CAP 6635 – Artificial Intelligence

Lecture 12: How do machines learn? (Part 4)

Oge Marques, PhD

Professor

College of Engineering and Computer Science

College of Business

FAU FLORIDA ATLANTIC UNIVERSITY

@ProfessorOge

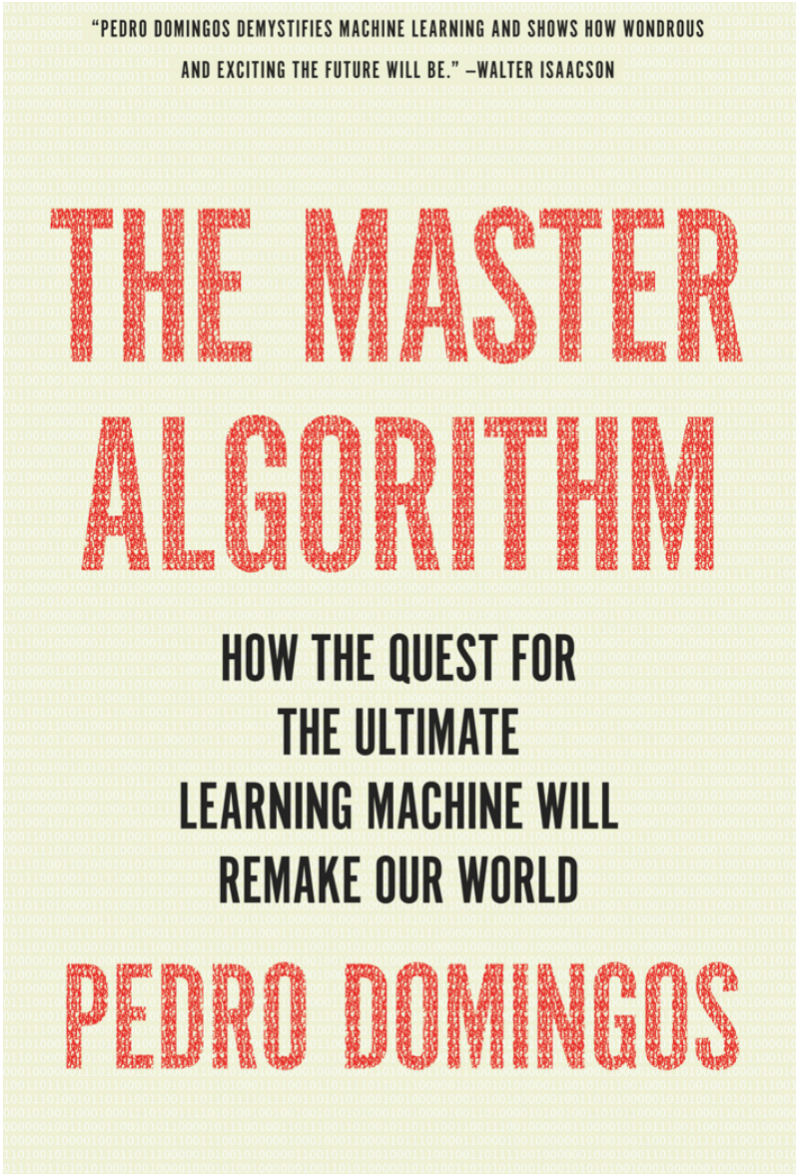
ProfessorOgeMarques


Previously
on CAP 6635…

The Master Algorithm

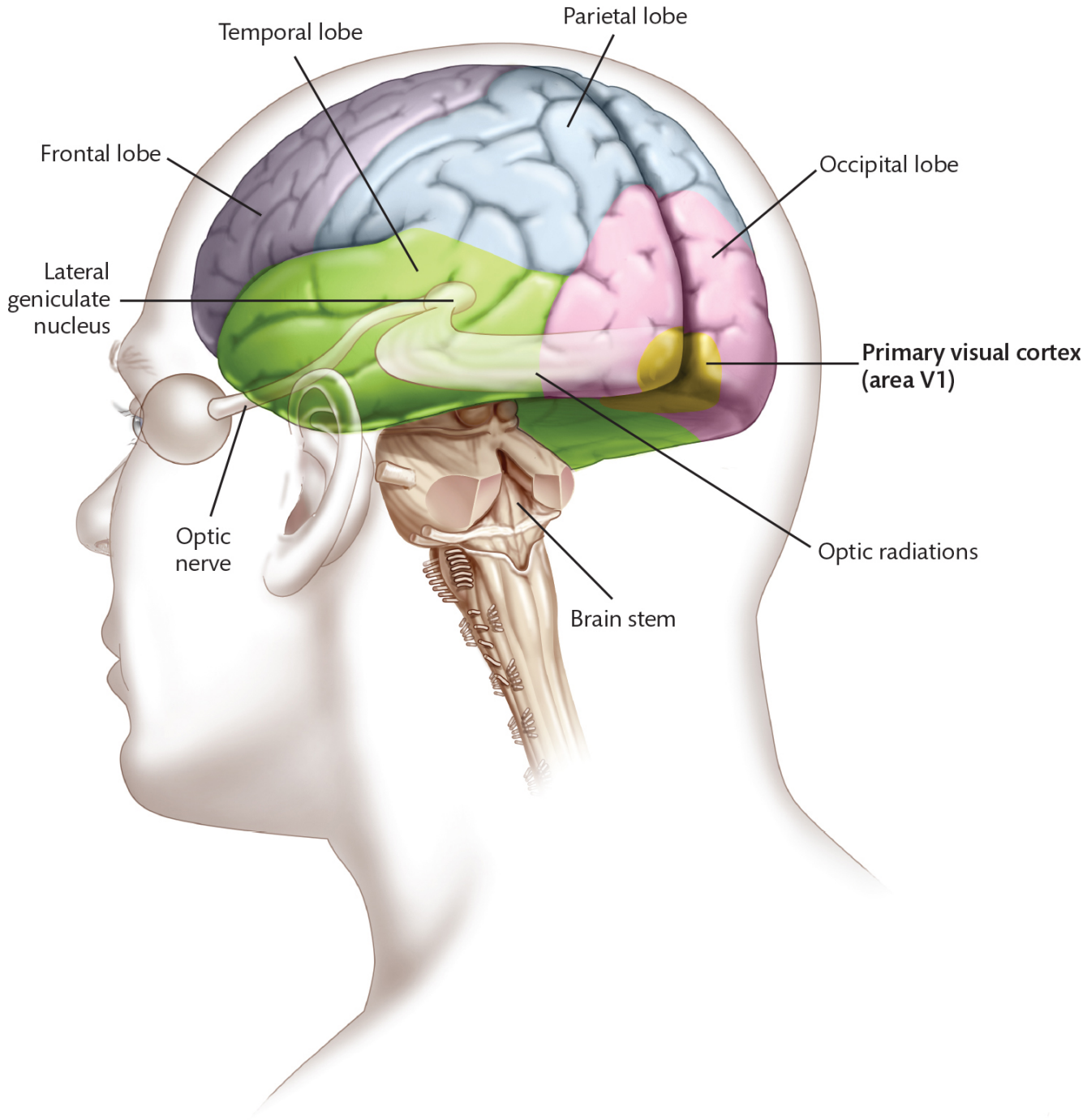
(by Pedro Domingos)

"All knowledge—past, present, and future—can be derived from data by a single, universal learning algorithm."



The connectionists

Our source of inspiration



Fundamental questions

What do we know about the (human) brain?

How much do we know about the (human) brain?

How do we learn about the (human) brain?

What about **other animals' brains**?

What can we achieve by **reverse engineering** the brain?

Biological neuron

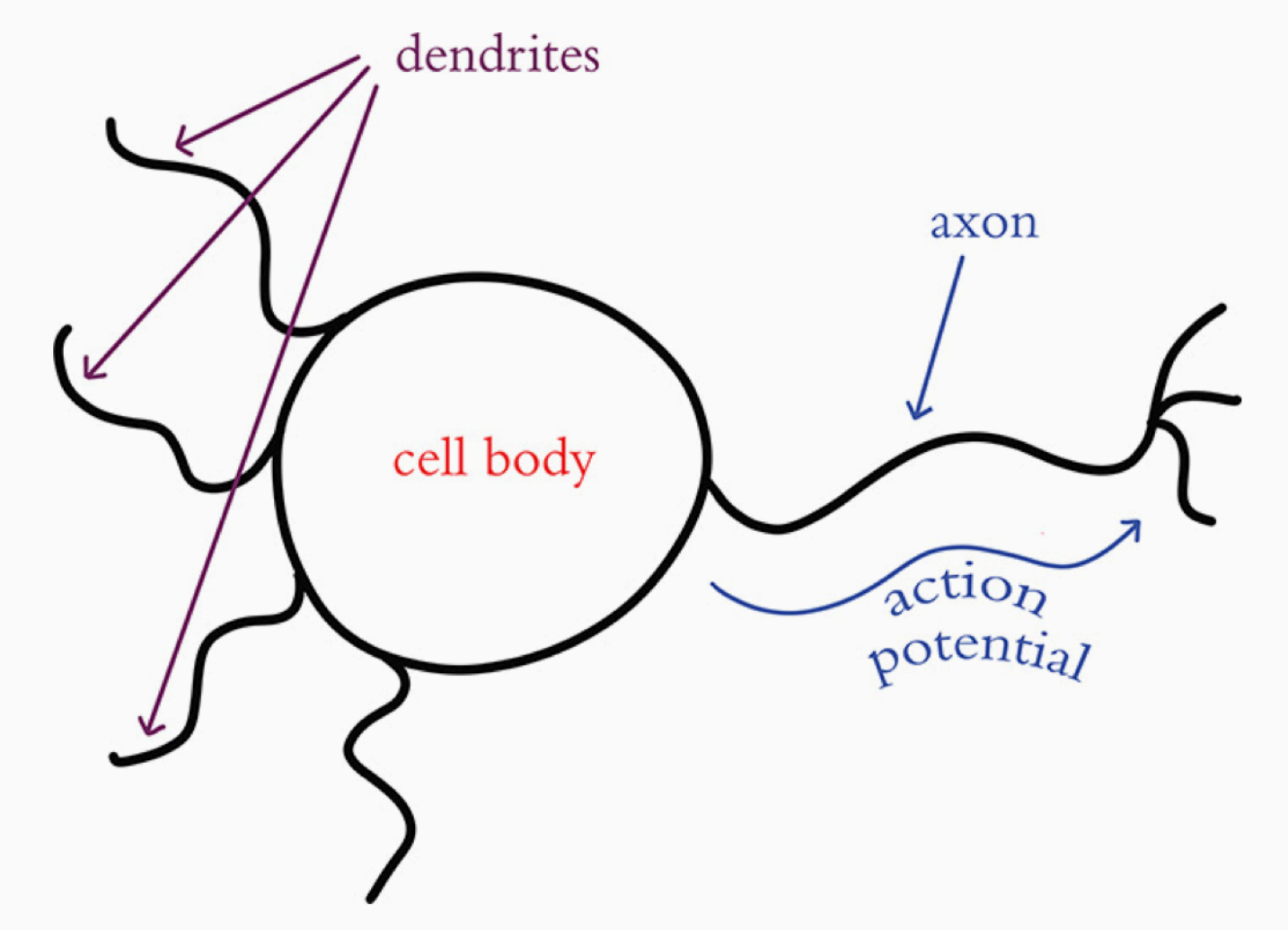
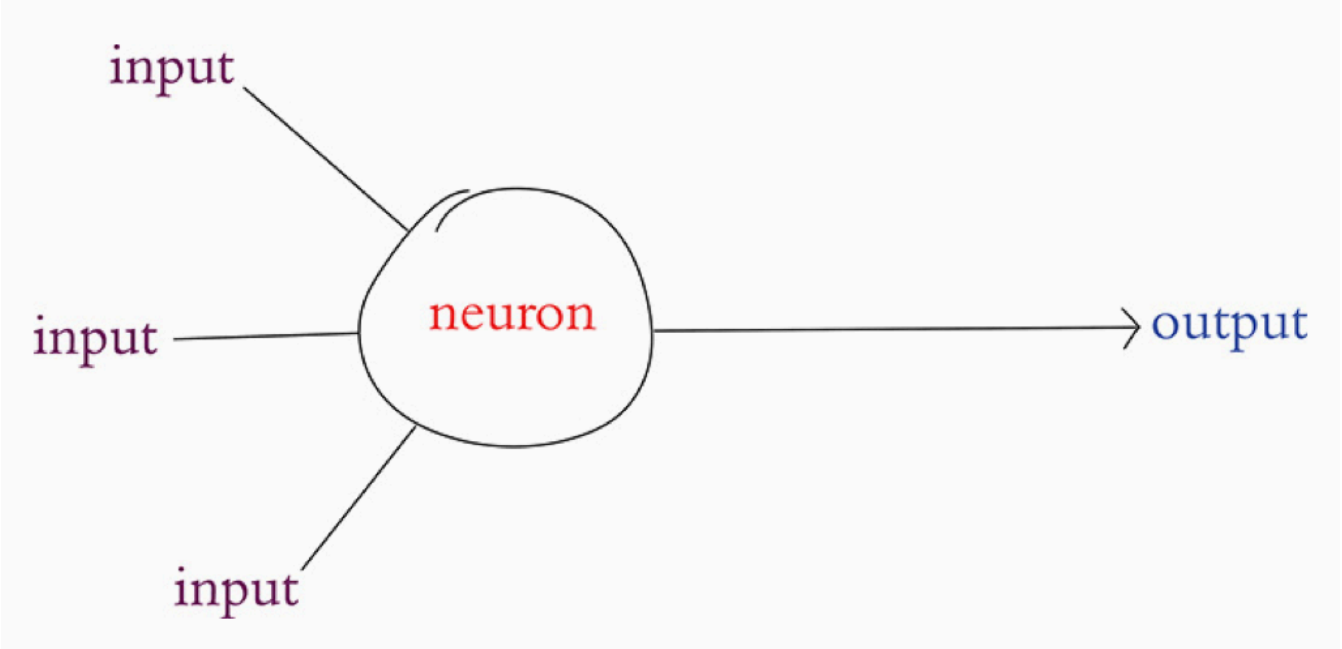


Figure from: “Deep Learning Illustrated” (Krohn, 2020)

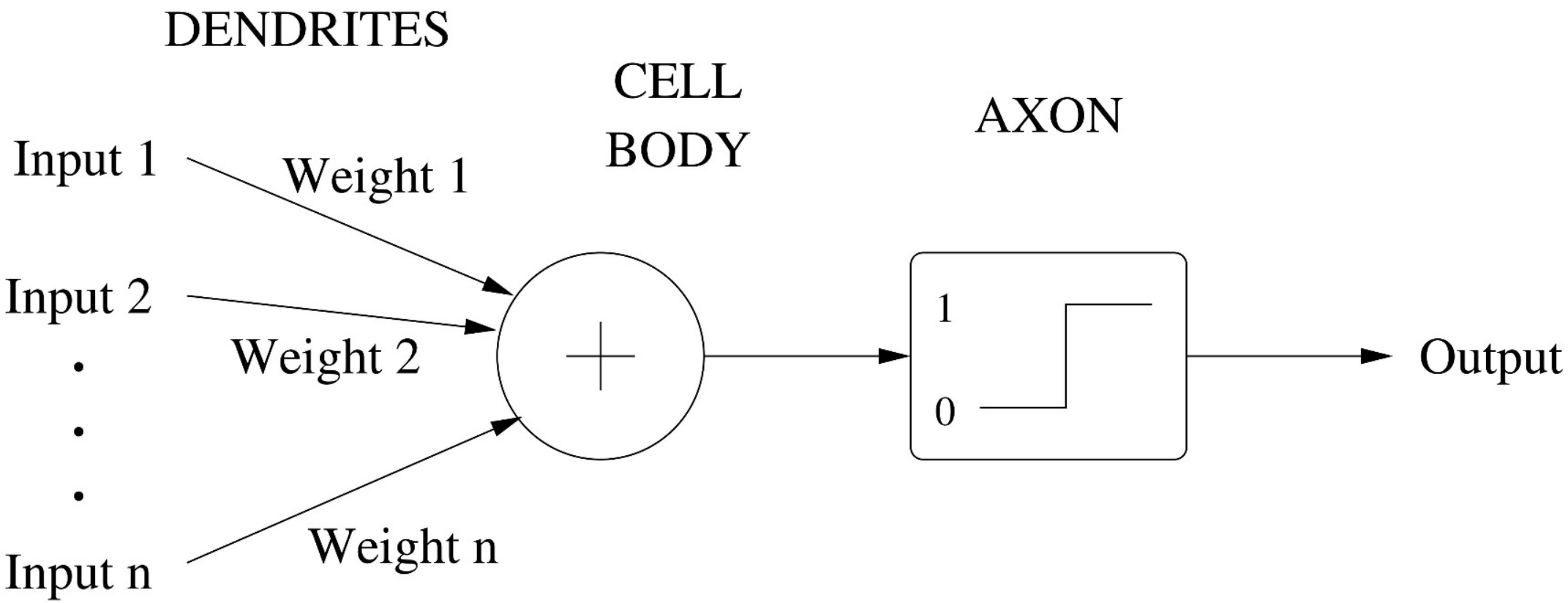
The perceptron



"Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and the organization in the brain. Psychological Review, 65, 386–408."

Figure from: "Deep Learning Illustrated" (Krohn, 2020)

An Artificial Neuron (Perceptron)



Source: P. Domingos

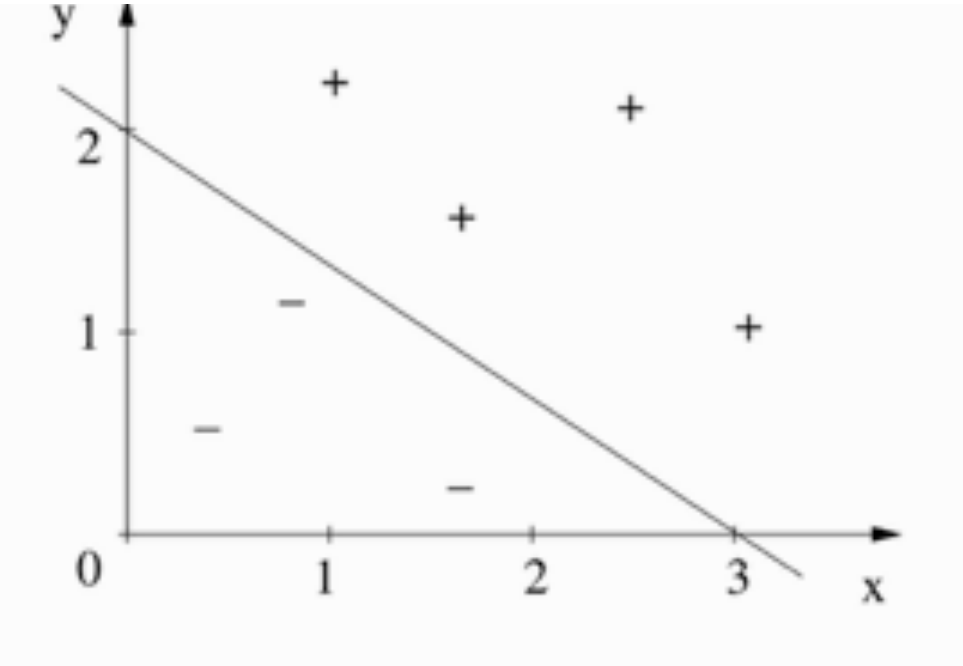
The Perceptron

- In a perceptron, a positive weight represents an *excitatory* connection, and a negative weight an *inhibitory* one. T
- he perceptron outputs 1 if the weighted sum of its inputs is above threshold, and 0 if it's below.
- By varying the weights and threshold, we can change the function that the perceptron computes.
- This ignores a lot of the details of how neurons work, of course, but we want to keep things as simple as possible; our goal is to develop a general-purpose learning algorithm, not to build a realistic model of the brain.

The Perceptron: example

- The boundary is the set of points where the weighted sum exactly equals the threshold, and a weighted sum is a linear function.
- For example, if the weights are 2 for x and 3 for y and the threshold is 6, the boundary is defined by the equation

$$2x + 3y = 6$$



The general equation for artificial neurons

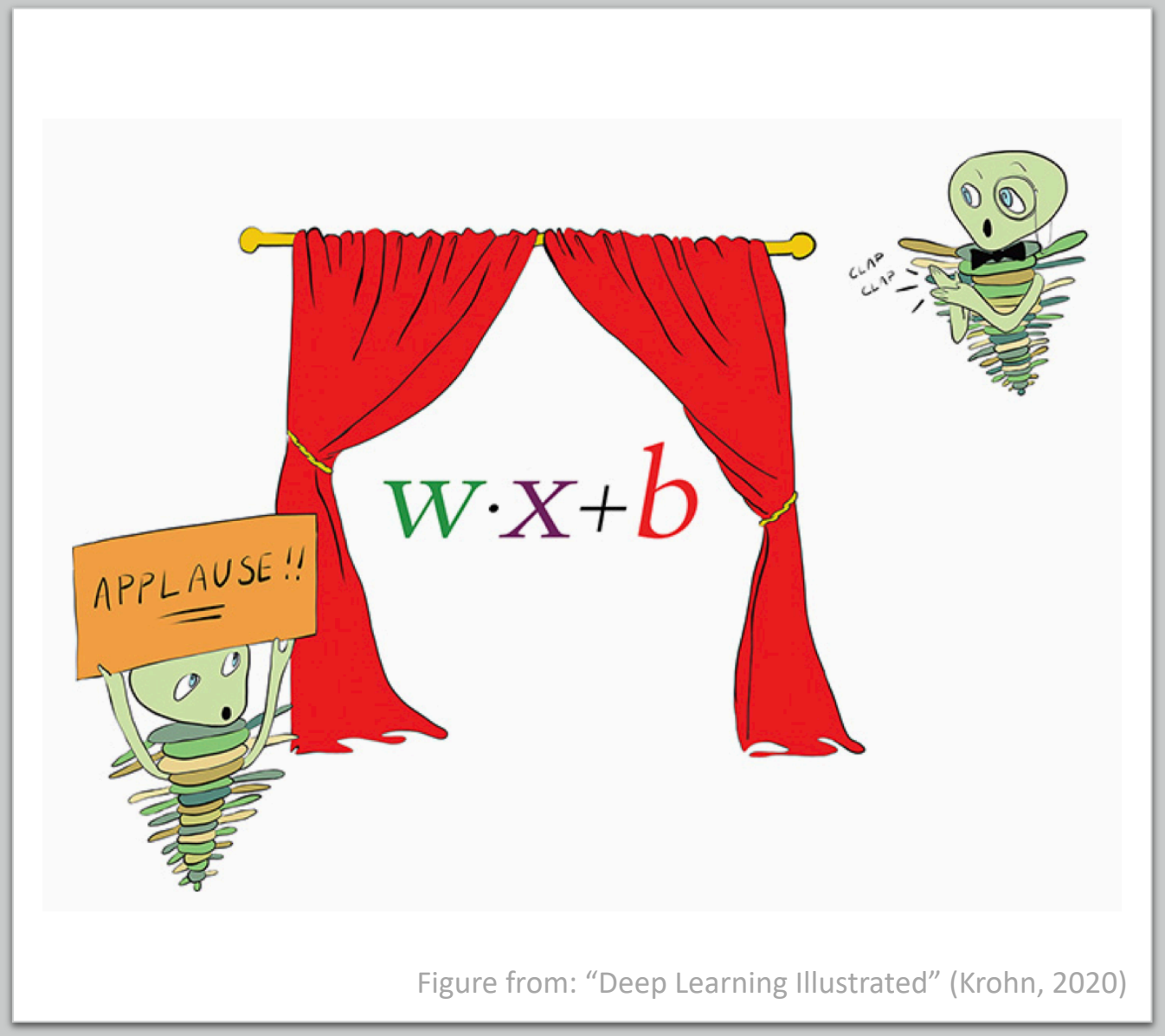
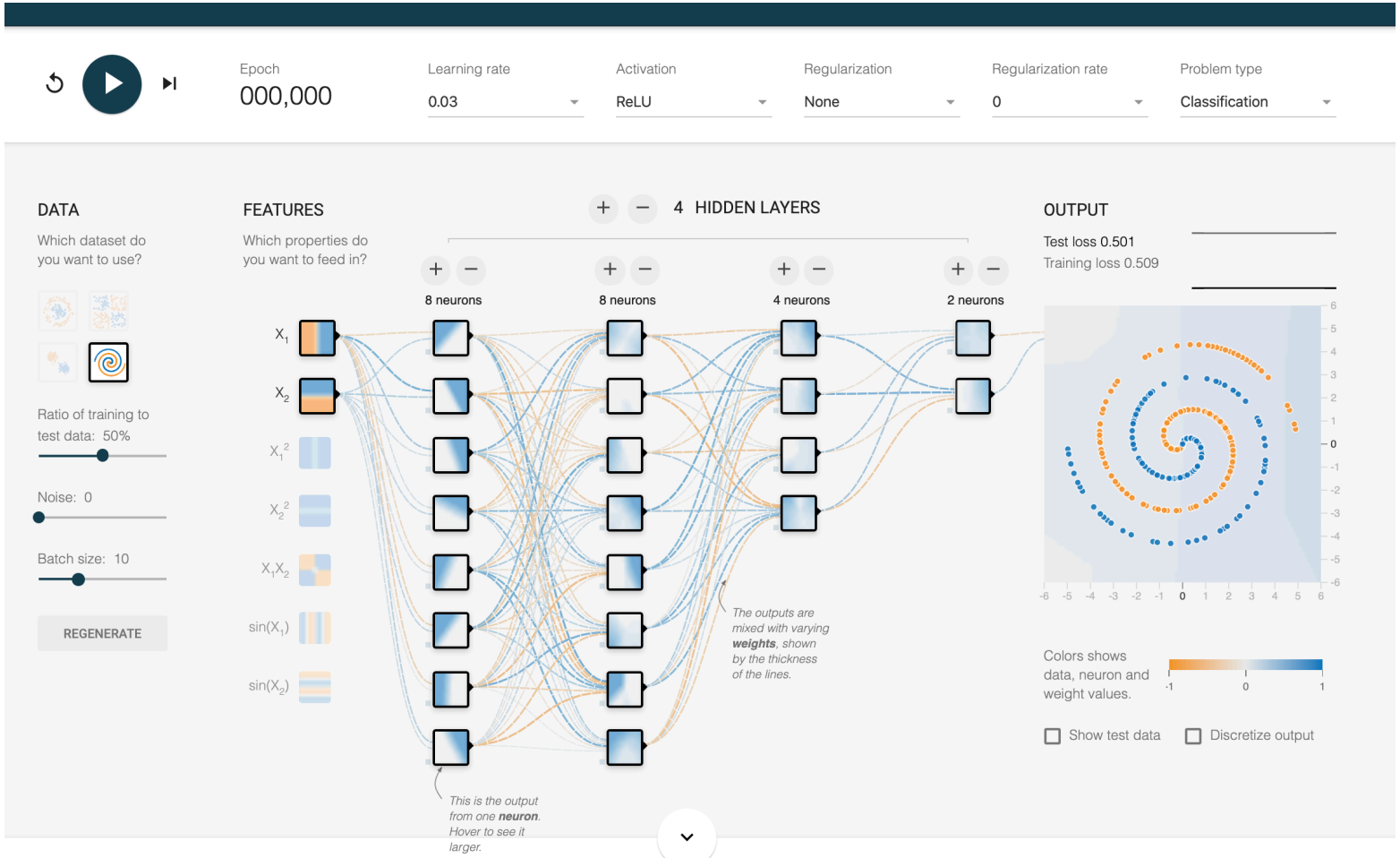


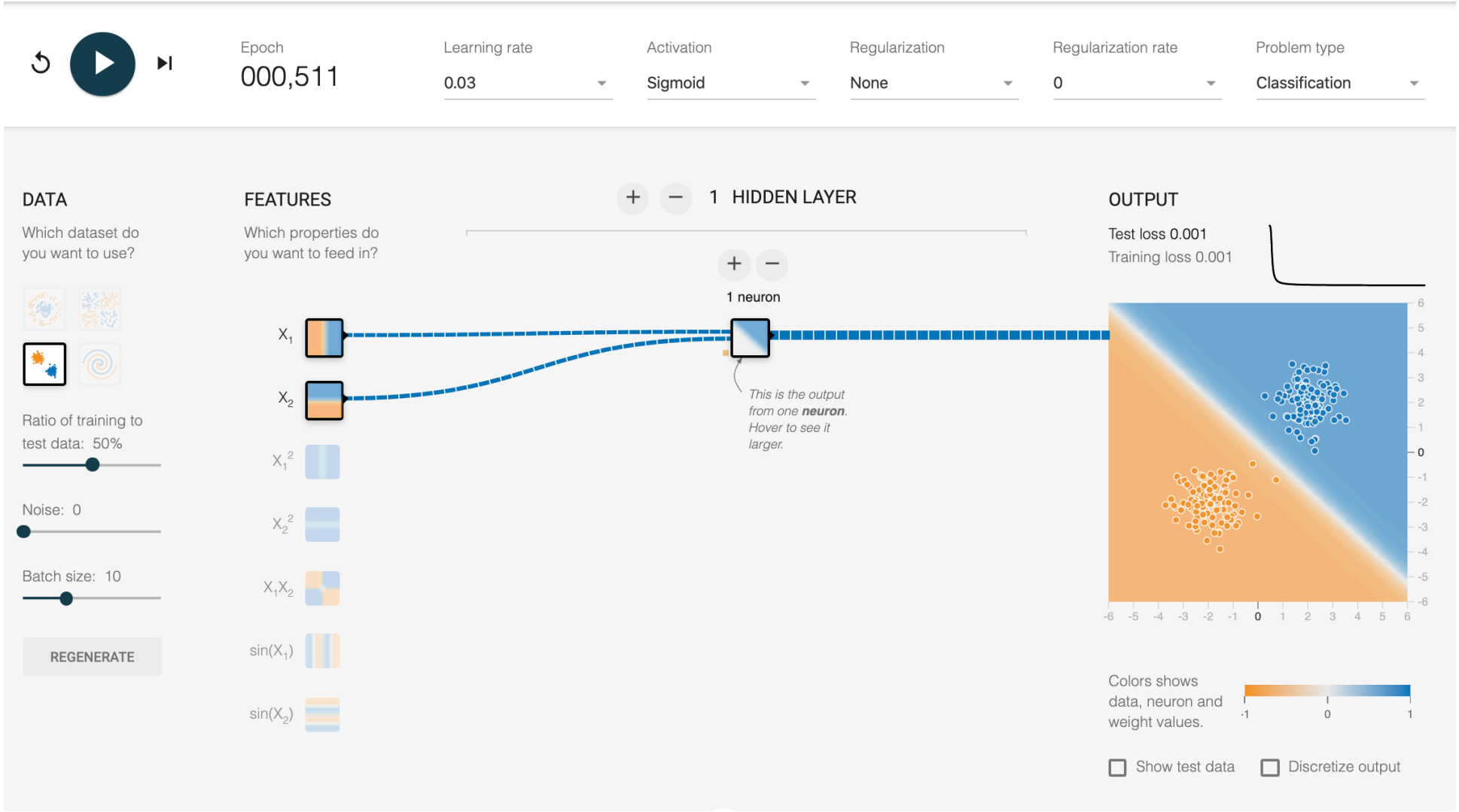
Figure from: “Deep Learning Illustrated” (Krohn, 2020)

TensorFlow Playground

https://playground.tensorflow.org/



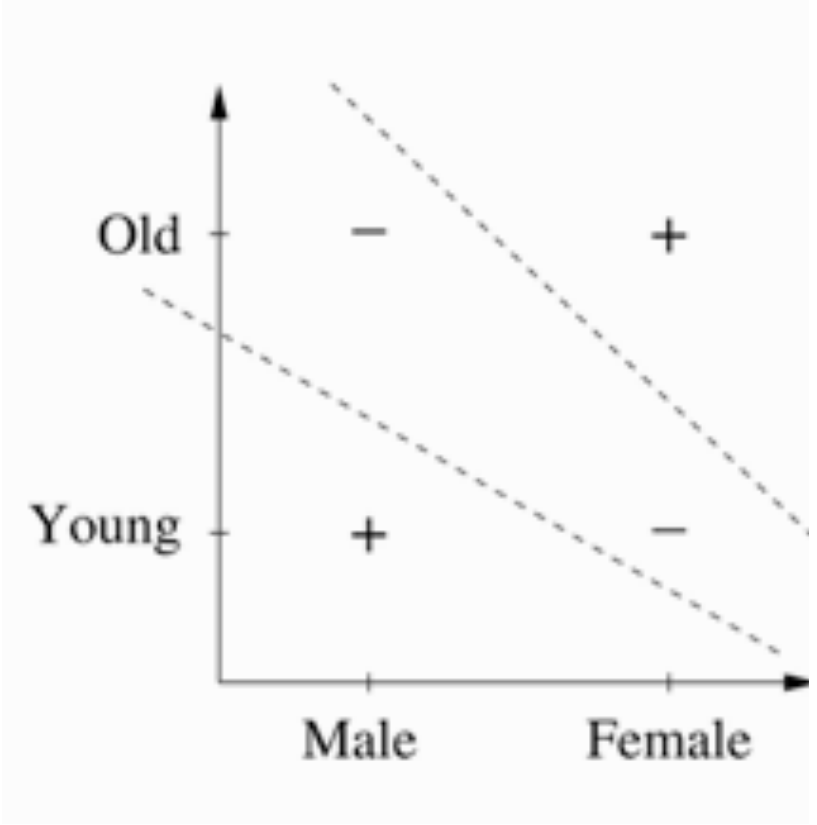
Playground time!



The XOR problem

- The problem with XOR is that there is no straight line capable of separating the positive from the negative examples.

- Since perceptrons can only learn linear boundaries, they can't learn XOR. And if they can't do even that, they're not a very good model of how the brain learns, or a viable candidate for the Master Algorithm.



The XOR problem

- Problem: a perceptron models only a single neuron's learning.
- Possible solution: add a hidden layer.

The problem is that there's no clear way to change the weights of the neurons in the "hidden" layers to reduce the errors made by the ones in the output layer.

Every hidden neuron influences the output via multiple paths, and every error has a thousand fathers.

Who do you blame? Or, conversely, who gets the credit for correct outputs? This credit-assignment problem shows up whenever we try to learn a complex model and is one of the central problems in machine learning.

Playground time!

Simulate the XOR problem in TensorFlow Playground.

Search (by trial-and-error) for a solution (there are many!).

Record your steps and findings.

Write down your conclusions and questions.

Epoch
000,365

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

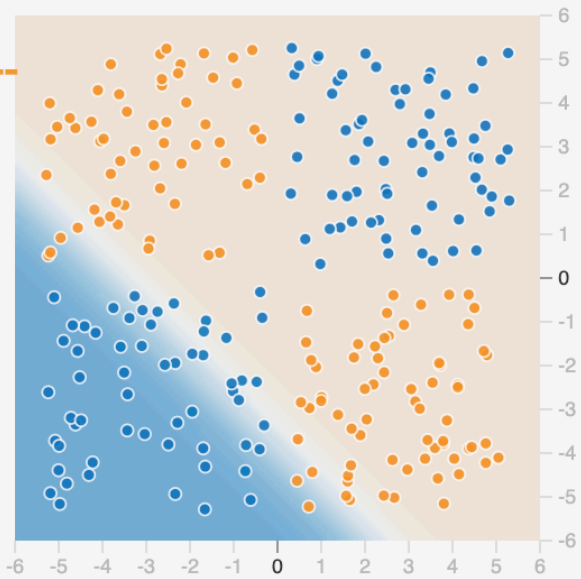
1 HIDDEN LAYER

1 neuron

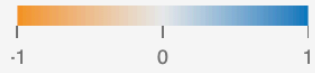
*This is the output from one **neuron**. Hover to see it larger.*

OUTPUT

Test loss 0.400
Training loss 0.420



Colors shows data, neuron and weight values.



Show test data

Discretize output

Epoch
000,238

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

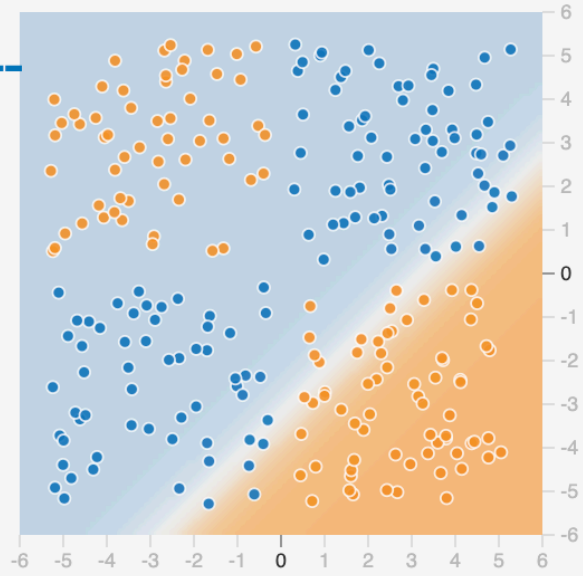
1 HIDDEN LAYER

1 neuron

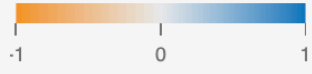
*This is the output from one **neuron**. Hover to see it larger.*

OUTPUT

Test loss 0.402
Training loss 0.373



Colors shows data, neuron and weight values.



Show test data

Discretize output

Epoch
000,263

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

2 HIDDEN LAYERS

4 neurons

1 neuron

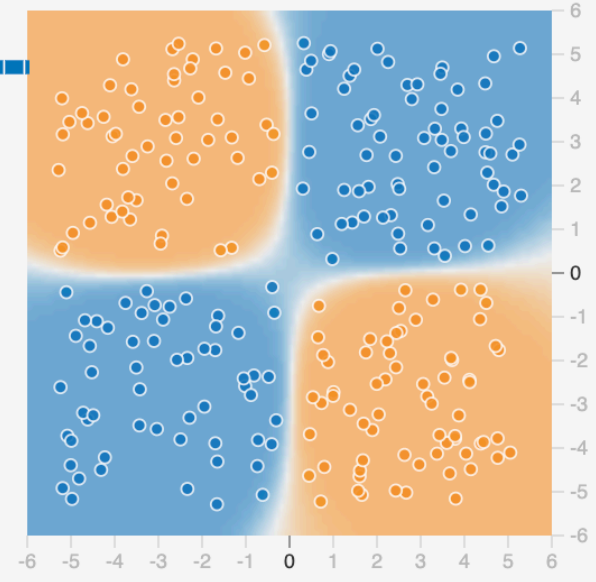
*This is the output from one **neuron**. Hover to see it larger.*

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

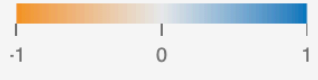
OUTPUT

Test loss 0.005

Training loss 0.002



Colors shows data, neuron and weight values.

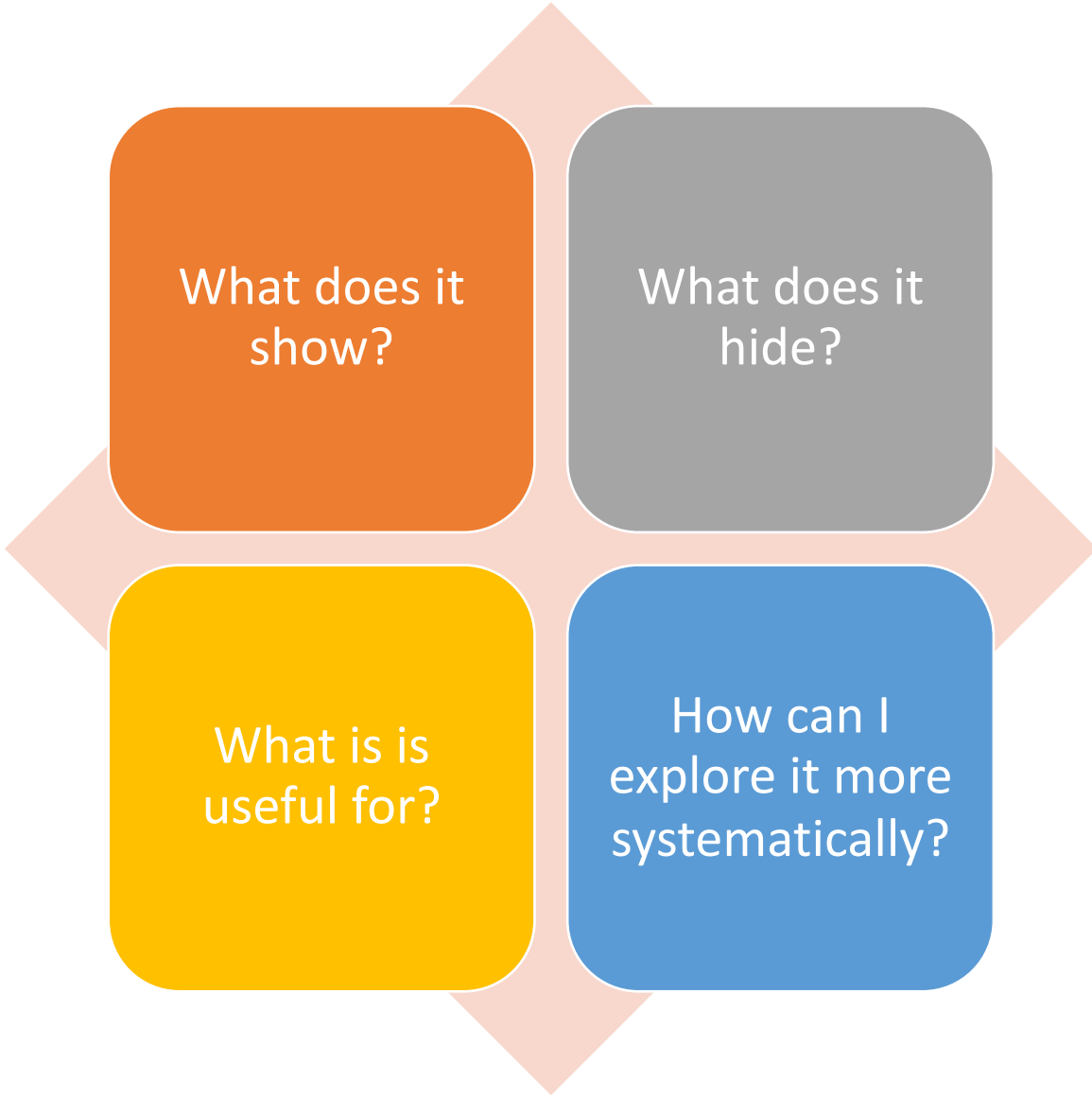


Show test data

Discretize output

New material starts here…

TensorFlow Playground

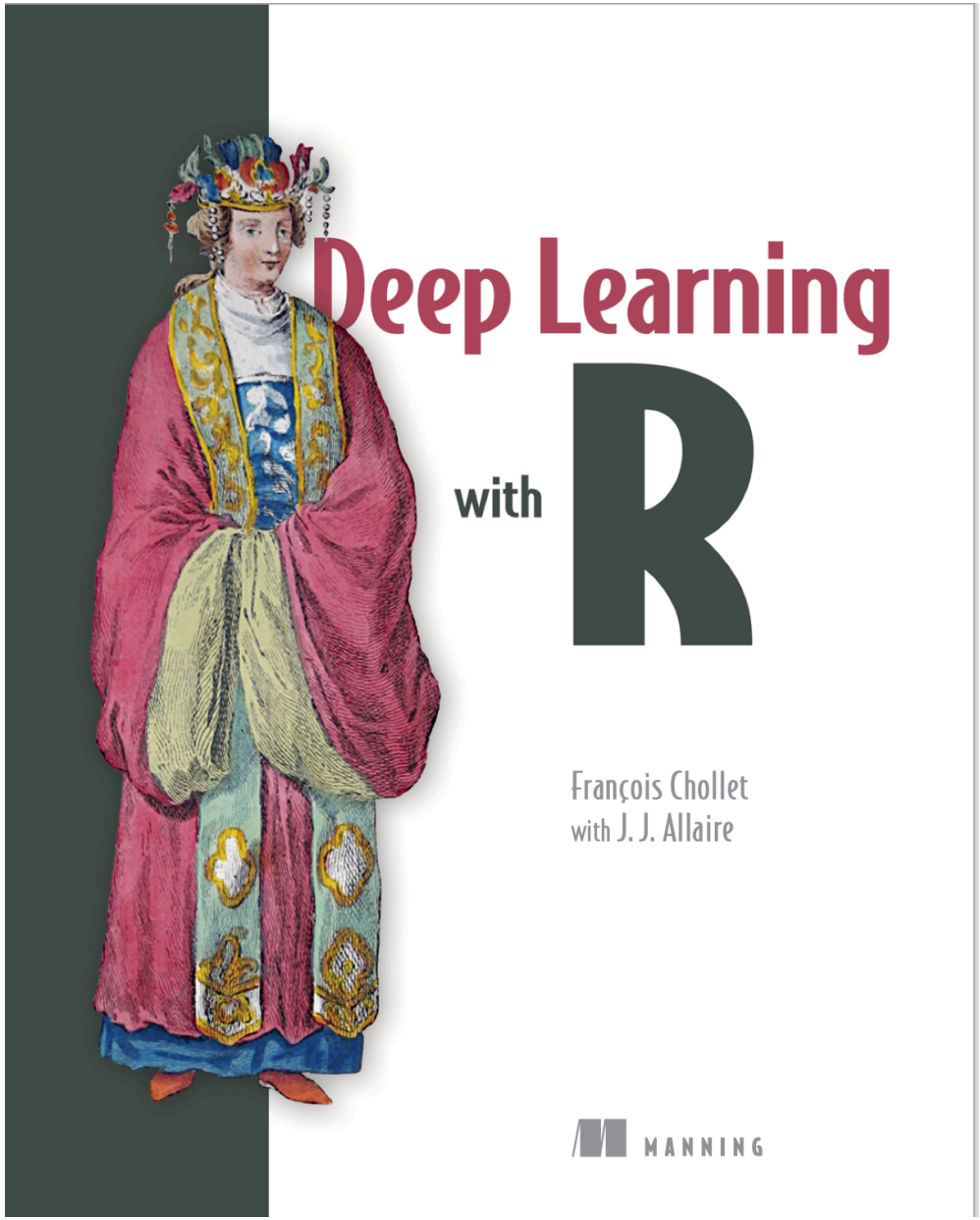


Highlights from

“Deep Learning with R”

by **François Chollet** and J.J. Allaire

FRANÇOIS CHOLLET works on deep learning at Google in Mountain View, CA. He is the creator of the Keras deep-learning library, as well as a contributor to the TensorFlow machine-learning framework. He also does deep-learning research, with a focus on computer vision and the application of machine learning to formal reasoning. His papers have been published at major conferences in the field, including the Conference on Computer Vision and Pattern Recognition (CVPR), the Conference and Workshop on Neural Information Processing Systems (NIPS), the International Conference on Learning Representations (ICLR), and others.



Deep neural networks

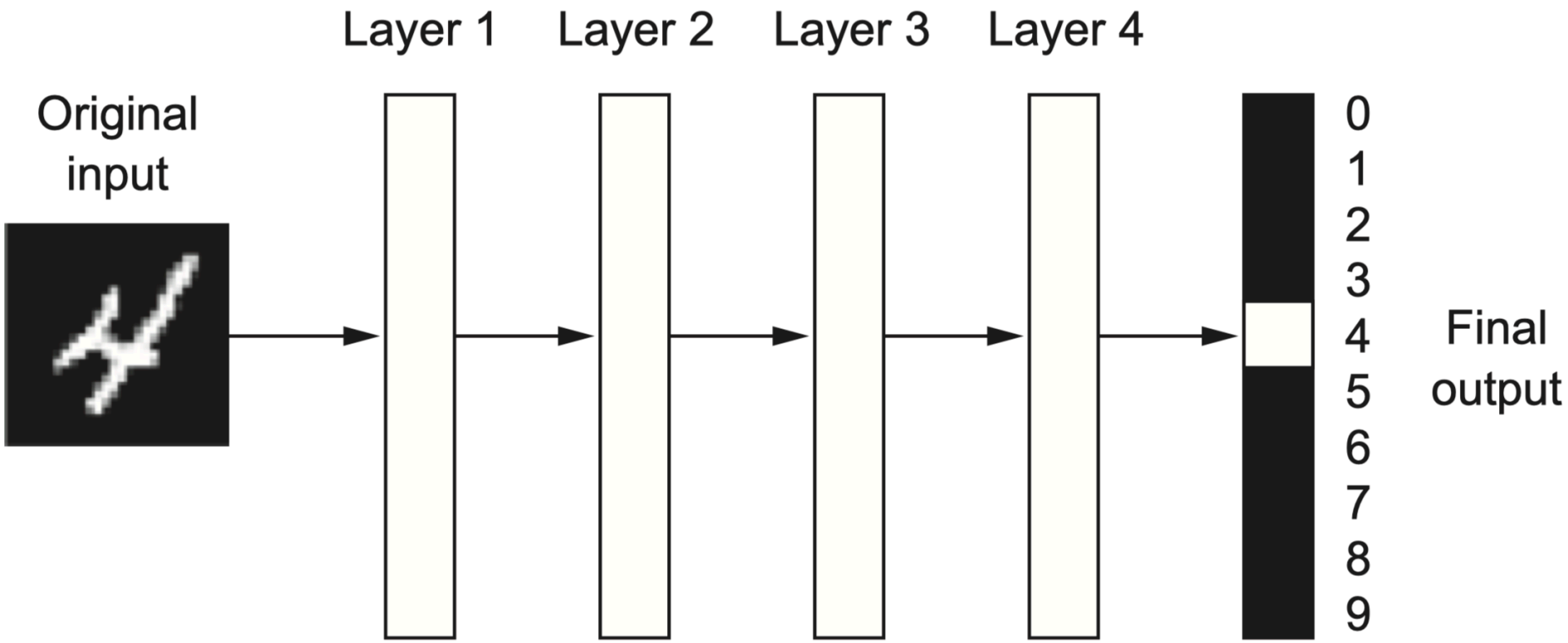
- For our purposes, deep learning is *a mathematical framework for learning representations from data*.
- More specifically: deep learning is a **multistage** way to learn data representations.
- It's a simple idea—but, as it turns out, very simple mechanisms, sufficiently scaled, can end up looking like magic.

Any sufficiently advanced technology is indistinguishable from magic.

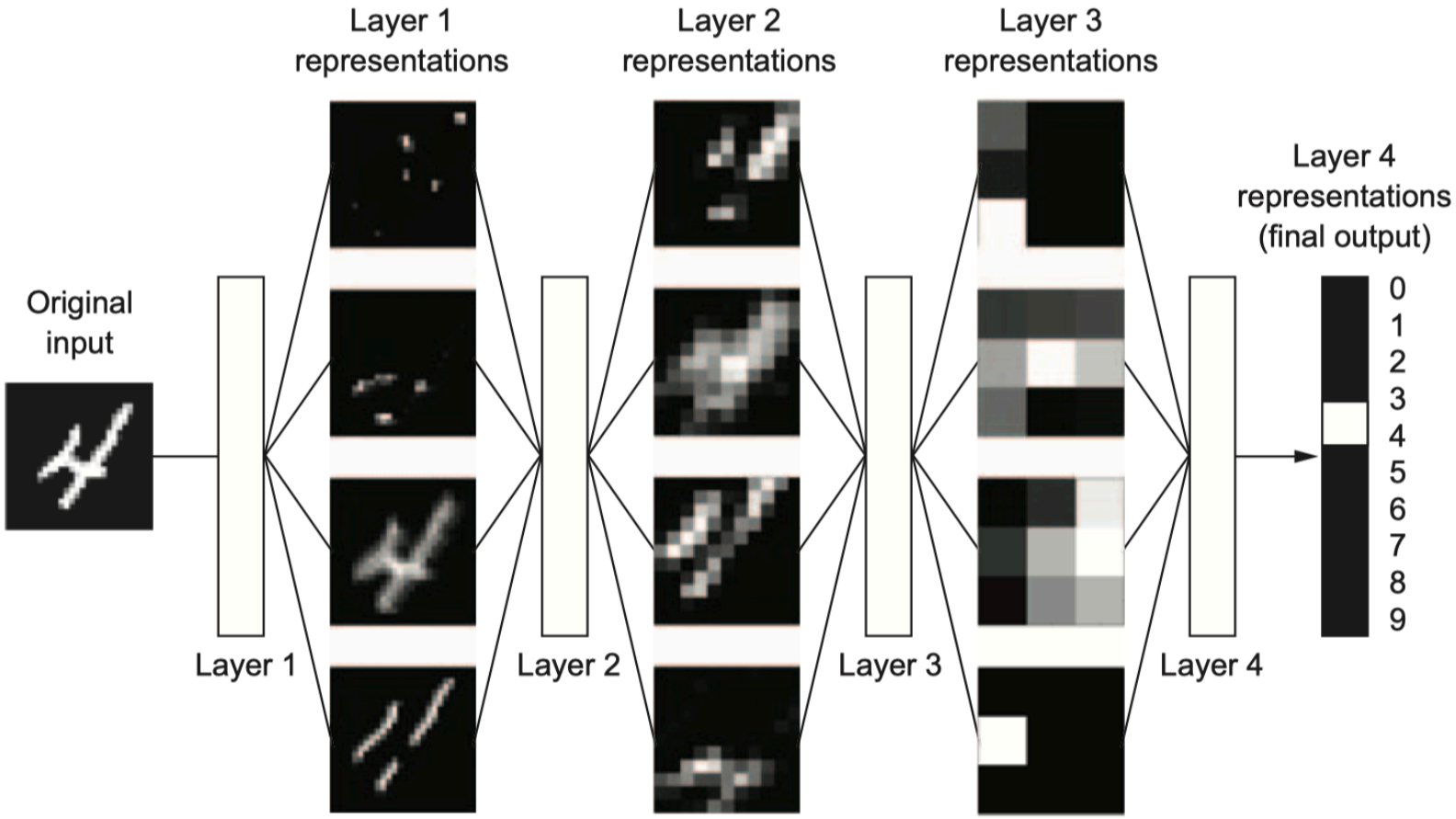
— Arthur C. Clarke —

AZ QUOTES

Example: a deep neural network for digit classification

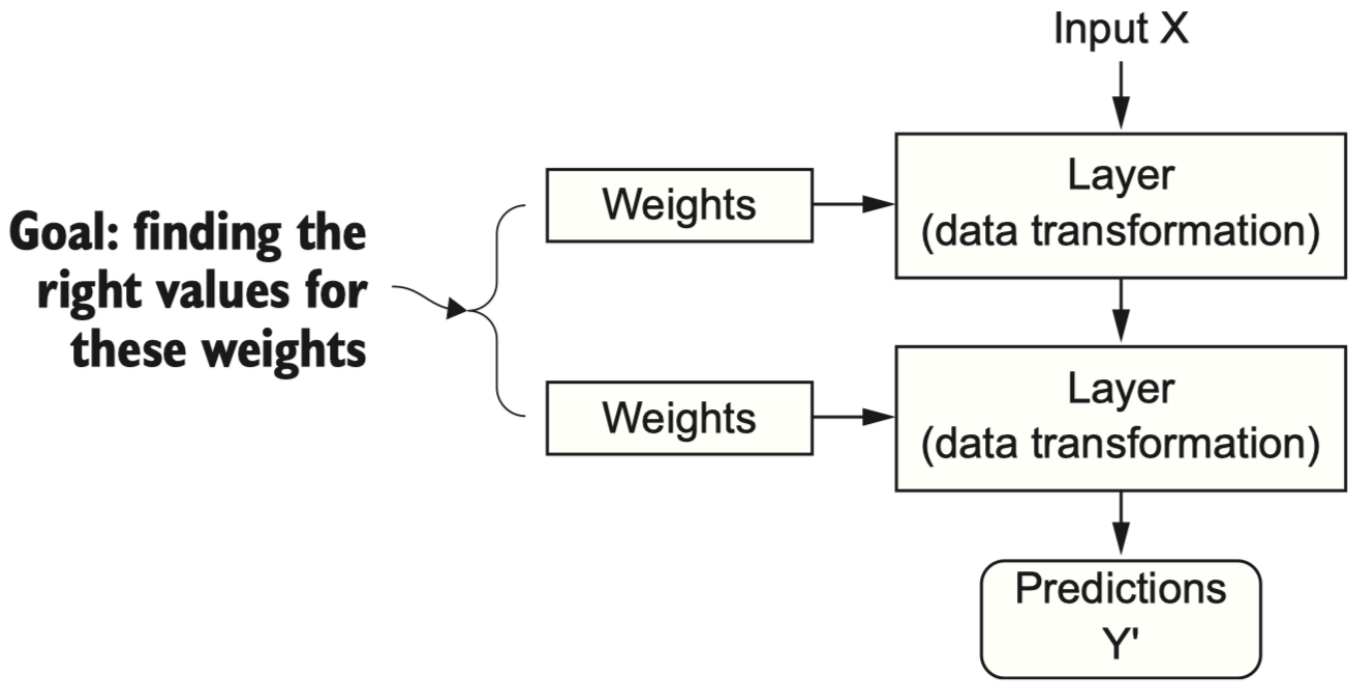


Example: deep representations learned by a digit-classification model



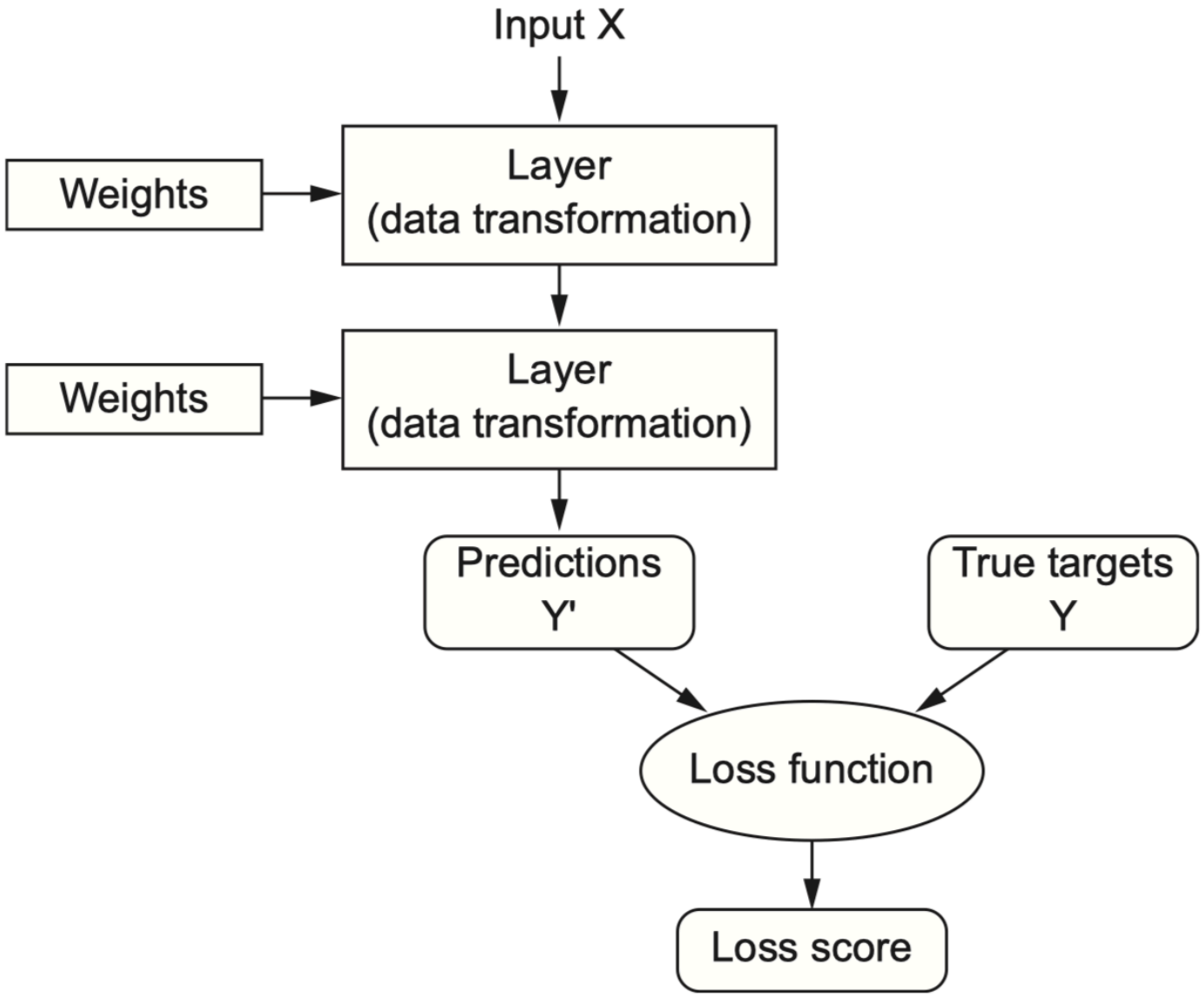
Understanding how deep learning works

A neural network is parameterized by its weights.



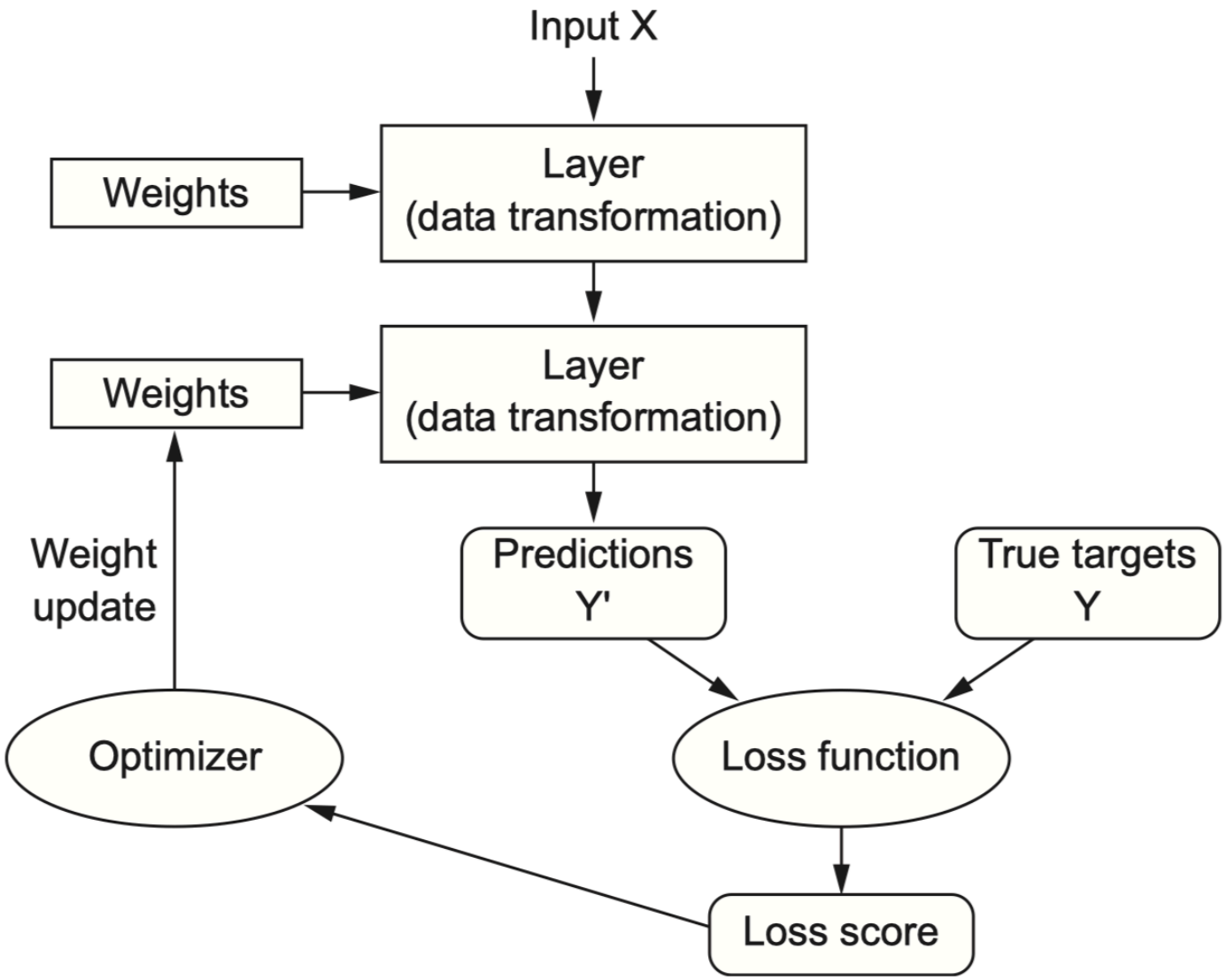
Understanding how deep learning works

A loss function measures the quality of the network's output.

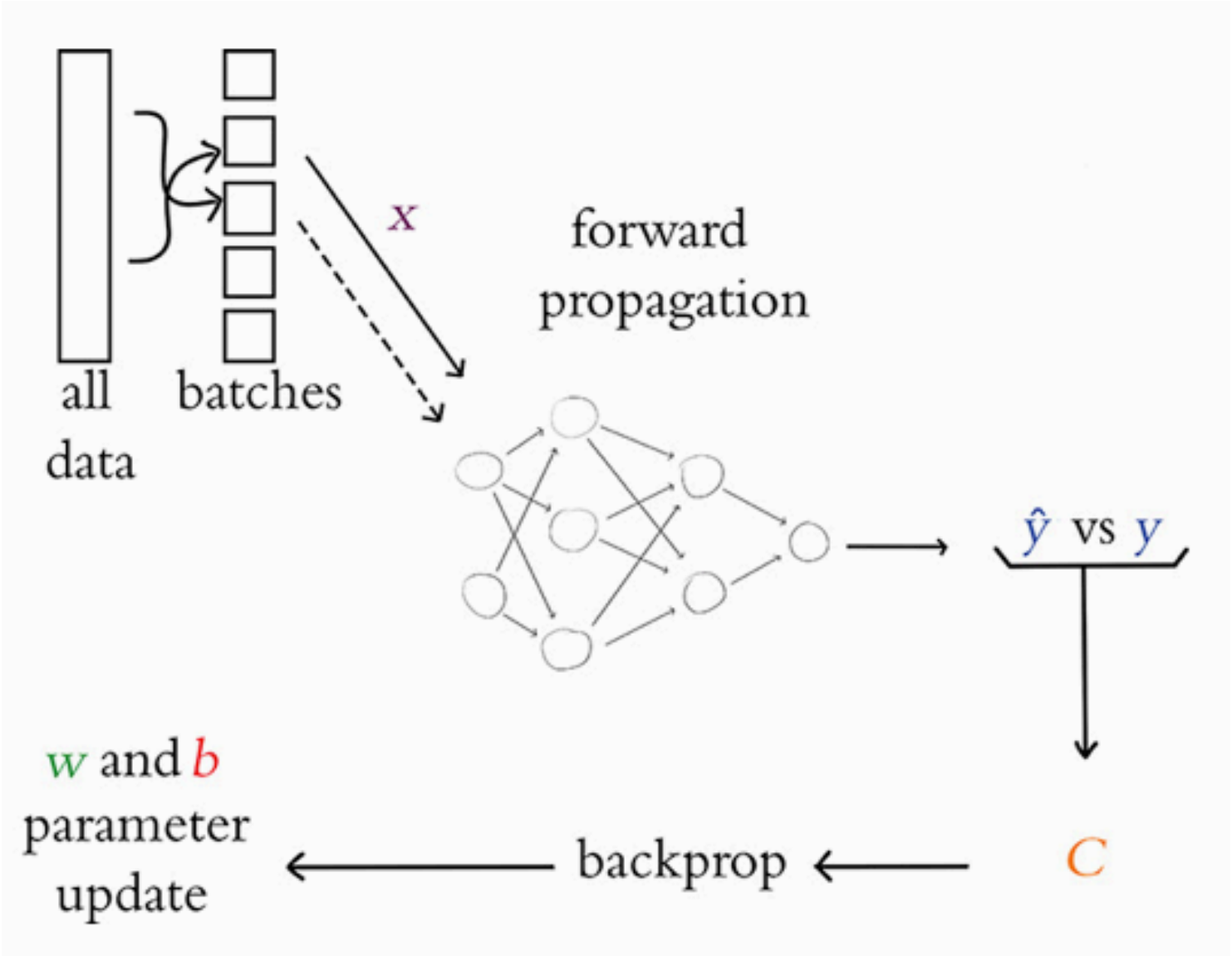


Understanding how deep learning works

The loss score is used as a feedback signal to adjust the weights.



Training a neural network

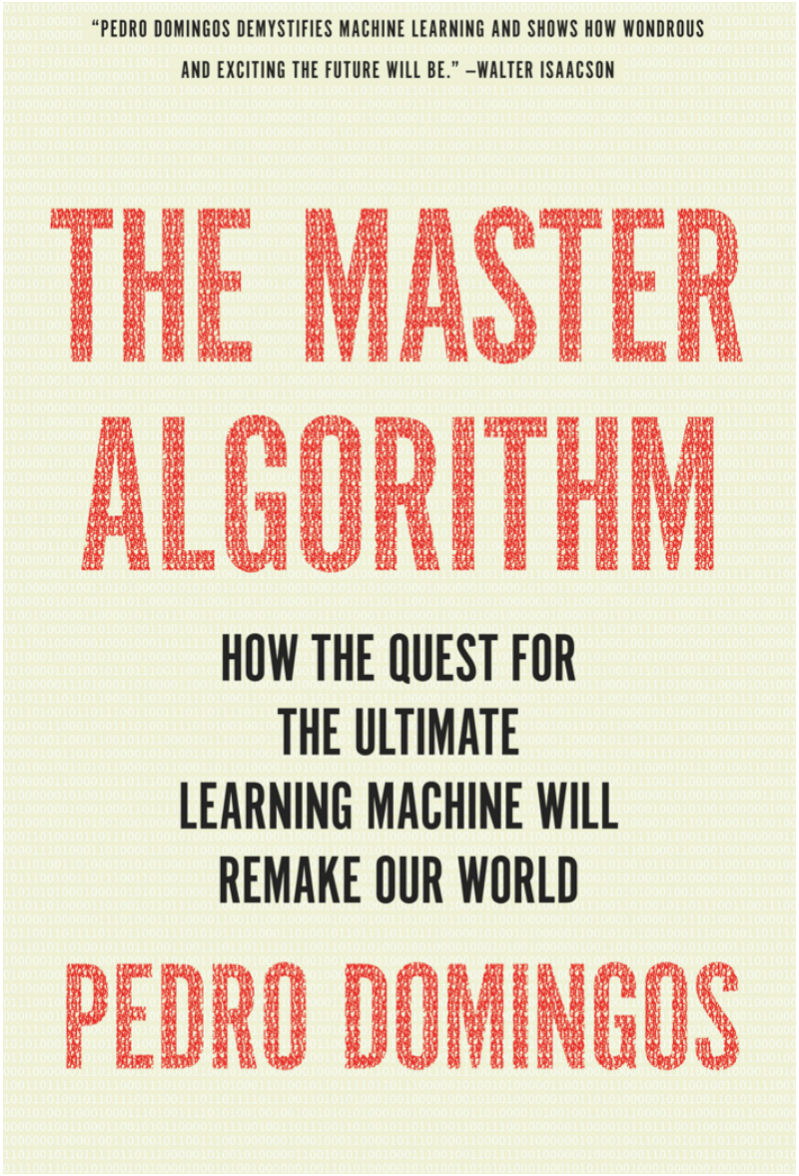


Training a neural network

Round of Training:

1. Sample a mini-batch of x values
2. Forward propagate x through network to estimate y with $\hat{y}$
3. Calculate cost C by comparing y and $\hat{y}$
4. Descend gradient of C to adjust w and b, enabling x to better predict y

Back to the book…



Boltzman machines

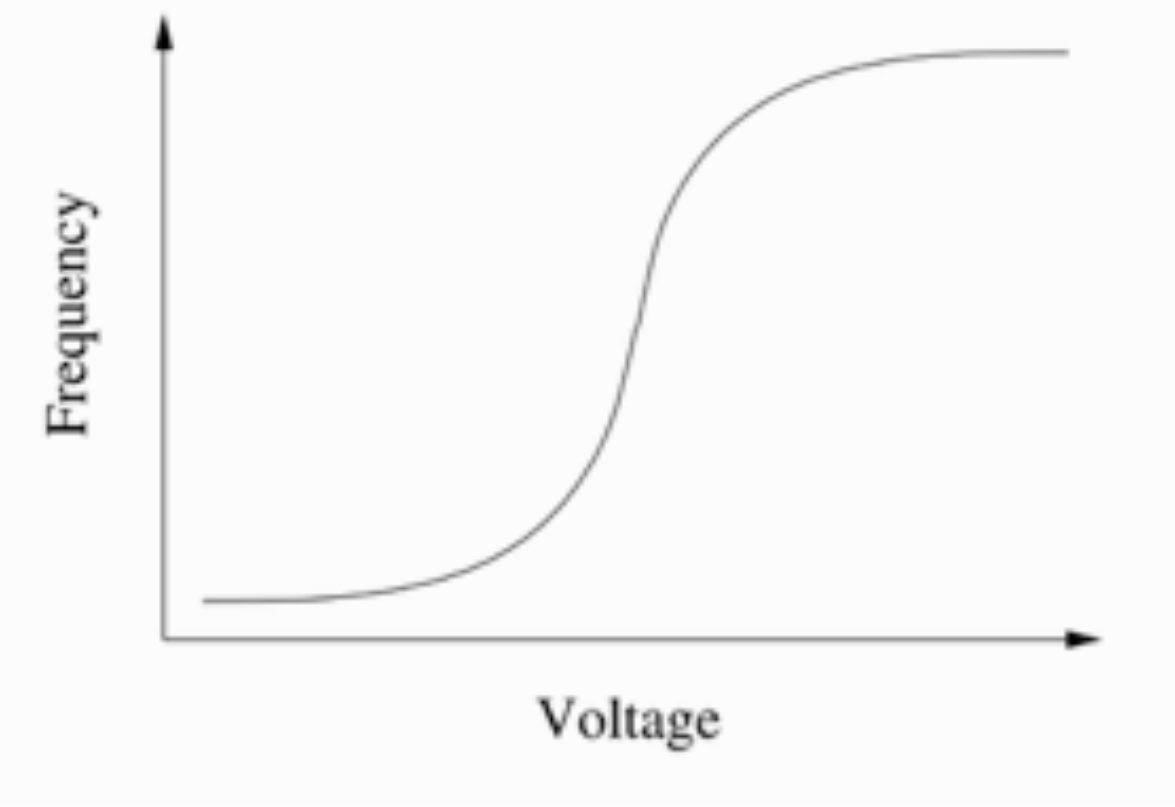
- David Ackley, Geoff Hinton, and Terry Sejnowski (1985) replaced the deterministic neurons in Hopfield networks with probabilistic ones.
- A neural network now had a probability distribution over its states, with higher-energy states being exponentially less likely than lower-energy ones.
- In fact, the probability of finding the network in a particular state was given by the well-known Boltzmann distribution from thermodynamics, so they called their network a **Boltzmann machine**.
- Boltzmann machines could solve the credit-assignment problem in principle, but in practice learning was very slow and painful, making this approach impractical for most applications.

Activation functions

Rather than a logic gate, a neuron is more like a voltage-to-frequency converter.

This curve, which looks like an elongated S, is variously known as the logistic, sigmoid, or S curve.

Domingos calls it “the most important curve in the world.”



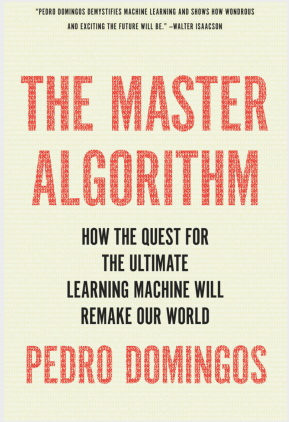
How do neural networks learn?

- Whenever the learner's "retina" sees a new image, that signal propagates forward through the network until it produces an output.
- Comparing this output with the desired one yields an error signal, which then propagates back through the layers until it reaches the retina.
- Based on this returning signal and on the inputs it had received during the forward pass, each neuron adjusts its weights.
- As the network sees more and more images (of two labeled classes), the weights gradually converge to values that let it discriminate between the two.

How do neural networks learn?

- Backpropagation, as this algorithm is known, is phenomenally more powerful than the perceptron algorithm.
- A single neuron could only learn straight lines.
- Given enough hidden neurons, a multilayer perceptron, as it's called, can represent arbitrarily convoluted frontiers.
- This makes backpropagation—or simply backprop—the connectionists' master algorithm.

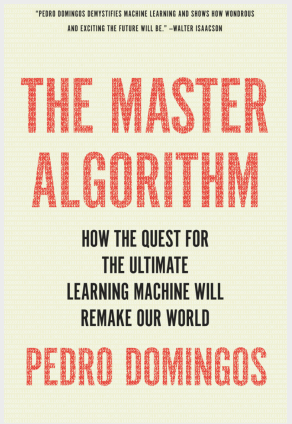
Is backprop the Master Algorithm?



- Some connectionists have been overheard claiming that backprop is the Master Algorithm and we just need to scale it up.

- But symbolists pour scorn on this notion.
- They point to a long list of things that humans can do but neural networks can’t.

Is backprop the Master Algorithm?



- Limitations of DL solutions:
  - Commonsense reasoning.
    - *Did Mary eat a shoe for lunch?*
    - *No, because Mary is a person, people only eat edible things, and shoes are not edible.*
    - Symbolic systems have no trouble with this—they just chain the relevant rules—but multilayer perceptrons can't do it; once they're done learning, they just compute the same fixed function over and over again.
  - Neural networks are not compositional, and compositionality is a big part of human cognition.
  - Humans—and symbolic models like sets of rules and decision trees—can explain their reasoning, while neural networks can't.

Where to go next?

- Learn the basics well!
  - Michael Nielsen's book: http://neuralnetworksanddeeplearning.com/
  - 3blue1brown YouTube channel: https://www.3blue1brown.com/
- Play!
  - Tensorflow playground: https://playground.tensorflow.org/
  - Many others!
- Search for deeper answers (research papers, etc.)