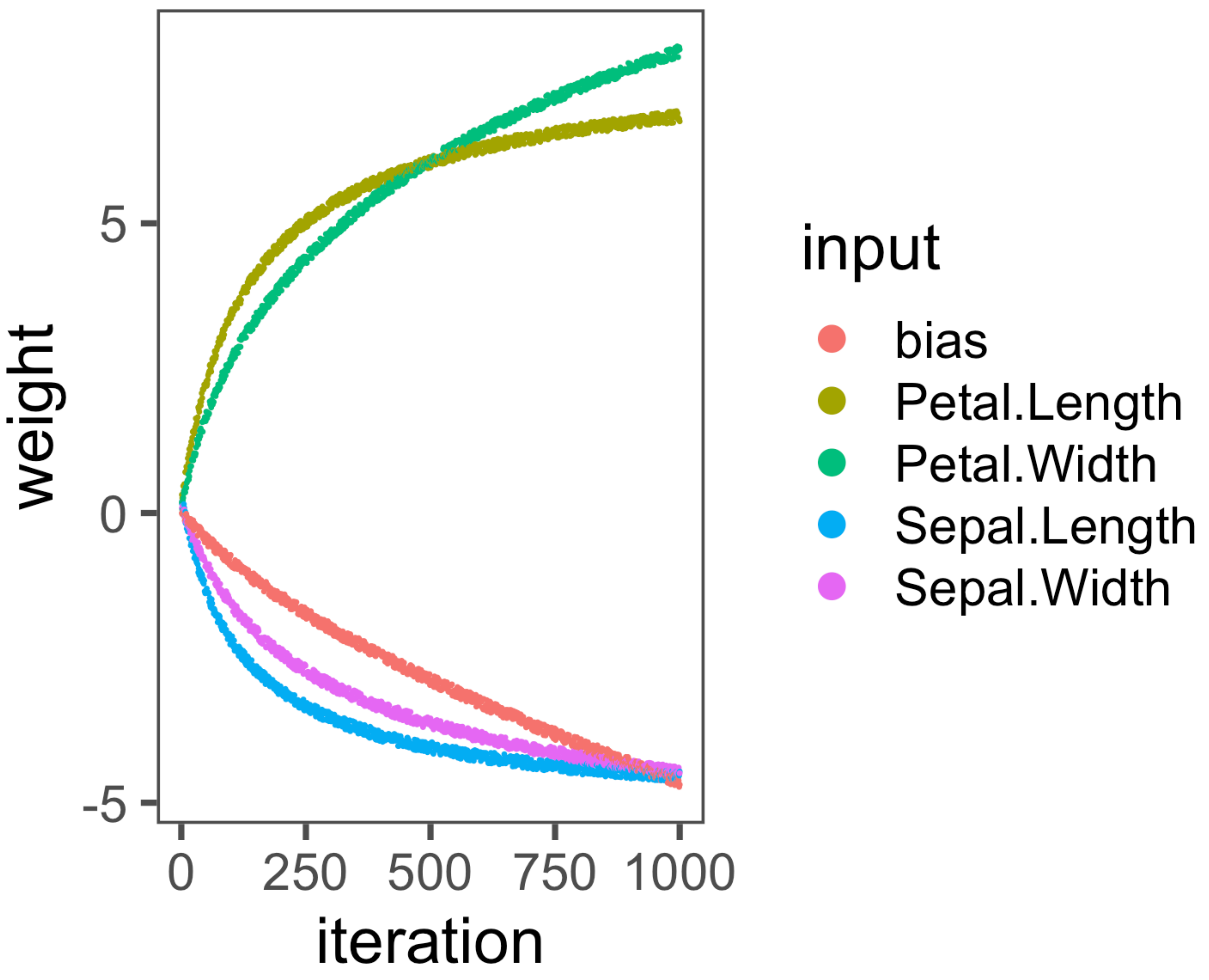
Logistic regression as an iris classifier

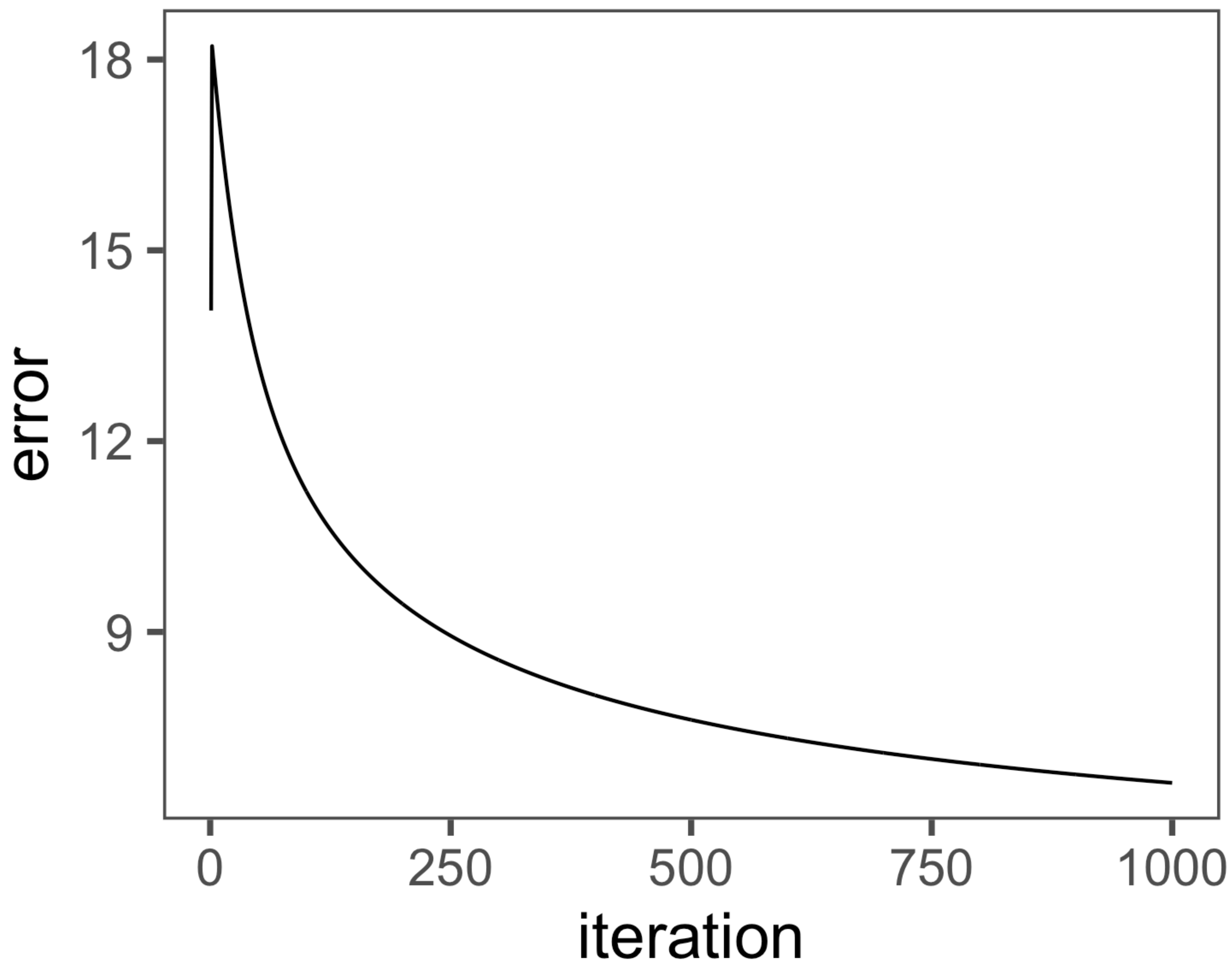
```
glm(Species ~ Sepal.Length + Sepal.Width
              + Petal.Length + Petal.Width,
              family = “binomial”)
```



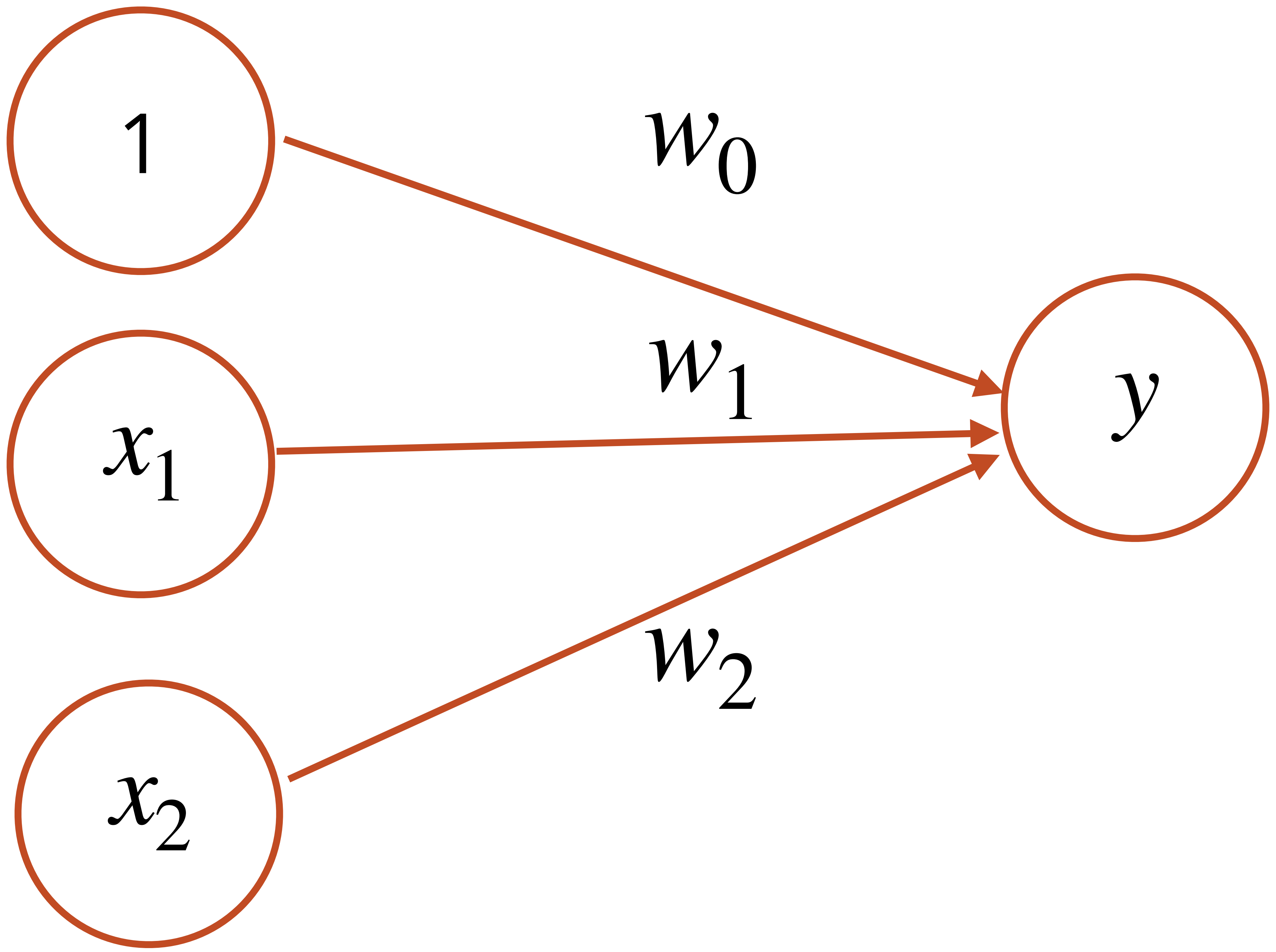
term	estimate	std.error	statistic	p.value
(Intercept)	-42.638	25.707	-1.659	.097
Sepal.Length	-2.465	2.394	-1.030	.303
Sepal.Width	-6.681	4.480	-1.491	.136
Petal.Length	9.429	4.737	1.991	.047
Petal.Width	18.286	9.743	1.877	.061

Learning an iris classifier

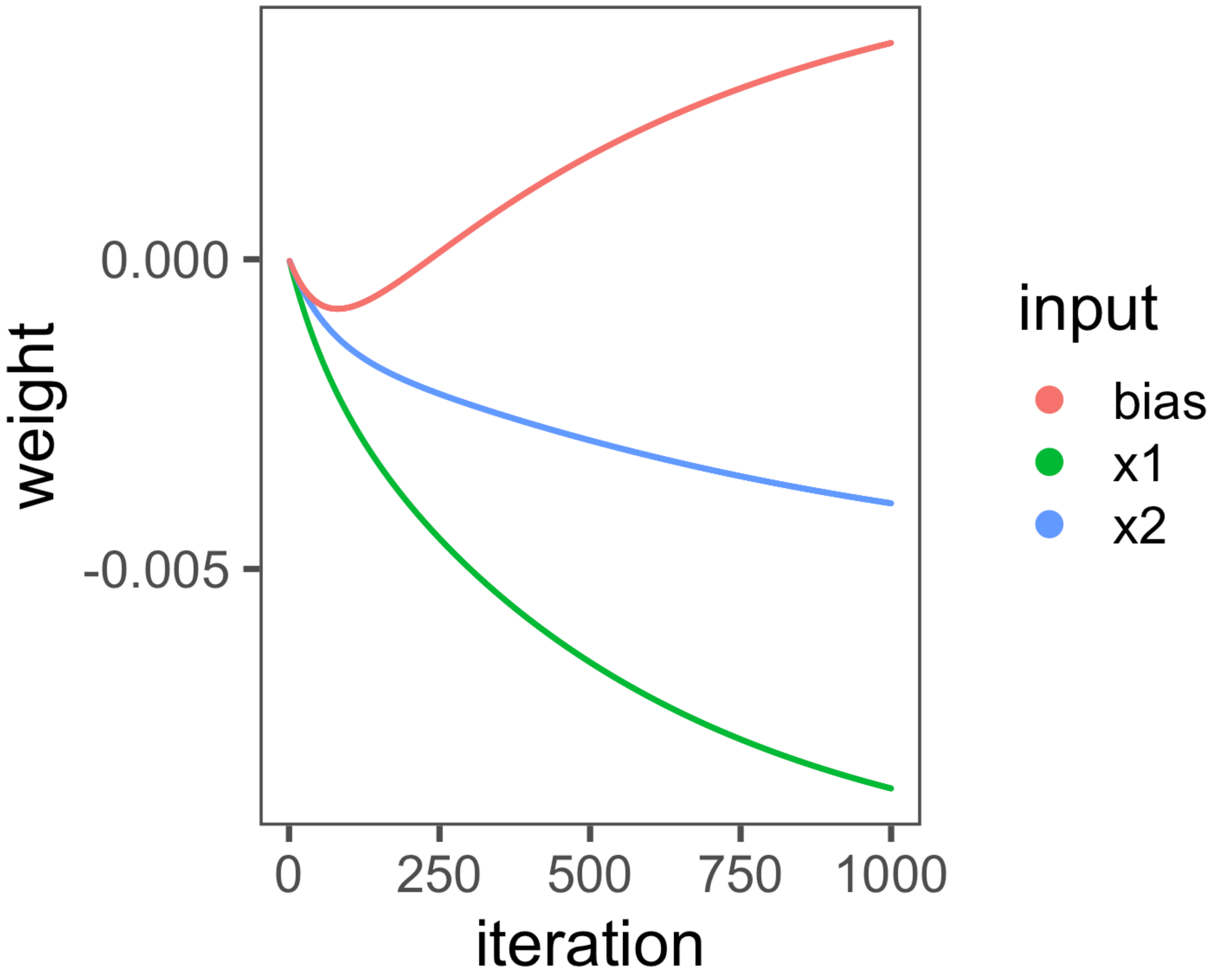


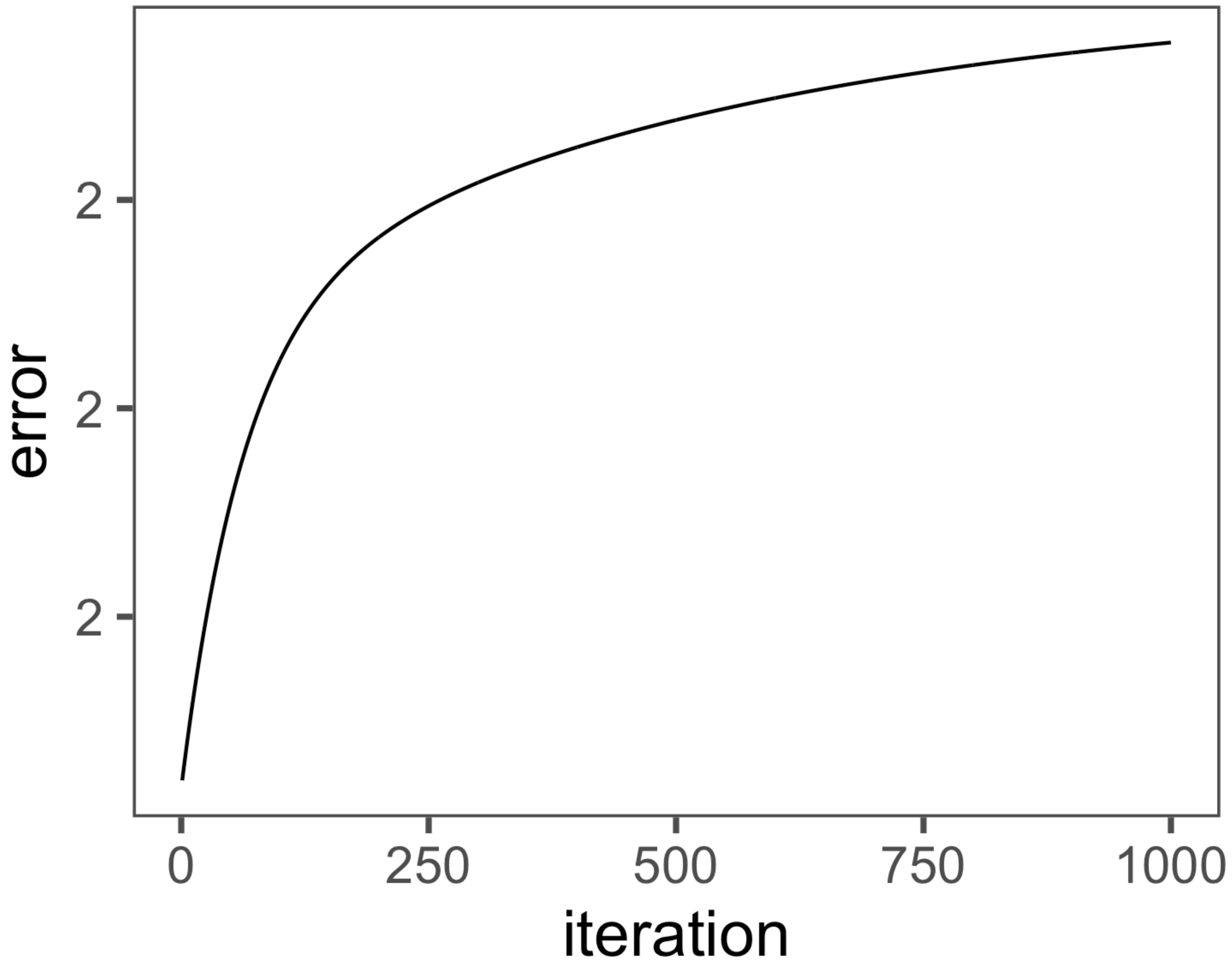


Building an XOR network



Learning the XOR function





Why can’t this network learn XOR?

How would regression solve xor?

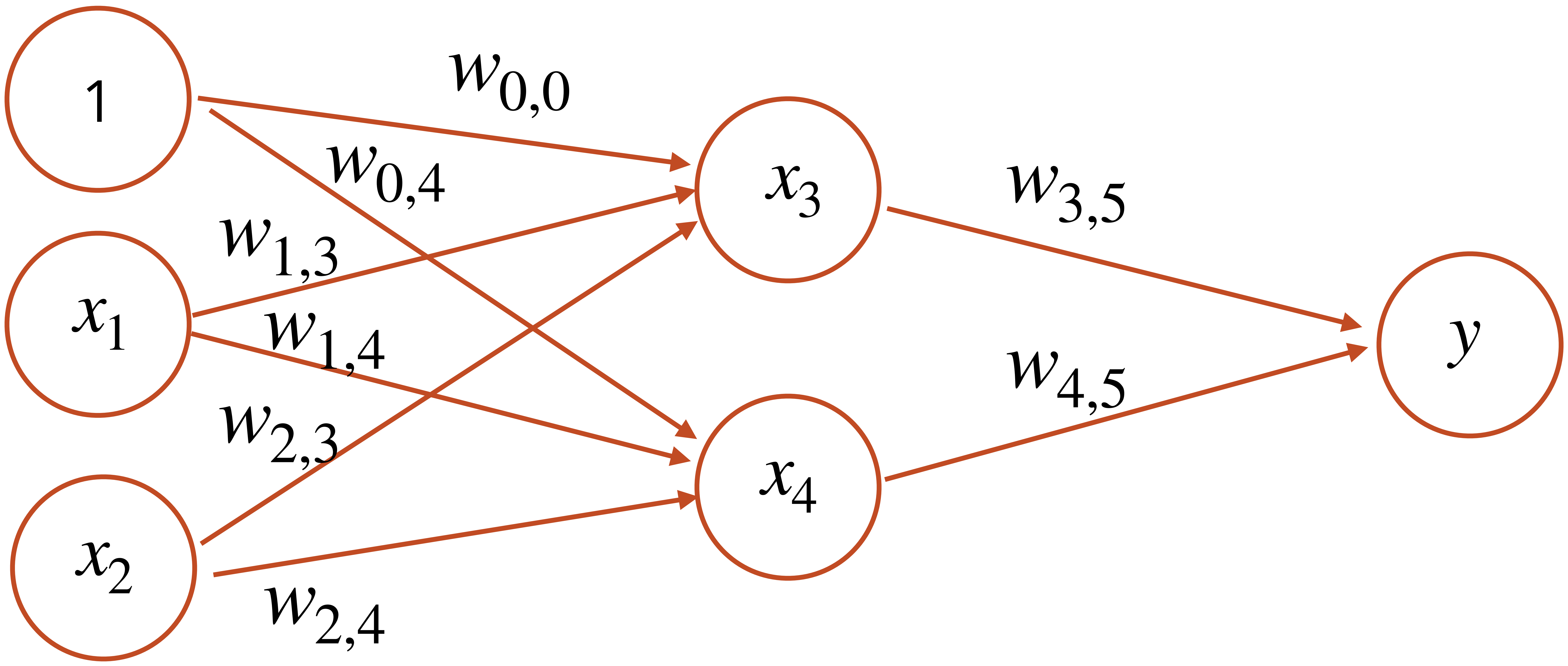
```
glm(y ~ x1 + x2, family = “binomial”)
```

```
glm(y ~ x1 * x2, family = “binomial”)
```

Need an $\mathbf{x_1 x_2}$ term!

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

How would a network solve xor?



x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Perceptrons

1. Simple neural networks generalize the Rescorla-Wagner model of associative learning

2. Perceptrons are general-purpose linear classifiers. They can solve lots of problems

3. But they can't solve all problems…