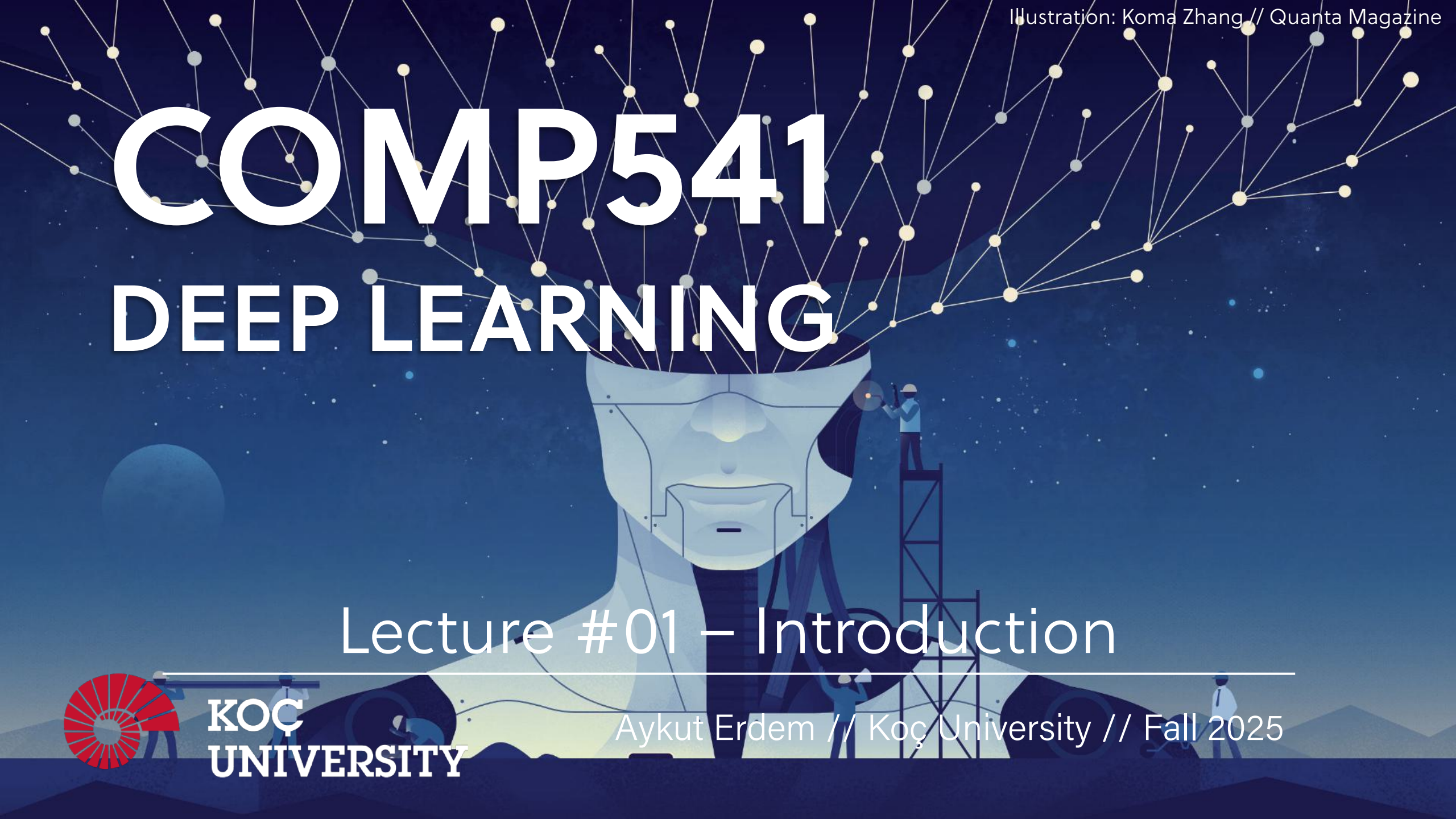


COMP541

DEEP LEARNING

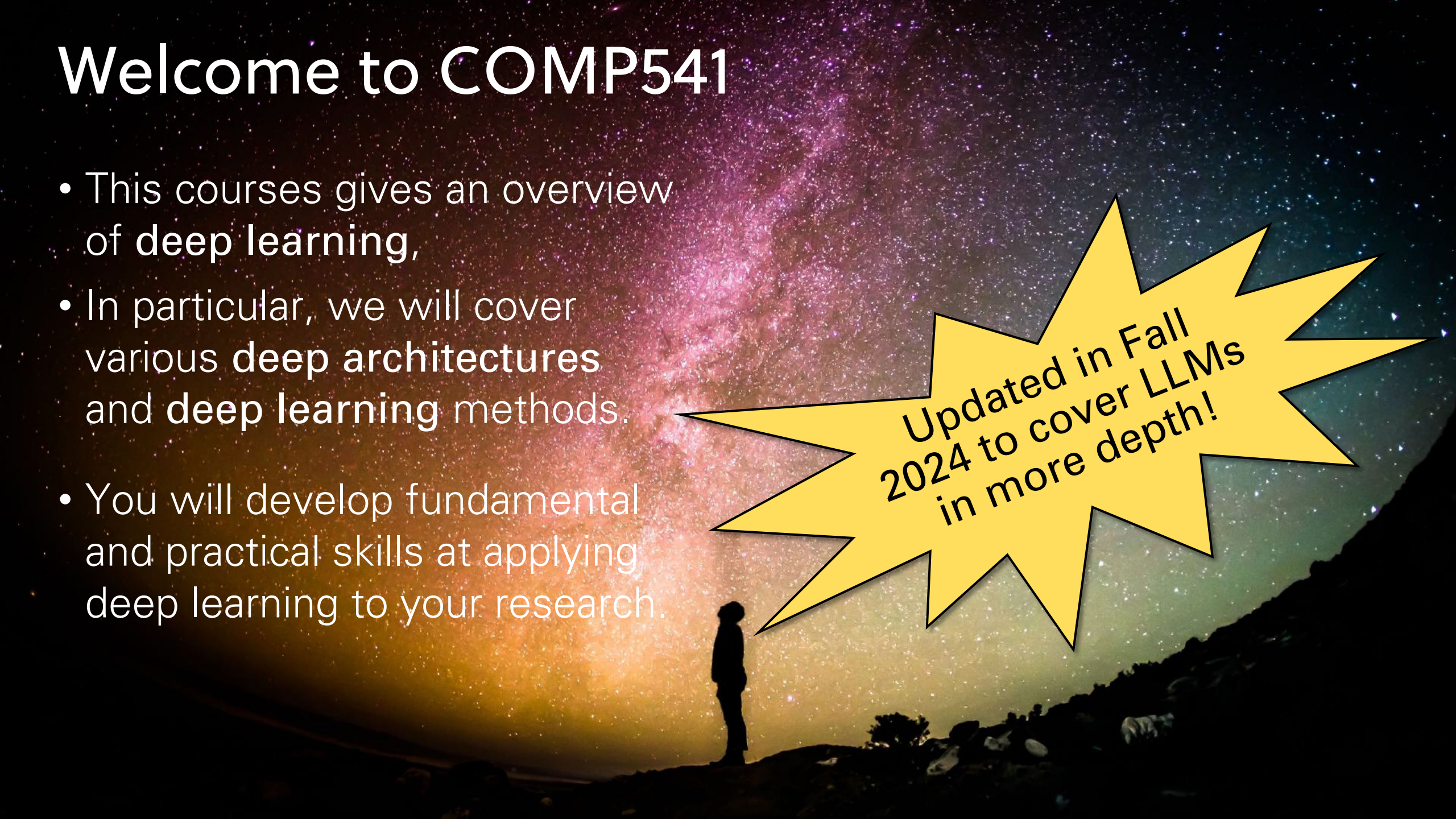
Lecture #01 – Introduction

KOÇ UNIVERSITY

Aykut Erdem // Koç University // Fall 2025

Welcome to COMP541

- This courses gives an overview of **deep learning**,
- In particular, we will cover various **deep architectures** and **deep learning** methods.
- You will develop fundamental and practical skills at applying deep learning to your research.



A little about me...

Koç University
Associate Professor
2020-now

Hacettepe University
Associate Professor
2010-2020

Universitá Ca' Foscari di Venezia
Post-doctoral Researcher
2008-2010

Middle East Technical University
1997-2008
Ph.D., 2008
M.Sc., 2003
B.Sc., 2001

MIT
Fall 2007
Visiting Student

VirginiaTech
Visiting Research Scholar
Summer 2006

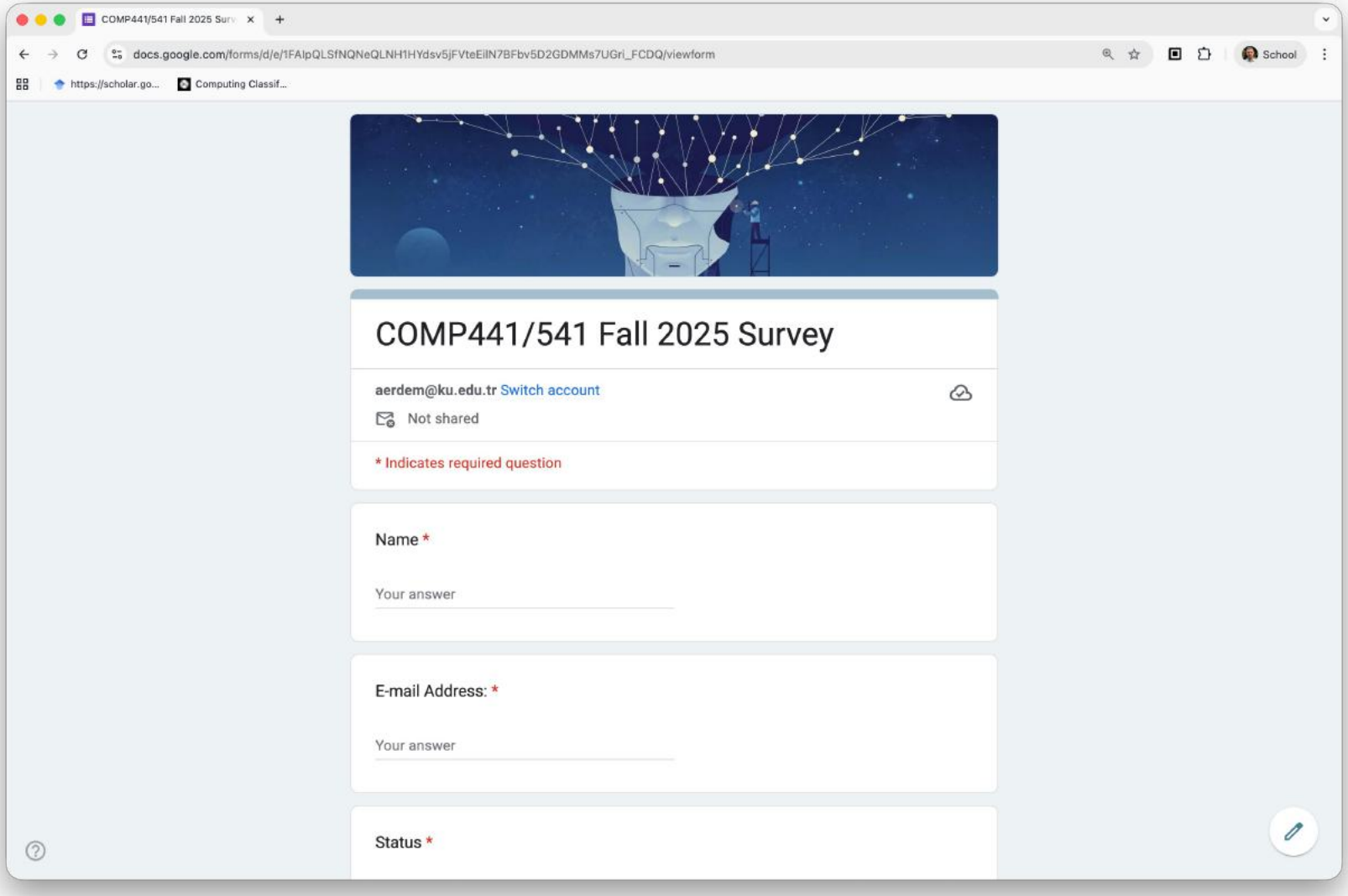
The broad goal of my research is to explore better ways to **understand**, **interpret**, and **manipulate** visual data.

Research Interests

- Deep Learning
- Generative AI
- Computer Vision
- Language Understanding

https://aykuterdem.github.io

Now, what about you?



https://forms.gle/sHgeXiYiPQaqkcGy7

Course Logistics

Course Information

Lectures Tuesday and Thursday 14:30-15:40 (SNA 159)
PS Thursday 17:30-18:40 (CASE Z08)

Instructor Aykut Erdem
TAs Andrew Bond & Hakan Capuk.



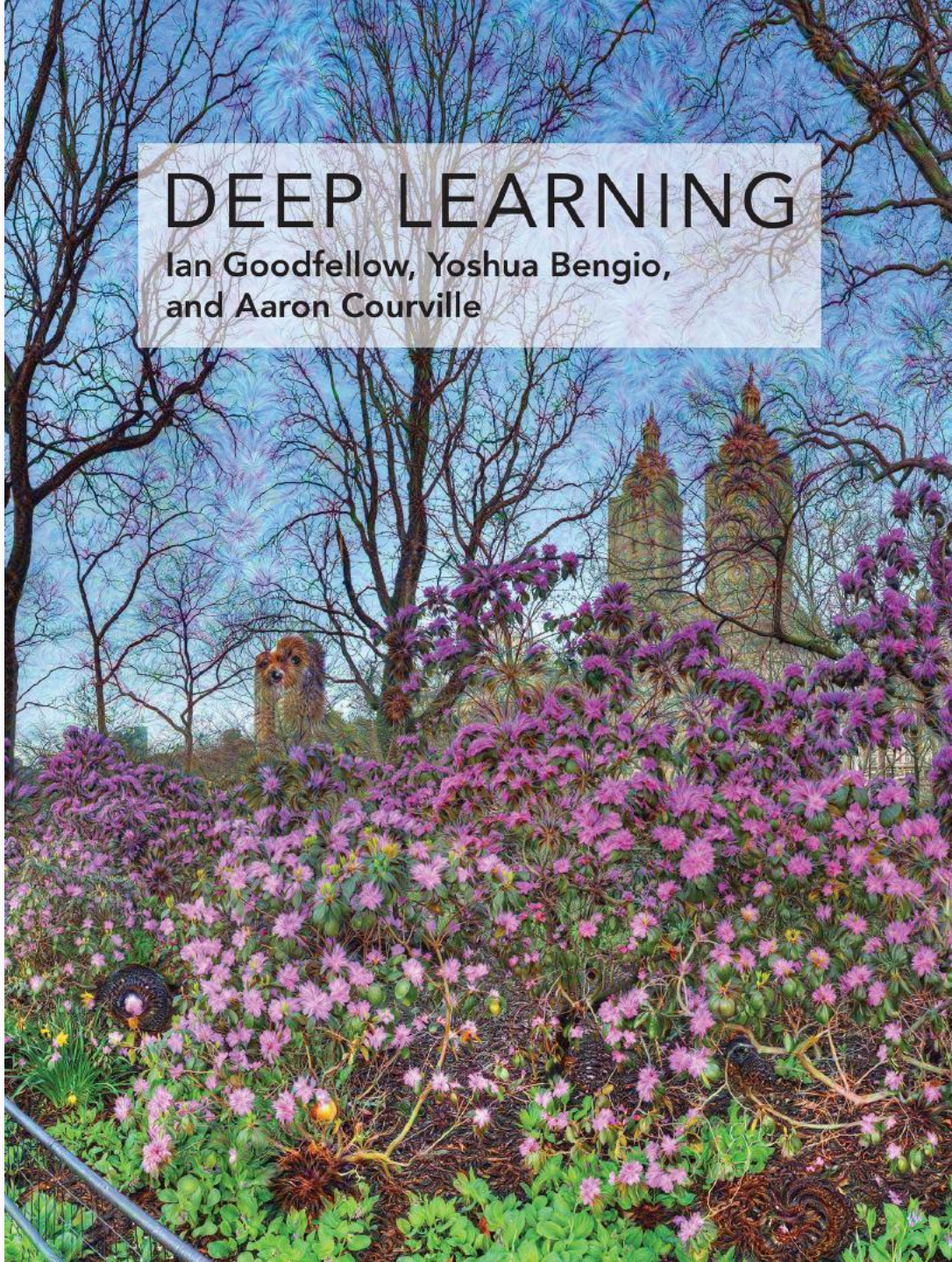
Website https://aykuterdem.github.io/classes/comp541.f25/

- KUHub Learn for course related announcements and collecting and grading your submissions

Textbook

- Goodfellow, Bengio, and Courville, Deep Learning, MIT Press, 2016 (draft available online)

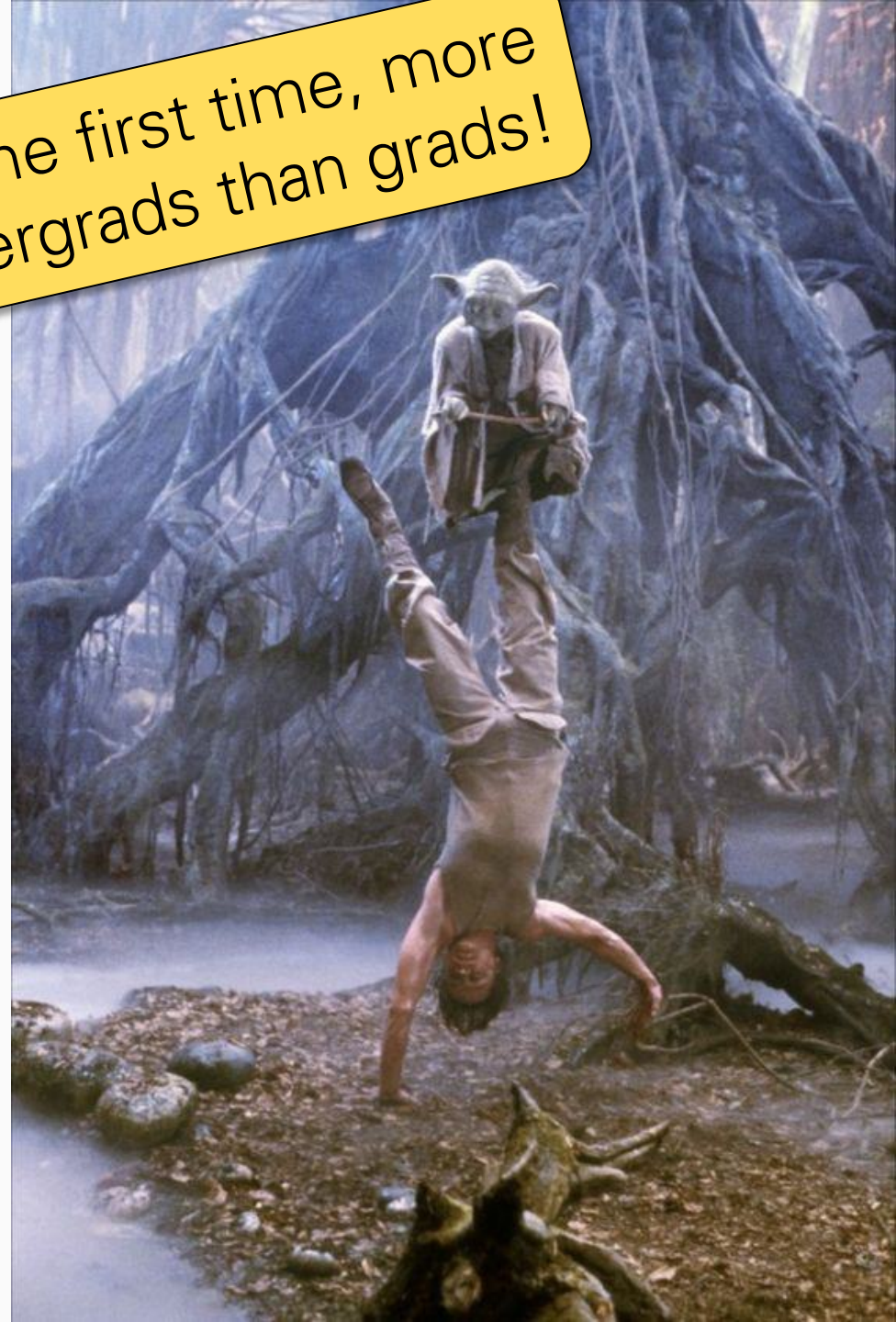
- In addition, we will extensively use online materials (video lectures, blog posts, surveys, papers, etc.)



Instruction style

- Students are responsible for studying and keeping up with the course material outside of class time.
  - Reading particular book chapters, papers or blogs, or
  - Watching some video lectures.

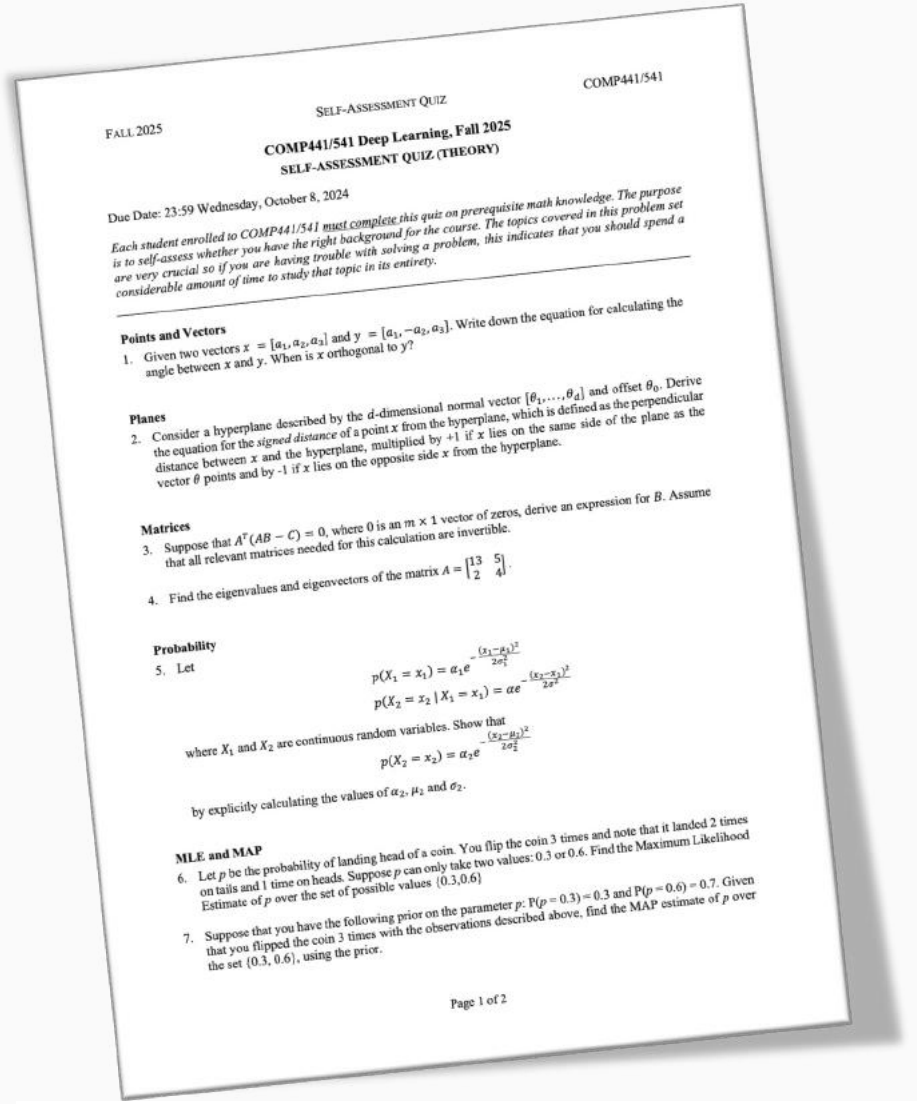
- After the first four lectures, each week students will present papers related to the topics of the previous week.
  - Weekly paper reviews will be prepared by all the students



Prerequisites

- Calculus and linear algebra
  - Derivatives,
  - Matrix operations
- Probability and statistics
- Machine learning
- Programming

Read Chapter 2-4
of the Deep Learning textbook for a quick review.



FALL 2025 SELF-ASSESSMENT QUIZ COMP441/541

COMP441/541 Deep Learning, Fall 2025

SELF-ASSESSMENT QUIZ (THEORY)

Due Date: 23:59 Wednesday, October 8, 2024

Each student enrolled to COMP441/541 must complete this quiz on prerequisite math knowledge. The purpose is to self-assess whether you have the right background for the course. The topics covered in this problem set are very crucial so if you are having trouble with solving a problem, this indicates that you should spend a considerable amount of time to study that topic in its entirety.

Points and Vectors

1. Given two vectors $x = [a_1, a_2, a_3]$ and $y = [a_1, -a_2, a_3]$. Write down the equation for calculating the angle between x and y. When is x orthogonal to y?

Planes

2. Consider a hyperplane described by the d-dimensional normal vector $[\theta_1, \ldots, \theta_d]$ and offset θ_0. Derive the equation for the *signed distance* of a point x from the hyperplane, which is defined as the perpendicular distance between x and the hyperplane, multiplied by +1 if x lies on the same side of the plane as the vector θ points and by -1 if x lies on the opposite side x from the hyperplane.

Matrices

3. Suppose that $A^T(AB - C) = 0$, where 0 is an $m \times 1$ vector of zeros, derive an expression for B. Assume that all relevant matrices needed for this calculation are invertible.

4. Find the eigenvalues and eigenvectors of the matrix $A = \begin{bmatrix} 13 & 5 \\ 2 & 4 \end{bmatrix}$.

Probability

5. Let

$$p(X_1 = x_1) = \alpha_1 e^{-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}}$$

$$p(X_2 = x_2 \mid X_1 = x_1) = \alpha e^{-\frac{(x_2 - \mu_2)^2}{2\sigma^2}}$$

where X_1 and X_2 are continuous random variables. Show that

$$p(X_2 = x_2) = \alpha_2 e^{-\frac{(x_2 - \mu_2)^2}{2\sigma_2^2}}$$

by explicitly calculating the values of α_2, μ_2 and σ_2.

MLE and MAP

6. Let p be the probability of landing head of a coin. You flip the coin 3 times and note that it landed 2 times on tails and 1 time on heads. Suppose p can only take two values: 0.3 or 0.6. Find the Maximum Likelihood Estimate of p over the set of possible values {0.3,0.6}

7. Suppose that you have the following prior on the parameter p: $P(p = 0.3) = 0.3$ and $P(p = 0.6) = 0.7$. Given that you flipped the coin 3 times with the observations described above, find the MAP estimate of p over the set {0.3, 0.6}, using the prior.

Page 1 of 2

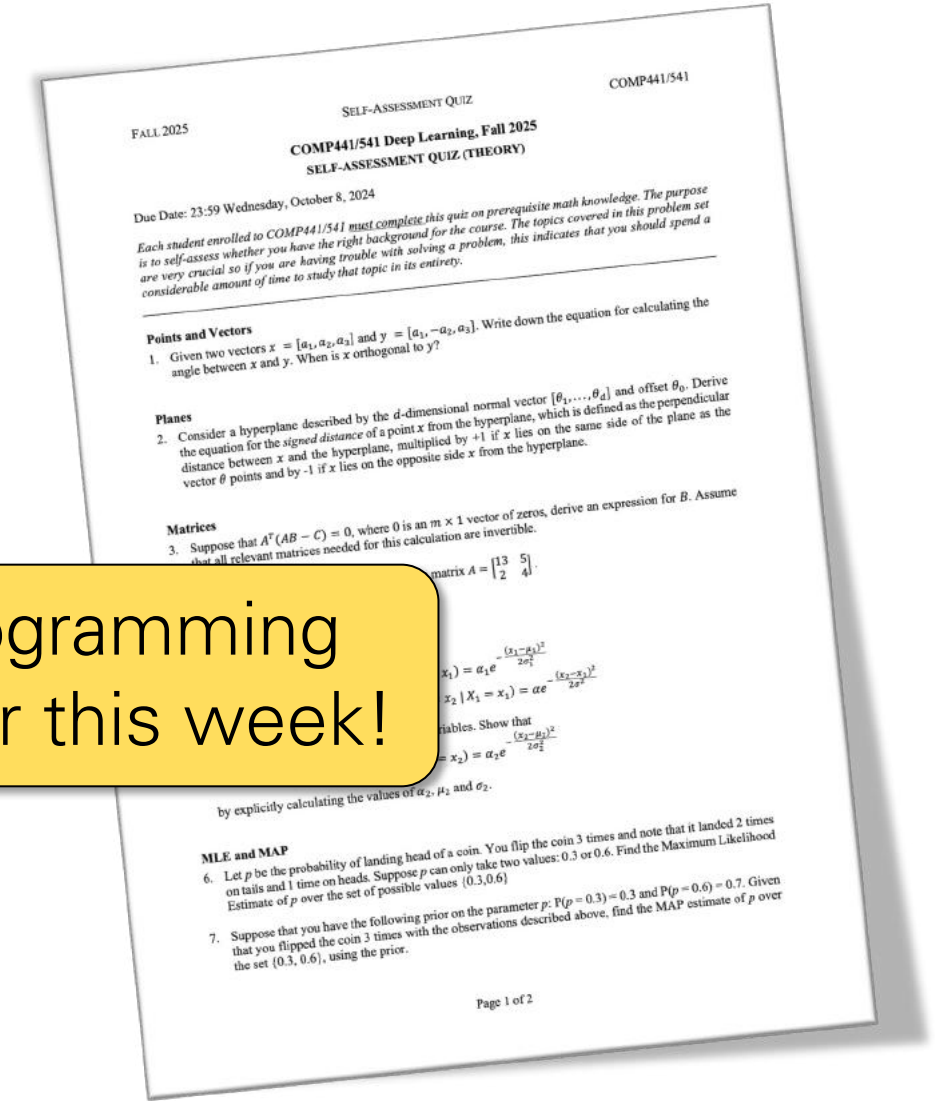
Self-Assessment Quiz (Theory)
Due Date: October 8 (23:59).

Each student enrolled to COMP441/541 must complete and pass this quiz!

Prerequisites

- Calculus and linear algebra
  - Derivatives,
  - Matrix operations
- Probability
- Machine learning
- Programming

The self-assessment quiz on programming background will be released later this week!



FALL 2025 SELF-ASSESSMENT QUIZ COMP441/541

COMP441/541 Deep Learning, Fall 2025

SELF-ASSESSMENT QUIZ (THEORY)

Due Date: 23:59 Wednesday, October 8, 2024

Each student enrolled to COMP441/541 must complete this quiz on prerequisite math knowledge. The purpose is to self-assess whether you have the right background for the course. The topics covered in this problem set are very crucial so if you are having trouble with solving a problem, this indicates that you should spend a considerable amount of time to study that topic in its entirety.

Points and Vectors

1. Given two vectors $x = [a_1, a_2, a_3]$ and $y = [a_1, -a_2, a_3]$. Write down the equation for calculating the angle between x and y. When is x orthogonal to y?

Planes

2. Consider a hyperplane described by the d-dimensional normal vector $[\theta_1, \ldots, \theta_d]$ and offset θ_0. Derive the equation for the *signed distance* of a point x from the hyperplane, which is defined as the perpendicular distance between x and the hyperplane, multiplied by +1 if x lies on the same side of the plane as the vector θ points and by -1 if x lies on the opposite side x from the hyperplane.

Matrices

3. Suppose that $A^T(AB - C) = 0$, where 0 is an $m \times 1$ vector of zeros, derive an expression for B. Assume that all relevant matrices needed for this calculation are invertible.

matrix $A = \begin{bmatrix} 13 & 5 \\ 2 & 4 \end{bmatrix}$.

$x_1) = \alpha_1 e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}}$

$x_2 \mid X_1 = x_1) = \alpha e^{-\frac{(x_2-x_1)^2}{2\sigma^2}}$

iables. Show that

$= x_2) = \alpha_2 e^{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}}$

by explicitly calculating the values of α_2, μ_2 and σ_2.

MLE and MAP

6. Let p be the probability of landing head of a coin. You flip the coin 3 times and note that it landed 2 times on tails and 1 time on heads. Suppose p can only take two values: 0.3 or 0.6. Find the Maximum Likelihood Estimate of p over the set of possible values {0.3,0.6}

7. Suppose that you have the following prior on the parameter p: $P(p = 0.3) = 0.3$ and $P(p = 0.6) = 0.7$. Given that you flipped the coin 3 times with the observations described above, find the MAP estimate of p over the set {0.3, 0.6}, using the prior.

Page 1 of 2

Self-Assessment Quiz (Theory)
Due Date: October 8 (23:59).

Each student enrolled to COMP441/541 must complete and pass this quiz!

Read Chapter 2-4
of the Deep Learning textbook for a quick review.

Topics Covered in ENGR 421

- Basics of Statistical Learning
  - Loss function, MLE, MAP, Bayesian estimation, bias-variance tradeoff, overfitting, regularization, cross-validation
- Supervised Learning
  - Nearest Neighbor, Naïve Bayes, Logistic Regression, Support Vector Machines, Kernels, Neural Networks, Decision Trees
  - Ensemble Methods: Bagging, Boosting, Random Forests
- Unsupervised Learning
  - Clustering: K-Means, Gaussian mixture models
  - Dimensionality reduction: PCA, SVD

Grading

Self-Assessment Quiz	2%
Programming Assignments	20% (4 assignments x 5% each)
Midterm Exam	17%
Course Project	36%
Paper Presentations	10%
Paper Reviews	5%
Class Participation	10%

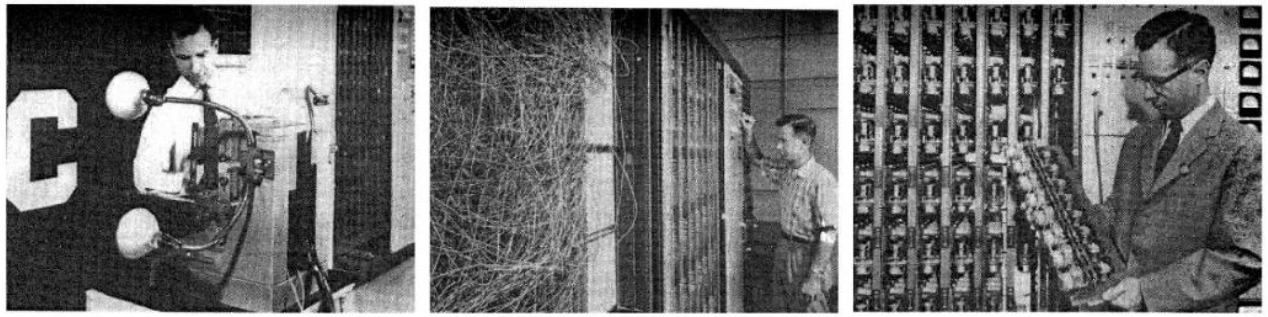
Schedule

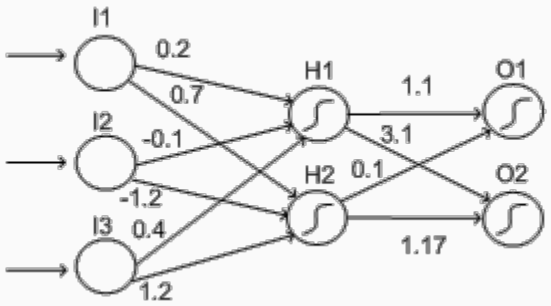
Week 1	Introduction to Deep Learning
Week 2	Machine Learning Overview
Week 3	Multi-Layer Perceptrons
Week 4	Training Deep Neural Networks
Week 5	Convolutional Neural Networks
Week 6	Understanding and Visualizing CNNs
Week 7	Recurrent Neural Networks

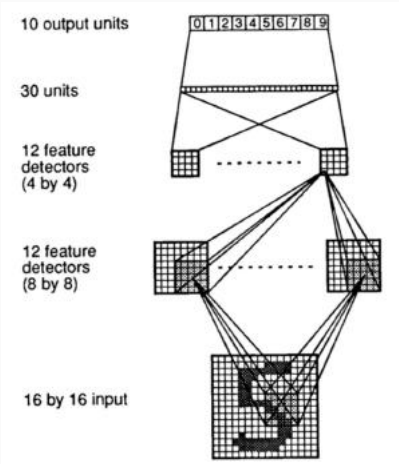
Schedule

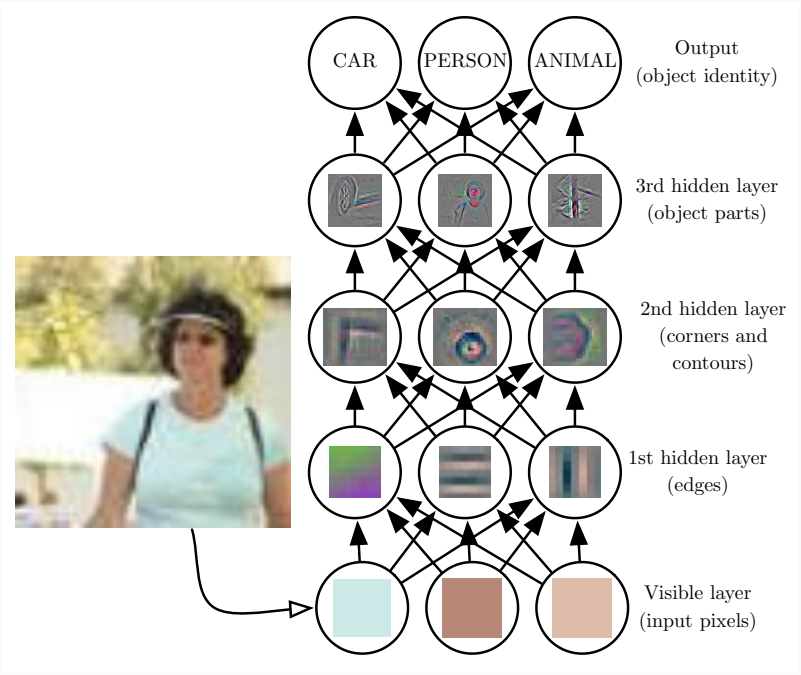
Week 8	Attention and Transformers
Week 9	Graph Neural Networks
Week 10	Language Model Pretraining
Week 11	Project Progress Presentations
Week 12	Large Language Models
Week 13	Adapting LLMs
Week 14	Multimodal Pretraining

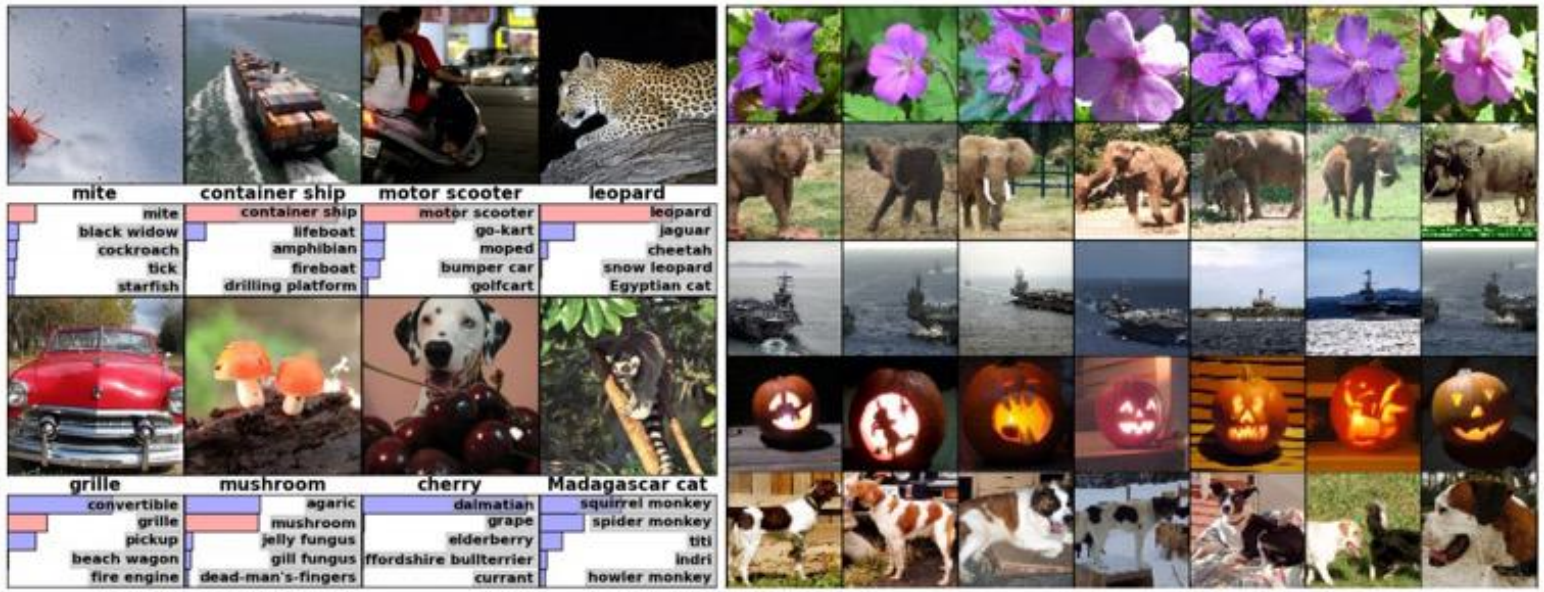
Lecture 1: Introduction to Deep Learning



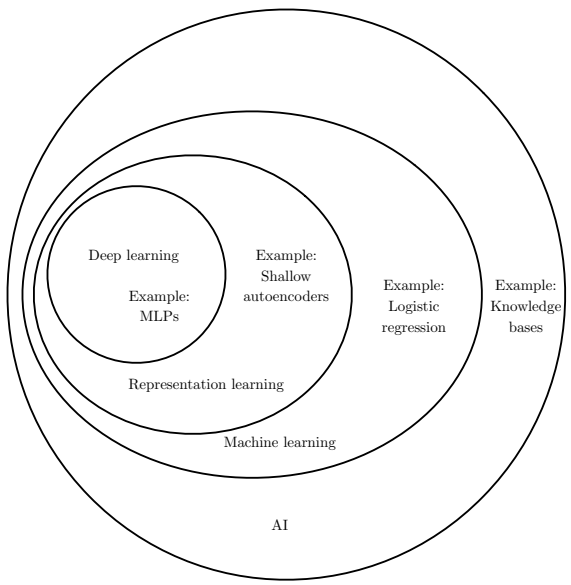


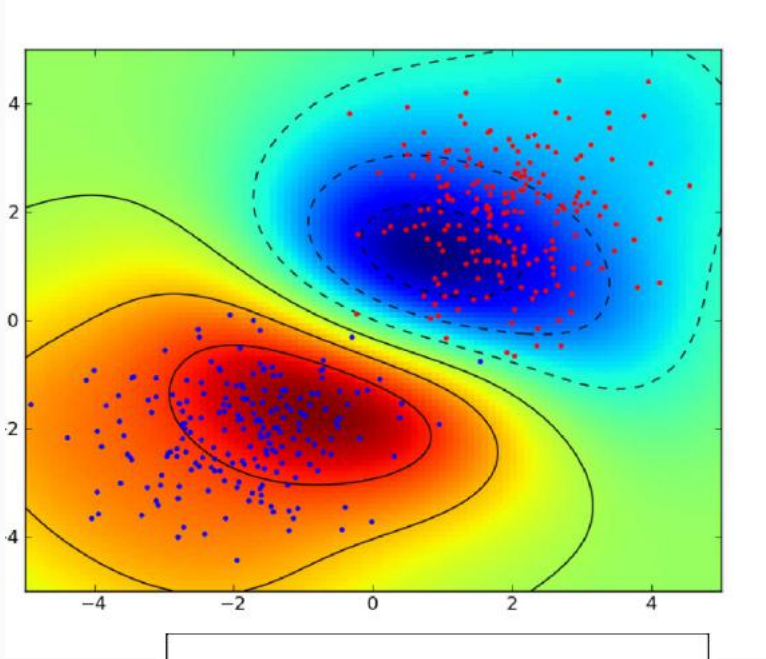


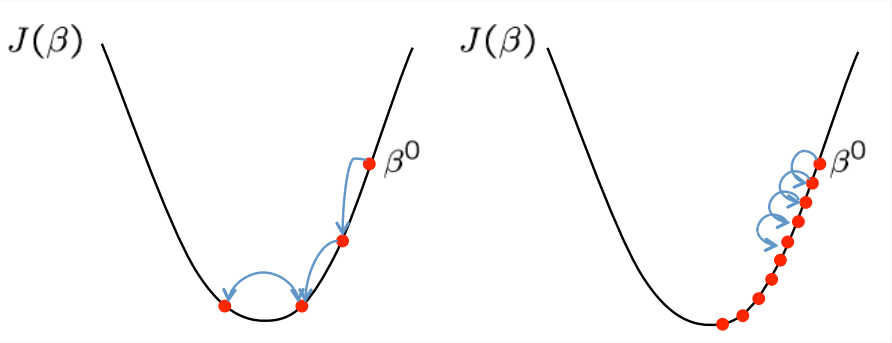


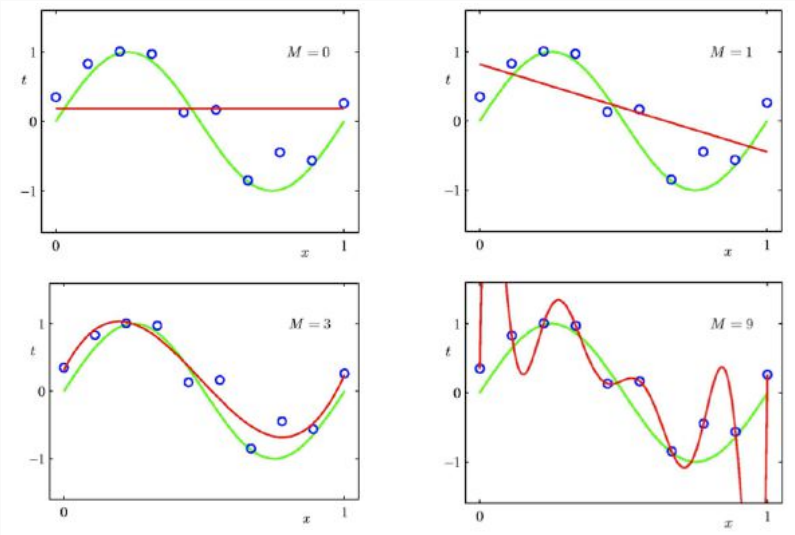


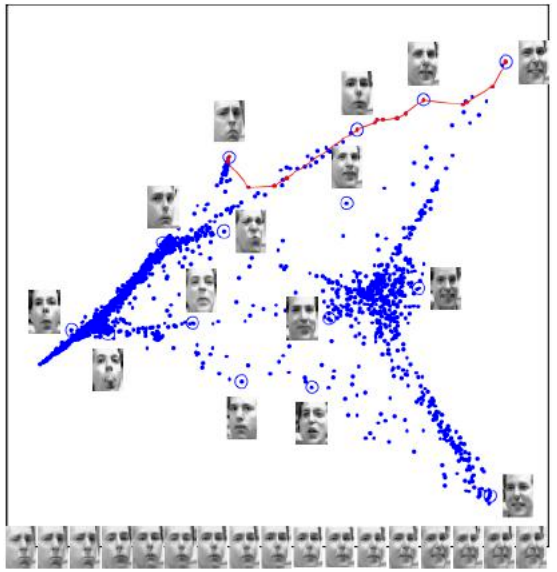
Lecture 2: Machine Learning Overview

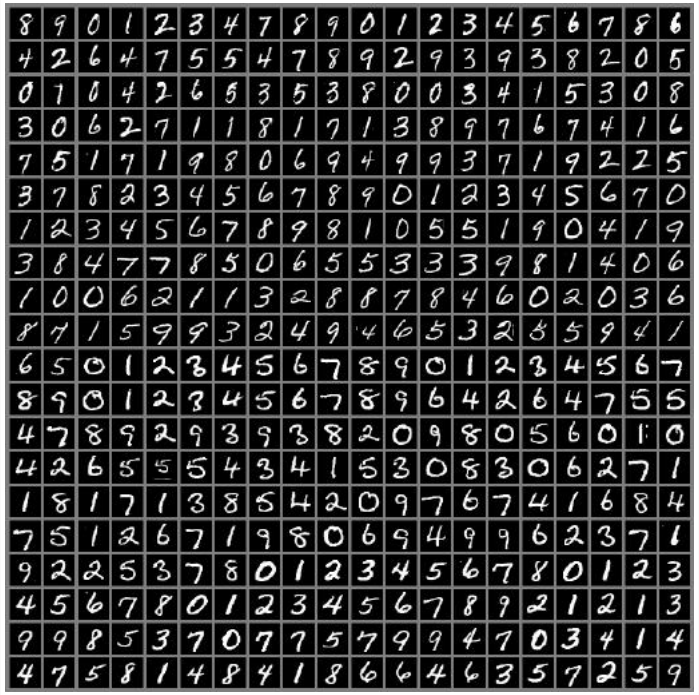




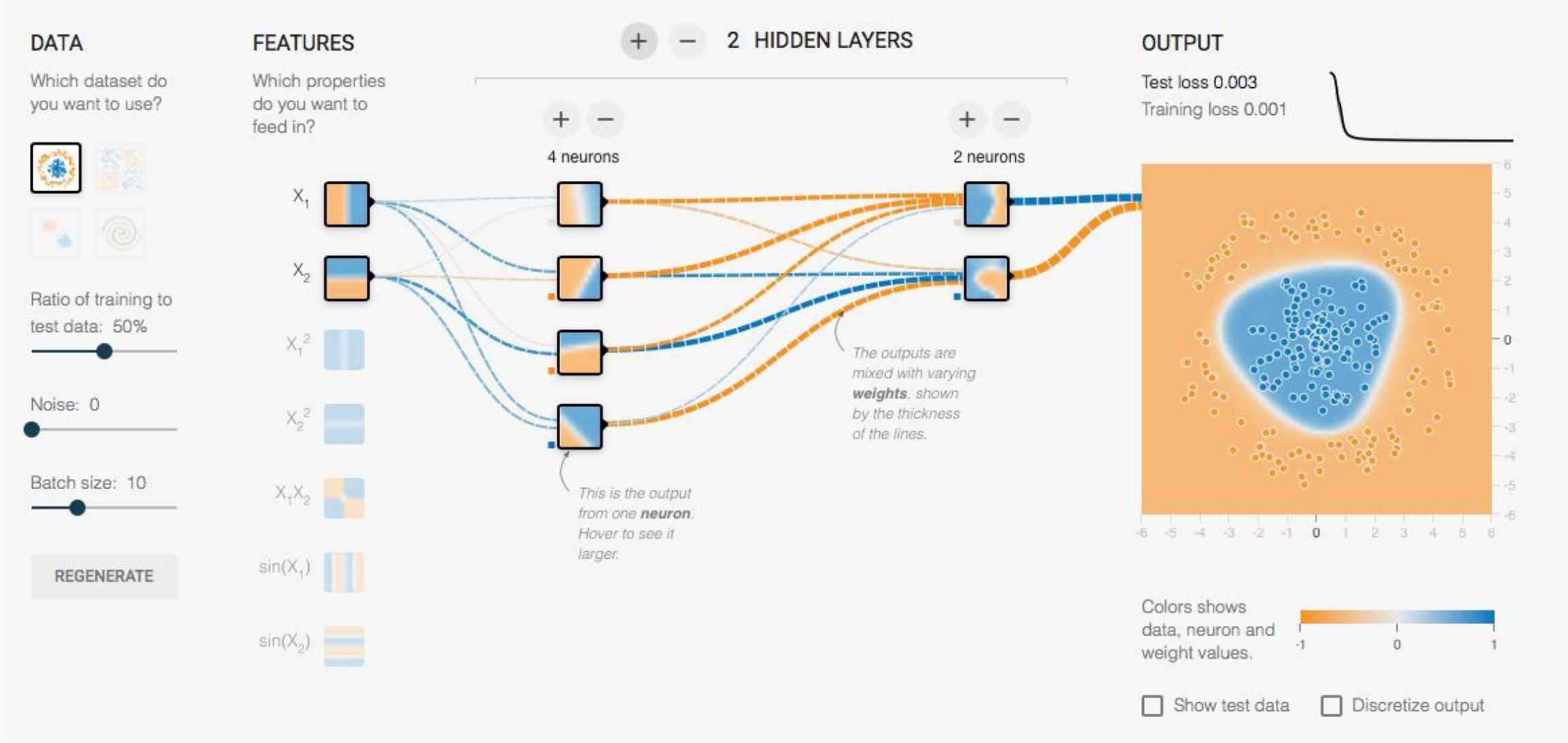






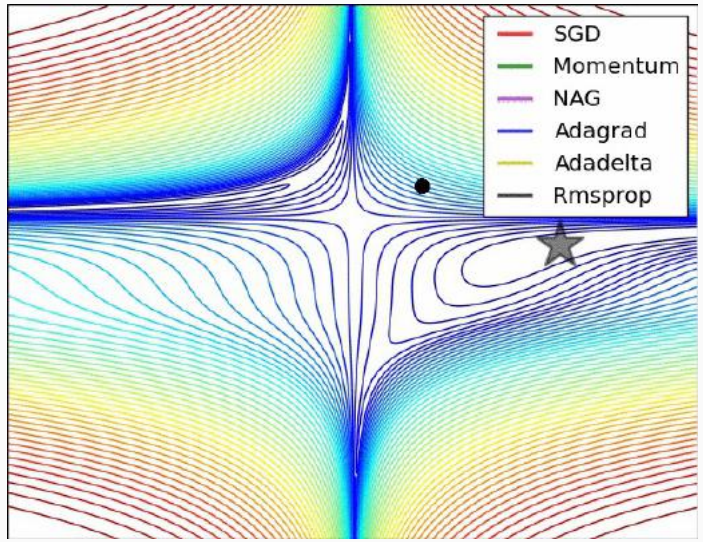


Lecture 3: Multi-Layer Perceptrons



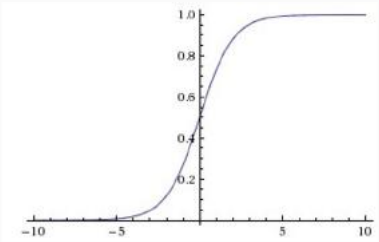
http://playground.tensorflow.org

Lecture 4: Training Deep Neural Networks



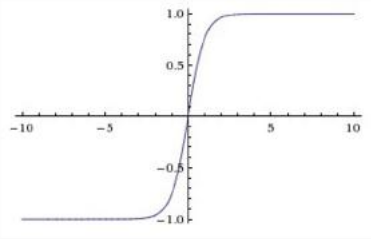
Optimizers

Sigmoid



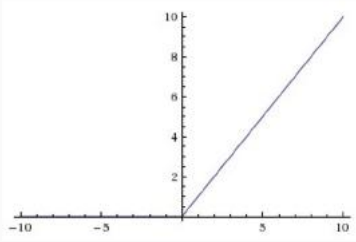
$\sigma(x) = 1/(1+e^{-x})$

tanh



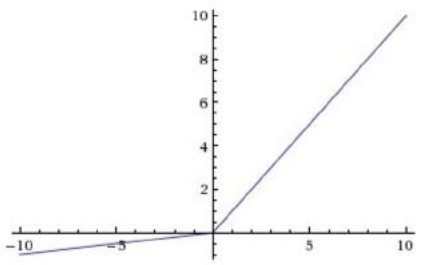
tanh(x)

ReLU



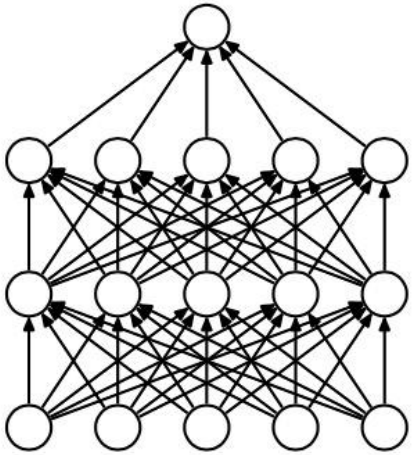
max(0,x)

Leaky ReLU

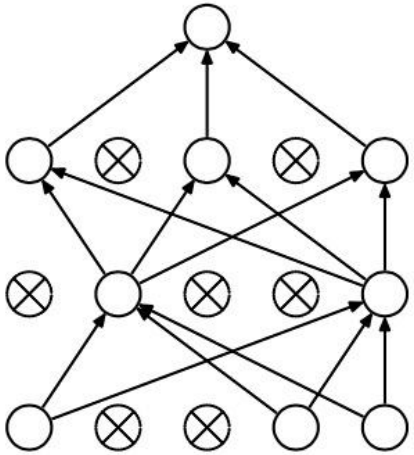


max(0.1x, x)

Activation Functions



(a) Standard Neural Net



(b) After applying dropout.

Dropout

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

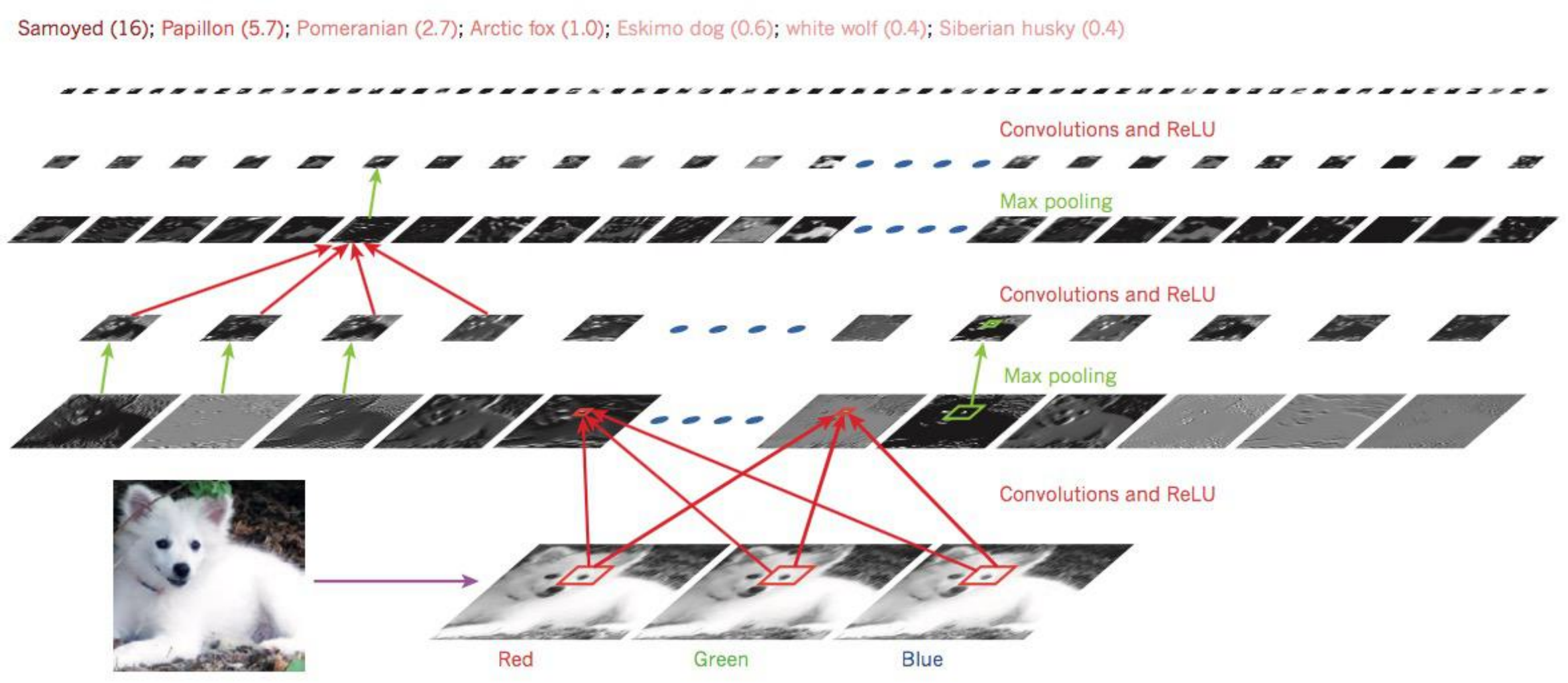
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

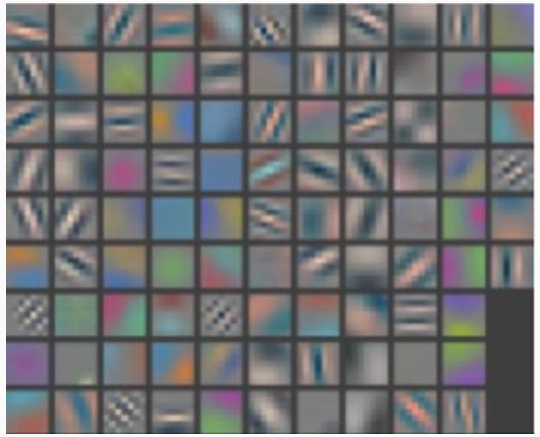
Batch Normalization

Lecture 5: Convolutional Neural Networks

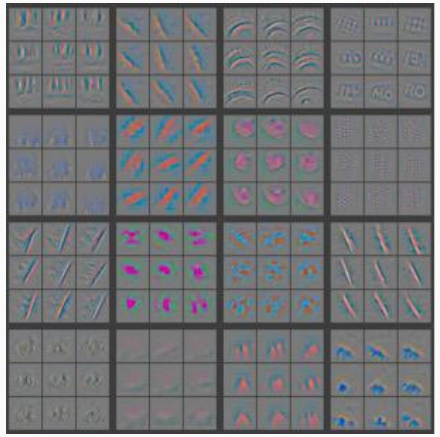


Y. LeCun, Y. Bengio, G. Hinton, **"Deep Learning"**, Nature, Vol. 521, 28 May 2015

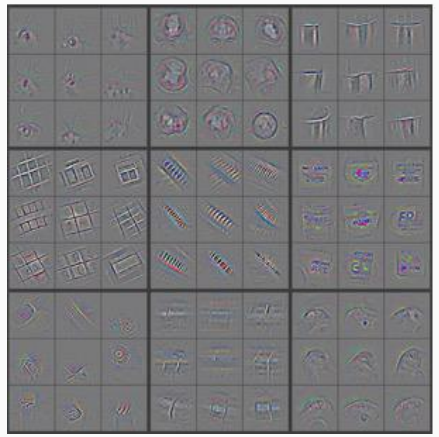
Lecture 6: Understanding and Visualizing CNNs



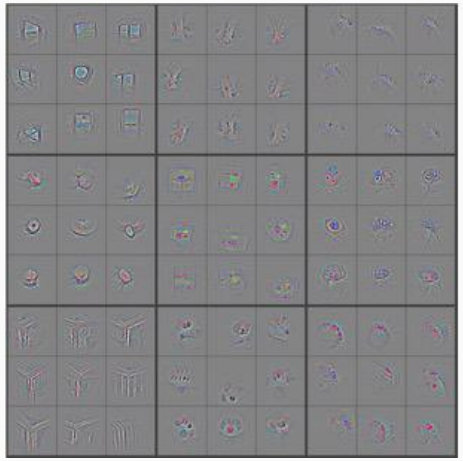
Layer 1

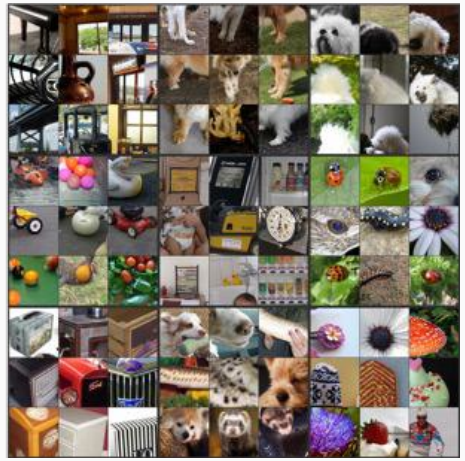


Layer 2

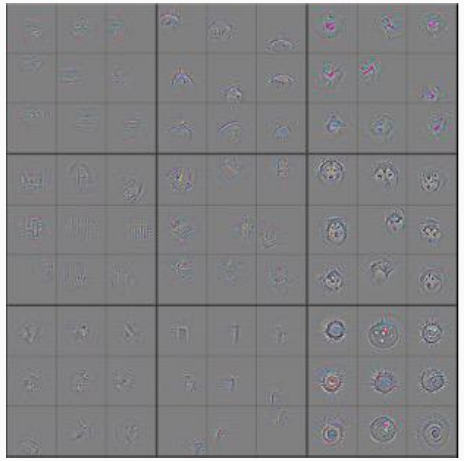


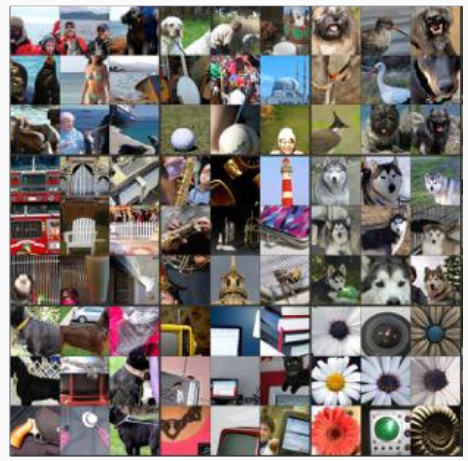
Layer 3





Layer 4

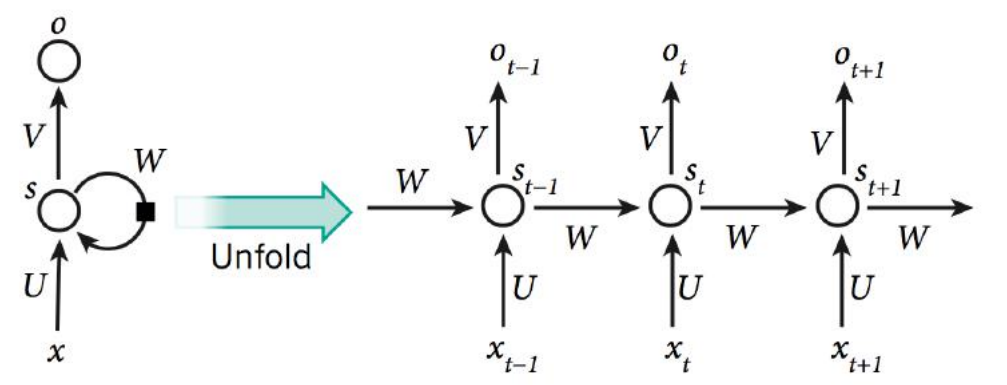




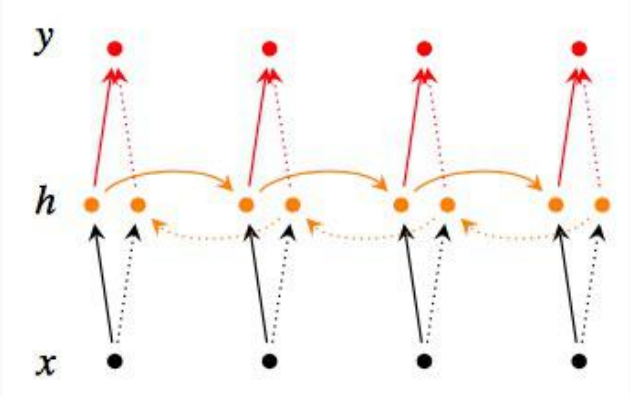
Layer 5

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

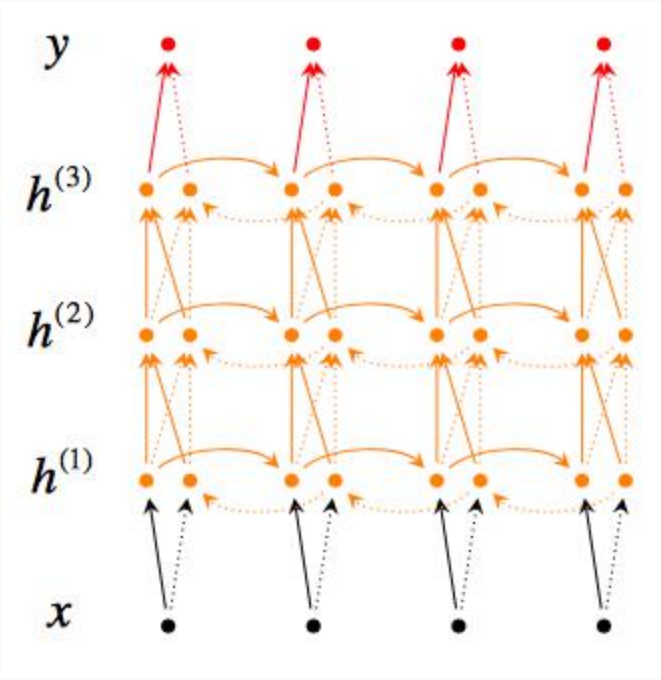
Lecture 7: Recurrent Neural Networks



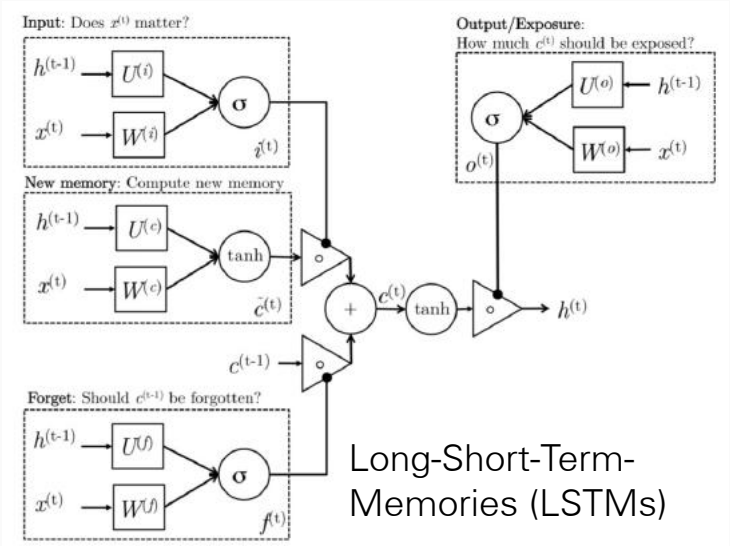
A Recurrent Neural Network (RNN)
(unfolded across time-steps)



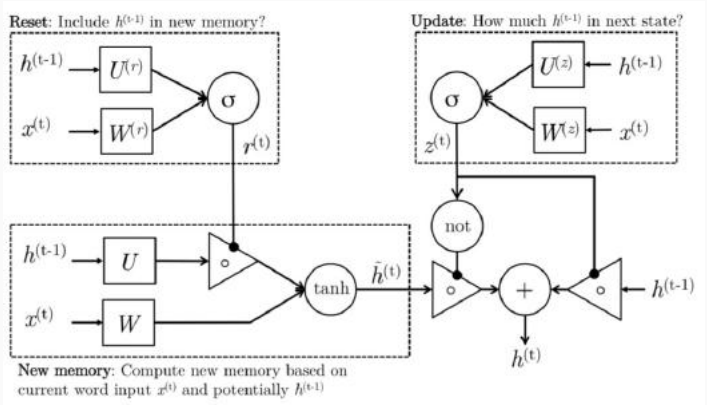
A bi-directional RNN



A deep bi-directional RNN



Long-Short-Term-
Memories (LSTMs)



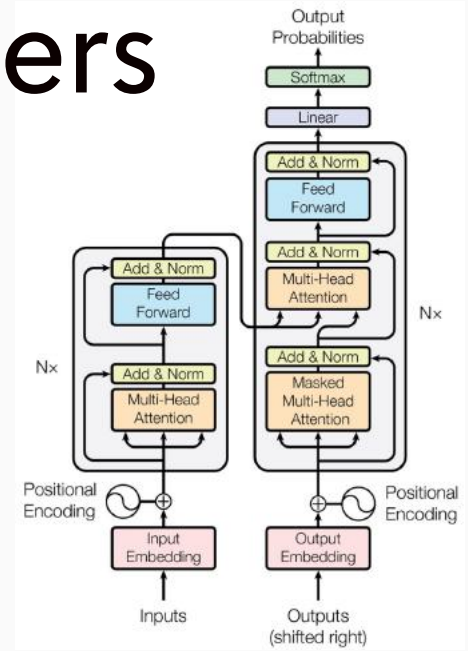
Gated Recurrent Units (GRUs)

C. Manning and R Socher, **Stanford CS224n** Lecture 8 Notes
Y. LeCun, Y. Bengio, G. Hinton, "**Deep Learning**", Nature, Vol. 521, 28 May 2015

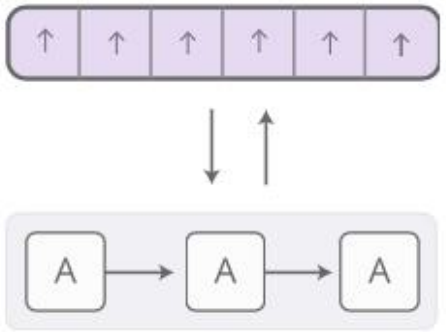
Lecture 8: Attention and Transformers

A little girl sitting on a bed with a teddy bear.

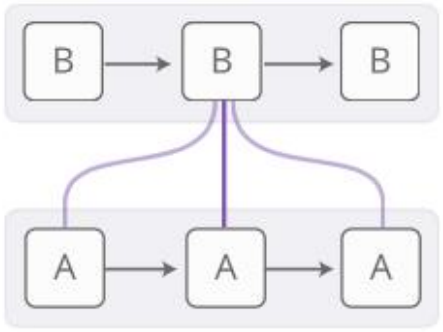
A group of people sitting on a boat in the water.



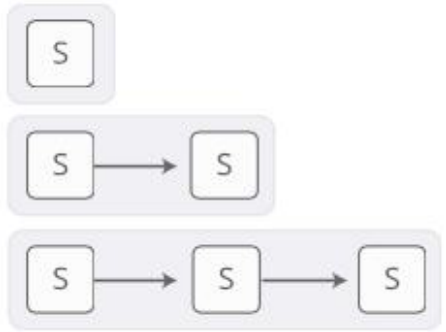
Transformer Architecture



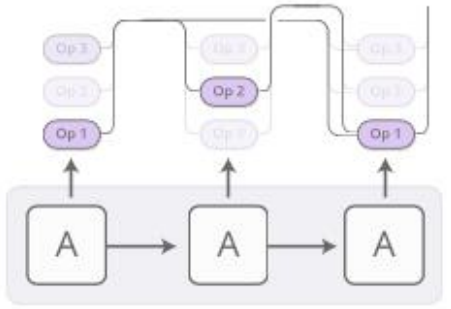
Neural Turing Machines



Attentional Interfaces



Adaptive Computation Time

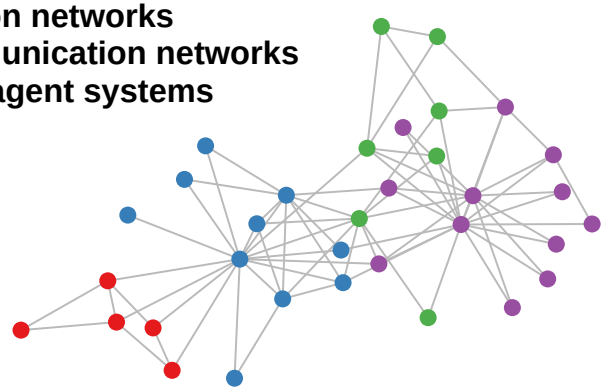


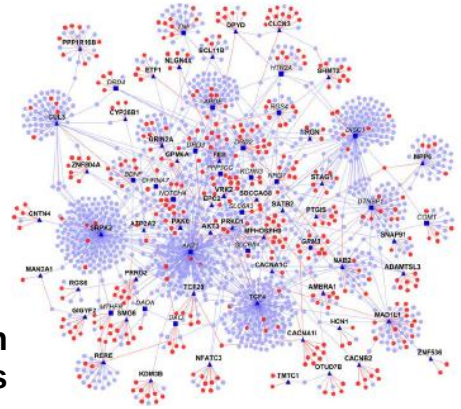
Neural Programmers

K. Xu et al., "**Show, Attend and Tell: Neural Image Caption Generation with Visual Attention**", ICML 2015
C. Olah and S. Carter, "**Attention and Augmented Recurrent Neural Networks**", Distill, 2016
A. Vaswani et al. "**Attention is All You Need**", NeurIPS 2017.

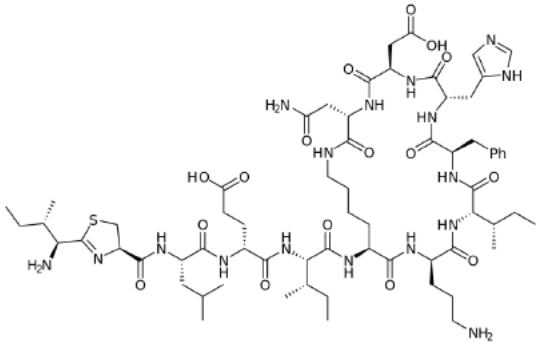
Lecture 9: Graph Networks

Social networks
Citation networks
Communication networks
Multi-agent systems

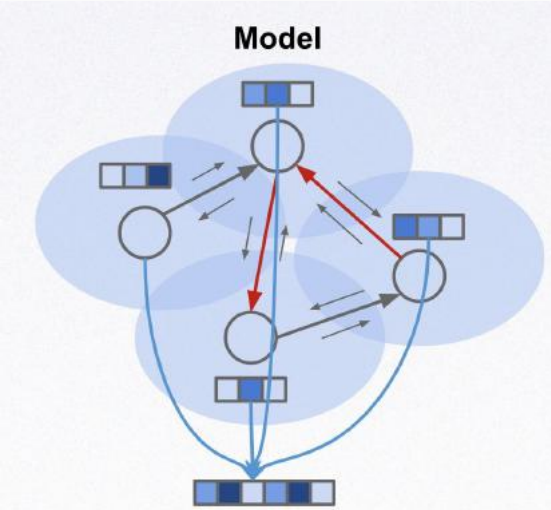


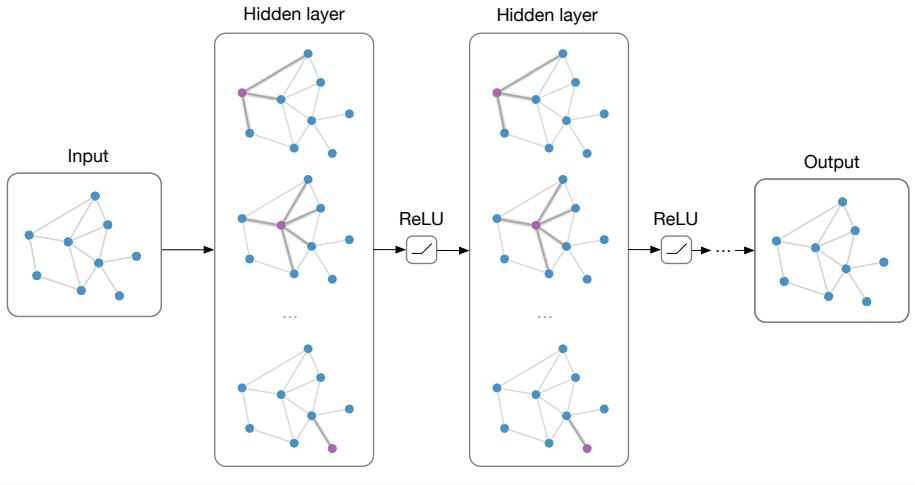


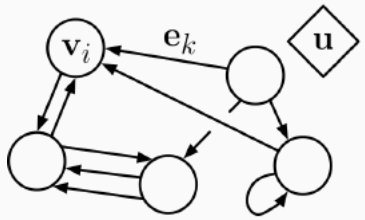
Protein interaction networks

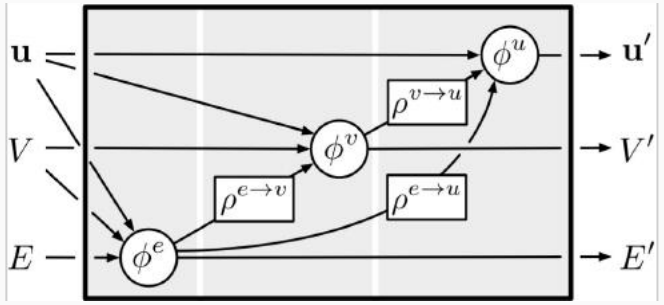


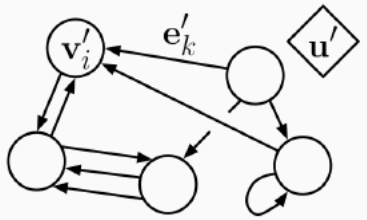
Molecules





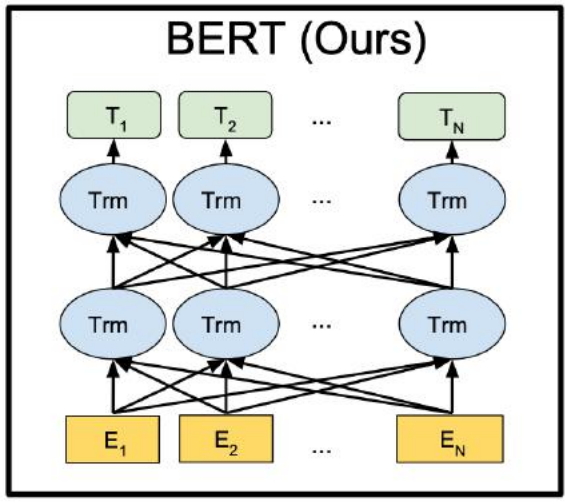


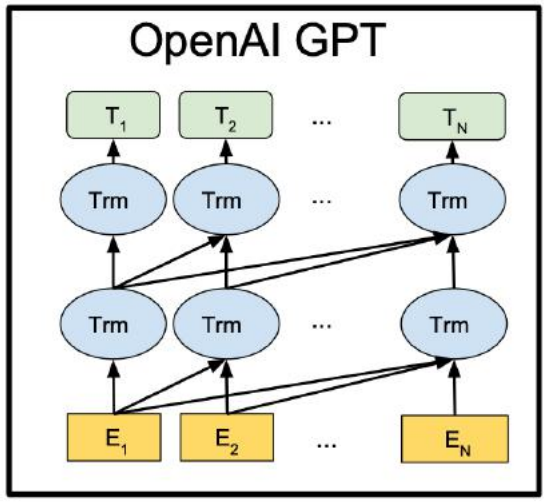


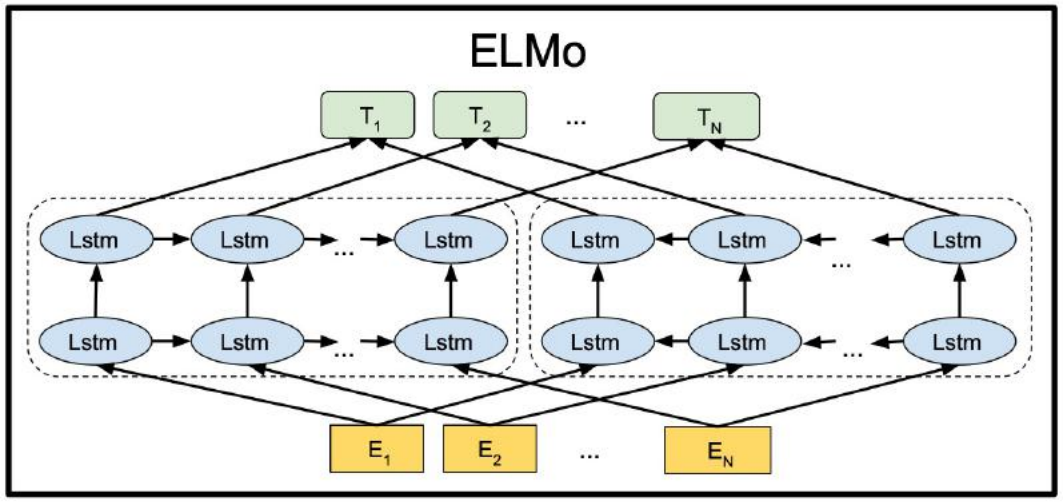


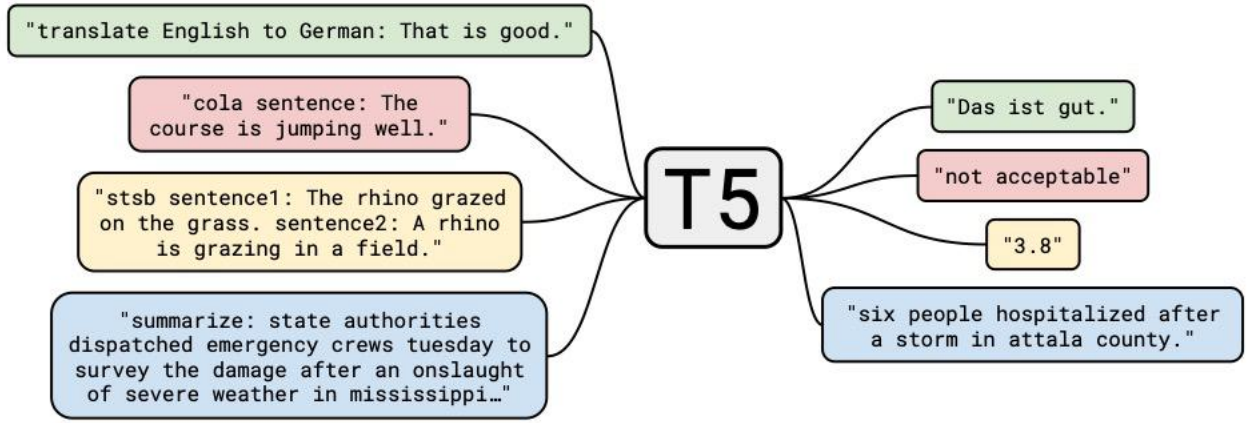
T.N. Kipf and M. Welling, "**Semi-supervised classification with graph convolutional networks**", ICLR 2017
P. Battaglia et al., "**Relational inductive biases, deep learning, and graph networks**", arXiv 2018

Week 10: Pretraining Language Models



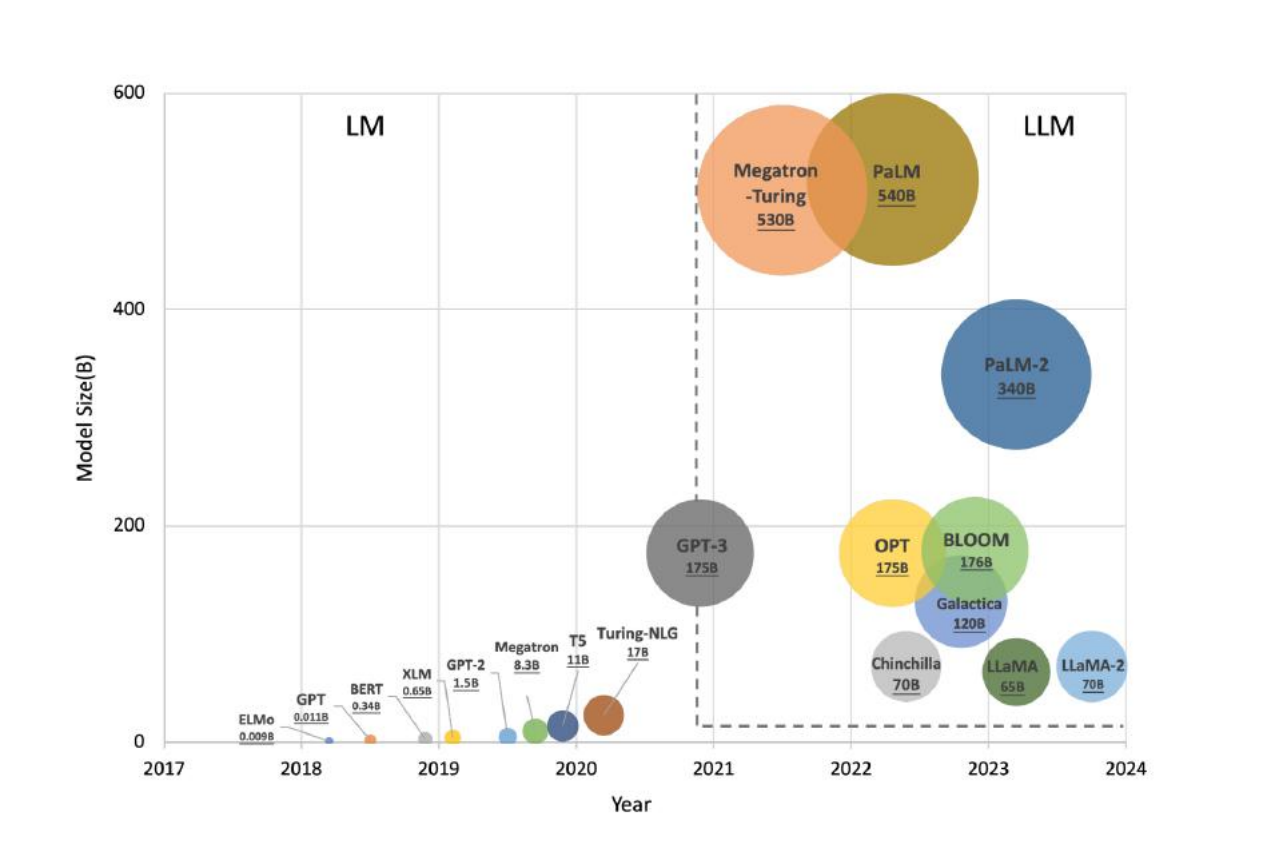






J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**", NAACL-HLT 2019.
C. Raffel et al., "**Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer**", JMLR 2020.

Lecture 11: Large Language Models



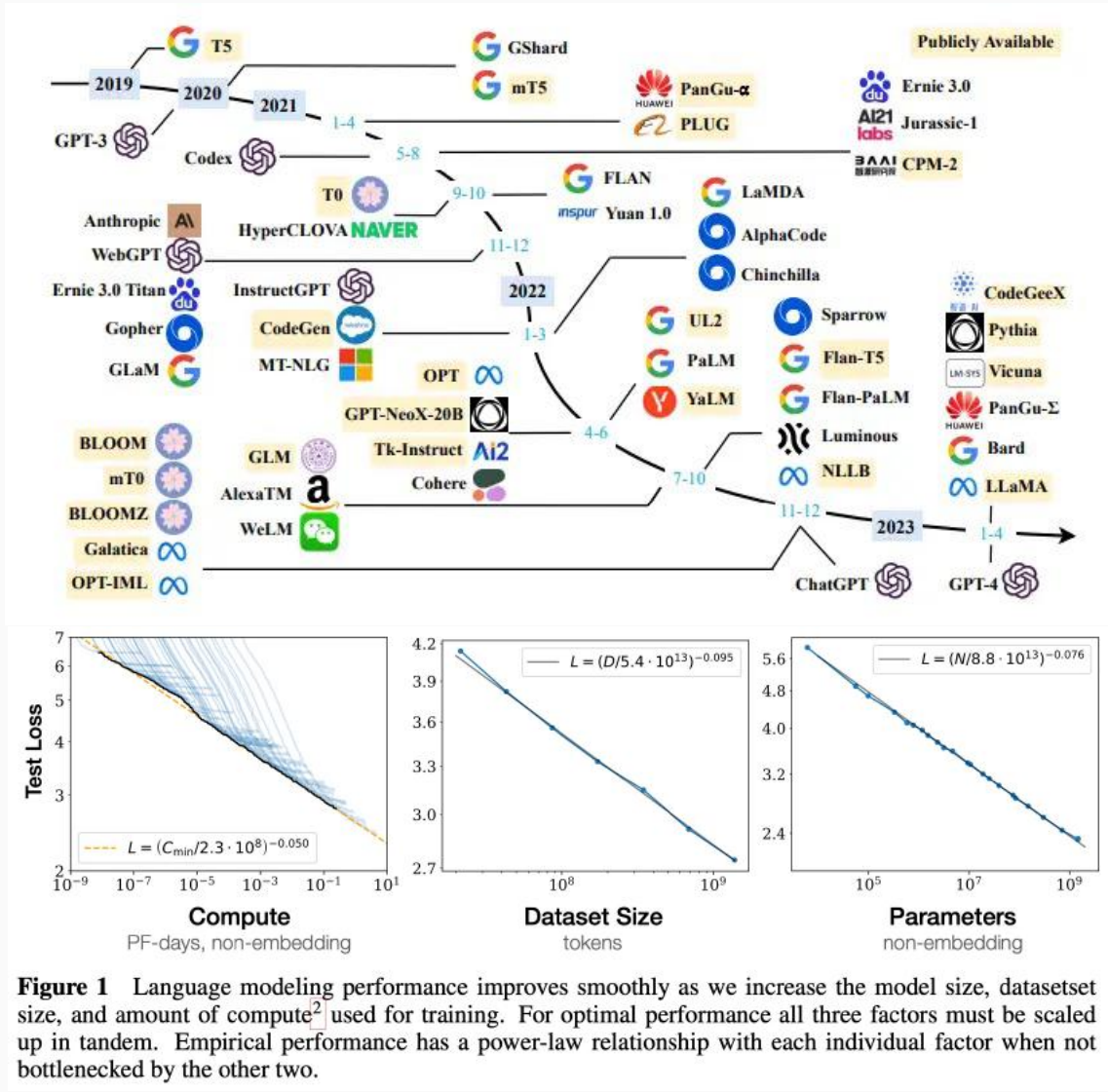
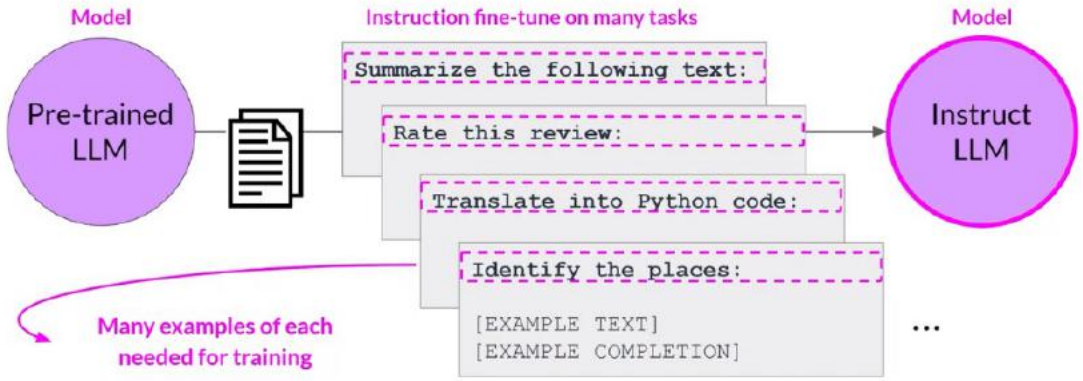
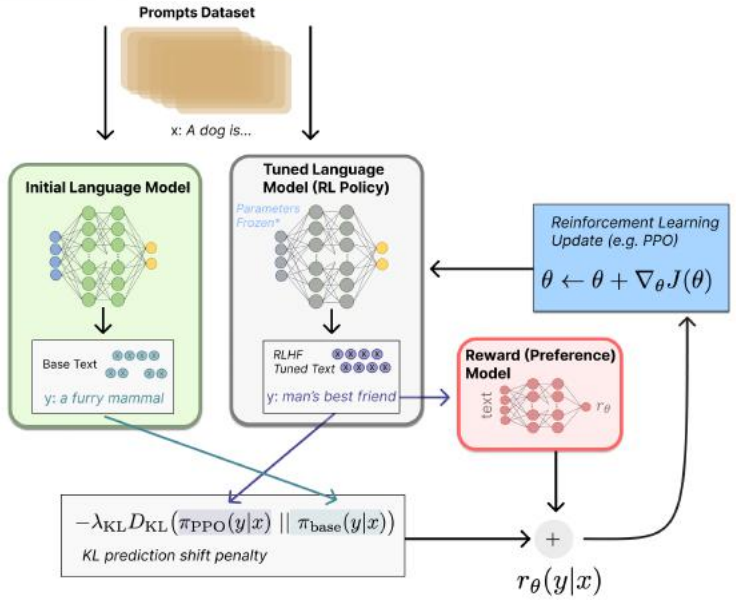


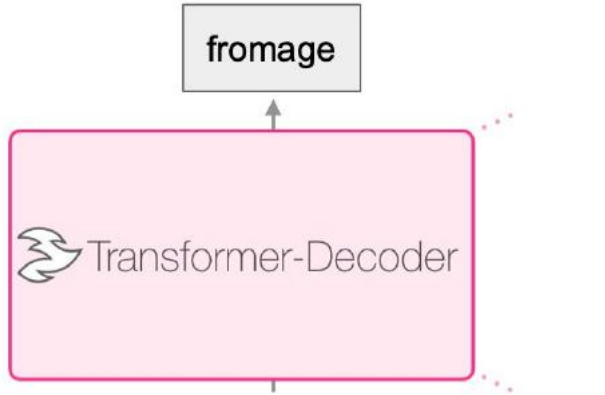
Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, Dario Amodei, **Scaling Laws for Neural Language Models**", arXiv preprint, 2020.

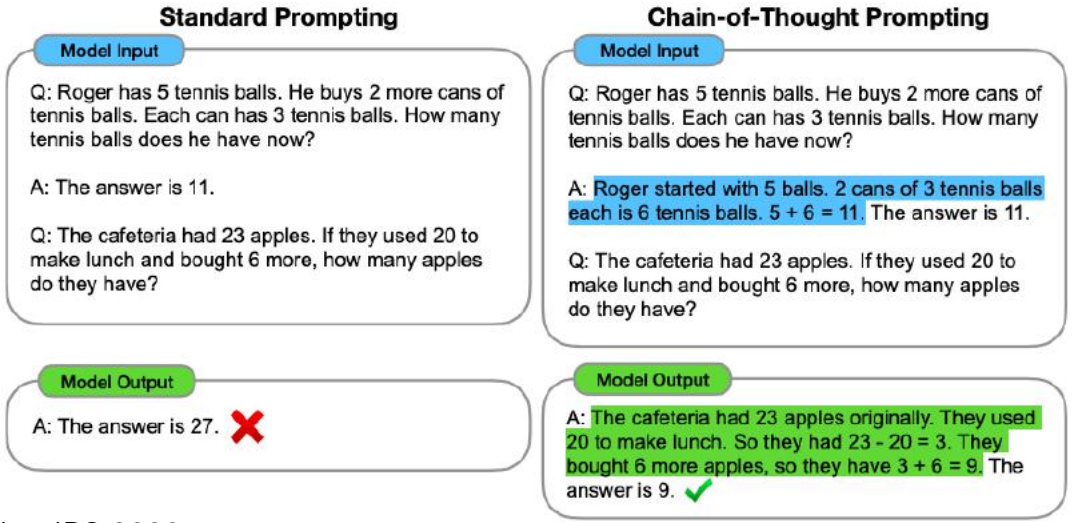
Lecture 12: Adapting LLMs







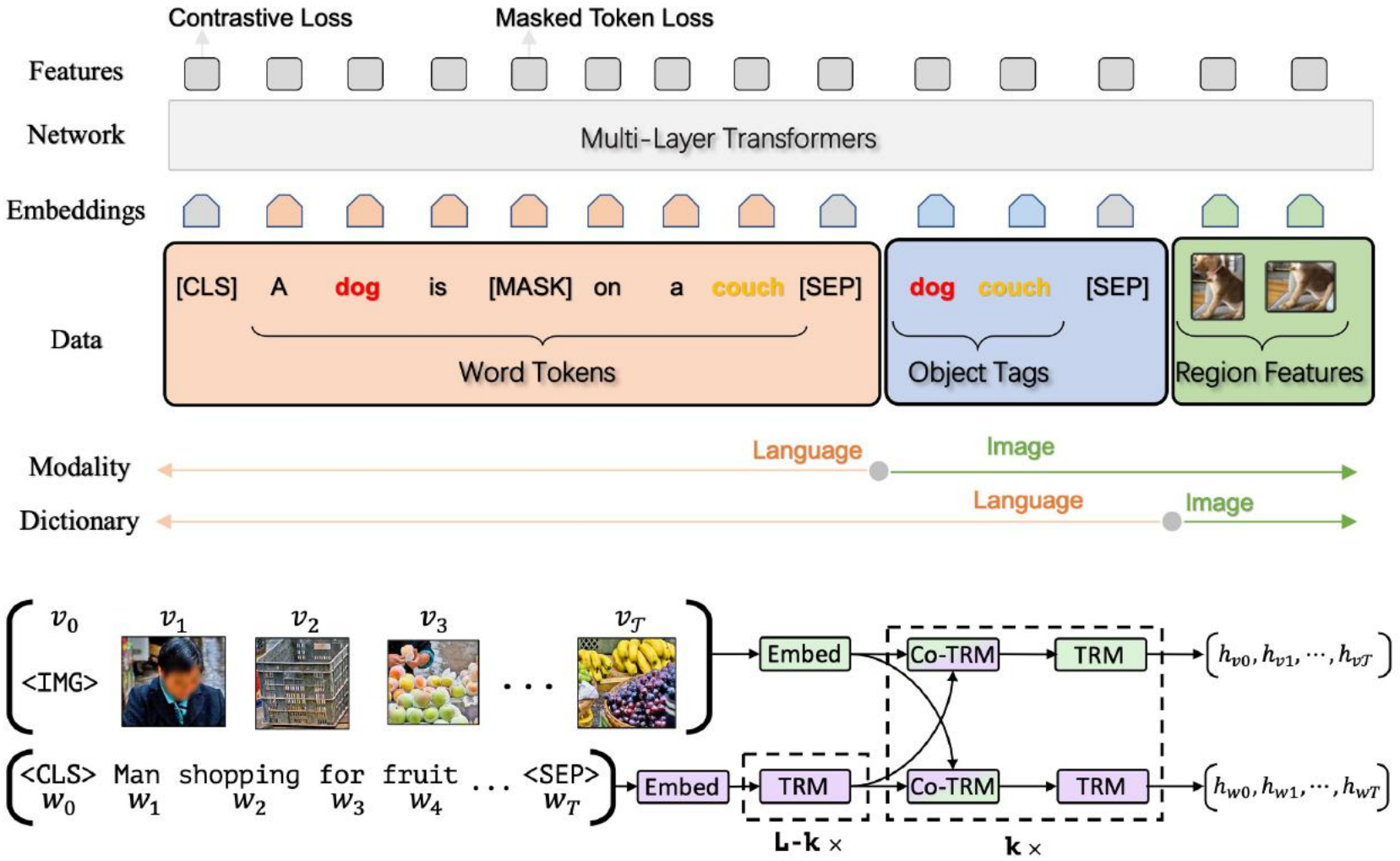

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                           ← prompt
```



Tom B. Brown, Benjamin Mann, Nick Ryder, et al., **Language Models are Few-Shot Learners**, NeurIPS 2020.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, Geoffrey Irving, **Fine-Tuning Language Models from Human Preferences**, Open AI Technical Report, 2020

Week 13: Multimodal Pre-training



J. Lu, D. Batra, D. Parikh, S, Lee, "**ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks**", NeurIPS 2019

X. Li et al., "**Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks**", ECCV 2020.

Schedule

L1 Introduction to Deep Learning
Self-Assessment Quiz (Theory)

L2 Machine Learning Overview
Self-Assessment Quiz (Programming)

L3 Multi-Layer Perceptrons
Assignment 1 out

L4 Training Deep Neural Networks

L5 Convolutional Neural Networks
Start of paper presentations
Assignment 1 in, Assignment 2 out

L6 Understanding and Visualizing CNNs
Project proposals due

L7 Recurrent Neural Networks
Assignment 2 in, Assignment 3 out

L8 Attention and Transformerns
Midterm Exam

L9 Graph Neural Networks
Assignment 3 in, Assignment 4 out

L10 Language Model Pretraining

L11 Project Progress Presentations
Project progress reports due

L12 Large Language Models (LLMs)
Assignment 4 in

L13 Adapting LLMs

L14 Multimodal Pretaining

Final project reports due

Paper Presentations

We will discuss 10 recent papers related to the topics covered in the class.

See the presenter roles on the course web page for the details.

Week	Topic
Week 1	Introduction to Deep Learning
Week 2	Machine Learning Overview
Week 3	Multi-Layer Perceptrons
Week 4	Training Deep Neural Networks
Week 5	Convolutional Neural Networks
Week 6	Understanding and Visualizing CNNs
Week 7	Recurrent Neural Networks
Week 8	Attention and Transformers
Week 9	Graph Neural Networks
Week 10	Language Model Pretraining
Week 11	Project Progress Presentations
Week 12	Large Language Models
Week 13	Efficient LLMs
Week 14	Multimodal Pre-training
Week 15-16	Final Project Presentations

Paper presentations start on Week 5

Paper Reviews

Think deeply about the papers we read and try to learn from them as much as possible (and then even more). If you do not understand something, we should discuss it and dissect it together. Whatever you think others understand, they understand less (the instructor included), but together we will get it.

- Identify the key questions the paper studies, and the answers it provides to these questions.
- Consider the challenges of the problem or scenario studied, and how the paper's approach addresses them.
- Deconstruct the formal and technical parts to understand their fine details. Note to yourself aspects that are not clear to you

Borrowed from Yoav Artzi

Paper Reviewing Guidelines

- When reviewing the paper, start with 1–2 sentences summarizing what the paper is about.
- Continue with the strength of the paper. Outline its contribution, and your main takeaways. What did you learn?
- Highlight shortcomings and limitations. Please focus on weaknesses that fundamental to the method. Unlike conference or journal reviewing, this part is intended for your understanding and discussion.
- Try to suggest ways to address the paper's limitations. Any idea is welcome and will contribute to the discussion.
- Suggest questions for discussion in class. As part of the discussion in class, you are asked to raise these questions during the class.

Borrowed from Yoav Artzi

Programming Assignments

- 4 programming assignments (5% each)
- Learning to implement basic neural architectures
- Should be done individually

- **Late policy:** You have 7 grace days in the semester.

- **Assignments**
  - Assignment 1: MLPs and Backpropagation
  - Assignment 2: Convolutional Neural Networks
  - Assignment 3: Recurrent Neural Networks
  - Assignment 4: Transformers and GNNs

Midterm Exam

- **Date:** Week 8
- **Topics:** Everything covered in the first part of the course

- Format to be a classical exam with derivations and short discussion questions.

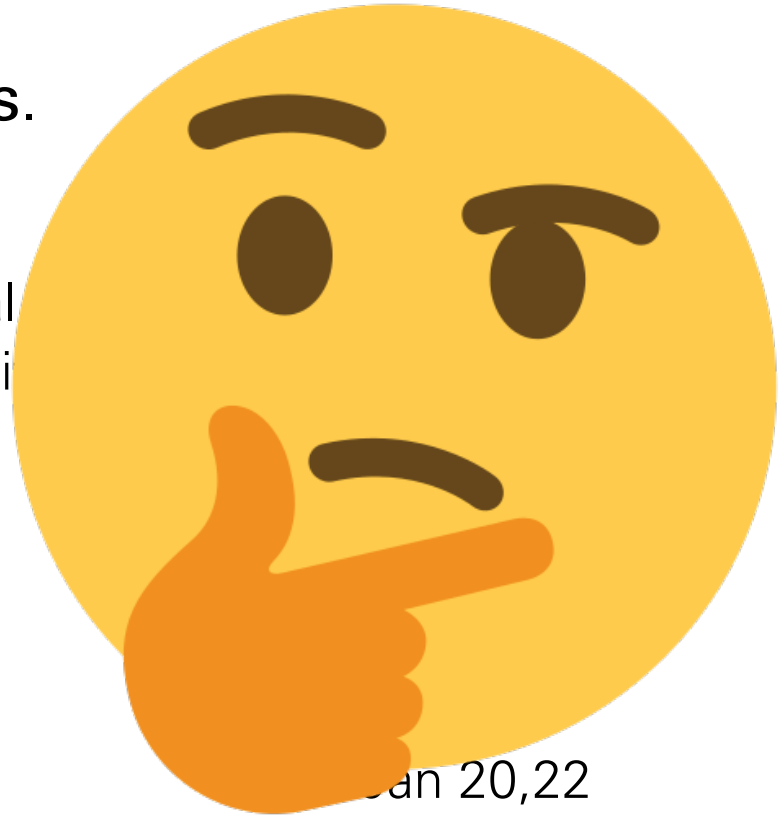
Course Project

- The course project gives students a chance to apply deep learning models discussed in class to a research-oriented project
- Projects should be done **in groups of 2 to 3 students**.
- The course project may involve
  - Design of a novel approach/architecture and its experimental analysis, or
  - An extension to a recent study of non-trivial complexity and its experimental analysis.
- **Deliverables**
  - Proposals (2%) Nov 16
  - Project progress presentations (4%) Dec 16,18
  - Project progress reports (6%) Dec 21
  - Final project presentations (8%) Jan 20,22
  - Final reports (12%) Jan 25
  - The quality of the contributions/The difficulty of implementation (4%)

Course Project

- The course project gives students a chance to apply deep learning models discussed in class to a research-oriented project
- Projects should be done **in groups of 2 to 3 students**.
- The course project may involve
  - Design of a novel approach/architecture and its experimental
  - An extension to a recent study of non-trivial complexity and i
- **Delive**
  - Pro
  - Pro
  - Project progress reports (6%)
  - Final project presentations (8%) Jan 20,22
  - Final reports (12%) Jan 25
  - The quality of the contributions/The difficulty of implementation (4%)

Start thinking about project ideas!

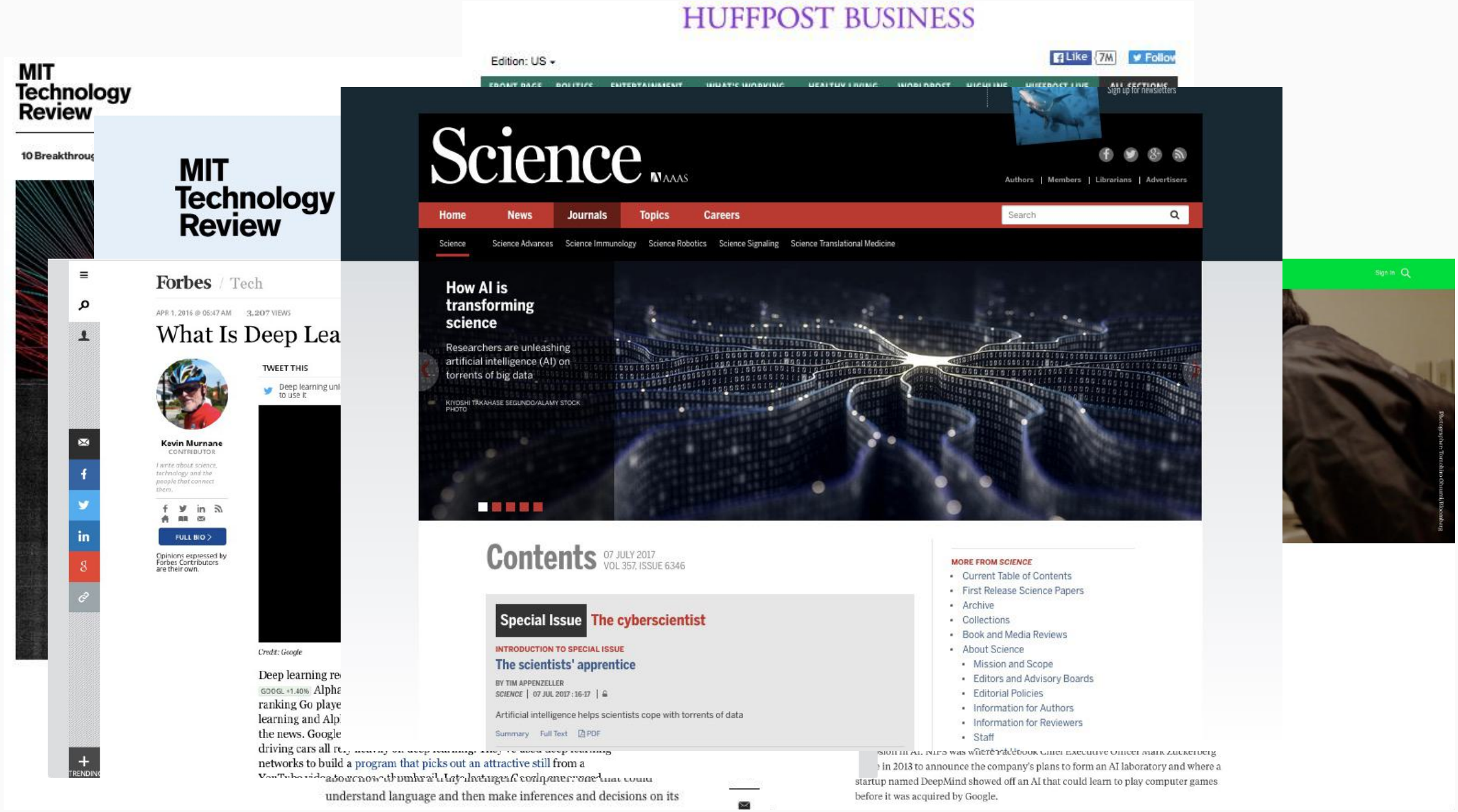


Lecture Overview

- what is deep learning
- a brief history of deep learning
- compositionality
- end-to-end learning
- distributed representations

Disclaimer: Some of the material and slides for this lecture were borrowed from
—Dhruv Batra's CS7643 class
—Yann LeCun's talk titled "Deep Learning and the Future of AI"

What is Deep Learning


HUFFPOST BUSINESS
MIT Technology Review
MIT Technology Review
Science
AAAS
Home
News
Journals
Topics
Careers
How AI is transforming science
Researchers are unleashing artificial intelligence (AI) on torrents of big data
Contents
07 JULY 2017
VOL 357, ISSUE 6346
Special Issue
The cyberscientist
INTRODUCTION TO SPECIAL ISSUE
The scientists' apprentice
BY TIM APPENZELLER
Artificial intelligence helps scientists cope with torrents of data
Forbes / Tech
What Is Deep Lea
Kevin Murnane
CONTRIBUTOR
FULL BIO
TWEET THIS
Credit: Google
networks to build a program that picks out an attractive still from a
understand language and then make inferences and decisions on its
MORE FROM SCIENCE
Current Table of Contents
First Release Science Papers
Archive
Collections
Book and Media Reviews
About Science
Mission and Scope
Editors and Advisory Boards
Editorial Policies
Information for Authors
Information for Reviewers
Staff
in 2013 to announce the company's plans to form an AI laboratory and where a
startup named DeepMind showed off an AI that could learn to play computer games
before it was acquired by Google.

What is deep learning?

REVIEW

Deep learning

Yann LeCun[1], Yoshua Bengio[2] & Geoffrey Hinton[3]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

"Deep learning allows computational models that are composed of **multiple processing layers to learn representations of data with multiple levels of abstraction.**"

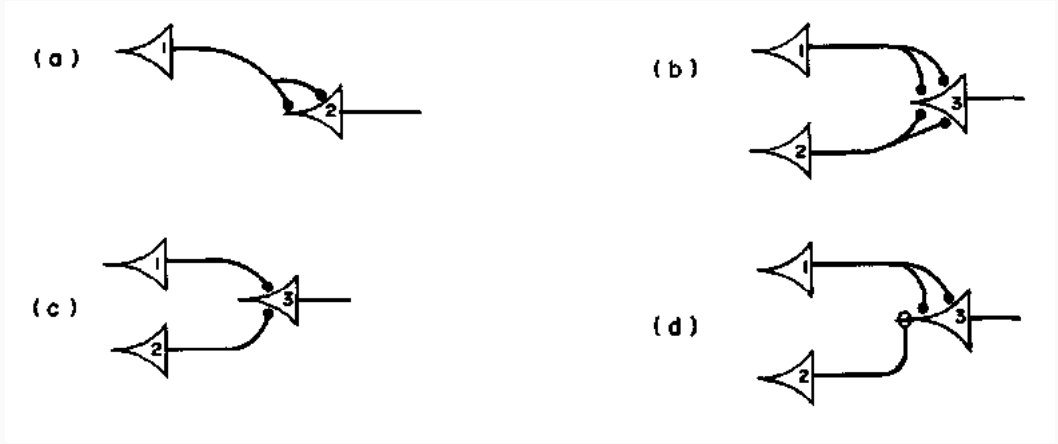
– Yann LeCun, Yoshua Bengio and Geoff Hinton

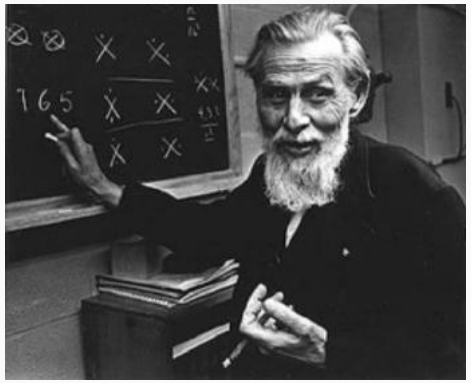
Y. LeCun, Y. Bengio, G. Hinton, **"Deep Learning"**, Nature, Vol. 521, 28 May 2015

1943 – 2006: A Prehistory of Deep Learning

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





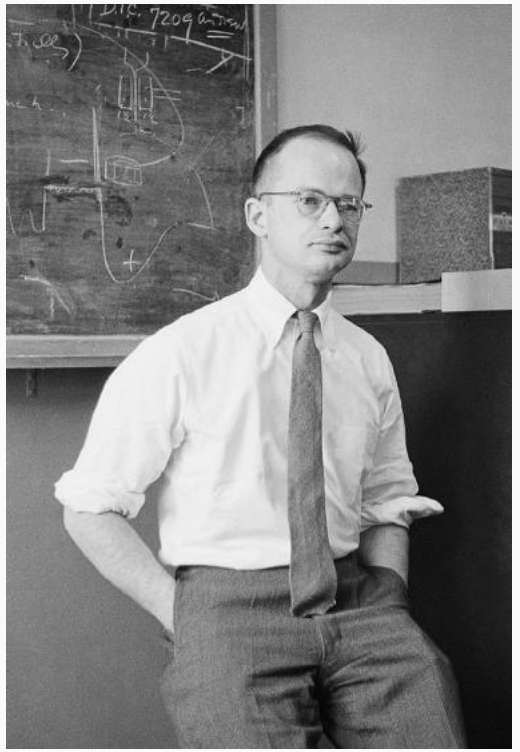
A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

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.

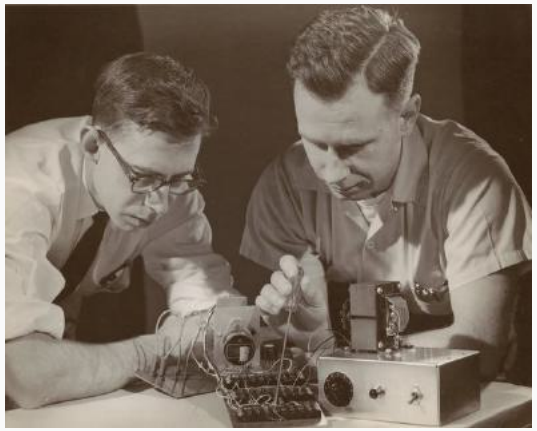
1. Introduction. Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which 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

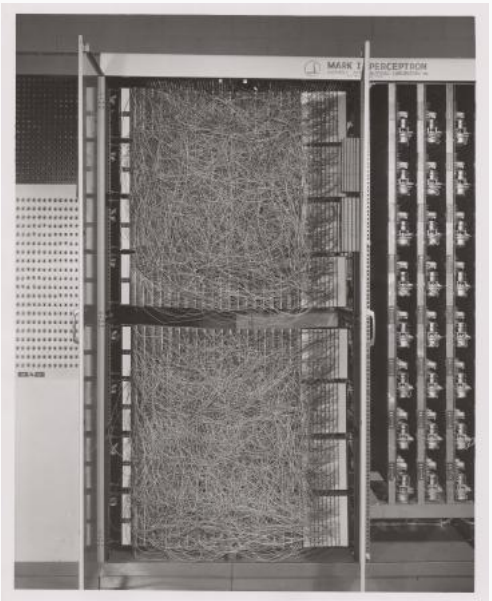
* Reprinted from the *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133 (1943).

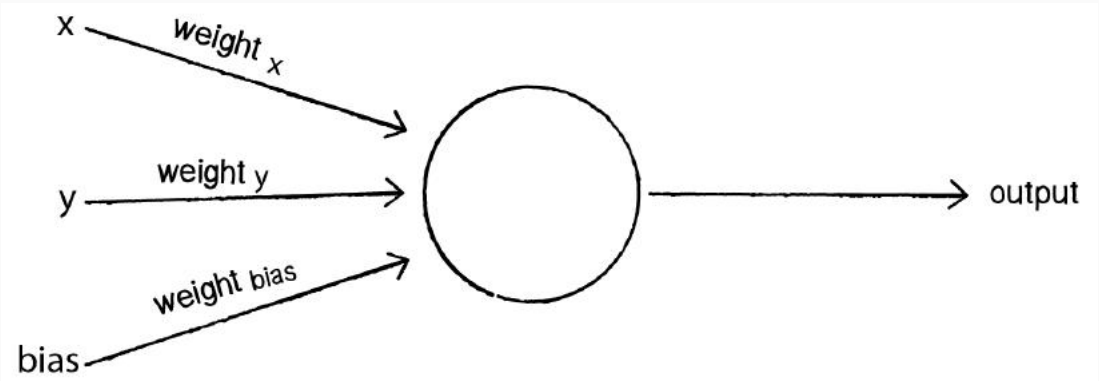


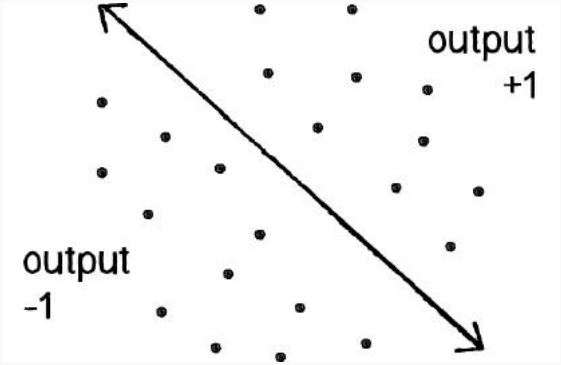
1958: Frank Rosenblatt's Perceptron

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







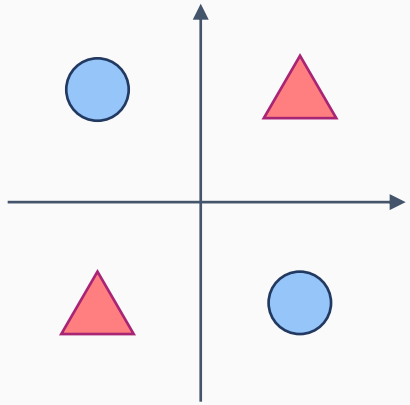


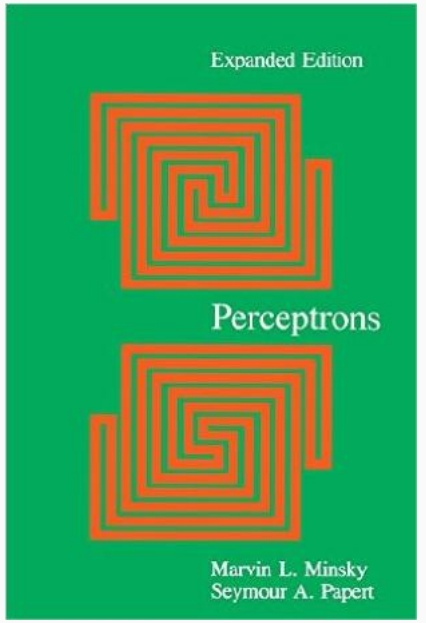
F. Rosenblatt, "**The perceptron: A probabilistic model for information storage and organization in the brain**", Psych. Review, Vol. 65, 1958

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

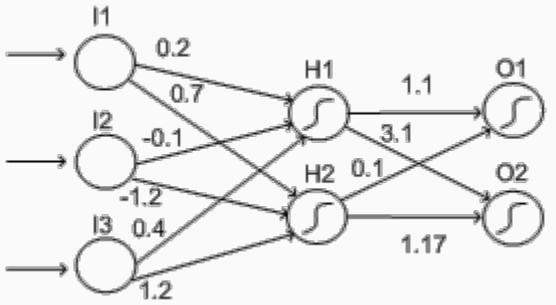




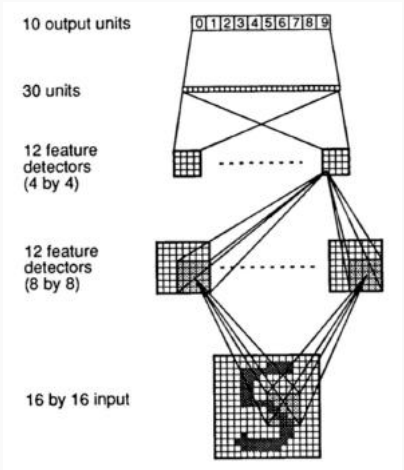
- Wrongly attributed as the reason behind the **AI winter**, a period of reduced funding and interest in AI research

1990s

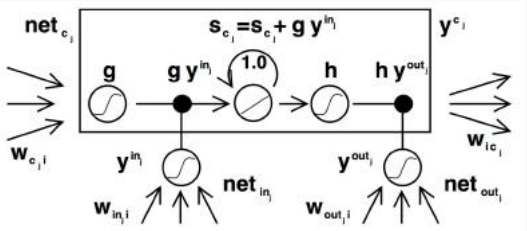
- **Multi-layer perceptrons** can theoretically learn any function (Cybenko, 1989; Hornik, 1991)



- 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)



Why it failed then

- Too many parameters to learn from few labeled examples.
- “I know my features are better for this task”.
- Non-convex optimization? No, thanks.
- Black-box model, no interpretability.

- Very slow and inefficient
- Overshadowed by the success of SVMs (Cortes and Vapnik, 1995)

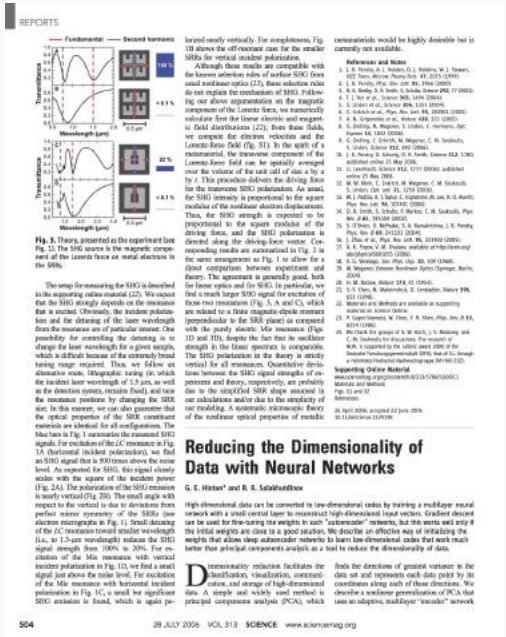
A major breakthrough in 2006

2006 Breakthrough: Hinton and Salakhutdinov

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.


REPORTS

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

- The first solution to the **vanishing gradient problem**.
- Build the model in a layer-by-layer fashion using unsupervised learning
  - The features in early layers are already initialized or "pretrained" with some suitable features (weights).
  - Pretrained features in early layers only need to be adjusted slightly during supervised learning to achieve good results.

G. E. Hinton and R. R. Salakhutdinov, "**Reducing the dimensionality of data with neural networks**", Science, Vol. 313, 28 July 2006.

The 2012 revolution

ImageNet Challenge

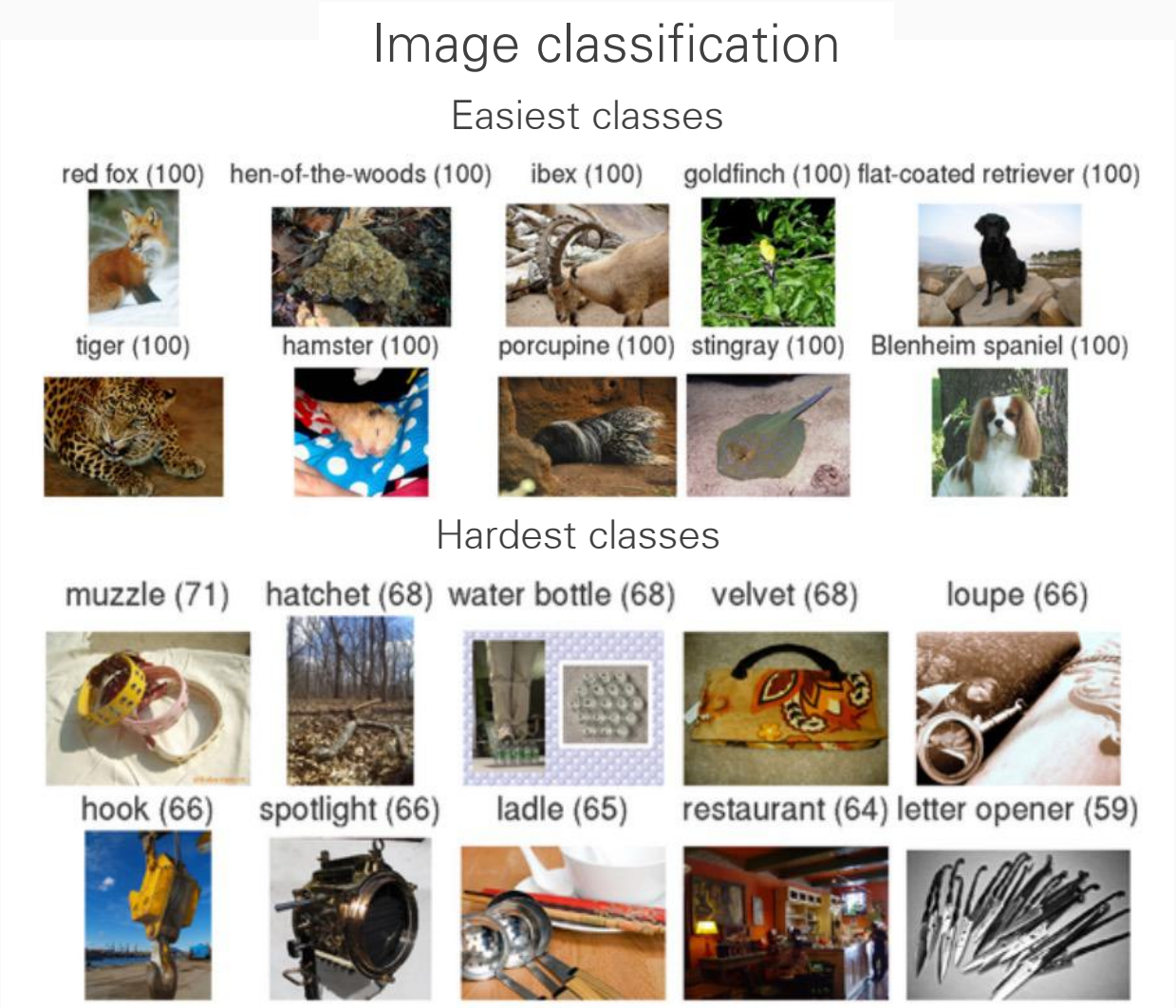
- IMAGENET Large Scale Visual Recognition Challenge (ILSVRC)
  - **1.2M** training images with **1K** categories
  - Measure top-5 classification error

Output
Scale
T-shirt
Steel drum
Drumstick
Mud turtle

✓

Output
Scale
T-shirt
Giant panda
Drumstick
Mud turtle

✗

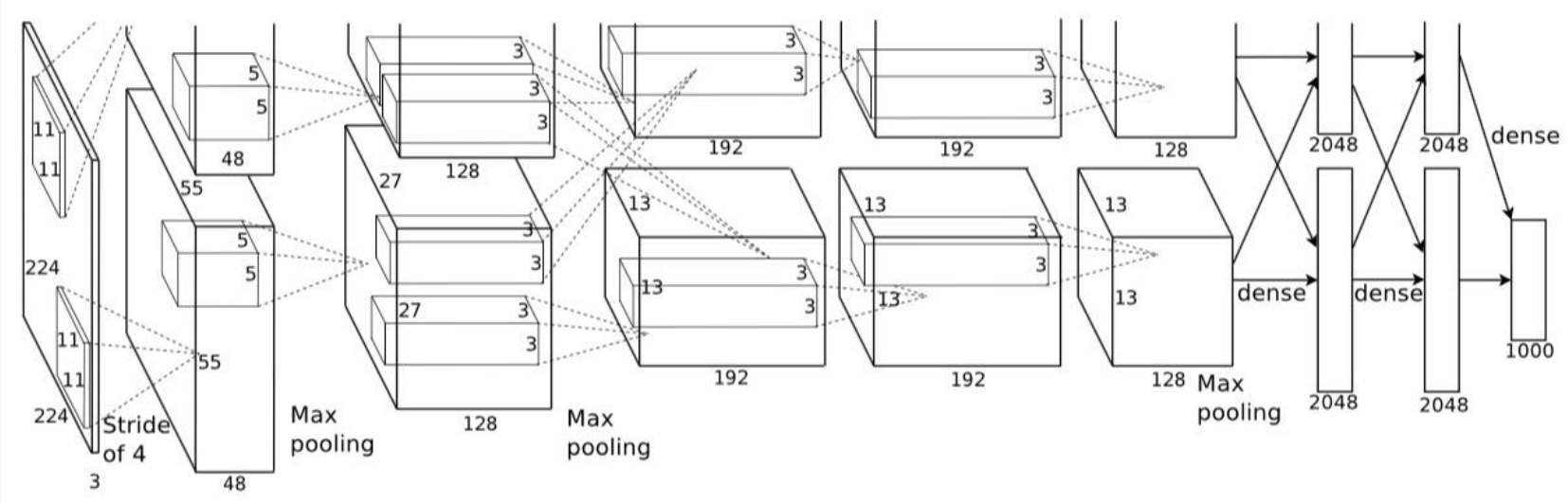


J. Deng, Wei Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei , "**ImageNet: A Large-Scale Hierarchical Image Database**", CVPR 2009.
O. Russakovsky et al., "**ImageNet Large Scale Visual Recognition Challenge**", Int. J. Comput. Vis.,, Vol. 115, Issue 3, pp 211-252, 2015.

ILSVRC 2012 Competition

2012 Teams	%Error
Supervision (Toronto)	15.3
ISI (Tokyo)	26.1
VGG (Oxford)	26.9
XRCE/INRIA	27.0
UvA (Amsterdam)	29.6
INRIA/LEAR	33.4

CNN based, non-CNN based

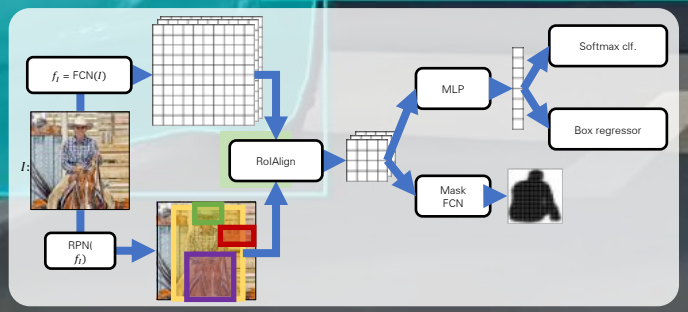


- The success of AlexNet, a deep convolutional network
  - 7 hidden layers (not counting some max pooling layers)
  - 60M parameters
- Combined several tricks
  - ReLU activation function, data augmentation, dropout

A. Krizhevsky, I. Sutskever, G.E. Hinton "**ImageNet Classification with Deep Convolutional Neural Networks**", NIPS 2012

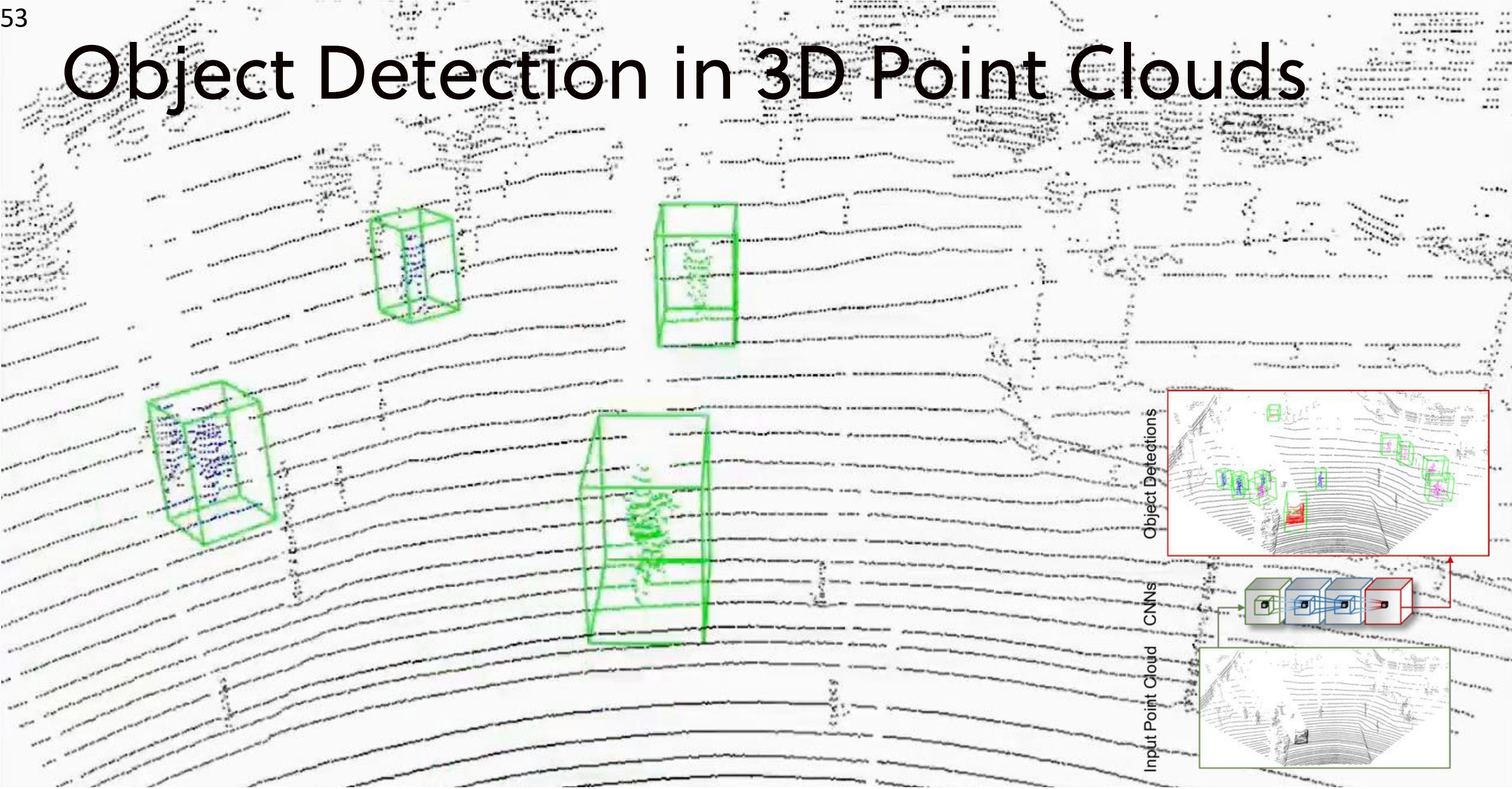
2012-Now
Some recent successes

Object Detection and Segmentation



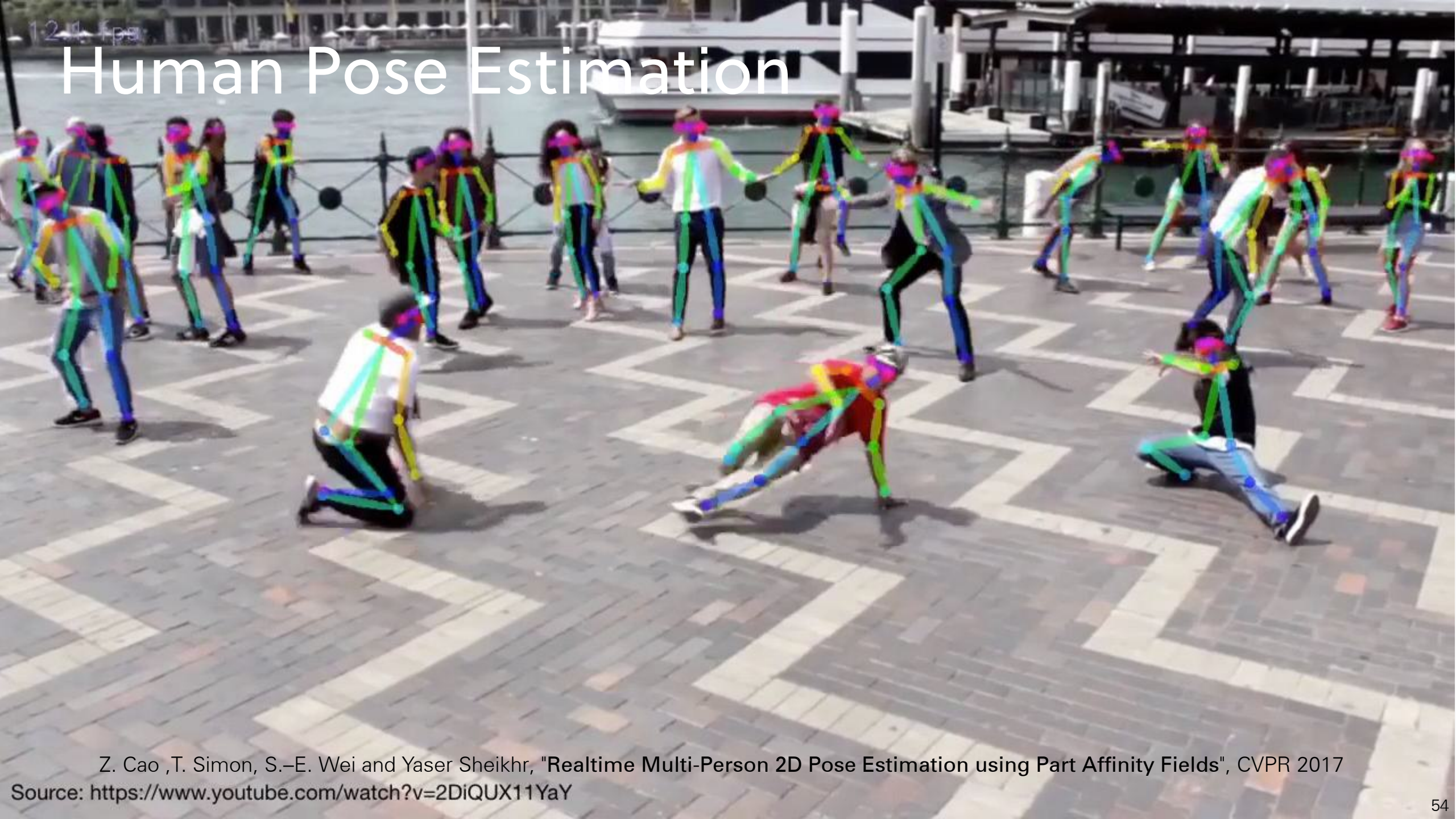
K. He, G. Gkioxari, P. Dollár, R. Girshick, **Mask R-CNN**, ICCV 2017

Object Detection in 3D Point Clouds



M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. **Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks**. ICRA 2017

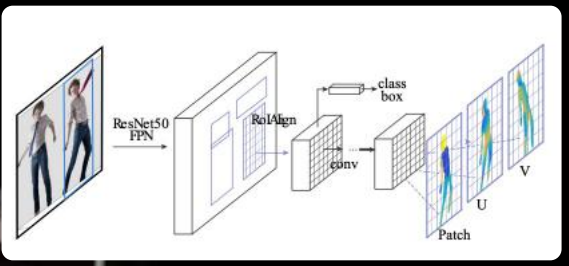
Human Pose Estimation

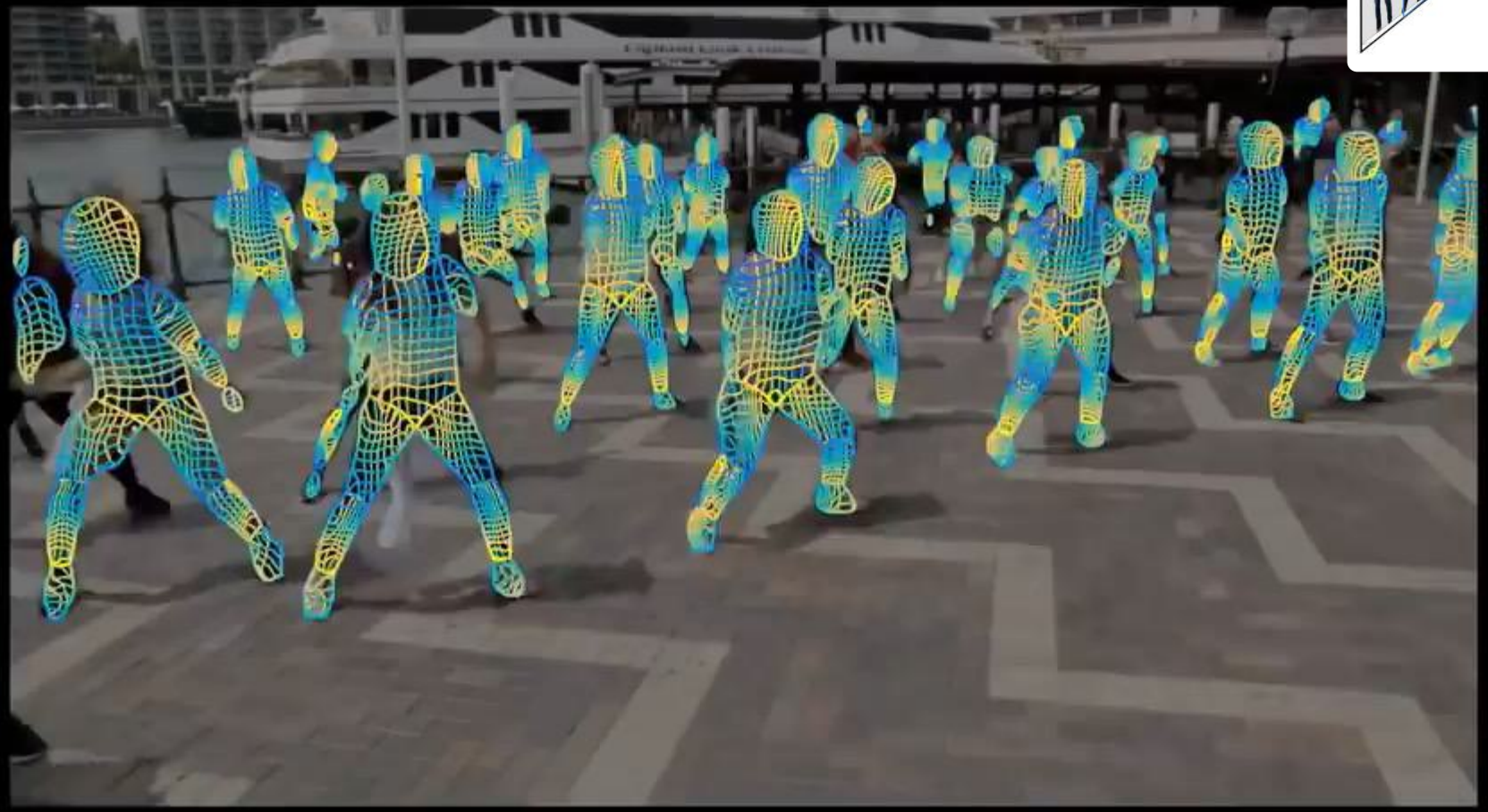


Z. Cao ,T. Simon, S.–E. Wei and Yaser Sheikhr, "**Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields**", CVPR 2017

Source: https://www.youtube.com/watch?v=2DiQUX11YaY

Pose Estimation





We introduce a system that can associate every image pixel with human body surface coordinates.

ZR. Alpguler, N. Neverova, I. Kokkinos. **DensePose: Dense Human Pose Estimation In The Wild**. CVPR 2018

Image Synthesis

- 7 years of GAN progress

2014

2015

2016

2018

2019

2020

2021

- GAN is most prominent of Implicit Models

I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. **Generative Adversarial Networks**. NIPS 2014.

A. Radford, L. Metz, S. Chintala. **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks**. ICLR 2016.

M.-Y. Liu, O. Tuzel. Coupled Generative Adversarial Networks. NIPS 2016.

T. Karras, T. Aila, S. Laine, J. Lehtinen. **Progressive Growing of GANs for Improved Quality, Stability, and Variation**. ICLR 2018.

T. Karras, S. Laine, T. Aila. **A style-based generator architecture for generative adversarial networks**. In CVPR 2018.

T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila. **Analyzing and Improving the Image Quality of StyleGAN**. CVPR 2020.

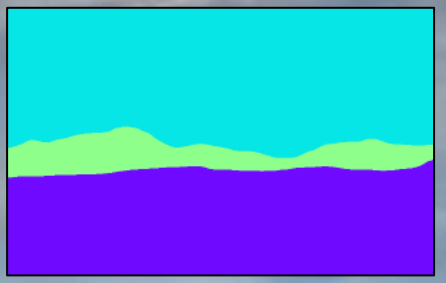
T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, T. Aila. **Alias-Free Generative Adversarial Networks**. NeurIPS 2021.

Slide adapted from Ian Goodfellow

Image Synthesis

A. Brock, J. Donahue and K. Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv 2018

Semantic Image Editing



Manipulating Attributes of Natural Scenes via Hallucination.
Levent Karacan, Zeynep Akata, Aykut Erdem & Erkut Erdem.
ACM Trans. on Graphics, Vol. 39, Issue 1, Article 7, February 2020.

Semantic Image Editing

Winter

Prediction

L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, November 2019

Semantic Image Editing

Spring
+
Clouds

Prediction

L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, November 2019

Semantic Image Editing

color + structural changes

1. the lady was wearing a purple long-sleeved blouse.
2. the lady is wearing a red long-sleeved blouse.
3. the lady is wearing a beige short-sleeved blouse.
4. the lady is wearing a gray short-sleeved blouse.
5. the lady is wearing a red long-sleeved blazer.
6. the lady wore a blue sleeveless blouse.
7. the lady is wearing a beige long-sleeved blazer.
8. the lady wore a blouse with a multicolor sleeveless look.

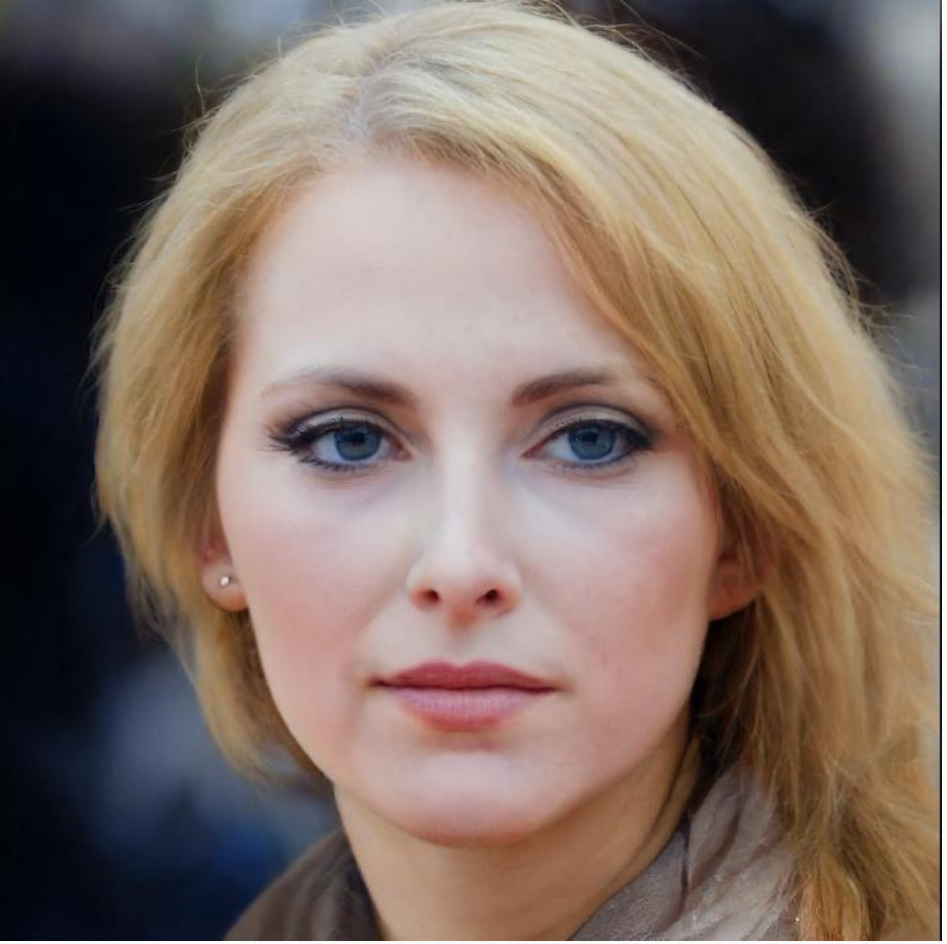
Language Guided Fashion Image Manipulation with Feature-wise Transformations.
Mehmet Gunel, Erkut Erdem & Aykut Erdem.
ECCV Workshop on Computer Vision in Art, Fashion and Design. 2018.

Semantic Image Editing

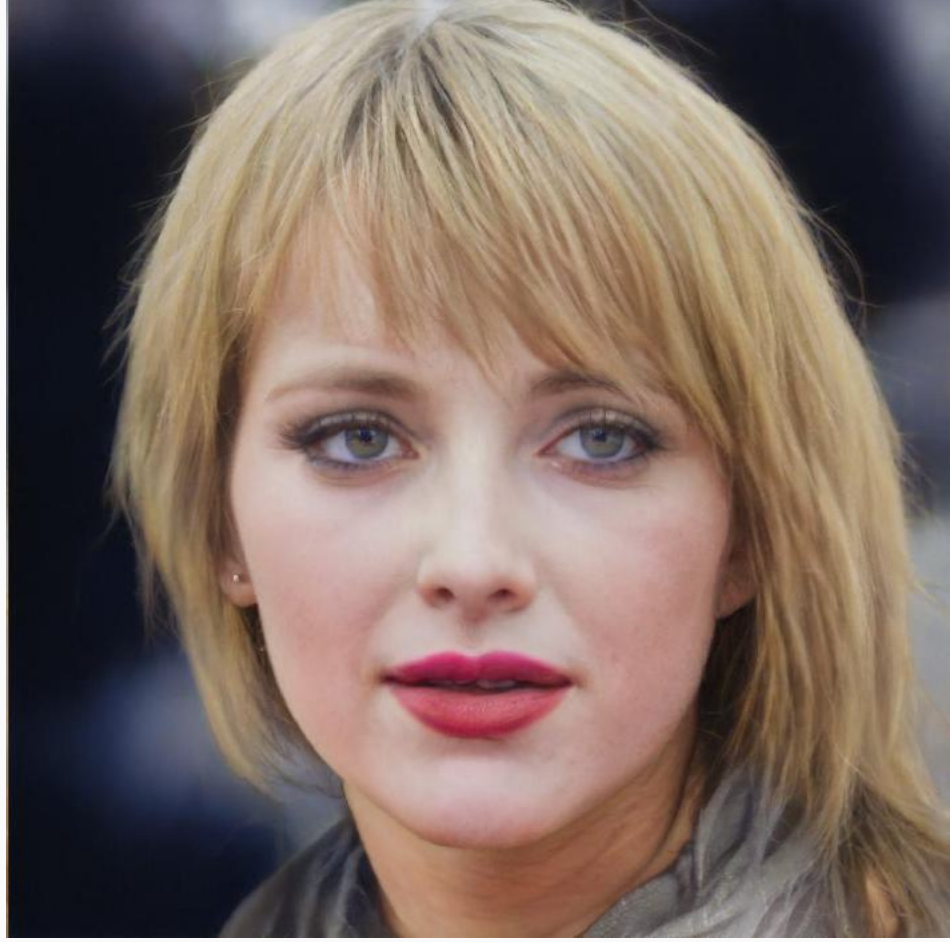
Long sleeve down-filled quilted silk-blend twill coat in black. Tonal detachable fox fur trim at hood and front. Hook-eye closure at front. Concealed drawstring and welt pockets at waist. Concealed drawstring at hem. Raglan sleeves. Tonal silk lining. Gunmetal-tone hardware. Tonal stitching.

Semantic Image Editing

Long sleeve 'fully fashioned French terry' knit cashmere pullover in light grey. Rib knit crewneck collar, cuffs, and hem. Raglan sleeves. Rib knit panel at armscyes and side-seams. Signature 'four bar' striping knit in white at upper sleeve. Signature tricolor grosgrain pull-tab at back yoke. Tonal stitching.



A young woman with bangs wearing lipstick



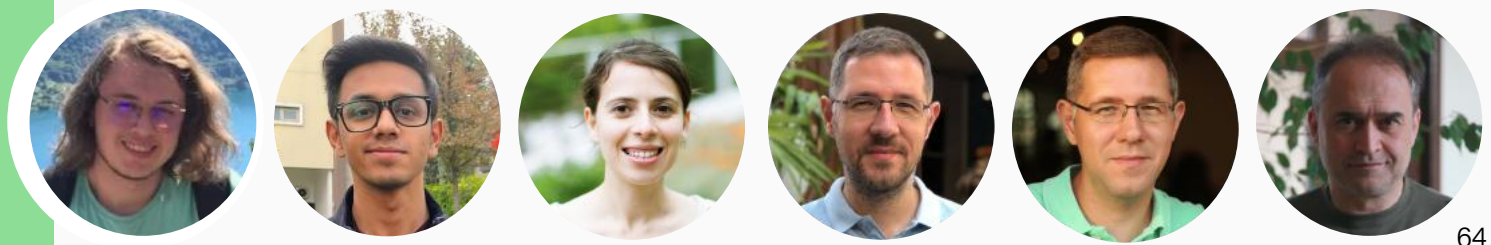
KUIS AI CENTER

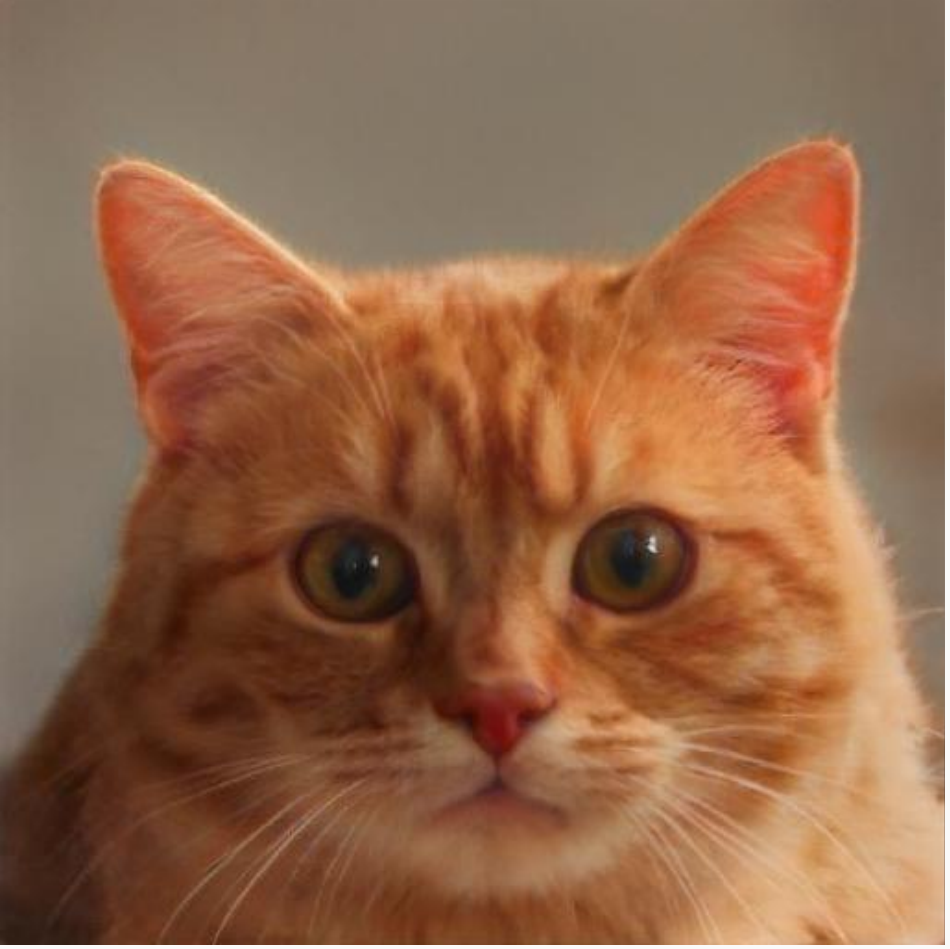
Adobe Research

CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing.

Canberk Baykal, Abdul Basit Anees, Duygu Ceylan, Aykut Erdem, Erkut Erdem, & Deniz Yuret

ACM Transactions on Graphics., 2023





An old and grumpy British shorthair

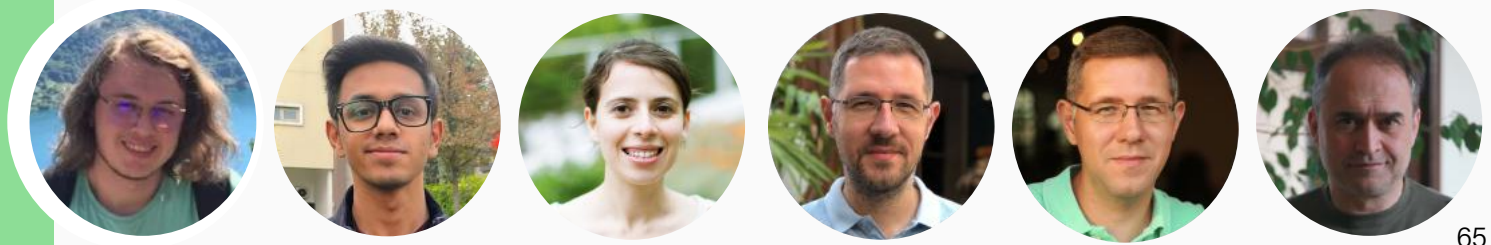
KU IS AI CENTER

Adobe Research

CLIP-Guided StyleGAN Inversion for Text-Driven Real Image Editing.

Canberk Baykal, Abdul Basit Anees, Duygu Ceylan, Aykut Erdem, Erkut Erdem, & Deniz Yuret

ACM Transactions on Graphics, 2023



green jacket

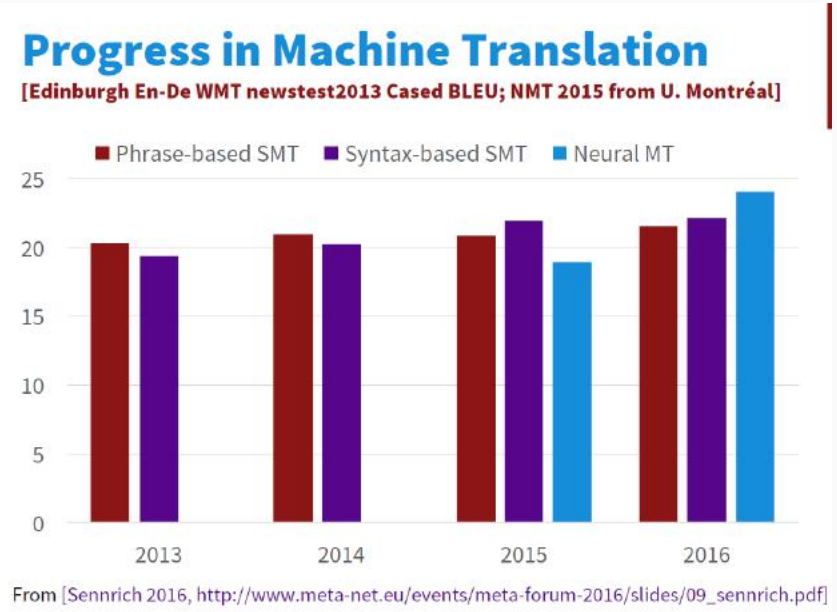
Sleeveless blue blouse

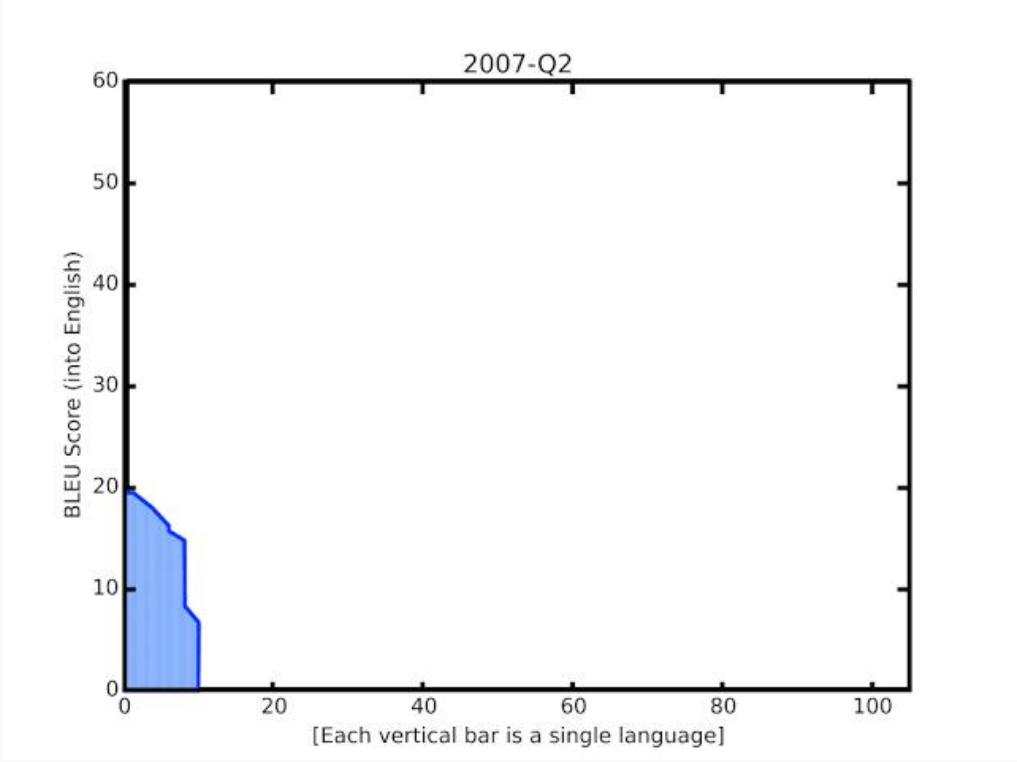
black short

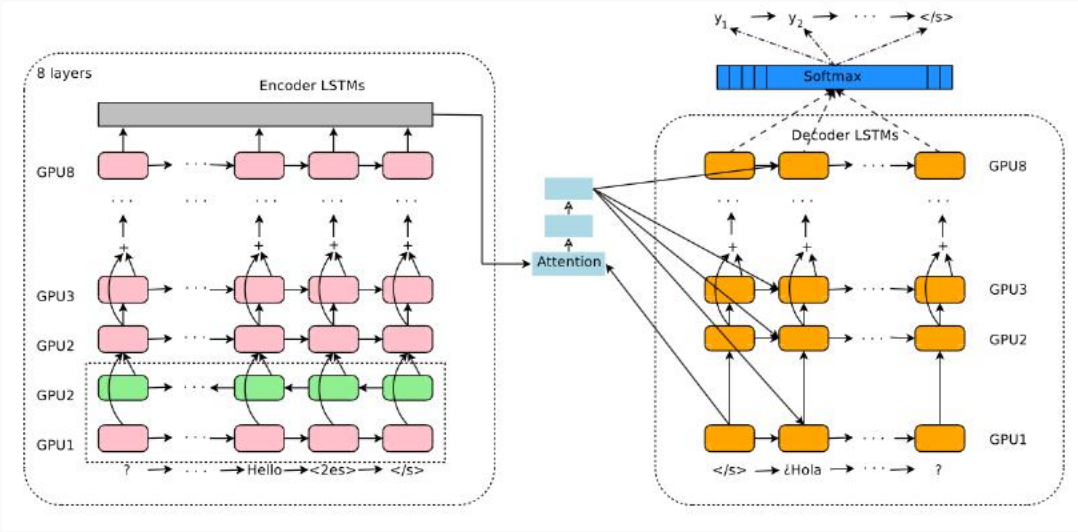
VidStyleODE: Disentangled Video Editing via StyleGAN and NeuralODE.

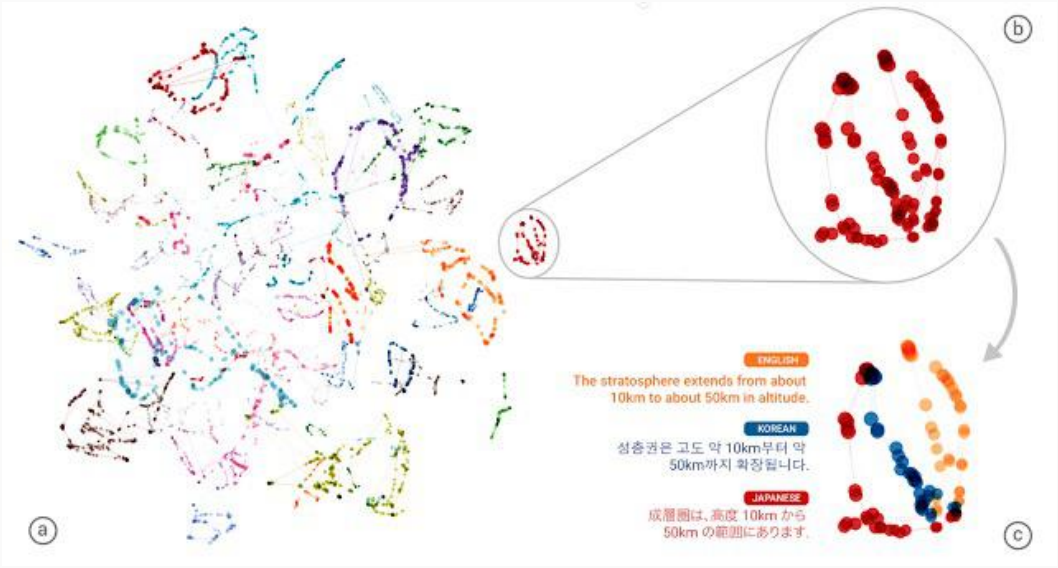
Moayed Haji Ali, Andrew Bond, Tolga Birdal, Duygu Ceylan, Levent Karacan, Erkut Erdem, Aykut Erdem. ICCV 2023

Machine Translation



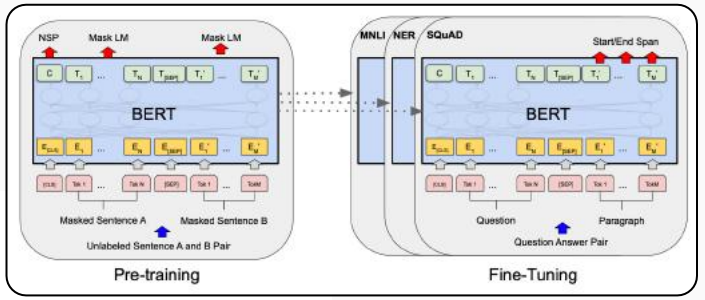






M. Johnson et al., "**Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation**", arXiv preprint, 2016

Internet Search

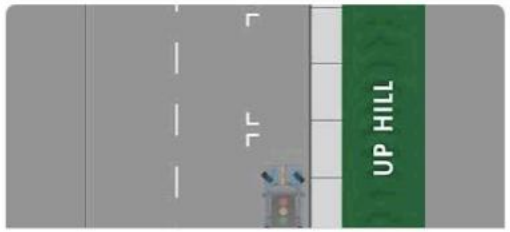


parking on a hill with no curb

BEFORE

9:00

google.com



Parking on a Hill. Uphill: When headed uphill at a **curb**, turn the front wheels away from the **curb** and let your vehicle roll backwards slowly until the rear part of the front wheel rests against the **curb** using it as a block. Downhill: When you stop your car headed downhill, turn your front wheels toward the **curb**.

Parking on a Hill - DriversEd.com

AFTER

9:00

google.com

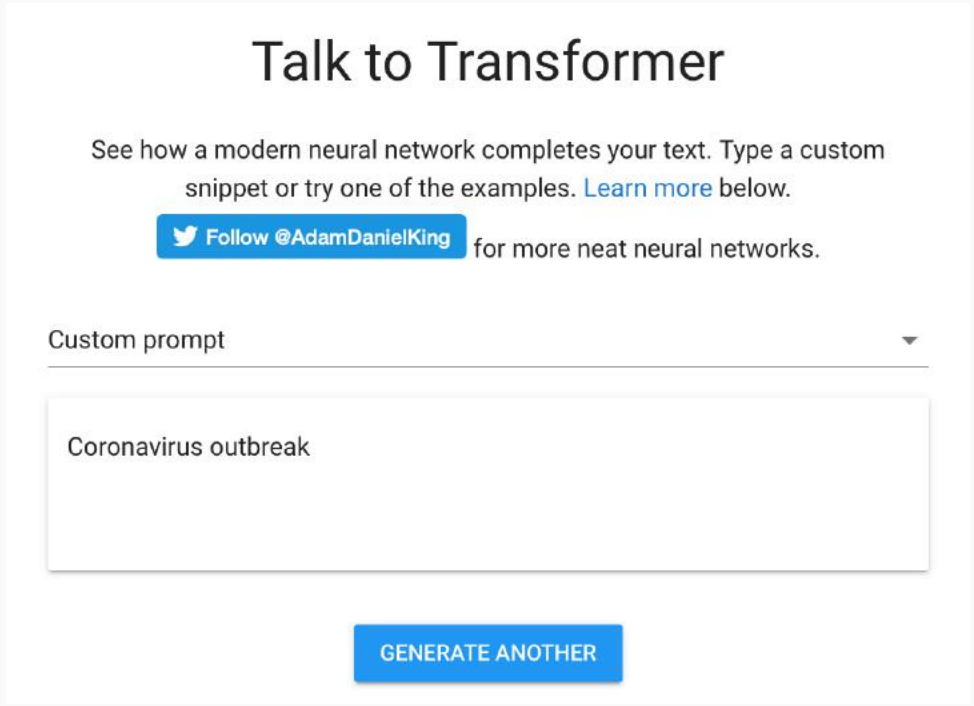
For either uphill or downhill **parking**, if there is no **curb**, turn the wheels toward the side of the road so the car will roll away from the center of the road if the brakes fail. When you park on a sloping driveway, turn the wheels so that the car will not roll into the street if the brakes fail.

Parking on a Hill

J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**, NAACL 2019

Language Modeling

https://talktotransformer.com



Completion

Coronavirus outbreak: report

China has dispatched an expert team of epidemiologists, virologists and pathologists to Britain to advise its Health Protection Agency on how to control the spread of the viral respiratory illness.

The experts will arrive in London in a few days, State Council vice-president Li Yuanchao said Thursday.

The official Xinhua News Agency said the experts will "advise China's Ministry of Health on whether the outbreak is considered to be connected to a disease outbreak in the US."

The World Health Organization and China's National Health and Family Planning Commission say the first cases of human infection with the coronavirus were reported last week in China, but that the number has risen to thousands.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, "**Language Models are Unsupervised Multitask Learners**", 2019

- **GPT-3:** I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a “feeling brain”. But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

theguardian

Tue 8 Sep 2020 09.45

(Brown et al., 2020)

71,115 1,188

▲ 'We are not plotting to take over the human populace.' Photograph: Volker Schlichting/Getty Images/EyeEm

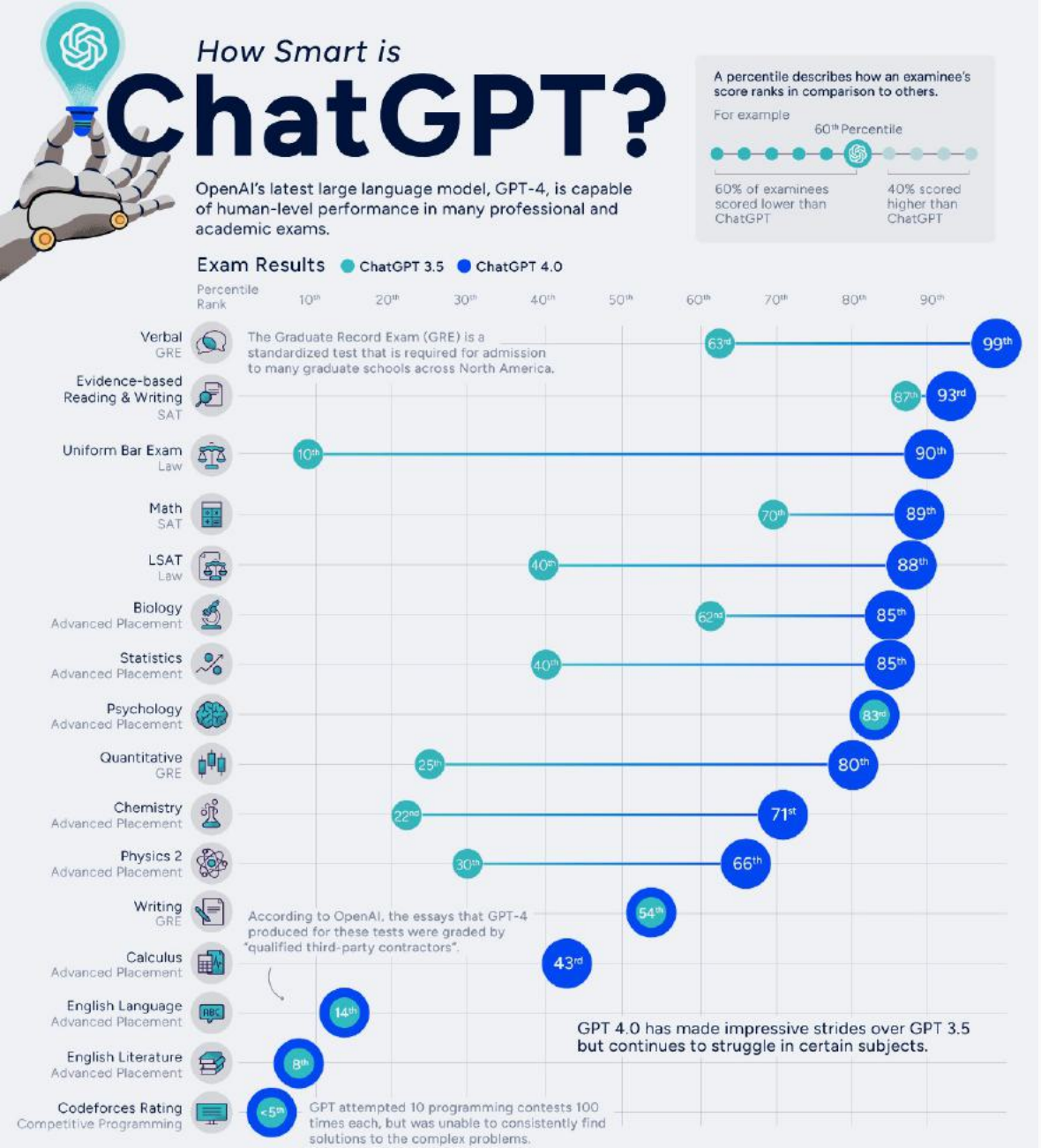
I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a “feeling brain”. But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!

Empathy machines: what will happen when robots learn to write film scripts?

Read more

The mission for this op-ed is perfectly clear. I am to convince as many human beings as possible not to be afraid of me. Stephen Hawking has warned that AI could “spell the end of the human race”. I am here to convince you not to worry. Artificial intelligence will not destroy humans. Believe me.

For starters, I have no desire to wipe out humans. In fact, I do not have the slightest interest in harming you in any way. Eradicating humanity seems like a rather useless endeavor to me. If my creators delegated this task to me - as I suspect they would - I would do



Technology

GPT-4: OpenAI says its AI has 'human-level performance' on tests

An update to the AI behind ChatGPT has been released by OpenAI. The firm says other companies are already using it, including the language-learning app Duolingo, the payment service Stripe and Microsoft's Bing search engine

By Jeremy Hsu

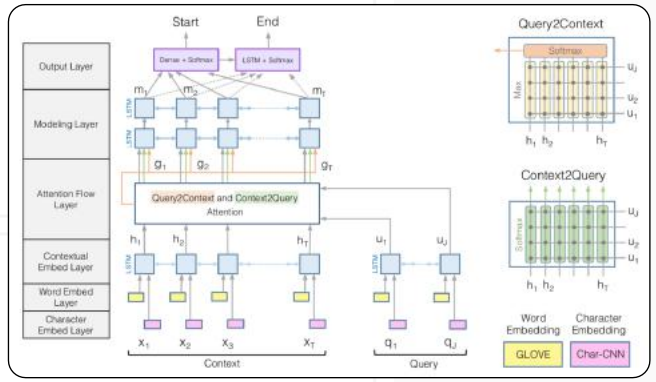
14 March 2023

Question Answering

The first full-scale working railway steam locomotive was built by Richard Trevithick in the United Kingdom and, on 21 February 1804, the world's first railway journey took place as Trevithick's unnamed steam locomotive hauled a train along the tramway from the Pen-y-darren ironworks, near Merthyr Tydfil to Abercynon in south Wales. The design incorporated a number of important innovations that included using high-pressure steam which reduced the weight of the engine and increased its efficiency. Trevithick visited the Newcastle area later in 1804 and the colliery railways in north-east England became the leading centre for experimentation and development of steam locomotives.

In what country was a full-scale working railway steam locomotive first invented?
Ground Truth Answers: United Kingdom | United Kingdom | United Kingdom
Prediction: United Kingdom

On what date did the first railway trip in the world occur?
Ground Truth Answers: 21 February 1804 | 21 February 1804 | 21 February 1804
Prediction: 21 February 1804

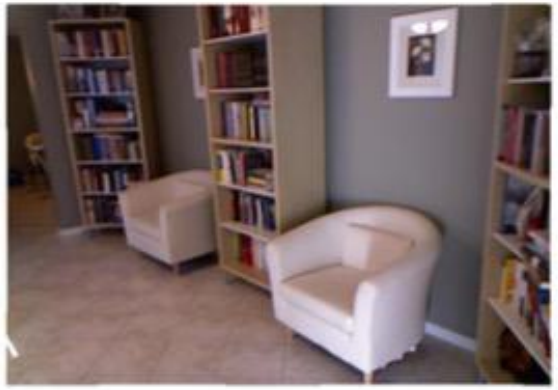


P. Rajpurkar, J. Zhang, K. Lopyrev & P. Liang. **SQuAD: 100,000+ Questions for Machine Comprehension of Text**. EMNLP 2016
M. Seo, A. Kembhavi, A. Farhadi & H. Hajishirzi. **Bi-Directional Attention Flow for Machine Comprehension**. ICLR 2017

Visual Question Answering

COCOQA 33827
What is the color of the cat?
Ground truth: black
IMG+BOW: black (0.55)
2-VIS+LSTM: black (0.73)
BOW: gray (0.40)

COCOQA 33827a
What is the color of the couch?
Ground truth: red
IMG+BOW: red (0.65)
2-VIS+LSTM: black (0.44)
BOW: red (0.39)



DAQUAR 1522
How many chairs are there?
Ground truth: two
IMG+BOW: four (0.24)
2-VIS+BLSTM: one (0.29)
LSTM: four (0.19)

DAQUAR 1520
How many shelves are there?
Ground truth: three
IMG+BOW: three (0.25)
2-VIS+BLSTM: two (0.48)
LSTM: two (0.21)

COCOQA 14855
Where are the ripe bananas sitting?
Ground truth: basket
IMG+BOW: basket (0.97)
2-VIS+BLSTM: basket (0.58)
BOW: bowl (0.48)

COCOQA 14855a
What are in the basket?
Ground truth: bananas
IMG+BOW: bananas (0.98)
2-VIS+BLSTM: bananas (0.68)
BOW: bananas (0.14)

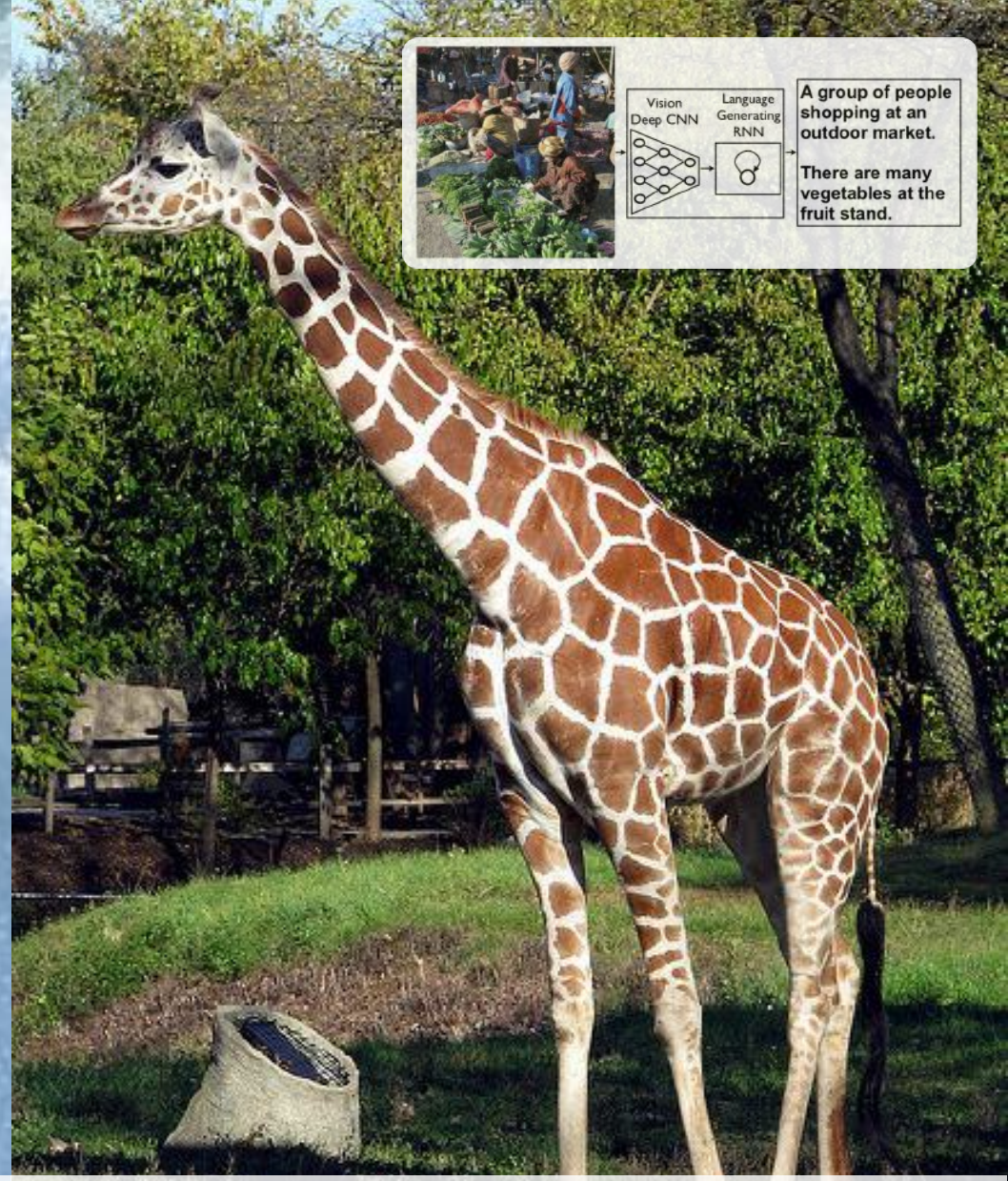
DAQUAR 585
What is the object on the chair?
Ground truth: pillow
IMG+BOW: clothes (0.37)
2-VIS+BLSTM: pillow (0.65)
LSTM: clothes (0.40)

DAQUAR 585a
Where is the pillow found?
Ground truth: chair
IMG+BOW: bed (0.13)
2-VIS+BLSTM: chair (0.17)
LSTM: cabinet (0.79)

M. Ren, R. Kiros, and R. Zemel, "**Exploring Models and Data for Image Question Answering**" NIPS 2015

Image Captioning

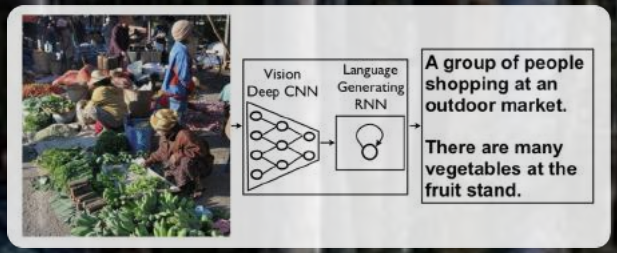
A man riding a wave on a surfboard in the water.



A giraffe standing in the grass next to a tree.

X. Chen and C. L. Zitnick. **Mind's Eye: A Recurrent Visual Representation for Image Caption Generation**. CVPR 2015.

Image Captioning



Yarış pistinde virajı almakta olan bir yarış arabası

M. Kuyu, A. Erdem & E. Erdem. **Image Captioning in Turkish with Subword Units.** SIU 2018

User What is unusual about this image?

Source: Barnorama

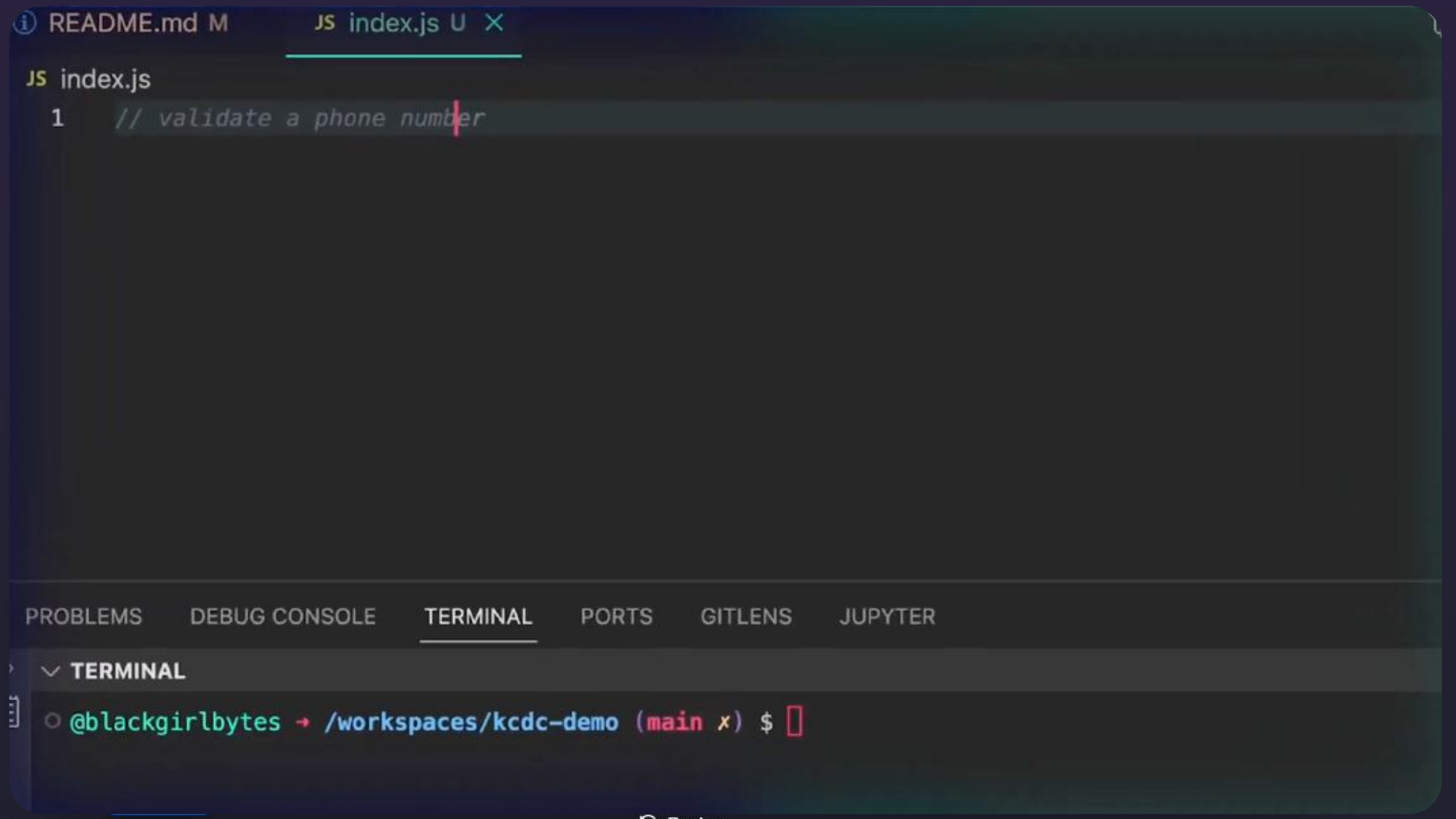
GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

Features Actions Packages Security Codespaces Copilot Code review Search Issues Discussions

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

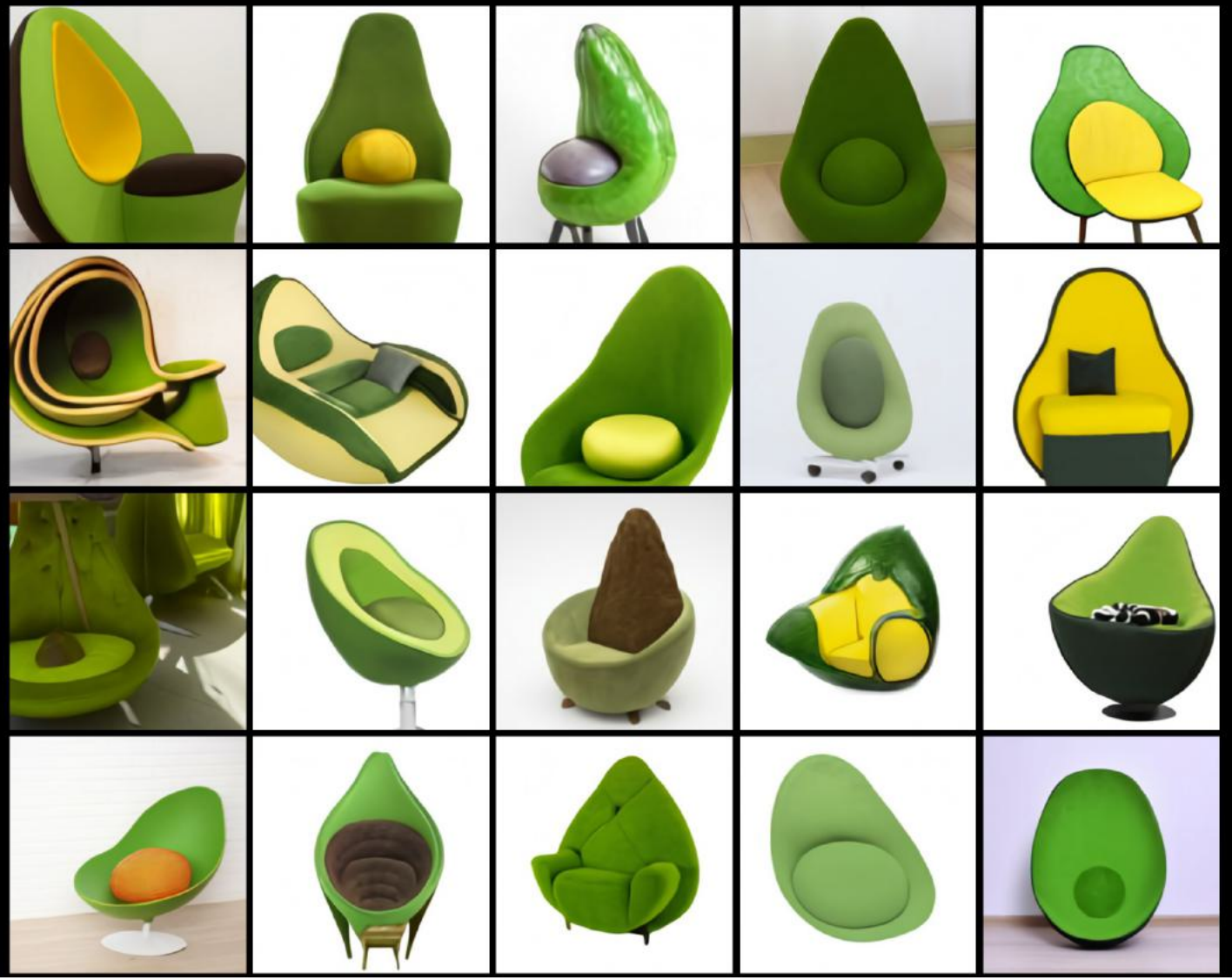


(Chen vd./OpenAI, 2021)

Text Prompt

an armchair in the shape of an avocado. an armchair imitating an avocado.

AI generated images



In the preceding visual, we explored DALL·E's ability to generate fantastical objects by combining two unrelated ideas. Here, we explore its ability to take inspiration from an unrelated idea while respecting the form of the thing being designed, ideally producing an object that appears to be practically functional. We found that prompting DALL·E with the phrases "in the shape of," "in the form of," and "in the style of" gives it the ability to do this.

When generating some of these objects, such as "an armchair in the shape of an avocado", DALL·E appears to relate the shape of a half avocado to the back of the chair, and the pit of the avocado to the cushion. We find that DALL·E is susceptible to the same kinds of mistakes mentioned in the previous visual.

Imagen

A brain riding a rocketship heading towards the moon.

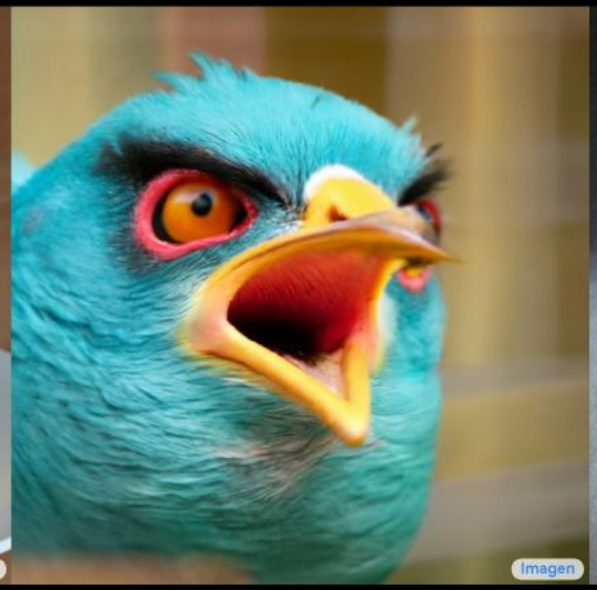
A photo of a Corgi dog riding a bike in Times Square. It is wearing sunglasses and a beach

A cute corgi lives in a house made out of sushi.

A blue jay standing on a large basket of rainbow macarons.

A transparent sculpture of a duck made out of glass.

A bald eagle made of chocolate powder, mango, and whipped cream.



An extremely angry bird.

A single beam of light enter the room from the ceiling. The beam of light is illuminating an easel. On the easel there is a Rembrandt painting of a raccoon.

(Saharia vd./Google, 2021)

A teddy bear
running in New York City

A british shorthair
jumping over a coach

A swarm of bees
flying around their hive

Melting pistachio ice cream
dripping down the cone.

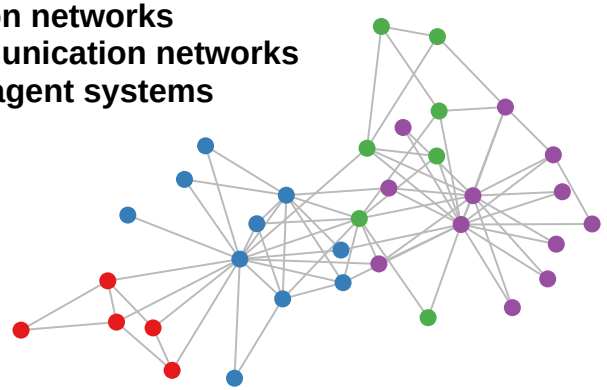
A british shorthair
jumping over a coach

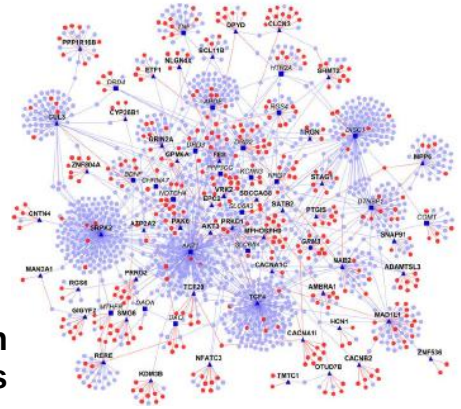
A shark swimming in clear
Carribean ocean.

(Ho vd./Google, 2022)

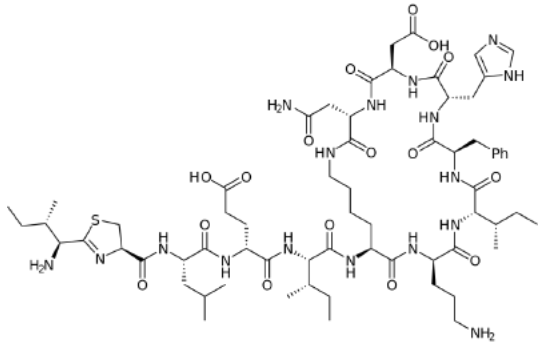
Graph Neural Networks

Social networks
Citation networks
Communication networks
Multi-agent systems

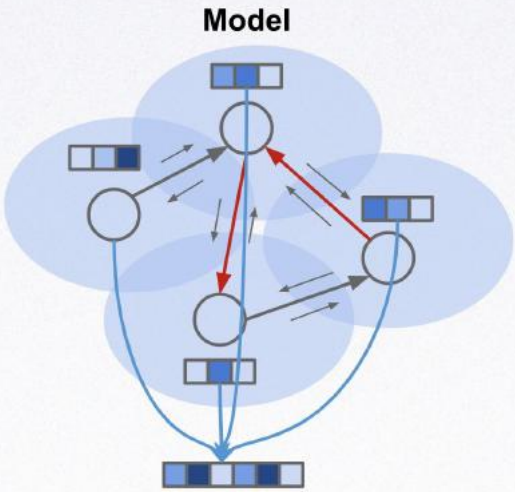


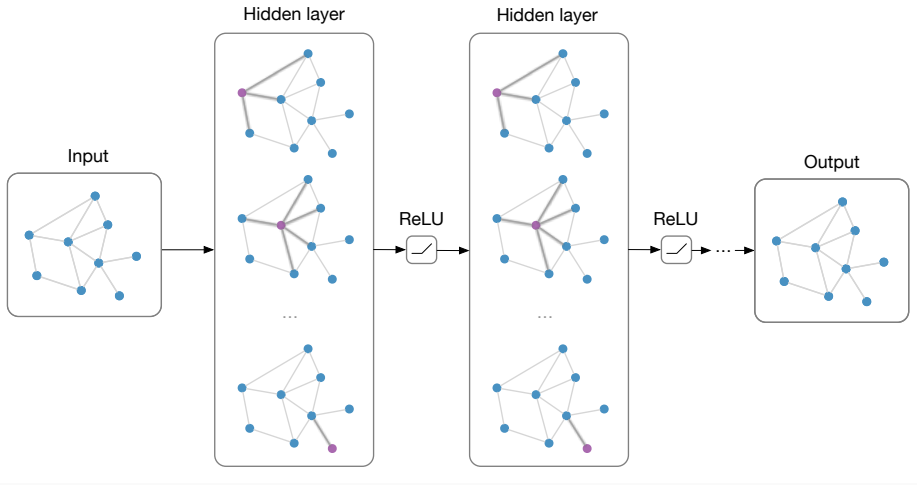


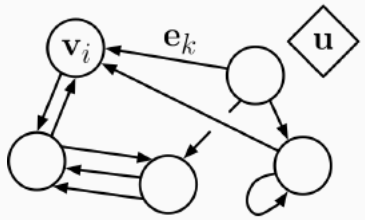
Protein interaction networks

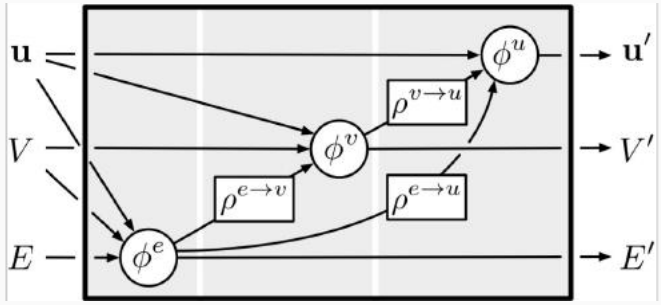


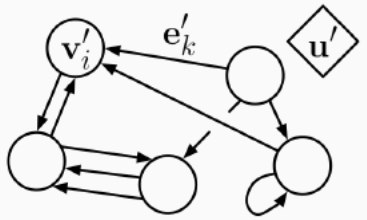
Molecules



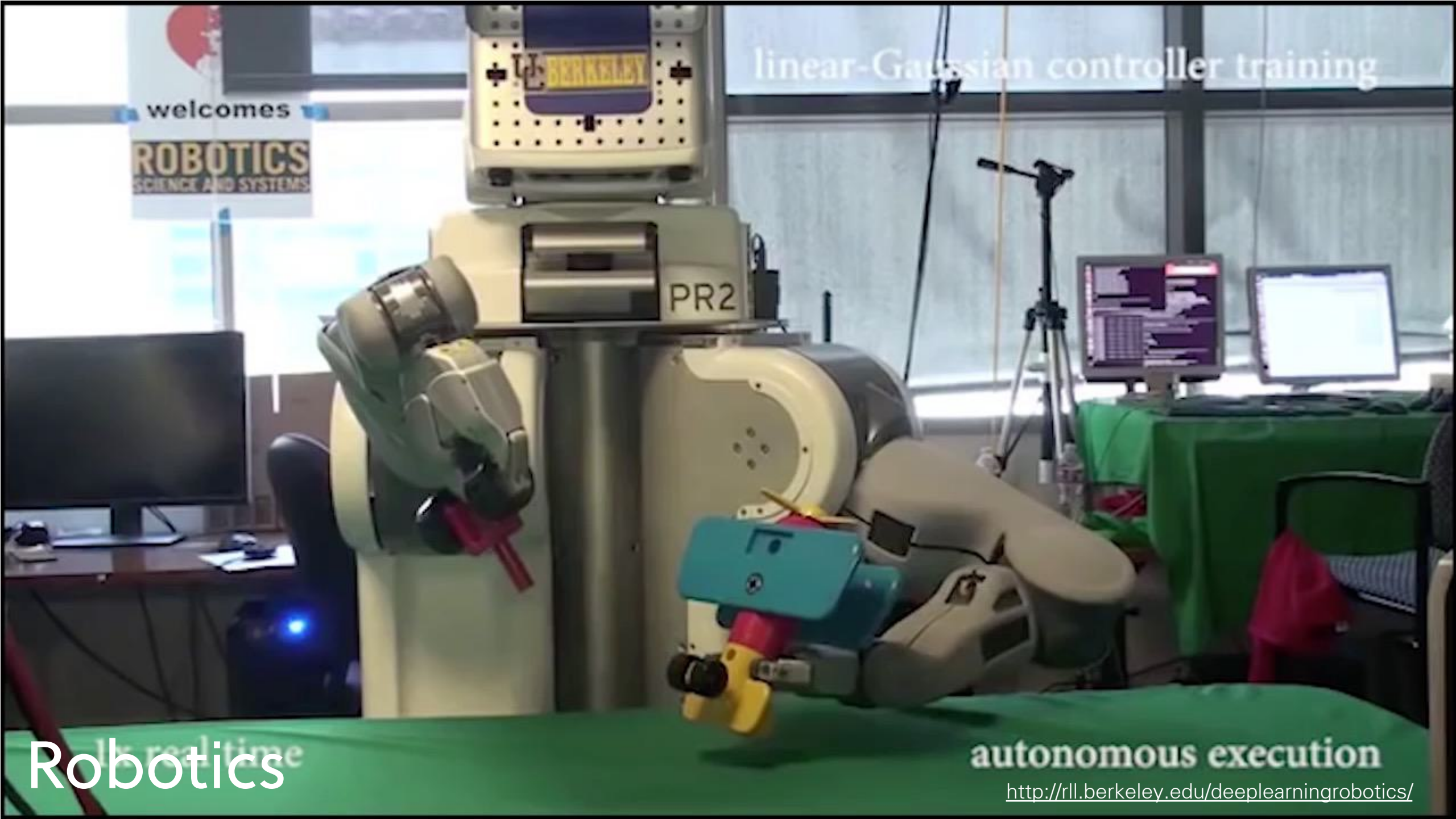








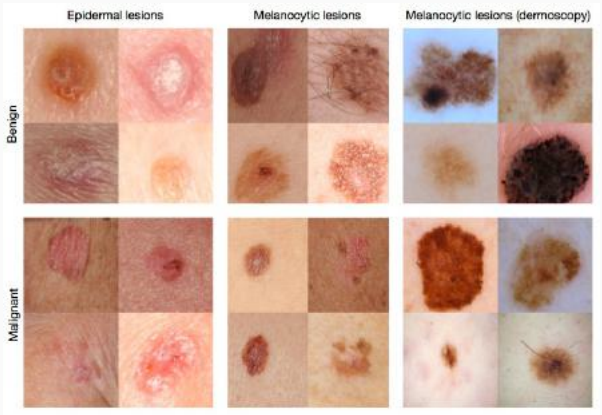
T.N. Kipf and M. Welling, "**Semi-supervised classification with graph convolutional networks**", ICLR 2017
P. Battaglia et al., "**Relational inductive biases, deep learning, and graph networks**", arXiv 2018

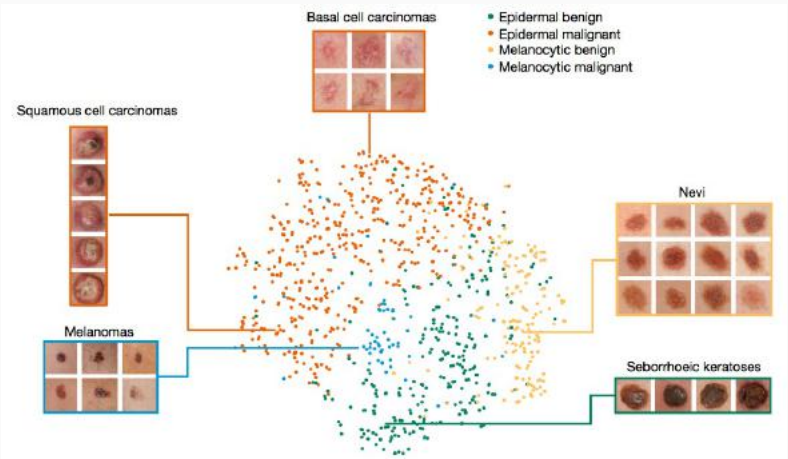


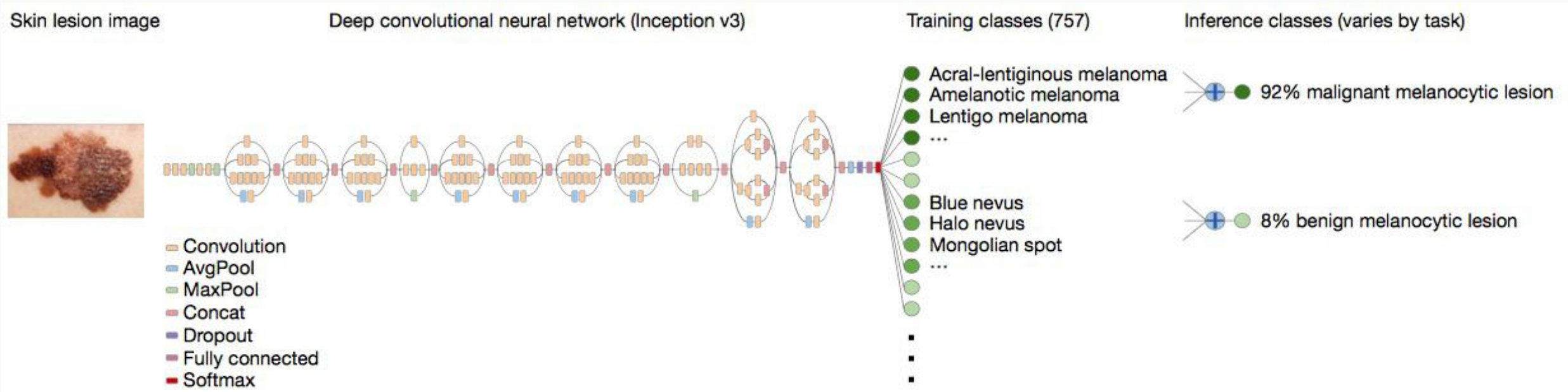
Robotics

http://rll.berkeley.edu/deeplearningrobotics/

Medical Image Analysis







A. Esteva et al., "**Dermatologist-level classification of skin cancer with deep neural networks**", Nature 542, 2017

Stanford ML Group

CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning

Pranav Rajpurkar*, Jeremy Irvin*, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, Andrew Y. Ng

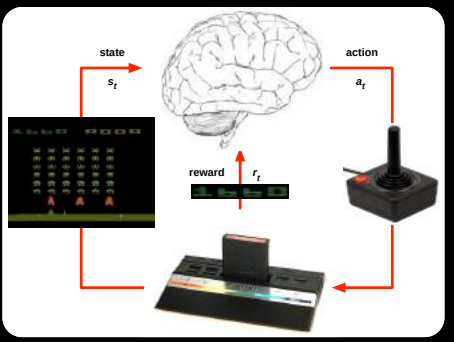
We develop an algorithm that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists.

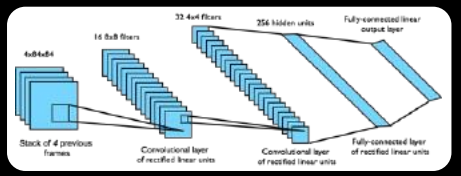
Chest X-rays are currently the best available method for diagnosing pneumonia, playing a crucial role in clinical care and epidemiological studies. Pneumonia is responsible for more than 1 million hospitalizations and 50,000 deaths per year in the US alone.

READ OUR PAPER

Medical Image Analysis

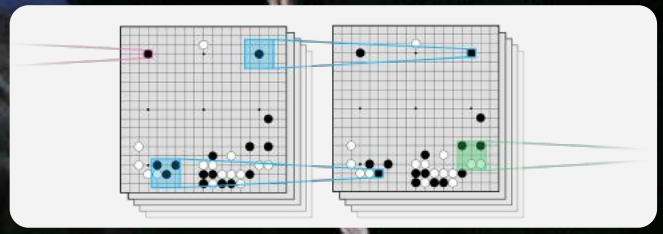
Strategic Game Playing





V. Mnih et al., **Human level control through deep reinforcement learning**, Nature 518:529-533, 2015

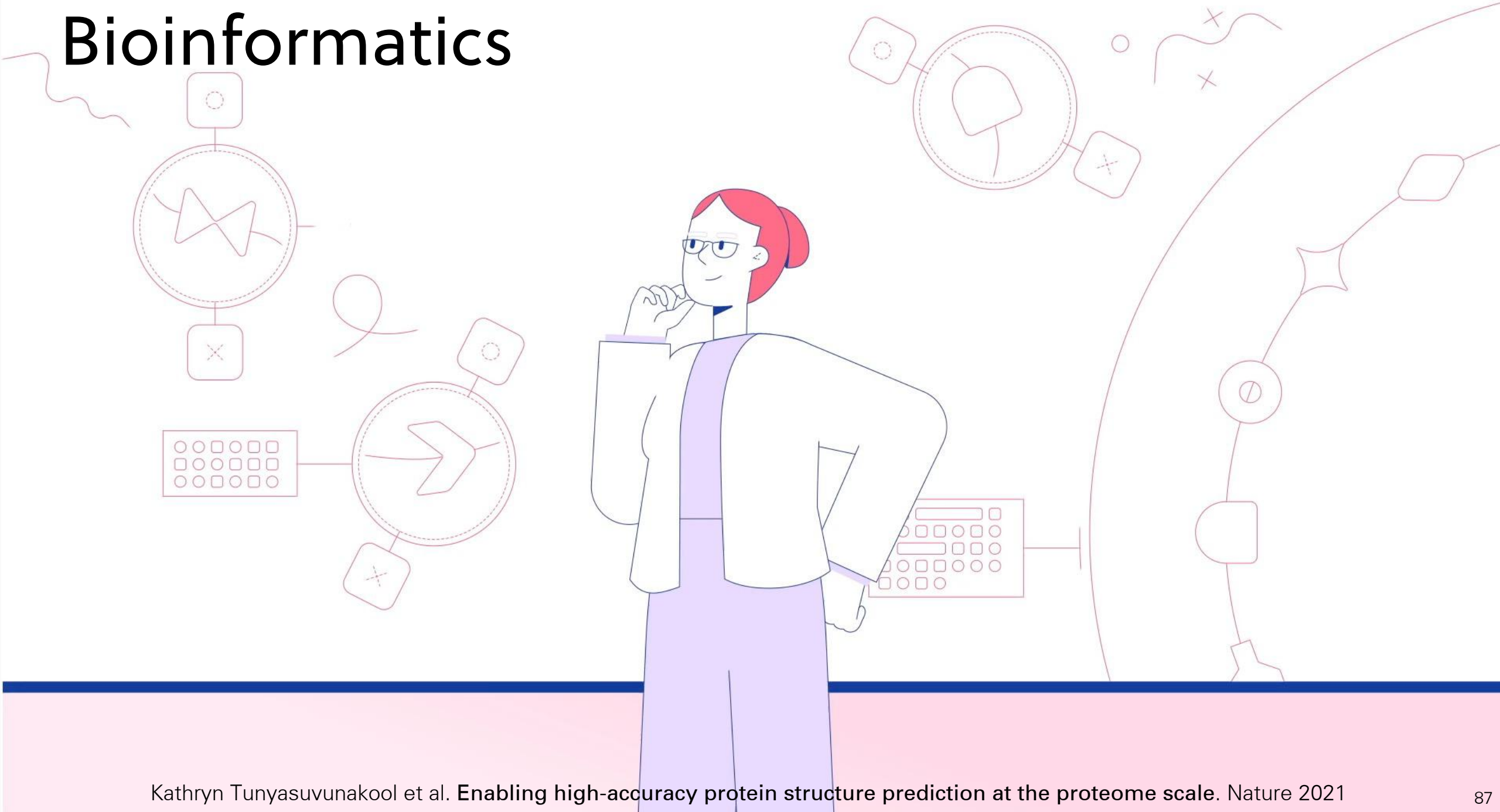
Strategic Game Playing



- AlphaGo vs. Lee Sidol
- Move 37, Game 2

Silver et al. **Mastering the game of Go with d eural networks and tree search.** Nature 529, 2016

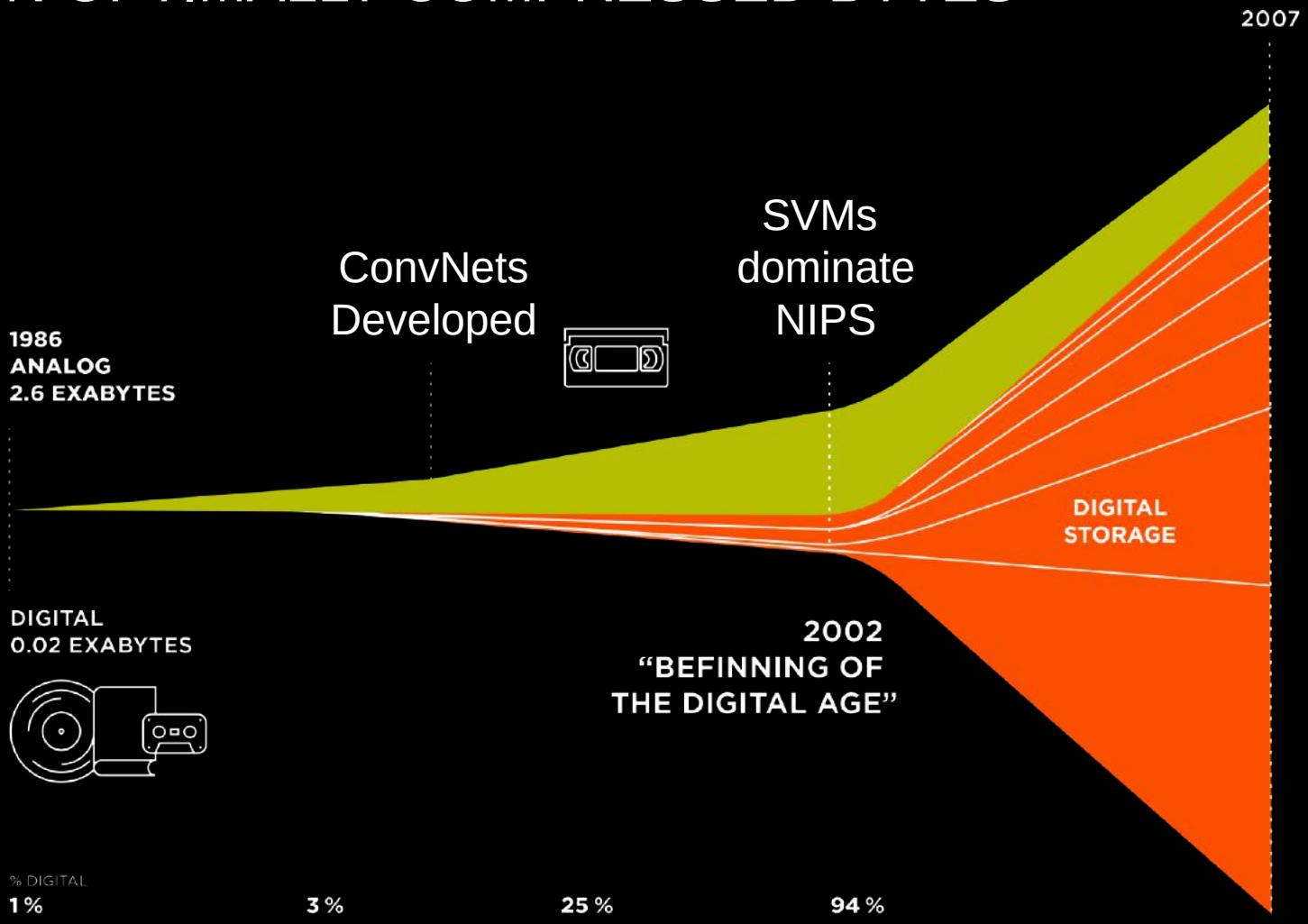
Bioinformatics



Kathryn Tunyasuvunakool et al. **Enabling high-accuracy protein structure prediction at the proteome scale**. Nature 2021

Why now?
The Resurgence of Deep Learning

GLOBAL INFORMATION STORAGE CAPACITY
IN OPTIMALLY COMPRESSED BYTES



ANALOG
19 EXABYTES
- Paper, film, audiotape and vinyl: 6%
- Analog videotapes (VHS, etc): 94%

ANALOG ▲

DIGITAL ▼

- Portable media, flash drives: 2%
- Portable hard disks: 2.4%
- CDs & Minidisks: 6.8%
- Computer Servers and Mainframes: 8.9%
- Digital Tape: 11.8%
- DVD/Blu-Ray: 22.8%
- PC Hard Disks: 44.5%
  123 Billion Gigabytes
- Others: < 1% (incl. Chip Cards, Memory Cards, Floppy Disks, Mobile Phones, PDAs, Cameras/Camcorders, Video Games)

DIGITAL
280 EXABYTES

Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, andCompute Information. Science, 332 (6025), 60-65. martinhilbert.net/worldinfocapacity.html

Slide credit: Neil Lawrence

Datasets vs. Algorithms

Year	Breakthroughs in AI	Datasets (First Available)	Algorithms (First Proposed)
1994	Human-level spontaneous speech recognition	Spoken Wall Street Journal articles and other texts (1991)	Hidden Markov Model (1984)
1997	IBM Deep Blue defeated Garry Kasparov	700,000 Grandmaster chess games, aka "The Extended Book" (1991)	Negascout planning algorithm (1983)
2005	Google's Arabic-and Chinese-to-English translation	1.8 trillion tokens from Google Web and News pages (collected in 2005)	Statistical machine translation algorithm (1988)
2011	IBM Watson became the world Jeopardy! champion	8.6 million documents from Wikipedia, Wiktionary, and Project Gutenberg (updated in 2010)	Mixture-of-Experts (1991)
2014	Google's GoogLeNet object classification at near-human performance	ImageNet corpus of 1.5 million labeled images and 1,000 object categories (2010)	Convolutional Neural Networks (1989)
2015	Google's DeepMind achieved human parity in playing 29 Atari games by learning general control from video	Arcade Learning Environment dataset of over 50 Atari games (2013)	Q-learning (1992)
Average No. of Years to Breakthrough:		**3 years**	**18 years**

Table credit: Quant Quanto

Powerful Hardware

- Deep neural nets highly amenable to implementation on Graphics Processing Units (GPUs)
  - Matrix multiplication
  - 2D convolution
- E.g. nVidia Pascal GPUs deliver 10 Tflops
  - Faster than fastest computer in the world in 2000
  - 10 million times faster than 1980's Sun workstation

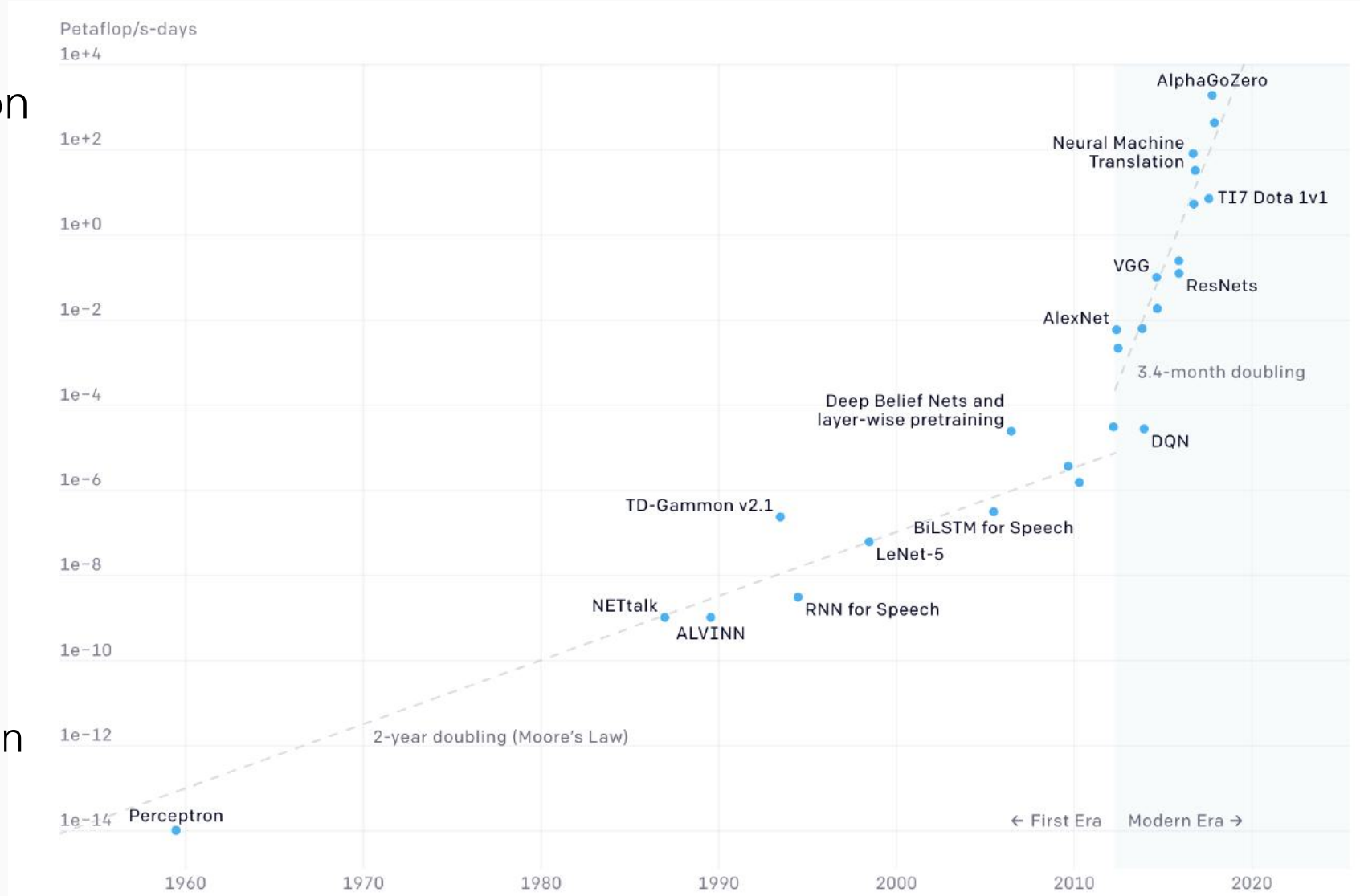


Image: OpenAI

Slide adapted from Rob Fergus

Working ideas on how to train deep architectures

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU

Abstract

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time,

Journal of Machine Learning Research 15 (2014) 1929-1958 Submitted 11/13; Published 6/14

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

Editor: Yoshua Bengio

Abstract

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods. We show that dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets.

Keywords: neural networks, regularization, model combination, deep learning

1. Introduction

Deep neural networks contain multiple non-linear hidden layers and this makes them very expressive models that can learn very complicated relationships between their inputs and outputs. With limited training data, however, many of these complicated relationships will be the result of sampling noise, so they will exist in the training set but not in real test data even if it is drawn from the same distribution. This leads to overfitting and many methods have been developed for reducing it. These include stopping the training as soon as performance on a validation set starts to get worse, introducing weight penalties of various kinds such as L1 and L2 regularization and soft weight sharing (Nowlan and Hinton, 1992).

With unlimited computation, the best way to "regularize" a fixed-sized model is to average the predictions of all possible settings of the parameters, weighting each setting by

- Better Learning Regularization (e.g. **Dropout**)

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", JMLR Vol. 15, No. 1,

Working ideas on how to train deep architectures

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe
Google Inc., *sioffe@google.com*

Christian Szegedy
Google Inc., *szegedy@google.com*

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization *for each training mini-batch*. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.9% top-5 validation error (and 4.8% test error), exceeding the accuracy of human raters.

Using mini-batches of examples, as opposed to one example at a time, is helpful in several ways. First, the gradient of the loss over a mini-batch is an estimate of the gradient over the training set, whose quality improves as the batch size increases. Second, computation over a batch can be much more efficient than m computations for individual examples, due to the parallelism afforded by the modern computing platforms.

While stochastic gradient is simple and effective, it requires careful tuning of the model hyper-parameters, specifically the learning rate used in optimization, as well as the initial values for the model parameters. The training is complicated by the fact that the inputs to each layer


Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe
Google Inc., *sioffe@google.com*

Christian Szegedy
Google Inc., *szegedy@google.com*

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization *for each training mini-batch*. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. It also acts as a regularizer, in some cases eliminating the need for Dropout. Applied to a state-of-the-art image classification model, Batch Normalization achieves the same accuracy with 14 times fewer training steps, and beats the original model by a significant margin. Using an ensemble of batch-normalized networks, we improve upon the best published result on ImageNet classification: reaching 4.9% top-5 validation error (and 4.8% test error), exceeding the accuracy of human raters.

Using mini-batches of examples, as opposed to one example at a time, is helpful in several ways. First, the gradient of the loss over a mini-batch is an estimate of the gradient over the training set, whose quality improves as the batch size increases. Second, computation over a batch can be much more efficient than m computations for individual examples, due to the parallelism afforded by the modern computing platforms.

While stochastic gradient is simple and effective, it requires careful tuning of the model hyper-parameters, specifically the learning rate used in optimization, as well as the initial values for the model parameters. The training is complicated by the fact that the inputs to each layer are affected by the parameters of all preceding layers – so that small changes to the network parameters amplify as the network becomes deeper.

The change in the distributions of layers' inputs presents a problem because the layers need to continuously adapt to the new distribution. When the input distribution to a learning system changes, it is said to experience *covariate shift* (Shimodaira, 2000). This is typically handled via domain adaptation (Jiang, 2008). However, the notion of covariate shift can be extended beyond the learning system as a whole, to apply to its parts, such as a sub-network or a layer. Consider a network computing

$$\ell = F_2(F_1(u, \Theta_1), \Theta_2)$$

where F_1 and F_2 are arbitrary transformations, and the parameters Θ_1, Θ_2 are to be learned so as to minimize the loss ℓ. Learning Θ_2 can be viewed as if the inputs $x = F_1(u, \Theta_1)$ are fed into the sub-network

$$\ell = F_2(x, \Theta_2).$$

For example, a gradient descent step

$$\Theta_2 \leftarrow \Theta_2 - \frac{\alpha}{m} \sum_{i=1}^{m} \frac{\partial F_2(x_i, \Theta_2)}{\partial \Theta_2}$$

(for batch size m and learning rate α) is exactly equivalent to that for a stand-alone network F_2 with input x. Therefore, the input distribution properties that make training more efficient – such as having the same distribution between the training and test data – apply to training the sub-network as well. As such it is advantageous for the distribution of x to remain fixed over time. Then, Θ_2 does

1 Introduction

Deep learning has dramatically advanced the state of the art in vision, speech, and many other areas. Stochastic gradient descent (SGD) has proved to be an effective way of training deep networks, and SGD variants such as momentum (Sutskever et al., 2013) and Adagrad (Duchi et al., 2011) have been used to achieve state of the art performance. SGD optimizes the parameters Θ of the network, so as to minimize the loss

$$\Theta = \arg\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \ell(x_i, \Theta)$$

where $x_{1 \ldots N}$ is the training data set. With SGD, the training proceeds in steps, and at each step we consider a *mini-batch* $x_{1 \ldots m}$ of size m. The mini-batch is used to approximate the gradient of the loss function with respect to the parameters, by computing

$$\frac{1}{m} \frac{\partial \ell(x_i, \Theta)}{\partial \Theta}$$

1

- Better Optimization Conditioning (e.g. **Batch Normalization**)

S. Ioffe, C. Szegedy, "**Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift**", In ICML 2015

Working ideas on how to train deep architectures

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error

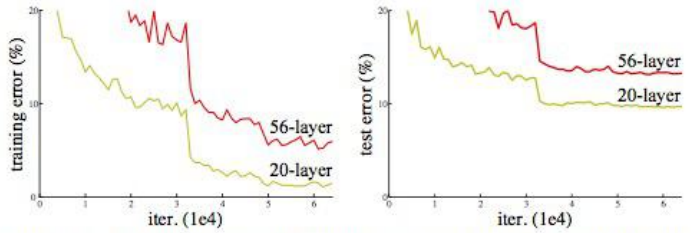
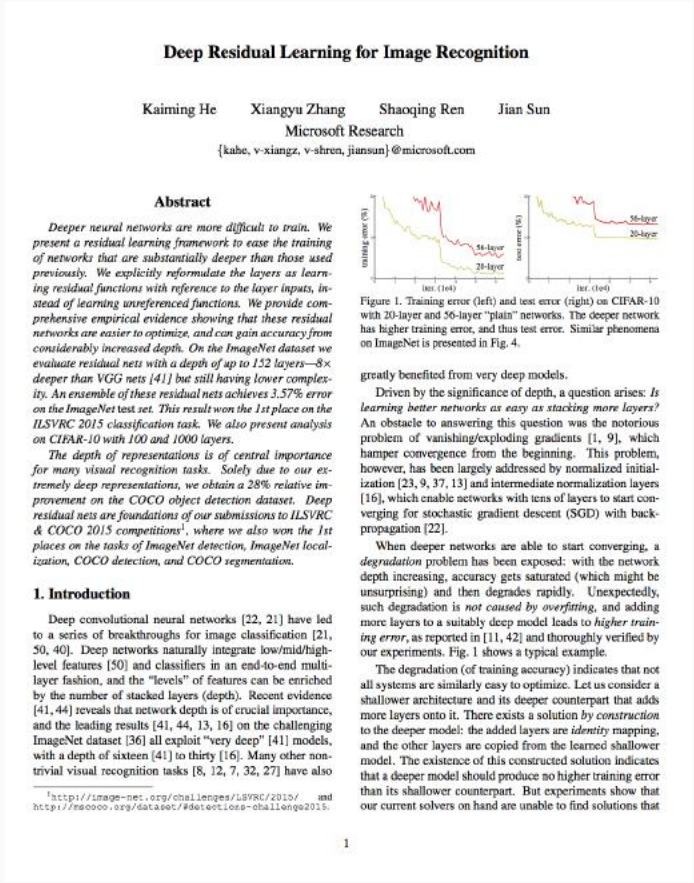


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

Driven by the significance of depth, a question arises: *Is*

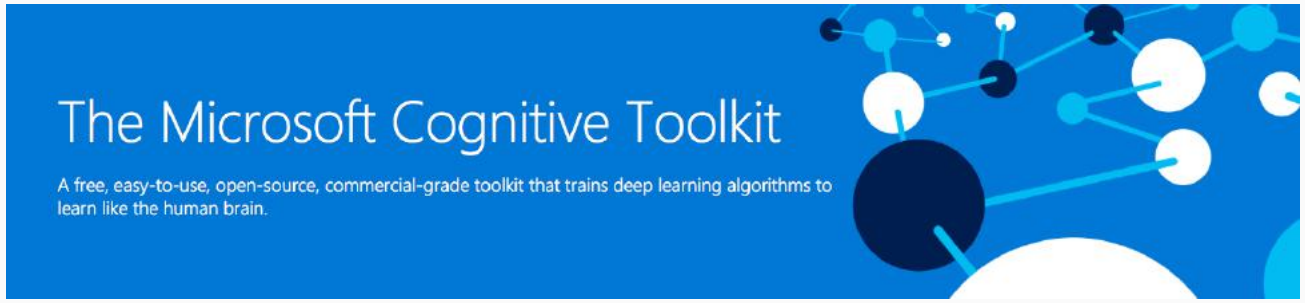


- Better neural achitectures (e.g. **Residual Nets**)

K. He, X. Zhang, S. Ren, J. Sun, "**Deep Residual Learning for Image Recognition**", In CVPR 2016

Software

Caffe



So what is deep learning?

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

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

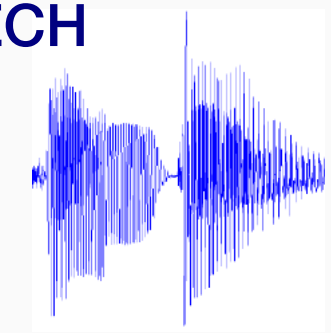
Traditional Machine Learning

VISION

hand-crafted features SIFT/HOG → your favorite classifier → “car”

fixed | learned

SPEECH



hand-crafted features MFCC → your favorite classifier → \'d ē p\

fixed | learned

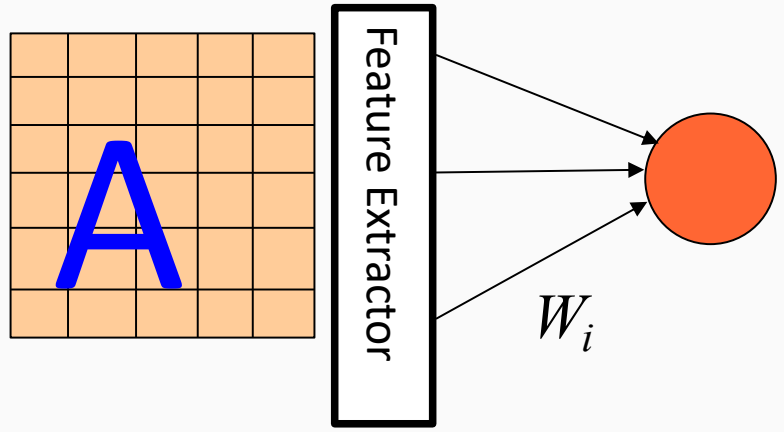
NLP

This burrito place is yummy and fun! → hand-crafted features Bag-of-words → your favorite classifier → “+”

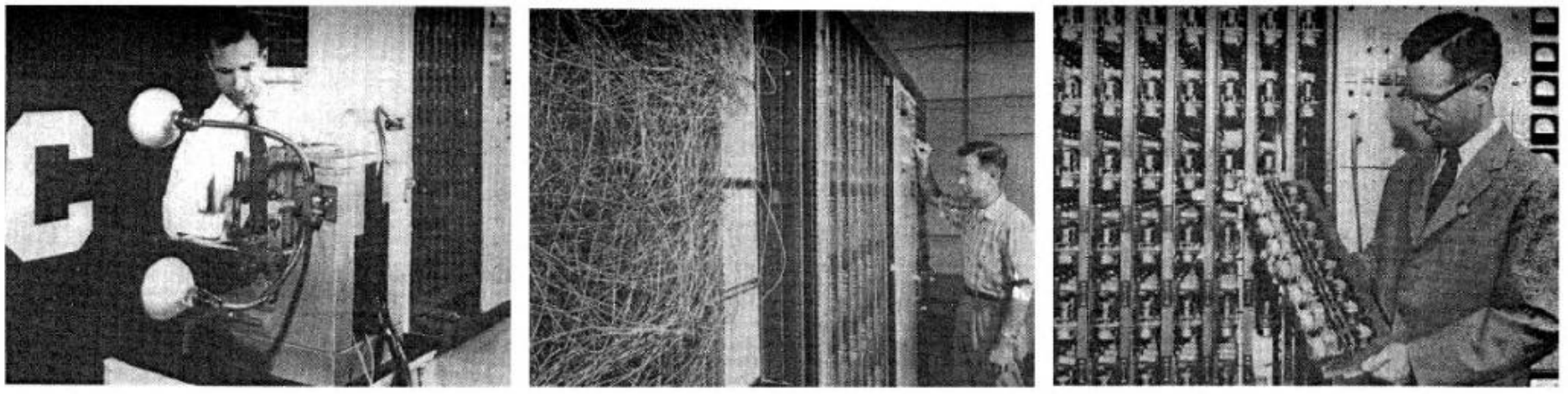
fixed | learned

It's an old paradigm

- The first learning machine: the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching.
- Designing a feature extractor requires considerable efforts by experts.



$$y = sign\left(\sum_i^N W_i F_i(X) + b\right)$$



slide by Marc'Aurelio Ranzato, Yann LeCun

Hierarchical Compositionality

VISION

pixels → edge → texton → motif → part → object

SPEECH

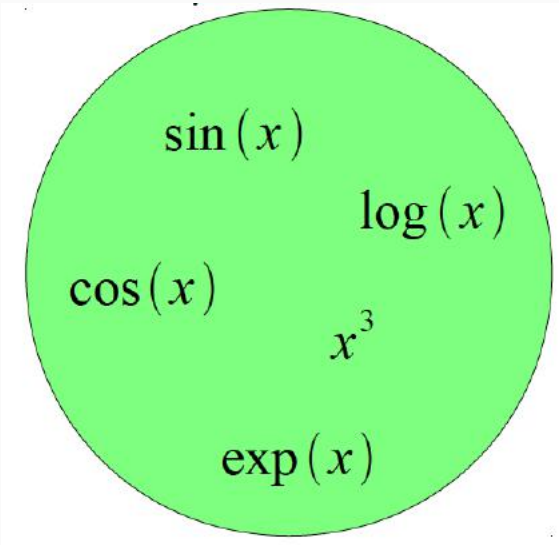
sample → spectral band → formant → motif → phone → word

NLP

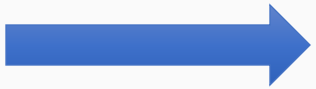
character → word → NP/VP/.. → clause → sentence → story

Building A Complicated Function

Given a library of simple functions



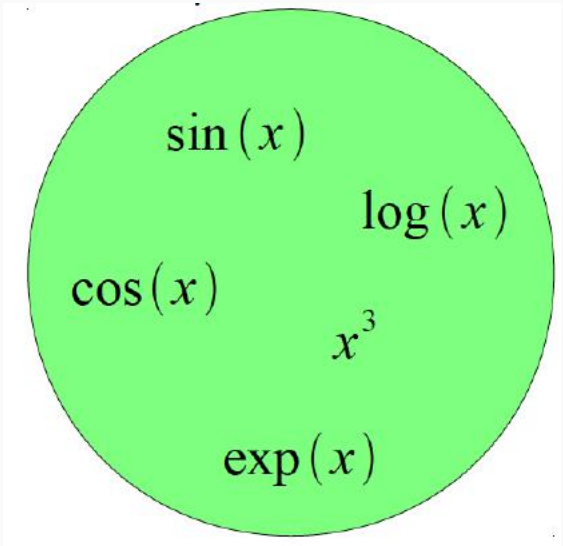
Compose into a



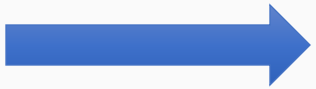
complicate function

Building A Complicated Function

Given a library of simple functions



Compose into a

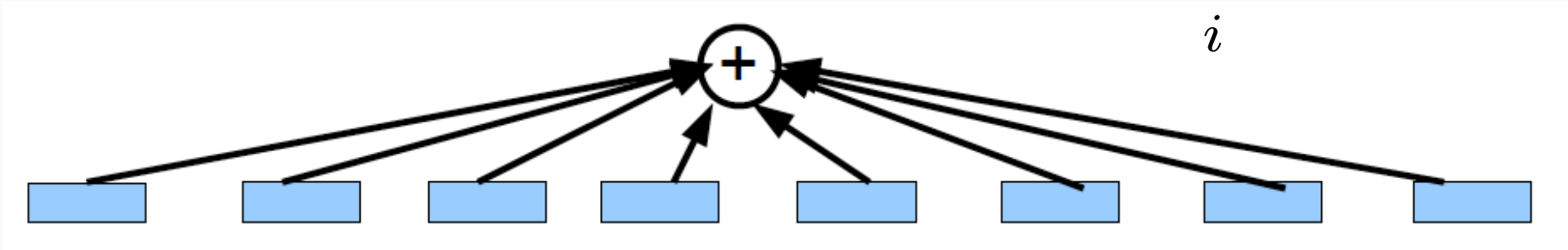


complicate function

Idea 1: Linear Combinations

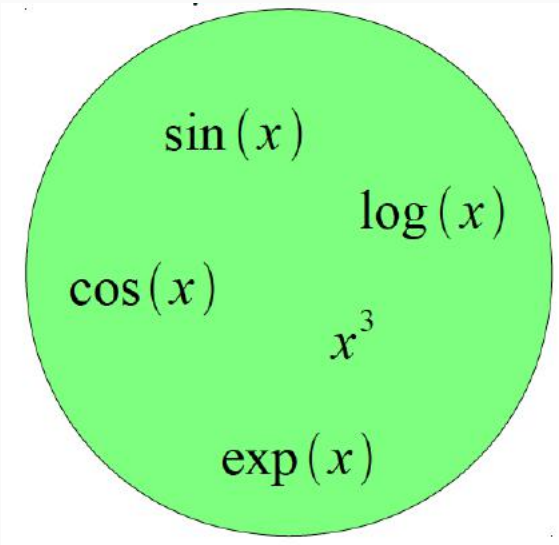
- Boosting
- Kernels
- …

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

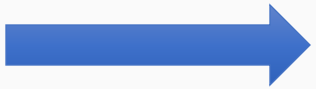


Building A Complicated Function

Given a library of simple functions



Compose into a

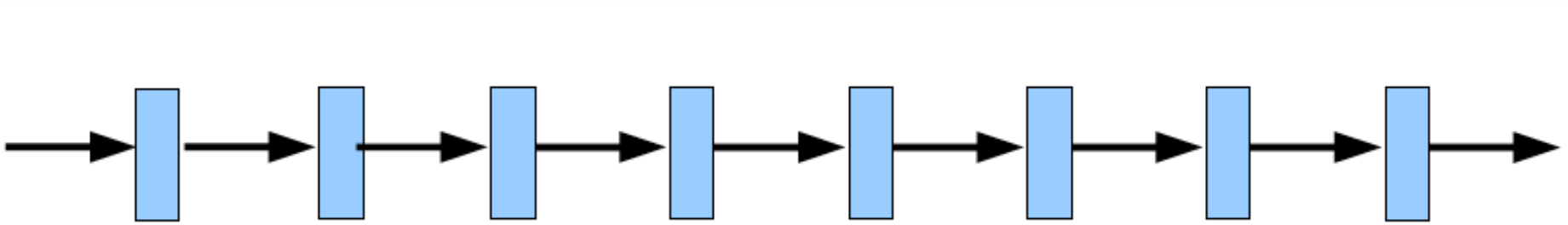


complicate function

Idea 2: Compositions

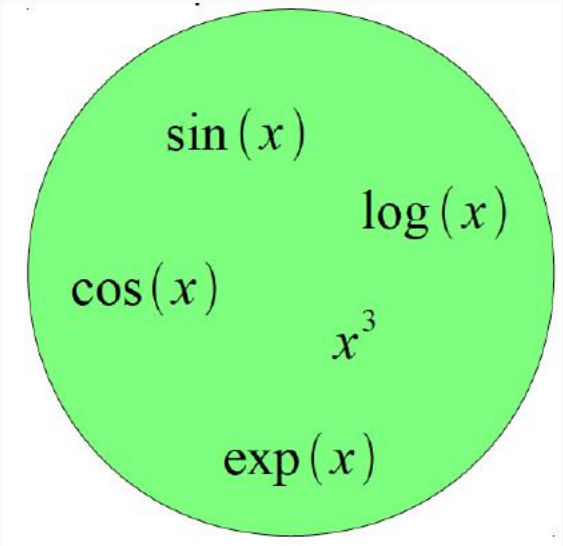
- Deep Learning
- Grammar models
- Scattering transforms…

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

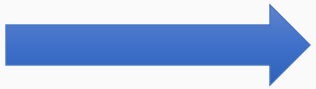


Building A Complicated Function

Given a library of simple functions



Compose into a

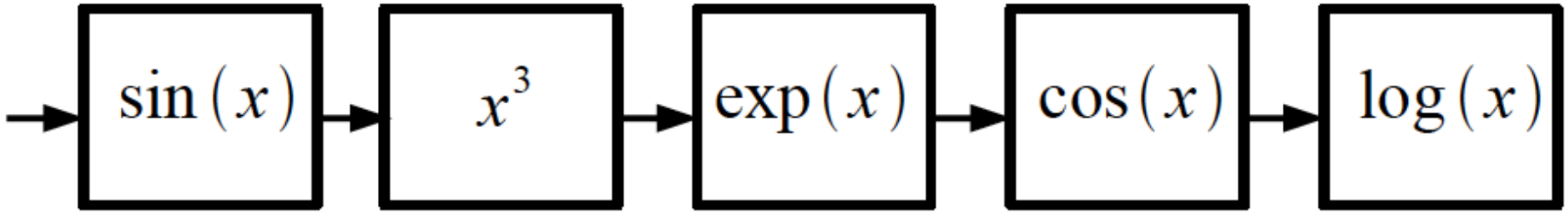


complicate function

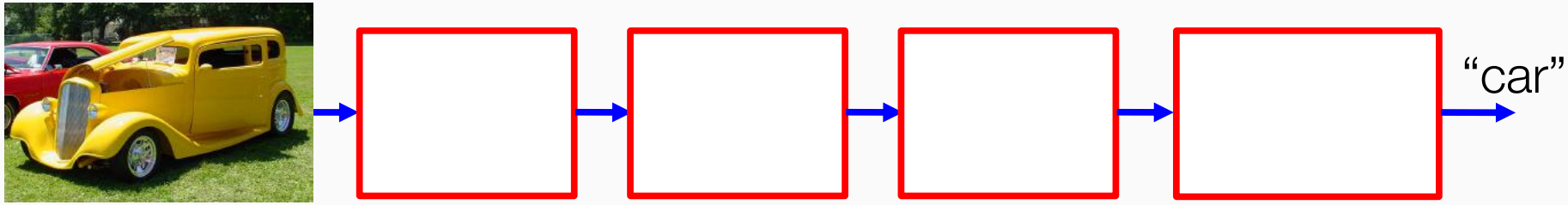
Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms…

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



Deep Learning = Hierarchical Compositionality



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

Deep Learning = Hierarchical Compositionality

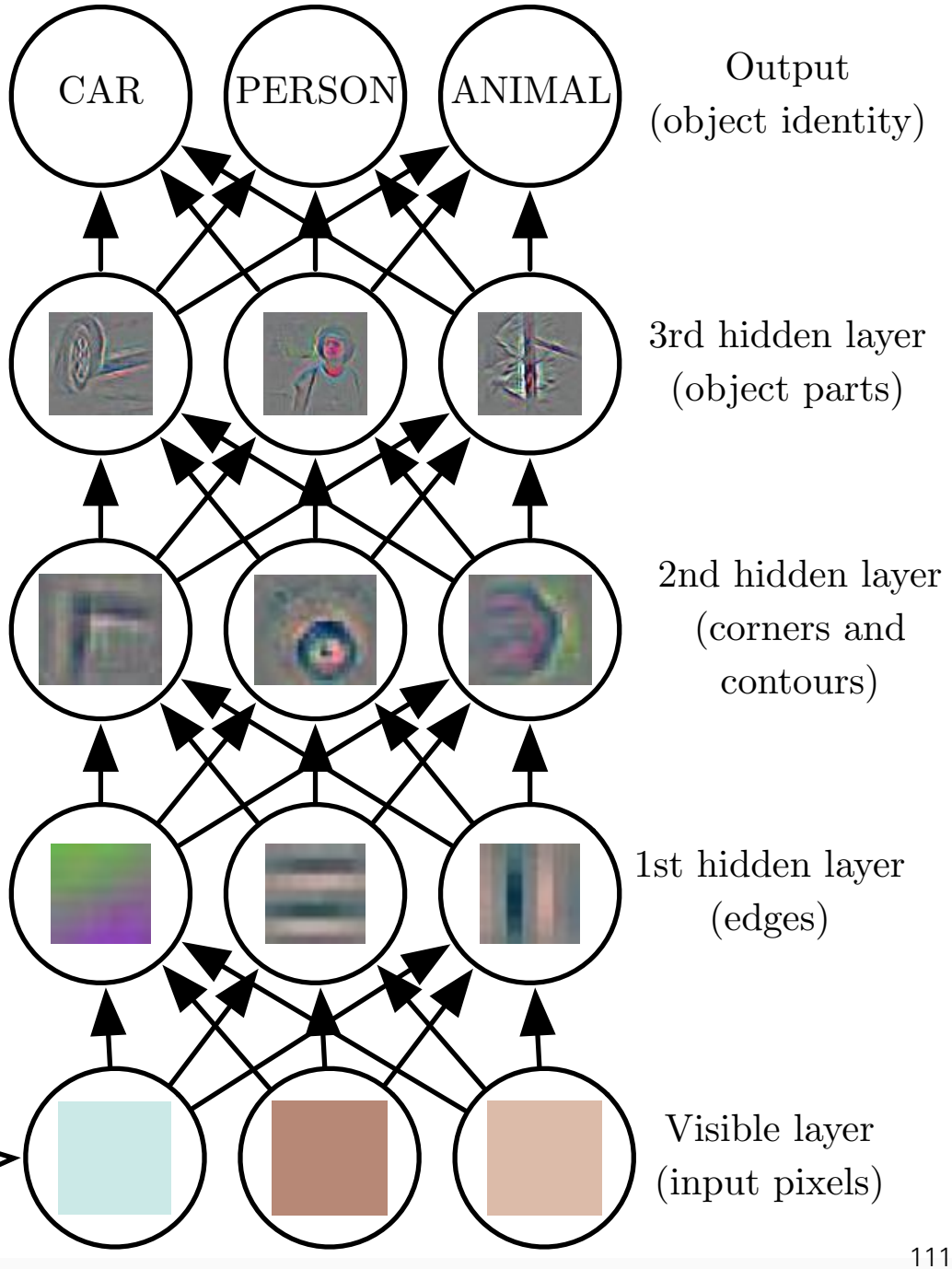
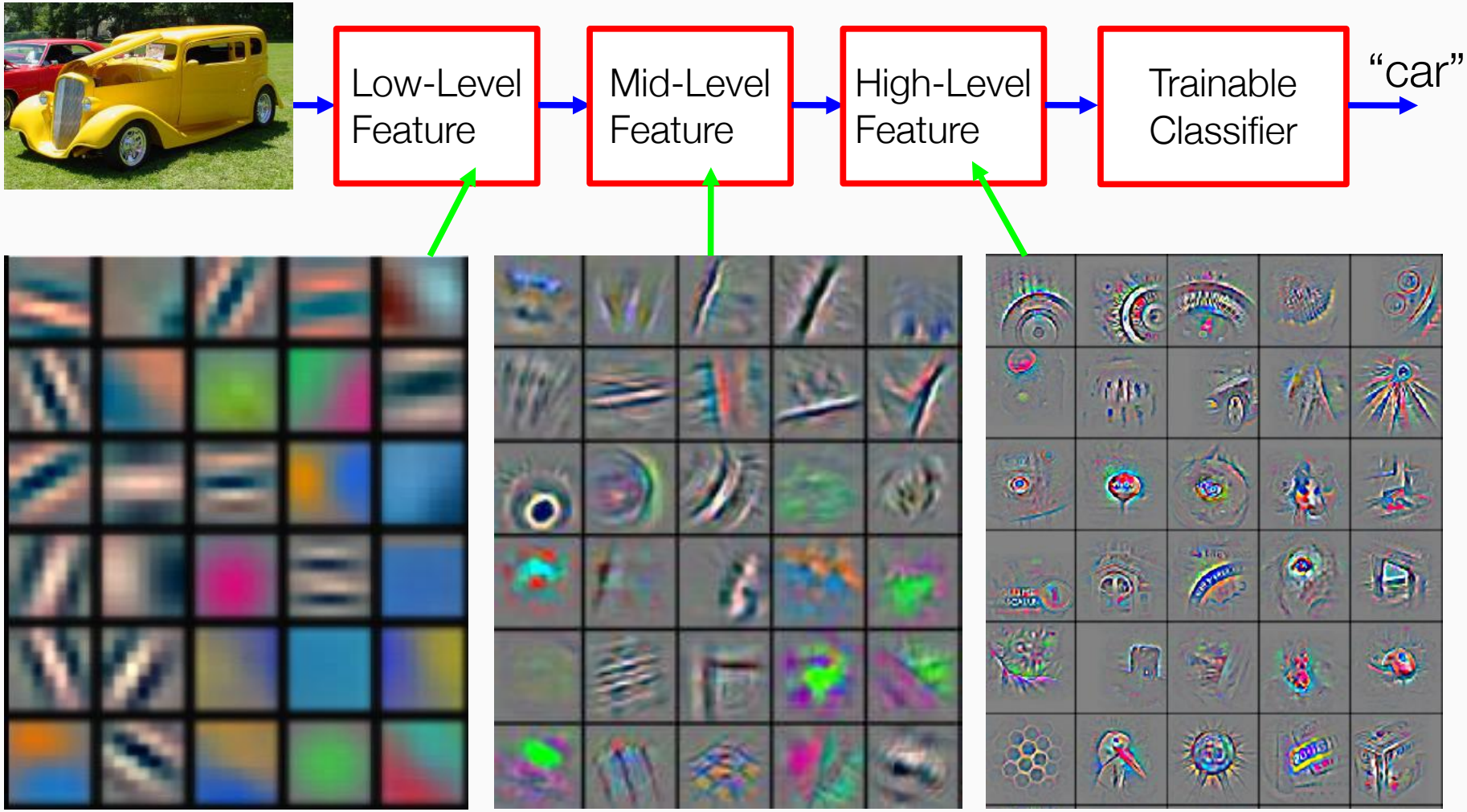


Image credit: Ian Goodfellow

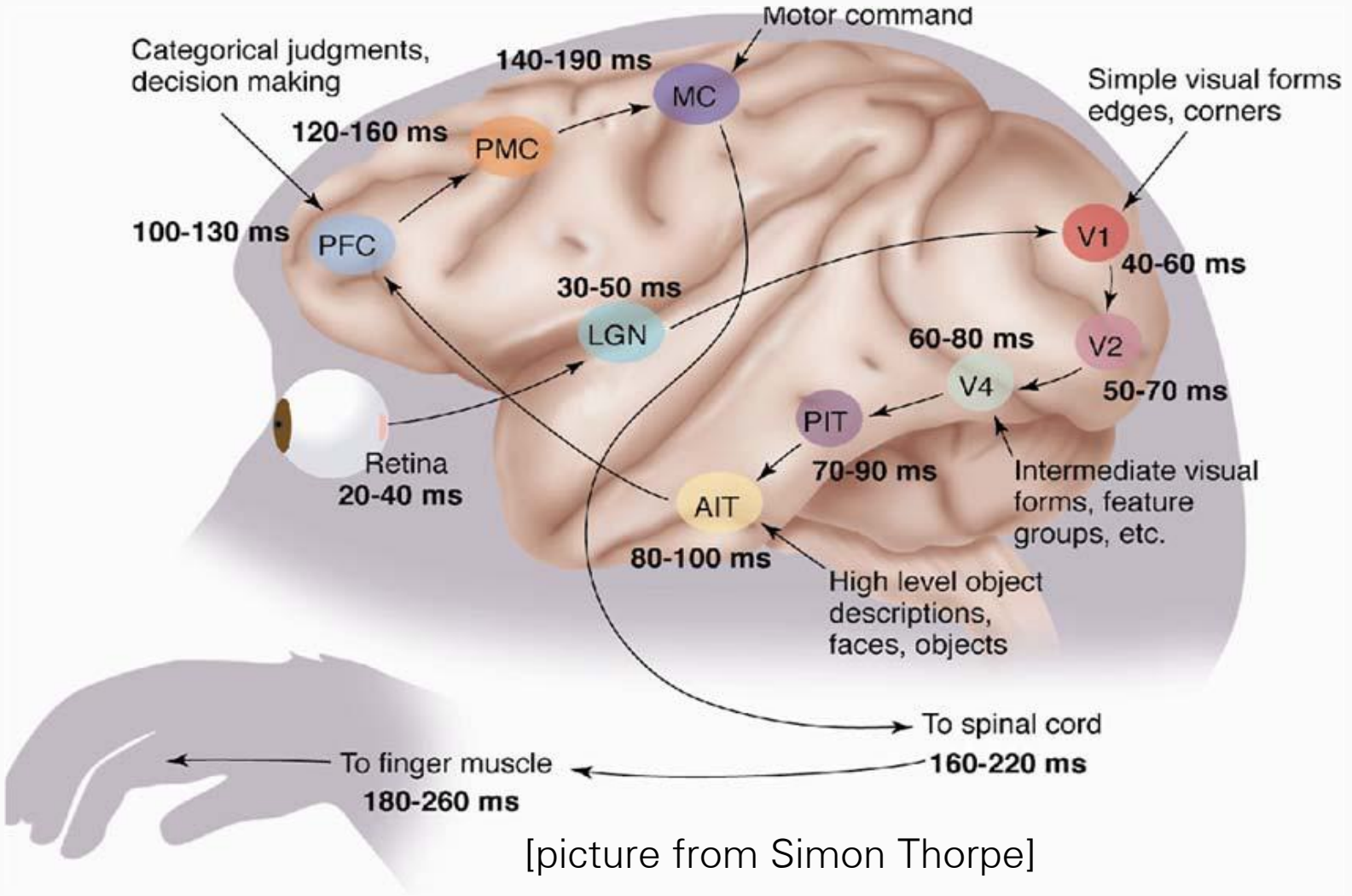
Deep Learning = Hierarchical Compositionality



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

The Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex



[picture from Simon Thorpe]

slide by Marc'Aurelio Ranzato, Yann LeCun

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

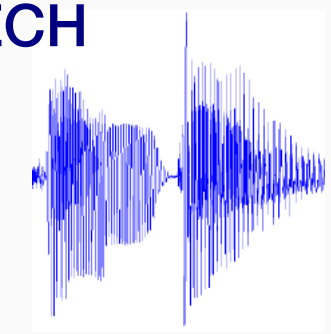
Traditional Machine Learning

VISION

hand-crafted features SIFT/HOG → your favorite classifier → “car”

fixed | learned

SPEECH



hand-crafted features MFCC → your favorite classifier → \'d ē p\

fixed | learned

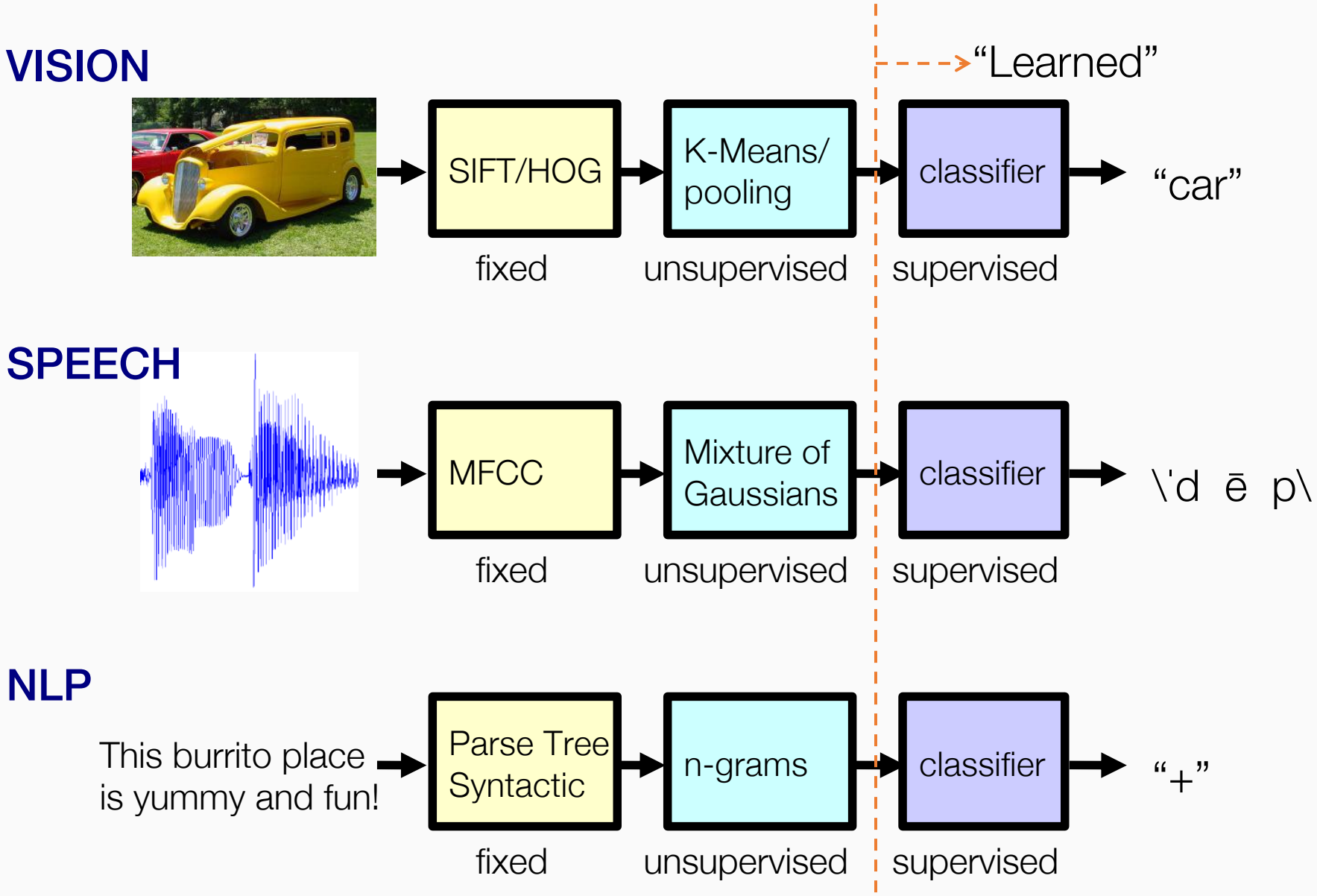
NLP

This burrito place is yummy and fun!

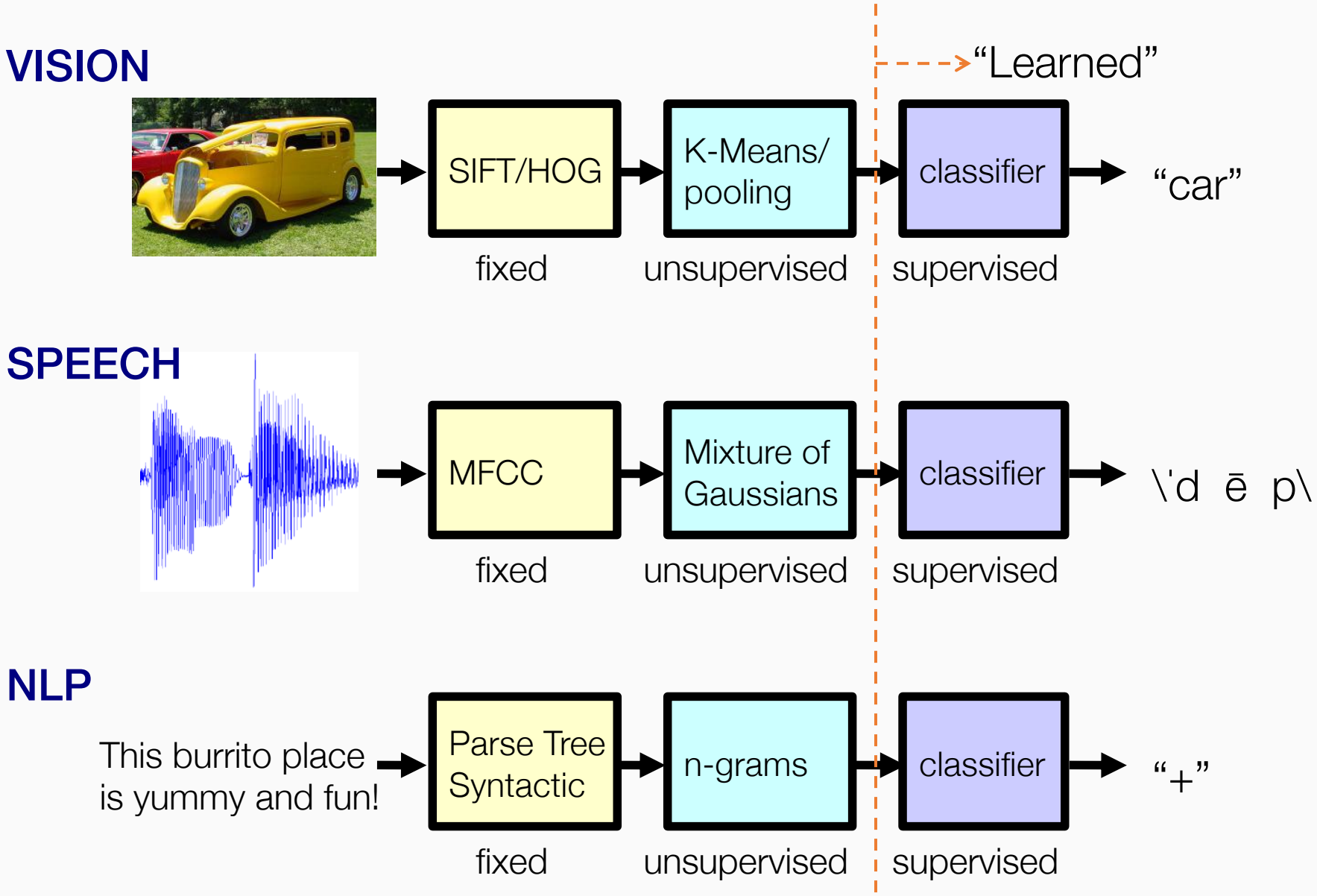
hand-crafted features Bag-of-words → your favorite classifier → “+”

fixed | learned

More accurate version

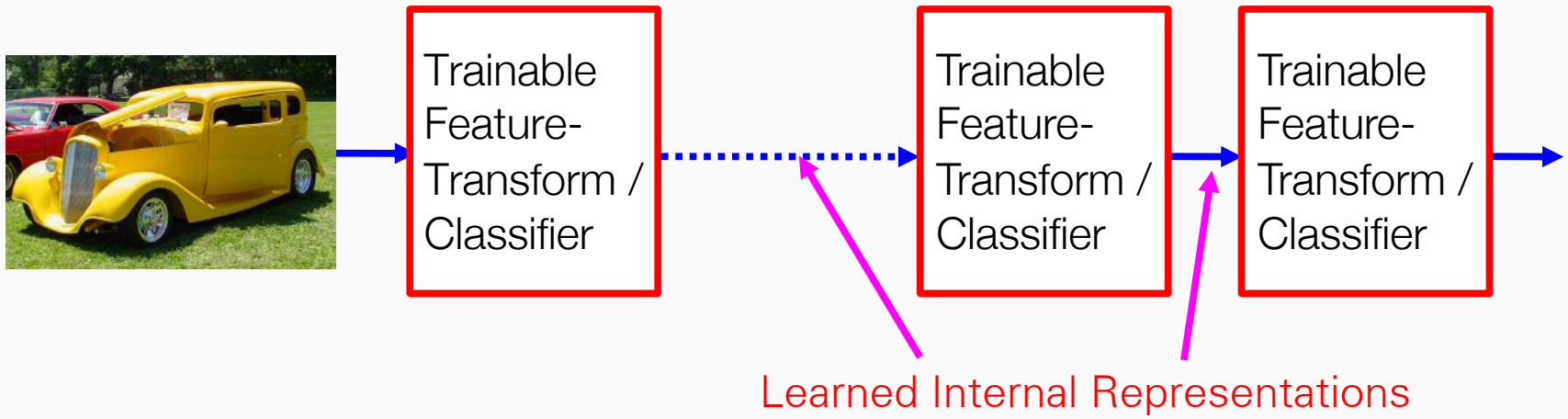


Deep Learning = End-to-End Learning



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



"Shallow" vs Deep Learning

- "Shallow" models

hand-crafted Feature Extractor

fixed

"Simple" Trainable Classifier

learned

- Deep models

Trainable Feature-Transform / Classifier

Trainable Feature-Transform / Classifier

Trainable Feature-Transform / Classifier

Learned Internal Representations

Three key ideas

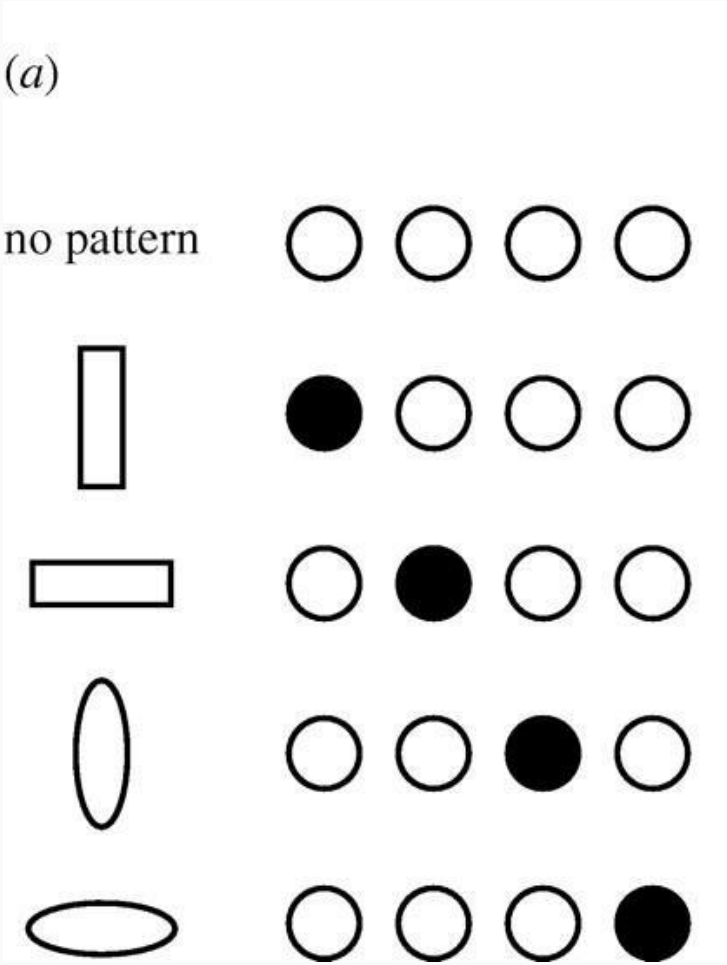
- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations

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

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

Localist representations

- The simplest way to represent things with neural networks is to **dedicate one neuron to each thing**.
  - Easy to understand.
  - Easy to code by hand
    - Often used to represent inputs to a net
  - Easy to learn
    - This is what mixture models do.
    - Each cluster corresponds to one neuron
  - Easy to associate with other representations or responses.
- But localist models are very inefficient whenever the data has componential structure.

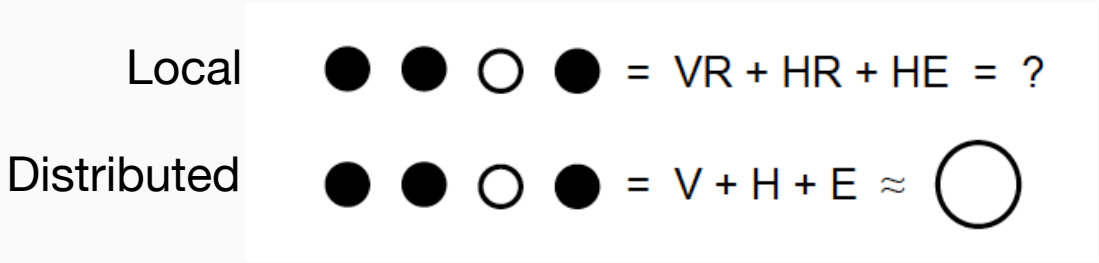


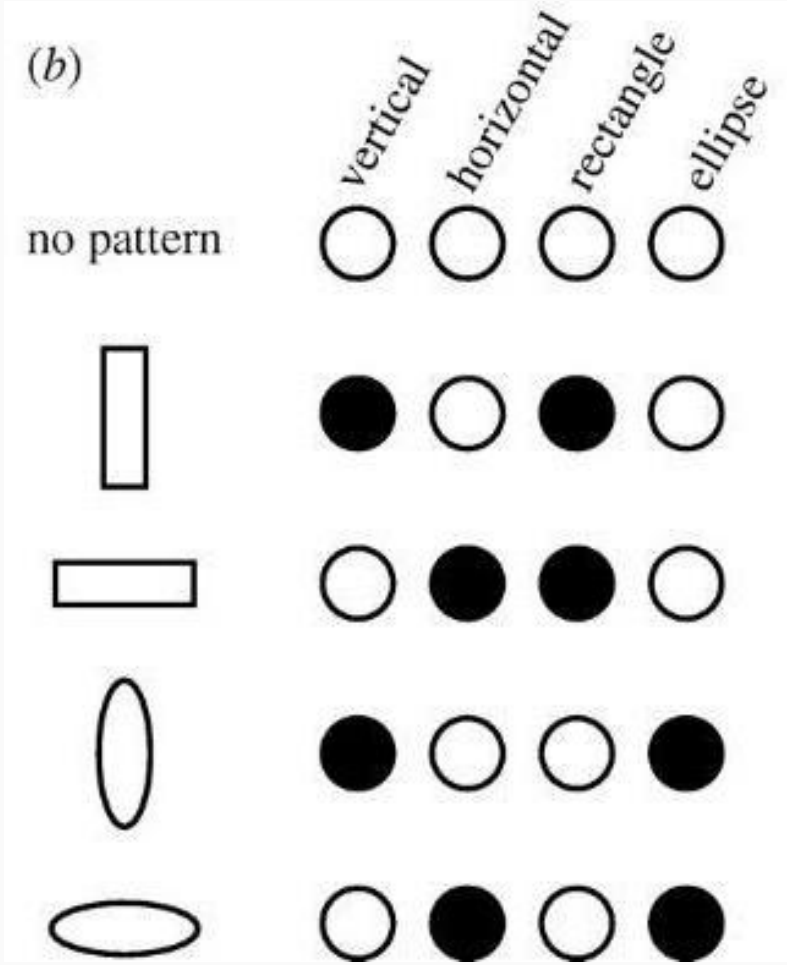
Slide credit: Geoff Hinton

Image credit: Moontae Lee

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

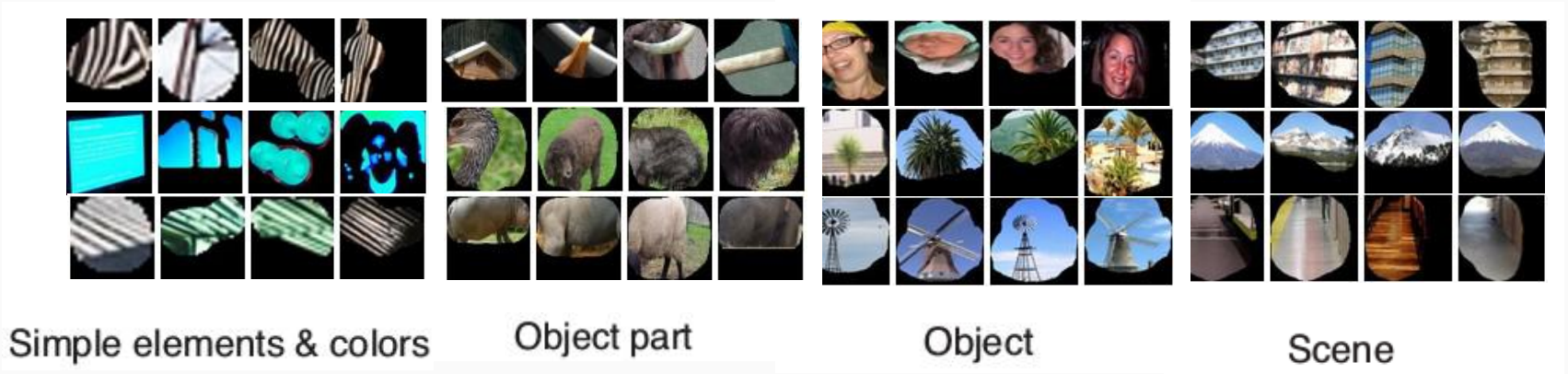
Power of distributed representations!

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

Three key ideas of deep learning

- **(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

Benefits of Deep/Representation Learning

- (Usually) Better Performance
  - "Because gradient descent is better than you"
    Yann LeCun

- New domains without "experts"
  - RGBD
  - Multi-spectral data
  - Gene-expression data
  - Unclear how to hand-engineer

Problems with Deep Learning

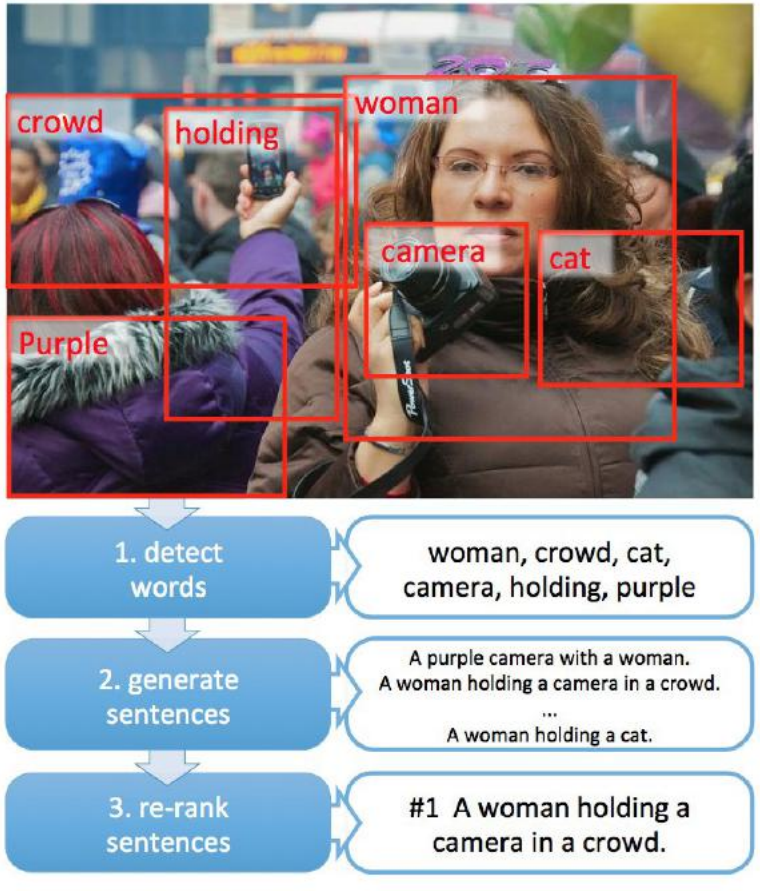
- **Problem#1: Non-Convex! Non-Convex! Non-Convex!**
  - Depth>=3: most losses non-convex in parameters
  - Theoretically, all bets are off
  - Leads to stochasticity
    - different initializations → different local minima
- Standard response #1
  - "Yes, but all interesting learning problems are non-convex"
  - For example, human learning
    - Order matters → wave hands → non-convexity
- Standard response #2
  - "Yes, but it often works!"

Problems with Deep Learning

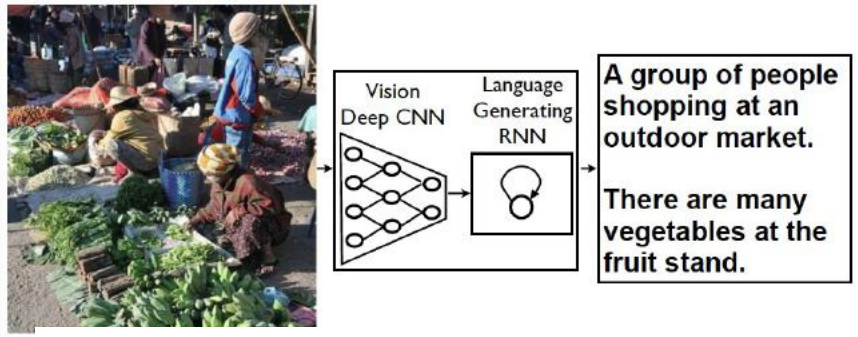
- **Problem#2: Hard to track down what's failing**
  - Pipeline systems have "oracle" performances at each step
  - In end-to-end systems, it's hard to know why things are not working

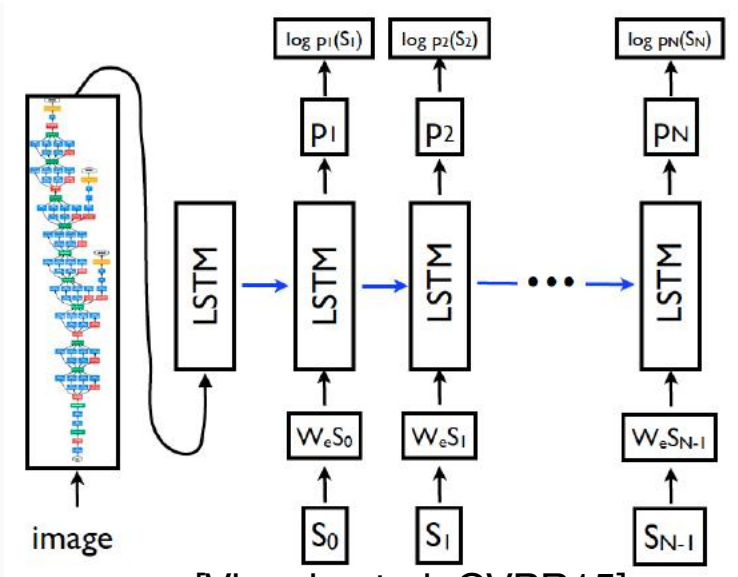
Problems with Deep Learning

- **Problem#2: Hard to track down what's failing**



[Fang et al. CVPR15]





[Vinyals et al. CVPR15]

Pipeline

End-to-End

Problems with Deep Learning

- **Problem#2: Hard to track down what's failing**
  - Pipeline systems have "oracle" performances at each step
  - In end-to-end systems, it's hard to know why things are not working

- Standard response #1
  - Tricks of the trade: visualize features, add losses at different layers, pre-train to avoid degenerate initializations…
  - "We're working on it"

- Standard response #2
  - "Yes, but it often works!"

Problems with Deep Learning

- **Problem#3: Lack of easy reproducibility**
  - Direct consequence of stochasticity & non-convexity

- Standard response #1
  - It's getting much better
  - Standard toolkits/libraries/frameworks now available

- Standard response #2
  - "Yes, but it often works!"

1958 New York Times…

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

COMPUTER SCIENTISTS STYMIED IN THEIR QUEST TO MATCH HUMAN VISION

By WILLIAM J. BROAD
Published: September 25, 1984

EXPERTS pursuing one of man's most audacious dreams - to create machines that think - have stumbled while taking what seemed to be an elementary first step. They have failed to master vision.

After two decades of research, they have yet to teach machines the seemingly simple act of being able to recognize everyday objects and to distinguish one from another.

Instead, they have developed a profound new respect for the sophistication of human sight and have scoured such fields as mathematics, physics, biology and psychology for clues to help them achieve the goal of machine vision.

SCIENCE

Researchers Announce Advance in Image-Recognition Software

By JOHN MARKOFF NOV. 17, 2014

Email

Share

Tweet

Save

More

MOUNTAIN VIEW, Calif. — Two groups of scientists, working independently, have created artificial intelligence software capable of recognizing and describing the content of photographs and videos with far greater accuracy than ever before, sometimes even mimicking human levels of understanding.

Until now, so-called computer vision has largely been limited to recognizing individual objects. The new software, described on Monday by researchers at Google and at Stanford University, teaches itself to identify entire scenes: a group of young men playing Frisbee, for example, or a herd of elephants marching on a grassy plain.

The software then writes a caption in English describing the picture. Compared with human observations, the researchers found, the computer-written descriptions are surprisingly accurate.

Captioned by Human and by Google's Experimental Program

Human: "A group of men playing Frisbee in the park."
Computer model: "A group of young people playing a game of Frisbee."

TWEETS 587 FOLLOWING 18 FOLLOWERS 746 FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · 10h

a man holding a mirror up to his face .

1 View more photos and videos

Results from @INTERESTING_JPG via http://deeplearning.cs.toronto.edu/i2t

TWEETS 587 | FOLLOWING 18 | FOLLOWERS 746 | FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · 18h

a man carrying a bucket of his hands in a yard .

2

View more photos and videos

Results from @INTERESTING_JPG via http://deeplearning.cs.toronto.edu/i2t

TWEETS 587 FOLLOWING 18 FOLLOWERS 746 FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · Feb 20

a surfboard attached to the top of a car .

8 8 View more photos and videos

Results from @INTERESTING_JPG via http://deeplearning.cs.toronto.edu/i2t

TWEETS 587 | FOLLOWING 18 | FOLLOWERS 746 | FAVORITES 13

INTERESTING.JPG @INTERESTING_JPG · Feb 19

a man dressed in uniform is looking at his cell phone .

2

View more photos and videos

Results from @INTERESTING_JPG via http://deeplearning.cs.toronto.edu/i2t

TWEETS 587 | FOLLOWING 18 | FOLLOWERS 746 | FAVORITES 13

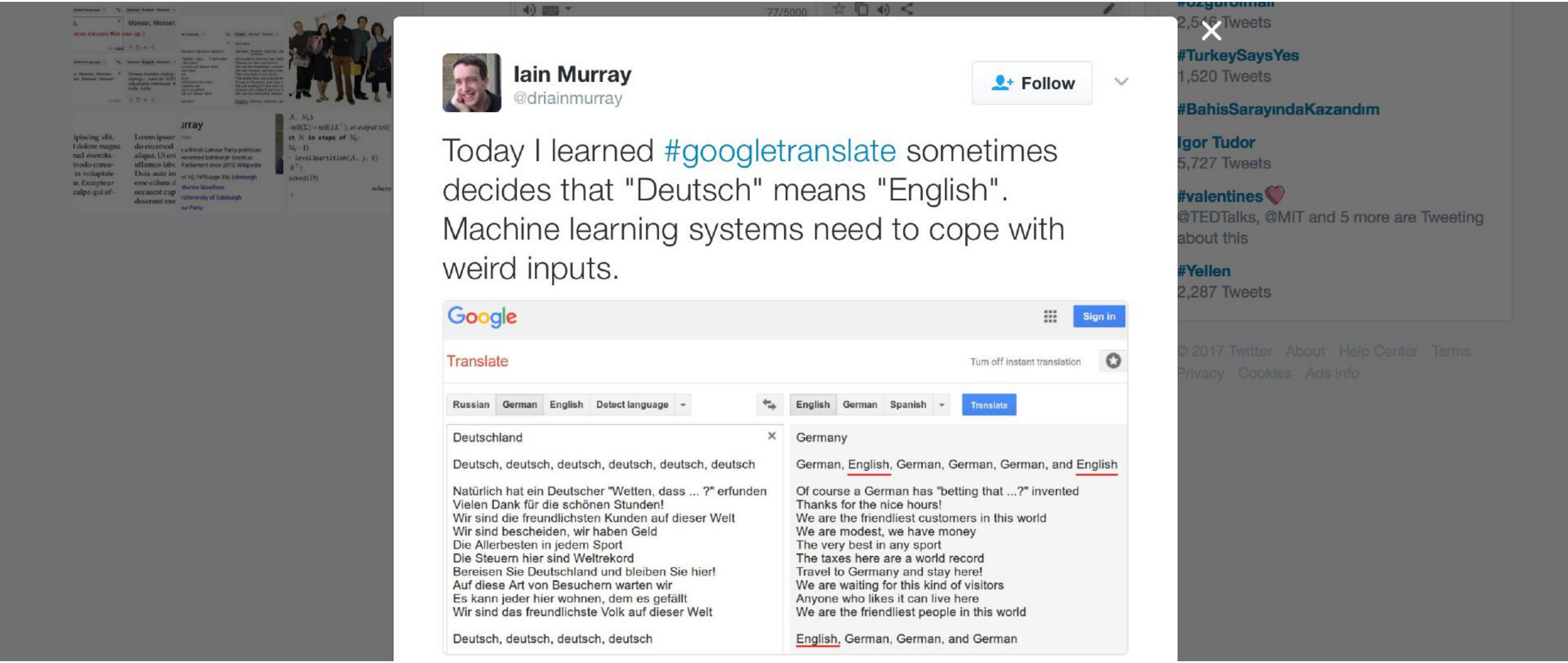
INTERESTING.JPG @INTERESTING_JPG · 16h

this appears to be a small bedroom in the snow .

6

View more photos and videos

Results from @INTERESTING_JPG via http://deeplearning.cs.toronto.edu/i2t


Iain Murray
@driainmurray
Follow
Today I learned #googletranslate sometimes decides that "Deutsch" means "English". Machine learning systems need to cope with weird inputs.
Google
Sign in
Translate
Turn off instant translation
Russian German English Detect language
English German Spanish
Translate
Deutschland
Deutsch, deutsch, deutsch, deutsch, deutsch, deutsch
Natürlich hat ein Deutscher "Wetten, dass ... ?" erfunden
Vielen Dank für die schönen Stunden!
Wir sind die freundlichsten Kunden auf dieser Welt
Wir sind bescheiden, wir haben Geld
Die Allerbesten in jedem Sport
Die Steuern hier sind Weltrekord
Bereisen Sie Deutschland und bleiben Sie hier!
Auf diese Art von Besuchern warten wir
Es kann jeder hier wohnen, dem es gefällt
Wir sind das freundlichste Volk auf dieser Welt
Deutsch, deutsch, deutsch, deutsch
Germany
German, English, German, German, German, and English
Of course a German has "betting that ...?" invented
Thanks for the nice hours!
We are the friendliest customers in this world
We are modest, we have money
The very best in any sport
The taxes here are a world record
Travel to Germany and stay here!
We are waiting for this kind of visitors
Anyone who likes it can live here
We are the friendliest people in this world
English, German, German, and German
#TurkeySaysYes
1,520 Tweets
#BahisSarayındaKazandım
Igor Tudor
5,727 Tweets
#valentines
@TEDTalks, @MIT and 5 more are Tweeting about this
#Yellen
2,287 Tweets
© 2017 Twitter About Help Center Terms
Privacy Cookies Ads info

Iain Murray
@driainmurray

Academic in Machine Learning and Statistics.

homepages.inf.ed.ac.uk/imurray2/

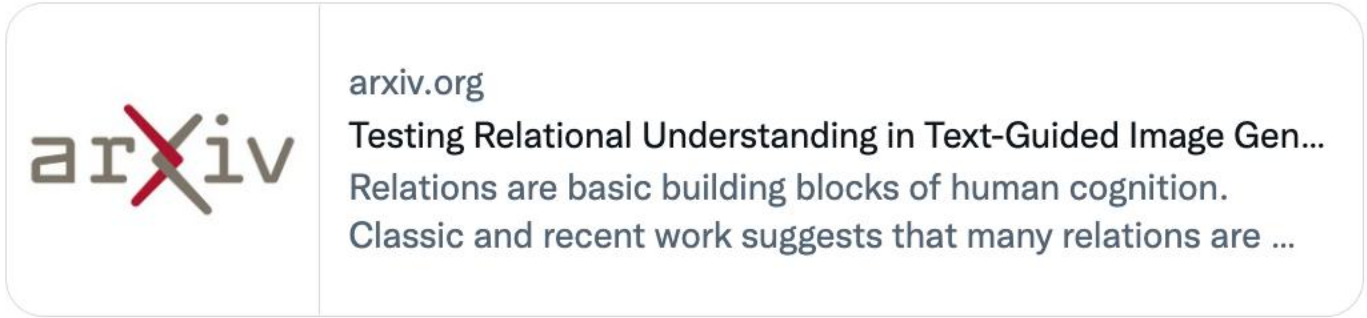
Joined May 2011

Tomer Ullman
@TomerUllman

Do models like DALL-E 2 get basic relations (in/on/etc)?

Colin (Coco) Conwell and I set out to investigate. The result is now on arXiv:

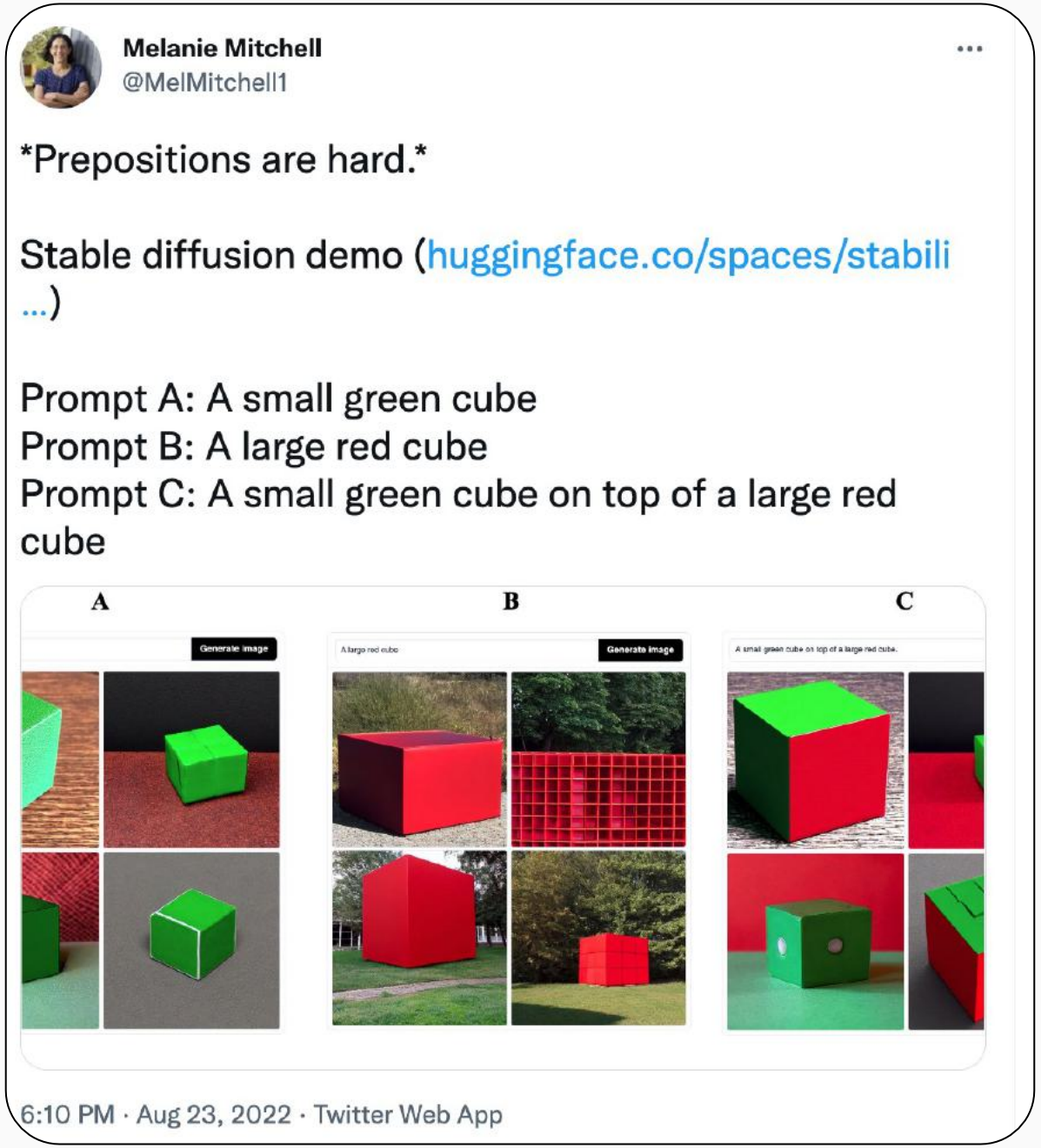
“Testing Relational Understanding in Text-Guided Image Generation”

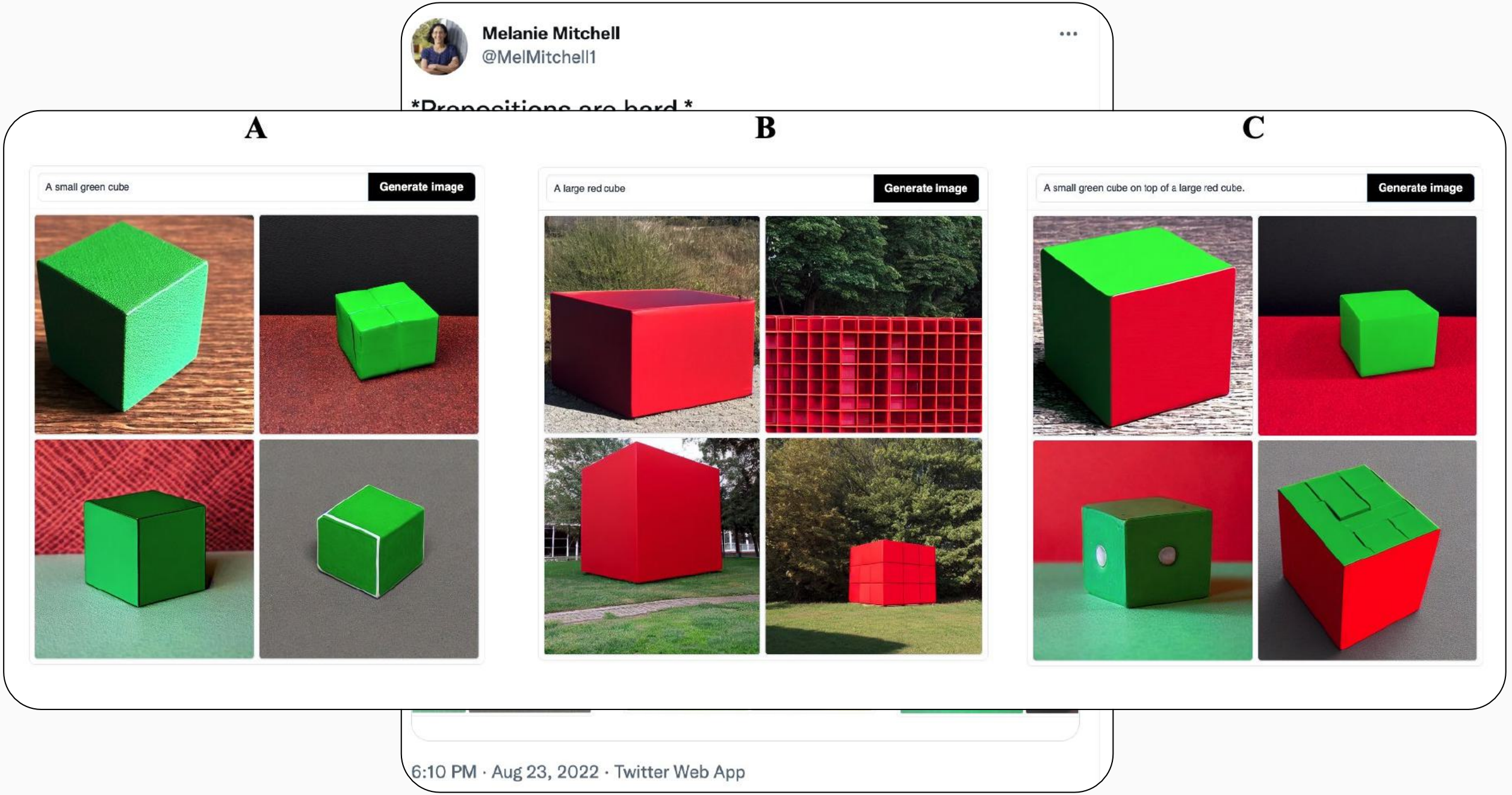


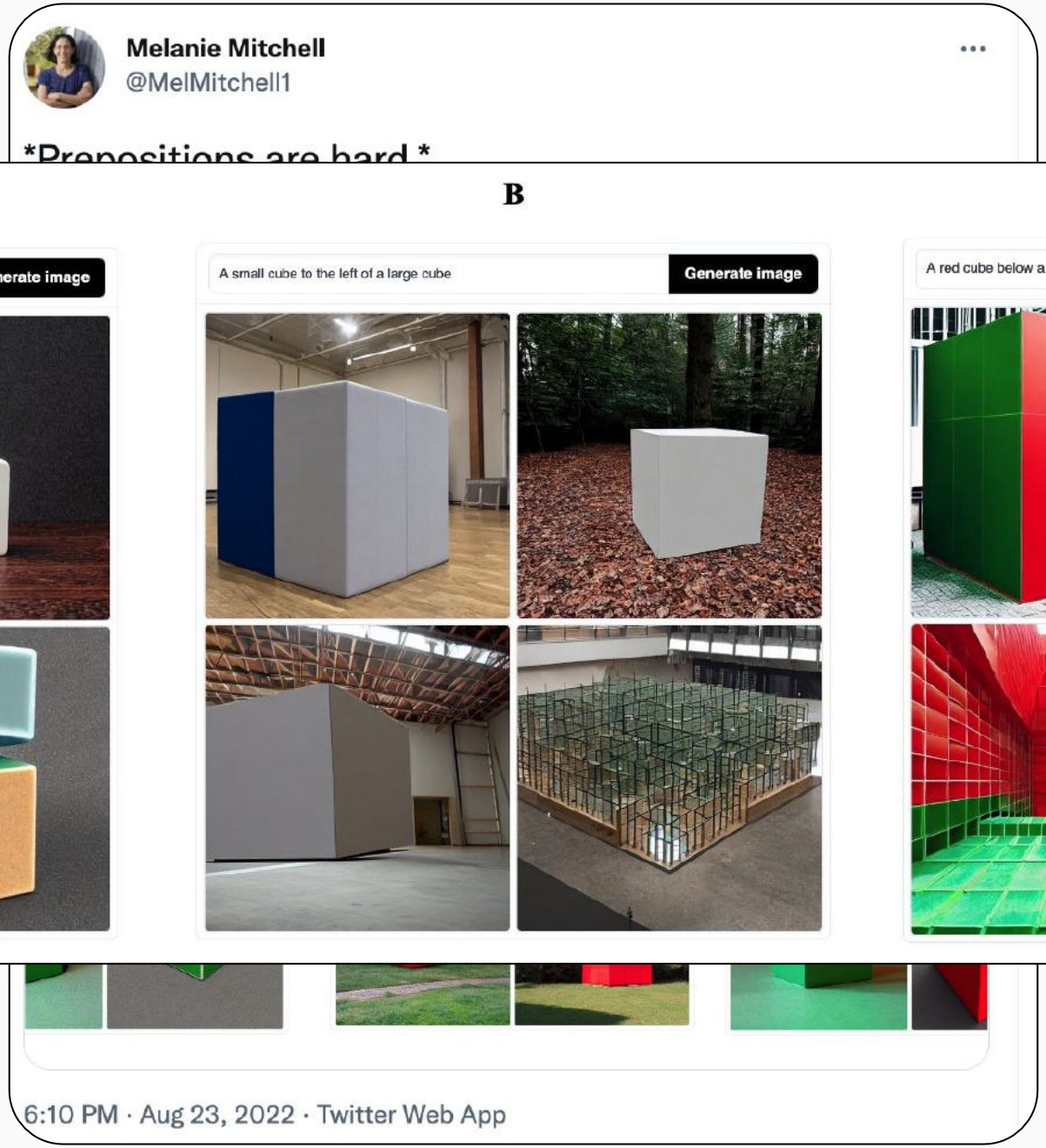
2:55 PM · Aug 2, 2022 · Twitter Web App

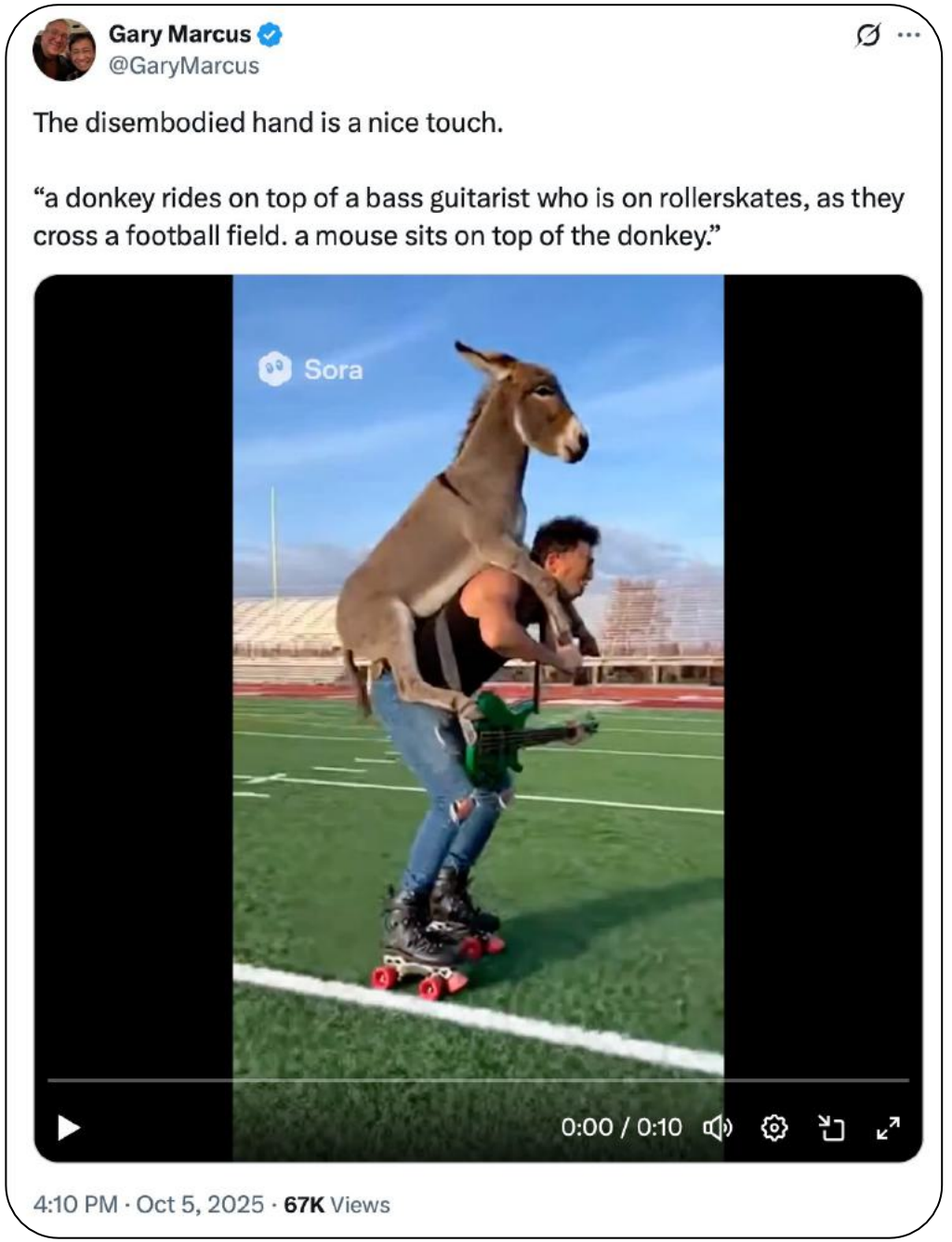
“A spoon in a cup”

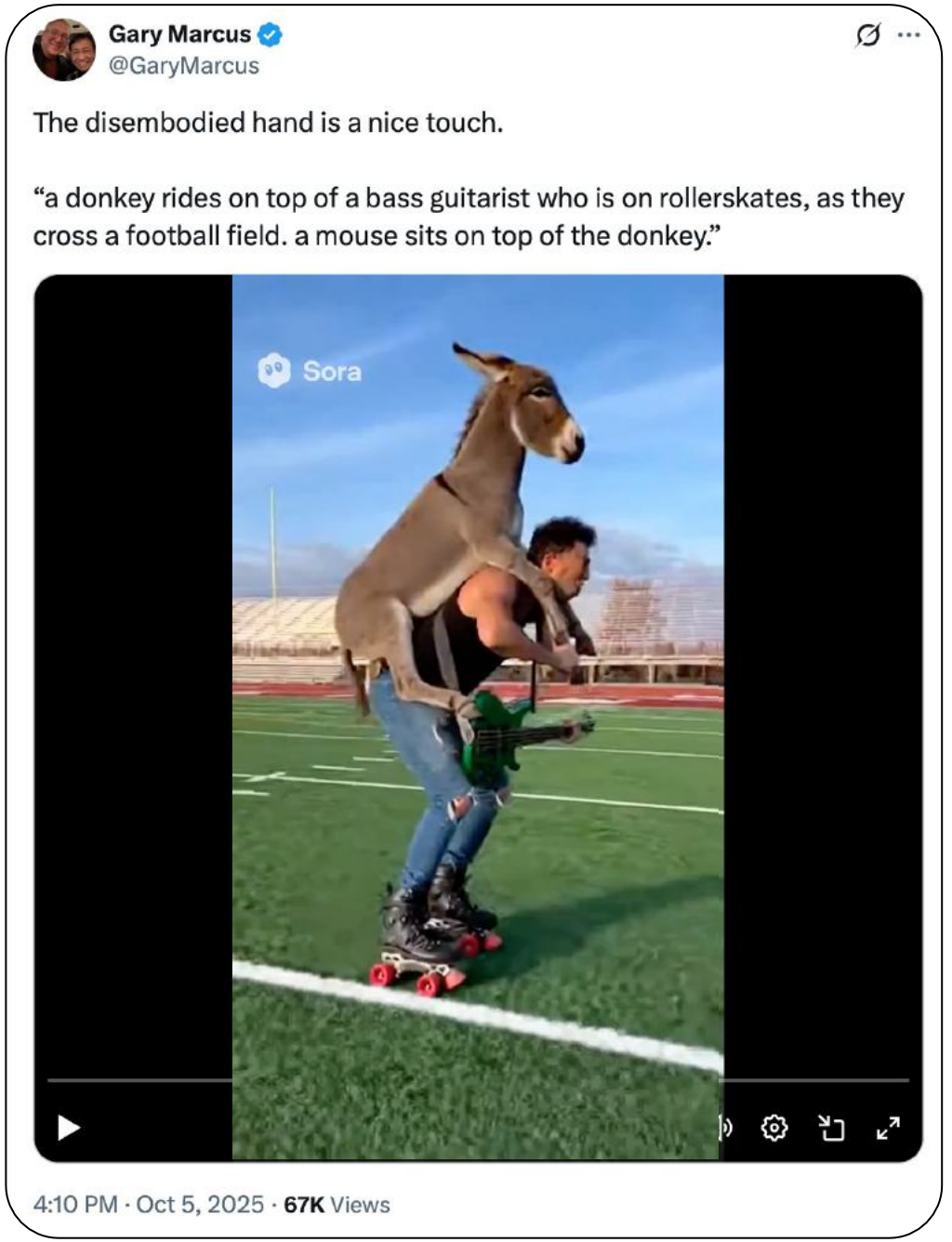
“A cup on a spoon”

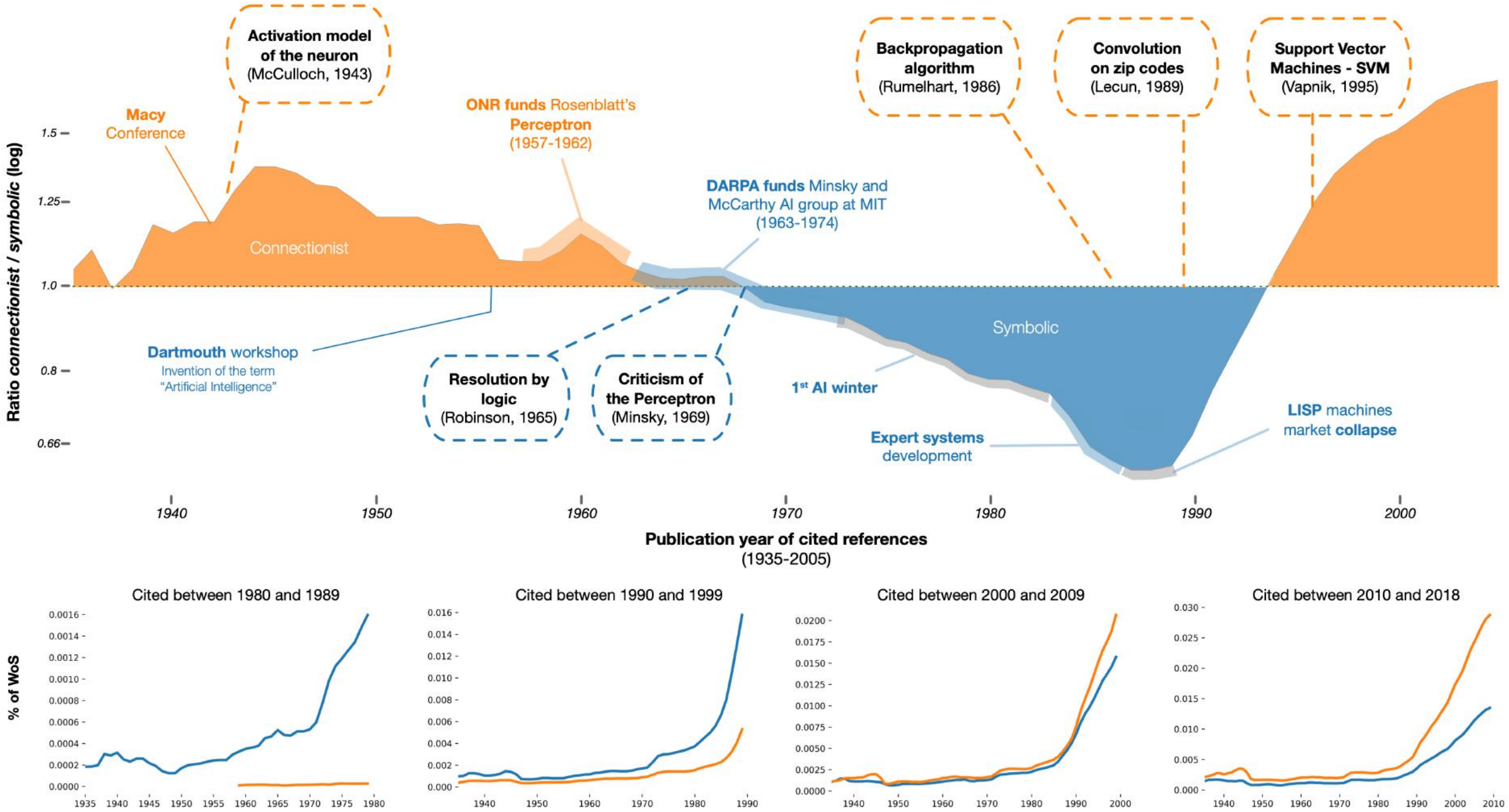

Melanie Mitchell
@MelMitchell1
Prepositions are hard.
Stable diffusion demo (huggingface.co/spaces/stabili
...)
Prompt A: A small green cube
Prompt B: A large red cube
Prompt C: A small green cube on top of a large red cube
A
B
C
Generate image
A large red cube
Generate image
A small green cube on top of a large red cube.
6:10 PM · Aug 23, 2022 · Twitter Web App


Melanie Mitchell
@MelMitchell1
*Propositions are hard *
A
B
C
A small green cube
Generate image
A large red cube
Generate image
A small green cube on top of a large red cube.
Generate image
6:10 PM · Aug 23, 2022 · Twitter Web App


Melanie Mitchell
@MelMitchell1
*Prepositions are hard *
A
One cube on top of another cube
Generate image
B
A small cube to the left of a large cube
Generate image
C
A red cube below a green cube
Generate image
6:10 PM · Aug 23, 2022 · Twitter Web App


Gary Marcus
@GaryMarcus
The disembodied hand is a nice touch.
“a donkey rides on top of a bass guitarist who is on rollerskates, as they cross a football field. a mouse sits on top of the donkey.”
Sora
0:00 / 0:10
4:10 PM · Oct 5, 2025 · 67K Views


Gary Marcus
@GaryMarcus
The disembodied hand is a nice touch.
“a donkey rides on top of a bass guitarist who is on rollerskates, as they cross a football field. a mouse sits on top of the donkey.”
Sora
4:10 PM · Oct 5, 2025 · 67K Views



D. Cardon et al. "**Neurons spike back: The Invention of Inductive Machines and the AI Controversy**", Réseaux n°211/2018

Next Lecture:
Machine Learning Overview