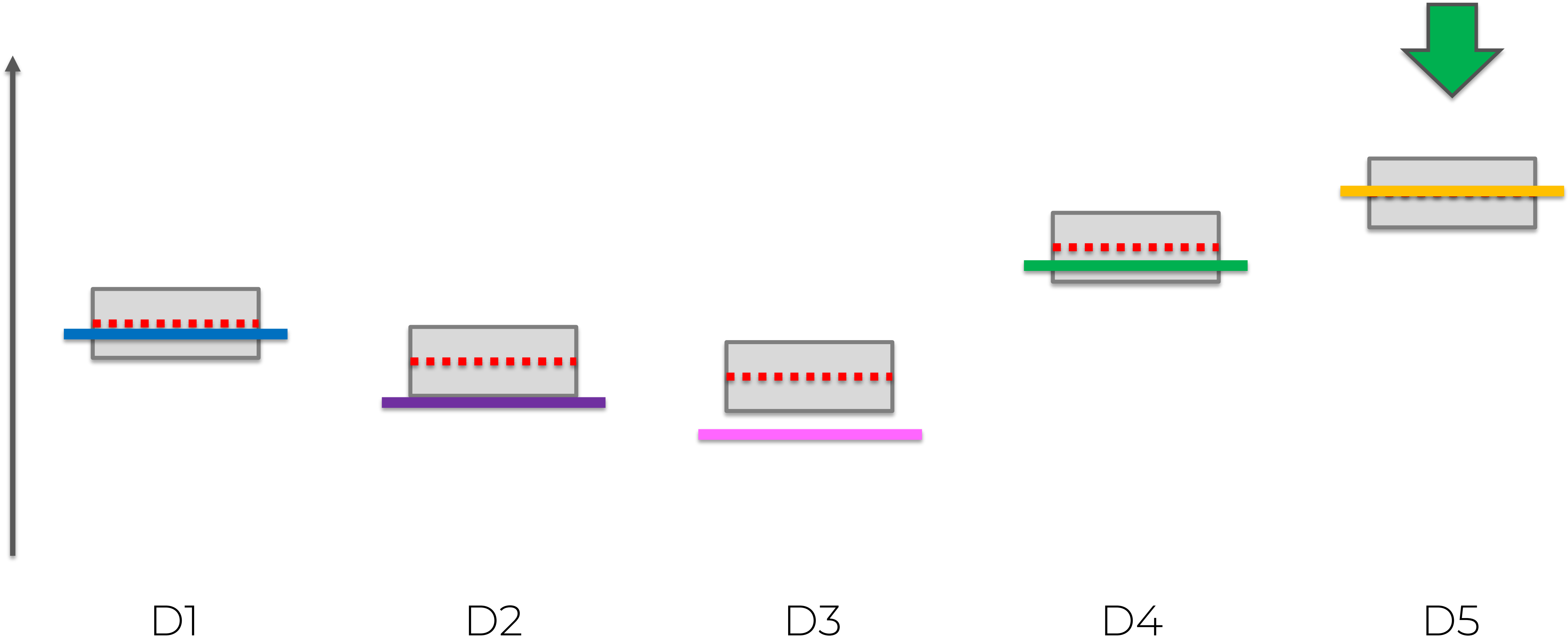
Upper Confidence Bound Algorithm



Thompson Sampling Algorithm Intuition

The Multi-Armed Bandit Problem

D1

D2

D3

D4

D5

The Multi-Armed Bandit Problem

- We have d arms. For example, arms are ads that we display to users each time they connect to a web page.
- Each time a user connects to this web page, that makes a round.
- At each round n, we choose one ad to display to the user.
- At each round n, ad i gives reward $r_i(n) \in \{0, 1\}$: $r_i(n) = 1$ if the user clicked on the ad i, 0 if the user didn't.
- Our goal is to maximize the total reward we get over many rounds.

Bayesian Inference

- Ad i gets rewards $\boldsymbol{y}$ from Bernoulli distribution $p(\boldsymbol{y}|\theta_i) \sim \mathcal{B}(\theta_i)$.
- θ_i is unknown but we set its uncertainty by assuming it has a uniform distribution $p(\theta_i) \sim \mathcal{U}([0,1])$, which is the prior distribution.
- Bayes Rule: we approach θ_i by the posterior distribution

$$\underbrace{p(\theta_i|\mathbf{y})}_{\text{posterior distribution}} = \frac{p(\mathbf{y}|\theta_i)p(\theta_i)}{\int p(\mathbf{y}|\theta_i)p(\theta_i)d\theta_i} \propto \underbrace{p(\mathbf{y}|\theta_i)}_{\text{likelihood function}} \times \underbrace{p(\theta_i)}_{\text{prior distribution}}$$

- We get $p(\theta_i|\mathbf{y}) \sim \beta(\text{number of successes} + 1, \text{number of failures} + 1)$
- At each round n we take a random draw $\theta_i(n)$ from this posterior distribution $p(\theta_i|\boldsymbol{y})$, for each ad i.
- At each round n we select the ad i that has the highest $\theta_i(n)$.

Thompson Sampling Algorithm

Step 1. At each round n, we consider two numbers for each ad i:

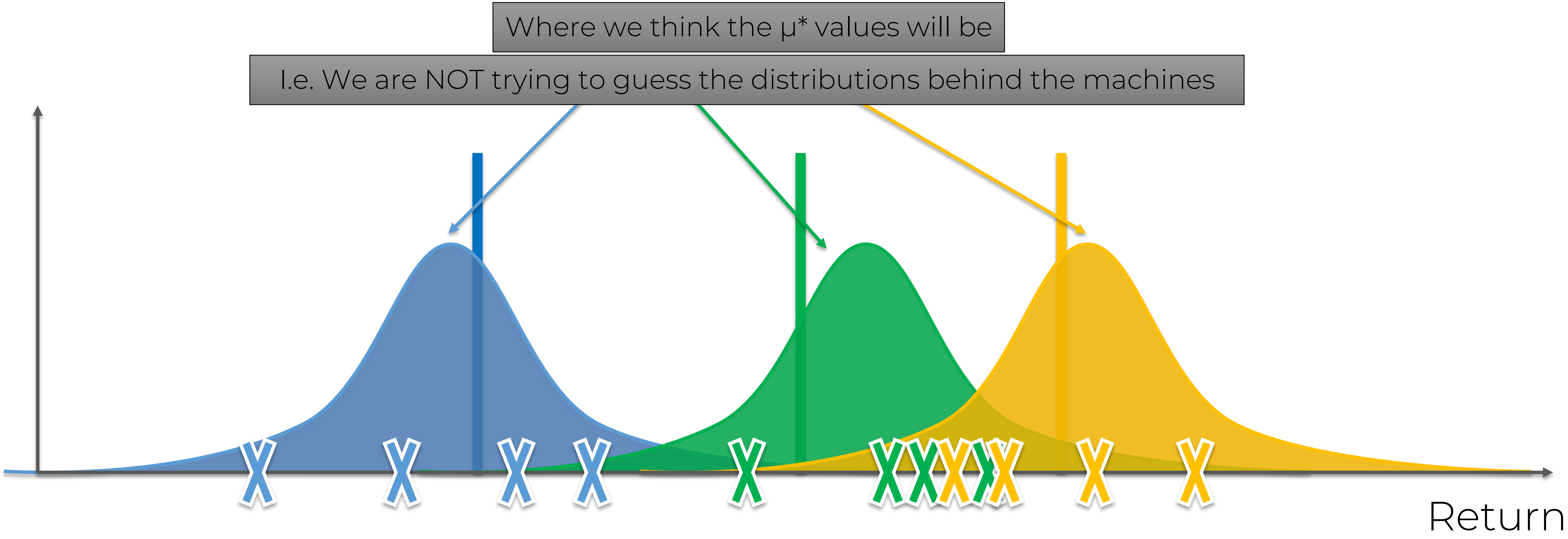
- $N_i^1(n)$ - the number of times the ad i got reward 1 up to round n,
- $N_i^0(n)$ - the number of times the ad i got reward 0 up to round n.

Step 2. For each ad i, we take a random draw from the distribution below:

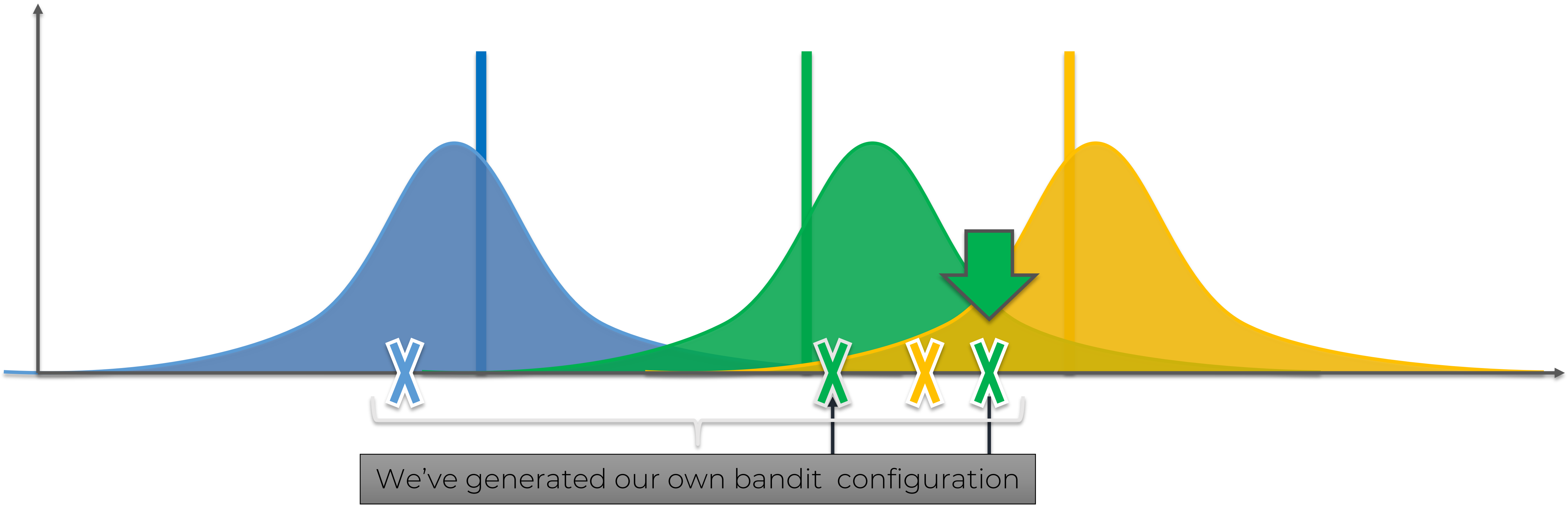
$$\theta_i(n) = \beta(N_i^1(n) + 1, N_i^0(n) + 1)$$

Step 3. We select the ad that has the highest $\theta_i(n)$.

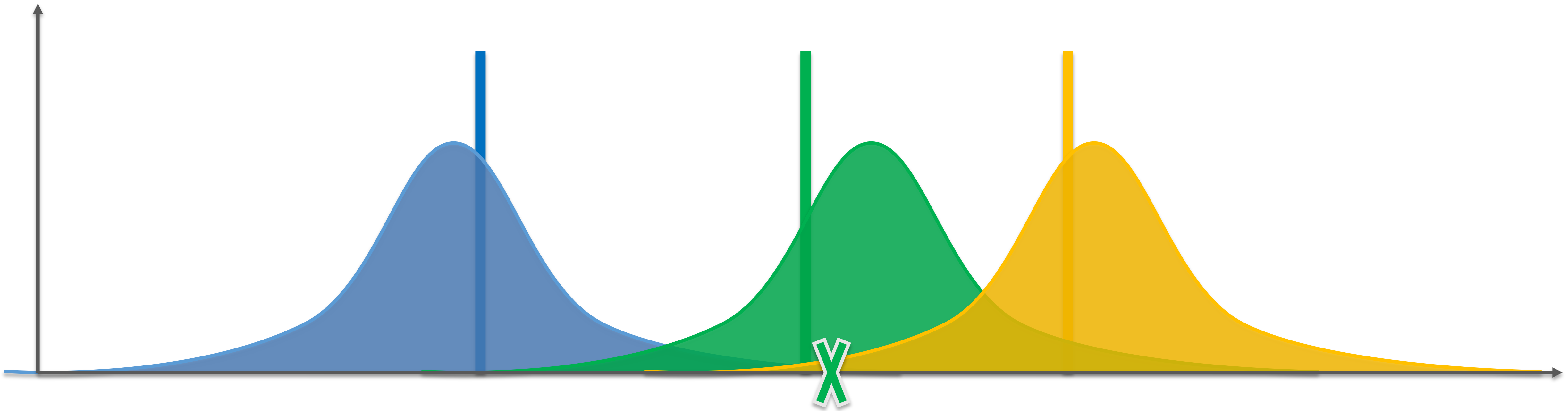
Thompson Sampling Algorithm



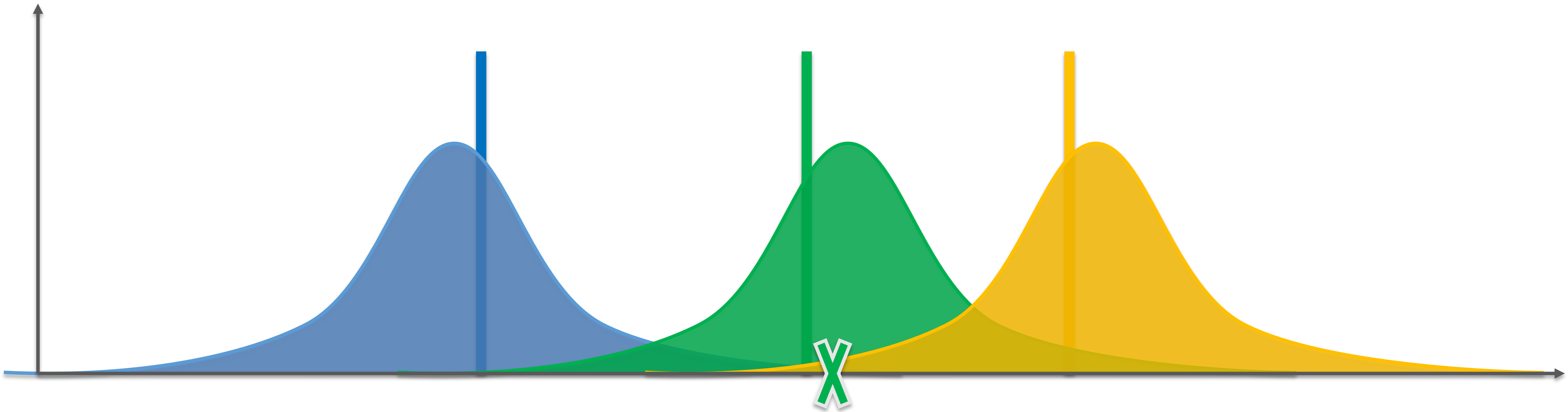
Thompson Sampling Algorithm



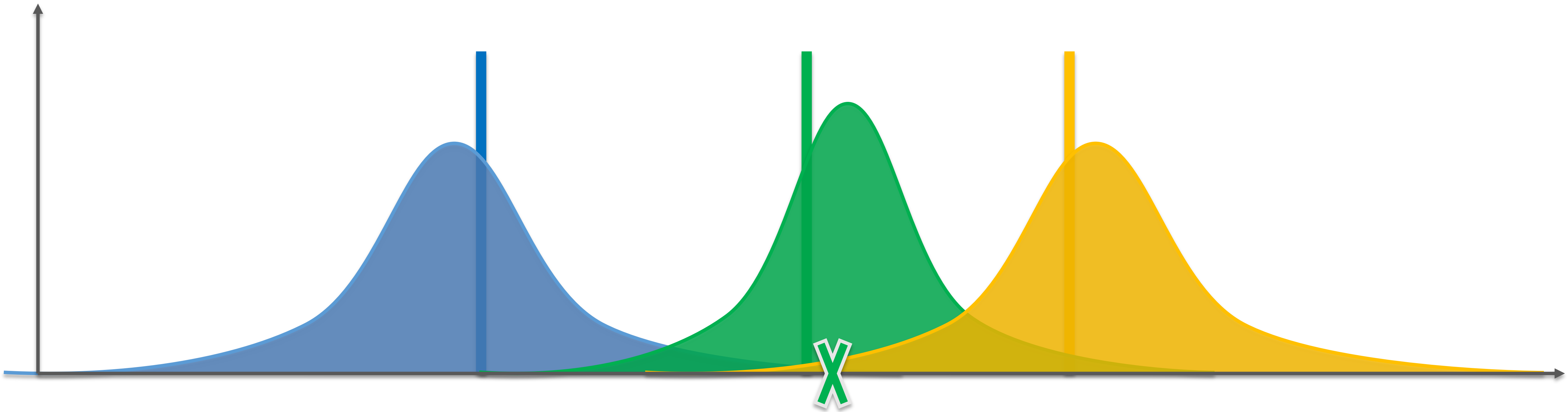
Thompson Sampling Algorithm



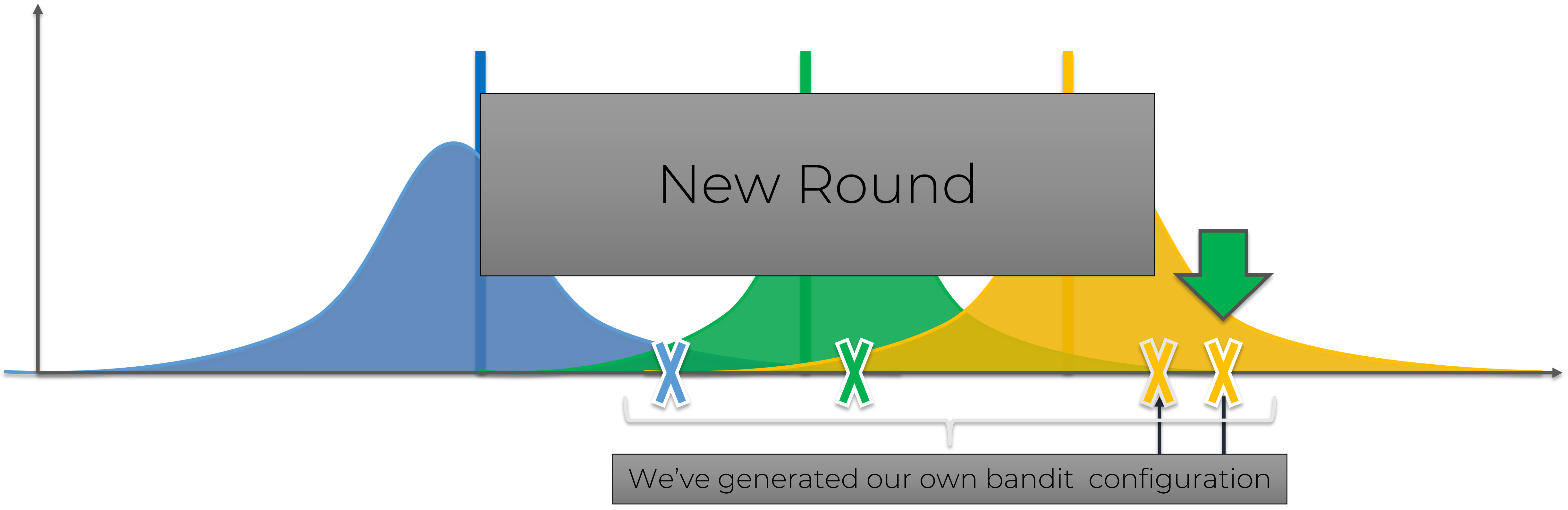
Thompson Sampling Algorithm



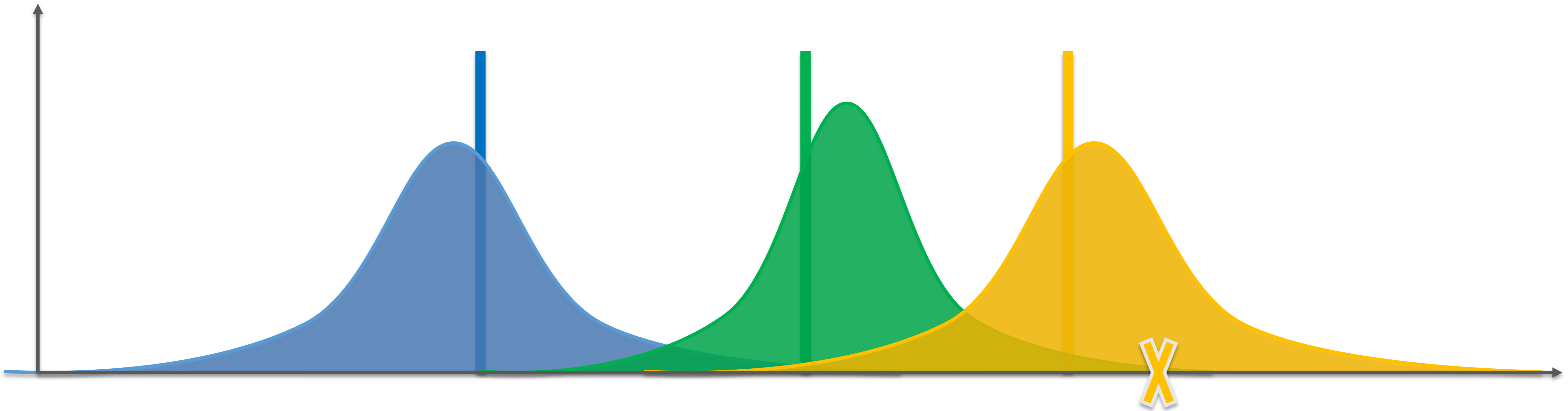
Thompson Sampling Algorithm



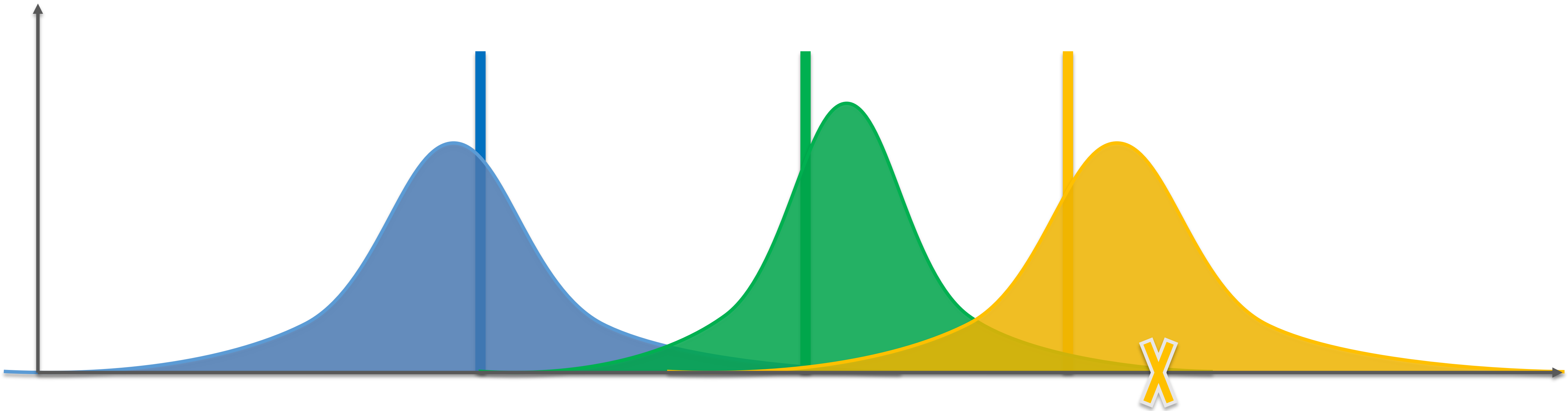
Thompson Sampling Algorithm



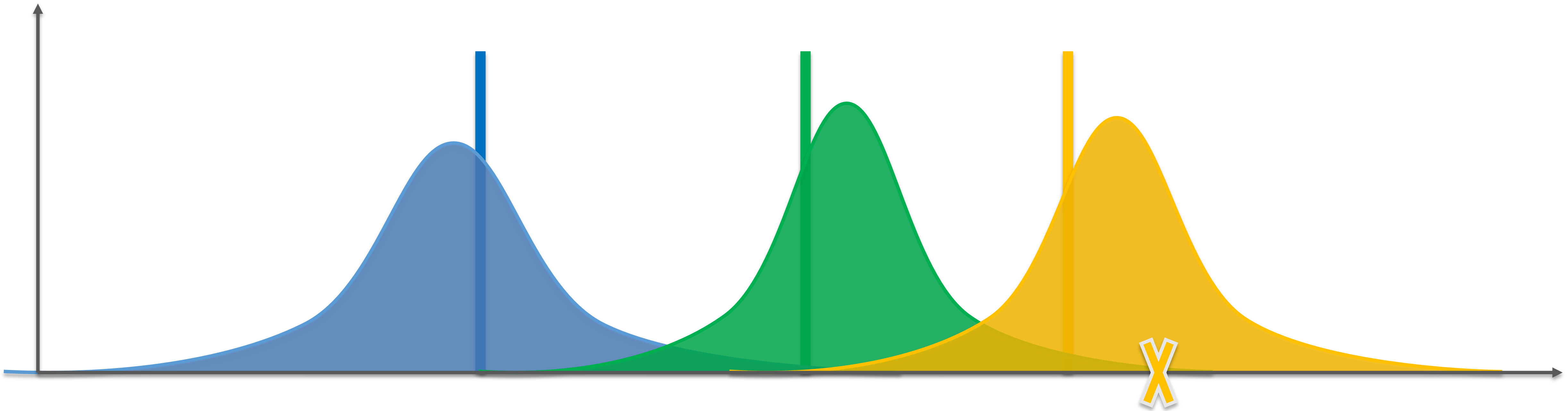
Thompson Sampling Algorithm



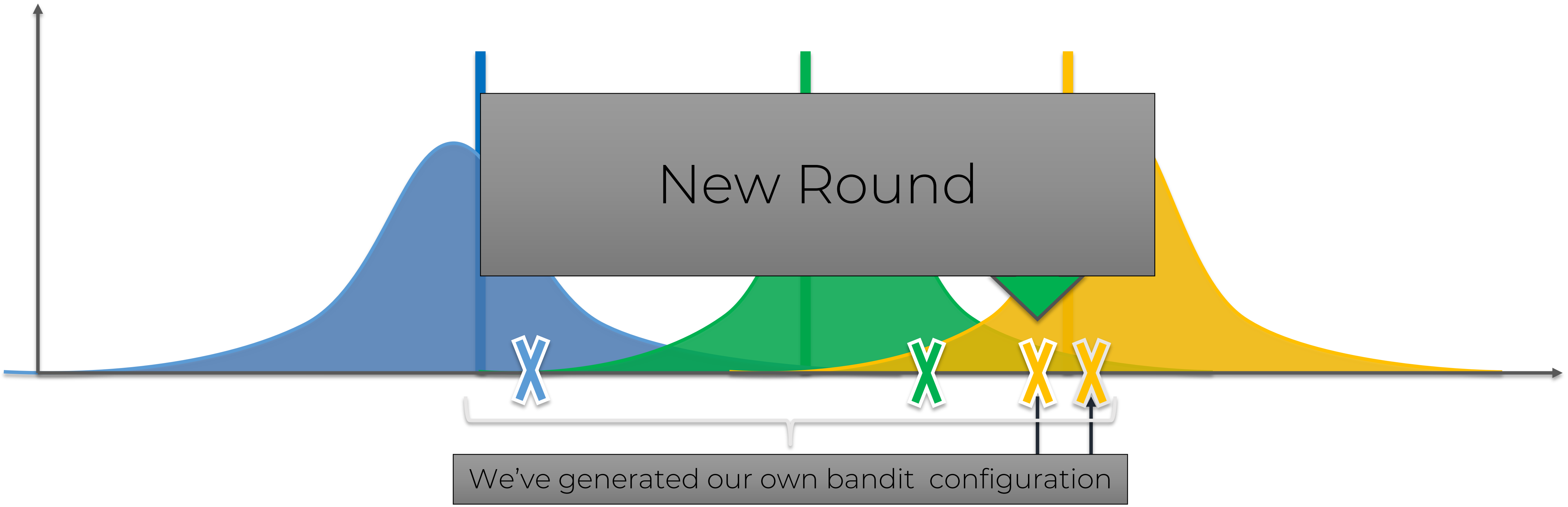
Thompson Sampling Algorithm



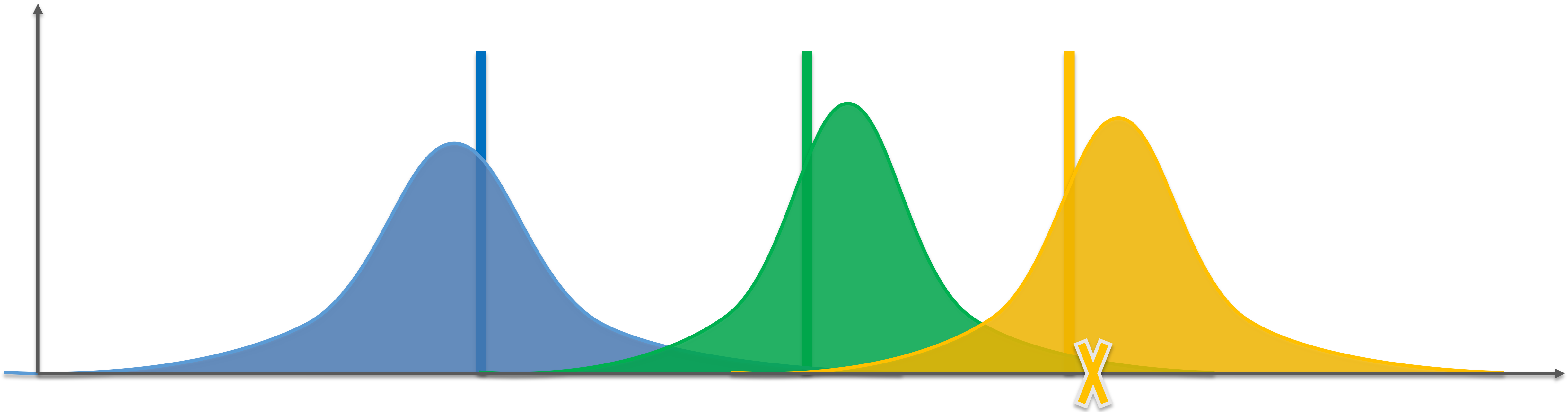
Thompson Sampling Algorithm



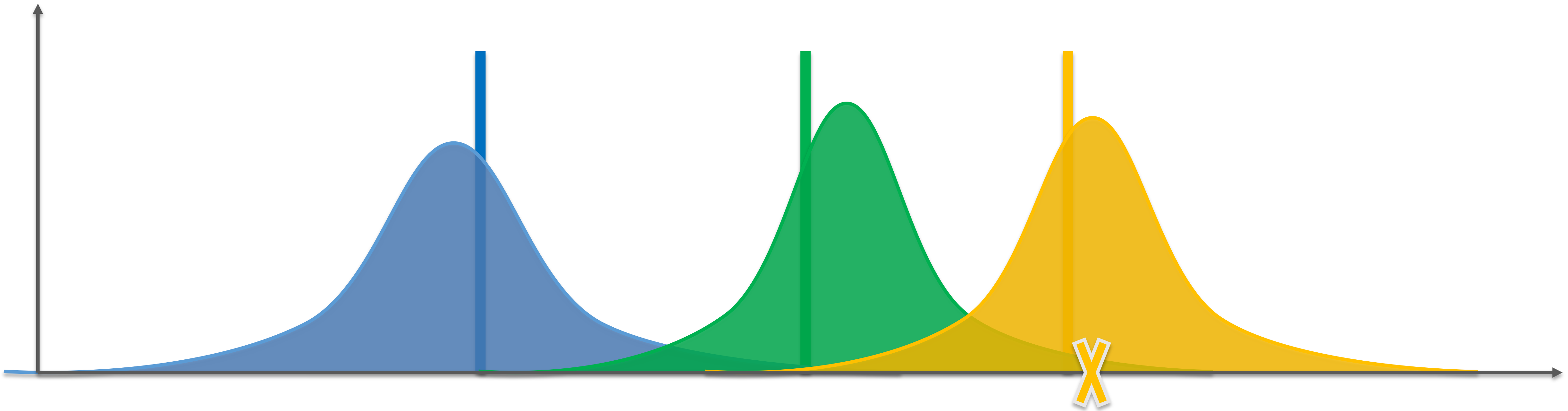
Thompson Sampling Algorithm



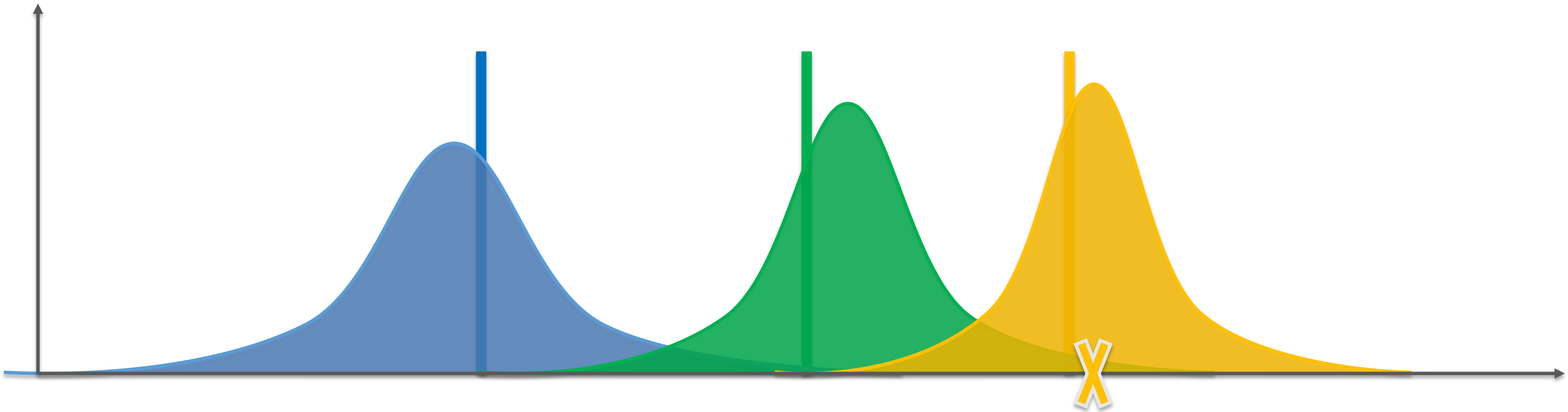
Thompson Sampling Algorithm



Thompson Sampling Algorithm



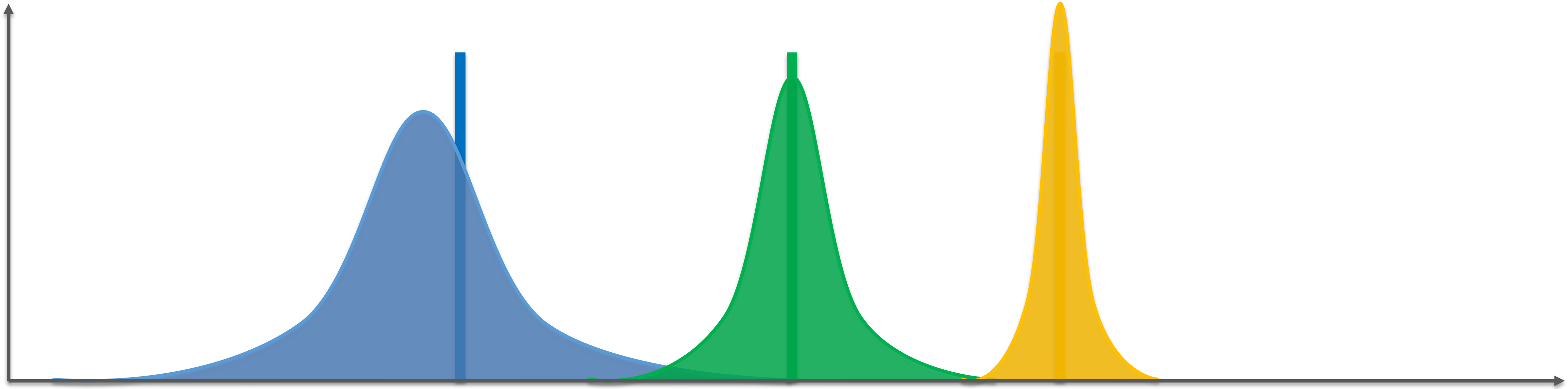
Thompson Sampling Algorithm



Thompson Sampling Algorithm

And so on…

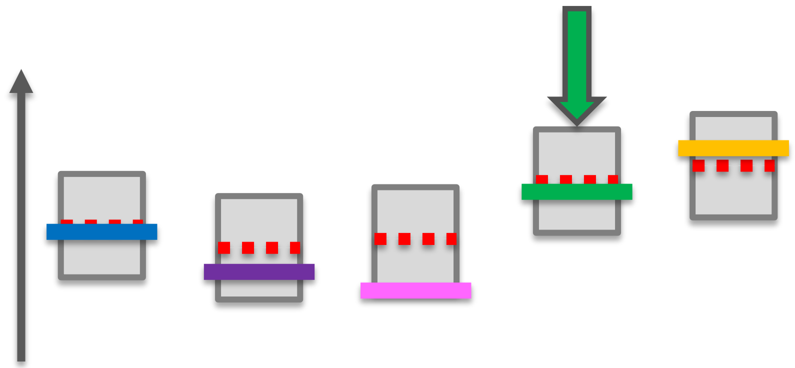
Thompson Sampling Algorithm



UCB vs Thompson Sampling

Thompson Sampling Algorithm

UCB



- Deterministic
- Requires update at every round

Thompson Sampling

- Probabilistic
- Can accommodate delayed feedback
- Better empirical evidence

Natural Language Processing (NLP)

Plan of Attack

Plan of Attack

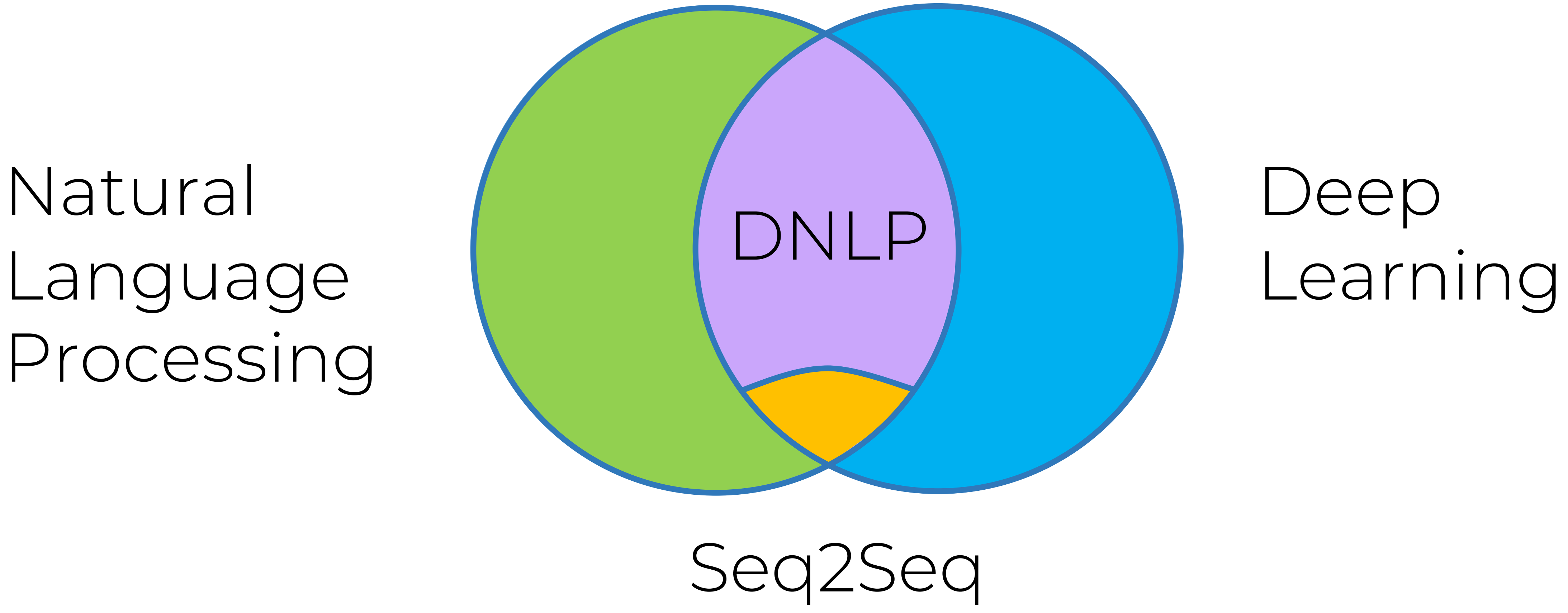
Here's what we will learn:

- Types of Natural Language Processing
- Classical vs Deep Learning Models
- End-to-end Deep Learning Models
- Bag-Of-Words

- Note: Seq2Seq and Chatbots are outside the scope of this course

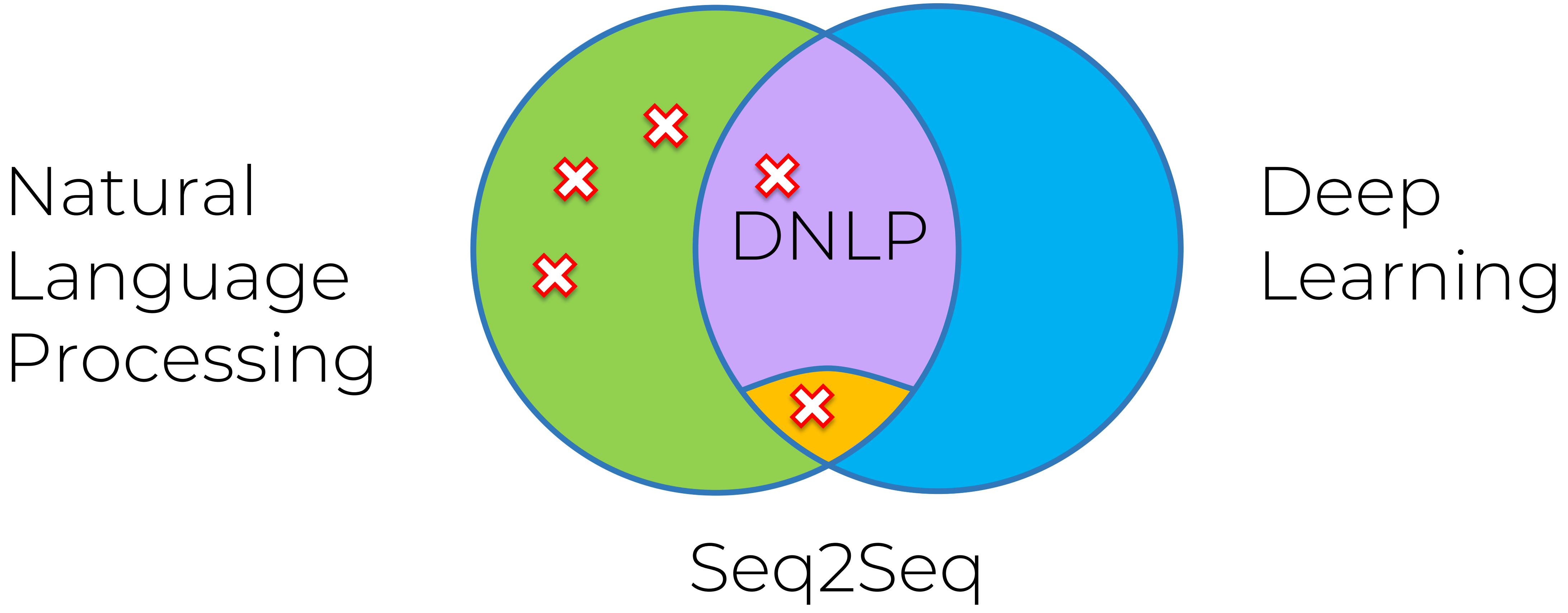
Types of NLP

Types of NLP



Classical vs Deep Learning Models

Classical vs Deep Learning Models



Classical vs Deep Learning Models

Some examples:

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)
3. Bag-of-words model (Classification)
4. CNN for text Recognition (Classification)

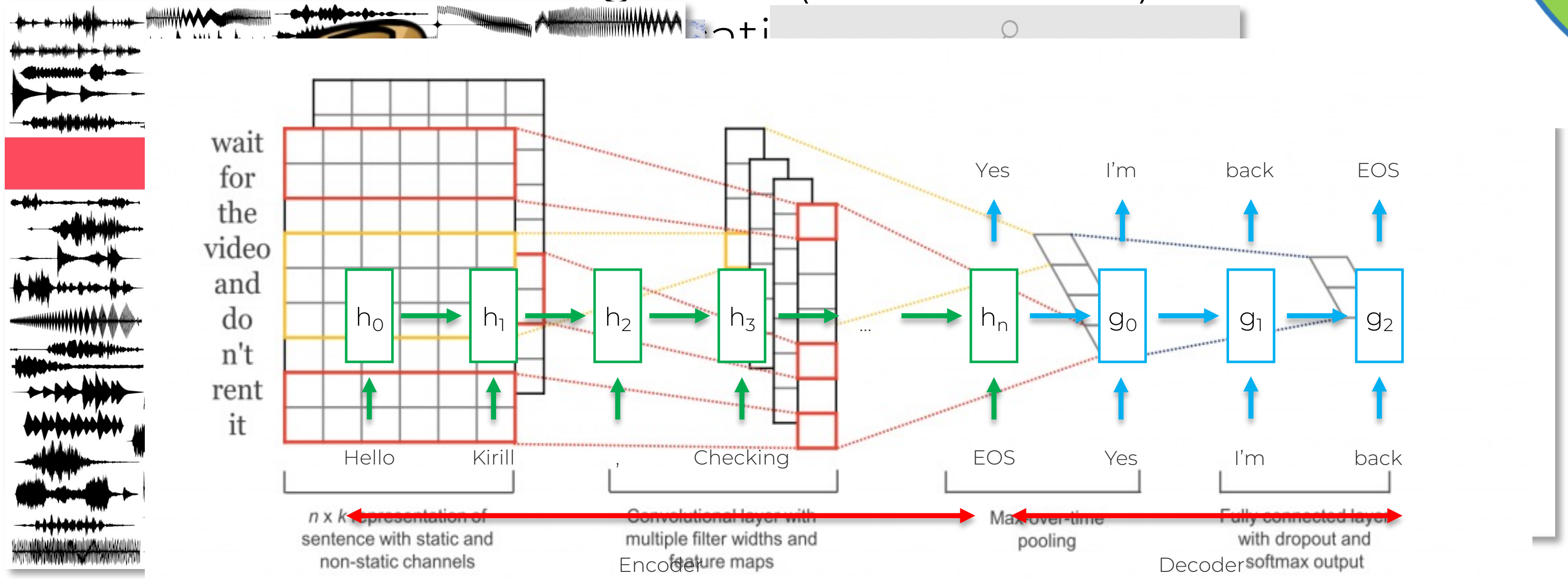
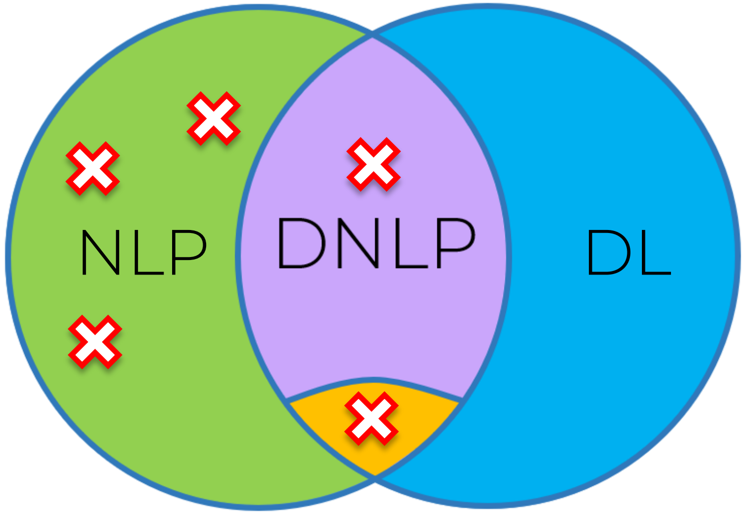


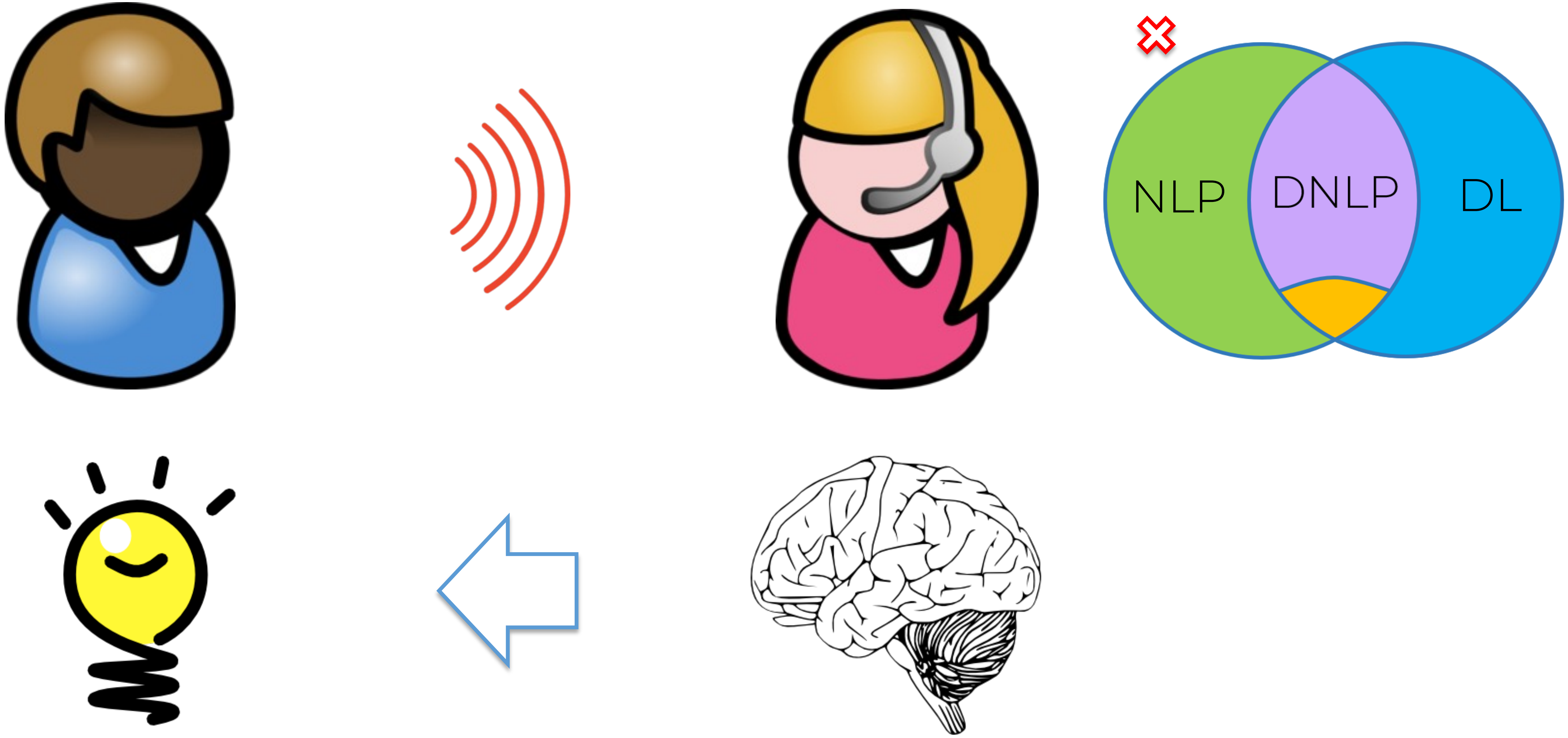
Image Source: www.wildml.com



Seq2Seq

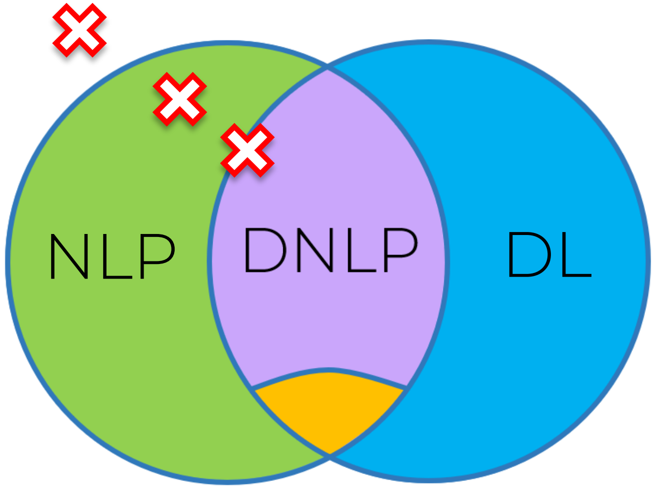
End-to-end Deep Learning Models

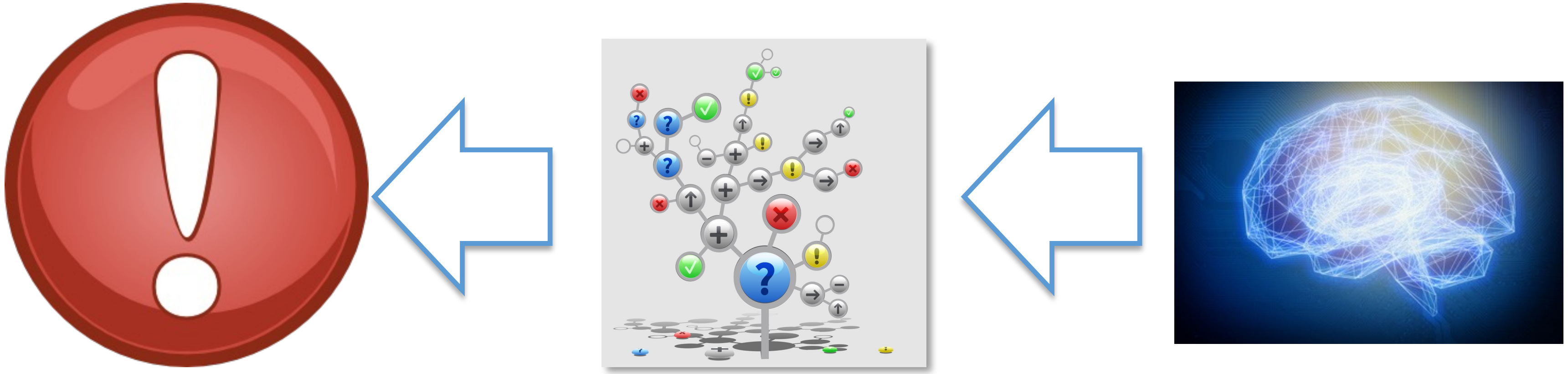
End-to-end Deep Learning Models



End-to-end Deep Learning Models

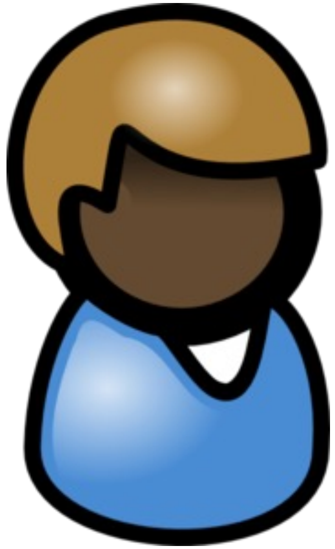
End-to-end Deep Learning Models

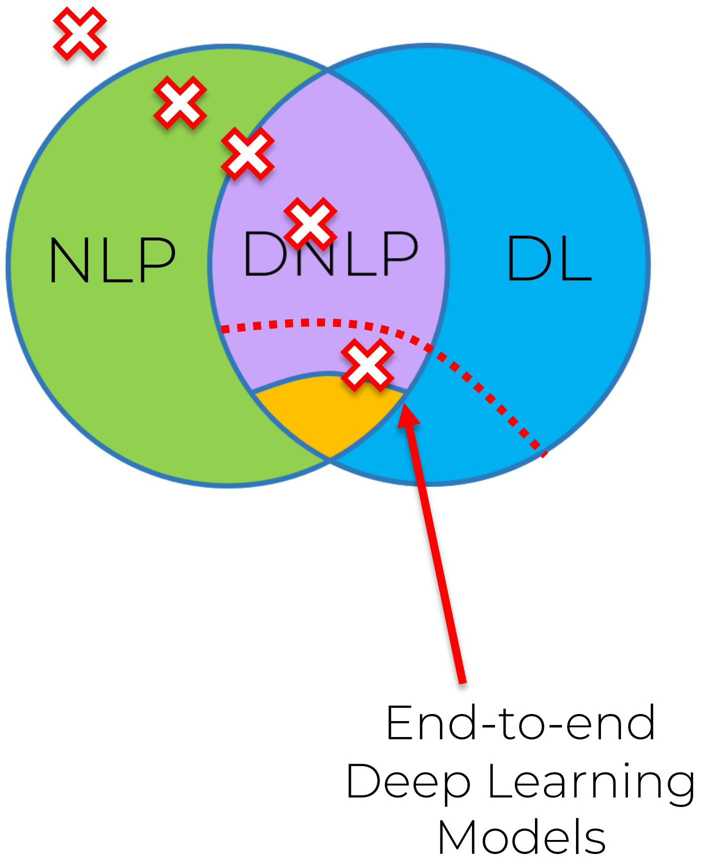


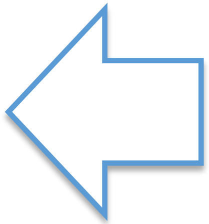


End-to-end Deep Learning Models

End-to-end Deep Learning Models







Bag-Of-Words

Telstra 6:19 pm 53%

Catch up? Inbox Business

Kirill Eremenko 6:18 pm

to me

Hello Kirill,

Checking if you are back to Oz. Let me know if you are around and keen to sync on how things are going. I defo could use some of your creative thinking to help with mine :)

Cheers,
V

Yes, I'm around. | I'm back! | Sorry, I'm not.

Reply Forward

Bag-Of-Words

Yes / No

Bag-Of-Words

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... , 0]

20,000 elements long

if

badminton

table

171,476 words

The Second Edition of the 20-volume Oxford English Dictionary contains full entries for **171,476 words** in current use, and **47,156** obsolete words. To this may be added around **9,500** derivative words included as subentries.

How many words are there in the English language?
https://en.oxforddictionaries.com/.../how-many-words-are-there-in-the-english-language

About this result Feedback

People also ask

How many words in the English language does the average person know?

Most adult native test-takers range from **20,000–35,000 words**. Average native test-takers of age 8 already know **10,000 words**. Average native test-takers of age 4 already know **5,000 words**. Adult native test-takers learn almost 1 new word a day until middle age. May 29, 2013

Lexical facts - The Economist
https://www.economist.com/blogs/johnson/2013/05/vocabulary-size

We have seen that the Oxford English Dictionary contains **171,476 words** in current use, whereas a vocabulary of just **3000 words** provides coverage for around 95% of common texts. If you do the math, that's 1.75% of the total number of words in use! Mar 14, 2013

How many words in the english language ? How many do i need to ...
https://www.lingholic.com/how-many-words-do-i-need-to-know-the-955-rule-in-langua...

Bag-Of-Words

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, … , 0]

20,000 elements long

SOS

EOS

Special Words

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, … , 0]

20,000 elements long

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3]

20,000 elements long

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3] → **Yes / No ?**

20,000 elements long

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3] → Yes / No ?

20,000 elements long

Training Data:

Hey mate, have you read about Hinton's capsule networks?	→	No
Did you like that recipe I sent you last week?	→	Yes
Hi Kirill, are you coming to dinner tonight?	→	Yes
Dear Kirill, would you like to service your car with us again?	→	No
Are you coming to Australia in December?	→	Yes
…	→	…

Bag-Of-Words

Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

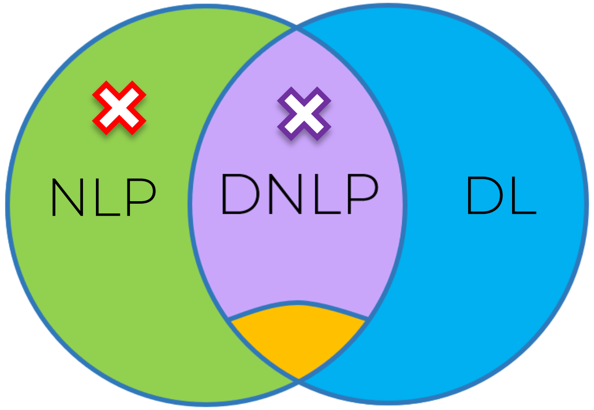
[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3] ⟶ **Yes / No ?**

20,000 elements long

Training Data:

[1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, … , 2]	⟶	**No**
[1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 0]	⟶	**Yes**
[1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, … , 1]	⟶	**Yes**
[1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, … , 1]	⟶	**No**
[1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, … , 1]	⟶	**Yes**
…	⟶	•••

Bag-Of-Words



Hello Kirill, Checking if you are back to Oz. Let me know if you are around … Cheers, V

[1, 1, 0, 0, 1, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 0, 1, 0, 0, … , 3] → Yes / No ?

20,000 elements long

Training Data:

[1, 1, 0, 0, 0 … 0, 1, 0, 1, 0, 1,
[1, 1, 0, 0, 0 … 0, 2, 0, 0, 0
[1, 1, 0, 0, 0 … 0, 1, 0, 0, 0,
[1, 1, 0, 0, 0 … 0, 1, 1, 0, 1, 0
[1, 1, 0, 0, 0 … 0, 1, 0, 0, 1,
…

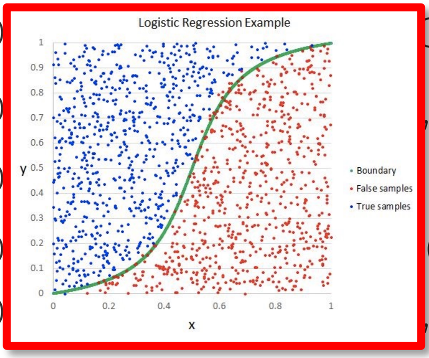
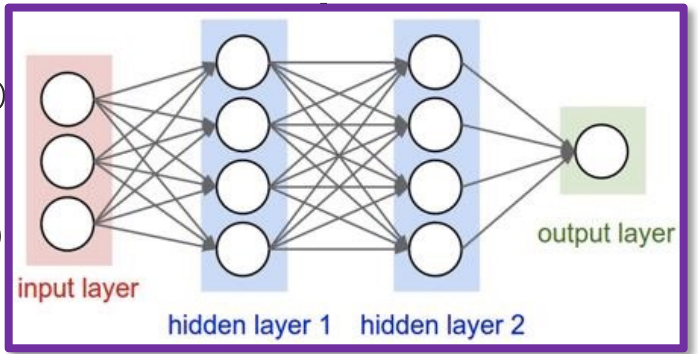


Image Source: www.helloacm.com



No
Yes
Yes
No
Yes
…

Artificial Neural Network Intuition

A Clip From
The Today Show, 1994

What is Deep Learning?

What is Deep Learning?

What is Deep Learning?

1956

1980

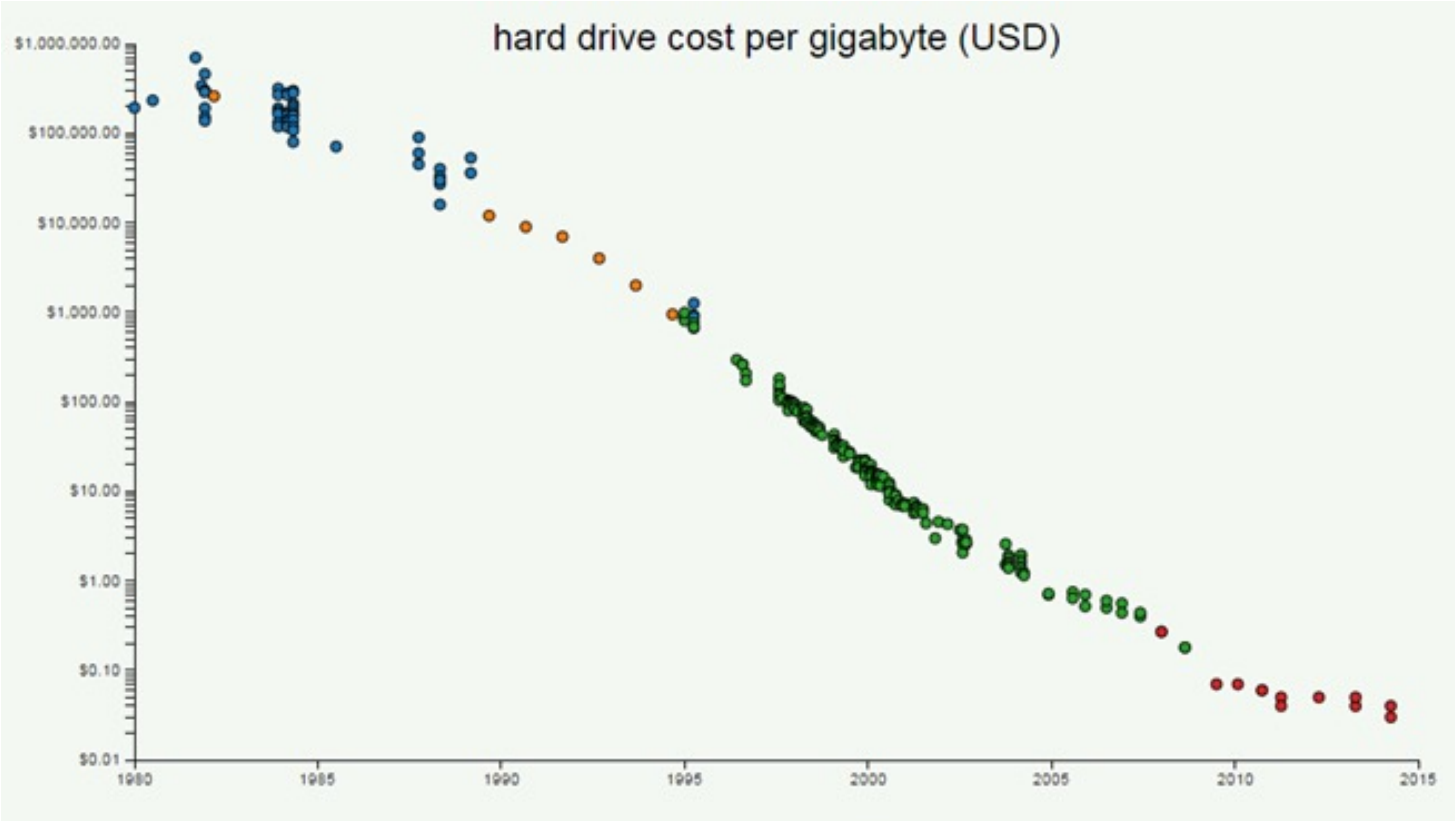
2017

2x

25,600x

What is Deep Learning?

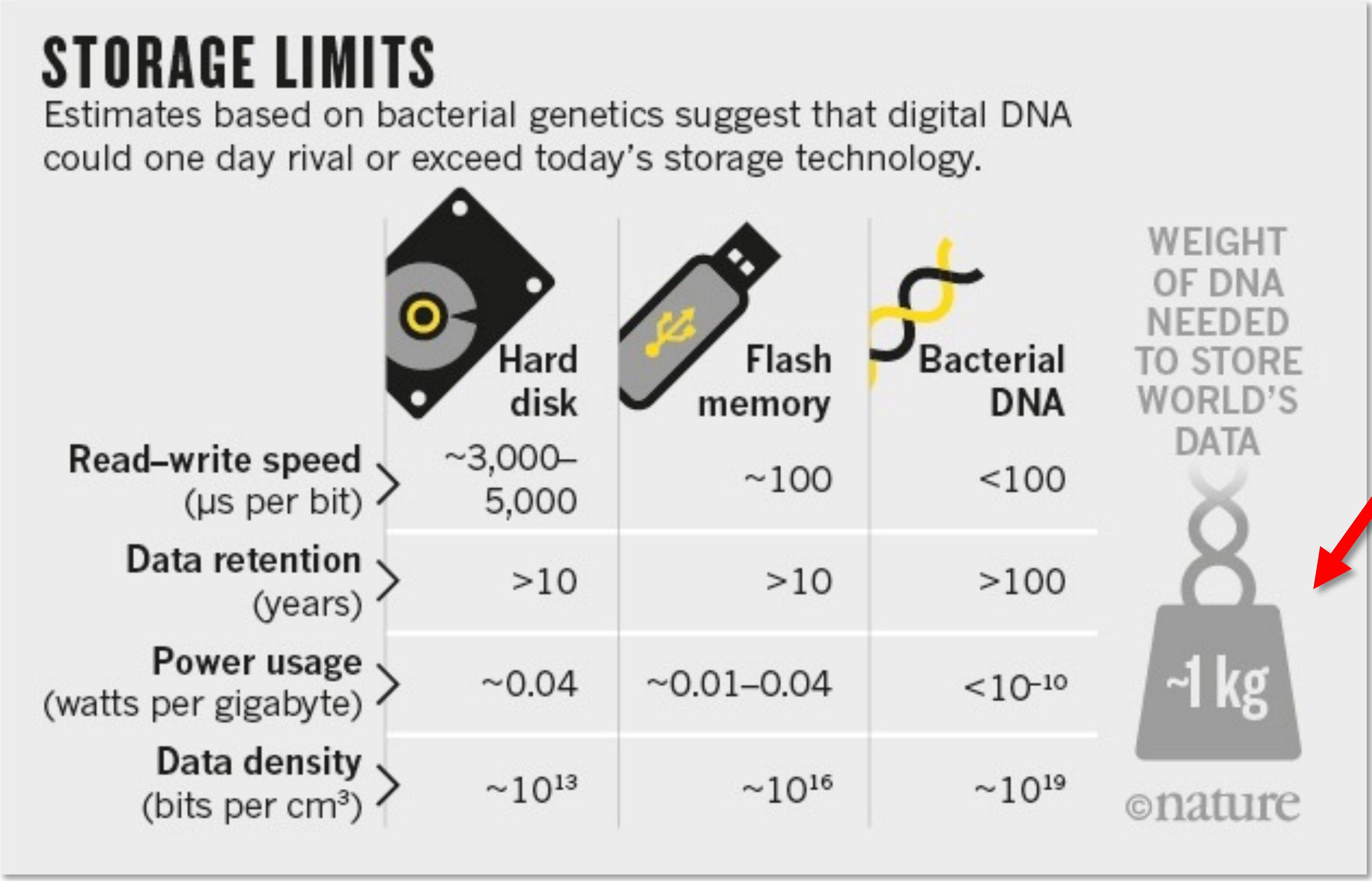
Log-scale



Source: mkomo.com

What is Deep Learning?

What is Deep Learning?



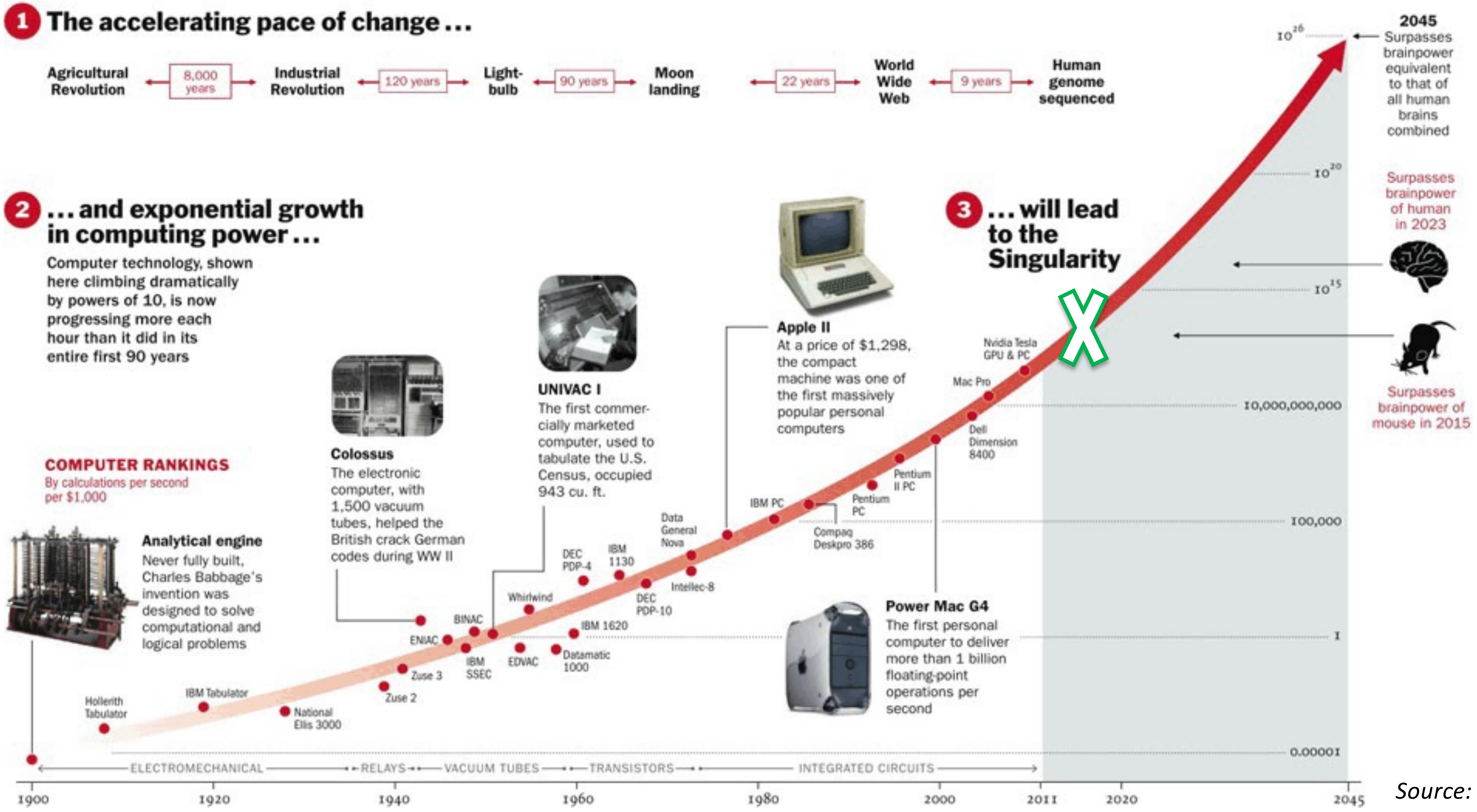
STORAGE LIMITS

Estimates based on bacterial genetics suggest that digital DNA could one day rival or exceed today's storage technology.

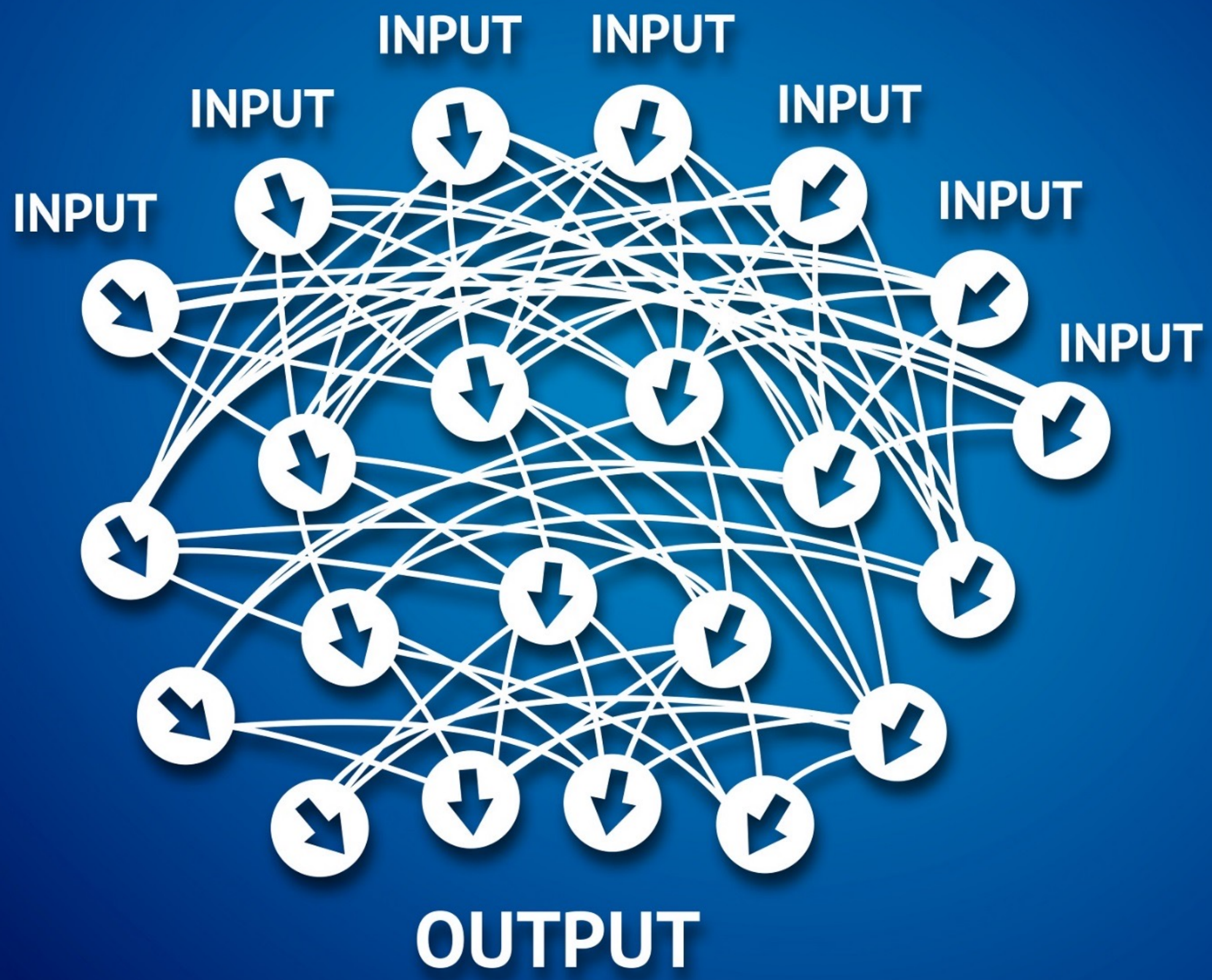
	Hard disk	Flash memory	Bacterial DNA
Read–write speed (μs per bit)	~3,000–5,000	~100	<100
Data retention (years)	>10	>10	>100
Power usage (watts per gigabyte)	~0.04	~0.01–0.04	$<10^{-10}$
Data density (bits per cm^3)	$\sim10^{13}$	$\sim10^{16}$	$\sim10^{19}$

Source: nature.com

What is Deep Learning?



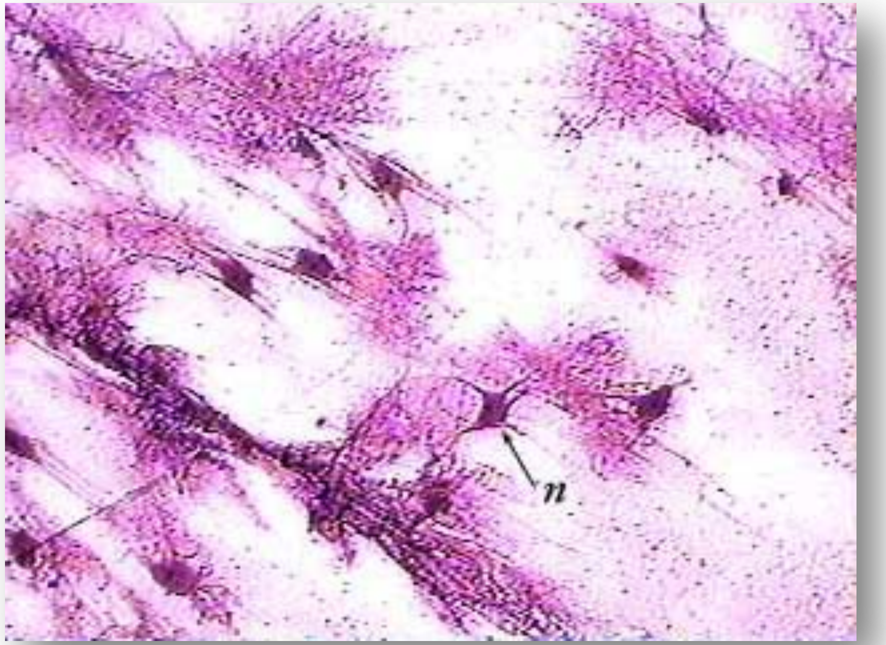
Source: Time Magazine


INPUT
INPUT
INPUT
INPUT
INPUT
INPUT
INPUT
OUTPUT

What is Deep Learning?

Geoffrey Hinton

What is Deep Learning?



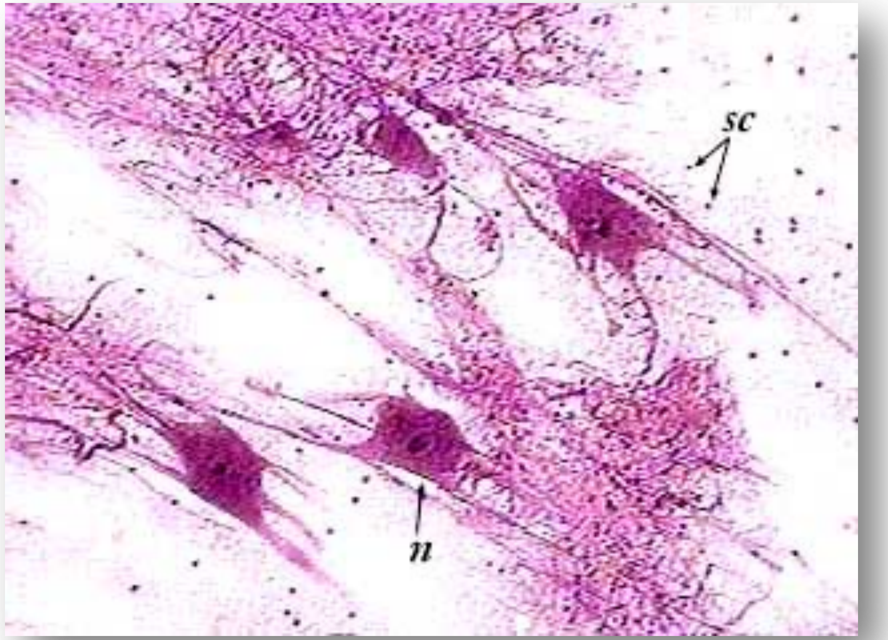
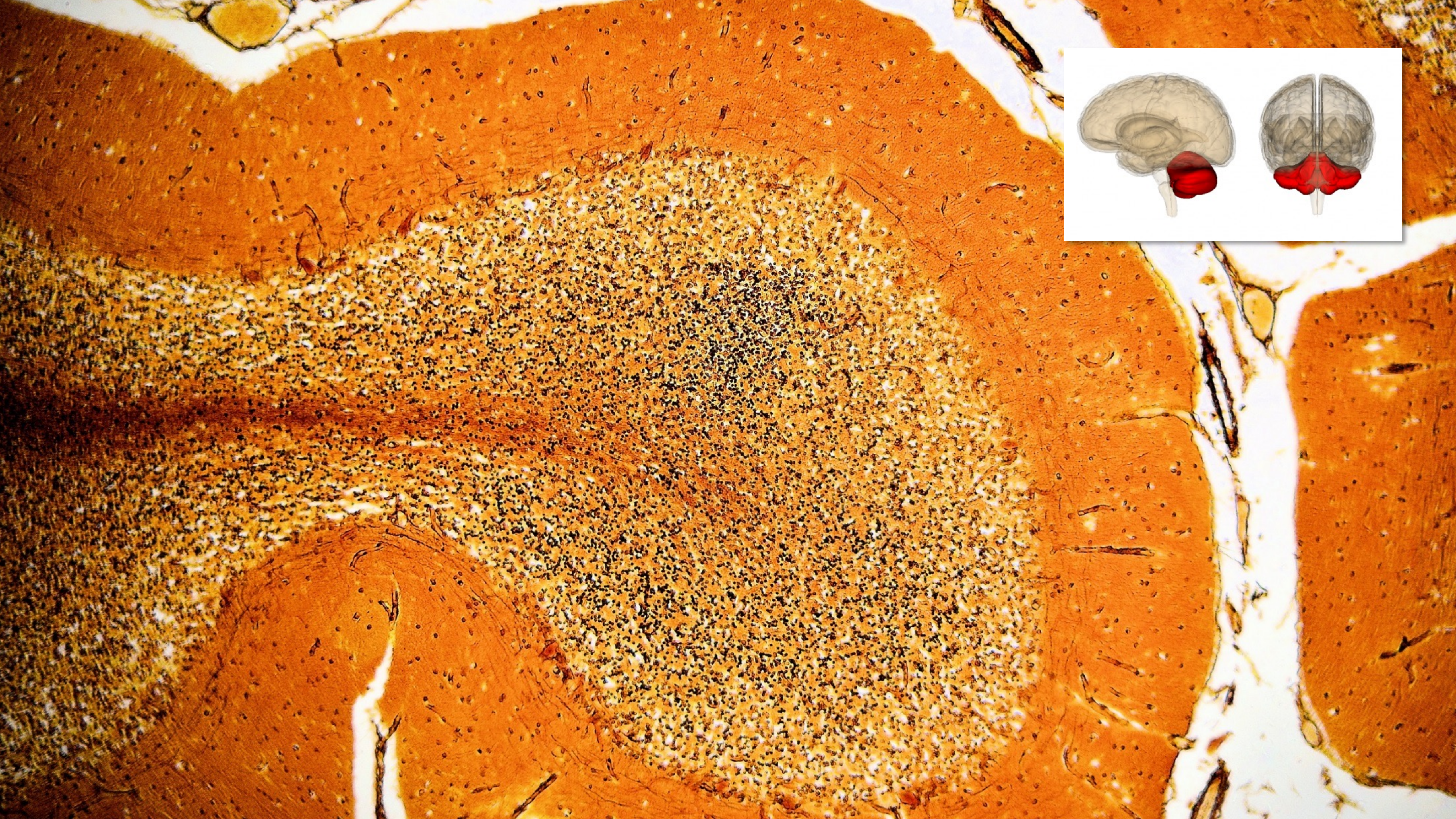
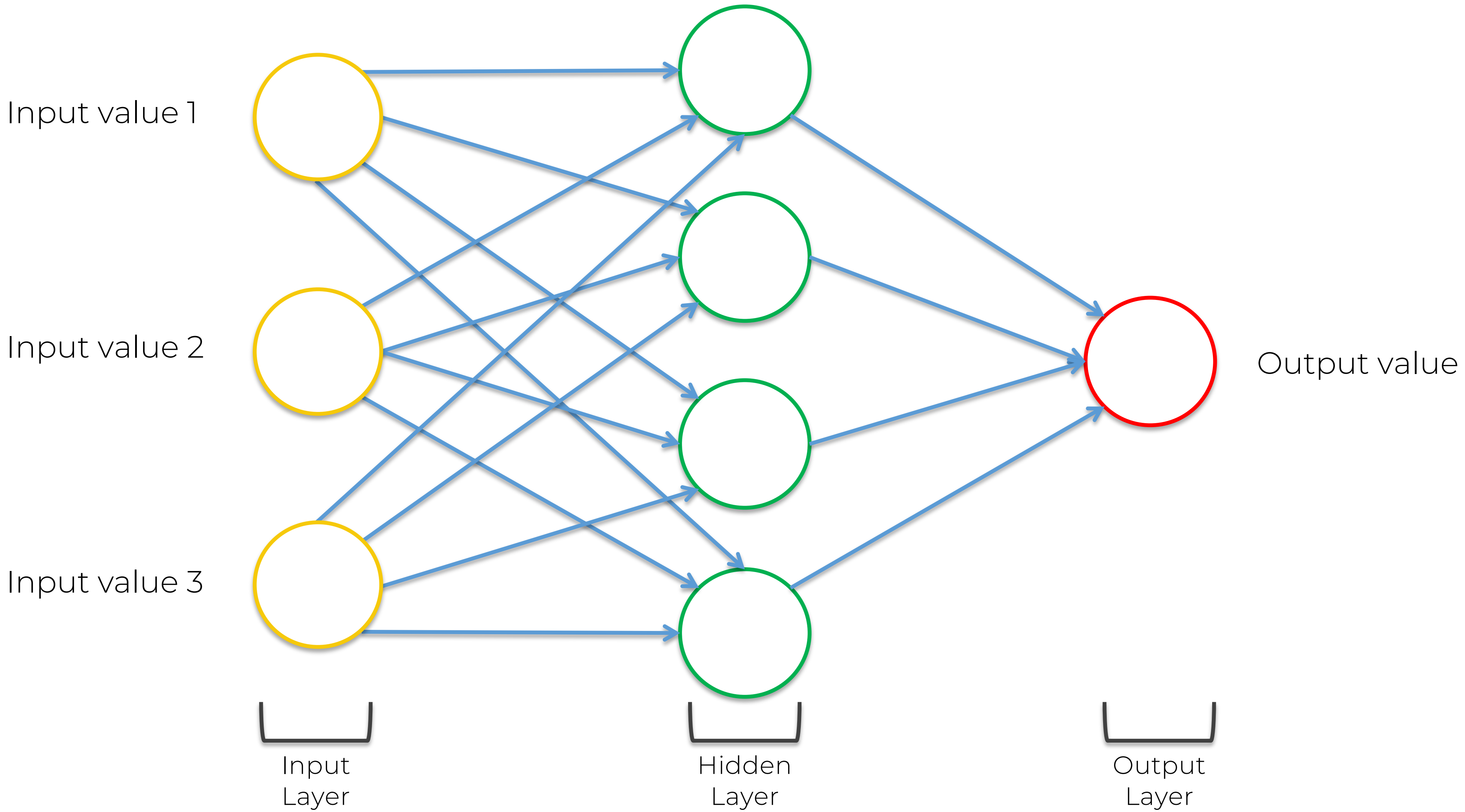


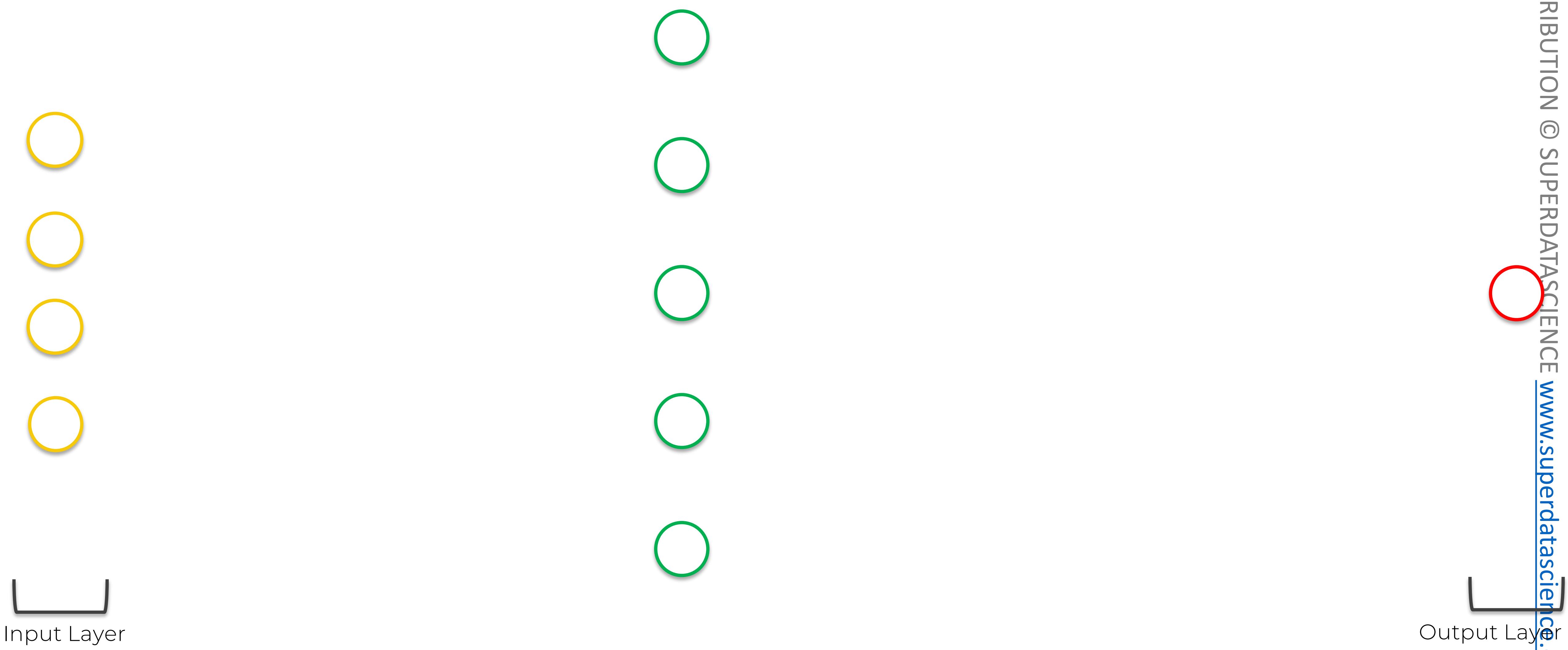
Image Source: www.austincc.edu



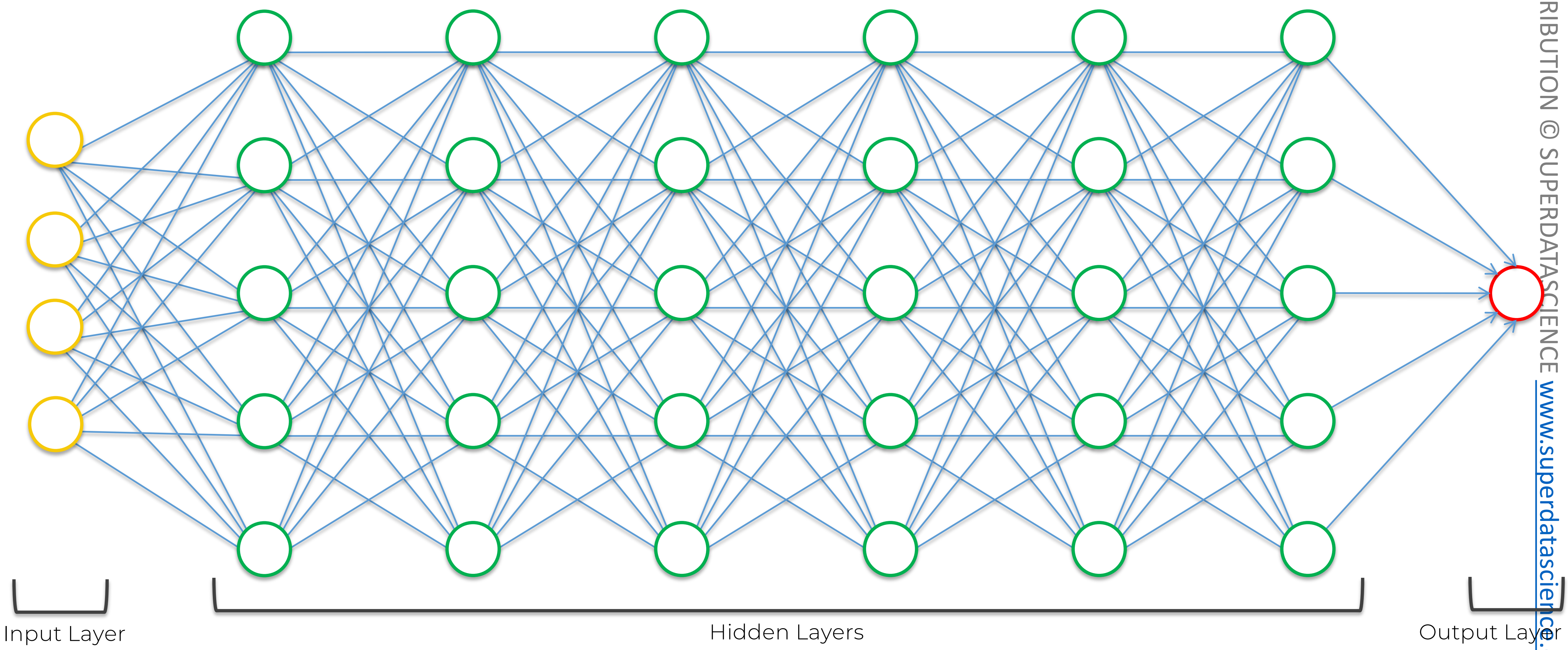
What is Deep Learning?



What is Deep Learning?



What is Deep Learning?



Supervised vs Unsupervised

Supervised vs Unsupervised

Supervised	Artificial Neural Networks	Used for Regression & Classification
	Convolutional Neural Networks	Used for Computer Vision
	Recurrent Neural Networks	Used for Time Series Analysis
Unsupervised	Self-Organizing Maps	Used for Feature Detection
	Deep Boltzmann Machines	Used for Recommendation Systems
	AutoEncoders	Used for Recommendation Systems

Plan of Attack

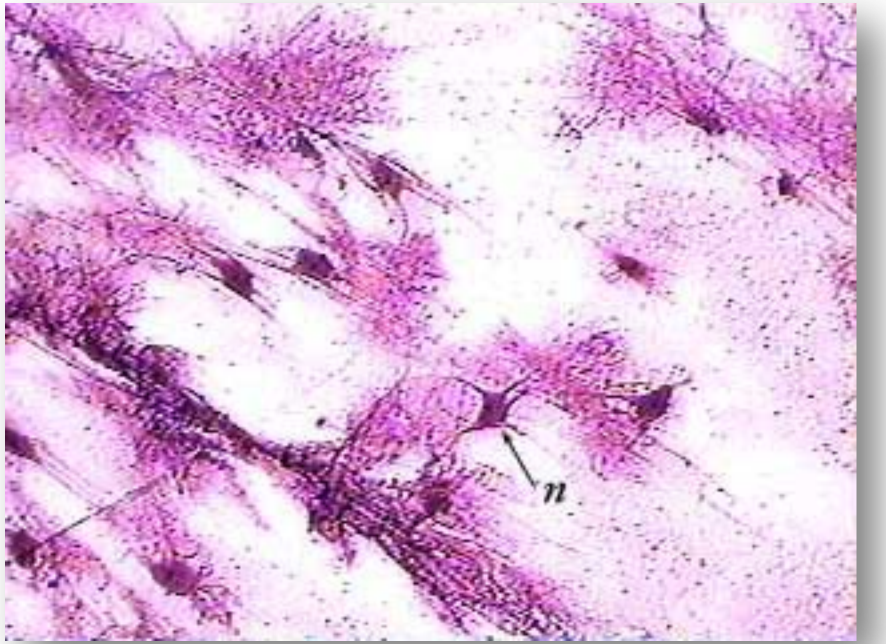
Plan of Attack

What we will learn in this section:

- The Neuron
- The Activation Function
- How do Neural Networks work? (example)
- How do Neural Networks learn?
- Gradient Descent
- Stochastic Gradient Descent
- Backpropagation

The Neuron

The Neuron



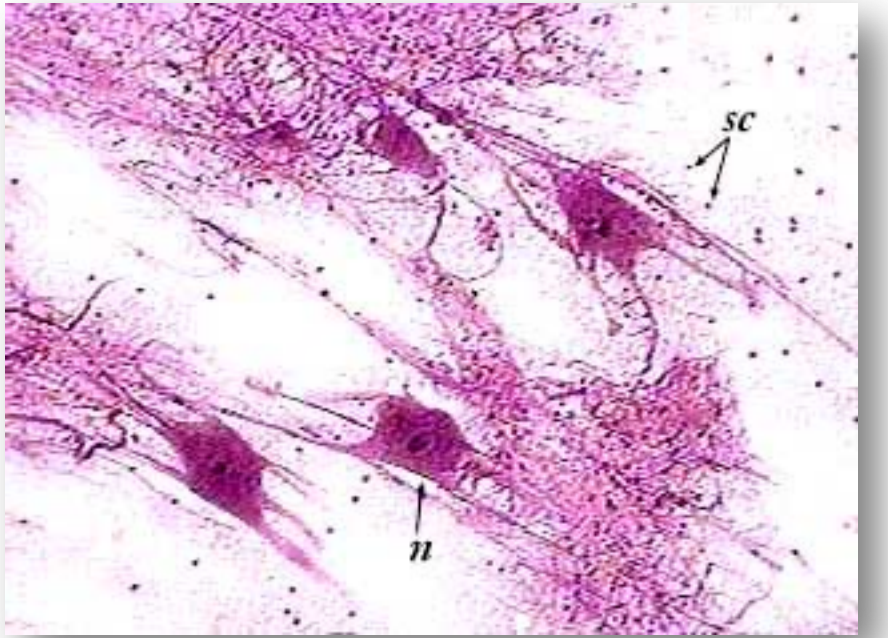


Image Source: www.austincc.edu

The Neuron

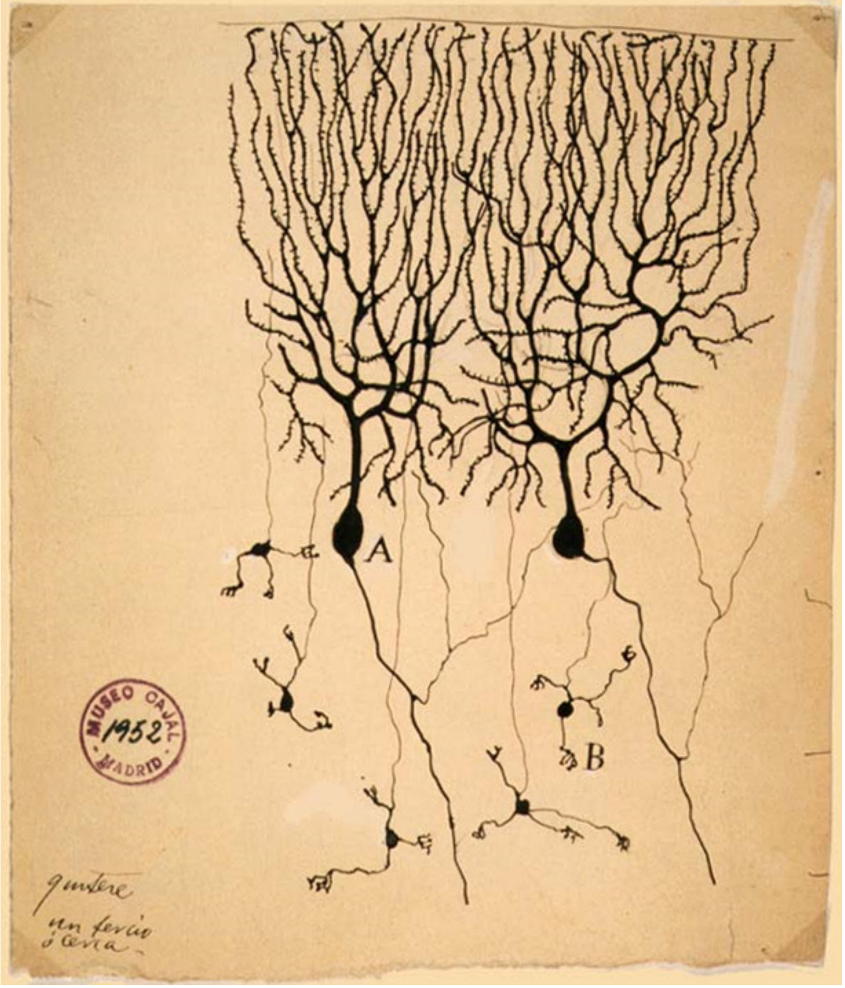


Image Source: Wikipedia

The Neuron

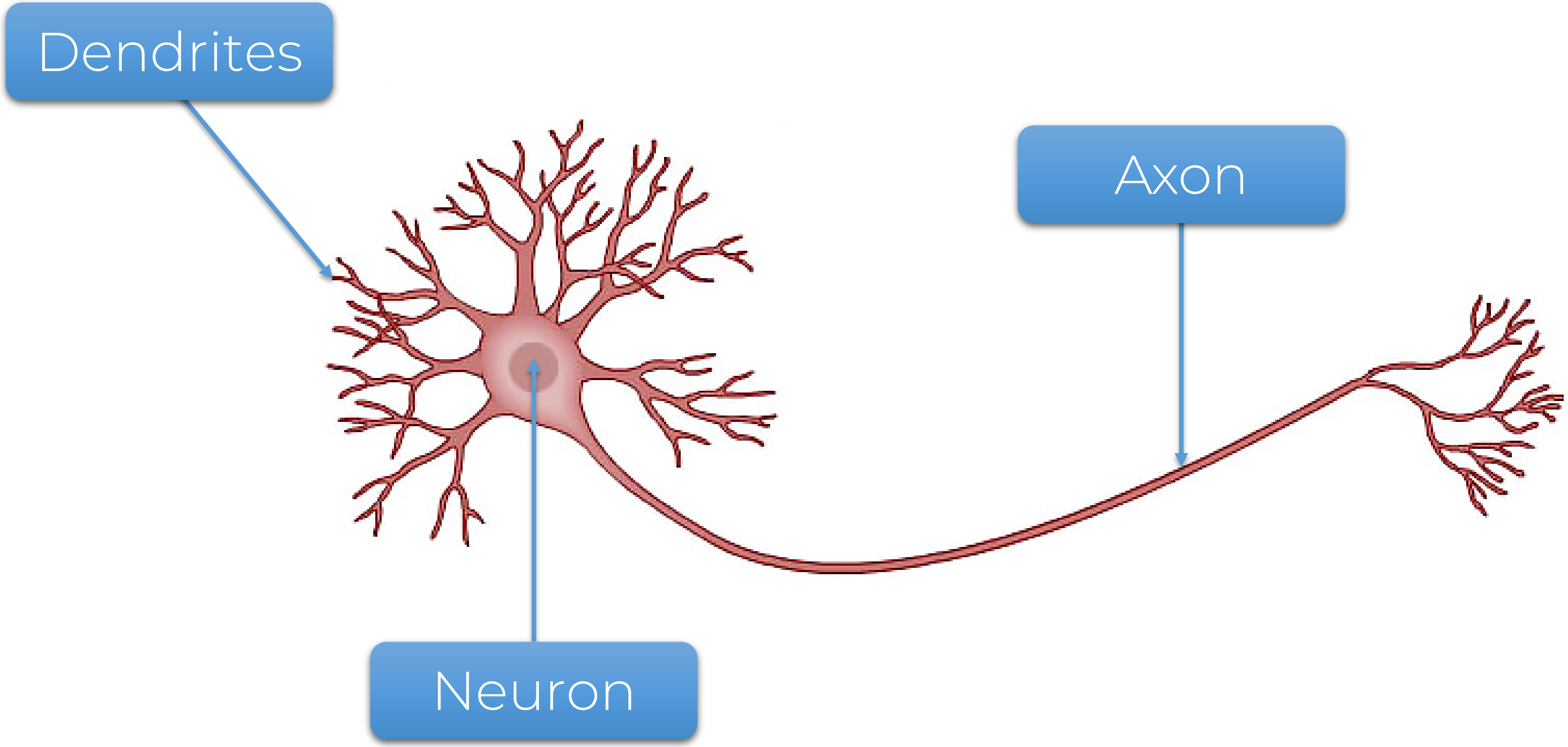


Image Source: Wikipedia

The Neuron

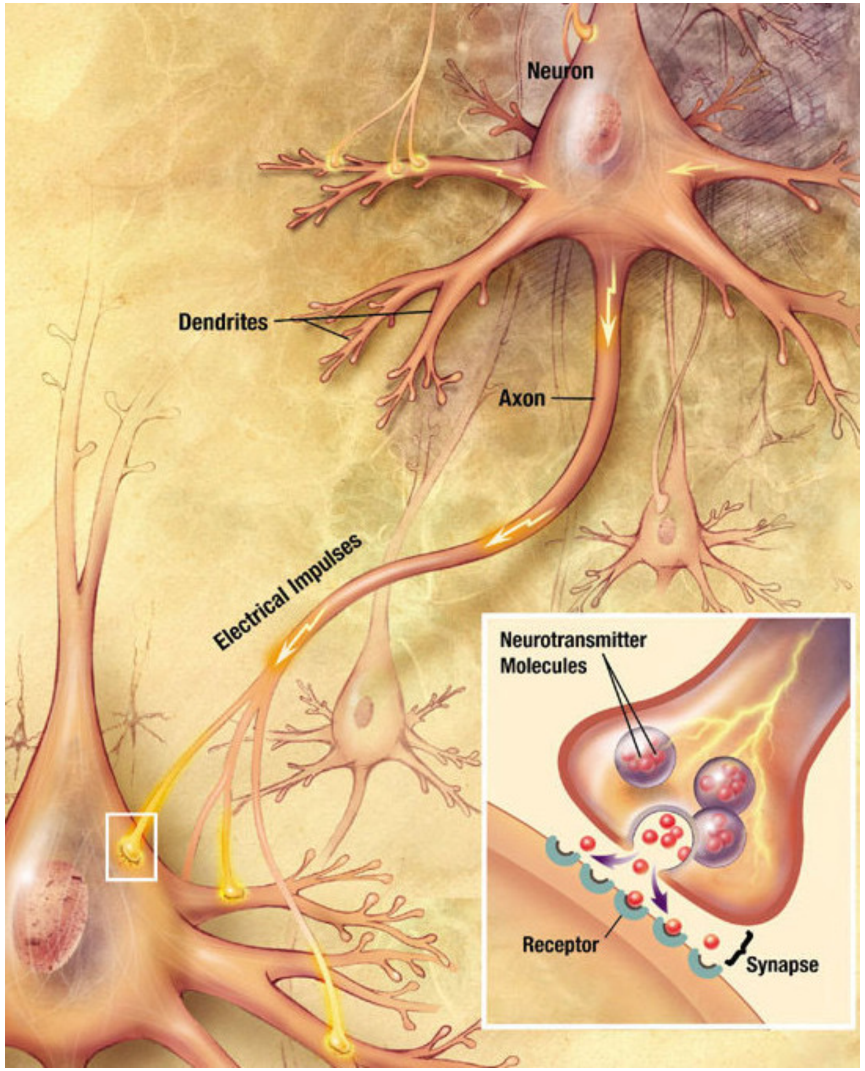
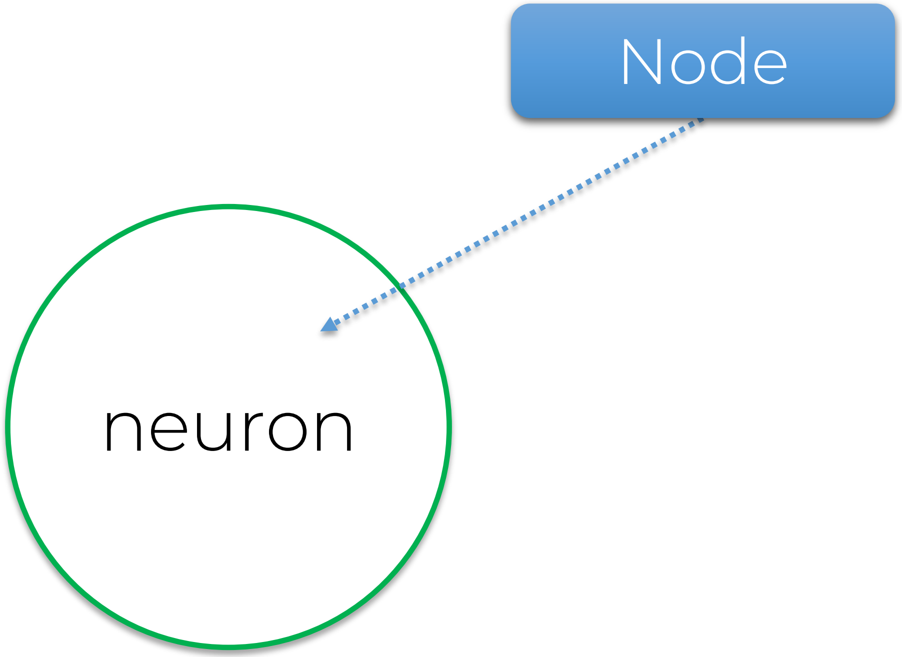
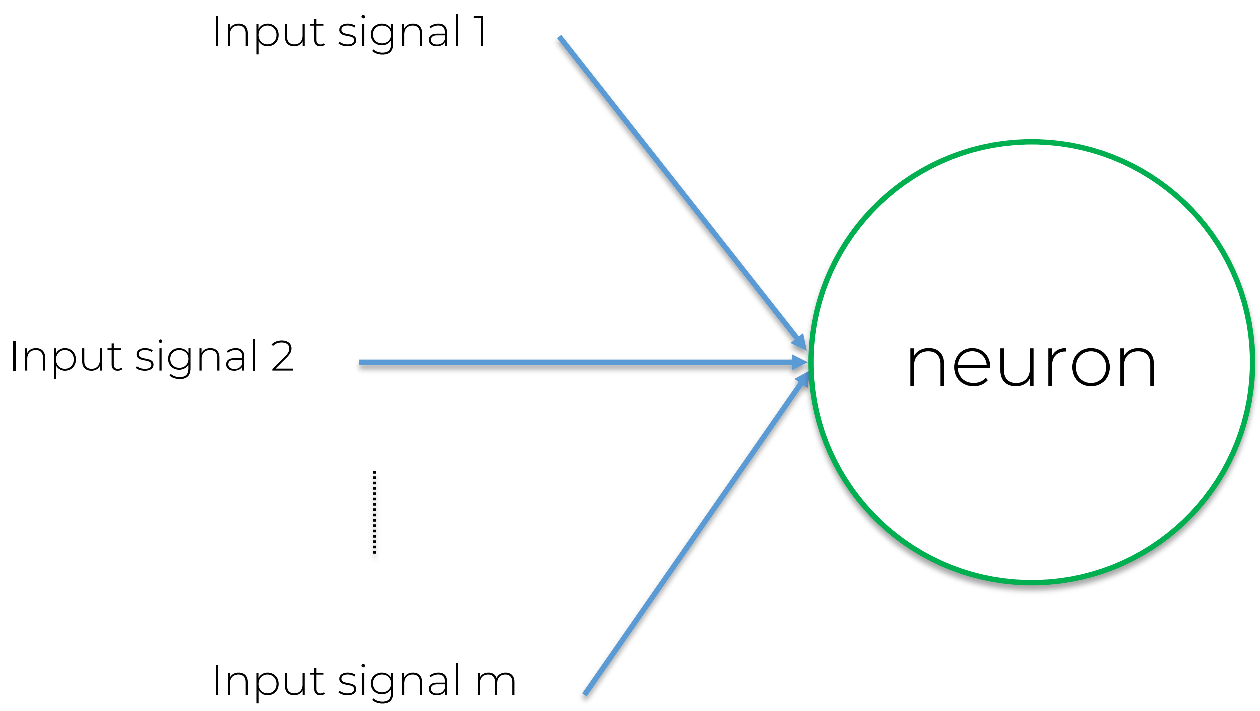


Image Source: Wikipedia

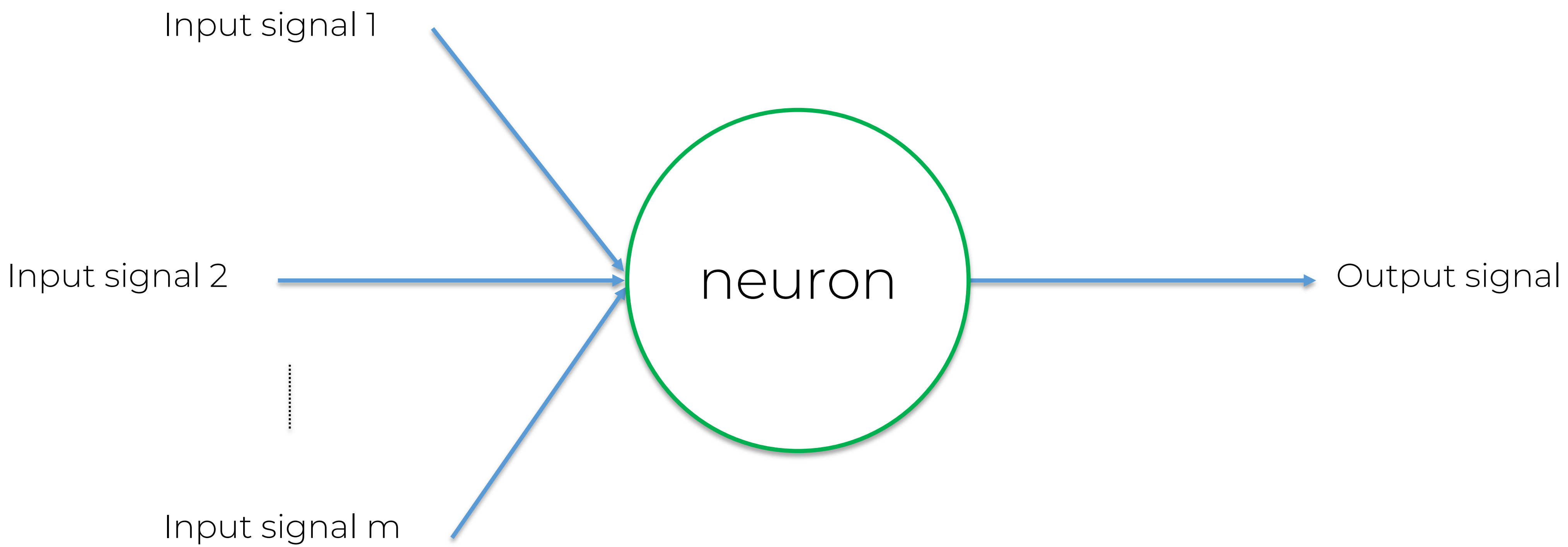
The Neuron



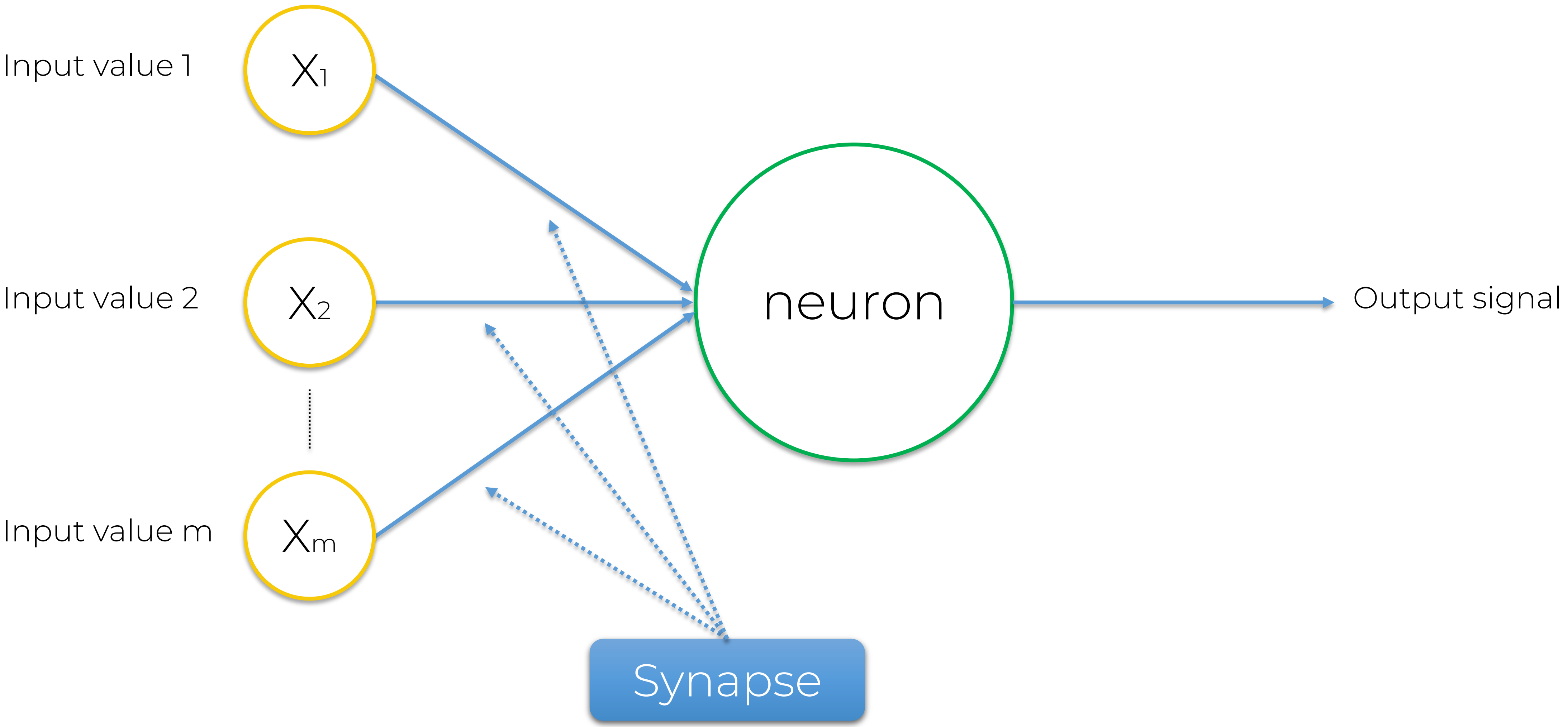
The Neuron



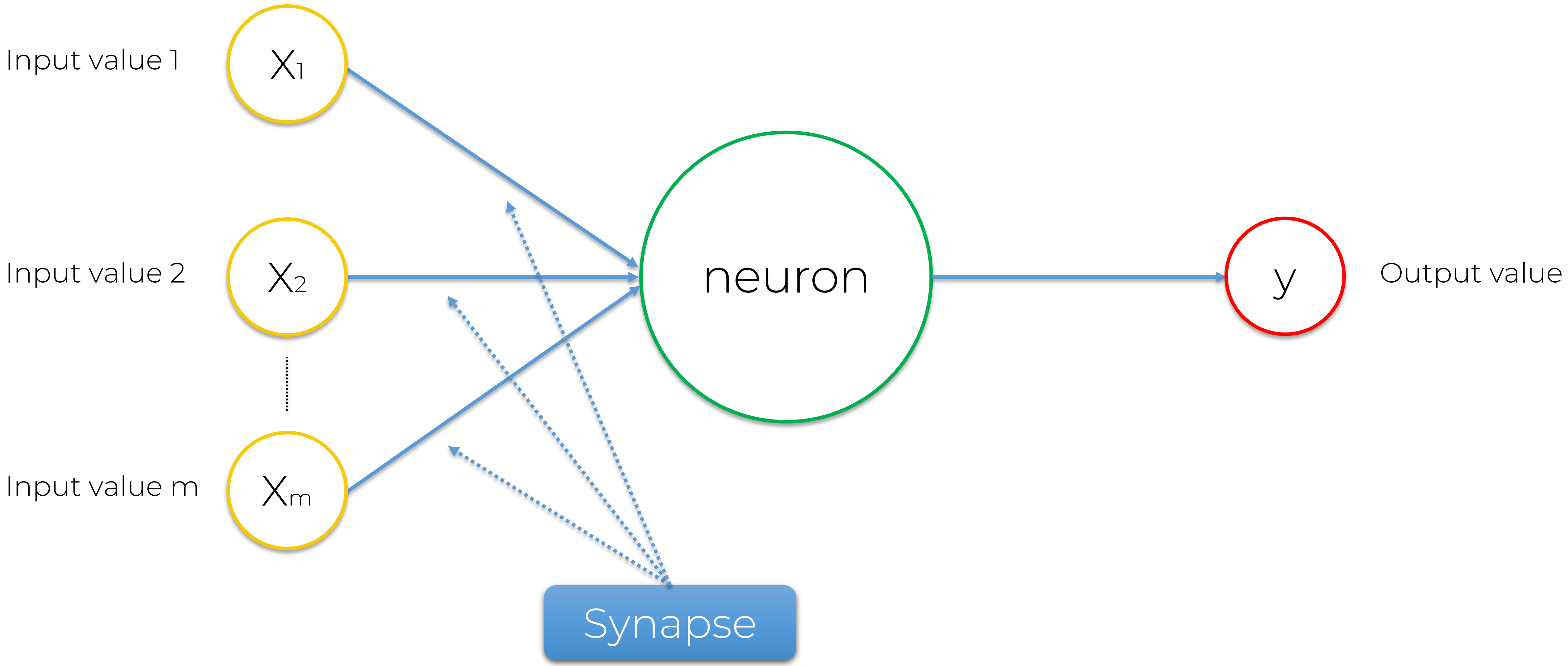
The Neuron


Input signal 1
Input signal 2
Input signal m
neuron
Output signal

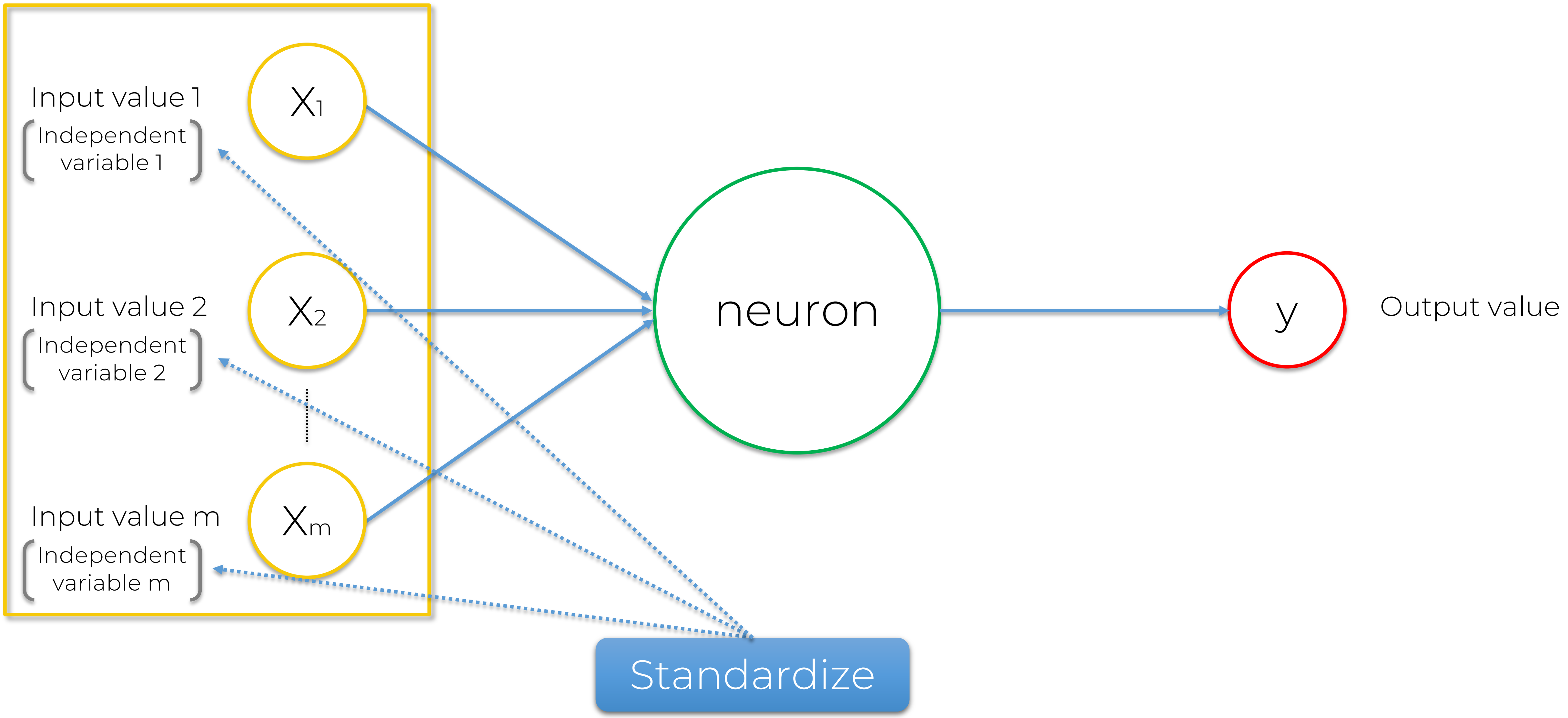
The Neuron


Input value 1
Input value 2
Input value m
X1
X2
Xm
neuron
Output signal
Synapse

The Neuron


Input value 1
Input value 2
Input value m
X_1
X_2
X_m
neuron
y
Output value
Synapse

The Neuron



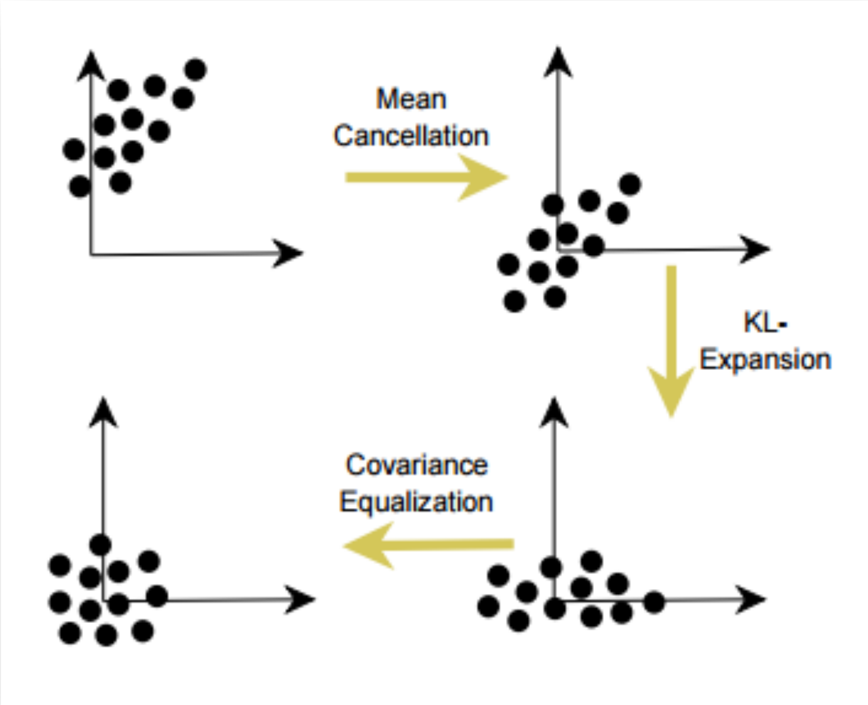
Input value 1
Independent variable 1
Input value 2
Independent variable 2
Input value m
Independent variable m
X1
X2
Xm
neuron
y
Output value
Standardize

The Neuron

Additional Reading:

Efficient BackProp

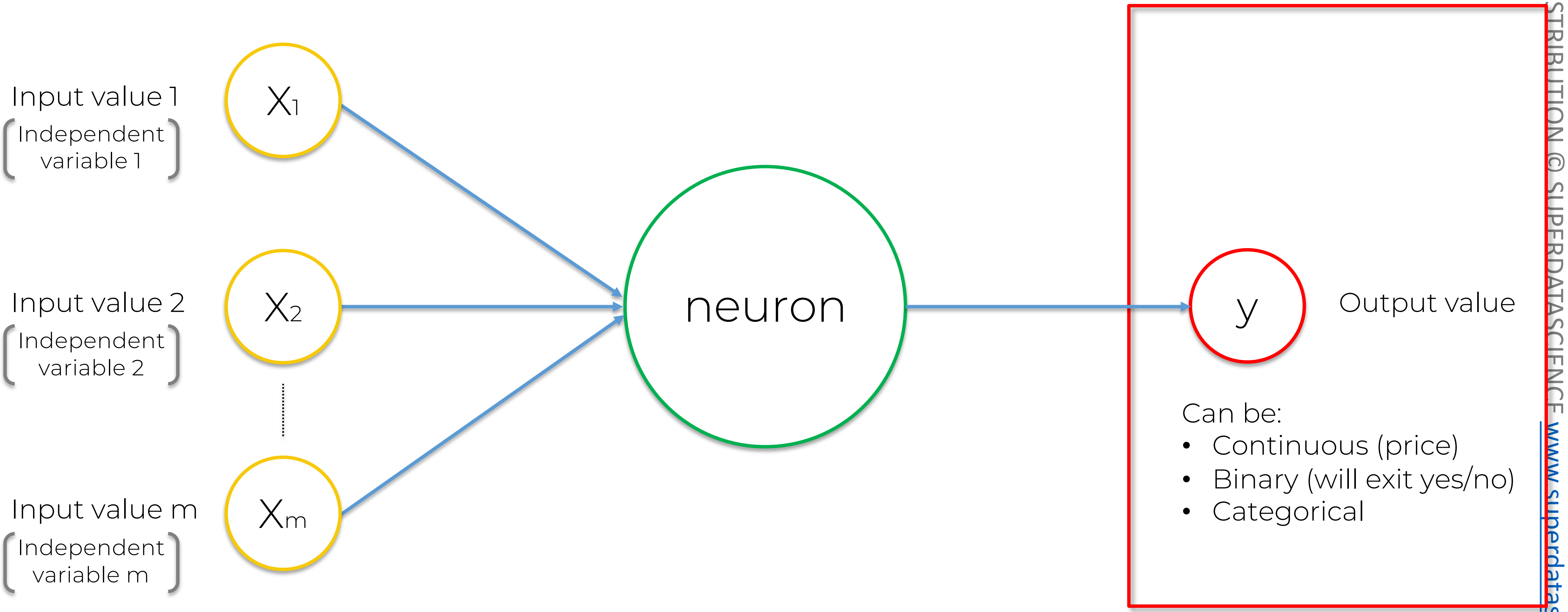
By Yann LeCun et al. (1998)



Link:

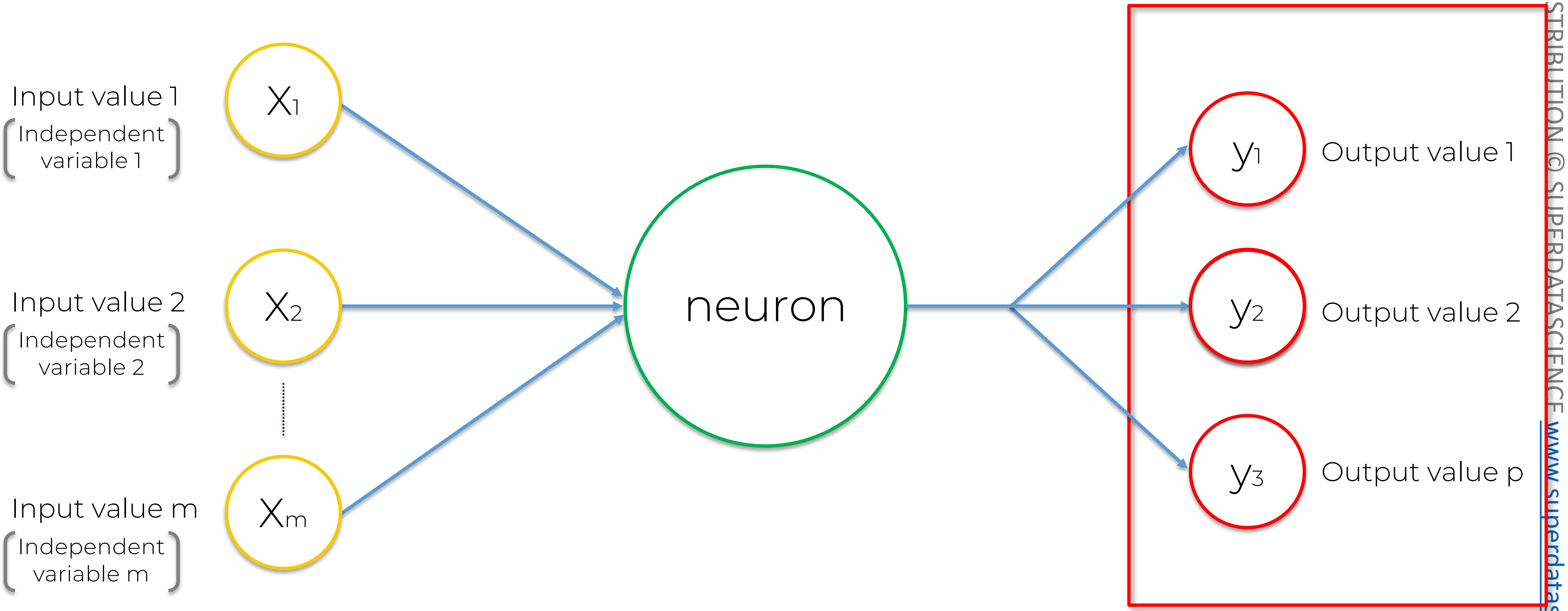
http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf

The Neuron



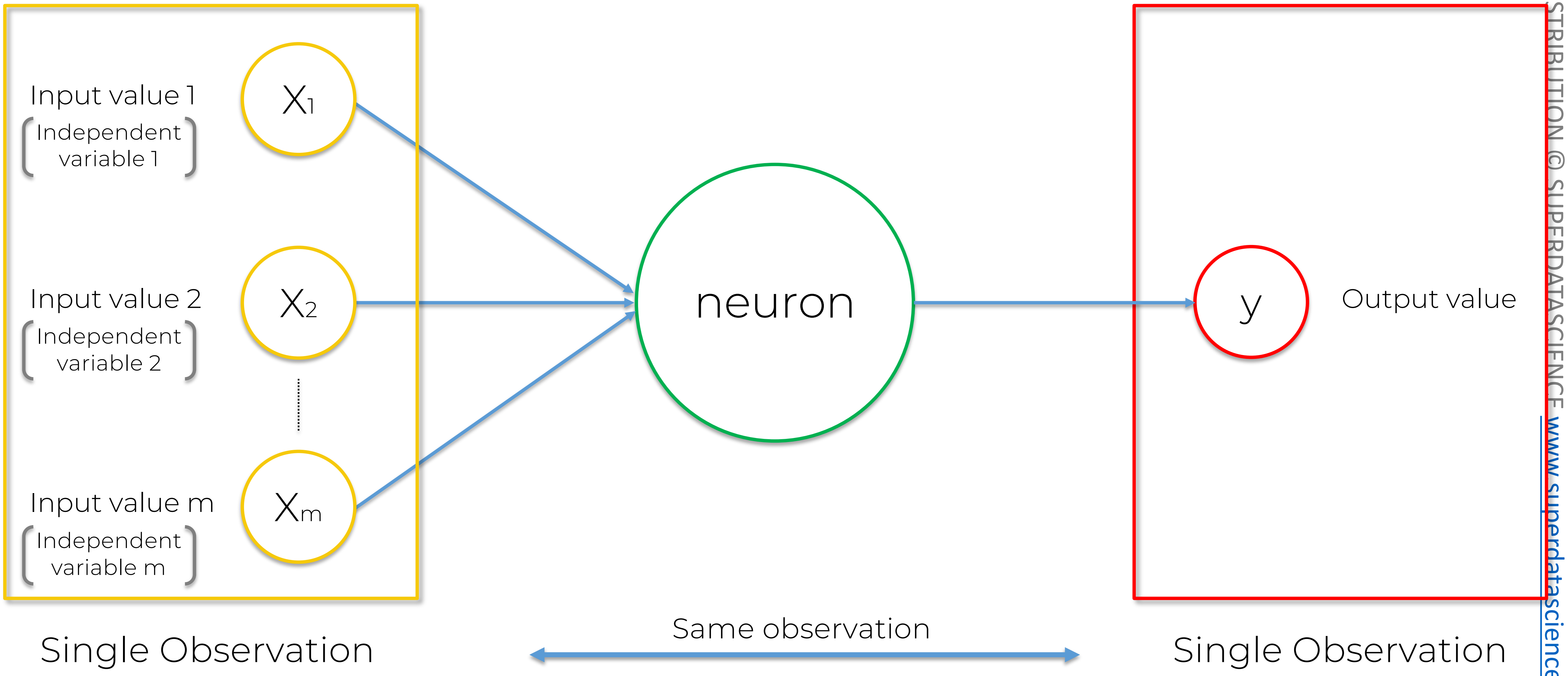
Input value 1
Independent variable 1
X_1
Input value 2
Independent variable 2
X_2
Input value m
Independent variable m
X_m
neuron
y
Output value
Can be:
• Continuous (price)
• Binary (will exit yes/no)
• Categorical

The Neuron

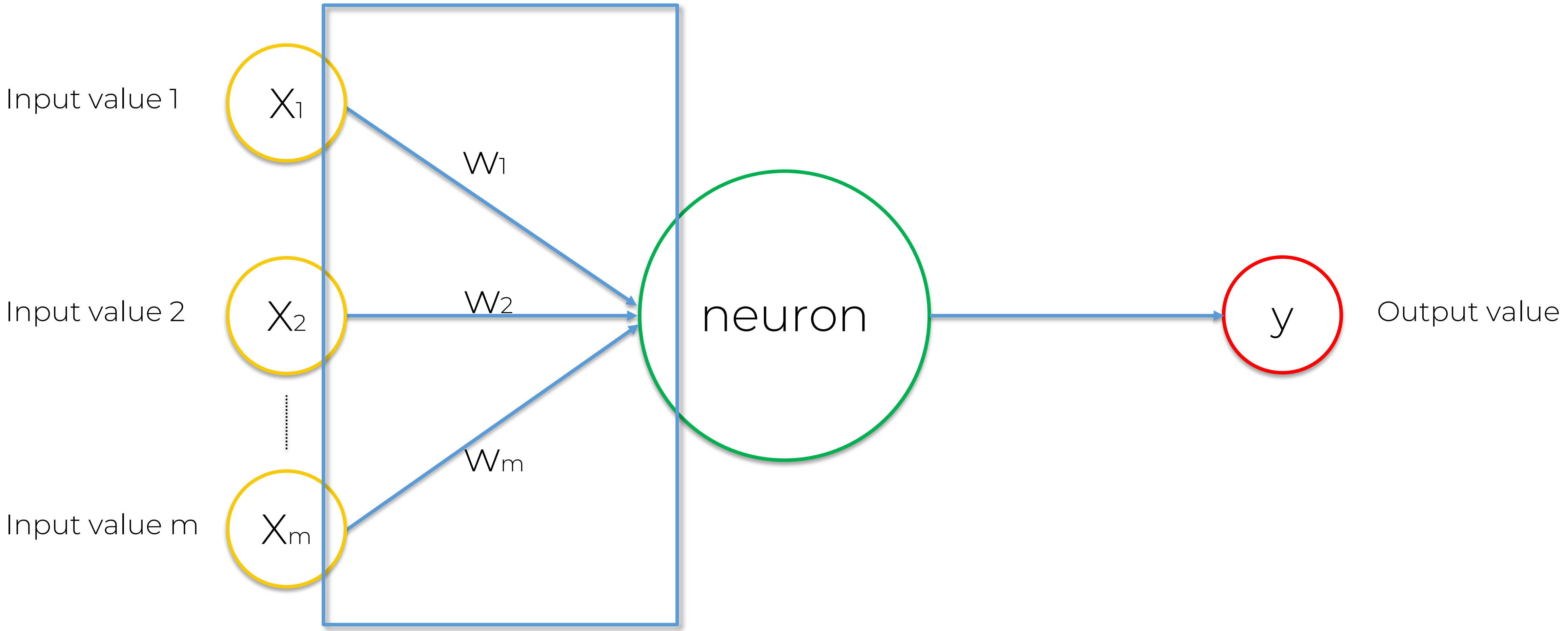


Input value 1
Independent variable 1
X_1
Input value 2
Independent variable 2
X_2
Input value m
Independent variable m
X_m
neuron
y_1
Output value 1
y_2
Output value 2
y_3
Output value p

The Neuron

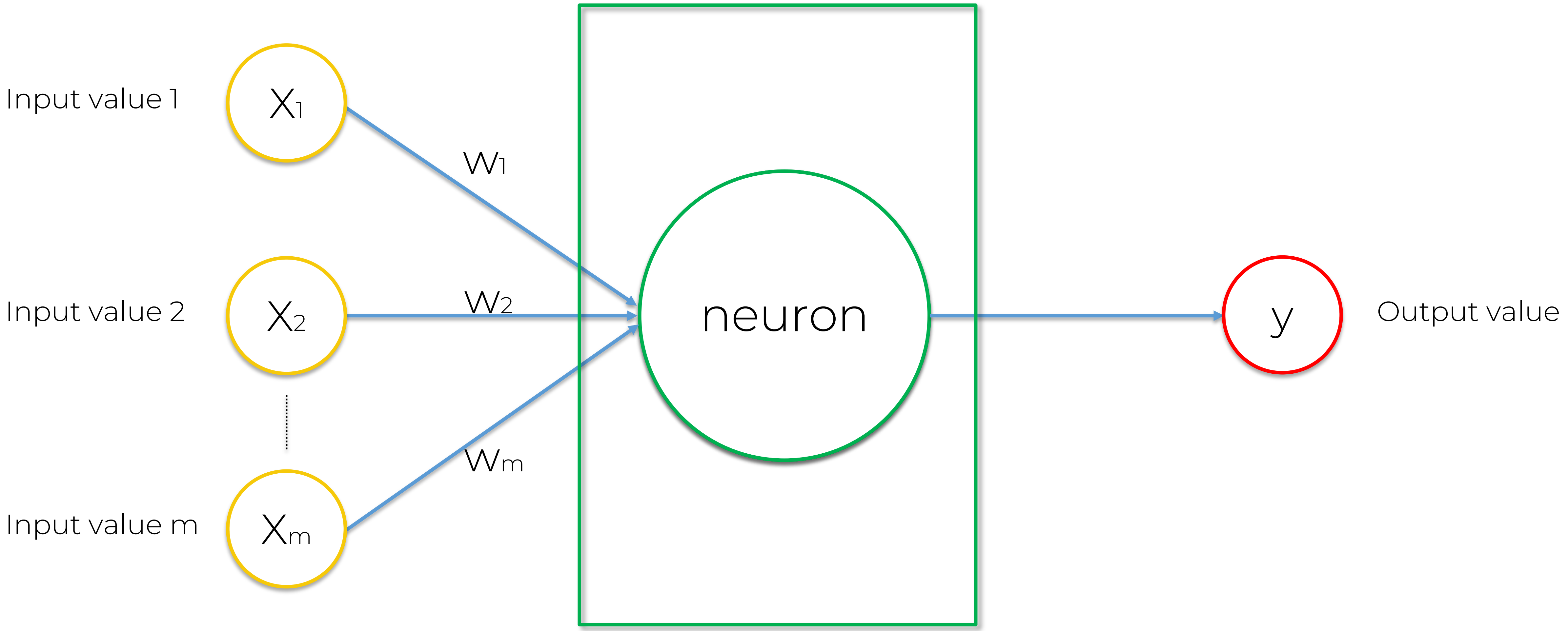

Input value 1
Independent variable 1
X_1
Input value 2
Independent variable 2
X_2
Input value m
Independent variable m
X_m
neuron
y
Output value
Single Observation
Same observation
Single Observation

The Neuron



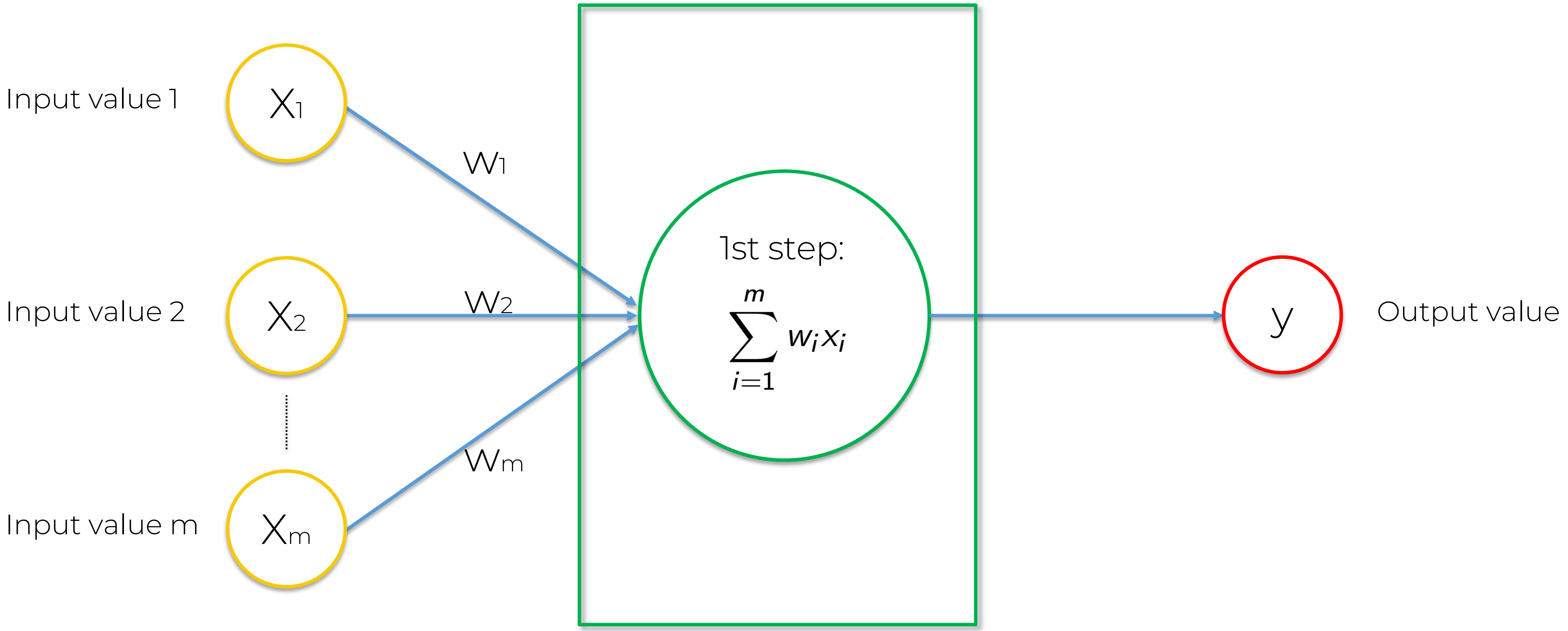
Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
neuron
y
Output value

The Neuron



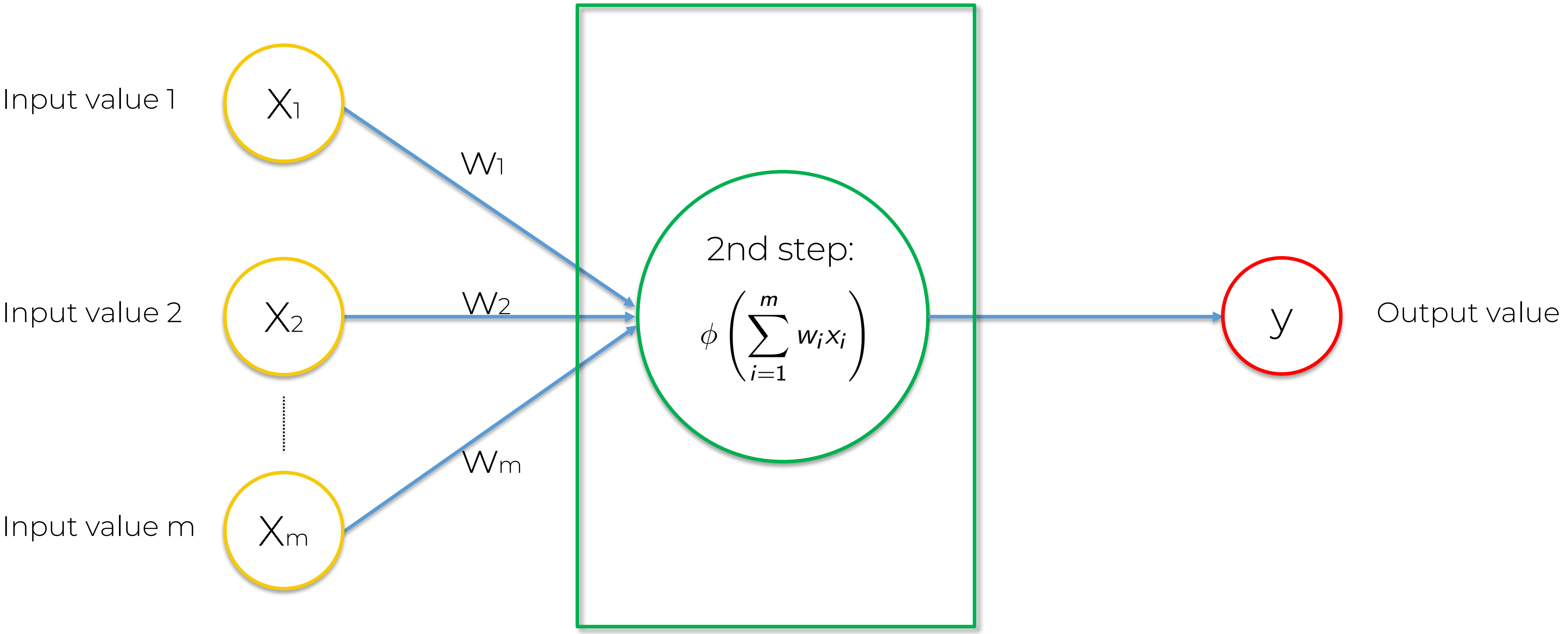
Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
neuron
y
Output value

The Neuron



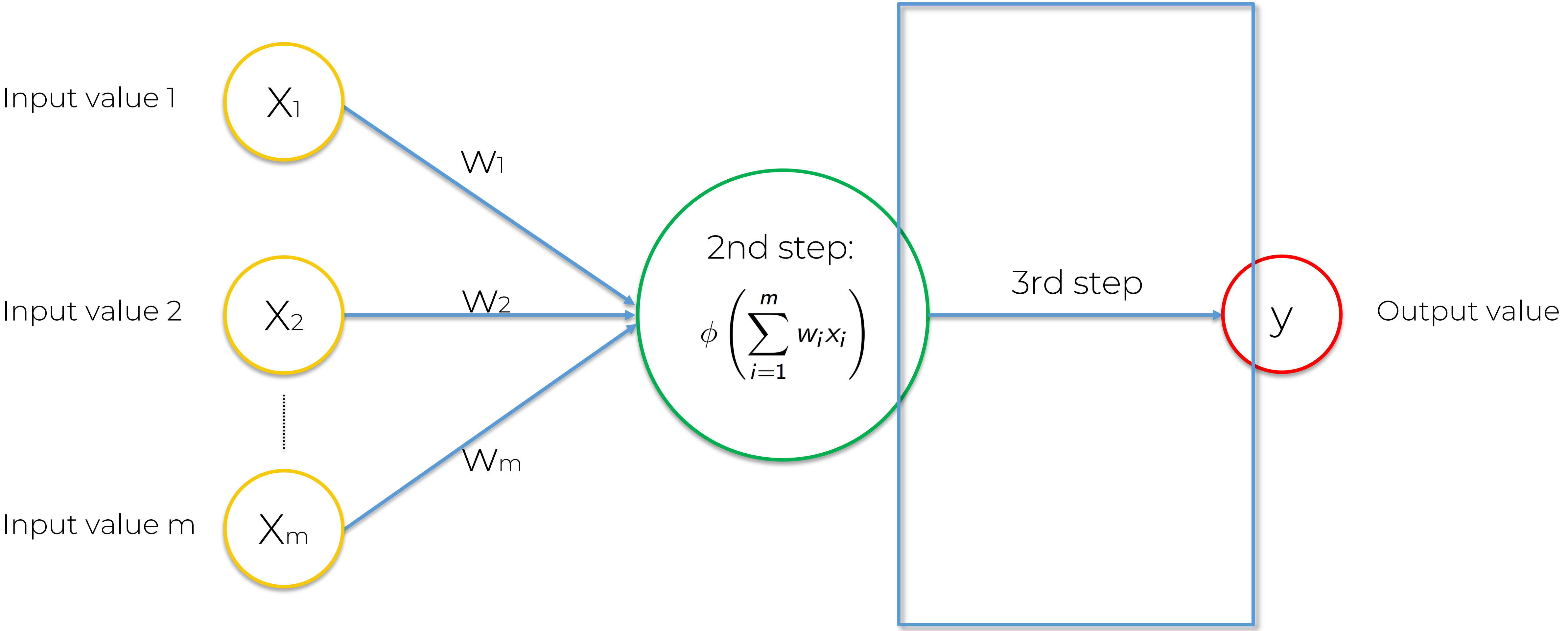
Input value 1
Input value 2
Input value m
X1
X2
Xm
W1
W2
Wm
1st step:
$\sum_{i=1}^{m} w_i x_i$
y
Output value

The Neuron



Input value 1
Input value 2
Input value m
X_1
X_2
X_m
W_1
W_2
W_m
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
y
Output value

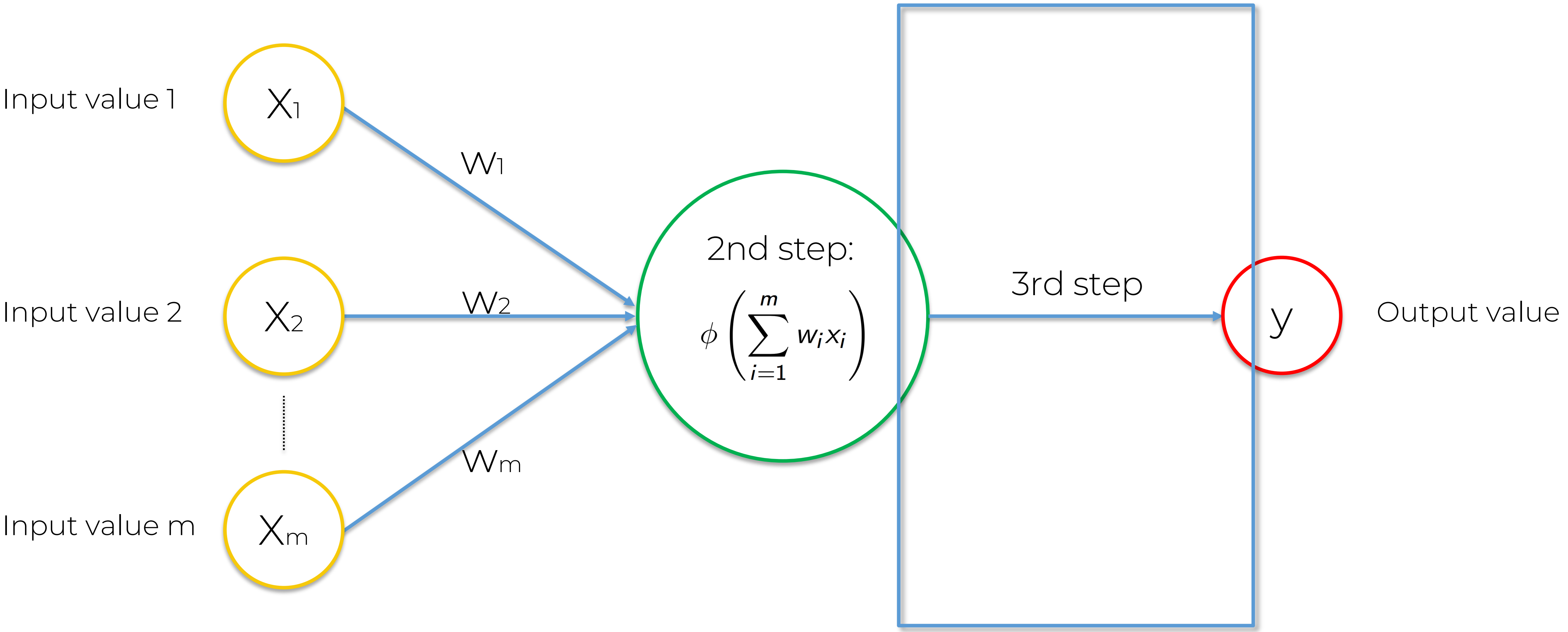
The Neuron



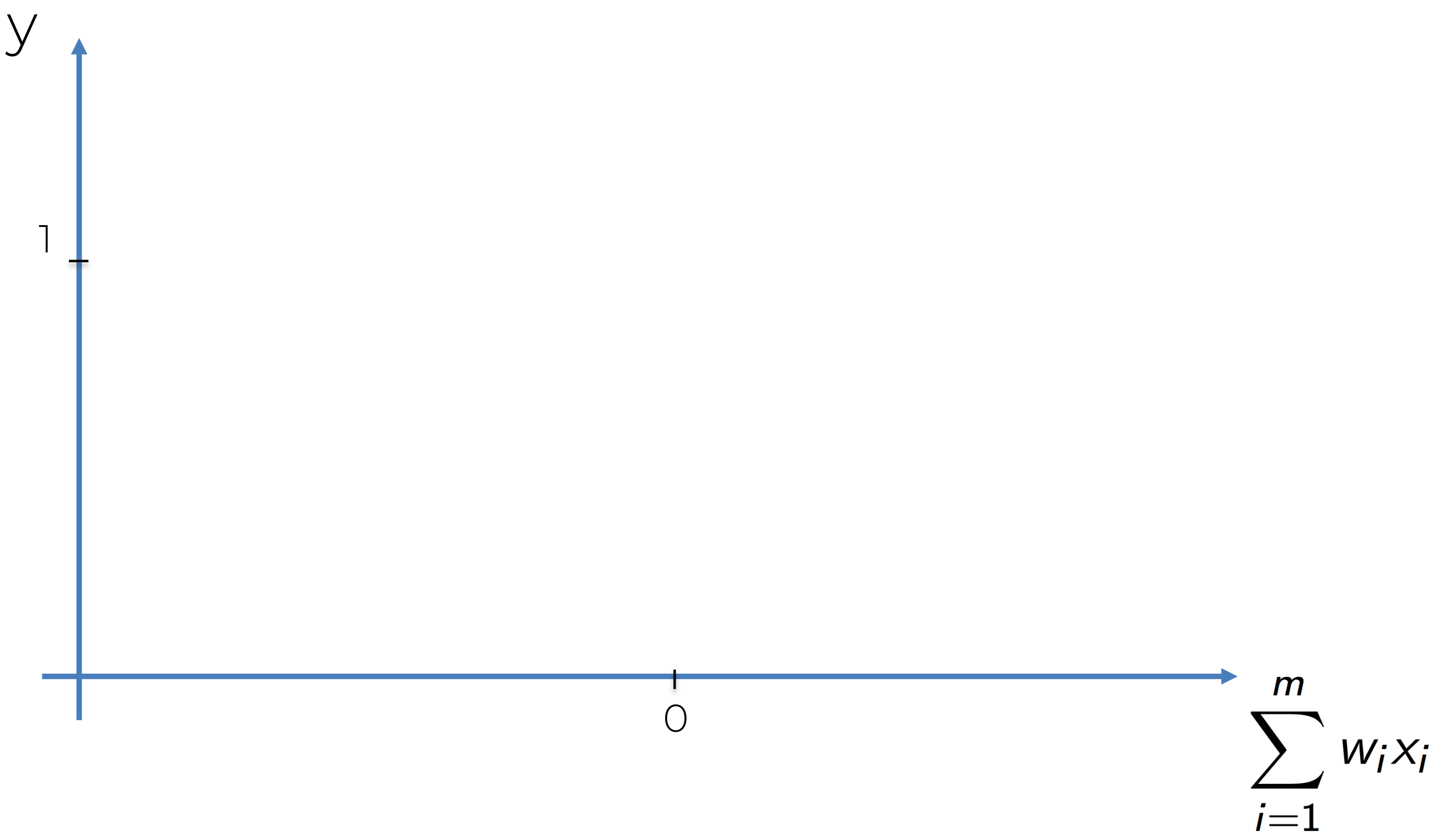
Input value 1
Input value 2
Input value m
X_1
X_2
X_m
W_1
W_2
W_m
2nd step:
$\phi\left(\sum_{i=1}^{m} w_i x_i\right)$
3rd step
y
Output value

The Activation Function

The Activation Function

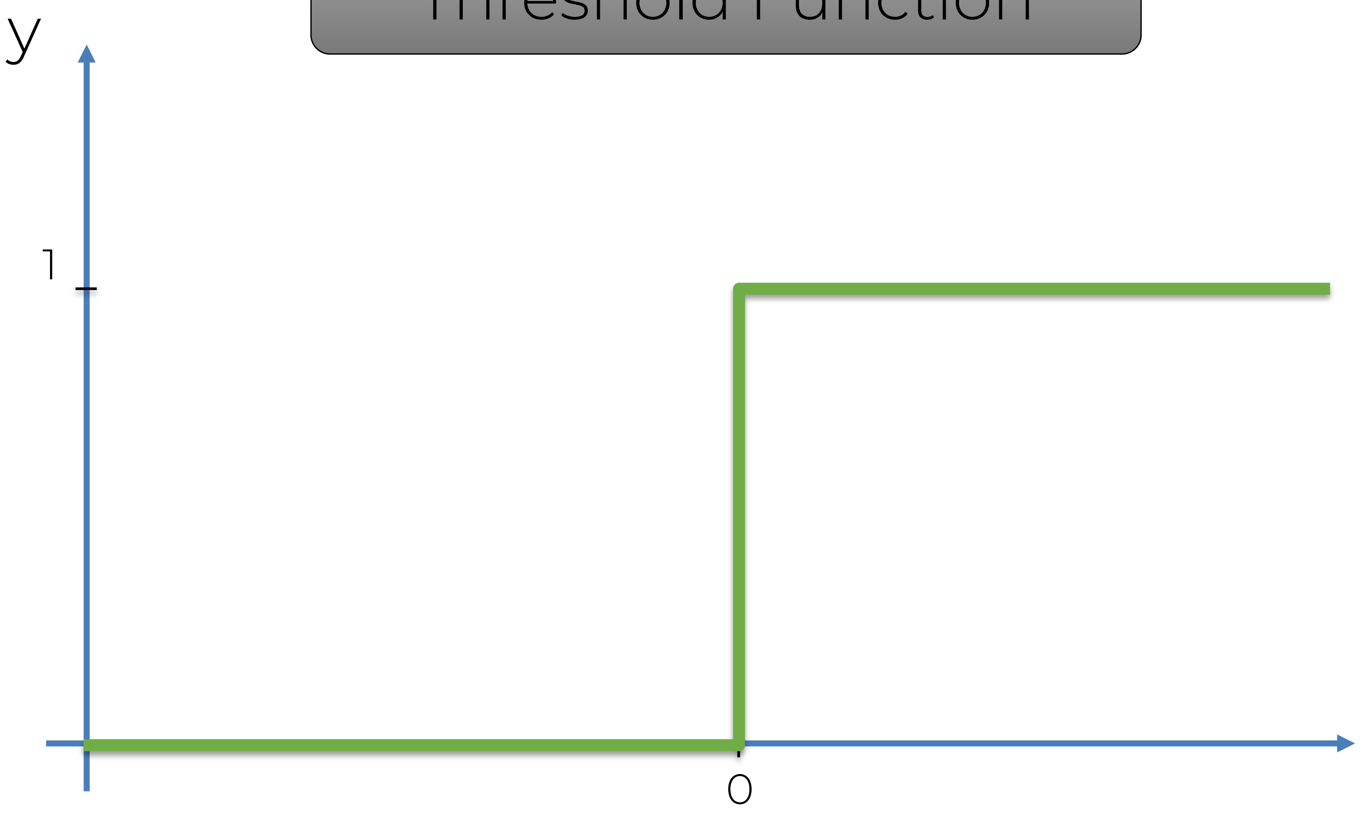


The Activation Function



The Activation Function

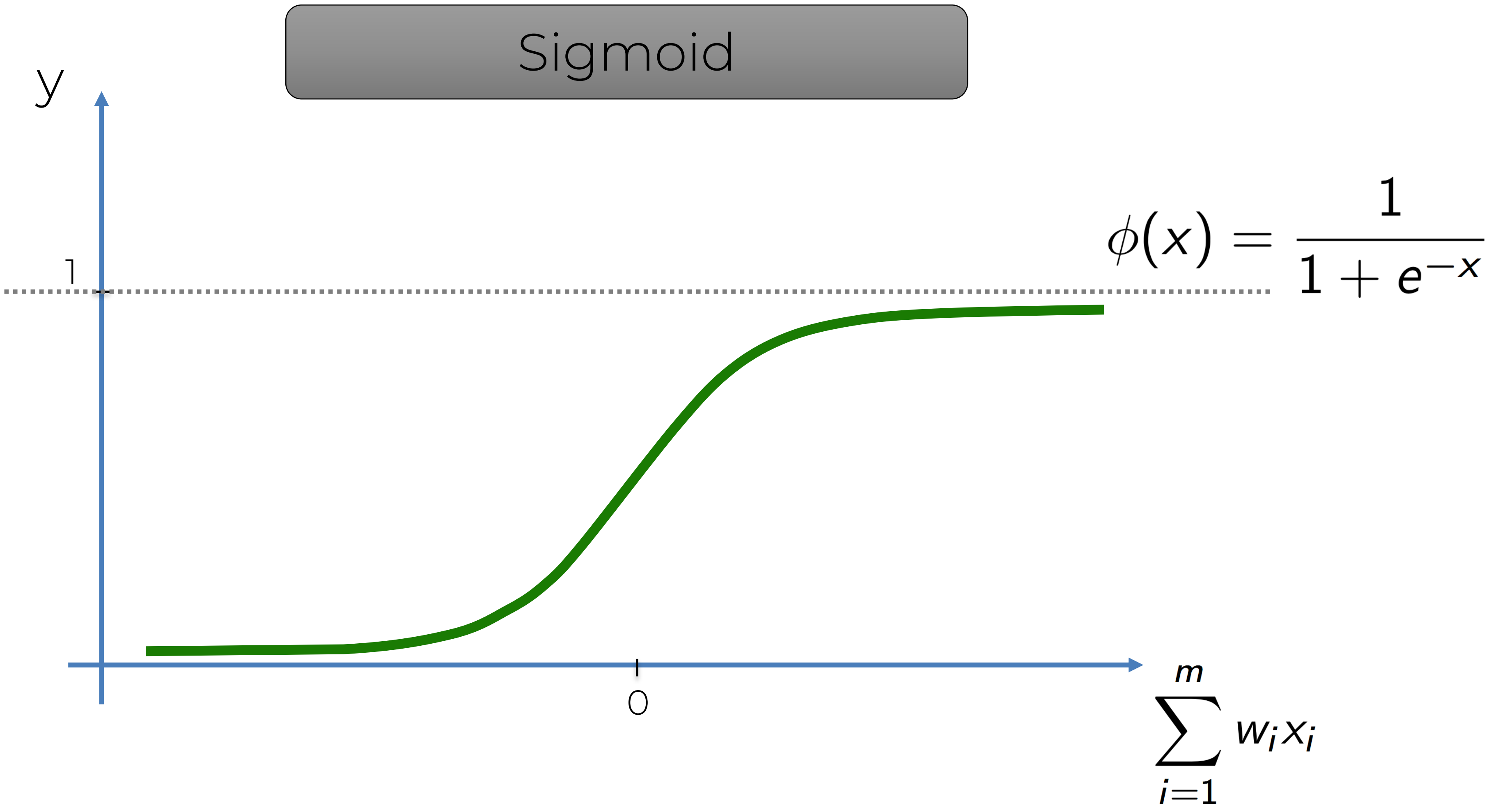
Threshold Function



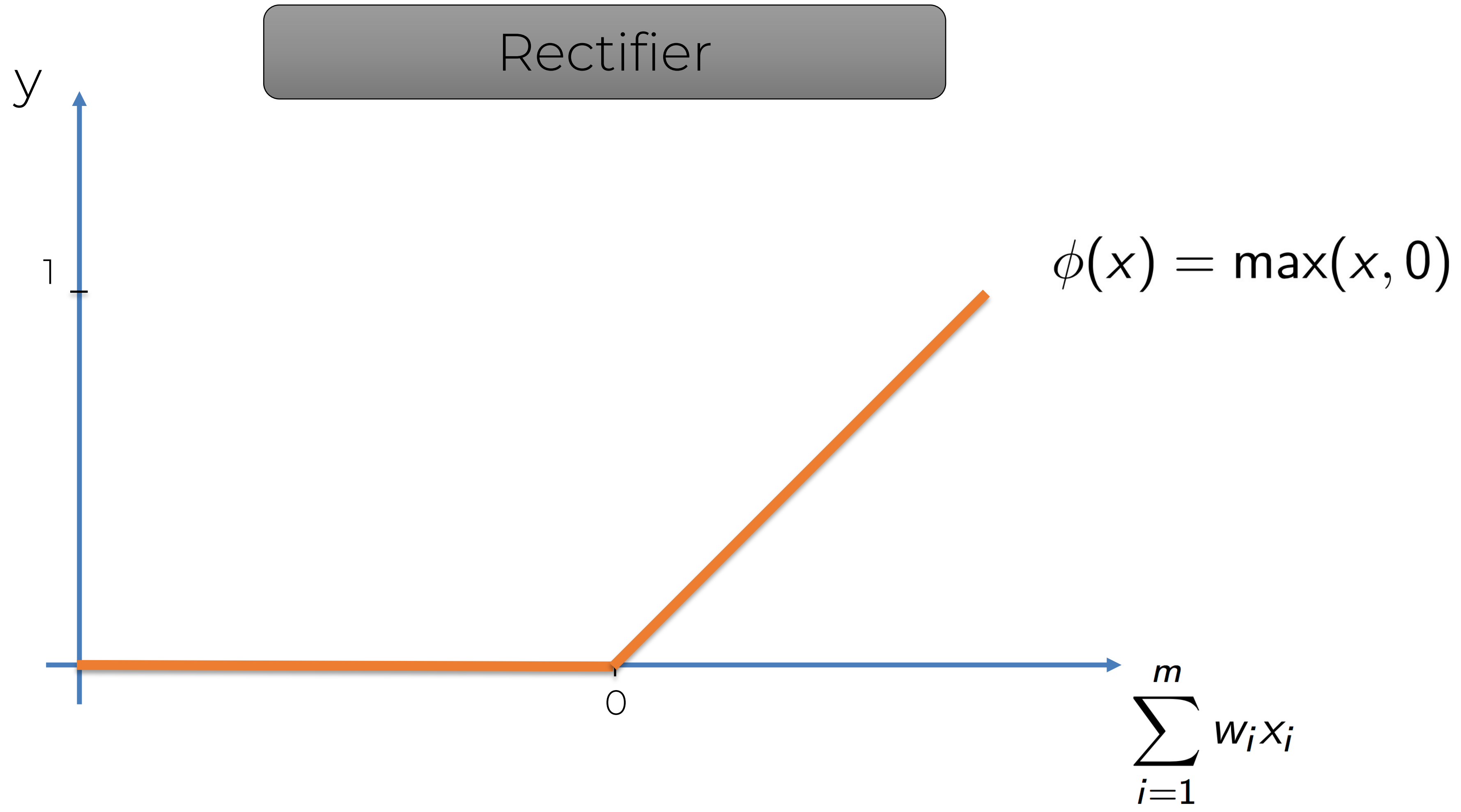
$$\phi(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

$$\sum_{i=1}^{m} w_i x_i$$

The Activation Function

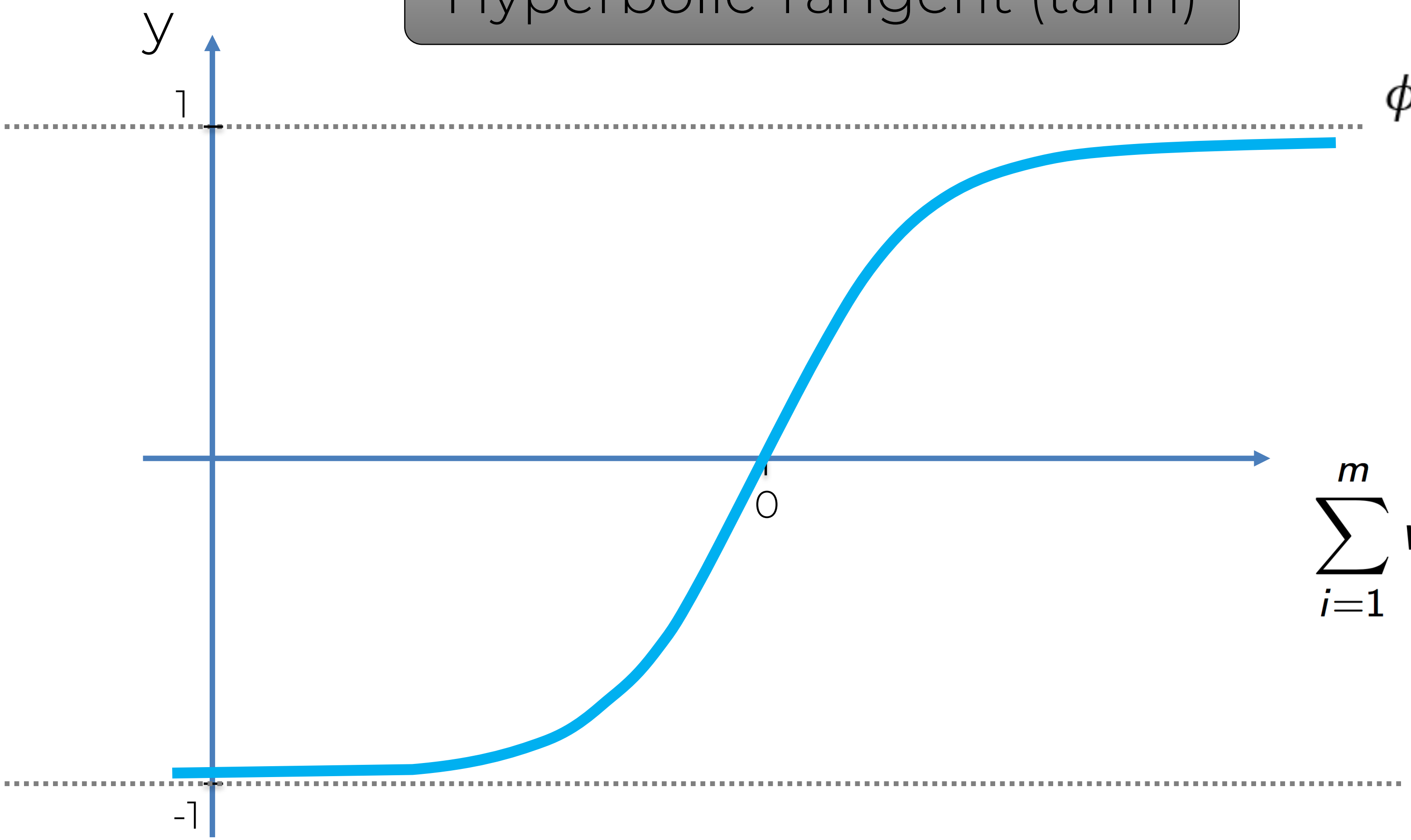


The Activation Function



The Activation Function

Hyperbolic Tangent (tanh)



$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

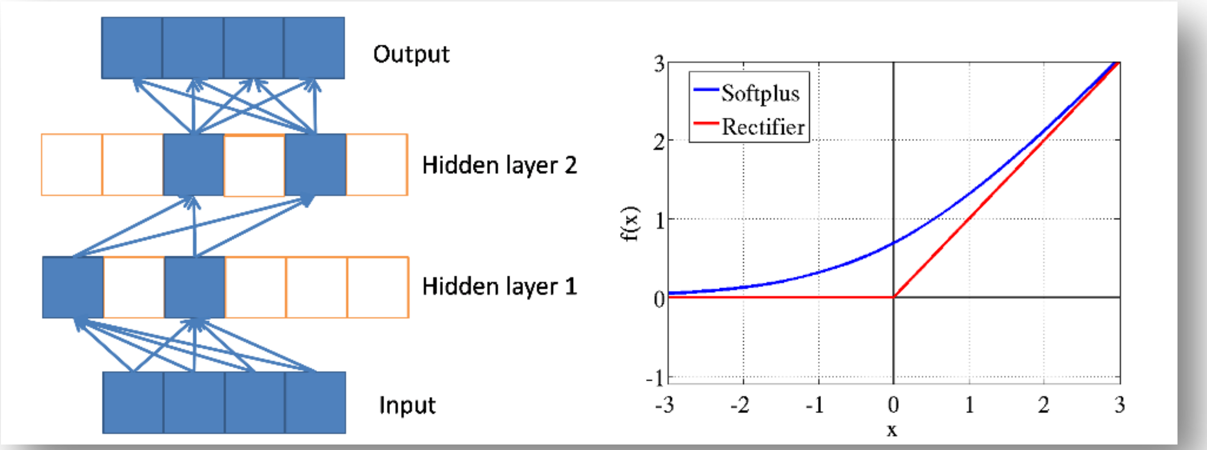
$$\sum_{i=1}^{m} w_i x_i$$

The Activation Function

Additional Reading:

Deep sparse rectifier neural networks

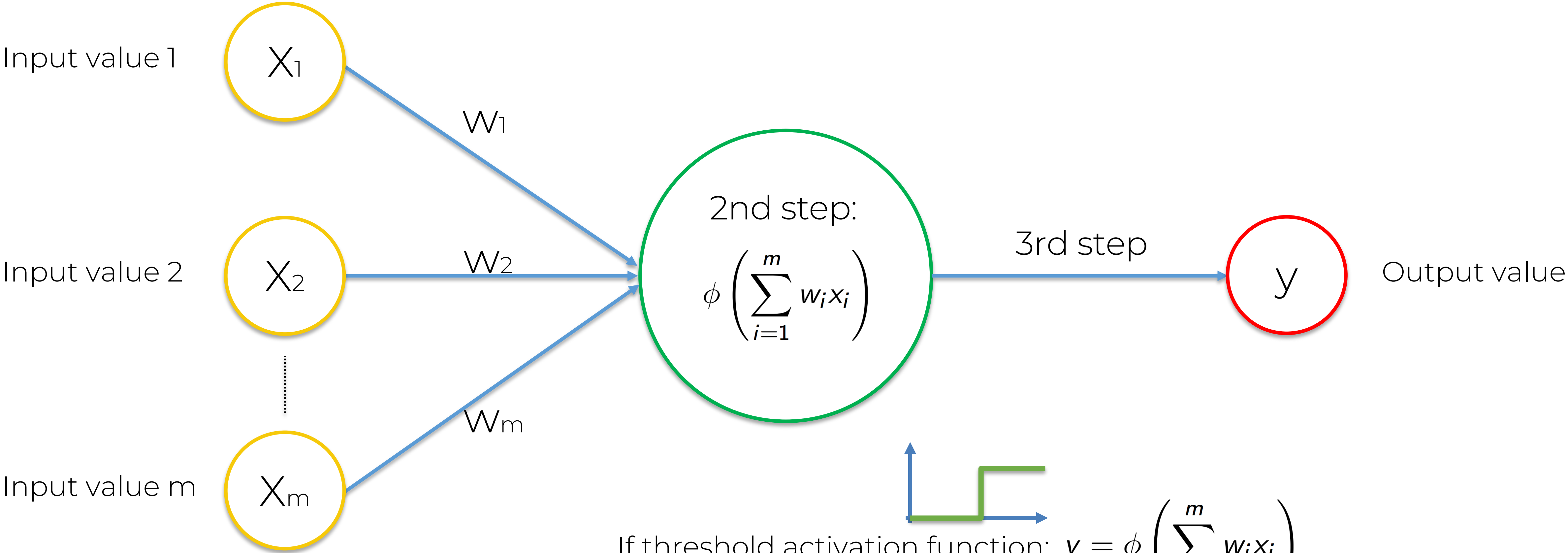
By Xavier Glorot et al. (2011)



Link:

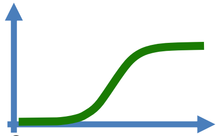
http://jmlr.org/proceedings/papers/v15/glorot11a/glorot11a.pdf

The Activation Function



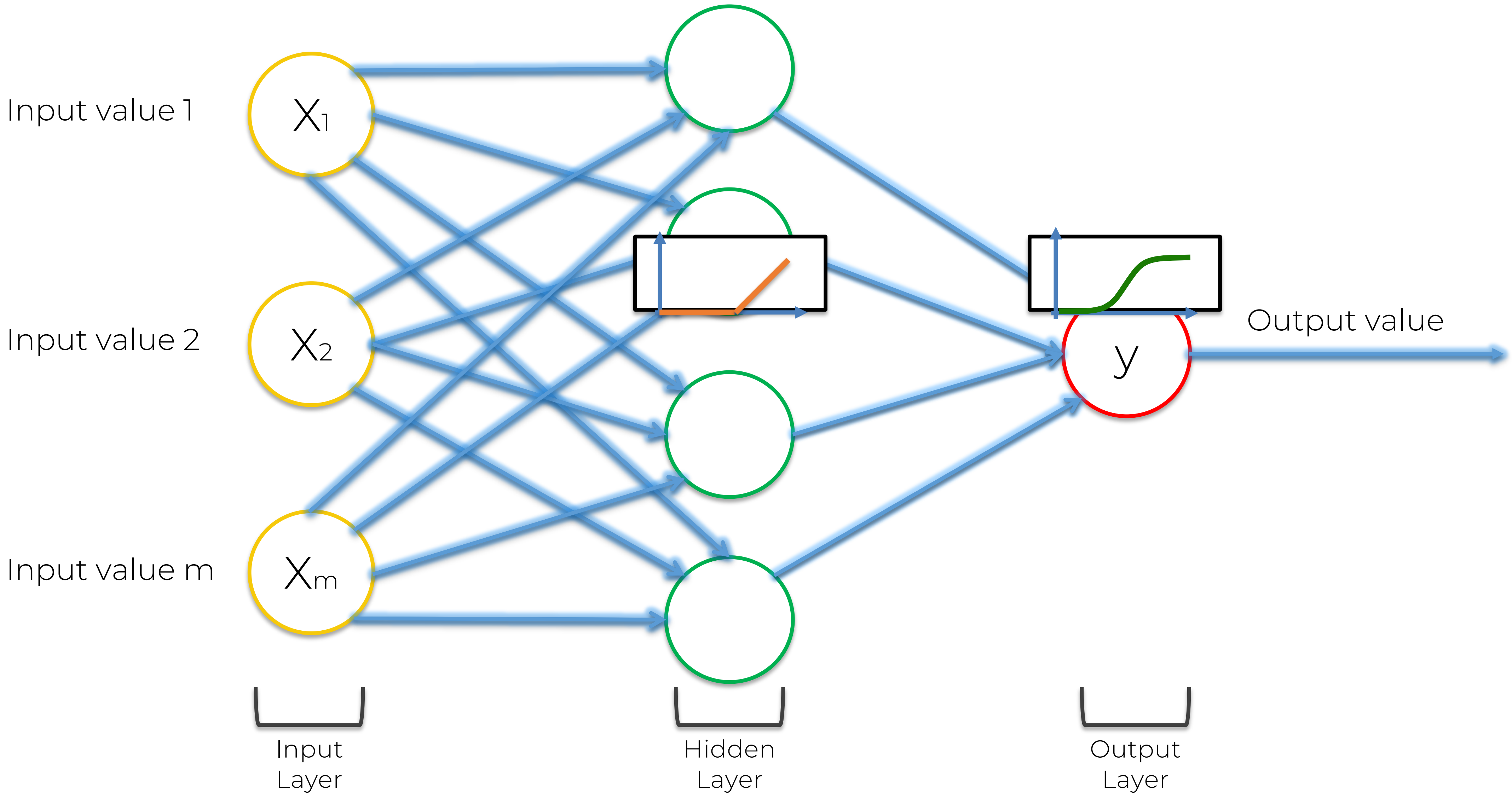
Assuming the DV is binary (y = 0 or 1)

If threshold activation function: $y = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$



If sigmoid activation function: $\mathbb{P}(y = 1) = \phi\left(\sum_{i=1}^{m} w_i x_i\right)$

The Activation Function



How do NNs Work?