Summer AI

A very very simple neuron

Today

We’ll discuss a very very simple neuron …

Neurons

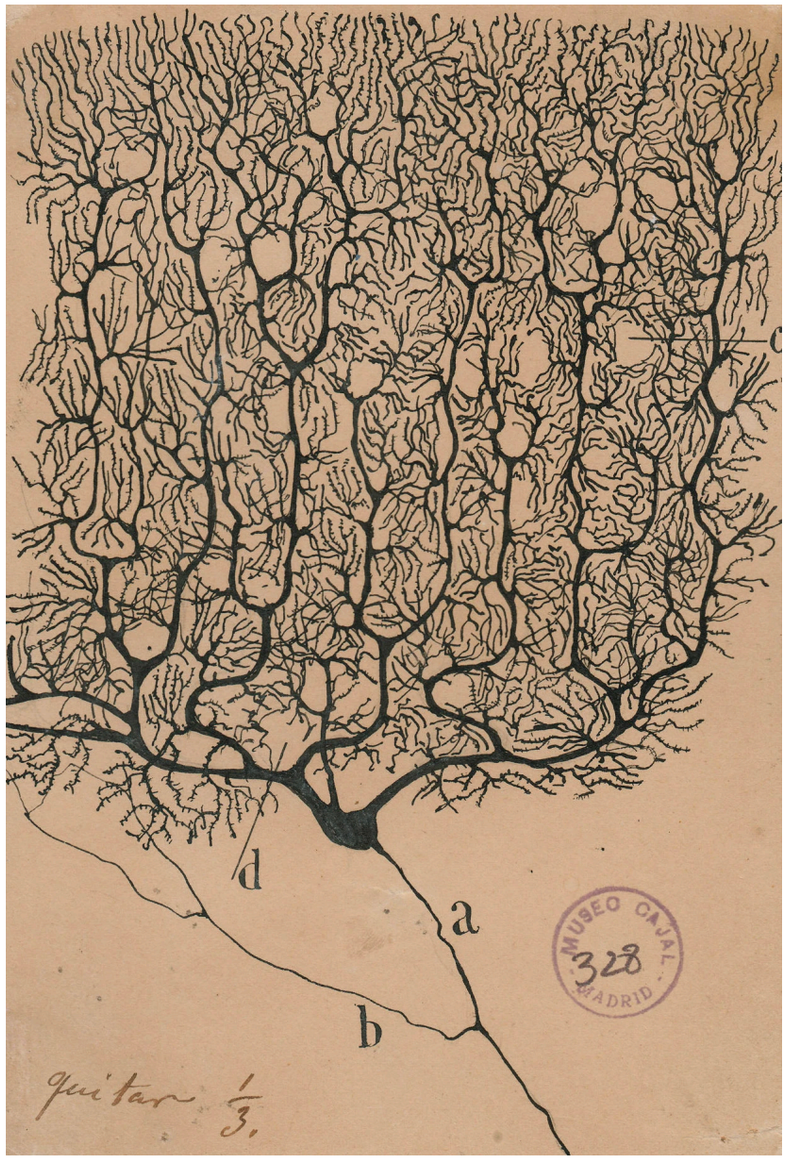
are in your brain.

you have **lots** of them.

- 86 billion neurons in the human brain.
- number of stars in the Milky Way 200 and 400 billion.

are responsible for your actions, thoughts, memories, …

are not completely understood

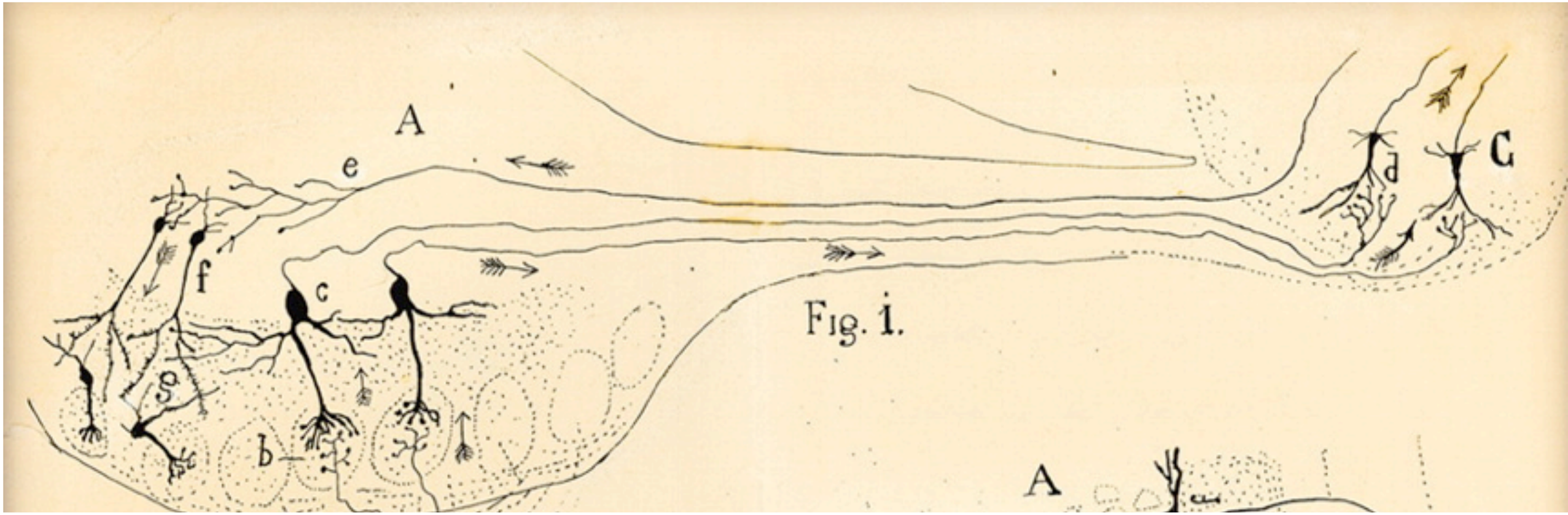


A neuron, conceptually

Conceptually, a neuron:

- receives inputs
- processes those inputs
- generates an output.

In practice, it's really complicated …

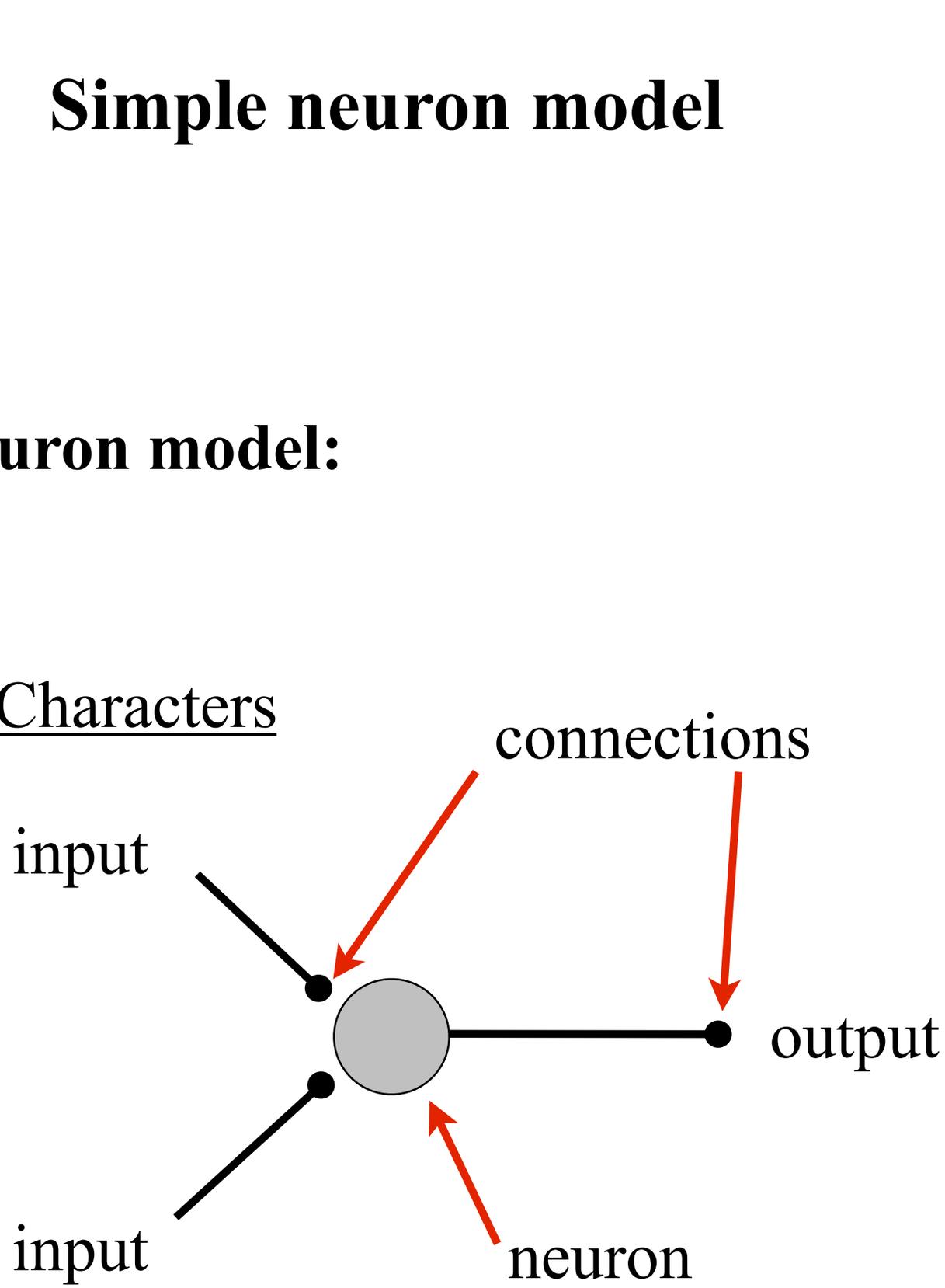


Schematic flow of current in visual system [Cajal]

Simple neuron model

Here, we'll simplify.

Consider **a simple neuron model:**



Q: What's been lost here?

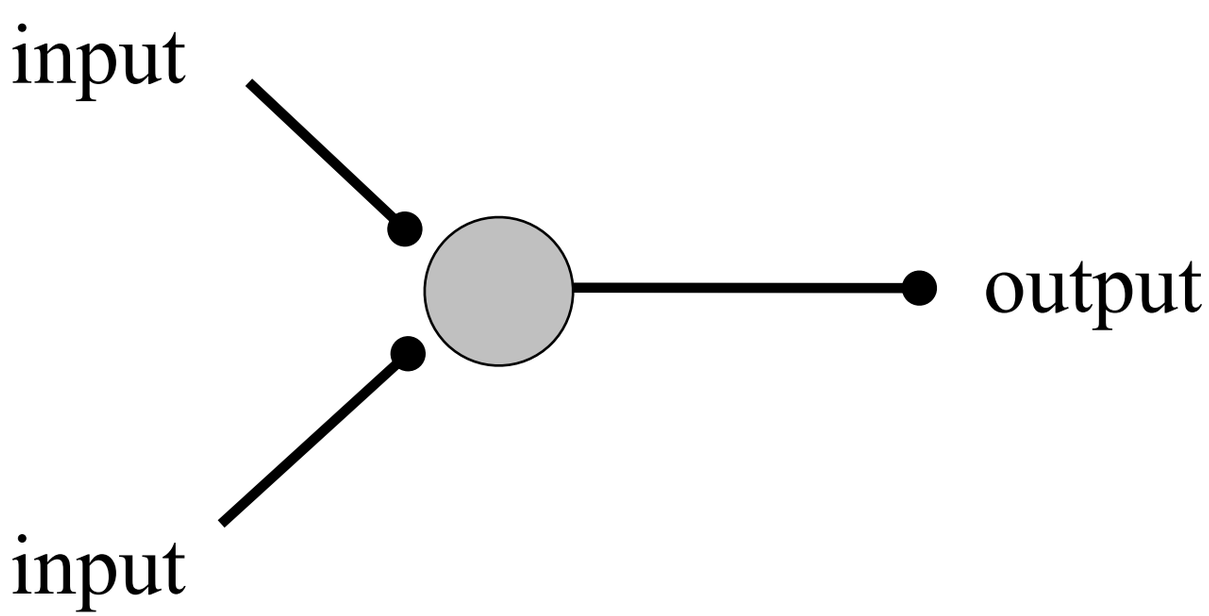
The "simplest" information processor

The Perceptron

– a very very simple neuron

Three elements:

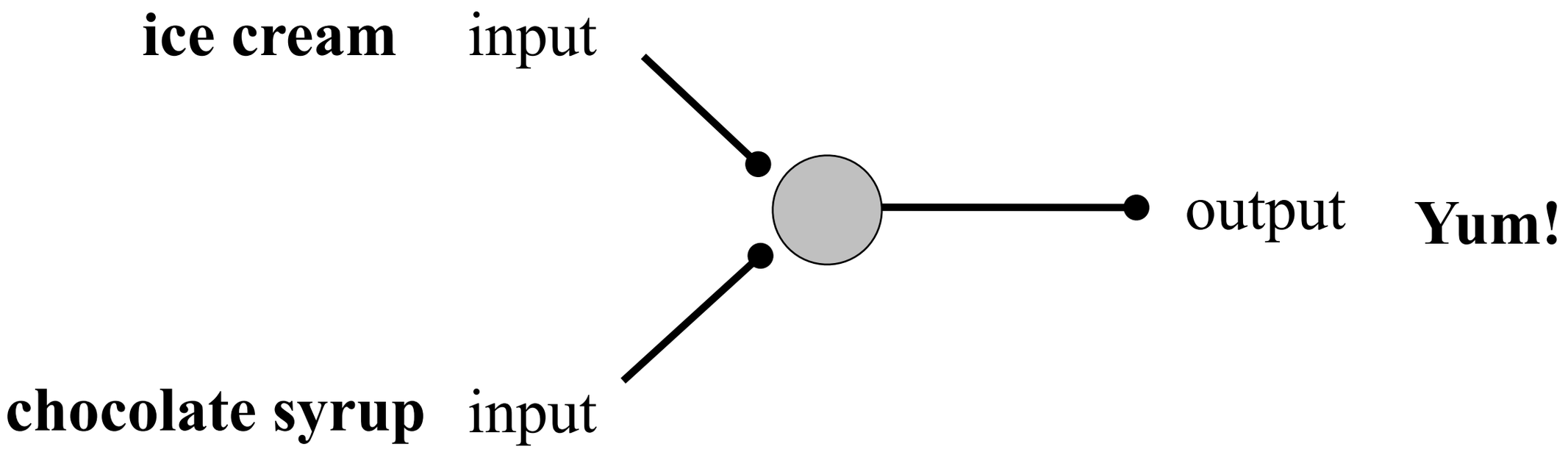
1. input(s)
2. "processor"
3. single output

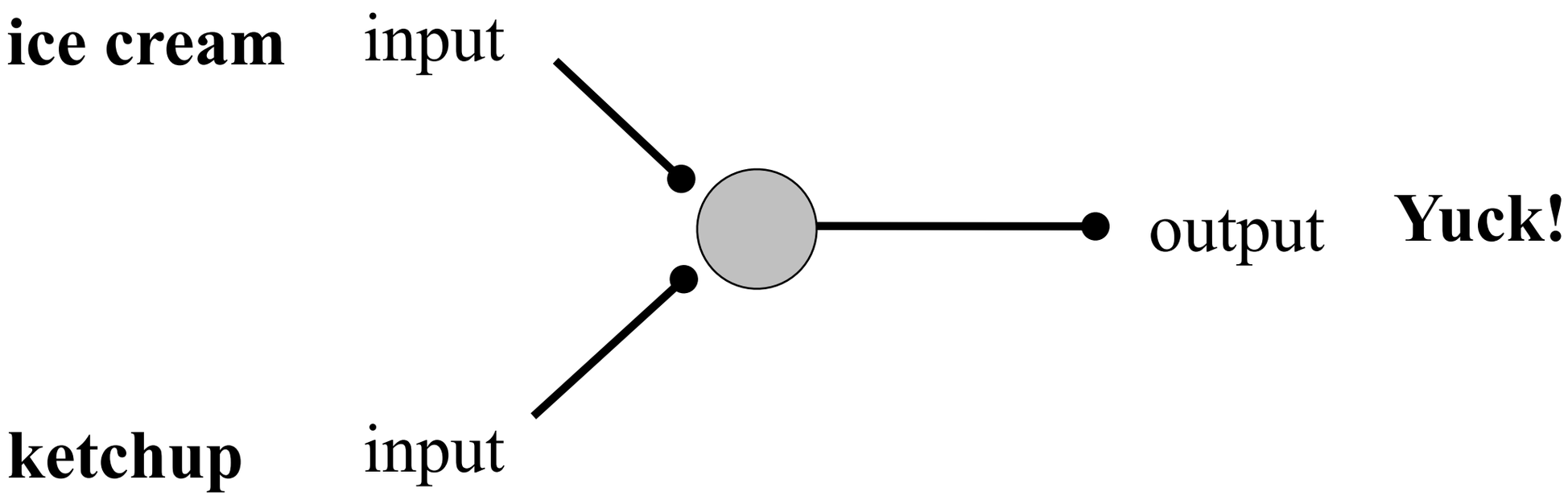


Feed-forward model progresses from left to right

input comes in, gets processed, output goes out

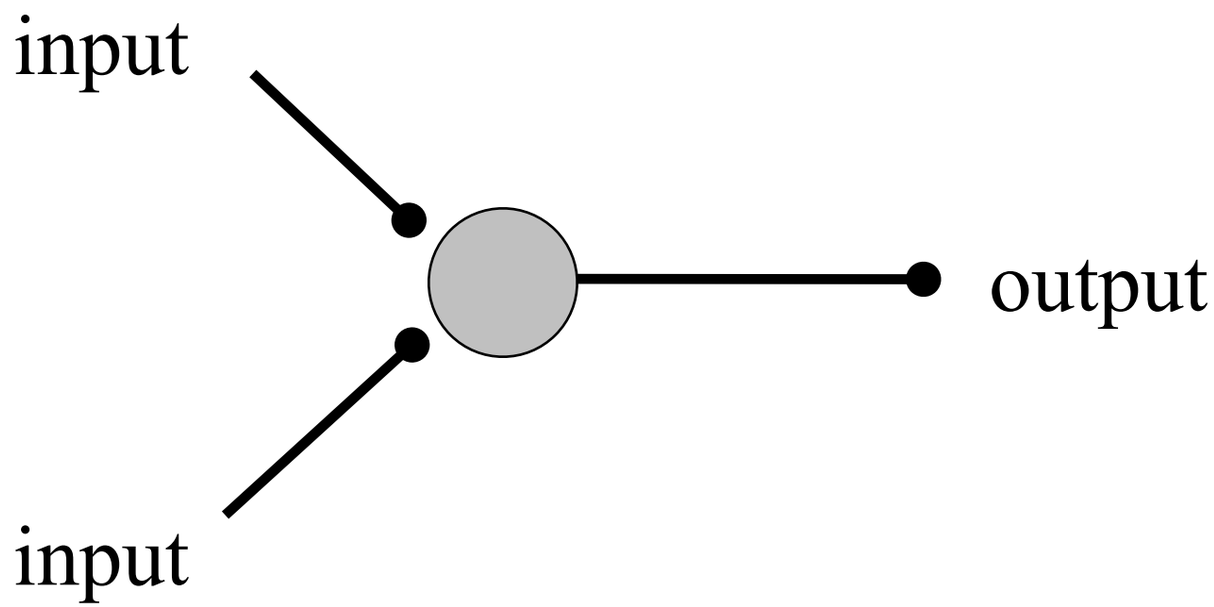
Perceptron: can do important things





Example with numbers …

The “simplest” information processor



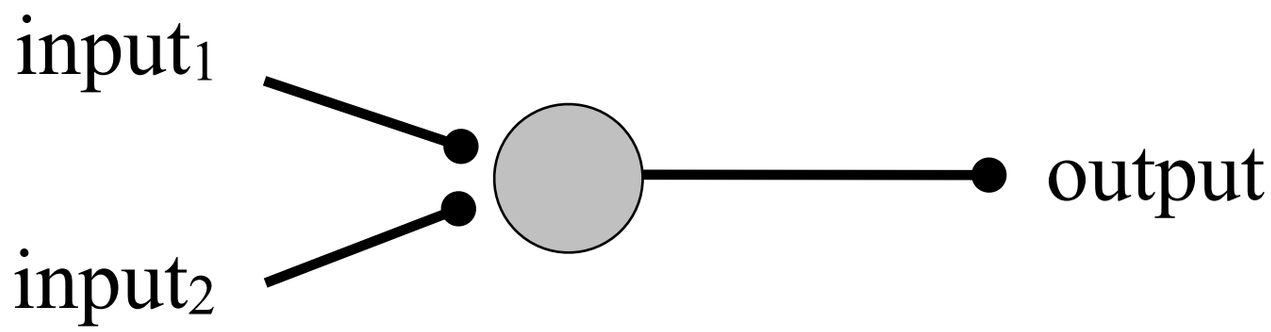
Divide information processing into 4 steps:

1. Receive inputs
2. Weight inputs
3. Sum weighted inputs
4. Generate output

Let’s go through each step, in a concrete example …

4 steps of information processing (Step 1)

Step 1. Receive inputs.



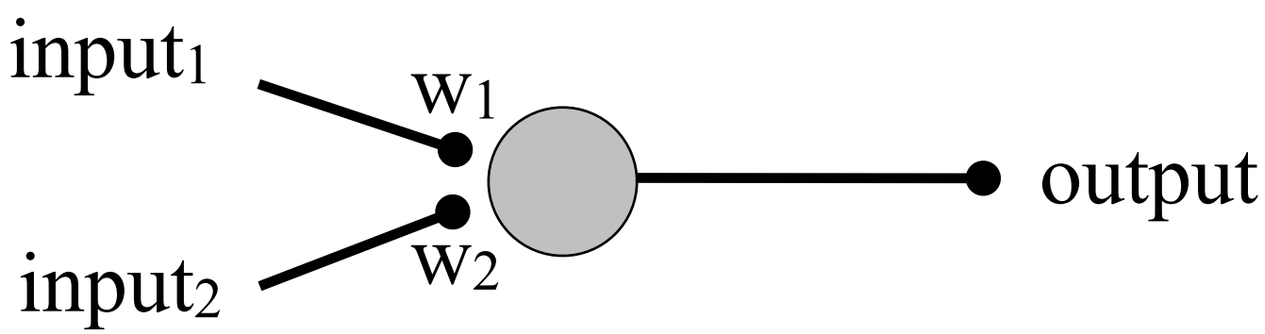
Example: a perceptron with two inputs.

Let's define: $input_1 = 12$

$input_2 = 4$

4 steps of information processing (Step 2)

Step 2. Weight inputs.



Each input sent to the neuron is **weighted**
= multiplied by some number.

Example: Let's define: $w_1 = 0.5$
$w_2 = -1$

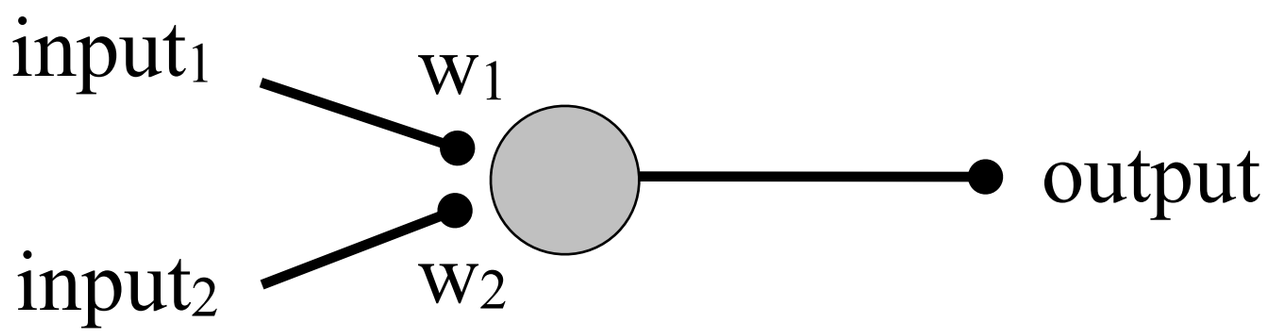
Now, "weight inputs": multiply each input by its weight.

$input_1 * w_1 = 12 * 0.5 = 6$

$input_2 * w_2 = 4 * -1 = -4$

4 steps of information processing (Step 3 & 4)

Step 3. Sum weighted inputs



$$input_1 * w_1 + input_2 * w_2 \quad = 6 + (-4) \quad = 2$$

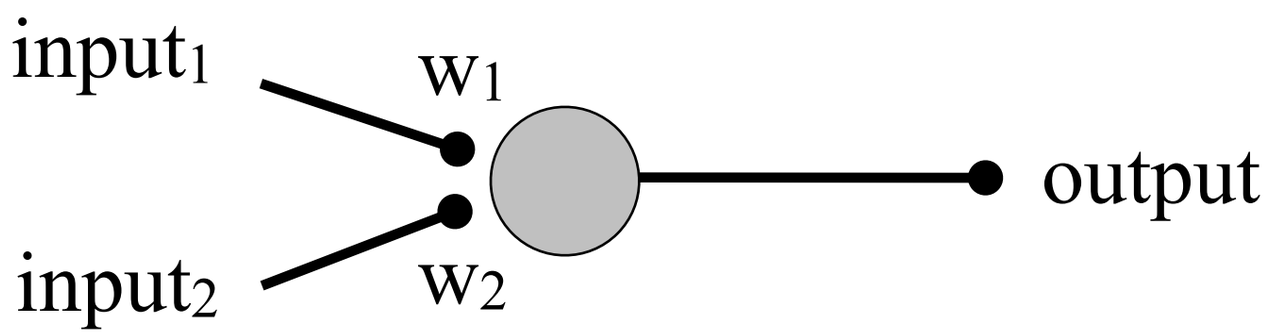
Step 4. Generate output.

Q: How?
A: Pass the summed weighted inputs through a special rule:

If the summed weighted input is “big enough”, then “fire”.

The Perceptron Algorithm

Summary:



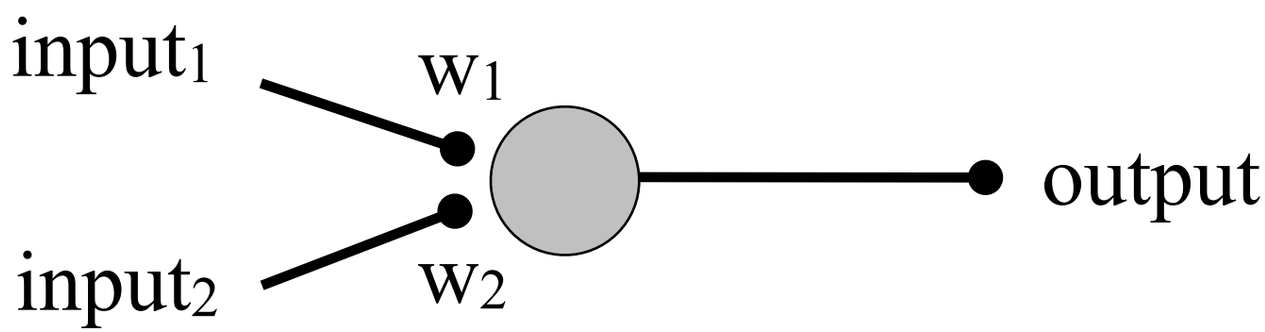
1. For every input, multiply that input by its weight.

2. Sum all of the weighted inputs

3. Compute the output of the perceptron based on that sum passed through a special rule.

(what is this rule?)

The special rule



If the summed weighted input is “big enough”, then “produce an output”

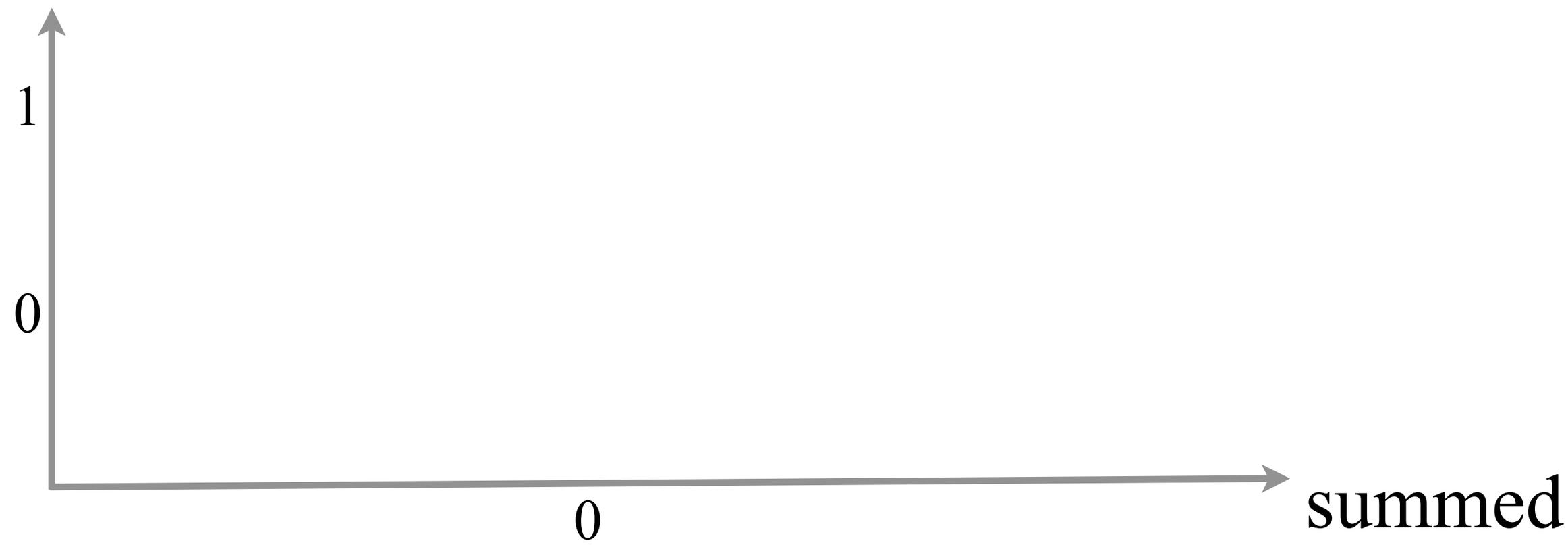
$$output = 0 \quad \text{if} \quad \text{input}_1 * w_1 + \text{input}_2 * w_2 < 0$$
$$output = 1 \quad \text{if} \quad \text{input}_1 * w_1 + \text{input}_2 * w_2 > 0$$

The special rule

Draw it

$$output = 0 \quad \text{if} \quad [\text{input}_1 * w_1 + \text{input}_2 * w_2] < 0$$

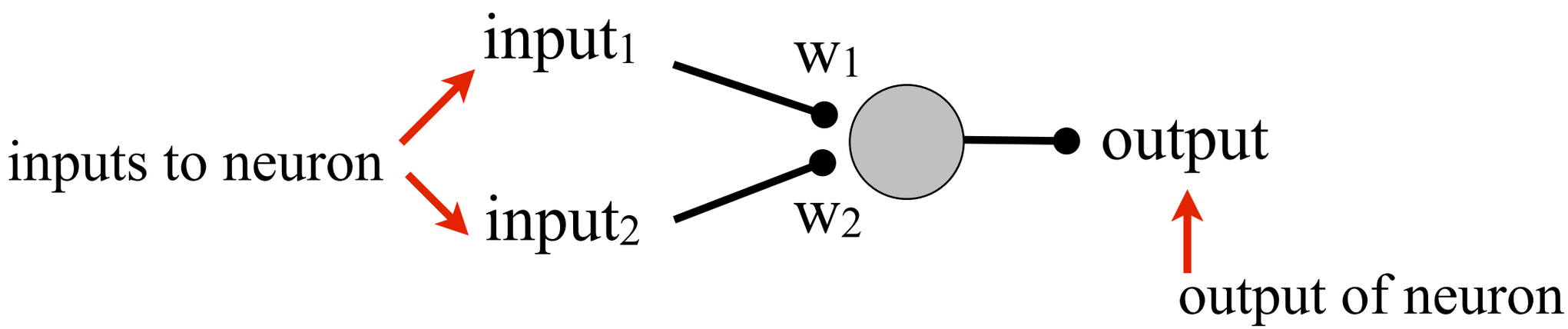
$$output = 1 \quad \text{if} \quad [\text{input}_1 * w_1 + \text{input}_2 * w_2] > 0$$



Q: What is the interpretation in words?

A: When neuron activity is large enough, it produces an output.

The perceptron: simple example



Ex. Inputs to neuron: $input_1 = 1$ $input_2 = 0$

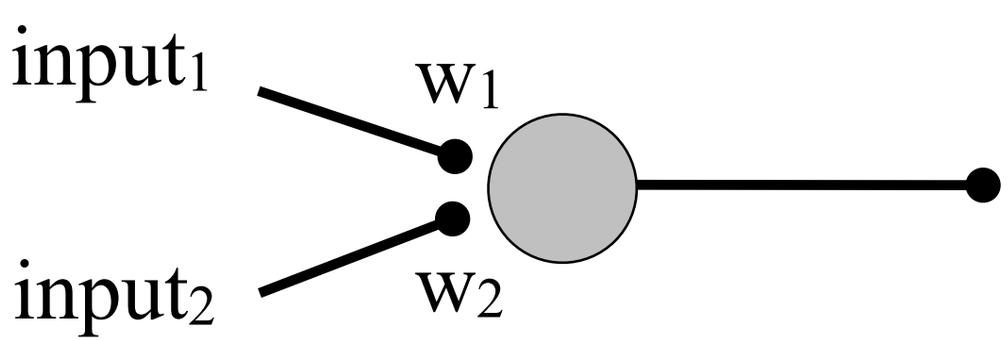
Connection weights: $w_1 = 0.5$ $w_2 = -0.5$

Q: What is *output ?*

$x = input_1\, w_1 + input_2\, w_2$

$x = 1*0.5 + 0*(-0.5) = 0.5$ $x > 0$ so $output = 1$

The perceptron: more generally



input1 = ice cream

input2 = ketchup

output = Yum or Yuck

What would you choose for weights?

How would you program this in Python?