Embedded Machine Learning for Edge Computing - Session 01

An intuitive introduction to Machine Learning

Amaya Dharmasiri

Oct/2022

Who am I

Amaya Dharmasiri

- BSc. Engineering (Electronics and Telecommunications)

  University of Moratuwa

  Sri Lanka

- Research Assistant

  Department of Computer Vision

  Mohammad Bin Zayed University of Artificial Intelligence

  United Arab Emirates

Google Scholar

What is Learning

‘The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something.’

Merriam Webster dictionary

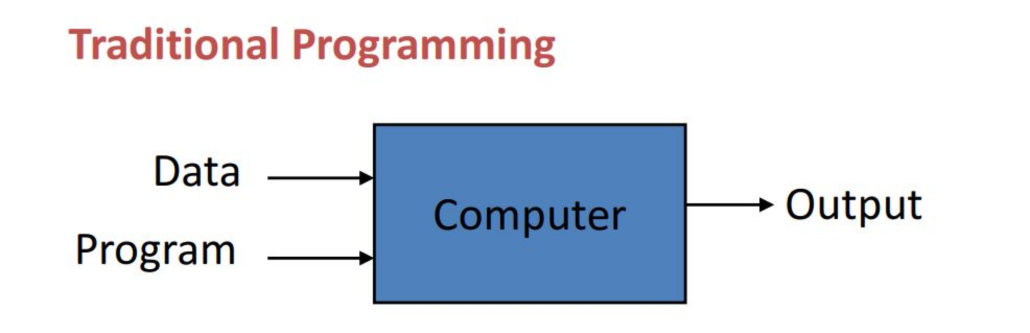
‘A computer program is said to learn from ***experience E*** with respect to some class of ***tasks T*** and ***performance measure P***, if its performance at tasks in T, as measured by P, improves with experience E.’

Tom Mitchell

Traditional Programming vs Machine Learning

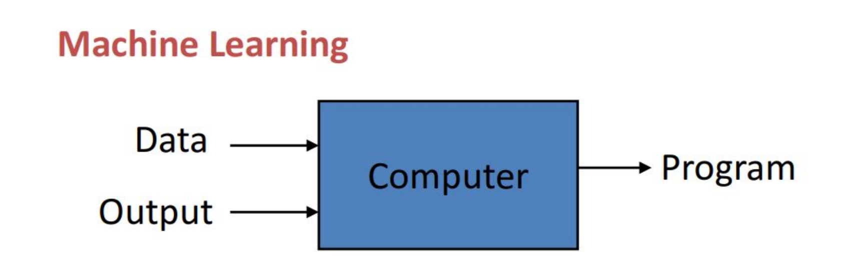
Adding 2 numbers - Traditional Programming

Google Colab notebook



Adding 2 numbers - Machine Learning

- Data - Dataset (Data, Label)
- Learning - An inductive bias, A learning algorithm
- Evaluation



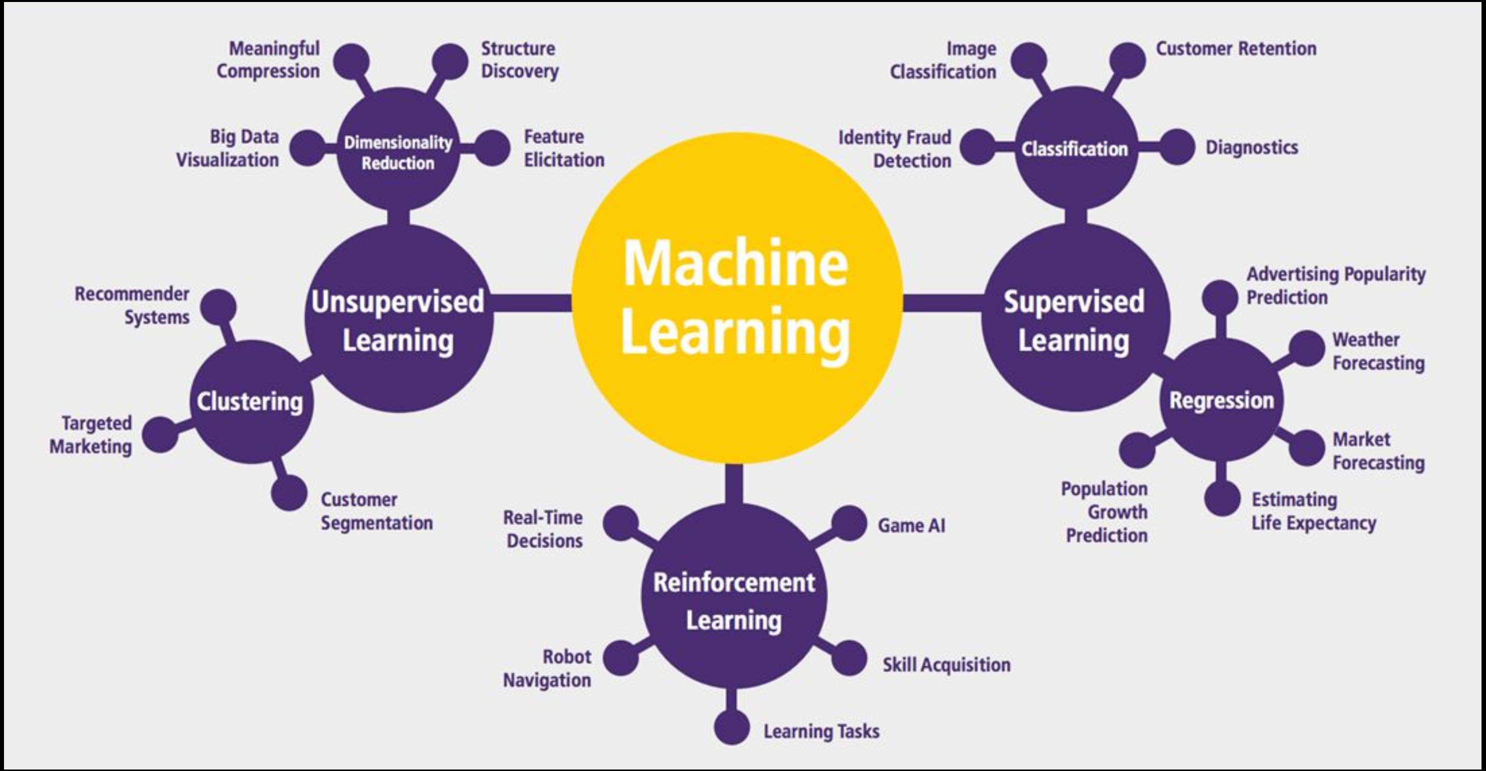
The concept of Learning in a ML system

- Learning = Improving with experience at some task
  - Improve over task *T*,
  - With respect to performance measure *P,*
  - Based on Experience, *E.*

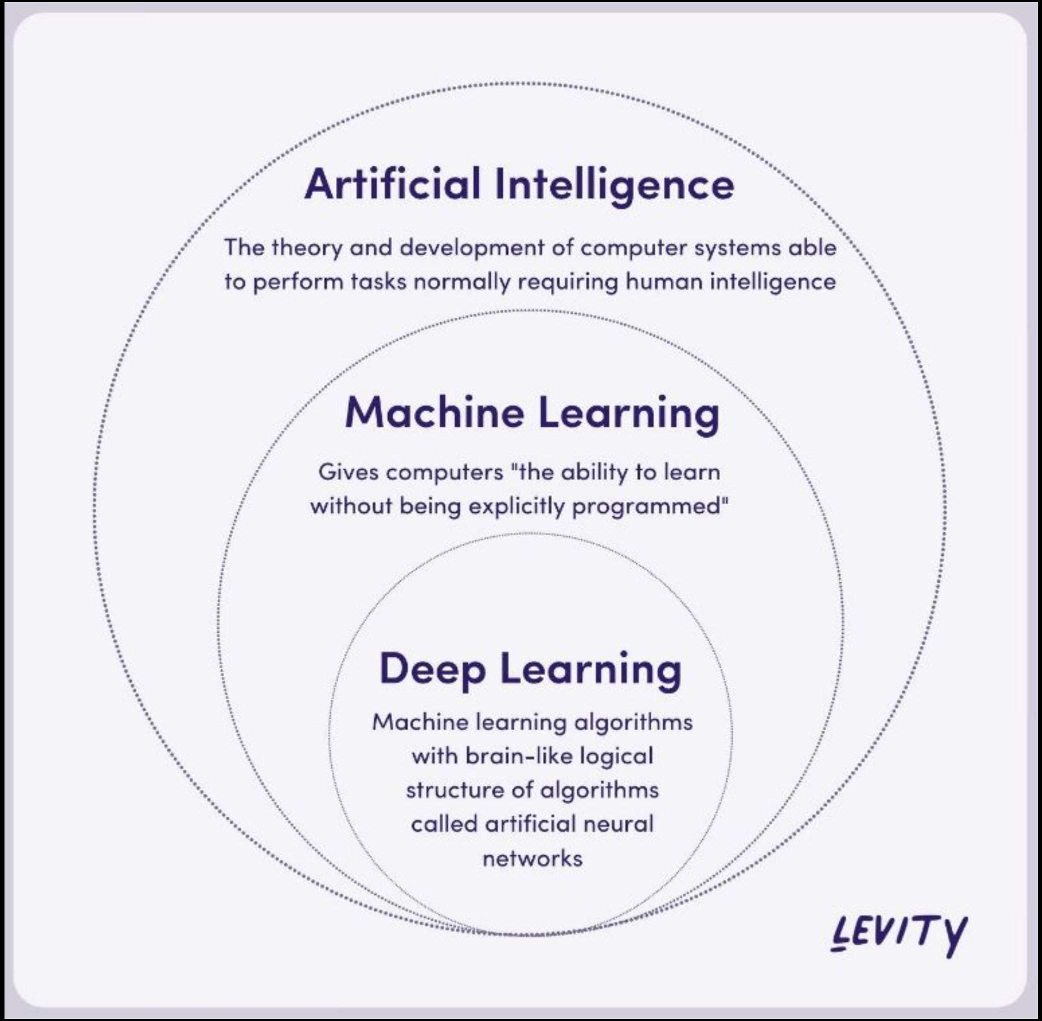
Why Machine Learning

- For many problems, it’s difficult to program the correct behavior by hand
- Hard to code up a solution by hand (e.g. vision, speech)
- System needs to adapt to a changing environments, customize themselves for individual users (e.g. spam detection)
- Mimic humans and replace monotonous tasks that require intelligence (e.g. handwritten digit recognition)
- Want the system to perform better than the human programmers
- Develop systems that are too difficult/expensive to construct manually because they require specific details skills/knowledge for the task (the knowledge engineering bottleneck)

Types of Machine Learning



AI, ML, and DL



Artificial Intelligence

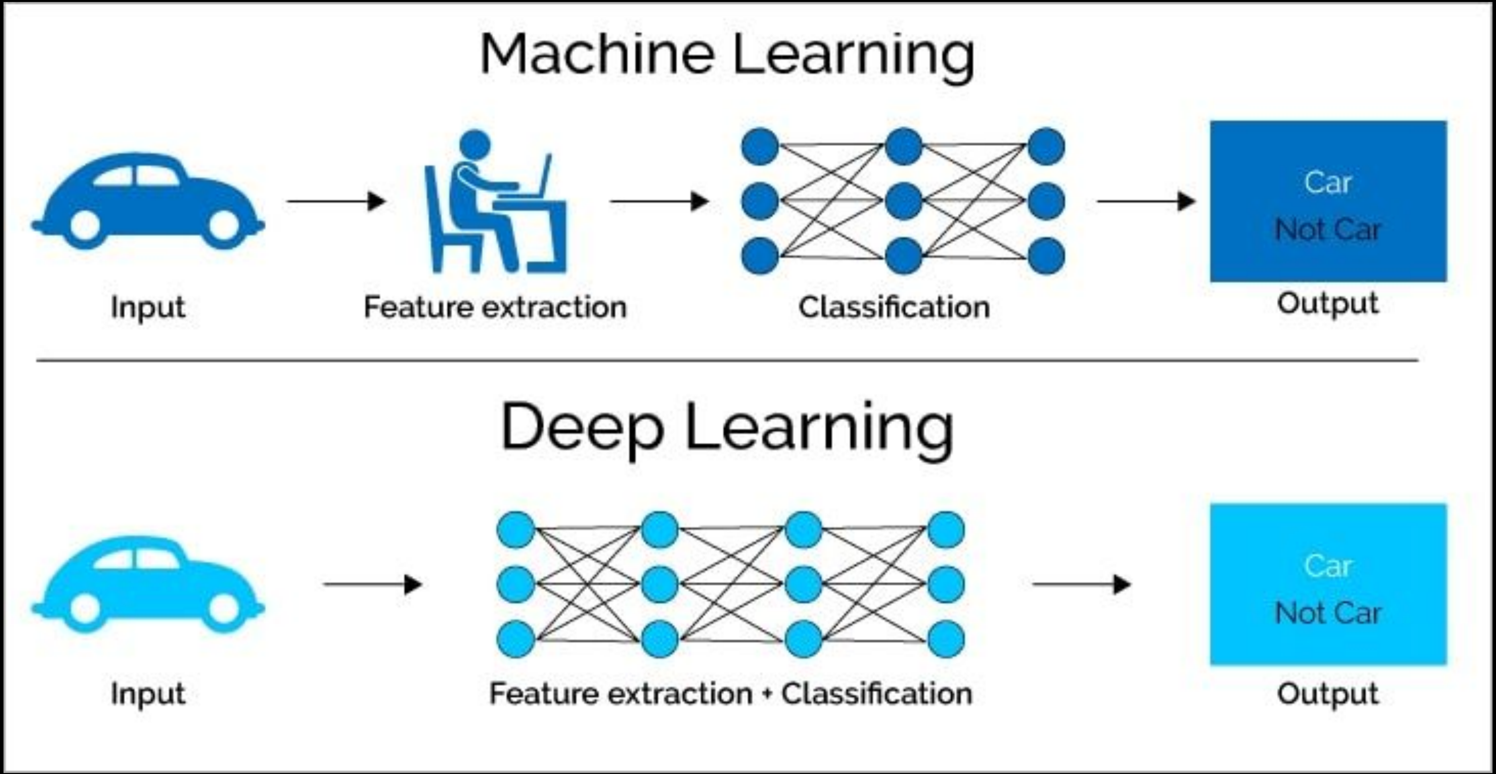
The theory and development of computer systems able to perform tasks normally requiring human intelligence

Machine Learning

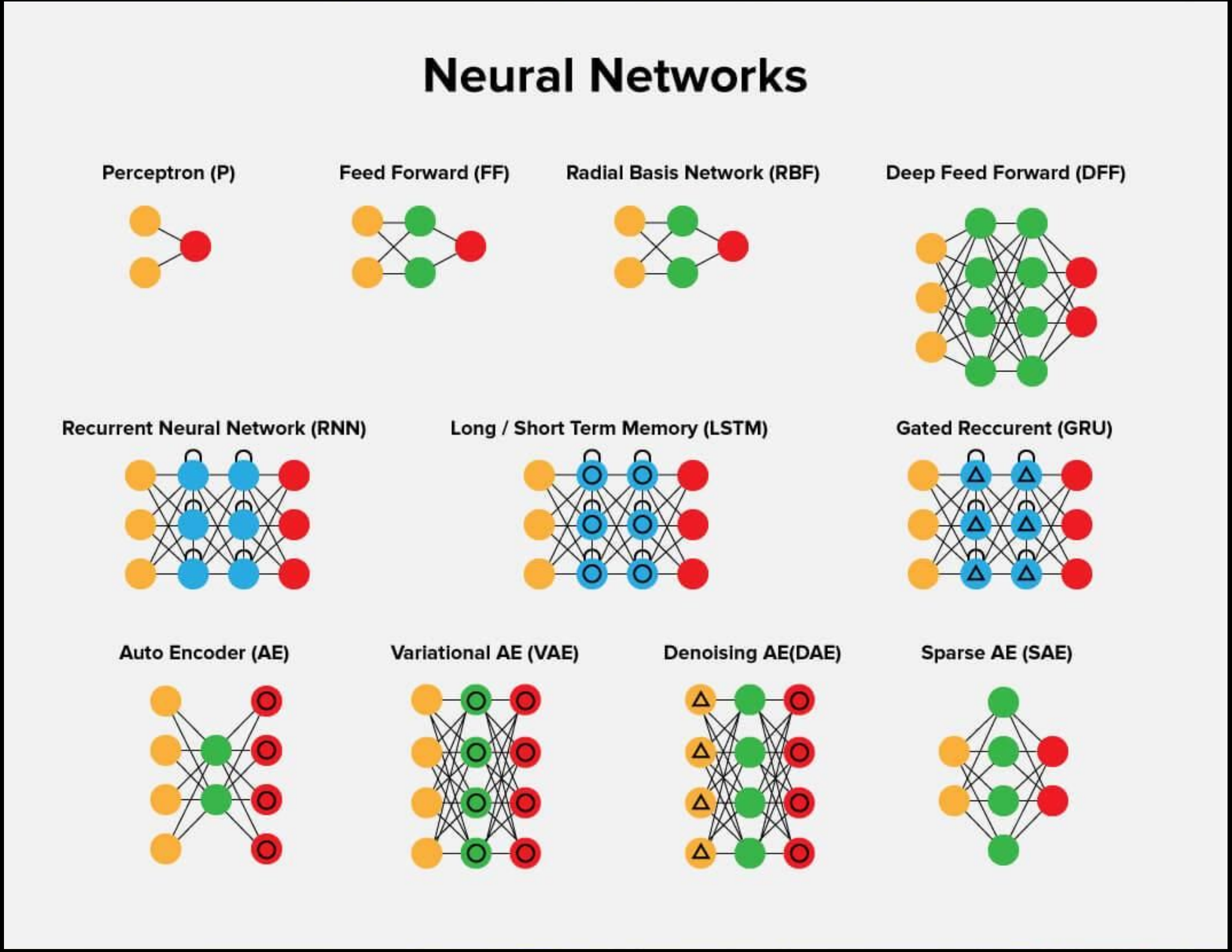
Gives computers "the ability to learn without being explicitly programmed"

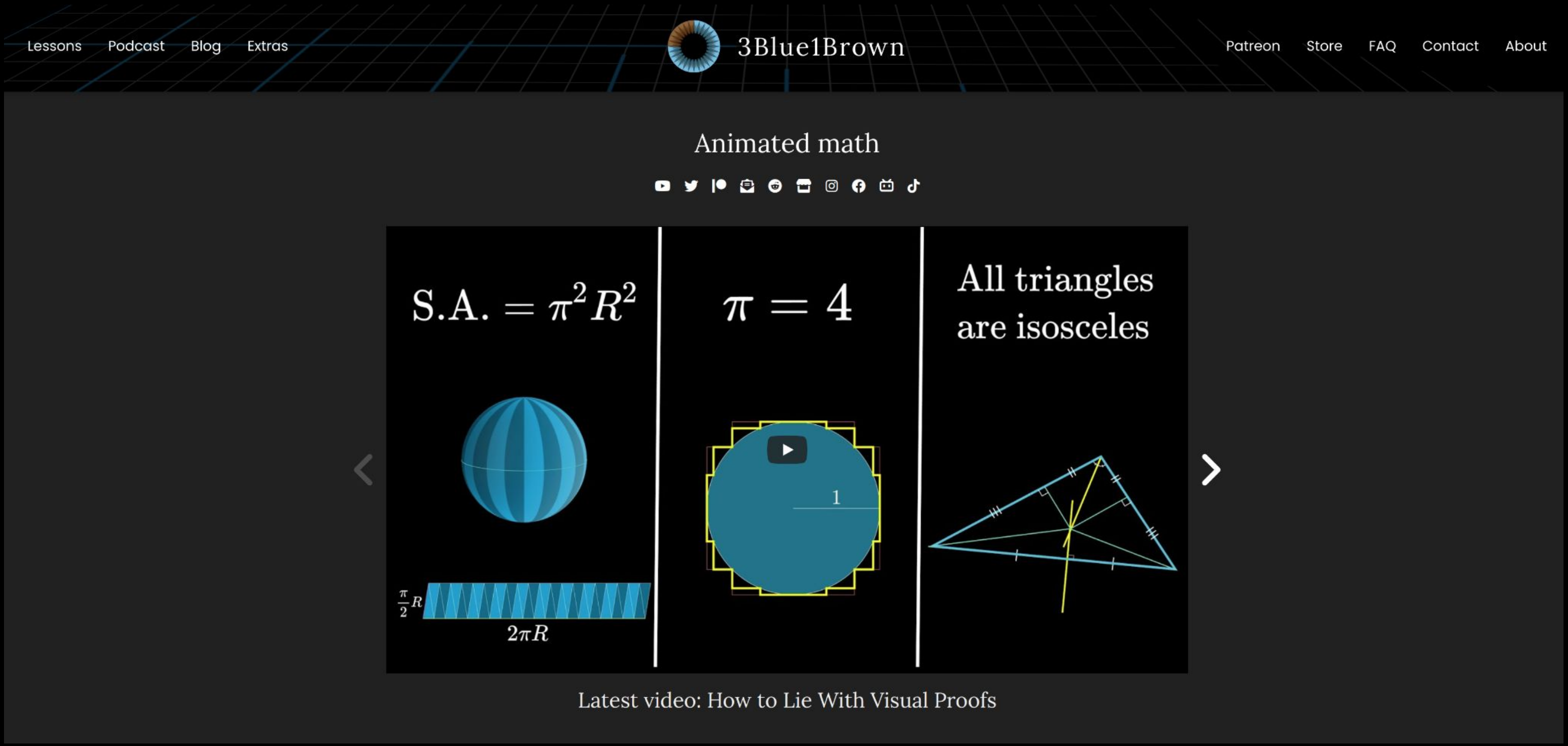
Deep Learning

Machine learning algorithms with brain-like logical structure of algorithms called artificial neural networks


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output

Neural Networks!





Lessons Podcast Blog Extras

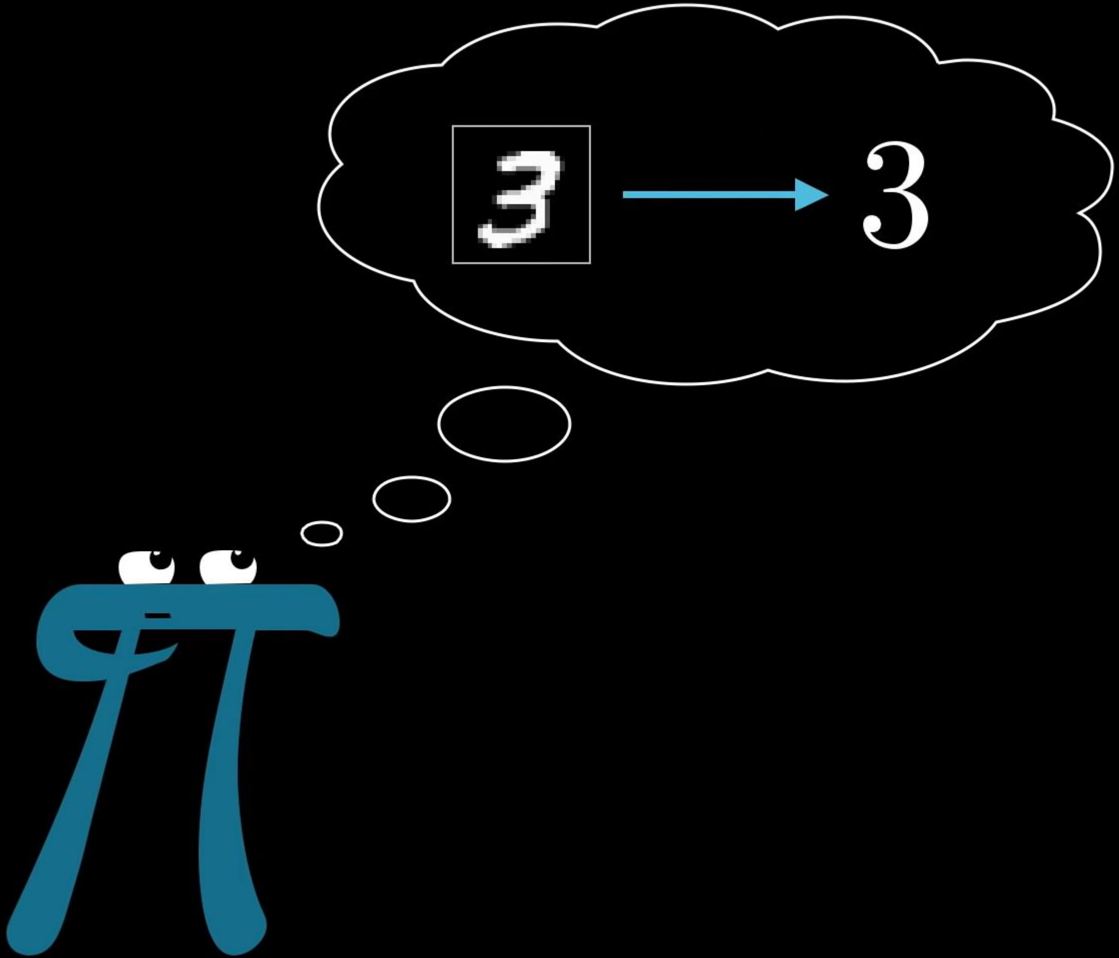
3Blue1Brown

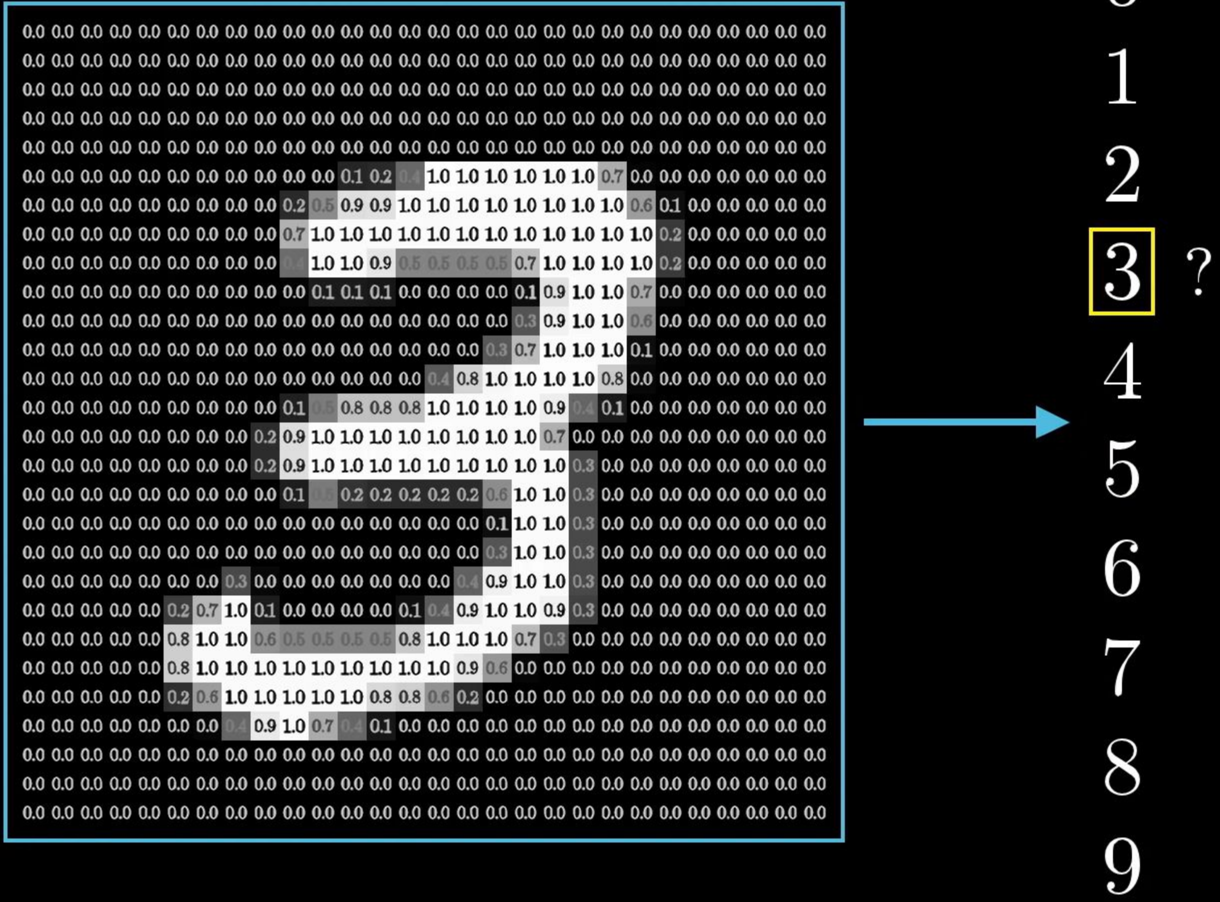
Patreon Store FAQ Contact About

Animated math

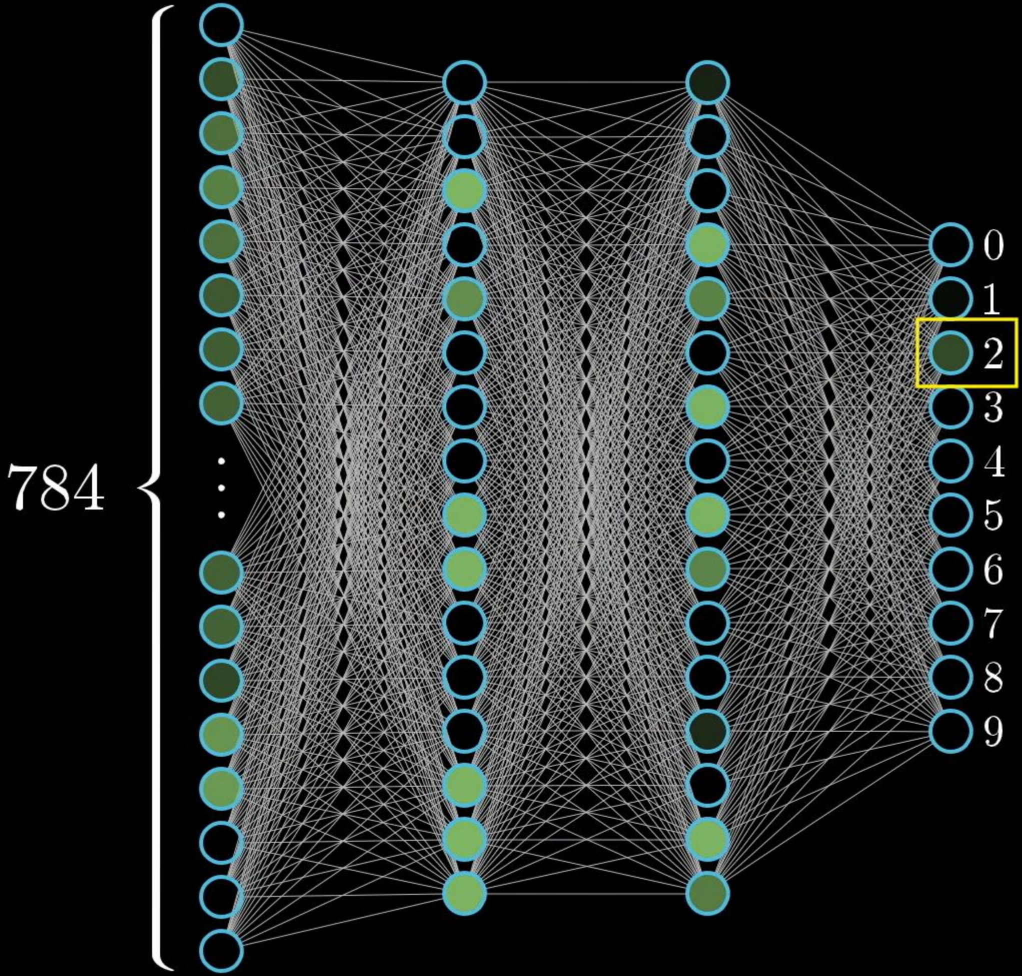
Latest video: How to Lie With Visual Proofs

https://www.3blue1brown.com/


3



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	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	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	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	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	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	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	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	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	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	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.4	1.0	1.0	1.0	1.0	1.0	1.0	0.7	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	0.0	0.0	0.0	0.2	0.5	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.6	0.1	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	0.0	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.2	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	0.0	0.4	1.0	1.0	0.9	0.5	0.5	0.5	0.5	0.7	1.0	1.0	1.0	1.0	0.2	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	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.9	1.0	1.0	0.7	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.9	1.0	1.0	0.6	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.7	1.0	1.0	1.0	0.1	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.8	1.0	1.0	1.0	1.0	0.8	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	0.0	0.0	0.0	0.1	0.5	0.8	0.8	0.8	1.0	1.0	1.0	1.0	0.9	0.4	0.1	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	0.0	0.0	0.2	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	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	0.0	0.0	0.0	0.0	0.2	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.3	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	0.0	0.0	0.0	0.0	0.1	0.5	0.2	0.2	0.2	0.2	0.2	0.6	1.0	1.0	0.3	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	1.0	1.0	0.3	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.0	1.0	0.3	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	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.9	1.0	1.0	0.3	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	0.2	0.7	1.0	0.1	0.0	0.0	0.0	0.0	0.1	0.4	0.9	1.0	1.0	0.9	0.3	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	0.8	1.0	1.0	0.6	0.5	0.5	0.5	0.5	0.8	1.0	1.0	1.0	0.7	0.3	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	0.0	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.6	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	0.0	0.0	0.0	0.2	0.6	1.0	1.0	1.0	1.0	1.0	0.8	0.8	0.6	0.2	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	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.9	1.0	0.7	0.4	0.1	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	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	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	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	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	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	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

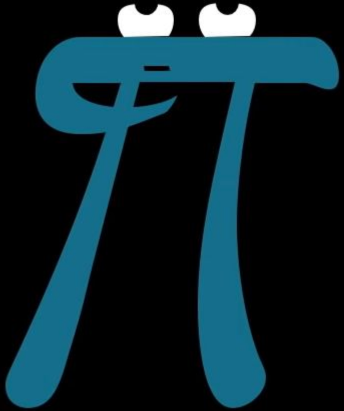


784
0
1
2
3
4
5
6
7
8
9

Neural network

What are the neurons?

How are they connected?



Neuron → Thing that holds a number

28

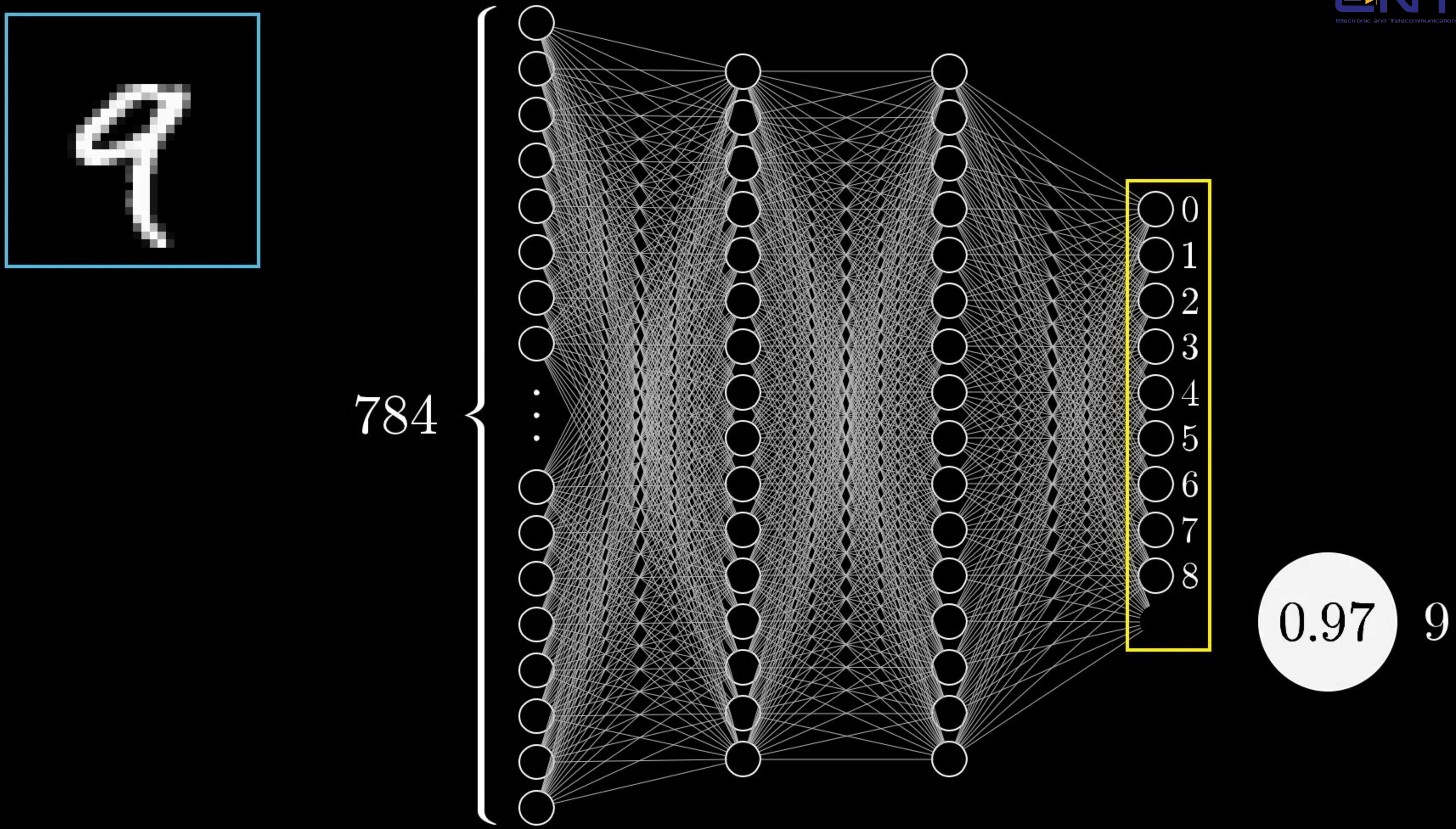
28

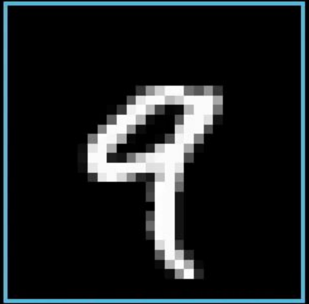
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	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	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	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	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	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	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	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	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	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	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	0.2		0.8	1.0	1.0	0.4	0.3	0.6	0.2	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	0.0	0.0	0.0	0.0	0.0	0.3	0.9	1.0	1.0	0.7	0.8	1.0	1.0	1.0	0.7	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	0.0	0.0	0.0	0.0	0.2	0.9	1.0	0.7	0.3	0.0	0.0	0.7	1.0	1.0	0.5	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	0.0	0.0	0.0	0.4	1.0	1.0	0.7	0.1	0.0	0.0	0.4	1.0	1.0	0.9	0.1	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	0.0	0.0	0.5	1.0	1.0	0.6	0.1	0.0	0.0	0.0	0.8	1.0	1.0	0.3	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	0.0	0.0	0.5	1.0	1.0	0.7	0.0	0.0	0.0	0.0	0.3	1.0	1.0	0.5	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	0.0	0.0	0.1	0.9	1.0	0.7	0.0	0.0	0.0	0.1	0.8	1.0	1.0	0.7	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	0.0	0.0	0.0	0.1	1.0	1.0	0.1	0.1	0.5	0.8	0.9	1.0	1.0	1.0	0.3	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	0.0	0.0	0.0	0.1	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.9	1.0	0.9	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	0.0	0.0	0.0	0.0	0.0	0.2	0.9	1.0	0.8	0.5	0.2	0.1	0.7	1.0	0.6	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	1.0	1.0	0.4	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.0	1.0	0.1	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.1	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	1.0	1.0	0.1	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	1.0	1.0	0.1	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	1.0	1.0	0.1	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	1.0	0.2	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	1.0	0.7	0.2	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.7	1.0	0.7	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.6	1.0	0.2	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	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

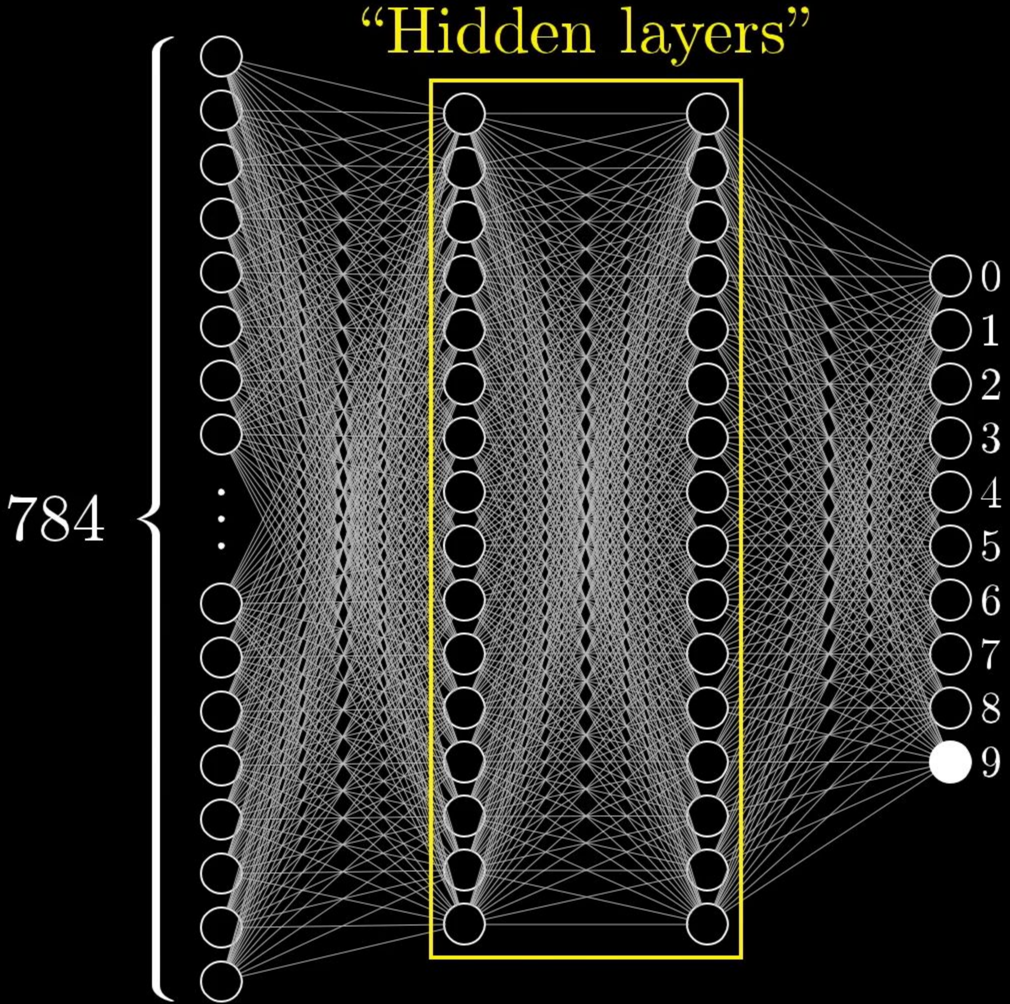
$$28 \times 28 = 784$$

0.58

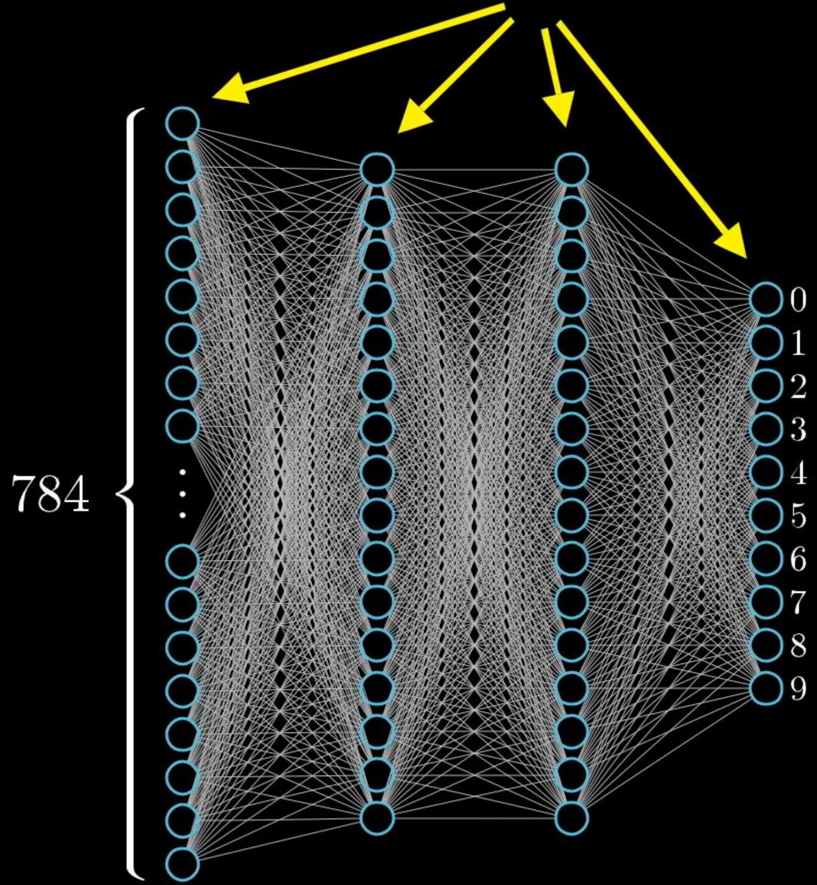
“Activation”

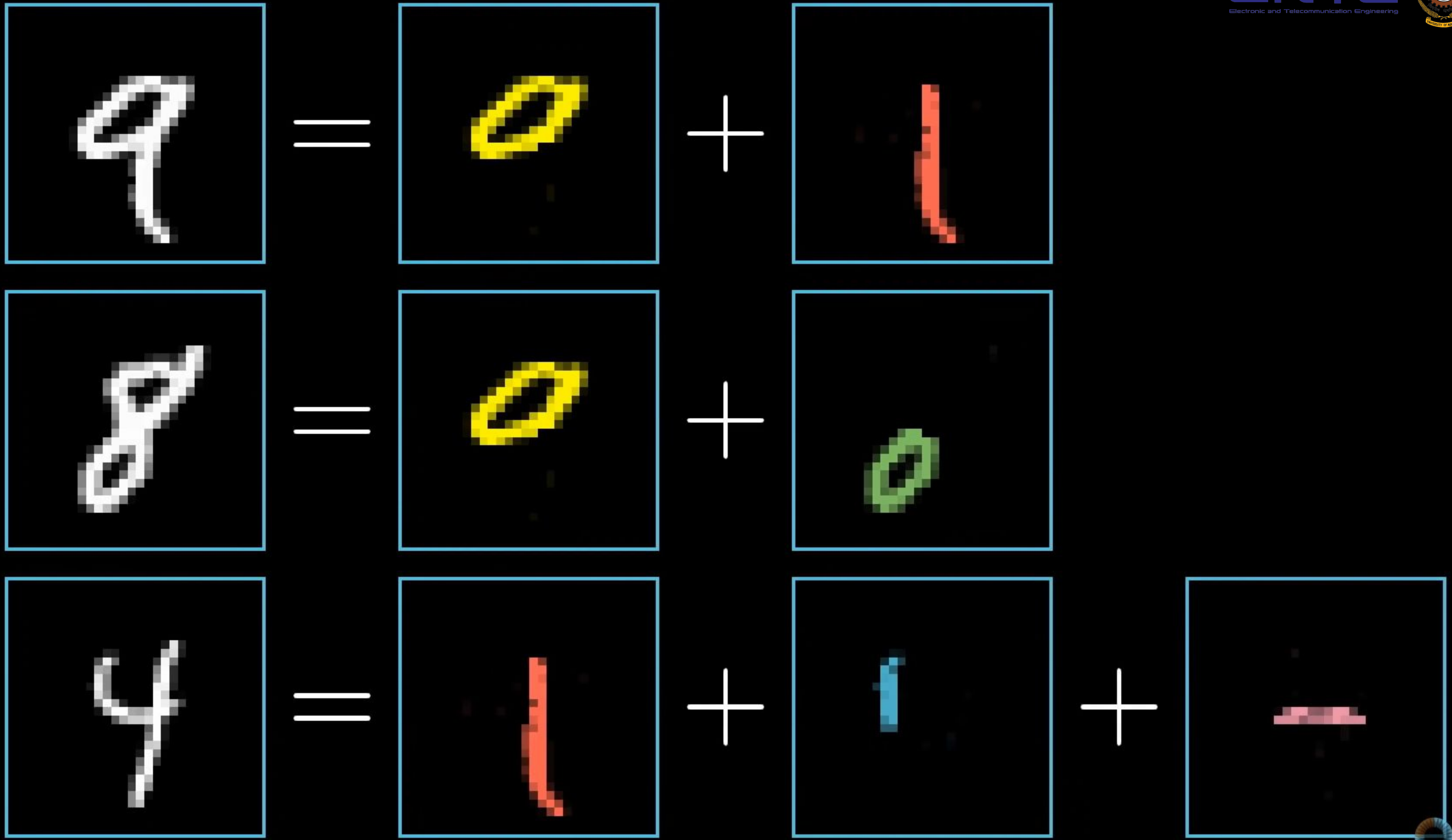

784
0
1
2
3
4
5
6
7
8
0.97
9

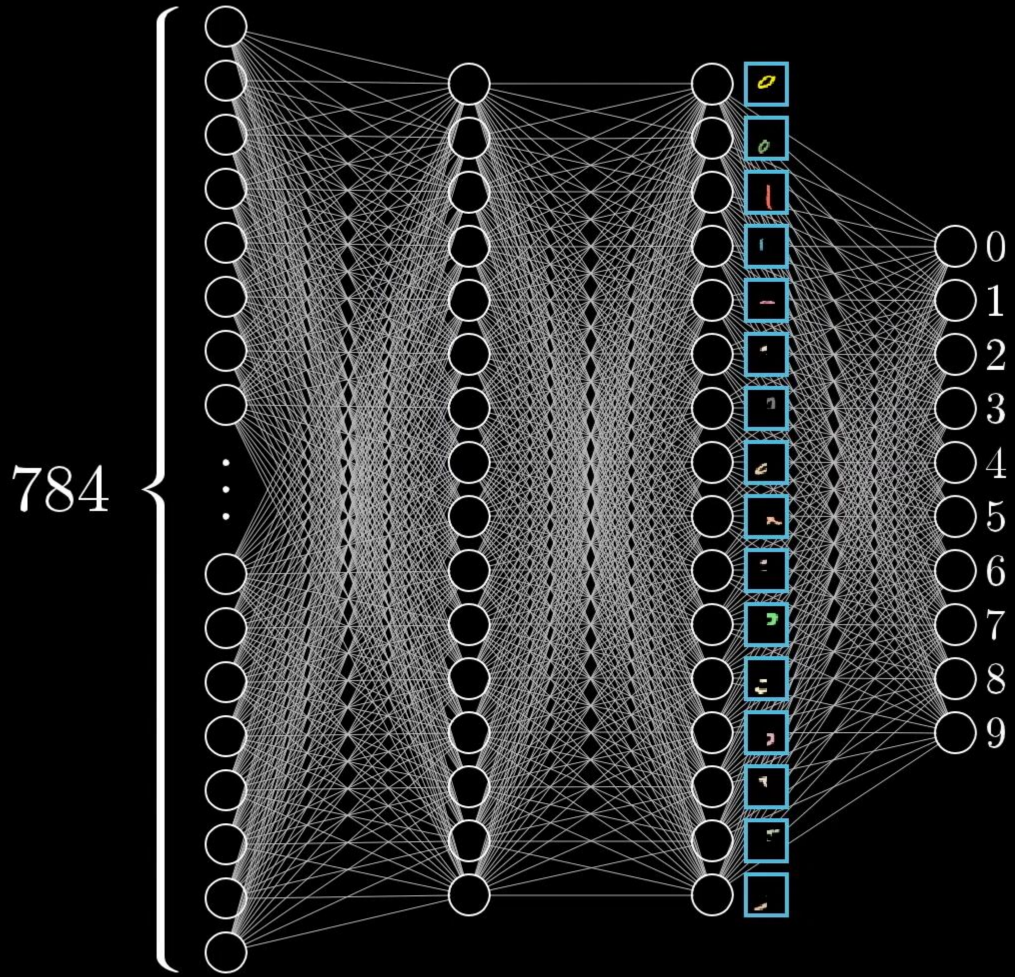


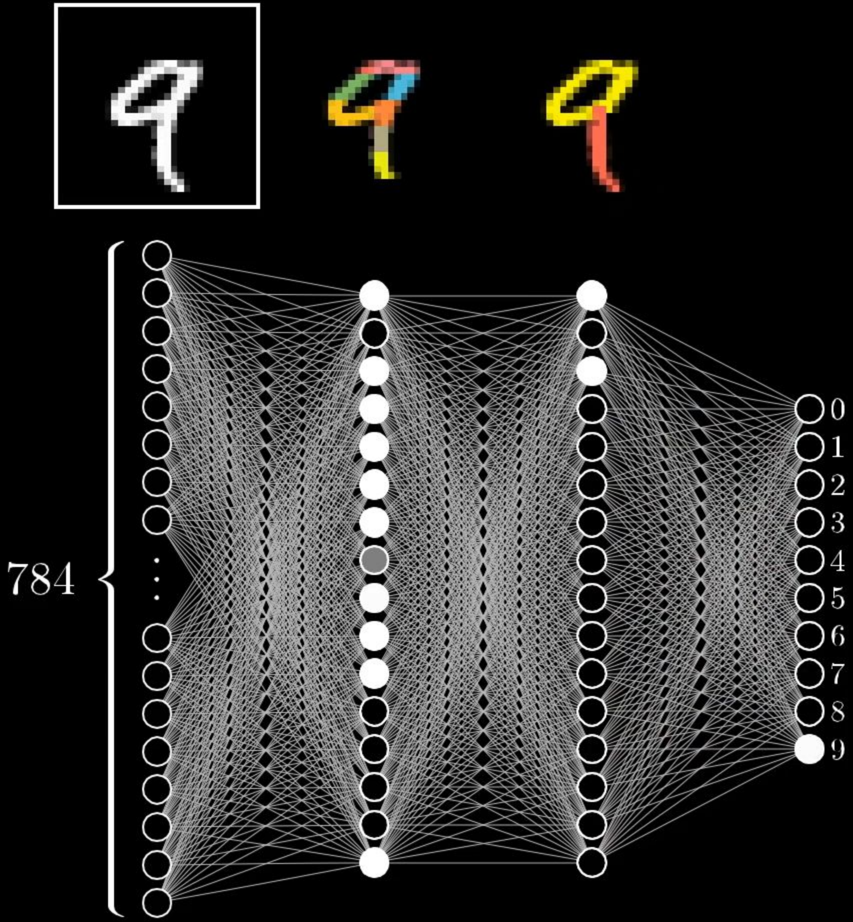

"Hidden layers"
784
0
1
2
3
4
5
6
7
8
9

Why the layers?

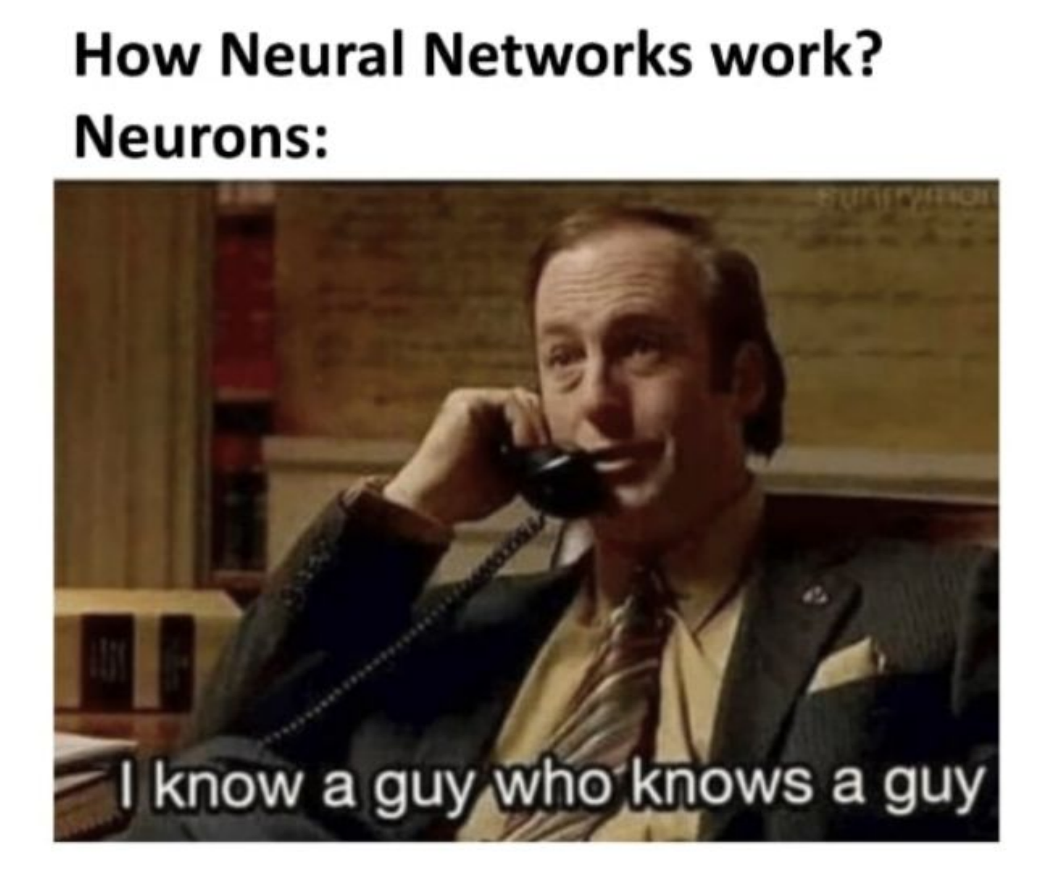





784
0
1
2
3
4
5
6
7
8
9

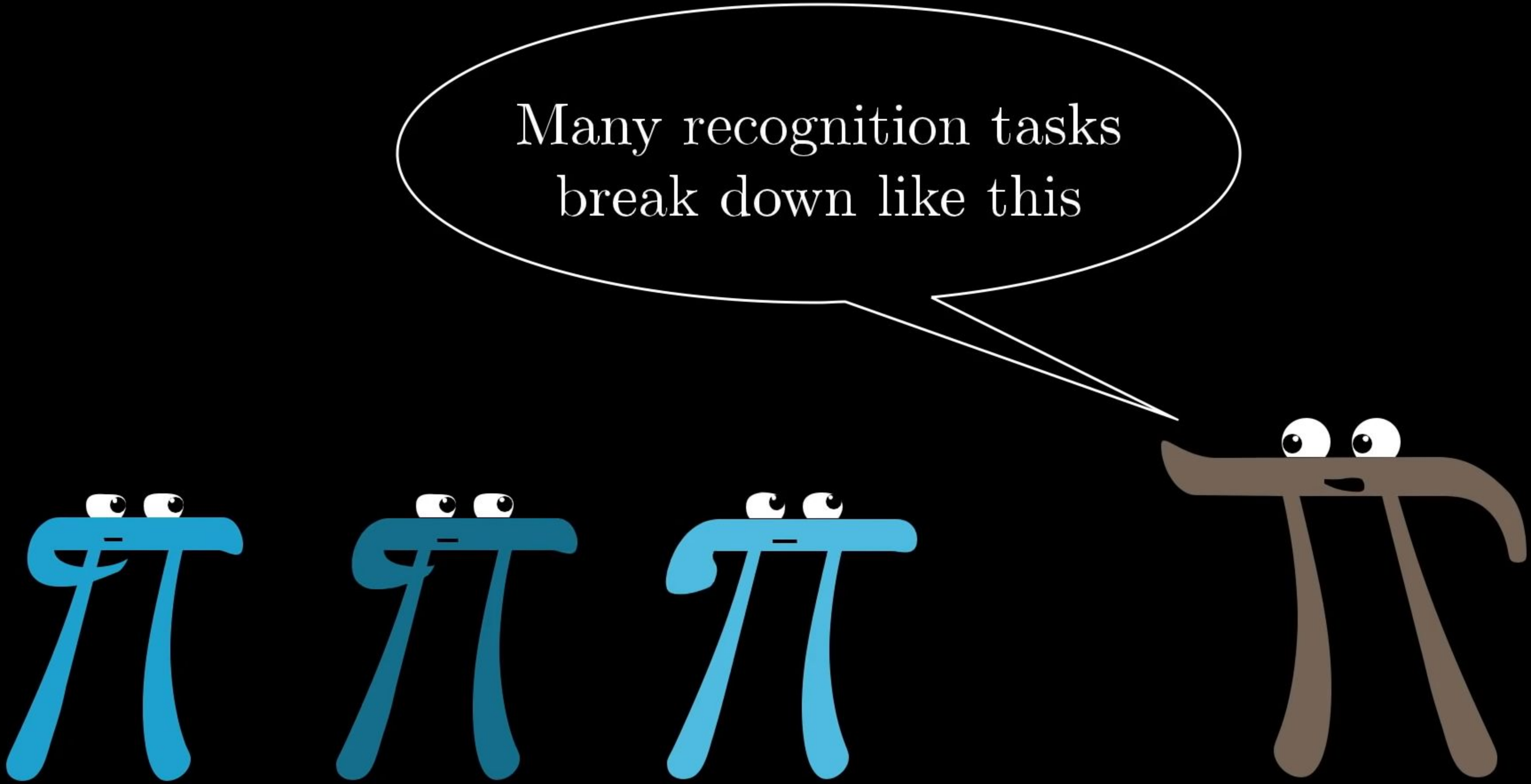

784
0
1
2
3
4
5
6
7
8
9

Neural Networks!

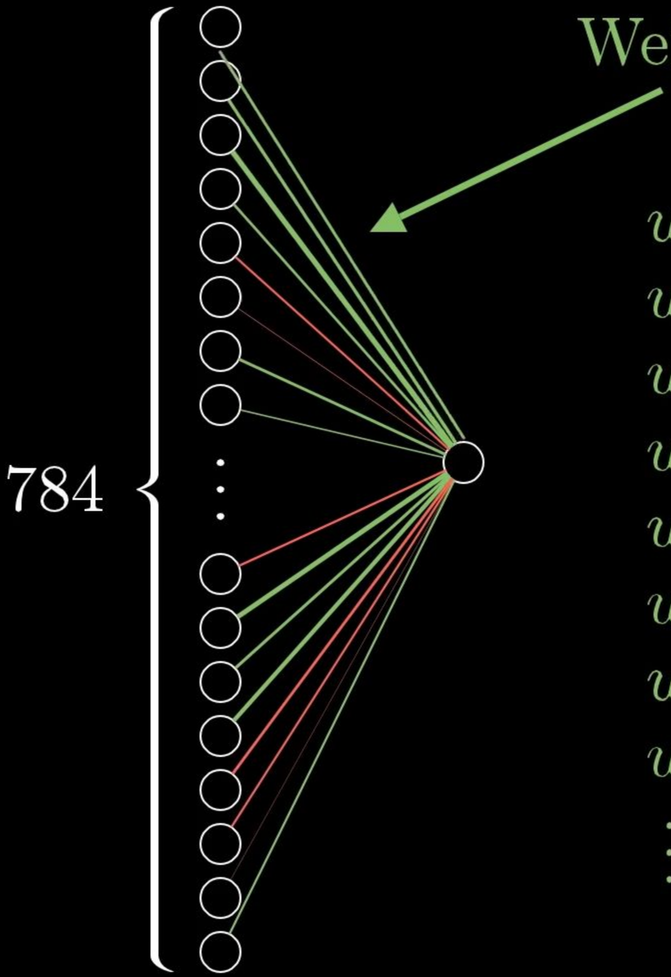


How Neural Networks work?

Neurons:


Many recognition tasks break down like this

Weights



w_1: 2.07

w_2: 2.31

w_3: 3.64

w_4: 1.87

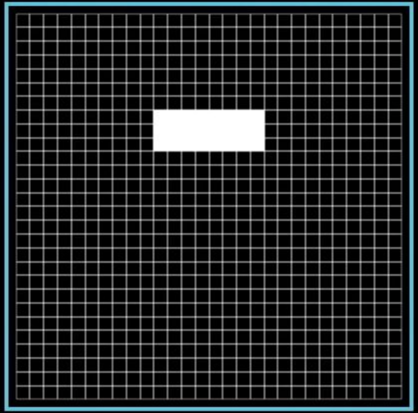
w_5: -1.51

w_6: -0.43

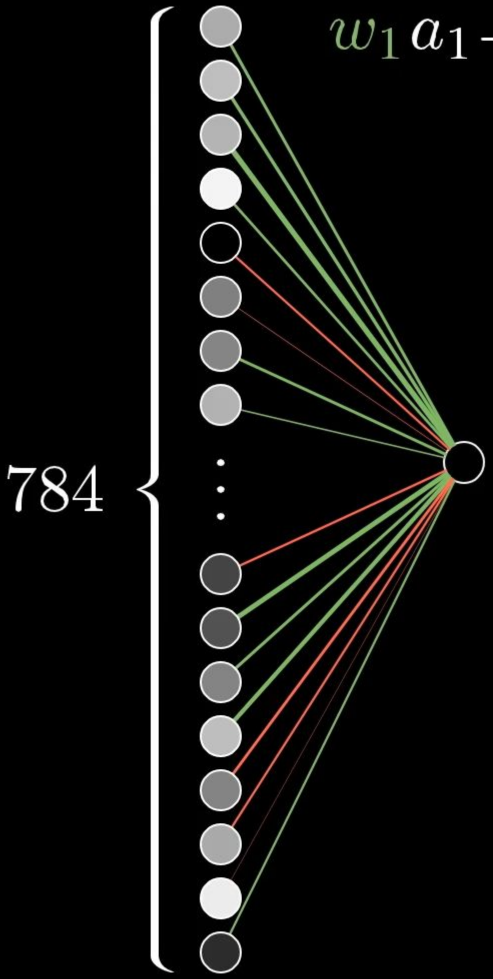
w_7: 2.01

w_8: 1.07

$\vdots$



$$w_1 a_1 + w_2 a_2 + w_3 a_3 + w_4 a_4 + \cdots + w_n a_n$$



w_1: 2.07

w_2: 2.31

w_3: 3.64

w_4: 1.87

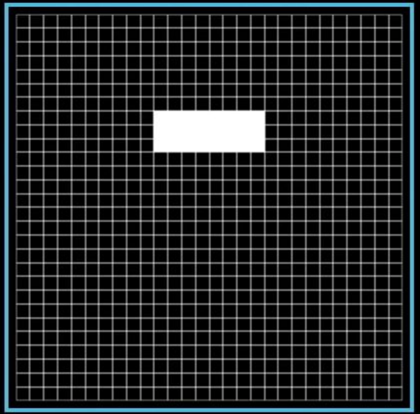
w_5: −1.51

w_6: −0.43

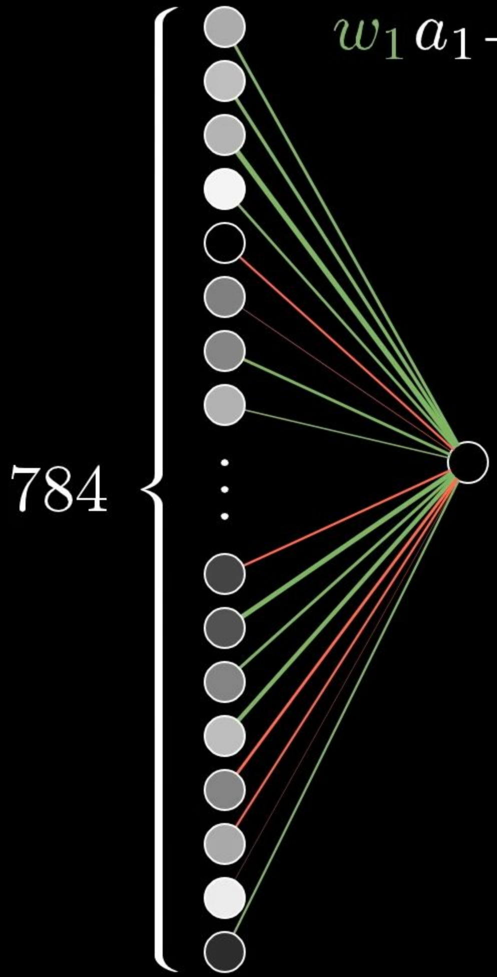
w_7: 2.01

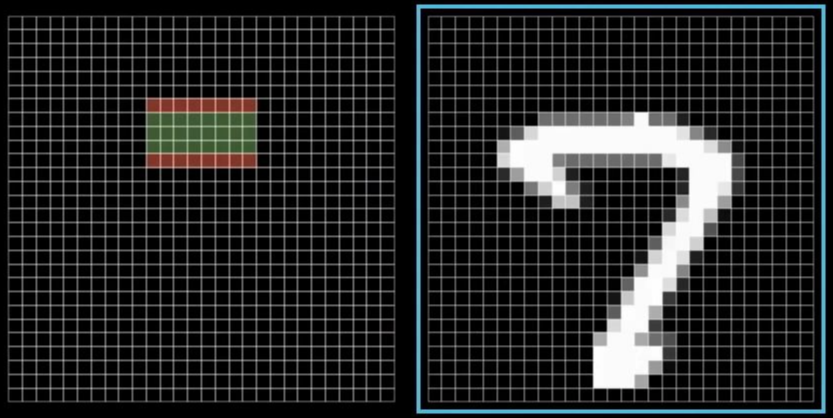
w_8: 1.07

$\vdots$

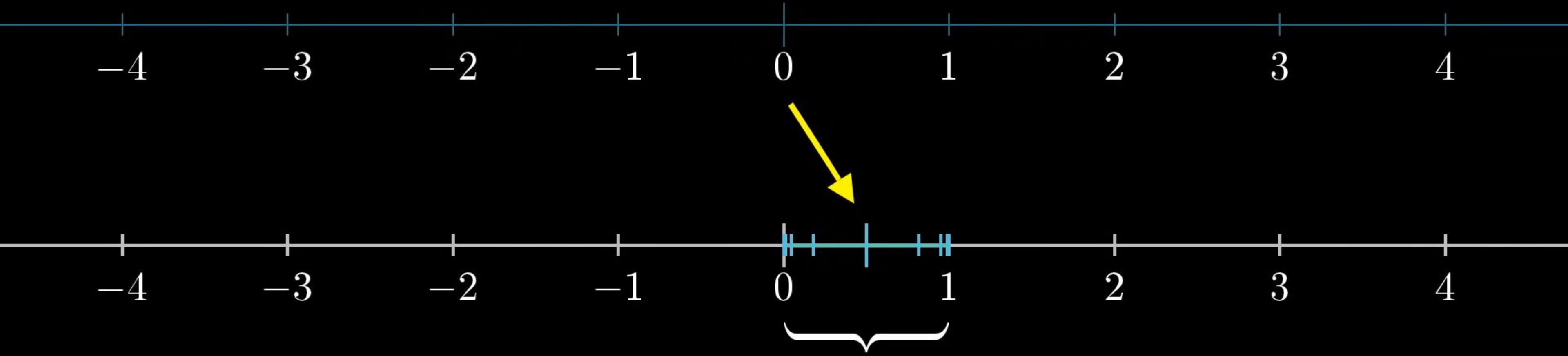


$$w_1 a_1 + w_2 a_2 + w_3 a_3 + w_4 a_4 + \cdots + w_n a_n$$





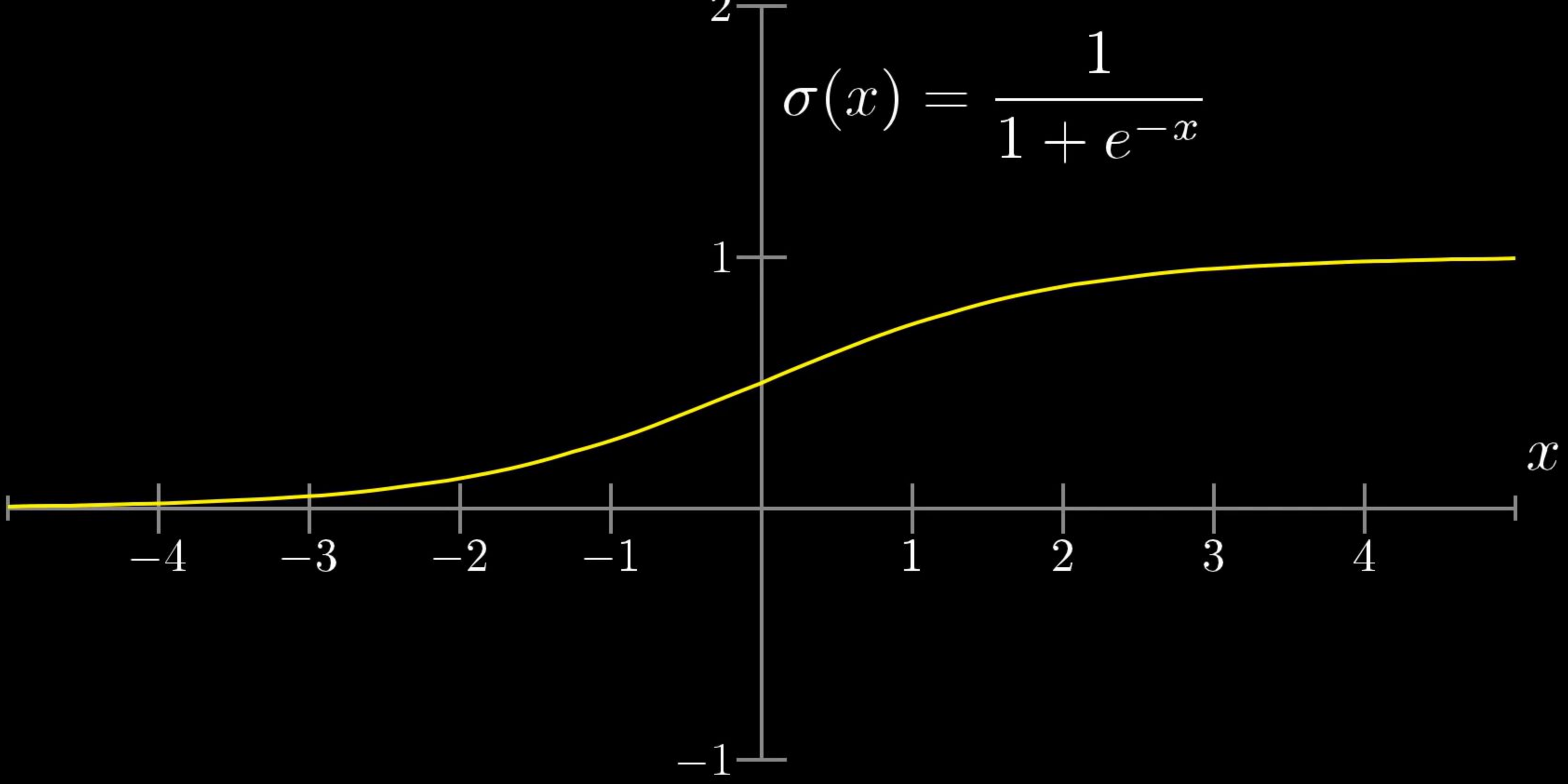
$$w_1 a_1 + w_2 a_2 + w_3 a_3 + w_4 a_4 + \cdots + w_n a_n$$



Activations should be in this range

Sigmoid

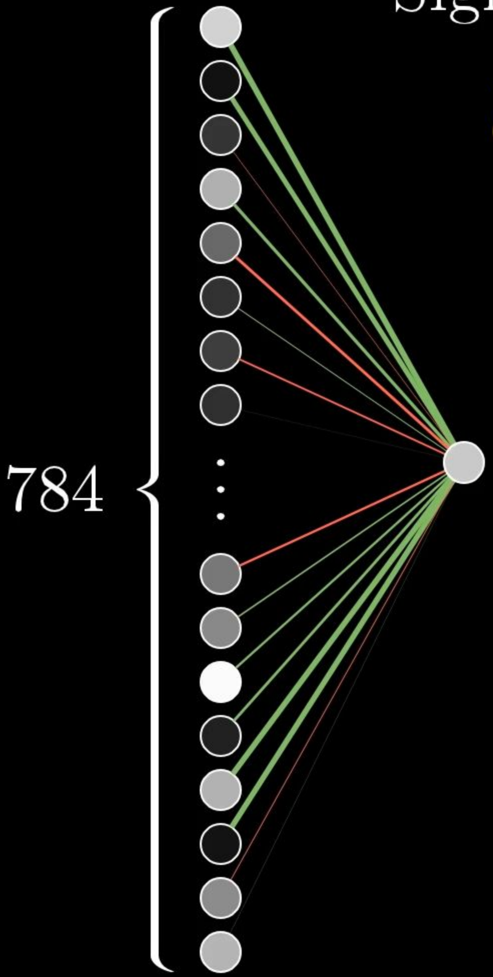
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



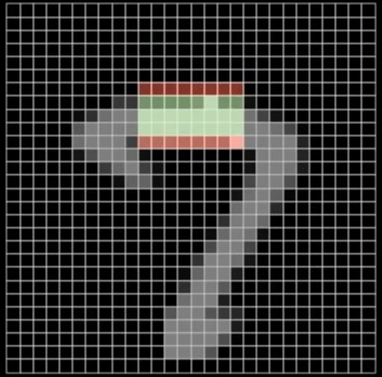
Sigmoid

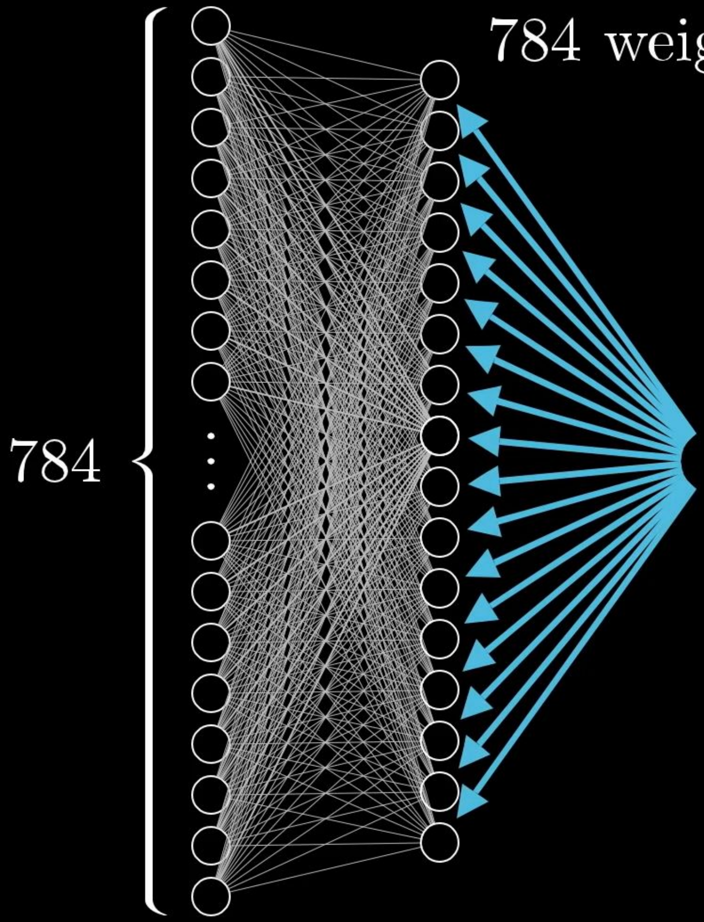
How positive is this?

$$\sigma\left(w_1 a_1 + w_2 a_2 + w_3 a_3 + \cdots + w_n a_n - 10\right)$$



Only activate meaningfully when weighted sum > 10

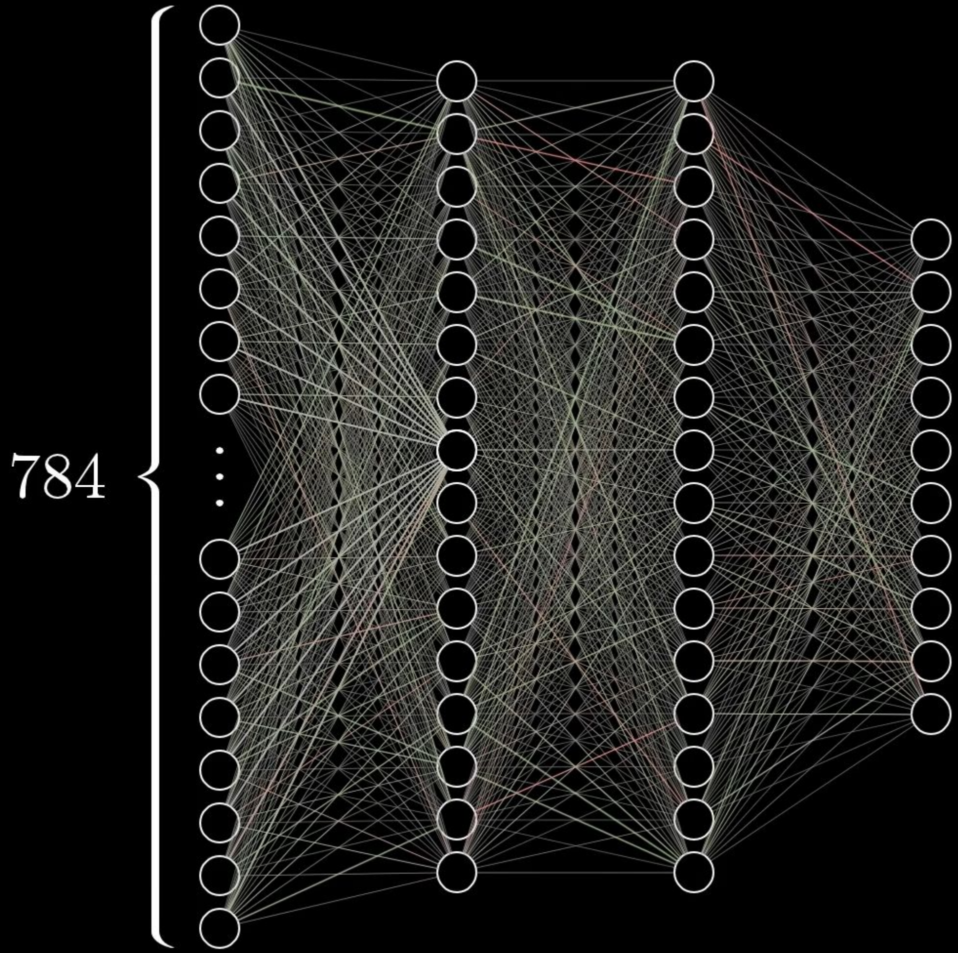




784 weights per neuron

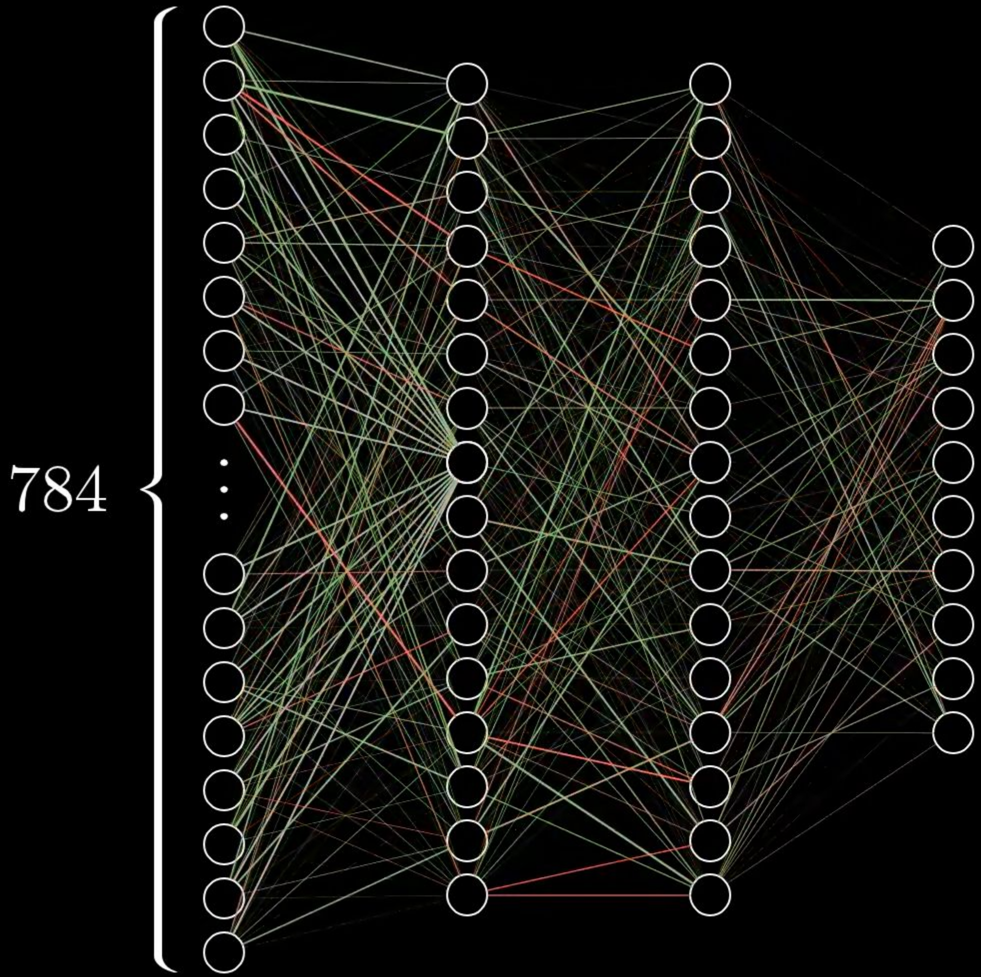
784

One bias for each



$$\boxed{\underset{\text{weights}}{784\times16+16\times16+16\times10} \qquad \underset{\text{biases}}{16+16+10}}$$

$$13{,}002$$



$$784\times16+16\times16+16\times10$$

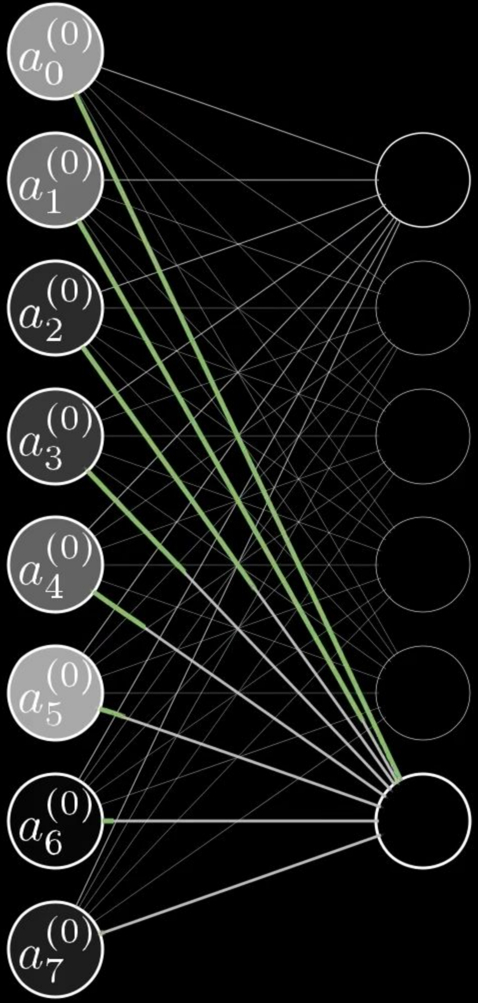
weights

$$16\ +16+10$$

biases

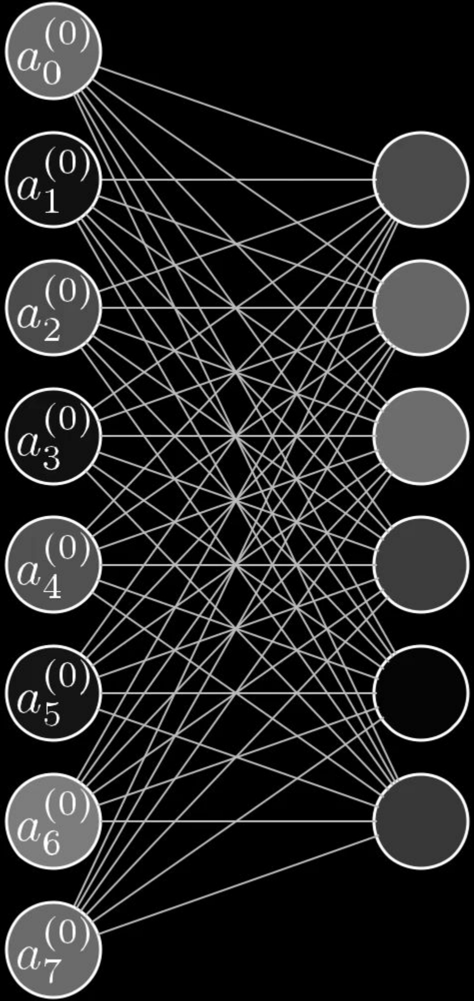
13,002

Learning → Finding the right weights and biases



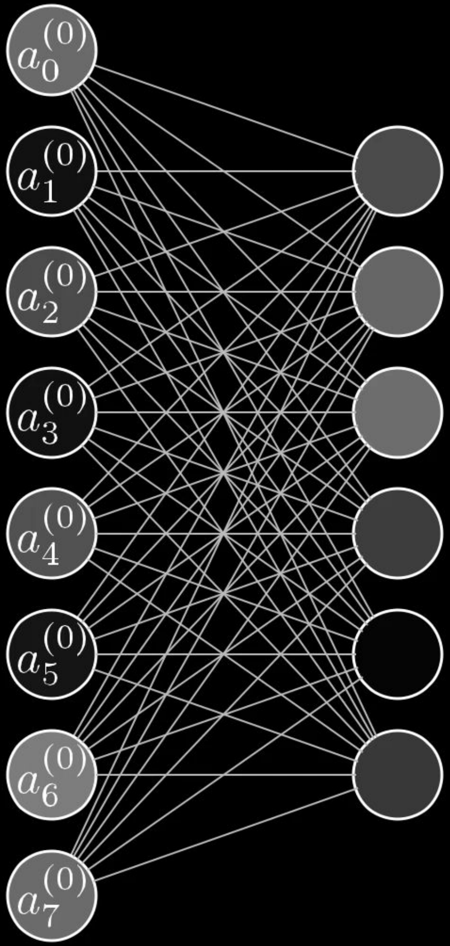
$$a_0^{(1)} = \overset{\substack{\text{Sigmoid}\\ \downarrow}}{\sigma}\left(w_{0,0}\; a_0^{(0)} + w_{0,1}\; a_1^{(0)} + \cdots + w_{0,n}\; a_n^{(0)} + \underset{\substack{\uparrow\\ \text{Bias}}}{b_0} \right)$$

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ \vdots \\ \boxed{?} \end{bmatrix}$$

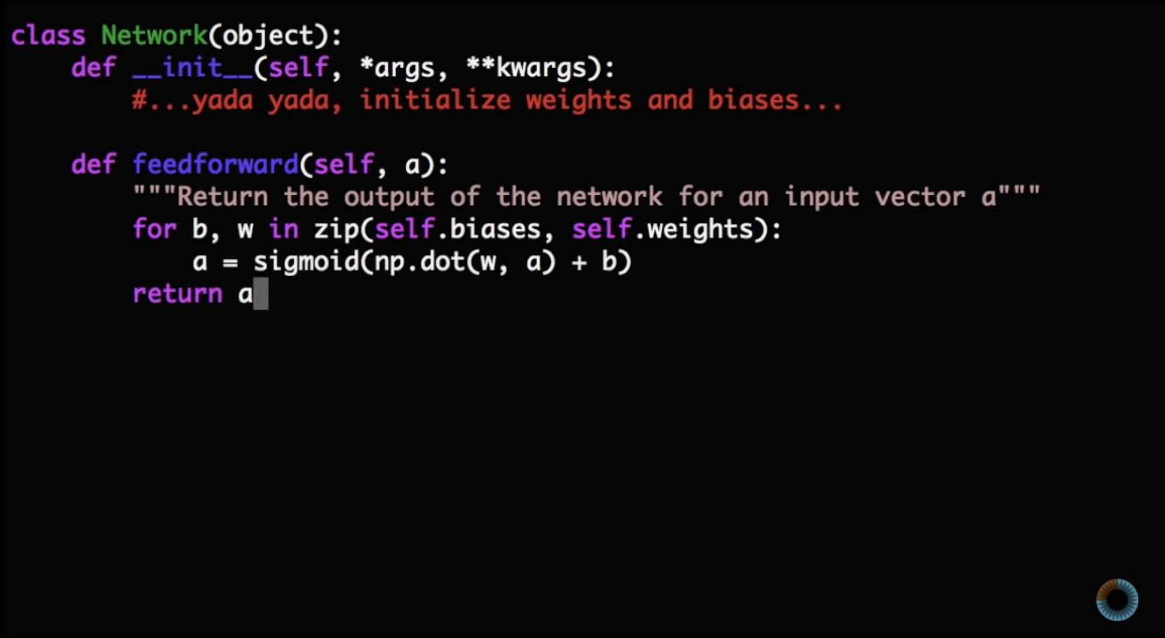


$$\mathbf{a}^{(1)} = \sigma\left(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b}\right)$$

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}\right)$$

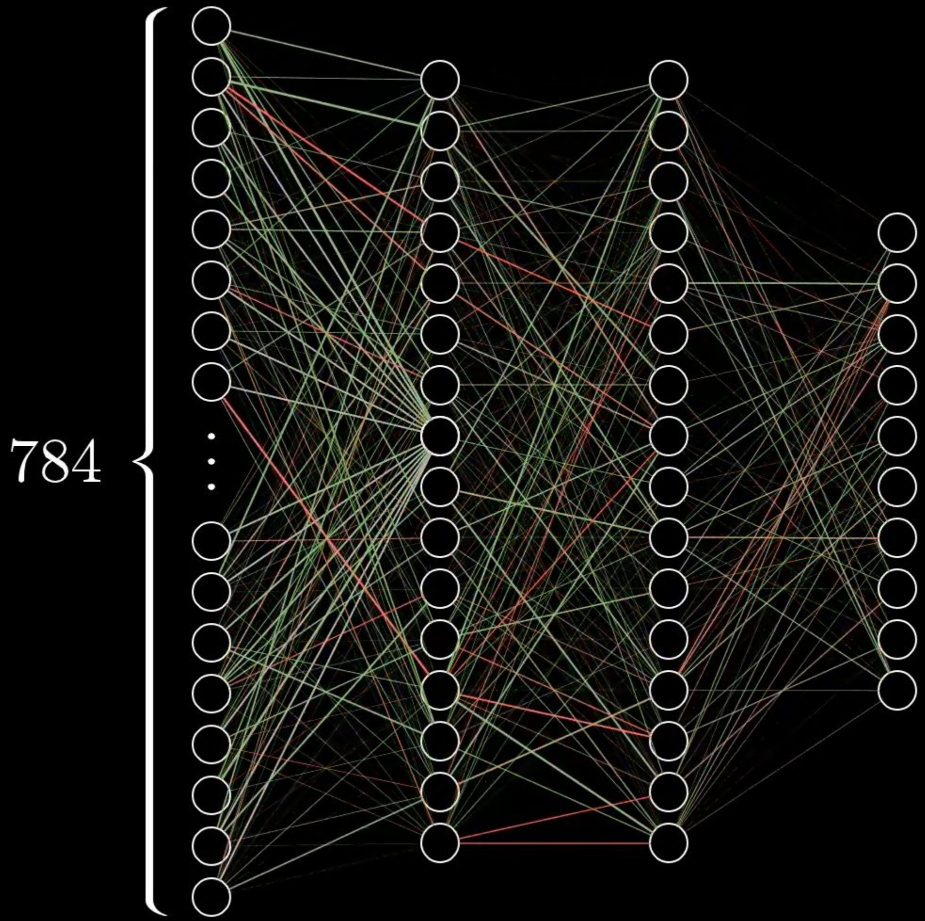


$$\mathbf{a}^{(1)} = \sigma\left(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b}\right)$$



```python
class Network(object):
    def __init__(self, *args, **kwargs):
        #...yada yada, initialize weights and biases...

    def feedforward(self, a):
        """Return the output of the network for an input vector a"""
        for b, w in zip(self.biases, self.weights):
            a = sigmoid(np.dot(w, a) + b)
        return a
```



$$784\times16+16\times16+16\times10$$

weights

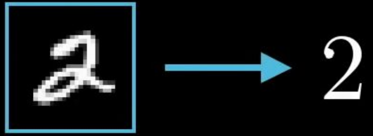
$$16\ +16+10$$

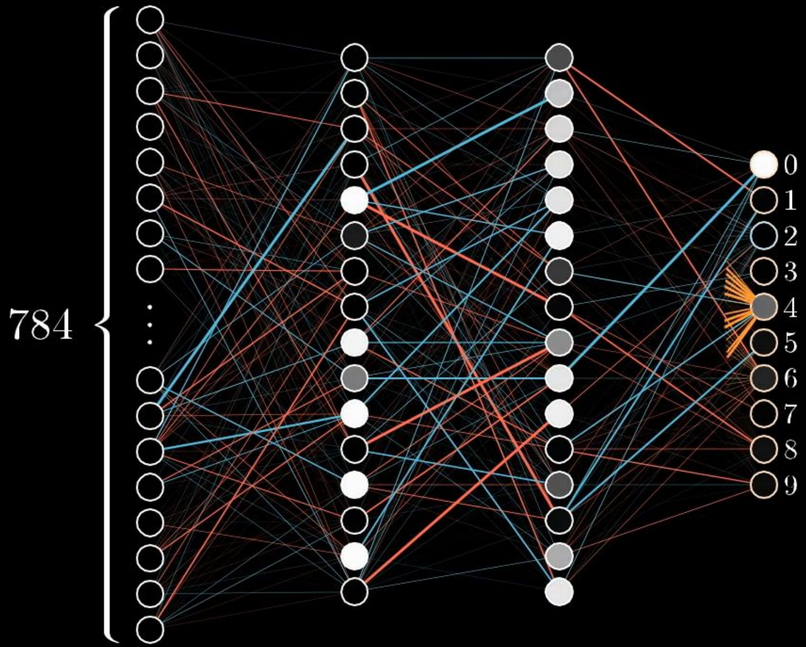
biases

13,002

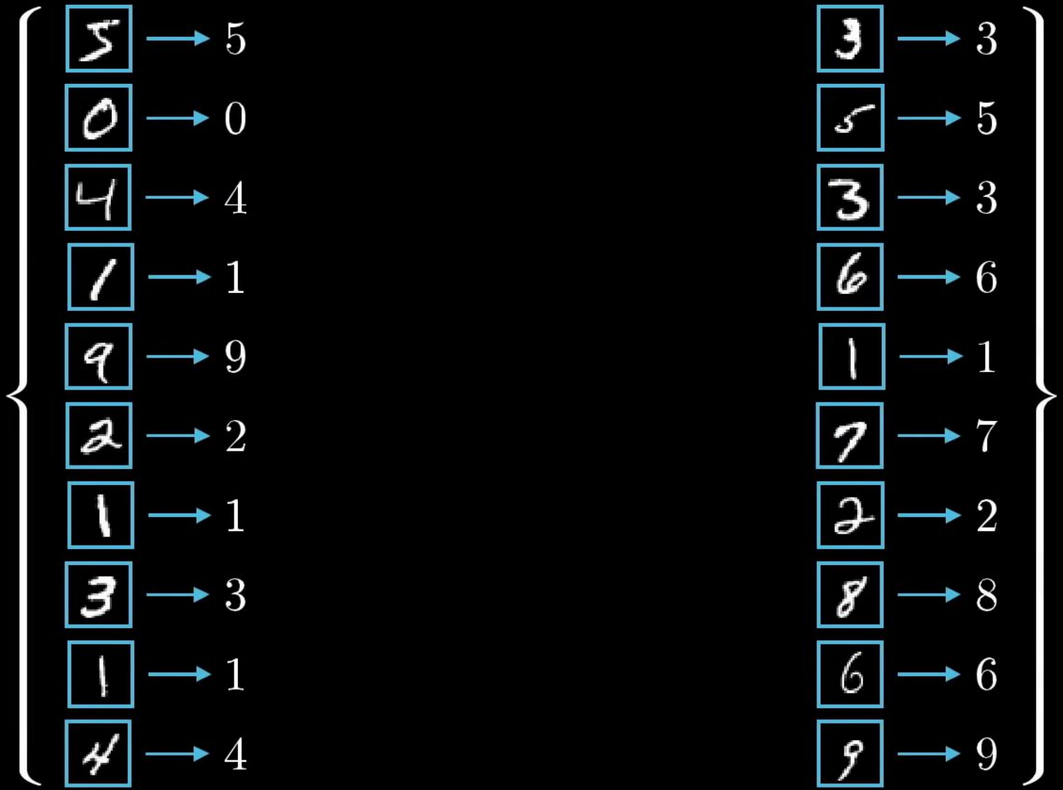
Learning → Finding the right weights and biases

Training in progress. . .

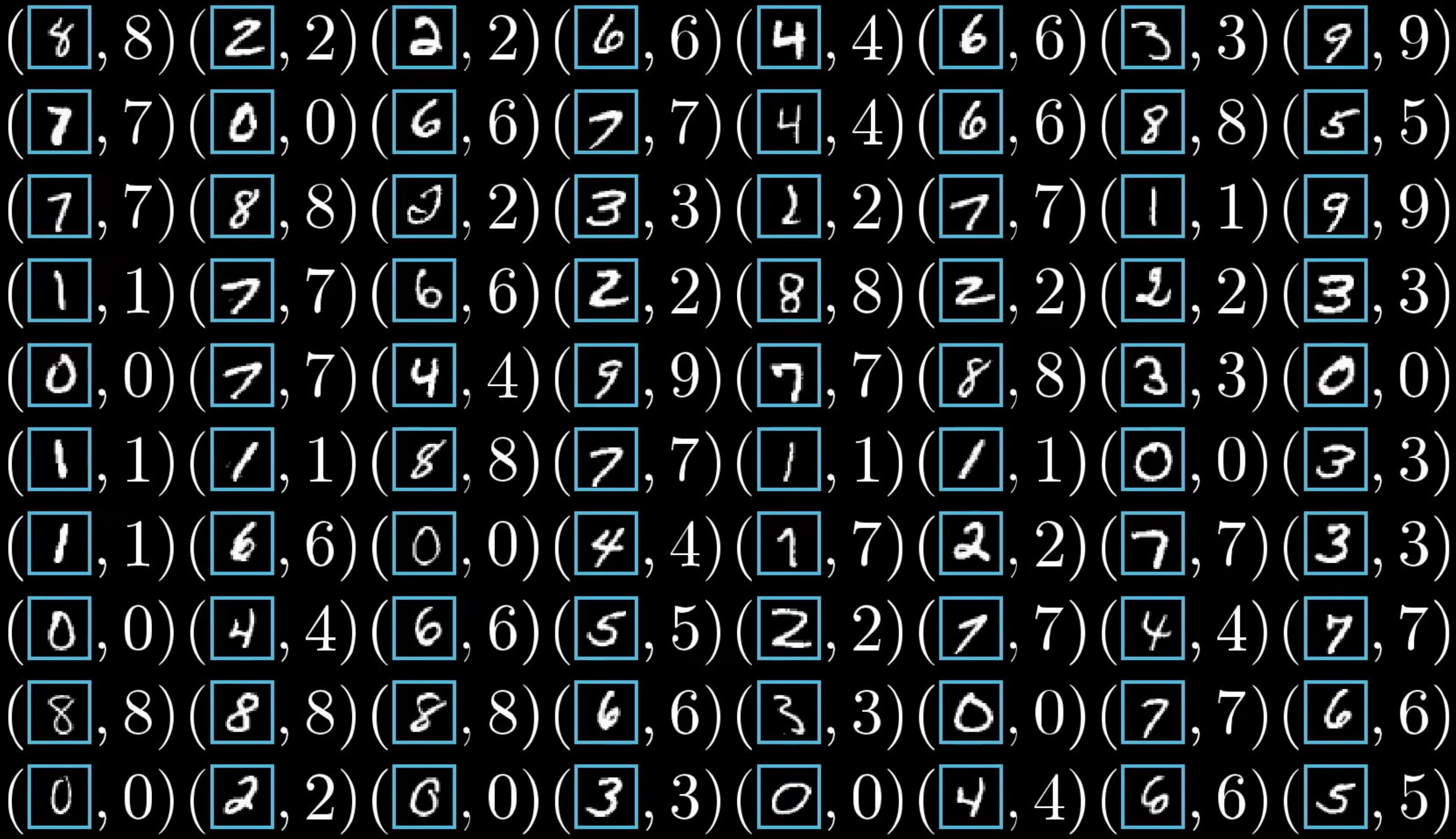


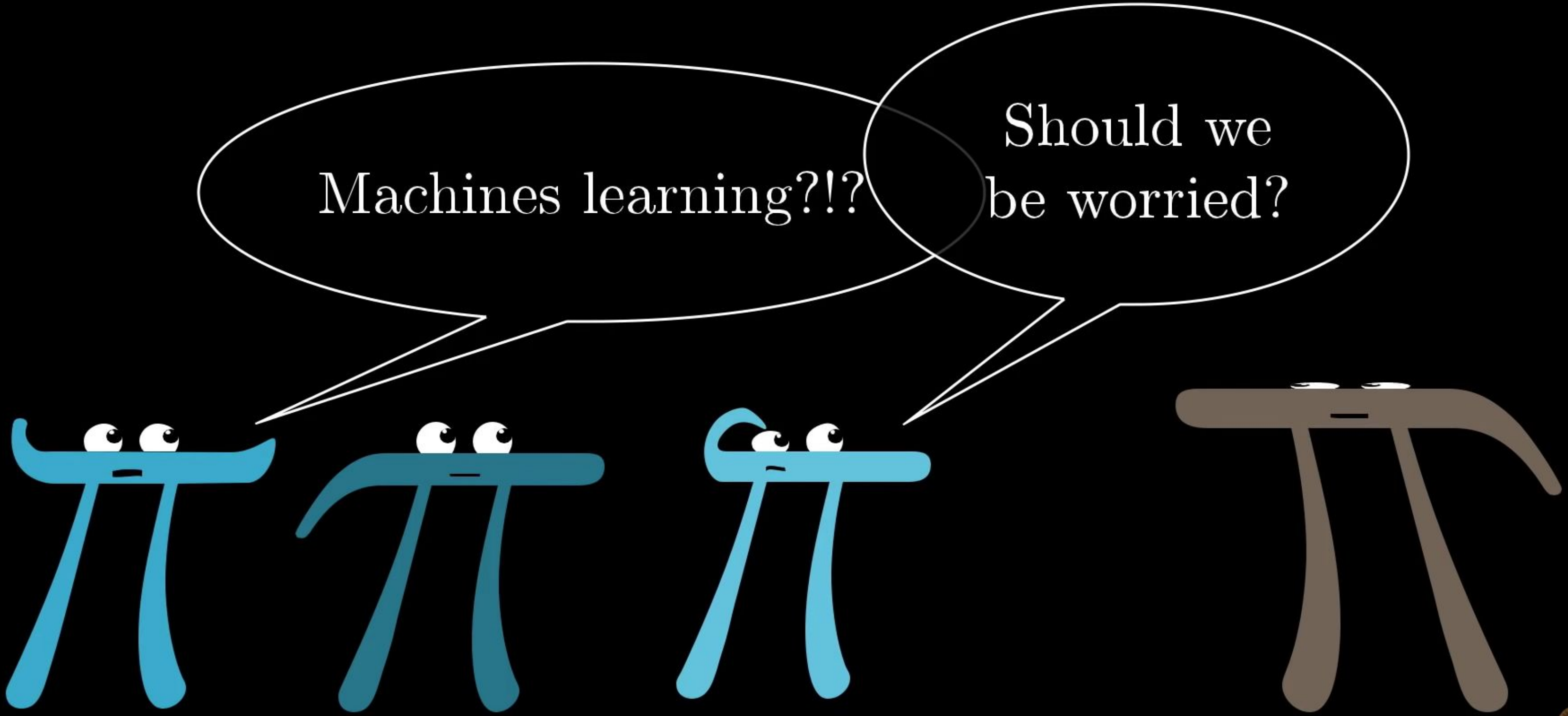


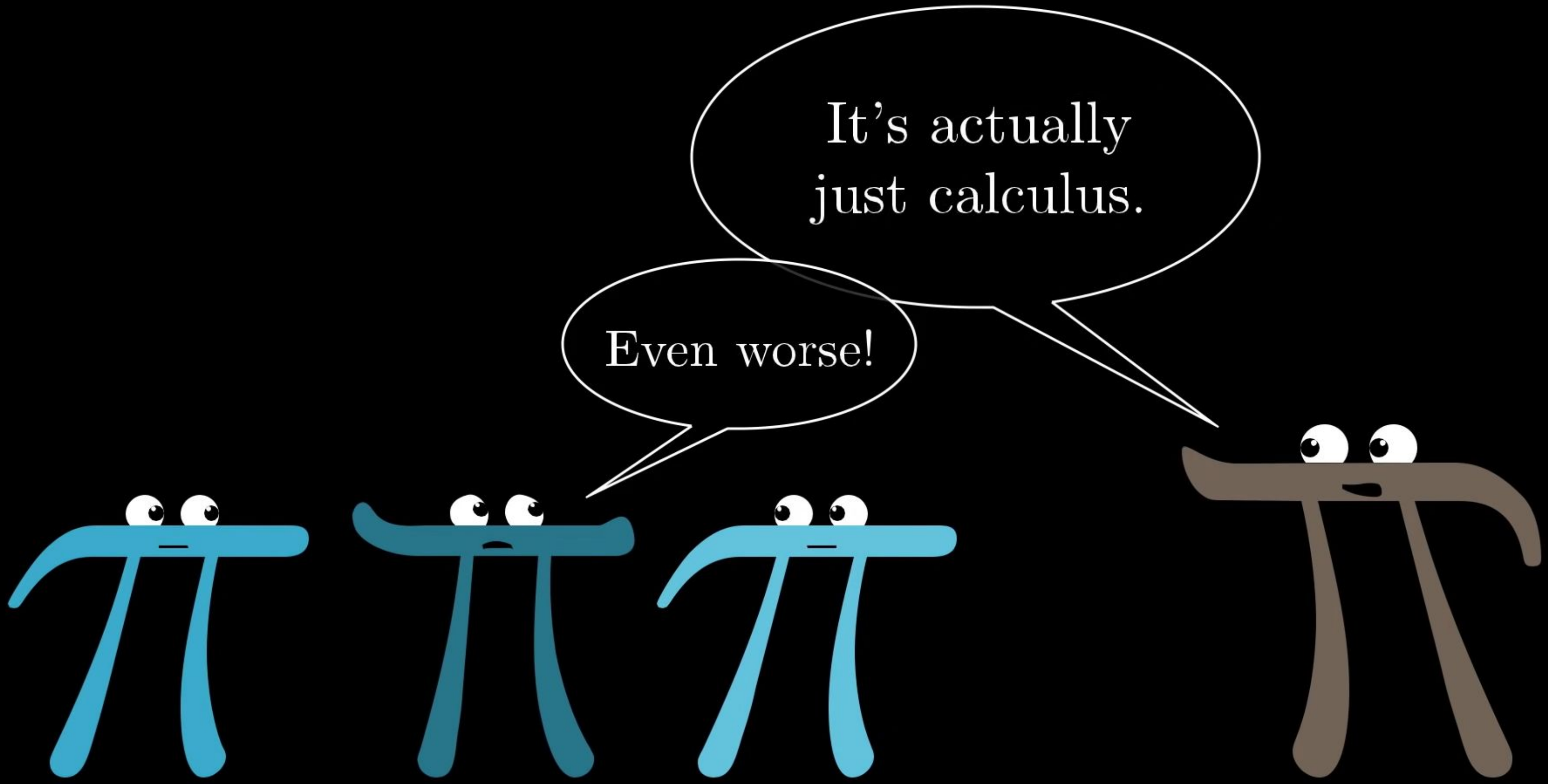
Train on these

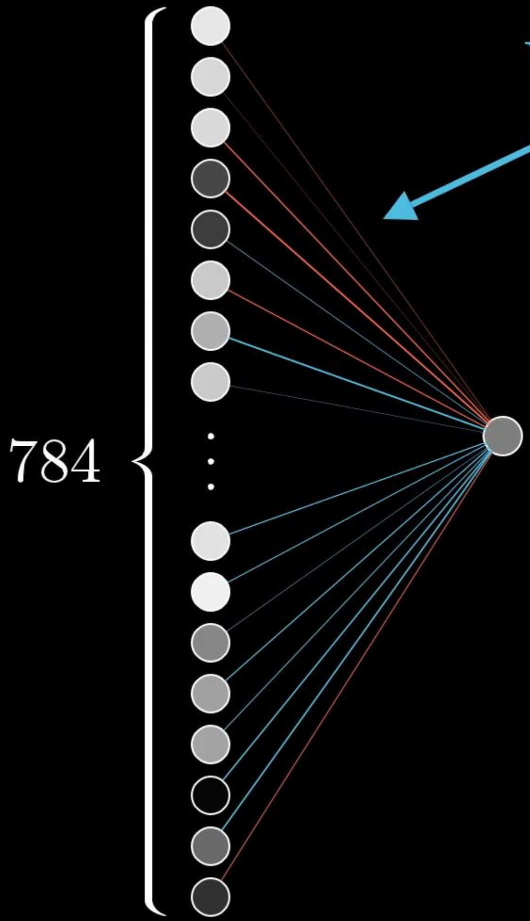


Test on these


(, 8) (, 2) (, 2) (, 6) (, 4) (, 6) (, 3) (, 9)
(, 7) (, 0) (, 6) (, 7) (, 4) (, 6) (, 8) (, 5)
(, 7) (, 8) (, 2) (, 3) (, 2) (, 7) (, 1) (, 9)
(, 1) (, 7) (, 6) (, 2) (, 8) (, 2) (, 2) (, 3)
(, 0) (, 7) (, 4) (, 9) (, 7) (, 8) (, 3) (, 0)
(, 1) (, 1) (, 8) (, 7) (, 1) (, 1) (, 0) (, 3)
(, 1) (, 6) (, 0) (, 4) (, 7) (, 2) (, 7) (, 3)
(, 0) (, 4) (, 6) (, 5) (, 2) (, 7) (, 4) (, 7)
(, 8) (, 8) (, 8) (, 6) (, 3) (, 0) (, 7) (, 6)
(, 0) (, 2) (, 0) (, 3) (, 0) (, 4) (, 6) (, 5)


Machines learning?!?
Should we be worried?

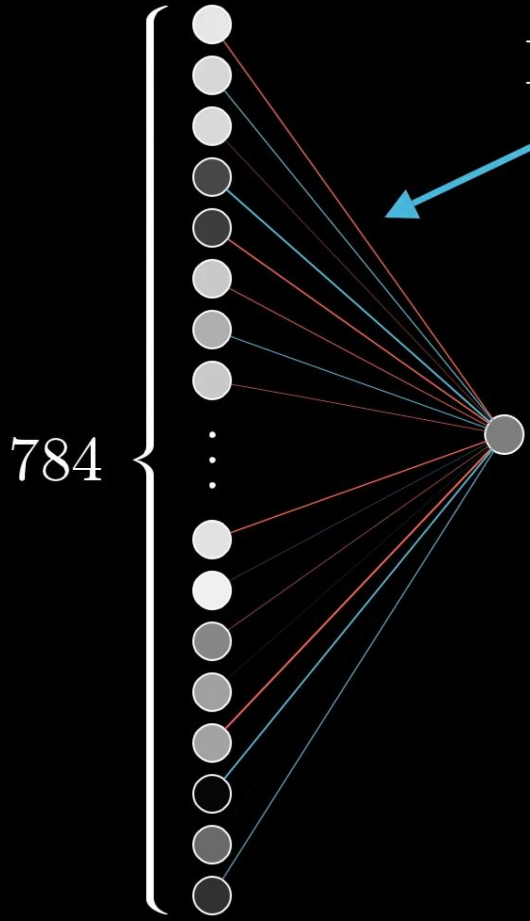

It's actually just calculus.
Even worse!



Weights

Bias

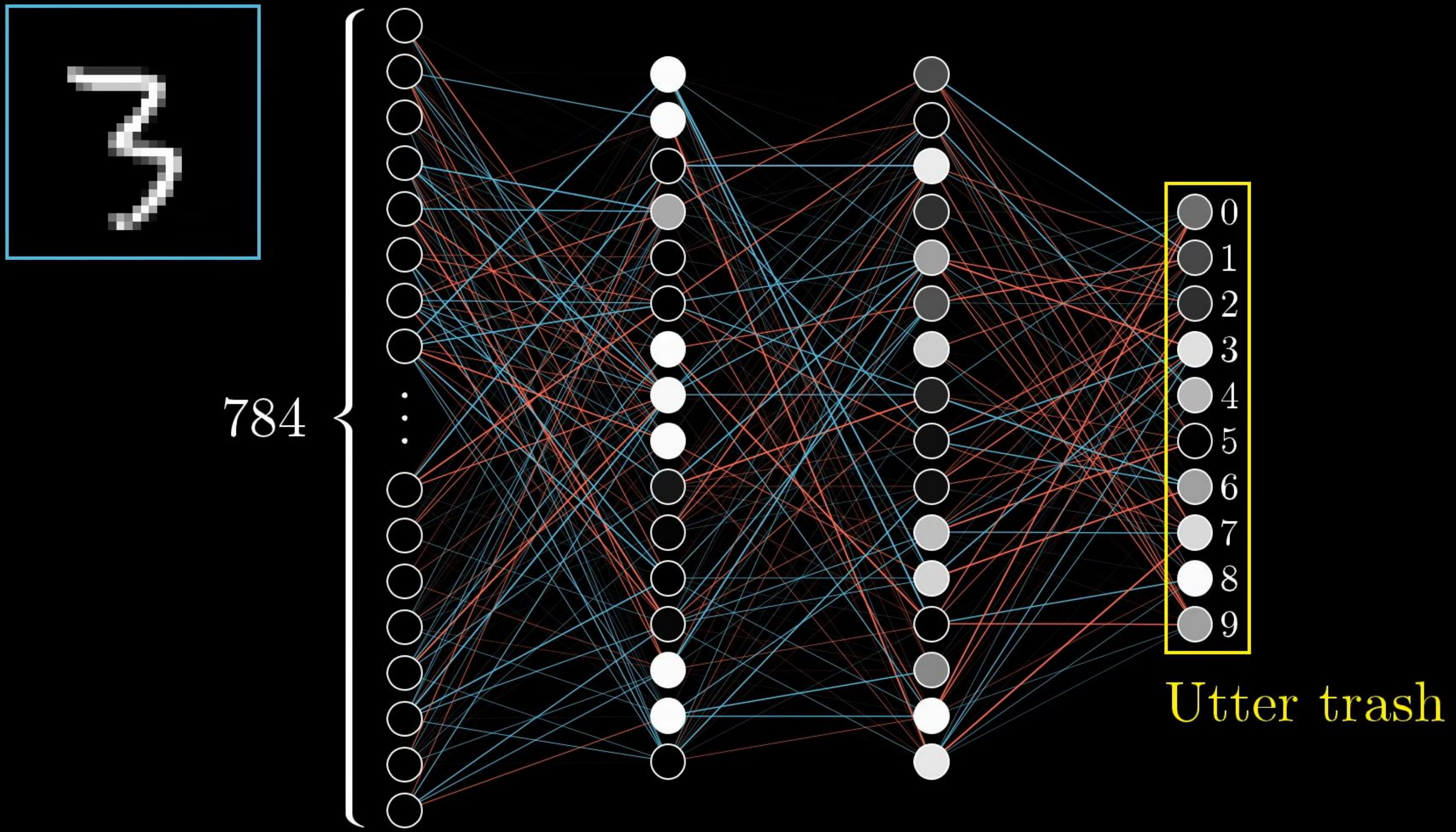
$$= \sigma(w_1 a_1 + w_2 a_2 + \cdots + w_n a_n + b)$$



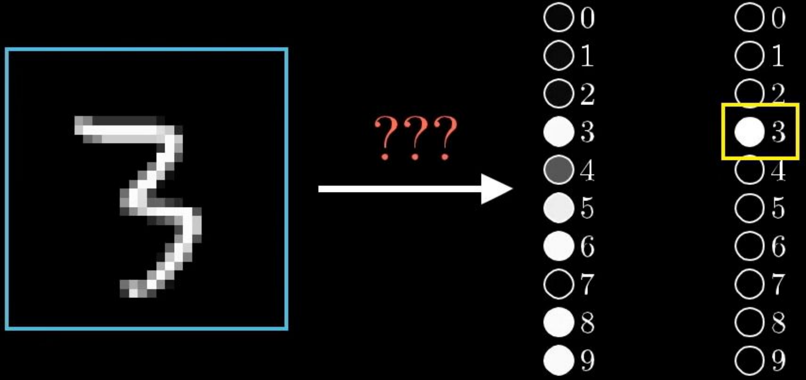
Initialize randomly

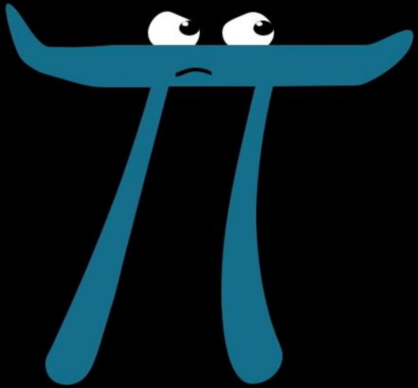
Initialize randomly

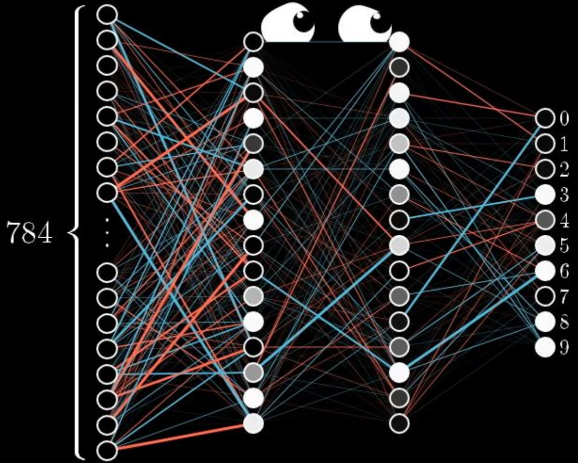
$$= \sigma(w_1 a_1 + w_2 a_2 + \cdots + w_n a_n + b)$$



Utter trash


???
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9

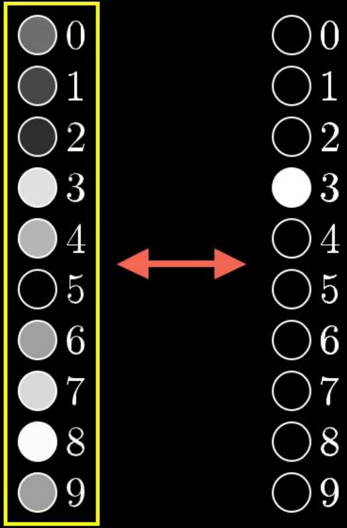



784
0
1
2
3
4
5
6
7
8
9

What's the "cost" of this difference?

Cost of

$$\left\{\begin{array}{l}(0.43-0.00)^2+\\(0.28-0.00)^2+\\(0.19-0.00)^2+\\(0.88-1.00)^2+\\(0.72-0.00)^2+\\(0.01-0.00)^2+\\(0.64-0.00)^2+\\(0.86-0.00)^2+\\(0.99-0.00)^2+\\(0.63-0.00)^2\end{array}\right.$$

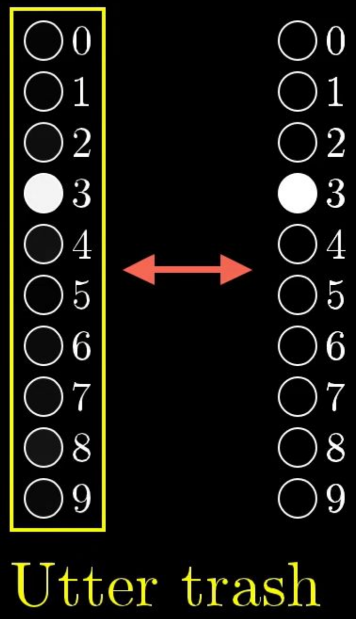


Utter trash

Cost of

$$0.03 \left\{ \begin{array}{lcl} 0.0006 & \leftarrow & (0.02 - 0.00)^2 + \\ 0.0007 & \leftarrow & (0.03 - 0.00)^2 + \\ 0.0039 & \leftarrow & (0.06 - 0.00)^2 + \\ 0.0009 & \leftarrow & (0.97 - 1.00)^2 + \\ 0.0055 & \leftarrow & (0.07 - 0.00)^2 + \\ 0.0004 & \leftarrow & (0.02 - 0.00)^2 + \\ 0.0022 & \leftarrow & (0.05 - 0.00)^2 + \\ 0.0033 & \leftarrow & (0.06 - 0.00)^2 + \\ 0.0072 & \leftarrow & (0.08 - 0.00)^2 + \\ 0.0018 & \leftarrow & (0.04 - 0.00)^2 \end{array} \right.$$

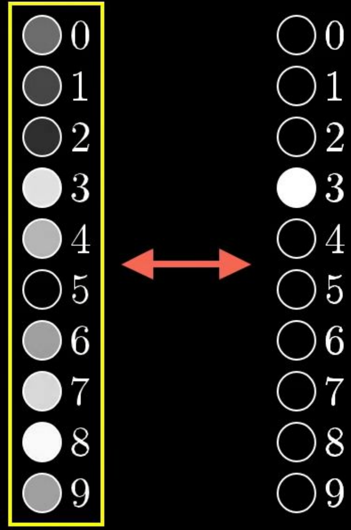
What's the "cost" of this difference?



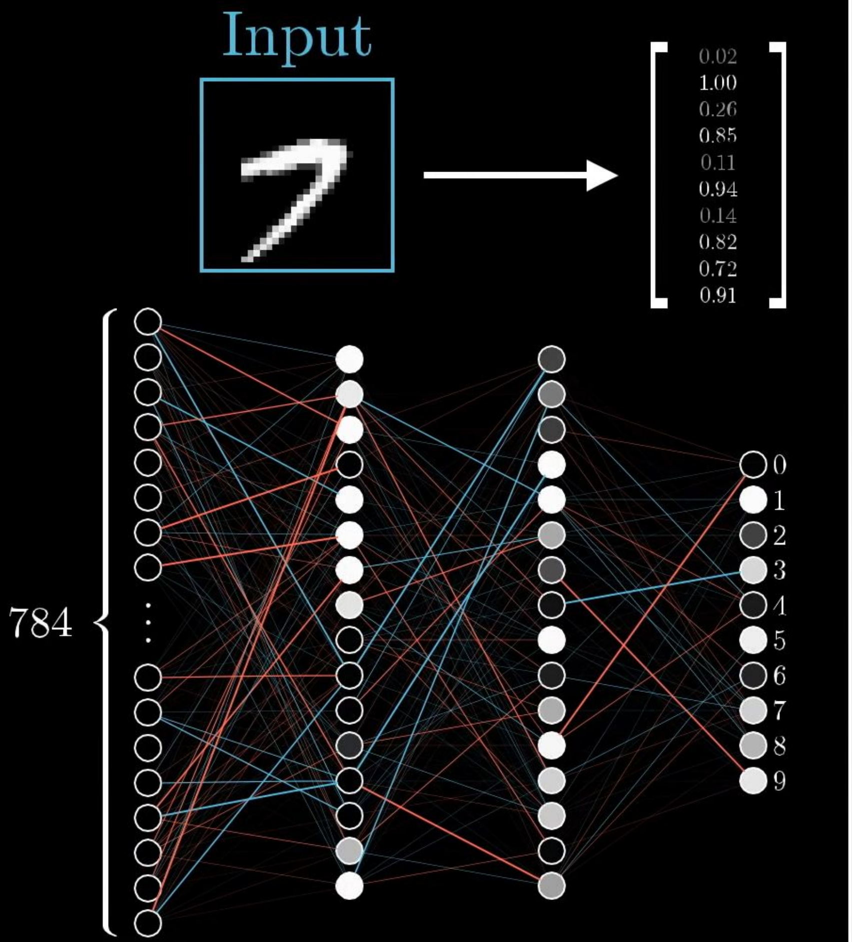
Cost of

$$3.37 \left\{ \begin{array}{l} 0.1863 \leftarrow (0.43 - 0.00)^2 + \\ 0.0809 \leftarrow (0.28 - 0.00)^2 + \\ 0.0357 \leftarrow (0.19 - 0.00)^2 + \\ 0.0138 \leftarrow (0.88 - 1.00)^2 + \\ 0.5242 \leftarrow (0.72 - 0.00)^2 + \\ 0.0001 \leftarrow (0.01 - 0.00)^2 + \\ 0.4079 \leftarrow (0.64 - 0.00)^2 + \\ 0.7388 \leftarrow (0.86 - 0.00)^2 + \\ 0.9817 \leftarrow (0.99 - 0.00)^2 + \\ 0.3998 \leftarrow (0.63 - 0.00)^2 \end{array} \right.$$

What's the "cost" of this difference?



Utter trash



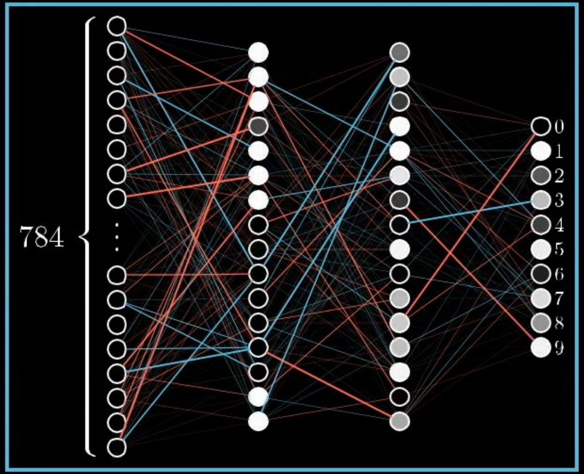
Neural network function

Input: 784 numbers (pixels)

Output: 10 numbers

Parameters: 13,002 weights/biases

Input



Cost: 5.4

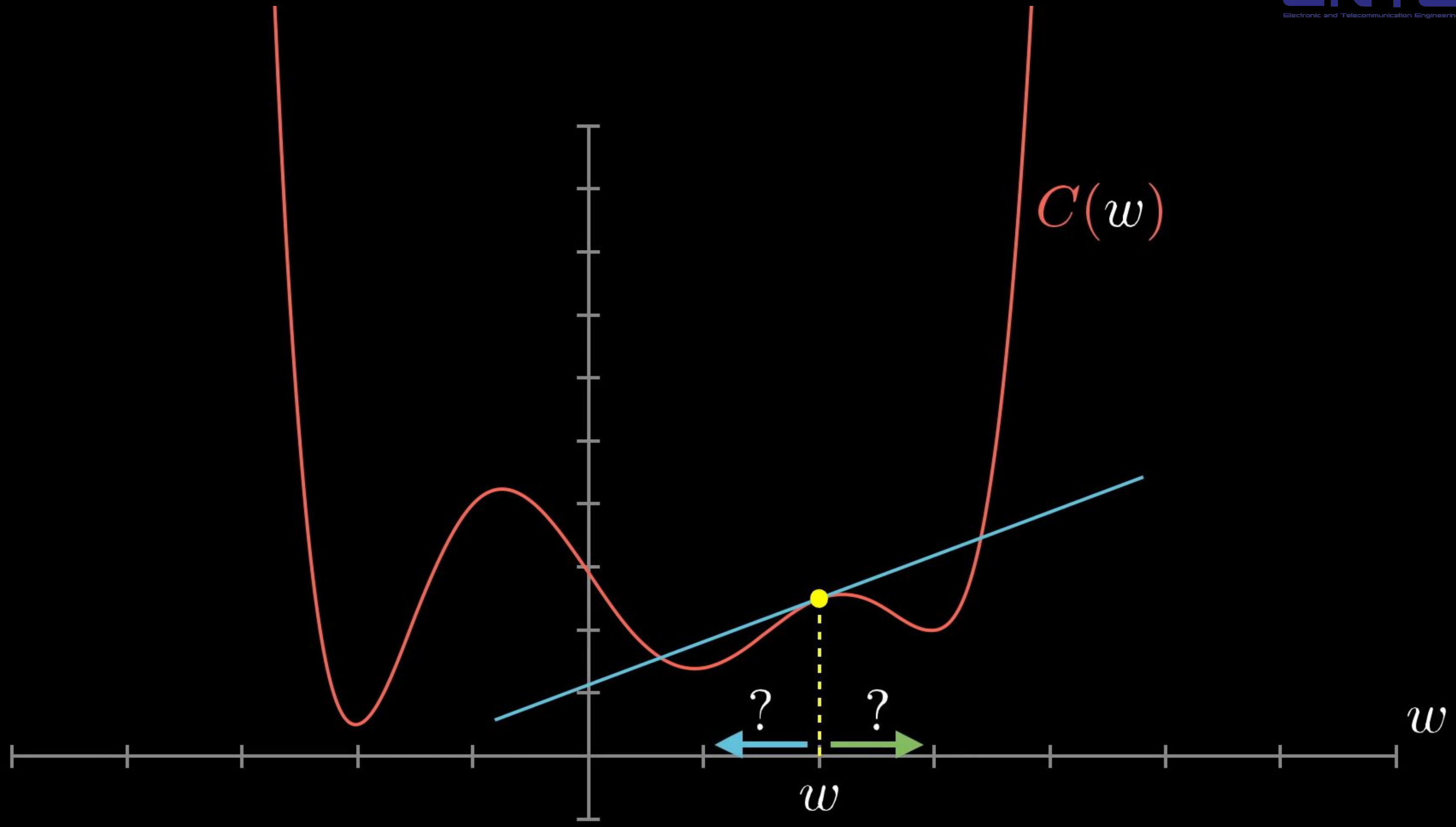
Cost function

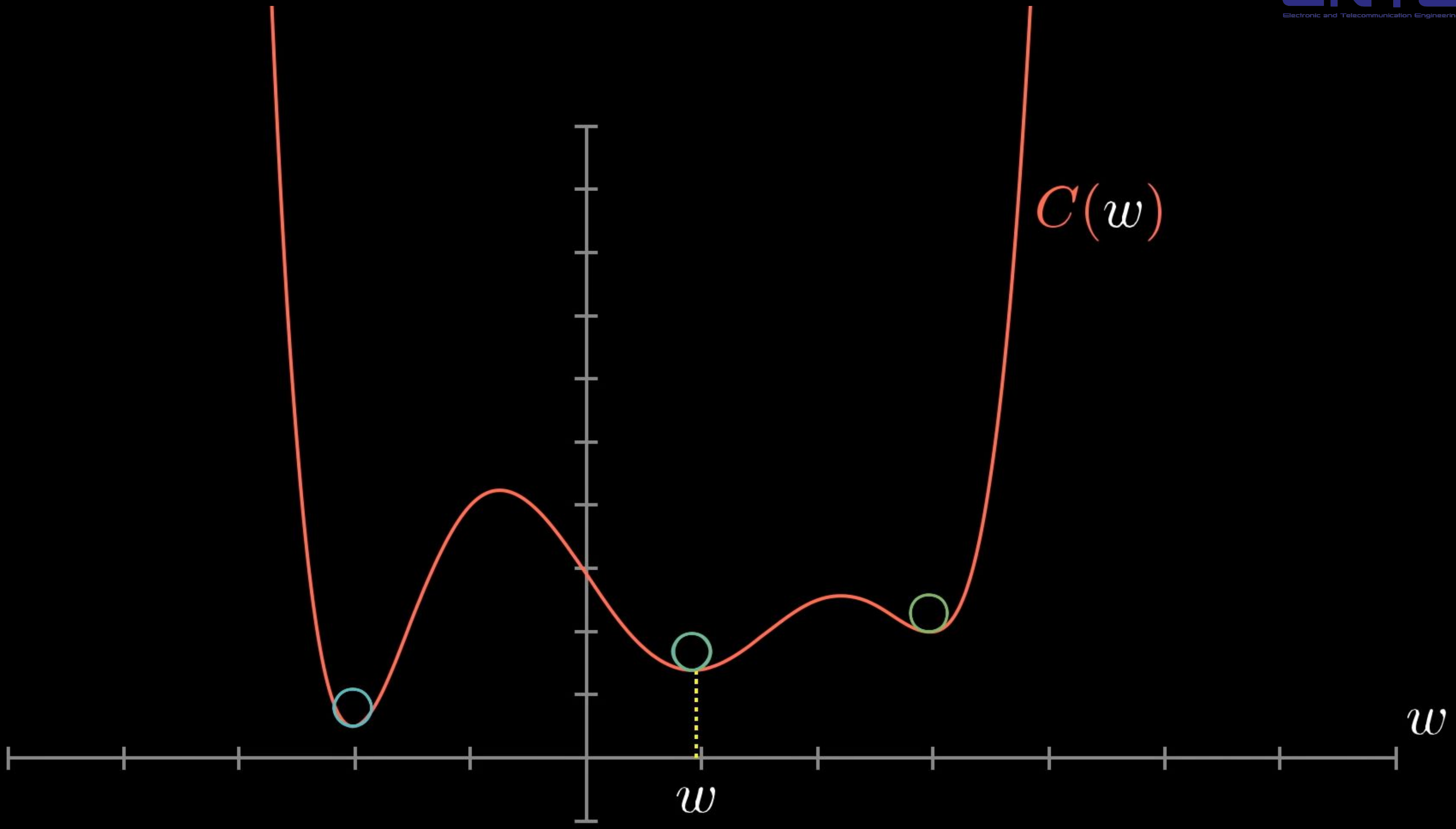
Input: 13,002 weights/biases

Output: 1 number (the cost)

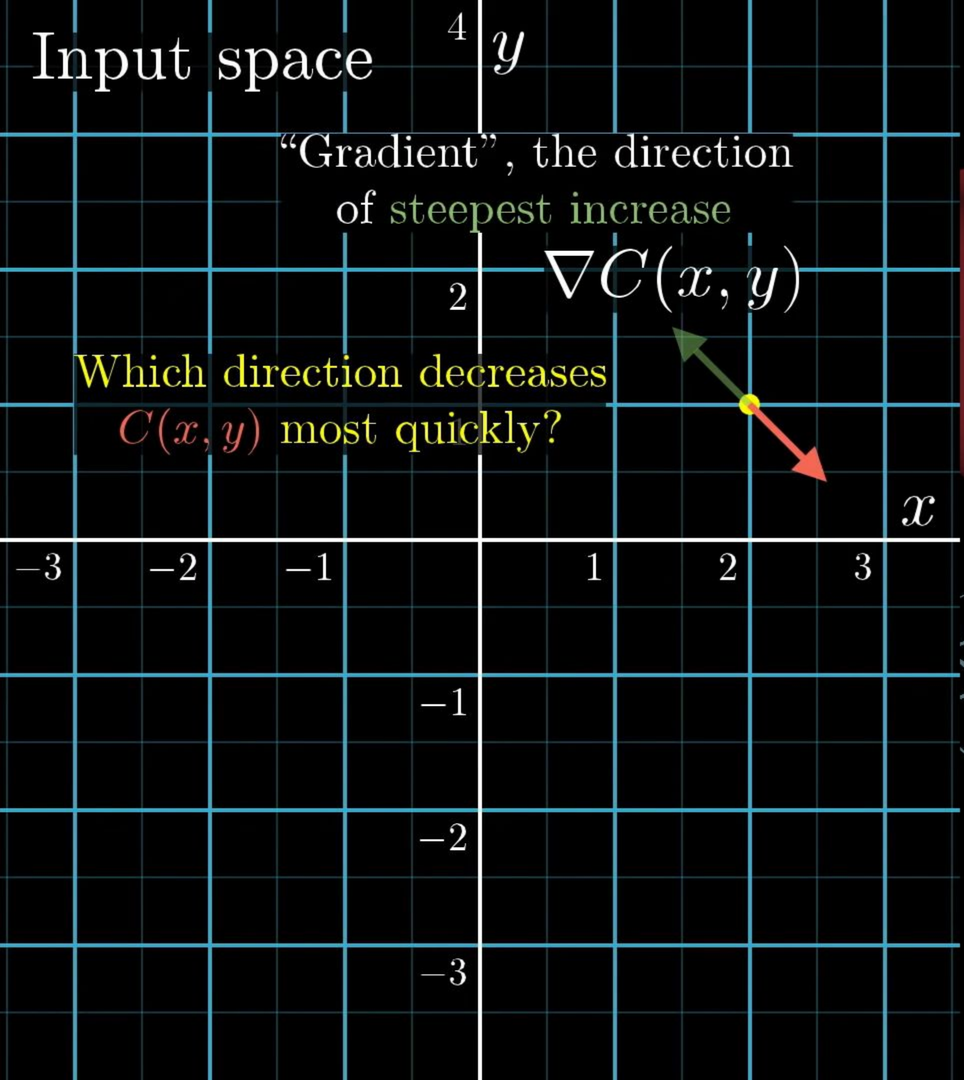
Parameters: Many, many, many training examples

(, 3)


$C(w)$
?
?
w
w


$C(w)$
w
w

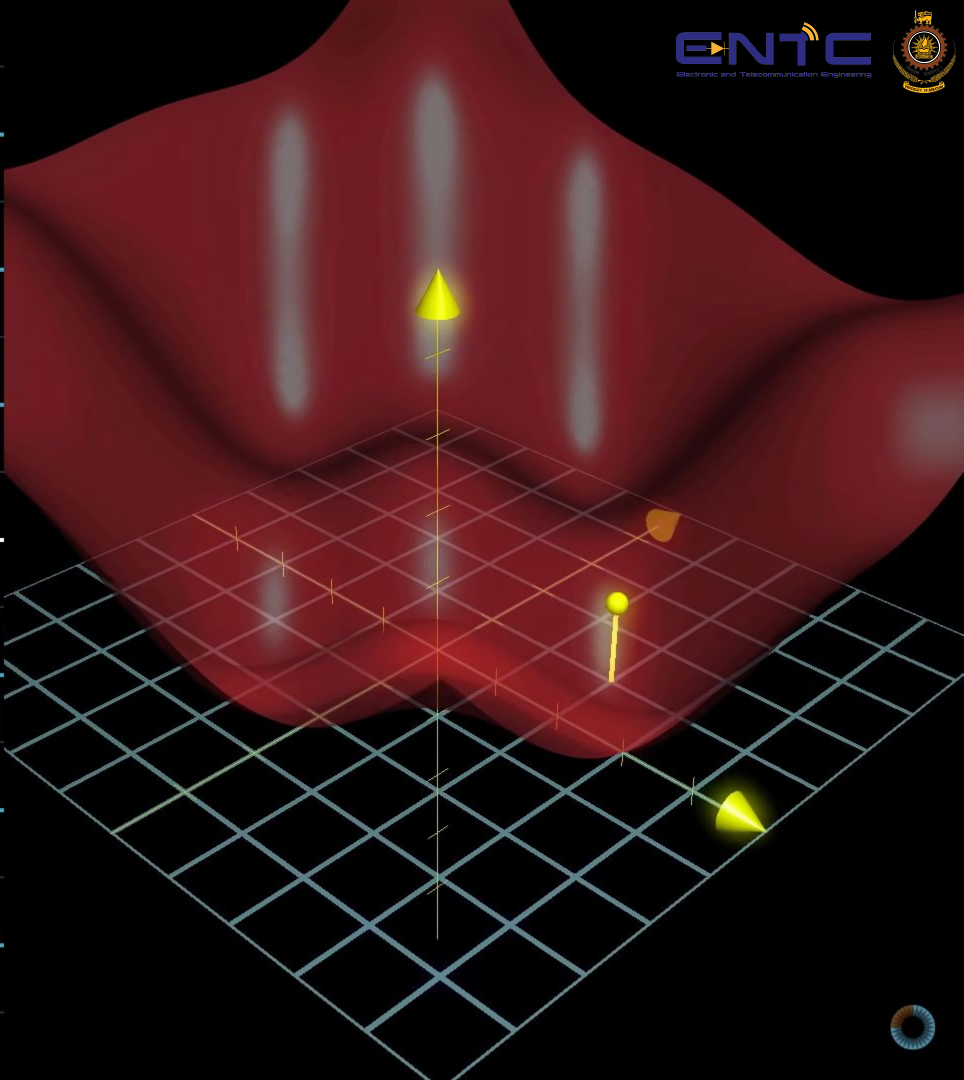
Input space



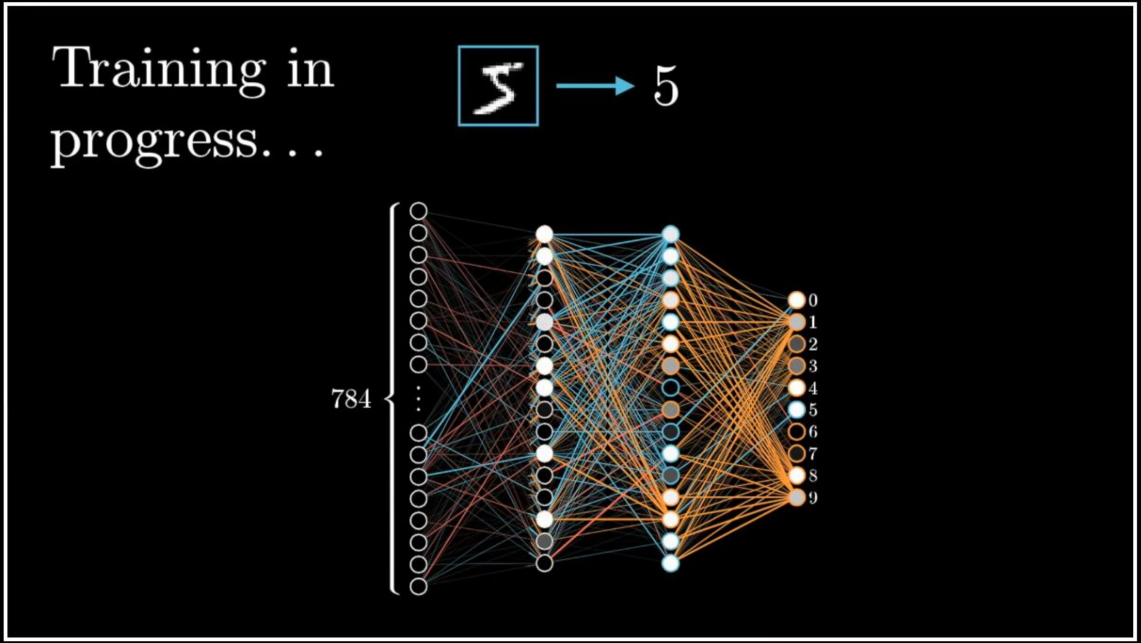
“Gradient”, the direction of steepest increase

$\nabla C(x, y)$

Which direction decreases $C(x, y)$ most quickly?

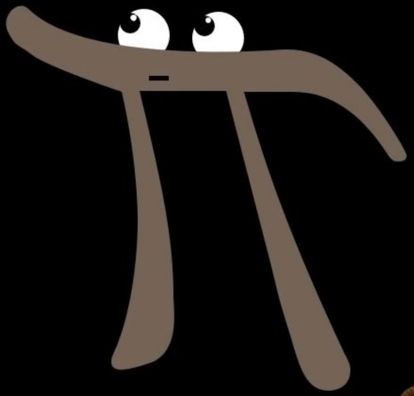


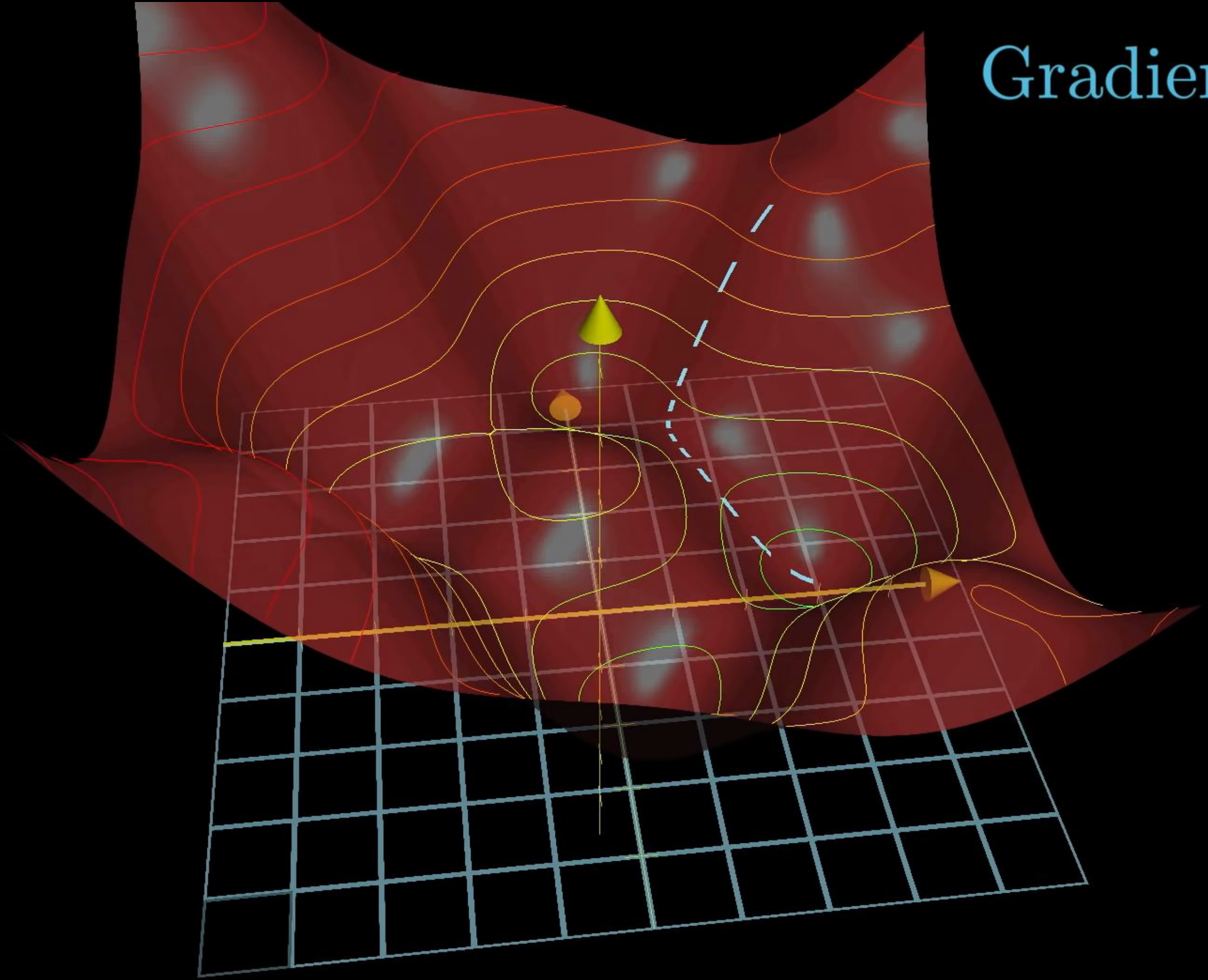
Backpropagation



Training in progress. . .

5





Gradient descent

$$\vec{\mathbf{W}} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{13,000} \\ w_{13,001} \\ w_{13,002} \end{bmatrix}$$

$$-\nabla C(\vec{\mathbf{W}}) = \begin{bmatrix} 0.31 \\ 0.03 \\ -1.25 \\ \vdots \\ 0.78 \\ -0.37 \\ 0.16 \end{bmatrix}$$

w_0 should increase somewhat

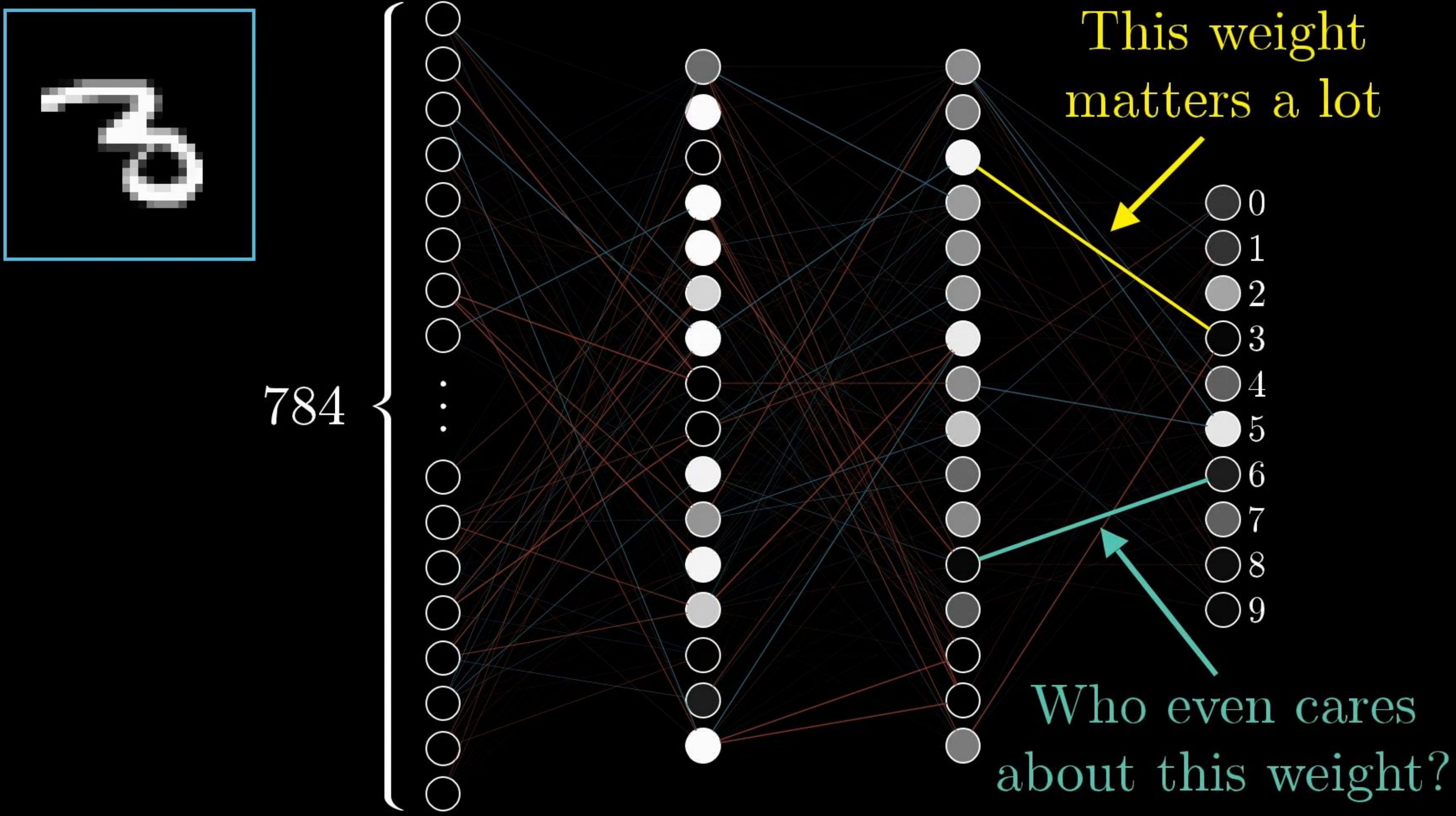
w_1 should increase a little

w_2 should decrease a lot

$w_{13,000}$ should increase a lot

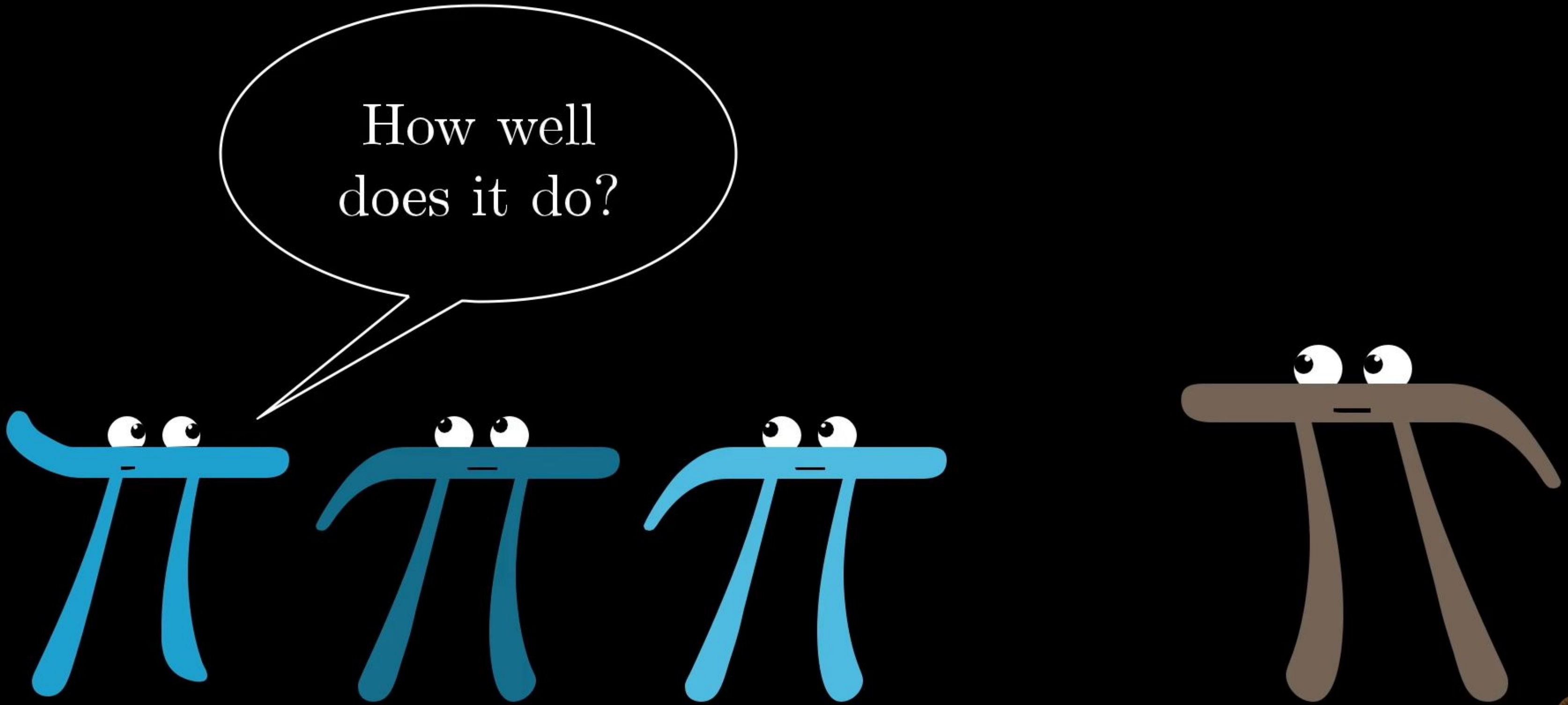
$w_{13,001}$ should decrease somewhat

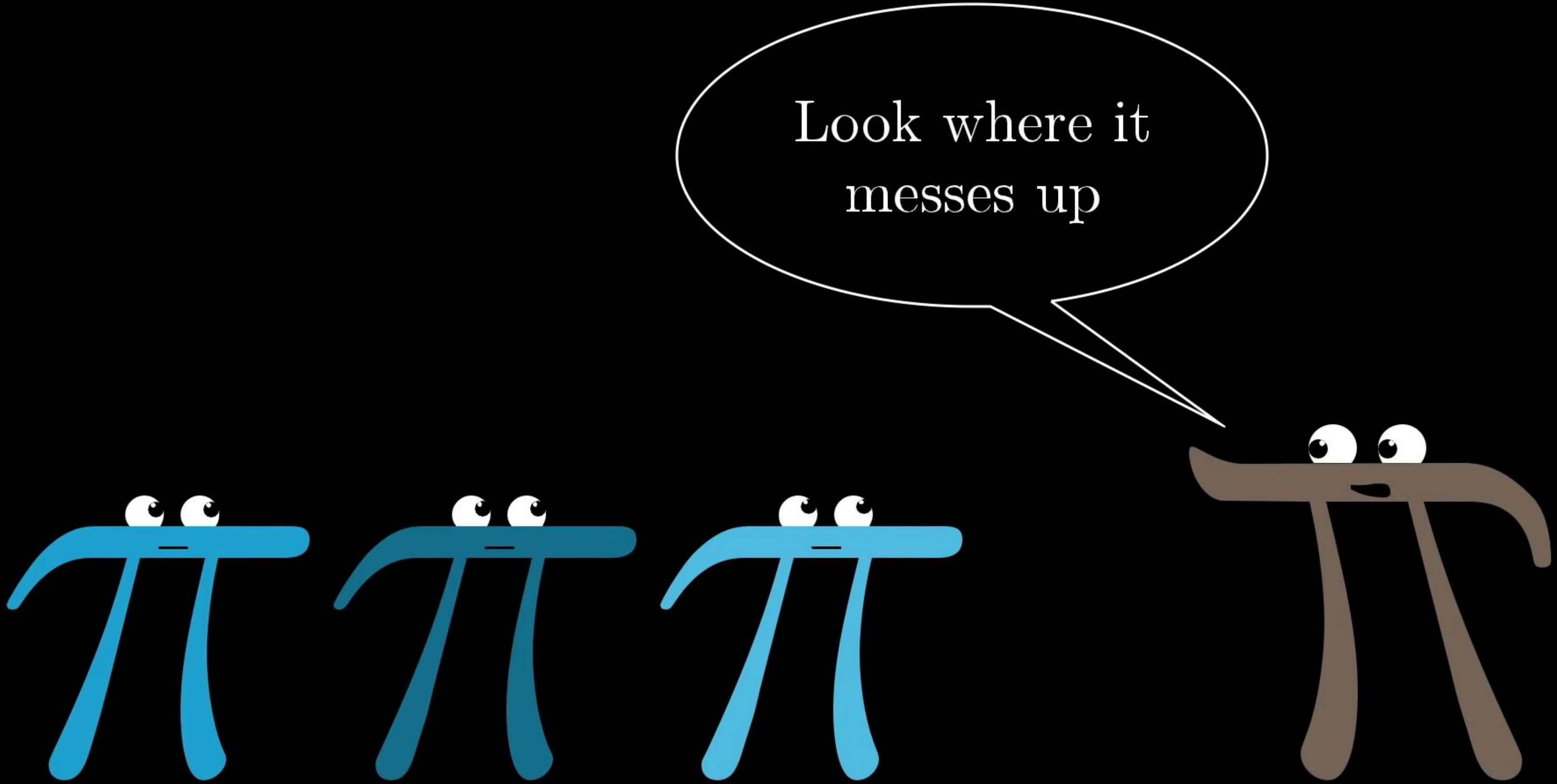
$w_{13,002}$ should increase a little



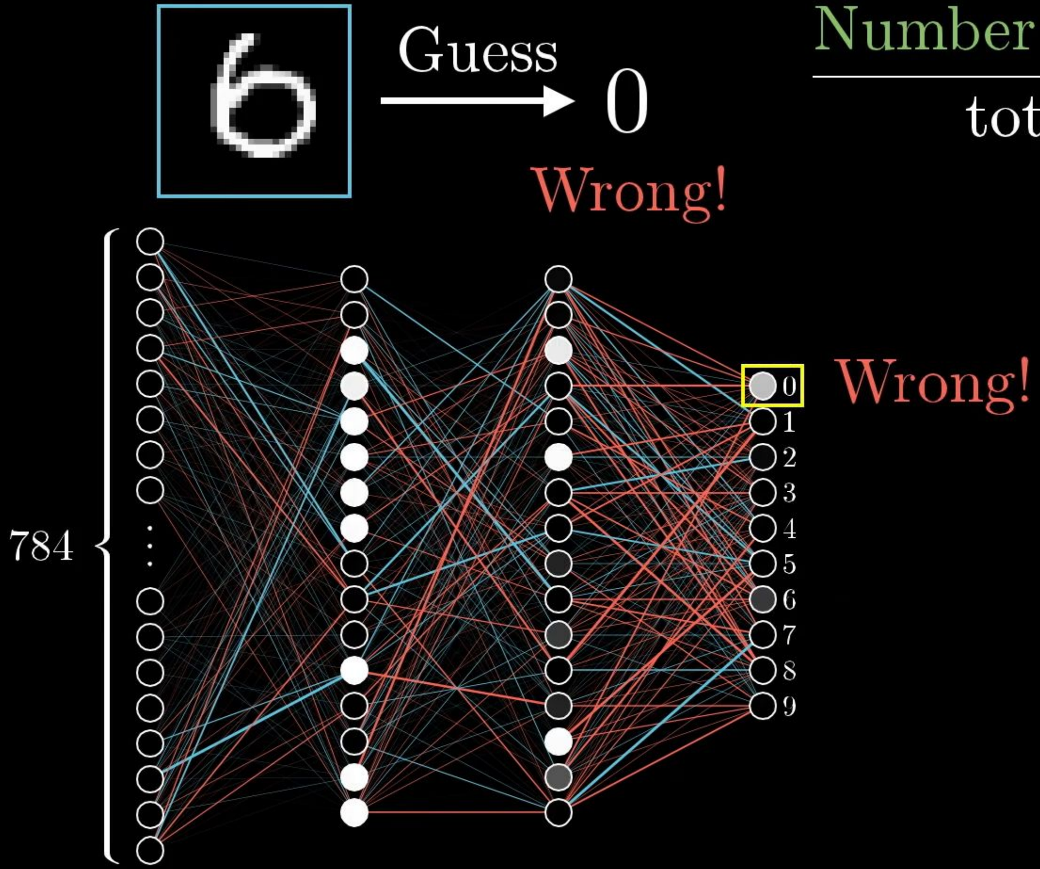
This weight matters a lot

Who even cares about this weight?


How well does it do?

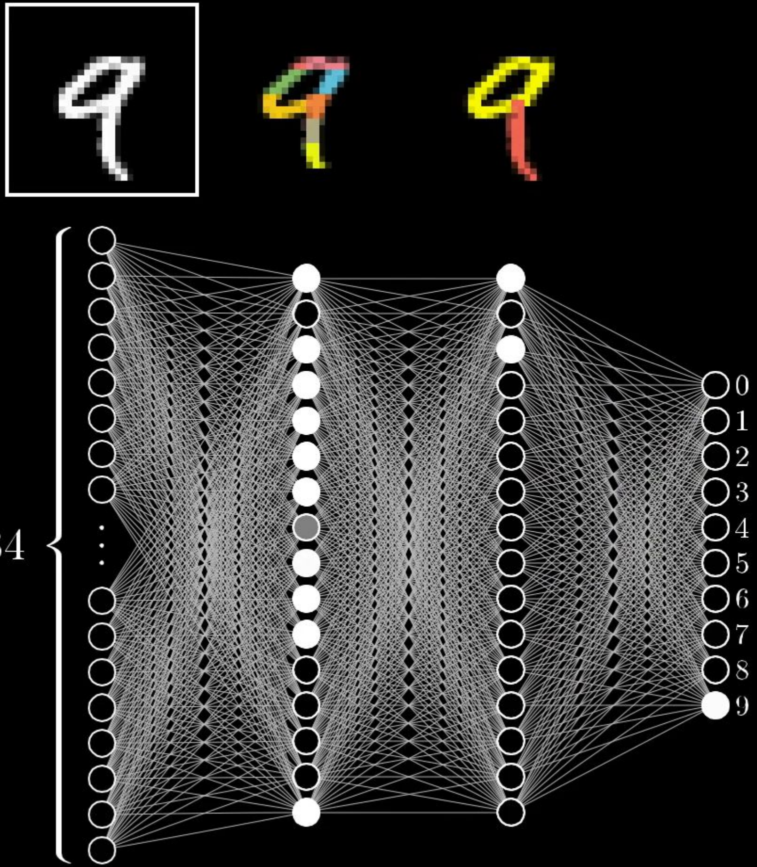

Look where it messes up

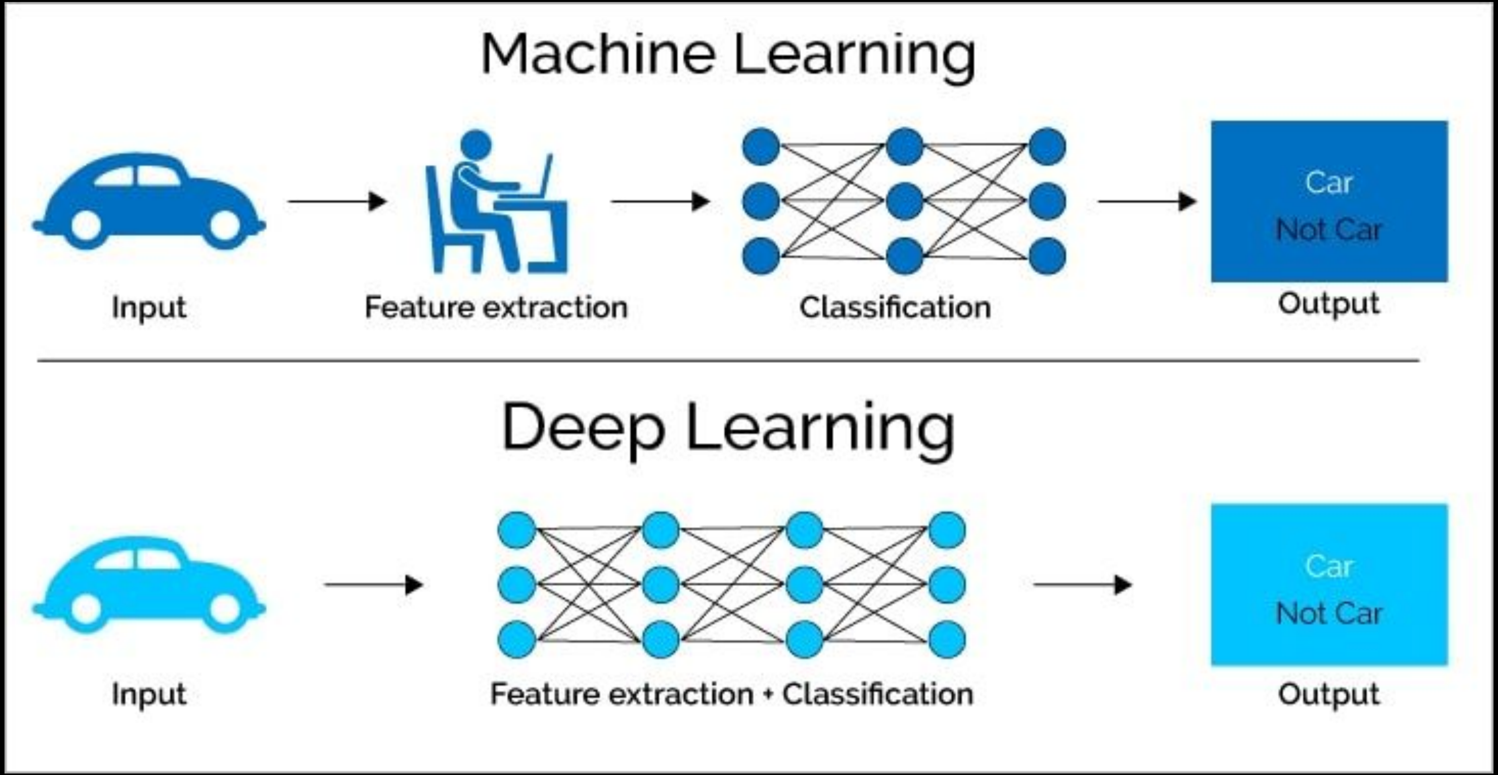
Testing data



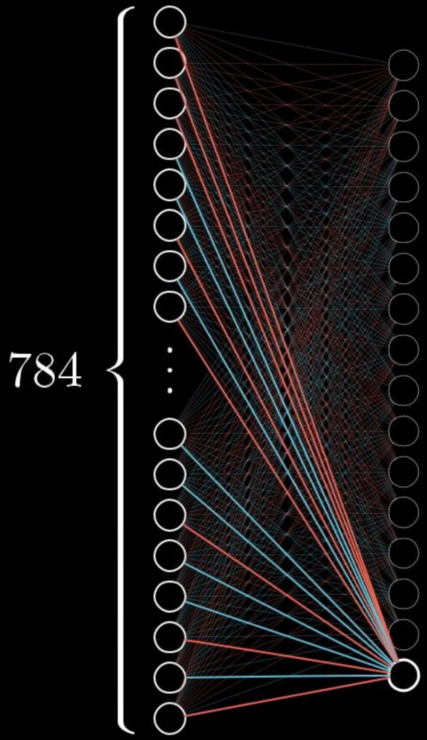
$$\frac{\text{Number correct}}{\text{total}} = \frac{248}{260} = 0.954$$

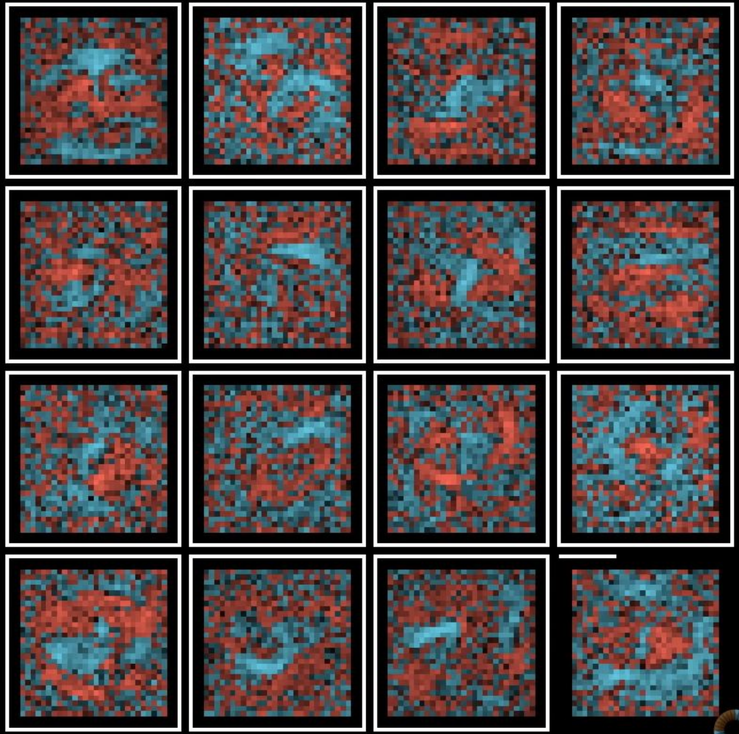
Does the network actually do this?



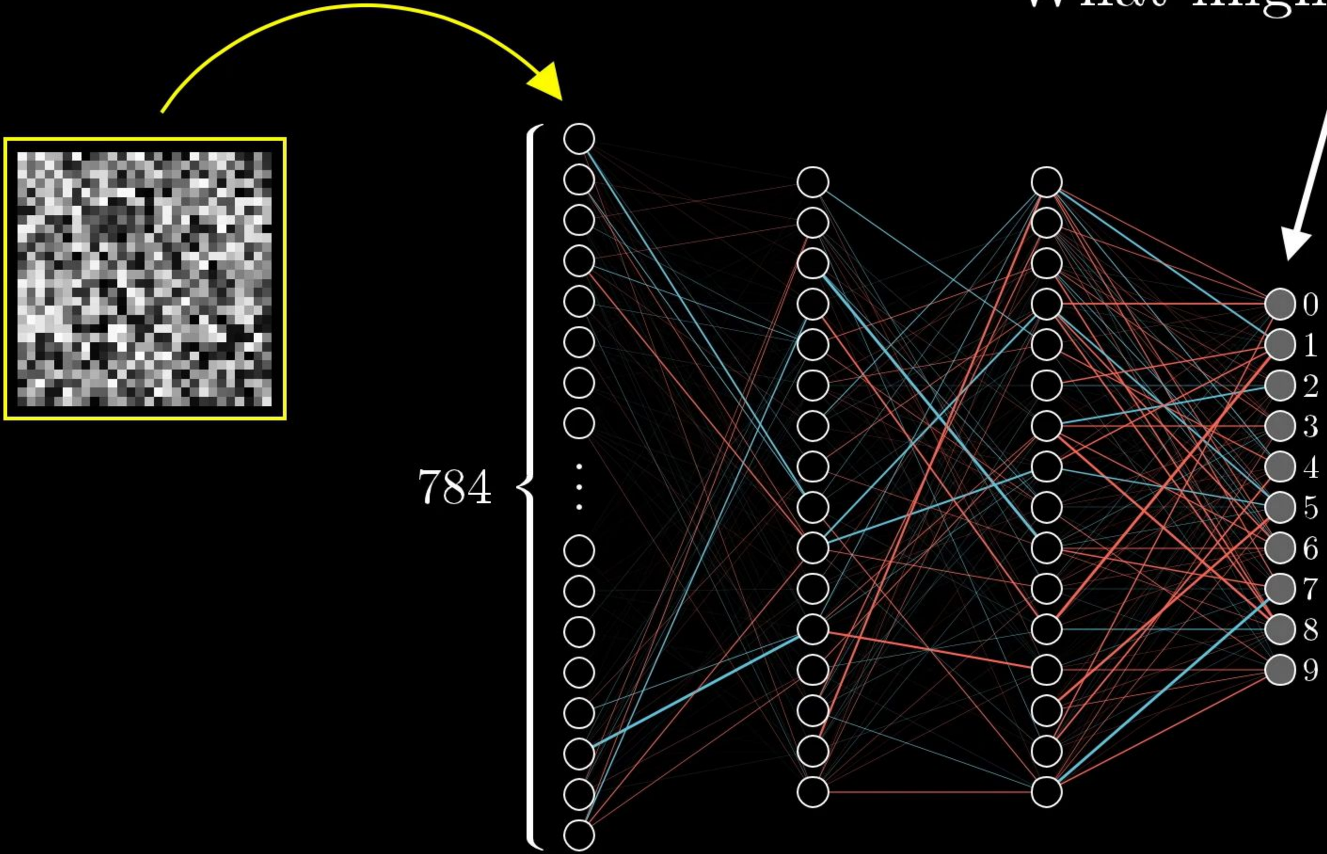

Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output

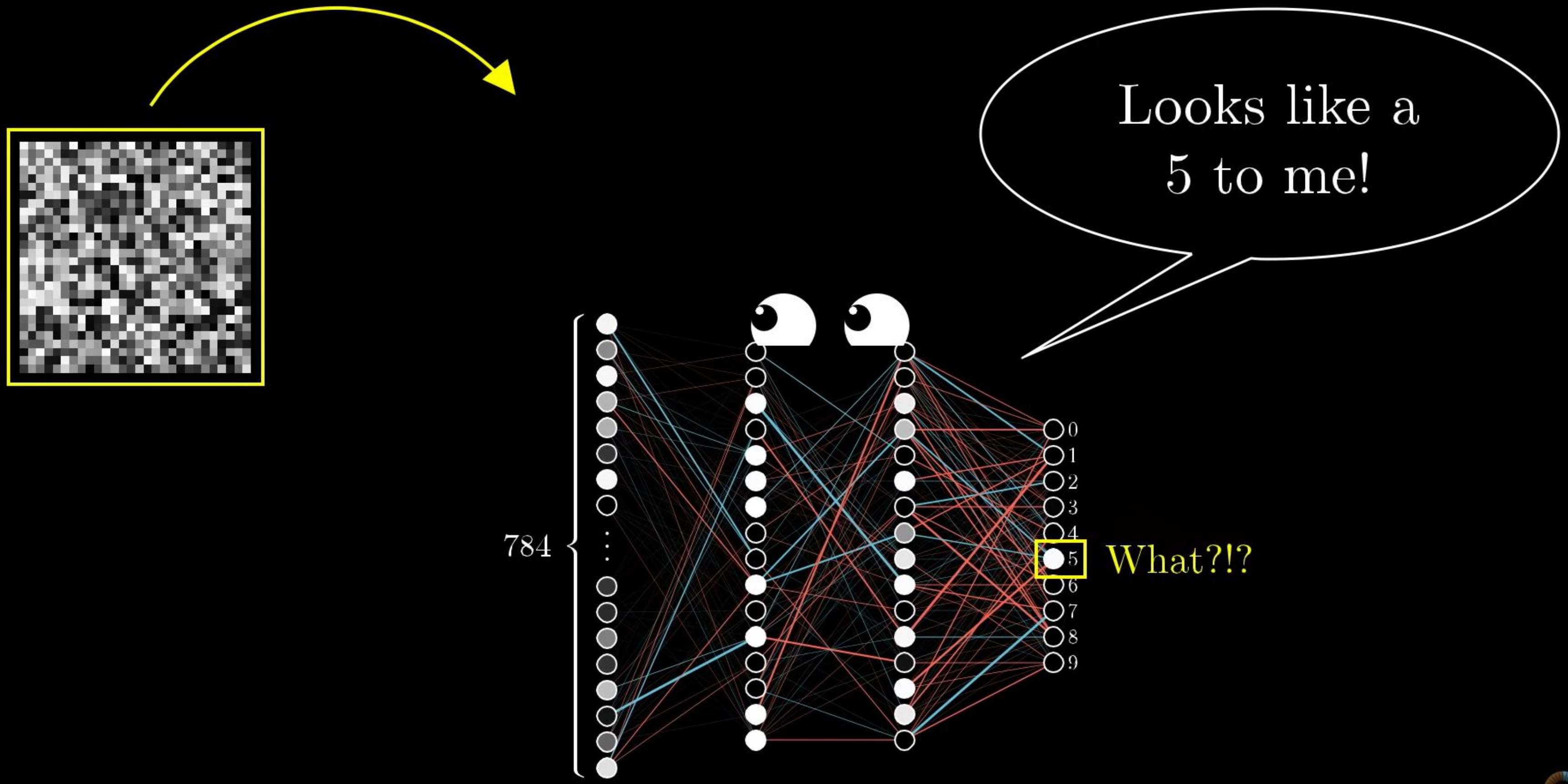
What second layer neurons look for

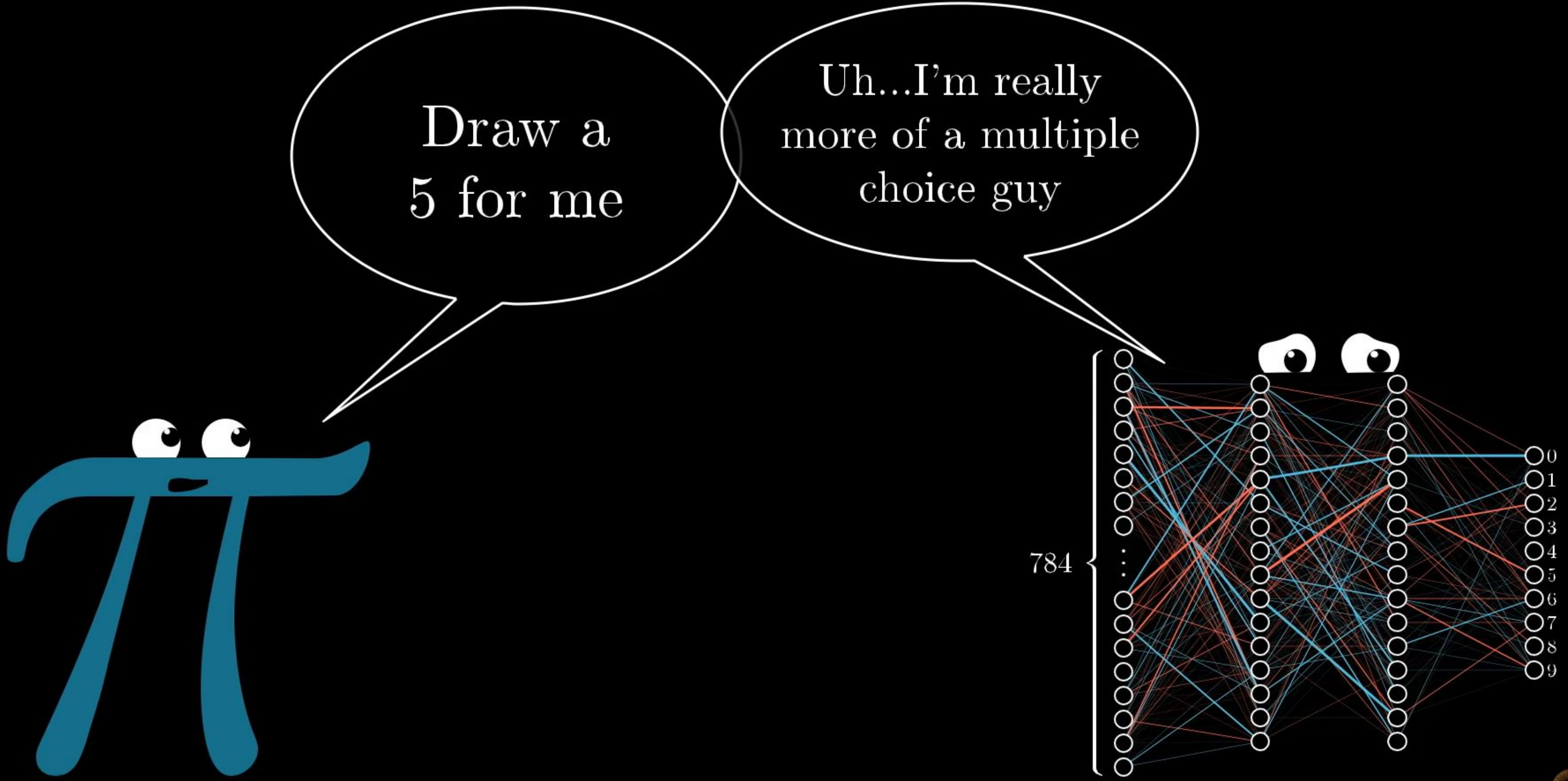




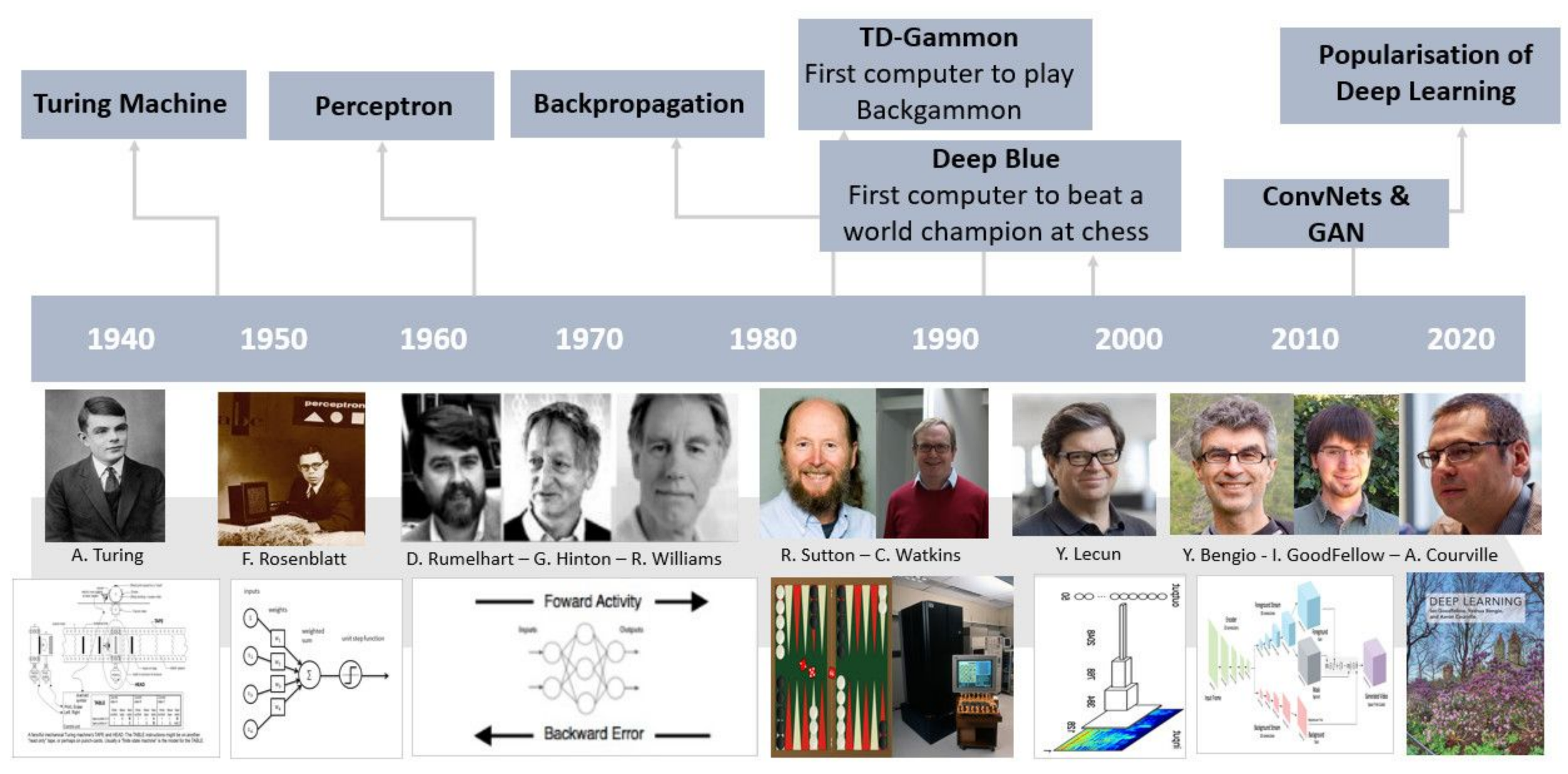
What might you expect?




Looks like a 5 to me!
784
0
1
2
3
4
5
6
7
8
9
What?!?


Draw a 5 for me
Uh...I'm really more of a multiple choice guy
784
0
1
2
3
4
5
6
7
8
9

How did it all start?



Turing Machine

Perceptron

Backpropagation

TD-Gammon
First computer to play Backgammon

Deep Blue
First computer to beat a world champion at chess

ConvNets & GAN

Popularisation of Deep Learning

1940 1950 1960 1970 1980 1990 2000 2010 2020

A. Turing

F. Rosenblatt

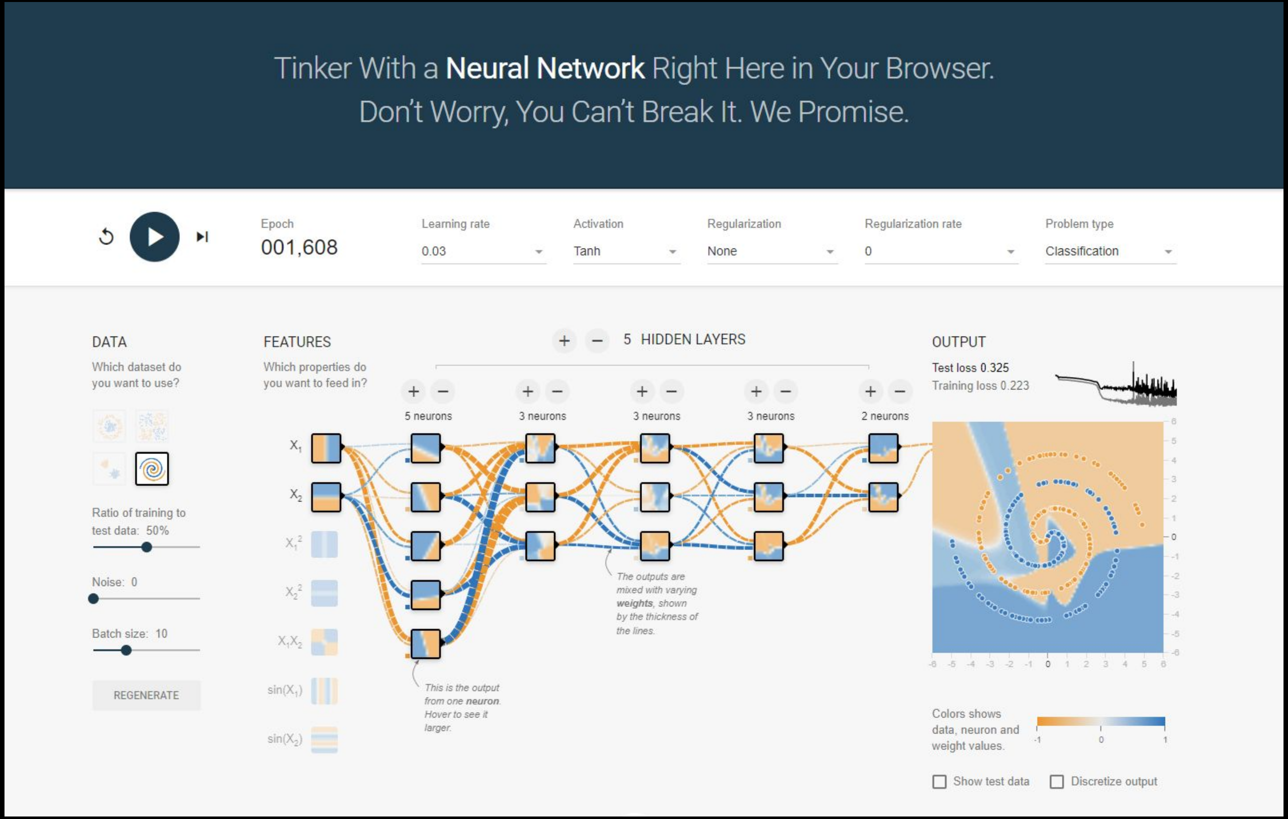
D. Rumelhart – G. Hinton – R. Williams

R. Sutton – C. Watkins

Y. Lecun

Y. Bengio - I. GoodFellow – A. Courville

Let's play!



Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

Epoch 001,608

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?

5 HIDDEN LAYERS

5 neurons · 3 neurons · 3 neurons · 3 neurons · 2 neurons

*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 0.325

Training loss 0.223

Colors shows data, neuron and weight values.

□ Show test data □ Discretize output

A recipe

1. Should I use ML on this problem? I Is there a pattern to detect? I Can I solve it analytically? I Do I have data?

2. Gather and organize data. I Preprocessing, cleaning, visualizing.

3. Establishing a baseline.

4. Choosing a model, loss, regularization, ...

5. Optimization (could be simple, could be a Phd...).

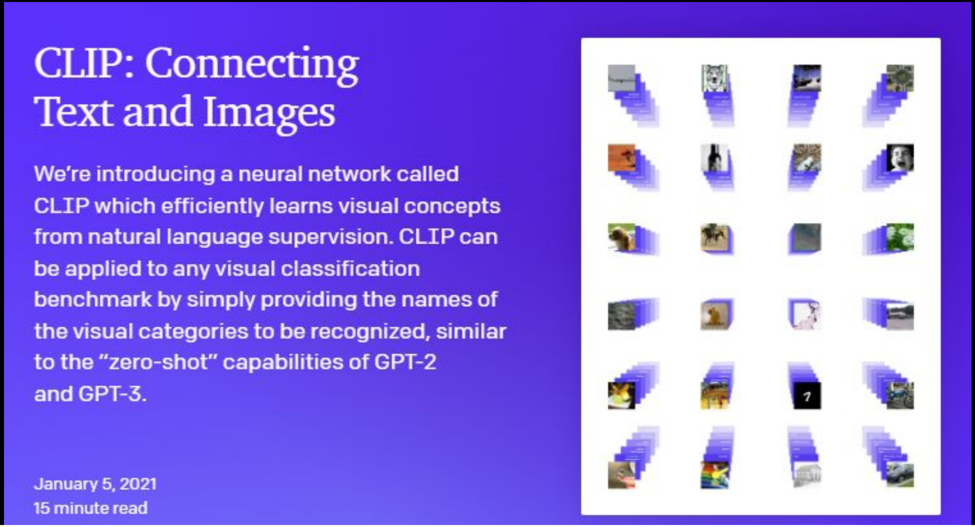
6. Hyperparameter search.

7. Analyze performance & mistakes, and iterate back to step 4 (or 2).

State-of-the-art models

What is CLIP?

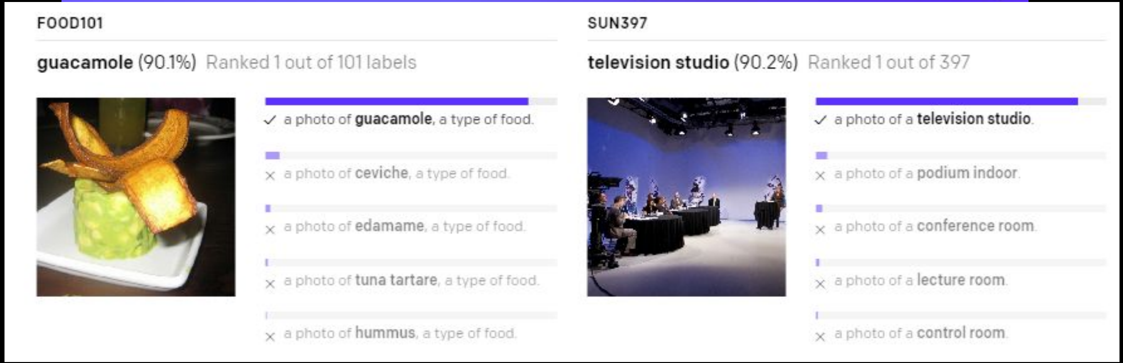
CLIP is a neural network trained on **a large set (400M) of image and text pairs**.



CLIP: Connecting Text and Images

We're introducing a neural network called CLIP which efficiently learns visual concepts from natural language supervision. CLIP can be applied to any visual classification benchmark by simply providing the names of the visual categories to be recognized, similar to the "zero-shot" capabilities of GPT-2 and GPT-3.

January 5, 2021
15 minute read



FOOD101

guacamole (90.1%) Ranked 1 out of 101 labels

✓ a photo of **guacamole**, a type of food.

× a photo of **ceviche**, a type of food.

× a photo of **edamame**, a type of food.

× a photo of **tuna tartare**, a type of food.

× a photo of **hummus**, a type of food.

SUN397

television studio (90.2%) Ranked 1 out of 397

✓ a photo of a **television studio**.

× a photo of a **podium indoor**.

× a photo of a **conference room**.

× a photo of a **lecture room**.

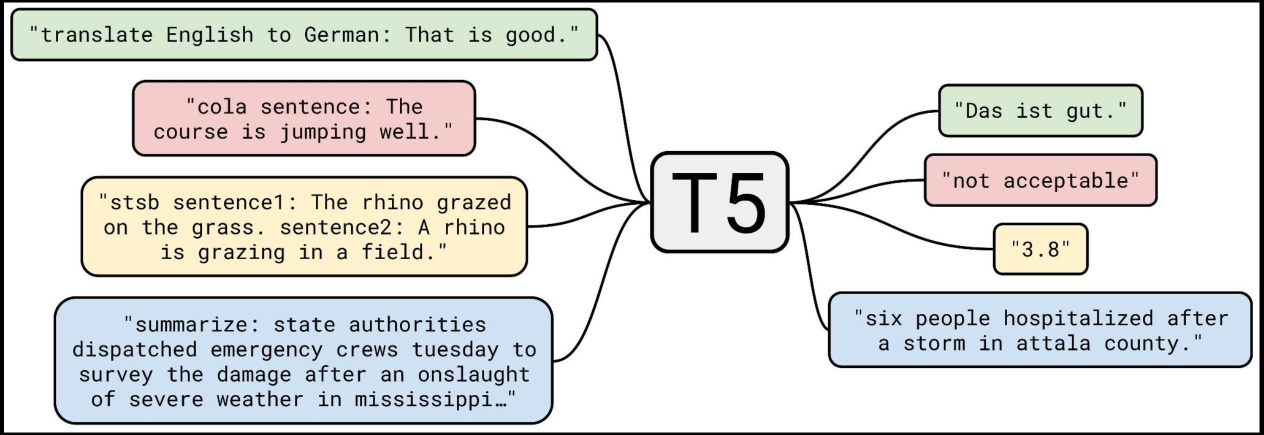
× a photo of a **control room**.

Colab notebook - CLIP

State-of-the-art models (Additional)

What is T5?

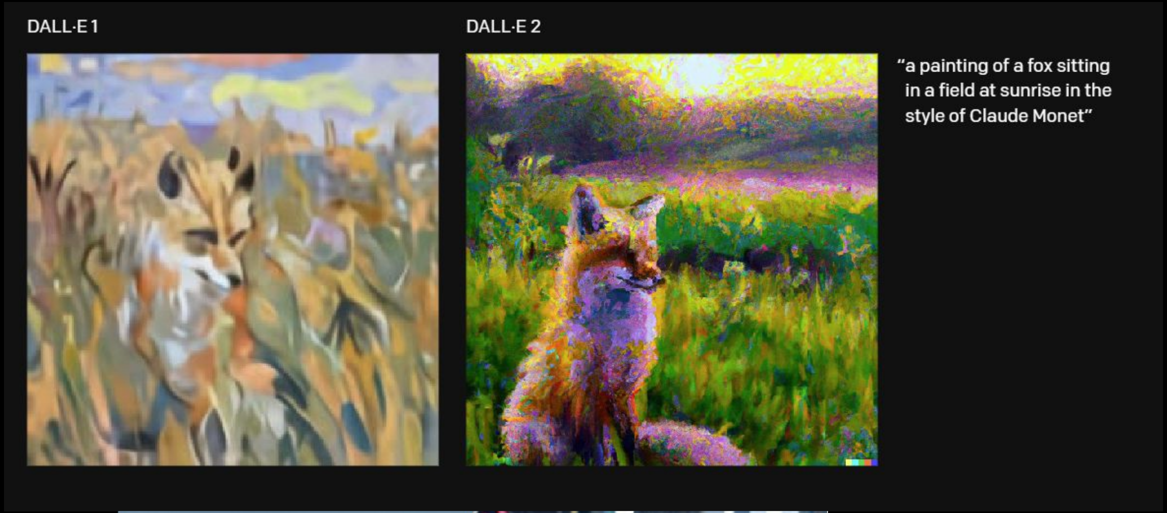
A text-to-text encoder decoder model


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Colab notebook - T5

Other cool stuff

https://openai.com/dall-e-2/



“a painting of a fox sitting in a field at sunrise in the style of Claude Monet”

https://imagen.research.google/

A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach hat.

Do you really need Machine Learning?

SPRINKLE A LITTLE
MACHINE LEARNING
ON IT

Key limitations of Machine Learning

- Ethics
- Data
- Interpretability
- Deterministic system
- Reproducibility

https://www.springboard.com/blog/data-science/when-not-to-use-ml/

Why shouldn’t I use Machine Learning

1. Data related issues - Garbage in, Garbage out - should have enough reliable data
2. Interpretability- ML models are often black box models
3. Technical debt -
4. Better alternatives - A simple solution that takes 1 week to build that is 90% accurate will **almost always** be chosen over a machine learning model that takes 3 months to build that is 95% accurate, Simpler= Better

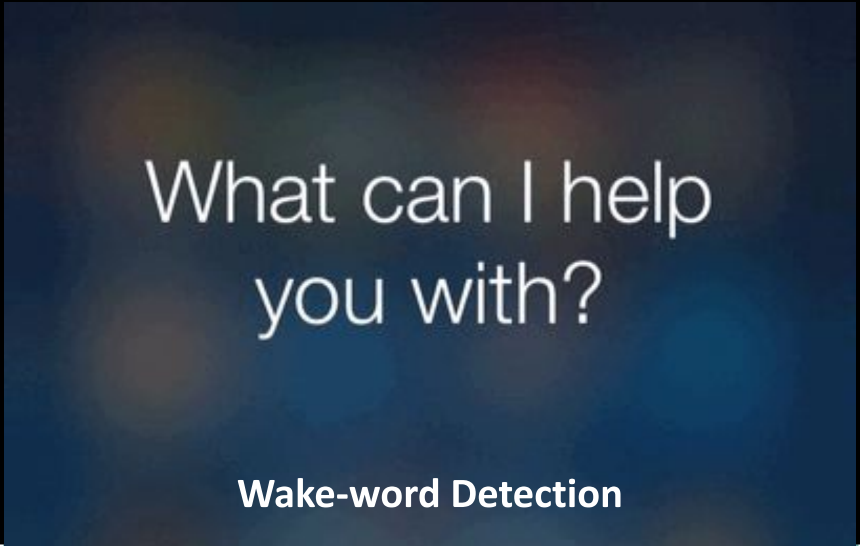
https://towardsdatascience.com/4-reasons-why-you-shouldnt-use-machine-learning-639d1d99fe11

Embedded Machine Learning for Edge Computing

An intuitive introduction to Machine Learning

Amaya Dharmasiri

Oct/2022



What can I help you with?

Wake-word Detection

Gesture Detection

https://machinelearning.apple.com/research/enabling-hand-gesture

Face/Person Detection