

# COMP541

## DEEP LEARNING

Lecture #01 – Introduction



Aykut Erdem // Koç University // Fall 2022

# Welcome to COMP541

- This course gives an overview of **deep learning**,
- In particular, we will cover various **deep architectures** and **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



Università 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



Virginia Tech  
Visiting Research Scholar  
Summer 2006



- I explore better ways to understand, interpret and manipulate visual data.
- My research interests span a diverse set of topics, ranging from image editing to visual saliency estimation, and to multimodal learning for integrated vision and language.



# Now, what about you?

The screenshot shows a Google Forms survey titled "COMP441/541 Fall 2022 Survey". The survey includes fields for Name, E-mail Address, and Status. The "Name" field is currently empty. The "E-mail Address" field is also empty. The "Status" field has a single option selected: "PhD: 1st year".

docs.google.com

### COMP441/541 Fall 2022 Survey

aerdem@ku.edu.tr (not shared) [Switch account](#)

\* Required

Name \*

Your answer

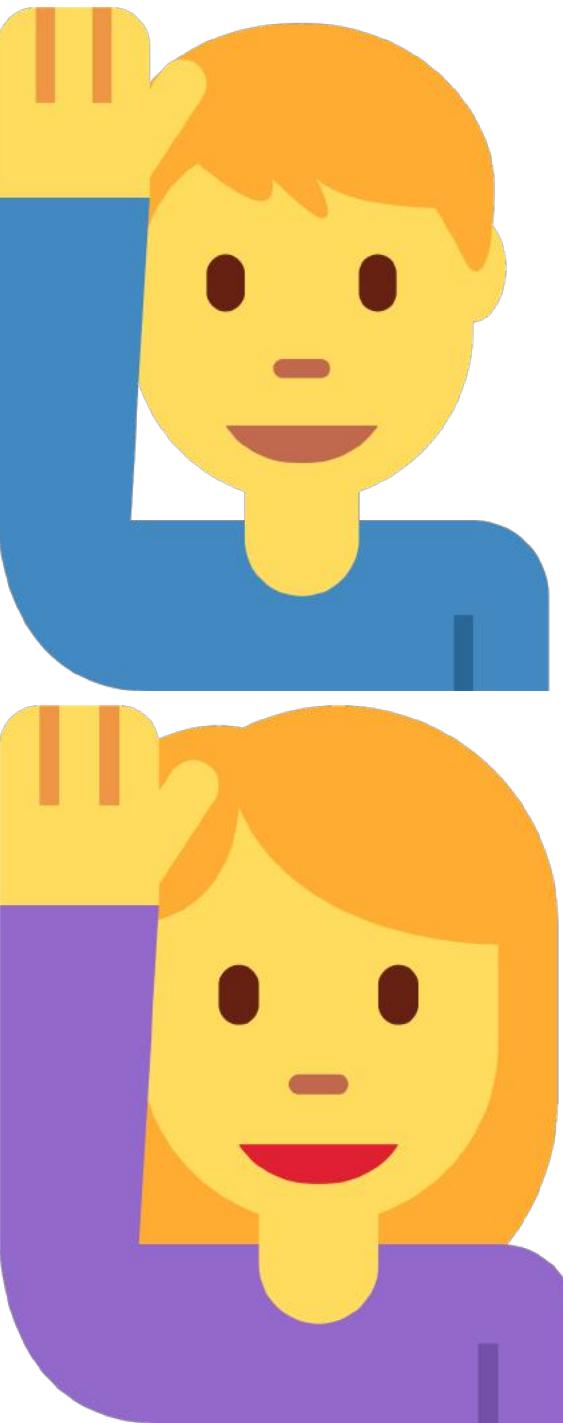
E-mail Address: \*

Your answer

Status \*

PhD: 1st year

<https://forms.gle/2XpzYb6wMurbBh7P9>



# Course Logistics

# Course Information

**Lectures** Monday and Wednesday 11:30-12:40 (Tower Second Floor)

**PS** Monday 17:30-18:40 Friday (SCI Z24)

**Instructor** Aykut Erdem

**TAs** Ilker Kesen and Emre Can Acikgoz

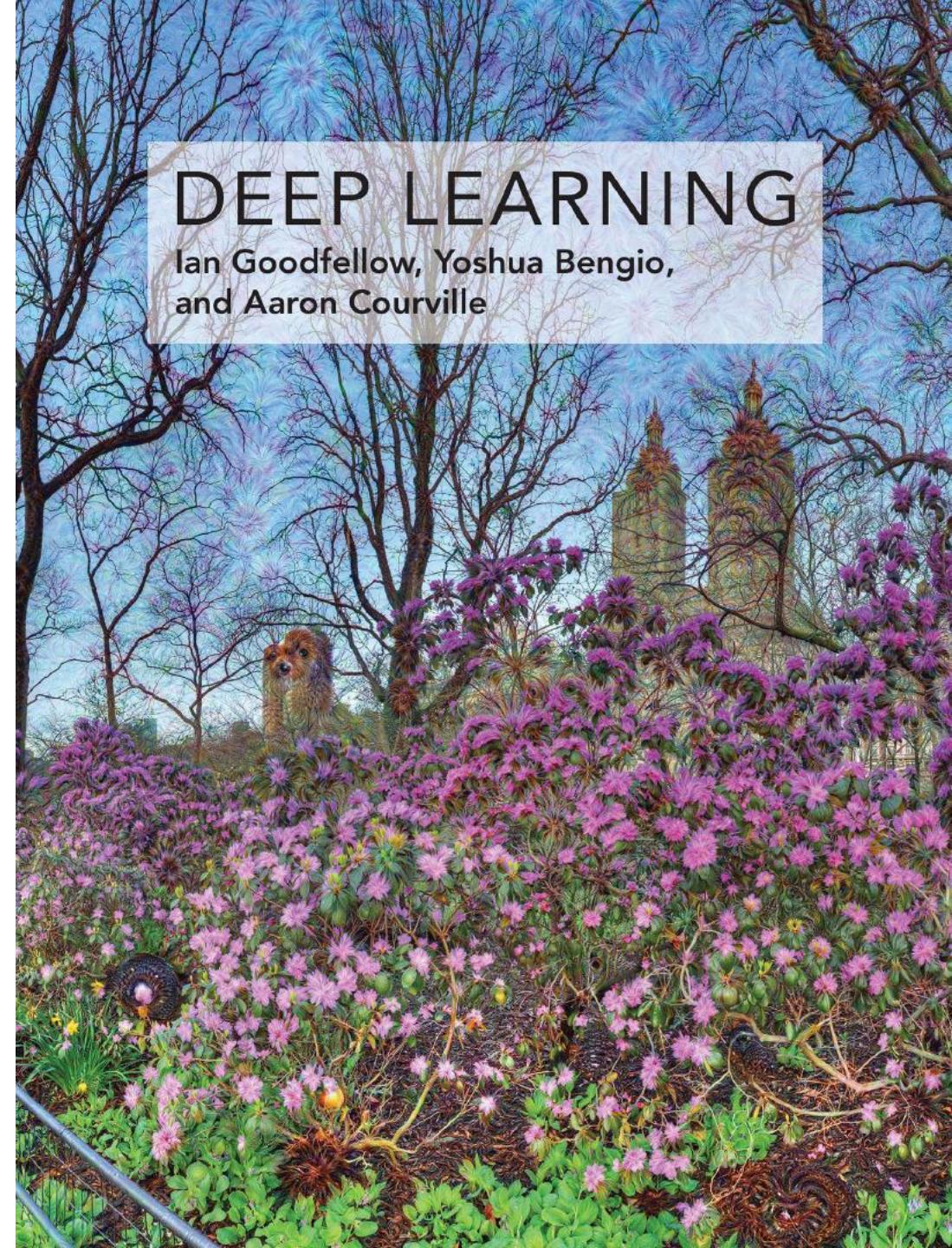


**Website** <https://aykuterdem.github.io/classes/comp541.f22/>

- Blackboard 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 text book for a quick review.

FALL 2022

SELF-ASSESSMENT QUIZ

COMP541 Deep Learning, Fall 2022

COMP541

SELF-ASSESSMENT QUIZ (THEORY)

Due Date: 5pm, Saturday, February 19, 2021

Each student enrolled to COMP541 *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  $[\theta_1, \dots, \theta_d]$  and offset  $\theta_0$ . 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  $\theta$  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

$$p(X_1 = x_1) = \alpha_1 e^{-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}}$$
$$p(X_2 = x_2 | X_1 = x_1) = \alpha_2 e^{-\frac{(x_2 - \mu_2)^2}{2\sigma_2^2}}$$

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  $\alpha_2$ ,  $\mu_2$  and  $\sigma_2$ .

**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 3 (23:59).

Each student enrolled to COMP541  
must complete and pass this quiz!

# Prerequisites

- Calculus and linear algebra
  - Derivatives,
  - Matrix operations
- Probability
- The self-assessment quiz on programming background will be released later this week!
- Machine learning
- Programming

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

SELF-ASSESSMENT QUIZ

FALL 2022

COMP541 Deep Learning, Fall 2022

SELF-ASSESSMENT QUIZ (THEORY)

COMP541

Due Date: 5pm, Saturday, February 19, 2021

Each student enrolled to COMP541 *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  $[\theta_1, \dots, \theta_d]$  and offset  $\theta_0$ . 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  $\theta$  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.

... of the matrix  $A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$

by explicitly calculating the values of  $x_1, x_2, x_3$ .

**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 3 (23:59).

Each student enrolled to COMP541 must complete and pass this quiz!

# 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	21%
Course Project	32%
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

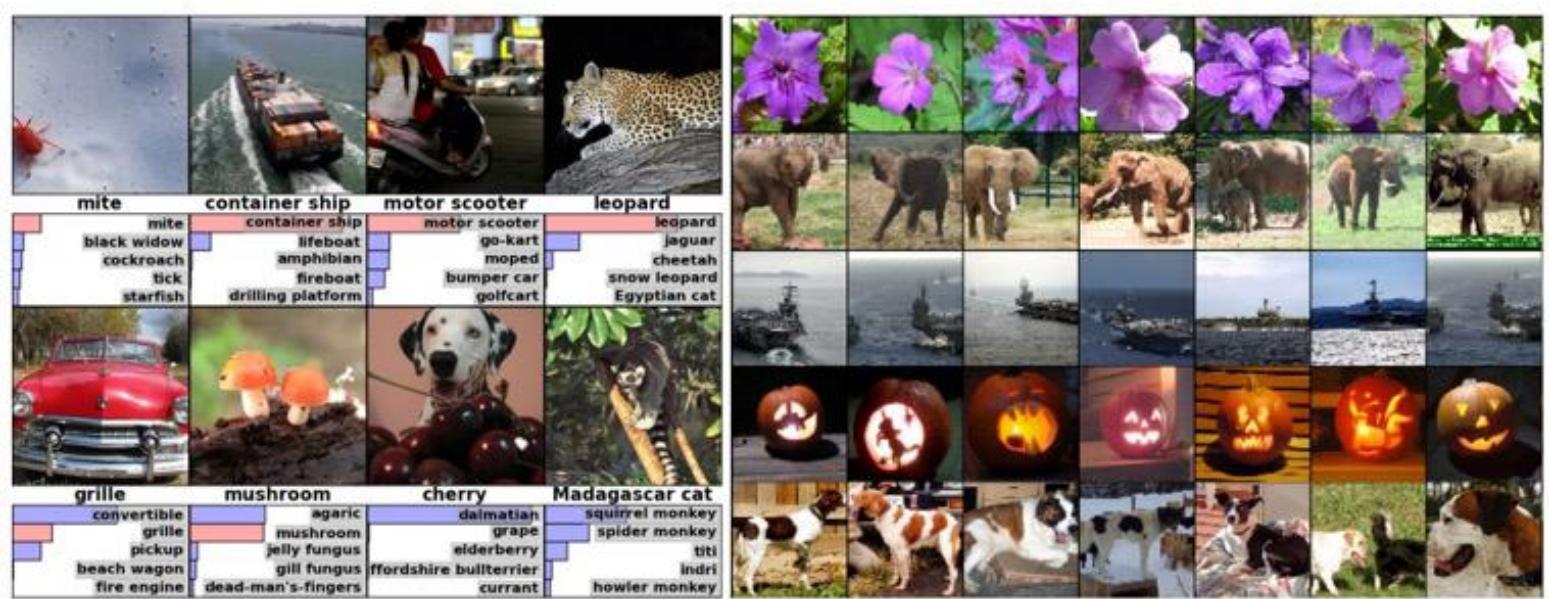
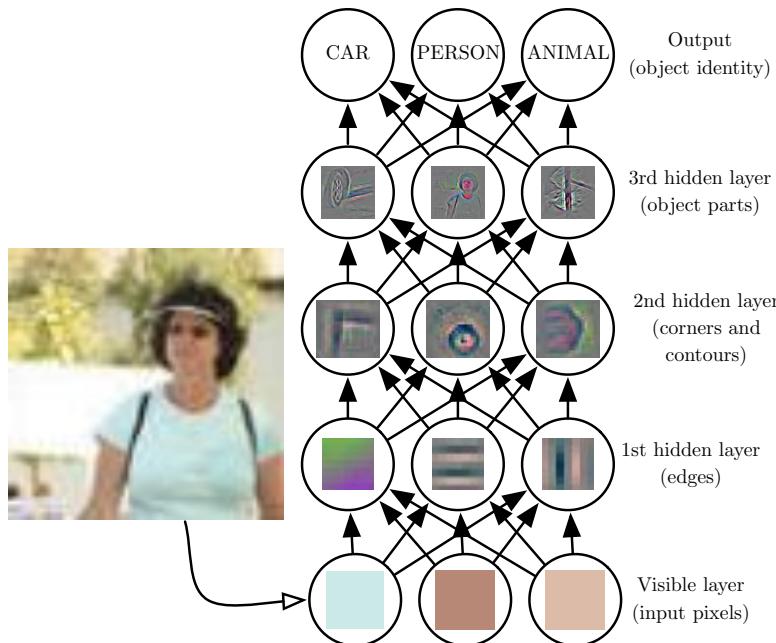
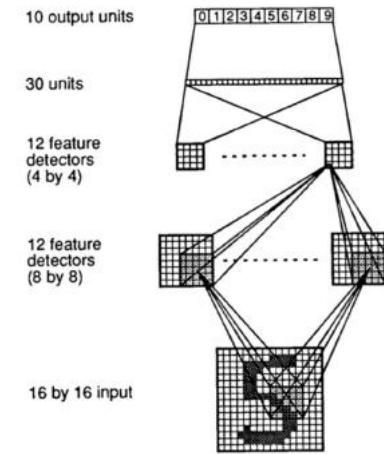
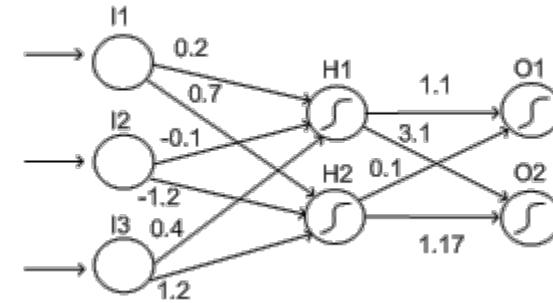
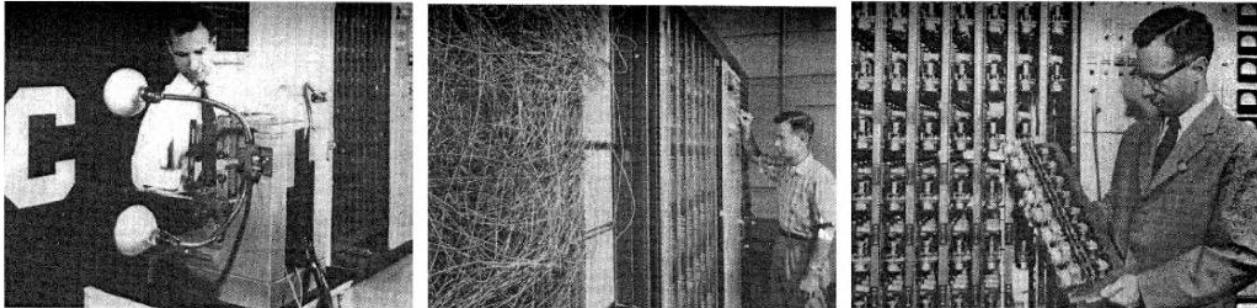
**Week 7**      [Winter Break]

**Week 8**      Recurrent Neural Networks

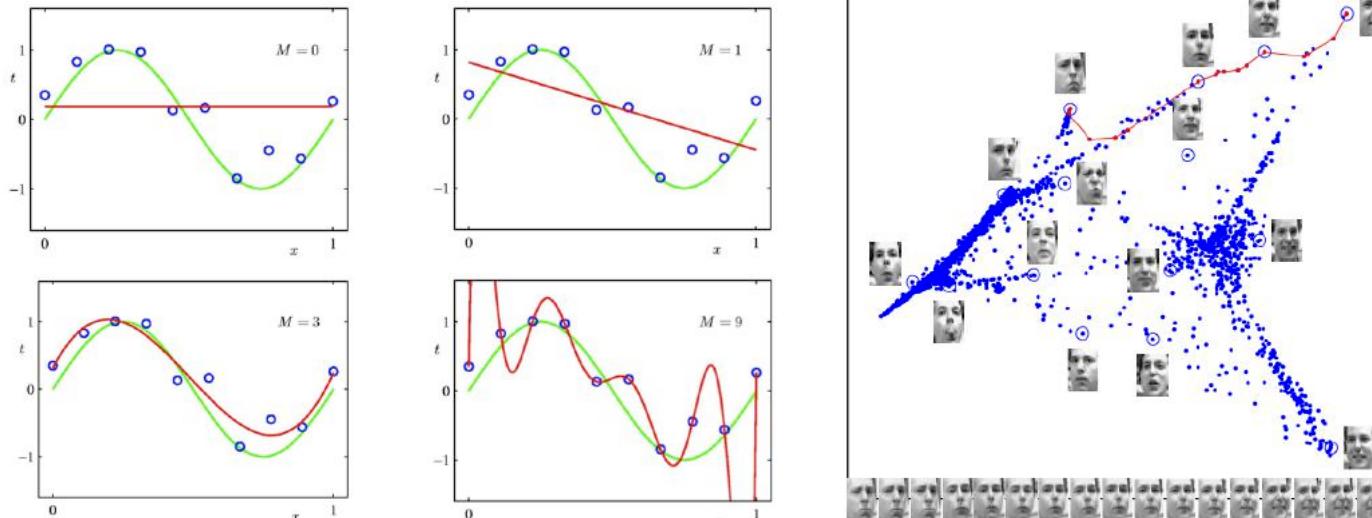
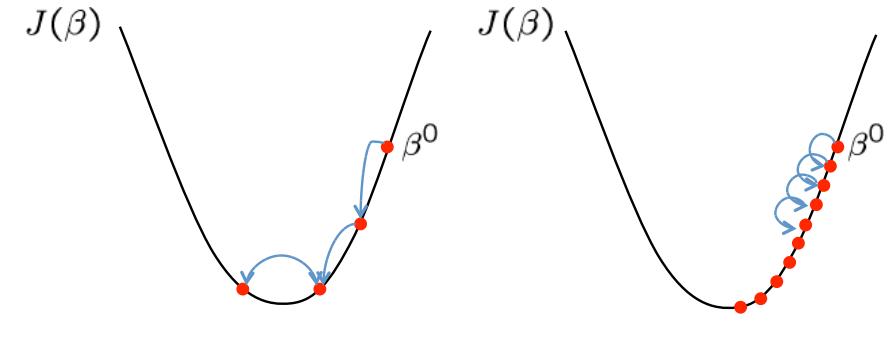
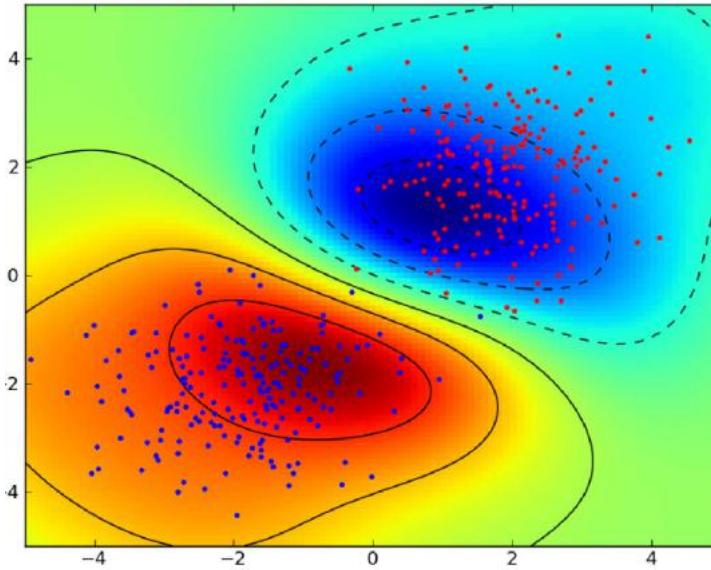
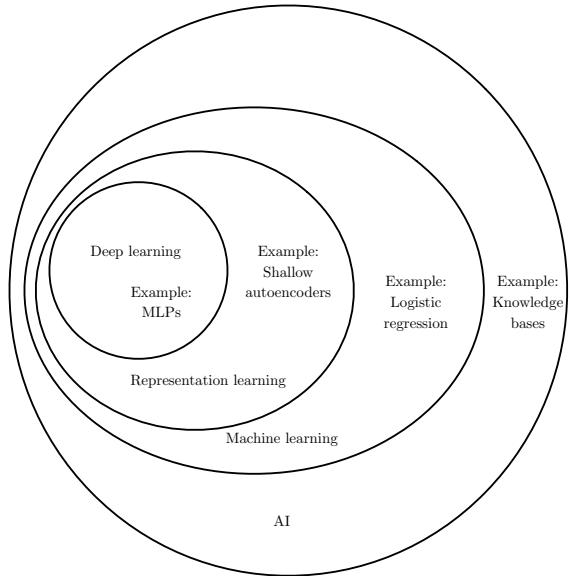
# Schedule

- |                |  |
|----------------|--|
| <b>Week 9</b>  | Attention and Transformers             |
| <b>Week 10</b> | Graph Neural Networks                  |
| <b>Week 11</b> | Autoencoders and Autoregressive Models |
| <b>Week 12</b> | Generative Adversarial Networks        |
| <b>Week 13</b> | Variational Autoencoders               |
| <b>Week 14</b> | Self-supervised Learning               |
| <b>Week 15</b> | Deep Neural Networks as Priors         |

# Lecture 1: Introduction to Deep Learning

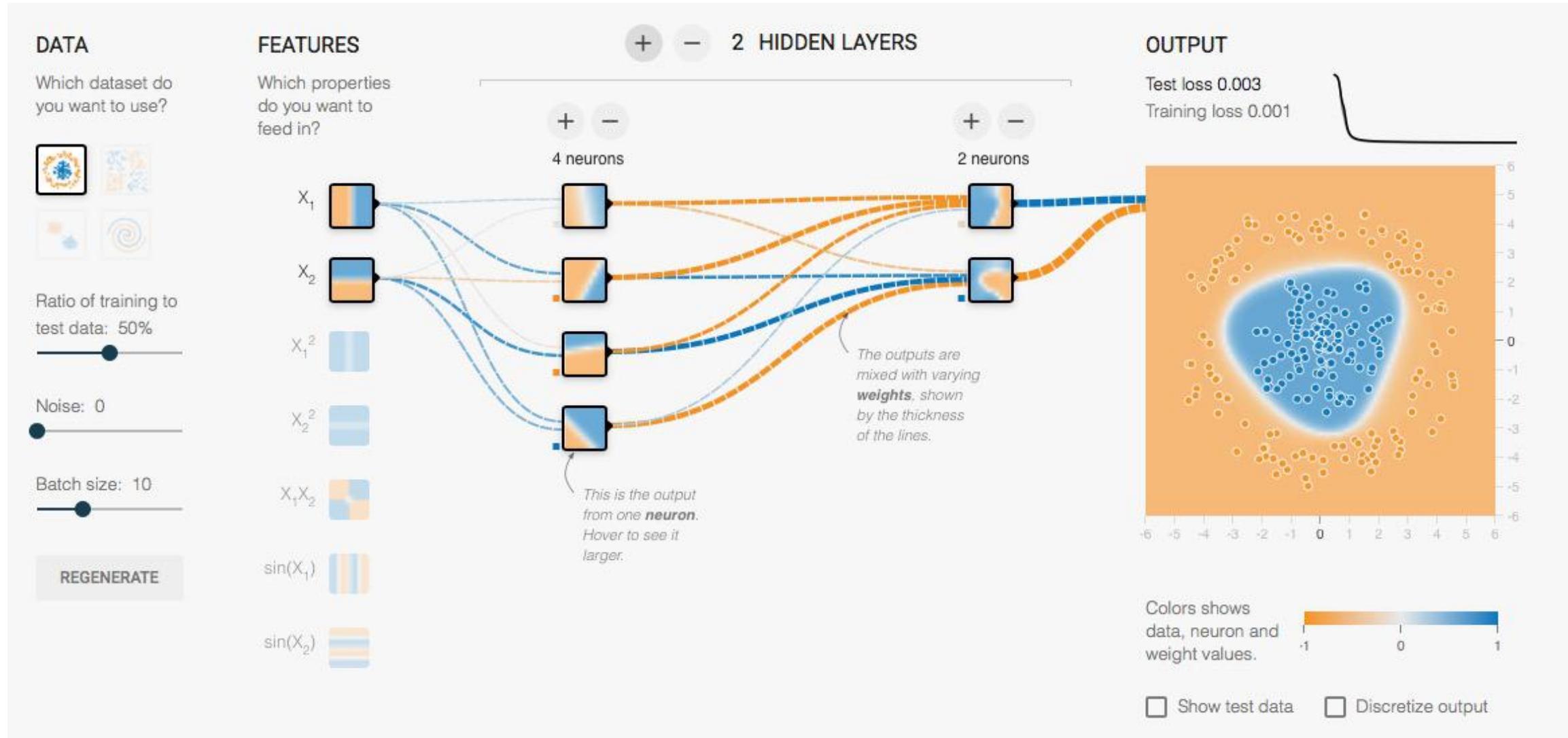


# Lecture 2: Machine Learning Overview

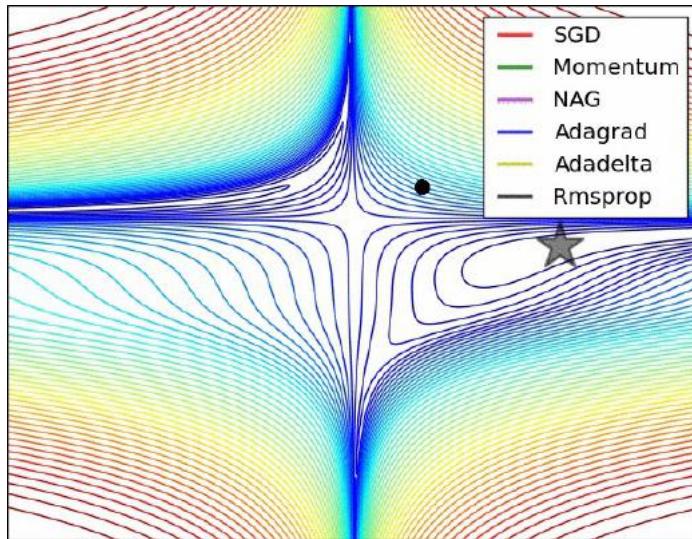


8	9	0	1	2	3	4	7	8	9	0	1	2	3	4	5	6	7	8	6
4	2	6	4	7	5	5	4	7	8	9	2	9	3	9	3	8	2	0	5
0	1	0	4	2	6	5	3	5	3	8	0	0	3	4	1	5	3	0	8
3	0	6	2	7	1	1	8	1	7	1	3	8	9	7	6	7	4	1	6
7	5	1	7	1	9	8	0	6	9	4	9	9	3	7	1	9	2	2	5
3	7	8	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	0
1	2	3	4	5	6	7	8	9	8	1	0	5	5	1	9	0	4	1	9
3	8	4	7	7	8	5	0	6	5	5	3	3	3	9	8	1	4	0	6
1	0	0	6	2	1	1	3	2	8	8	7	8	4	6	0	2	0	3	6
8	7	1	5	9	9	3	2	4	9	4	6	5	3	2	8	5	9	4	1
6	5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5	6	7	8	9	6	4	2	6	4	7	5	5
4	7	8	9	2	9	3	9	3	8	2	0	9	8	0	5	6	0	1	0
4	2	6	5	5	4	3	4	1	5	3	0	8	3	0	6	2	7	1	
1	8	1	7	1	3	8	5	4	2	0	9	7	6	7	4	1	6	8	4
7	5	1	2	6	7	1	9	8	0	6	9	4	9	9	6	2	3	7	1
9	2	2	5	3	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3
4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	2	1	2	1	3
9	9	8	5	3	7	0	7	7	5	7	9	9	4	7	0	3	4	1	4
4	7	5	8	1	4	8	4	1	8	6	6	4	4	3	5	7	2	5	9

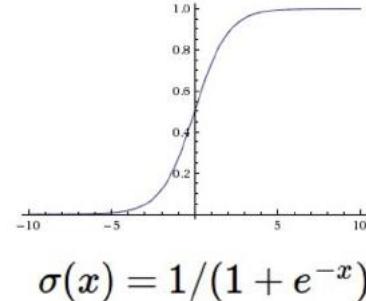
# Lecture 3: Multi-Layer Perceptrons



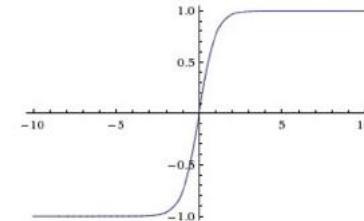
# Lecture 4: Training Deep Neural Networks



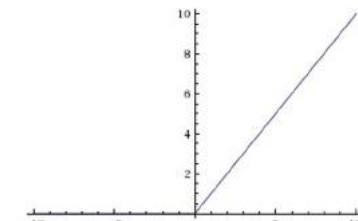
Sigmoid



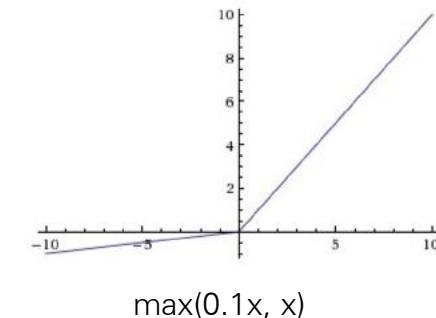
tanh



ReLU

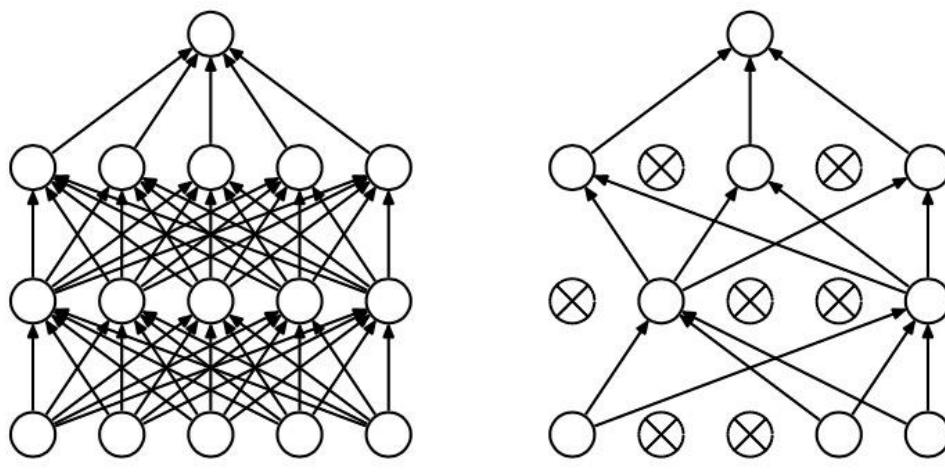


Leaky ReLU



## Activation Functions

Optimizers



(a) Standard Neural Net

(b) After applying dropout.

Dropout

**Input:** Values of  $x$  over a mini-batch:  $\mathcal{B} = \{x_1 \dots m\}$ ;  
Parameters to be learned:  $\gamma, \beta$

**Output:**  $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

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

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

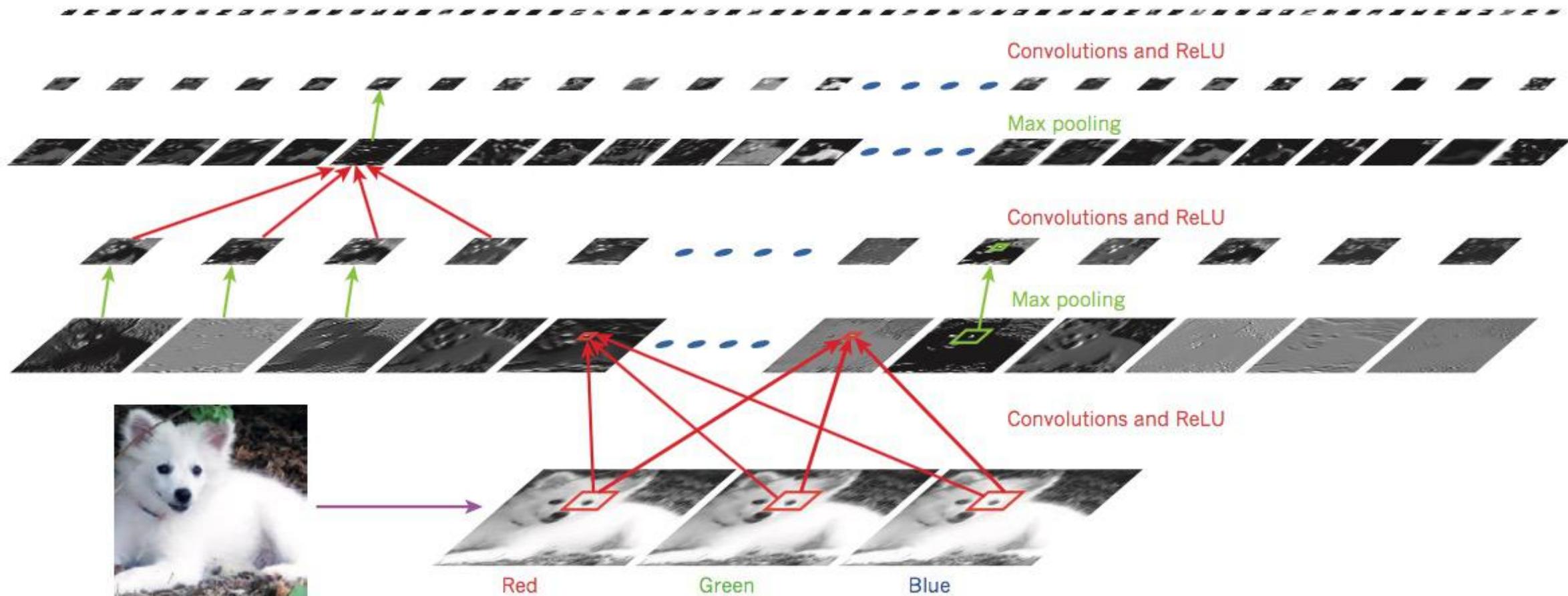
$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

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

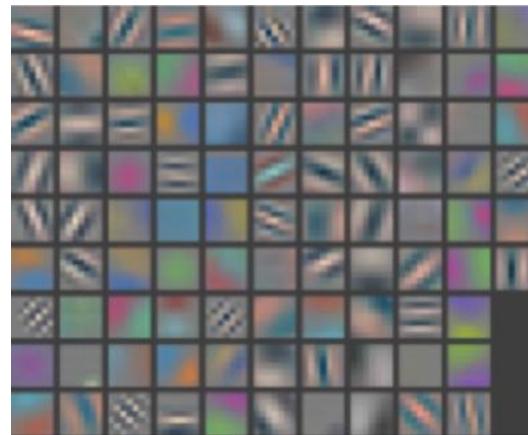
Batch Normalization

# Lecture 5: Convolutional Neural Networks

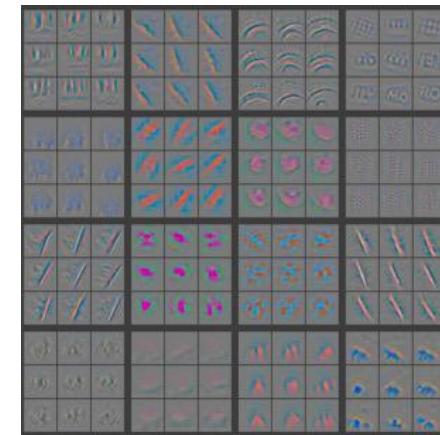
Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



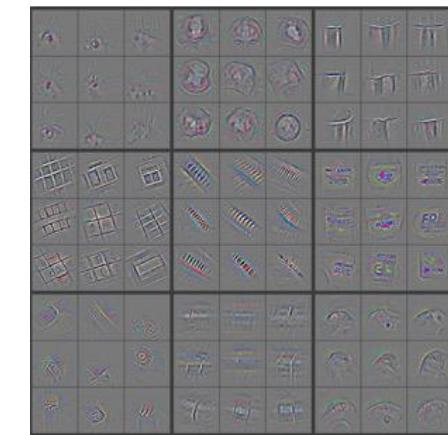
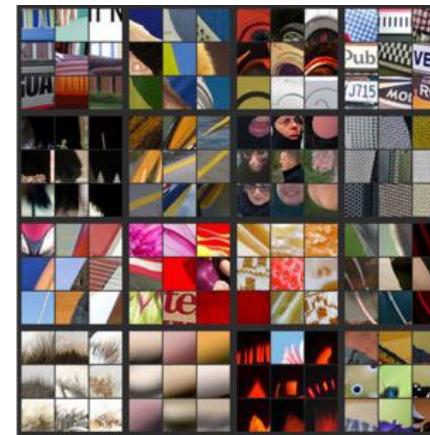
# Lecture 6: Understanding and Visualizing CNNs



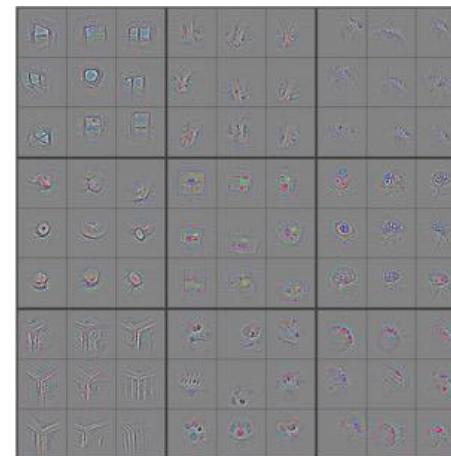
Layer 1



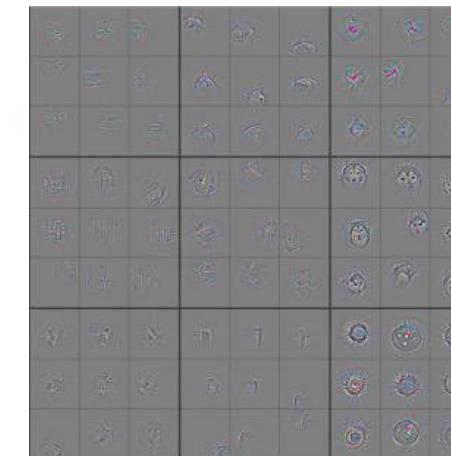
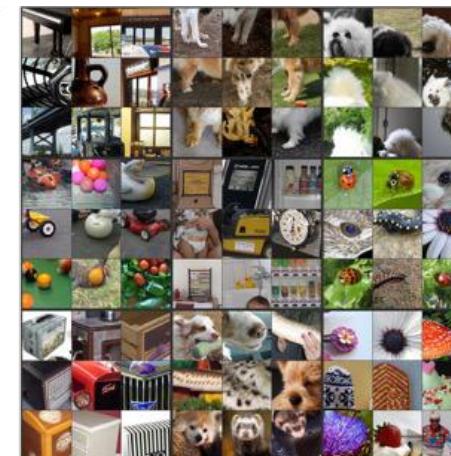
Layer 2



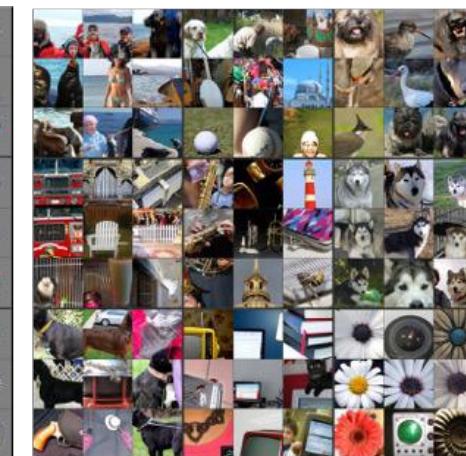
Layer 3



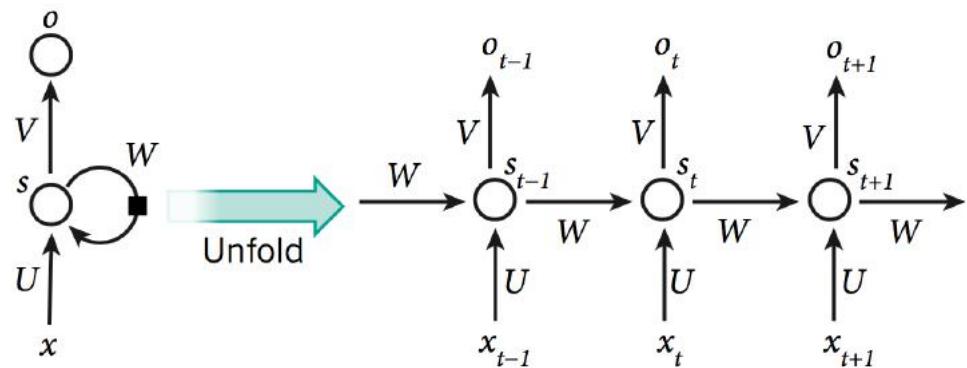
Layer 4



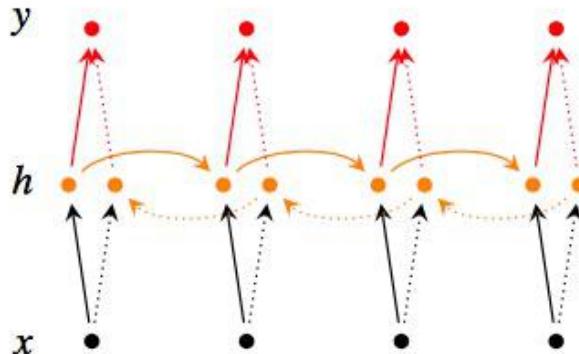
Layer 5



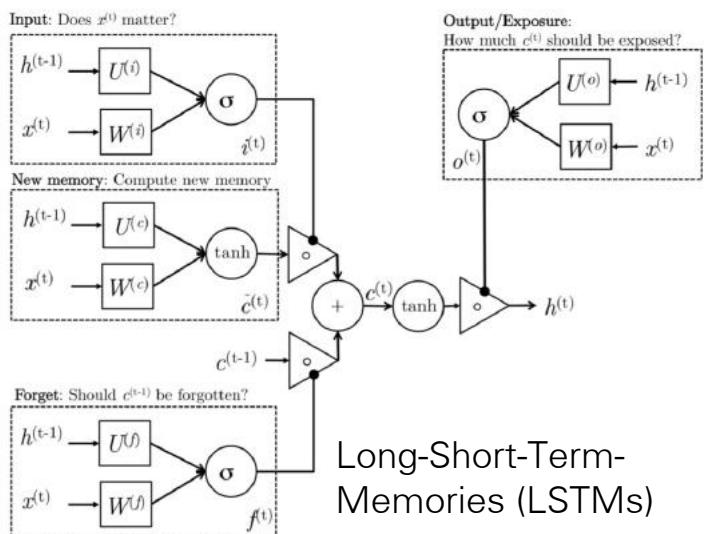
# Lecture 7: Recurrent Neural Networks



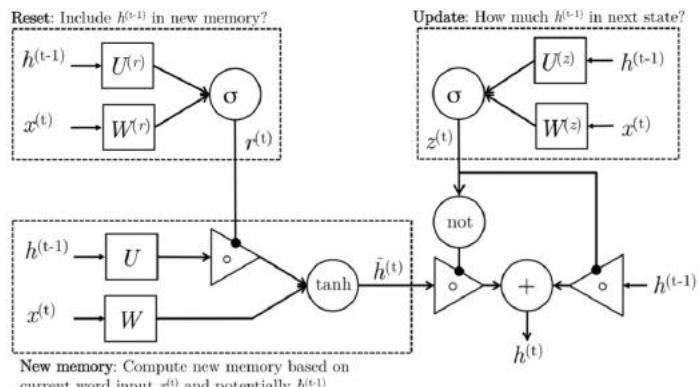
A Recurrent Neural Network (RNN)  
(unfolded across time-steps)



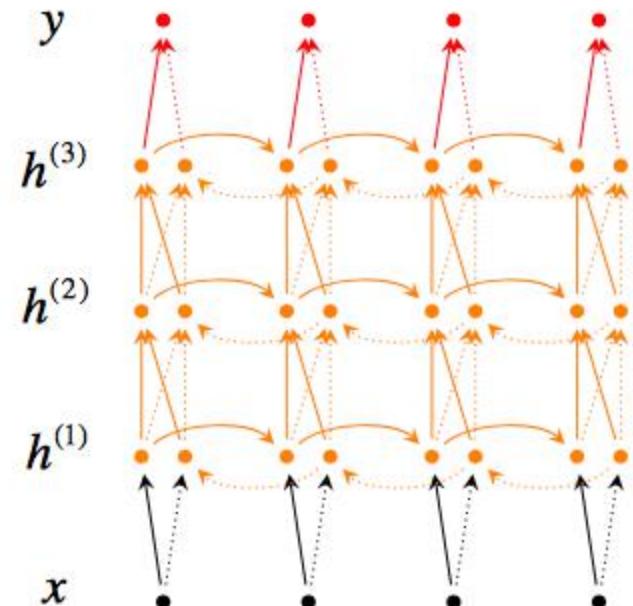
A bi-directional RNN



Long-Short-Term-Memories (LSTMs)



Gated Recurrent Units (GRUs)



A deep bi-directional RNN

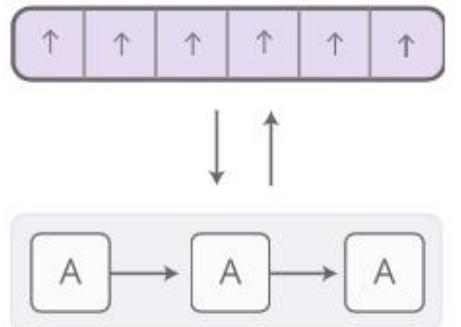
# Lecture 8: Attention and Transformers



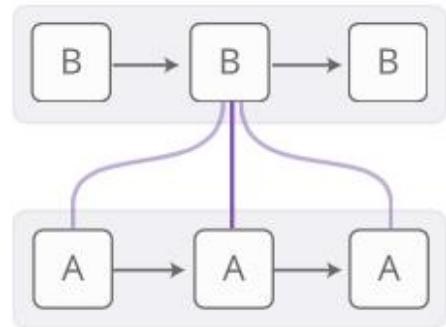
A little girl sitting on a bed with a teddy bear.



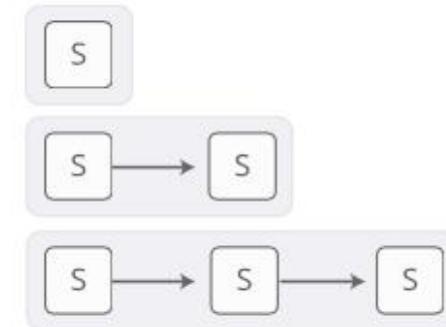
A group of people sitting on a boat in the water.



**Neural Turing  
Machines**

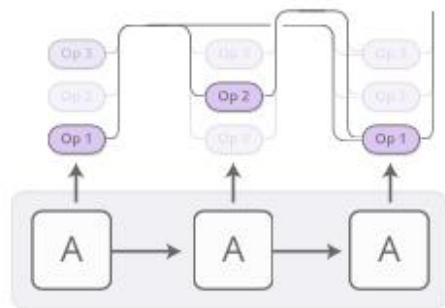


**Attentional  
Interfaces**



**Adaptive  
Computation Time**

**Transformer Architecture**



**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.

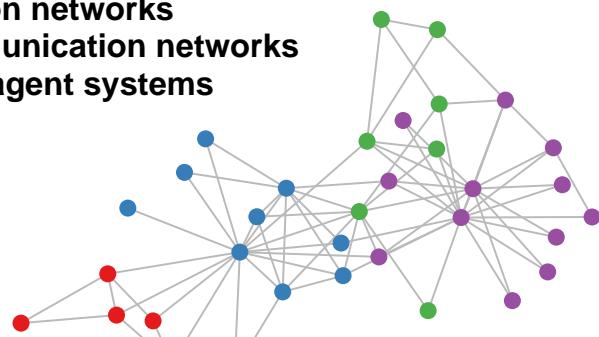
# Lecture 9: Graph Networks

Social networks

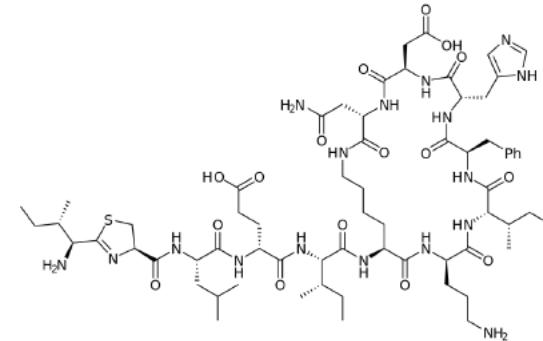
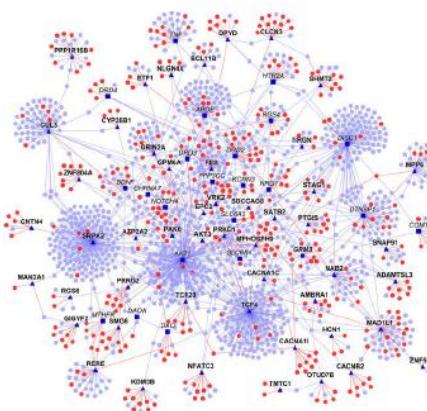
Citation networks

Communication networks

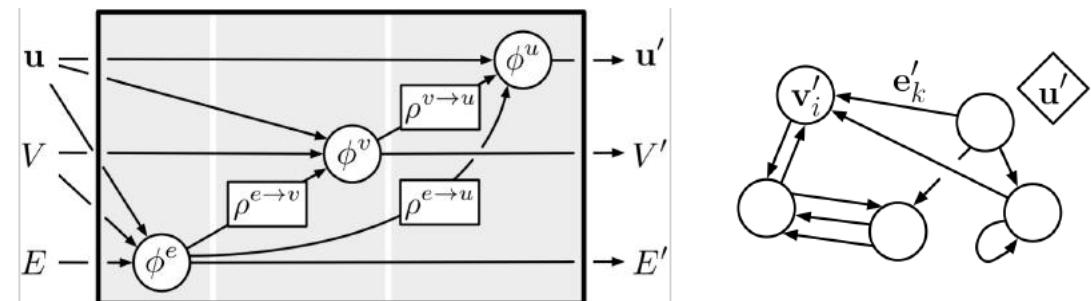
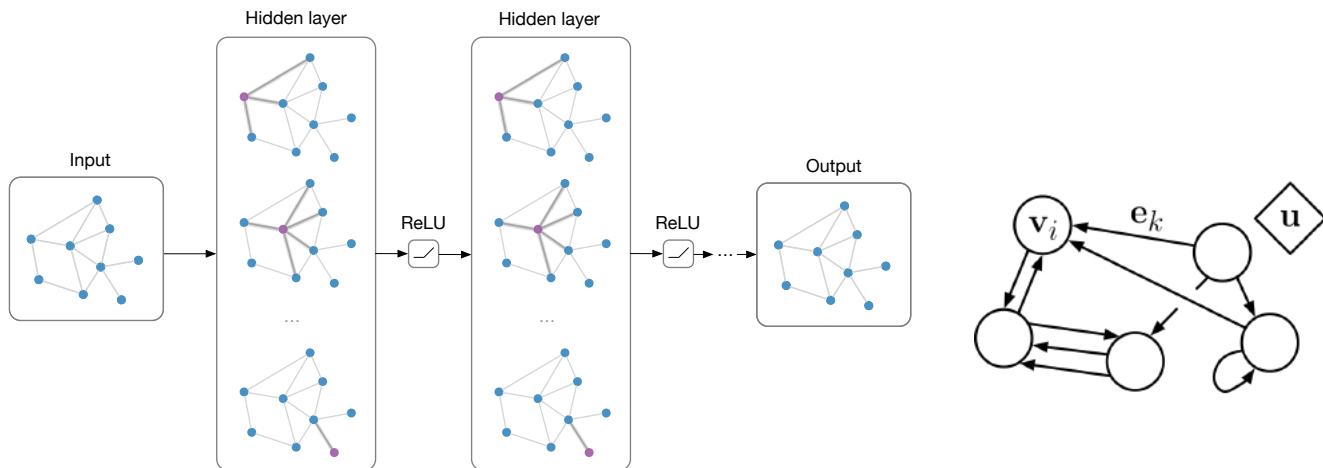
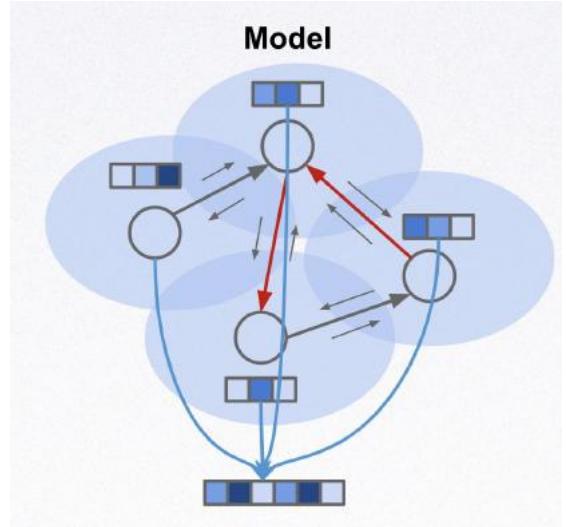
Multi-agent systems



Protein interaction  
networks



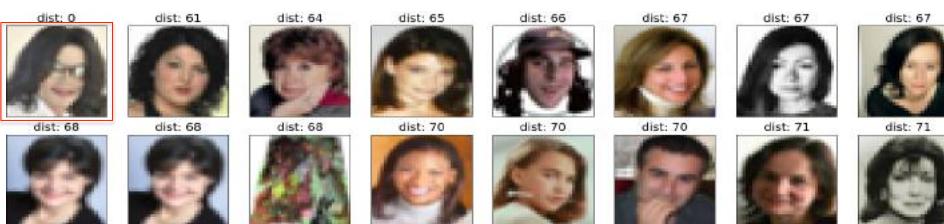
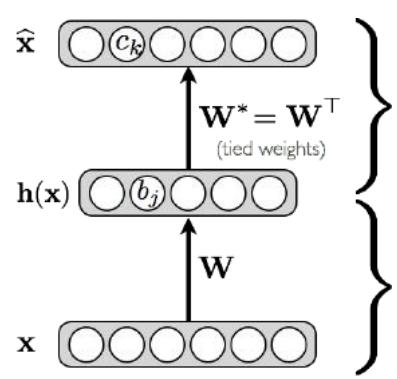
Molecules



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

# Lecture 10: Autoencoders and Autoregressive Models



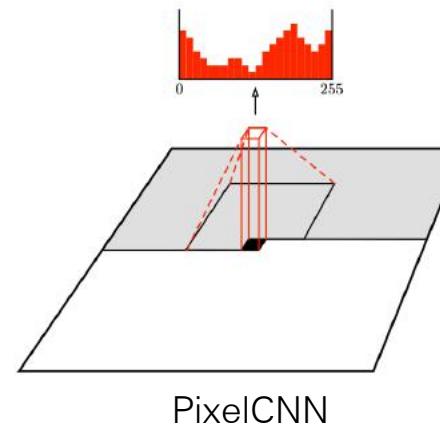
Decoder

$$\begin{aligned}\hat{x} &= o(\hat{a}(x)) \\ &= \text{sigm}(\mathbf{c} + \mathbf{W}^* \mathbf{h}(x))\end{aligned}$$

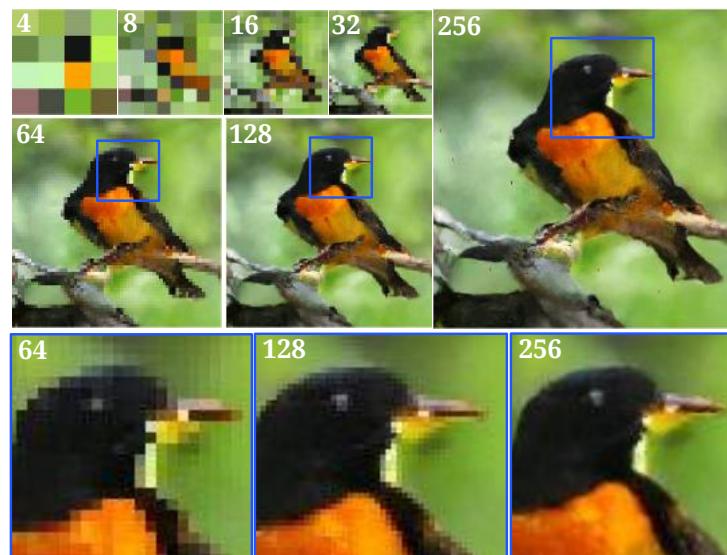
For binary units

Encoder

$$\begin{aligned}h(x) &= g(a(x)) \\ &= \text{sigm}(\mathbf{b} + \mathbf{W}x)\end{aligned}$$



Class conditioned samples generated by PixelCNN



Text-to-image synthesis with Parallel Multiscale PixelCNNs

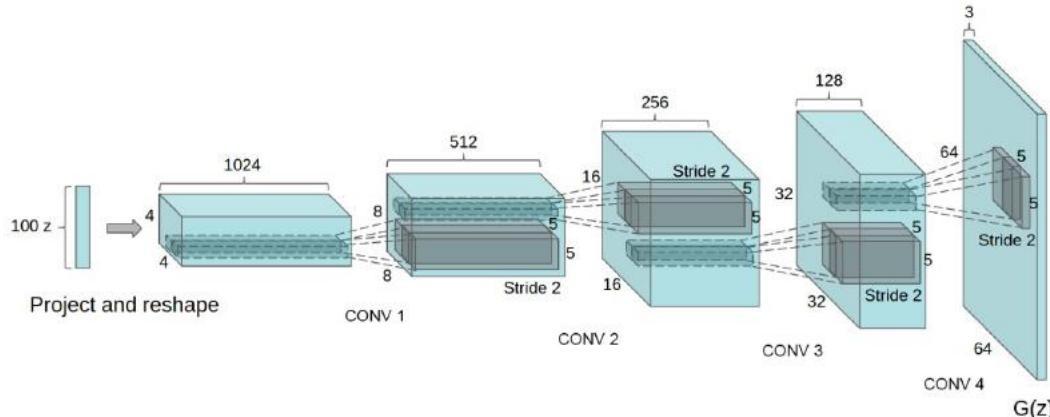
*"A yellow bird with a black head, orange eyes and an orange bill."*

A. Krizhevsky and G. E. Hinton, "Using Very Deep Autoencoders for Content-Based Image Retrieval", ESANN 2011

A. van den Oord et al., "Conditional Image Generation with PixelCNN Decoders", NeurIPS 2016

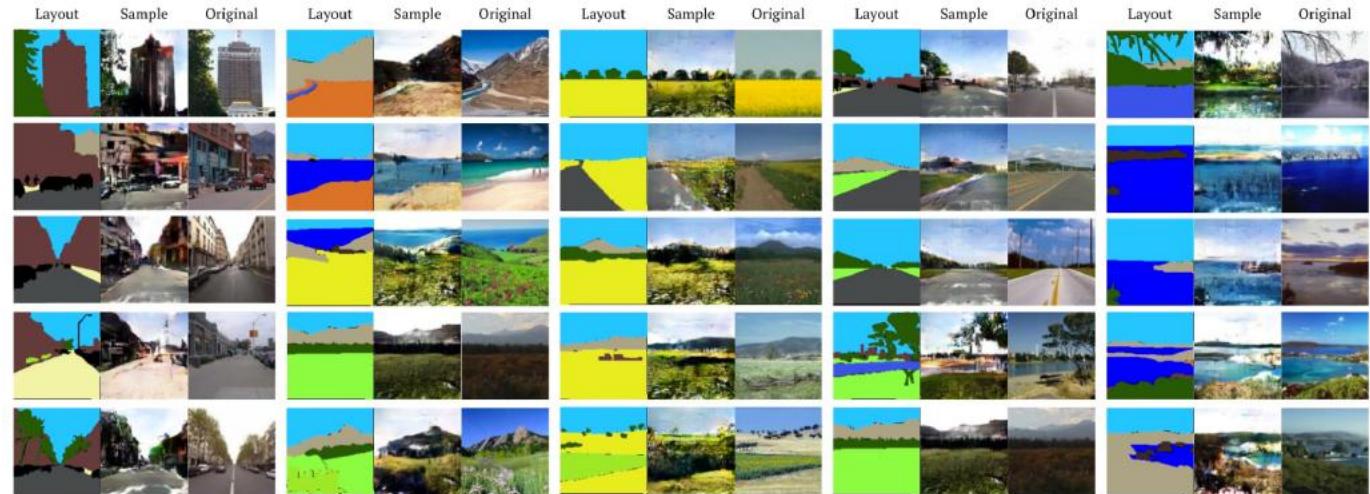
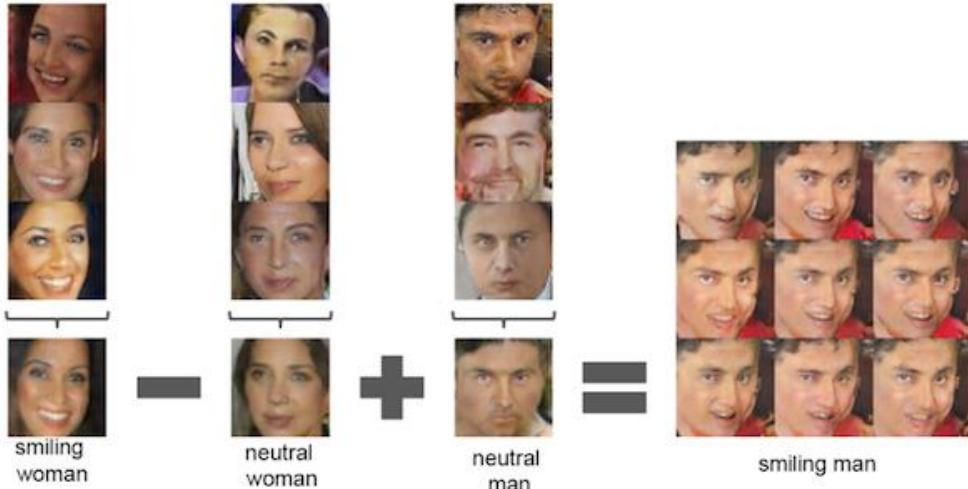
S. Reed et al., "Parallel Multiscale Autoregressive Density Estimation", ICML 2017

# Lecture 11: Generative Adversarial Networks



Class-conditioned samples generated by BigGAN

$$\min_{\theta} \max_{\omega} \mathbb{E}_{x \sim Q} [\log D_{\omega}(x)] + \mathbb{E}_{x \sim P_{\theta}} [\log(1 - D_{\omega}(x))]$$



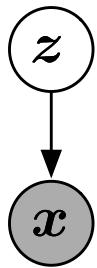
I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets", NIPS 2014.

A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", ICLR 2016

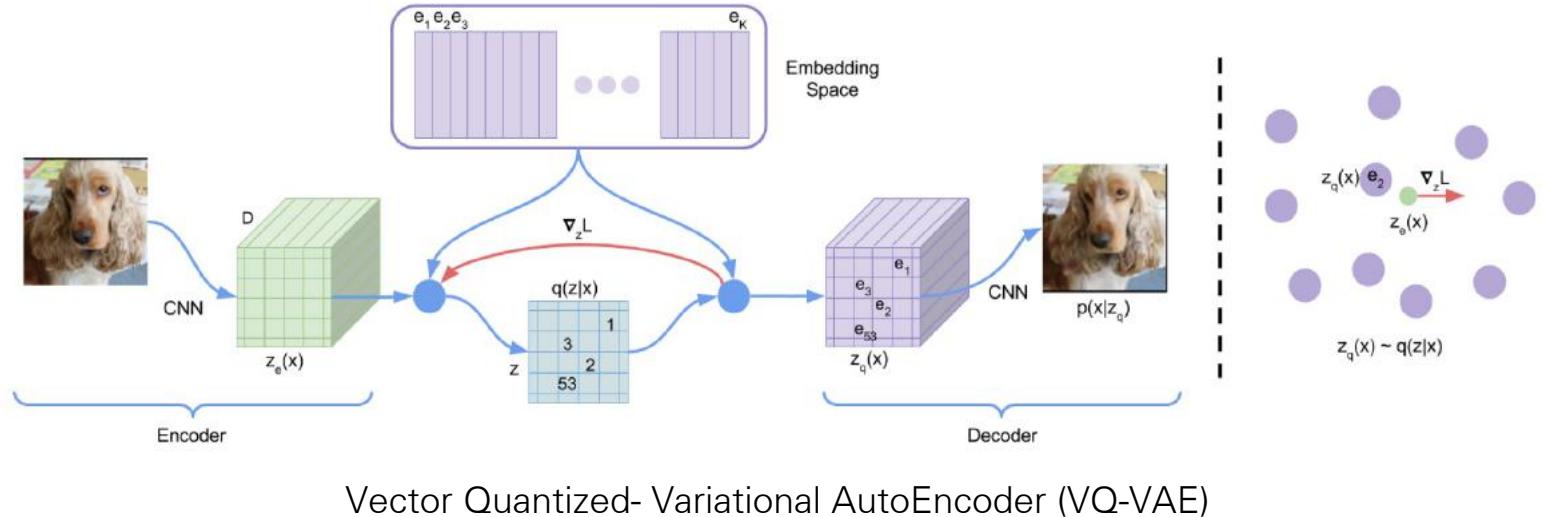
L. Karacan, Z. Akata, A. Erdem and E. Erdem, "Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts", arXiv preprint 2016

A. Brock, J. Donahue, K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", ICLR2019

# Lecture 12: Variational Autoencoders



$$\begin{aligned} \log p(\mathbf{x}) &\geq \log p(\mathbf{x}) - D_{\text{KL}}(q(z) \| p(z | \mathbf{x})) \\ &= \mathbb{E}_{z \sim q} \log p(\mathbf{x}, z) + H(q) \end{aligned}$$



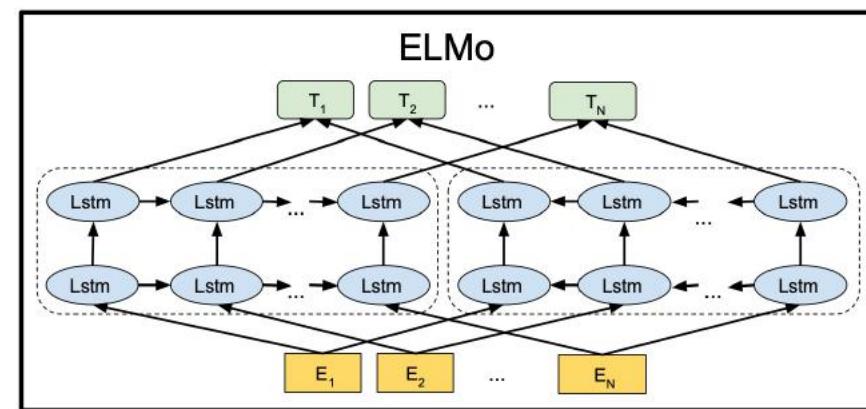
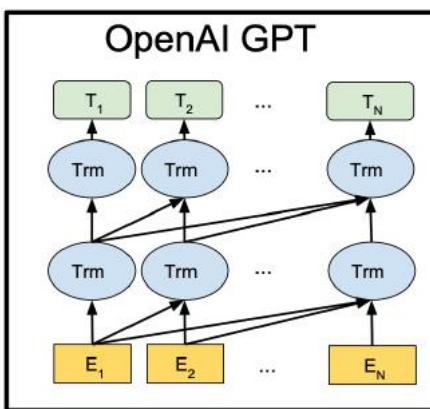
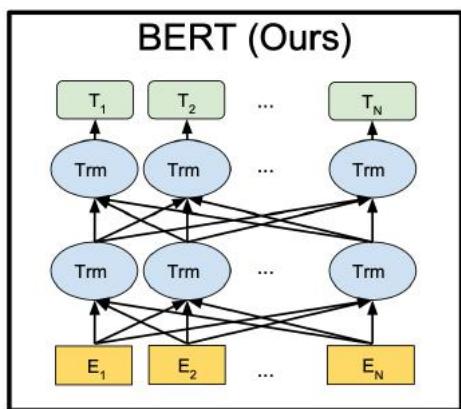
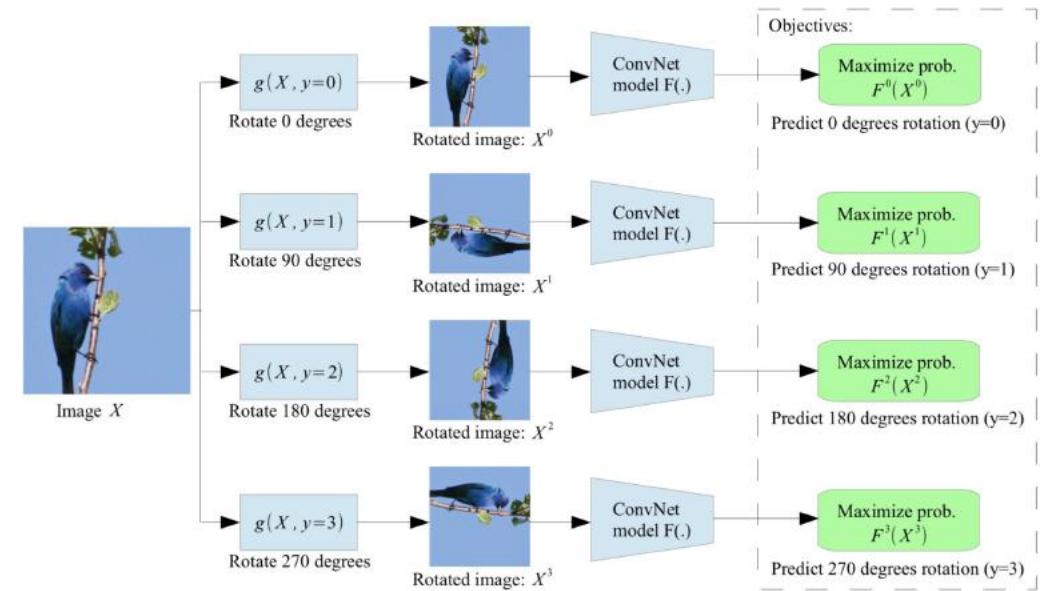
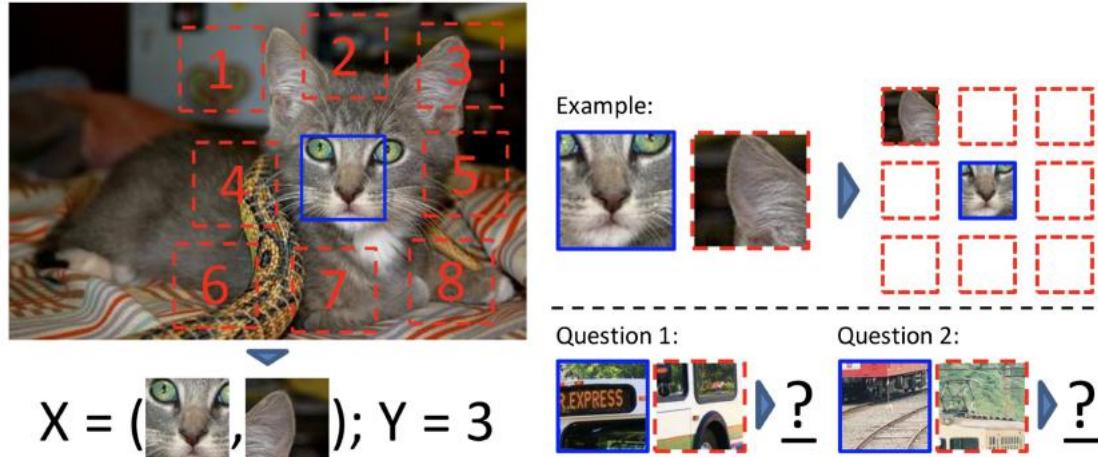
Synthetic images generated by VQ-VAE2

D. P. Kingma and M. Welling, "Auto-encoding variational Bayes", ICLR 2014

A. van den Oord, O. Vinyals, K. Kavukcuoglu, "Neural Discrete Representation Learning", NeurIPS 2017

A. Razavi, A. van den Oord, O. Vinyals, "Generating Diverse High-Fidelity Images with VQ-VAE-2",

# Lecture 13: Self-supervised Learning



C. Doersch, A. Gupta, A. A. Efros, "Unsupervised Visual Representation Learning by Context Prediction", ICCV 2015.

S. Gidaris, P. Singh, N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations", ICLR2018.

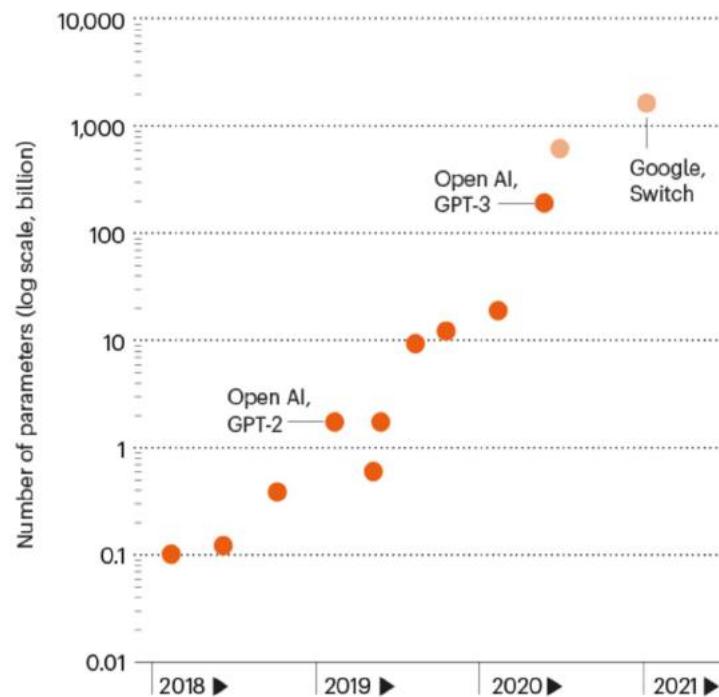
J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", NAACL-HLT 2019.

# Lecture 14: Deep Neural Networks as Priors

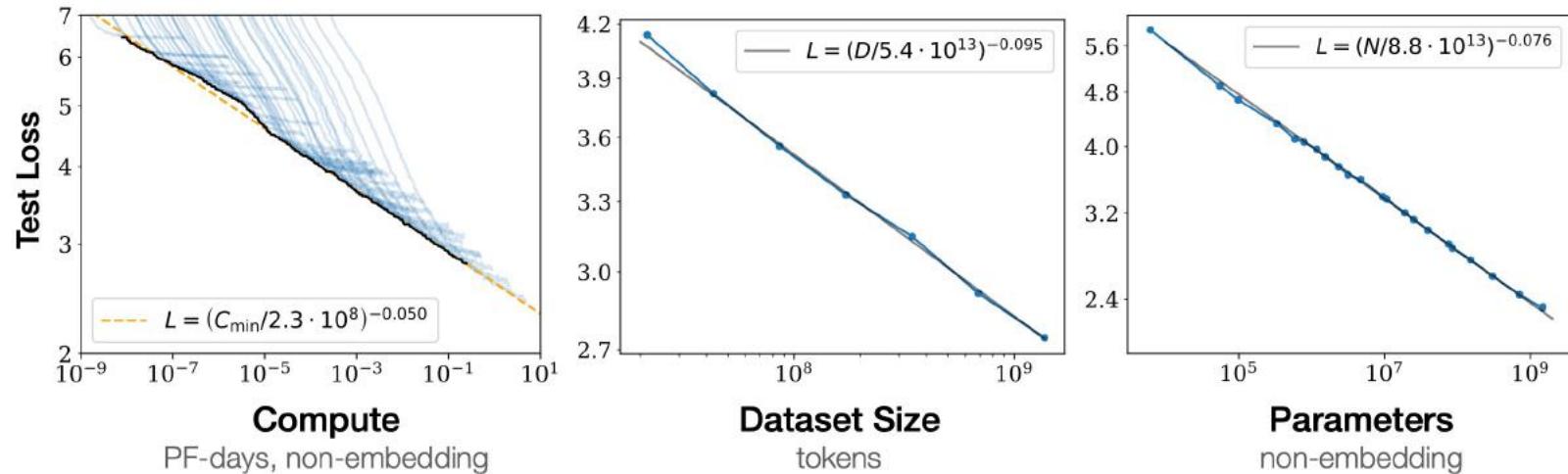
## LARGER LANGUAGE MODELS

The scale of text-generating neural networks is growing exponentially, as measured by the models' parameters (roughly, the number of connections between neurons).

● 'Dense' models ● 'Sparse' models\*



\*Google's 1.6-trillion parameter 'sparse' model has performance equivalent to that of 10 billion to 100 billion parameter 'dense' models. ©nature



**Figure 1** Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute<sup>2</sup> 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.

# 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

[Start of paper presentations](#)

## L5 Convolutional Neural Networks

[Assignment 1 due, Assignment 2 out](#)

## L6 Understanding and Visualizing CNNs

[Project proposals due](#)

## L7 Recurrent Neural Networks

[Assignment 2 due, Assignment 3 out](#)

## L8 Attention and Transformers

[Midterm Exam](#)

## L9 Graph Neural Networks

[Assignment 3 due, Assignment 4 out](#)

## L10 Autoencoders and Autoregressive

[Models](#)

[Project progress reports due](#)

## L11 Generative Adversarial Networks

[Assignment 4 due](#)

## L12 Variational Autoencoders

## L13 Self-supervised Learning

## L14 Deep Neural Networks as Priors

[Final project reports due](#)

# Paper Presentations

We will discuss 8 recent papers related to the topics covered in the class.

- (14 mins) One group of students will be responsible from providing an overview of the paper.
- (8 mins) Another group will present the strengths of the paper.
- (8 mins) Another one will discuss the weaknesses of the paper.
- (10 mins) QA

See the rubrics on the course web page for the details,

Date	Topic
Oct 3	Introduction to Deep Learning
Oct 10	Machine Learning Overview
Oct 17	Multi-Layer Perceptrons
Oct 24	Training Deep Neural Networks
Oct 31	Convolutional Neural Networks
Nov 7	Understanding and Visualizing CNNs
Nov 14	<i>Winter Break</i>
Nov 21	Recurrent Neural Networks
Nov 28	Attention and Transformers
TBA	Midterm Exam
Dec 5	Graph Neural Networks
Dec 12	Autoencoders and Autoregressive Models
Dec 19	Generative Adversarial Networks
Dec 26	Variational Autoencoders
Jan 2	Self-supervised Learning
Jan 9	Deep Neural Networks as Priors
Jan 16	Final Project Presentations
Jan 23	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

# 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 are 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.

# 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

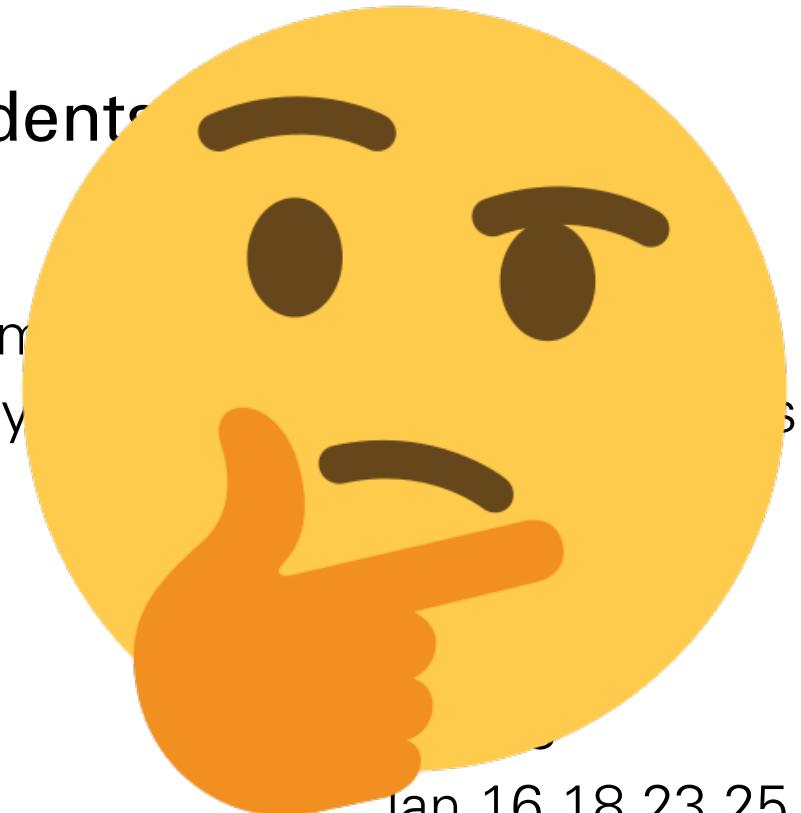
- **Date:** December 3 or 4
- **Topics:** Everything covered in the first part of the course
- Format to be a classical exam with derivations and short discussion questions.

# 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 3
  - Project progress reports (6%) Dec 18
  - Final project presentations (8%) Jan 16,18,23,25
  - Final reports (12%) Jan 29
  - 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 experiments
    - An extension to a recent study of non-trivial complexity
  - Deliverables
    - Proposals (2%)
    - Project progress reports (6%)
    - Final project presentations (8%)
    - Final reports (12%)
    - The quality of the contributions/The difficulty of implementation (4%)
- Start thinking about project ideas!**



Jan 16,18,23,25

Jan 29

# 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

10 Breakthrough

# MIT Technology Review

Forbes / Tech

APR 1, 2016 @ 06:47 AM

3,207 VIEWS

## What Is Deep Lea



Kevin Murnane  
CONTRIBUTOR

I write about science,  
technology and the  
people that connect  
them.



Opinions expressed by  
Forbes Contributors  
are their own.

TWEET THIS

Deep learning unl  
to use it

Credit: Google

Deep learning re  
GOOGL +1.40% Alpha  
ranking Go playe  
learning and Alp  
the news. Google  
driving cars all re

networks to build a program that picks out an attractive still from a

understand language and then make inferences and decisions on its

# HUFFPOST BUSINESS

Edition: US ▾

Like 7M Follow

FRONT PAGE POLITICS ENTERTAINMENT INDIE WORKING HEALTHY LIVING WORLDVIEW LIFESTYLE BUSINESS

ALL SECTIONS

Sign up for newsletters

# Science AAAS

Home News Journals Topics Careers

Authors | Members | Librarians | Advertisers

Search

Science Science Advances Science Immunology Science Robotics Science Signaling Science Translational Medicine

## How AI is transforming science

Researchers are unleashing  
artificial intelligence (AI) on  
torrents of big data

KIYOSHI TAKAHASE SEGUNDO/ALAMY STOCK  
PHOTO



## Contents 07 JULY 2017

VOL 357, ISSUE 6346

### Special Issue The cyberscientist

#### INTRODUCTION TO SPECIAL ISSUE

#### The scientists' apprentice

BY TIM APPENZELLER  
SCIENCE | 07 JUL 2017 : 16-17 | 6

Artificial intelligence helps scientists cope with torrents of data

Summary Full Text PDF

#### 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, FACEBOOK CHIEF EXECUTIVE OFFICER MARK ZUCKERBERG  
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.

Sign in



Photographer: Tomohiro Ohsumi/Bloomberg

# What is deep learning?



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

– Yann LeCun, Yoshua Bengio and Geoff Hinton

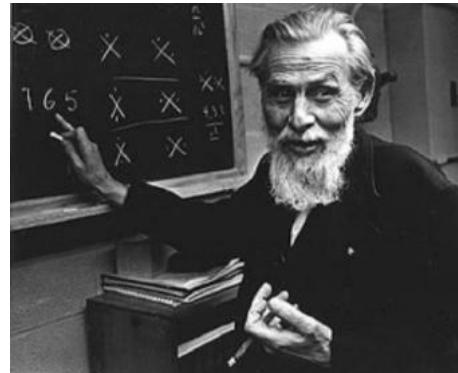
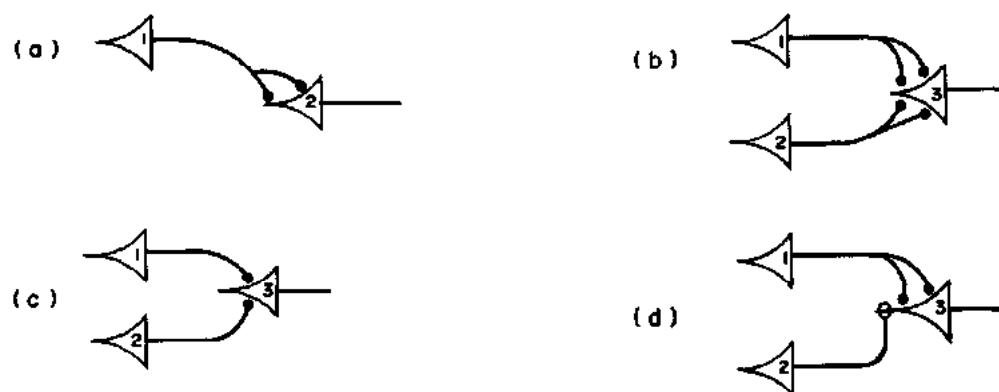


Y. LeCun, Y. Bengio, G. Hinton, "Deep Learning", Nature, Vol. 521, 28 May 2015

# 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



Action of Mathematical Biology, Vol. 32, No. 1/2, pp. 96-113, 1990  
Printed in Great Britain

0875-8246/90/0100-0096\$03.00 ©  
Princeton Press Inc.  
Society for Mathematical Biology

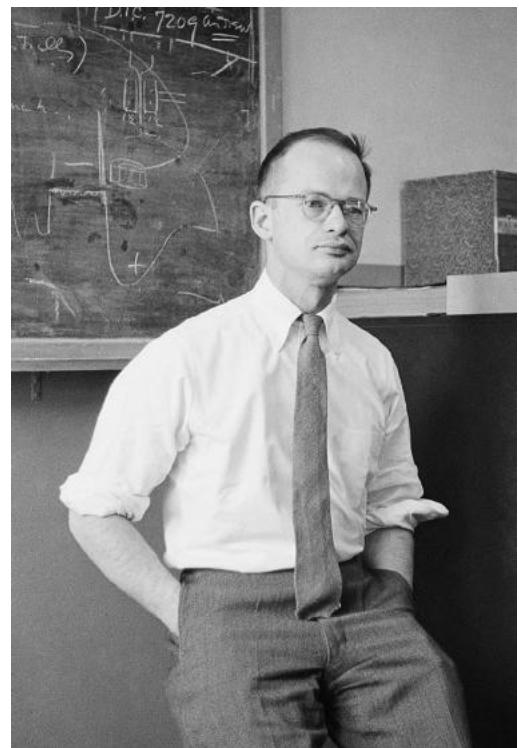
## A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY\*

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

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that two particular choices among possible neurophysiological assumptions are equivalent, in the sense that for any behavior 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.

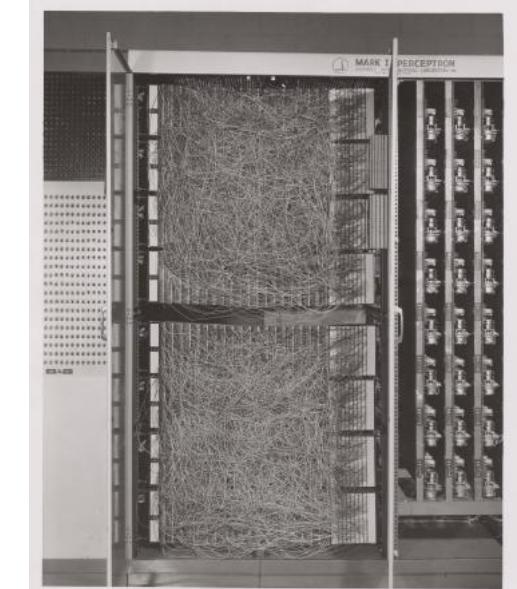
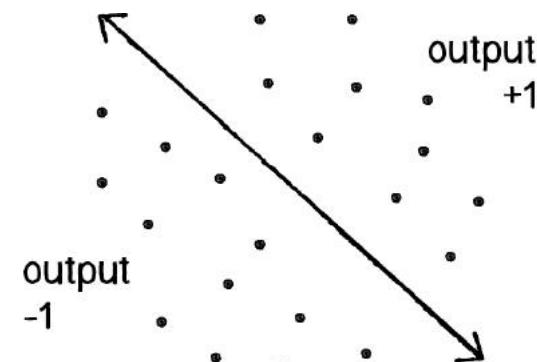
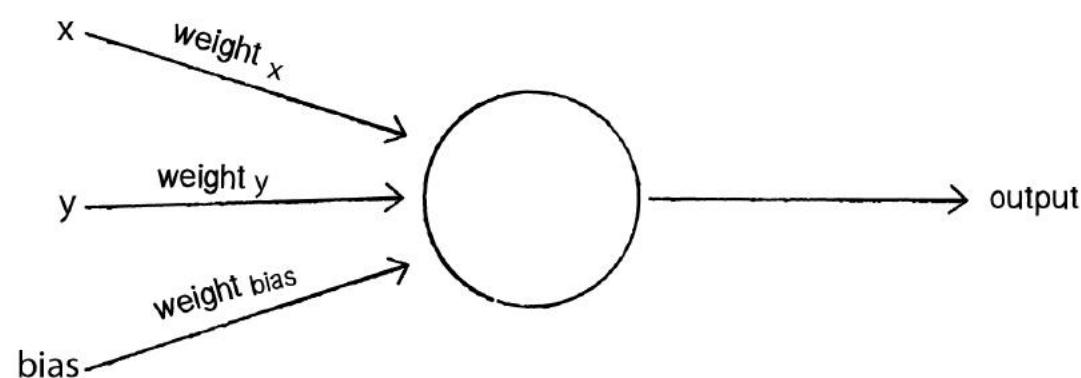
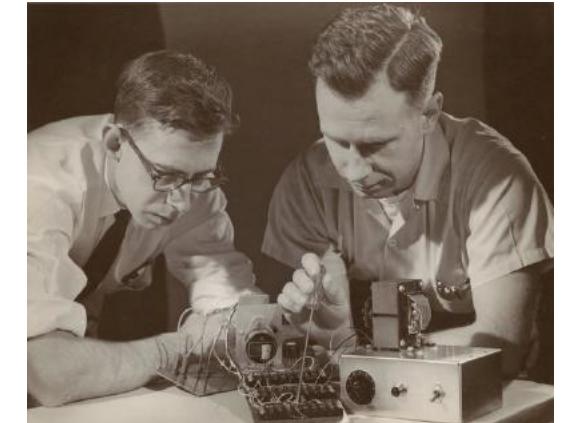
**I. Introduction.** Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurrence, 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  $< 1 \text{ ms}^{-1}$  in thin axons, which are usually short, to  $> 150 \text{ ms}^{-1}$  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 reciprocity 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  $< 0.25 \text{ ms}$ . 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

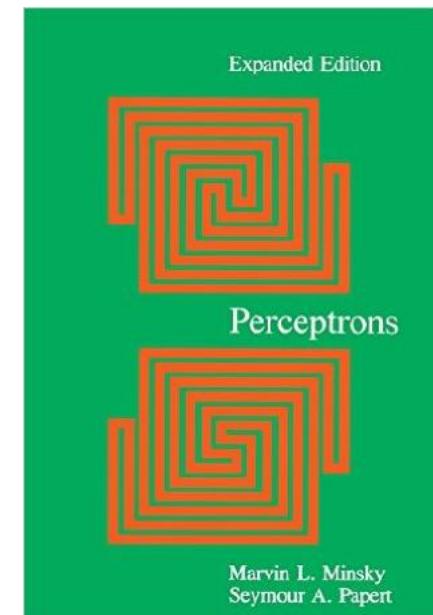
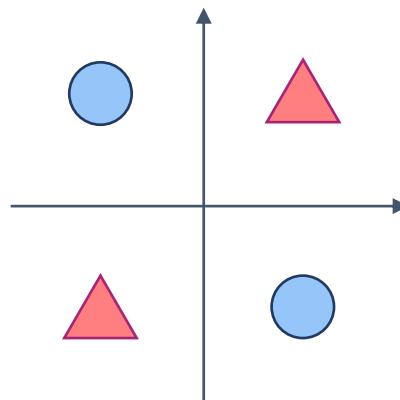


# 1969: Marvin Minsky and Seymour Papert

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

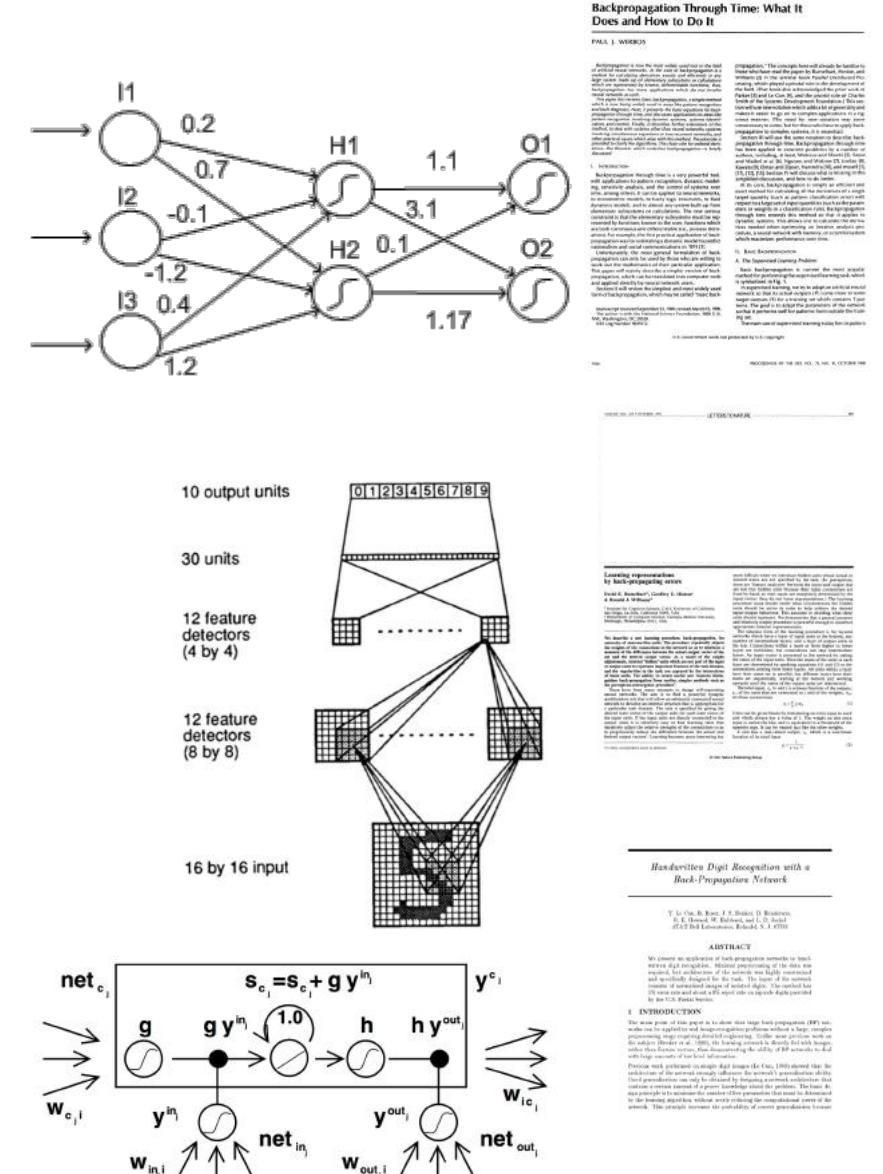


- Perceptrons can only represent linearly separable functions.
  - such as **XOR** Problem
- Wrongly attributed as the reason behind the **AI winter**, a period of reduced funding and interest in AI research



# 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.

- 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.

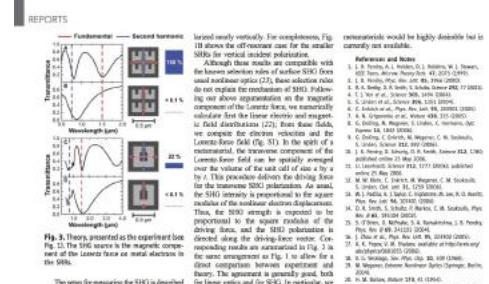


Fig. 3. Theory, presented at the experiment (see Fig. 2A). Vertical polarized source is the magnetic component of the Lorenz force or static electric field.

References and Notes

1. G. E. Hinton, R. R. Salakhutdinov, *Science*, **313**, 504 (2006).
2. J. B. Pendleton, *Phys. Rev. Lett.*, **43**, 2070 (1979).
3. J. B. Pendleton, *Phys. Rev. Lett.*, **43**, 2076 (1979).
4. J. B. Pendleton, *Phys. Rev. Lett.*, **43**, 2080 (1979).
5. J. Yariv et al., *Science*, **308**, 1446 (2005).
6. K. Saitoh et al., *Phys. Rev. Lett.*, **98**, 173901 (2007).
7. K. Saitoh et al., *Phys. Rev. Lett.*, **98**, 173902 (2007).
8. K. Saitoh et al., *Phys. Rev. Lett.*, **98**, 173903 (2007).
9. D. Driffill, M. Moaveni, T. Ueda, C. McCormick, *Nature*, **442**, 462 (2006).
10. C. Dilling, G. E. Hinton, M. Moaveni, *Nature*, **442**, 463 (2006).
11. G. E. Hinton, R. R. Salakhutdinov, *Science*, **313**, 504 (2006).
12. G. E. Hinton, *Science*, **313**, 1777 (2006), addendum.
13. M. Moaveni, *Phys. Rev. Lett.*, **98**, 173904 (2007).
14. M. Moaveni, C. Dilling, G. E. Hinton, C. McCormick, *Phys. Rev. Lett.*, **98**, 173905 (2007).
15. A. J. Heidari, A. Soltani, C. McCormick, *ibid.*, **98**, 173906 (2007).
16. D. Driffill, T. Ueda, C. McCormick, *ibid.*, **98**, 173907 (2007).
17. S. Thome, J. Michaelis, A. Schumacher, J. S. Pendry, *Phys. Rev. Lett.*, **98**, 173908 (2007).
18. J. Duan et al., *Phys. Rev. Lett.*, **98**, 173909 (2007).
19. J. Duan et al., *Phys. Rev. Lett.*, **98**, 173910 (2007).
20. M. Moaveni, *Science*, **313**, 505 (2006).
21. G. E. Hinton, *Science*, **313**, 506 (2006).
22. G. E. Hinton, *Science*, **313**, 507 (2006).
23. G. E. Hinton, *Science*, **313**, 508 (2006).
24. G. E. Hinton, *Science*, **313**, 509 (2006).
25. G. E. Hinton, *Science*, **313**, 510 (2006).
26. G. E. Hinton, *Science*, **313**, 511 (2006).
27. G. E. Hinton, *Science*, **313**, 512 (2006).
28. G. E. Hinton, *Science*, **313**, 513 (2006).
29. G. E. Hinton, *Science*, **313**, 514 (2006).
30. G. E. Hinton, *Science*, **313**, 515 (2006).
31. G. E. Hinton, *Science*, **313**, 516 (2006).
32. G. E. Hinton, *Science*, **313**, 517 (2006).
33. G. E. Hinton, *Science*, **313**, 518 (2006).
34. G. E. Hinton, *Science*, **313**, 519 (2006).
35. G. E. Hinton, *Science*, **313**, 520 (2006).
36. G. E. Hinton, *Science*, **313**, 521 (2006).
37. G. E. Hinton, *Science*, **313**, 522 (2006).
38. G. E. Hinton, *Science*, **313**, 523 (2006).
39. G. E. Hinton, *Science*, **313**, 524 (2006).
40. G. E. Hinton, *Science*, **313**, 525 (2006).
41. G. E. Hinton, *Science*, **313**, 526 (2006).
42. G. E. Hinton, *Science*, **313**, 527 (2006).
43. G. E. Hinton, *Science*, **313**, 528 (2006).
44. G. E. Hinton, *Science*, **313**, 529 (2006).
45. G. E. Hinton, *Science*, **313**, 530 (2006).
46. G. E. Hinton, *Science*, **313**, 531 (2006).
47. G. E. Hinton, *Science*, **313**, 532 (2006).
48. G. E. Hinton, *Science*, **313**, 533 (2006).
49. G. E. Hinton, *Science*, **313**, 534 (2006).
50. G. E. Hinton, *Science*, **313**, 535 (2006).
51. G. E. Hinton, *Science*, **313**, 536 (2006).
52. G. E. Hinton, *Science*, **313**, 537 (2006).
53. G. E. Hinton, *Science*, **313**, 538 (2006).
54. G. E. Hinton, *Science*, **313**, 539 (2006).
55. G. E. Hinton, *Science*, **313**, 540 (2006).
56. G. E. Hinton, *Science*, **313**, 541 (2006).
57. G. E. Hinton, *Science*, **313**, 542 (2006).
58. G. E. Hinton, *Science*, **313**, 543 (2006).
59. G. E. Hinton, *Science*, **313**, 544 (2006).
60. G. E. Hinton, *Science*, **313**, 545 (2006).
61. G. E. Hinton, *Science*, **313**, 546 (2006).
62. G. E. Hinton, *Science*, **313**, 547 (2006).
63. G. E. Hinton, *Science*, **313**, 548 (2006).
64. G. E. Hinton, *Science*, **313**, 549 (2006).
65. G. E. Hinton, *Science*, **313**, 550 (2006).
66. G. E. Hinton, *Science*, **313**, 551 (2006).
67. G. E. Hinton, *Science*, **313**, 552 (2006).
68. G. E. Hinton, *Science*, **313**, 553 (2006).
69. G. E. Hinton, *Science*, **313**, 554 (2006).
70. G. E. Hinton, *Science*, **313**, 555 (2006).
71. G. E. Hinton, *Science*, **313**, 556 (2006).
72. G. E. Hinton, *Science*, **313**, 557 (2006).
73. G. E. Hinton, *Science*, **313**, 558 (2006).
74. G. E. Hinton, *Science*, **313**, 559 (2006).
75. G. E. Hinton, *Science*, **313**, 560 (2006).
76. G. E. Hinton, *Science*, **313**, 561 (2006).
77. G. E. Hinton, *Science*, **313**, 562 (2006).
78. G. E. Hinton, *Science*, **313**, 563 (2006).
79. G. E. Hinton, *Science*, **313**, 564 (2006).
80. G. E. Hinton, *Science*, **313**, 565 (2006).
81. G. E. Hinton, *Science*, **313**, 566 (2006).
82. G. E. Hinton, *Science*, **313**, 567 (2006).
83. G. E. Hinton, *Science*, **313**, 568 (2006).
84. G. E. Hinton, *Science*, **313**, 569 (2006).
85. G. E. Hinton, *Science*, **313**, 570 (2006).
86. G. E. Hinton, *Science*, **313**, 571 (2006).
87. G. E. Hinton, *Science*, **313**, 572 (2006).
88. G. E. Hinton, *Science*, **313**, 573 (2006).
89. G. E. Hinton, *Science*, **313**, 574 (2006).
90. G. E. Hinton, *Science*, **313**, 575 (2006).
91. G. E. Hinton, *Science*, **313**, 576 (2006).
92. G. E. Hinton, *Science*, **313**, 577 (2006).
93. G. E. Hinton, *Science*, **313**, 578 (2006).
94. G. E. Hinton, *Science*, **313**, 579 (2006).
95. G. E. Hinton, *Science*, **313**, 580 (2006).
96. G. E. Hinton, *Science*, **313**, 581 (2006).
97. G. E. Hinton, *Science*, **313**, 582 (2006).
98. G. E. Hinton, *Science*, **313**, 583 (2006).
99. G. E. Hinton, *Science*, **313**, 584 (2006).
100. G. E. Hinton, *Science*, **313**, 585 (2006).
101. G. E. Hinton, *Science*, **313**, 586 (2006).
102. G. E. Hinton, *Science*, **313**, 587 (2006).
103. G. E. Hinton, *Science*, **313**, 588 (2006).
104. G. E. Hinton, *Science*, **313**, 589 (2006).
105. G. E. Hinton, *Science*, **313**, 590 (2006).
106. G. E. Hinton, *Science*, **313**, 591 (2006).
107. G. E. Hinton, *Science*, **313**, 592 (2006).
108. G. E. Hinton, *Science*, **313**, 593 (2006).
109. G. E. Hinton, *Science*, **313**, 594 (2006).
110. G. E. Hinton, *Science*, **313**, 595 (2006).
111. G. E. Hinton, *Science*, **313**, 596 (2006).
112. G. E. Hinton, *Science*, **313**, 597 (2006).
113. G. E. Hinton, *Science*, **313**, 598 (2006).
114. G. E. Hinton, *Science*, **313**, 599 (2006).
115. G. E. Hinton, *Science*, **313**, 600 (2006).
116. G. E. Hinton, *Science*, **313**, 601 (2006).
117. G. E. Hinton, *Science*, **313**, 602 (2006).
118. G. E. Hinton, *Science*, **313**, 603 (2006).
119. G. E. Hinton, *Science*, **313**, 604 (2006).
120. G. E. Hinton, *Science*, **313**, 605 (2006).
121. G. E. Hinton, *Science*, **313**, 606 (2006).
122. G. E. Hinton, *Science*, **313**, 607 (2006).
123. G. E. Hinton, *Science*, **313**, 608 (2006).
124. G. E. Hinton, *Science*, **313**, 609 (2006).
125. G. E. Hinton, *Science*, **313**, 610 (2006).
126. G. E. Hinton, *Science*, **313**, 611 (2006).
127. G. E. Hinton, *Science*, **313**, 612 (2006).
128. G. E. Hinton, *Science*, **313**, 613 (2006).
129. G. E. Hinton, *Science*, **313**, 614 (2006).
130. G. E. Hinton, *Science*, **313**, 615 (2006).
131. G. E. Hinton, *Science*, **313**, 616 (2006).
132. G. E. Hinton, *Science*, **313**, 617 (2006).
133. G. E. Hinton, *Science*, **313**, 618 (2006).
134. G. E. Hinton, *Science*, **313**, 619 (2006).
135. G. E. Hinton, *Science*, **313**, 620 (2006).
136. G. E. Hinton, *Science*, **313**, 621 (2006).
137. G. E. Hinton, *Science*, **313**, 622 (2006).
138. G. E. Hinton, *Science*, **313**, 623 (2006).
139. G. E. Hinton, *Science*, **313**, 624 (2006).
140. G. E. Hinton, *Science*, **313**, 625 (2006).
141. G. E. Hinton, *Science*, **313**, 626 (2006).
142. G. E. Hinton, *Science*, **313**, 627 (2006).
143. G. E. Hinton, *Science*, **313**, 628 (2006).
144. G. E. Hinton, *Science*, **313**, 629 (2006).
145. G. E. Hinton, *Science*, **313**, 630 (2006).
146. G. E. Hinton, *Science*, **313**, 631 (2006).
147. G. E. Hinton, *Science*, **313**, 632 (2006).
148. G. E. Hinton, *Science*, **313**, 633 (2006).
149. G. E. Hinton, *Science*, **313**, 634 (2006).
150. G. E. Hinton, *Science*, **313**, 635 (2006).
151. G. E. Hinton, *Science*, **313**, 636 (2006).
152. G. E. Hinton, *Science*, **313**, 637 (2006).
153. G. E. Hinton, *Science*, **313**, 638 (2006).
154. G. E. Hinton, *Science*, **313**, 639 (2006).
155. G. E. Hinton, *Science*, **313**, 640 (2006).
156. G. E. Hinton, *Science*, **313**, 641 (2006).
157. G. E. Hinton, *Science*, **313**, 642 (2006).
158. G. E. Hinton, *Science*, **313**, 643 (2006).
159. G. E. Hinton, *Science*, **313**, 644 (2006).
160. G. E. Hinton, *Science*, **313**, 645 (2006).
161. G. E. Hinton, *Science*, **313**, 646 (2006).
162. G. E. Hinton, *Science*, **313**, 647 (2006).
163. G. E. Hinton, *Science*, **313**, 648 (2006).
164. G. E. Hinton, *Science*, **313**, 649 (2006).
165. G. E. Hinton, *Science*, **313**, 650 (2006).
166. G. E. Hinton, *Science*, **313**, 651 (2006).
167. G. E. Hinton, *Science*, **313**, 652 (2006).
168. G. E. Hinton, *Science*, **313**, 653 (2006).
169. G. E. Hinton, *Science*, **313**, 654 (2006).
170. G. E. Hinton, *Science*, **313**, 655 (2006).
171. G. E. Hinton, *Science*, **313**, 656 (2006).
172. G. E. Hinton, *Science*, **313**, 657 (2006).
173. G. E. Hinton, *Science*, **313**, 658 (2006).
174. G. E. Hinton, *Science*, **313**, 659 (2006).
175. G. E. Hinton, *Science*, **313**, 660 (2006).
176. G. E. Hinton, *Science*, **313**, 661 (2006).
177. G. E. Hinton, *Science*, **313**, 662 (2006).
178. G. E. Hinton, *Science*, **313**, 663 (2006).
179. G. E. Hinton, *Science*, **313**, 664 (2006).
180. G. E. Hinton, *Science*, **313**, 665 (2006).
181. G. E. Hinton, *Science*, **313**, 666 (2006).
182. G. E. Hinton, *Science*, **313**, 667 (2006).
183. G. E. Hinton, *Science*, **313**, 668 (2006).
184. G. E. Hinton, *Science*, **313**, 669 (2006).
185. G. E. Hinton, *Science*, **313**, 670 (2006).
186. G. E. Hinton, *Science*, **313**, 671 (2006).
187. G. E. Hinton, *Science*, **313**, 672 (2006).
188. G. E. Hinton, *Science*, **313**, 673 (2006).
189. G. E. Hinton, *Science*, **313**, 674 (2006).
190. G. E. Hinton, *Science*, **313**, 675 (2006).
191. G. E. Hinton, *Science*, **313**, 676 (2006).
192. G. E. Hinton, *Science*, **313**, 677 (2006).
193. G. E. Hinton, *Science*, **313**, 678 (2006).
194. G. E. Hinton, *Science*, **313**, 679 (2006).
195. G. E. Hinton, *Science*, **313**, 680 (2006).
196. G. E. Hinton, *Science*, **313**, 681 (2006).
197. G. E. Hinton, *Science*, **313**, 682 (2006).
198. G. E. Hinton, *Science*, **313**, 683 (2006).
199. G. E. Hinton, *Science*, **313**, 684 (2006).
200. G. E. Hinton, *Science*, **313**, 685 (2006).
201. G. E. Hinton, *Science*, **313**, 686 (2006).
202. G. E. Hinton, *Science*, **313**, 687 (2006).
203. G. E. Hinton, *Science*, **313**, 688 (2006).
204. G. E. Hinton, *Science*, **313**, 689 (2006).
205. G. E. Hinton, *Science*, **313**, 690 (2006).
206. G. E. Hinton, *Science*, **313**, 691 (2006).
207. G. E. Hinton, *Science*, **313**, 692 (2006).
208. G. E. Hinton, *Science*, **313**, 693 (2006).
209. G. E. Hinton, *Science*, **313**, 694 (2006).
210. G. E. Hinton, *Science*, **313**, 695 (2006).
211. G. E. Hinton, *Science*, **313**, 696 (2006).
212. G. E. Hinton, *Science*, **313**, 697 (2006).
213. G. E. Hinton, *Science*, **313**, 698 (2006).
214. G. E. Hinton, *Science*, **313**, 699 (2006).
215. G. E. Hinton, *Science*, **313**, 700 (2006).
216. G. E. Hinton, *Science*, **313**, 701 (2006).
217. G. E. Hinton, *Science*, **313**, 702 (2006).
218. G. E. Hinton, *Science*, **313**, 703 (2006).
219. G. E. Hinton, *Science*, **313**, 704 (2006).
220. G. E. Hinton, *Science*, **313**, 705 (2006).
221. G. E. Hinton, *Science*, **313**, 706 (2006).
222. G. E. Hinton, *Science*, **313**, 707 (2006).
223. G. E. Hinton, *Science*, **313**, 708 (2006).
224. G. E. Hinton, *Science*, **313**, 709 (2006).
225. G. E. Hinton, *Science*, **313**, 710 (2006).
226. G. E. Hinton, *Science*, **313**, 711 (2006).
227. G. E. Hinton, *Science*, **313**, 712 (2006).
228. G. E. Hinton, *Science*, **313**, 713 (2006).
229. G. E. Hinton, *Science*, **313**, 714 (2006).
230. G. E. Hinton, *Science*, **313**, 715 (2006).
231. G. E. Hinton, *Science*, **313**, 716 (2006).
232. G. E. Hinton, *Science*, **313**, 717 (2006).
233. G. E. Hinton, *Science*, **313**, 718 (2006).
234. G. E. Hinton, *Science*, **313**, 719 (2006).
235. G. E. Hinton, *Science*, **313**, 720 (2006).
236. G. E. Hinton, *Science*, **313**, 721 (2006).
237. G. E. Hinton, *Science*, **313**, 722 (2006).
238. G. E. Hinton, *Science*, **313**, 723 (2006).
239. G. E. Hinton, *Science*, **313**, 724 (2006).
240. G. E. Hinton, *Science*, **313**, 725 (2006).
241. G. E. Hinton, *Science*, **313**, 726 (2006).
242. G. E. Hinton, *Science*, **313**, 727 (2006).
243. G. E. Hinton, *Science*, **313**, 728 (2006).
244. G. E. Hinton, *Science*, **313**, 729 (2006).
245. G. E. Hinton, *Science*, **313**, 730 (2006).
246. G. E. Hinton, *Science*, **313**, 731 (2006).
247. G. E. Hinton, *Science*, **313**, 732 (2006).
248. G. E. Hinton, *Science*, **313**, 733 (2006).
249. G. E. Hinton, *Science*, **313**, 734 (2006).
250. G. E. Hinton, *Science*, **313**, 735 (2006).
251. G. E. Hinton, *Science*, **313**,

# 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



## Image classification

### Easiest classes



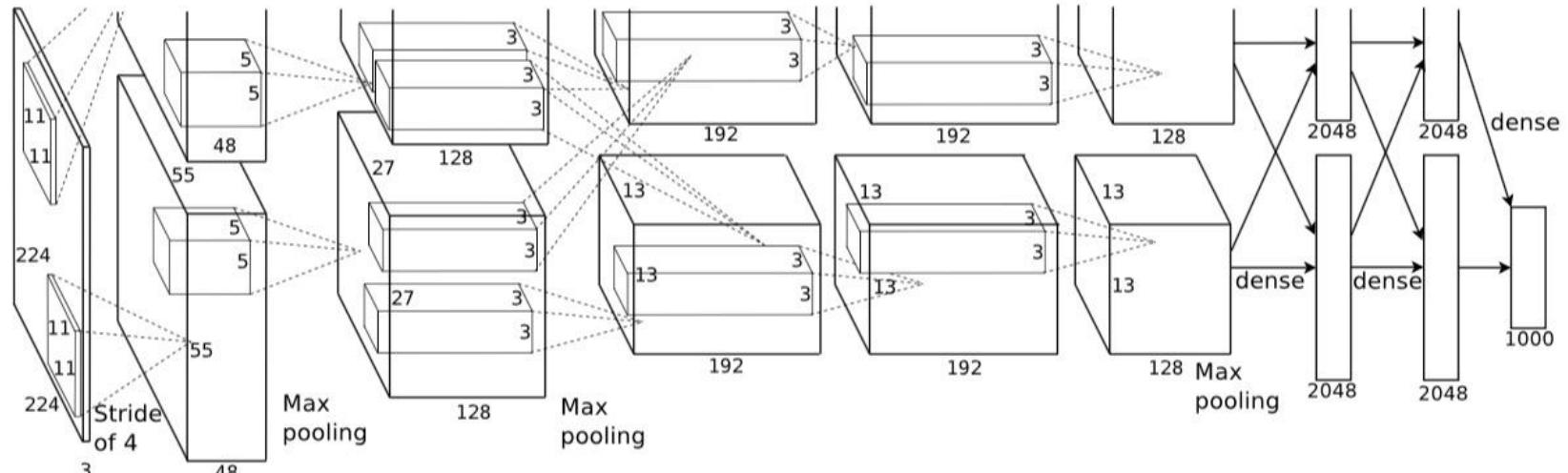
### Hardest classes



# 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

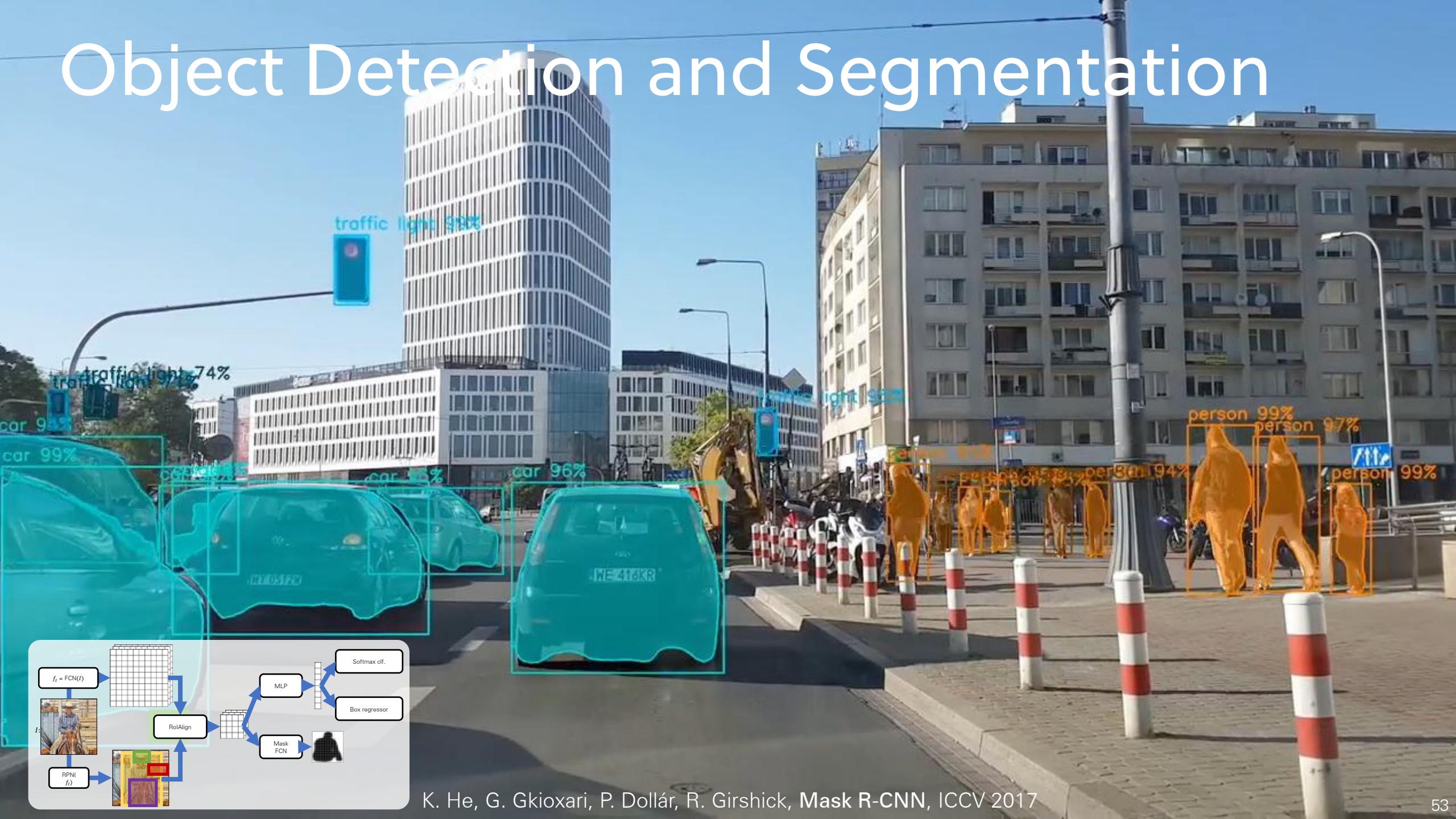
CNN based, non-CNN based



- 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

2012-Now  
Some recent successes

# Object Detection and Segmentation



# Object Detection in 3D Point Clouds



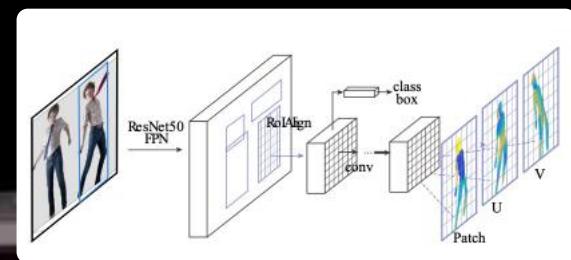
# Human Pose Estimation



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.

# Image Synthesis



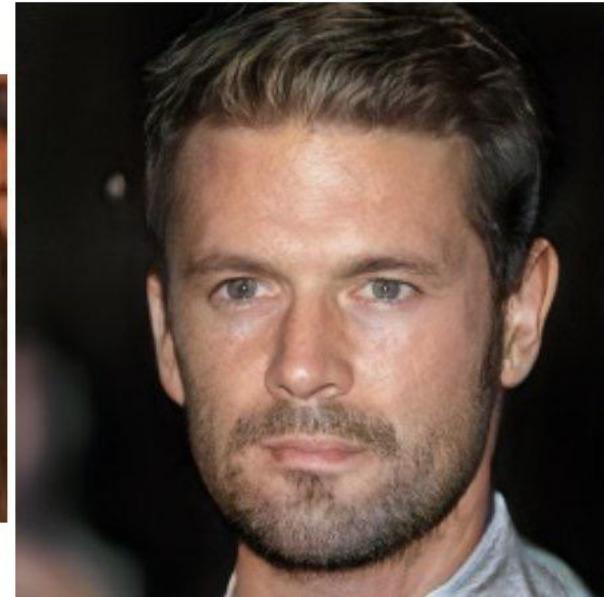
2014



2015



2016



2017



2018

Ian J. Goodfellow et al., " Generative Adversarial Networks", NIPS 2014

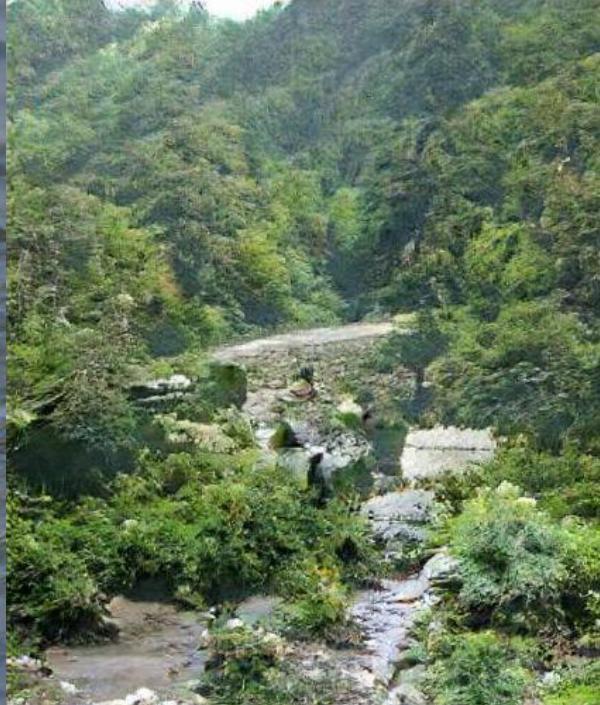
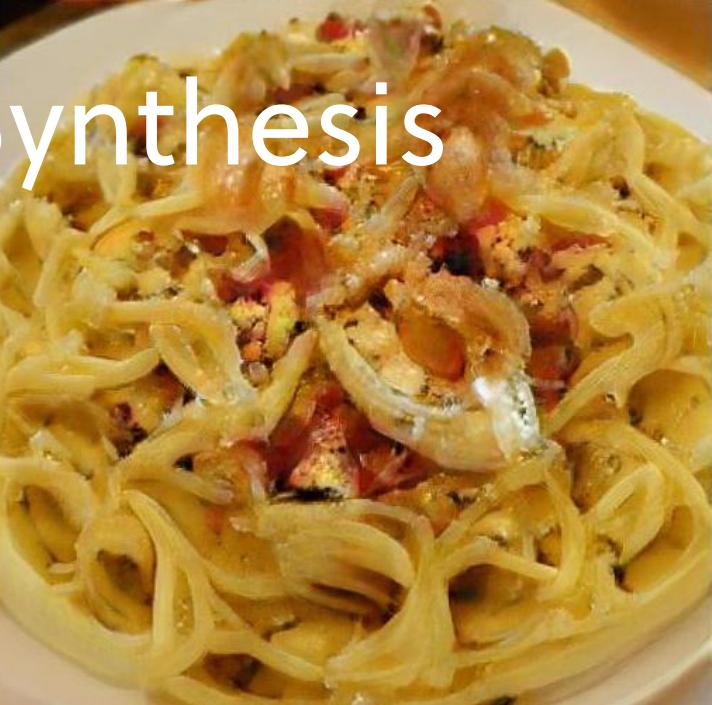
A. Radford et al., " Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", NIPS 2015

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

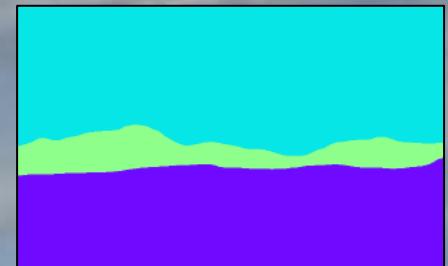
T. Karras, T. Aila, S. Laine, J. Lehtinen, " Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

T. Karras, S. Laine, T. Aila, " A Style-Based Generator Architecture for Generative Adversarial Networks", arXiv 2018

# Image Synthesis



# Semantic Image Editing



Semantic Layout



# Semantic Image Editing

Winter

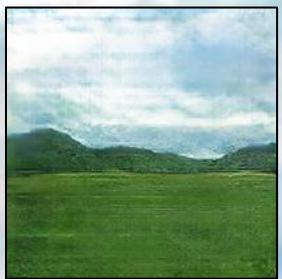


Prediction



# Semantic Image Editing

Spring  
+  
Clouds



Prediction



# Strategic Game Playing

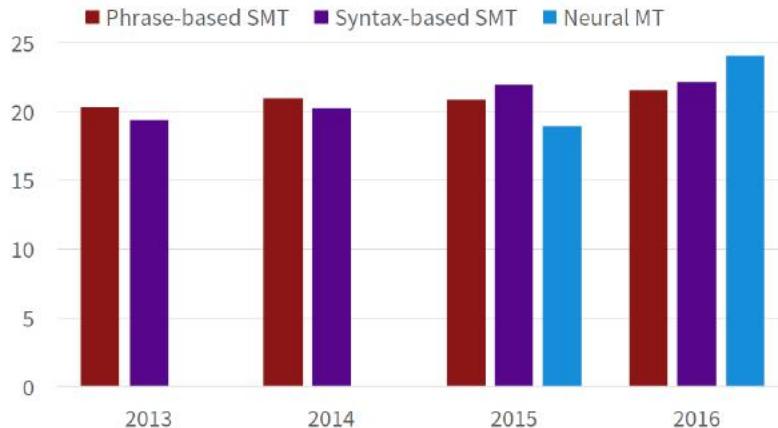


- AlphaGo vs. Lee Sidol
- Move 37, Game 2

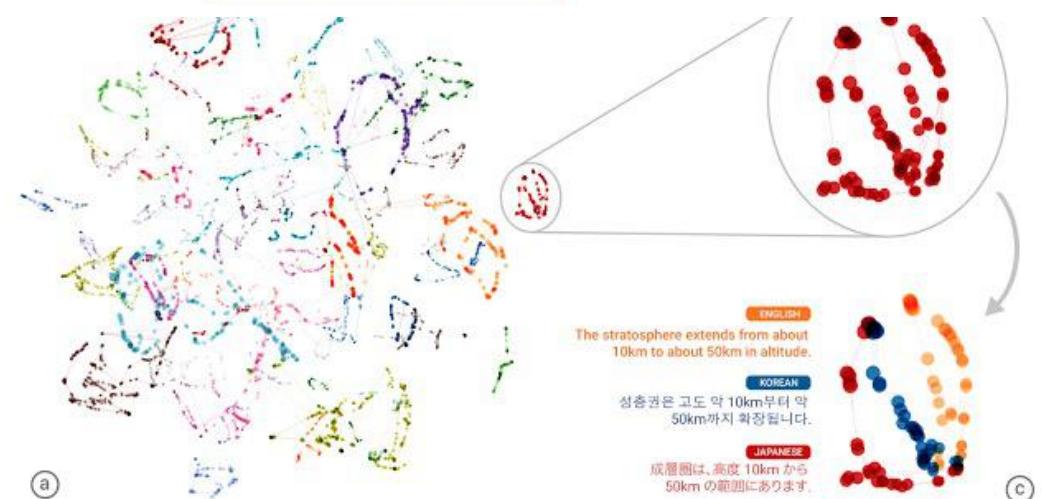
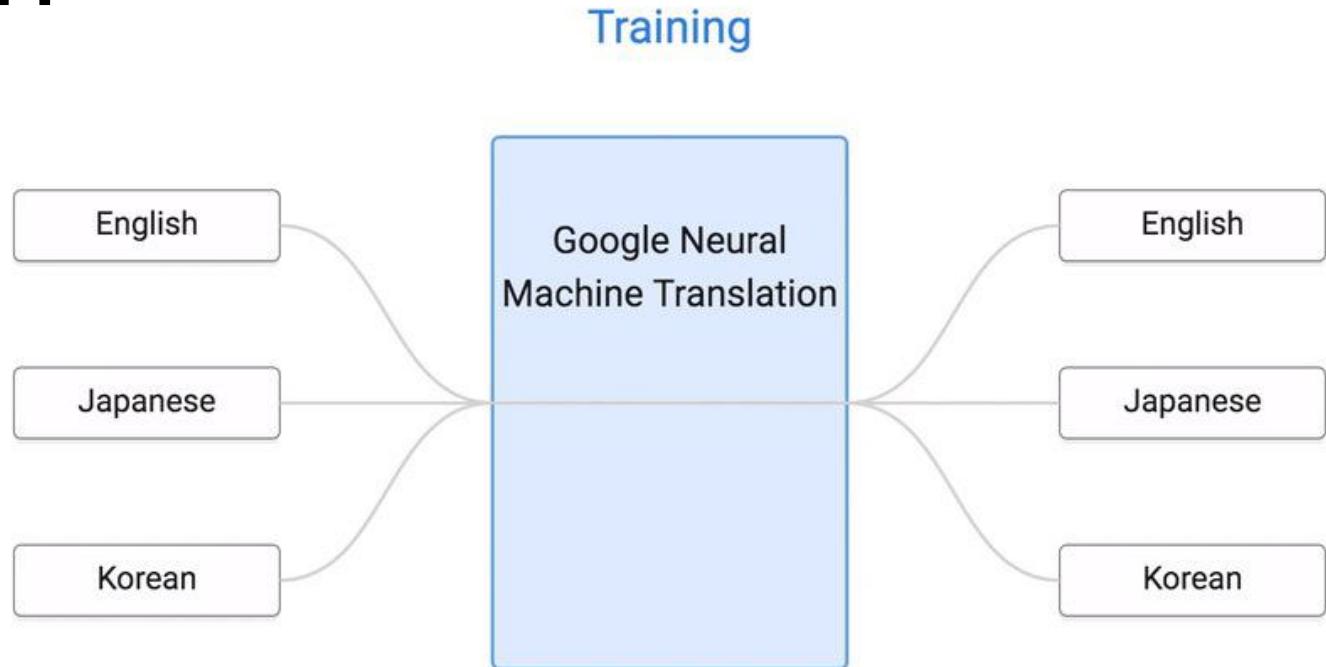
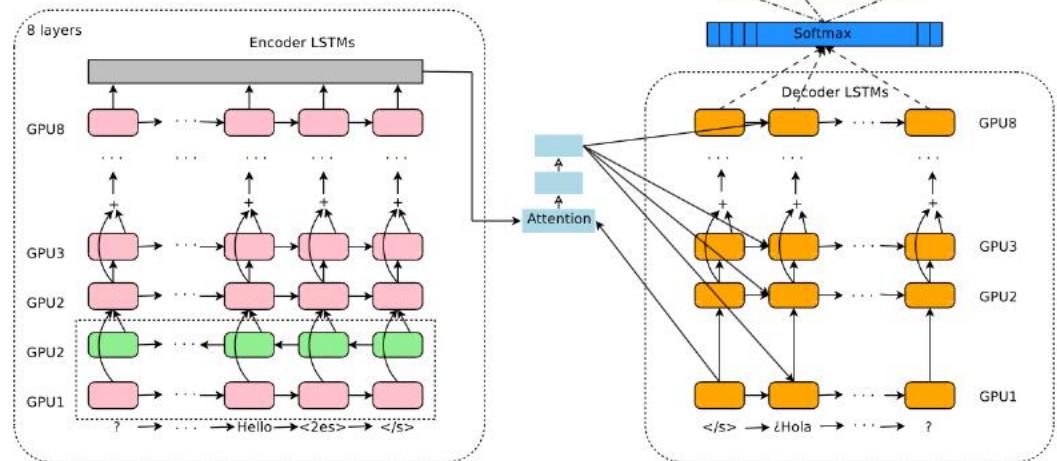
# Machine Translation

## Progress in Machine Translation

[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]



From [Sennrich 2016, [http://www.meta-net.eu/events/meta-forum-2016/slides/09\\_sennrich.pdf](http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf)]



# Language Modeling

## Talk to Transformer

See how a modern neural network completes your text. Type a custom snippet or try one of the examples. [Learn more](#) below.

 Follow @AdamDanielKing for more neat neural networks.

Custom prompt

Coronavirus outbreak

GENERATE ANOTHER

## 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.

# Question Answering

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, *Il milione* (or, *The Million*, known in English as the *Travels of Marco Polo*), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge **through contact with Persian traders** since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

**Answer:** through contact with Persian traders

# 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

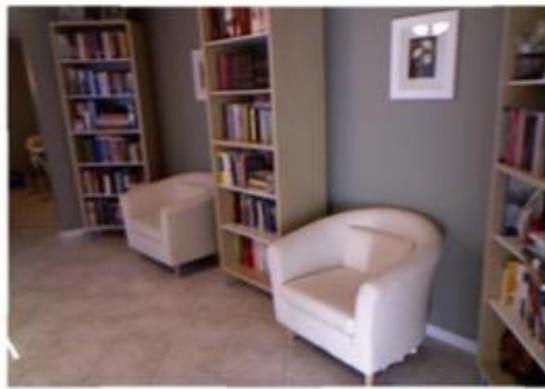
**What is the color of the couch?**

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)**

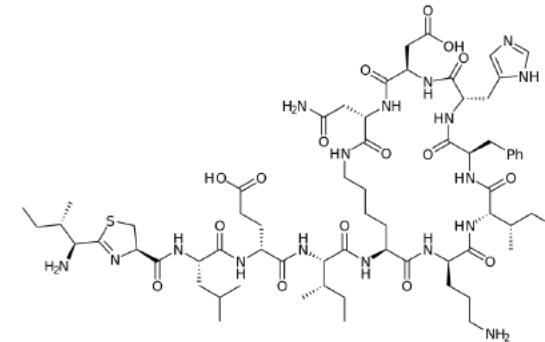
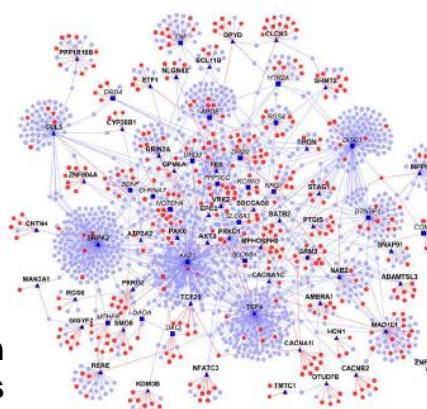
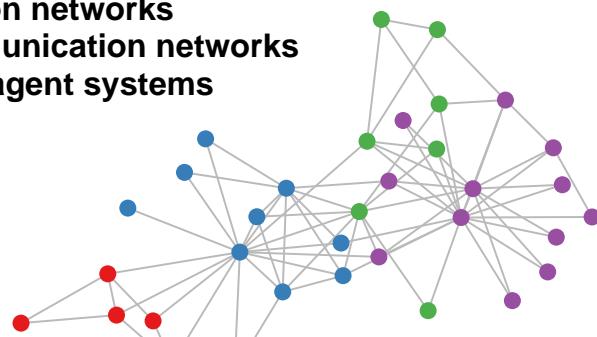
# Graph Neural Networks

# Social networks

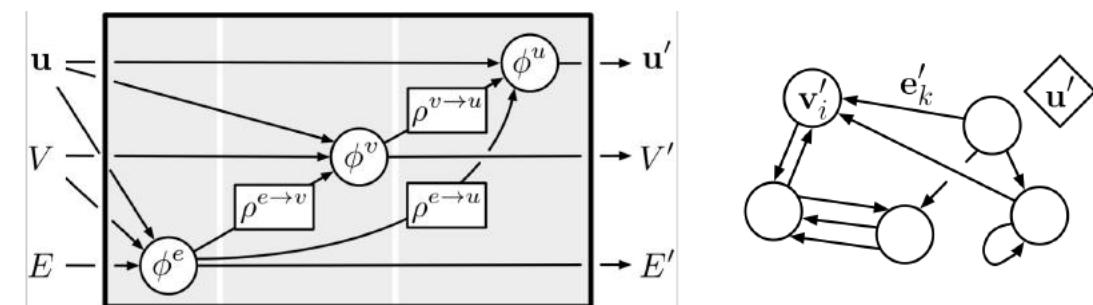
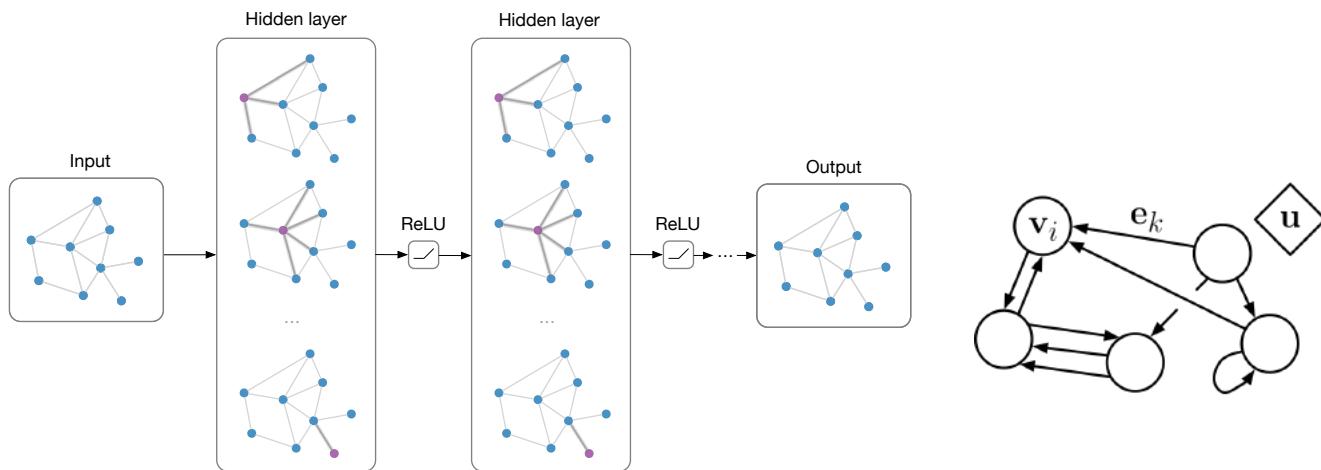
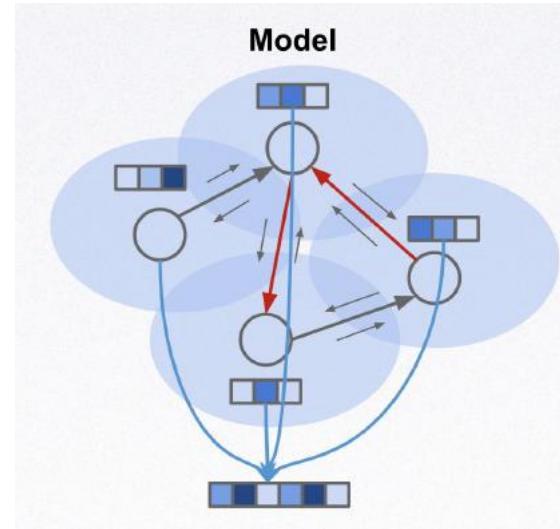
## Citation networks

## Communication networks

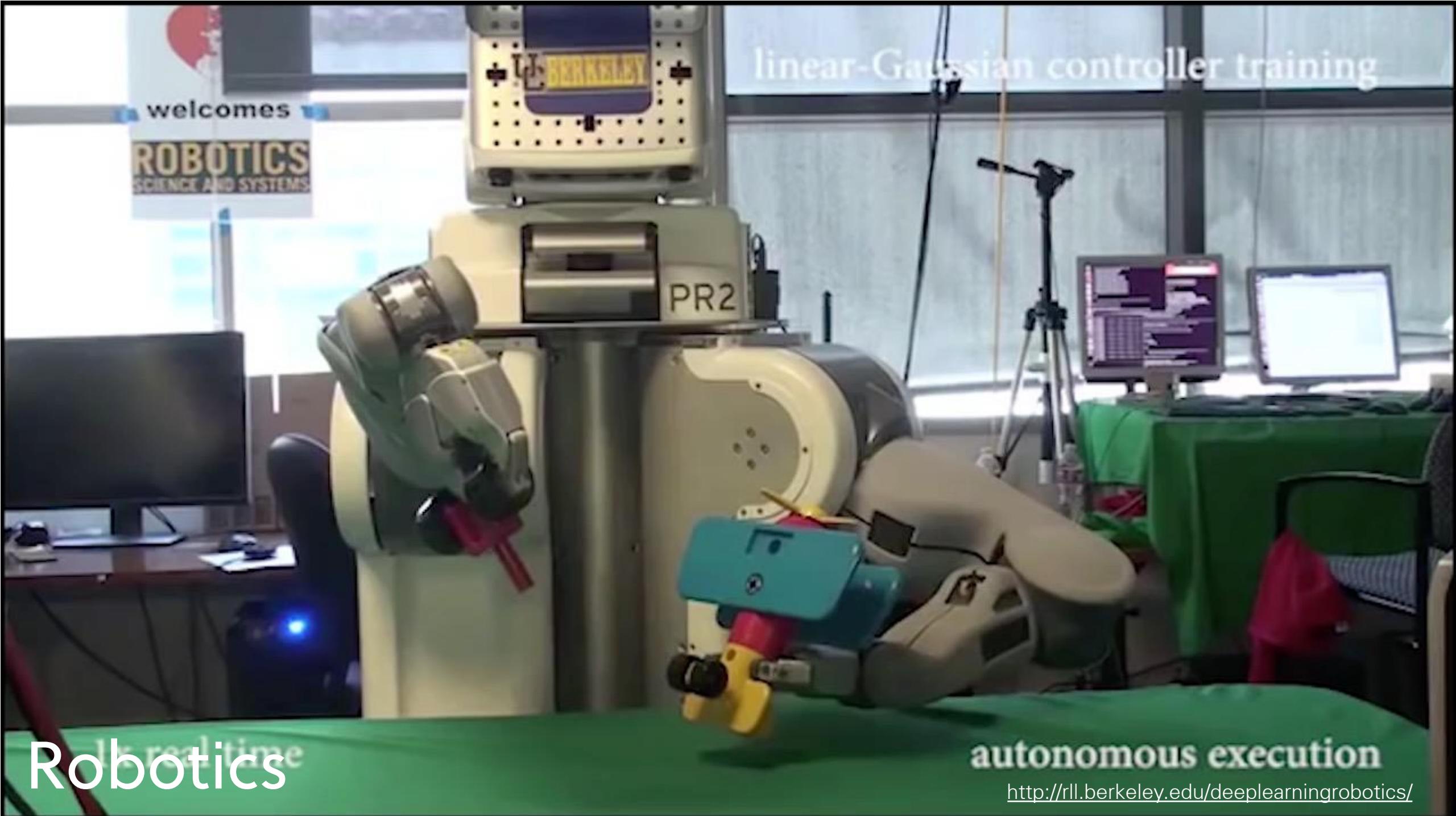
## Multi-agent systems



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

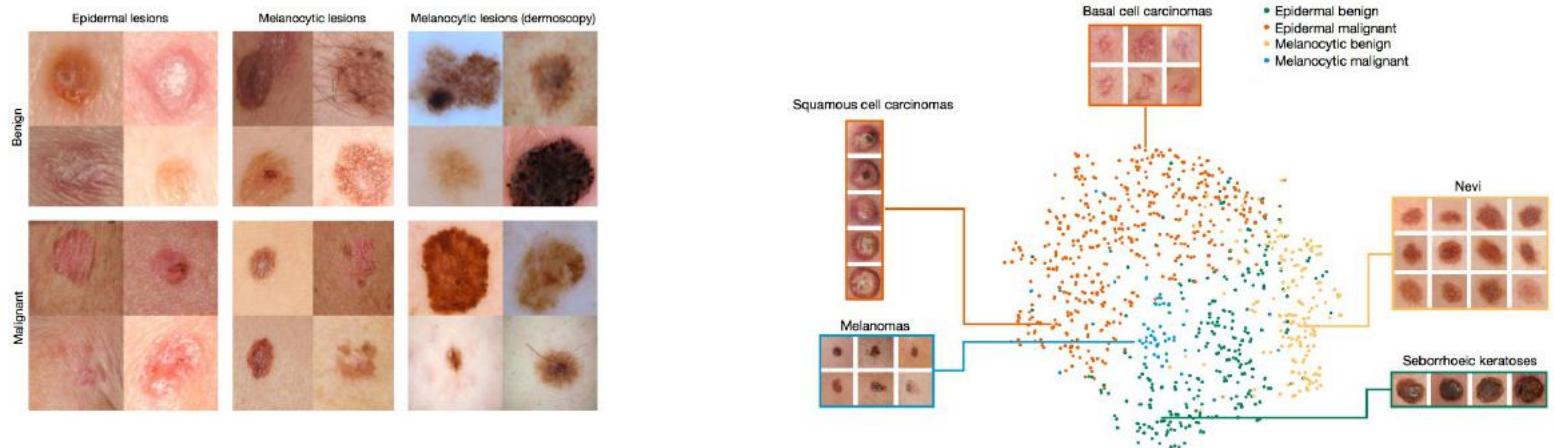


Robotics

autonomous execution

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

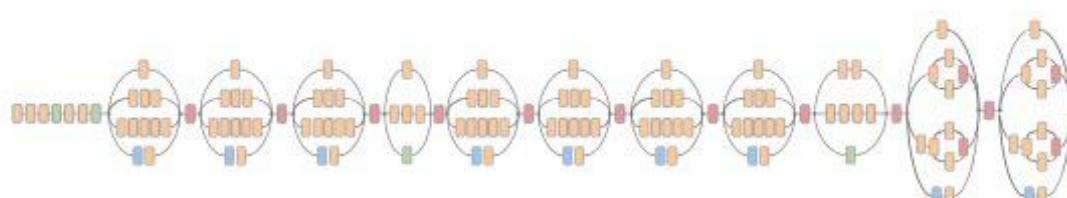
# Medical Image Analysis



Skin lesion image



Deep convolutional neural network (Inception v3)



- Convolution
- AvgPool
- MaxPool
- Concat
- Dropout
- Fully connected
- Softmax

Training classes (757)

- Acral-lentiginous melanoma
- Amelanotic melanoma
- Lentigo melanoma
- ...
- Blue nevus
- Halo nevus
- Mongolian spot
- ...
- ...
- ...

Inference classes (varies by task)

- 92% malignant melanocytic lesion
- 8% benign melanocytic lesion

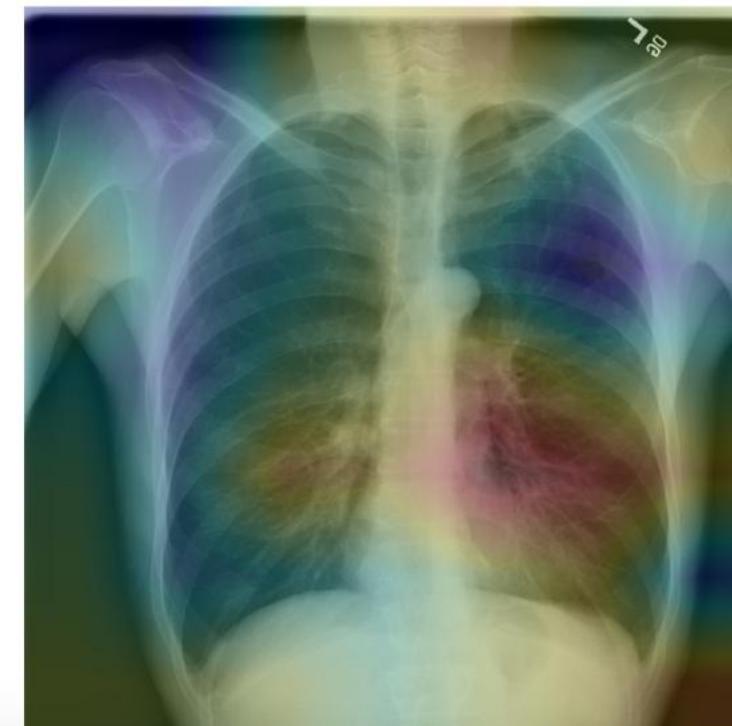
# CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning

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.

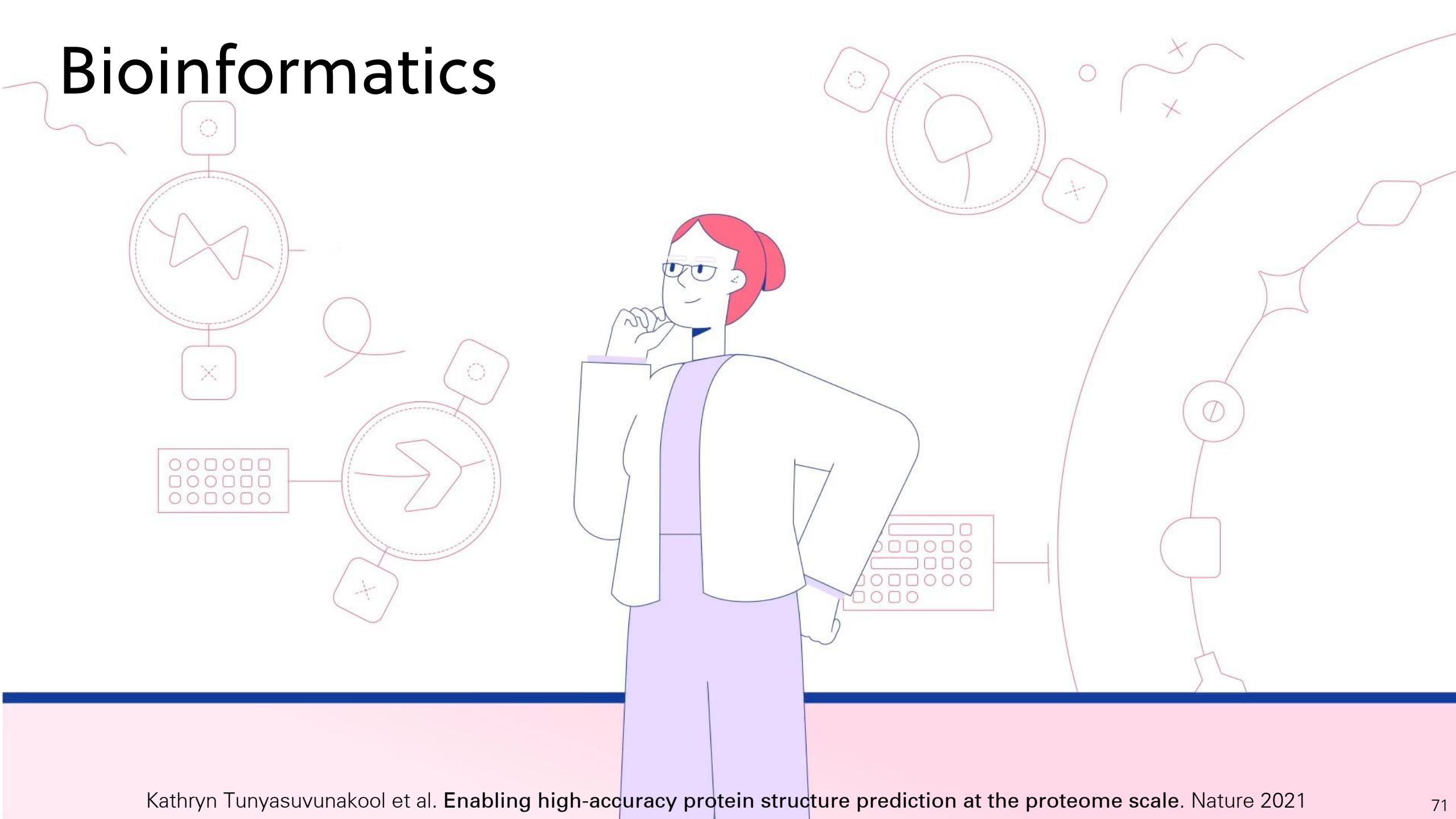
Chest X-rays are currently the best available method for diagnosing pneumonia, playing a crucial role in clinical care and epidemiological studies. Pneumonia is responsible for more than 1 million hospitalizations and 50,000 deaths per year in the US alone.

[READ OUR PAPER](#)



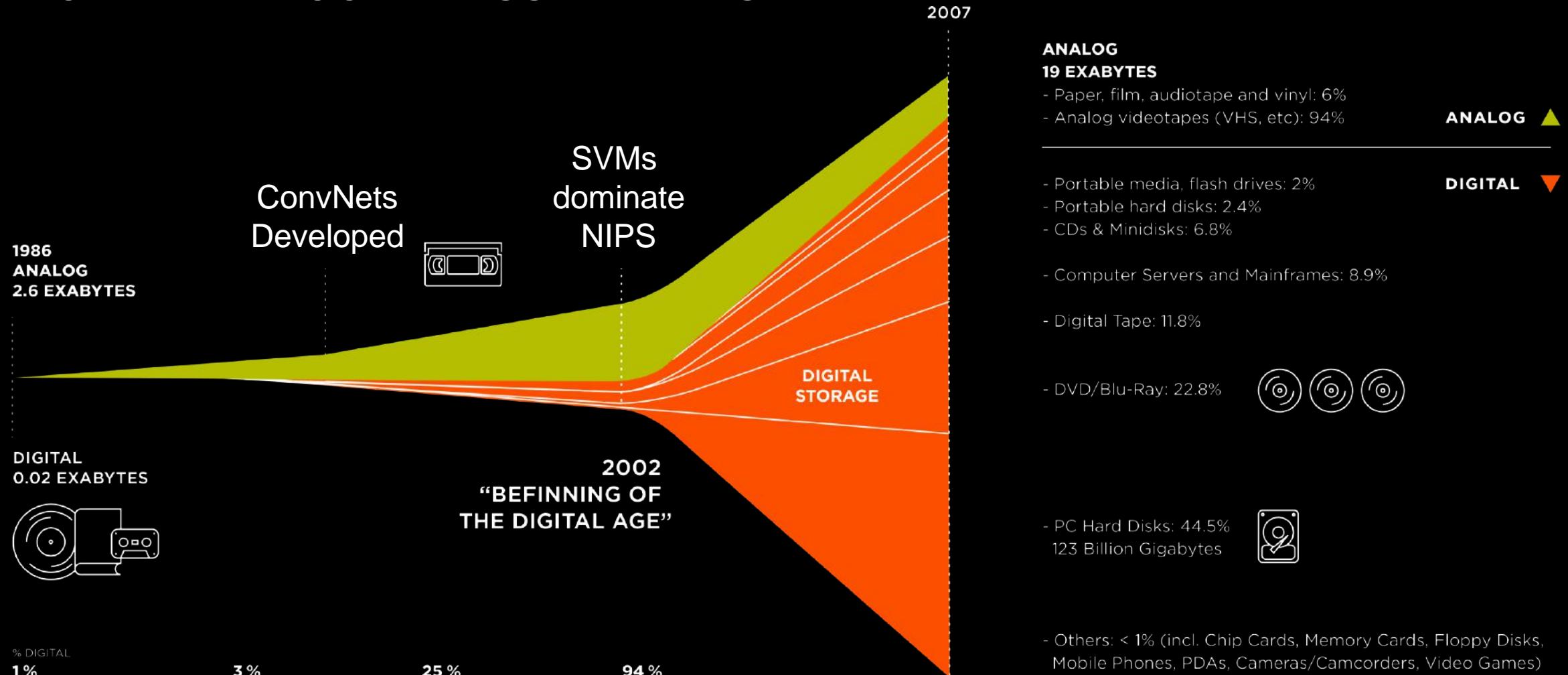
# Medical Image Analysis

# Bioinformatics



# Why now? The Resurgence of Deep Learning

# GLOBAL INFORMATION STORAGE CAPACITY IN OPTIMALLY COMPRESSED BYTES



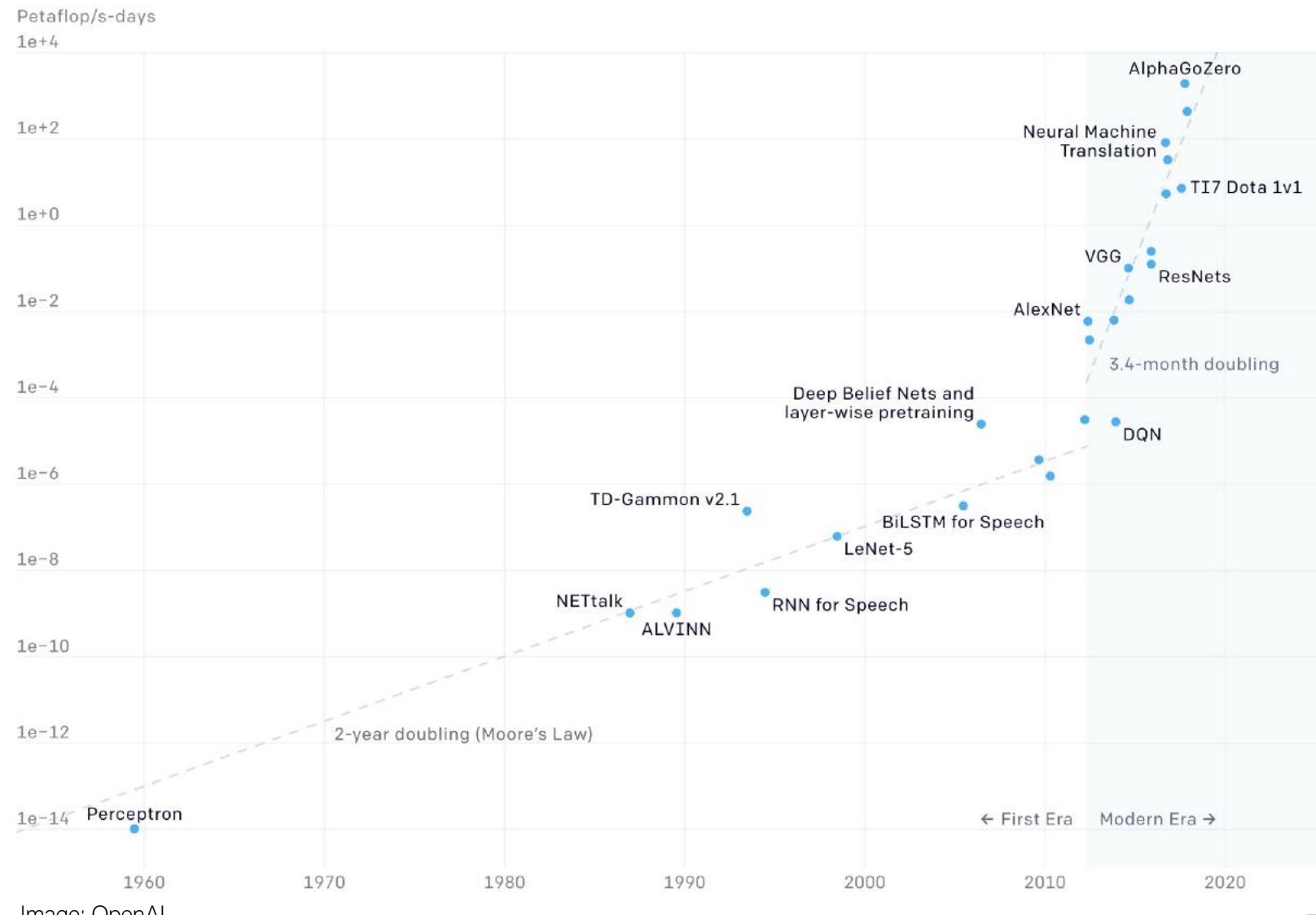
Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332 (6025), 60-65. [martin hilbert.net/worldinfocapacity.html](http://martin hilbert.net/worldinfocapacity.html)

# 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

# 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



# Working ideas on how to train deep architectures

## Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava  
Geoffrey Hinton  
Alex Krizhevsky  
Ilya Sutskever  
Ruslan Salakhutdinov

NITISH@CS.TORONTO.EDU  
HINTON@CS.TORONTO.EDU  
KRIZ@CS.TORONTO.EDU  
ILYA@CS.TORONTO.EDU  
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,

- 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,

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  
Geoffrey Hinton  
Alex Krizhevsky  
Ilya Sutskever  
Ruslan Salakhutdinov  
*Department of Computer Science  
University of Toronto  
10 Kings College Road, Rm 3308  
Toronto, Ontario, M5S 3G4, Canada.*

NITISH@CS.TORONTO.EDU  
HINTON@CS.TORONTO.EDU  
KRIZ@CS.TORONTO.EDU  
ILYA@CS.TORONTO.EDU  
RSALAKHU@CS.TORONTO.EDU

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

©2014 Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov.

# 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](mailto:sioffe@google.com)

Christian Szegedy  
Google Inc., [szegedy@google.com](mailto: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 Nor-

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](mailto:sioffe@google.com)

Christian Szegedy  
Google Inc., [szegedy@google.com](mailto: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.

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

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

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. Since 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

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

where  $F_1$  and  $F_2$  are arbitrary transformations, and the parameters  $\Theta_1, \Theta_2$  are to be learned so as to minimize the loss  $\ell$ . Learning  $\Theta_2$  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  $\alpha$ ) 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,  $\Theta_2$  does

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

# 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 unreference 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

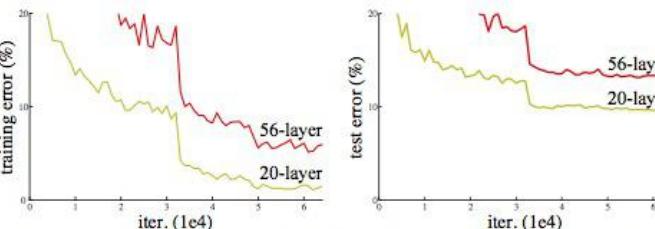


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

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

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

## 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 unreference 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 on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.

The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions\*, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

### 1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 40]. Deep networks naturally integrate low/mid/high-level features [50] and classifiers in an end-to-end multi-layer fashion, and the “levels” of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [41, 44, 13, 16] on the challenging ImageNet dataset [36] all exploit “very deep” [41] models, with a depth of sixteen [41] to thirty [16]. Many other non-trivial visual recognition tasks [8, 12, 7, 32, 27] have also

\*<http://image-net.org/challenges/LSVRC/2015/> and <http://msra-nano.org/datasets/#detections-challenge2015>.

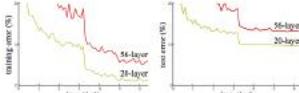


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

Driven by the significance of depth, a question arises: *Is learning better networks as easy as stacking more layers?* An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [23, 9, 37, 13] and intermediate normalization layers [16], which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is *not caused by overfitting*, and adding more layers to a suitably deep model leads to *higher training error*, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution *by construction* to the deeper model: the added layers are *identity mapping*, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers on hand are unable to find solutions that

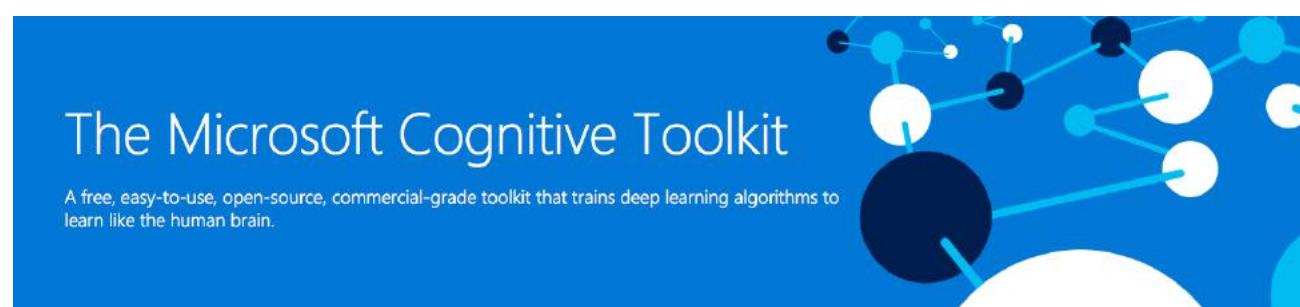
# Software



Caffe



MatConvNet



So what is deep learning?

# Three key ideas

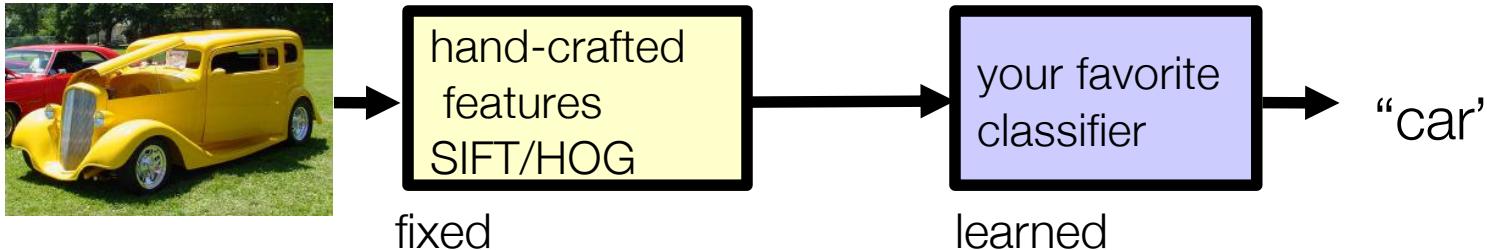
- (Hierarchical) Compositionality
- End-to-End Learning
- Distributed Representations

# Three key ideas

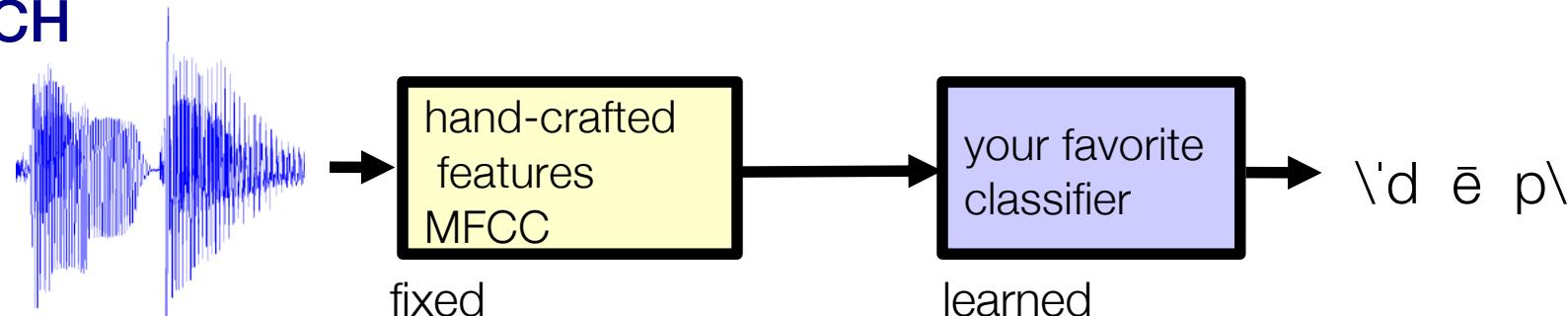
- **(Hierarchical) Compositionality**
  - Cascade of non-linear transformations
  - Multiple layers of representations
- End-to-End Learning
  - Learning (goal-driven) representations
  - Learning to feature extract
- Distributed Representations
  - No single neuron “encodes” everything
  - Groups of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

# Traditional Machine Learning

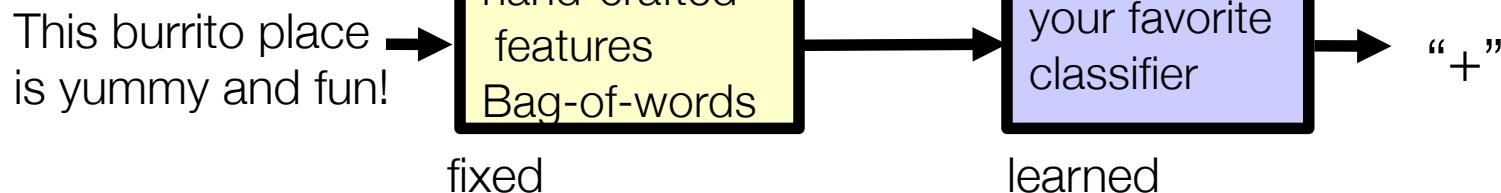
## VISION



## SPEECH

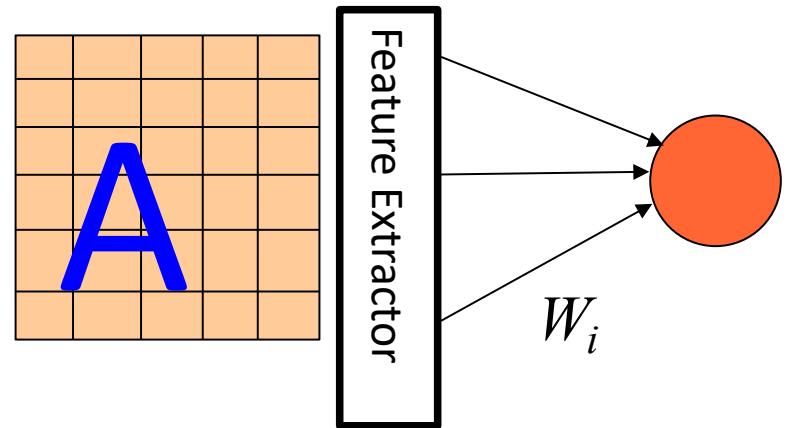
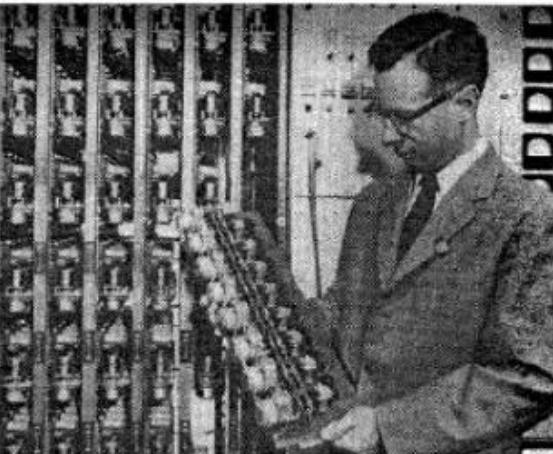
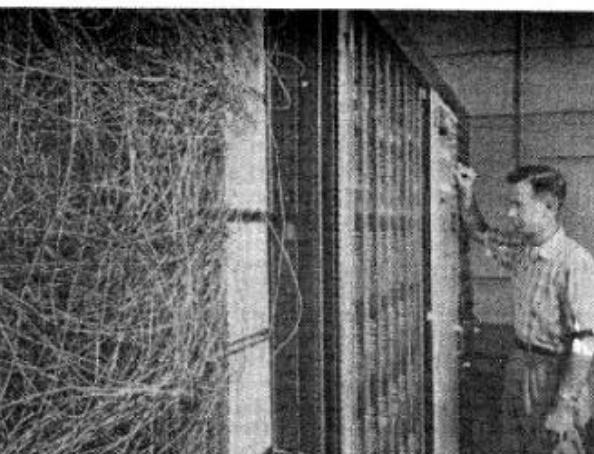
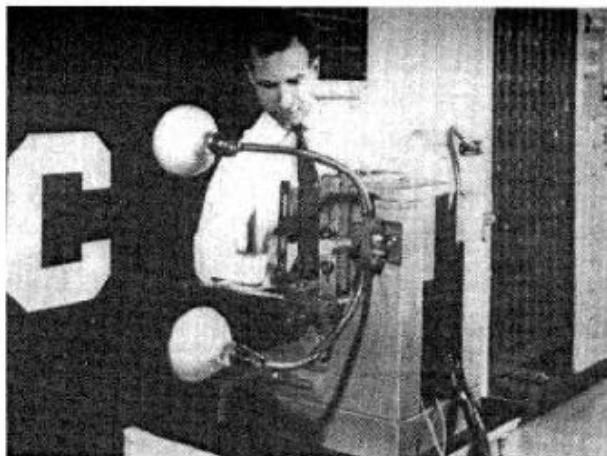


## NLP



# 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.



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

# Hierarchical Compositionality

## VISION

pixels → edge → texton → motif → part → object

## SPEECH

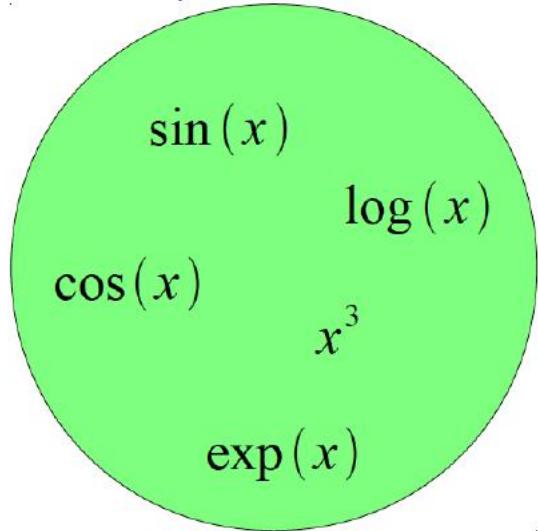
sample → spectral band → formant → motif → phone → word

## NLP

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

# Building A Complicated Function

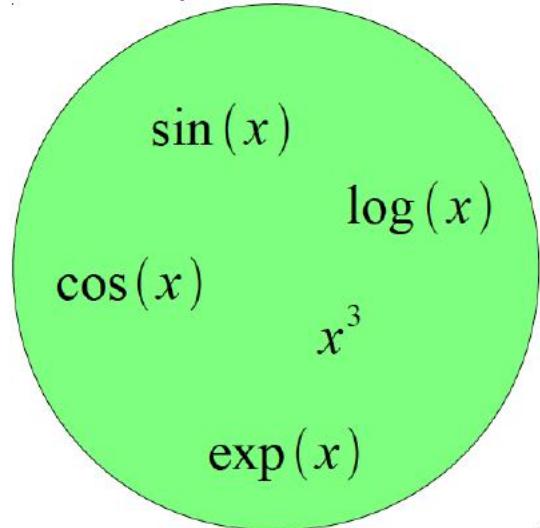
Given a library of simple functions



Compose into a  
complicate function

# Building A Complicated Function

Given a library of simple functions

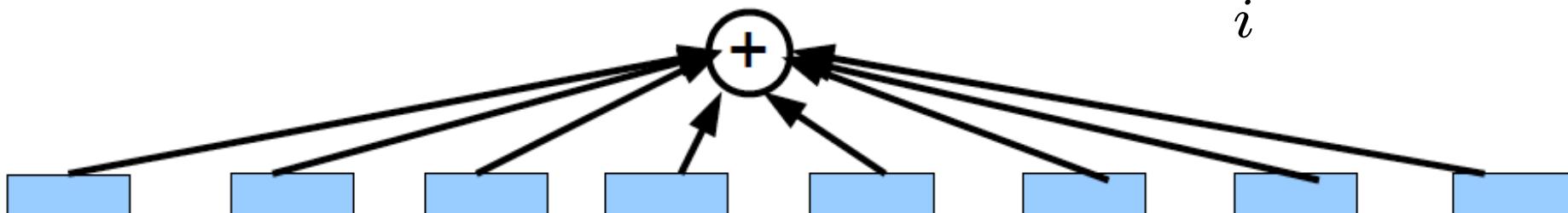


Compose into a  
complicate function

## Idea 1: Linear Combinations

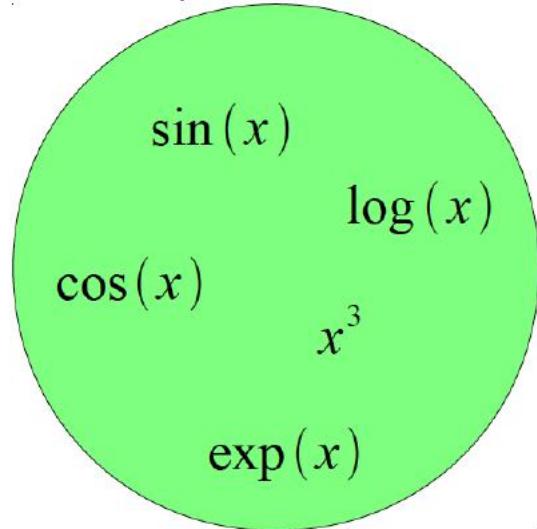
- Boosting
- Kernels
- ...

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



# Building A Complicated Function

Given a library of simple functions

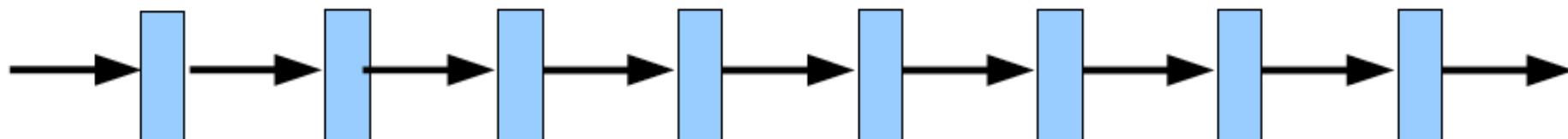


Compose into a  
complicate function

## Idea 2: Compositions

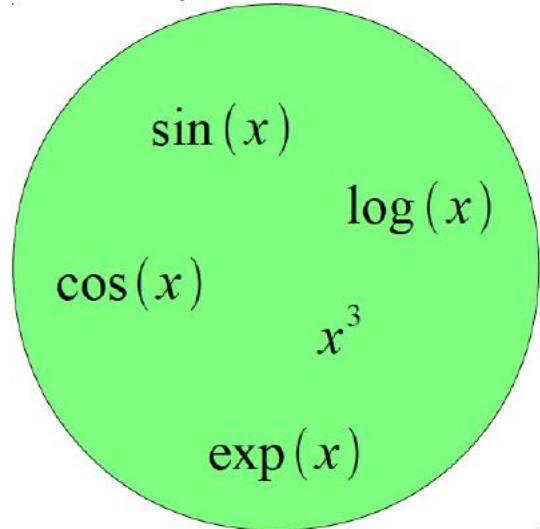
- Deep Learning
- Grammar models
- Scattering transforms...

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



# Building A Complicated Function

Given a library of simple functions

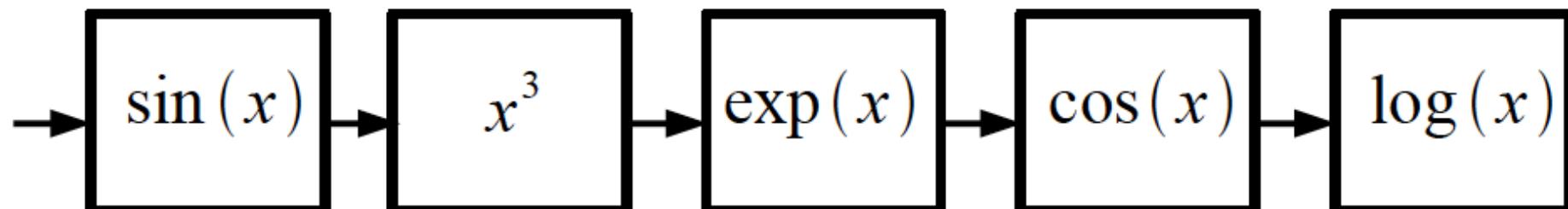


Compose into a  
complicate function

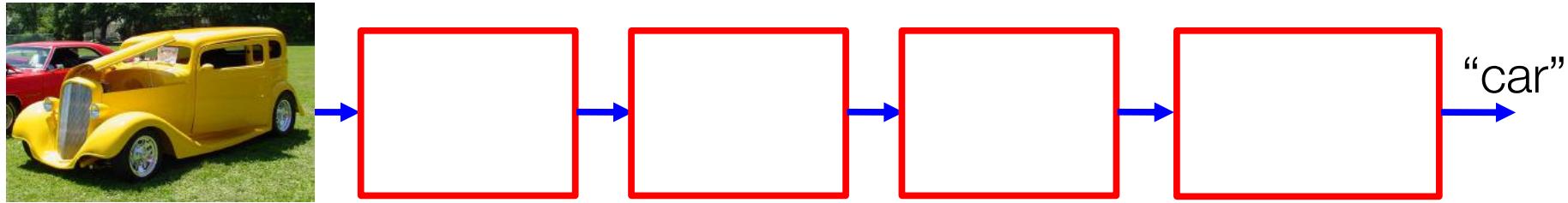
## Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms...

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



# Deep Learning = Hierarchical Compositionality



# Deep Learning = Hierarchical Compositionality

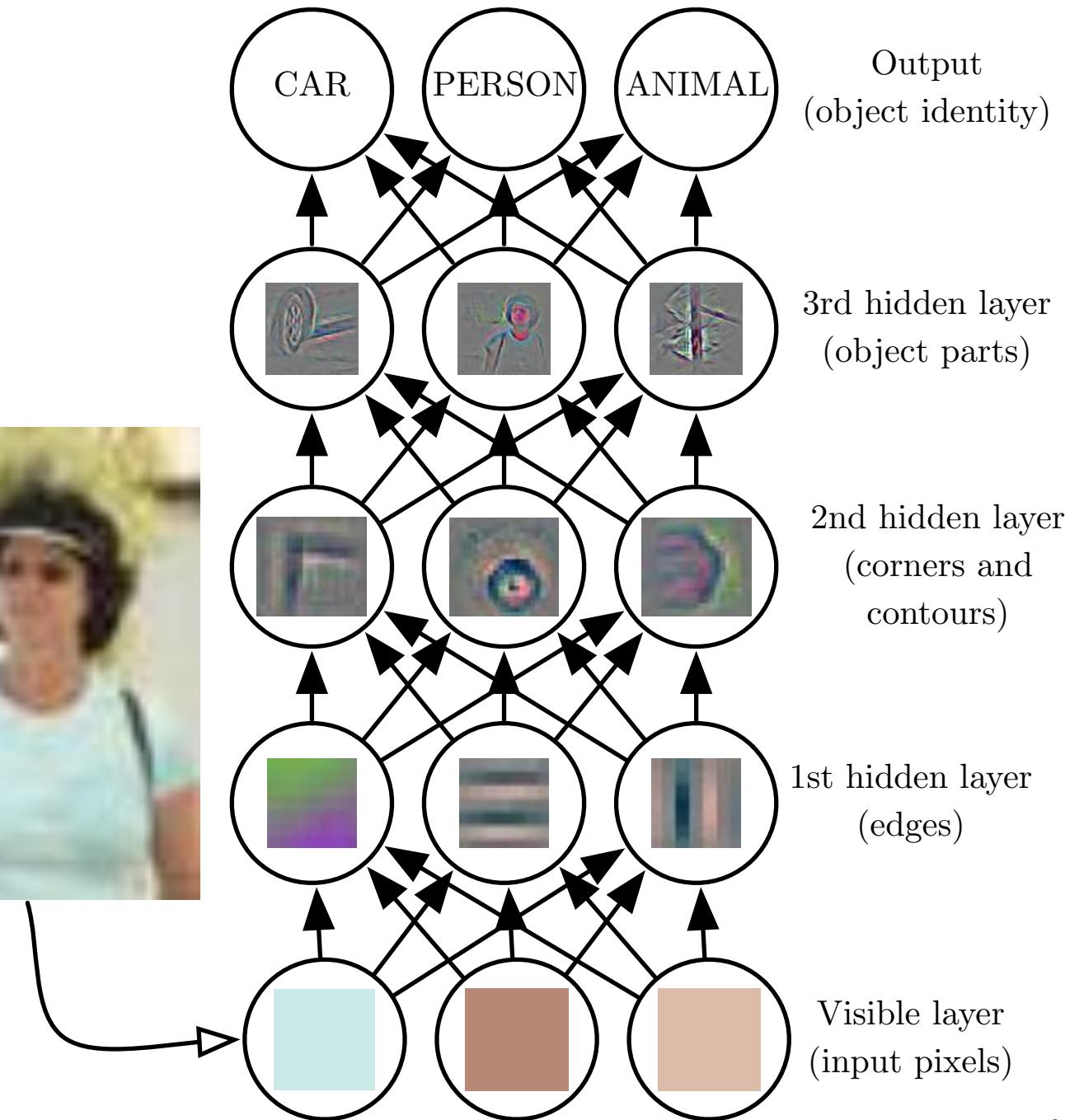
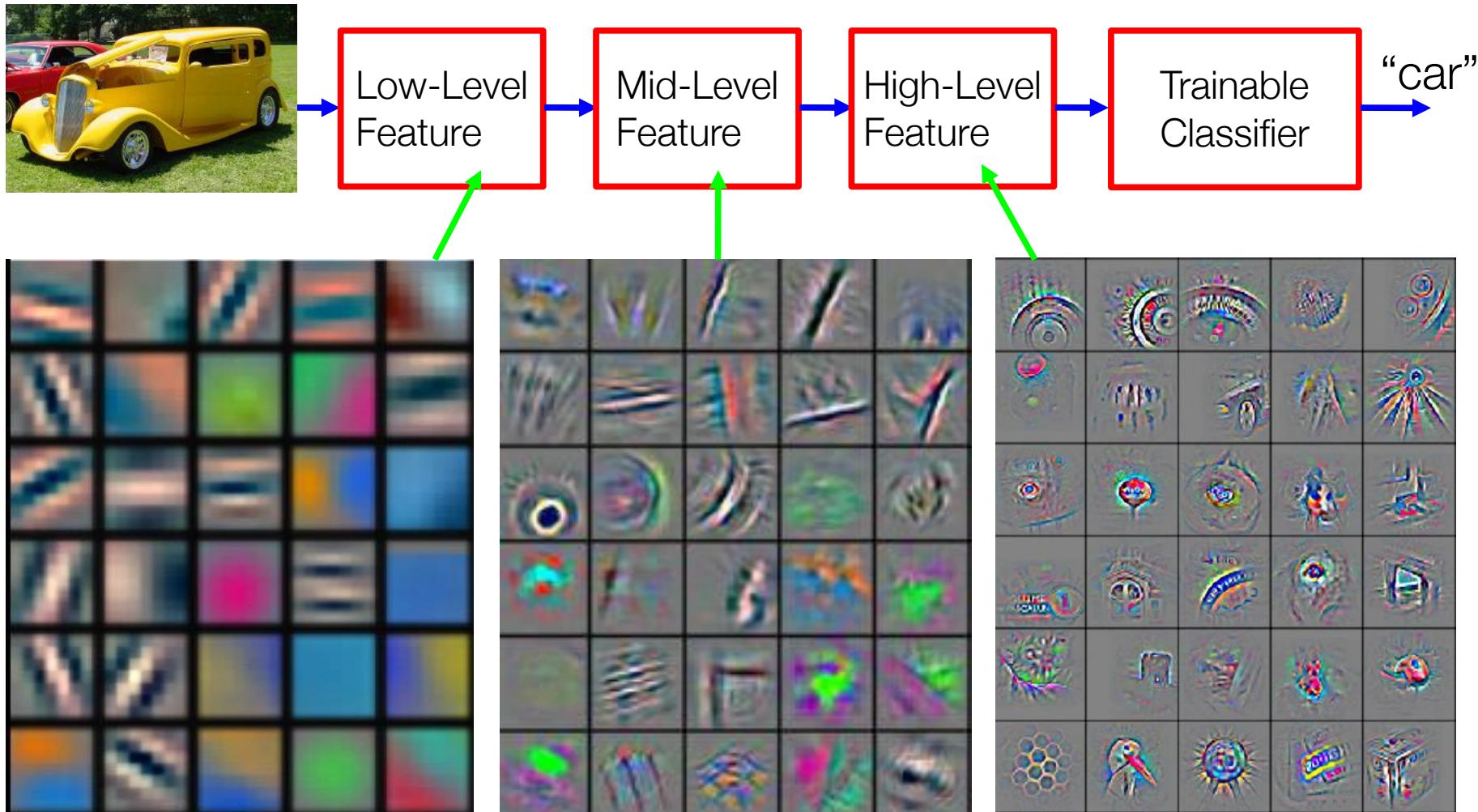


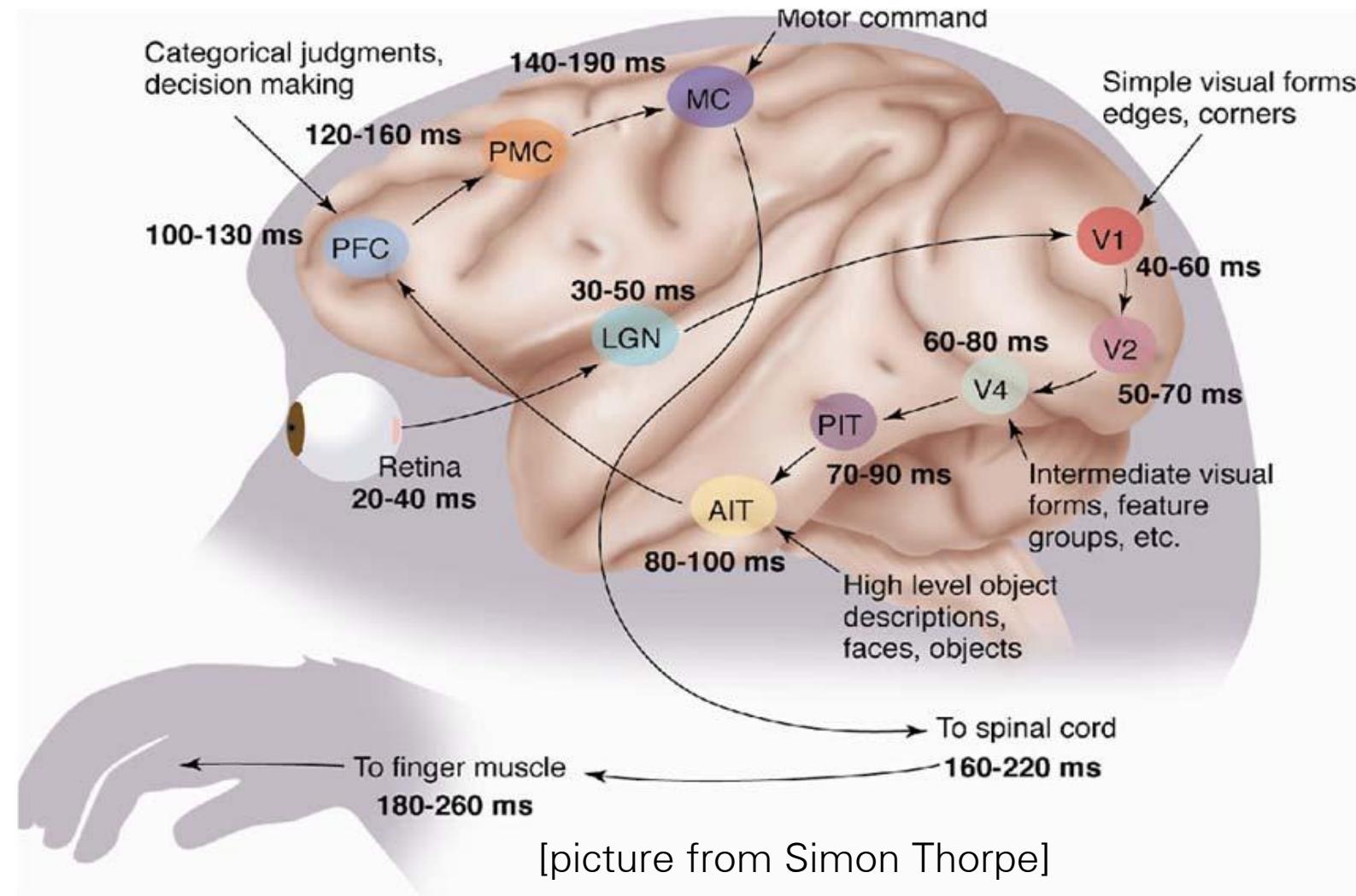
Image credit: Ian Goodfellow

# Deep Learning = Hierarchical Compositionality



# The Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex

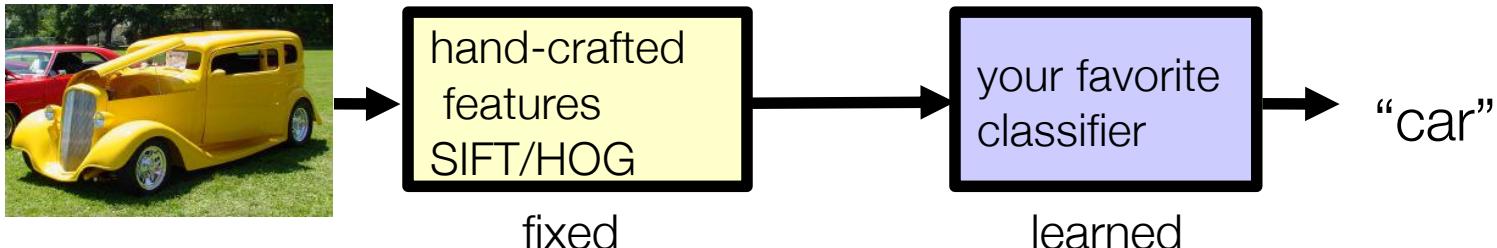


# Three key ideas

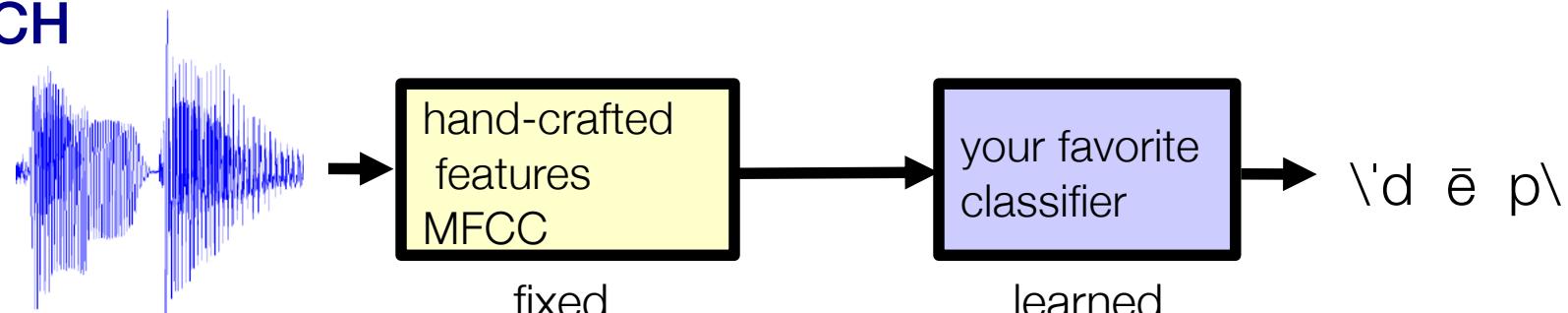
- (Hierarchical) Compositionality
  - Cascade of non-linear transformations
  - Multiple layers of representations
- **End-to-End Learning**
  - Learning (goal-driven) representations
  - Learning to feature extract
- Distributed Representations
  - No single neuron “encodes” everything
  - Groups of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

# Traditional Machine Learning

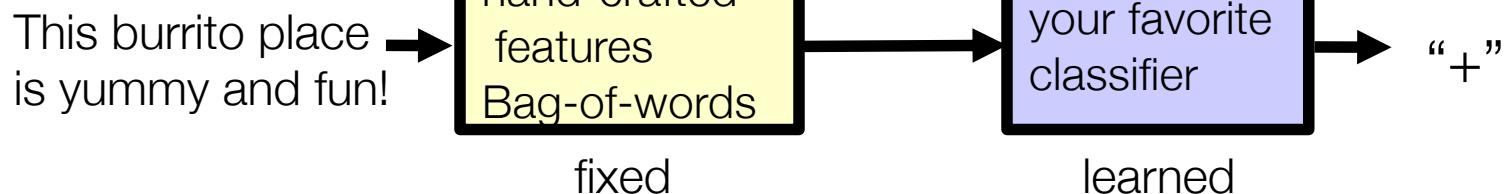
## VISION



## SPEECH

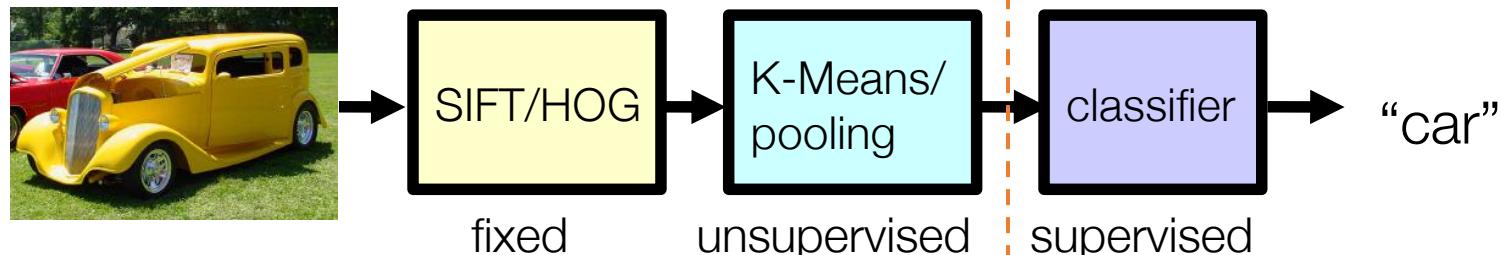


## NLP

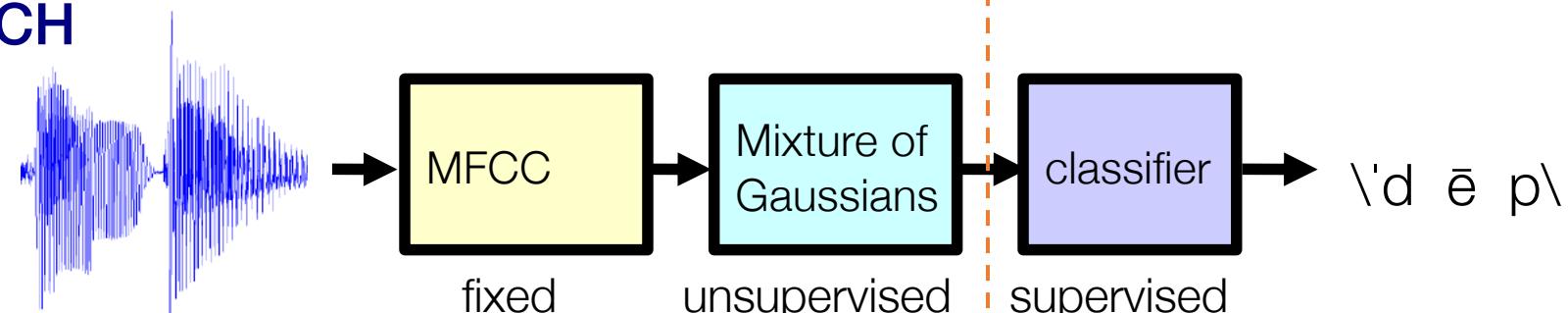


# More accurate version

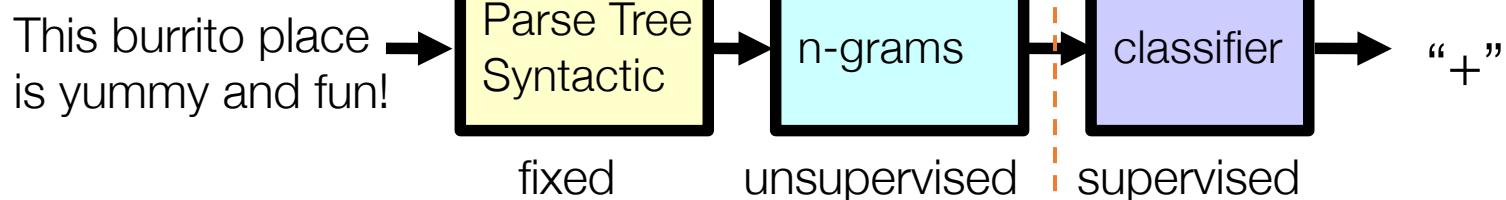
## VISION



## SPEECH

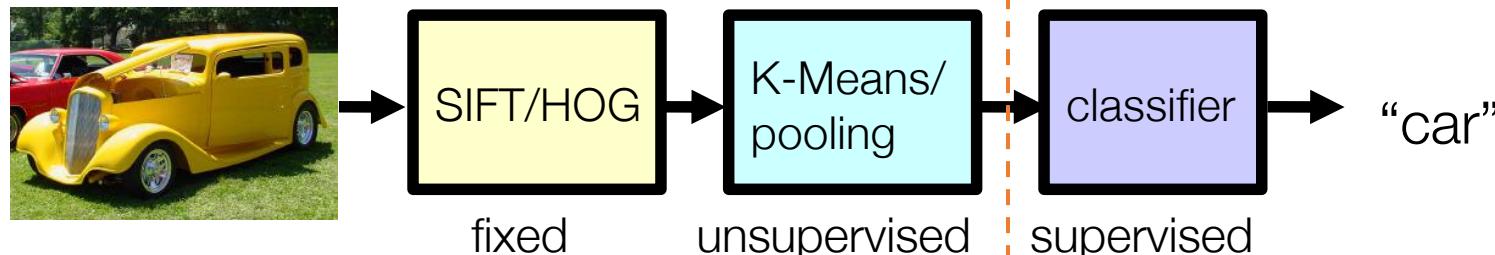


## NLP

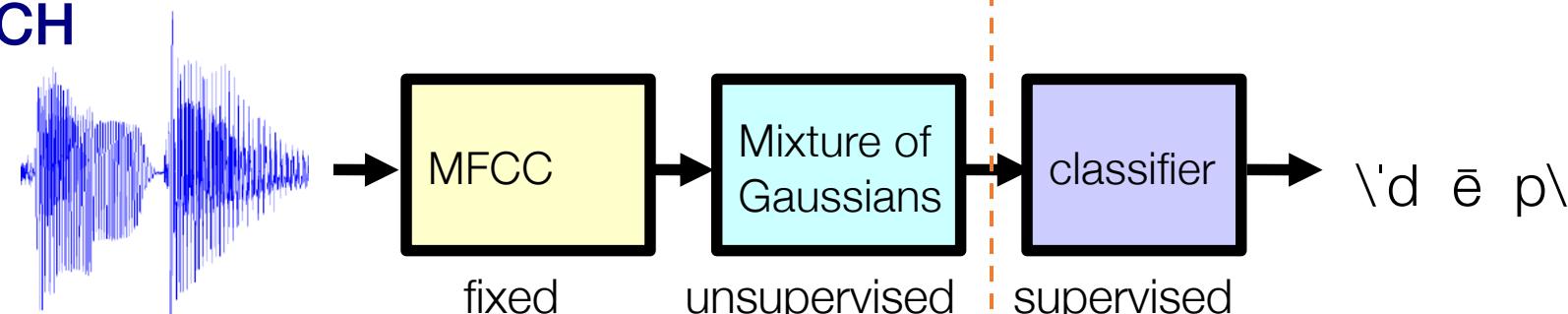


# Deep Learning = End-to-End Learning

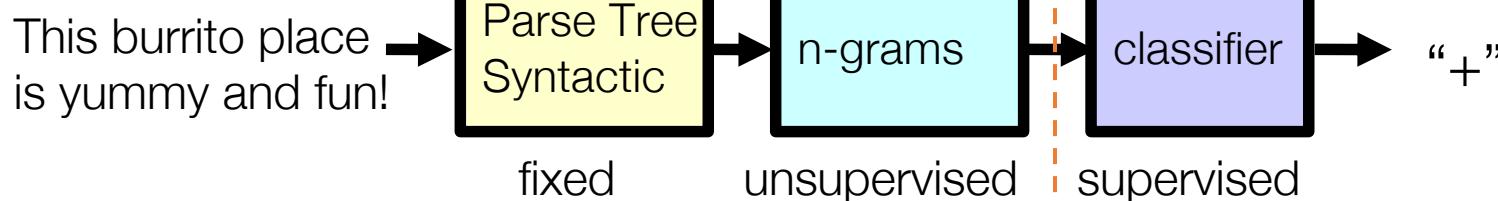
## VISION



## SPEECH

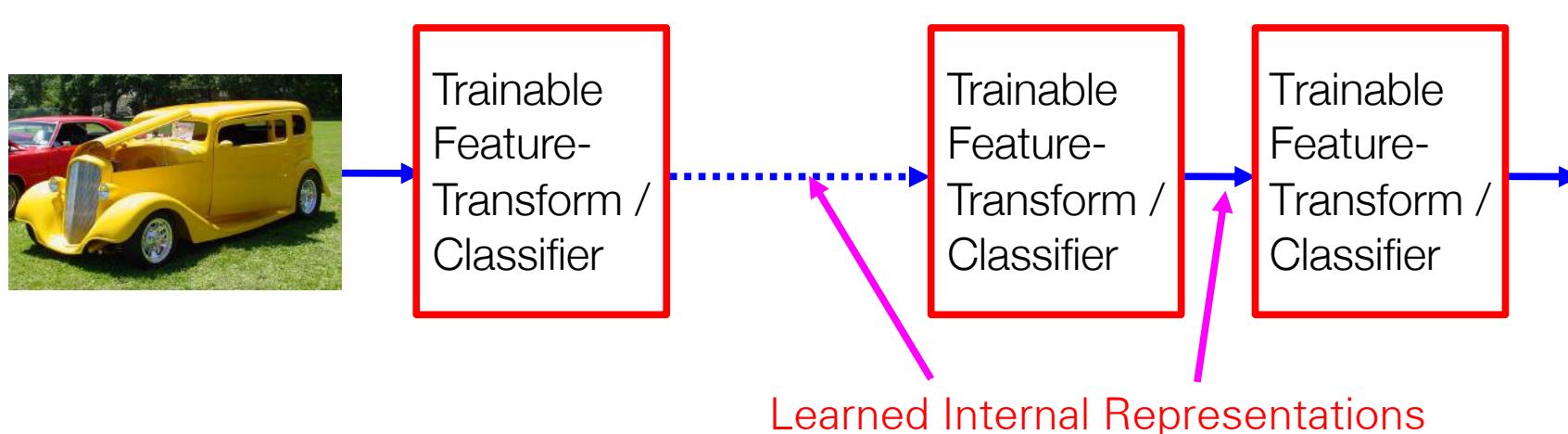


## NLP



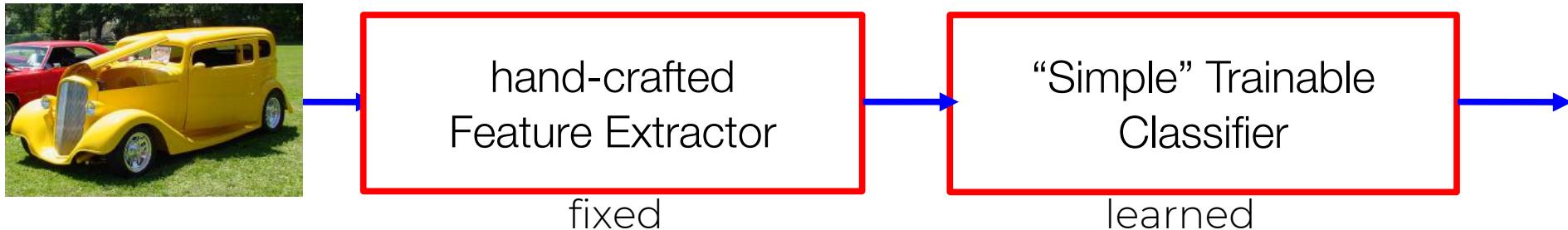
# Deep Learning = End-to-End Learning

- 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

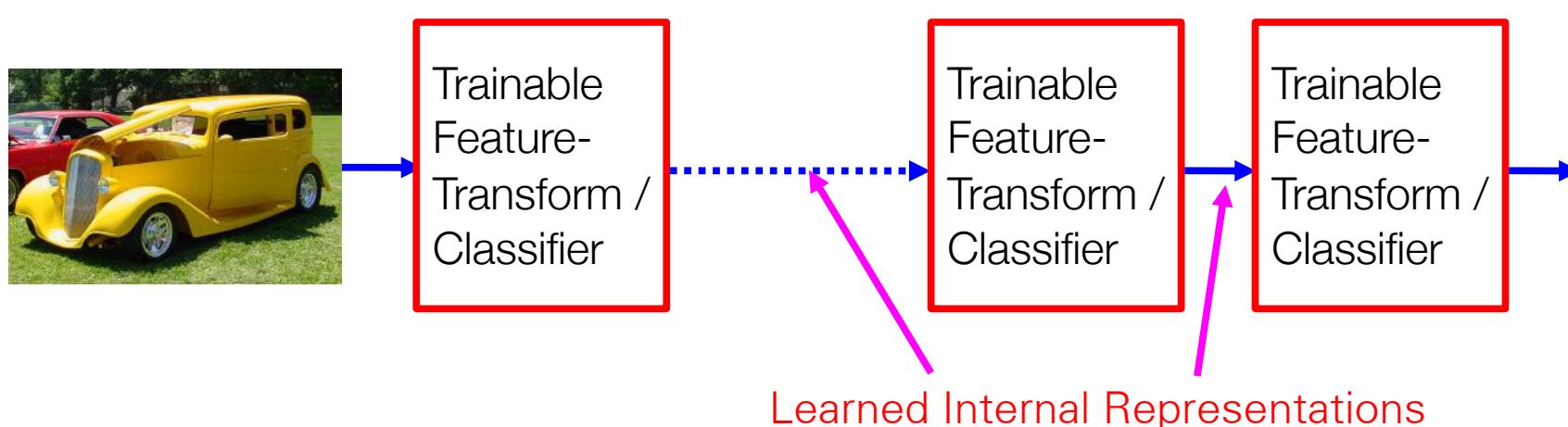


# “Shallow” vs Deep Learning

- “Shallow” models



- Deep models

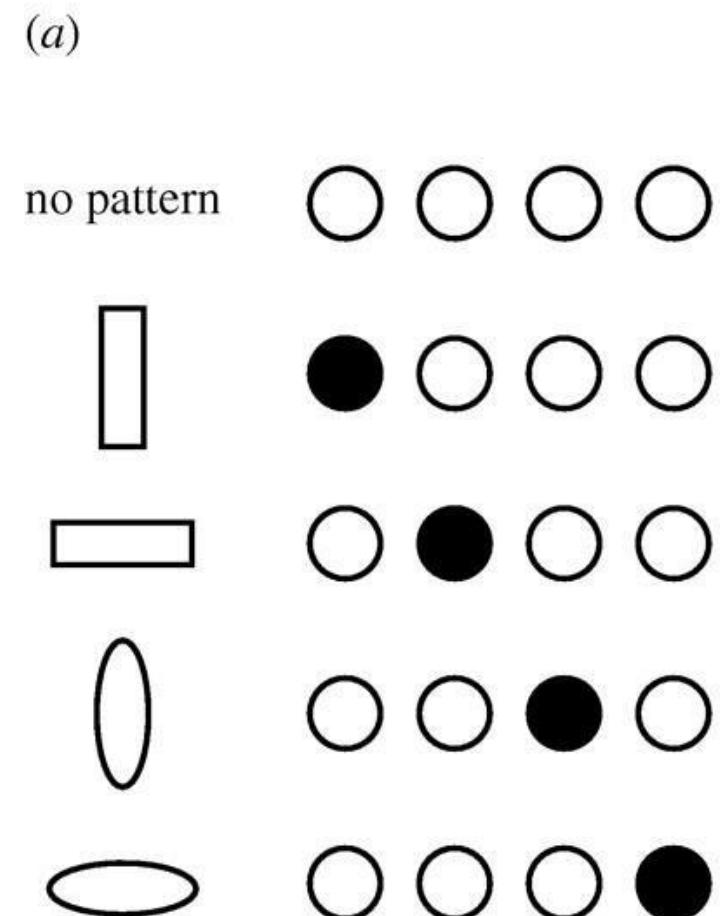


# Three key ideas

- (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

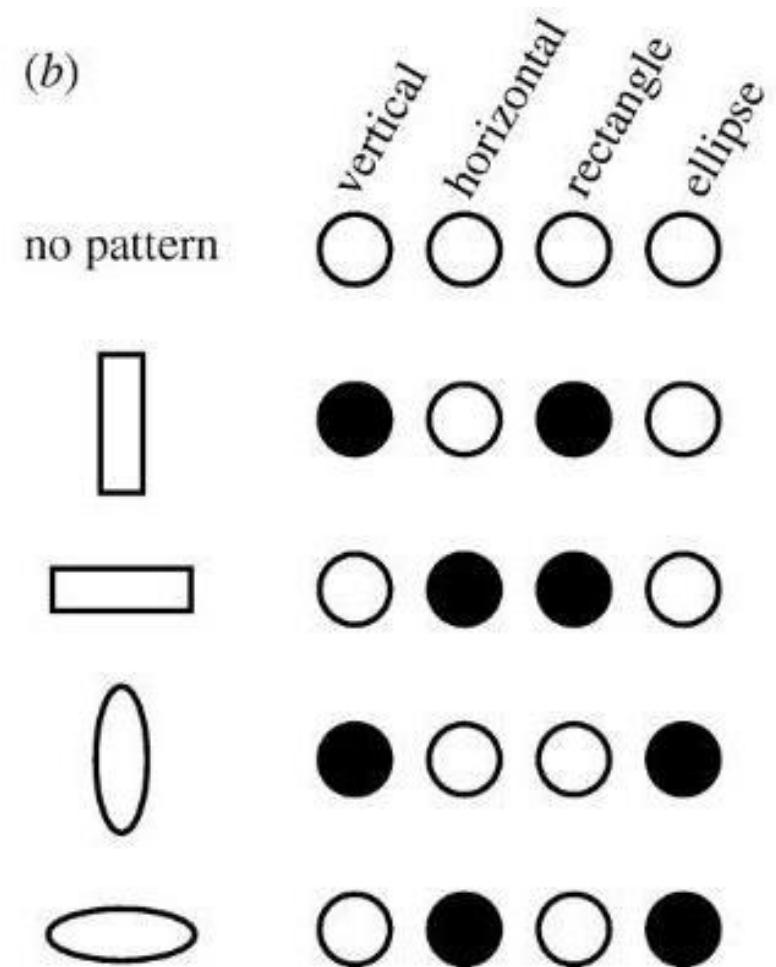
- The simplest way to represent things with neural networks is to **dedicate one neuron to each thing.**
  - Easy to understand.
  - Easy to code by hand
    - Often used to represent inputs to a net
  - Easy to learn
    - This is what mixture models do.
    - Each cluster corresponds to one neuron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



# 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

$$\begin{array}{ll} \text{Local} & \bullet \bullet \circ \bullet = \text{VR} + \text{HR} + \text{HE} = ? \\ \\ \text{Distributed} & \bullet \bullet \circ \bullet = \text{V} + \text{H} + \text{E} \approx \bigcirc \end{array}$$



# Power of distributed representations!

## Scene Classification

bedroom



mountain



- Possible internal representations:

- Objects
- Scene attributes
- Object parts
- Textures



Simple elements & colors

Object part

Object

Scene

# 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 $\geq 3$ : 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

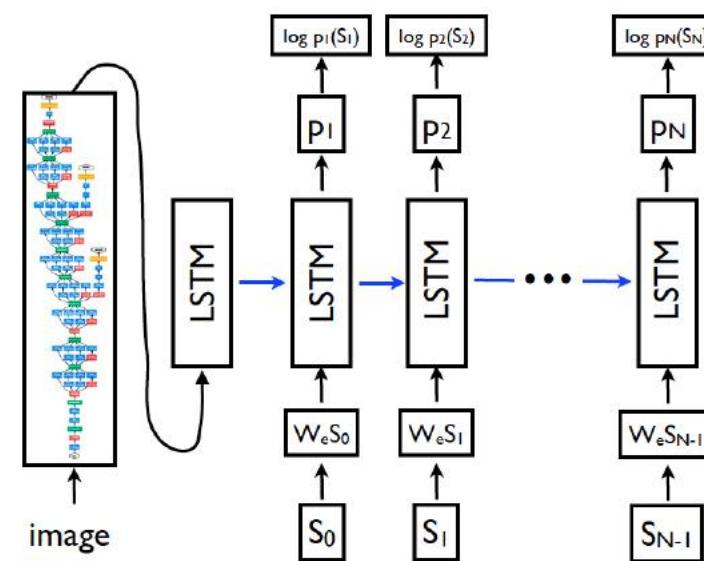
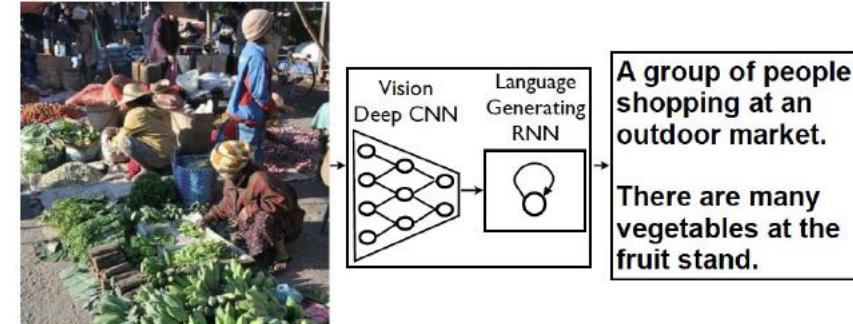
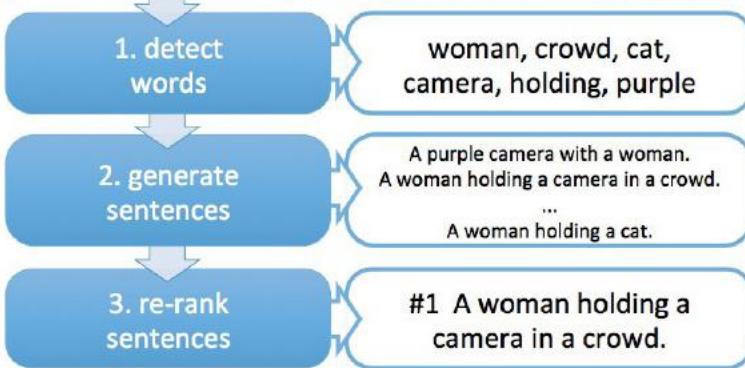
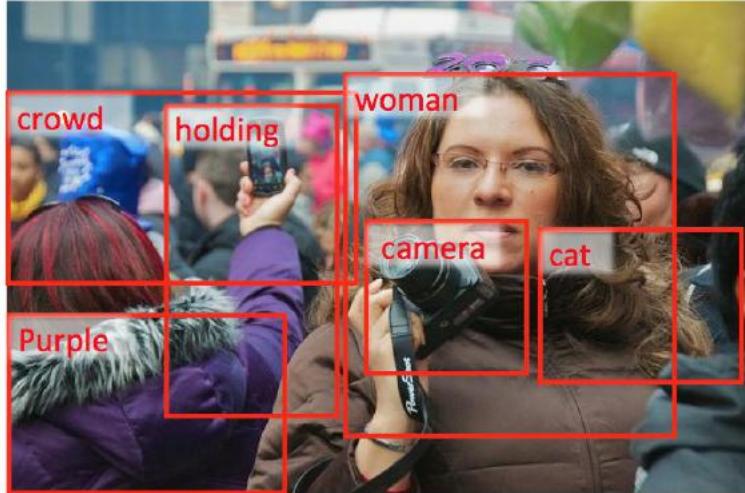
- “Yes, but it often works!”

# 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



[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!”

# 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 be-

ings, 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.

1958 New York Times...

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.

**The New York Times****Science**

WORLD

U.S.

N.Y. / REGION

BUSINESS

TECHNOLOGY

SCIENCE

HEALTH

SPORTS

OPINION

ENVIRONMENT SPACE &amp; COSMOS

# 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.

 FACEBOOK TWITTER GOOGLE+ EMAIL SHARE PRINT REPRINTS

---

**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 .



[View more photos and videos](#)

TWEETS

587

FOLLOWING

18

FOLLOWERS

746

FAVORITES

13



**INTERESTING.JPG** @INTERESTING\_JPG · 18h

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



[View more photos and videos](#)

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](#)

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 .



[View more photos and videos](#)

TWEETS

587

FOLLOWING

18

FOLLOWERS

746

FAVORITES

13



**INTERESTING.JPG** @INTERESTING\_JPG · 16h

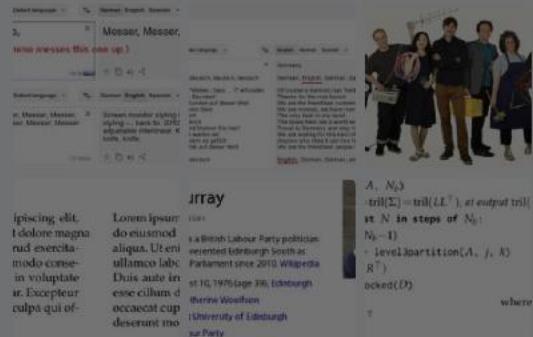
this appears to be a small bedroom in the snow .



6

...

[View more photos and videos](#)



Iain Murray  
@driainmurray

Follow

Today I learned **#googletranslate** sometimes decides that "Deutsch" means "English". Machine learning systems need to cope with weird inputs.



Translate

Turn off instant translation

Sign in

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

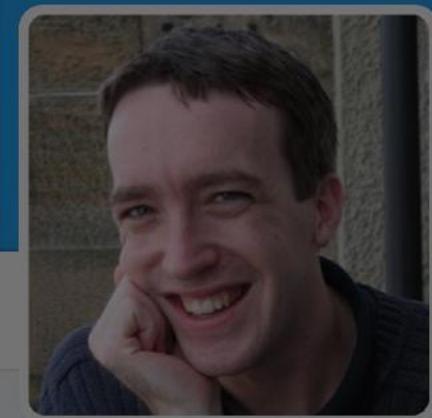
Deutsch, deutsch, deutsch, deutsch

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

A sidebar from Twitter showing various trending topics and user profiles. At the top, there's a section for "#TurkeySaysYes" with 1,520 tweets. Below that is "#BahisSarayındaKazandım". On the right, there are profiles for "Igor Tudor" (5,727 tweets) and "#valentines" (with a heart emoji). At the bottom of the sidebar, there are links for "About", "Help Center", "Terms", "Privacy", "Cookies", and "Ads info".



## Iain Murray

@driainmurray

Academic in Machine Learning and Statistics.

[homepages.inf.ed.ac.uk/imurray2/](http://homepages.inf.ed.ac.uk/imurray2/)

Joined May 2011

 **Iain Murray**  
@driainmurray



More fun pushing [#googletranslate](#)'s neural net into weird states. (BTW try GT on real text if you haven't recently. It's often amazing.)





RETWEETS 120 LIKES 184



X

© 2017 Twitter About Help Center Terms  
Privacy Cookies Ads info

## Hierarchical Text-Conditional Image Generation with CLIP Latents

Aditya Ramesh\*  
OpenAI  
aramesh@openai.com

Prafulla Dhariwal\*  
OpenAI  
prafulla@openai.com

Alex Nichol\*  
OpenAI  
alex@openai.com

Casey Chu\*  
OpenAI  
casey@openai.com

Mark Chen  
OpenAI  
mark@openai.com

### Abstract

Contrastive models like CLIP have been shown to learn robust representations of images that capture both semantics and style. To leverage these representations for image generation, we propose a two-stage model: a prior that generates a CLIP image embedding given a text caption, and a decoder that generates an image conditioned on the image embedding. We show that explicitly generating image representations improves image diversity with minimal loss in photorealism and caption similarity. Our decoders conditioned on image representations can also produce variations of an image that preserve both its semantics and style, while varying the non-essential details absent from the image representation. Moreover, the joint embedding space of CLIP enables language-guided image manipulations in a zero-shot fashion. We use diffusion models for the decoder and experiment with both autoregressive and diffusion models for the prior, finding that the latter are computationally more efficient and produce higher-quality samples.



## Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

Chitwan Saharia\*, William Chan\*, Saurabh Saxena†, Lala Li†, Jay Whang†,  
Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan,  
S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans,  
Jonathan Ho†, David J Fleet†, Mohammad Norouzi\*  
{sahariac, williamchan, mnorouzi}@google.com  
{srbs, lala, jwhang, jonathanho, davidfleet}@google.com

Google Research, Brain Team  
Toronto, Ontario, Canada



## Scaling Autoregressive Models for Content-Rich Text-to-Image Generation

Jiahui Yu\*, Yuanzhong Xu†, Jing Yu Koh†, Thang Luong†, Gunjan Baid†  
Zirui Wang†, Vijay Vasudevan†, Alexander Ku†  
Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson  
Wei Han, Zarana Parekh, Xin Li, Han Zhang  
Jason Baldridge†, Yonghui Wu\*  
{jiahuiyu, yuanzx, jycoh, thangluong, gunjanbaid, ziruiw, vrv, alexku, jasonbaldrige, yonghui}@google.com§

\* Equal contribution. † Core contribution.

Google Research

Parti-350M



Parti-750M



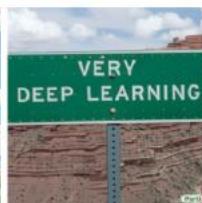
Parti-3B



Parti-20B



A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!



A green sign that says "Very Deep Learning" and is at the edge of the Grand Canyon.  
Puffy white clouds are in the sky.



A blue Porsche 356 parked in front of a yellow brick wall.

# Stable Diffusion

## High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach<sup>1</sup> \*    Andreas Blattmann<sup>1</sup> \*    Dominik Lorenz<sup>1</sup>    Patrick Esser<sup>2</sup>    Björn Ommer<sup>1</sup>

<sup>1</sup>Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany

<https://github.com/CompVis/latent-diffusion>

### Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state of the art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including unconditional image generation, text-to-image synthesis, and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

### 1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [64, 65]. In contrast, the promising results of GANs [3, 26, 39] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [79], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

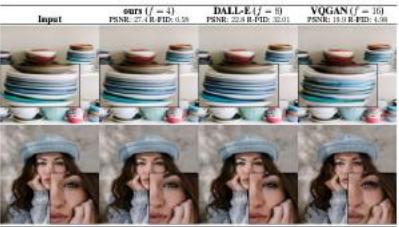


Figure 1. Boosting the upper bound on achievable quality with less aggressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512<sup>2</sup> px. We denote the spatial downsampling factor by  $f$ . Reconstruction FIDs [28] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

results in image synthesis [29, 82] and beyond [7, 44, 47, 56], and define the state-of-the-art in class-conditional image synthesis [15, 30] and super-resolution [70]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [82] or stroke-based synthesis [52], in contrast to other types of generative models [19, 45, 67]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [65].

**Democratizing High-Resolution Image Synthesis** DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 71]. Although the reweighted variational objective [29] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (e.g. 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

A high tech solarpunk utopia in the Amazon rainforest

Generate image



\*The first two authors contributed equally to this work.

# Stable Diffusion

## High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach<sup>1</sup> \*    Andreas Blattmann<sup>1</sup> \*    Dominik Lorenz<sup>1</sup>    Patrick Esser<sup>2</sup>    Björn Ommer<sup>1</sup>

<sup>1</sup>Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany

<https://github.com/CompVis/latent-diffusion>

### Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state of the art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including unconditional image generation, text-to-image synthesis, and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

### 1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [64, 65]. In contrast, the promising results of GANs [3, 26, 39] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [79], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive



Figure 1. Boosting the upper bound on achievable quality with less aggressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512<sup>2</sup> px. We denote the spatial downsampling factor by  $f$ . Reconstruction FIDs [28] and PSNR are calculated on ImageNet-val [12]; see also Tab. 8.

results in image synthesis [29, 82] and beyond [7, 44, 47, 56], and define the state-of-the-art in class-conditional image synthesis [15, 30] and super-resolution [70]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [82] or stroke-based synthesis [52], in contrast to other types of generative models [19, 45, 67]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [65]. **Democratizing High-Resolution Image Synthesis** DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 71]. Although the reweighted variational objective [29] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (e.g. 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

A small cabin on top of a snowy mountain, no blur 4k resolution, ultra detailed

Generate image



\*The first two authors contributed equally to this work.



**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”



arxiv.org

Testing Relational Understanding in Text-Guided Image Gen...  
Relations are basic building blocks of human cognition.  
Classic and recent work suggests that many relations are ...

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

**“A spoon in a cup”**



**“A cup on a spoon”**





Melanie Mitchell  
@MelMitchell1

...

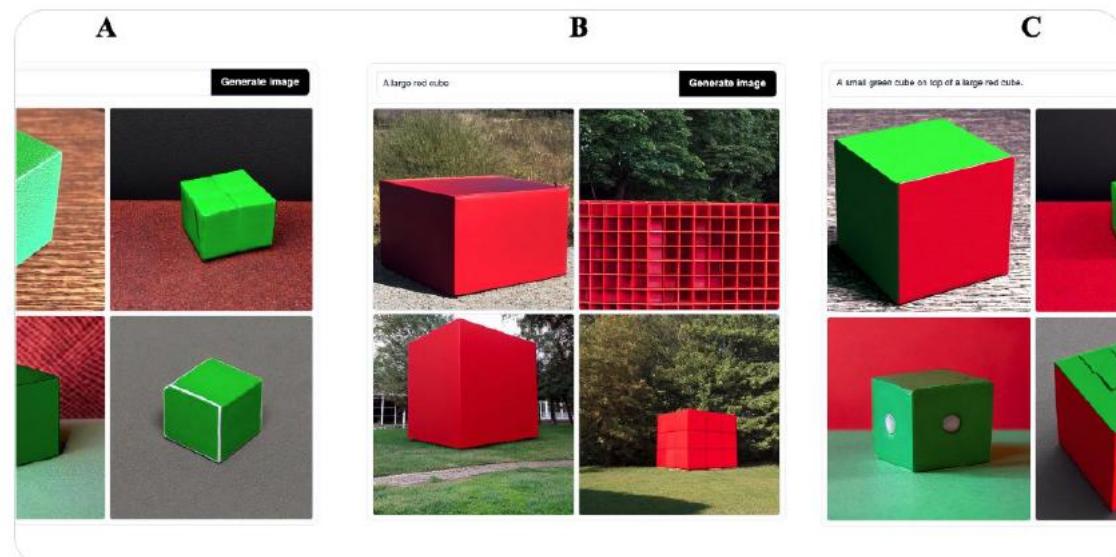
\*Prepositions are hard.\*

Stable diffusion demo ([huggingface.co/spaces/stabilityai/stable-diffusion](https://huggingface.co/spaces/stabilityai/stable-diffusion))

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



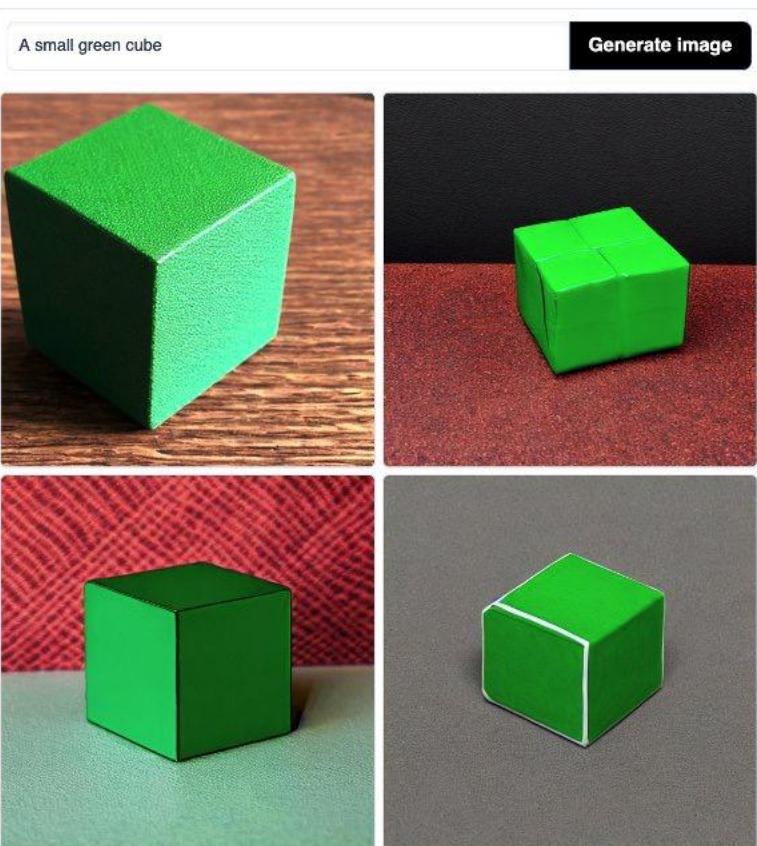
6:10 PM · Aug 23, 2022 · Twitter Web App



Melanie Mitchell  
@MelMitchell1

...

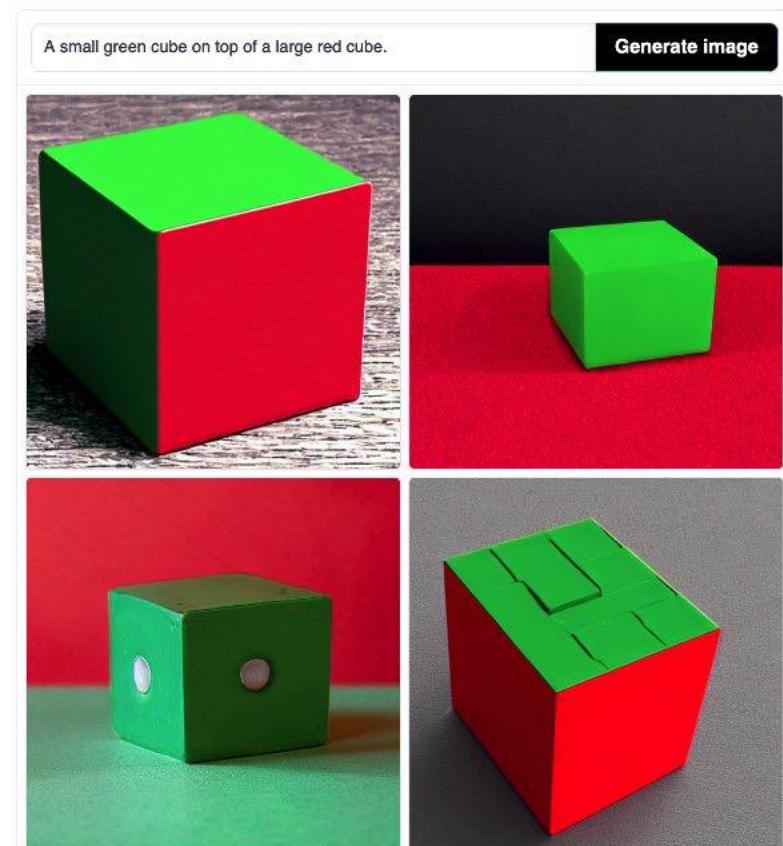
A



B



C





Melanie Mitchell

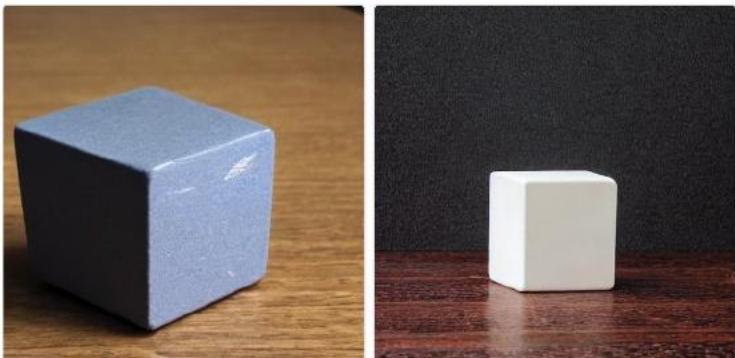
@MelMitchell1

...

A

One cube on top of another cube

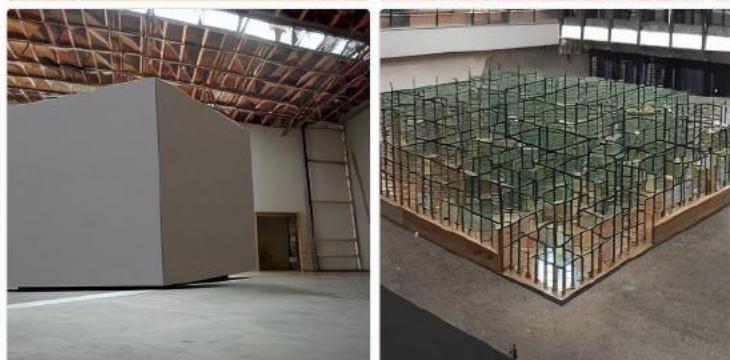
Generate image



B

A small cube to the left of a large cube

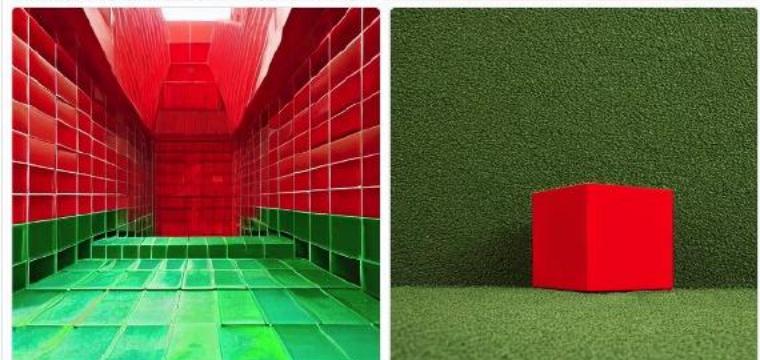
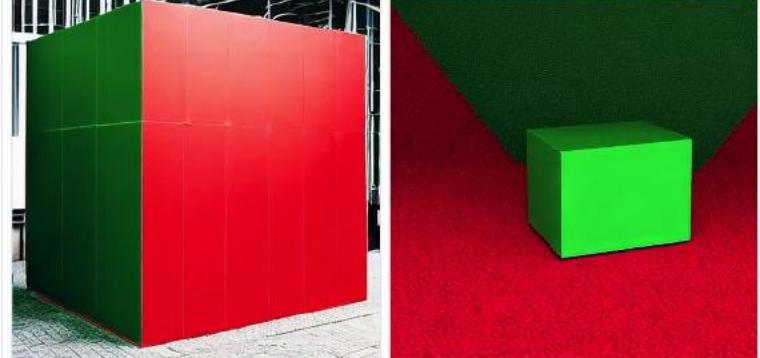
Generate Image



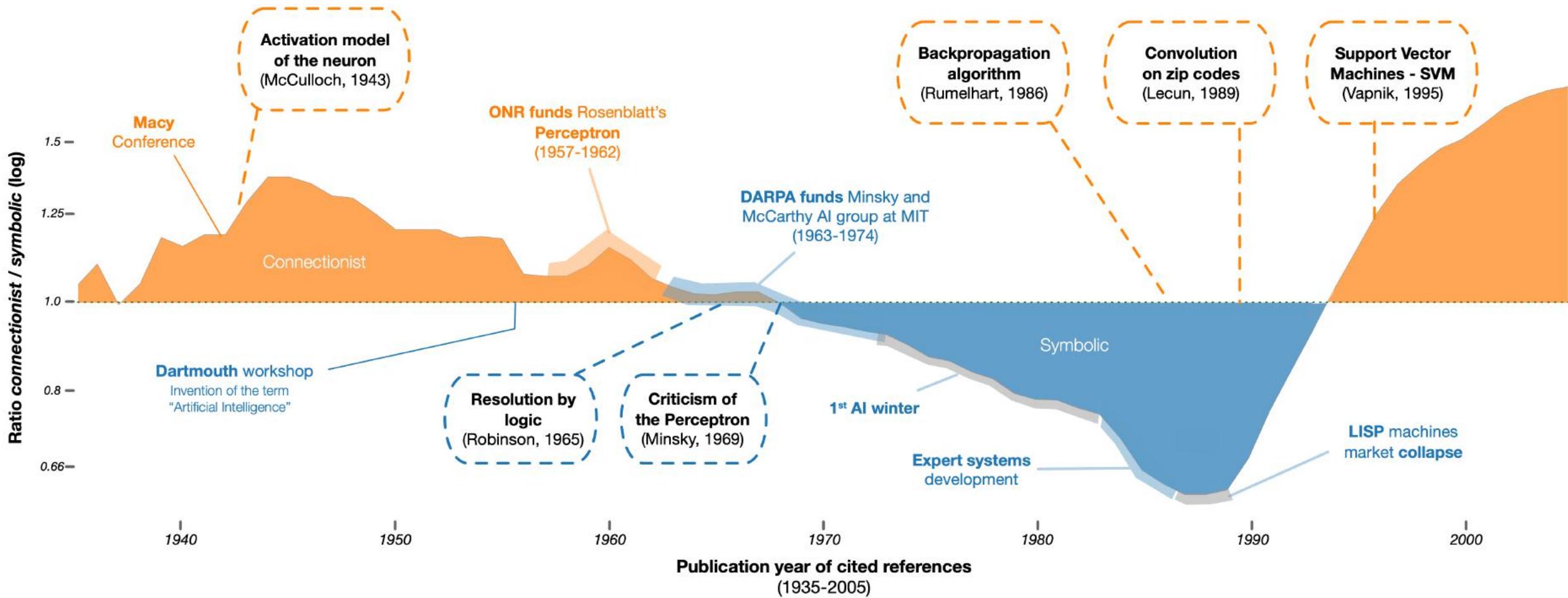
C

A red cube below a green cube

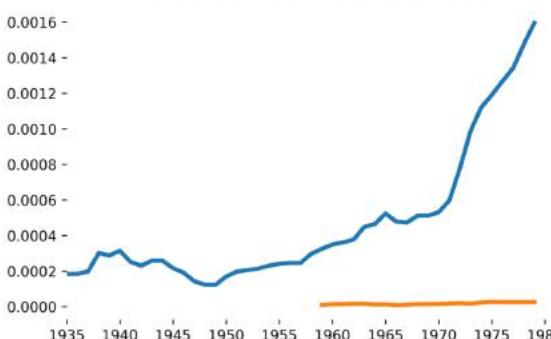
Generate image



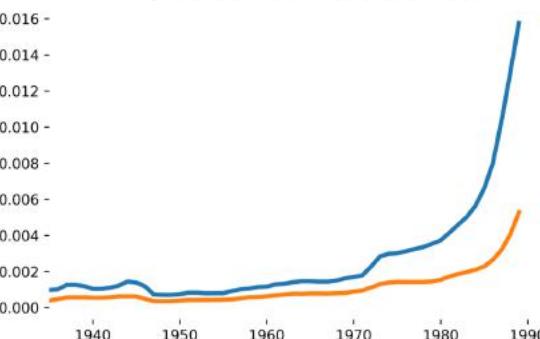
6:10 PM · Aug 23, 2022 · Twitter Web App



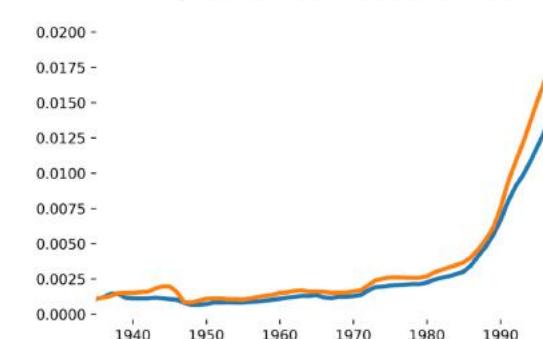
Cited between 1980 and 1989



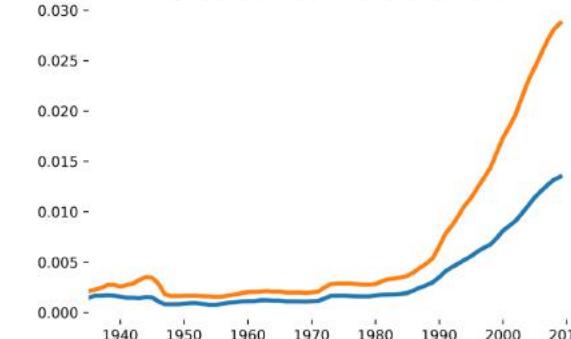
Cited between 1990 and 1999



Cited between 2000 and 2009

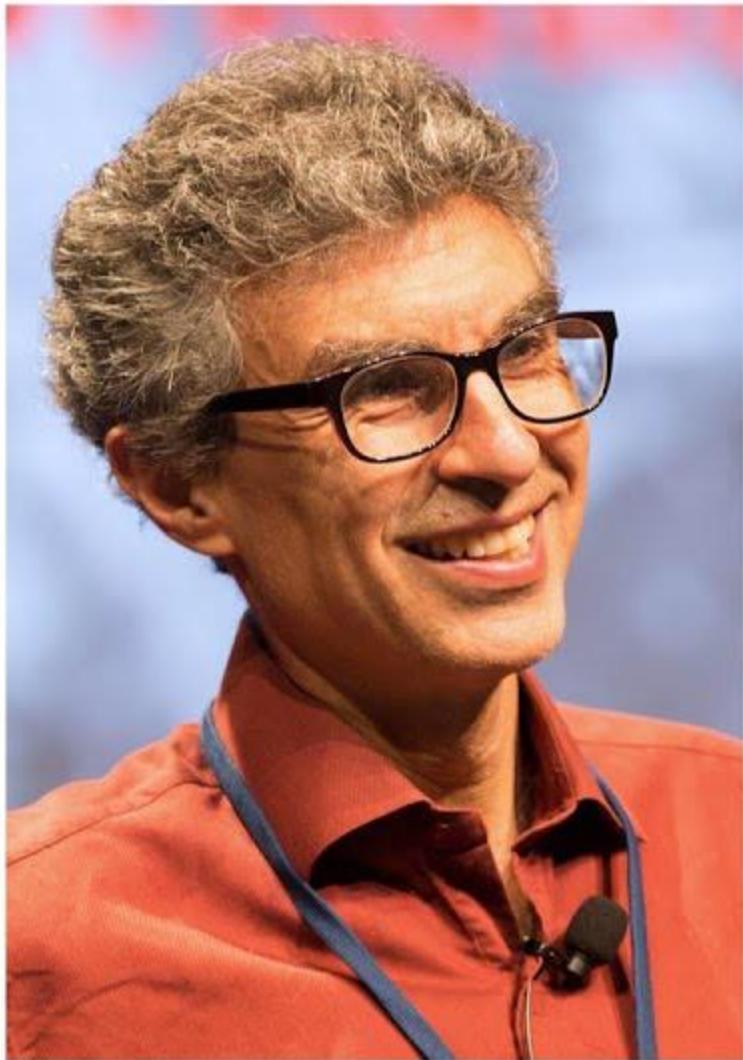


Cited between 2010 and 2018



# AI DEBATE : YOSHUA BENGIO | GARY MARCUS

---



Gary Marcus  
—  
Yoshua Bengio



# **Next Lecture:**

# **Machine Learning Overview**