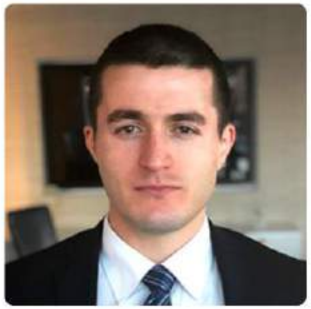
Lex Fridman

Lecture 1:

Deep Learning

6.S094: Deep Learning for Self-Driving Cars



Lex Fridman

Instructor

Jack Terwilliger

TA

Julia Kindelsberger

TA

Dan Brown

TA

Michael Glazer

TA

Li Ding

TA

Spencer Dodd

TA

Benedikt Jenik

TA

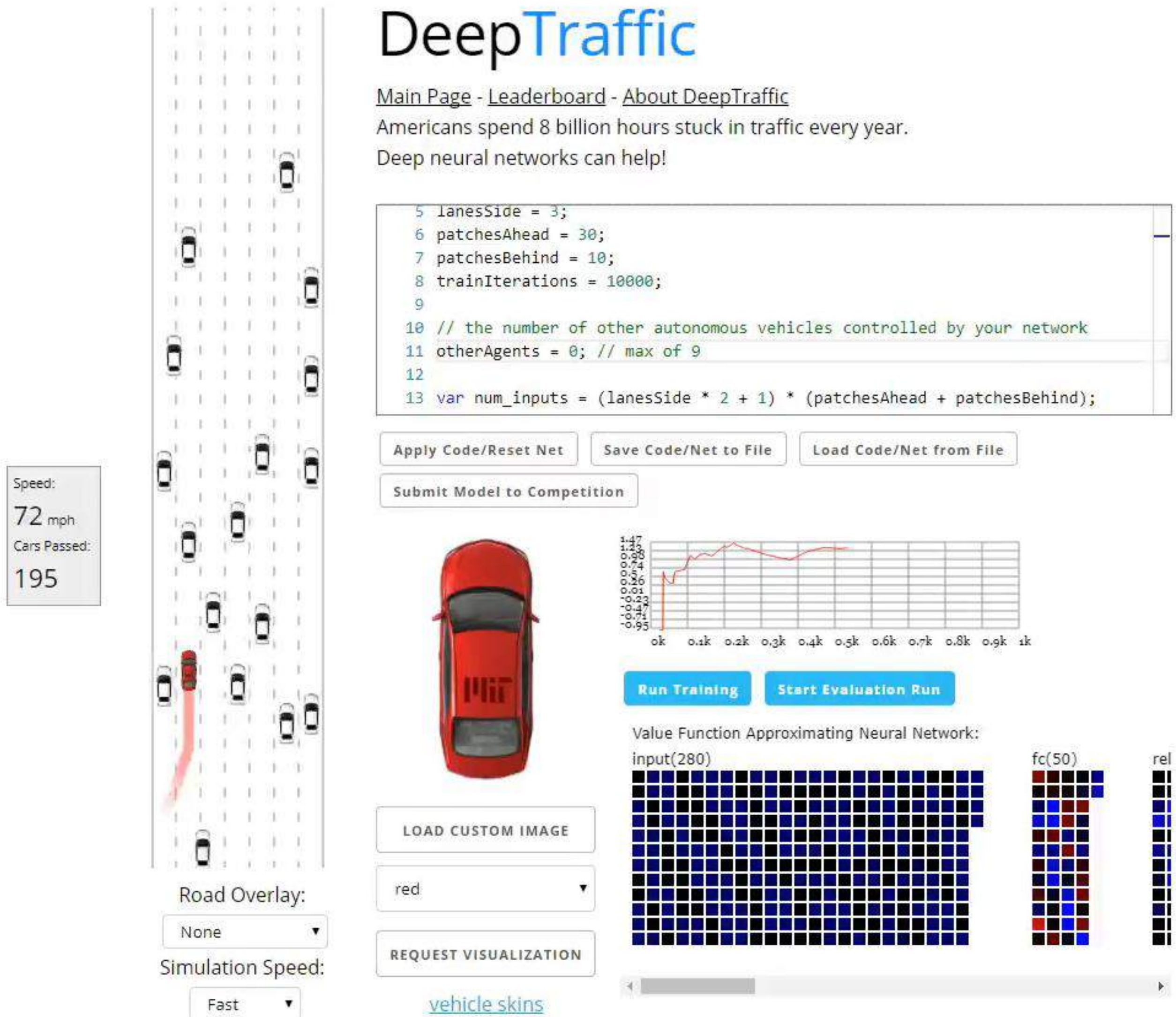
6.S094: Deep Learning for Self-Driving Cars

2018

2017

- **Website:** selfdrivingcars.mit.edu
- **Email:** deepcars@mit.edu
- **Slack:** deep-mit.slack.com
- **For registered MIT students:**
  - Create an account on the website.
  - DeepTraffic 2.0 neural network competition entry that achieves 65mph by 11:59pm, Fri, Jan 19
- **Competitions**
  - DeepTraffic (Deep RL in Browser)
  - SegFuse (Deep Learning in Video)
  - DeepCrash (Deep RL + Computer Vision)
- **Guest Speakers** (see schedule)
- **2018 Shirts** (free in-person)

DeepTraffic: Deep Reinforcement Learning



SegFuse: Dynamic Driving Scene Segmentation

DeepCrash: Deep RL for High-Speed Crash Avoidance

DeepTesla: End-to-End Driving

Lectures and Guest Talks

Lecture Mon, Jan 8, 7pm Room 54-100

Deep Learning: Overview and Recent Advances

[Slides] - [Lecture Video] *(Available Soon)*

Lecture Tue, Jan 9, 7pm Room 54-100

Self-Driving Cars: Overview and Recent Advances

[Slides] - [Lecture Video] *(Available Soon)*

Lecture Wed, Jan 10, 7pm Room 54-100

Deep RL for Driving Fast and Avoiding Crashes

[Slides] - [Lecture Video] *(Available Soon)*

Lecture Thu, Jan 11, 7pm Room 54-100

Deep Learning for Driving Scene Understanding

[Slides] - [Lecture Video] *(Available Soon)*

Guest Talk Fri, Jan 12, **1pm** Room 32-123 * **Notice:** Different time and room!

Sacha Arnoud

Director of Engineering, Waymo

Guest Talk Tue, Jan 16, 7pm Room 54-100

Emilio Frazzoli

CTO, nuTonomy

Previously: Professor, MIT

Lecture Wed, Jan 17, 7pm Room 54-100

Deep Learning for Driver State Sensing

[Slides] - [Lecture Video] *(Available Soon)*

Guest Talk Thu, Jan 18, 7pm Room 54-100

Oliver Cameron

CEO, Voyage

Previously: Head, Udacity Self-Driving Car Program

Guest Talk Fri, Jan 19, 7pm Room 54-100

Sterling Anderson

Co-Founder, Aurora

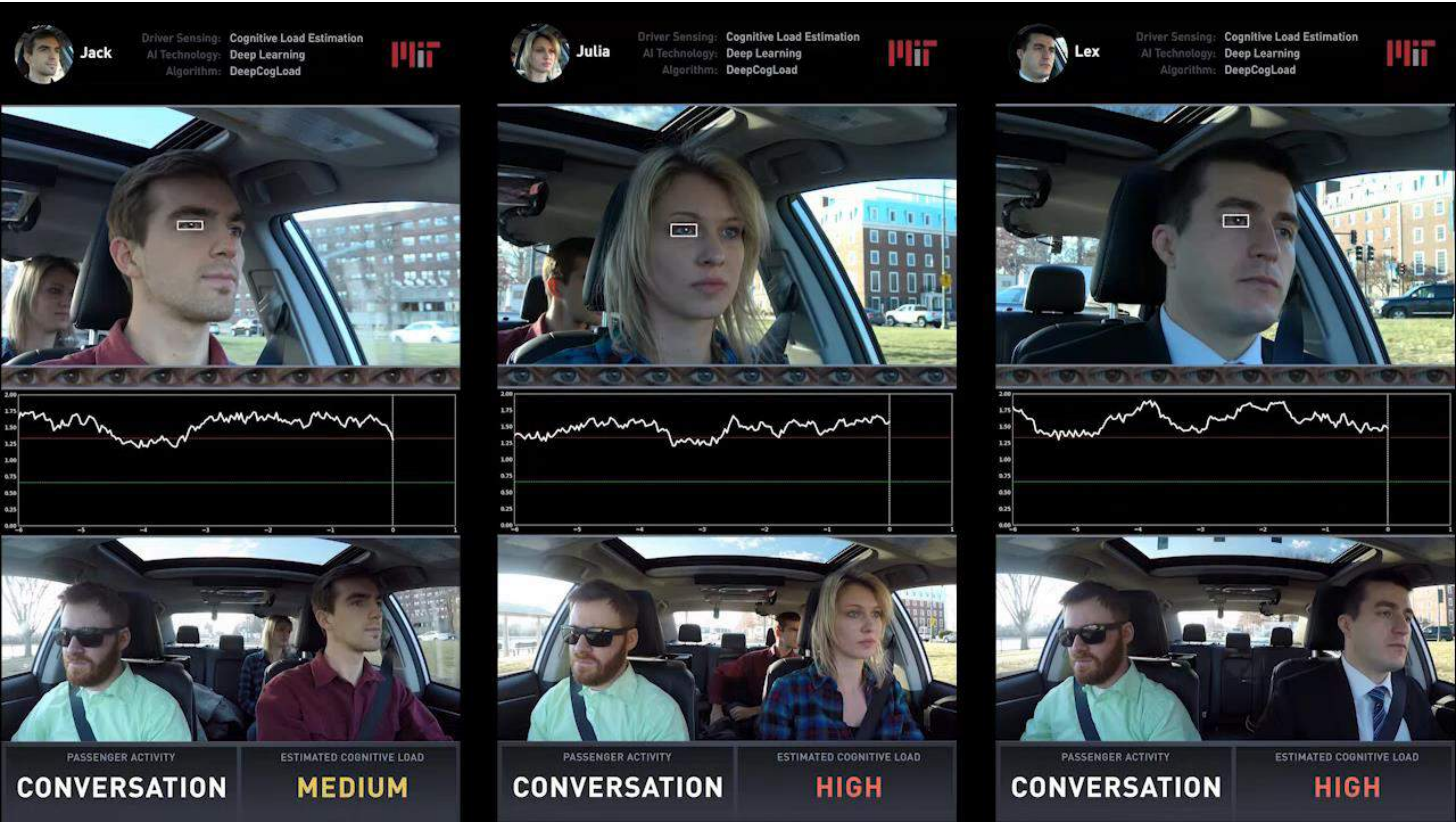
Previously: Director, Tesla Autopilot

Why Self-Driving Cars?

- Quite possibly, the first wide reaching and profound integration of personal robots in society.
  - **Wide reaching:** 1 billion cars on the road.
  - **Profound:** Human gives control of his/her life directly to robot.
  - **Personal:** One-on-one relationship of communication, collaboration, understanding and trust.

A self-driving car may be more a Personal Robot and less a perfect Perception-Control system. Why:

- **Flaws need humans:**
  The scene understanding problem requires much more than pixel-level labeling
- **Exist with humans:**
  Achieving both an enjoyable and safe driving experience may require "driving like a human".



Why Self-Driving Cars?

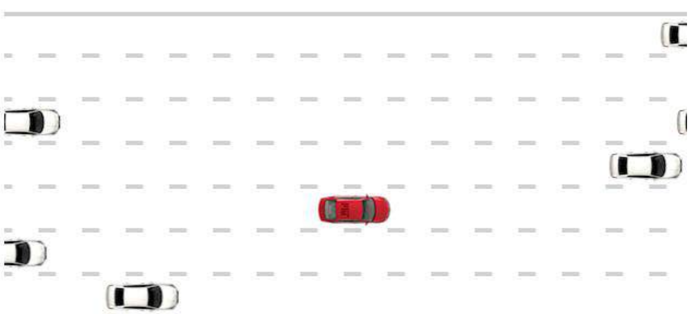
- Opportunity to explore the **nature of intelligence** and the role of intelligent systems in society, because full autonomy may require **human-level artificial intelligence.**

See also our class exploring
human-level artificial intelligence:
MIT 6.S099 Artificial General Intelligence
https://agi.mit.edu

Human-Centered Artificial Intelligence Approach

90% ← No — Human Needed — Yes → 10%

Solve the perception-control problem where **possible**:

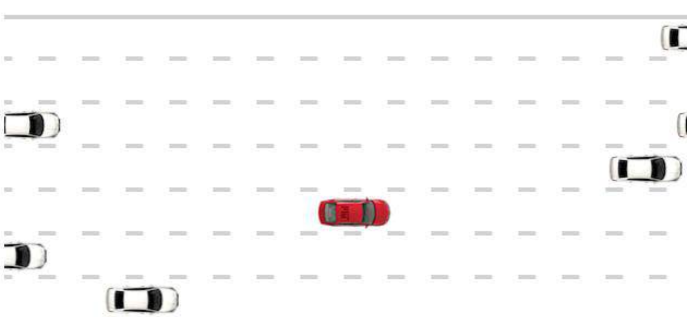


And where **not possible**: involve the human

Why Deep Learning?

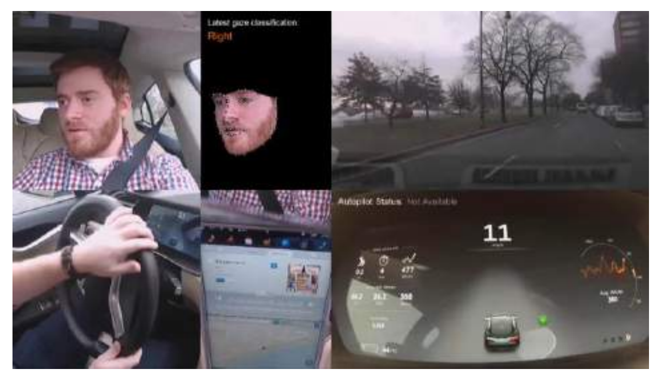
Deep Learning:
Learn effective perception-control from **data**

Solve the perception-control problem where **possible**:



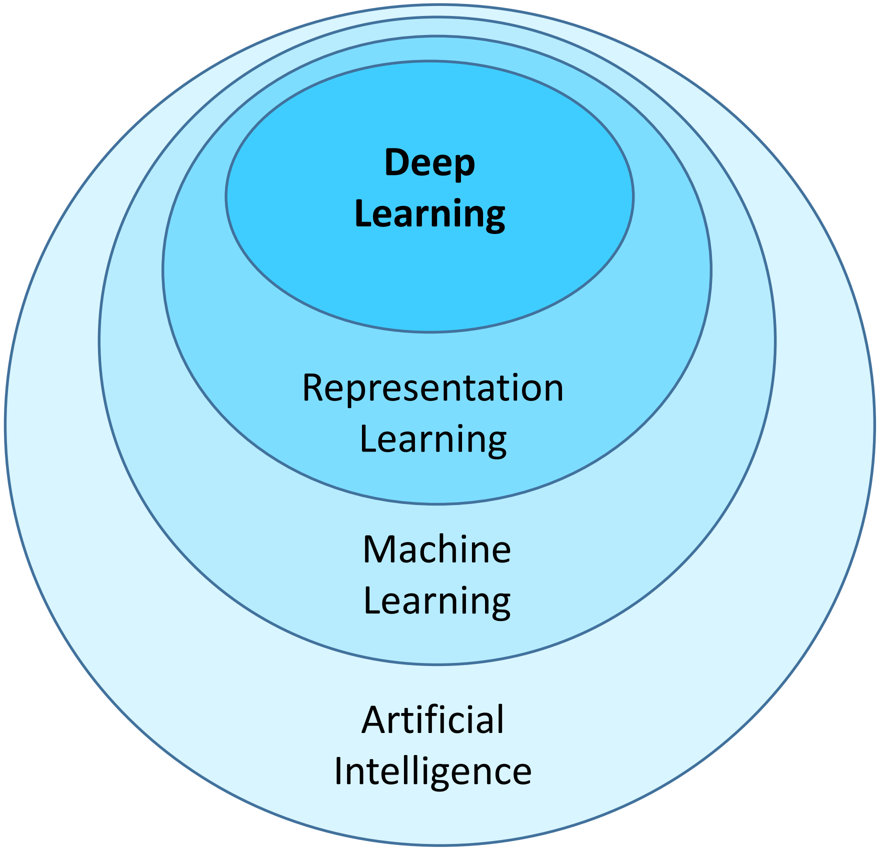
Deep Learning:
Learn effective human-robot interaction from **data**

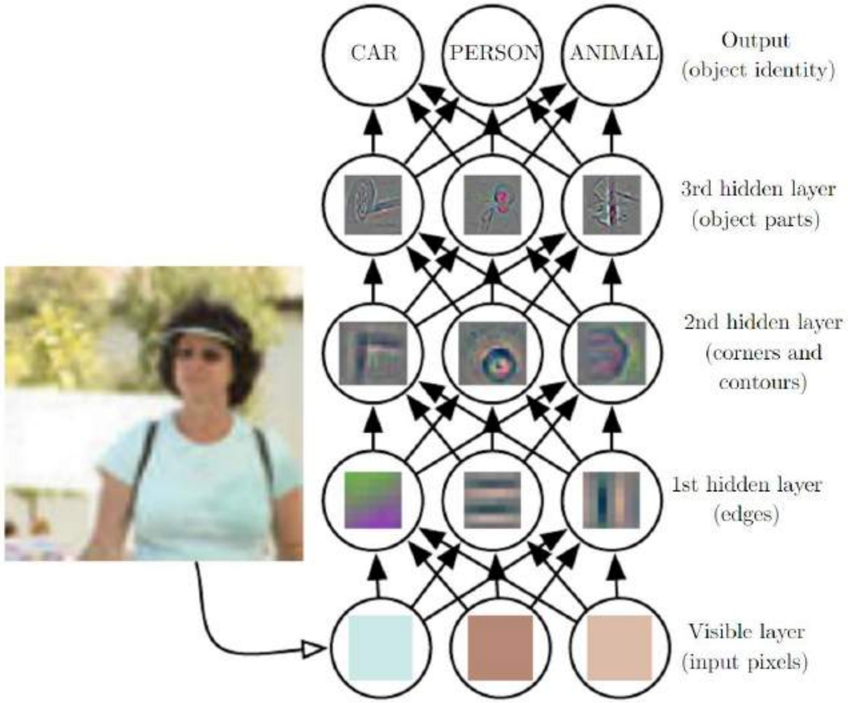
And where **not possible**: involve the human



Deep Learning is **Representation Learning**

(aka Feature Learning)

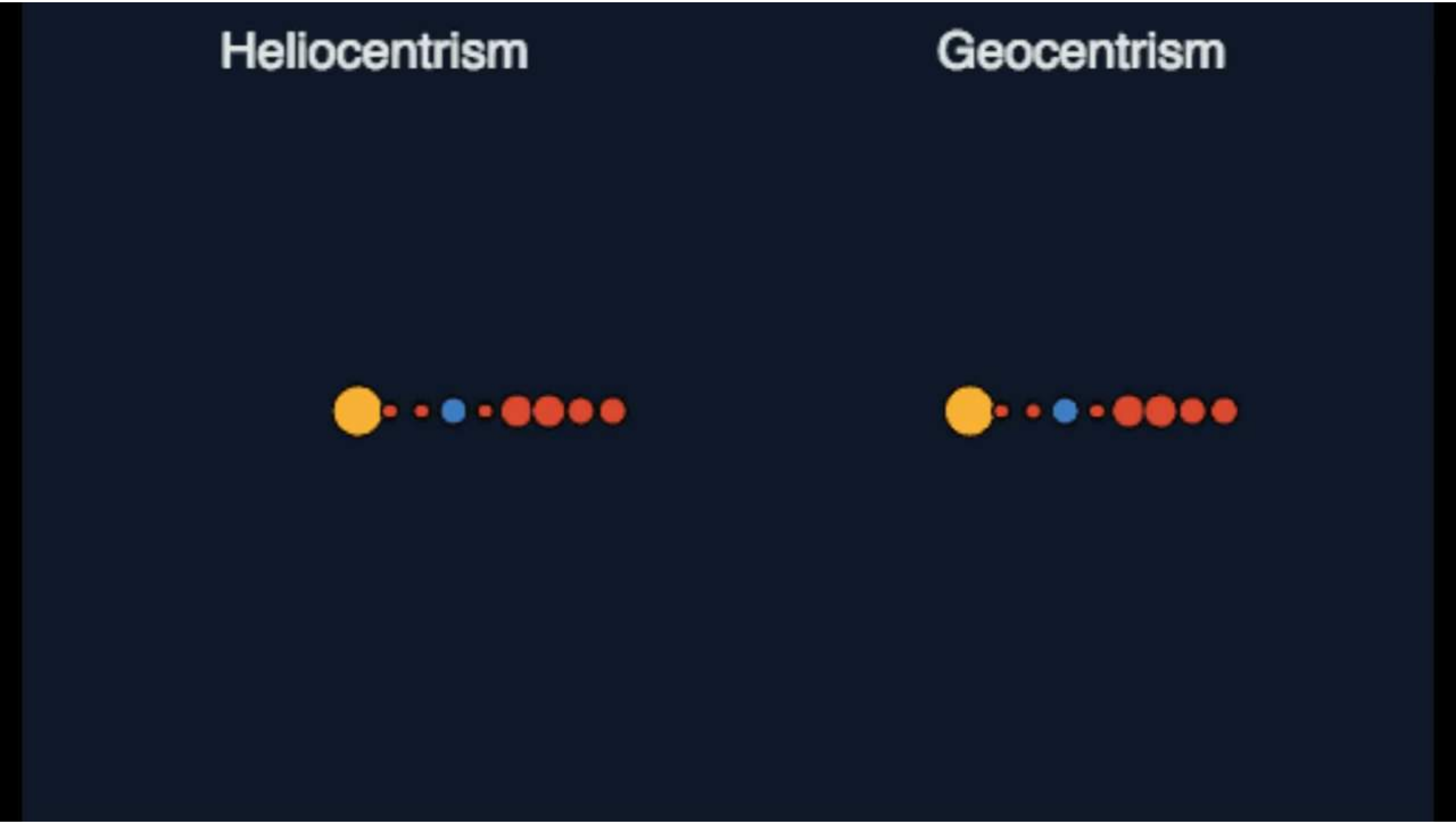




Intelligence: Ability to accomplish complex goals.

Understanding: Ability to turn complex information to into simple, useful information.

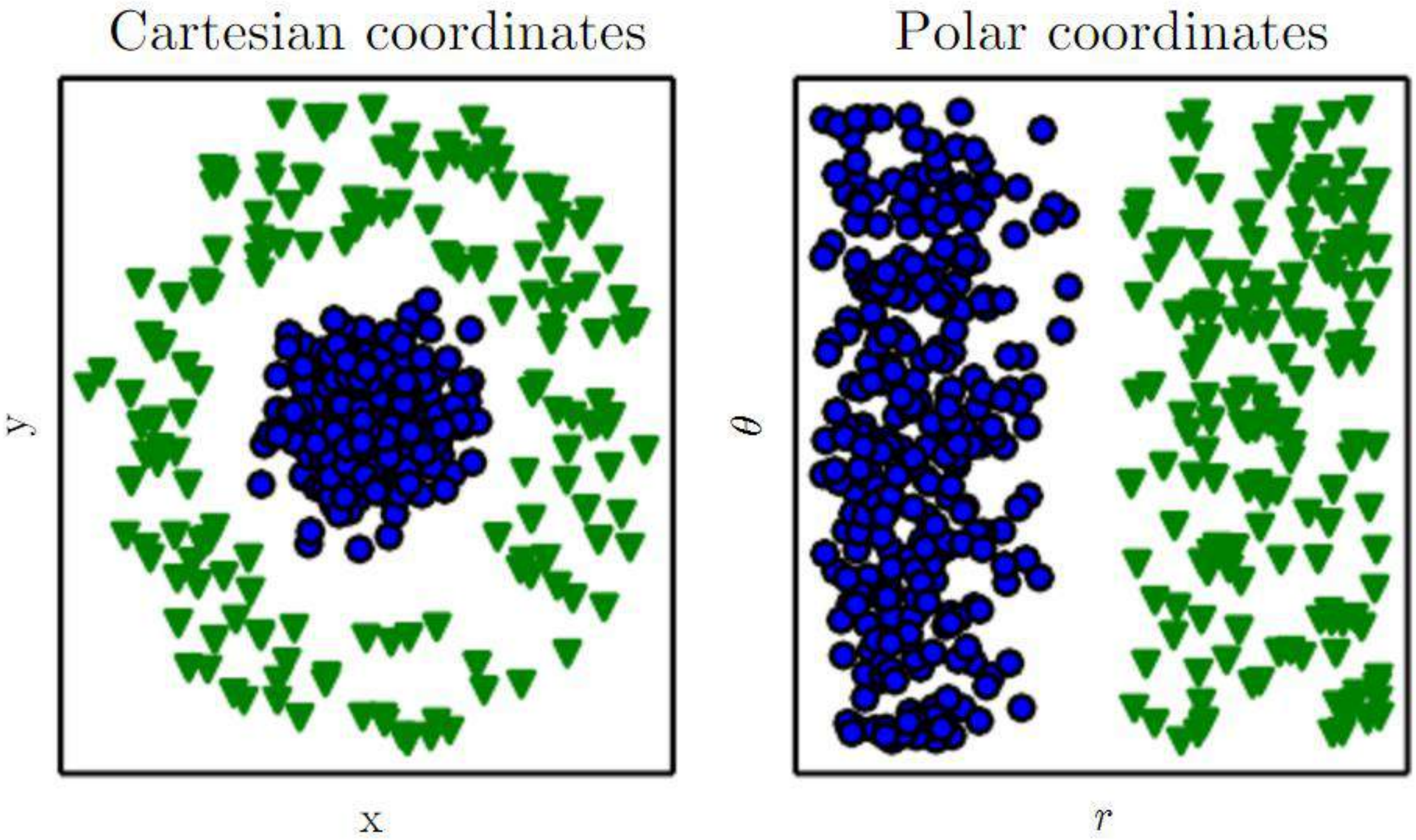
Representation Matters



Sun-Centered Model
(Formalized by Copernicus in 16th century)

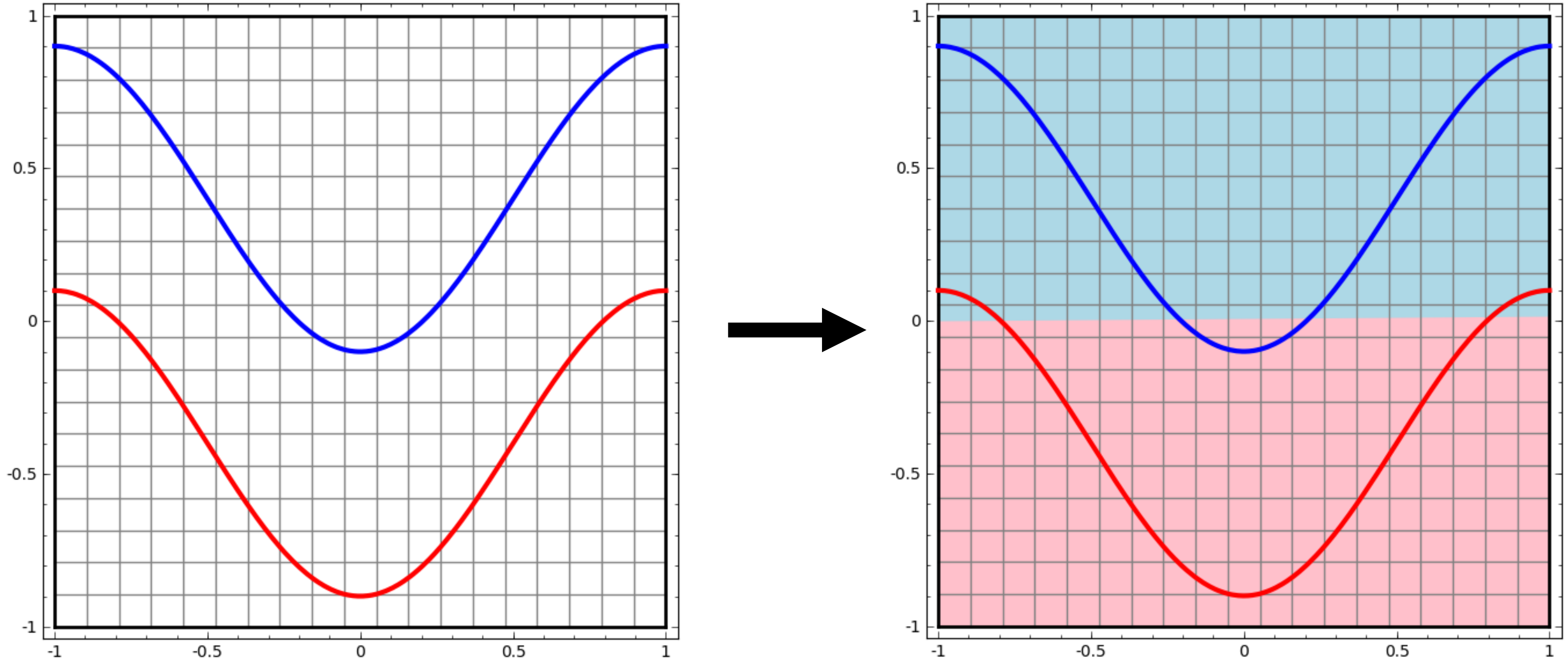
Earth-Centered Model

Representation Matters



Task: Draw a line to separate the **green triangles** and **blue circles**.

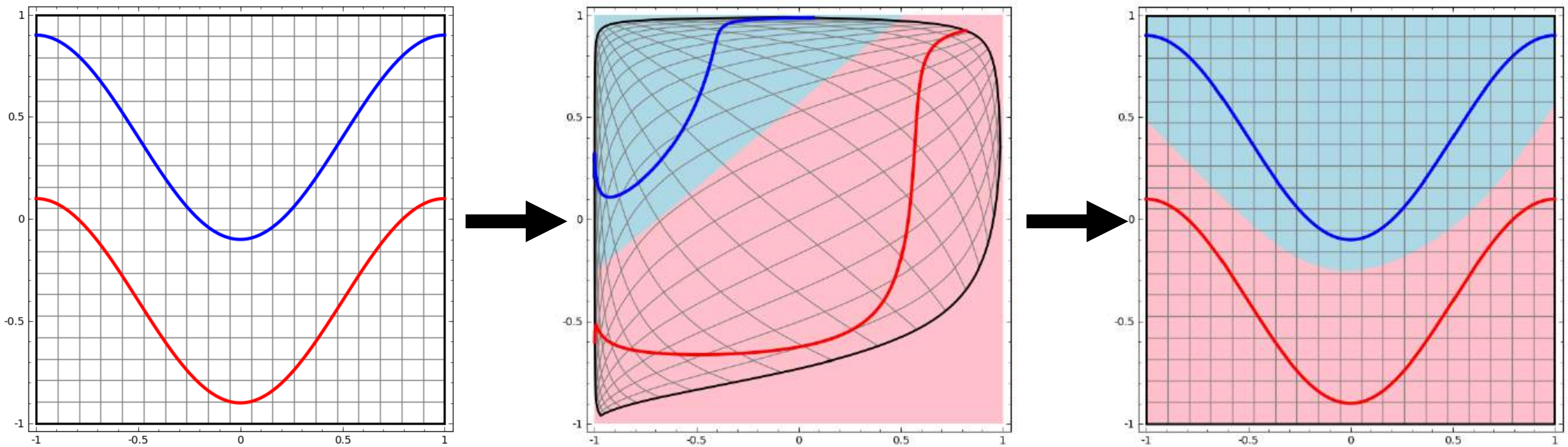
Representation Matters



Task: Draw a line to separate the **blue curve** and **red curve**

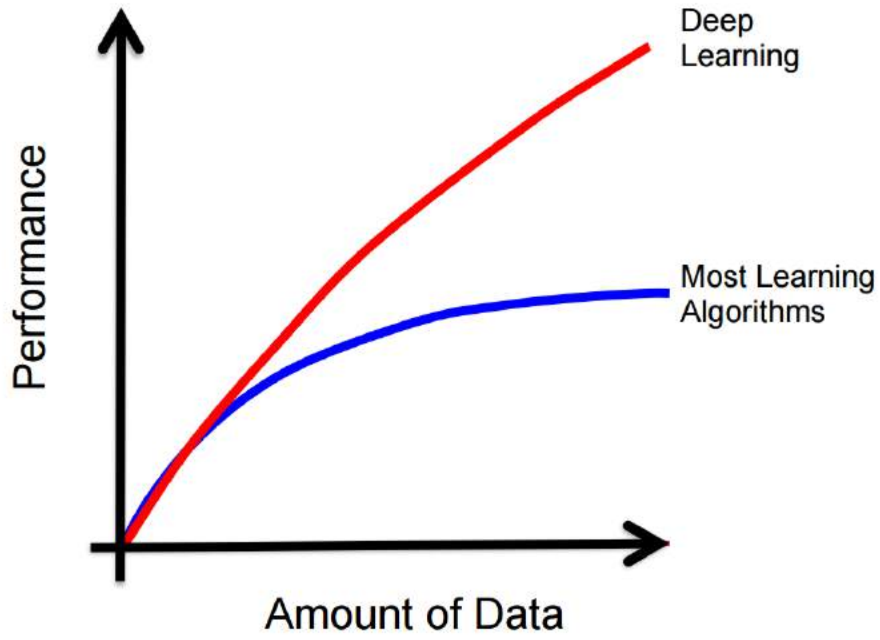
Deep Learning is **Representation Learning**

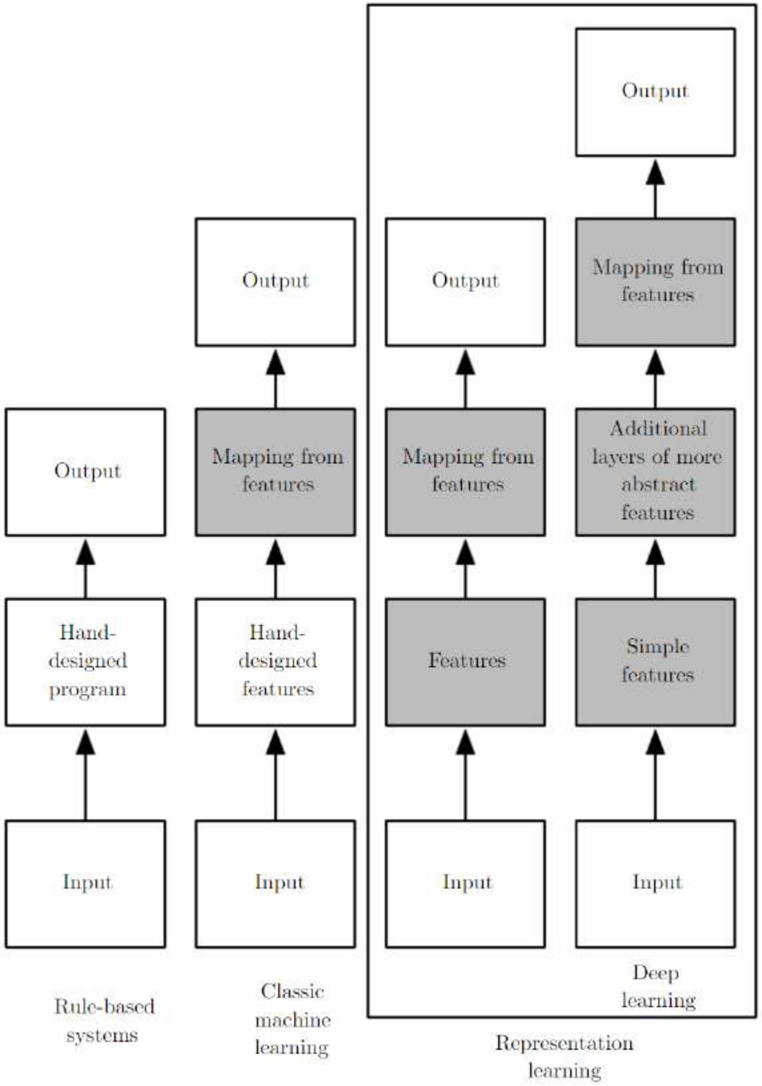
(aka Feature Learning)



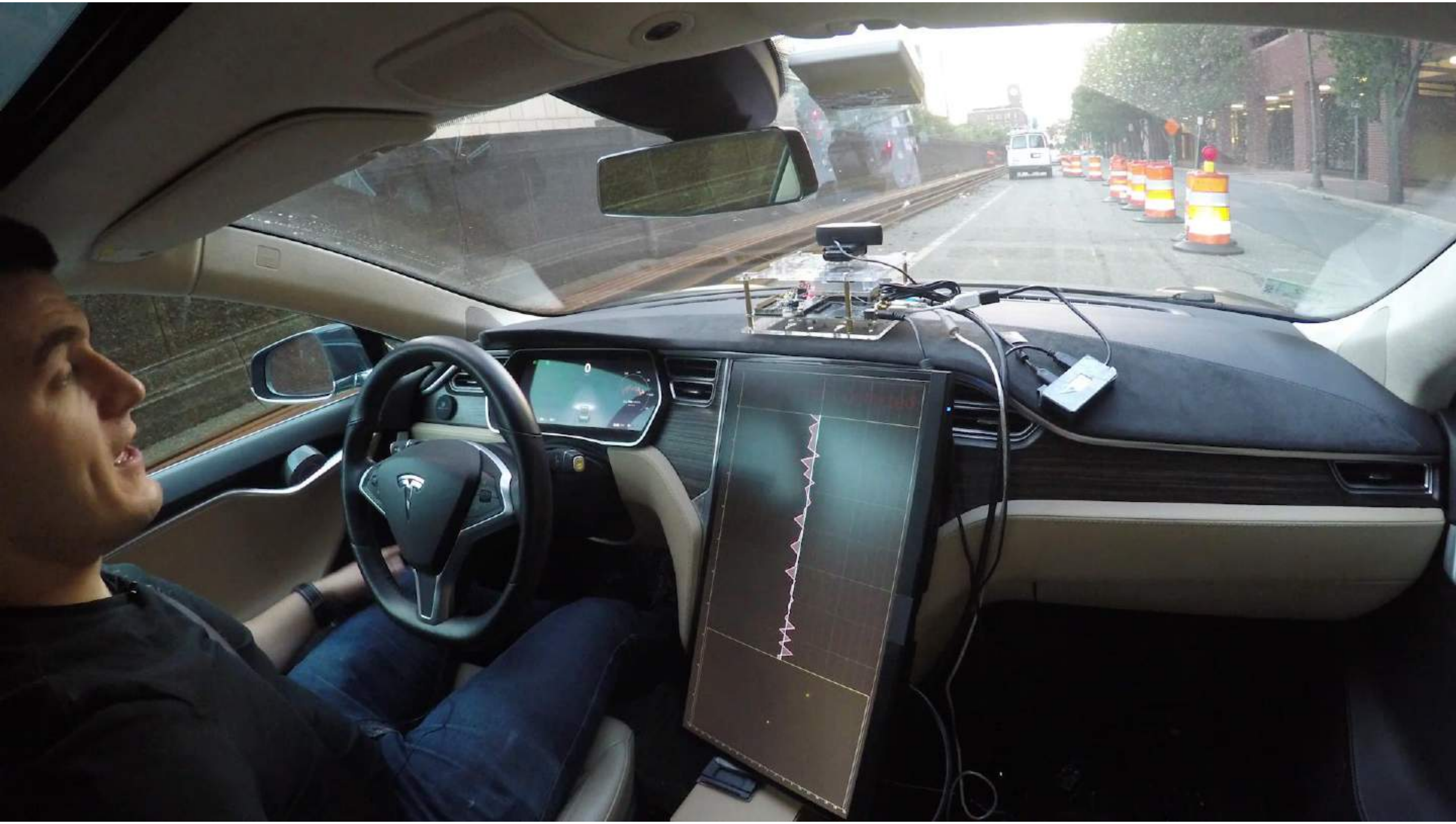
Task: Draw a line to separate the **blue curve** and **red curve**

Deep Learning: **Scalable** Machine Learning



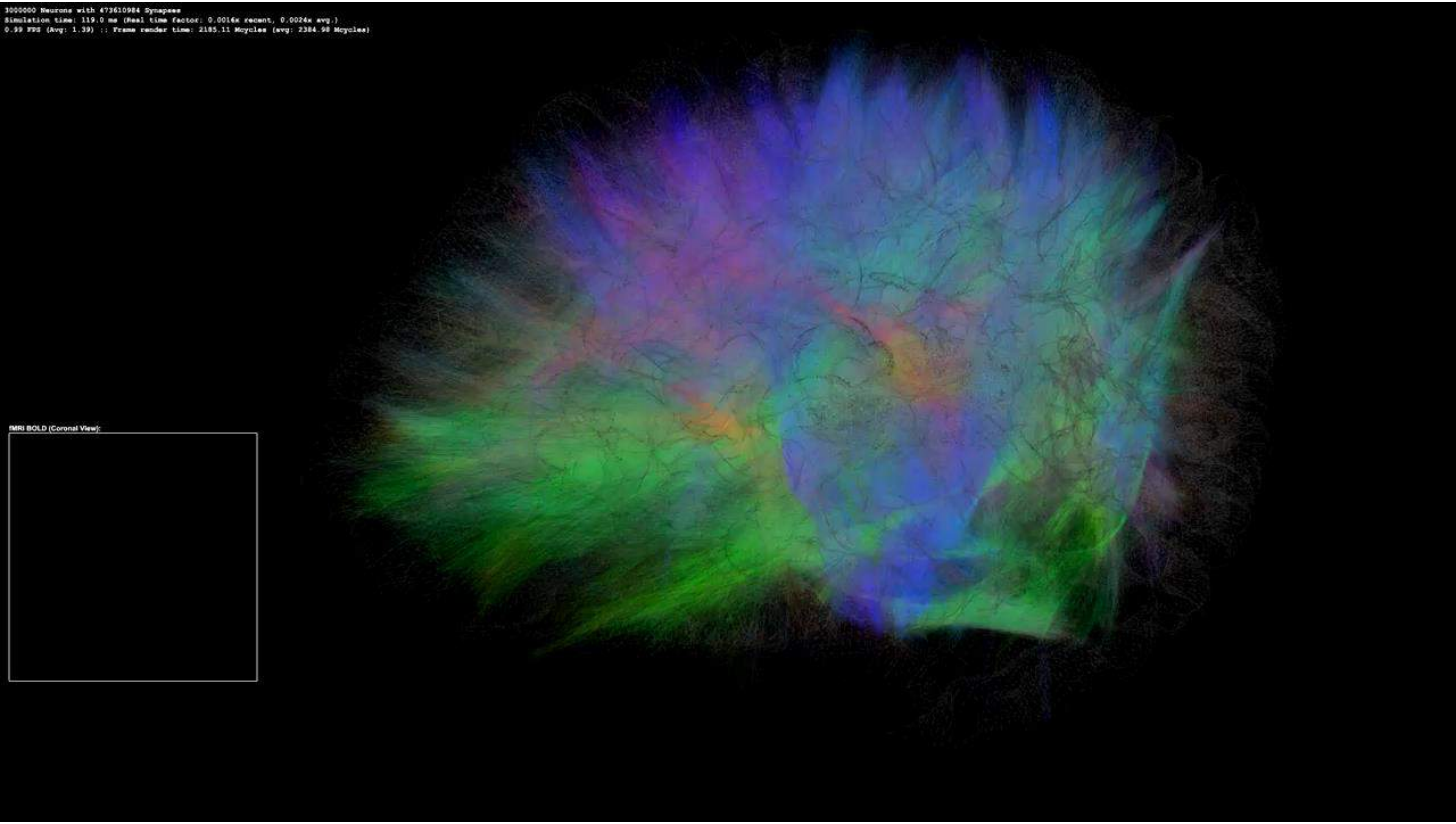


- Deep learning approaches improve with **more data**.
- Artificial intelligence system in the real-world are all about generalizing over the **edge cases**



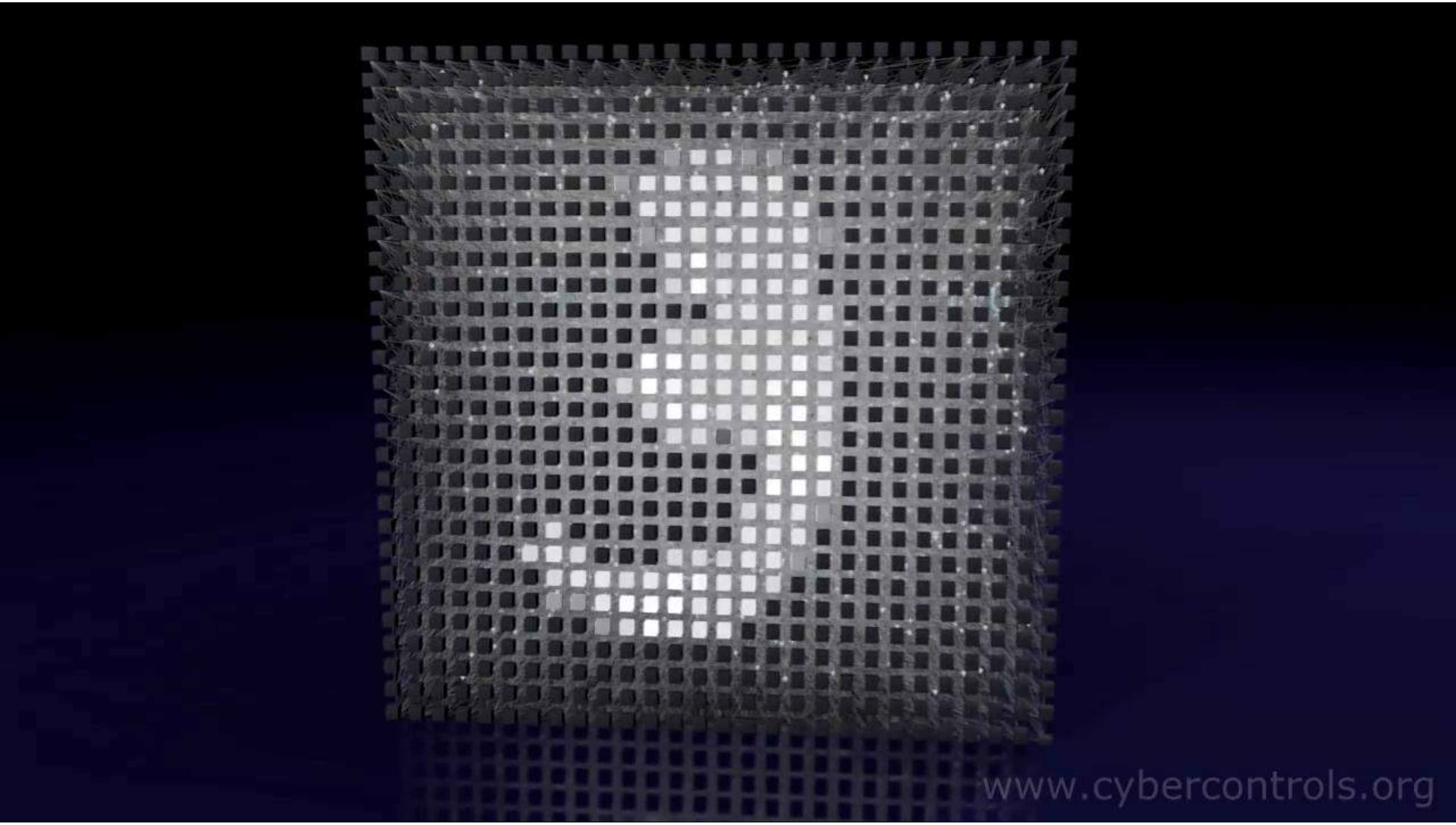
Biological Neural Network

- Thalamocortical brain network (simulation video shown below)
  - 3 million neurons, 476 million synapses
- Full human brain:
  - 100 billion neurons, 1,000 trillion synapses

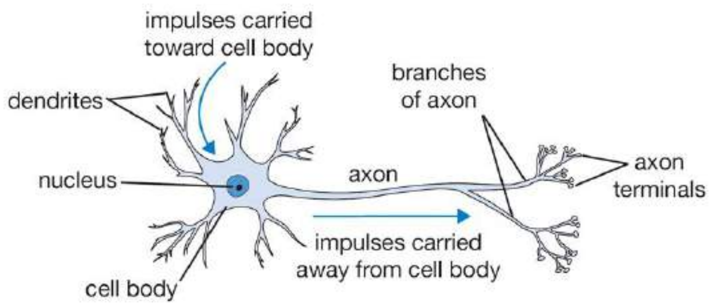


Artificial Neural Network

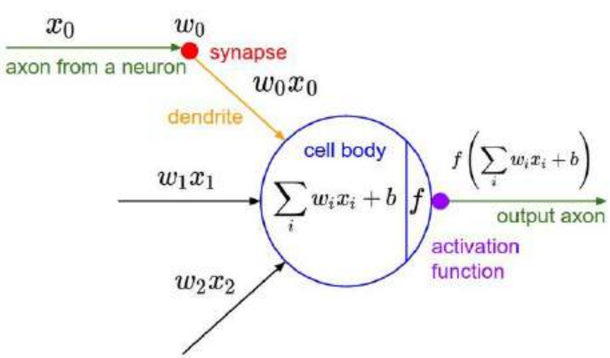
- **Human neural network:** 100 billion neurons, 1,000 trillion synapses
- **ResNet-152 neural network:** 60 million synapses



Neuron: Biological Inspiration for Computation



- **Neuron:** computational building block for the brain



- **(Artificial) Neuron:** computational building block for the “neural network”

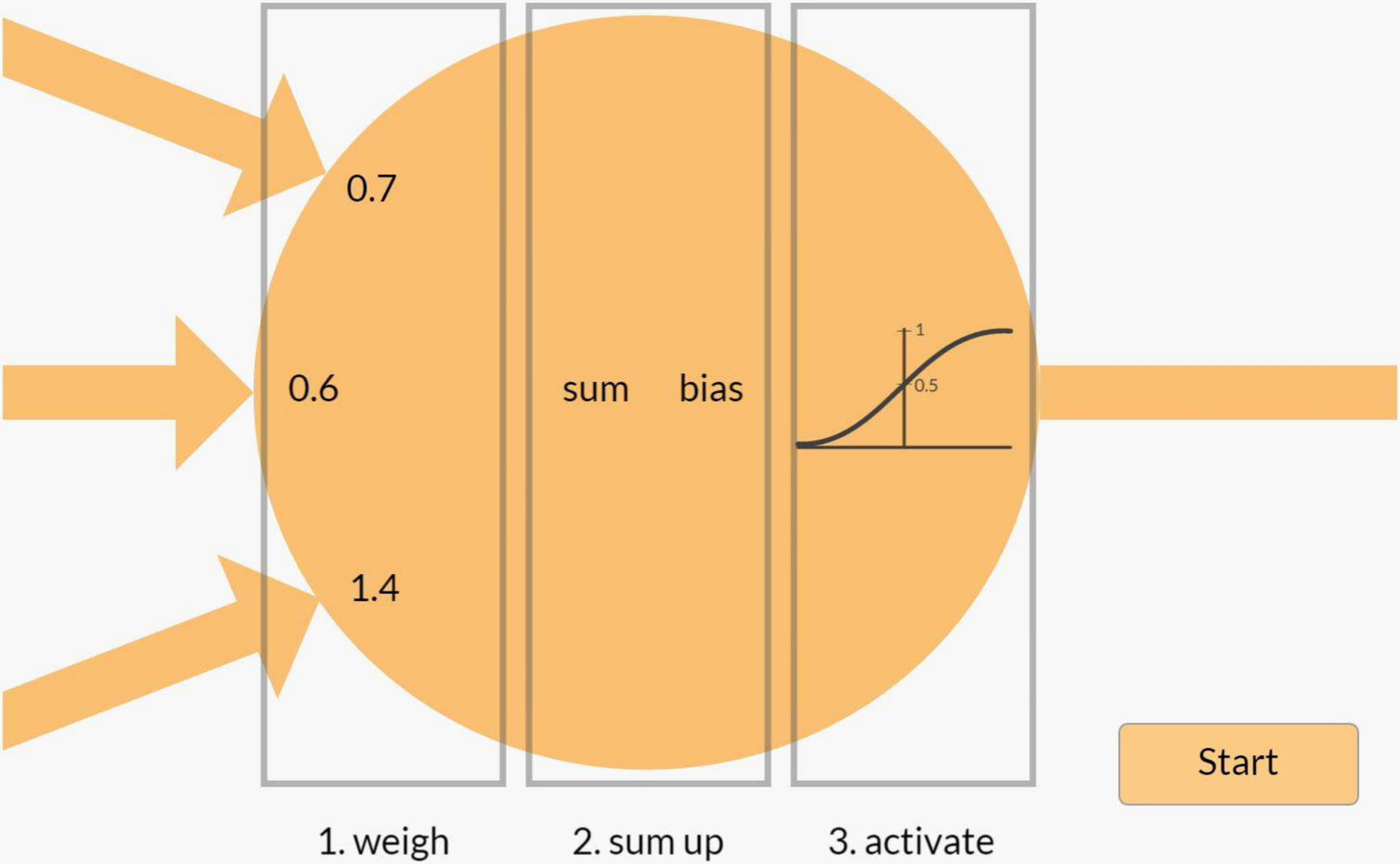
Differences (among others):

- **Parameters:** Human brains have ~10,000,000 times synapses than artificial neural networks.
- **Topology:** Human brains have no “layers”. Topology is complicated.
- **Async:** The human brain works asynchronously, ANNs work synchronously.
- **Learning algorithm**: ANNs use gradient descent for learning. Human brains use … (we don’t know)
- **Processing speed**: Single biological neurons are slow, while standard neurons in ANNs are fast.
- **Power consumption:** Biological neural networks use very little power compared to artificial networks
- **Stages:** Biological networks usually don't stop / start learning. ANNs have different fitting (train) and prediction (evaluate) phases.

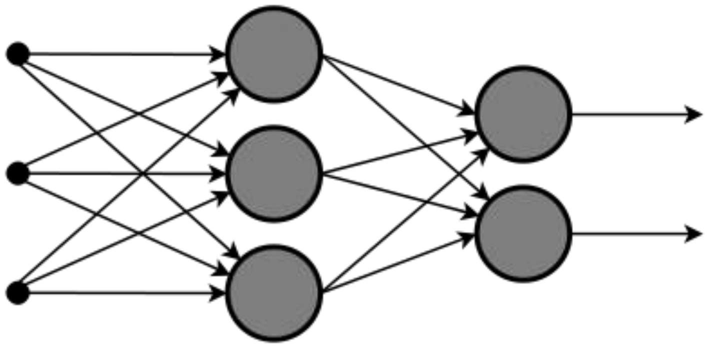
Similarity (among others):

- Distributed computation on a large scale.

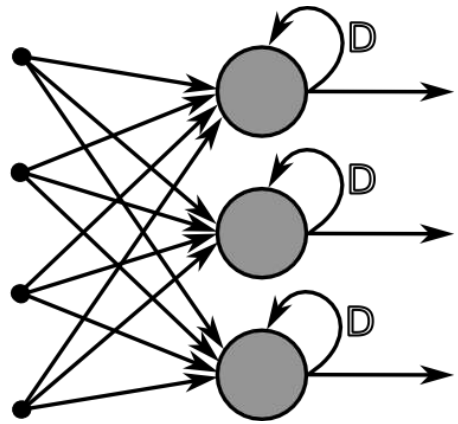
Neuron: Forward Pass



Combining Neurons into Layers



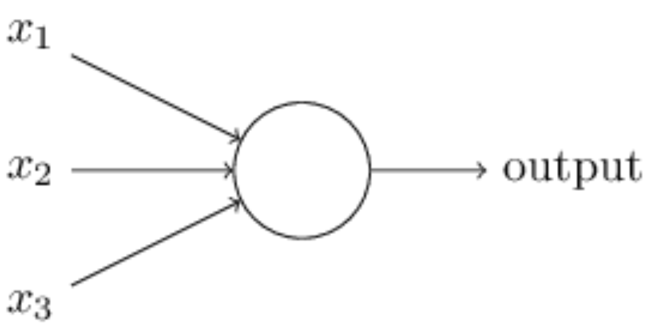
Feed Forward Neural Network

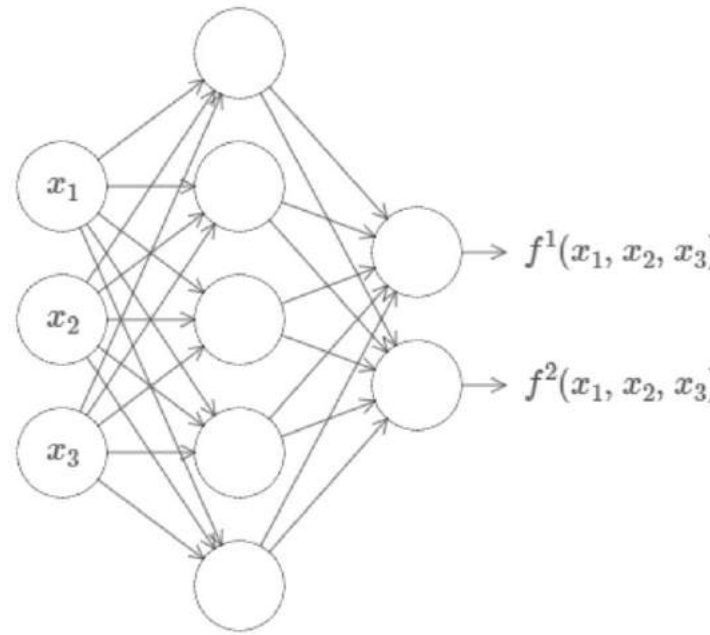


Recurrent Neural Network

- Have state memory
- Are hard to train

Combing Neurons in Hidden Layers:
The “Emergent” Power to Approximate



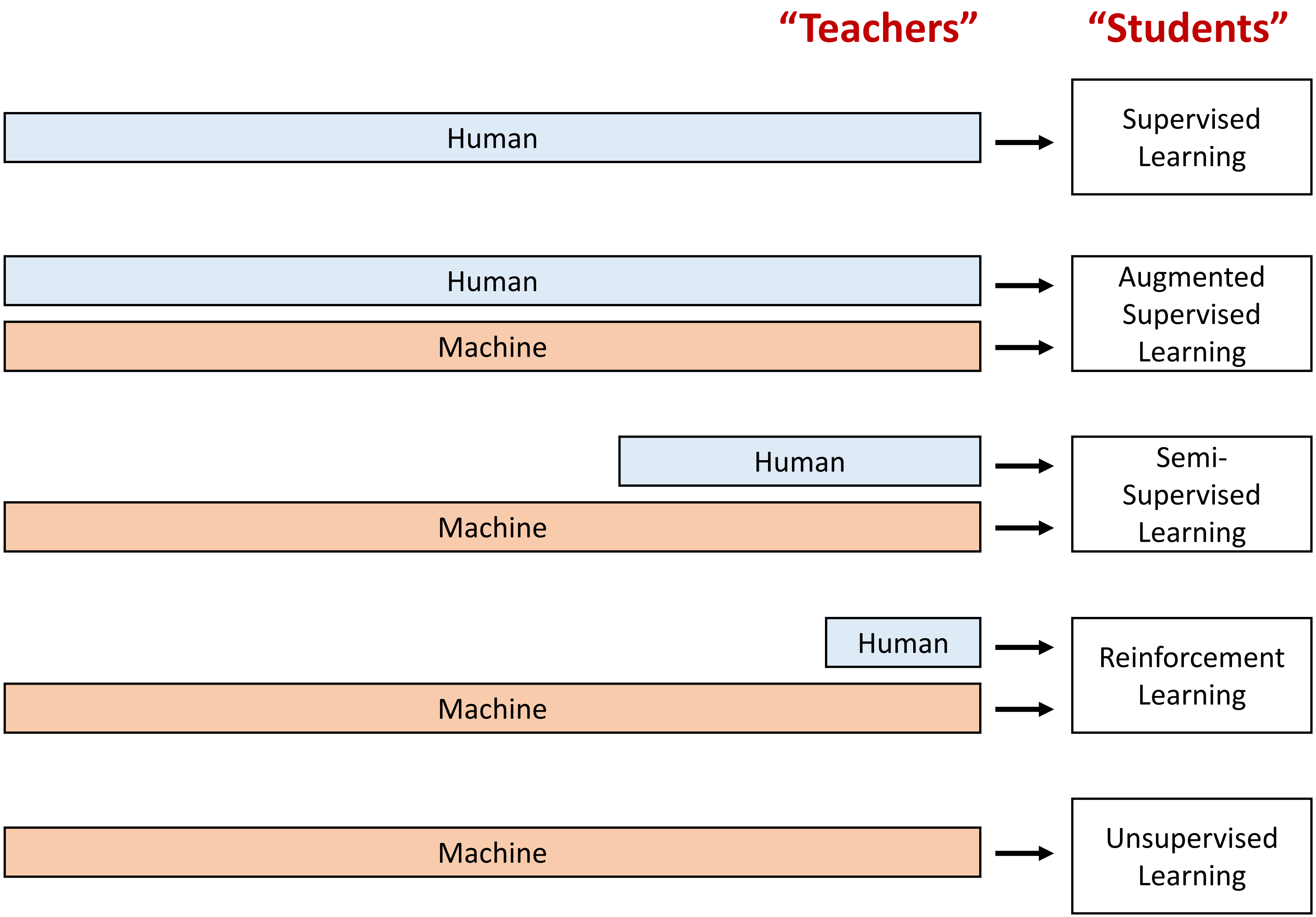


Universality: For any arbitrary function f(x), there exists a neural network that closely approximate it for any input x

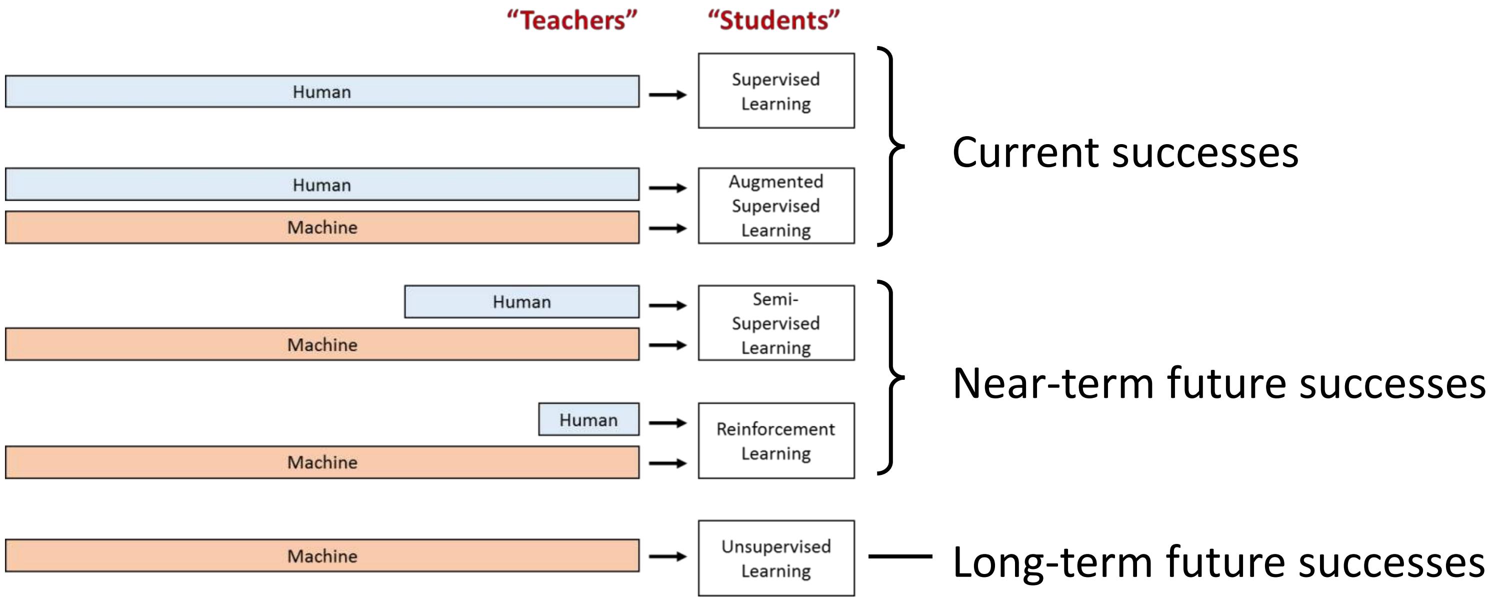
Universality is an incredible property!* And it holds for just 1 hidden layer.

* Given that we have good algorithms for training these networks.

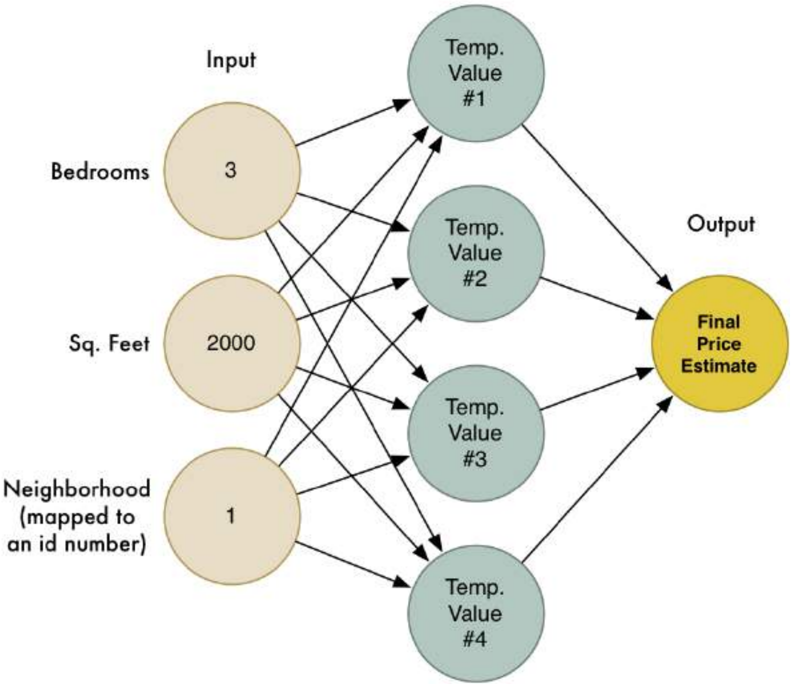
Deep Learning from Human and Machine

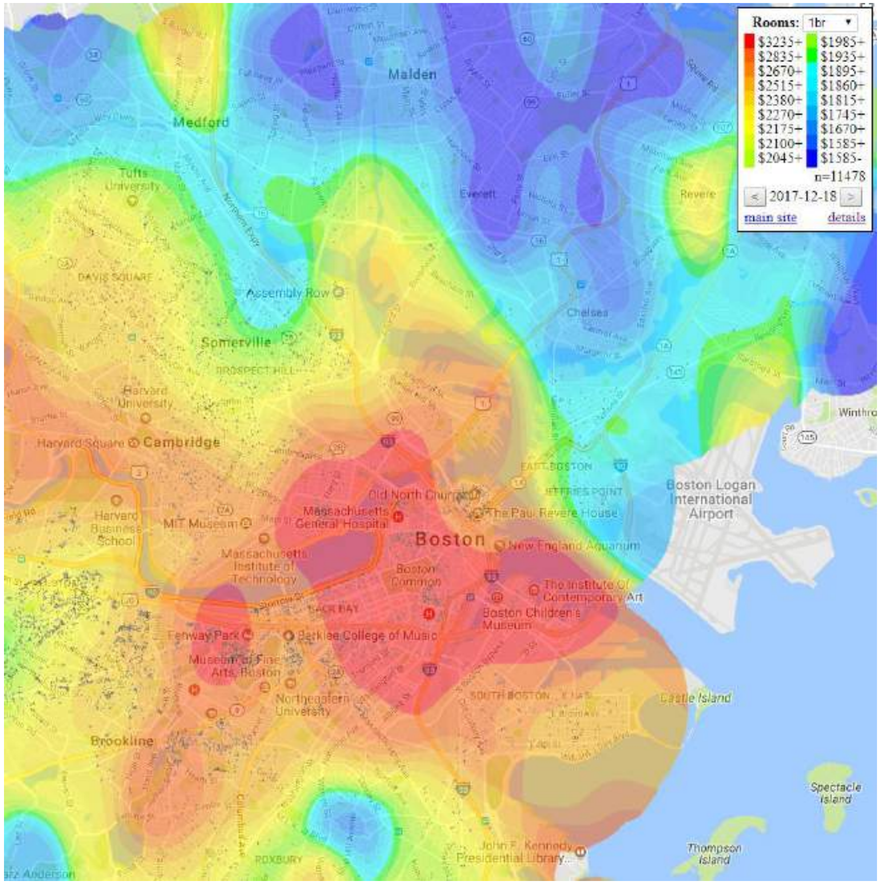


Deep Learning from Human and Machine



Special Purpose Intelligence:
Estimating Apartment Cost

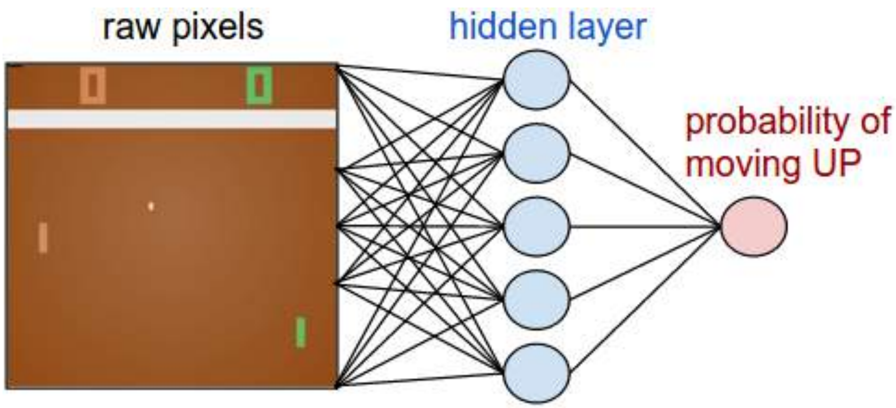




(Toward) General Purpose Intelligence:
Pong to Pixels

Andrej Karpathy. "Deep Reinforcement Learning: Pong from Pixels." 2016.

Policy Network:

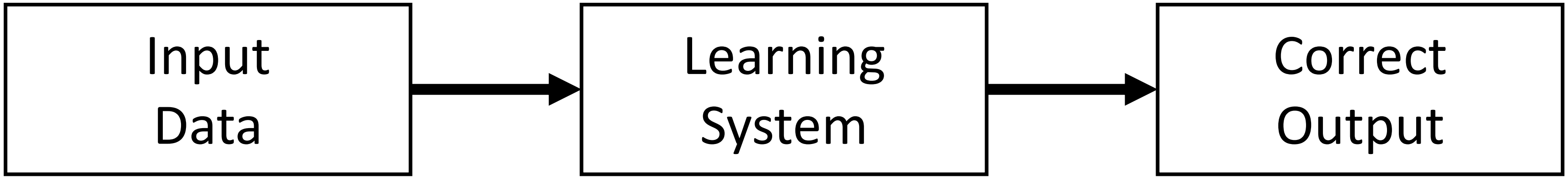


- 80x80 image (difference image)
- 2 actions: up or down
- 200,000 Pong games

This is a step towards general purpose artificial intelligence!

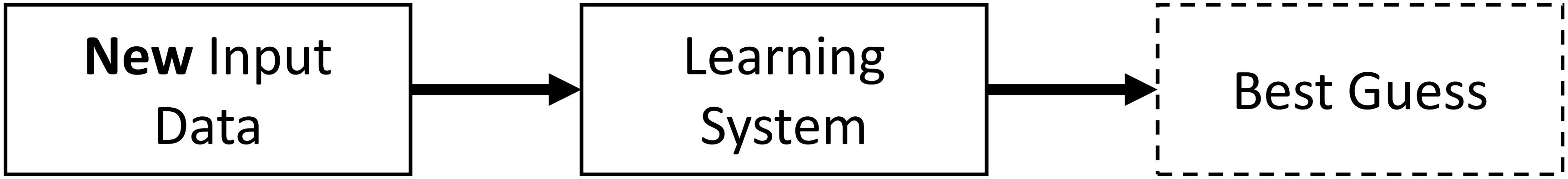
Deep Learning: Training and Testing

Training Stage:



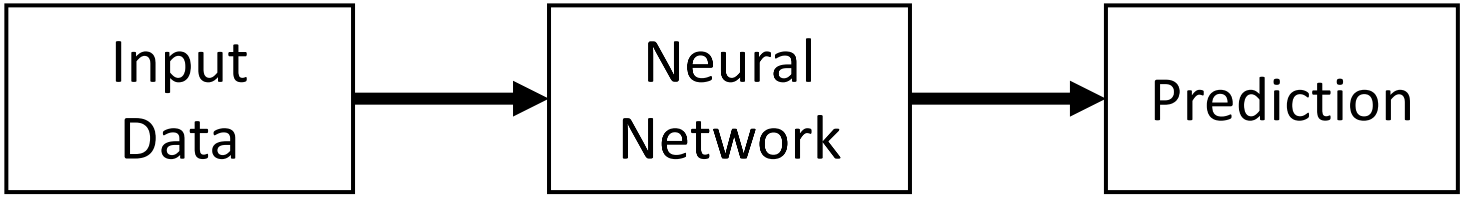
(aka "Ground Truth")

Testing Stage:

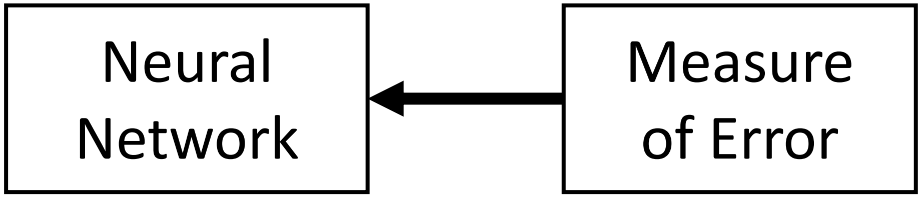


How Neural Networks Learn: Backpropagation

Forward Pass:

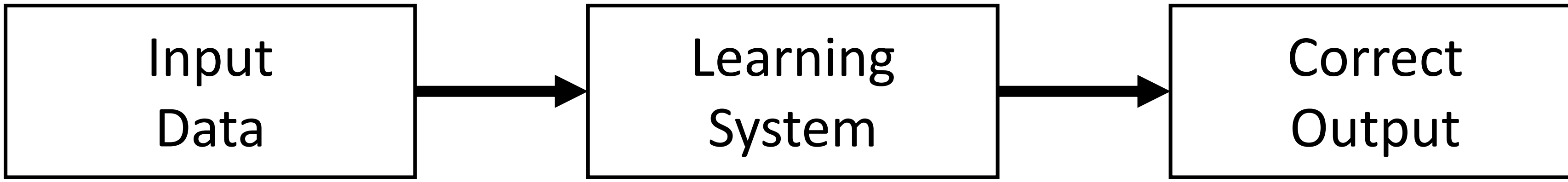


Backward Pass (aka Backpropagation):



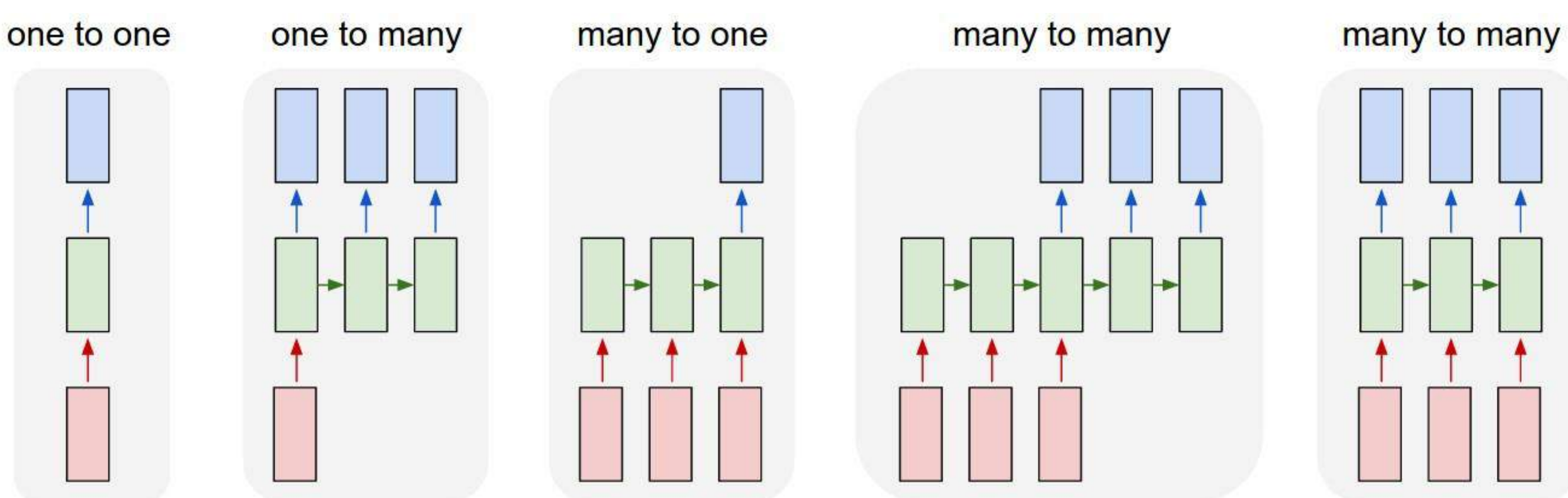
Adjust to Reduce Error

What can we do with Deep Learning?

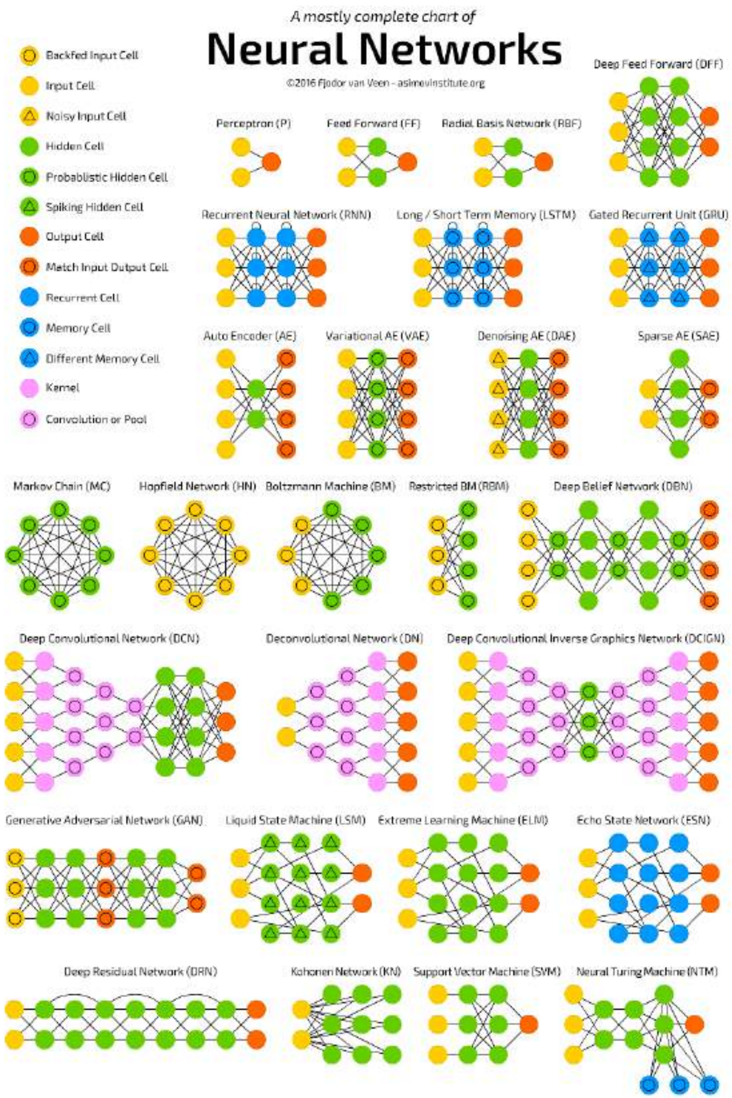


- Number
- Vector of numbers
- Sequence of numbers
- Sequence of vectors of numbers

- Number
- Vector of numbers
- Sequence of numbers
- Sequence of vectors of numbers



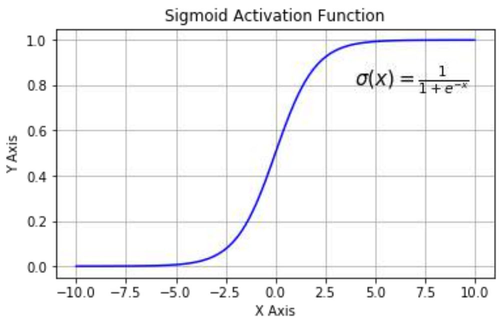
Useful Deep Learning Terms

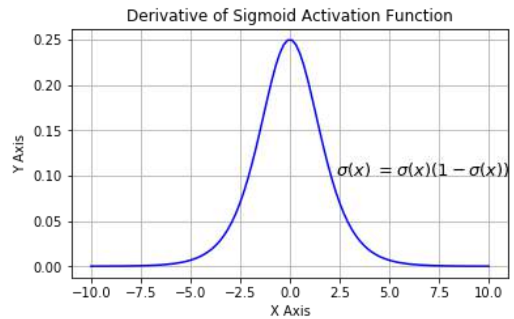


- Basic terms:
  - **Deep Learning ≈ Neural Networks**
  - **Deep Learning** is a subset of **Machine Learning**
- Terms for neural networks:
  - **MLP:** Multilayer Perceptron
  - **DNN:** Deep neural networks
  - **RNN:** Recurrent neural networks
    - **LSTM:** Long Short-Term Memory
  - **CNN:** Convolutional neural networks
  - **DBN:** Deep Belief Networks
- Neural network operations:
  - Convolution
  - Pooling
  - Activation function
  - Backpropagation

Key Concepts:

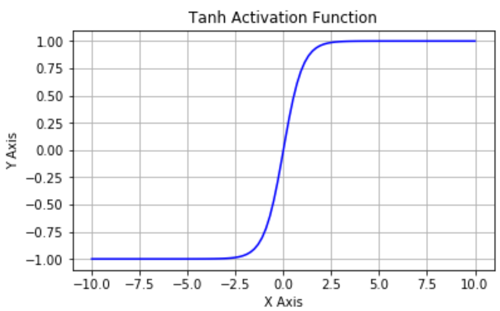
Activation Functions

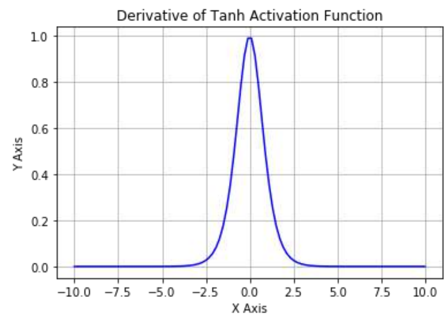




Sigmoid

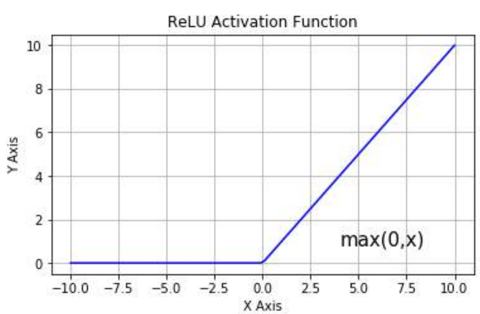
- Vanishing gradients
- Not zero centered

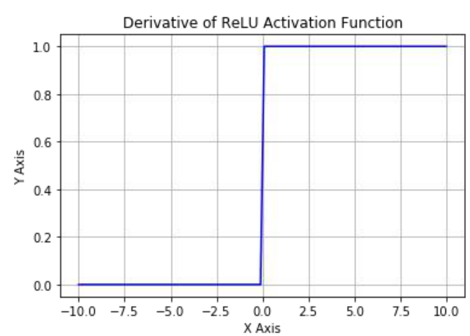




Tanh

- Vanishing gradients

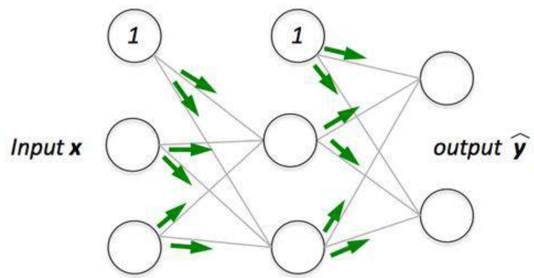


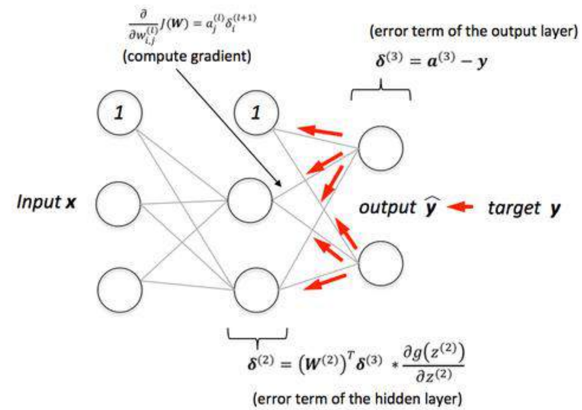


ReLU

- Not zero centered

Backpropagation





Task: Update the **weights** and **biases** to decrease **loss function**

Subtasks:

1. Forward pass to compute network output and "error"
2. Backward pass to compute gradients
3. A fraction of the weight's gradient is subtracted from the weight.

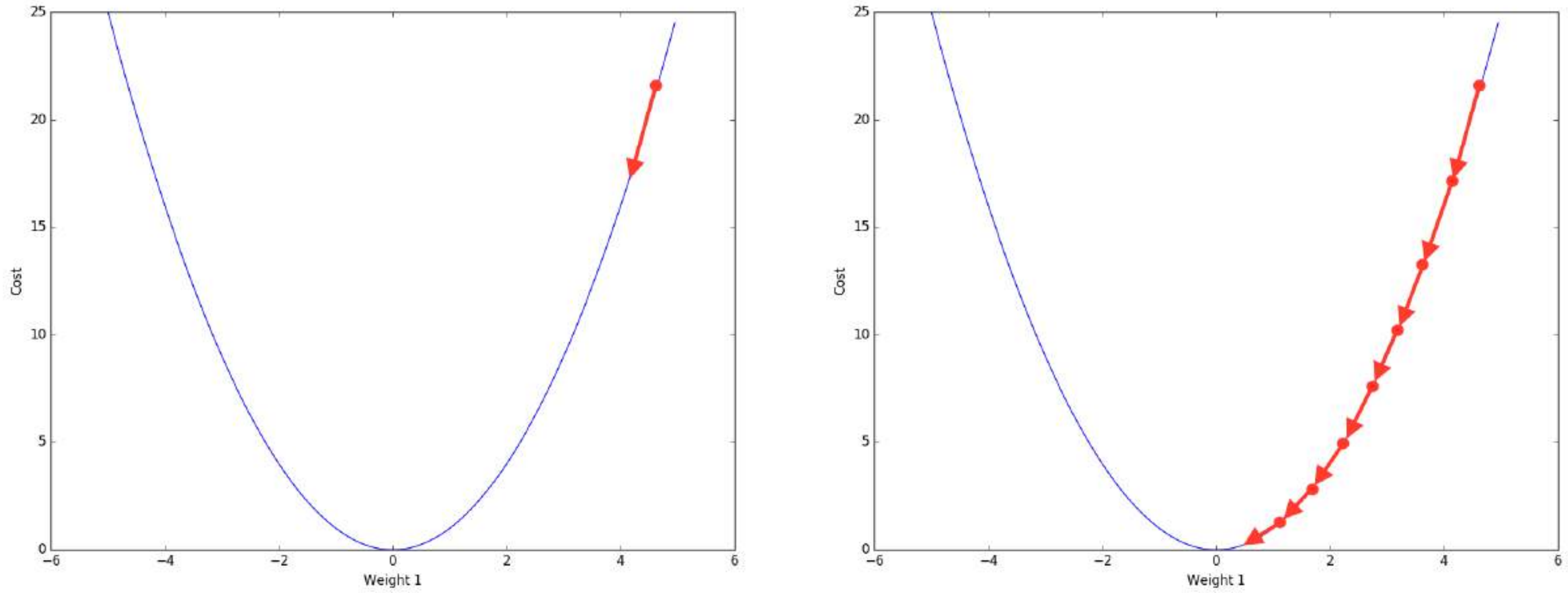
Learning Rate

Loss function:

$$C = \frac{(y-a)^2}{2}$$

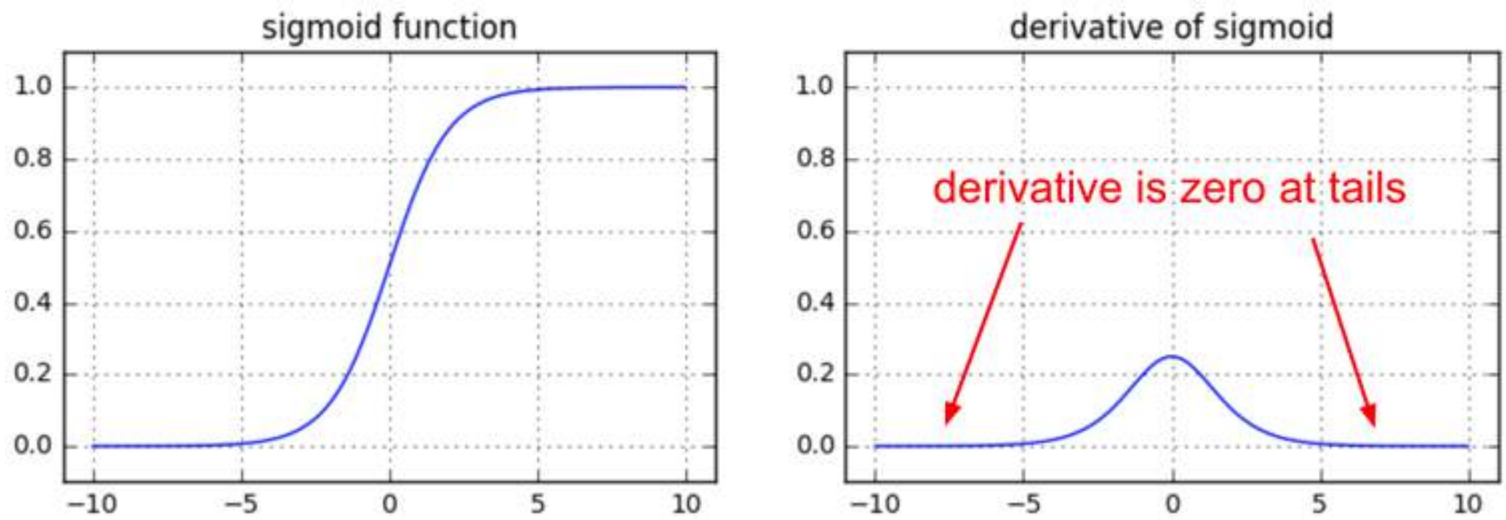
Learning is an Optimization Problem

Task: Update the **weights** and **biases** to decrease **loss function**



Use mini-batch or stochastic gradient descent.

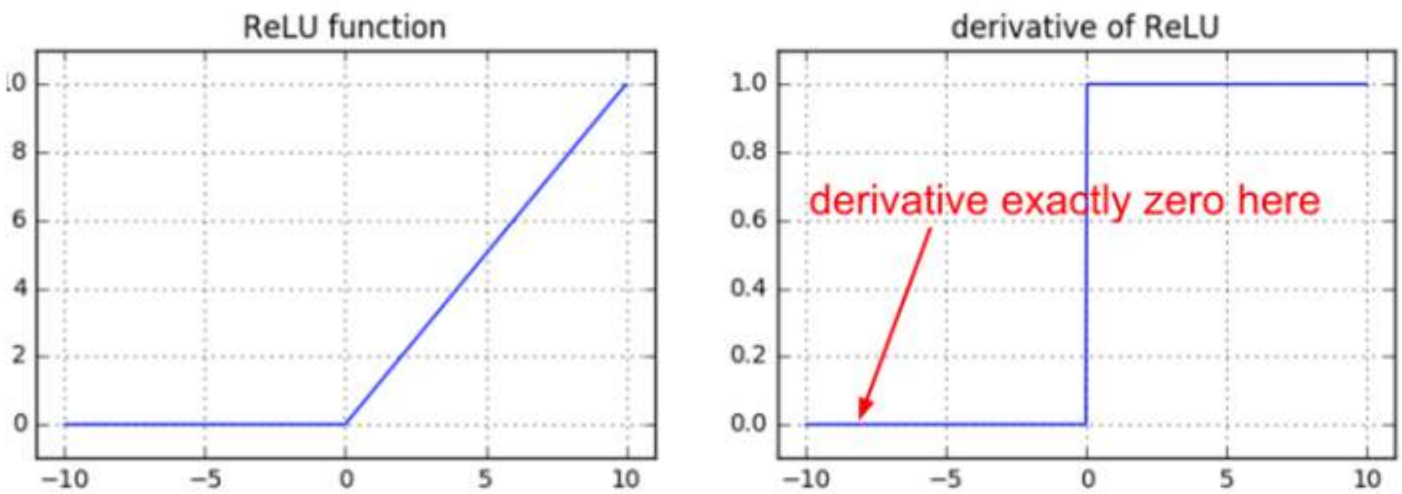
Optimization is Hard: Vanishing Gradients



$$\frac{d\sigma(x)}{dx} = (1 - \sigma(x))\,\sigma(x)$$

Partial derivatives are small = Learning is slow

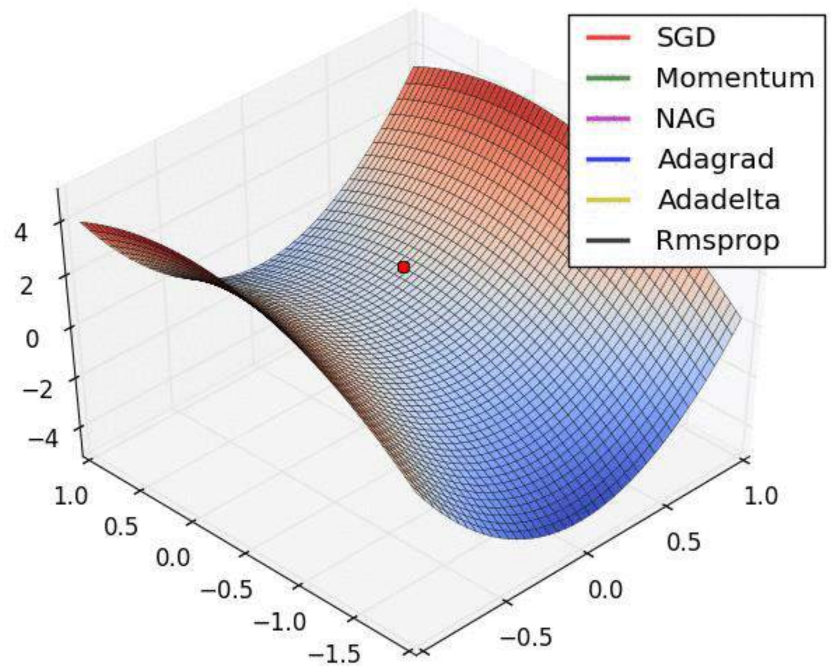
Optimization is Hard: Dying ReLUs



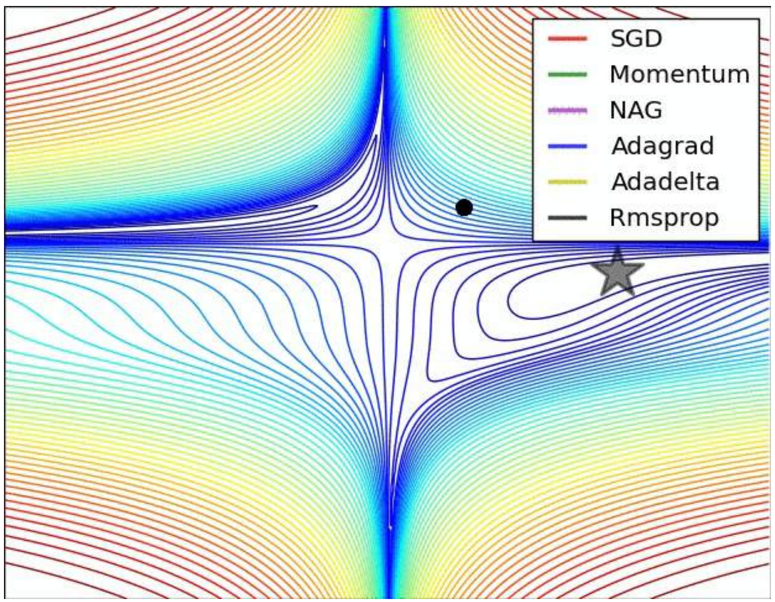
- If a neuron is initialized poorly, it might not fire for entire training dataset.
- Large parts of your network could be dead ReLUs!

References: [102]

Optimization is Hard: Saddle Point



Hard to break symmetry



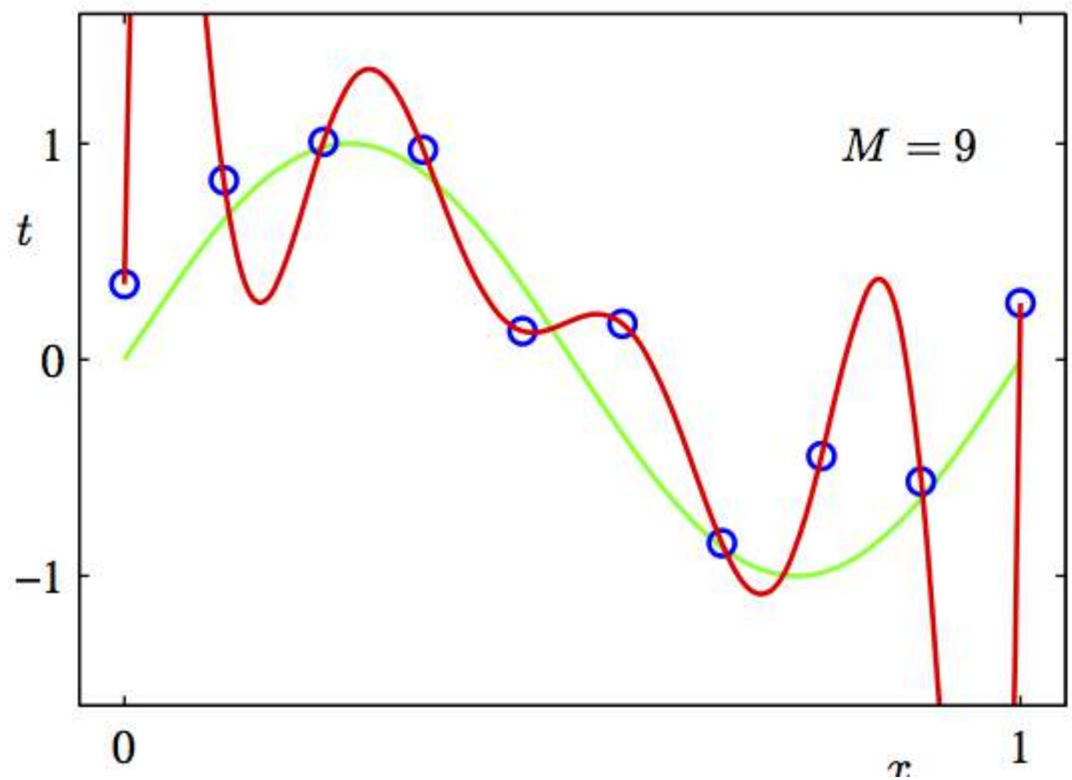
Vanilla SGD gets your there, but is slow sometimes.

References: [104]

Key Concepts:

Overfitting and Regularization

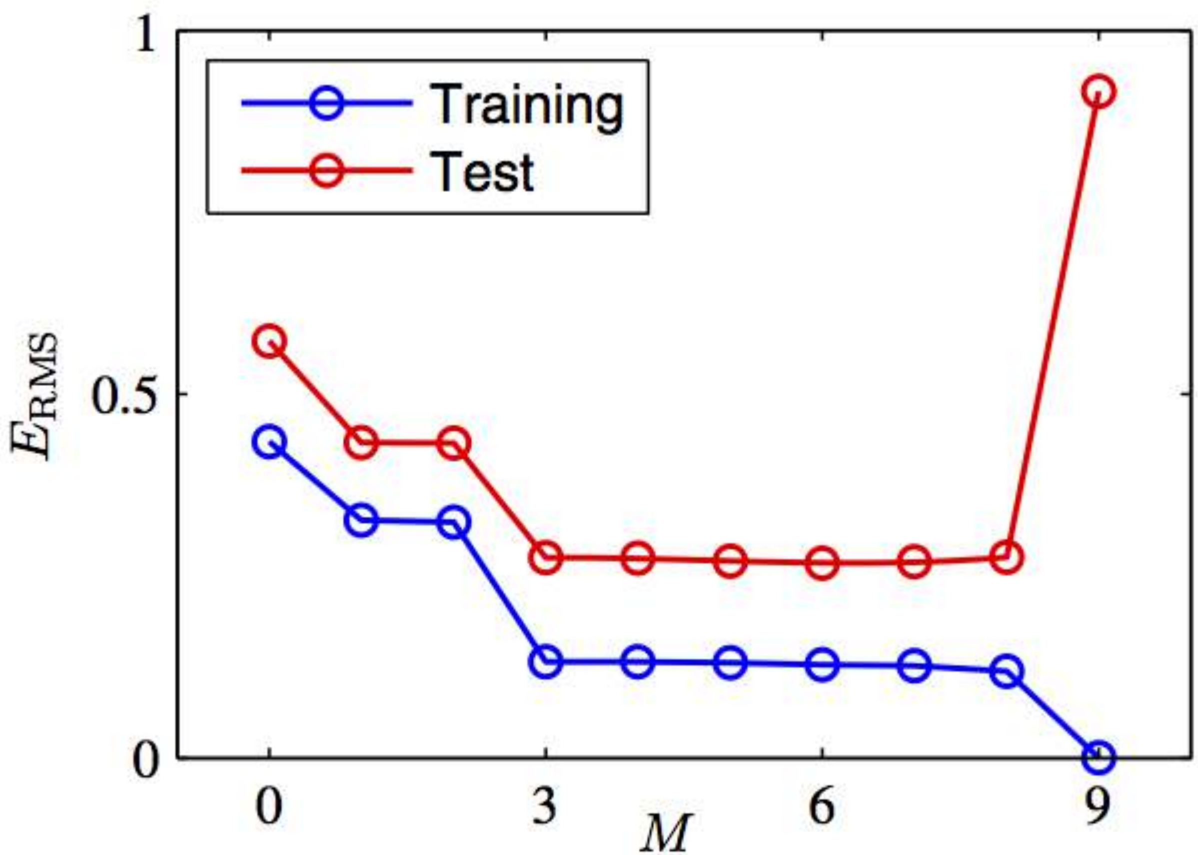
- Help the network **generalize** to data it hasn't seen.
- Big problem for **small datasets**.
- Overfitting example (a sine curve vs 9-degree polynomial):



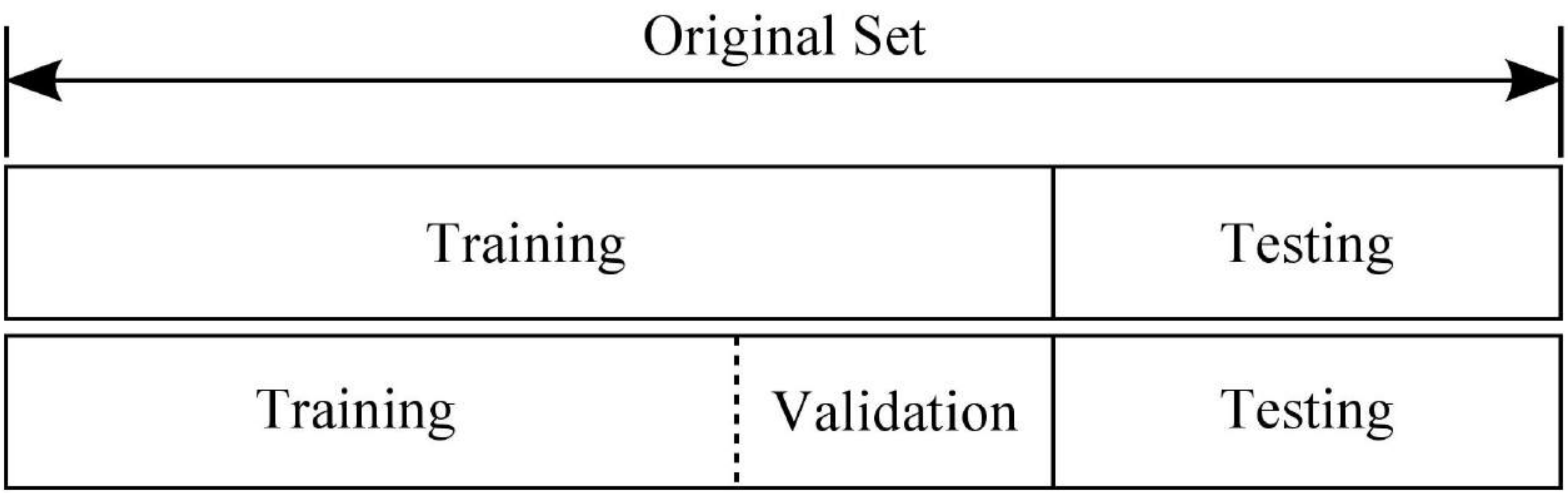
Key Concepts:

Overfitting and Regularization

- Overfitting: The error decreases in the training set but increases in the test set.



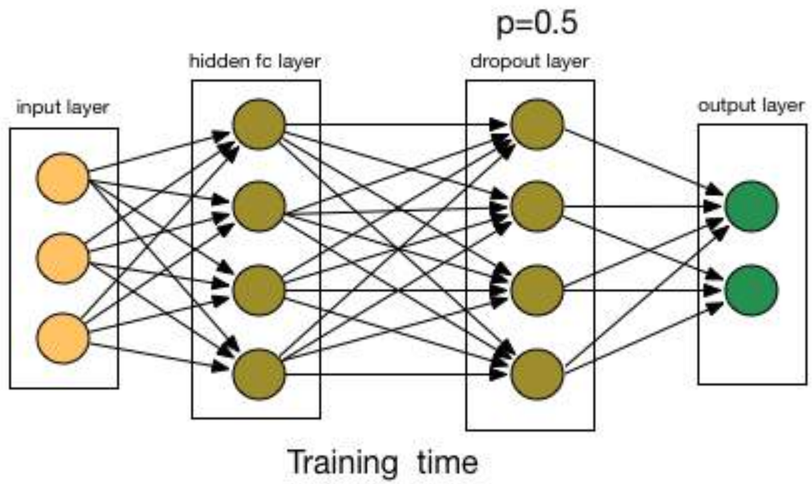
Regularization: Early Stoppage



- Create "validation" set (subset of the training set).
  - Validation set is assumed to be a representative of the testing set.
- **Early stoppage:** Stop training (or at least save a checkpoint) when performance on the validation set decreases

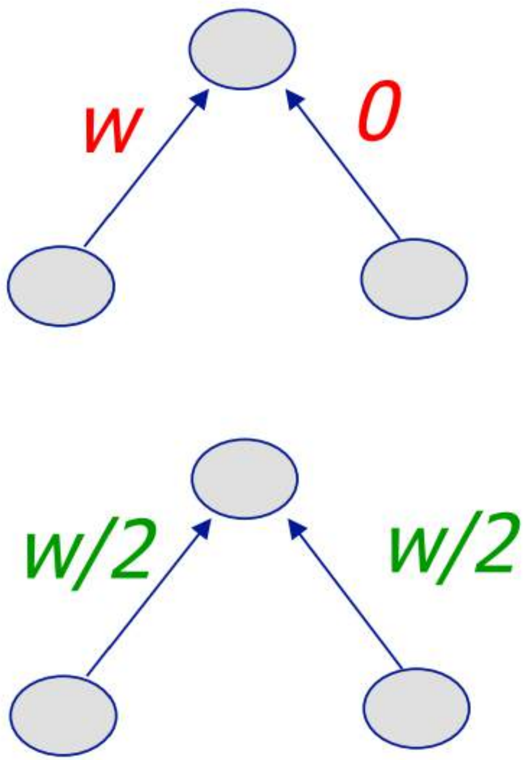
Key Concepts:

Regularization: Dropout



- **Dropout:** Randomly remove some nodes in the network (along with incoming and outgoing edges)
- Notes:
  - Usually *p >= 0.5* (*p* is probability of keeping node)
  - Input layers *p* should be much higher (and use noise instead of dropout)
  - Most deep learning frameworks come with a dropout layer

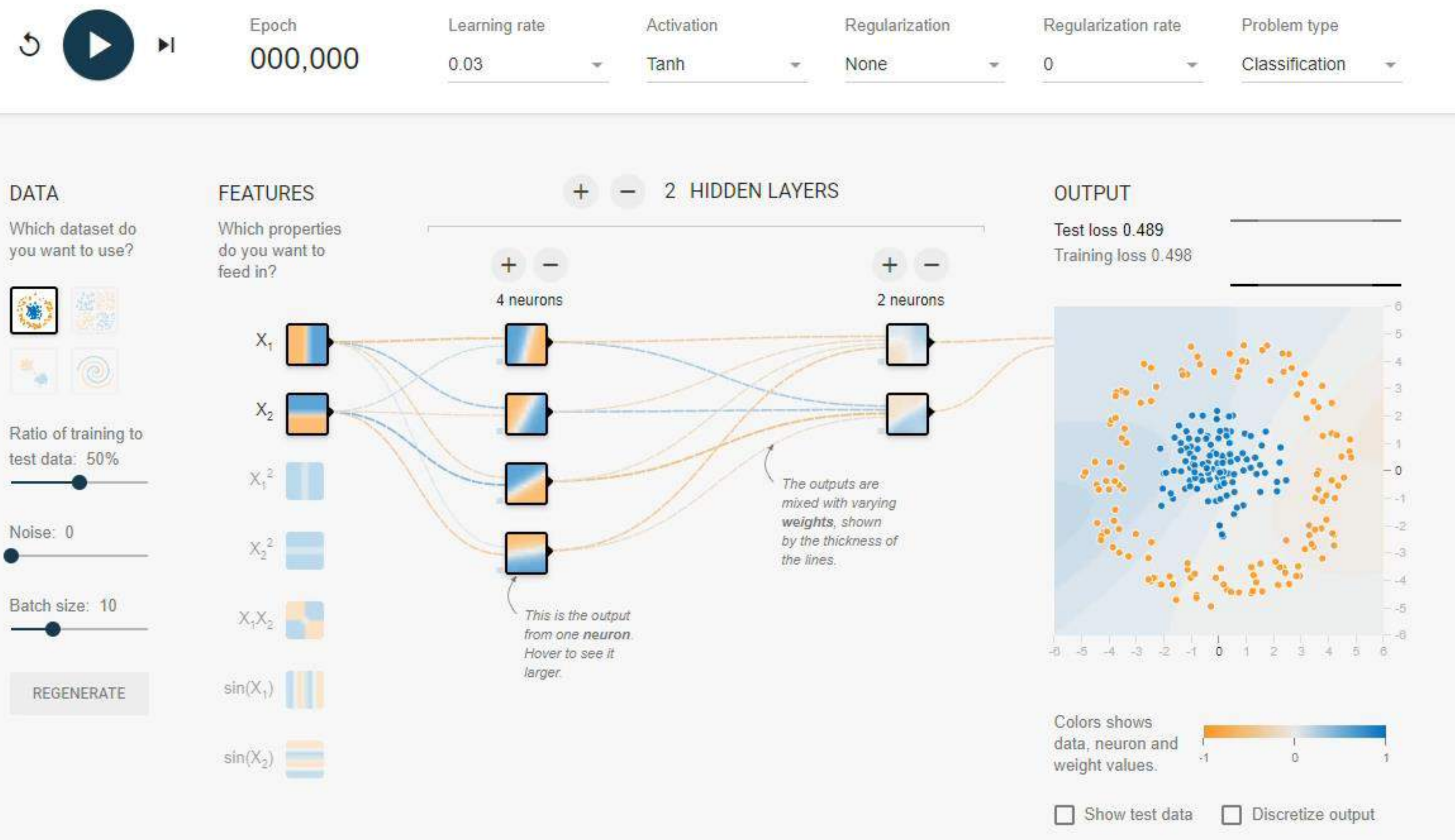
Regularization: Weight Penalty *(aka Weight Decay)*



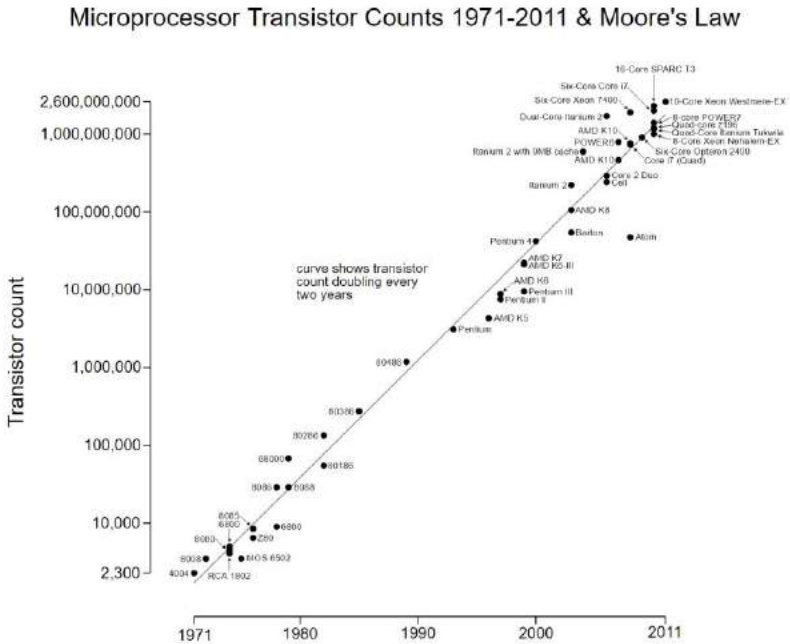
- **L2 Penalty:** Penalize squared weights. Result:
  - Keeps weight small unless error derivative is very large.
  - Prevent from fitting sampling error.
  - Smoother model (output changes slower as the input change).
  - If network has two similar inputs, it prefers to put half the weight on each rather than all the weight on one.
- **L1 Penalty:** Penalize absolute weights. Result:
  - Allow for a few weights to remain large.

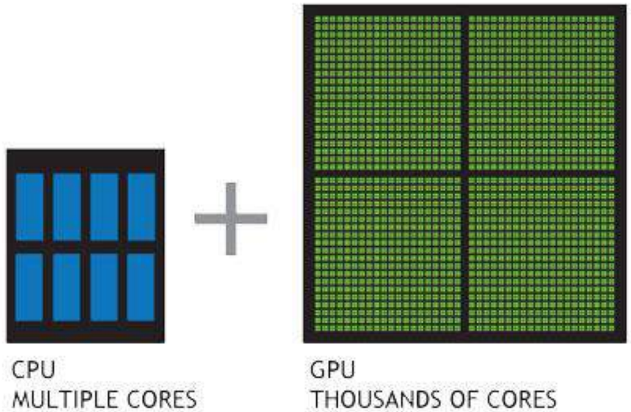
Neural Network Playground

http://playground.tensorflow.org



Deep Learning Breakthroughs: What Changed?

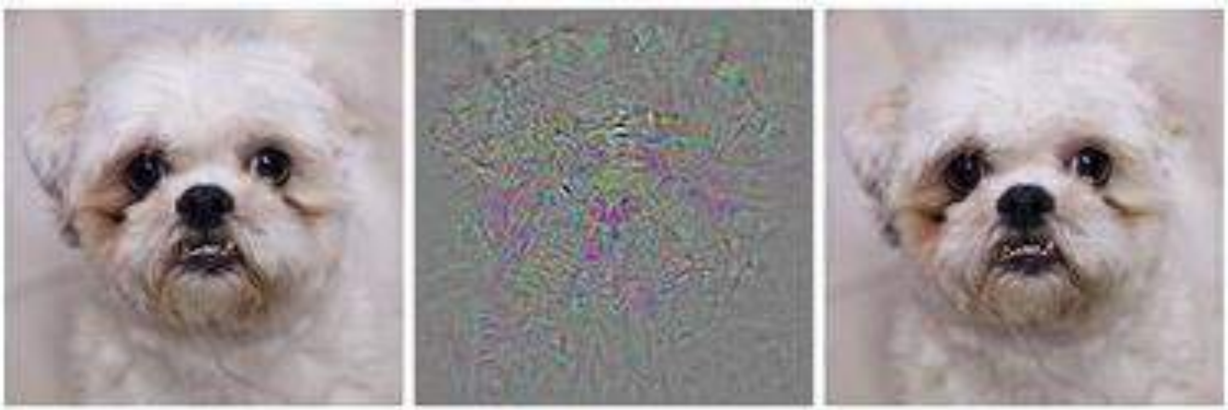




- **Compute**
  CPUs, GPUs, ASICs
- **Organized large(-ish) datasets**
  Imagenet
- **Algorithms and research:**
  Backprop, CNN, LSTM
- **Software and Infrastructure**
  Git, ROS, PR2, AWS, Amazon Mechanical Turk, TensorFlow, …
- **Financial backing of large companies**
  Google, Facebook, Amazon, …

Deep Learning:
Our intuition about what's "hard" is flawed (in complicated ways)

Visual perception: 540,000,000 years of data
Bipedal movement: 230,000,000 years of data
Abstract thought: 100,000 years of data

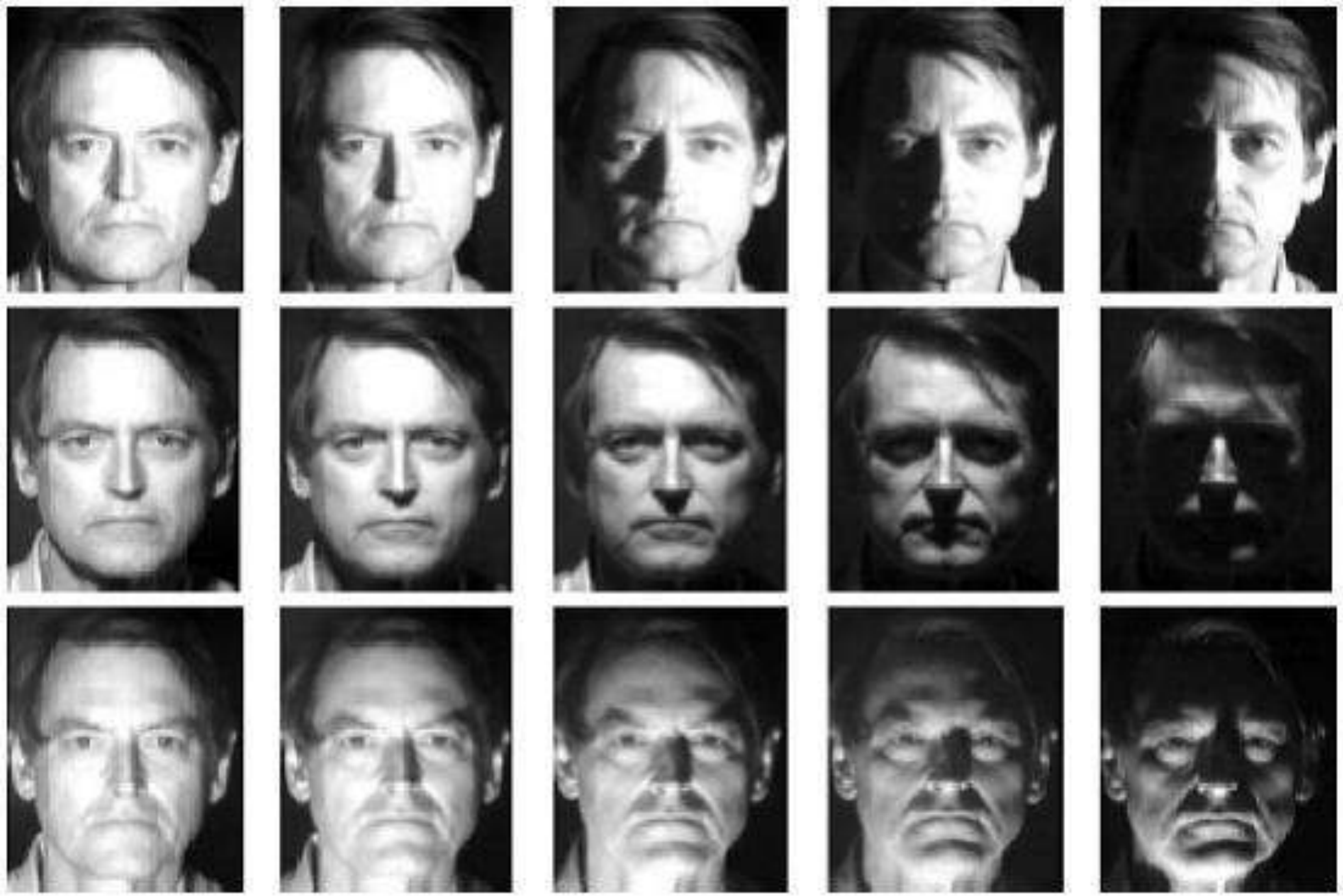


Prediction: **Dog** + Distortion Prediction: **Ostrich**

"Encoded in the large, highly evolve sensory and motor portions of the human brain is a **billion years of experience** about the nature of the world and how to survive in it.… Abstract thought, though, is a new trick, perhaps less than **100 thousand years** old. We have not yet mastered it. It is not all that intrinsically difficult; it just seems so when we do it."
- Hans Moravec, Mind Children (1988)

Deep Learning is Hard:

Illumination Variability



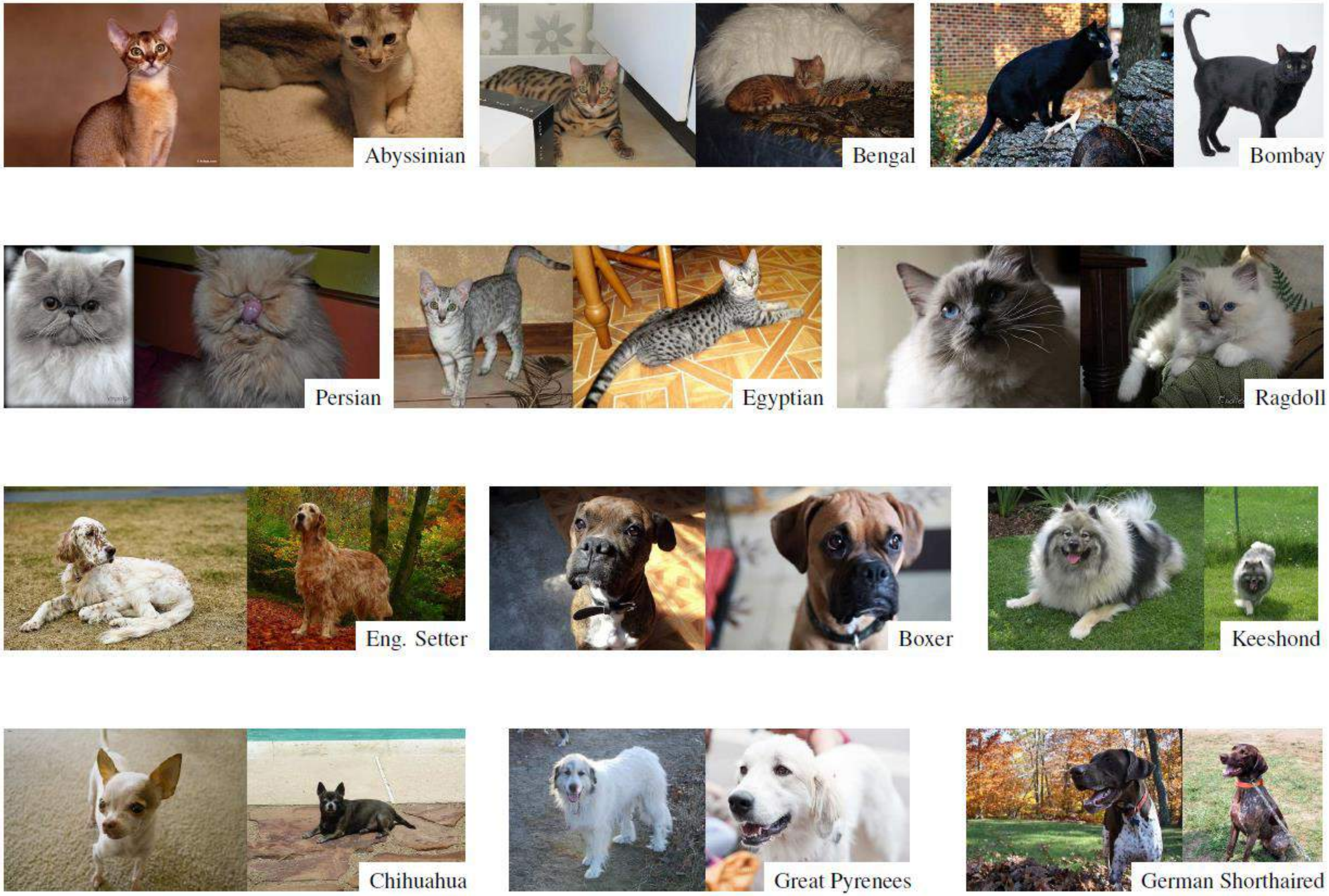
Deep Learning is Hard:

Pose Variability and Occlusions

Figure 1. **The deformable and truncated cat.** Cats exhibit (al-

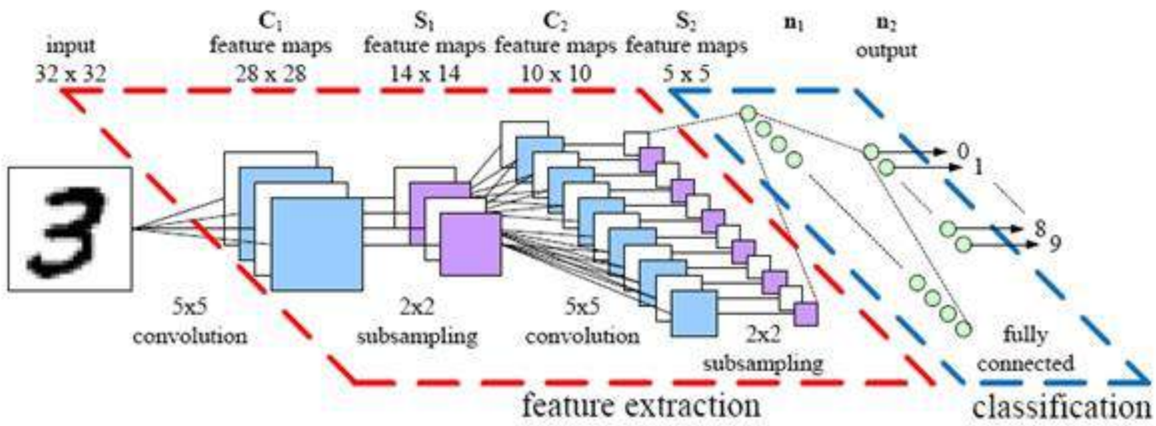
Parkhi et al. "The truth about cats and dogs.“ 2011.

Deep Learning is Hard:
Intra-Class Variability



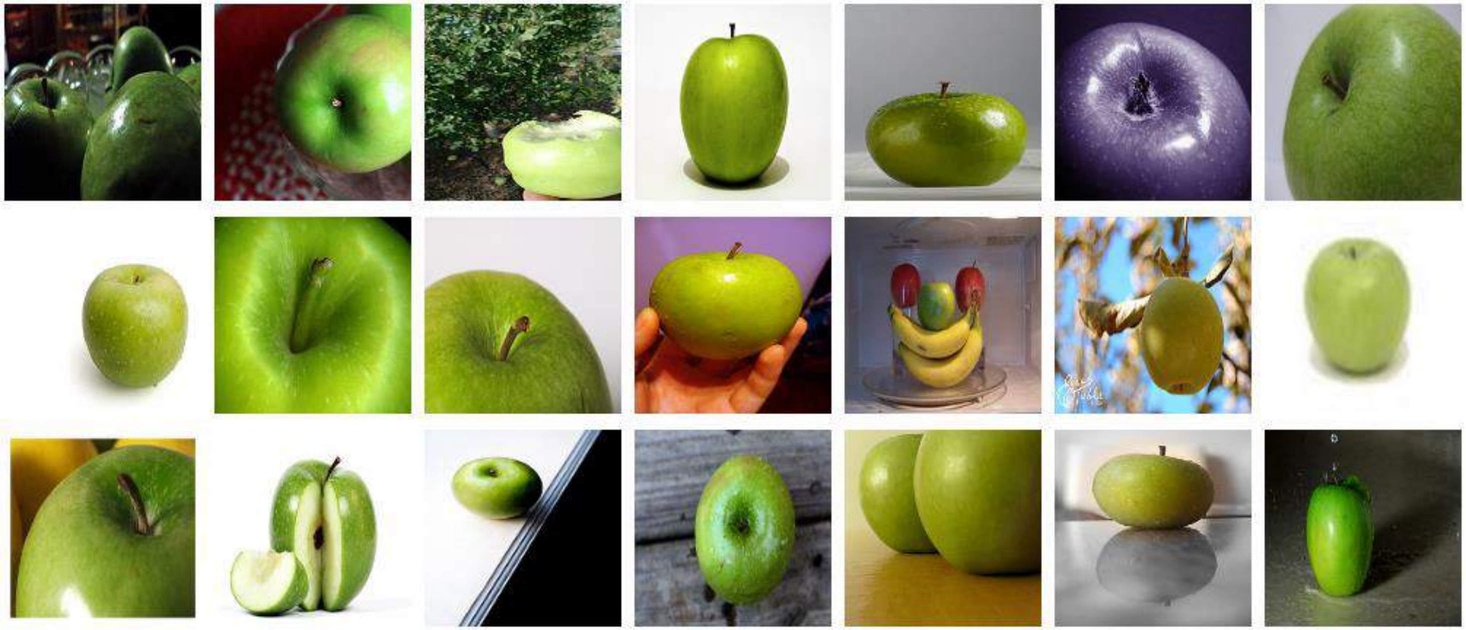
Parkhi et al. "Cats and dogs." 2012.

Object Recognition / Classification



What is ImageNet?

- **ImageNet:** dataset of 14+ million images (21,841 categories)
- Let's take the high level category of **fruit** as an example:
    - Total 188,000 images of fruit
    - There are 1206 Granny Smith apples:



What is ImageNet?

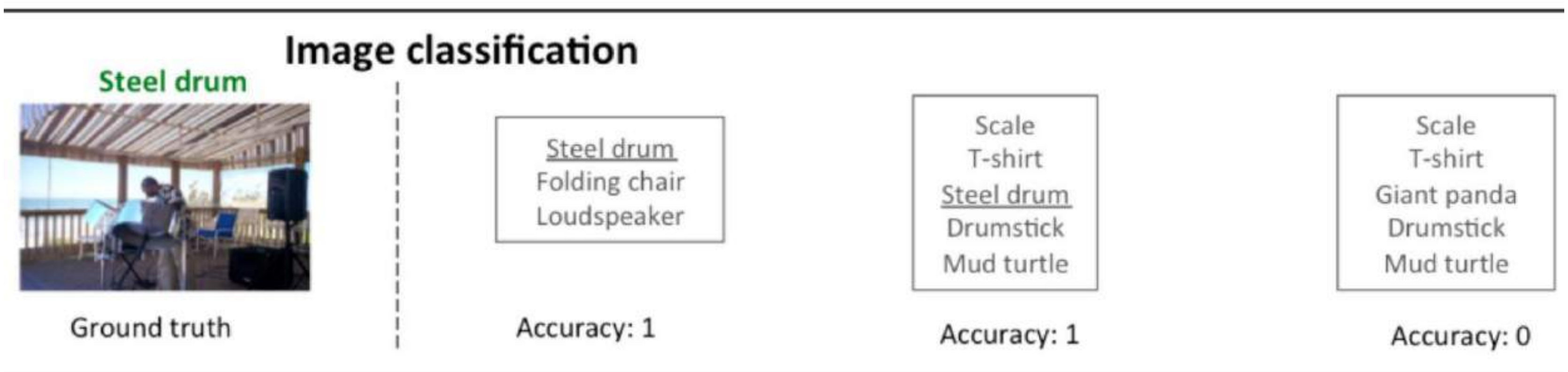
Dataset →
- **ImageNet:** dataset of 14+ million images

Competition →
- **ILSVRC:** ImageNet Large Scale Visual Recognition Challenge

Networks →
- AlexNet (2012)
- ZFNet (2013)
- VGGNet (2014)
- GoogLeNet (2014)
- ResNet (2015)
- CUImage (2016)
- Squeeze-and-Excitation Networks (2017)

ILSVRC Challenge Evaluation for Classification

- Top 5 error rate:
  - You get 5 guesses to get the correct label

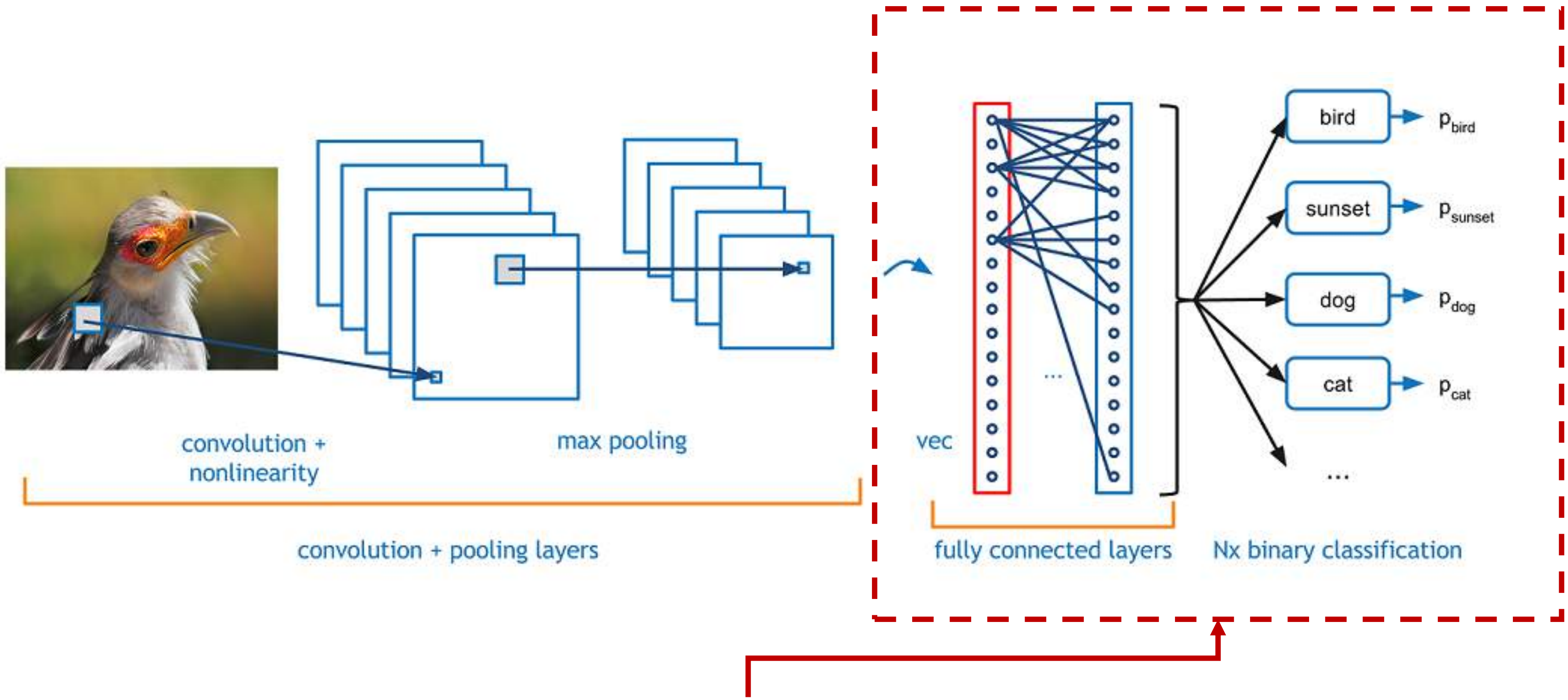


- ~20% reduction in accuracy for Top 1 vs Top 5
- Human annotation is a binary task: “apple” or “not apple”

- Human error: 5.1%
  - Surpassed in 2015
- **2018:** ImageNet Challenge moves to Kaggle

- **AlexNet (2012): First CNN (15.4%)**
  - 8 layers
  - 61 million parameters
- **ZFNet (2013): 15.4% to 11.2%**
  - 8 layers
  - More filters. Denser stride.
- **VGGNet (2014): 11.2% to 7.3%**
  - Beautifully uniform:
    3x3 conv, stride 1, pad 1, 2x2 max pool
  - 16 layers
  - 138 million parameters
- **GoogLeNet (2014): 11.2% to 6.7%**
  - Inception modules
  - 22 layers
  - 5 million parameters
    (throw away fully connected layers)
- **ResNet (2015): 6.7% to 3.57%**
  - More layers = better performance
  - 152 layers
- **CUImage (2016): 3.57% to 2.99%**
  - Ensemble of 6 models
- **SENet** (2017): 2.99% to 2.251%
  - Squeeze and excitation block: network is allowed to adaptively adjust the weighting of each feature map in the convolutional block.

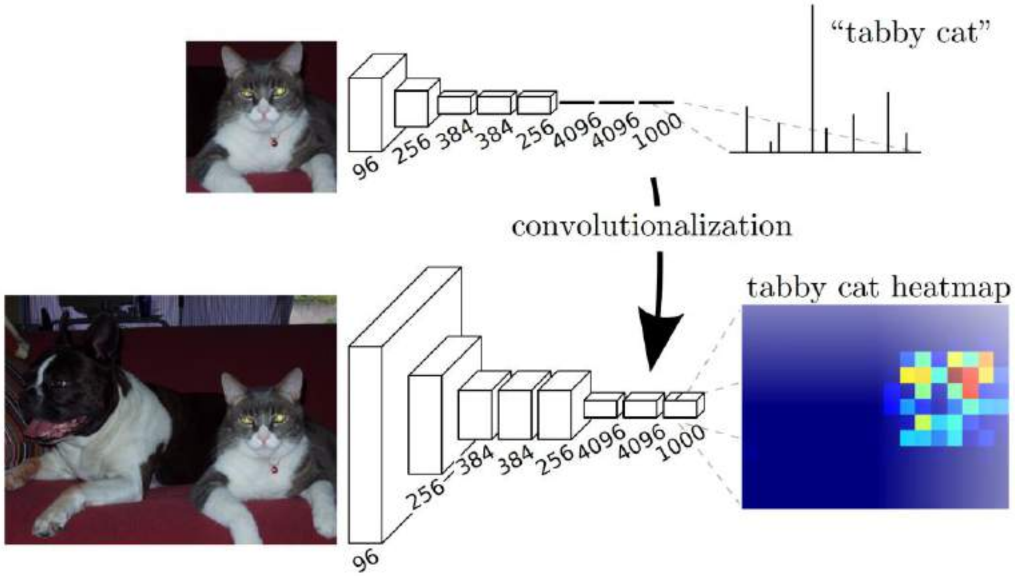
Same Architecture, Many Applications



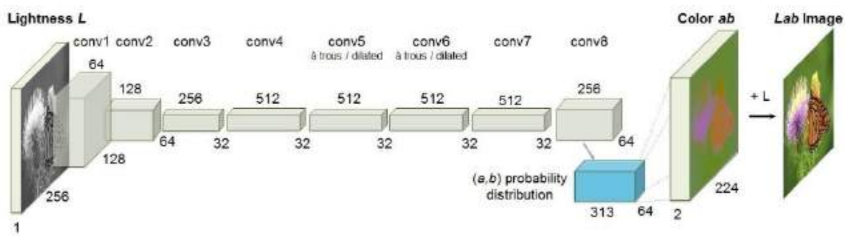
This part might look different for:

- Different image classification domains
- Image captioning with recurrent neural networks
- Image object localization with bounding box
- Image segmentation with fully convolutional networks
- Image segmentation with deconvolution layers

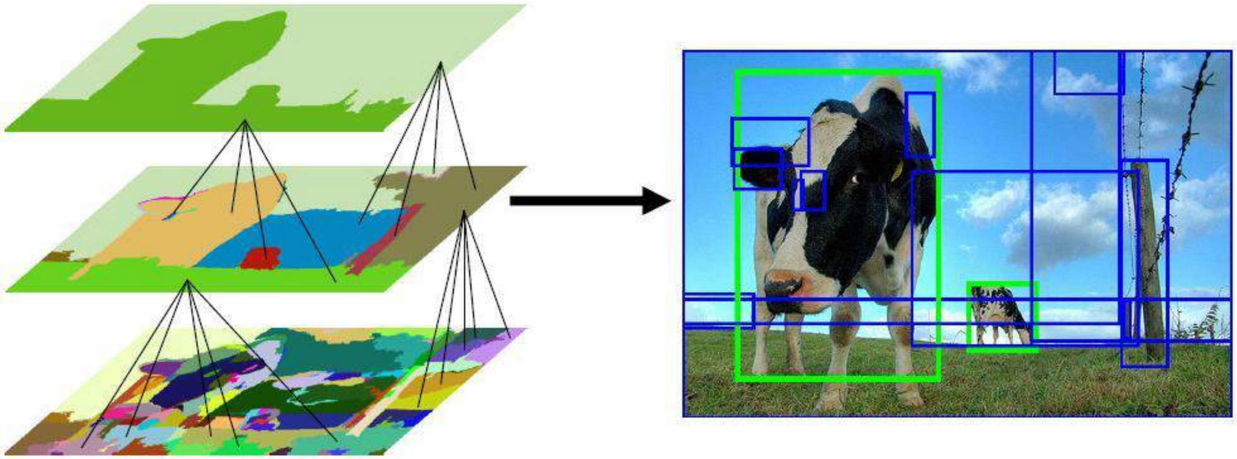
Pixel-Level Full Scene Segmentation

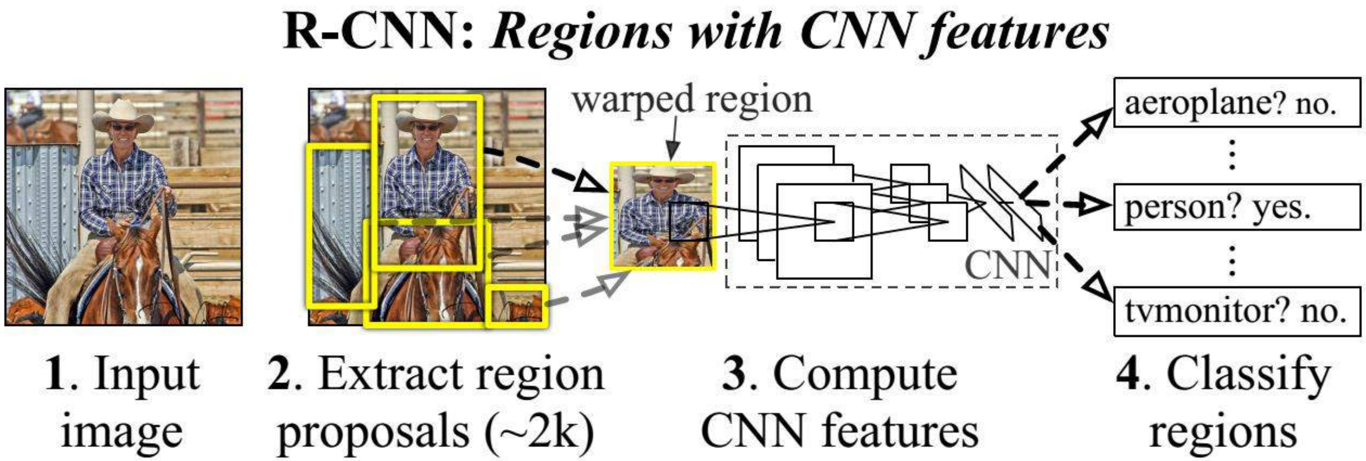


Colorization of Images

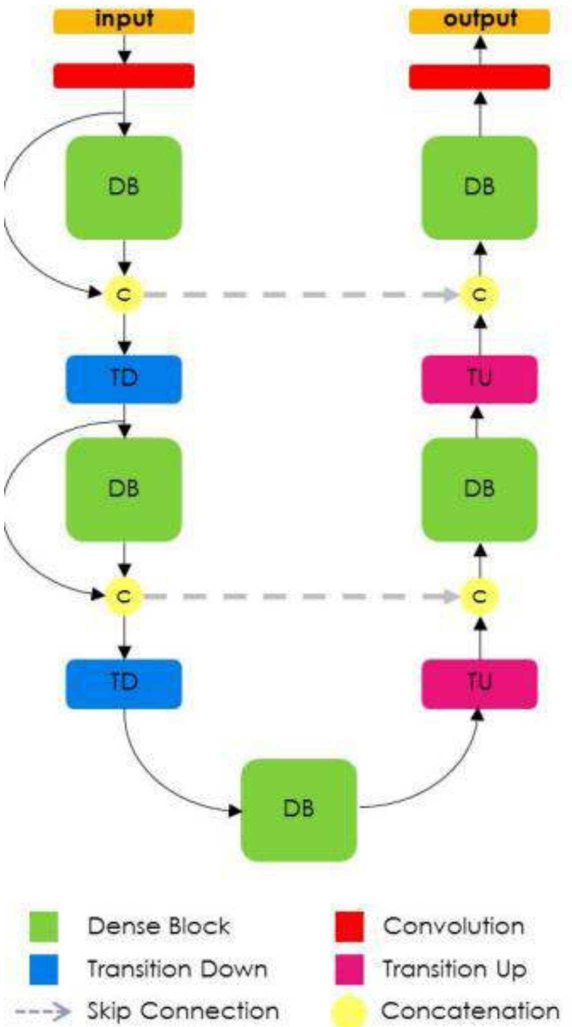


Object Detection



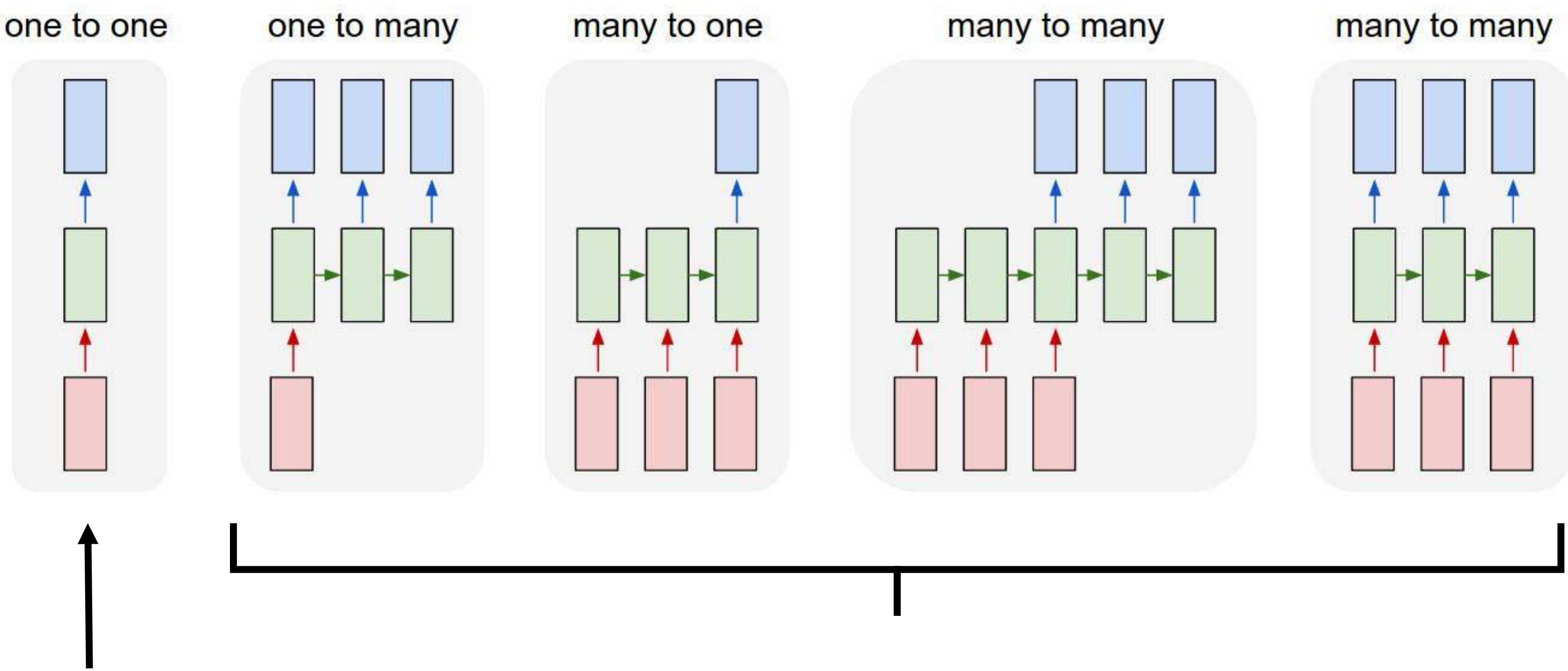


Background Removal (2017)



pix2pixHD: generate high-resolution photo-realistic images from semantic label maps (2017)

Flavors of Neural Networks



"Vanilla" Neural Networks

Recurrent Neural Networks

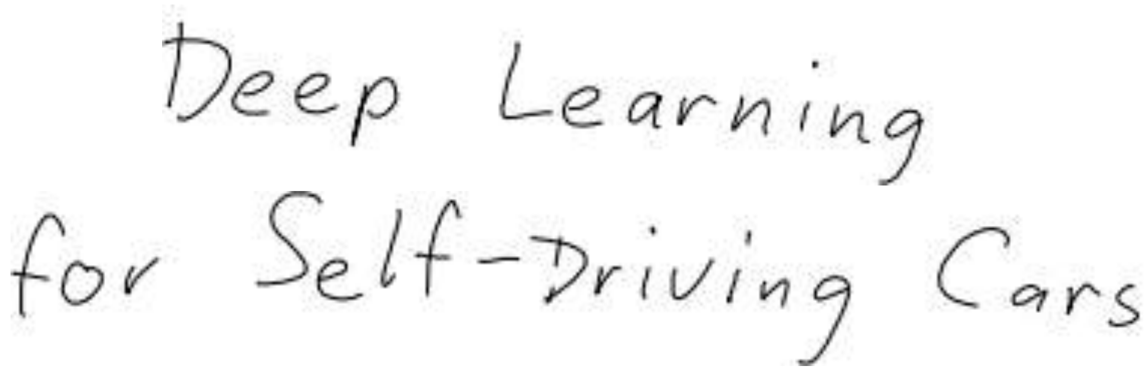
Handwriting Generation from Text

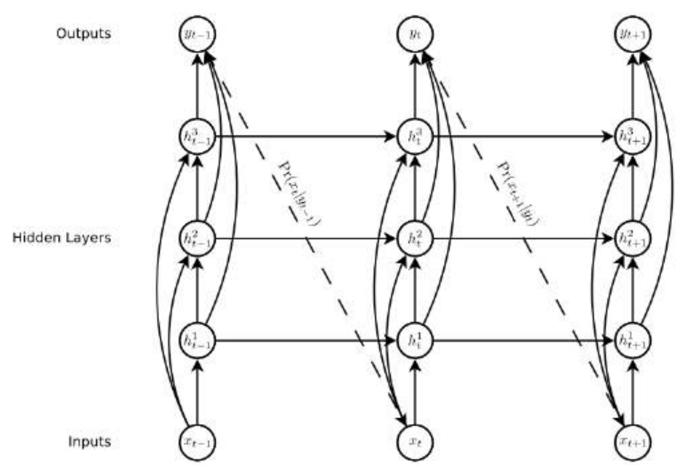
Input:

Text --- up to 100 characters, lower case letters work best

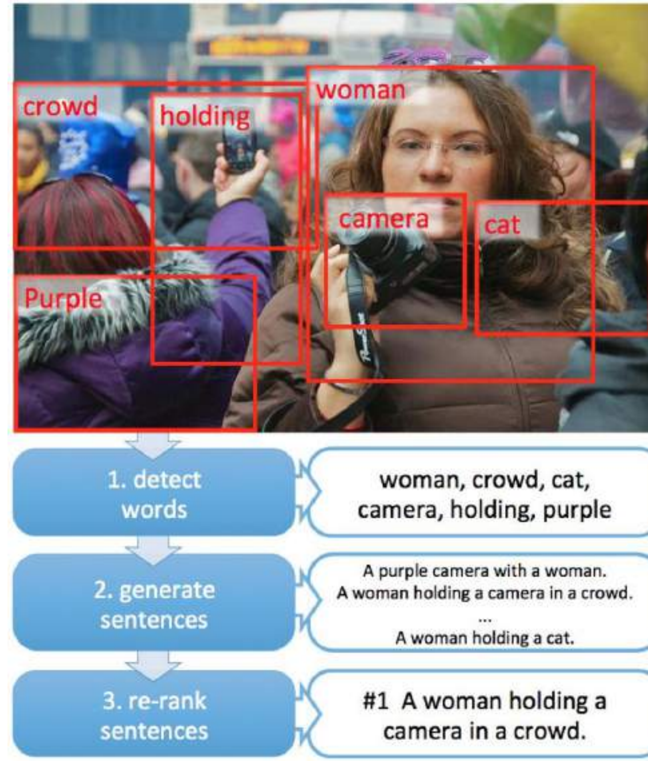
Deep Learning for Self Driving Cars

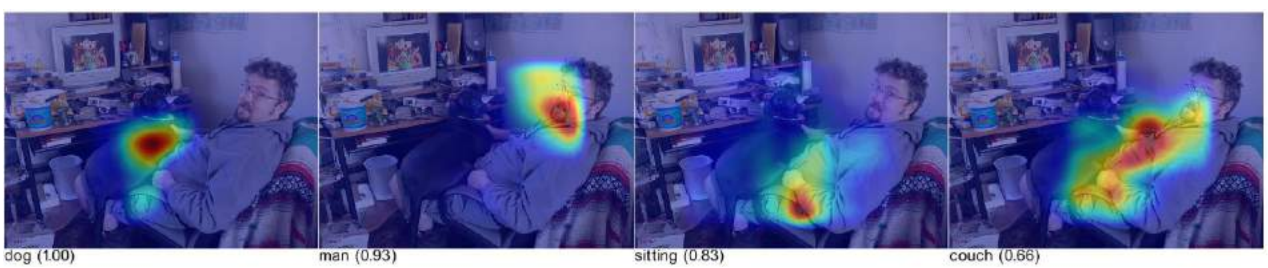
Output:





Applications: Image Caption Generation

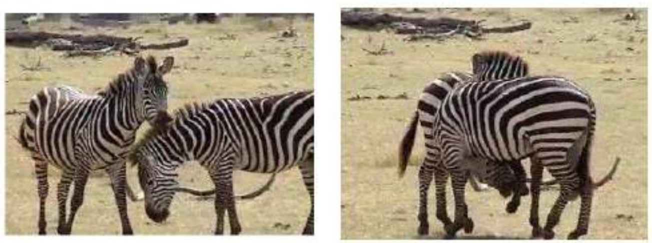




Video Description Generation

Correct descriptions.

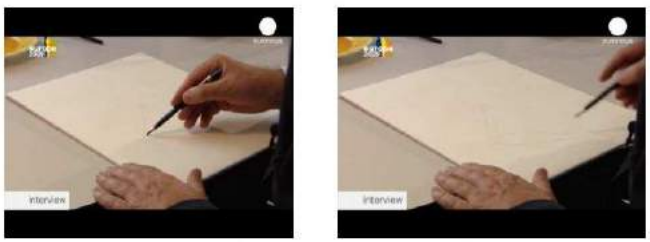
S2VT: A man is doing stunts on his bike.



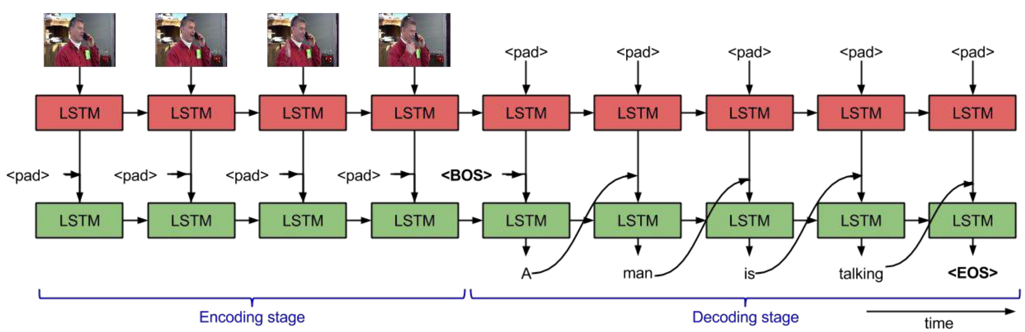
S2VT: A herd of zebras are walking in a field.

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.



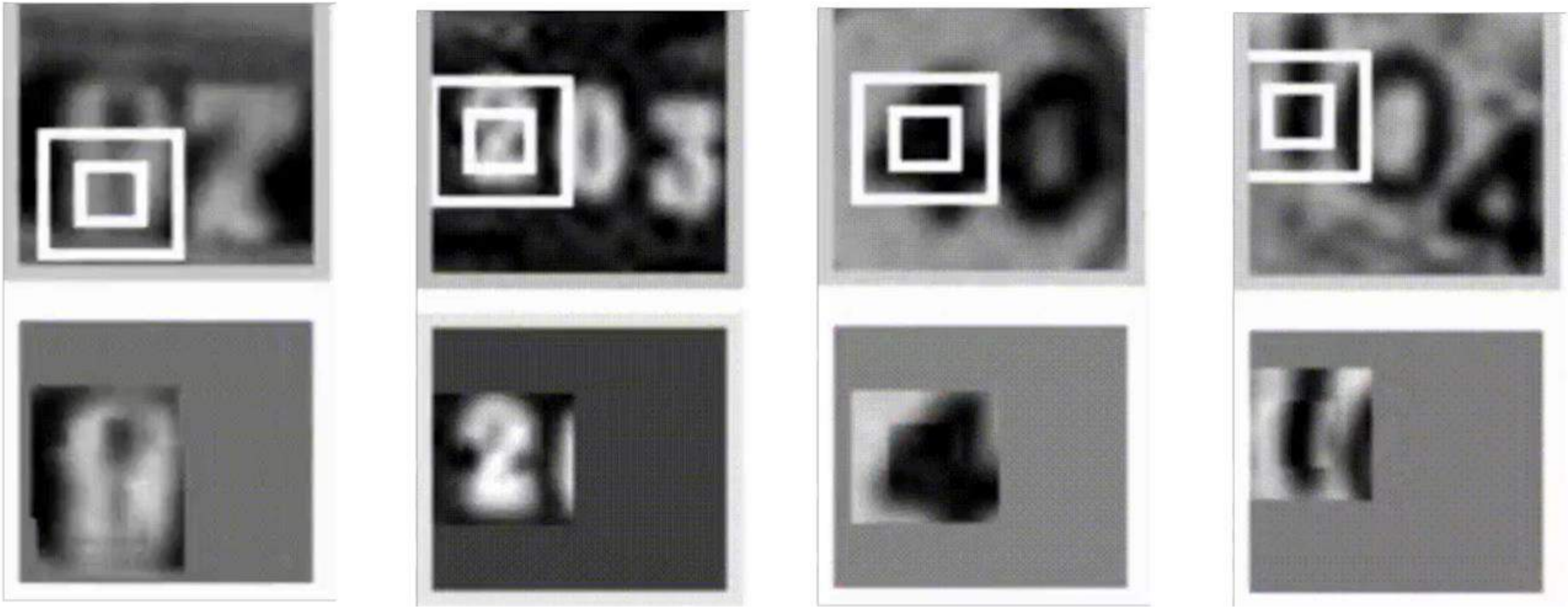
S2VT: A man is cutting a piece of a pair of a paper.

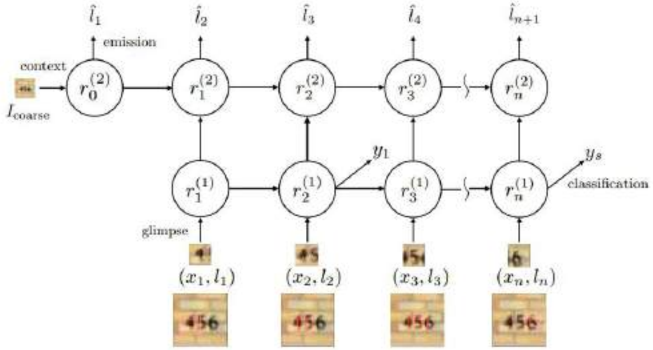


Venugopalan et al.
"Sequence to sequence-video to text." 2015.

Code: https://vsubhashini.github.io/s2vt.html

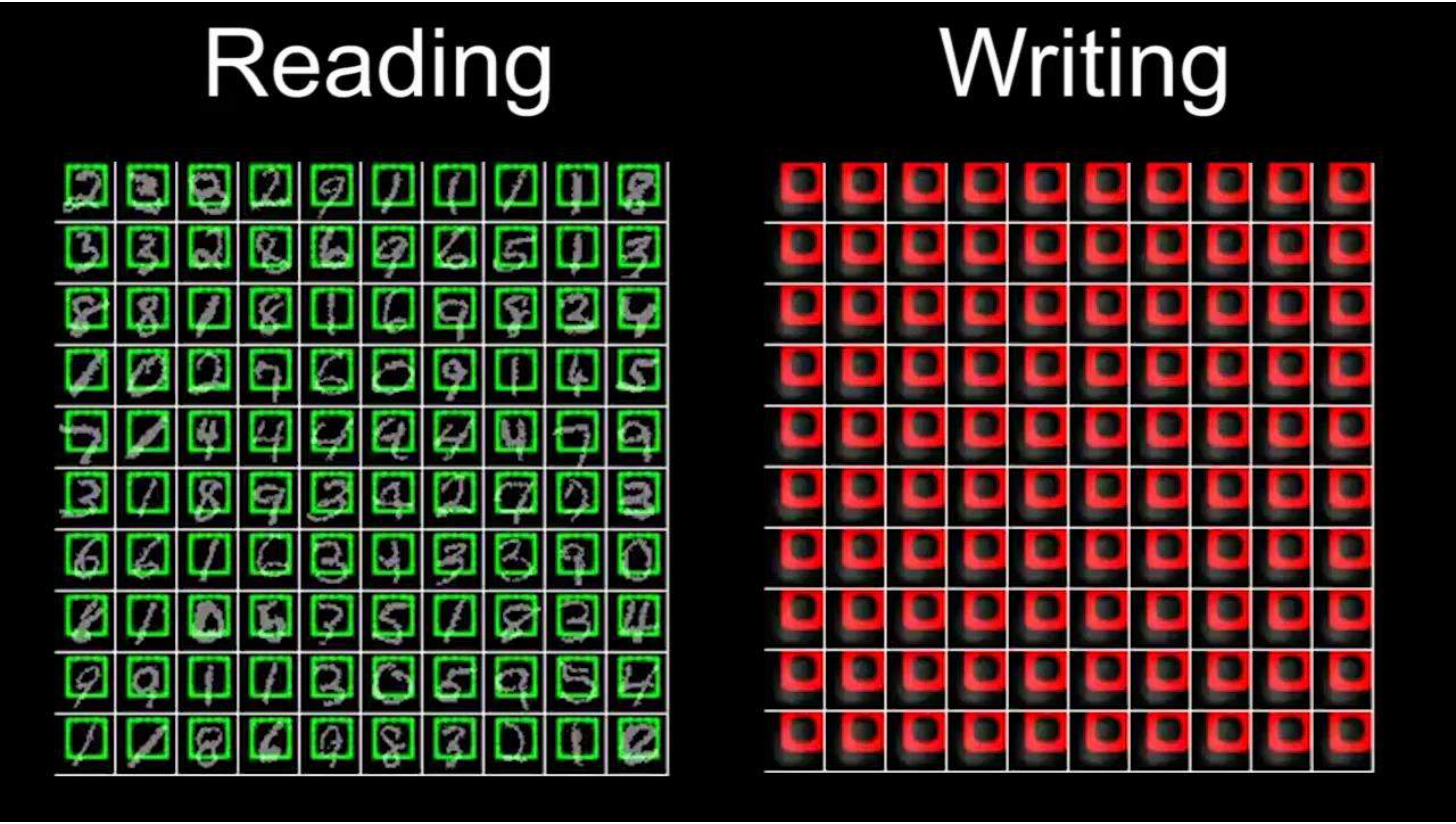
Modeling Attention Steering





Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. **"Multiple object recognition with visual attention."** (2014).

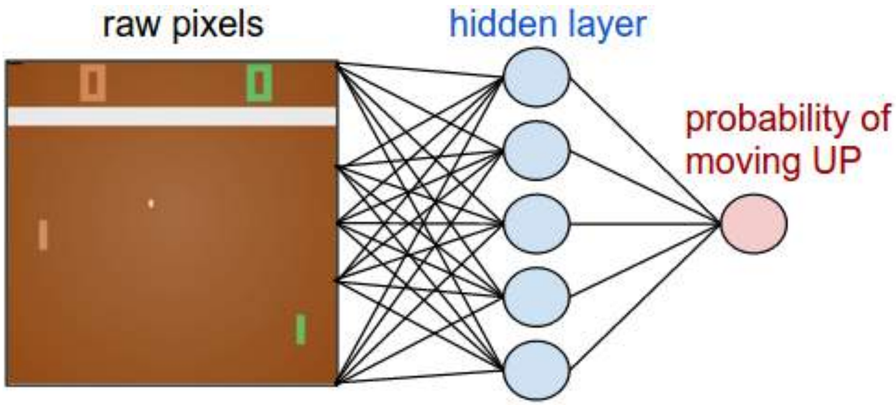
Drawing with Selective Attention



Gregor et al. "DRAW: A recurrent neural network for image generation." (2015). Code: https://github.com/ericjang/draw

(Toward) General Purpose Intelligence:
Pong to Pixels

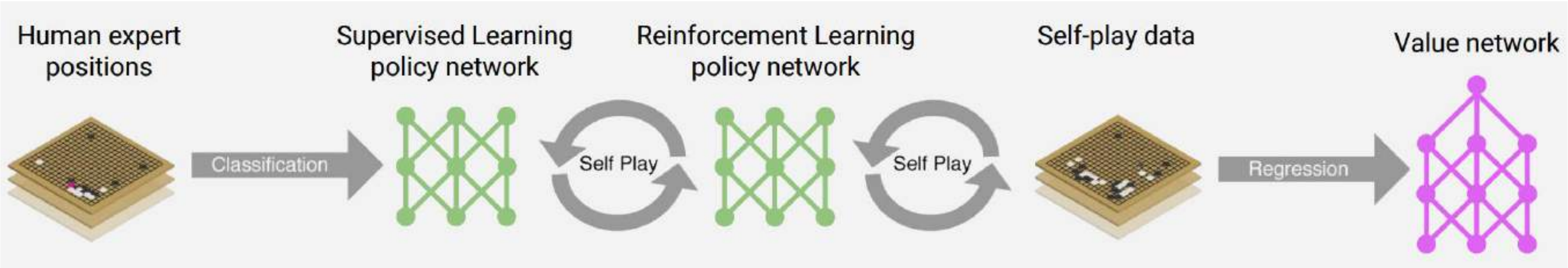
Policy Network:

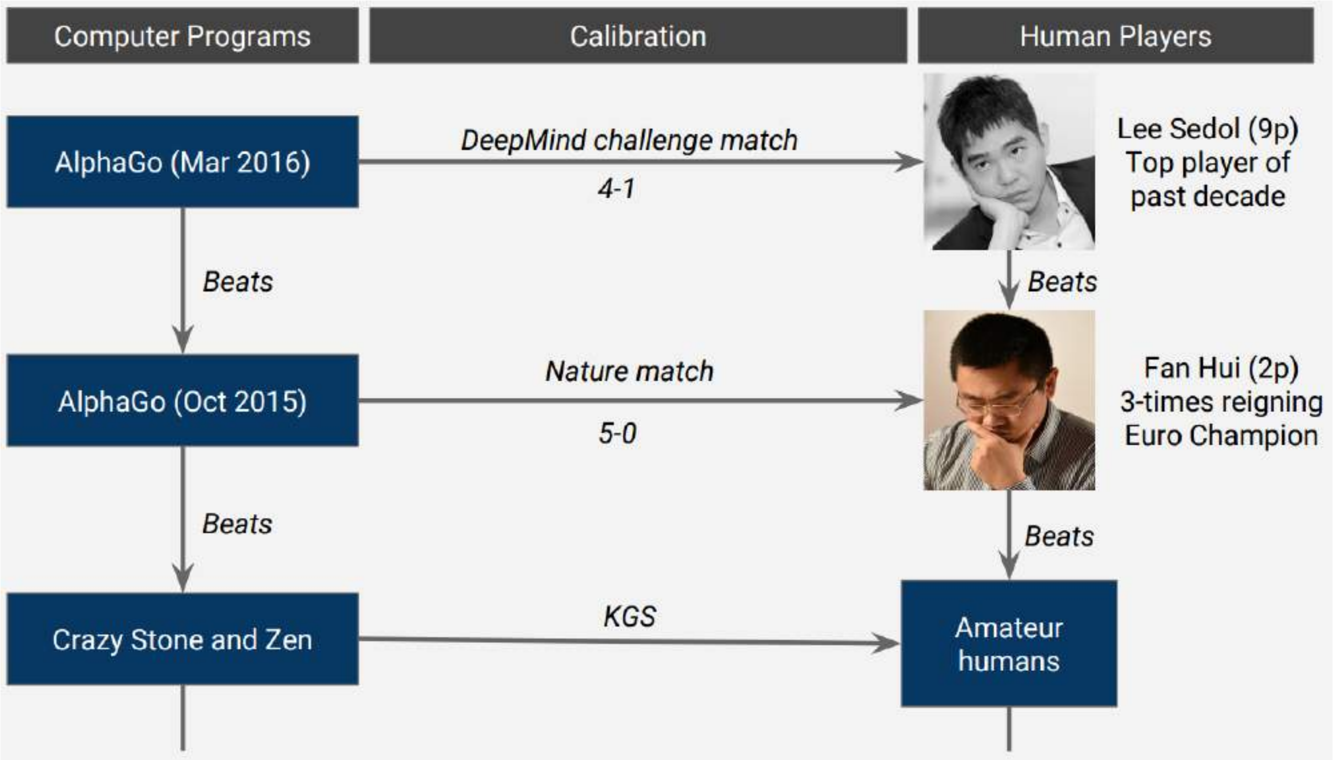


- 80x80 image (difference image)
- 2 actions: up or down
- 200,000 Pong games

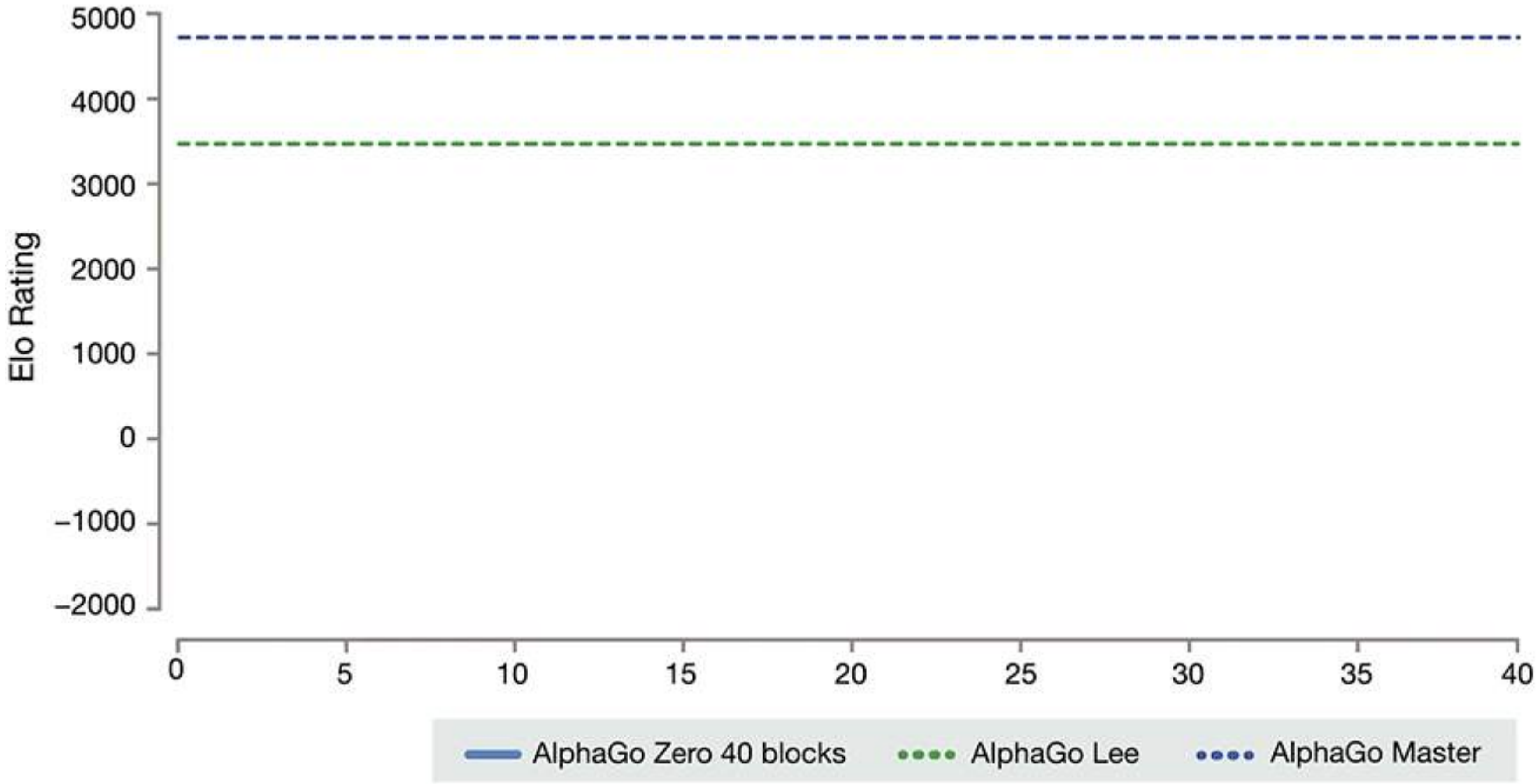
This is a step towards general purpose artificial intelligence!

AlphaGo (2016) Beat Top Human at **Go**



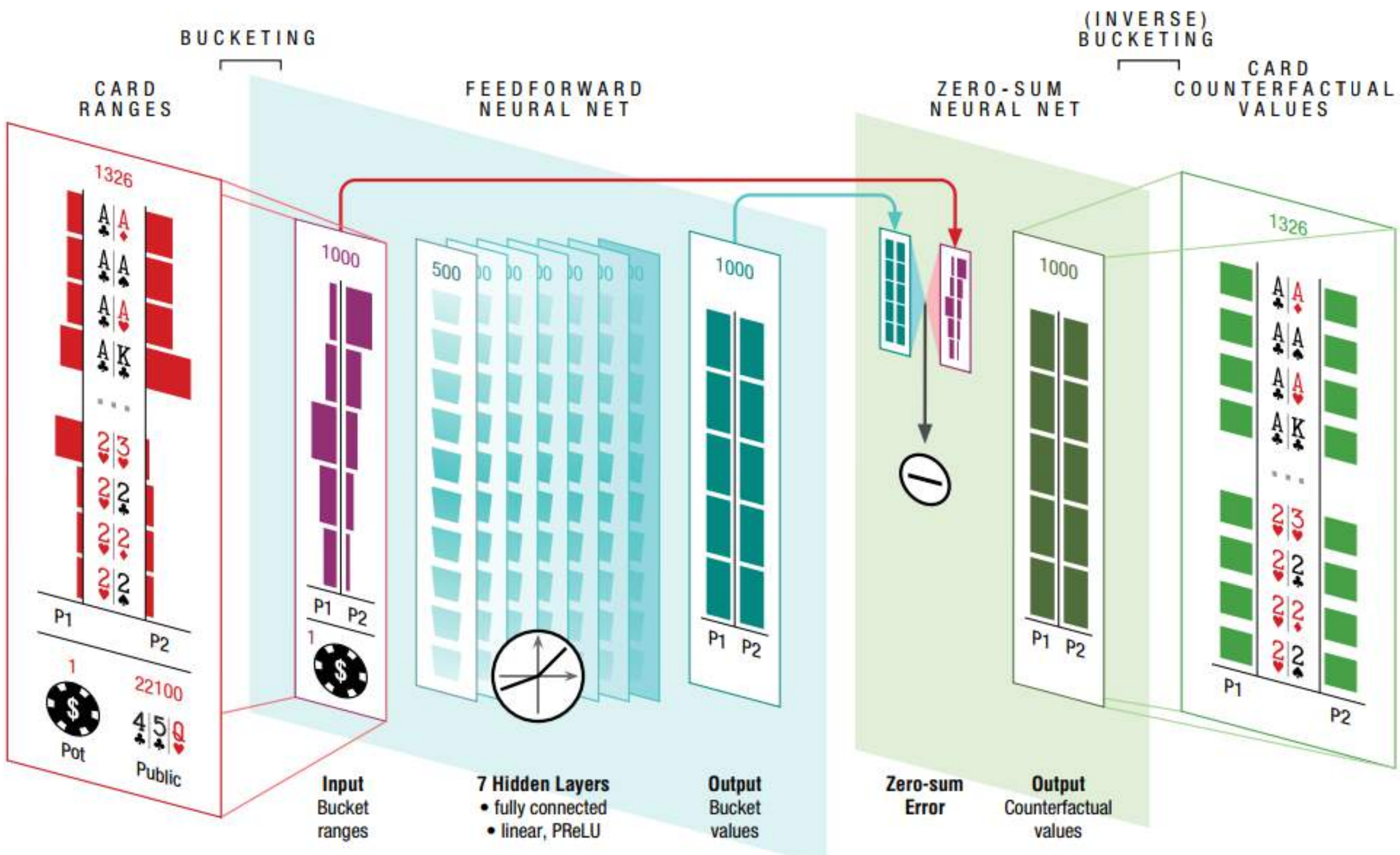


AlphaGo Zero (2017): Beats AlphaGo



DeepStack first to beat professional poker players (2017)

(in heads-up poker)

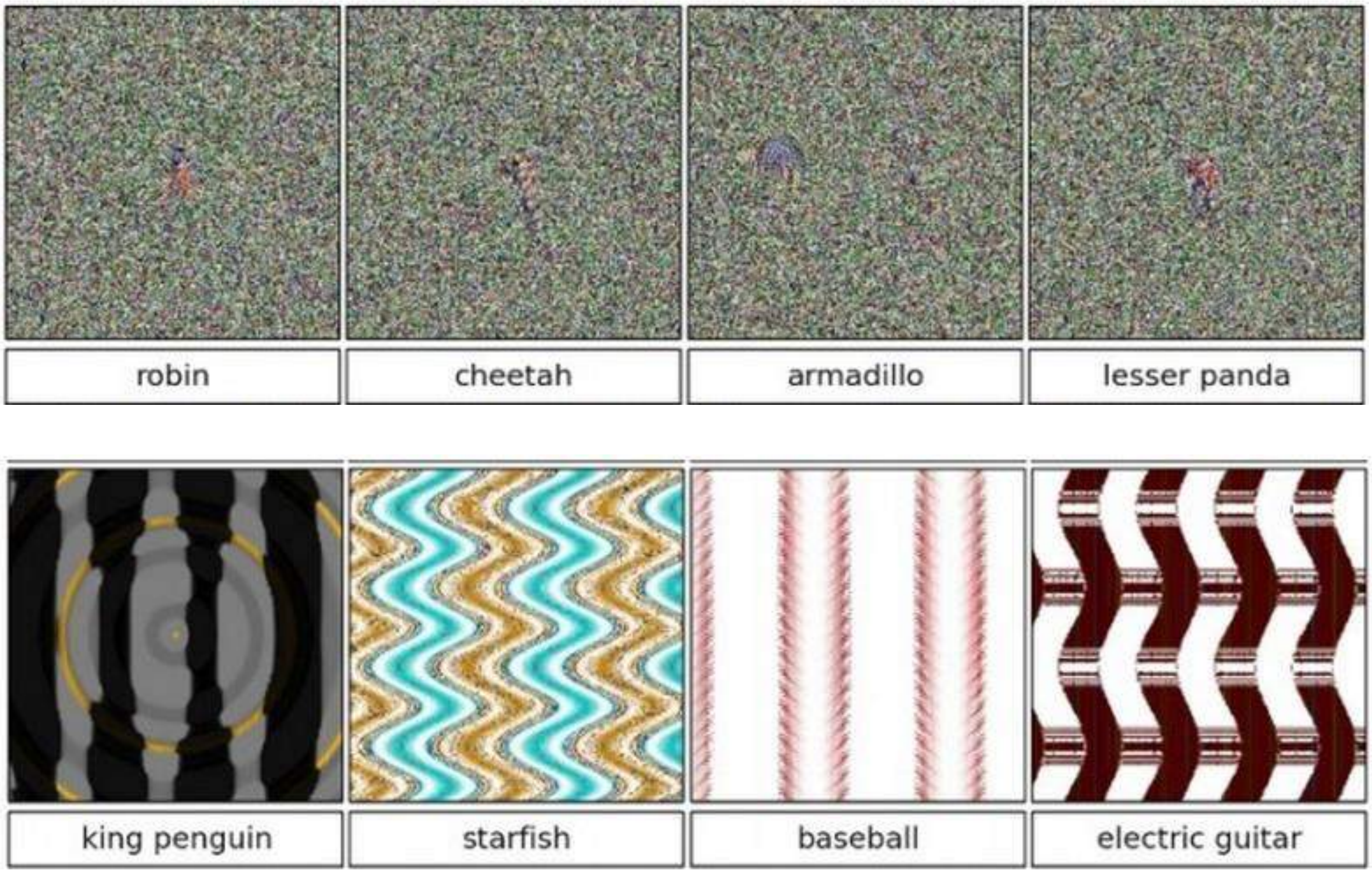


Current Drawbacks

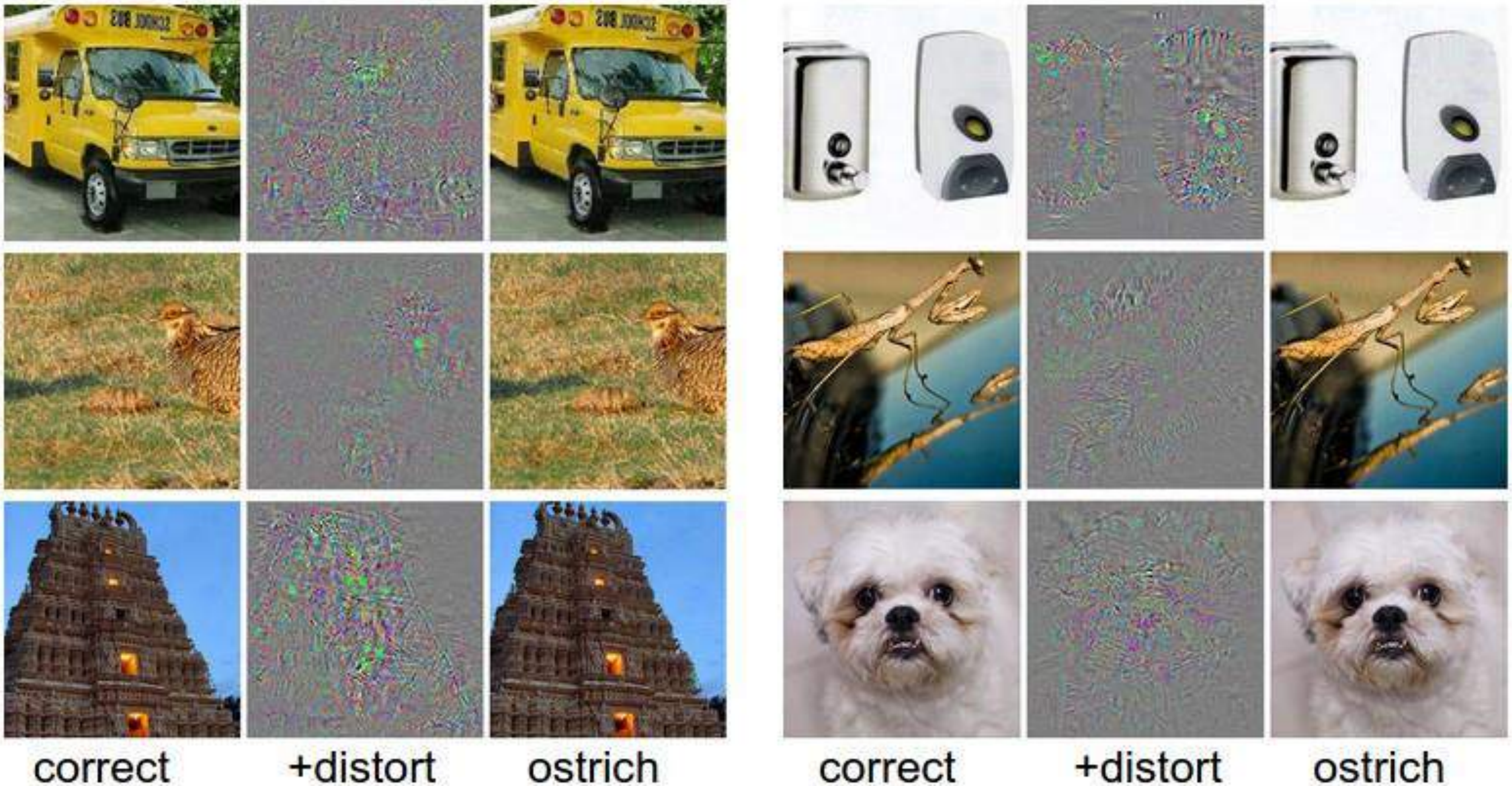
Defining a good reward function is difficult… **Coast Runners:** Discovers local pockets of high reward ignoring the "implied" bigger picture goal of finishing the race.

In addition, specifying a reward function for self-driving cars raises ethical questions…

Robustness:
>99.6% Confidence in the Wrong Answer



Robustness:
Fooled by a Little Distortion



Current Challenges

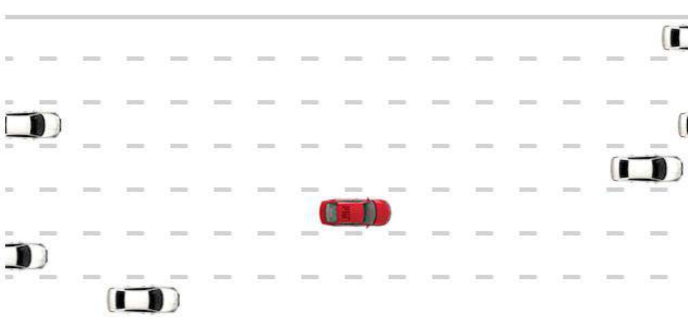
- **Transfer learning:** Unable to transfer representation to most reasonably related domains except in specialized formulations.
    - **Understanding:** Lacks "reasoning" or ability to truly derive "understanding" as previously defined on anything but specialized problem formulations. (Definition used: Ability to turn complex information to into simple, useful information.)
- Requires **big** data: inefficient at learning from data
- Requires **supervised** data: costly to annotate real-world data
- **Not fully automated:** Needs hyperparameter tuning for training: learning rate, loss function, mini-batch size, training iterations, momentum, optimizer selection, etc.
- **Reward:** Defining a good reward function is difficult.
- **Transparency:** Neural networks are for the most part black boxes (for real-world applications) even with tools that visualize various aspects of their operation.
- **Edge cases:** Deep learning is not good at dealing with edge cases.

Why Deep Learning?

Deep Learning:
Learn effective perception-control from **data**

Deep Learning:
Learn effective human-robot interaction from **data**

Solve the perception-control problem where **possible**:



And where **not possible**: involve the human

Thank You

NVIDIA

Google

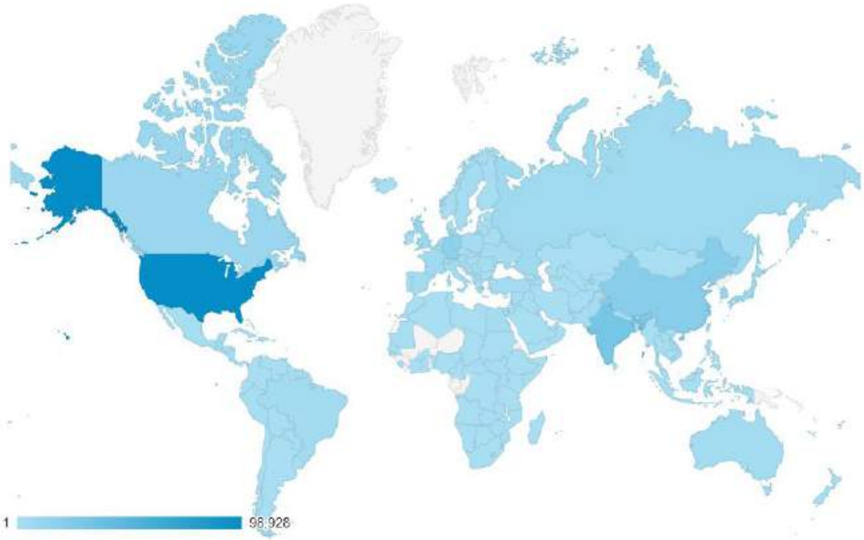
Autoliv

amazon alexa

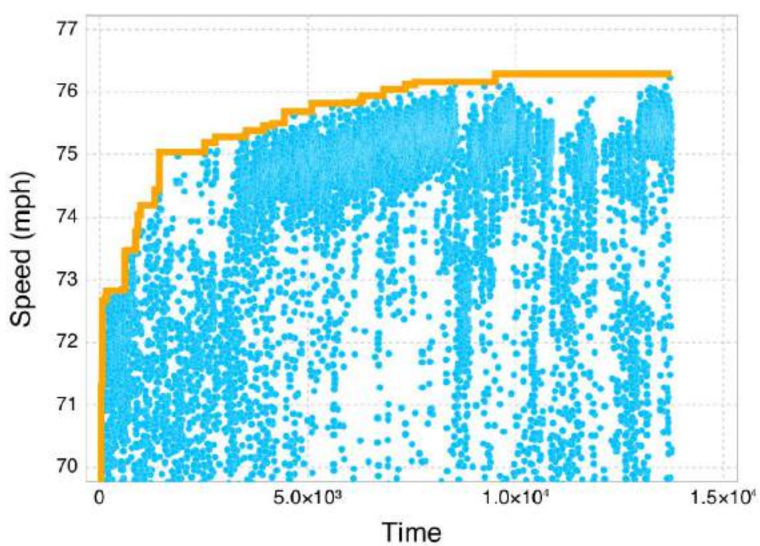
auto

CSRC
Collaborative Safety Research Center
TOYOTA

Thank You



	Country	Sessions	% Sessions
1.	United States	98,928	32.89%
2.	India	29,352	9.76%
3.	China	20,407	6.78%
4.	Germany	15,718	5.23%
5.	South Korea	10,493	3.49%
6.	Canada	8,728	2.90%
7.	United Kingdom	8,717	2.90%
8.	Japan	7,543	2.51%
9.	Russia	6,594	2.19%
10.	Taiwan	6,353	2.11%



Next lecture: ***Self-Driving Cars***