BRAINTANK
DEEP LEARNING

Week 2: Diamonds are a Data Scientist's Best Friend

August 9, 2021

1. Neural Networks
2. Normalization
3. Training and Testing

Problem:

After the success of week one of BrainTank, figures in the tech world have been paying attention. A mysterious wealthy business-man has seen the amazing Cow-Predictor algorithm we developed last week, and now he wants us to use our deep learning ability to make an algorithm for him. He is a collector of rare diamonds and wants us to create a deep learning algorithm to be able to predict at what price he should sell his diamonds at. He has given us a list of information about all diamonds in his collection and he wants us to use all that information to make an algorithm that can predict price.

About the Dataset

7,845 Diamonds

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
…	…	…	…	…	…

About the Dataset

Feature 1: Shape

Diamond Shapes: {“Cushion”, “Emerald”, “Heart”, “Oval”, “Pear”, “Princess”, “Radiant”, “Round”}

About the Dataset

Feature 2: Carat

Diamond Carats: 0.19 Carats -> 5.6 Carats

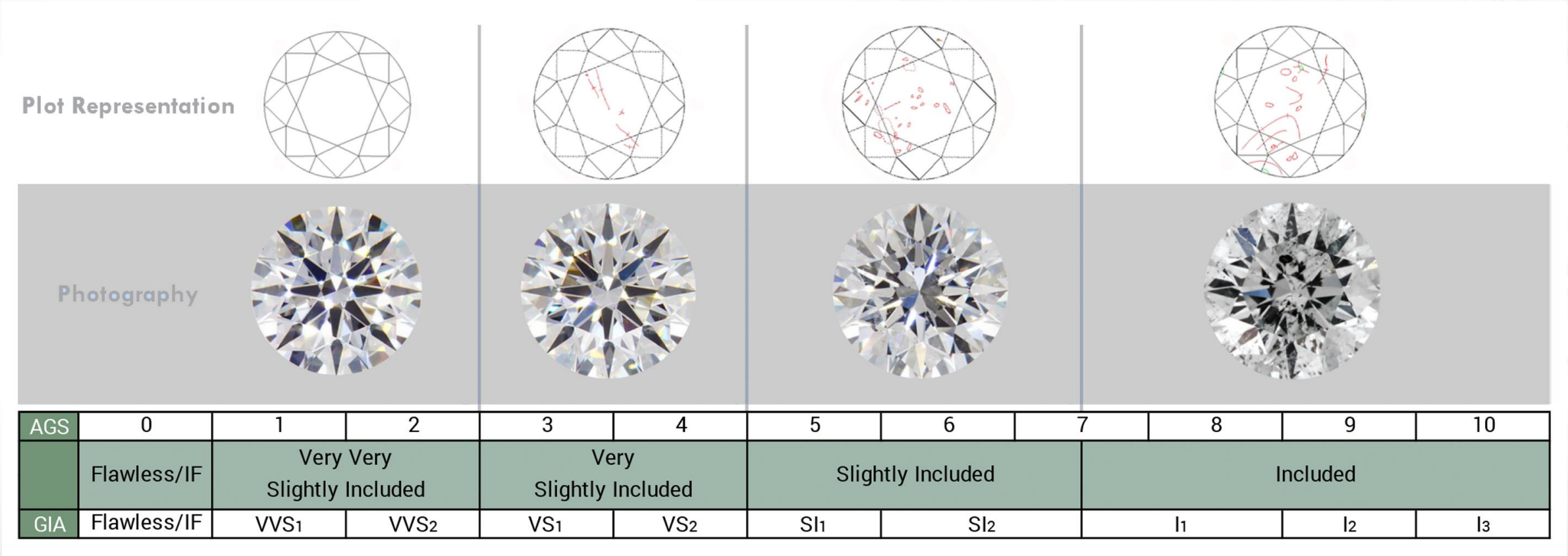
About the Dataset

Feature 3: Colour

Diamond Colours:{“D”, “E”, “F”, “G”, “H”, “I”, “J”}

About the Dataset

Feature 4: Clarity



AGS	0	1	2	3	4	5	6	7	8	9	10
	Flawless/IF	Very Very Slightly Included		Very Slightly Included		Slightly Included			Included		
GIA	Flawless/IF	VVS_1	VVS_2	VS_1	VS_2	SI_1	SI_2		I_1	I_2	I_3

Diamond Clarities: {“FL”, “IF”, “VVS1”, “VVS2”, “VS1”, “VS2”, “SL1”, “SL2”}

About the Dataset

Target: Cost

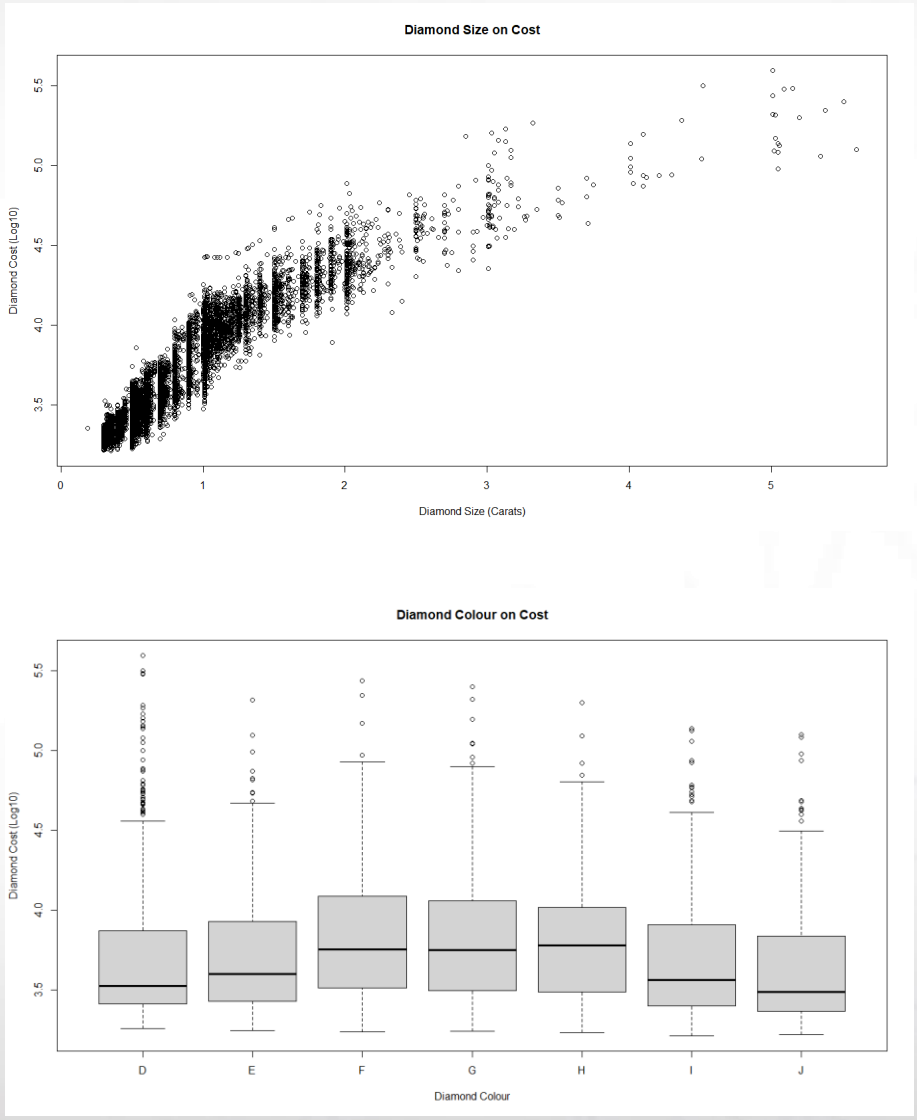
Diamond Example:	
Shape	“Round”
Carat	0.5
Colour	F
Clarity	VVS1

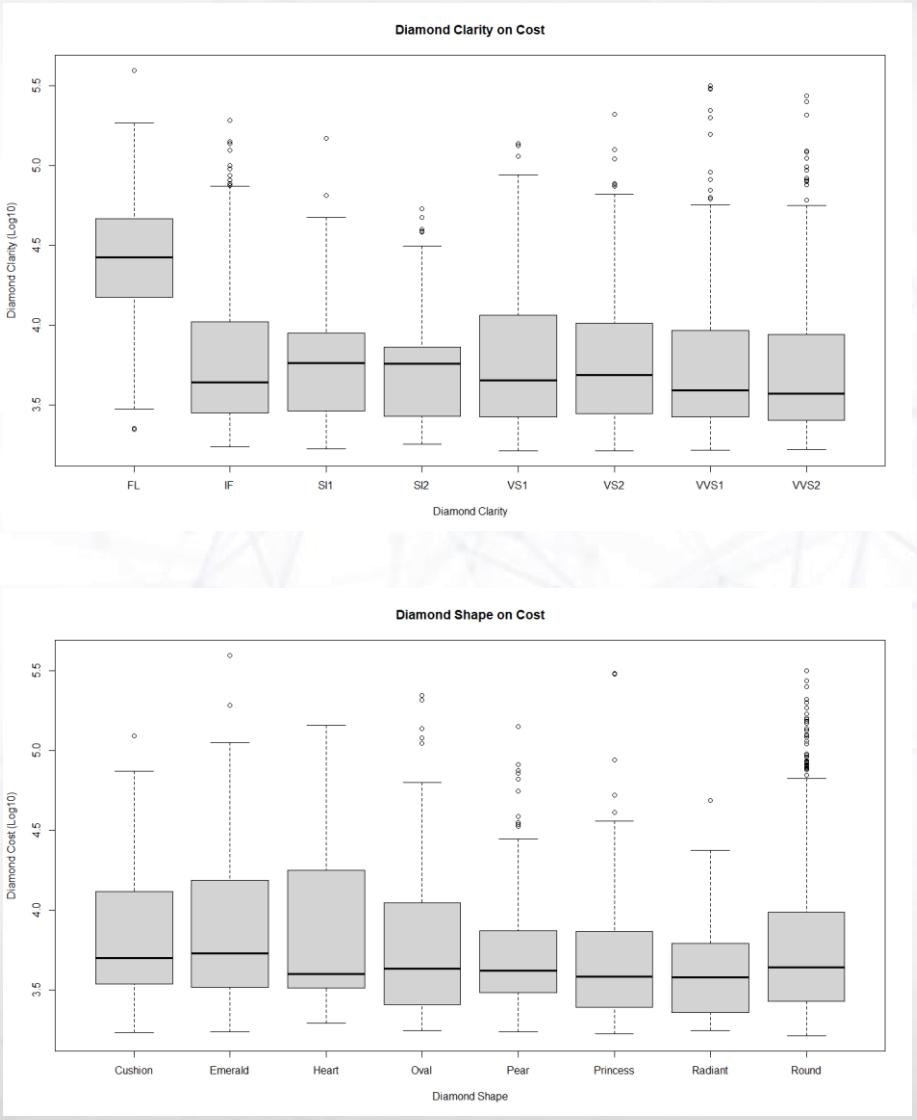
$3,745.16

Diamond Prices: $1,628.67 CAD -> $393,183.50 CAD

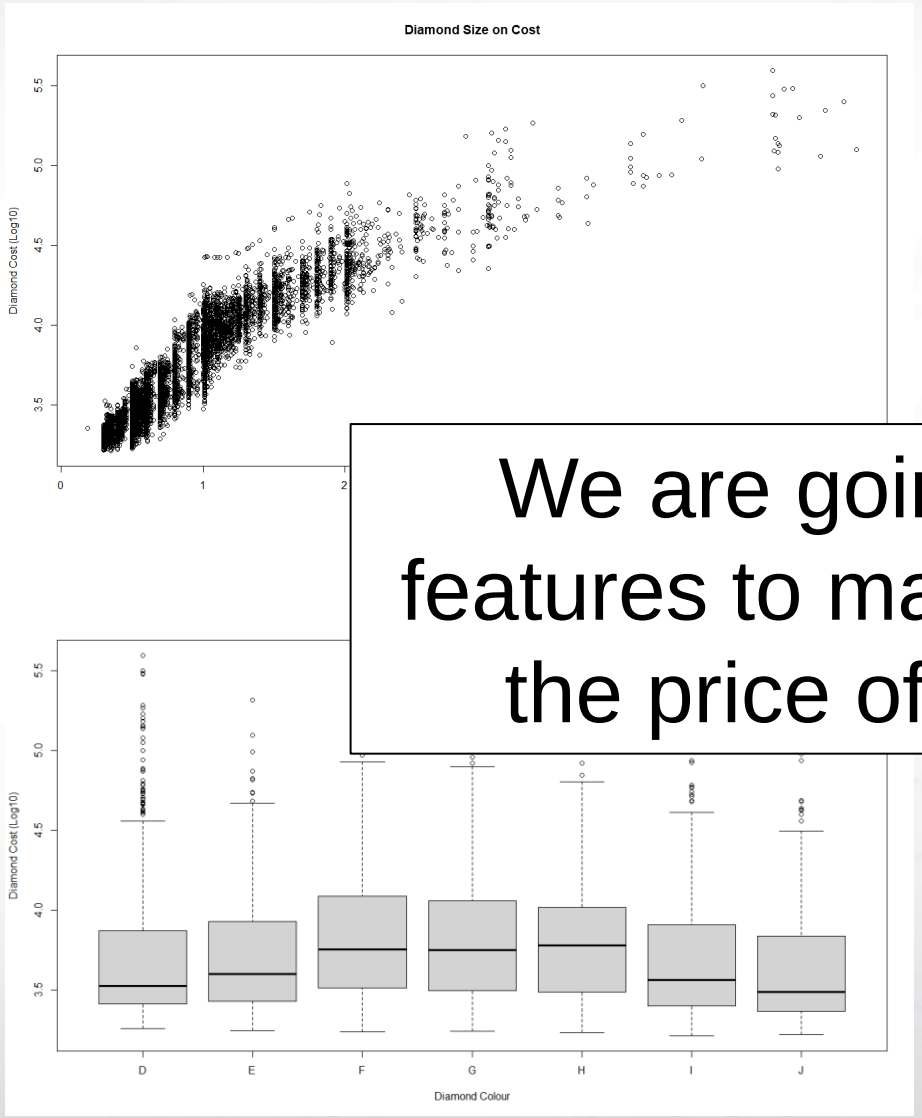
Shape, carat, colour, and clarity are features that determine cost

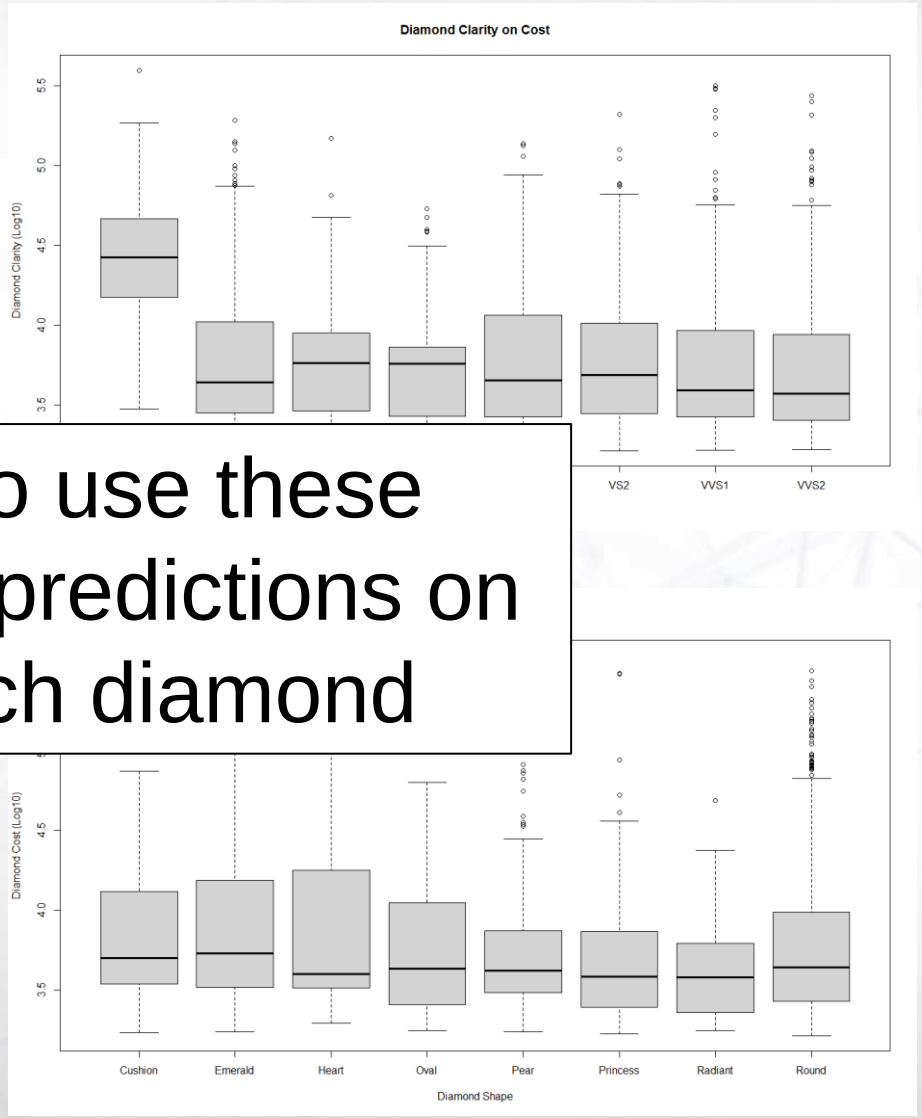
About the Dataset





About the Dataset





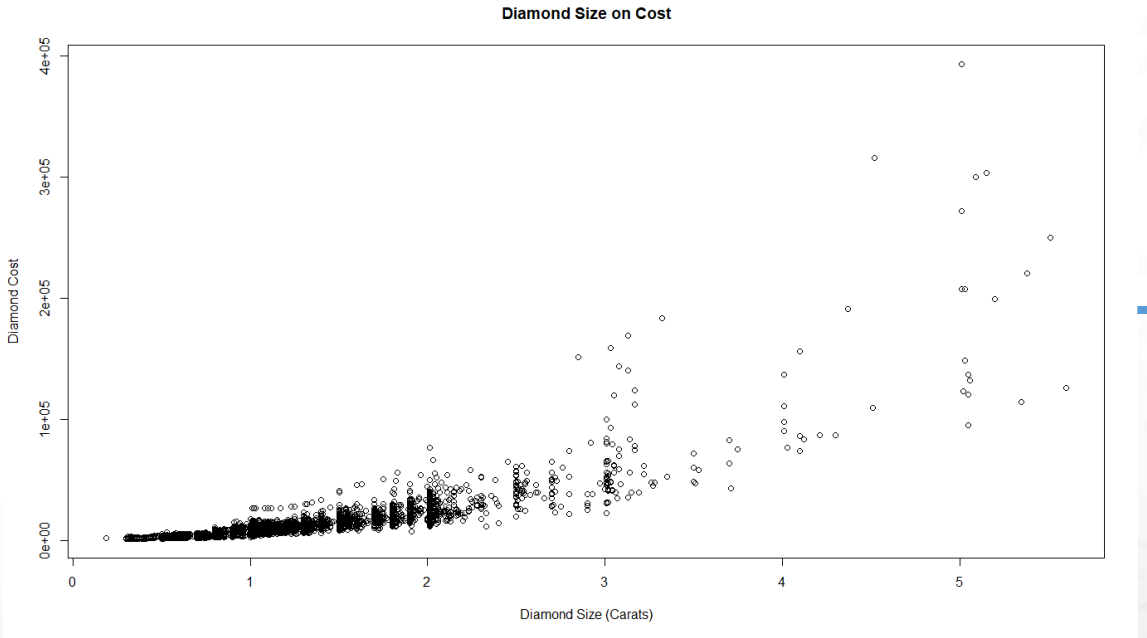
We are going to use these features to make predictions on the price of each diamond

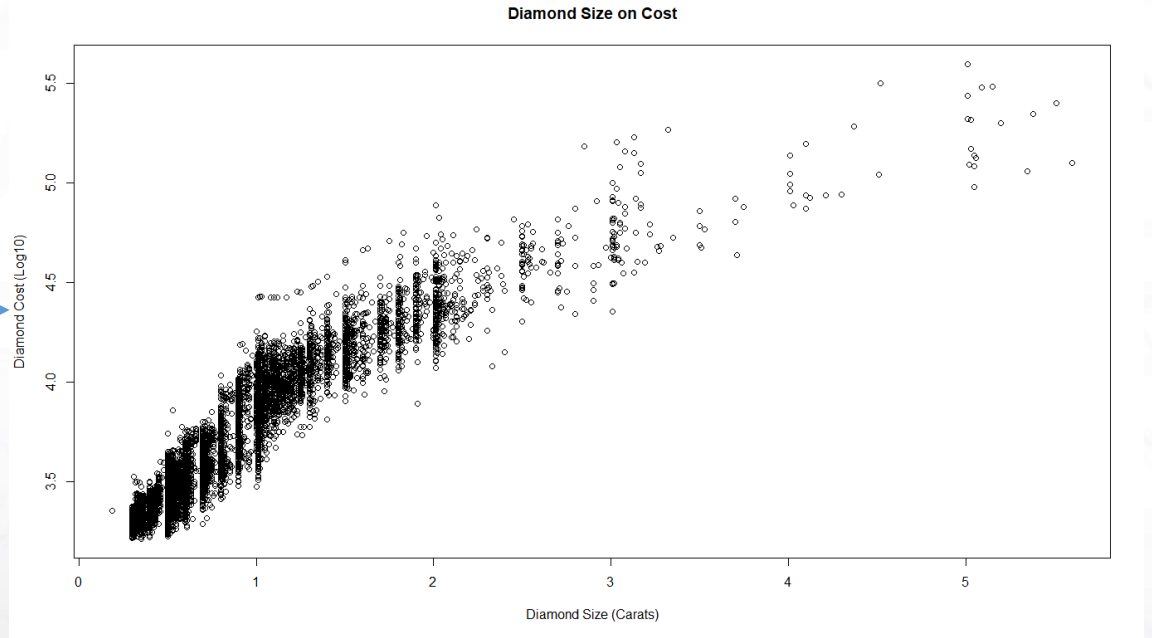
Morphing the dataset:

BRAINTANK
DEEP LEARNING

1. Adjust variables so they have a linear nature
2. Normalize data

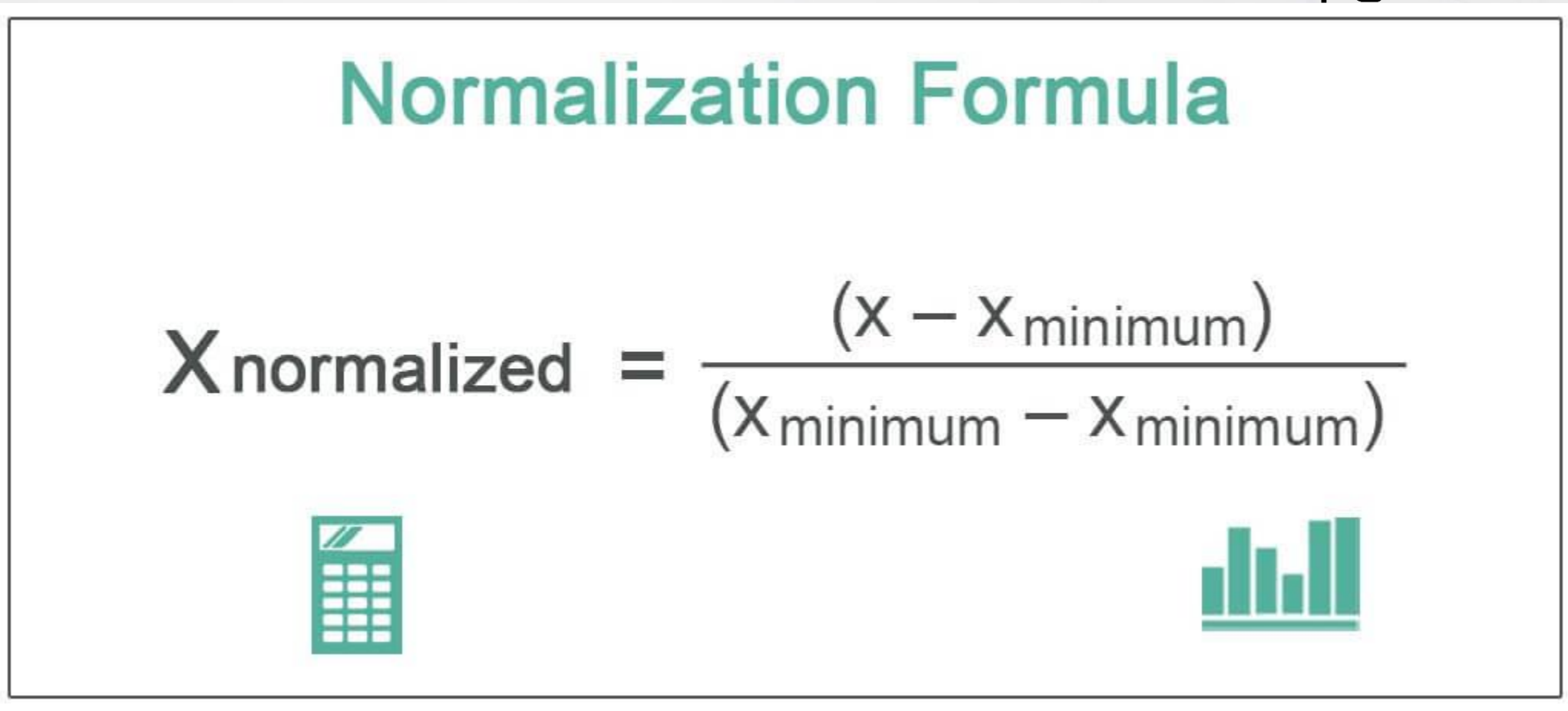
Adjust variables so are linear:



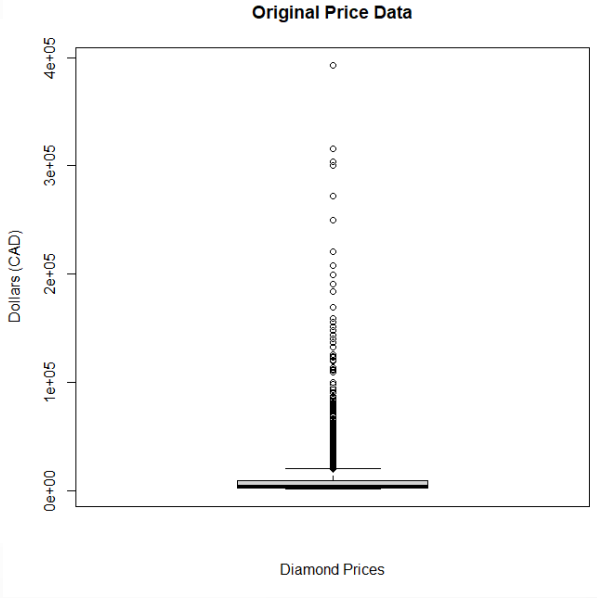


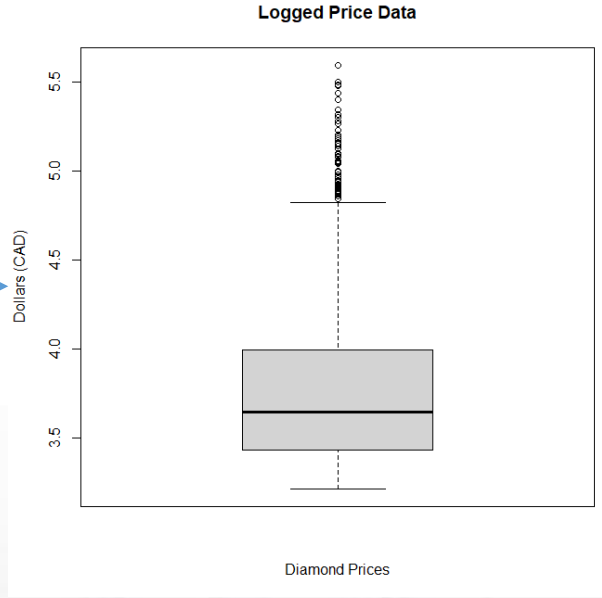
Our Prices data has an exponential curve. We want to adjust that so values can fall more evenly between their minimum and maximum values. We doing this by taking the log10() of each price

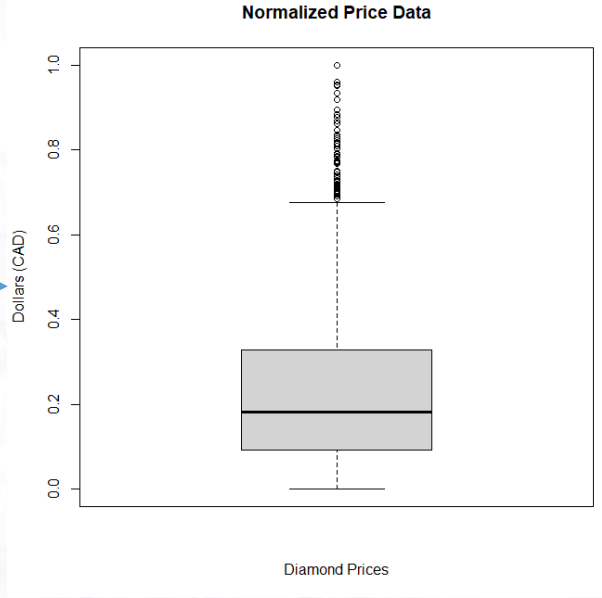
Normalize Data:

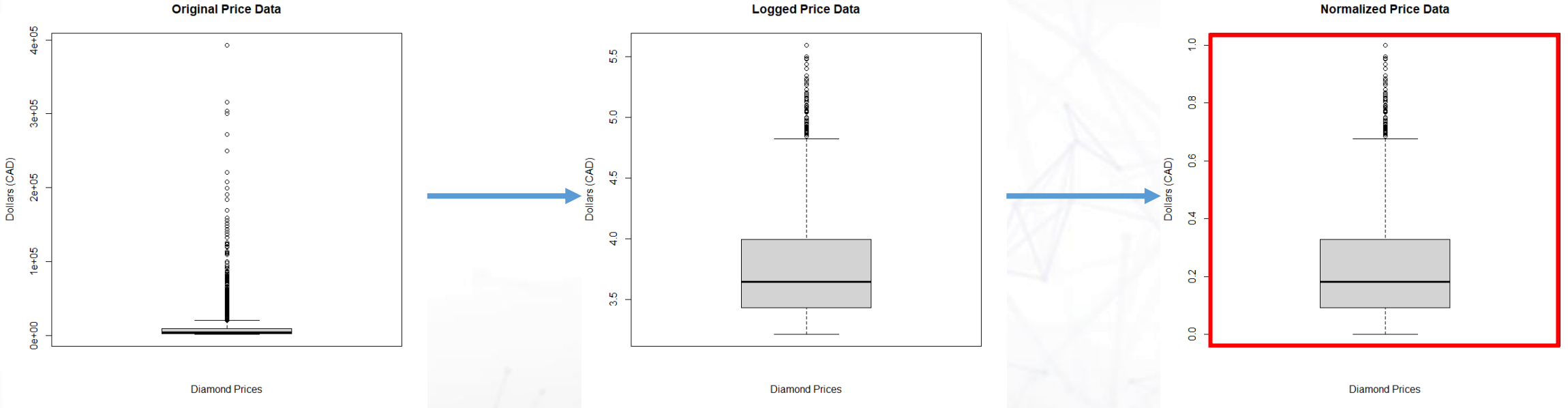


We want all of our features and our target to fall in between 0 and 1. We can achieve that by using this formula. We do this because Neural Networks do not like big numbers

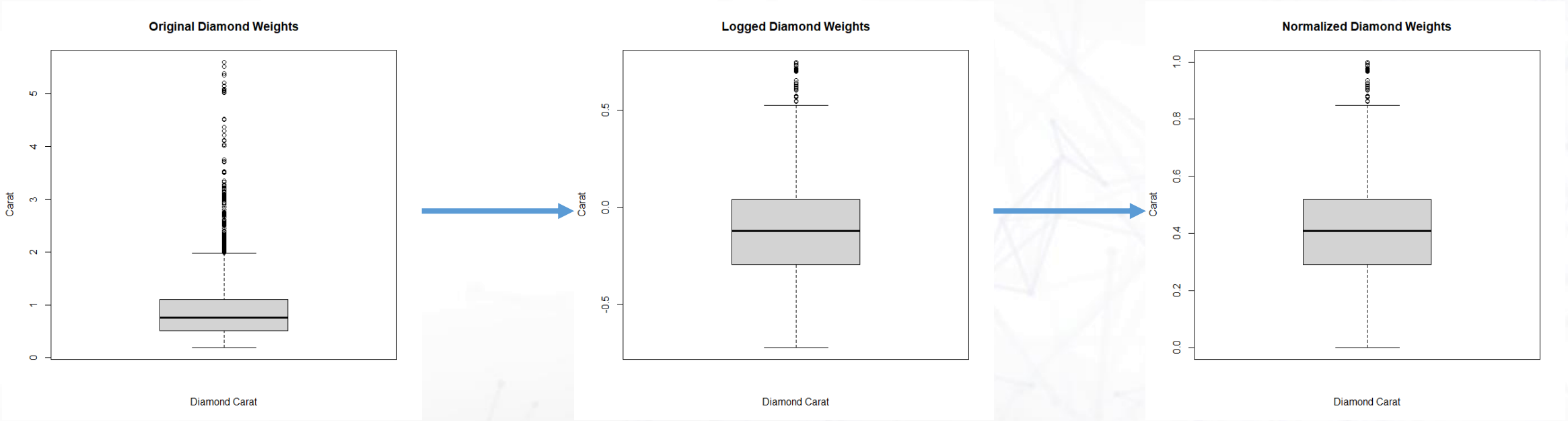

Original Price Data
Dollars (CAD)
0e+00
1e+05
2e+05
3e+05
4e+05
Diamond Prices

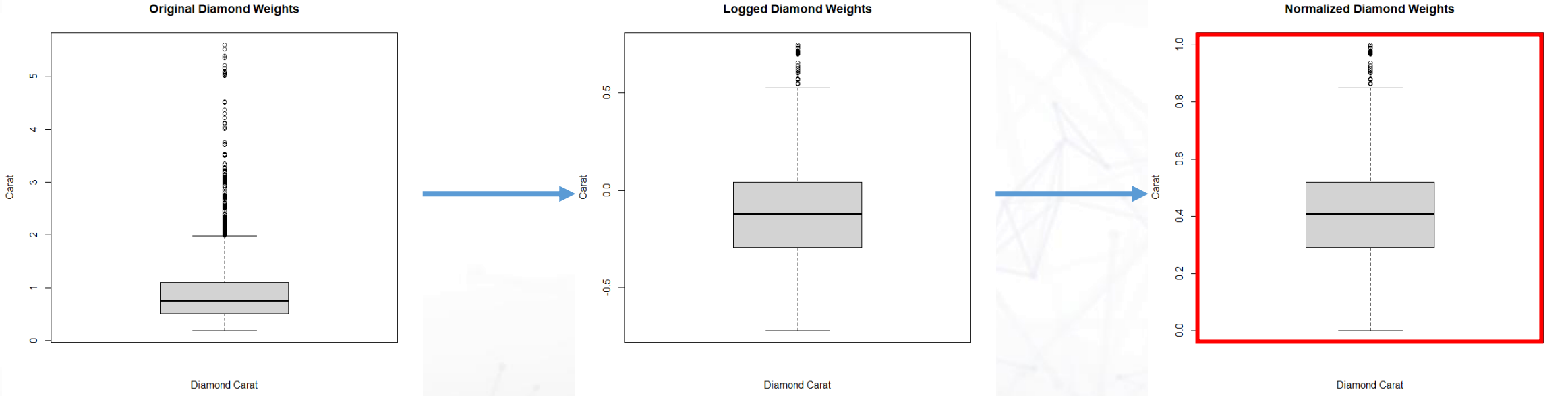

Logged Price Data
Dollars (CAD)
3.5
4.0
4.5
5.0
5.5
Diamond Prices


Normalized Price Data
Dollars (CAD)
0.0
0.2
0.4
0.6
0.8
1.0
Diamond Prices

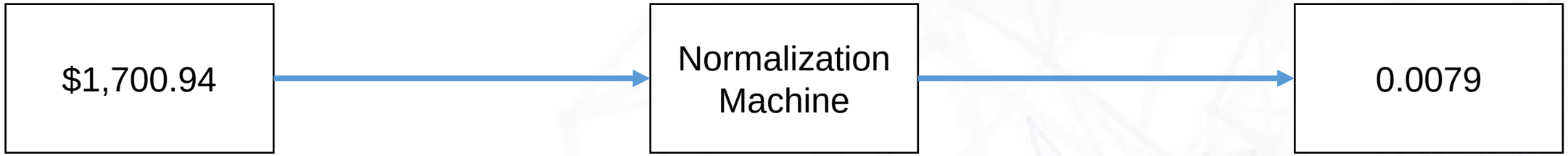


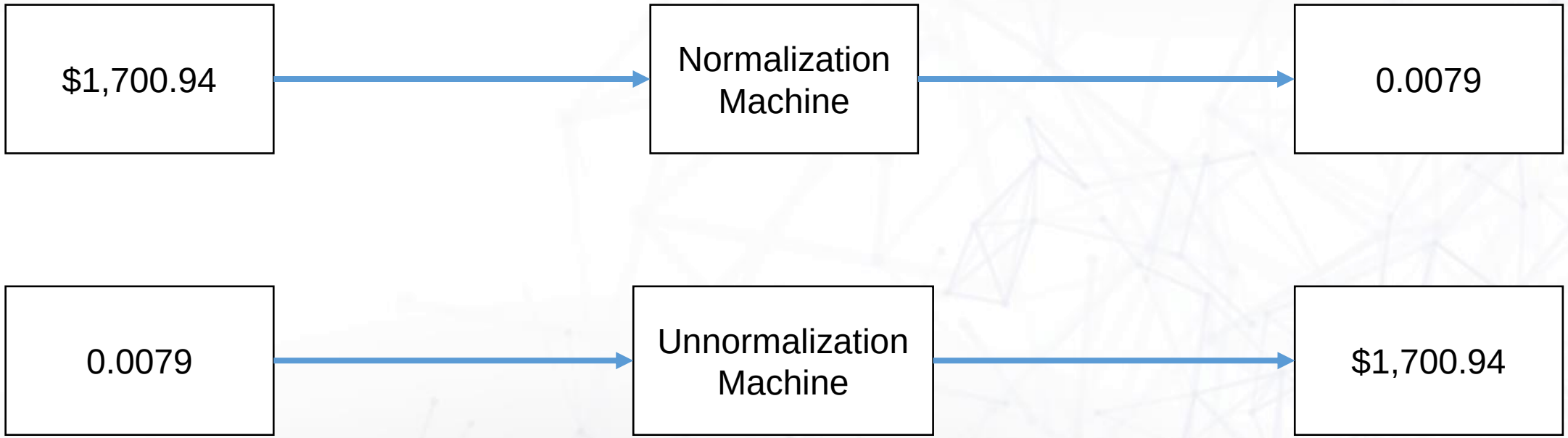
The left and the right sets of data contain the exact same information, but the normalized data is easier for the model to use

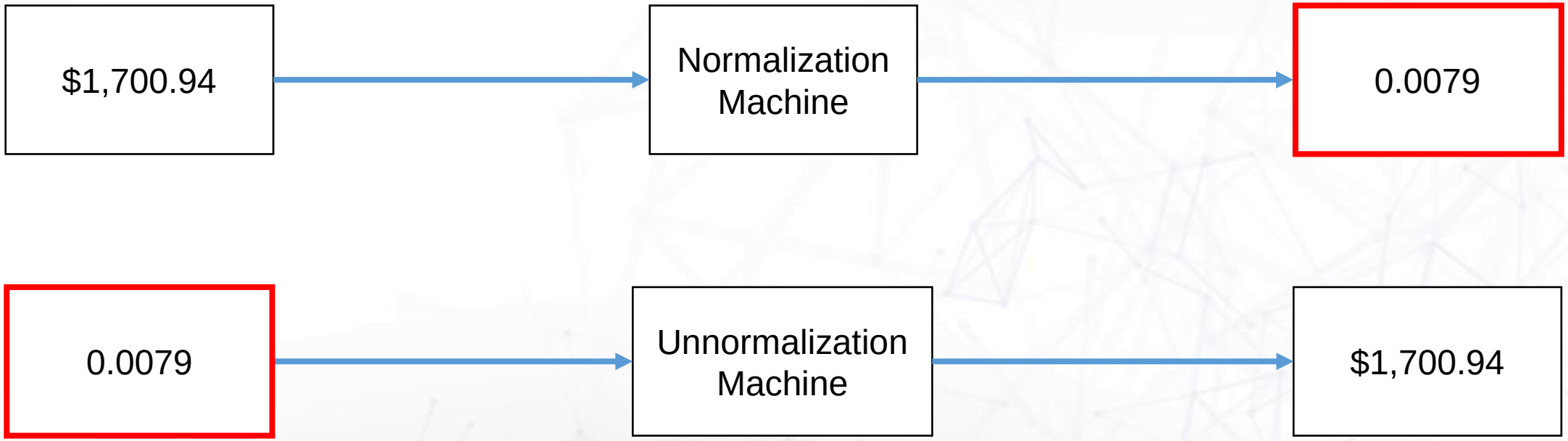

Original Diamond Weights
Carat
0
1
2
3
4
5
Diamond Carat
Logged Diamond Weights
Carat
-0.5
0.0
0.5
Diamond Carat
Normalized Diamond Weights
Carat
0.0
0.2
0.4
0.6
0.8
1.0
Diamond Carat



The left and the right sets of data contain the exact same information, but the normalized data is easier for the model to use


$1,700.94
Normalization Machine
0.0079


$1,700.94
Normalization Machine
0.0079
0.0079
Unnormalization Machine
$1,700.94



The normalized values have properties that neural networks love, while it is difficult for neural networks to work with large exponential values like our prices are.

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
…	…	…	…	…	…

1 to 10 of 7845 entries Filter

▲	Price	Carat	D	E	F	G	H	I	J	FL	IF	VVS1	VVS2	VS1	VS2	SI1	SI2	Cushion	Emerald	Heart	Oval	Pear	Princess	Radiant	Round
0	0.007913461796532396	0.05730129390018485	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
1	0.03399082776760116	0.05730129390018485	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0.05339124909704686	0.05730129390018485	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
100	0.19117304805782878	0.1515711645101664	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
1000	0.18872036940097583	0.11275415896487989	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1001	0.16070730422465399	0.11275415896487989	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
1002	0.13484110407479374	0.11275415896487989	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
1003	0.15651729968338313	0.11275415896487989	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
1004	0.18236210618944565	0.11645101663585952	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
1005	0.189282574308784	0.12014787430683918	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Show 10 per page

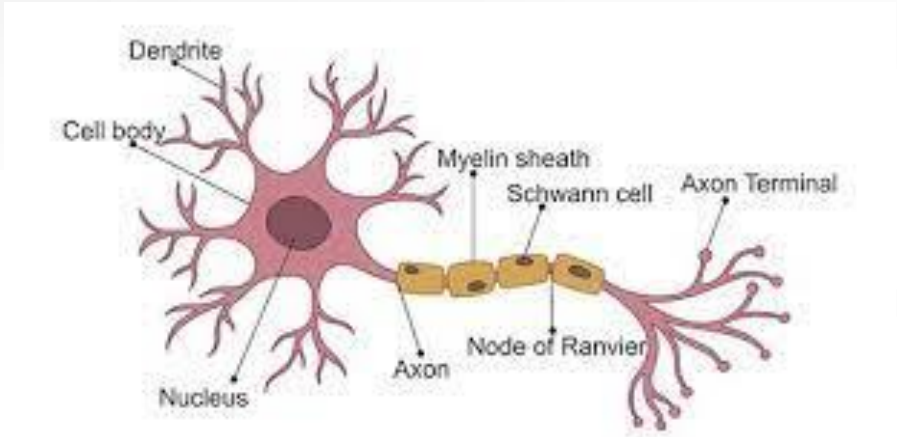
1 2 10 100 700 780 785

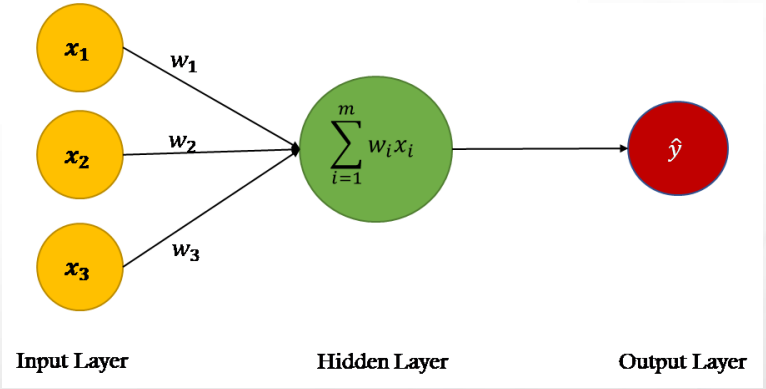
Target

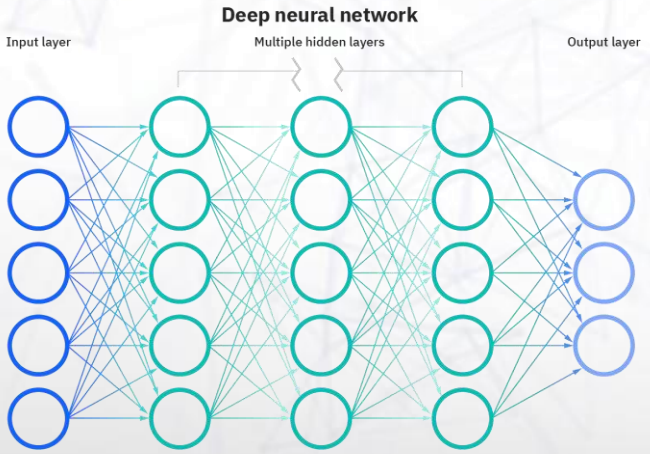
24 x Features

Neural Networks

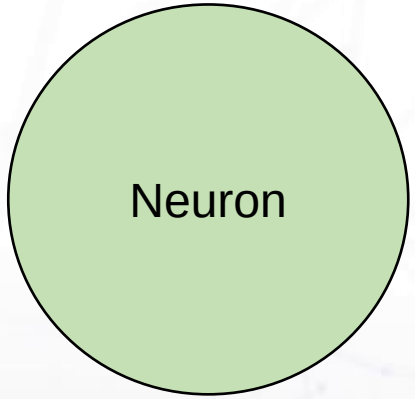
Neural Networks







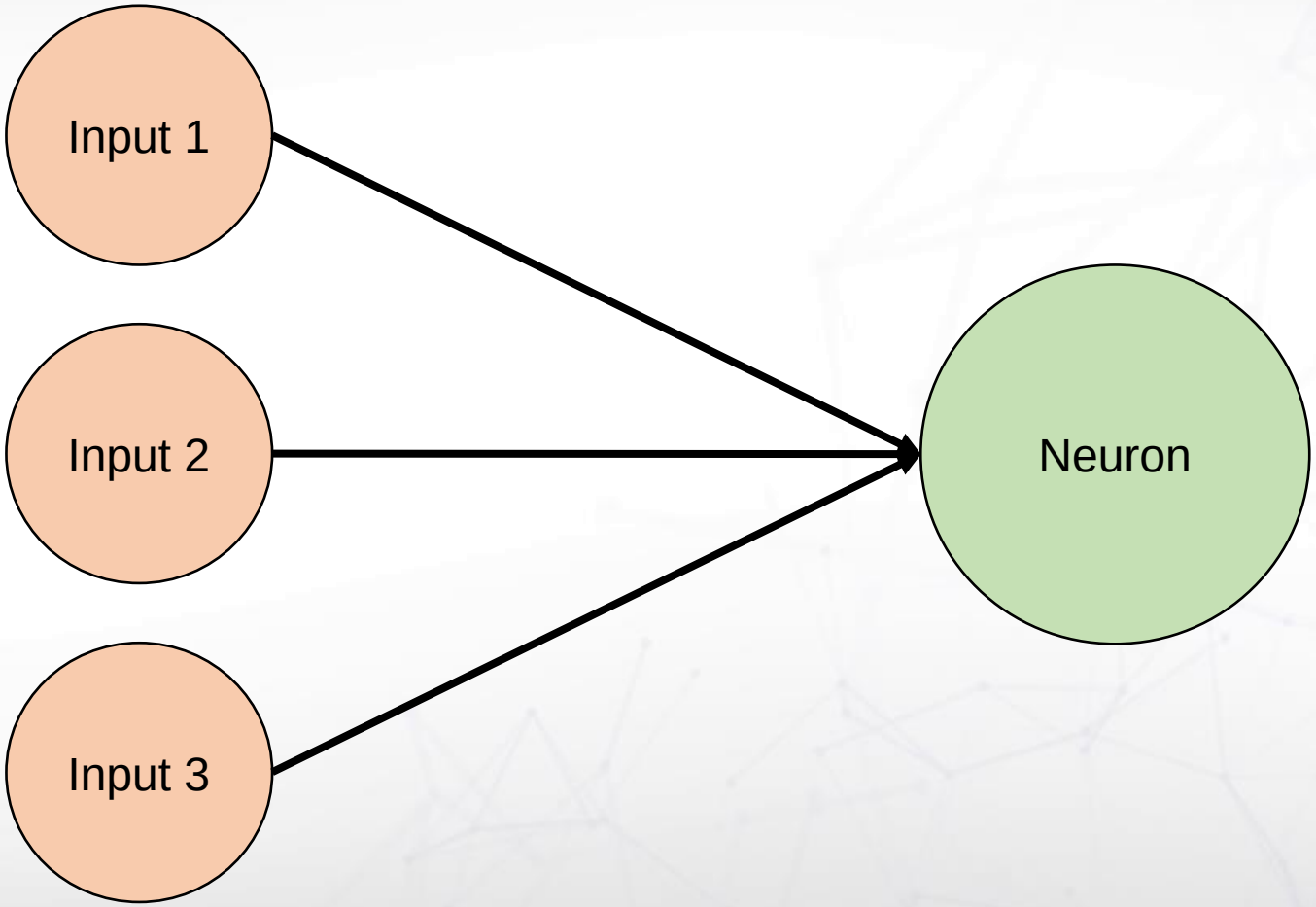
The Neuron



Parts of a Neural Network

1. Neuron

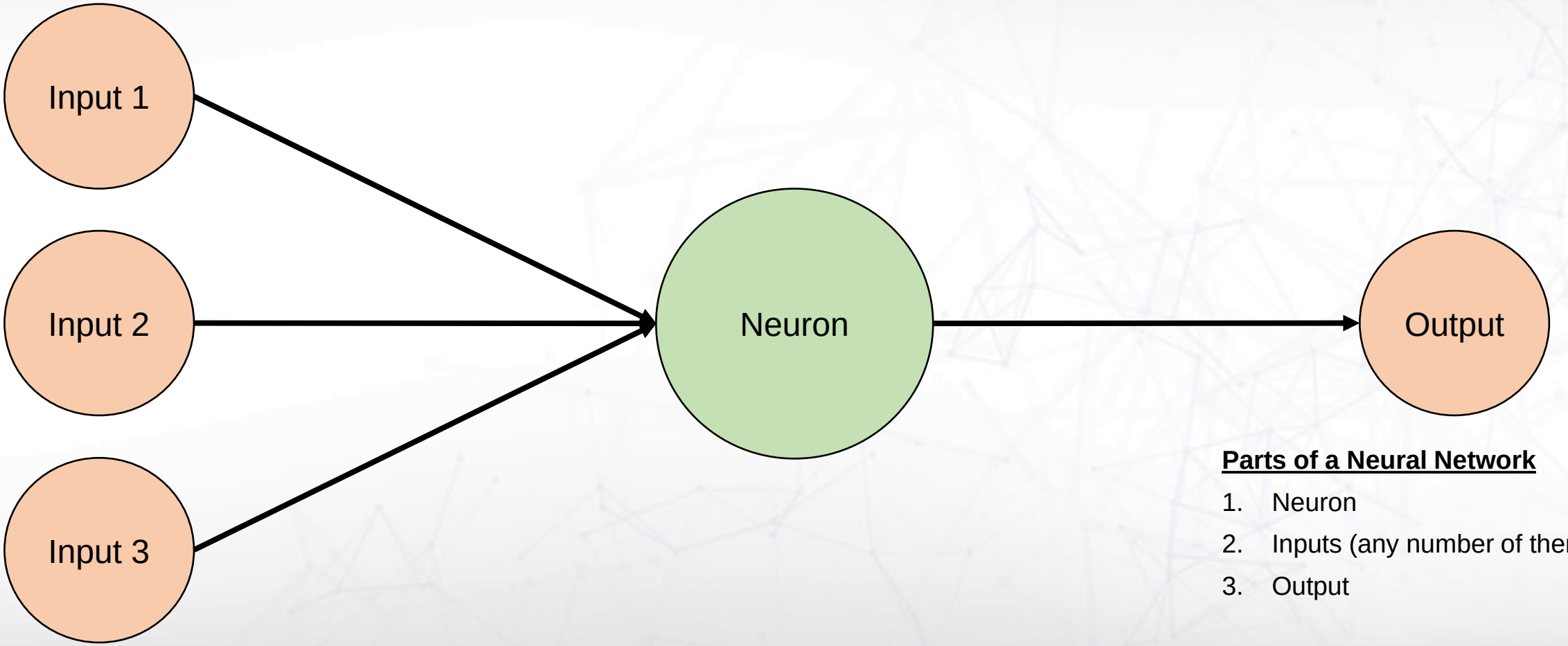
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)

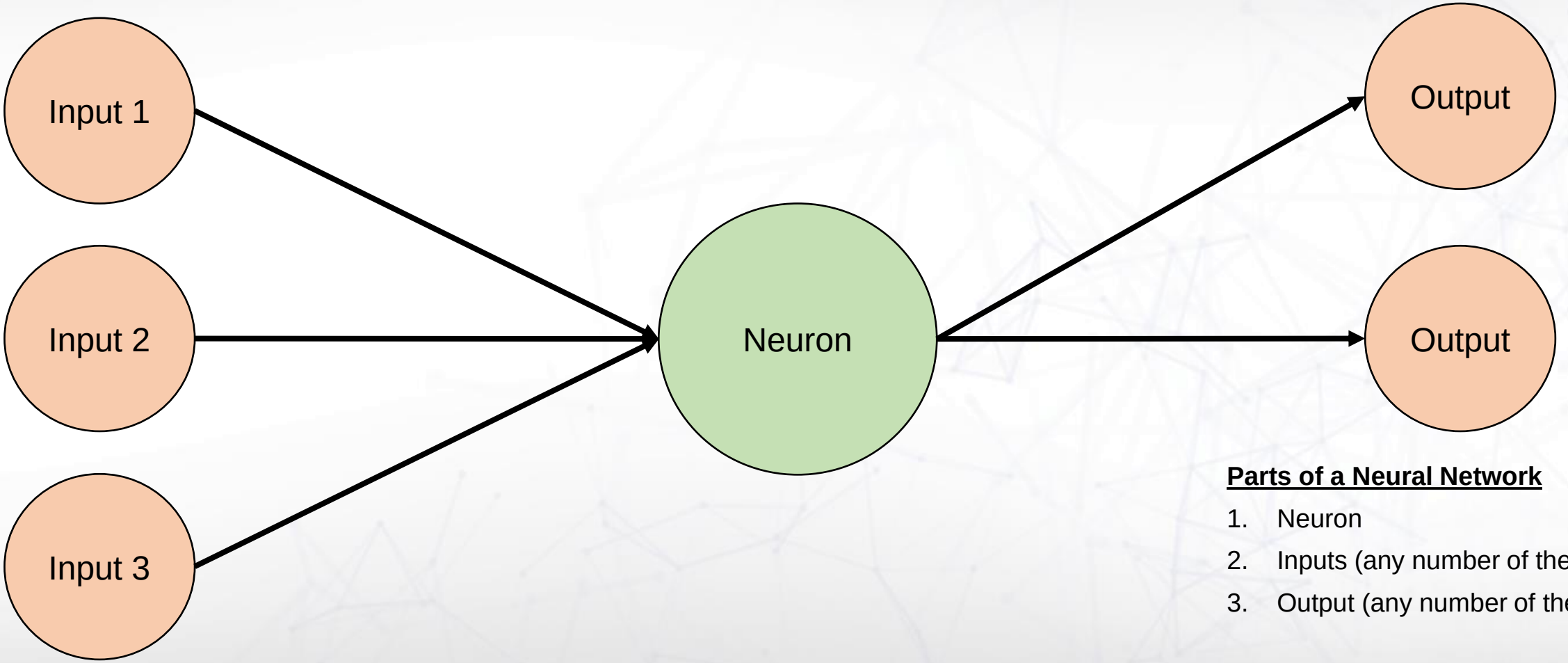
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output

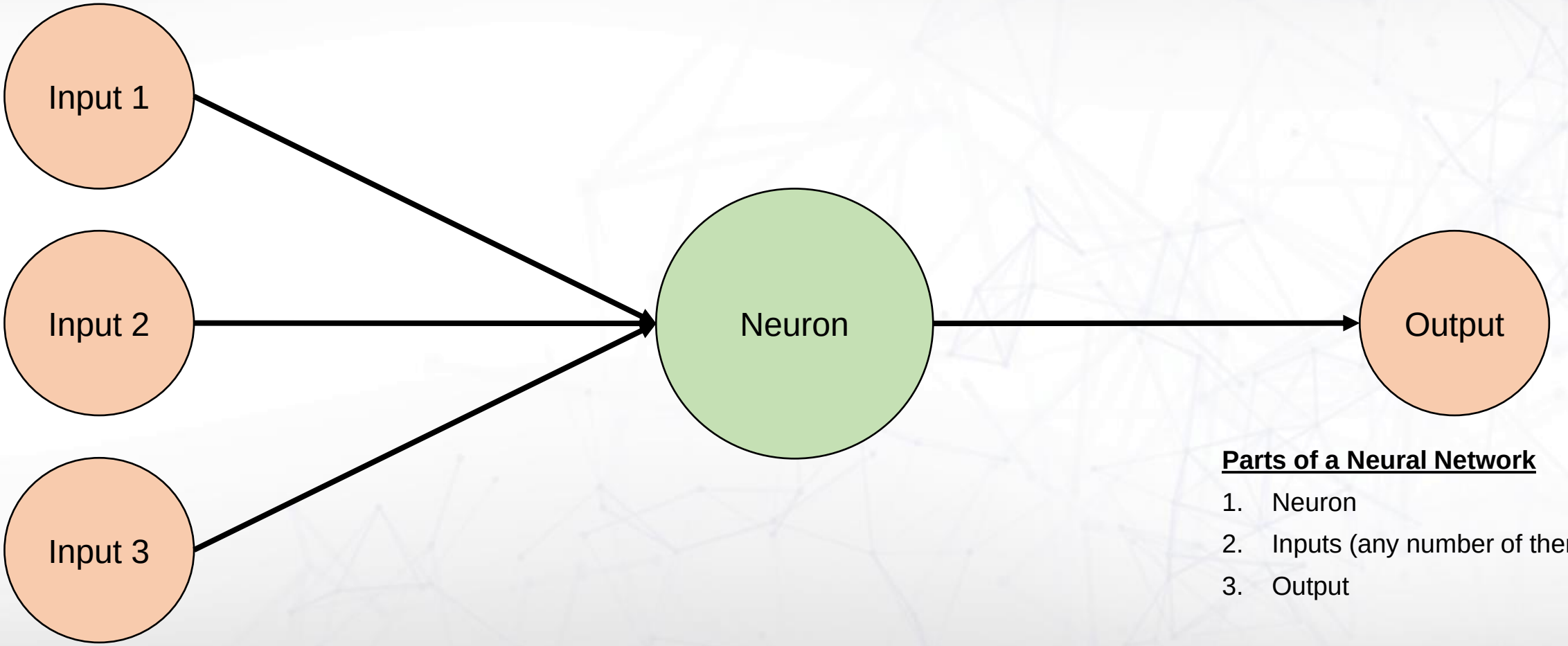
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output (any number of them)

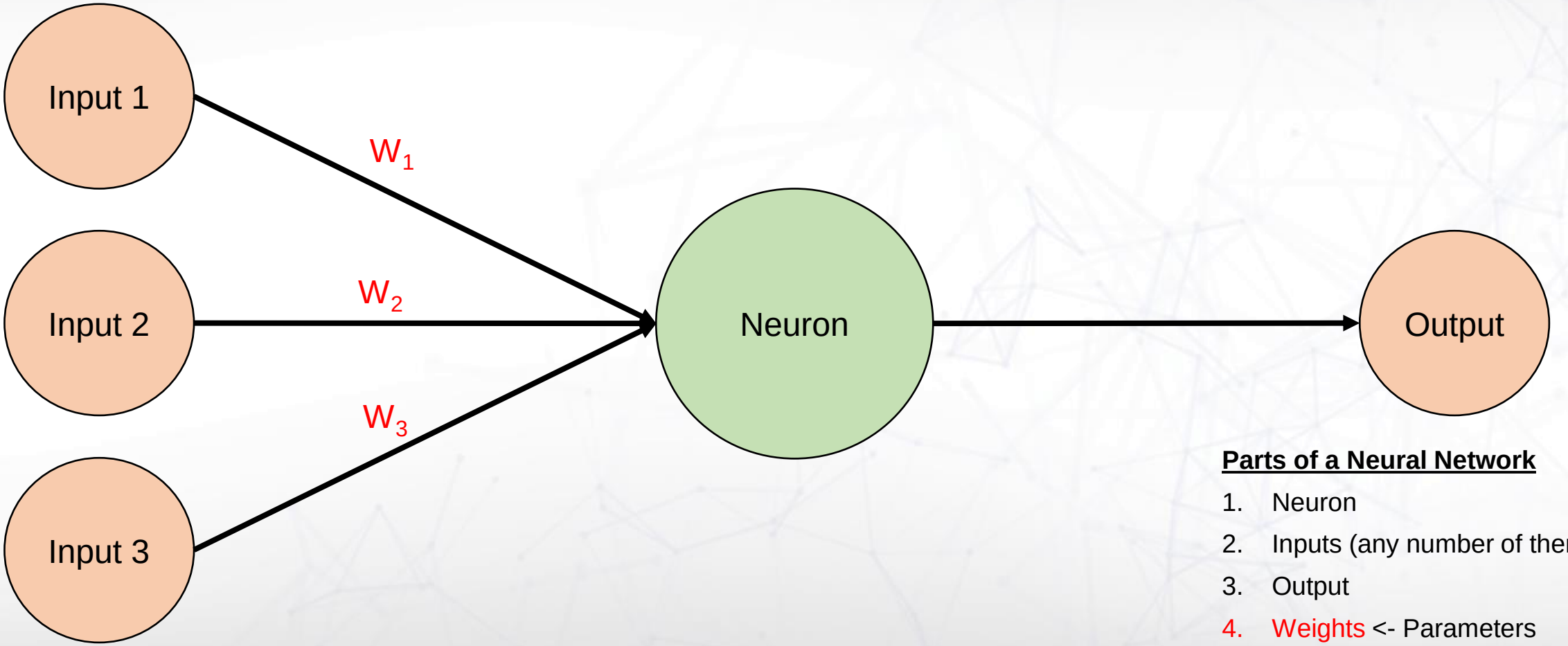
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output

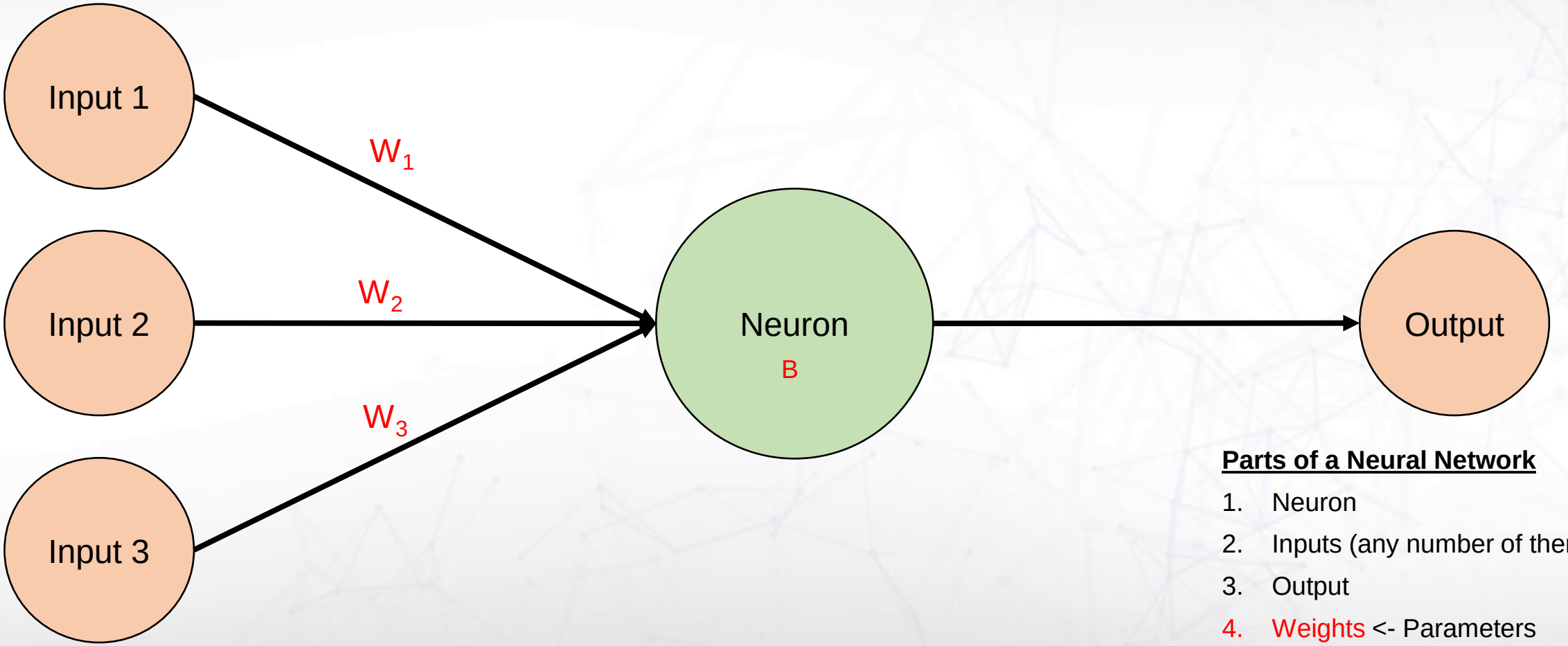
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters

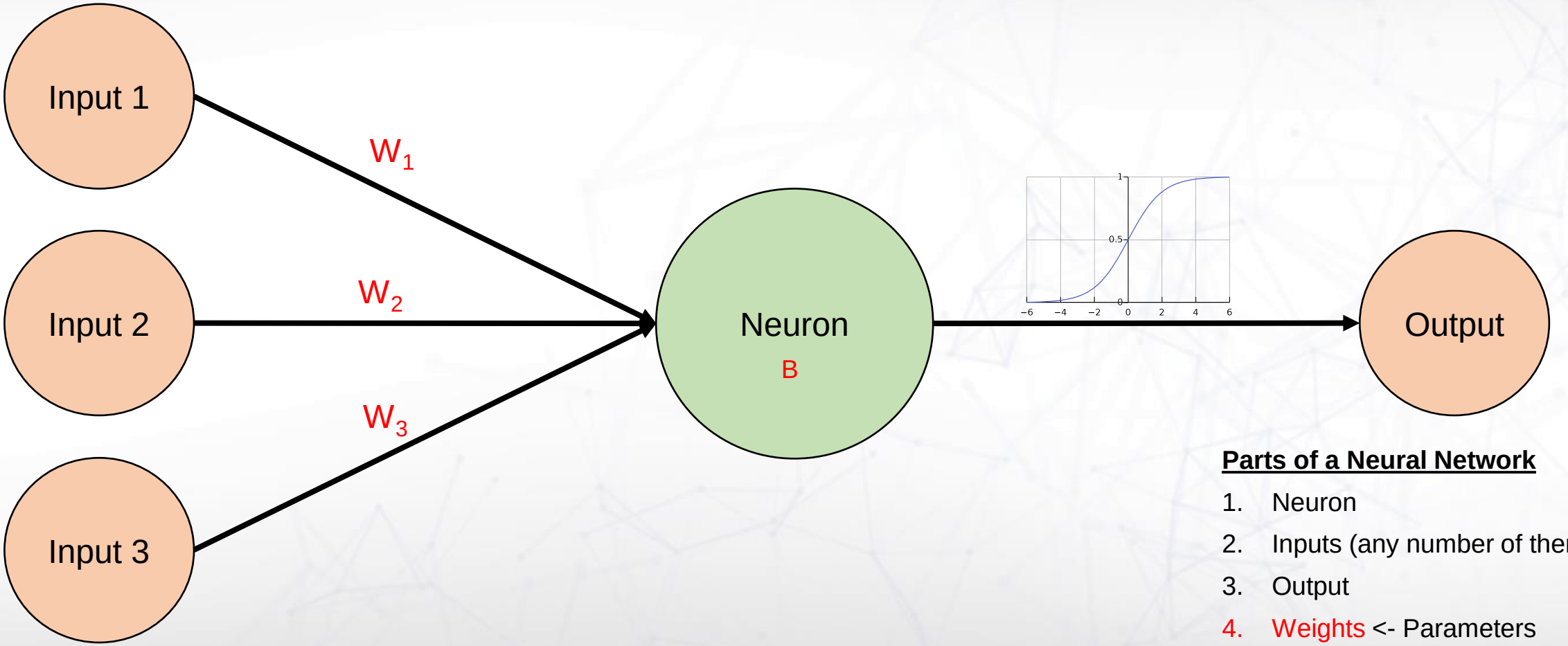
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter

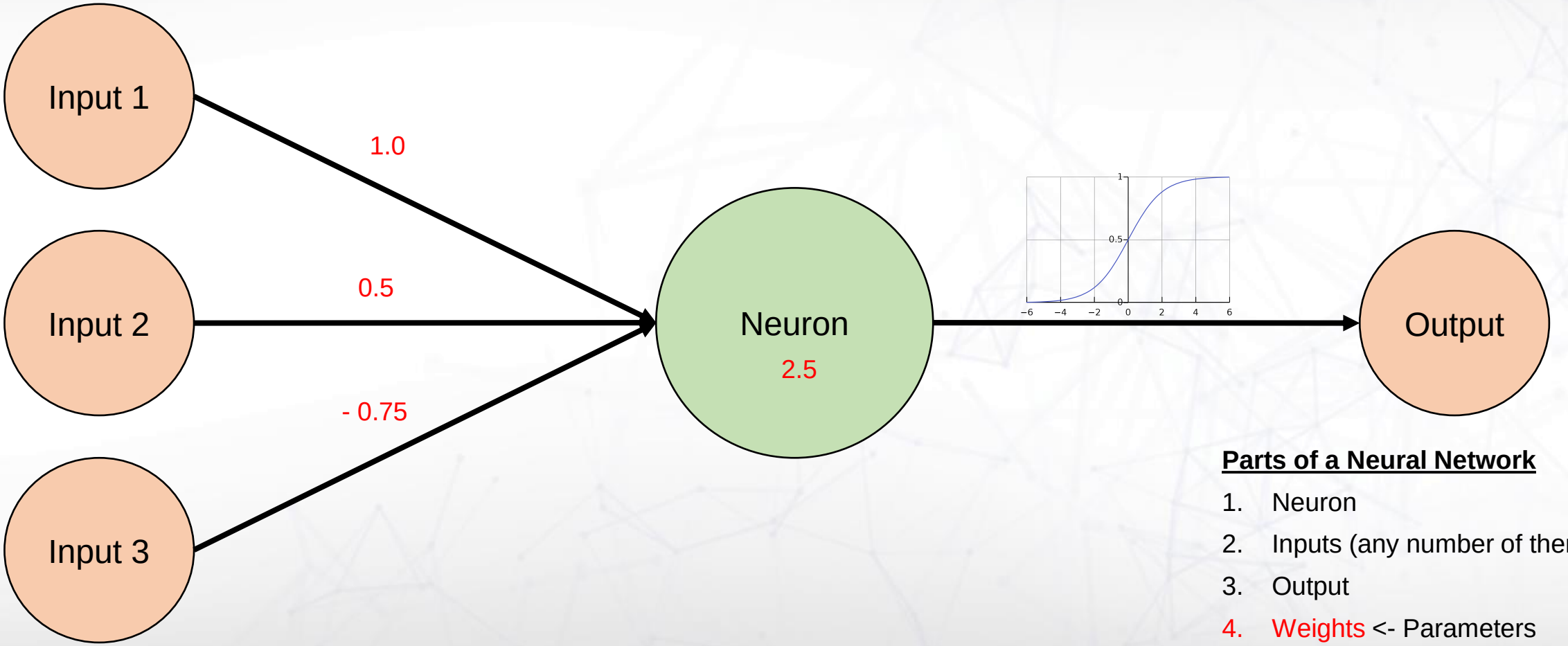
The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

The Neuron

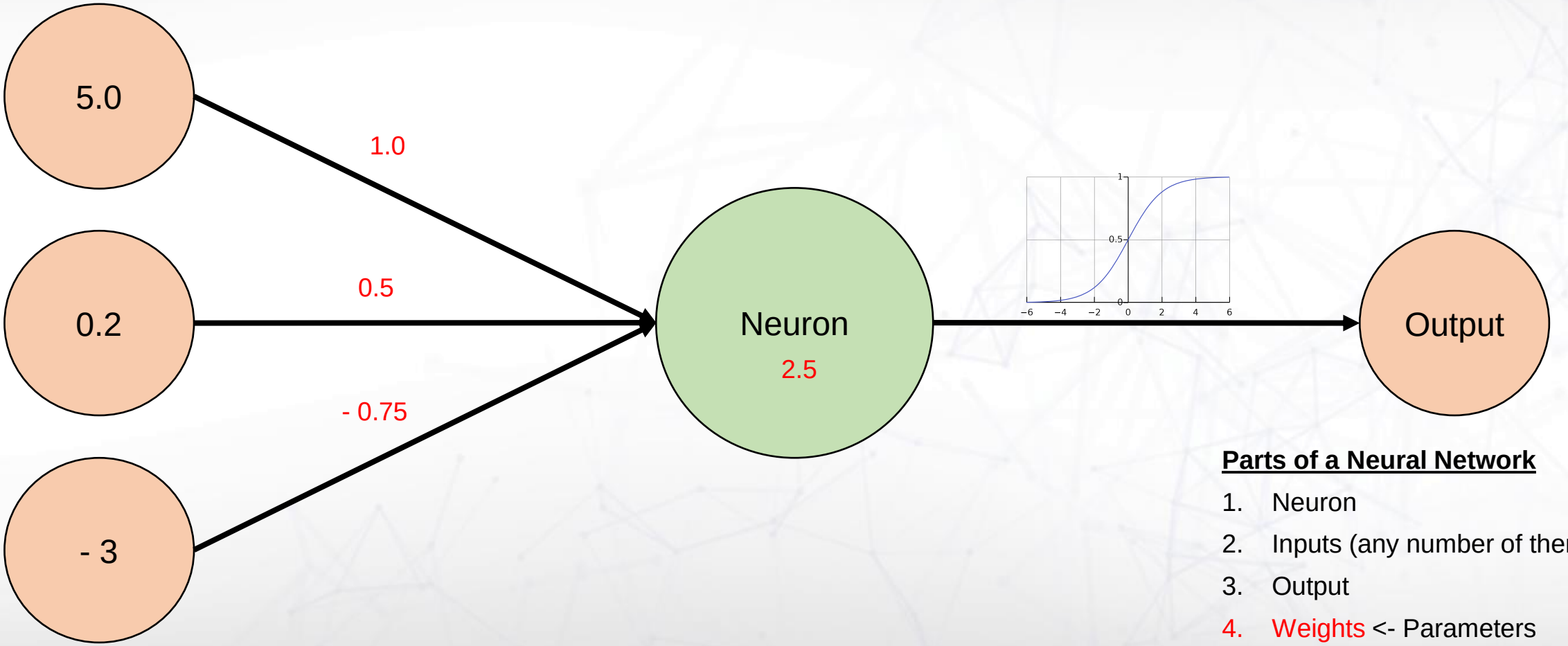


Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

The Neuron

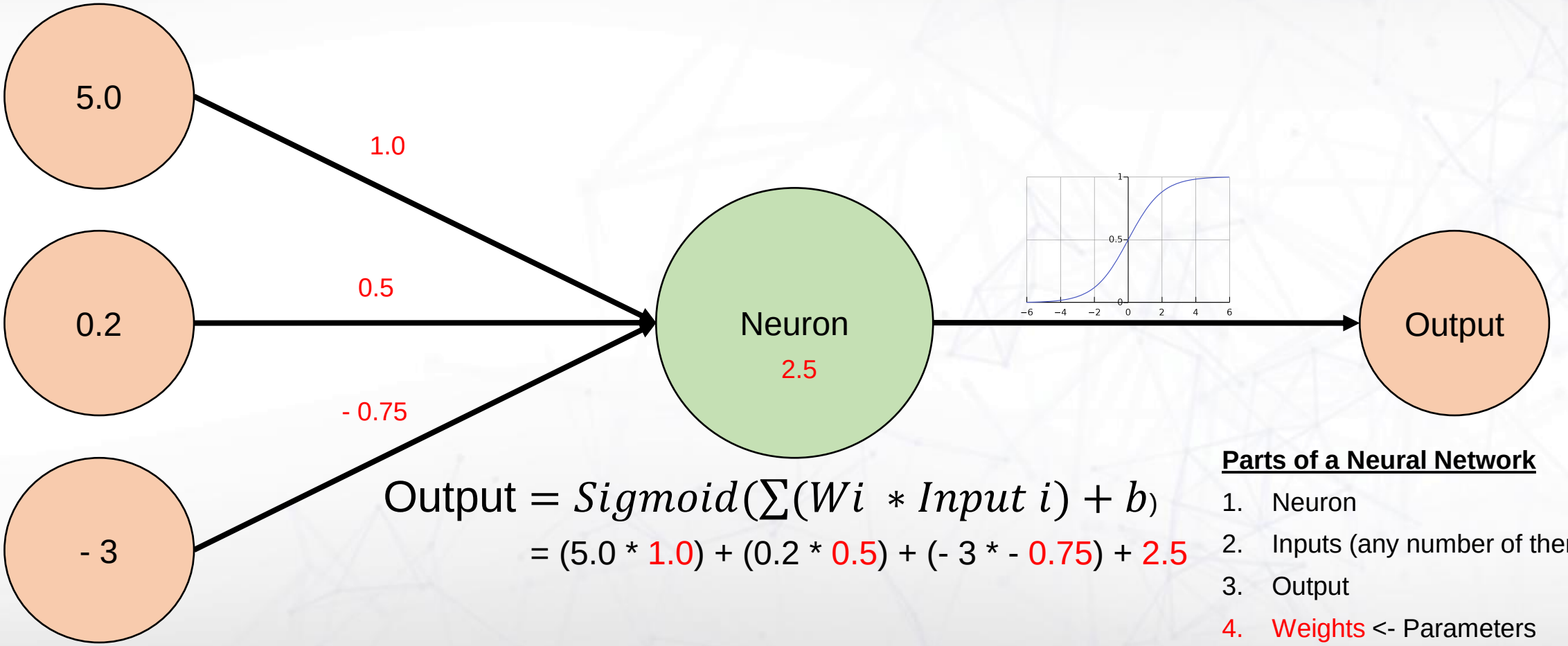


Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

The Neuron



Output = $Sigmoid(\sum(Wi * Input\ i) + b)$

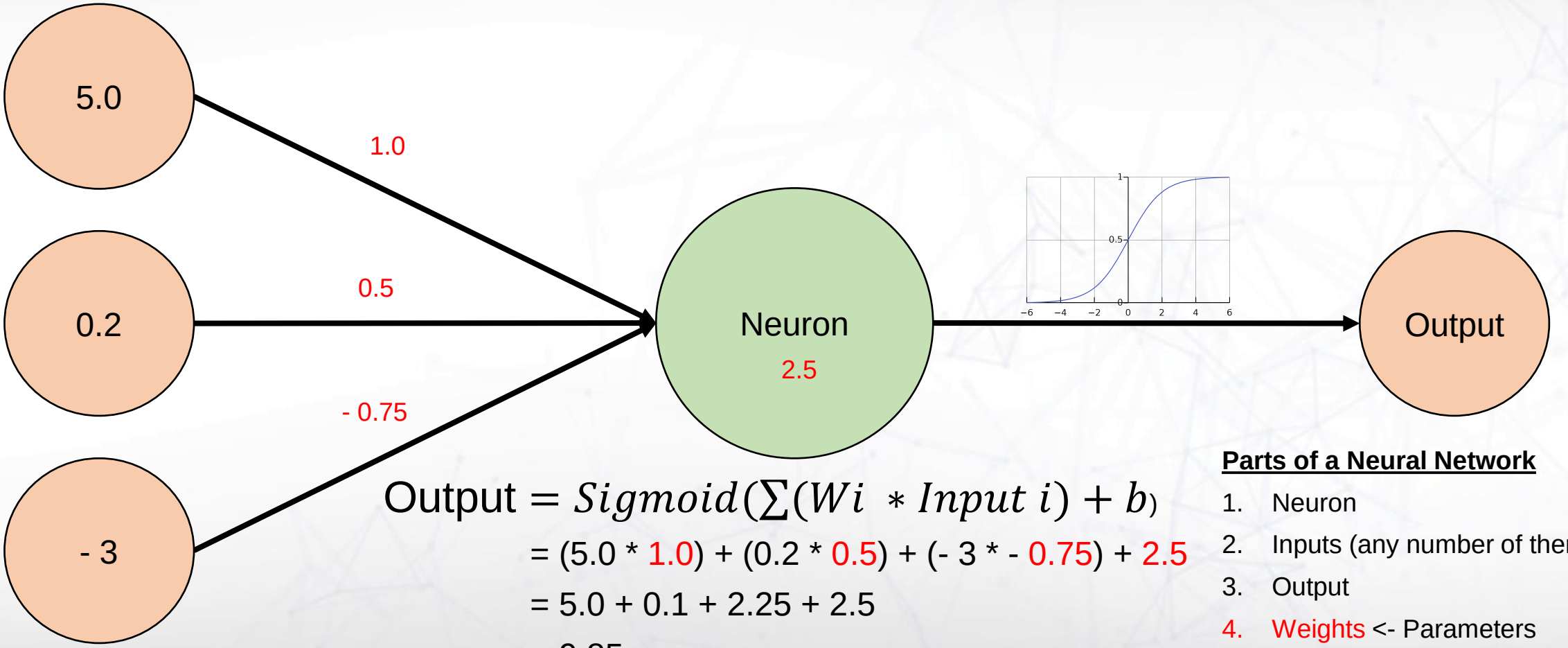
= (5.0 * 1.0) + (0.2 * 0.5) + (- 3 * - 0.75) + 2.5

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

The Neuron



$$\text{Output} = Sigmoid(\sum(Wi * Input\ i) + b)$$

= (5.0 * 1.0) + (0.2 * 0.5) + (- 3 * - 0.75) + 2.5

= 5.0 + 0.1 + 2.25 + 2.5

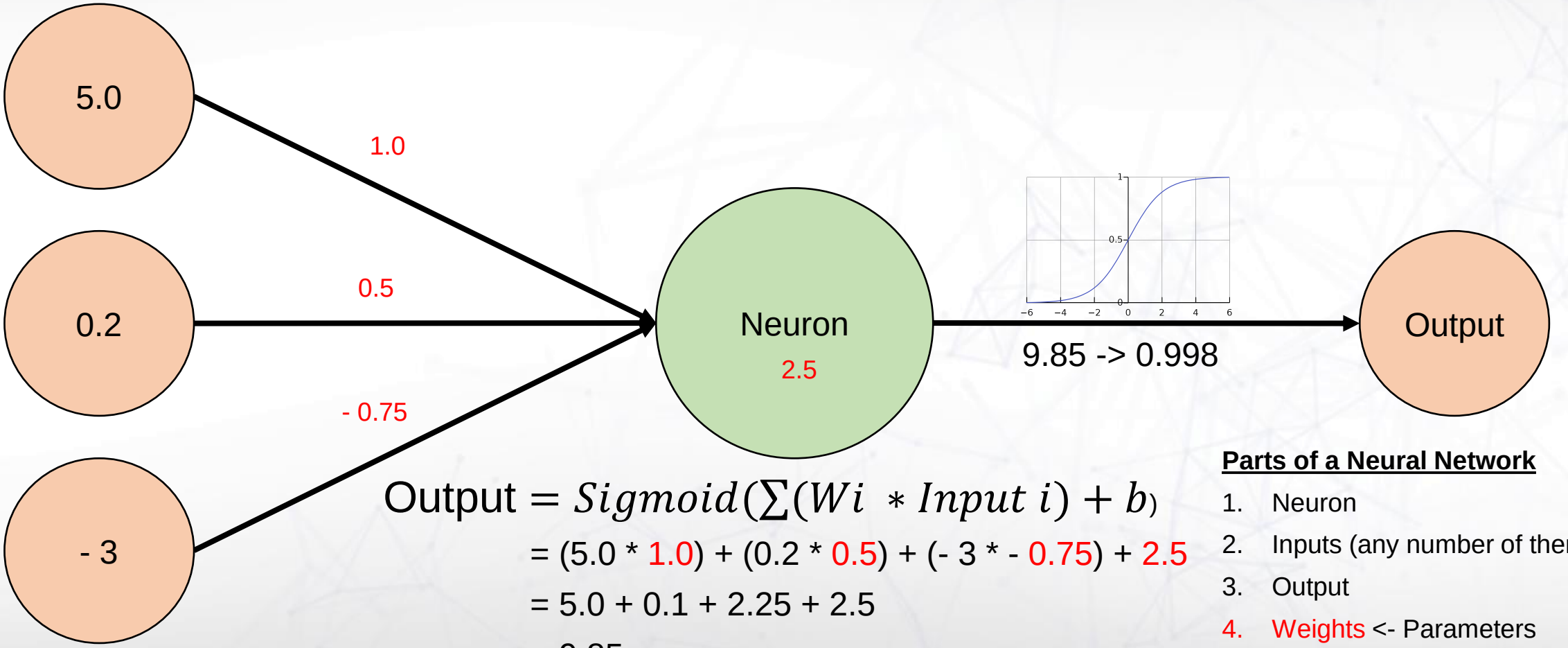
= 9.85

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

The Neuron



Output $= Sigmoid(\sum(Wi * Input\ i) + b)$

= (5.0 * 1.0) + (0.2 * 0.5) + (- 3 * - 0.75) + 2.5

= 5.0 + 0.1 + 2.25 + 2.5

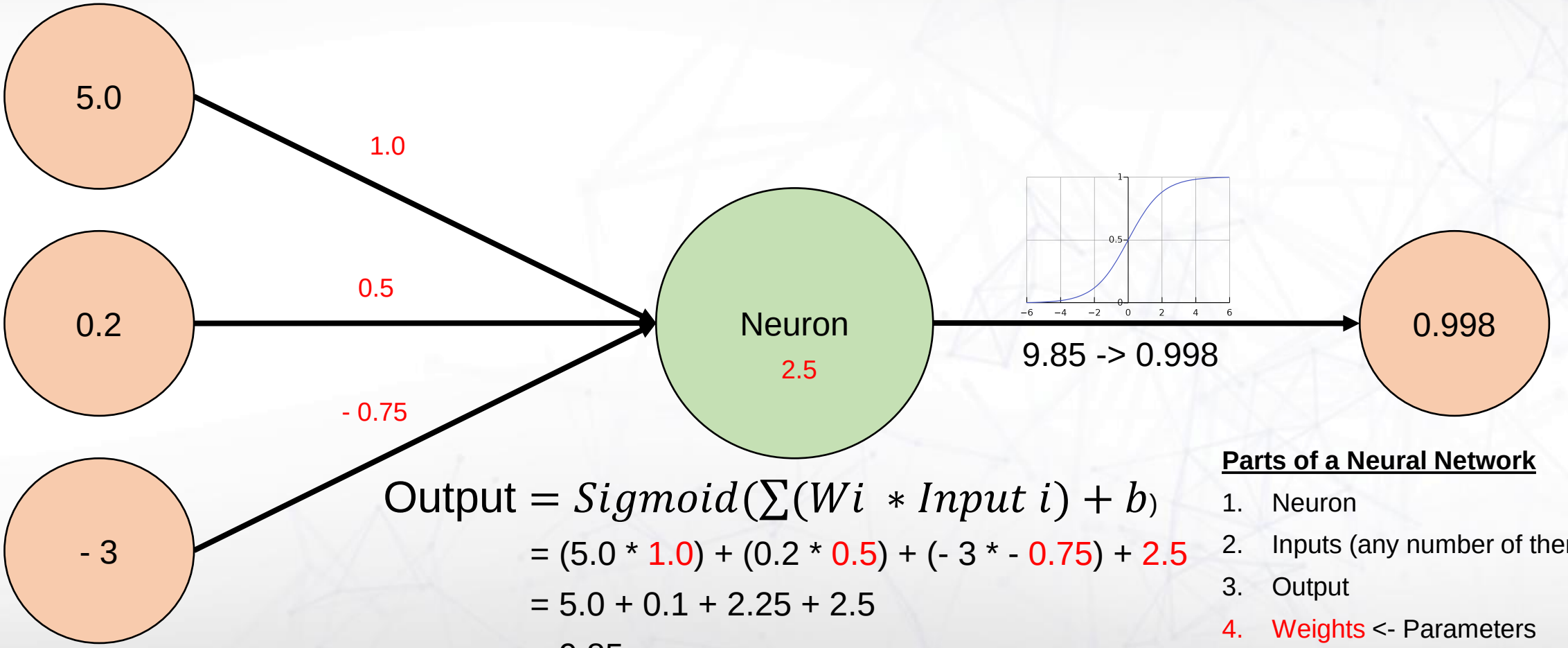
= 9.85

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

The Neuron



Output $= Sigmoid(\sum(Wi * Input\ i) + b)$

= (5.0 * 1.0) + (0.2 * 0.5) + (- 3 * - 0.75) + 2.5

= 5.0 + 0.1 + 2.25 + 2.5

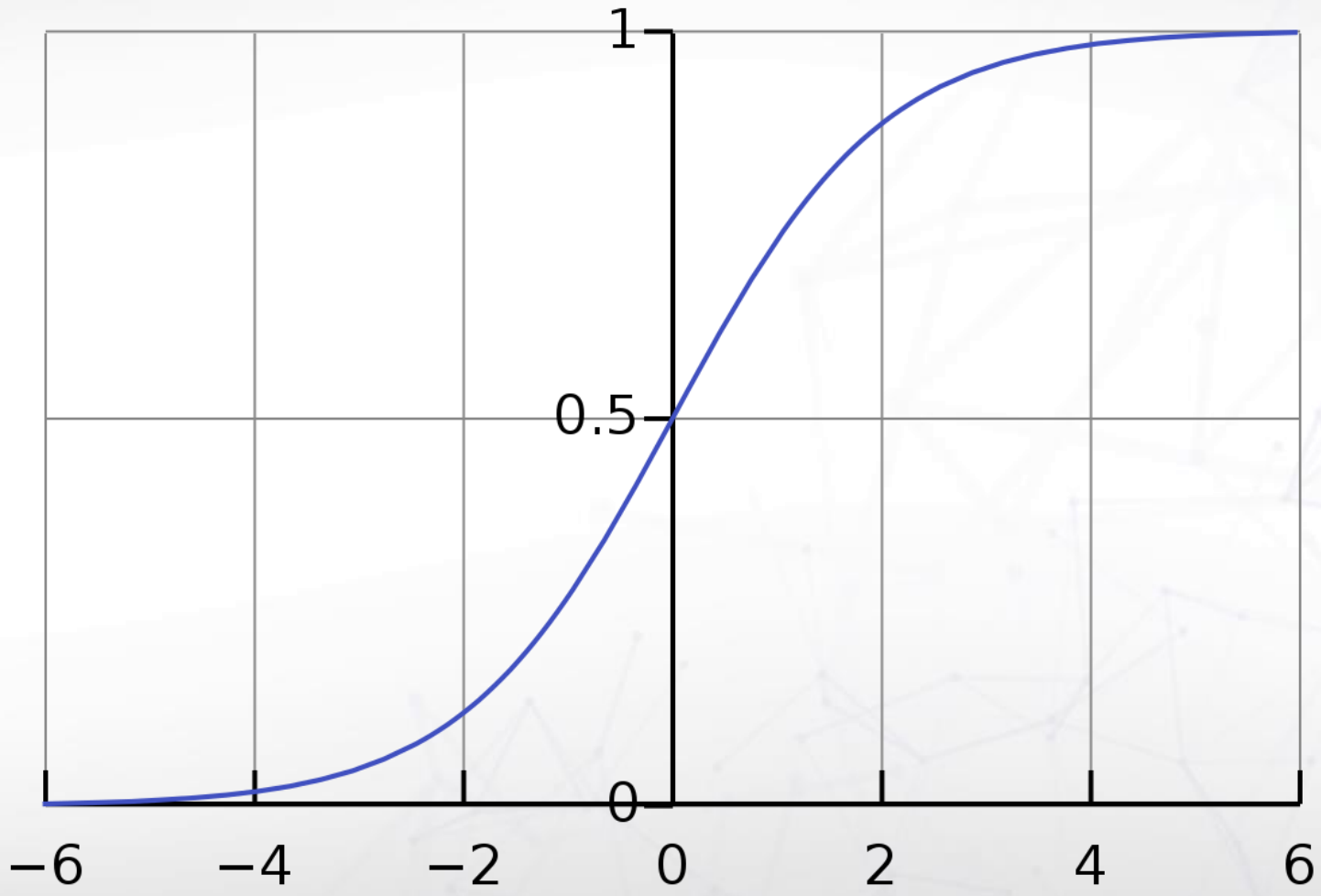
= 9.85

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

Sigmoid (non-linearity):



Sigmoid:

Takes an input and puts it in between the values of 0 and 1

Input 0 -> Output 0.5

Input 1 -> Output 0.75

Input 2 -> Output 0.83

Input 100 -> Output ~ 1.00

Input -1 -> Output 0.25

Input -2 -> Output 0.17

Input -100 -> ~ 0.00

Why? It makes your ANN nonlinear (math term).

Intuition: The model can learn that:

0 -> False, 1 -> True

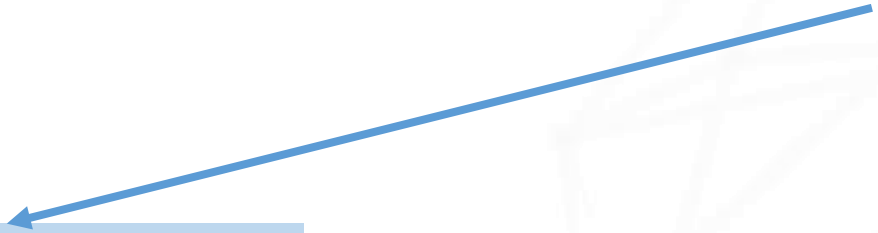
Why do these Neural Networks work?

Let’s look at an example

Question: Should I walk my dog today?

Let’s look at an example

Question: Should I walk my dog today?



How sunny is it today?

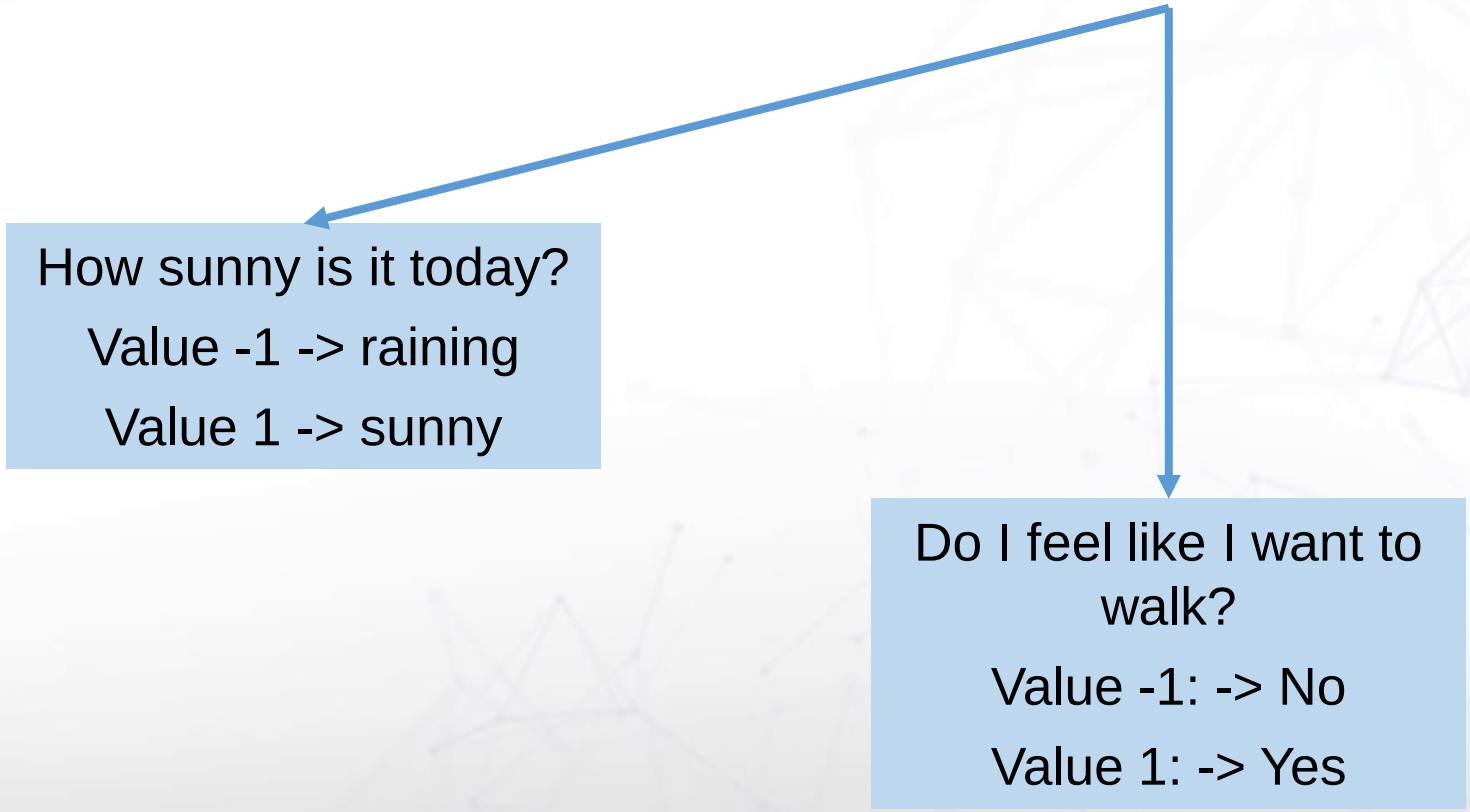
Value -1 -> raining

Value 1 -> sunny

Let’s look at an example

BRAINTANK DEEP LEARNING

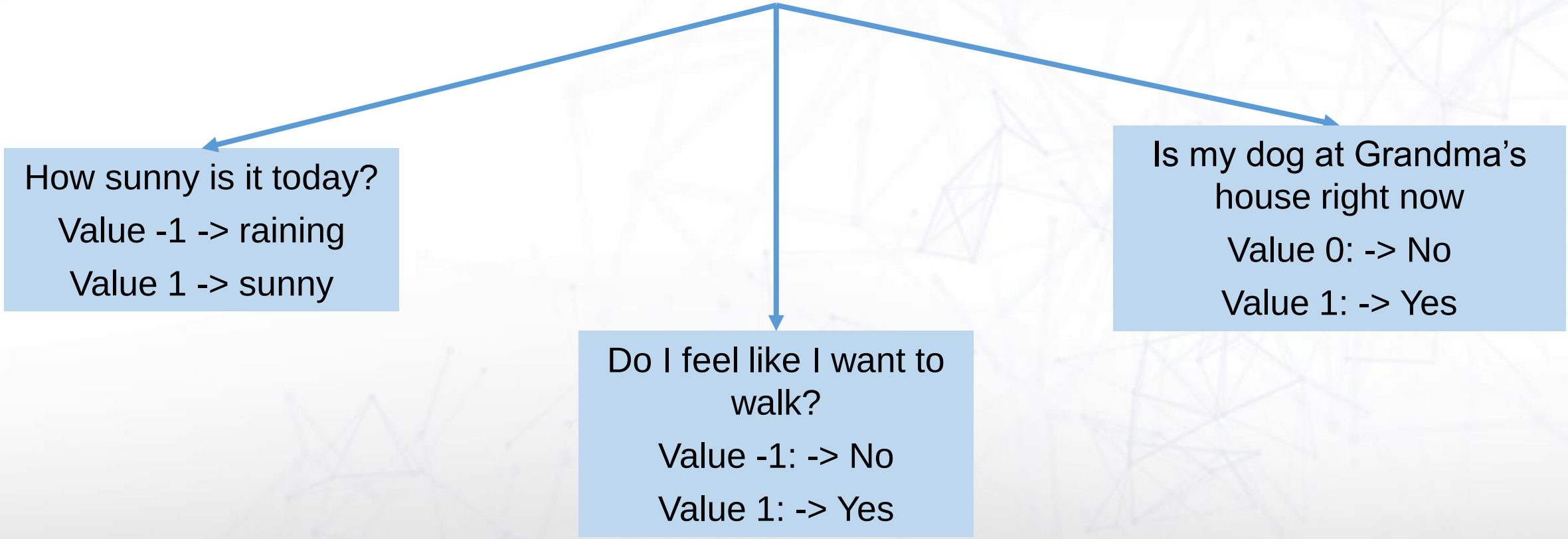
Question: Should I walk my dog today?



Let’s look at an example

BRAINTANK DEEP LEARNING

Question: Should I walk my dog today?

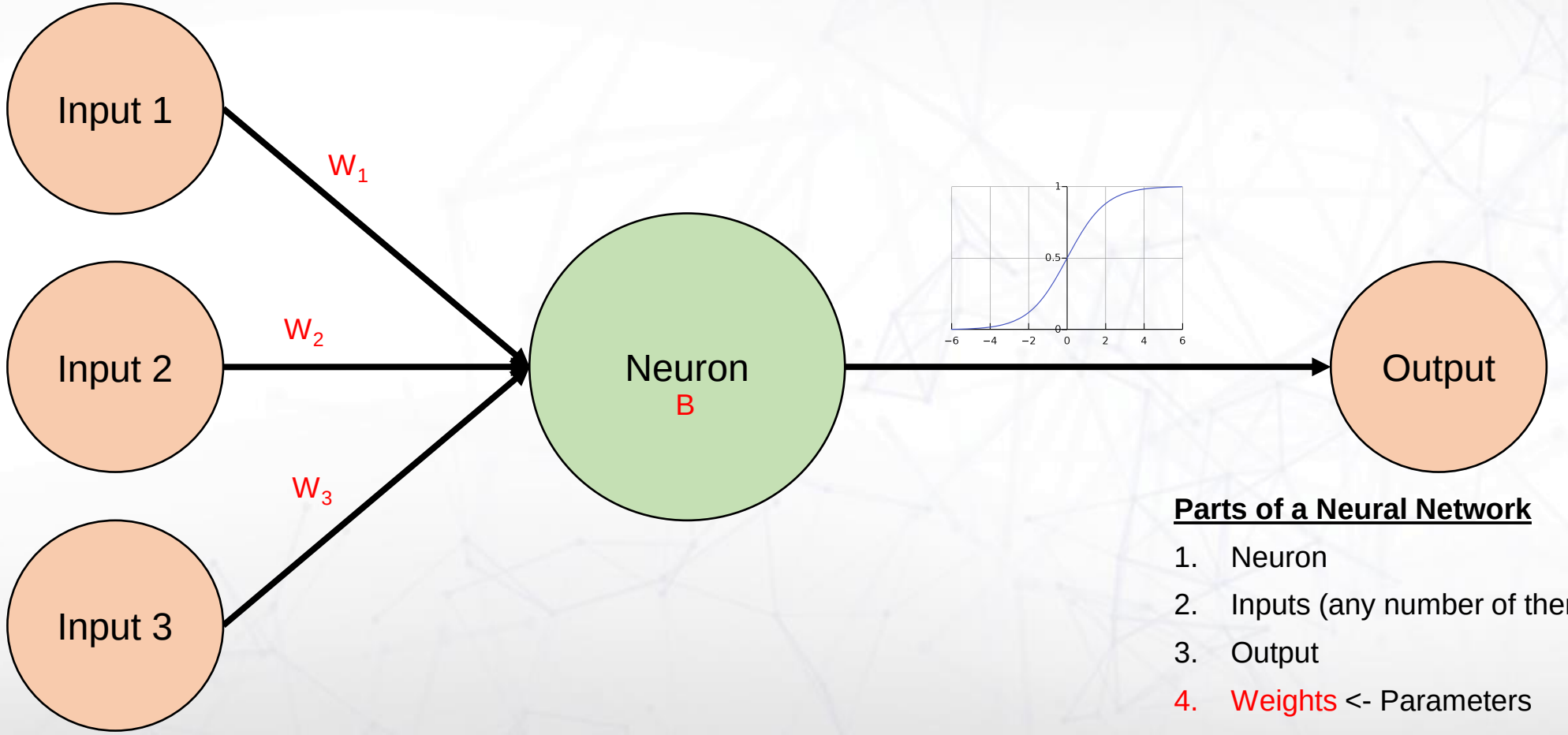


The Neuron

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

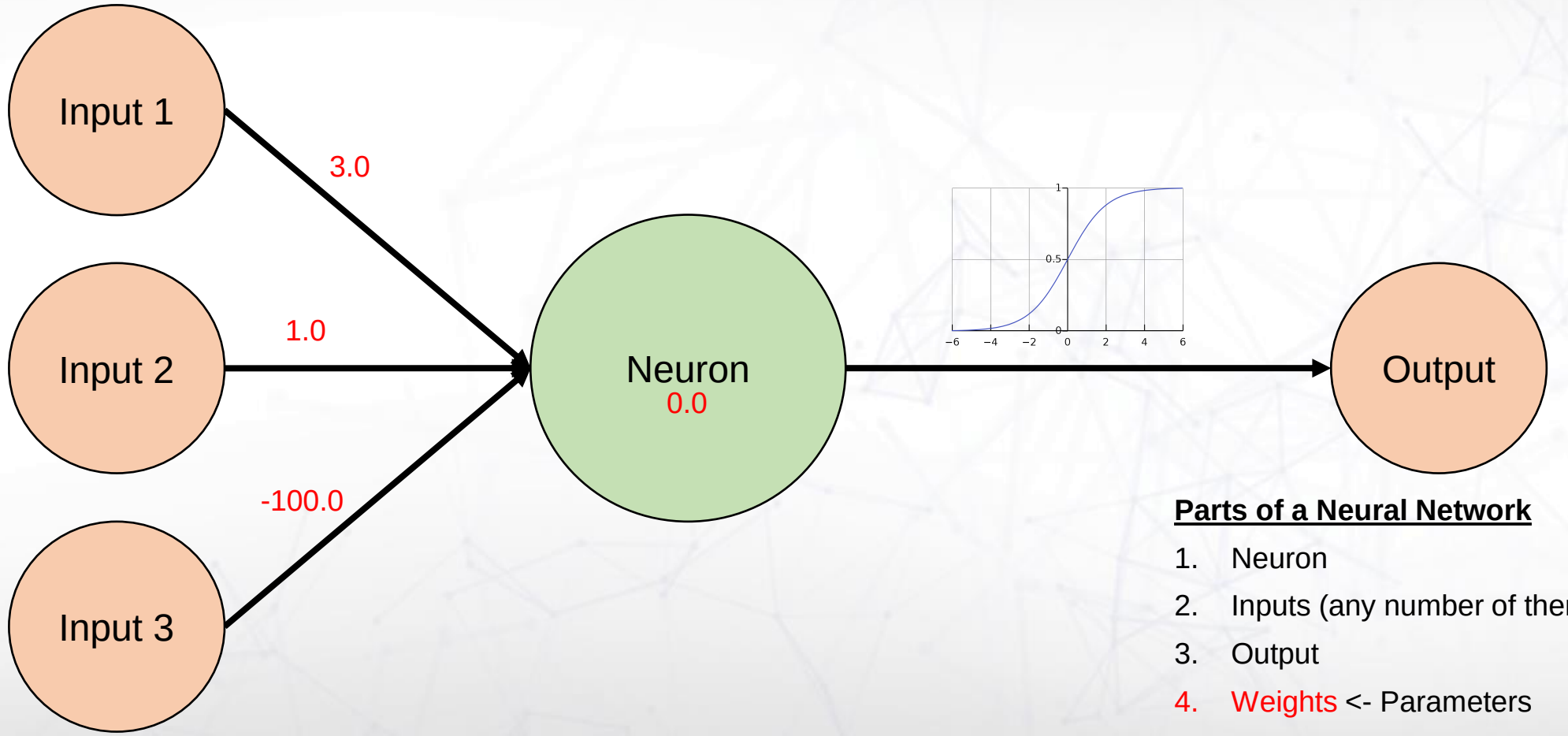
Remember: Parameters are just numbers

The Neuron

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

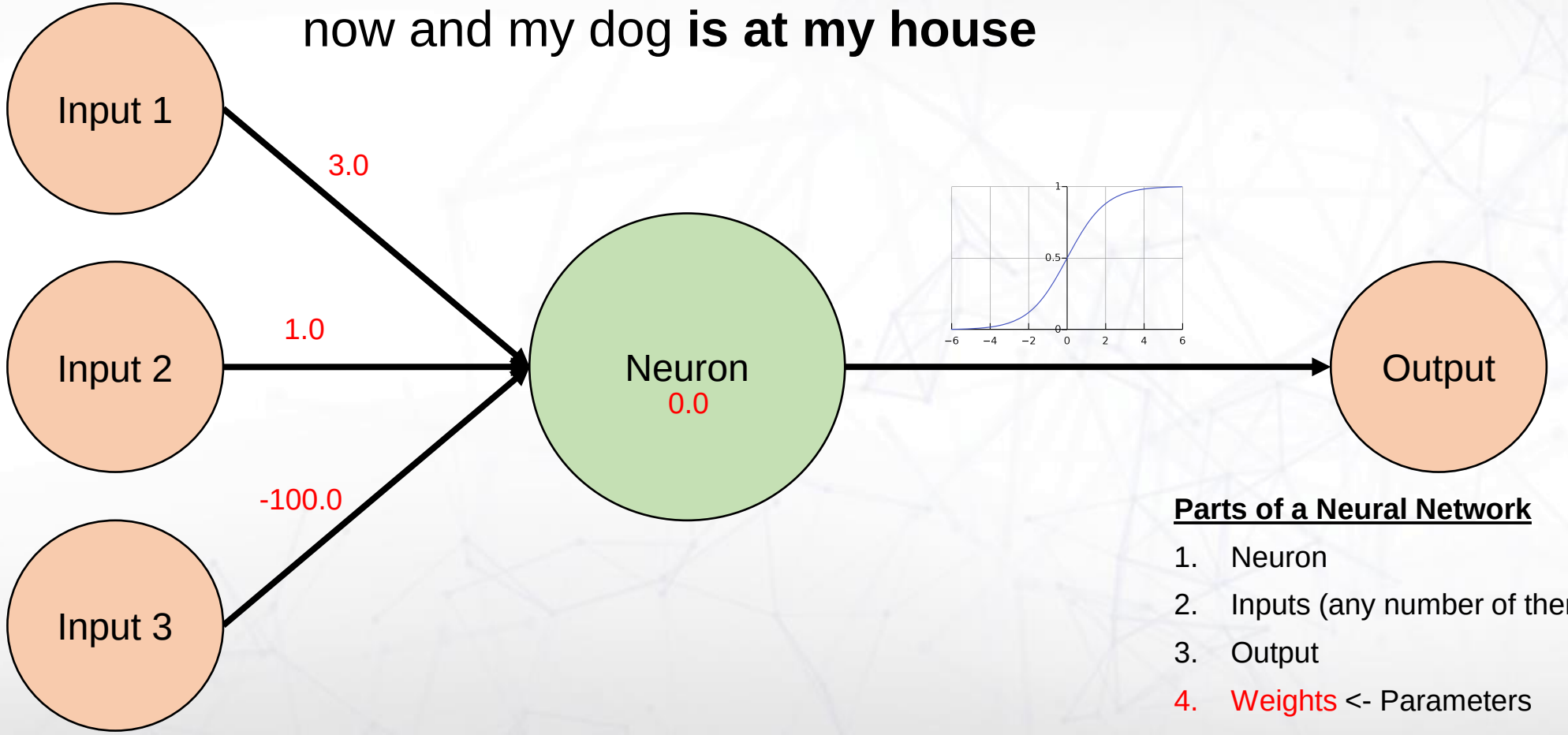
The Neuron

Scenario: Day is **sunny**, I'd **love** to go for a walk right now and my dog **is at my house**

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

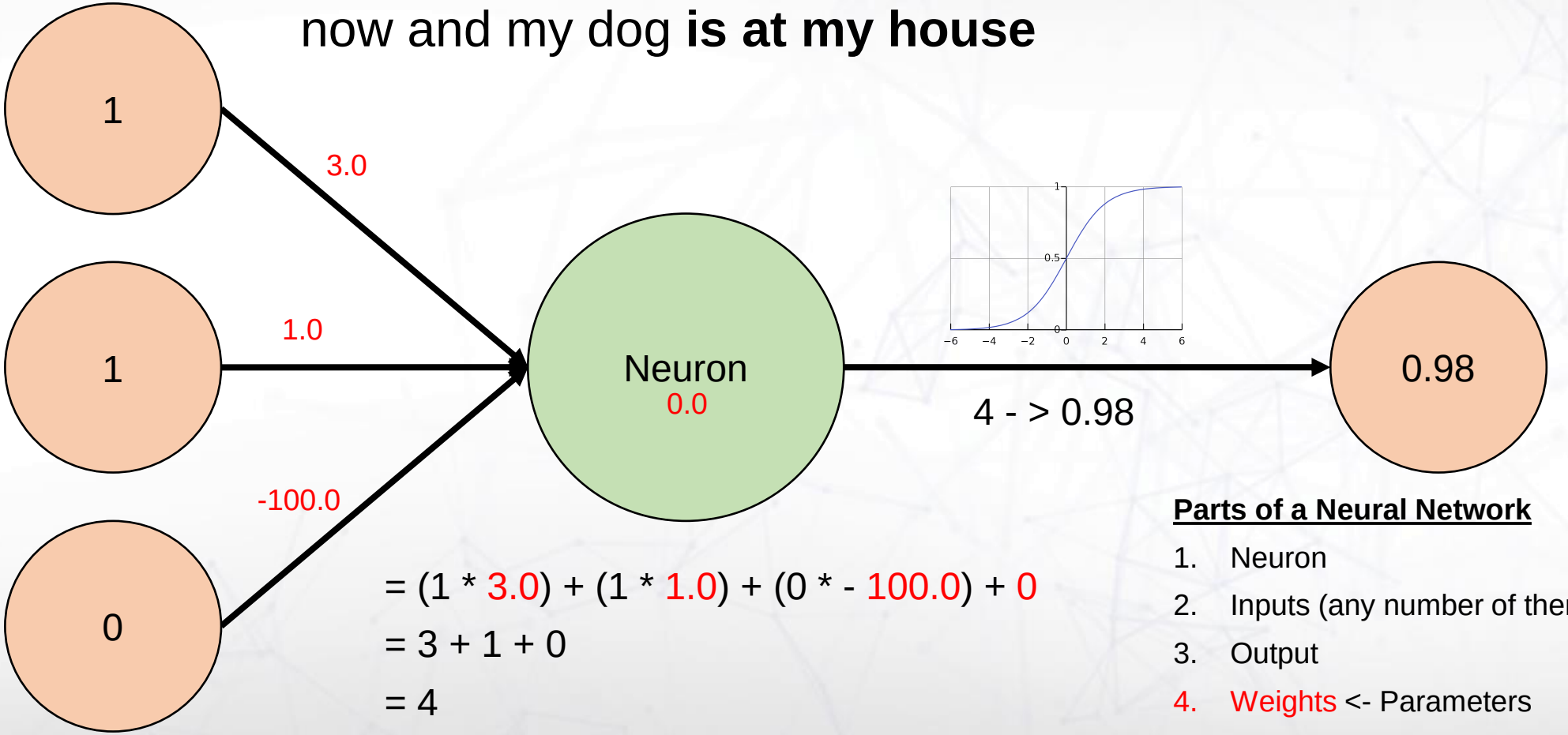
The Neuron

Scenario: Day is **sunny**, I'd **love** to go for a walk right now and my dog **is at my house**

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



= (1 * 3.0) + (1 * 1.0) + (0 * - 100.0) + 0
= 3 + 1 + 0
= 4

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

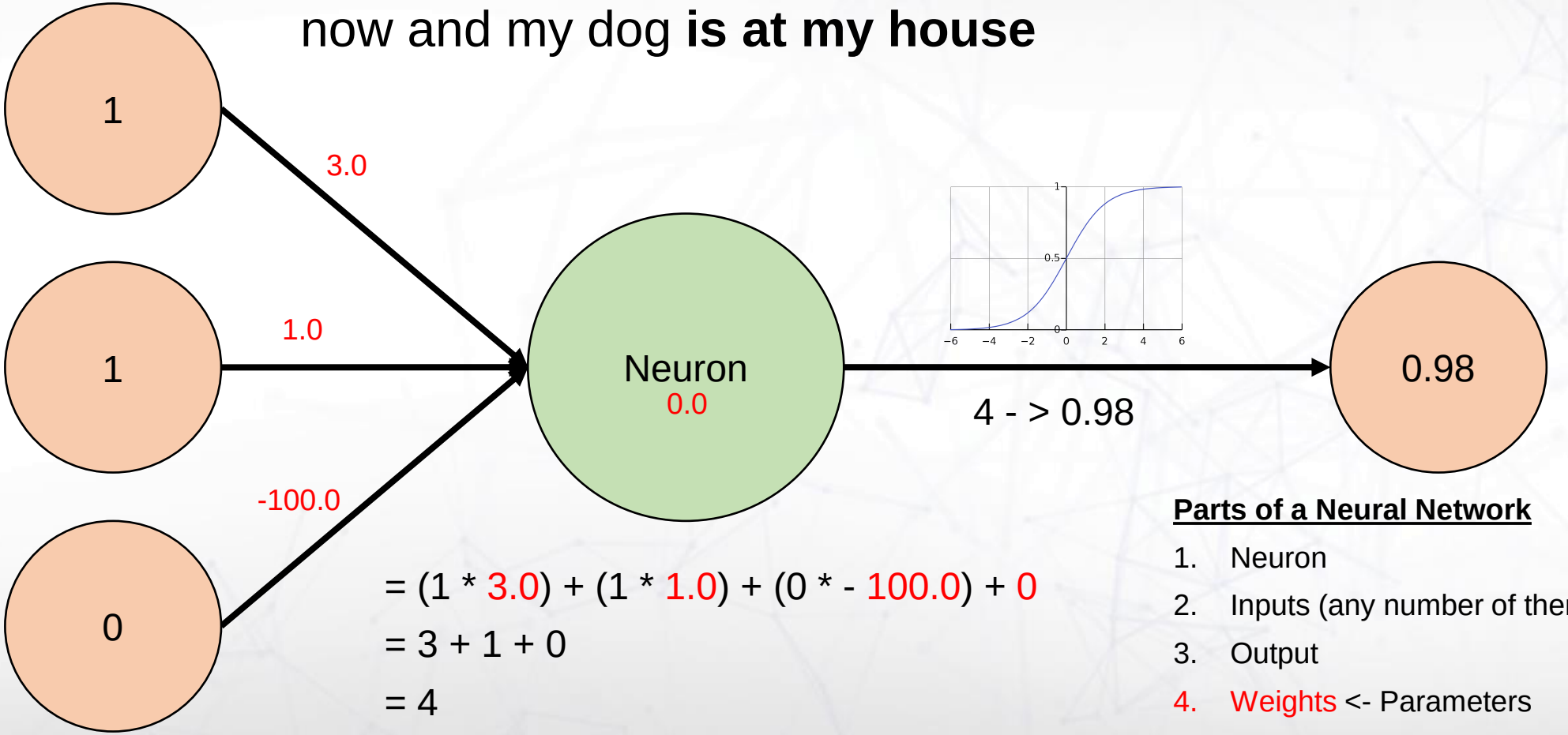
The Neuron

Scenario: Day is **normal**, I'd **like** to go for a walk right now and my dog **is at my house**

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



= (1 * 3.0) + (1 * 1.0) + (0 * - 100.0) + 0
= 3 + 1 + 0
= 4

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

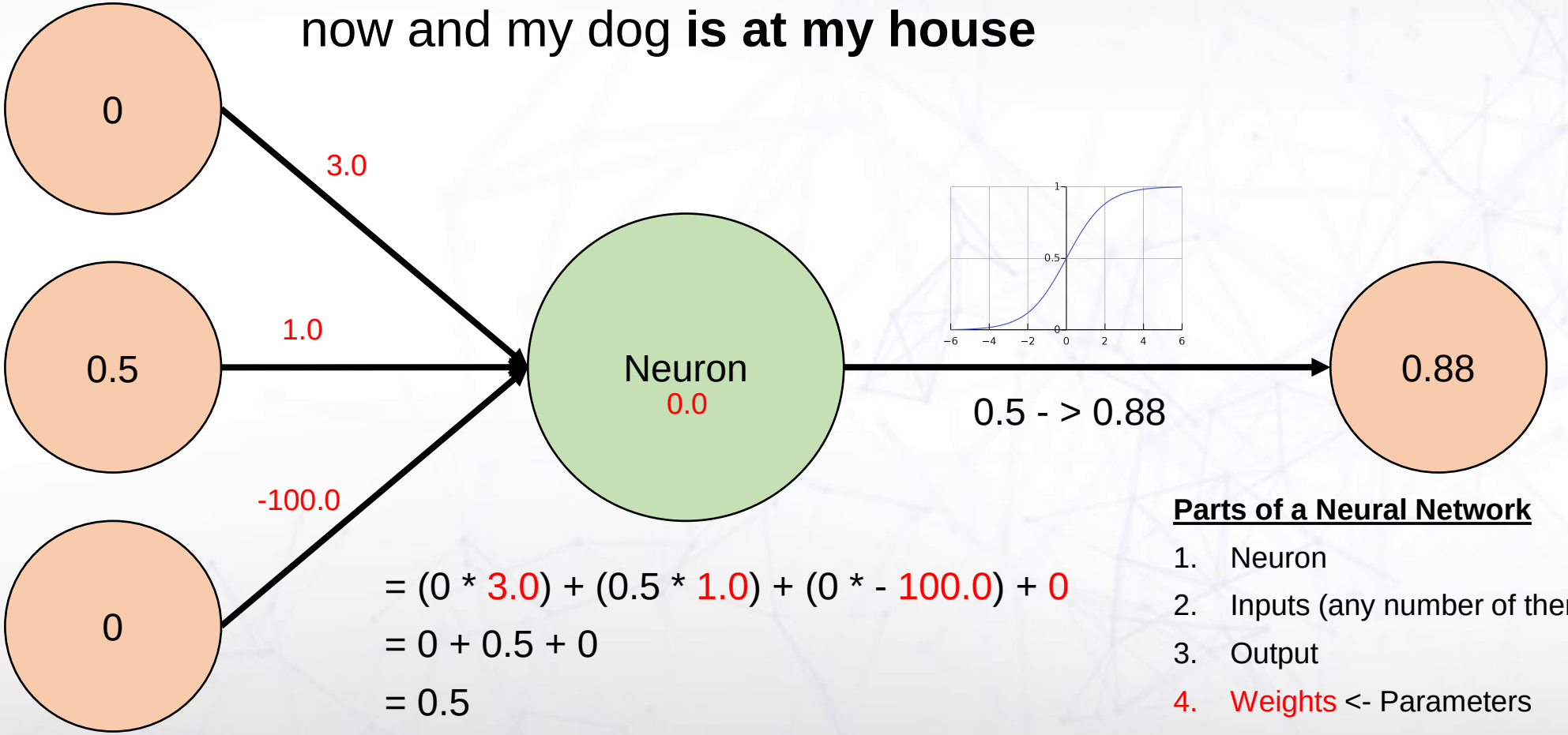
The Neuron

Scenario: Day is **normal**, I'd **like** to go for a walk right now and my dog **is at my house**

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



= (0 * 3.0) + (0.5 * 1.0) + (0 * - 100.0) + 0
= 0 + 0.5 + 0
= 0.5

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

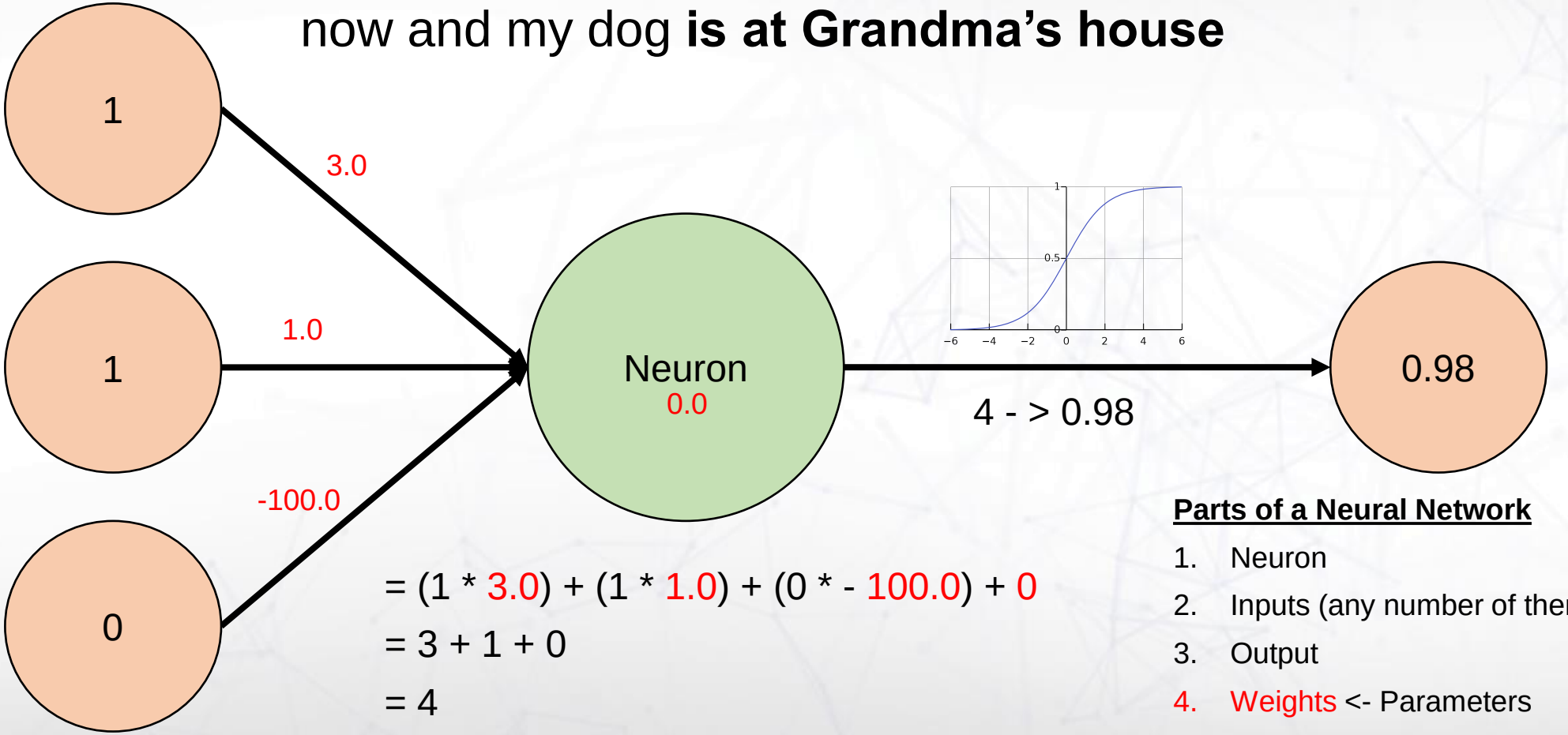
The Neuron

Scenario: Day is **sunny**, I'd **love** to go for a walk right now and my dog **is at Grandma's house**

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



= (1 * 3.0) + (1 * 1.0) + (0 * - 100.0) + 0
= 3 + 1 + 0
= 4

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

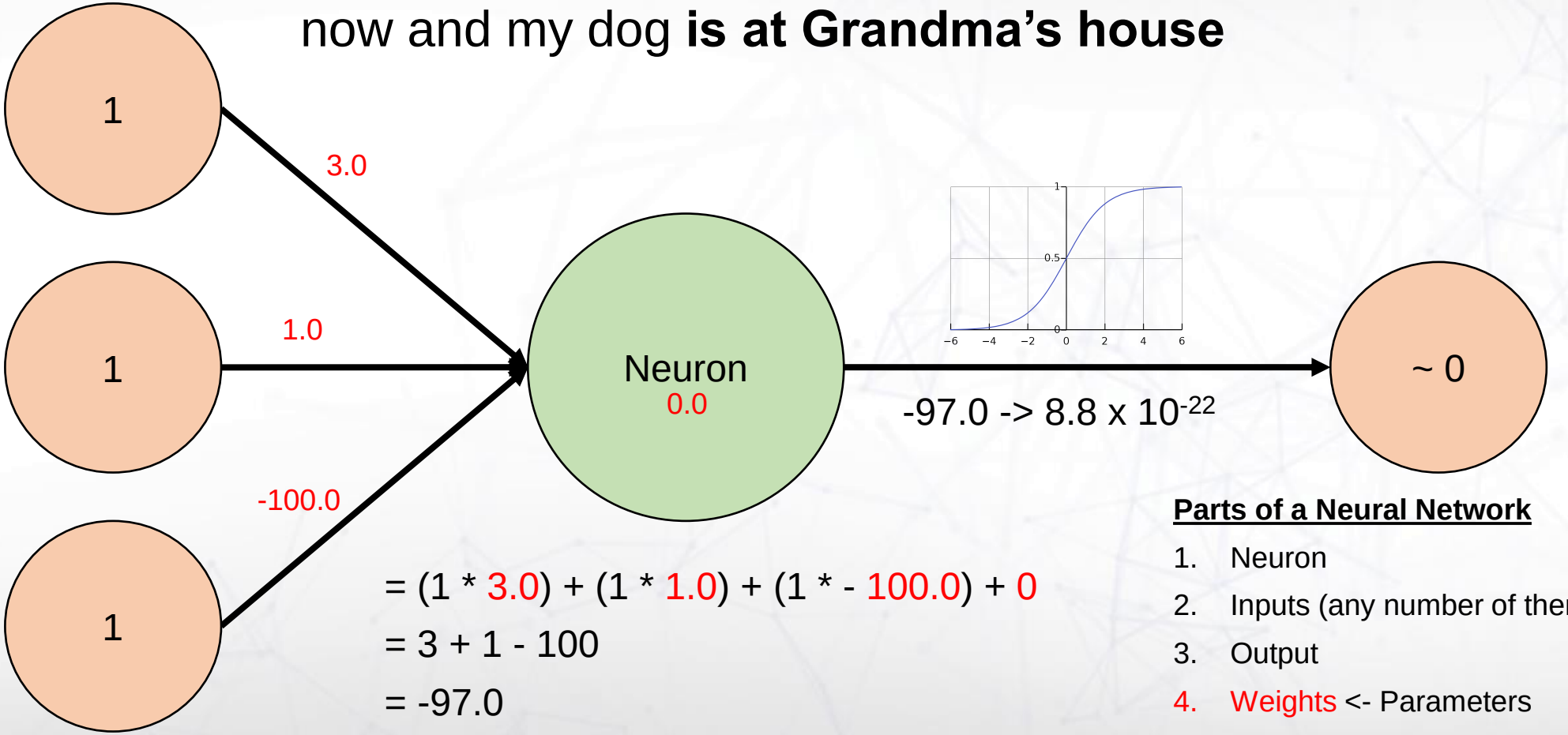
The Neuron

Scenario: Day is **sunny**, I'd **love** to go for a walk right now and my dog **is at Grandma's house**

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



= (1 * 3.0) + (1 * 1.0) + (1 * - 100.0) + 0
= 3 + 1 - 100
= -97.0

Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

Remember: Parameters are just numbers

Takeaways:

Even though this neural network was small it was able to pick up on truths about human life

1. If your dog is not at house it is nearly impossible for you to walk it. The model will predict 0 every time.
2. The day being sunny is a strong indicator that you take your dog for a walk
3. The fact that you want to walk is a factor to determining if you will walk your dog but it is not as strong as other factors

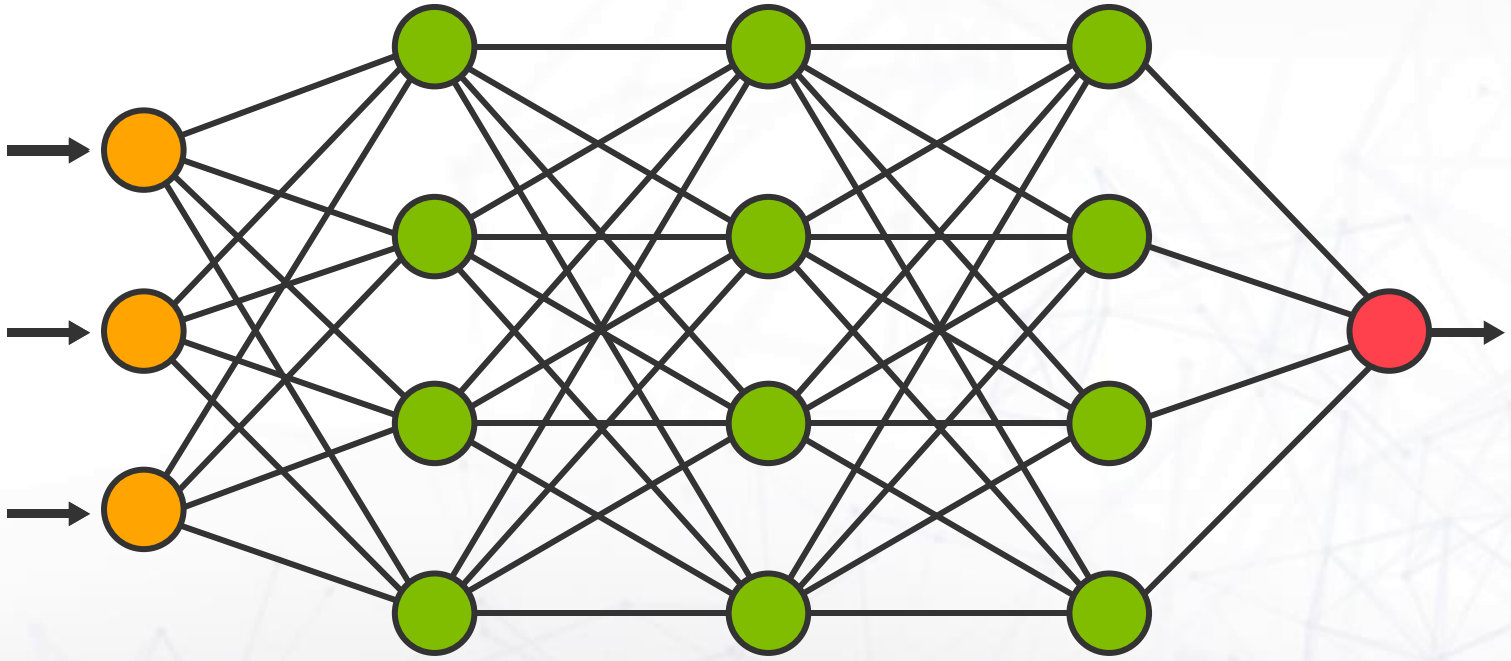
If one neuron is strong, thousands are stronger

BRAINTANK DEEP LEARNING

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes



INPUT LAYER

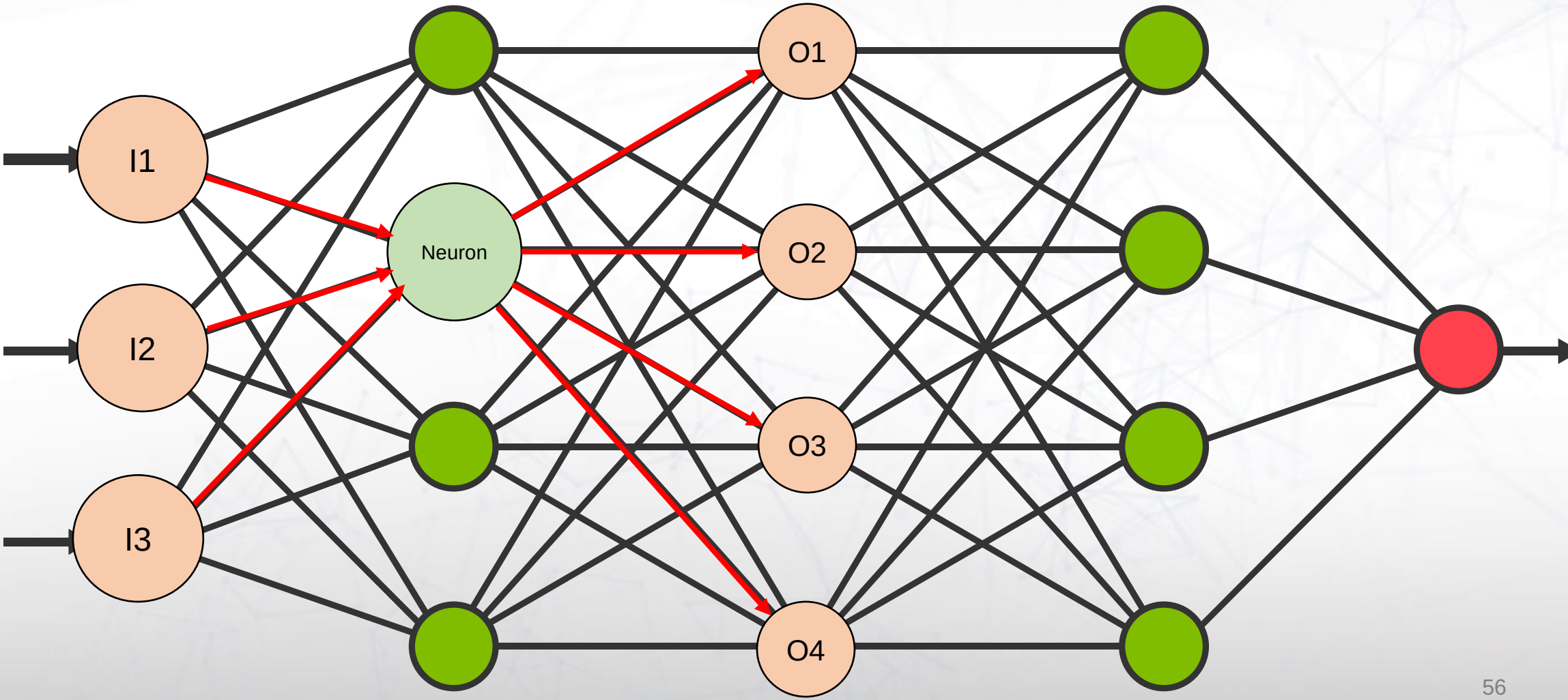
HIDDEN LAYERS

OUTPUT LAYER

Likeliness I will walk my dog

If one neuron is strong, thousands are stronger

BRAINTANK DEEP LEARNING



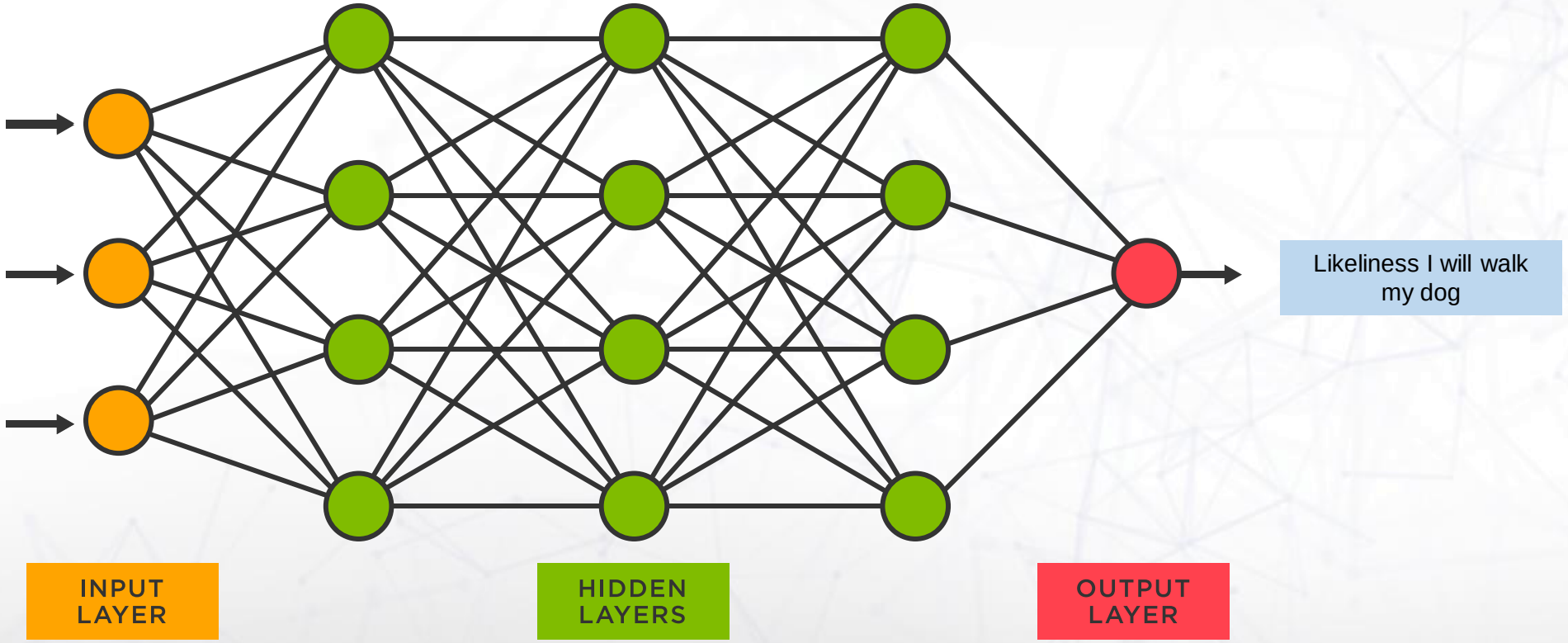
If one neuron is strong, thousands are stronger

BRAINTANK DEEP LEARNING

How sunny is it today?
Value -1 -> raining
Value 1 -> sunny

Do I feel Like Walking?
Value -1 -> no
Value 1 -> yes

Is the dog at Grandma's right now?
Value 0 -> no
Value 1 -> yes

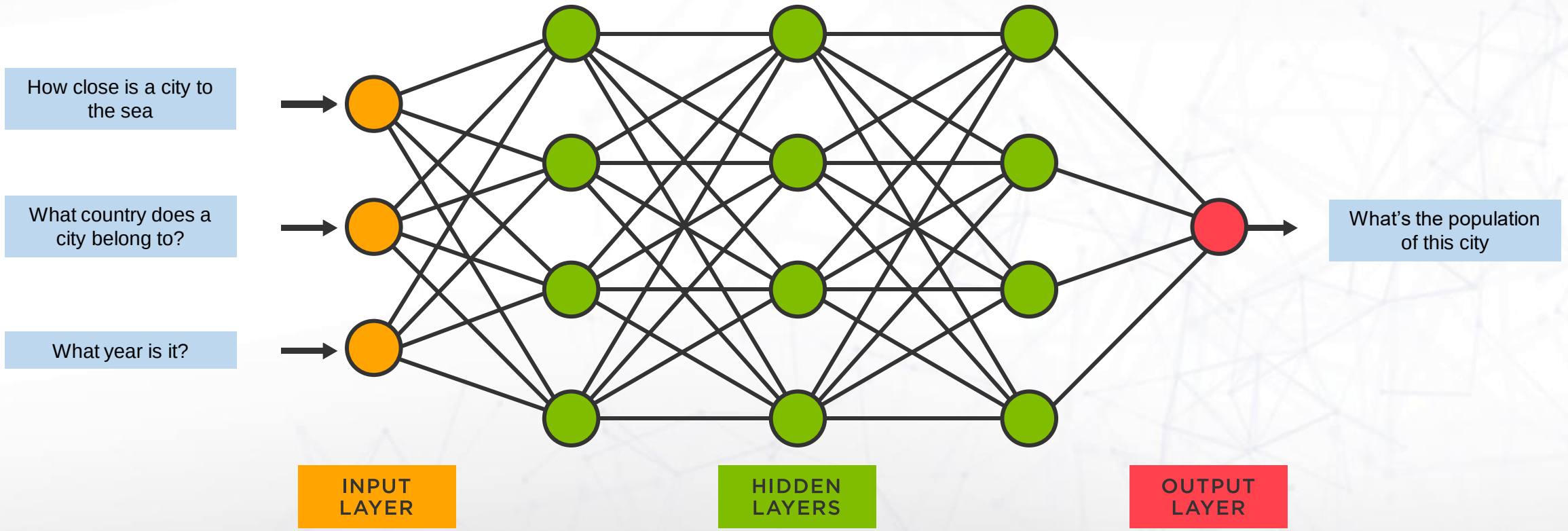


Likeliness I will walk my dog

More Neurons → More Power

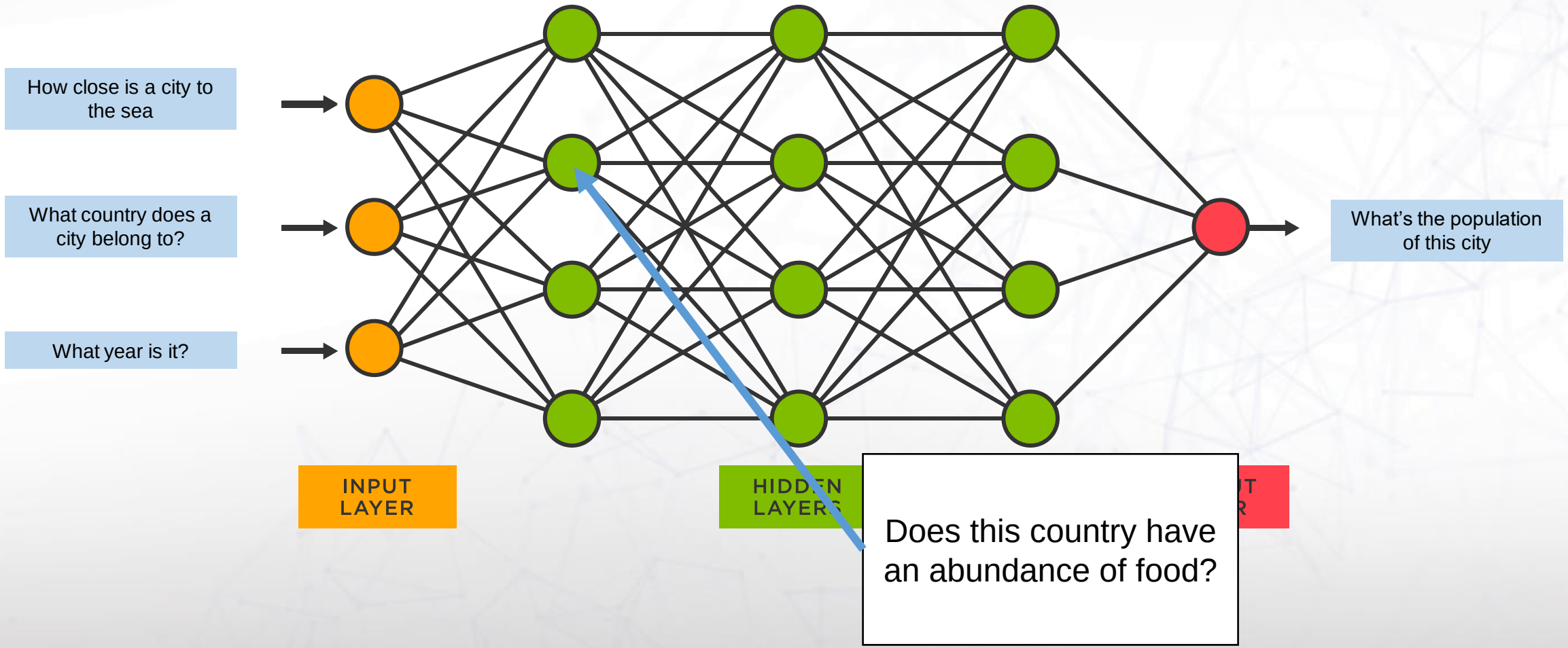
If one neuron is strong, thousands are stronger

BRAINTANK DEEP LEARNING

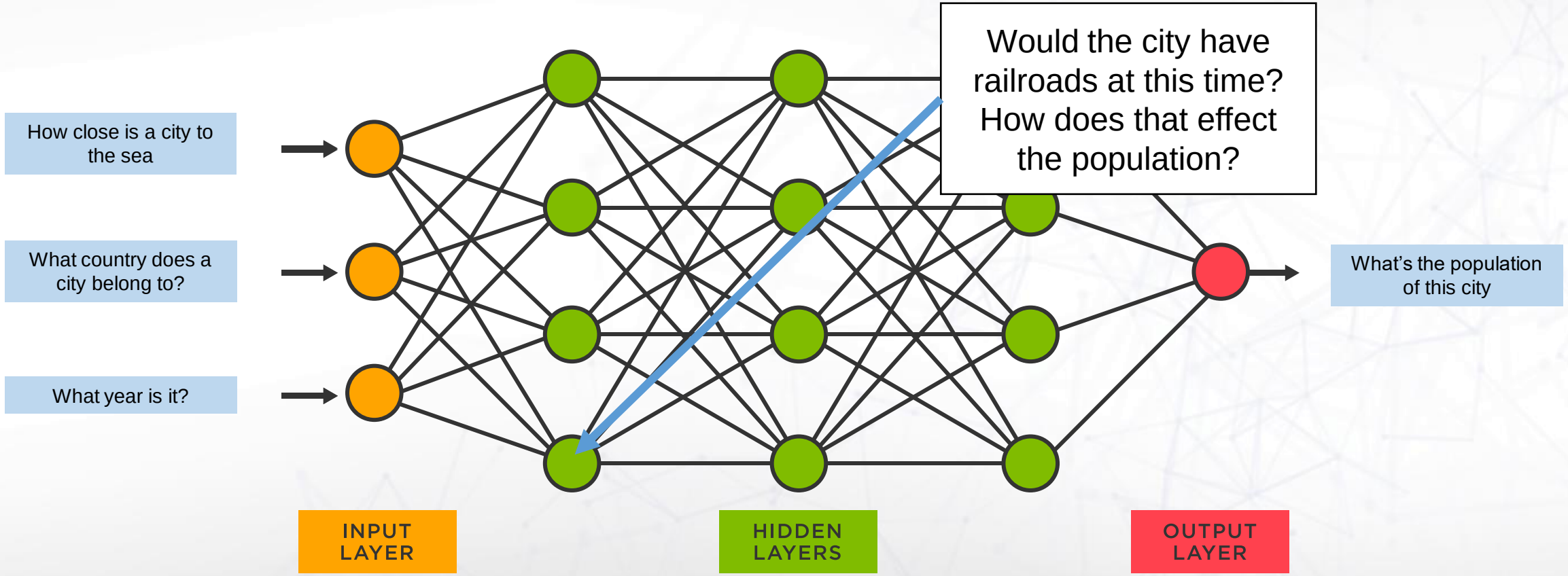


If one neuron is strong, thousands are stronger

BRAINTANK DEEP LEARNING

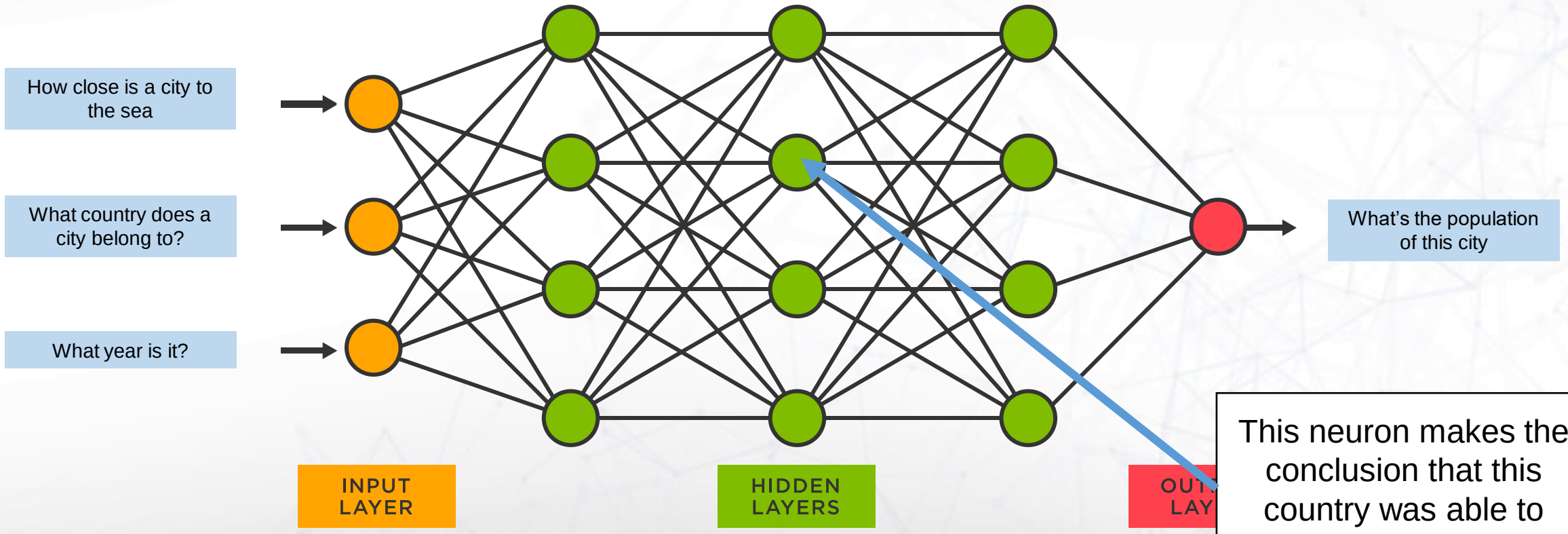


If one neuron is strong, thousands are stronger

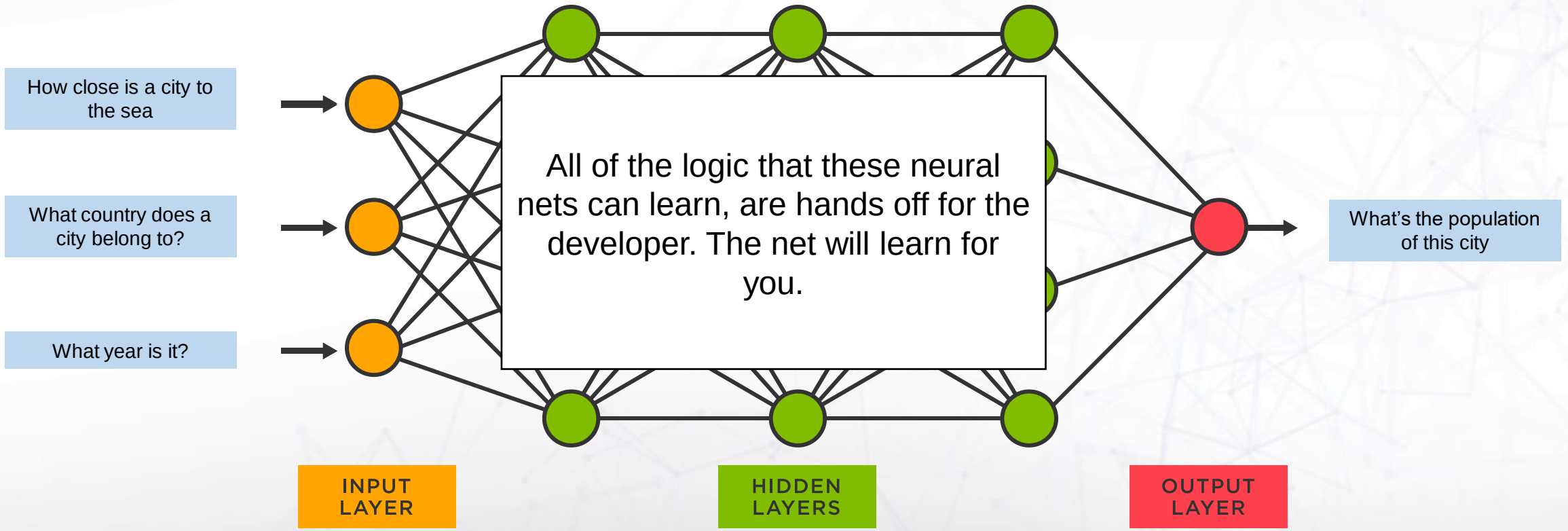


If one neuron is strong, thousands are stronger

BRAINTANK DEEP LEARNING



If one neuron is strong, thousands are stronger



Training Neural Networks

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
…	…	…	…	…	…

1 to 10 of 7845 entries Filter

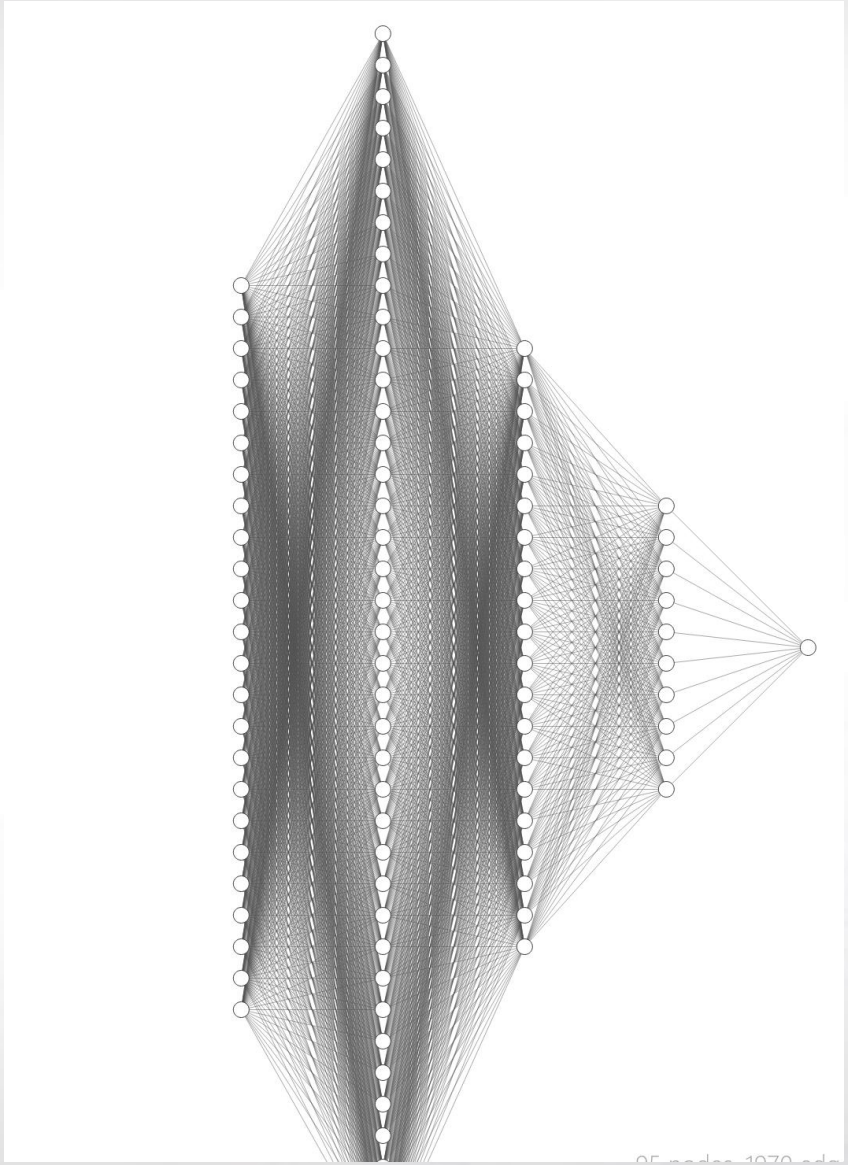
▲	Price	Carat	D	E	F	G	H	I	J	FL	IF	VVS1	VVS2	VS1	VS2	SI1	SI2	Cushion	Emerald	Heart	Oval	Pear	Princess	Radiant	Round
0	0.007913461796532396	0.05730129390018485	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
1	0.03399082776760116	0.05730129390018485	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
10	0.05339124909704686	0.05730129390018485	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
100	0.19117304805782878	0.1515711645101664	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
1000	0.18872036940097583	0.11275415896487989	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1001	0.16070730422465399	0.11275415896487989	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
1002	0.13484110407479374	0.11275415896487989	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
1003	0.15651729968338313	0.11275415896487989	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
1004	0.18236210618944565	0.11645101663585952	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
1005	0.189282574308784	0.12014787430683918	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Show 10 per page

1 2 10 100 700 780 785

Target

24 x Features



Our Model:

Activation Layer: 24 nodes

Hidden Layer 1: 40 nodes

Hidden Layer 2: 20 nodes

Hidden Layer 3: 10 nodes

Output Layer: 1 node

The 3 Tenets of Deep Learning:

BRAINTANK DEEP LEARNING

Dataset:

- A set of data that has information that we can use to teach the model how to preform better.

Model:

- Mathematical function (equation) that takes an input and creates a desired output. It is defined by its parameters, which are numbers we train/change to get a desired result.

Loss:

- A mathematical equation to measure how well/poorly your model is doing at completing a task.

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
...	...	...	...	...	...

The 3 Tenets of Deep Learning:

BRAINTANK DEEP LEARNING

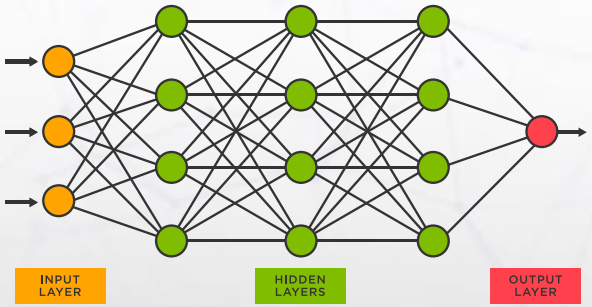
Dataset:

- A set of data that has information that we can use to teach the model how to preform better.

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
...	...	...	...	...	...

Model:

- Mathematical function (equation) that takes an input and creates a desired output. It is defined by its parameters, which are numbers we train/change to get a desired result.

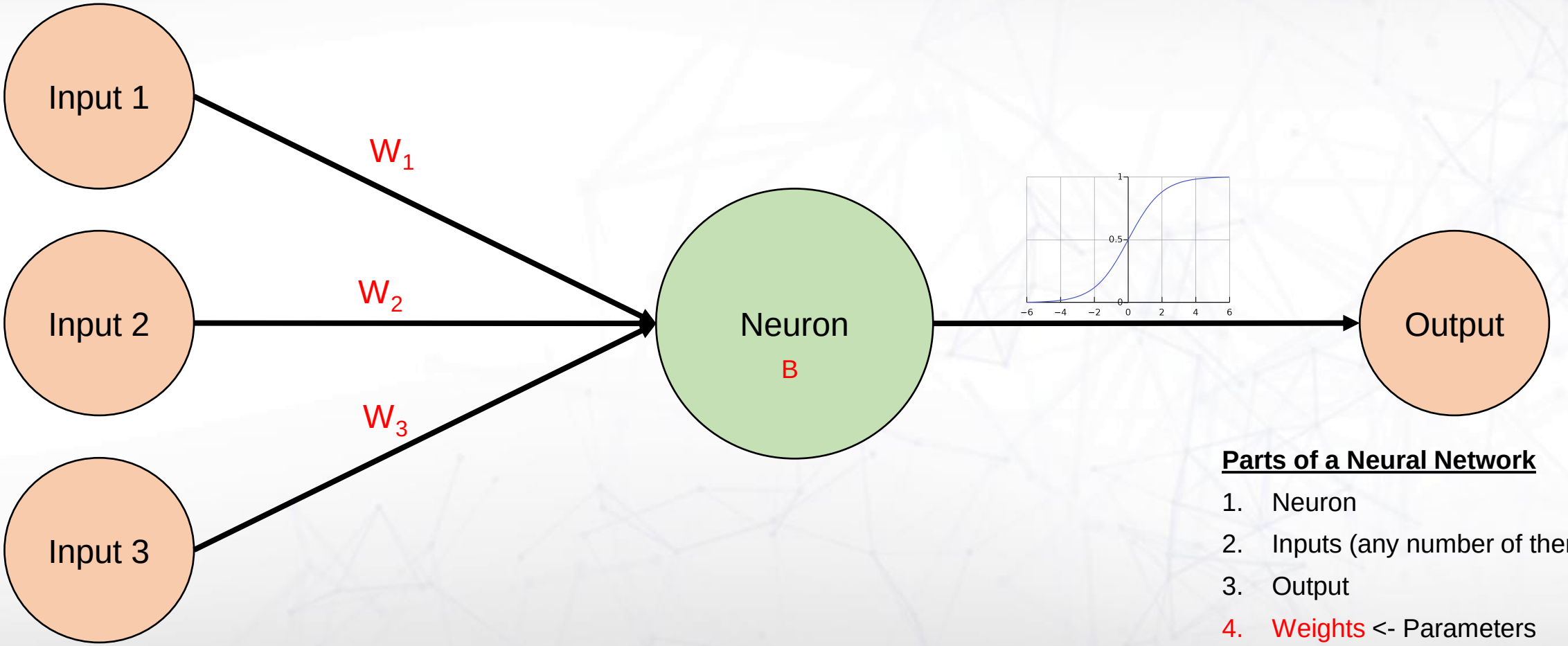


Parameters: All of the weights and biases in the NN

Loss:

- A mathematical equation to measure how well/poorly your model is doing at completing a task.

The Neuron



Parts of a Neural Network

1. Neuron
2. Inputs (any number of them)
3. Output
4. Weights <- Parameters
5. Bias <- Parameter
6. NonLinearity

The 3 Tenets of Deep Learning:

BRAINTANK DEEP LEARNING

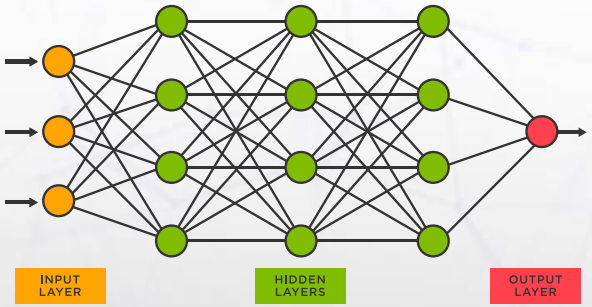
Dataset:

- A set of data that has information that we can use to teach the model how to preform better.

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
...	...	...	...	...	...

Model:

- Mathematical function (equation) that takes an input and creates a desired output. It is defined by its parameters, which are numbers we train/change to get a desired result.



Parameters: All of the weights and biases in the NN

Loss:

- A mathematical equation to measure how well/poorly your model is doing at completing a task.

The 3 Tenets of Deep Learning:

BRAINTANK DEEP LEARNING

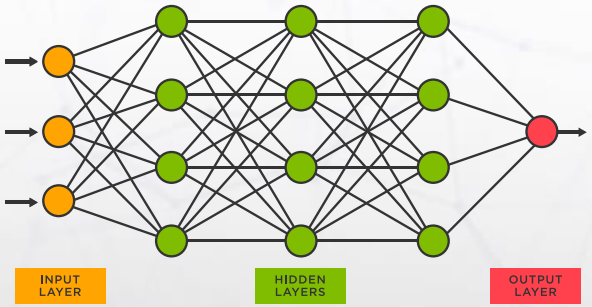
Dataset:

- A set of data that has information that we can use to teach the model how to preform better.

Diamond ID	Shape	Carat (Weight)	Colour	Clarity	Price (CAD)
2	Cushion	0.5	J	VS2	1700.94
28	Emerald	0.51	G	IF	1971.94
345	Round	1.09	H	VS1	11,724.09
6301	Princess	1.53	D	FL	23,434.00
432	Pear	0.3	D	VVS1	1500.00
5493	Emerald	5.6	J	VVS2	125,840.39
1230	Cushion	4.21	I	VVS1	86734.81
7000	Pear	3.7	D	VS1	63779.06
...	...	...	...	...	...

Model:

- Mathematical function (equation) that takes an input and creates a desired output. It is defined by its parameters, which are numbers we train/change to get a desired result.



Parameters: All of the weights and biases in the NN

Loss:

- A mathematical equation to measure how well/poorly your model is doing at completing a task.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

The Deep Learning Flow:

1. Initialize your model. Randomly assign your parameters a value.
2. Grab a single entry from the dataset. An input and the ground truth associated with that input.
3. Take that input and put it through the model and save the result.
4. Use the loss function to measure how different the result is from the ground truth.
5. Give the OPTIMIZER, it will then measure how it needs to change the parameters (numbers we change to get a desired result).
6. Repeat steps 2-5 many many times, until the model can perform the task you are asking it to do

The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

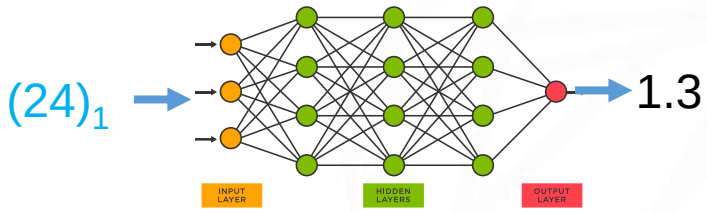
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



2,039031.526030981

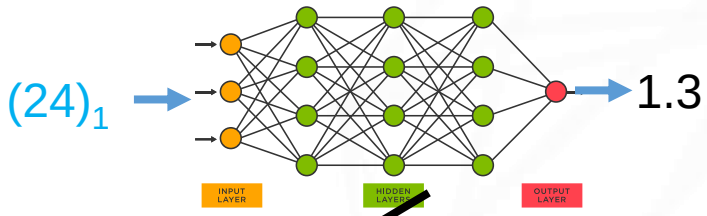
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



Result:

1.3

$2,039031.52

Ground Truth:

0.18

$4,371.04

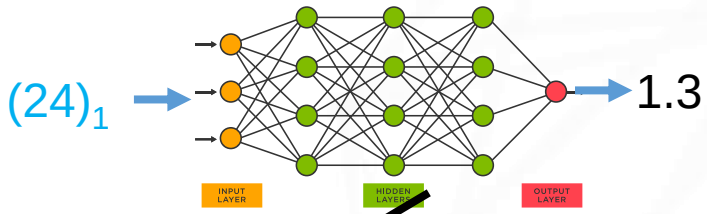
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



Result:

1.3

$2,039031.52

Ground Truth:

0.18

$4,371.04

Loss Function (measure of model success):

$L = (\text{Result} - \text{Ground Truth})^2$

$L = (1.3 - 0.18)^2$

$L = 1.254$

2,039031.526030981

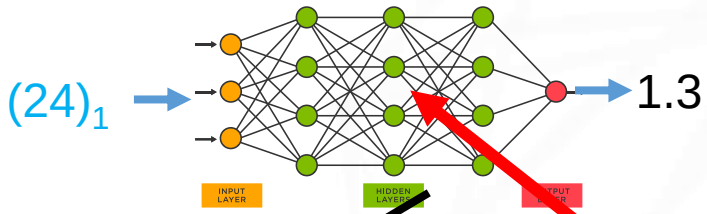
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



Result:

1.3

$2,039031.52

Ground Truth:

0.18

$4,371.04

Loss Function (measure of model success):

$L = (\text{Result} - \text{Ground Truth})^2$

$L = (1.3 - 0.18)^2$

$L = 1.254$

OPTIMIZER:

The OPTIMIZER will look at the loss, and use it to determine how to adjust the models parameters. We want the loss to be small and the model will try and reduce it for us.

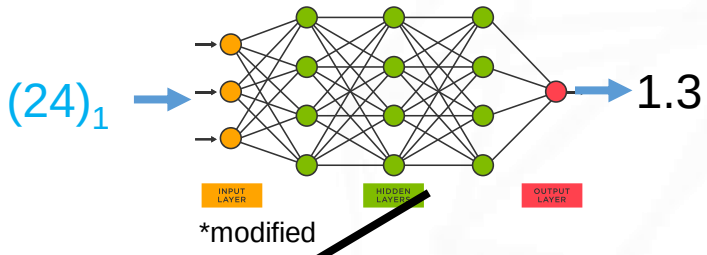
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



Result:

1.3

$2,039031.52

Ground Truth:

0.18

$4,371.04

Loss Function (measure of model success):

$L = (\text{Result} - \text{Ground Truth})^2$

$L = (1.3 - 0.18)^2$

$L = 1.254$

OPTIMIZER:

The OPTIMIZER will look at the loss, and use it to determine how to adjust the models parameters. We want the loss to be small and the model will try and reduce it for us.

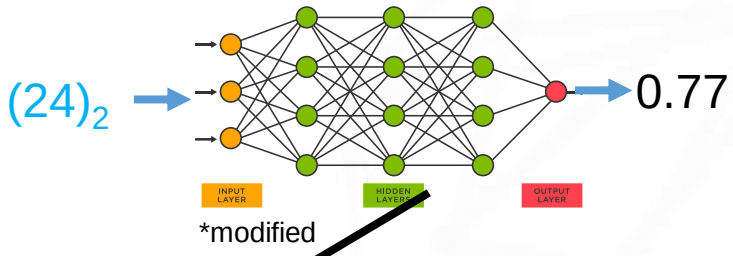
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



Result:

0.77

$111,316.61

Ground Truth:

0.52

$28,240.31

Loss Function (measure of model success):

$L = (\text{Result} - \text{Ground Truth})^2$

$L = (0.77 - 0.52)^2$

$L = 0.0625$

OPTIMIZER:

The OPTIMIZER will look at the loss, and use it to determine how to adjust the models parameters. We want the loss to be small and the model will try and reduce it for us.

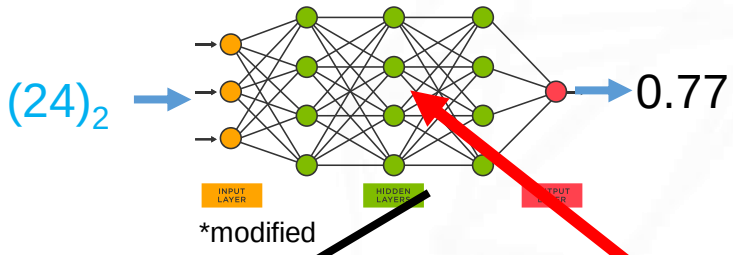
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



Result:

0.77

$111,316.61

Ground Truth:

0.52

$28,240.31

Loss Function (measure of model success):

$L = (\text{Result} - \text{Ground Truth})^2$

$L = (0.77 - 0.52)^2$

$L = 0.0625$

OPTIMIZER:

The OPTIMIZER will look at the loss, and use it to determine how to adjust the models parameters. We want the loss to be small and the model will try and reduce it for us.

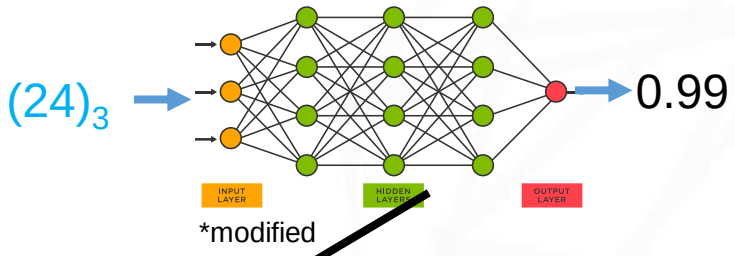
The Deep Learning Flow:

BRAINTANK DEEP LEARNING

Dataset:

Price of Diamond	Input Features
0.18	$(24)_1$
0.52	$(24)_2$
1.00	$(24)_3$
0.00	$(24)_4$
0.35	$(24)_5$
…	

Model:



*modified

Result:

0.99

$372,192.54

Ground Truth:

1.00

$393,183.50

Loss Function (measure of model success):

$L = (\text{Result} - \text{Ground Truth})^2$

$L = (0.99 - 1.00)^2$

$L = 0.0001$

OPTIMIZER:

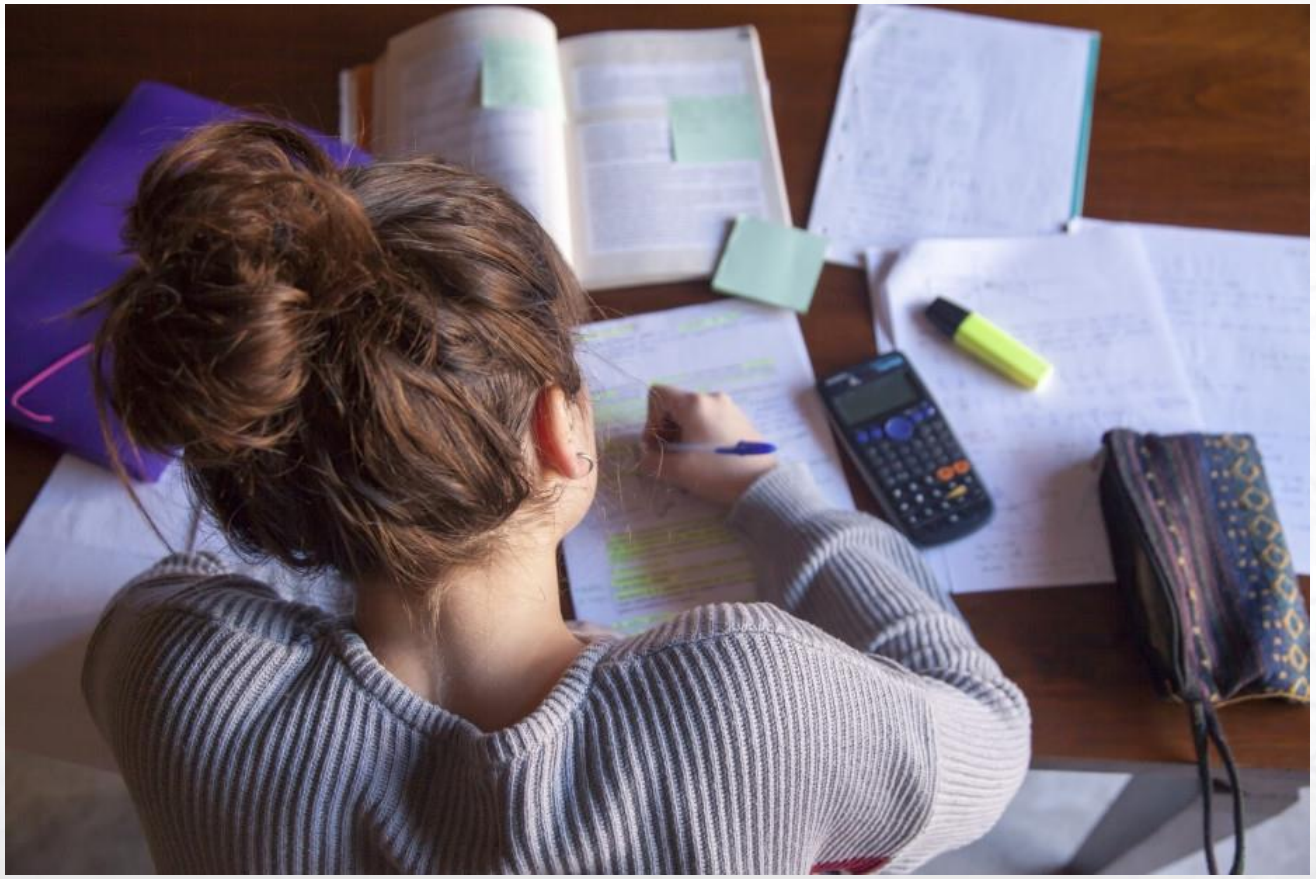
The OPTIMIZER will look at the loss, and use it to determine how to adjust the models parameters. We want the loss to be small and the model will try and reduce it for us.

Training and Testing

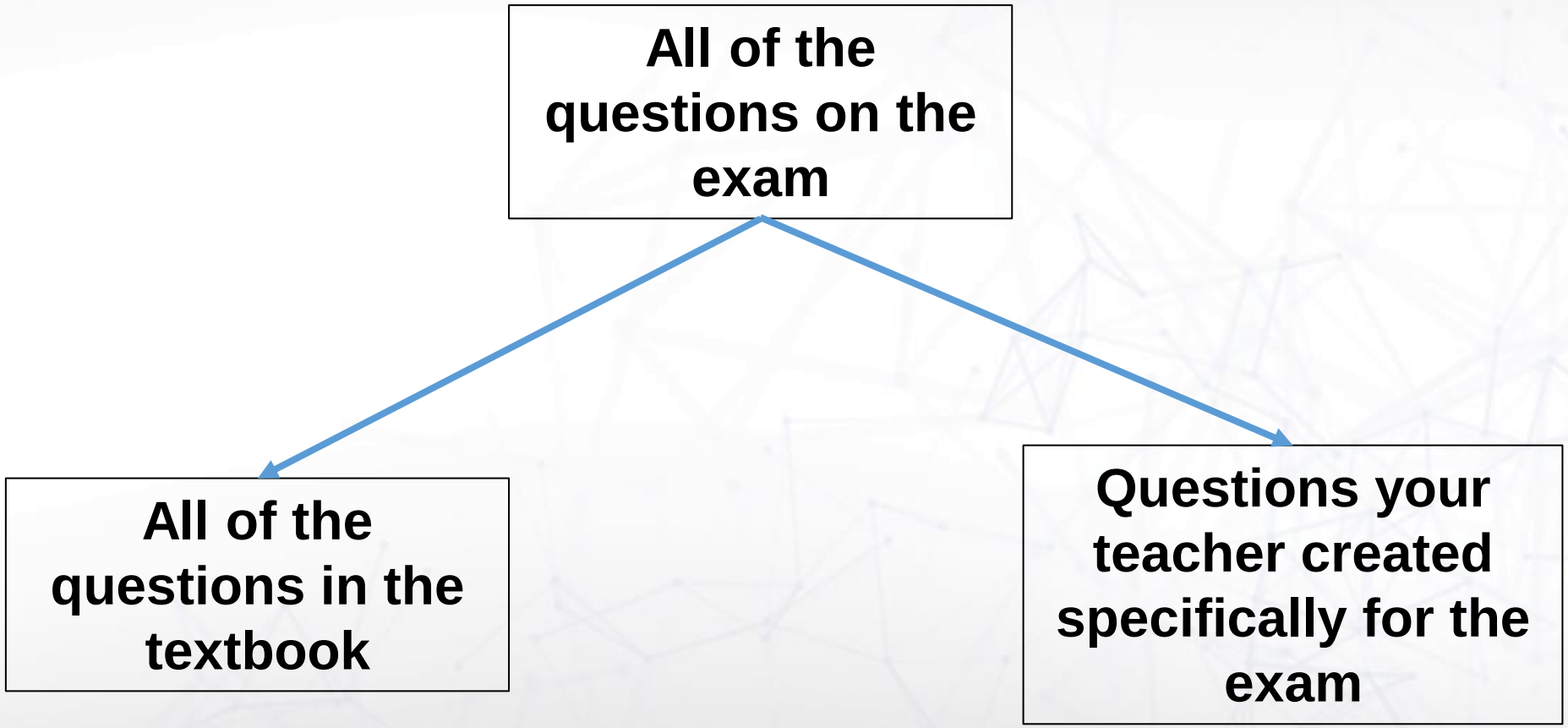
BRAINTANK
DEEP LEARNING

How do we know our model is doing a good job at performing the task we are asking it to do?

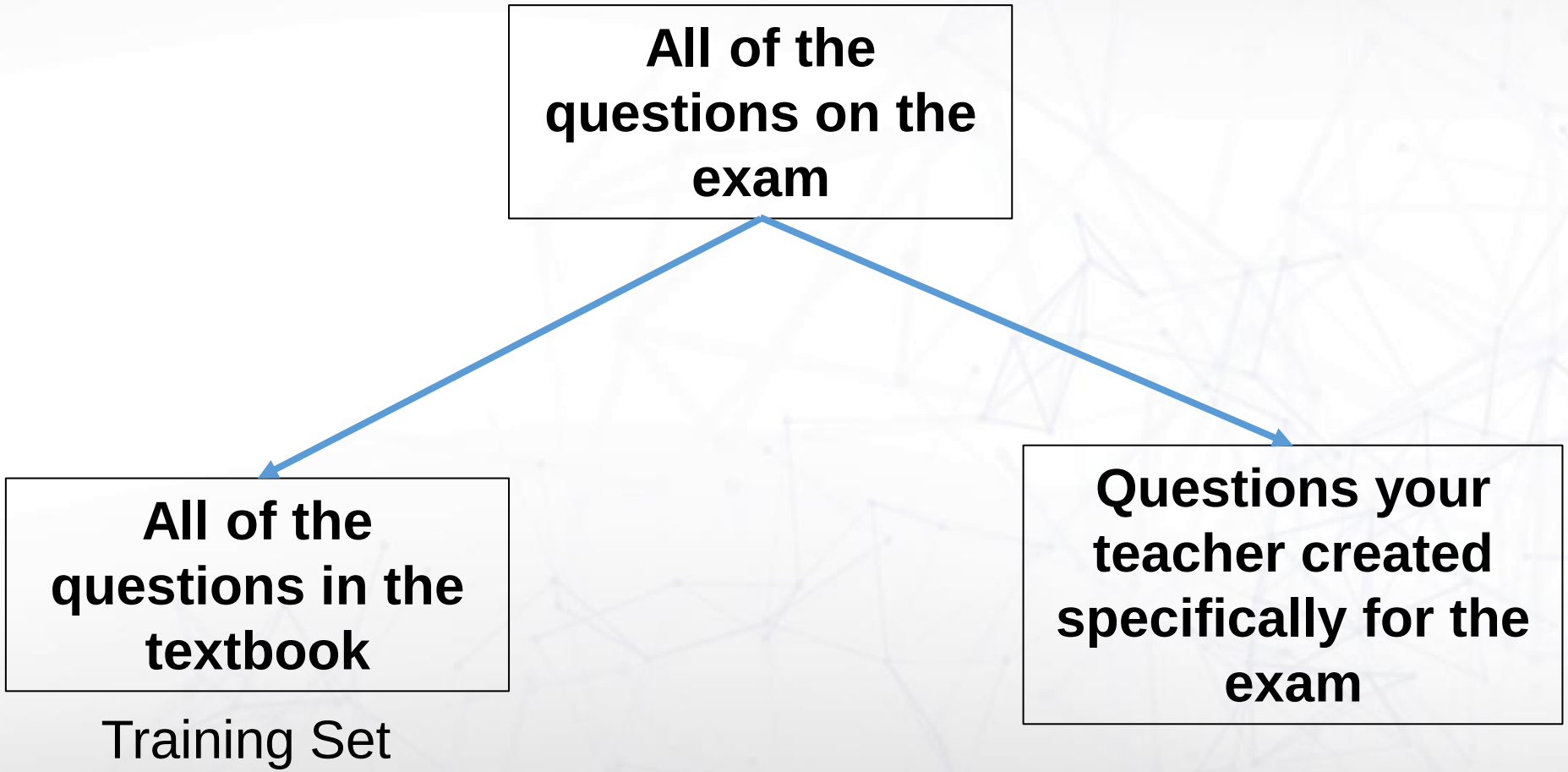
Studying for a test



Studying for a test

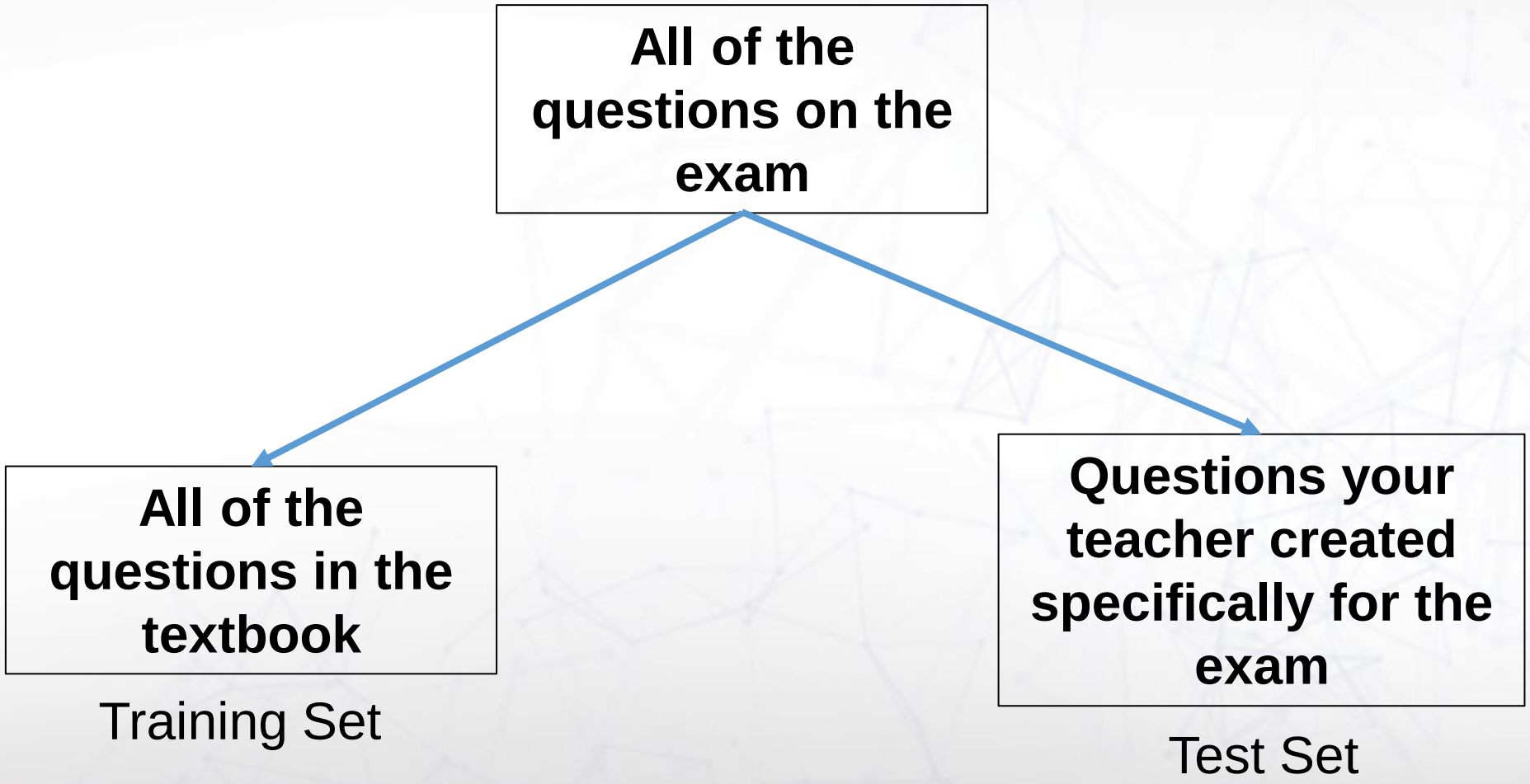


Studying for a test

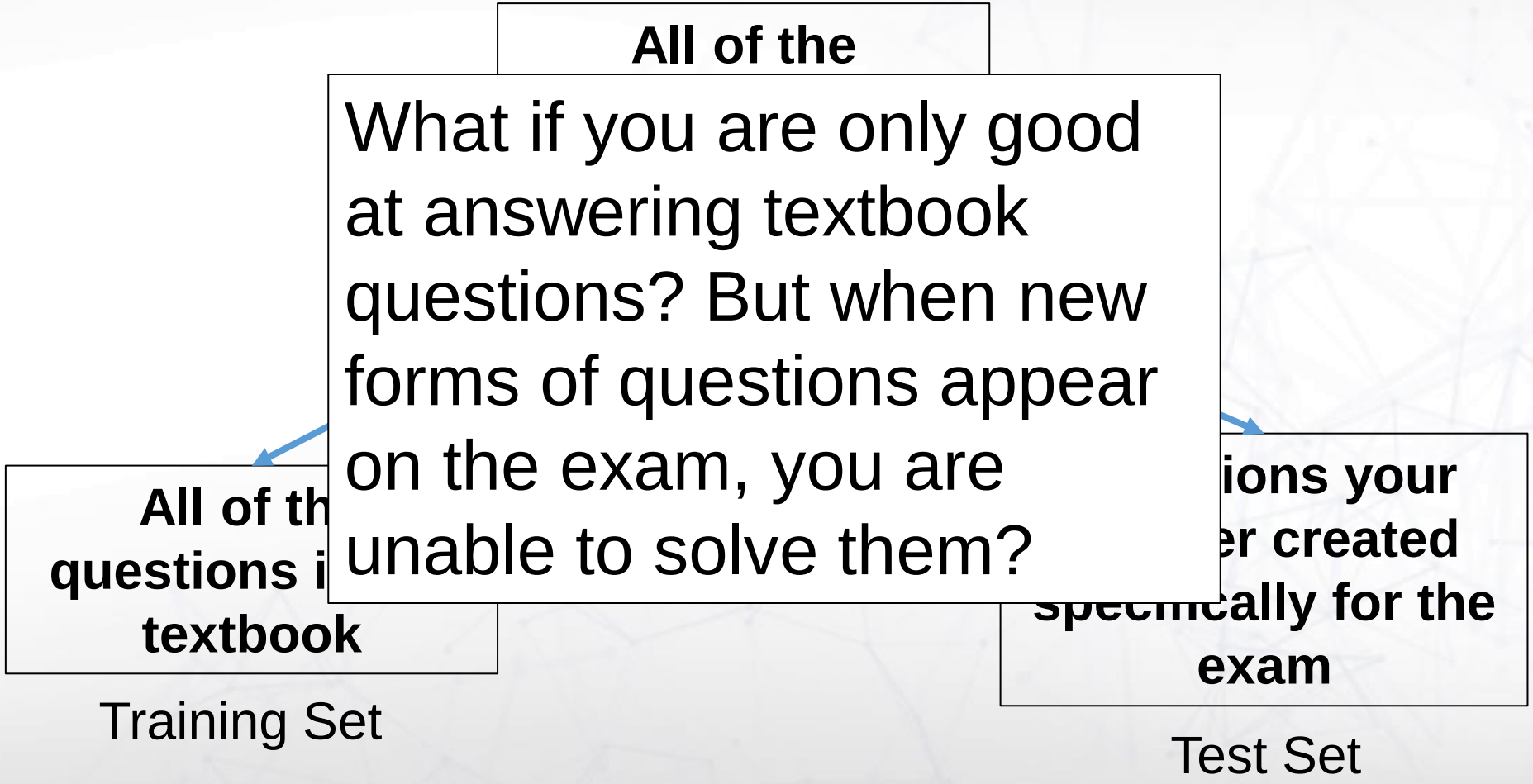


Training Set

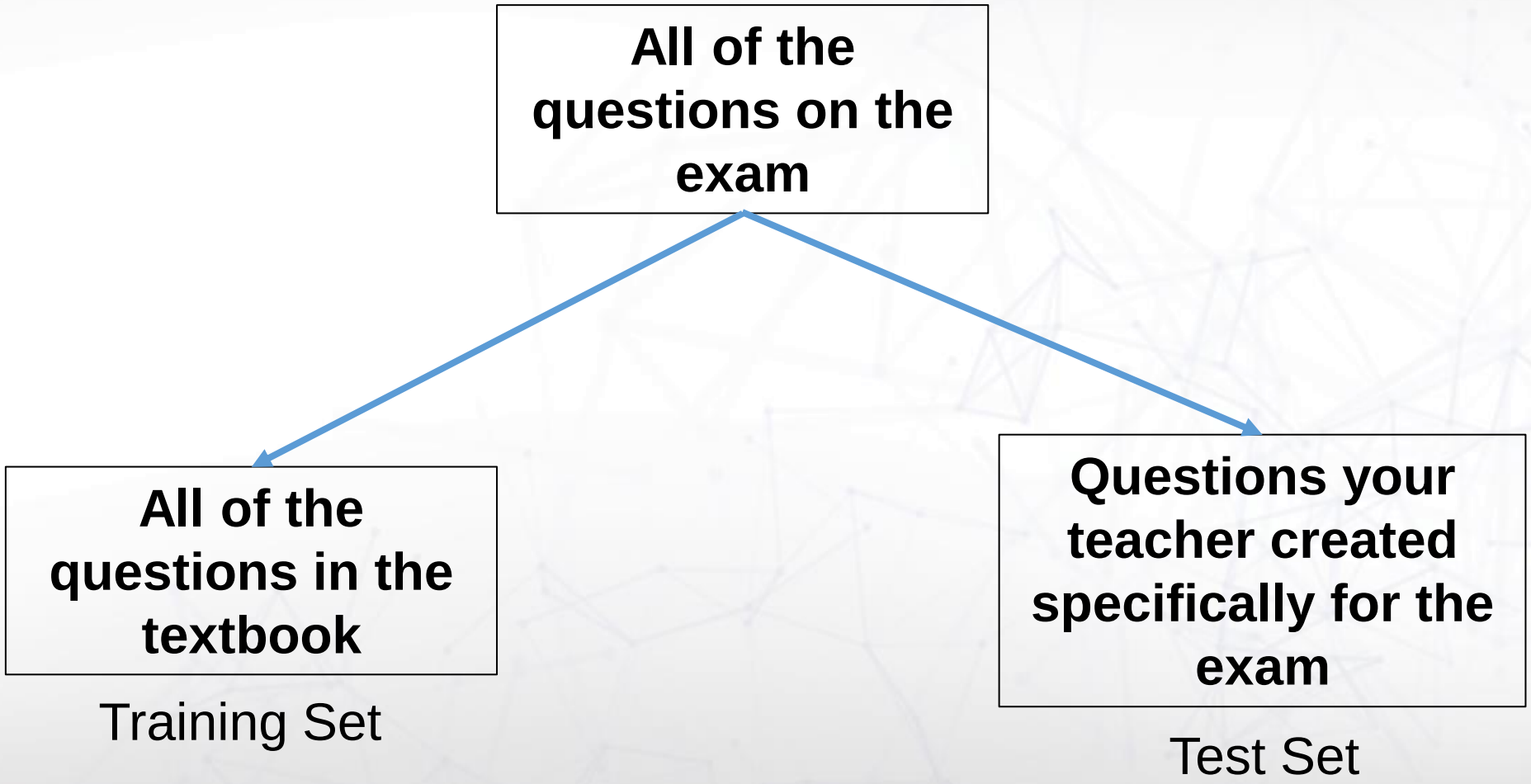
Studying for a test



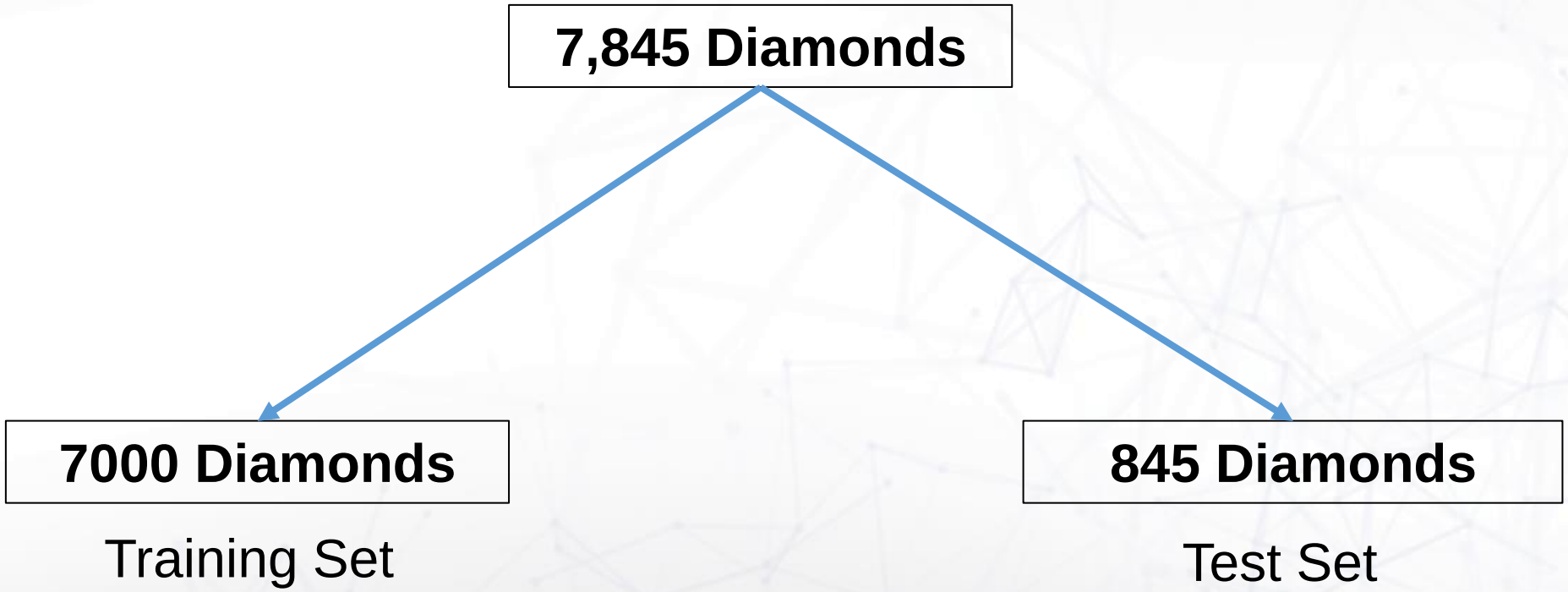
Studying for a test



Studying for a test



How our model studies for a test



How our model studies for a test

If our model is able to correctly predict prices of diamonds in the test set, it is a strong indicator that our model has learned the features that effect diamond prices (good). Instead of memorizing and copying the textbook (bad)

7000

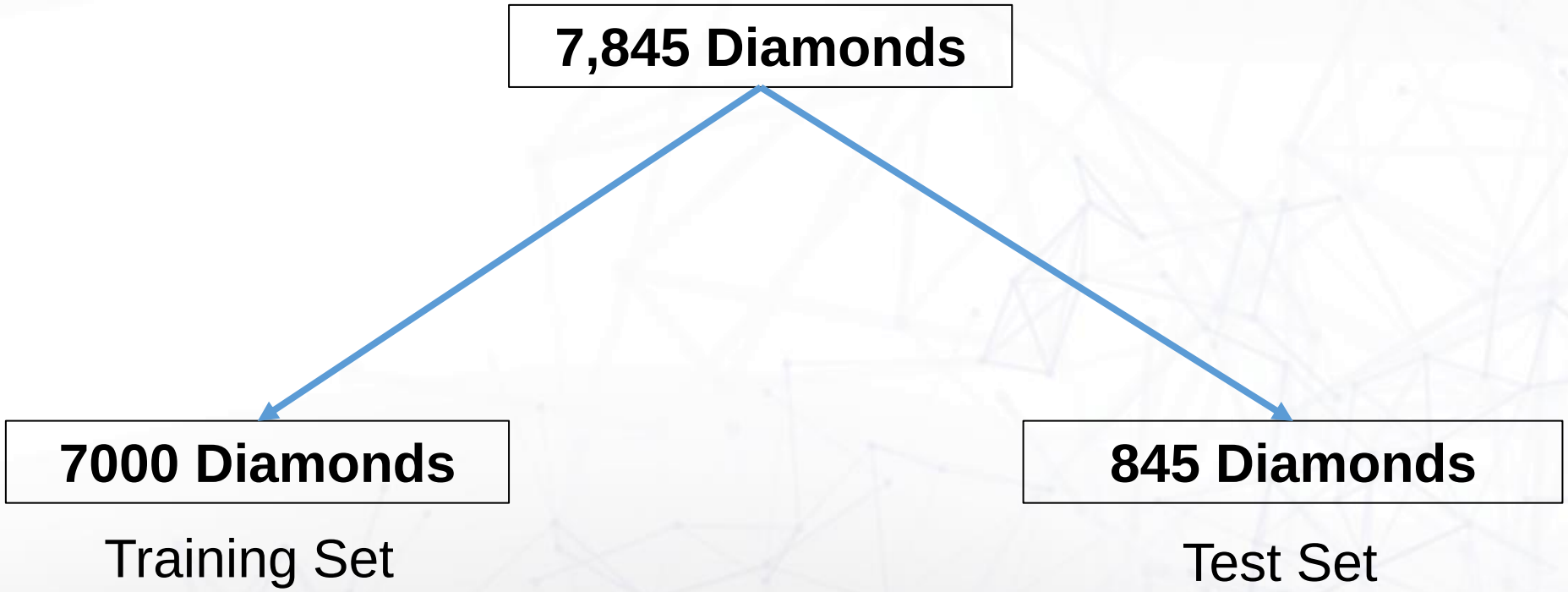
Tra

We train the model on these 7000 diamonds

nds

We test the model on these 845 never before seen diamonds

How our model studies for a test



BRAINTANK

DEEP LEARNING