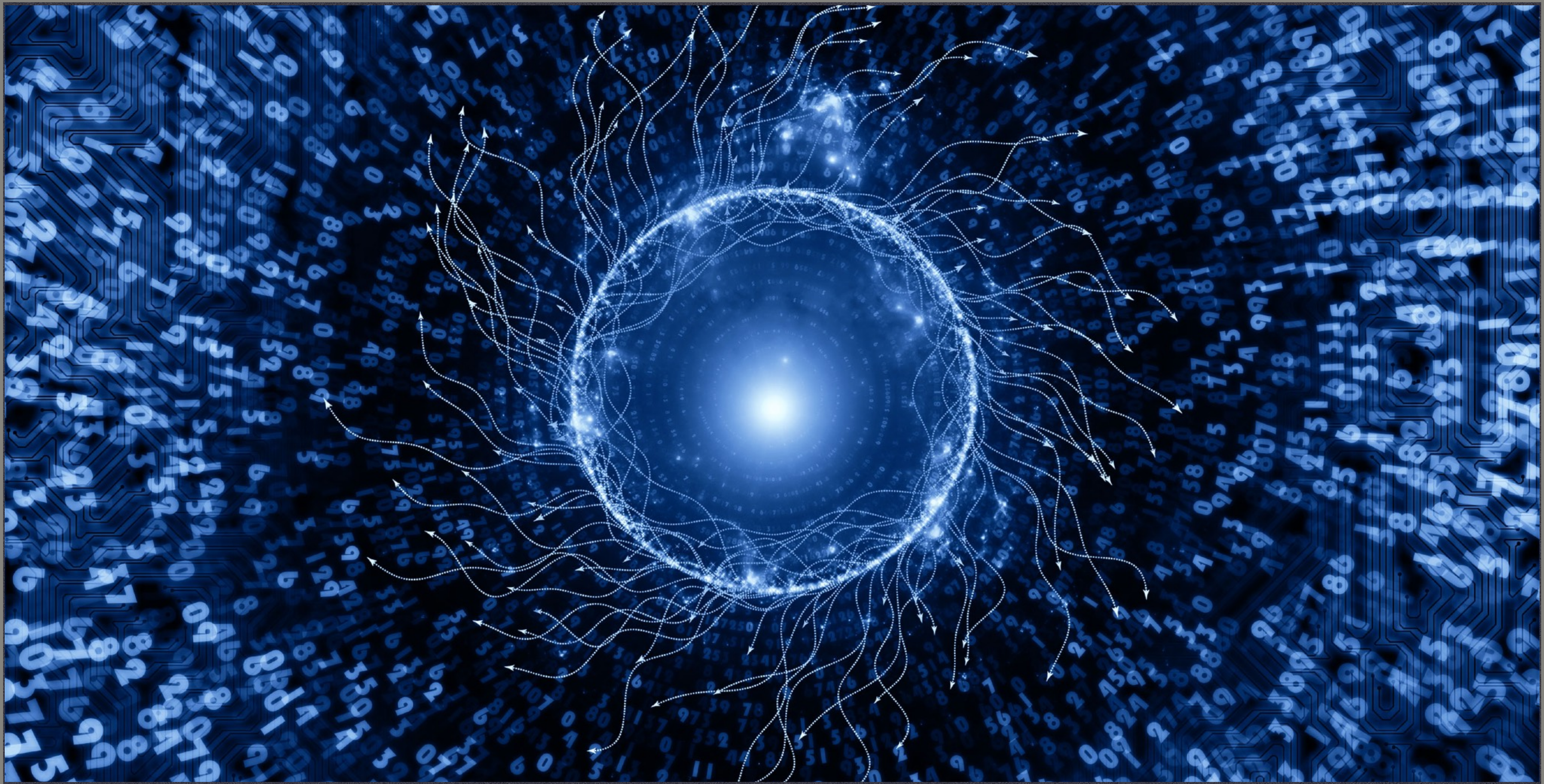
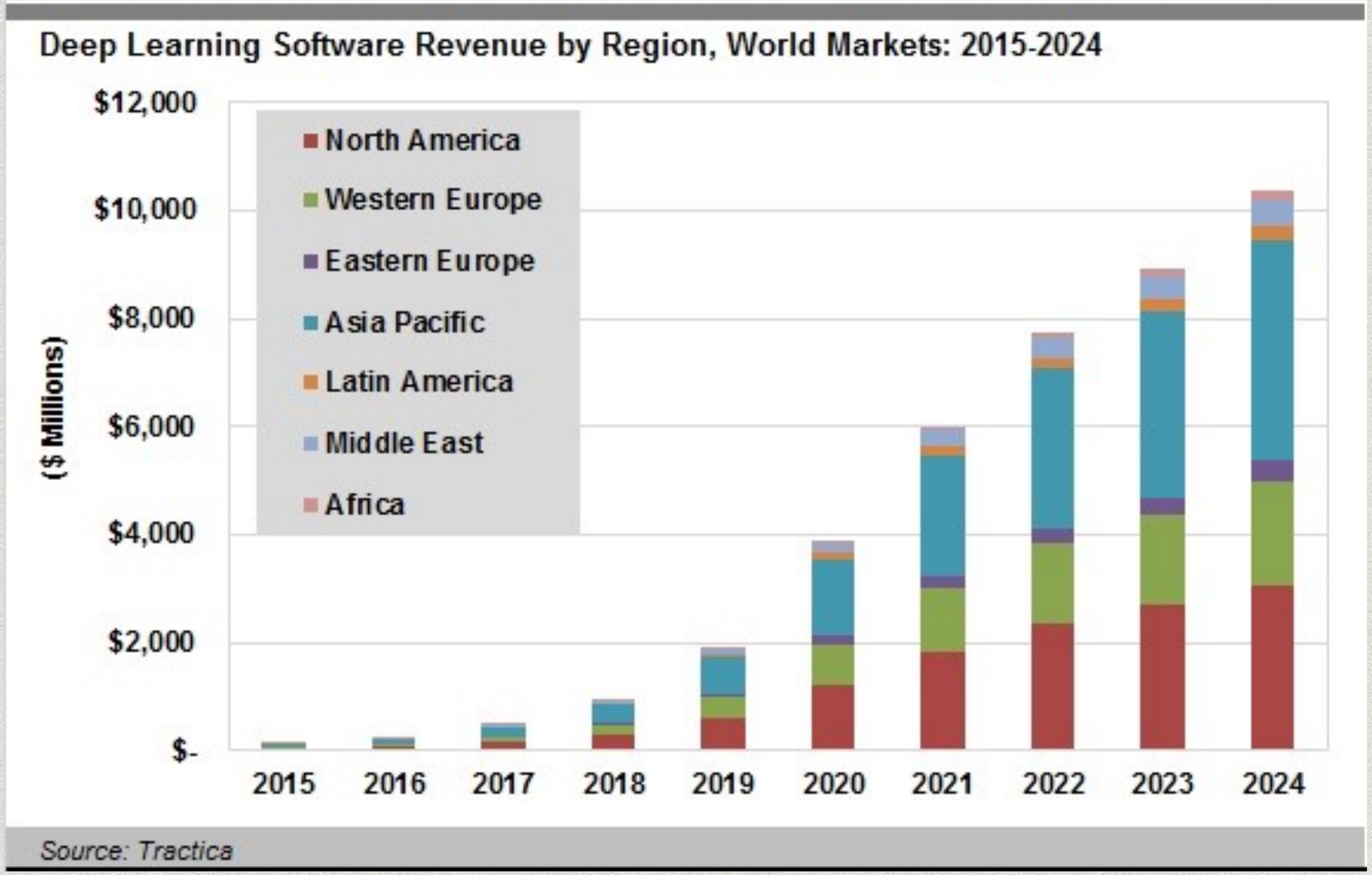


DEEP LEARNING

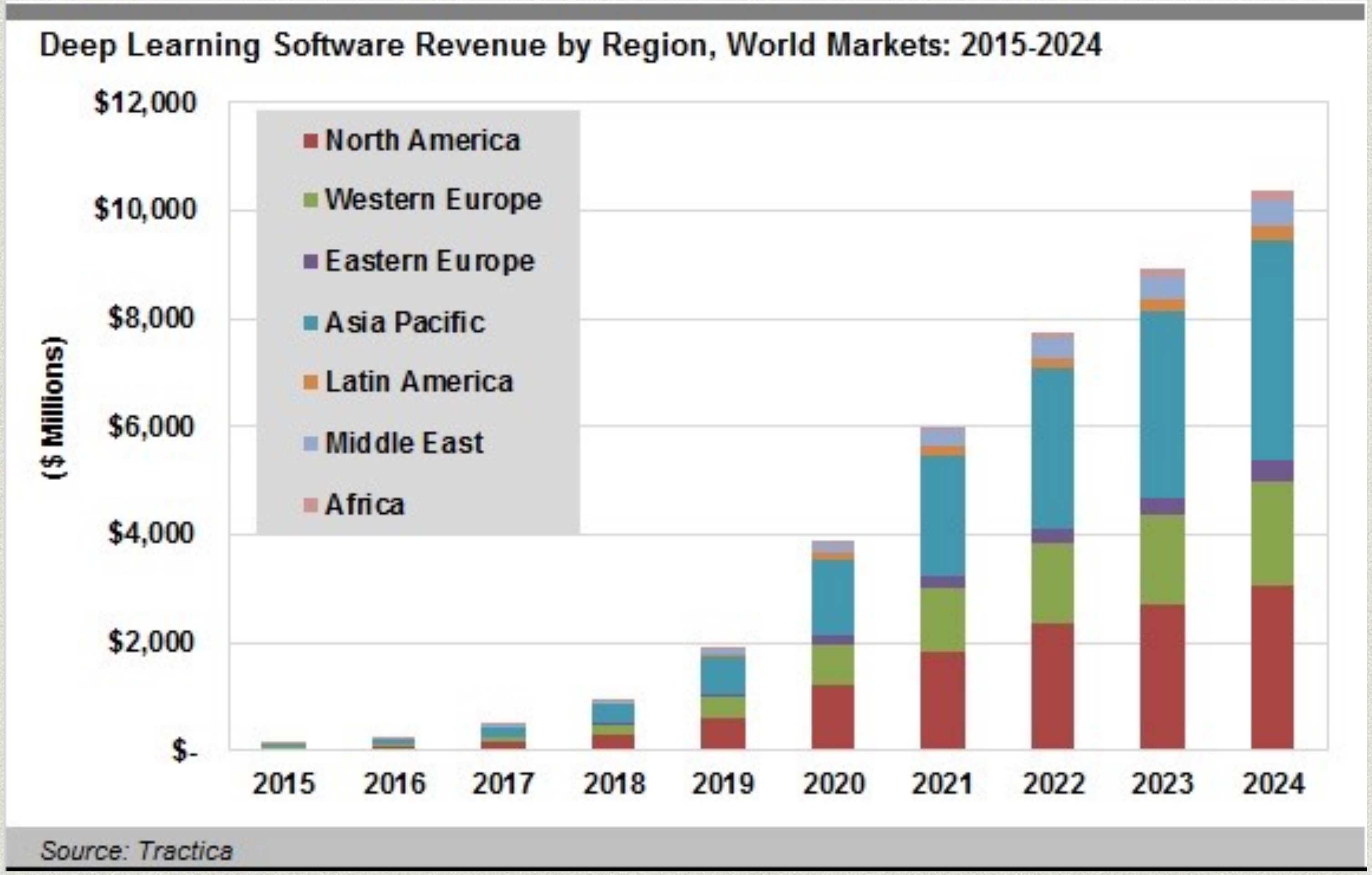
THE SOLUTION TO ALL MY PROBLEMS… RIGHT?

Corentin Lapeyre | COOP/CSG | 2017-06-27

The hype


Deep Learning Software Revenue by Region, World Markets: 2015-2024
$12,000
$10,000
$8,000
$6,000
$4,000
$2,000
$-
($ Millions)
North America
Western Europe
Eastern Europe
Asia Pacific
Latin America
Middle East
Africa
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Source: Tractica

The hype



- Artificial Intelligence (AI) and specifically Deep Learning (DL) are exploding
- Forrester: 16% of current US jobs replaced by 2025

Deep Learning everywhere

Deep Learning for everyone

Deep Learning for everyone

TensorFlow
Google

Caffe

theano

Deep Learning for everyone

TensorFlow
Google

Caffe

torch

theano

Deep Learning for everyone

TensorFlow
Google

Caffe

Keras

DL4j
Java

torch

theano

cuDNN

mxnet

. . .

Deep Learning for everyone

TensorFlow
Google

Caffe

Keras

DL4j
Java

torch

theano

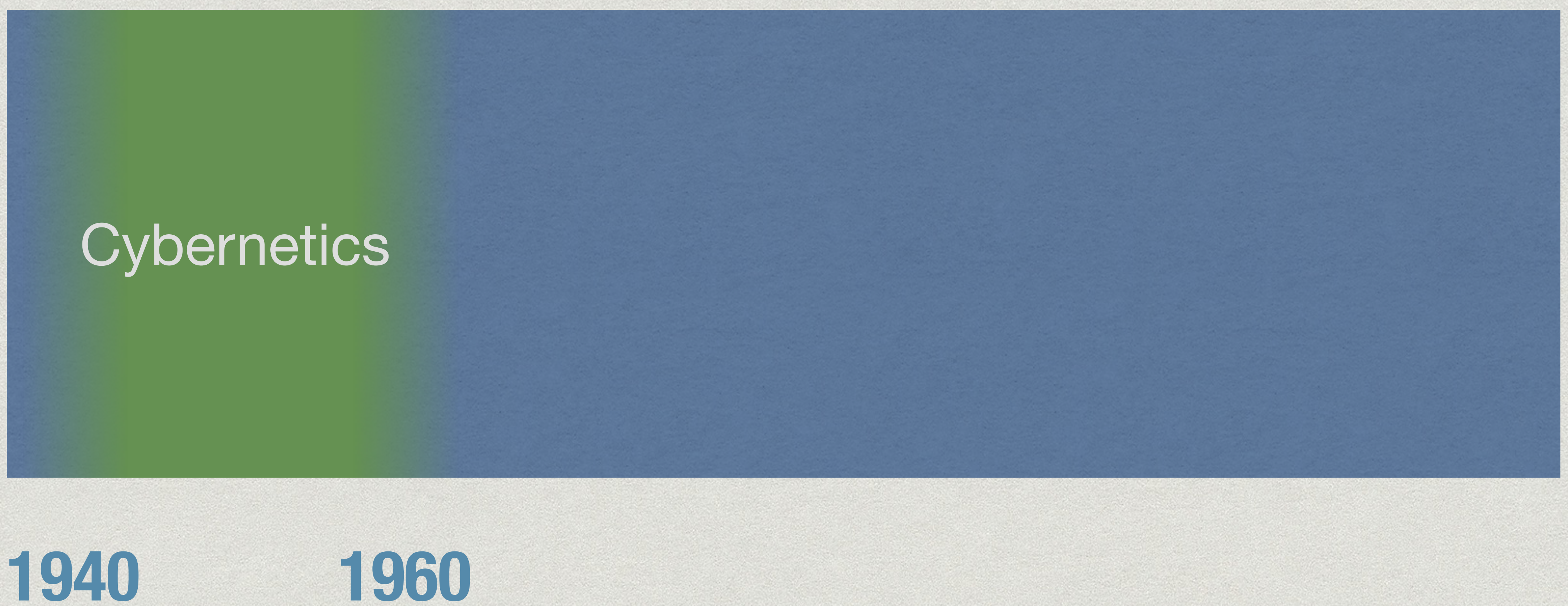
cuDNN

mxnet

. . .

Everybody wants you to use their framework

New stuff?

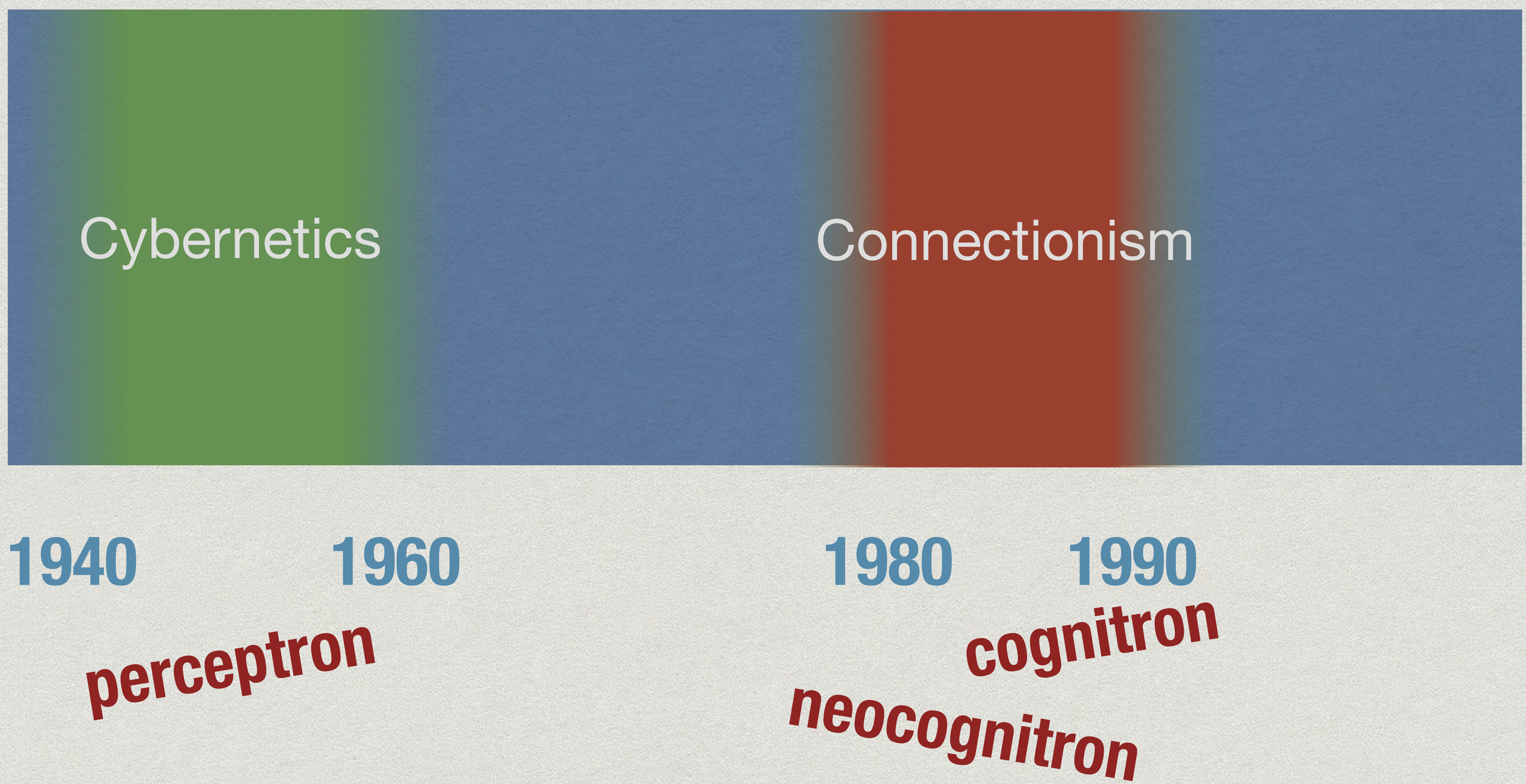


New stuff?

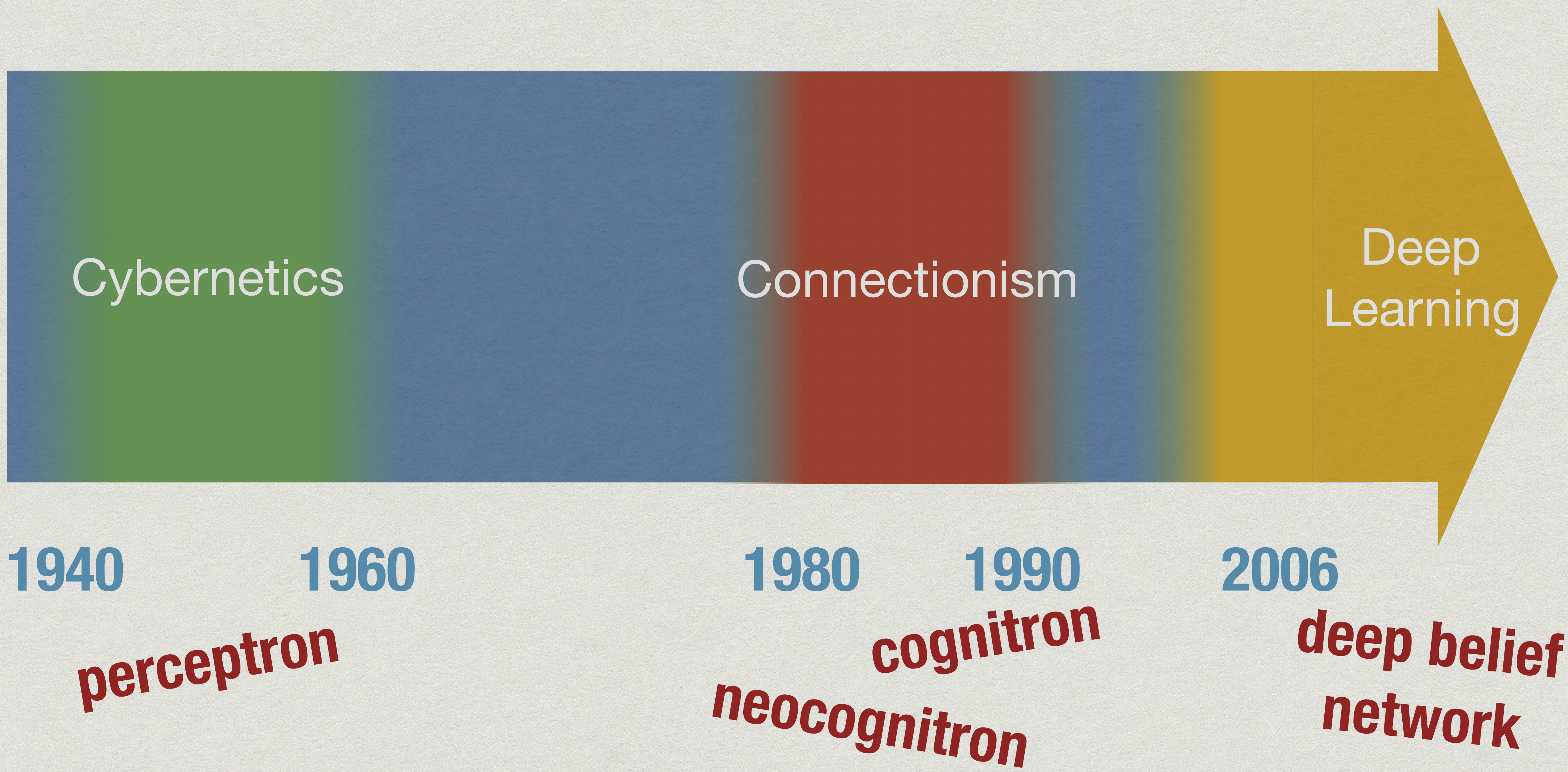
1940 **1960**

perceptron

New stuff?


Cybernetics
Connectionism
1940
1960
1980
1990
perceptron
cognitron
neocognitron

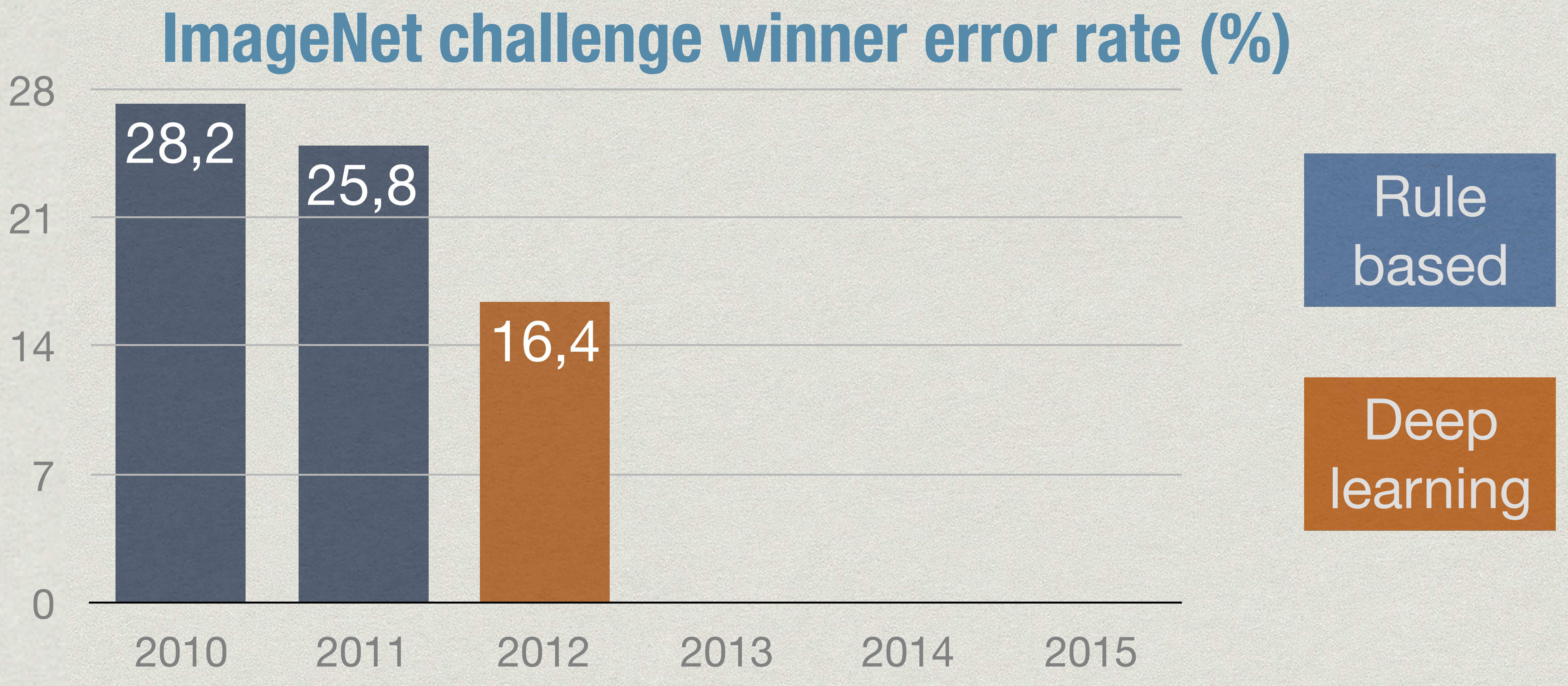
New stuff?



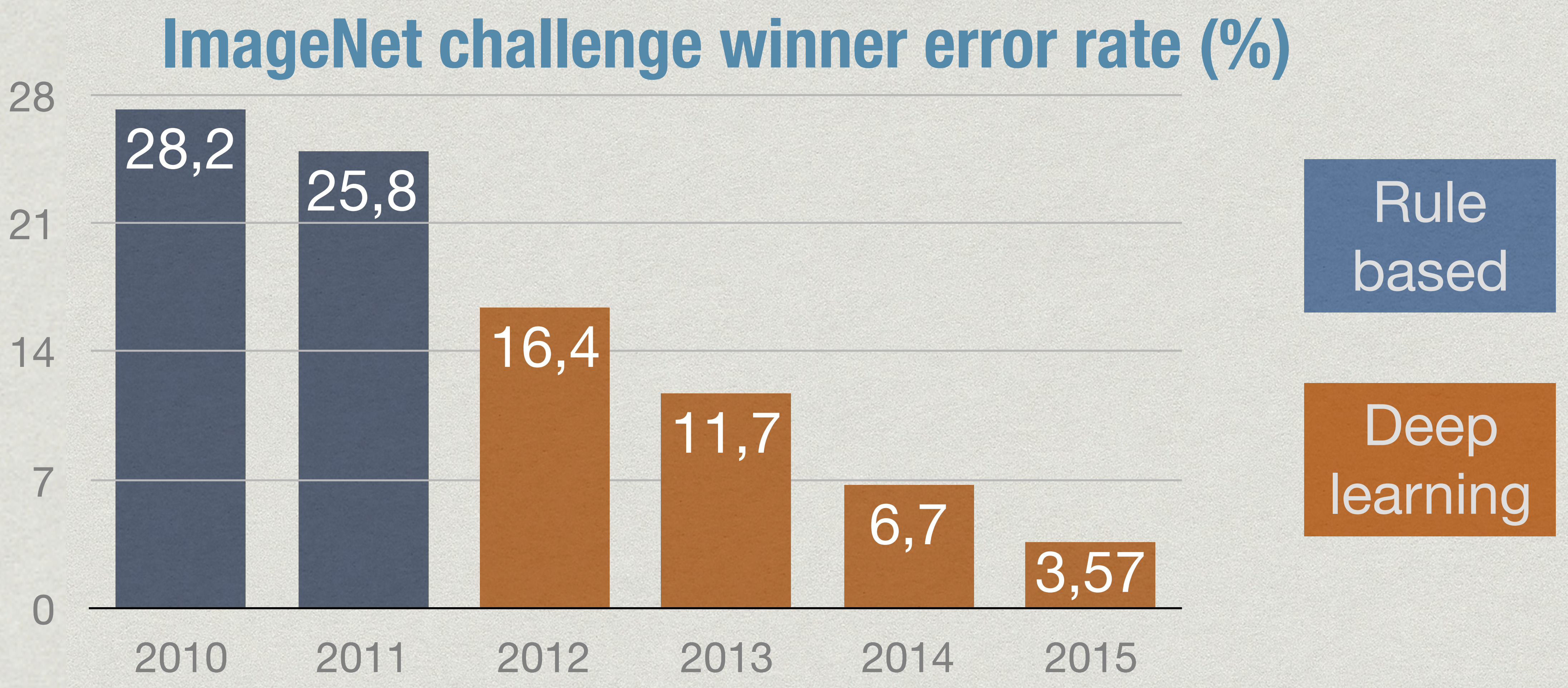
« Deep Learning » is **not** a new idea. It's a new set of techniques, combined with increased computational power

But the hype is new

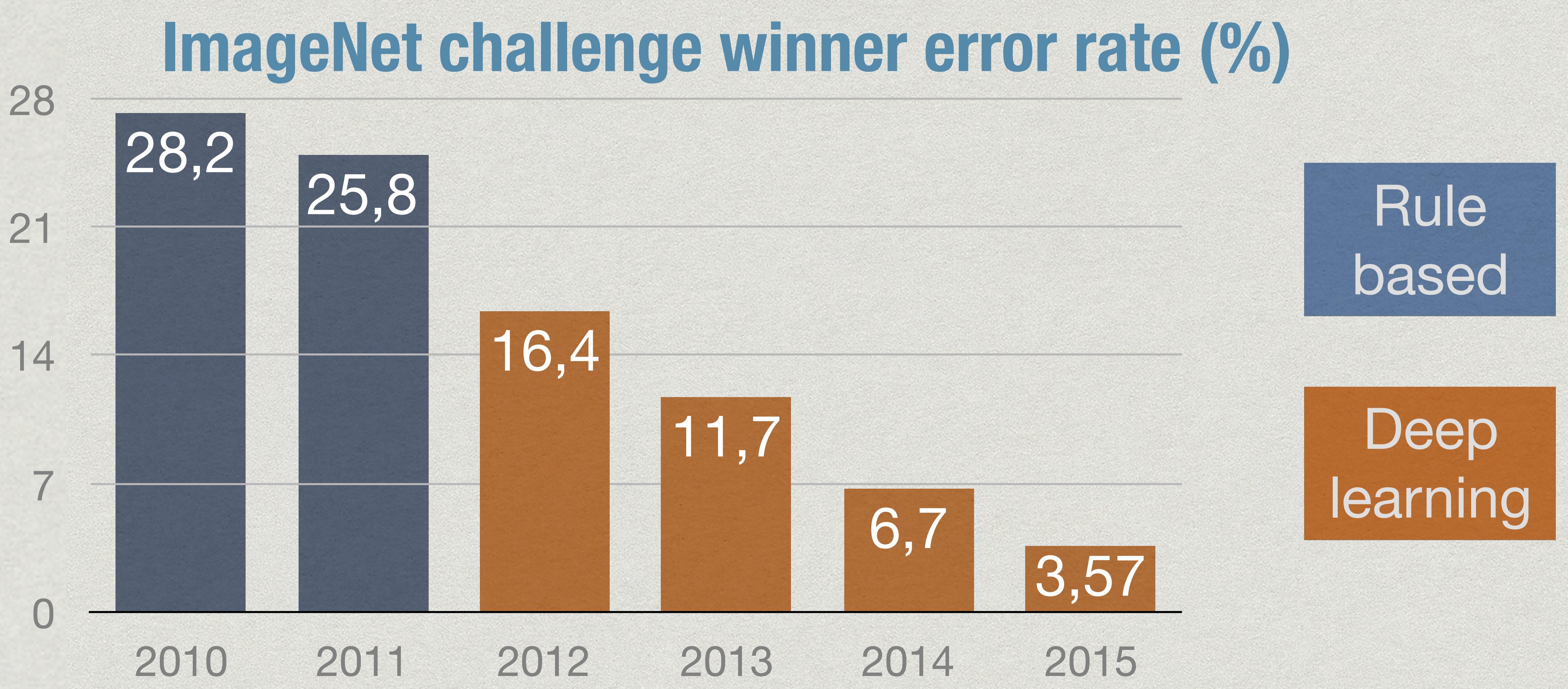
But the hype is new



But the hype is new

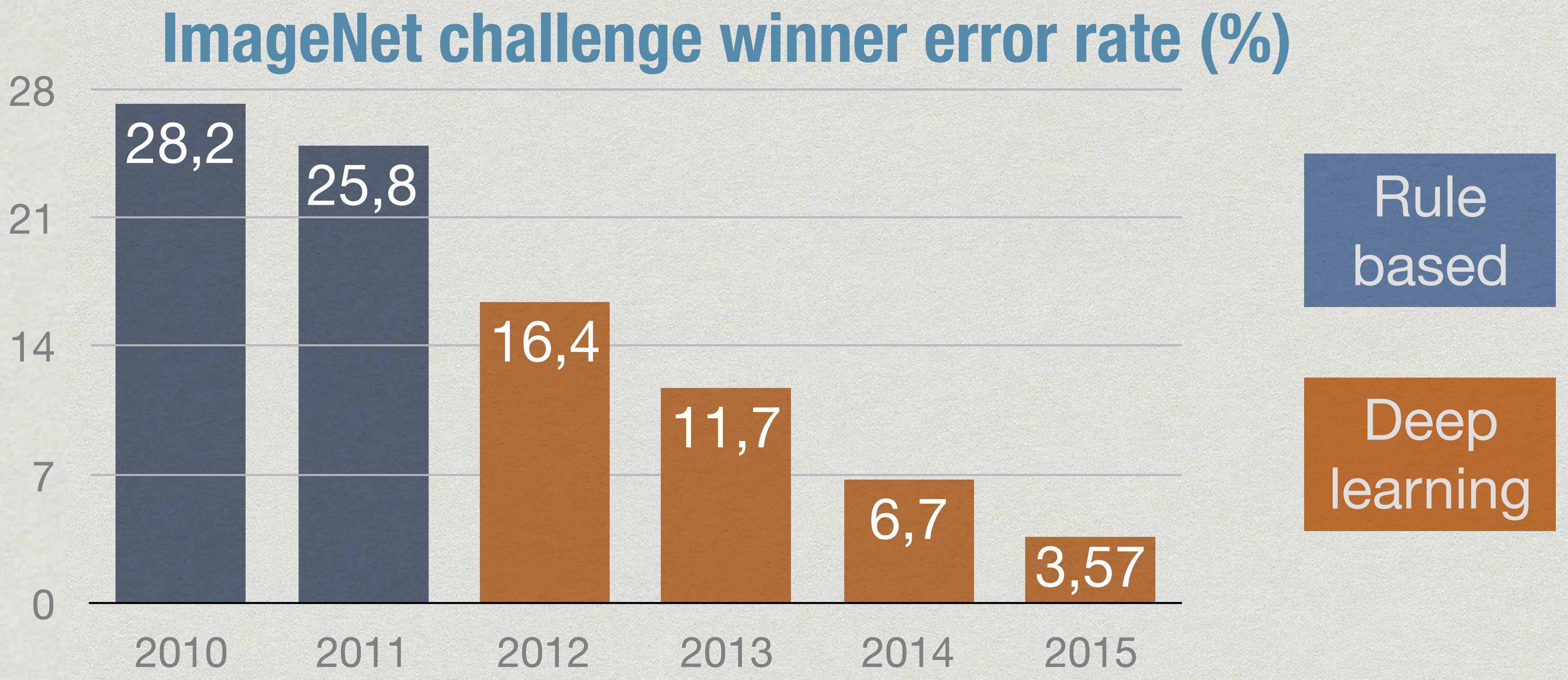


But the hype is new



Deep learning became the superstar in 2012. Since then, nothing compares to it for this challenge

But the hype is new



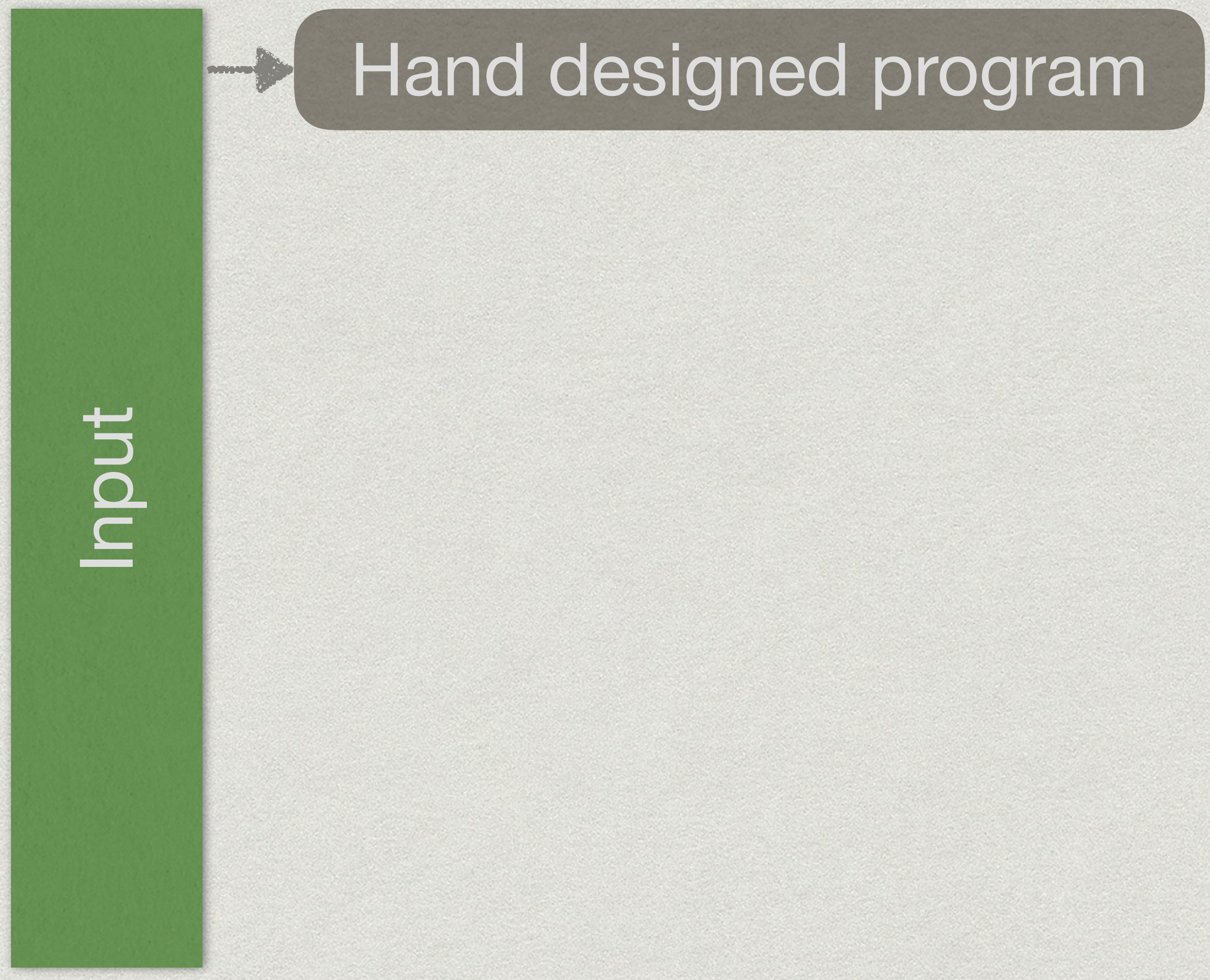
Same happened to the PASCAL Visual Object Classes challenge, etc…

Deep learning became the superstar in 2012. Since then, nothing compares to it for this challenge

WHAT THE HECK IS DEEP LEARNING?

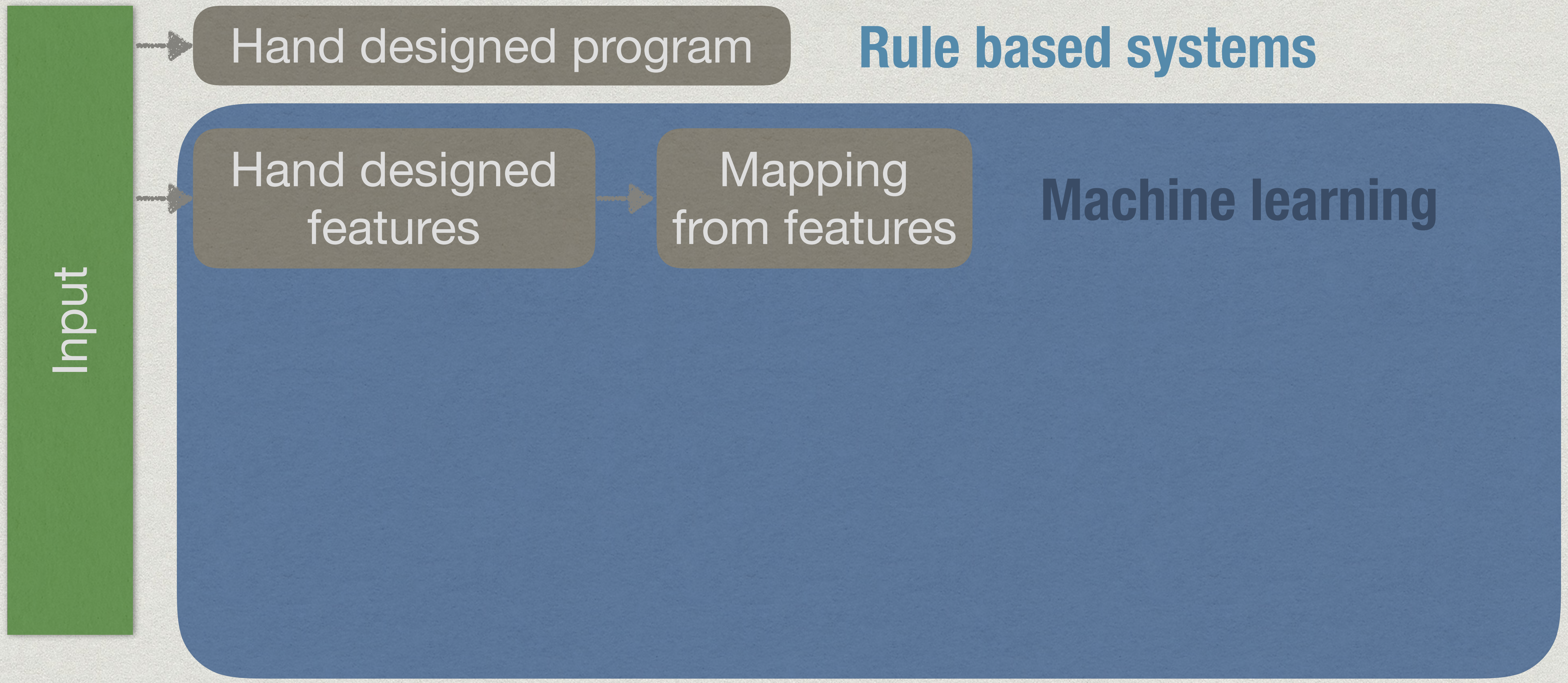
I DID A REGRESSION ONCE. IT’S DEEP LEARNING, RIGHT?

Machine / Deep learning

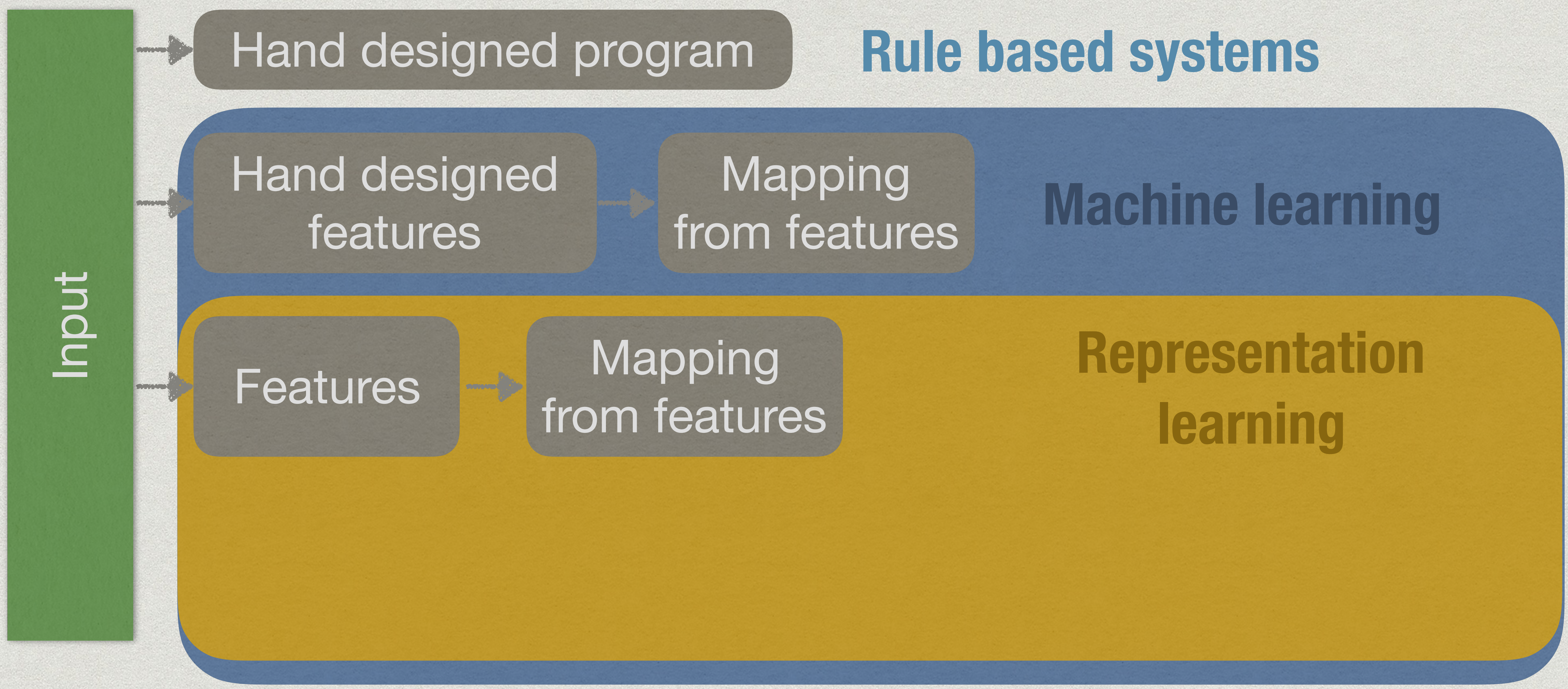


Rule based systems

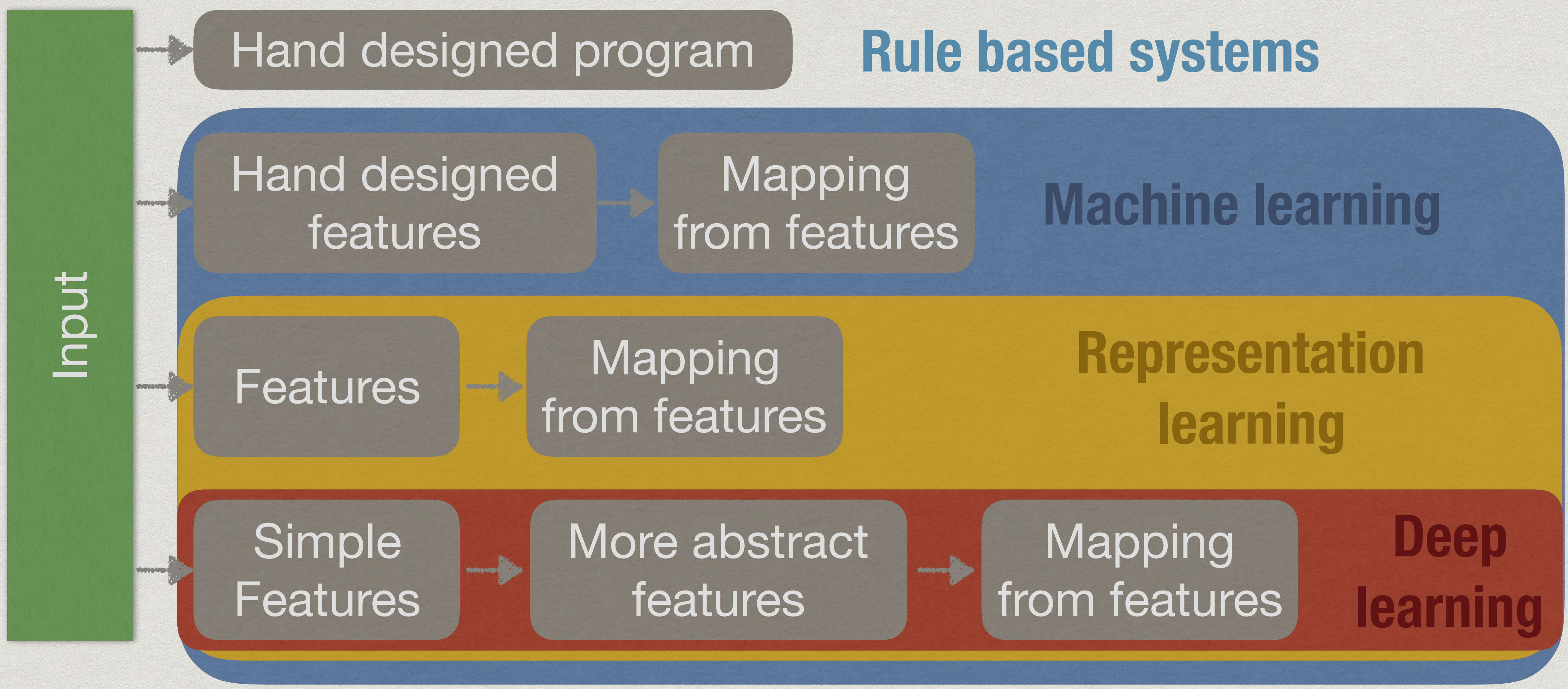
Machine / Deep learning



Machine / Deep learning

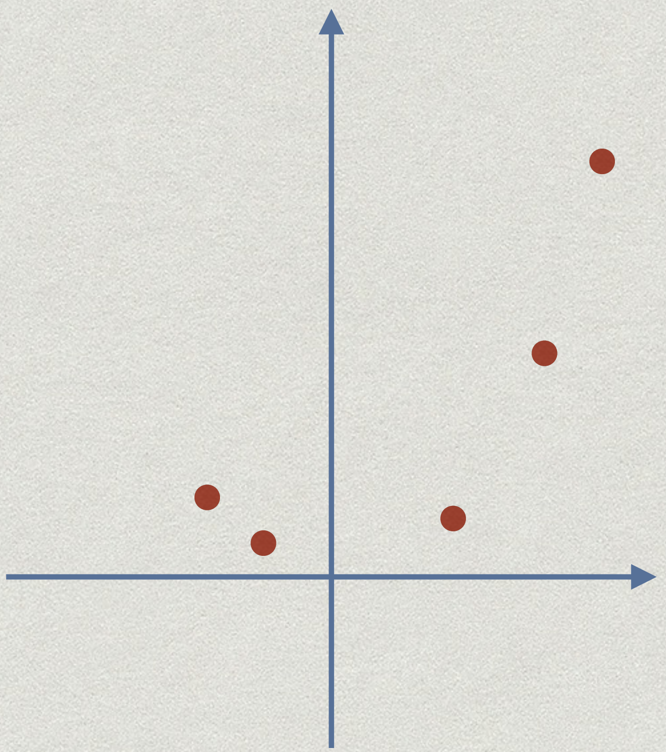


Machine / Deep learning



Deep learning is a subset of **Machine learning**

Machine / Deep learning



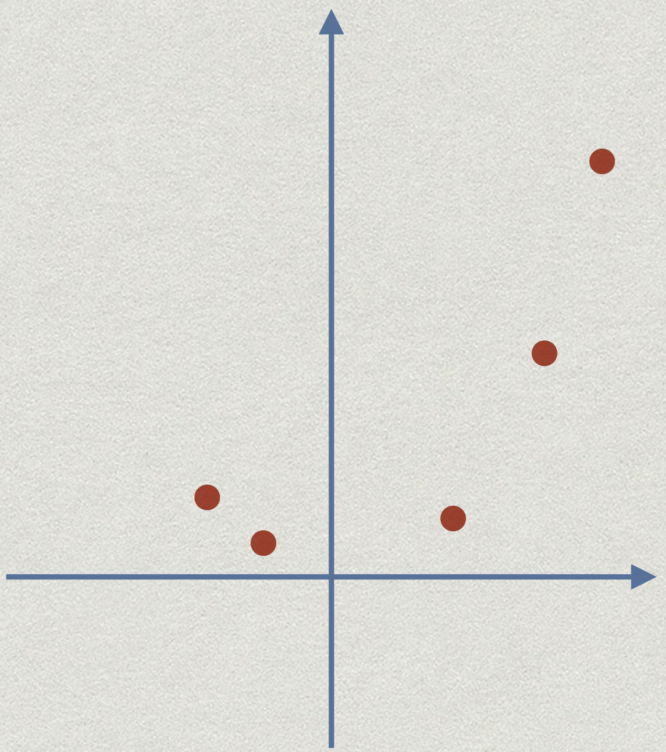
Rule based systems

Machine learning

Representation learning

Deep learning

Machine / Deep learning



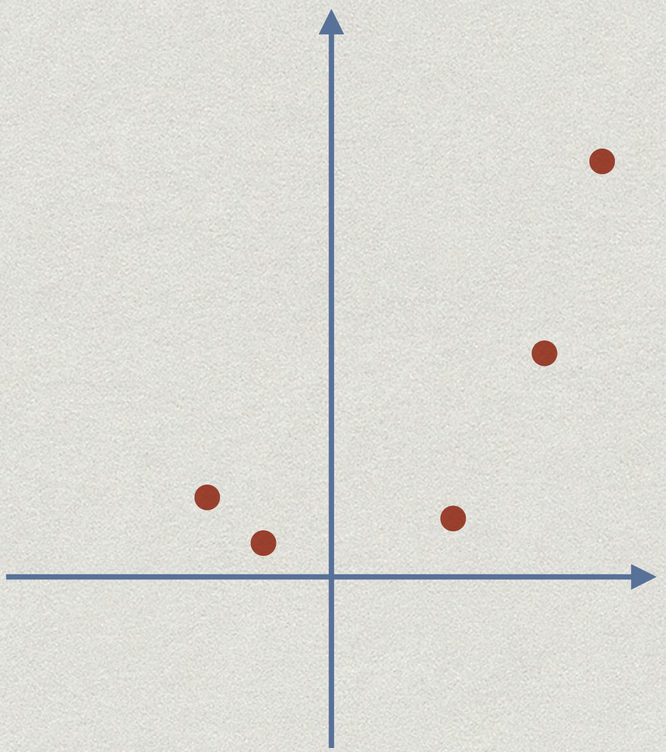
Rule based systems $y = 3x^2$

Machine learning

Representation learning

Deep learning

Machine / Deep learning



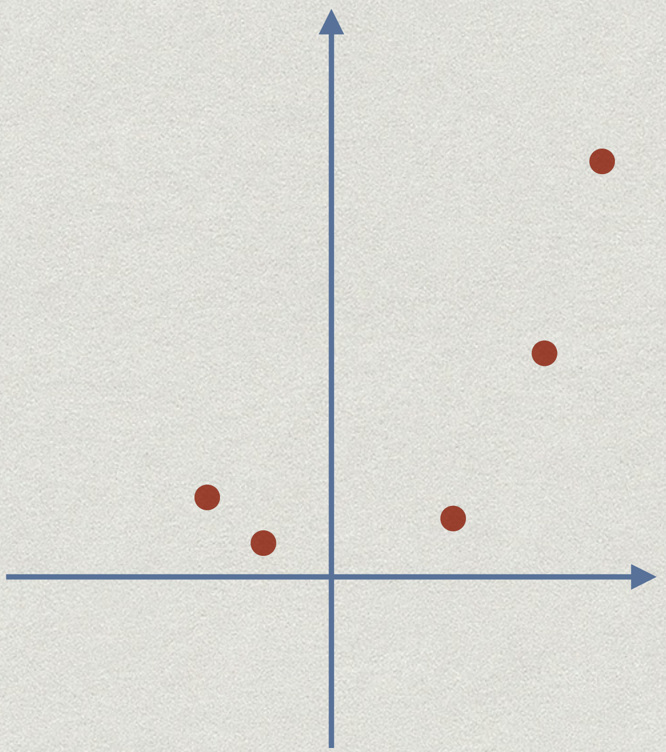
Rule based systems $y = 3x^2$

Machine learning $y = ax^b + c$

Representation learning

Deep learning

Machine / Deep learning



Rule based systems $y = 3x^2$

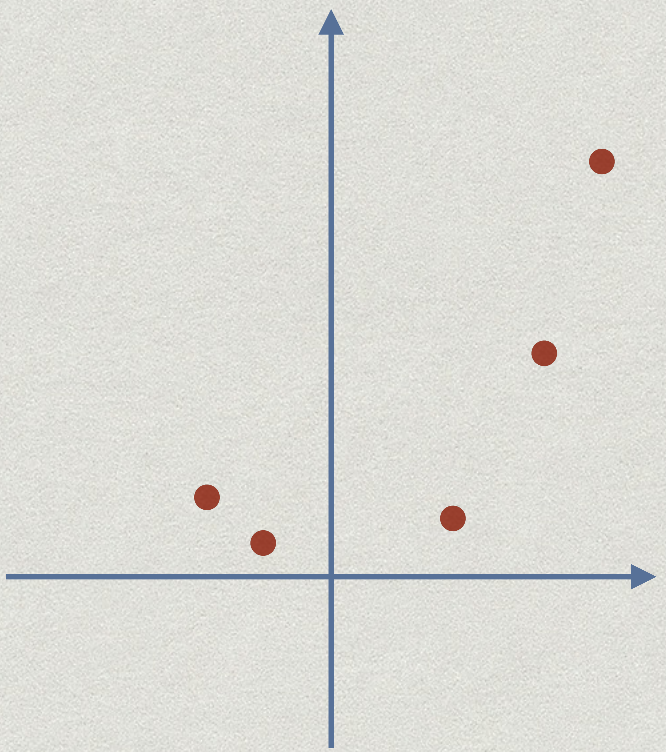
Machine learning $y = ax^b + c$

Representation learning

Deep learning

$$y = \sum_{i=0}^{N} a_i x^i$$

Machine / Deep learning



Rule based systems $y = 3x^2$

Machine learning $y = ax^b + c$

Representation learning

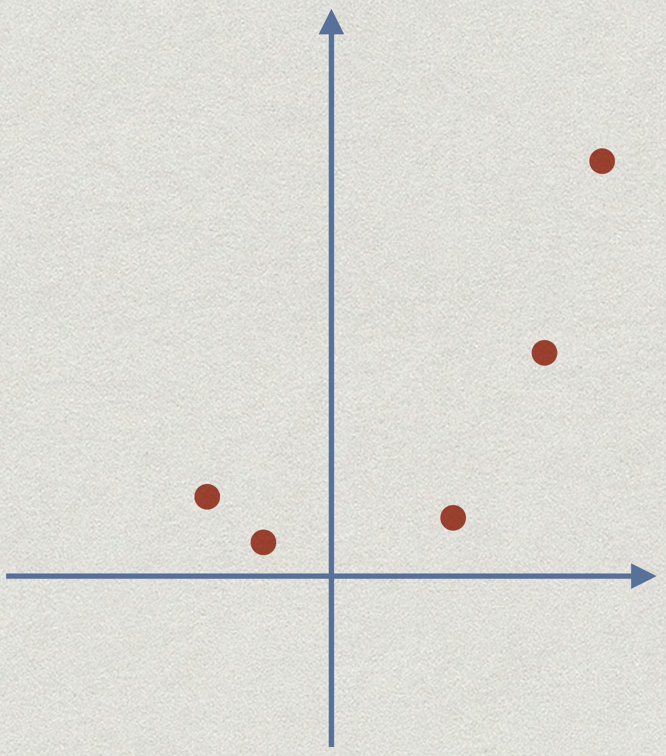
Deep learning (N very big)

$$y = \sum_{i=0}^{N} a_i x^i$$

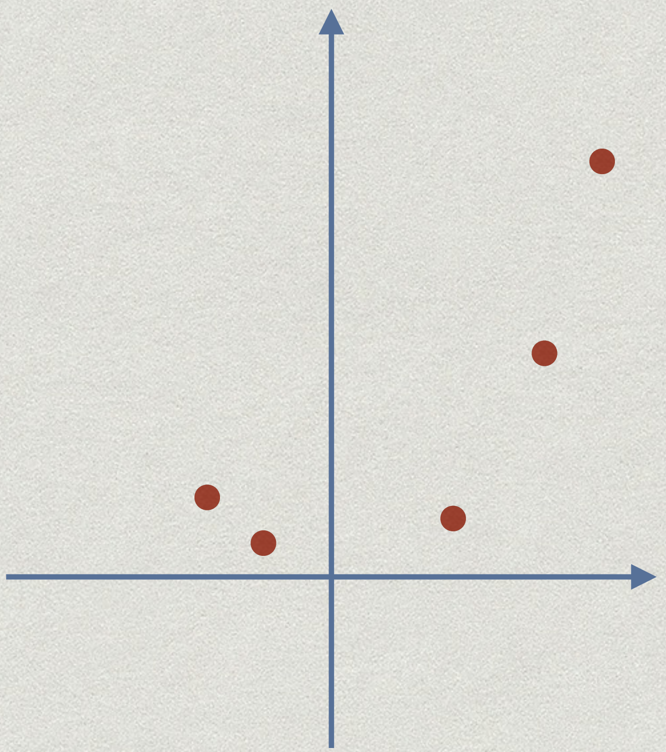
In **Deep learning** you learn **everything**.

This is **very powerful**, but can be **very inefficient**.

Objective: generalization

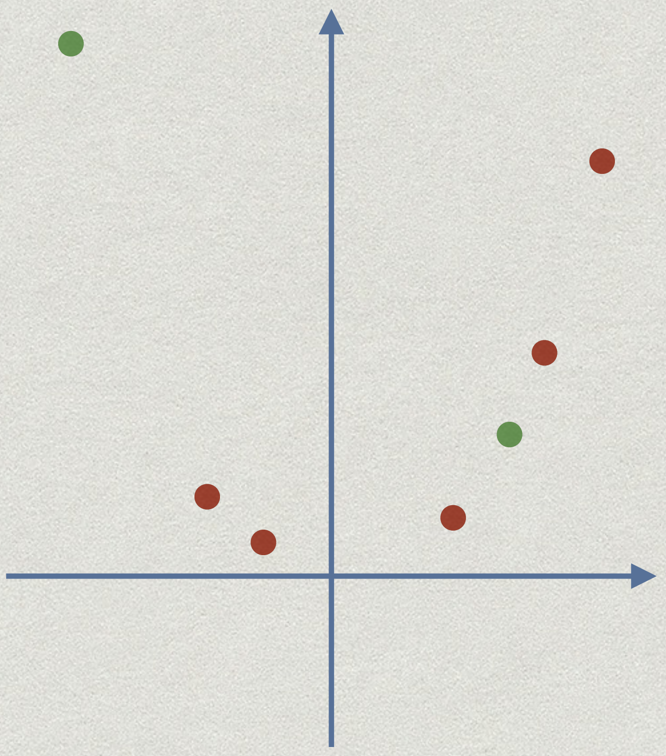


Objective: generalization



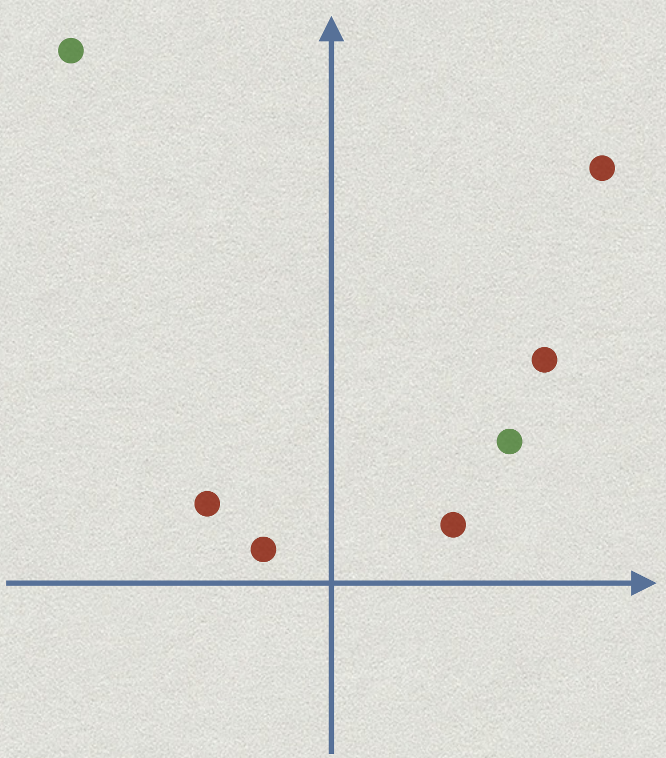
- Machine learning strategy:
  - observe the **training** data to learn the **features**
  - **not** learn the **noise**

Objective: generalization

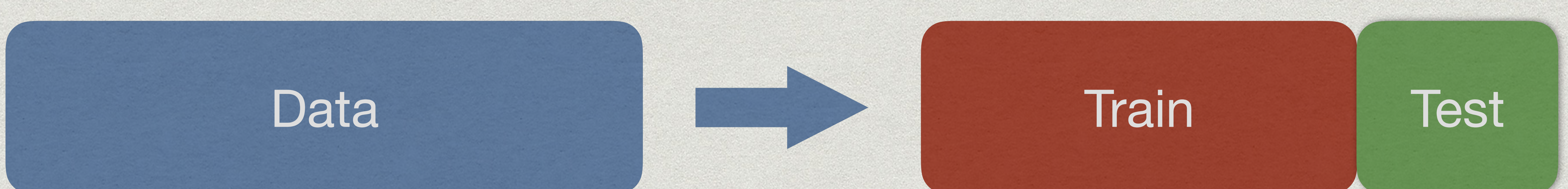


- Machine learning strategy:
  - observe the training data to learn the **features**
  - <u>**not**</u> learn the **noise**
  - generalize well to the test data (good prediction)

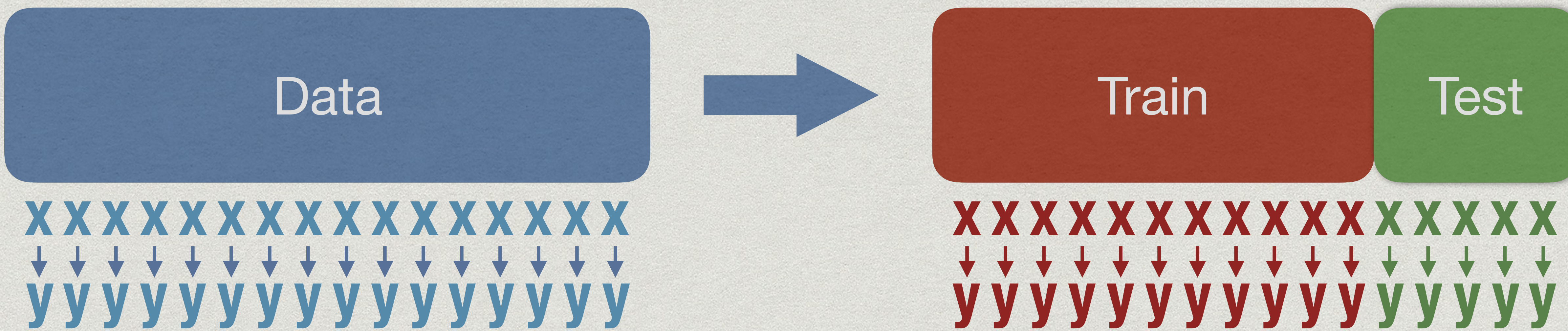
Objective: generalization



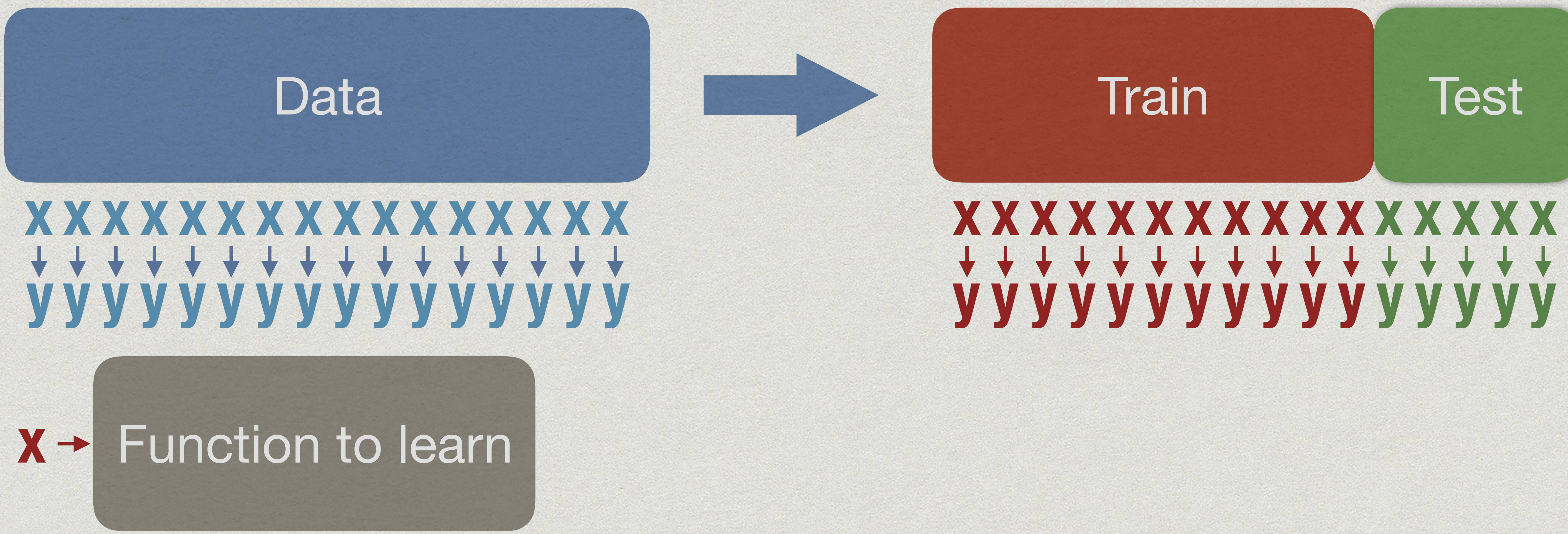
- Machine learning strategy:
  - observe the training data to learn the **features**
  - <u>**not**</u> learn the **noise**
  - generalize well to the test data (good prediction)



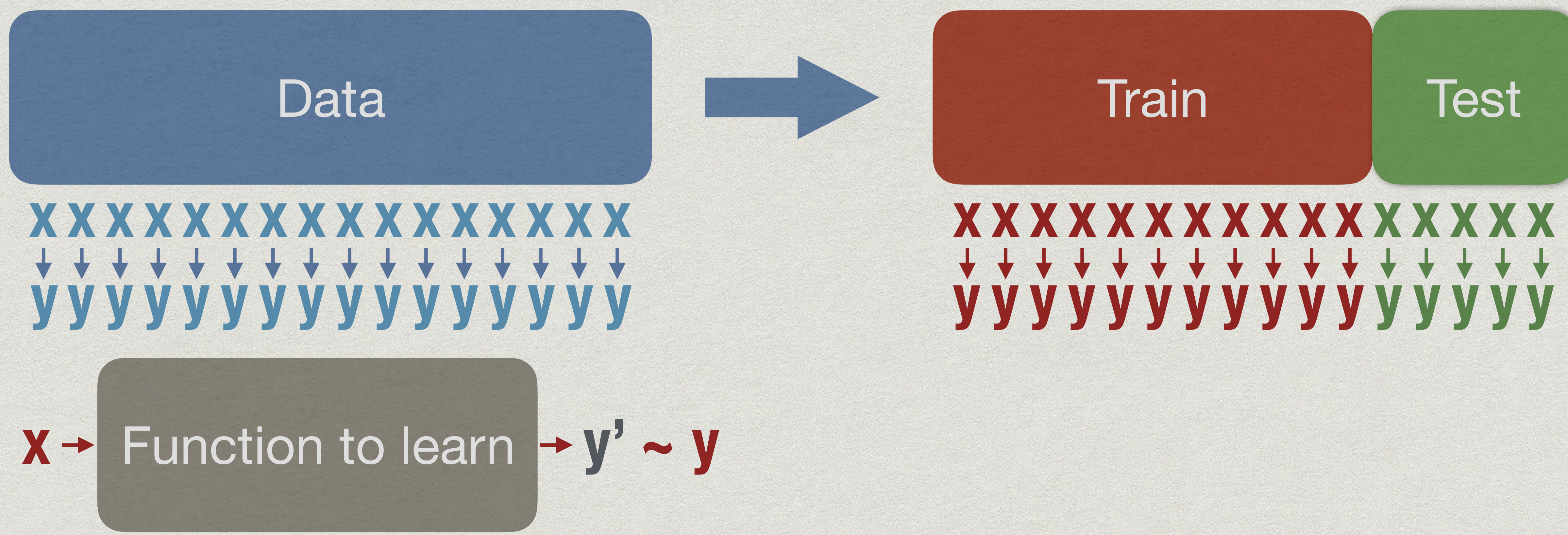
Machine learning



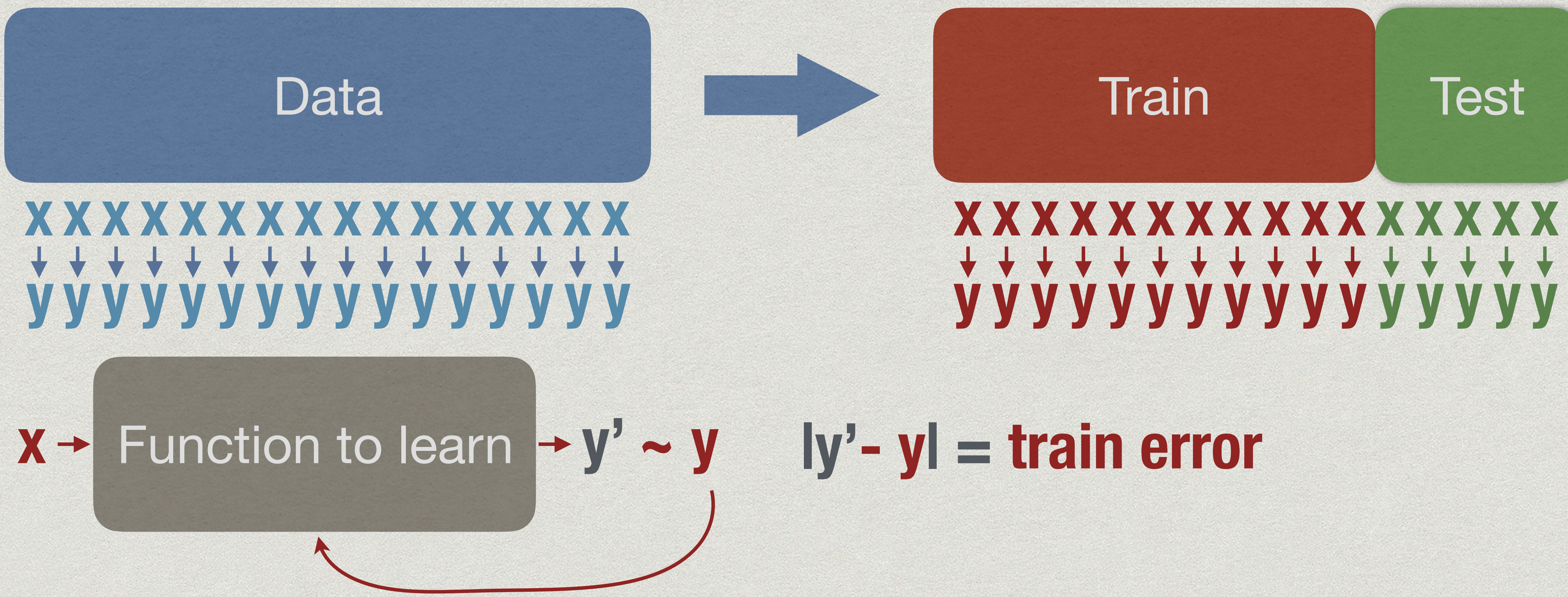
Machine learning



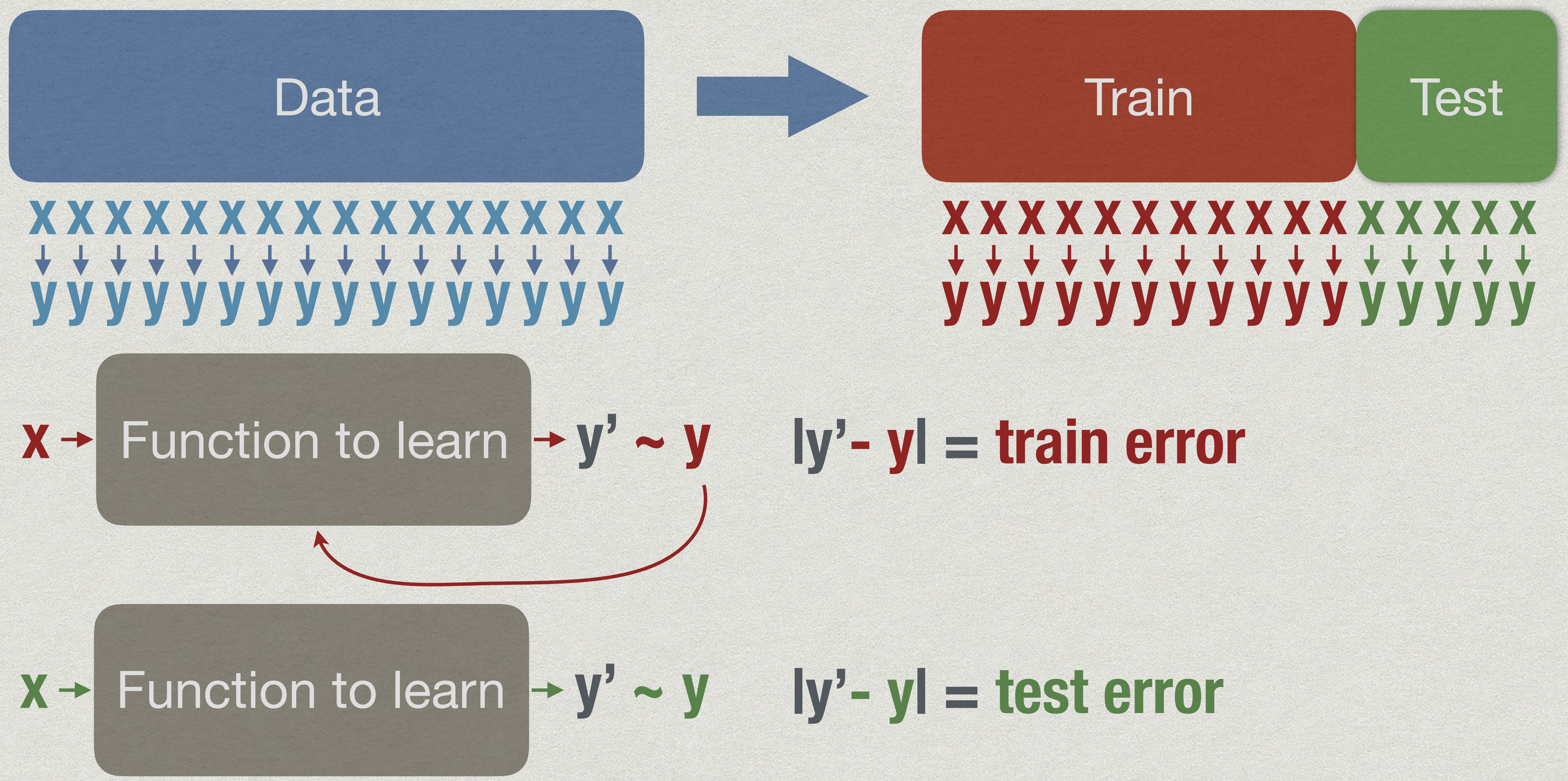
Machine learning



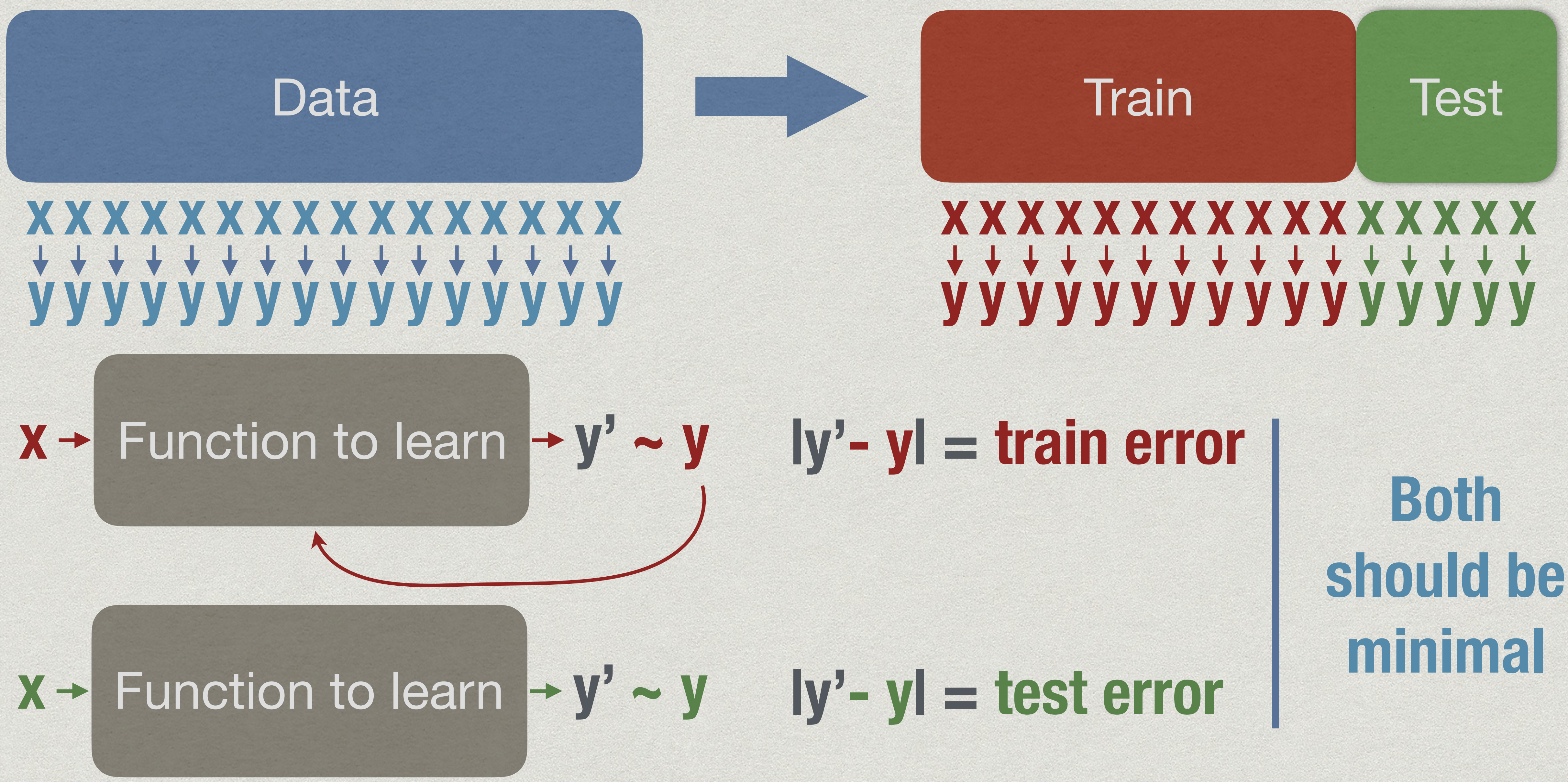
Machine learning



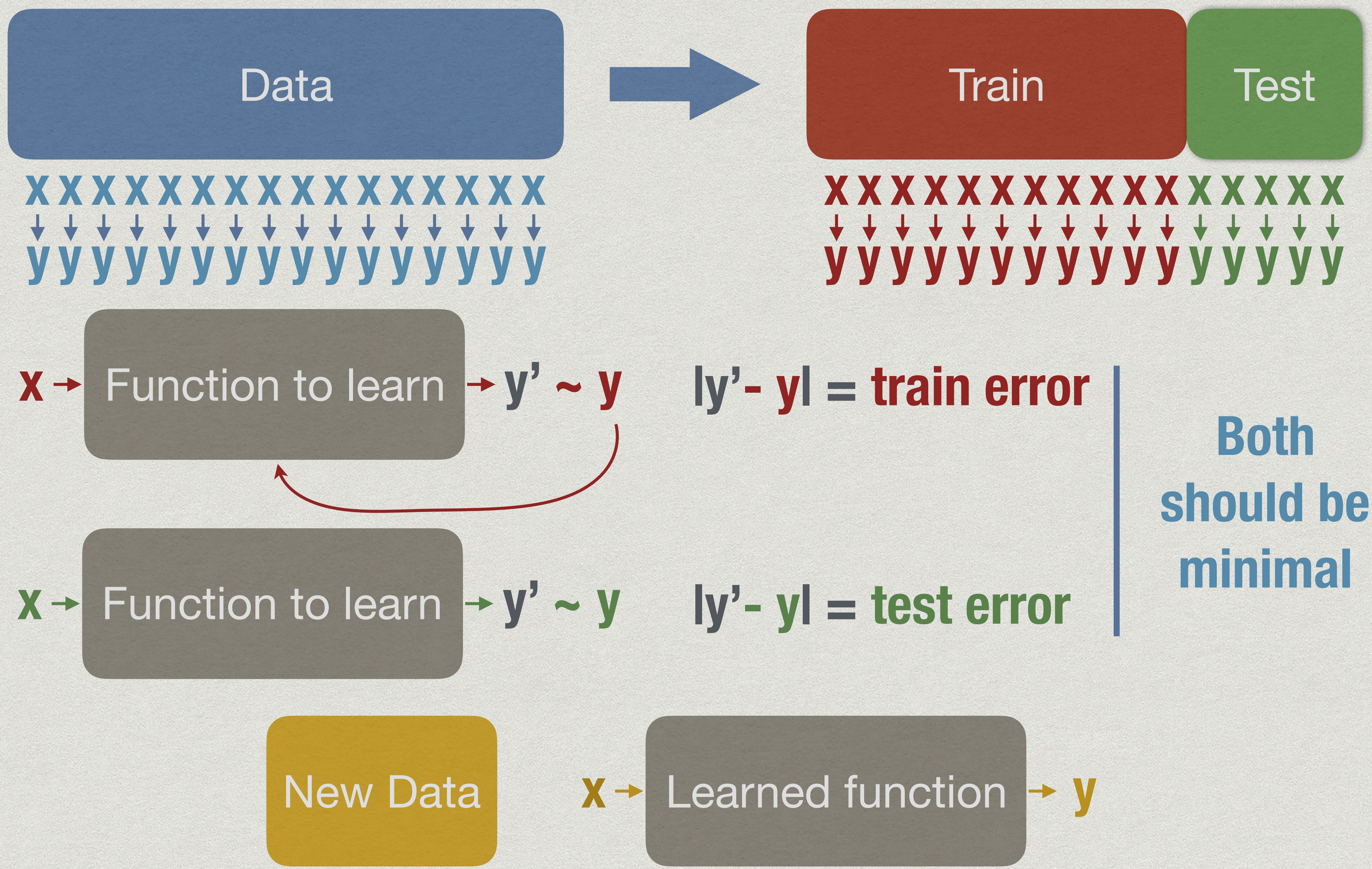
Machine learning



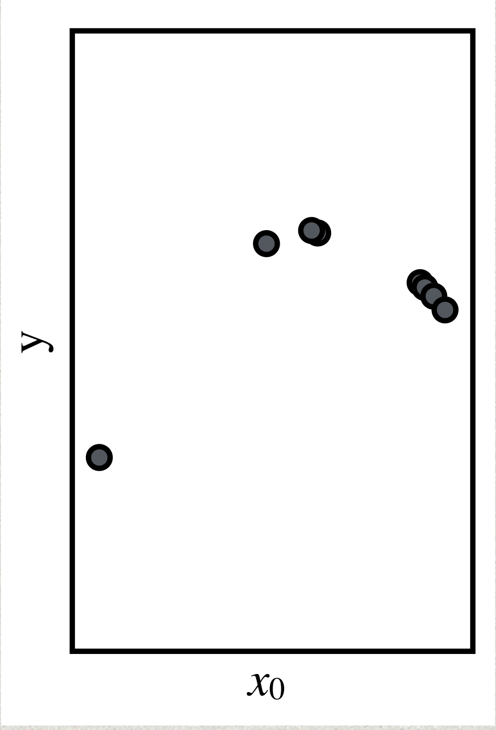
Machine learning



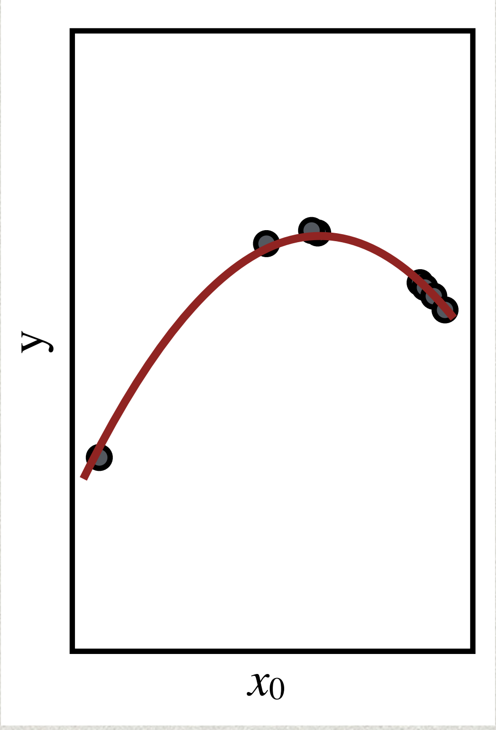
Machine learning



Under / Overfitting



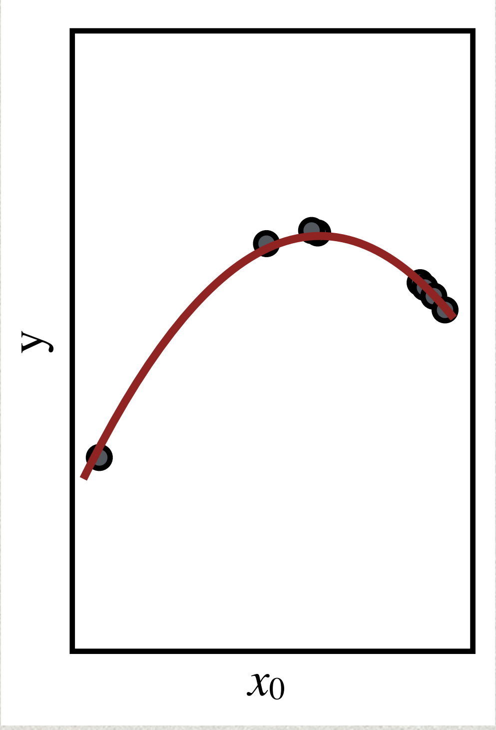
Under / Overfitting



Human fitting :

« Hey, this looks like a 2nd order polynomial »

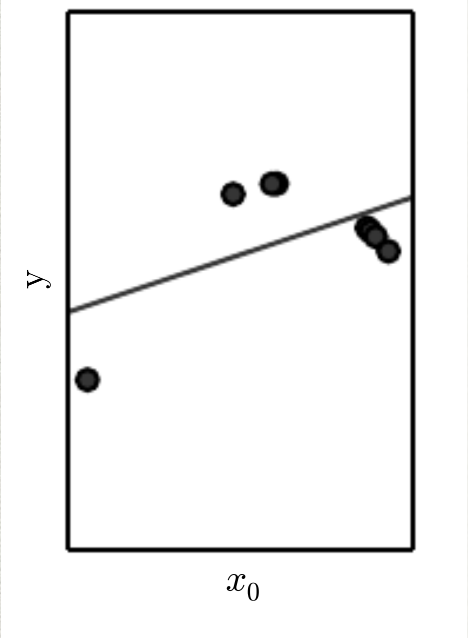
Under / Overfitting



Human fitting :

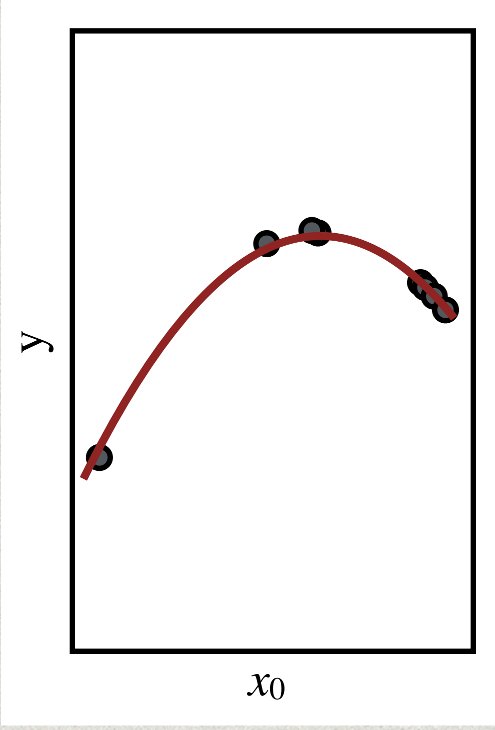
« Hey, this looks like a 2nd order polynomial »

Learned fitting :



N = 1

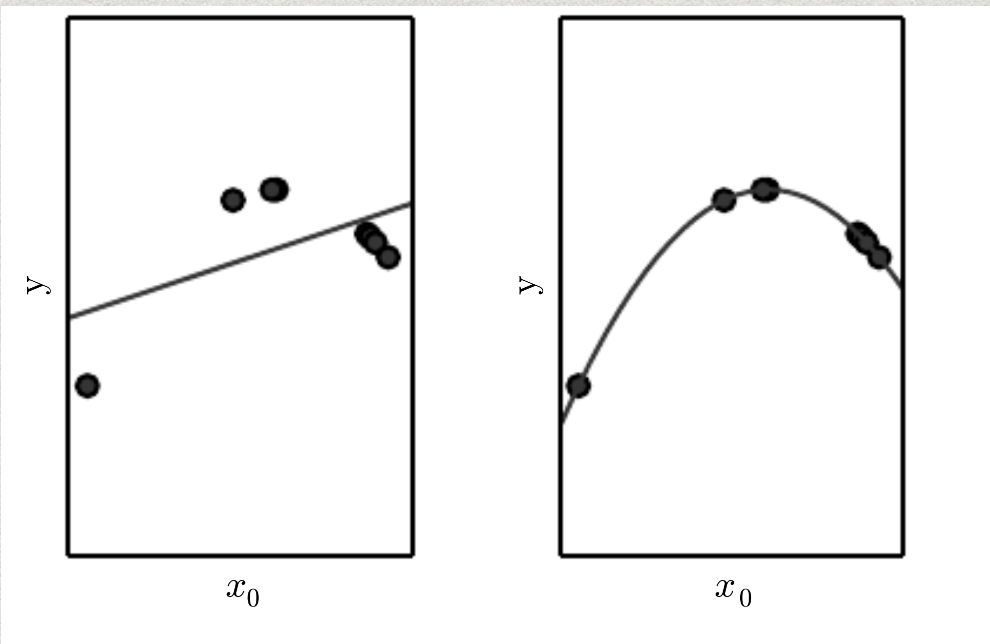
Under / Overfitting



Human fitting :

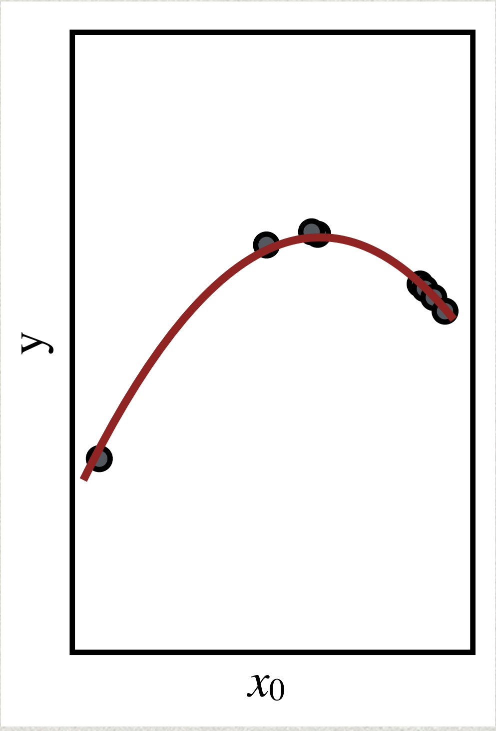
« Hey, this looks like a 2^{nd} order polynomial »

Learned fitting :



N = 1 **N = 2**

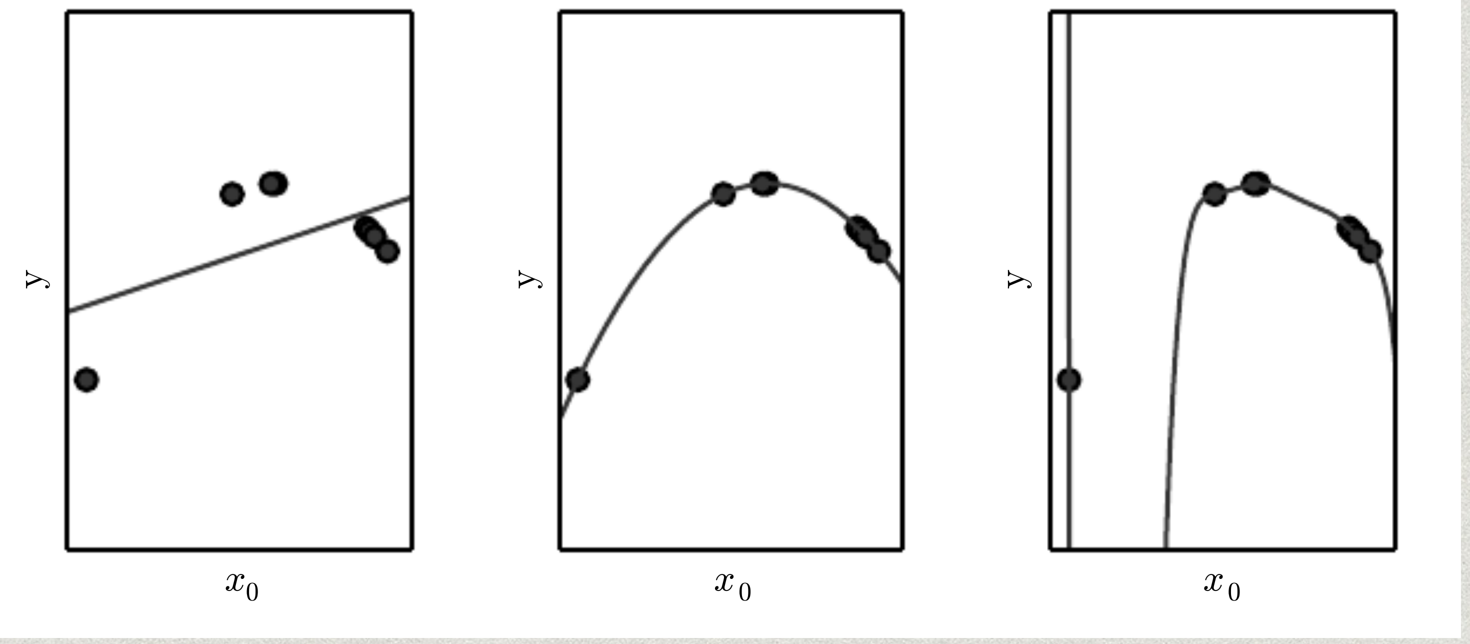
Under / Overfitting



Human fitting :

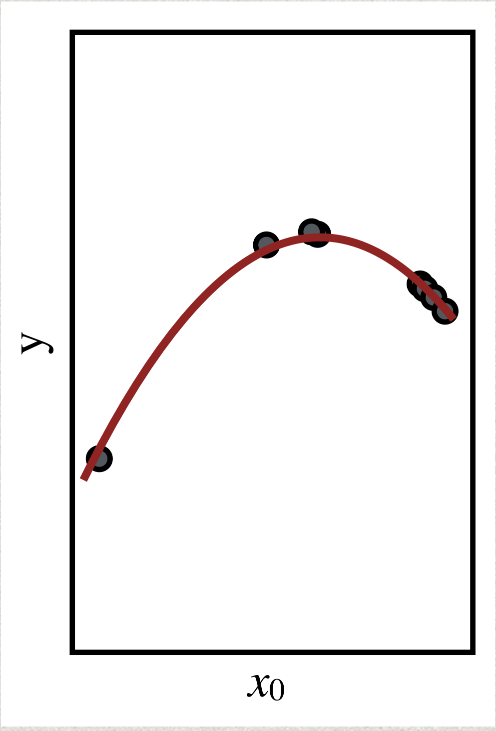
« Hey, this looks like a 2nd order polynomial »

Learned fitting :



N = 1 **N = 2** **N = N_{pts}**

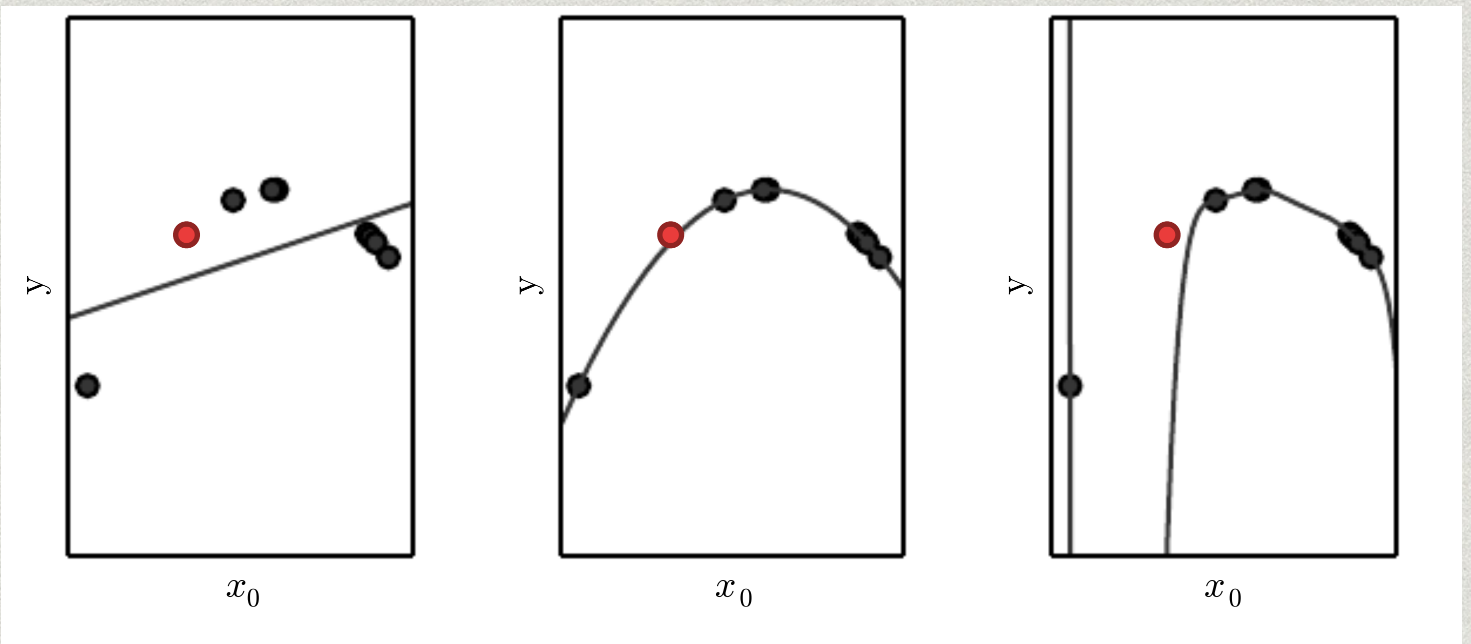
Under / Overfitting



Human fitting :

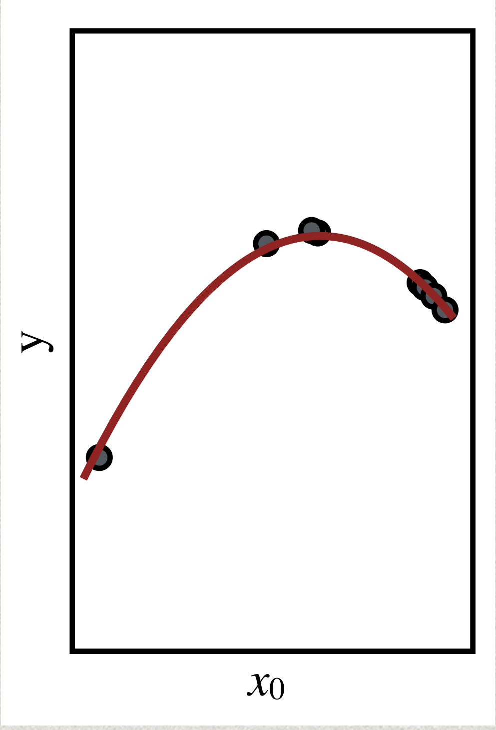
« Hey, this looks like a 2nd order polynomial »

Learned fitting :



N = 1 **N = 2** **N = N_{pts}**

Under / Overfitting

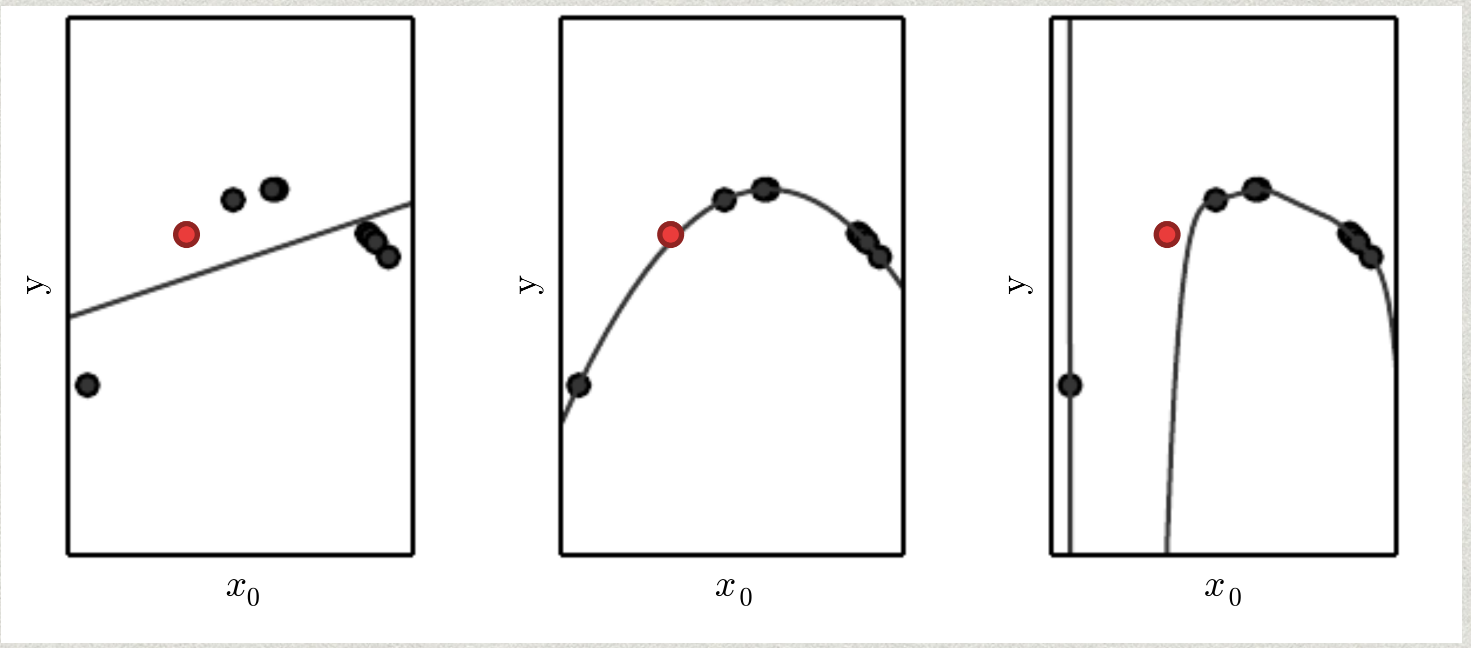


Human fitting :

« Hey, this looks like a 2nd order polynomial »

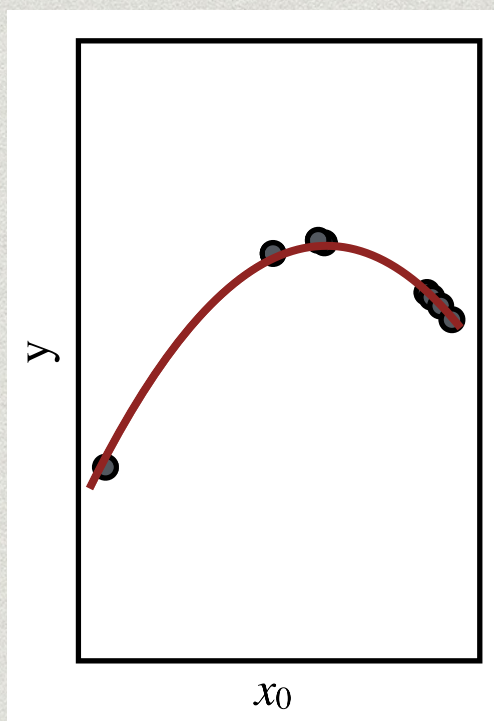
Learned fitting :

Underfitting



N = 1 **N = 2** **N = N_{pts}**

Under / Overfitting

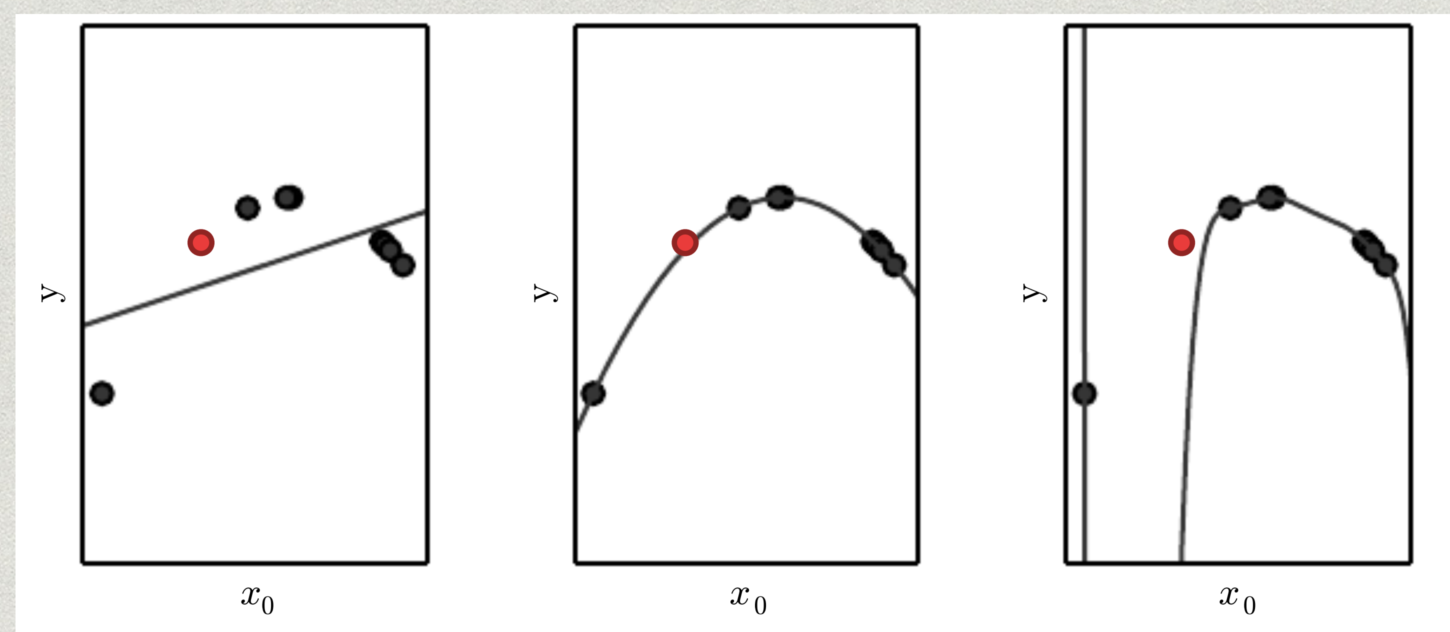


Human fitting :

« Hey, this looks like a 2^{nd} order polynomial »

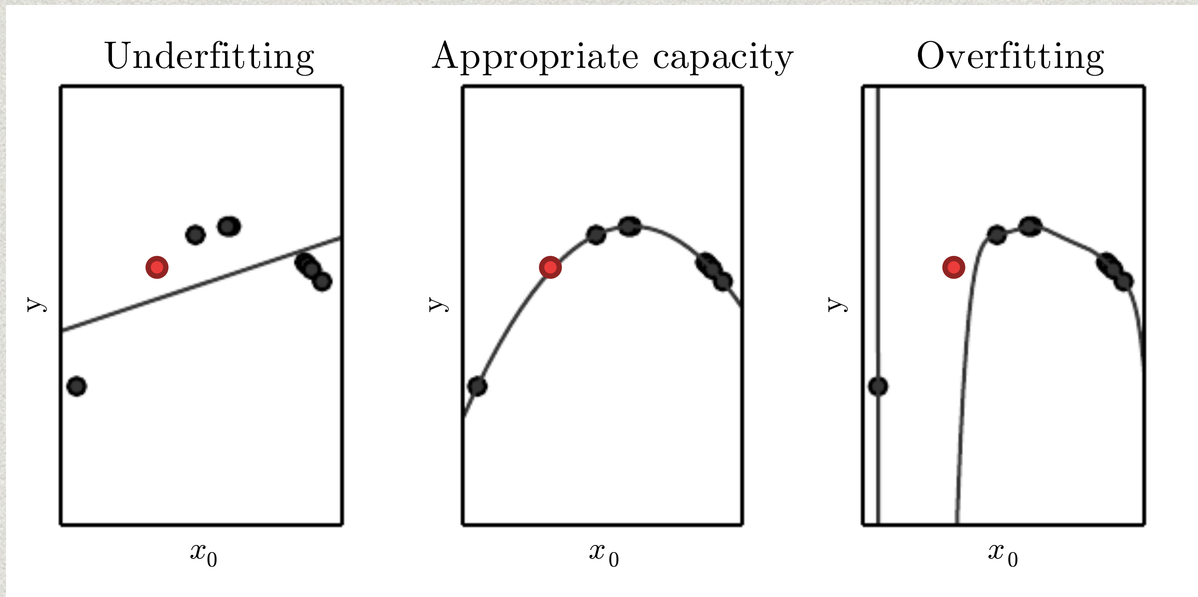
Learned fitting :

Underfitting **Overfitting**



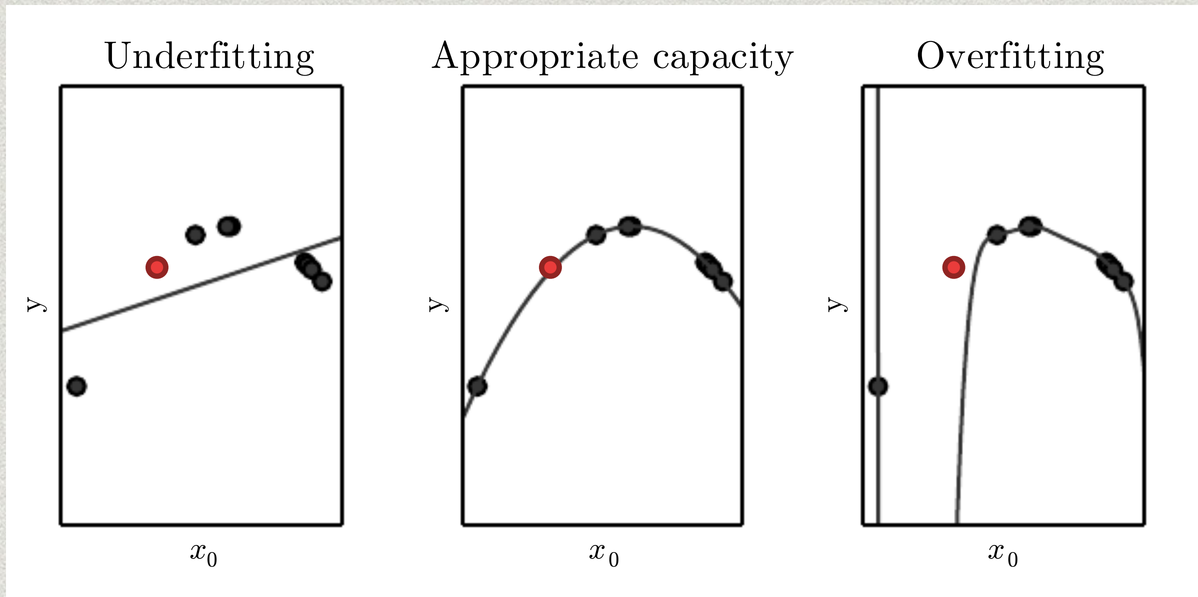
N = 1 **N = 2** **N = N_{pts}**

Under / Overfitting



The order N is called the *capacity*

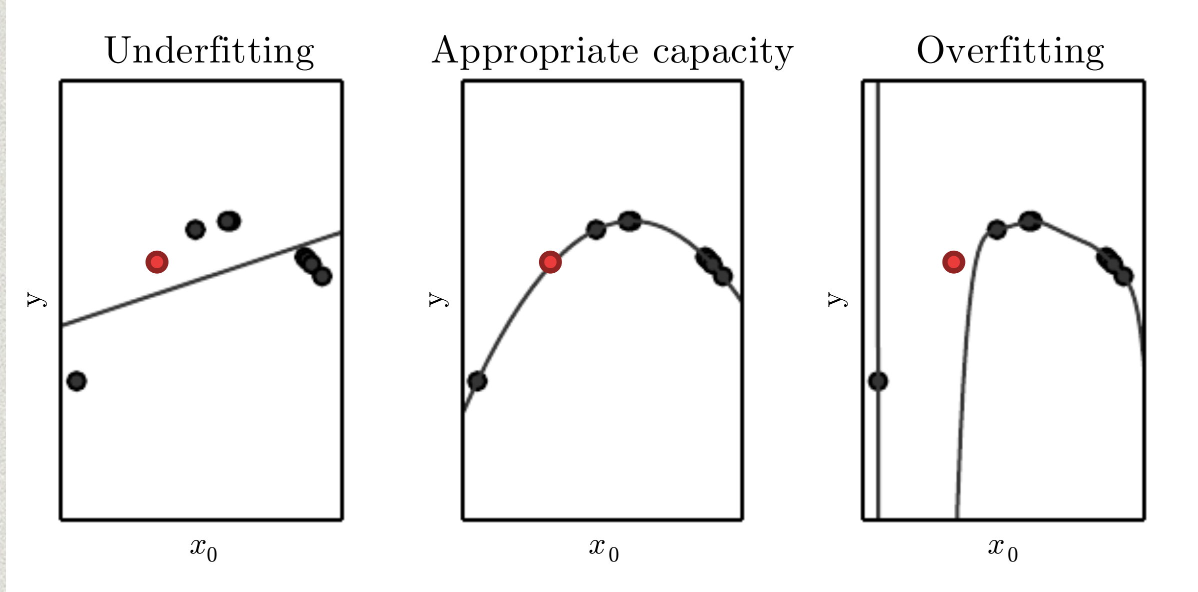
Under / Overfitting



Optimal

The order N is called the *capacity*

Under / Overfitting

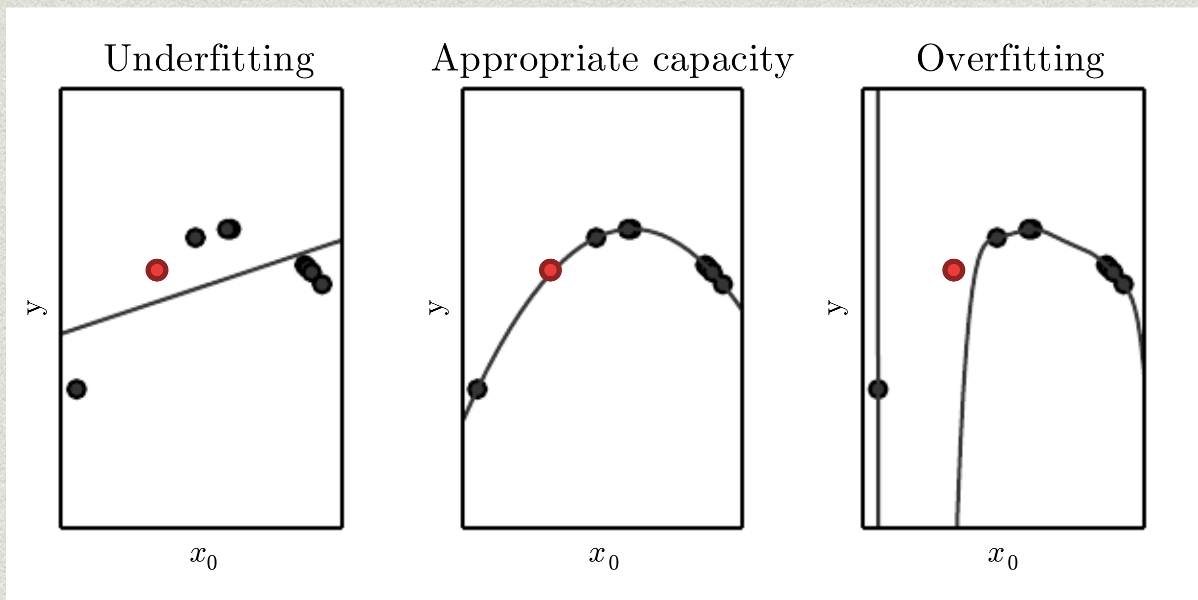


Training error

Optimal

The order N is called the *capacity*

Under / Overfitting



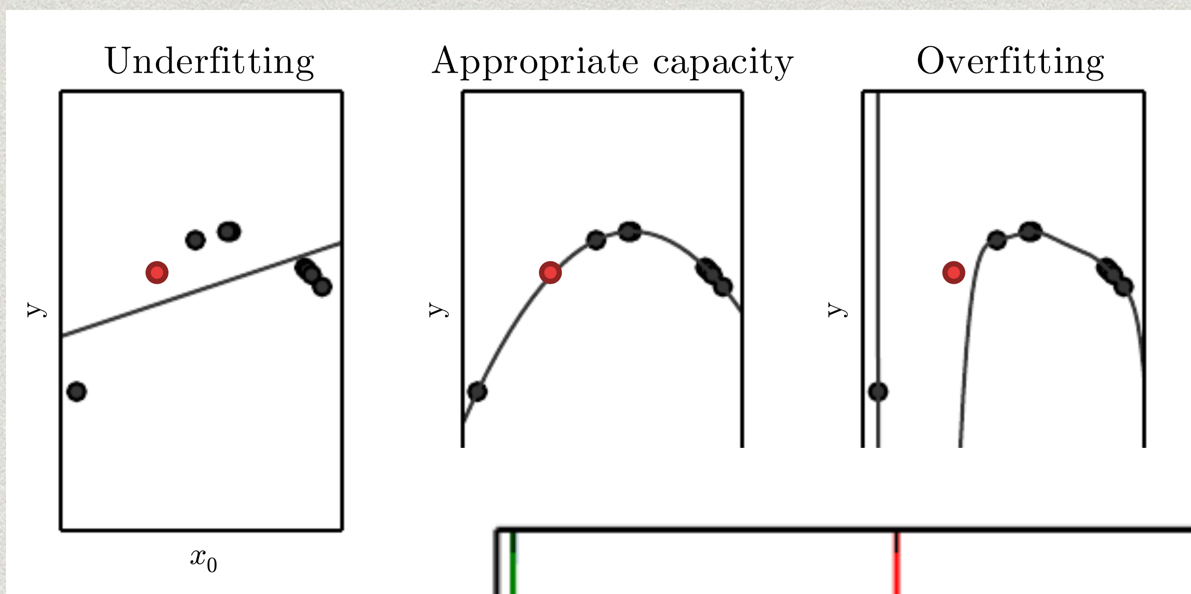
Training error

Optimal

Generalization (test) error

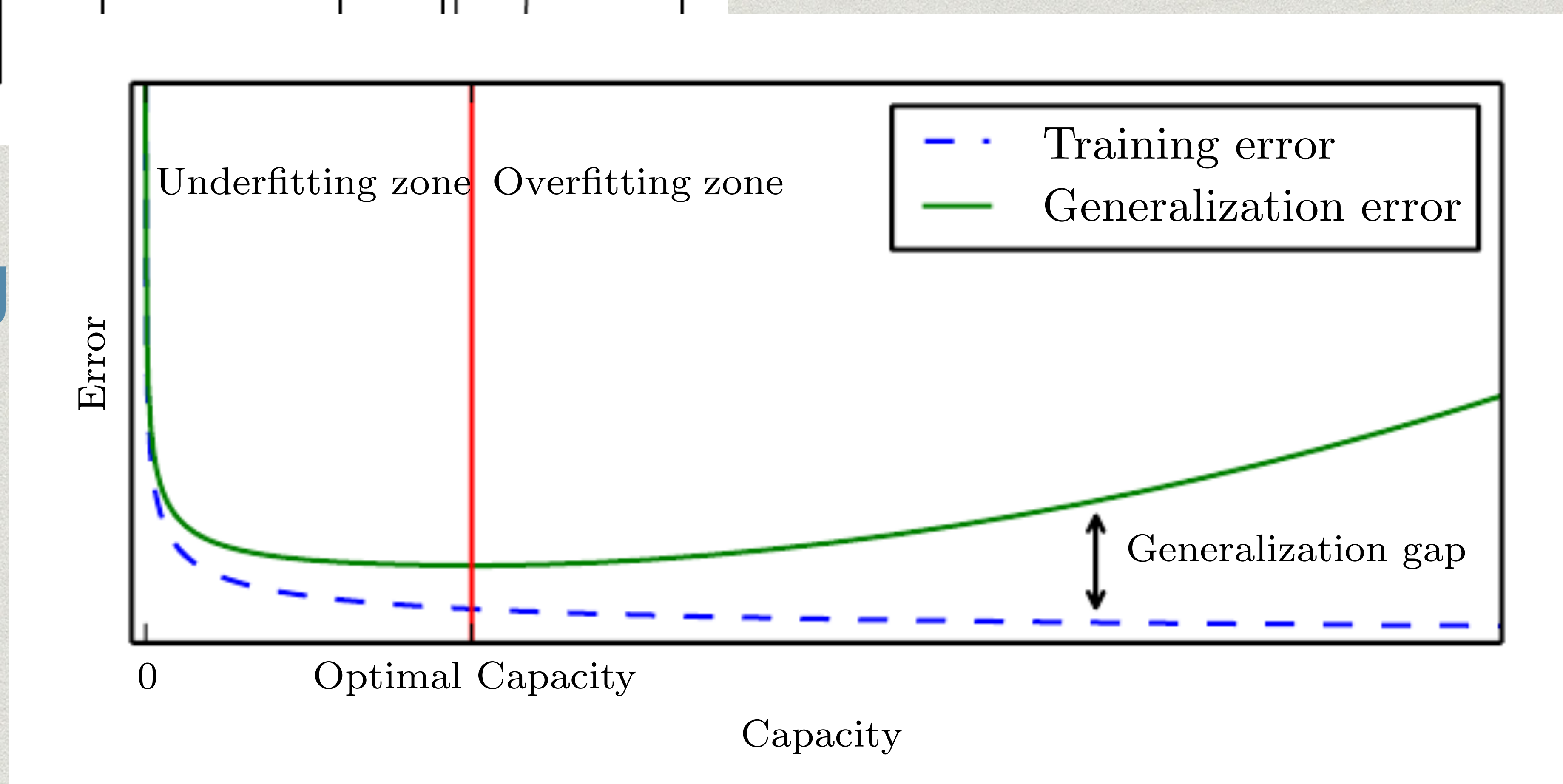
The order N is called the *capacity*

Under / Overfitting

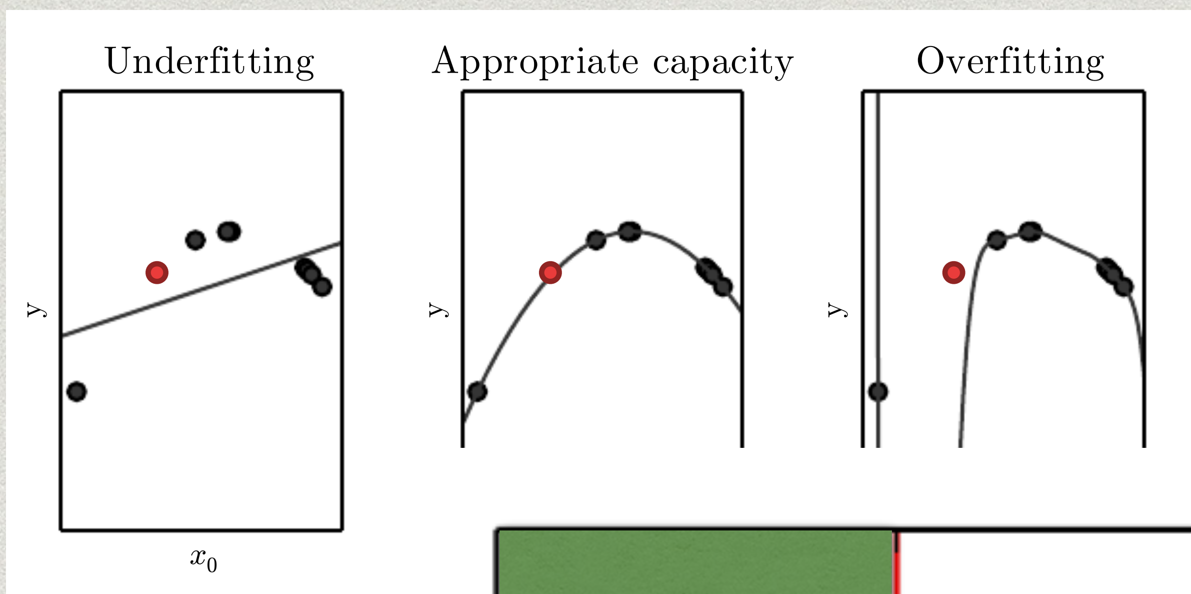


Training error

The order N is called the *capacity*

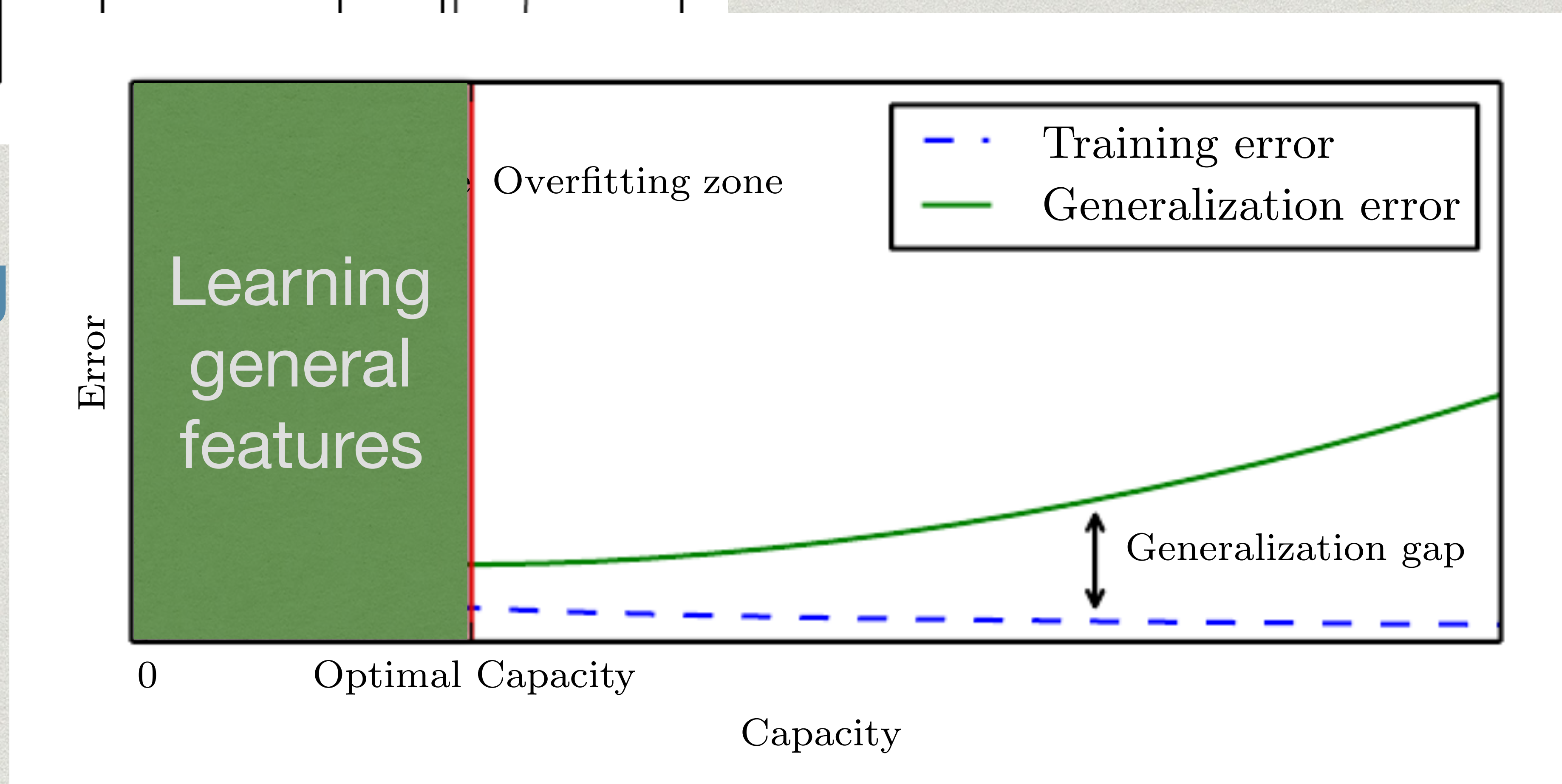


Under / Overfitting

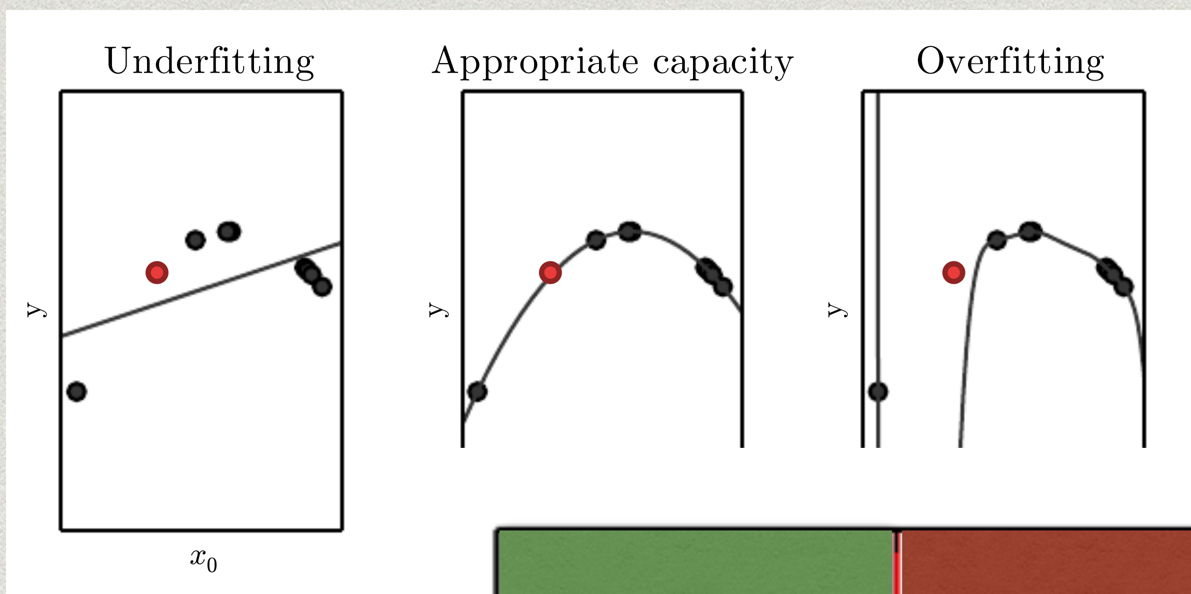


Training error

The order N is called the ***capacity***

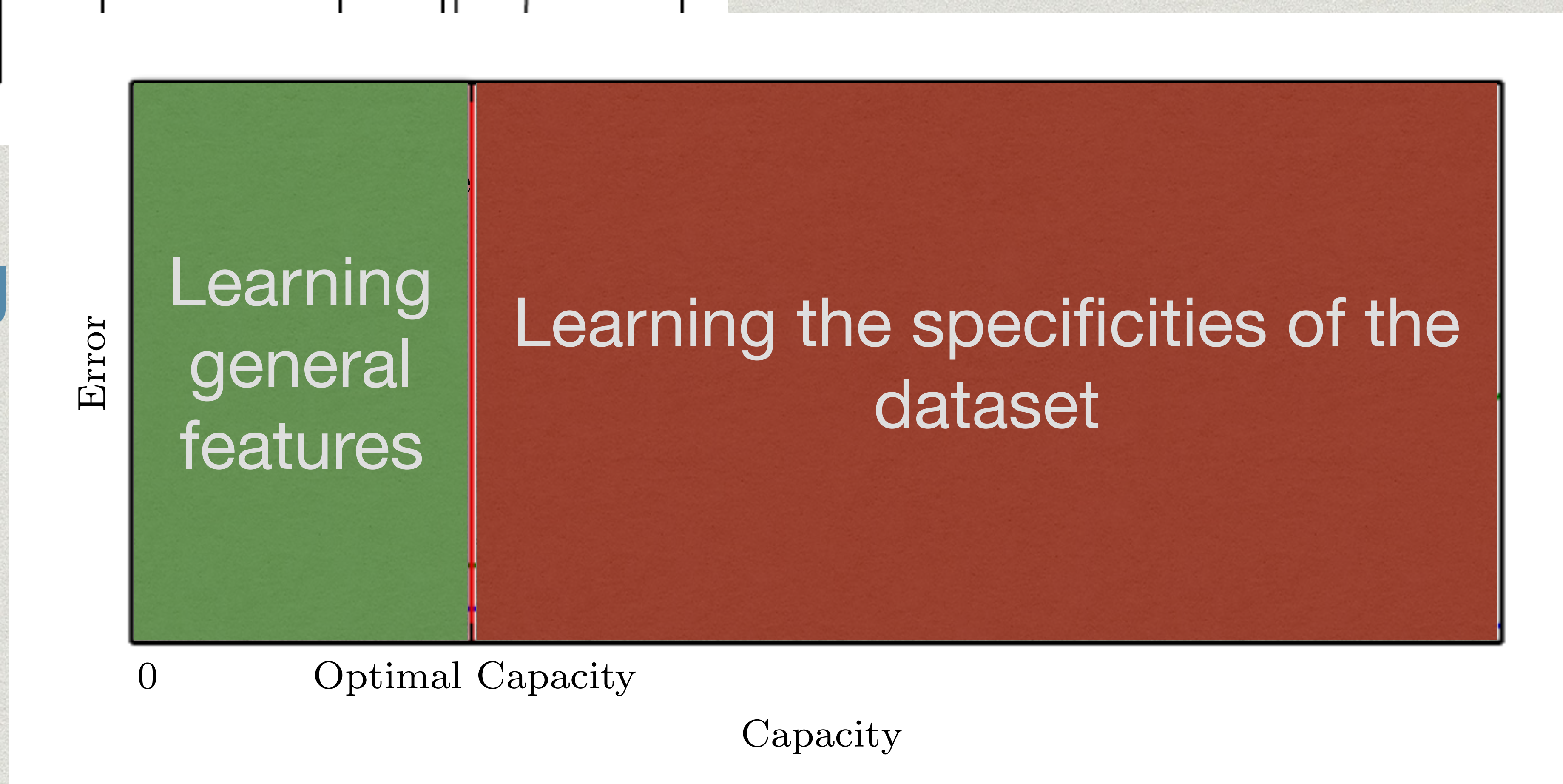


Under / Overfitting



Training error

The order N is called the *capacity*



Deep Learning is…

- a form of Machine Learning
- It specializes in data where the fit function is **very hard to express** (like image processing)

Traditional approaches	Deep Learning
Manual pre-selection of data to concentrate on important features	Input the « raw » data, to include maximum features

Striking a balance

- The full game of Deep Learning is:
  - to be able to express complex functions to represent the features in the raw data
  - achieve good generalization to new data: avoid overfitting

Machine learning $y = ax^b + c$

Deep learning $y = \sum_{i=0}^{N} a_i x^i$

Striking a balance

- The full game of Deep Learning is:
  - to be able to express complex functions to represent the features in the raw data
  - achieve good generalization to new data: avoid overfitting

Machine learning $y = ax^b + c$

Deep learning $y = \sum_{i=0}^{N} a_i x^i$

Problem: you must guess the function

Striking a balance

- The full game of Deep Learning is:
  - to be able to express complex functions to represent the features in the raw data
  - achieve good generalization to new data: avoid overfitting

Machine learning $y = ax^b + c$

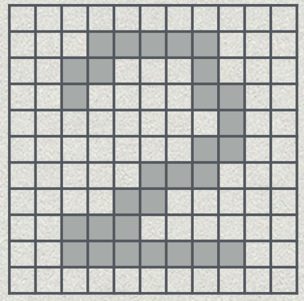
Problem: you must guess the function

Deep learning $y = \sum_{i=0}^{N} a_i x^i$

You don't need to know it. But dangerous: very prone to overfitting!

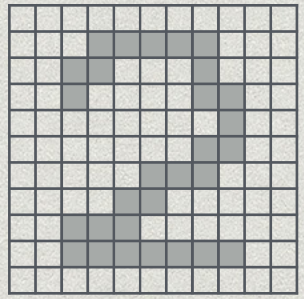
THE CURSE OF BIG DIMENSIONALITY

Raw data has high dimension



- 28 x 28 pixel image = 784 independent dimensions

Raw data has high dimension



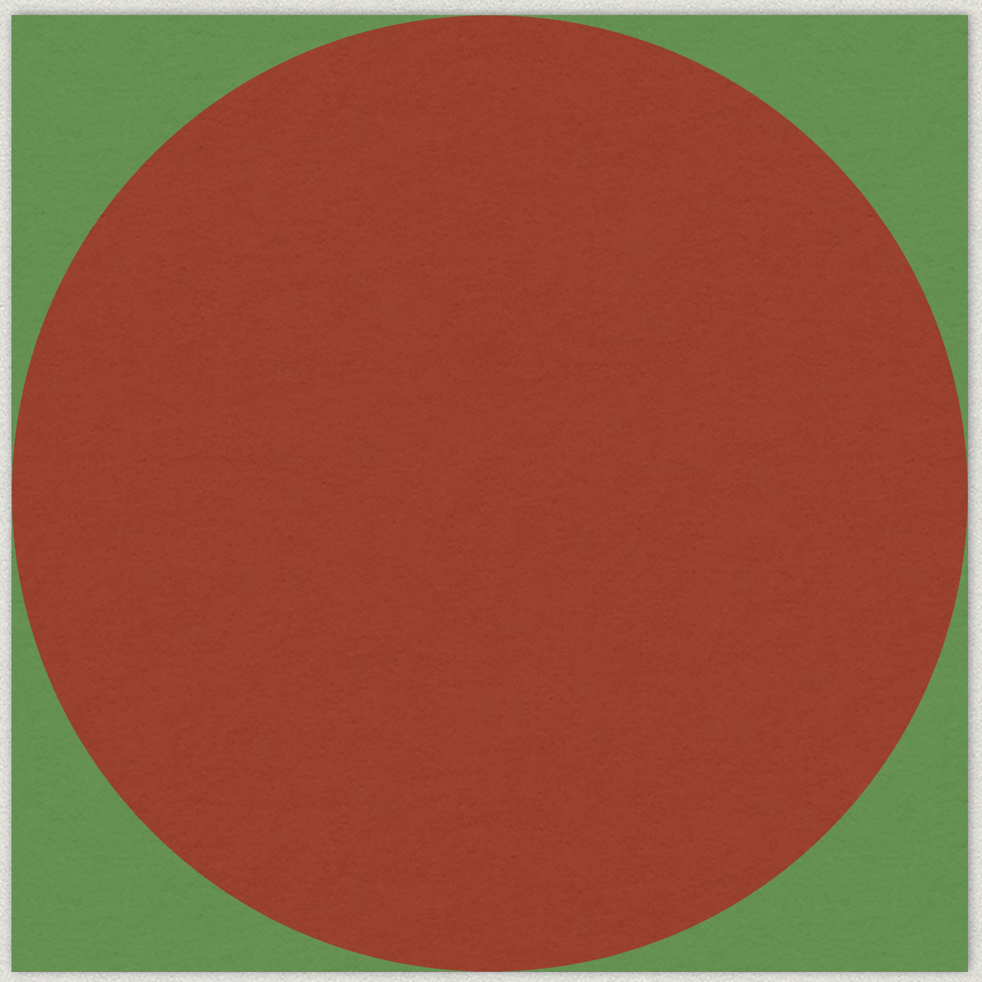
- 28 x 28 pixel image = 784 independent dimensions
- 256 x 256 pixel image with 3 color channels = 196,608 independent dimensions!!

High dimension sucks

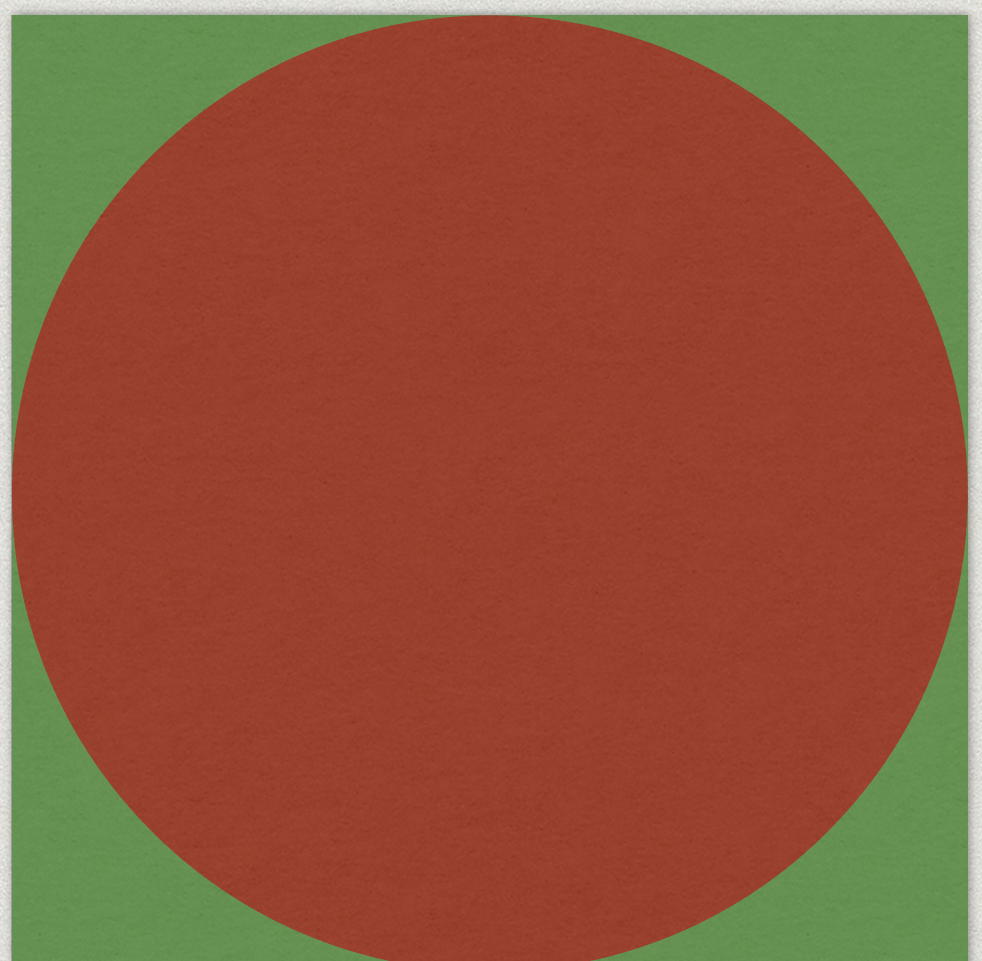
High dimension sucks

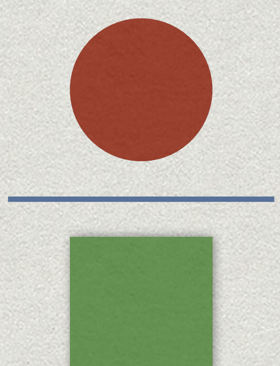
Very High dimension sucks **exponentially more**

Example: let’s talk spheres



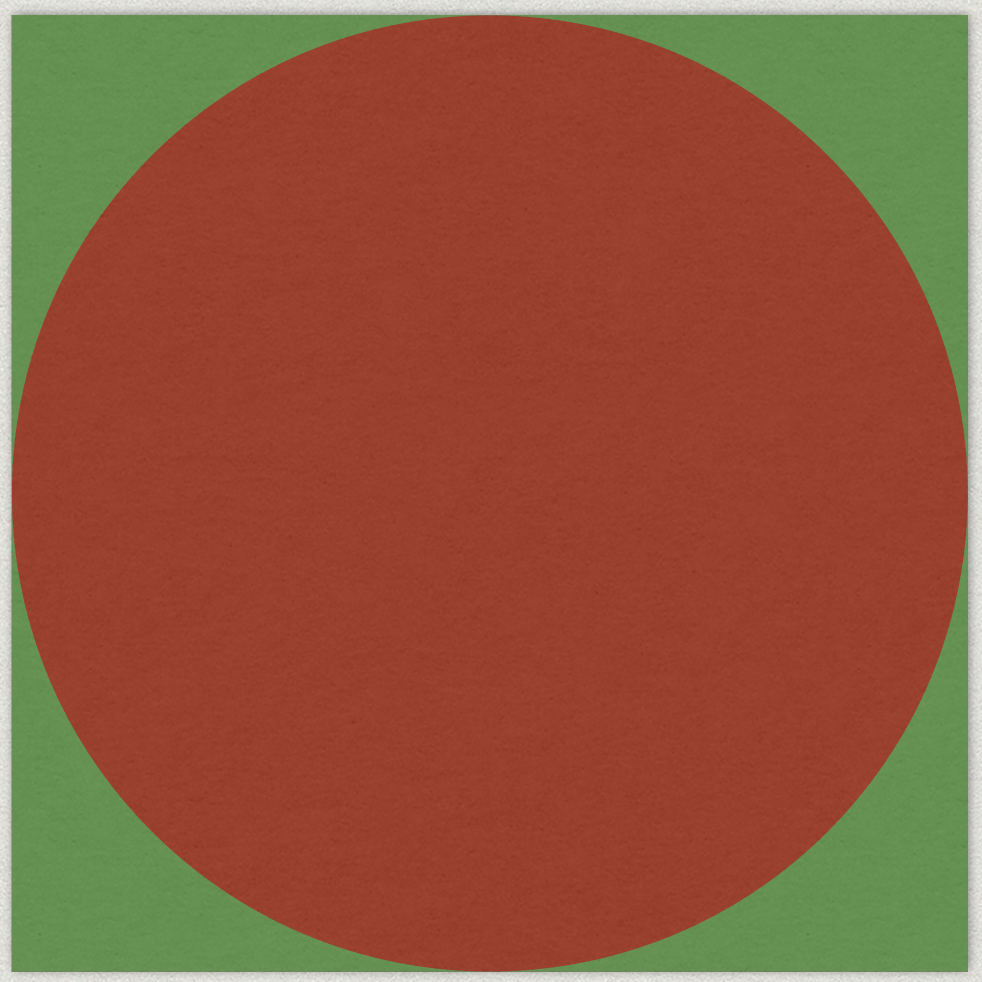
Example: let's talk spheres

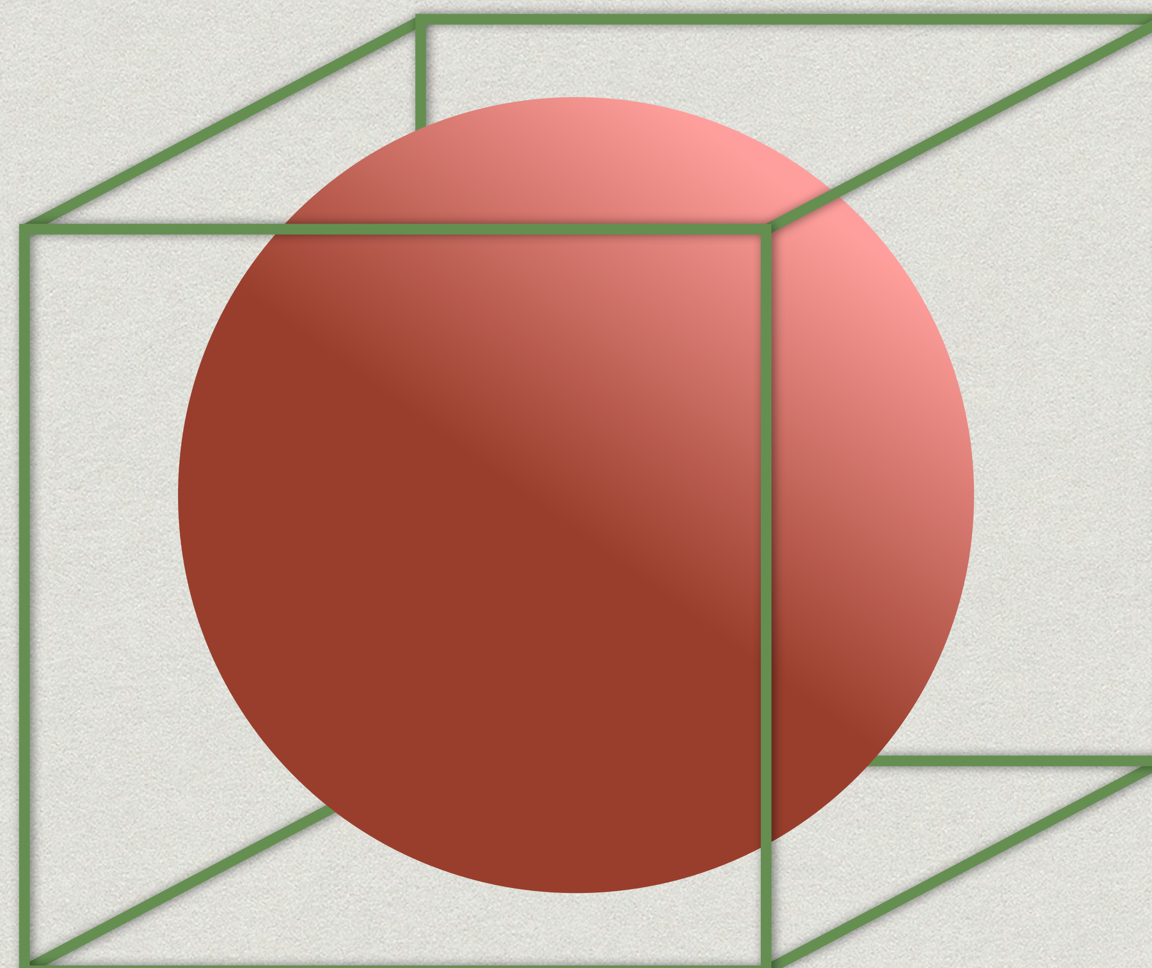


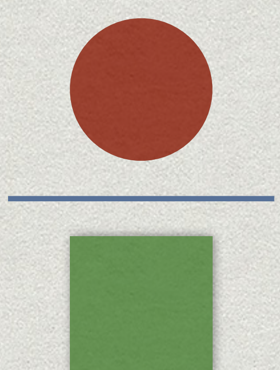


= 78.5 %

Example: let’s talk spheres

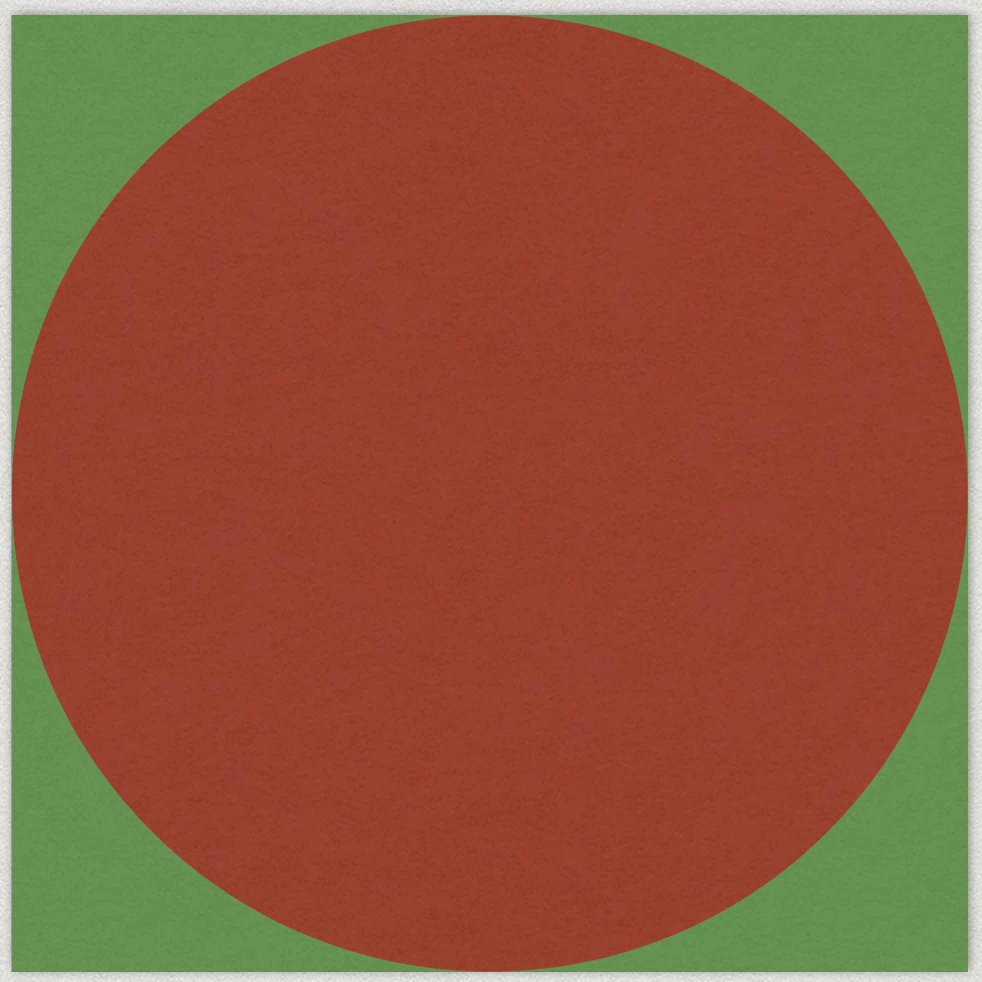


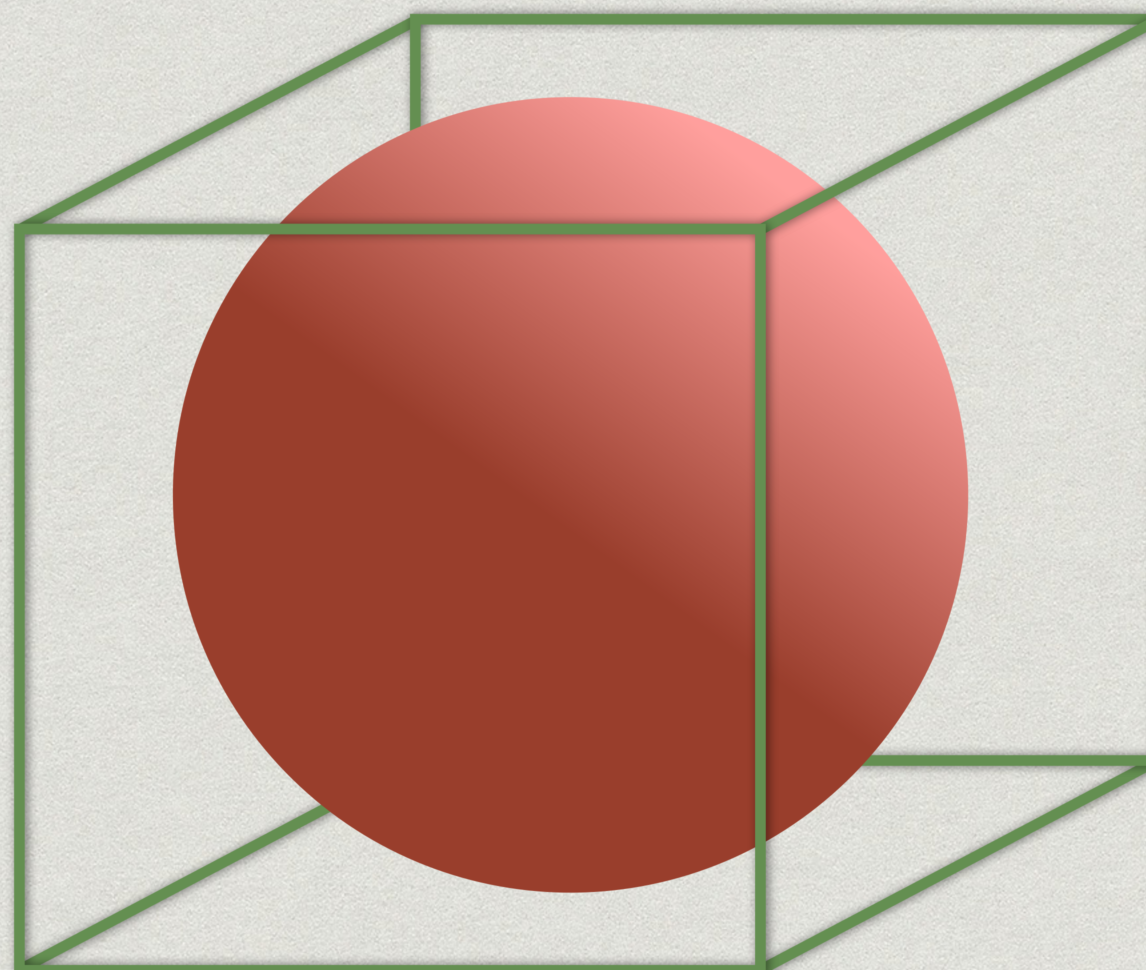


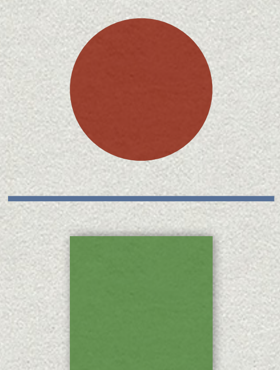


= 78.5 %

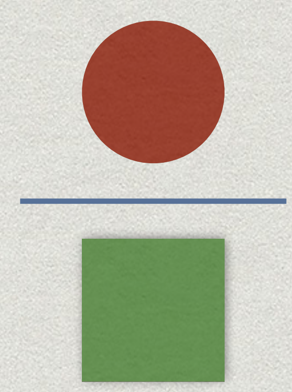
Example: let’s talk spheres







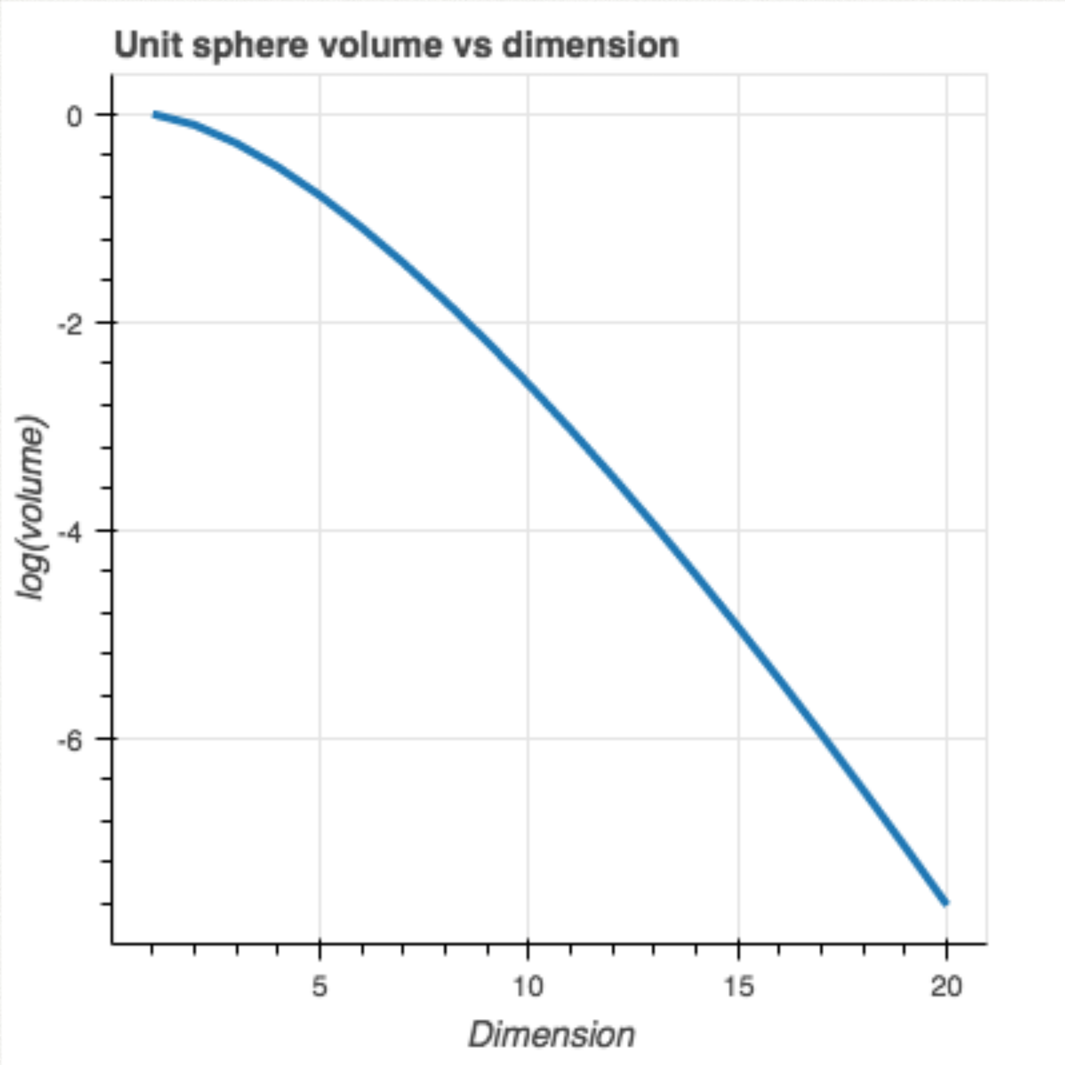
= 78.5 %



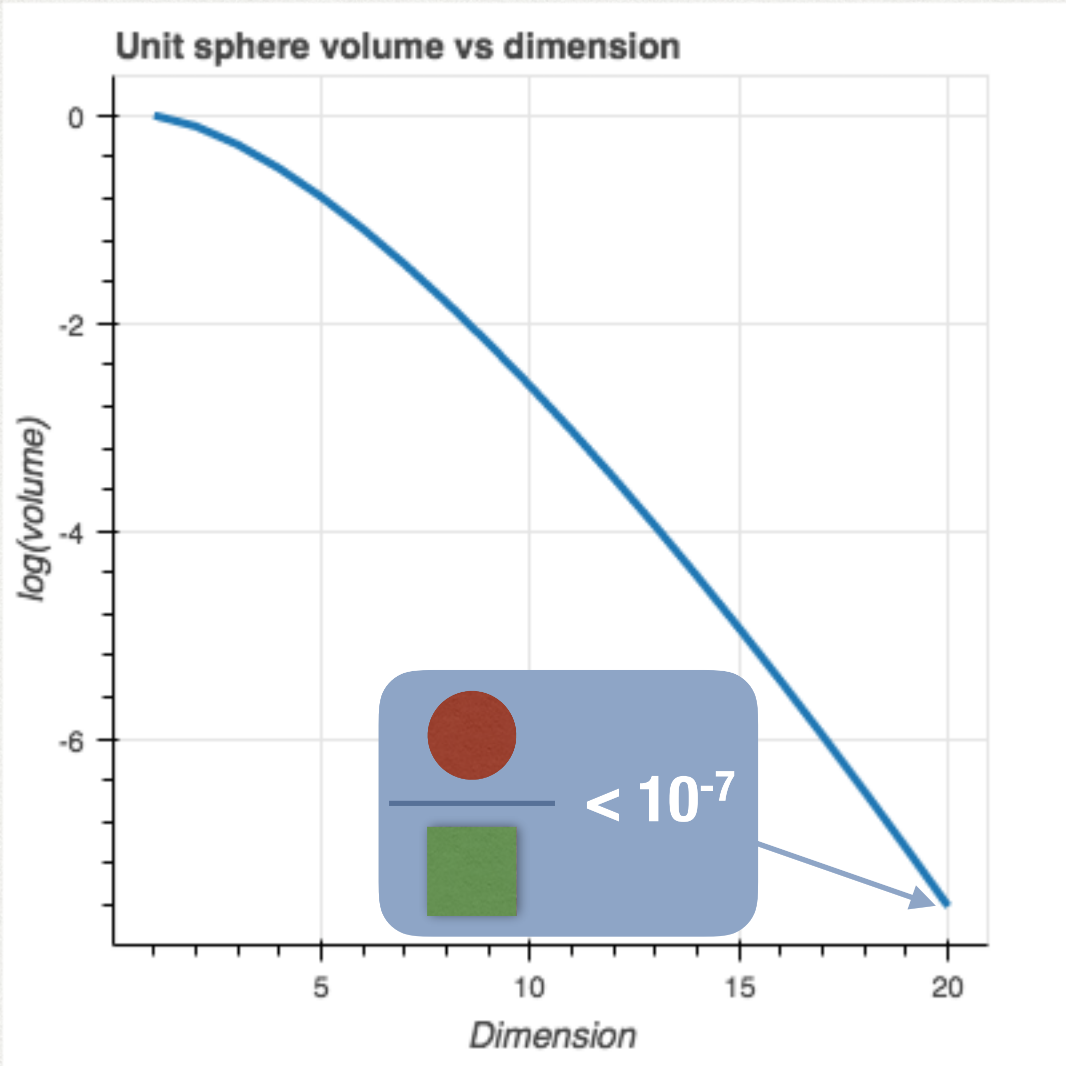
= 52.3 %

… ?

N-dimensional ball

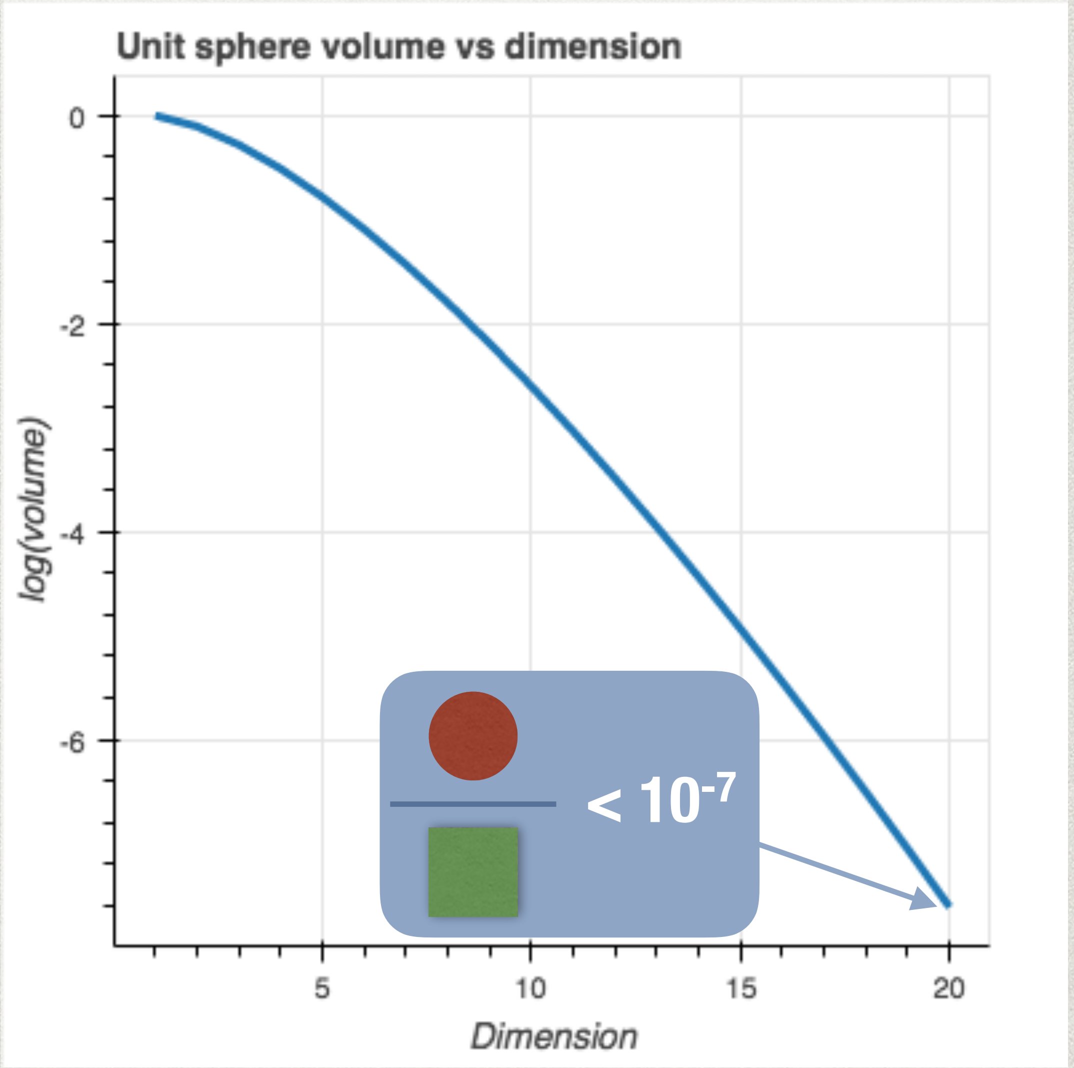


N-dimensional ball



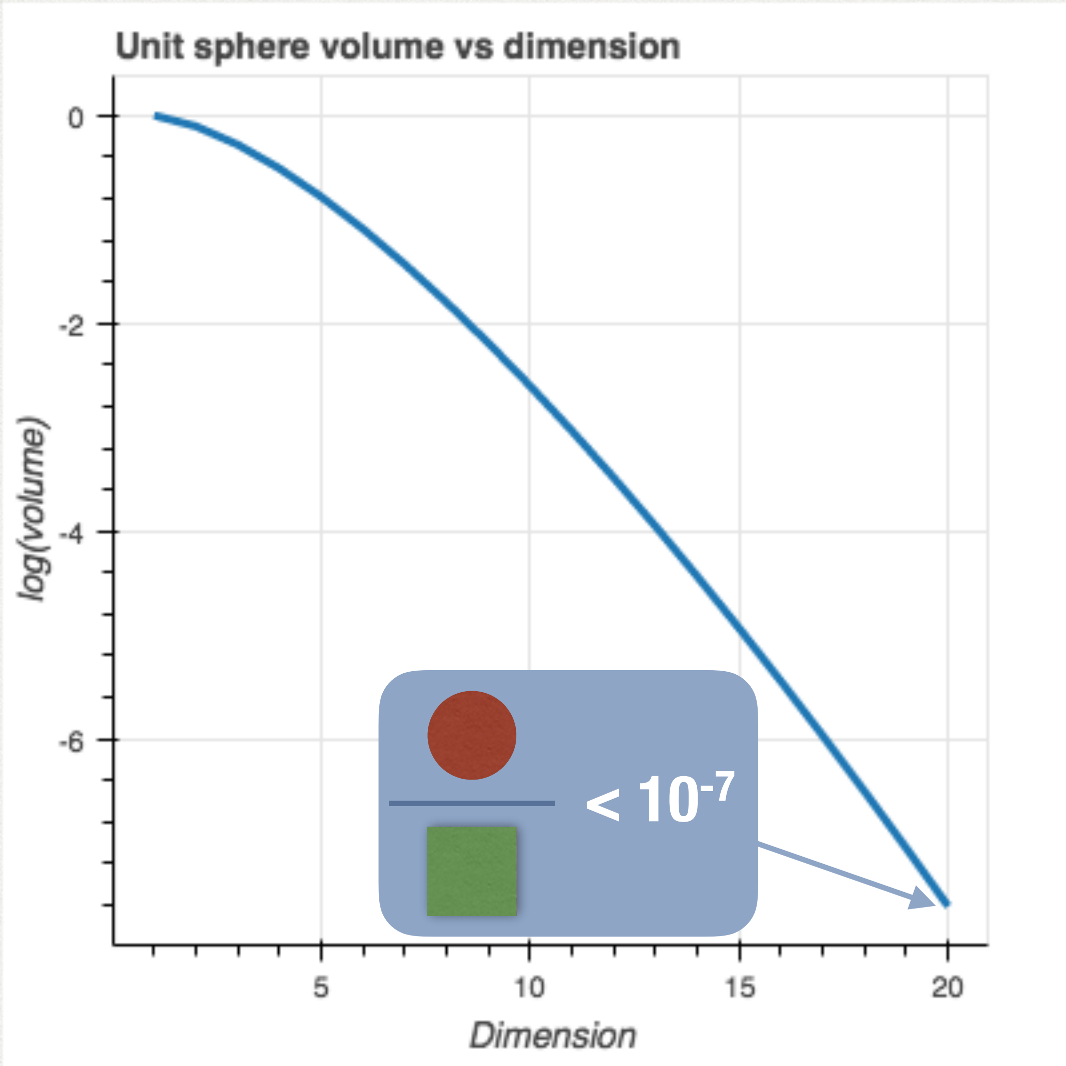
- The n-ball volume ratio tends to 0 (fast) in high dimension

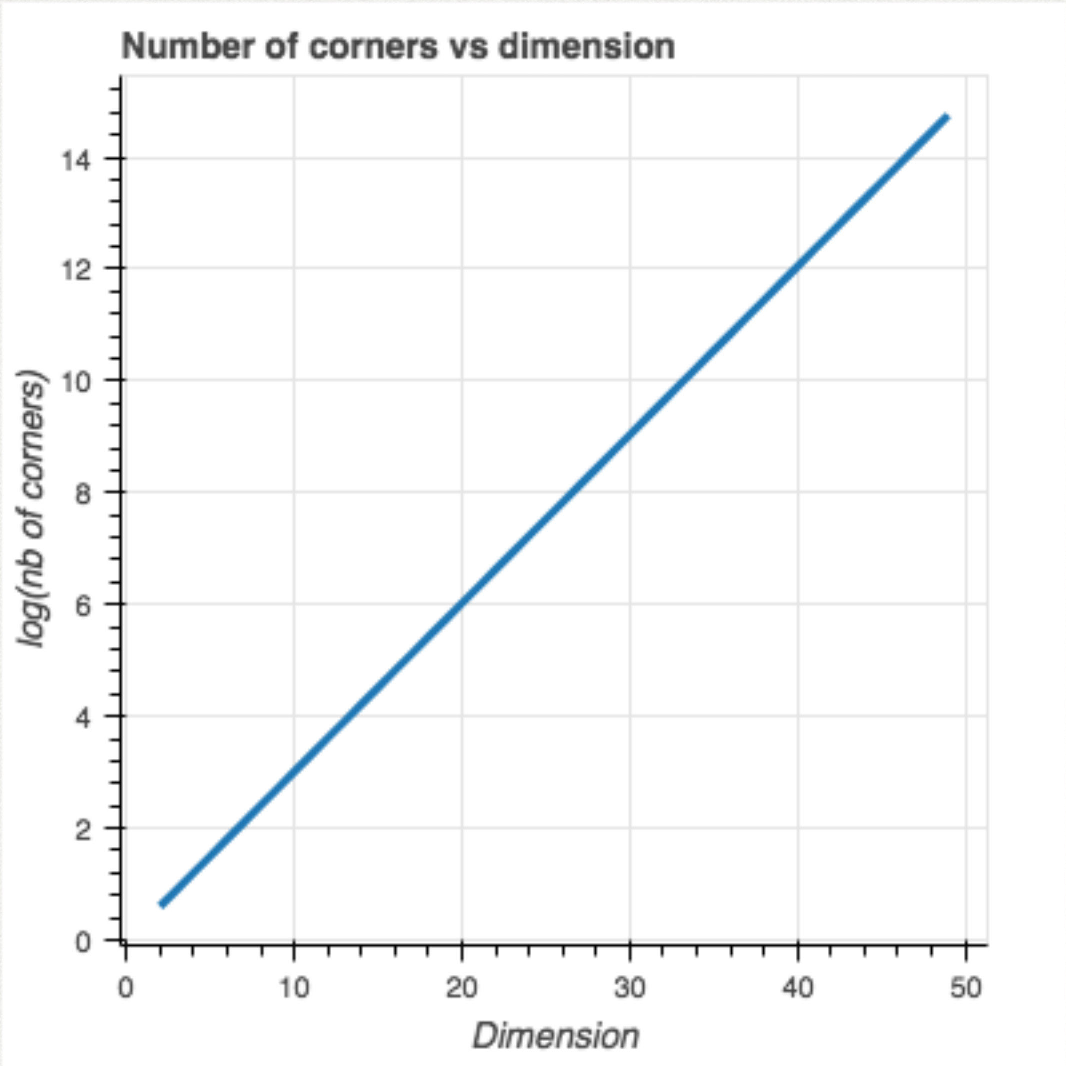
N-dimensional ball



- The n-ball volume ratio tends to 0 (fast) in high dimension
- Most points are « in the corners »

N-dimensional ball

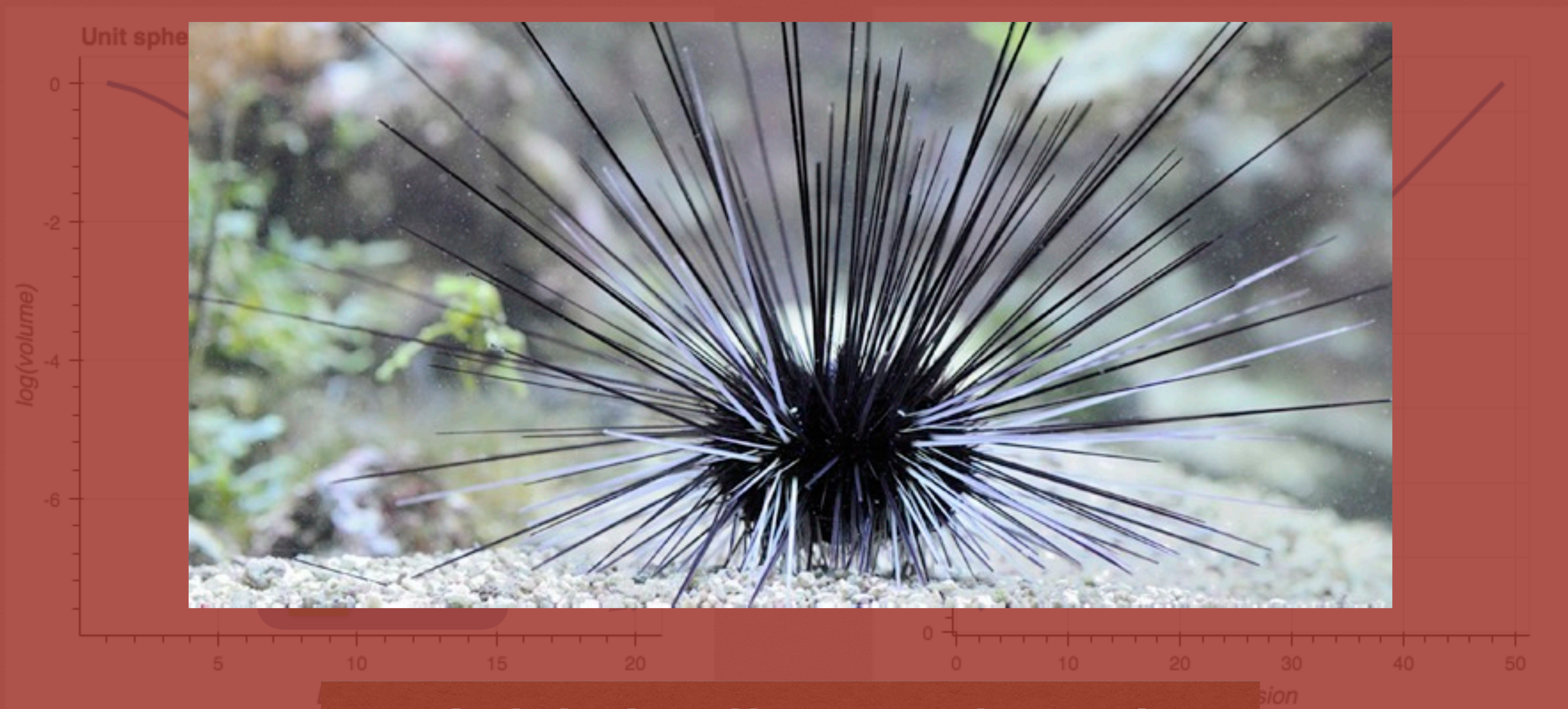




- The n-ball volume ratio tends to 0 (fast) in high dimension
- Most points are « in the corners »

- The number of corners explodes too!

N-dimensional ball

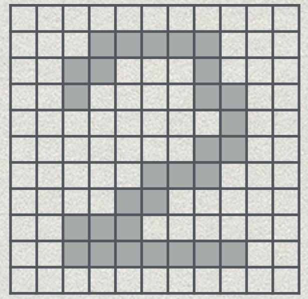


A high dimensional space looks like this…

- The n-ball volu… 0 (fast) in high… …f corners explodes too.
- Most points are « in the corners »

Samples in High Dimension

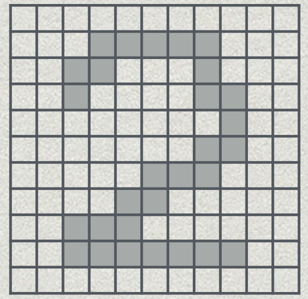
Example: 2D images (with 256 levels / channel)



Dimensions	Space size	Number of corners
784	$\mathbf{10^{1,888}}$	$\mathbf{10^{236}}$

Samples in High Dimension

Example: 2D images (with 256 levels / channel)



Dimensions	Space size	Number of corners
784	$10^{1,888}$	10^{236}
196,608	$10^{473,479}$	$10^{59,185}$

Samples in High Dimension

Samples in High Dimension

- Key take aways in high dimension
  - « I have a **LOT** of samples! »

Samples in High Dimension

- Key take aways in high dimension
  - « I have a **LOT** of samples! »
    => No, you don't. They are **very sparse**

Samples in High Dimension

- Key take aways in high dimension
  - « I have a **LOT** of samples! »
    => No, you don't. They are **<u>very sparse</u>**
  - You are learning a function using no points in the middle and almost none in the corners… See the problem?

Samples in High Dimension

- Key take aways in high dimension
  - « I have a **LOT** of samples! »
    => No, you don't. They are **very sparse**
  - You are learning a function using no points in the middle and almost none in the corners… See the problem?

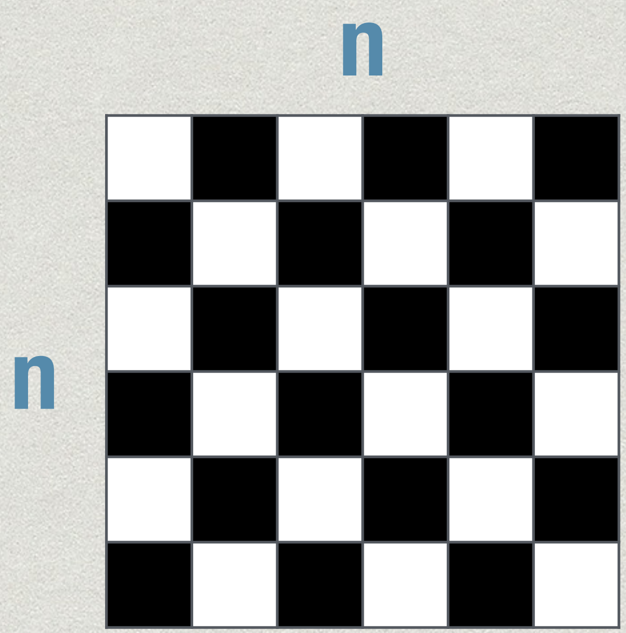
Machine learning $y = ax^b + c$

Deep learning $y = \sum_{i=0}^{N} a_i x^i$

This is a case of extreme generalization

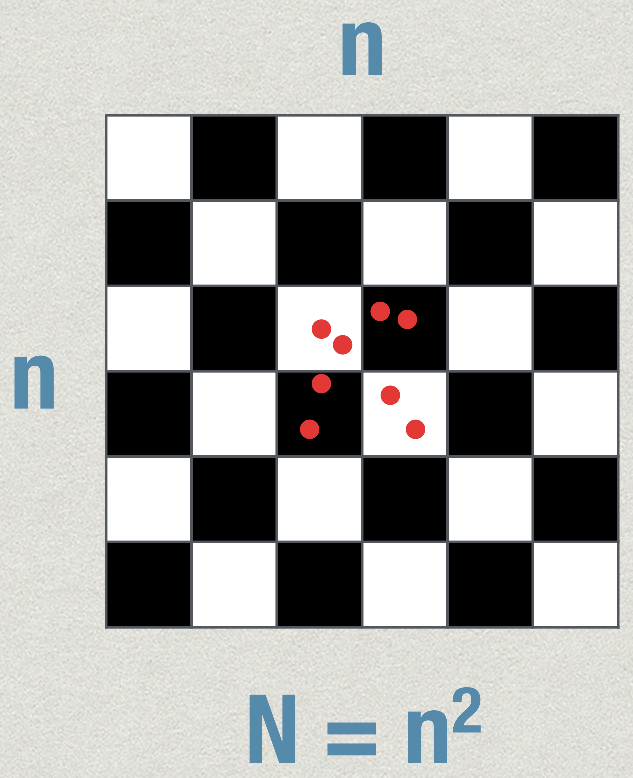
How do we solve this?

Using prior knowledge



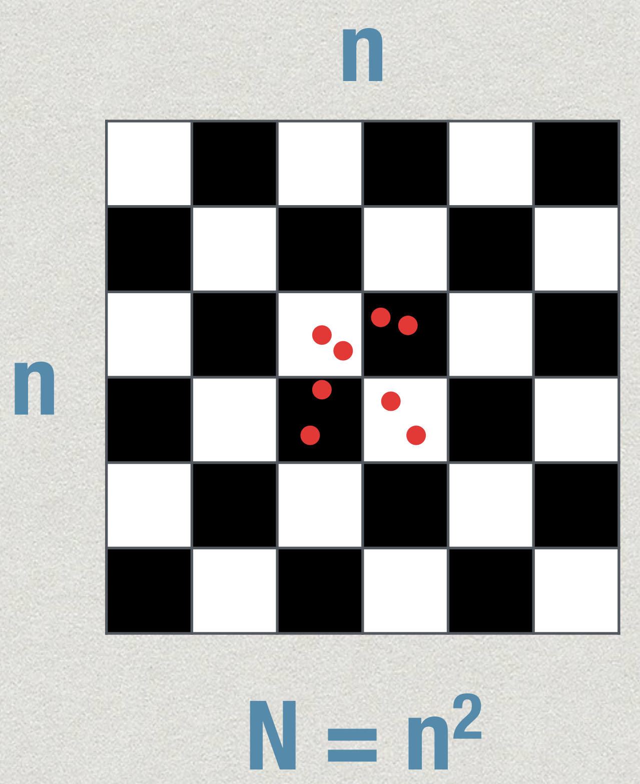
$N = n^2$

Using prior knowledge



- Classical machine learning (random forest, clustering…)
  - prior = smoothness
  - Number of examples needed: O(N) (several per square)

Using prior knowledge



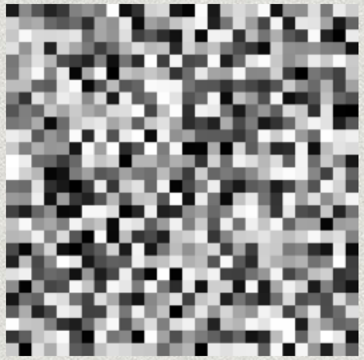
- Classical machine learning (random forest, clustering…)
  - prior = smoothness
  - Number of examples needed: O(N) (several per square)
- Deep neural network: build your own function using ***prior knowledge***
  - Ex: texture
  - End result: O(log(N)) points needed!

The manifold

Let's take 28x28 images :

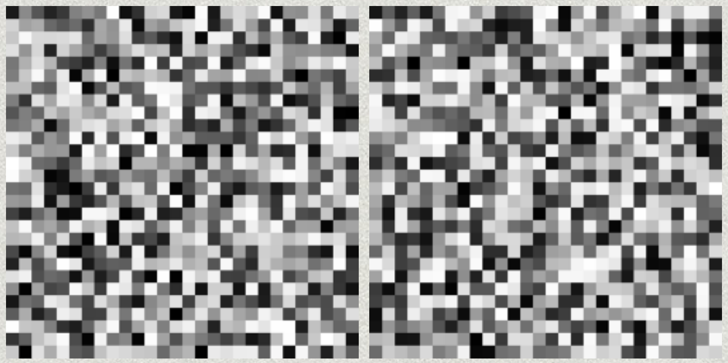
The manifold

Let's take 28x28 images :



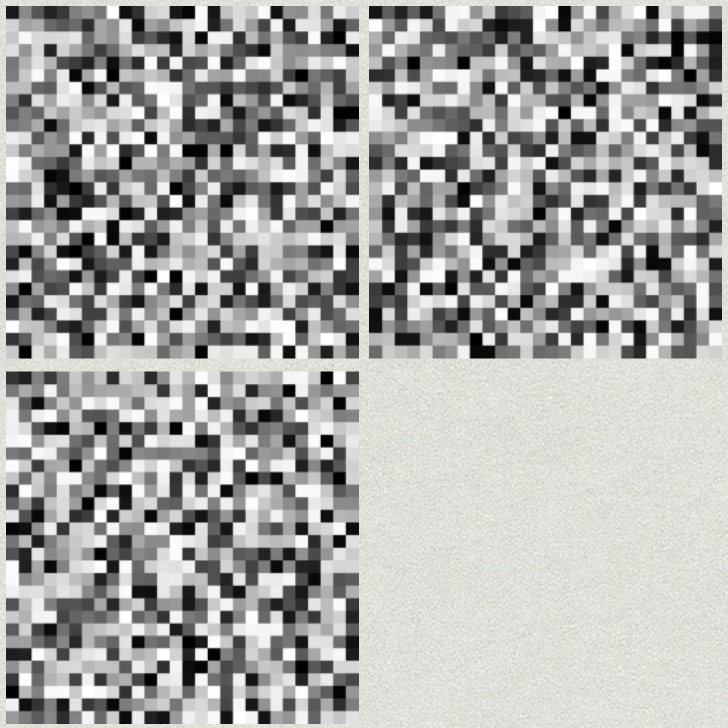
The manifold

Let’s take 28x28 images :



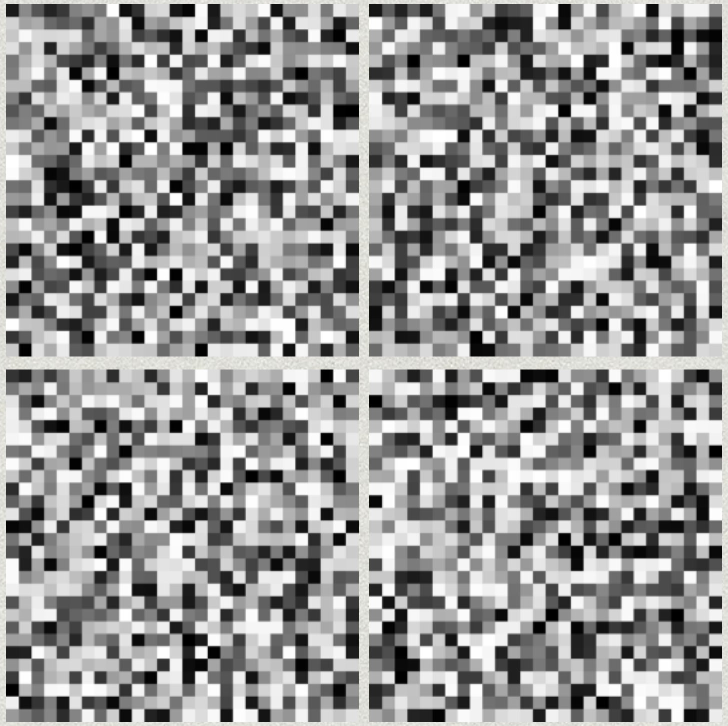
The manifold

Let's take 28x28 images :

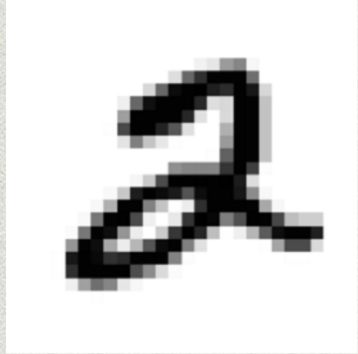


The manifold

Let's take 28x28 images :

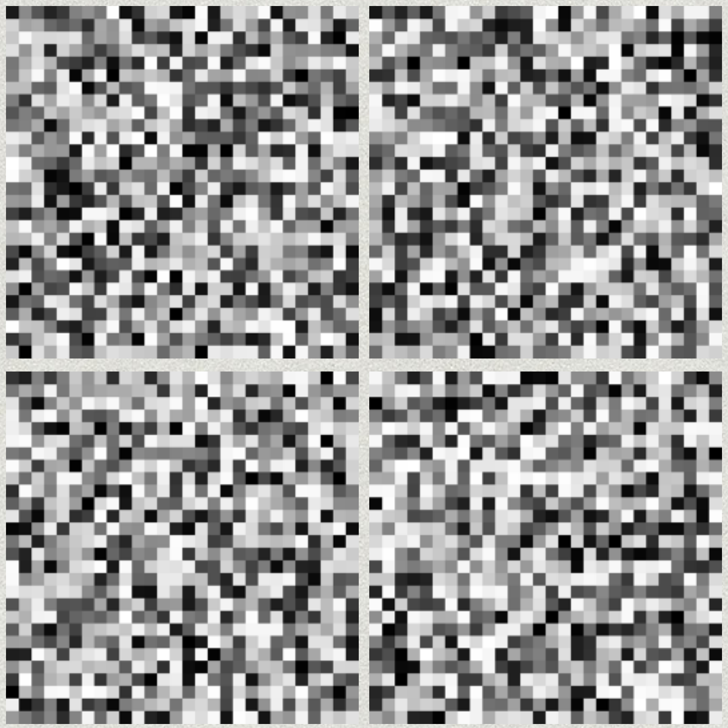


How long before I get this ?

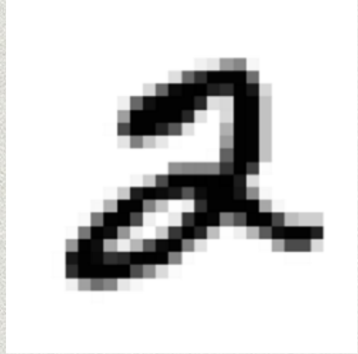


The manifold

Let's take 28x28 images :

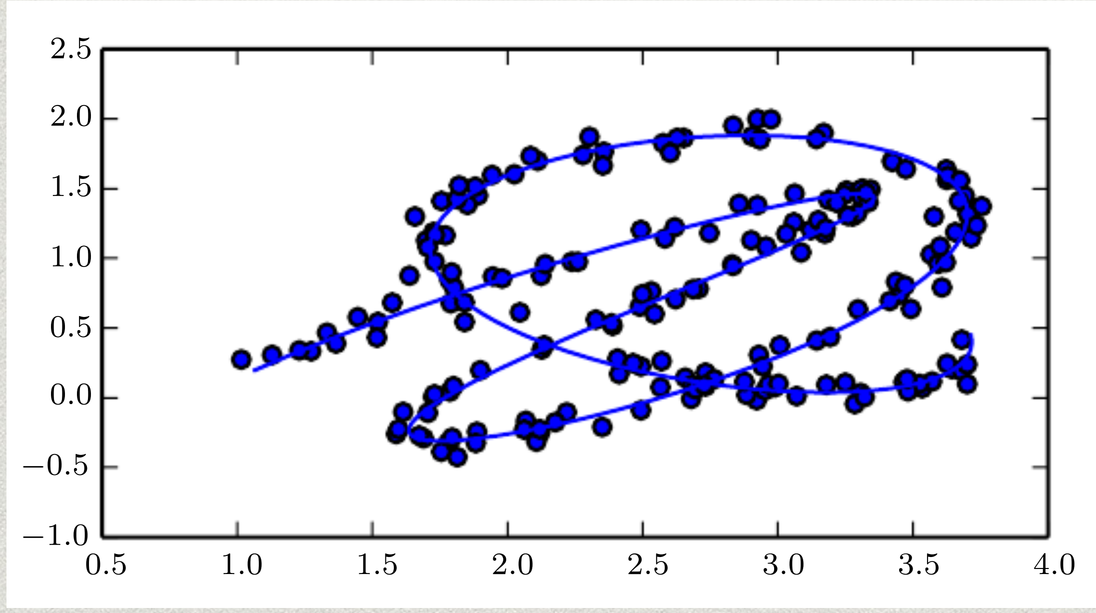


How long before I get this ?



The <u>manifold</u>

- The input space is huge
- But we don't need to cover it all
- We just need to describe the manifold of « relevant » points



Deep Learning Priors

- The data is *inside* a high dimensional space, but the manifold of interest is much smaller
- The data comes from a **composition of features**
- The features can assemble at several levels of hierarchy

Deep Learning Priors

- The data is *inside* a high dimensional space, but the manifold of interest is much smaller
- The data comes from a **composition of features**
- The features can assemble at several levels of hierarchy

Wait a minute… you said :

Machine learning $y = ax^b + c$

Deep learning $y = \sum_{i=0}^{N} a_i x^i$

Deep Learning Priors

- The data is *inside* a high dimensional space, but the manifold of interest is much smaller
- The data comes from a **composition of features**
- The features can assemble at several levels of hierarchy

Wait a minute… you said :

Machine learning $y = ax^b + c$

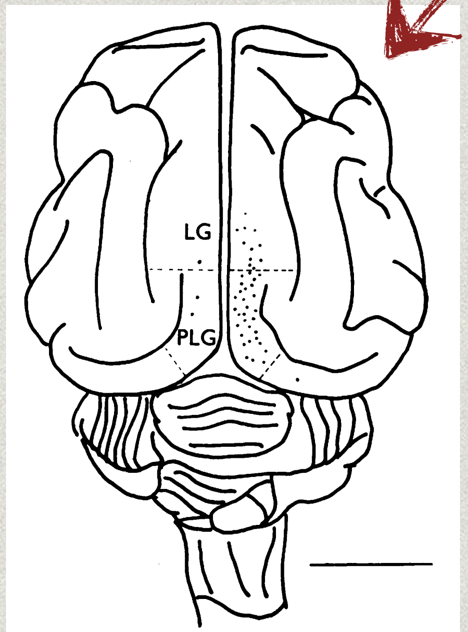
Deep learning $y = \sum_{i=0}^{N} a_i x^i$

This is not like general machine learning:
- you do not specify the fitted function
- you only give these « vague » priors
- they are enough for the function to focus on the manifold of « important » points

ARTIFICIAL NEURAL NETWORKS TO THE RESCUE

« Neural » Networks?

Cat brain (sorry…)

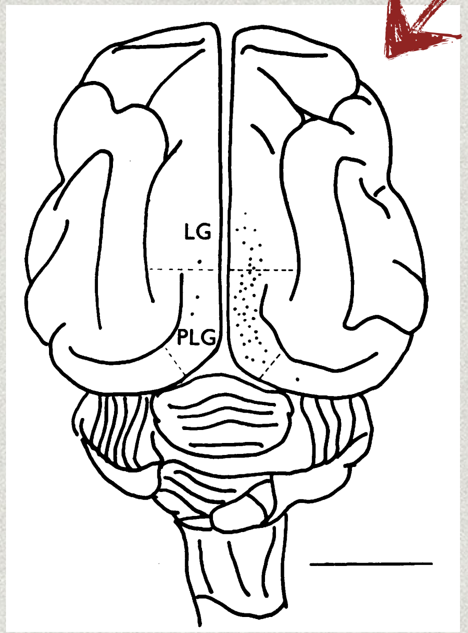


Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.

« Neural » Networks?

- Loosely inspired from biological systems

Cat brain (sorry…)

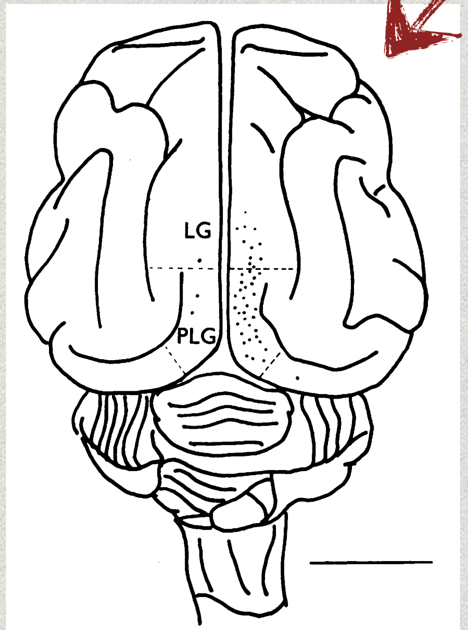


Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.

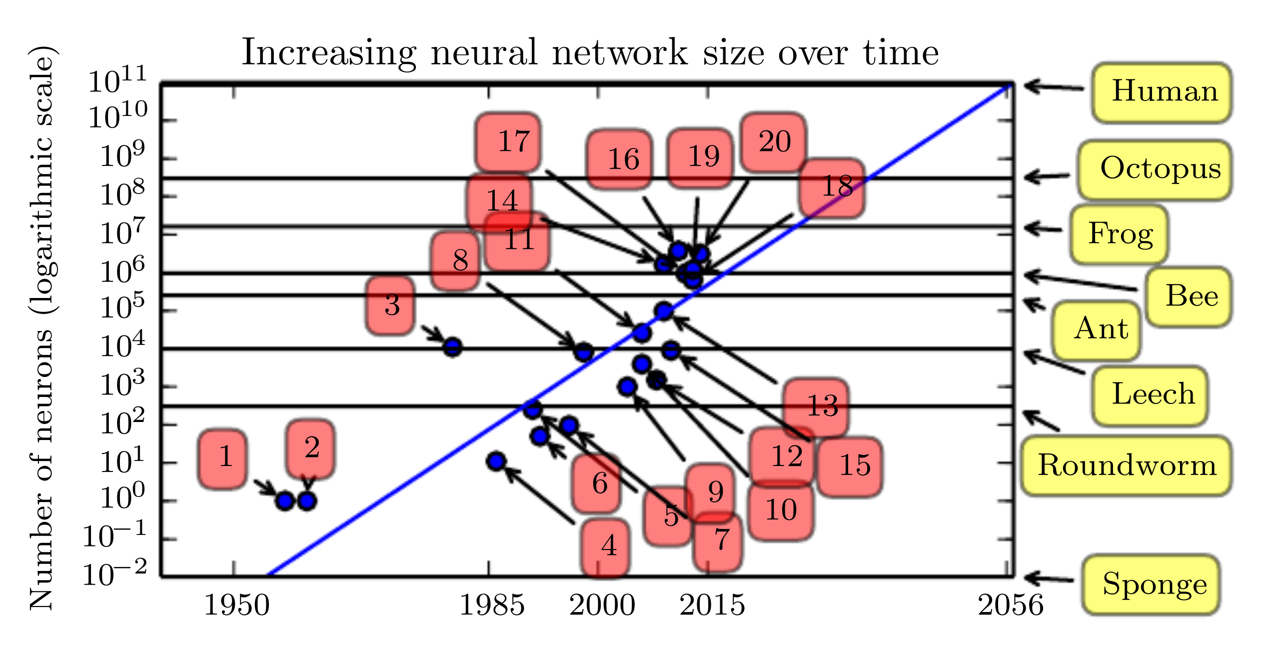
« Neural » Networks?

- Loosely inspired from biological systems
- Number of neurons and connections now reaching mammalian values

Cat brain (sorry…)



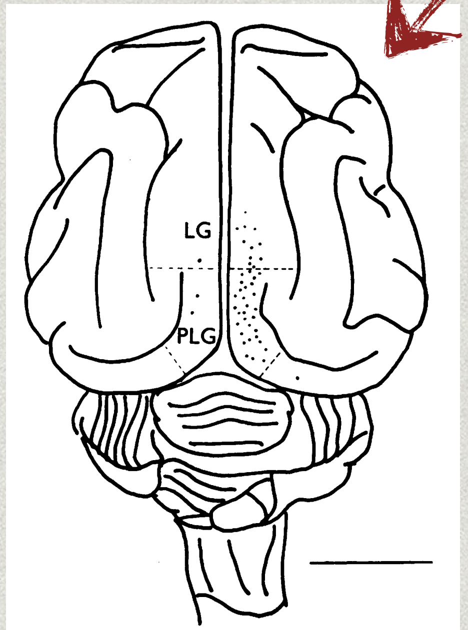
Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.



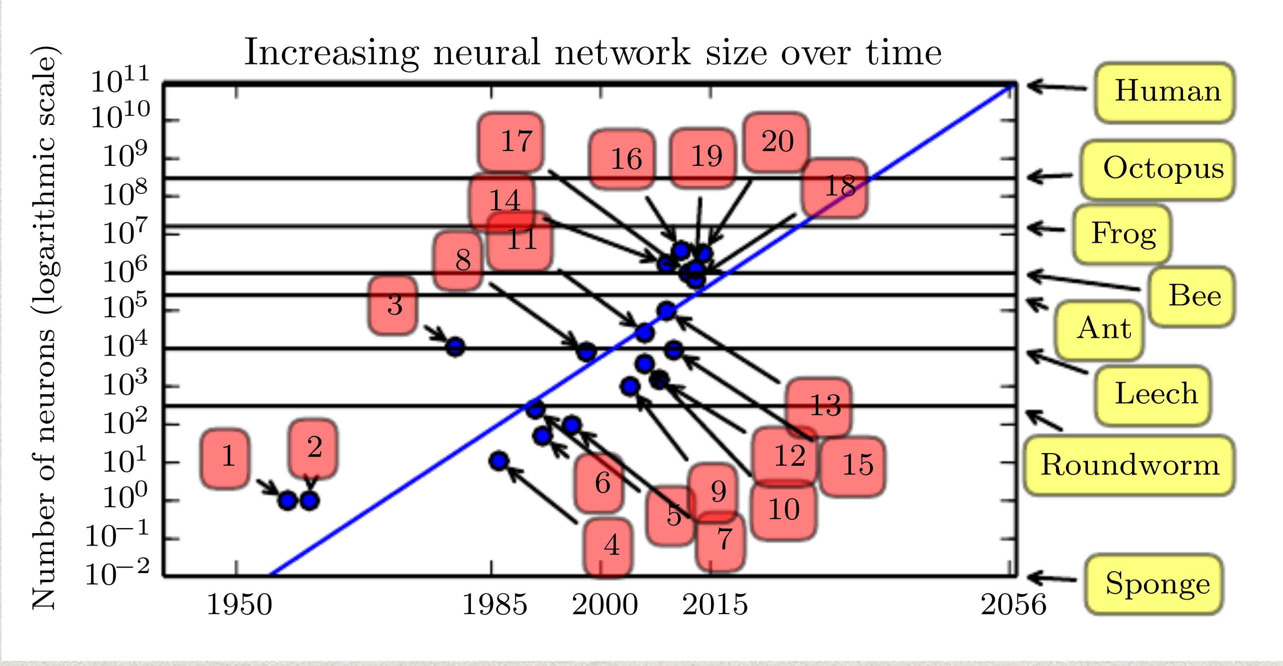
« Neural » Networks?

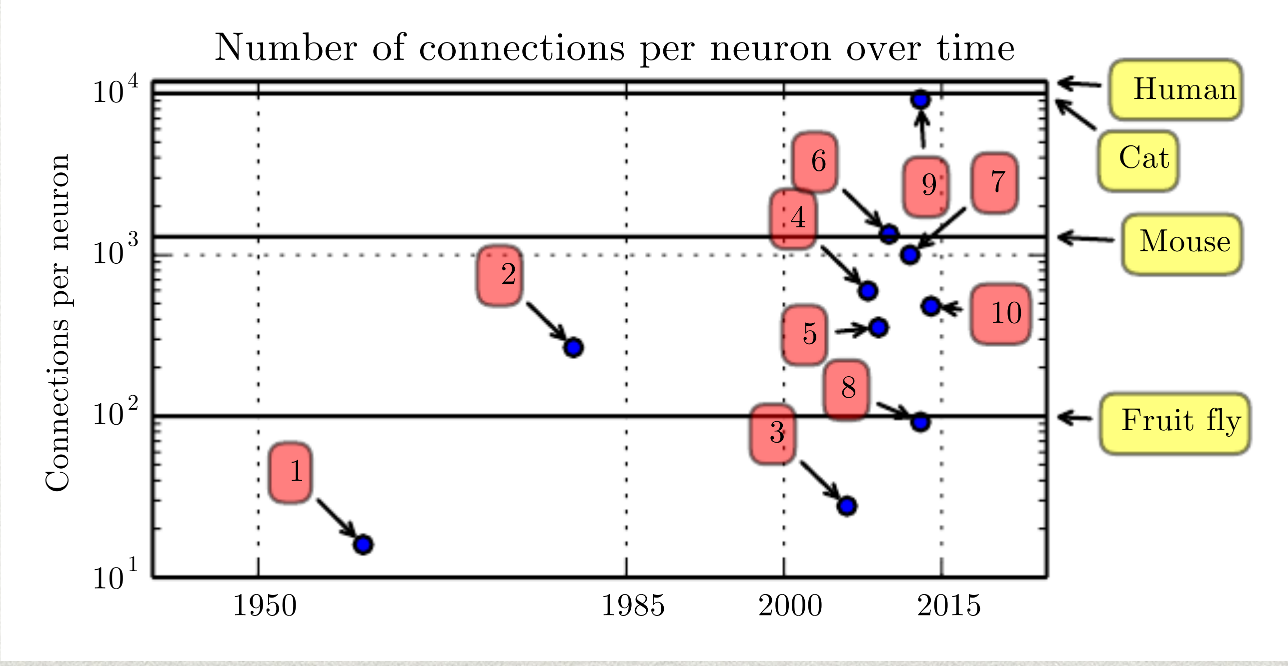
* Loosely inspired from biological systems
* Number of neurons and connections now reaching mammalian values

Cat brain (sorry…)

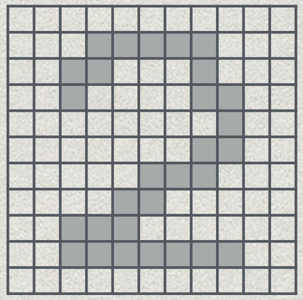


Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.

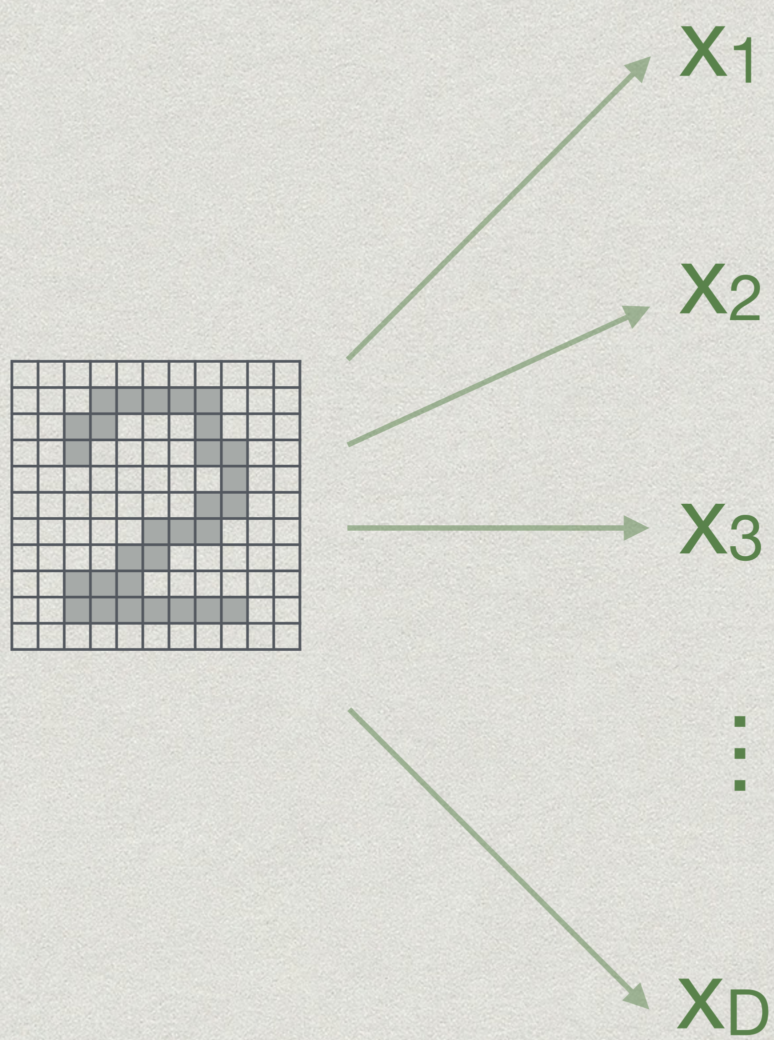




What is a « neuron »?

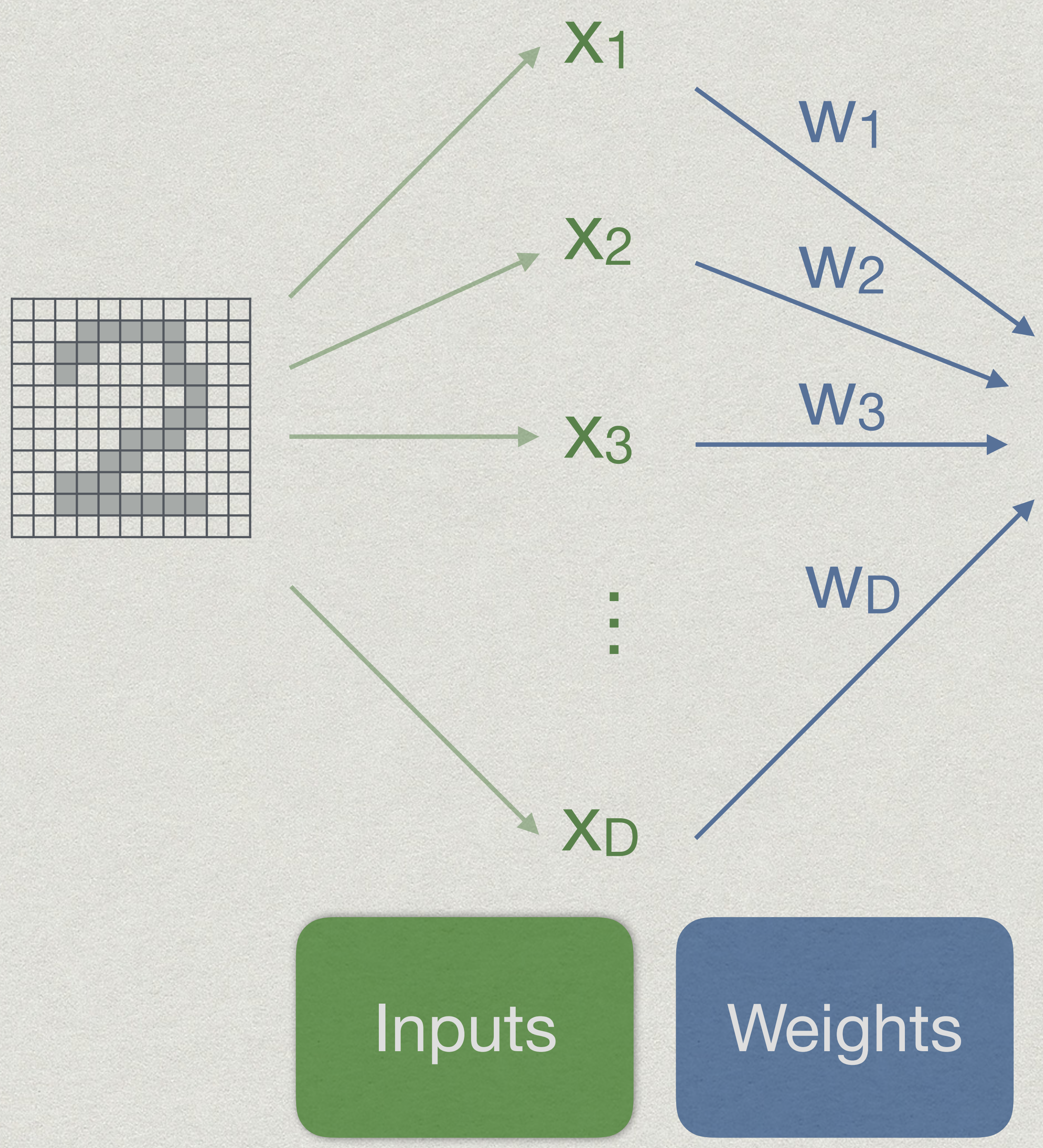


What is a « neuron »?

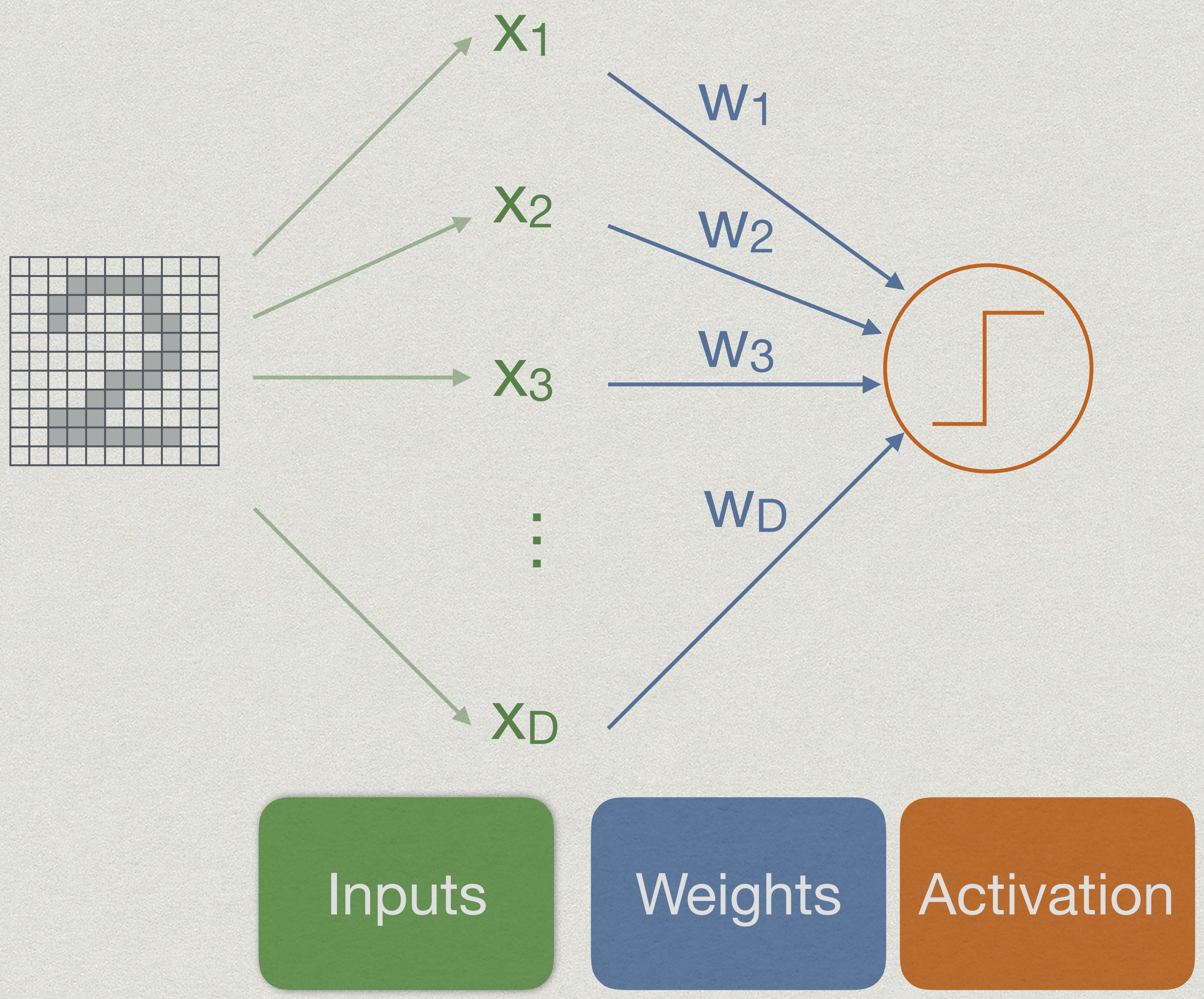


Inputs

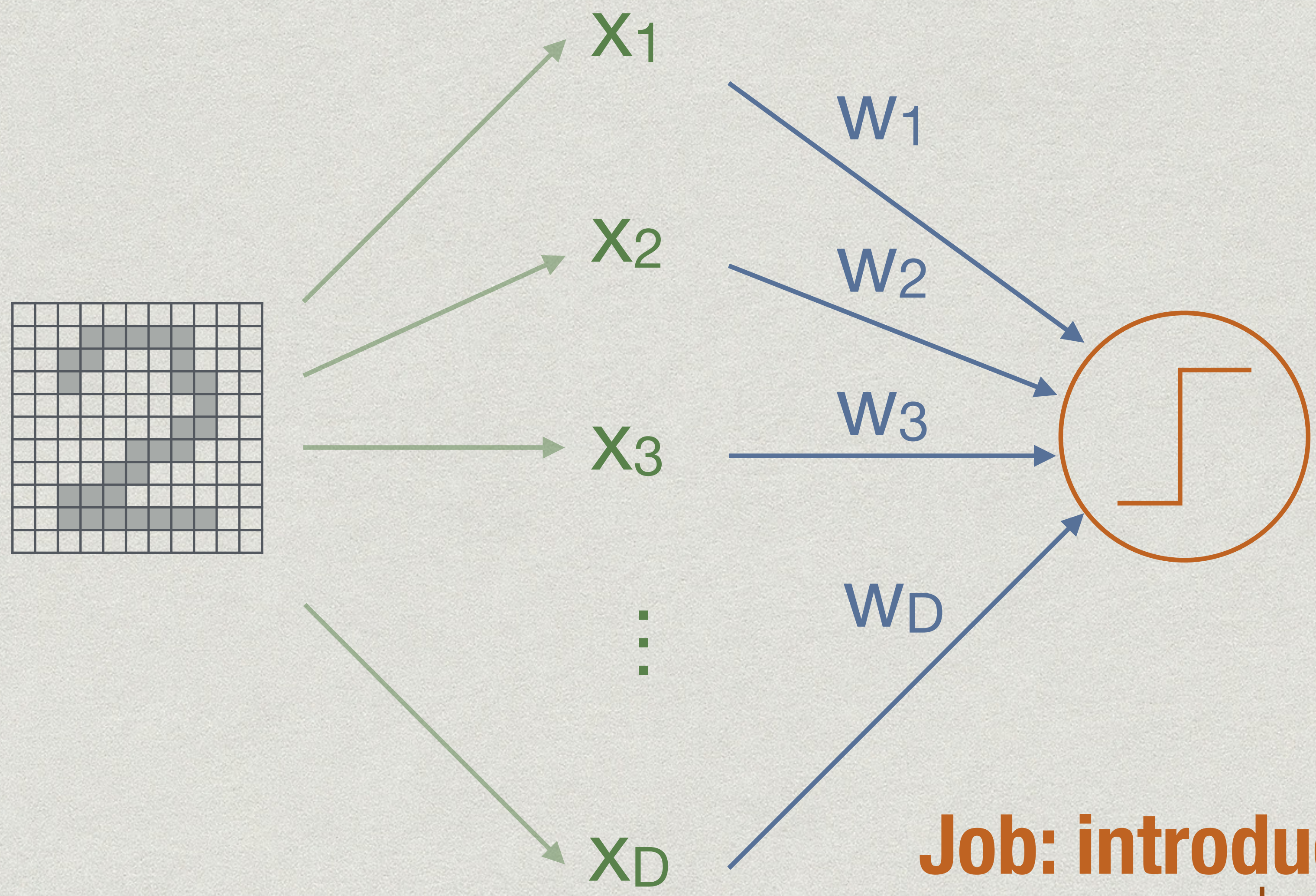
What is a « neuron »?



What is a « neuron »?



What is a « neuron »?



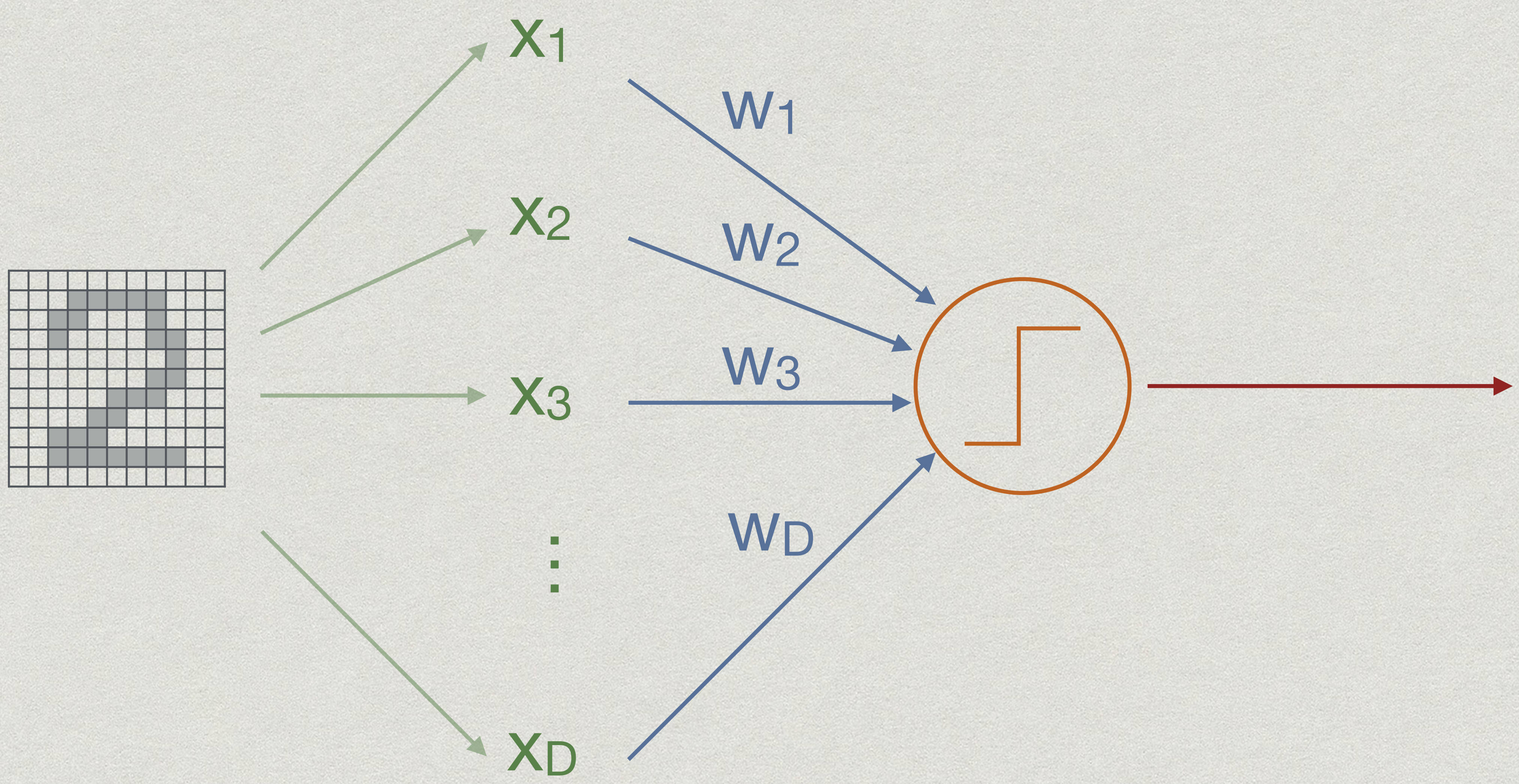
Job: introduce non-linearity

Inputs

Weights

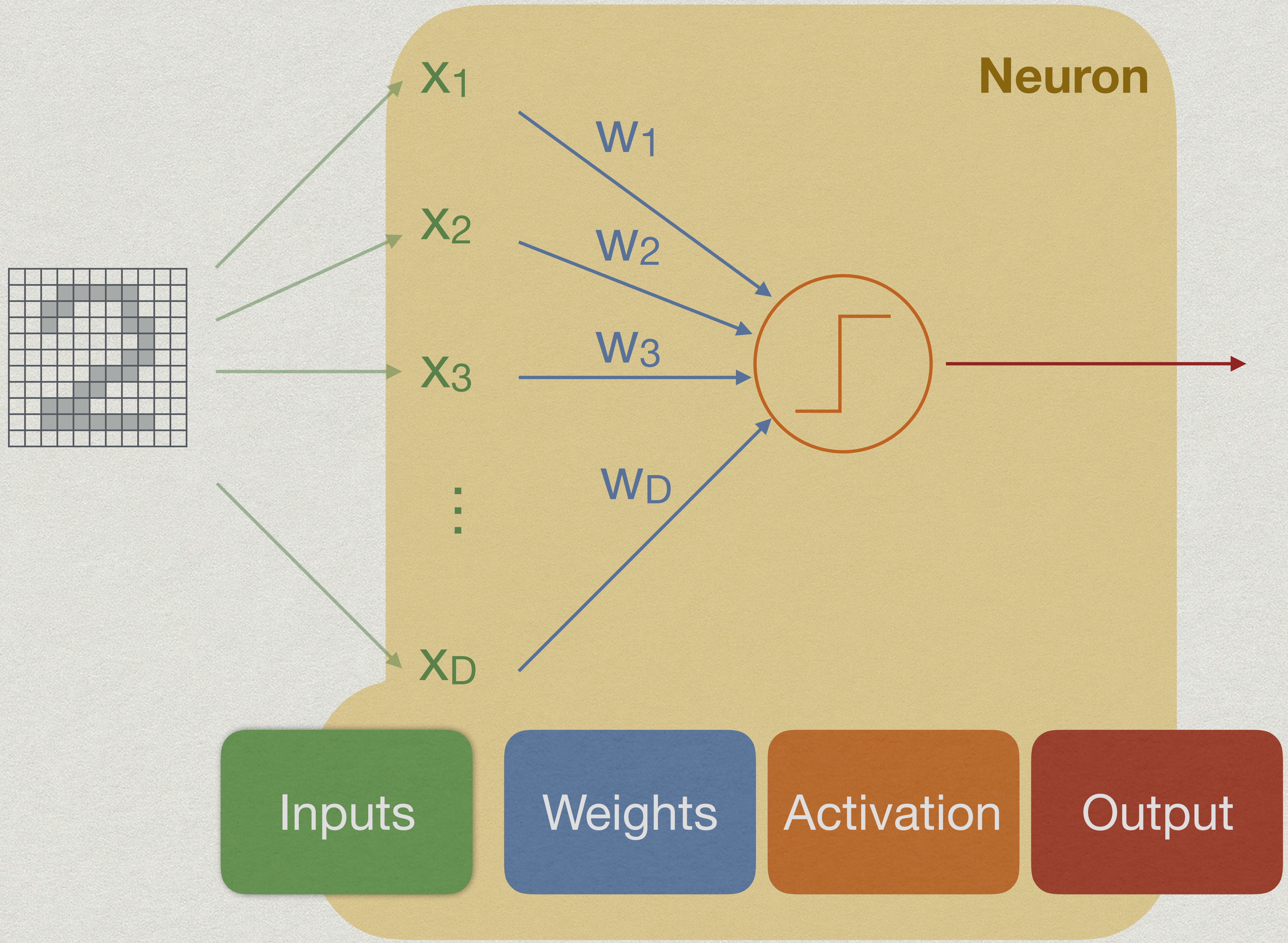
Activation

What is a « neuron »?

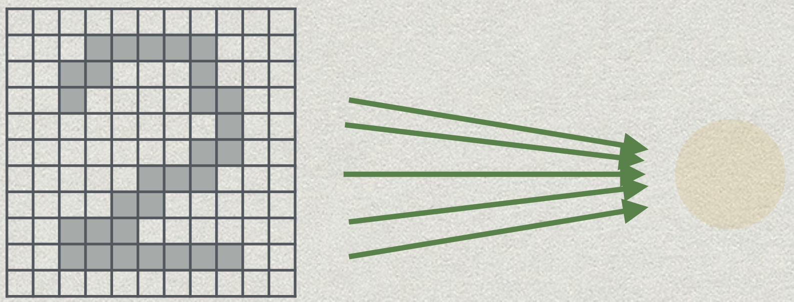


Inputs | Weights | Activation | Output

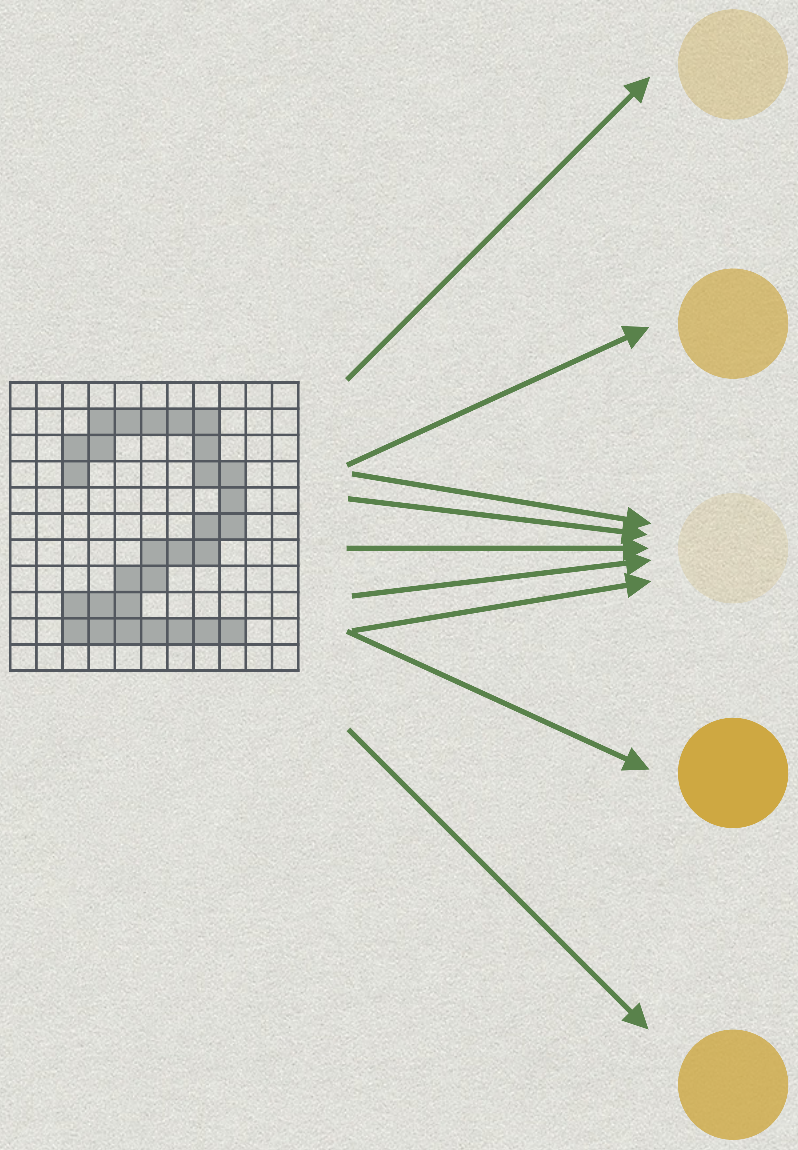
What is a « neuron »?



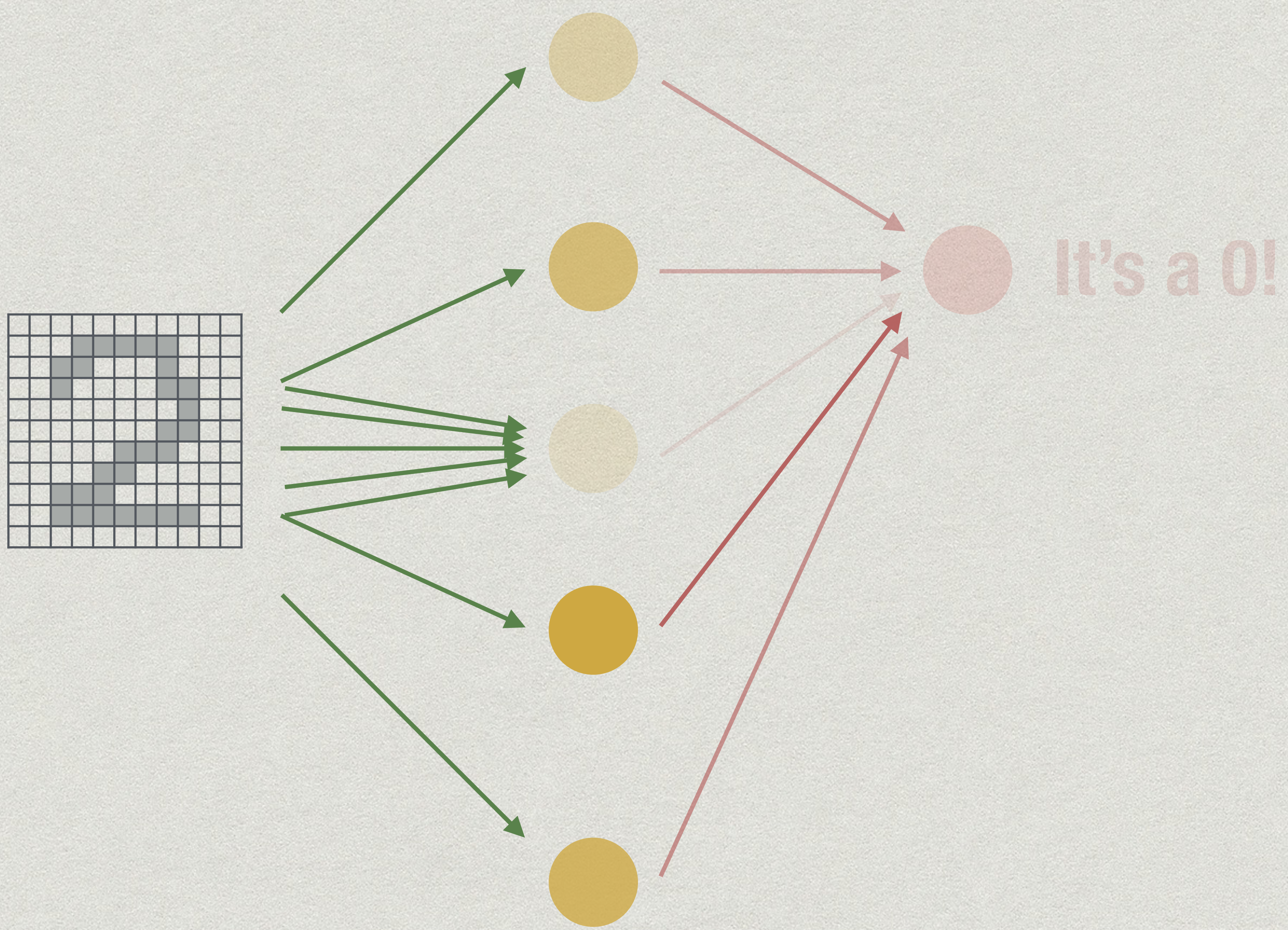
Assembling neurons



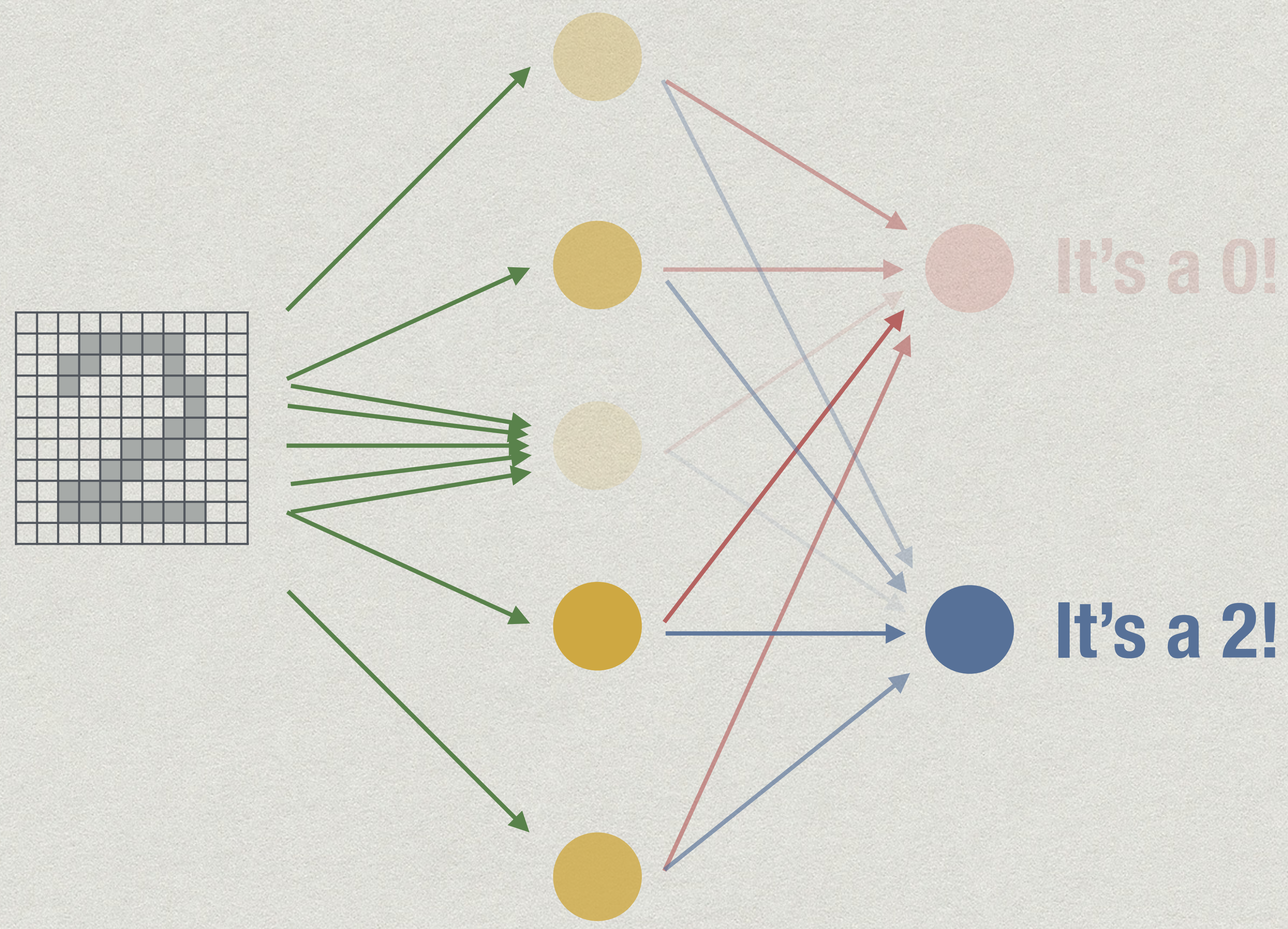
Assembling neurons



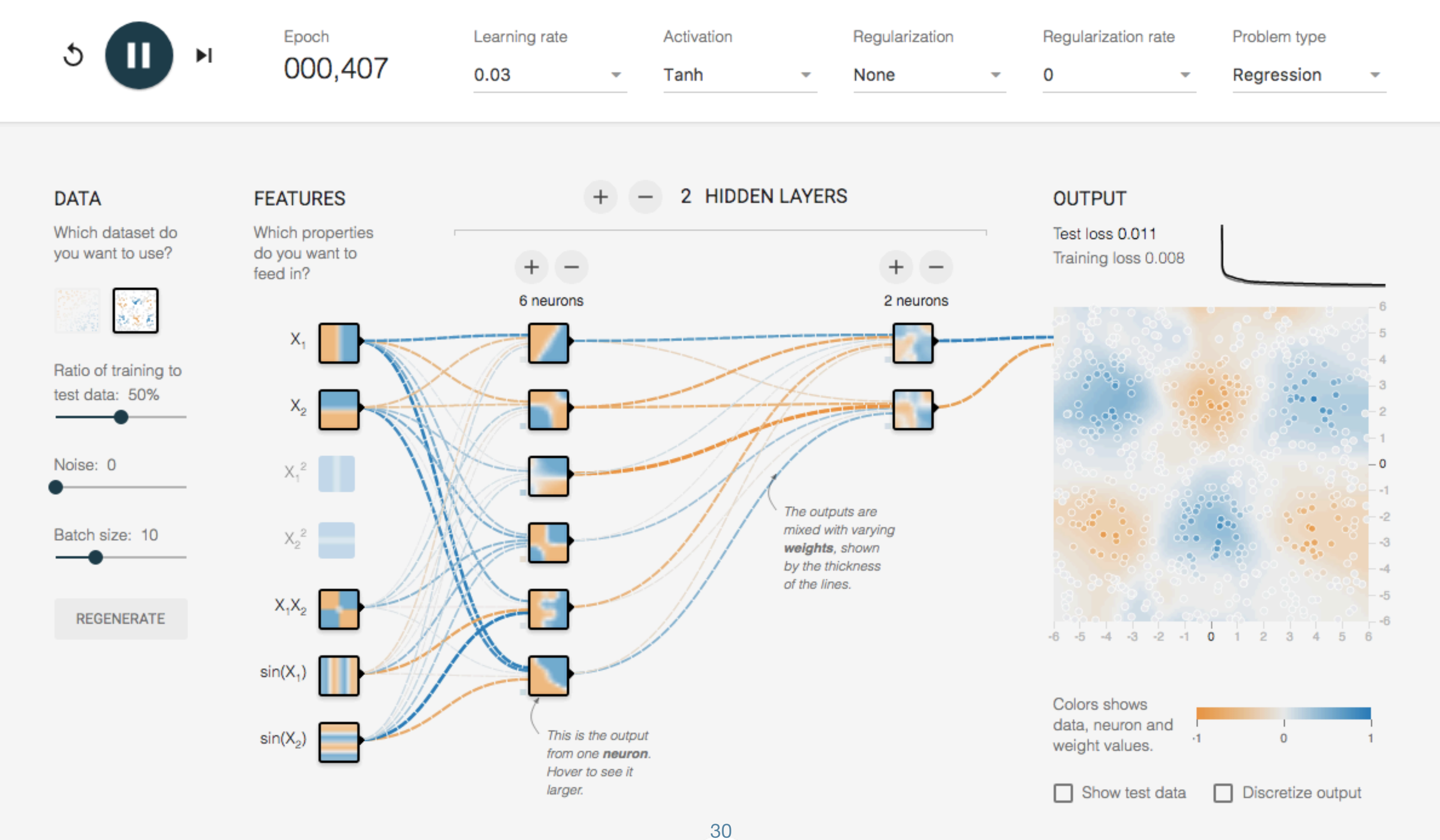
Assembling neurons



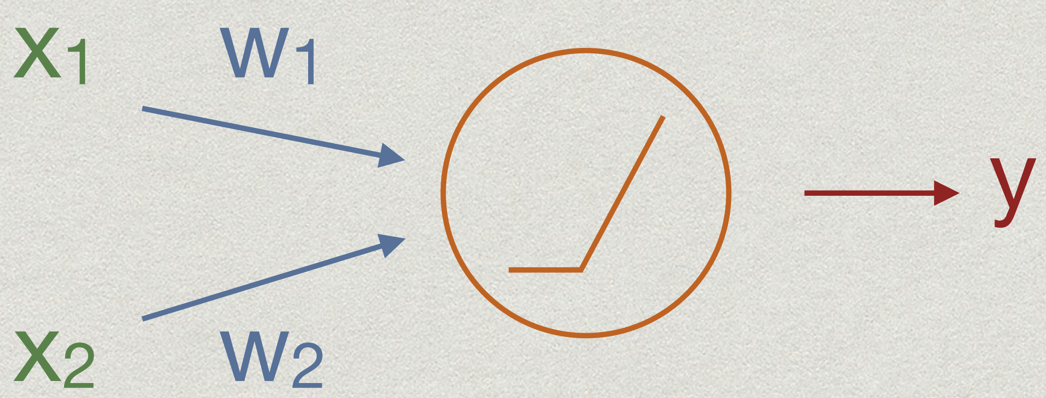
Assembling neurons



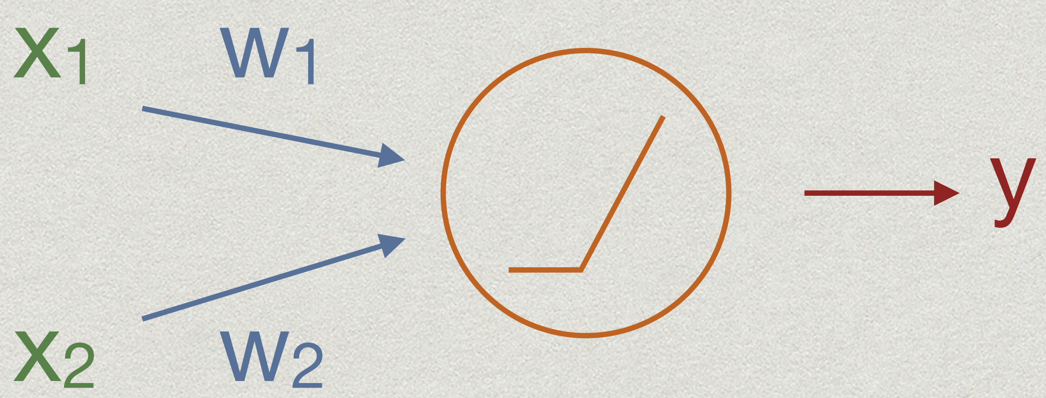
Play around! http://playground.tensorflow.org/



Is my function a neuron?

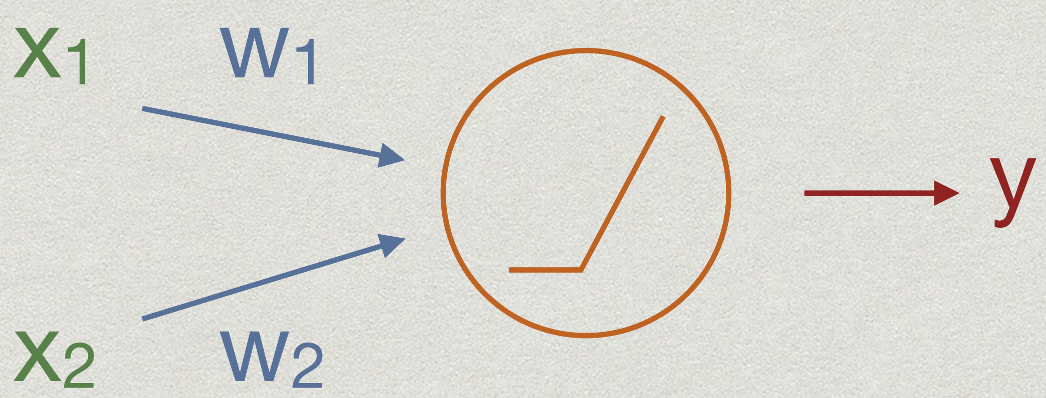


Is my function a neuron?



$$y = \max(x_1 w_1 + x_2 w_2, 0)$$

Is my function a neuron?



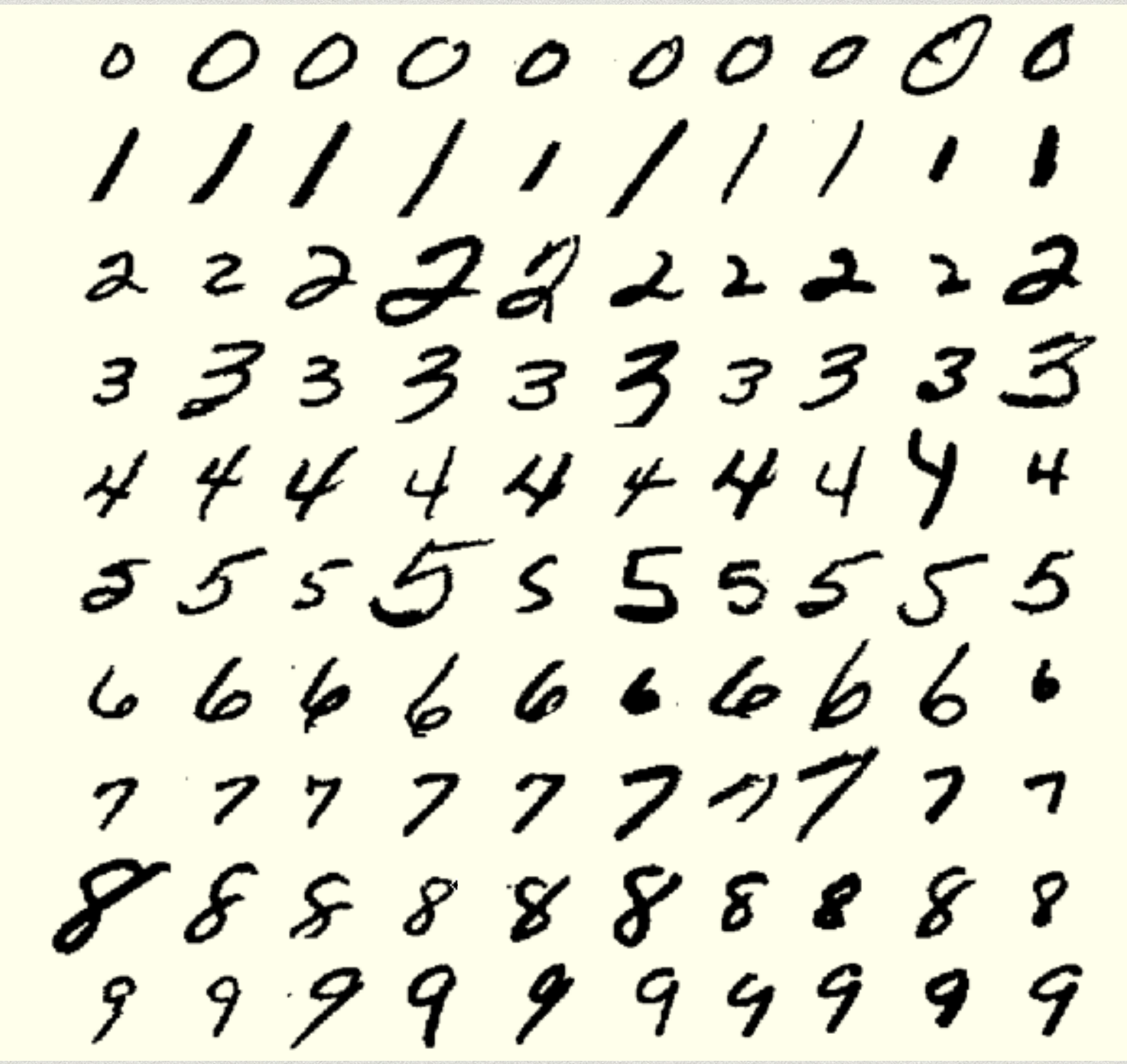
$$y = \max(x_1 w_1 + x_2 w_2, 0)$$

- Neural nets are just computational graphs
- You can represent many (any?) operations with neural nets
- But it does not mean you are doing deep learning…

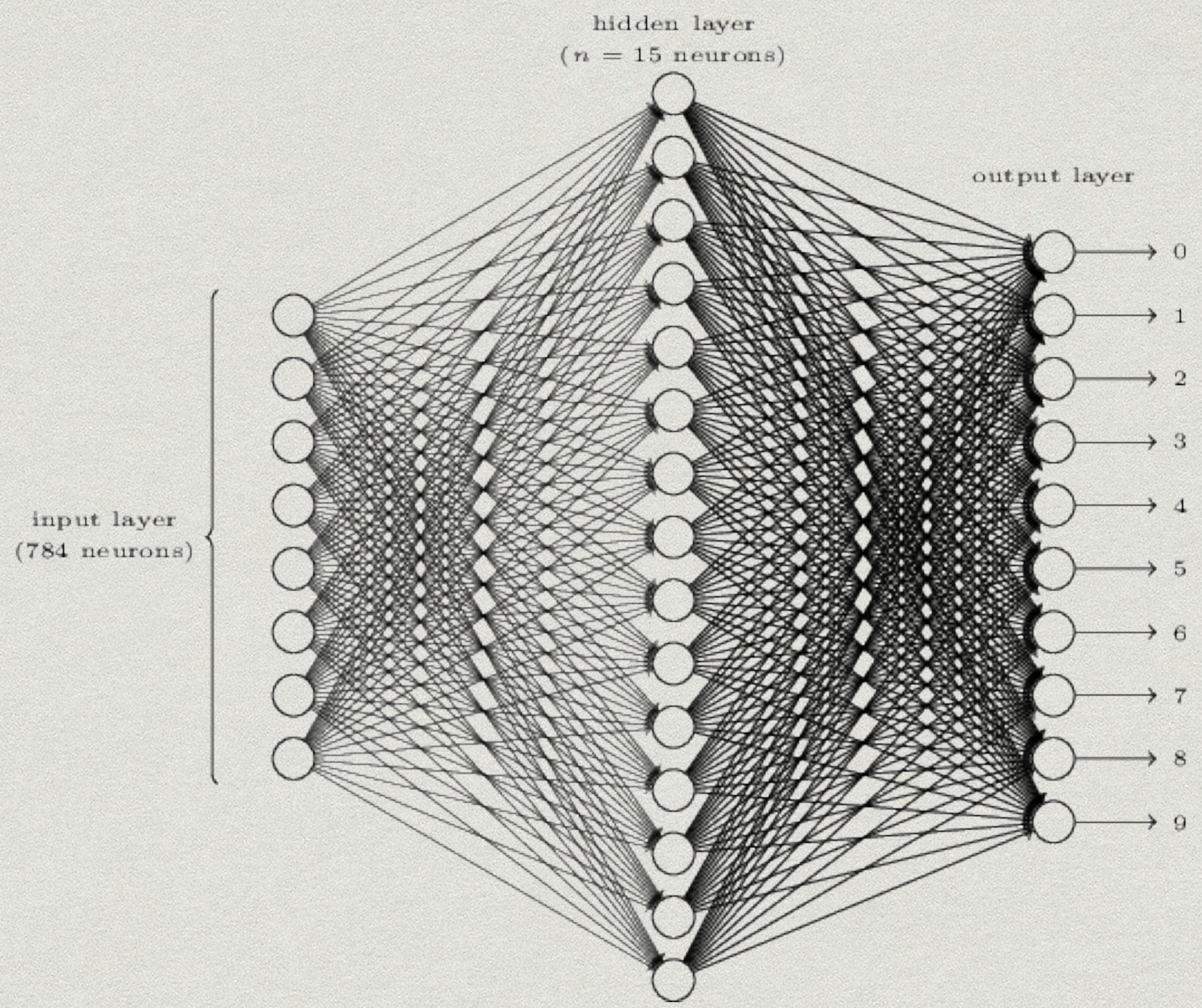
DEEP CLASSIFIERS

Example: the MNIST dataset

- Large database of handwritten digits (stored as 28x28 pixel images)
- You recognize these instantly…
- … but how can we build a net that recognizes them?

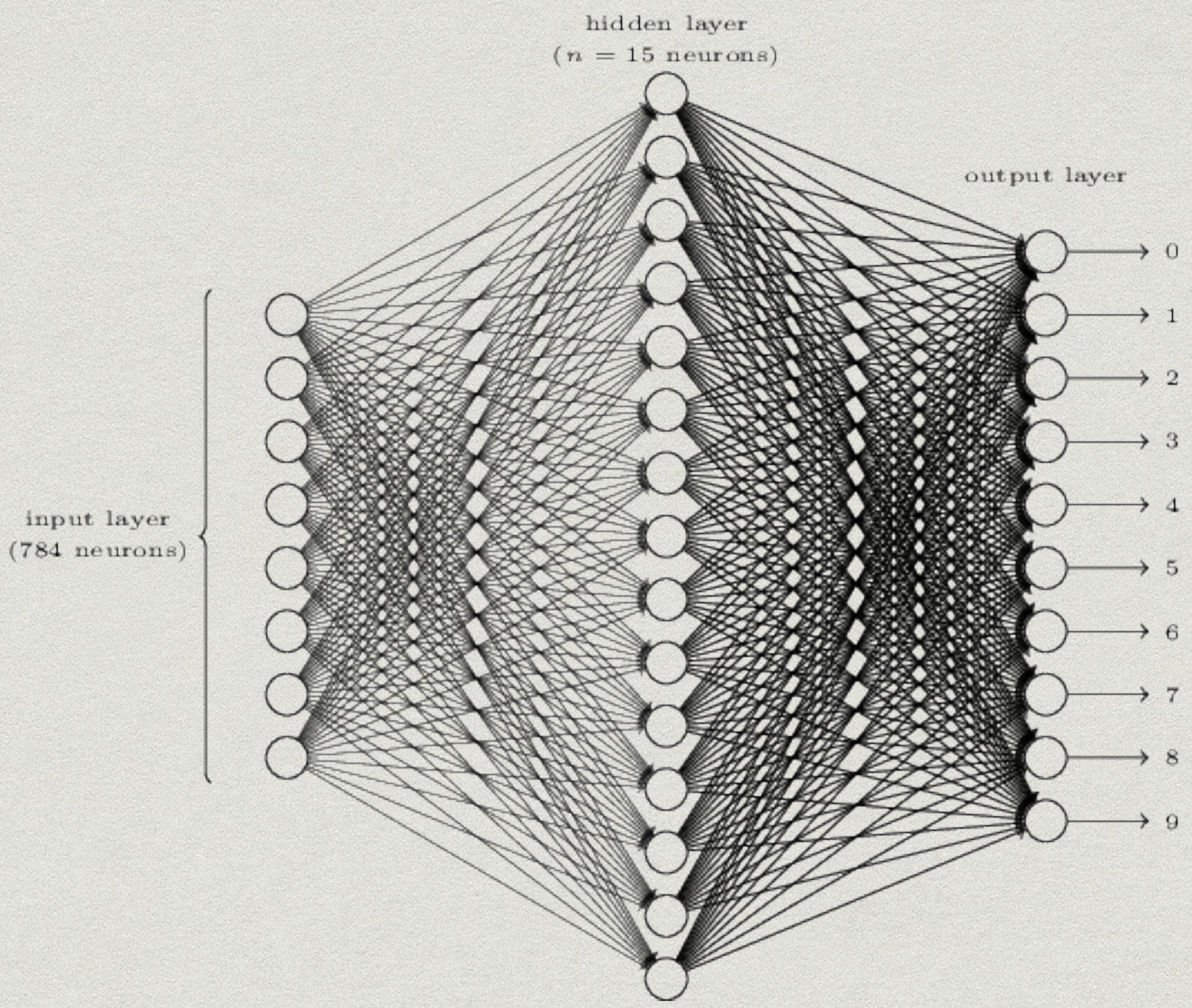


A simple neural net



- Simple « Multi-Layer Perceptron » (MLP)
  - 28x28 = 784 pixels on input
  - 0 → 9: 10 outputs
  - 1 hidden layer

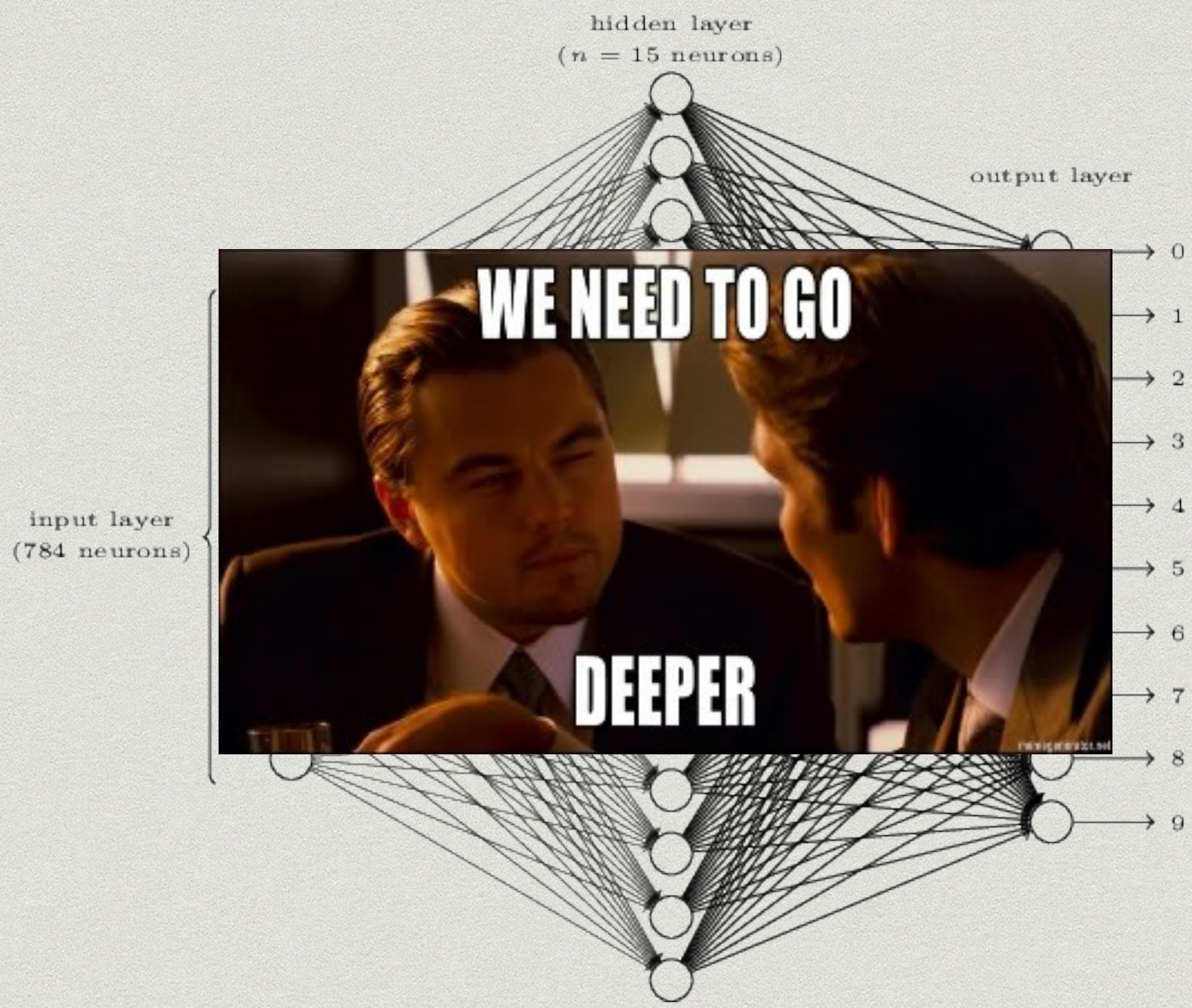
A simple neural net



- Simple « Multi-Layer Perceptron » (MLP)
  - 28x28 = 784 pixels on input
  - 0 → 9: 10 outputs
  - 1 hidden layer

91.5% accuracy
= 8.5% error

A simple neural net



- Simple « Multi-Layer Perceptron » (MLP)
  - 28x28 = 784 pixels on input
  - 0 → 9: 10 outputs
  - 1 hidden layer

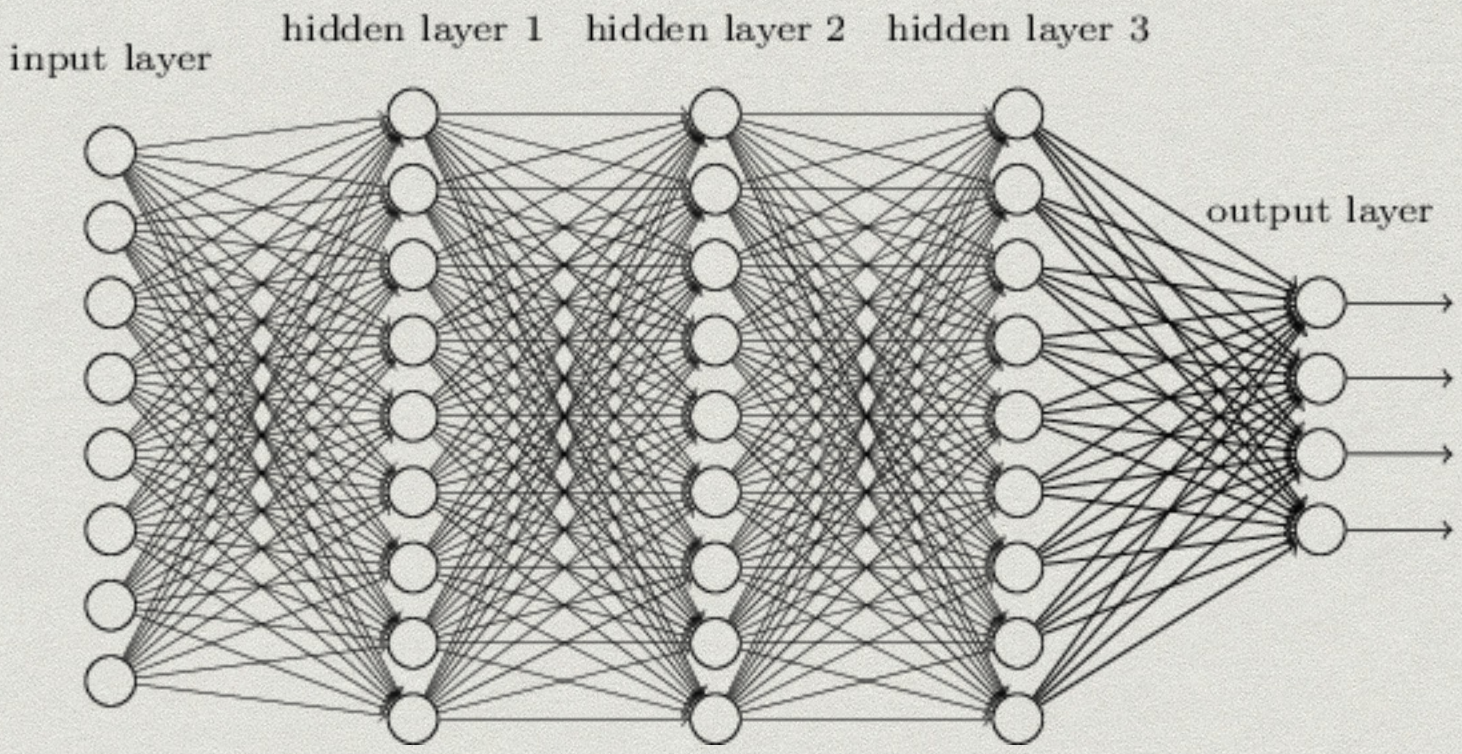
91.5% accuracy
= 8.5% error

Easy right?

With enough neurons and depth, you can replicate any function!

Easy right?

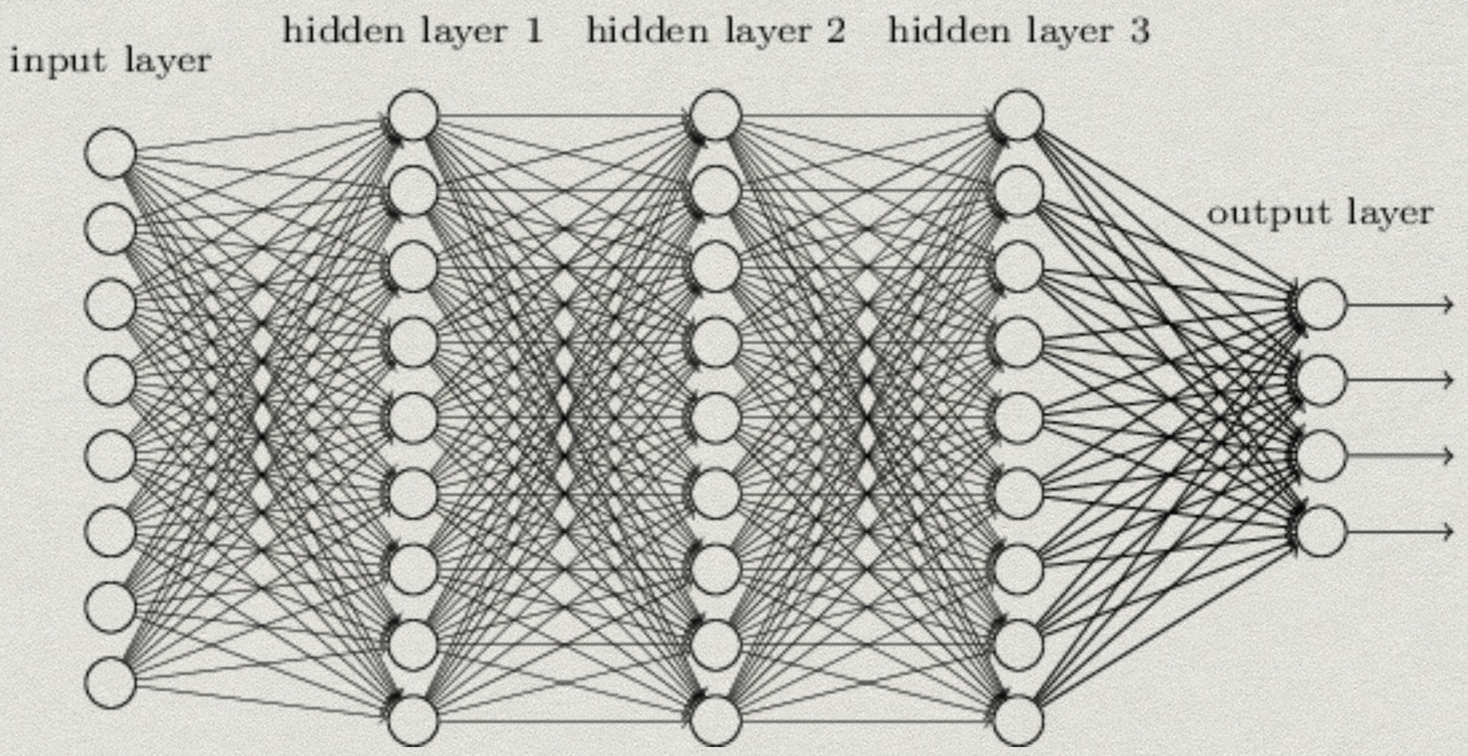
With enough neurons and depth, you can replicate any function!

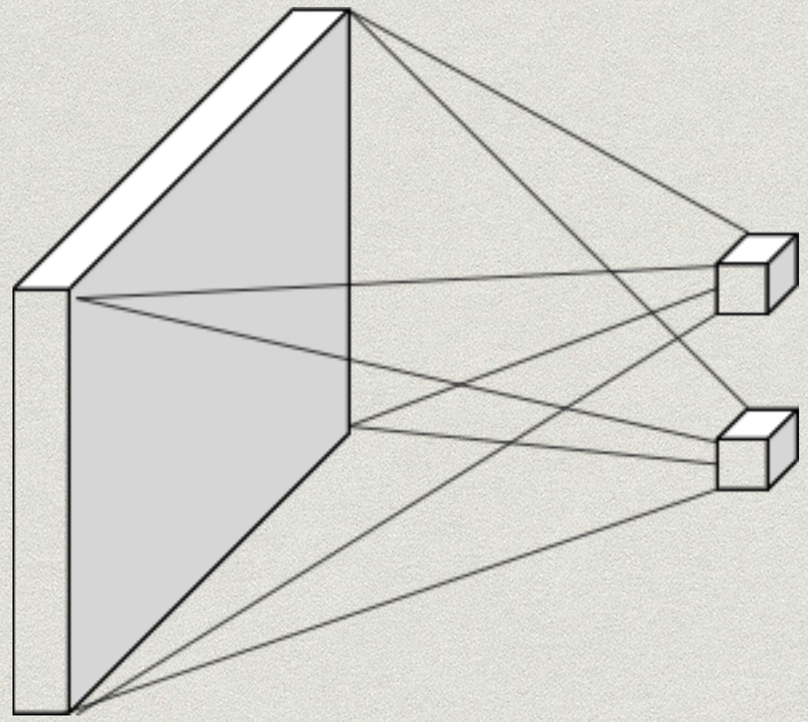


Yes. But, this becomes cumbersome fast…

Easy right?

With enough neurons and depth, you can replicate any function!



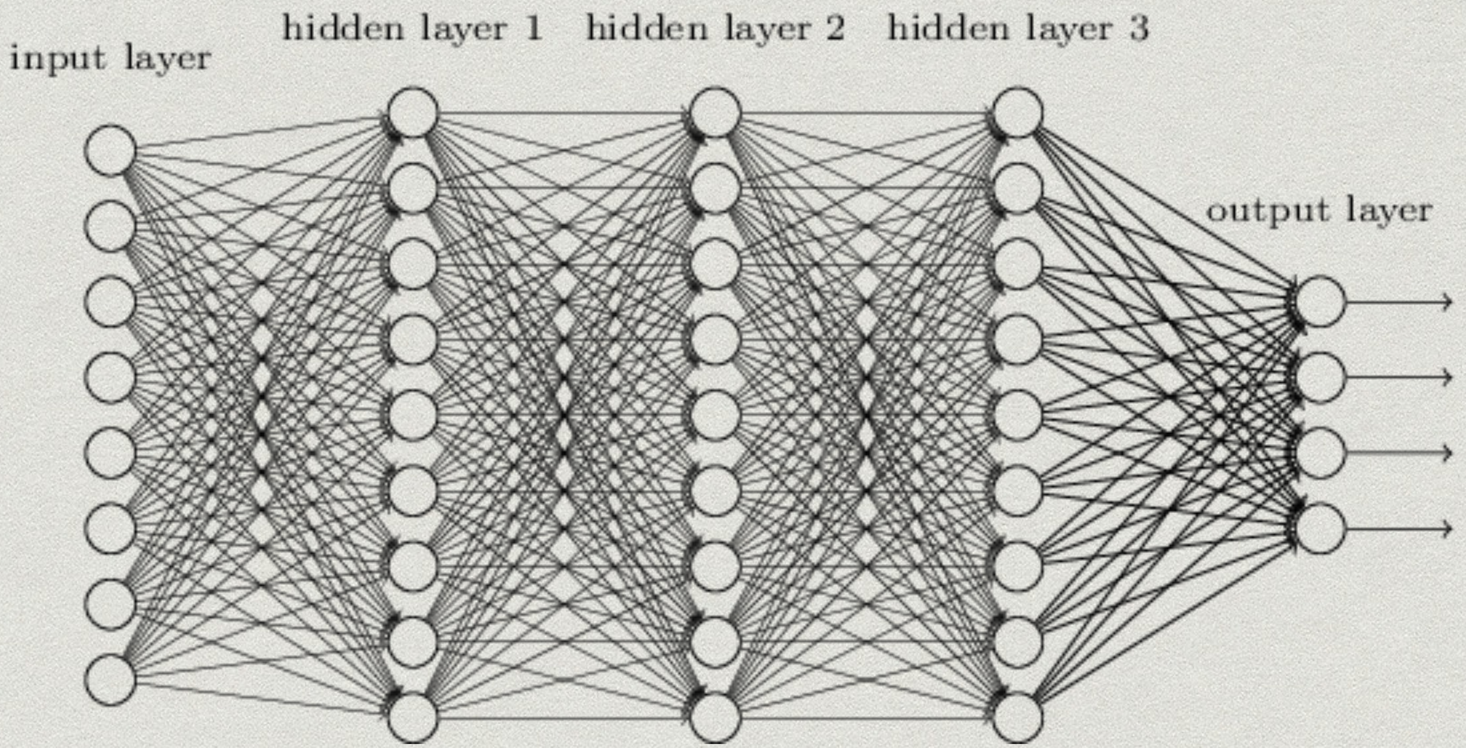


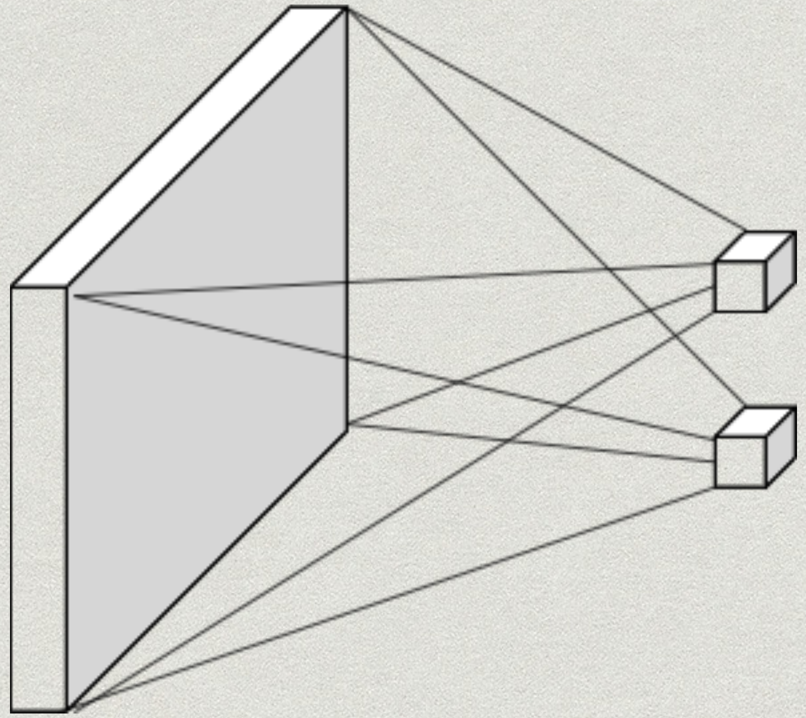
Yes. But, this becomes cumbersome fast…

Especially for big input images (e.g. 256x256 px)

Easy right?

With enough neurons and depth, you can replicate any function!



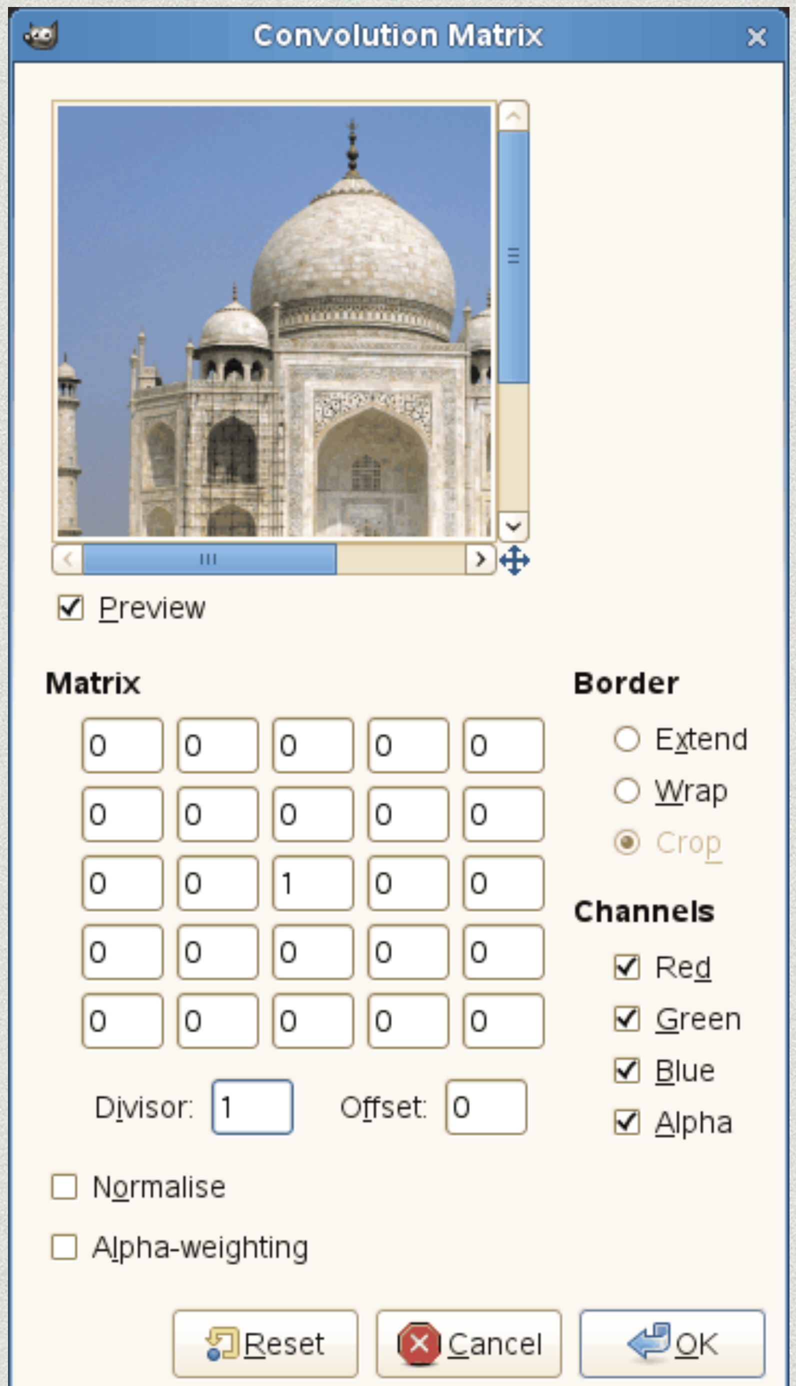


Yes. But, this becomes cumbersome fast…

Especially for big input images (e.g. 256x256 px)

To improve this approach, the neurons can be connected differently

Image filters (Gimp docs)



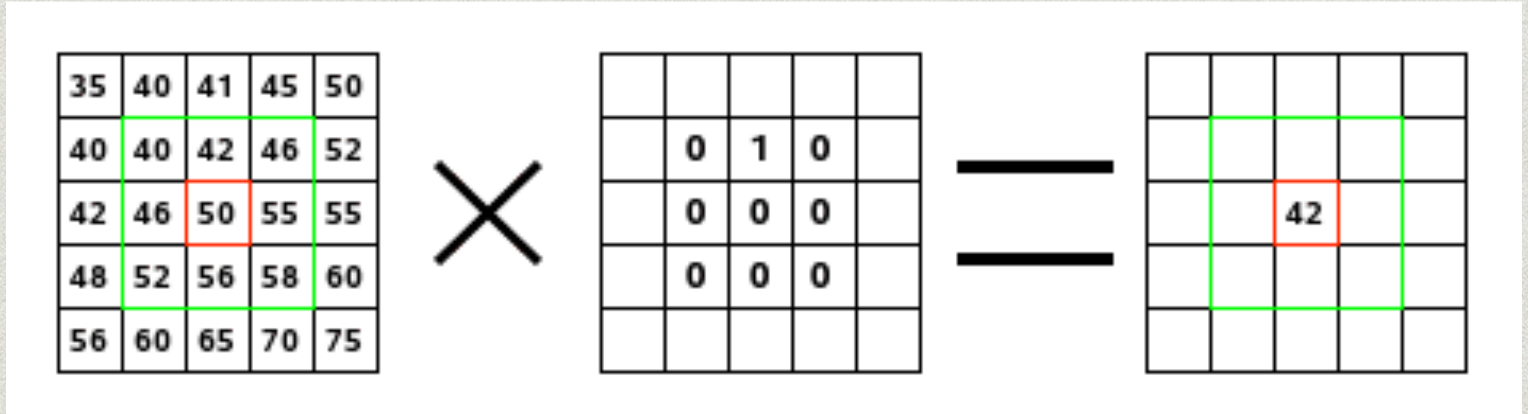
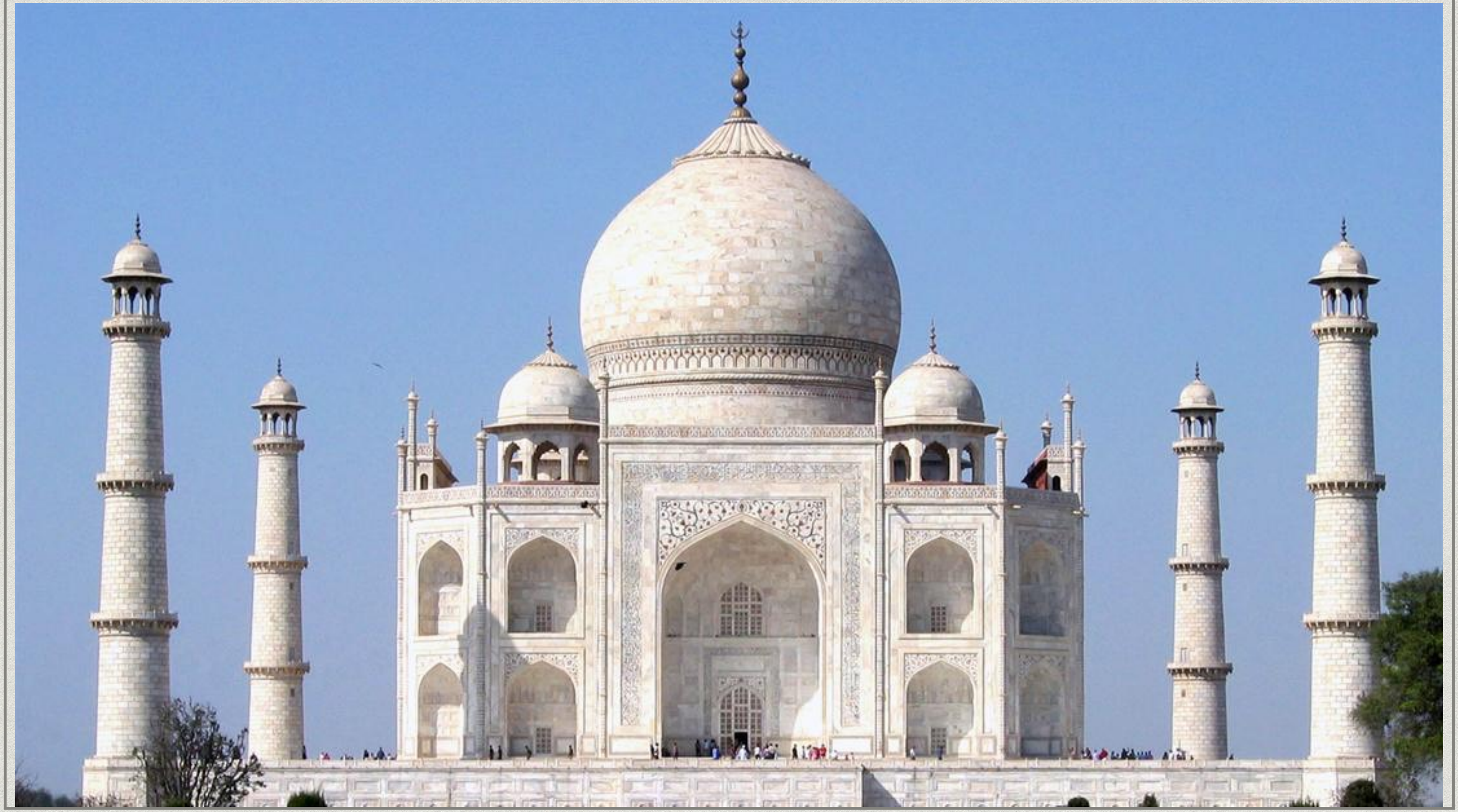


Image Kernel Result

This is in effect a convolution of the image by the filter

Base

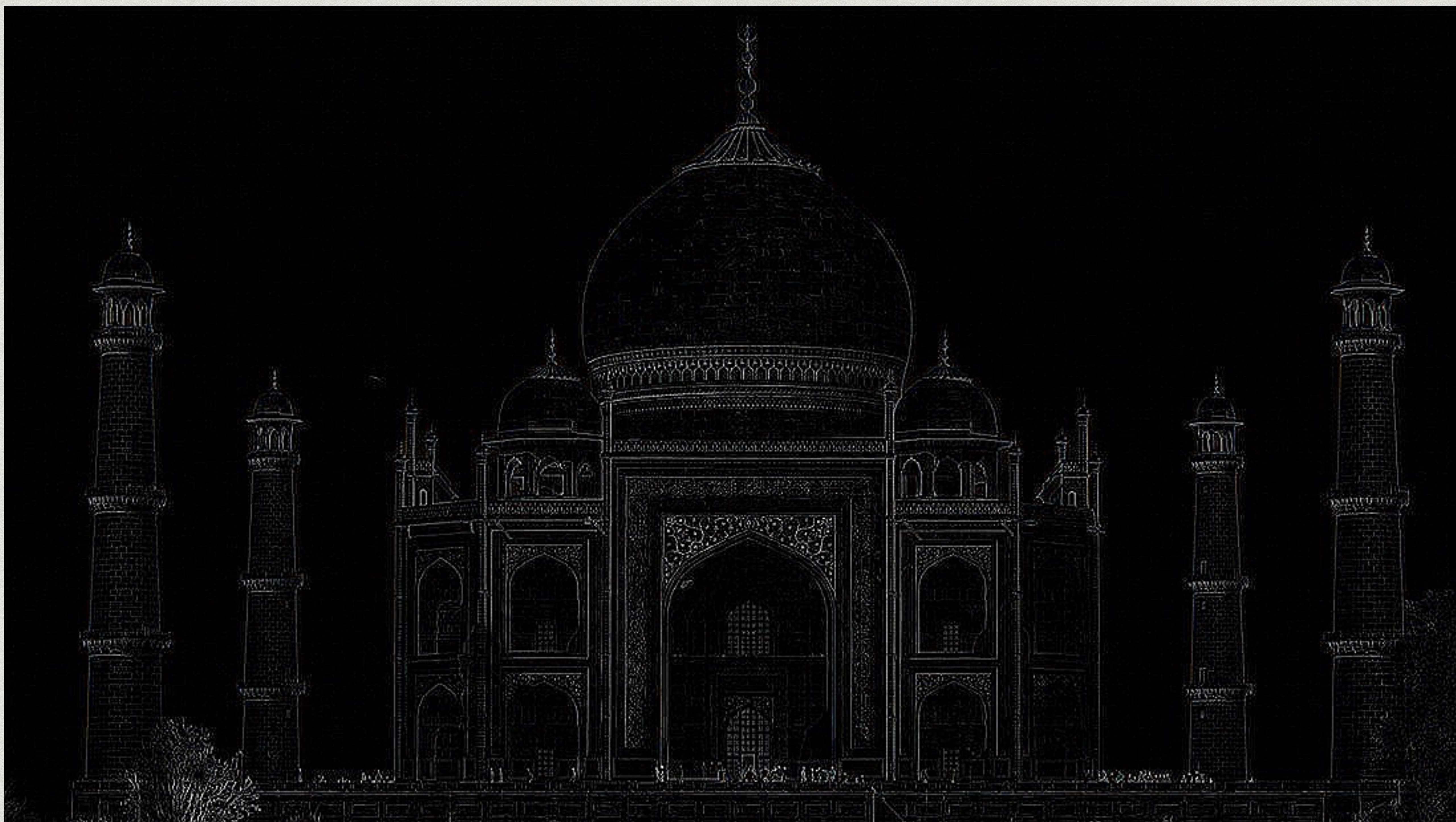


Edge detection

	0	1	0	
	1	-4	1	
	0	1	0	

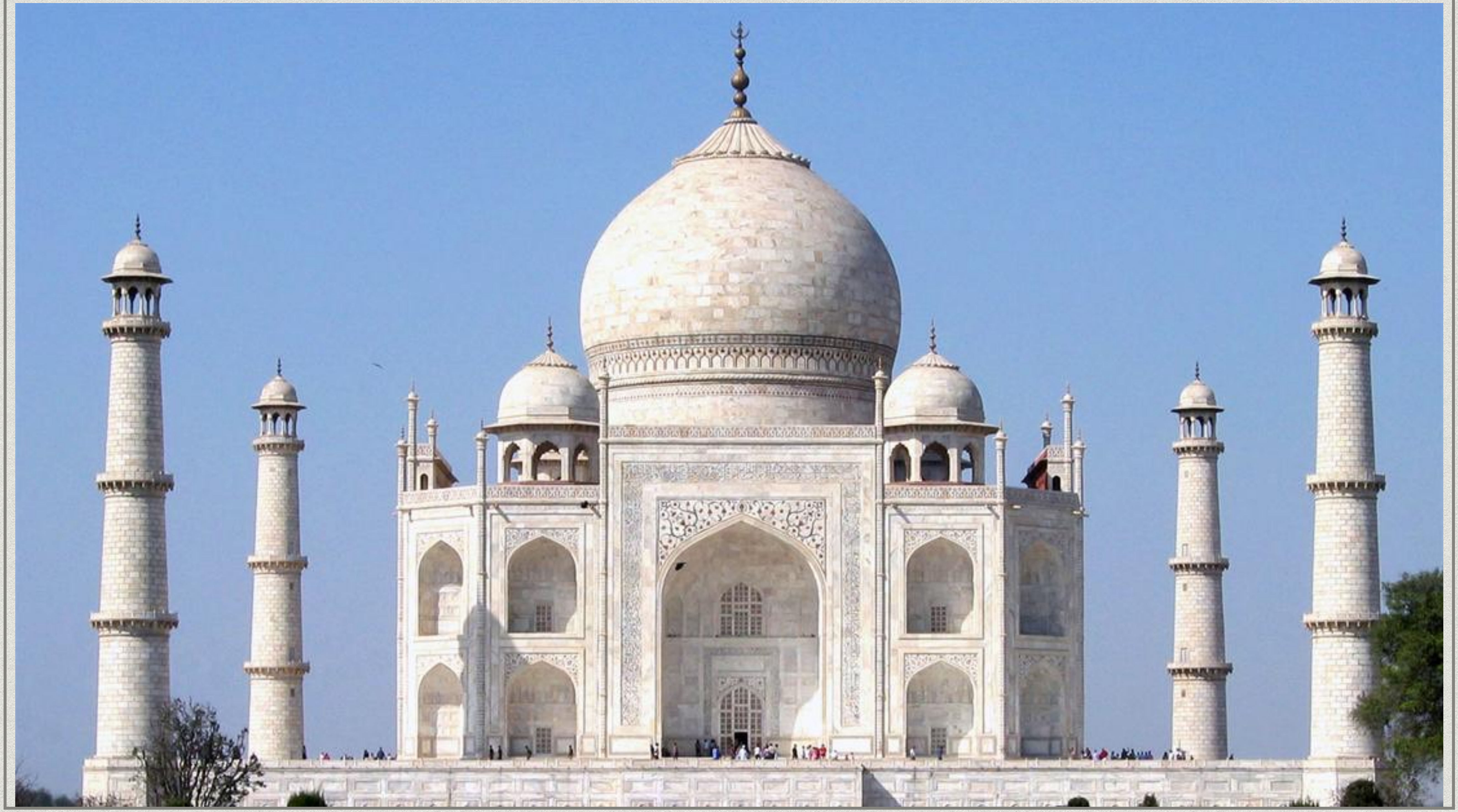
Edge detection

	0	1	0	
	1	-4	1	
	0	1	0	



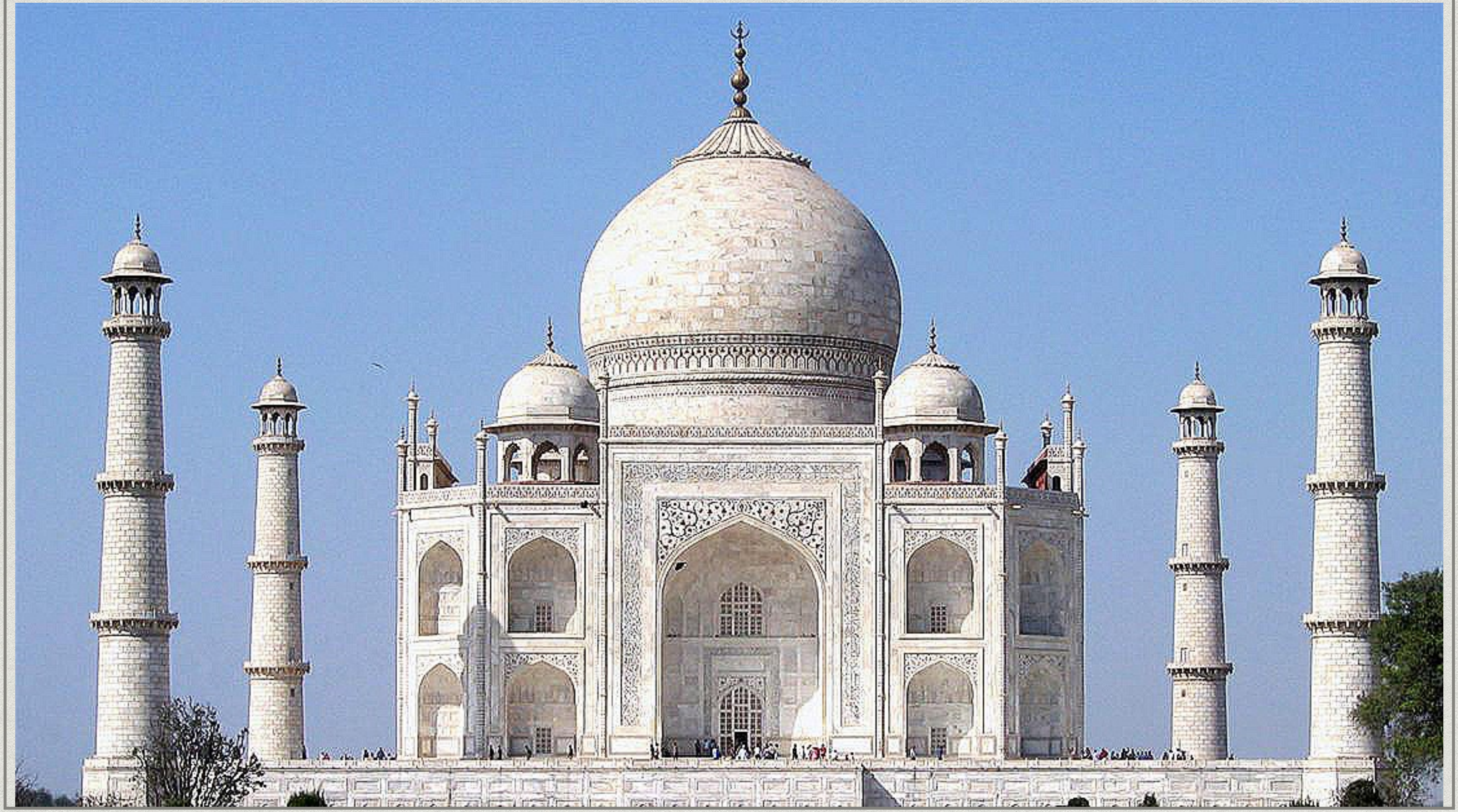
Sharpen

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

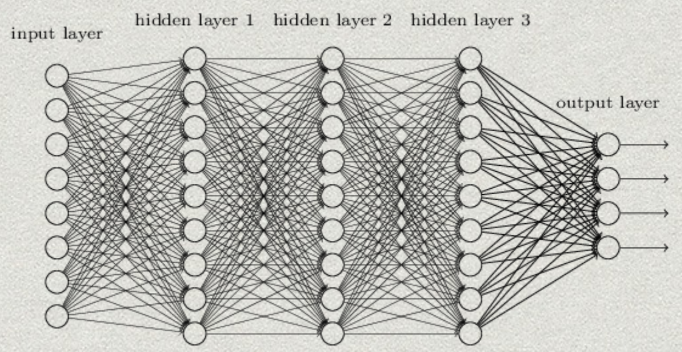


Sharpen

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

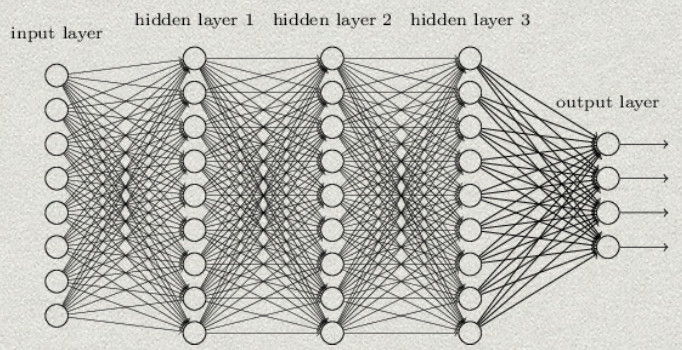


Building smarter layers



Fully Connected Layer

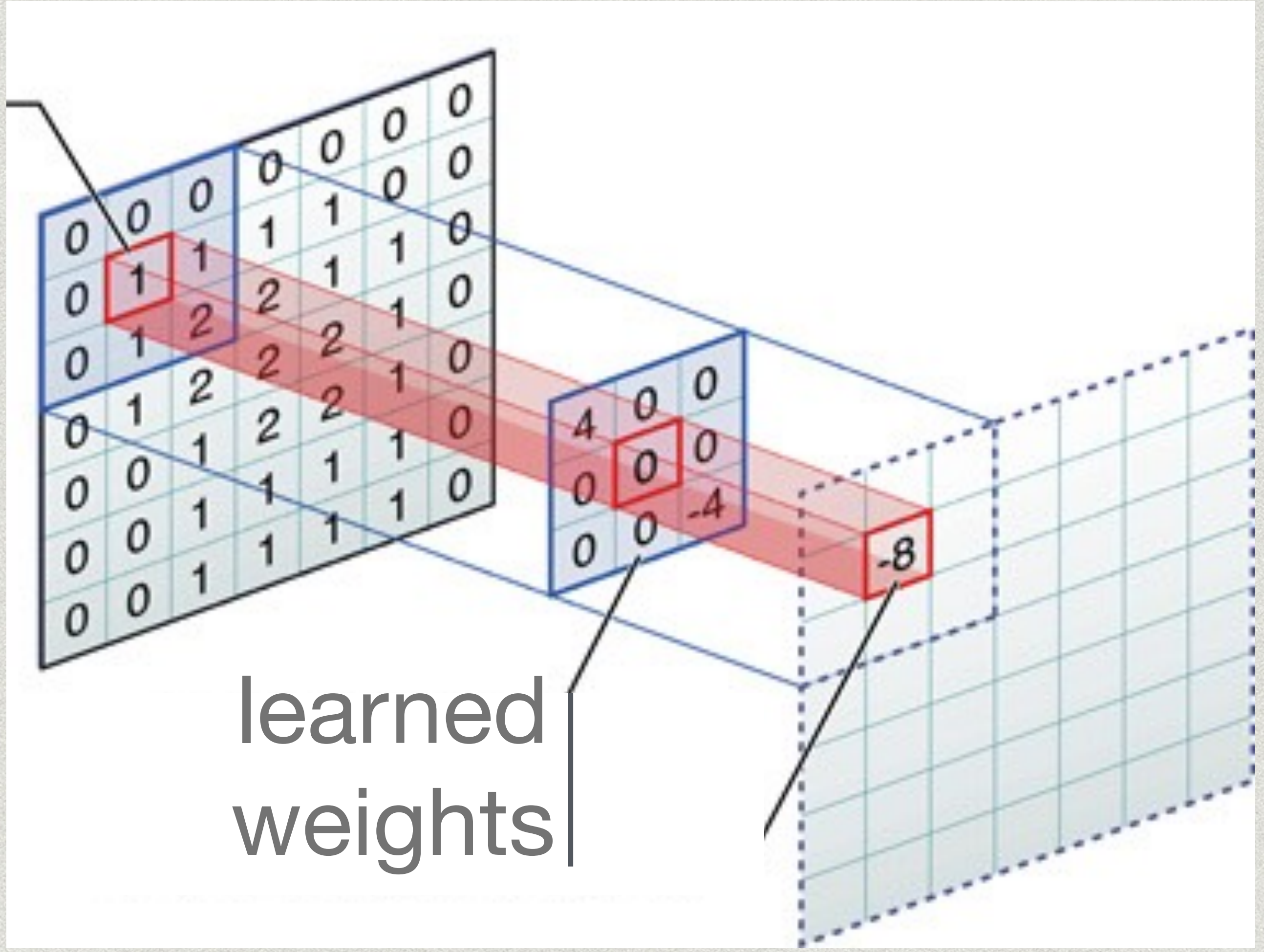
Building smarter layers



Fully Connected Layer

Shared weights using <u>convolution</u> :

You learn the kernel weights, then share over the full input



Deep MNIST

LeNet-5 (1998) : ~1% error

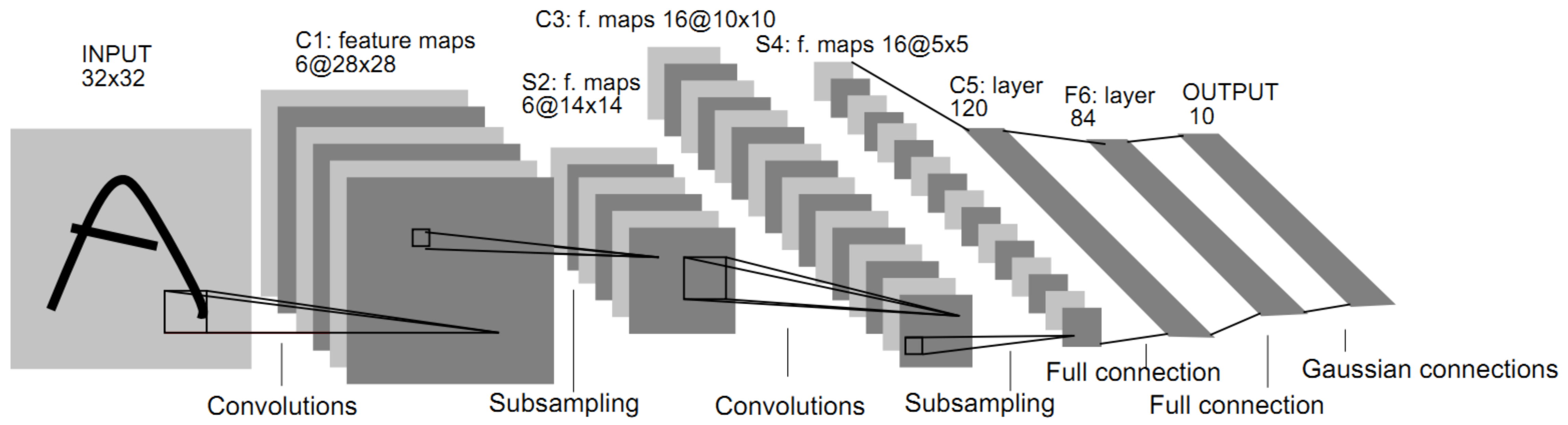


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998

Deep MNIST

LeNet-5 (1998) : ~1% error

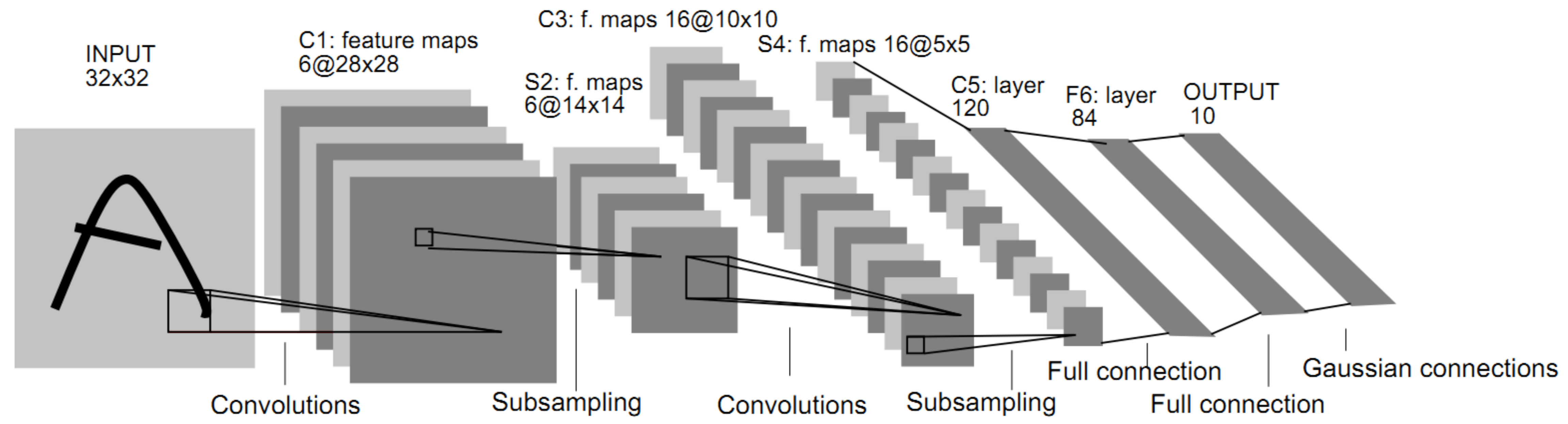


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998

Best result today: 0.21% error (less than humans!)

Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus Regularization of Neural Network using DropConnect, International Conference on Machine Learning 2013

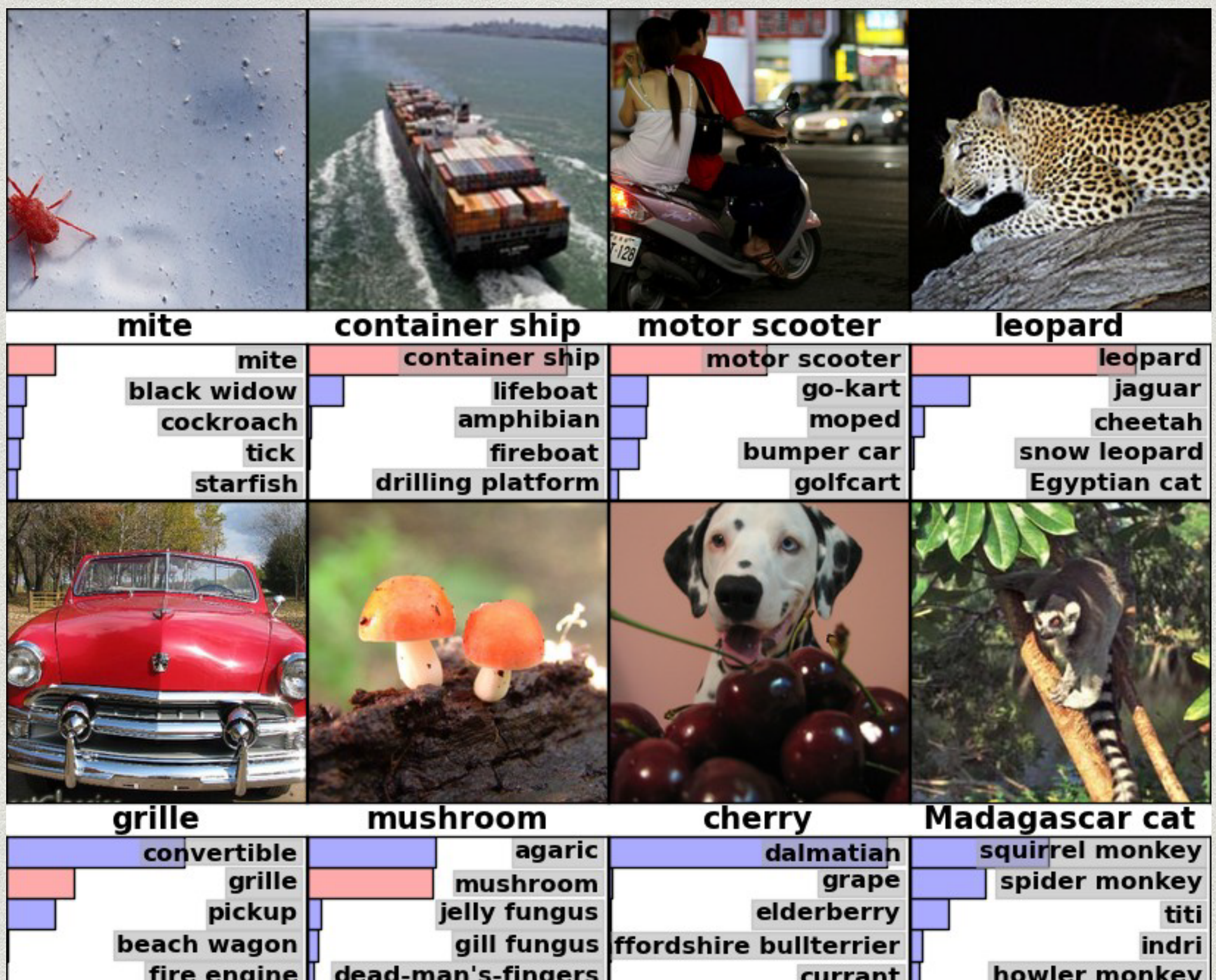
ImageNet

Popular AI challenge:

- **Crowdsourced labeling of image database (14 million labeled images)**
- **Competing algorithms try to classify them**

ImageNet classification challenge

ImageNet



Popular AI challenge:

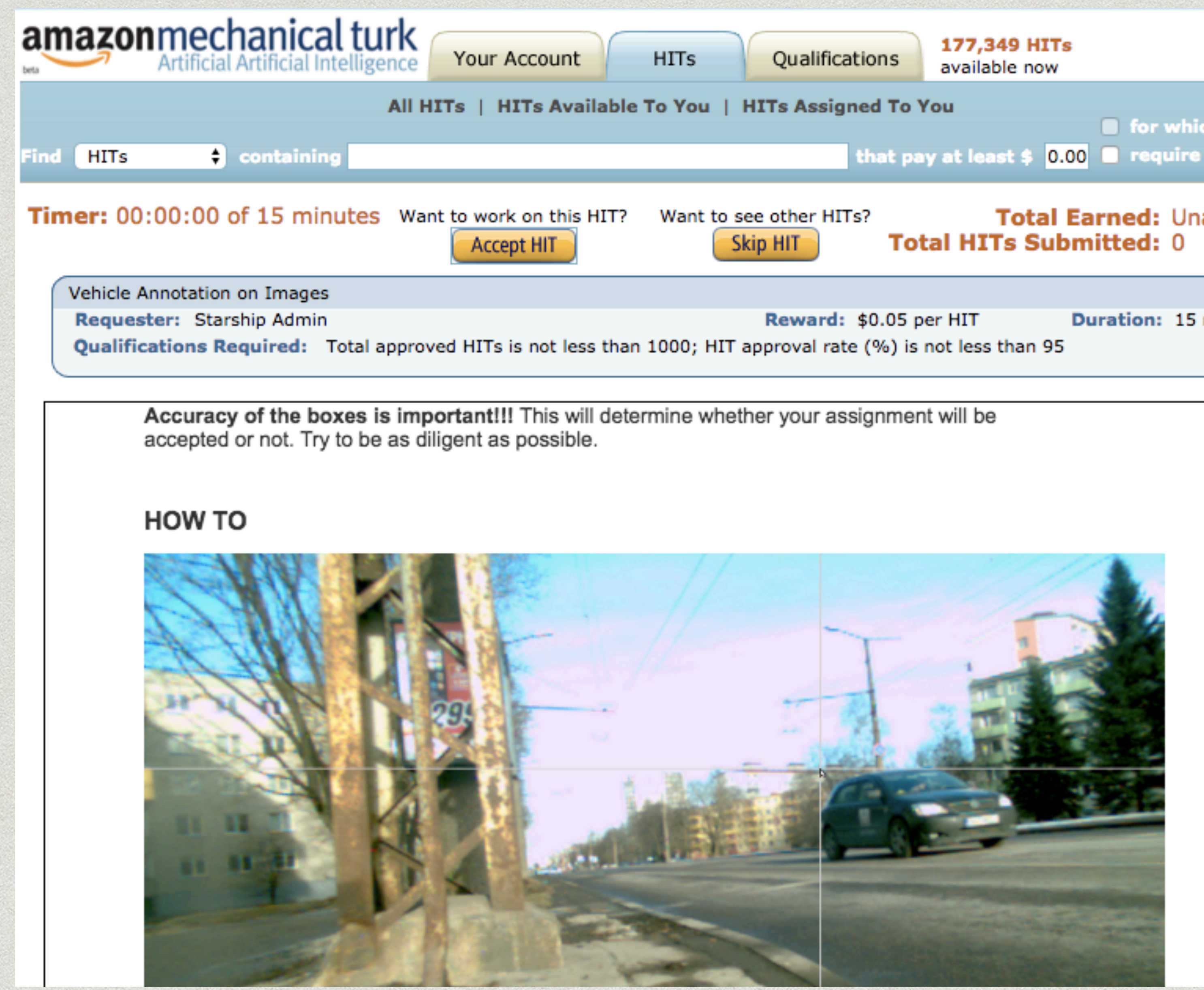
- **Crowdsourced labeling of image database (14 million labeled images)**
- **Competing algorithms try to classify them**

Wait, 14 million??

Amazon calls it « Human Intelligence Tasks »

€

¥ $



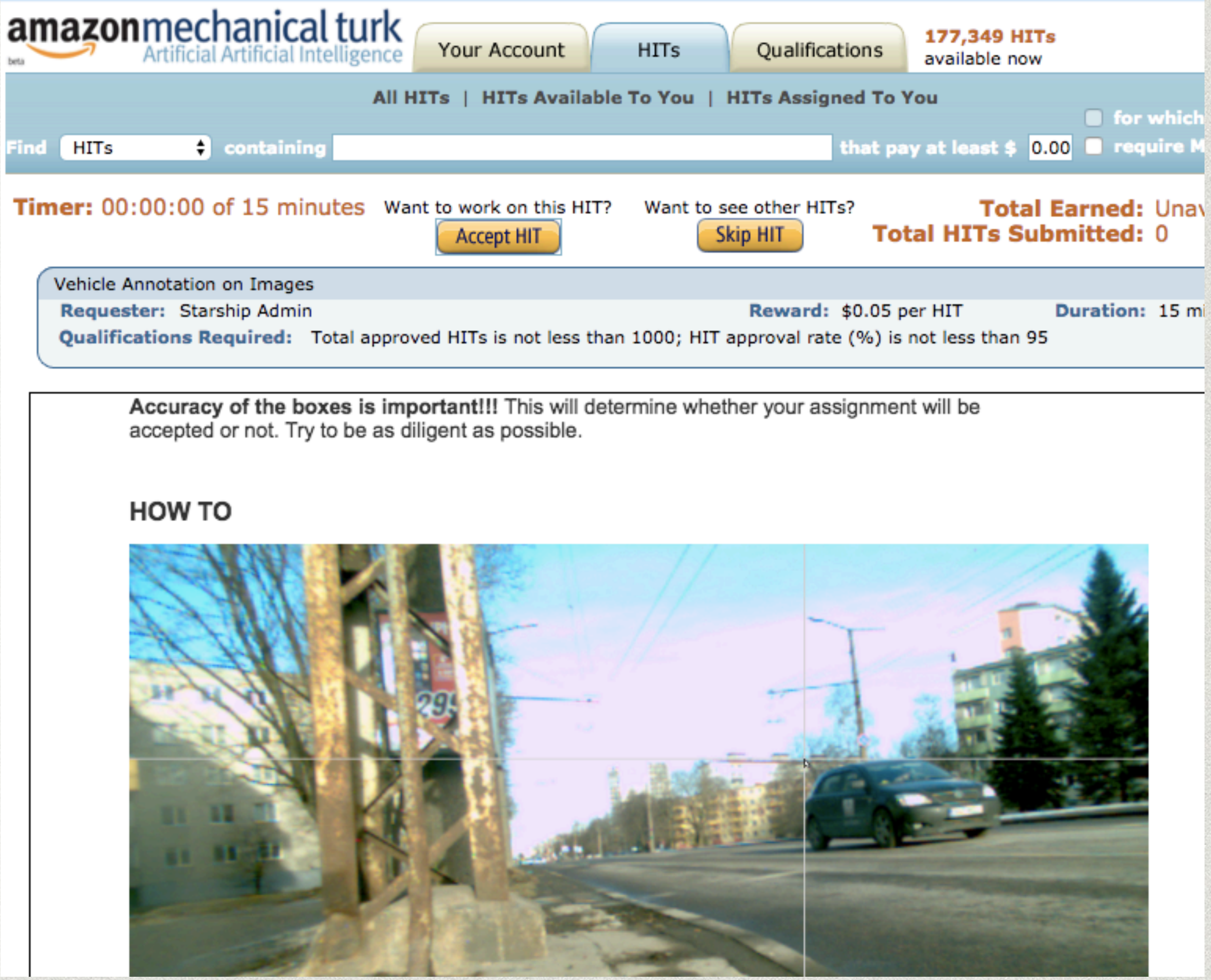
Wait, 14 million??

Amazon calls it « Human Intelligence Tasks »

€

¥ $

It's probably pretty boring though…

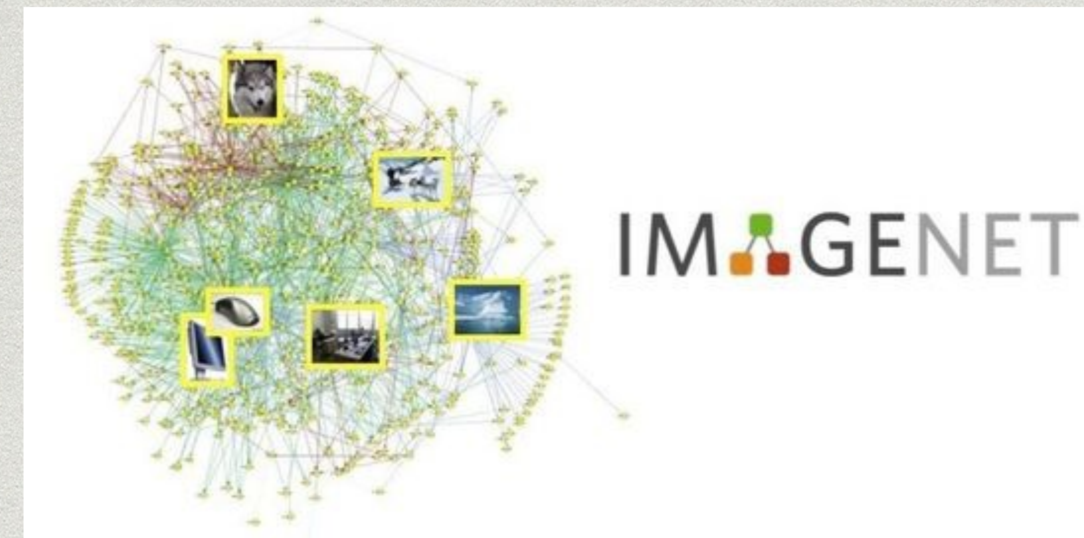


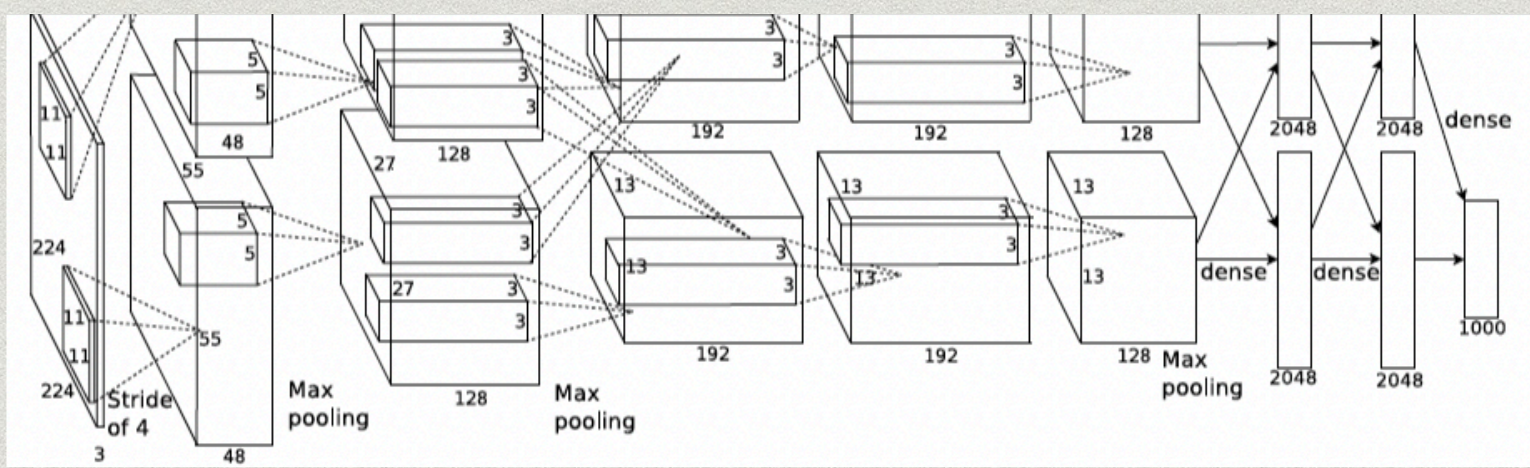
ImageNet

- Images are **Big Data** compared to MNIST

ImageNet

- Images are **Big Data** compared to MNIST

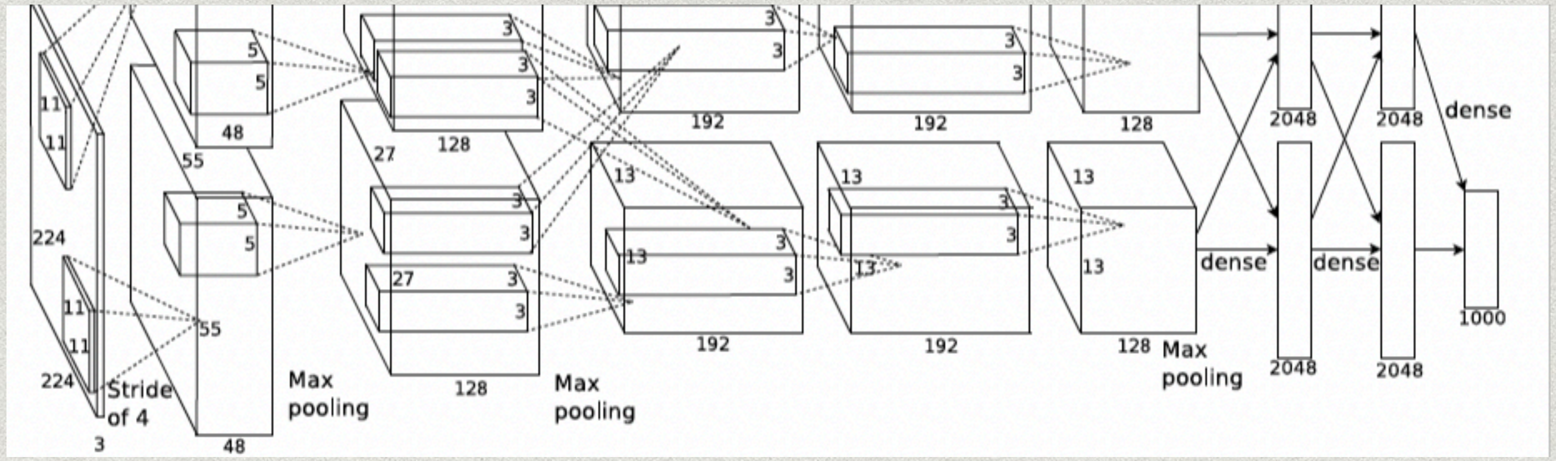




AlexNet: ImageNet 2012 winner

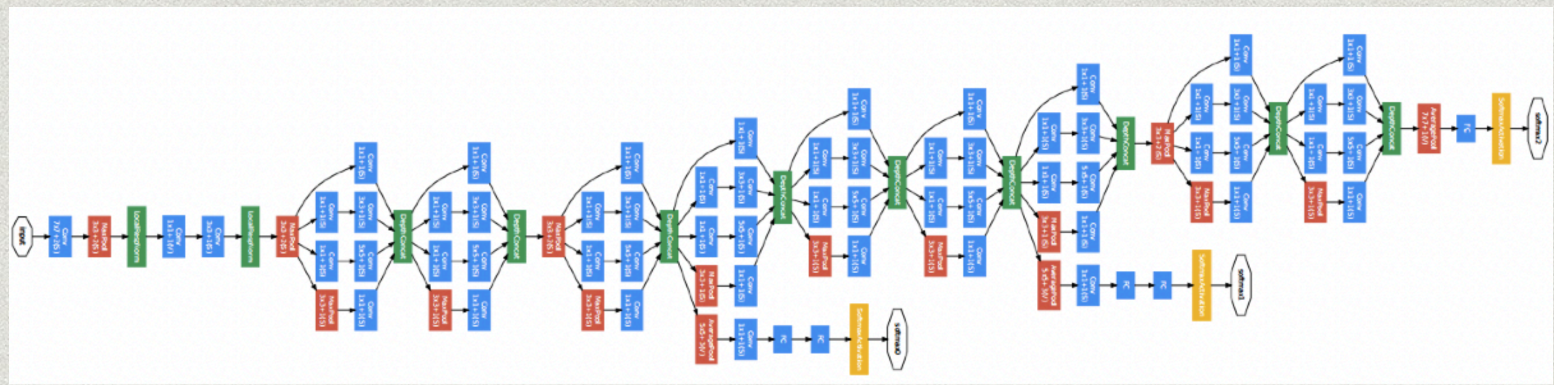
ImageNet

- Images are **Big Data** compared to MNIST

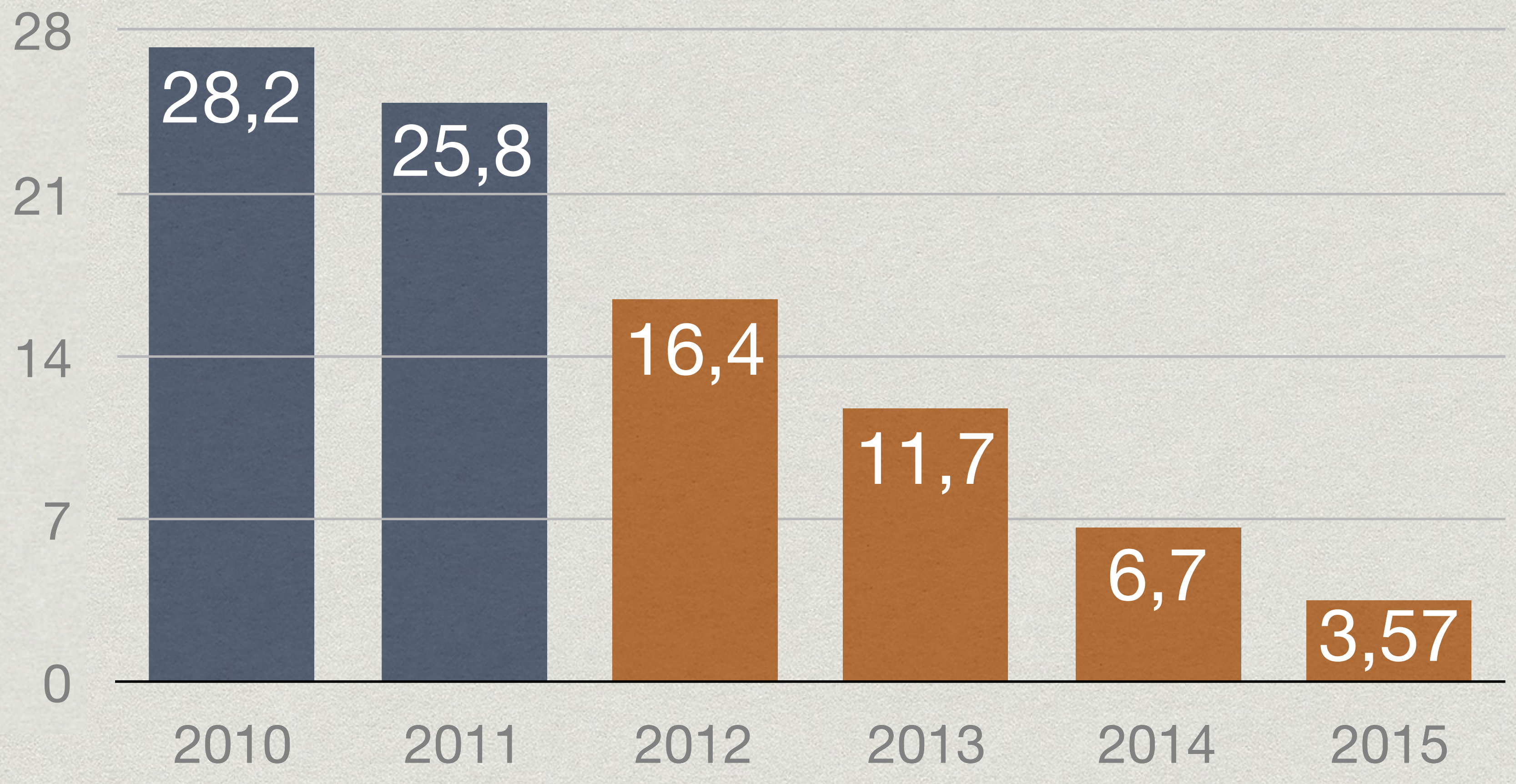


AlexNet: ImageNet 2012 winner

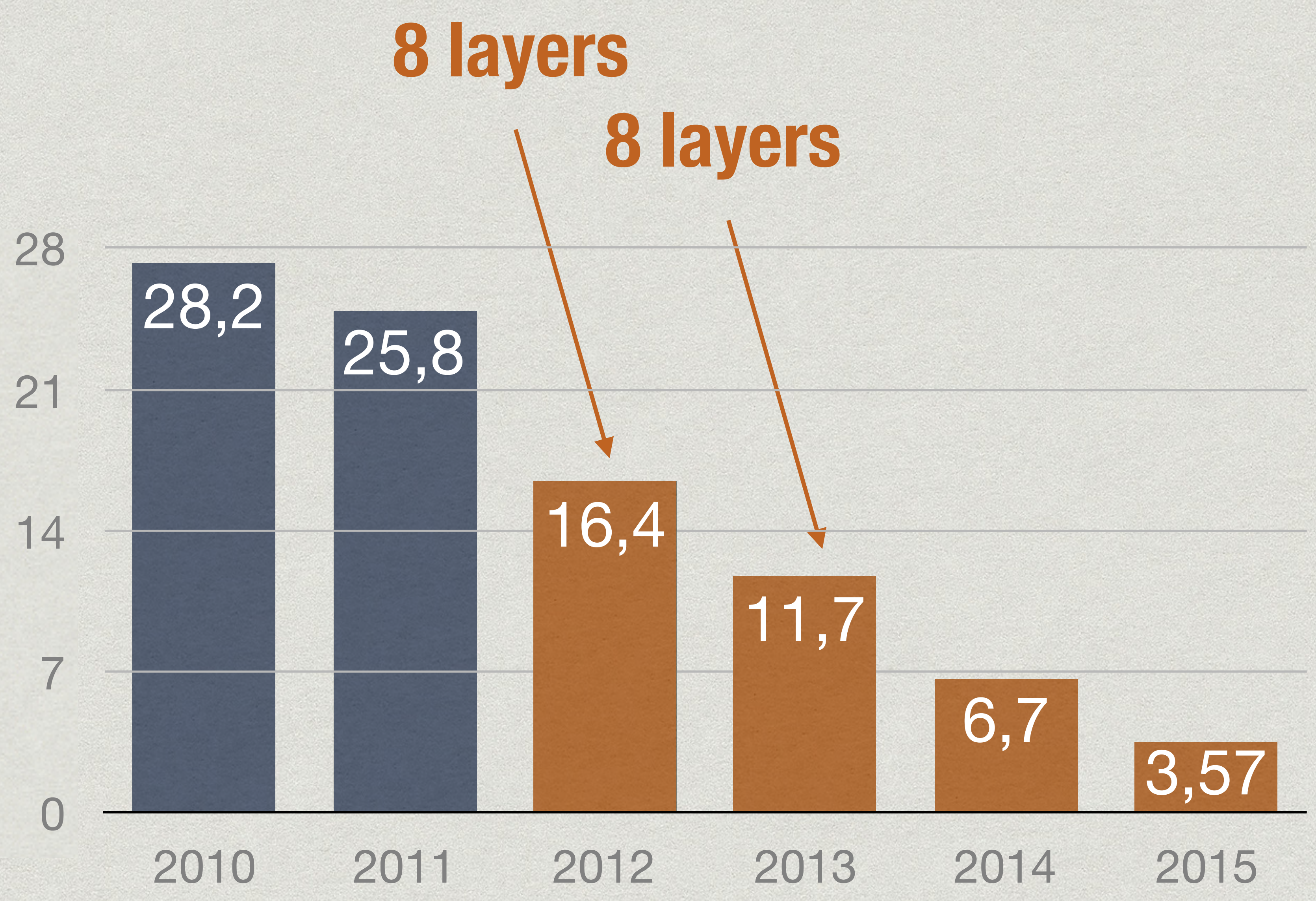
GoogLeNet: ImageNet 2014 winner



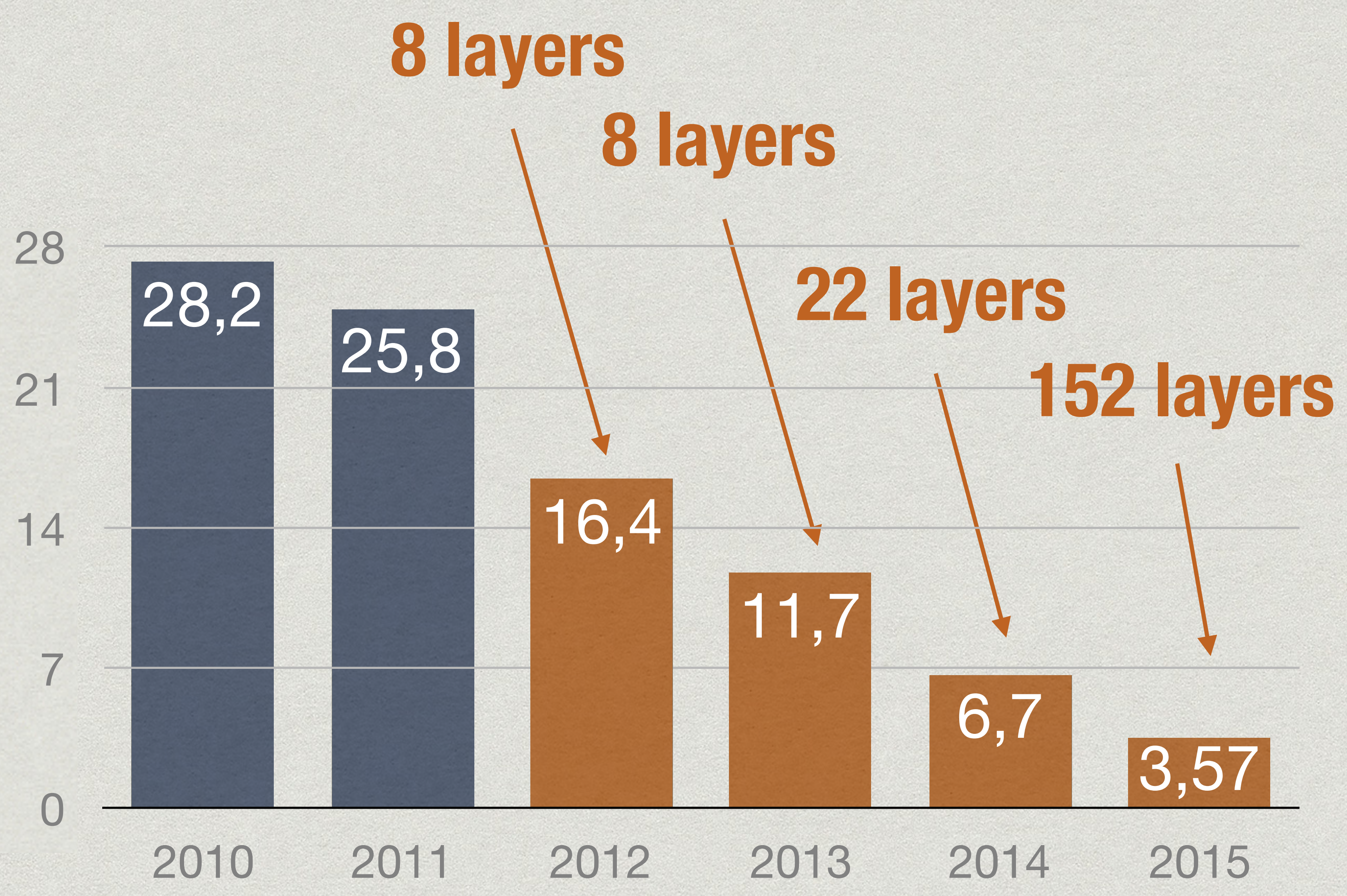
Deeper and deeper…



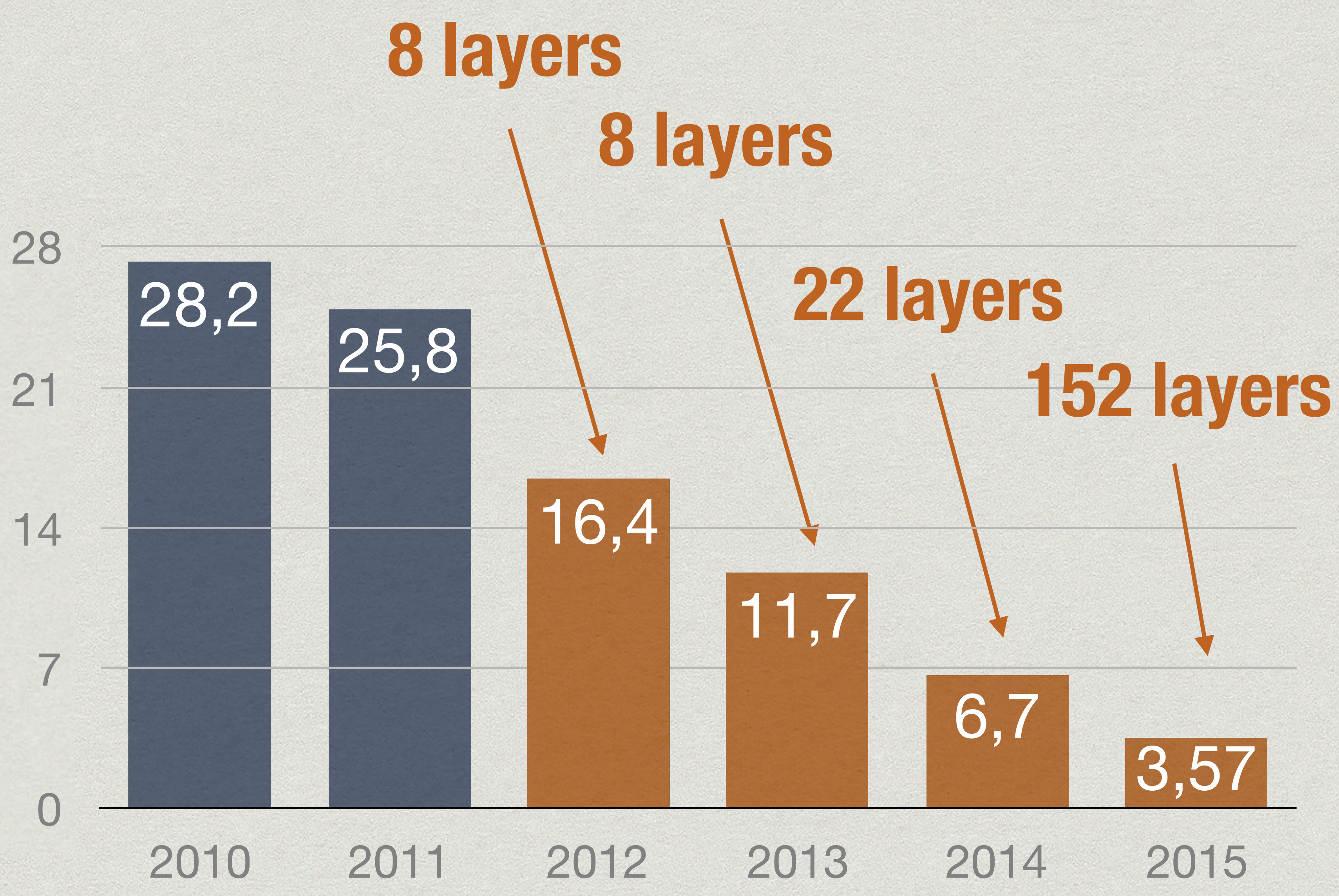
Deeper and deeper…



Deeper and deeper…



Deeper and deeper



How do I use this?

- Do **not** expect to make sense of the function.
  - GoogleNet (22 layers) = 11,193,984 parameters
  - ResNet (153 layers) = 25,636,712 parameters
- Deep neural **classifiers** are high performers
- But « artificial intelligence » is not just about classification! Can it do anything else?

ARTIFICIAL « INTELLIGENCE »

« Most of human and animal learning is unsupervised learning »

-Yann LeCun

« Most of human and animal learning is unsupervised learning »

-Yann LeCun

Big deal :

« Most of human and animal learning is unsupervised learning »

-Yann LeCun

Big deal :

- **Founding Director of the NYU Center for Data Science**

« Most of human and animal learning is unsupervised learning »

-Yann LeCun

Big deal :

- **Founding Director of the NYU Center for Data Science**
- **Director of AI research at Facebook**

3 types of learning

3 types of learning

Reinforcement learning

3 types of learning

Reinforcement learning

Supervised learning

3 types of learning

Reinforcement learning

Supervised learning

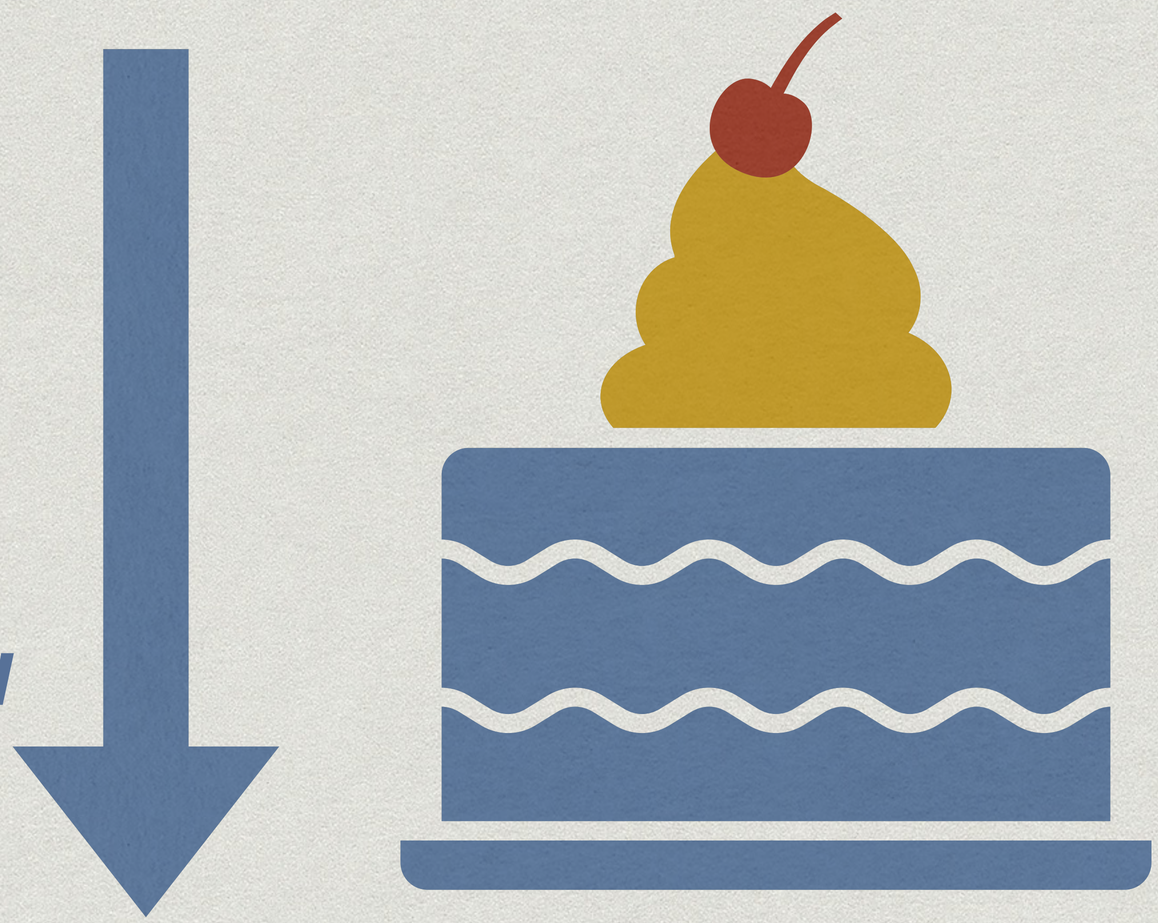
Unsupervised learning

3 types of learning

Today — Reinforcement learning

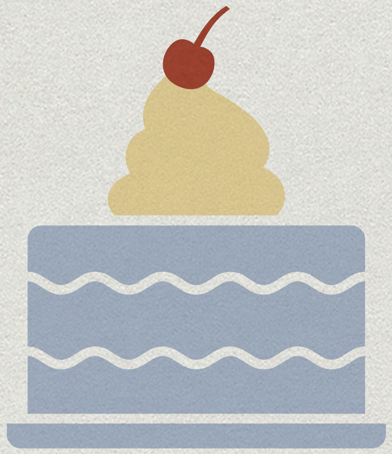
Today — Supervised learning

Tomorrow — Unsupervised learning



Time

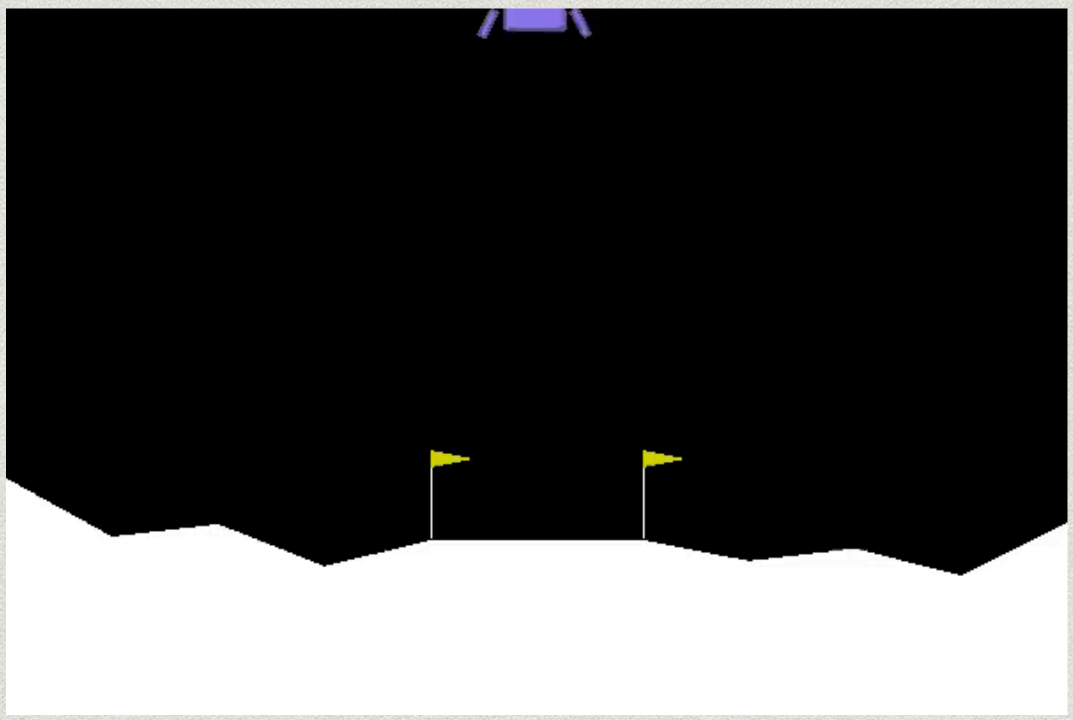
Reinforcement learning



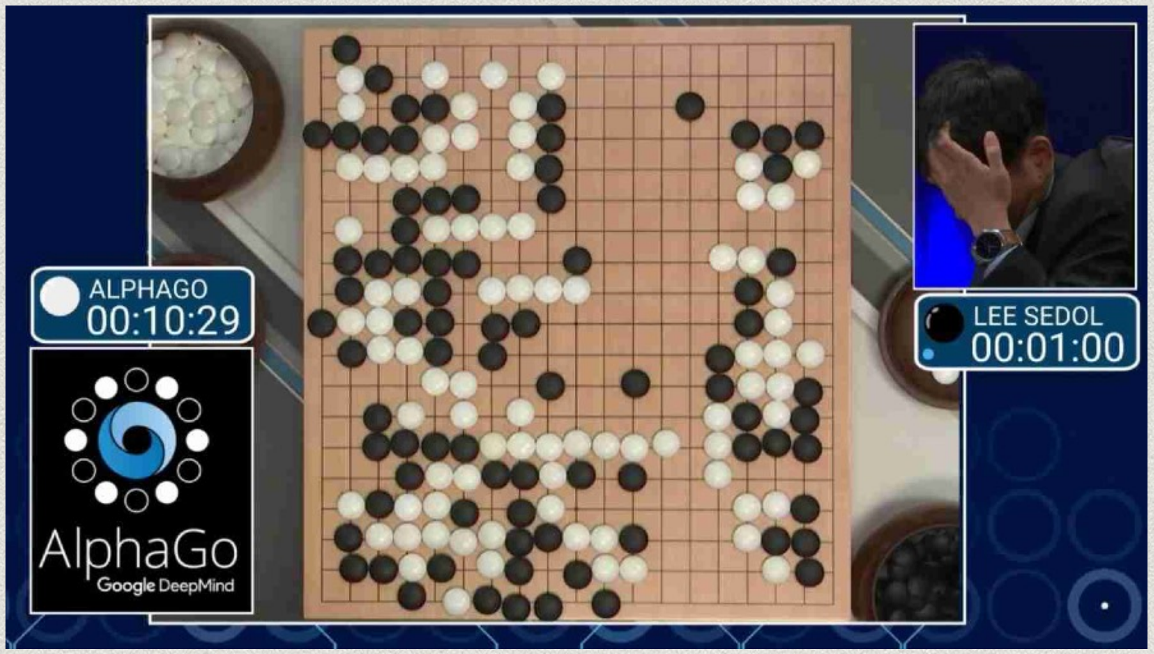
The cherry!

* Input: a game. Output: the score. No limit to the number of playthroughs

OpenAI Gym



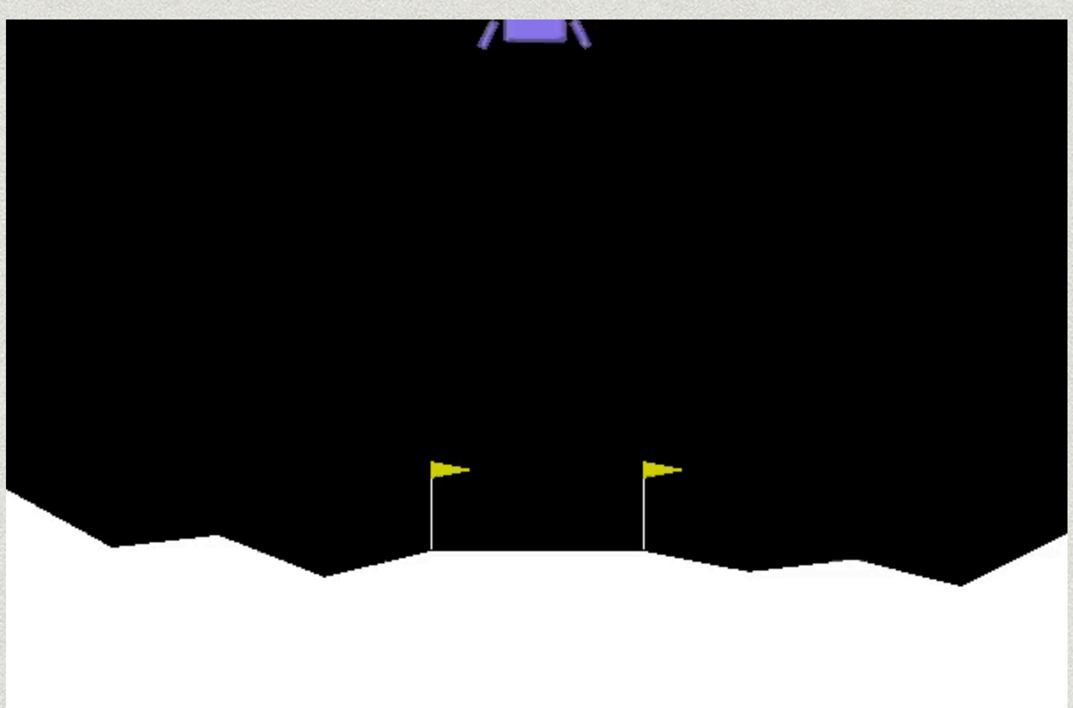
AlphaGo

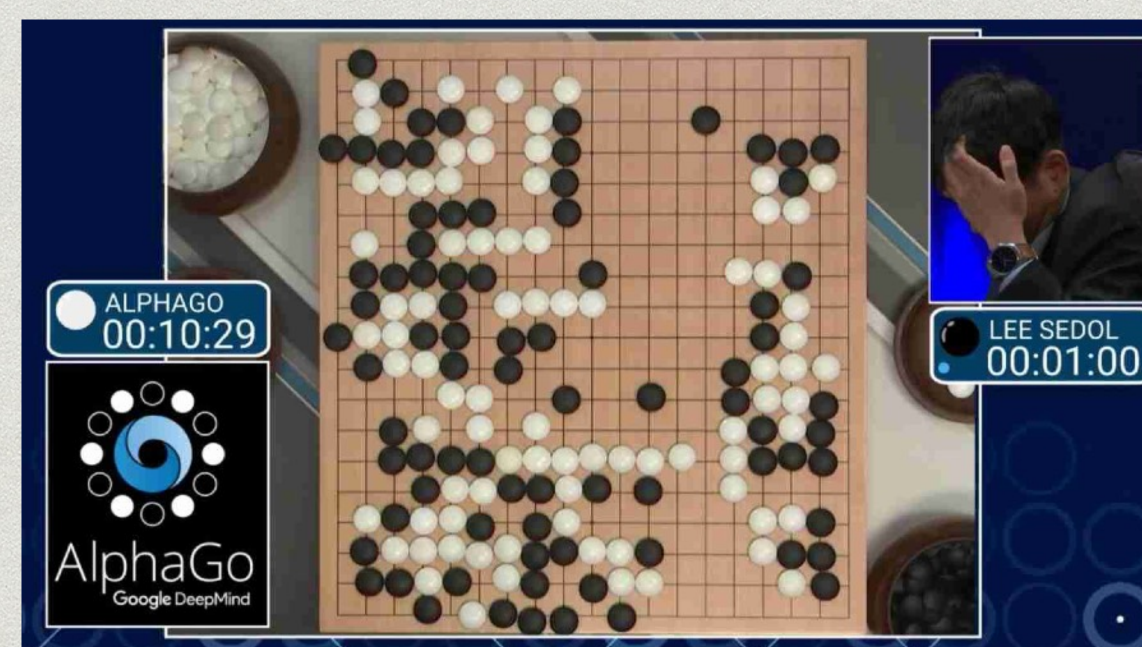


Reinforcement learning

The cherry!

* Input: a game. Output: the score. No limit to the number of playthroughs





Supervised learning

The icing!

- Learn to predict from a finite set of labeled data
- Example: classifiers IMAGENET
- Great, but needs a lot of data…

Unsupervised learning

The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

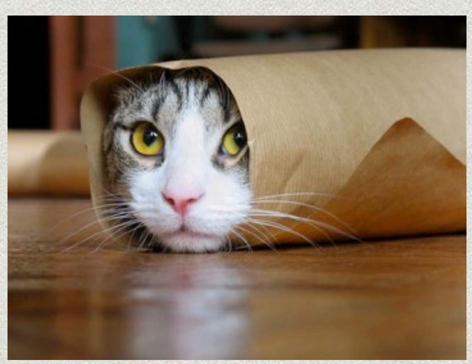
Unsupervised learning

The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

Supervised learning

cat!



cat!

cat!

Unsupervised learning

The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

Supervised learning

cat!

cat!

cat!

cat?

Unsupervised learning

The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

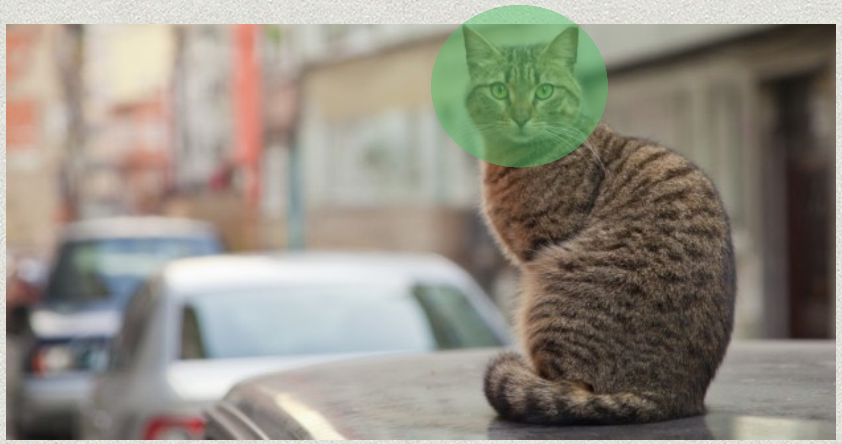
Unsupervised learning

Unsupervised learning

The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

Unsupervised learning

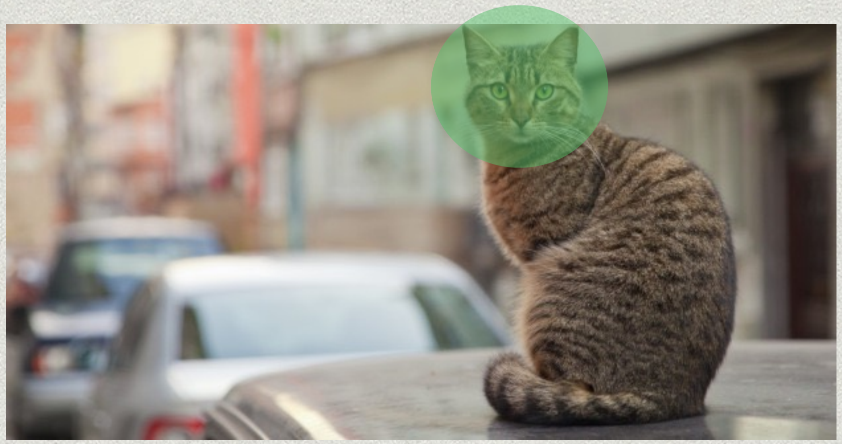


Unsupervised learning

The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

Unsupervised learning

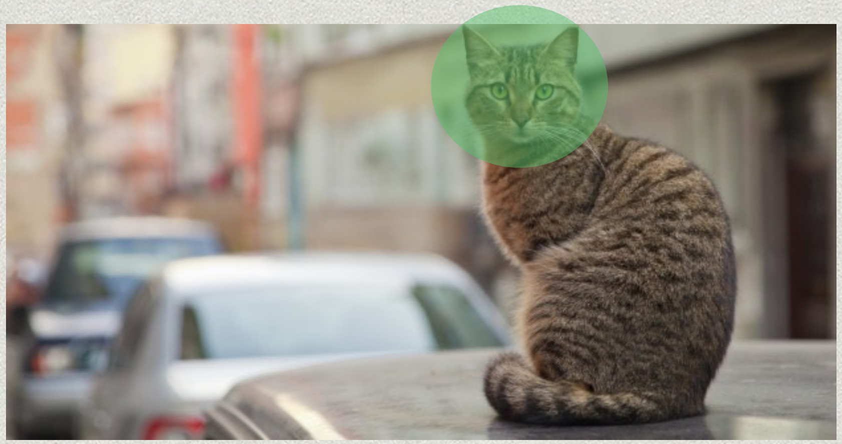


Unsupervised learning

The cake!

* To teach a child to recognize a cat, he doesn’t need to see 1 million cats

Unsupervised learning



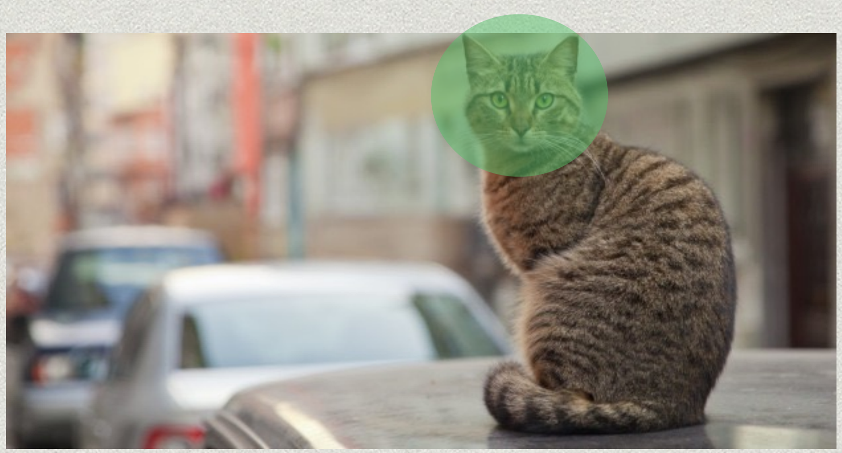
Do you find « similar » elements?

Unsupervised learning

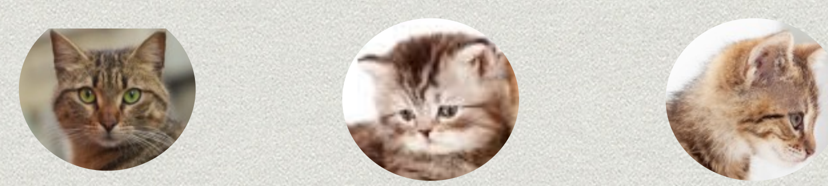
The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

Unsupervised learning



Do you find « similar » elements?

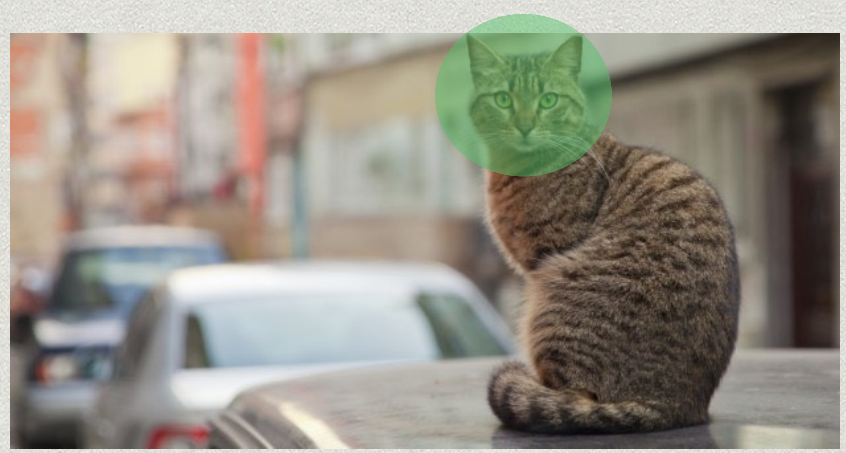


Unsupervised learning

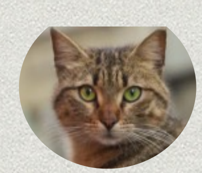
The cake!

- To teach a child to recognize a cat, he doesn't need to see 1 million cats

Unsupervised learning



Do you find « similar » elements?

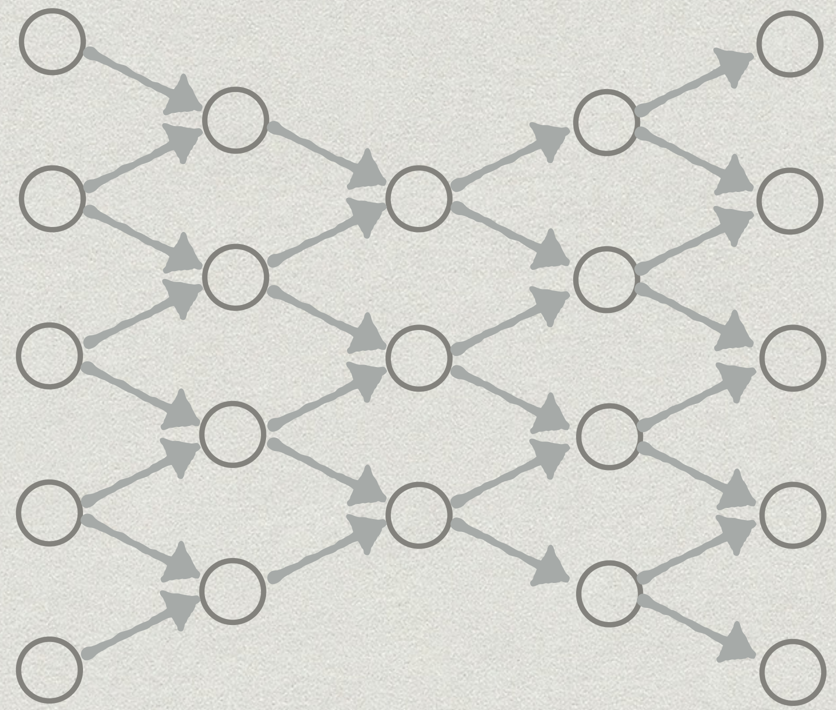


Good! That's called a cat

« True » AI is unsupervised

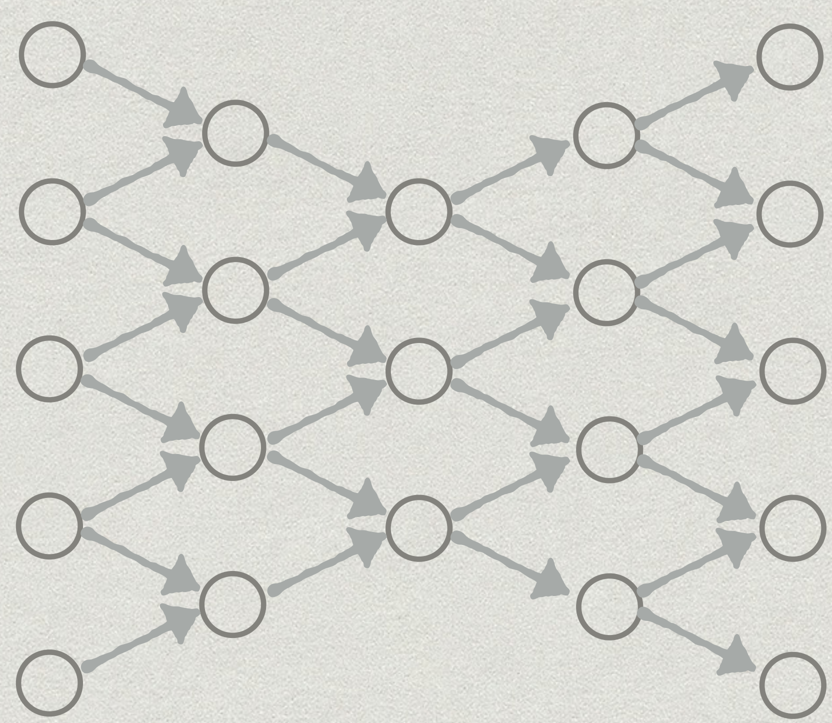
- « Intelligent » beings *infer* relations / classes
- They gather unlabeled information at all times
- Then learn the names for them quickly

Unsupervised example: the autoencoder

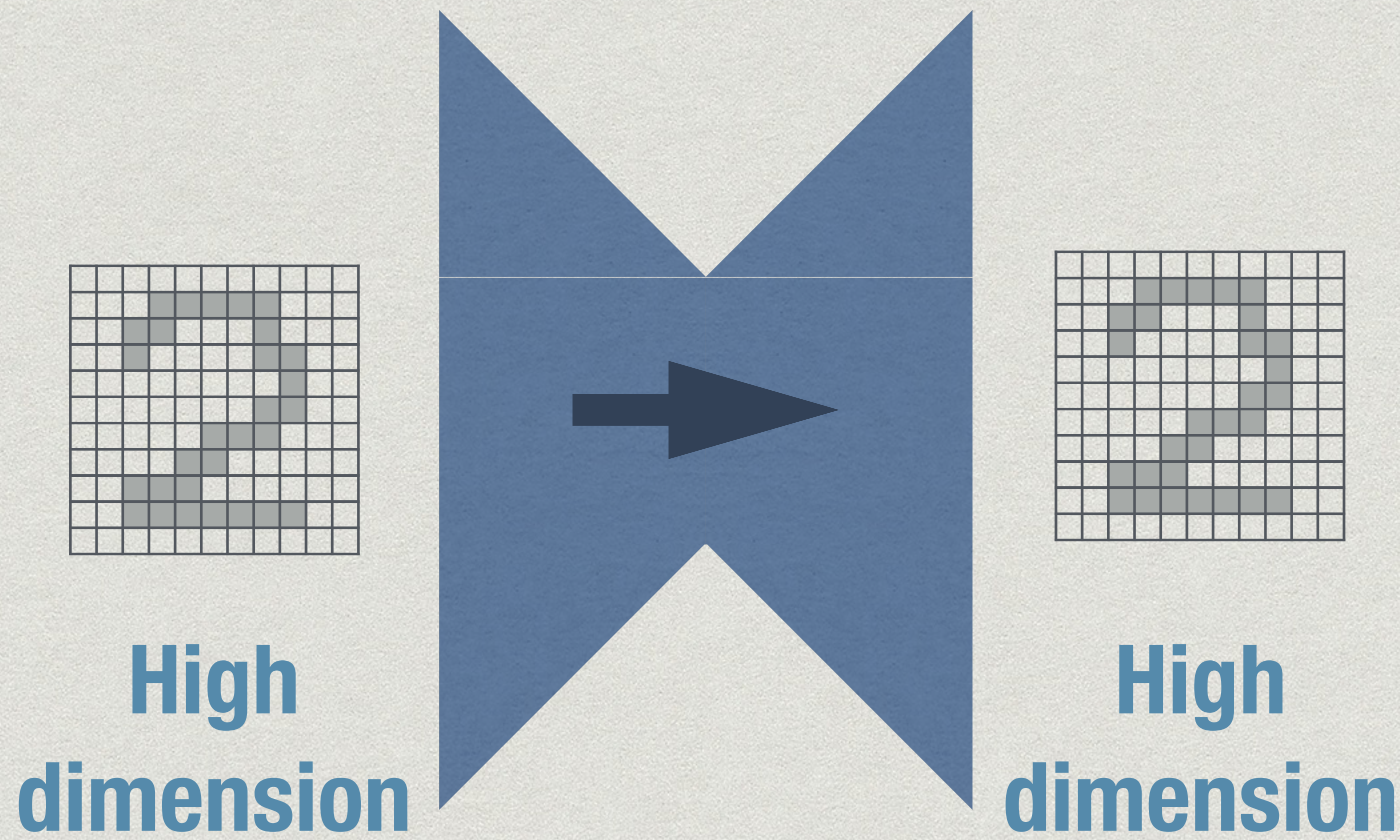


Autoencoder

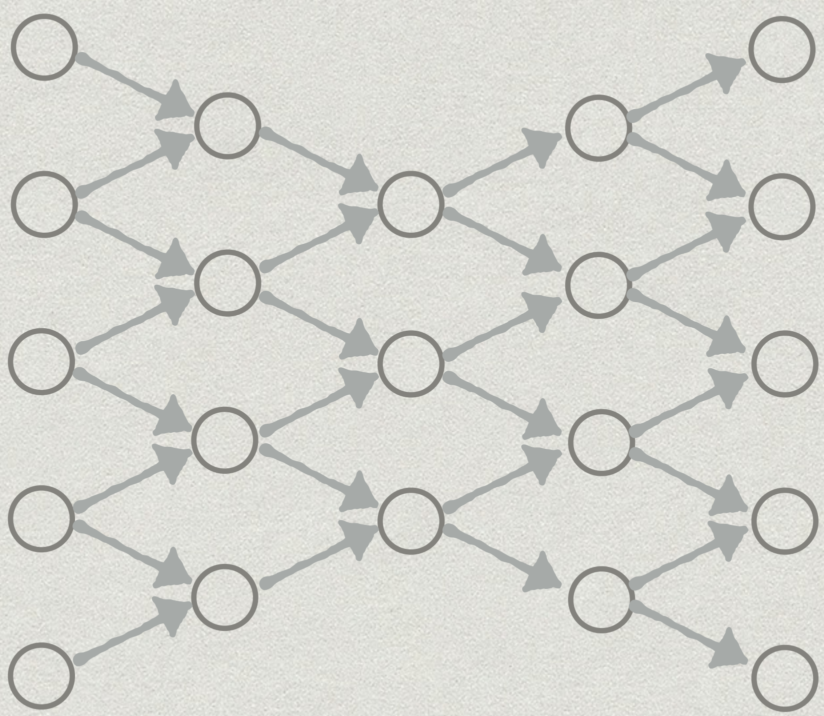
Unsupervised example: the autoencoder



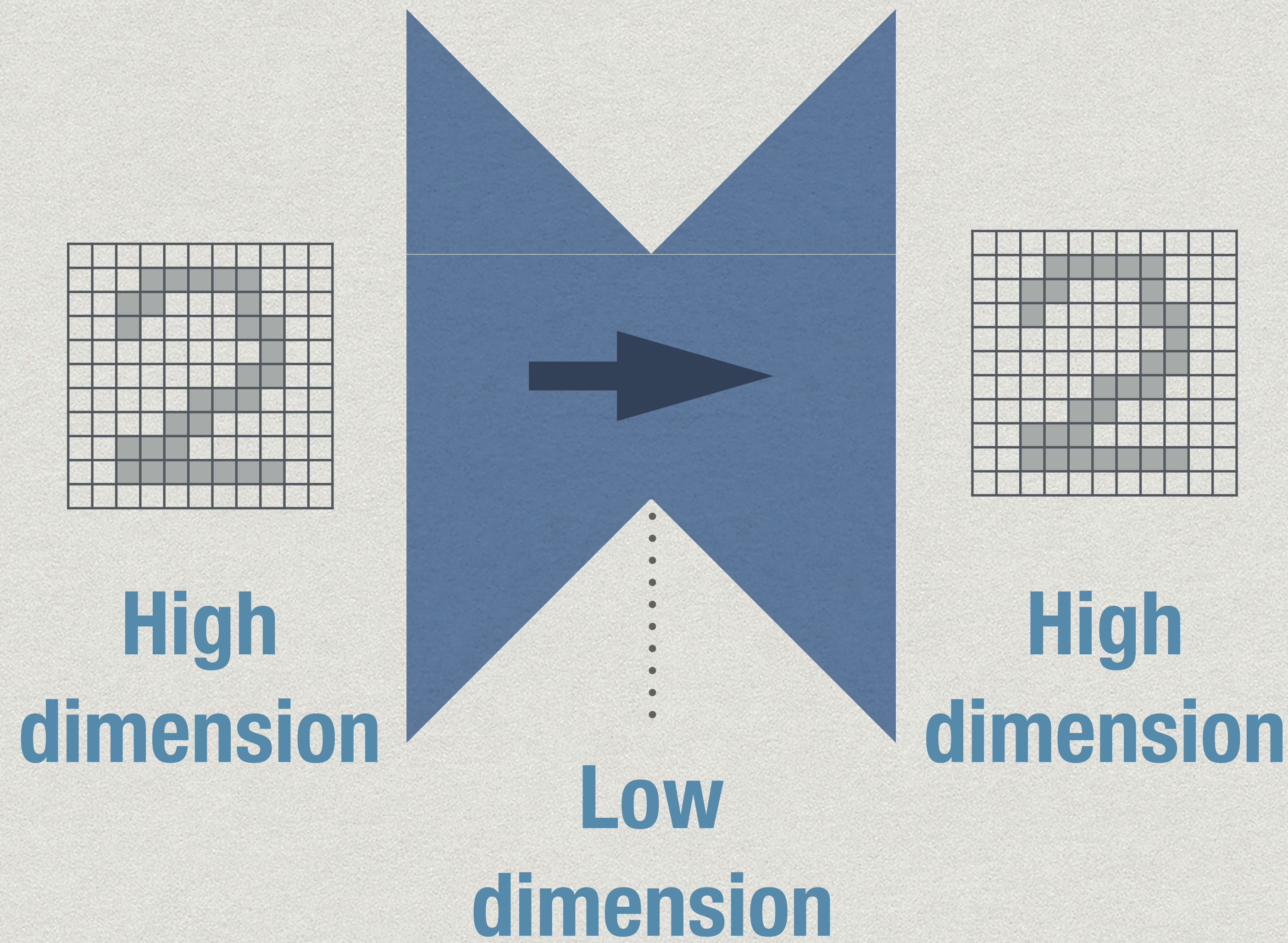
Autoencoder



Unsupervised example: the autoencoder



Autoencoder



- Success means:
  - meaningful features have been extracted at the bottleneck ***without guidance***
  - they are a low dimensional representation of most important features

GREAT. SO HOW DO WE USE IT?

GREAT. SO HOW DO WE USE IT?

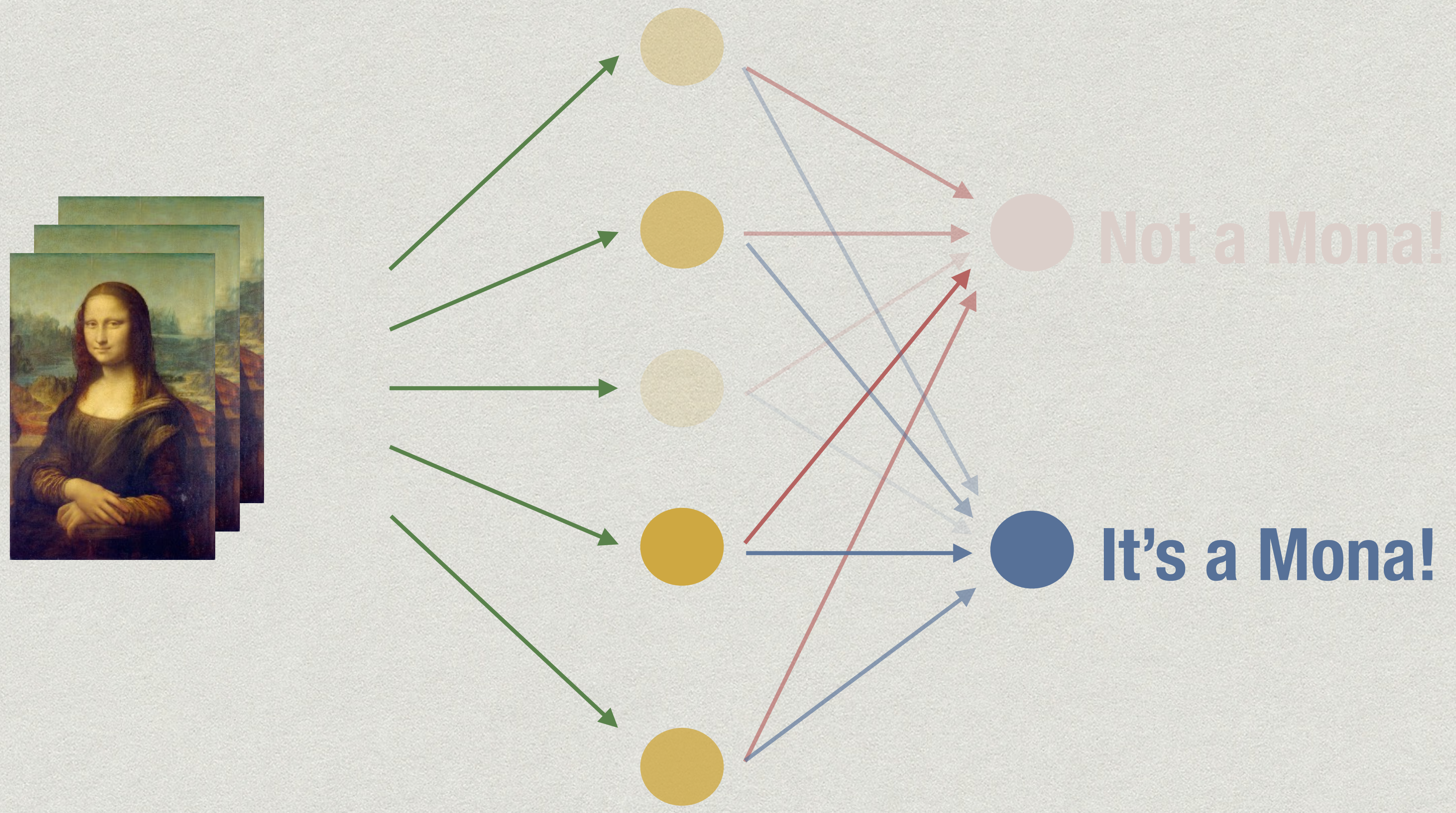
GENERATIVE ADVERSARIAL NETWORKS

« Generative Adversarial Networks is the most interesting idea in the last ten years in machine learning »

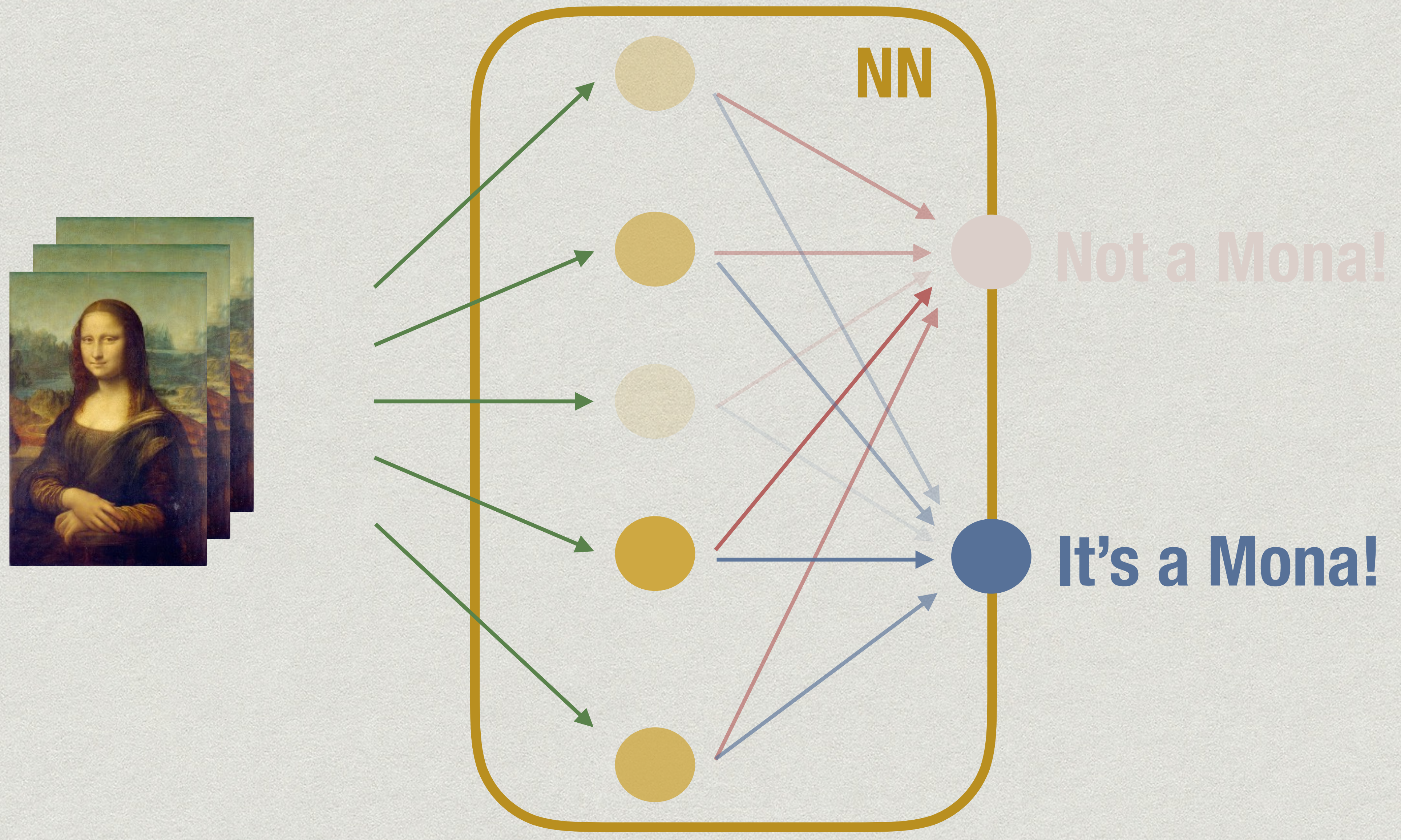
-Yann LeCun

The big deal guy

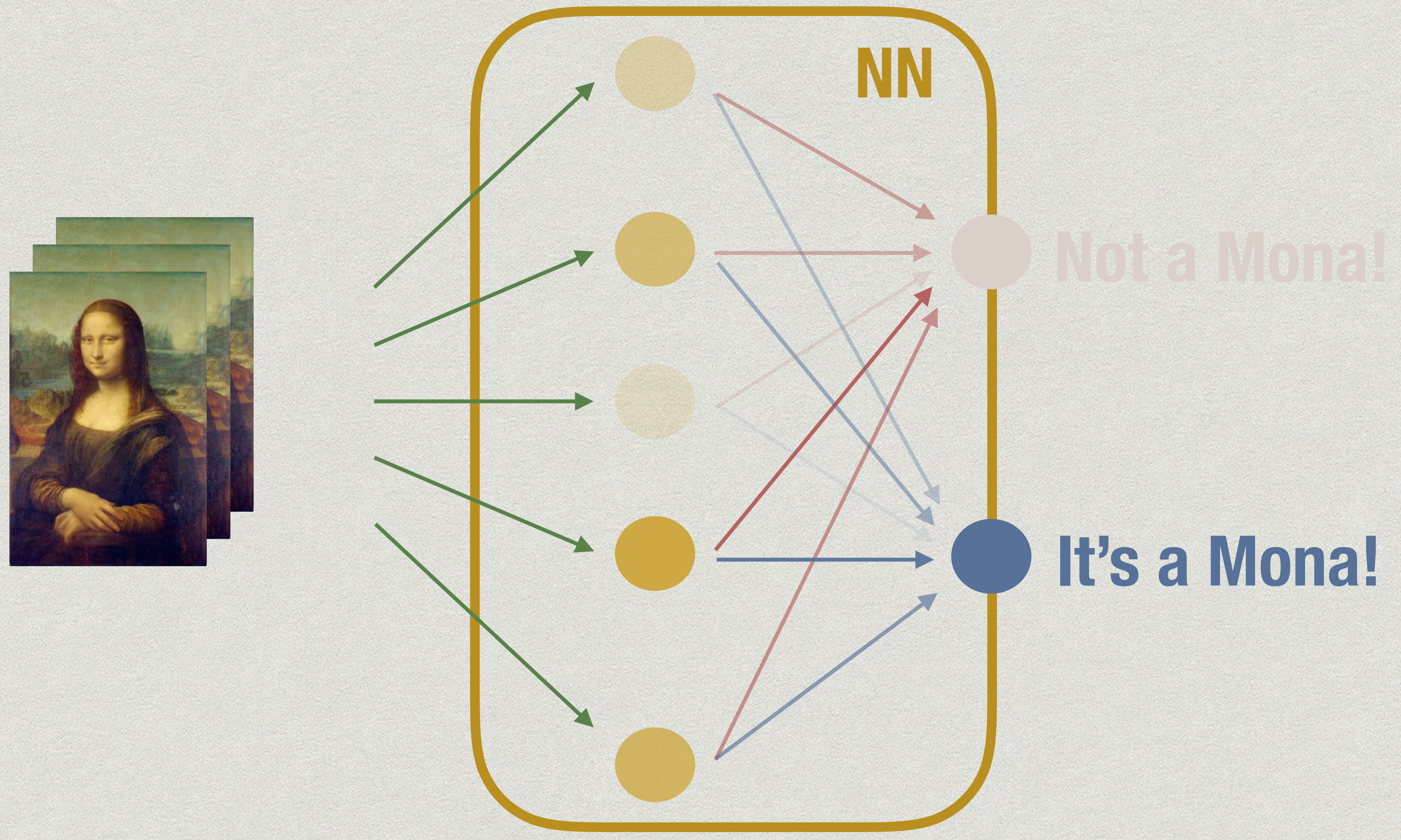
Classifiers / Regressors



Classifiers / Regressors

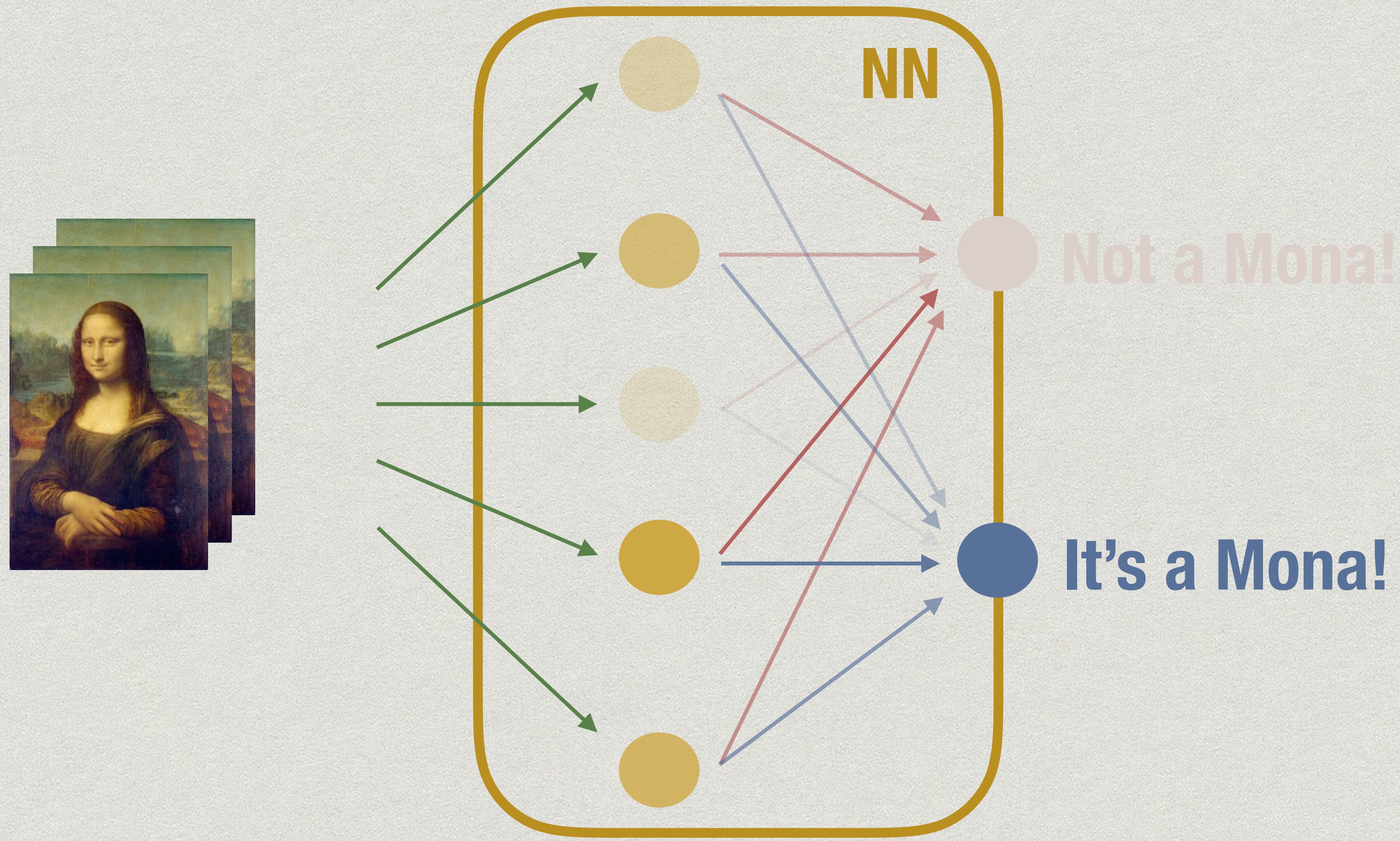


Classifiers / Regressors



Can answer a specific question
= low dimensional output

Classifiers / Regressors



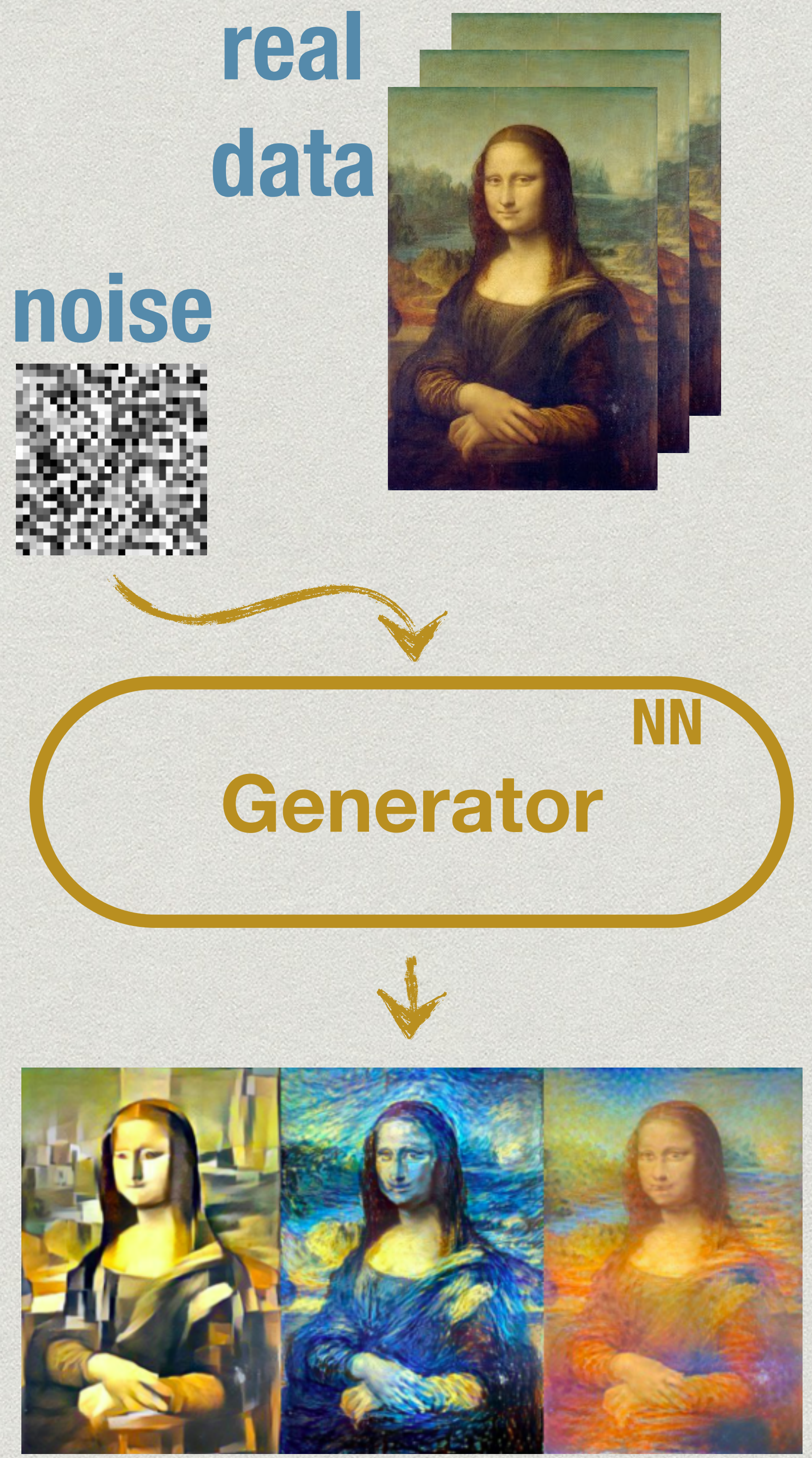
Can answer a specific question = low dimensional output

Cannot generate complex data = high dimensional output

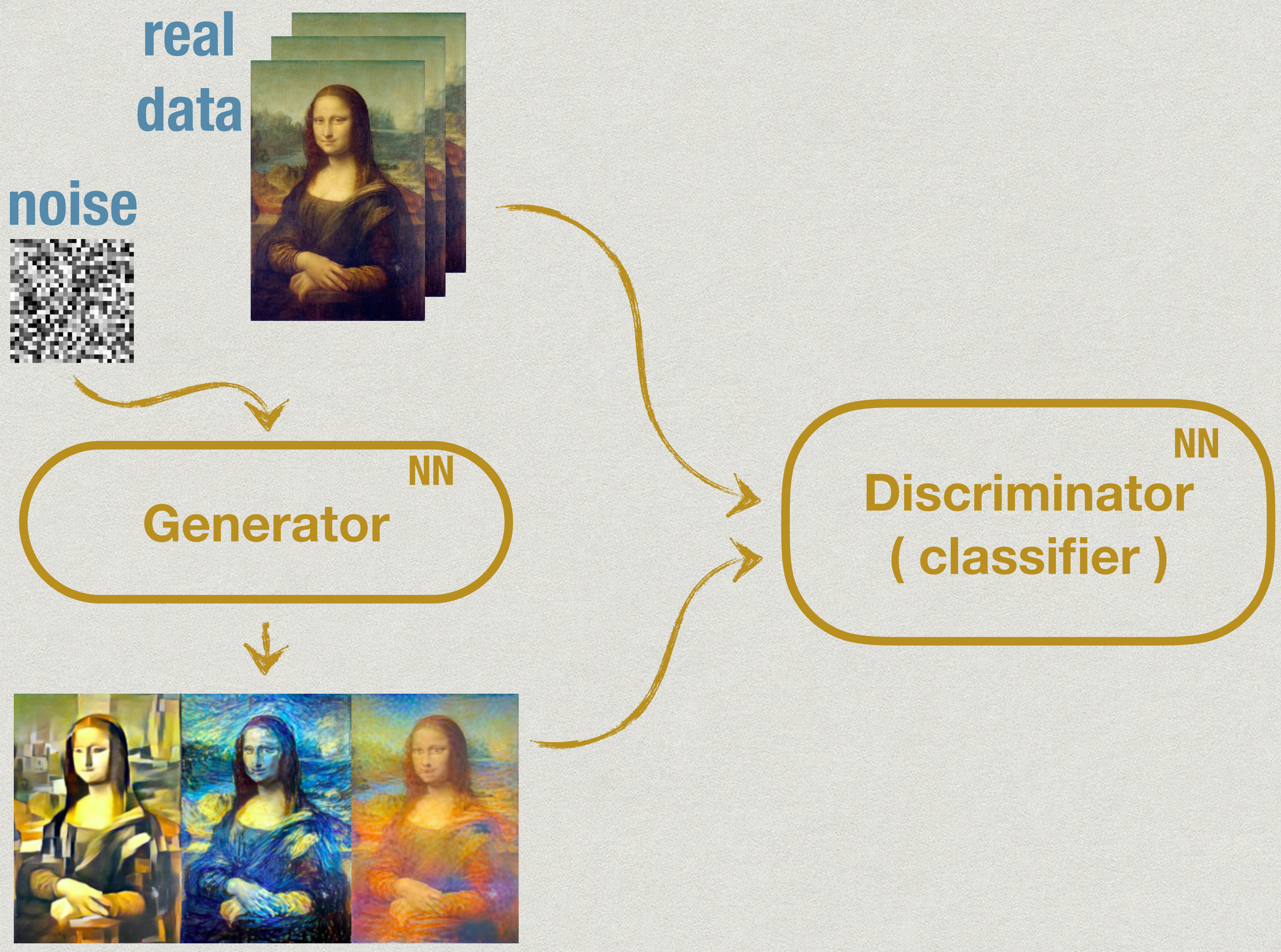
Adverserial generators

real data

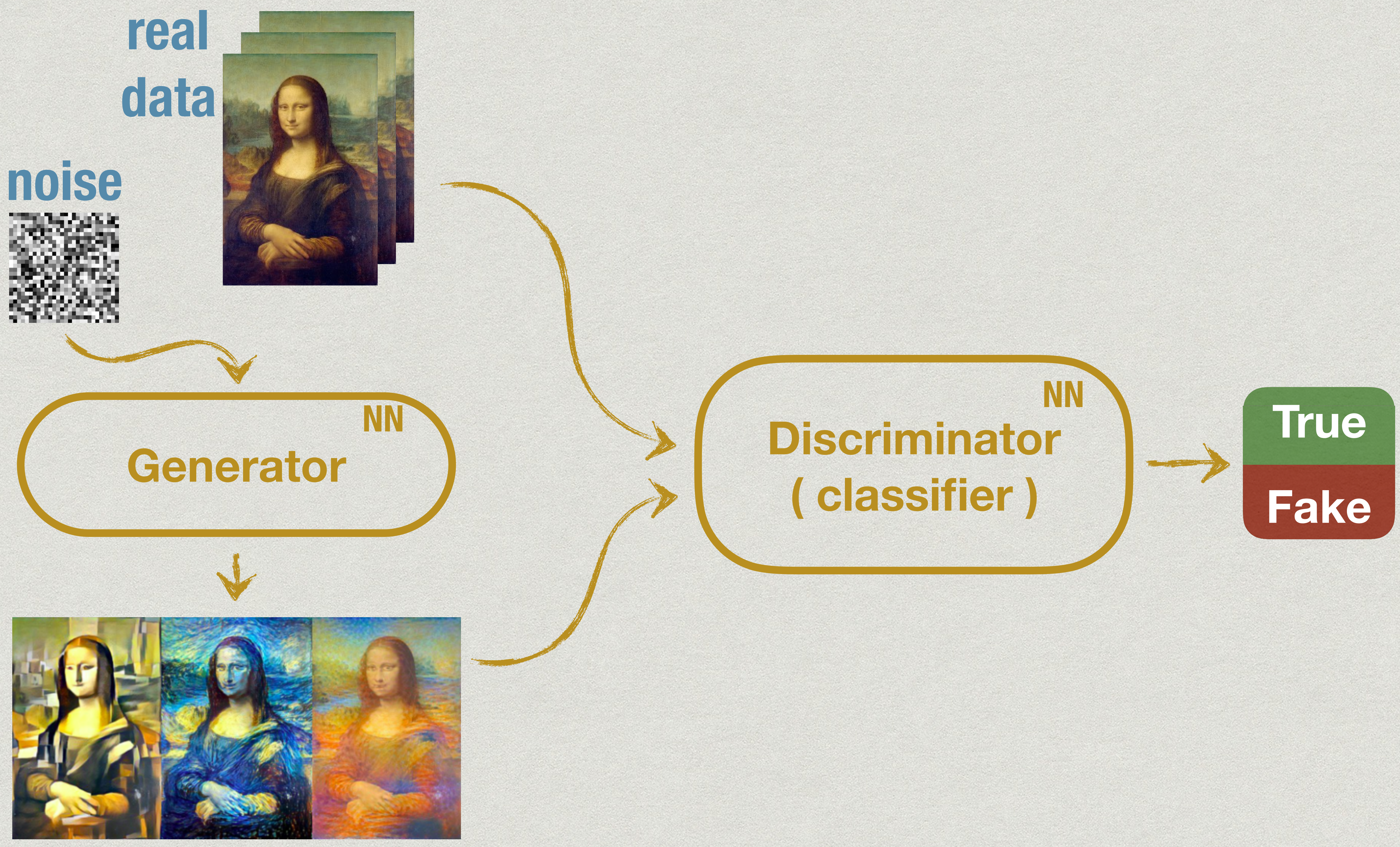
Adverserial generators



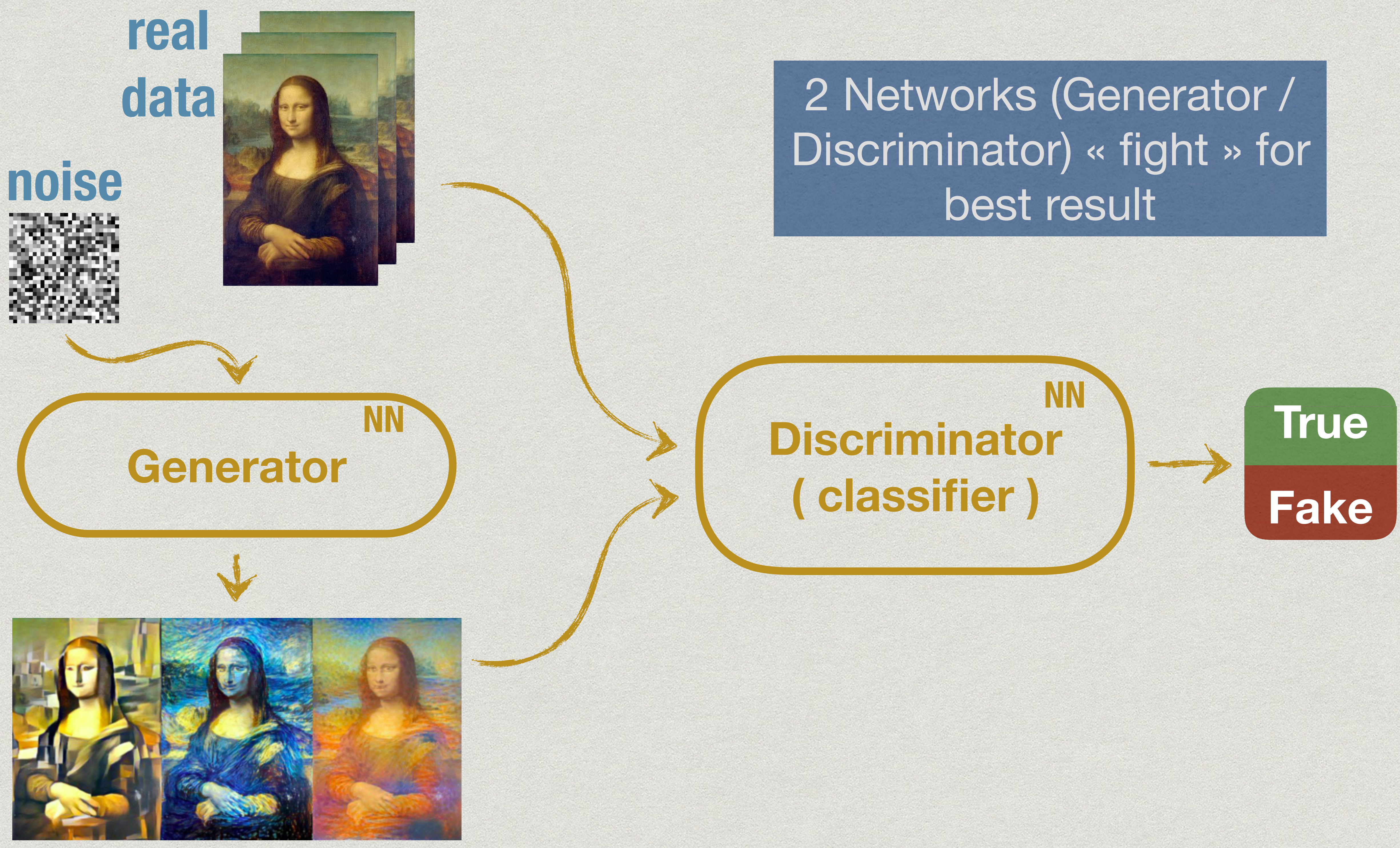
Adverserial generators



Adverserial generators



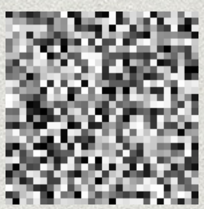
Adverserial generators



Adverserial generators

real data

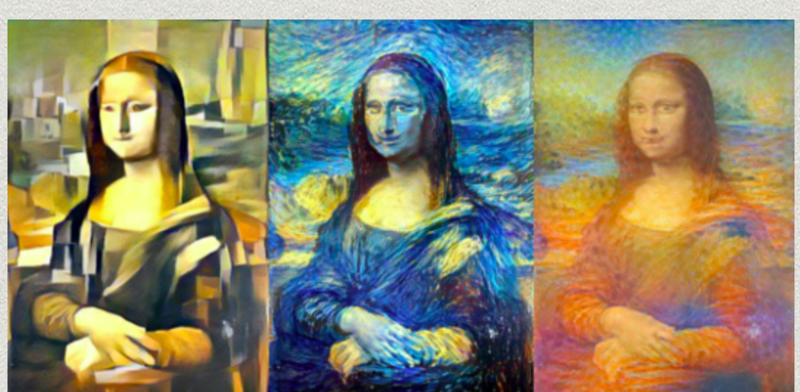
noise



2 Networks (Generator / Discriminator) « fight » for best result

Generator

NN



Discriminator
(classifier)

NN

True

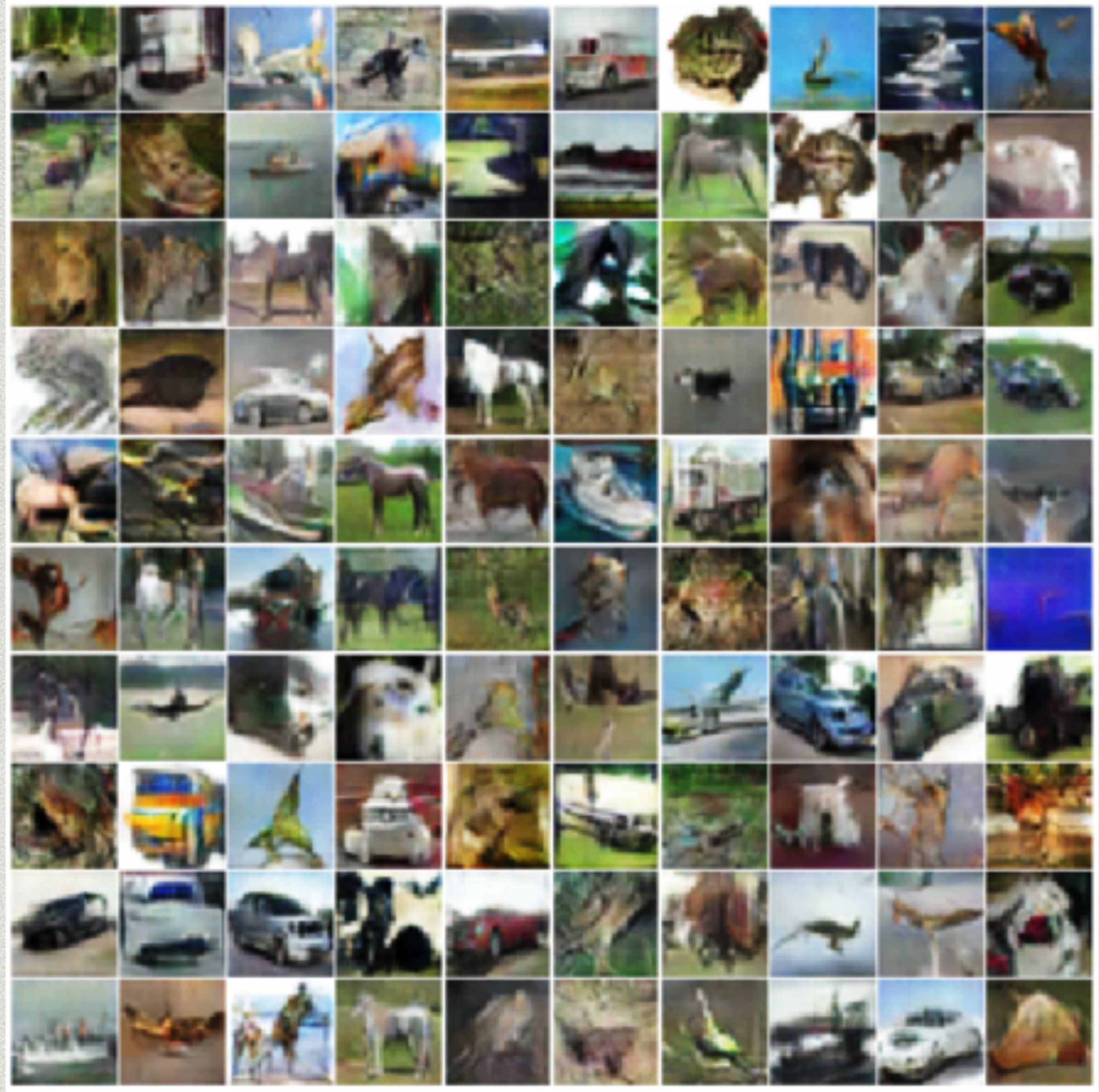
Fake

Once trained, the generator can create « plausible » images

GAN examples

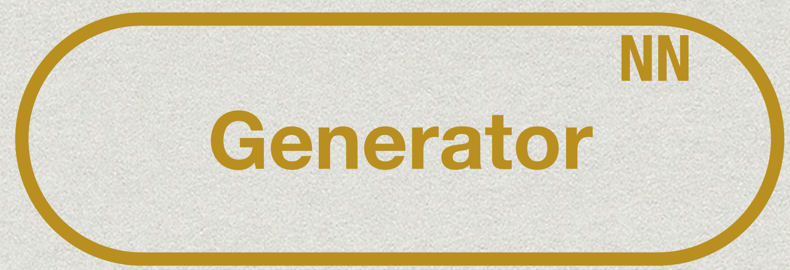
Generated bedroom images

GAN examples



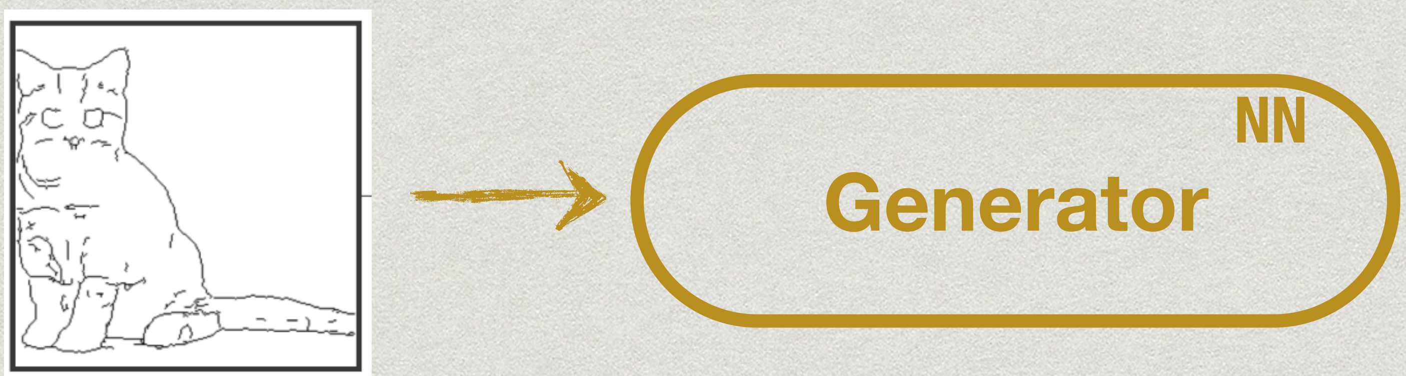
Realistic images according to CIFAR-10 dataset

Pix2Pix: a more useful GAN



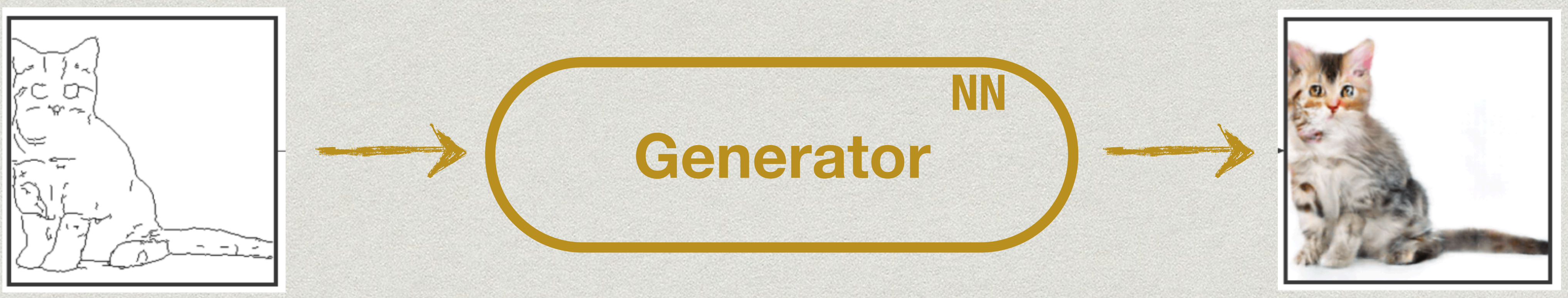
The generator can be tweaked to accept an input

Pix2Pix: a more useful GAN



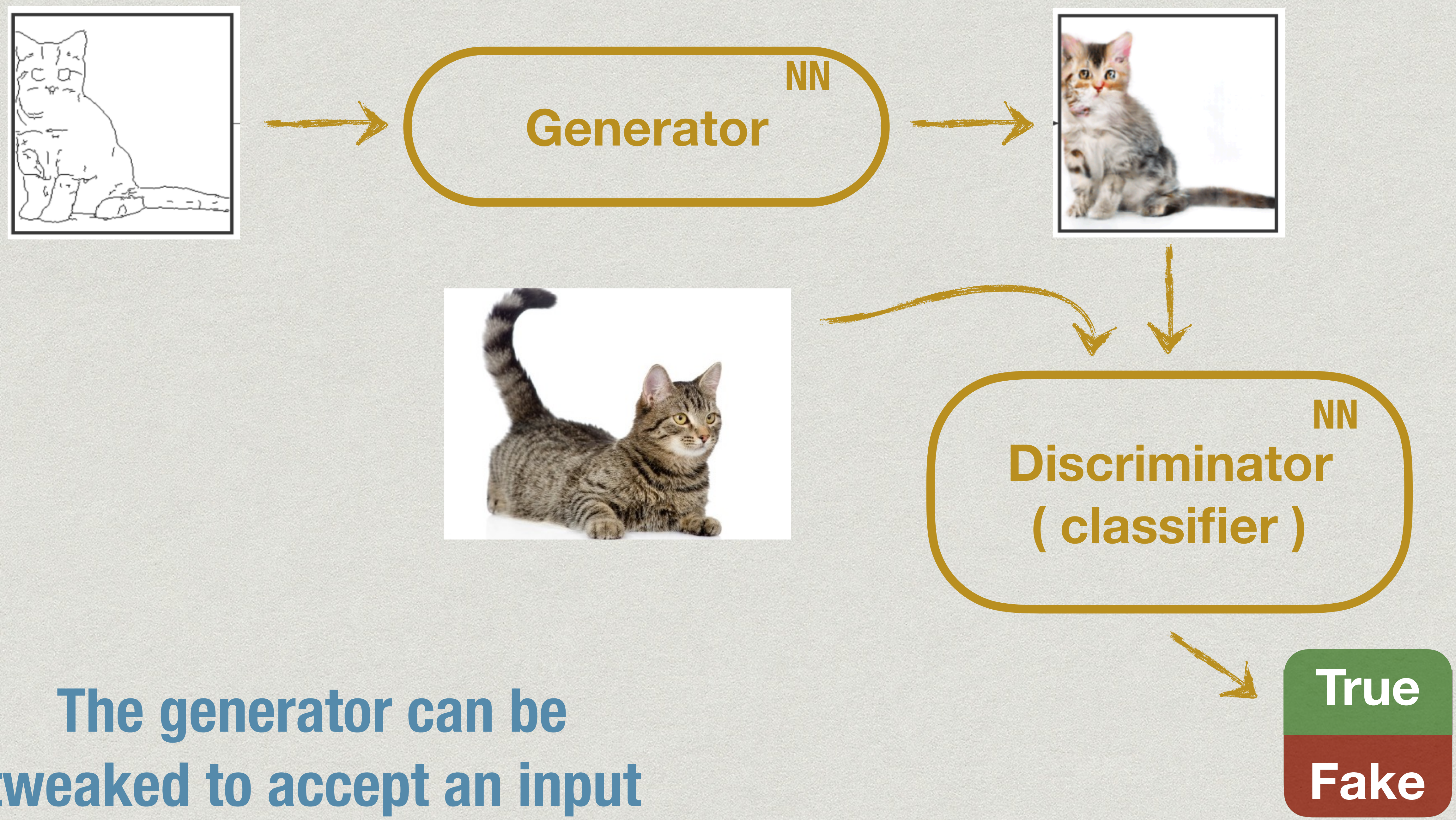
The generator can be tweaked to accept an input

Pix2Pix: a more useful GAN



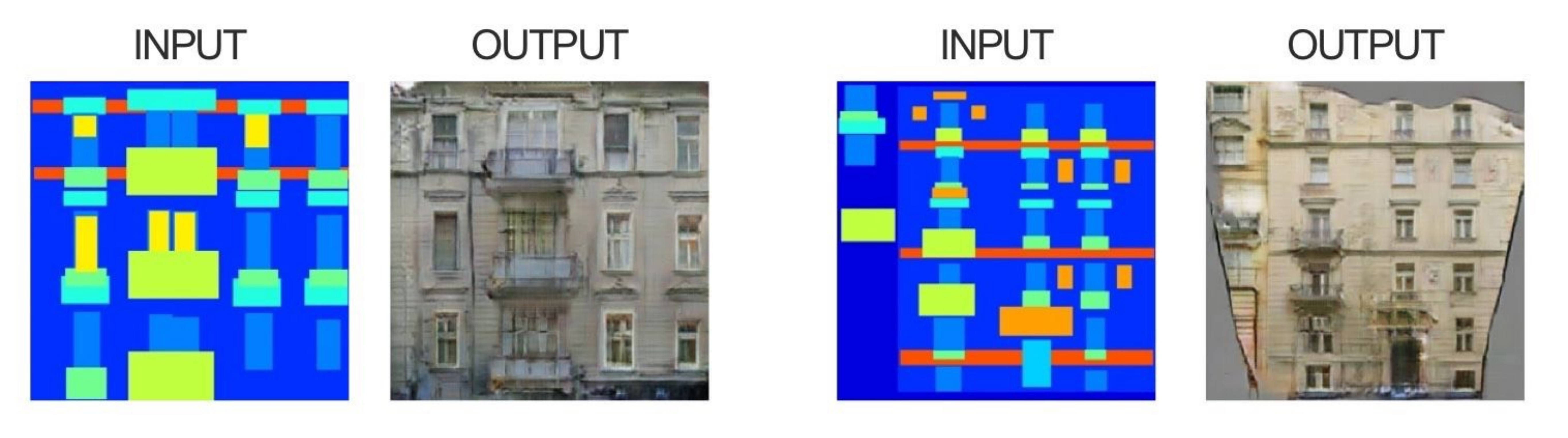
The generator can be tweaked to accept an input

Pix2Pix: a more useful GAN



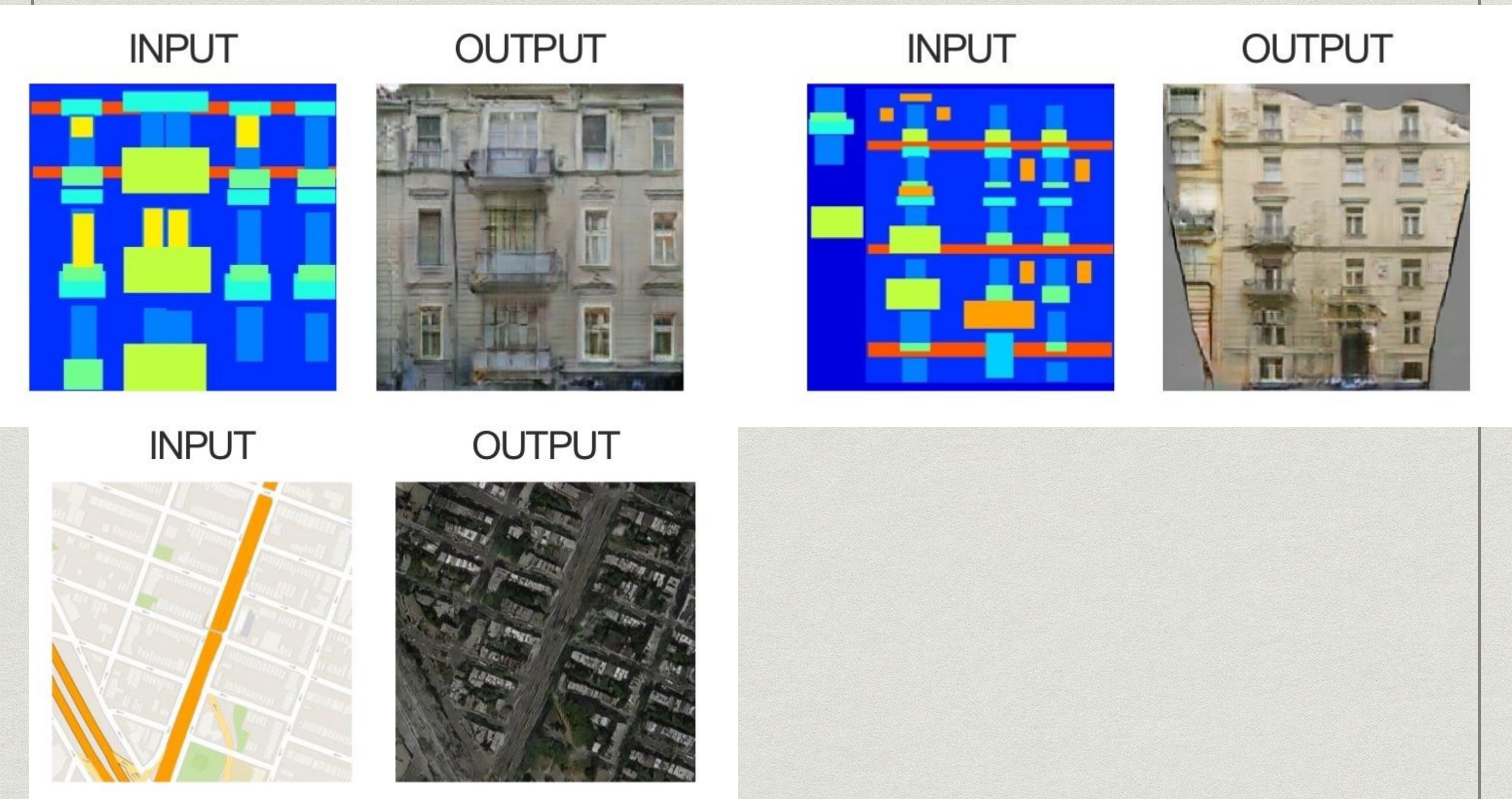
The generator can be tweaked to accept an input

Pix2Pix: a more useful GAN



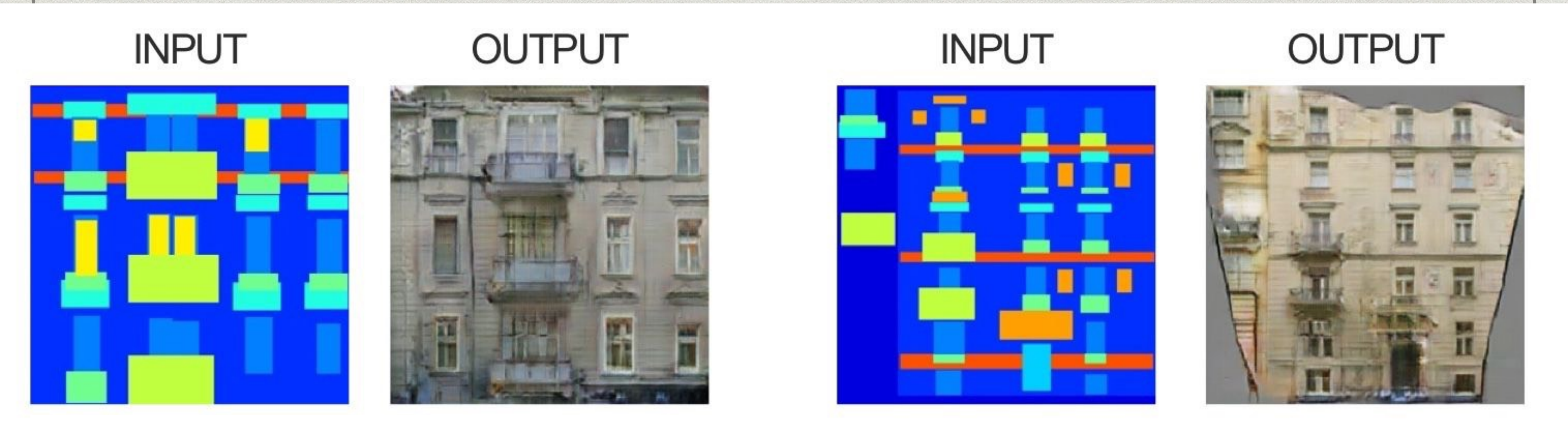
live test (you draw!) at: https://affinelayer.com/pixsrv/

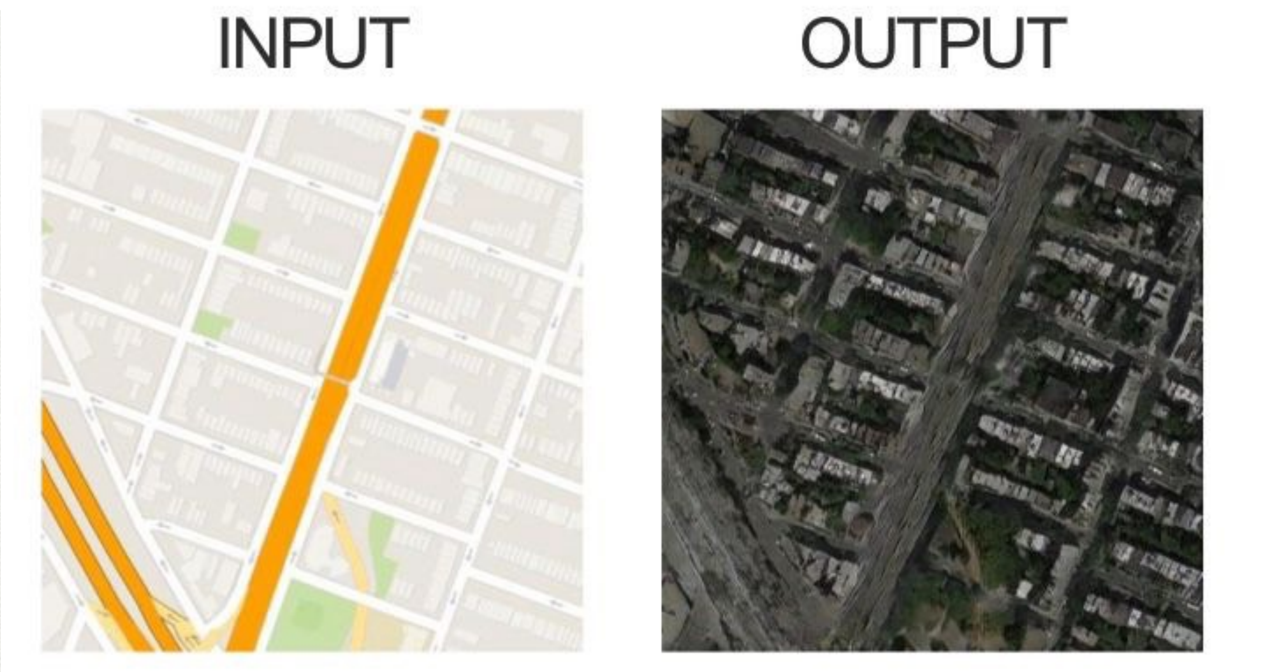
Pix2Pix: a more useful GAN

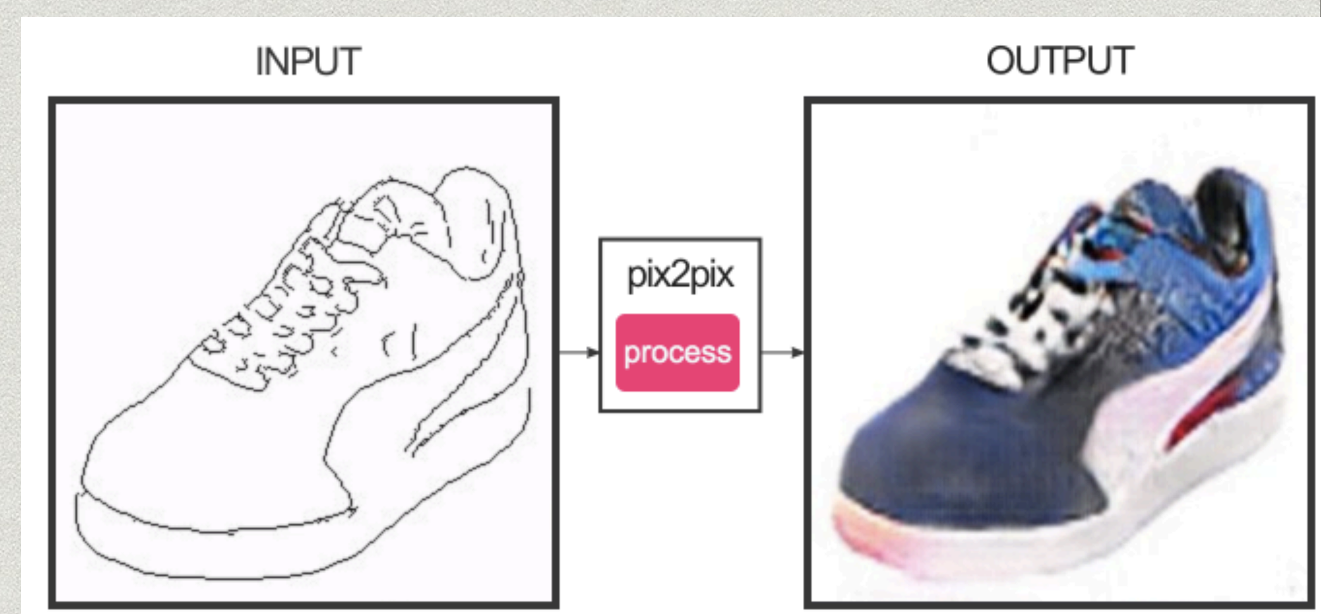


live test (you draw!) at: https://affinelayer.com/pixsrv/

Pix2Pix: a more useful GAN







live test (you draw!) at: https://affinelayer.com/pixsrv/

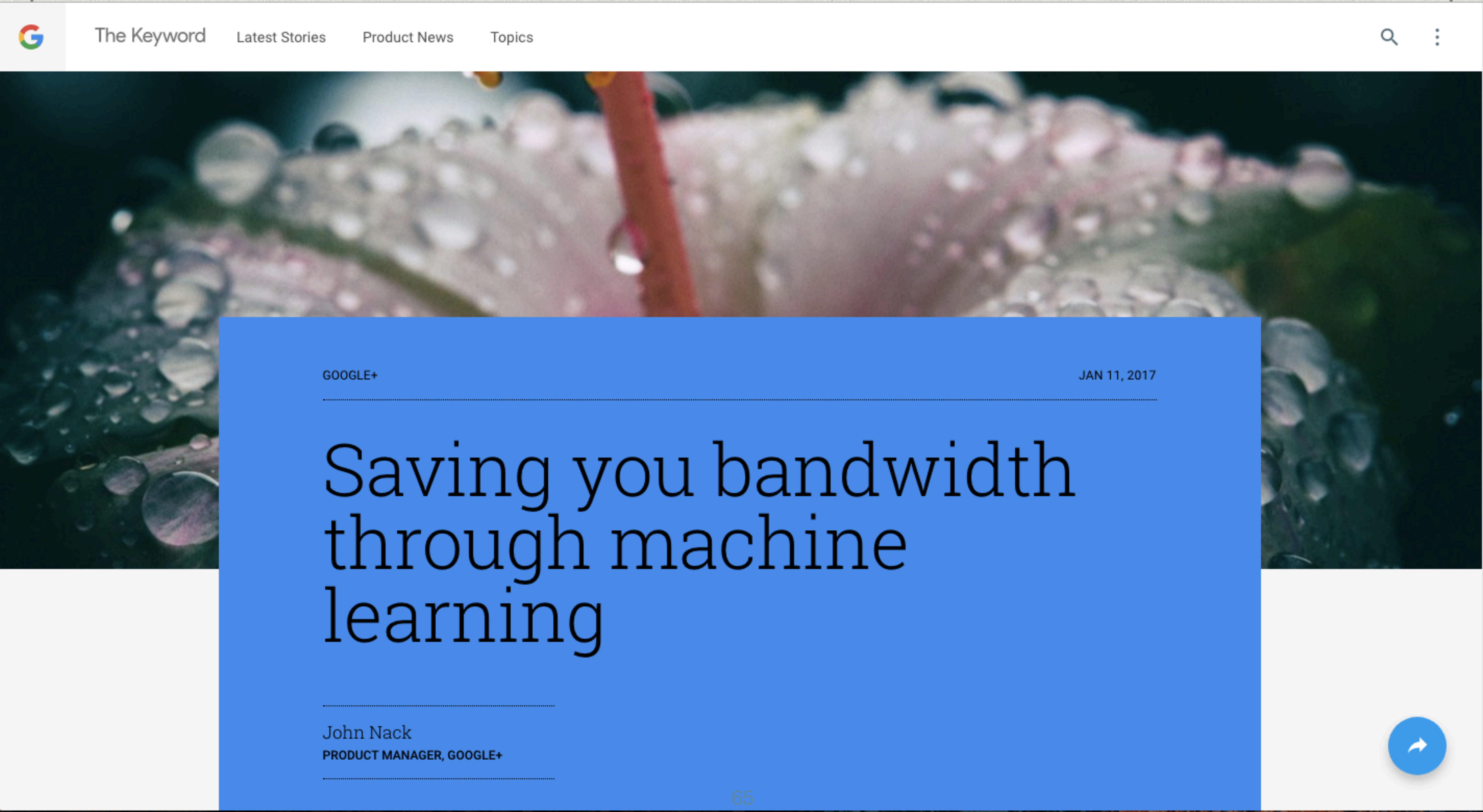
SRGAN

Super - Resolution GAN

- **Output: 64x64 images (from the `Large-scale CelebFaces Attributes` dataset)**
- **Input: degraded 16x16 image**
- **GAN learns to reproduce « credible » images**

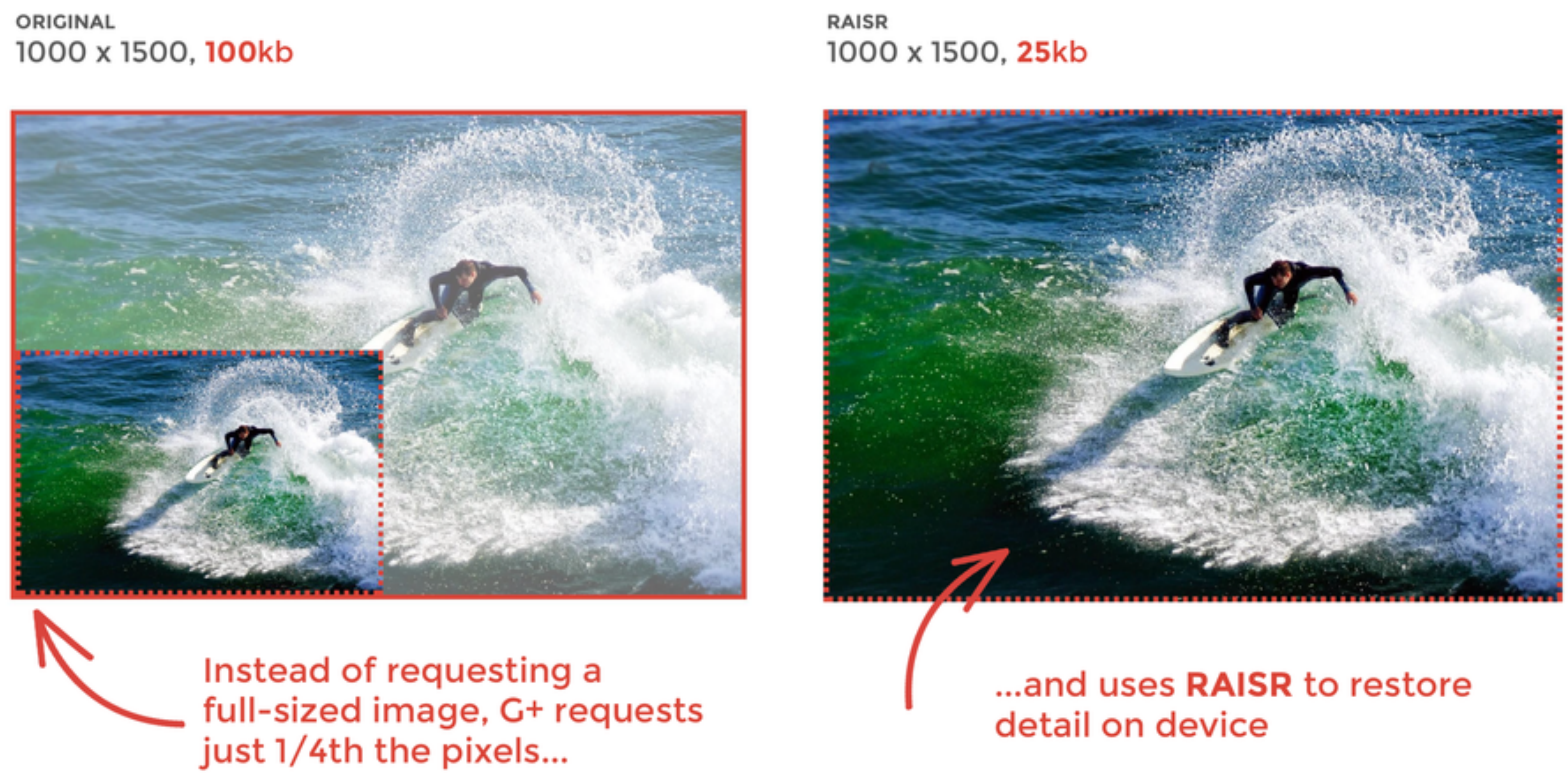
Google + (RAISR)

https://blog.google/products/google-plus/saving-you-bandwidth-through-machine-learning/



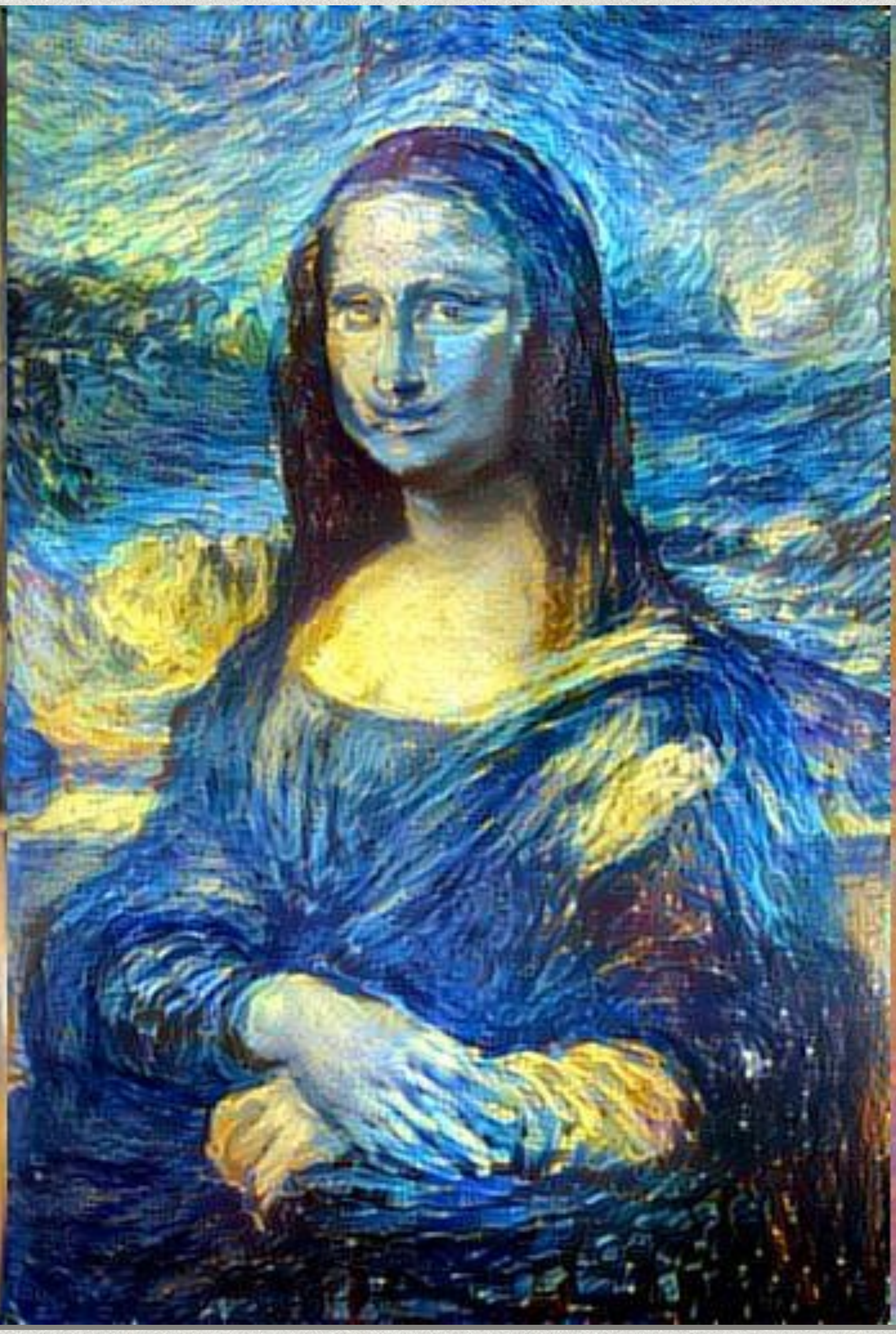
Google + (RAISR)

https://blog.google/products/google-plus/saving-you-bandwidth-through-machine-learning/

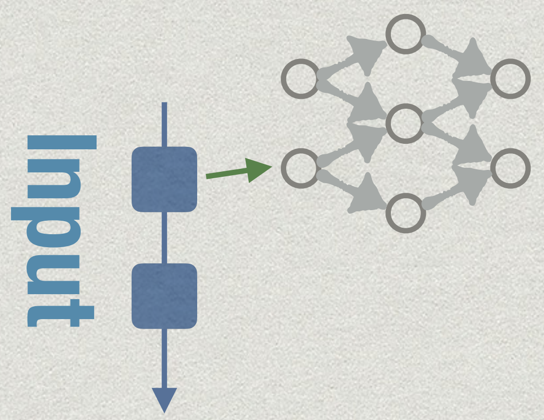


Style Transfer

Picasso **van Gogh** **Monnet**

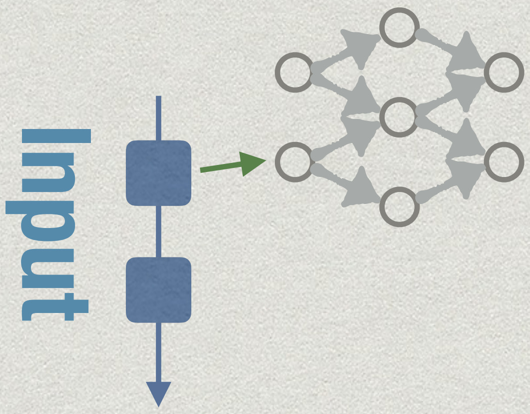


And so many more…



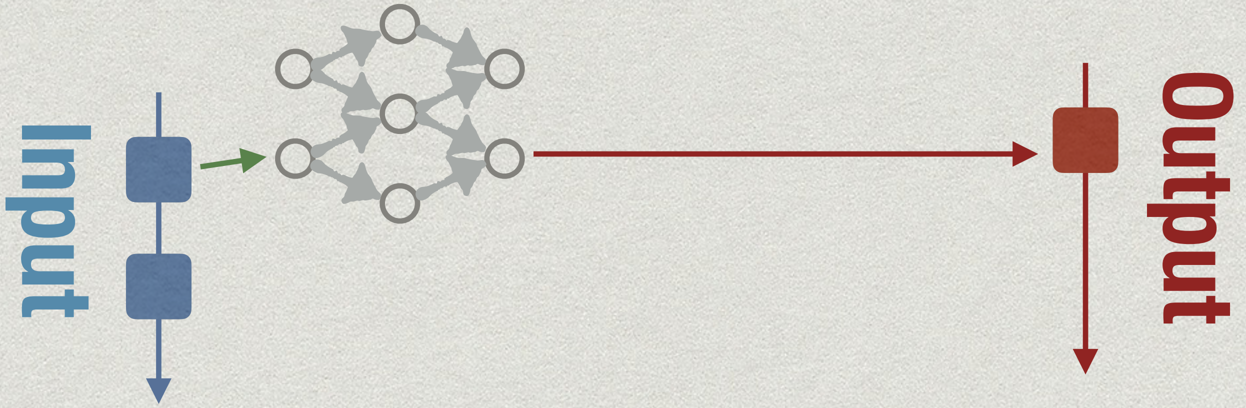
And so many more…

- Recurrent NN:



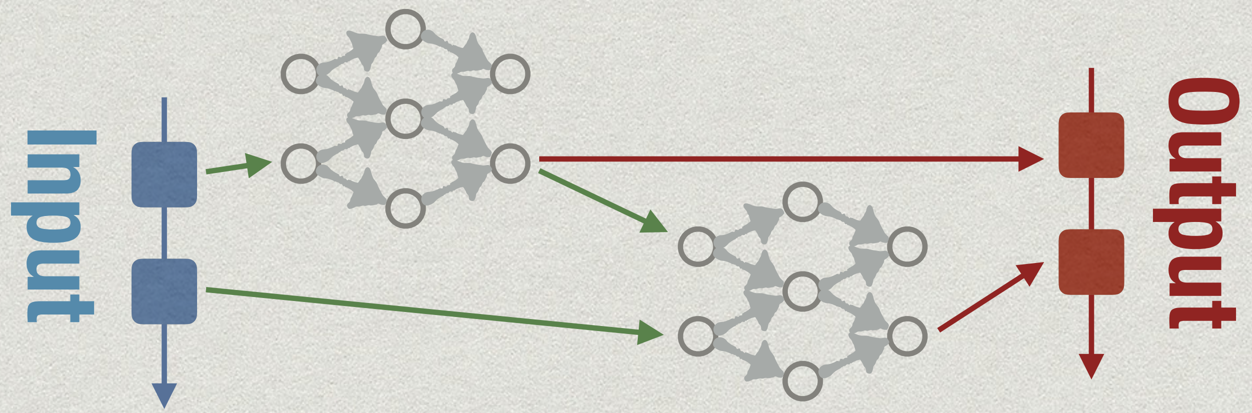
And so many more…

* Recurrent NN:

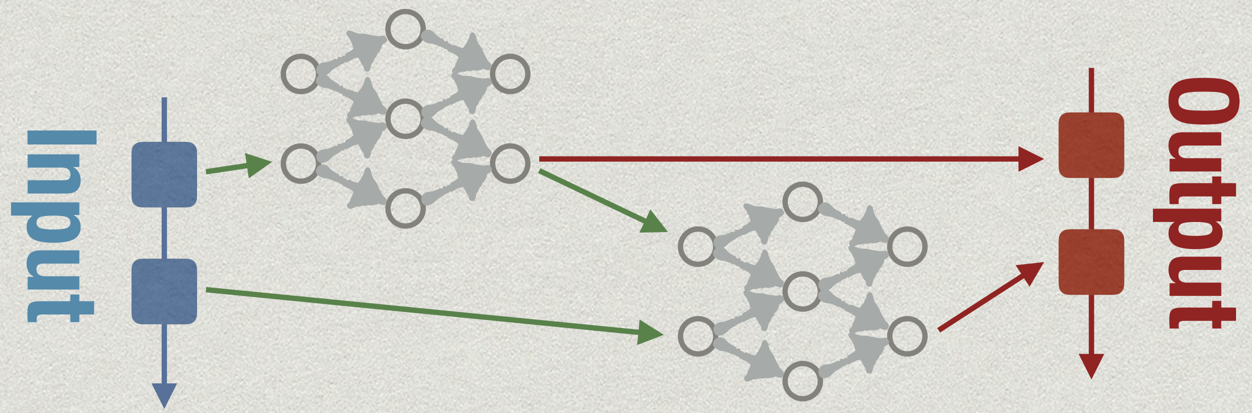


And so many more…

* Recurrent NN:

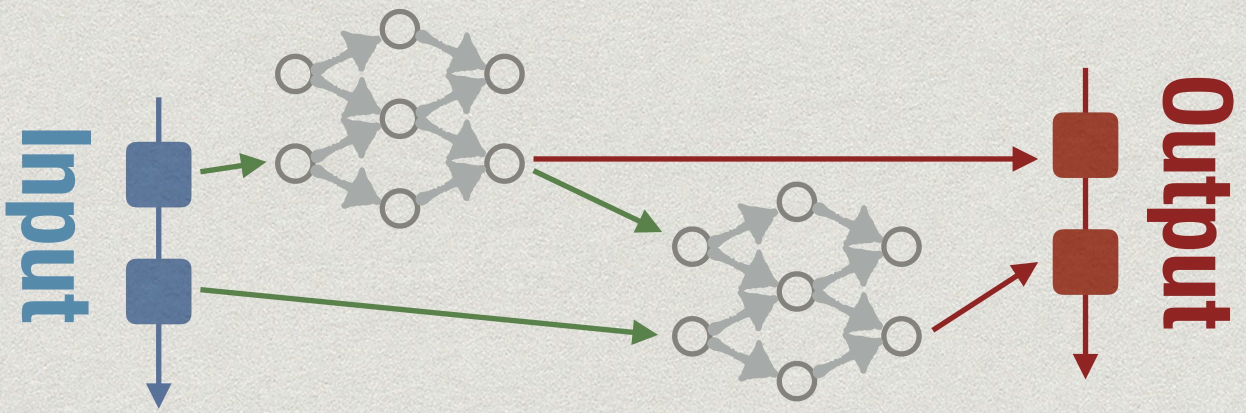


And so many more…



- Recurrent NN:
  - *e.g.* LSTM (Long Short-term Memory), a convolution over time. Used in speech recognition

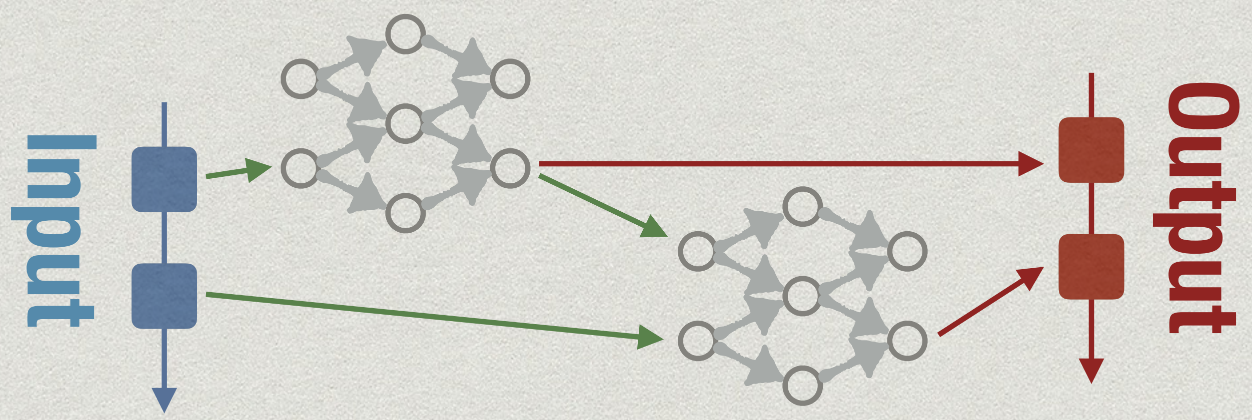
And so many more…



- Recurrent NN:
  - *e.g.* LSTM (Long Short-term Memory), a convolution over time. Used in speech recognition
- Variational Autoencoders: another unsupervised generative model like GANs

And so many more…

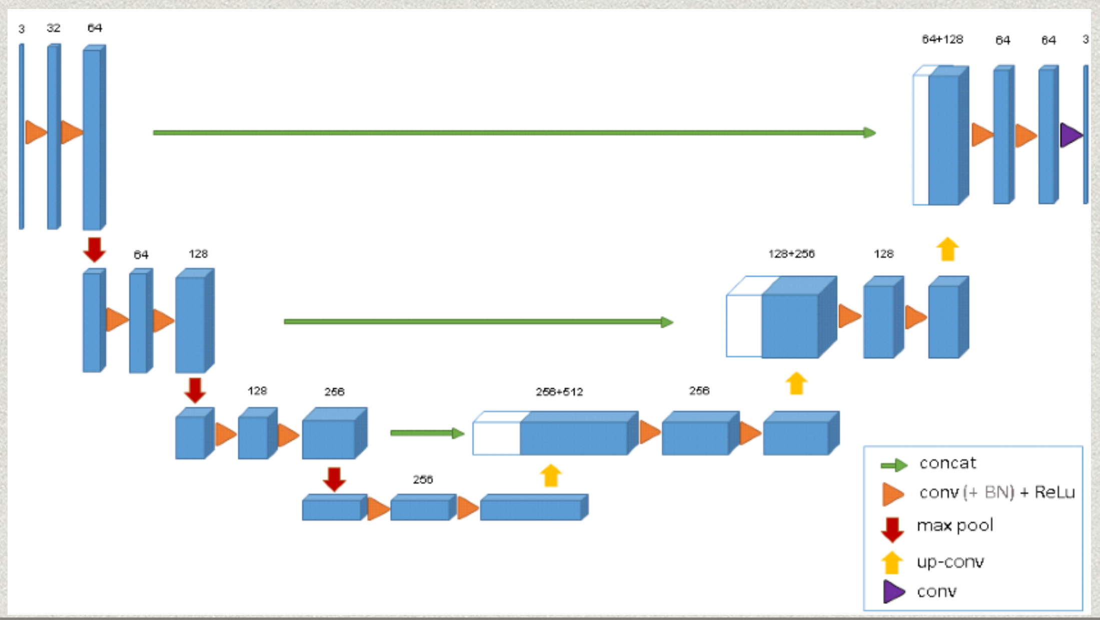
- Recurrent NN:



  - *e.g.* LSTM (Long Short-term Memory), a convolution over time. Used in speech recognition

- Variational Autoencoders: another unsupervised generative model like GANs

- U-Nets :



- etc…

DEEP LEARNING IN CFD: SOME EXAMPLES

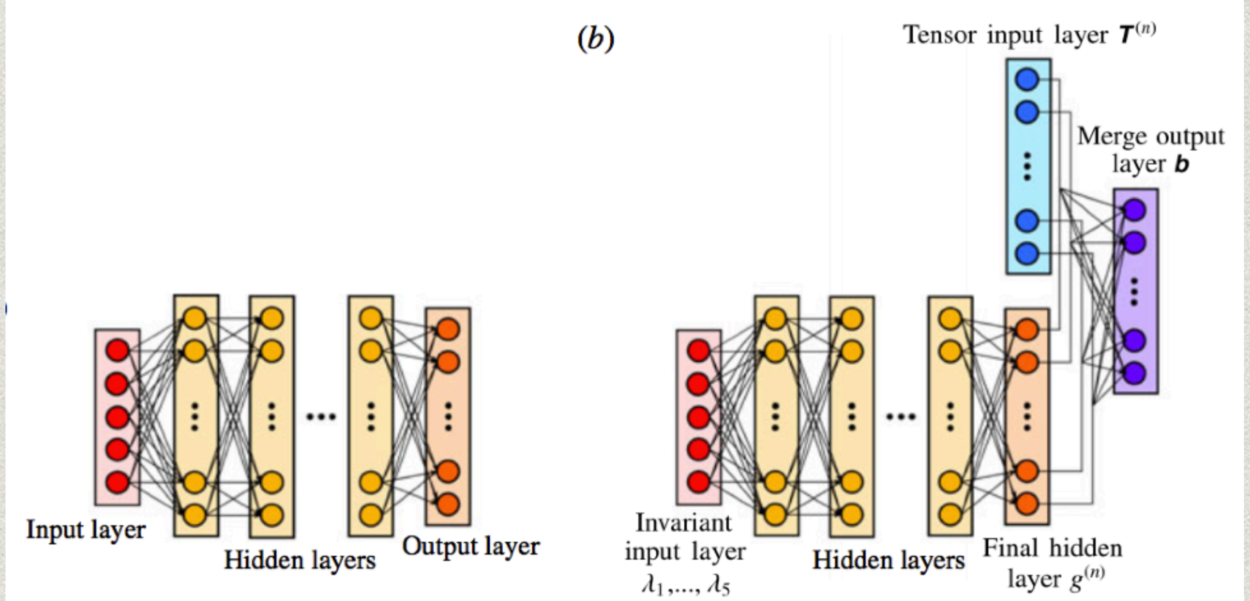
Post Processing

	POD / DMD	Deep Neural Networks
Computational speed		
Physically interpretable		
Ability to capture multi-scale		
Invariance by rotation/scaling		

Suggestion: propose a dataset for a fluid mechanics challenge (like ImageNet)

Kutz, J. Nathan. "Deep learning in fluid dynamics." Journal of Fluid Mechanics 814 (2017): 1-4.

RANS modelling



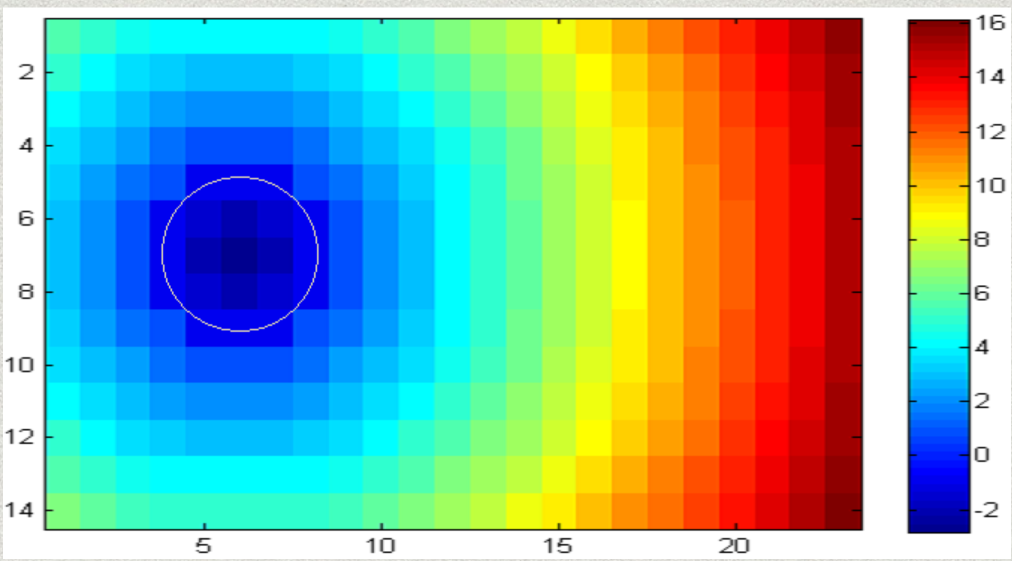
Input: S and R (strain / rotation rate tensors)

Output: Reynolds stress anisotropy tensor

- Authors compared a simple MLP (not great) with a smarter NN accounting for rotational invariance
- They obtained excellent predictions (much better than a quadratic eddy viscosity model)

Ling, Julia, Andrew Kurzawski, and Jeremy Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance." Journal of Fluid Mechanics 807 (2016): 155-166.

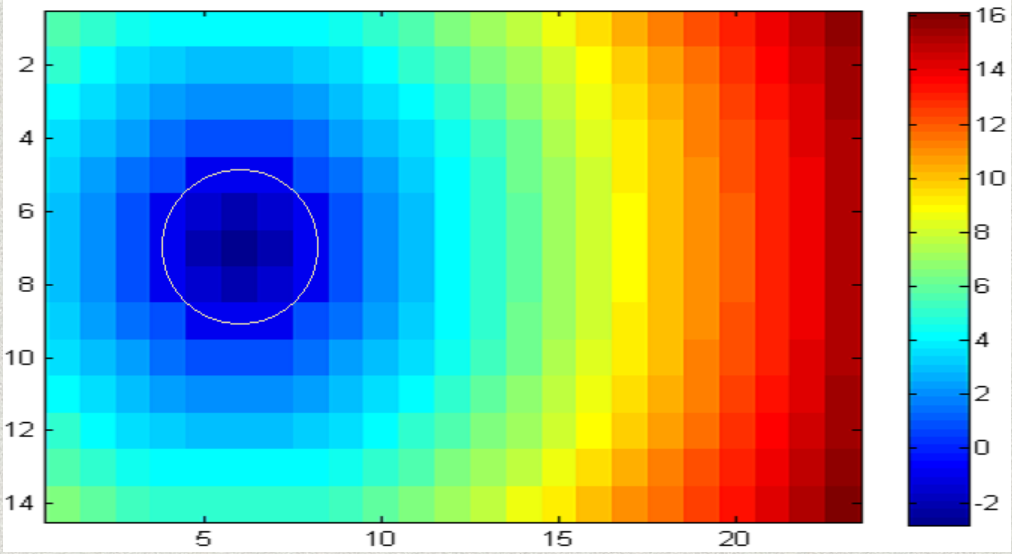
LBM imitation



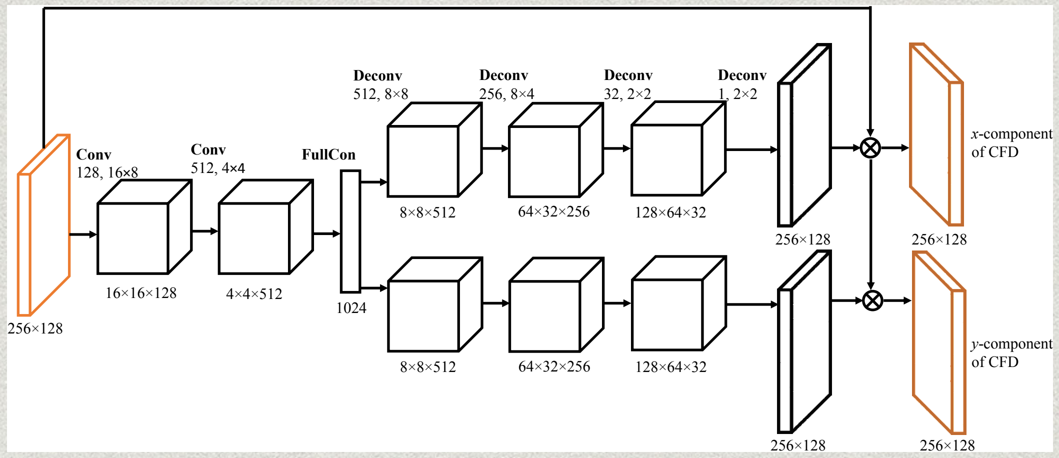
Signed Distance Function

Guo, Xiaoxiao, Wei Li, and Francesco Iorio. "Convolutional neural networks for steady flow approximation." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

LBM imitation

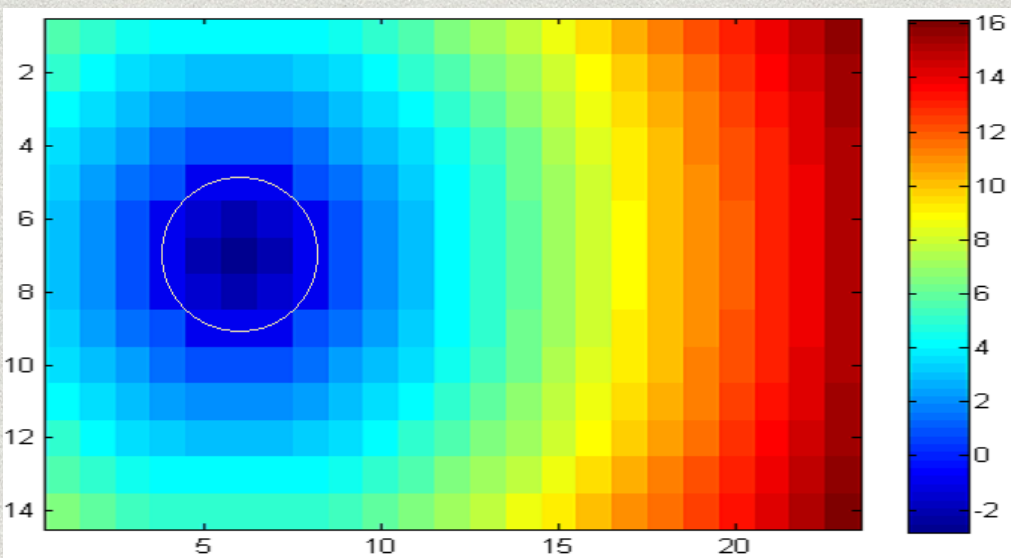


Signed Distance Function

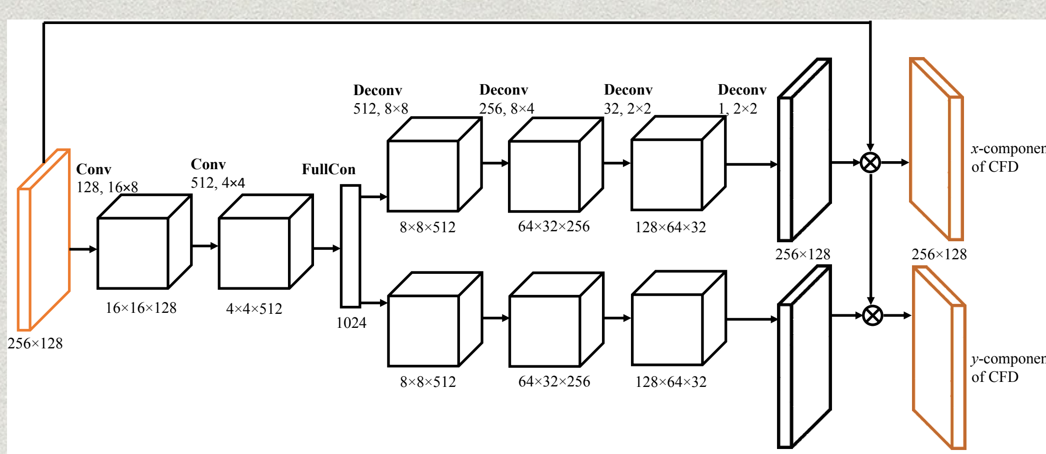


Guo, Xiaoxiao, Wei Li, and Francesco Iorio. "Convolutional neural networks for steady flow approximation." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

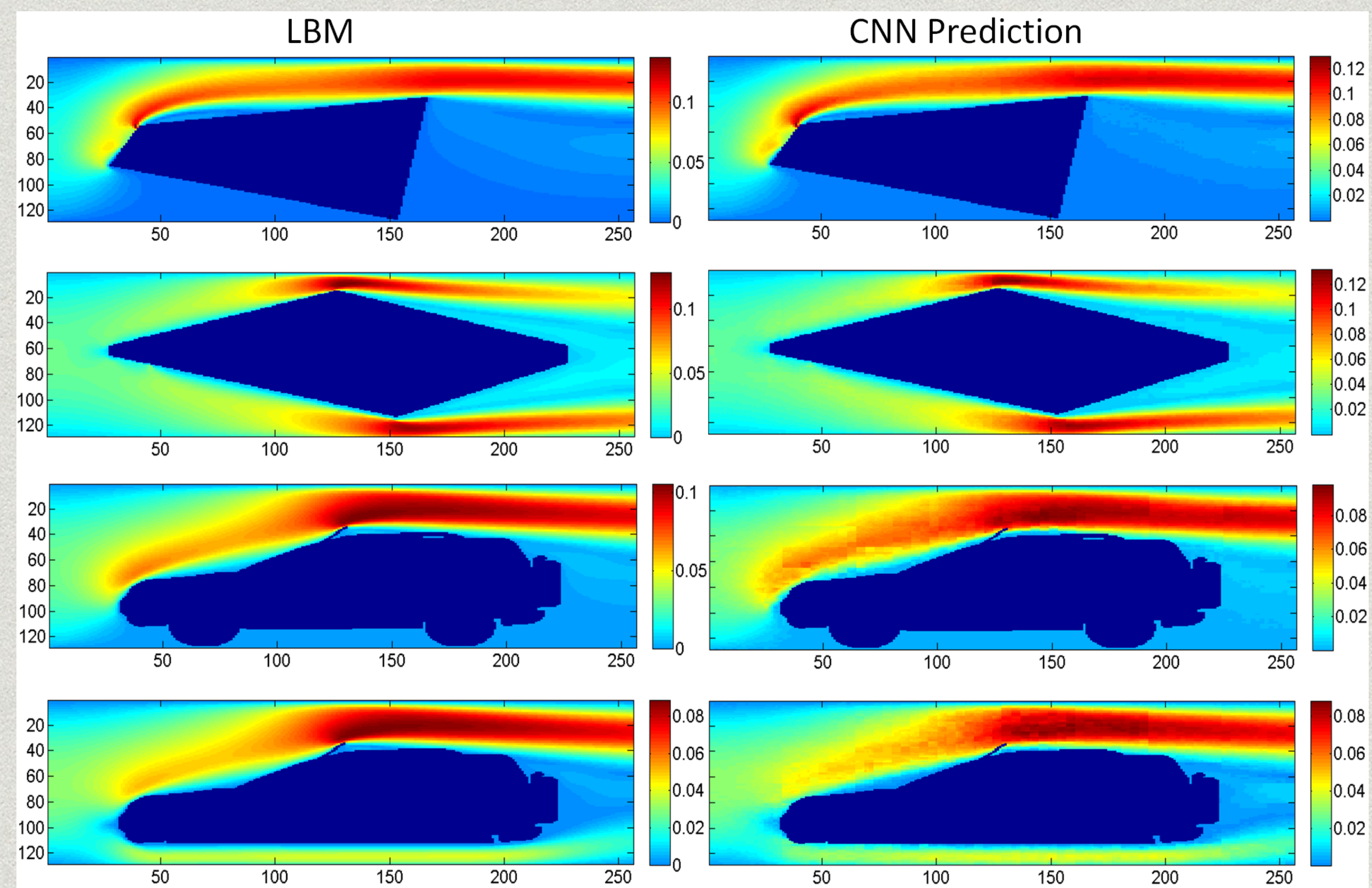
LBM imitation



Signed Distance Function



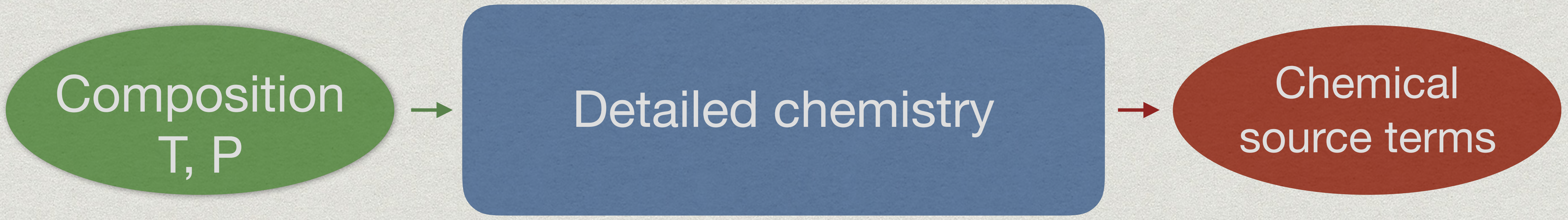
OpenLB (Karlsrühe) and « Proprietary LBM solver » (autodesk)



Guo, Xiaoxiao, Wei Li, and Francesco Iorio. "Convolutional neural networks for steady flow approximation." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

Chemistry

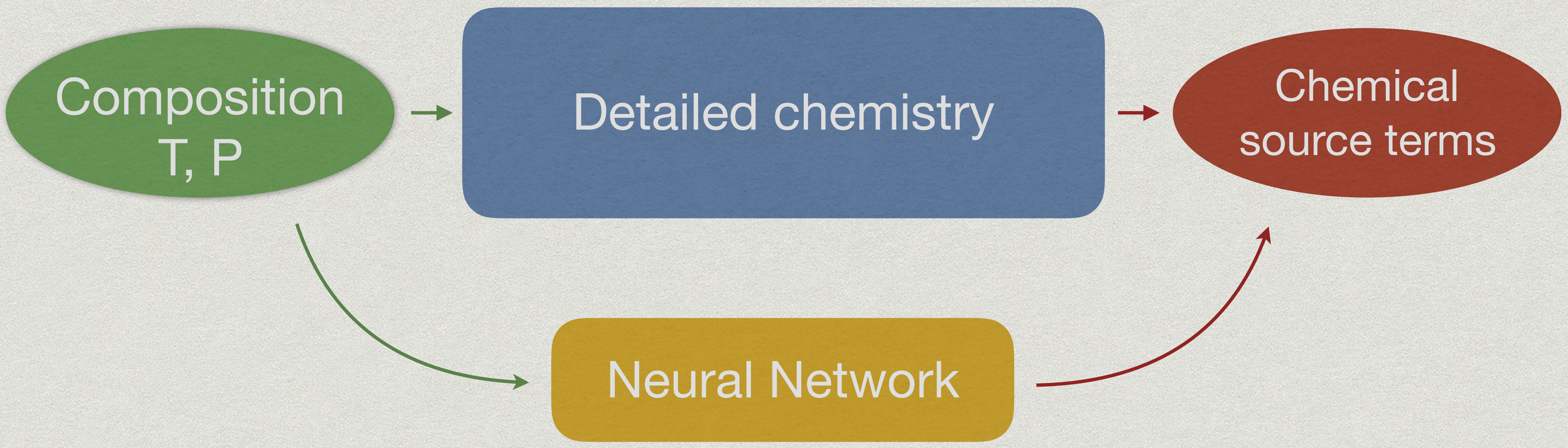
* Expensive chemical schemes operate over wide ranges of input parameters



Under review…

Chemistry

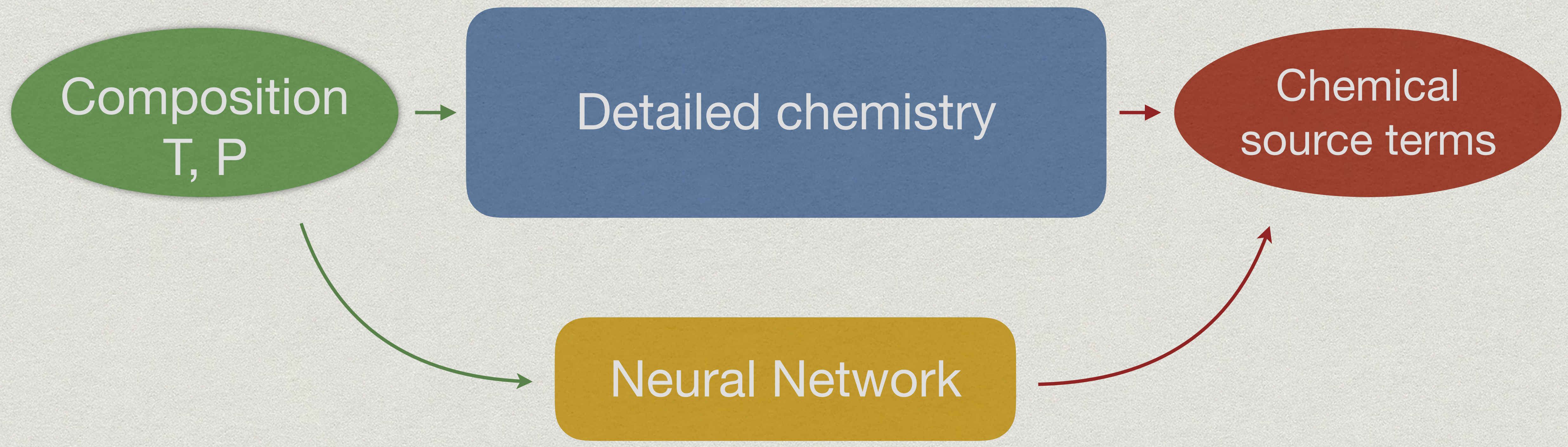
* Expensive chemical schemes operate over wide ranges of input parameters



Under review…

Chemistry

- Expensive chemical schemes operate over wide ranges of input parameters



Authors reduced cost of chemical computation by 100

Under review…

More importantly…

Anything else you can think of!

CONCLUSION

You can do it!

You can do it!

- Machine learning is on the rise…

You can do it!

- Machine learning is on the rise…
- …but Deep learning can be **breathtaking**. **Now** is the time to ride the tide!

You can do it!

- Machine learning is on the rise…
- …but Deep learning can be **breathtaking**. **Now** is the time to ride the tide!
- To leverage DL, you need a problem where multi-scale hierarchy (in time, space…) is important

You can do it!

- Machine learning is on the rise…
- …but Deep learning can be **breathtaking**. **Now** is the time to ride the tide!
- To leverage DL, you need a problem where multi-scale hierarchy (in time, space…) is important
- Complex fluid flows are probably good candidates (*e.g.* turbulence). Get to work!

You can do it!

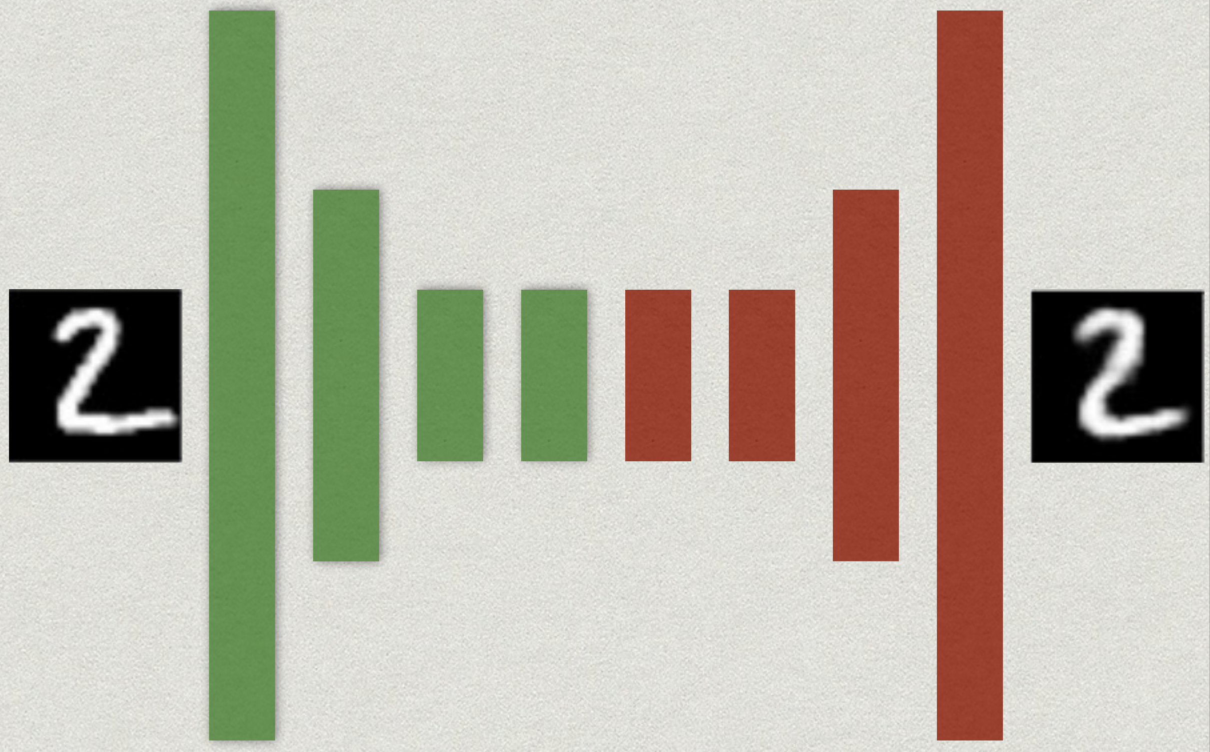
- Machine learning is on the rise…
- …but Deep learning can be **breathtaking**. **Now** is the time to ride the tide!
- To leverage DL, you need a problem where multi-scale hierarchy (in time, space…) is important
- Complex fluid flows are probably good candidates (*e.g.* turbulence). Get to work!

If you want to get your hands dirty, just ask!
Tensorflow / Theano already run at CERFACS

How do I start?

```
from keras.layers import Input, Dense, Conv2D, MaxPooling2D, UpSampling2D
from keras.models import Model
from keras import backend as K
from keras.datasets import mnist
import numpy as np

input_img = Input(shape=(28, 28, 1))

x = Conv2D(16, (3, 3), activation='relu', padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
encoded = MaxPooling2D((2, 2), padding='same')(x)

# at this point the representation is (4, 4, 8) i.e. 128-dimensional

x = Conv2D(8, (3, 3), activation='relu', padding='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(16, (3, 3), activation='relu')(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

autoencoder = Model(input_img, decoded)
autoencoder.compile(optimizer='adadelta', loss='binary_crossentropy')

(x_train, _), (x_test, _) = mnist.load_data()

x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
x_train = np.reshape(x_train, (len(x_train), 28, 28, 1))
x_test = np.reshape(x_test, (len(x_test), 28, 28, 1))
autoencoder.fit(x_train, x_train,
                epochs=50,
                batch_size=128,
                shuffle=True,
                validation_data=(x_test, x_test))
```

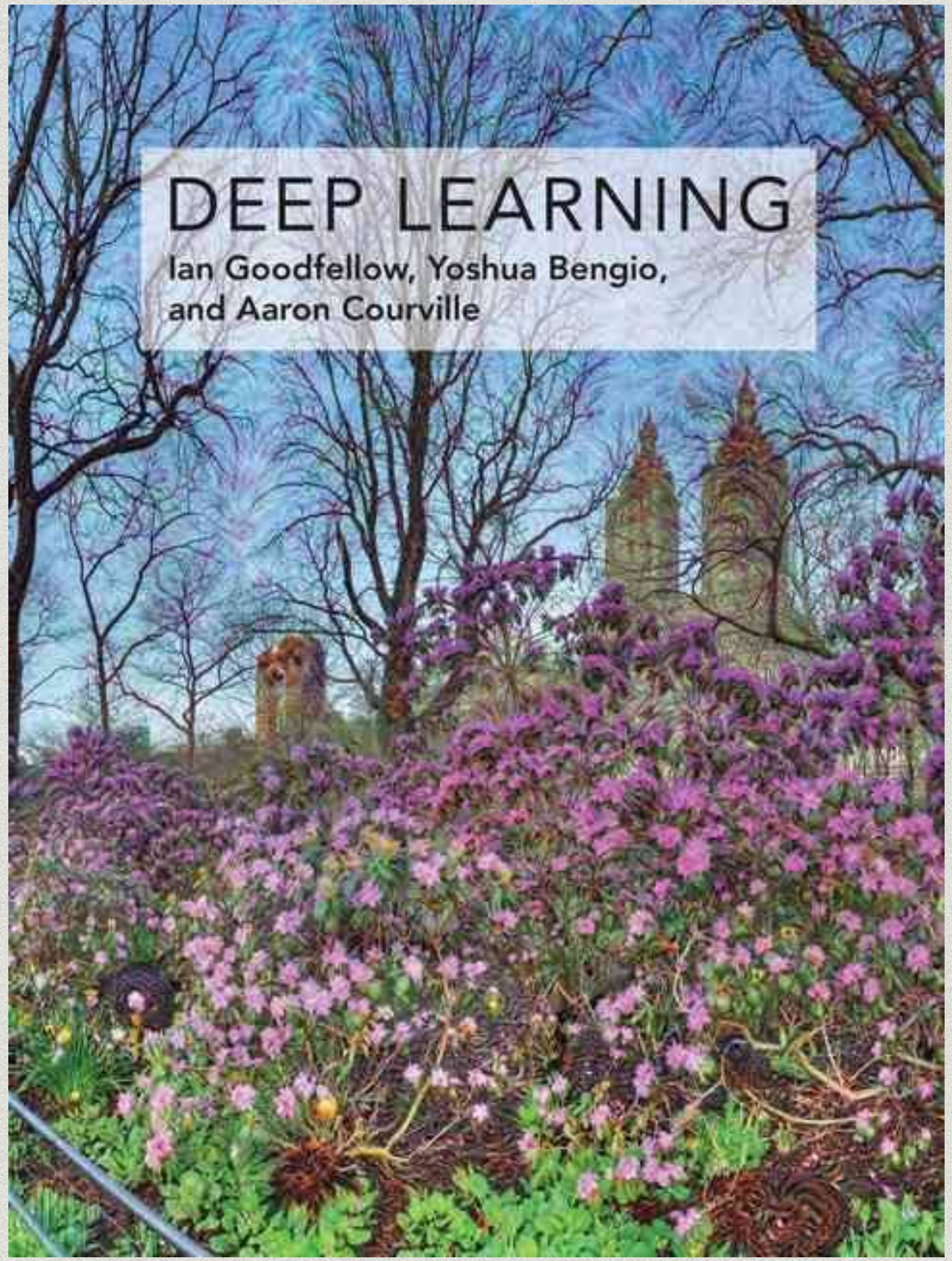
A fully convolutional autoencoder: 40 lines of code



A few lines of python / lua / java and you're off

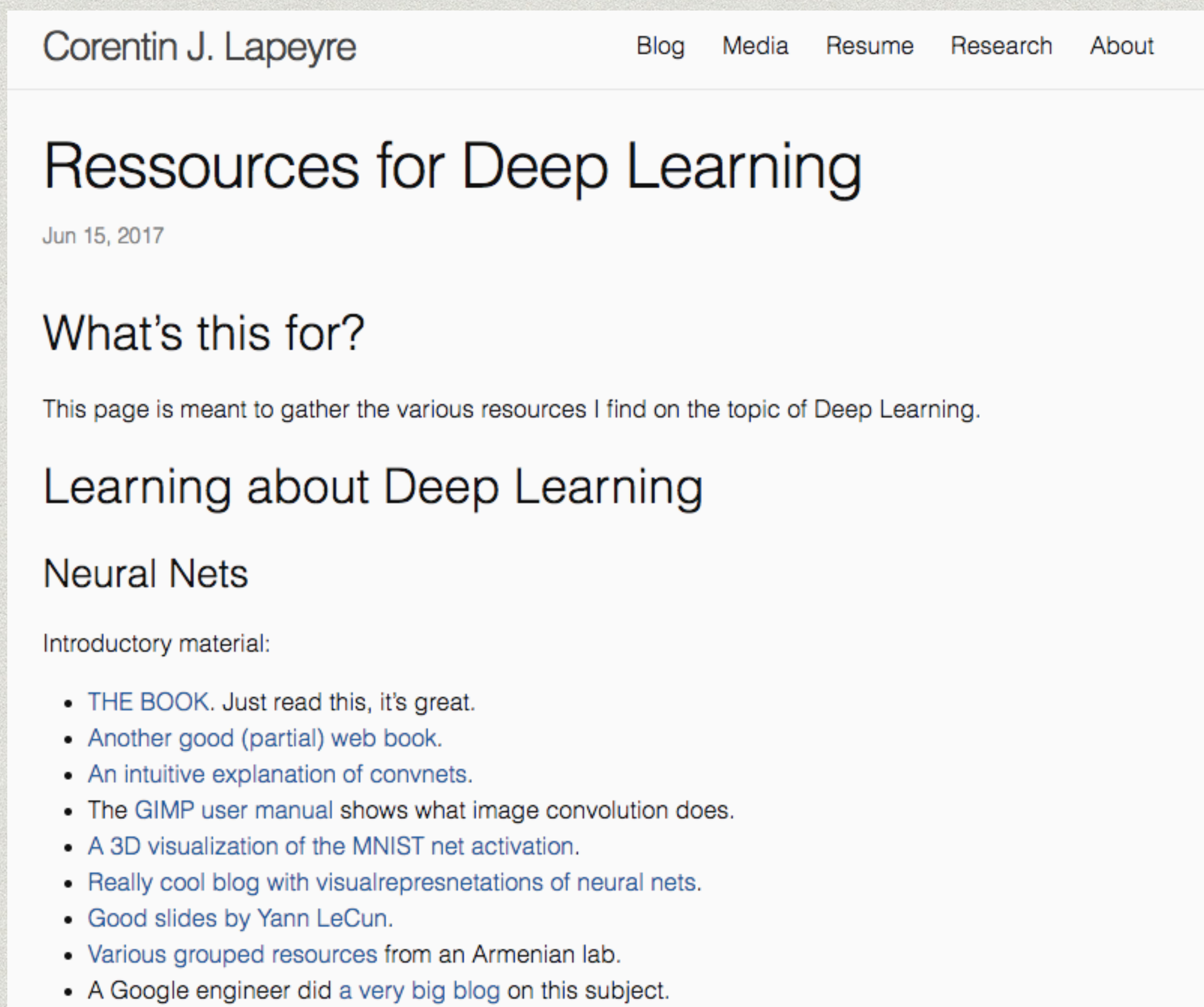
Going further

- I have left out a lot of swear words here: *cross-entropy*, *backpropagation*, *dropout*, *pooling*…
- The web has tremendous amounts of ressources on the subject. Tutorials, web-books, videos…
- This book is in the library → (also online)



Thanks for coming!

For this talk and more ressources, visit :


Corentin J. Lapeyre   Blog   Media   Resume   Research   About

Ressources for Deep Learning

Jun 15, 2017

What's this for?

This page is meant to gather the various resources I find on the topic of Deep Learning.

Learning about Deep Learning

Neural Nets

Introductory material:

- THE BOOK. Just read this, it's great.
- Another good (partial) web book.
- An intuitive explanation of convnets.
- The GIMP user manual shows what image convolution does.
- A 3D visualization of the MNIST net activation.
- Really cool blog with visualrepresnetations of neural nets.
- Good slides by Yann LeCun.
- Various grouped resources from an Armenian lab.
- A Google engineer did a very big blog on this subject.

https://clapeyre.github.io
> Blog
> Ressources for Deep Learning

Or just come to see me and we can talk :)