Lecture 22:
Course Recap
Open Problems in Computer Vision

Assignment 6: Generative Models

Generative Adversarial Networks
Variational Autoencoders

Due on Wednesday 12/9, 11:59pm EST

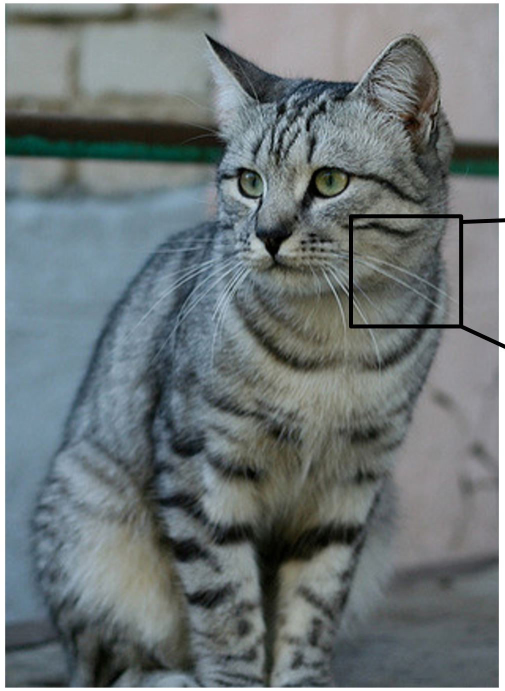
This Course:
Deep Learning for Computer Vision

Deep Learning for Computer Vision

Building artificial systems that process, perceive, and reason about visual data

Problem: Semantic Gap

What you see



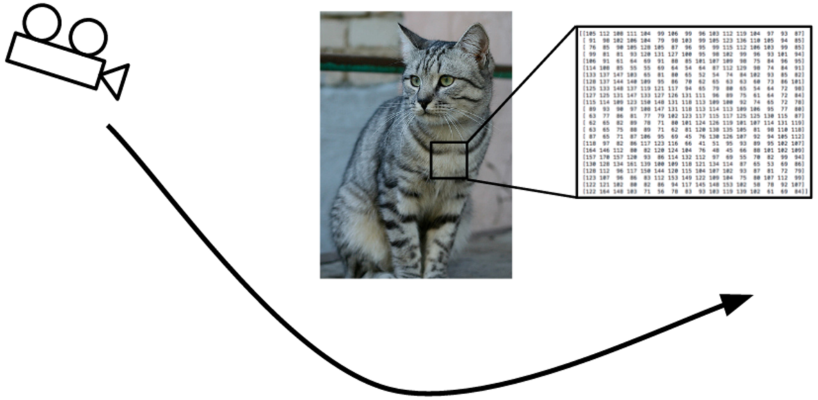
This image by Nikita is licensed under CC-BY 2.0

What computer sees

```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

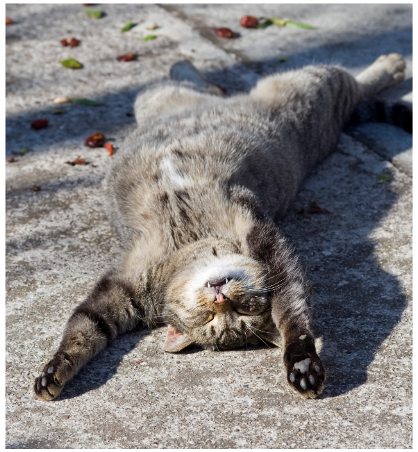
Problem: Visual Data is Complex!

Viewpoint



Illumination

Deformation



Occlusion

Clutter

Intraclass Variation

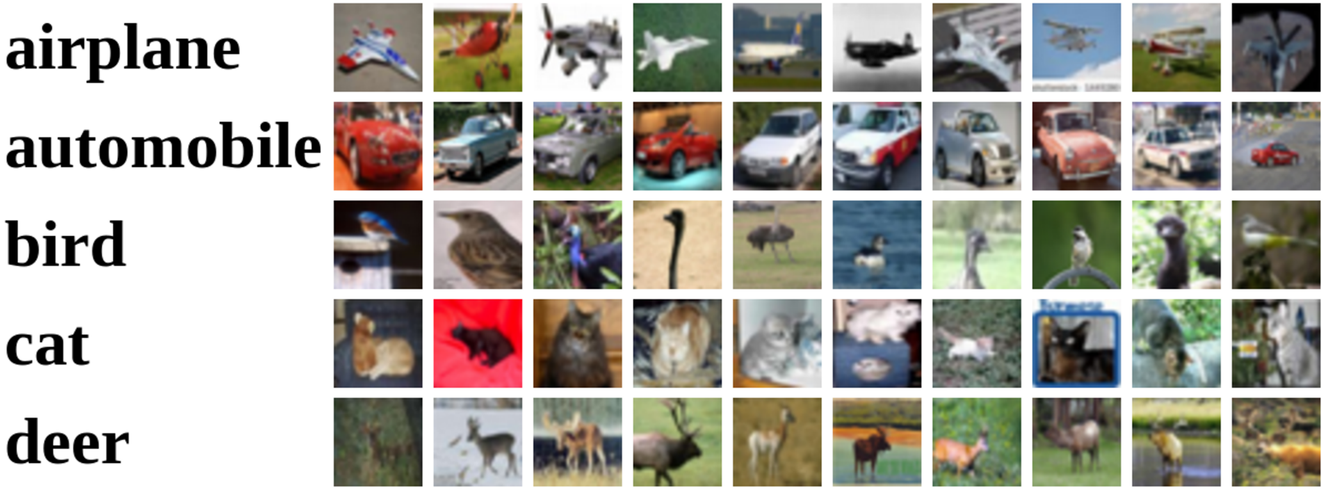
Machine Learning: Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

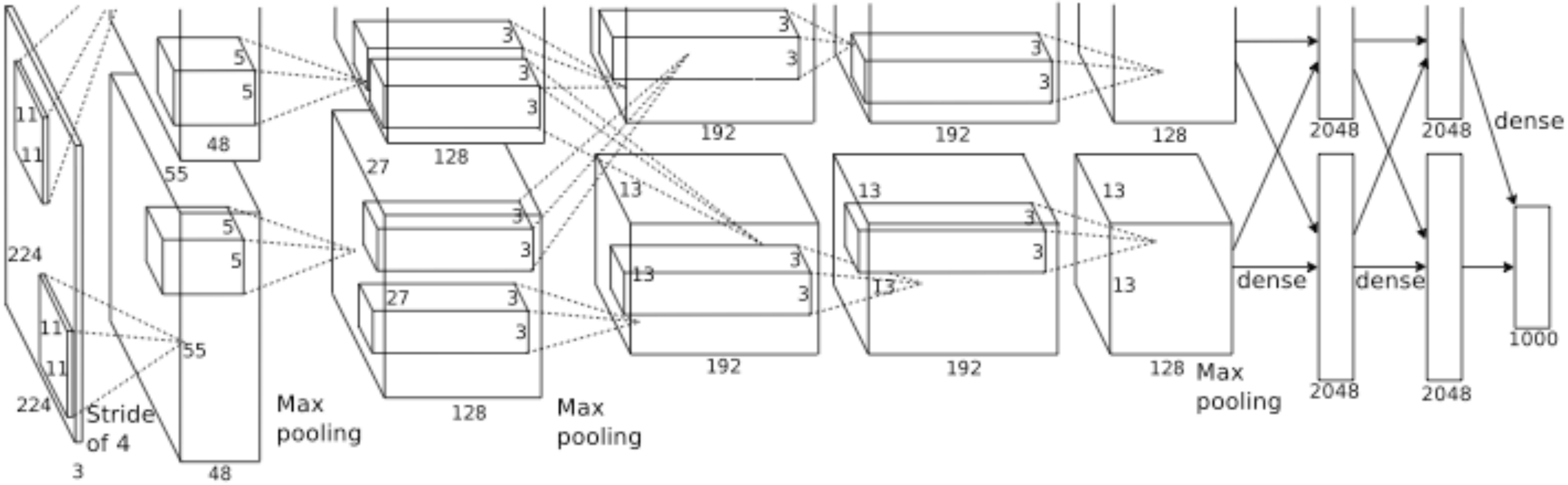
```python
def train(images, labels):
  # Machine learning!
  return model
```

```python
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

Example training set



Model: Deep Convolutional Networks



Krizhevsky, Sutskever, and Hinton, NeurIPS 2012

IMAGENET Large Scale Visual Recognition Challenge

The Image Classification Challenge:

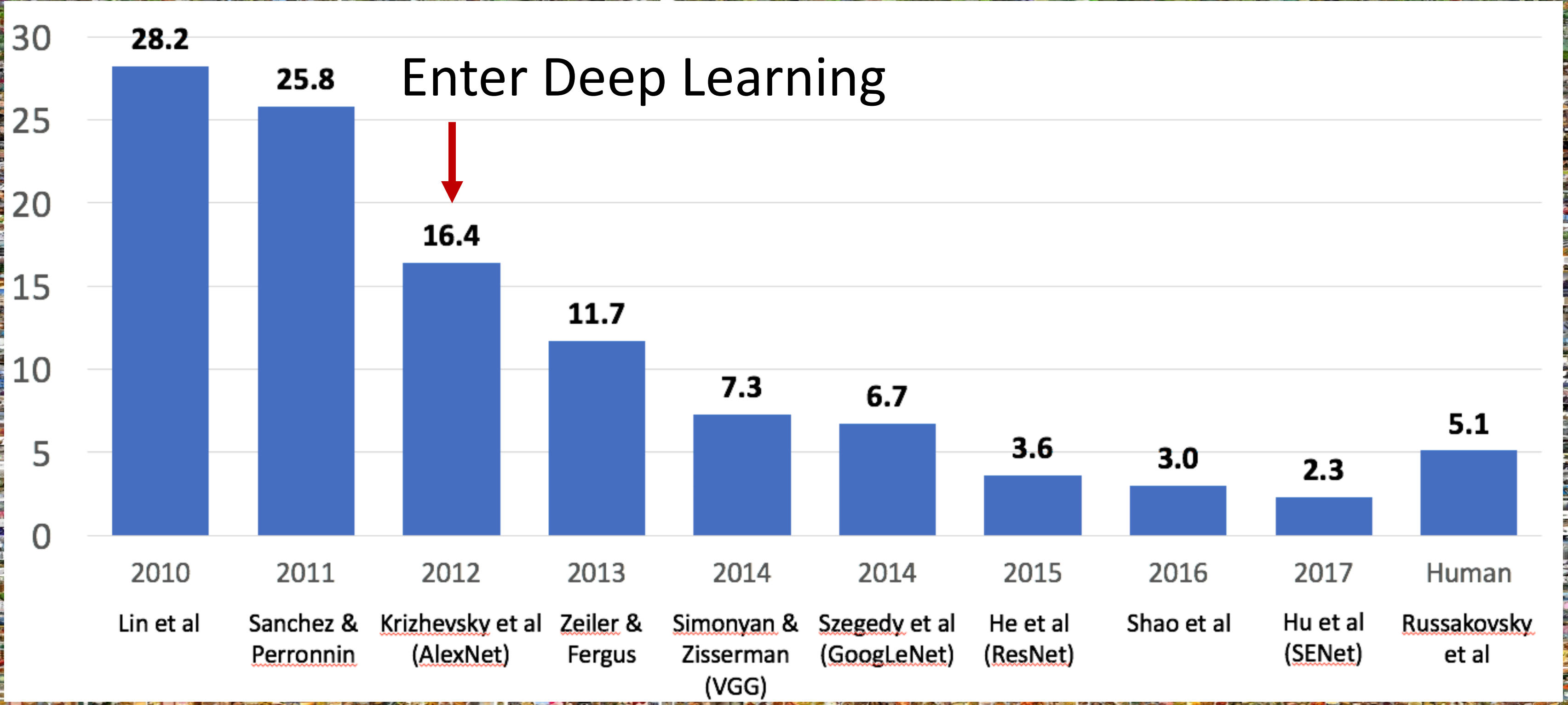
1,000 object classes

1,431,167 images

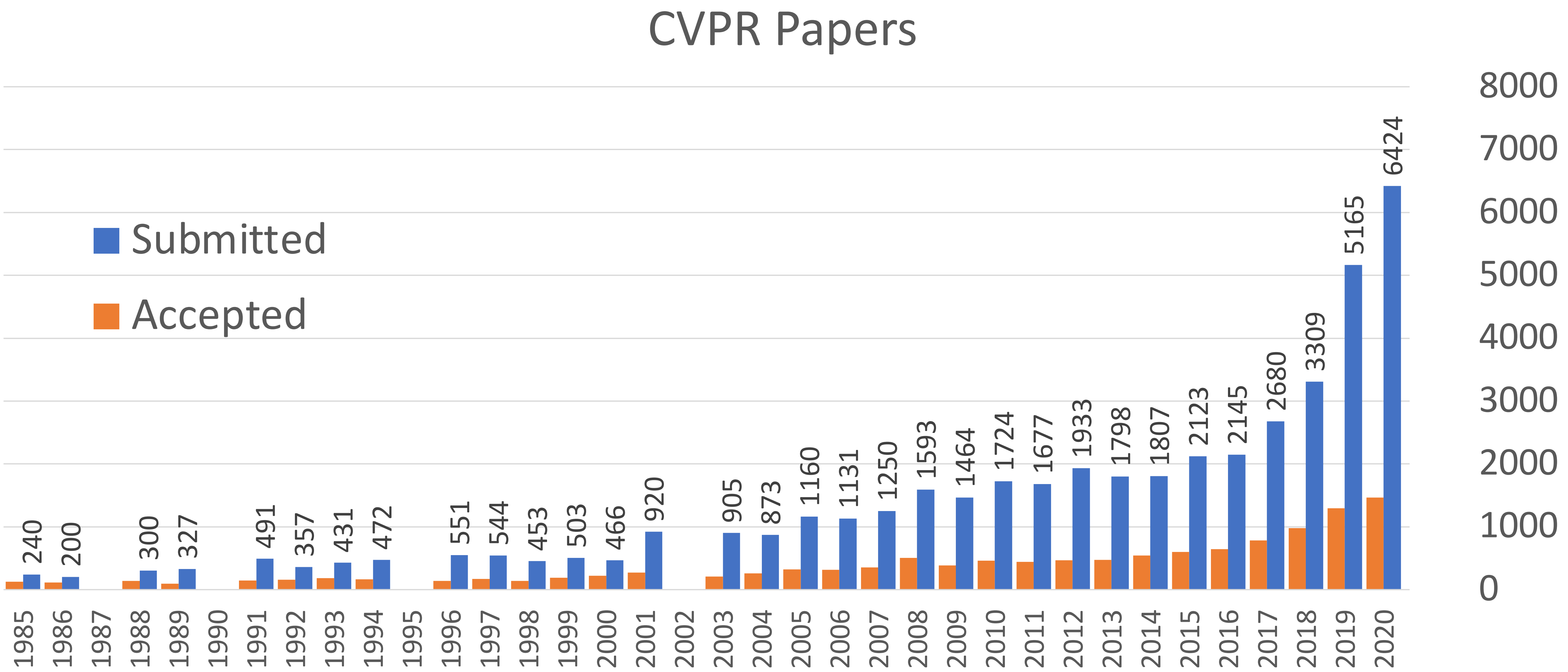
Output:
Scale
T-shirt
Steel drum
Drumstick
Mud turtle

Deng et al, 2009
Russakovsky et al. IJCV 2015

IMAGENET Large Scale Visual Recognition Challenge

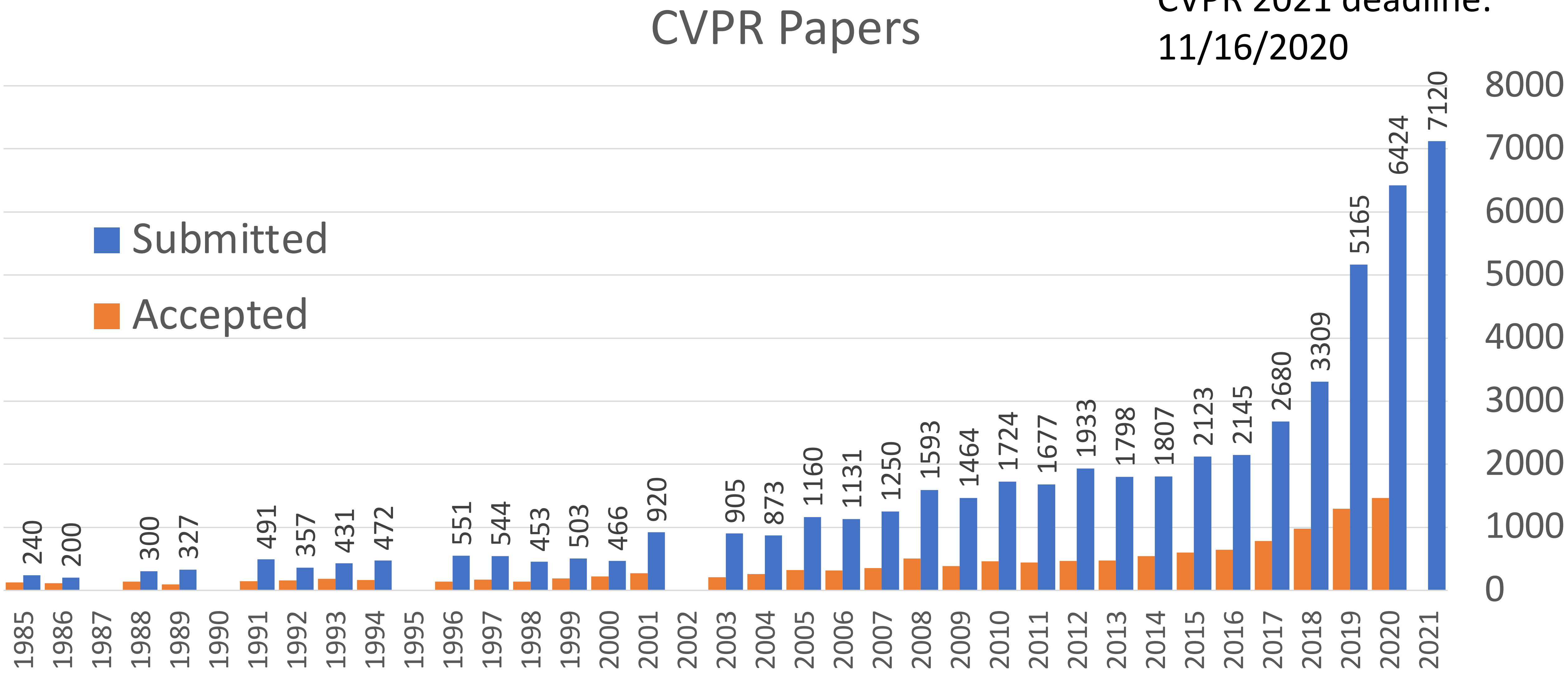


2012 to Present: Deep Learning Explosion

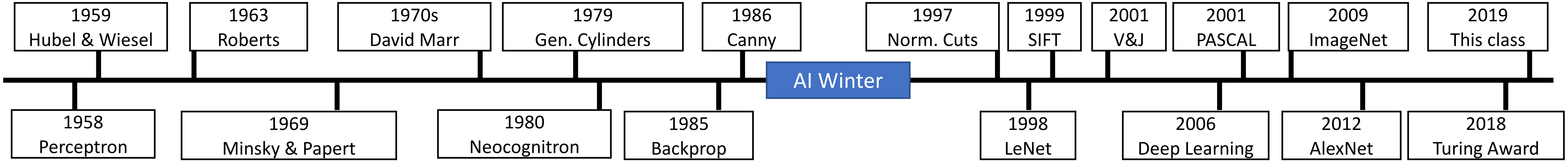


2012 to Present: Deep Learning Explosion

CVPR 2021 deadline:
11/16/2020



Deep Learning wasn’t invented overnight!



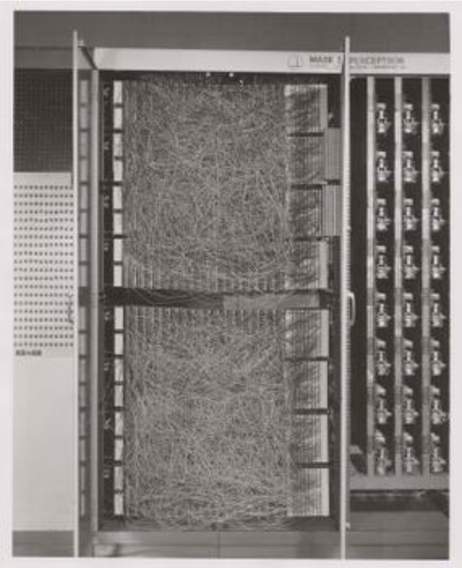
Deep Learning wasn't invented overnight!

1959 Hubel & Wiesel | 1963 Roberts | 1970s David Marr | 1979 Gen. Cylinders | 1986 Canny | 1997 Norm. Cuts | 1999 SIFT | 2001 V&J | 2001 PASCAL | 2009 ImageNet | 2019 This class

AI Winter

1958 Perceptron | 1969 Minsky & Papert | 1980 Neocognitron | 1985 Backprop | 1998 LeNet | 2006 Deep Learning | 2012 AlexNet | 2018 Turing Award

Perceptron

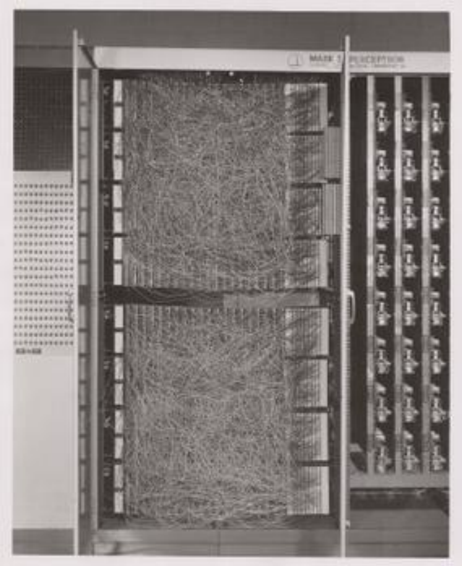


Frank Rosenblatt, ~1957

Deep Learning wasn't invented overnight!

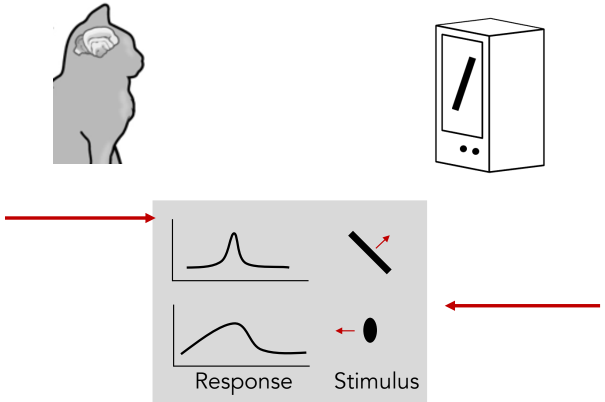
- 1958 Perceptron
- 1959 Hubel & Wiesel
- 1963 Roberts
- 1969 Minsky & Papert
- 1970s David Marr
- 1979 Gen. Cylinders
- 1980 Neocognitron
- 1985 Backprop
- 1986 Canny
- AI Winter
- 1997 Norm. Cuts
- 1998 LeNet
- 1999 SIFT
- 2001 V&J
- 2001 PASCAL
- 2006 Deep Learning
- 2009 ImageNet
- 2012 AlexNet
- 2018 Turing Award
- 2019 This class

Perceptron



Frank Rosenblatt, ~1957

Simple and Complex cells

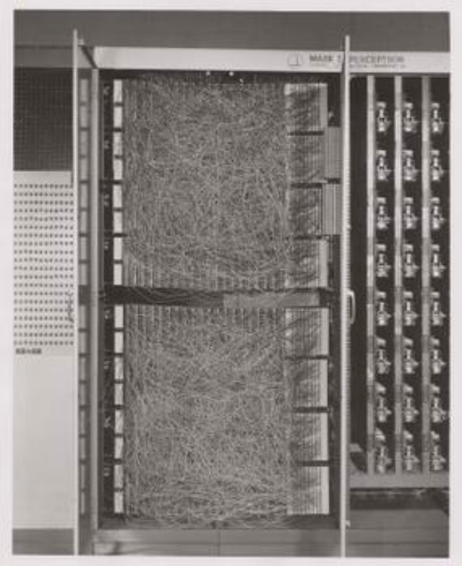


Hubel and Wiesel, 1959

Deep Learning wasn't invented overnight!

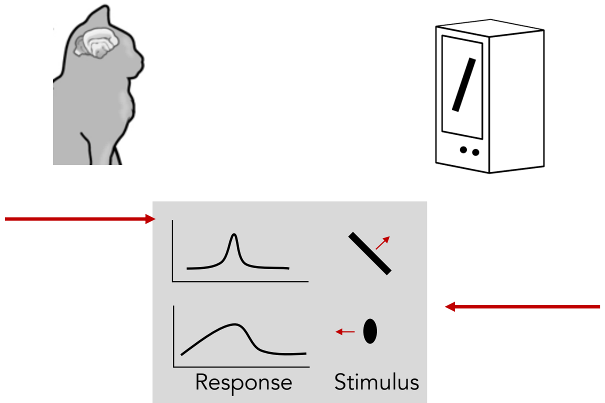
1958 Perceptron	1959 Hubel & Wiesel	1963 Roberts	1969 Minsky & Papert	1970s David Marr	1979 Gen. Cylinders	1980 Neocognitron	1985 Backprop	1986 Canny	AI Winter	1997 Norm. Cuts	1998 LeNet	1999 SIFT	2001 V&J	2001 PASCAL	2006 Deep Learning	2009 ImageNet	2012 AlexNet	2018 Turing Award	2019 This class

Perceptron



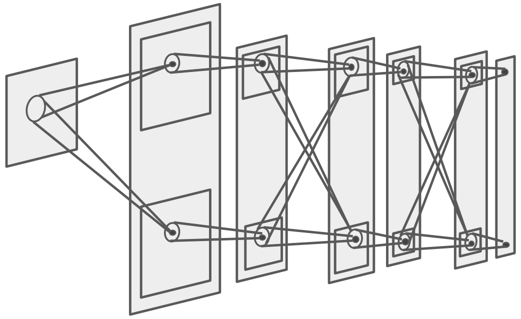
Frank Rosenblatt, ~1957

Simple and Complex cells



Hubel and Wiesel, 1959

Neocognitron



Fukushima, 1980

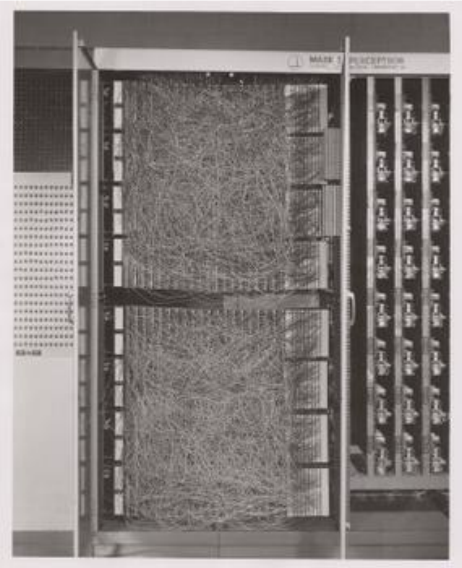
Deep Learning wasn't invented overnight!

1959 Hubel & Wiesel
1963 Roberts
1970s David Marr
1979 Gen. Cylinders
1986 Canny
1997 Norm. Cuts
1999 SIFT
2001 V&J
2001 PASCAL
2009 ImageNet
2019 This class

AI Winter

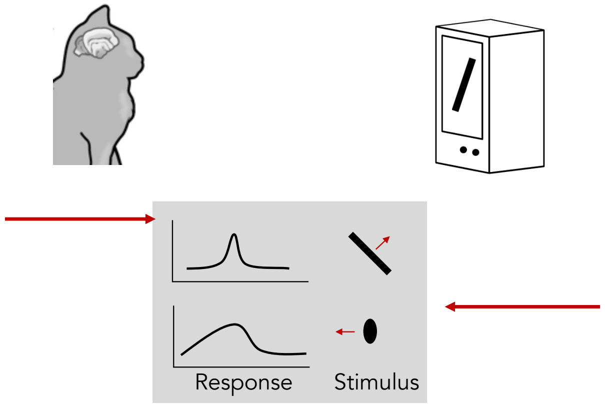
1958 Perceptron
1969 Minsky & Papert
1980 Neocognitron
1985 Backprop
1998 LeNet
2006 Deep Learning
2012 AlexNet
2018 Turing Award

Perceptron



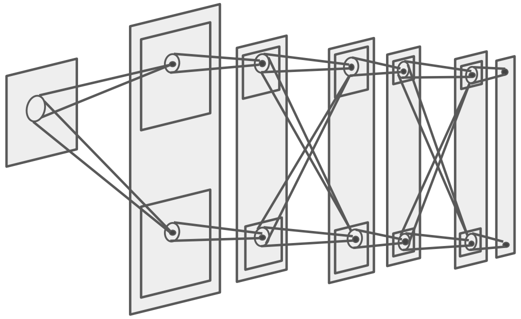
Frank Rosenblatt, ~1957

Simple and Complex cells



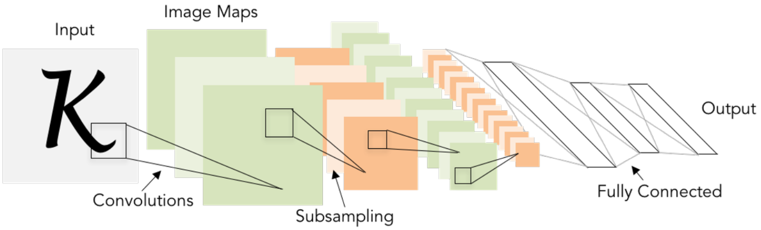
Hubel and Wiesel, 1959

Neocognitron



Fukushima, 1980

Convolutional Networks



LeCun et al, 1998

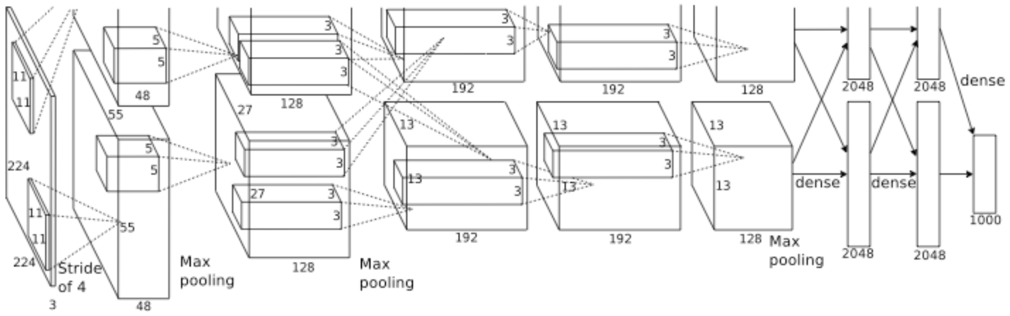
Deep Learning wasn't invented overnight!

1959 Hubel & Wiesel | 1963 Roberts | 1970s David Marr | 1979 Gen. Cylinders | 1986 Canny | 1997 Norm. Cuts | 1999 SIFT | 2001 V&J | 2001 PASCAL | 2009 ImageNet | 2019 This class

AI Winter

1958 Perceptron | 1969 Minsky & Papert | 1980 Neocognitron | 1985 Backprop | 1998 LeNet | 2006 Deep Learning | 2012 AlexNet | 2018 Turing Award

AlexNet



Krizhevsky, Sutskever, and Hinton, 2012

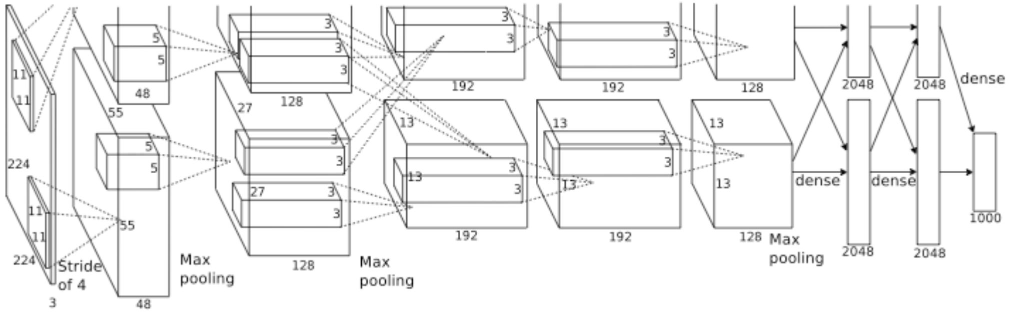
Deep Learning wasn't invented overnight!

1959 Hubel & Wiesel
1963 Roberts
1970s David Marr
1979 Gen. Cylinders
1986 Canny
1997 Norm. Cuts
1999 SIFT
2001 V&J
2001 PASCAL
2009 ImageNet
2019 This class

AI Winter

1958 Perceptron
1969 Minsky & Papert
1980 Neocognitron
1985 Backprop
1998 LeNet
2006 Deep Learning
2012 AlexNet
2018 Turing Award

AlexNet



Krizhevsky, Sutskever, and Hinton, 2012

2018 Turing Award

Yoshua Bengio

Geoffrey Hinton

Yann LeCun

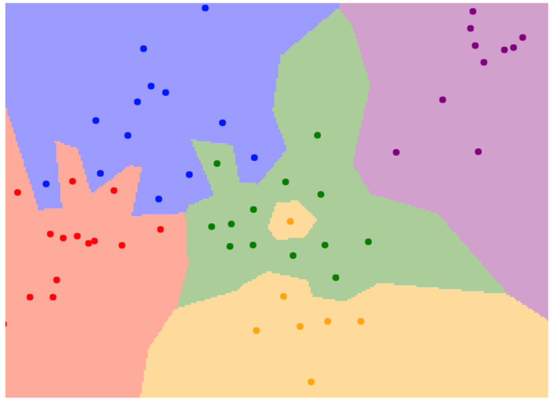
Deep Learning wasn't invented overnight!

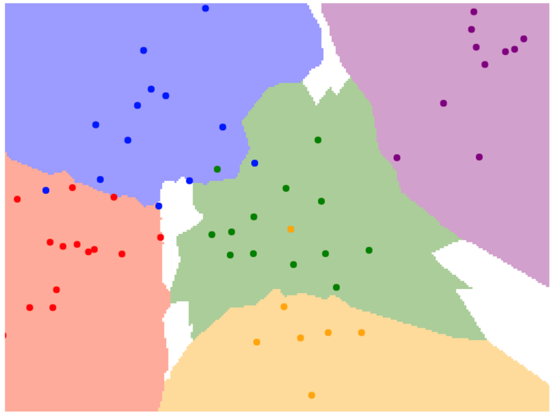
1958 Perceptron
1959 Hubel & Wiesel
1963 Roberts
1969 Minsky & Papert
1970s David Marr
1979 Gen. Cylinders
1980 Neocognitron
1985 Backprop
1986 Canny
AI Winter
1997 Norm. Cuts
1998 LeNet
1999 SIFT
2001 V&J
2001 PASCAL
2006 Deep Learning
2009 ImageNet
2012 AlexNet
2018 Turing Award
2020 This class

Fall 2020: This class

Simple Classifiers: kNN and Linear Classifiers

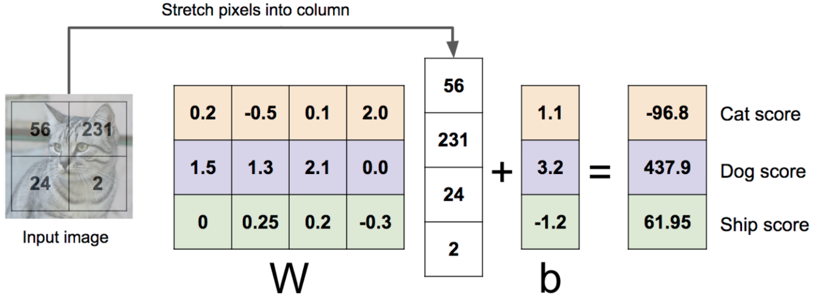
1-NN classifier

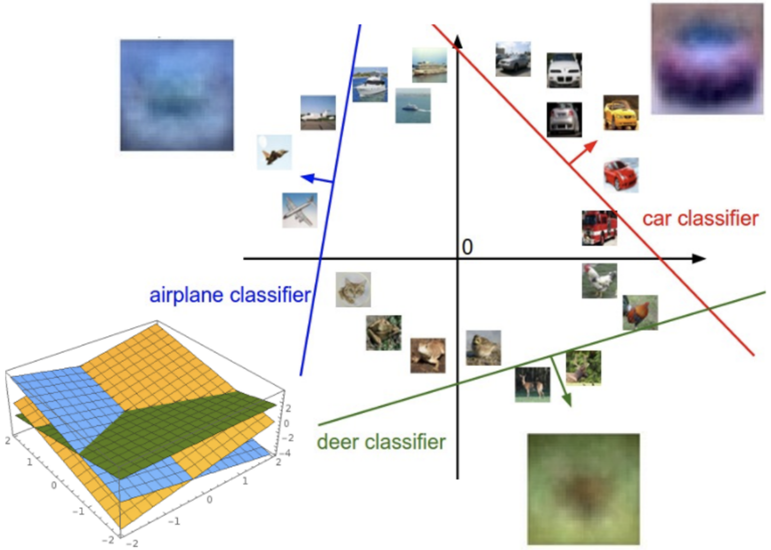




5-NN classifier

Linear Classifiers: y = Wx + b



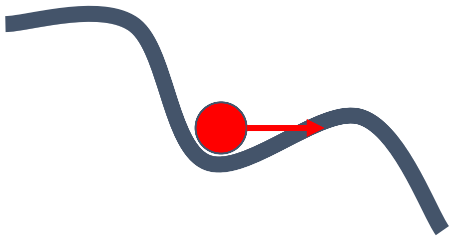


Optimization with Gradient Descent

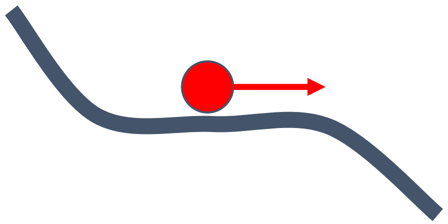
```python
# Vanilla gradient descent
w = initialize_weights()
for t in range(num_steps):
  dw = compute_gradient(loss_fn, data, w)
  w -= learning_rate * dw
```

Problems with Gradient Descent

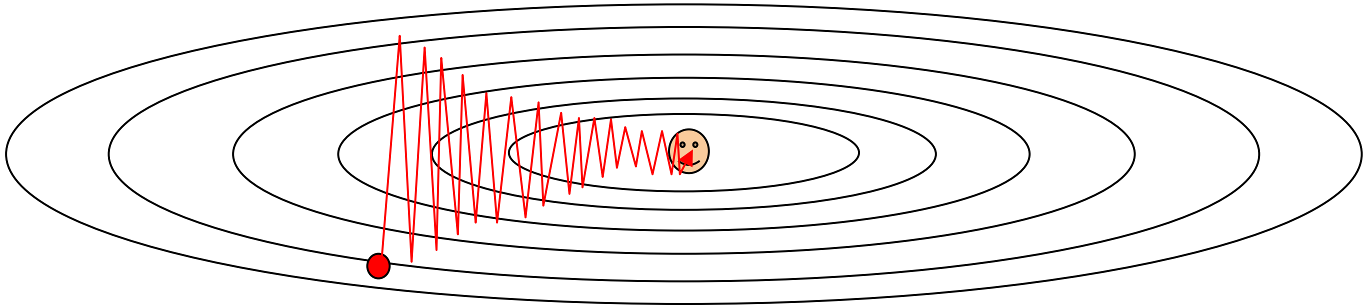
Local Minima



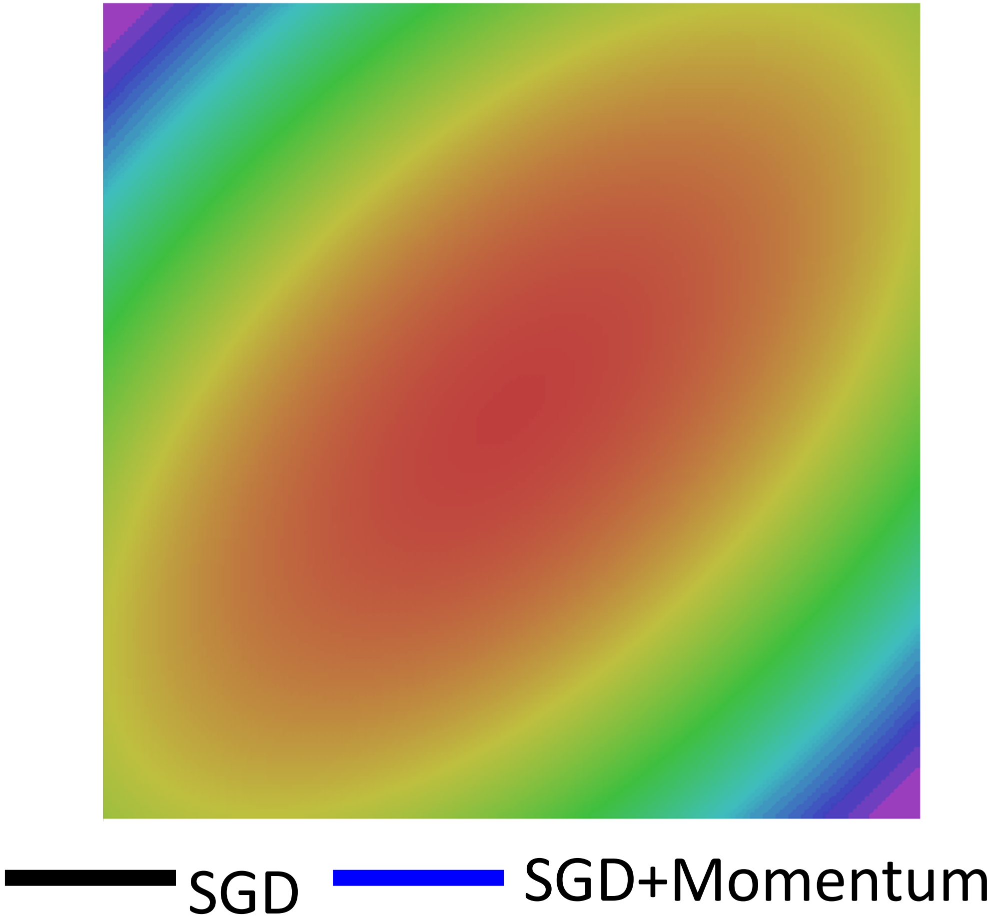
Saddle points



Poor Conditioning



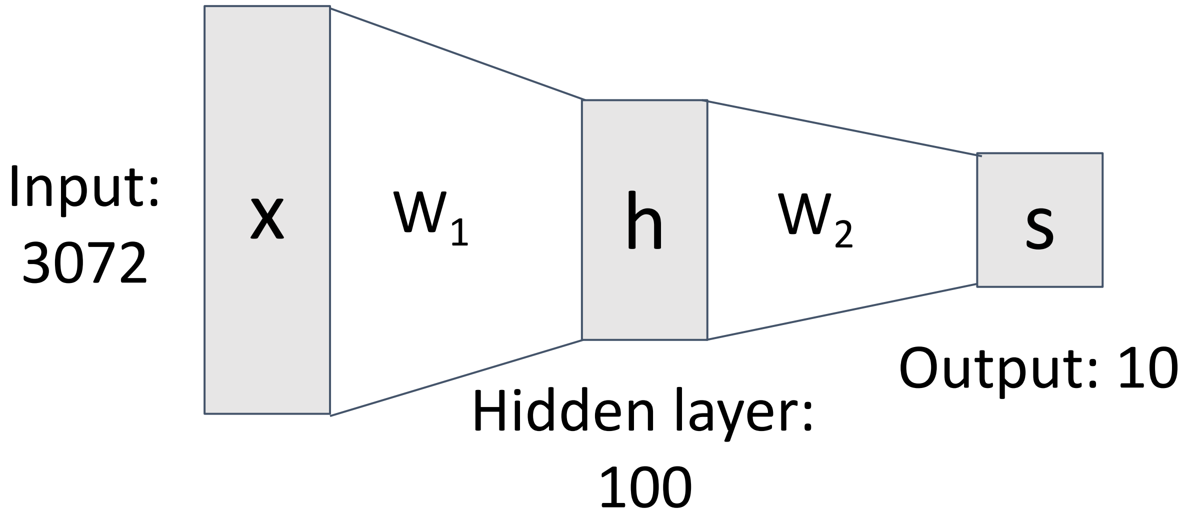
Gradient Noise



Gradient Descent Improvements

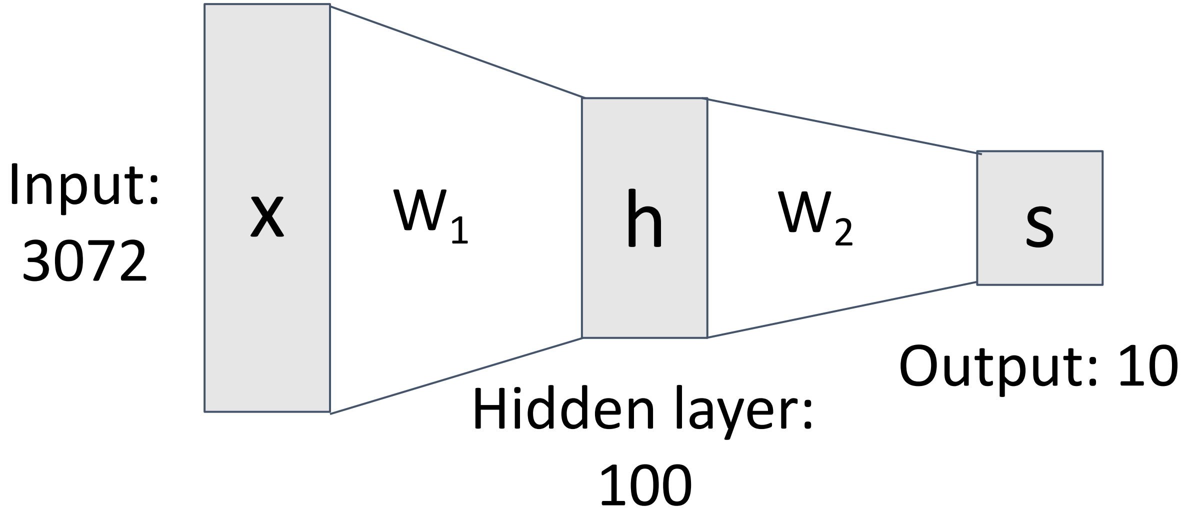
Algorithm	Tracks first moments (Momentum)	Tracks second moments (Adaptive learning rates)	Leaky second moments	Bias correction for moment estimates
SGD	✗	✗	✗	✗
SGD+Momentum	✓	✗	✗	✗
Nesterov	✓	✗	✗	✗
AdaGrad	✗	✓	✗	✗
RMSProp	✗	✓	✓	✗
Adam	✓	✓	✓	✓

More Complex Models: Neural Networks



$$f = W_2 \max(0, W_1 x)$$

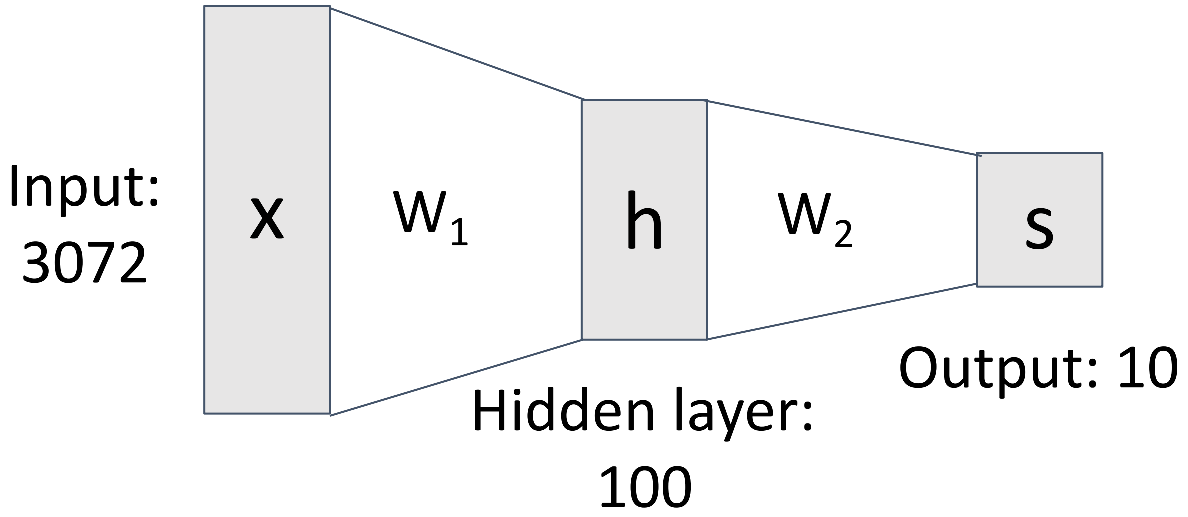
More Complex Models: Neural Networks



$$f = W_2 \max(0, W_1 x)$$

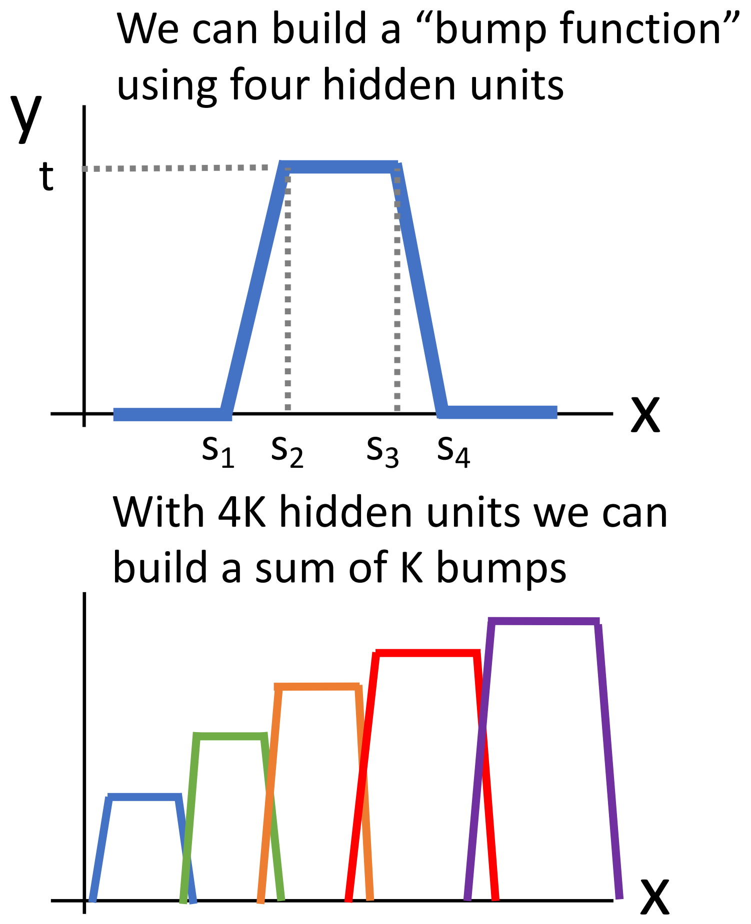
Learns bank of templates

More Complex Models: Neural Networks



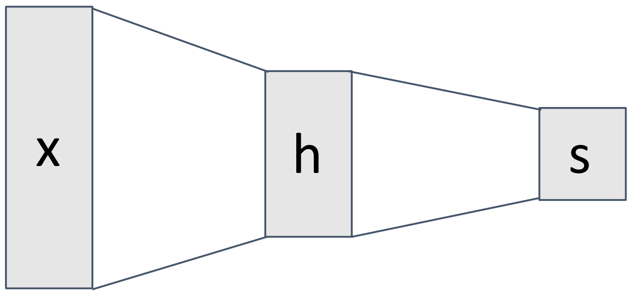
$$f = W_2 \max(0, W_1 x)$$

Universal Approximation

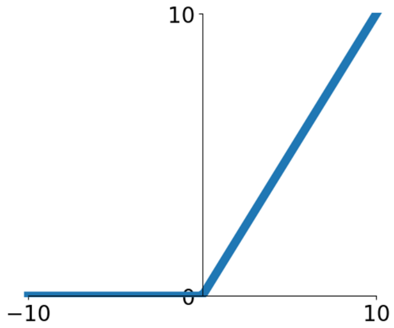


More Complex Models: Convolutional Networks

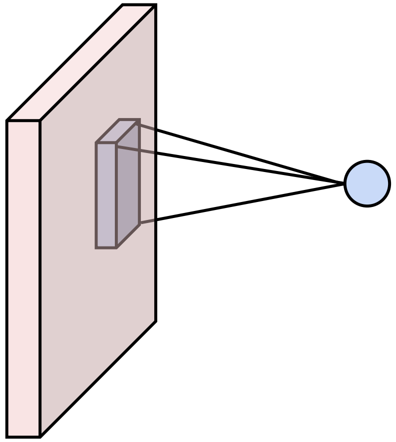
Fully-Connected Layers



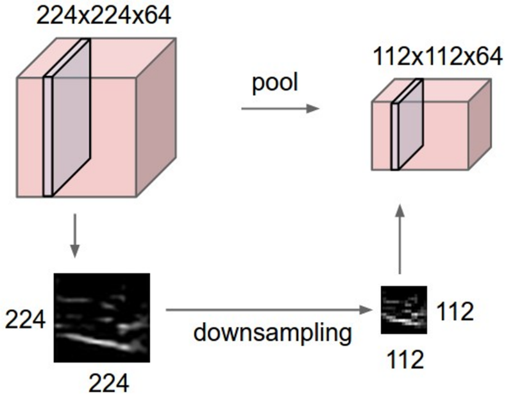
Activation Function



Convolution Layers

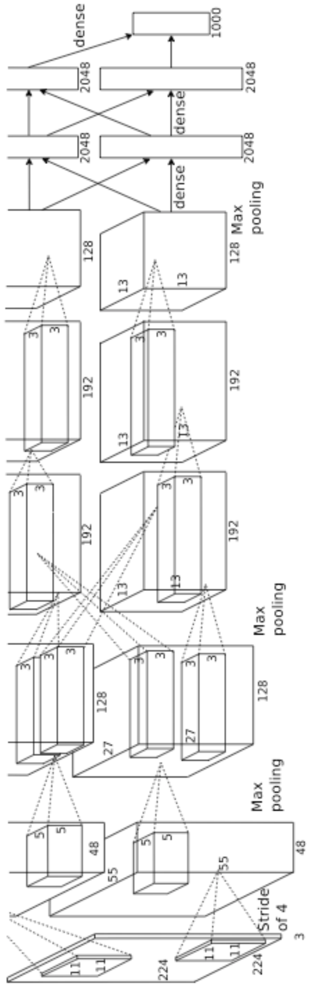


Pooling Layers

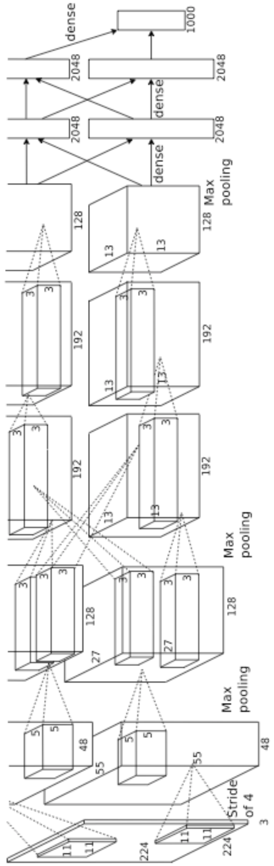


Normalization

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}}$$



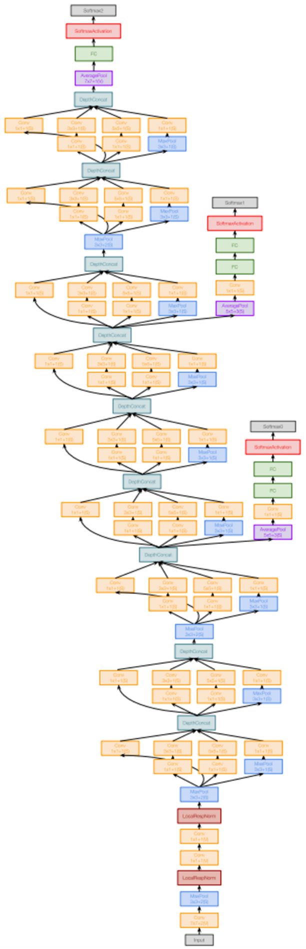
CNN Architectures



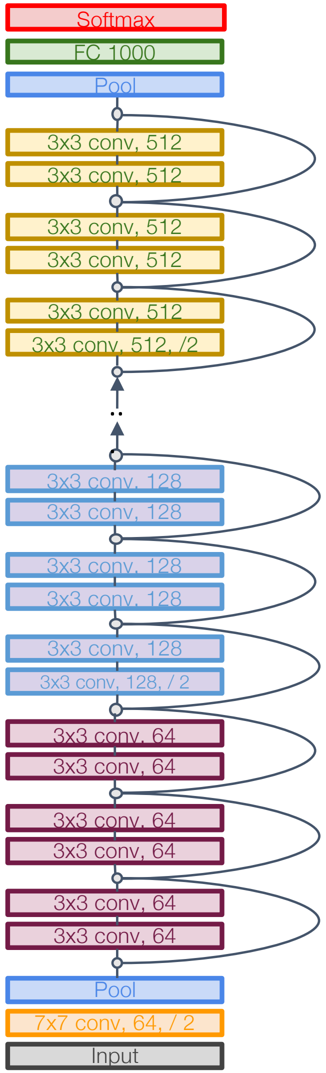
AlexNet

VGG16

VGG19

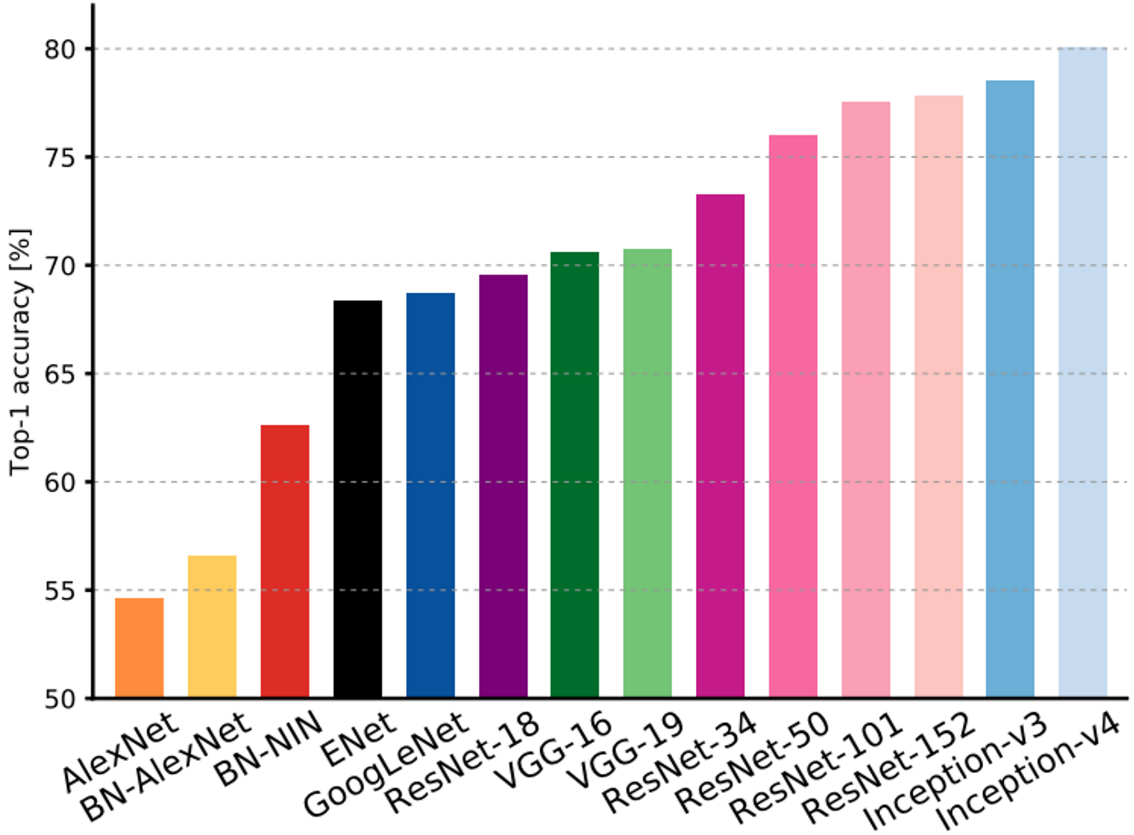


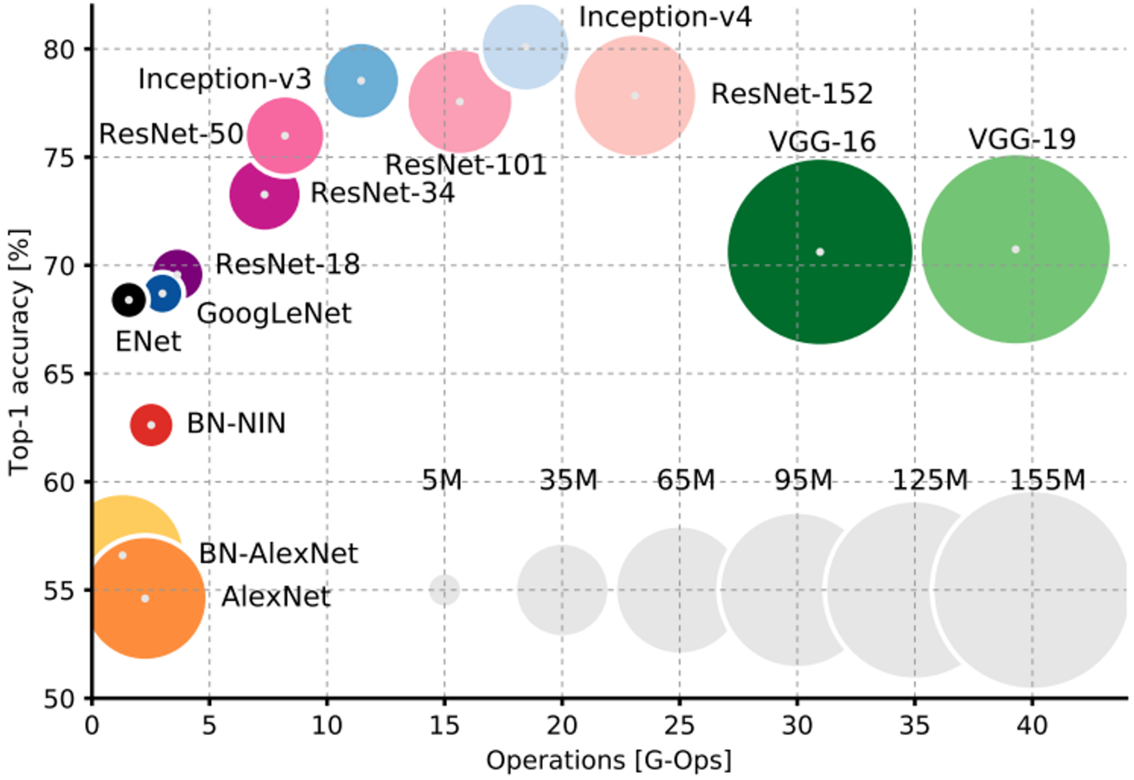
GoogLeNet



ResNet

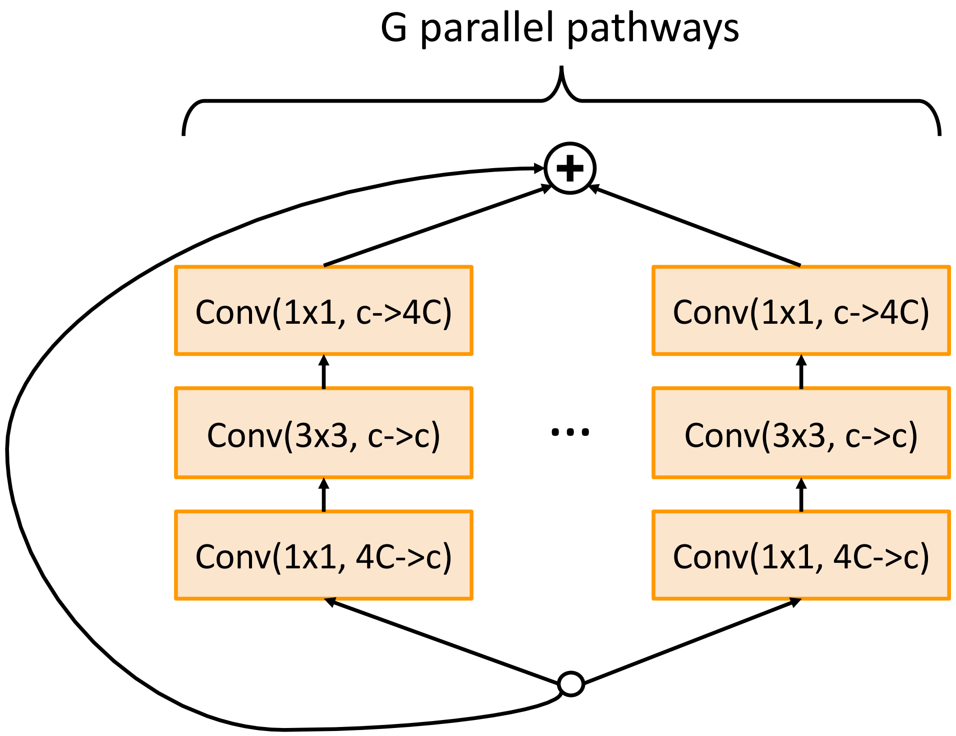
CNN Architectures: Efficiency



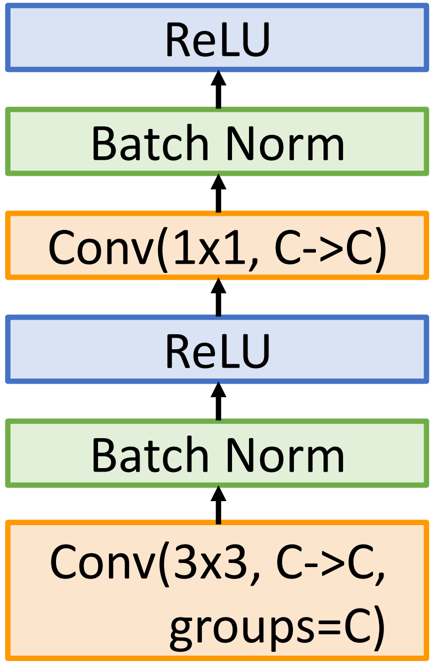


Canziani et al, "An analysis of deep neural network models for practical applications", 2017

CNN Architecture: Efficiency

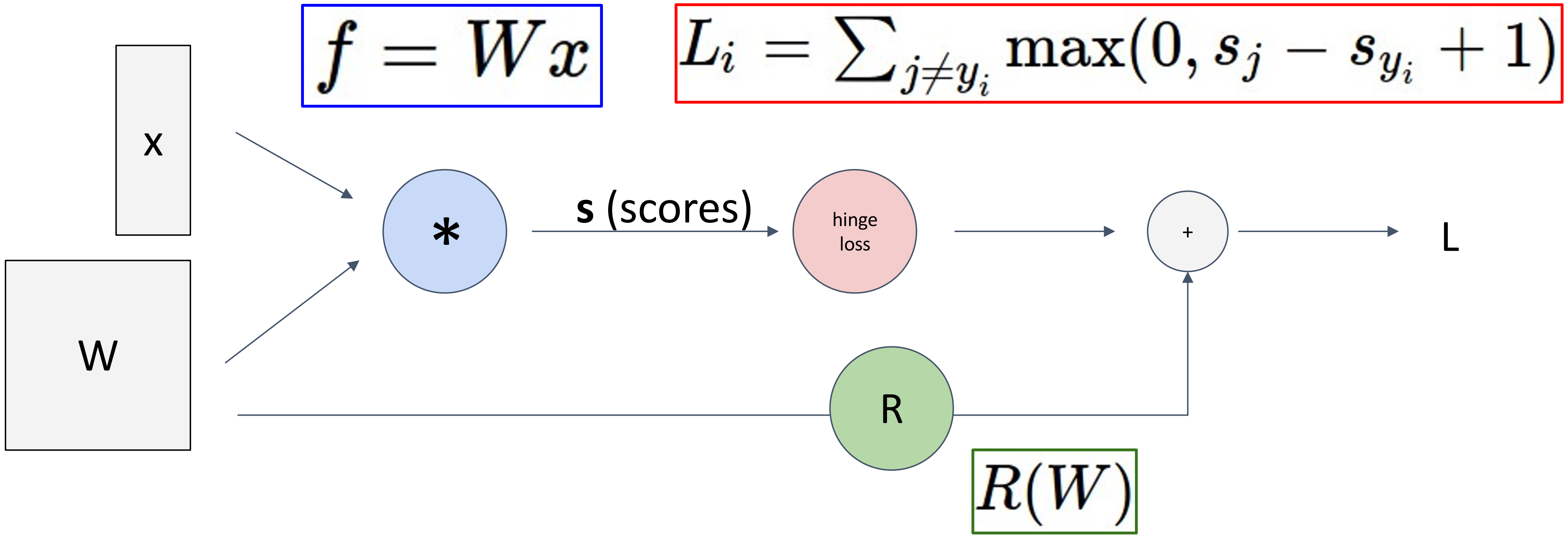


ResNeXt

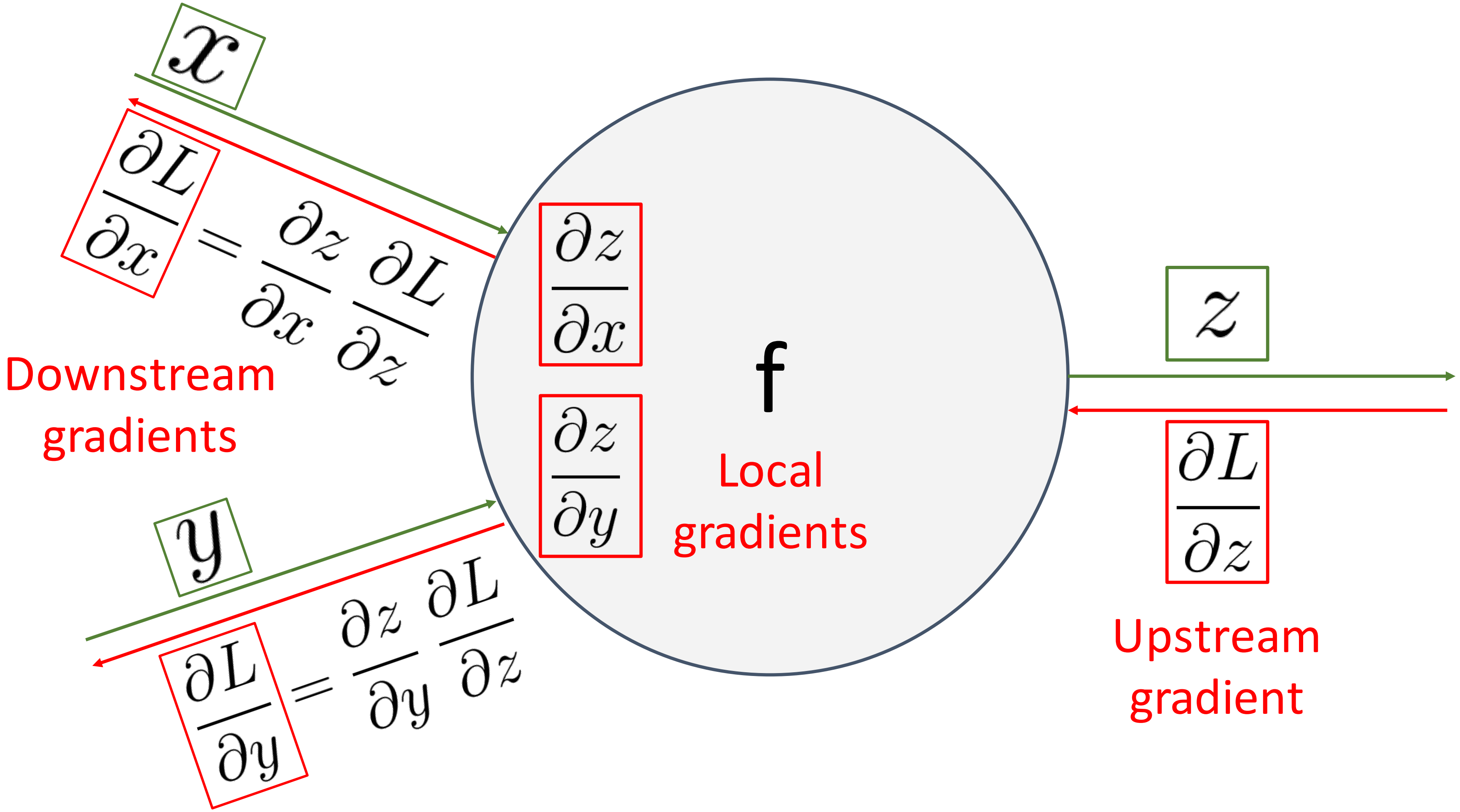


MobileNets

Representing Networks: Computational Graphs



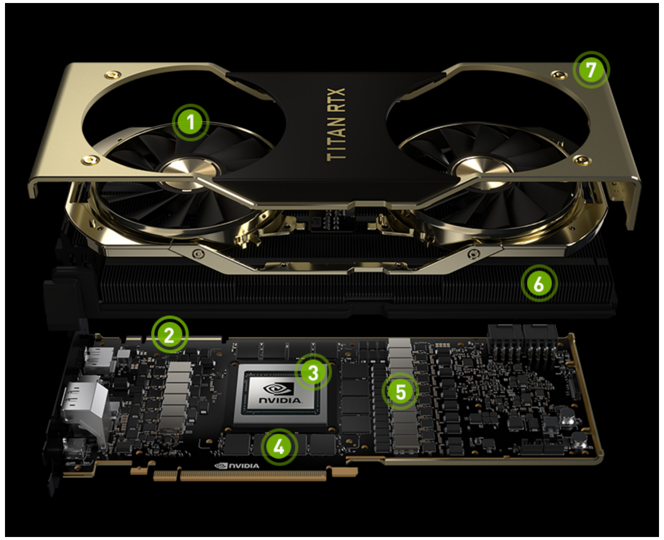
Computing Gradients: Backpropagation



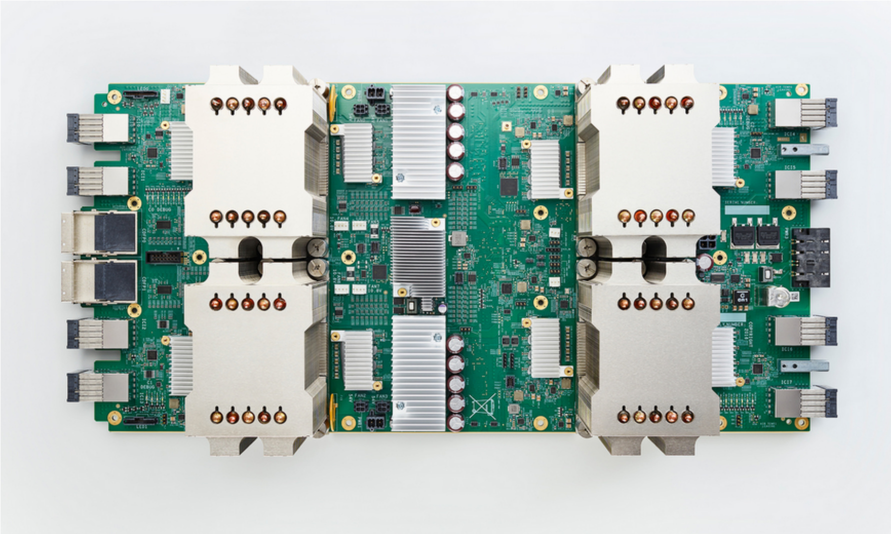
Deep Learning Hardware and Software

CPU

GPU



TPU



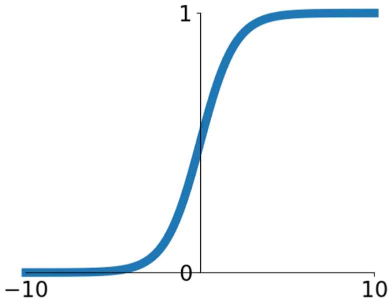
Static Graphs vs **Dynamic Graphs**

PyTorch vs **TensorFlow**

Training Neural Networks: Activation Functions

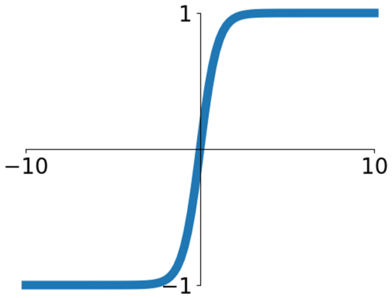
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



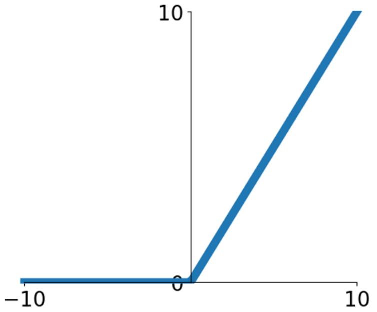
tanh

$\tanh(x)$



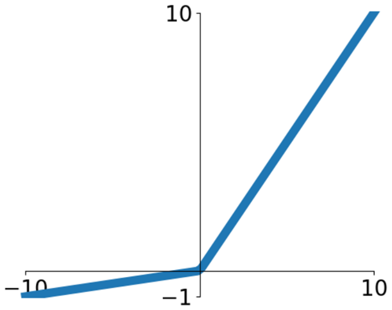
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

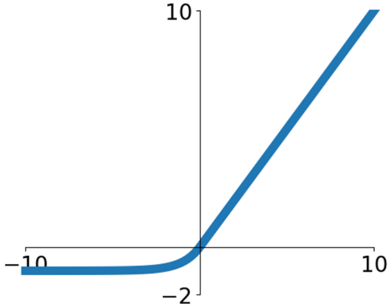


Maxout

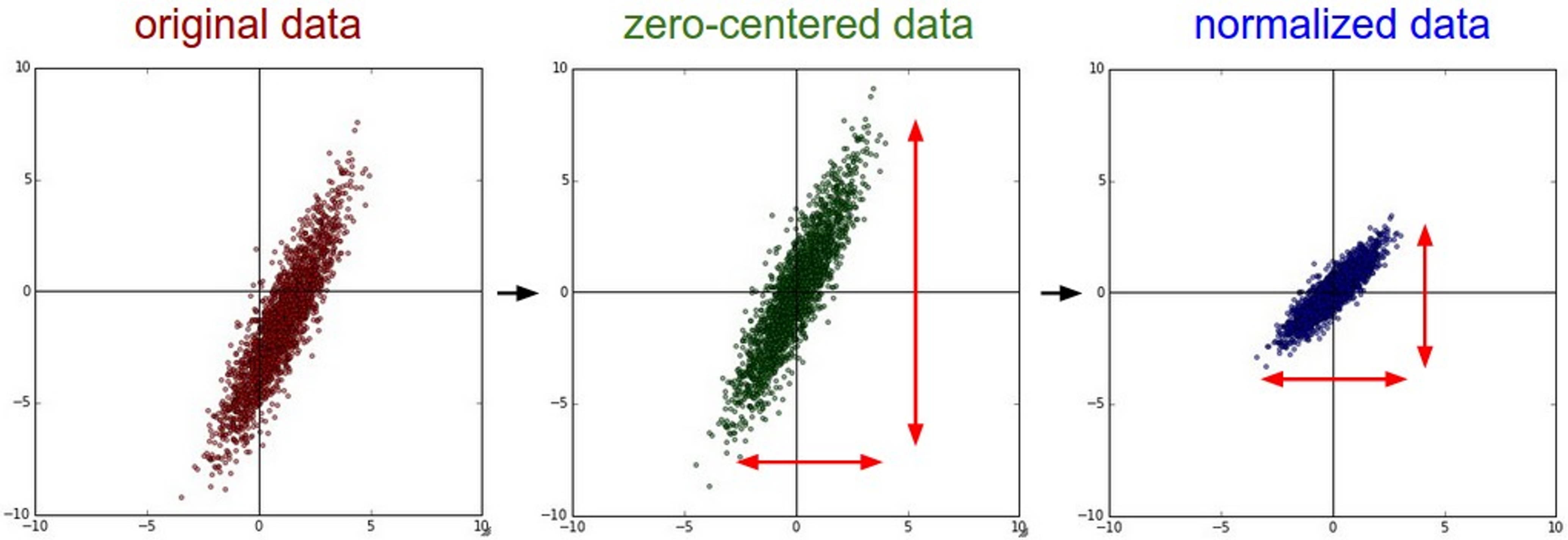
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



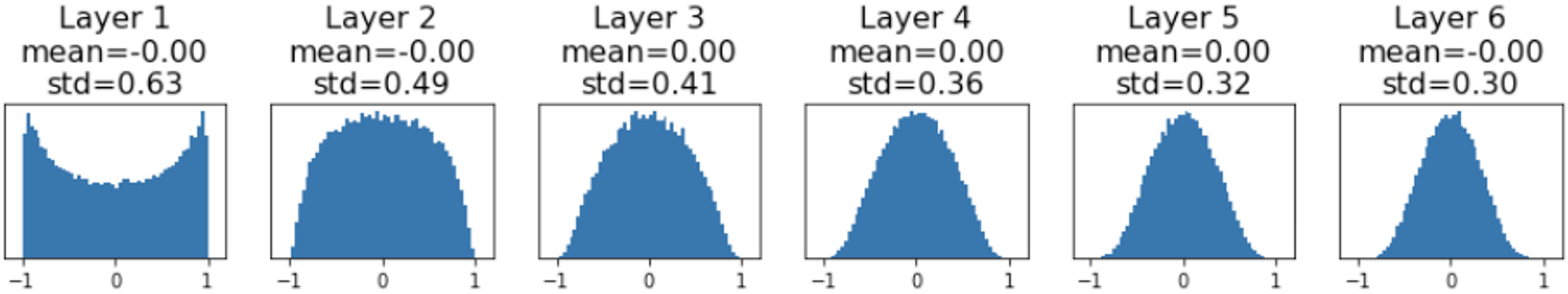
Training Neural Networks: Data Preprocessing



Training Neural Networks: Weight Initialization

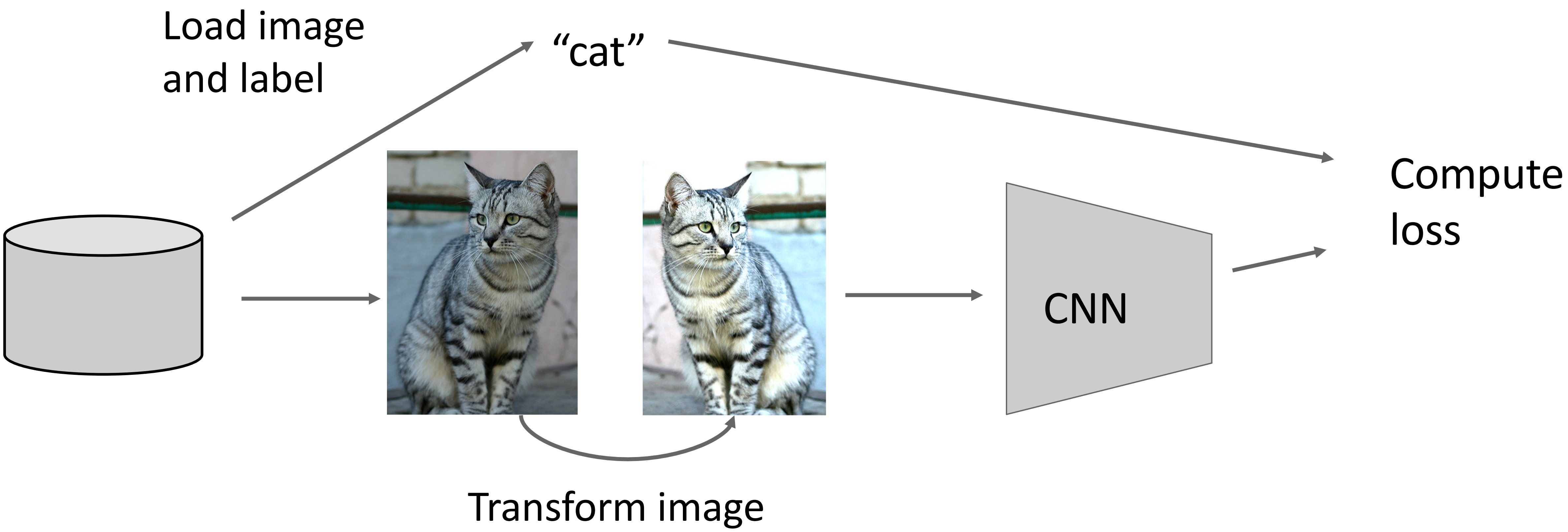
```
dims = [4096] * 7
hs = []
x = np.random.randn(16, dims[0])
for Din, Dout in zip(dims[:-1], dims[1:]):
    W = np.random.randn(Din, Dout) / np.sqrt(Din)
    x = np.tanh(x.dot(W))
    hs.append(x)
```

"Just right": Activations are nicely scaled for all layers!



Glorot and Bengio, "Understanding the difficulty of training deep feedforward neural networks", AISTAT 2010

Training Neural Networks: Data Augmentation



Training Neural Networks: Regularization

Training: Add randomness

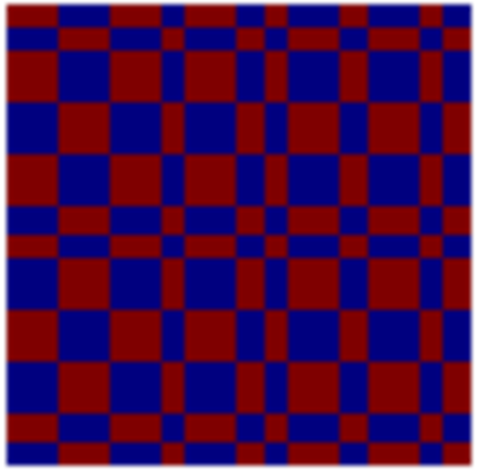
Testing: Marginalize out randomness

Examples:

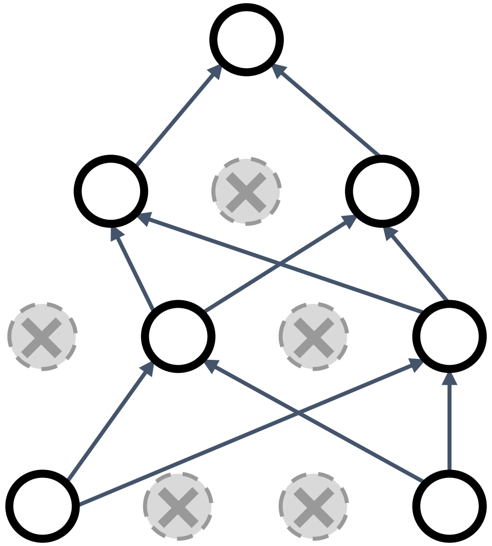
Batch Normalization

Data Augmentation

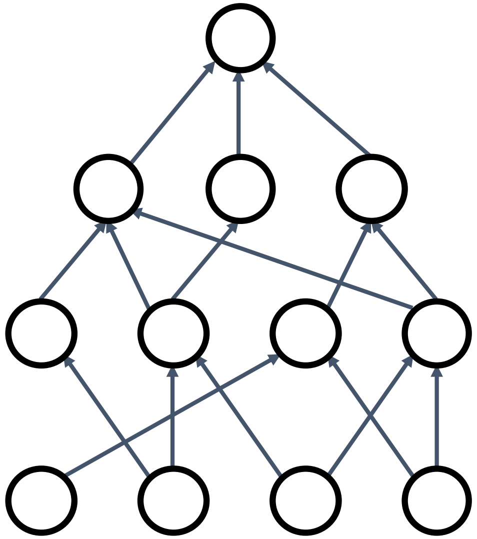
Fractional pooling



Dropout

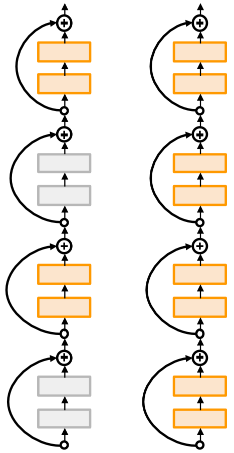


DropConnect



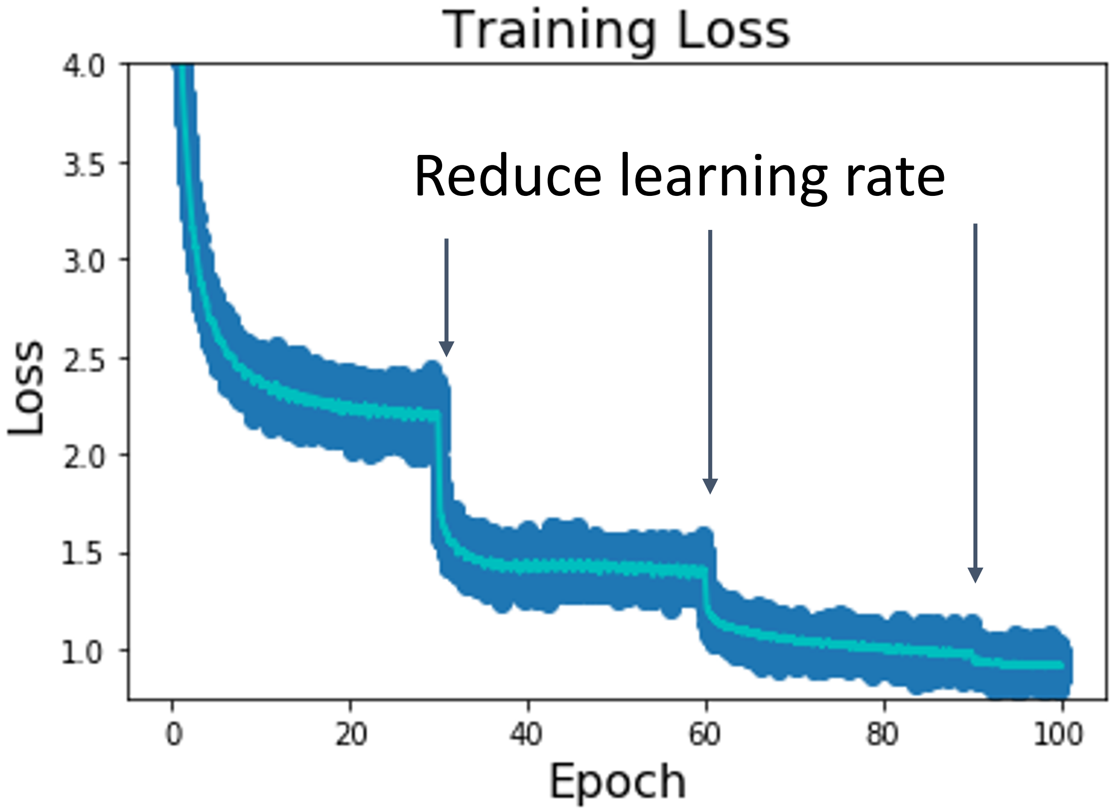
Cutout

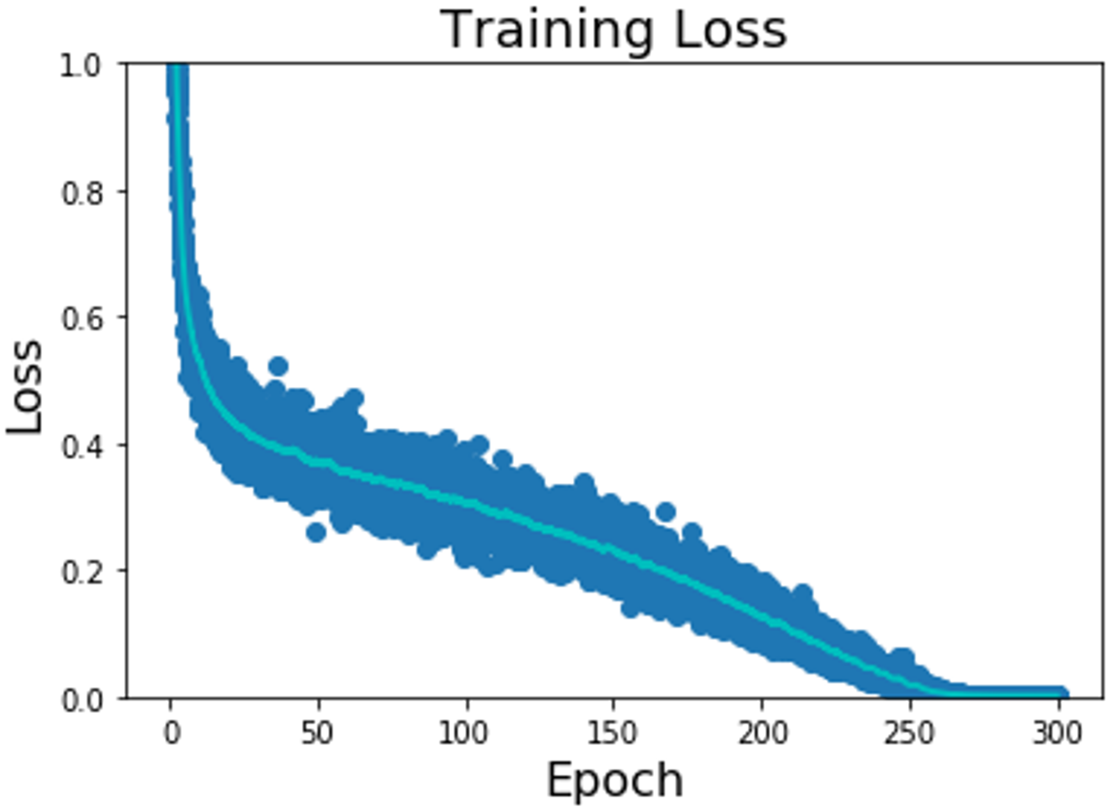
Stochastic Depth



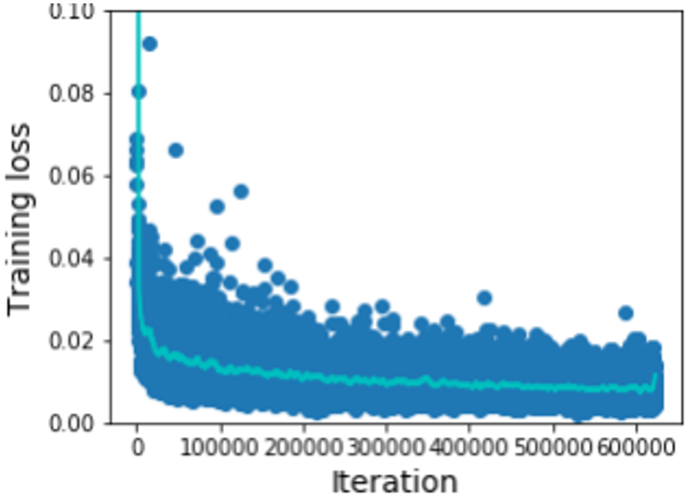
Mixup

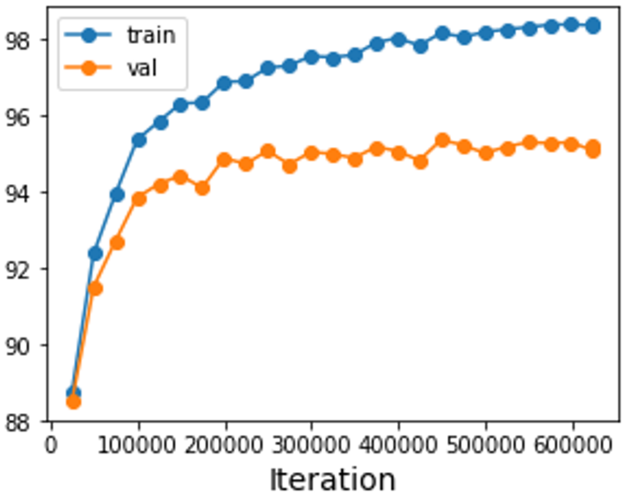
Training Neural Networks: Learning Rate Schedules

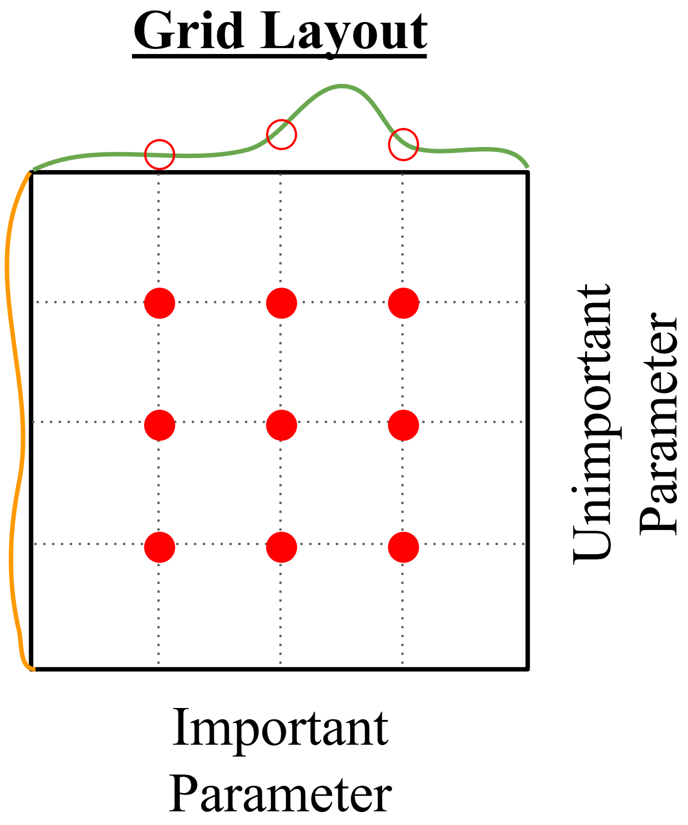


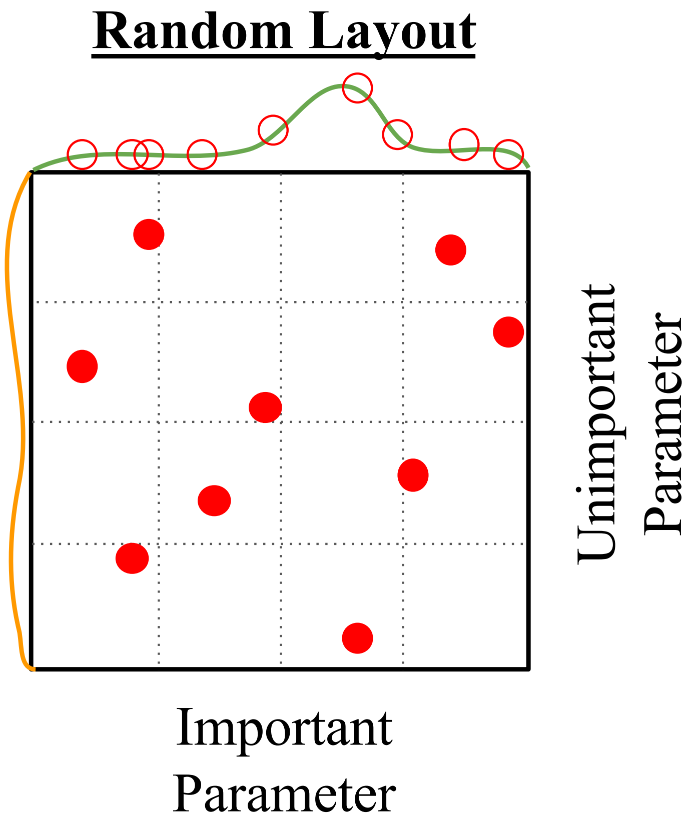


Training Neural Networks: Choosing Hyperarameters



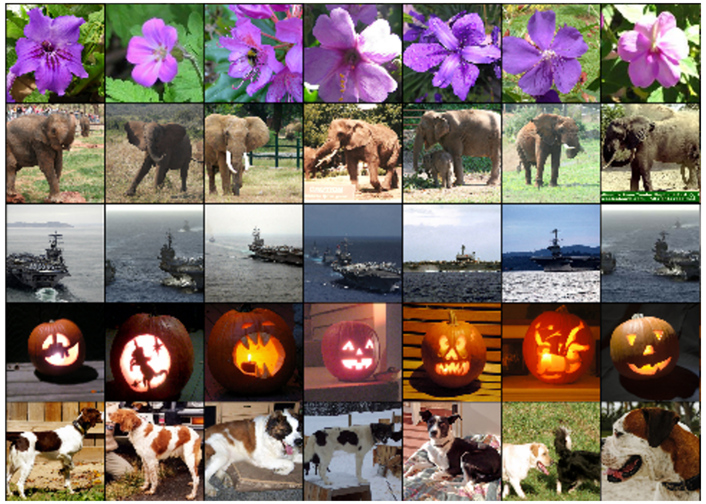






Visualizing and Understanding CNNs

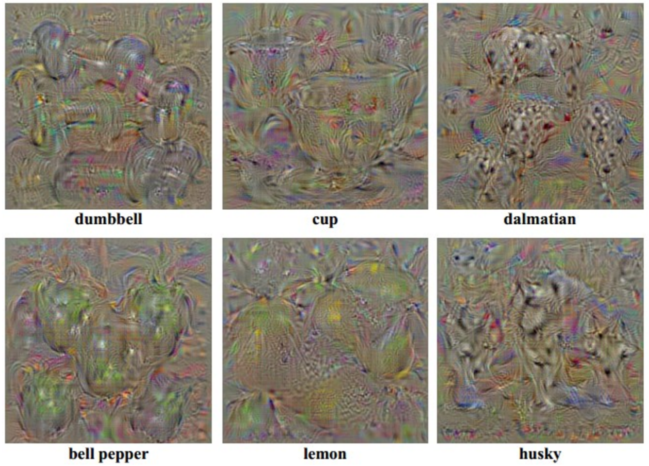
Nearest Neighbor

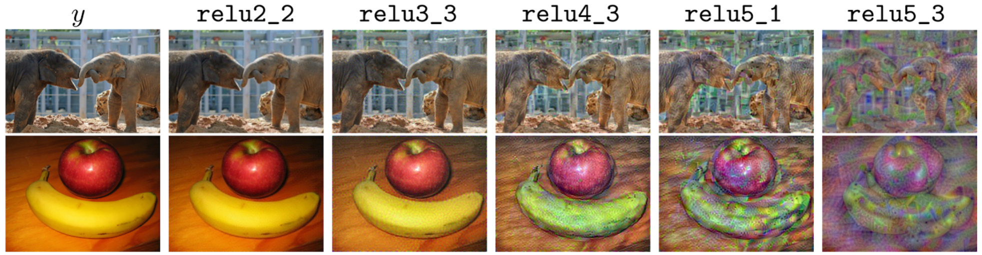


Maximally Activating Patches

(Guided) Backprop

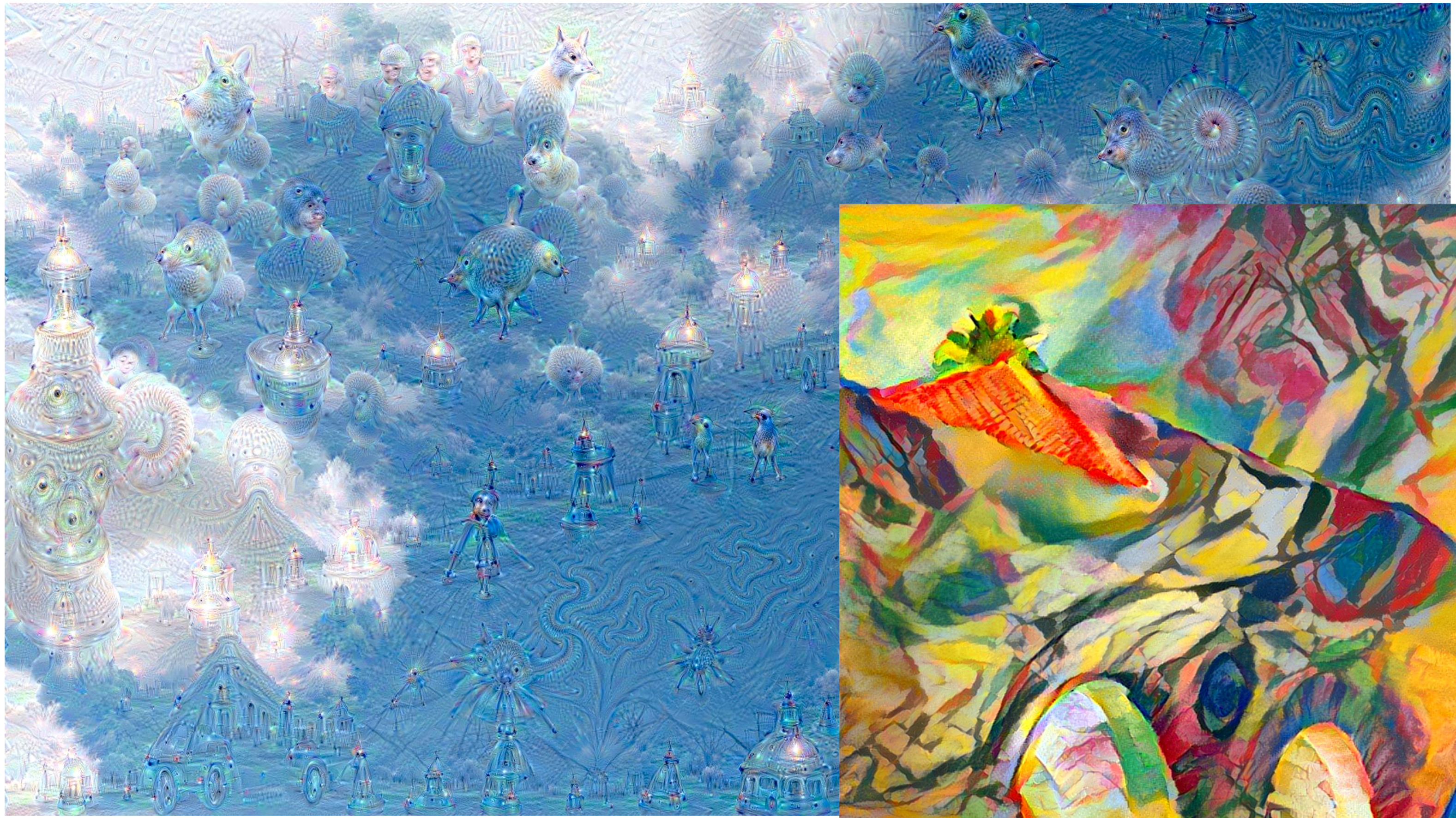
Synthetic Images via Gradient Ascent





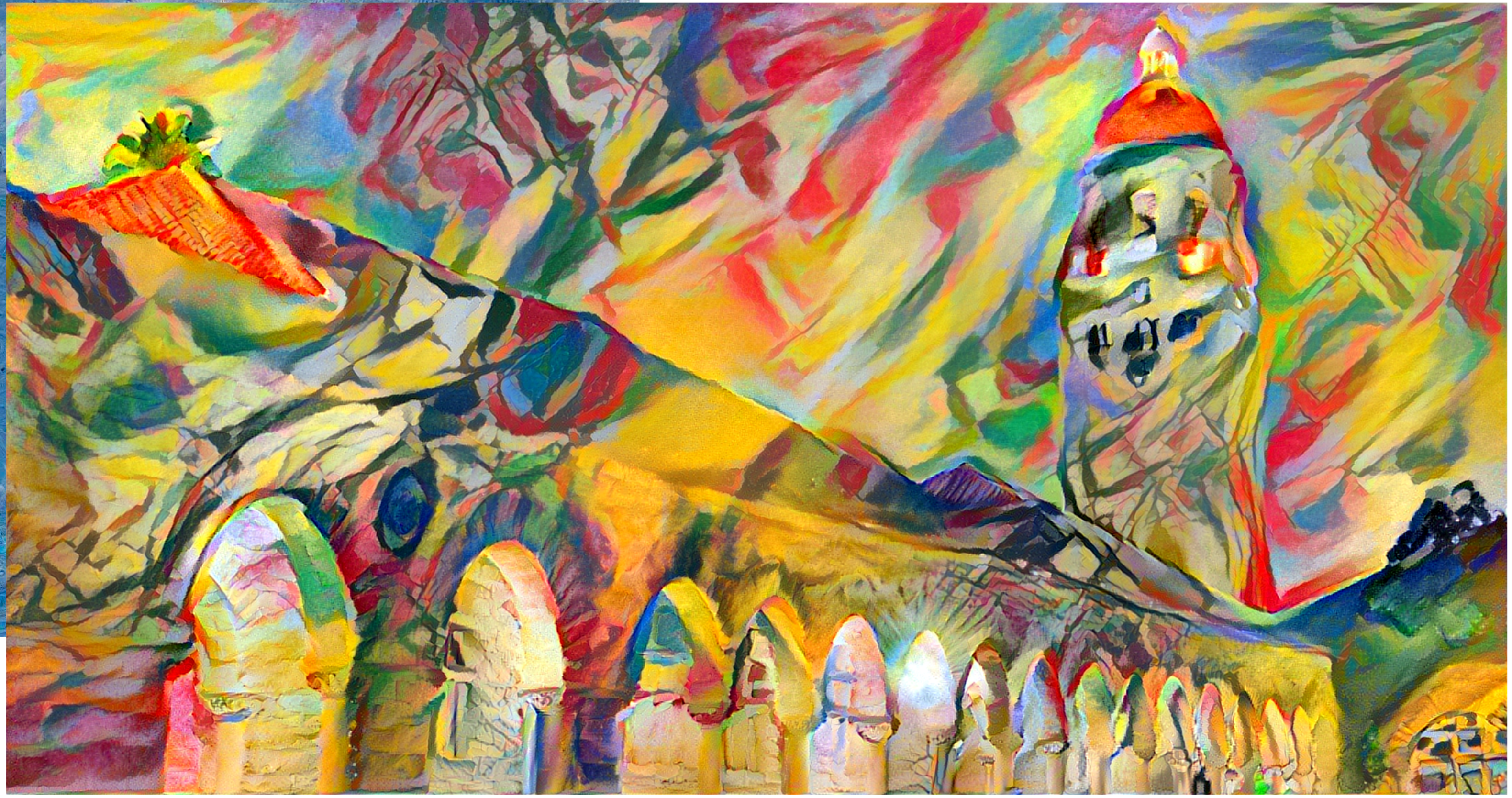
Feature Inversion

Making Art with CNNs

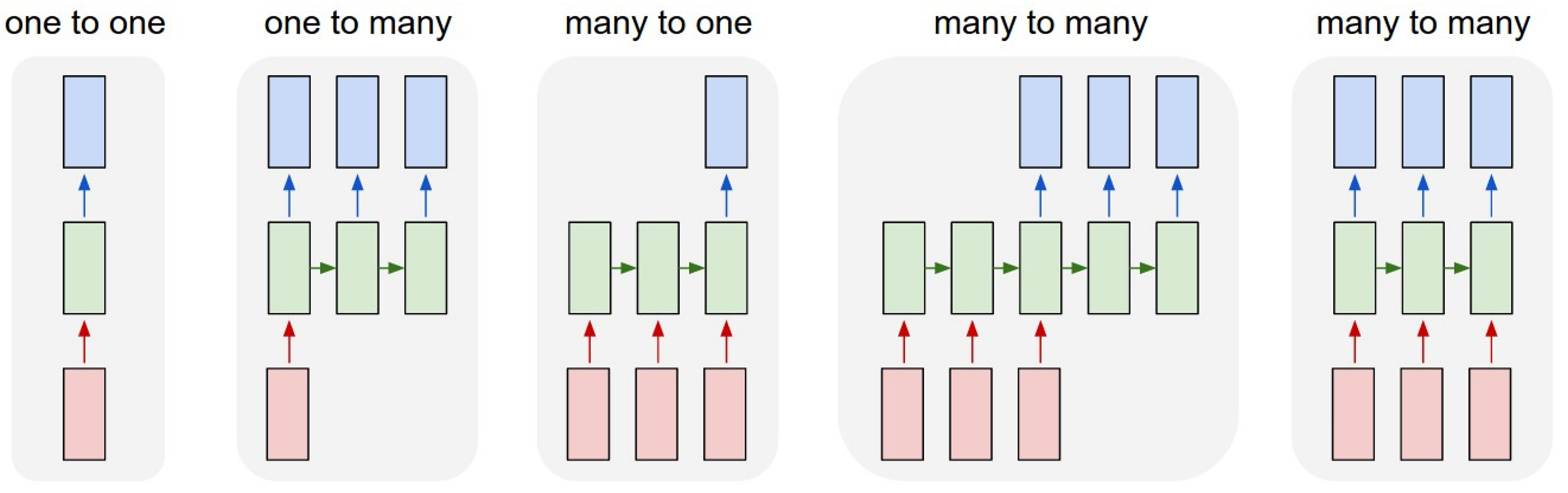


DeepDream

Style Transfer

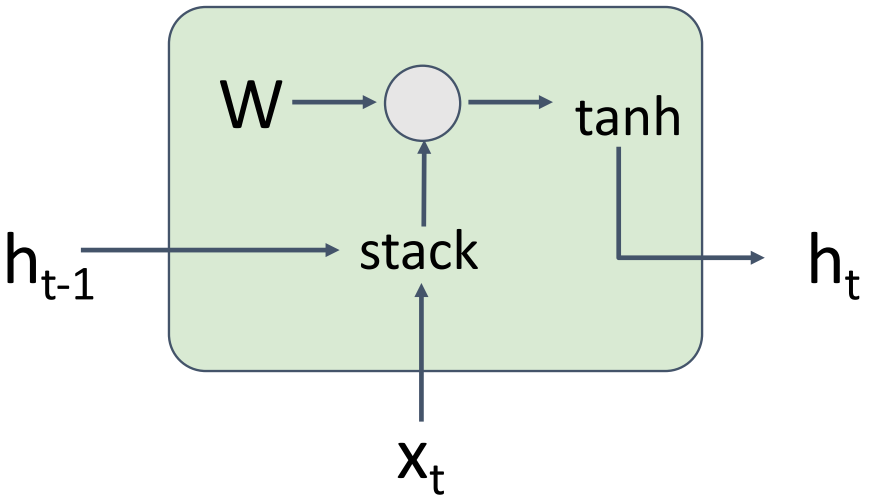


Recurrent Neural Networks: Process Sequences

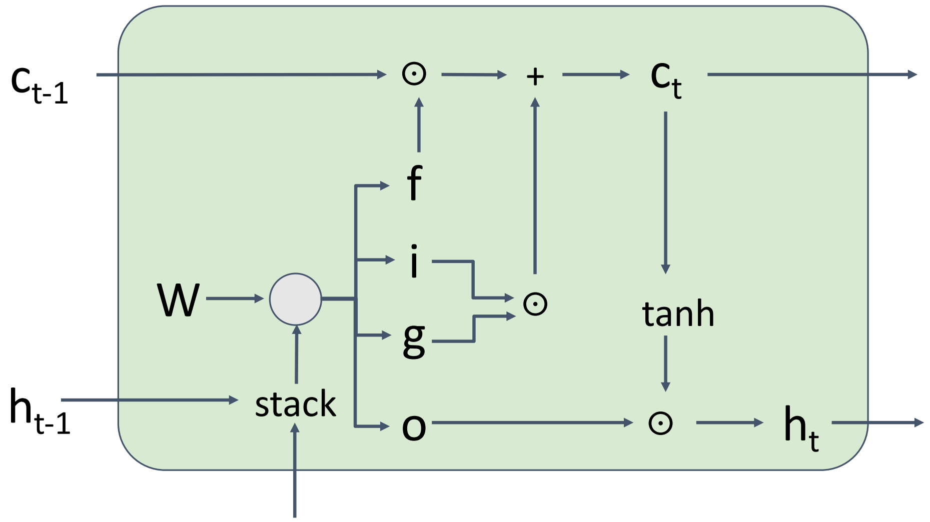


Recurrent Neural Networks: Architectures

Vanilla Recurrent Network

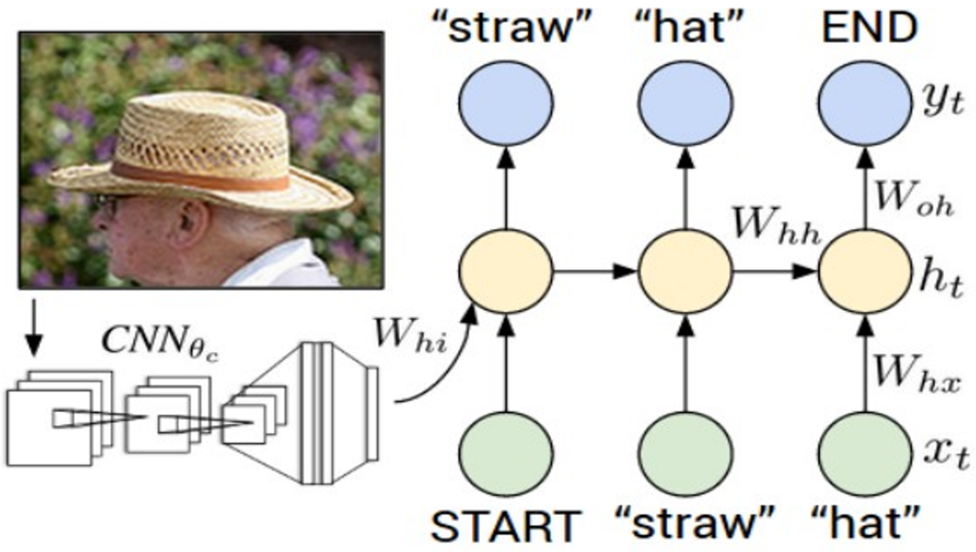


Long Short Term Memory (LSTM)



Recurrent Neural Networks: Image Captioning

Captions generated using neuraltalk2
All images are CC0 Public domain: cat suitcase, cat tree, dog, bear, surfers, tennis, giraffe, motorcycle



A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two giraffes standing in a grassy field

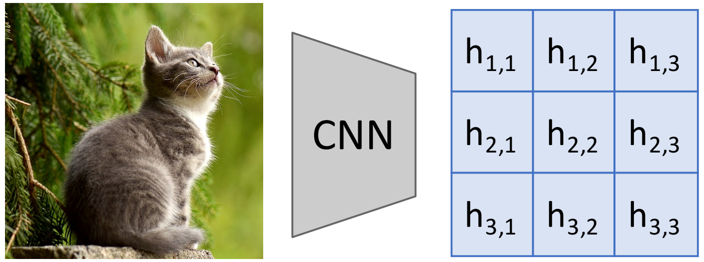
A man riding a dirt bike on a dirt track

Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015

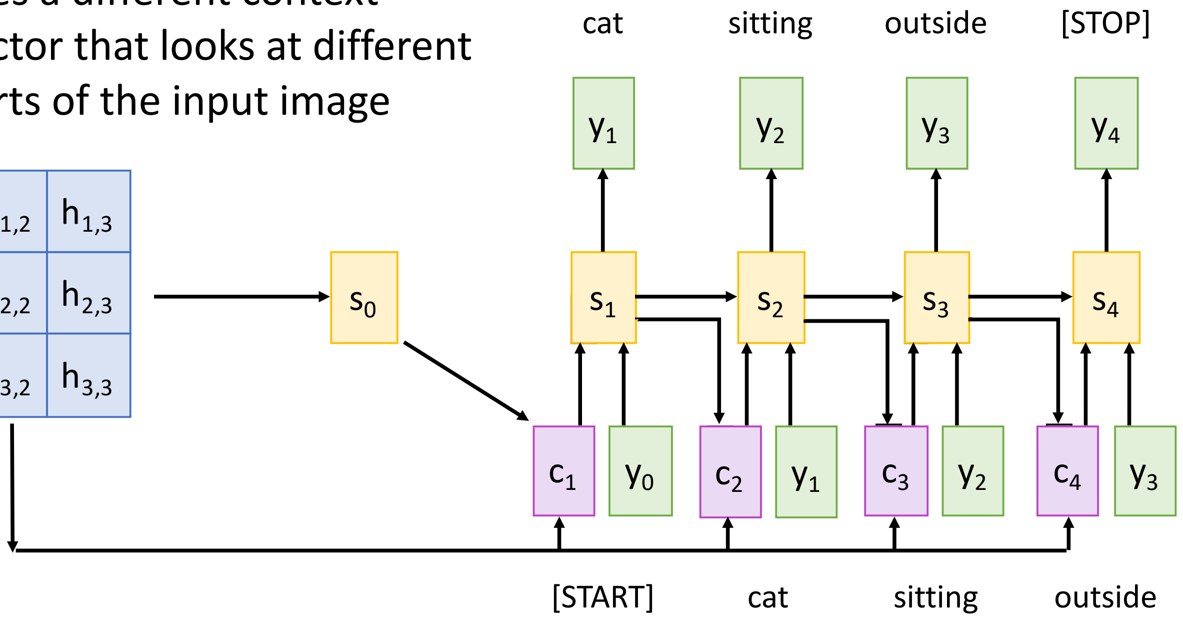
Attention

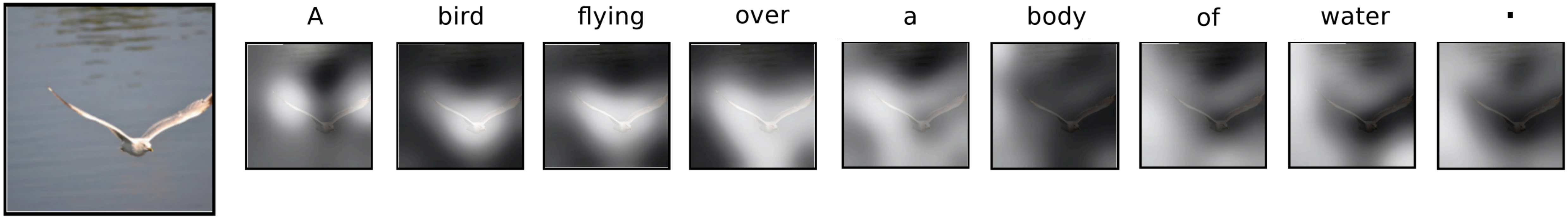
$e_{t,i,j} = f_{att}(s_{t-1}, h_{i,j})$
$a_{t,:,:} = \text{softmax}(e_{t,:,:})$
$c_t = \sum_{i,j} a_{t,i,j} h_{i,j}$

Each timestep of decoder uses a different context vector that looks at different parts of the input image



Use a CNN to compute a grid of features for an image

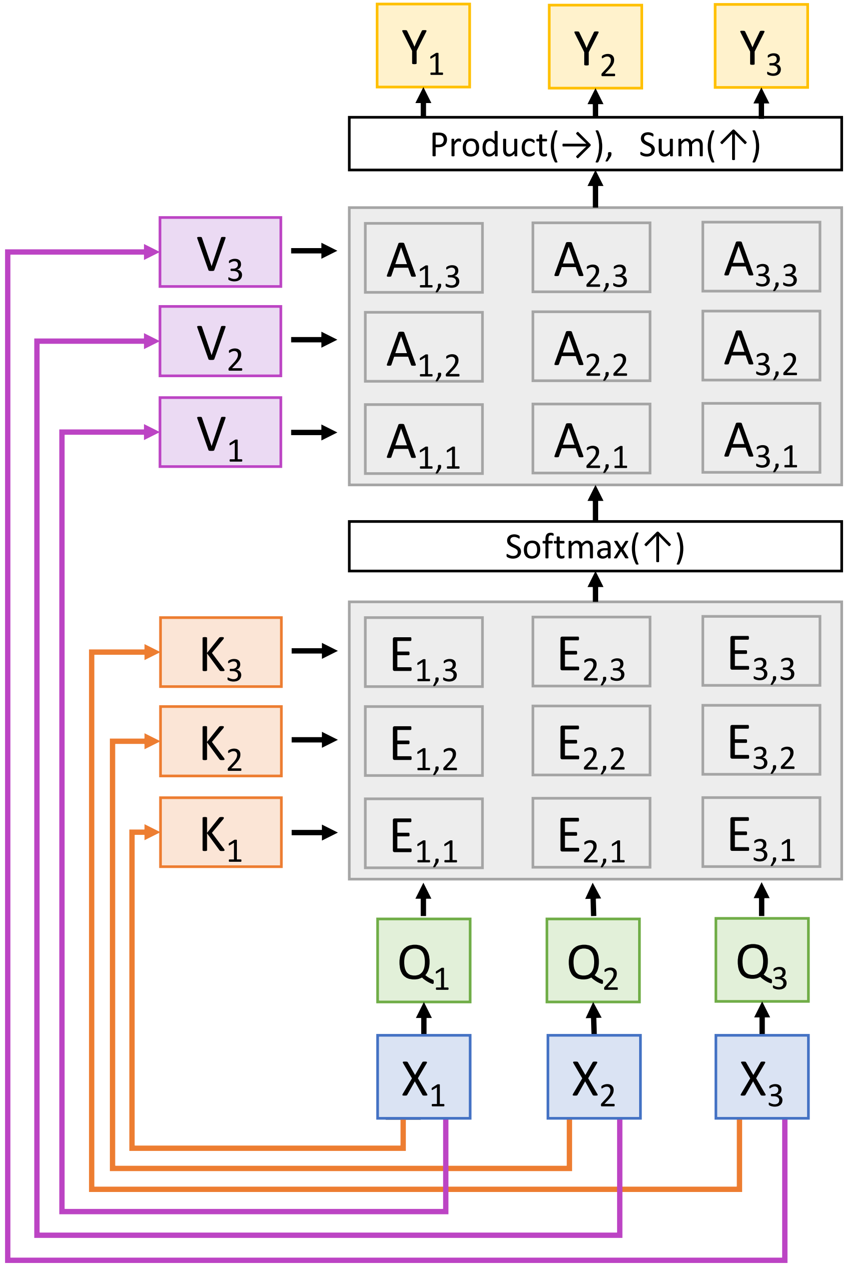




Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Self-Attention Layer

One **query** per **input vector**



Inputs:

Input vectors: **X** (Shape: N_X x D_X)

Key matrix: **W_K** (Shape: D_X x D_Q)

Value matrix: W_V (Shape: D_X x D_V)

Query matrix: **W_Q** (Shape: D_X x D_Q)

Computation:

Query vectors: **Q** = **XW_Q**

Key vectors: **K** = **XW_K** (Shape: N_X x D_Q)

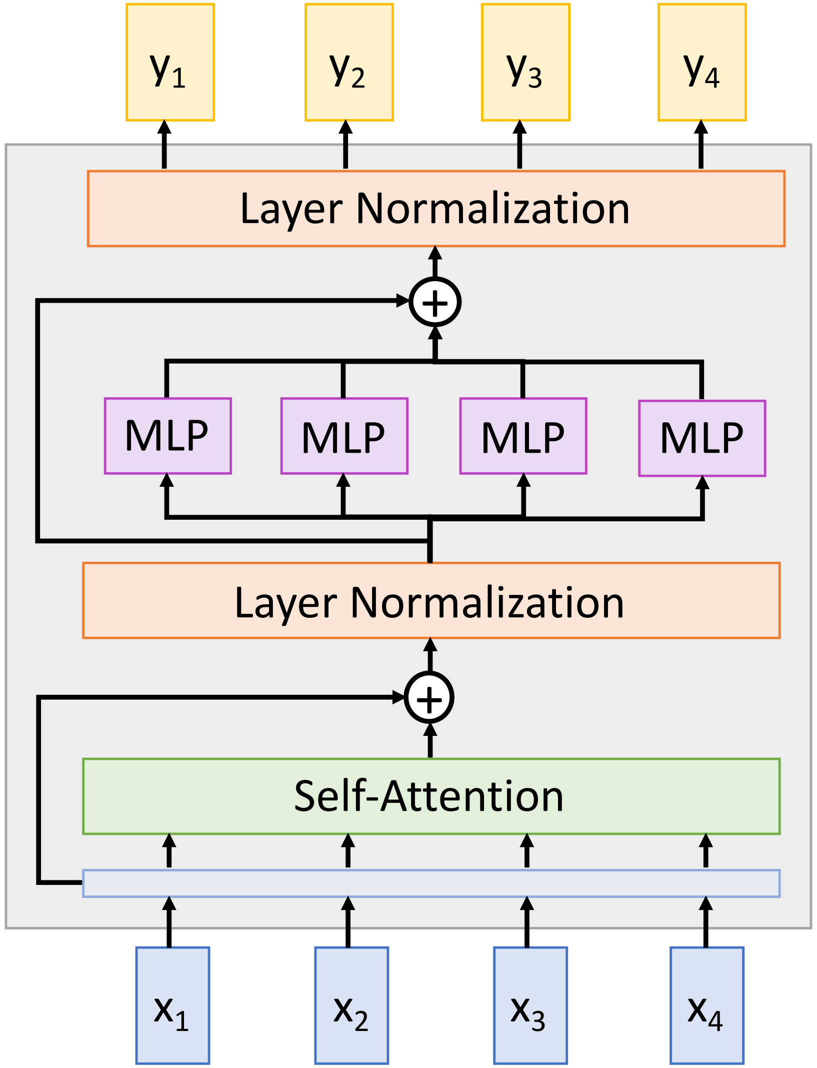
Value Vectors: **V** = **XW_V** (Shape: N_X x D_V)

Similarities: E = **QK^T** (Shape: N_X x N_X) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

Attention weights: A = softmax(E, dim=1) (Shape: N_X x N_X)

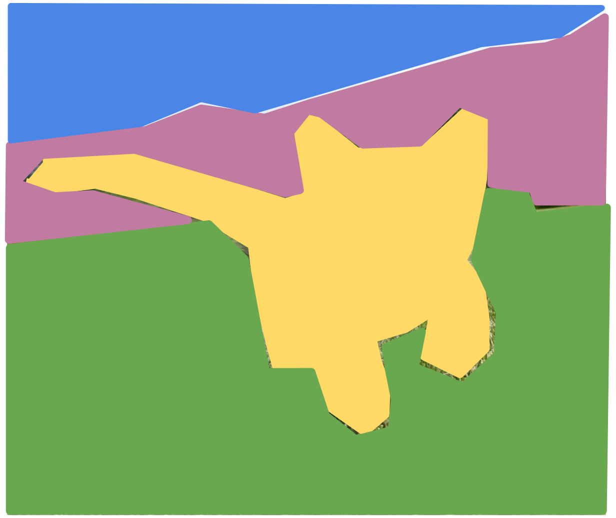
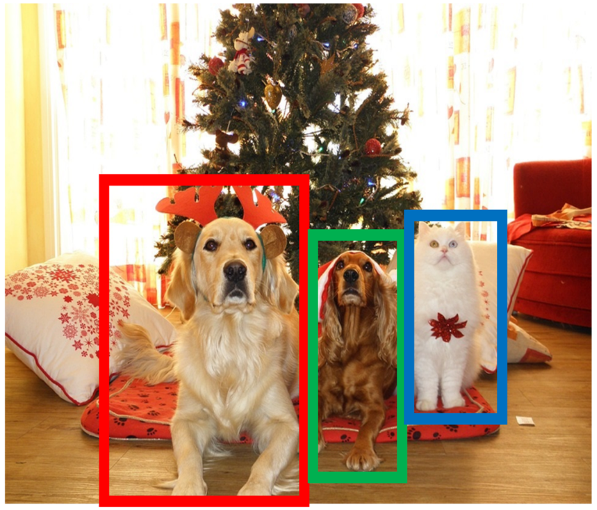
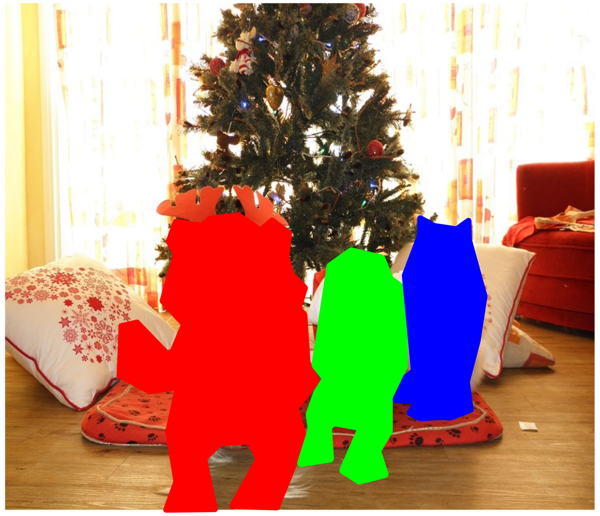
Output vectors: Y = A**V** (Shape: N_X x D_V) $Y_i = \sum_j A_{i,j} V_j$

Attention is all you need: The Transformer



Vaswani et al, "Attention is all you need", NeurIPS 2017

Computer Vision Tasks

Classification	Semantic Segmentation	Object Detection	Instance Segmentation
			
CAT	GRASS, CAT, TREE, SKY	DOG, DOG, CAT	DOG, DOG, CAT

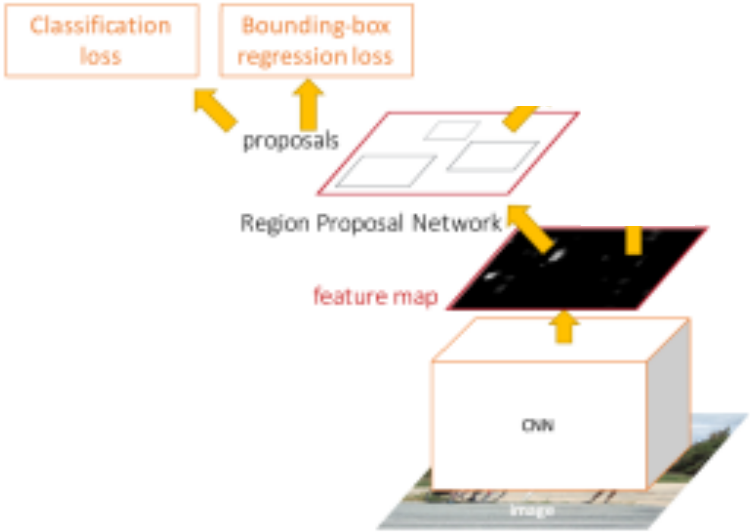
No spatial extent

No objects, just pixels

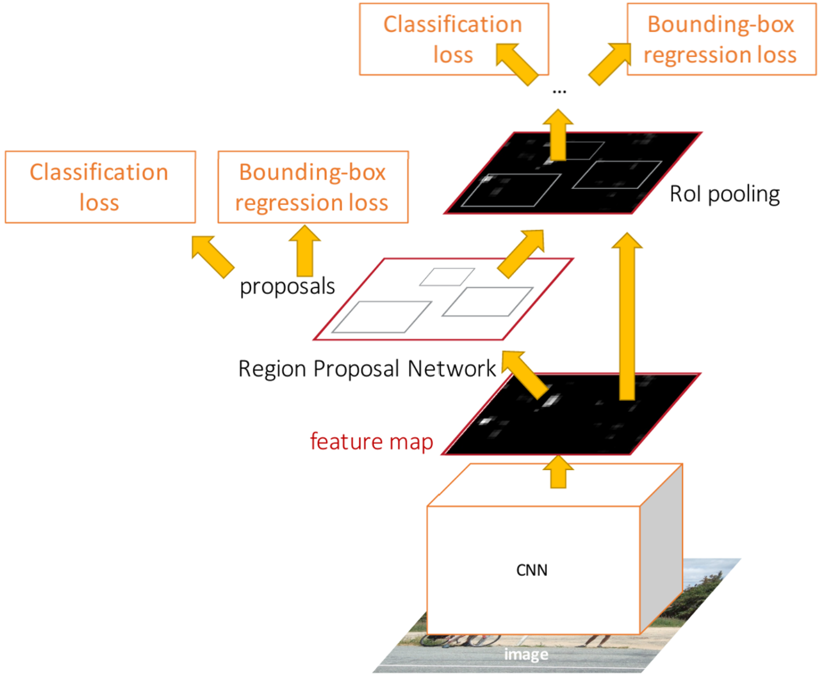
Multiple Objects

Object Detection: Single Stage vs Two Stage

Single-Stage:
YOLO, SSD, RetinaNet
Make all predictions
with a CNN



Two-Stage:
Faster R-CNN
Use RPN to predict proposals,
classify them with second stage



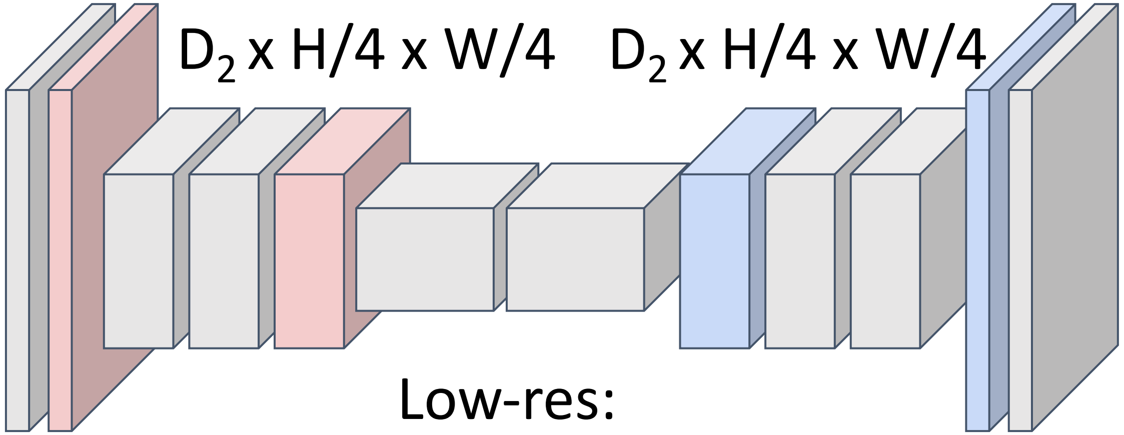
Semantic Segmentation: Fully Convolutional Network

Downsampling: Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling: linterpolation, transposed conv

Input:
$3 \times H \times W$



High-res:
$D_1 \times H/2 \times W/2$

Med-res:
$D_2 \times H/4 \times W/4$

Low-res:
$D_3 \times H/4 \times W/4$

Med-res:
$D_2 \times H/4 \times W/4$

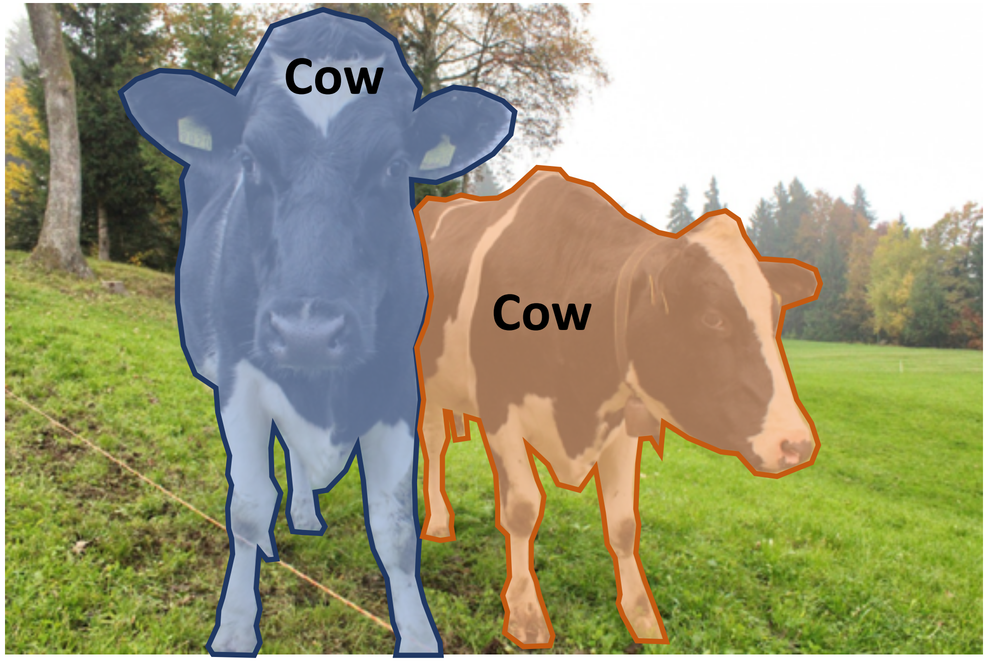
High-res:
$D_1 \times H/2 \times W/2$

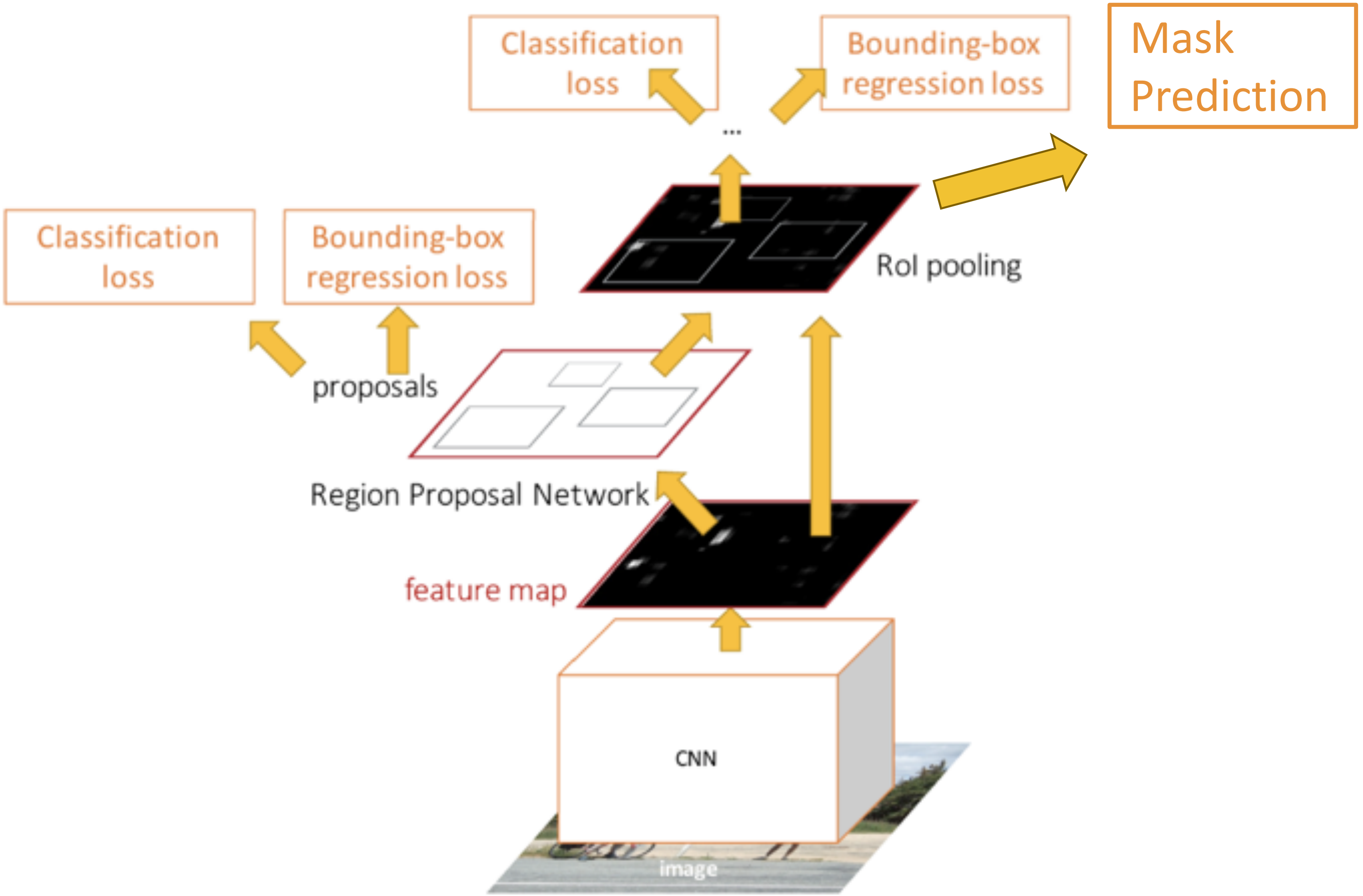
Predictions:
$H \times W$

Loss function: Per-Pixel cross-entropy

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Instance Segmentation: Detection + Segmentation



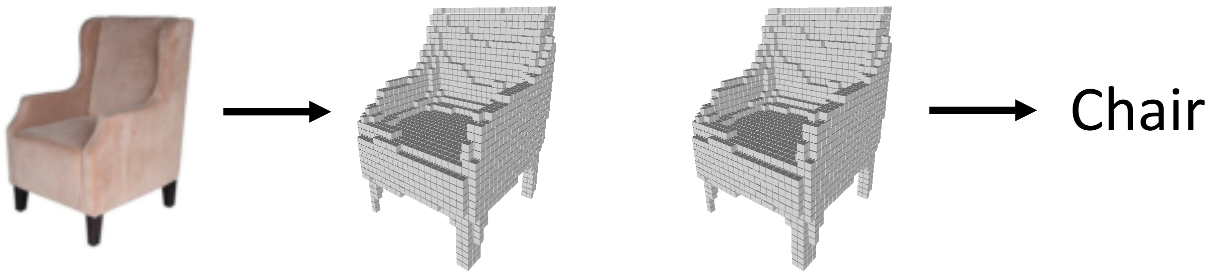


He et al, "Mask R-CNN", ICCV 2017

Adding a Dimension: 3D Deep Learning

Predicting 3D Shapes
from single image

Processing 3D
input data

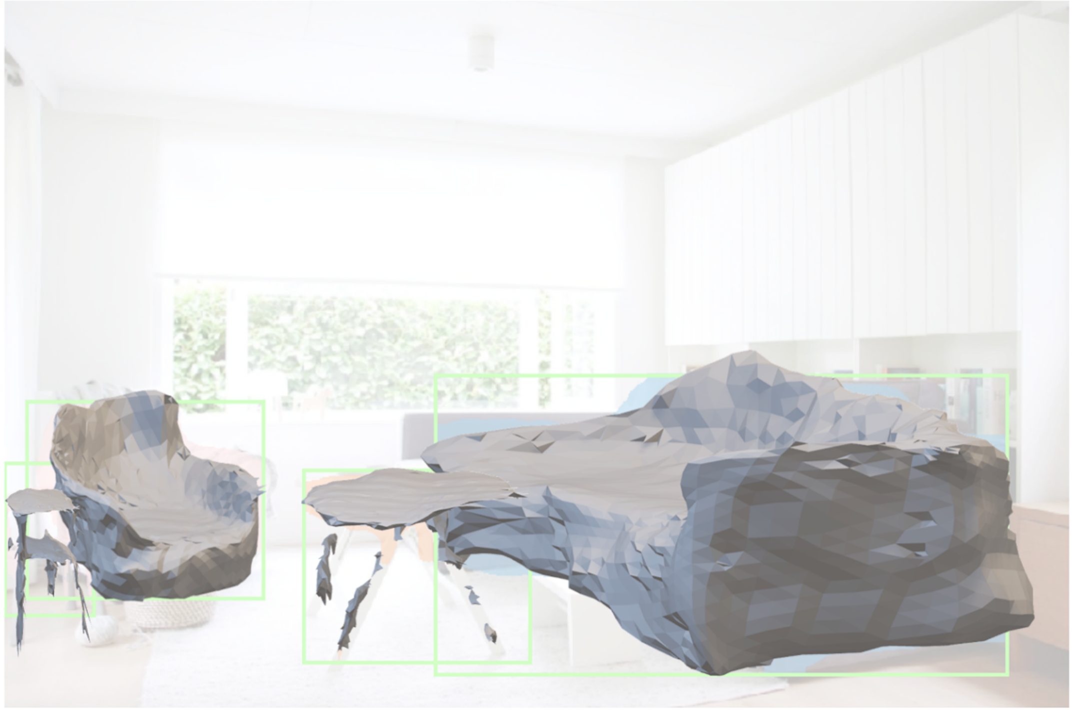


Input Image

3D Shape

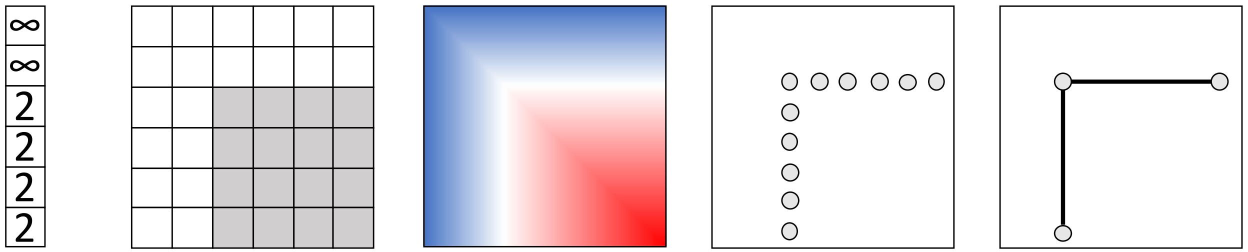
3D Shape

Mesh R-CNN



Gkioxari, Malik, and Johnson, ICCV 2019

3D Shape Representations



Depth Map

Voxel Grid

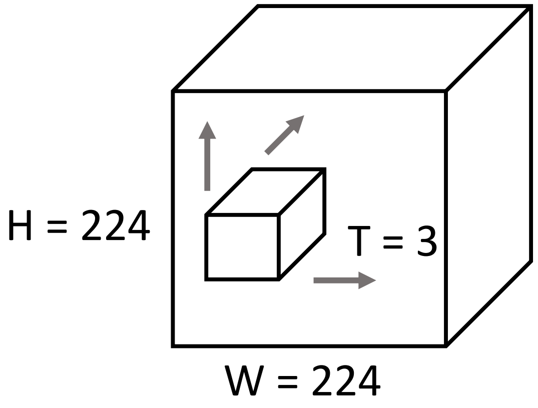
Implicit Surface

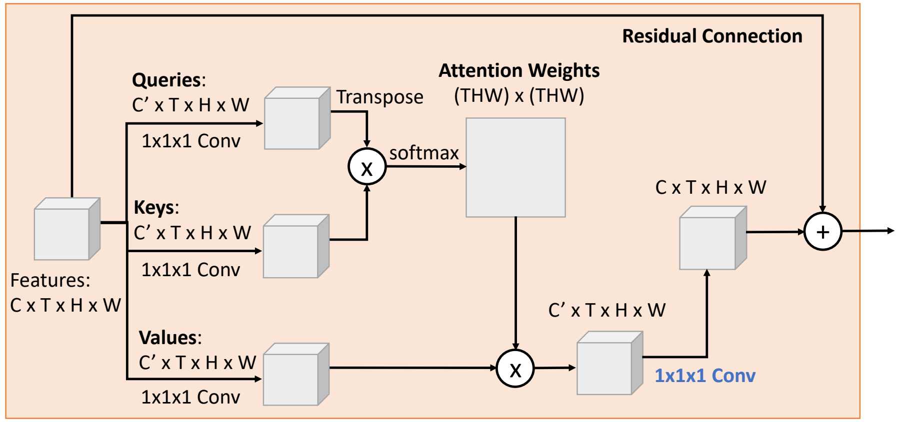
Pointcloud

Mesh

Adding a Dimension: Deep Learning on Video

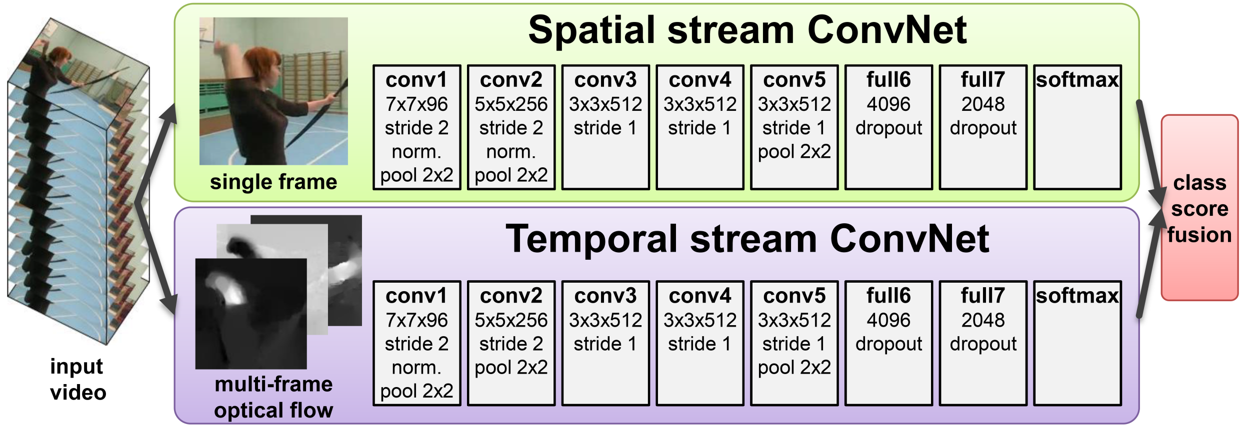
3D CNNs

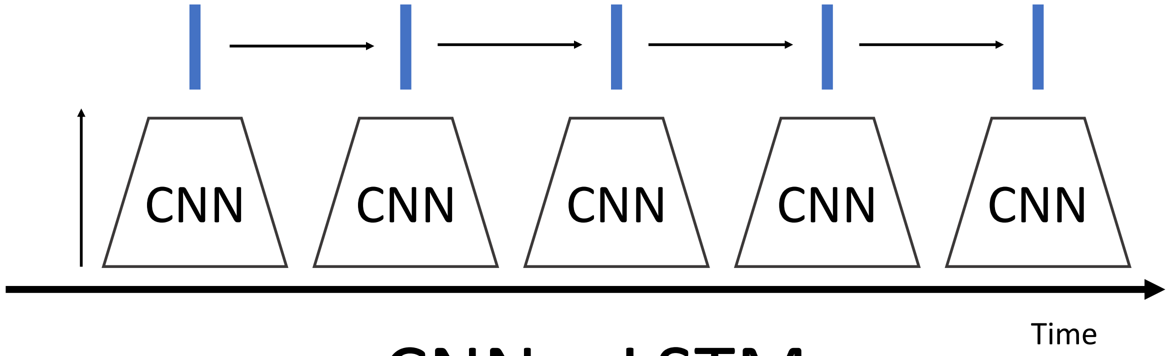




Self-Attention

Two Stream Networks





CNN + LSTM

Generative Models

Autoregressive Models directly maximize likelihood of training data:

$$p_\theta(x) = \prod_{i=1}^{N} p_\theta(x_i | x_1, \ldots, x_{i-1})$$

Good image quality, can evaluate with perplexity. Slow to generate data, needs tricks to scale up.

Variational Autoencoders introduce a latent z, and maximize a lower bound:

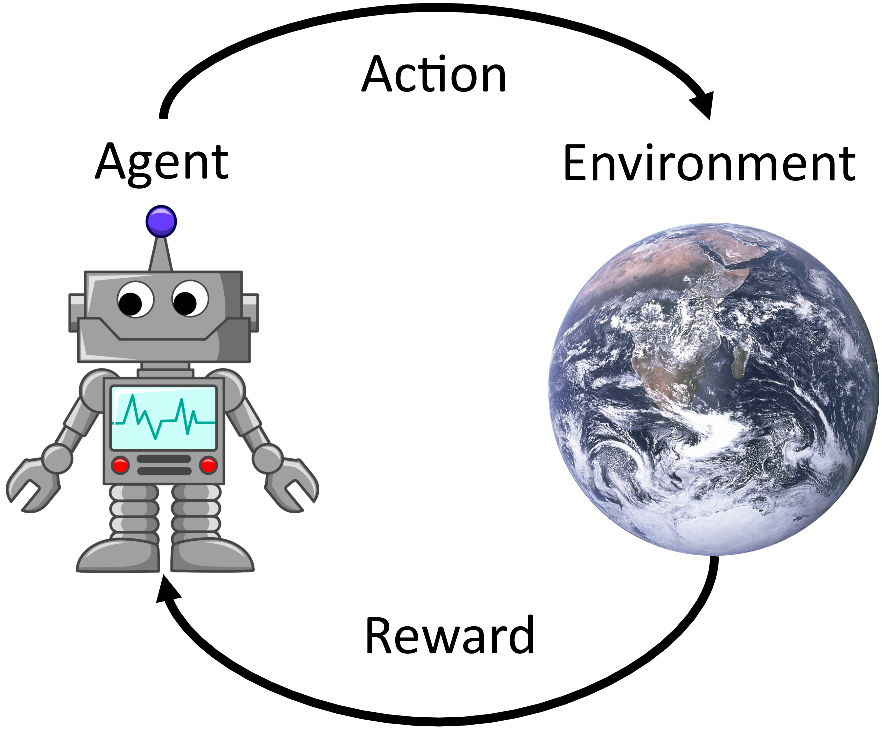
$$p_\theta(x) = \int_Z p_\theta(x|z)p(z)dz \geq E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$

Latent z allows for powerful interpolation and editing applications.

Generative Adversarial Networks give up on modeling p(x), but allow us to draw samples from p(x). Difficult to evaluate, but best qualitative results today

Reinforcement Learning

RL trains **agents** that interact with an **environment** and learn to maximize **reward**



Q-Learning: Train network $Q_\theta(s, a)$ to estimate future rewards for every (state, action) pair. Use Bellman Equation to define loss function for training Q

Policy Gradients: Train a network $\pi_\theta(a \mid s)$ that takes state as input, gives distribution over which action to take in that state. Use REINFORCE Rule for computing gradients

What’s Next?

Prediction #1: We will discover interesting new types of deep models

Example: Neural ODE

Residual Network: $h_{t+1} = h_t + f(h_t, \theta_t)$

Looks kind of like numerical integration…

Chen et al, "Neural Ordinary Differential Equations", NeurIPS 2018

Example: Neural ODE

Residual Network: $h_{t+1} = h_t + f(h_t, \theta_t)$
Looks kind of like numerical integration…

Neural ODE: Hidden "states" are the solutions of

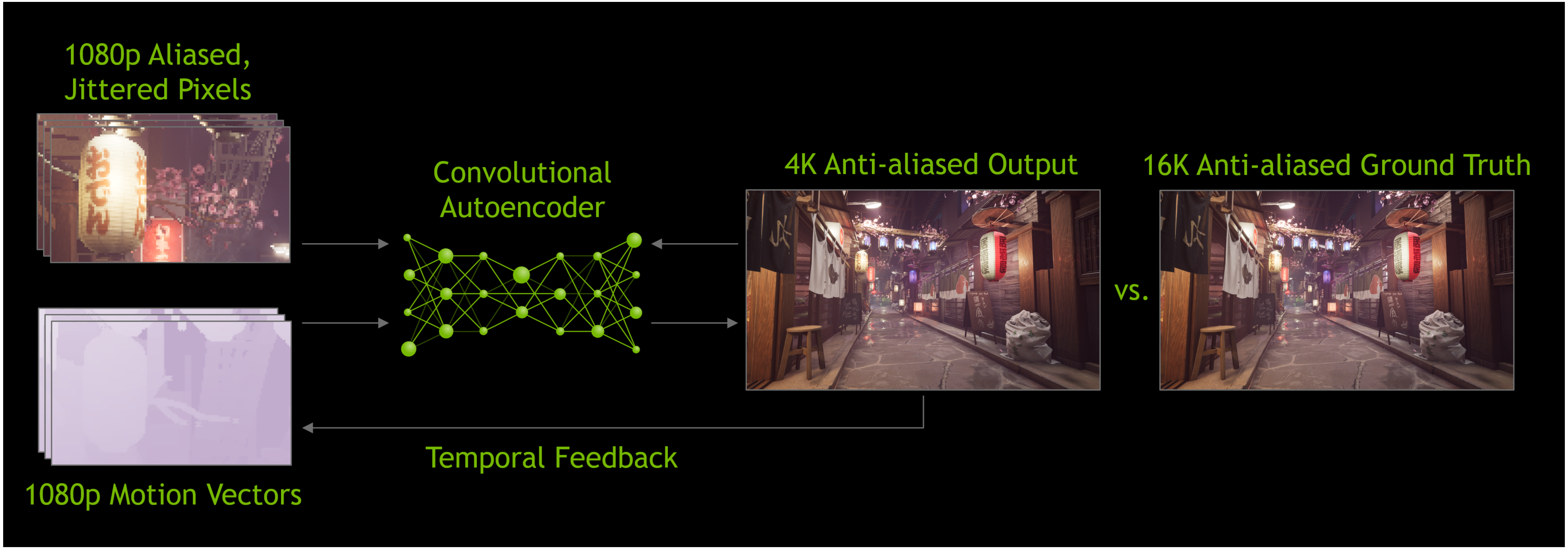
$$\frac{dh}{dt} = f(h(t), t, \theta)$$

A deep network with infinitely many layers!

Chen et al, "Neural Ordinary Differential Equations", NeurIPS 2018

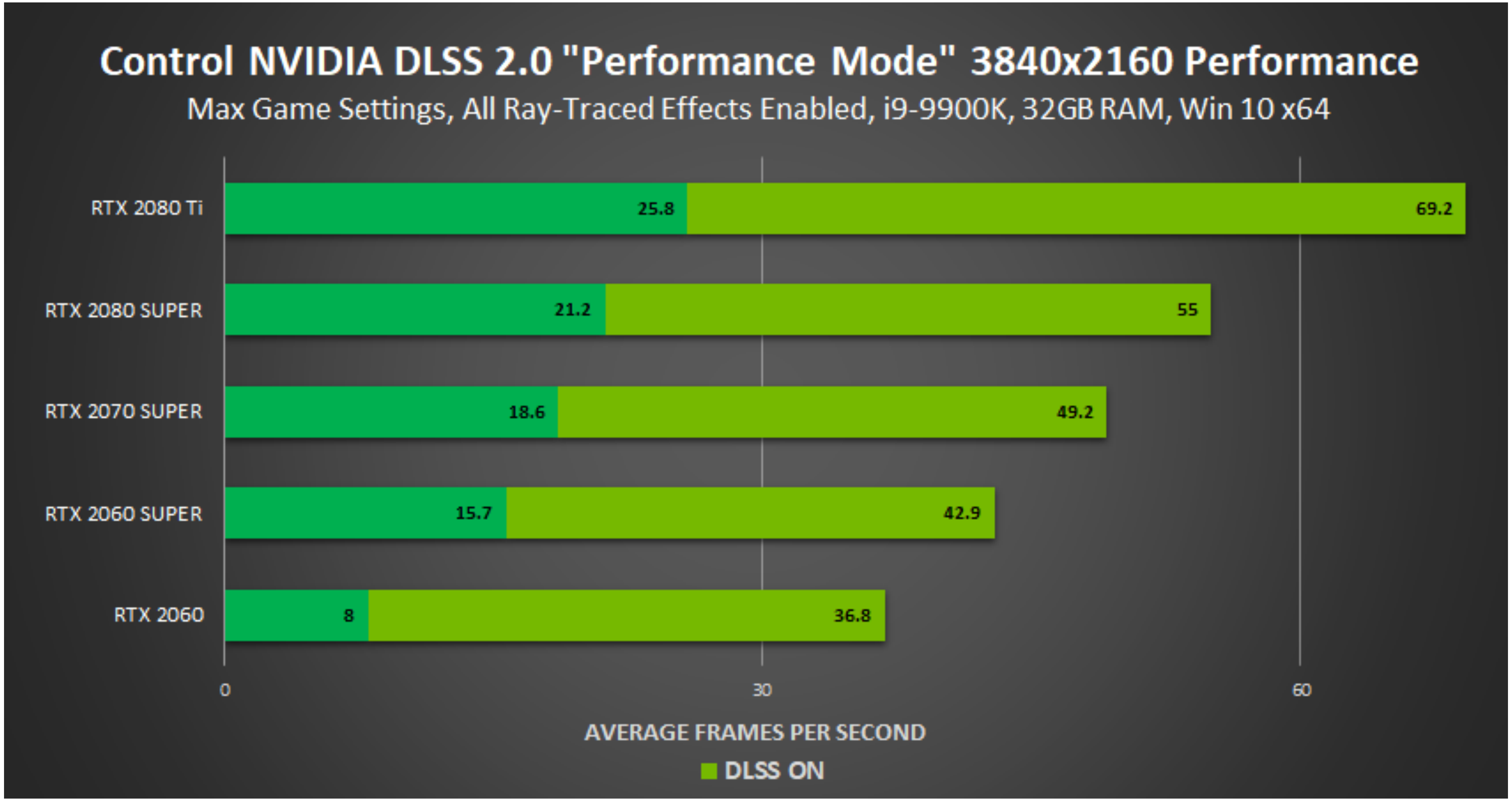
Prediction #2: Deep Learning will find new applications

Deep Learning for Graphics: NVIDIA DLSS



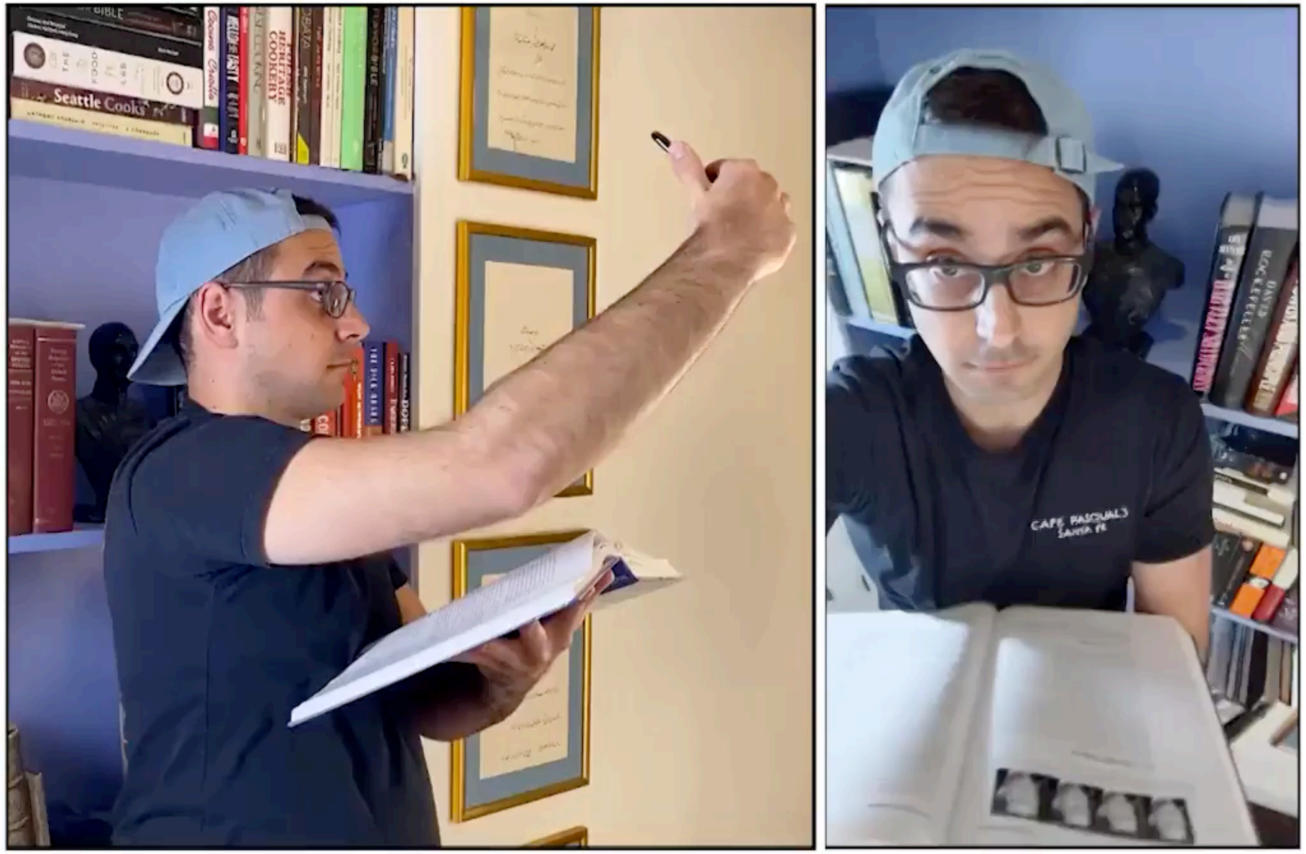
https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/

Deep Learning for Graphics: NVIDIA DLSS



https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/

Deep Learning for Graphics: Nerfie

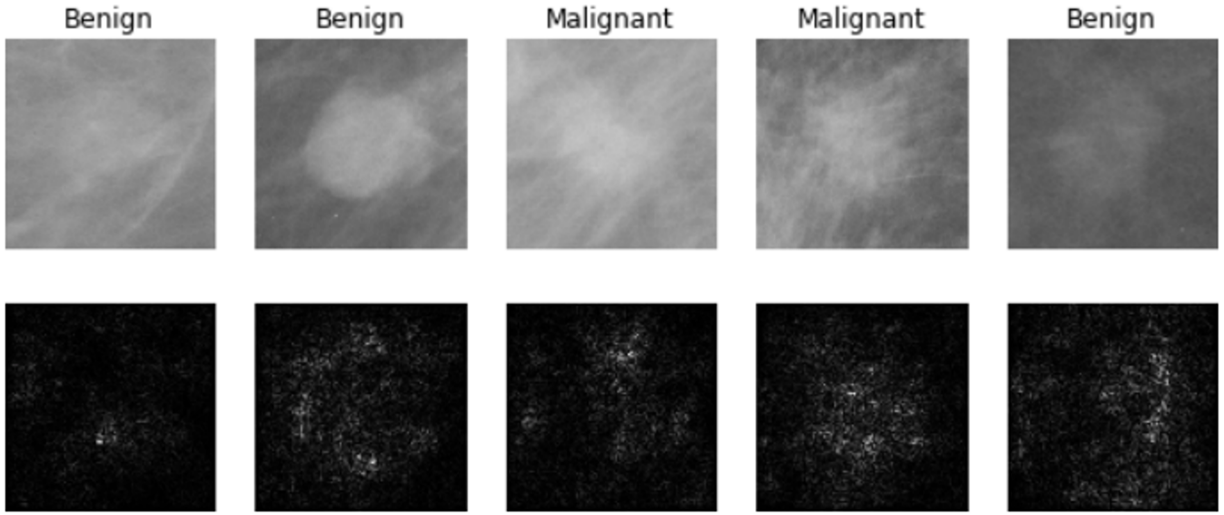


(a) Capture Process (b) Input

Park et al, “Deformable Neural Radiance Fields”, arXiv 2020, https://nerfies.github.io/

Deep Learning for scientific applications

Medical Imaging



Levy et al, 2016 Figure reproduced with permission

Galaxy Classification

Dieleman et al, 2014

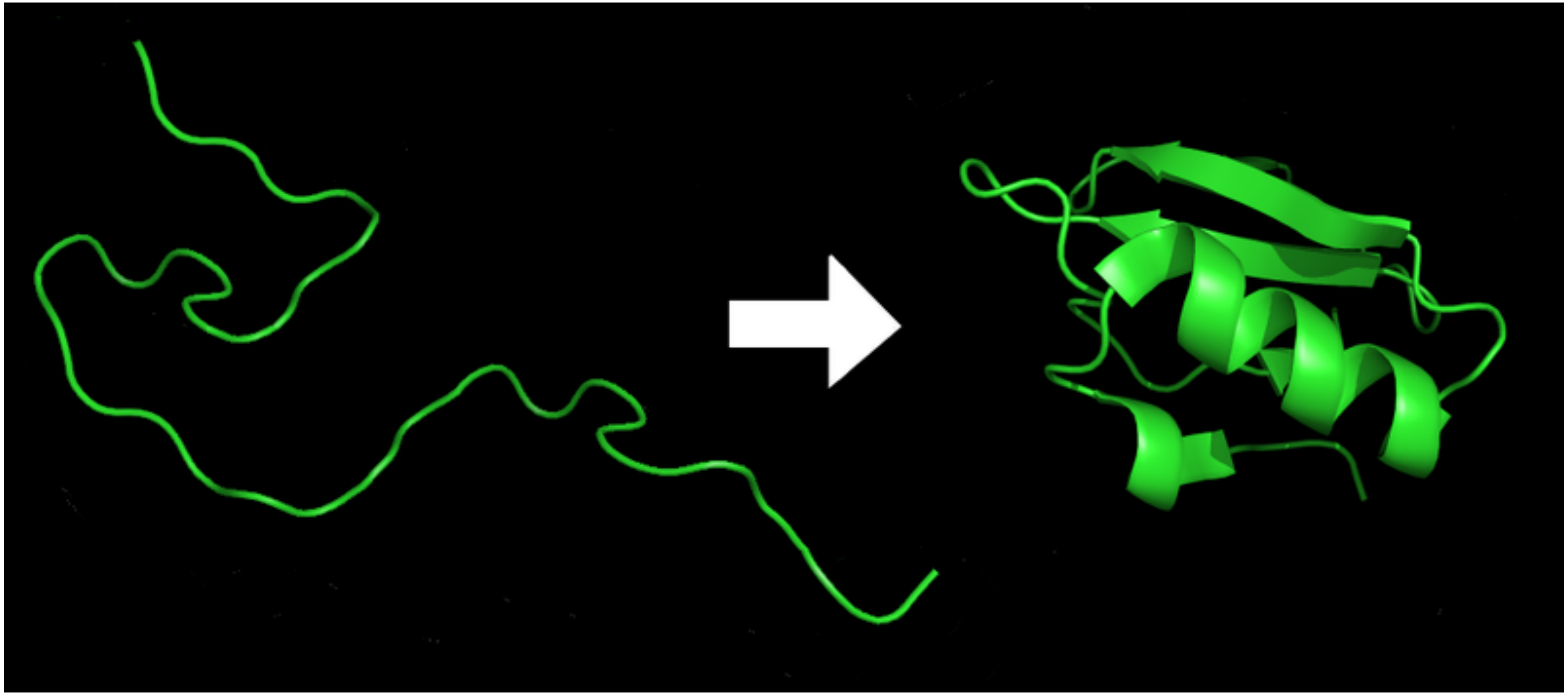
From left to right: public domain by NASA, usage permitted by ESA/Hubble, public domain by NASA, and public domain.

Whale recognition

Kaggle Challenge

This image by Christin Khan is in the public domain and originally came from the U.S. NOAA.

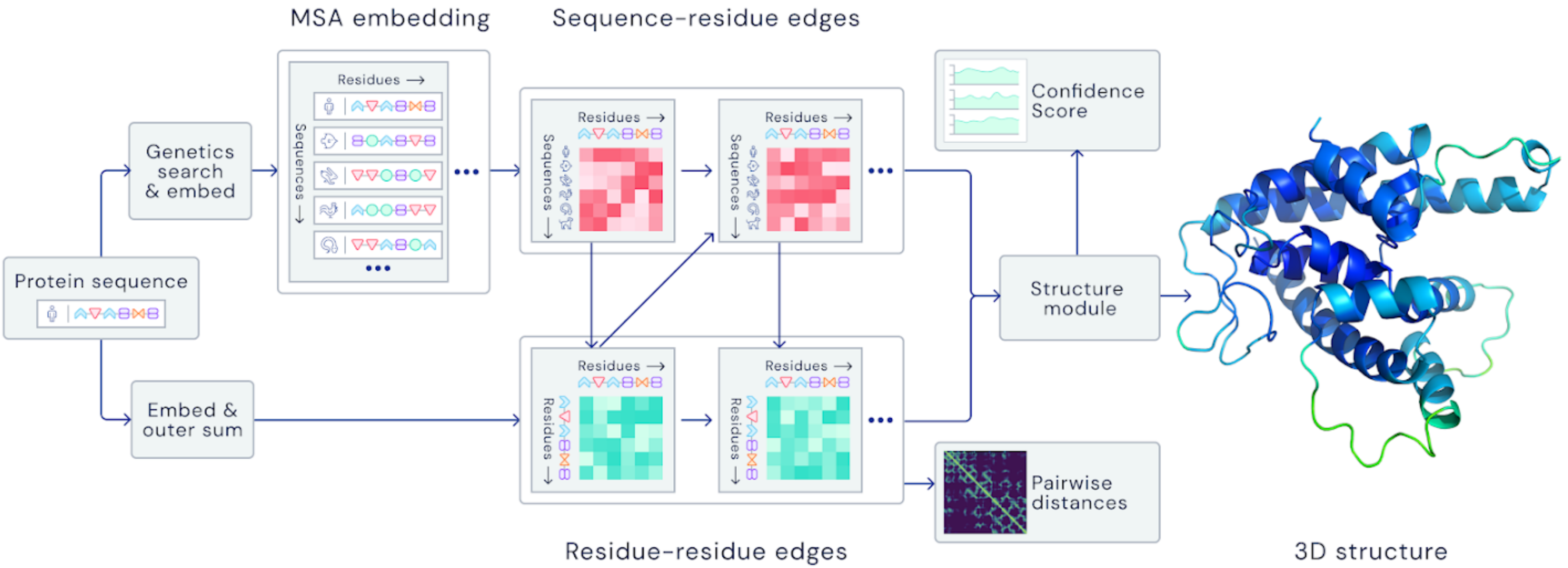
Deep Learning for Science: **Protein Folding**



Input: 1D sequence of amino acids

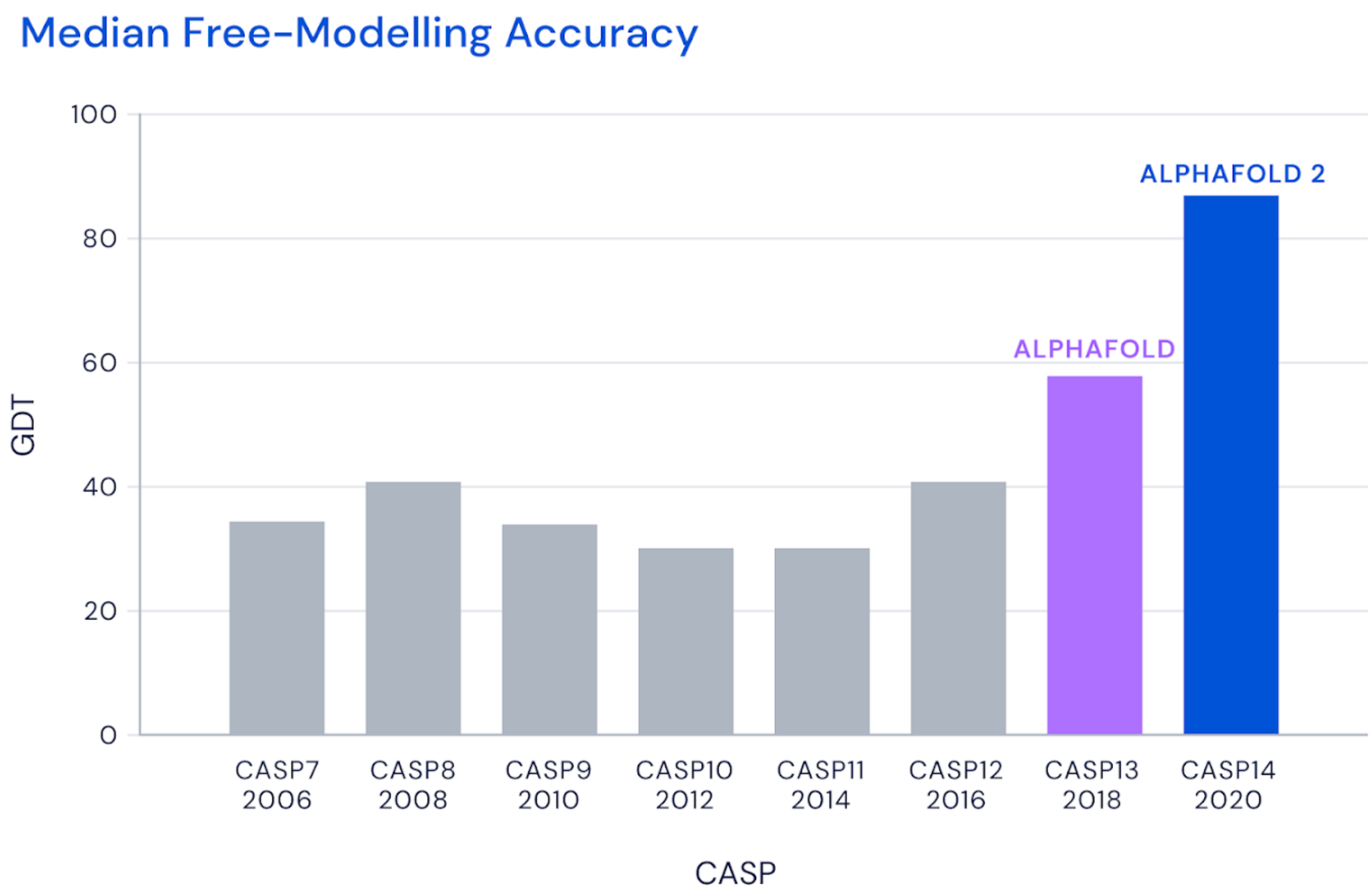
Output: 3D protein structure

Deep Learning for Science: **AlphaFold 2**



https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

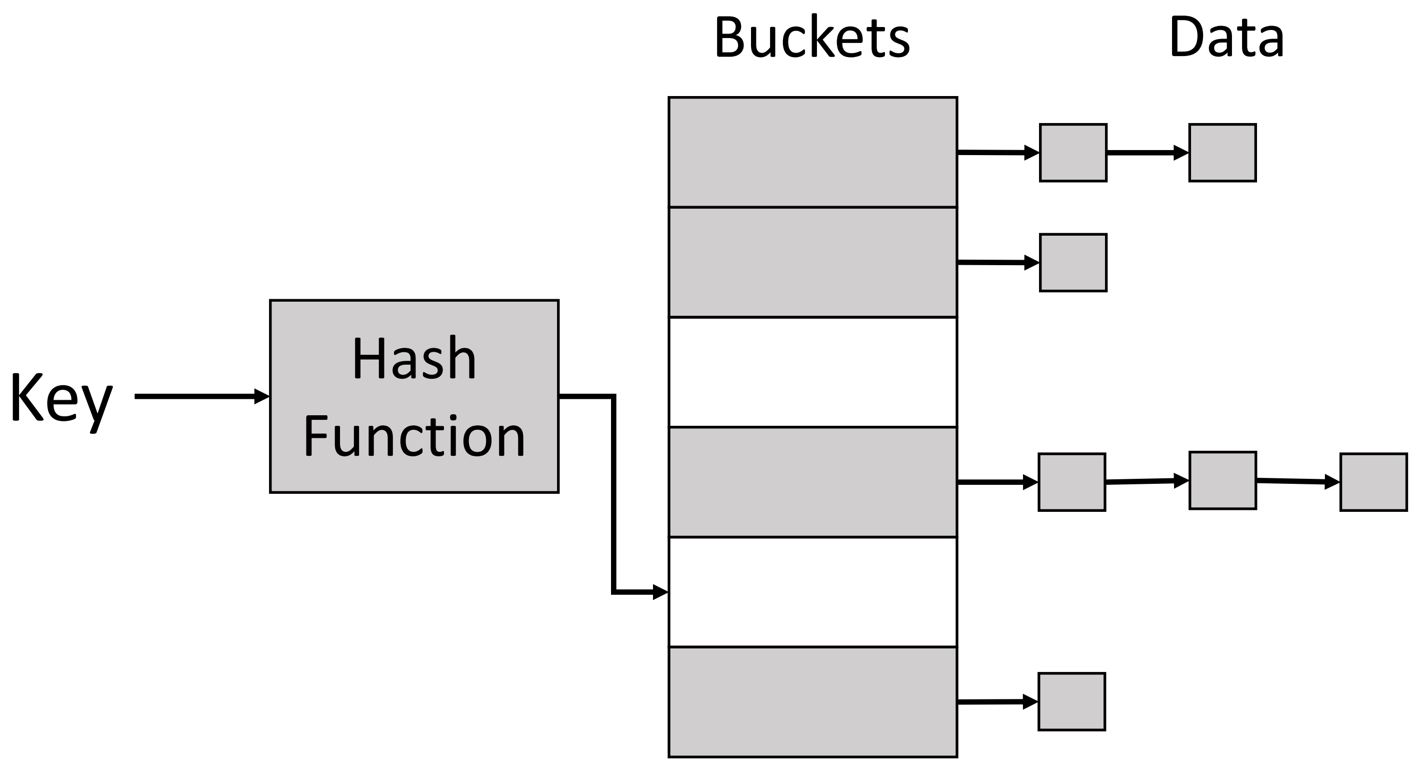
Deep Learning for Science: **AlphaFold 2**



https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

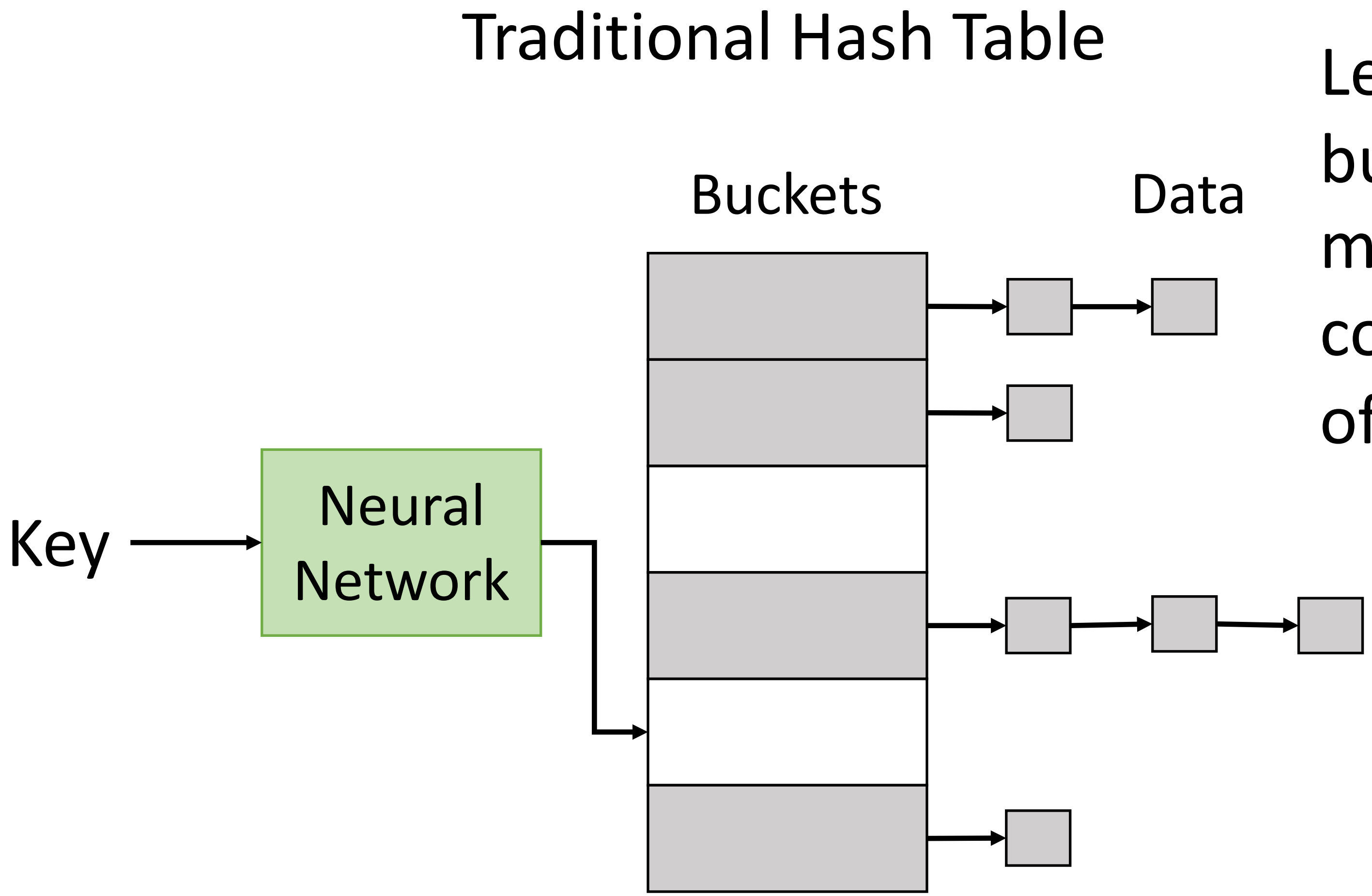
Deep Learning for Computer Science

Traditional Hash Table



Kraska et al, "The Case for Learned Index Structures", SIGMOD 2018

Deep Learning for Computer Science

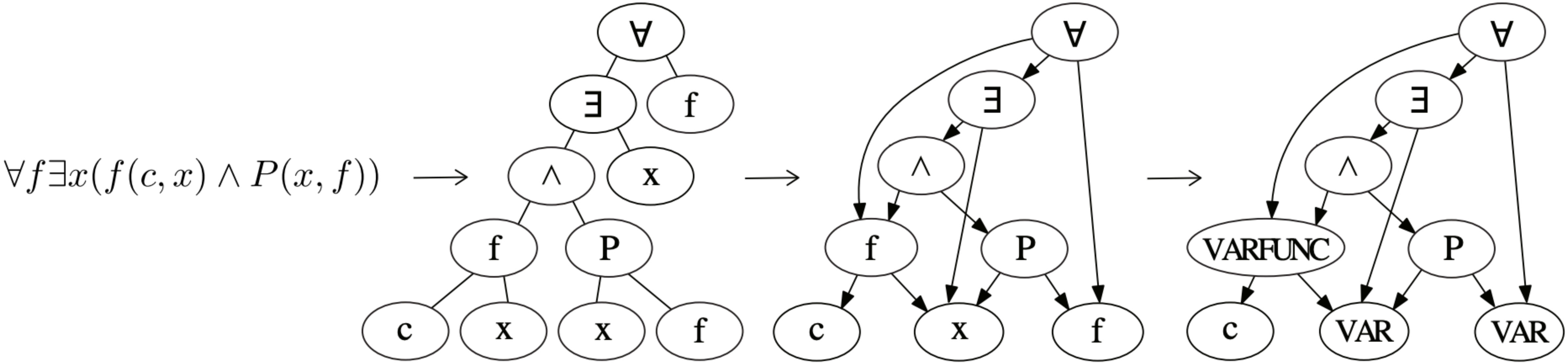


Learn to assign keys to buckets in a way that minimizes hash collisions for the types of data you encounter

Kraska et al, “The Case for Learned Index Structures”, SIGMOD 2018

Deep Learning for Mathematics

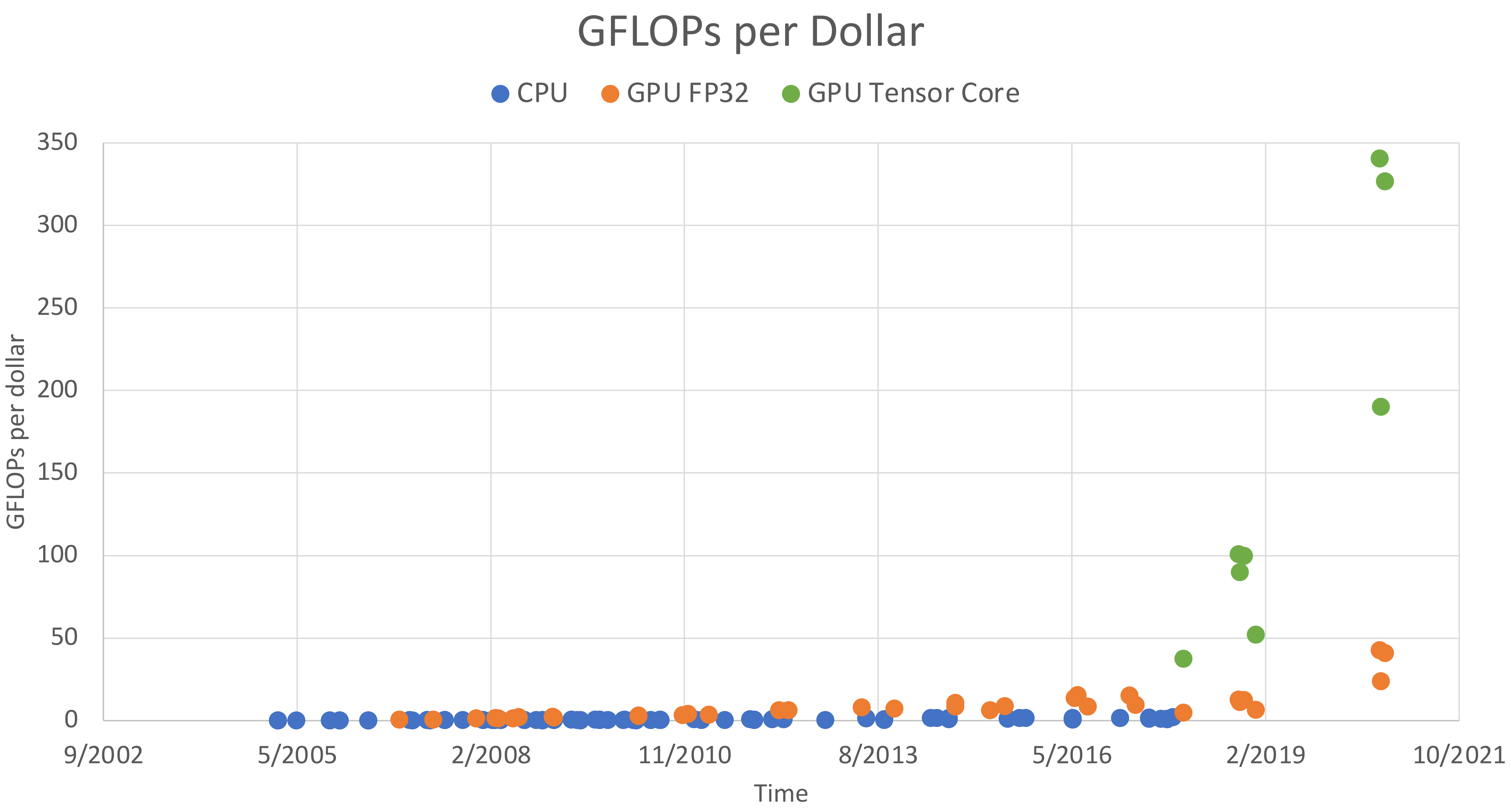
Convert mathematical expressions into graphs, process then with graph neural networks!

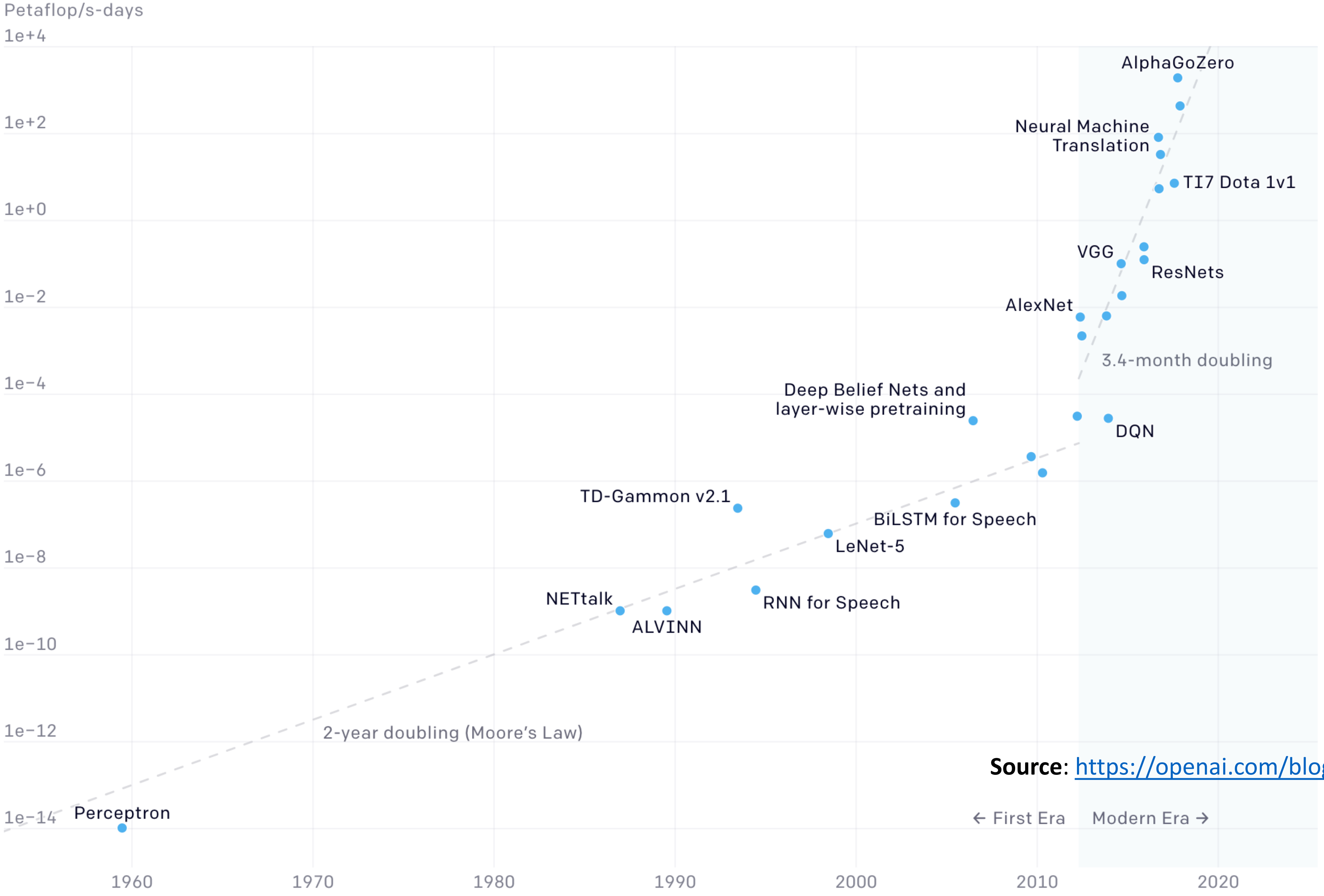


Applications: Theorem proving, symbolic integration

Wang et al, "Premise Selection for Theorem Proving by Deep Graph Embedding", NeurIPS 2017
Kaliszyk et al, "Reinforcement Learning of Theorem Proving", NeurIPS 2018
Lample and Charton, "Deep Learning for Symbolic Mathematics", arXiv 2019

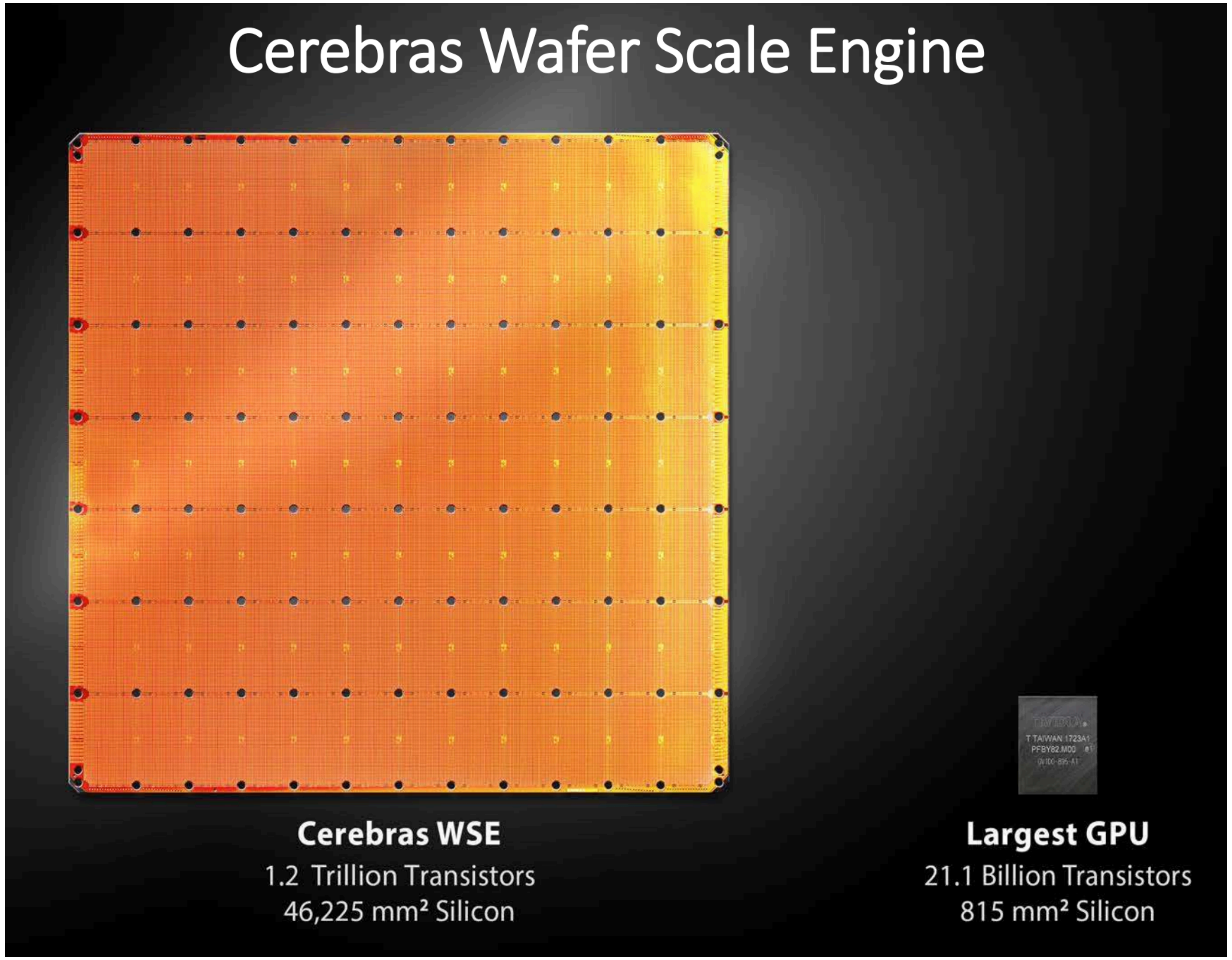
Prediction #3:
Deep Learning will use more data and compute


GFLOPs per Dollar
CPU
GPU FP32
GPU Tensor Core
350
300
250
200
150
100
50
0
GFLOPs per dollar
9/2002
5/2005
2/2008
11/2010
8/2013
5/2016
2/2019
10/2021
Time



Source: https://openai.com/blog/ai-and-compute/

New Hardware for Deep Learning



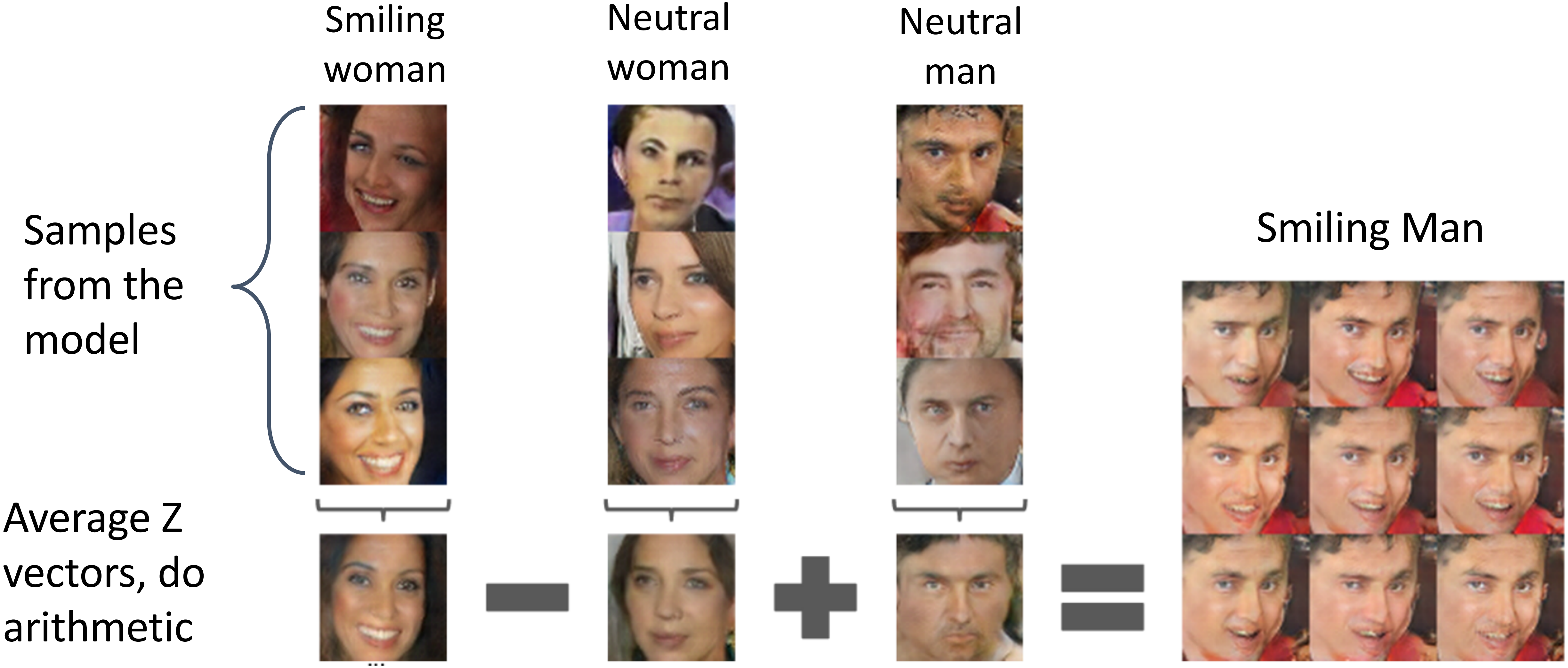
SPECIFICATIONS

Sparse Linear Algebra Compute Cores	**400,000**
On-chip Memory	**18 GB SRAM**
Memory Bandwidth	**9.6 PB/sec**
Core-to-Core Bandwidth	**100 Pb/sec**
Maximum Power Requirement	**20 kW**
System IO	**12x100 GbE**
Cooling	**Air-cooled**
Dimensions	**15 Rack Units (26.25")**

Cerebras Systems, "Wafer-Scale Deep Learning", 2019; https://secureservercdn.net/198.12.145.239/a7b.fcb.myftpupload.com/wp-content/uploads/2019/08/HC31_1.13_Cerebras.SeanLie.v02.pdf

Problem #1:
Models are biased

Recall: Vector Arithmetic with GANs



Radford et al, ICLR 2016

Vector Arithmetic with Word Vectors

Training: Input a large corpus of text, learn to represent each word with a vector

Can used trained vectors to solve analogies:
Man is to **King** as **Woman** is to **x**?

Find nearest neighbor to: **Man – King + Woman**

Mikolov et al, "Distributed Representations of Words and Phrases and their Compositionality", NeurIPS 2013
Mikolov et al, "Linguistic Regularities in Continuous Space Word Representations", NAACL HLT 2013

Gender Bias in Word Vectors

Extreme *she*	**Extreme *he***
1. homemaker	1. maestro
2. nurse	2. skipper
3. receptionist	3. protege
4. librarian	4. philosopher
5. socialite	5. captain
6. hairdresser	6. architect
7. nanny	7. financier
8. bookkeeper	8. warrior
9. stylist	9. broadcaster
10. housekeeper	10. magician

Gender stereotype *she*-*he* analogies

sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	lovely-brilliant

Gender appropriate *she*-*he* analogies

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Bolukbasi et al, "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings", NeurIPS 2016

Economic Bias in Visual Classifiers

Ground-Truth: Soap
Source: UK, $1890/month

DeVries et al, “Does Object Recognition Work for Everyone?”, CVPR Workshops, 2019

Economic Bias in Visual Classifiers

Ground-Truth: Soap
Source: UK, $1890/month

Azure: toilet, design, art, sink
Clarifai: people, faucet, healthcare, lavatory, wash closet
Google: product, liquid, water, fluid, bathroom accessory
Amazon: sink, indoors, bottle, sink faucet
Watson: gas tank, storage tank, toiletry, dispenser, soap dispenser
Tencent: lotion, toiletry, soap dispenser, dispenser, after shave

DeVries et al, “Does Object Recognition Work for Everyone?”, CVPR Workshops, 2019

Economic Bias in Visual Classifiers

Ground-Truth: Soap
Source: Nepal, $288/month

Ground-Truth: Soap
Source: UK, $1890/month

Azure: toilet, design, art, sink
Clarifai: people, faucet, healthcare, lavatory, wash closet
Google: product, liquid, water, fluid, bathroom accessory
Amazon: sink, indoors, bottle, sink faucet
Watson: gas tank, storage tank, toiletry, dispenser, soap dispenser
Tencent: lotion, toiletry, soap dispenser, dispenser, after shave

DeVries et al, "Does Object Recognition Work for Everyone?", CVPR Workshops, 2019

Economic Bias in Visual Classifiers

Ground-Truth: Soap
Source: Nepal, $288/month

Azure: food, cheese, bread, cake, sandwich
Clarifai: food, wood, cooking, delicious, healthy
Google: food, dish, cuisine, comfort food, spam
Amazon: food, confectionary, sweets, burger
Watson: food, food product, turmeric, seasoning
Tencent: food, dish, matter, fast food, nutriment

Ground-Truth: Soap
Source: UK, $1890/month

Azure: toilet, design, art, sink
Clarifai: people, faucet, healthcare, lavatory, wash closet
Google: product, liquid, water, fluid, bathroom accessory
Amazon: sink, indoors, bottle, sink faucet
Watson: gas tank, storage tank, toiletry, dispenser, soap dispenser
Tencent: lotion, toiletry, soap dispenser, dispenser, after shave

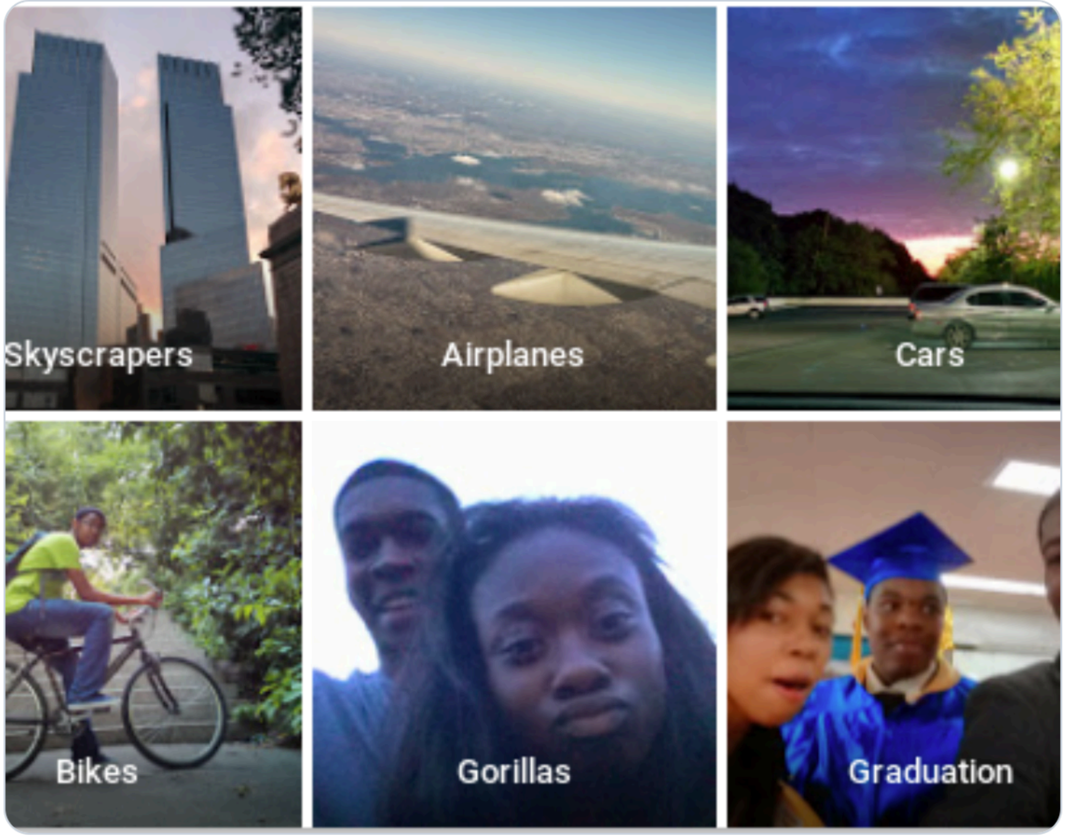
Commercial object recognition systems work best for objects found in high-income western houseolds

DeVries et al, "Does Object Recognition Work for Everyone?", CVPR Workshops, 2019

Racial Bias in Visual Classifiers

https://jacky.wtf
@jackyalcine

Google Photos, y'all fucked up. My friend's not a gorilla.



Source: https://twitter.com/jackyalcine/status/615329515909156865 (2015)

Racial Bias in Visual Classifiers

Commercial gender classifiers fail much more often for women with dark skin

Classifier	Metric	All	F	M	Darker	Lighter	DF	DM	LF	LM
MSFT	PPV(%)	93.7	89.3	97.4	87.1	99.3	79.2	94.0	98.3	**100**
	Error Rate(%)	6.3	10.7	2.6	12.9	0.7	**20.8**	6.0	1.7	0.0
	TPR (%)	93.7	96.5	91.7	87.1	99.3	92.1	83.7	**100**	98.7
	FPR (%)	6.3	8.3	3.5	12.9	0.7	**16.3**	7.9	1.3	0.0
Face++	PPV(%)	90.0	78.7	99.3	83.5	95.3	65.5	**99.3**	94.0	99.2
	Error Rate(%)	10.0	21.3	0.7	16.5	4.7	**34.5**	0.7	6.0	0.8
	TPR (%)	90.0	98.9	85.1	83.5	95.3	98.8	76.6	**98.9**	92.9
	FPR (%)	10.0	14.9	1.1	16.5	4.7	**23.4**	1.2	7.1	1.1
IBM	PPV(%)	87.9	79.7	94.4	77.6	96.8	65.3	88.0	92.9	**99.7**
	Error Rate(%)	12.1	20.3	5.6	22.4	3.2	**34.7**	12.0	7.1	0.3
	TPR (%)	87.9	92.1	85.2	77.6	96.8	82.3	74.8	**99.6**	94.8
	FPR (%)	12.1	14.8	7.9	22.4	3.2	**25.2**	17.7	5.20	0.4

Buolamwini and Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification", FAT* 2018

Making ML Work for Everyone

Wang et al, “Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations”, ICCV 2019

Hutchinson and Mitchell, ”50 Years of Test (Un) fairness: Lessons for Machine Learning”, CFAT 2019

Mitchell et al, “Model Cards for Model Reporting”, CFAT 2019

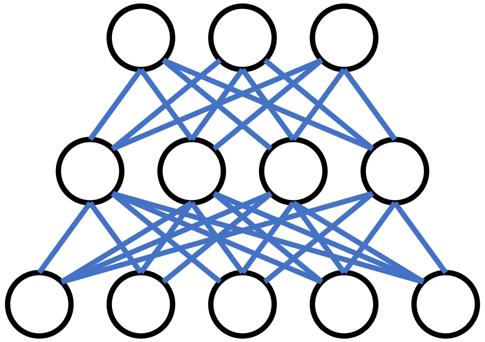
Zhang et al, “Mitigating unwanted biases with adversarial learning”, AAAI 2018

Buolamwini and Gebru, “Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification”, CFAT 2018

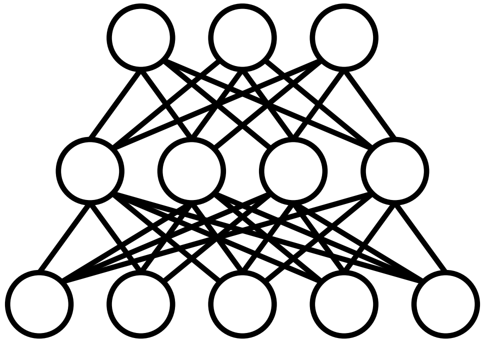
Problem #2:
Need new theory?

Empirical Mystery: Good Subnetworks

Step 1: Randomly initialize a network



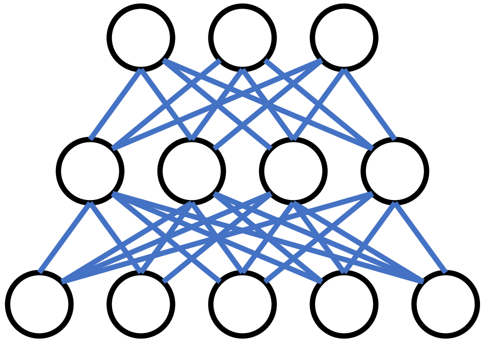
Step 2: Train on your favorite dataset



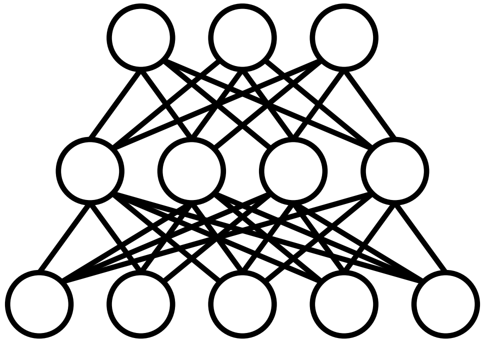
Han et al, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", ICLR 2016

Empirical Mystery: Good Subnetworks

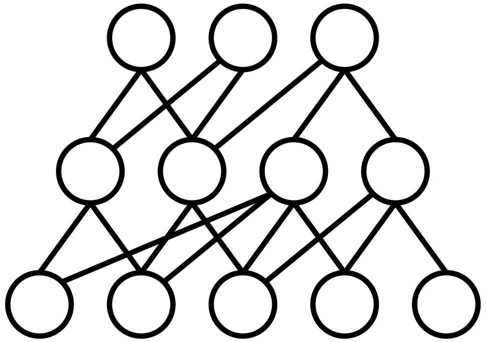
Step 1: Randomly initialize a network



Step 2: Train on your favorite dataset



Step 3: Remove weights of small magnitude

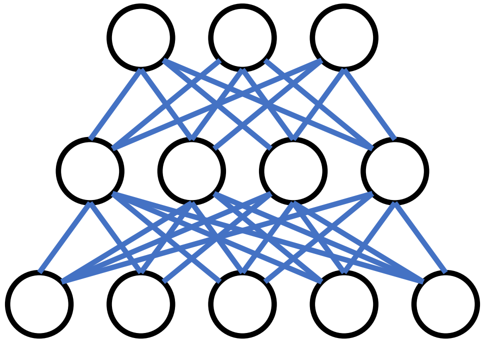


Pruned network works about the same as full network in (2)!

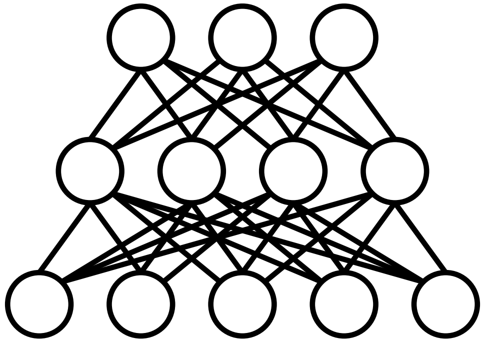
Han et al, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", ICLR 2016

Empirical Mystery: Good Subnetworks

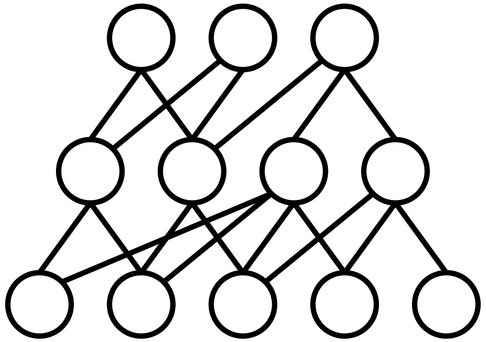
Step 1: Randomly initialize a network



Step 2: Train on your favorite dataset

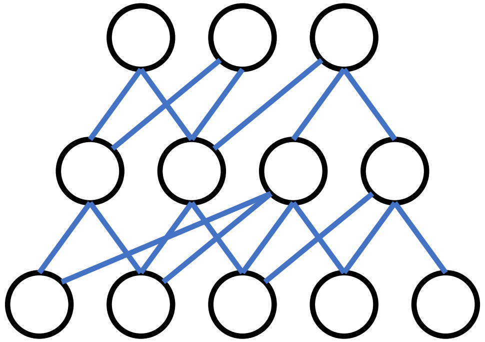


Step 3: Remove weights of small magnitude



Pruned network works about the same as full network in (2)!

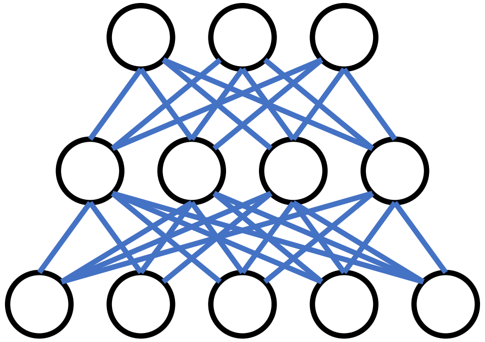
Step 4: Return pruned network weights to initial values



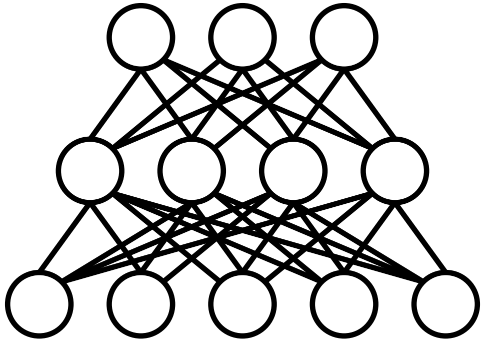
Frankle and Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks", ICLR 2019

Empirical Mystery: Good Subnetworks

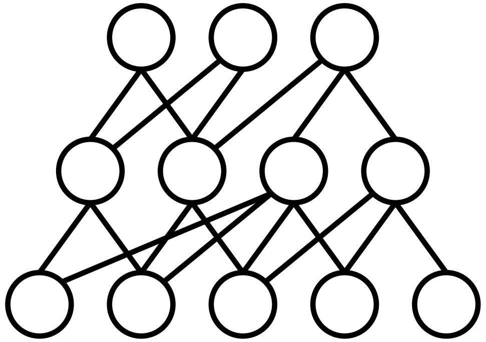
Step 1: Randomly initialize a network



Step 2: Train on your favorite dataset

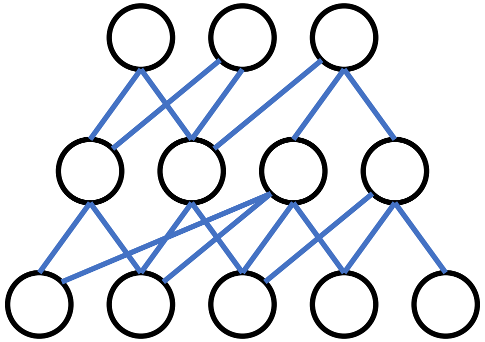


Step 3: Remove weights of small magnitude

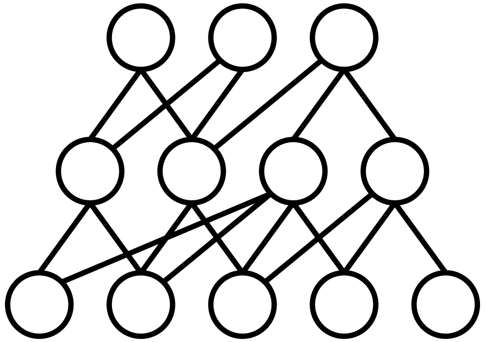


Pruned network works about the same as full network in (2)!

Step 4: Return pruned network weights to initial values



Step 5: Train pruned network; it works almost as good as (2)!

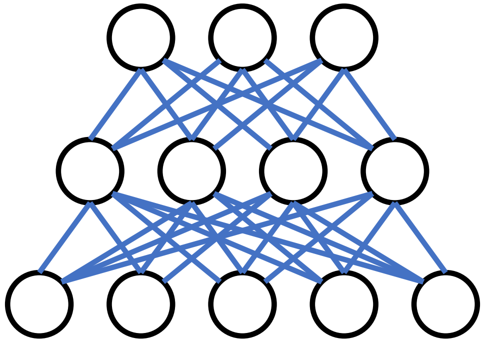


Lottery Ticket Hypothesis: Within a random deep network is a good subnet that won the “initialization lottery”

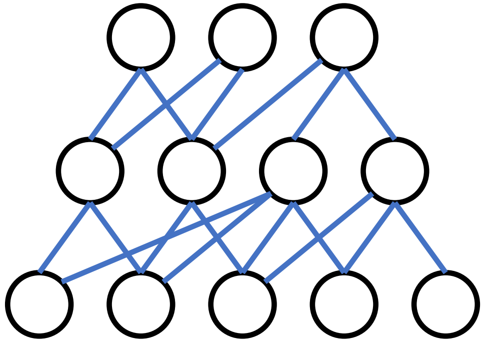
Frankle and Carbin, “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks”, ICLR 2019

Empirical Mystery: Good Subnetworks

Step 1: Randomly initialize a network



Step 2: Find an untrained subnet that works for classification!

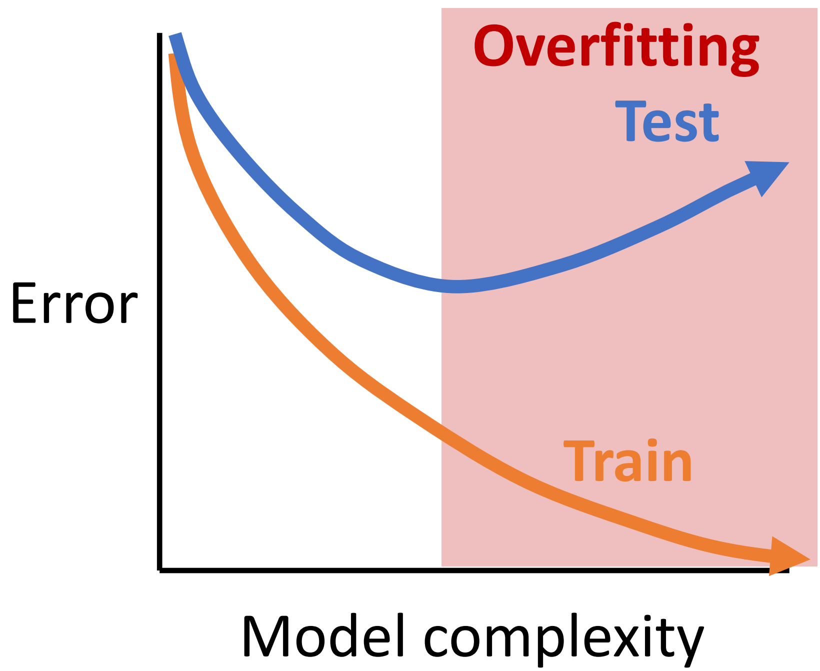


I think we are missing something about how to train and initialize deep nets, what training actually does

Ramanujan et al, “What's Hidden in a Randomly Weighted Neural Network?”, arXiv 2019

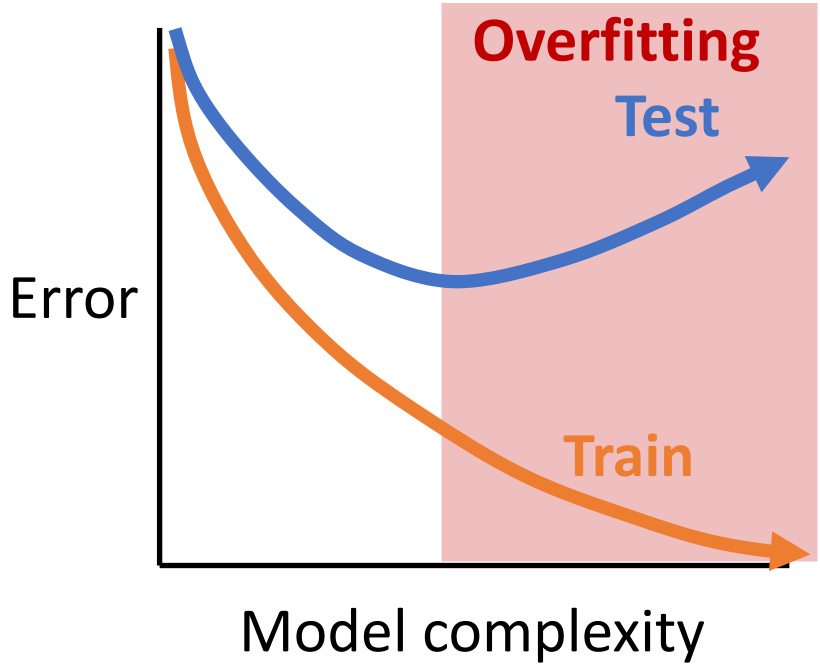
Empirical Mystery: Generalization

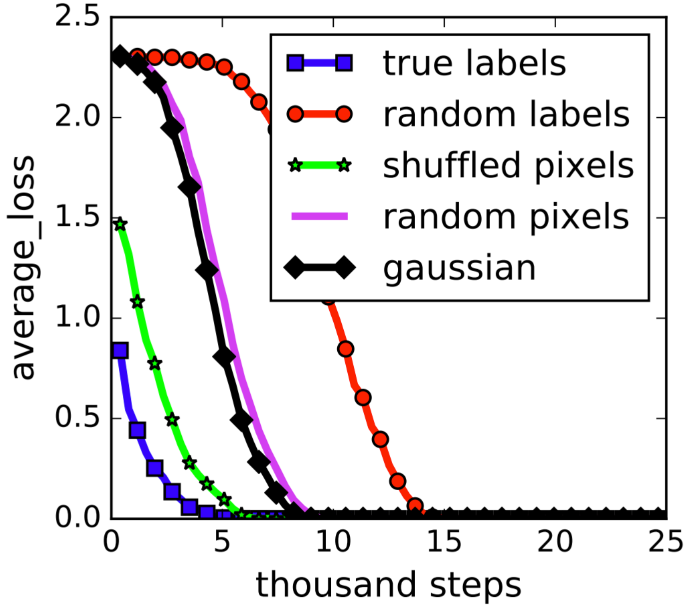
What we expect from classical statistical learning theory:



Empirical Mystery: Generalization

What we expect from classical statistical learning theory:

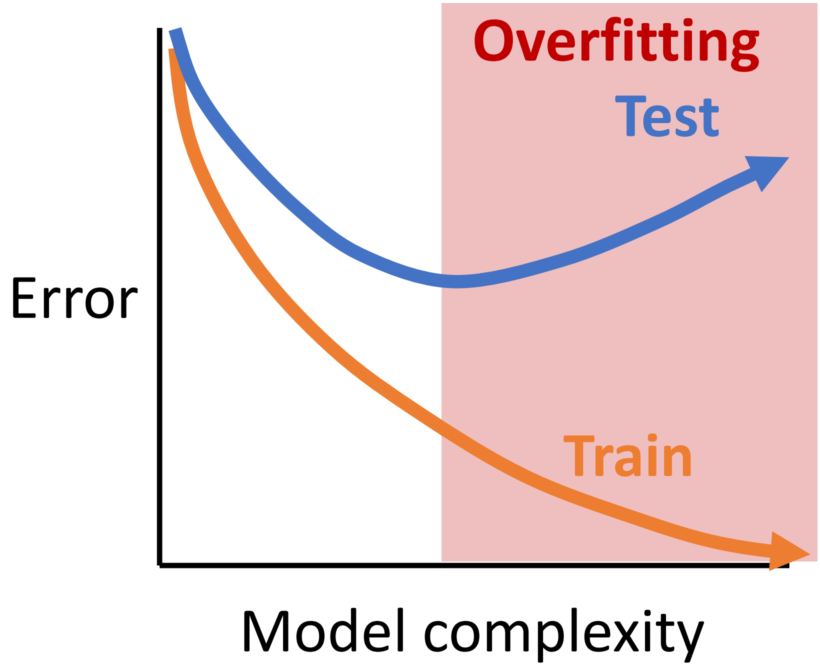


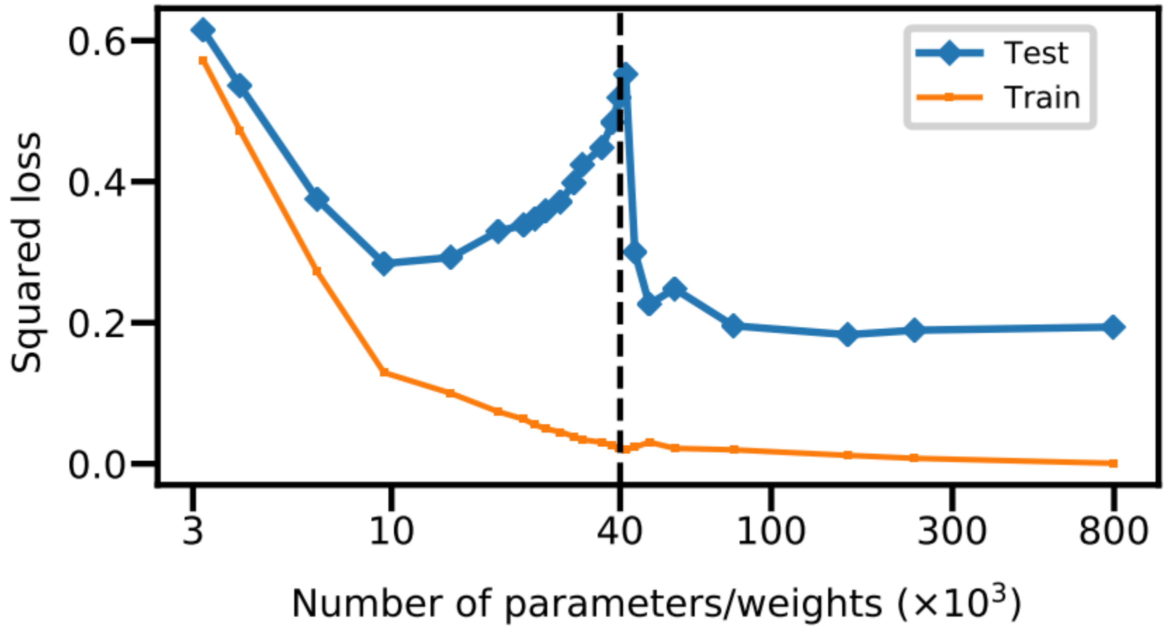


Deep networks can achieve 0 training loss on CIFAR with random labels. When we train the same model on real data, why doesn't it overfit?

Empirical Mystery: Generalization

What we expect from classical statistical learning theory:



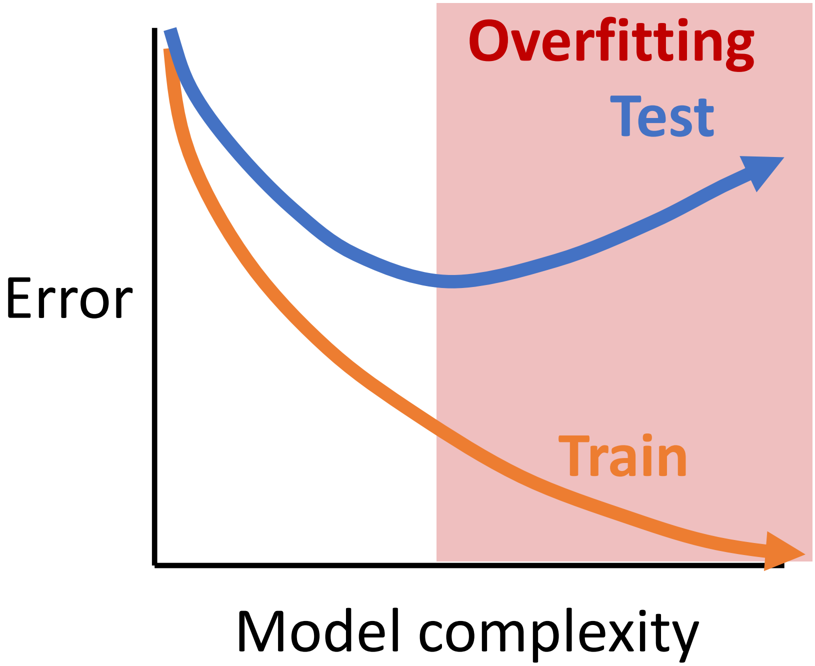


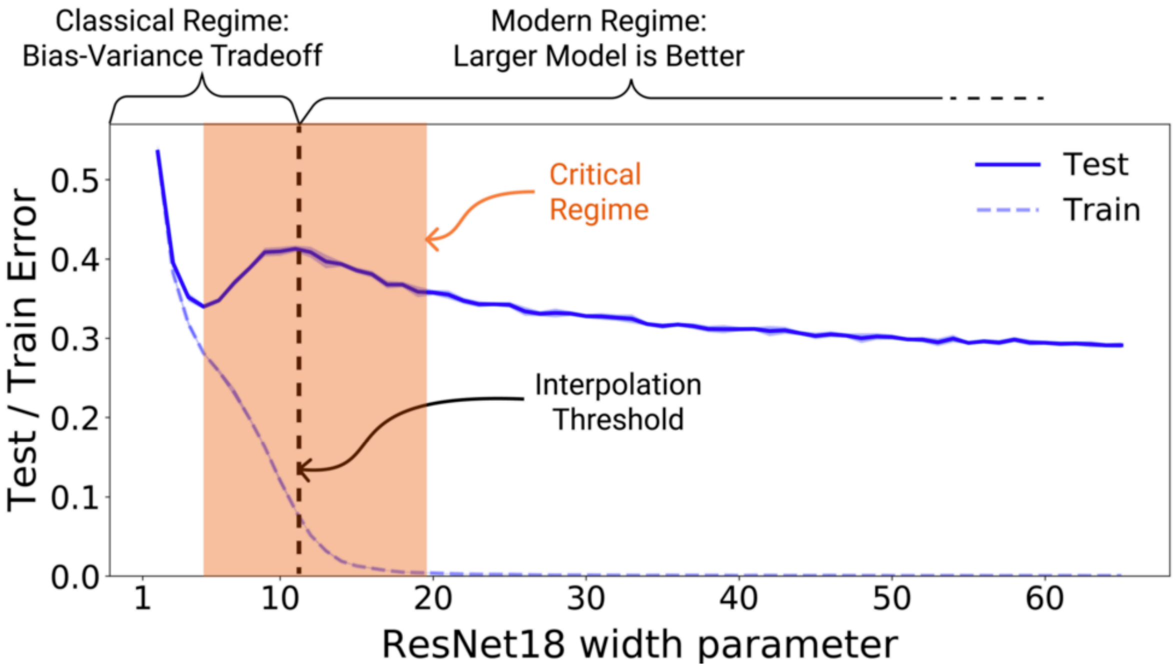
"Double Descent" for fully-connected models on CIFAR does not match theory!

Belkin et al, "Reconciling modern machine learning practice and the bias-variance trade-of", PNAS 2019

Empirical Mystery: Generalization

What we expect from classical statistical learning theory:





Similar result for ResNets on CIFAR
(when training with label noise)

Nakkiran et al, "Deep Double Descent: Where Bigger Models and More Data Hurt", arXiv 2019

Problem #3:
Deep Learning needs a lot of labeled training data

New Datasets for Low-Shot Learning

MNIST Dataset
10 classes: Digits 0 to 9
28x28 grayscale images
6k images per class (5k train, 1k test)

Omniglot Dataset
1623 classes: Letters from 50 alphabets
20 images per class

Lake et al, "Human-level concept learning through probabilistic program induction," Science 2015

New Datasets for Low-Shot Learning

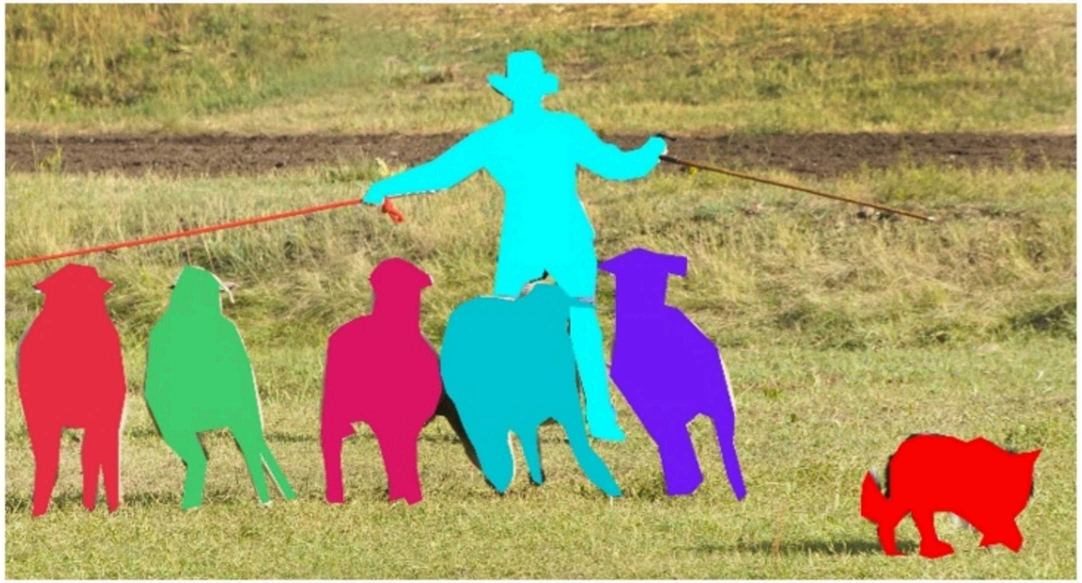
KMNIST Dataset
10 classes: 3832 Kanji characters
64x64 grayscale images
1 to 1766 images per class

Omniglot Dataset
1623 classes: Letters from 50 alphabets
20 images per class

Lake et al, "Human-level concept learning through probabilistic program induction," Science 2015

New Datasets for Low-Shot Learning

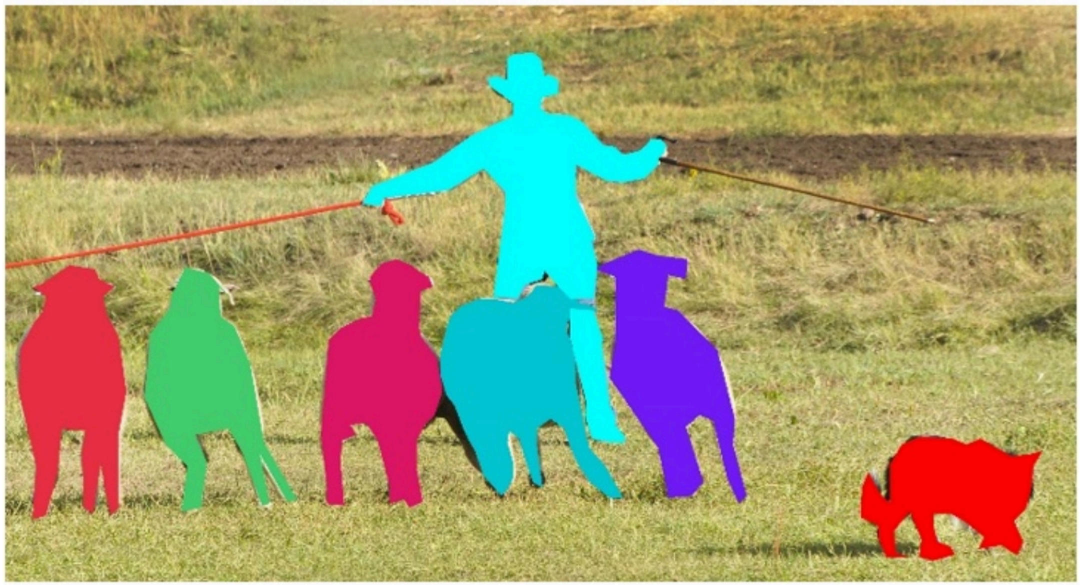
COCO Dataset
118k images
80 categories
1.2M object instances



Lin et al, "Microsoft COCO: Common Objects in Context", ECCV 2014

New Datasets for Low-Shot Learning

COCO Dataset
118k images
80 categories
1.2M object instances



Lin et al, "Microsoft COCO: Common Objects in Context", ECCV 2014

LVIS Dataset
160k images
>1000 categories
~2M object instances

Gupta et al, "LVIS: A Dataset for Large Vocabulary Instance Segmentation", CVPR 2019

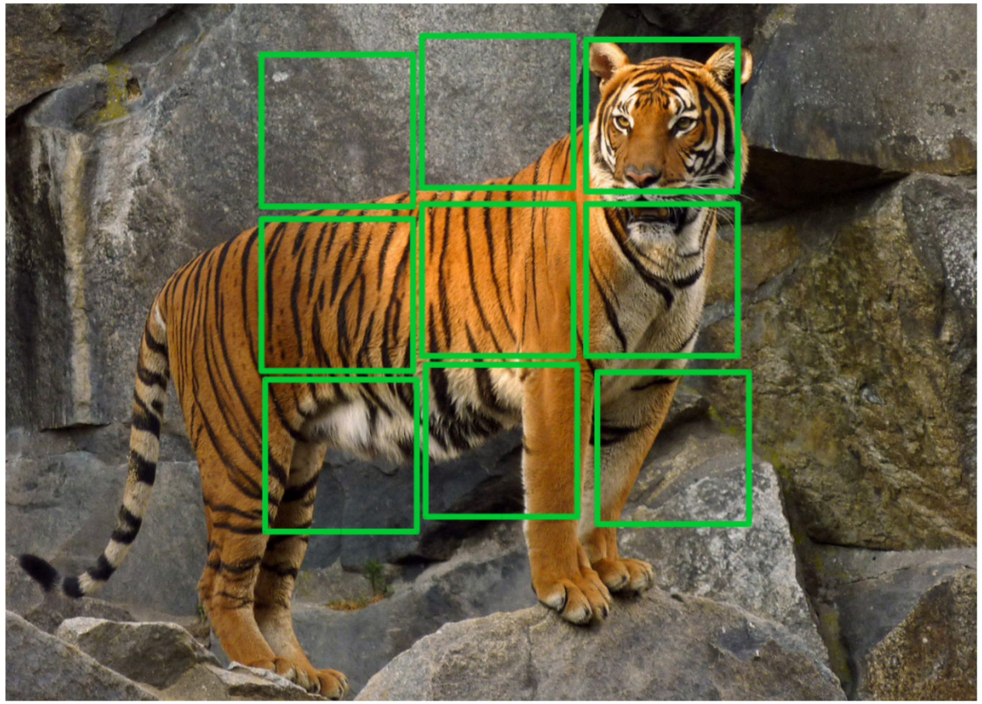
Using Unlabeled Data: Self-Supervised Learning

Step 1: Train a CNN on some ”pretext task” that does not require labeled data

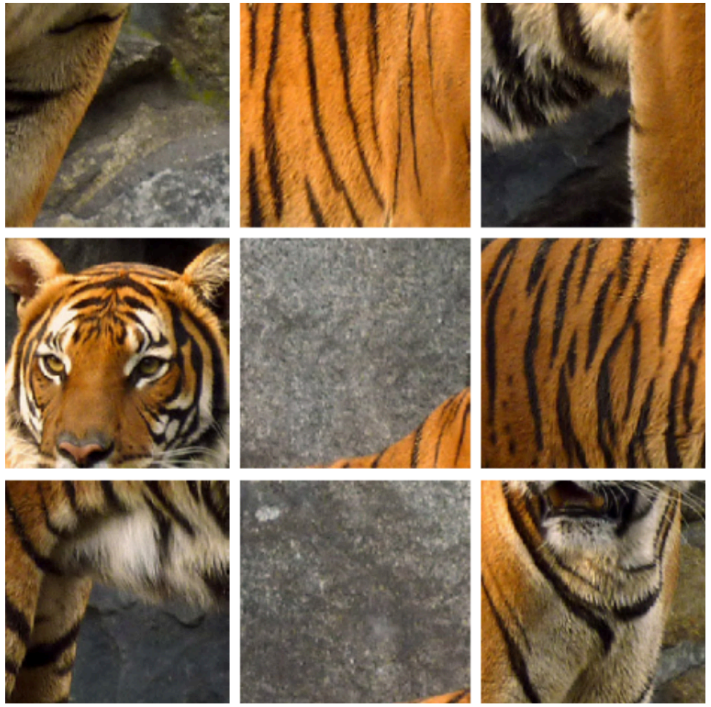
Step 2: Fine-tune CNN on target task (hopefully using not much labeled data)

Self-Supervised Learning: Jigsaw Puzzles

Source Image



Shuffled patches



Network unscrambles

Noroozi and Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles", ECCV 2016

Self-Supervised Learning: Colorization

Input: Grayscale image

Output: Color Image

Zhang et al, "Colorful Image Colorization", ECCV 2016
Zhang et al, "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction", ECCV 2016

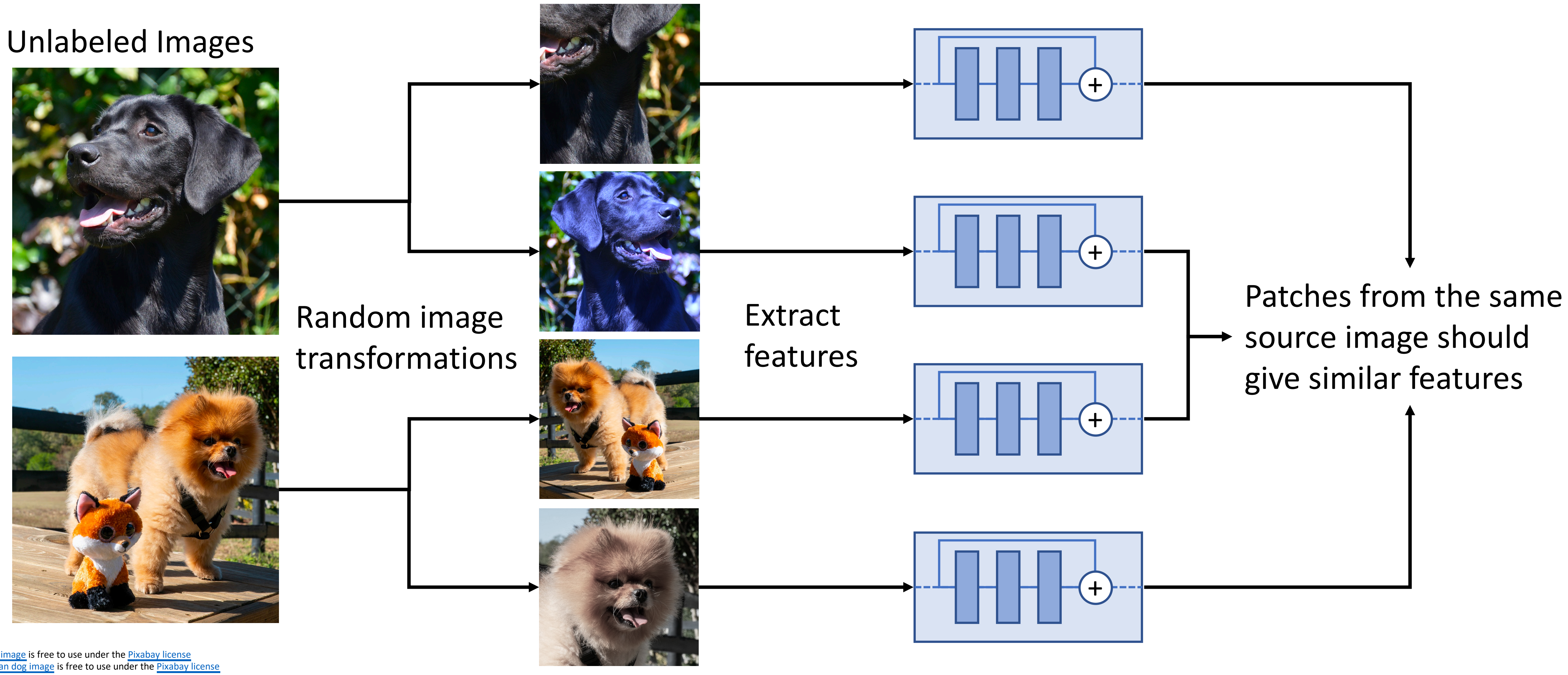
Self-Supervised Learning: Inpainting

Input: Image with a hole

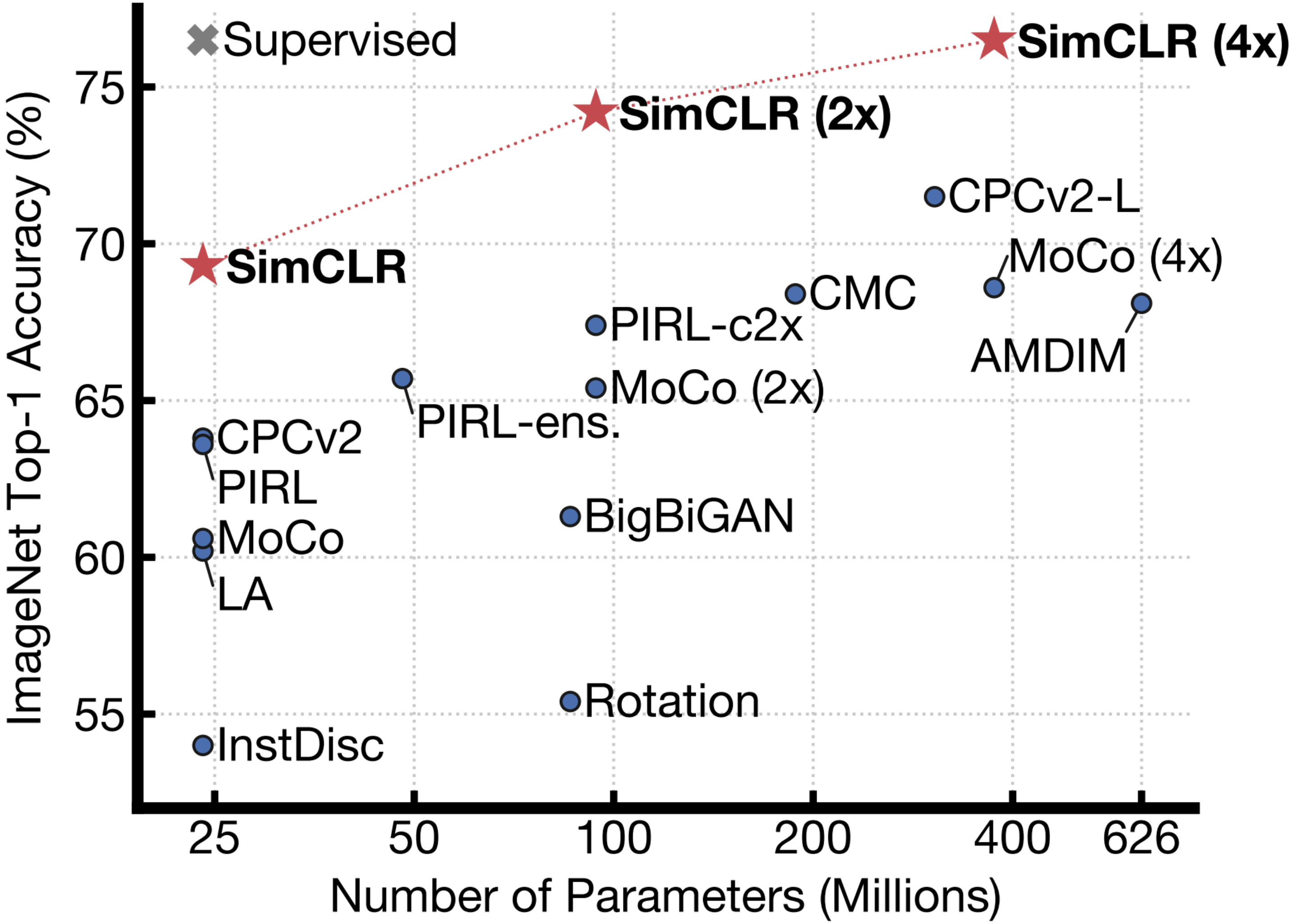
Output: Hole filled in

Pathak et al, "Context Encoders: Feature Learning by Inpainting", CVPR 2016

Self-Supervised Learning: Contrastive Learning



Self-Supervised Learning: Contrastive Learning



- Misra and van der Maaten, "Self-supervised learning of pretext-invariant representations" CVPR 2020 (PIRL)
- He et al, "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020 (MoCo)
- Chen et al, "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020 (SimCLR)
- Chen et al, "Improved Baselines with Momentum Contrastive Learning", arXiv 2020 (MoCo-v2)
- Chen et al, "Big Self-Supervised Models are Strong Semi-Supervised Learners", arXiv 2020 (SimCLR-v2)
- Caron et al, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", arXiv 2020
- Chen and He, "Exploring Simple Siamese Representation Learning", arXiv 2020

Problem #4: Deep Learning doesn't "Understand" the world

Language Models lack common sense

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be

Examples generated using https://talktotransformer.com/

Language Models lack common sense

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

Examples generated using https://talktotransformer.com/

Language Models lack common sense

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

I see a black dog and a brown horse. The bigger animal's color is

Examples generated using https://talktotransformer.com/

Language Models lack common sense

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

I see a black dog and a brown horse. The bigger animal's color is *black, and the smaller is brown.*

Examples generated using https://talktotransformer.com/

Language Models lack common sense

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

I see a black dog and a brown horse. The bigger animal's color is *black, and the smaller is brown.*

One of my parents is a doctor and the other is a professor. My father is a professor. My mother is

Examples generated using https://talktotransformer.com/

Language Models lack common sense

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

I see a black dog and a brown horse. The bigger animal's color is *black, and the smaller is brown.*

One of my parents is a doctor and the other is a professor. My father is a professor. My mother is *a social worker. They're super smart people.*

Examples generated using https://talktotransformer.com/

Language Models lack common sense

What about GPT-3?

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

I see a black dog and a brown horse. The bigger animal's color is *black, and the smaller is brown.*

One of my parents is a doctor and the other is a professor. My father is a professor. My mother is *a social worker. They're super smart people.*

Examples generated using https://talktotransformer.com/

Language Models lack common sense

What about GPT-3?

Bold = prompt written by me
Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be *35. That was only a few years ago. Most things in life just continue to improve.*

I see a black dog and a brown horse. The bigger animal's color is *black, and the smaller is brown.*

One of my parents is a doctor and the other is a professor. My father is a professor. My mother is *a social worker. They're super smart people.*

Examples generated using https://talktotransformer.com/

Language Models lack common sense

Bold = prompt written by me

Italic = completion written by GPT-2

What about GPT-3?

THE VERGE | TECH | REVIEWS | SCIENCE | CREATORS | ENTERTAINMENT | VIDEO | MORE

MICROSOFT | TECH | ARTIFICIAL INTELLIGENCE

8

Microsoft exclusively licenses OpenAI's groundbreaking GPT-3 text generation model

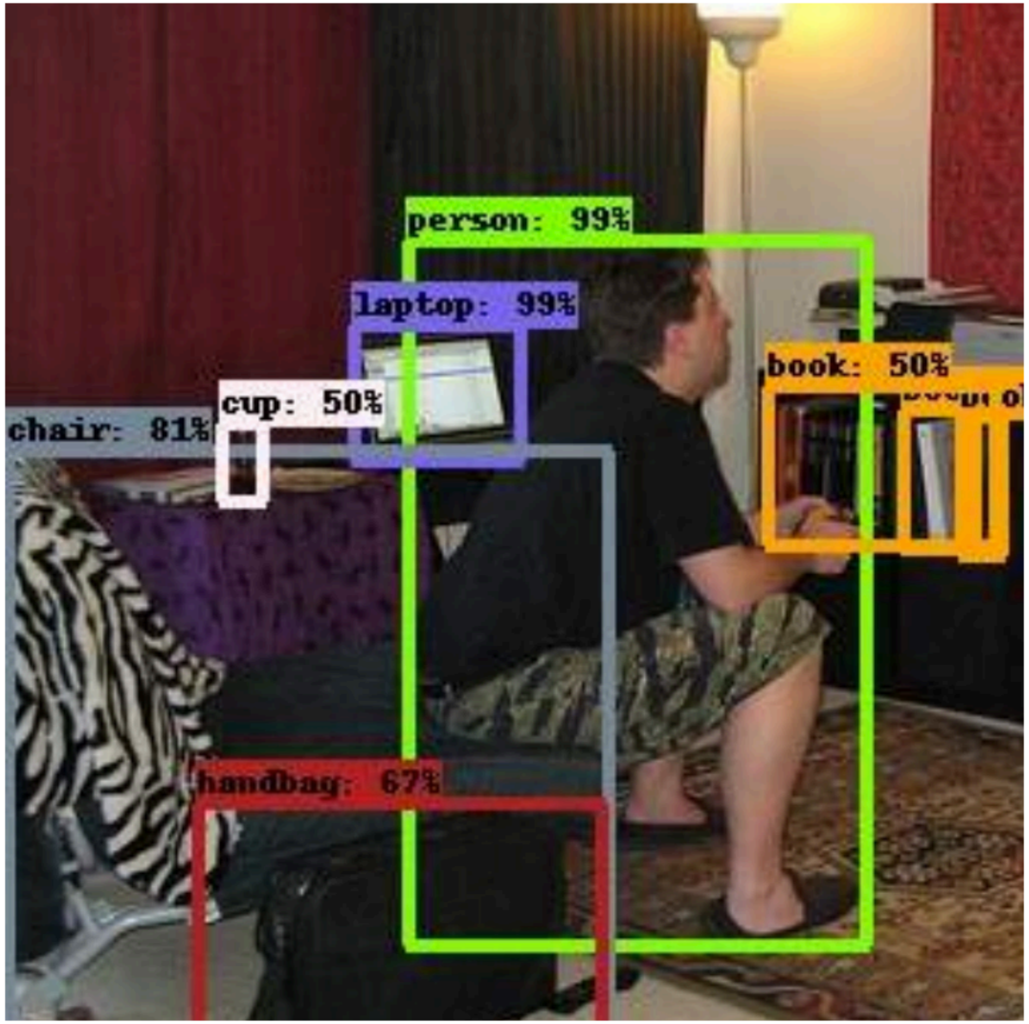
Microsoft will get to use the underlying technology of the AI model in its products

By Nick Statt | @nickstatt | Sep 22, 2020, 4:08pm EDT

Exclusive Microsoft license means I can't play with it!

"The Elephant in the Room"

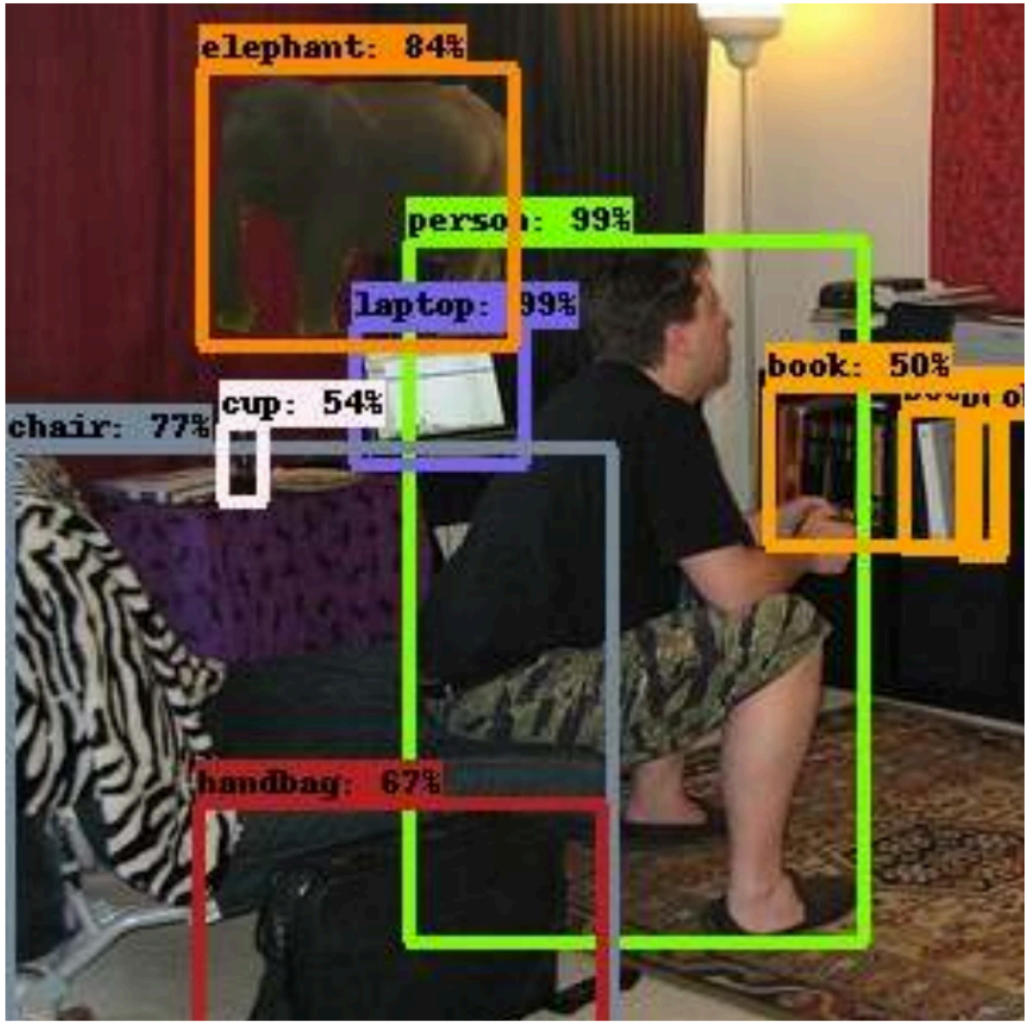
Modern object detectors seem to work well!



Rosenfeld et al, "The Elephant in the Room", arXiv 2018

"The Elephant in the Room"

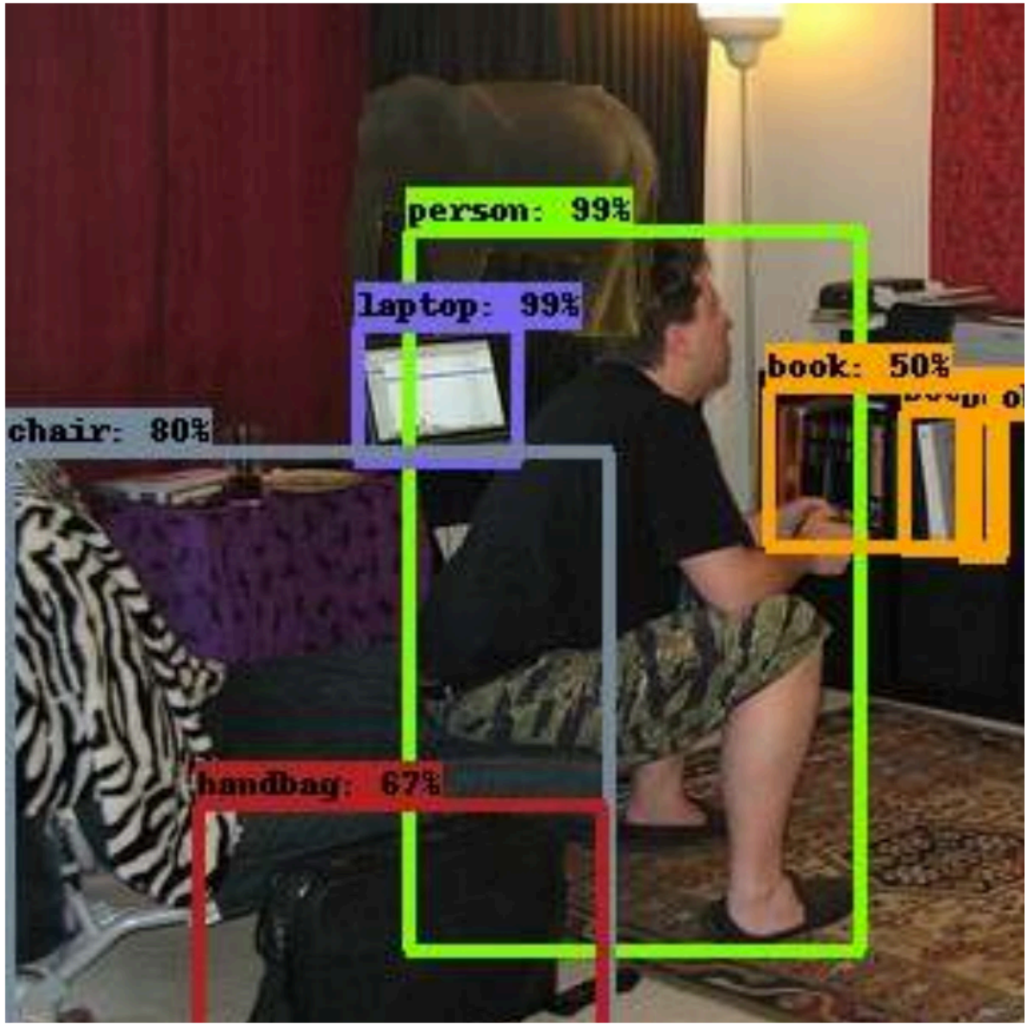
We add an out-of-context elephant to the scene;
Sometimes it is detected
Sometimes it messes up other objects: cup



Rosenfeld et al, "The Elephant in the Room", arXiv 2018

“The Elephant in the Room”

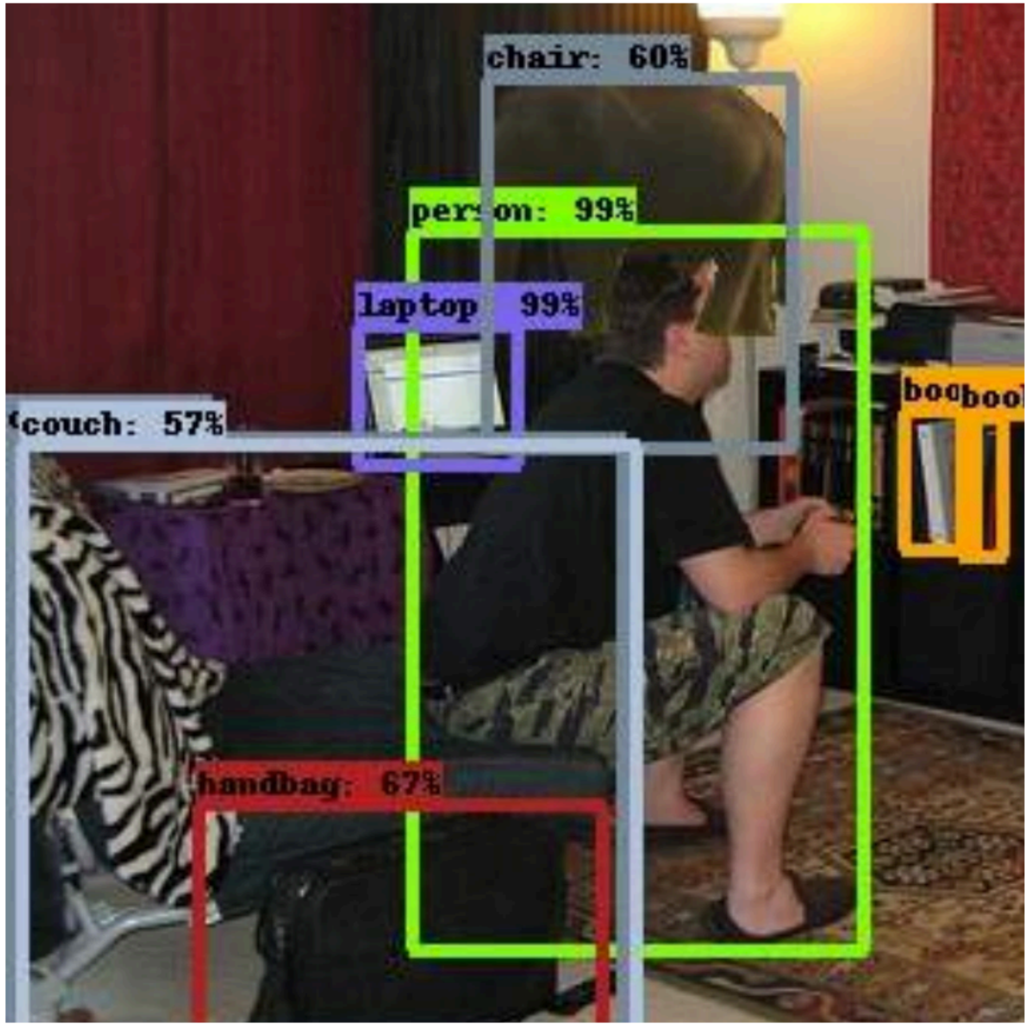
We add an out-of-context elephant to the scene; Sometimes it is missed



Rosenfeld et al, “The Elephant in the Room”, arXiv 2018

“The Elephant in the Room”

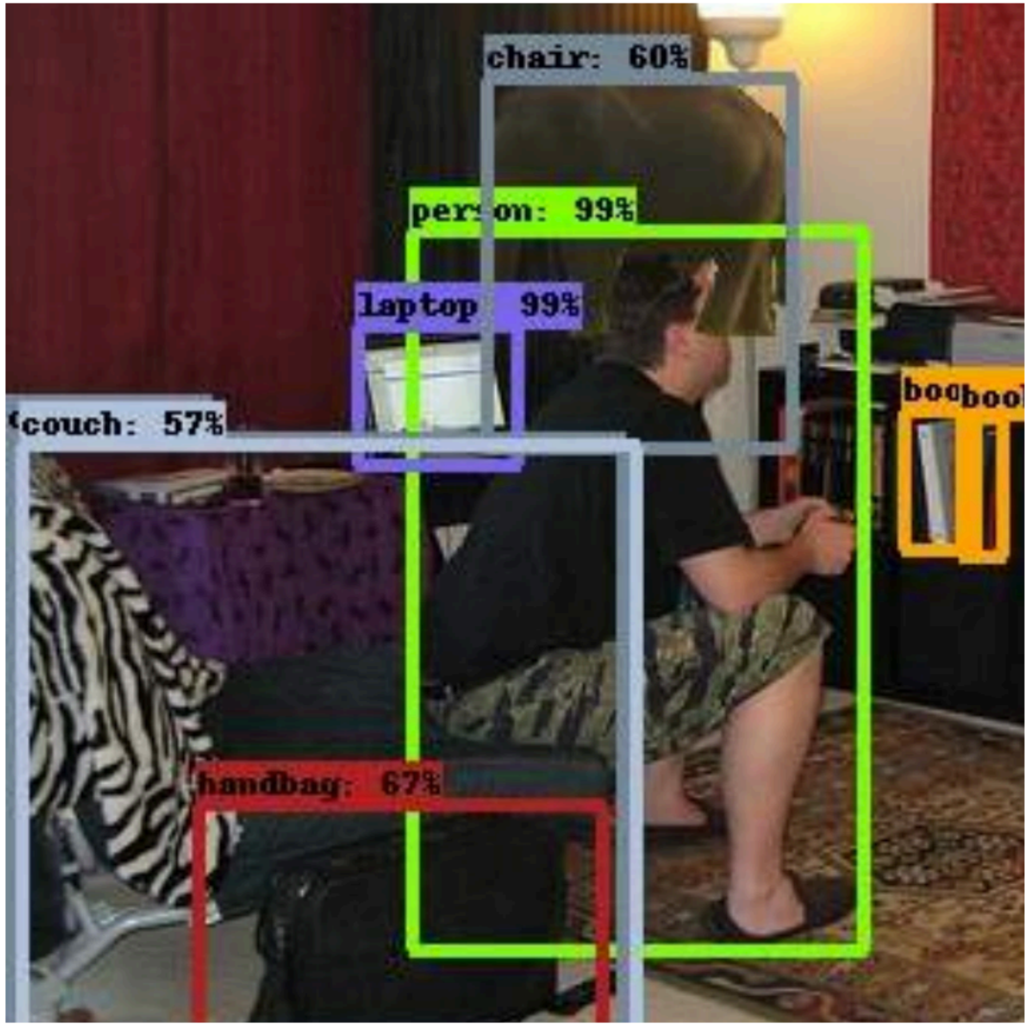
We add an out-of-context elephant to the scene; Sometimes it is assigned the wrong label
Or mess up other objects! (cup, couch)



Rosenfeld et al, “The Elephant in the Room”, arXiv 2018

"The Elephant in the Room"

We add an out-of-context elephant to the scene;
Sometimes it is assigned the wrong label
Or mess up other objects! (cup, couch)



Conclusion: CNNs "see" in a very different way from us. They can fail catastrophically on images even slightly different from those seen during training. How can we fix this?

Rosenfeld et al, "The Elephant in the Room", arXiv 2018

Deep Learning: Problems and Predictions

Predictions:
New deep learning models
New applications
More compute, new hardware

Problems:
Models are biased
Need new theory
Using less data
Understanding the world

Deep Learning: Problems and Predictions

Predictions:
New deep learning models
New applications
More compute, new hardware

Problems:
Models are biased
Need new theory
Using less data
Understanding the world

Now is a great time to be working in computer vision and machine learning!

Thanks GSIs!

Yunseok Jang

Mohamed El Banani

Danish Syed

Yashmeet Gambhir

Thank You!