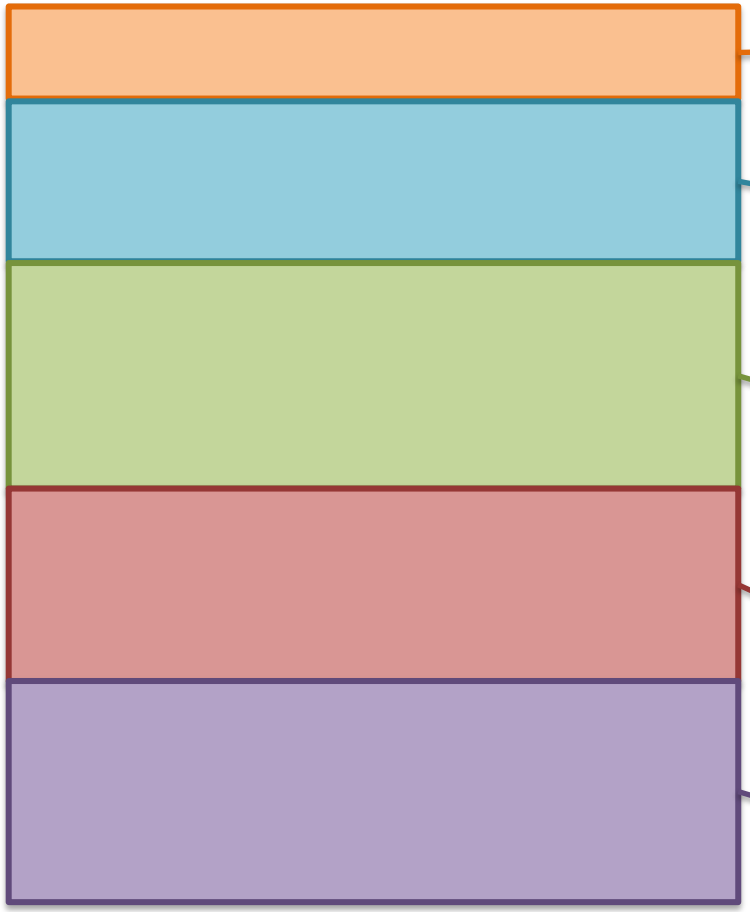
Simulating Genetic Data - Script



required python modules

Define useful functions

Define global variables

Define model-specific and distribution-sampled variables (priors) within a loop

Add sampled values to ms command

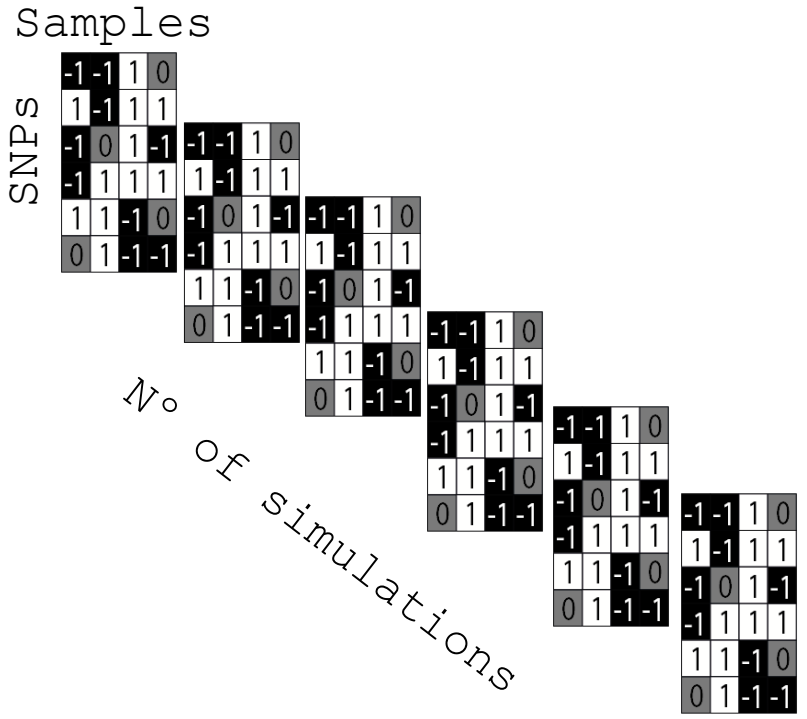
Save simulations and parameters

Simulating Genetic Data - Script

Expected outputs:

Parameters

	Theta	T1	T2	T3	Ne
Sim1					
Sim2					
Sim3					
Sim4					



Simulating Genetic Data - Script

Examples of ms commands from the script

- Scenario 1:

ms 270 1000 -s 1 -t 0.280081 -I 2 160 110 -n 2 1.875624 -en 0.001289 1 35.544203 -en 0.001289 2 79.531376 -em 0.001289 1 2 1.263896 -em 0.001289 2 1 1.919980 -eg 0.017061 1 38.360445 -eg 0.017061 2 43.970864 -em 0.017061 1 2 0.301733 -em 0.017061 2 1 3.967323 -eg 0.127240 1 0 -eg 0.127240 2 0 -ej 0.293251 1 2

- Scenario 2:

ms 270 1000 -s 1 -t 0.197474 -I 2 160 110 -n 2 1.881221 -en 0.000138 2 13.295751 -em 0.000138 1 2 3.076617 -em 0.000138 2 1 3.641901 -eg 0.010041 2 23.017965 -em 0.010041 1 2 2.368179 -em 0.010041 2 1 0.699033 -eg 0.130037 2 0 -ej 0.756109 1 2

Scenario 3:

ms 270 1000 -s 1 -t 0.388039 -I 2 160 110 -n 2 1.378919 -em 0.000606 1 2 3.543538 -em 0.000606 2 1 1.884716 -em 0.006555 1 2 1.994695 -em 0.006555 2 1 3.124491 -ej 0.237054 1 2

Simulating Genetic Data - Script

- **Practical Exercise 3:**

- Copy the commands in PopPlanner and try to visualize what is being simulated (you can change the Max Time if you need).

- Is it difficult to see the scenario in PopPlanner? Do you have any thoughts about why simulations show like that?

- Now visualize the segregating sites and the trees using -T >> trees.tre. Can you see a difference in the output of these simulations compared to the simulations from Exercise 2?

- Add trees from different scenarios into FigTree and see if you can see any similarities or differences.

Course goals

- Conceive and simulate genetic data under competing demographic scenarios 

- Understand deep learning background and how a CNN works

- How to use deep learning to compare demographic scenarios

- Simulate genomic regions with selective sweeps and use CNN to detect such regions on real genomes

Introduction to Supervised Machine Learning and Convolutional neural Networks (CNNs)

Program

Part I: A gentle introduction to supervised deep learning.

Credit for some slides:
Matteo Fumagalli and Flora Jay

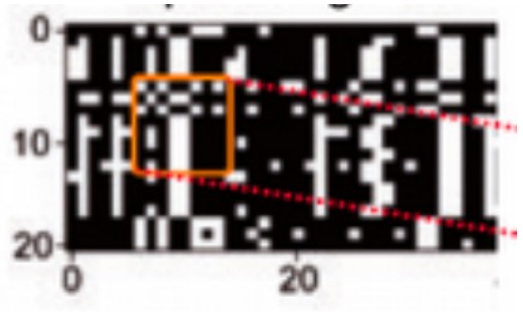
What is machine learning?

TASK:
predict y from x

Angermueller et al Mol Syst Biol. (2016) 12: 878

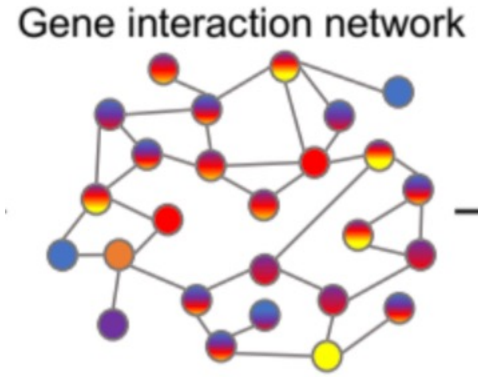
Matteo Fumagalli and Flora Jay. Machine learning and deep learning for demographic and selection inference. Lecture

What is the data?



Flagel et al. (2019) *MBE*

Images

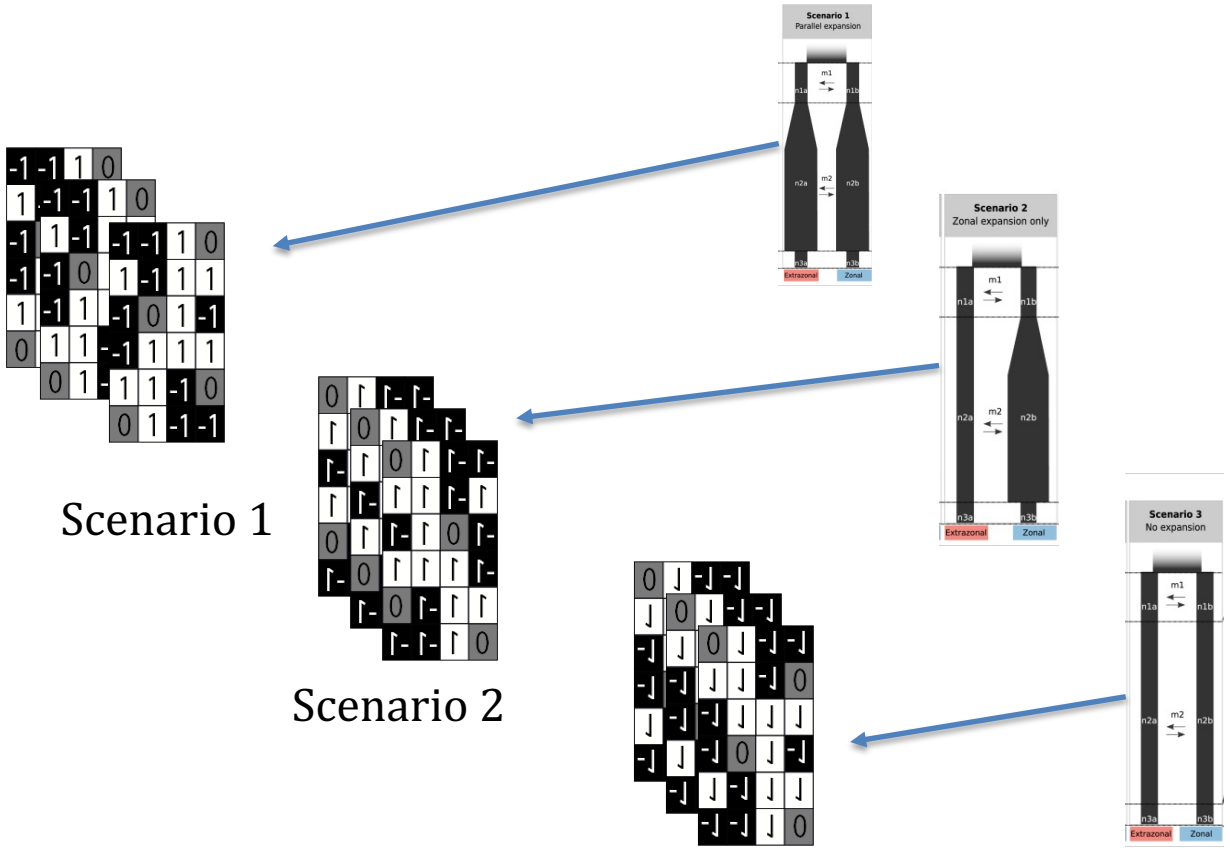


Li et al. (2022) *Nat Biom Eng*

Graphs

https://www.datarobot.com/blog/the-importance-of-machine-learning-data/

Supervised learning



Scenario 1

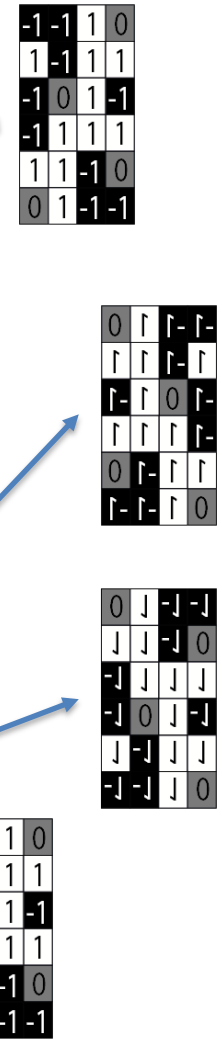
Scenario 2

Scenario 3

Data is labeled

Parameters

	Theta	T1	T2	T3	Ne
Sim1					
Sim2					
Sim3					
Sim4					



Supervised learning

- Supervised = Learn a relationship (a general model) linking input data (or features) to observed labels

Classification (predict a class)

Regression (predict a variable)

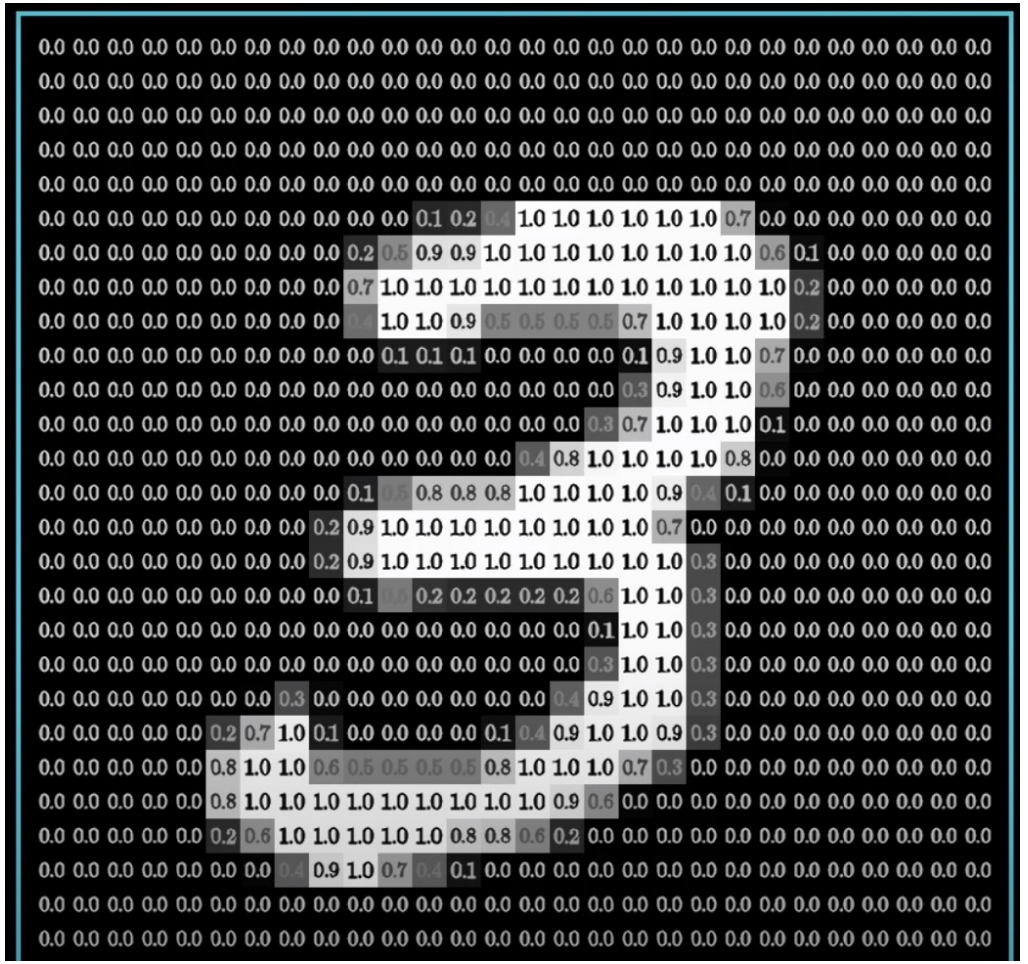
What for:
- Predict labels of new unlabeled samples (eg what's on an image?),
- Understand better the relationship between features and the label (eg understand which set of genes allow to predict a disease risk),
- ...

Matteo Fumagalli and Flora Jay. Machine learning and deep learning for demographic and selection inference. Lecture

Image recognition

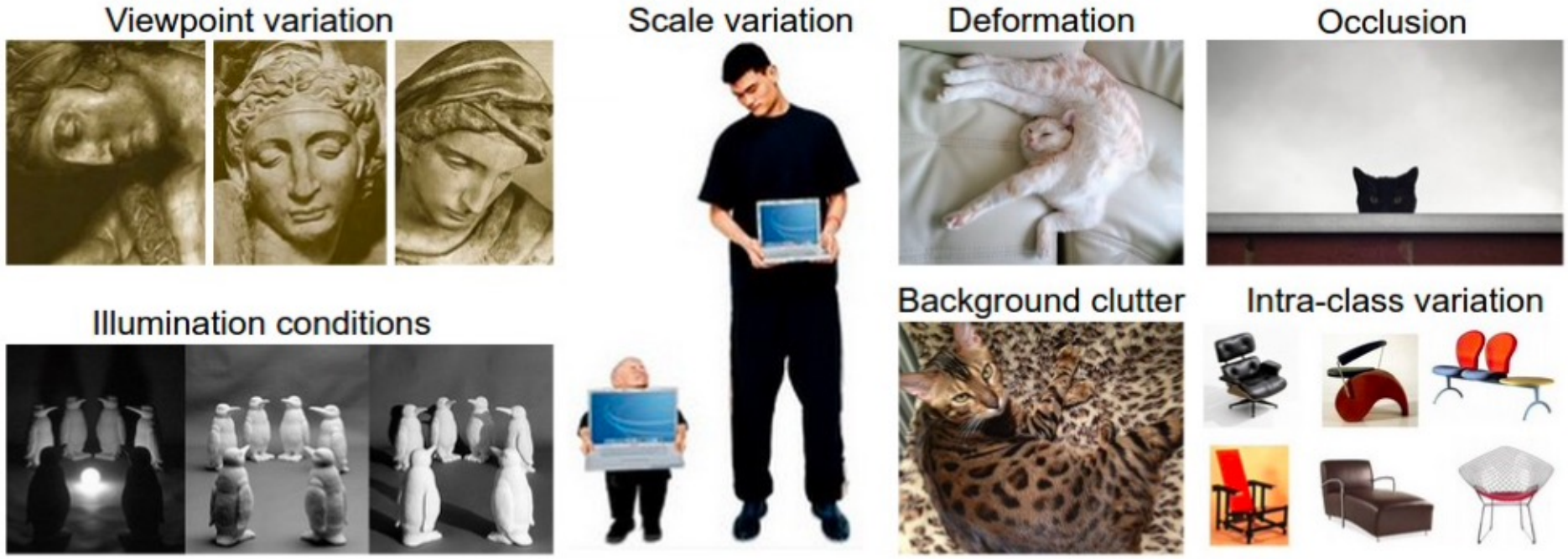
What do you see?

What the computer sees?



https://www.3blue1brown.com/lessons/neural-networks

Challenges



Matteo Fumagalli and Flora Jay. Machine learning and deep learning for demographic and selection inference. Lecture

Image recognition

Data-Driven approach (need a large training set)

The MNIST dataset (70,000 hanwritten numbers)

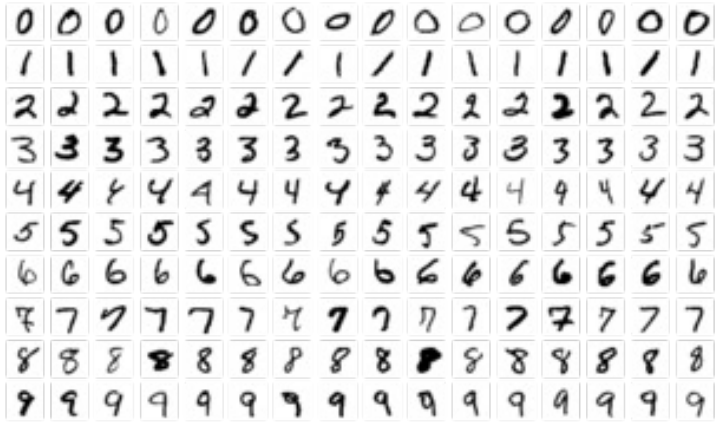
https://blog.filestack.com/api/from-mnist-classification-to-intelligent-check-processing/

Image recognition

Data-Driven approach (need a large training set)

The MNIST dataset (70,000 hanwritten numbers)

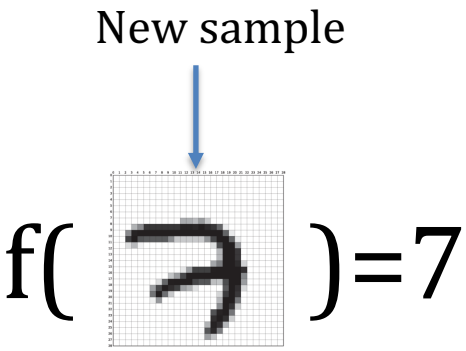
x



y

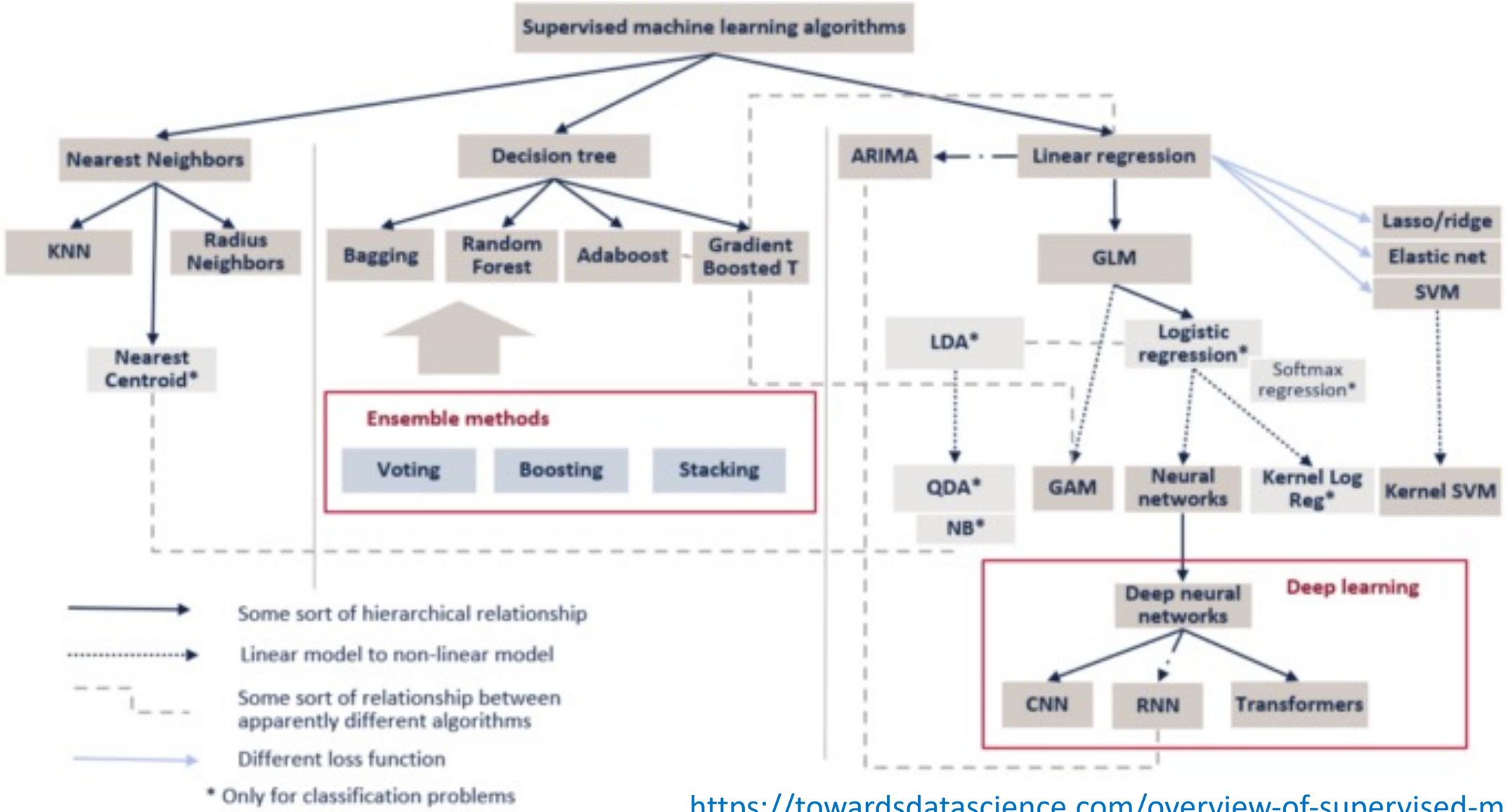
0
1
2
3
4
5
6
7
8
9

New sample



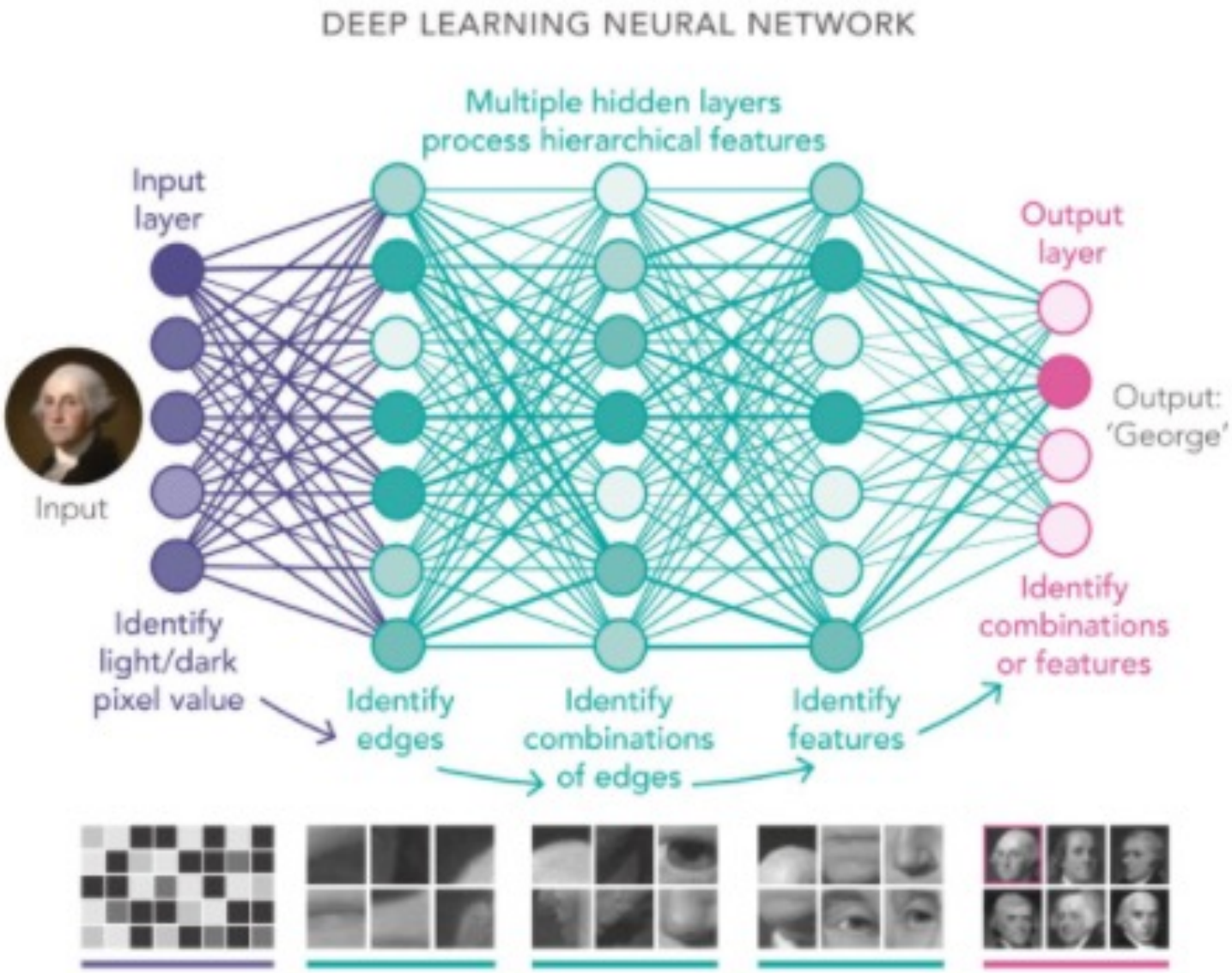
Matteo Fumagalli and Flora Jay. Machine learning and deep learning for demographic and selection inference. Lecture

Deep Neural Networks



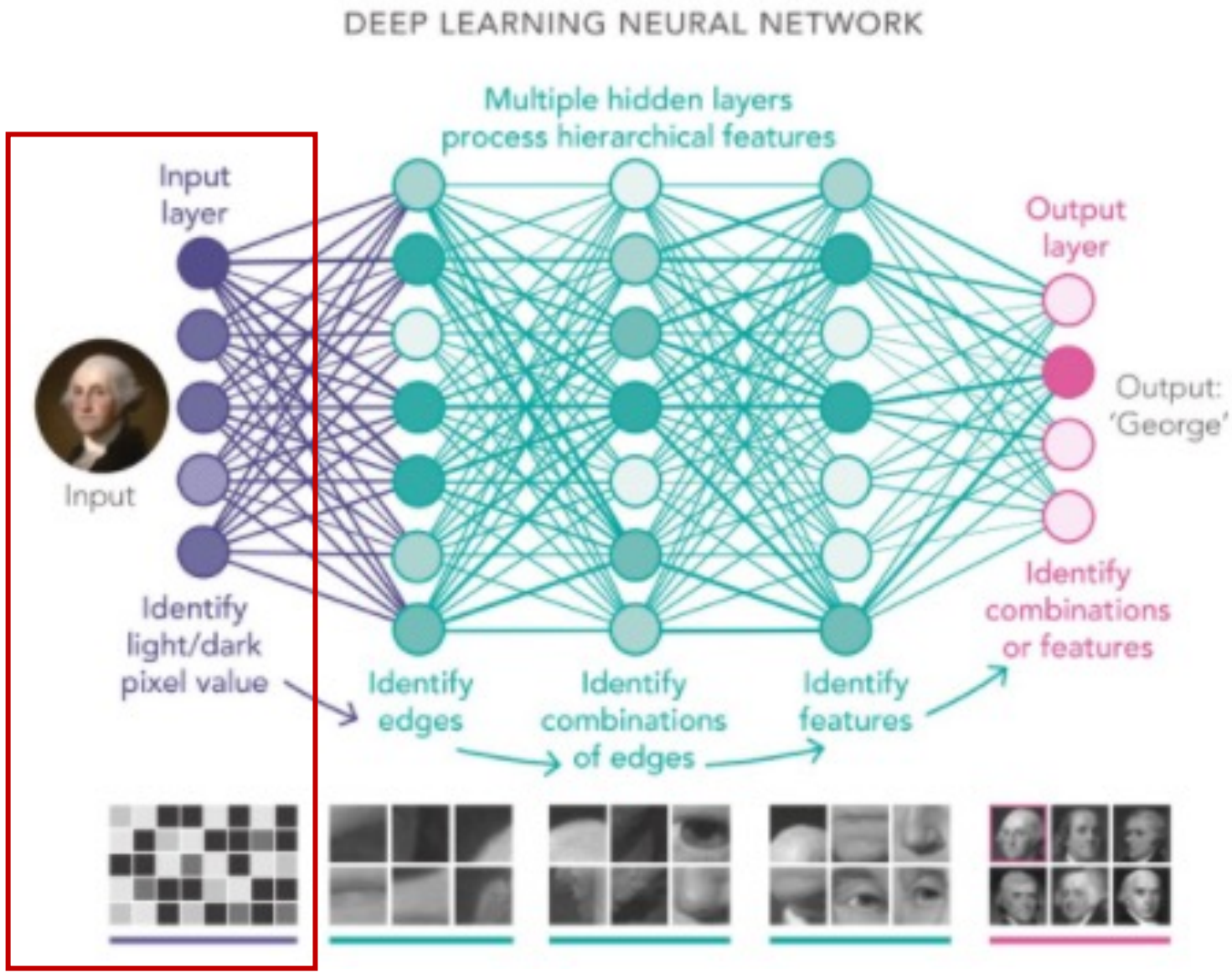
https://towardsdatascience.com/overview-of-supervised-machine-learning-algorithms-a5107d036296

Deep Neural Networks



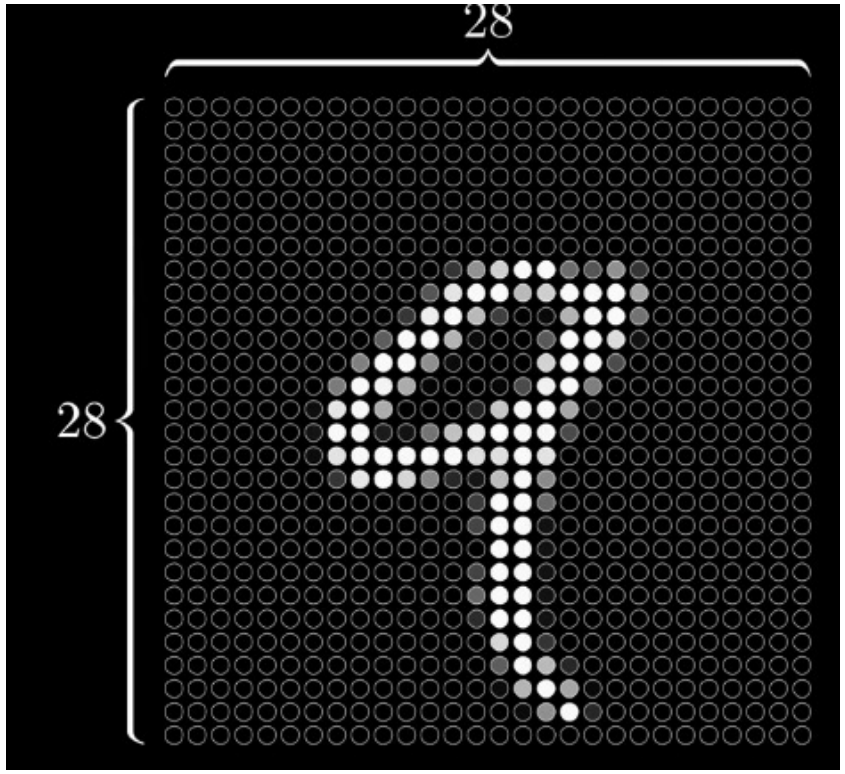
Waldrop (2019) *PNAS*

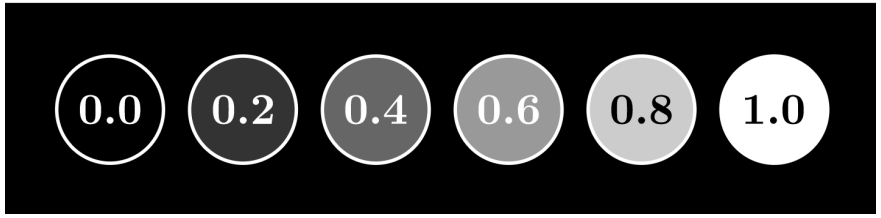
Deep Neural Networks

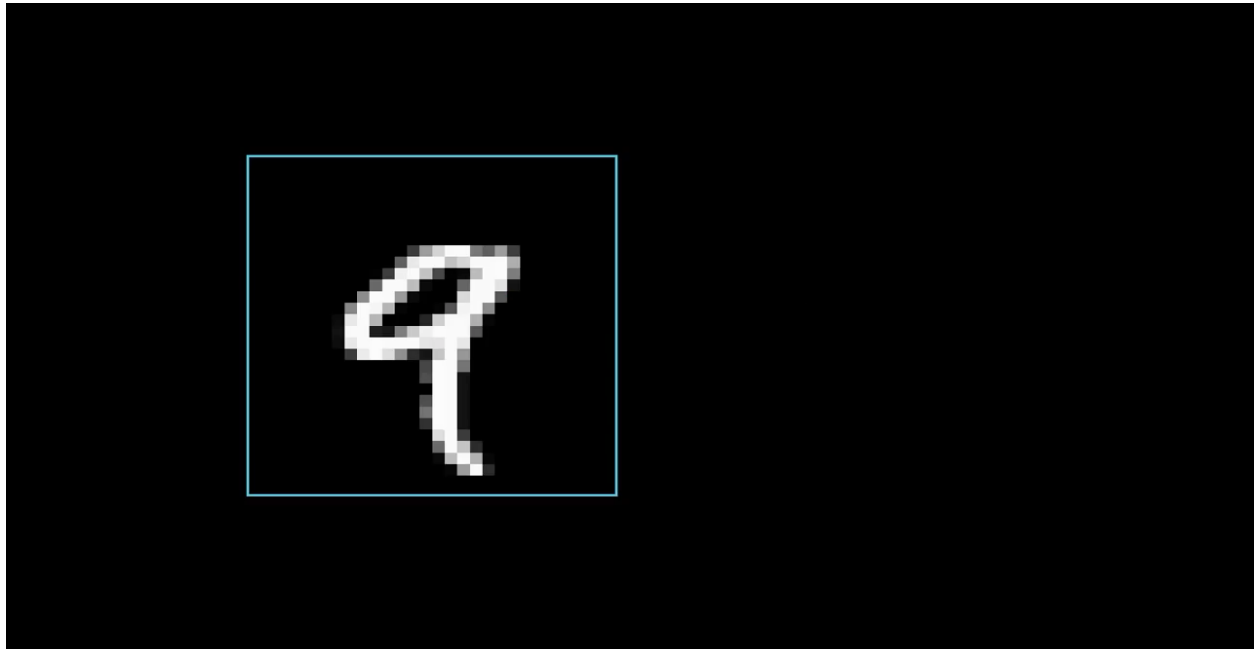


Waldrop (2019) *PNAS*

Input Layer

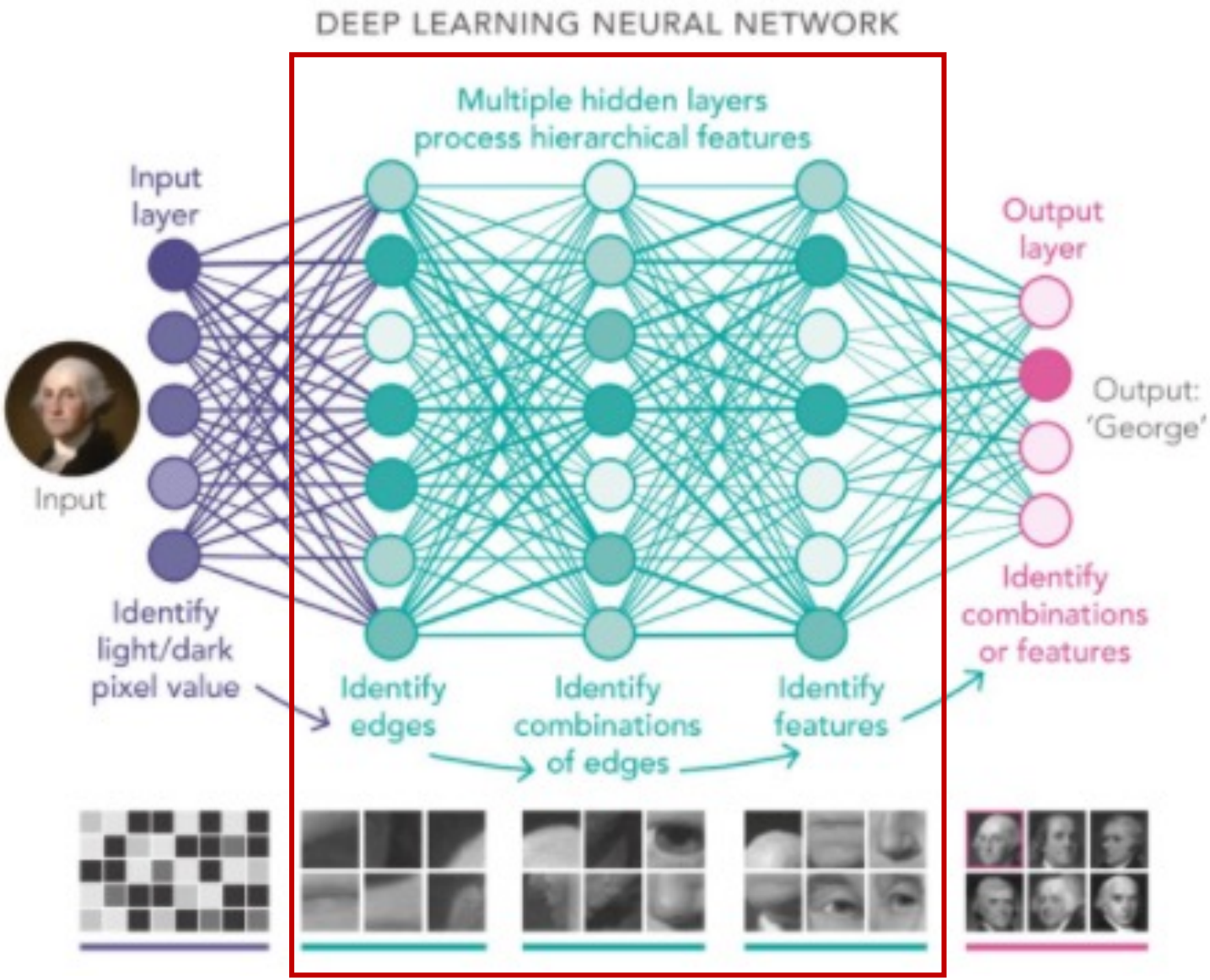






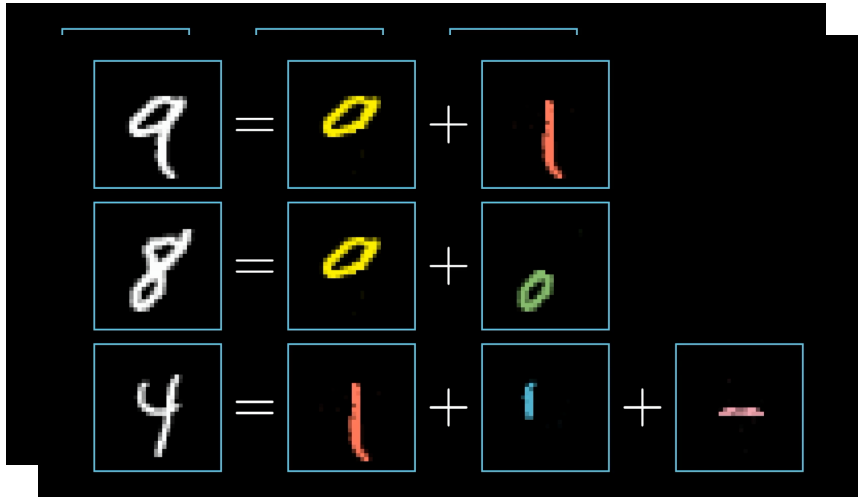
https://www.3blue1brown.com/lessons/neural-networks

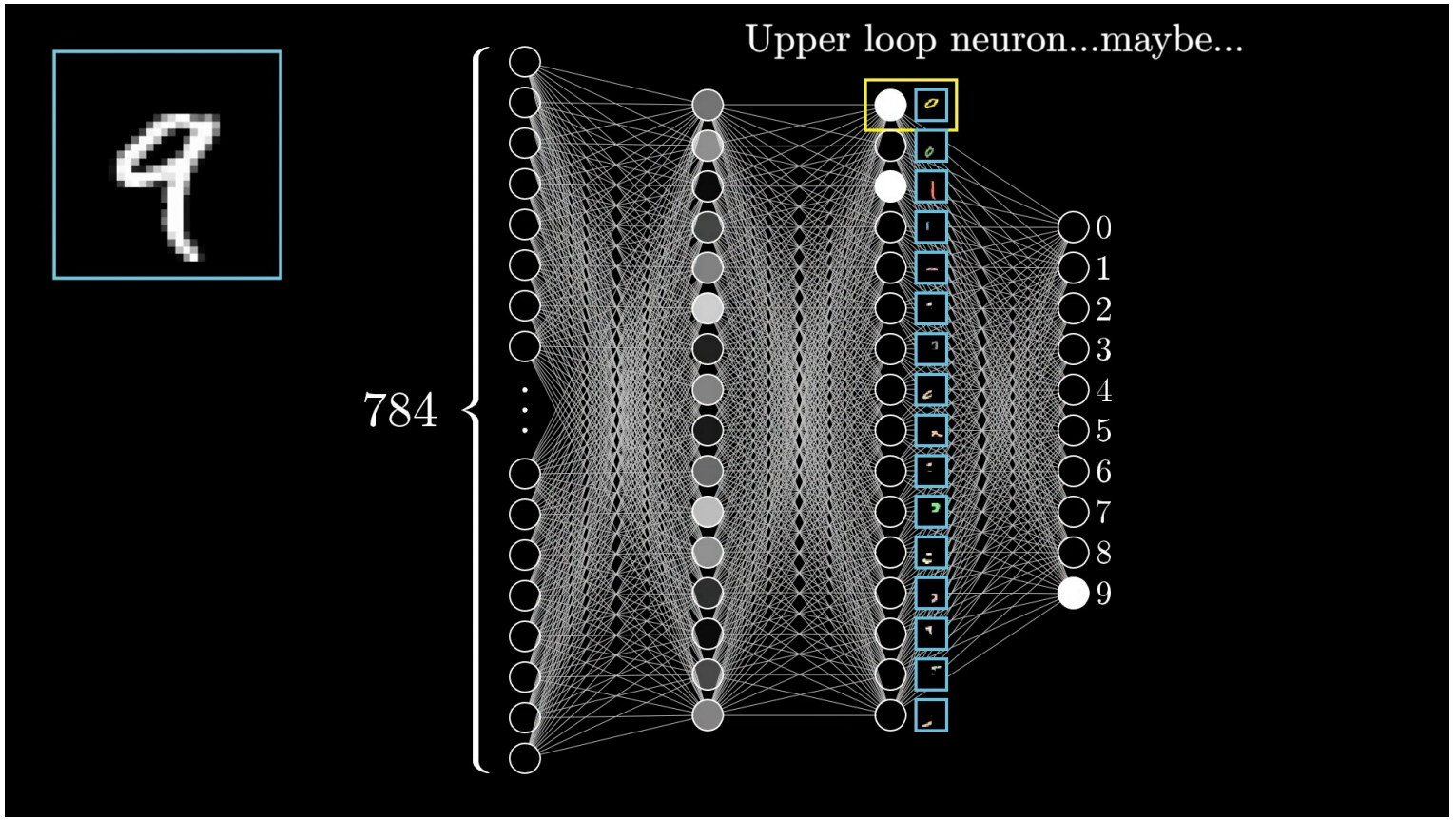
Deep Neural Networks



Waldrop (2019) *PNAS*

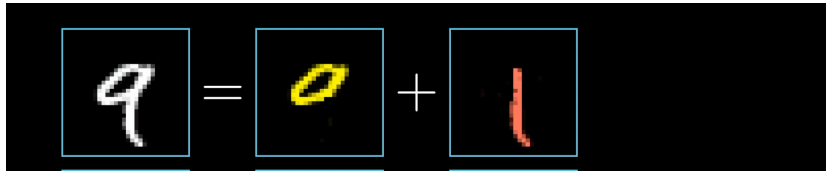
Hidden Layers process hierarchical features

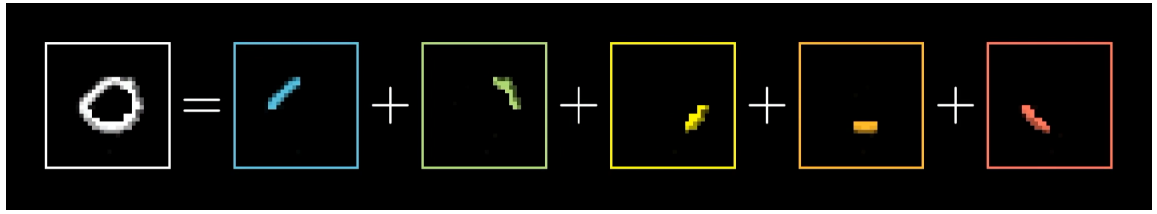


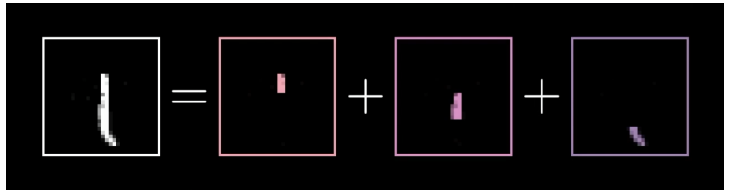


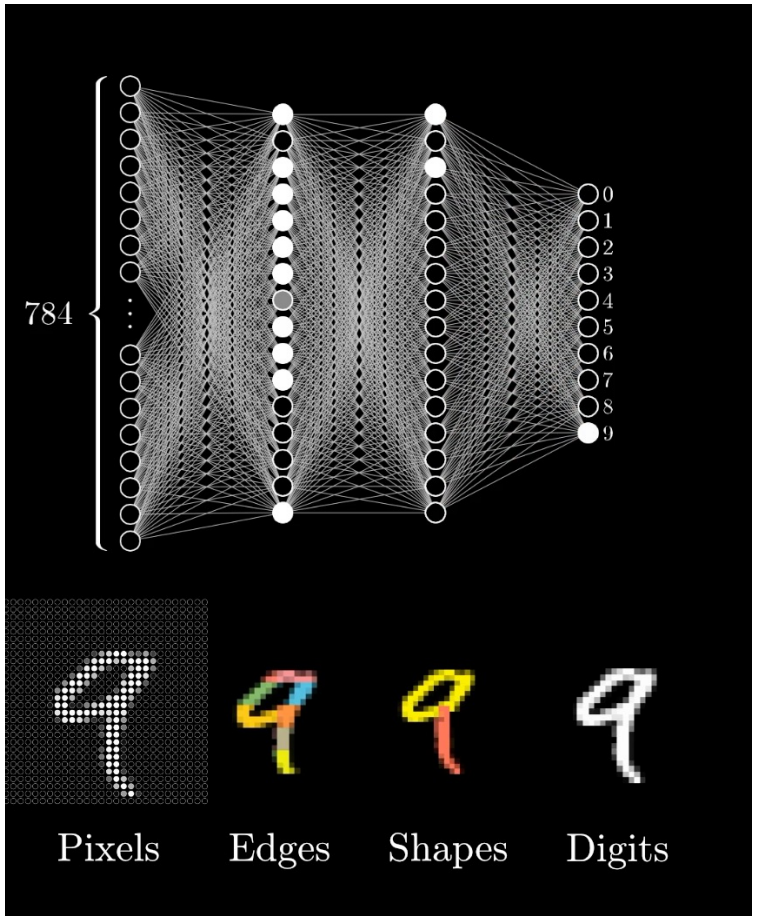
https://www.3blue1brown.com/lessons/neural-networks

Hidden Layers process hierarchical features



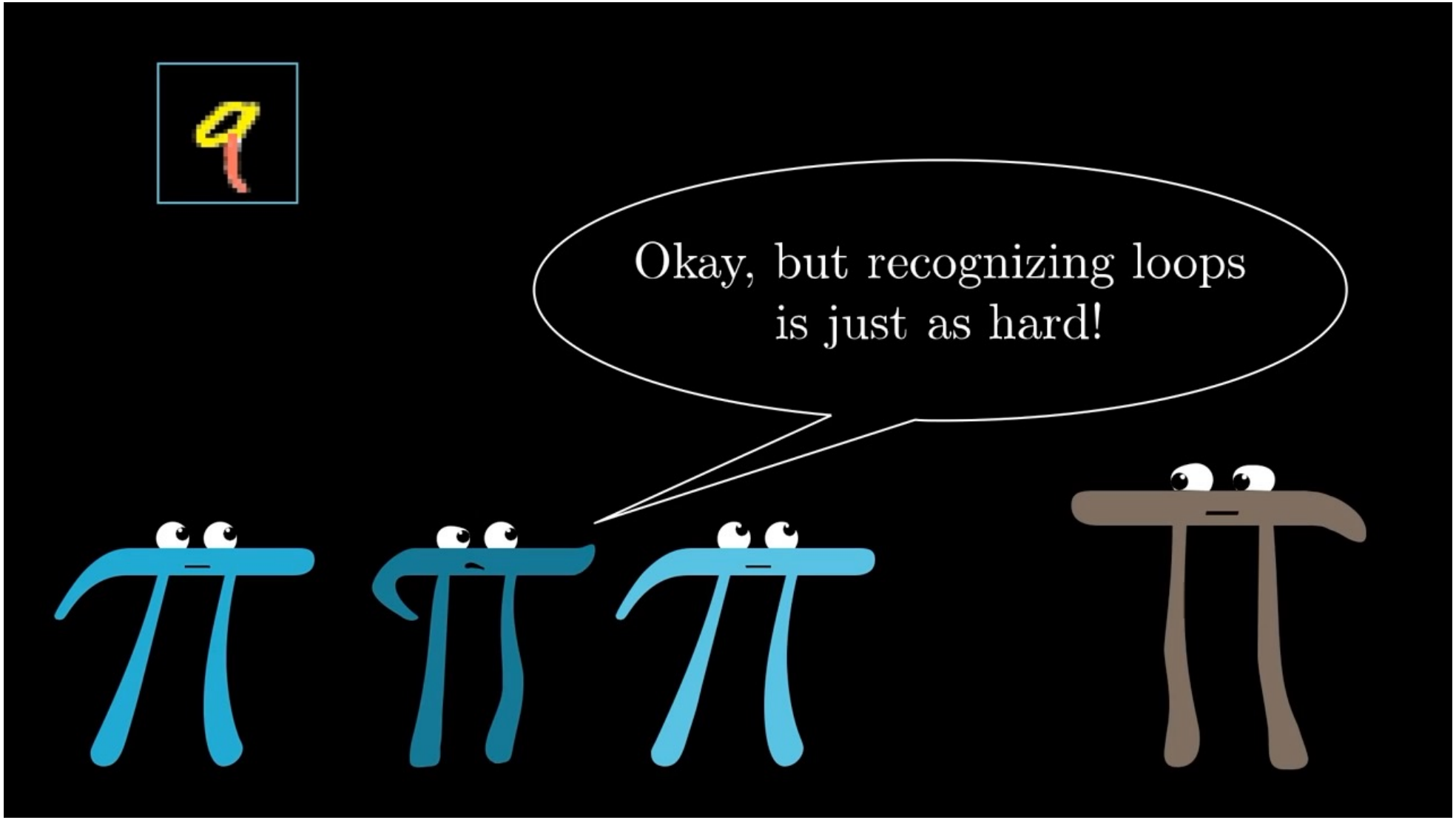






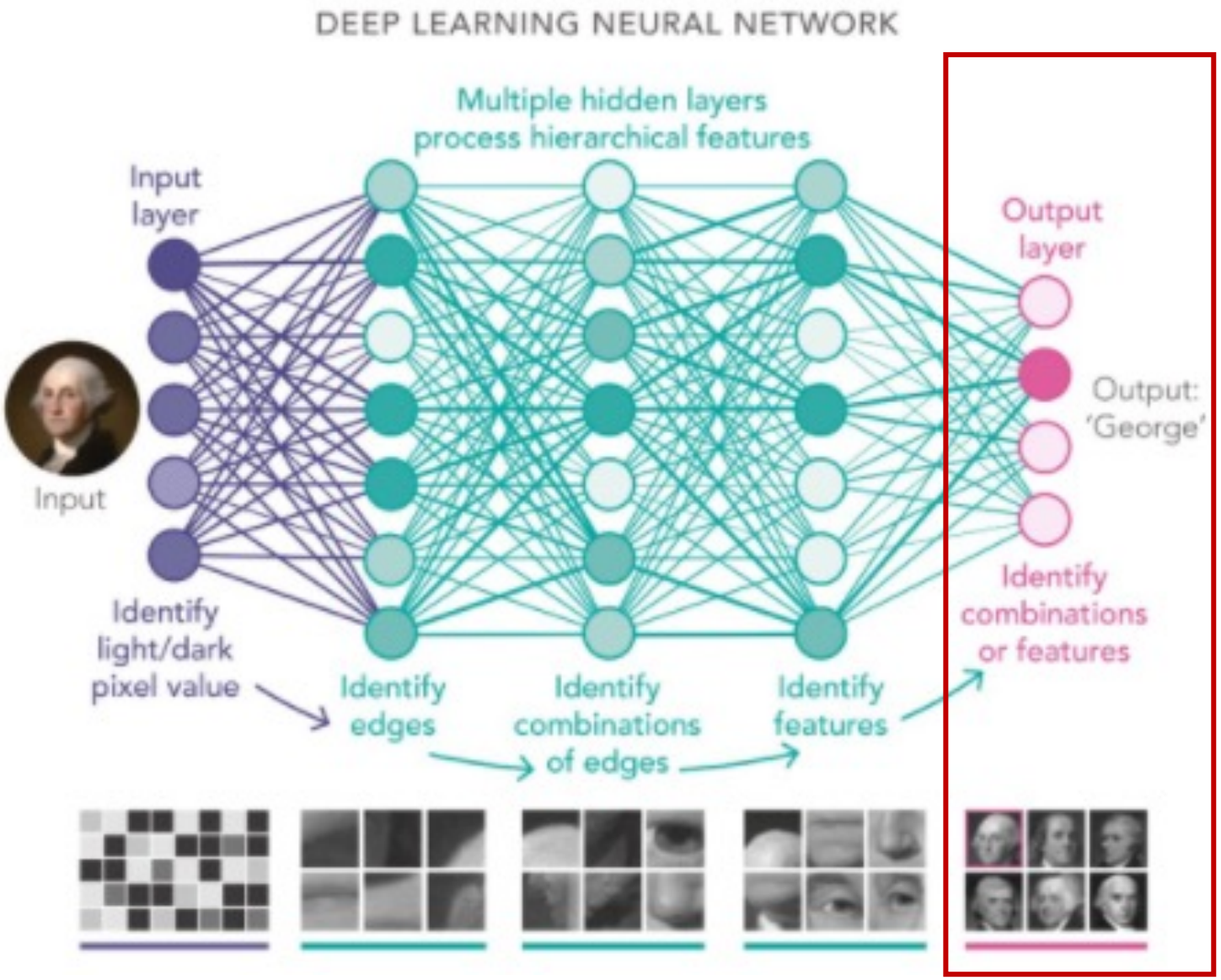
https://www.3blue1brown.com/lessons/neural-networks

Hidden Layers process hierarchical features



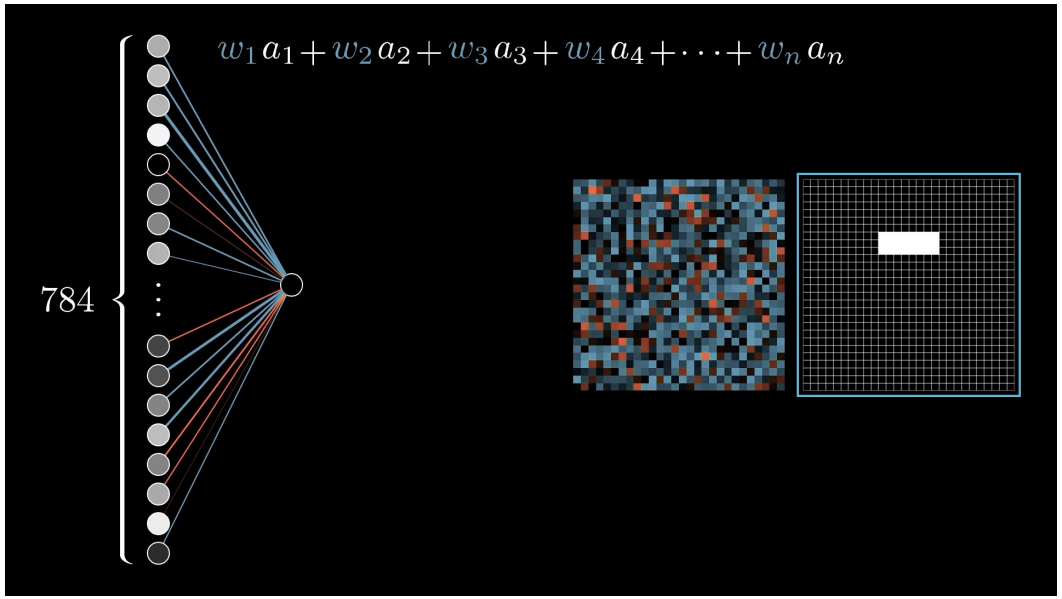
https://www.3blue1brown.com/lessons/neural-networks

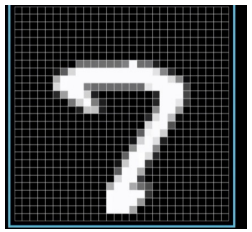
Deep Neural Networks



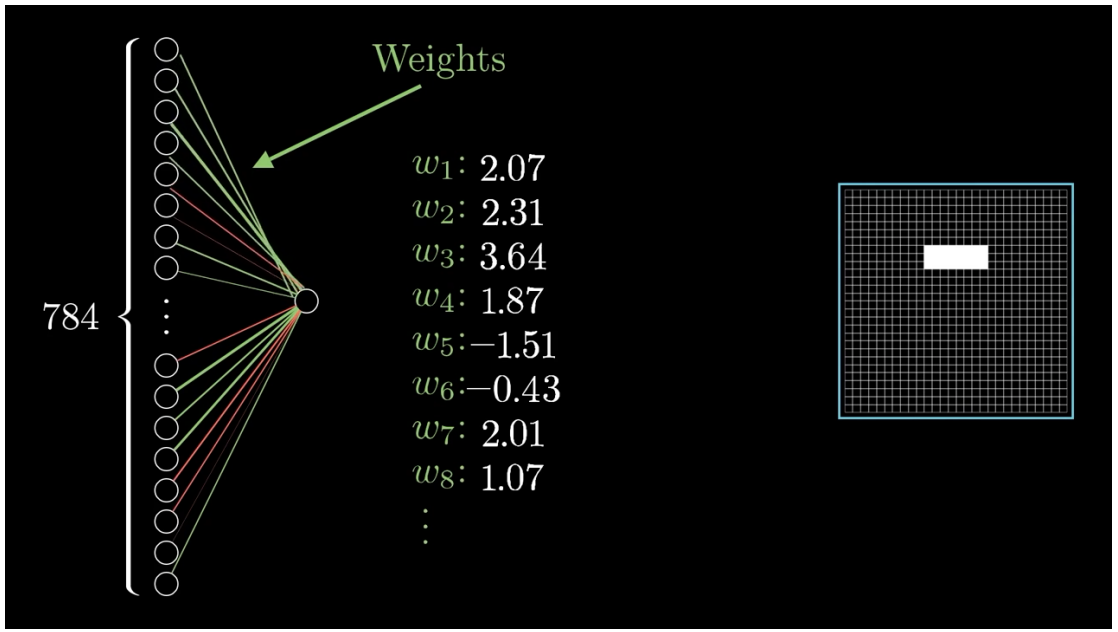
Waldrop (2019) *PNAS*

Network Weights



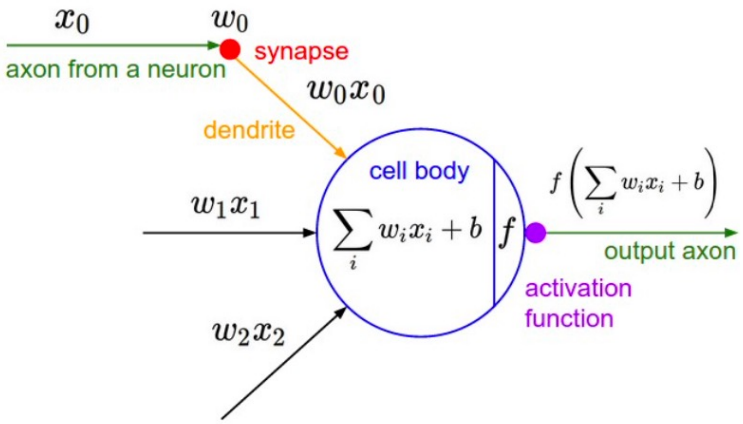


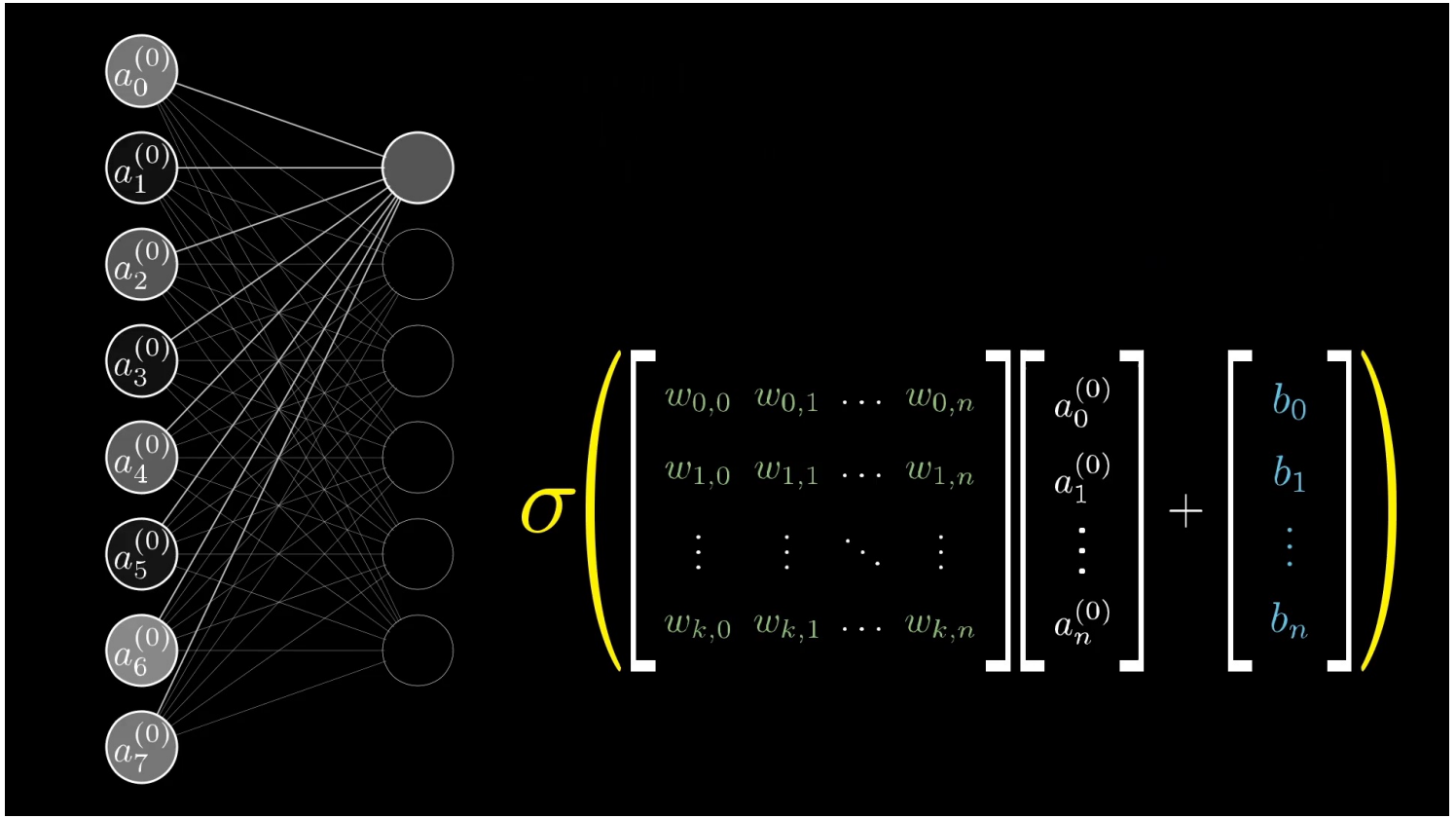
Neuron in the secon layer to check whether the image contains this one, specific edge.



https://www.3blue1brown.com/lessons/neural-networks

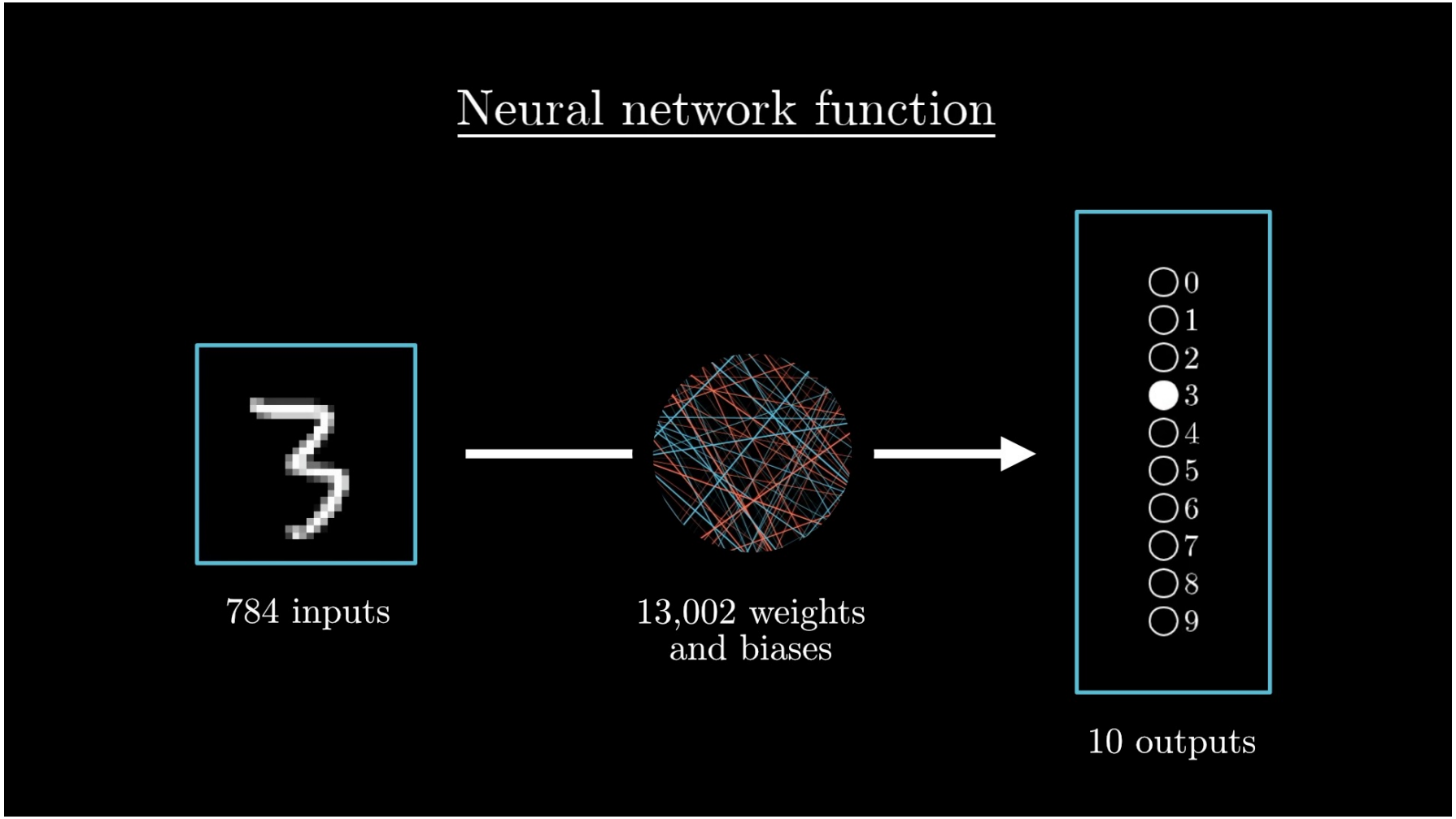
Network Weights





https://www.3blue1brown.com/lessons/neural-networks

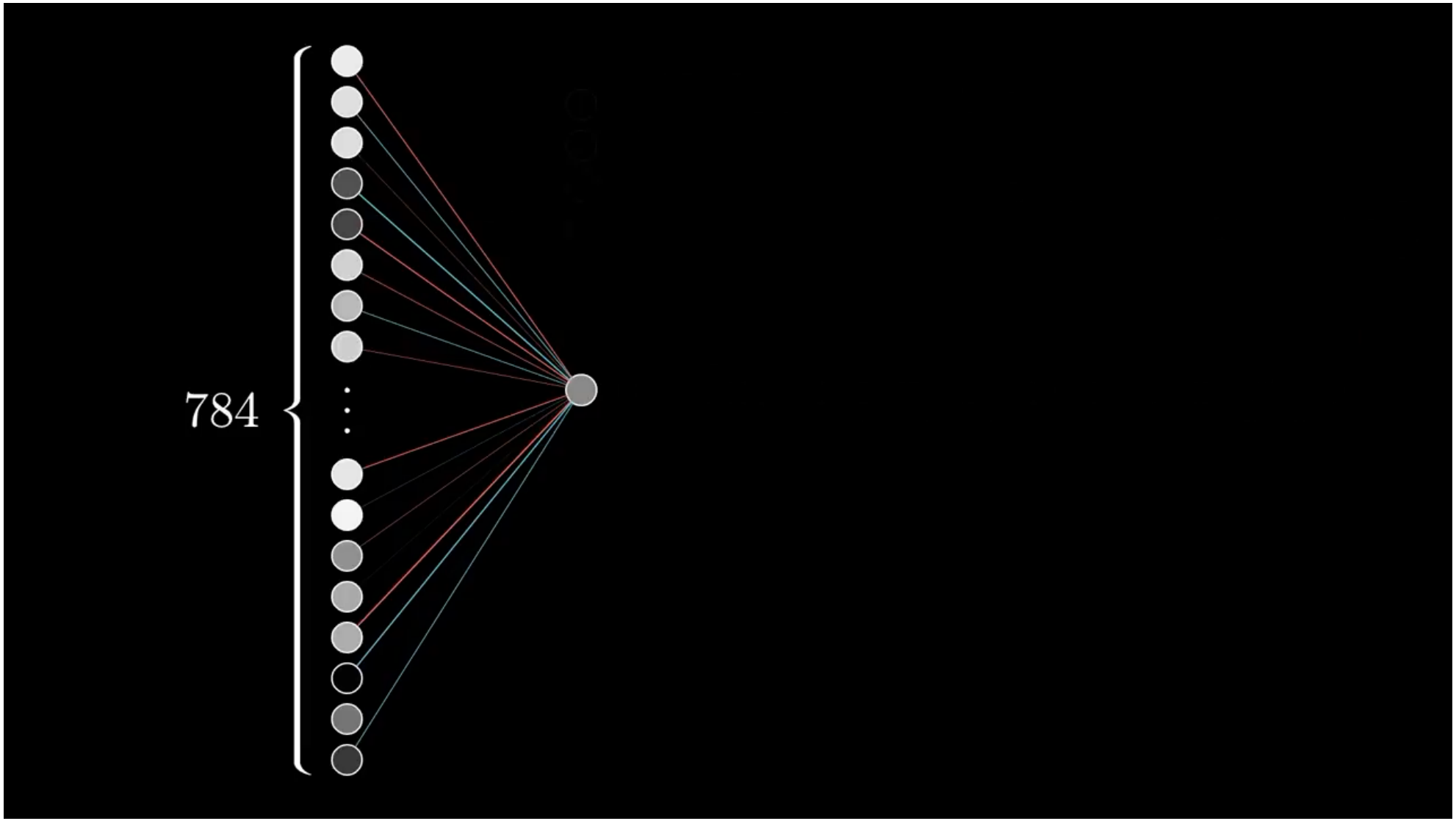
Score function



https://www.3blue1brown.com/lessons/neural-networks

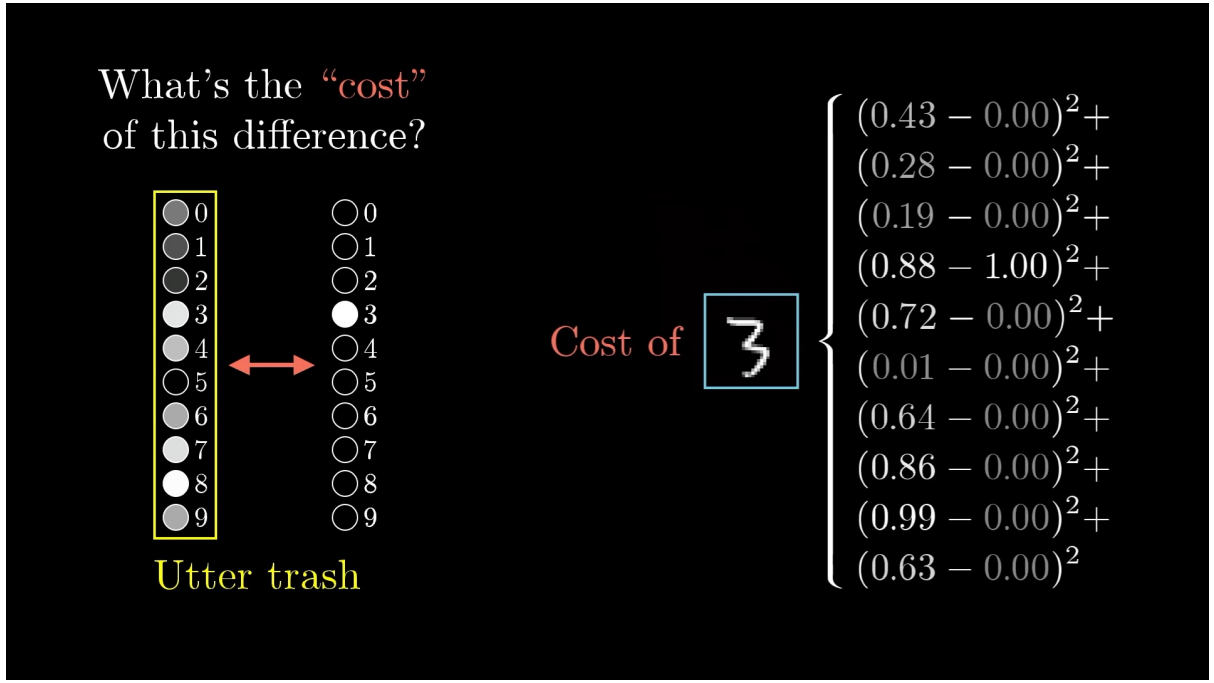
How the network learns?

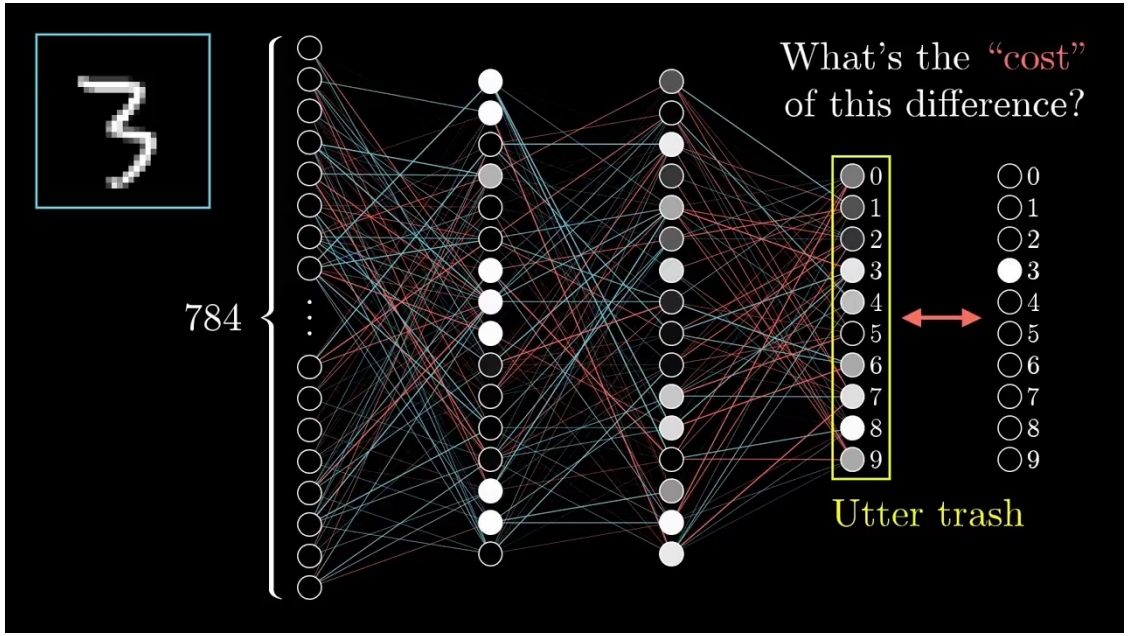
Cost function



https://www.3blue1brown.com/lessons/gradient-descent

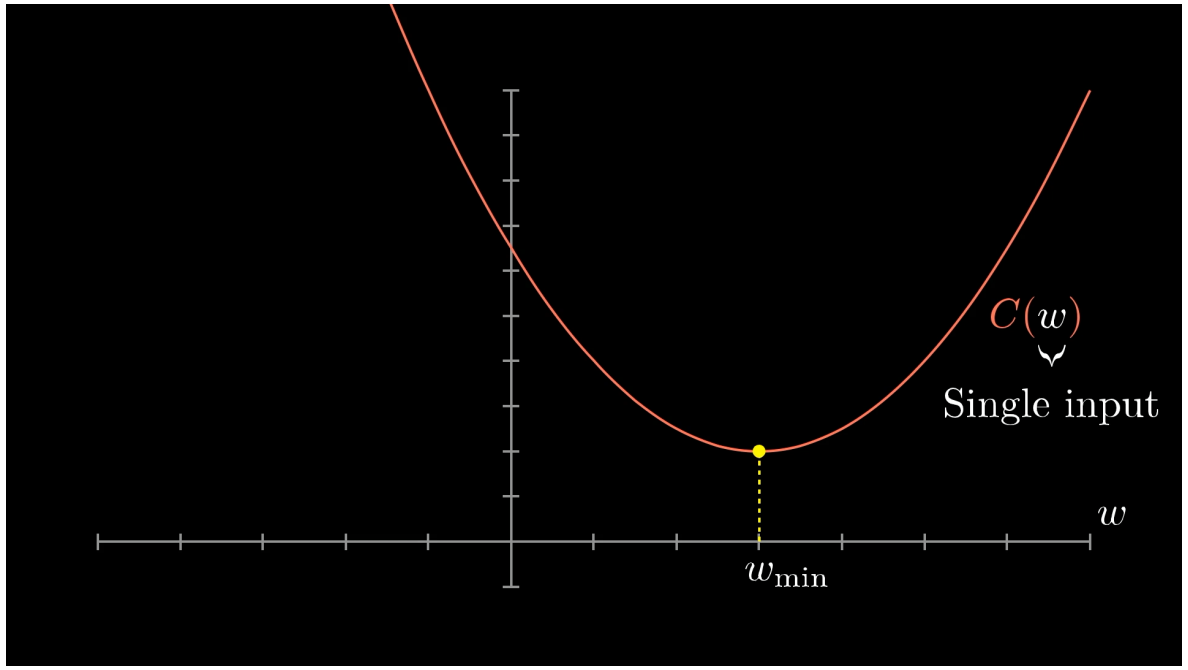
Cost Function

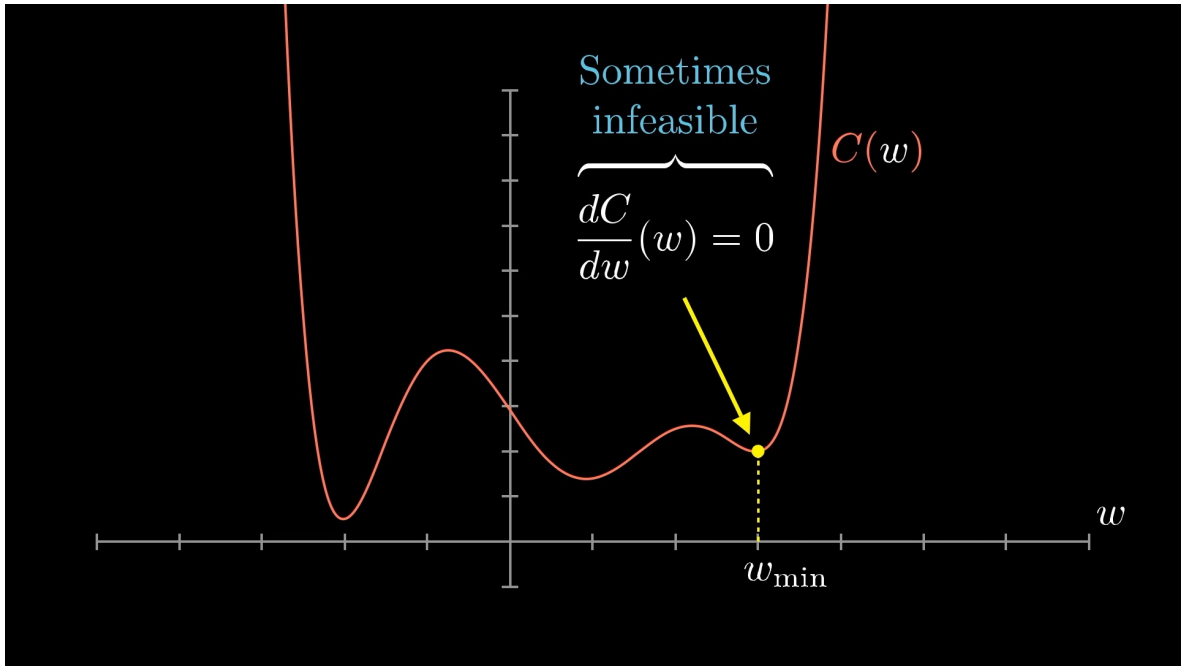




https://www.3blue1brown.com/lessons/gradient-descent

Cost Function

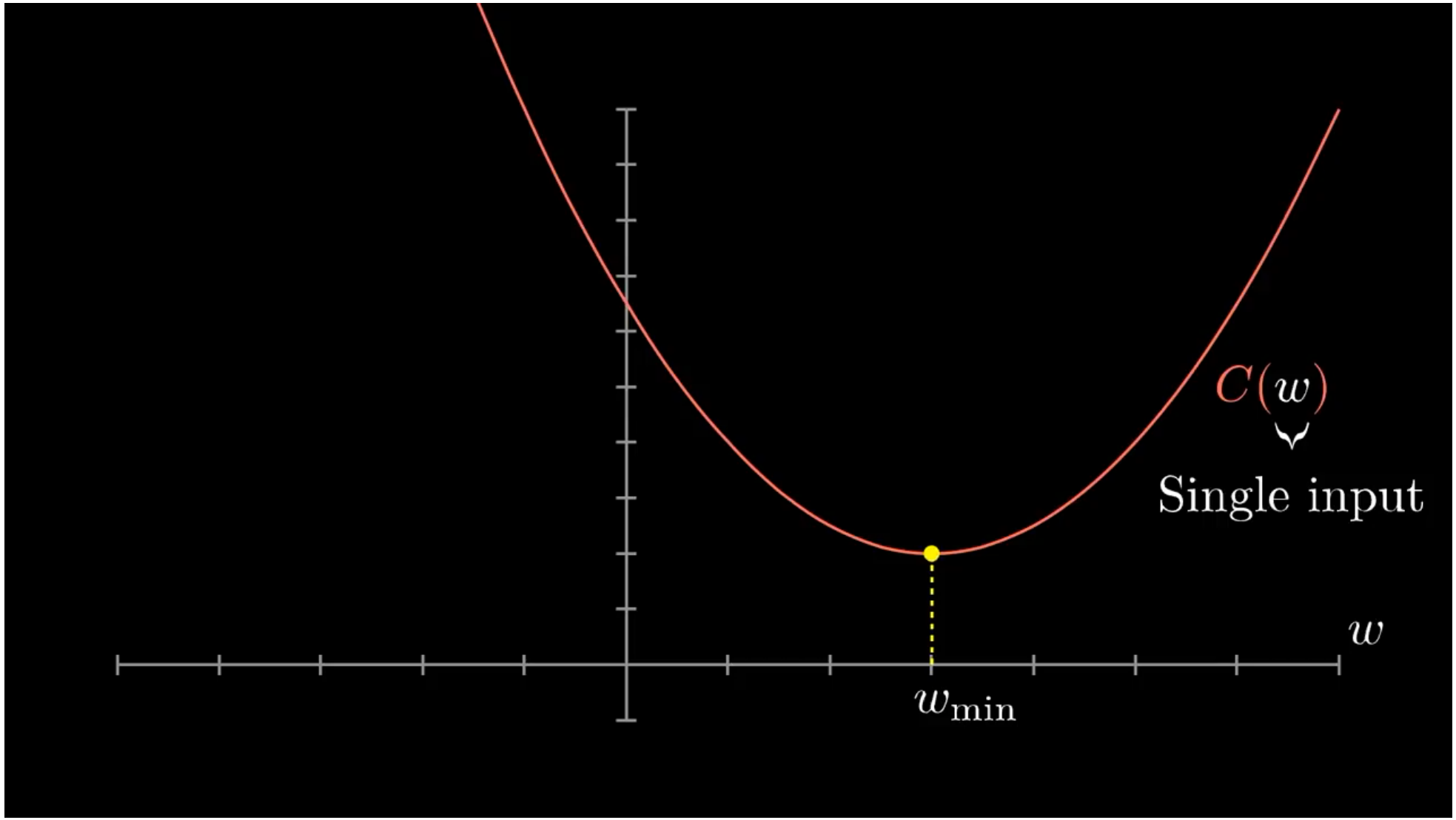




https://www.3blue1brown.com/lessons/gradient-descent

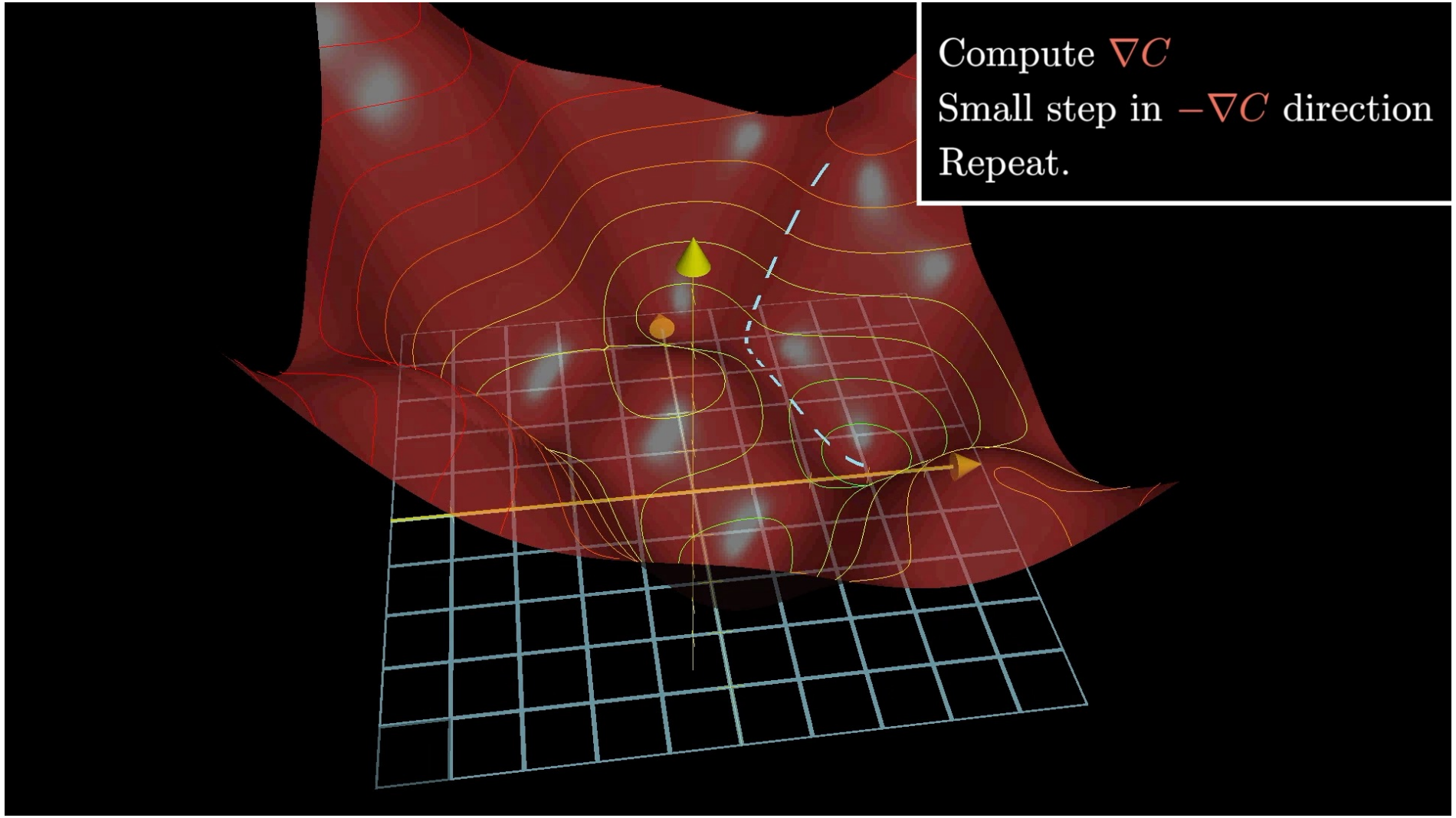
Optimization

Gradient Descent



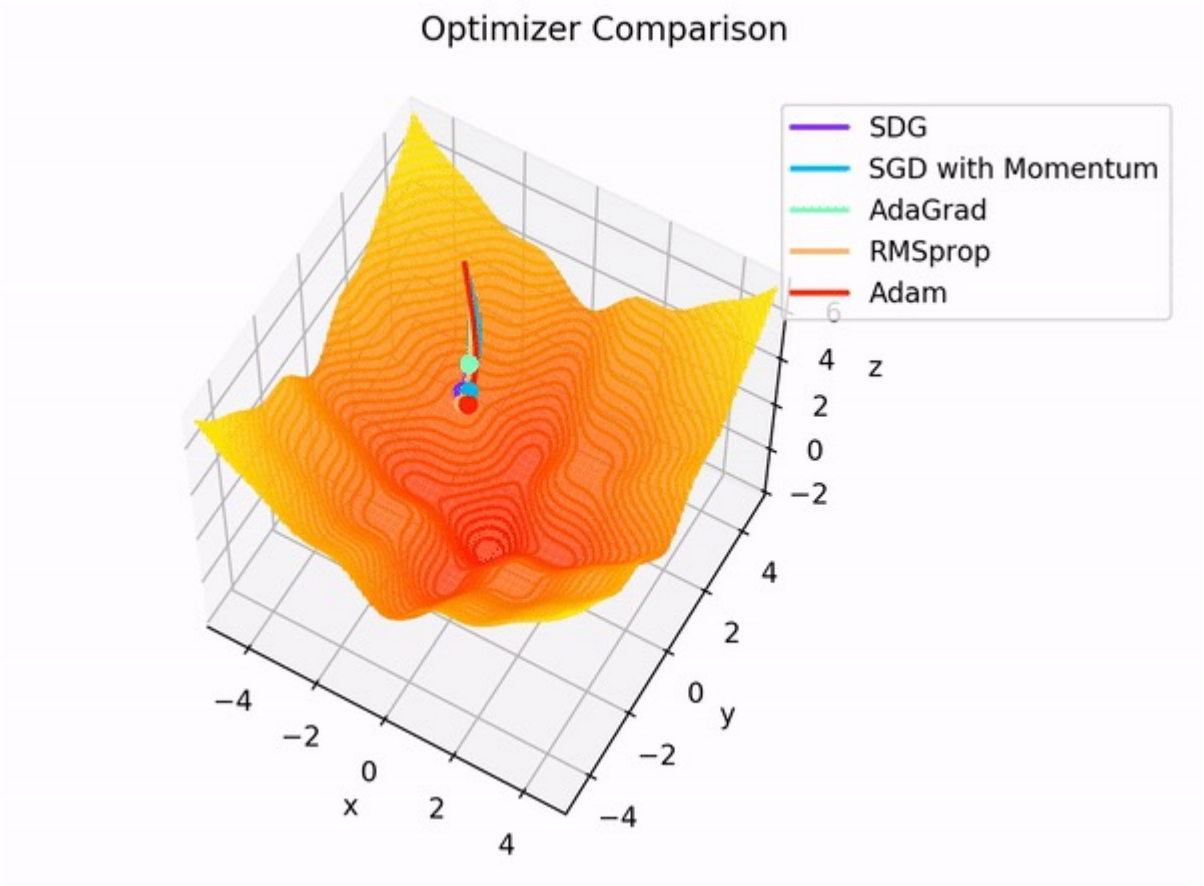
https://www.3blue1brown.com/lessons/gradient-descent

Gradient Descent



https://www.3blue1brown.com/lessons/gradient-descent

Gradient Descent



https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d

Gradient Descent

13,002 weights and biases

$$\vec{\mathbf{W}} = \begin{bmatrix} 2.25 \\ -1.57 \\ 1.98 \\ \vdots \\ -1.16 \\ 3.82 \\ 1.21 \end{bmatrix} -$$

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

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

https://www.3blue1brown.com/lessons/gradient-descent

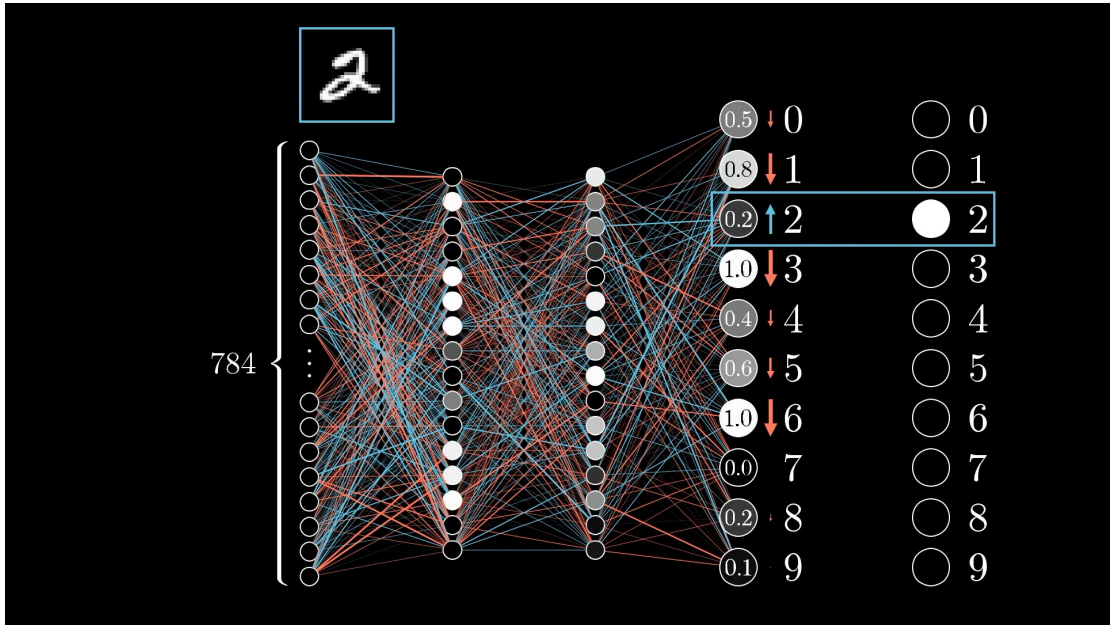
Gradient Descent

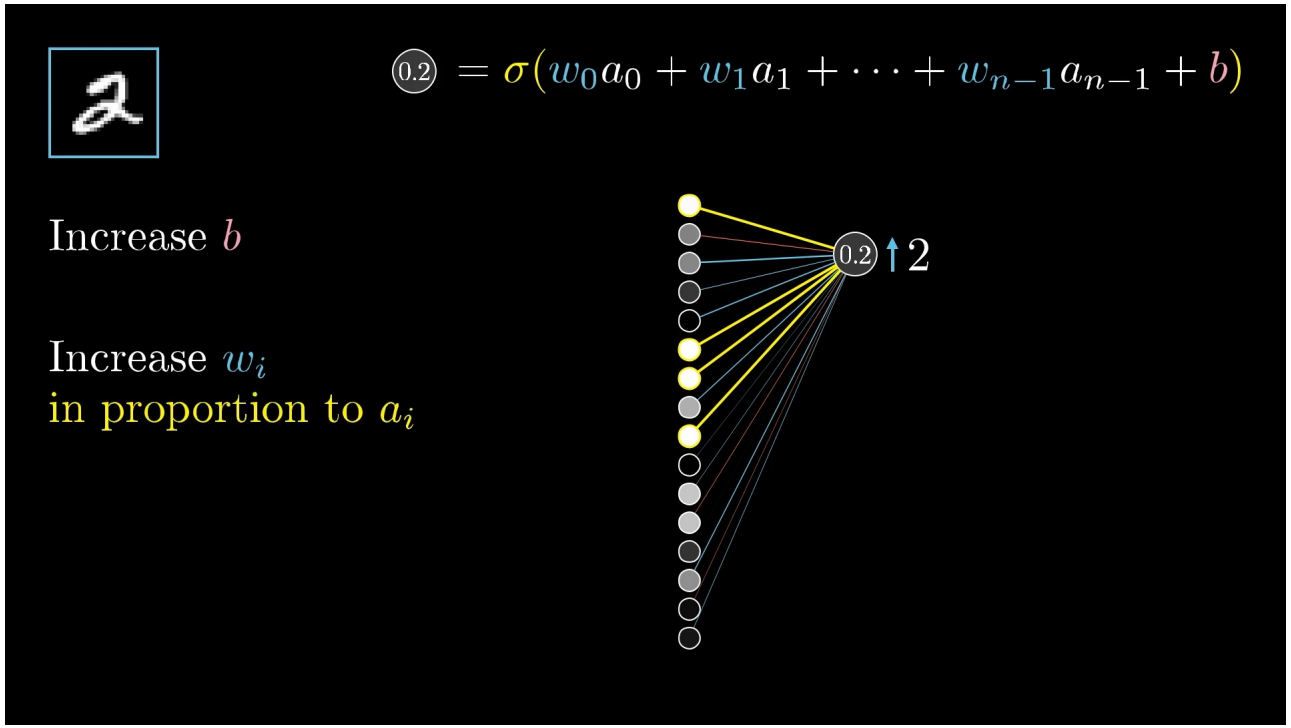
$$\vec{\mathbf{W}} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{13,000} \\ w_{13,001} \\ w_{13,002} \end{bmatrix}$$

$$-\nabla C(\vec{\mathbf{W}}) = \left.\begin{bmatrix} 0.31 \\ 0.03 \\ -1.25 \\ \vdots \\ 0.78 \\ -0.37 \\ 0.16 \end{bmatrix}\right\} \text{Example gradient}$$

https://www.3blue1brown.com/lessons/gradient-descent

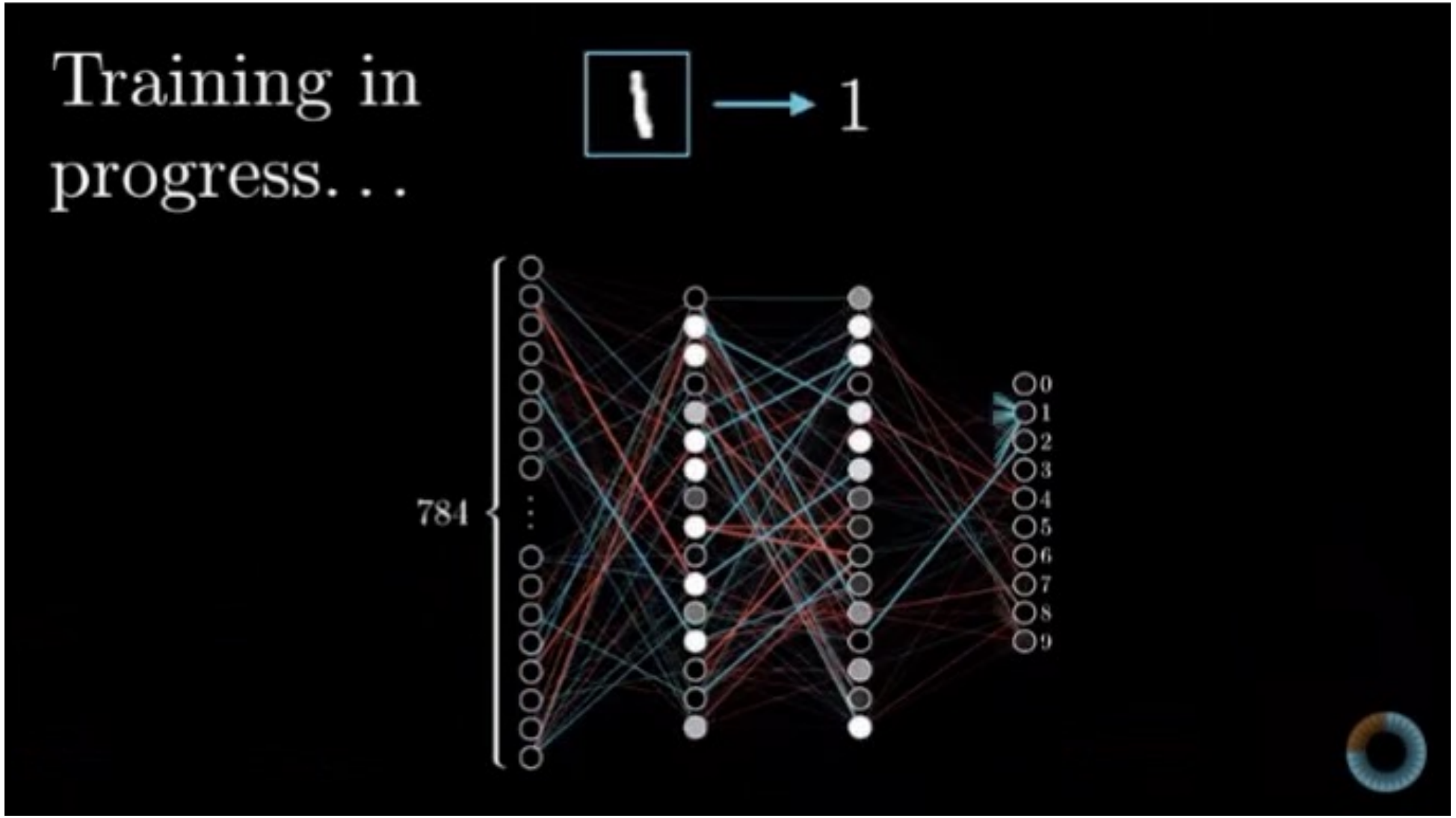
Backpropagation





https://www.3blue1brown.com/lessons/backpropagation

Backpropagation



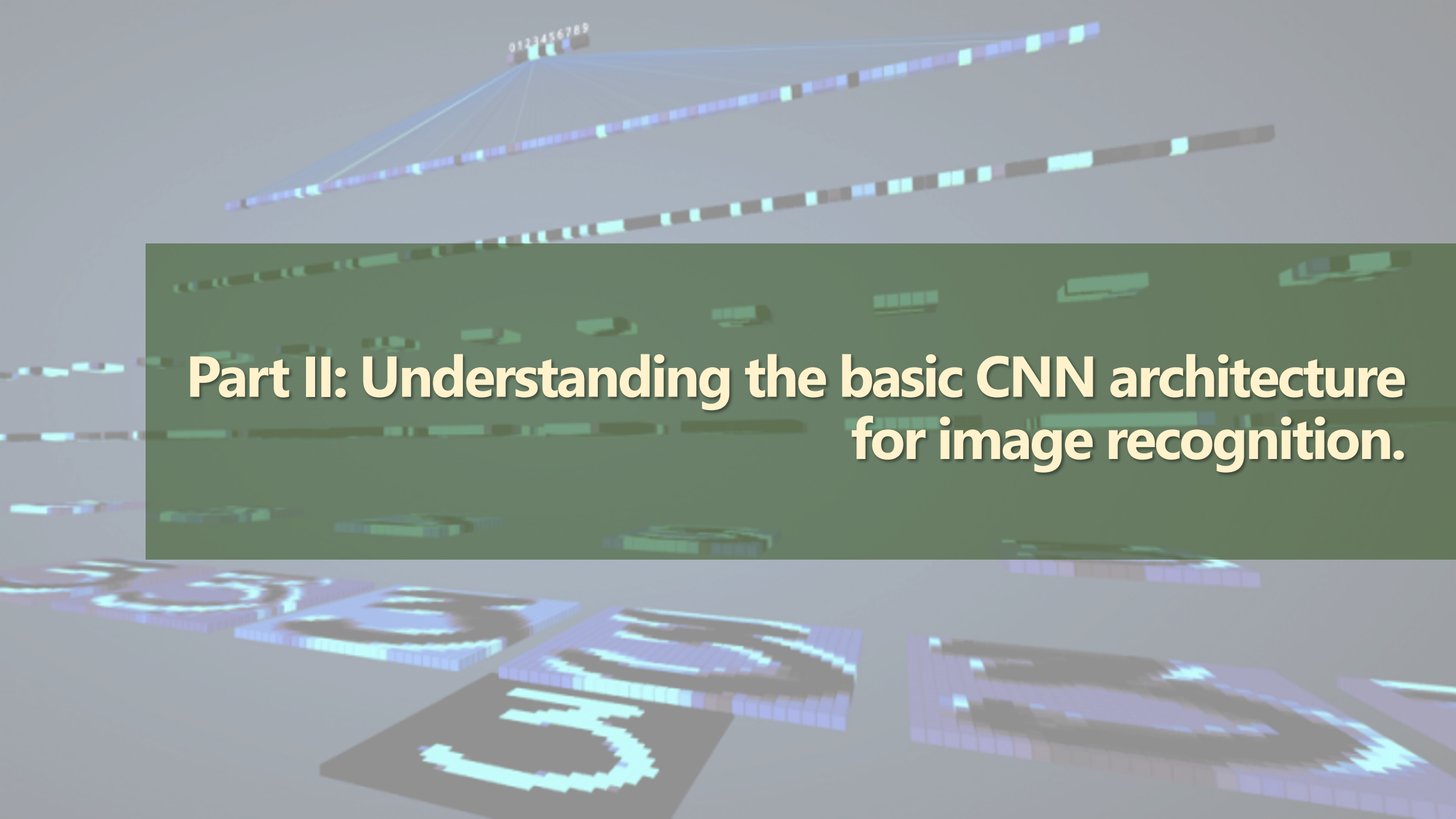
https://www.3blue1brown.com/lessons/backpropagation

- **Practical Exercise 1:**

- Access https://playground.tensorflow.org.

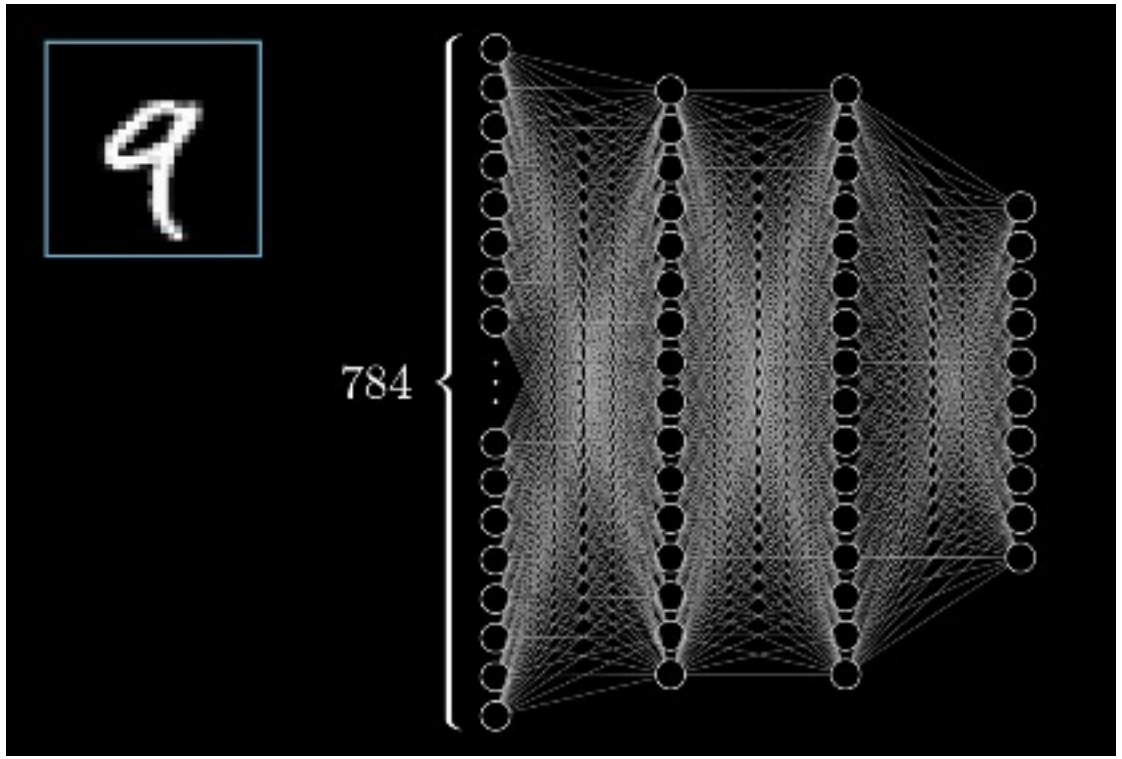
- Use the graphical interface to see how the neural network results changes when you modify the number of neurons, the input features, and the number of hidden layers.

- Compare the learning capacity when using a sigmoid and a ReLU activation. We will talk about these functions later.

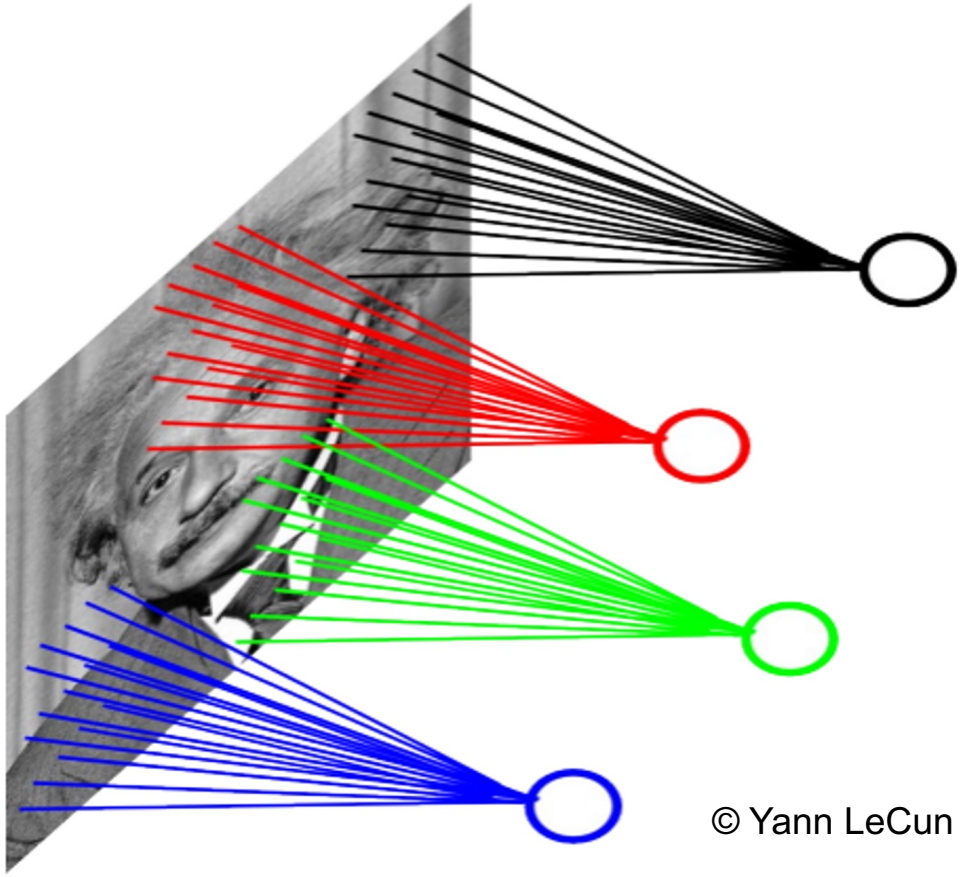


Part II: Understanding the basic CNN architecture for image recognition.

Convolutional Neural Networks (CNNs)

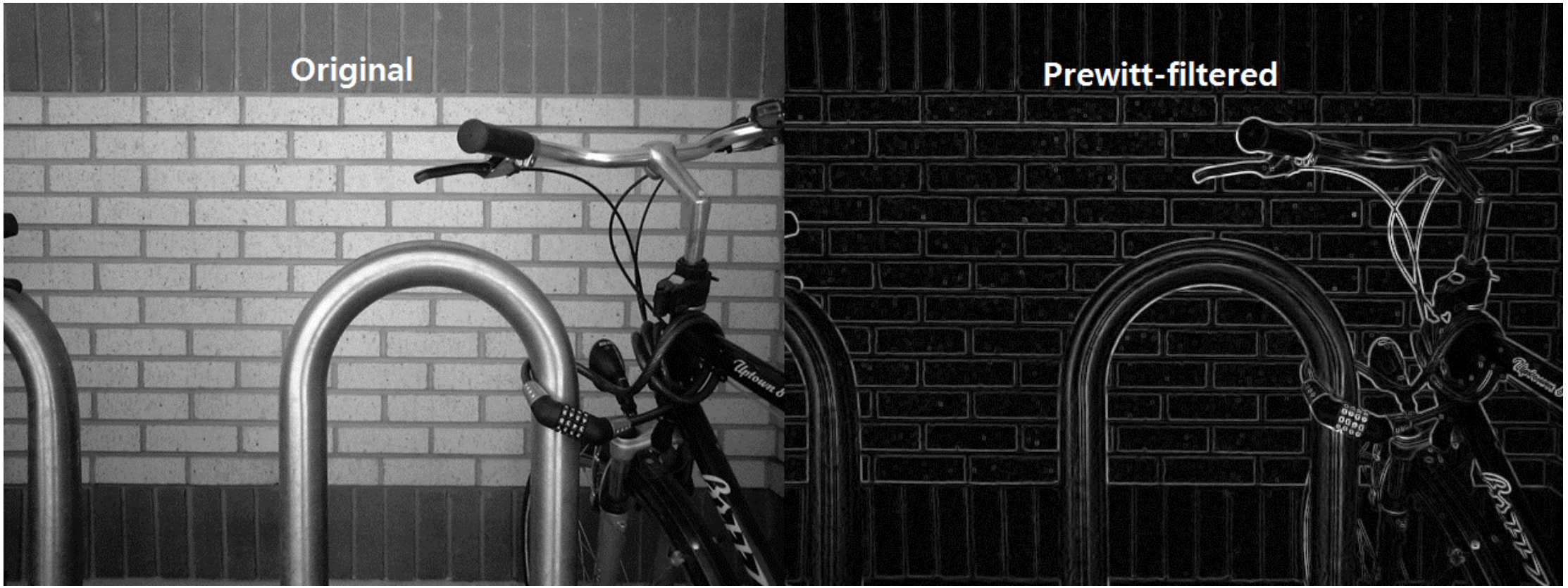


Convnet
locally connected
→ smaller number of parameters to learn



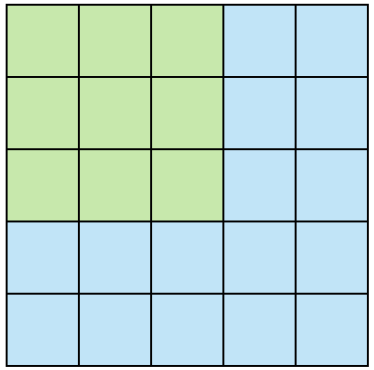
© Yann LeCun

CNNs

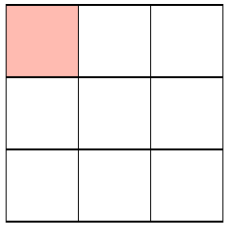


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

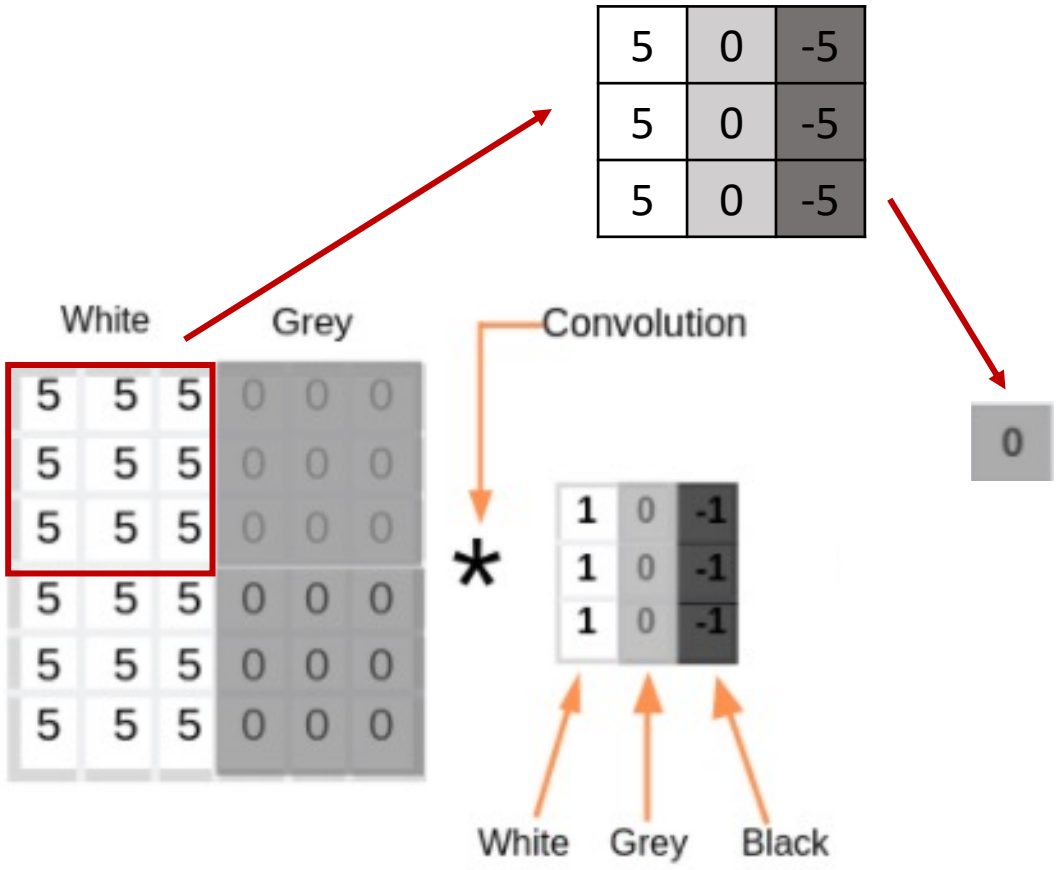
CNNs



Stride 1

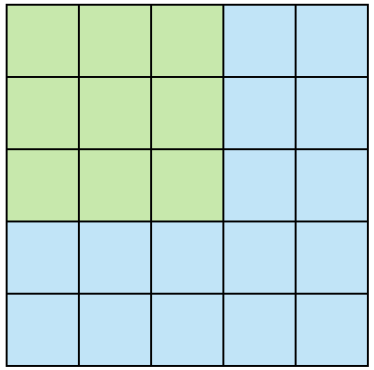


Feature Map

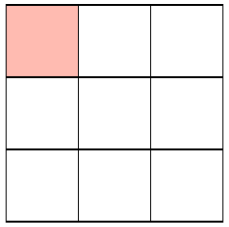


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

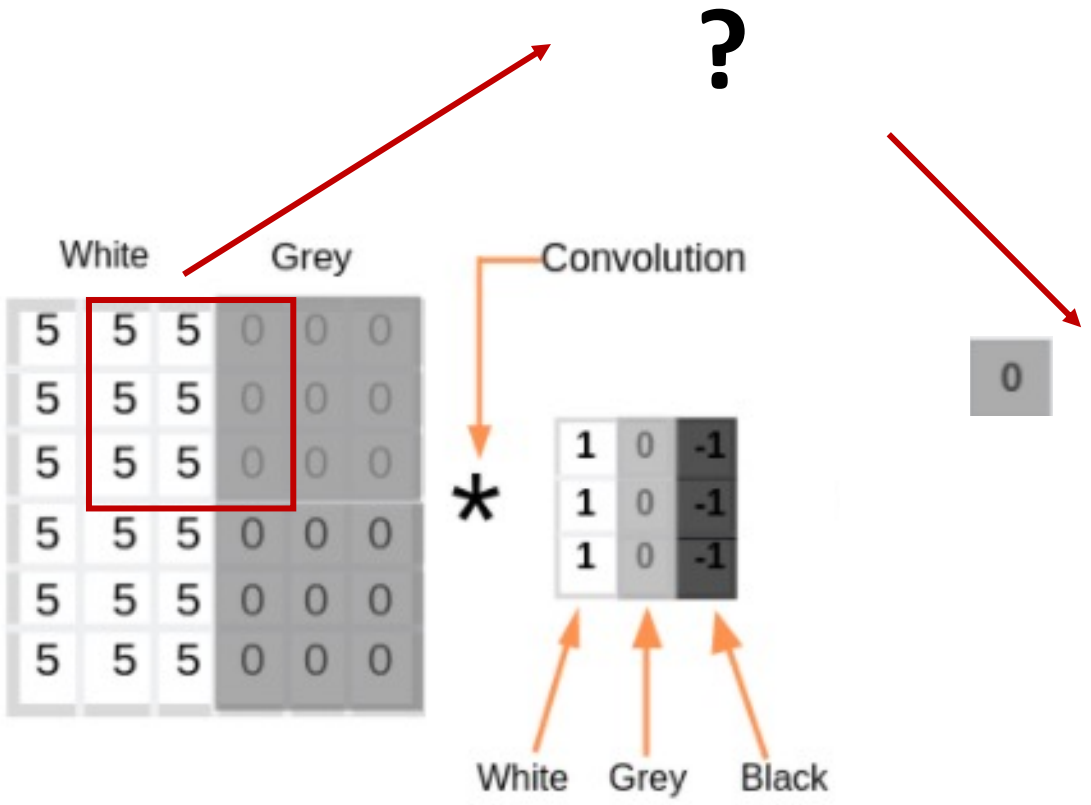
CNNs



Stride 1

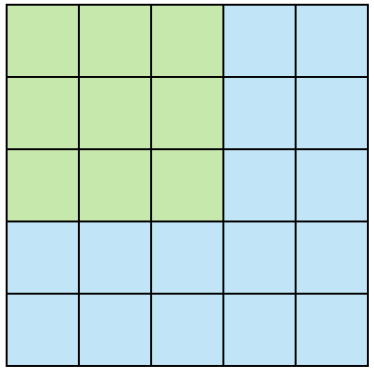


Feature Map

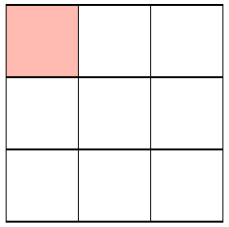


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

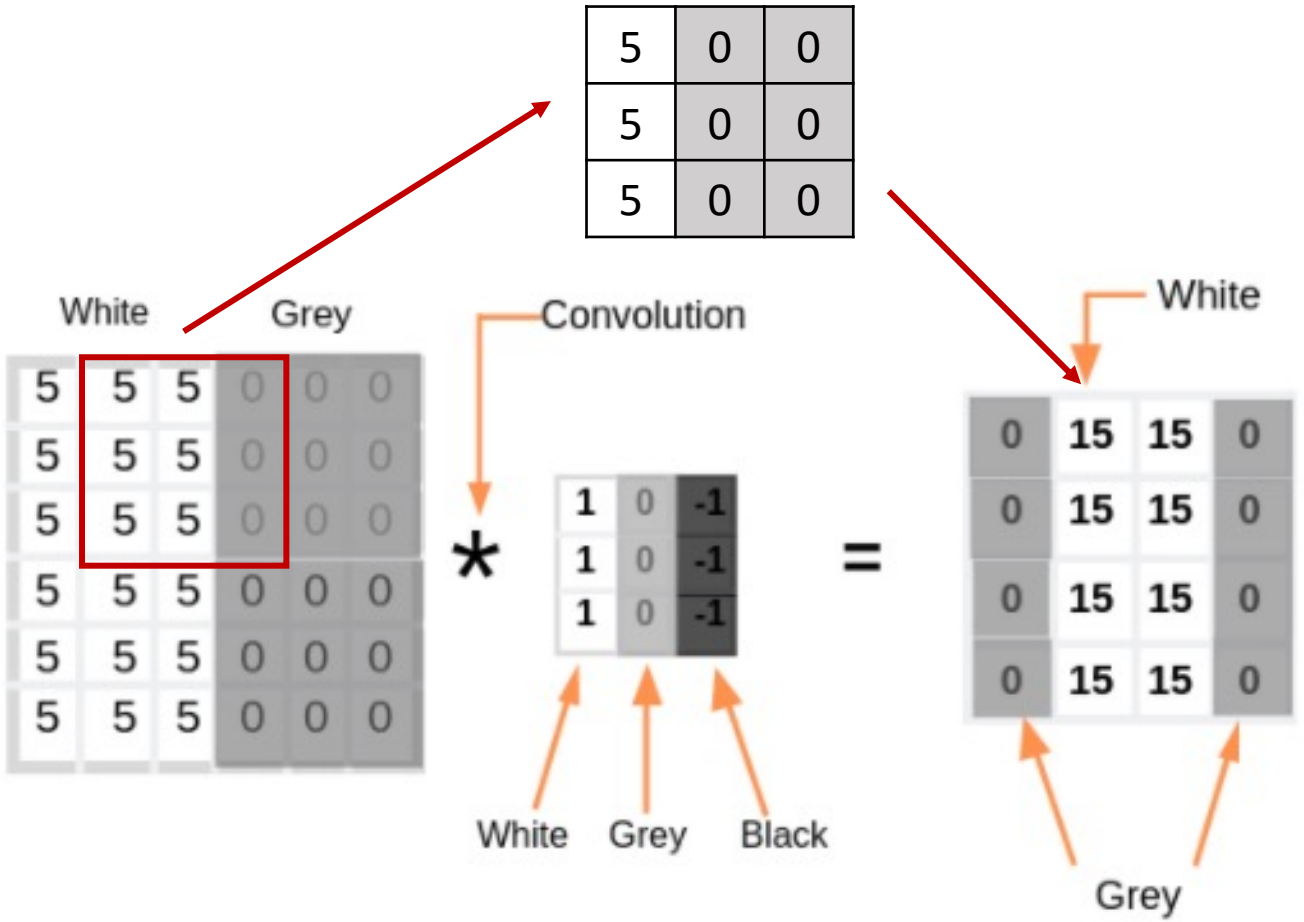
CNNs



Stride 1

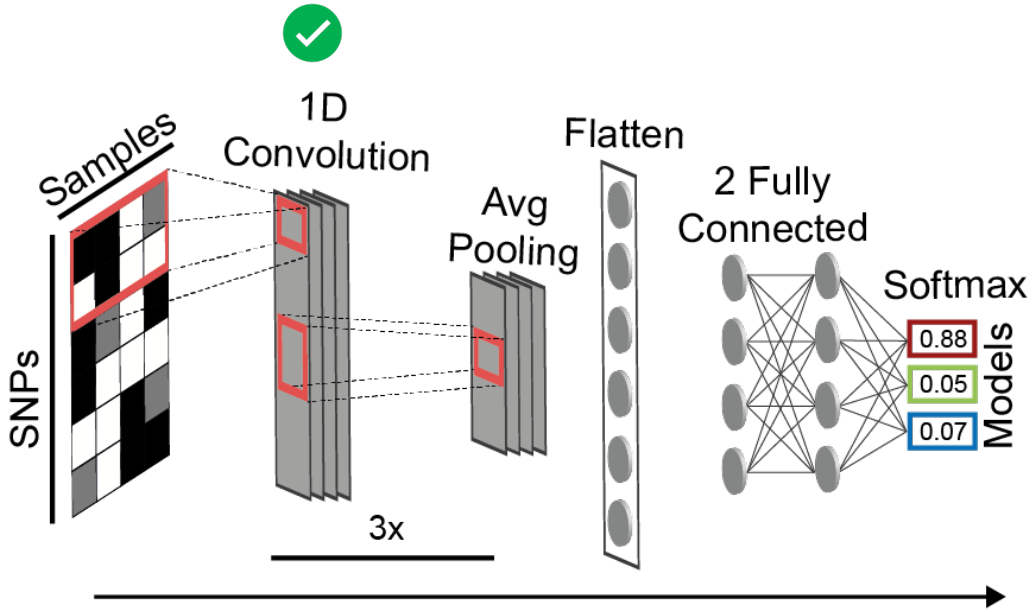


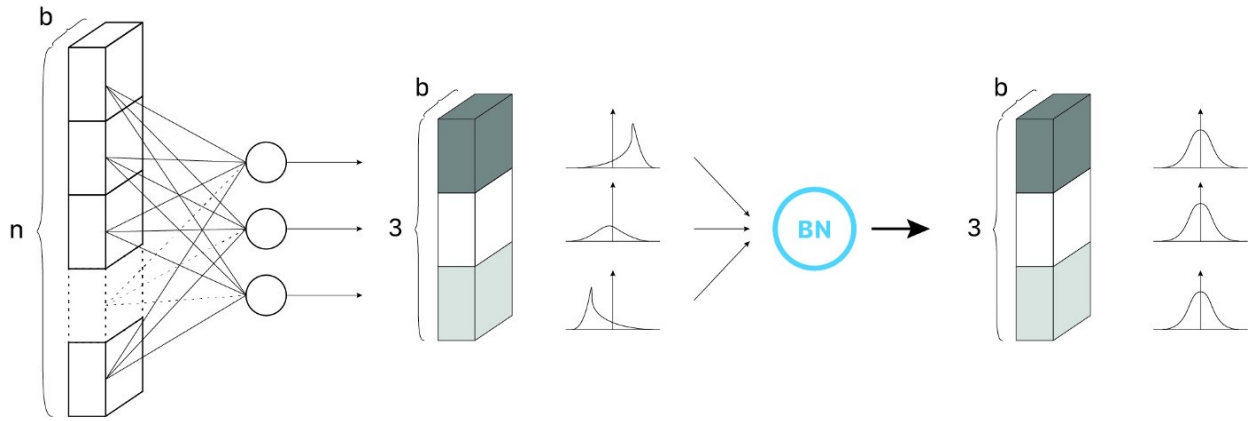
Feature Map



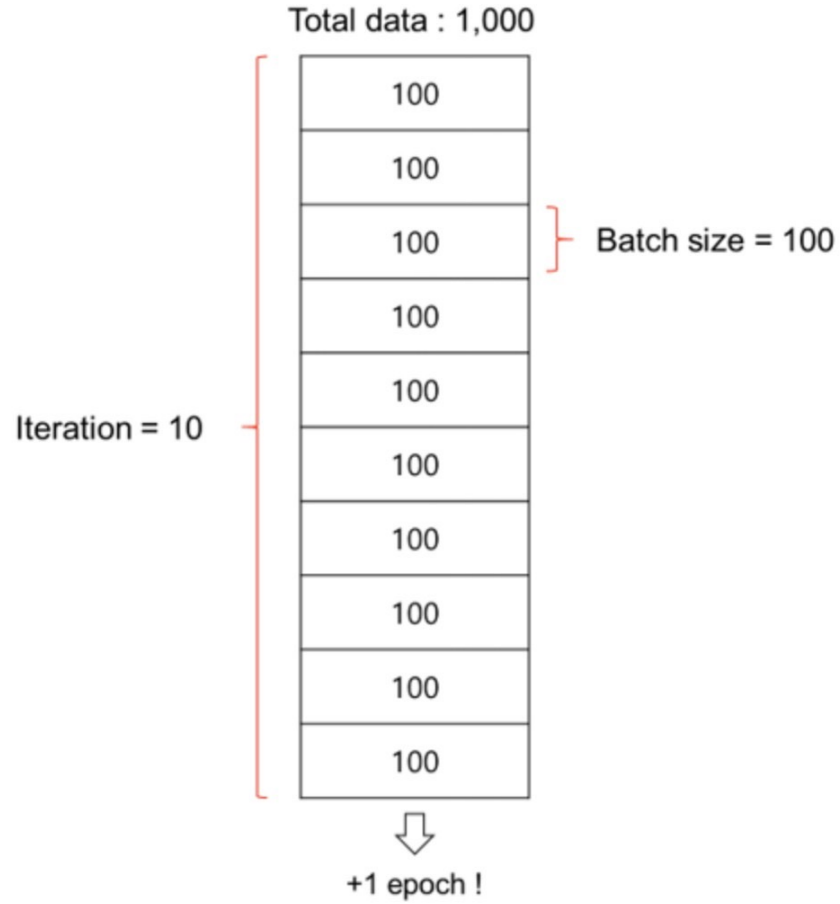
https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

CNN Script





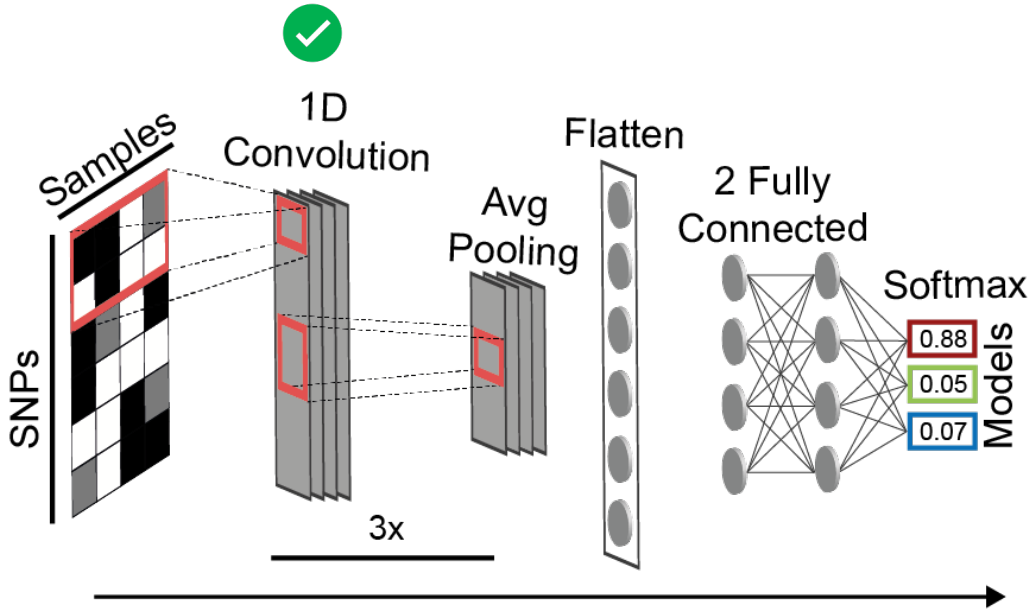
https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338

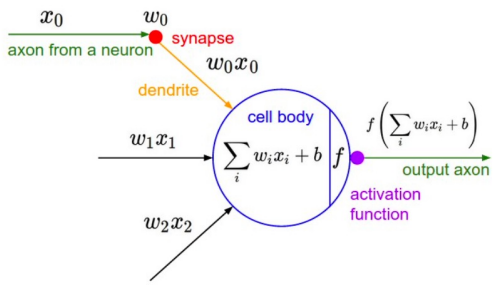


https://jerryan.medium.com/batch-size-a15958708a6

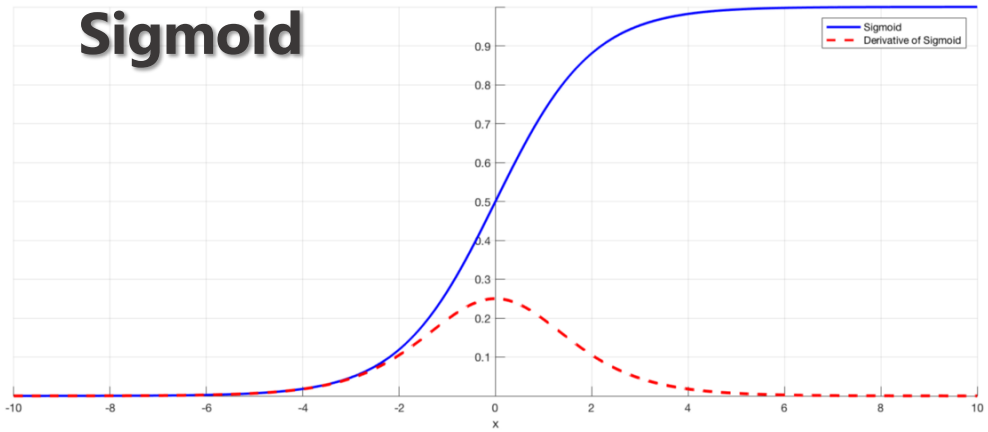
Kirschner et al. (2022) *Nat Comm*

CNN Script



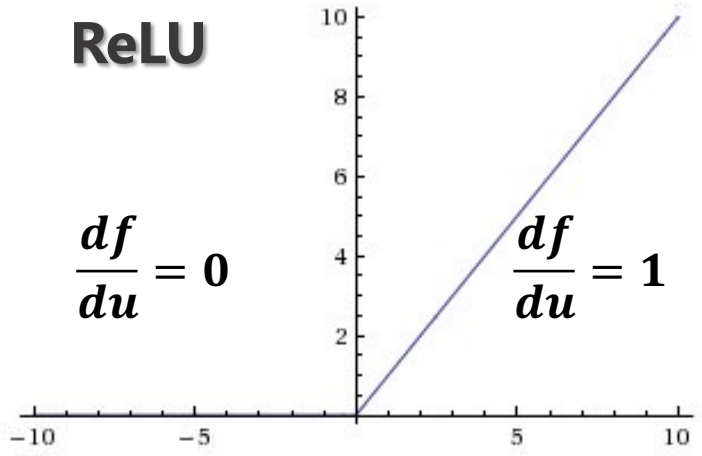


Sigmoid



https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

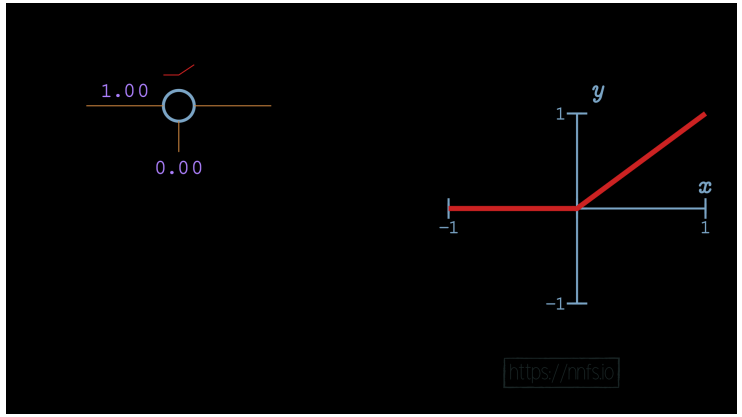
ReLU



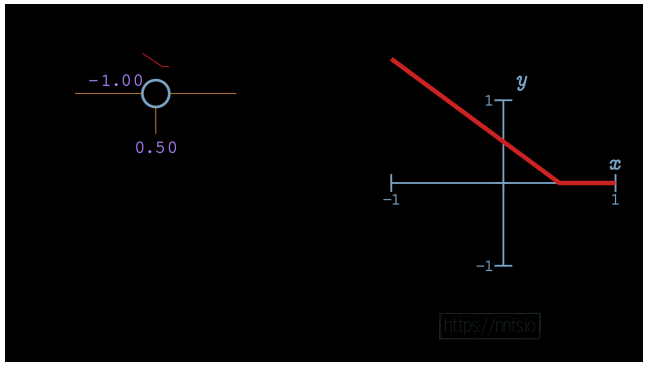
https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook

Kirschner et al. (2022) *Nat Comm*

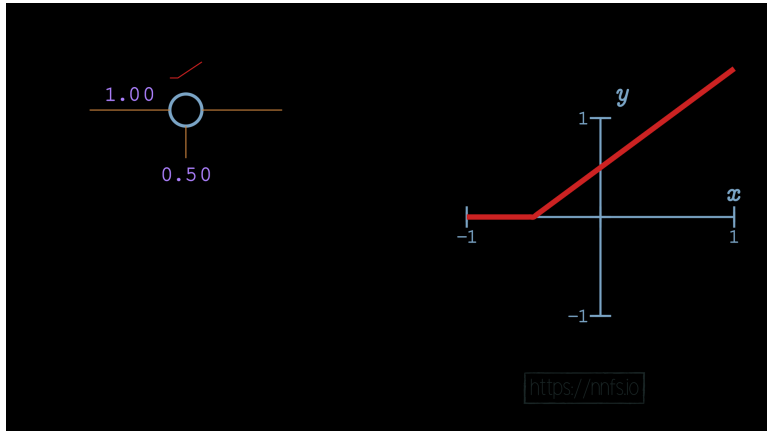
CNN Script – ReLU



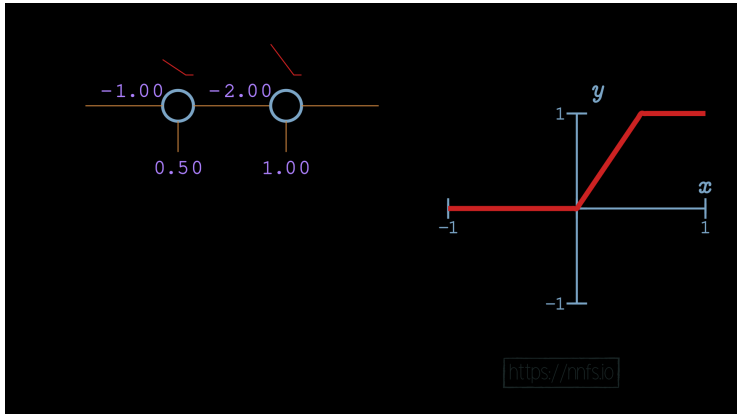
Weight = 1
Bias = 0



Weight = -1
Bias = 0.5



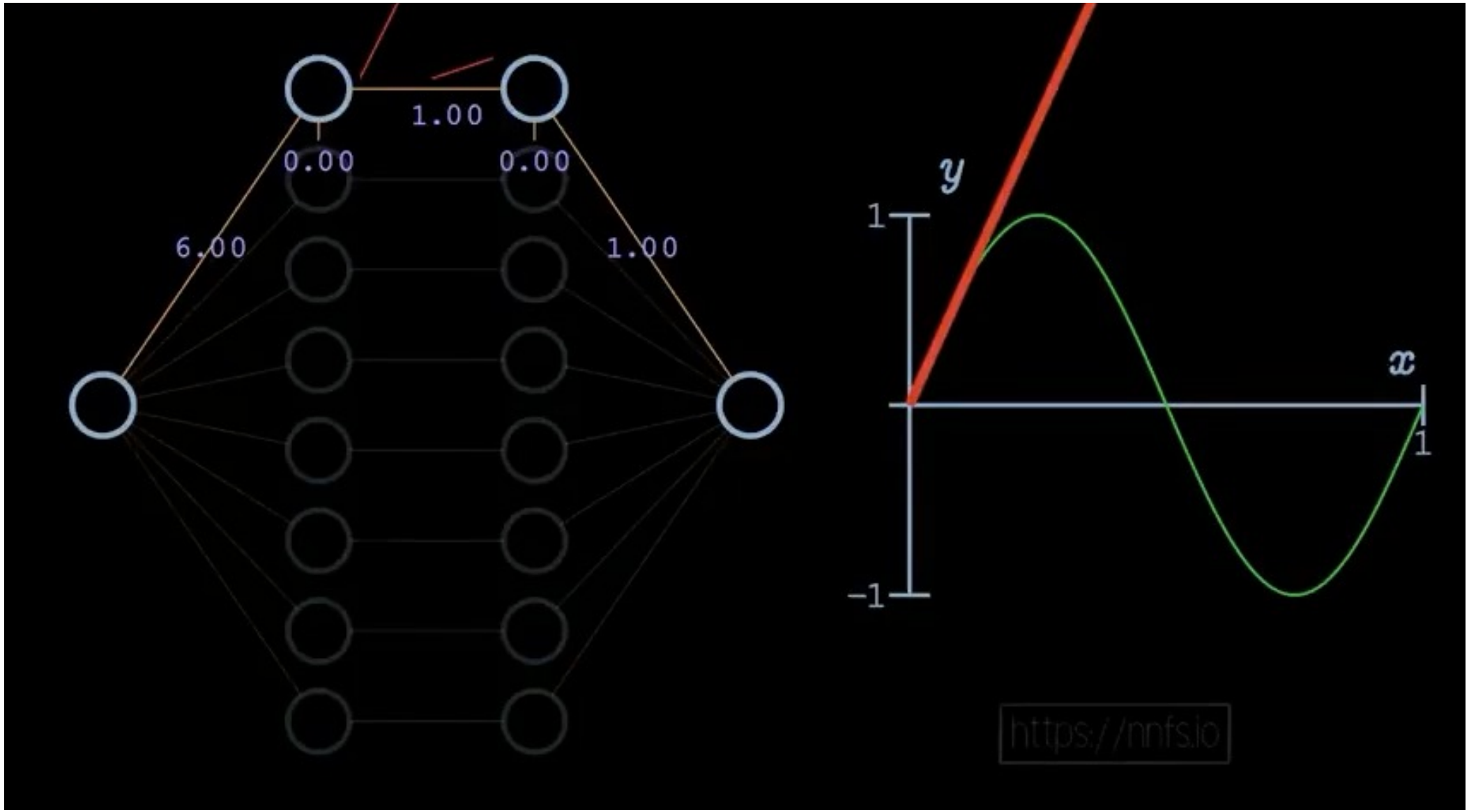
Weight = 1
Bias = 0.5



2 neurons

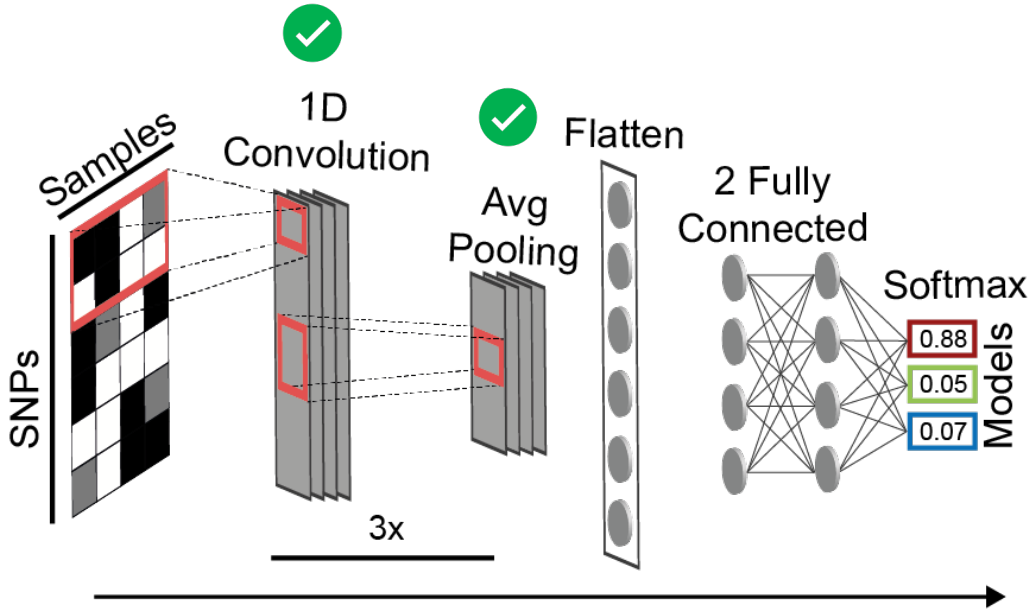
https://nnfs.io/mvp/

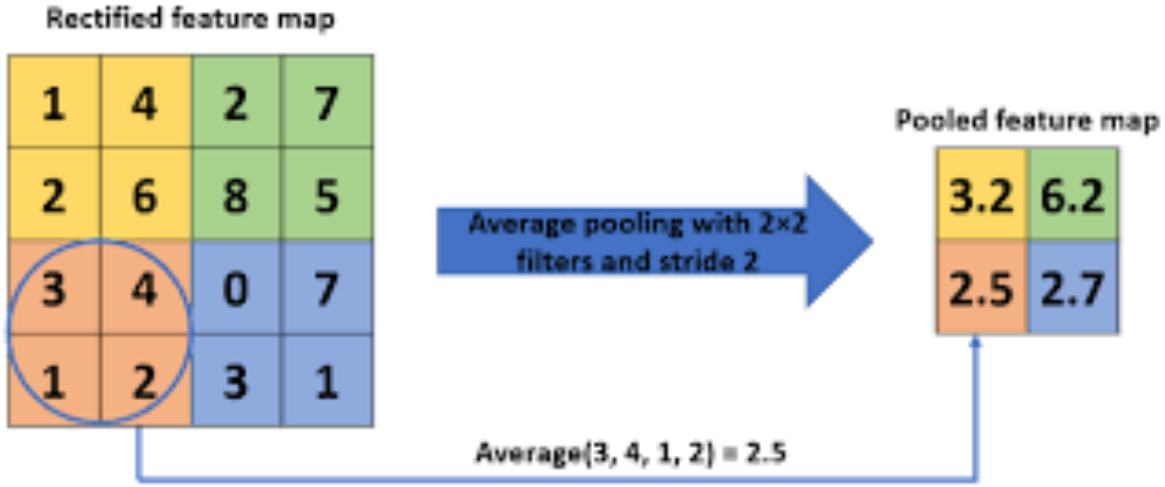
CNN Script – ReLU



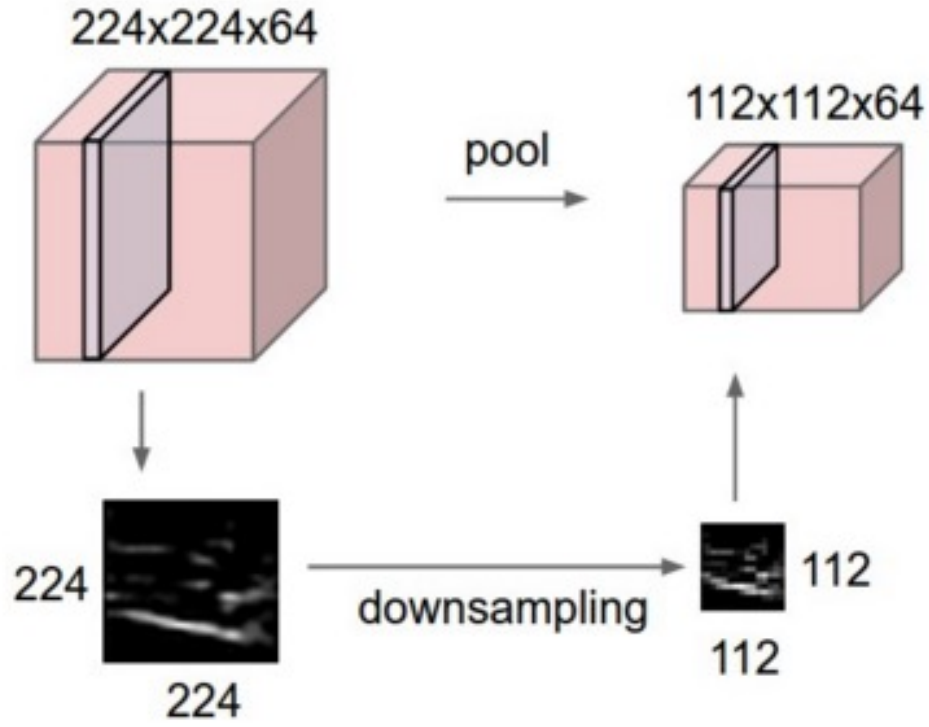
https://nnfs.io/mvp/

CNN Script





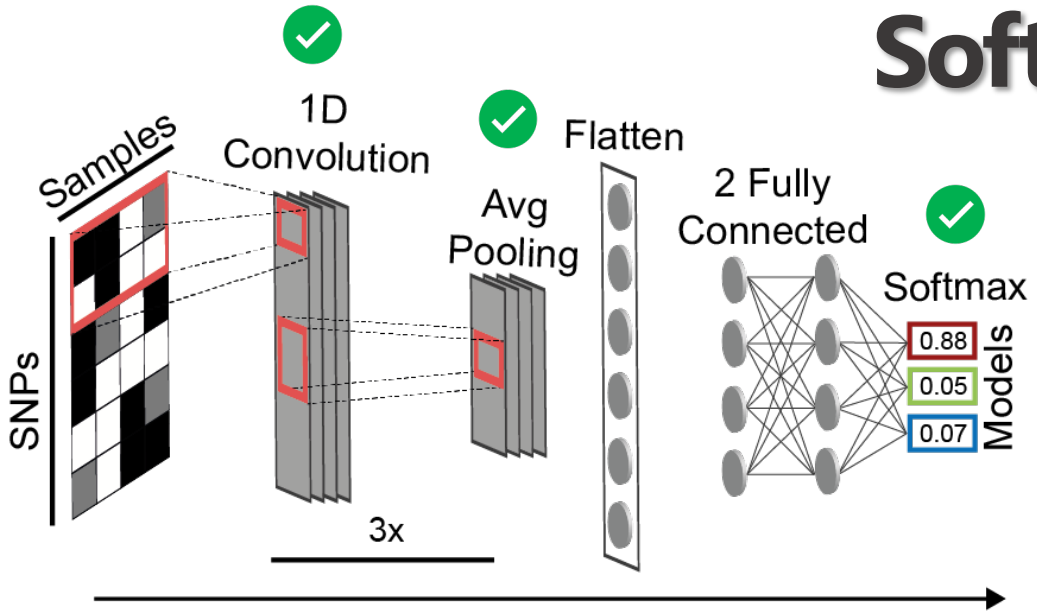
Gholamalinezhad & Khosravi (2020) *arXiv*

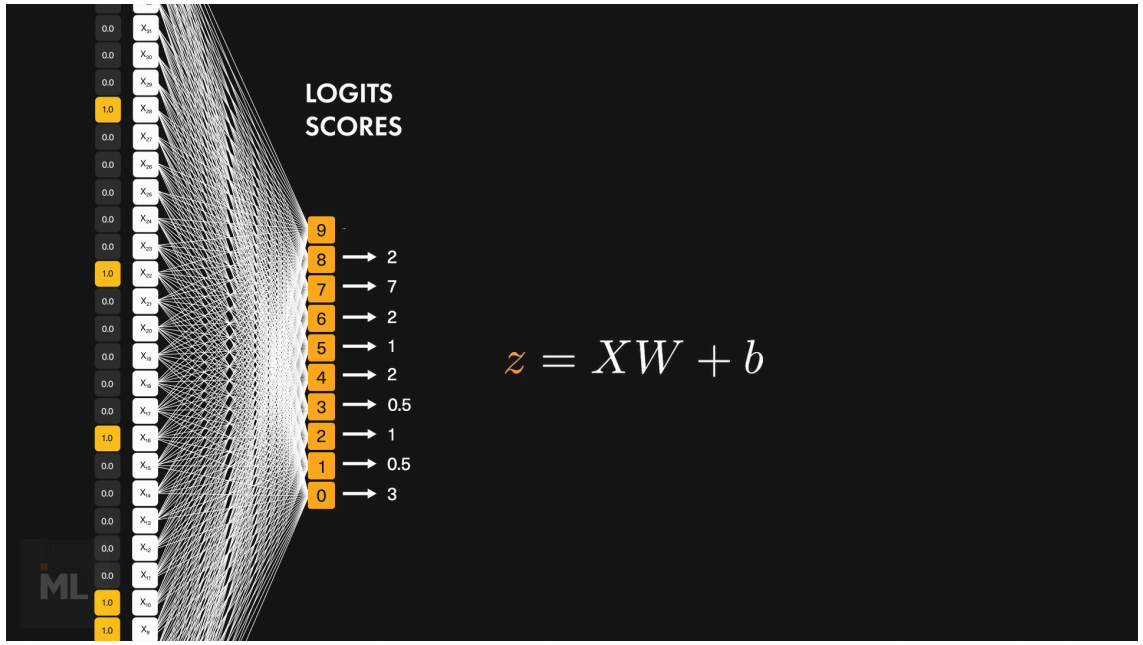


https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/pooling_layer

Kirschner et al. (2022) *Nat Comm*

Softmax layer for Classification





$$z = XW + b$$

Hyperparameters

Libraries for deep Learning:
Keras, Tensorflow, pyTorch

Table 2. Central parameters of a neural network and recommended settings.

Name	Range	Default value
Learning rate	0.1, 0.01, 0.001, 0.0001	0.01
Batch size	64, 128, 256	128
Momentum rate	0.8, 0.9, 0.95	0.9
Weight initialization	Normal, Uniform, Glorot uniform	Glorot uniform
Per-parameter adaptive learning rate methods	RMSprop, Adagrad, Adadelta, Adam	Adam
Batch normalization	Yes, no	Yes
Learning rate decay	None, linear, exponential	Linear (rate 0.5)
Activation function	Sigmoid, Tanh, ReLU, Softmax	ReLU
Dropout rate	0.1, 0.25, 0.5, 0.75	0.5
L1, L2 regularization	0, 0.01, 0.001	

Some of the best hyperparameter optimization libraries are:

1. Scikit-learn
2. Scikit-Optimize
3. Optuna
4. Hyperopt
5. Ray.tune
6. Talos
7. BayesianOptimization
8. Metric Optimization Engine (MOE)
9. Spearmint
10. GPyOpt
11. SigOpt
12. Fabolas

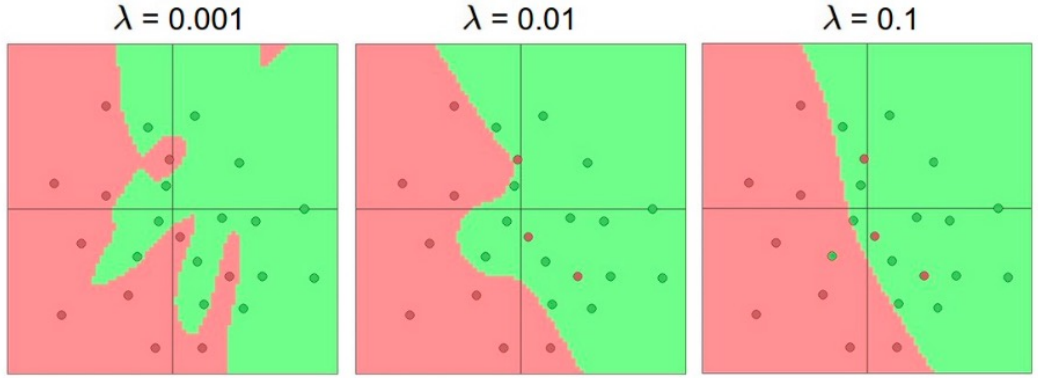
https://neptune.ai/blog/hyperparameter-tuning-in-python-complete-guide#hyperopt

Hyperparameters

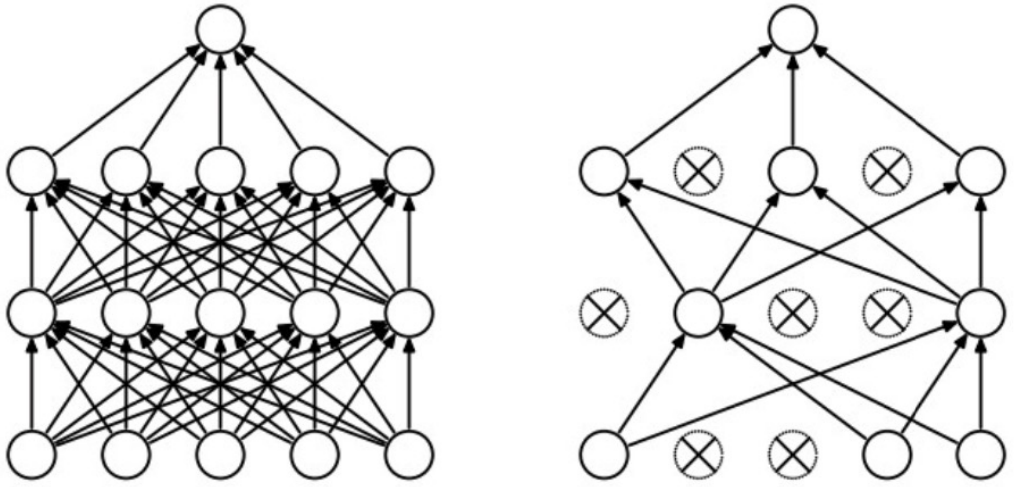
Table 2. Central parameters of a neural network and recommended settings.

Name	Range	Default value
Learning rate	0.1, 0.01, 0.001, 0.0001	0.01
Batch size	64, 128, 256	128
Momentum rate	0.8, 0.9, 0.95	0.9
Weight initialization	Normal, Uniform, Glorot uniform	Glorot uniform
Per-parameter adaptive learning rate methods	RMSprop, Adagrad, Adadelta, Adam	Adam
Batch normalization	Yes, no	Yes
Learning rate decay	None, linear, exponential	Linear (rate 0.5)
Activation function	Sigmoid, Tanh, ReLU, Softmax	ReLU
Dropout rate	0.1, 0.25, 0.5, 0.75	0.5
L1, L2 regularization	0, 0.01, 0.001	

Regularization



Options: L2, L1, maxnorm and dropout.

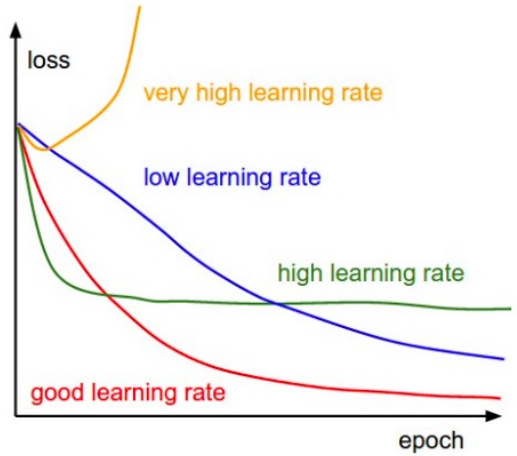


(a) Standard Neural Net (b) After applying dropout.

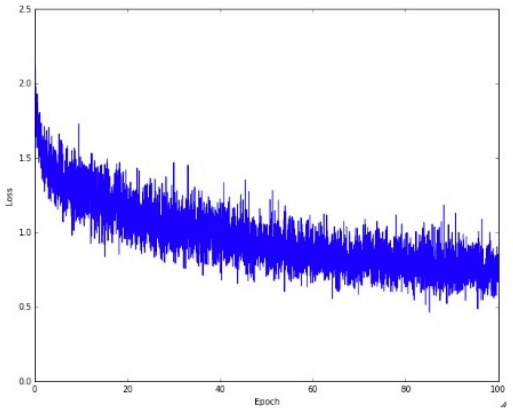
Hyperparameters

Table 2. Central parameters of a neural network and recommended settings.

Name	Range	Default value
Learning rate	0.1, 0.01, 0.001, 0.0001	0.01
Batch size	64, 128, 256	128
Momentum rate	0.8, 0.9, 0.95	0.9
Weight initialization	Normal, Uniform, Glorot uniform	Glorot uniform
Per-parameter adaptive learning rate methods	RMSprop, Adagrad, Adadelta, Adam	Adam
Batch normalization	Yes, no	Yes
Learning rate decay	None, linear, exponential	Linear (rate 0.5)
Activation function	Sigmoid, Tanh, ReLU, Softmax	ReLU
Dropout rate	0.1, 0.25, 0.5, 0.75	0.5
L1, L2 regularization	0, 0.01, 0.001	



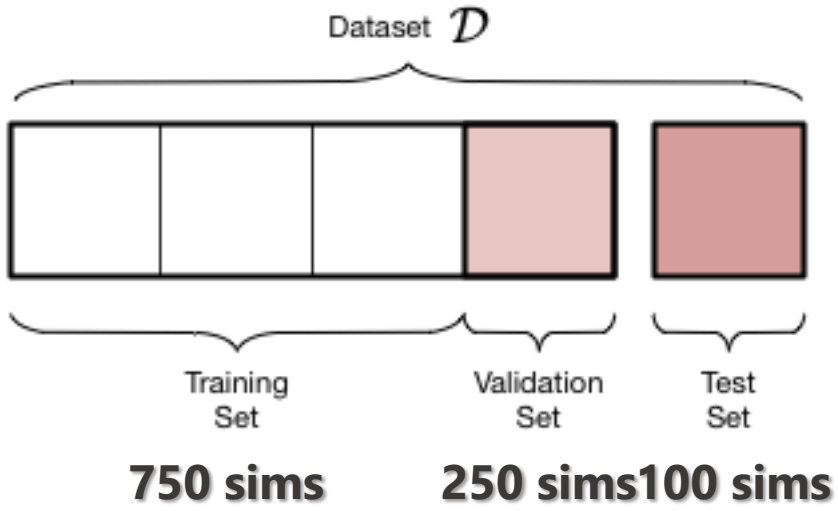
effects of different learning rates

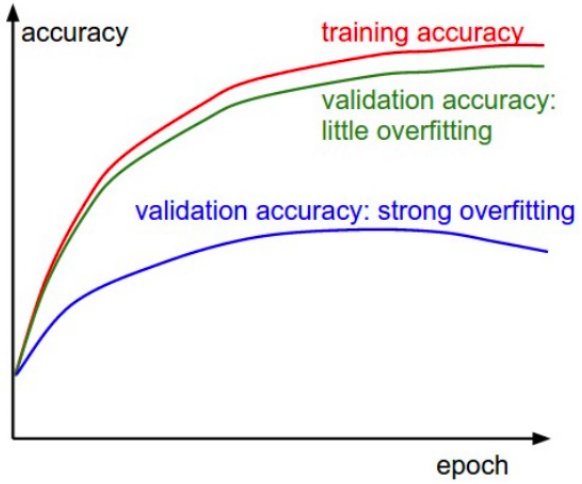


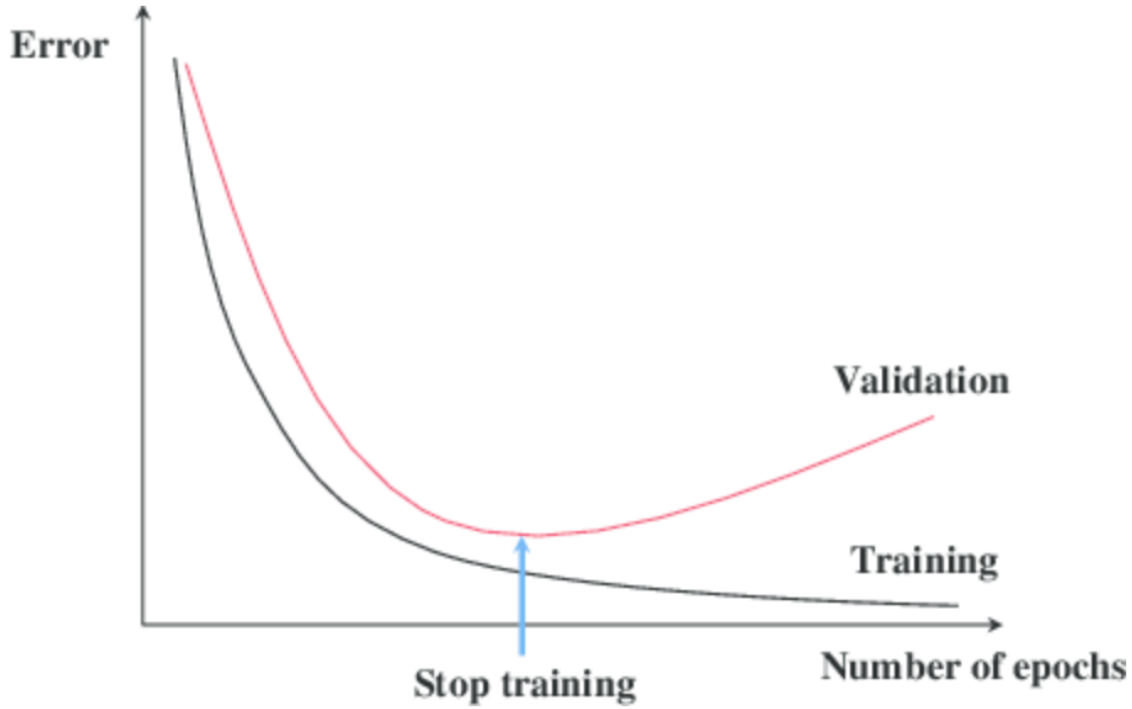
loss decay

Early Stop to avoid overfitting

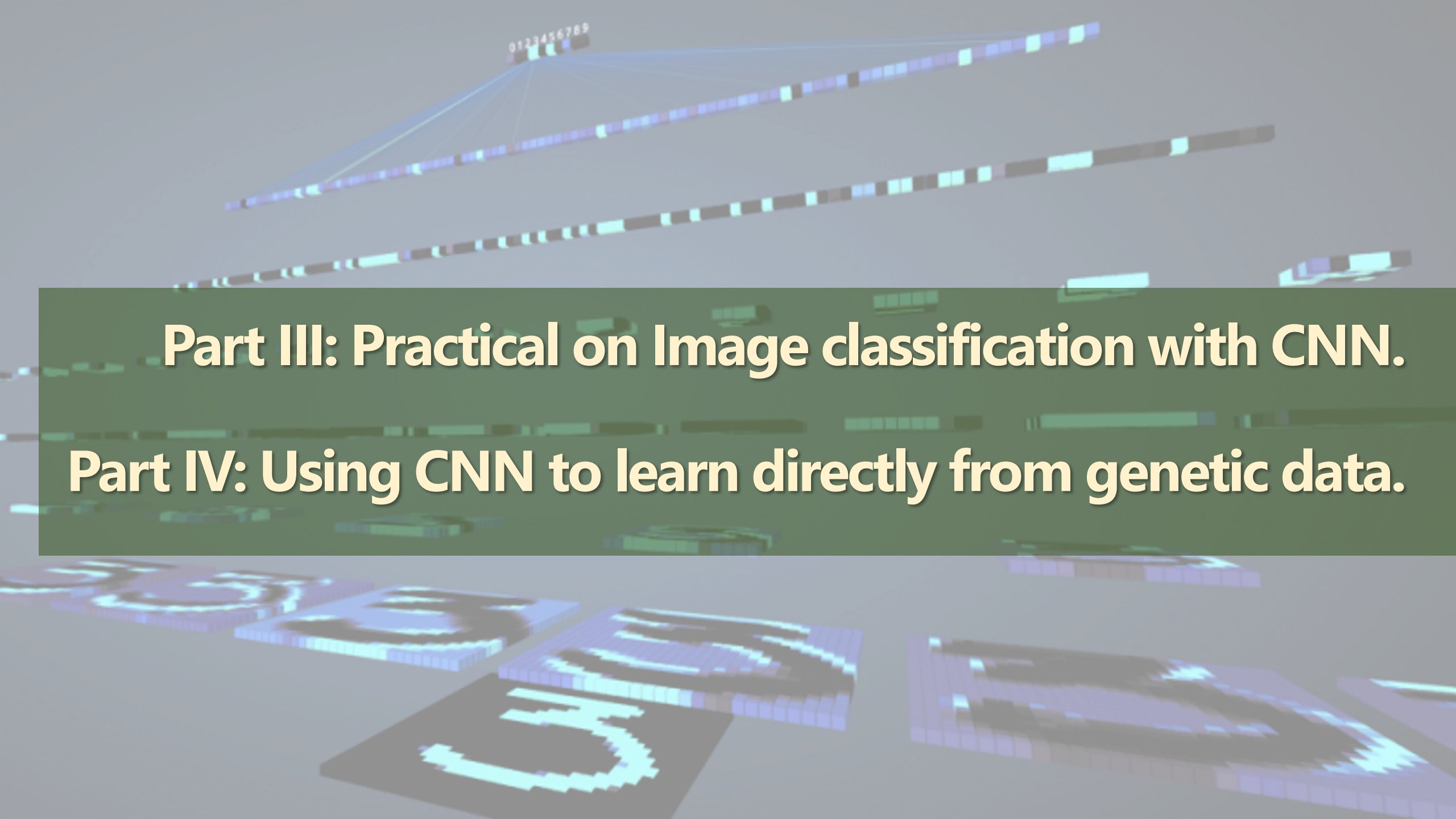
https://se-education.org/learningresources/contents/ai/ml.html







https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd



Part III: Practical on Image classification with CNN.

Part IV: Using CNN to learn directly from genetic data.

- **Practical Exercise 2:**

- Access https://poloclub.github.io/cnn-explainer/

- Scroll down to the Understanding hyperparameters section. Try to understand what is kernel size, padding, and stride.

- Use the graphical interface to try to better understand how convolutions work and what features are being extracted from the images.

- You can also upload the alignment images from different scenarios (but notice the network was not trained with that type of image). Does the network give an output prediction for the alignments?