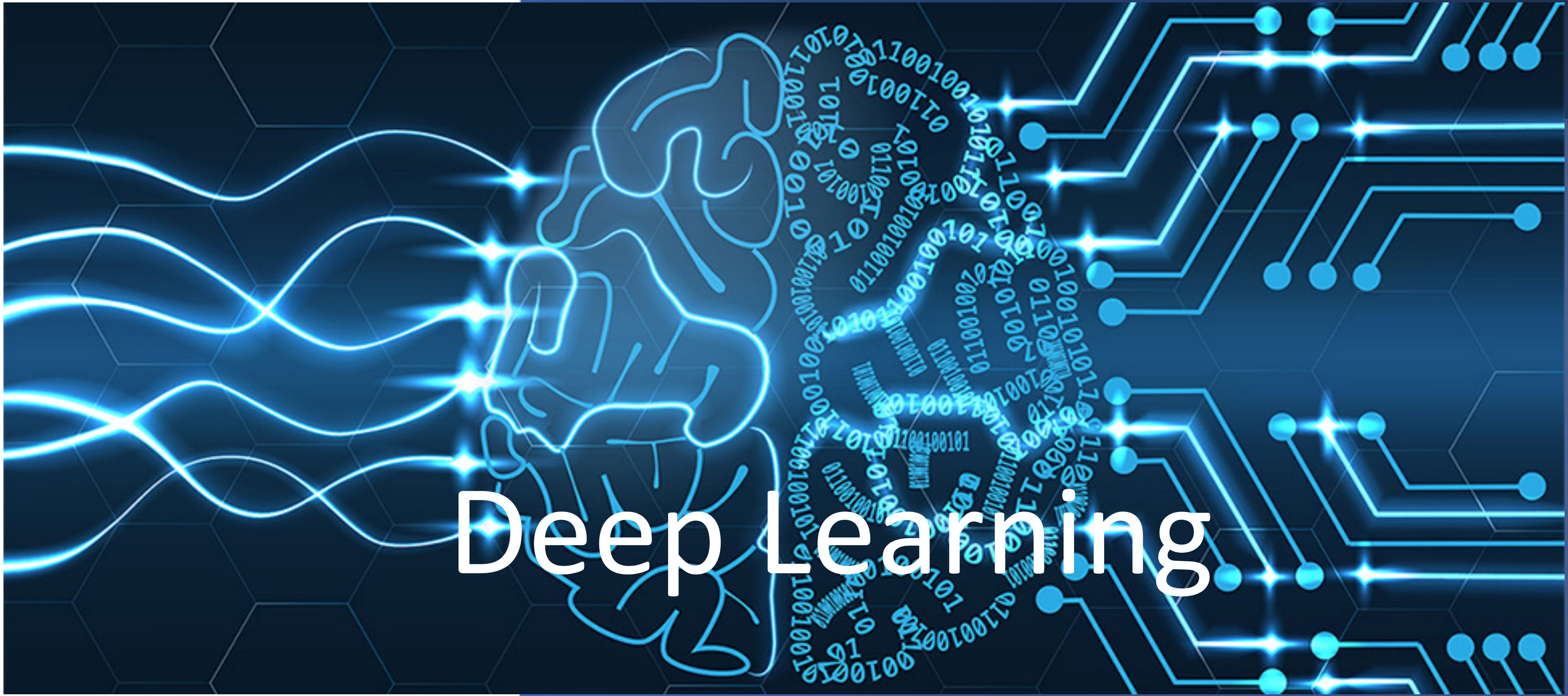
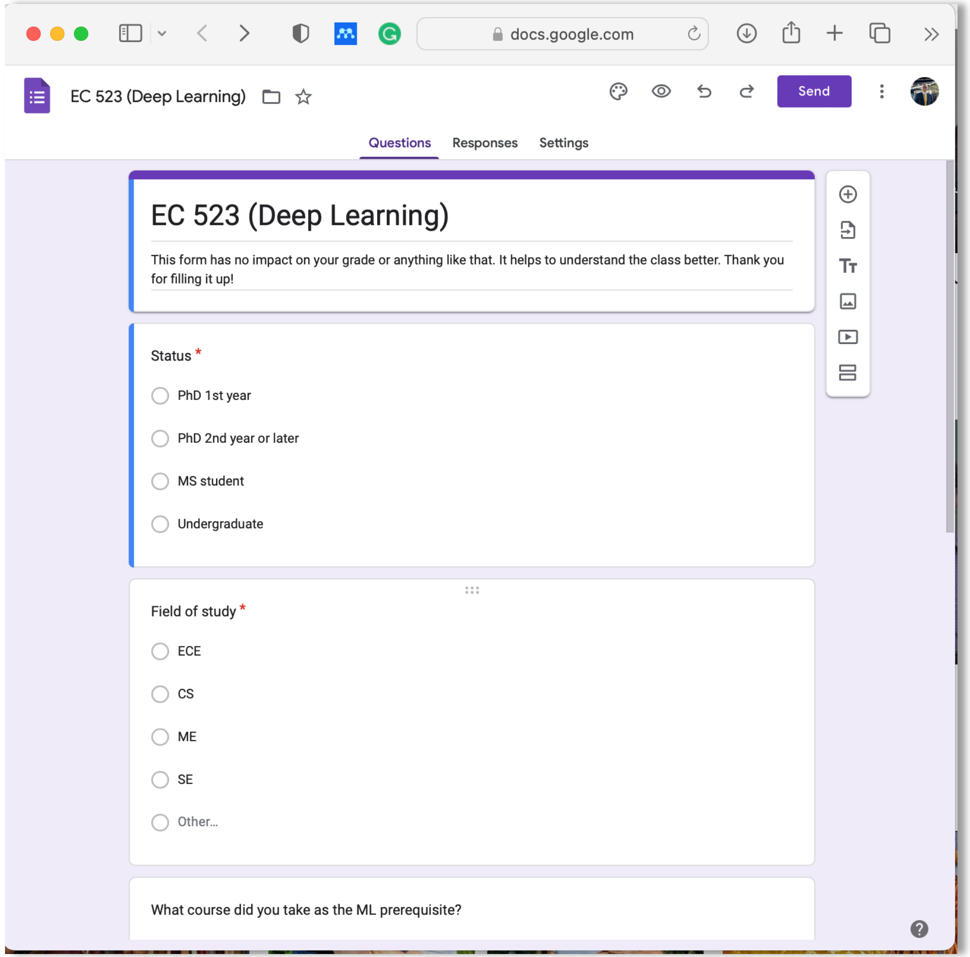


Deep Learning

with Prof. Kayhan Batmanghelich

Now, what about you?


docs.google.com

EC 523 (Deep Learning)

Send

Questions Responses Settings

EC 523 (Deep Learning)

This form has no impact on your grade or anything like that. It helps to understand the class better. Thank you for filling it up!

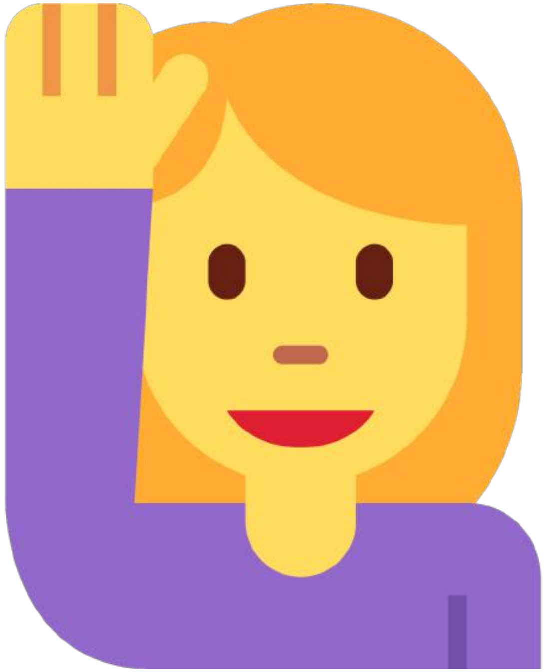
Status *

- PhD 1st year
- PhD 2nd year or later
- MS student
- Undergraduate

Field of study *

- ECE
- CS
- ME
- SE
- Other...

What course did you take as the ML prerequisite?



https://forms.gle/hRceDWfK52qSKxm16

Outline

- Course website & logistics
- Syllabus
  - Problem sets
  - Projects
  - Grading
- What is deep learning?

Deep Learning

Course Information

Course Logistics

Sections:
ENG EC523 A1
M/W 2:30-4:15pm, PHO 203

Instructor:
Kayhan Batmanghelich,
office hours M 10-11am outside PHO 421

Teaching Assistant:
Li Sun (lisun@bu.edu)
Thursday 2pm-3pm, PHO 4th floor

Graders/Additional Staff:
Jordan Koseski
Priyank Negi

Li Sun “The TA”

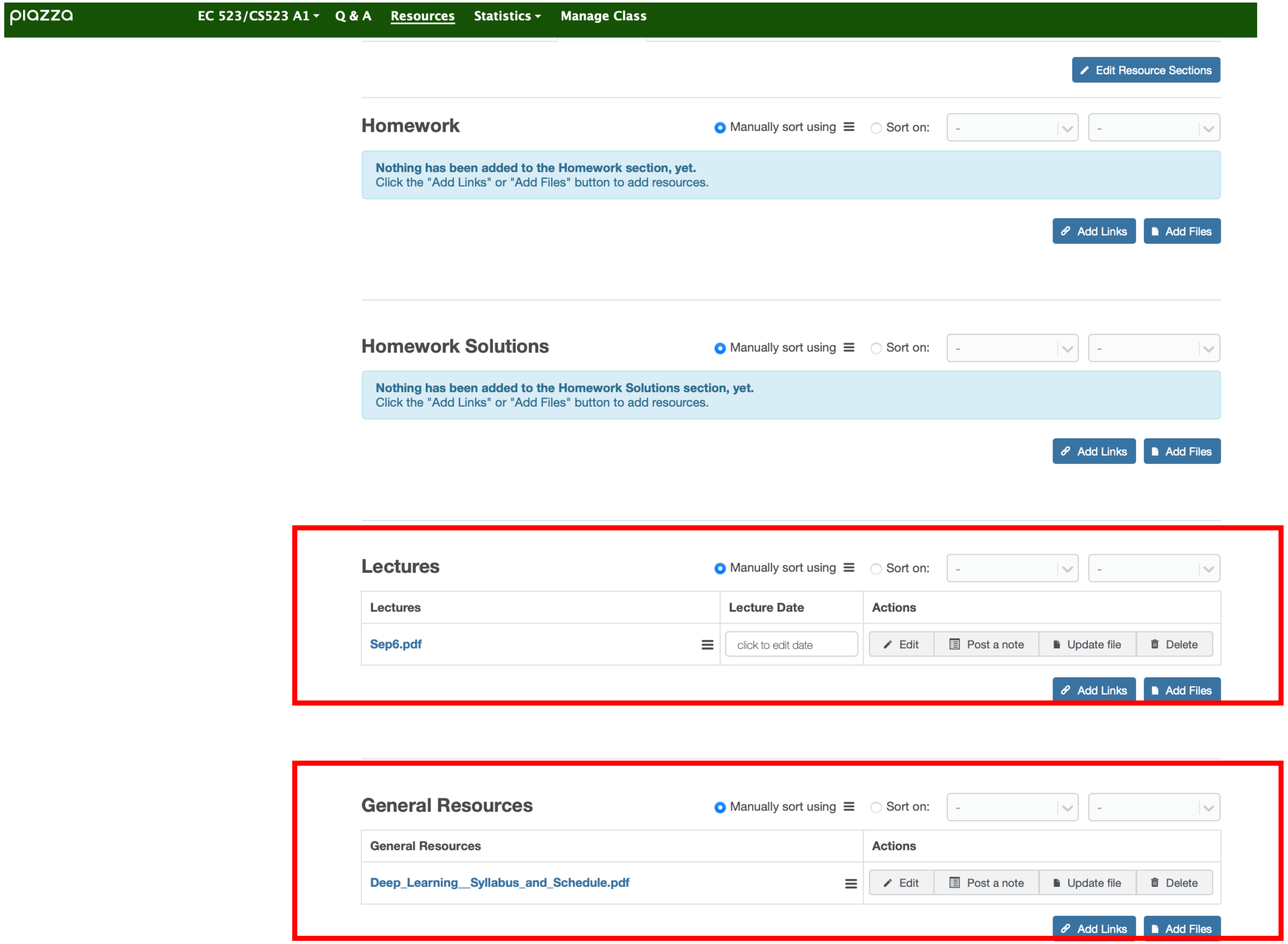
Piazza

How to Contact us: Please use Piazza for all communication; if your question is only directed to the instructors, please make a post to “Individual Student(s) / Instructor(s)” and select “Instructors".

Piazza: https://piazza.com/bu/fall2023/ec523cs523a1
We will be using piazza for online discussions, questions, and to post assignments.

Gradescope: we will be using Gradescope for submitting and grading assignments.

On Piazza


piazza
EC 523/CS523 A1
Q & A
Resources
Statistics
Manage Class
Edit Resource Sections
Homework
Manually sort using
Sort on:
Nothing has been added to the Homework section, yet.
Click the "Add Links" or "Add Files" button to add resources.
Add Links
Add Files
Homework Solutions
Manually sort using
Sort on:
Nothing has been added to the Homework Solutions section, yet.
Click the "Add Links" or "Add Files" button to add resources.
Add Links
Add Files
Lectures
Manually sort using
Sort on:
Lectures
Lecture Date
Actions
Sep6.pdf
click to edit date
Edit
Post a note
Update file
Delete
Add Links
Add Files
General Resources
Manually sort using
Sort on:
General Resources
Actions
Deep_Learning__Syllabus_and_Schedule.pdf
Edit
Post a note
Update file
Delete
Add Links
Add Files

Pre-Requisites

Course Pre-requisites

This is an upper-level undergraduate/graduate course.
All students should have the following skills:

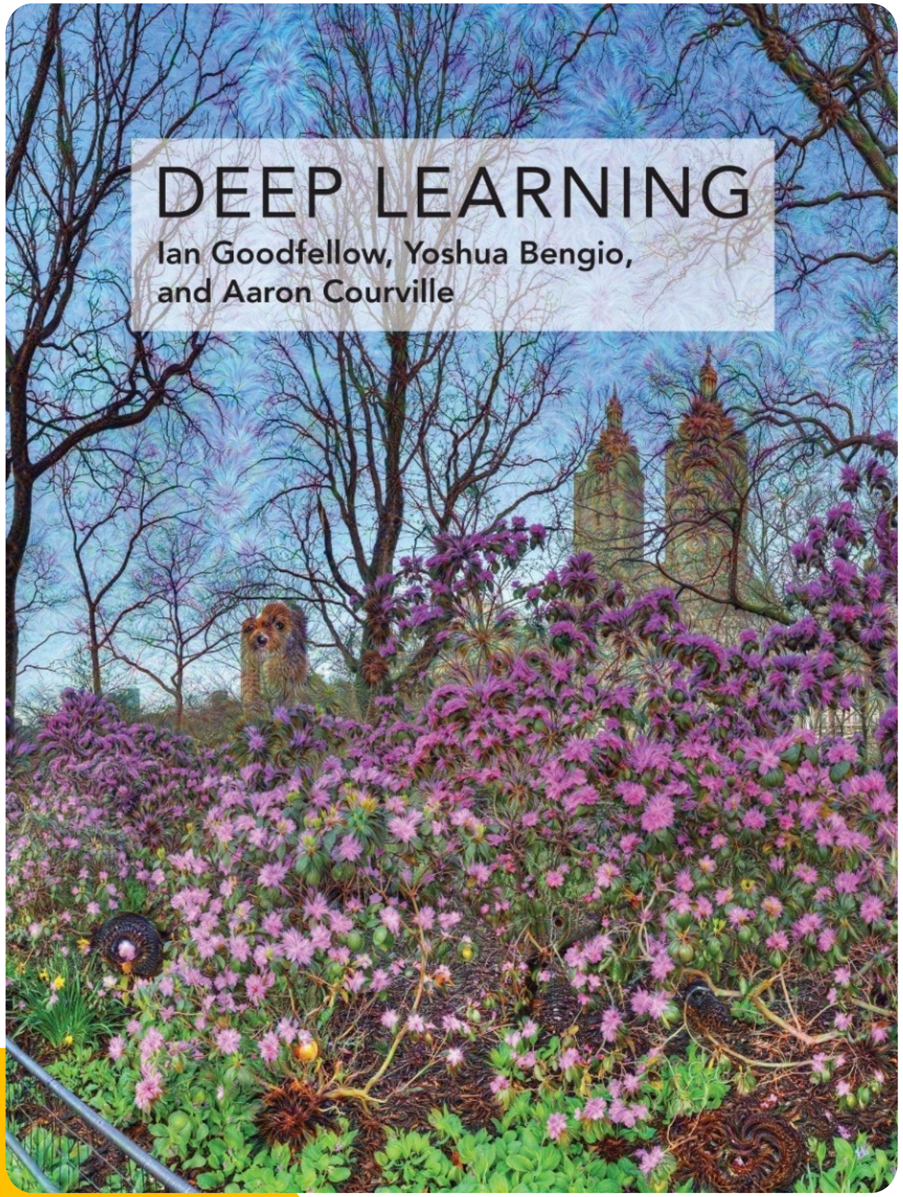
- Calculus, Linear Algebra
- Probability & Statistics
- Ability to code in Python
- Background in machine learning

Shared Computing Center

- Projects (and possibly some homework) will be done using GPUs from the SCC
  - Can use Google Colab for most/all homework assignments
  - If you have your own GPU resources, feel free to use
- We have access to 28 NVIDIA P100s
- Will have an introduction to using SCC later this month

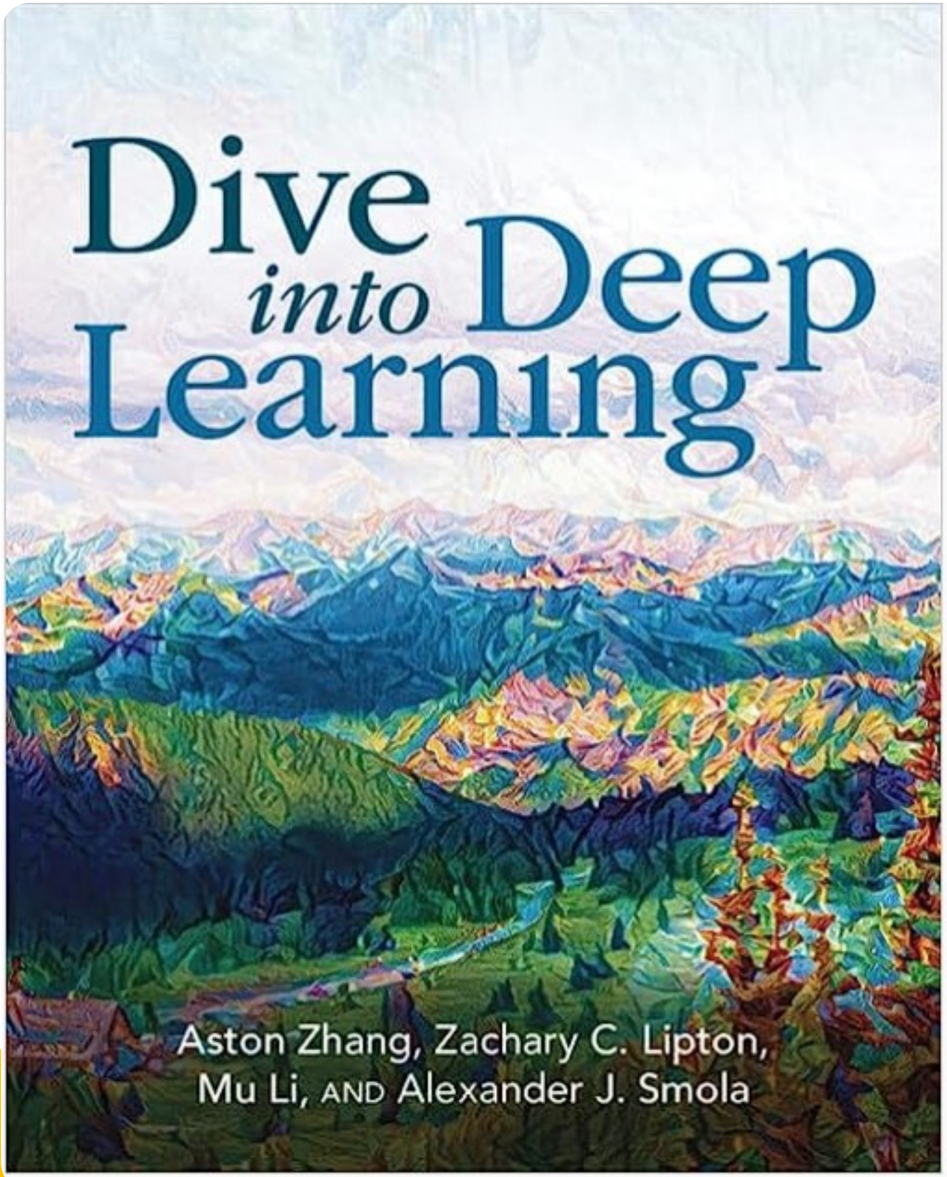
Topics and tentative schedule

1. Course overview
2. Math review I
3. Math review II
4. Neural Network Basics I
5 Neural Network Basics II
6. Neural Network Basics III
7. Training Neural Networks I
8. Training Neural Networks II
9. Training Neural Networks III
10. CNNs I
11. CNNS II
12. Advanced Architecture Design
13. Unsupervised Deep Learning I
14. Unsupervised Deep Learning II
15. Unsupervised Deep Learning III
16. Unsupervised Deep Learning IV
17. RNNs I
19. Transformers I
20. Transformers II
21. Deep reinforcement learning I
22. Deep RL II
23. Explainability I
24. Explainability II
24. Self-supervised Learning



Textbook

Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning.



Textbook

Authors

Aston Zhang

Amazon

Zack C. Lipton

CMU and Amazon

Mu Li

Amazon

Alex J. Smola

Amazon

Deliverables/Graded Work

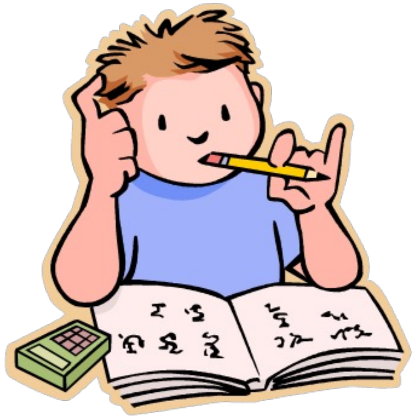
- There will be five homework assignments, each consisting of written and/or coding problems, and a final project.
- The course grade consists of the following:
  - Homeworks (hw1 and best 3 of 2-5) 50%
  - Project (including all components) 45%
  - Class/Piazza participation 5%

Late policy & academic honesty

- Please see syllabus
- Your work should be your own
- If you consult an online source please cite it in your work

Homeworks

- Five homeworks
- Roughly every two weeks
- A mix of written questions and coding
- Coding for HW3-5 will be in PyTorch
- Will drop lowest score in HW 2-5
- HW1 out **next week**
  - ML/math review
  - You cannot drop this



Project

- The project will be done in teams of 3-4 students
- Projects will have several deliverables including
  - a proposal
  - progress update(s)
  - final report
  - Github repository
  - in-class presentation
- Project grade is based on all of the above components

[BU Shared Computing Cluster/]

Sample Projects

Deep Reinforcement Learning for Playing Atari Games

Giuliano Conte gaconte@bu.edu
Ravi Teja Darbha rdarbha@bu.edu
Sheng Xiao shengx@bu.edu
G Siva Perumal siva@bu.edu
Shreeya Khadka skhadka@bu.edu

1. Task

Reinforcement learning (RL) is a method in which an agent learns by interacting with its environment. It involves making a series of decisions to maximize the rewards accumulated by the agent for each legal step it takes. This is achieved by encouraging the agent to choose a winning strategy from a set of available strategies by positively rewarding the winning strategy and discouraging it from choosing a bad strategy by penalizing the losing strategy. By this process of reinforcing our agent for every action it takes, RL eventually tunes its parameters to take an optimal action out of all the available actions based on the current state of the game [1].

Integration of RL with neural networks has brought about great results. For example, Google's Alpha Go which was based on deep Q-learning was able to beat a human Go expert [2]. With these recent advances in mind, we plan to implement and explore deep RL algorithms for online learning agents in a gaming environment.

the convolutional neural networks (CNNs) to integrate the performance of an agent over time [5]. This however didn't show much improvements over the DQN except for cases with partial observability where states aren't fully observable. In order to improve the speed of learning T.Schaul, et al. proposed 'prioritized experience replay', where important experiences are replayed more often [6]. Z. Wang, et al. proposed a new architecture called the 'Dueling network architecture' which improves the policy evaluation in case there are many actions which give similar rewards [7].

3. Approach

Reinforcement learning is a type of machine learning that focuses on how an agent should interact with environment [8]. Typically, a reinforcement learning agent will receive a series of state $(s_1, s_2, ..., s_n)$ over time n, and needs to make a series of actions $(a_1, a_2, ..., a_n)$, which leads to a series of reward $(r_1, r_2, ..., r_n)$ at every time step. The goal is to maximize cumulative reward $R = f(r_1, r_2, ..., r_n)$ according to certain metrics. Because the environment is usu-

Sample Projects

Image Compression Using Deep Learning

H. Kubra Cilingir, M. Ozan Tezcan, Sivaramakrishnan Sankarapandian
kubra@bu.edu, mtezcan@bu.edu , sivark@bu.edu

Figure 1. (left) ort best performing algorithm (CNN-AE-FT), (right) original image

1. Task

Image compression has an important role in data transfer and storage, especially due to the data explosion that is increasing significantly faster than Moore's Law.[1] It is a challenging task since there are highly complex unknown correlations between the pixels, as a result, it is hard to find and recover them. We want to find a well-compressed representation for images and, design and test networks that are able to

addressed the problem of JPEG compression for small images where the amount of redundant information is small. This network follows the classical three stages process of compression: encoding, quantization and decoding. Encoding and decoding are done iteratively using two different architecture of RNNs, Long Short Term Memory (LSTM) and convolutional/ deconvolutional LSTMs. The advantage of this method is that the compression ratio can be increased

Sample Projects

Gender Prediction Using Character-Level Language Models

Andrew Cutler, Mehrnoosh Sarmashghi, Ali Siahkamari
{acut,msmshgi,siaa}@bu.edu

1. Task

Given a collection of someone's facebook statuses, can you predict their gender or personality? With a good language model (such as in the brain of the reader) natural language is rich with information about the author. However, given the number of colloquialisms and misspellings on Facebook, typical dictionary-based language models fail. We will implement a character-based CNN as well as an LSTM language model as feature extractors on the statuses. These will be followed by a fully connected classification network. We have the a five-dimensional personality vector (openness, conscientiousness, extraversion, agreeableness, neuroticism) for 2.5 million facebook users, as well as their gender. These labels will allow us to test performance of language model as well as explore benefits from domain adaptation.

2. Related Work

Author gender is closely related to sentimentality analysis, which has been richly explored. N-gram bag-of-words techniques, perform quite well and have long been the standard language model for these prediction tasks. However, these models discard syntactic information (eg. "threatening a child" vs "a threatening child") that is important to understand

matrix are large, it can lead to a situation called exploding gradients.

These problems are addressed by the Long Short-Term Memory neural network architecture which is capable of learning long-term dependencies, see[1]

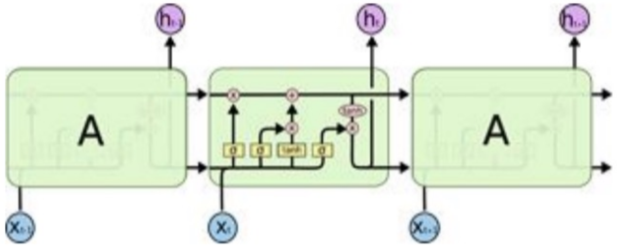


Figure 1. The repeating module in an LSTM

In this work, we exploit an LSTM language model on the Facebook statuses to predict gender and personality of people.

Yann Le Cun et al. argue that LSTM is a general model and require searching a larger parameter space[2]. Though there is little theoretical work to support this claim, empirical results are promising. Even with modest amounts of training data, they are able to come near N-gram performance.

Domain Adaptation and Multi-task learning are related topics that are still getting much attention for research.

Sample Projects

Video Generation with Generative Adversarial Networks

Michael Clifford, Casey Fitzpatrick
{jcliff, cfitz}@bu.edu

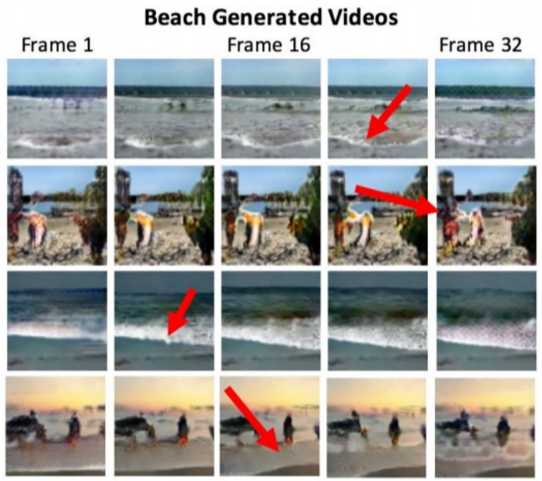


Figure 1. Some generations from the two-stream model. The red arrows highlight motions. Please see http://mit.edu/vondrick/tinyvideo for animated movies. Source [1].

1. Task

Our ultimate task was to design a generative adversarial network (GAN) capable of conditionally generating realistic, short video clips given a static image input, reproducing the results of [1] in a Keras environment with a TensorFlow backend (as opposed

The second reason this problem is hard is more fundamental with respect to generative video modeling. Namely, the approach used for generation consists of a forked parallel architecture that attempts to distinguish the foreground from background in order generate motion in the foreground while keeping the background fixed. The two streams are them joined together at the end of the generator using a branched mask layer designed to emphasize a fixed background while steadily tracking the motion of the foreground in time. While very interesting, according to Vondrick this approach has not been used in purely generative contexts before. So even the "state-of-the-art" (Vondrick's results) aren't particularly impressive in most cases. This is bleeding edge stuff!

The final reason this problem is difficult is related to the known (and perhaps fundamental) issues with training GAN models. Specifically, since GAN models train based on minimax games, the gradient descent process is seeking a saddle point as opposed to a local minimum. It is inherently a harder to find such a region in a manifold, so convergence guarantees are out the window. Additionally, the models are hard to

Questions?

What is Deep Learning?

Course Information

I am a professor and it is the first day of my graduate-level deep learning class. Write an inspiring song in the style Kanye West to the students during the class!

(Beat drops)

(Intro)
Yo, yo, yo, listen up, class
We 'bout to dive deep, no looking back
This ain't just numbers, this is art
Deep learning, let's tear it apart

(Verse 1)
Uh, welcome to the grad life, the hard strife
TensorFlow, PyTorch, cutting like a knife
But don't get it twisted, it ain't just code
It's the future, baby, it's the open road

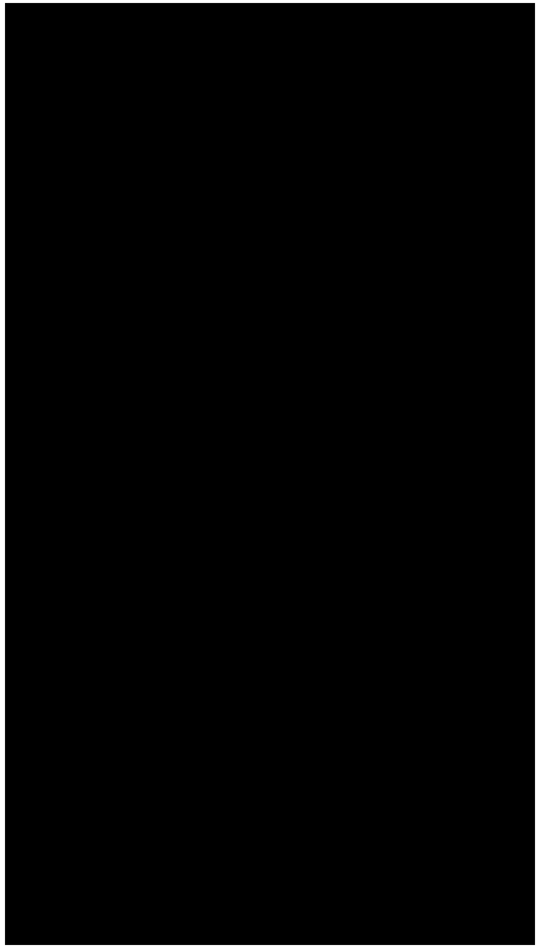
(Pre-Chorus)
We got layers, yeah, neural and deep
From perceptrons to LSTMs, no cheap seats
ConvNets, GANs, making history
This ain't just math, it's a victory

(Chorus)
So let's go, let's go, let's make some noise
Deep learning, it's more than just a toy
We're the future, yeah, we're the chosen few
Let's make some magic, that's what we do

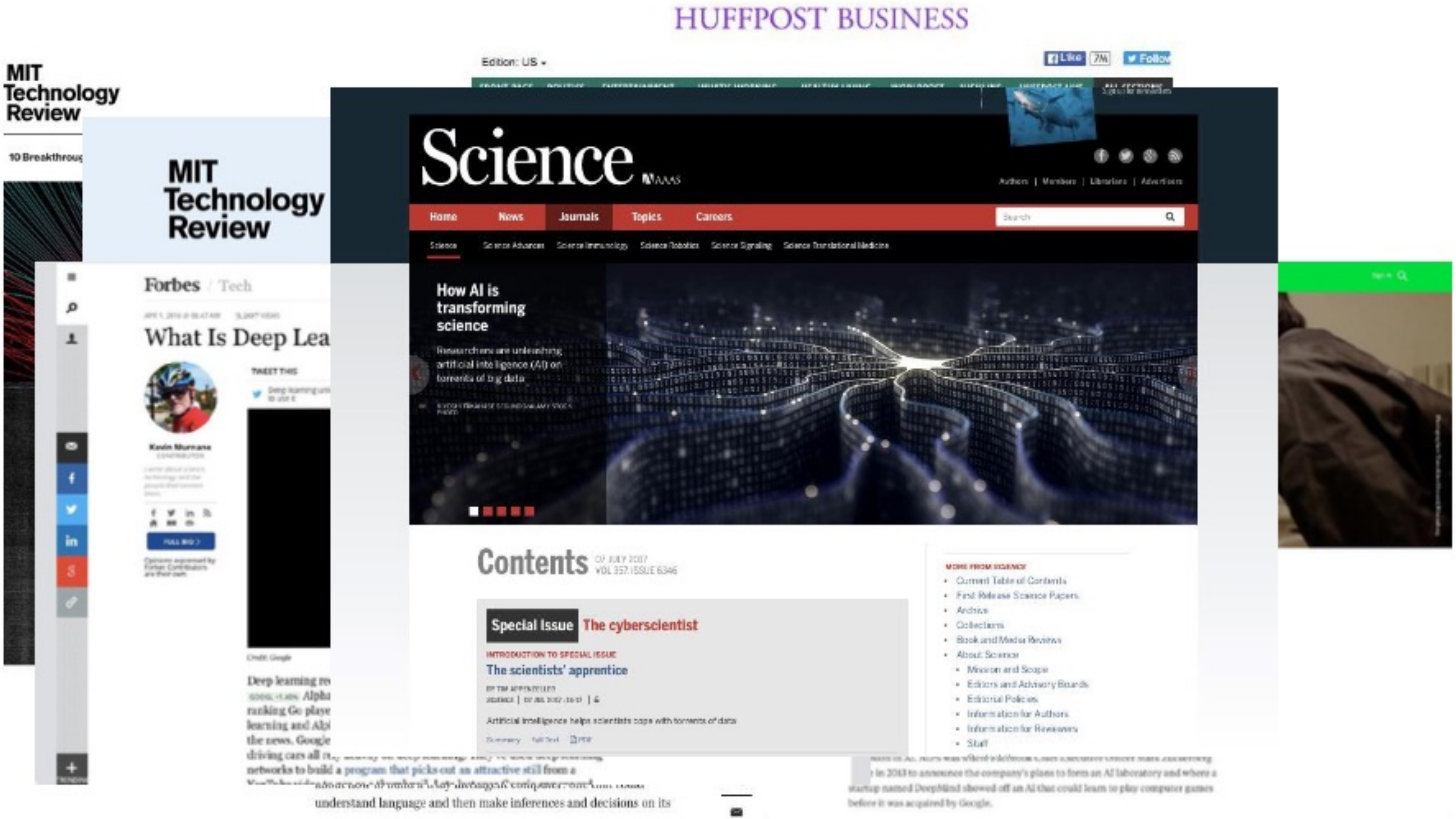
(Verse 2)
You might struggle, yeah, you might fall
Backpropagation got you hitting a wall
But remember, even Kanye had his lows
808s & Heartbreak, but look how he rose

(Bridge)
So when you're stuck on that gradient descent
Remember, every genius had their discontent
You're the next pioneers, the next wave
Deep learning, y'all, it's how we'll save

(Chorus)
So let's go, let's go, let's make some noise
Deep learning, it's more than just a toy



UBERDUCK



Slide credit: Aykut Erdem // Koç University // Fall 2022

What is deep learning?


REVIEW

Deep learning

"Deep learning allows computational models that are composed of **multiple processing layers to learn representations of data with multiple levels of abstraction.**"
— Yann LeCun, Yoshua Bengio and Geoff Hinton

Y. LeCun, Y. Bengio, G. Hinton, **"Deep Learning"**, Nature, Vol. 521, 28 May 2015

What is deep learning?

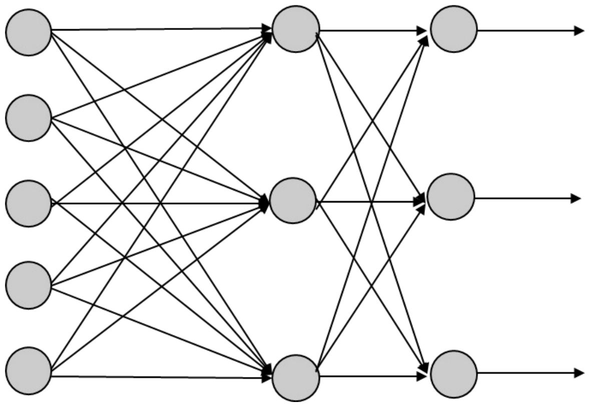
Neural network architectures that contain many layers.

Learn strong *representations* of highly complex and unstructured data.

Especially: images, text, or sound.

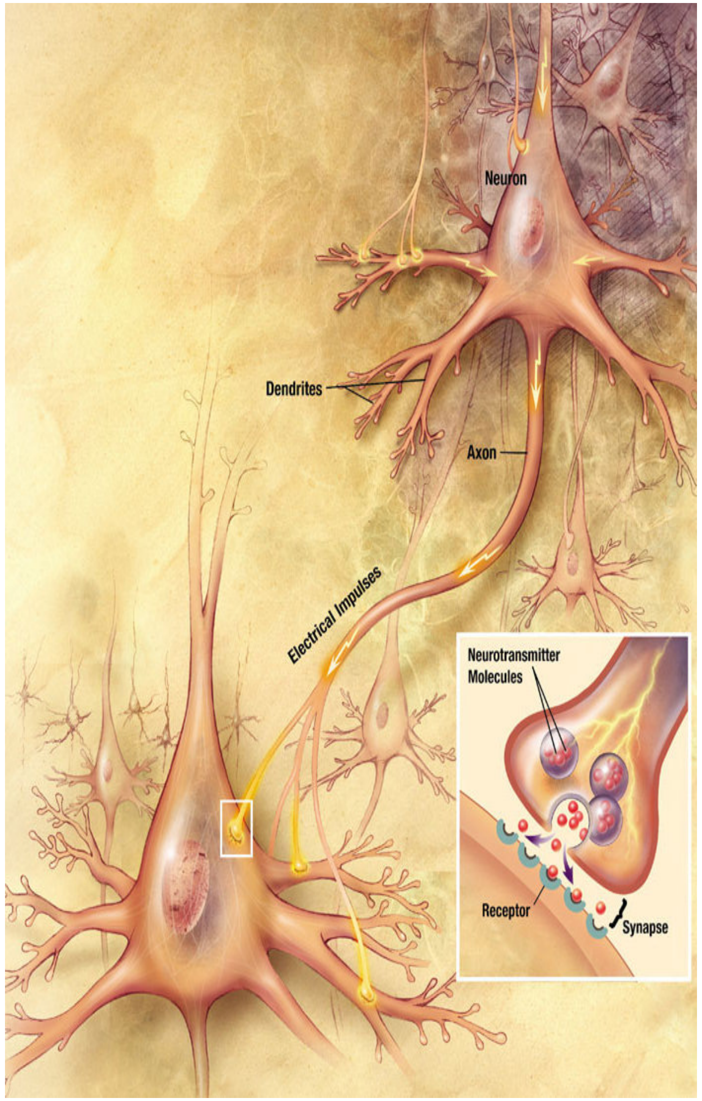
Neurons in the Brain

Inspired "Artificial Neural Networks"



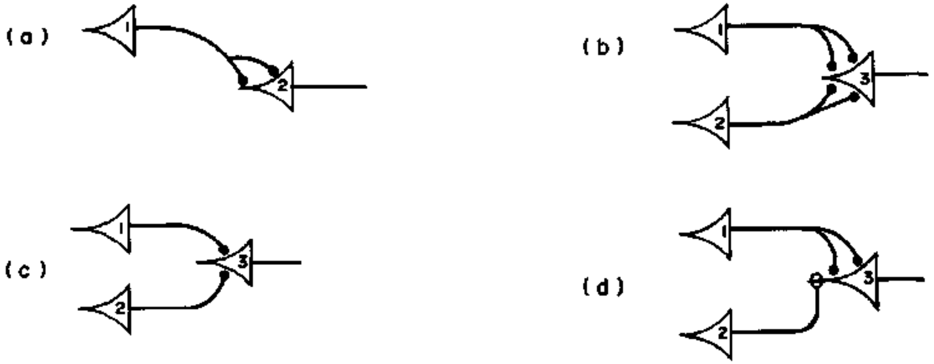
Neurons are cells that process chemical and electrical signals and transmit these signals to neurons and other types of cells

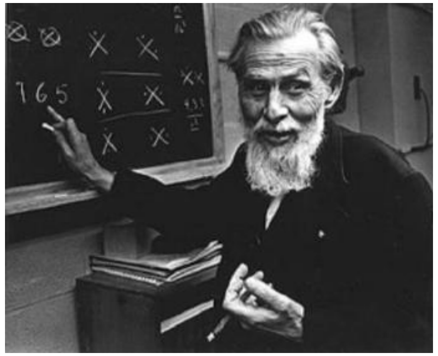
A brief timeline


Neuron
Dendrites
Axon
Electrical Impulses
Neurotransmitter Molecules
Receptor
Synapse

1943: Warren McCulloch and Walter Pitts

- First computational model
- Neurons as logic gates (AND, OR, NOT)
- A neuron model that sums binary inputs and outputs a 1 if the sum exceeds a certain threshold value, and otherwise outputs a 0





Bulletin of Mathematical Biology Vol. 52, No. 1/2, pp. 99–115, 1990.
Printed in Great Britain.

0092-8240/90$3.00+0.00
Pergamon Press plc
Society for Mathematical Biology

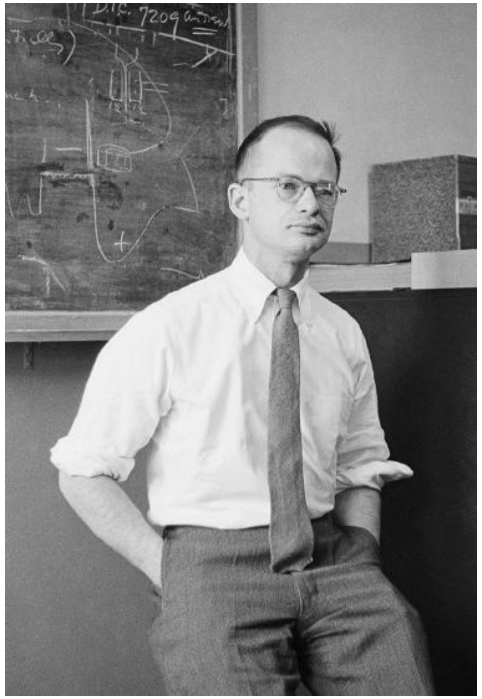
A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

1. Introduction. Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from $<1\ ms^{-1}$ in thin axons, which are usually short, to $>150\ ms^{-1}$ in thick axons, which are usually long. The time for axonal conduction is consequently of little importance in determining the time of arrival of impulses at points unequally remote from the same source. Excitation across synapses occurs predominantly from axonal terminations to somata. It is still a moot point whether this depends upon irreciprocity of individual synapses or merely upon prevalent anatomical configurations. To suppose the latter requires no hypothesis *ad hoc* and explains known exceptions, but any assumption as to cause is compatible with the calculus to come. No case is known in which excitation through a single synapse has elicited a nervous impulse in any neuron, whereas any neuron may be excited by impulses arriving at a sufficient number of neighboring synapses within the period of latent addition, which lasts <0.25 ms. Observed temporal summation of impulses at greater intervals

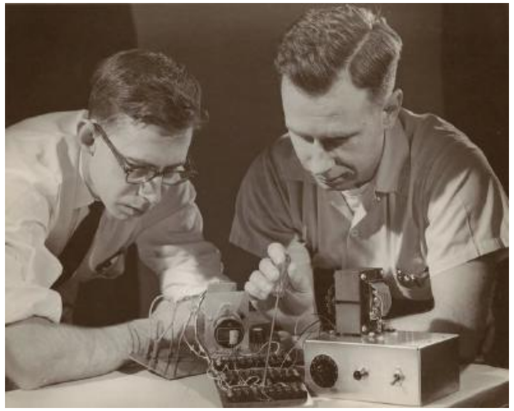
* Reprinted from the *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133 (1943).

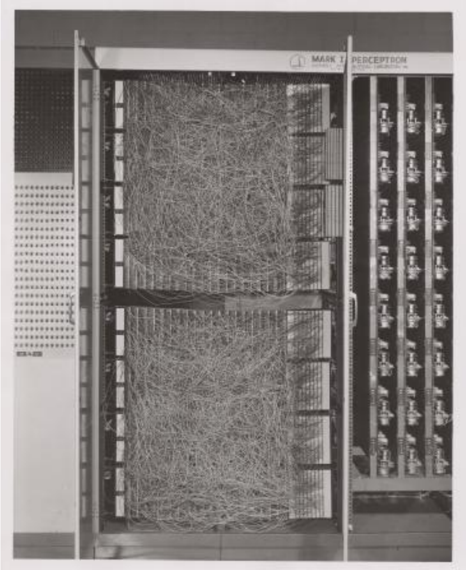


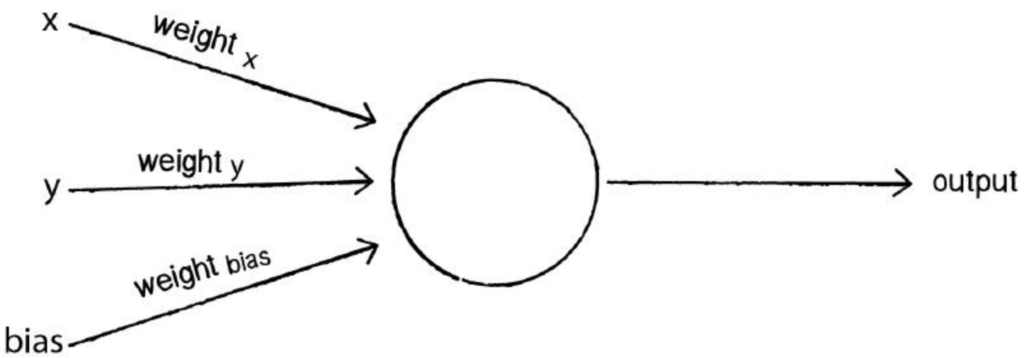
Slide credit: Aykut Erdem // Koç University // Fall 2022

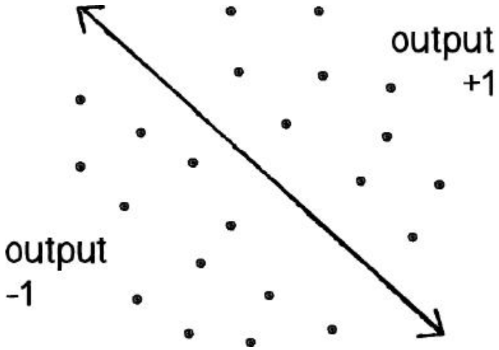
1958: Frank Rosenblatt's Perceptron

- A computational model of a **single neuron**
- Solves a **binary classification problem**
- Simple training algorithm
- Built using specialized hardware



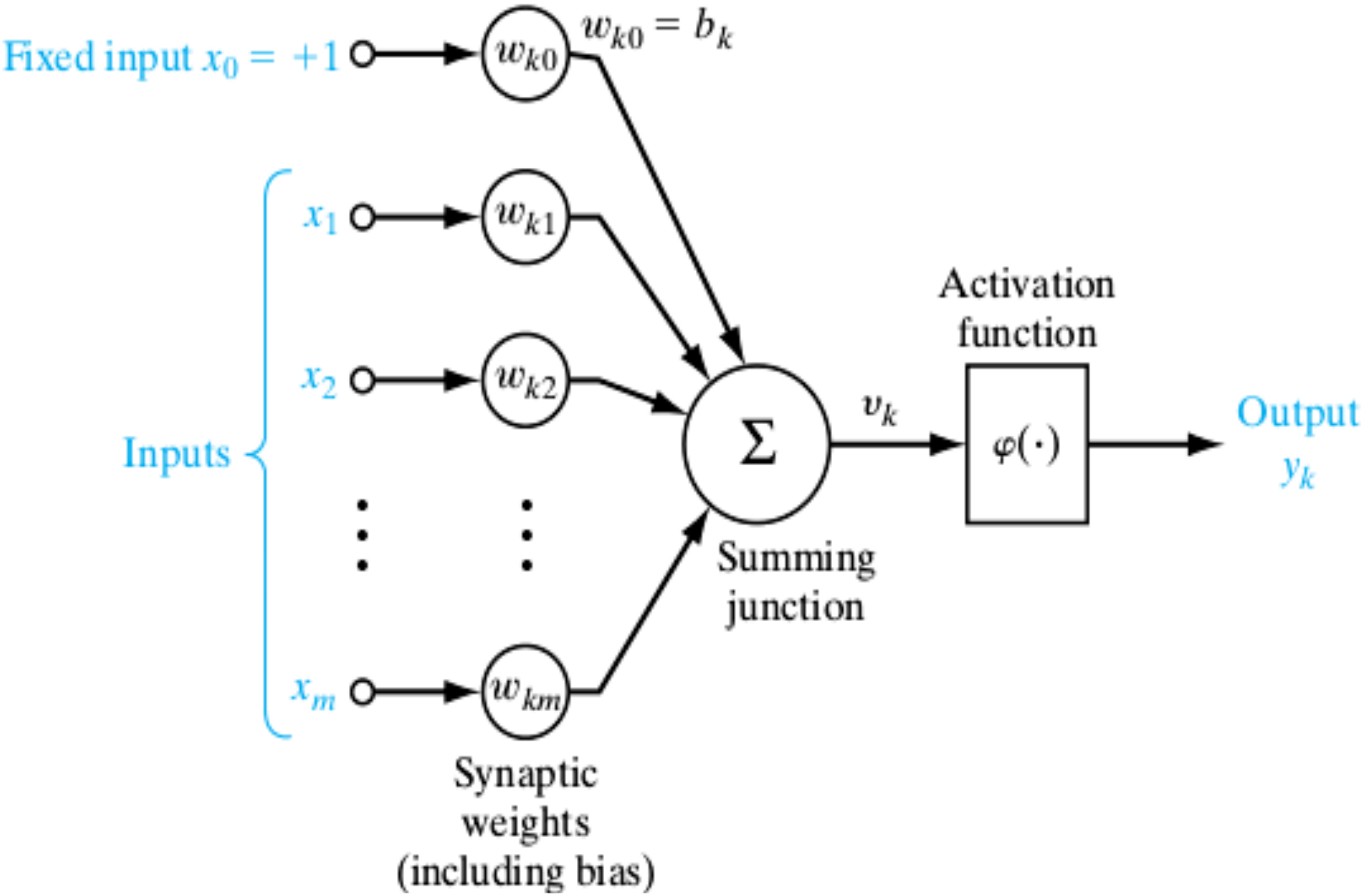






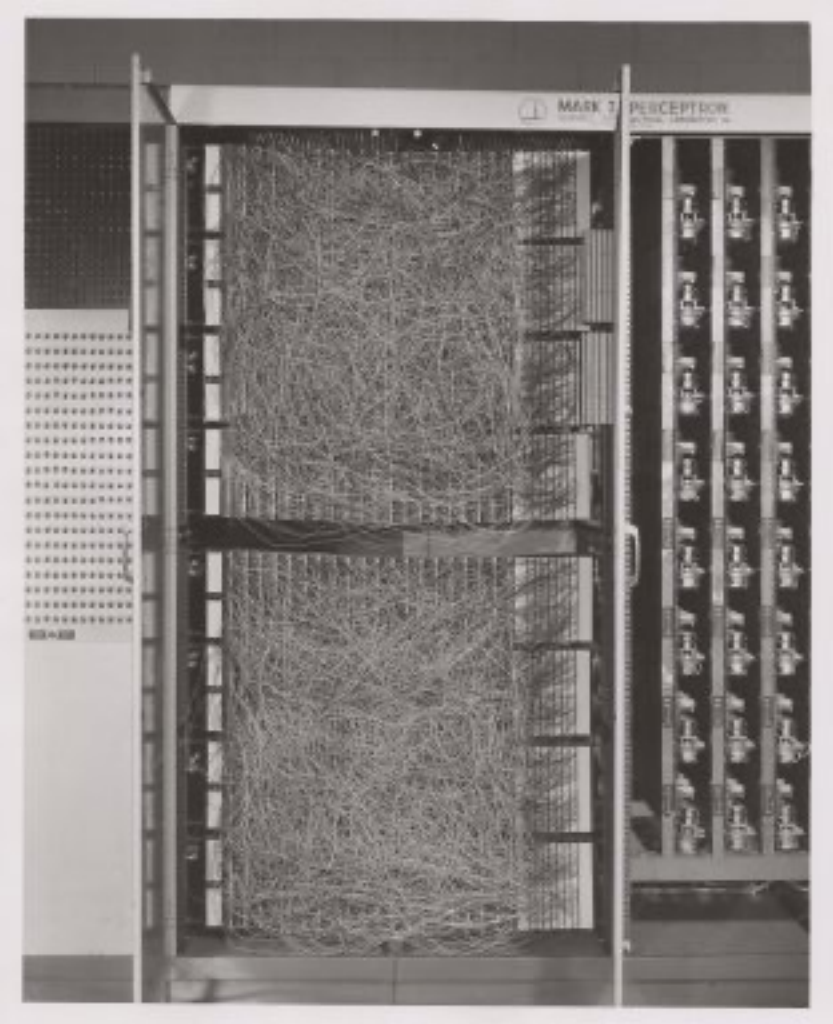
Slide credit: Aykut Erdem // Koç University // Fall 2022

Perceptron



Fixed input $x_0 = +1$
w_{k0}
$w_{k0} = b_k$
x_1
w_{k1}
x_2
w_{k2}
Inputs
x_m
w_{km}
Σ
v_k
Activation function
$\varphi(\cdot)$
Output
y_k
Summing junction
Synaptic weights (including bias)

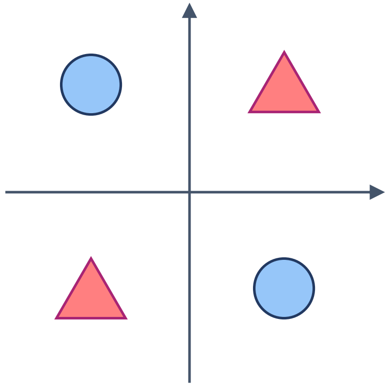
Perceptron



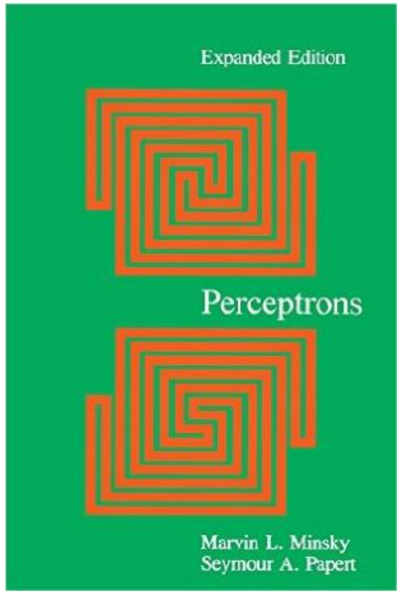
1969: Marvin Minsky and Seymour Papert

"No machine can learn to recognize X unless it possesses, at least potentially, some scheme for representing X." (p. xiii)

- Perceptrons can only represent linearly separable functions.
  - such as **XOR** Problem



- Wrongly attributed as the reason behind the **AI winter**, a period of reduced funding and interest in AI research

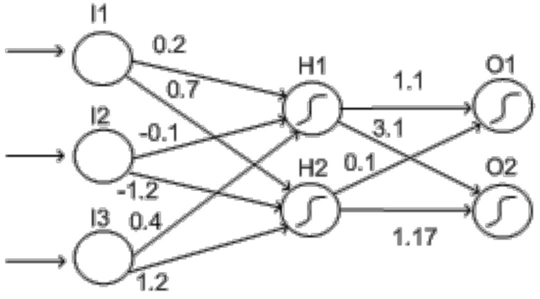


1990s

- **Multi-layer perceptrons** can theoretically learn any function (Cybenko, 1989; Hornik, 1991)

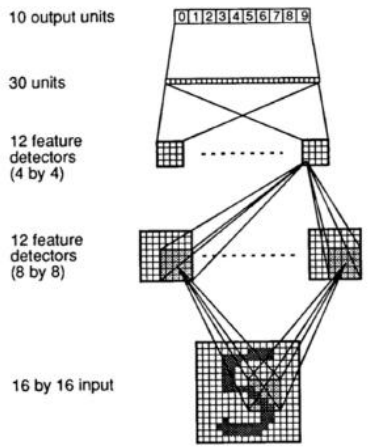
- Training multi-layer perceptrons
  - **Back propagation** (Rumelhart, Hinton, Williams, 1986)
  - **Backpropagation through time** (BPTT) (Werbos, 1988)

- New neural architectures
  - **Convolutional neural nets** (LeCun et al., 1989)
  - **Long-short term memory networks** (LSTM) (Schmidhuber, 1997)




Backpropagation Through Time: What It Does and How to Do It

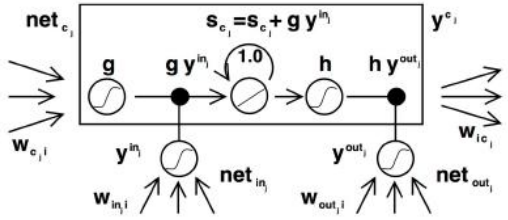
PAUL J. WERBOS




Handwritten Digit Recognition with a Back-Propagation Network

ABSTRACT

1 INTRODUCTION

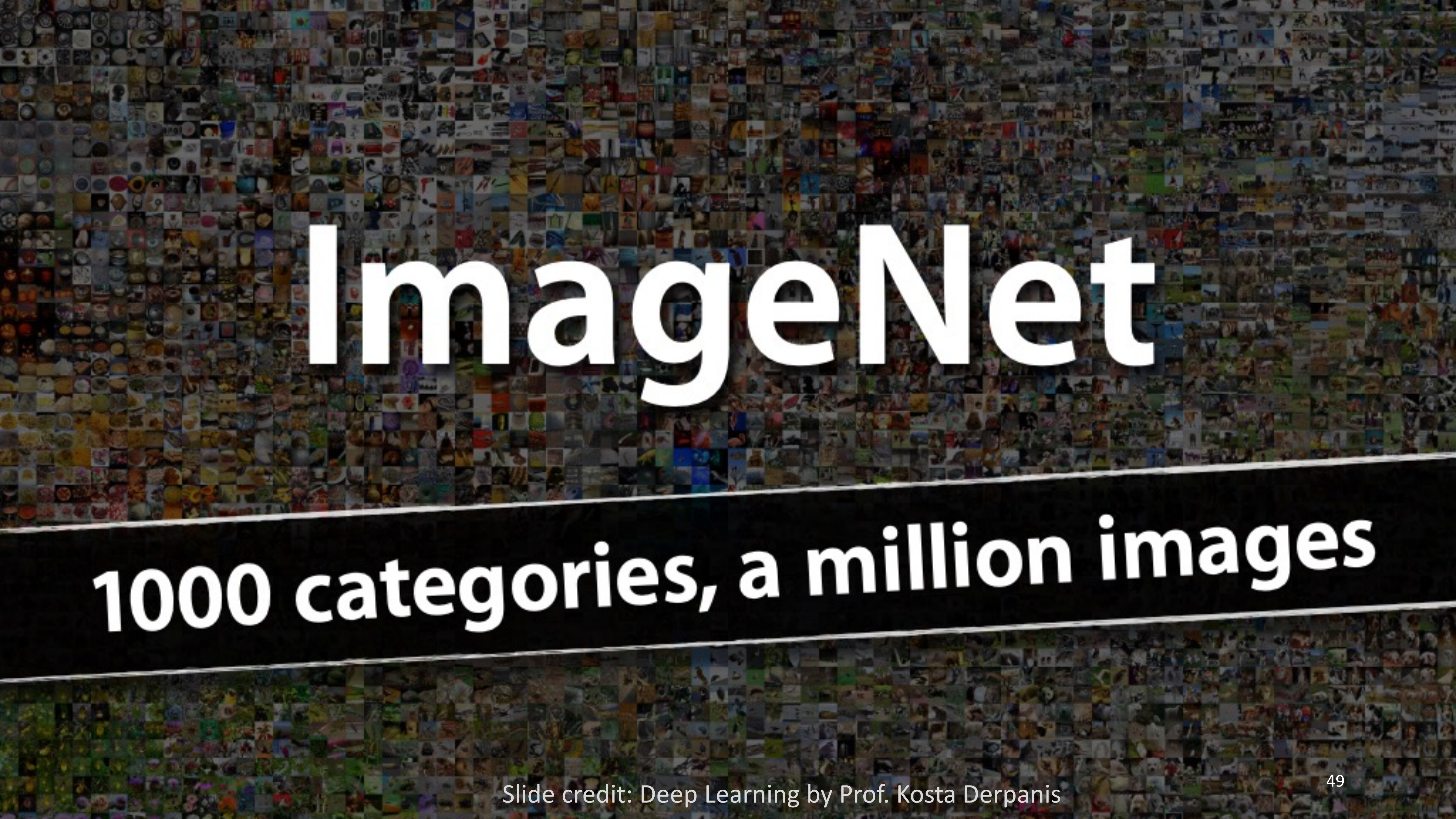


AI Winter

Brief Timeline

- 1943: McCulloch and Pitts
- 1957: Rosenblatt's Perceptron
- 1969: Minsky and Papert's *Perceptrons*
- '70s/'80s: Symbolic AI
- 1986: Backpropagation takes hold
- Late 80s –Mid 90s: Resurgence of NNs
- 1992: Kernel Methods (Support Vector Machines)
- Mid 90s – 2010: Statistical AI / Graphical Models
- 2006: Hinton's Deep Belief Nets
- 2010+: Deep Learning Gains Popularity
- 2029: …Skynet?


2012



Slide credit: Deep Learning by Prof. Kosta Derpanis

ImageNet Classification with Deep Convolutional Neural Networks

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

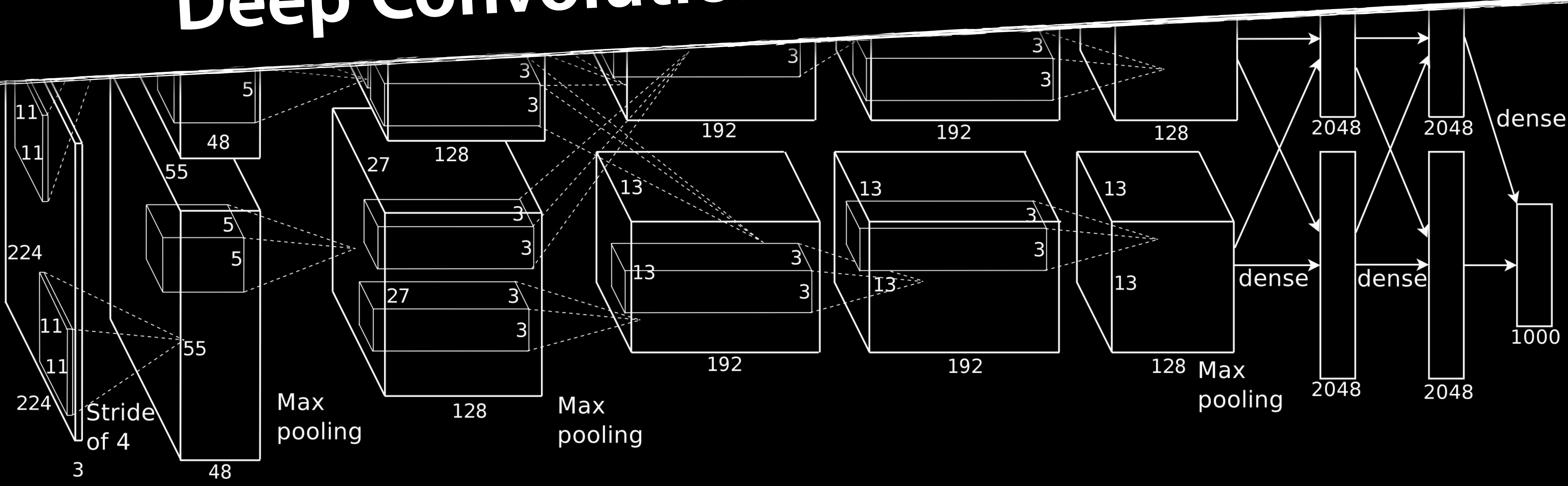
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

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

Abstract

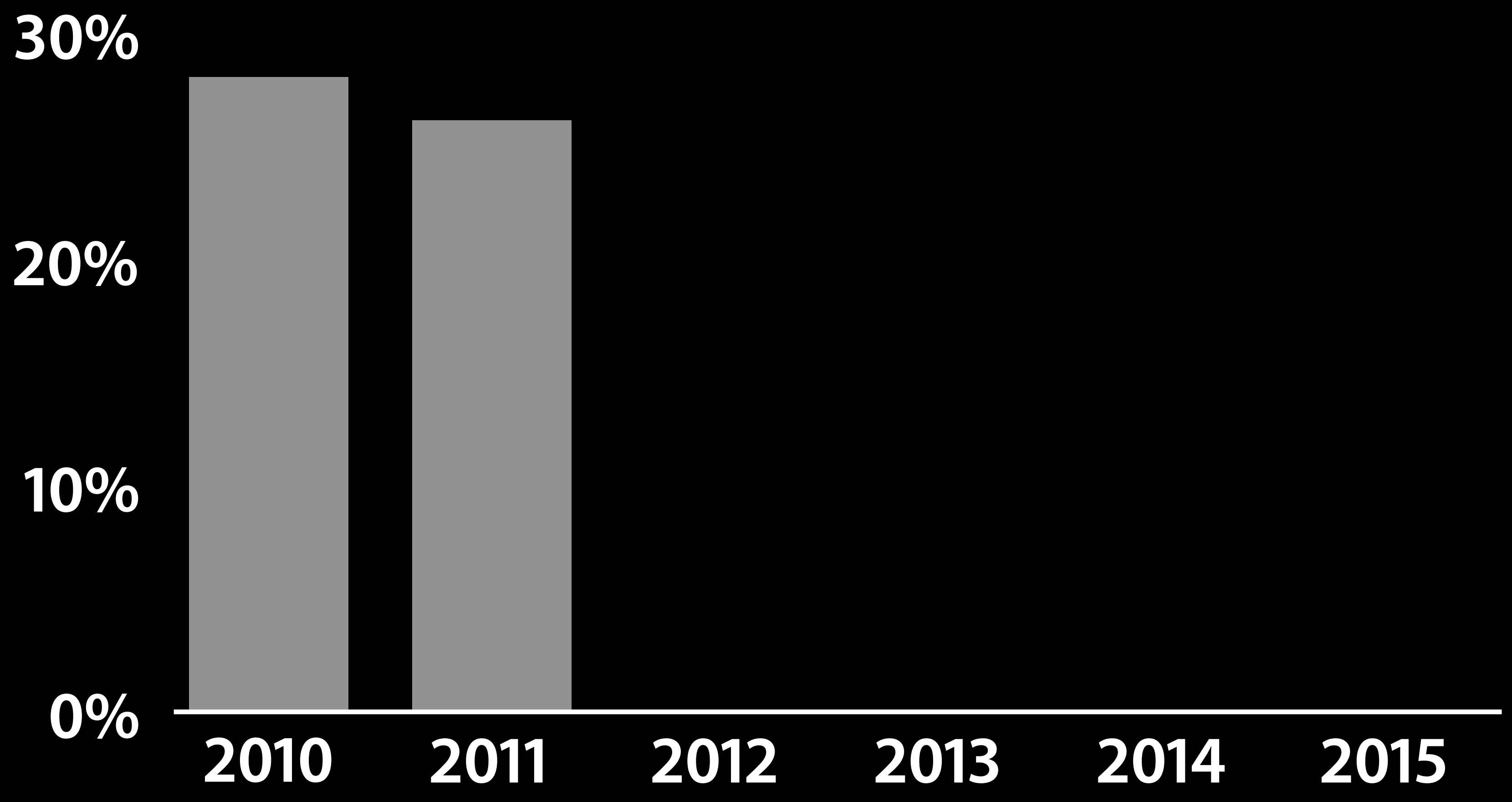
We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Deep Convolutional Neural Network



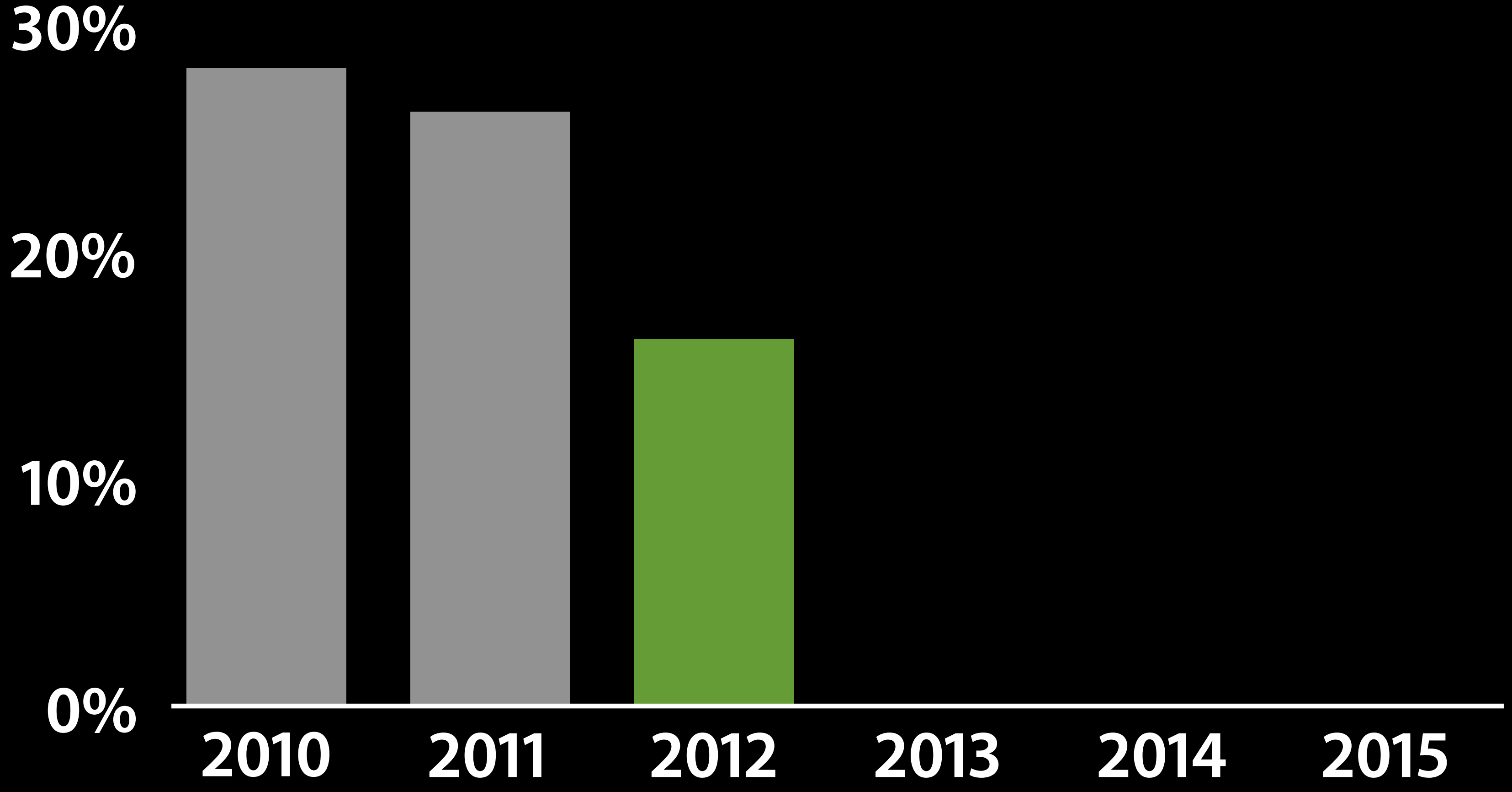
Slide credit: Deep Learning by Prof. Kosta Derpanis

ImageNet Error Rate (Best of 5 Guesses)



Slide credit: Deep Learning by Prof. Kosta Derpanis

ImageNet Error Rate (Best of 5 Guesses)

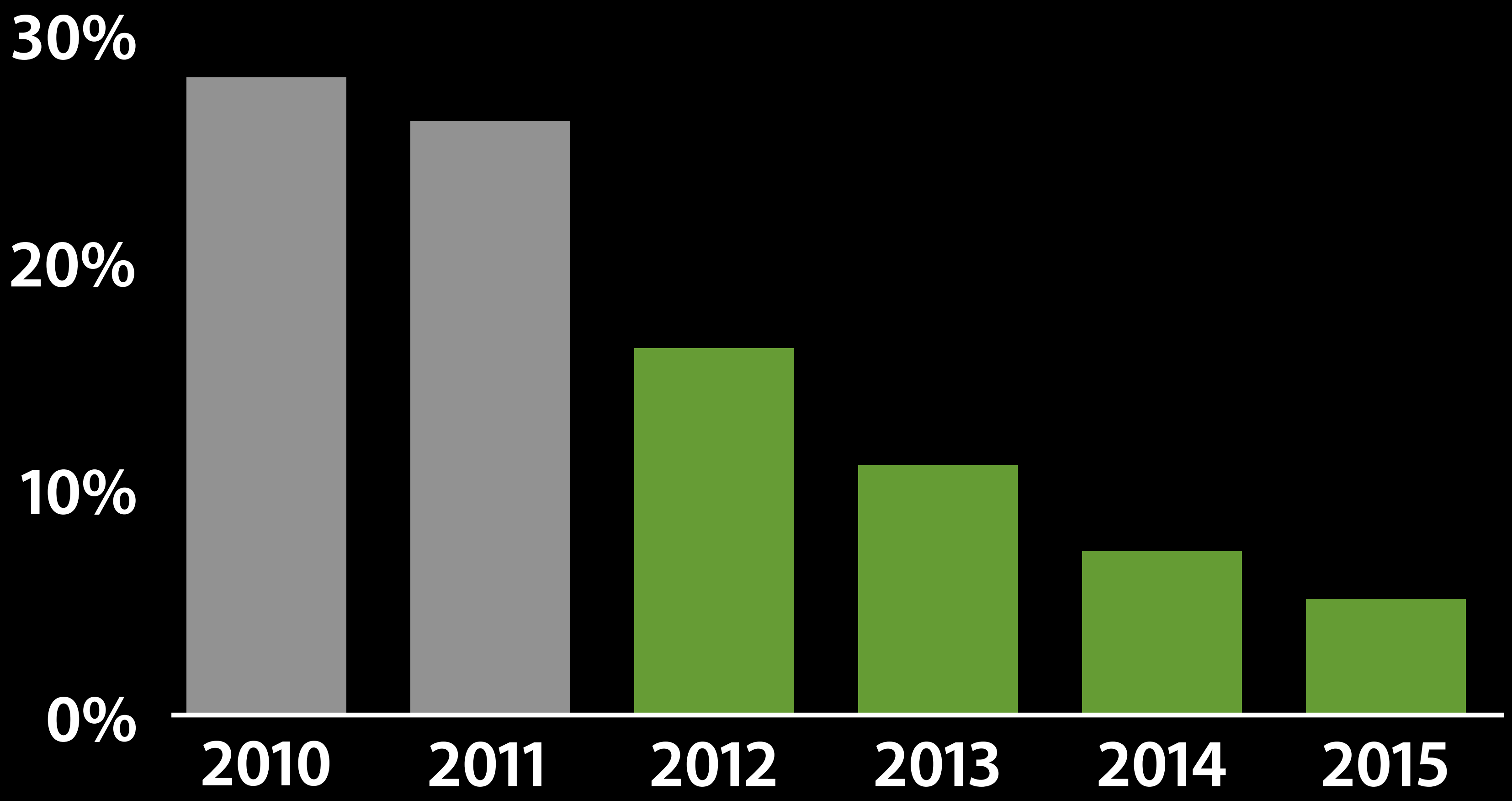


Slide credit: Deep Learning by Prof. Kosta Derpanis

Computer Vision Community

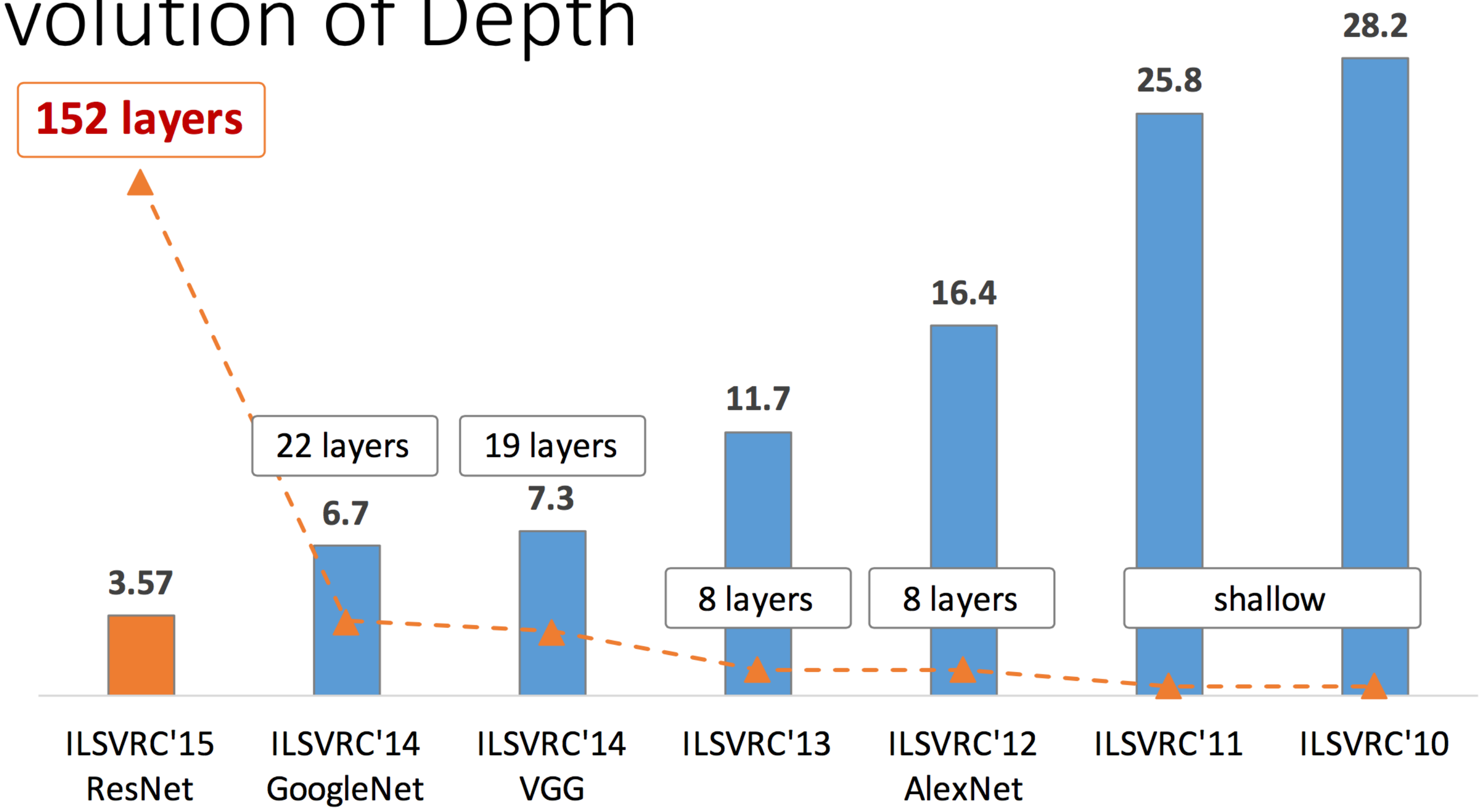
Slide credit: Deep Learning by Prof. Kosta Derpanis

ImageNet Error Rate (Best of 5 Guesses)



Slide credit: Deep Learning by Prof. Kosta Derpanis

Revolution of Depth

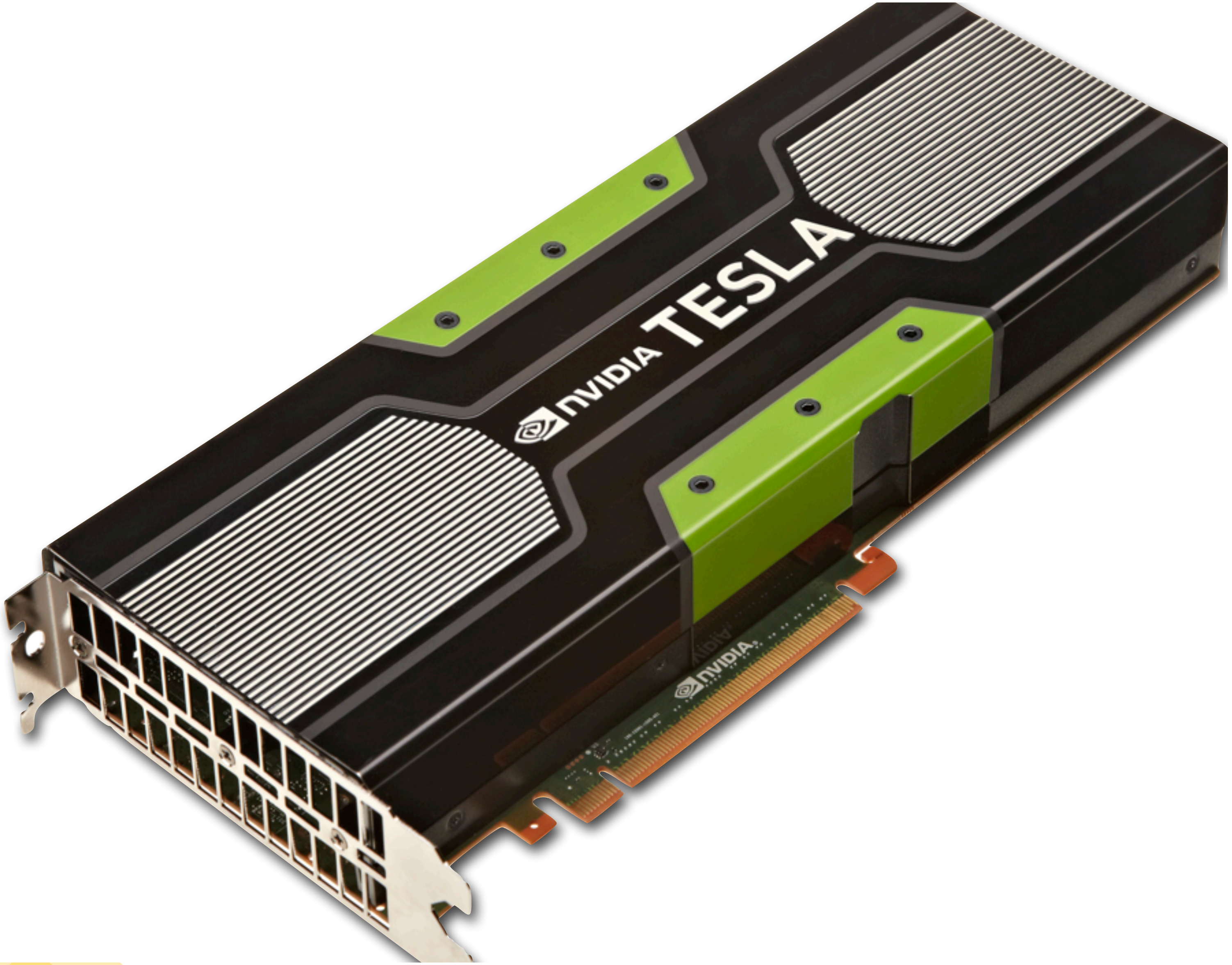


ImageNet Classification top-5 error (%)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

“What’s changed?”

Slide credit: Deep Learning by Prof. Kosta Derpanis


NVIDIA TESLA

Deep Learning Frameworks

Slide credit: Deep Learning by Prof. Kosta Derpanis

mxnet

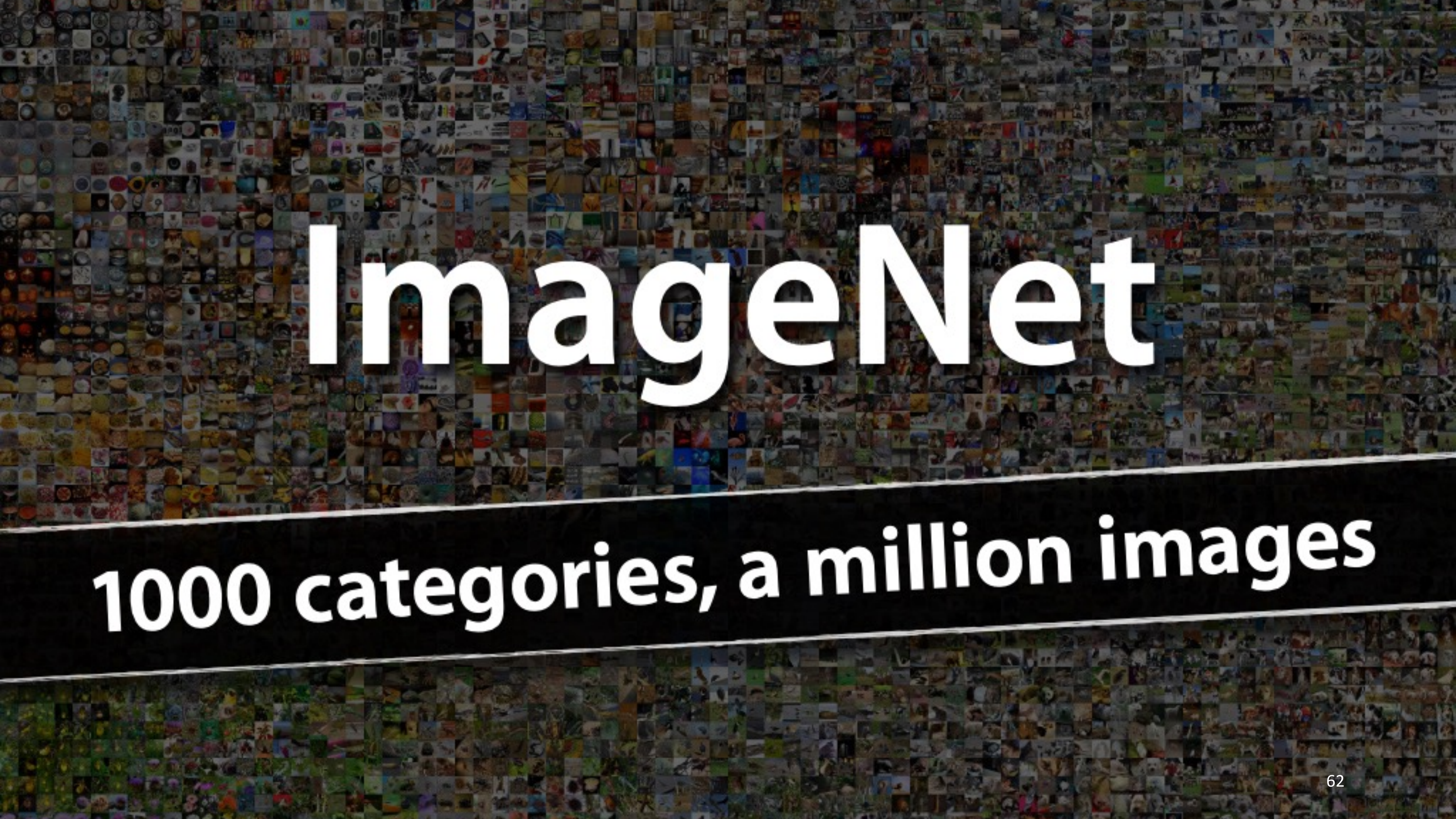
Microsoft CNTK

PyTorch

Chainer

TensorFlow

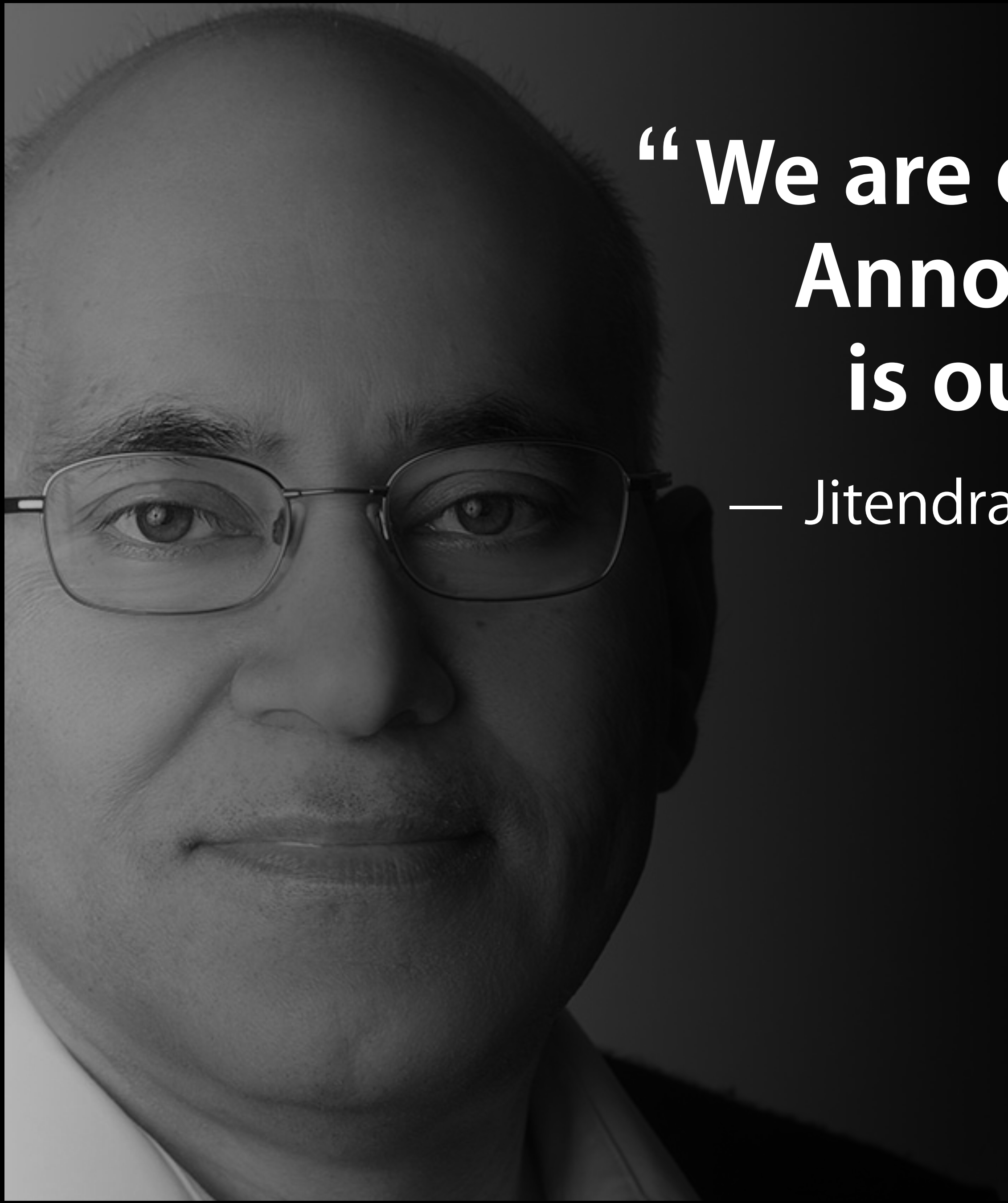
Slide credit: Deep Learning by Prof. Kosta Derpanis



ImageNet

1000 categories, a million images

Slide credit: Deep Learning by Prof. Kosta Derpanis

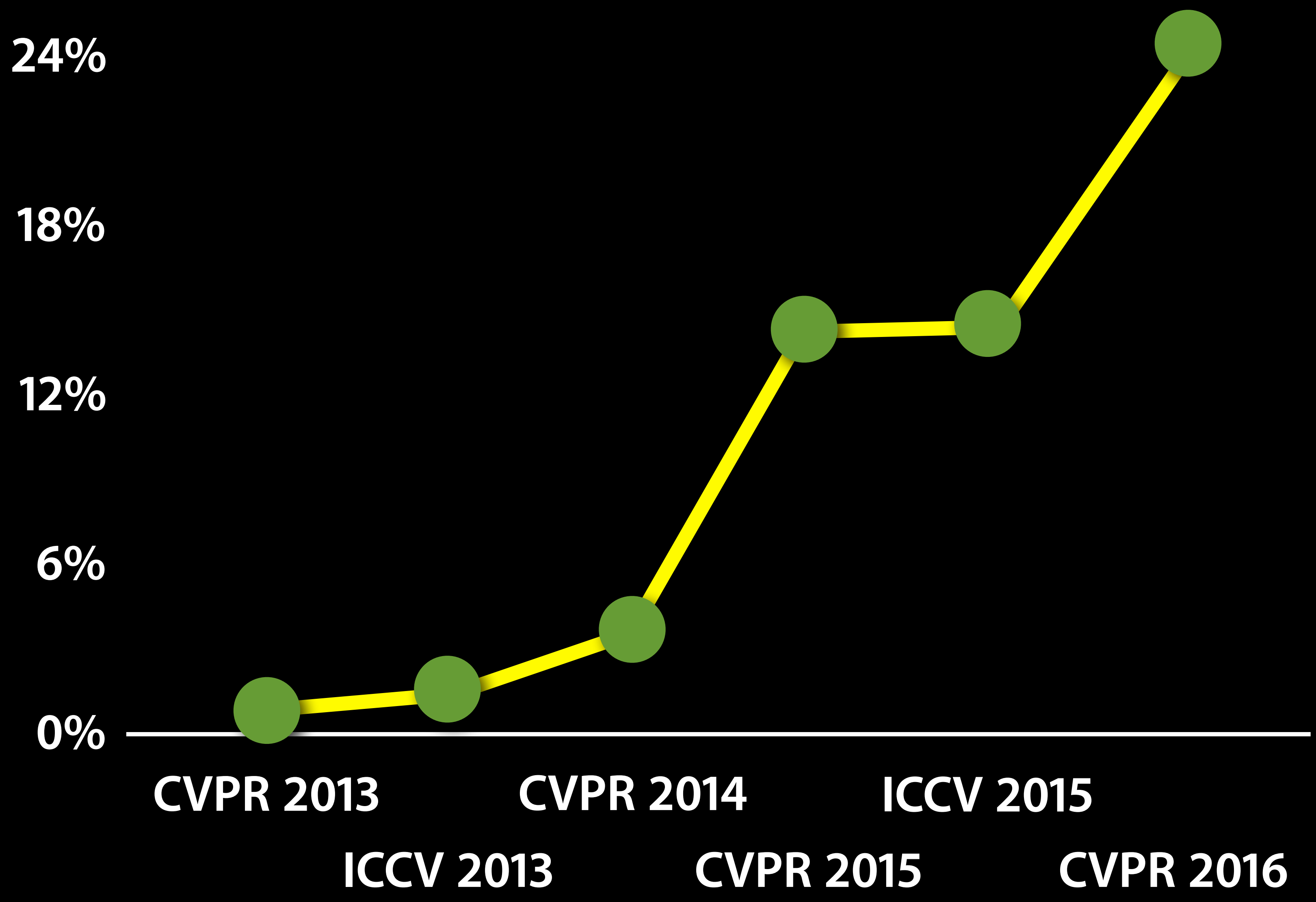


"We are drug addicts! Annotated data is our heroin."

— Jitendra Malik (UC Berkeley)

Slide credit: Deep Learning by Prof. Kosta Derpanis

Deep Learning Takes Over Computer Vision



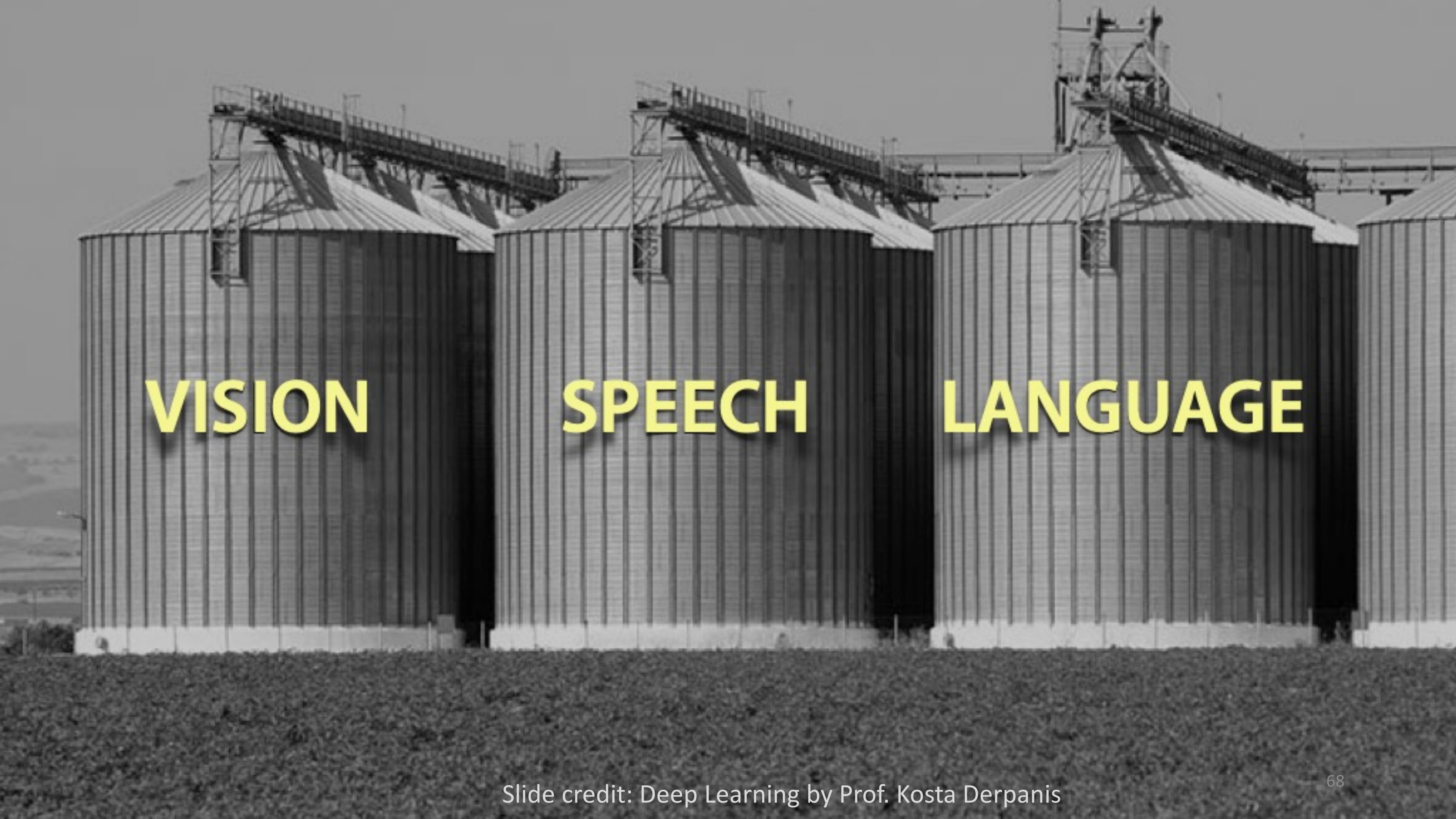
Slide credit: Deep Learning by Prof. Kosta Derpanis

Computer vision industry will grow from \$1.1 billion in 2016 to \$26.2 billion by 2025

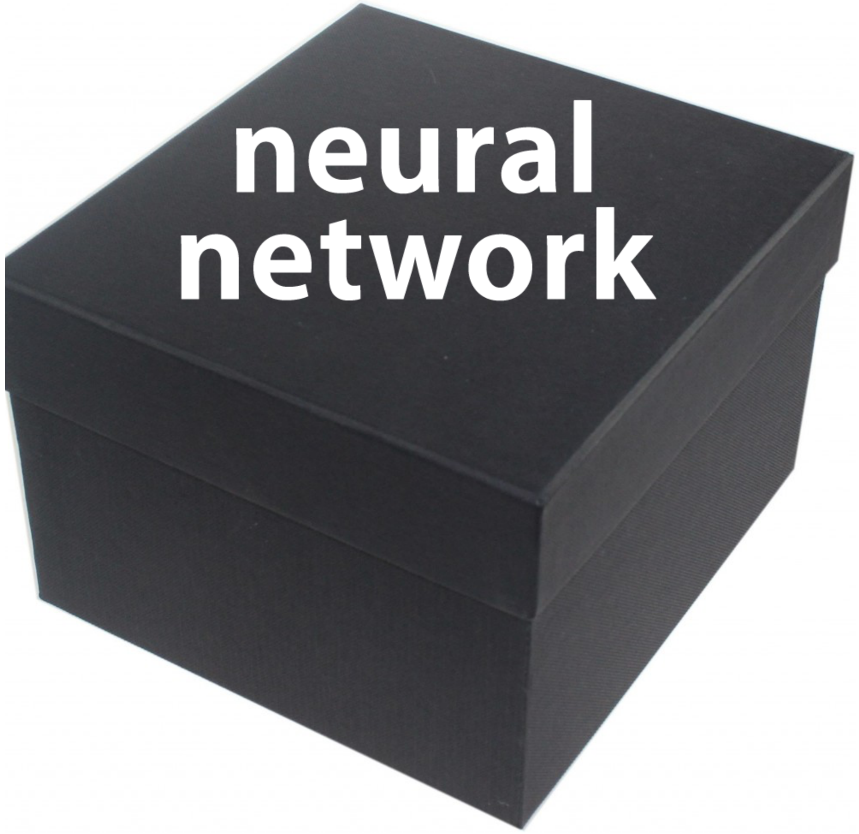
Source: Tractica (2020)

Slide credit: Deep Learning by Prof. Kosta Derpanis

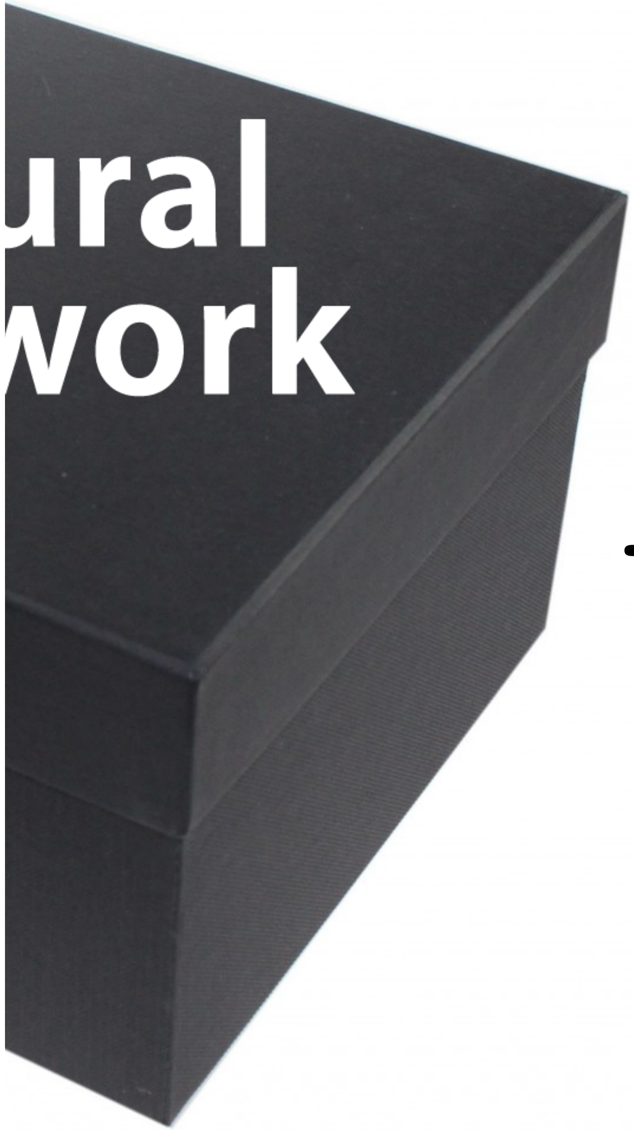
Slide credit: Deep Learning by Prof. Kosta Derpanis



Slide credit: Deep Learning by Prof. Kosta Derpanis



\+ **data**

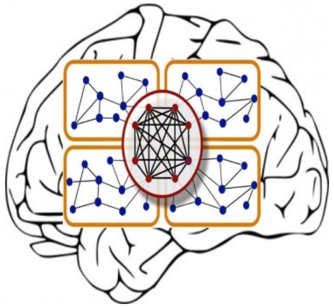


+ **data**

Machine Learning from Big Data: Achievements

YouTube Broadcast Yourself™

Artificial Neural Network



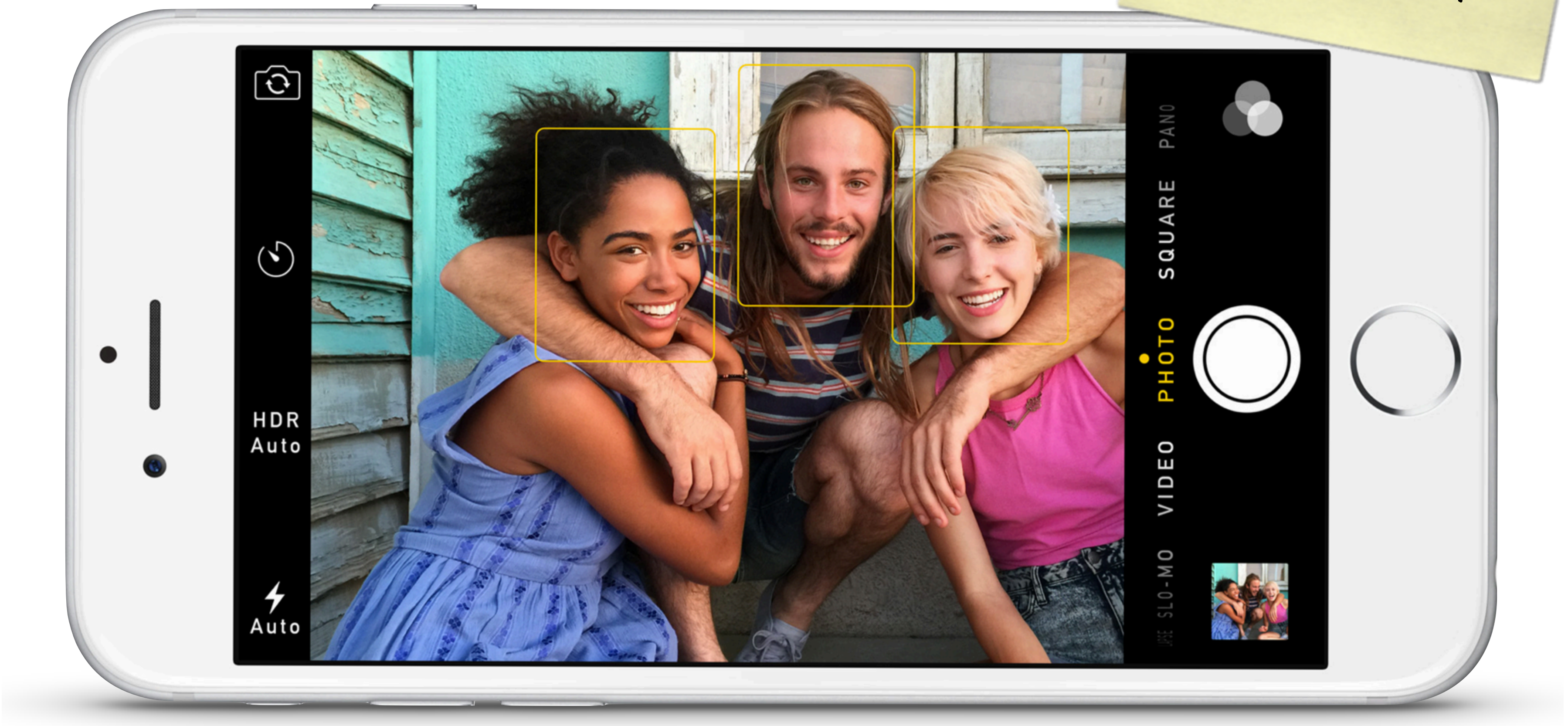
ML watches YouTube for three straight days!
(and learns to recognize cats?!)

http://www.npr.org/2012/06/26/155792609/a-massive-google-network-learns-to-identify

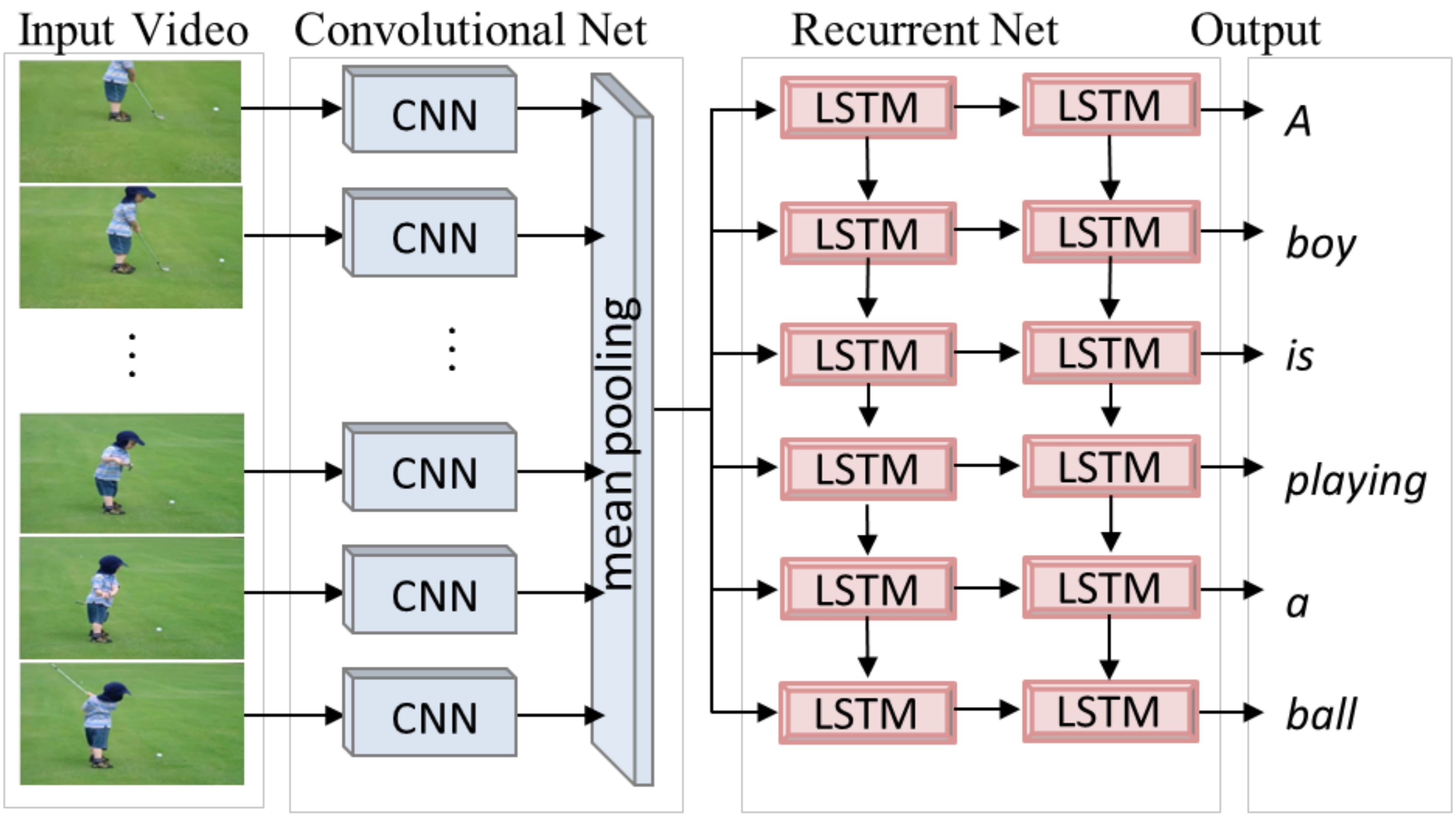
Building High-level Features Using Large Scale Unsupervised Learning
Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeffrey Dean, and Andrew Y. Ng

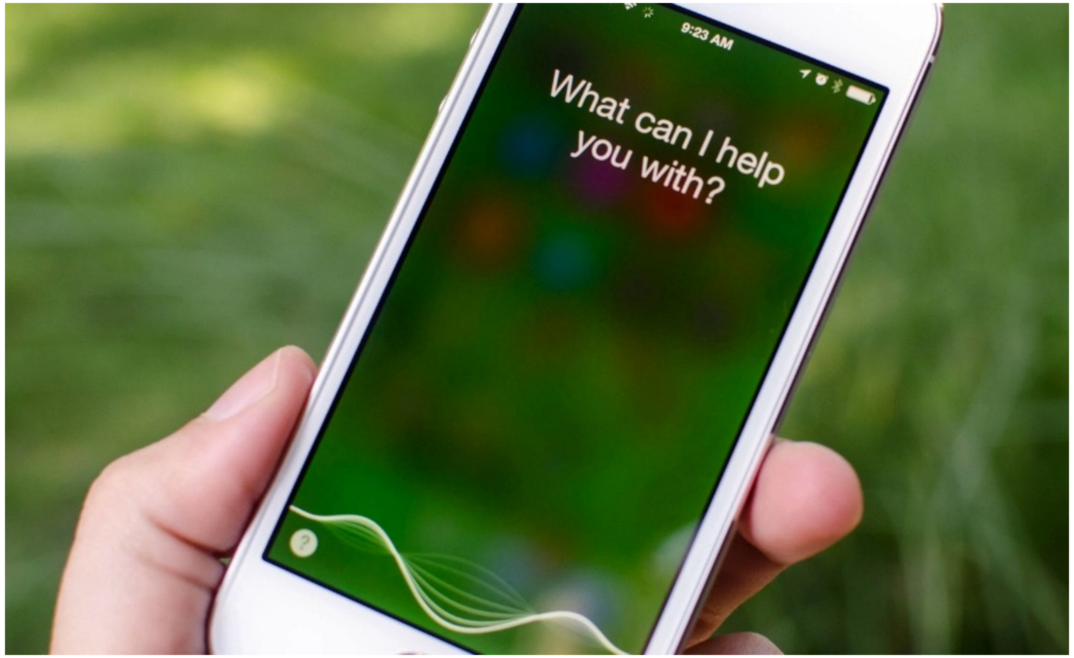
Applications

Face Detection



Slide credit: Deep Learning by Prof. Kosta Derpanis

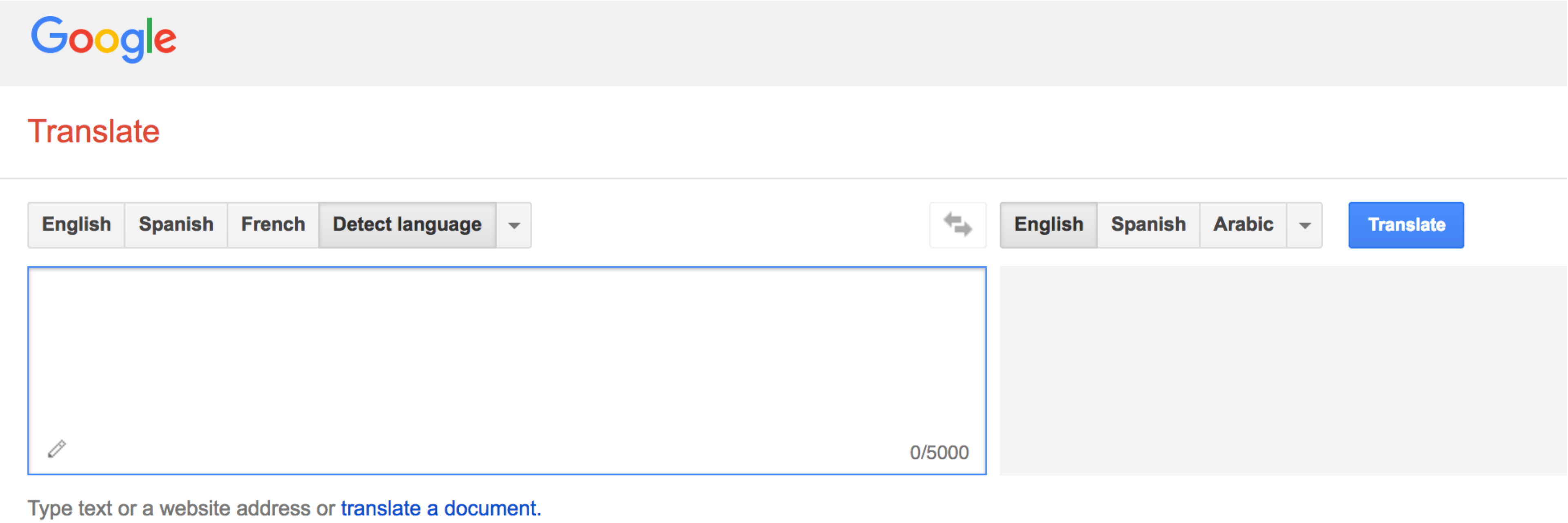

Input Video
Convolutional Net
Recurrent Net
Output
CNN
CNN
CNN
CNN
CNN
mean pooling
LSTM
A
boy
is
playing
a
ball


9:23 AM
What can I help
you with?


amazon

WaveNet in the Google Assistant

- https://deepmind.com/blog/wavenet-generative-model-raw-audio/
- https://www.youtube.com/watch?reload=9&v=D5VN56jQMWM



The spirit is willing but the flesh is weak → The vodka is good but the meat is rotten

AI Art

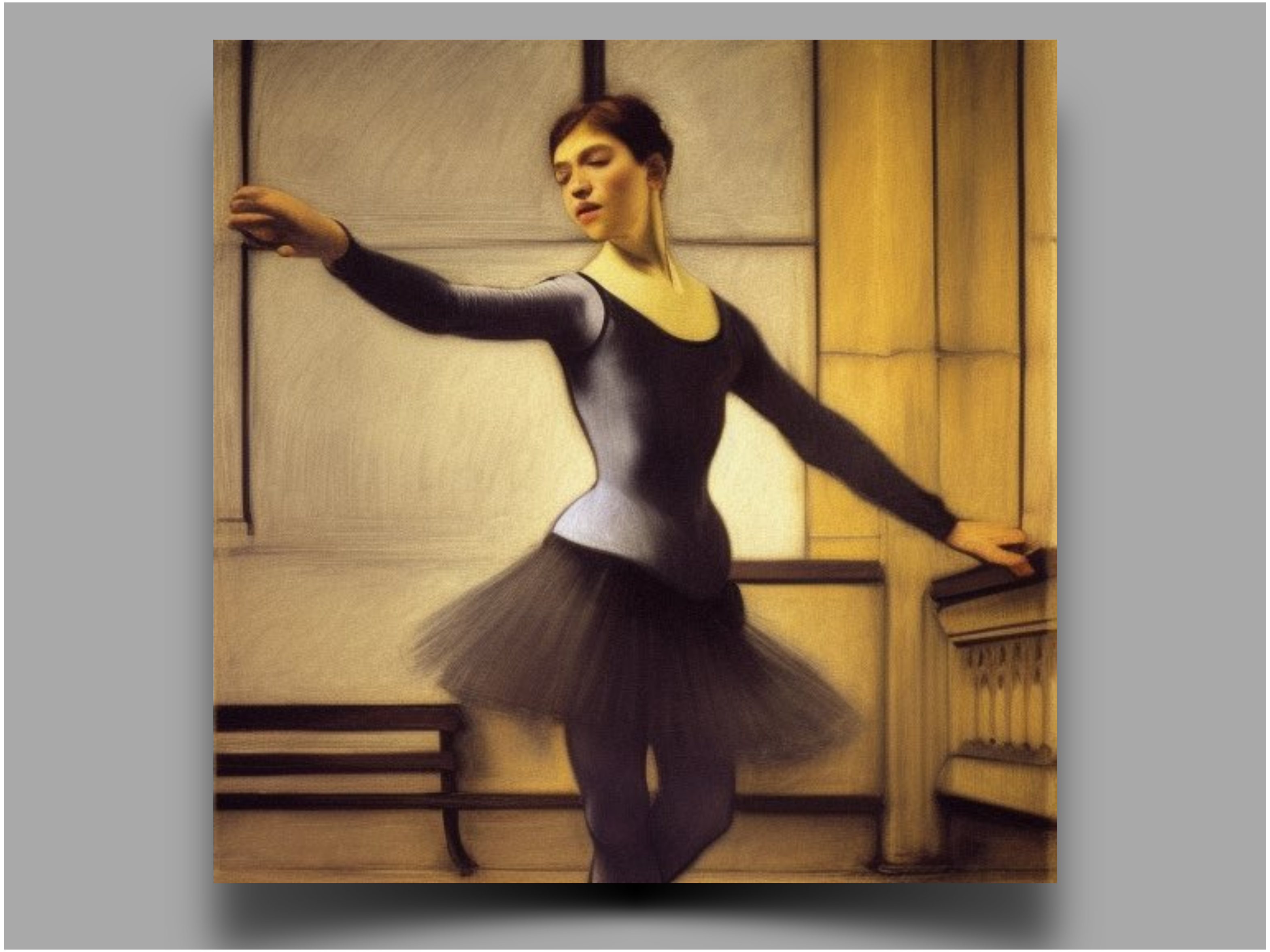
Slide credit: Deep Learning by Prof. Kosta Derpanis

Text to Image with Diffusion Models

- Teddy bears swimming at the Olympics 400m butterfly event.

An astronaut riding a horse in a photorealistic style

Slide credit: Deep Learning by Prof. Kosta Derpanis



Slide credit: Deep Learning by Prof. Kosta Derpanis


LEE SEDOL

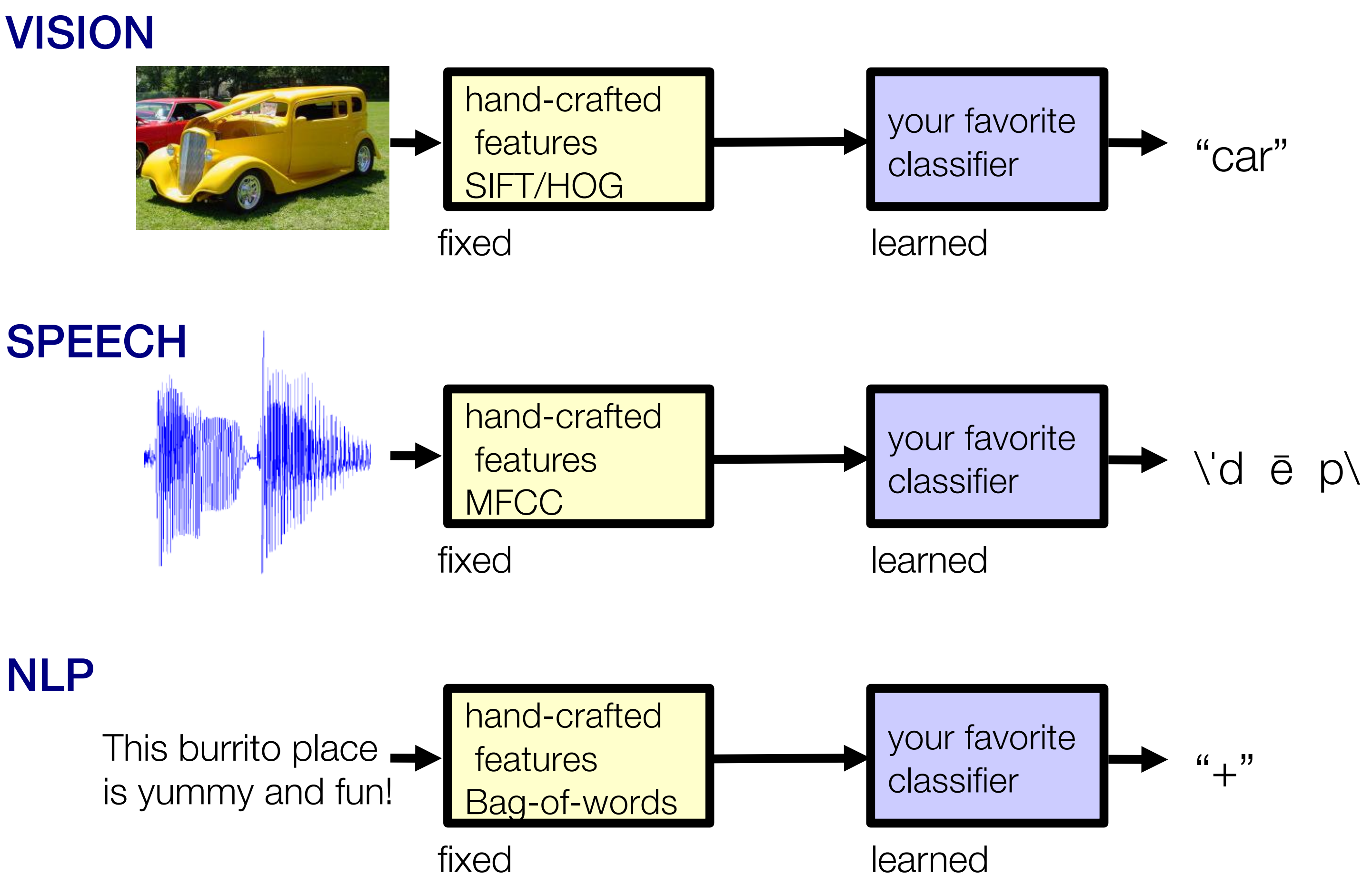
Deep Reinforcement Learning for Atari Games

- https://www.youtube.com/watch?v=V1eYniJ0Rnk

Three key ideas

- **(Hierarchical) Compositionality**
  - Cascade of non-linear transformations
  - Multiple layers of representations
- End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extract
- Distributed Representations
  - No single neuron "encodes" everything
  - Groups of neurons work together

Traditional Machine Learning



Slide credit: Aykut Erdem // Koç University // Fall 2022

Hierarchical Compositionality

VISION

pixels → edge → texton → motif → part → object

SPEECH

sample → spectral band → formant → motif → phone → word

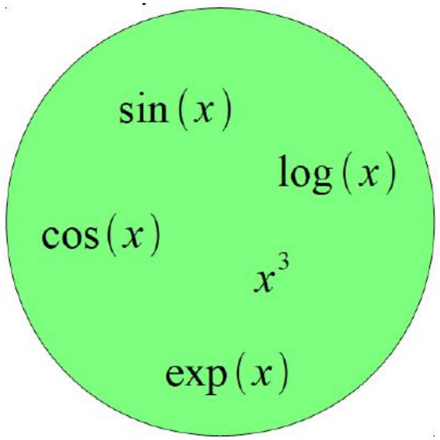
NLP

character → word → NP/VP/.. → clause → sentence → story

Slide credit: Aykut Erdem // Koç University // Fall 2022

Building A Complicated Function

Given a library of simple functions

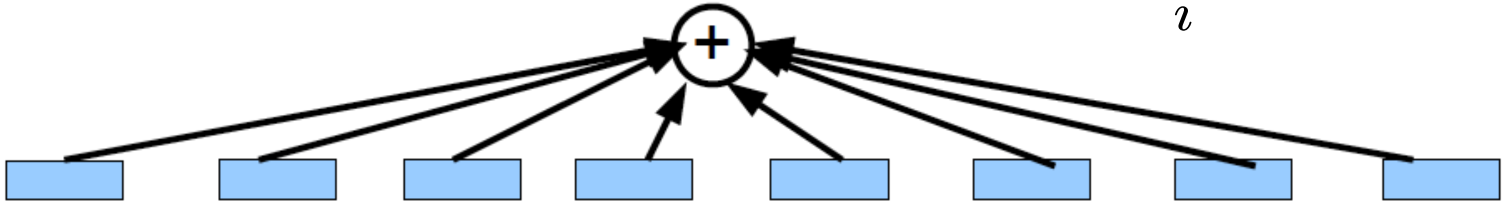


Compose into a complicate function

Idea 1: Linear Combinations

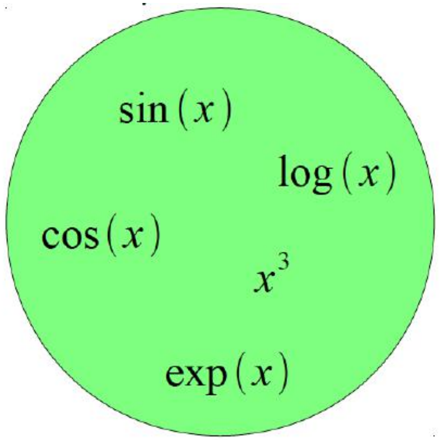
- Boosting
- Kernels
- …

$$f(x) = \sum_i \alpha_i g_i(x)$$



Building A Complicated Function

Given a library of simple functions

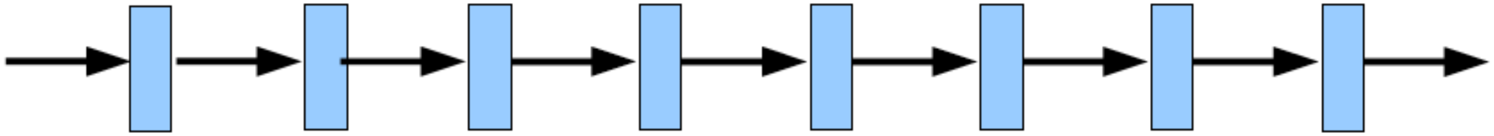


Compose into a complicate function

Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms…

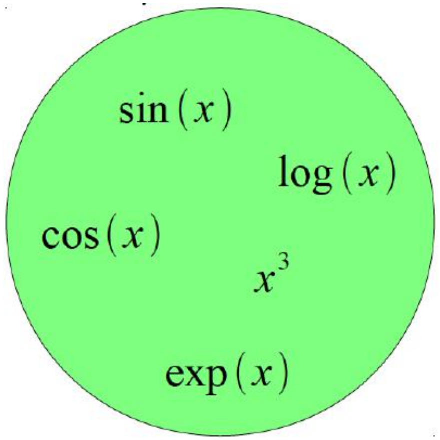
$$f(x) = g_1(g_2(\ldots(g_n(x)\ldots))$$



Slide credit: Aykut Erdem // Koç University // Fall 2022

Building A Complicated Function

Given a library of simple functions

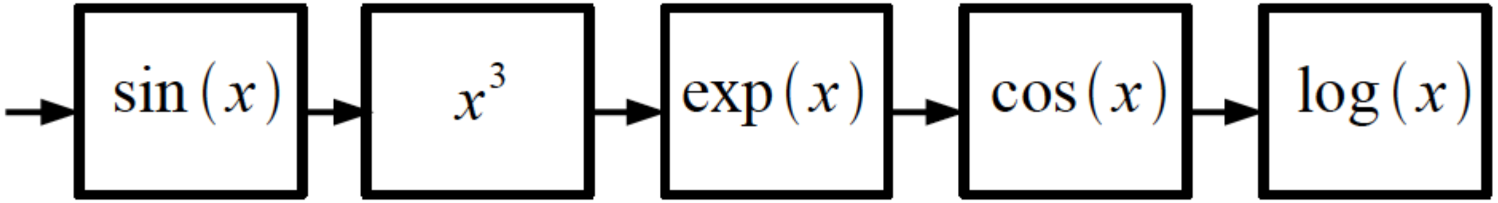


Compose into a complicate function

Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms…

$$f(x) = \log(\cos(\exp(\sin^3(x))))$$



Deep Learning = Hierarchical Compositionality

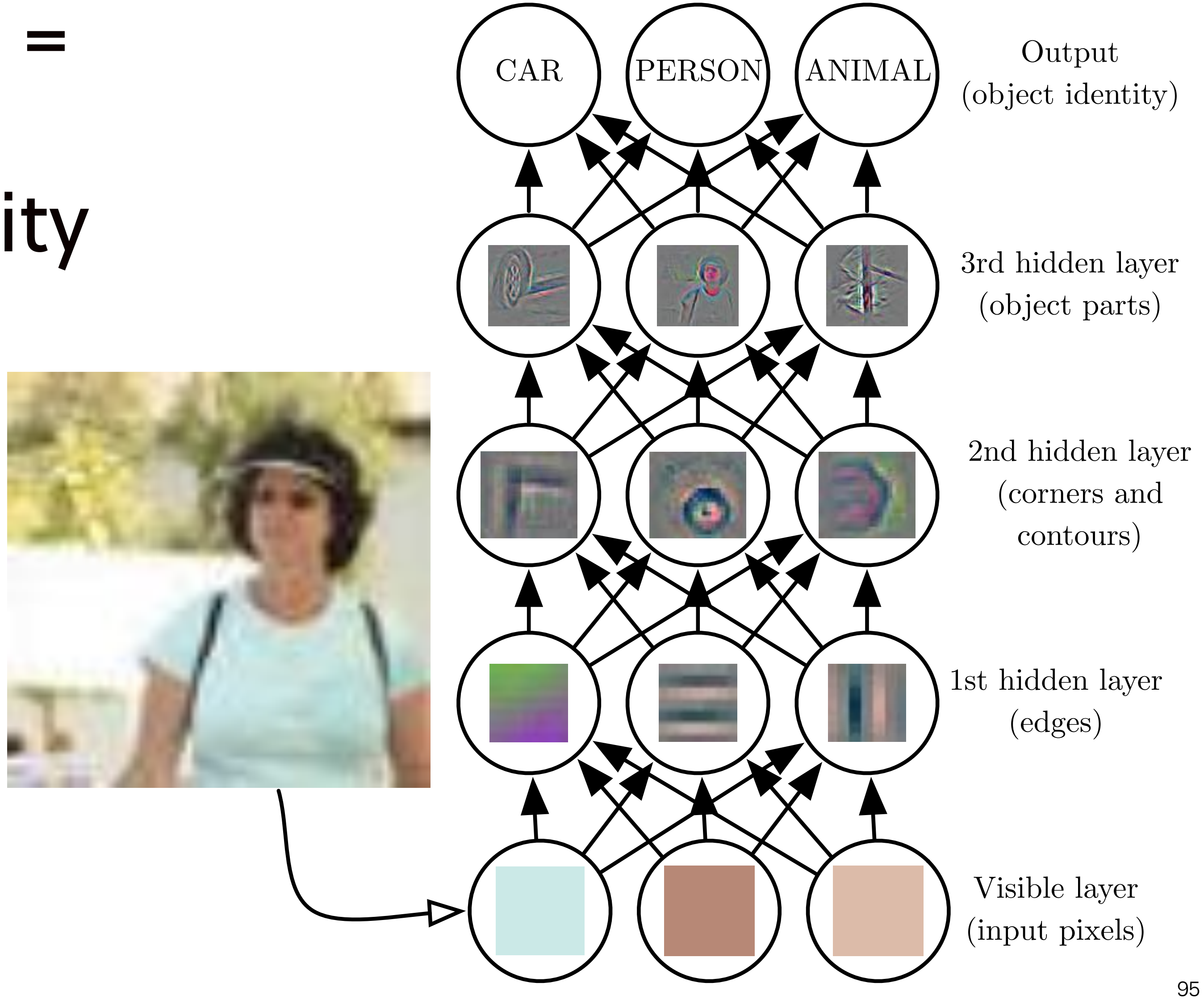
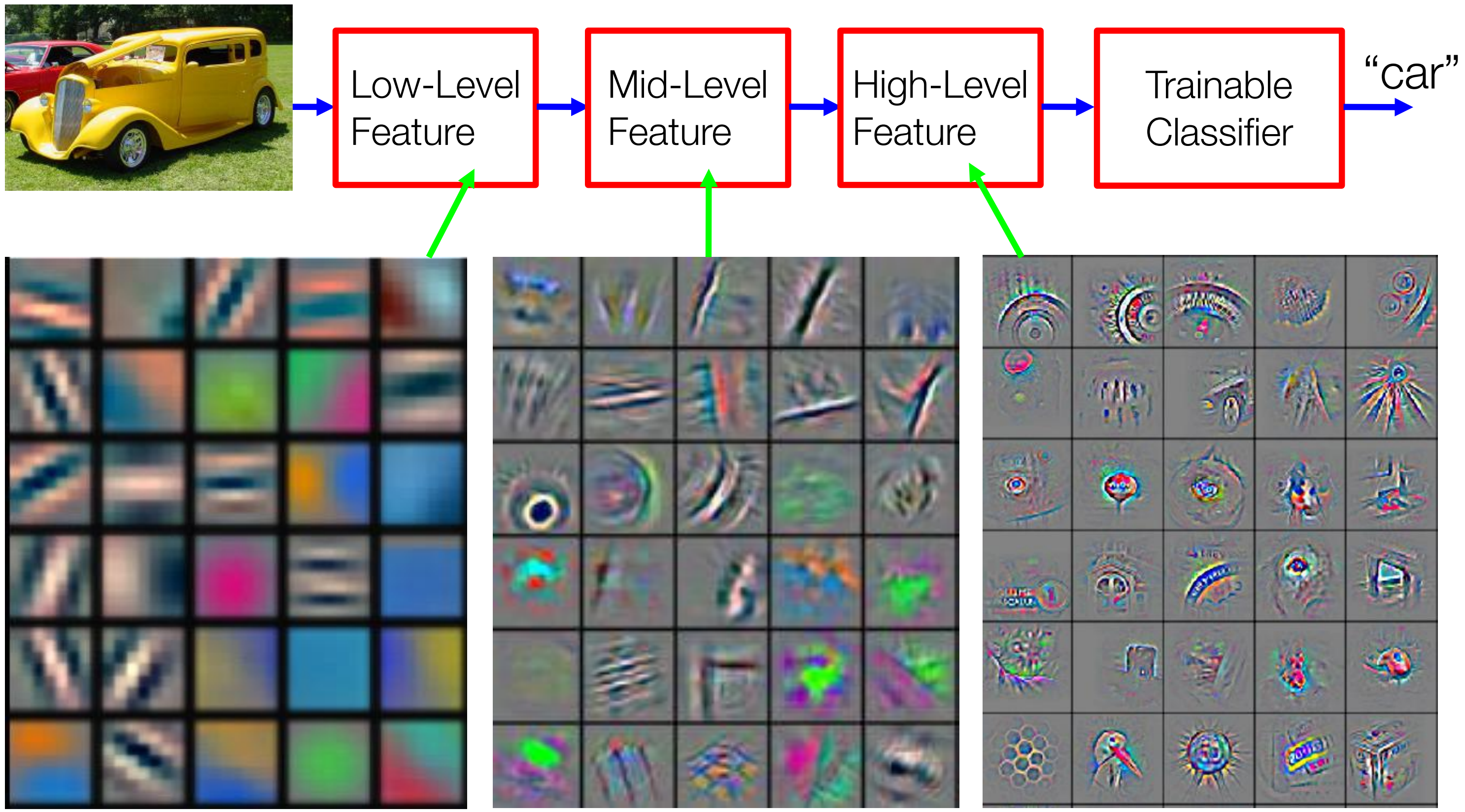


Image credit: Ian Goodfellow

95

Slide credit: Aykut Erdem // Koç University // Fall 2022

Deep Learning = Hierarchical Compositionality



M.D. Zeiler and R. Fergus, "**Visualizing and Understanding Convolutional Networks**", In ECCV 2014

Three key ideas

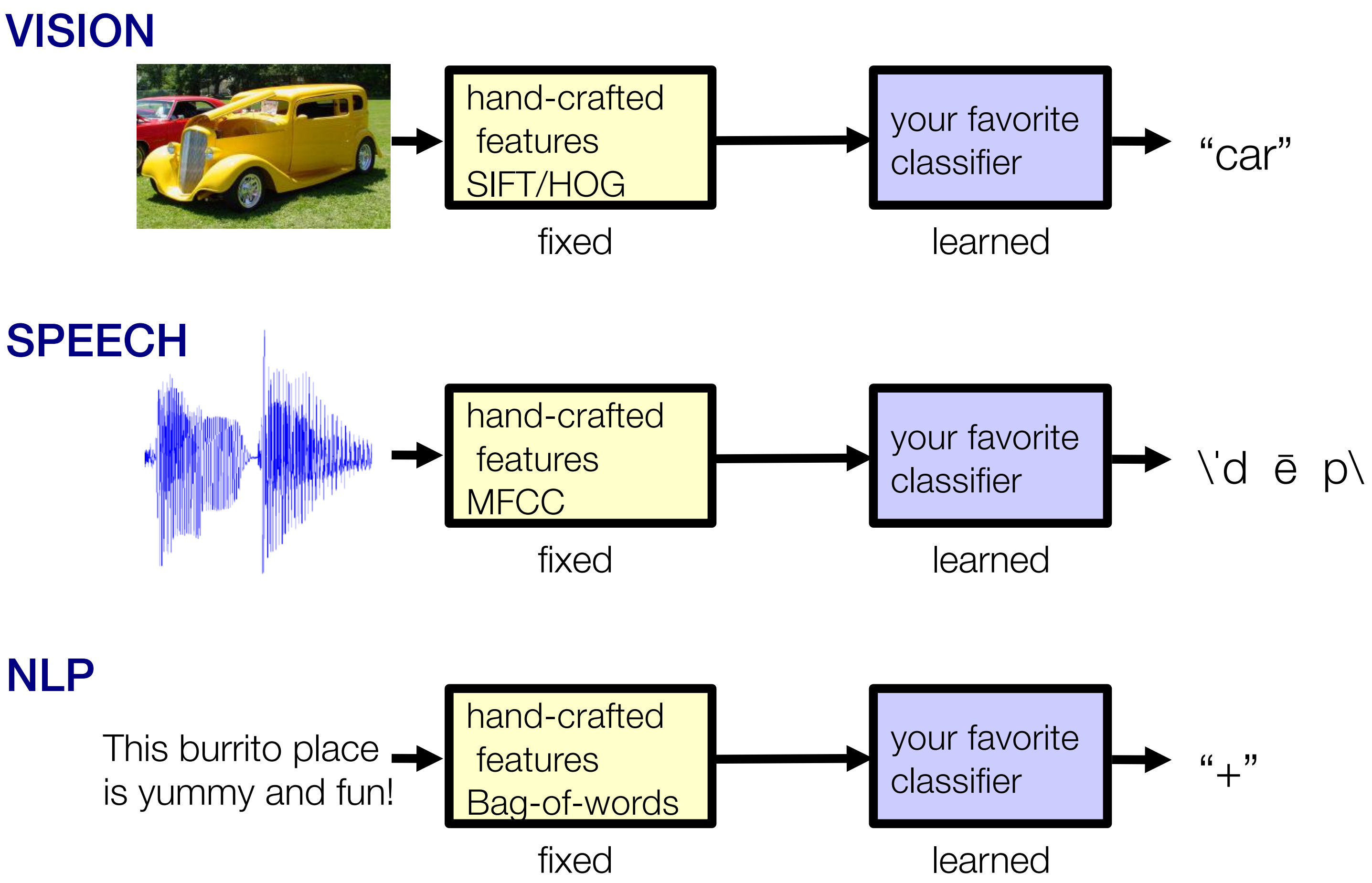
- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- **End-to-End Learning**
  - Learning (goal-driven) representations
  - Learning to feature extract

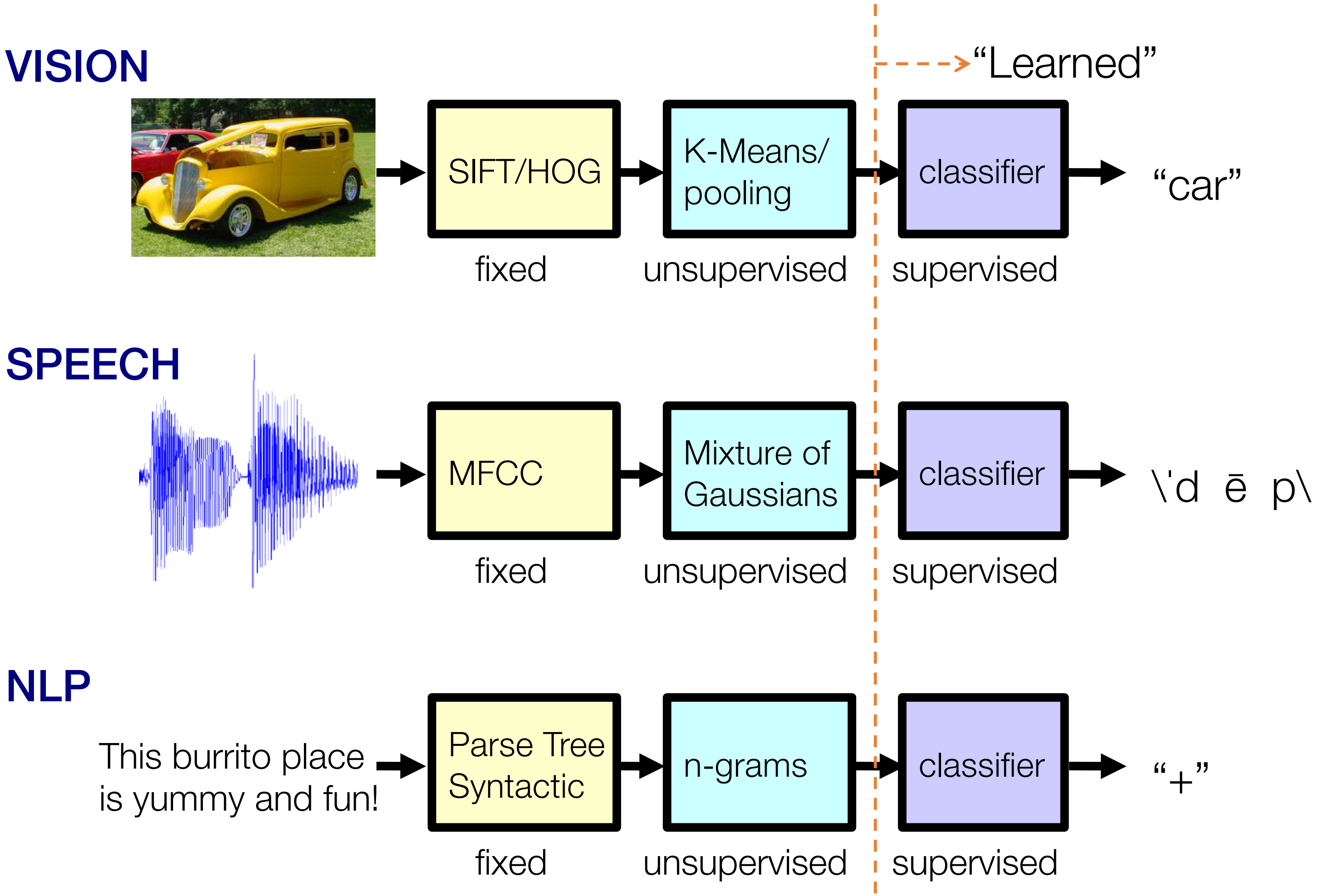
- Distributed Representations
  - No single neuron "encodes" everything
  - Groups of neurons work together

Slide credit: Aykut Erdem // Koç University // Fall 2022

Traditional Machine Learning



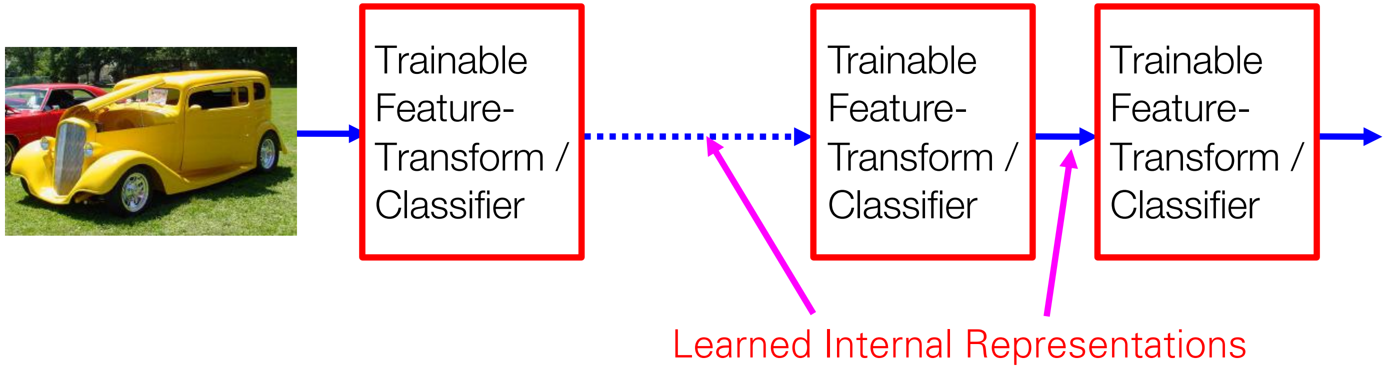
Deep Learning = End-to-End Learning



101

Deep Learning = End-to-End Learning

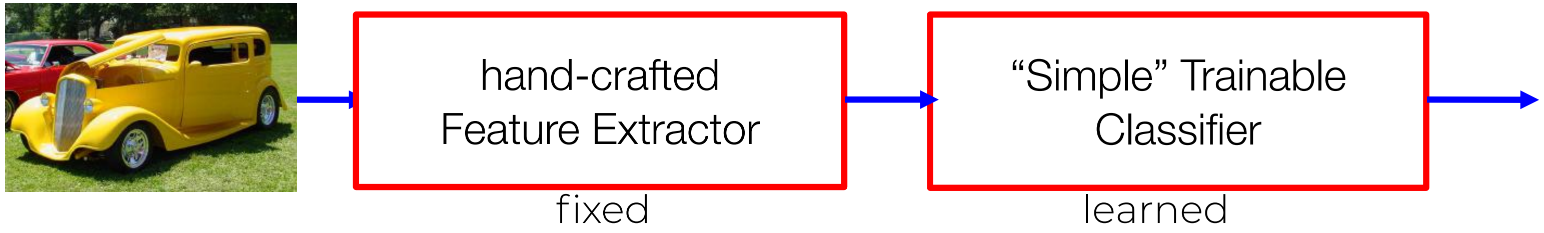
- A hierarchy of trainable feature transforms
  - Each module transforms its input representation into a higher-level one.
  - High-level features are more global and more invariant
  - Low-level features are shared among categories



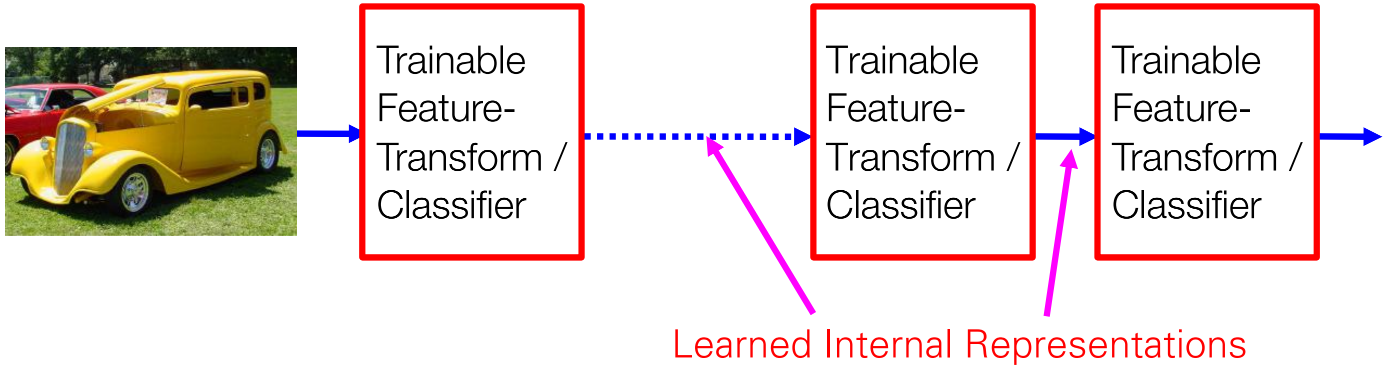
102

“Shallow” vs Deep Learning

- “Shallow” models



- Deep models



103

Three key ideas

- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

- End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extract

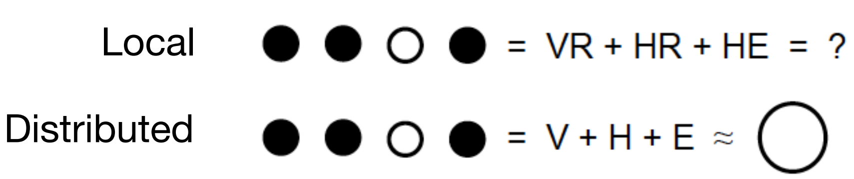
- **Distributed Representations**
  - No single neuron "encodes" everything
  - Groups of neurons work together

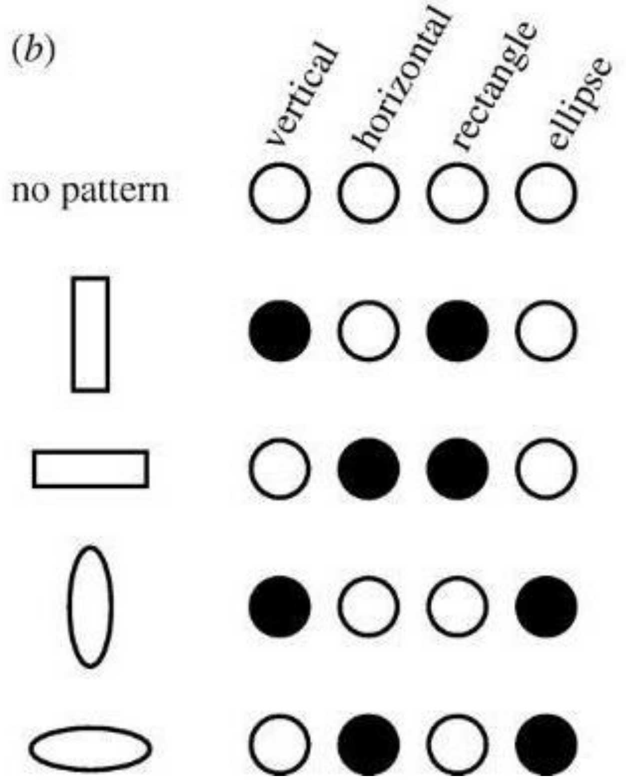
104

Slide credit: Aykut Erdem // Koç University // Fall 2022

Distributed Representations

- Each neuron must represent something, so this must be a local representation.
- **Distributed representation** means a many-to-many relationship between two types of representation (such as concepts and neurons).
  - Each concept is represented by many neurons
  - Each neuron participates in the representation of many concepts





Slide credit: Geoff Hinton

Image credit: Moontae Lee

Power of distributed representations!

Scene Classification

bedroom

mountain

- Possible internal representations:
  - Objects
  - Scene attributes
  - Object parts
  - Textures

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba "**Object Detectors Emerge in Deep Scene CNNs**", ICLR 2015

Slide credit: Bolei Zhou