Introduction to Neural Networks

Fundamentals of Artificial Intelligence
Fabien Cromieres
Kyoto University

Introduction to Neural Networks

- Finally, the fancy part!

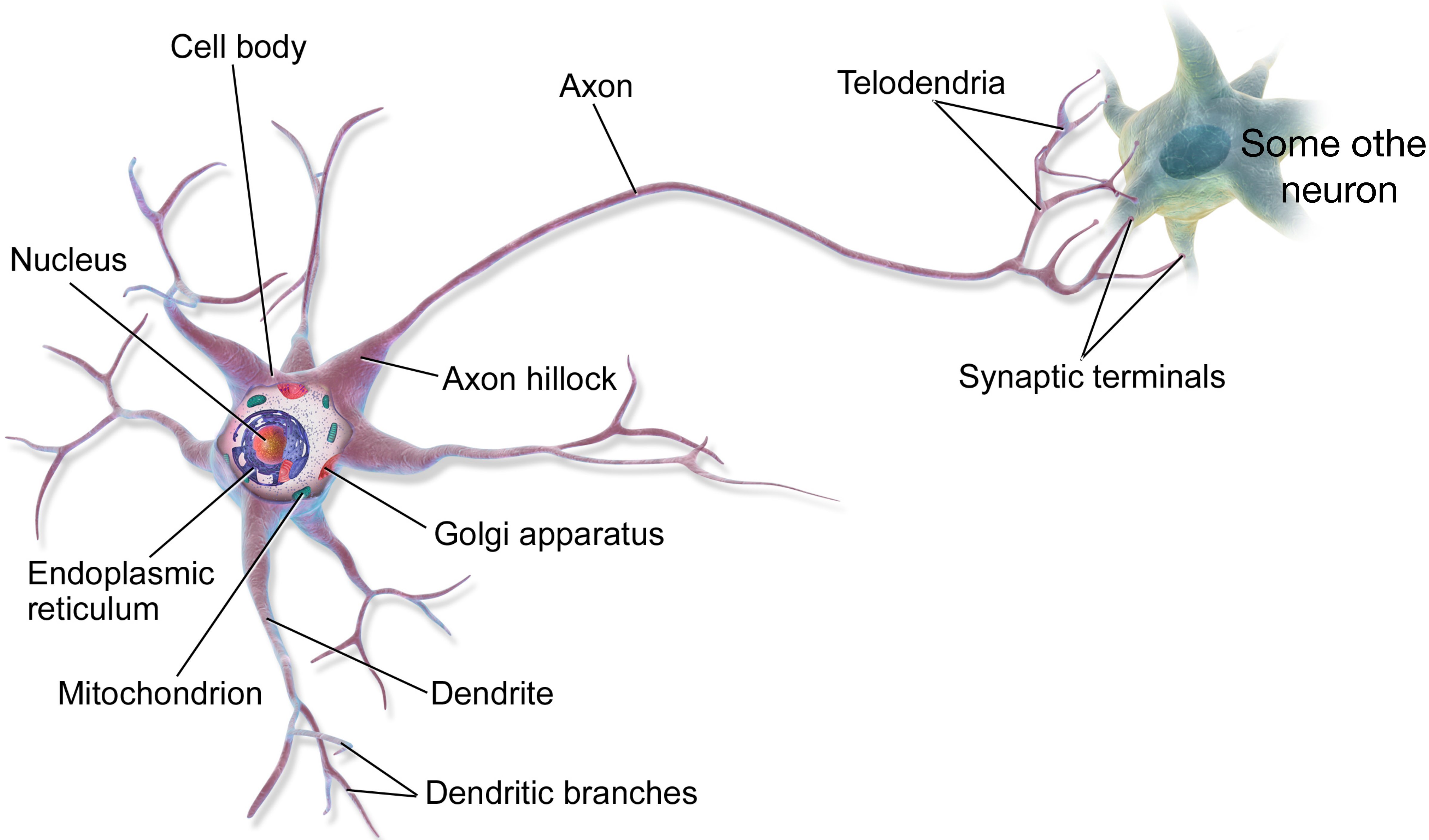
- **Neural Networks** are currently the most efficient and most used models for Machine Learning (and for AI in general, actually)

- Today, we discuss neurons, both the **biological neurons** and the **artificial neurons**

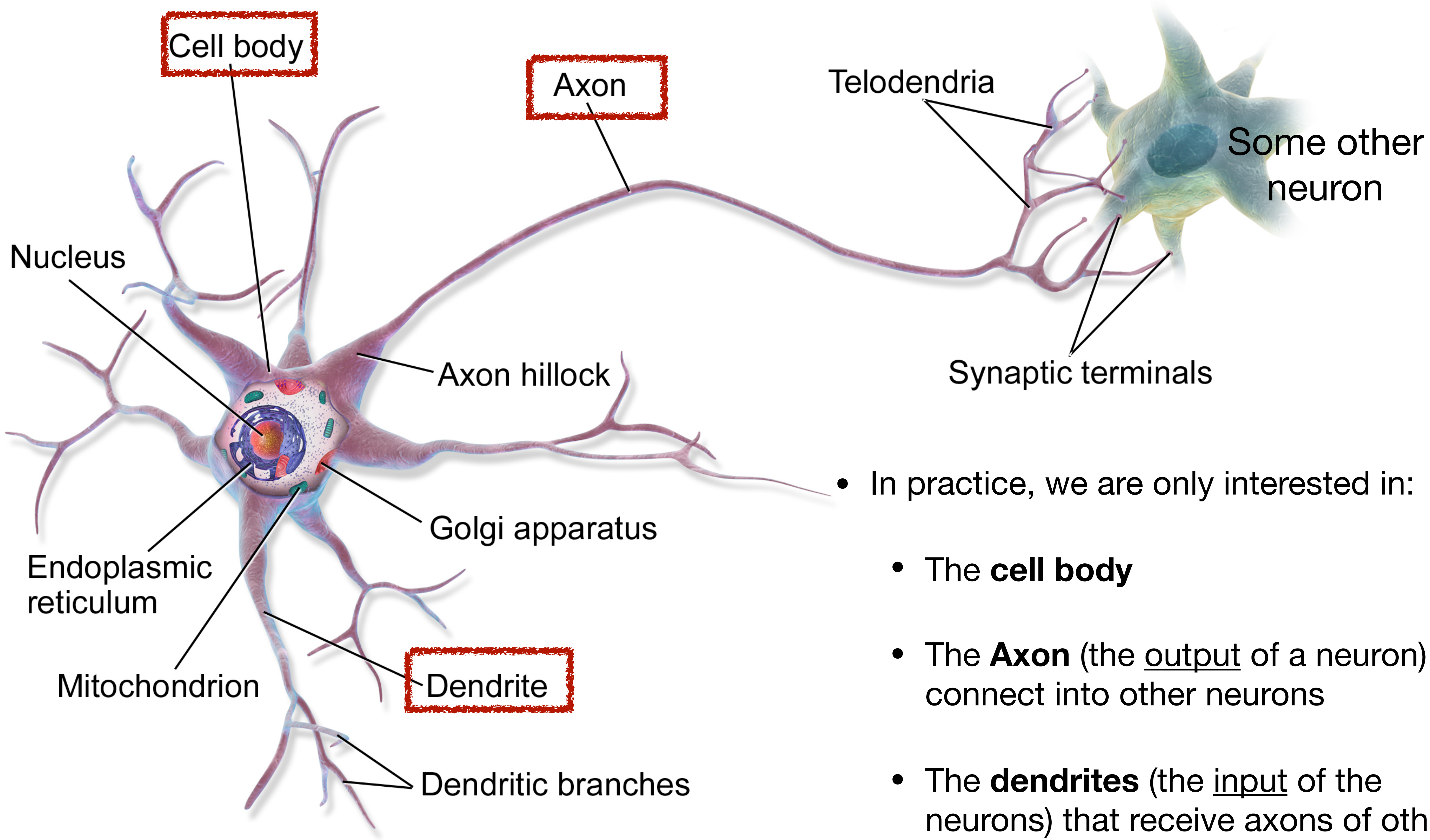
The real (biological) Neurons

- 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



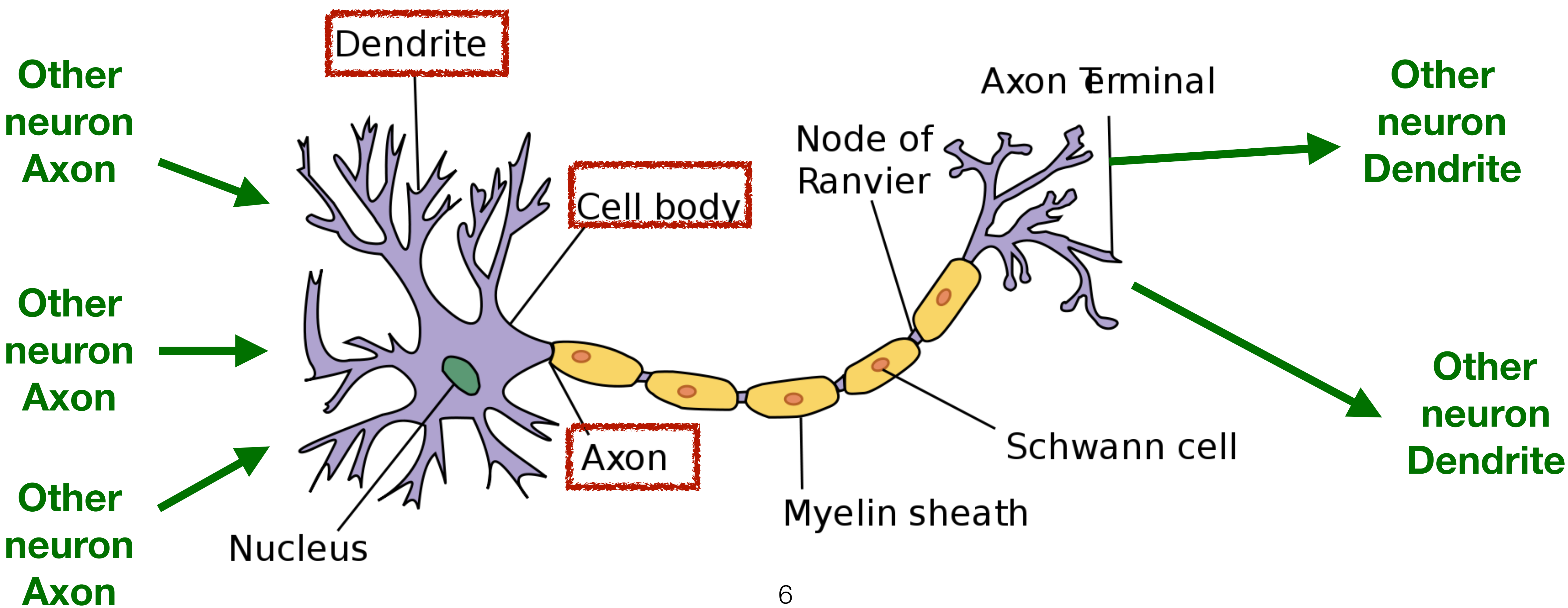
Anatomy of a neuron



- 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

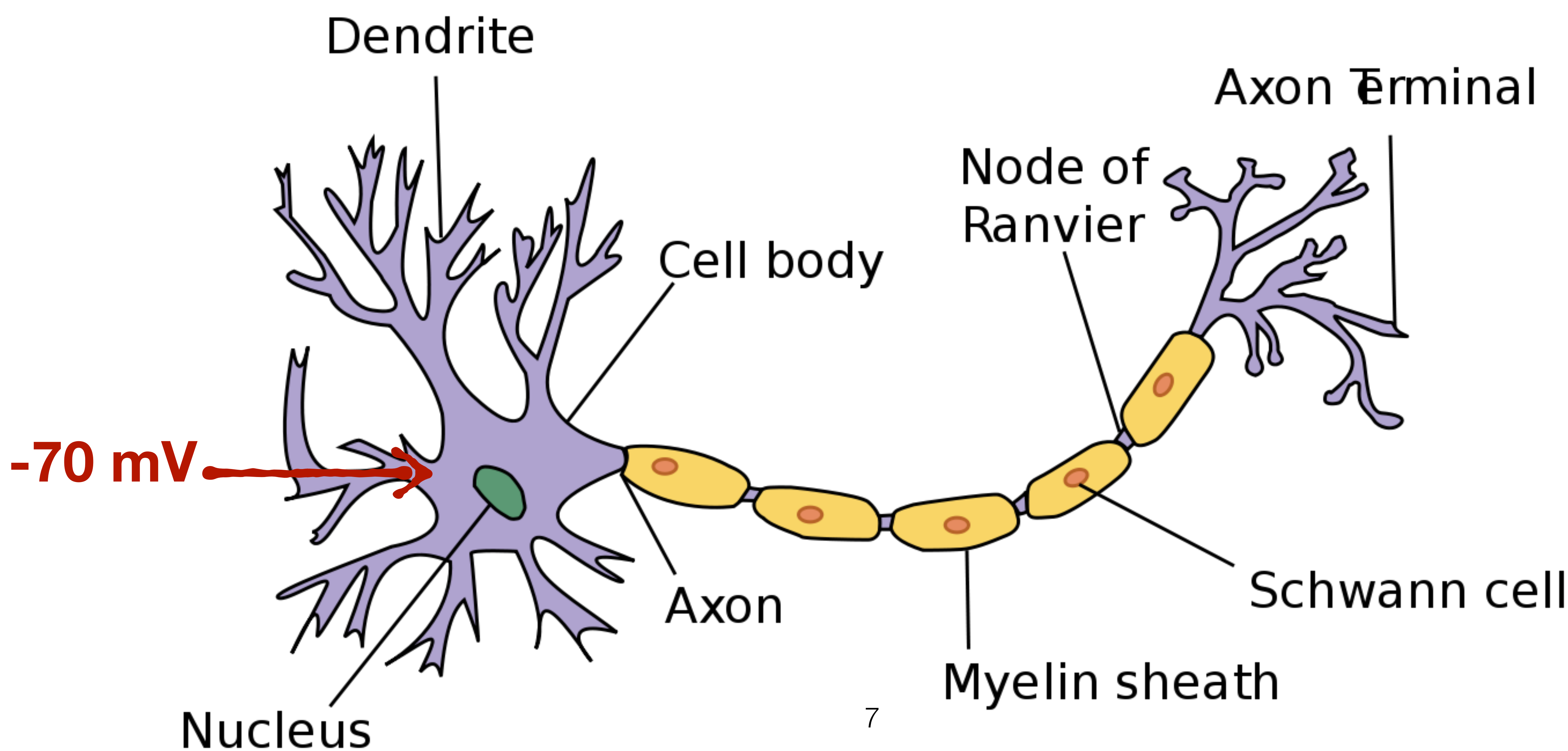
What a neuron do

- 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



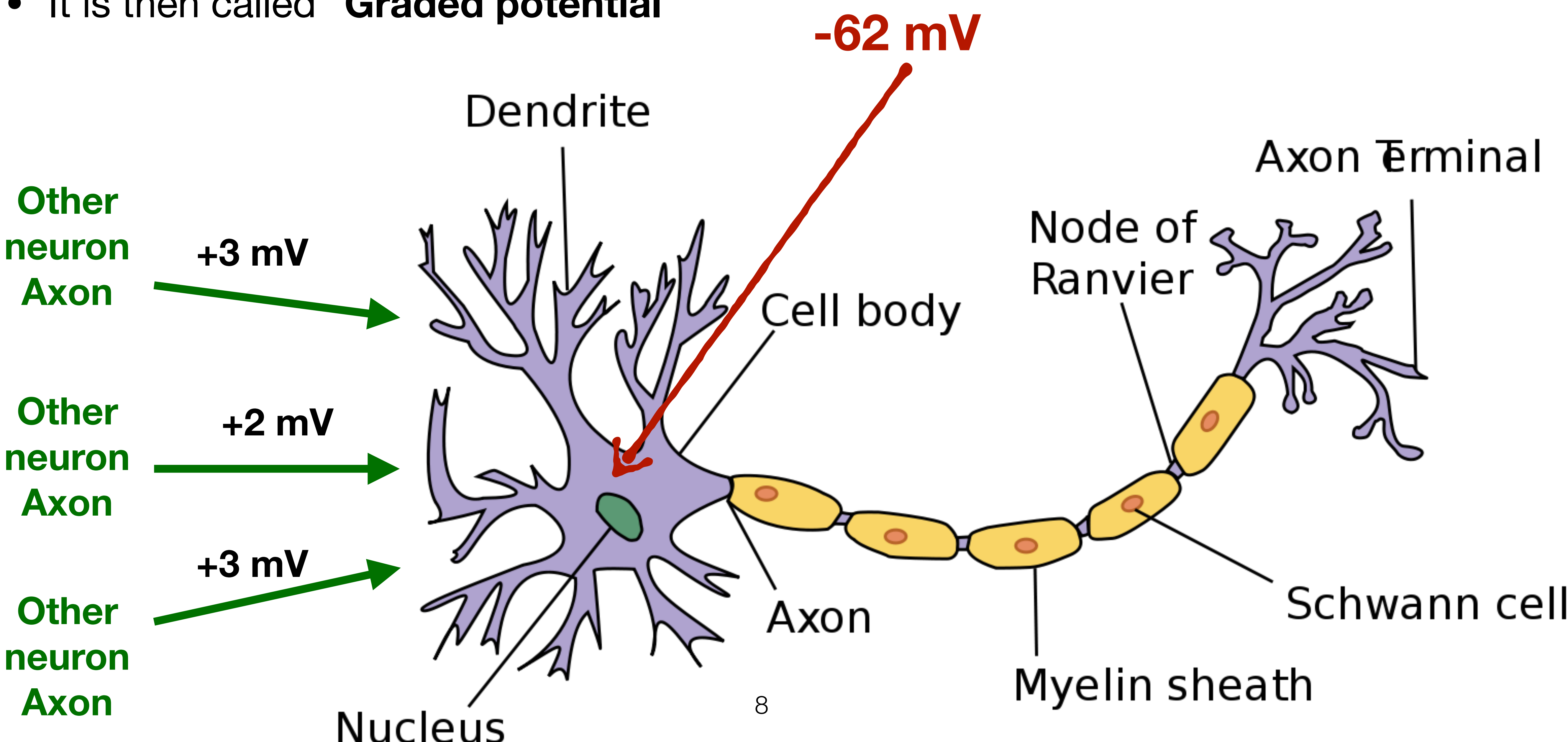
What a neuron do

- 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



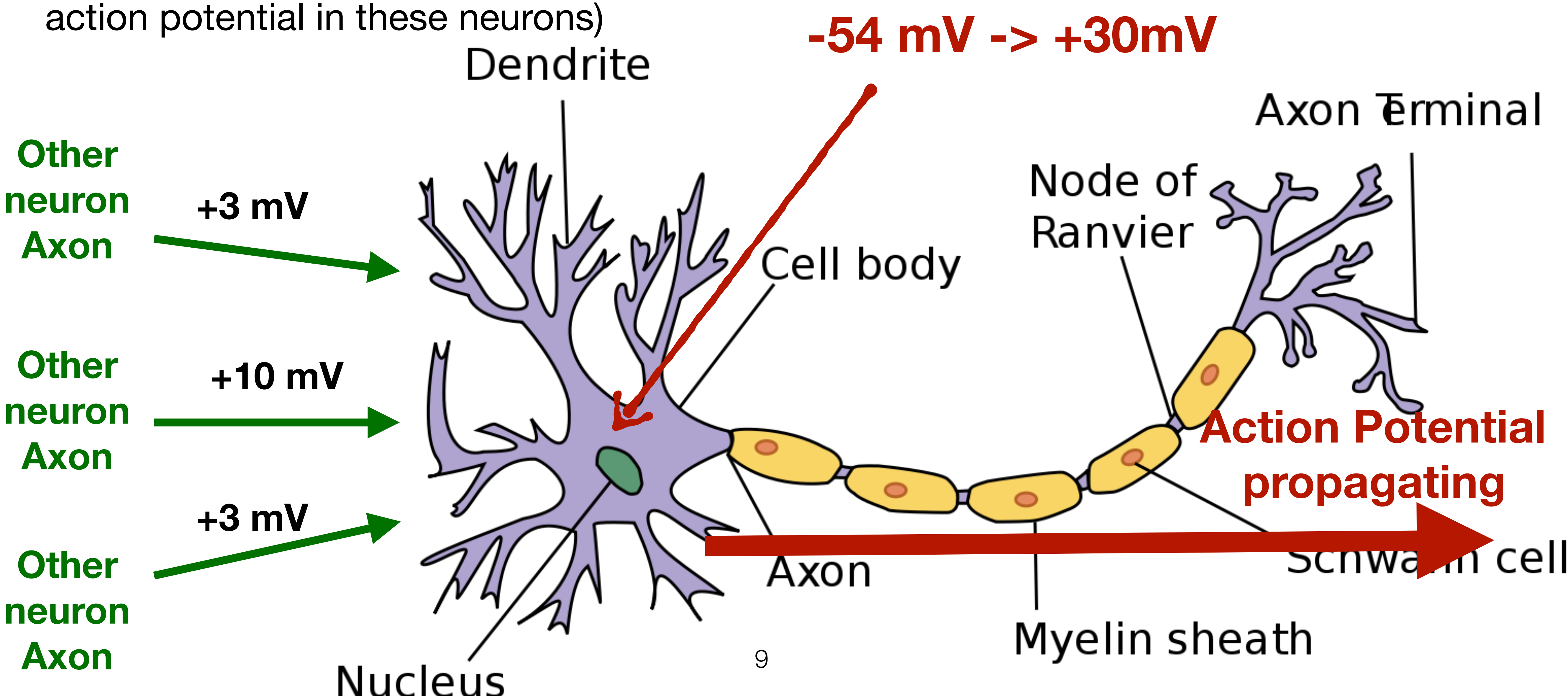
What a neuron do

- 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**"

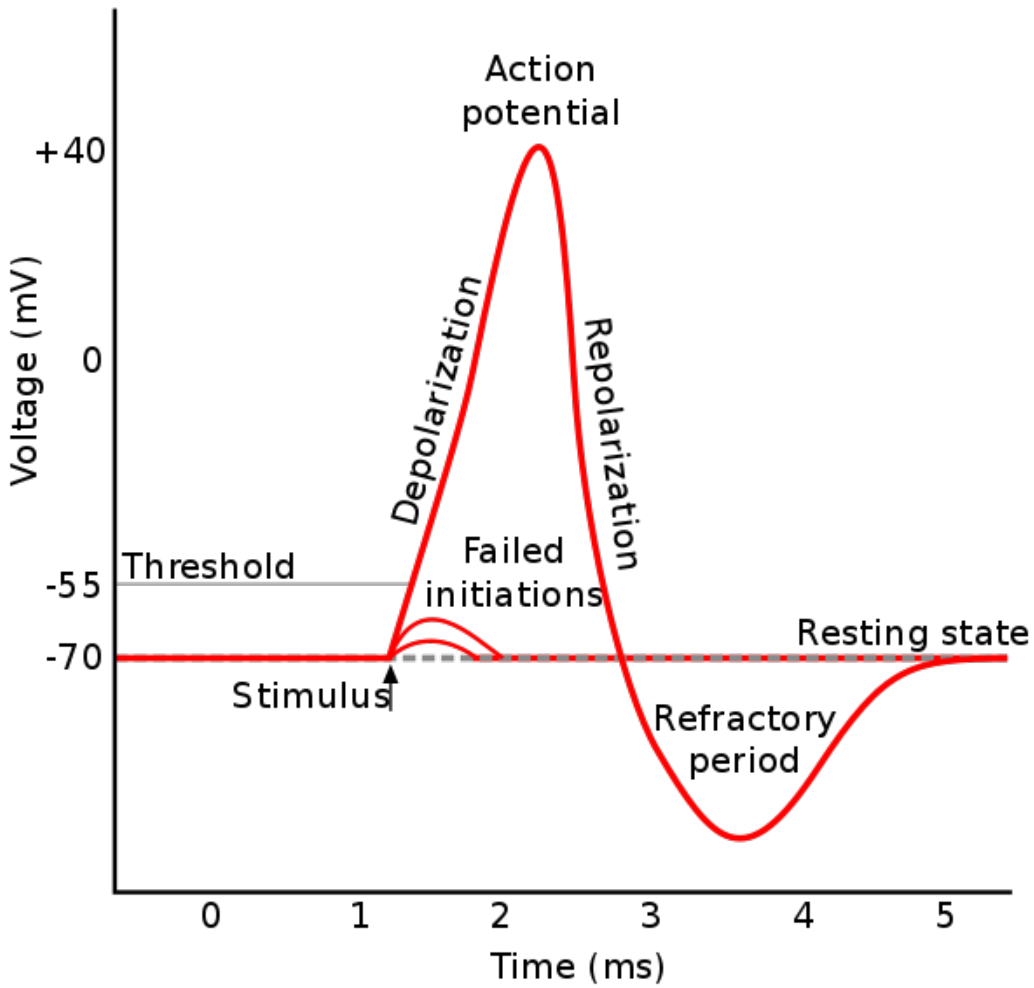


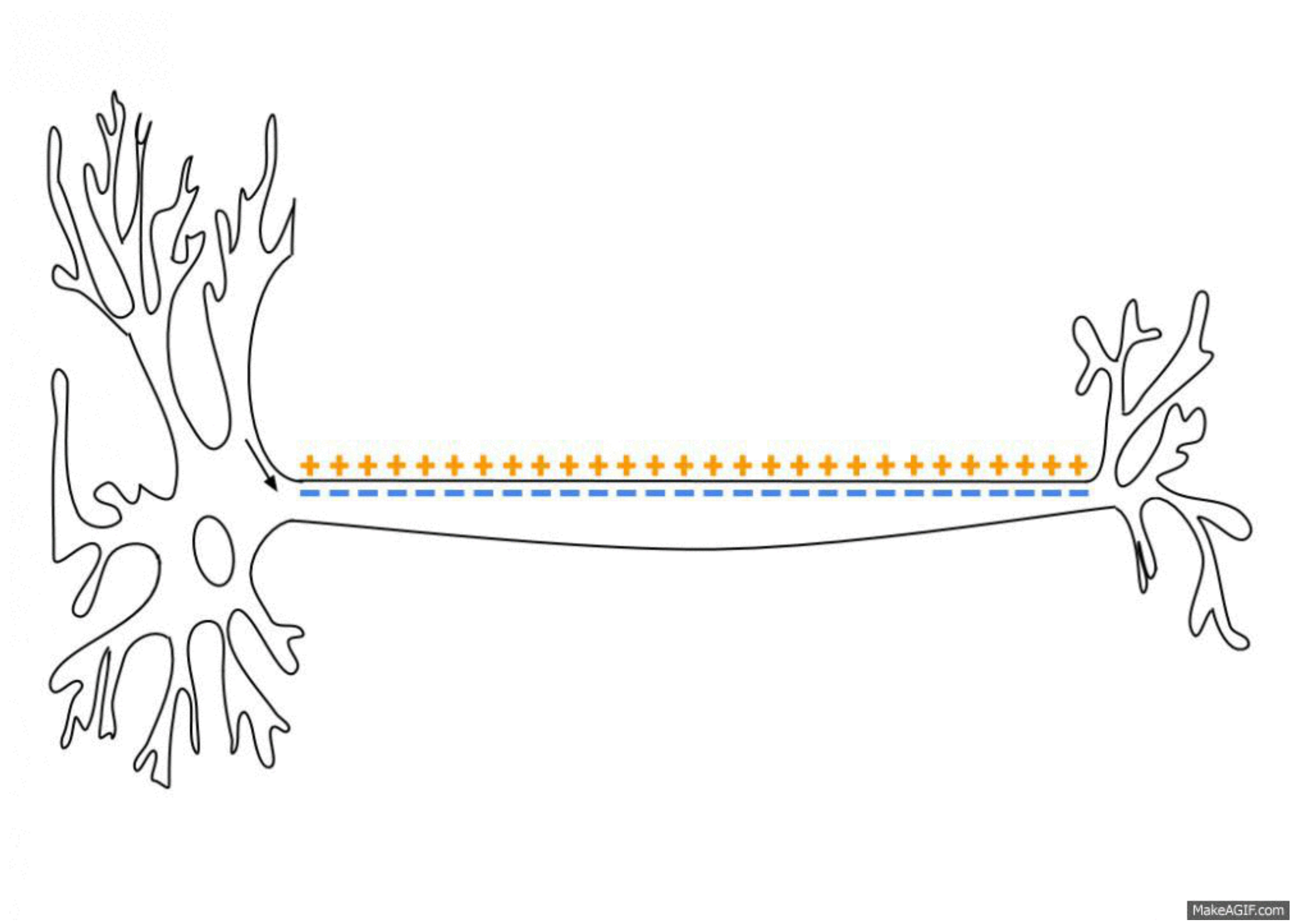
What a neuron do

- If the potential inside a neuron goes <u>above -55mV</u> 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

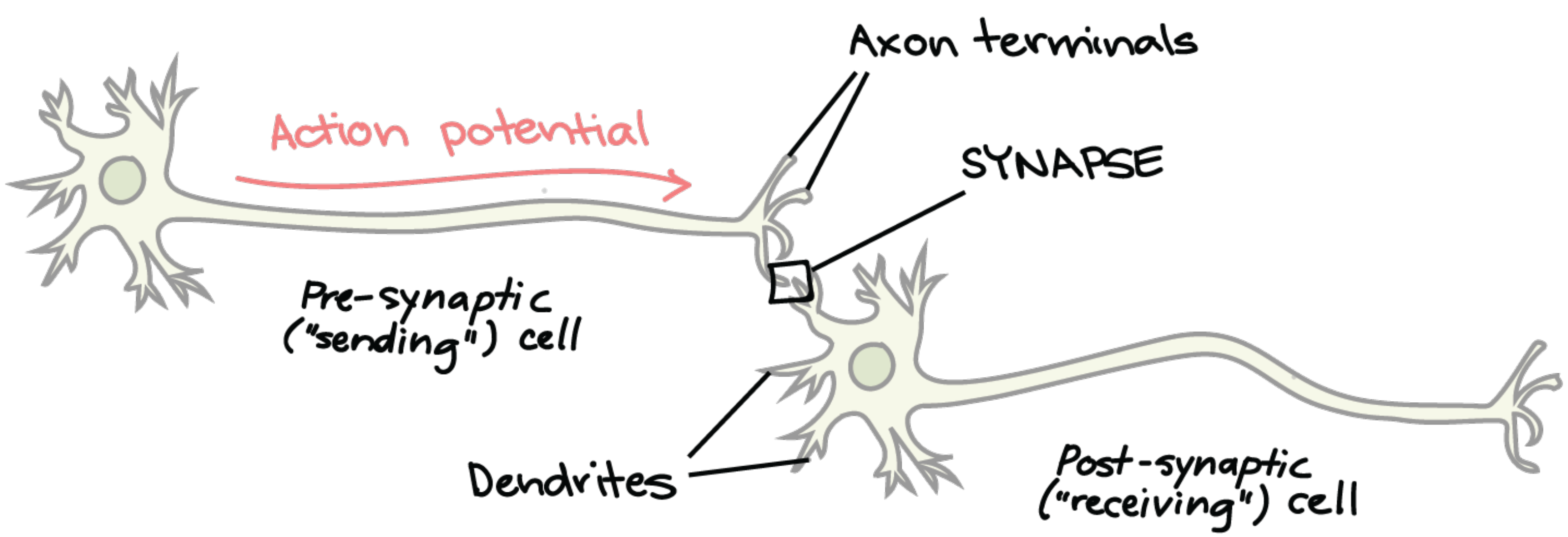




From Wikipedia, by users Chris 73 and Laurentaylorj

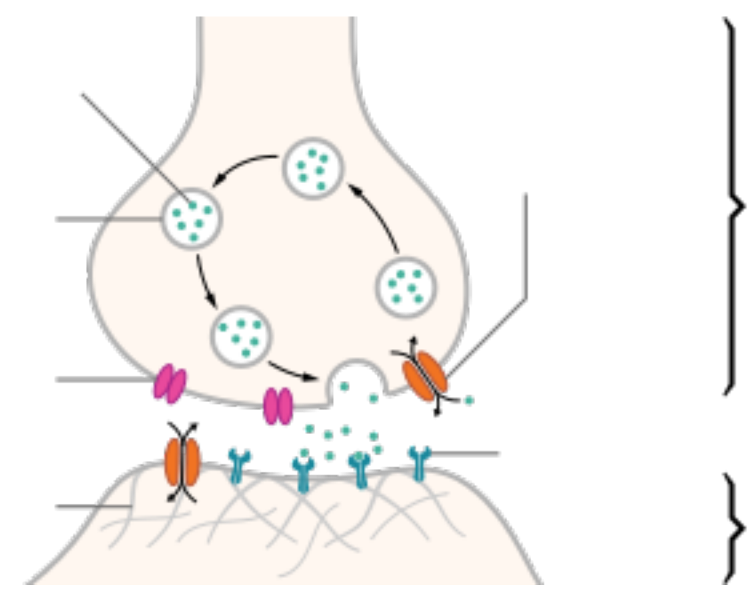
Synaptic connections

- 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**



From Khan Academy

Synapse



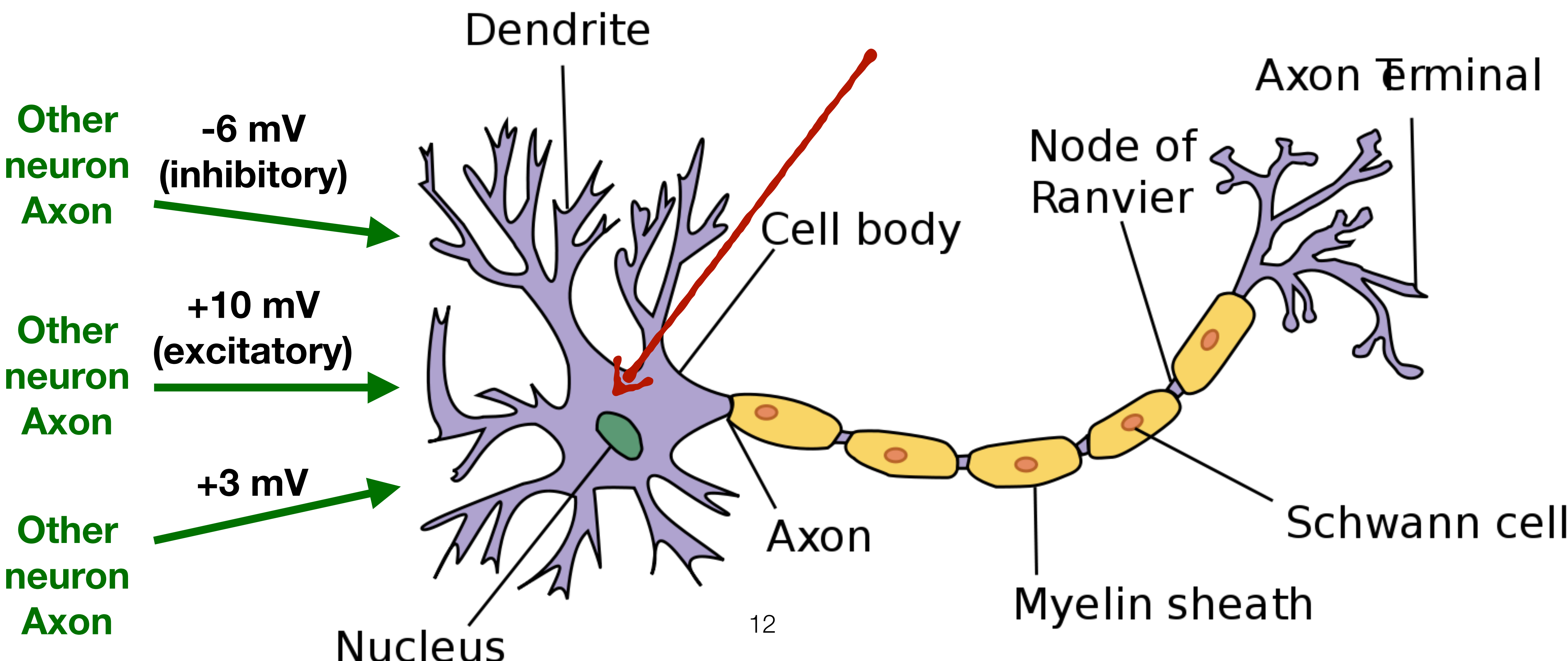
FromWikipedia

Synaptic connection

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

(70mV = Resting Potential)

70mV - 6mV + 10mV + 3mV = 63 mV

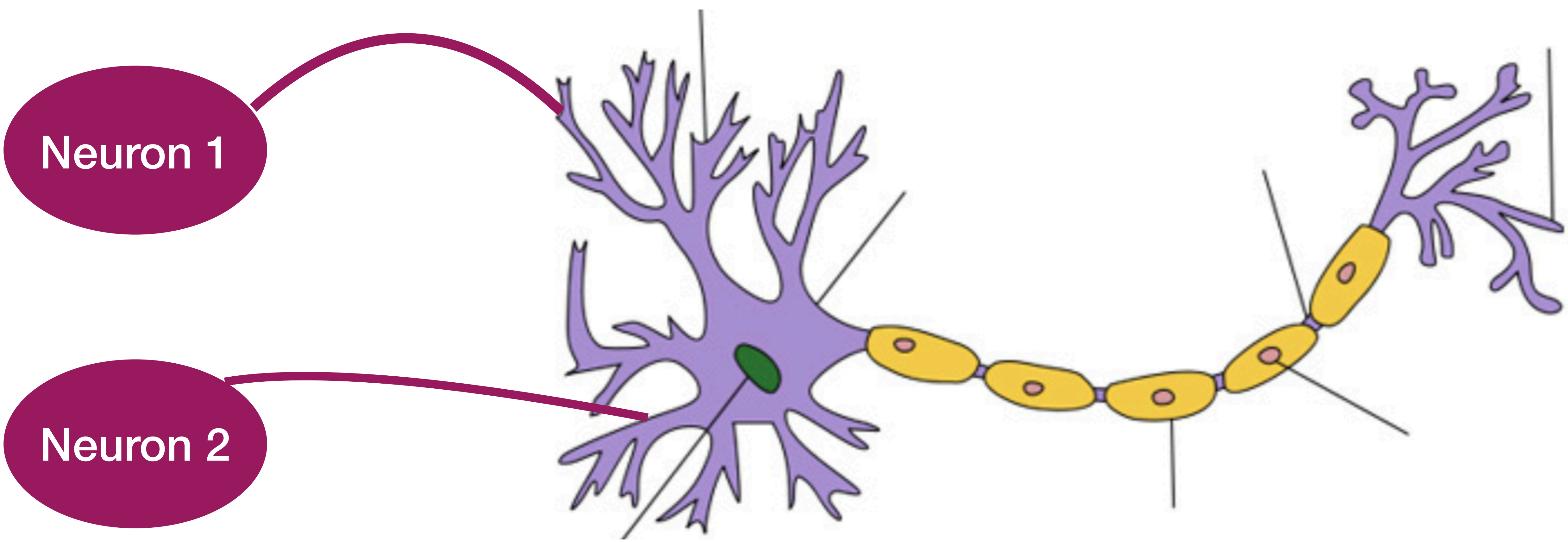


The biological process in a neuron

- 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)



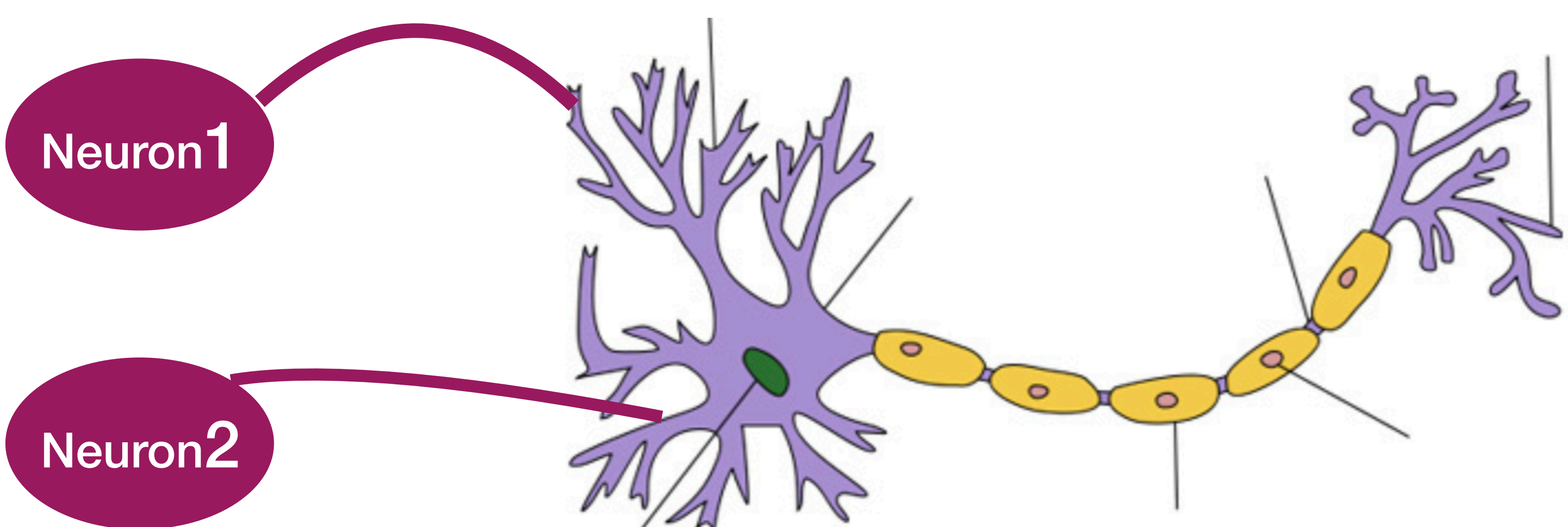
Equation of a Neuron with 2 dendrites

- 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 <u>larger than a given threshold</u>, the neuron fires an <u>*action potential*</u> on the **axone**



Equation of a Neuron with 2 dendrites

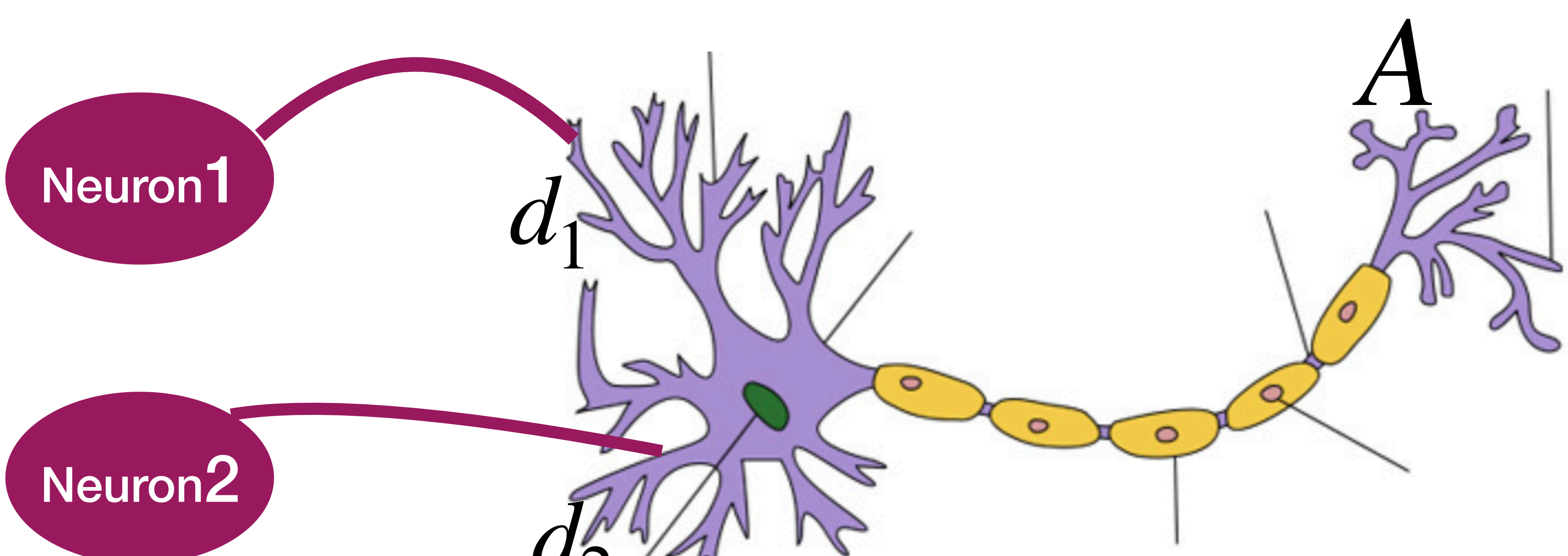
- Now, let us try to describe “mathematically” what this neuron is doing:

    1. Some input potentials arrives on its **dendrites** $\boxed{d_1 \quad d_2}$

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

    $$\boxed{gradedP = -70mV + d_1 + d_2}$$

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



$A = -70mV$

If $gradedP <= -55mV$

$A = +30mV$

If $gradedP > -55mV$

Equation of a Neuron with 2 dendrites

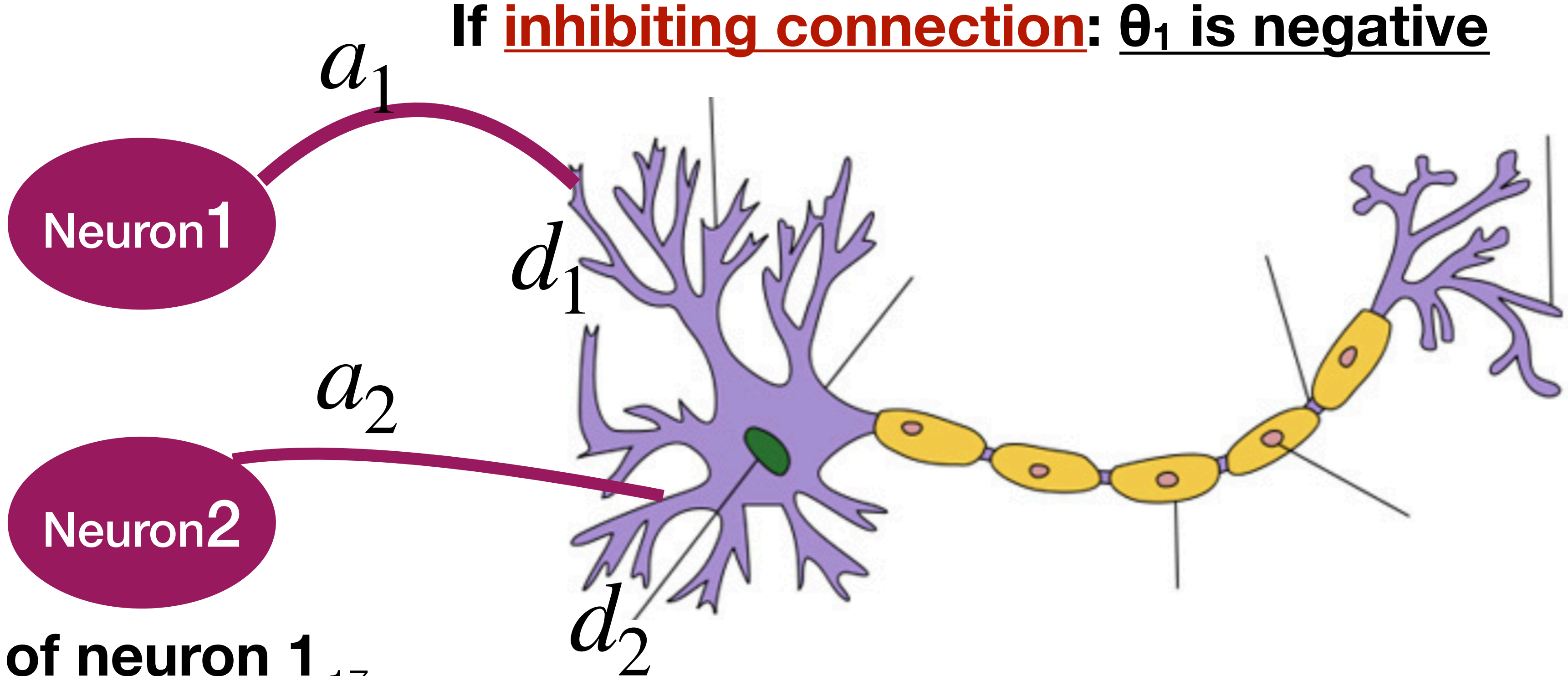
- The <u>dendrite potential</u> will <u>depend</u> on the **type of the synaptic connection**. Let us suppose the synaptic connection can be represented by a <u>parameter θ</u>

$$\boxed{\begin{aligned} d_1 &= \theta_1 \times a_1 \\ d_2 &= \theta_2 \times a_2 \end{aligned}}$$

If <u>strong excitatory connection</u> between the axone of Neuron1 and the dendrite: <u>θ_1 is large</u>

If <u>weak excitatory connection</u> between the axone of Neuron1 and the dendrite: <u>θ_1 is small</u>

If <u>inhibiting connection</u>: <u>θ_1 is negative</u>



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:

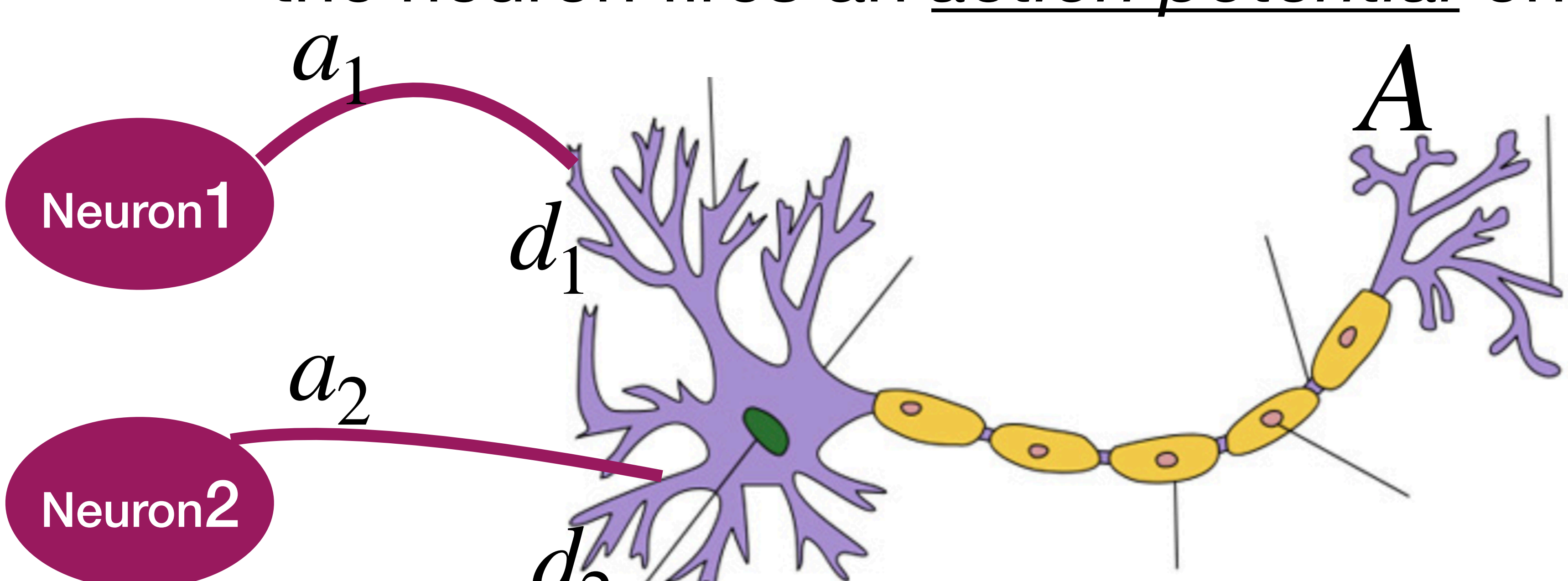
$$\boxed{d_1 = \theta_1 \times a_1 \qquad d_2 = \theta_2 \times a_2}$$

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

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

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

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



$$A = -70mV$$

If $gradedP \leq -55mV$

$$A = +30mV$$

If $gradedP > -55mV$

Equation of a Neuron with 2 dendrites

- Let us combine the two equations:

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

$$A = -70mV$$

If $gradedP \leq -55mV$

$$A = +30mV$$

If $gradedP > -55mV$

$$A = -70mV$$

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

$$A = +30mV$$

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

$$A = -70mV$$

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

$$A = +30mV$$

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

Equation of a Neuron with 2 dendrites

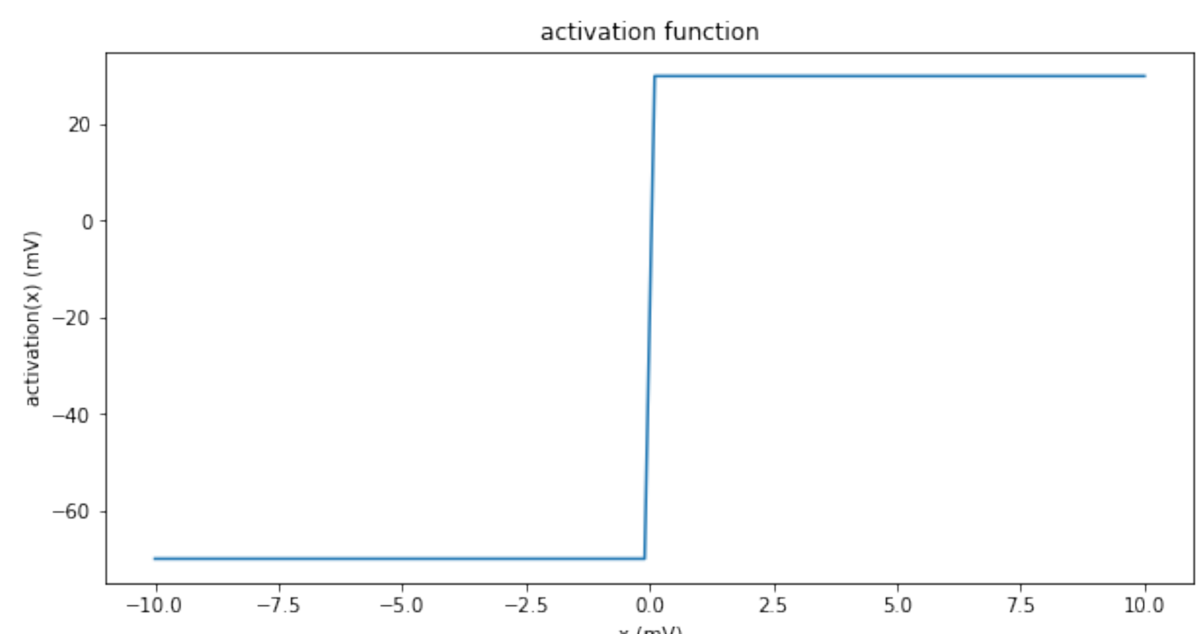
$A = -70mV$

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

$A = +30mV$

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

- We can define an **activation** function:
    - activation(*x*) = - *70mV* **if** *x* <= *0 // 30mV* ***if*** *x* > *0*



Then we can write:

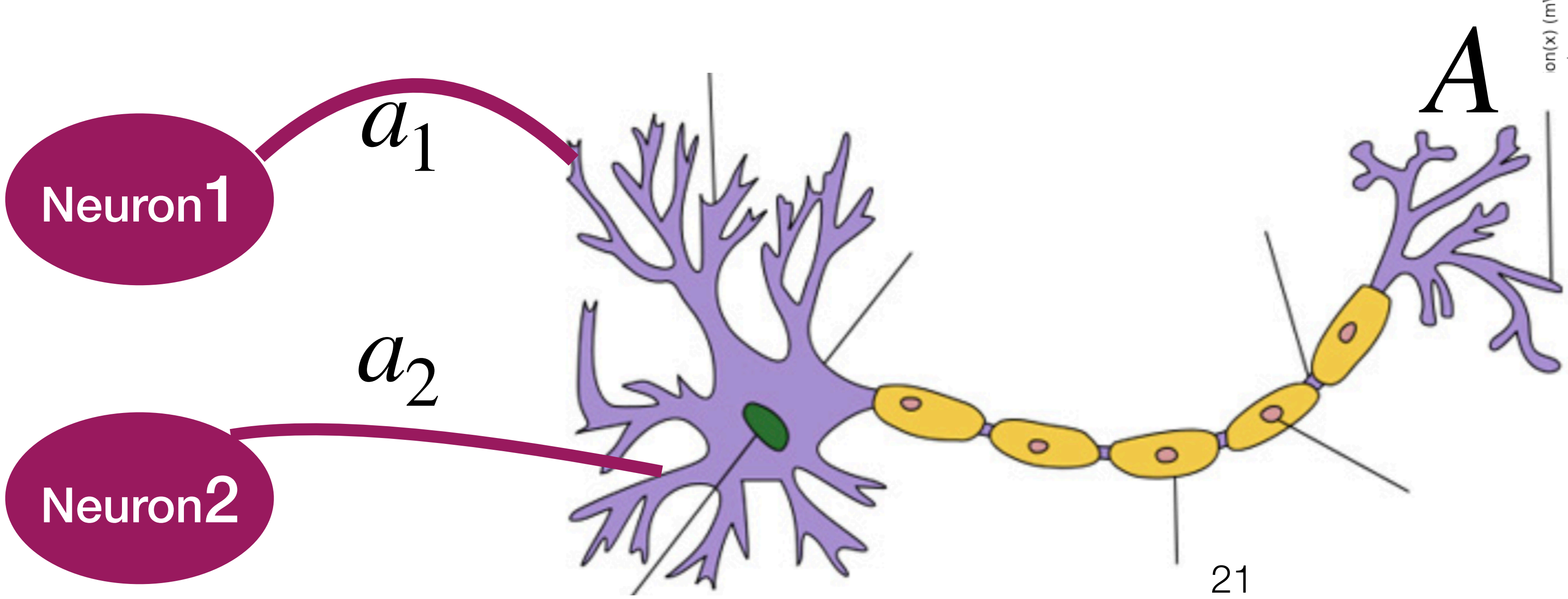
$$A = activation(-15mV + \theta_1 \times a_1 + \theta_2 \times a_2)$$

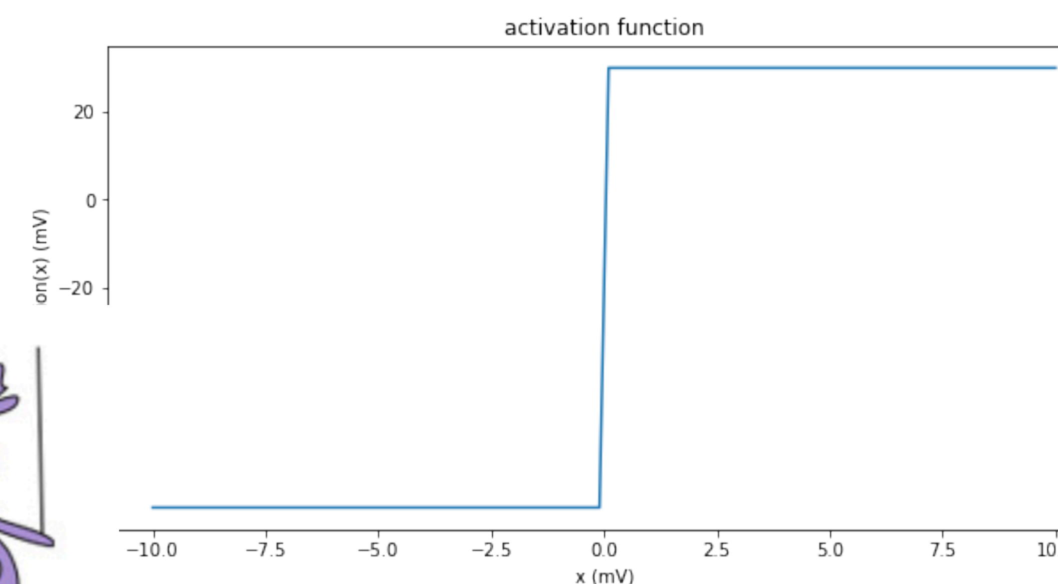
Equation of a Neuron with 2 dendrites

- Our final equation:

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

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

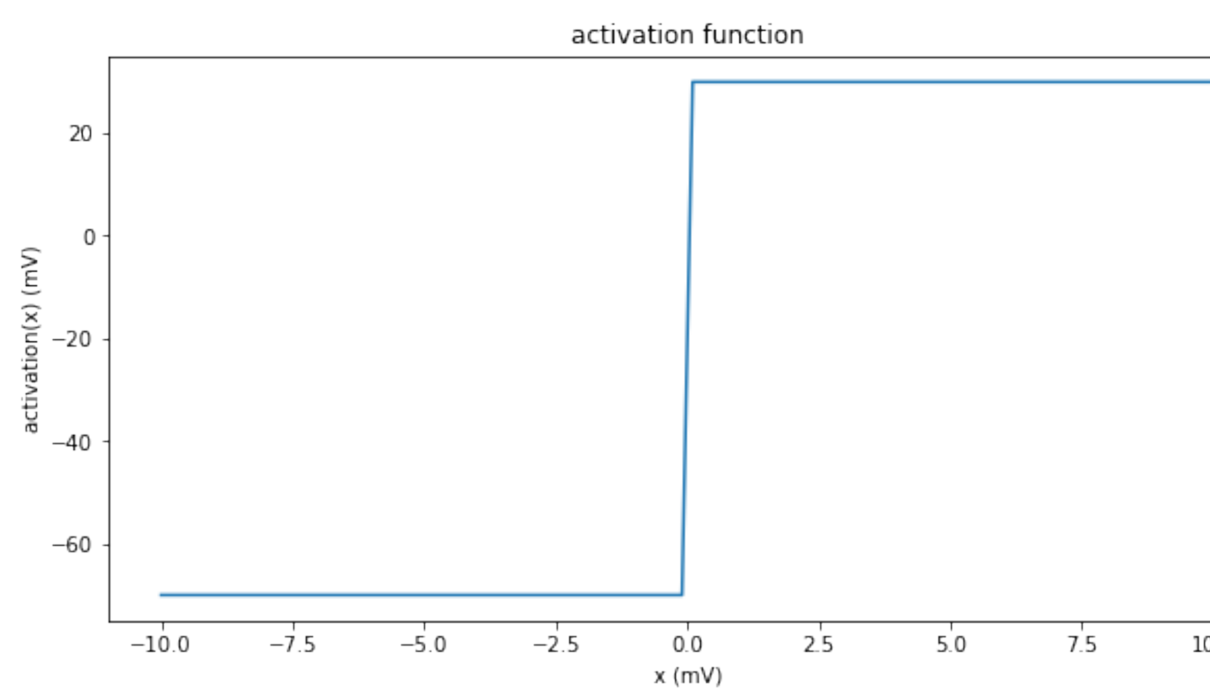




Equation of a Neuron with 2 dendrites

- Our final equation: $\boxed{A = activation(-15mV + \theta_1 \times a_1 + \theta_2 \times a_2)}$
  - 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)* = -70mV **if** x < 0 // +30mV **if** x > 0

This equation reminds me something



From Biology to Machine Learning

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

activation(x) = -70mV **if** x < 0 **//** +30mV **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 —> 0 and +30 mV —> 1

From Biology to Machine Learning

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

activation(x) = ***0* if** x < 0 ***//*** ***1* if** x > 0

- Then, let us say <u>our action potential is one</u>, and our<u> resting potential is zero</u>
    - -70mV —> 0 and +30 mV —> 1
- Similarly, let us replace the threshold value -15mV by a parameter θ_0

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

activation(x) = ***0* if** x < 0 ***//*** ***1* if** x > 0

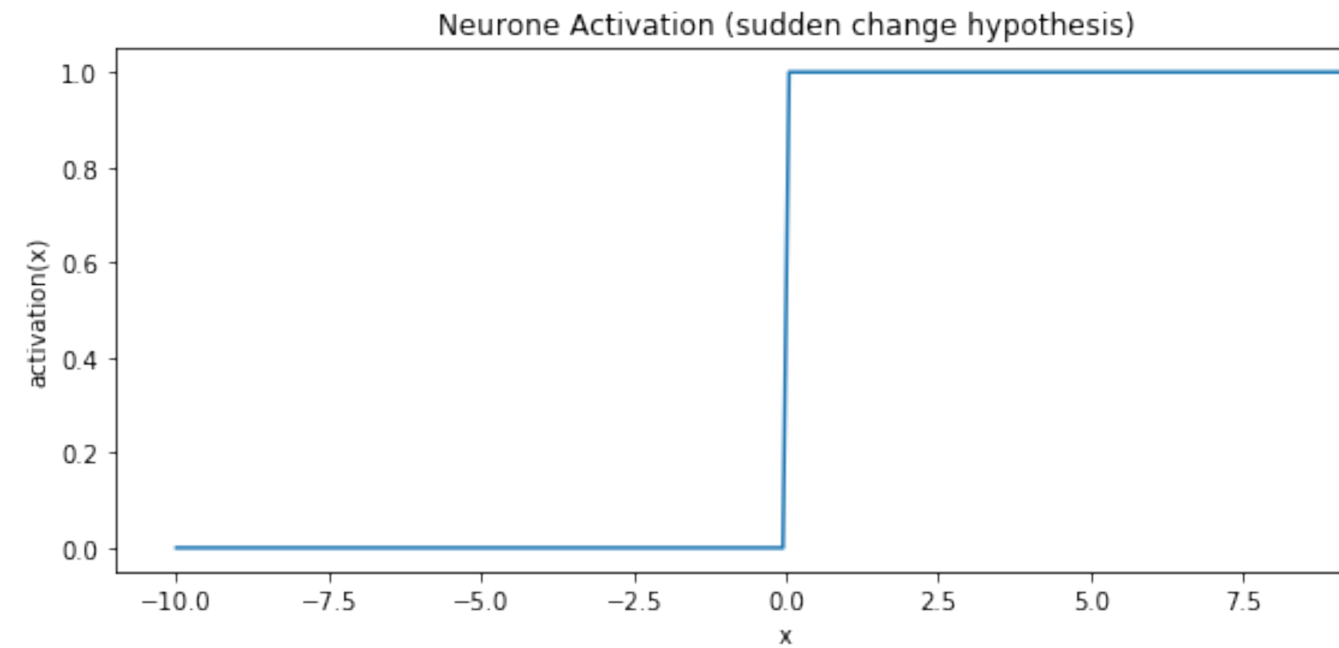
From Biology to Machine Learning

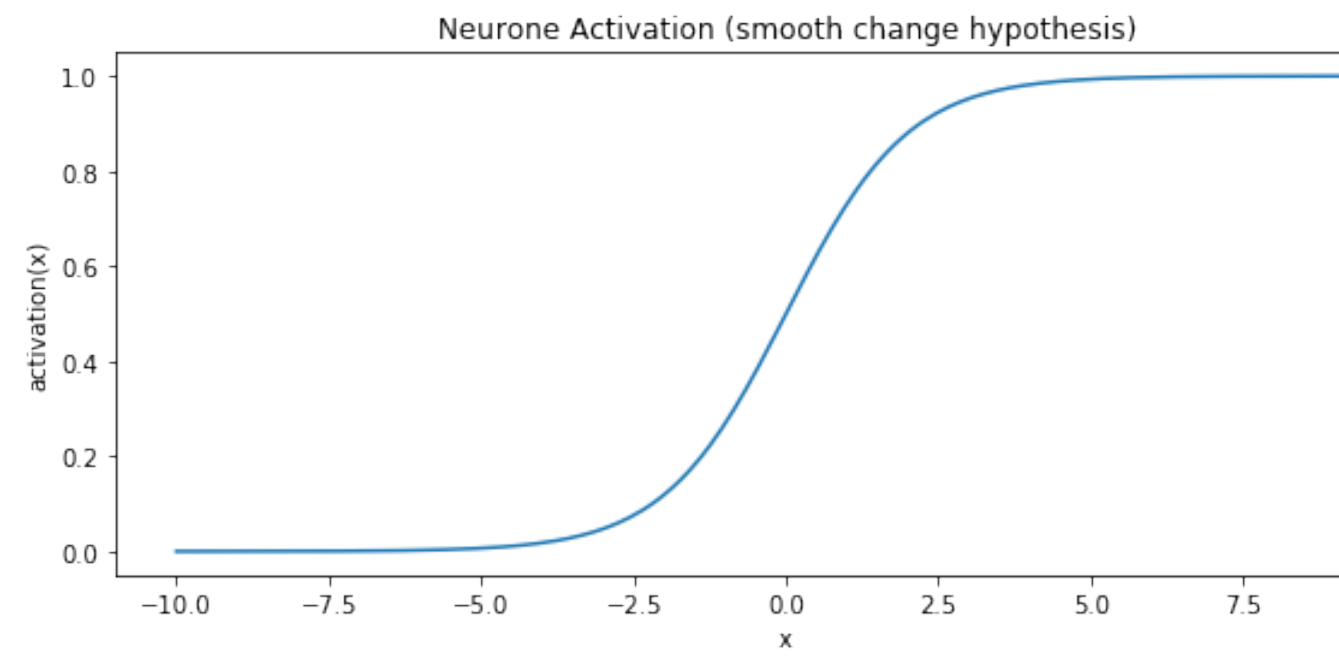
- Finally, let us consider once more the activation function

- Maybe in practice, the change on the output is not so sudden

- We could consider the activation to be smoother

activation(x) = ***0*** **if** x < 0 ***//*** ***1*** **if** x > 0

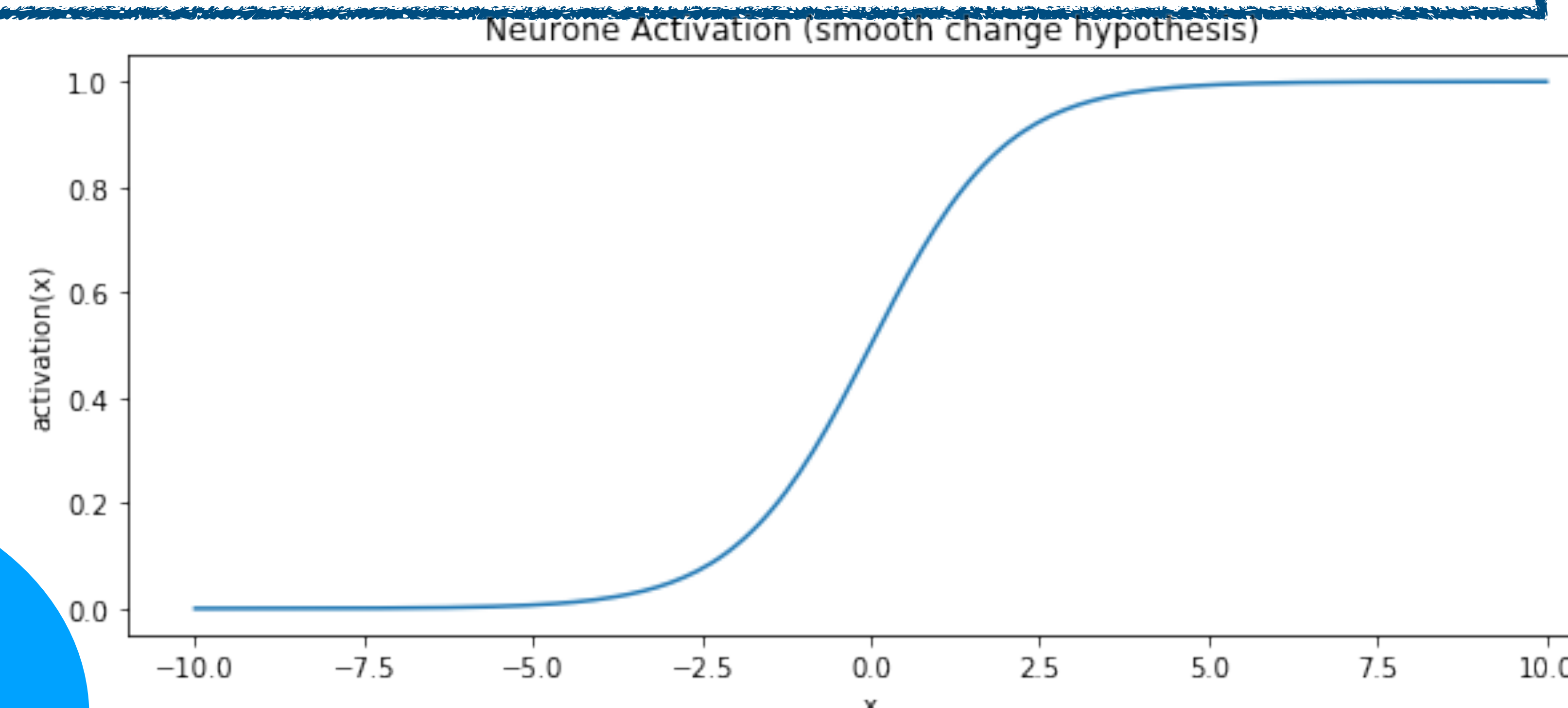




From Biology to Machine Learning

- So, finally, we have this:

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

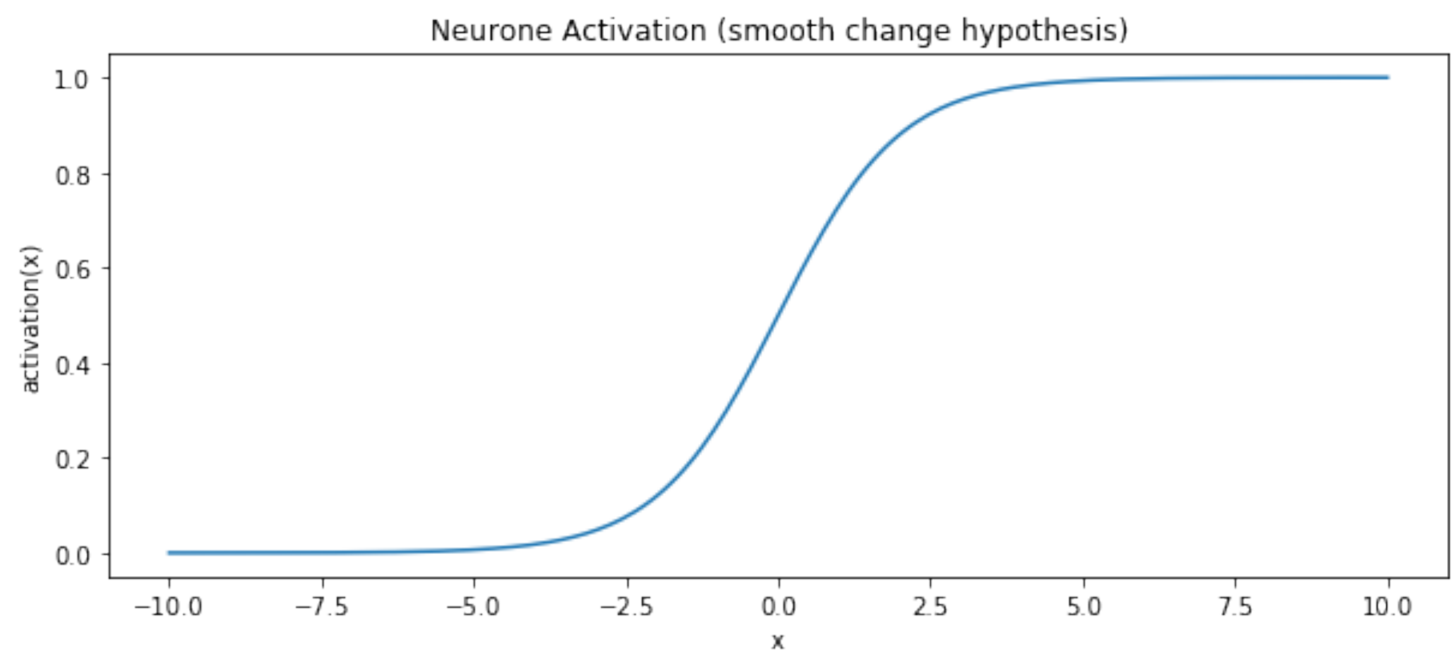


I knew it!
The logistic
classifier!

From Biology to Machine Learning

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

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



Logistic Classifier Equation:

$$score(income, age) = \theta_0 + \theta_1 \times income + \theta_2 \times age$$

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

From Biology to Machine Learning

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

- Under a <u>few simplifying assumption</u>, the **logistic classifier** is a <u>good approximation</u> 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**

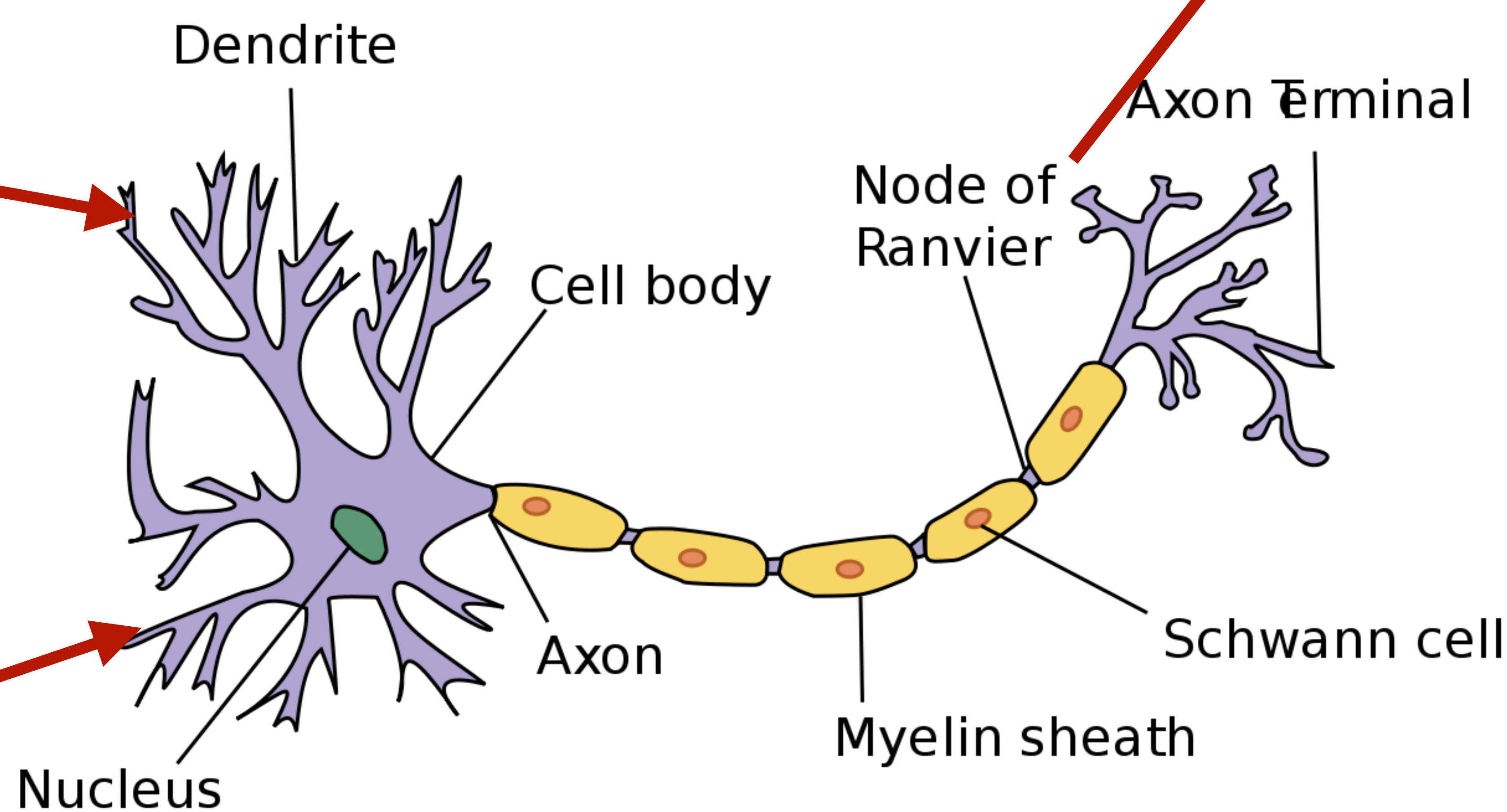
A neuron as a binary classifier

-70mV or 30mV impulse
-70mV: Left-Wing
30mV: Right Wing

VOTE

INCOME

eg. 12 mV impulse



AGE

eg. 3.2 mV impulse

Remember this?

Example

Example Data

	income	vote
1	40	0
2	70	1
3	20	0

- Suppose these parameters:

$$\theta_0 = -8$$
$$\theta_1 = 0.1$$

$$score(income) = \theta_0 + \theta_1 \times income$$

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

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

	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!

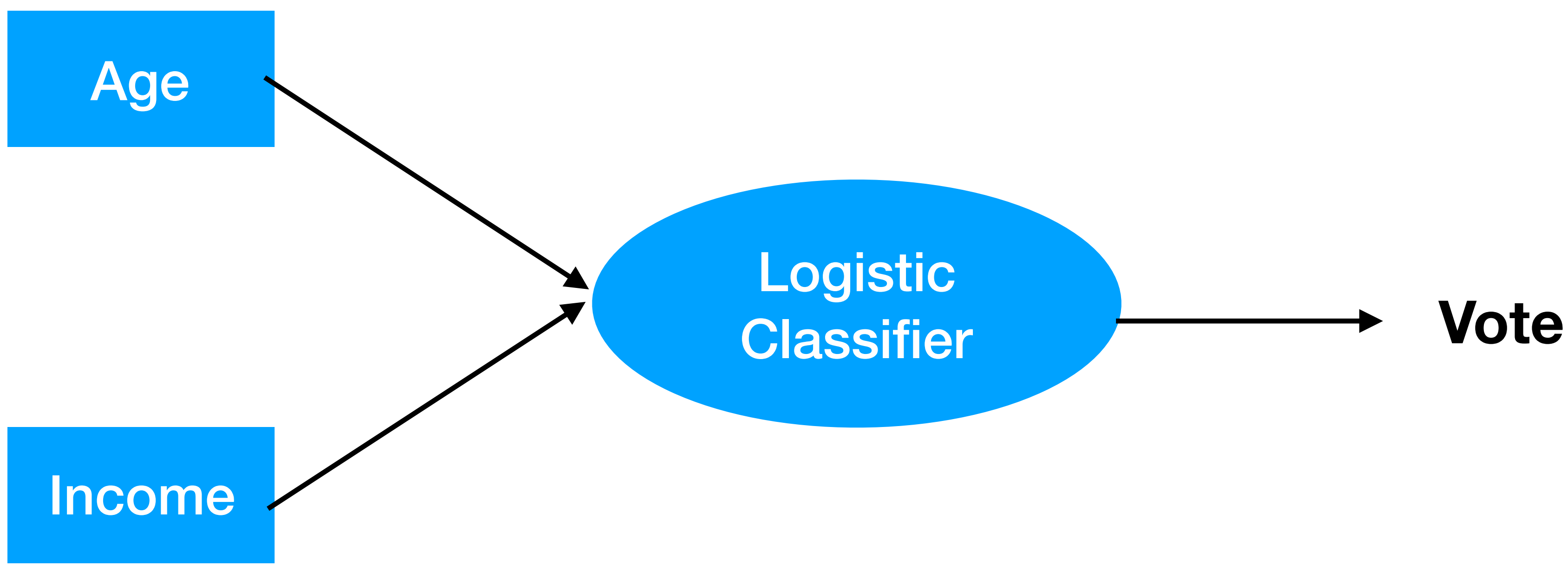
Neural Networks

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

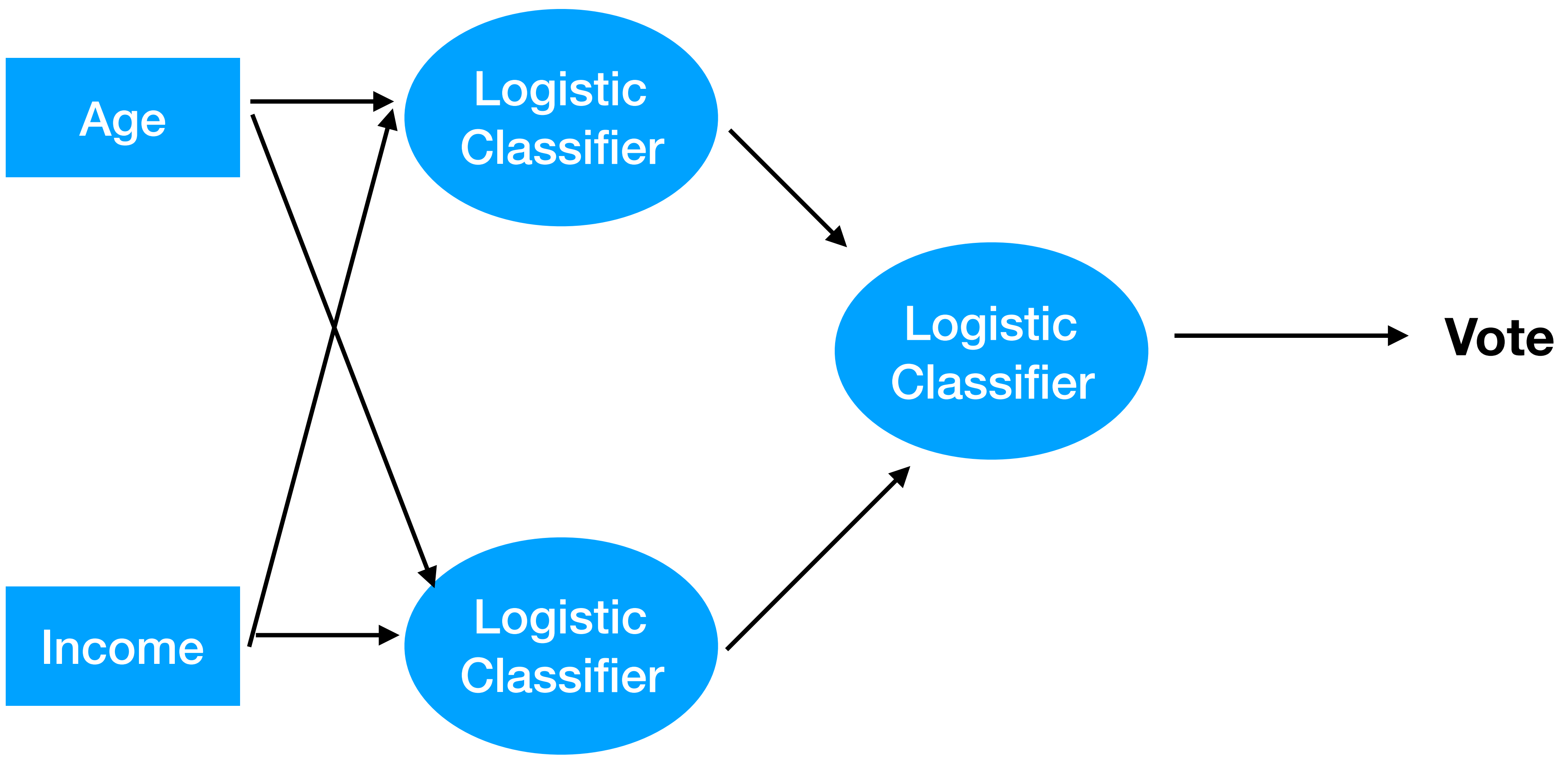
$$A = activation(\theta_0 + \theta_1 \times a_1 + \theta_2 \times a_2 + \ldots + \theta_{7000} \times a_{7000})$$

- How about interconnecting logistic classifiers? Will it work?

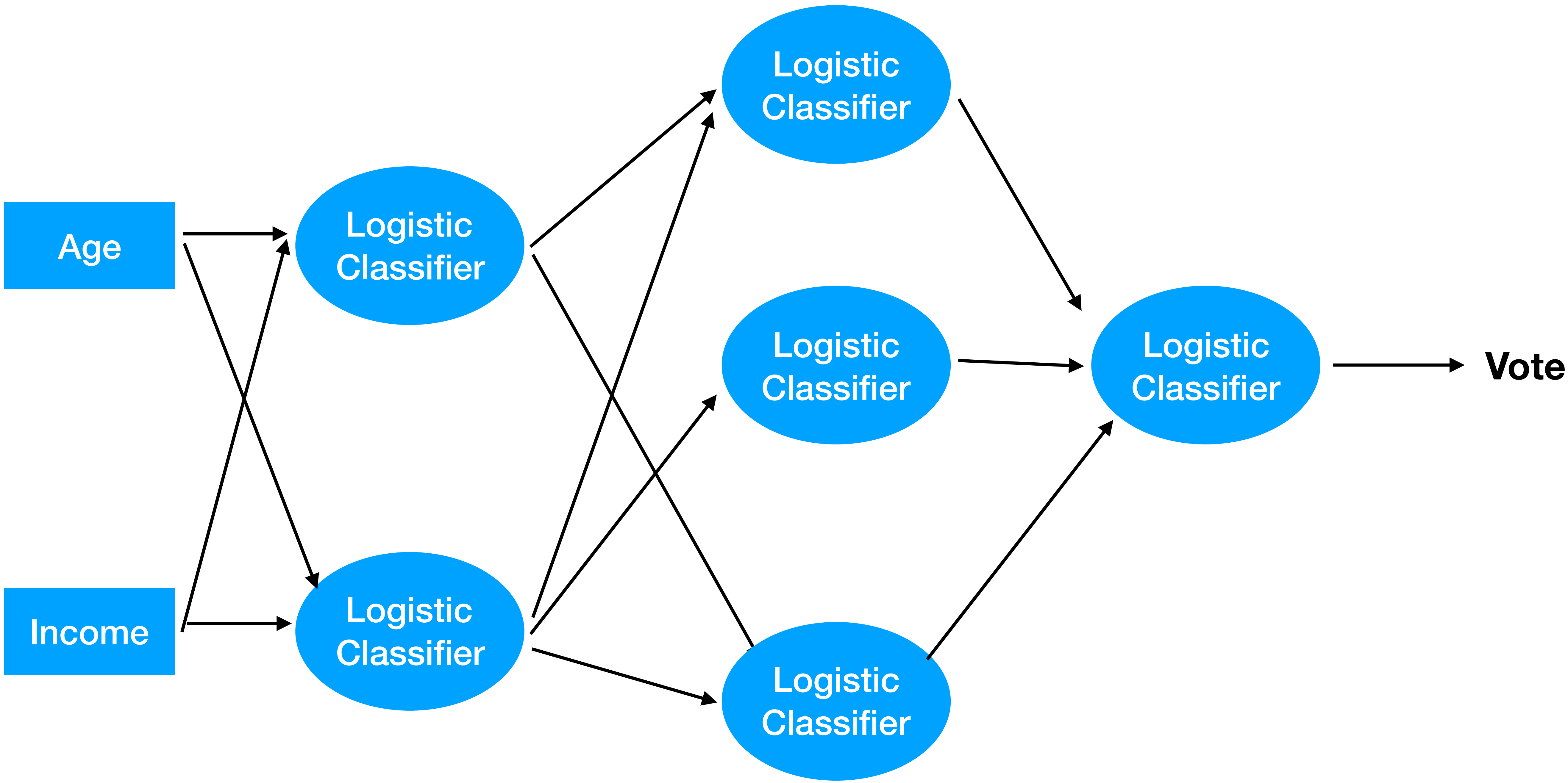
One Neuron



Three Neurons

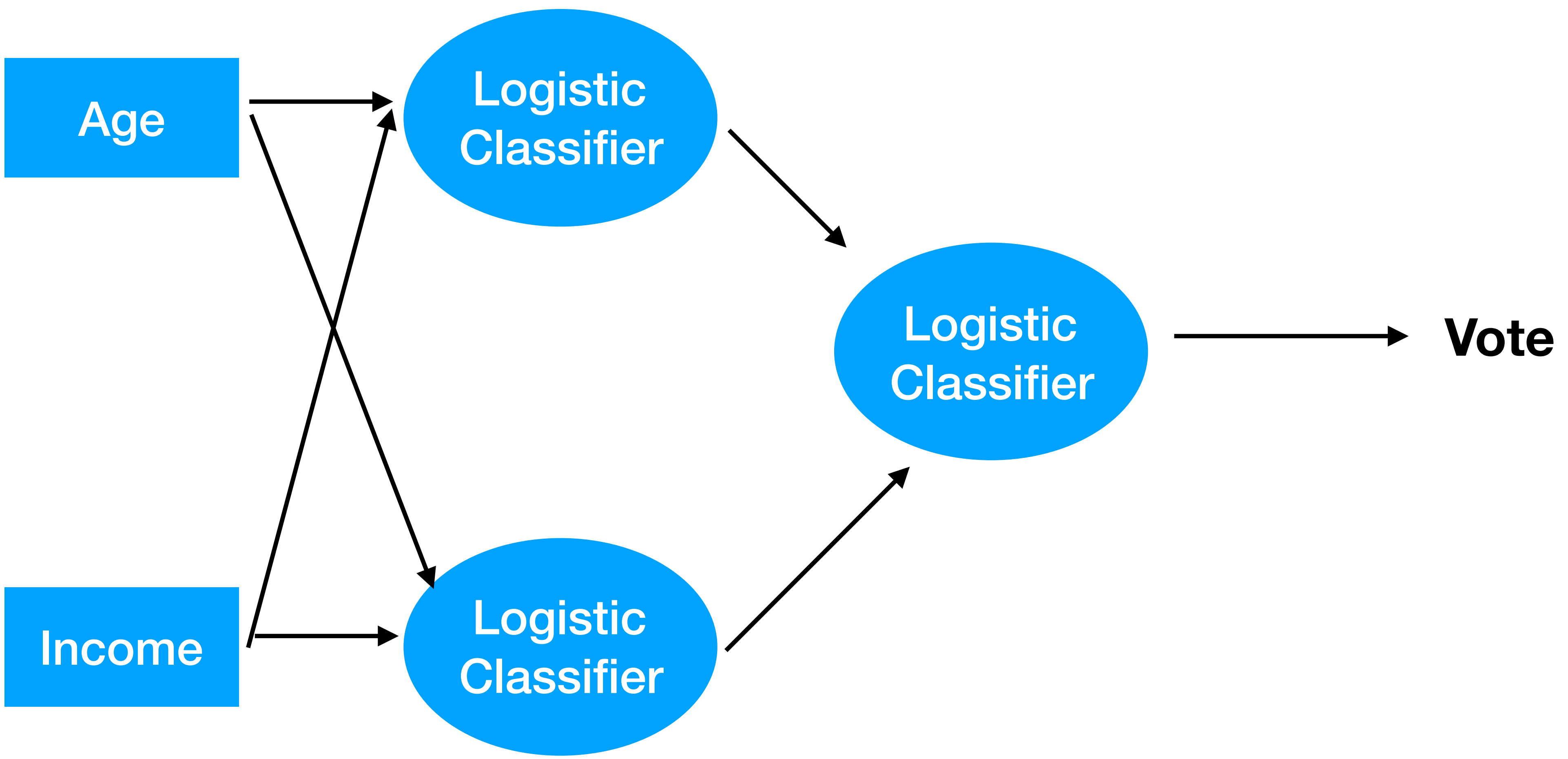


More Neurons



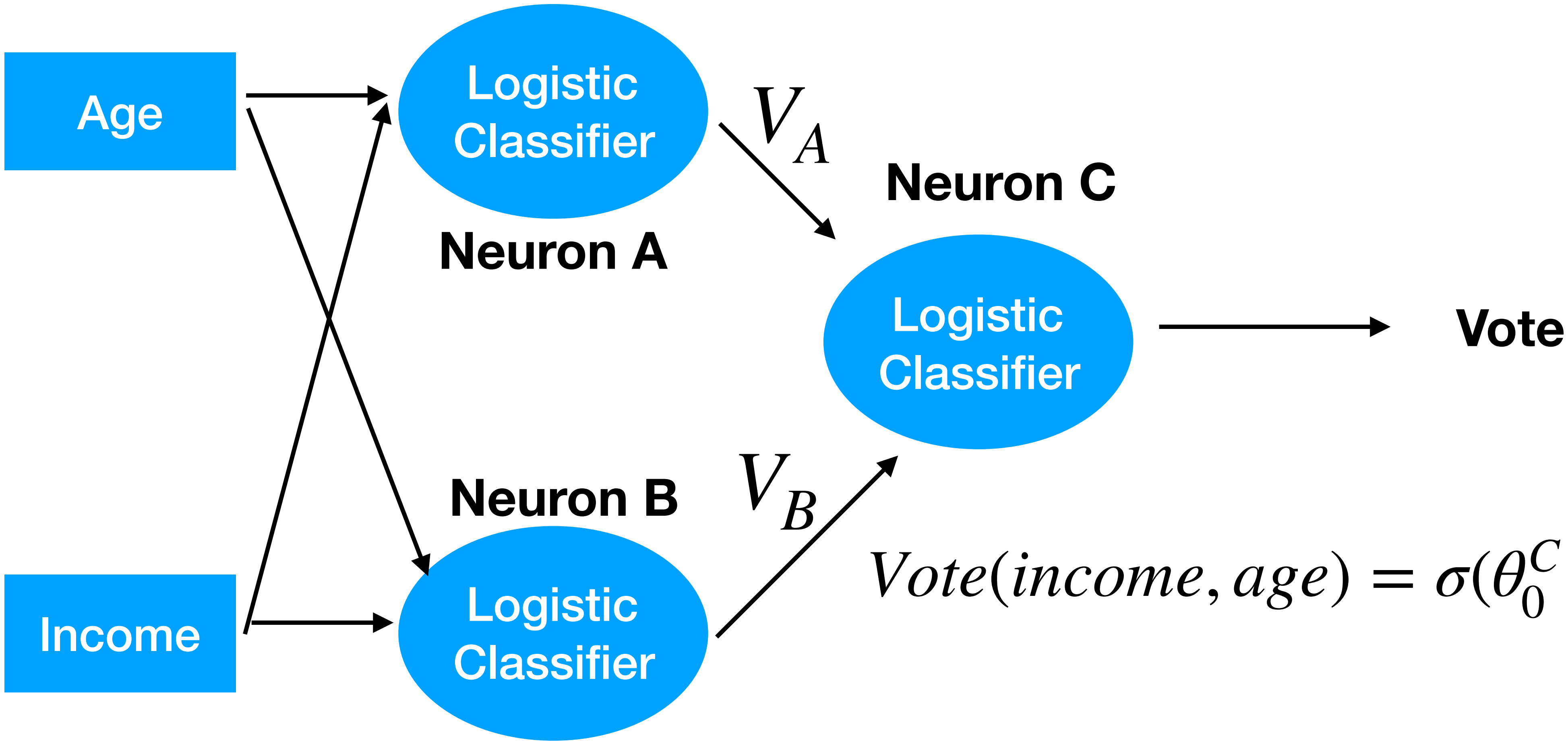
Three Neurons

What do we compute in this case?



Three Neurons

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



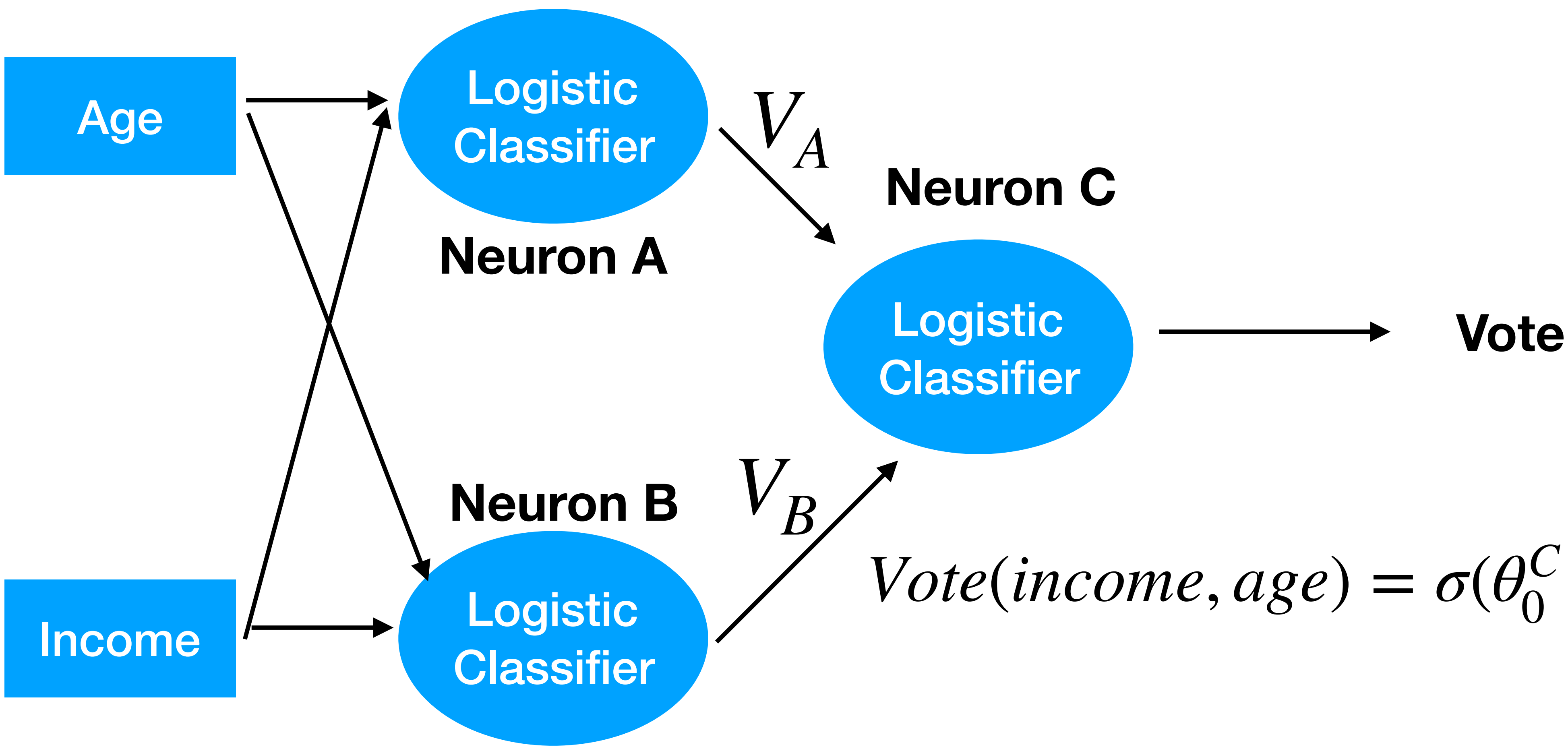
$$Vote(income, age) = \sigma(\theta_0^C + \theta_1^C \times V_A + \theta_2^C \times V_C)$$

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

Three Neurons

We now have a model with 9 parameters

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



$$Vote(income, age) = \sigma(\theta_0^C + \theta_1^C \times V_A + \theta_2^C \times V_C)$$

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

Does it work?

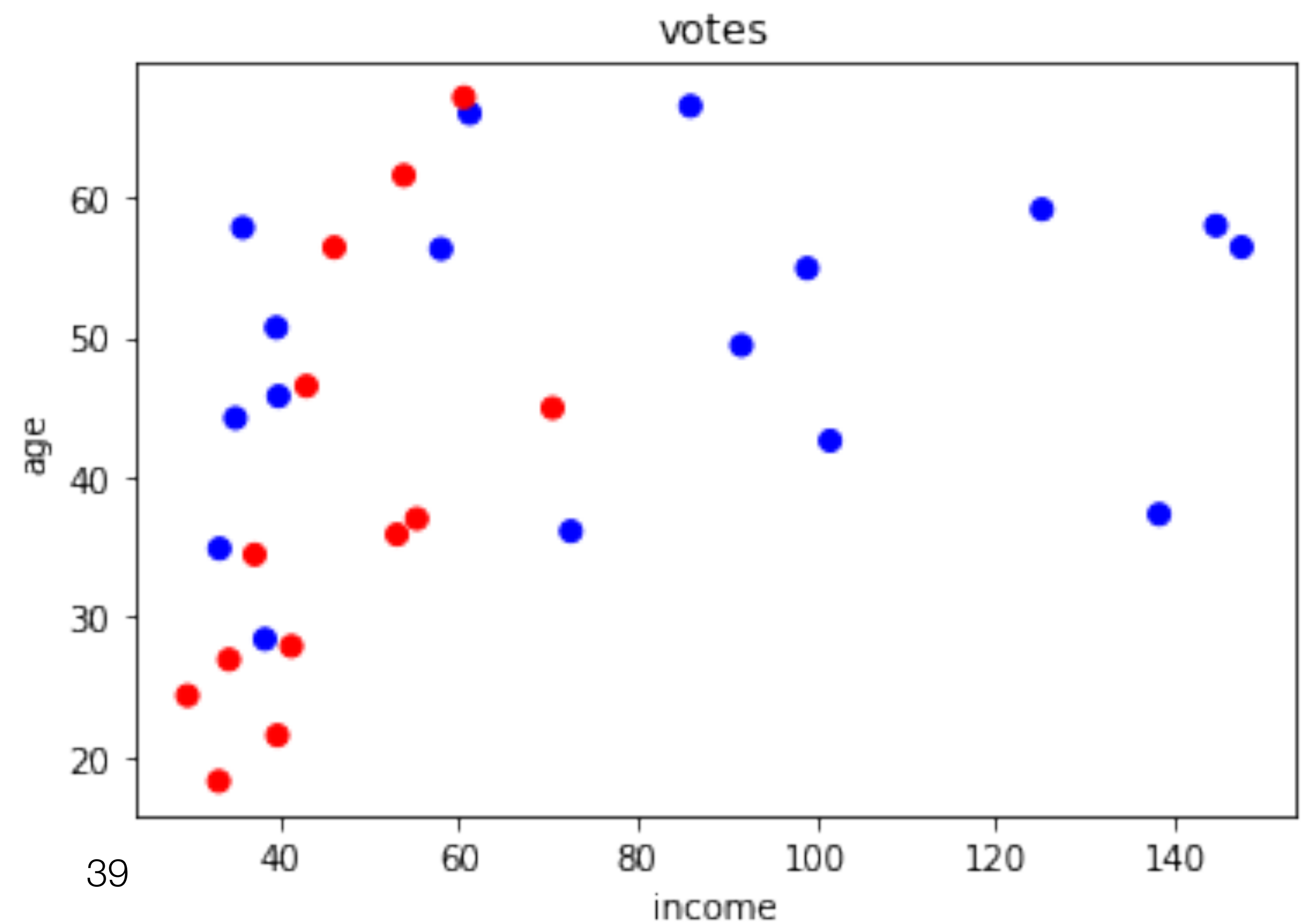
- Can we really get something better by connecting artificial neurons like this?
- Yes!
- The “power” of a network increase with the number of neurons
- If we connect many simple logistic classifiers, we get a more powerful binary classifier

Does it work?

- Let us look again at our voting data

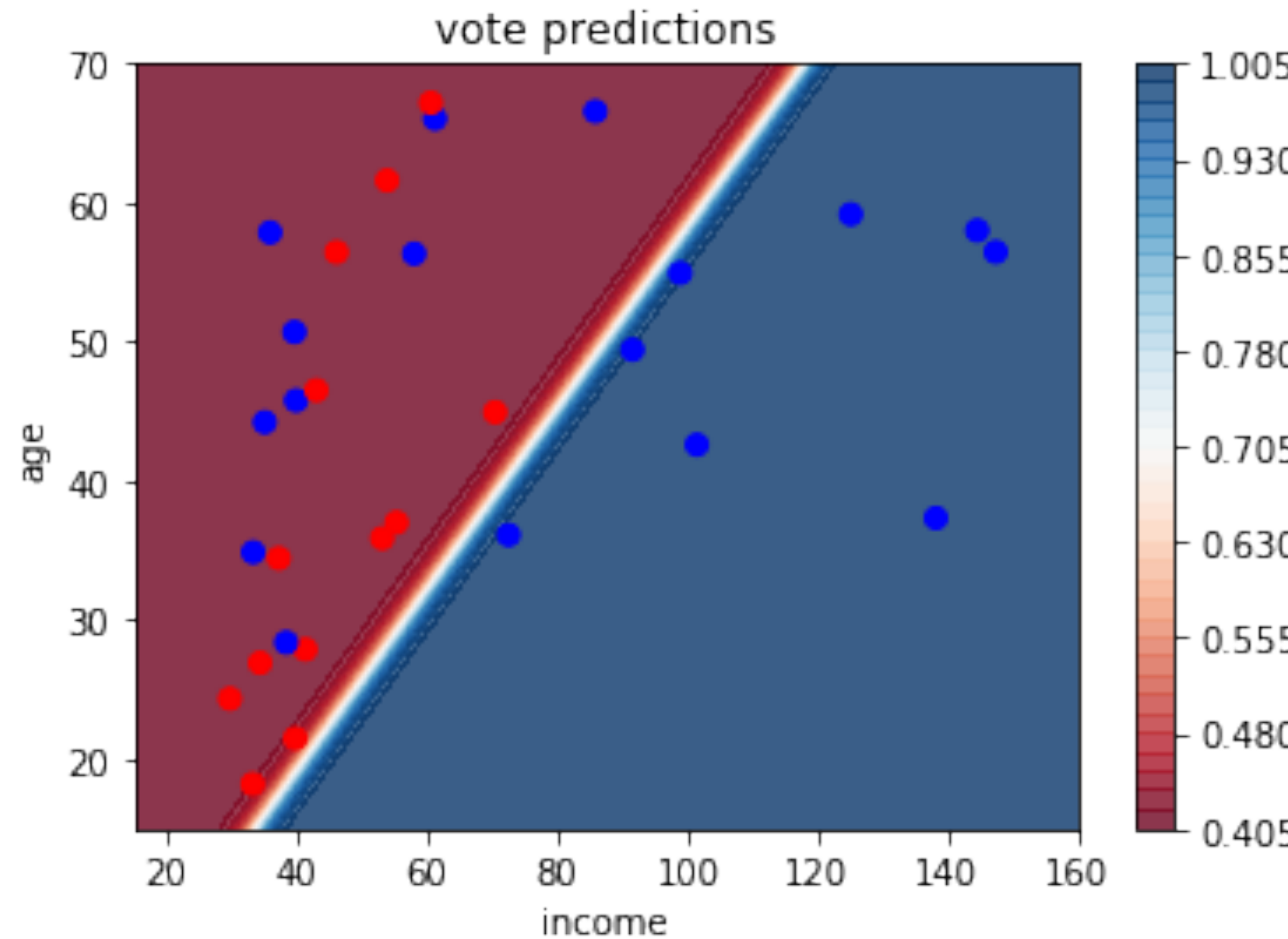
- Let us try to make different neural networks learn to predict vote knowing the income and age of somebody

	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

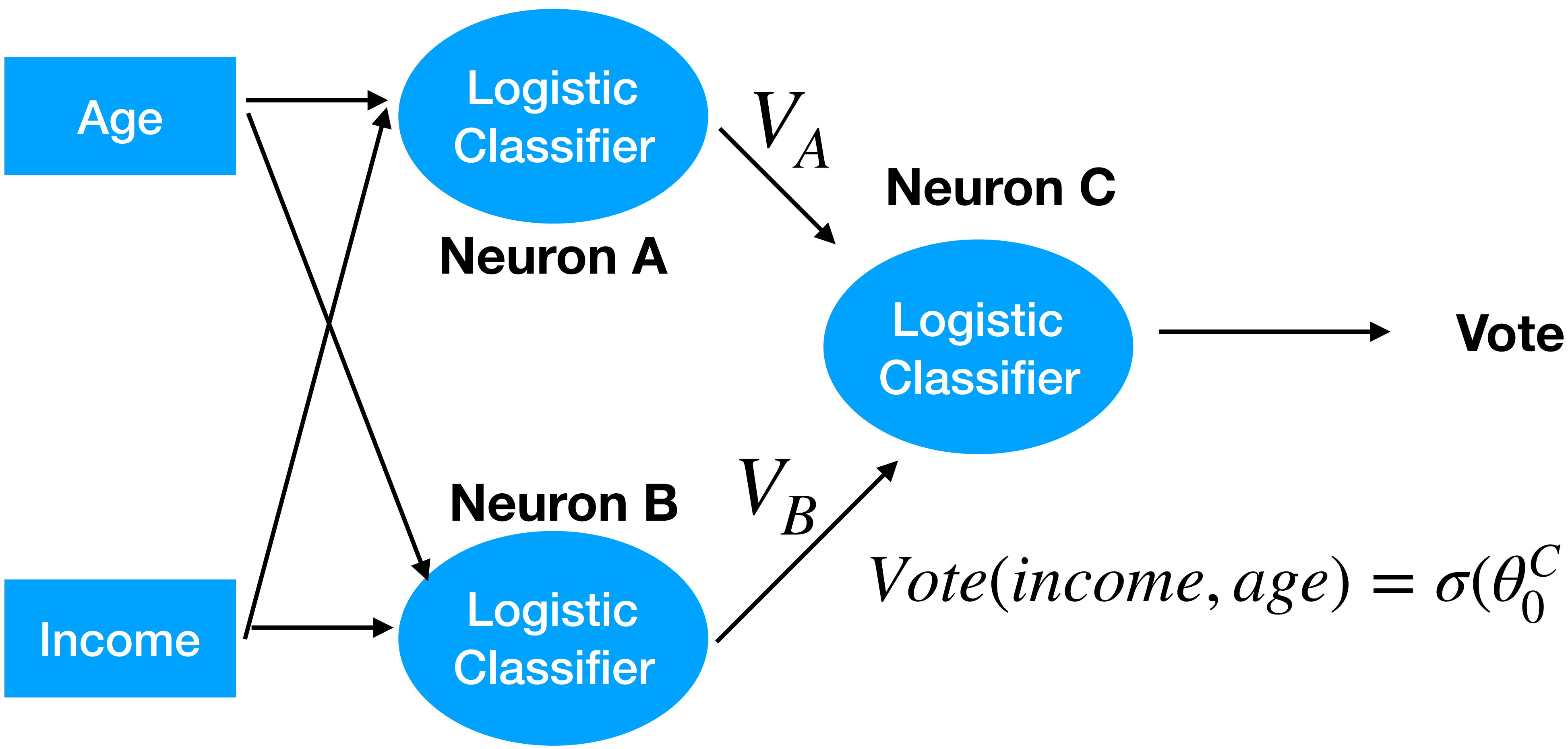
- A **Single Neuron** is just a simple **logistic classifier**
  - Like we saw <u>last week</u>
- Class boundary is a straight line
- Only 2 zones



- **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

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

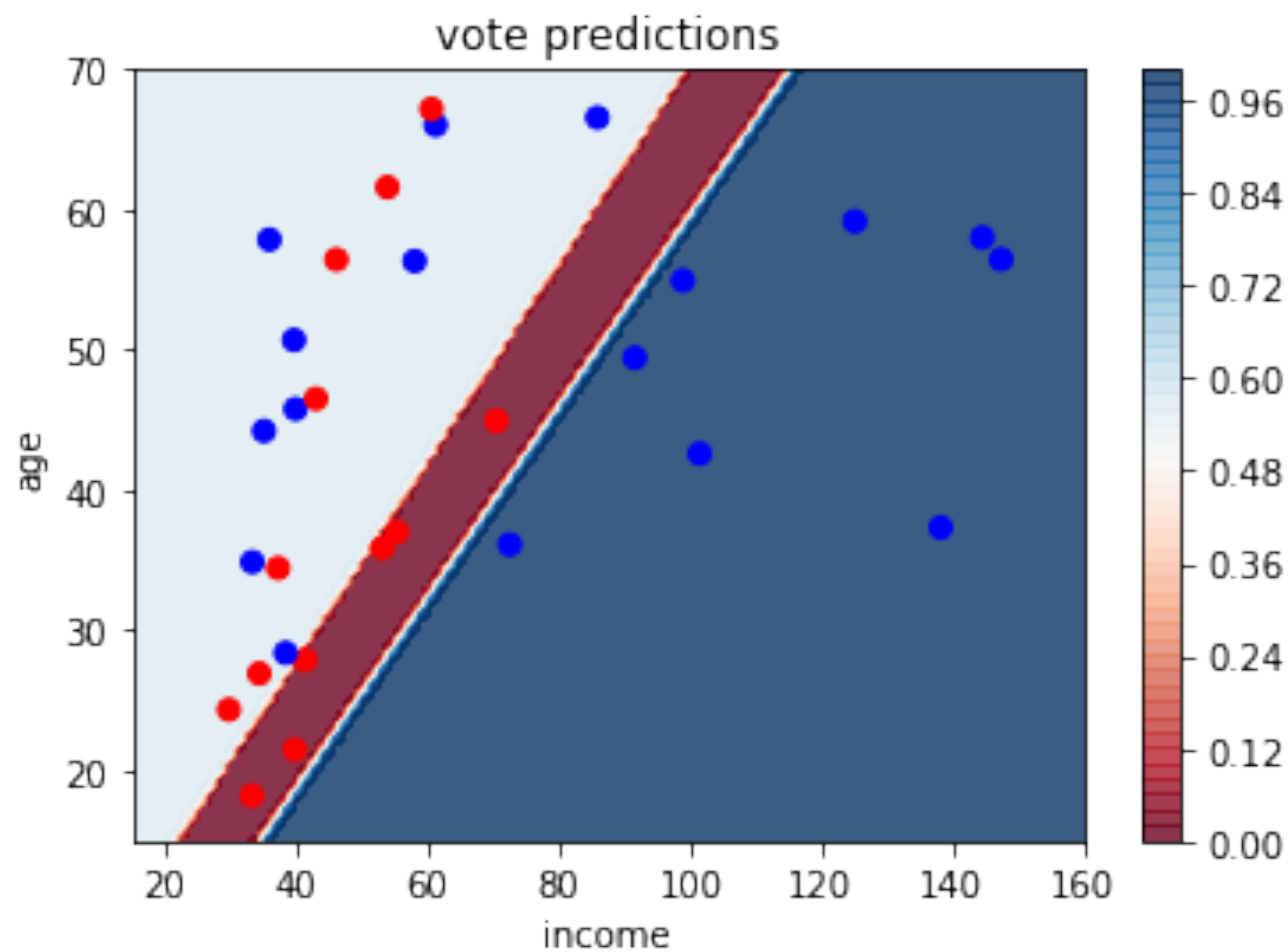


$$Vote(income, age) = \sigma(\theta_0^C + \theta_1^C \times V_A + \theta_2^C \times V_C)$$

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

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**

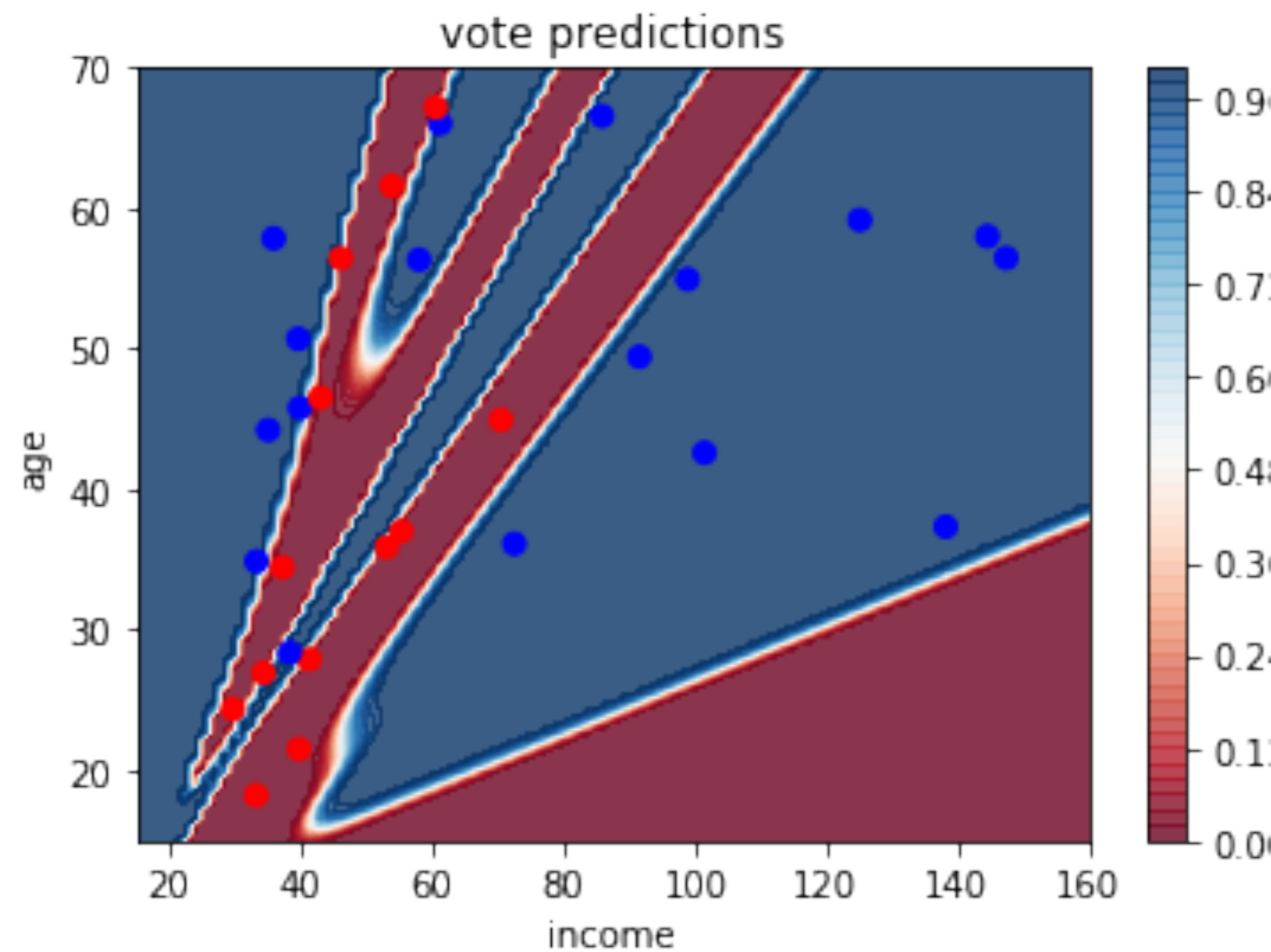
- **Darker blue area** means the classifier is **certain** people in this age/income zone will vote for the **Right-Wing party**

- **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



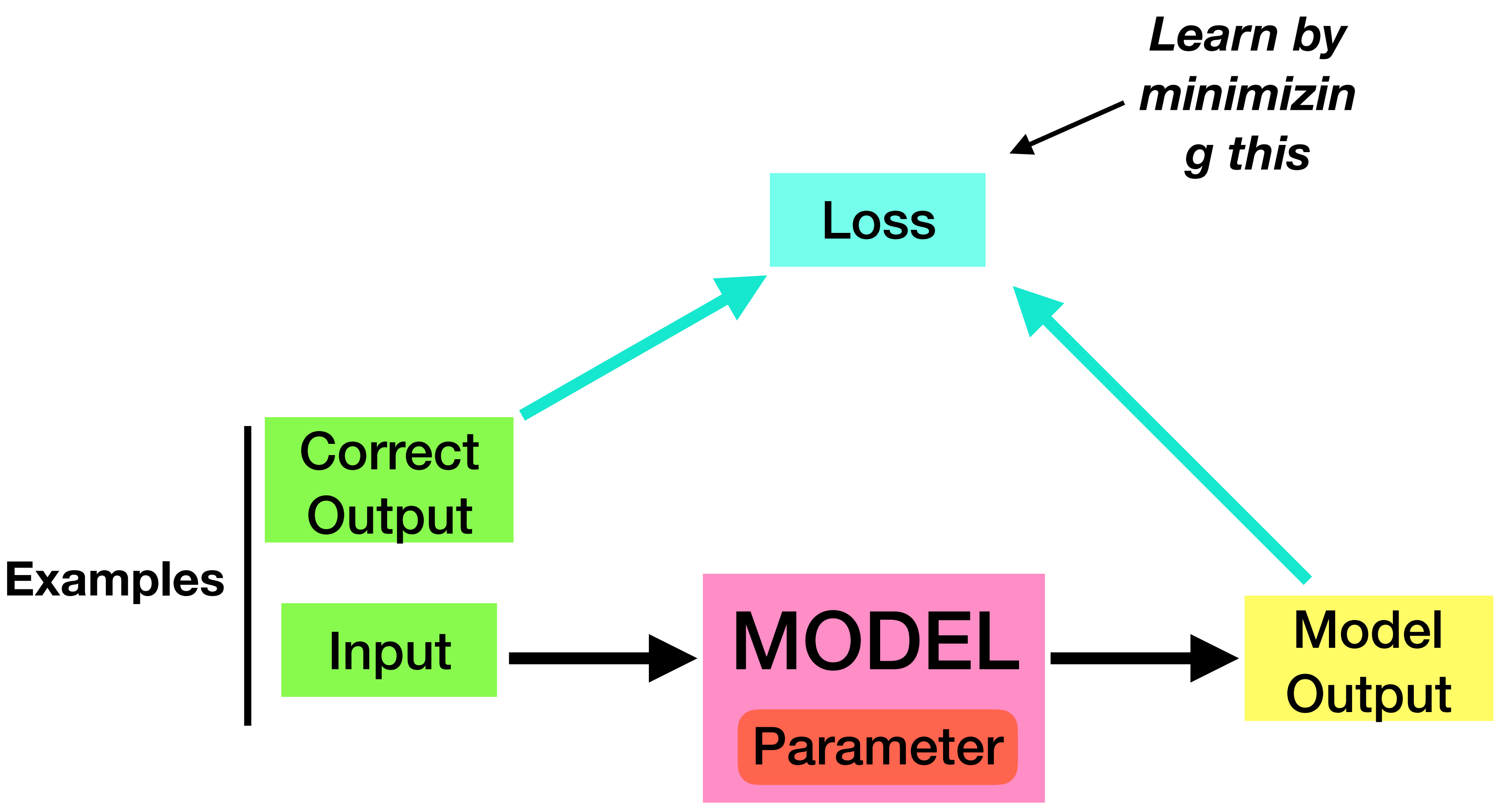
- **Darker red** area means the classifier is **certain** people in this age/income zone will vote for the **Left-Wing party**

- **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**

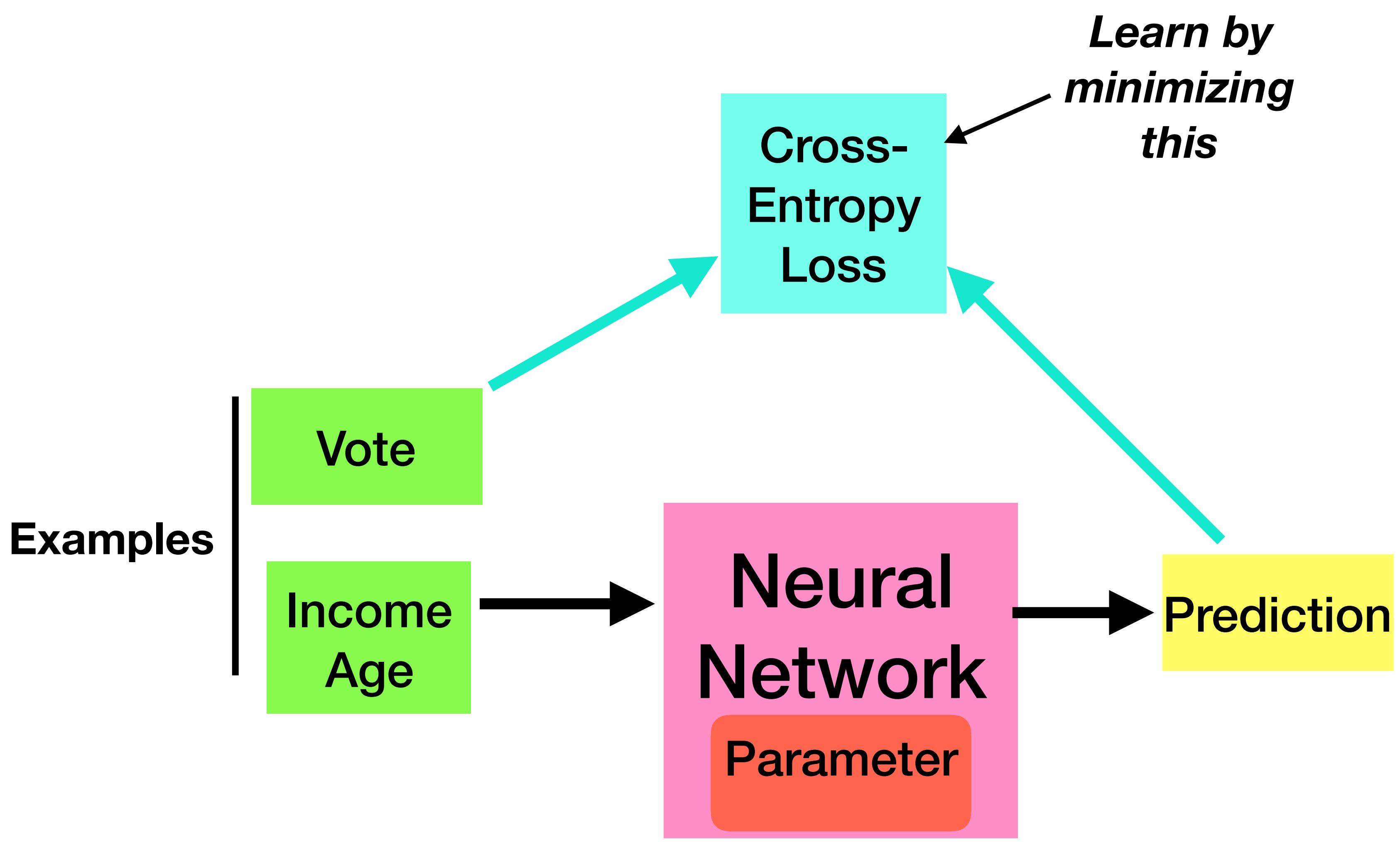
Supervised Learning

- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that compute how different the model output is from the correct output
  - ***Examples*** of input and correct output



Supervised Learning

- In supervised learning, we usually have:
    - A **MODEL**: a “parameterized” function that takes input and produce output
    - A ***Loss***: A function that compute how different the model output is from the correct output
    - ***Examples*** of input and correct output (cigarets smoked, age of death)



Making the connections

- What we have seen so far:

Increasing complexity

Linear regression

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

Logistic Classifier

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

$$prediction = \sigma(score)$$

Neural Network: Combination of logistic classifier

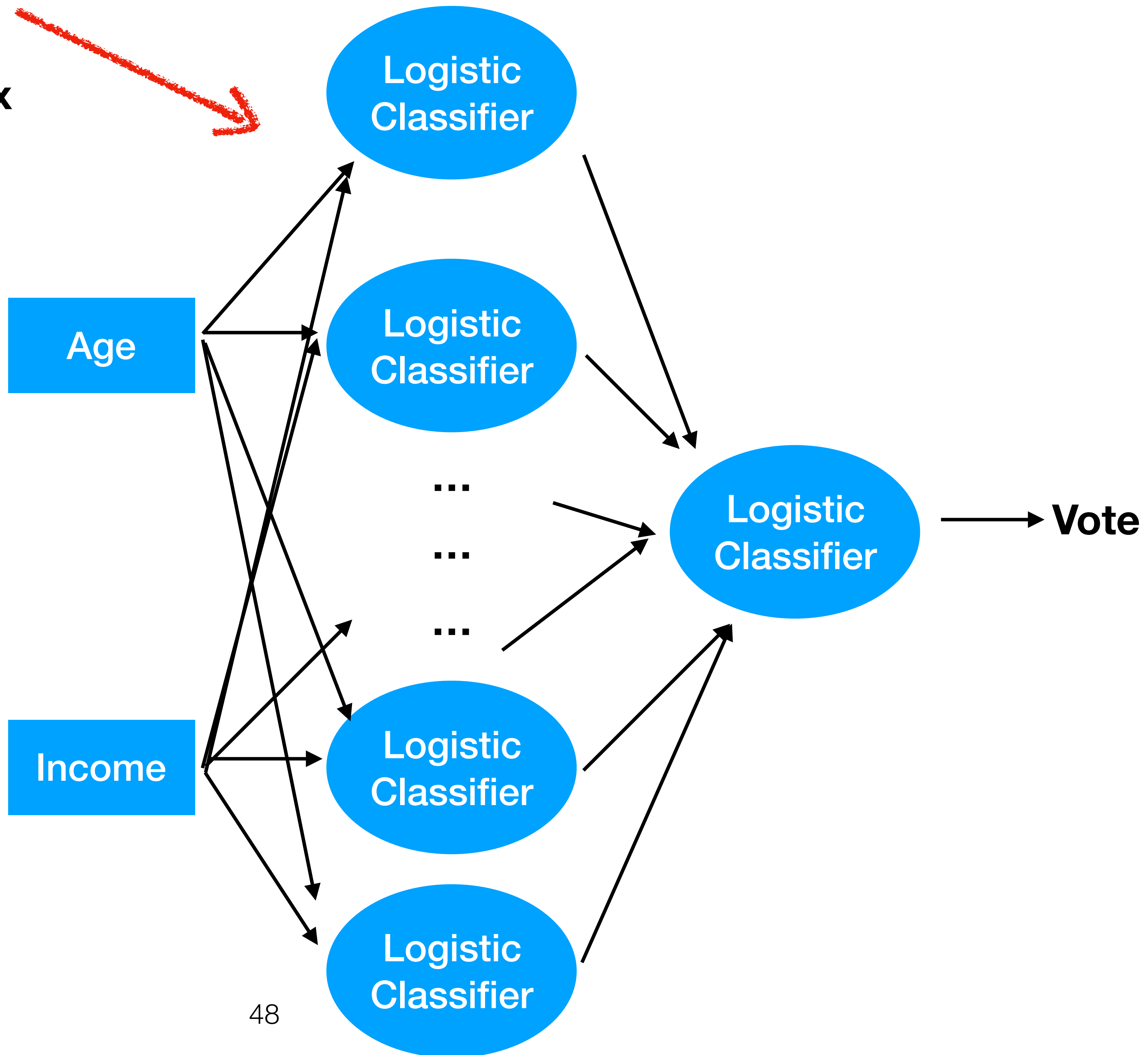
All trained by gradient descent on the loss

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

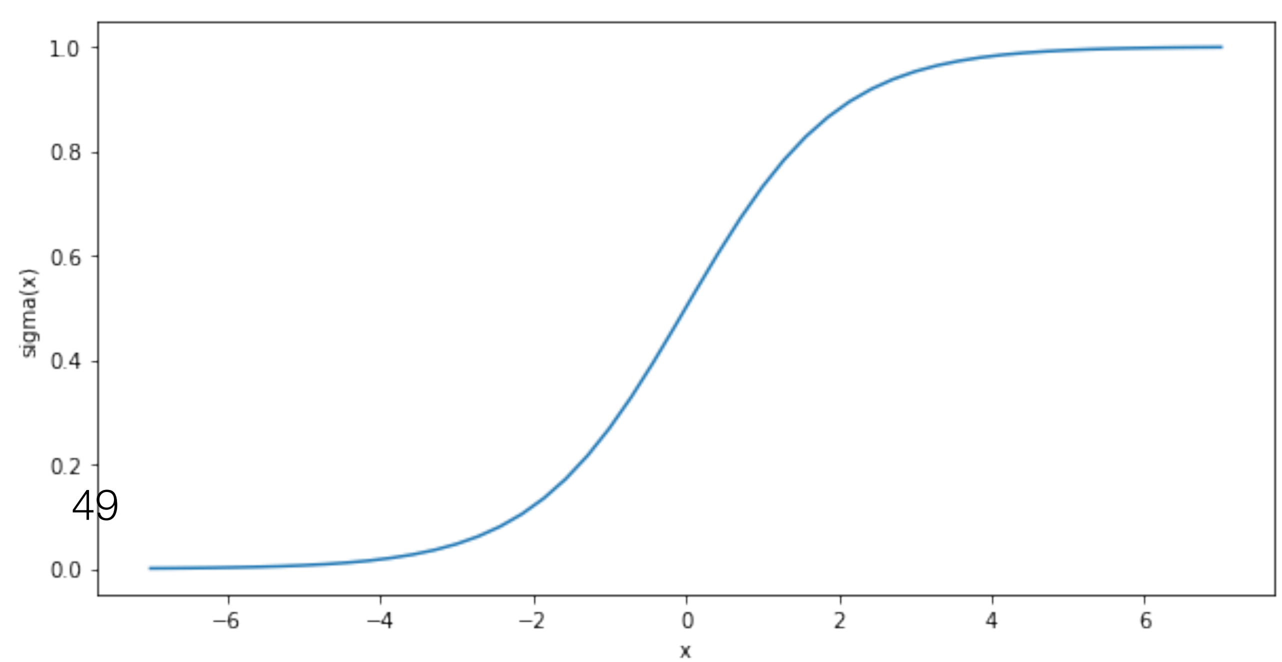


But note that this 2-layers architecture is usually not the most efficient

On “Activation” Functions

- So far, we have have applied the “sigmoid function” (a.k.a “logistic function”) to the output of the Neuron
    - Historically, the first to be used
    - It behaves similarly to Biological Neurons, as we have seen
    - But, not the most efficient in practice

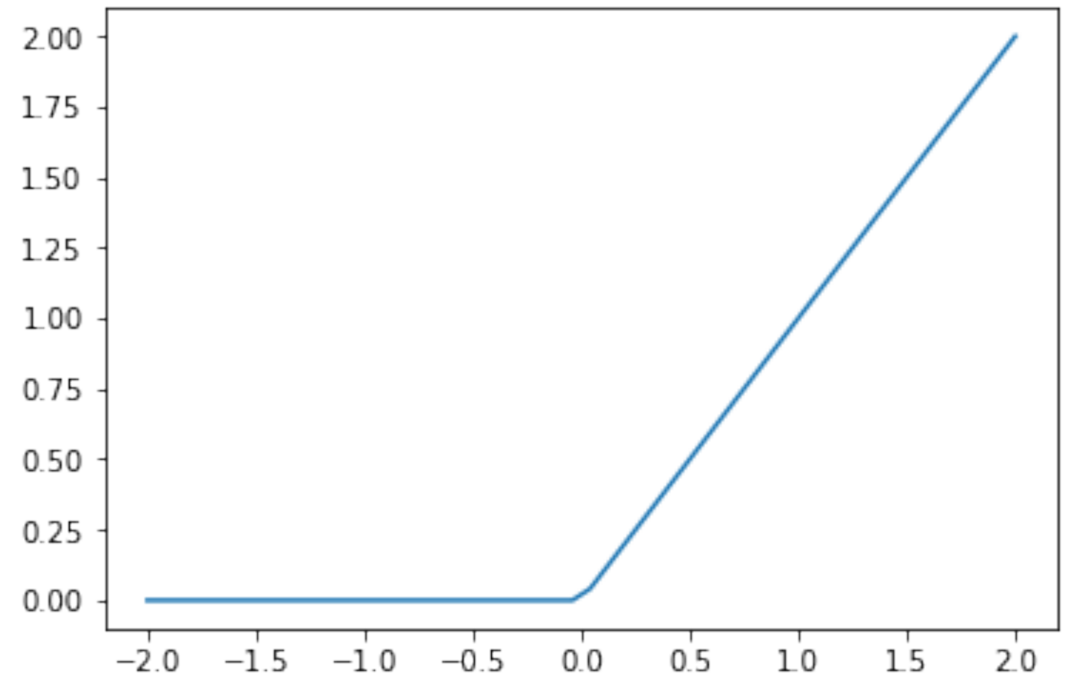
$$\sigma(x) = \frac{1}{1 + exp(-x)}$$



On “Activation” Functions

- There are many possible functions to choose from
- One that is simple and works very well: “Rectified Linear Unit”
- Very fast to compute
- Very efficient
- Less similar to biological Neurons

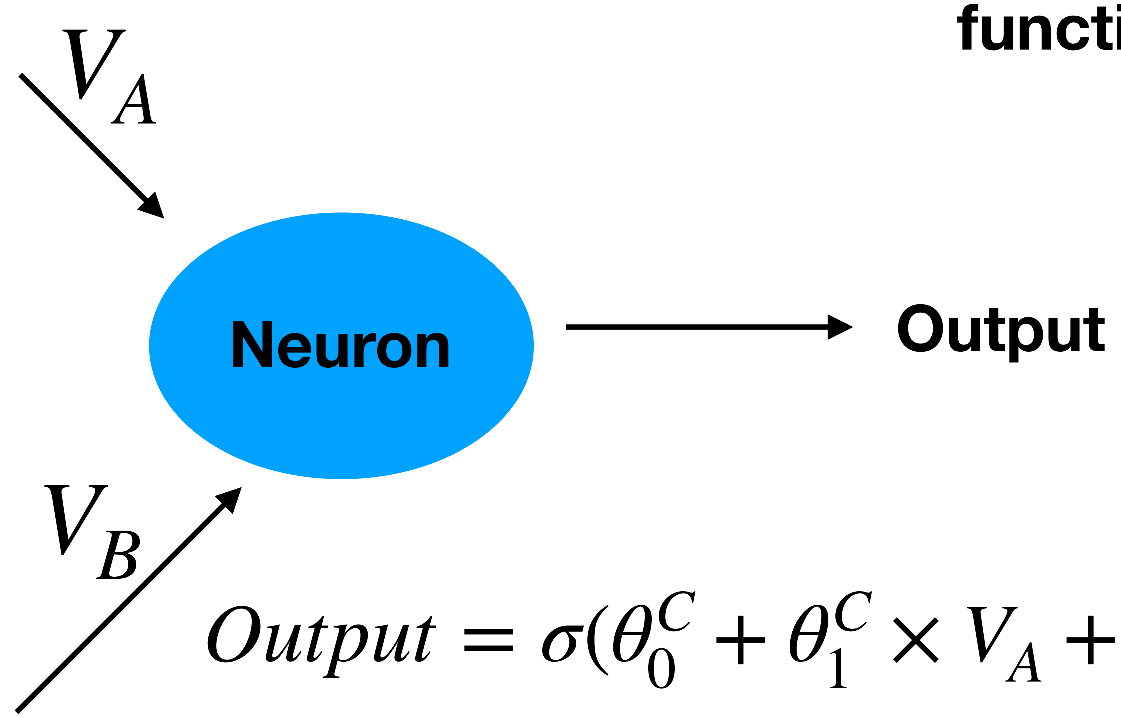
$$ReLU(x) = max(x,0)$$



Different activations

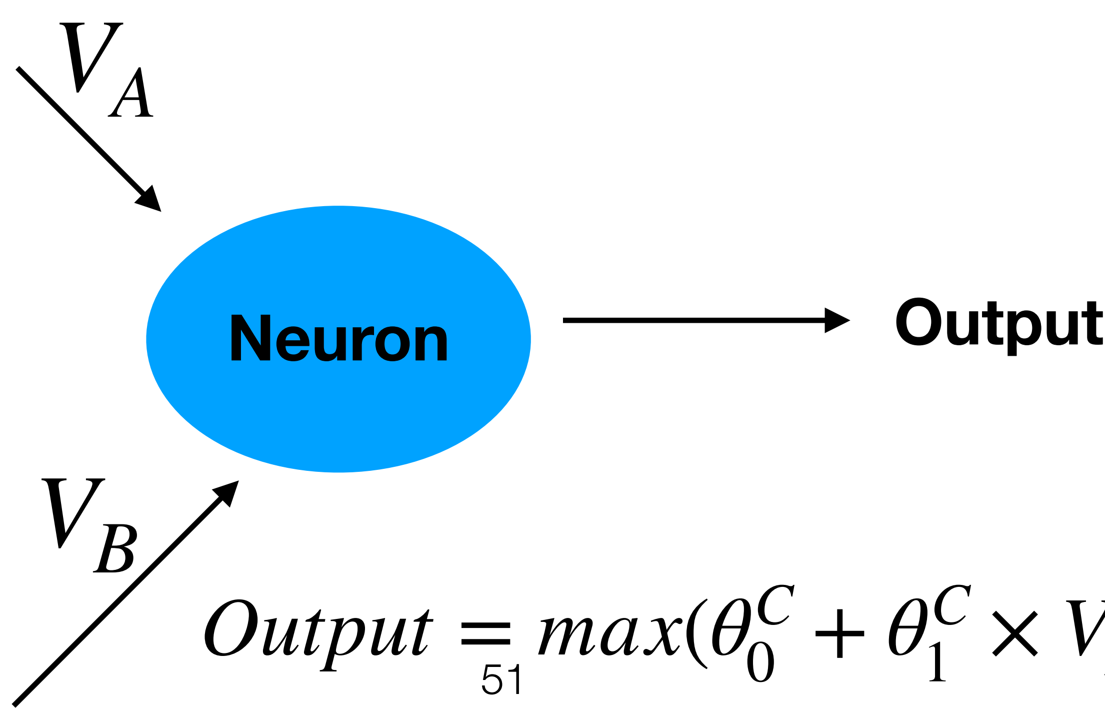
(Finding the best activation functions is an active area of research)

Neuron with sigmoid activation



$$Output = \sigma(\theta_0^C + \theta_1^C \times V_A + \theta_2^C \times V_C)$$

Neuron with ReLU activation



$$Output = max(\theta_0^C + \theta_1^C \times V_A + \theta_2^C \times V_C, 0)$$

Funny calculations: what is the power of the brain?

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

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

- Therefore, to <u>“simulate” one neuron computation</u>, we need about 7000 multiplications and 7000 additions

- It seems a neuron cannot activate more than 200 times per seconds

- To <u>simulate a neuron in real time</u>, we therefore need to compute about (7000 + 7000) x 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) x 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 200x10^9 / 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 x 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