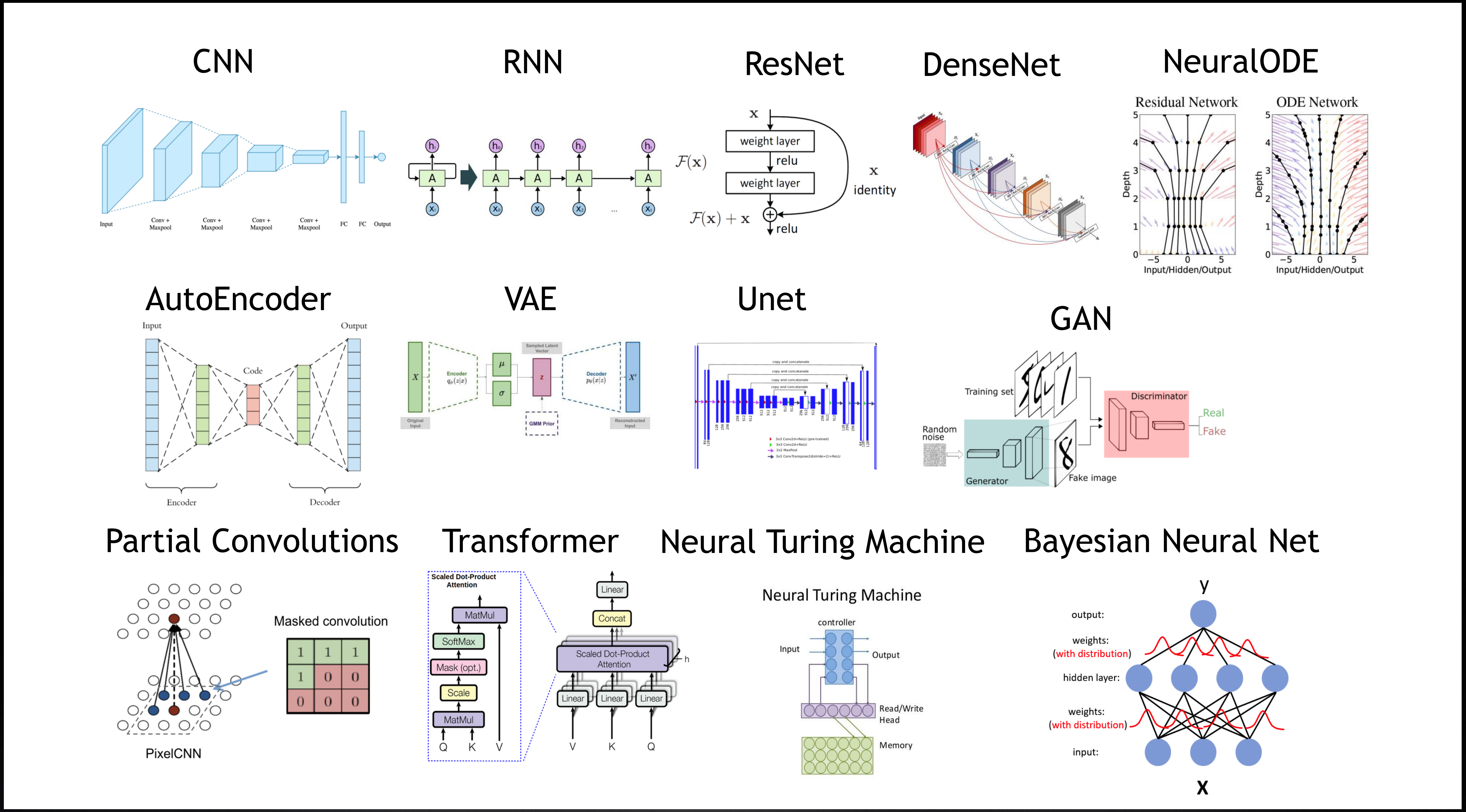
NVIDIA

# DEEP LEARNING ARCHITECTURES

David M. Hall, June 23, 2020

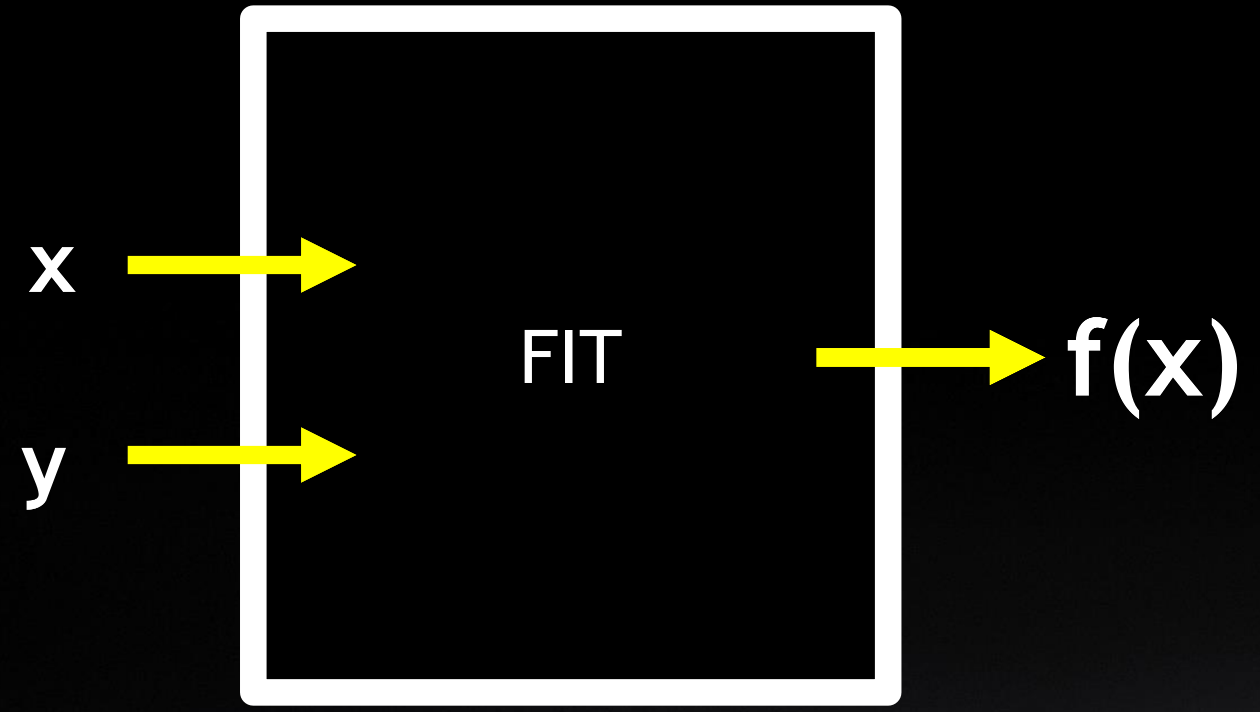
AI4ESS: AI for Earth System Science Workshop and Hackathon

# DEEP LEARNING MODEL ZOO

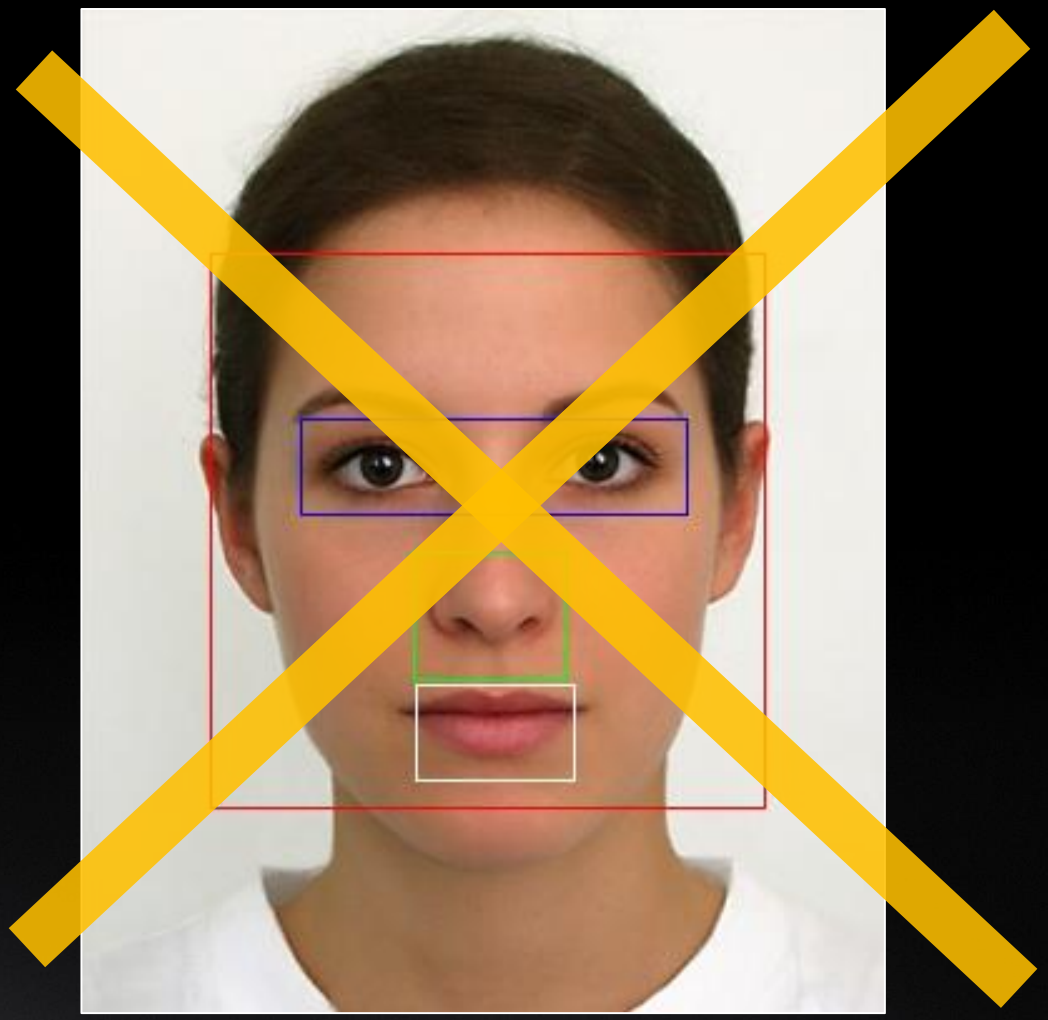


# DEEP LEARNING'S CENTRAL PREMISE

LEARN FUNCTIONS FROM DATA




FEATURE ENGINEERING NOT REQUIRED



# THE MASTER ALGORITHM

## Universal Machine Learning is an Ideal, Not Yet a Reality

<u>TOPIC</u>

*"How to build AGI and to use it safely"*

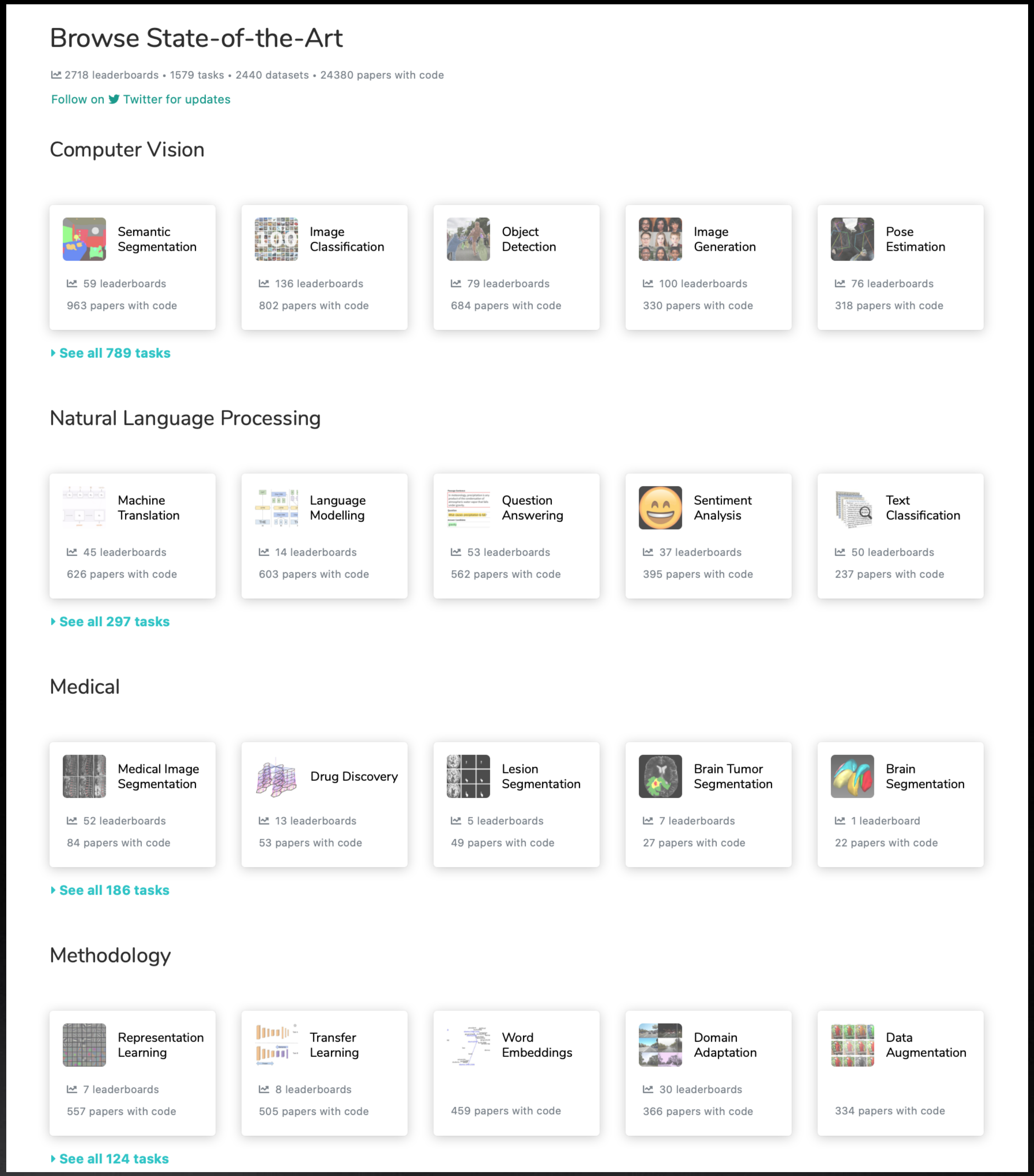
*"How to Solve Global Climate Change"*

<u>DISSERTATION</u>

# TRACKING THE STATE OF THE ART

## Arxiv Sanity Preserver


## Papers With Code




Distill

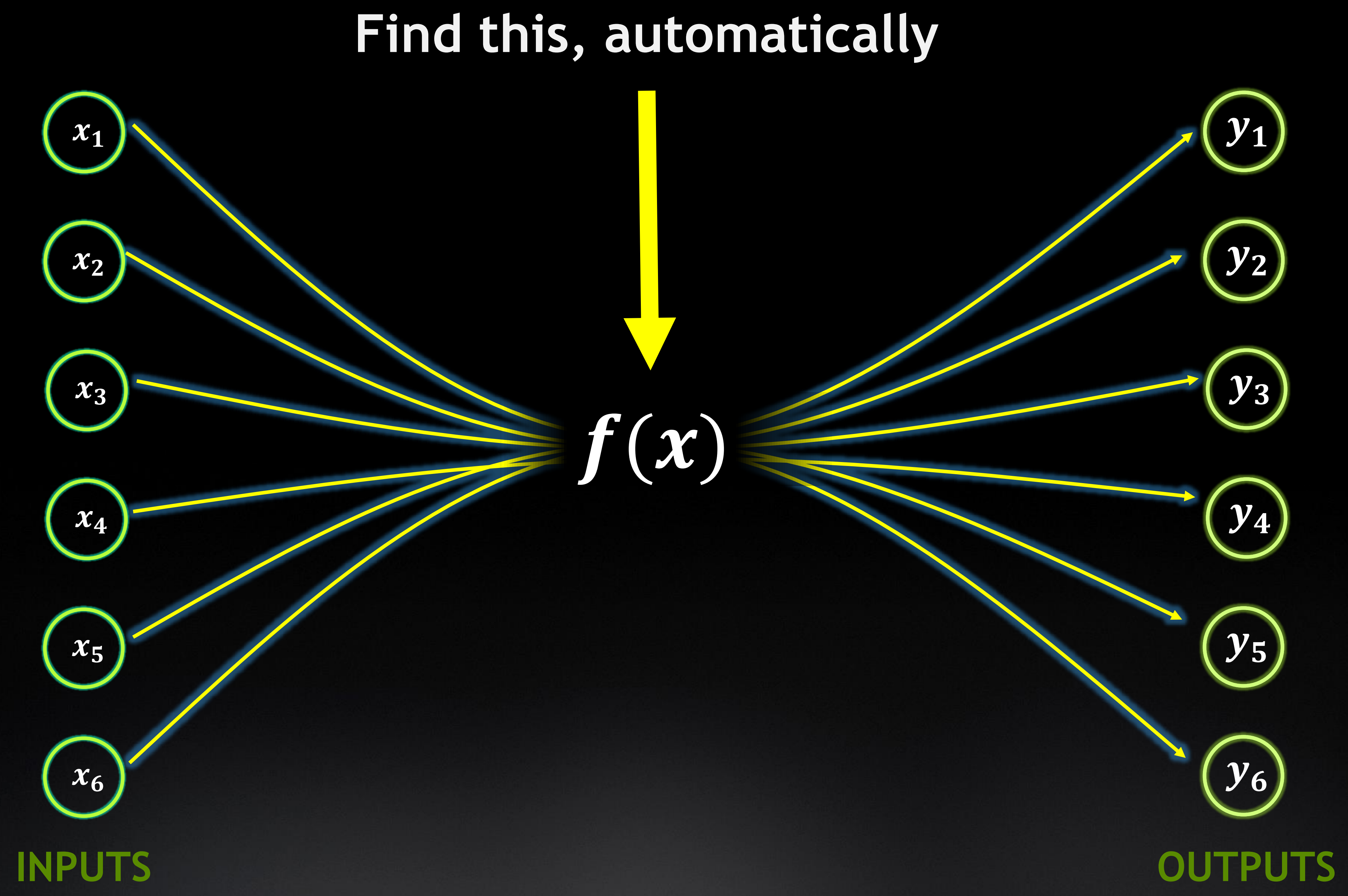
YouTube

# AGENDA

- Deep Learning Basics
- Fully Connected Networks
- CNNs
- ResNets
- Encoder-Decoders
- Masked Convolutions
- Generative Models
- Transformers
- AutoML

# DEEP LEARNING BASICS

# REVERSE-ENGINEER FUNCTIONS FROM EXAMPLES



# IT'S A NEW WAY TO BUILD SOFTWARE

TEMP, PRESSURE, MOISTURE


PROBABILITY OF RAIN

## HAND-WRITTEN FUNCTION

```
Function1(T,P,Q)

update_mass()

update_momentum()

update_energy()

do_macrophysics()

do_microphysics()

y = get_precipitation()

return y
```

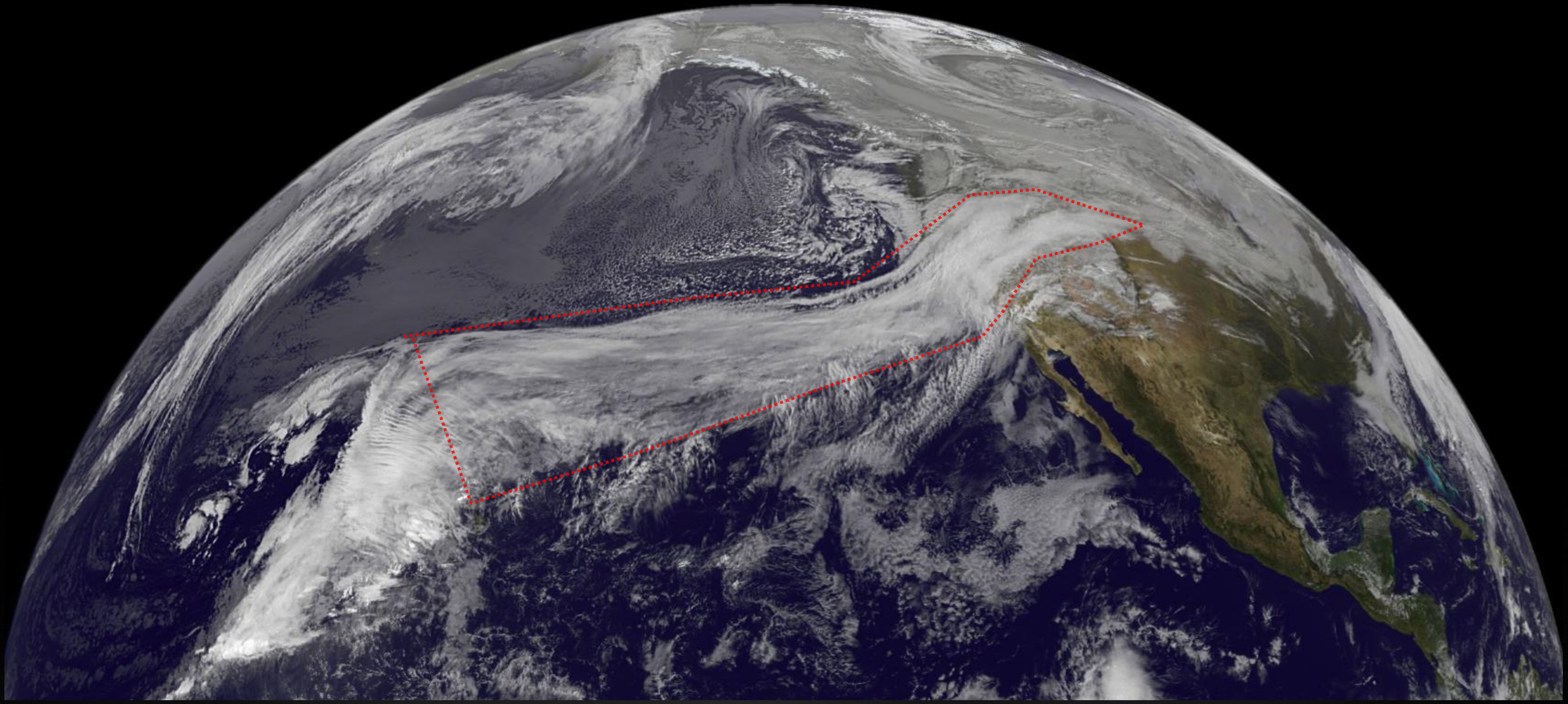
Convert expert
knowledge into a function

## LEARNED FUNCTION

```
Function1(T,P,Q)

A = relu( w1 * [T,P,Q] + b1)

B = relu( w2 * A       + b2)

C = relu( w3 * B       + b3)

D = relu( w4 * C       + b4)

E = relu( w5 * D       + b5)

y = sigmoid(w6 * E     + b6)

return y
```

Reverse-engineer a function
from inputs / outputs

# COMPLEX PHENOMENA ARE BEST DESCRIBED IMPLICTLY

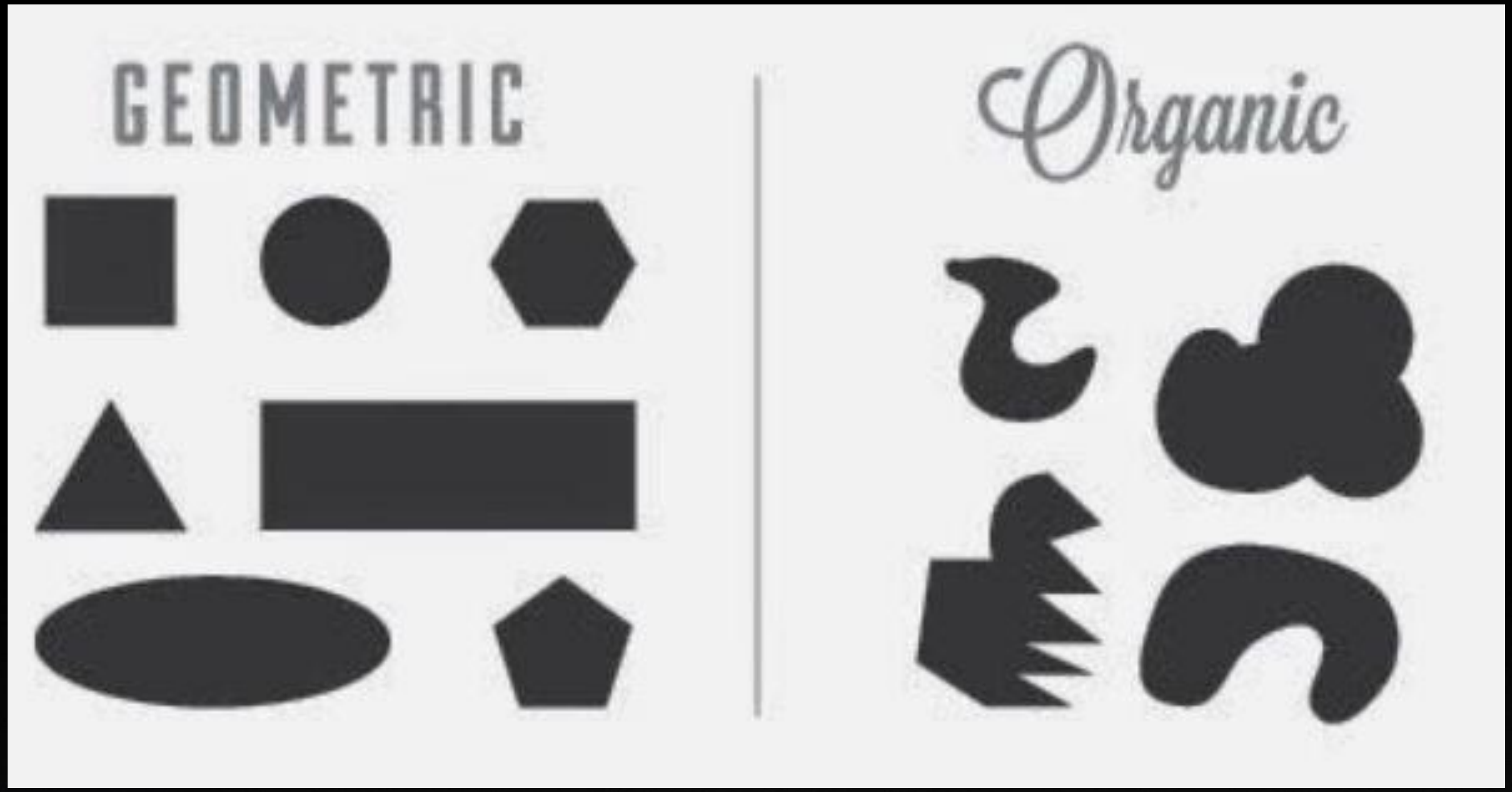


EXAMPLE: ATMOSPHERIC RIVER

# FORWARD AND REVERSE ENGINEERING ARE COMPLIMENTARY

**SOFTWARE DEVELOPMENT**

ENGINEERED
PROGRAMMED
LABOR INTENSIVE
EXPLICIT
EXPLAINABLE
HEURISTIC
SIMPLE
FROM EXPERTISE

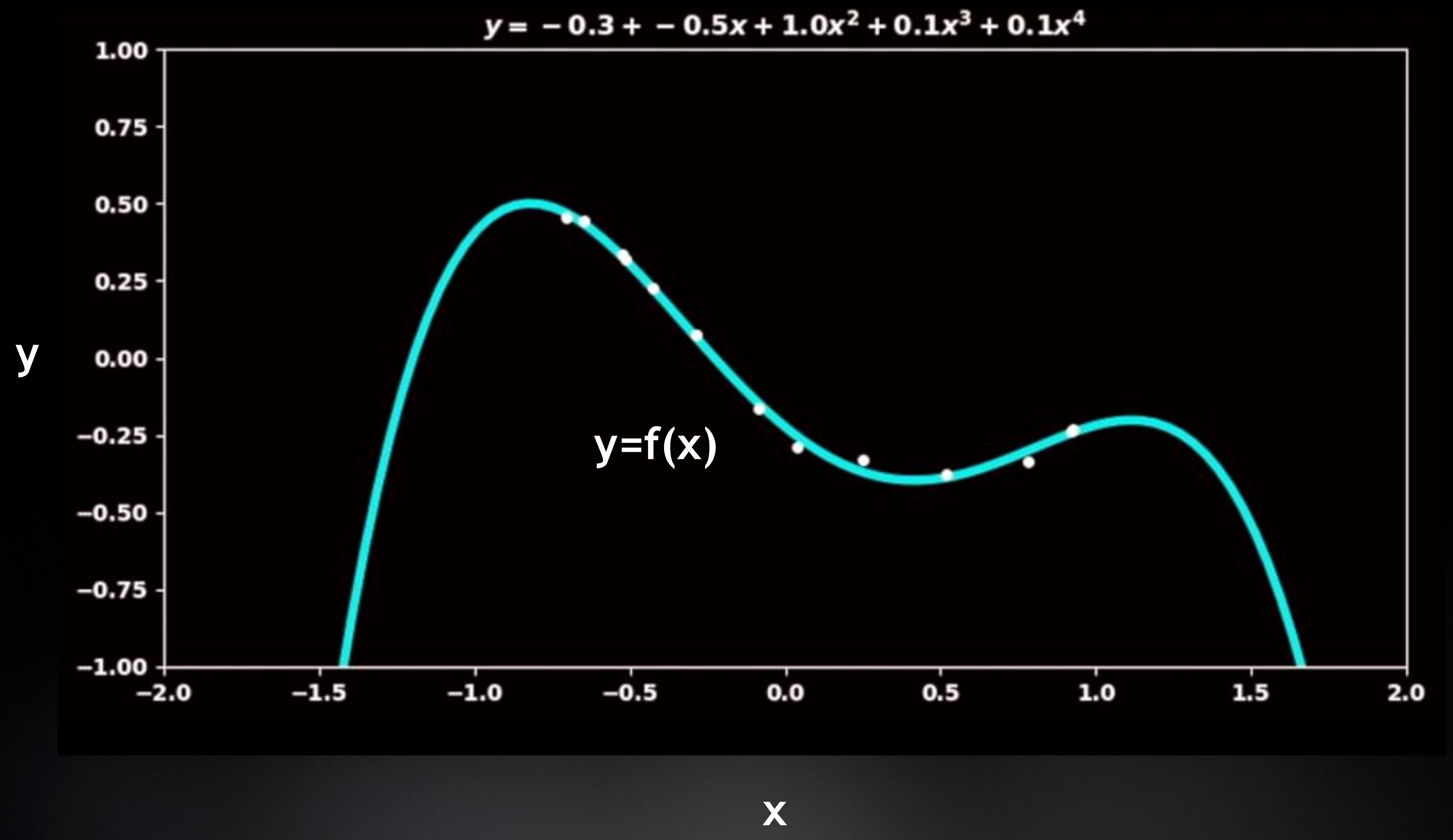



**MACHINE LEARNING**

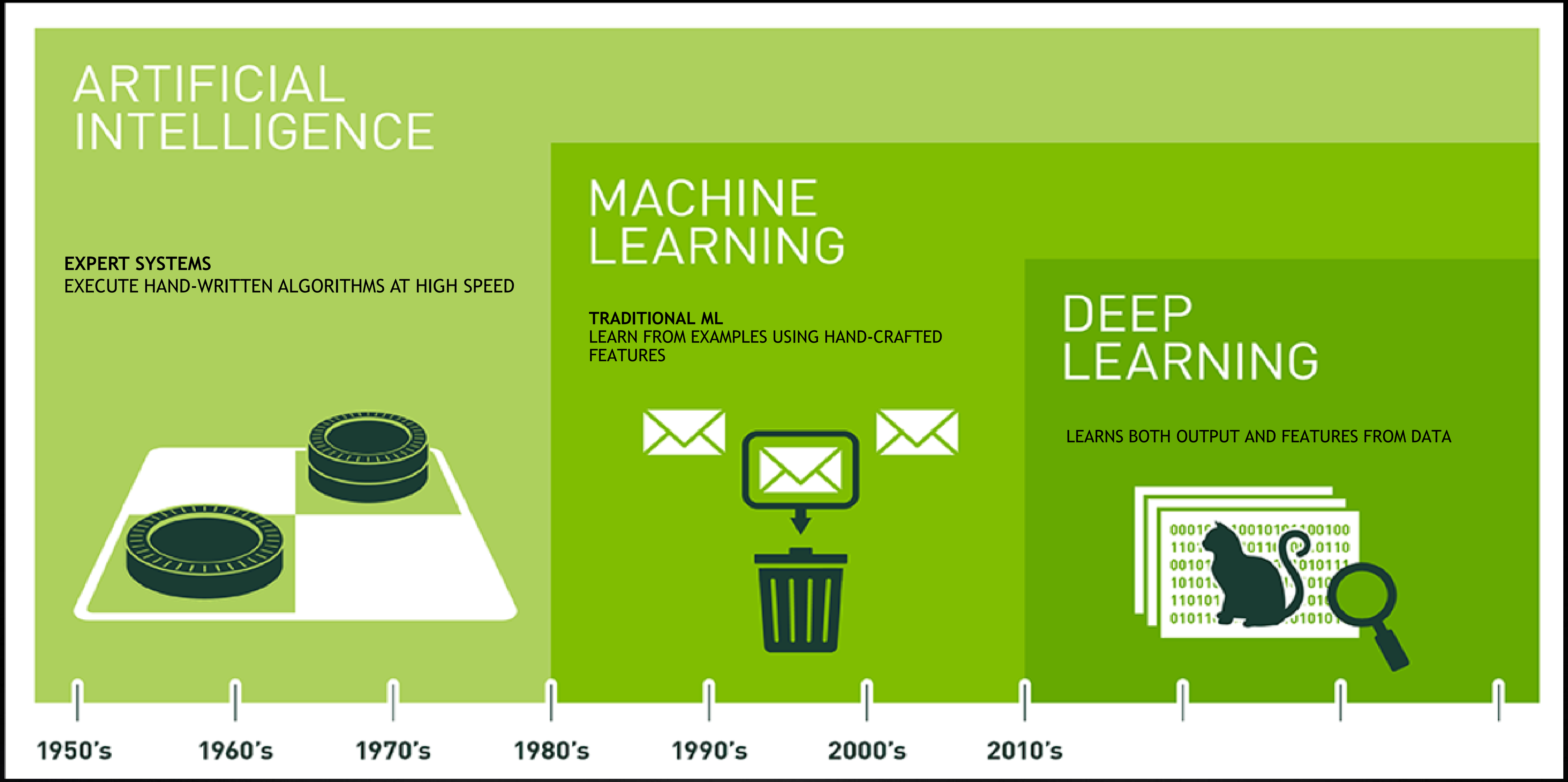
REVERSE ENGINEERED
LEARNED
AUTOMATIC
IMPLICIT
SUBTLE
REALISTIC
COMPLEX
FROM EXAMPLES

For best results, combine them

# MACHINE LEARNING IS CURVE FITTING, GENERALIZED



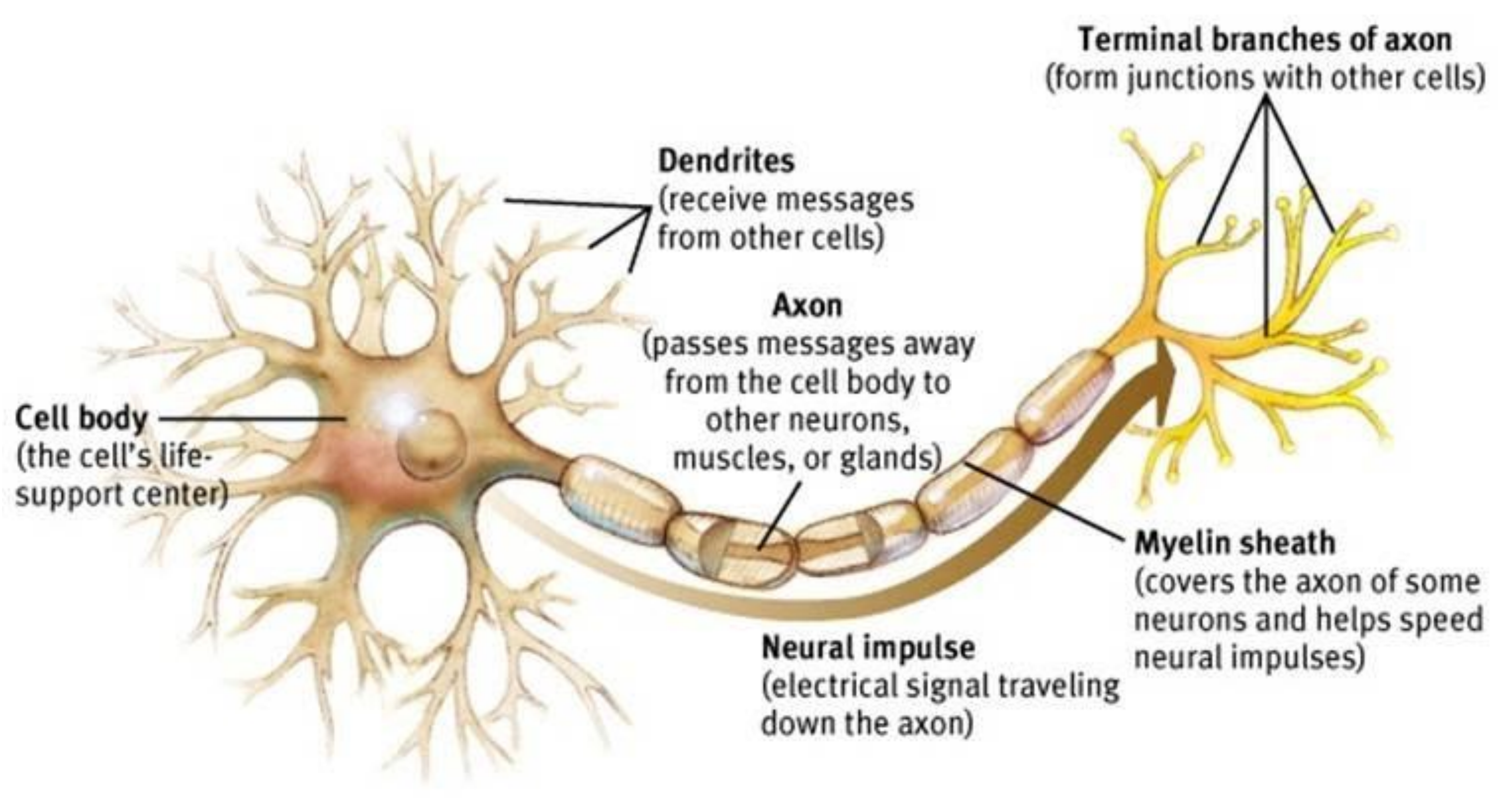
# AI, MACHINE LEARNING, DEEP LEARNING



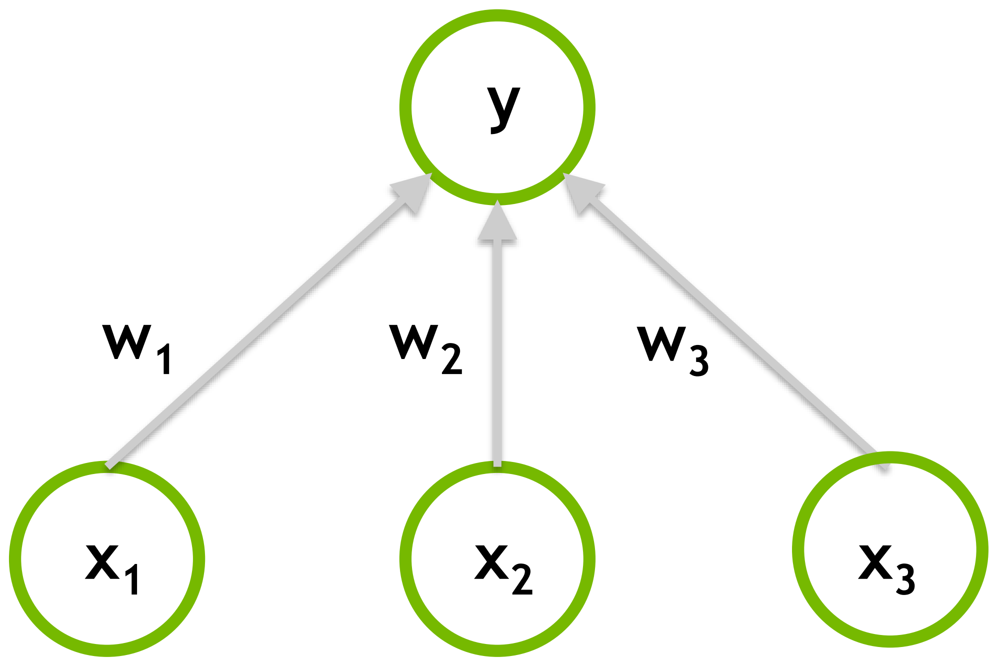
# ARTIFICIAL NEURONS

## Are simple equations with a set of adjustable parameters

**Biological neuron**




**Artificial neuron**

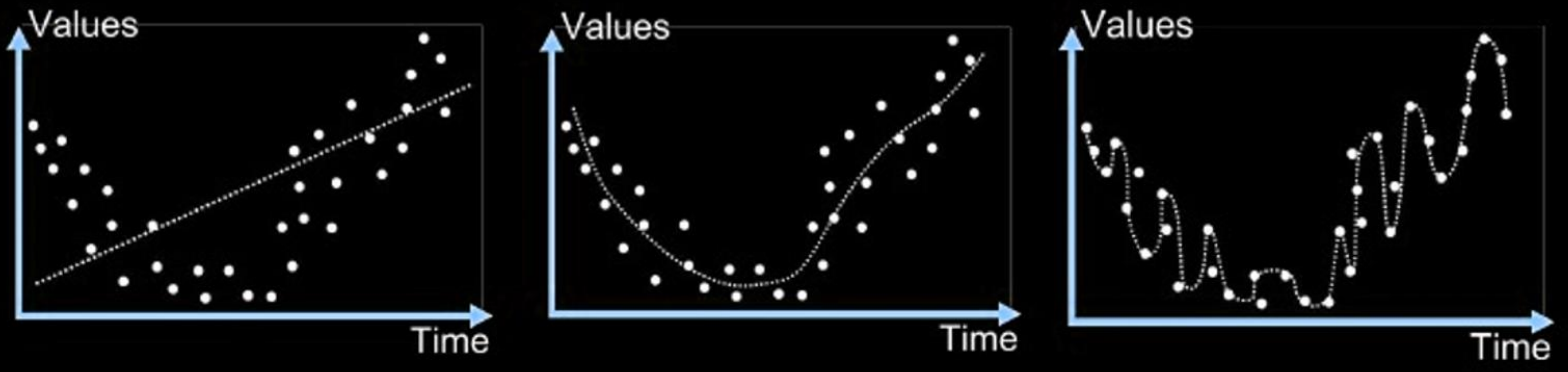



$$y=f(w_1x_1+w_2x_2+w_3x_3)$$

https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7

# ADJUST MODEL CAPACITY TO FIT YOUR DATA

A good model is one that generalizes to new data



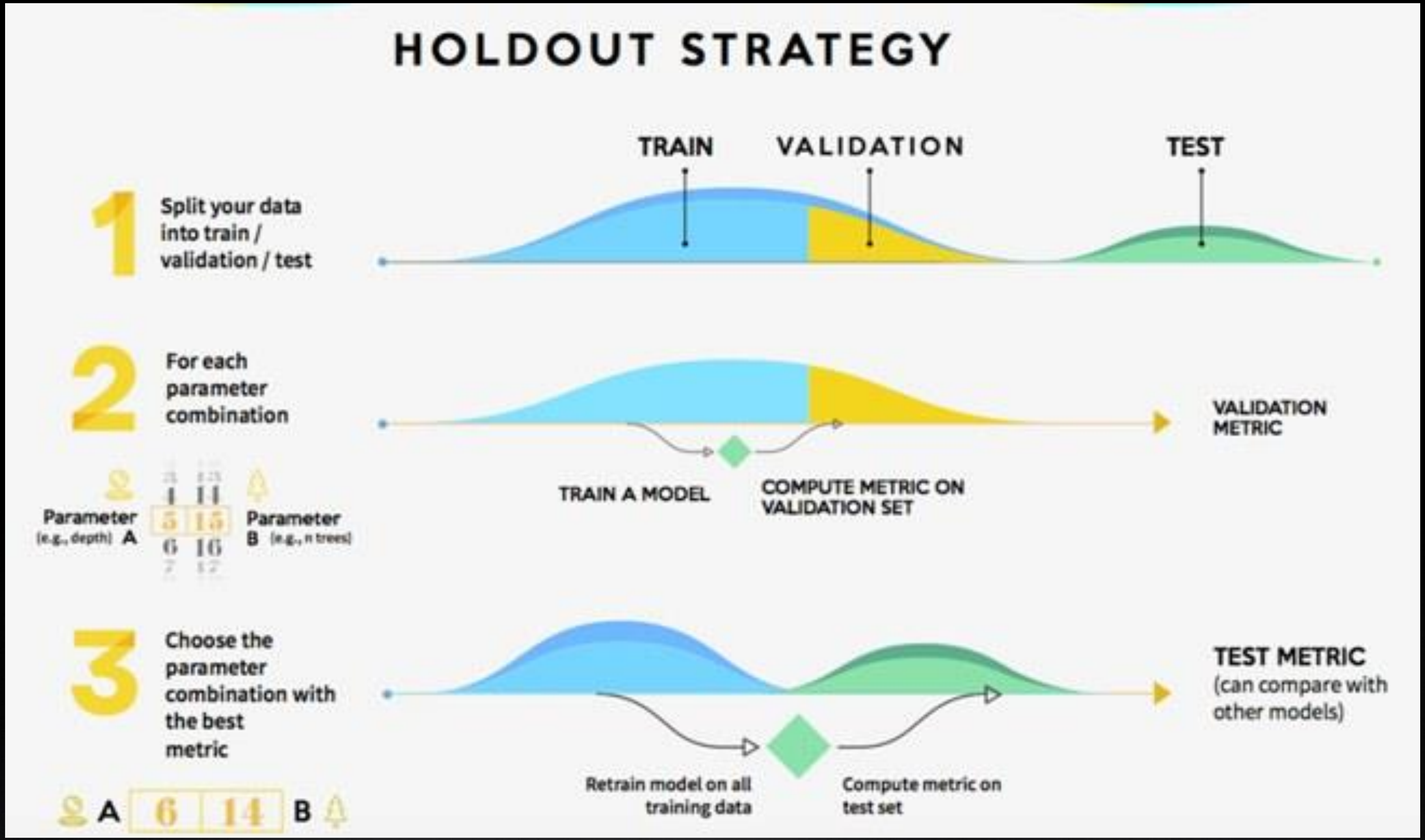

UNDERFIT

GOOD FIT

OVER FIT

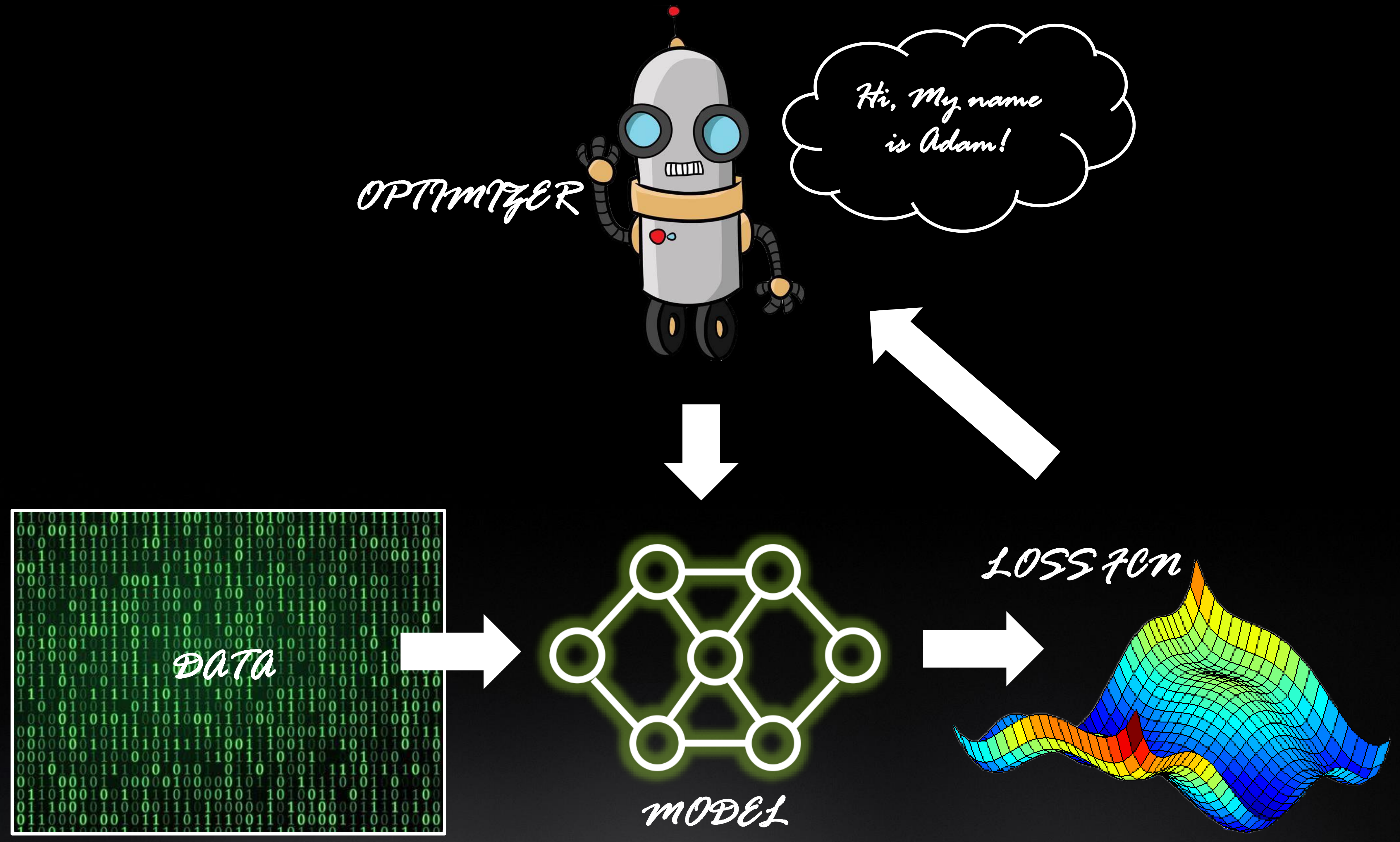
# KEEP TRAINING AND TEST DATA SEPERATE

## KEEP TEST, TRAINING, AND VALIDATION DATA SEPERATE



# TRAINING

## DATA, MODEL, LOSS, AND OPTIMIZER



# TRAINING: SEARCHING FOR A GOOD SOLUTION

Model training is a form of search, performed by the optimizer.

**Adjust W1,W2 to minimize loss**

**COULD USE:**

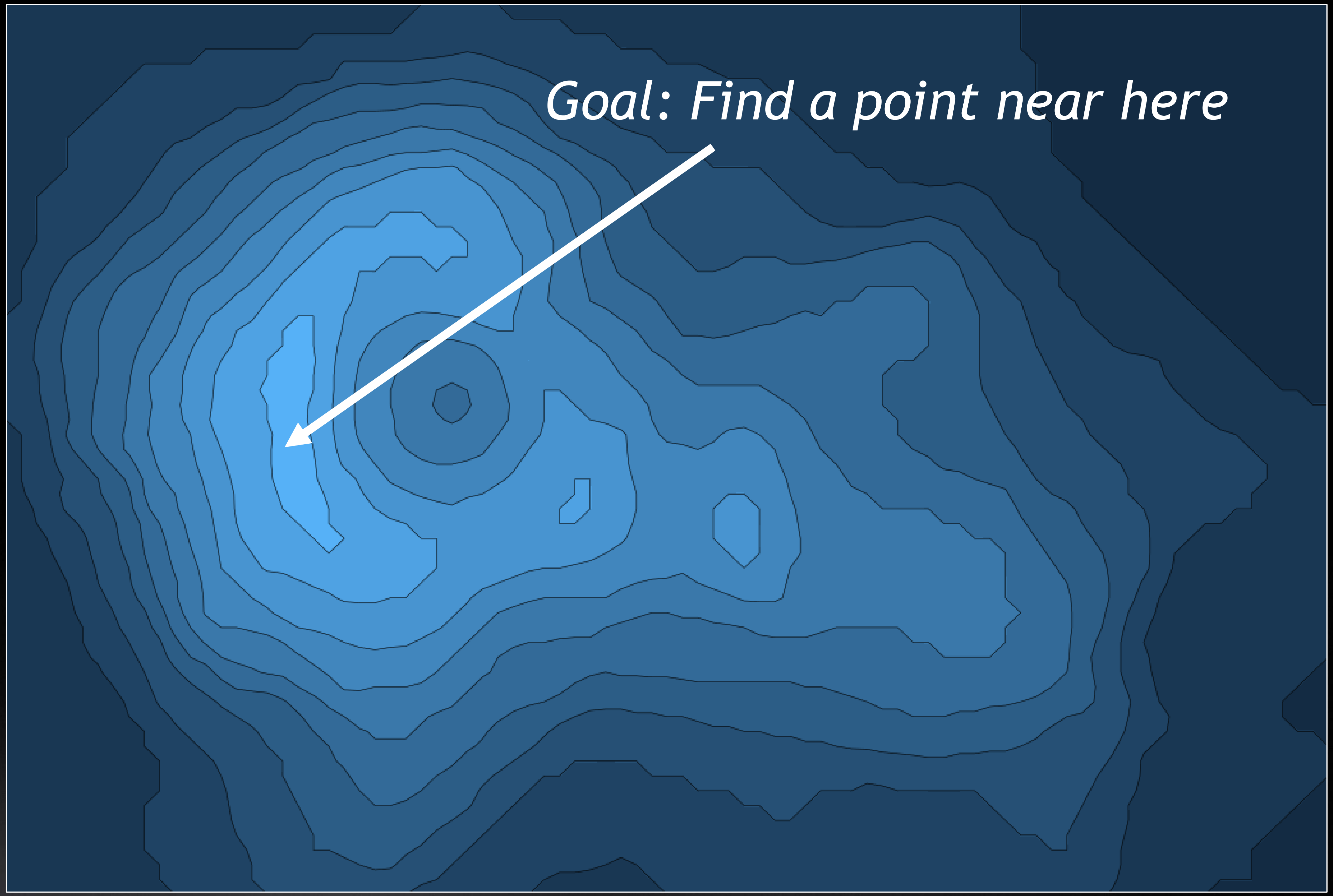
- Grid Search
- Evolutionary Algorithms
- Conjugate Gradient
- Newton's method
- Other 2nd order methods

**ACTUALLY USE:**

- Gradient Descent

LOSS

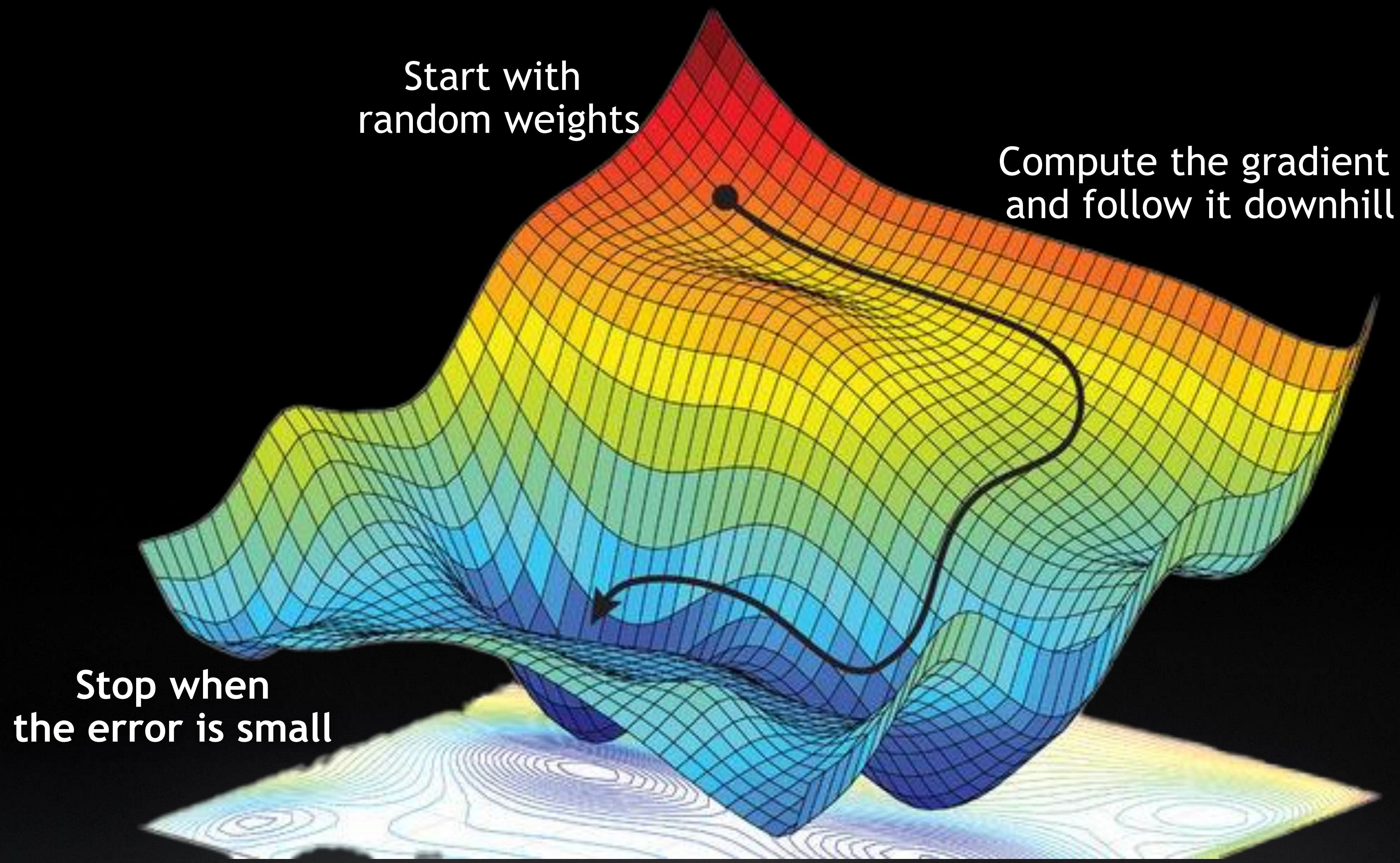
$W_2$




$W_1$

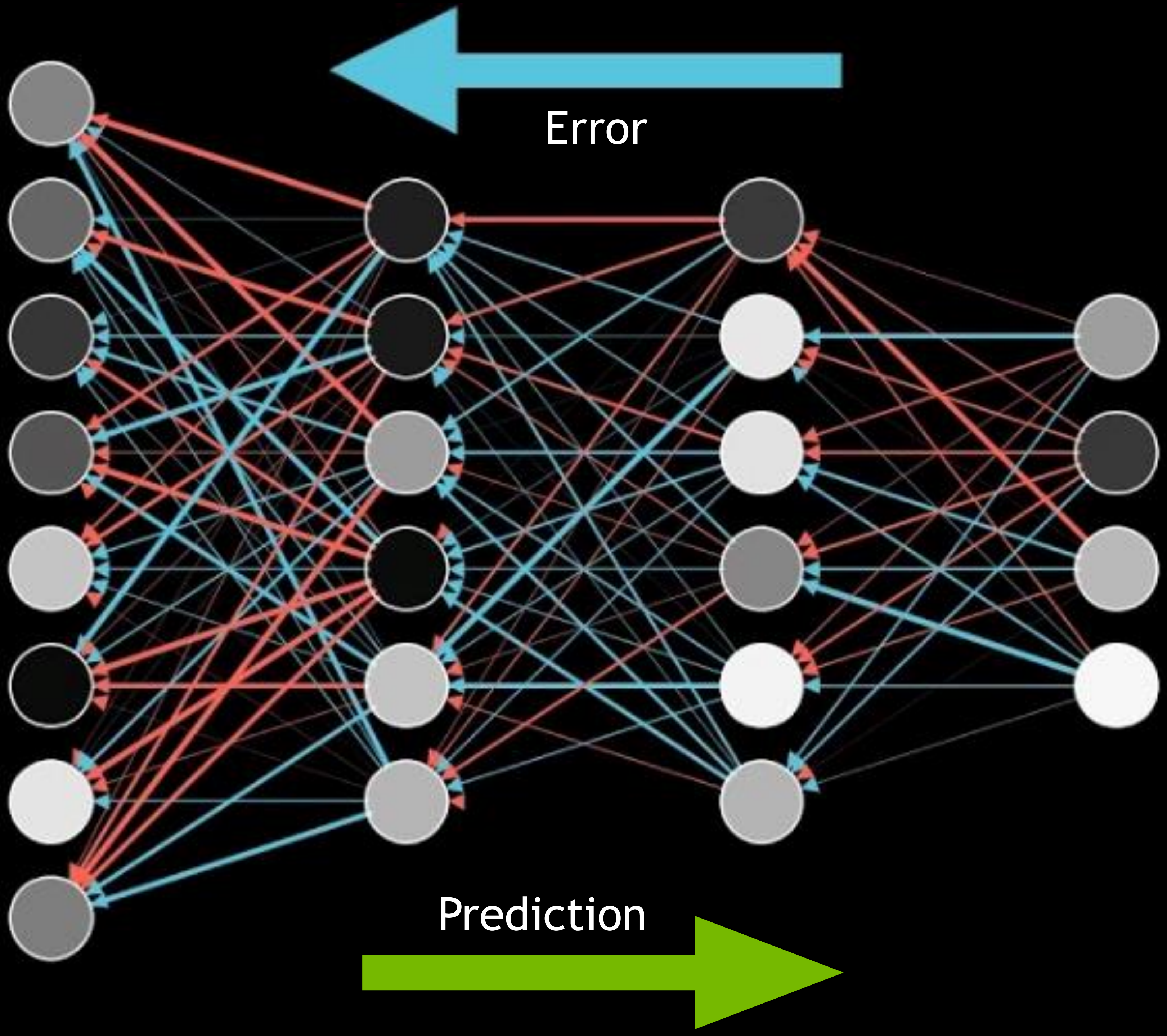
# GRADIENT DESCENT

Finding as solution is as easy as falling down a hill



# BACKPROPAGATION

Compute the gradient, by efficiently assigning blame

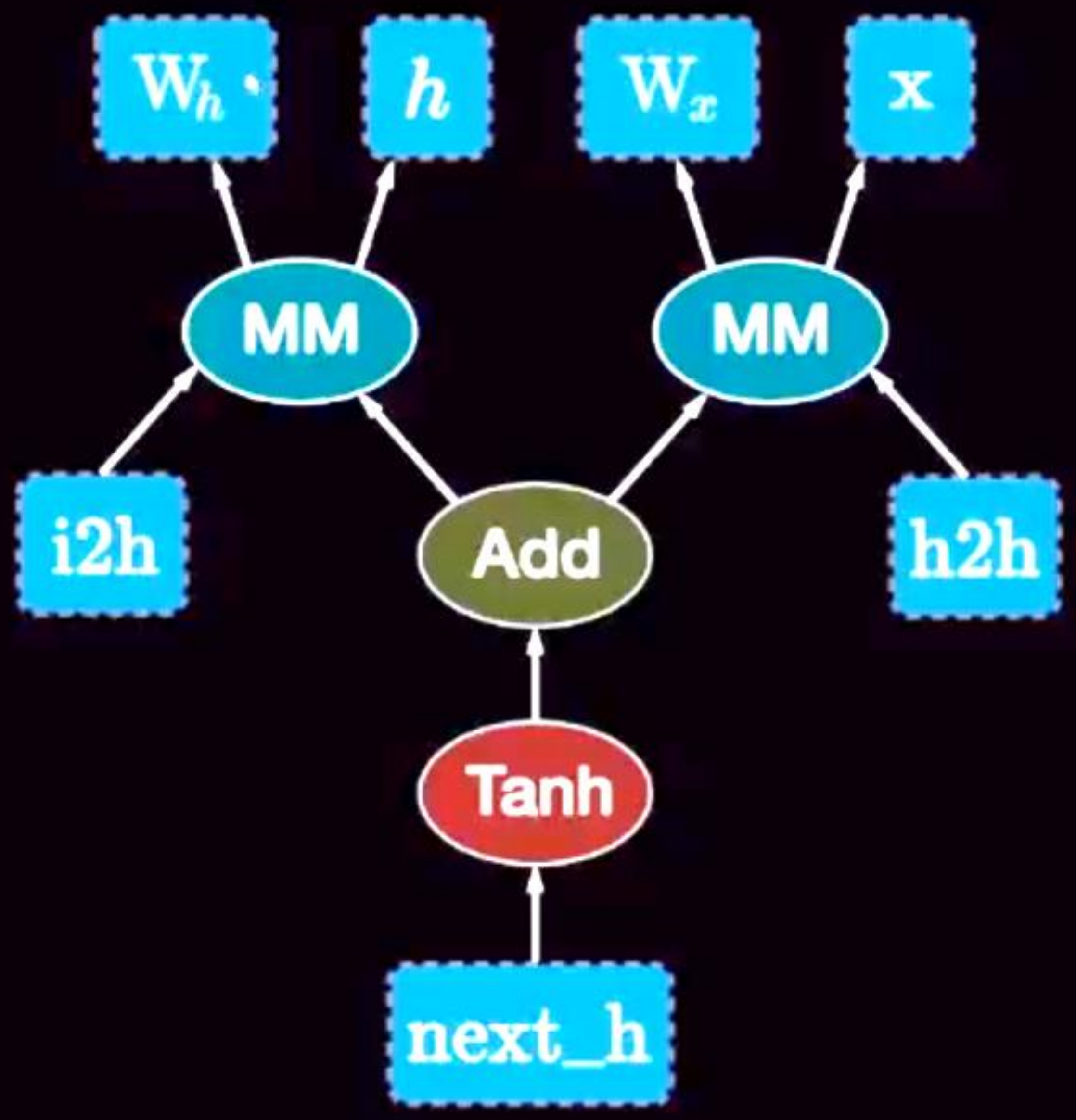


# AUTOGRAD

Let a framework keep track of your gradient, so you don’t have to

## PyTorch Autograd

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))

i2h = torch.mm(W_x, x.t())
h2h = torch.mm(W_h, prev_h.t())
next_h = i2h + h2h
next_h = next_h.tanh()

next_h.backward(torch.ones(1, 20))
```

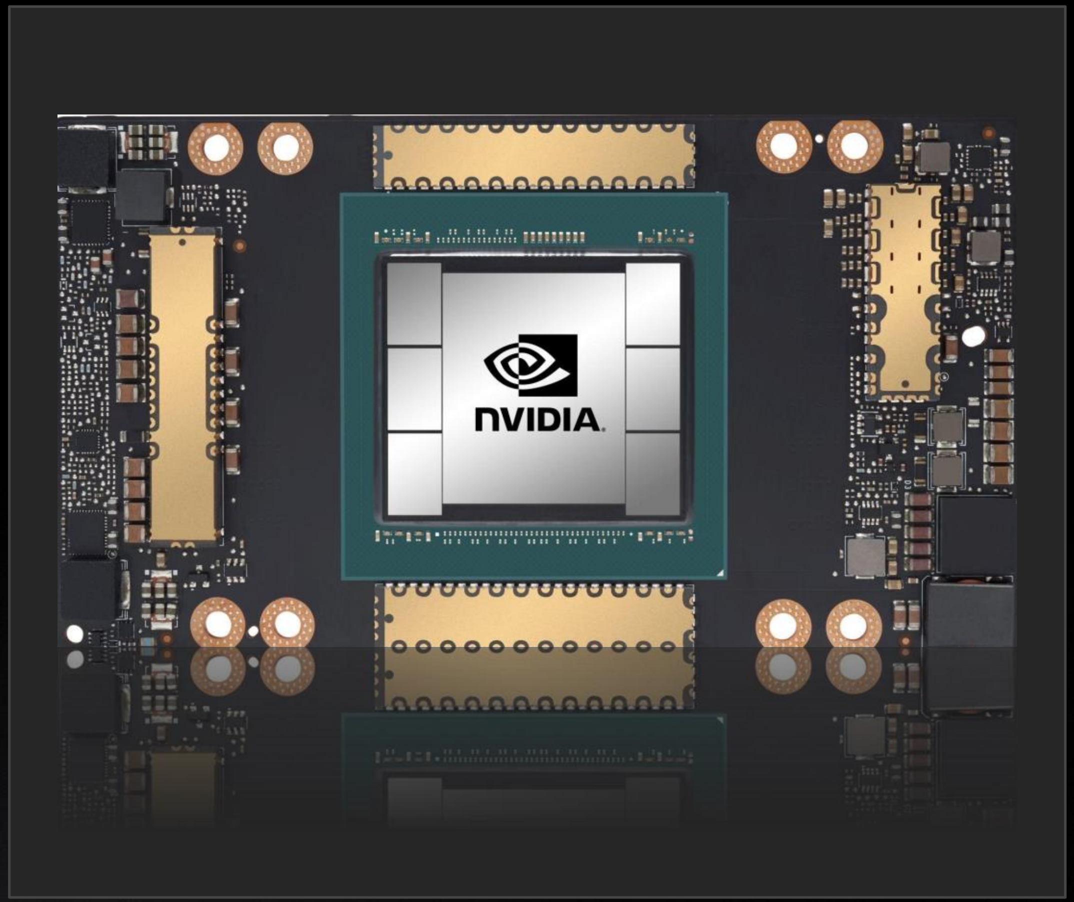


# WHAT YOU NEED TO MAKE IT WORK

You need three main ingredients (and some skill)

LARGE QUANTITIES OF DATA


ML FRAMEWORK




GPU ACCELERATOR

# DEEP LEARNING FRAMEWORK

Many frameworks to choose from (but not Fortran)

PyTorch
K + TensorFlow

**Python**

mxnet
GLUON

**C++**

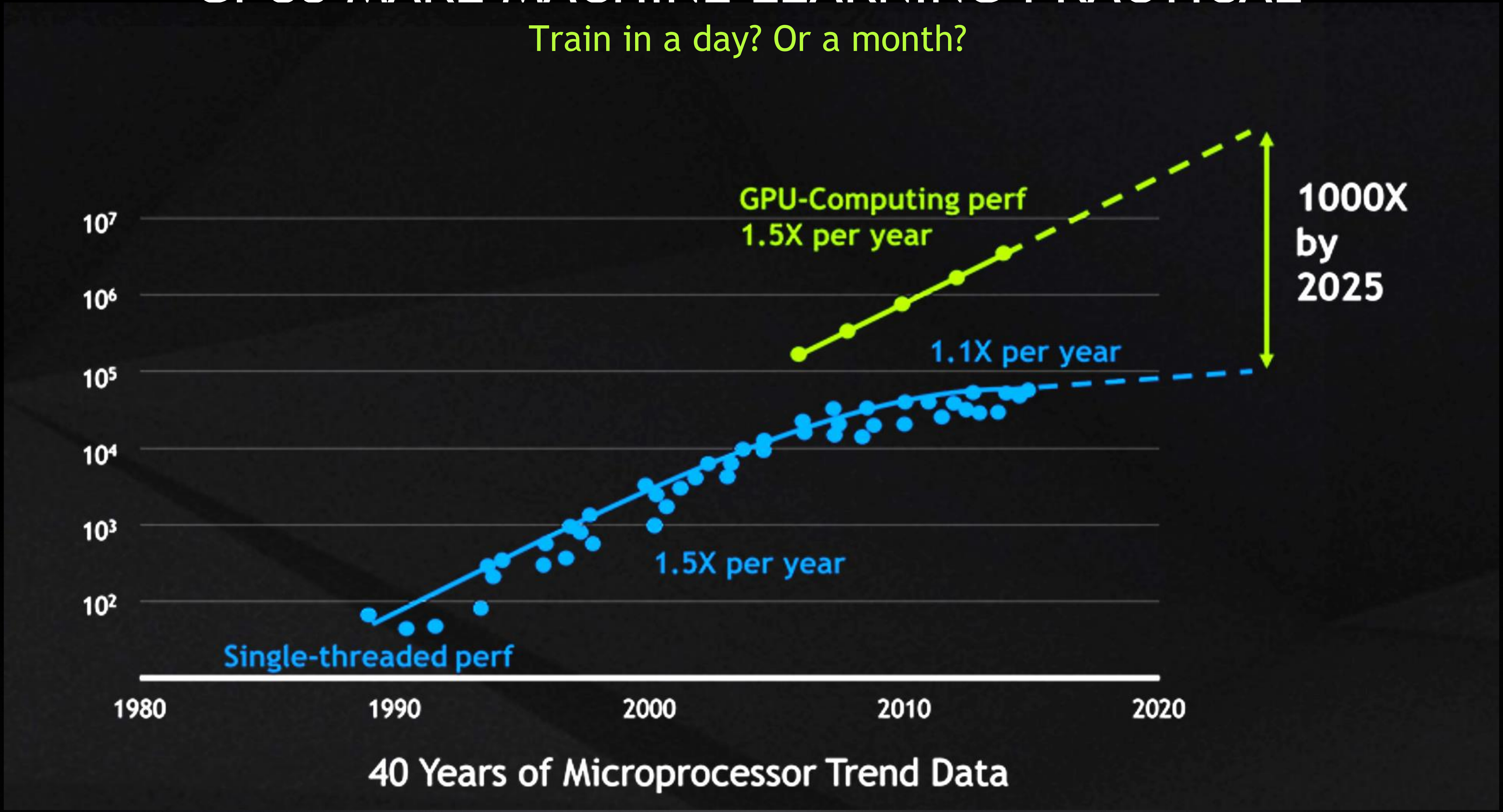
flux

**Julia**

Chainer · mxnet · Caffe2 · Microsoft CNTK · TensorFlow · Keras · GLUON · PYTORCH · theano

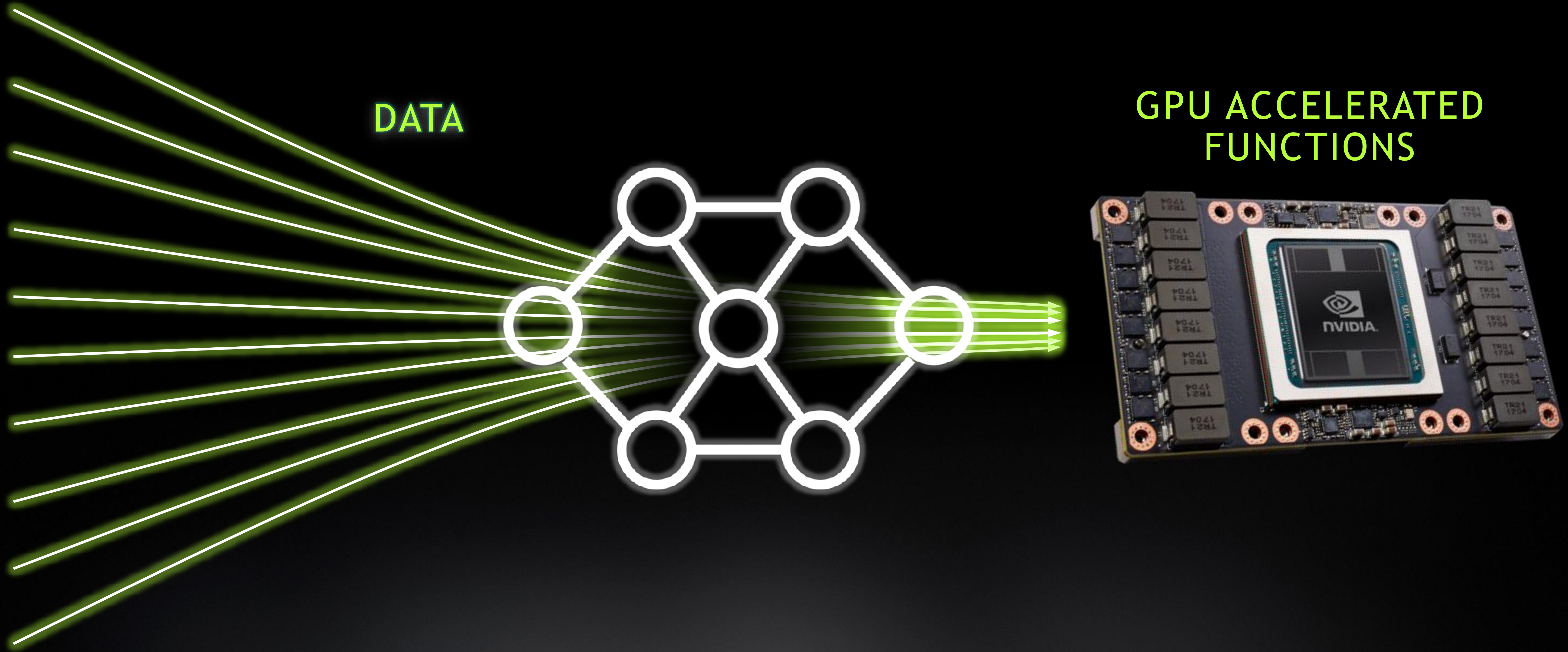
# GPUS MAKE MACHINE LEARNING PRACTICAL

Train in a day? Or a month?



# LEARNED FUNCTIONS ARE GPU ACCELERATED

Next level software. No porting required.



# HOW CAN I GET ACCESS TO A POWERFUL GPU?

## Many way to take advantage of NVIDIA GPUs for Deep Learning

NVIDIA Quadro
Laptop or Workstation

IBM Cloud
Google Cloud
aws
Microsoft Azure

Cloud Computing Services
(Free hours to start)


National Supercomputers
(Apply for compute)

colab

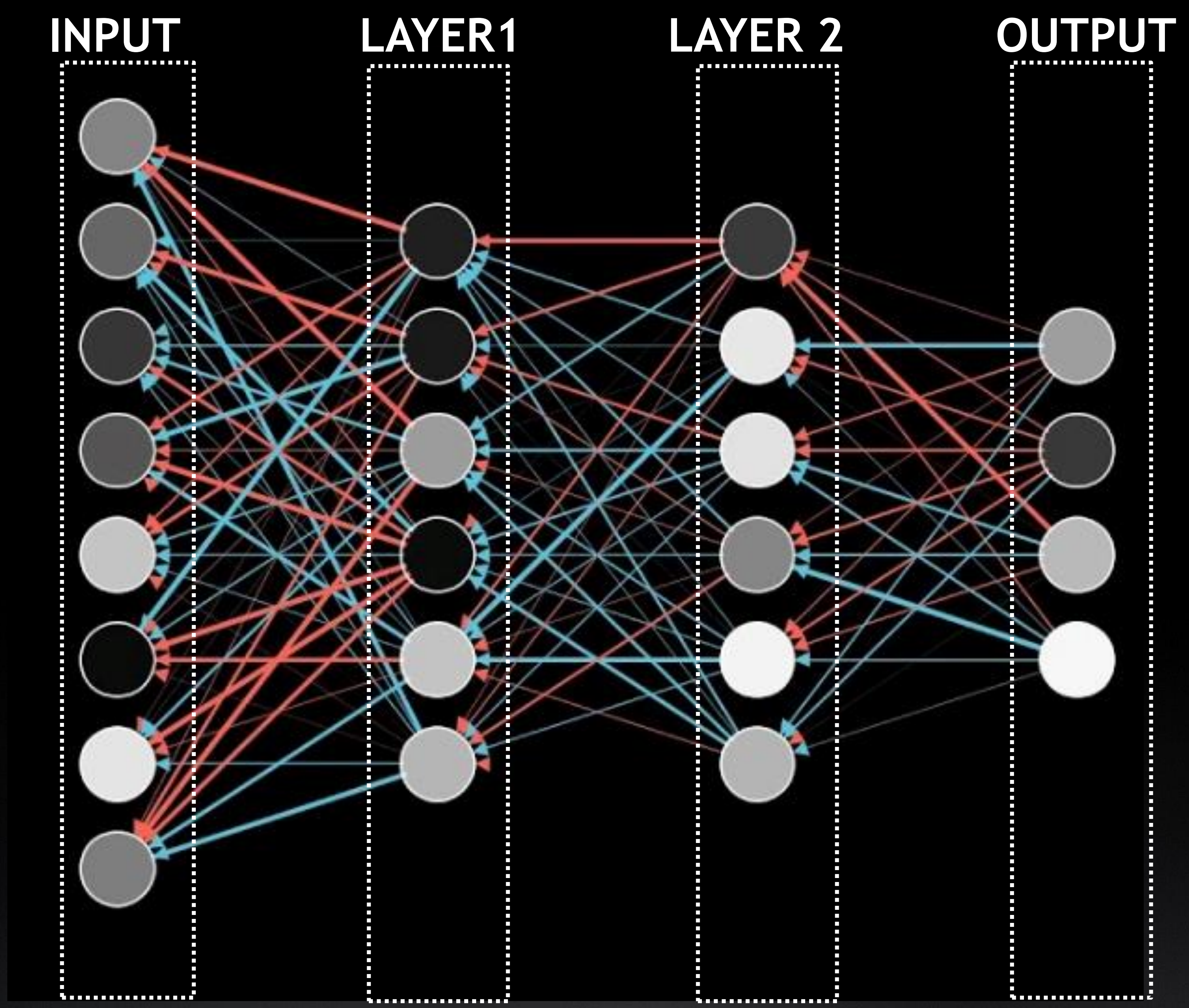
Google Colab
(1 Free NVIDIA GPU)

# FULLY CONNECTED NETWORKS

(MULTI-LAYER PERCPTRONS)

# FULLY CONNECTED NETWORKS

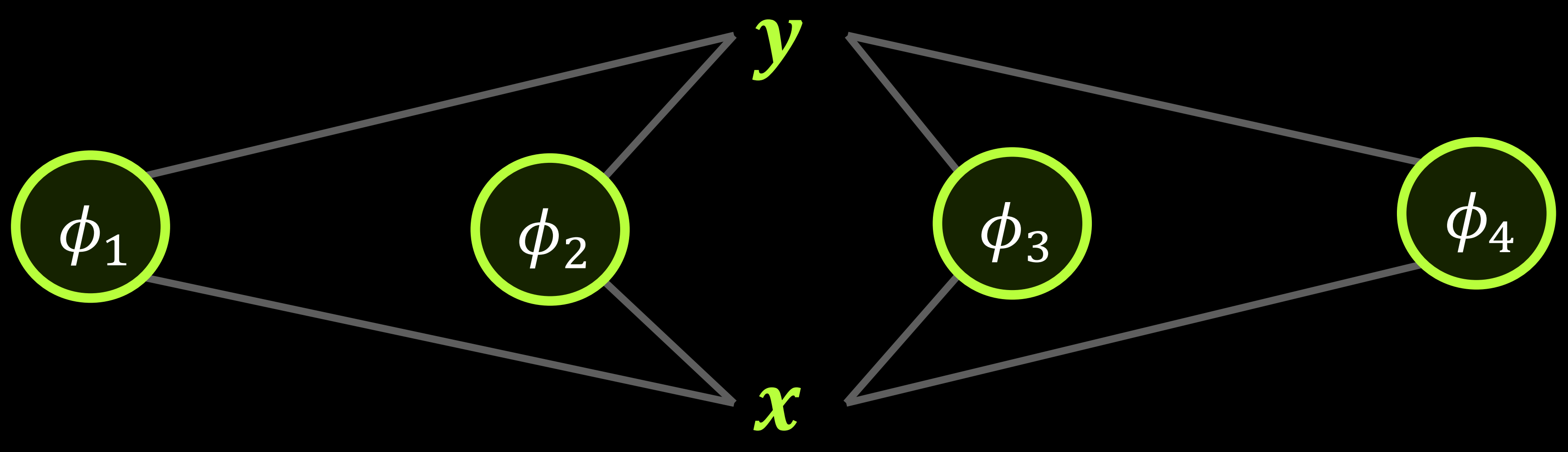
A given neuron is connected to every neuron in the previous layer



# SINGLE LAYER NEURAL NETWORKS

A series expansion over basis functions $\phi$.

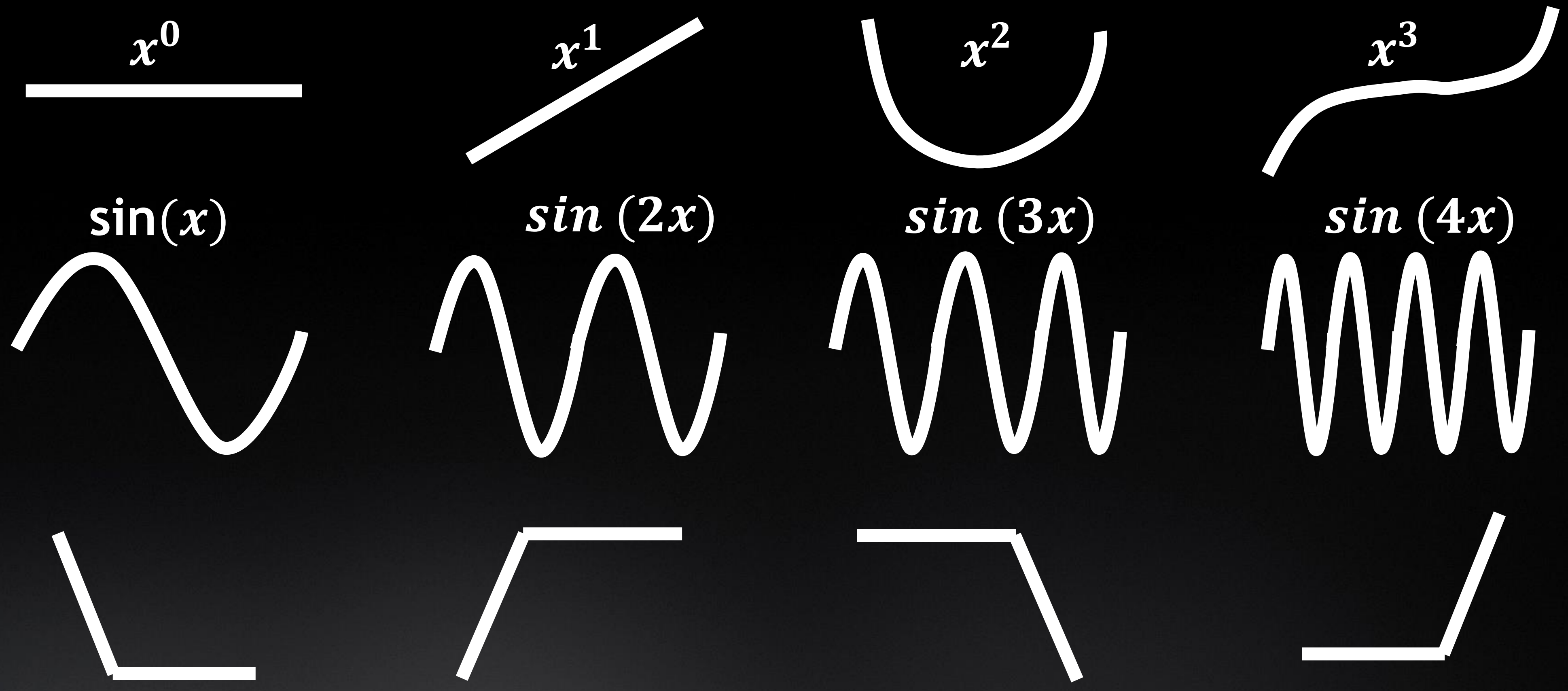
$$y = \sum_i w_i \, \phi_i(x + b_i)$$




TAYLOR SERIES

FOURIER SERIES

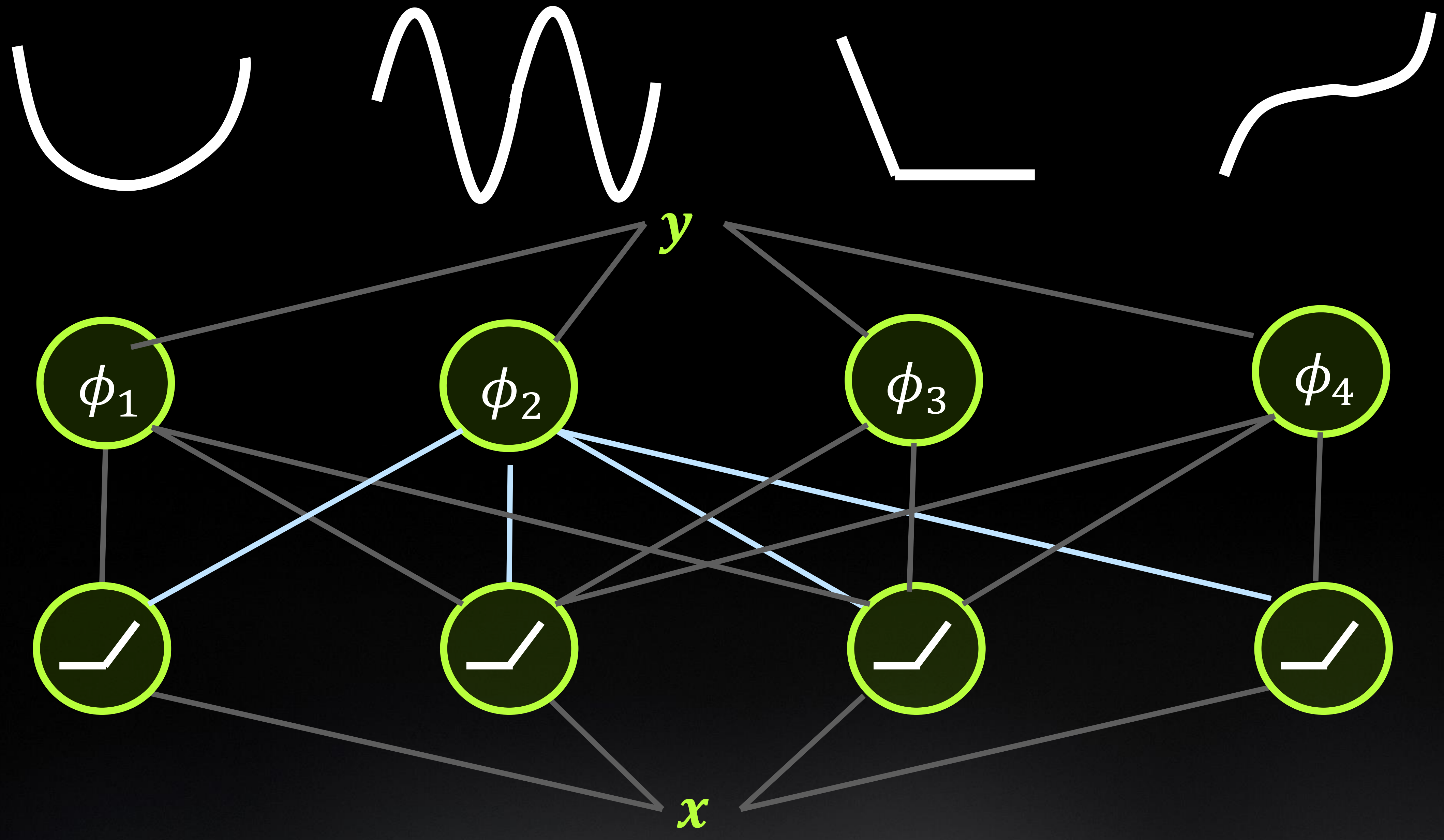
RELU




$(x > 0)\ ?\ x\ :\ 0$

# TWO LAYER NEURAL NETWORKS

Learn the function and the basis functions at the same time



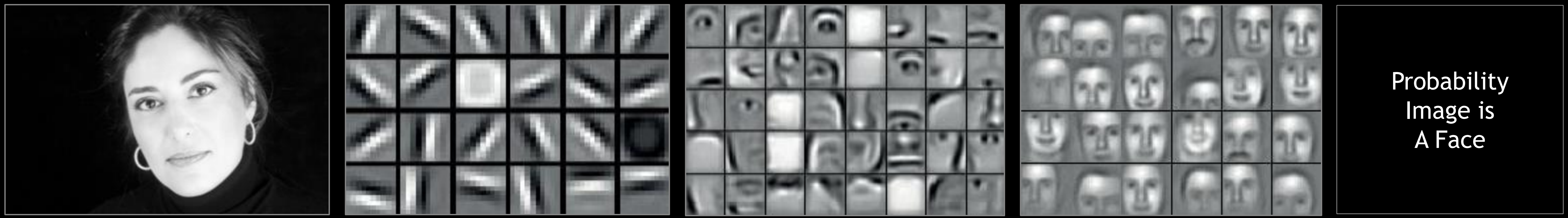

L2: LEARNED BASIS FCNS

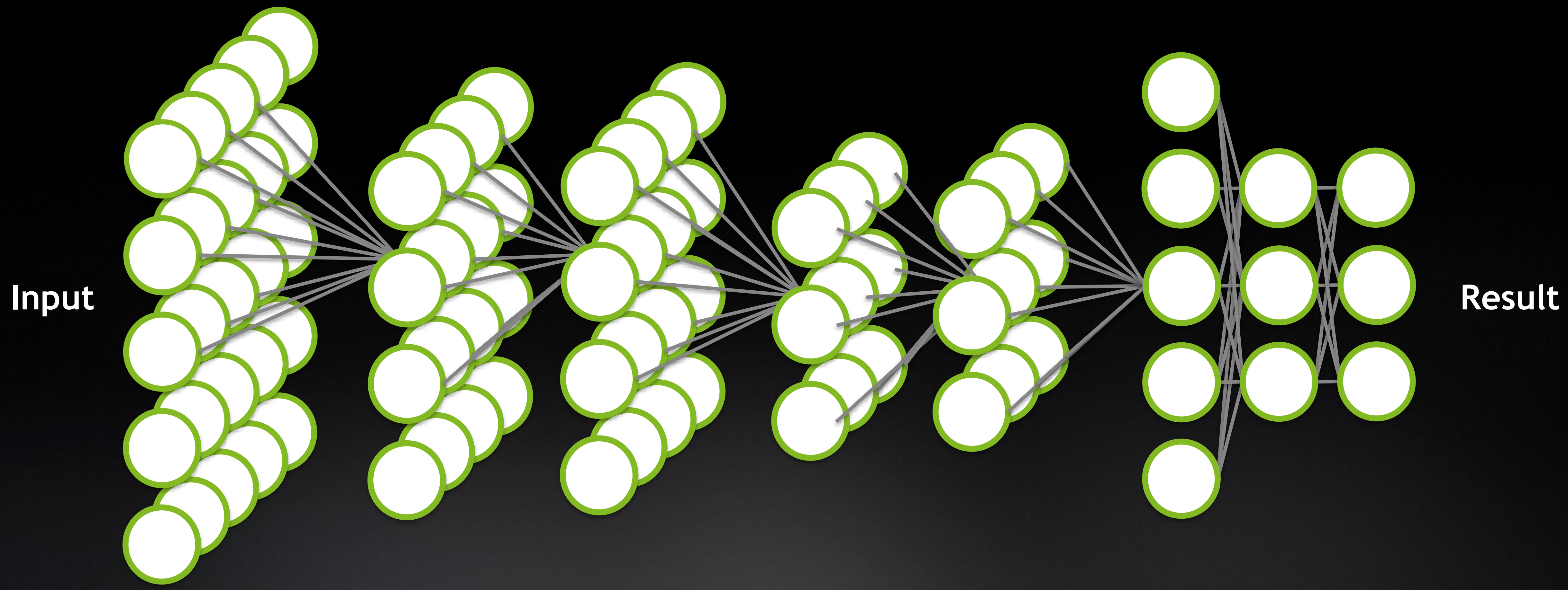
L1: RELU BASIS FCNS

# DEEPER NEURAL NETWORKS

## More layers allows for more levels of abstraction

Input | Low-level features | Mid-level features | High-level features | Output




https://web.eecs.umich.edu/~honglak/icml09-ConvolutionalDeepBeliefNetworks.pdf

# Large Scale Visual Recognition Challenge 2012

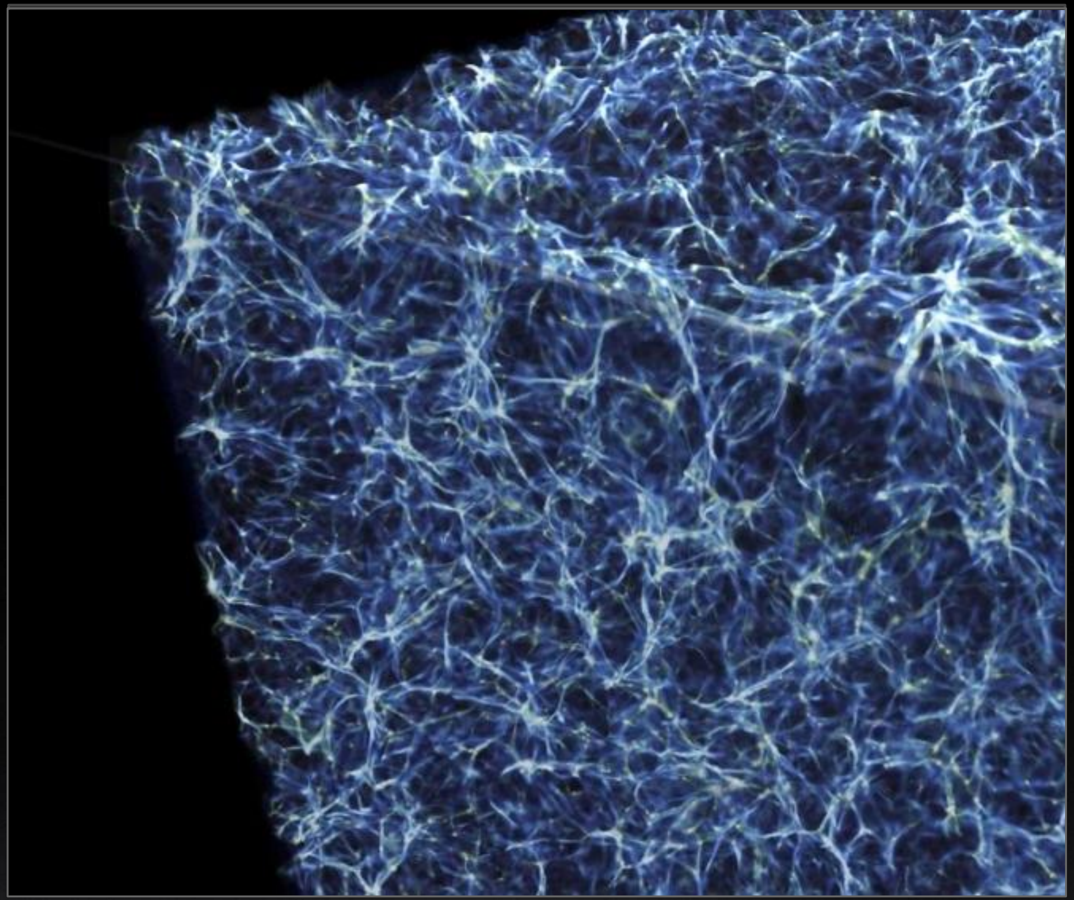

The Imagenet competition: Automatically classify images from 1000 different categories

# CONVOLUTIONAL NEURAL NETWORKS

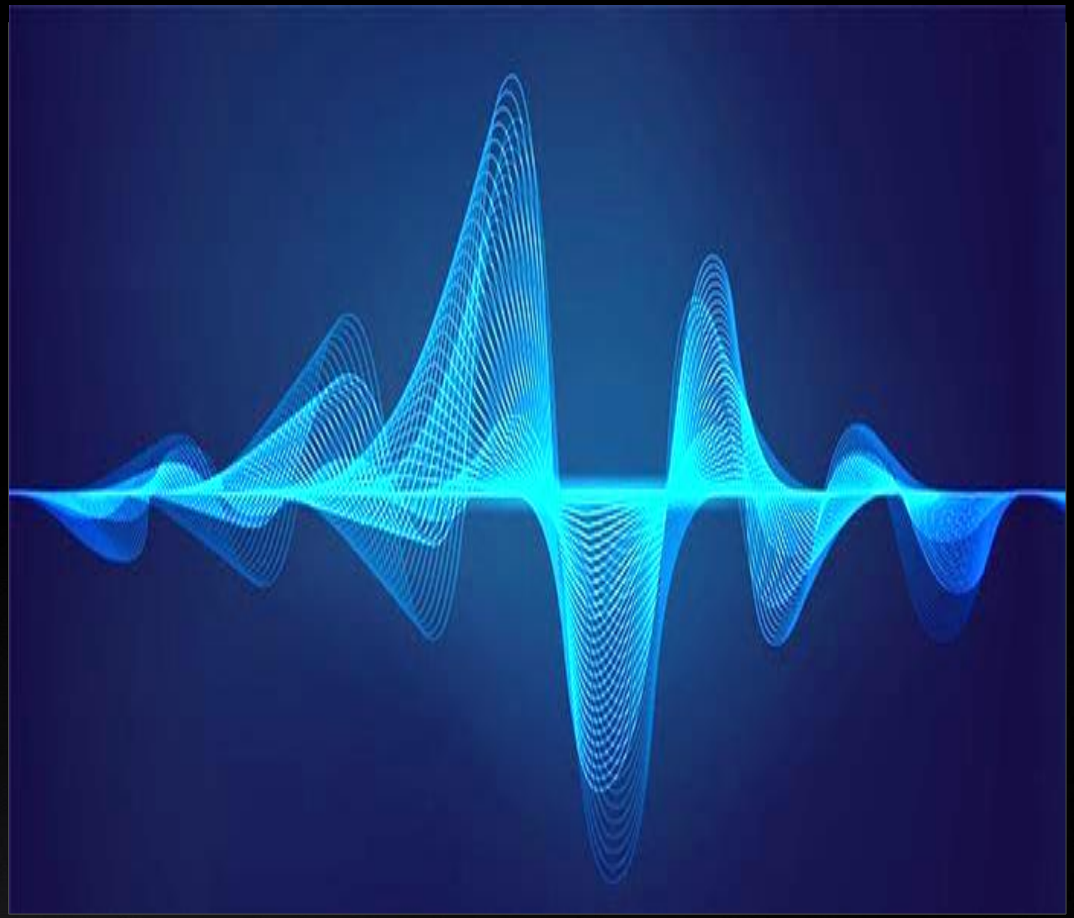
# WHAT ARE CNNS USED FOR?

## Problems with translational invariance

Computer Vision
Invariance in 2d space



Computational Physics
Invariance in 3d space



Audio and Time Series
Invariance in time

# COMPUTER VISION TASKS

## Each task requires a different model and data setup

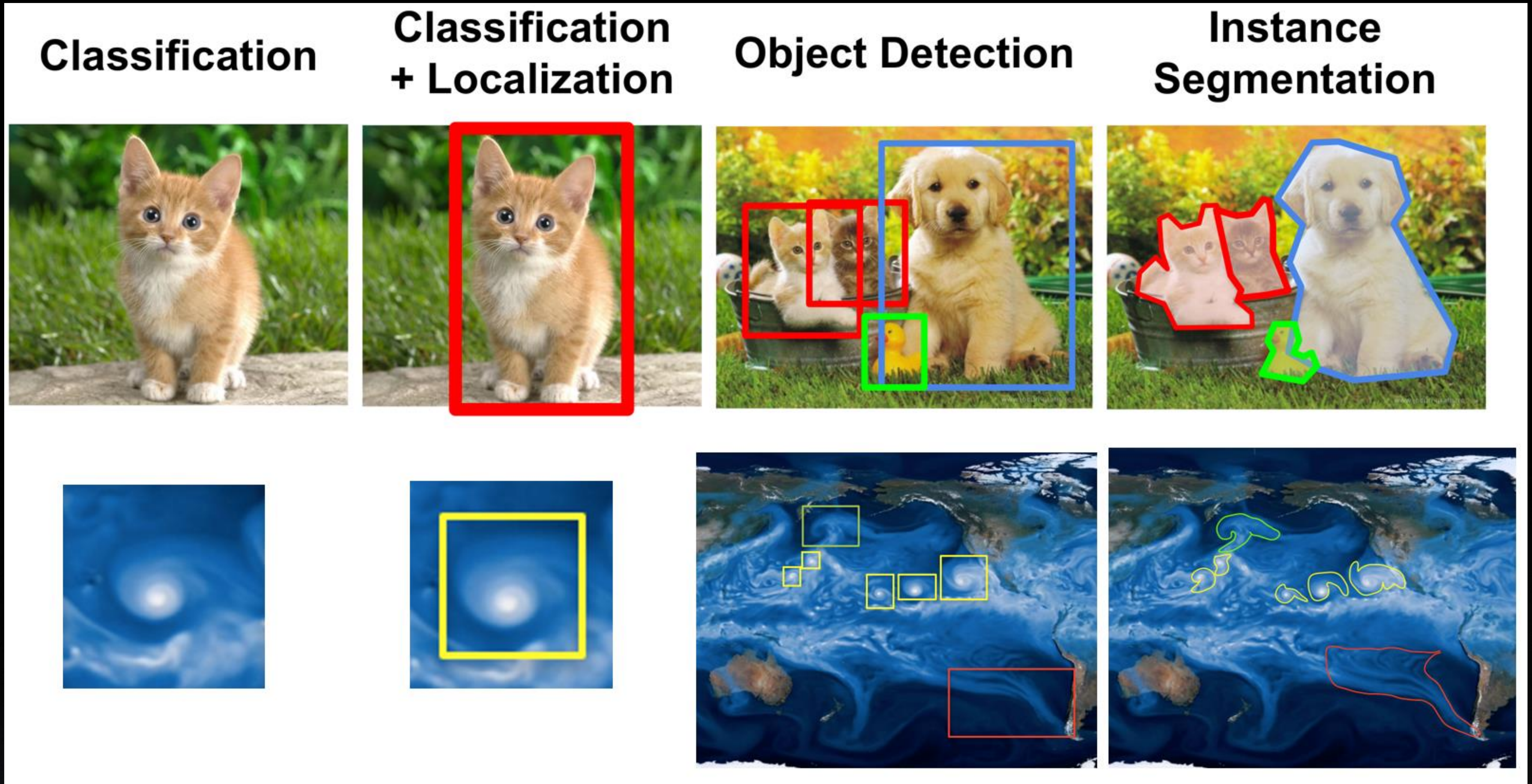
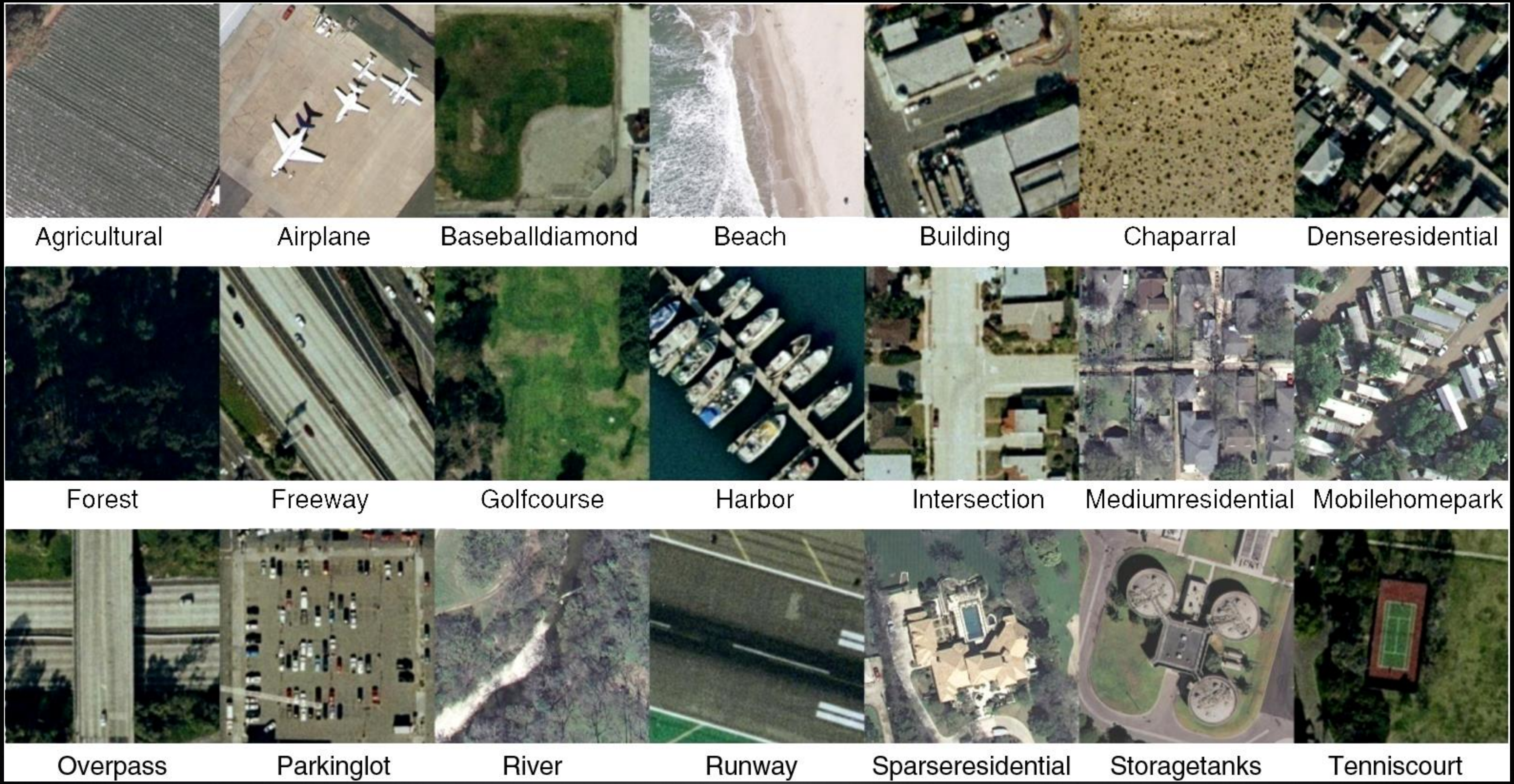



Image Credit: NERSC

# CLASSIFICATION

## Example: Classifying Land Use



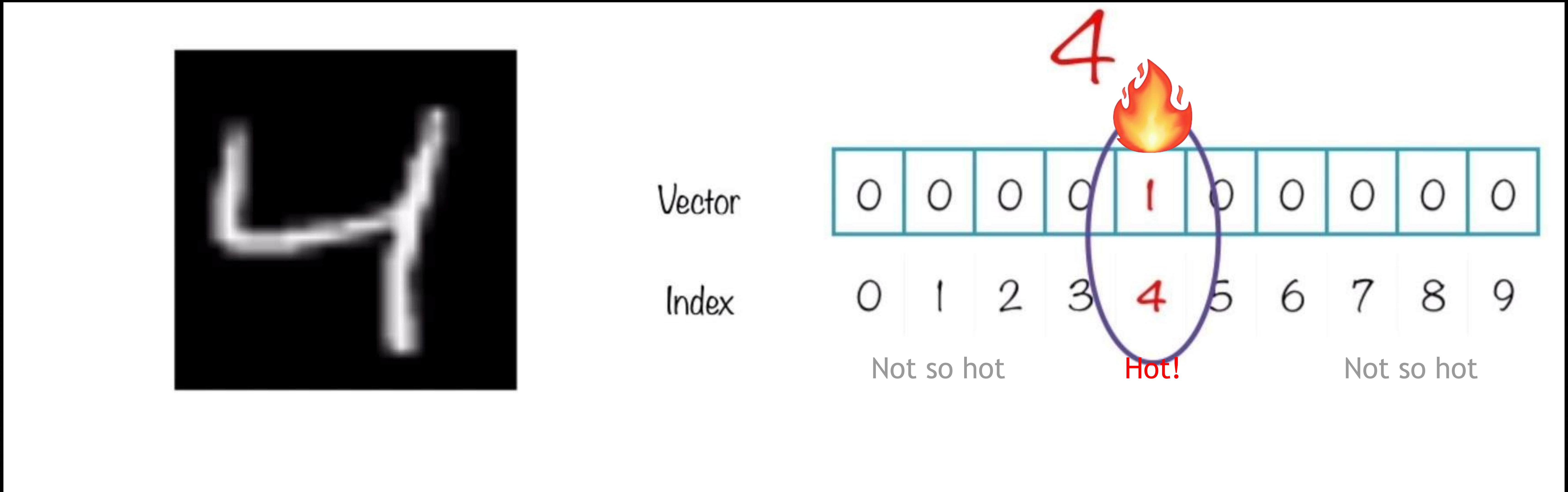

UC Merced Land Use Database

# ONE-HOT ENCODING

Input: Pixels, Output: One-hot encoding

INPUT:PIXEL VALUES

OUTPUT: ONE-HOT VECTOR

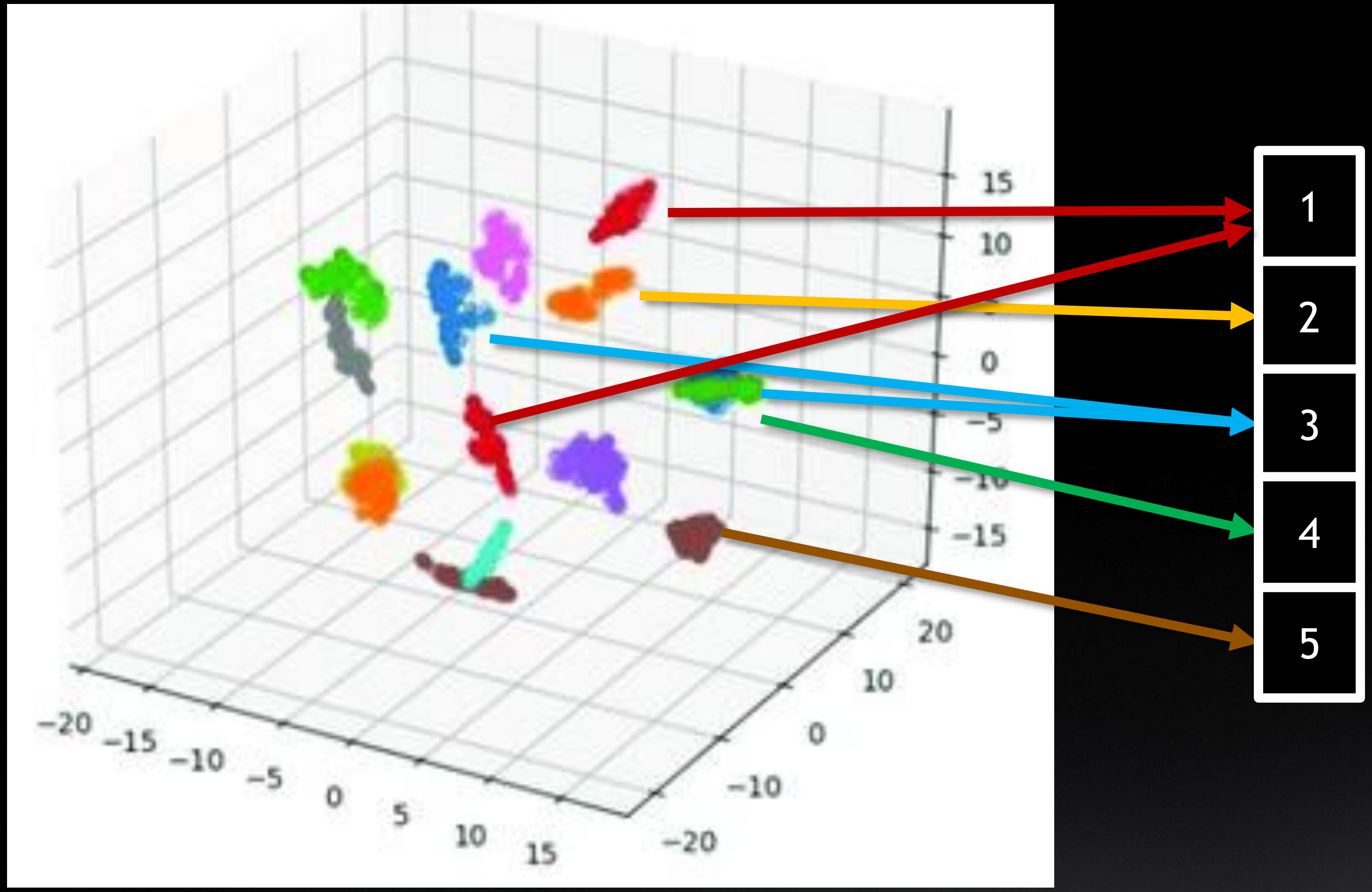



https://blog.carbonteq.com/practical-image-recognition-with-tensorflow/

# IMAGES ARE POINTS, WITH MANY DIMENSIONS

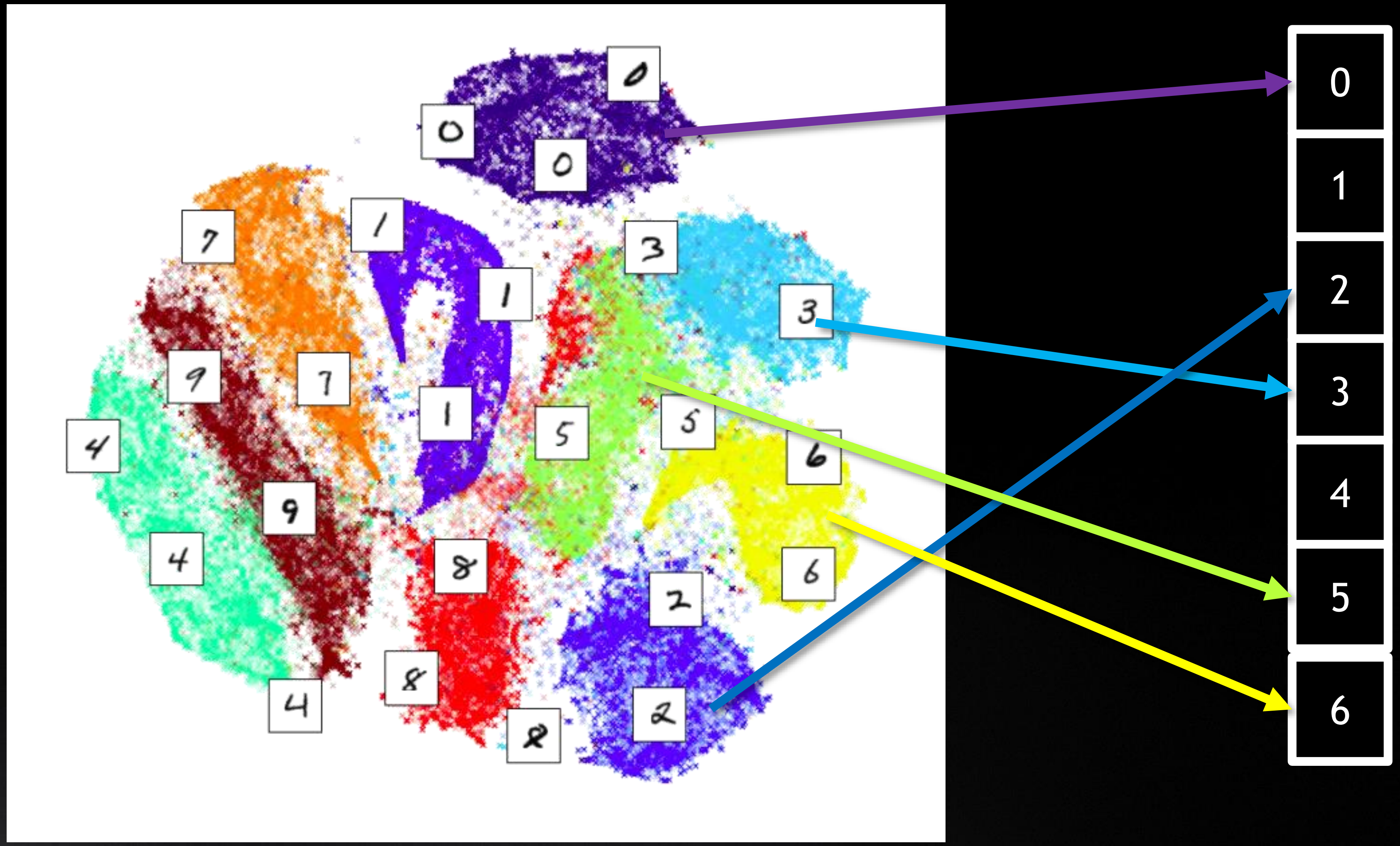
IN: 3-D Vector

OUT: 1 hot vector

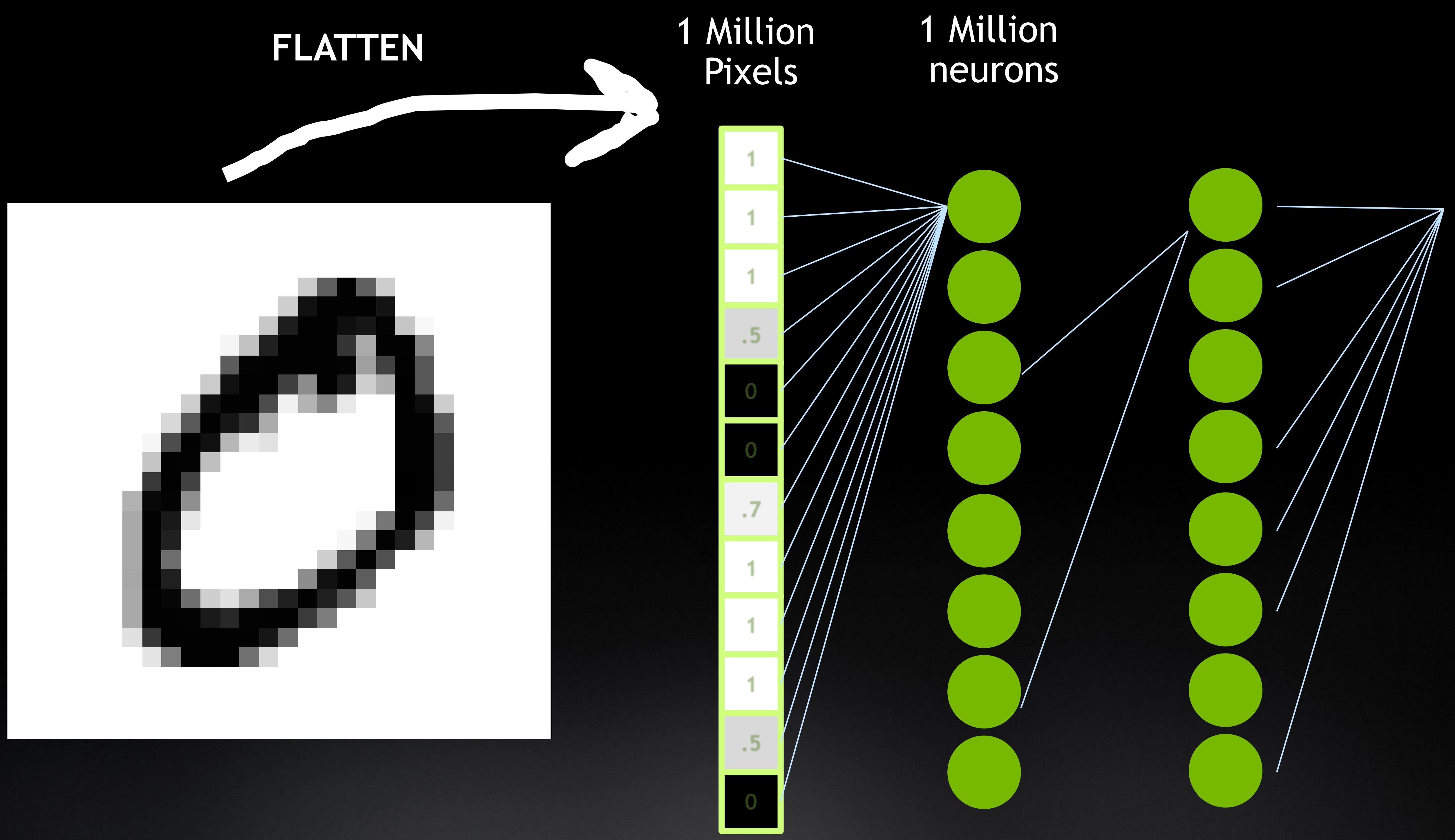



IN: 784-D Vector

OUT: 1-hot vector

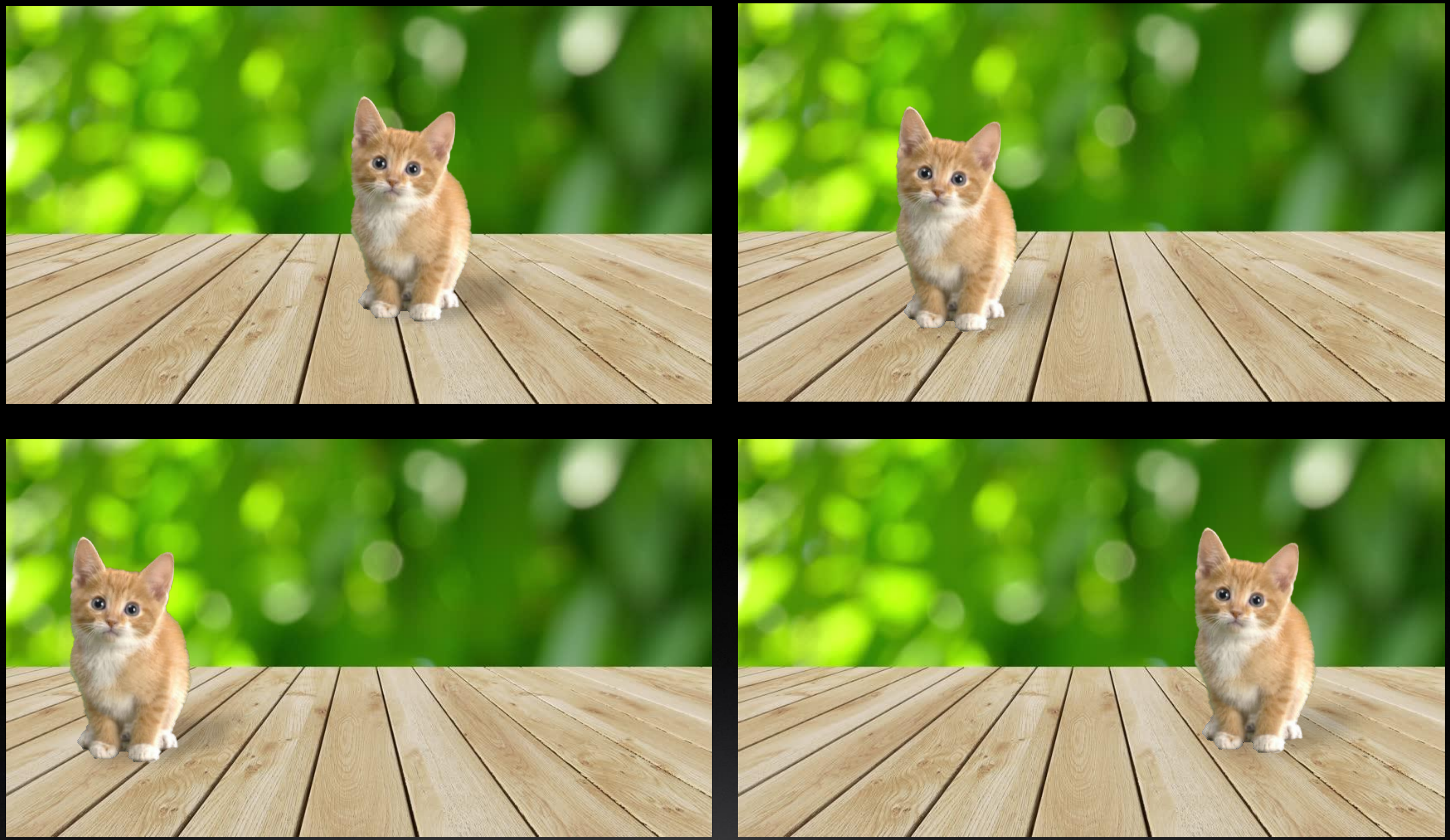


# FULLY CONNECTED NETWORKS AND IMAGES DON'T MIX



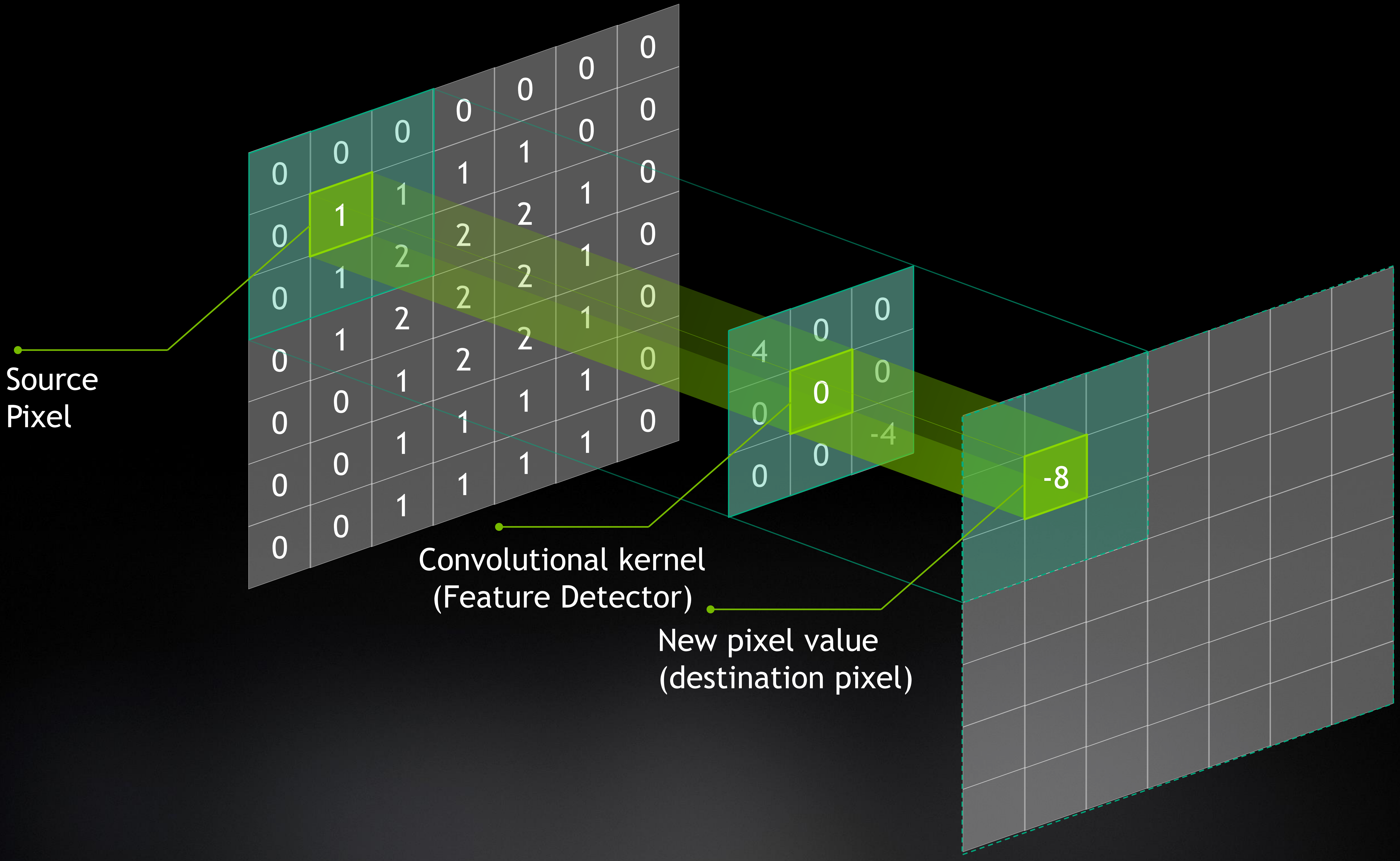
# TRANSLATIONAL EQUIVARIANCE

Objects in nature look the same from place to place

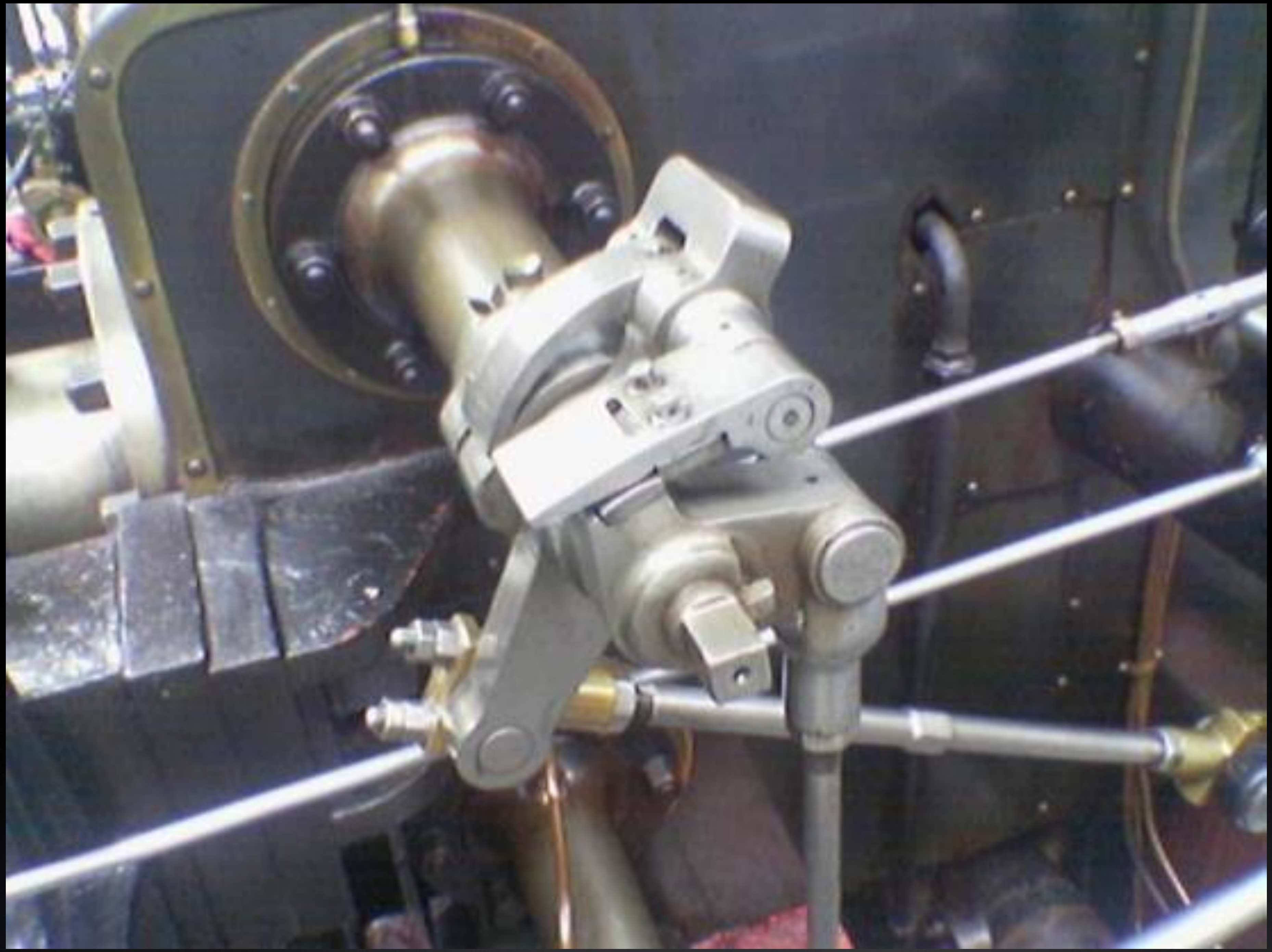


# WHAT IS A CONVOLUTION?

## A small matrix transformation, applied at each point of the image



# CONVOLUTION EXAMPLE: SOBEL FILTER



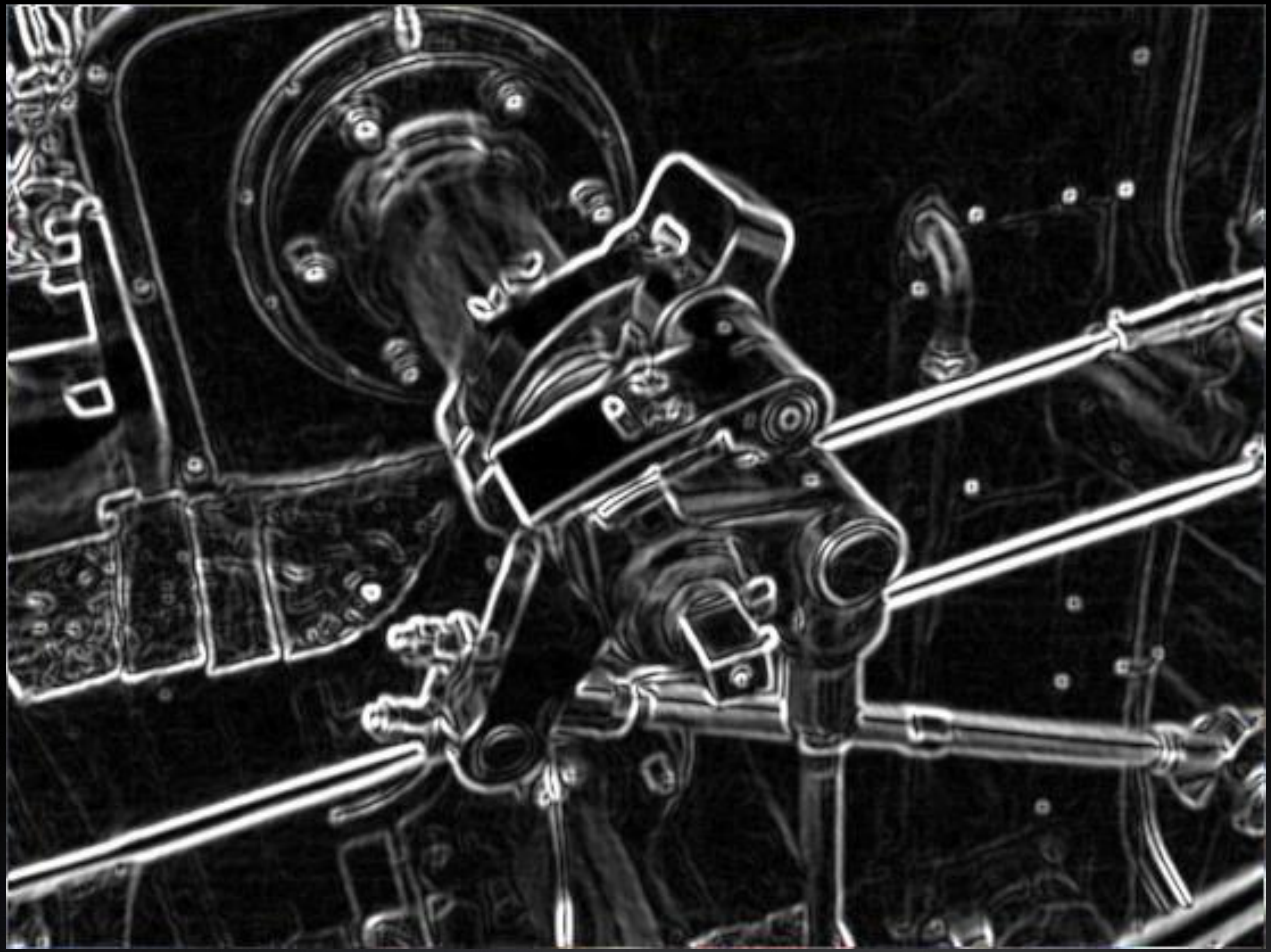
$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

Image source: https://en.wikipedia.org/wiki/Sobel_operator

# CONVOLUTION EXAMPLE: SOBEL FILTER



$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

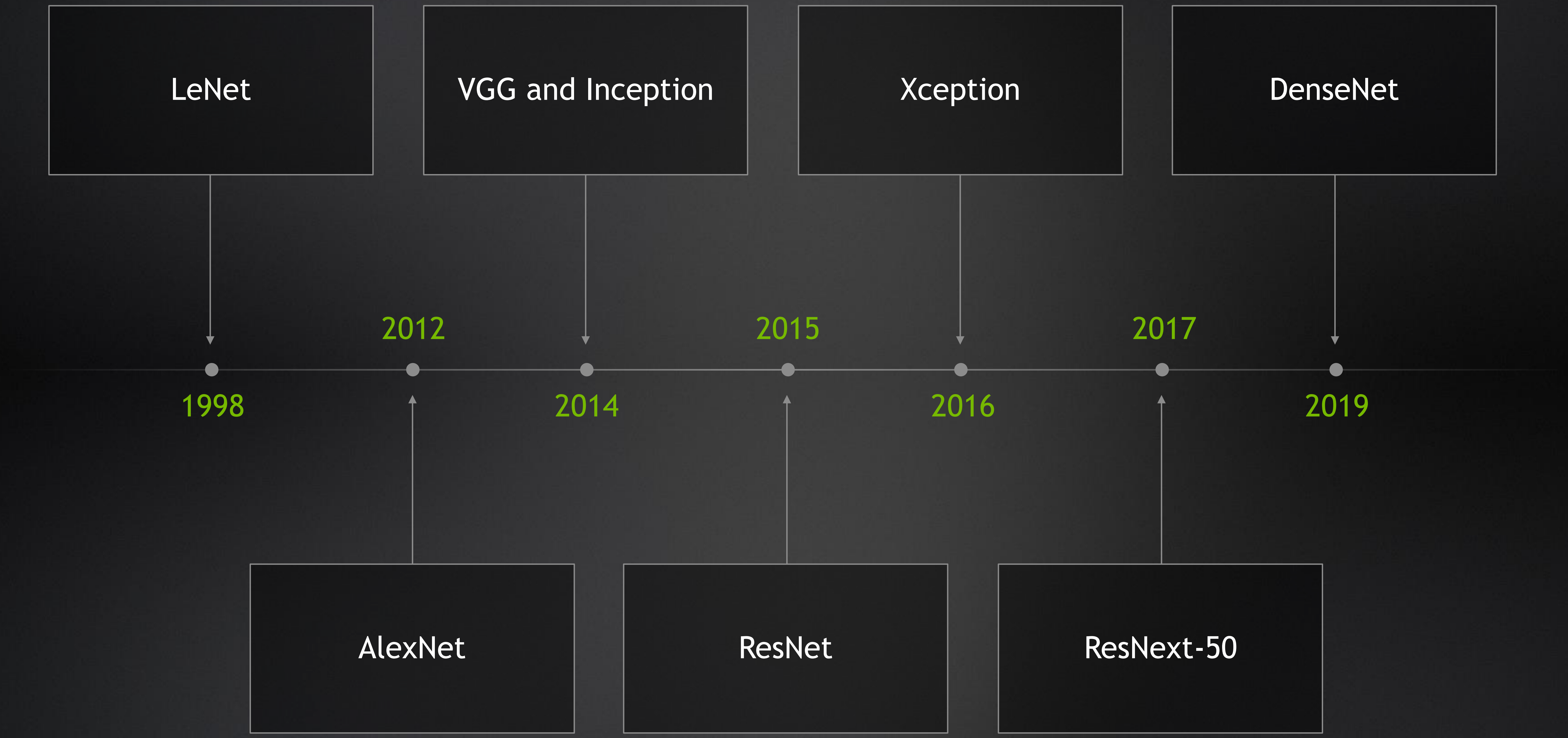
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

Image source: https://en.wikipedia.org/wiki/Sobel_operator

# CLASSIFICATION

# CLASSIFIER EVOLUTION OVER TIME




https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

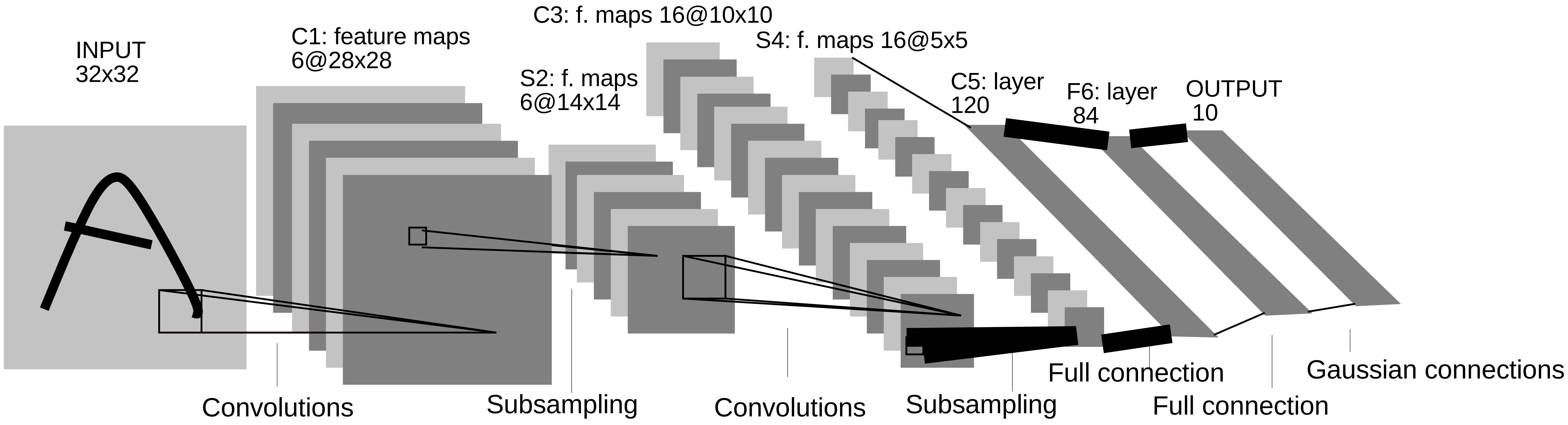
# LENET-5

## (1988) Yann LeCun. Hand written recognition. 60k parameters.

PROC. OF THE IEEE, NOVEMBER 1998

Gradient-Based Learning Applied to Document Recognition

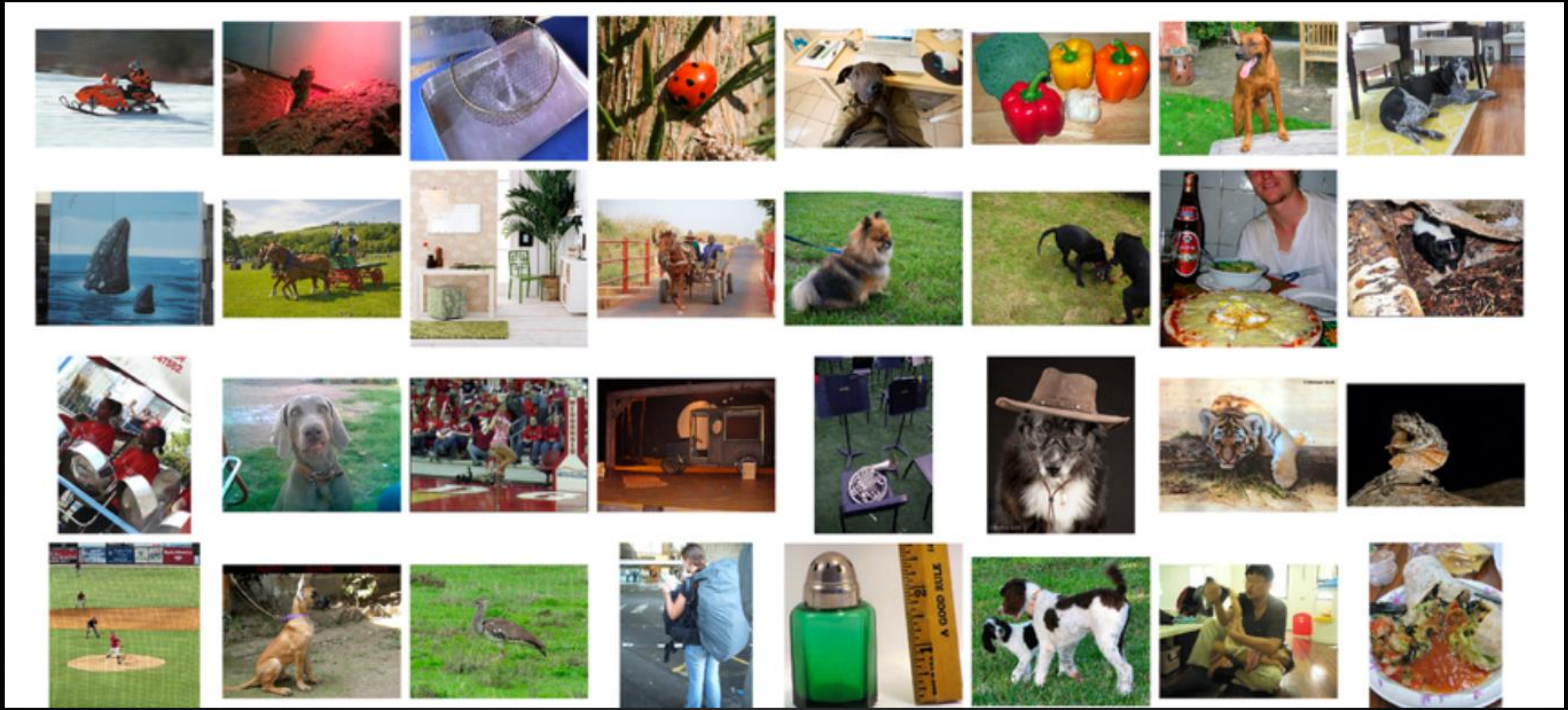
Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner




https://en.wikipedia.org/wiki/LeNet.

# IMAGENET ILSVR COMPETITION

## Large Scale Visual Recognition Competition (2010-2017)



https://en.wikipedia.org/wiki/ImageNet

# ALEXNET

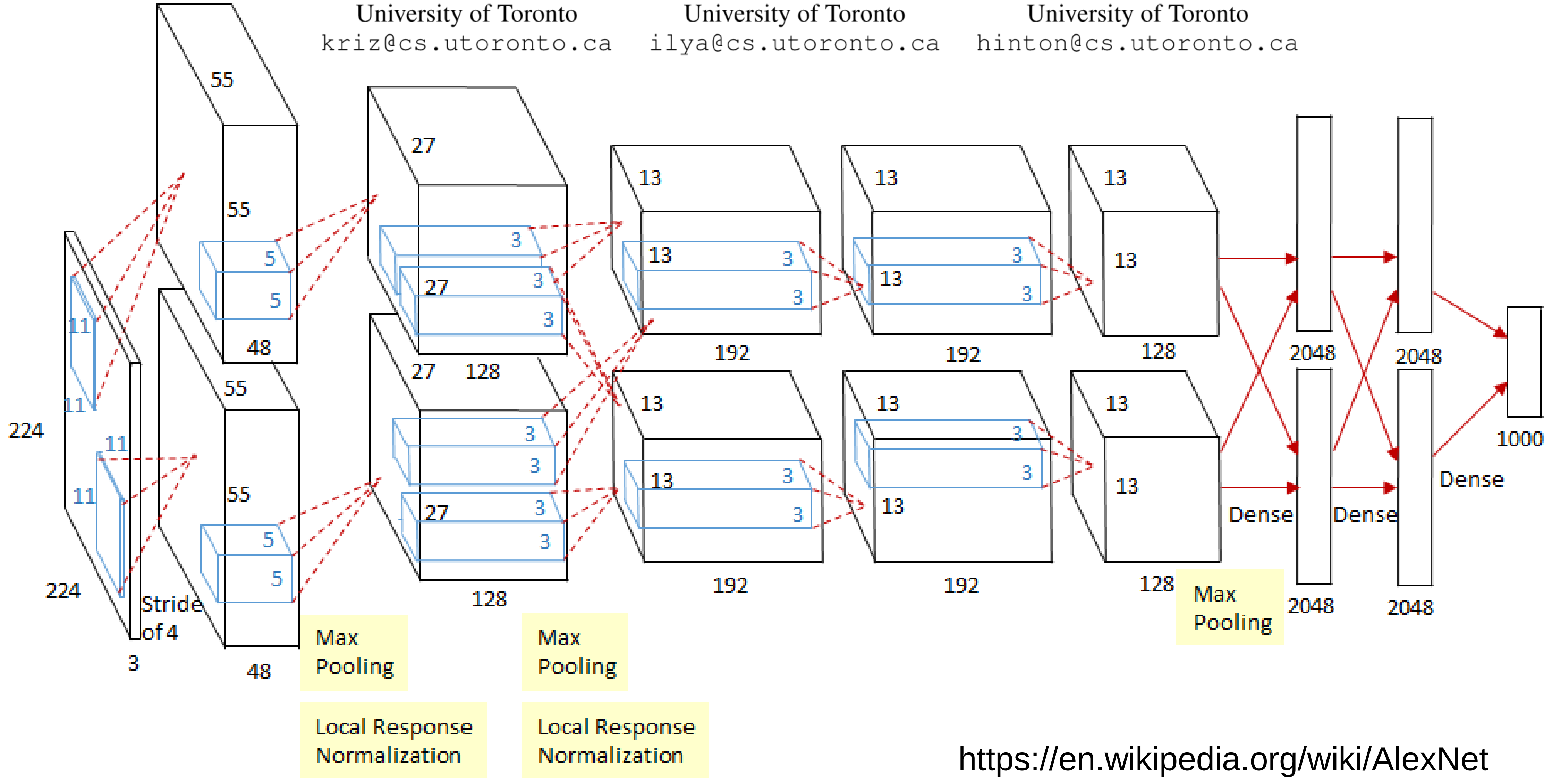
## (2012): Krizevsky, Sutskever, Hinton. ImageNet winner.

**ImageNet Classification with Deep Convolutional Neural Networks**

**Alex Krizhevsky**
University of Toronto
kriz@cs.utoronto.ca

**Ilya Sutskever**
University of Toronto
ilya@cs.utoronto.ca

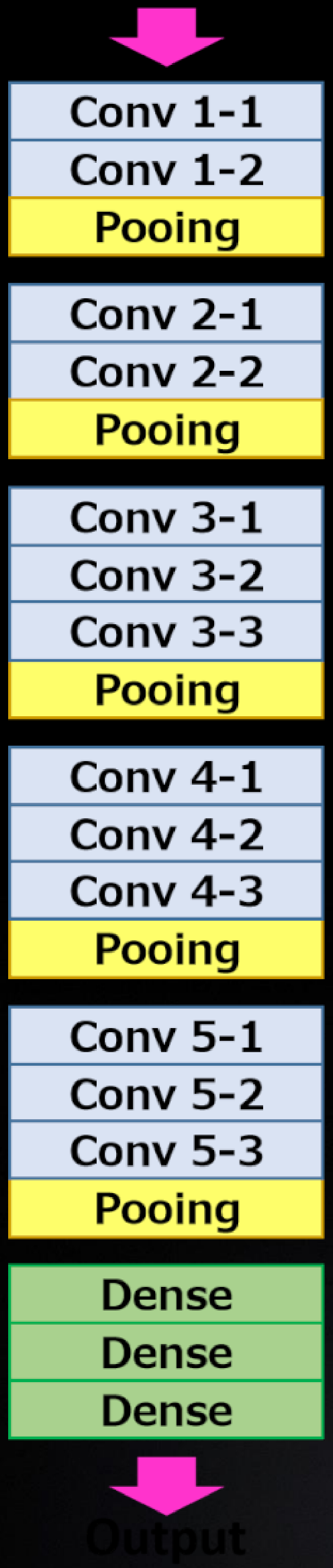
**Geoffrey E. Hinton**
University of Toronto
hinton@cs.utoronto.ca

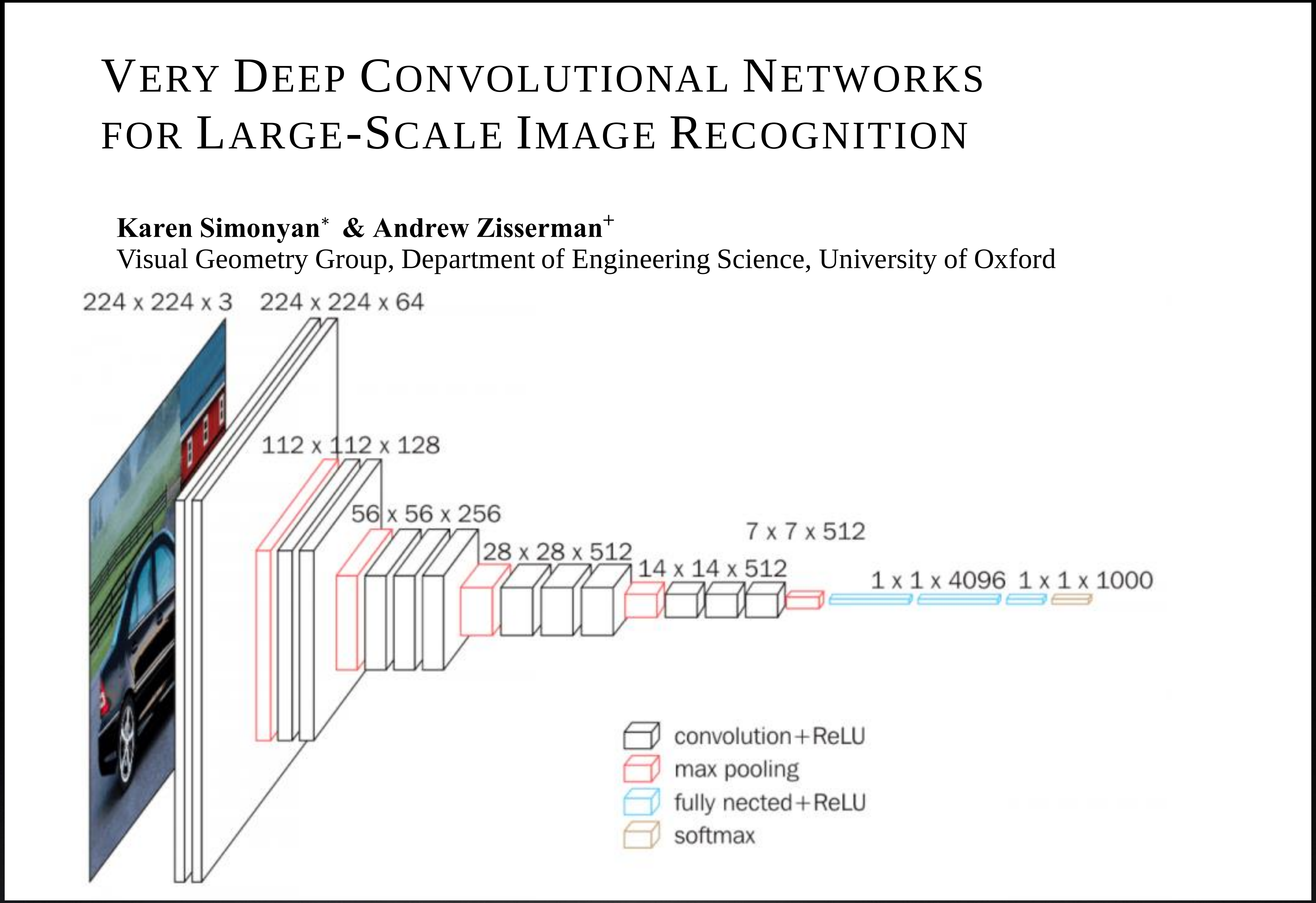



https://en.wikipedia.org/wiki/AlexNet

# VGG-16

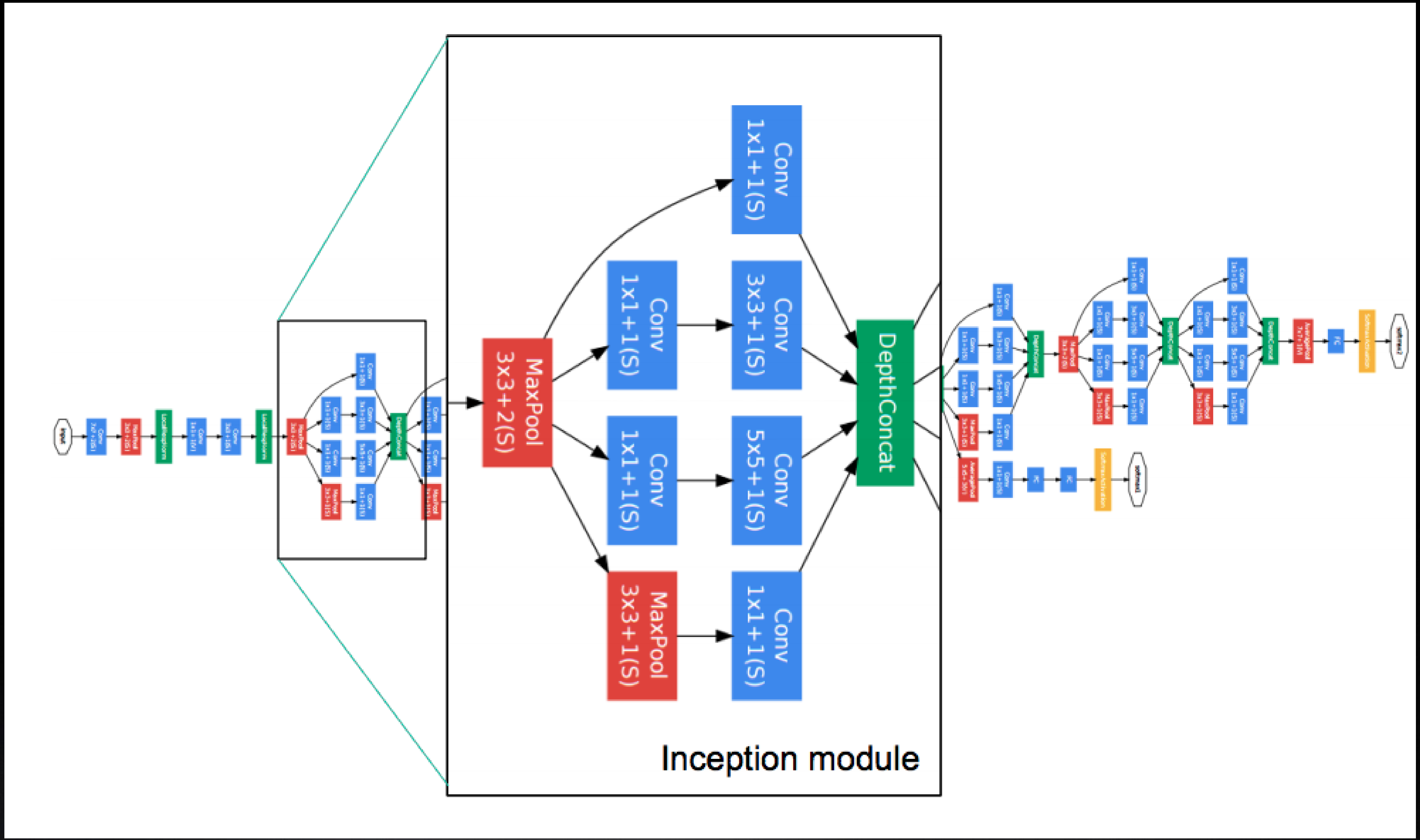
2014. ImageNet runner up. Simple, clean architecture.



# INCEPTION-V1 (GOOGLENET)

## 2014. Train different size convolutions in parallel

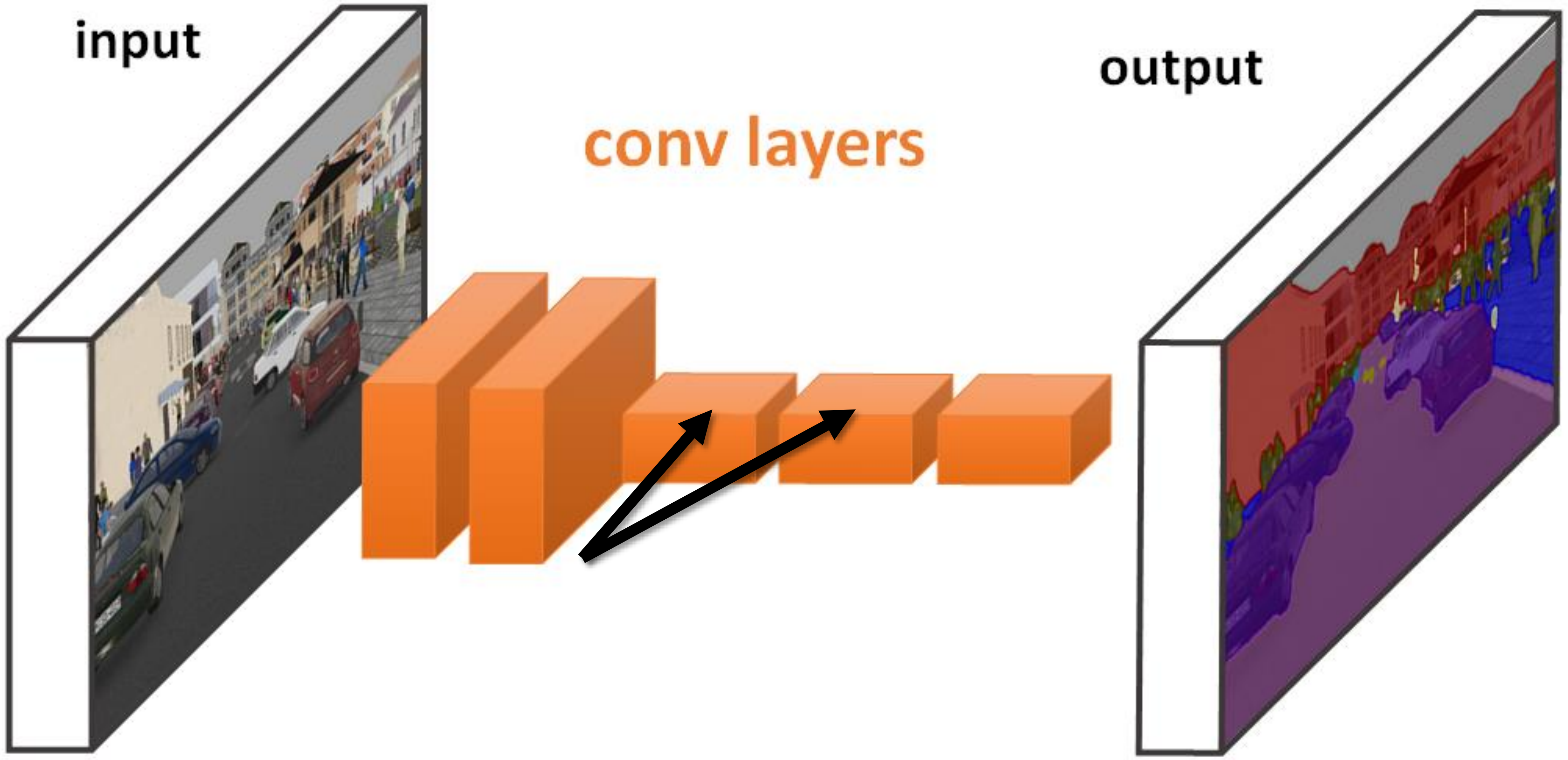


# FULLY CONVOLUTIONAL NETWORKS (FCN)

## 2015: Convert fully connect layers into convolutions of the same size

**Fully Convolutional Networks for Semantic Segmentation**

Jonathan Long*     Evan Shelhamer*     Trevor Darrell
UC Berkeley



# RESNETS

# MODELING TRENDS: DEEPER AND LARGER

LeNet
5 Layers,
60 k Params
1998

AlexNet
8 layers
60 M Params
2012

VGG-16
16 Layers
138 M Params
2014

ResNet
50 Layers
23 M Params
2015

Xception
2016

ResNext-50
2017

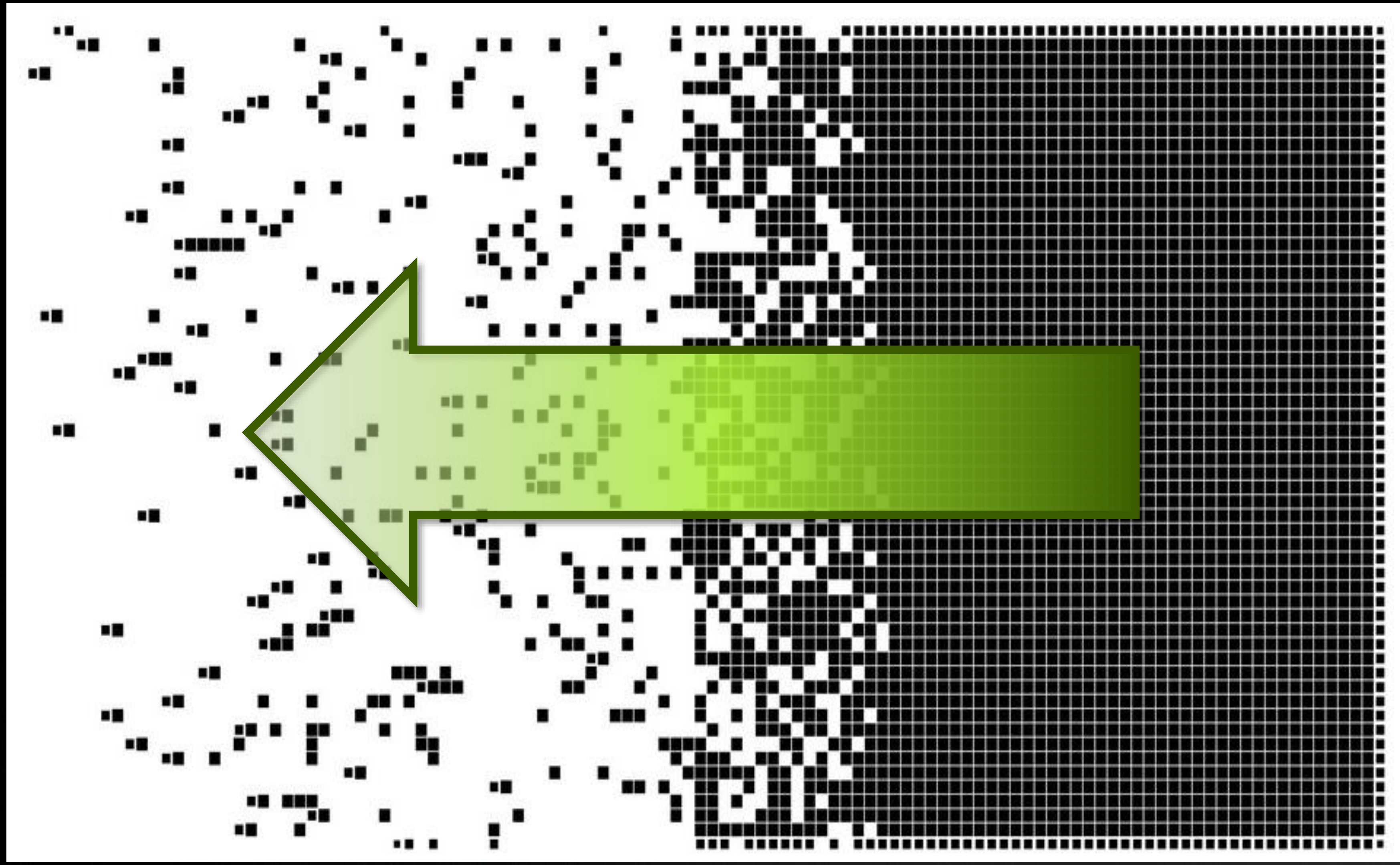
DenseNet
2019

*Source: Source information is 14 pt, italic*

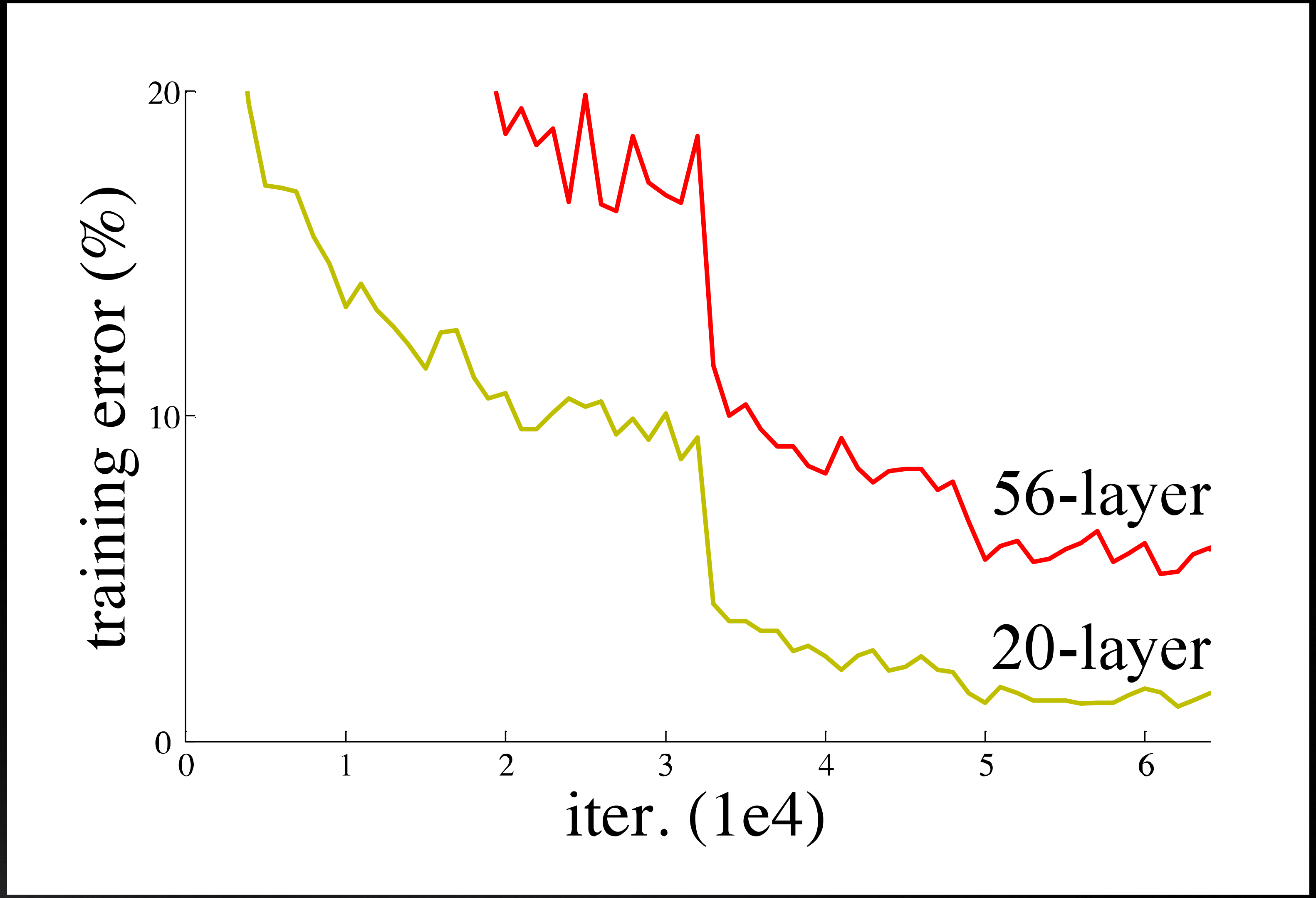
# PROBLEM: VANISHING GRADIENTS

Error signal decays exponentially as it propagates backward through the network

## ERROR SIGNAL VANISHES DURING BACKPROP



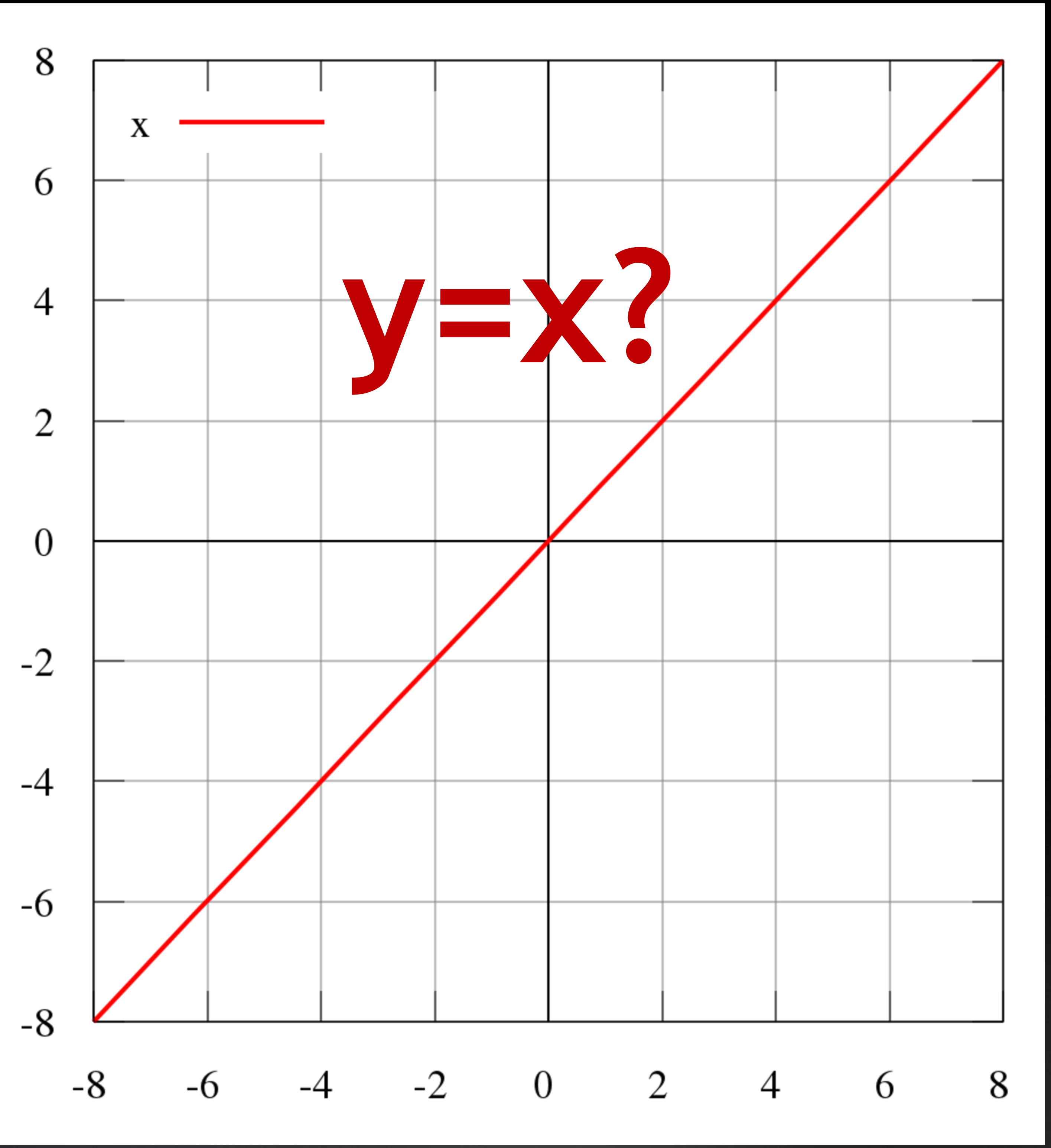
## DEEPER NETWORKS WERE HARDER TO TRAIN




https://www.arxiv-vanity.com/papers/1512.03385/

# PROBLEM: THE MISSING IDENTITY

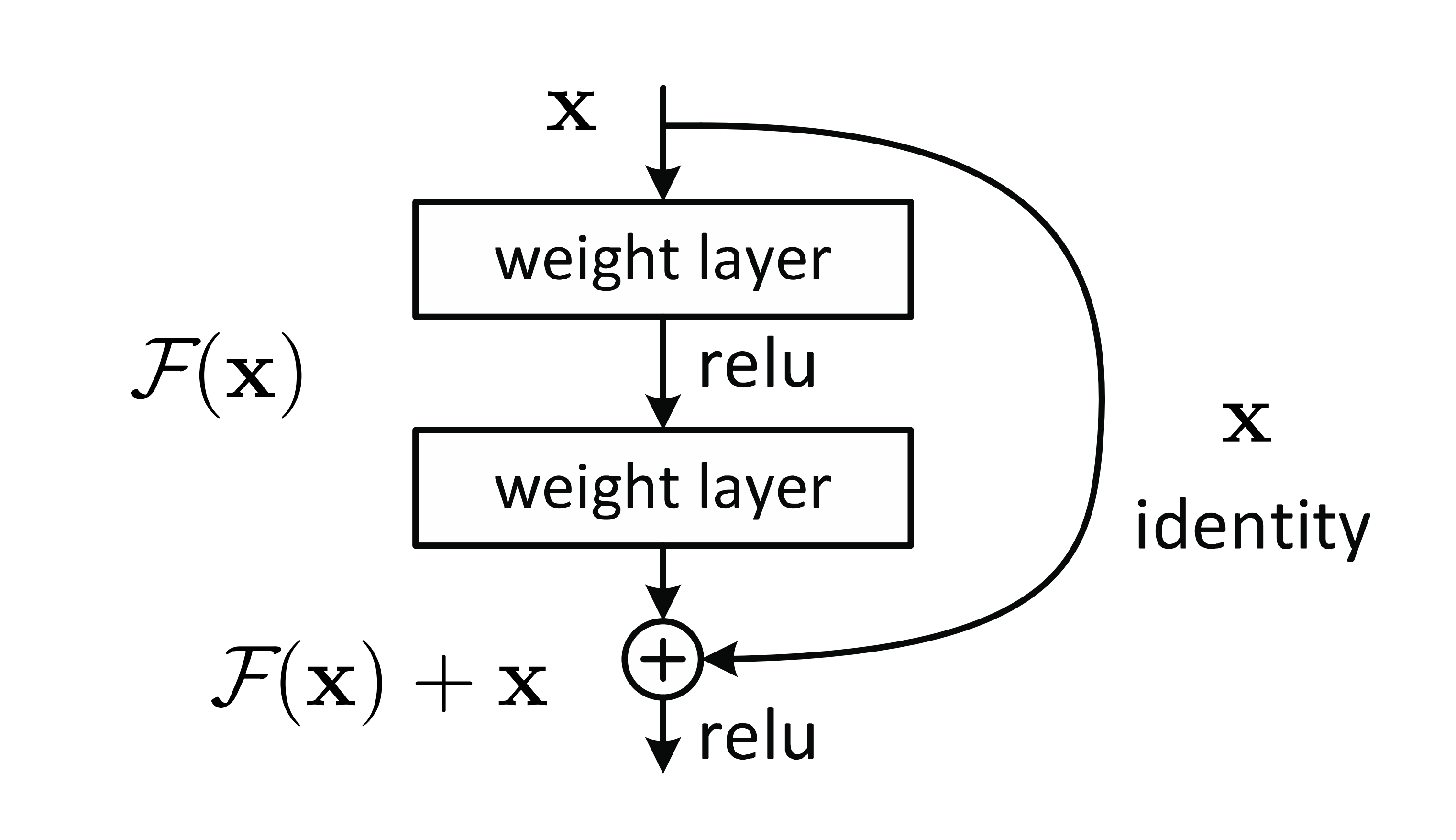
2015: Neural nets had a hard time learning the identity function!



# RESNETS AND SKIP CONNECTIONS

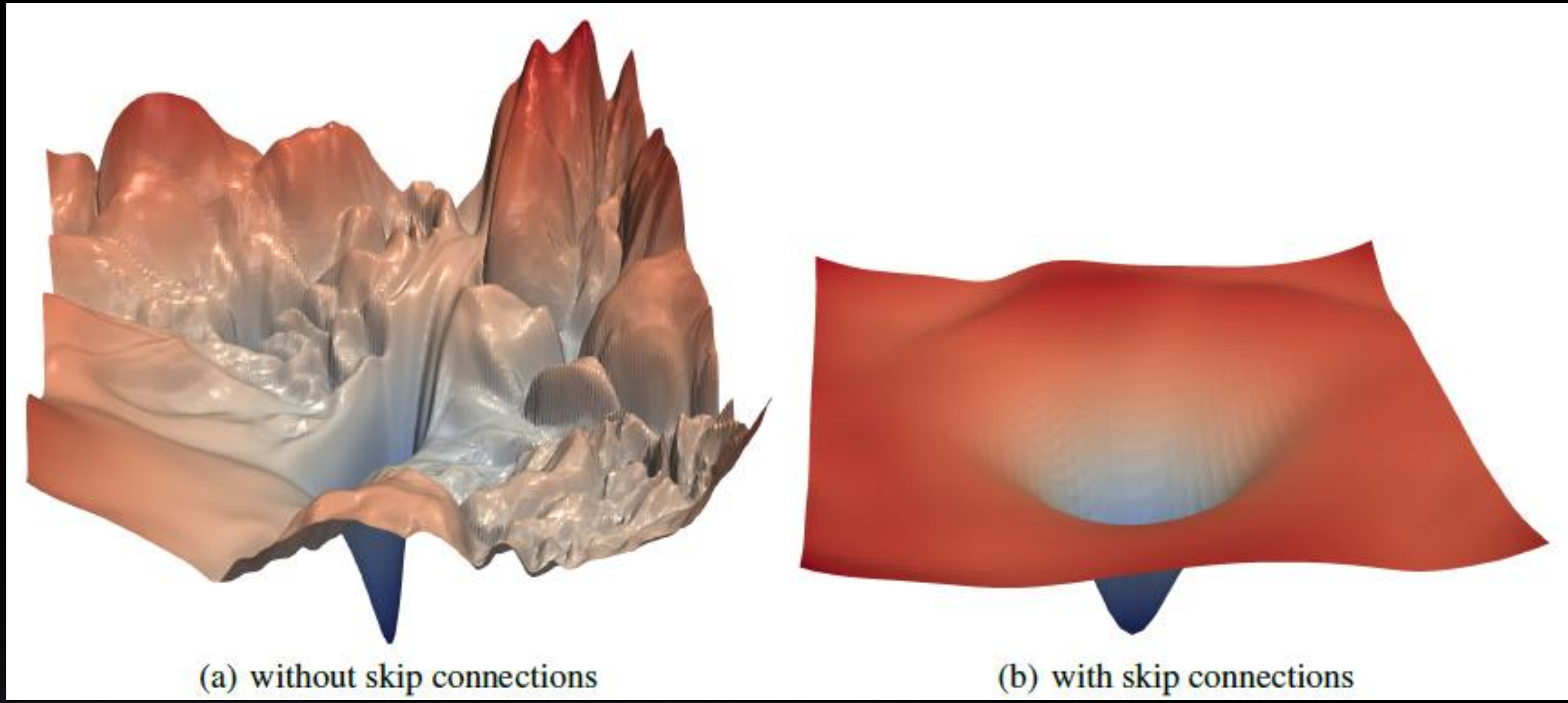
## (aka Highway Networks)

### ADD THE INPUT TO OUTPUT




https://arxiv.org/pdf/1512.03385.pdf

### DRAMTICALLY SIMPLIFIES THE LOSS LANDSCAPE

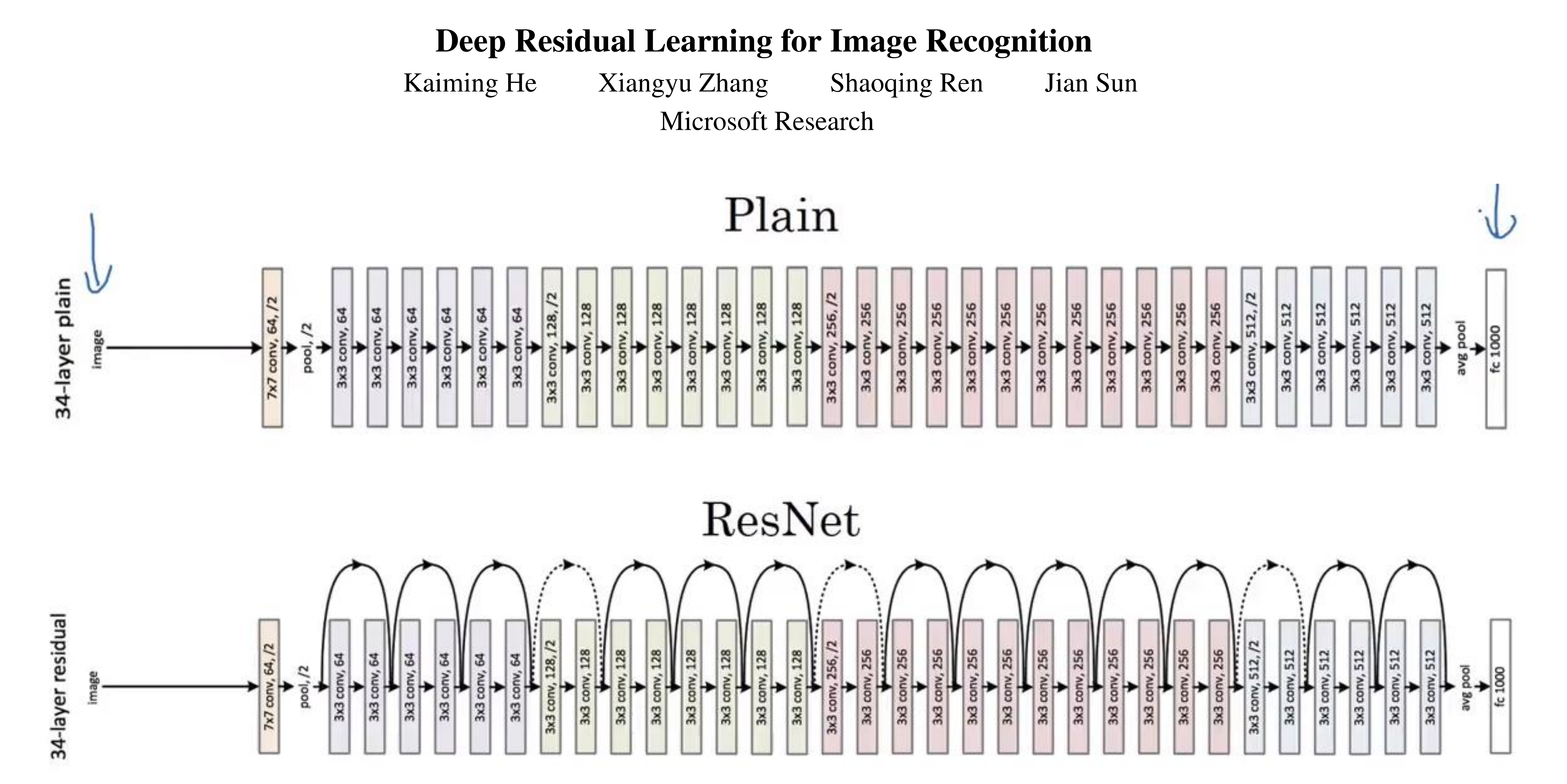



https://arxiv.org/abs/1712.09913

https://jithinjk.github.io/blog/nn_loss_visualized.md.html

# RESNET-50

## 2015 Microsoft Research. 50 Layers, 23M params.




https://arxiv.org/pdf/1409.1556.pdf

# DENSENET

## 2017

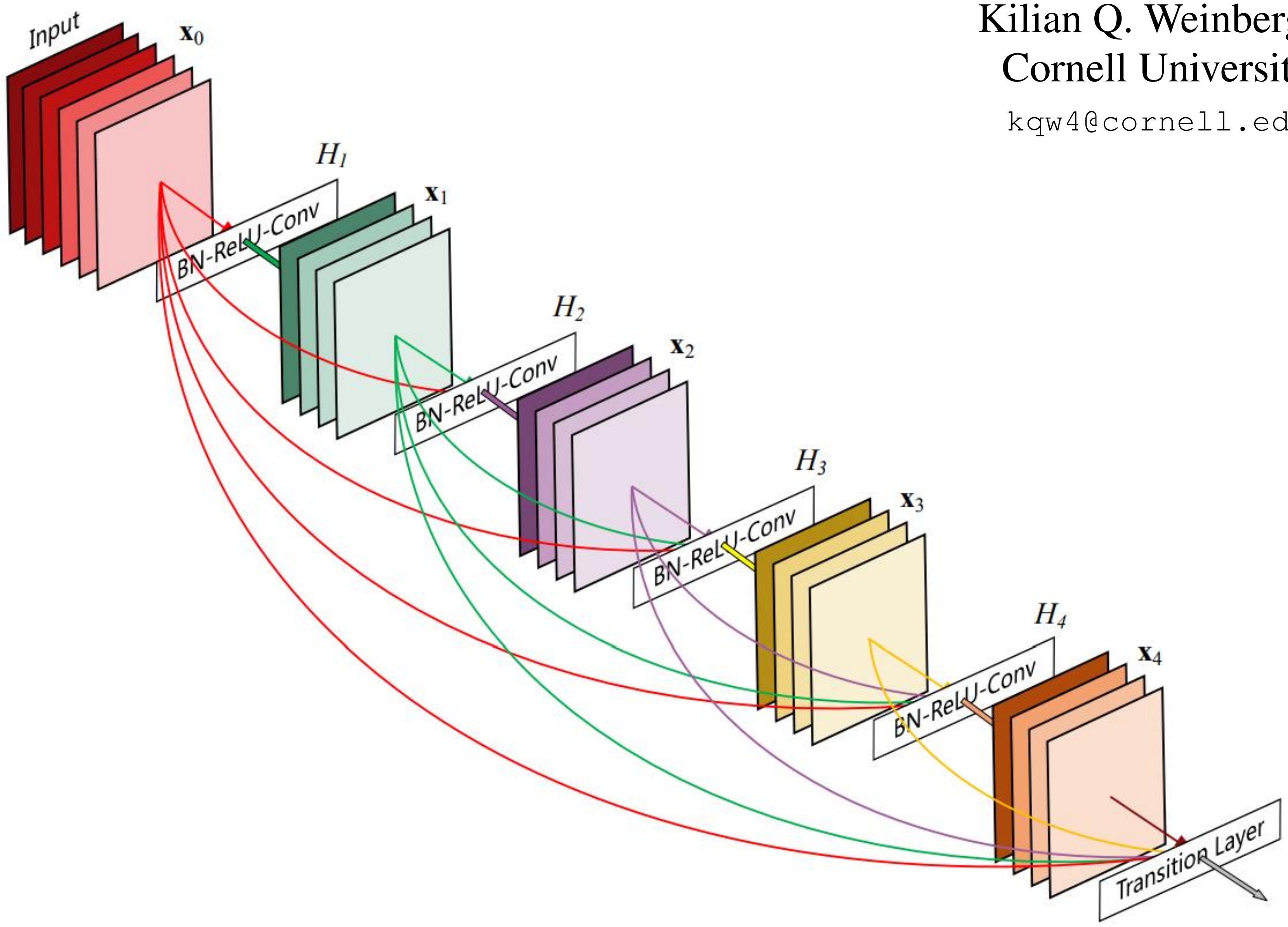
**Densely Connected Convolutional Networks**

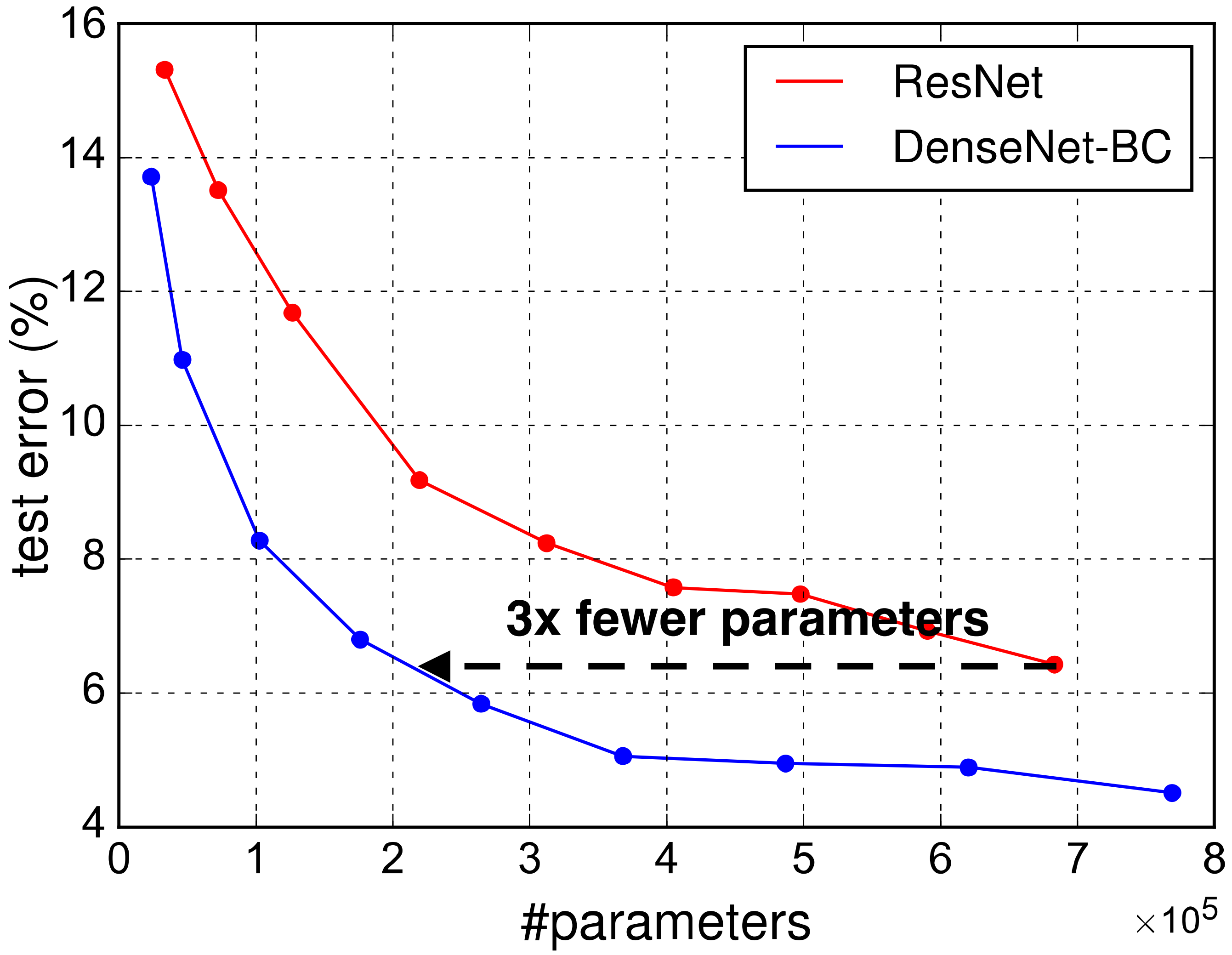
Gao Huang*
Cornell University
gh349@cornell.edu

Zhuang Liu*
Tsinghua University
liuzhuang13@mails.tsinghua.edu.cn

Laurens van der Maaten
Facebook AI Research
lvdmaaten@fb.com

Kilian Q. Weinberger
Cornell University
kqw4@cornell.edu



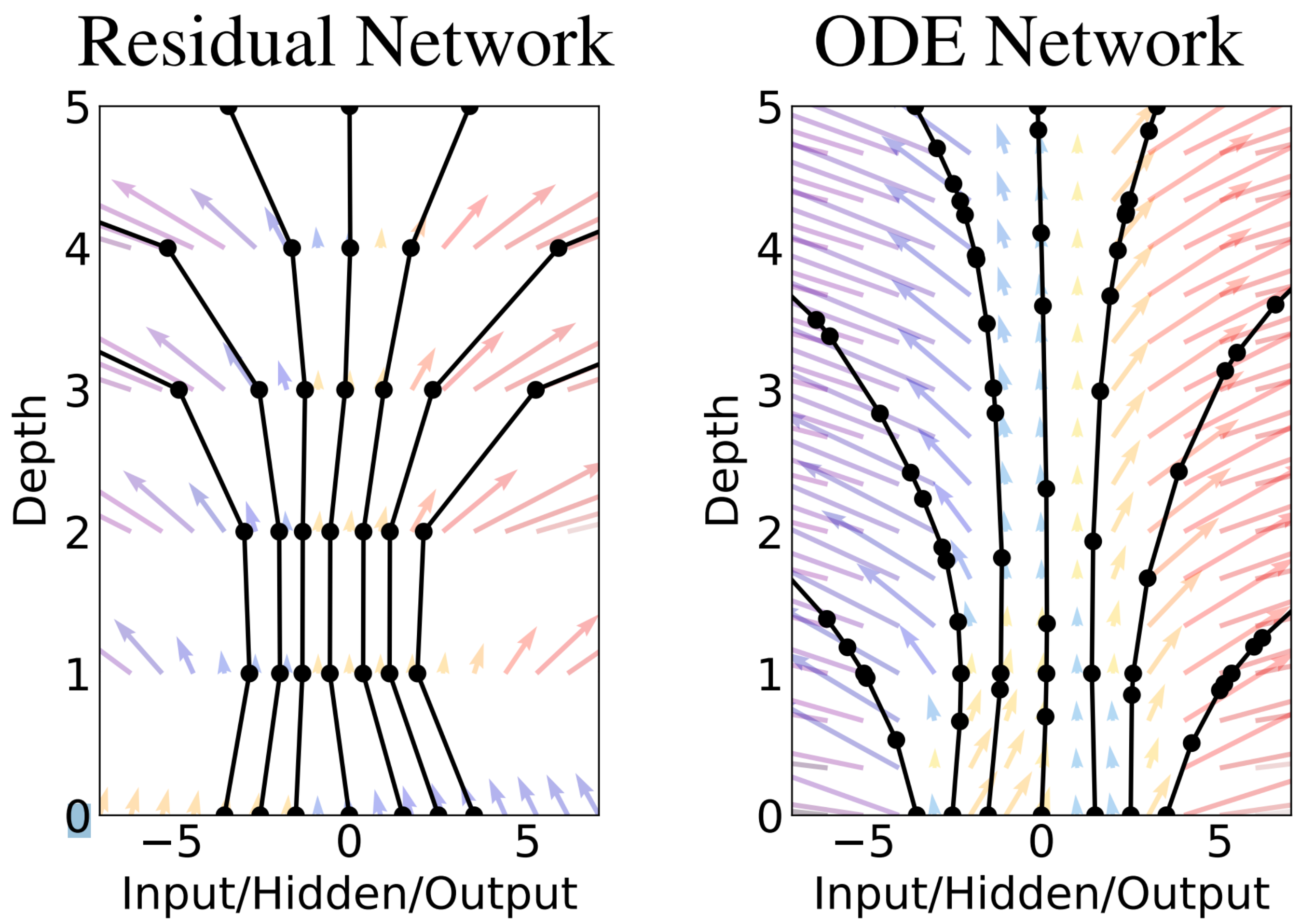

https://arxiv.org/abs/1608.06993

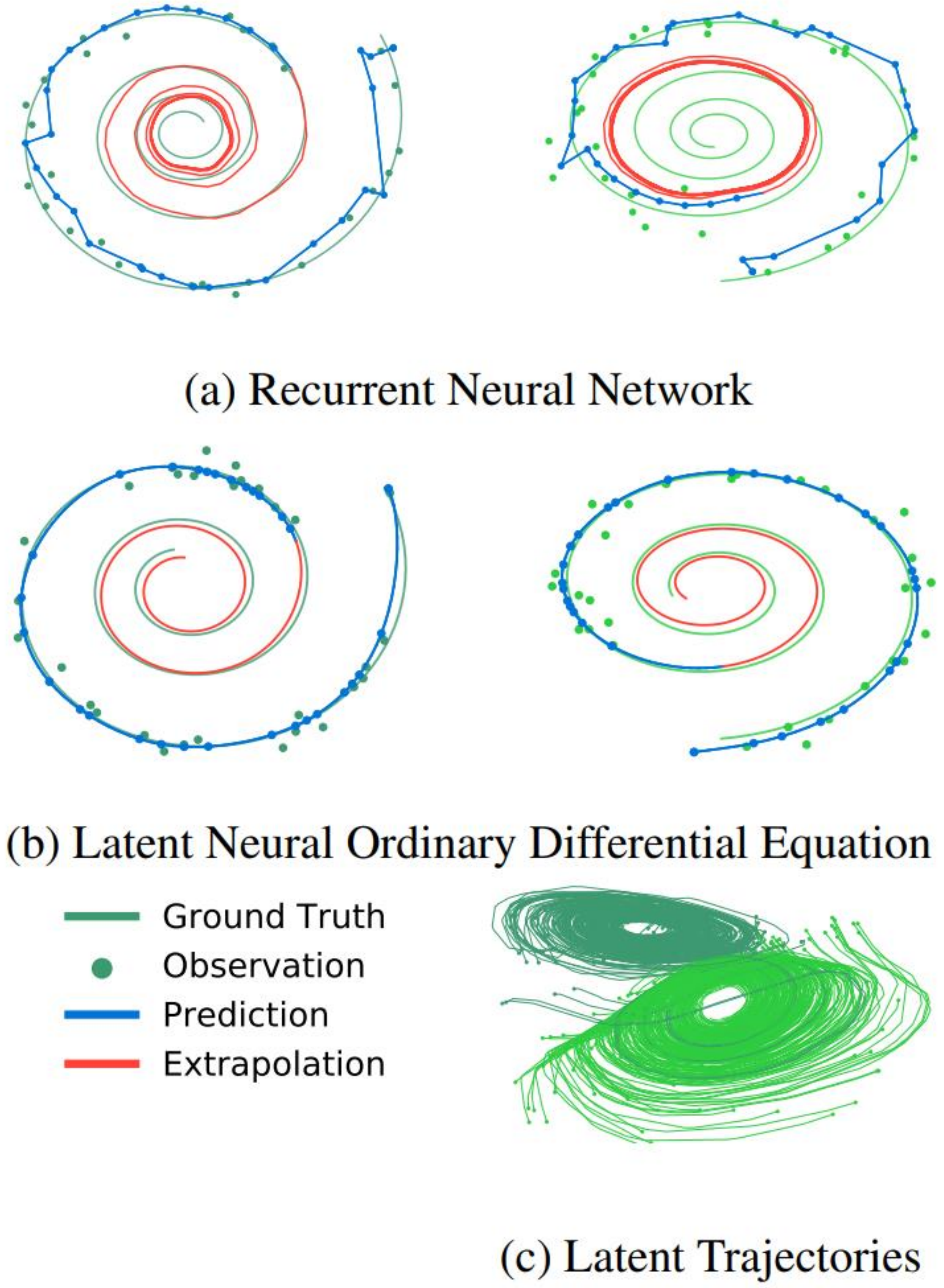
# NEURAL-ODES

2018

## Neural Ordinary Differential Equations

Ricky T. Q. Chen*, Yulia Rubanova*, Jesse Bettencourt*, David Duvenaud
University of Toronto, Vector Institute
{rtqichen, rubanova, jessebett, duvenaud}@cs.toronto.edu




(a) Recurrent Neural Network

(b) Latent Neural Ordinary Differential Equation

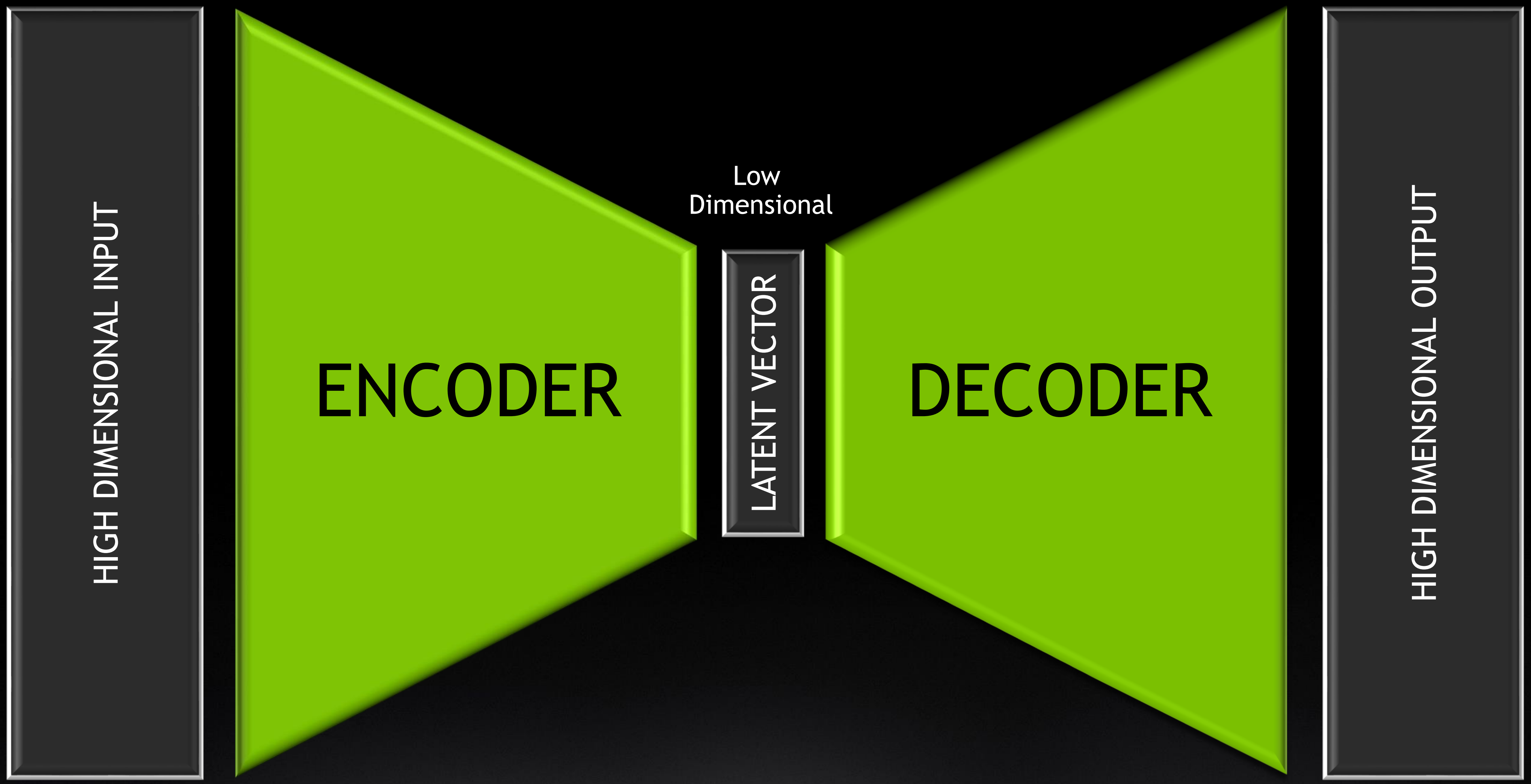
(c) Latent Trajectories

https://www.depthfirstlearning.com/2019/NeuralODEs

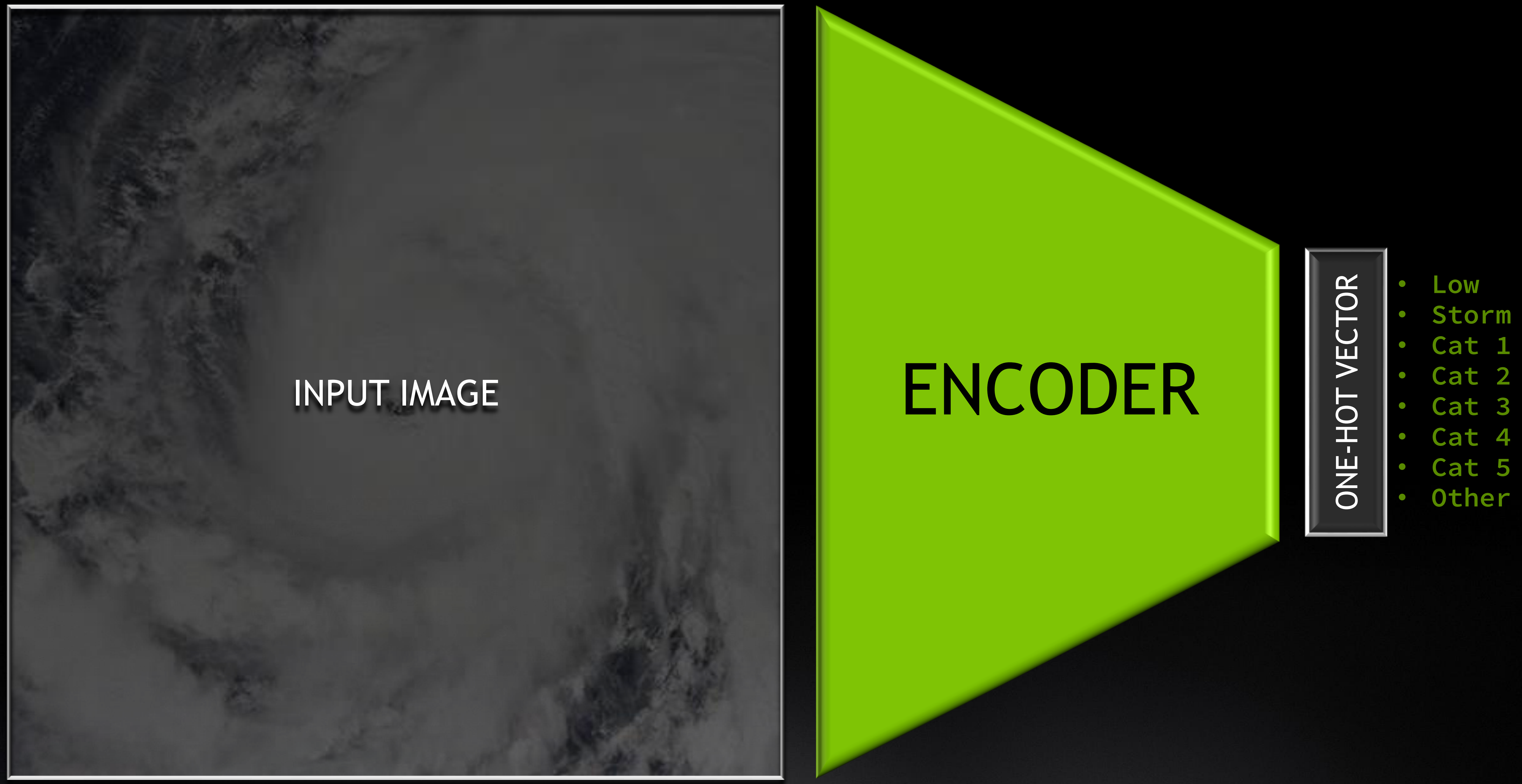
# ENCODER-DECODERS

# ENCODERS AND DECODERS

## Networks connecting high and low dimensional spaces



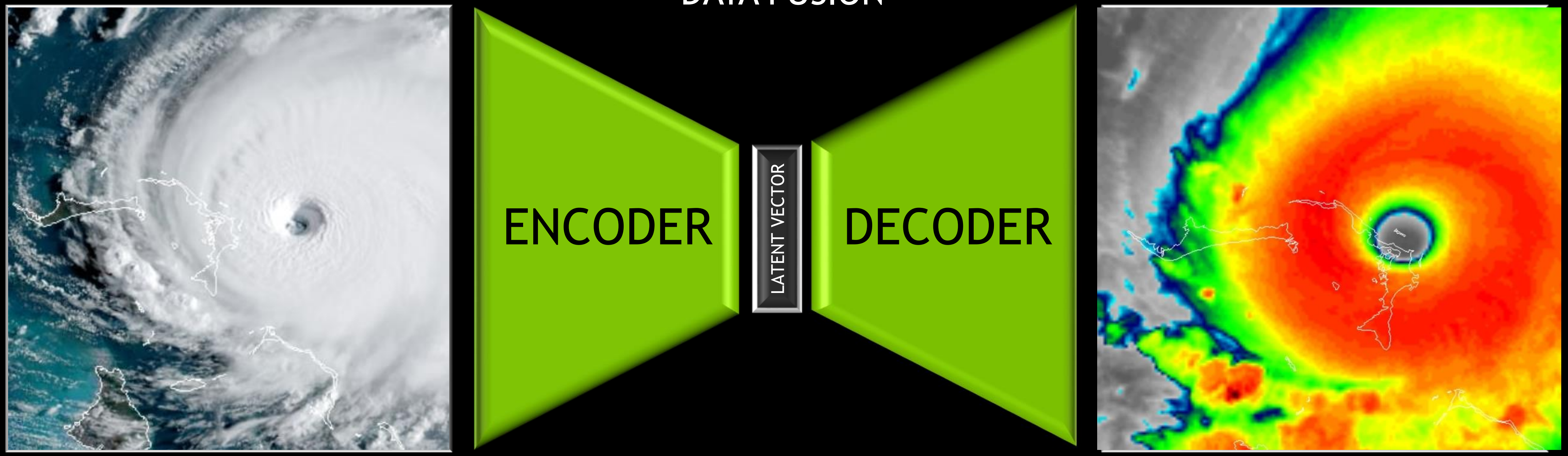
# CLASSIFIER: IMAGE → CLASS ENCODER



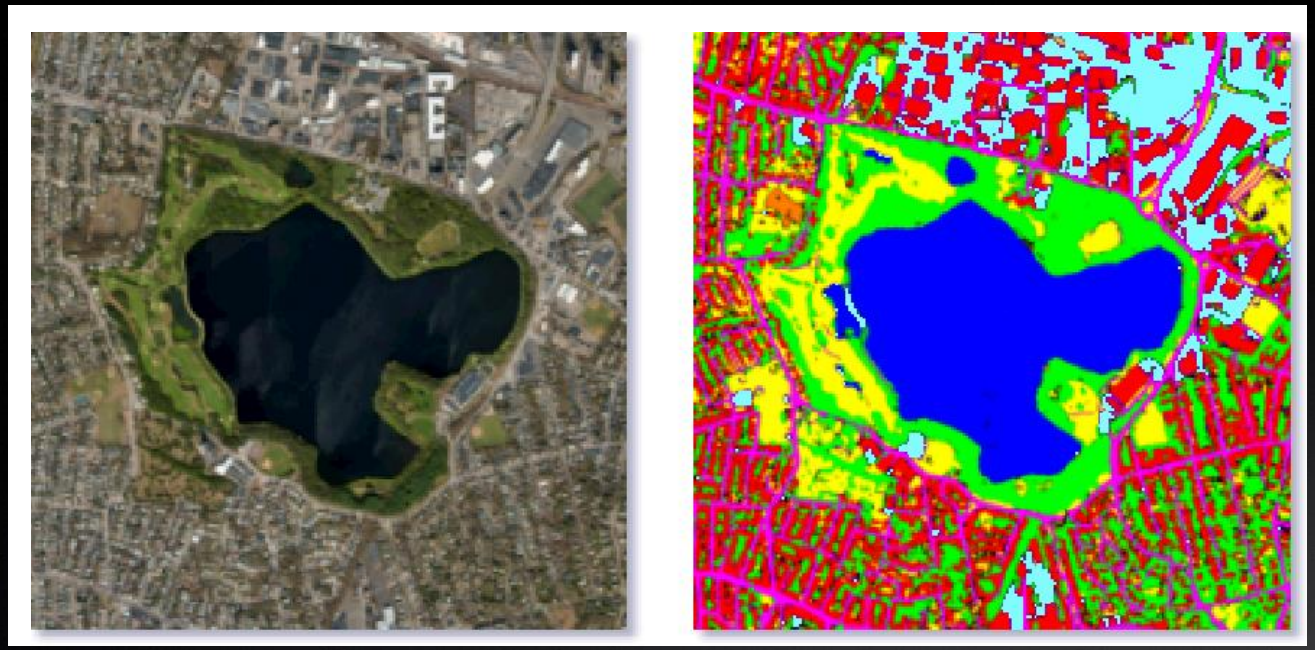
# IMAGE TO IMAGE

## Encoder-Decoder network with Images at both ends

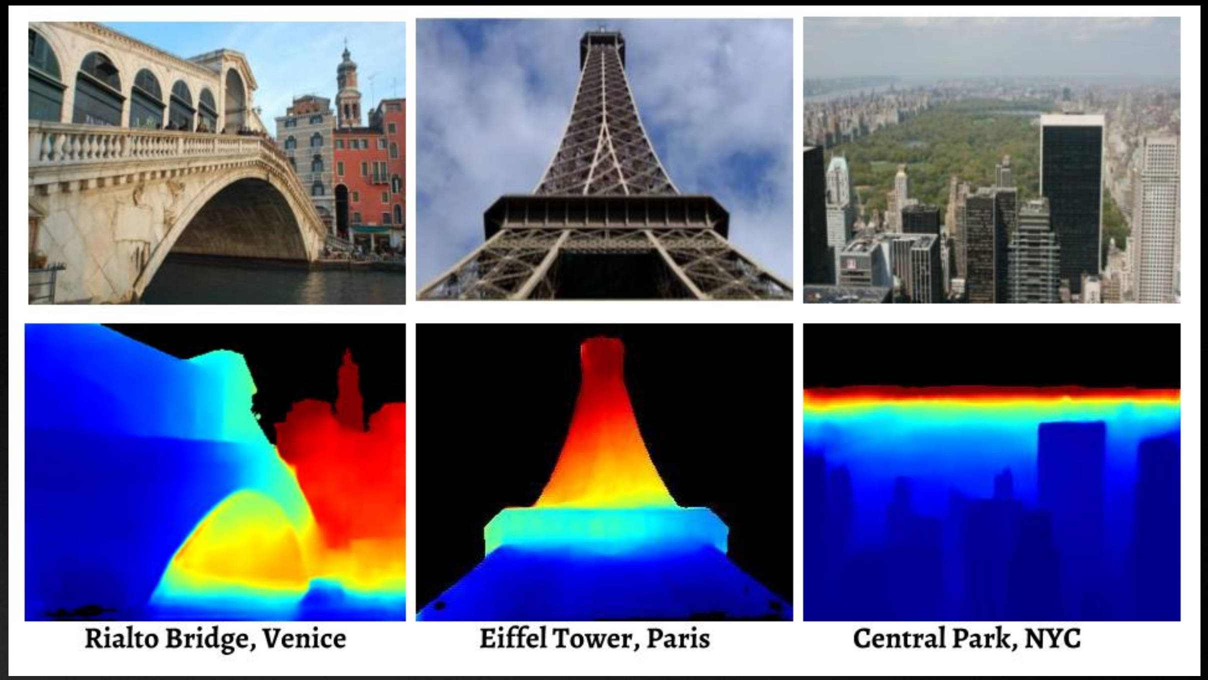
DATA FUSION




SEGMENTATION



DEPTH PREDICTION

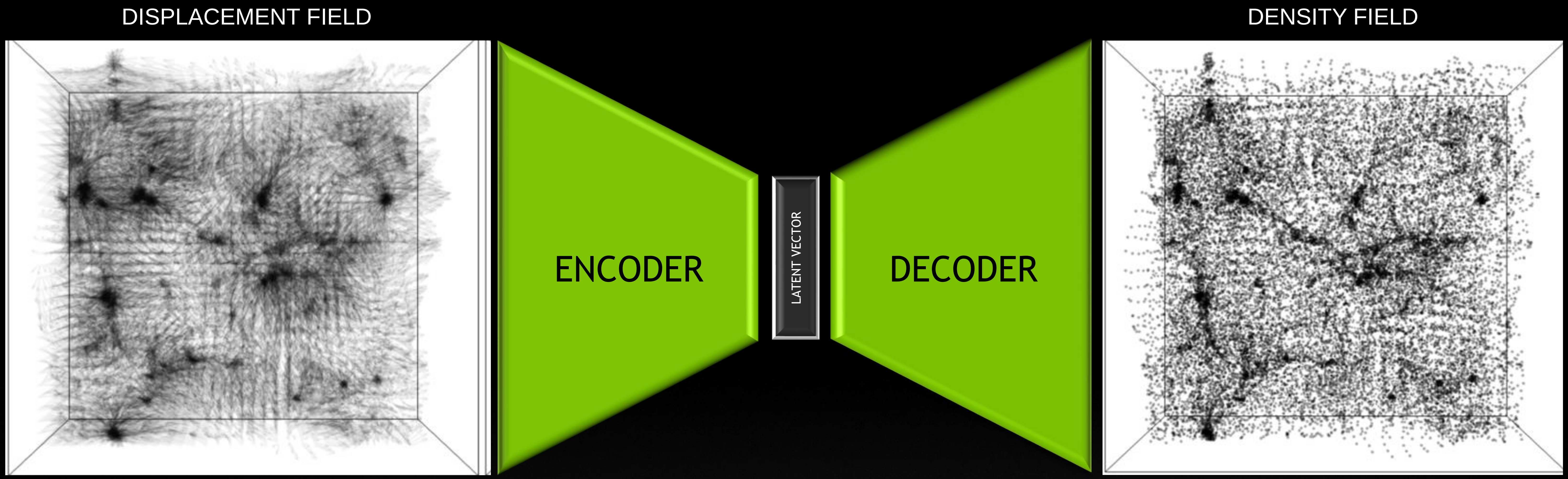



http://liu.diva-portal.org/smash/get/diva2:1182913/FULLTEXT01.pdf

https://research.cs.cornell.edu/megadepth/

# VOLUME TO VOLUME

Input and Output can have 1,2,3 spatial dimensions or more

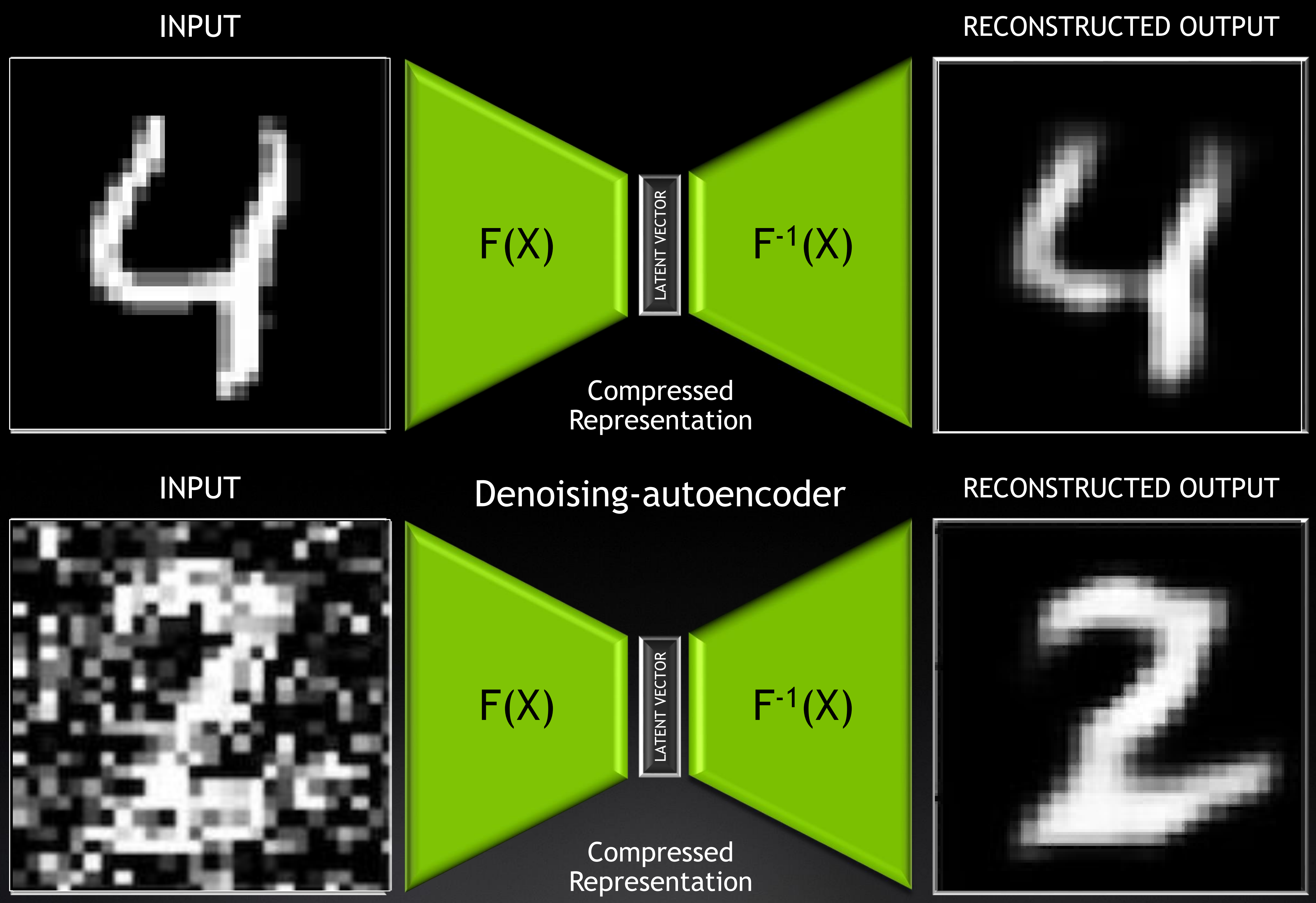



**D3M: Learning to Predict the Cosmological Structure Formation**

https://arxiv.org/pdf/1811.06533.pdf

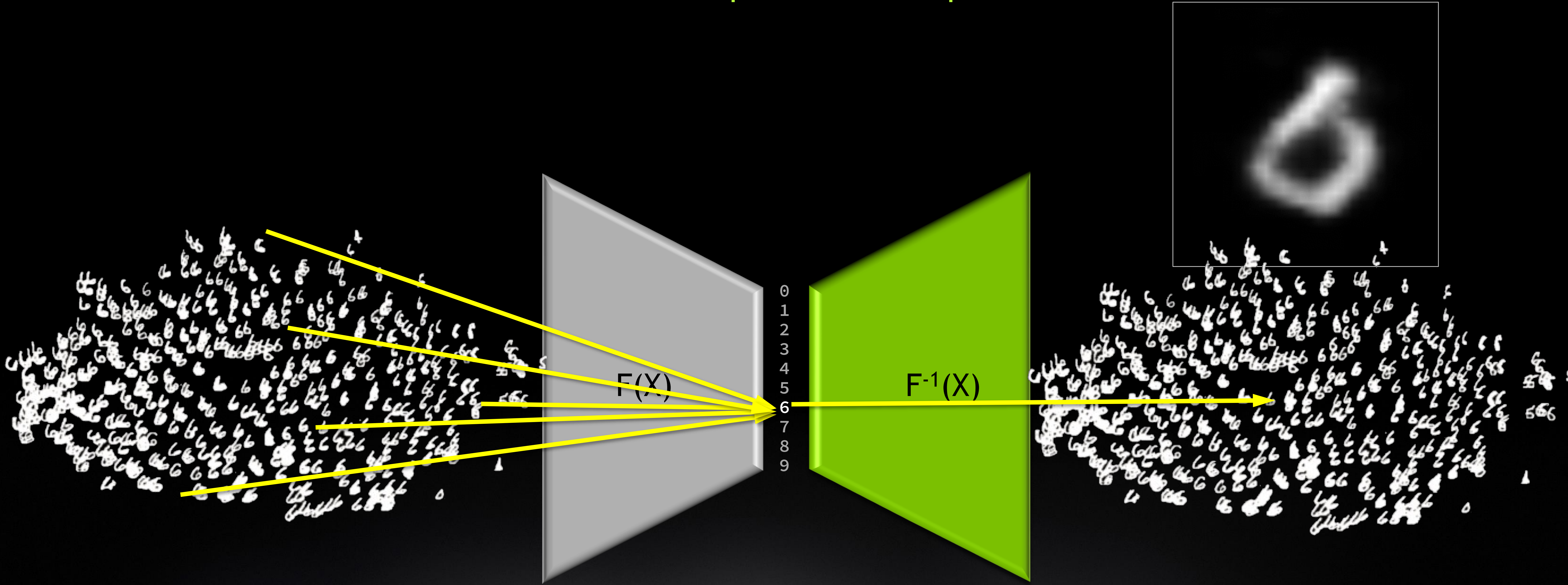
# AUTOENCODER

## Adaptive Data Compression and Noise Removal



# WHY IS MY DECODER OUTPUT FUZZY?

Decoded Output is Not Unique

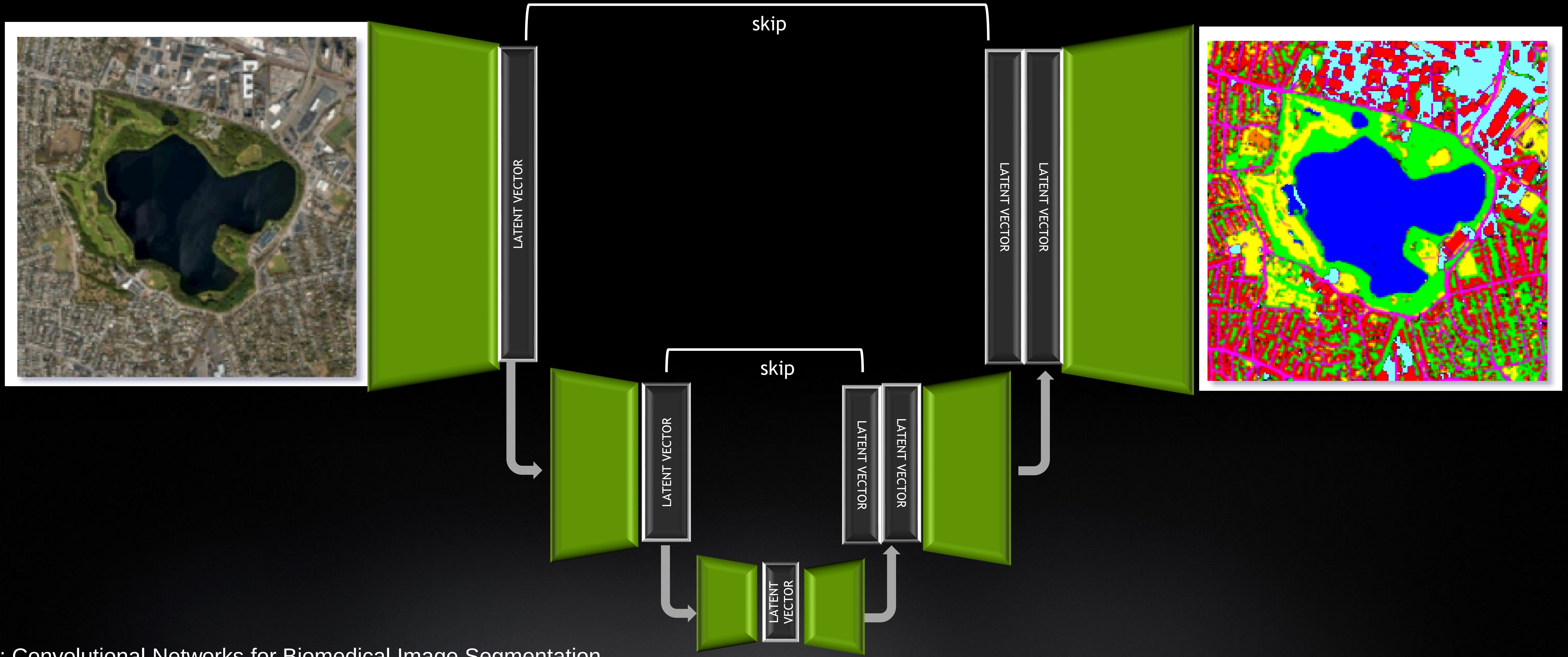



Many instances map to one class

One class represents many instances

# UNET (2015)

## Nested encoder-decoders at multiple spatial scales



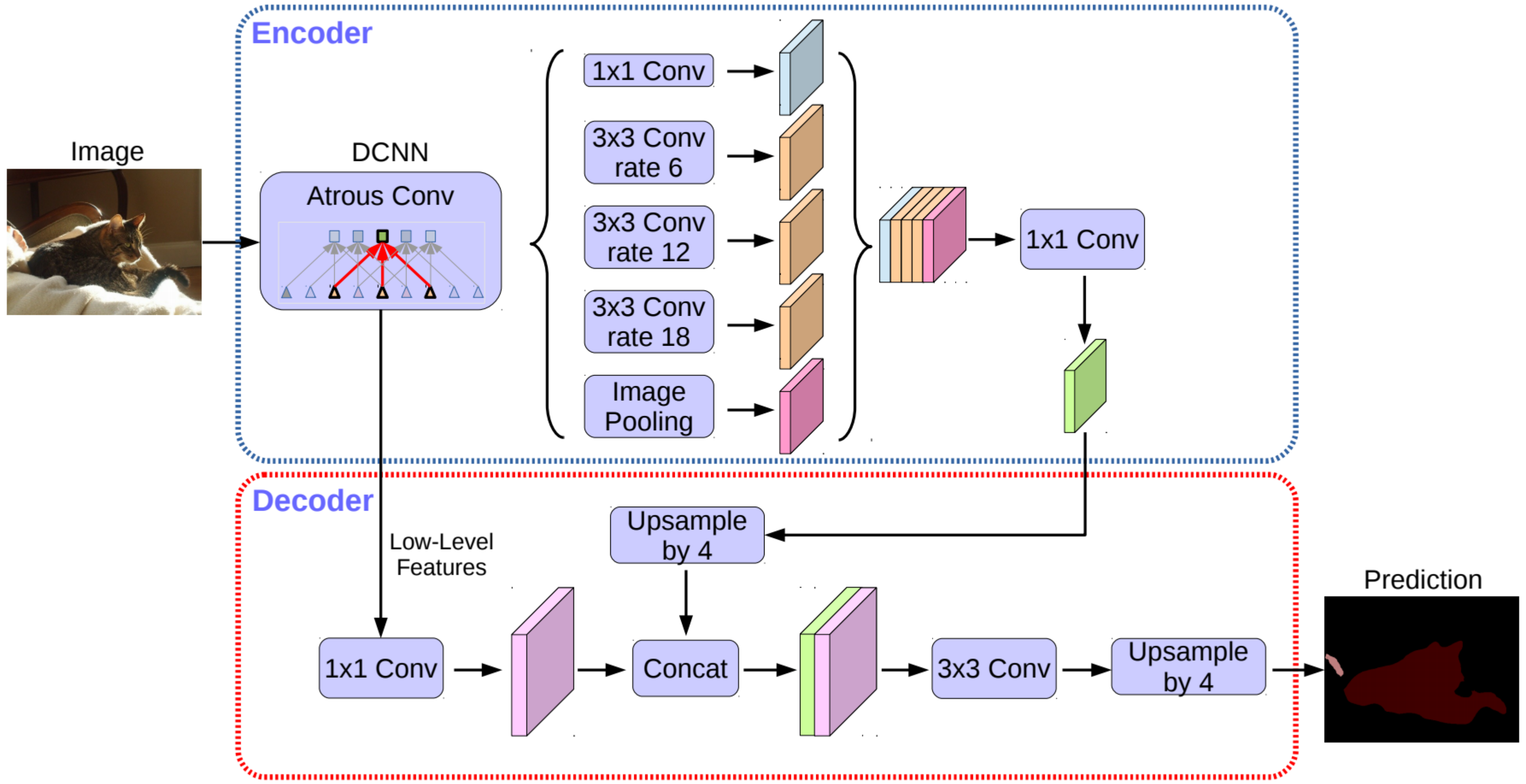

U-Net: Convolutional Networks for Biomedical Image Segmentation
Olaf Ronneberger, Philipp Fischer, and Thomas Brox
https://arxiv.org/pdf/1505.04597.pdf

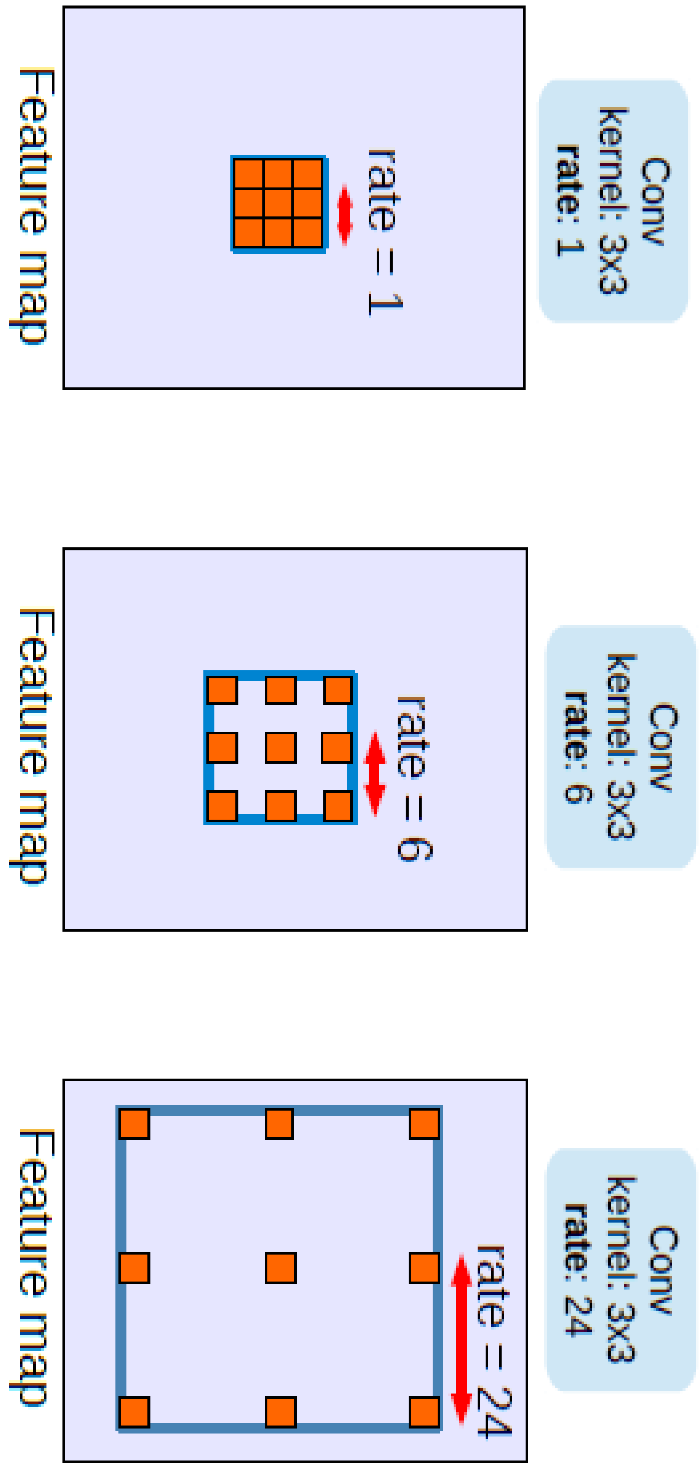
# DEEPLAB V3+

## Another encoder decoder design for accurate segmentation

**Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation**

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam



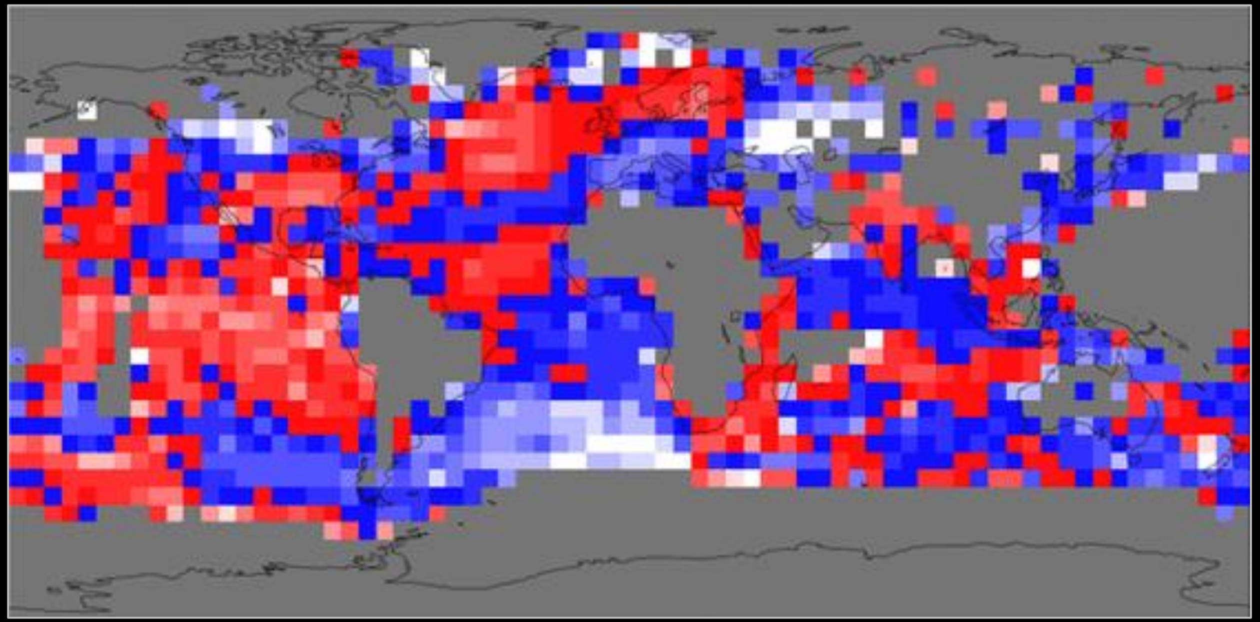



# MASKED CONVOLUTIONS AND INPAINTING

# PARTIAL CONVLUTIONS

## Convolutions that ignore missing or invalid data

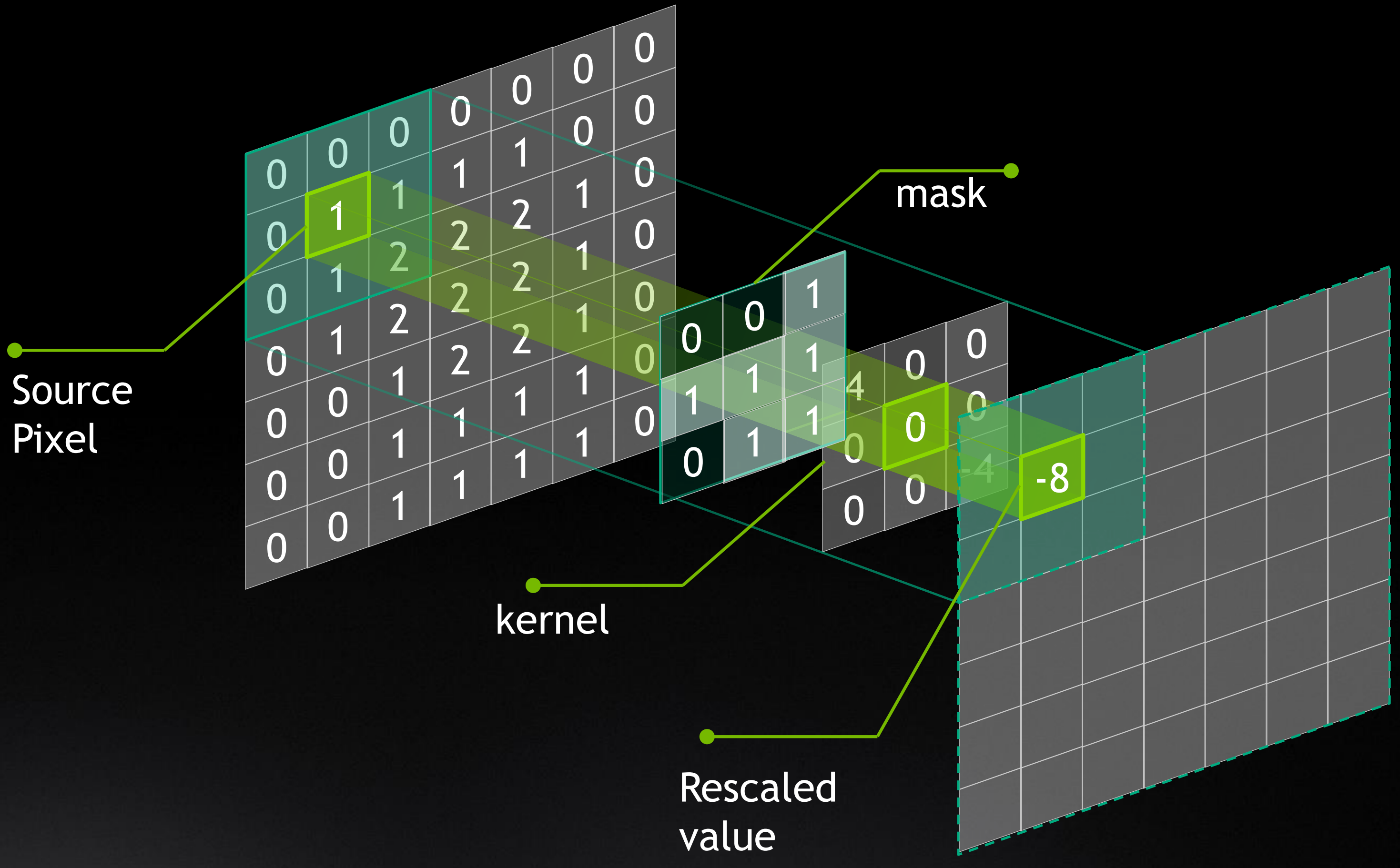
**Data**



**Mask**

### Masked Convolution Operation

$$x' = \begin{cases} \mathbf{W}^T(\mathbf{X} \odot \mathbf{M}) \frac{\mathrm{sum}(\mathbf{1})}{\mathrm{sum}(\mathbf{M})} + b, & \text{if } \mathrm{sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases}$$




https://www.nature.com/articles/s41561-020-0582-5

https://arxiv.org/abs/1804.07723

# TASK: INPAINTING

Repair an image that has missing data

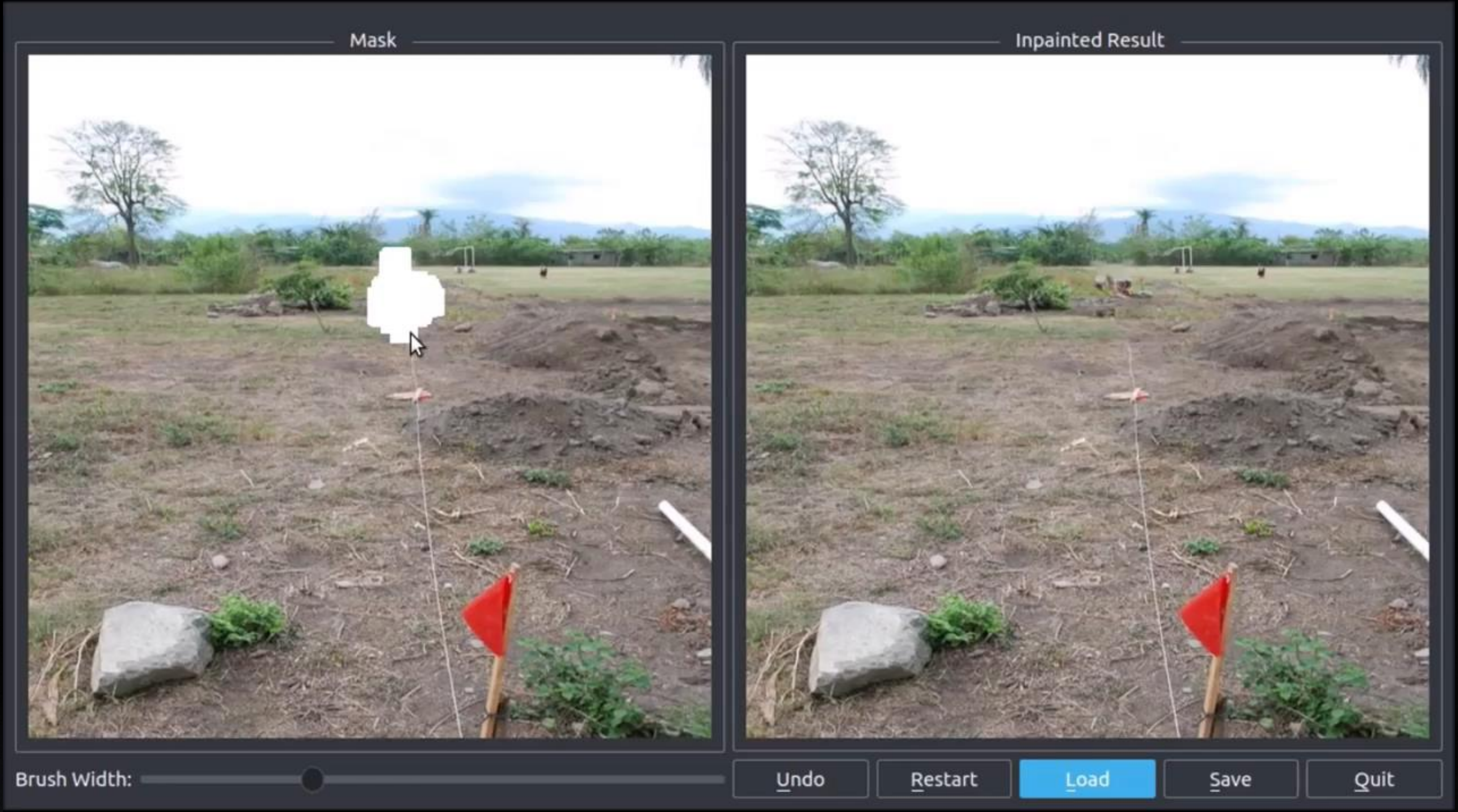


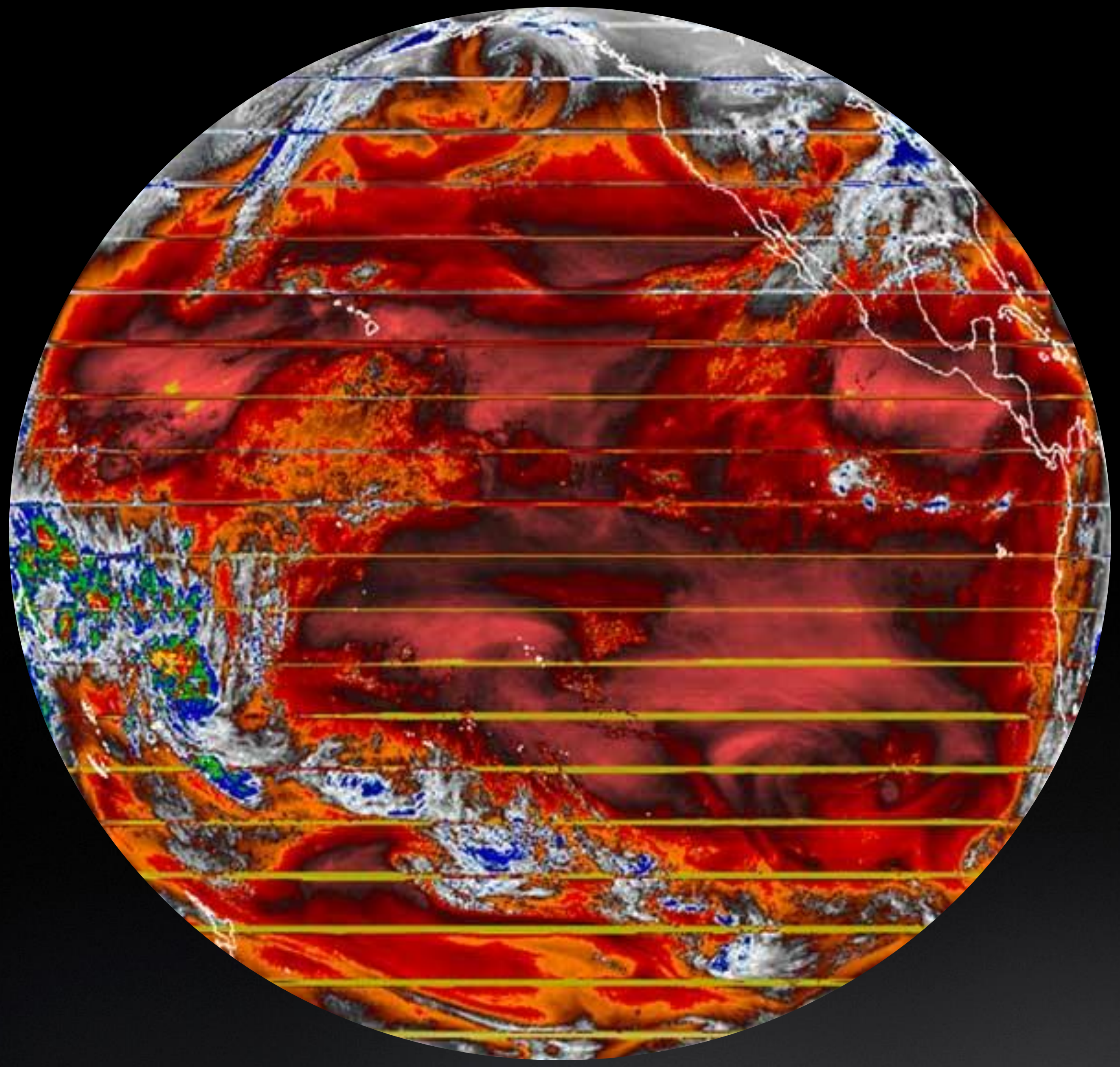

Image Inpainting for Irregular Holes Using Partial Convolutions

Guilin Liu Fitsum A. Reda Kevin J. Shih Ting-Chun Wang Andrew Tao Bryan Catanzaro
NVIDIA Corporation

# INPAINTING APPLICATIONS

## Fill in missing observations, or remove unwanted objects

### GOES-17: Repair Missing Data



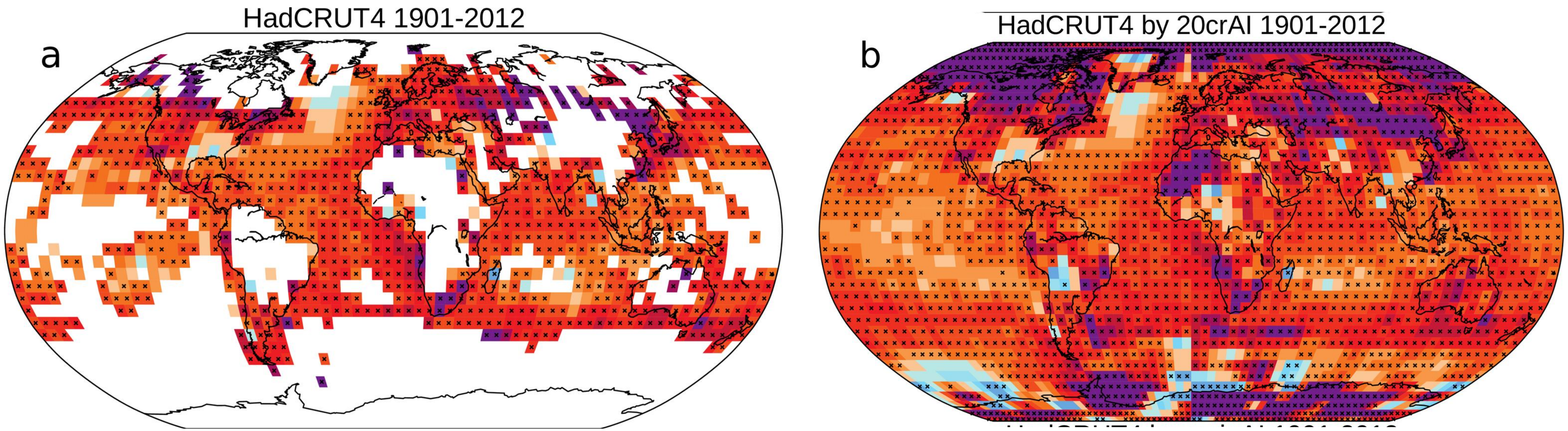
### Remove Clouds, Haze, Shadows

# TRANSFER LEARNING FOR INPAINTING

Train on model data to repair observational data

## Artificial intelligence reconstructs missing climate information

Christopher Kadow[1,2], David Matthew Hall[3] and Uwe Ulbrich[2]




https://www.nature.com/articles/s41561-020-0582-5

# GENERATIVE MODELS

# GENERATIVE MODELS

## Generate Specific Examples from a Learned Distribution

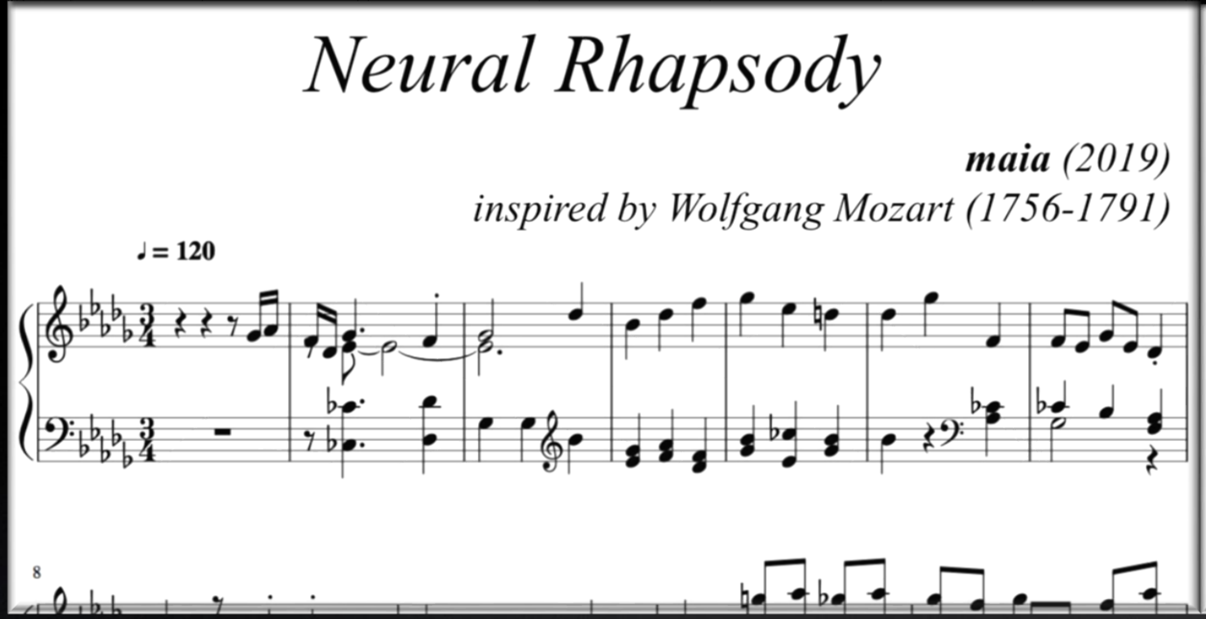

A Style-Based Generator Architecture for Generative Adversarial Networks

https://thispersondoesnotexist.com

https://arxiv.org/pdf/1812.04948.pdf

# GENERATE NEW EXAMPLES

You can generate a new example of nearly any type of data

SAMPLE GENERATION


**Context (human-written):** In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

**GPT-2:** The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

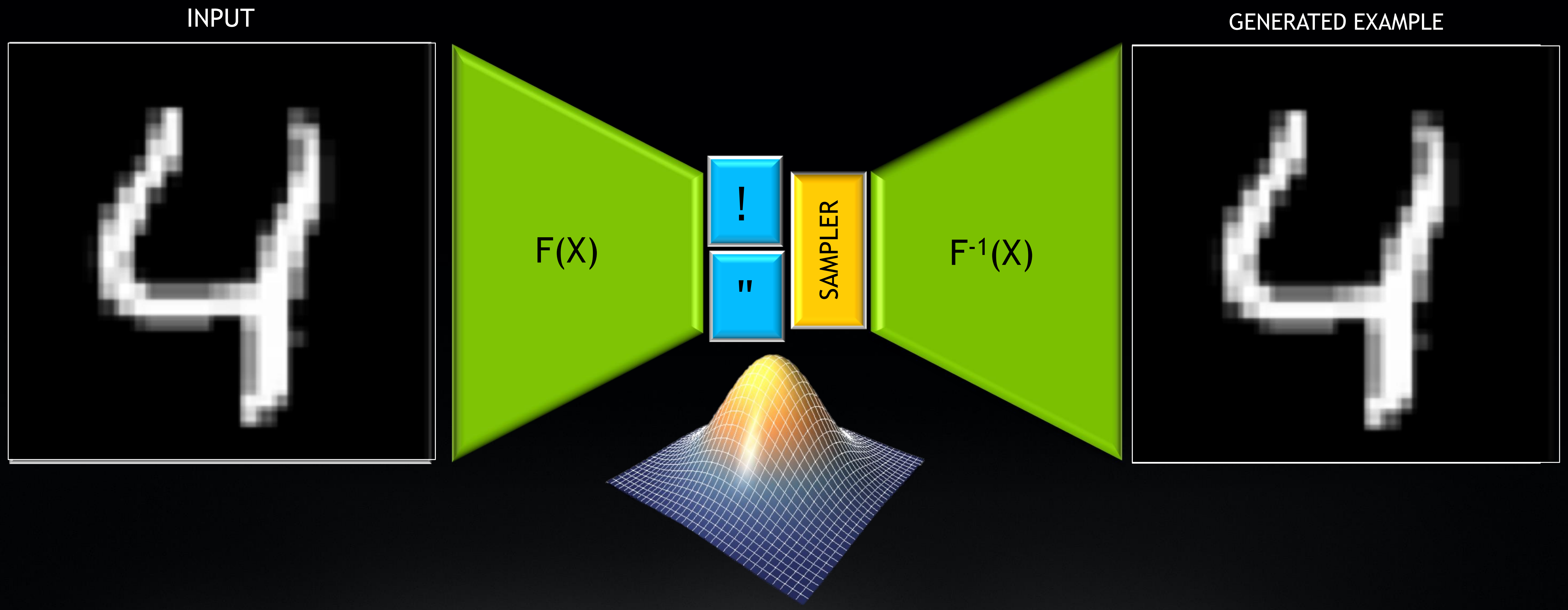
Pérez and the others then ventured further into the valley. "By the time we reached the top



NVIDIA

# VAE: VARIATIONAL AUTOENCODER

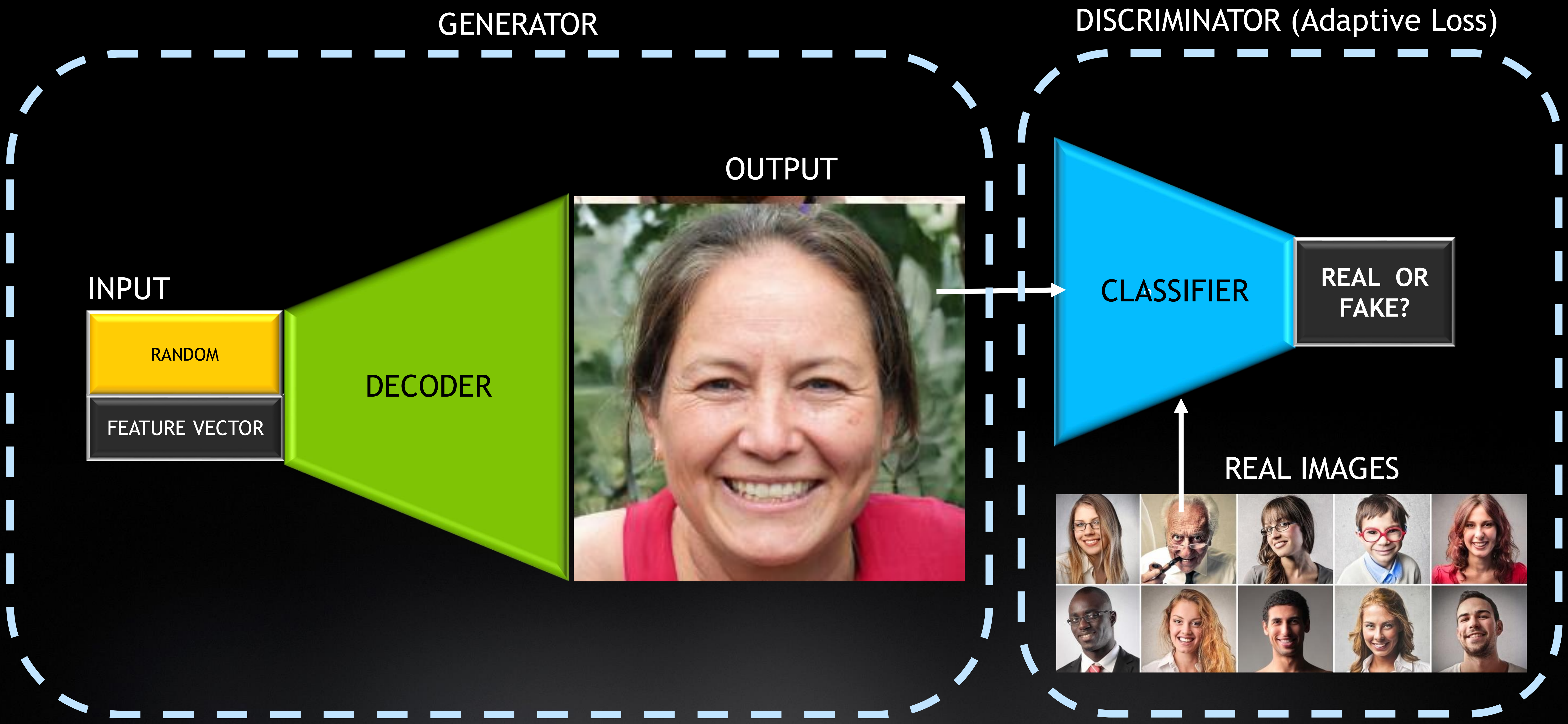
An autoencoder that learns Gaussian Distributions




https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

# GAN: GENERATIVE ADVERSARIAL NETWORK

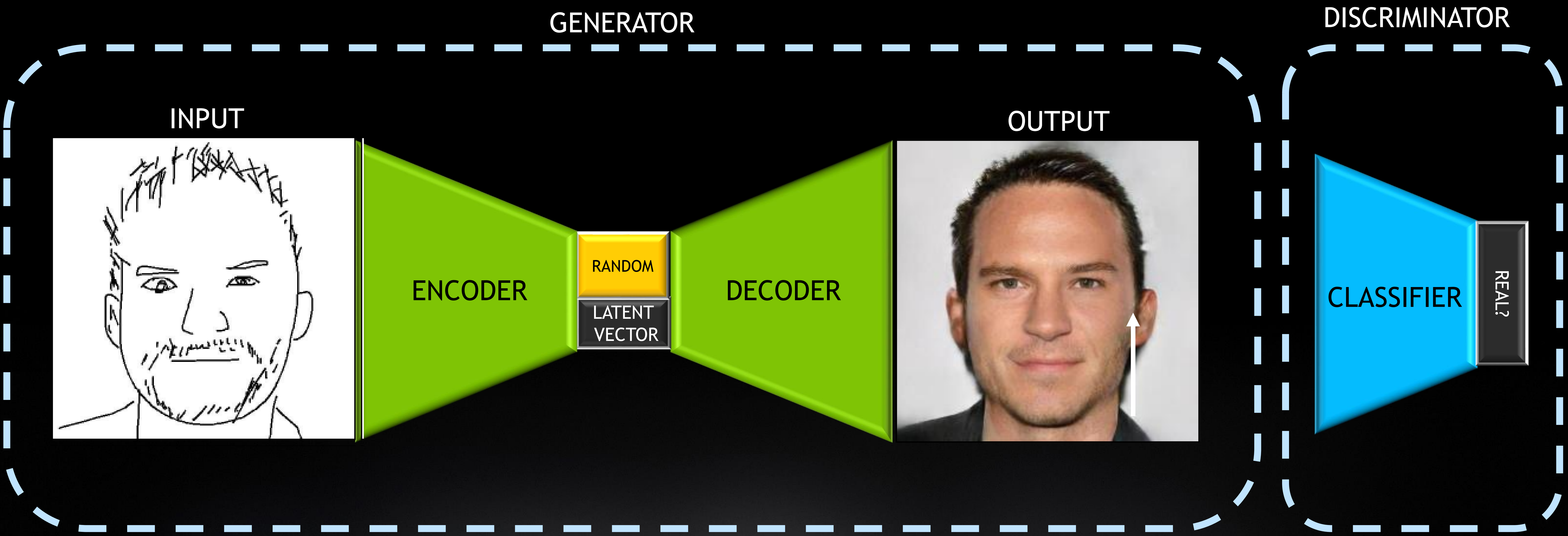
A trick for training a decoder to produce samples indistinguishable from real ones




https://en.wikipedia.org/wiki/Generative_adversarial_network

# CONDITIONAL GAN

## Generate Synthetic Images, conditioned upon the input

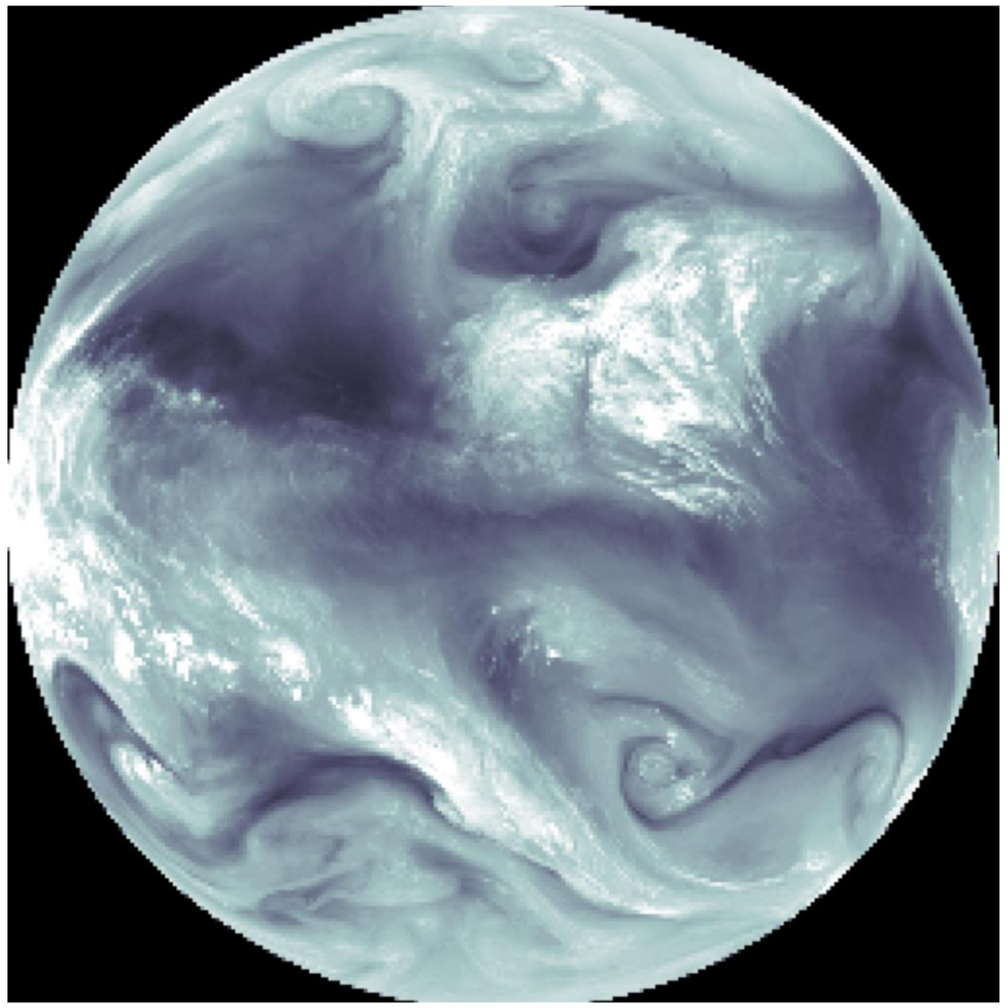



DeepFaceDrawing: Deep Generation of Face Images from Sketches
https://arxiv.org/pdf/2006.01047.pdf

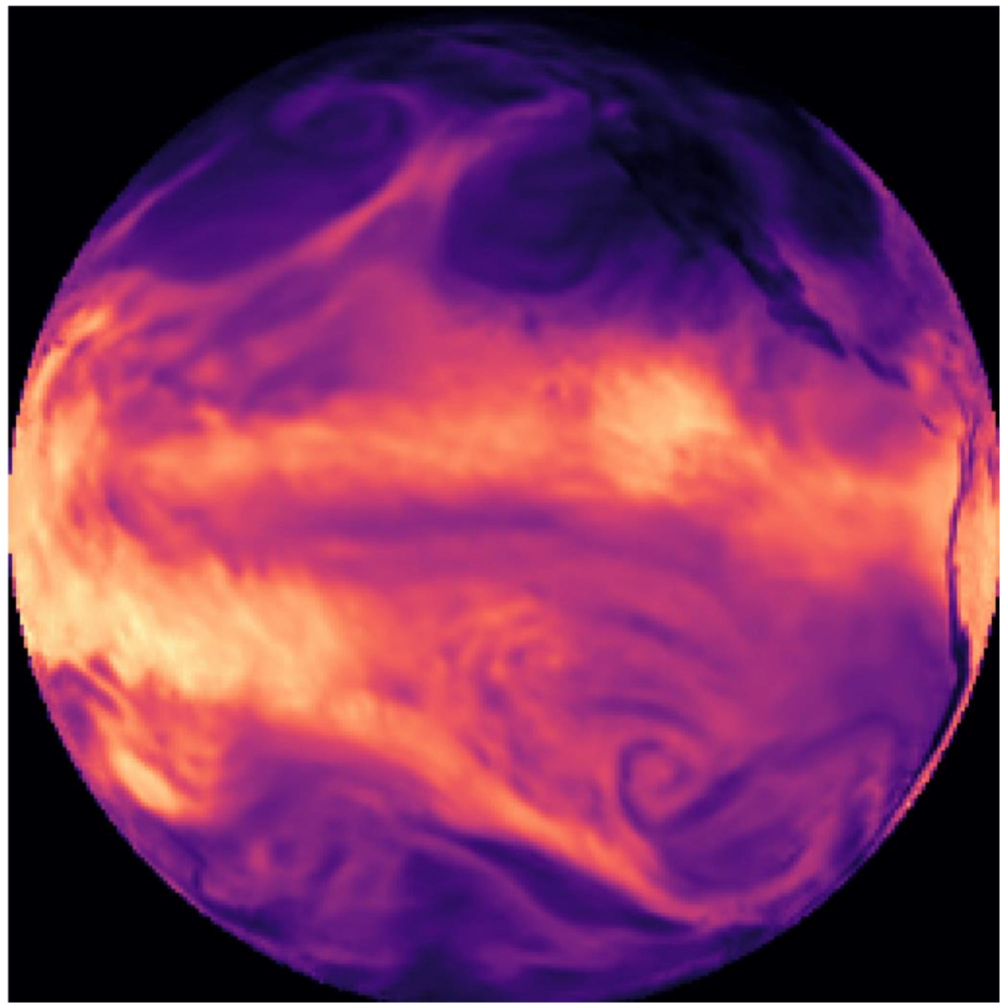
# CONDITIONAL GAN EXAMPLE

## Map from satellite observations to model variables

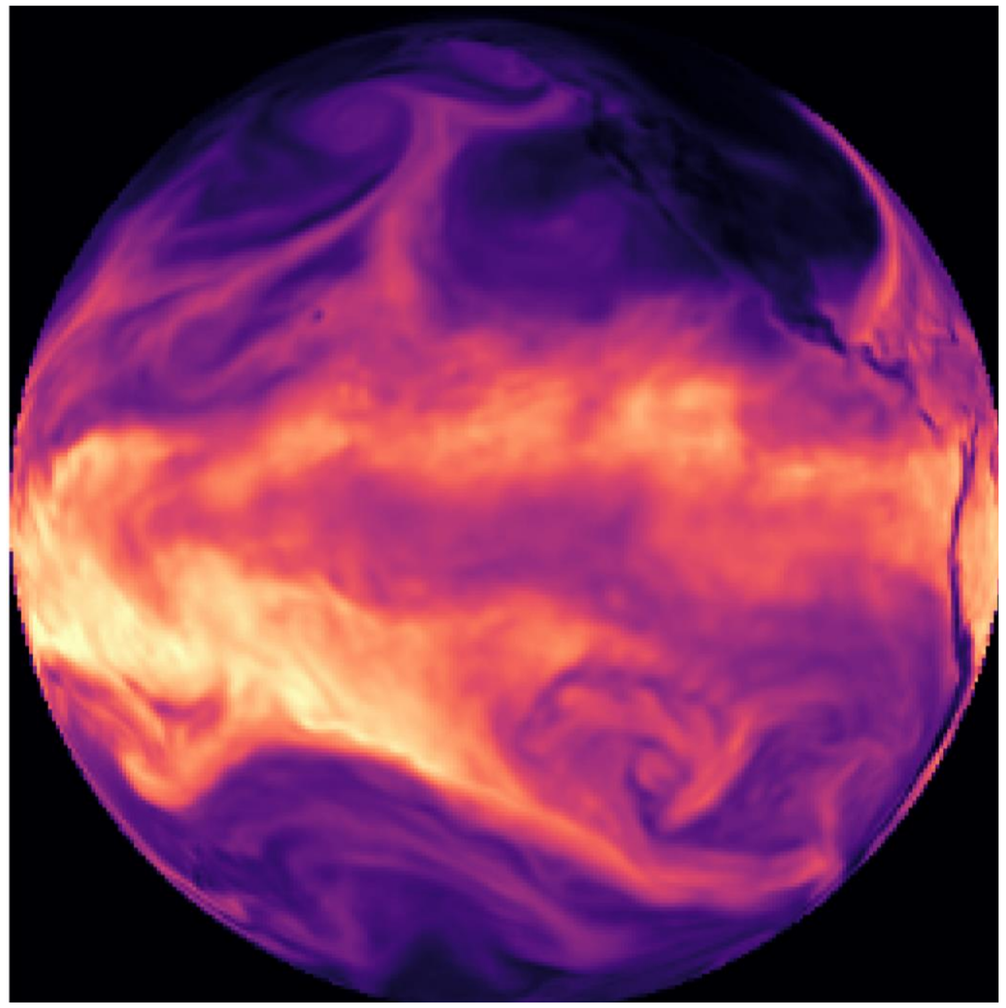
GOES-15 BAND 3



GENERATED WATER VAPOR



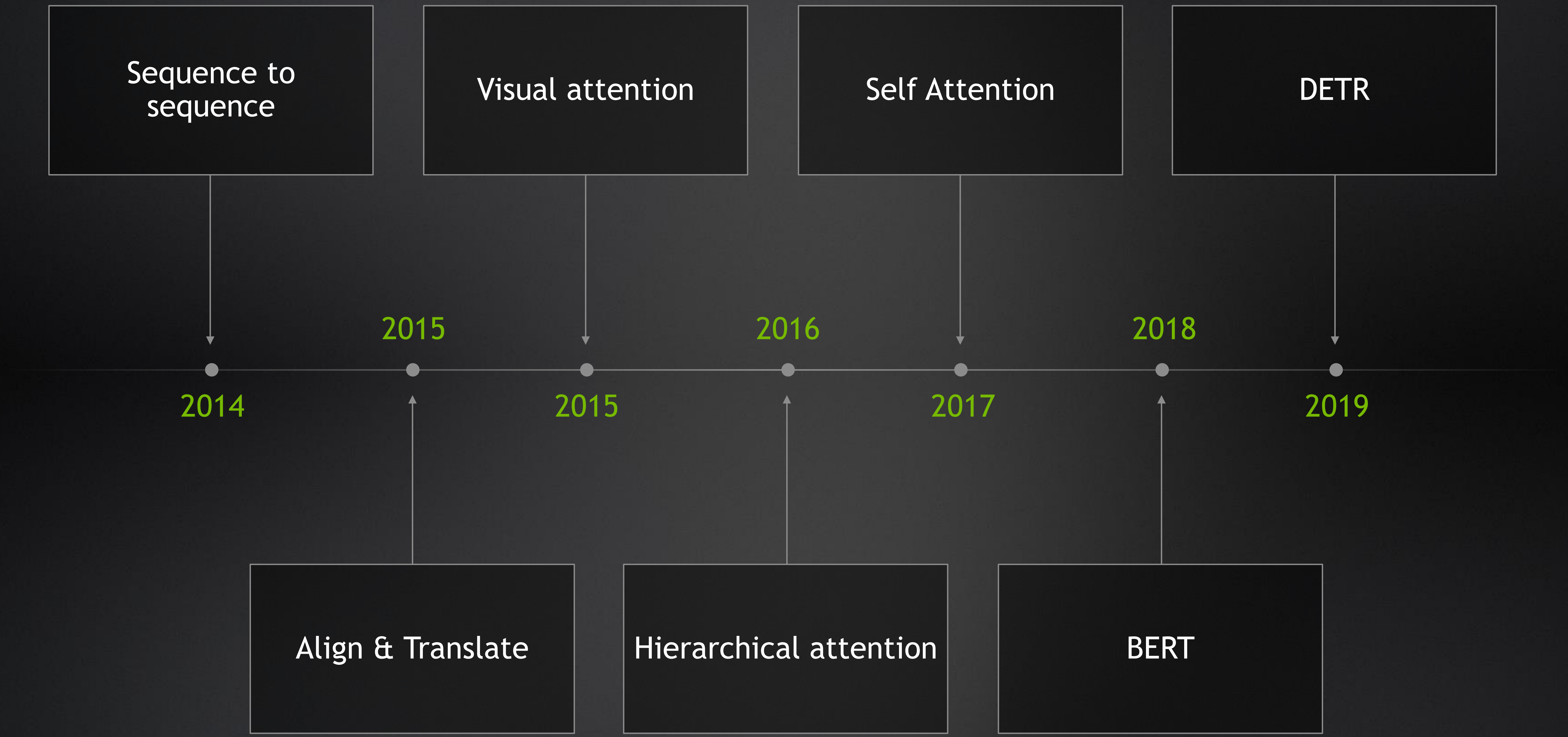
TARGET: GFS WATER VAPOR



doi:10.5065/D6BZ64XQ

# RNNS, ATTENTION, AND TRANSFORMERS

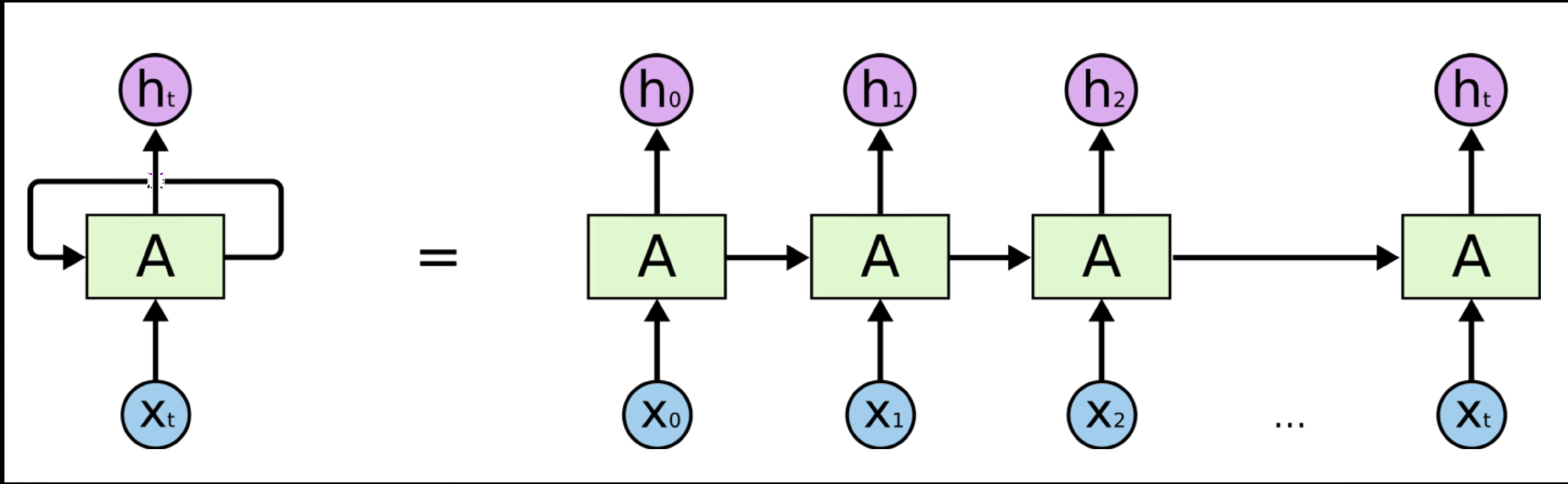
# ATTENTION AND TRANSFORMERS




*Source: Source information is 14 pt, italic*

# RECURRENT NEURAL NETS

## (1986) Neural networks for sequences




RNN, Unrolled over time

https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

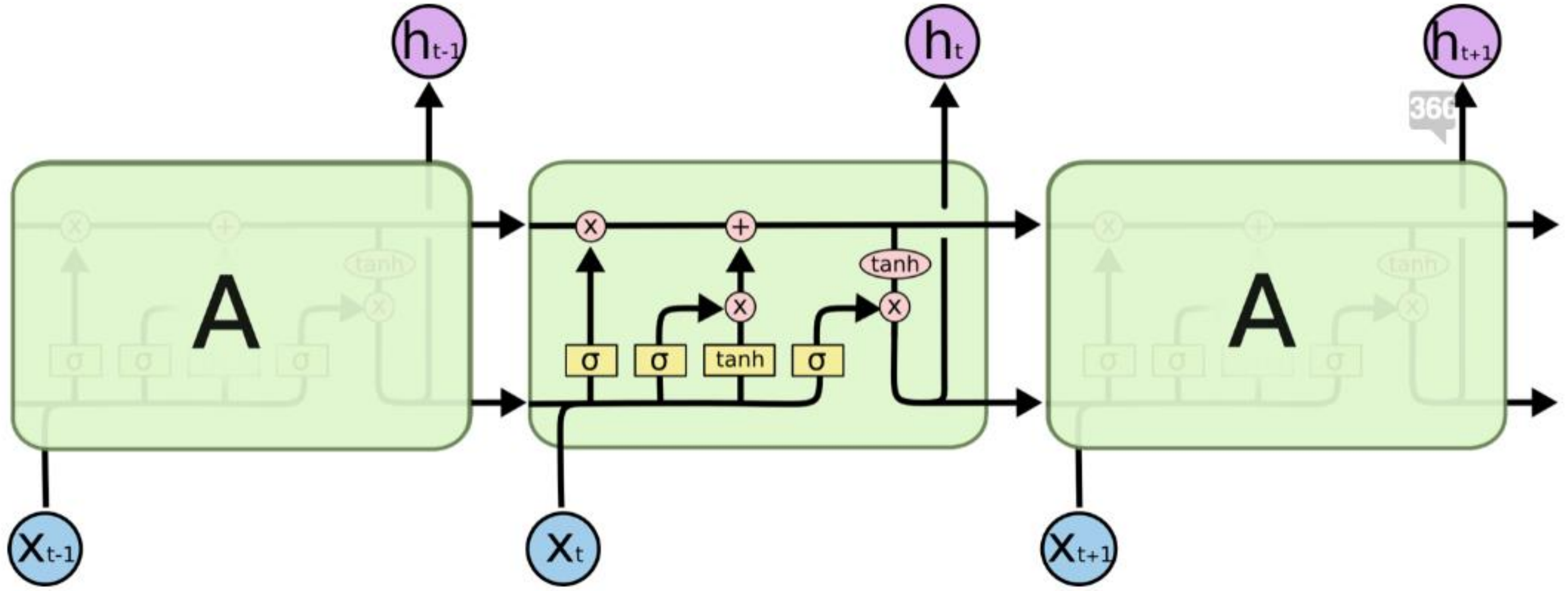
# LSTM

## (1997) Long Short-term Memory Units

LONG SHORT-TERM MEMORY

NEURAL COMPUTATION 9(8):1735–1780, 1997

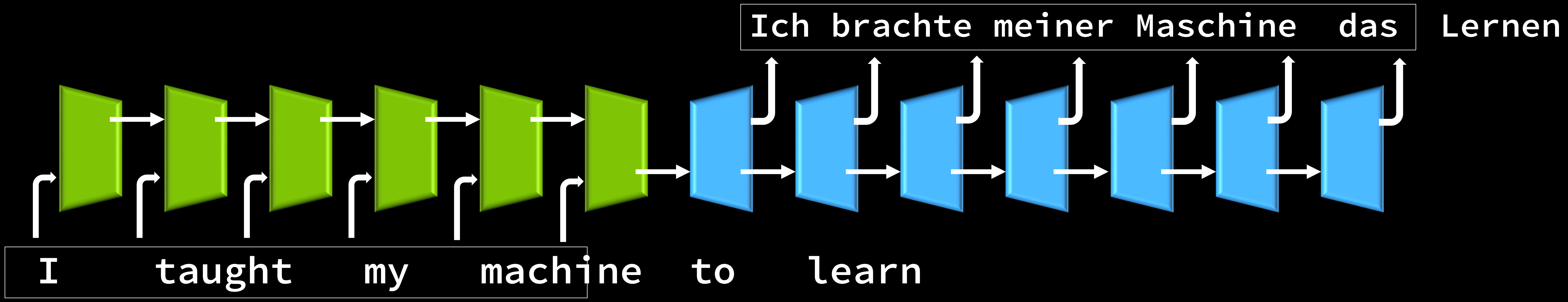
Sepp Hochreiter    Jürgen Schmidhuber




https://colah.github.io/posts/2015-08-Understanding-LSTMs/

# SEQUENCE TO SEQUENCE

(2014) Encoder-decoder pattern for sequences




**Sequence to Sequence Learning with Neural Networks**

---

**Ilya Sutskever**
Google

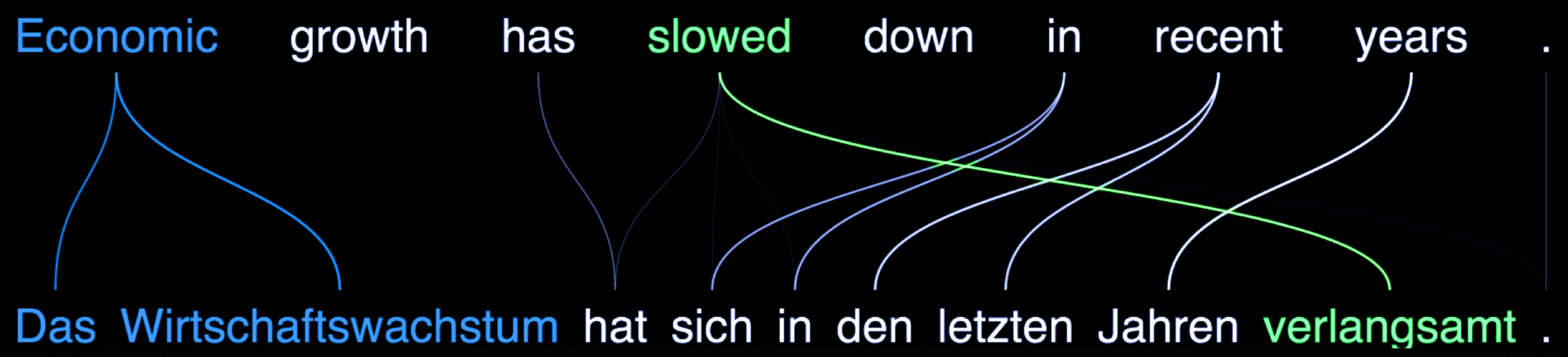
**Oriol Vinyals**
Google

**Quoc V. Le**
Google

https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

# ATTENTION

Adjustable weights based on context

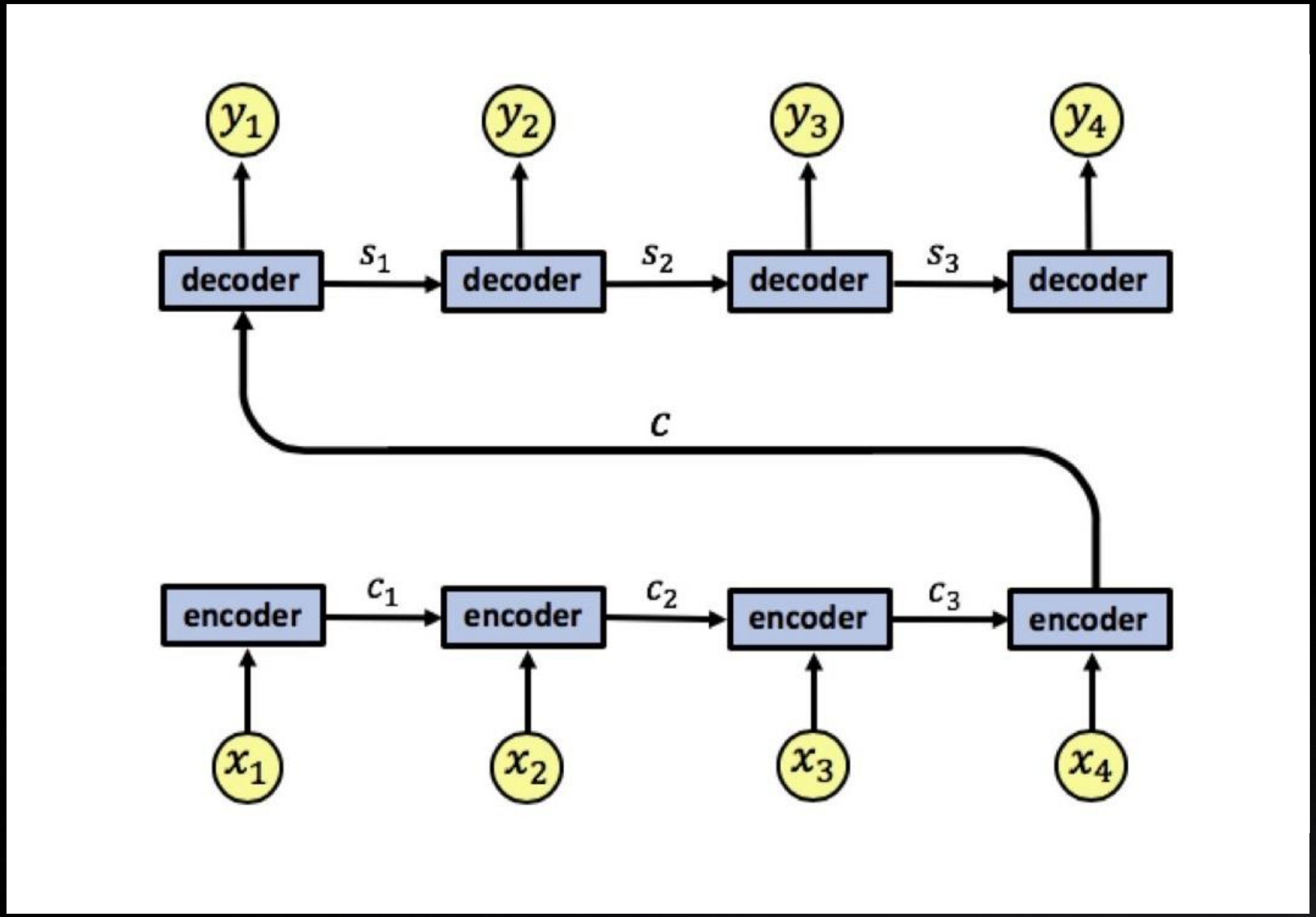



https://www.datasciencecentral.com/profiles/blogs/seq2seq
https://devblogs.nvidia.com/introduction-neural-machine-translation-gpus-part-3/

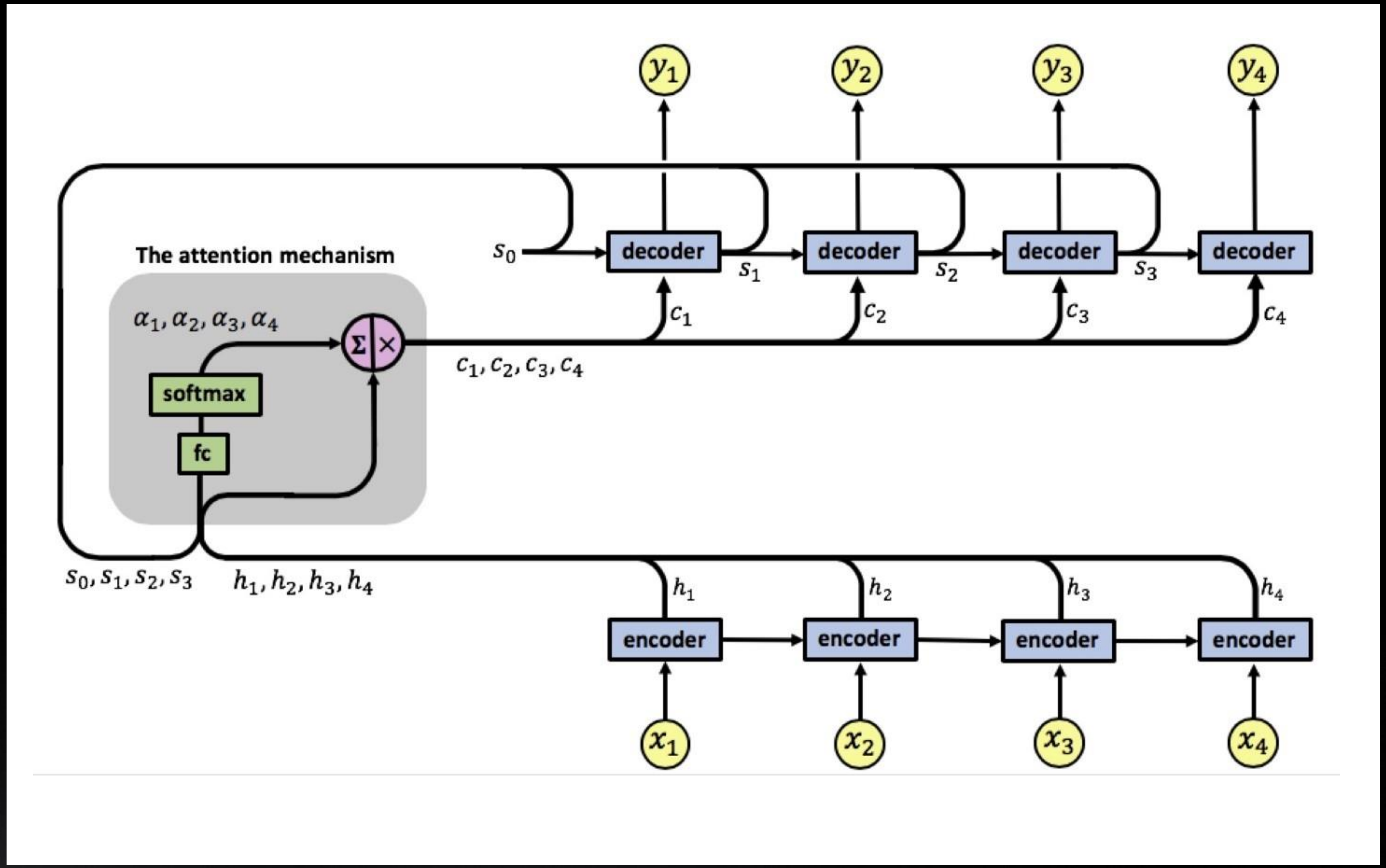
# ATTENTION ENCODER-DECODER

A more accurate way to translate long sequences

RNN




RNN + ATTENTION




https://medium.com/datadriveninvestor/attention-in-rnns-321fbcd64f05

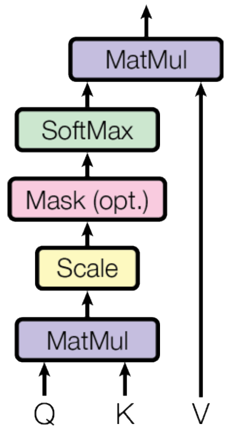
# THE TRANSFORMER

## (2017) Attention is all you need!

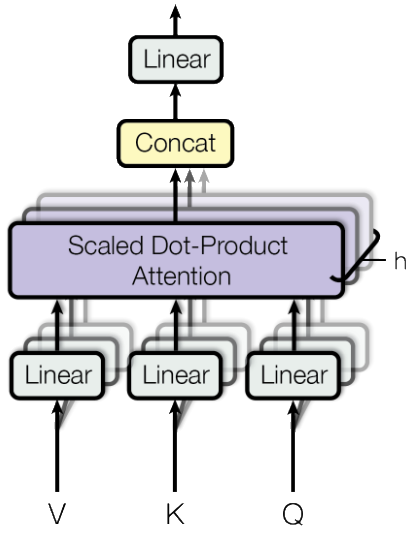
# Attention Is All You Need

**Ashish Vaswani*** Google Brain avaswani@google.com
**Noam Shazeer*** Google Brain noam@google.com
**Niki Parmar*** Google Research nikip@google.com
**Jakob Uszkoreit*** Google Research usz@google.com
**Llion Jones*** Google Research llion@google.com
**Aidan N. Gomez* †** University of Toronto aidan@cs.toronto.edu
**Łukasz Kaiser*** Google Brain lukaszkaiser@google.com
**Illia Polosukhin* ‡** illia.polosukhin@gmail.com

Scaled Dot-Product Attention




Multi-Head Attention



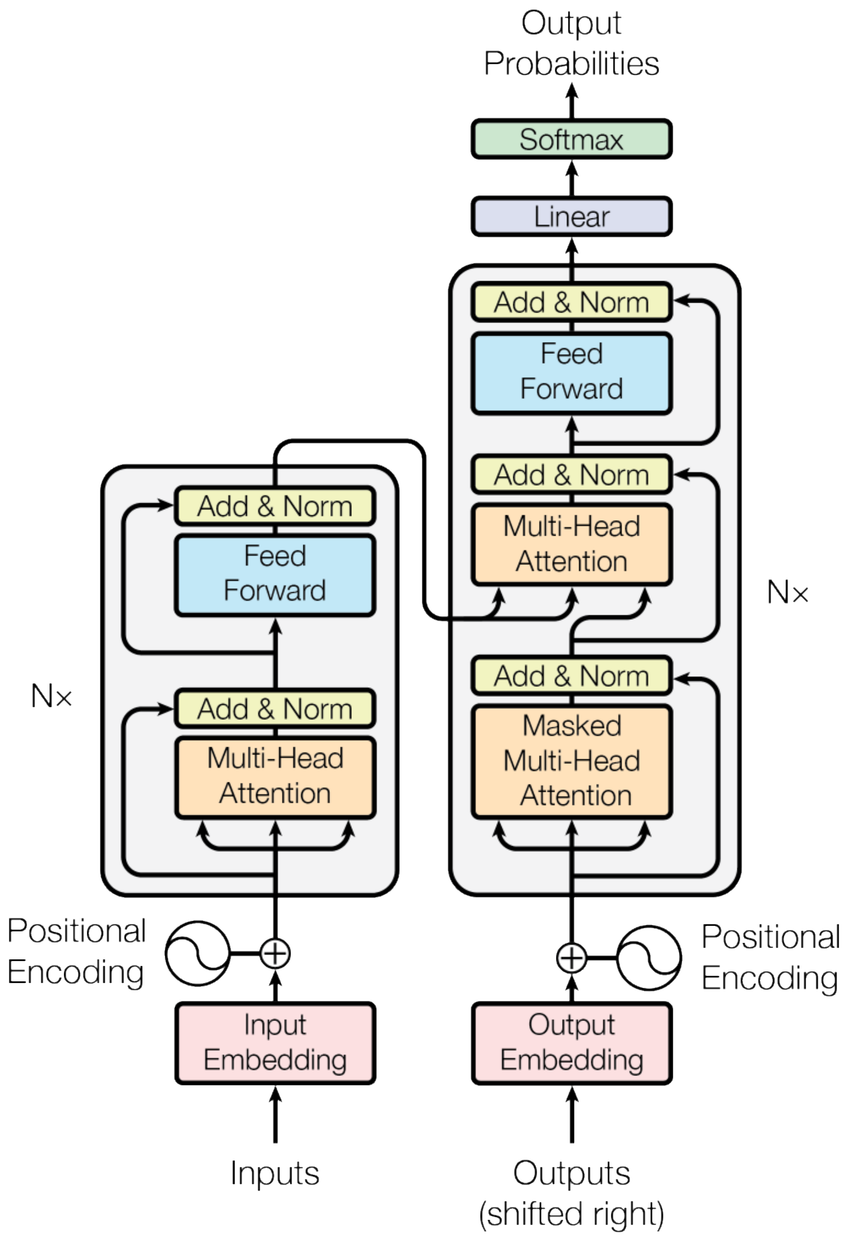
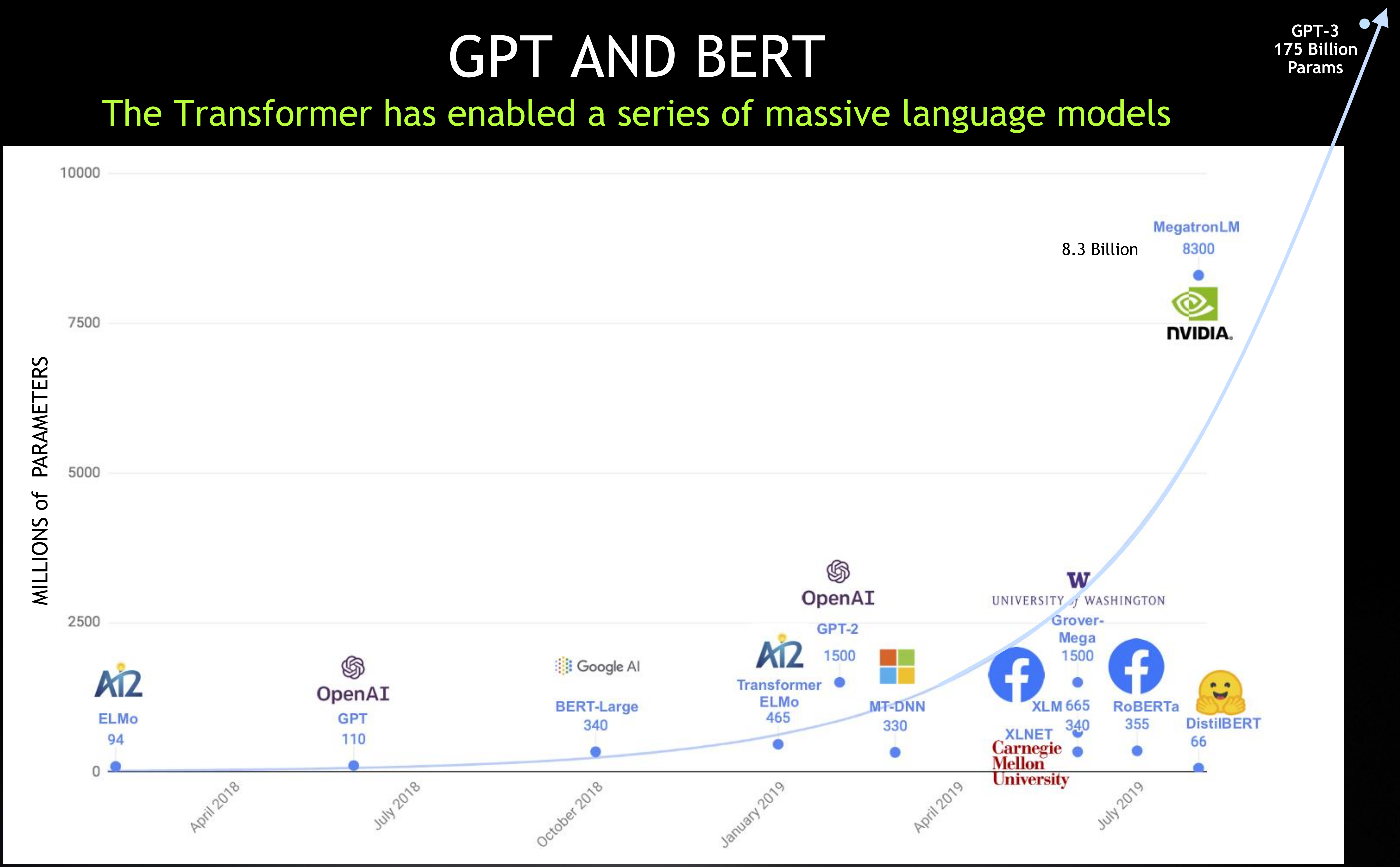




Figure 1: The Transformer - model architecture.

http://jalammar.github.io/illustrated-transformer/ https://arxiv.org/abs/1706.03762

# GPT AND BERT

## The Transformer has enabled a series of massive language models




best language papers encoder-decoder-transformers

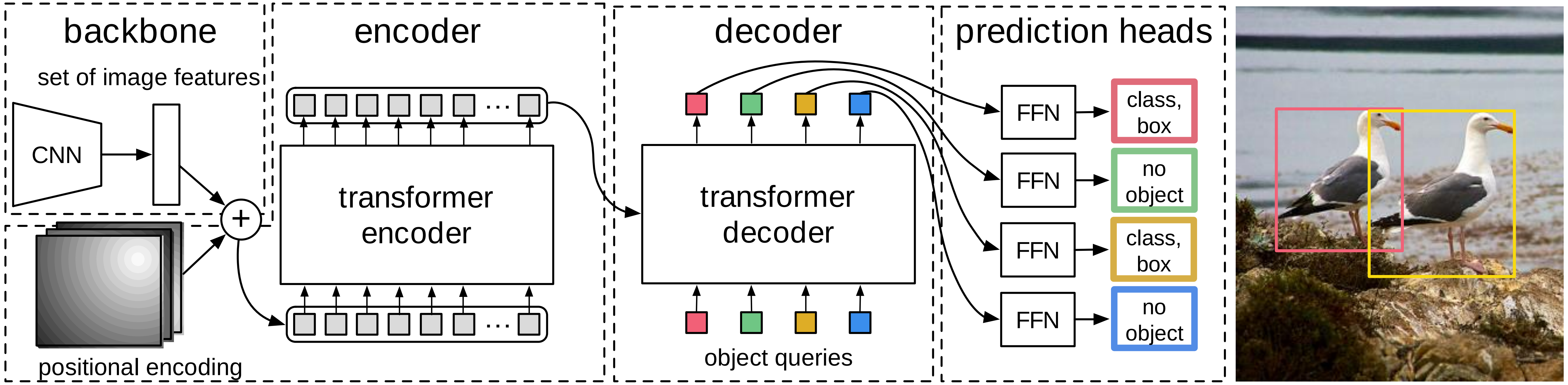
# DETNET

## (2020) End-to-end object detection with transformers

### End-to-End Object Detection with Transformers

Nicolas Carion*, Francisco Massa*, Gabriel Synnaeve, Nicolas Usunier,
Alexander Kirillov, and Sergey Zagoruyko

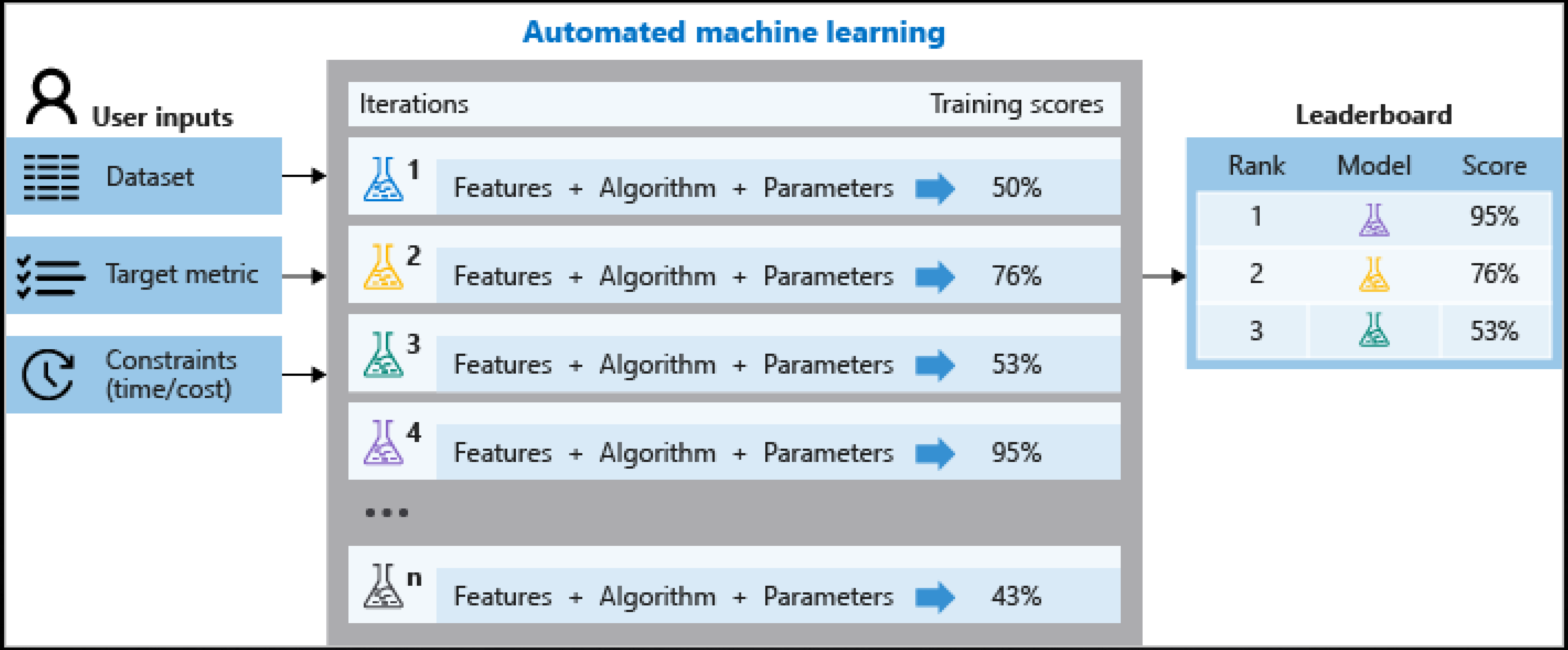
Facebook AI



# AUTO ML

# AUTO ML

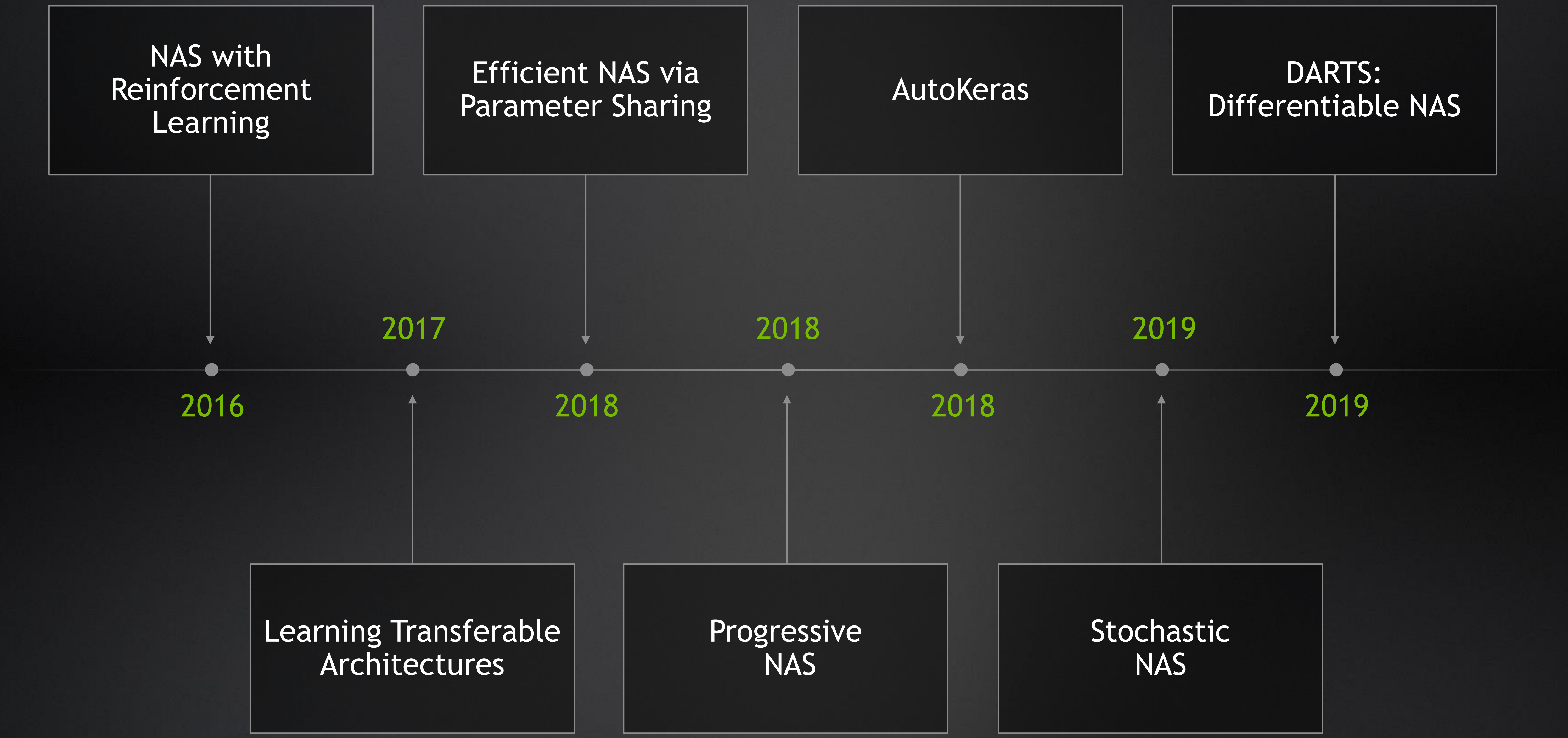
## Automating the work of the data scientist




https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml

# NEURAL ARCHITECTURE SEARCH

## Automating model design and selection



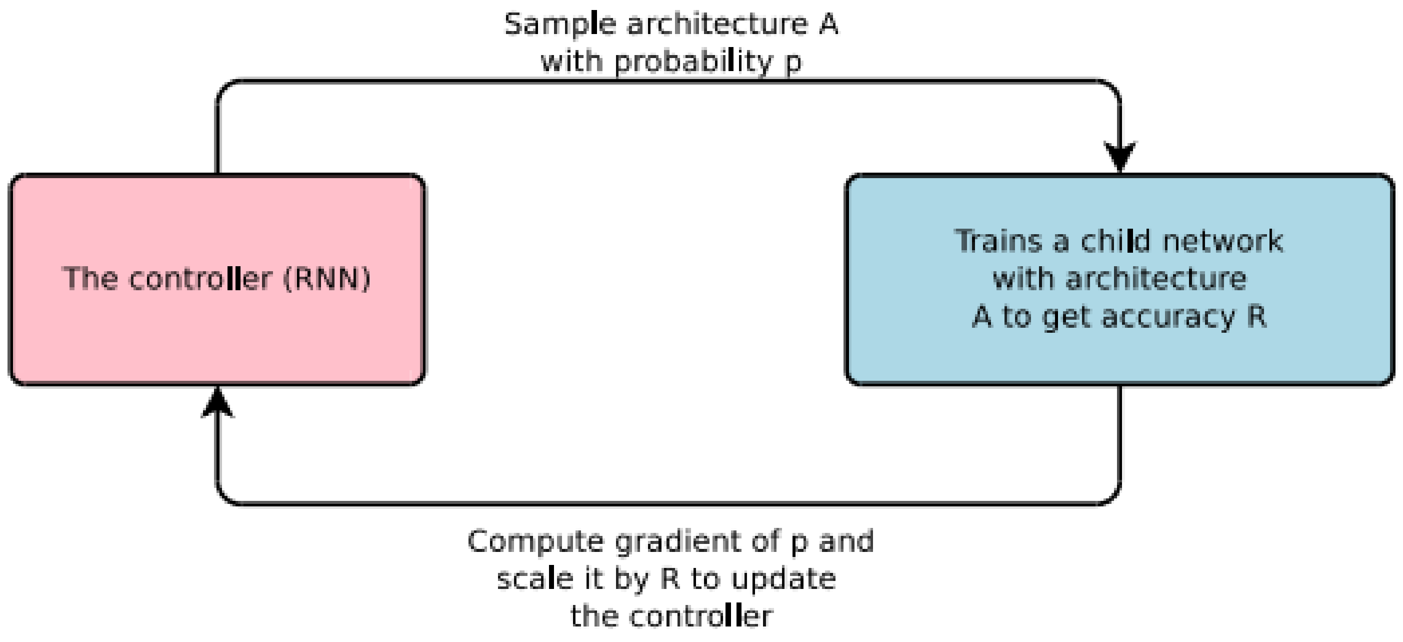

https://heartbeat.fritz.ai/research-guide-for-neural-architecture-search-b250c5b1b2e5

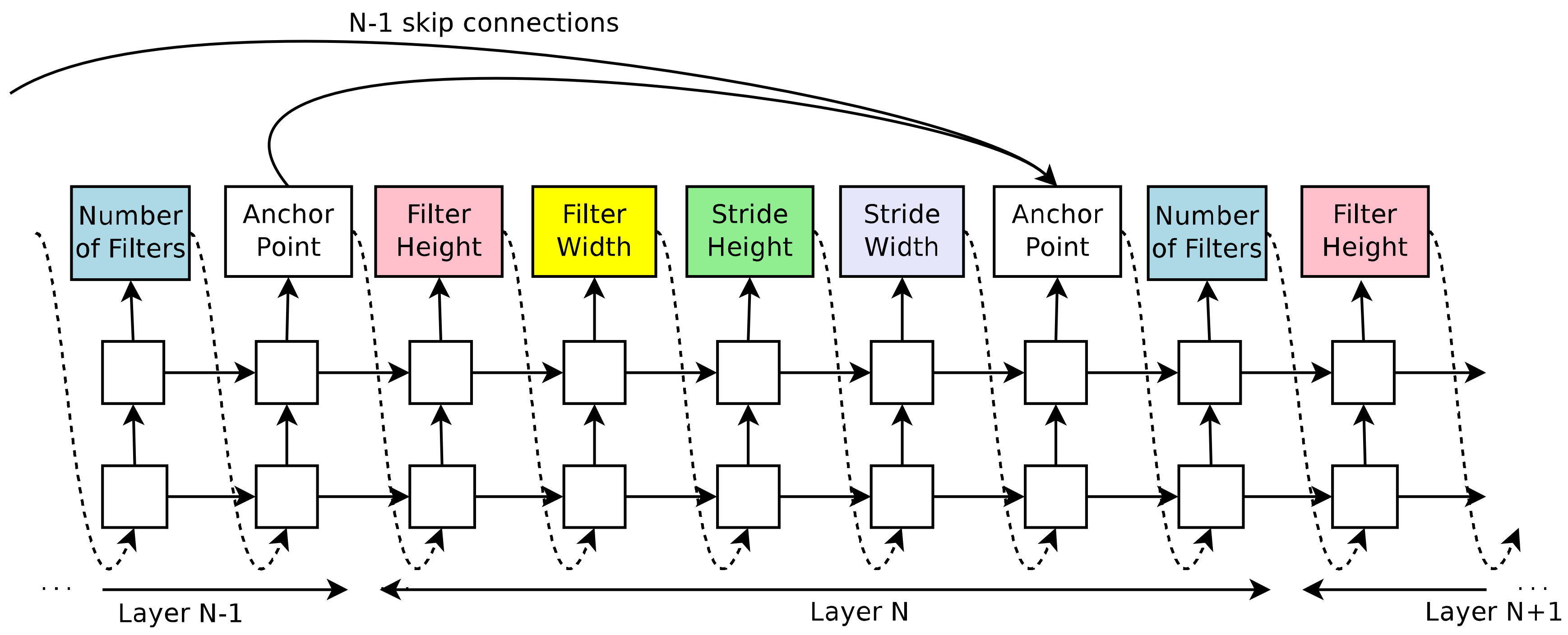
# NAS

## Google Brain 2016

# NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

**Barret Zoph,* Quoc V. Le**
Google Brain



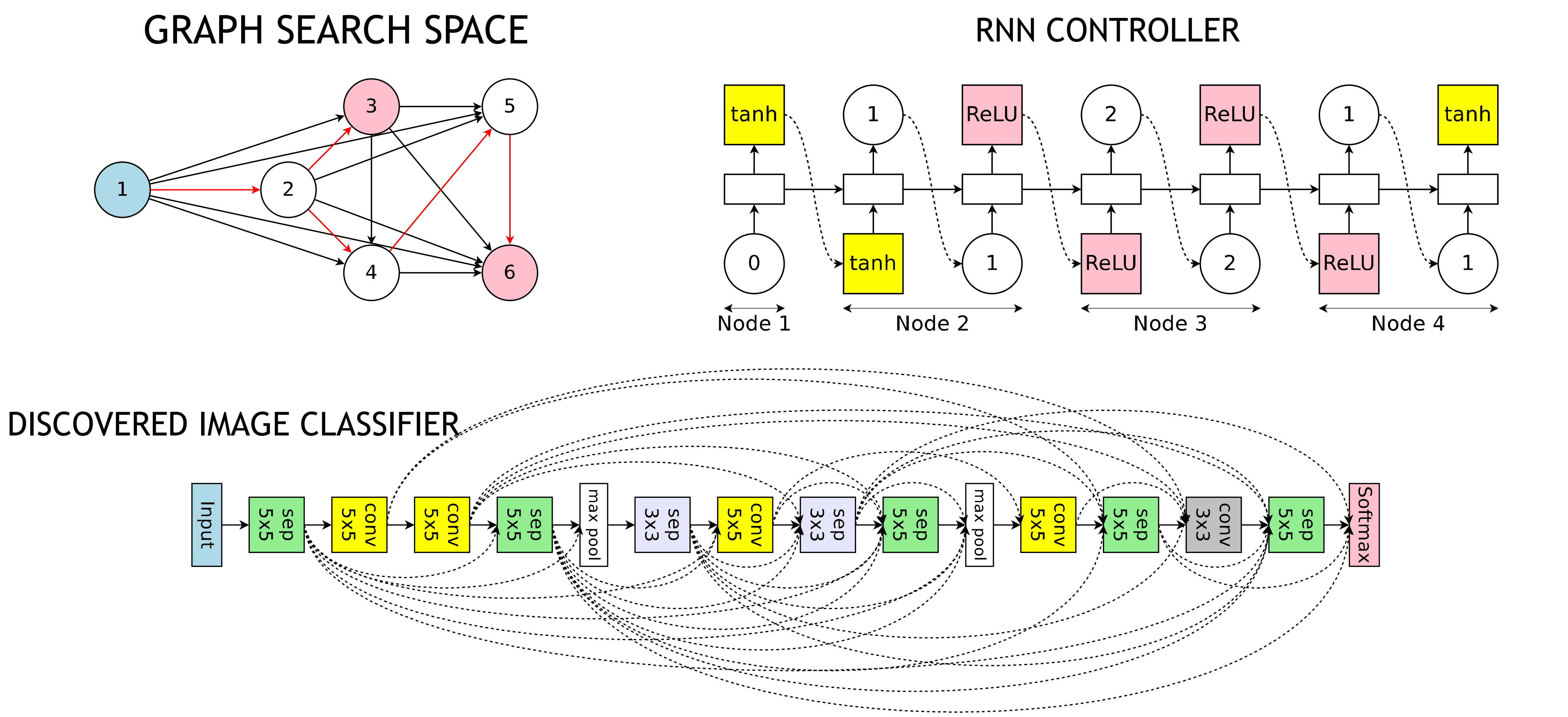



# EFFICIENT NAS

## 2018 Google Brain, CMU, Stanford

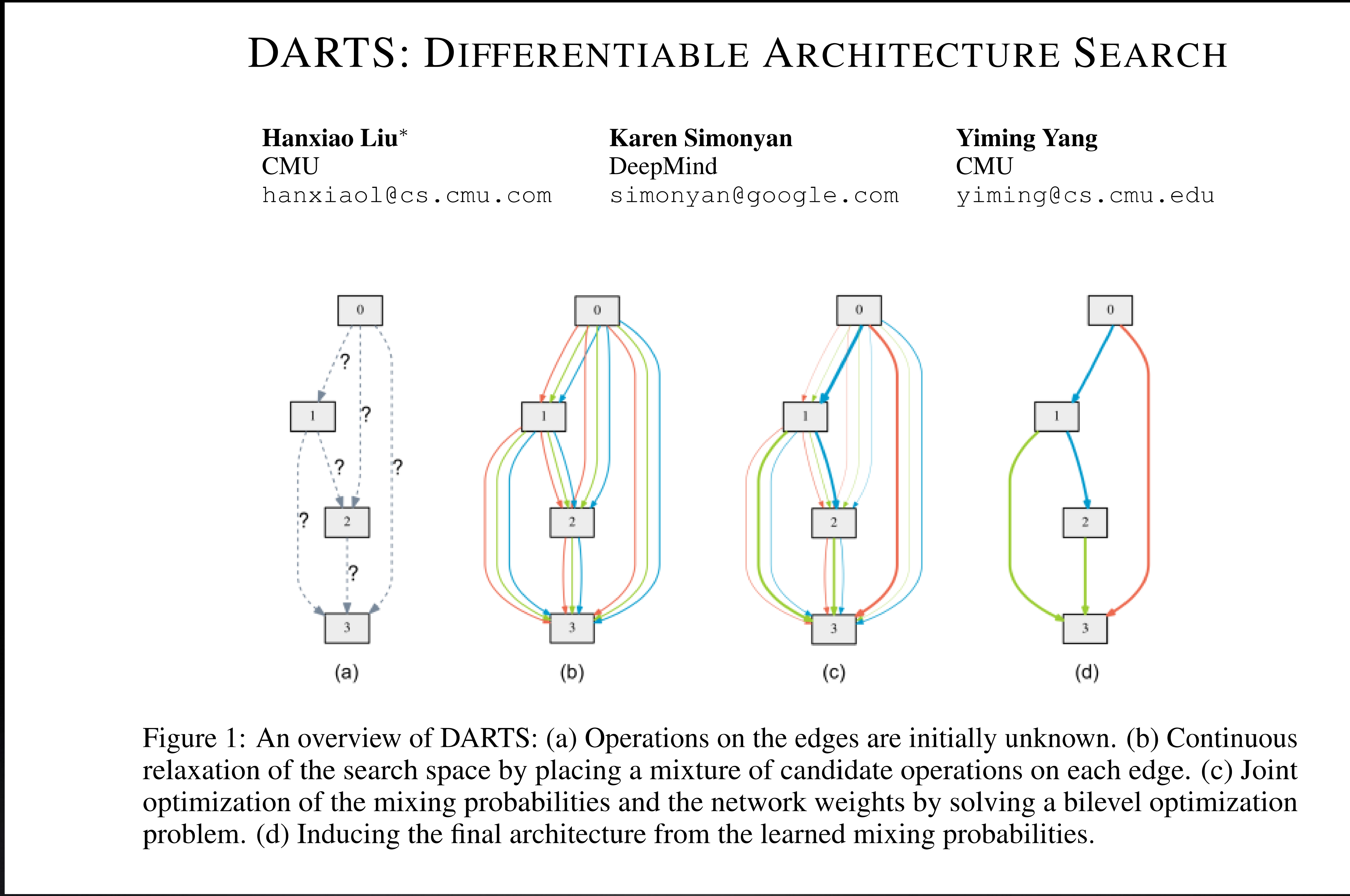
### Efficient Neural Architecture Search via Parameter Sharing

Hieu Pham [*12] Melody Y. Guan [*3] Barret Zoph [1] Quoc V. Le [1] Jeff Dean [1]



# DARTS

## 2019 Google DeepMind, CMU



# SUMMARY

- Eventually AutoML and NAS may make model selection obsolete
- Many models in the model zoo
- Model architectures are task specific
- Encode what you know, learn the rest
- Use pretrained models when you can
- Use transfer learning when you can
- AI changes very quickly. Use Arxiv sanity to keep up.

dhall@nvidia.com