Machine Learning from scratch

Manolo Fernandez Perez
Departent of Life Sciences, Imperial College London

@ManoloLearning
manolofperez@gmail.com
sites.google.com/site/manolofperez

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

- Use CNN to detect regions with selective sweeps on real genomes

Program

Part I: A gentle introduction to supervised deep learning.

Credit for some slides:
Matteo Fumagalli and Flora Jay

The Golem of Prague

Statistical Rethinking 2023

https://github.com/rmcelreath/stat_rethinking_2023

https://trips-tickets.com/the_golem_of_prague/

https://www.atlantajewishtimes.com/the-golem-of-prague/

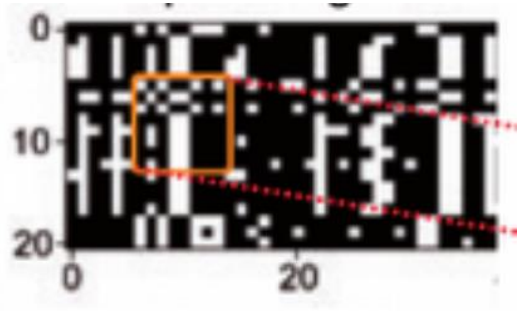
What is machine learning?

TASK:
predict y from x

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

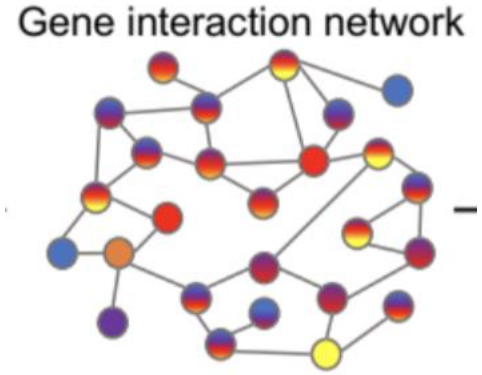
Matteo Fumagalli and Flora Jay. Machine learning and deep learning for demographic and selection inference. Lecture

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/

What is the model?

Unsupervised / Supervised Tasks

- Learning something from **data**
- Either **unsupervised** (no labels) or **supervised** (discrete or continuous labels)

- What's a label ? a target class (discrete) or a target value (continuous) observed for each sample

Data are not always labeled. They can also have multiclass labels

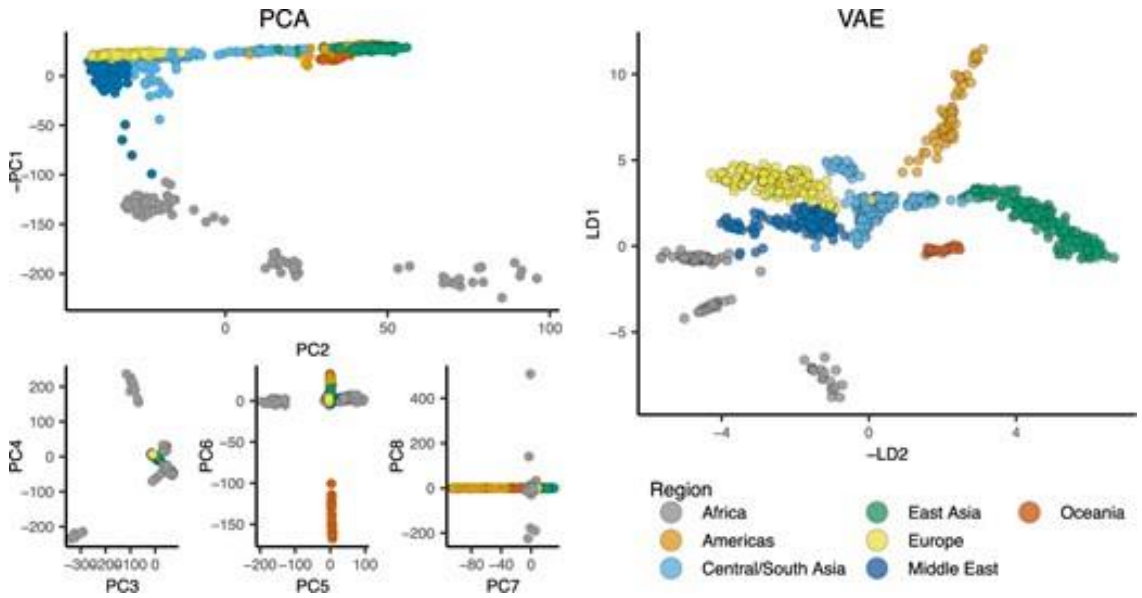
ex : pic of dog/person/car..., price of house, level of cholesterol

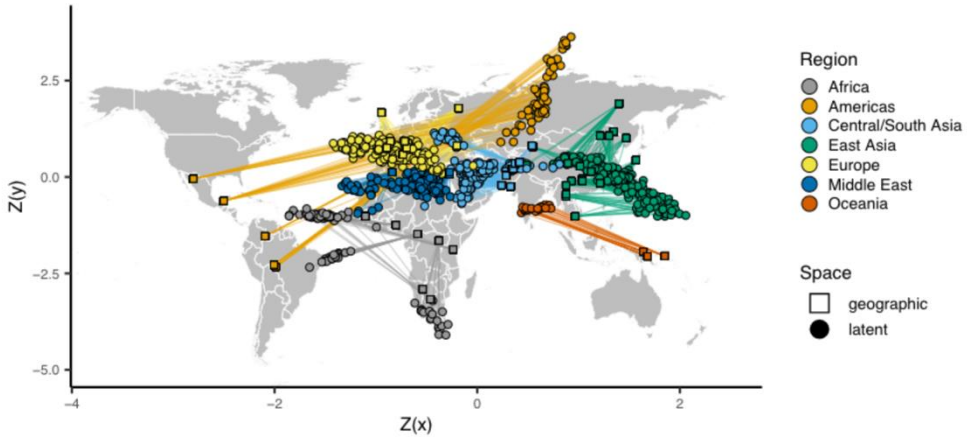
What is the model?

.Unsupervised = discovering patterns in data without prior knowledge

You do NOT have labels, or you do NOT use them

- Dimension reduction methods, e.g. PCA, Matrix factorization
- Clustering algorithms, e.g. K-means, hierarchical clustering, ...
- Outlier detection (can be then used for filtering, ..)
- ...





Battey et al. (2021) *G3*

What is the model?

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

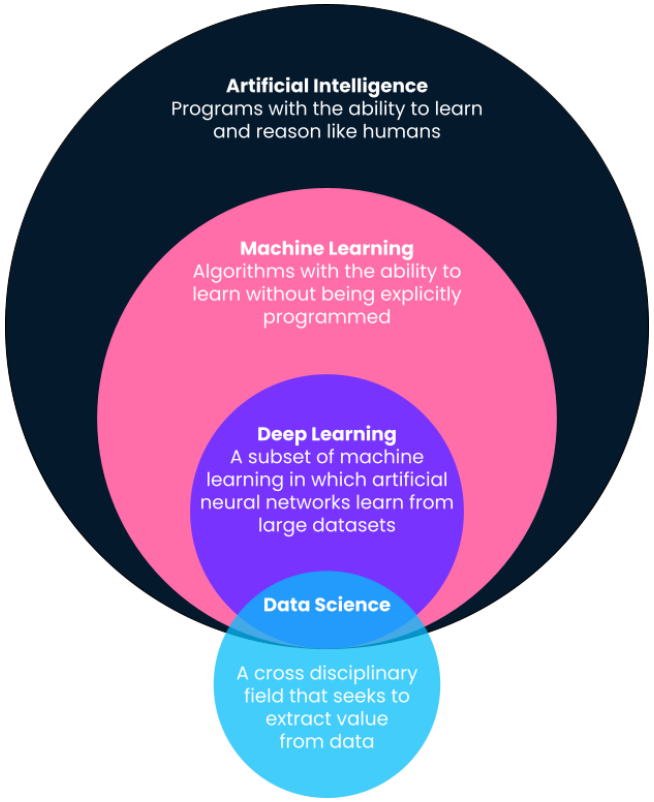
Classification (predict a class)

Regression (predict a variable)

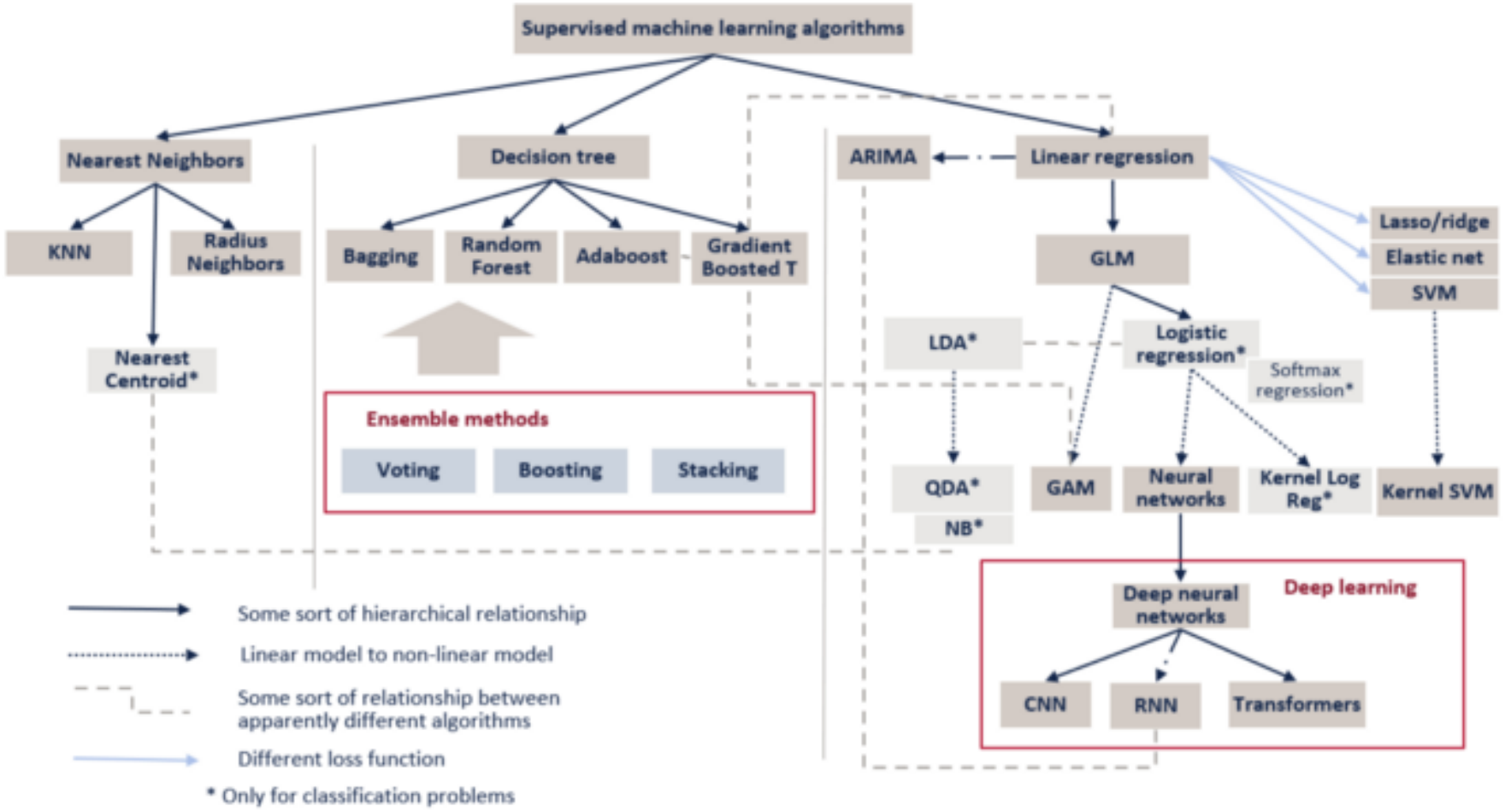
What for:
- Predict labels of new unlabeled samples (eg what's on an image?),
- Understand better the relationship between features and the label (eg understand which set of genes allow to predict a disease risk),
- ...

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

Deep Neural Networks

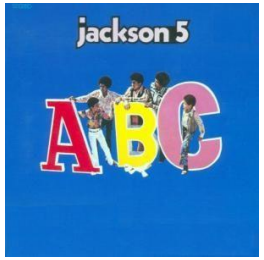


https://www.datacamp.com/blog/what-is-machine-learning



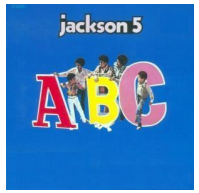
https://towardsdatascience.com/overview-of-supervised-machine-learning-algorithms-a5107d036296

Why Deep Learning?

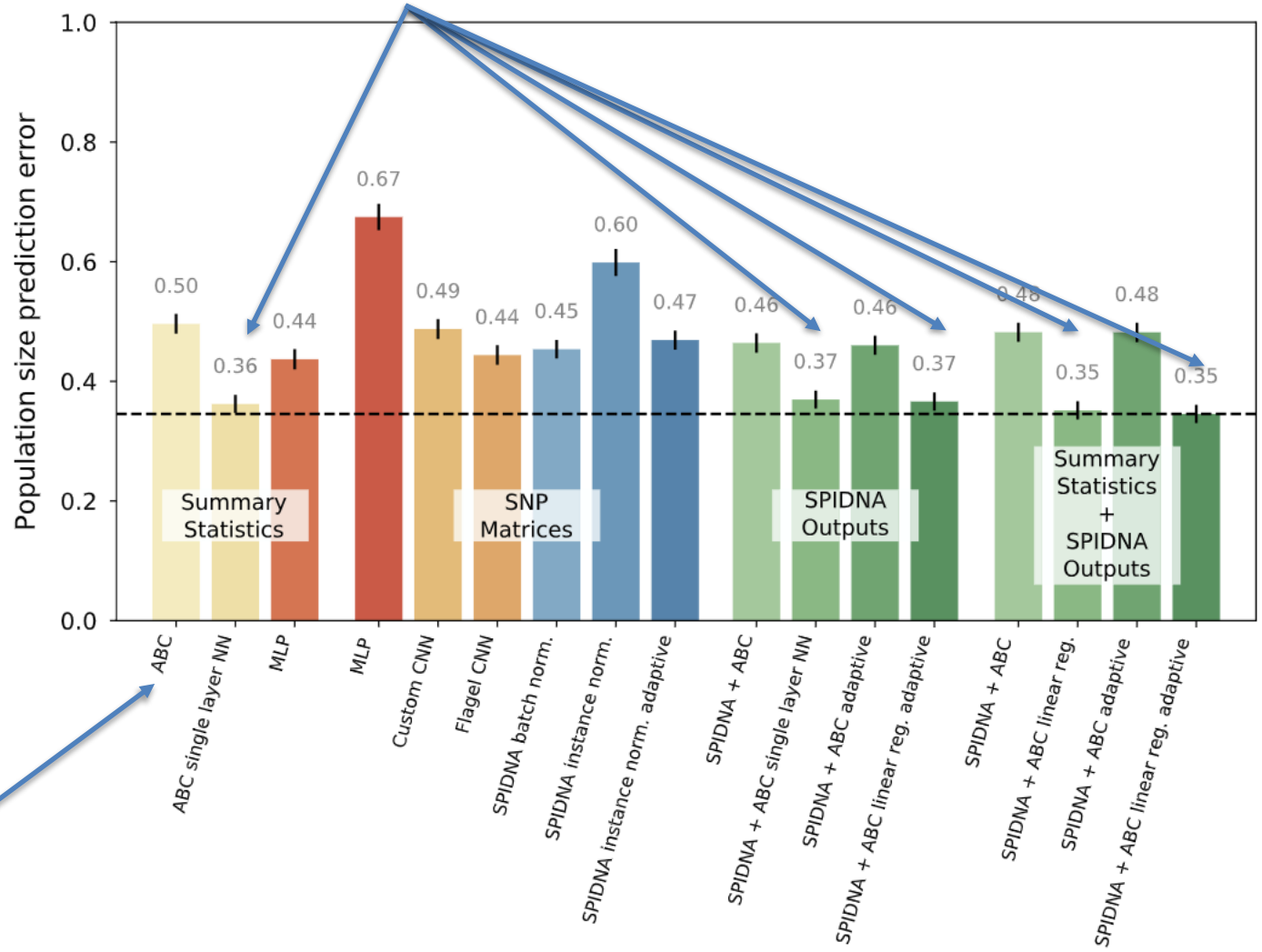


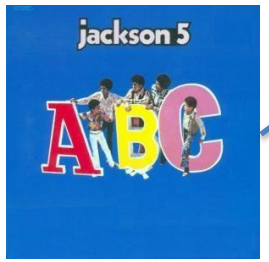
Deep Learning

CJ Battey
Twitter 2018



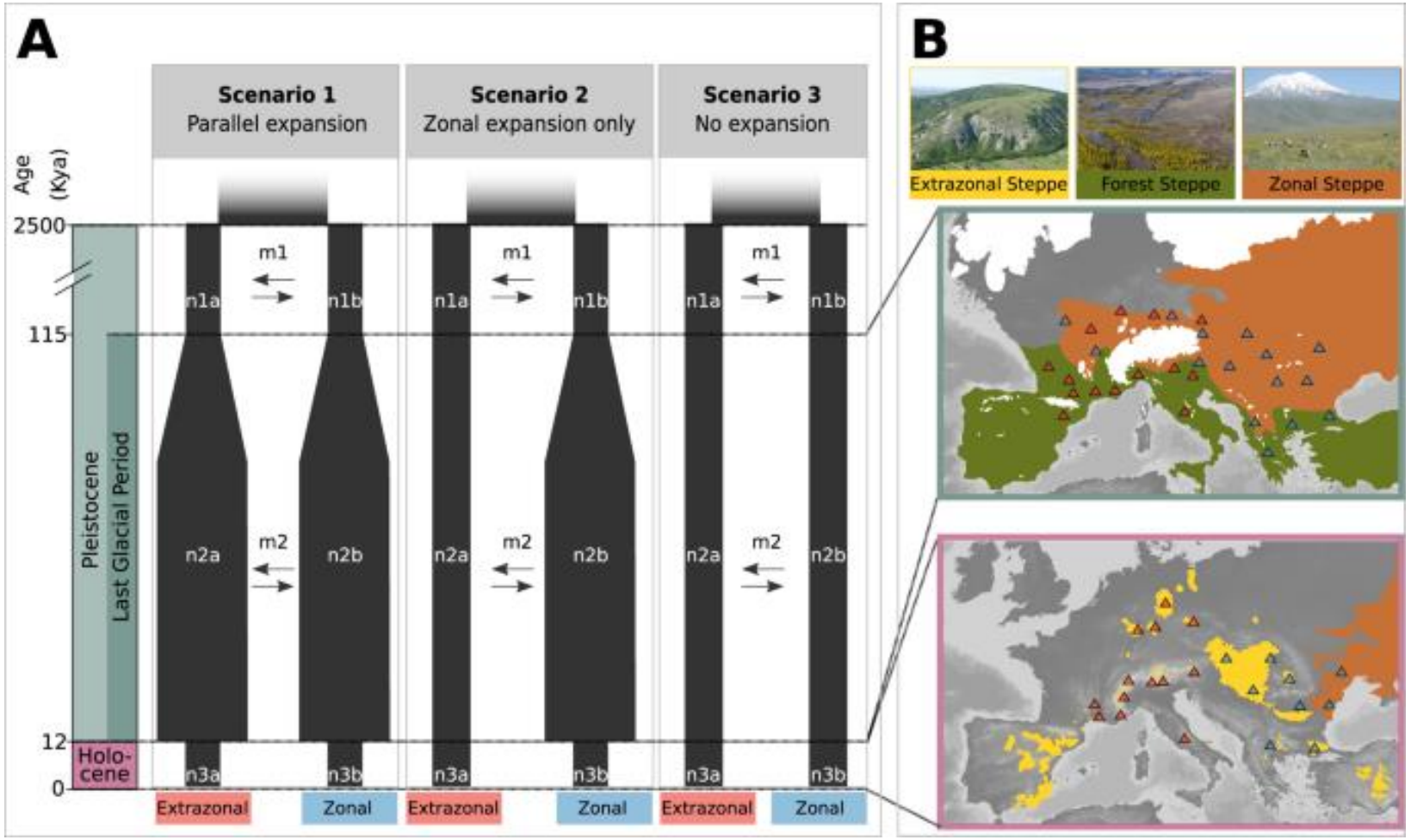
Why Deep Learning?





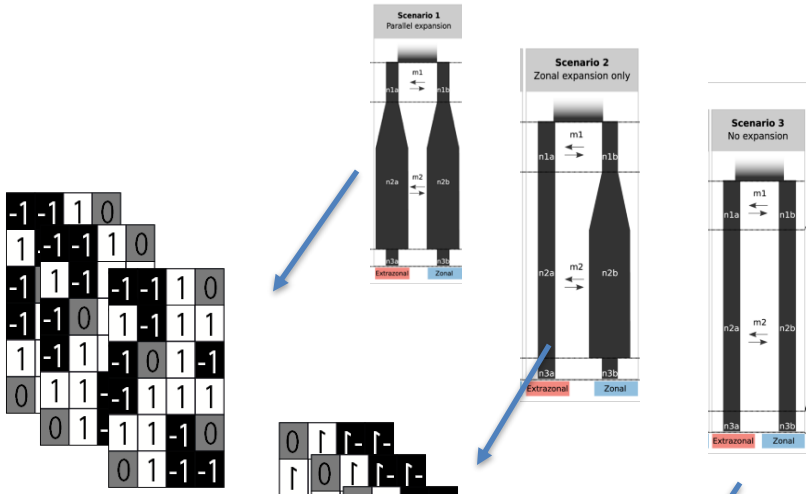
Sanchez et al. (2020) Mol Ecol Res

Supervised Deep Learning ~ ABC



Kirschner et al. (2022) *Nat Comm*

Supervised Deep learning ~ ABC



Scenario 1

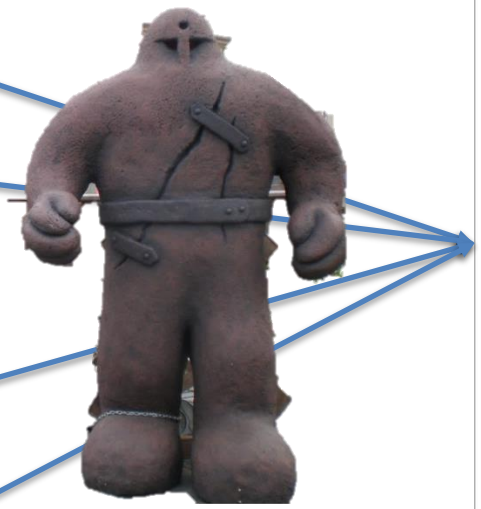
Scenario 2

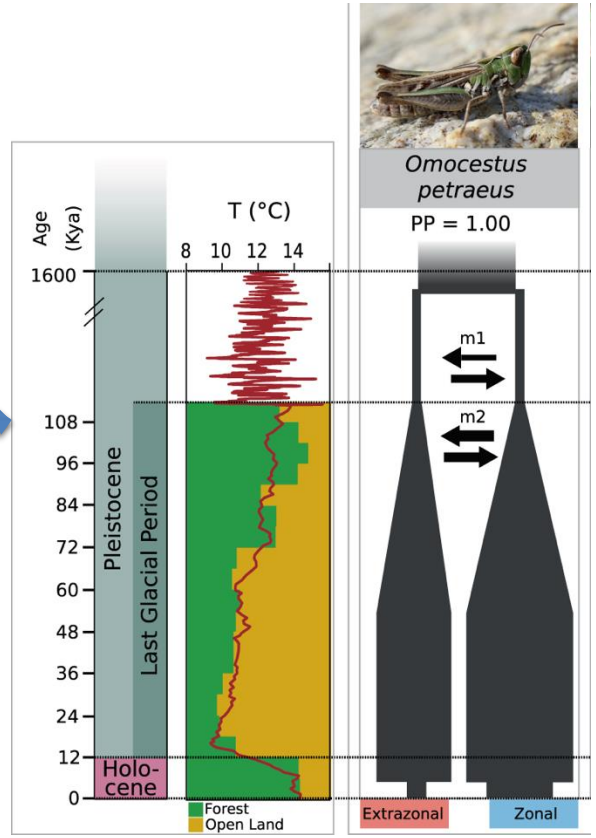
Scenario 3

Data is labeled

Parameters

	θ	T1	T2	T3	Ne
Sim1					
Sim2					
Sim3					
Sim4					





1. Define Models

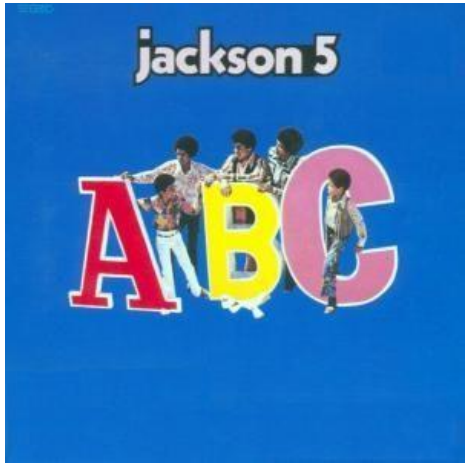
2. Sample parameters from a given distributions (prior)

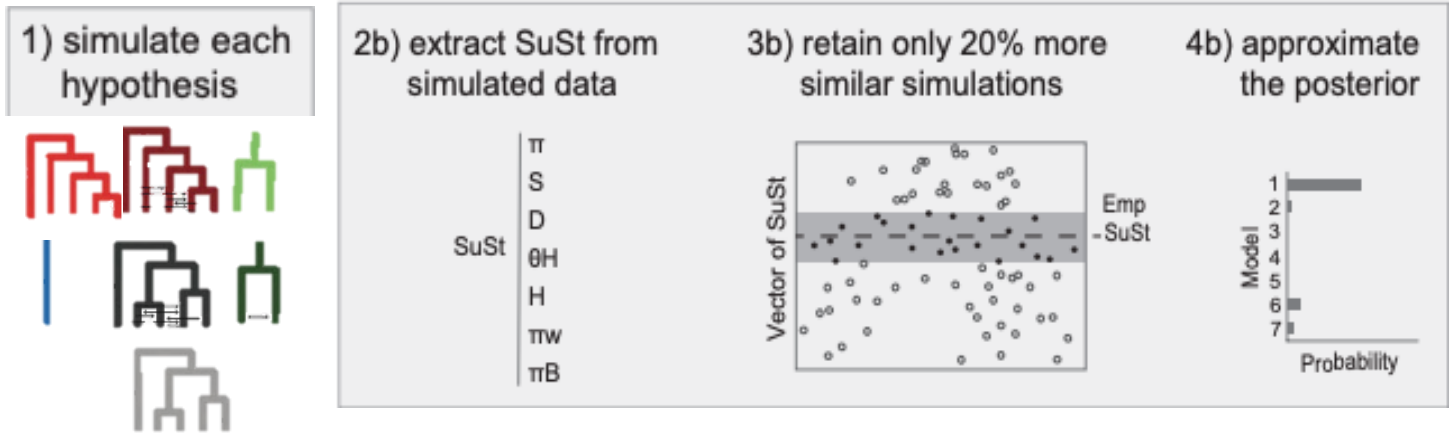
3. Simulate data for each model using the sampled parameter values

4. Calculate Summary Statistics (SuSt) from simulations and from the empirical data

5. Compare simulated and empirical data retaining only simulations that are more similar.

ABC





Modified from
Perez et al. (2022) *Mol Ecol Res*

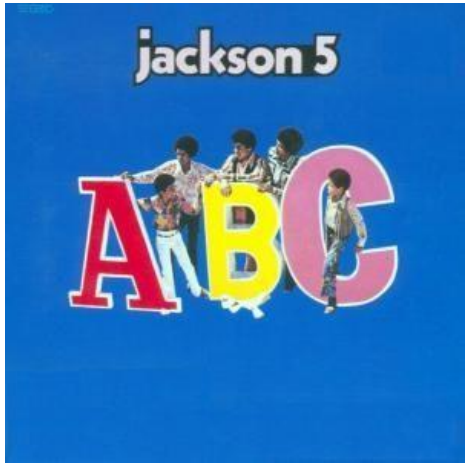
ABC

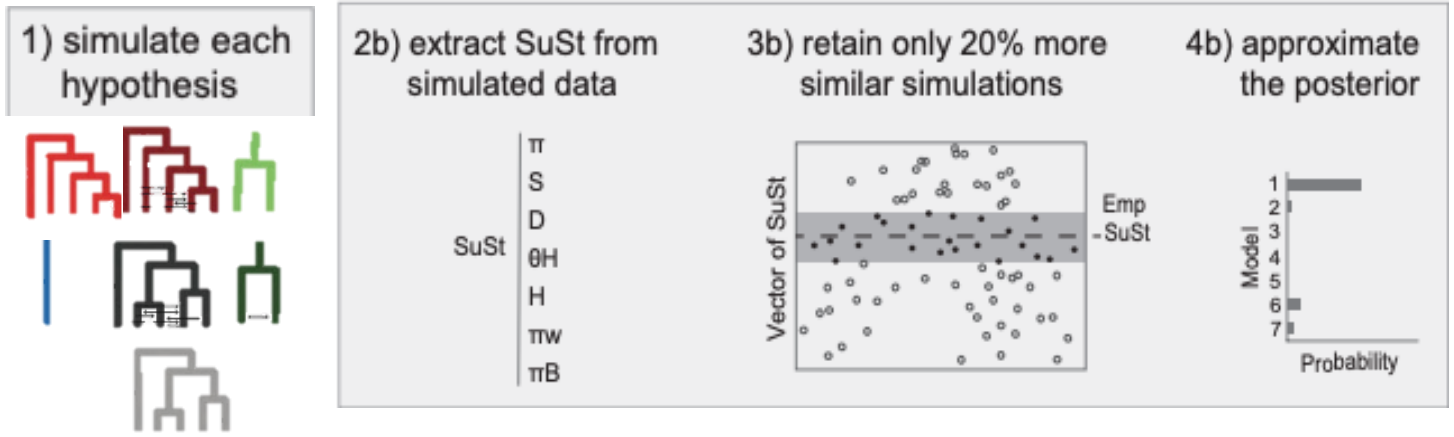
ABC implies 3 **approximations**:

1. finite # simulations

2. informativeness of S

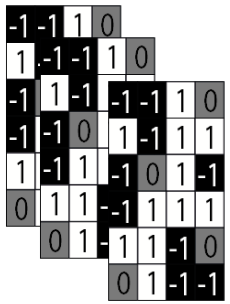
3. S don't match S* exactly



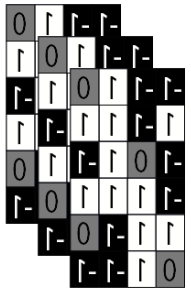


Modified from
Perez et al. (2022) *Mol Ecol Res*

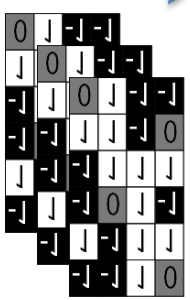
Supervised Deep Learning ~ ABC



Scenario 1

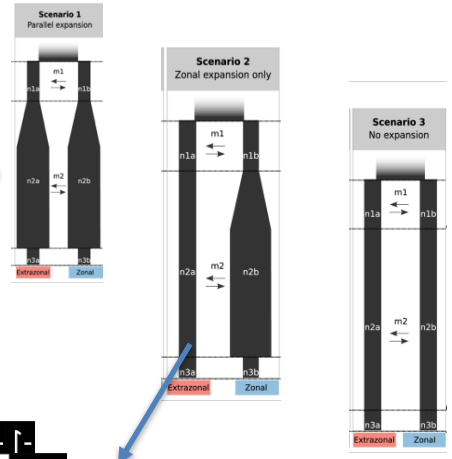


Scenario 2



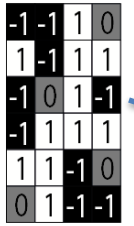
Scenario 3

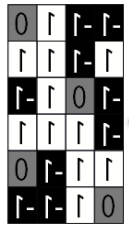
Data is labeled

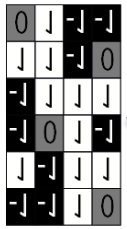


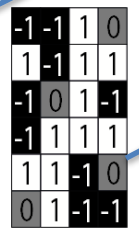
Parameters

	θ	T1	T2	T3	Ne
Sim1					
Sim2					
Sim3					
Sim4					









DL witchcraft

Cuca – Brazilian Legend

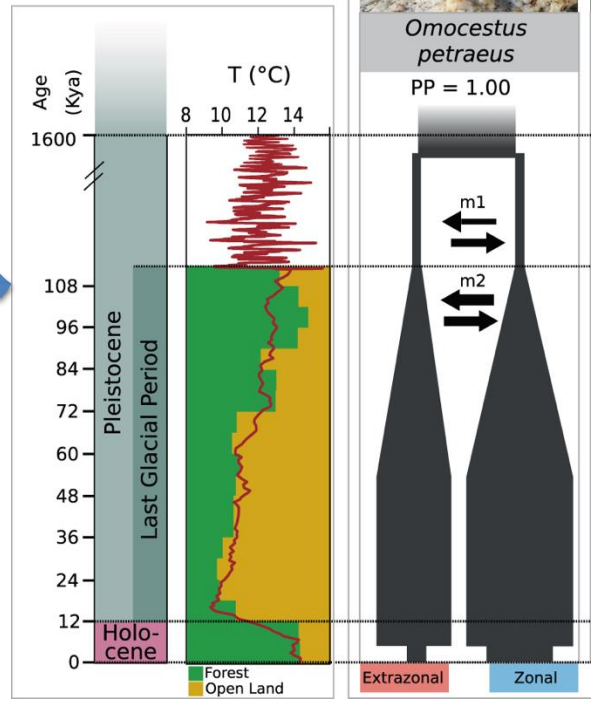
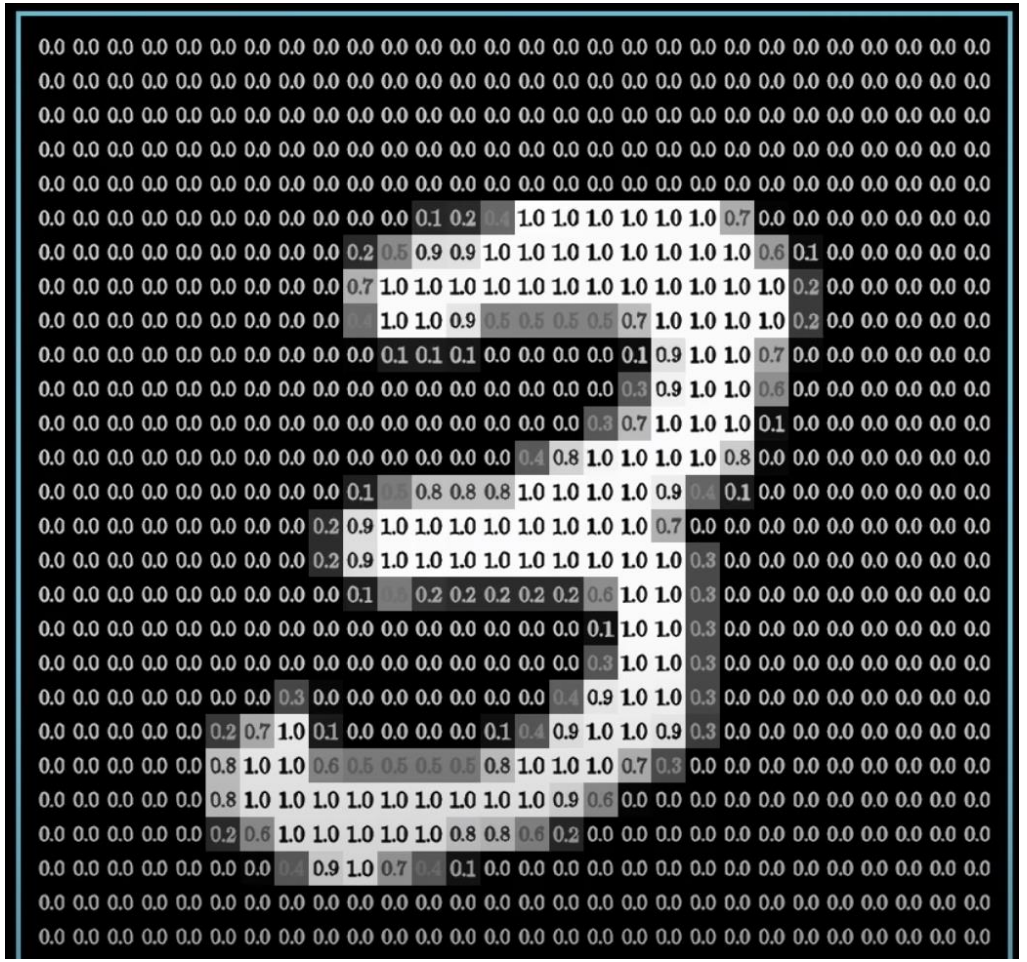


Image recognition

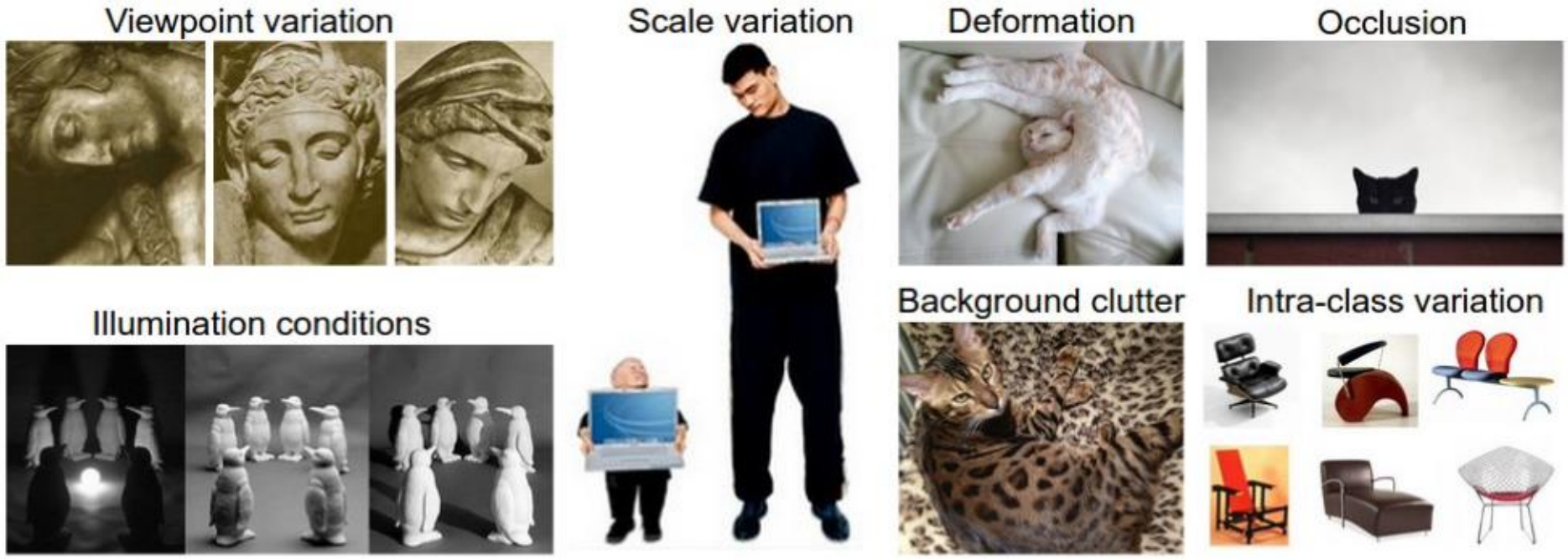
What do you see?

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

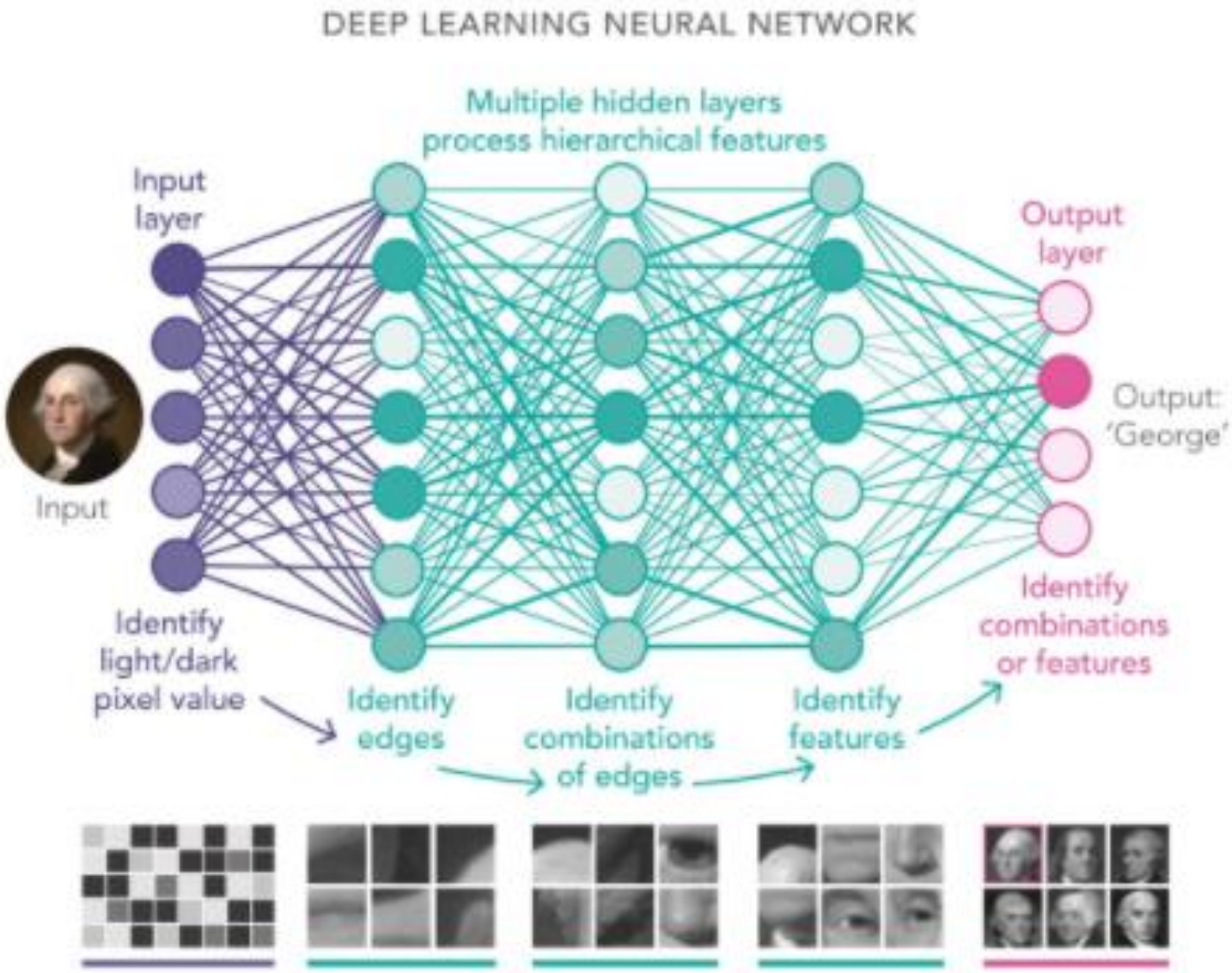
Data-Driven approach (need a large training set)

The MNIST dataset (70,000 hanwritten numbers)

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

Deep Neural Networks

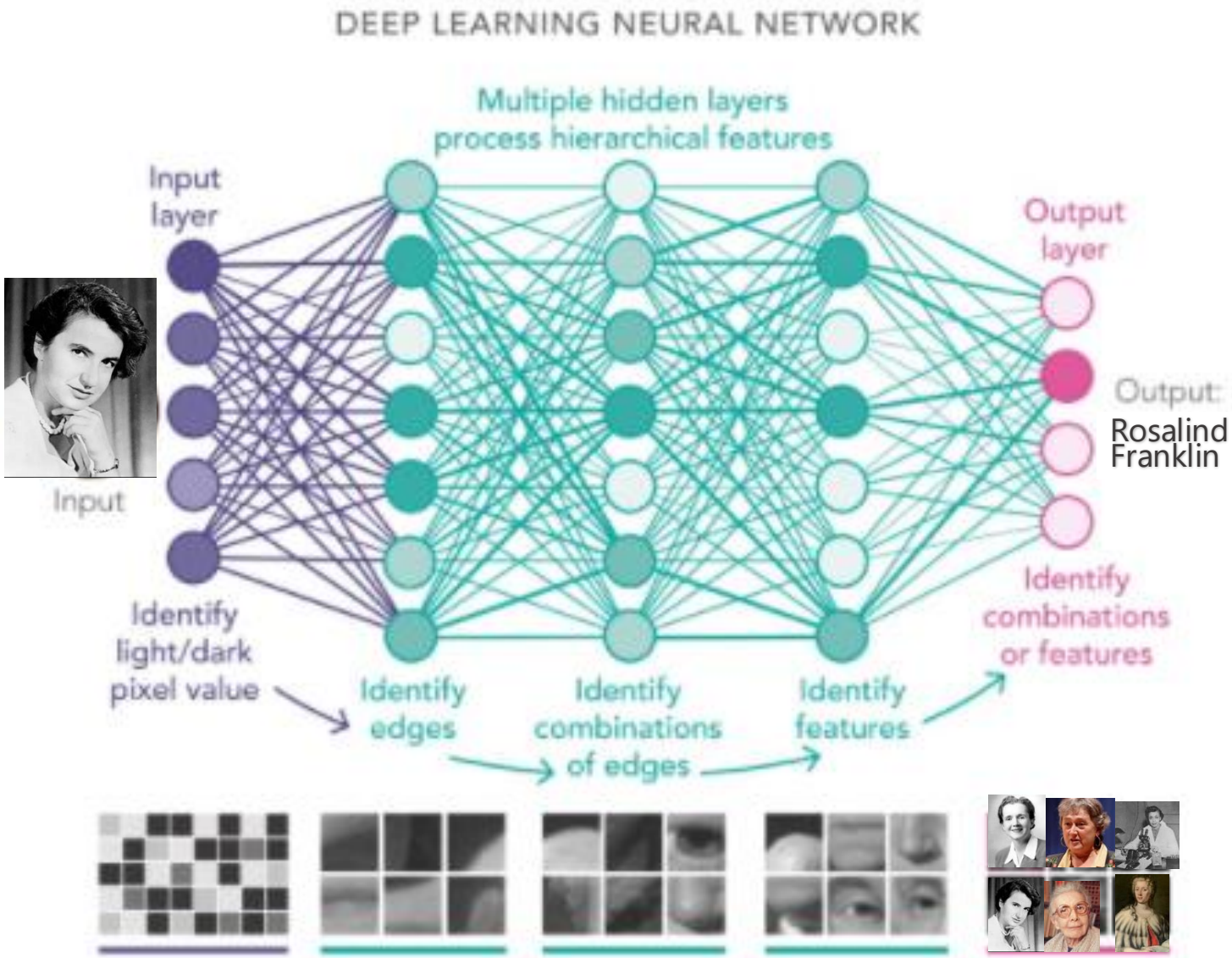
Multi-Layer Perceptron (MLP)



Waldrop (2019) *PNAS*

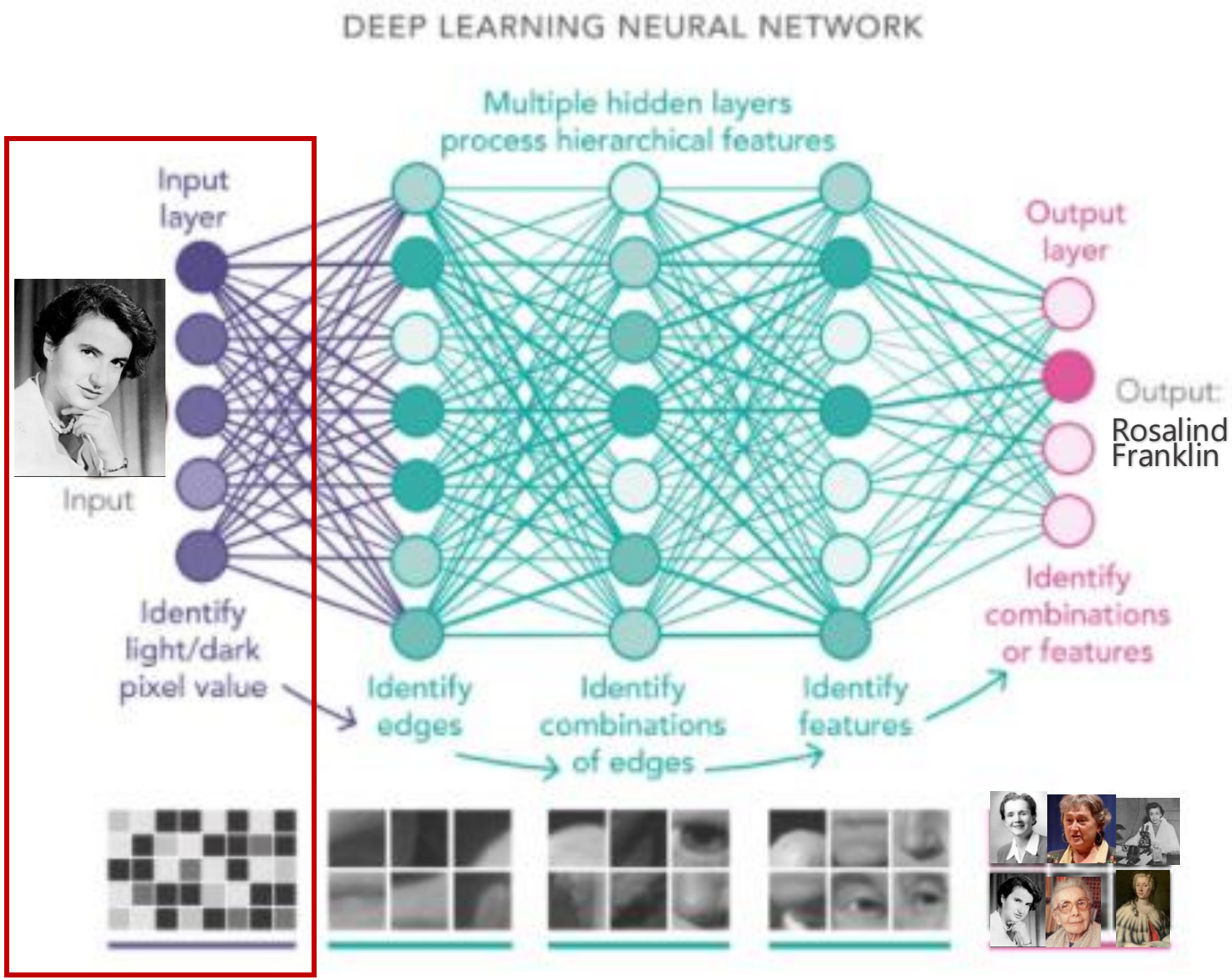
Deep Neural Networks

Multi-Layer Perceptron (MLP)



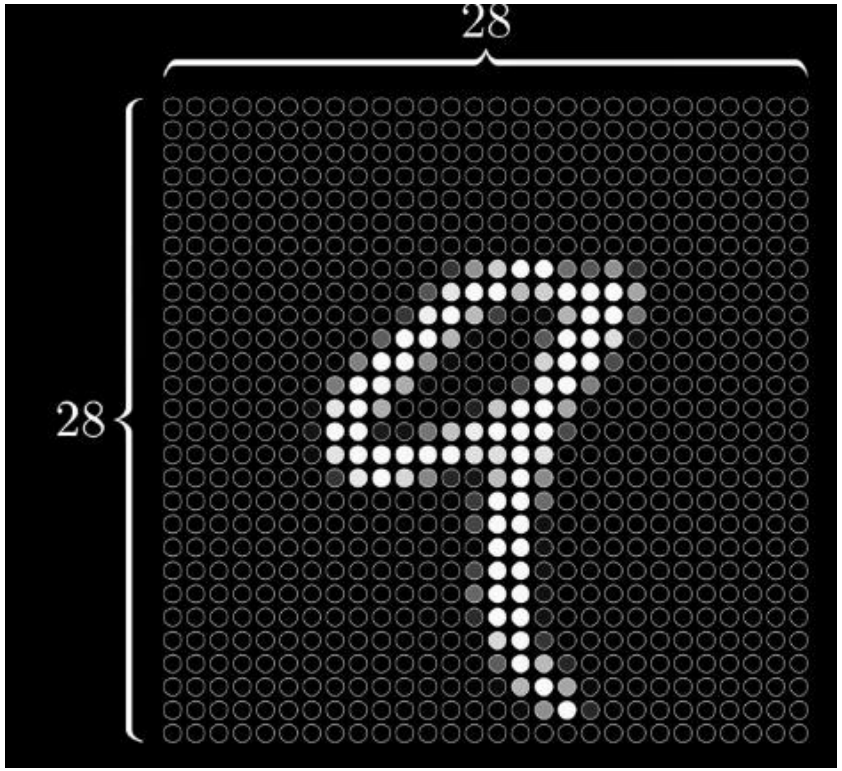
Adapted from Waldrop (2019)
PNAS

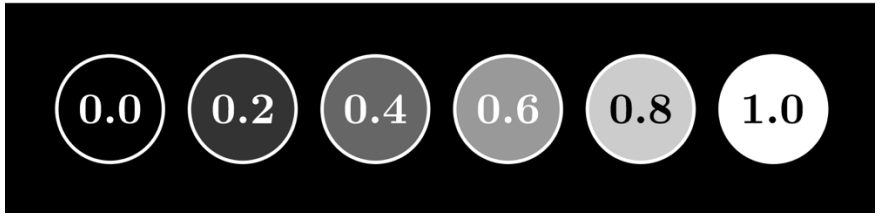
Deep Neural Networks

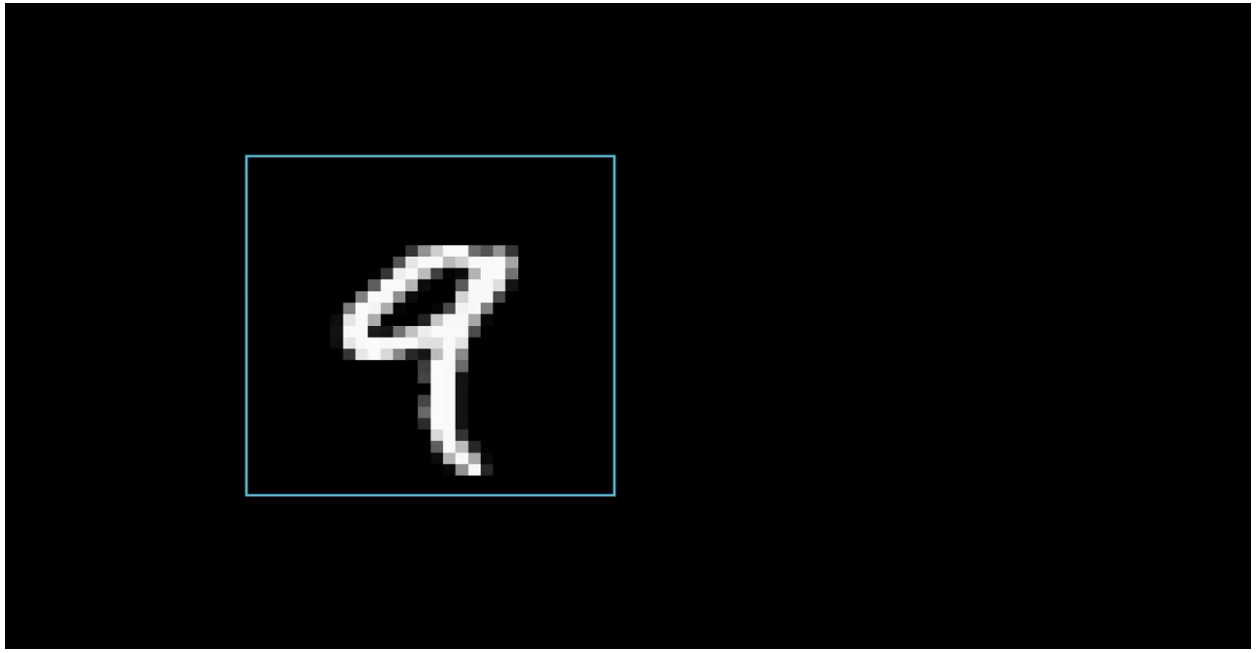


Adapted from Waldrop (2019)
PNAS

Input Layer

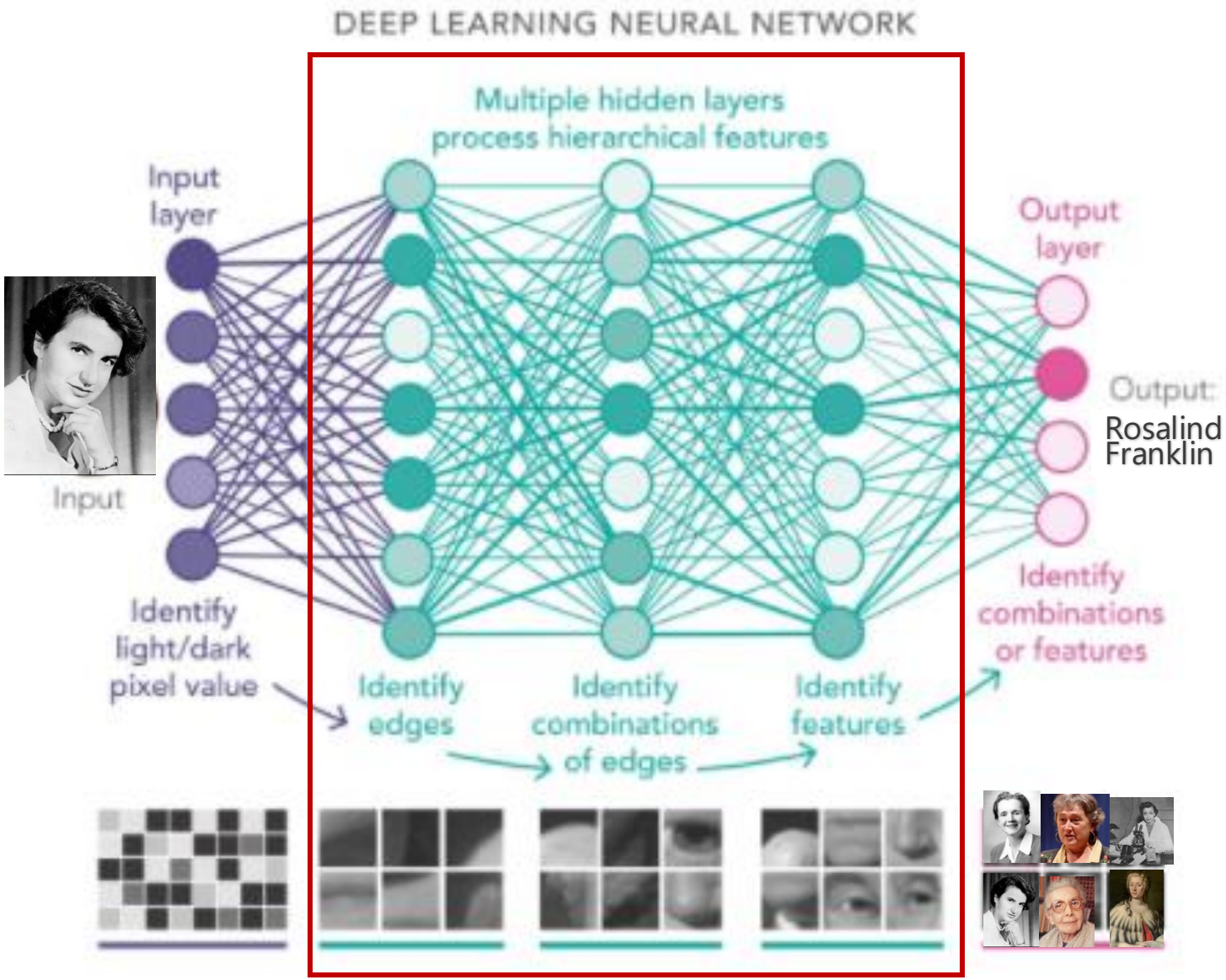






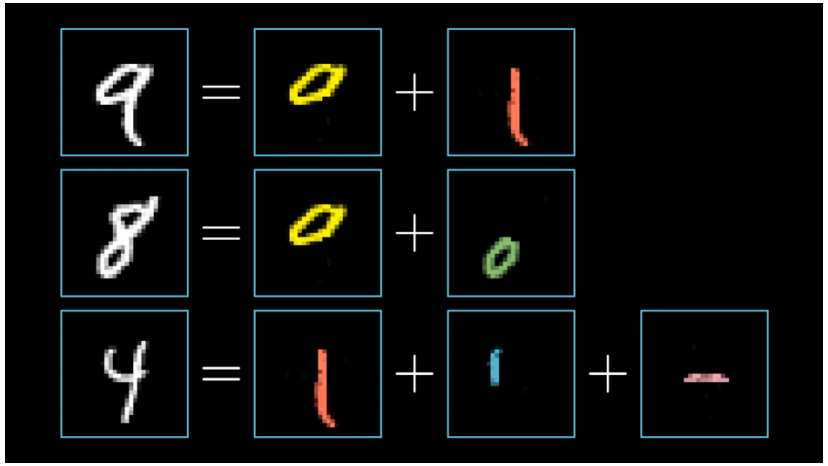
https://www.3blue1brown.com/lessons/neural-networks

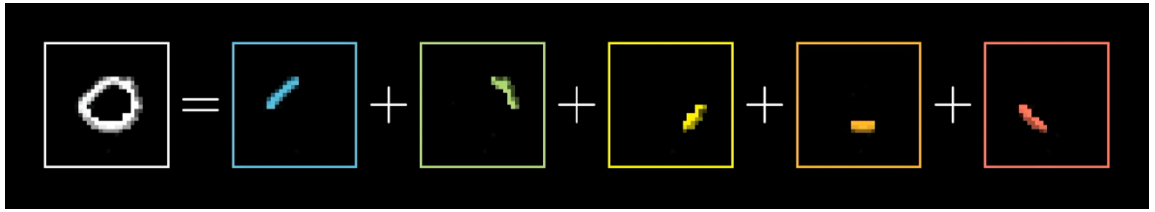
Deep Neural Networks

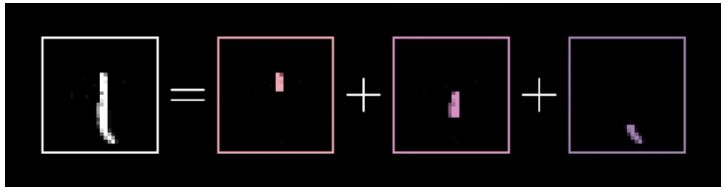


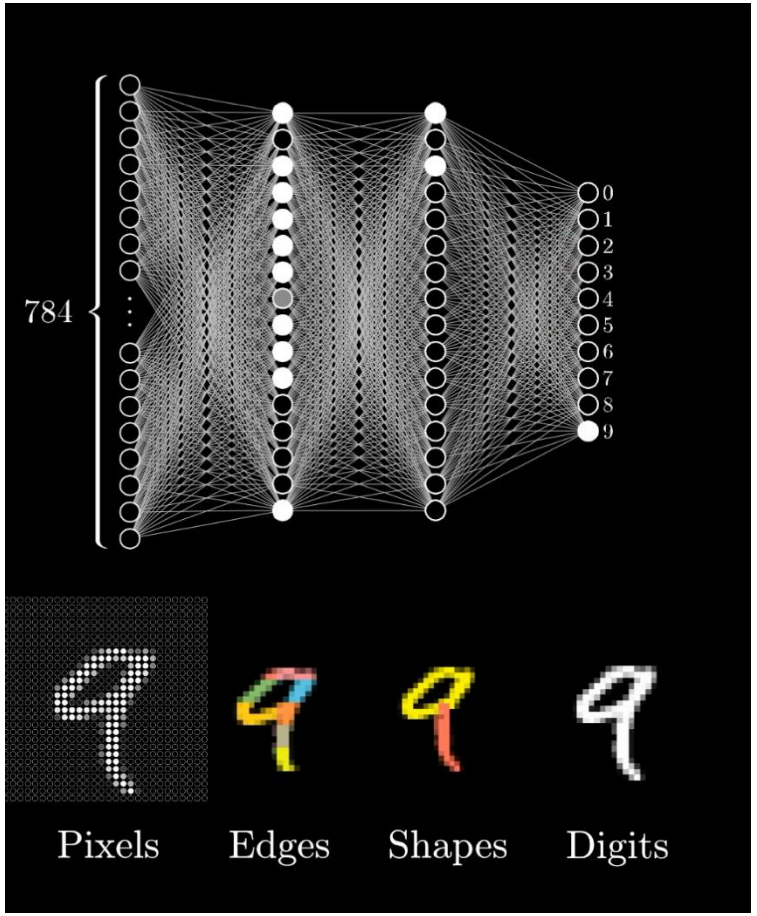
Adapted from Waldrop (2019)
PNAS

Hidden Layers process hierarchical features



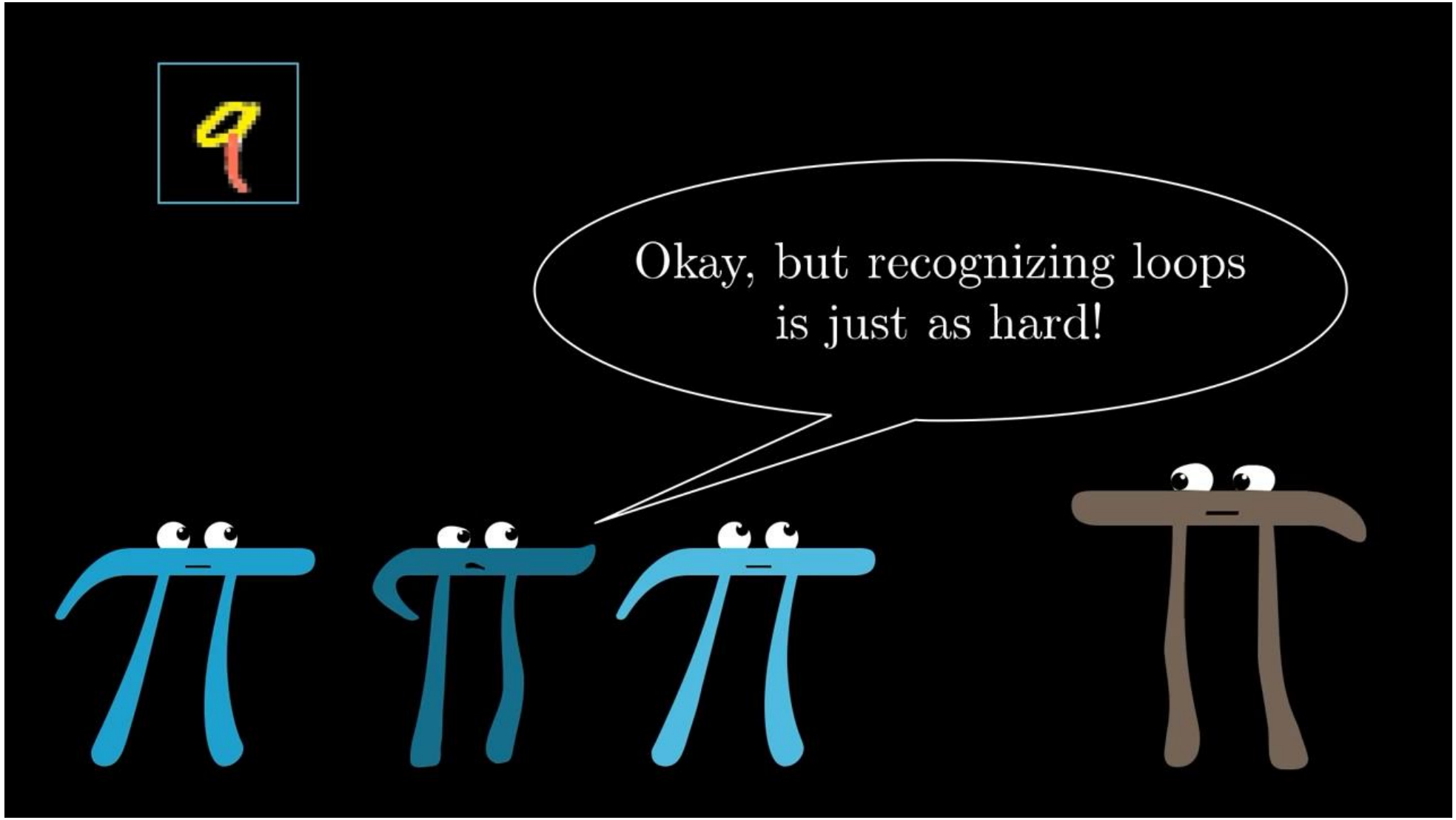






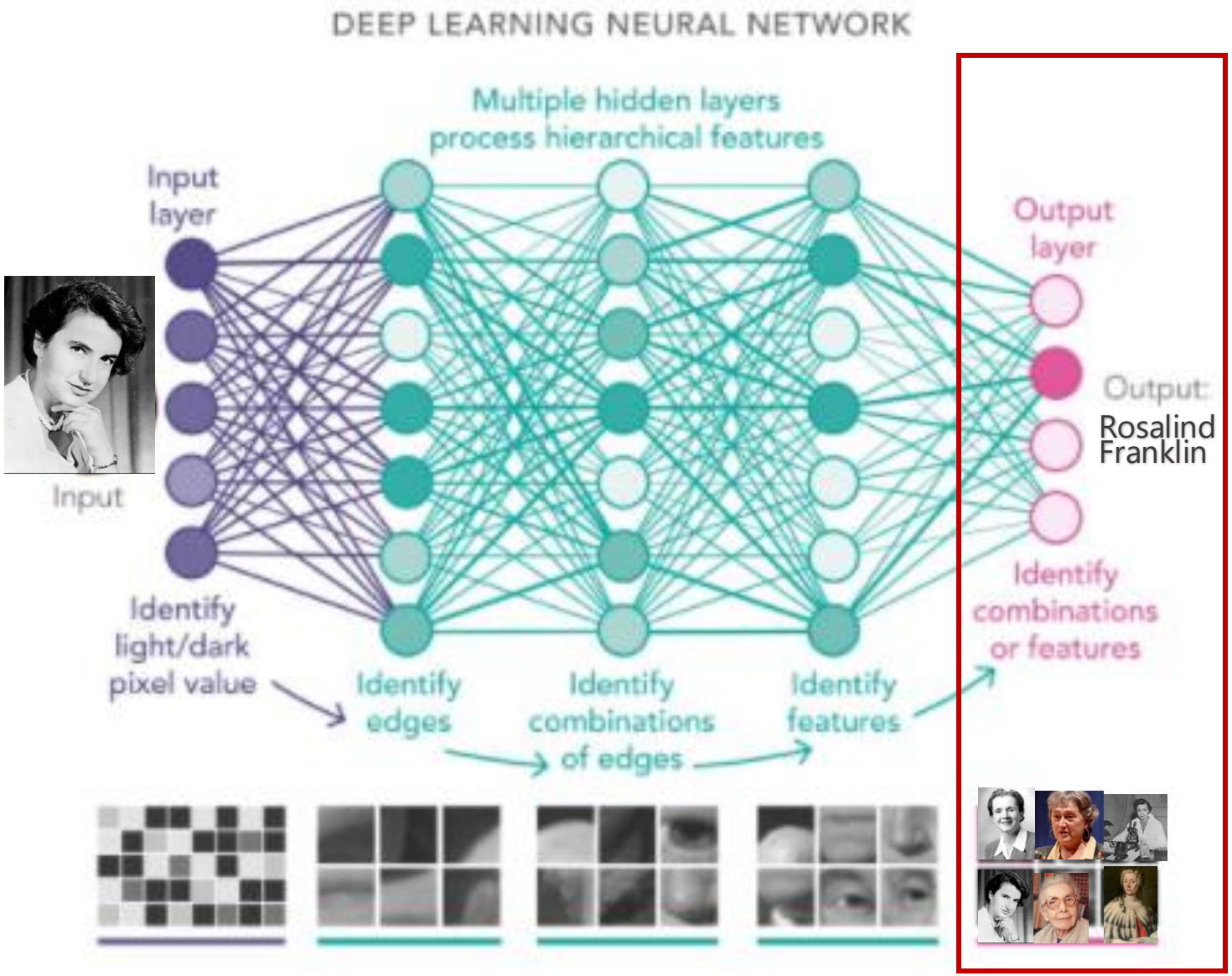
https://www.3blue1brown.com/lessons/neural-networks

Hidden Layers process hierarchical features



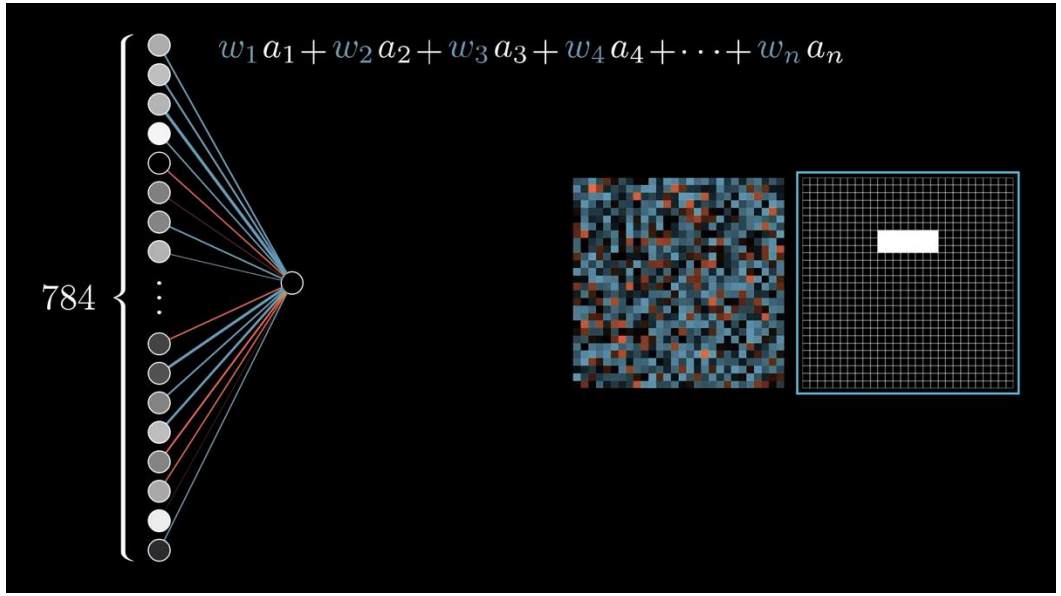
https://www.3blue1brown.com/lessons/neural-networks

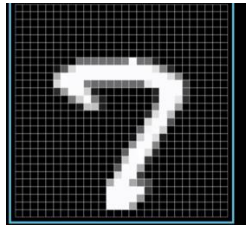
Deep Neural Networks



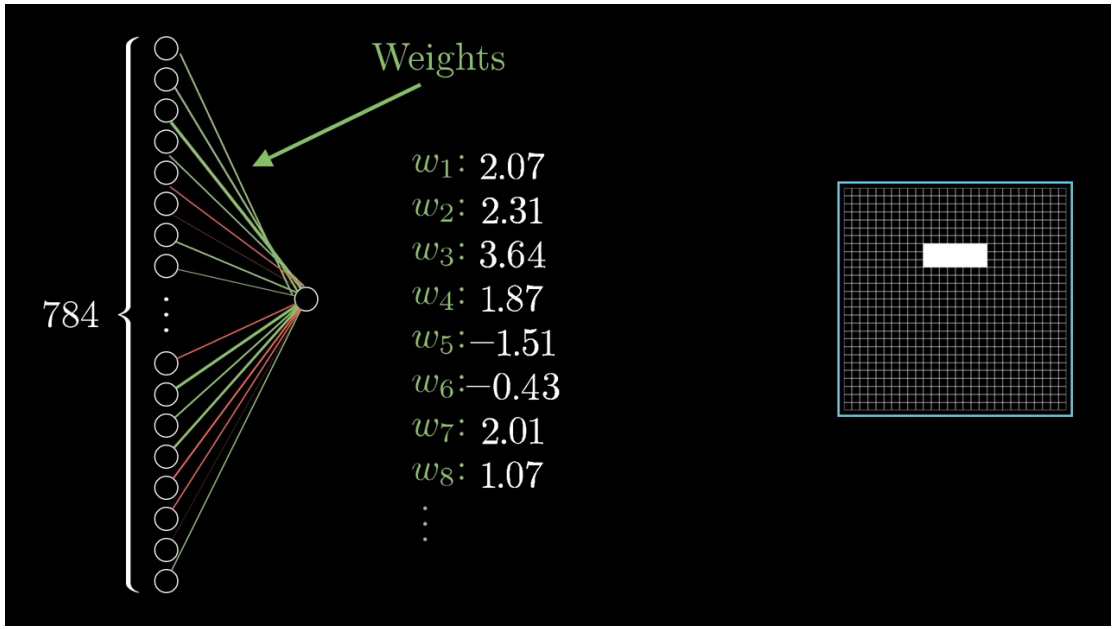
Adapted from Waldrop (2019)
PNAS

Network Weights



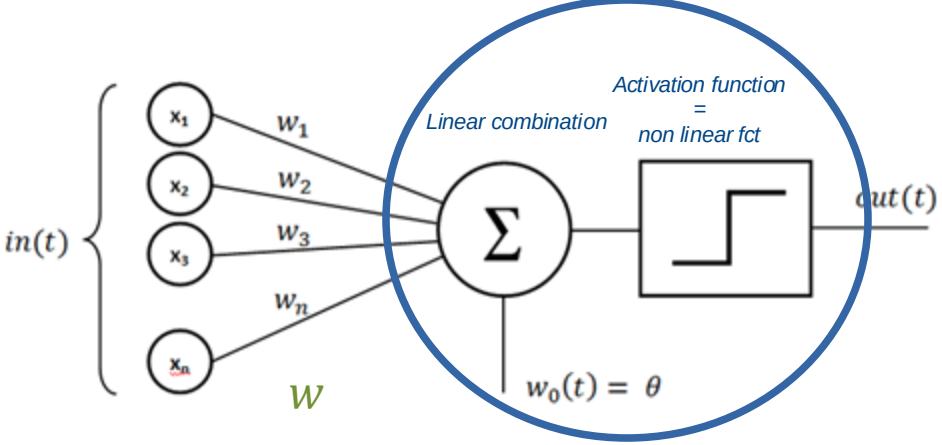


Neuron in the second layer to check whether the image contains this one, specific edge.



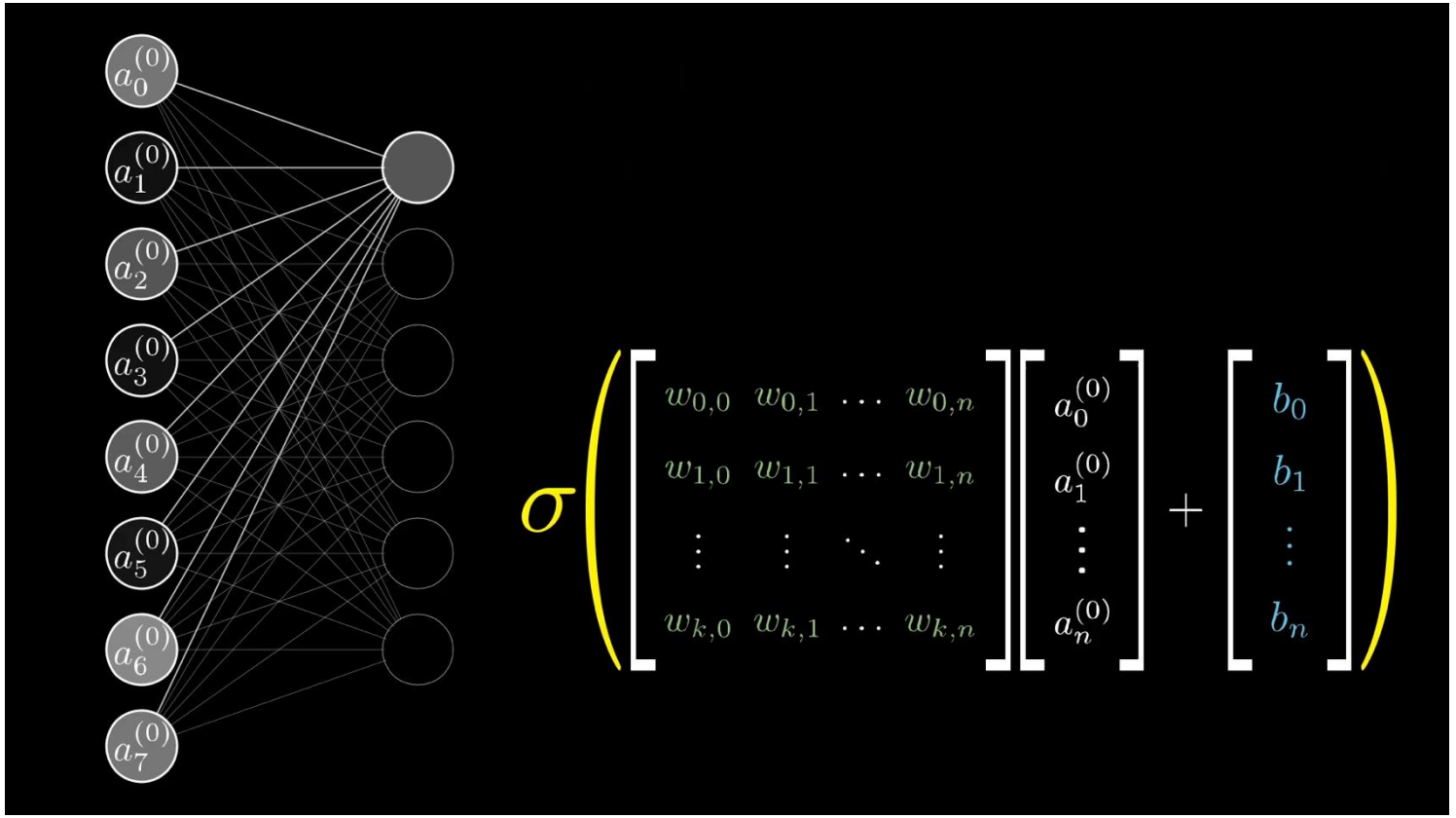
https://www.3blue1brown.com/lessons/neural-networks

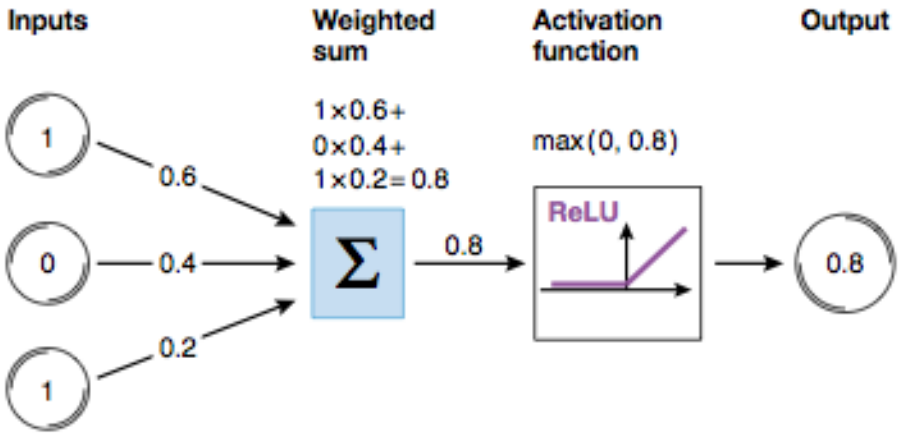
Network Weights



1 neuron = f (∑)
Linear comb. + Non Linear activation function f()

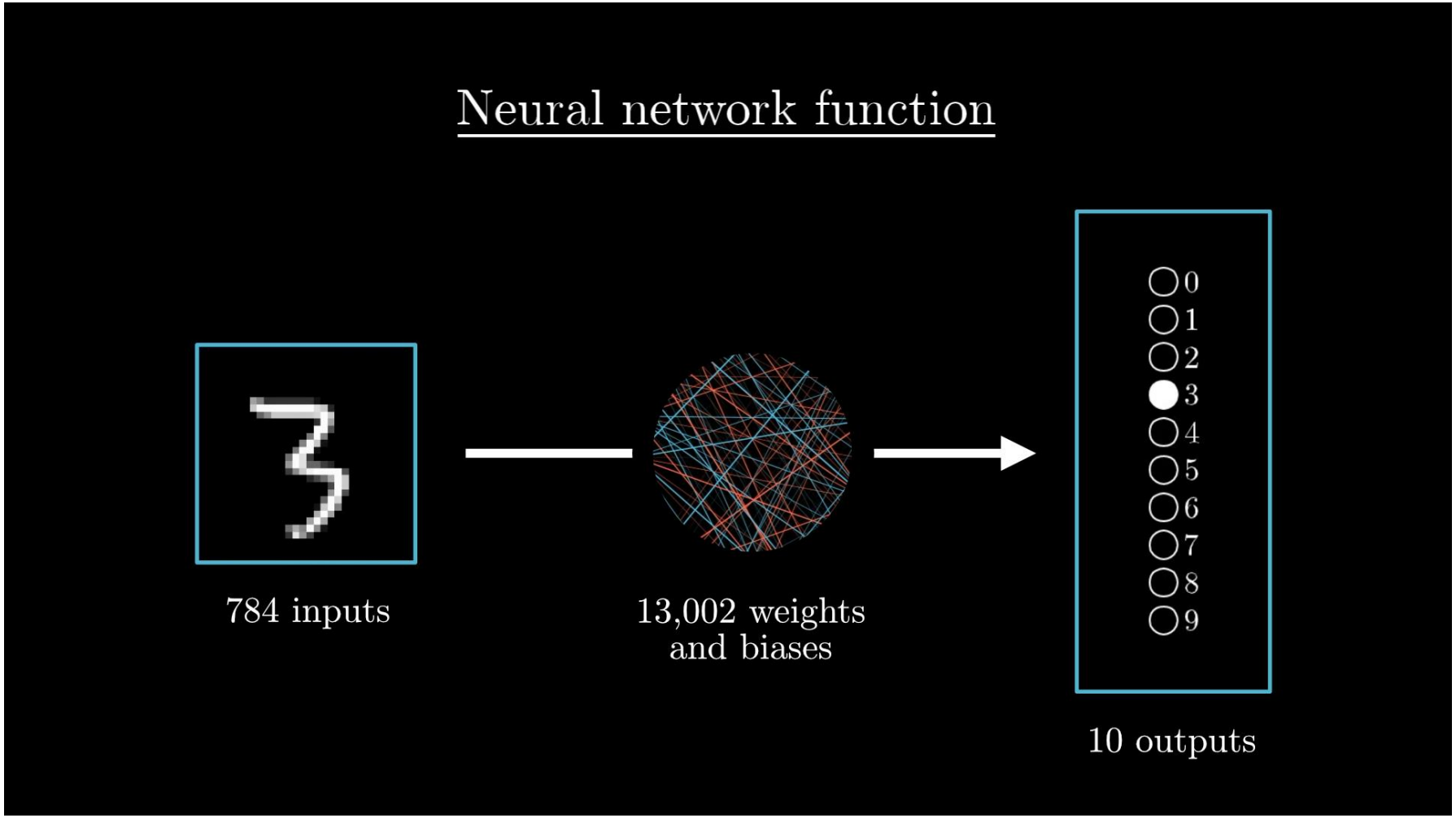
$$out = f(w_0 + w_1x_1 + w_2x_2 + \ldots)$$





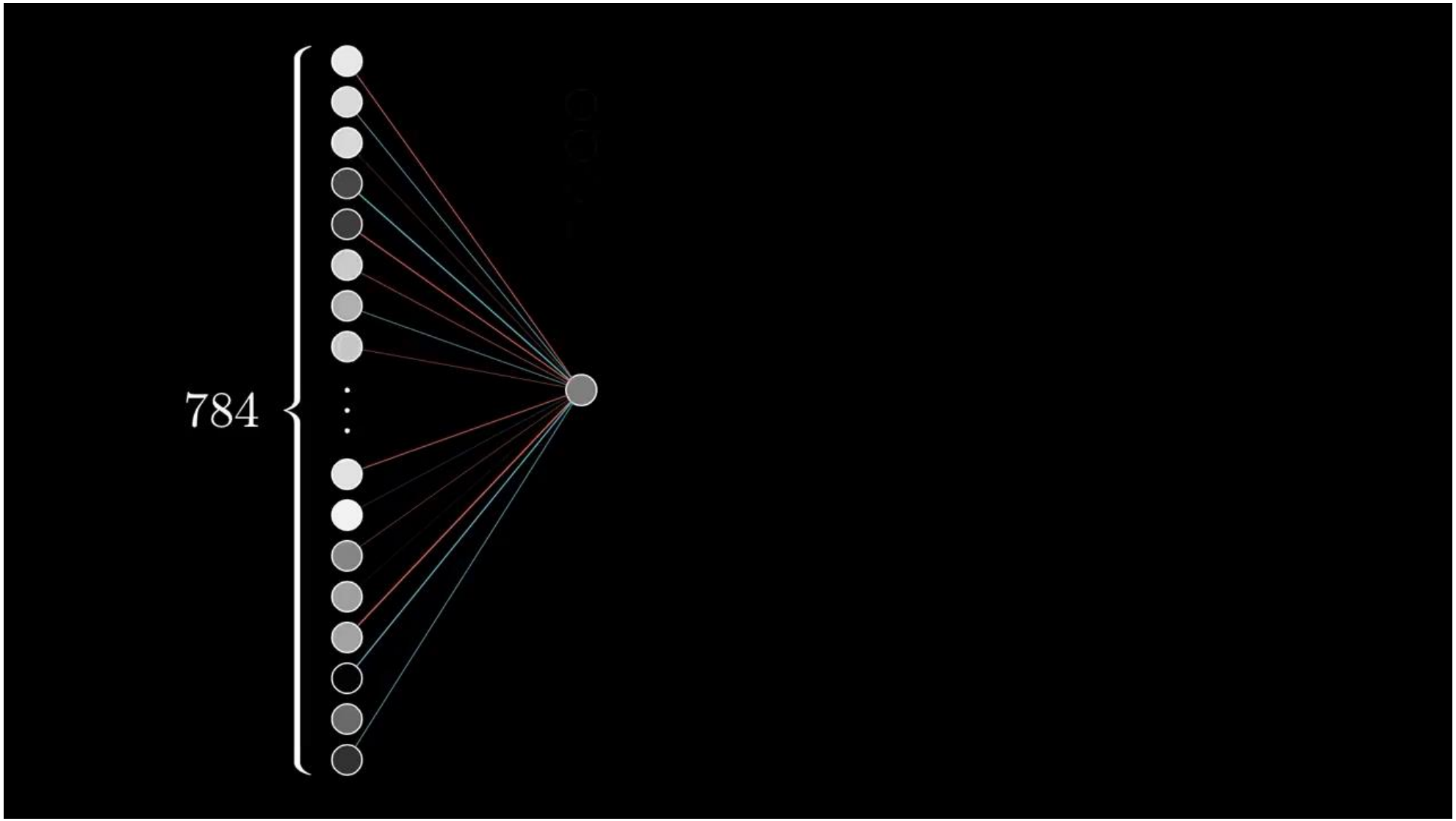
https://www.3blue1brown.com/lessons/neural-networks

Score function



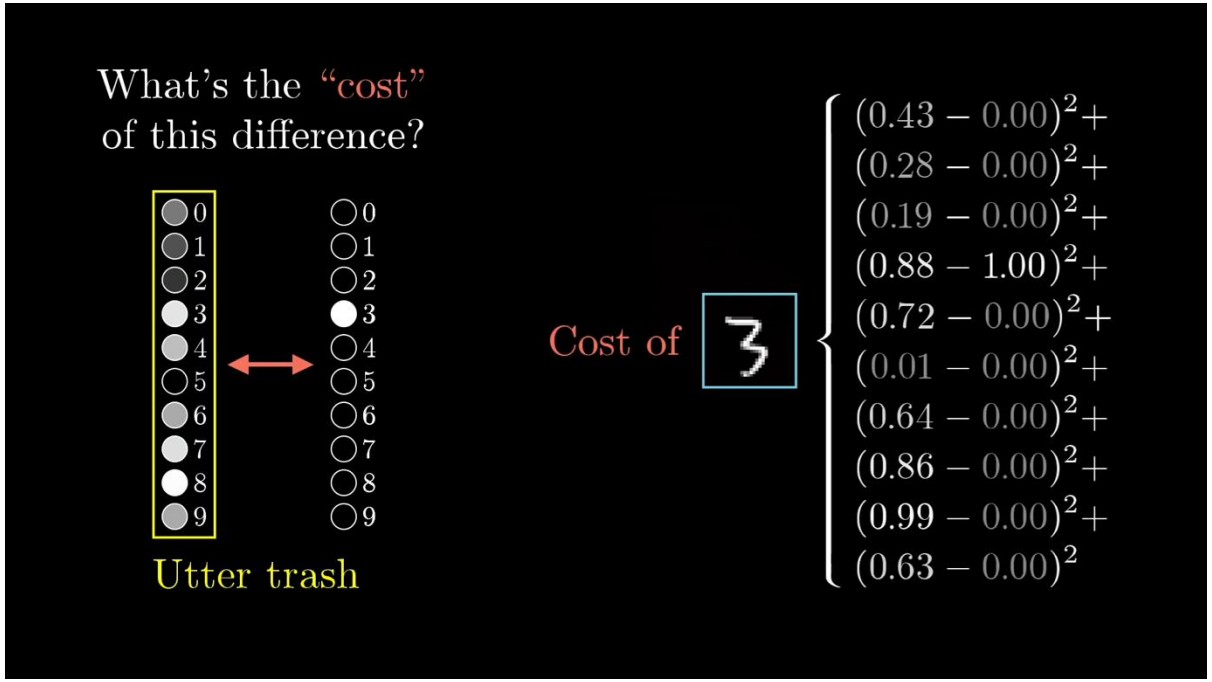
https://www.3blue1brown.com/lessons/neural-networks

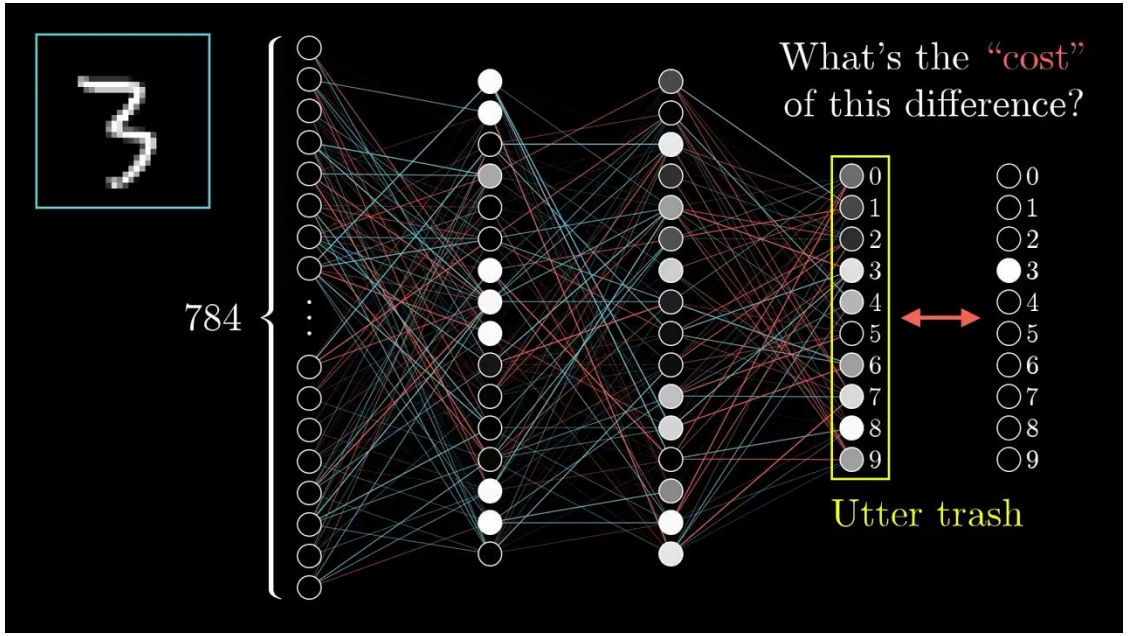
How the network learns? Cost function



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

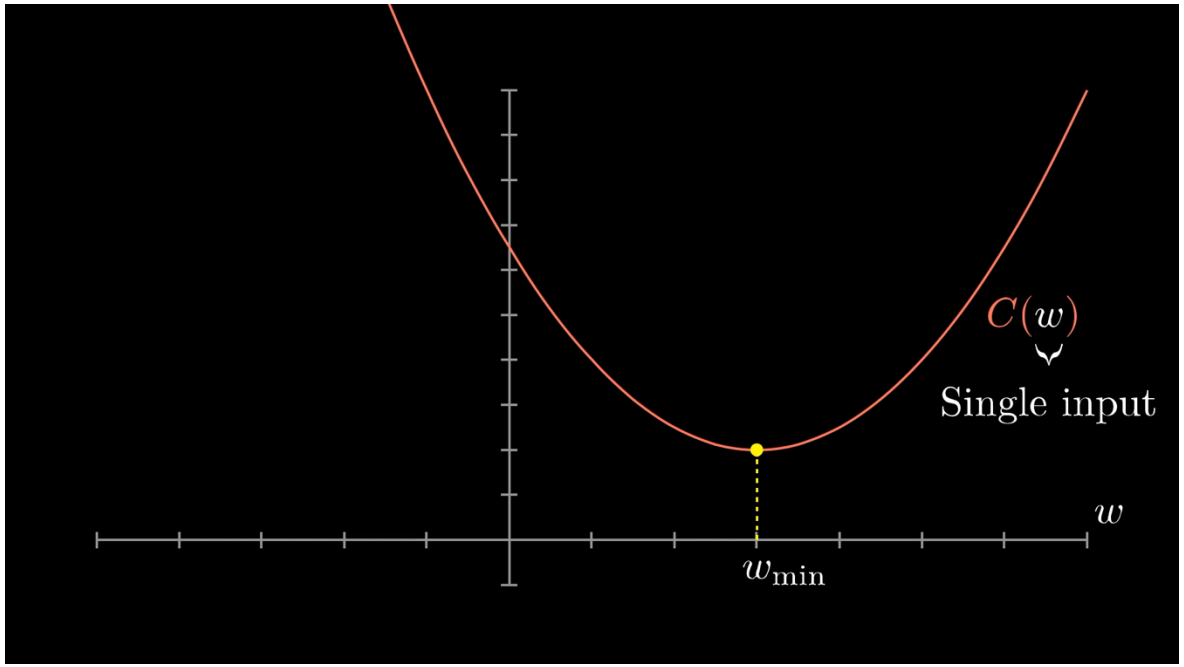
Cost Function

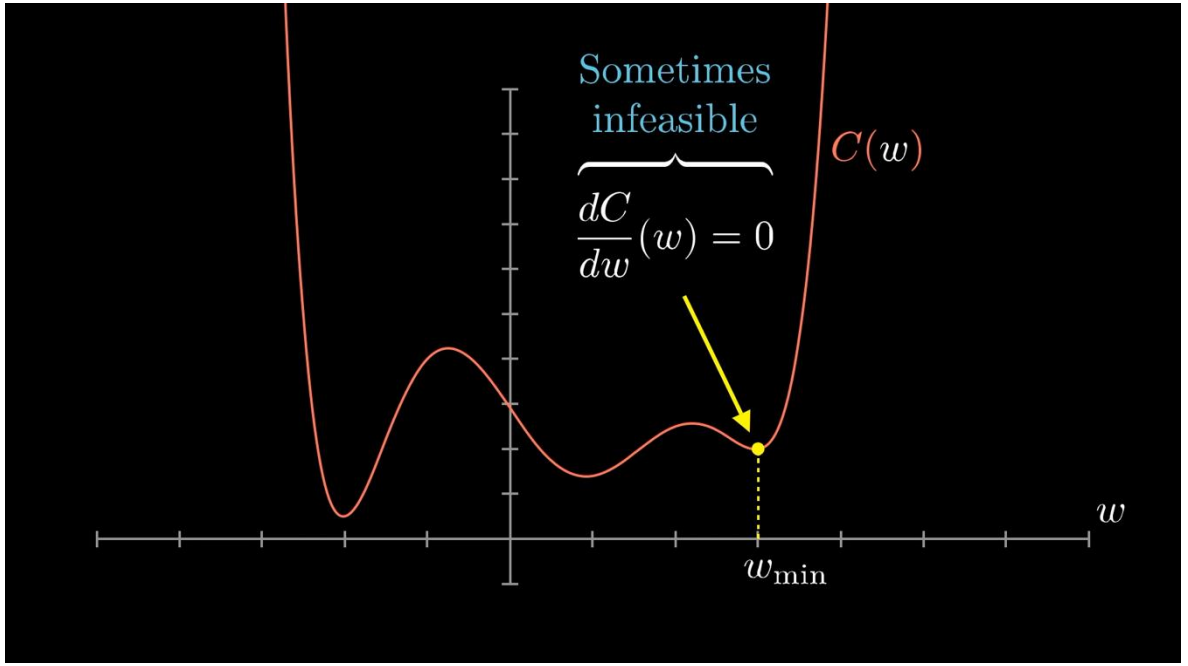




https://www.3blue1brown.com/lessons/gradient-descent

Cost Function

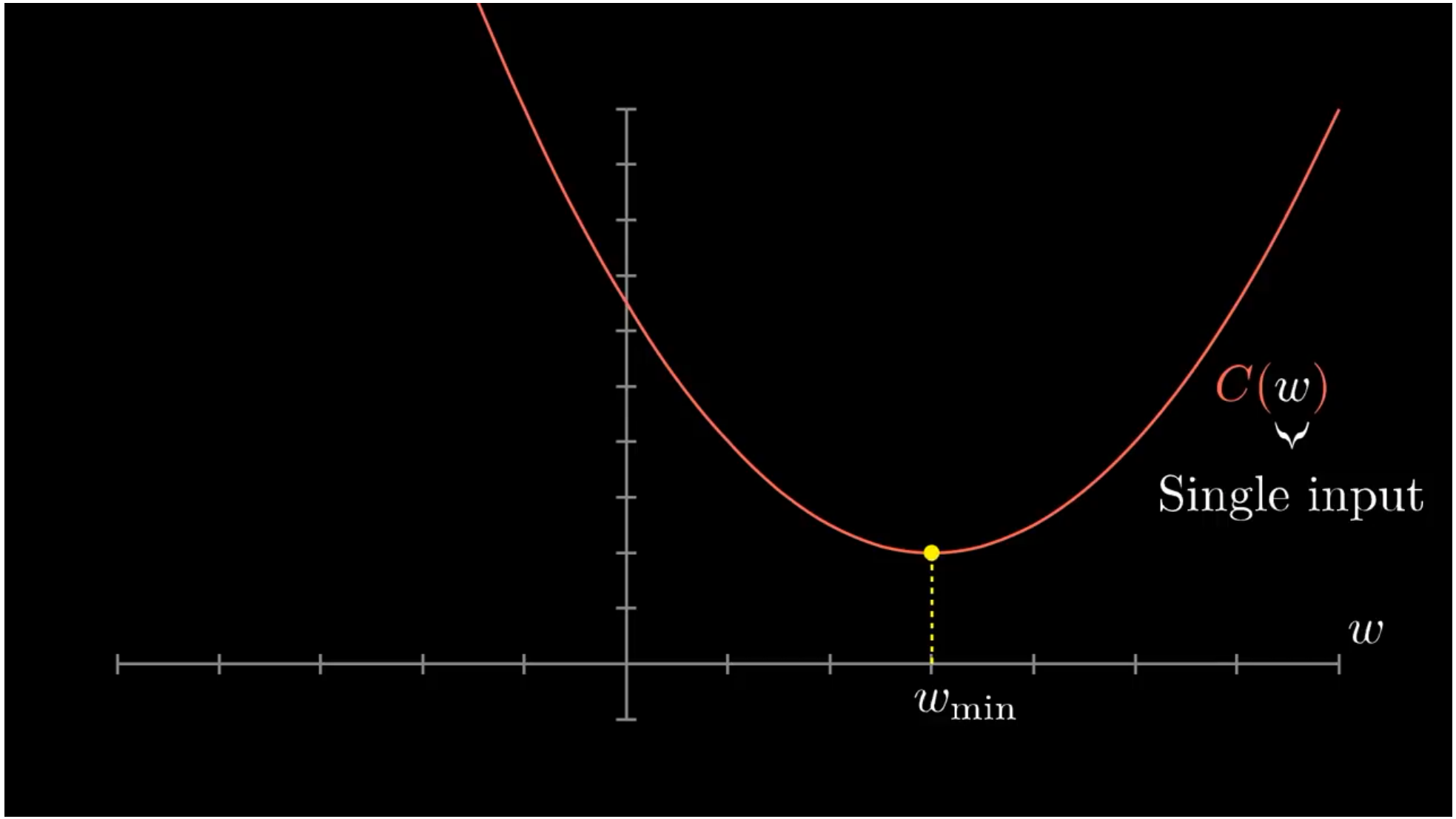




https://www.3blue1brown.com/lessons/gradient-descent

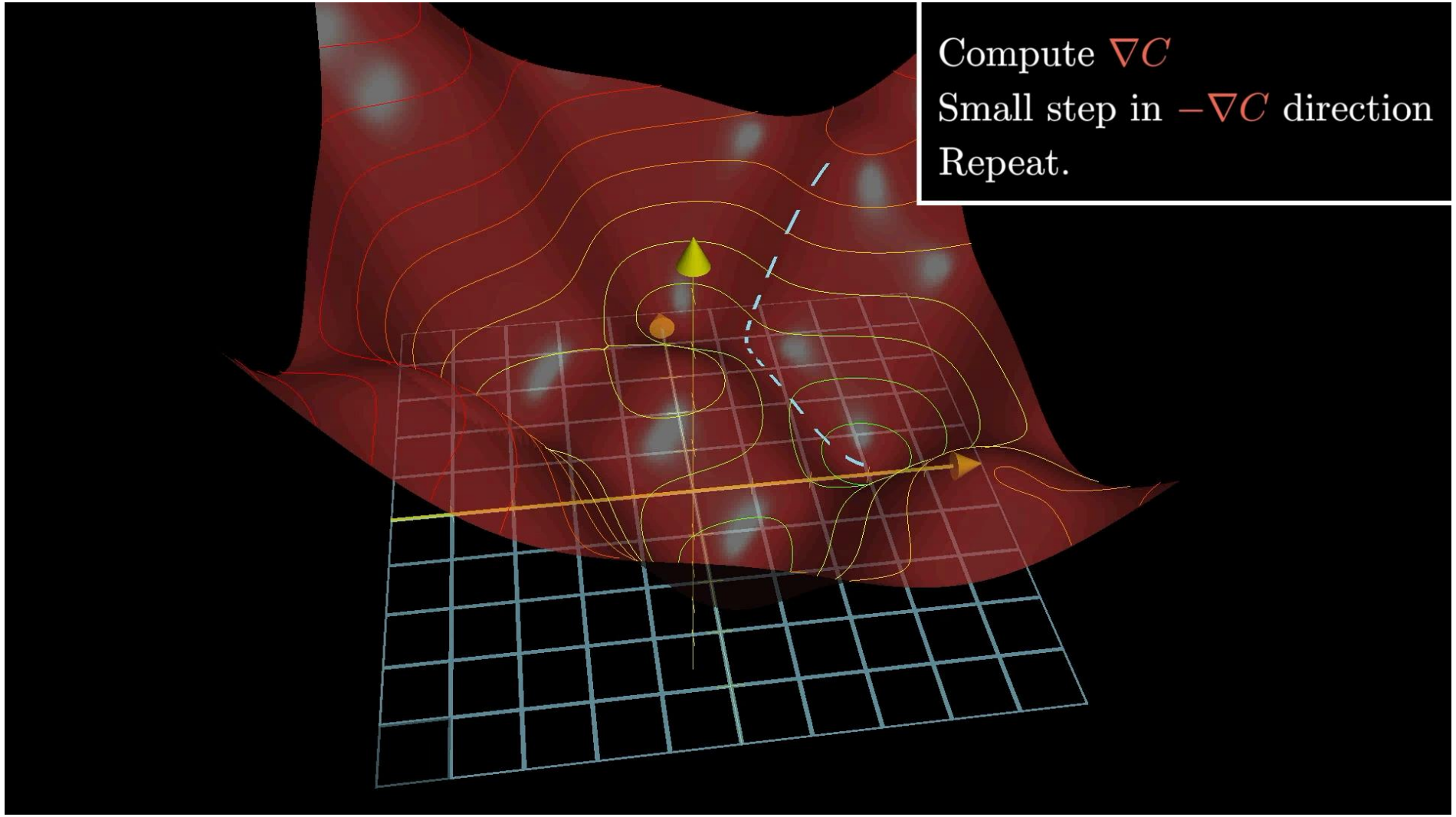
Optimization

Gradient Descent



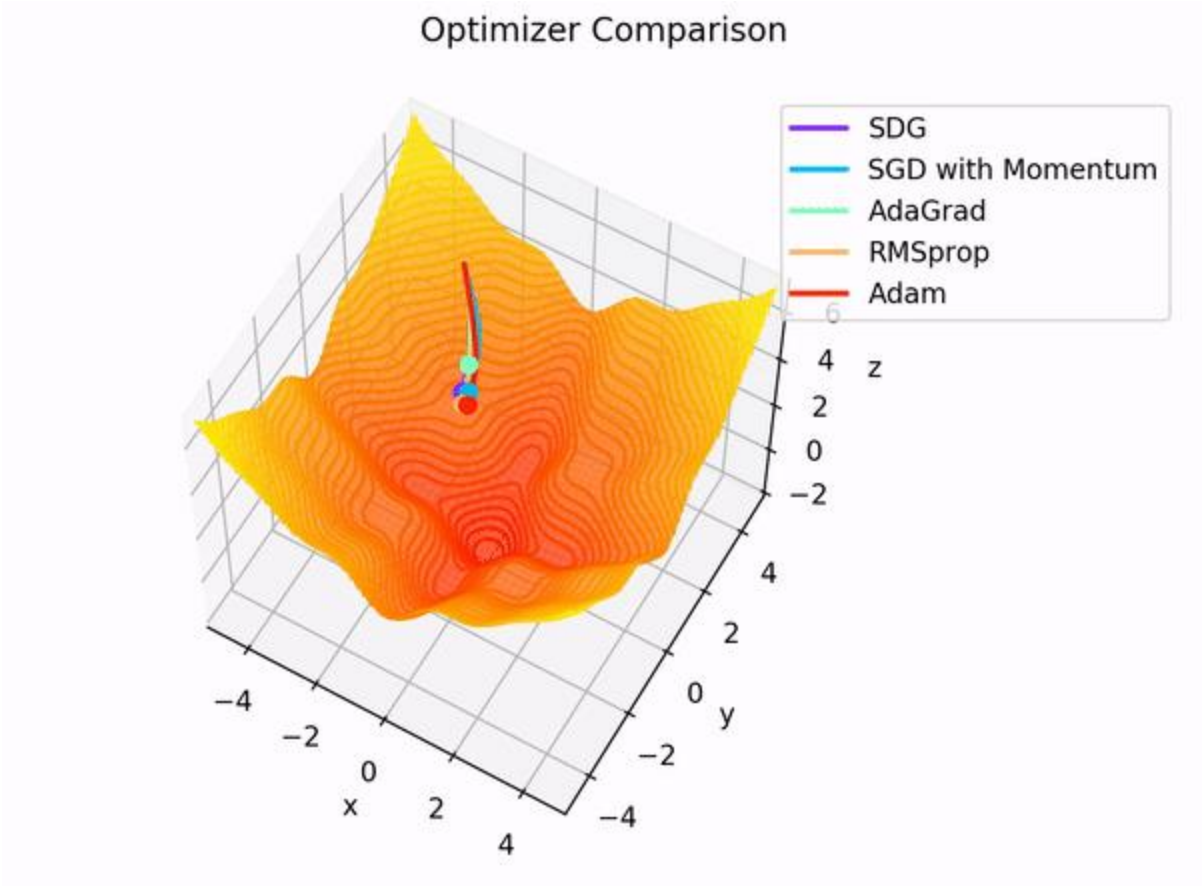
https://www.3blue1brown.com/lessons/gradient-descent

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

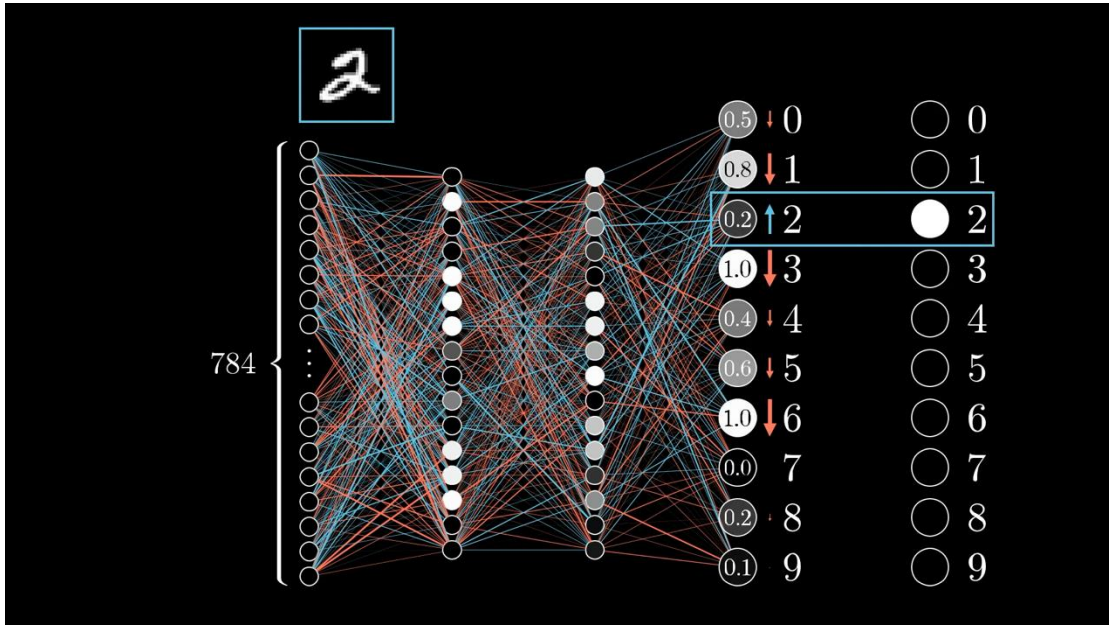
Gradient Descent

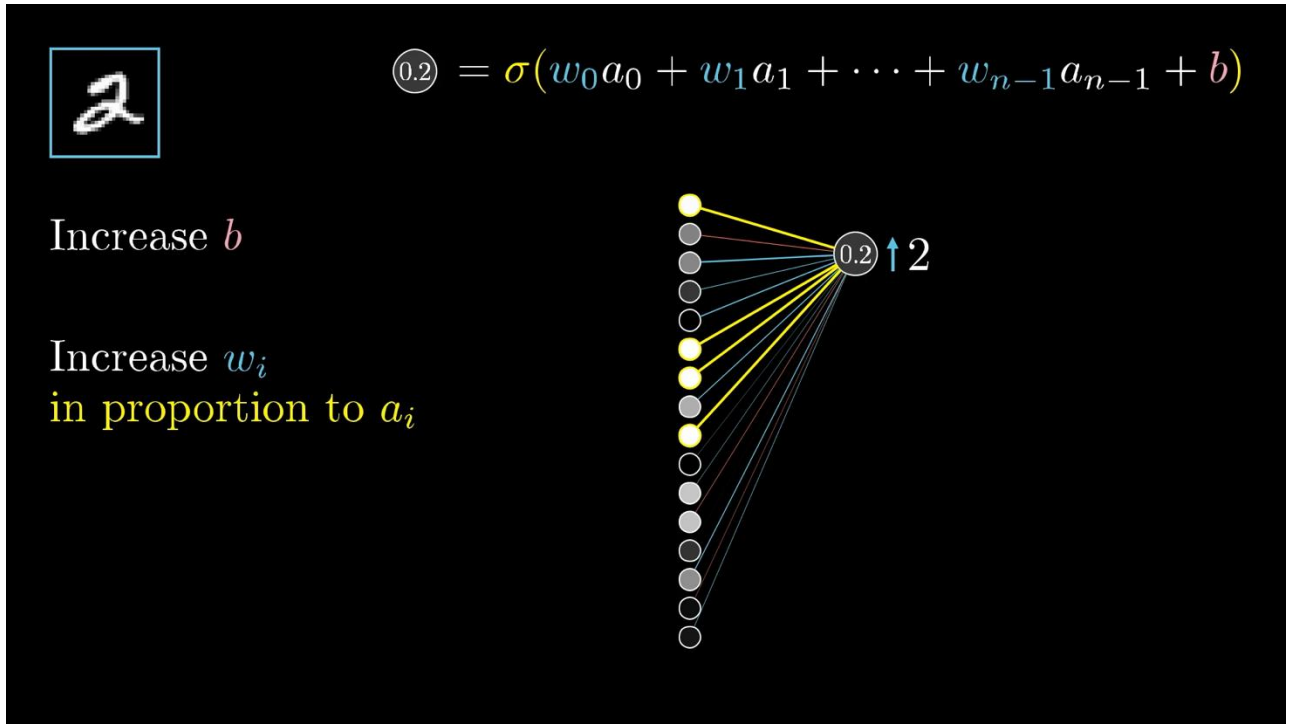
$$\vec{\mathbf{W}} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{13,000} \\ w_{13,001} \\ w_{13,002} \end{bmatrix}$$

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

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

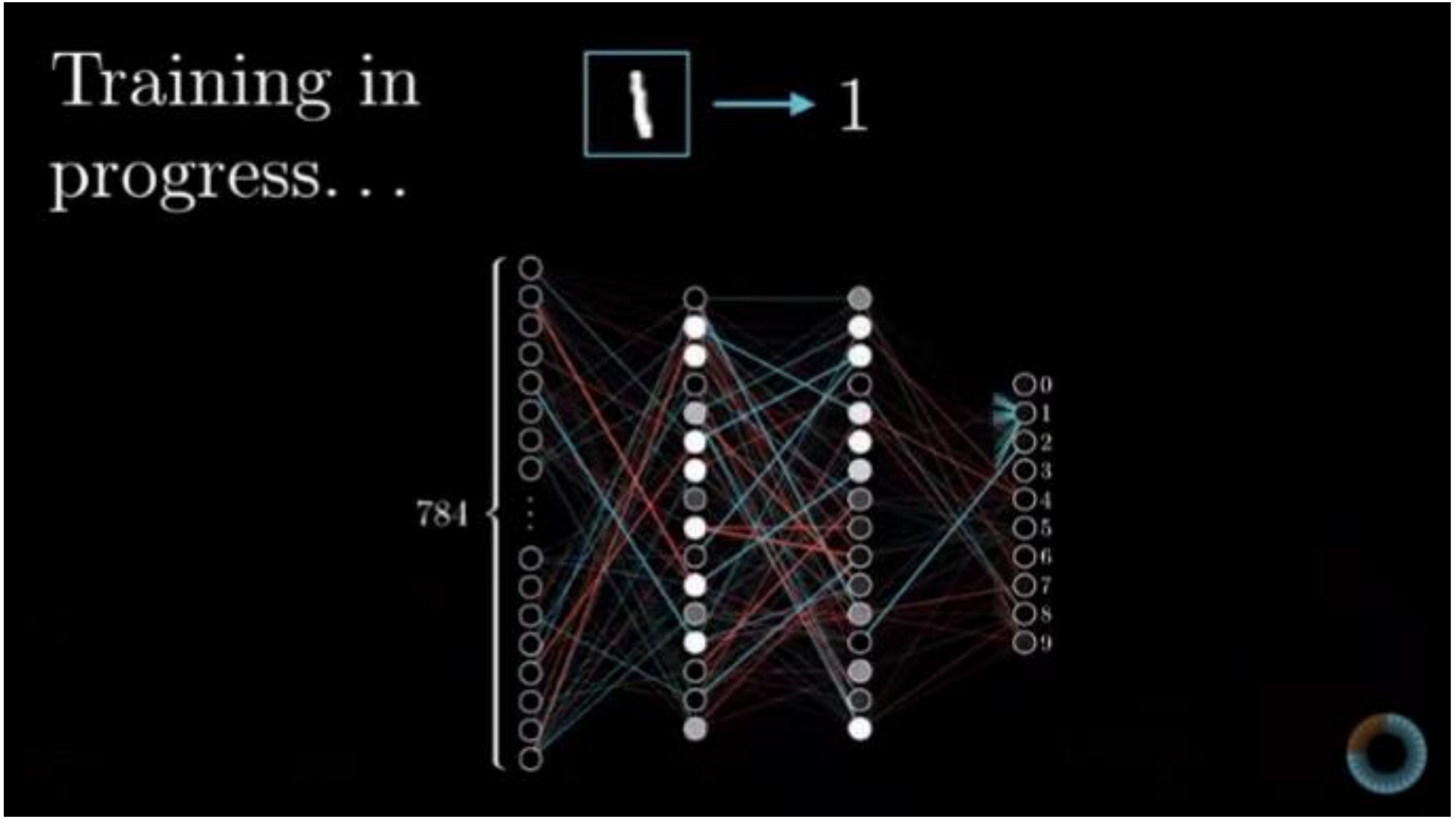
Backpropagation





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

Backpropagation



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

Training NNs

- Training = estimating all the weights $w_{ij}^{(l)}$ for all layers (l) with the aim of minimizing the loss or cost function

- Loss/cost function: how well your classifier/regressor do ?

→ define a function quantifying how far you are from the truth (e.g., Mean Squared Error)

$$loss = \frac{1}{n} \sum_{examplei=1}^{n} \left(y_i^{predicted} - y_i^{truth}\right)^2$$

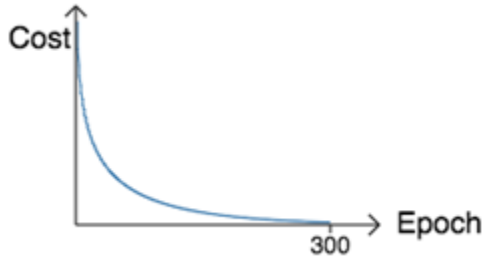
- Training algorithm :

- Step 1 Initialize randomly the weights

For each epoch :

Step 2 Forward pass your examples (eg labeled images of cats and birds) through nnet to compute predicted values (as previously explained)
Step 3 Compute loss
Step 4 Backward propagation
Step 5 Update weights (gradient descent algorithm)

Repeat from step 2 until a plateau is reached



- **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.

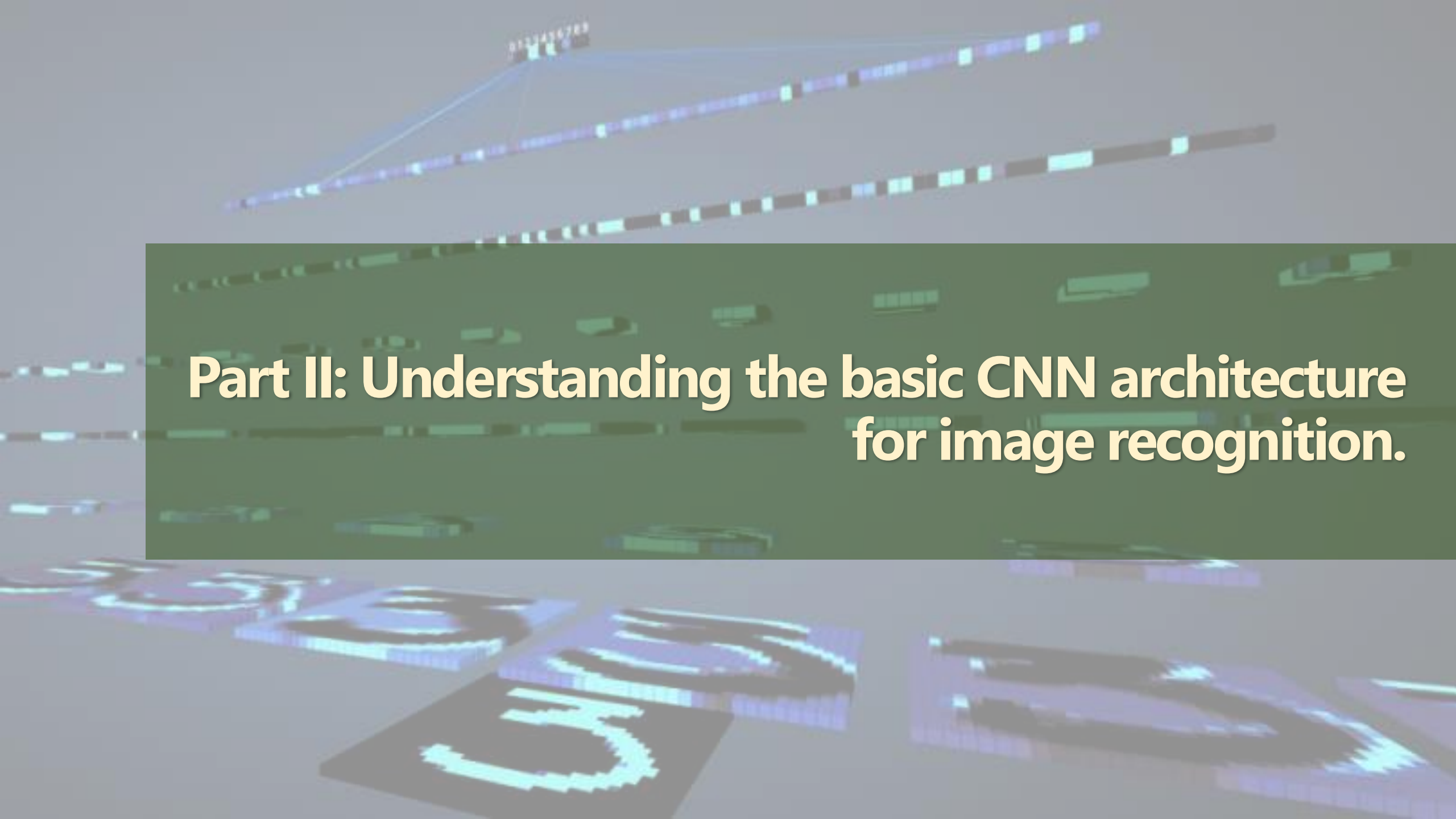
Goals

- Conceive and simulate genetic data under competing demographic scenarios

- Understand deep learning background and how a CNN works

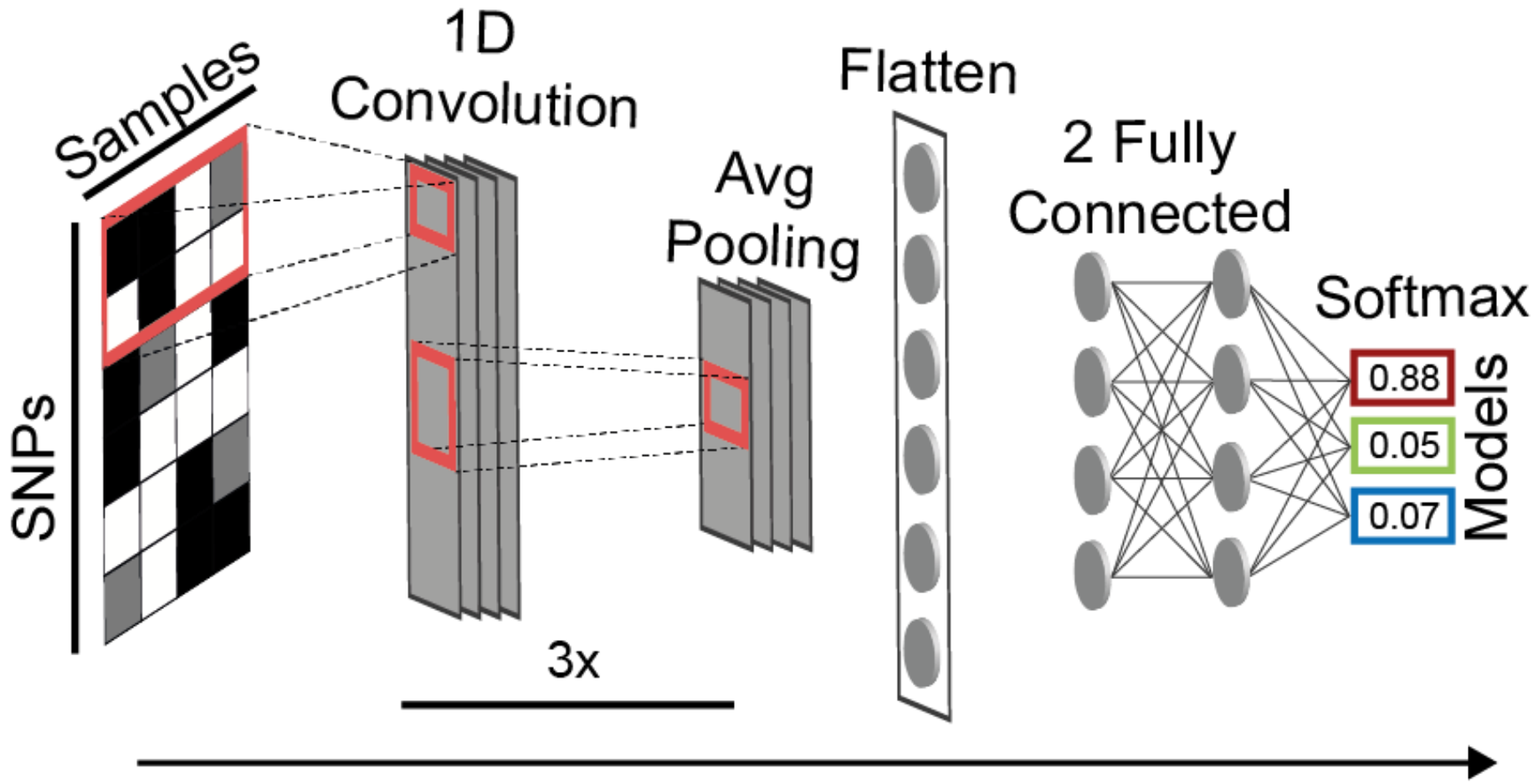
- How to use deep learning to compare demographic scenarios

- Use CNN to detect regions with selective sweeps on real genomes



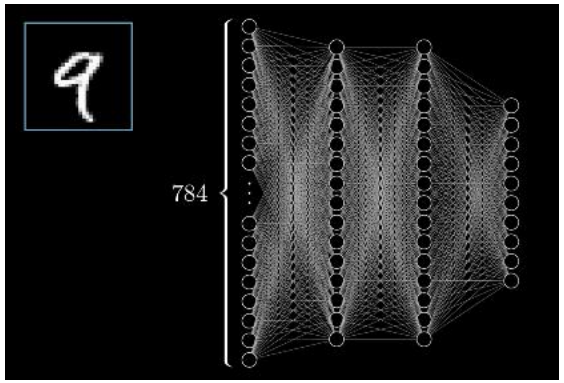
Part II: Understanding the basic CNN architecture for image recognition.

CNN architecture



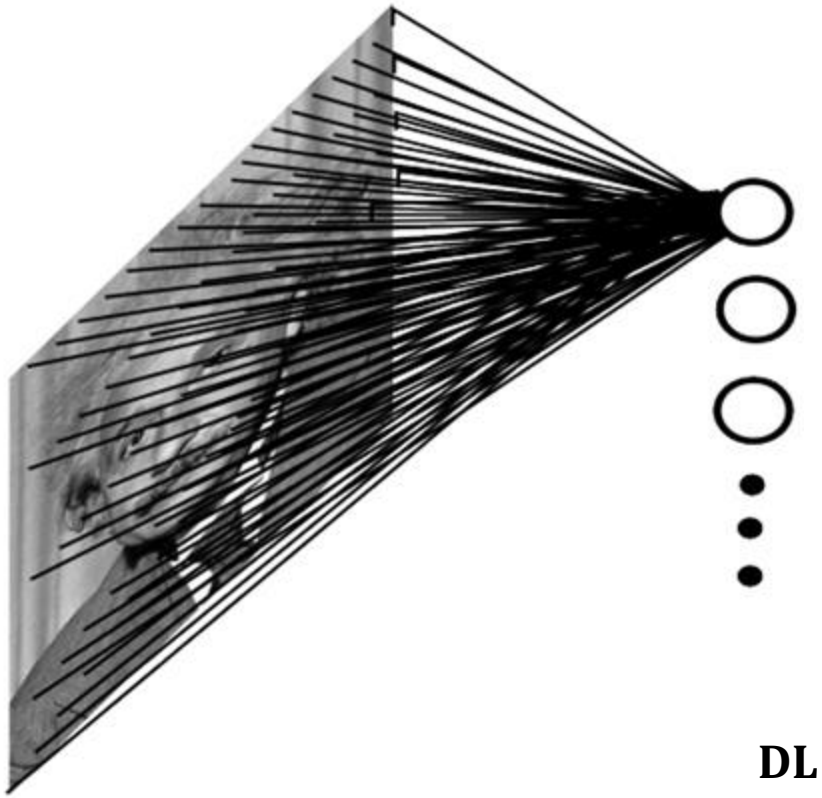
Kirschner et al. (2022) *Nat Comm*

Convolutional Neural Networks (CNNs)

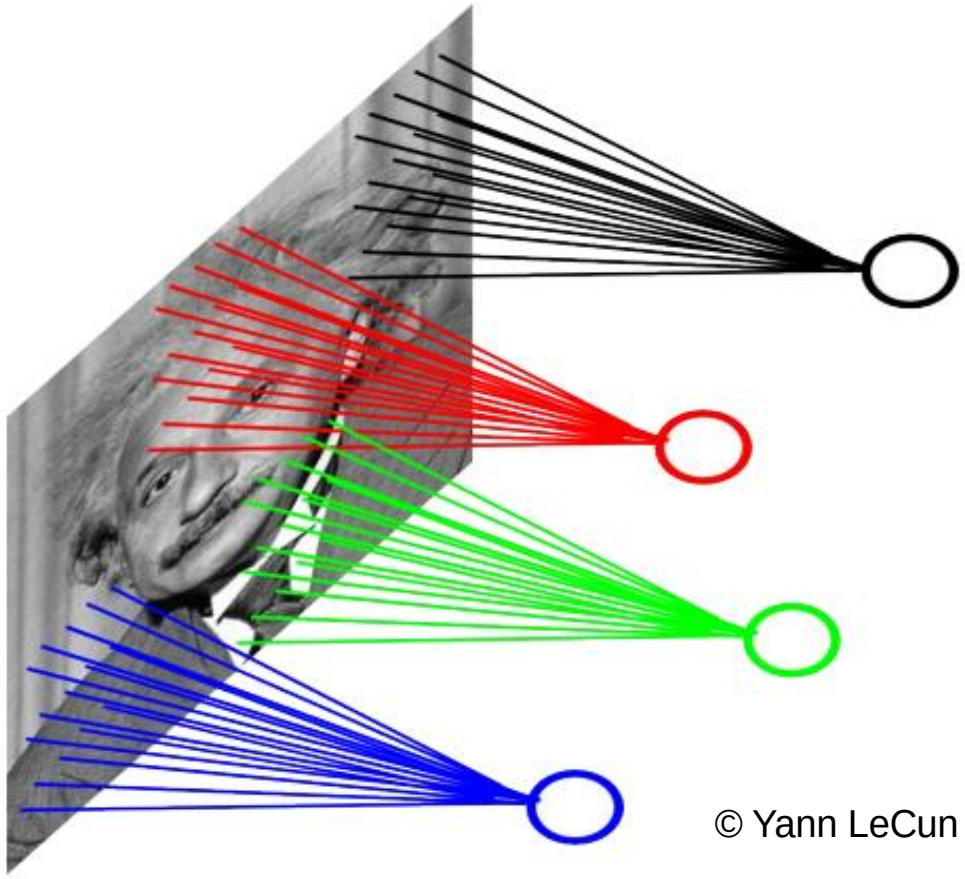


MLP
Fully connected

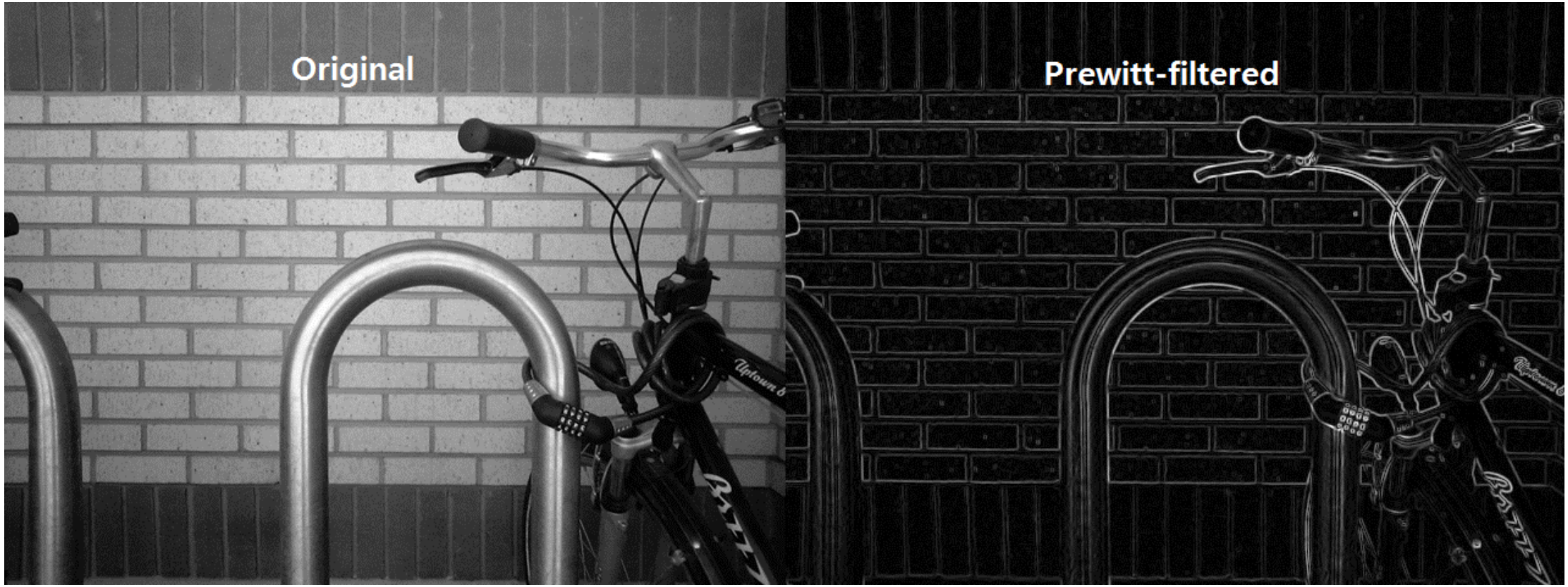
Convnet
locally connected
→ smaller number of parameters to learn



DL witchcraft

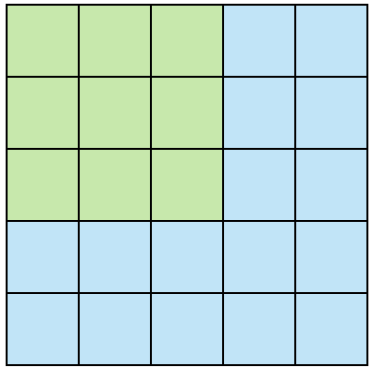


Convolutional Layers

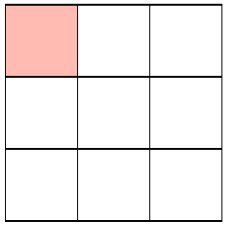


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

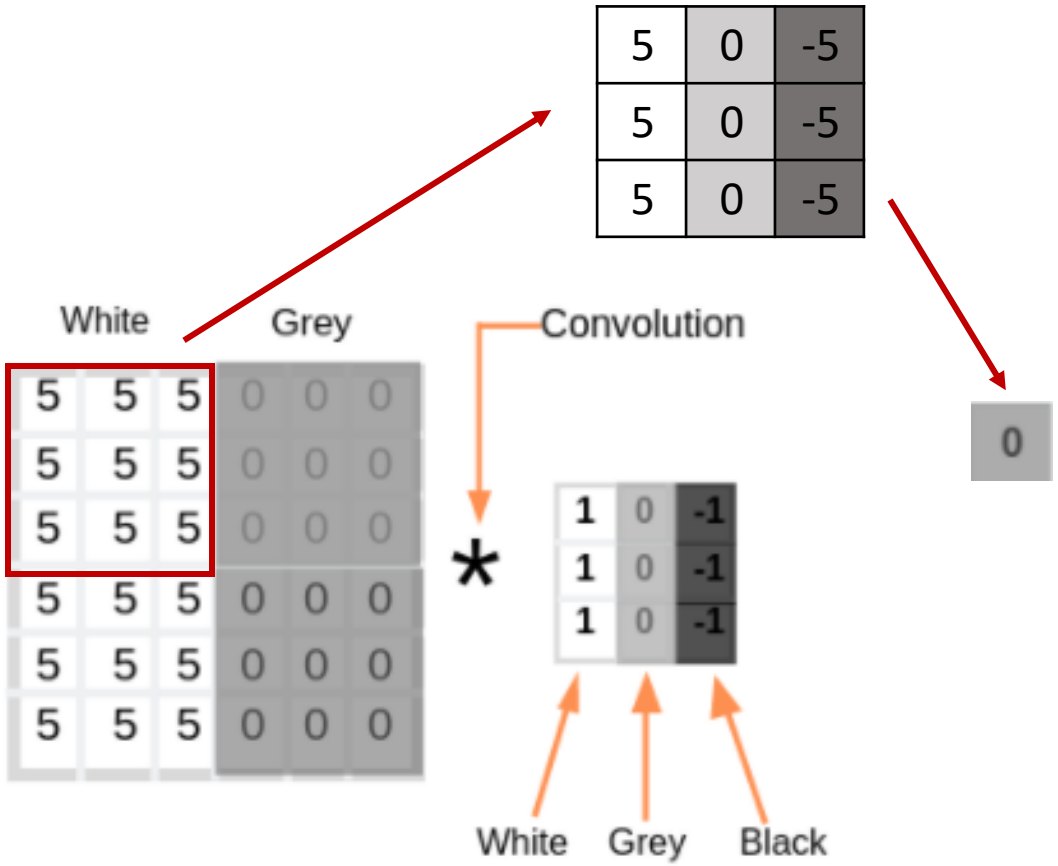
Convolutional Layers



Stride 1

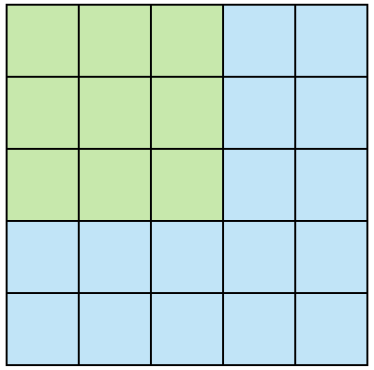


Feature Map

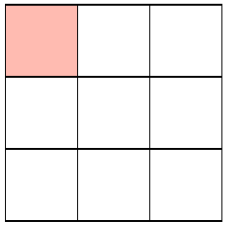


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

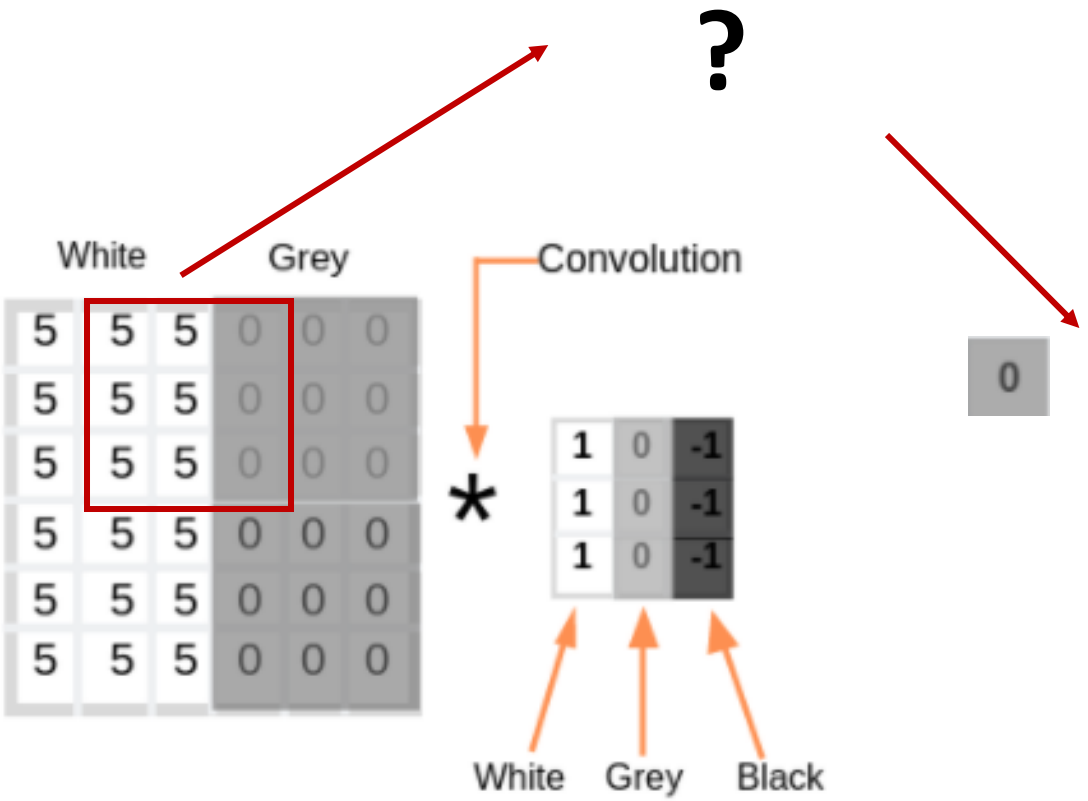
Convolutional Layers



Stride 1

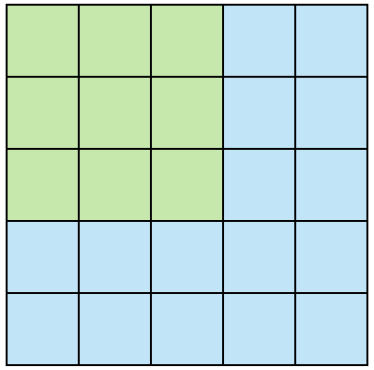


Feature Map

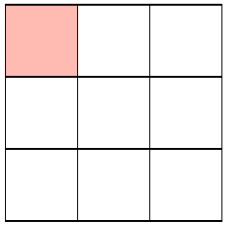


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

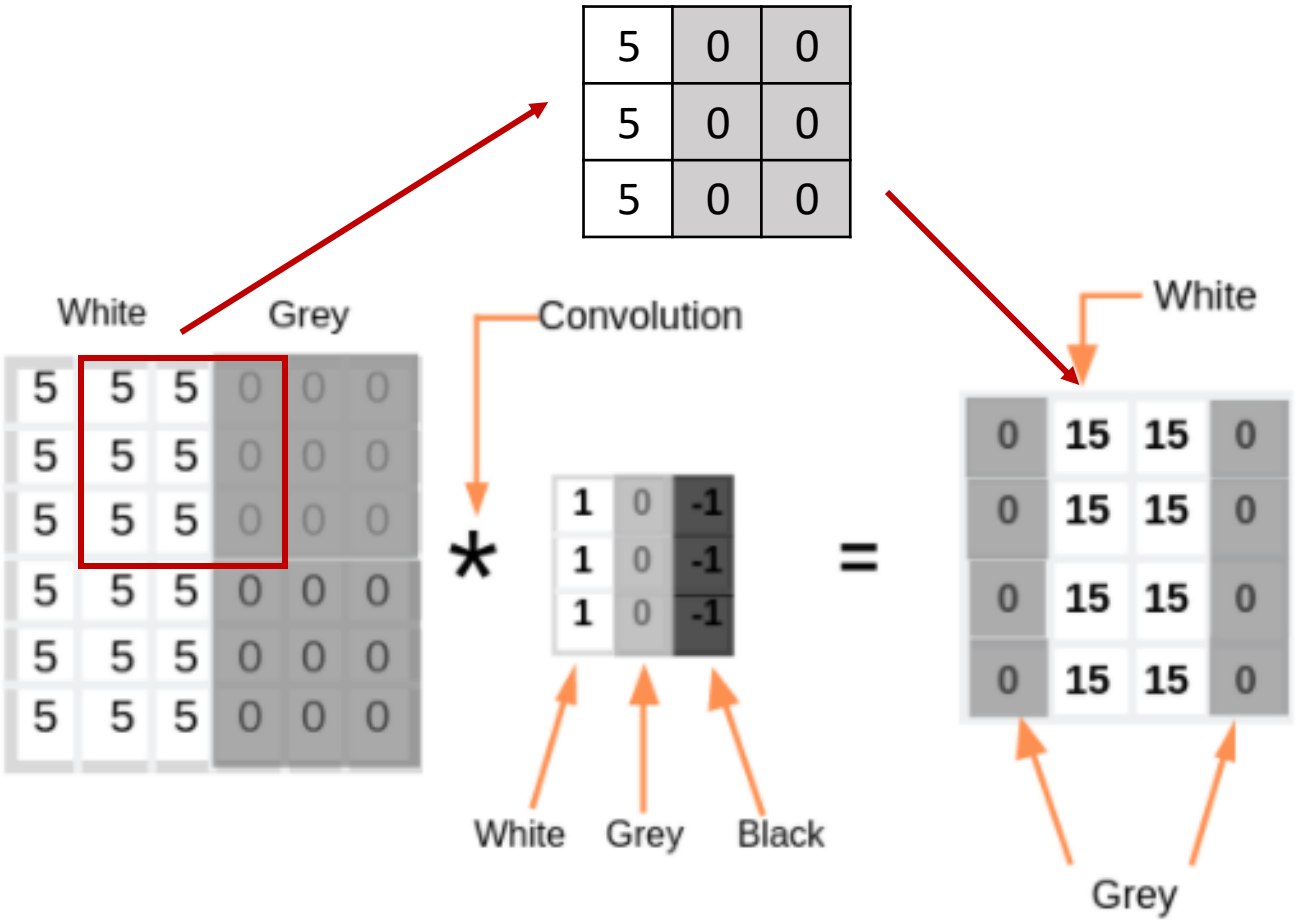
Convolutional Layers



Stride 1

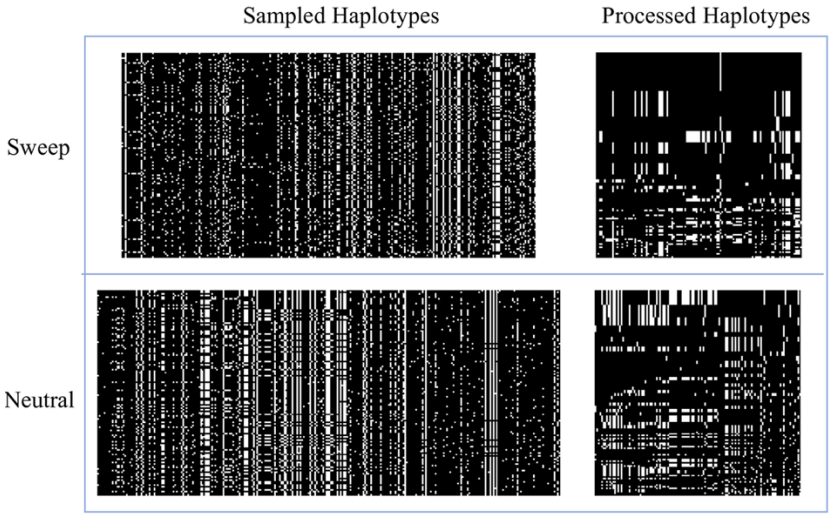


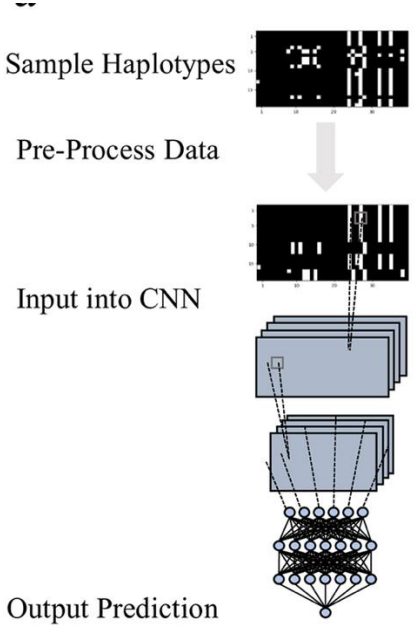
Feature Map

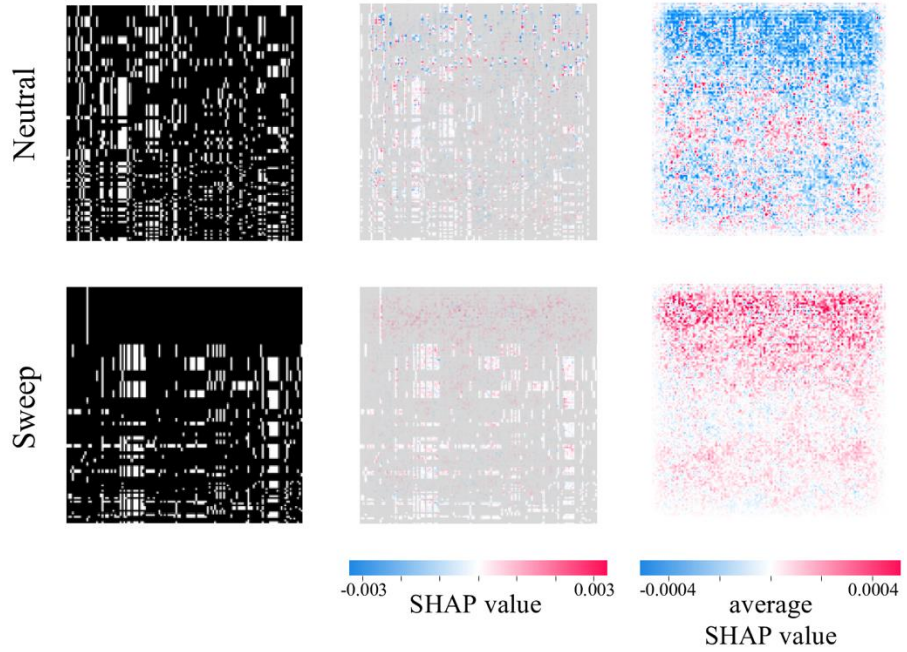


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

Neural Network for Genomic data

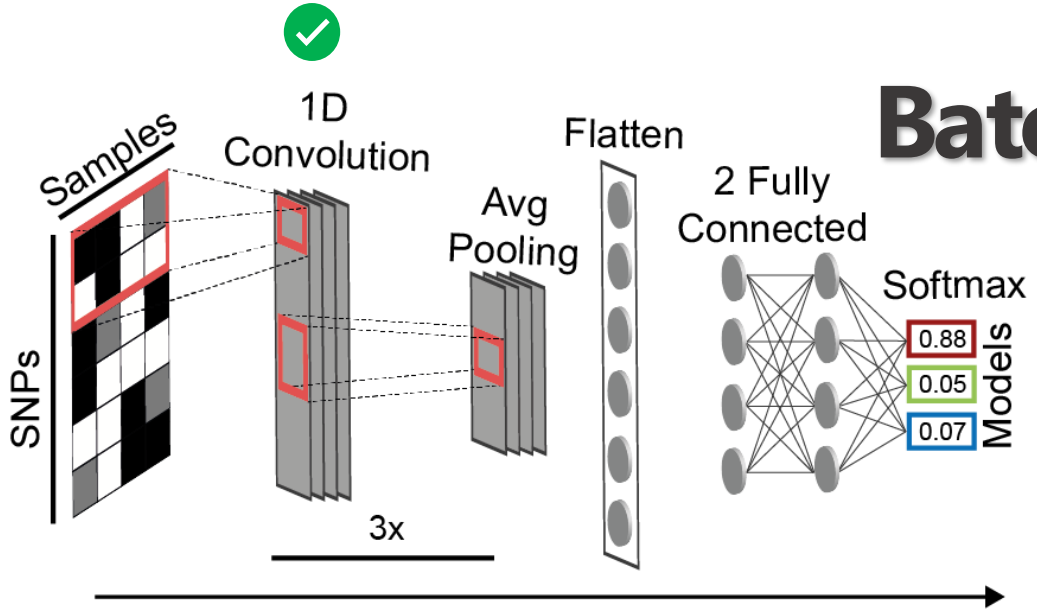




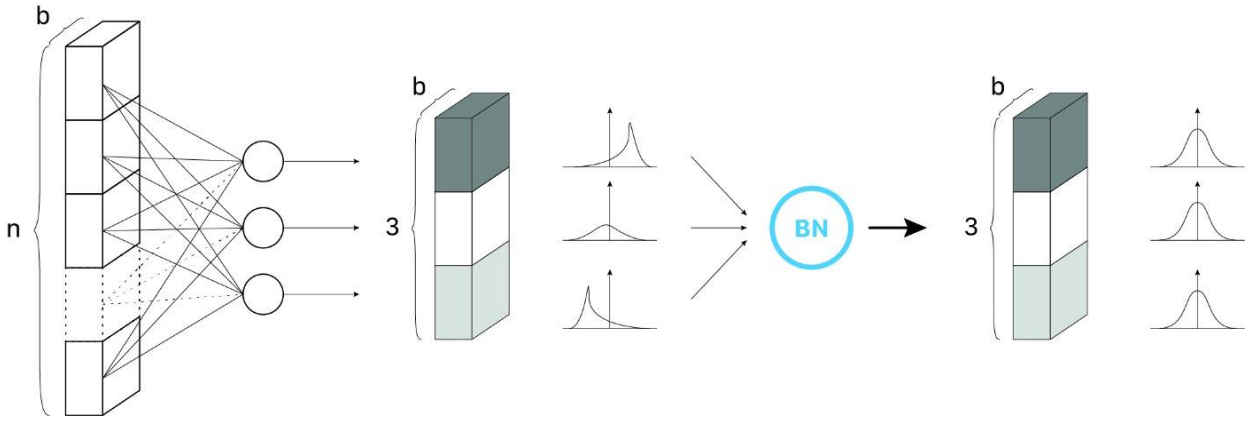


Cecil & Sudgen (2023) *PLoS Comp Biol*

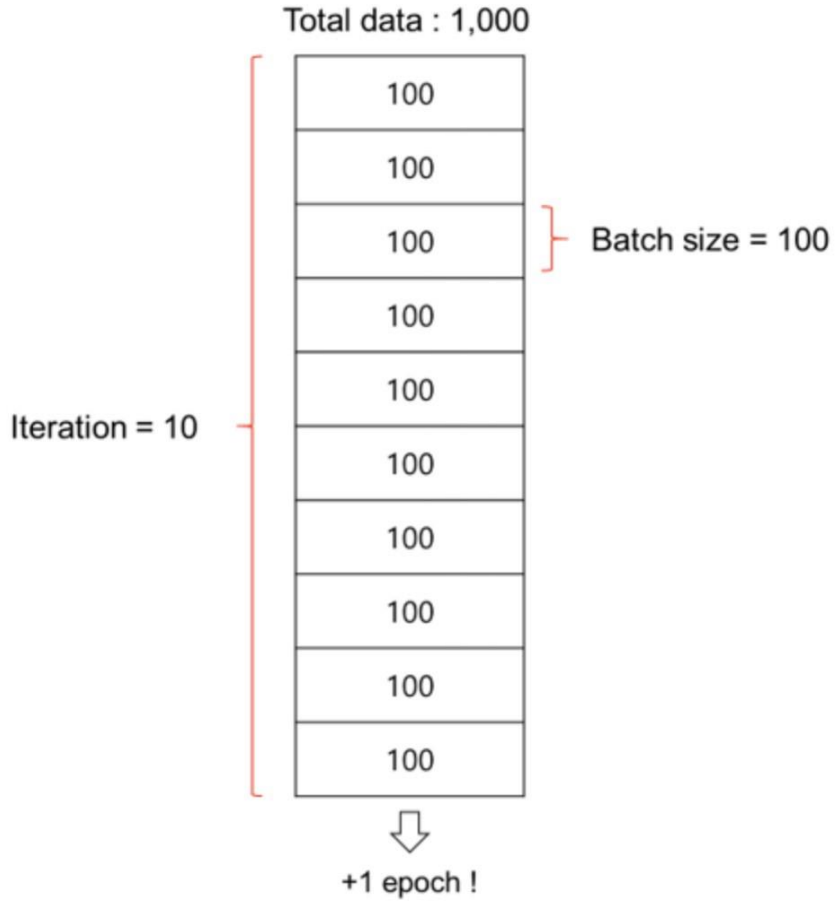
Batches and Batch Normalization



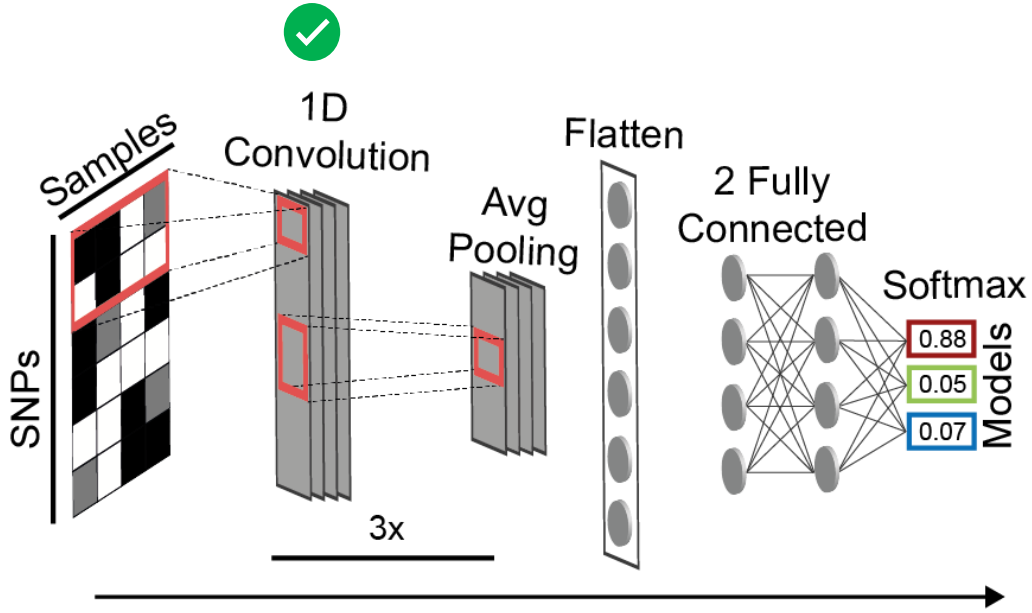
Kirschner et al. (2022) *Nat Comm*



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



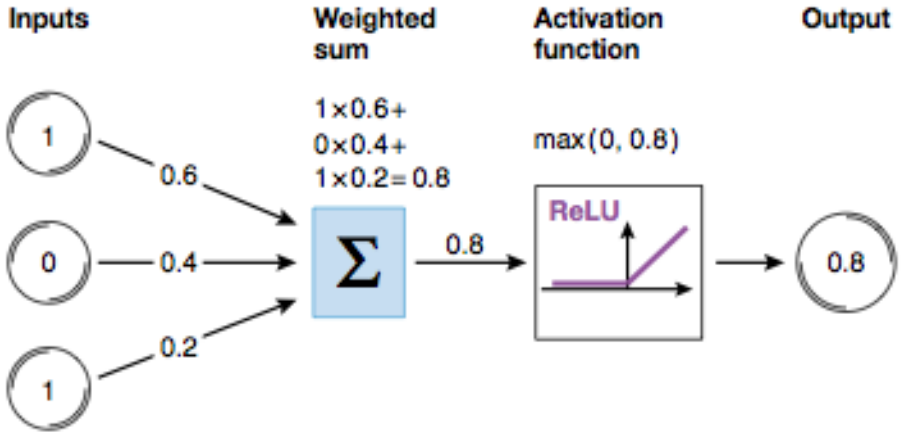
https://jerryan.medium.com/batch-size-a15958708a6



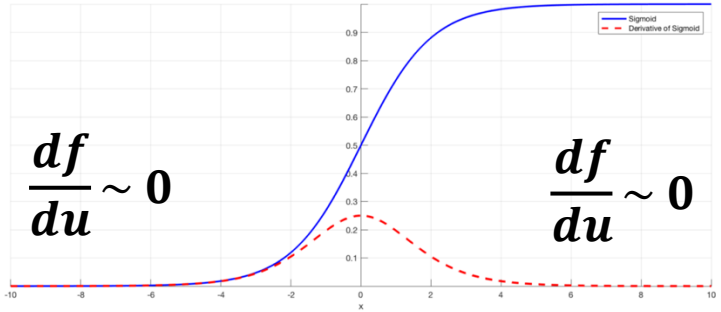
Kirschner et al. (2022) *Nat Comm*

Activation Function

DL witchcraft

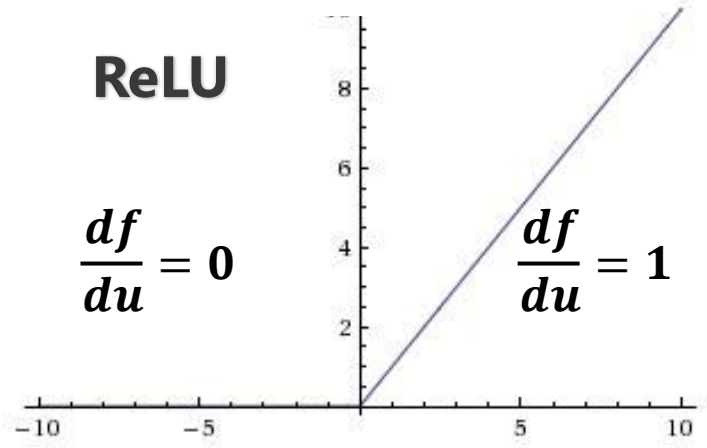


Sigmoid – Vanishing gradients



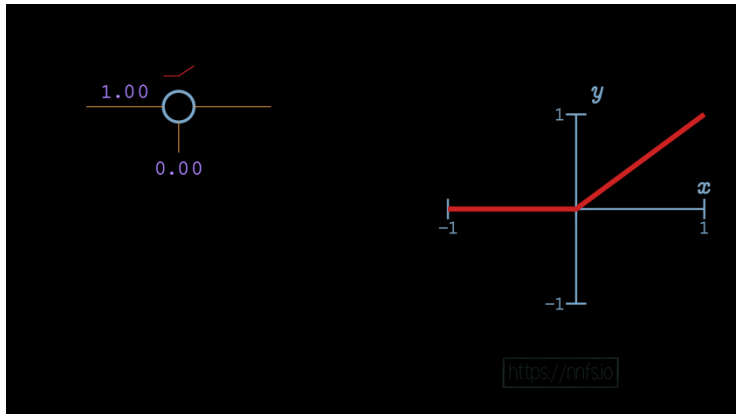
https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

ReLU

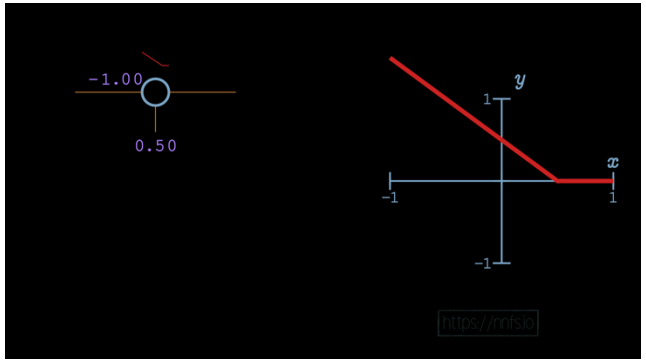


https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook

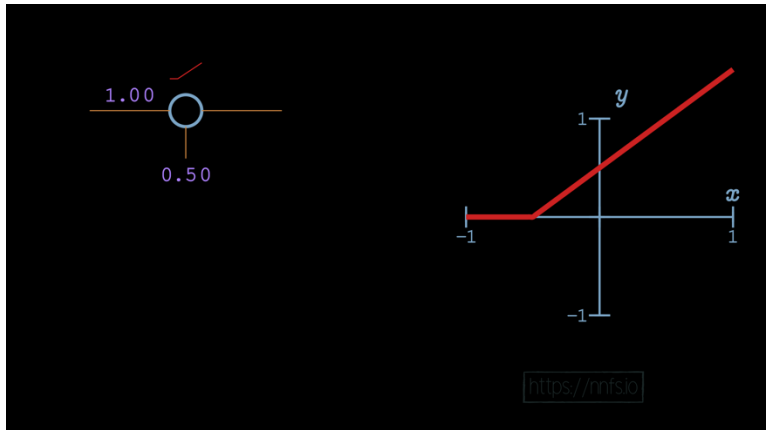
CNN Script – ReLU



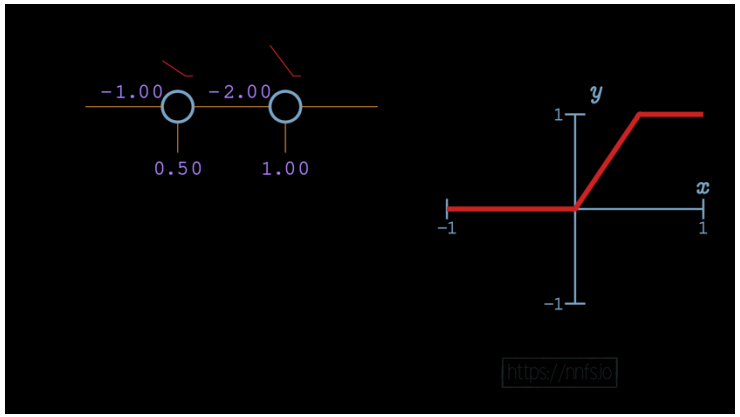
Weight = 1
Bias = 0



Weight = -1
Bias = 0.5



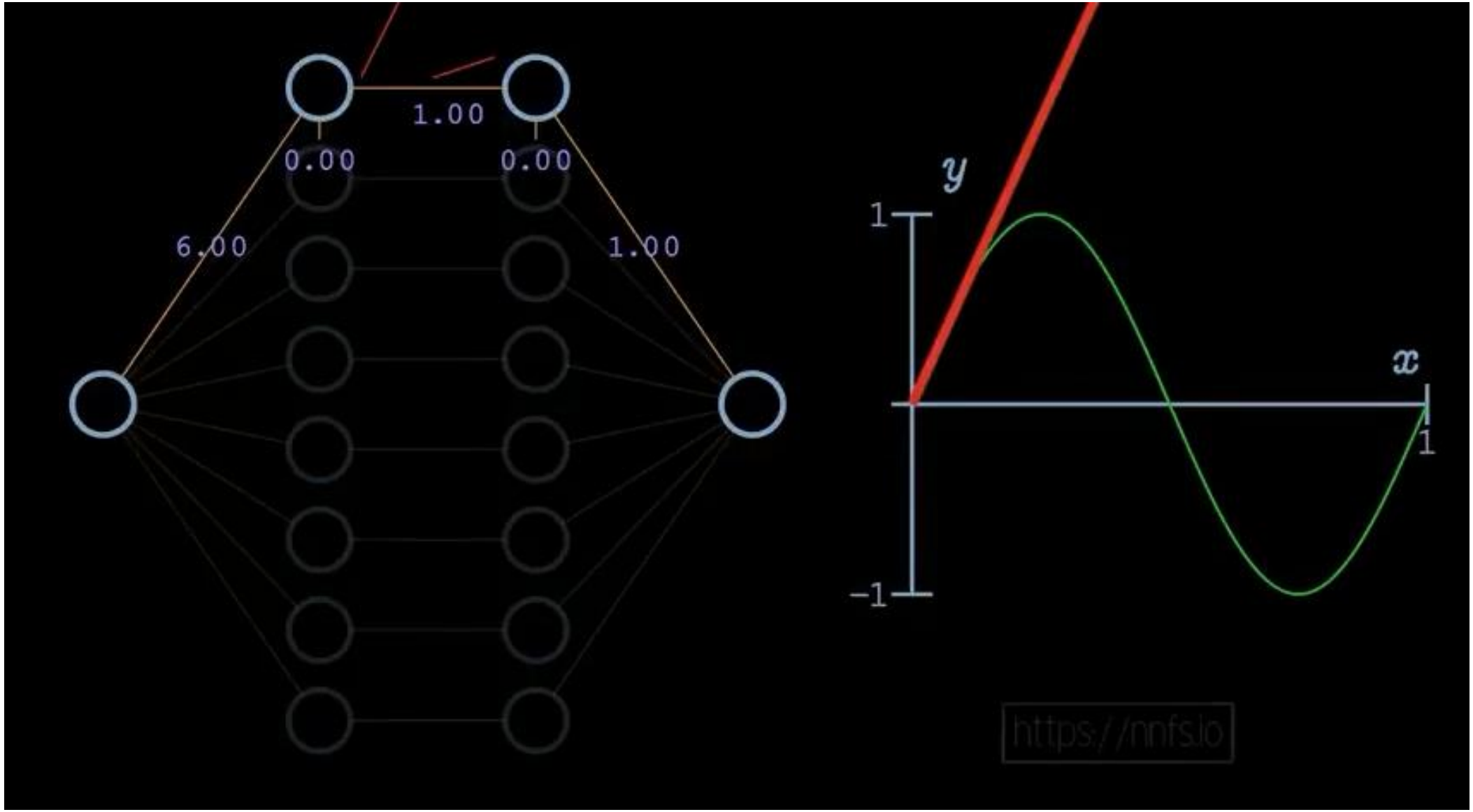
Weight = 1
Bias = 0.5



2 neurons

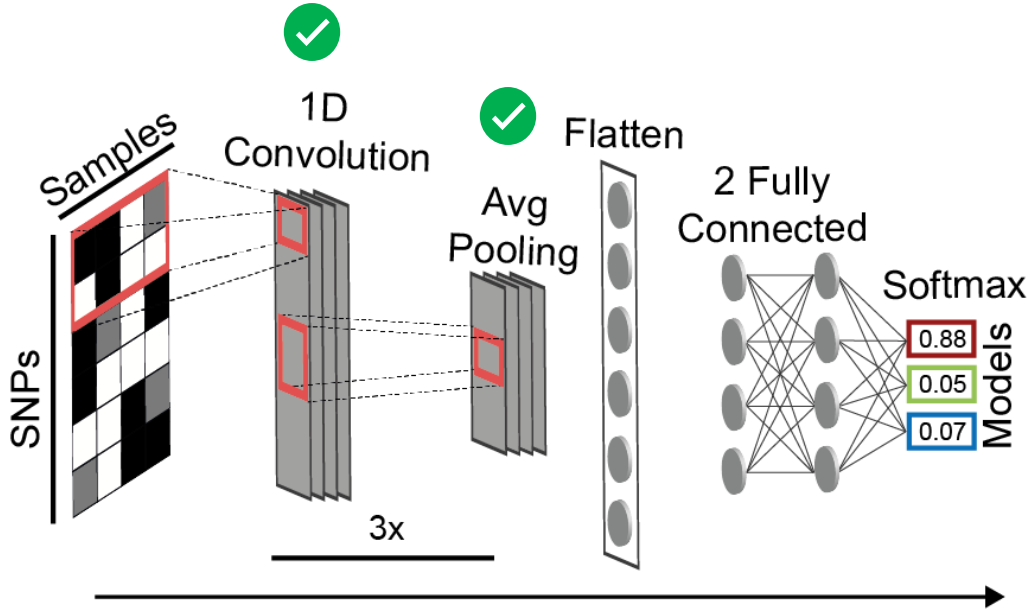
https://nnfs.io/mvp/

CNN Script – ReLU

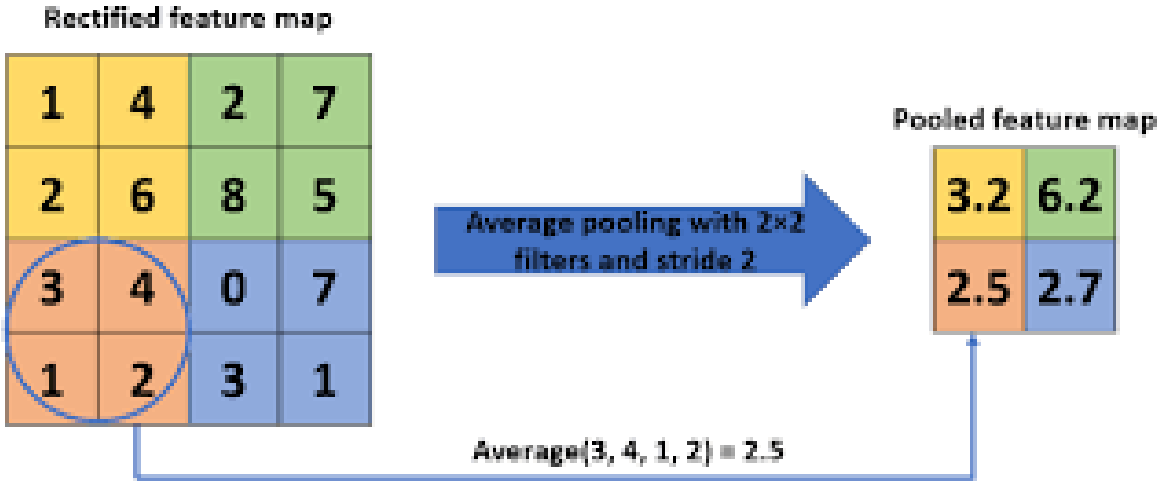


https://nnfs.io/mvp/

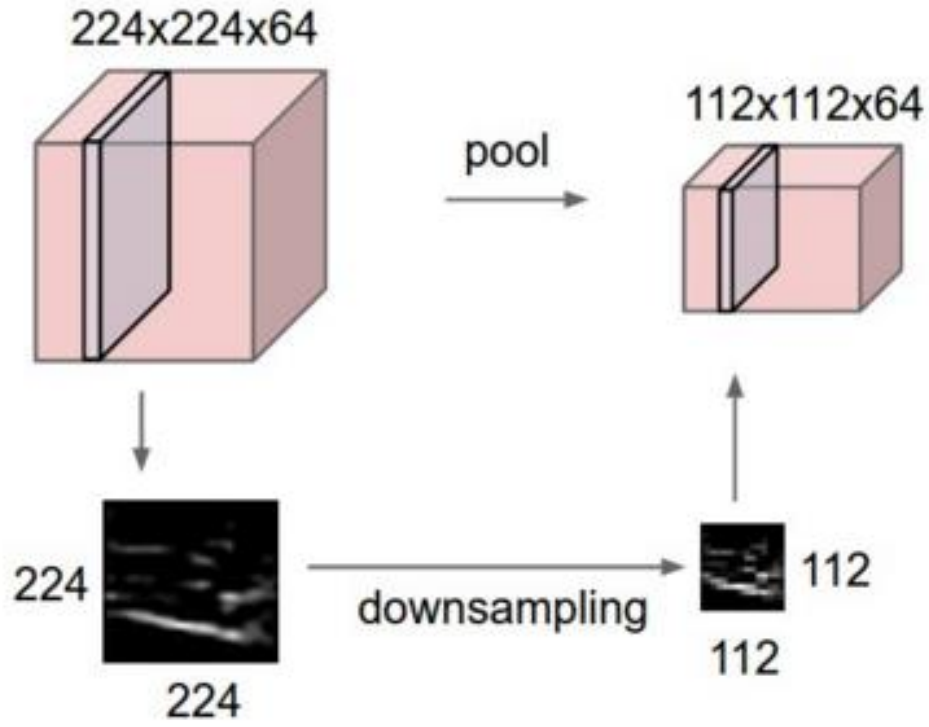
Pooling



Kirschner et al. (2022) *Nat Comm*

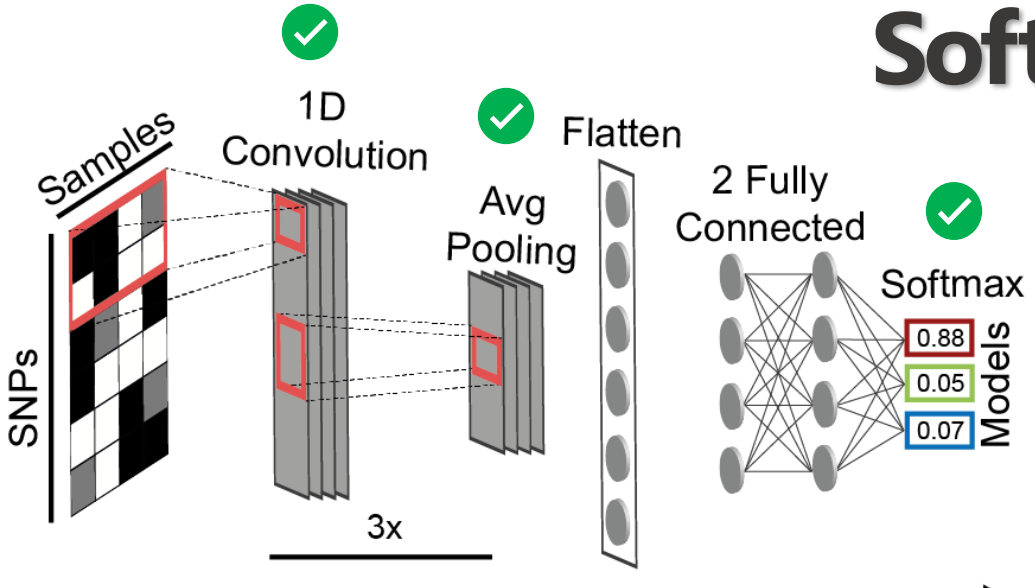


Gholamalinezhad & Khosravi (2020) *arXiv*

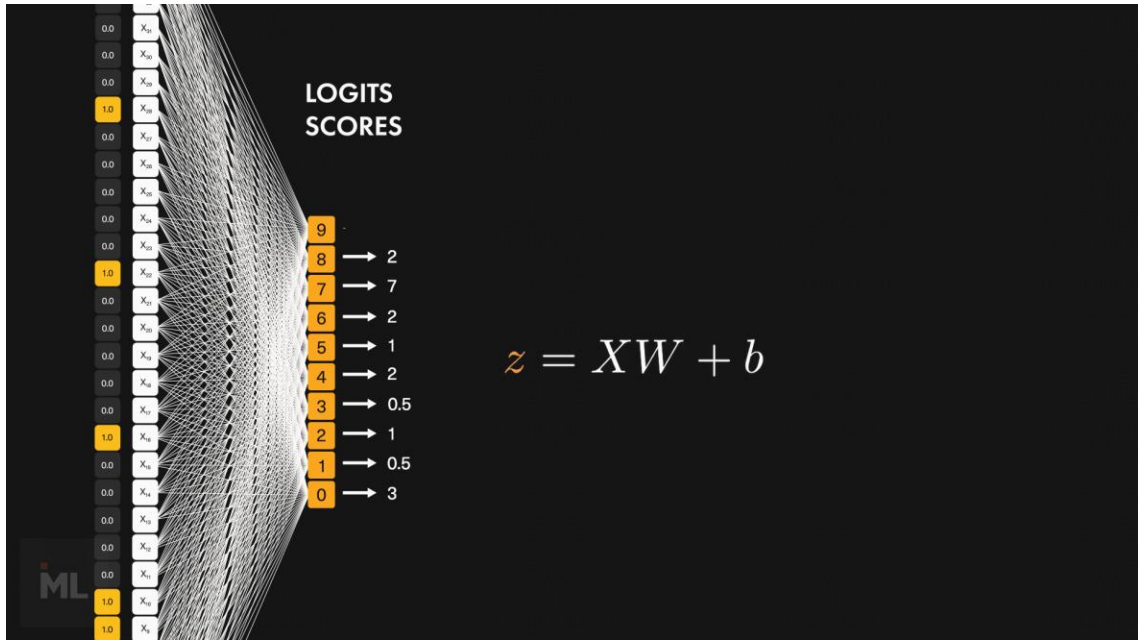


https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/pooling_layer

Softmax layer for Classification



Kirschner et al. (2022) *Nat Comm*



Hyperparameters

Libraries for deep Learning:
Keras, Tensorflow, pyTorch

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

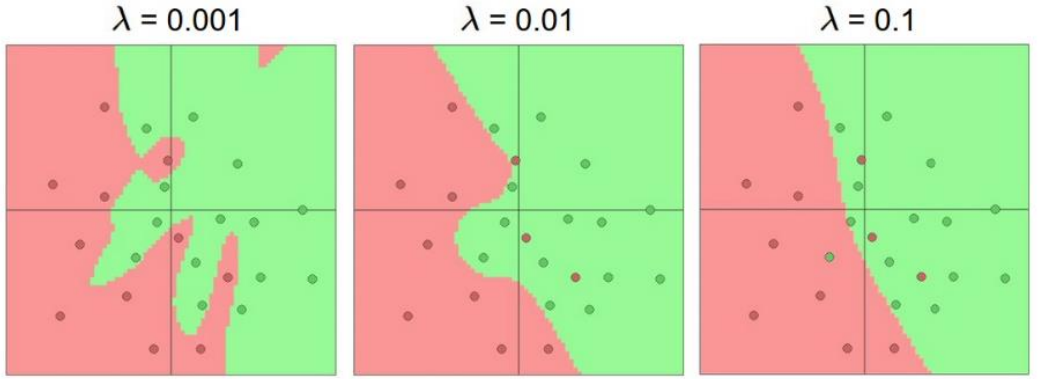
https://neptune.ai/blog/hyperparameter-tuning-in-python-complete-guide#hyperopt

Hyperparameters

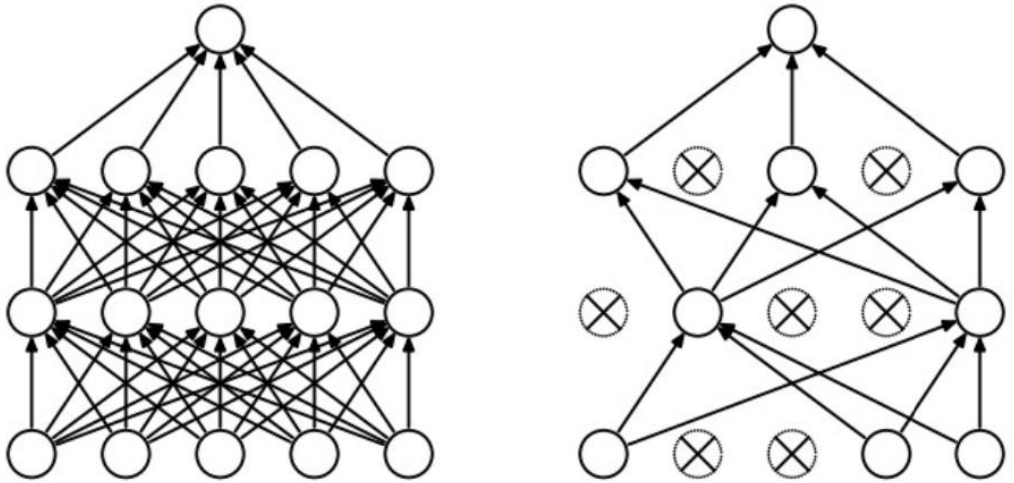
Table 2. Central parameters of a neural network and recommended settings.

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.

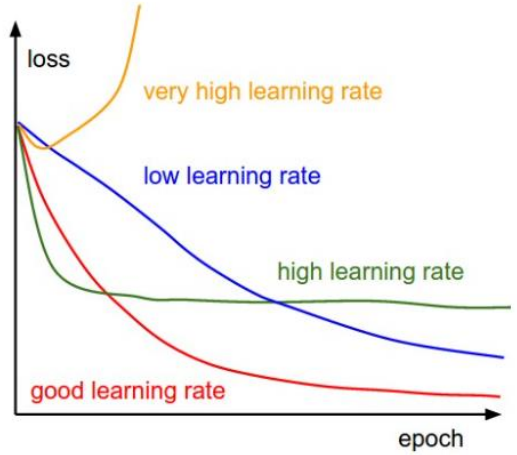


(a) Standard Neural Net (b) After applying dropout.

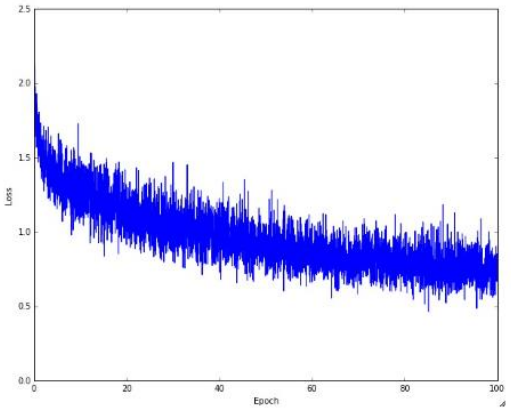
Hyperparameters

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	



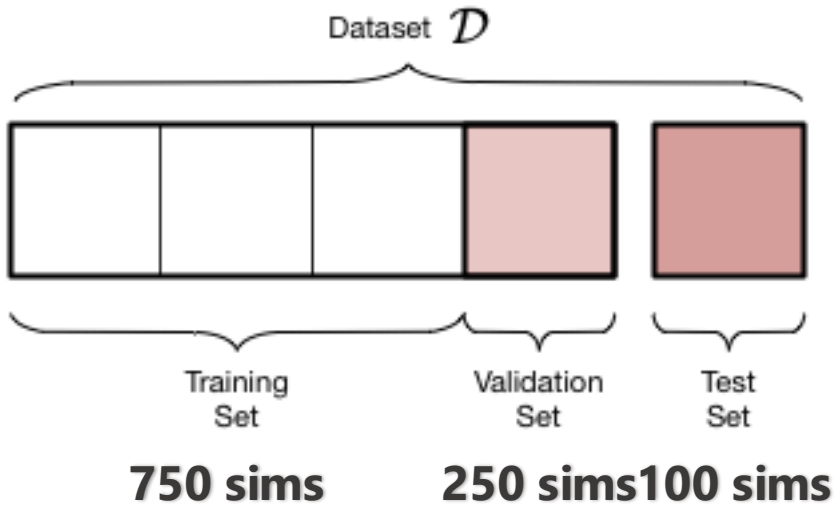
effects of different learning rates

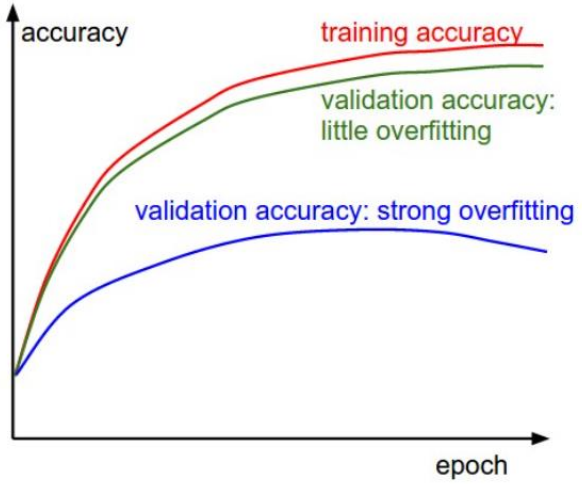


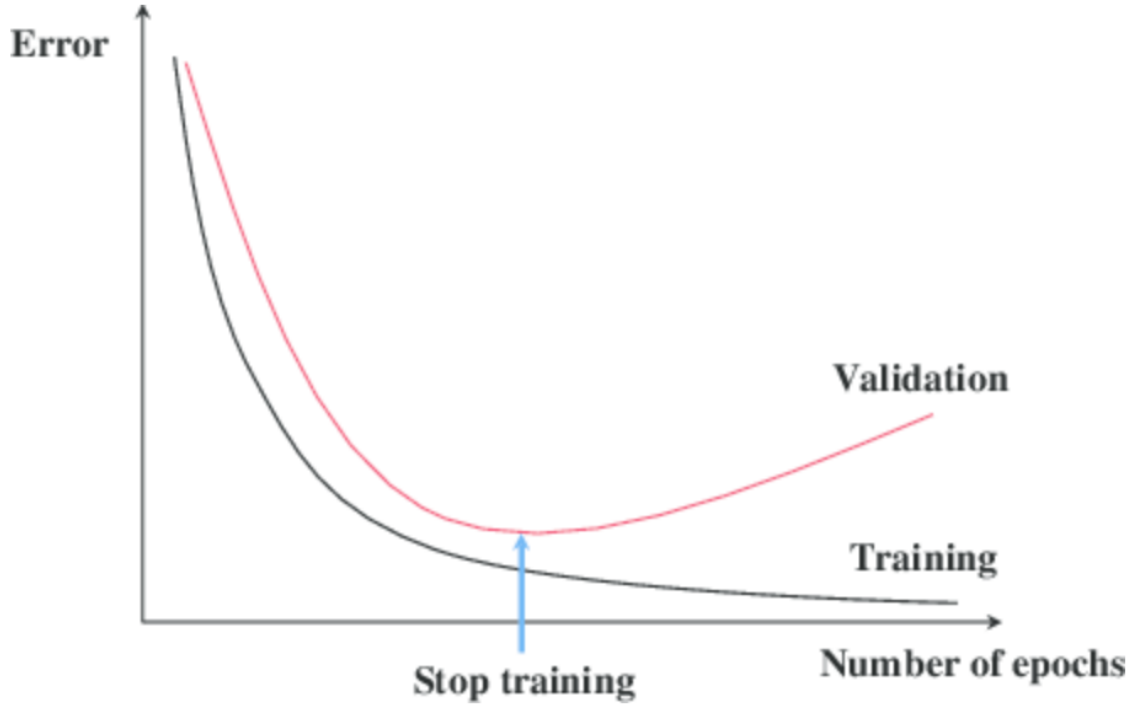
loss decay

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.

- **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 (Model1 to Model3.png) 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?

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

- Use CNN to detect regions with selective sweeps on real genomes

DL witchcraft