

Introduction to Neural Networks

Fundamentals of Artificial Intelligence

Fabien Cromieres

Kyoto University

<http://lotus.kuee.kyoto-u.ac.jp/~fabien/lectures/IA/>

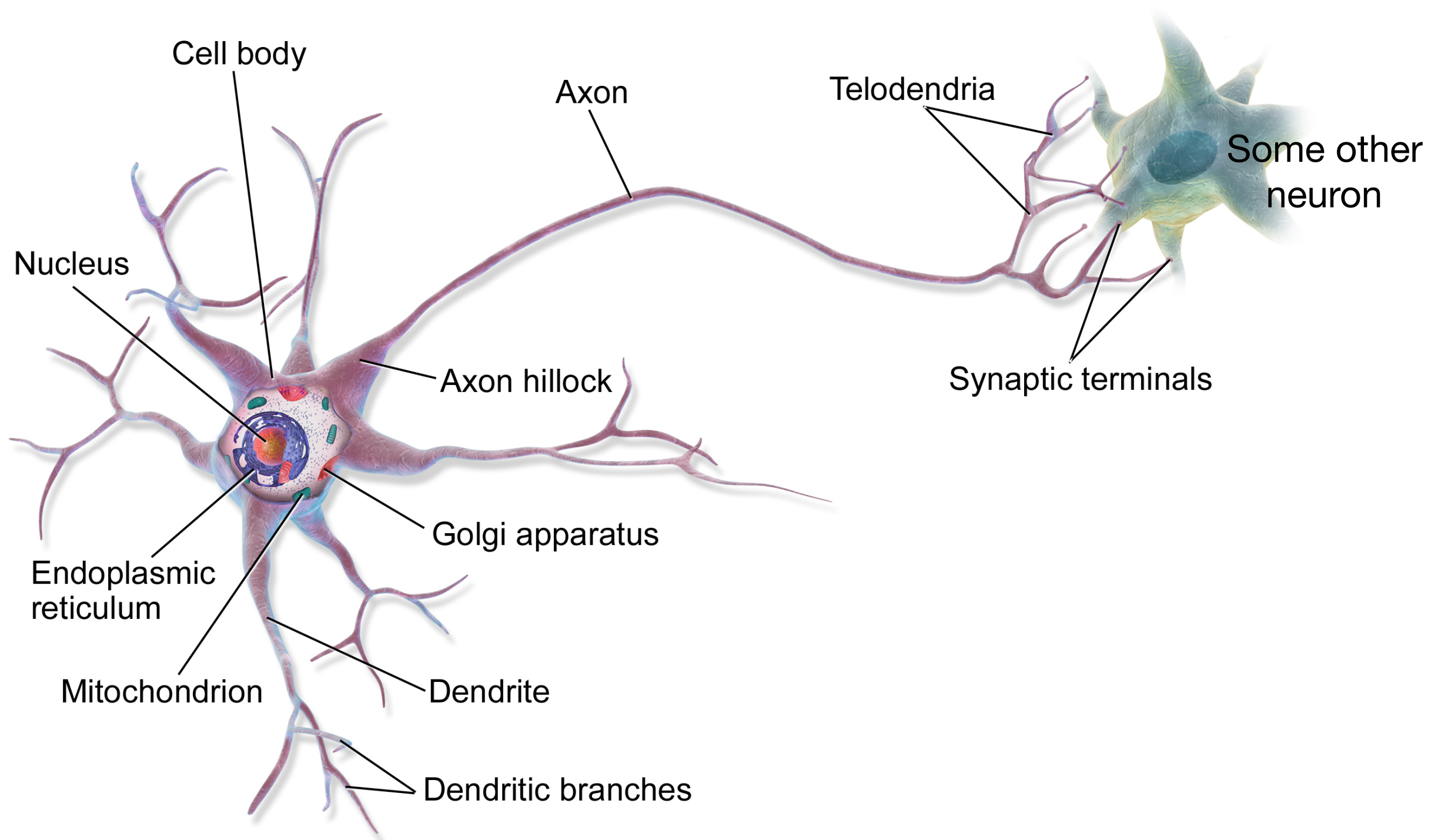
Introduction to Neural Networks

- Finally, the fancy part!
- **Neural Networks** are currently the most efficient and most used models for Machine Learning (and for AI in general, actually)
- Today, we discuss neurons, both the **biological neurons** and the **artificial neurons**

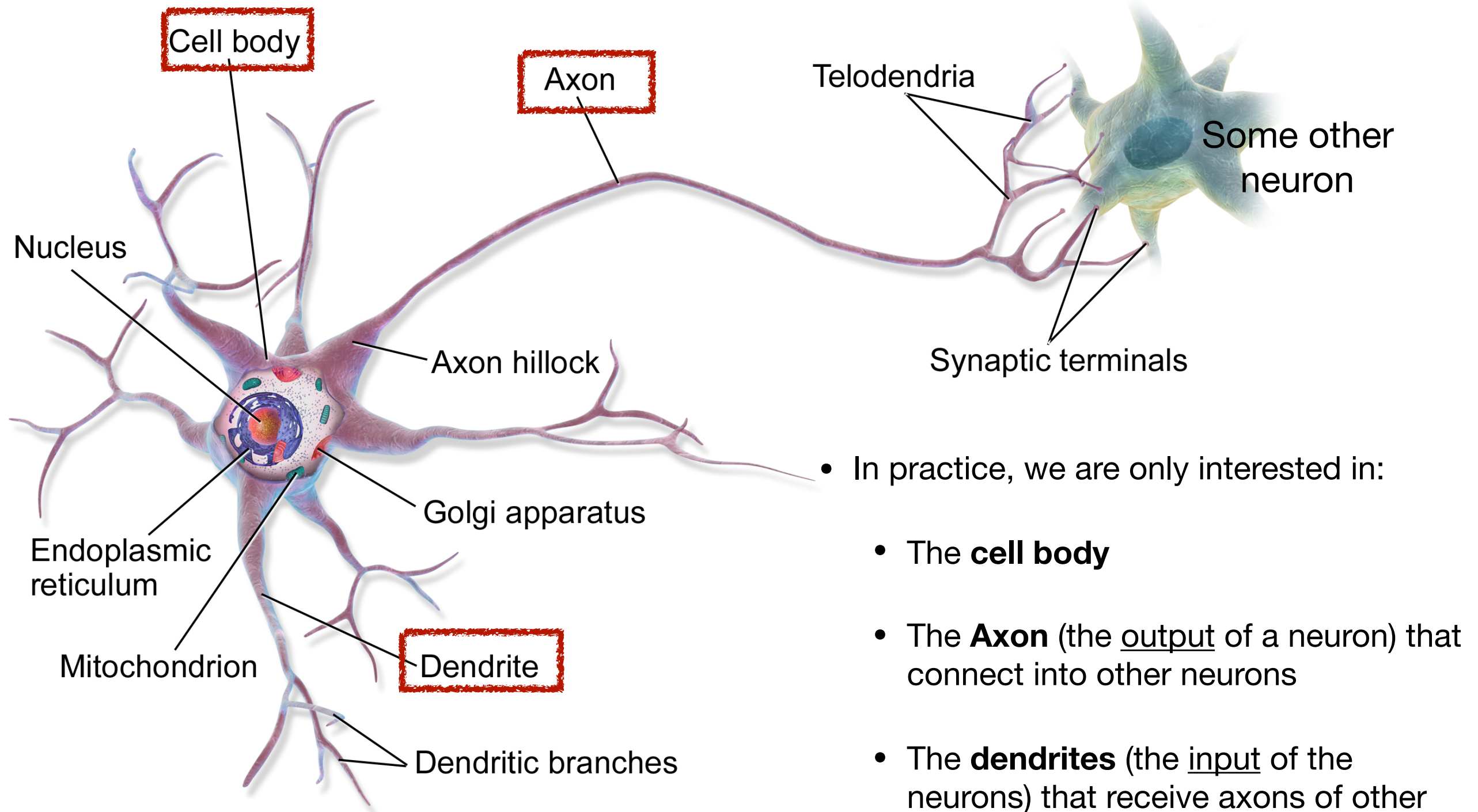
The real (biological) Neurons

- We are going to spend a good amount of time discussing **biological neurons**
- But keep in mind that the goal of AI is not to simulate precisely a human brain
- The goal is to have software that do what we want (image recognition, etc.)
- It is actually perfectly possible to explain **Artificial Neural Networks** without mentioning how they relate to the brain
- But I think it is very interesting to see the parallel

Anatomy of a neuron



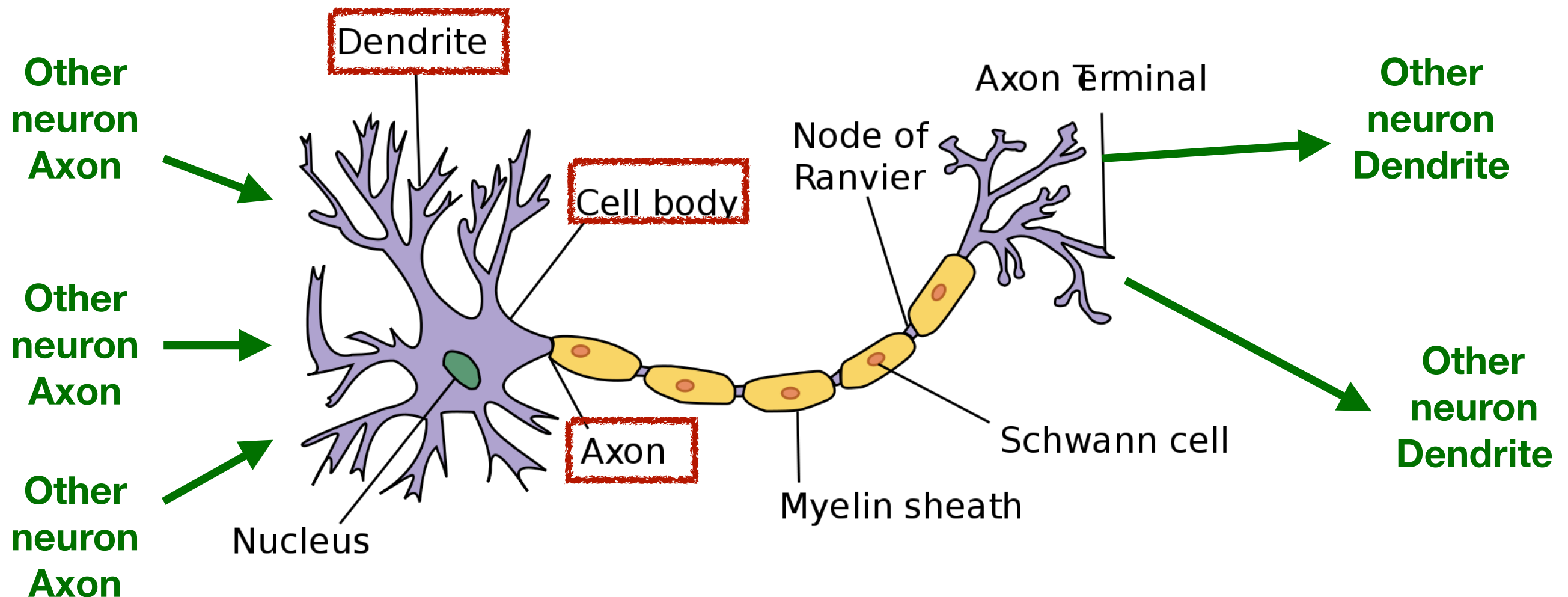
Anatomy of a neuron



- In practice, we are only interested in:
 - The **cell body**
 - The **Axon** (the output of a neuron) that connect into other neurons
 - The **dendrites** (the input of the neurons) that receive axons of other neurons

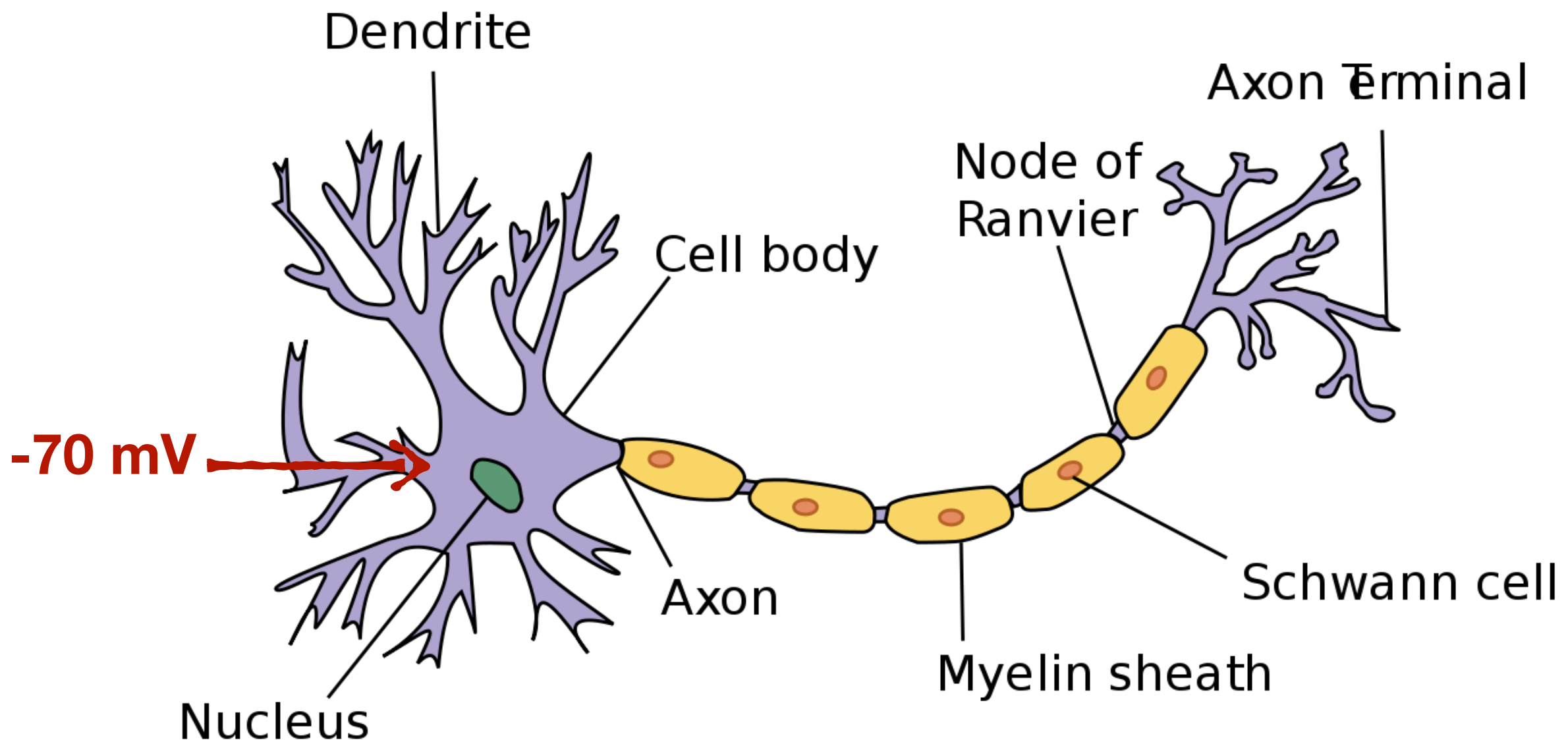
What a neuron do

- The neuron receive input from other neurons from its **dendrites**
- If the sum of the input is above a certain **threshold**, it sends a signal (called *Action Potential*) on its **axon** to other neurons



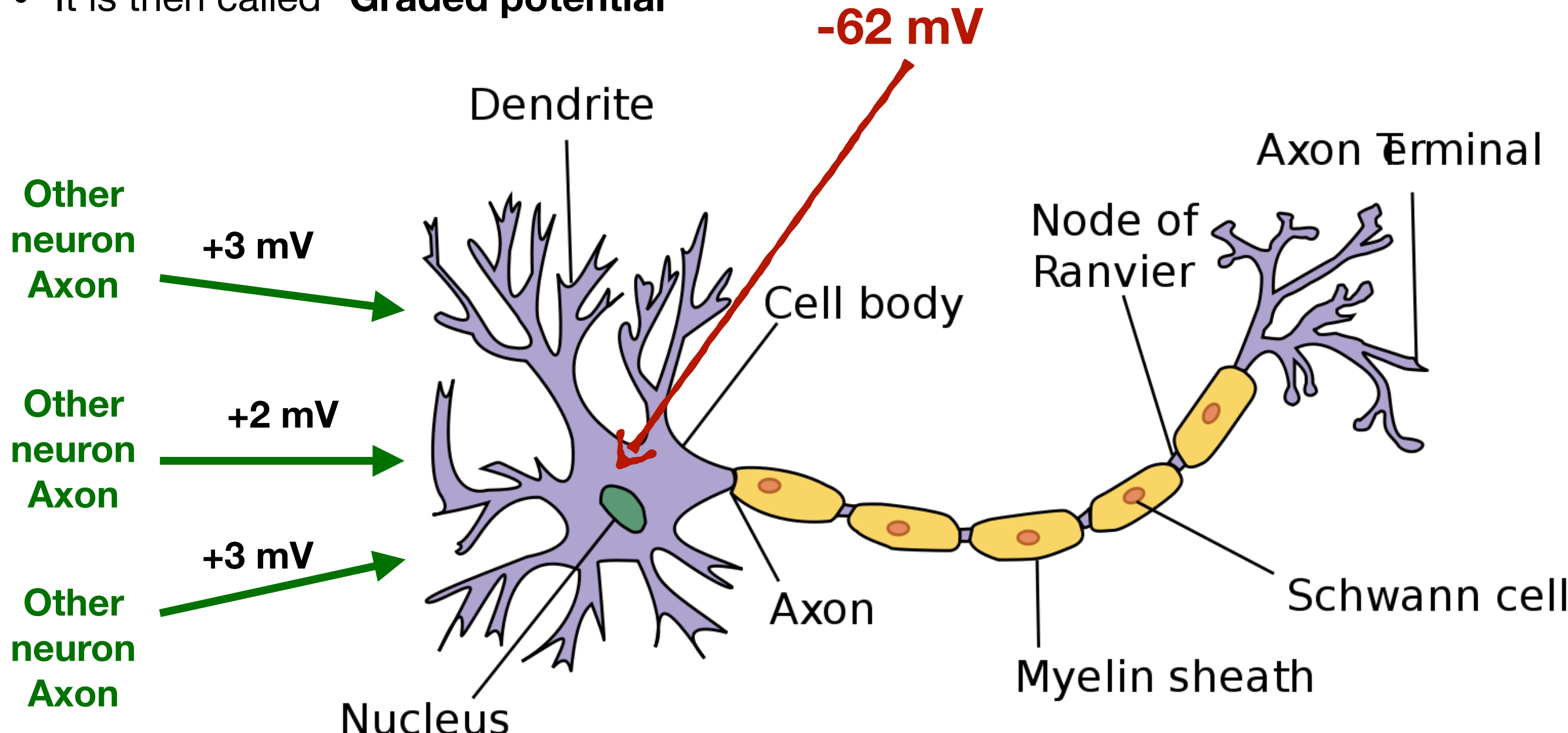
What a neuron do

- At rest, the inside of a neuron has an electric potential of -70mv
- It is the “**Resting Potential**”
- Chemically created by Potassium and Sodium ions



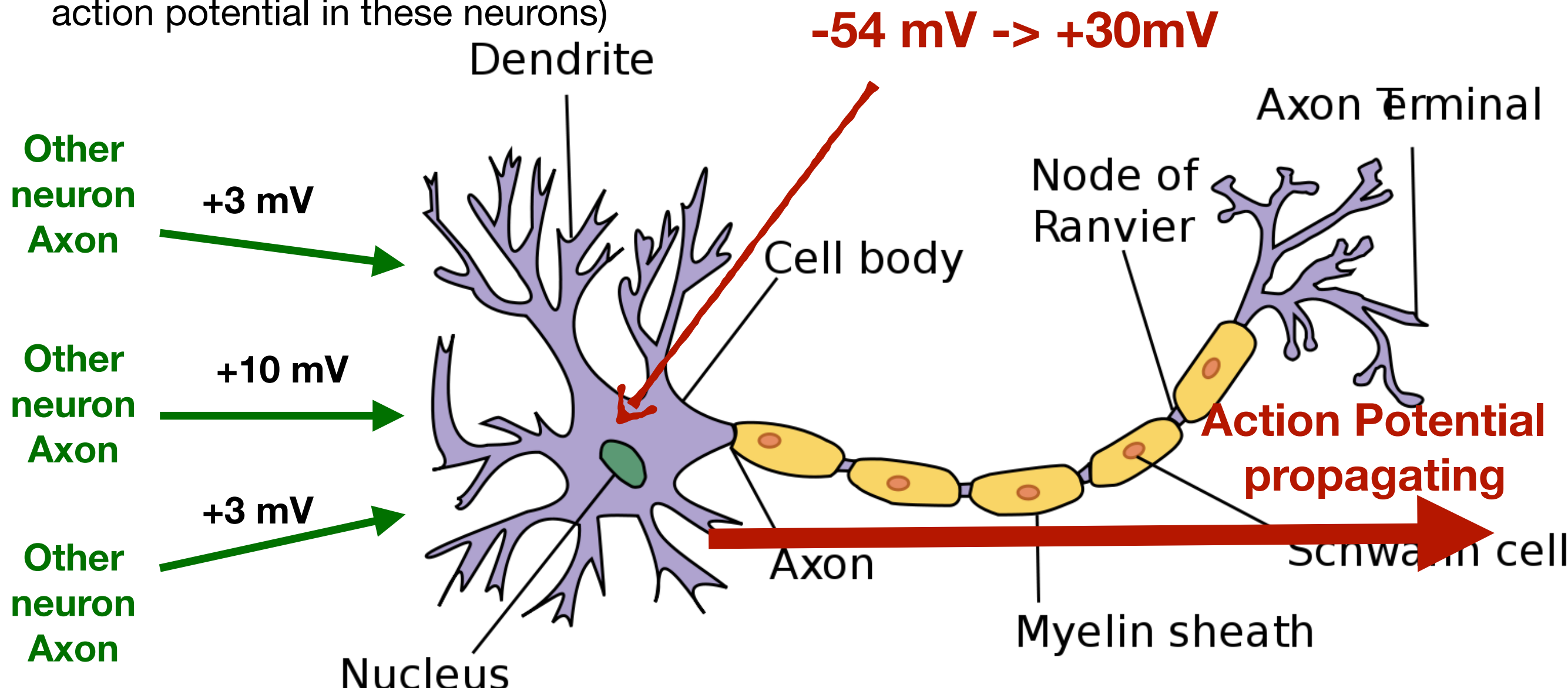
What a neuron do

- The neuron might receive electric potential from other neurons on its **dendrite**
- The potential inside the cell body can then increase
- It is then called “**Graded potential**”

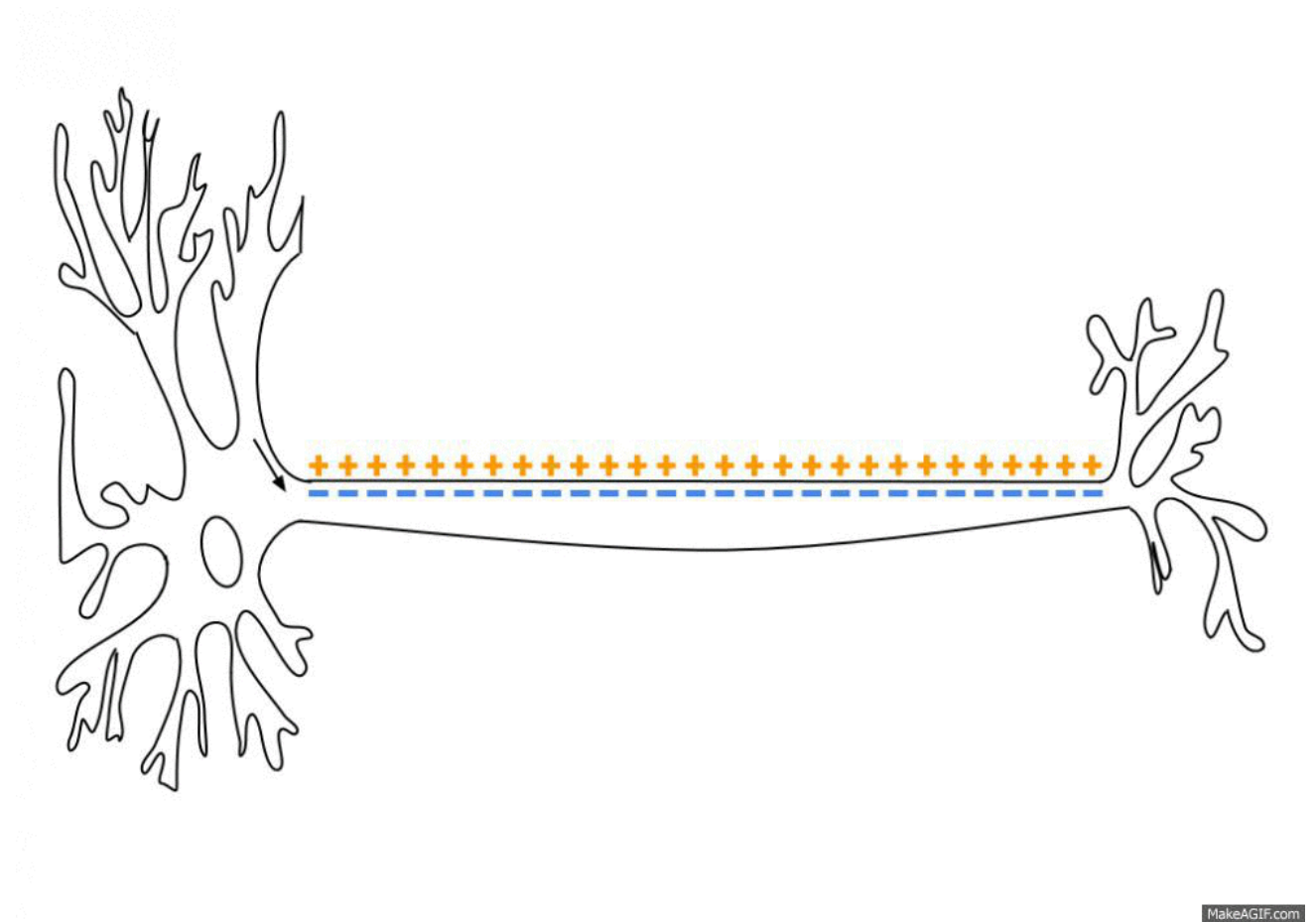
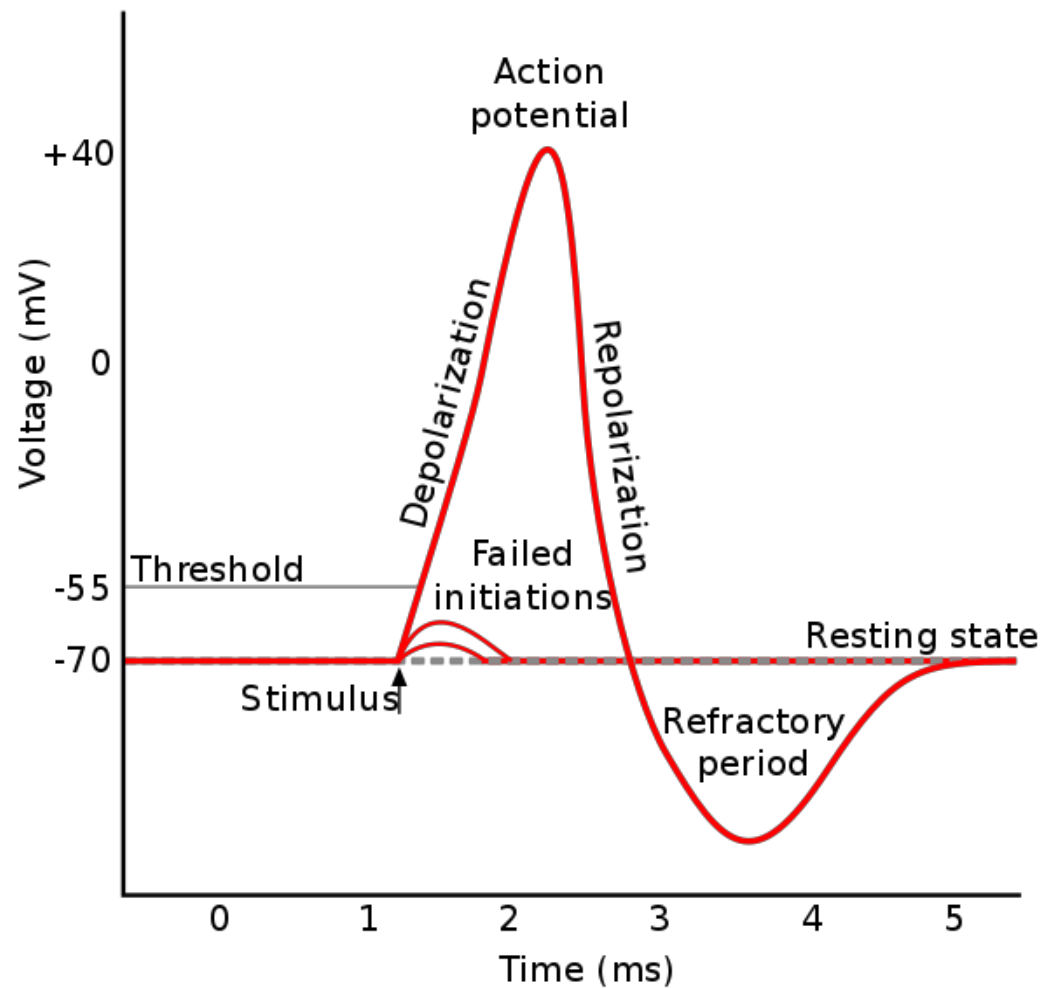


What a neuron do

- If the potential inside a neuron goes above -55mV something happens
- A chemical chain reaction will suddenly increase the potential to +30mV (called the **Action Potential**)
- This potential will propagate on the **axon** to other neurons (and may then trigger action potential in these neurons)

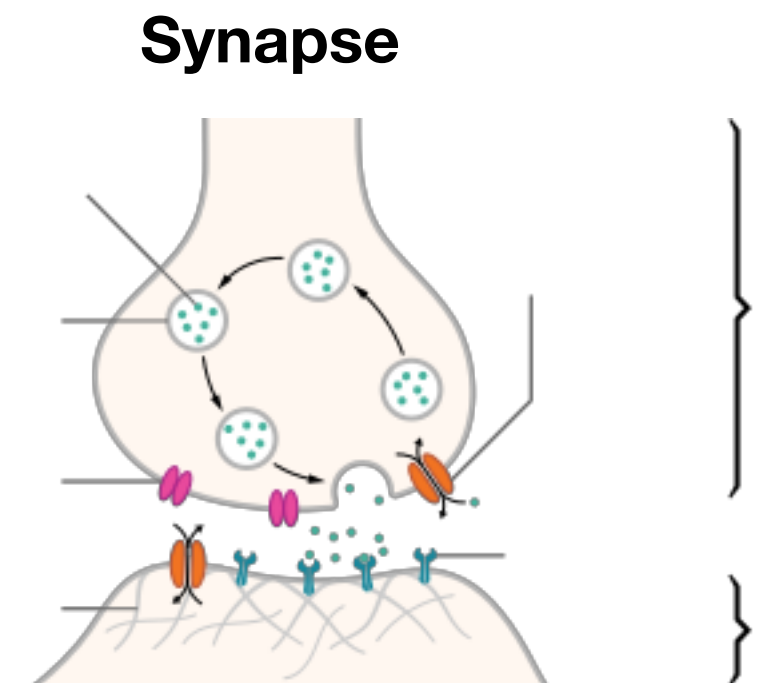
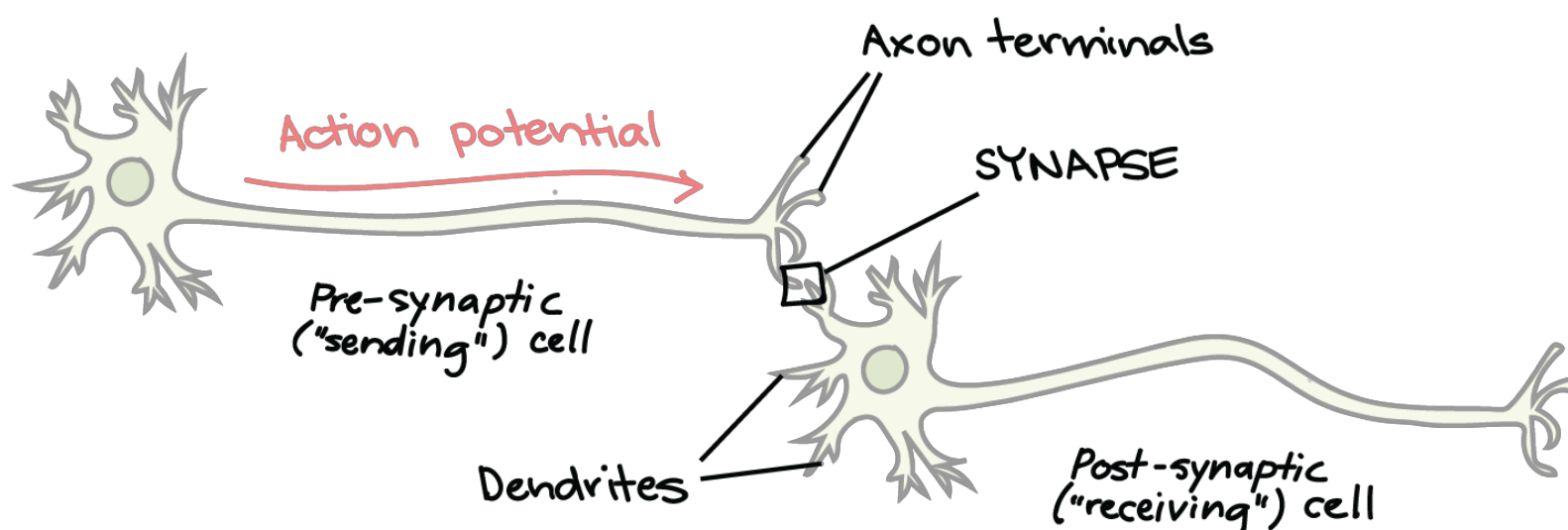


The Action Potential



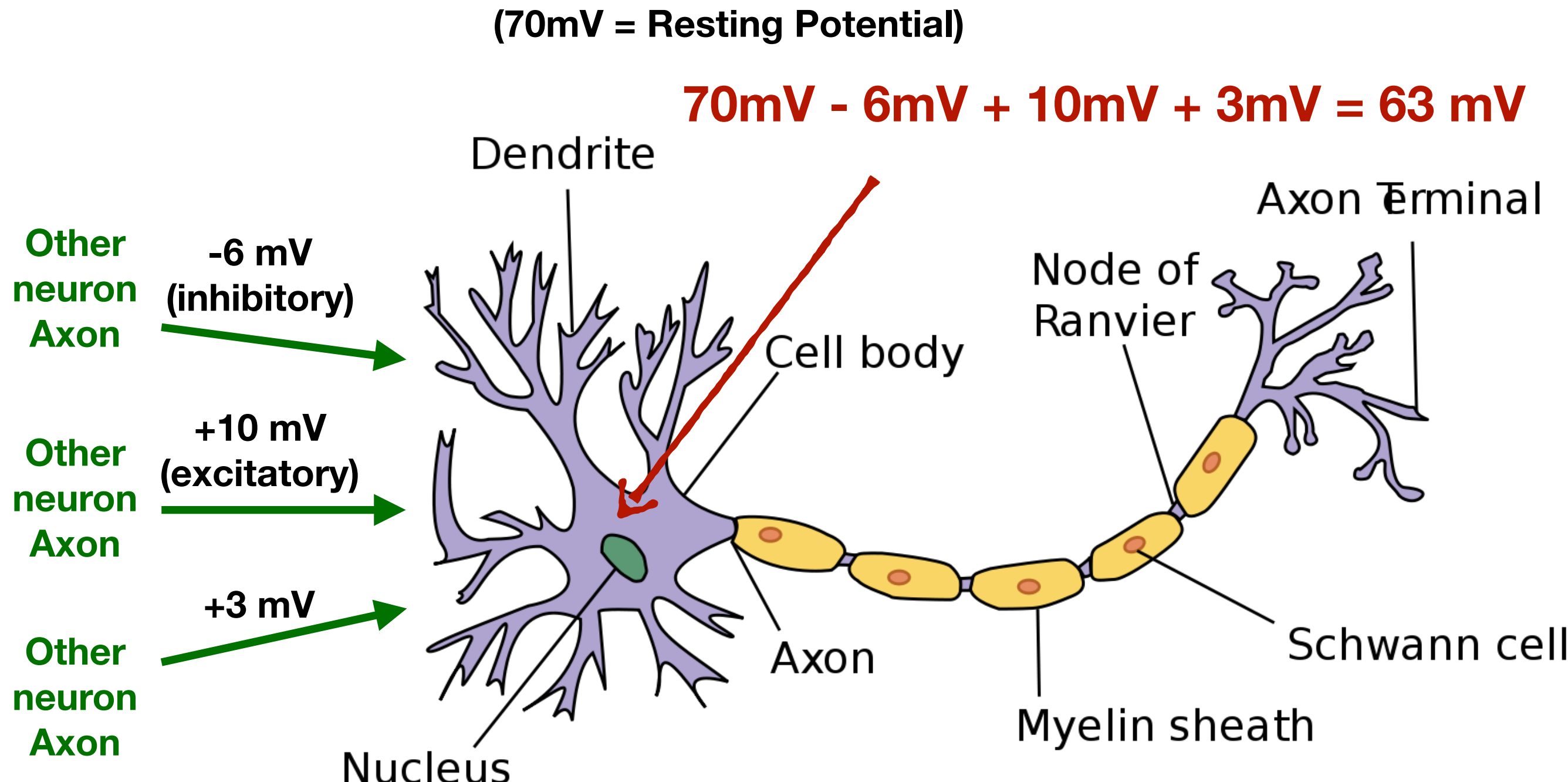
Synaptic connections

- The axon of a neuron is connected to the dendrites of other neurons through a “connector” called a **synapse**
- The actual potential passed to the dendrite will depend on the synapse
- There can be **excitatory synapses** or **inhibitory synapses**



Synaptic connection

- The type of synaptic connection influence the effect of the action potential of previous neurons

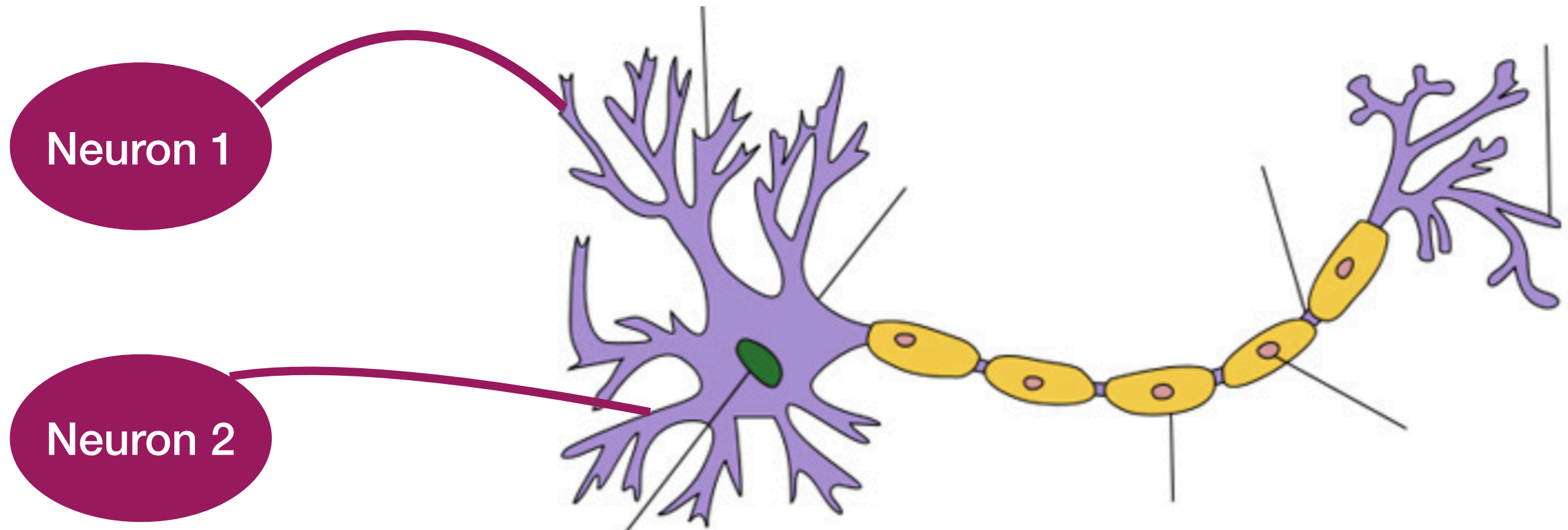


The biological process in a neuron

- For a biological view about how all of this is actually happening in the neurons, let us watch a 14 minutes video by Paul Andersen:
 - <https://www.youtube.com/watch?v=HYLyhXRp298>
- Lot of Biology and Chemistry in this video; but **do not worry if you do not understand everything**.

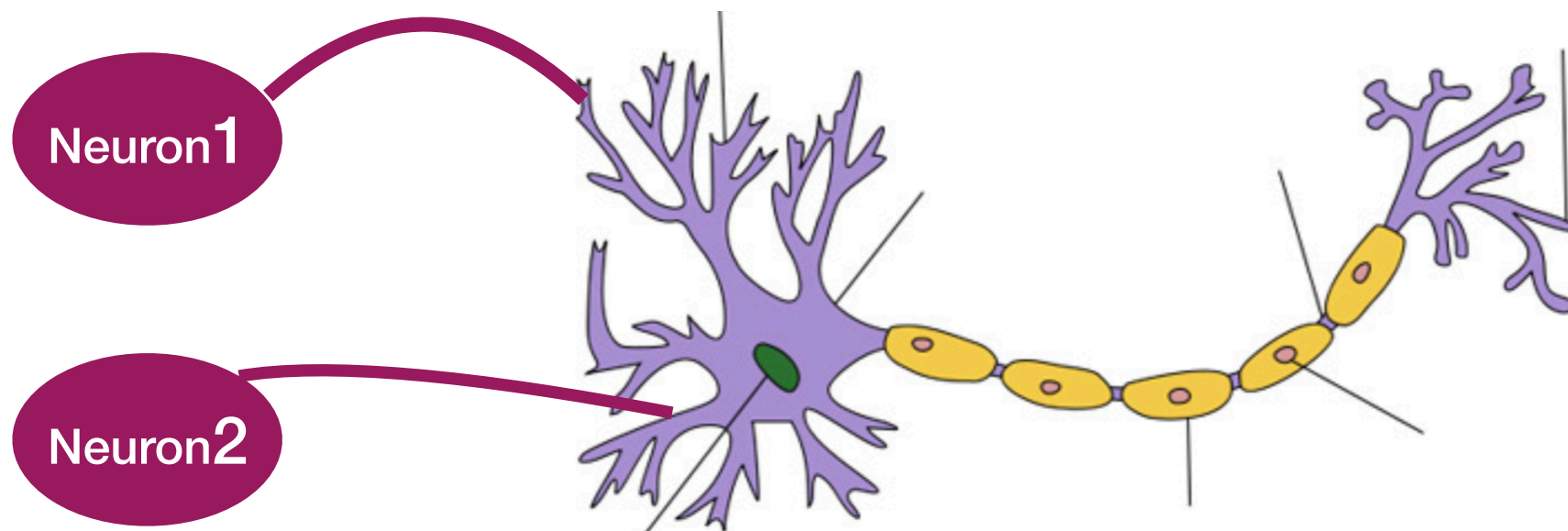
From biology to math and Machine Learning

- Now, let us try to describe “mathematically” what a neuron is doing
- Let us consider a neuron with only 2 dendrites connected to other neurons (2 inputs)



Equation of a Neuron with 2 dendrites

- Now, let us try to describe “mathematically” what this neuron is doing:
 1. Some input potentials arrives on its **dendrites**
 2. The potential are accumulated in the **body** (*graded potential*)
 3. If the *graded potential* is larger than a given threshold, the neuron fires an *action potential* on the **axone**



Equation of a Neuron with 2 dendrites

- Now, let us try to describe “mathematically” what this neuron is doing:

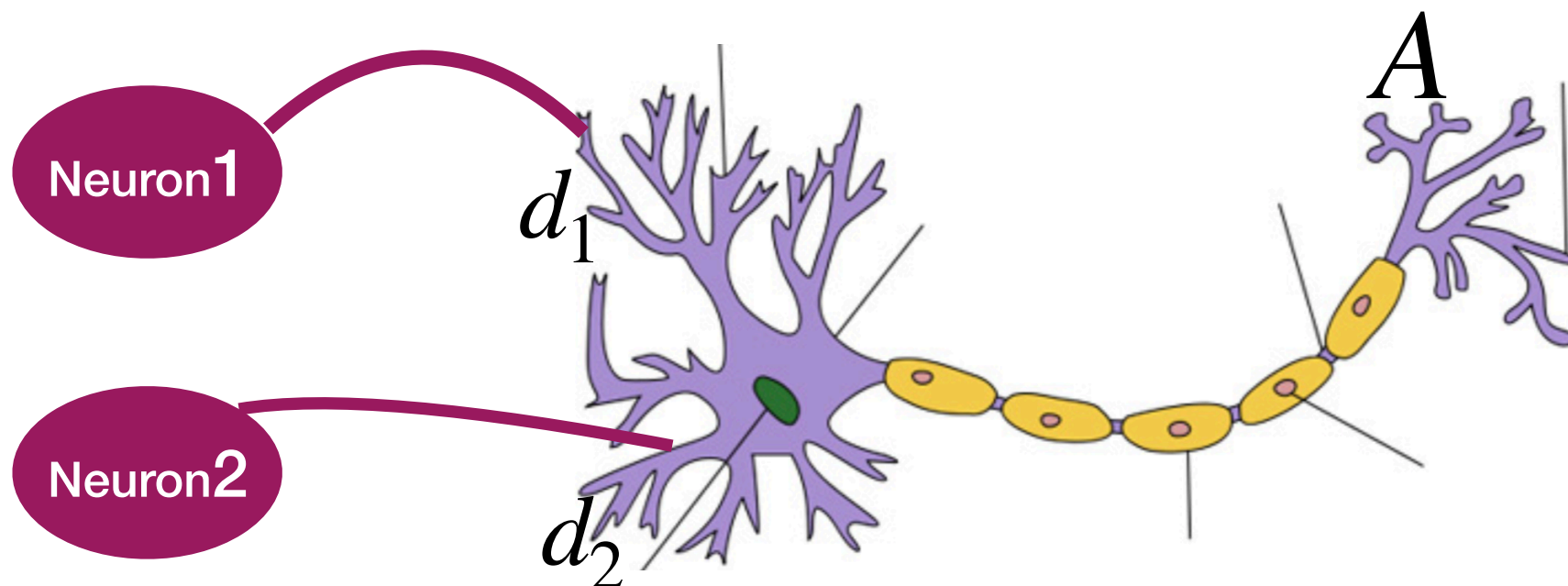
1. Some input potentials arrives on its **dendrites**

d_1 d_2

2. The potentials are accumulated in the **body** (*graded potential*)

$$\text{graded}P = -70mV + d_1 + d_2$$

3. If the *graded potential* is larger than a given threshold, the neuron fires an action potential on the **axone**



$$A = -70mV$$

$$\text{If } \text{graded}P \leq -55mV$$

$$A = +30mV$$

$$\text{If } \text{graded}P > -55mV$$

Equation of a Neuron with 2 dendrites

- The dendrite potential will depend on the **type of the synaptic connection**. Let us suppose the synaptic connection can be represented by a parameter θ

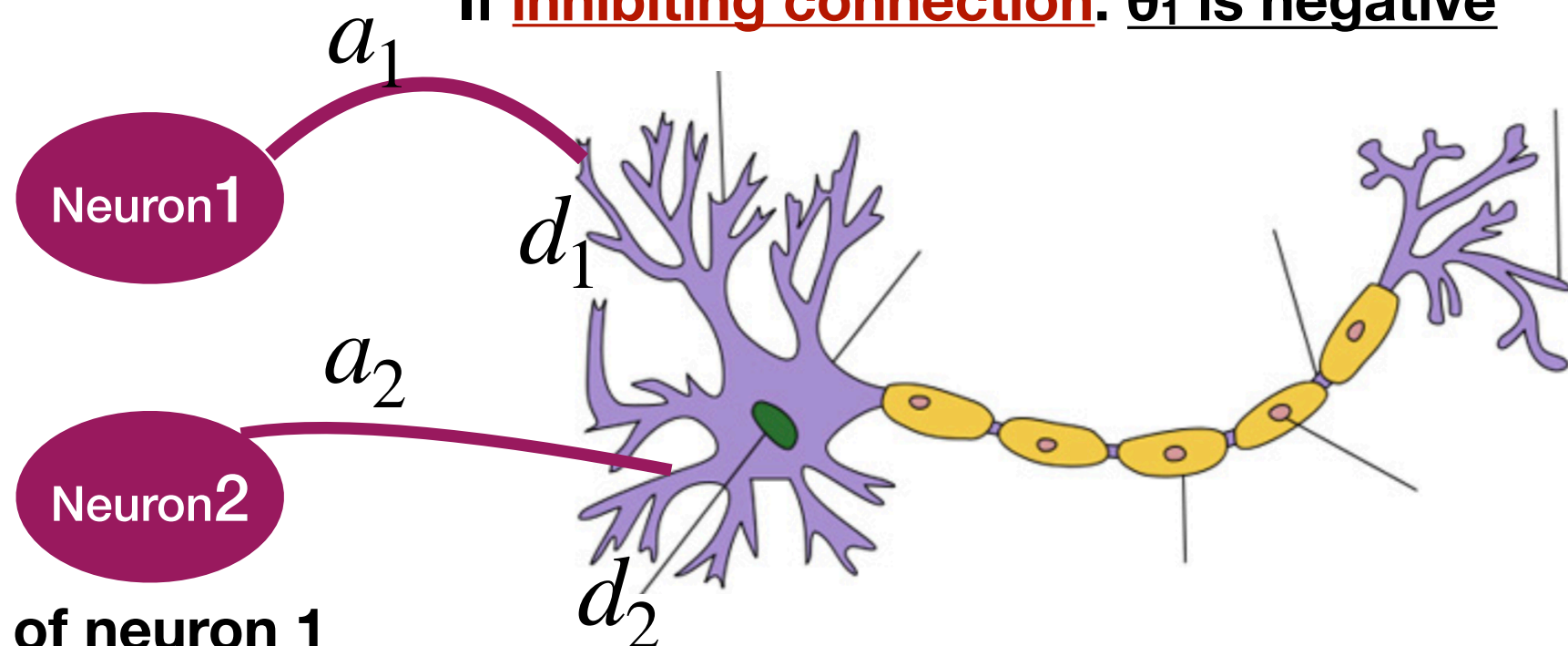
$$d_1 = \theta_1 \times a_1$$

$$d_2 = \theta_2 \times a_2$$

If **strong excitatory connection** between the axone of Neuron1 and the dendrite: θ_1 is large

If **weak excitatory connection** between the axone of Neuron1 and the dendrite: θ_1 is small

If **inhibiting connection**: θ_1 is negative



a_1 is action potential on axone of neuron 1

Equation of a Neuron with 2 dendrites

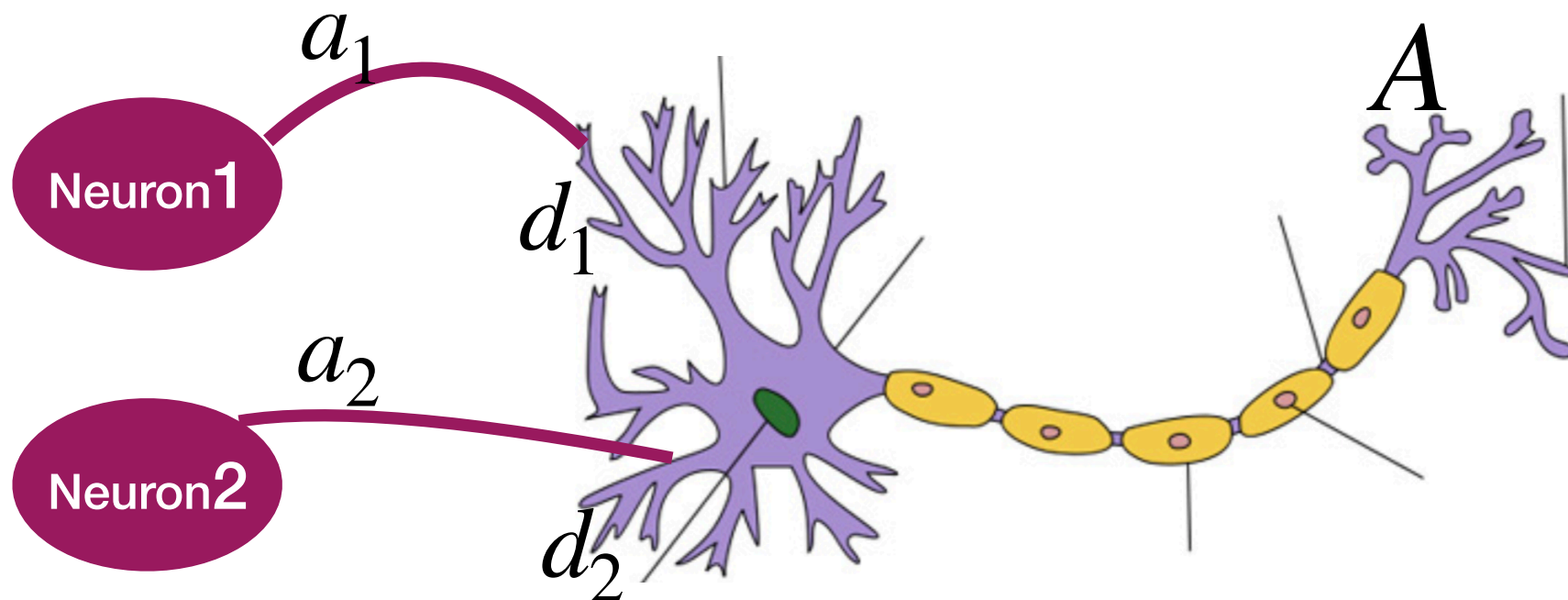
- Now, let us try to describe “mathematically” what this neuron is doing:

$$d_1 = \theta_1 \times a_1 \quad d_2 = \theta_2 \times a_2$$

1. Some input potentials arrives on its **dendrites**
2. The potential are accumulated in the **body** (*graded potential*)

$$\text{graded}P = -70mV + \theta_1 \times a_1 + \theta_2 \times a_2$$

3. If the *graded potential* is larger than a given threshold, the neuron fires an action potential on the **axone**



$$A = -70mV$$

$$\text{If } \text{graded}P \leq -55mV$$

$$A = +30mV$$

$$\text{If } \text{graded}P > -55mV$$

Equation of a Neuron with 2 dendrites

- Let us combine the two equations:

$$\textit{gradedP} = -70mV + \theta_1 \times a_1 + \theta_2 \times a_2$$

$$A = -70mV$$

$$\text{If } -70mV + \theta_1 \times a_1 + \theta_2 \times a_2 \leq -55mV$$

$$A = +30mV$$

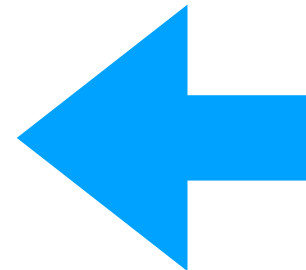
$$\text{If } -70mV + \theta_1 \times a_1 + \theta_2 \times a_2 > -55mV$$

$$A = -70mV$$

$$\text{If } \textit{gradedP} \leq -55mV$$

$$A = +30mV$$

$$\text{If } \textit{gradedP} > -55mV$$



$$A = -70mV$$

$$\text{If } \theta_1 \times a_1 + \theta_2 \times a_2 - 15mV \leq 0$$

$$A = +30mV$$

$$\text{If } \theta_1 \times a_1 + \theta_2 \times a_2 - 15mV > 0$$

Equation of a Neuron with 2 dendrites

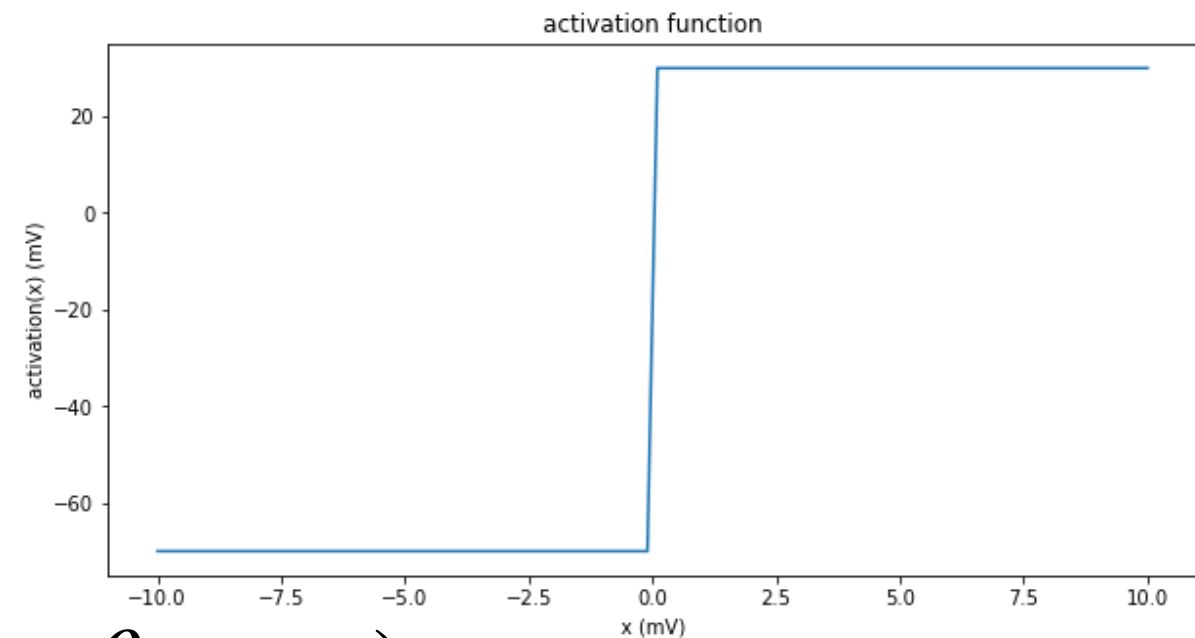
$$A = -70mV$$

$$\text{If } \theta_1 \times a_1 + \theta_2 \times a_2 - 15mV \leq 0$$

$$A = +30mV$$

$$\text{If } \theta_1 \times a_1 + \theta_2 \times a_2 - 15mV > 0$$

- We can define an **activation** function:
 - $\text{activation}(x) = -70mV$ **if** $x \leq 0$ // $30mV$ **if** $x > 0$



Then we can write:

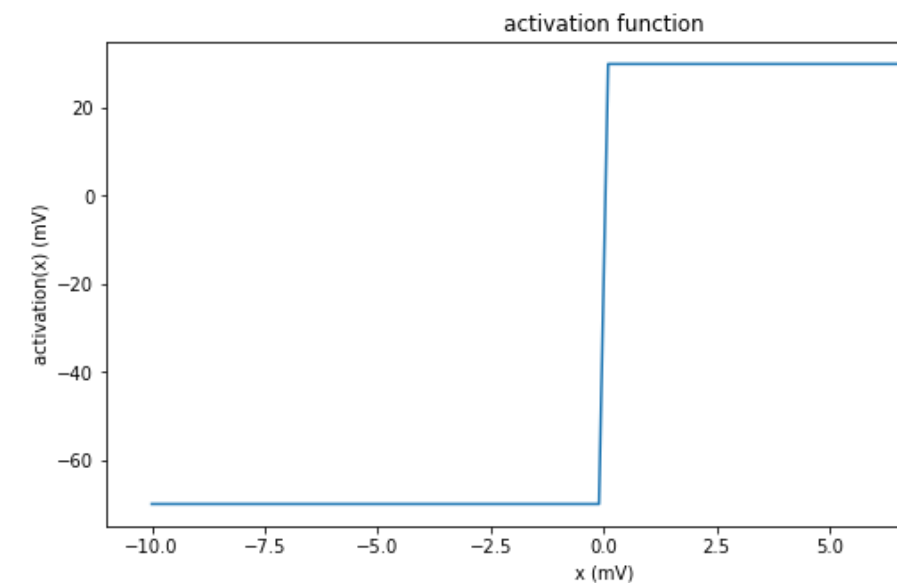
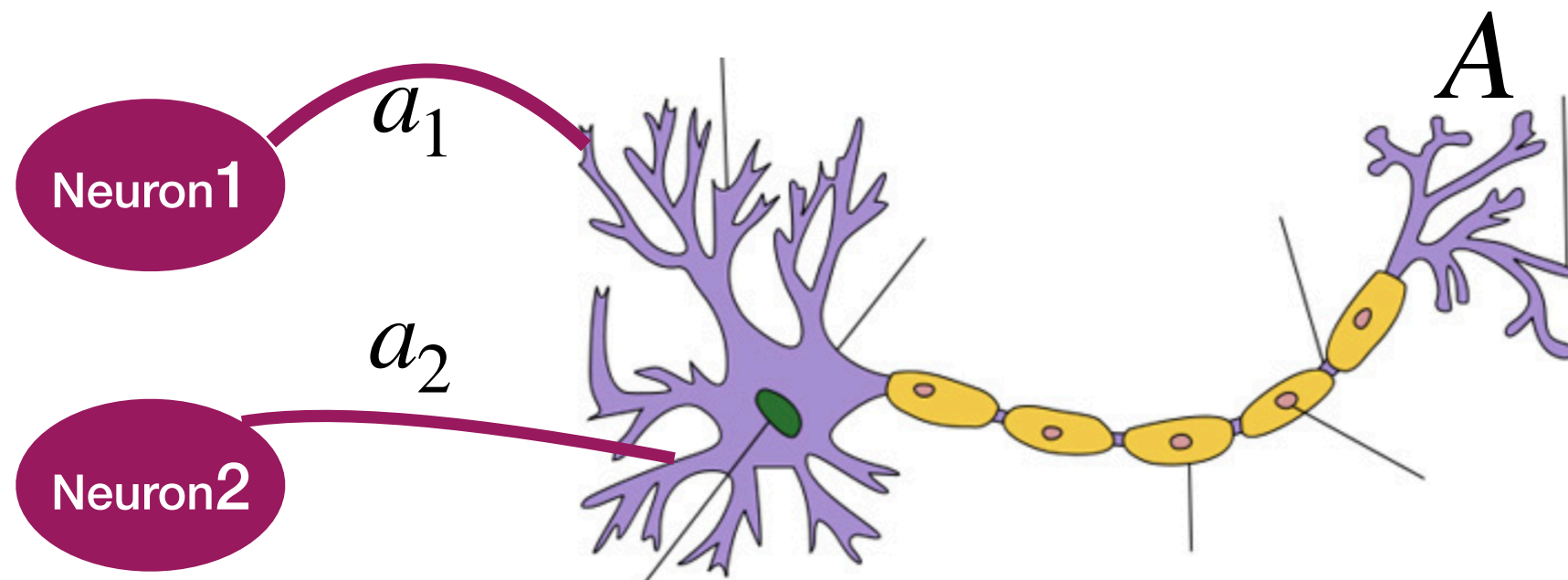
$$A = \text{activation}(-15mV + \theta_1 \times a_1 + \theta_2 \times a_2)$$

Equation of a Neuron with 2 dendrites

- Our final equation:

$$A = activation(\theta_1 \times a_1 + \theta_2 \times a_2)$$

- A : Action Potential of our Neuron
- a_1 : action potential from Neuron1
- θ_1 : represent the strength of synaptic connection between Neuron1 and our neuron
- $activation(x) = -70\text{mV}$ **if** $x < 15\text{mV}$ **else** $+30\text{mV}$



Equation of a Neuron with 2 dendrites

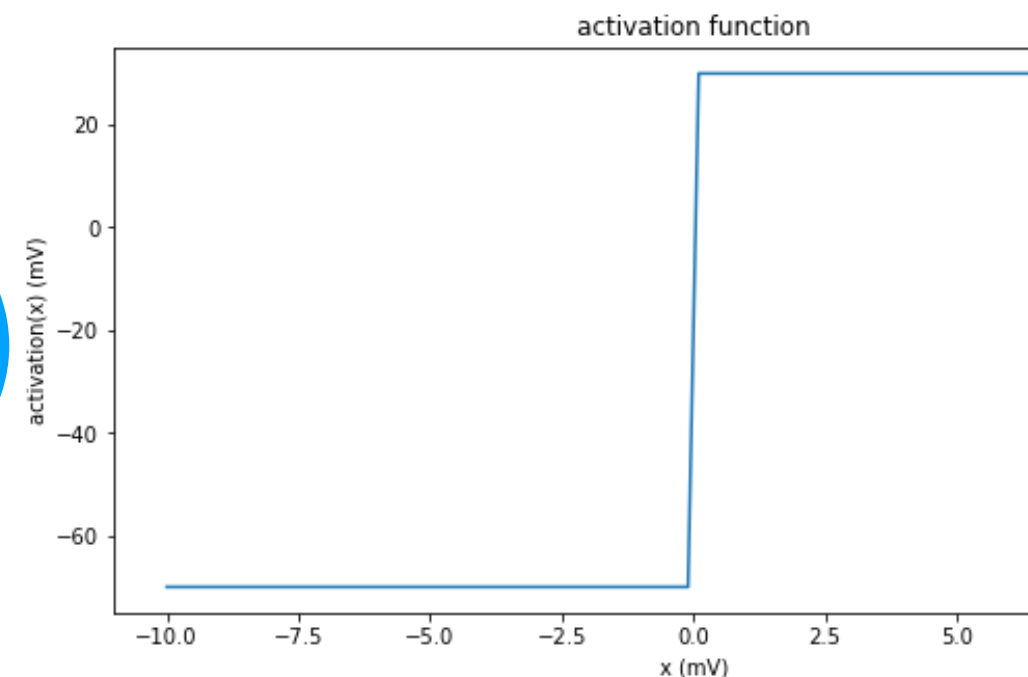
Our final equation:

$$A = \text{activation}(-15\text{mV} + \theta_1 \times a_1 + \theta_2 \times a_2)$$

- A : Action Potential of our Neuron
- a_1 : action potential from Neuron1
- θ_1 : represent the strength of synaptic connection between Neuron1 and our neuron
- $\text{activation}(x) = -70\text{mV}$ if $x < 0$ // $+30\text{mV}$ if $x > 0$



This equation reminds me something



From Biology to Machine Learning

$$A = \text{activation}(-15\text{mV} + \theta_1 \times a_1 + \theta_2 \times a_2)$$

$$\text{activation}(x) = -70\text{mV} \text{ if } x < 0 \text{ // } +30\text{mV} \text{ if } x > 0$$

- 70mV, +30mV and -15mV are values that come from the chemical process in a Neuron
- But from the general point of view of how a neuron is working, their precise value is not important
- What matter is the “**All or None**” behavior: the neurons is activated, or it is not
- Then, let us say our action potential is one, and our resting potential is zero
 - -70mV $\rightarrow$ 0 and +30 mV $\rightarrow$ 1

From Biology to Machine Learning

$$A = \text{activation}(-15\text{mV} + \theta_1 \times a_1 + \theta_2 \times a_2)$$

$$\text{activation}(x) = \mathbf{0} \text{ if } x < 0 \quad // \quad \mathbf{1} \text{ if } x > 0$$

- Then, let us say our action potential is one, and our resting potential is zero
 - $-70\text{mV} \rightarrow 0$ and $+30\text{mV} \rightarrow 1$
- Similarly, let us replace the threshold value -15mV by a parameter θ_0

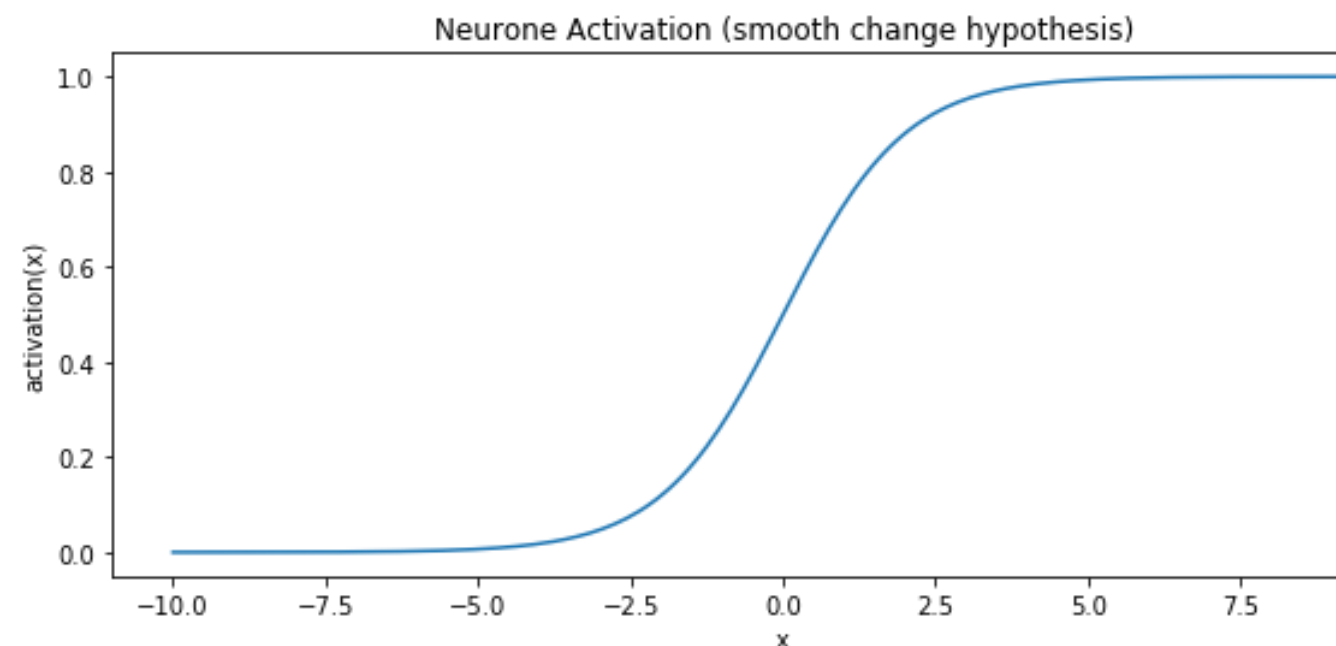
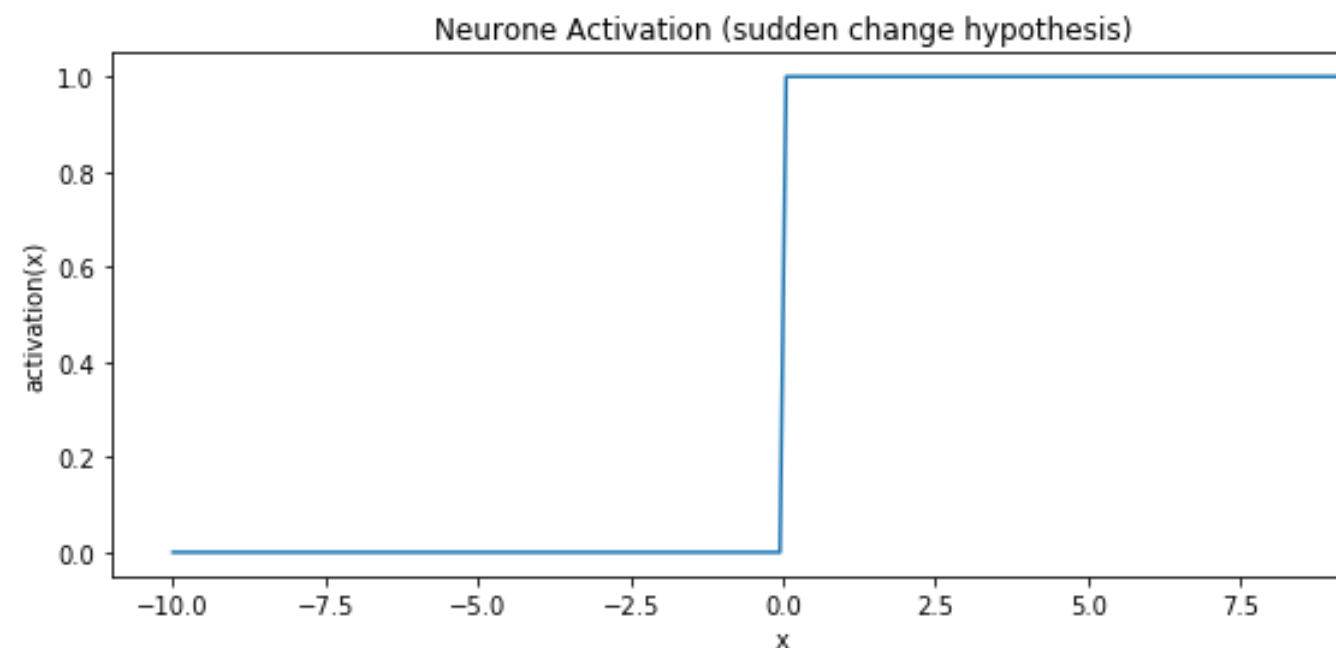
$$A = \text{activation}(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2)$$

$$\text{activation}(x) = \mathbf{0} \text{ if } x < 0 \quad // \quad \mathbf{1} \text{ if } x > 0$$

From Biology to Machine Learning

- Finally, let us consider once more the activation function
- Maybe in practice, the change on the output is not so sudden
- We could consider the activation to be smoother

$$\text{activation}(x) = \mathbf{0} \text{ if } x < 0 \quad // \quad \mathbf{1} \text{ if } x > 0$$

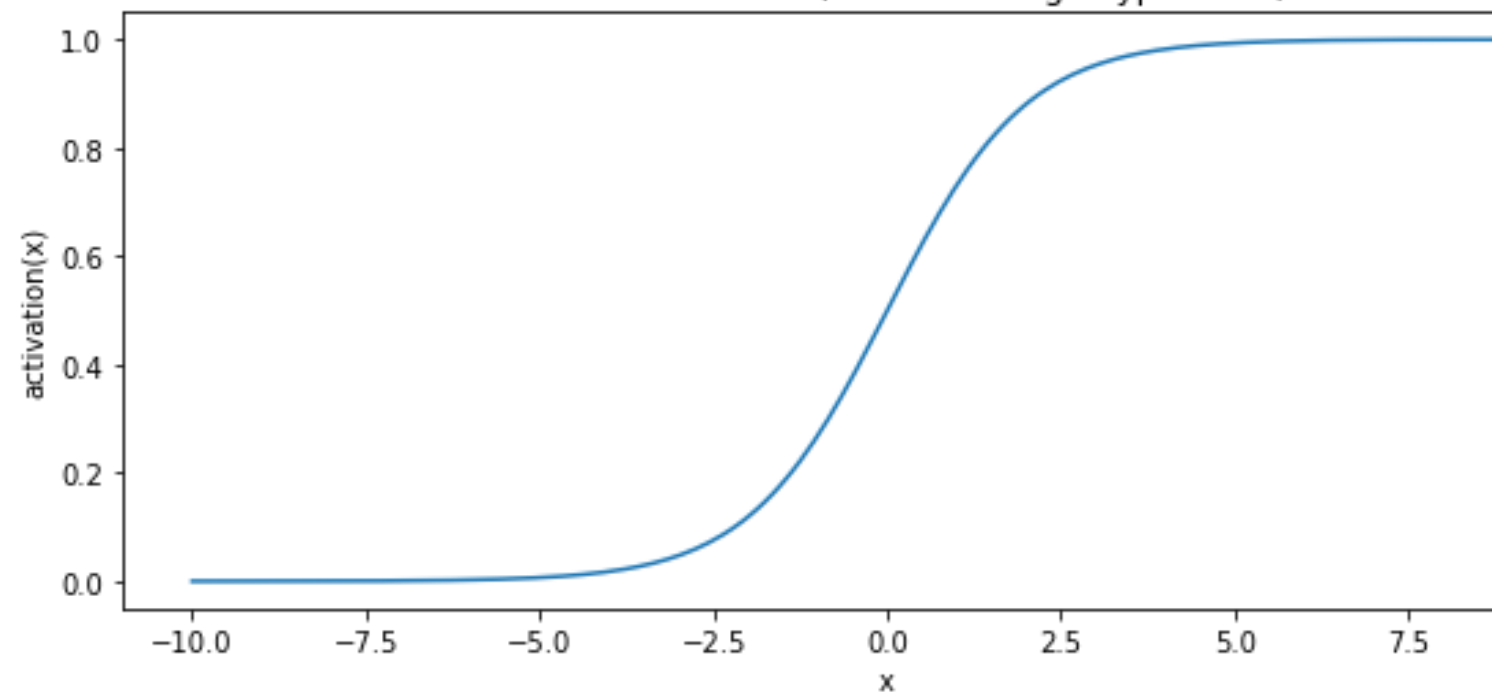


From Biology to Machine Learning

- So, finally, we have this:

$$A = \text{activation}(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2)$$

Neurone Activation (smooth change hypothesis)



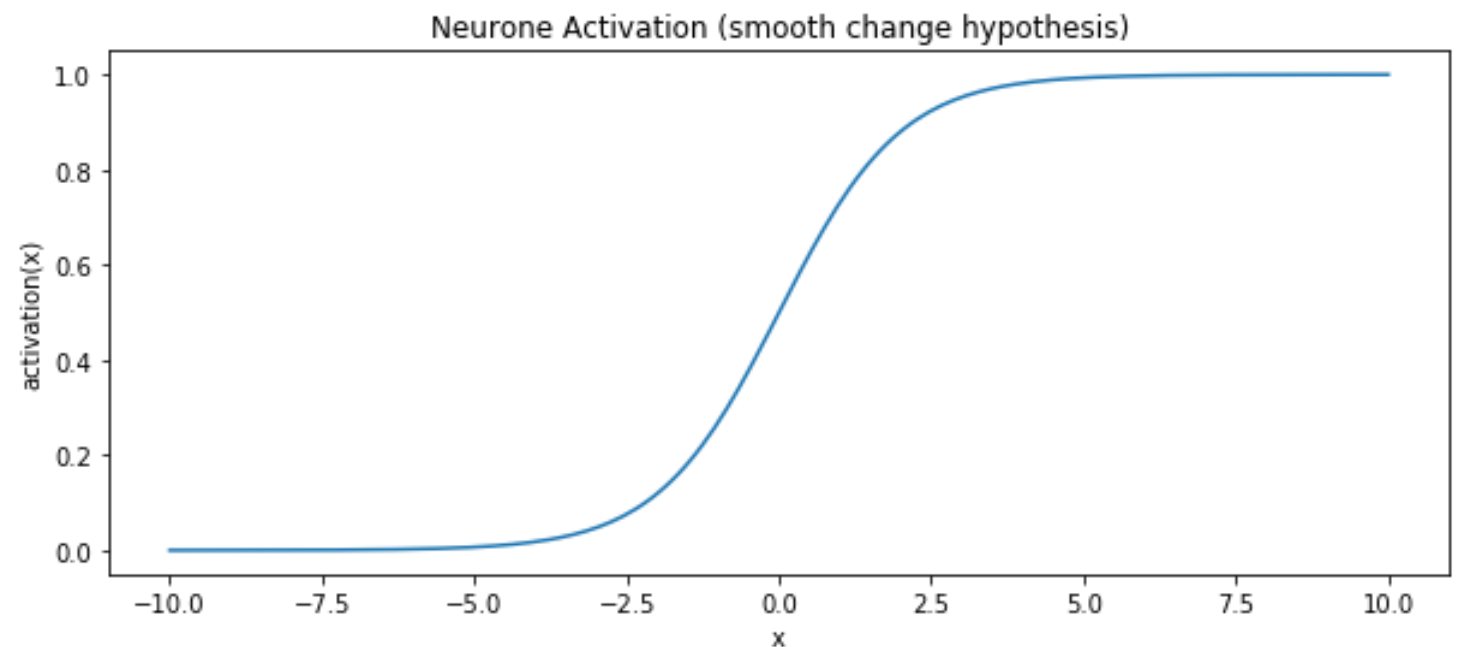
I knew it!
The logistic
classifier!



From Biology to Machine Learning

Neuron Equation: $A = \text{activation}(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2)$

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



Logistic Classifier
Equation:

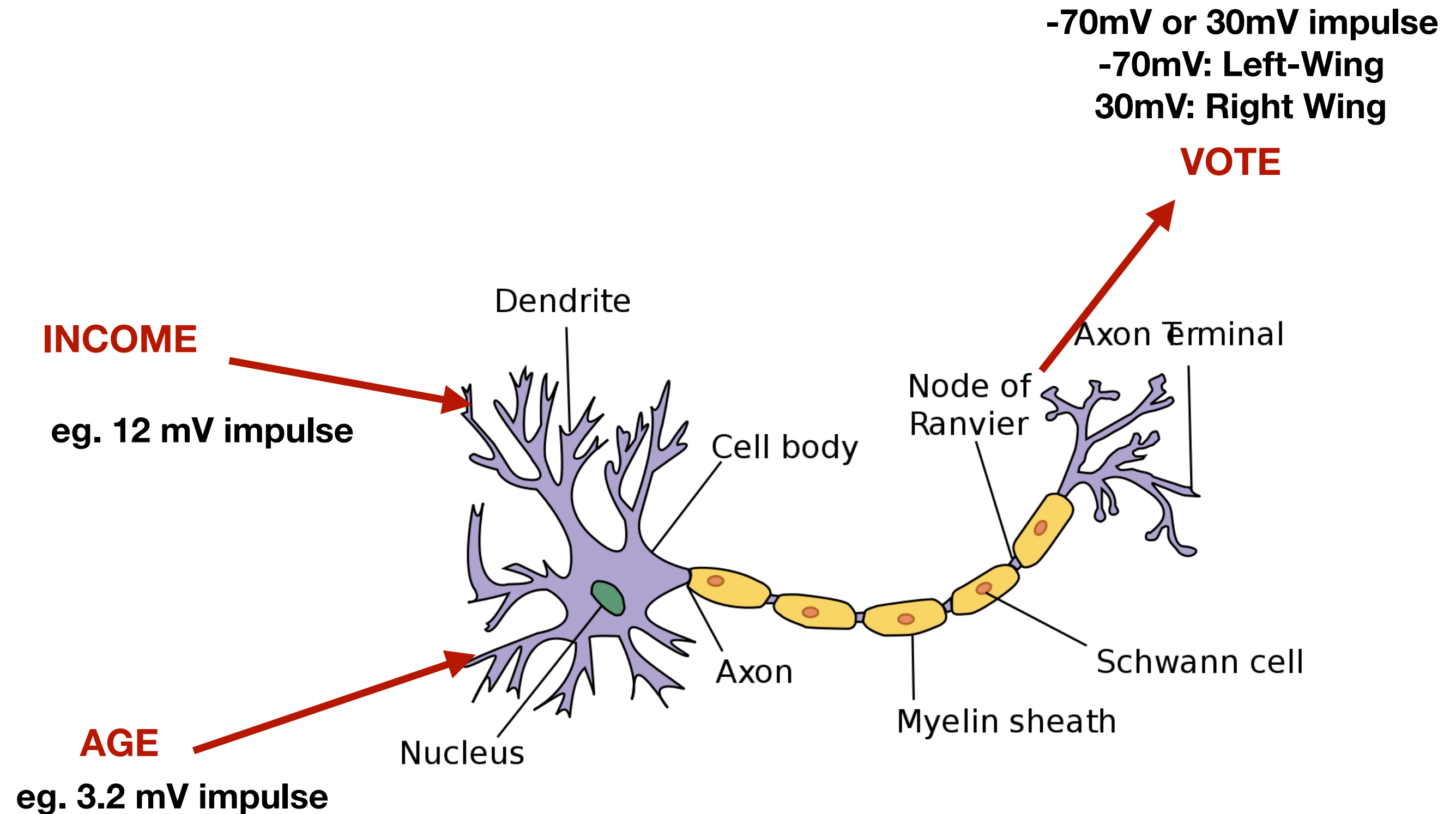
$$\text{score}(\text{income}, \text{age}) = \theta_0 + \theta_1 \times \text{income} + \theta_2 \times \text{age}$$
$$V_{\text{model}} = \sigma(\text{score})$$

From Biology to Machine Learning

$$A = \text{activation}(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2)$$

- Under a few simplifying assumption, the **logistic classifier** is a good approximation of the processing done by a **neuron**
- Therefore, we can also call a logistic classifier an **artificial Neuron**
- Or we can say that a **Neuron** is a **biological logistic classifier**

A neuron as a binary classifier



Remember this?

Example

$$\theta_0 = -8$$

$$\theta_1 = 0.1$$

Example Data

	income	vote
1	40	0
2	70	1
3	20	0

- Suppose these parameters:

$$\text{score}(\text{income}) = \theta_0 + \theta_1 \times \text{income}$$

$$V_{\text{model}} = \sigma(\text{score})$$

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

	income	vote	Score	Prediction V	Predicted Class	Cost
1	40	0	-4	0.018	0	
2	70	1	-1	0.269	0	
3	20	0	-6	0.002	0	

You were actually doing what a neuron do!

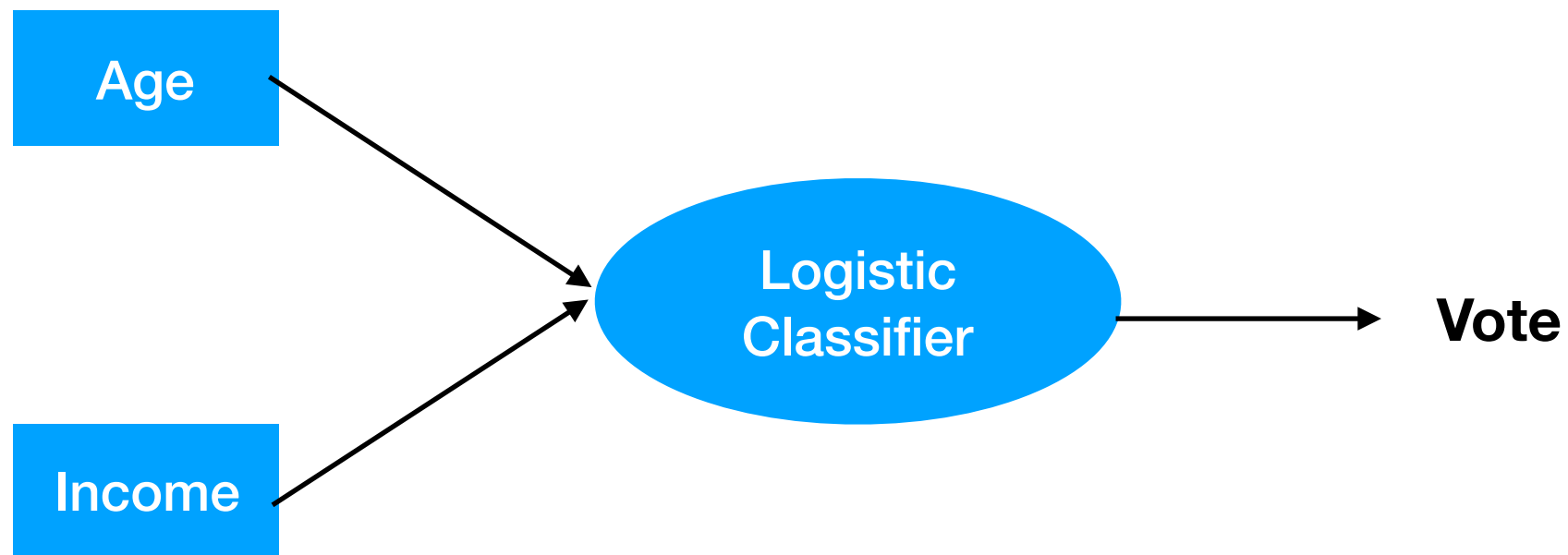
Neural Networks

- Neurons in the brain are very interconnected
- In fact, an average neuron has about 7000 dendrite connections

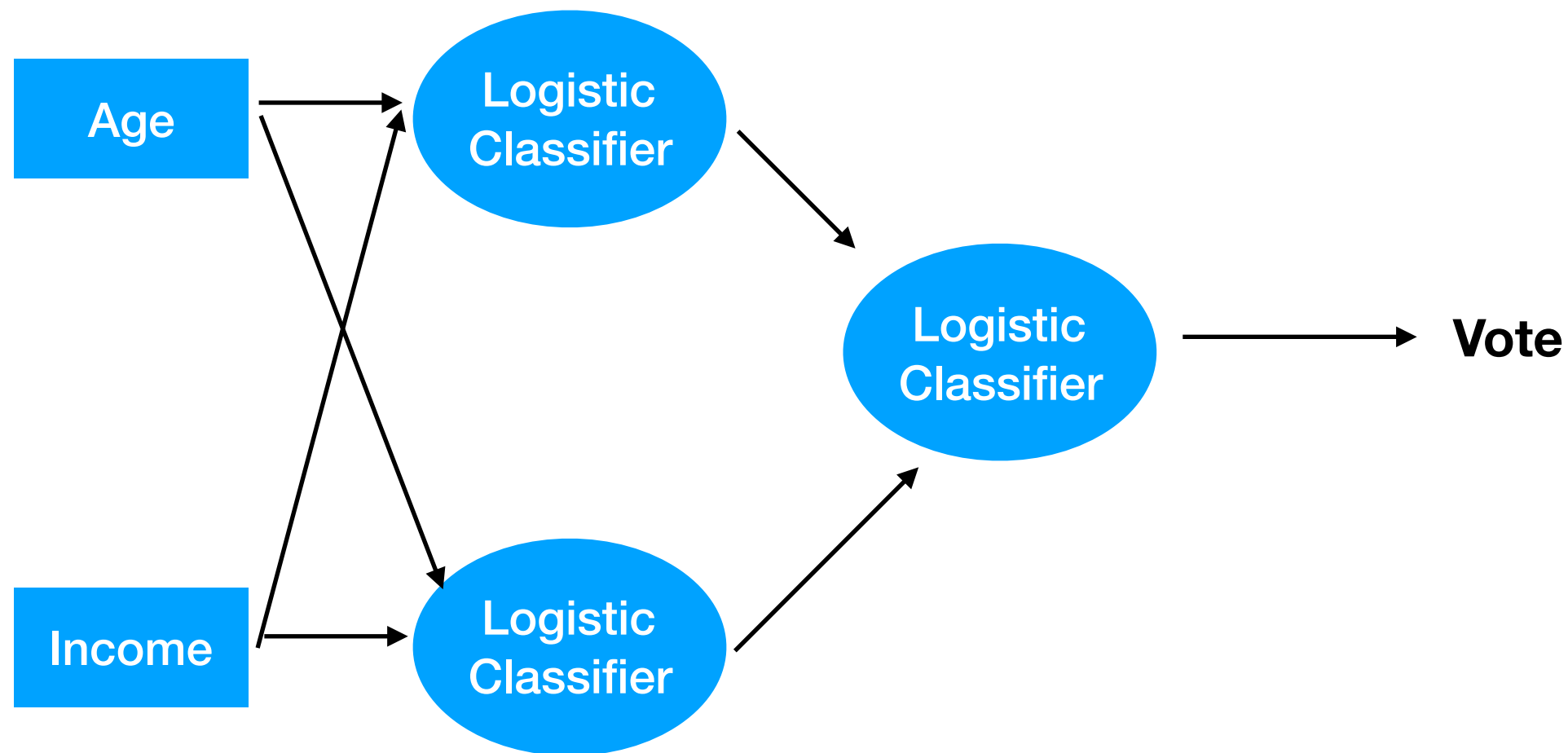
$$A = \textit{activation}(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2 + \dots + \theta_{7000} \times a_{7000})$$

- How about interconnecting logistic classifiers? Will it work?

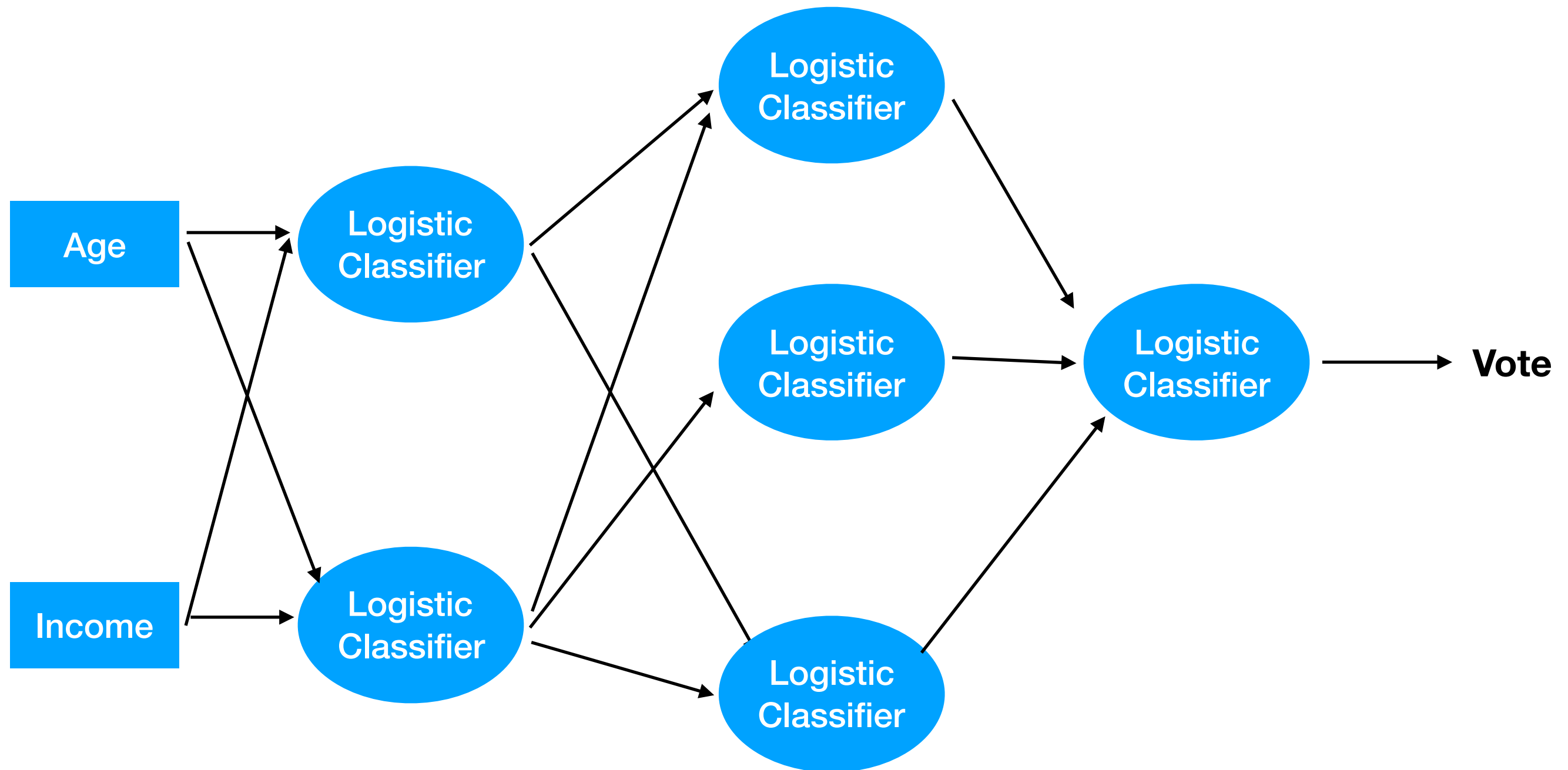
One Neuron



Three Neurons

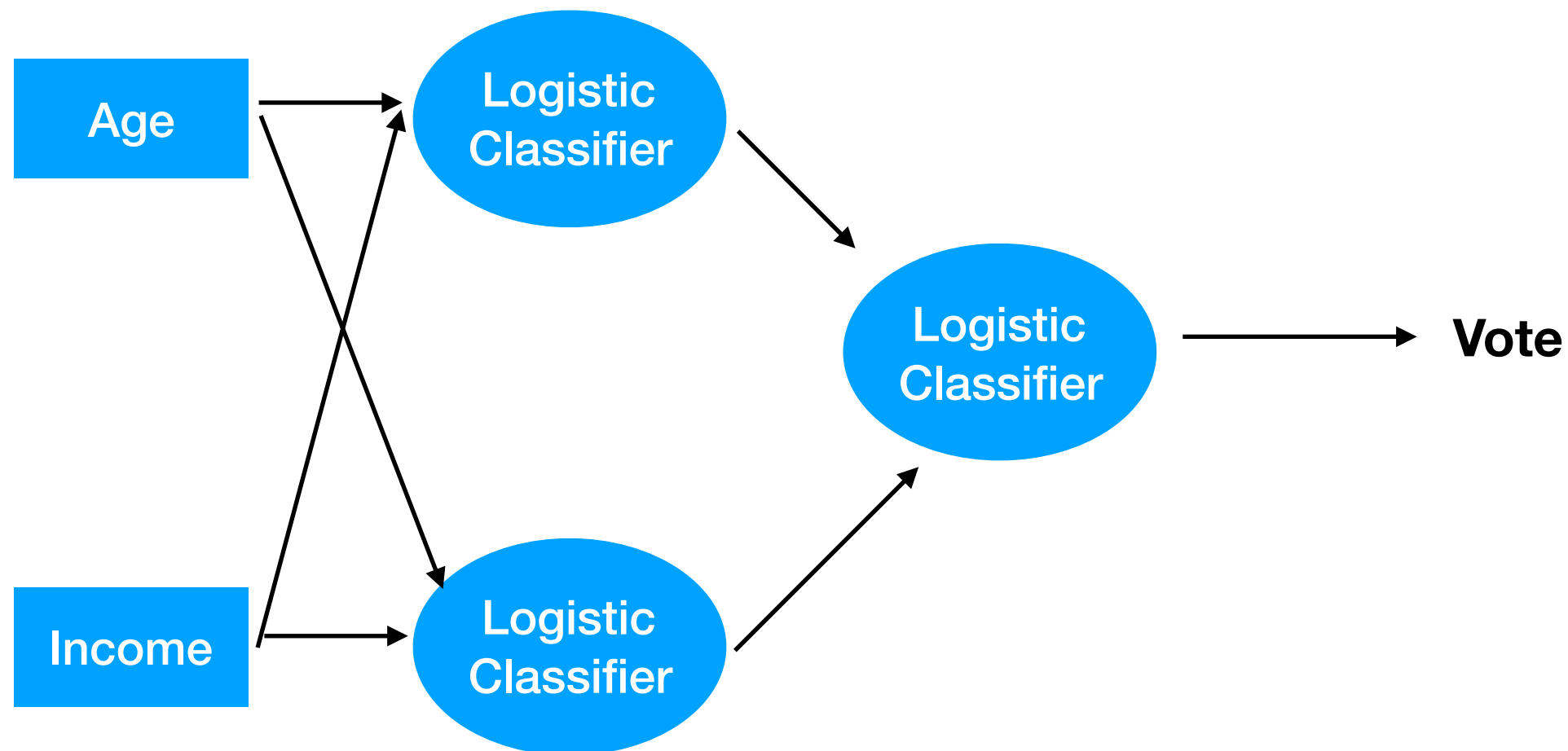


More Neurons



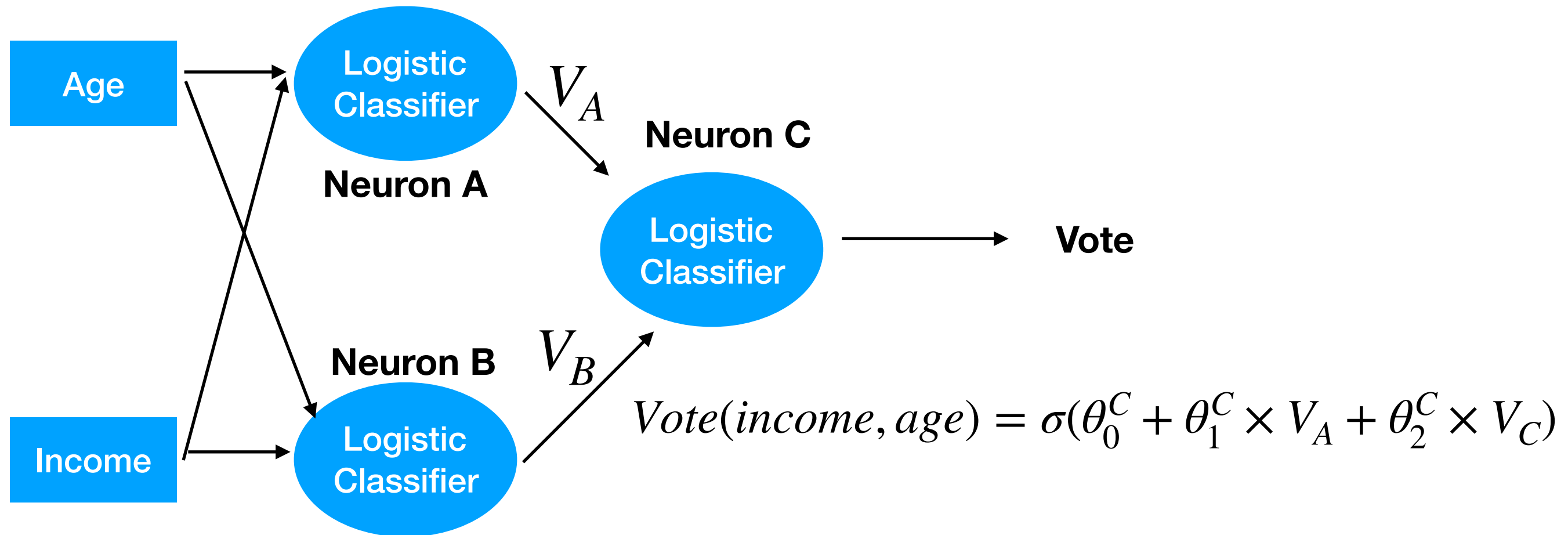
Three Neurons

What do we compute in this case?



Three Neurons

$$V_A(\text{income}, \text{age}) = \sigma(\theta_0^A + \theta_1^A \times \text{income} + \theta_2^A \times \text{age})$$



$$V_B(\text{income}, \text{age}) = \sigma(\theta_0^B + \theta_1^B \times \text{income} + \theta_2^B \times \text{age})$$

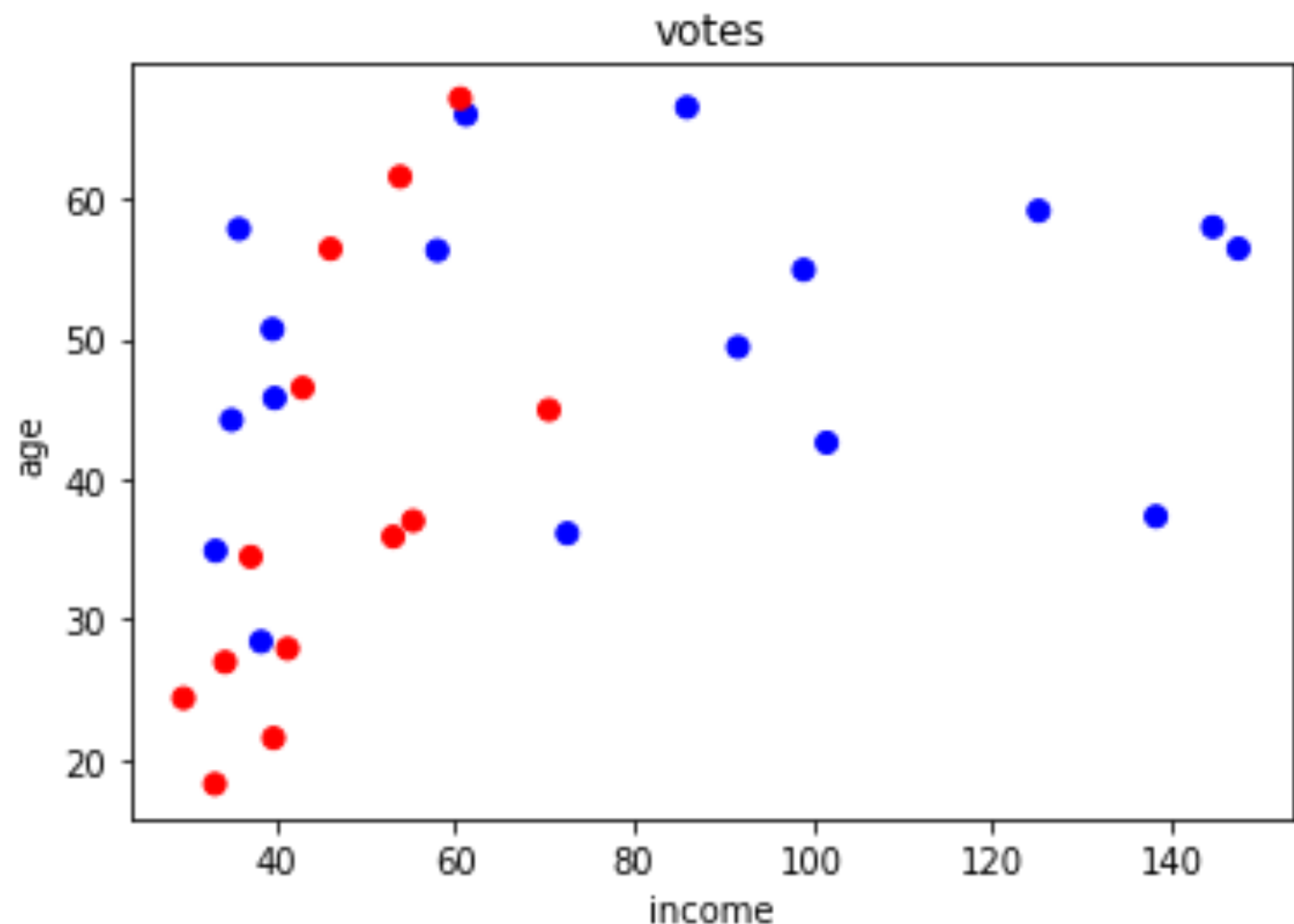
Does it work?

- Can we really get something better by connecting artificial neurons like this?
- Yes!
- The “power” of a network increase with the number of neurons
- If we connect many simple logistic classifiers, we get a more powerful binary classifier

Does it work?

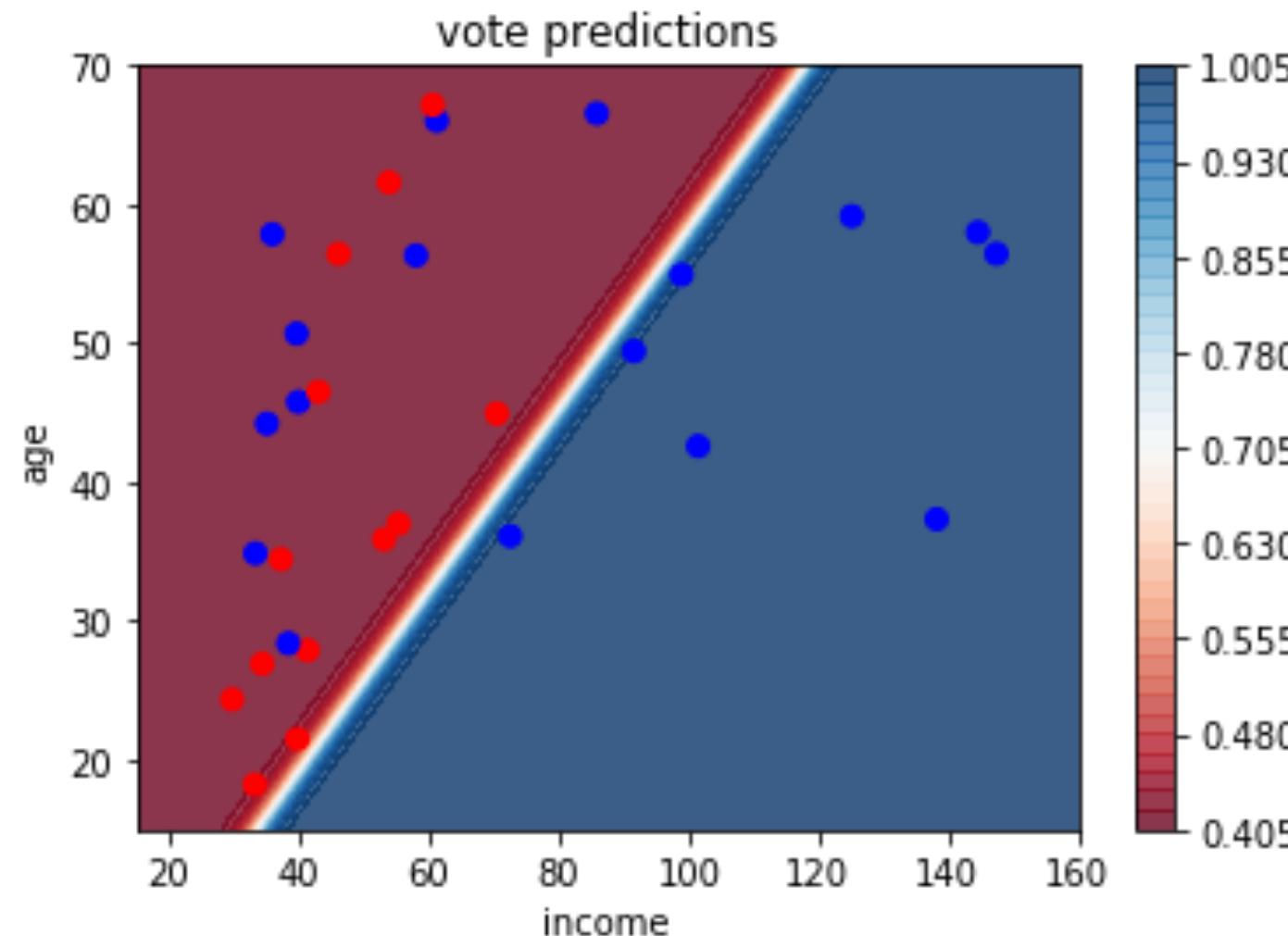
- Let us look again at our voting data
- Let us try to make different neural networks learn to predict vote knowing the income and age of somebody

	income	age	vote
0	39.0	42.0	L
1	30.0	21.0	L
2	47.0	65.0	L
3	69.0	50.0	R
4	52.0	53.0	R
5	110.0	28.0	R
...	...	...	...



Single Neuron

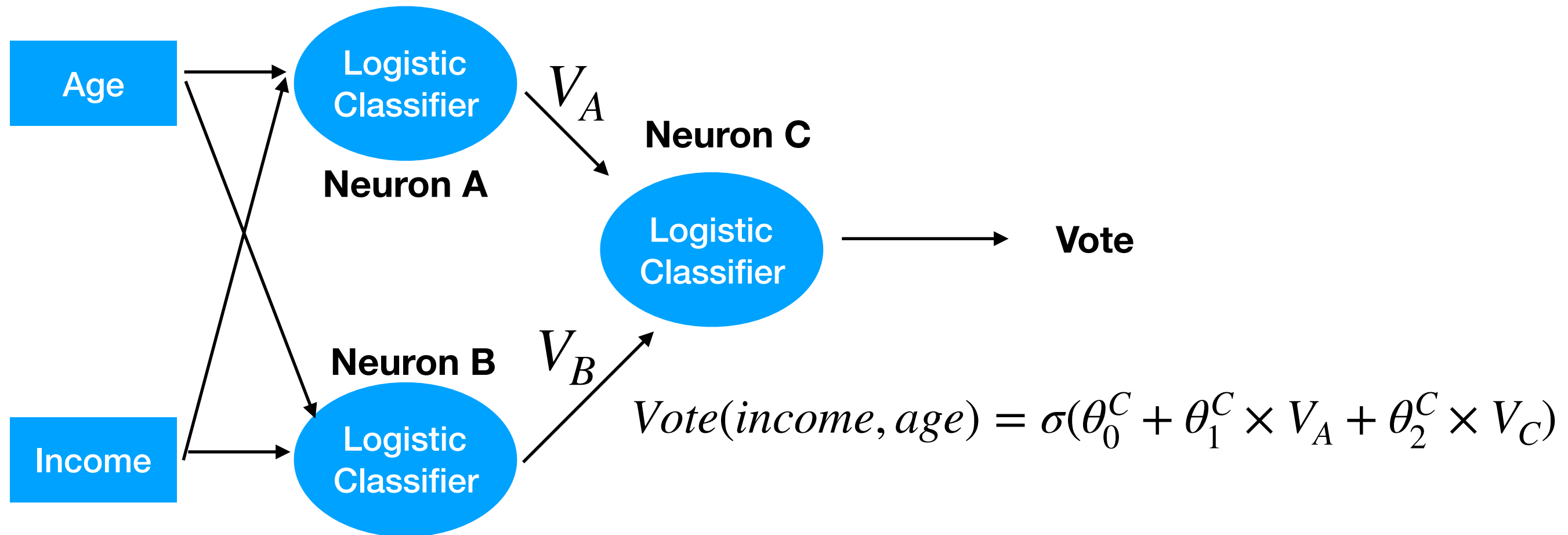
- A **Single Neuron** is just a simple **logistic classifier**
 - Like we saw last week
- Class boundary is a straight line
- Only 2 zones



- **Darker red** area means the classifier is **certain** people in this age/income zone will vote for the **Left-Wing party**
- **Darker blue area** means the classifier is **certain** people in this age/income zone will vote for the **Right-Wing party**
- **White-ish** area means the classifier is **uncertain**

Three Neurons

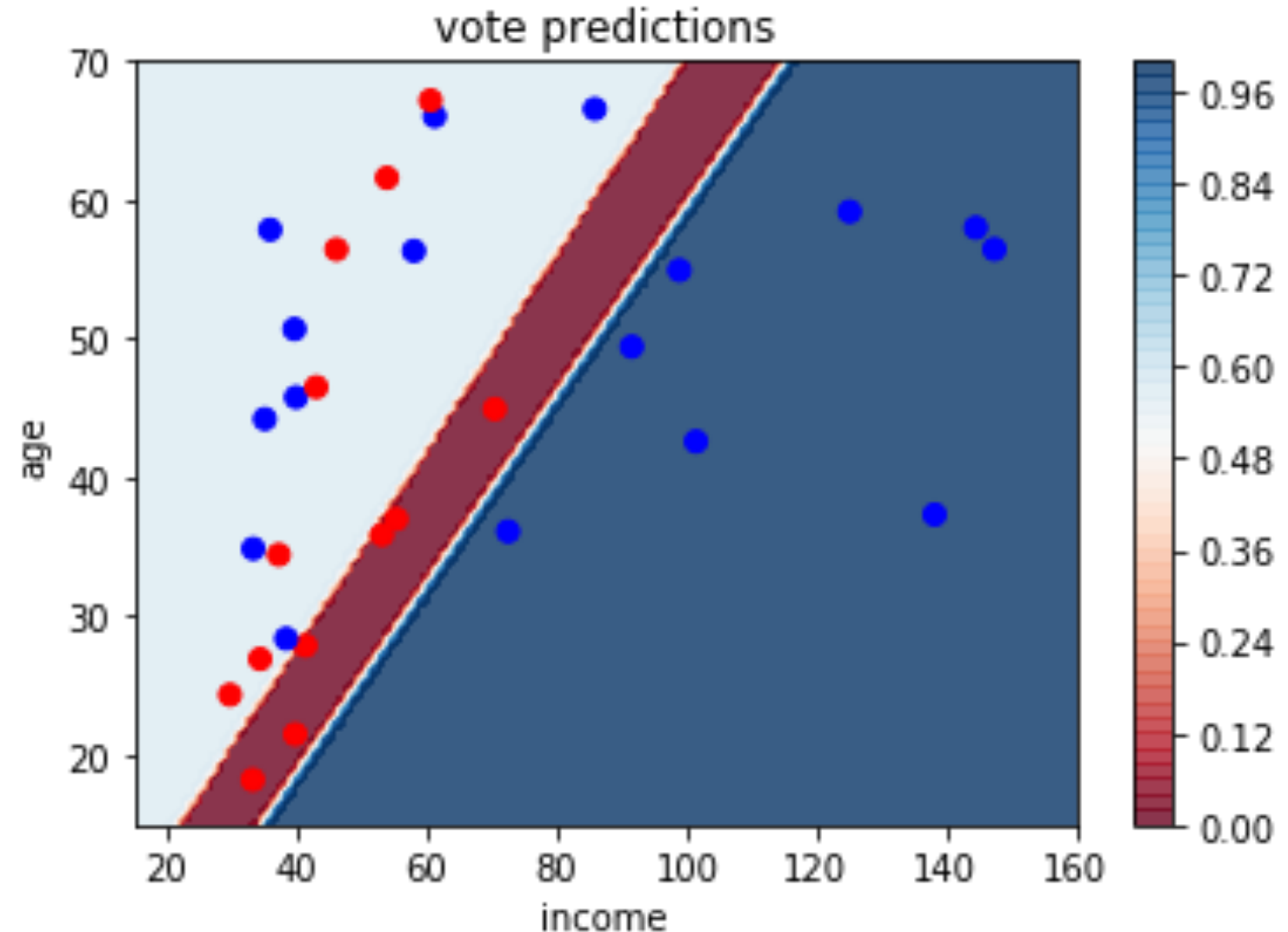
$$V_A(\text{income}, \text{age}) = \sigma(\theta_0^A + \theta_1^A \times \text{income} + \theta_2^A \times \text{age})$$



$$V_B(\text{income}, \text{age}) = \sigma(\theta_0^B + \theta_1^B \times \text{income} + \theta_2^B \times \text{age})$$

Three Neurons

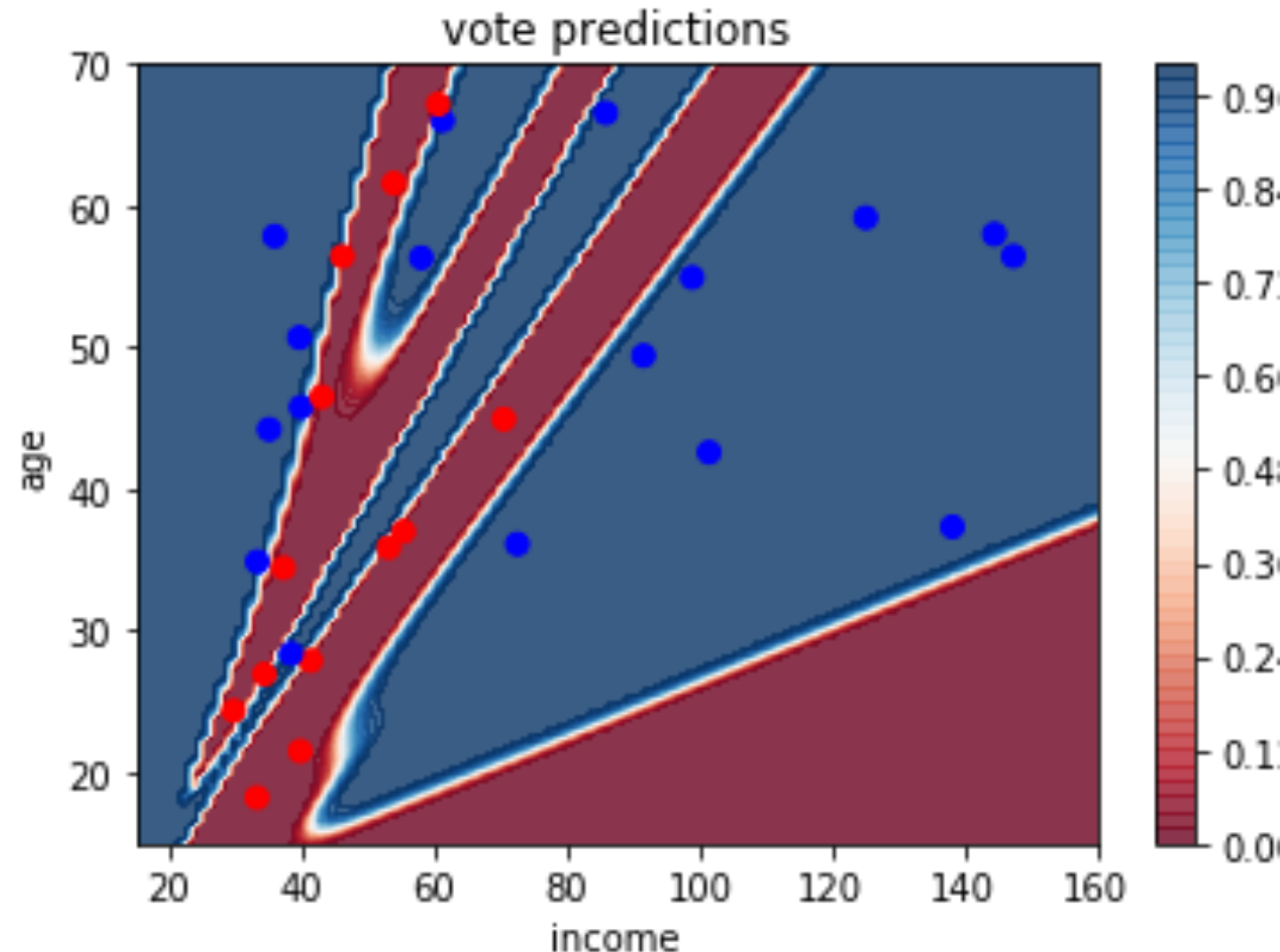
- With three neurons: the classifier can define a third zone of uncertainty



- **Darker red** area means the classifier is **certain** people in this age/income zone will vote for the **Left-Wing party**
- **Darker blue area** means the classifier is **certain** people in this age/income zone will vote for the **Right-Wing party**
- **White-ish** area means the classifier is **uncertain**

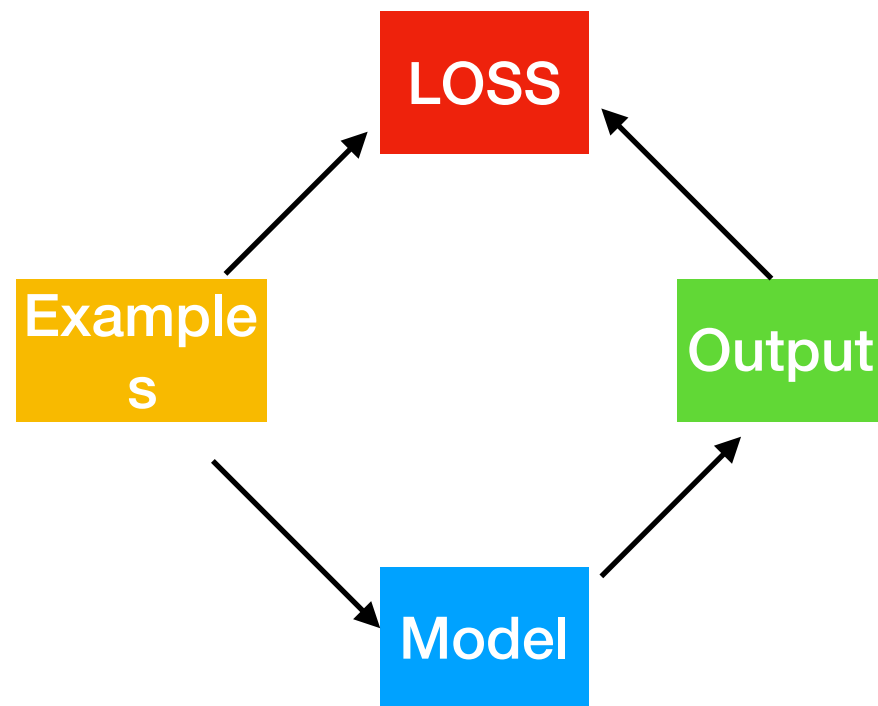
~100 Neurons

- The Neural Network can now perfectly predict each example
- There seems to be some overfitting



- **Darker red** area means the classifier is **certain** people in this age/income zone will vote for the **Left-Wing party**
- **Darker blue area** means the classifier is **certain** people in this age/income zone will vote for the **Right-Wing party**
- **White-ish** area means the classifier is **uncertain**

How to find the parameters θ of the Neural Network?

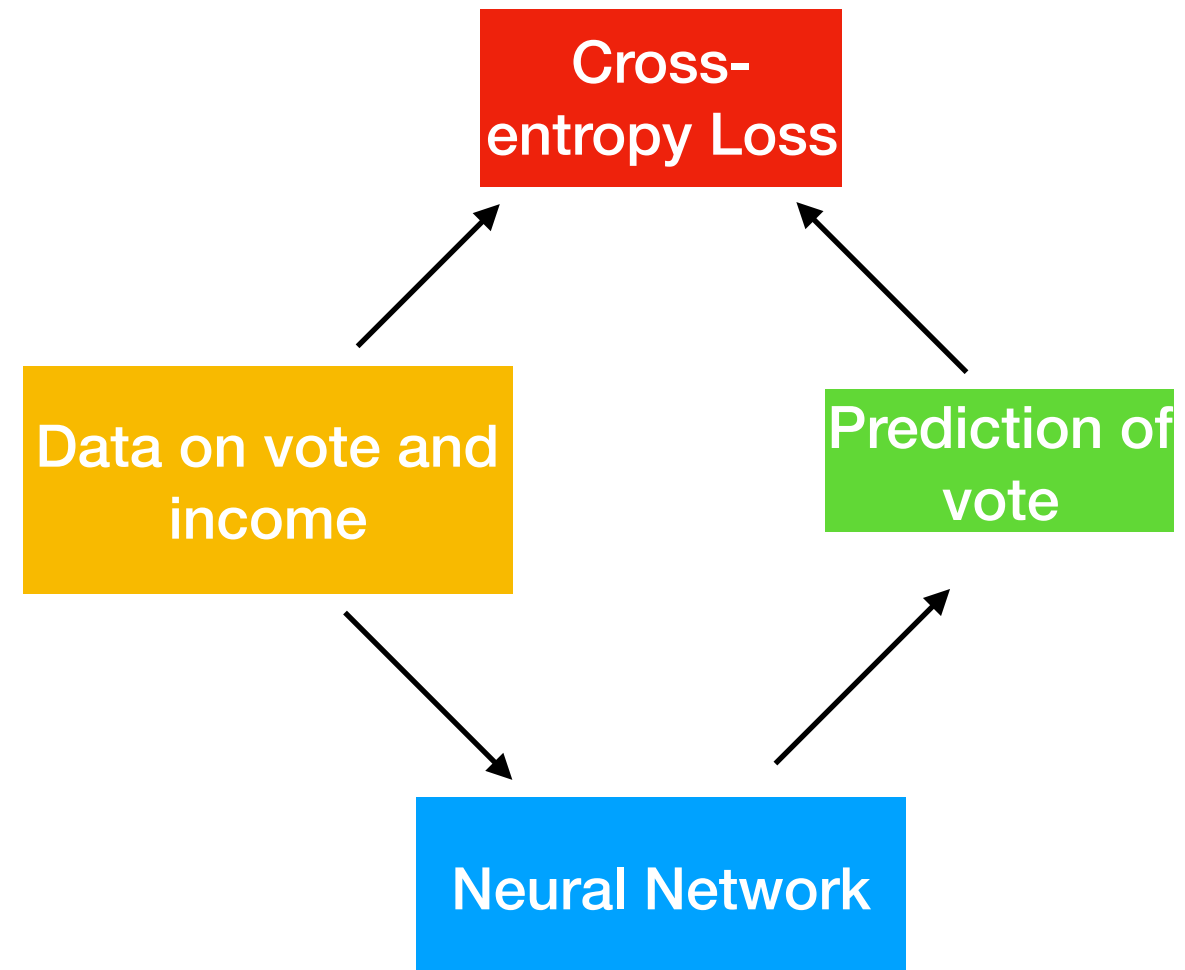


We have some examples

We have a model with some parameters

We have a loss function that compute the difference between the example and the prediction of our model

We minimize the loss to obtain the best parameters for our model



Making the connections

- What we have seen so far:

Linear regression

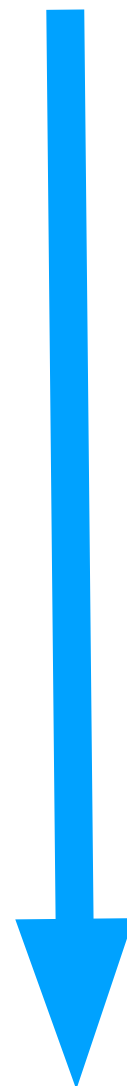
$$f(x, y) = \theta_0 + \theta_1 \times x + \theta_2 \times y$$

Logistic Classifier

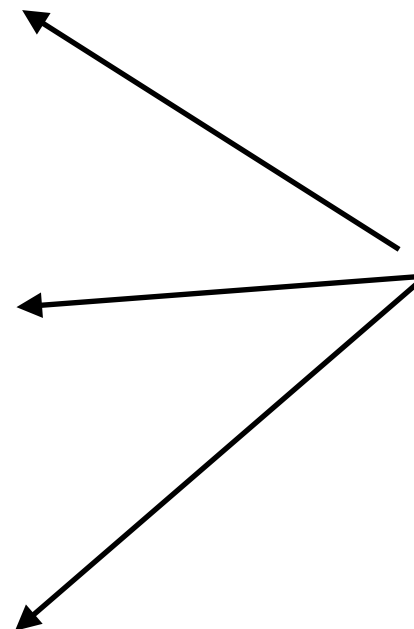
$$\text{score}(x, y) = \theta_0 + \theta_1 \times x + \theta_2 \times y$$
$$\text{prediction} = \sigma(\text{score})$$

**Neural Network: Combination of
logistic classifier**

Increasing
complexity



**All trained by gradient
descent on the loss**

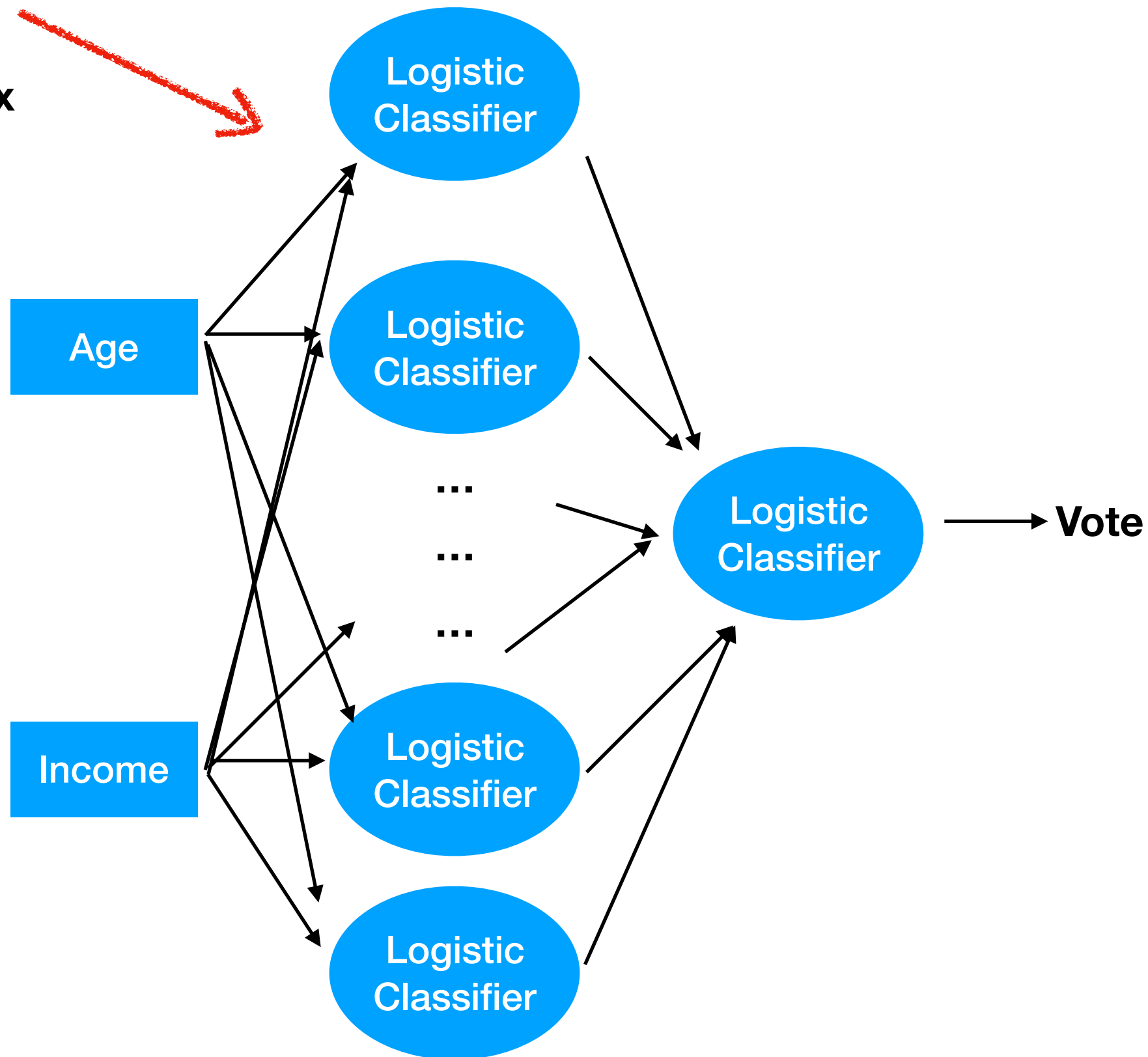


Does it work? A theoretical answer

- Indeed, we do get better classifiers by connecting simple logistic classifiers
- Universal approximation theorem:
 - **Any** function can be **approximated** by a neural network with **2 layers of neurons**

Universal Approximation theorem

Stack enough neurons here,
and you can learn from any
data, no matter how complex
it is



But note that this 2-
layers architecture is
usually not the most
efficient

Funny calculations: what is the power of the brain?

- We mentioned before that on average, a **human neuron** has 7000 dendrite connections:

$$A = \text{activation}(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2 + \dots + \theta_{7000} \times a_{7000})$$

- Therefore, to “simulate” one neuron computation, we need about 7000 multiplications and 7000 additions
- It seems a neuron cannot activate more than 200 times per seconds
- To simulate a neuron in real time, we therefore need to compute about $(7000 + 7000) \times 200 = 2\,800\,000$ operations per second

Funny calculations: what is the power of the brain?

- To simulate a neuron in real time, we therefore need to compute about $(7000 + 7000) \times 200 = 2\,800\,000$ operations per second
- In computer technology, we use the term FLOPS (floating point operation per seconds)
- A current computer with an intel processor should have a power of about 200 Gflops
- Therefore, it can simulate, in real time, about $200 \times 10^9 / 2\,800\,000 = 71\,000$ neurons
- How many neurons in the brain?

Funny calculations: what is the power of the brain?

- A intel CPU can simulate, in real time, about 71 000 neurons
- How many neurons in the brain?
 - About 100 billions !
 - (actually more like 80 billions)

Funny calculations: what is the power of the brain?

- What is the number of FLOPS for the brain?
 - $1e11 \times 2\,800\,000 = 2.8e17$ FLOPS
 - $= 280\,000$ TeraFLOPS $= 280$ PetaFLOPS

Most Powerful computer in the world?

- As of 2018, the most powerful (known) computer in the world is the IBM Summit, located in Oak Ridge Laboratory
 - Made of almost 40 000 CPUs/GPUs
 - Computation power: about 150 Petaflops
- -> Still not enough to simulate one human brain!
- However, we are getting close...



Funny calculations: what is the power of the brain?

- Previous calculations should not be considered **too** seriously
 - We did a lot of biological and mathematical approximations about how the brain work
- Still, it gives us some idea about how powerful the human brain is and why AI is difficult:
 - Even a supercomputer filling several room cannot match the computation power of a human brain that is 1000s of times smaller
- At the same time, computer are **beginning** to be competitive with the brain
 - Might be an explanation of why AI is now starting to become more successful

On Biology and Machine Learning

- Today, we spent a good time discussing biological neurons and the brain
- But do not think Neural Networks and AI are about simulating a human brain!
 - In practice, we do not mind doing things that would not happen with a biological neurons if it suits us
- In practice, we use Artificial Neural Networks because they are efficient for the tasks we want to do
 - And they also have some mathematic justification
- You could say it is “almost” a coincidence that the tools we use are similar to the way the brain work
- Also note that gradient descent is the way Artificial Neural Networks learn, but not the way the brain learn

Next Time

- Start discussing Neural Network architectures
 - Feed-Forward Neural Networks
 - Convolutional Neural Networks
 - Recurrent Neural Networks
- Mathematical aspect: BackPropagation, Matrix multiplication