Machine Learning: Neural Nets

A Motivating Example: Ryan's Question

Let's think about using Euclidean distance here:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

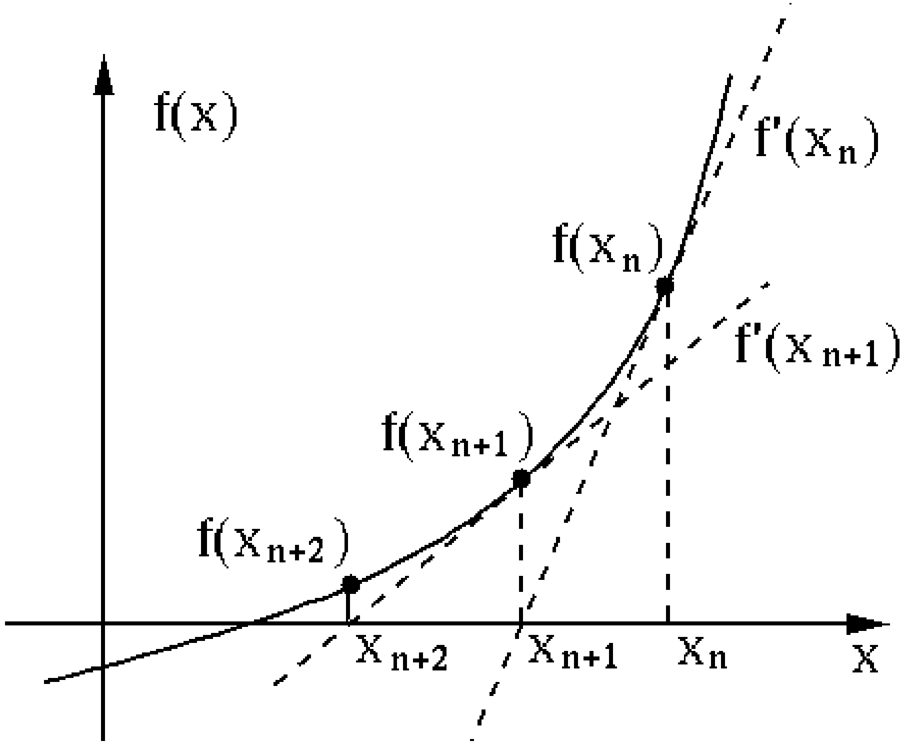
+

+

On Learning: Every Program “Learns”

Newton’s Method for Finding Roots

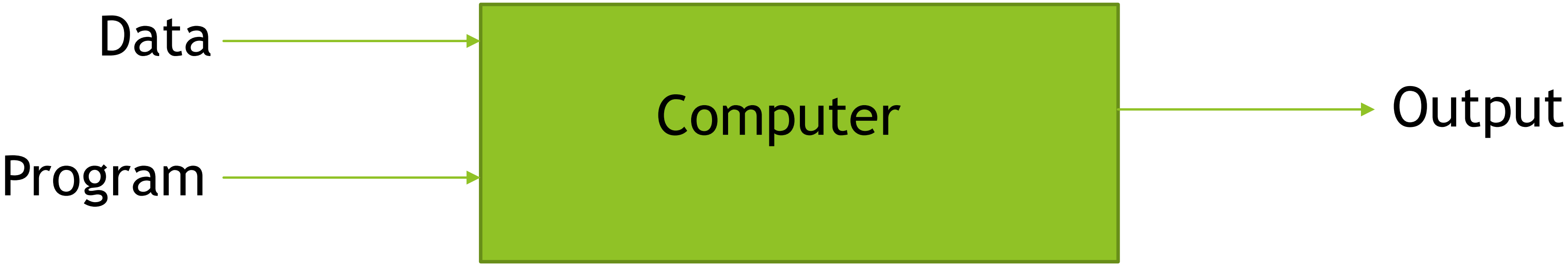
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



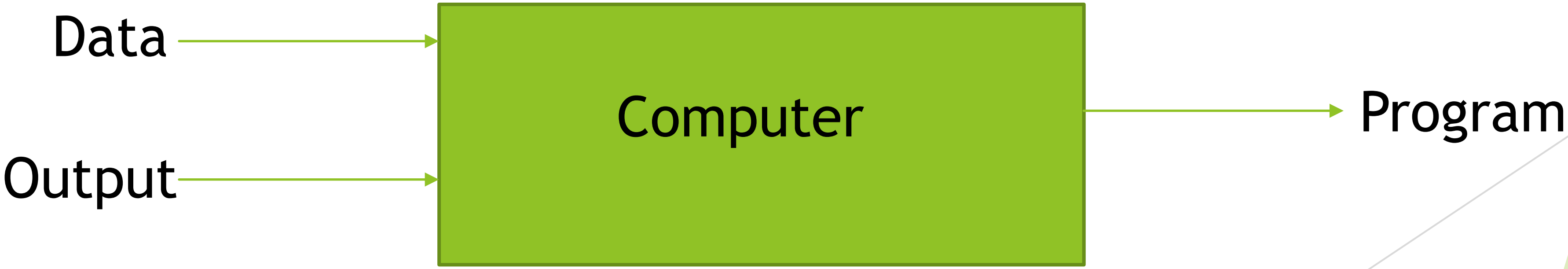
Newton’s Method “learns” roots

On Learning: Every Program "Learns"

Traditional Program: Newton's Method for finding roots



Machine Learning: Curve fitting by linear regression



Every Machine Learning Method Requires:

- Choosing training data and an evaluation method
- Representation of the features
  - Representing examples by feature vectors (Remember our cobra and dart frog?!)
- Distance metric for feature vectors
  - What kind of distance between the vectors do we want to consider?
- Objective function and constraints
- Optimization method for learning the model
  - How do we improve our model based on its performance?

How can we learn to identify images?

Input

An Image

"Photo of a number"

Model

Output

Knowledge about that image

"Value of the number"

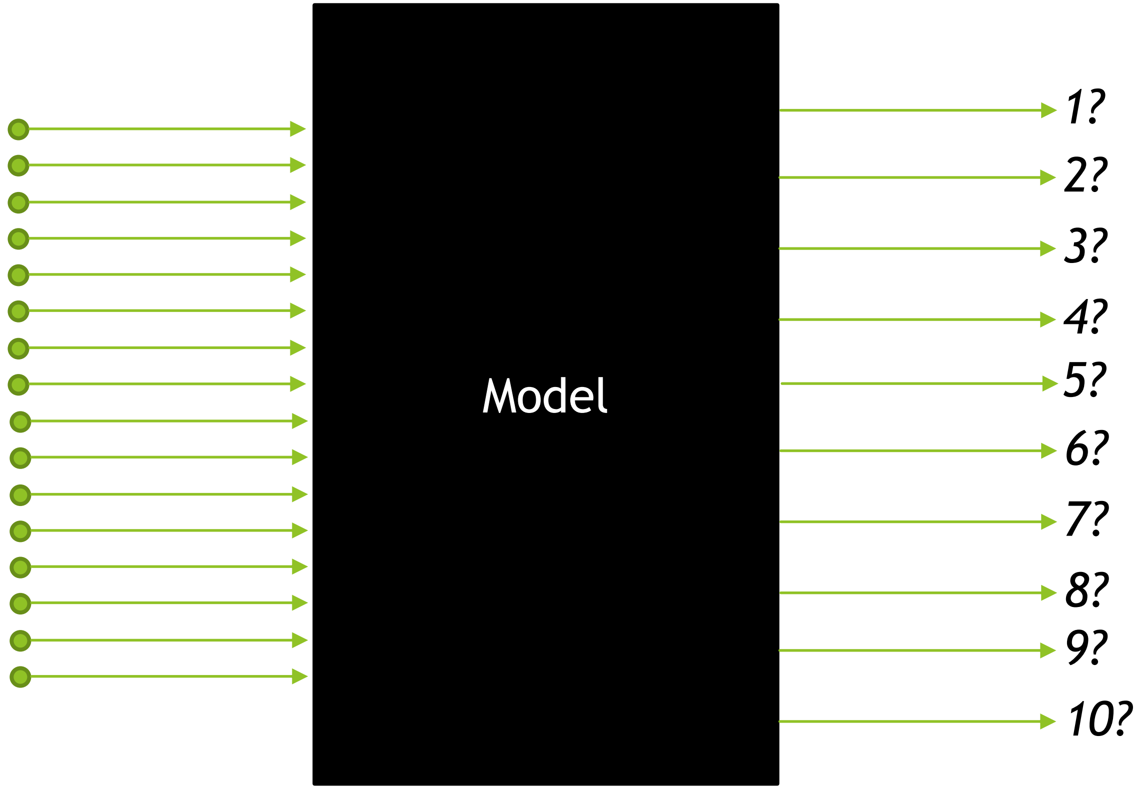
How can we learn to identify images?

“Representation of the features”

Input “Layer”

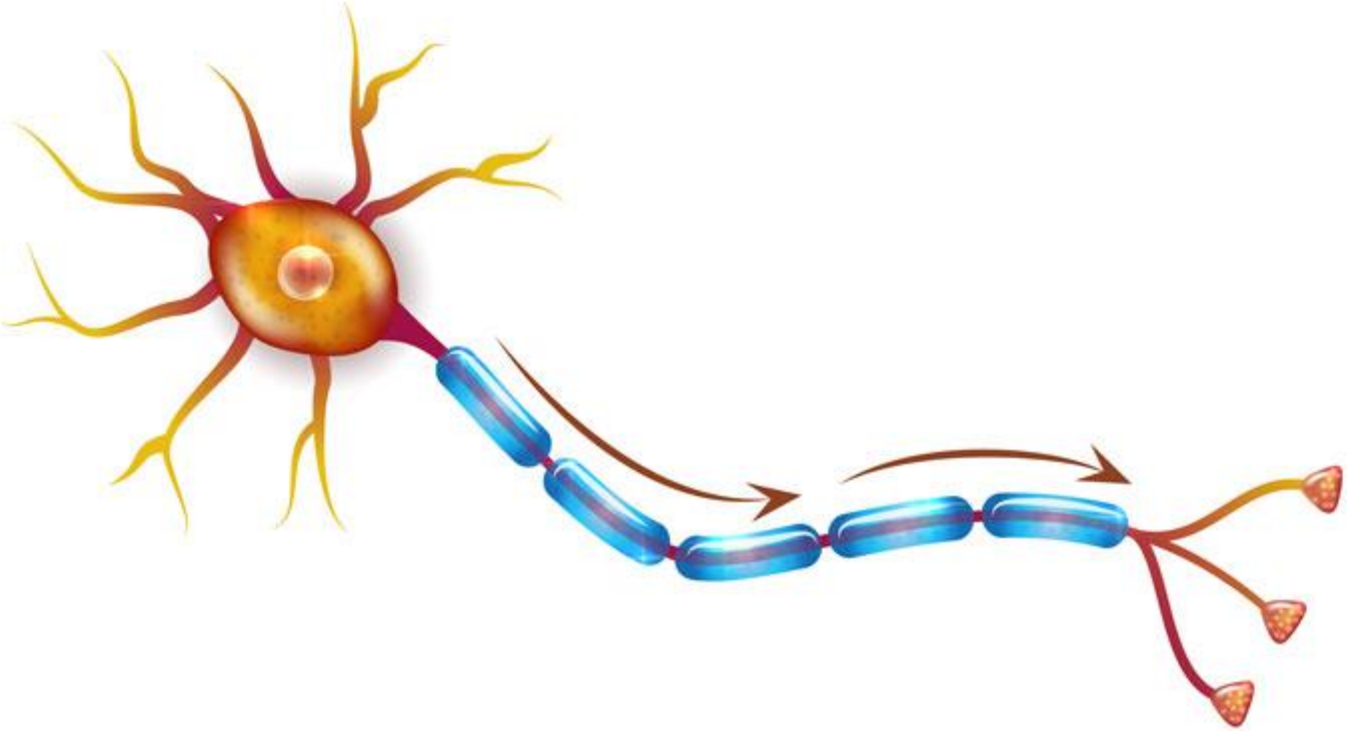
Output “Layer”

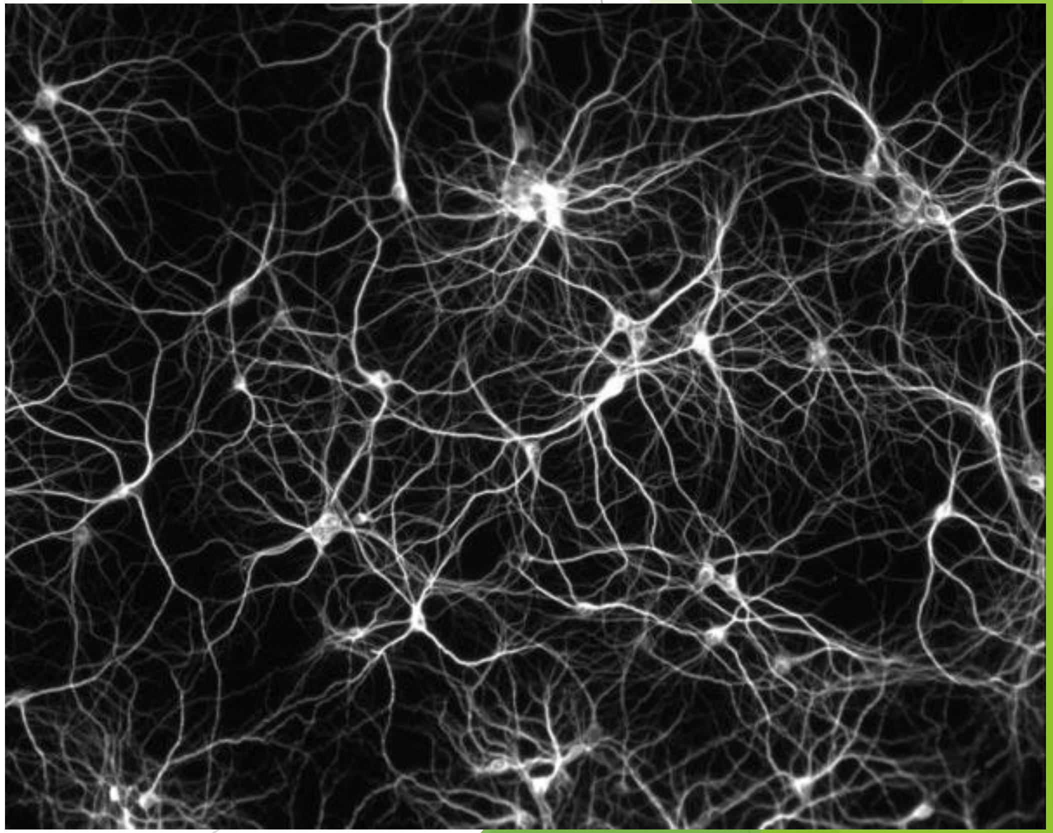
Pixel Vector



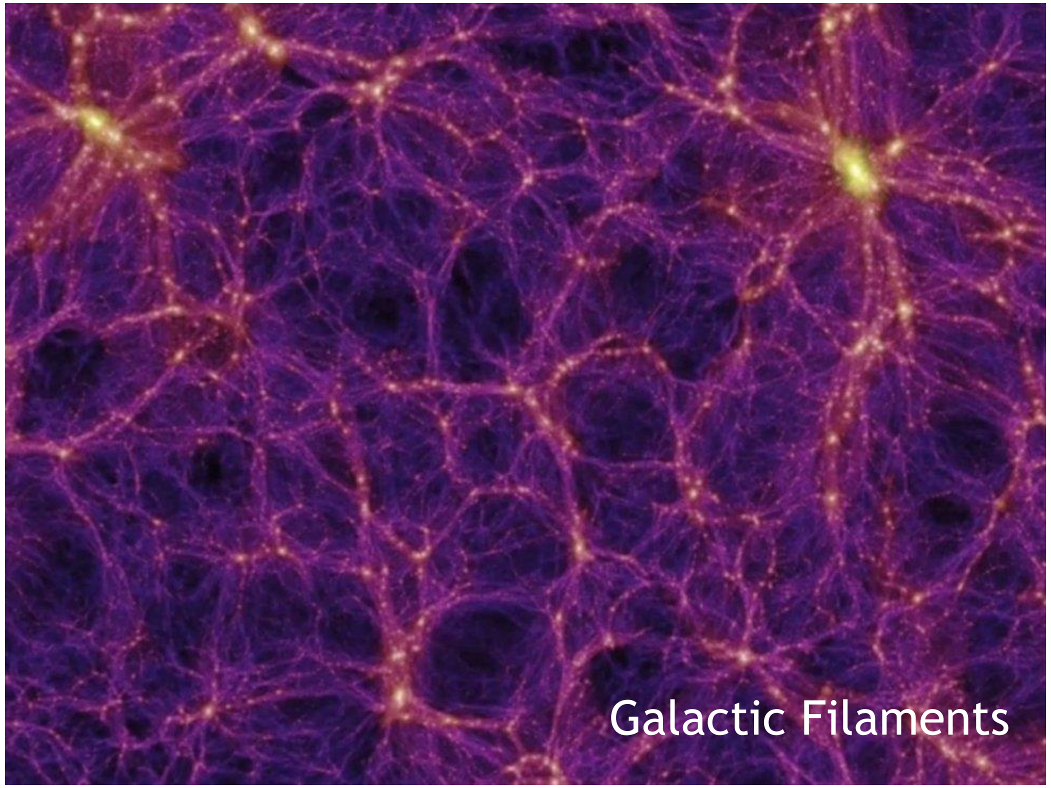
Probability of each number?

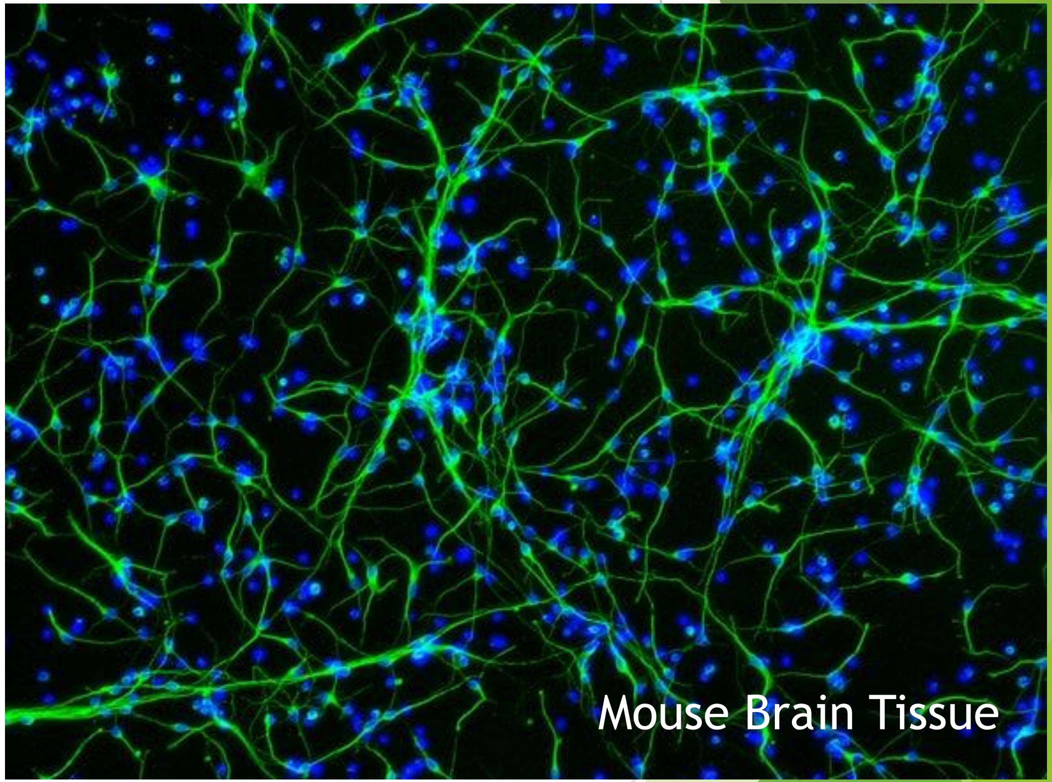
Mathematicians turn to Nature



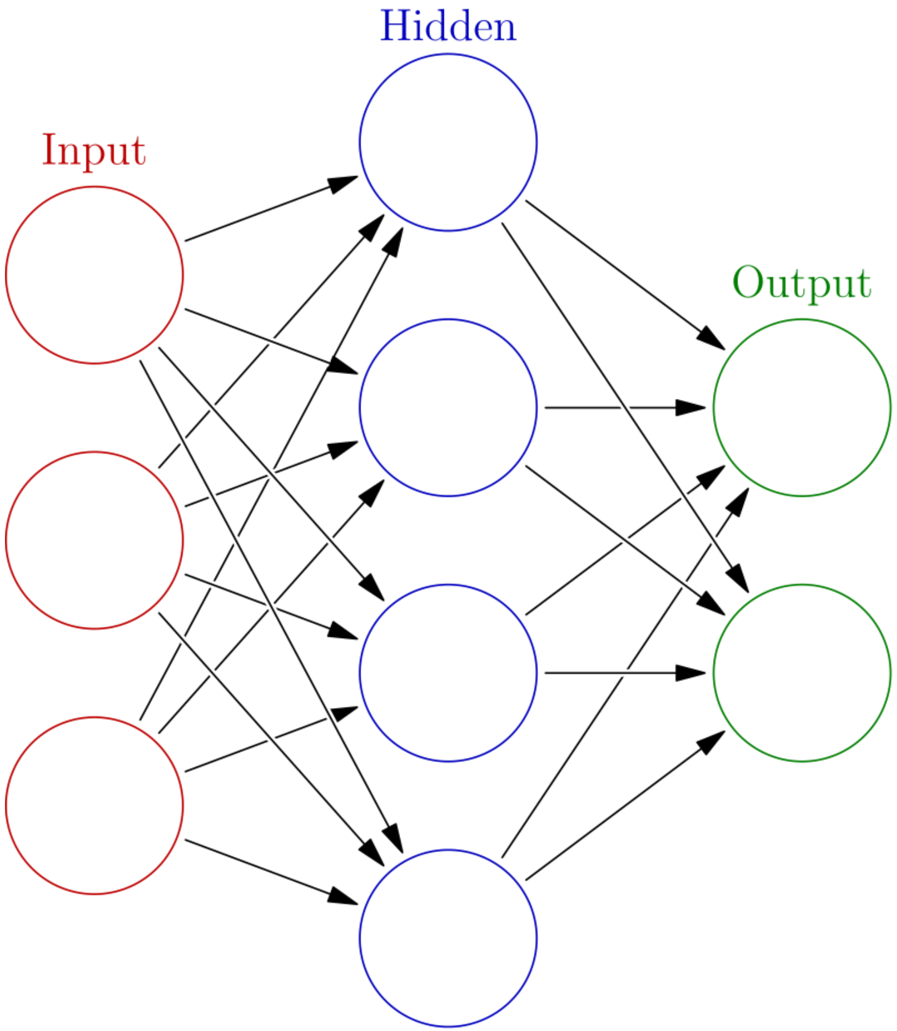


Mathematicians turn to Nature





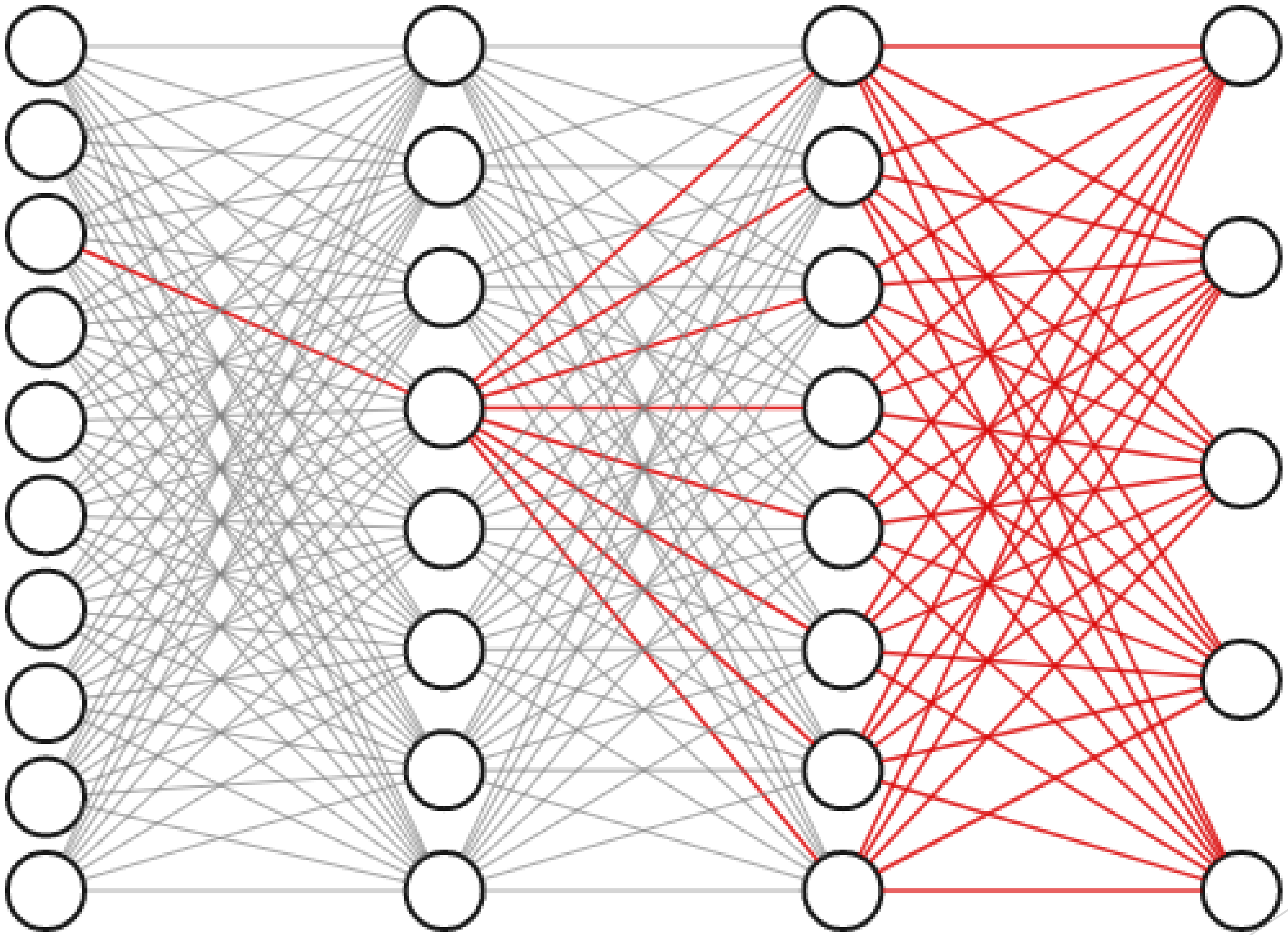
The Artificial Neural Net (ANN)



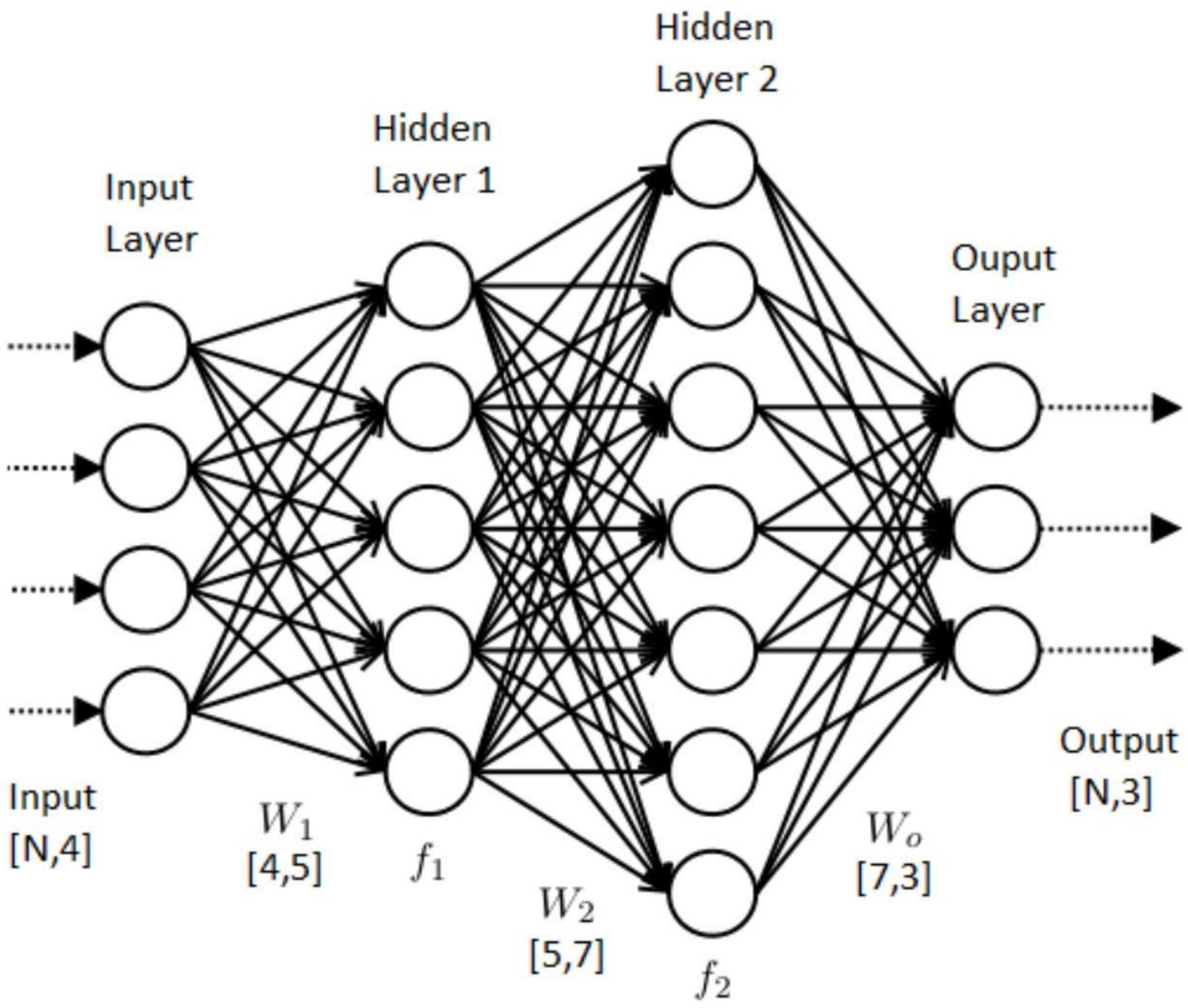
- Weights (along edges)
- Activation Function

https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Learning Weights (training your ANN)



Learning Weights (training your ANN)



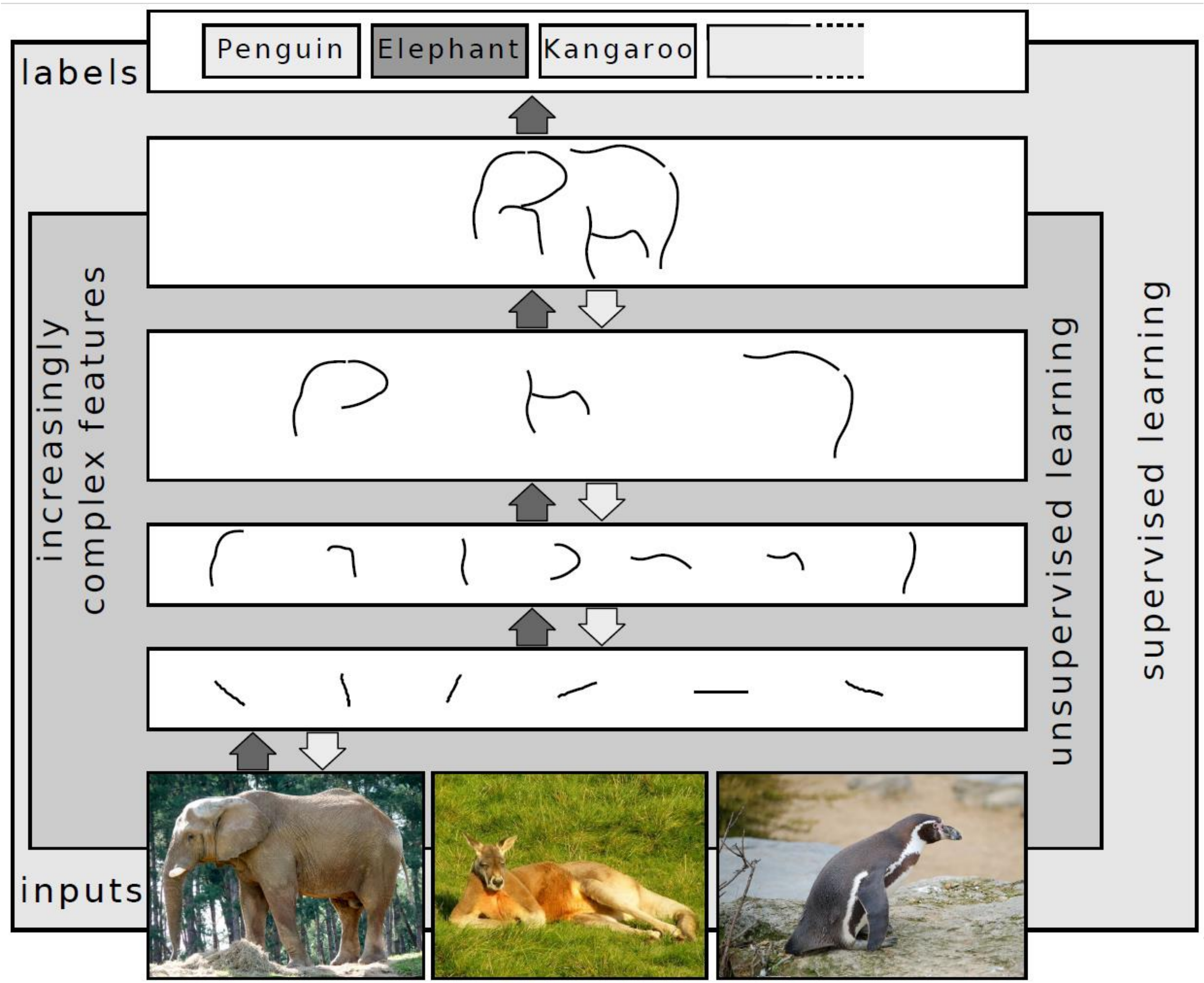
A Motivating Example: Ryan’s Question

Instead of looking at a distance, we identify common features:

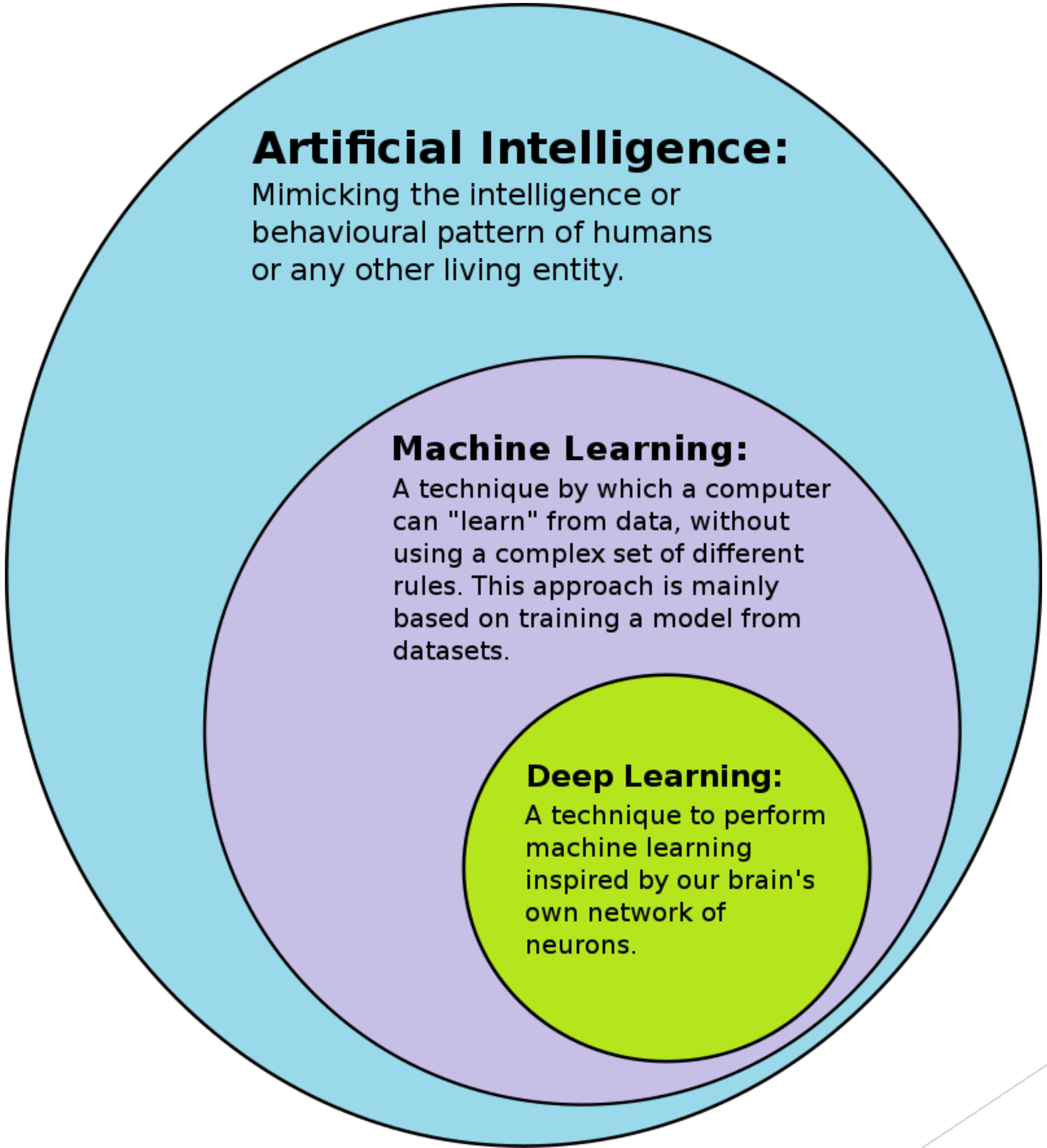
+

+

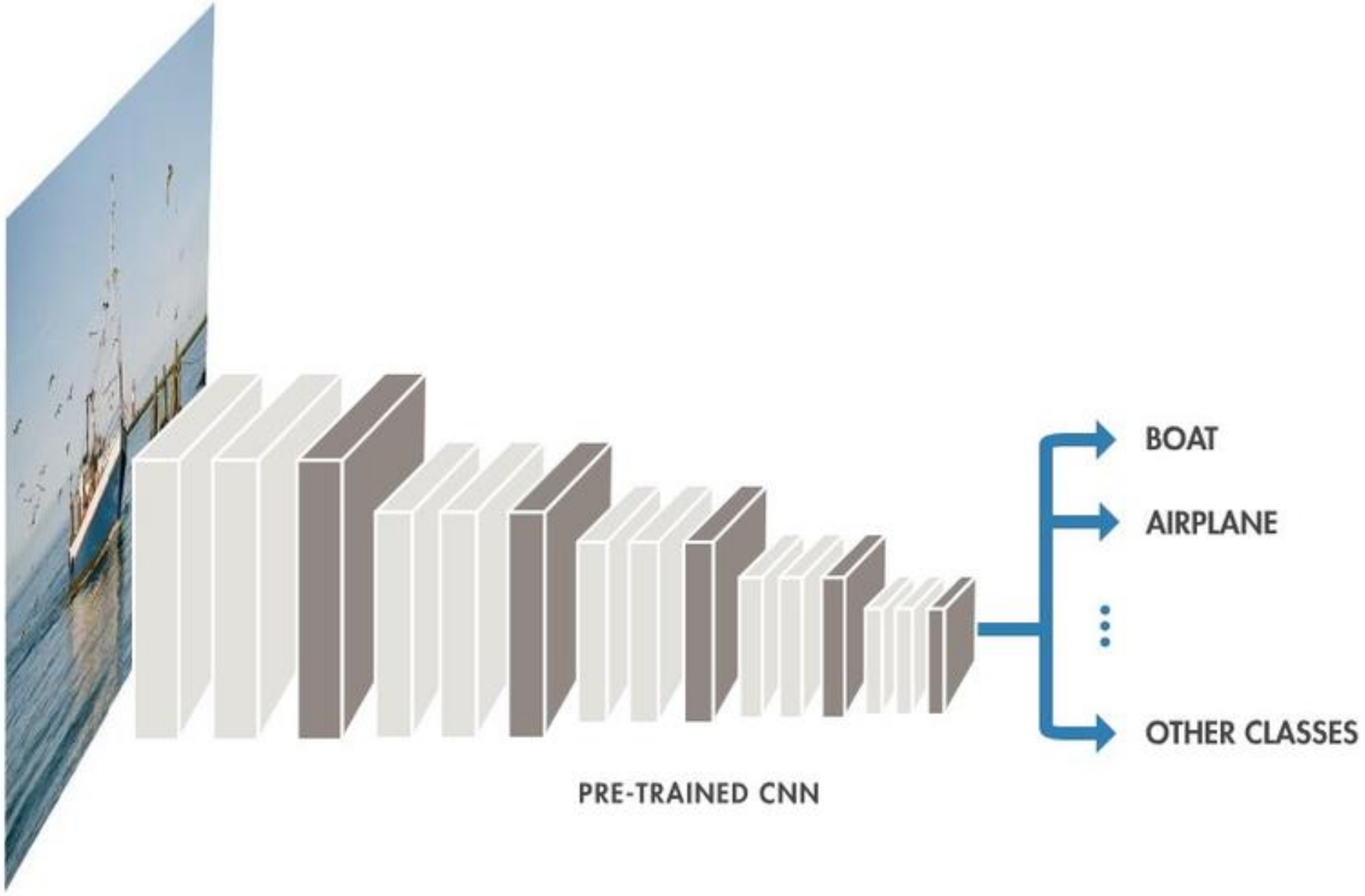
Ryan’s Question



Deep Learning



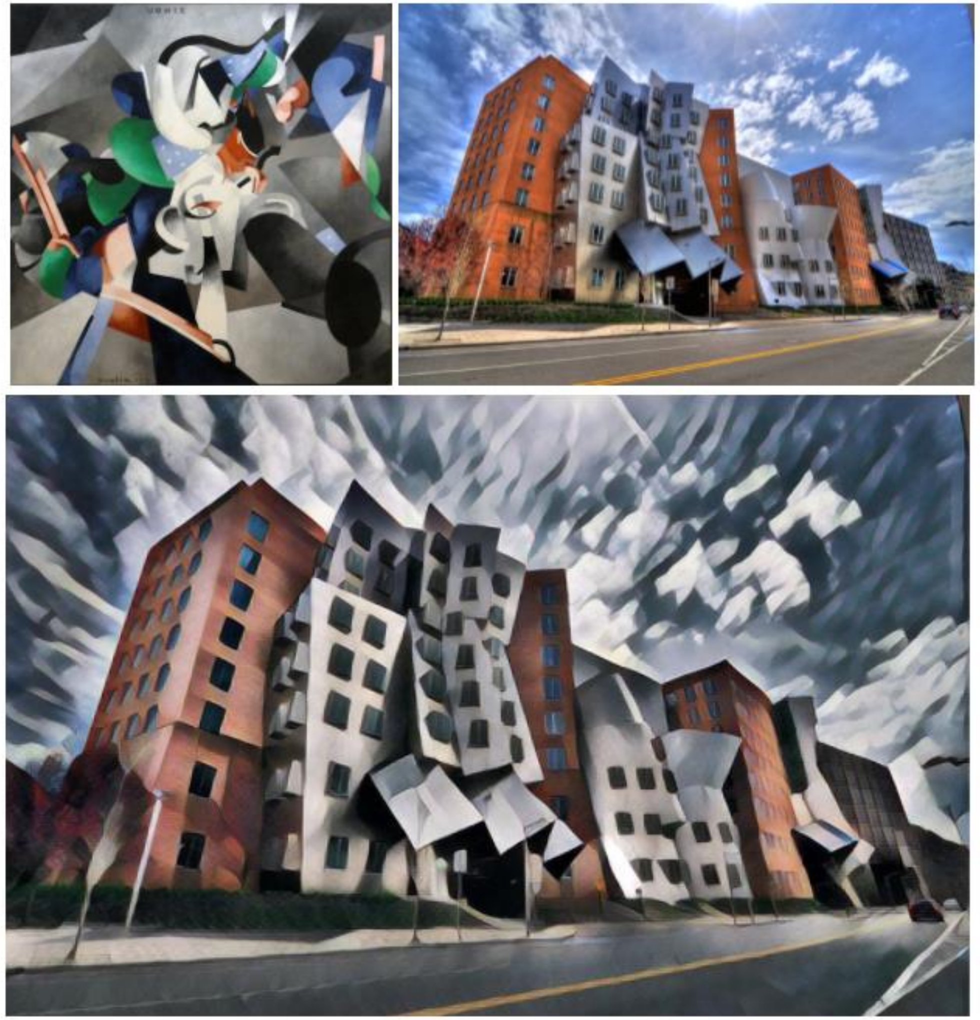
Deep Learning



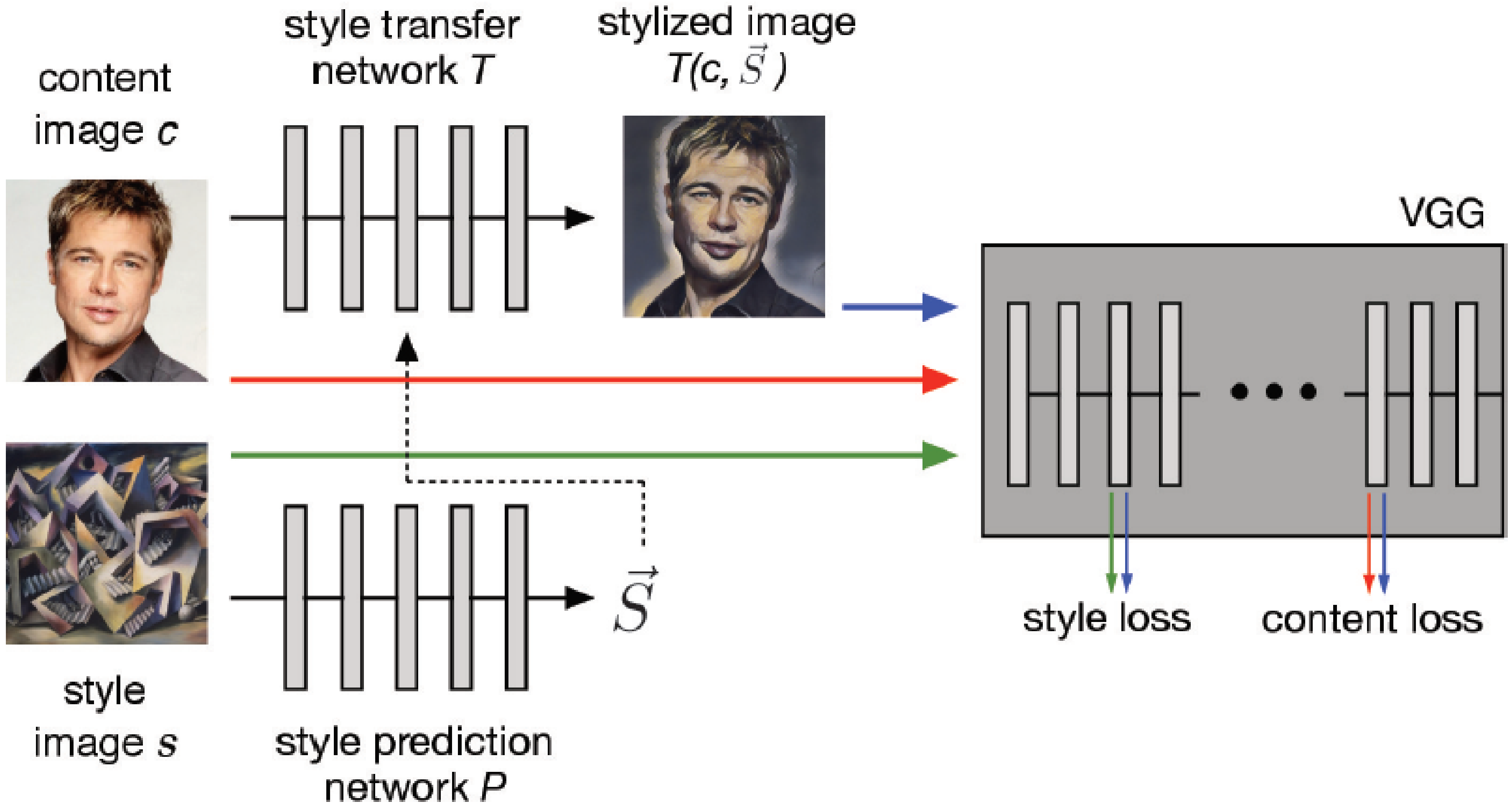
MNIST Handwriting Classifier

CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
Linear Classifiers			
linear classifier (1-layer NN)	none	12.0	LeCun et al. 1998
linear classifier (1-layer NN)	deskewing	8.4	LeCun et al. 1998
pairwise linear classifier	deskewing	7.6	LeCun et al. 1998
K-Nearest Neighbors			
K-nearest-neighbors, Euclidean (L2)	none	5.0	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	none	2.83	Kenneth Wilder, U. Chicago
K-nearest-neighbors, Euclidean (L2)	deskewing	2.4	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	deskewing, noise removal, blurring	1.80	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring	1.73	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 1 pixel shift	1.33	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 2 pixel shift	1.22	Kenneth Wilder, U. Chicago
K-NN with non-linear deformation (IDM)	shiftable edges	0.54	Keysers et al. IEEE PAMI 2007
K-NN with non-linear deformation (P2DHMDM)	shiftable edges	0.52	Keysers et al. IEEE PAMI 2007
K-NN, Tangent Distance	subsampling to 16x16 pixels	1.1	LeCun et al. 1998
K-NN, shape context matching	shape context feature extraction	0.63	Belongie et al. IEEE PAMI 2002
large conv. net, unsup pretraining [elastic distortions]	none	0.39	Ranzato et al., NIPS 2006
large conv. net, unsup pretraining [no distortions]	none	0.53	Jarrett et al., ICCV 2009
large/deep conv. net, 1-20-40-60-80-100-120-120-10 [elastic distortions]	none	0.35	Ciresan et al. IJCAI 2011
committee of 7 conv. net, 1-20-P-40-P-150-10 [elastic distortions]	width normalization	0.27 +-0.02	Ciresan et al. ICDAR 2011
committee of 35 conv. net, 1-20-P-40-P-150-10 [elastic distortions]	width normalization	0.23	Ciresan et al. CVPR 2012

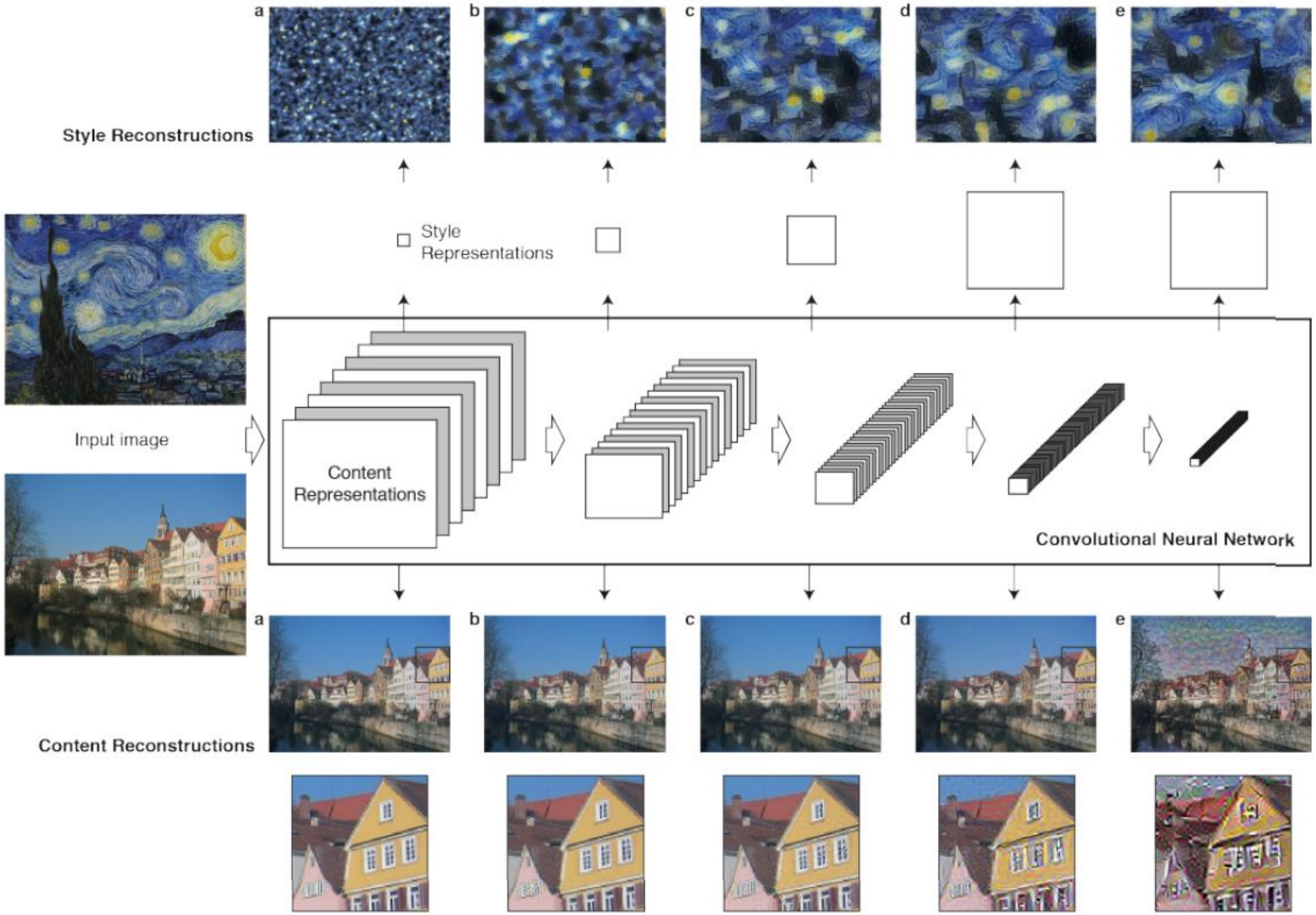
Fast Style Transfer



Fast Style Transfer



Fast Style Transfer



Now you try!

- “tf2_arbitrary_image_stylization.ipynb”
- Use TensorFlow to create style transferred images