Illustration: Maria Nguyen // Quanta Magazine

CMP784

DEEP LEARNING

Lecture #01 – Introduction

Erkut Erdem // Hacettepe University // Fall 2021

HACETTEPE UNIVERSITY COMPUTER VISION LAB

Welcome to CMP784

- An overview of various deep architectures and learning methods

- Develop fundamental and practical skills at applying deep learning to your research.

A little about me...

Koç University-İş Bank
Artificial Intelligence Center
Adjunct Faculty
2020-now

Hacettepe University
Associate Professor
2010-now

Télécom ParisTech
Post-doctoral Researcher
2009-2010

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

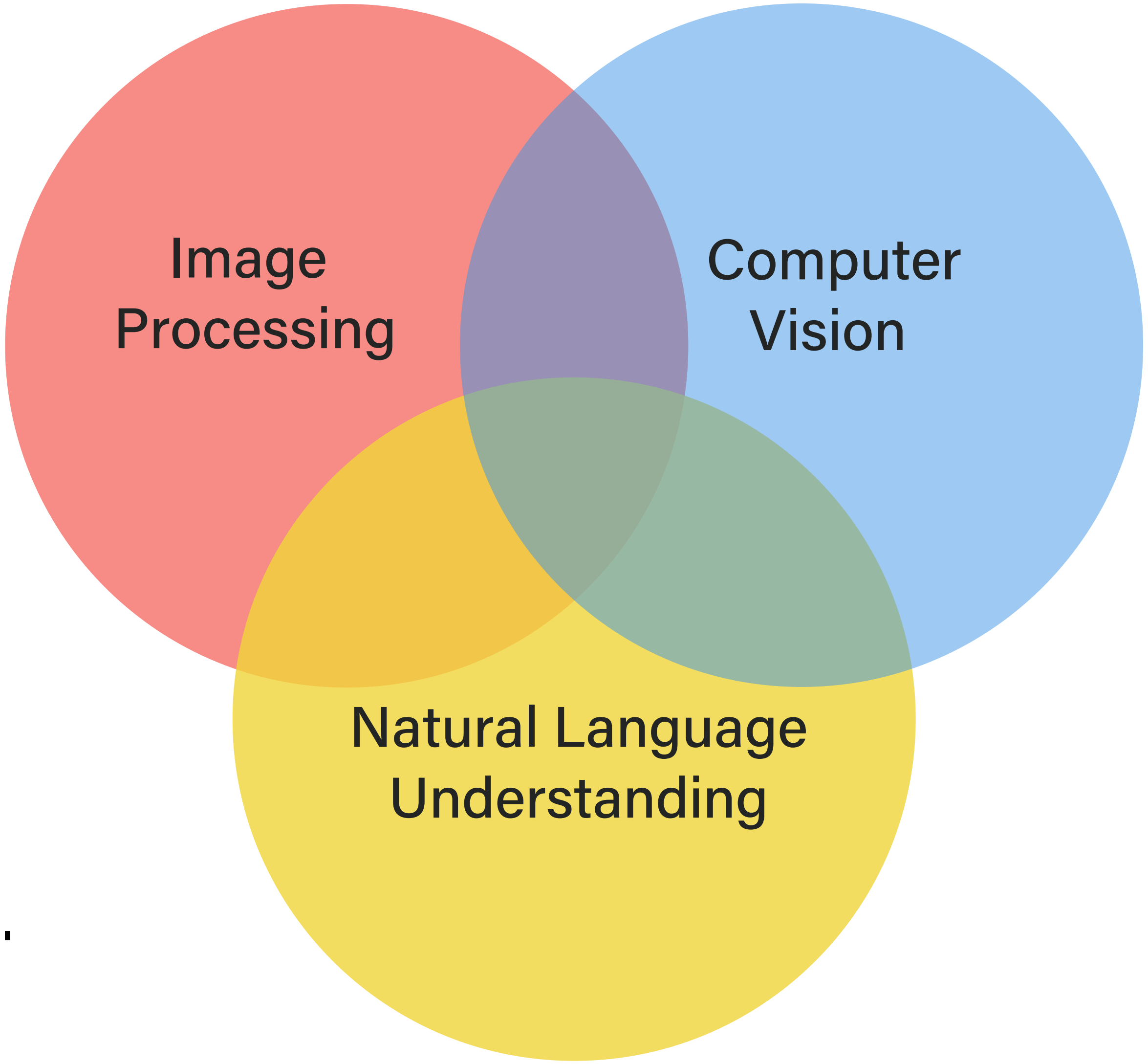
UCLA
Fall 2007
Visiting Student

VirginiaTech
Visiting Research Scholar
Summer 2006

http://web.cs.hacettepe.edu.tr/~erkut
@erkuterdem
erkut@cs.hacettepe.edu.tr

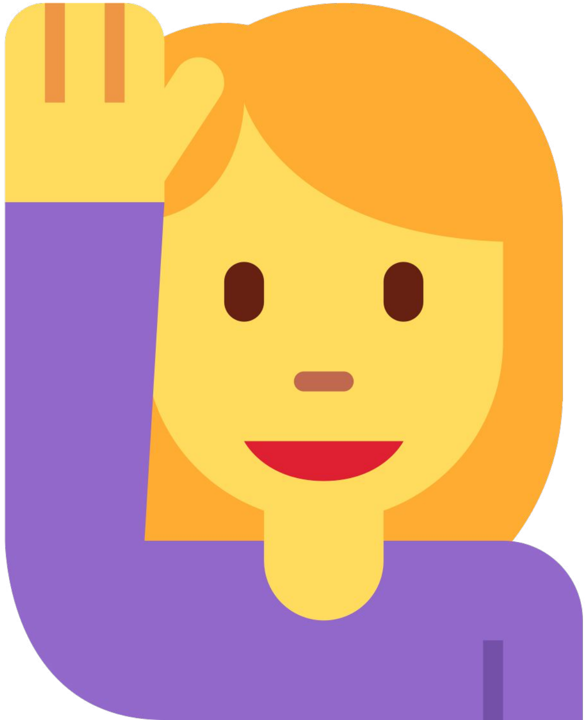
Research Interests

- I study better ways to understand and process 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?

- Introduce yourselves
  - Who are you?
  - Who do you work with if you have a thesis supervisor?
  - What made you interested in this class?
  - What are your expectations?
  - What do you know about machine learning and deep learning?

Course Logistics

Course information

Time/Location	09:00-12:00pm Wednesday, Zoom
Instructor	Erkut Erdem

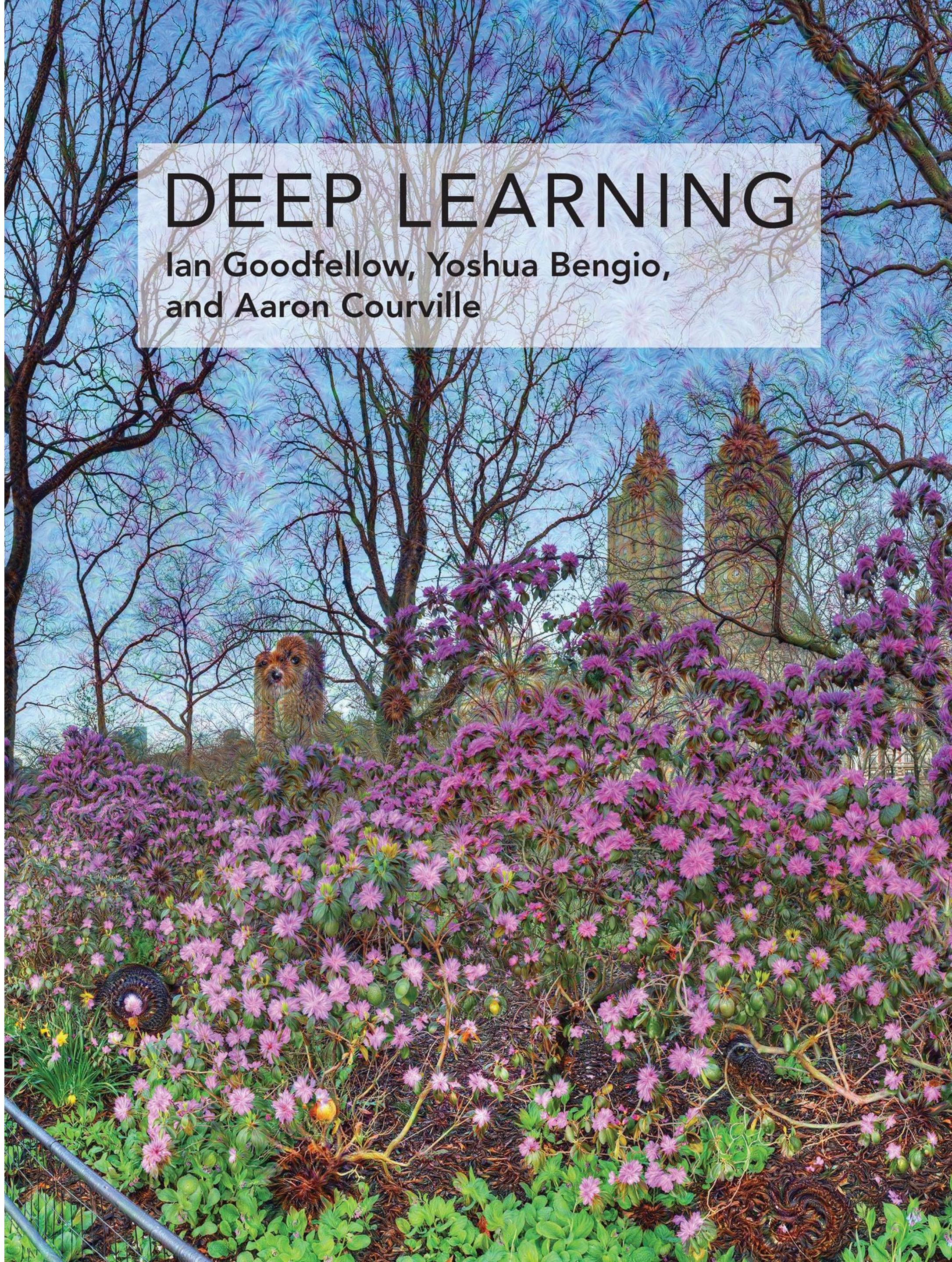
- piazza for course related announcements:

  https://piazza.com/hacettepe.edu.tr/fall2021/cmp784

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.)



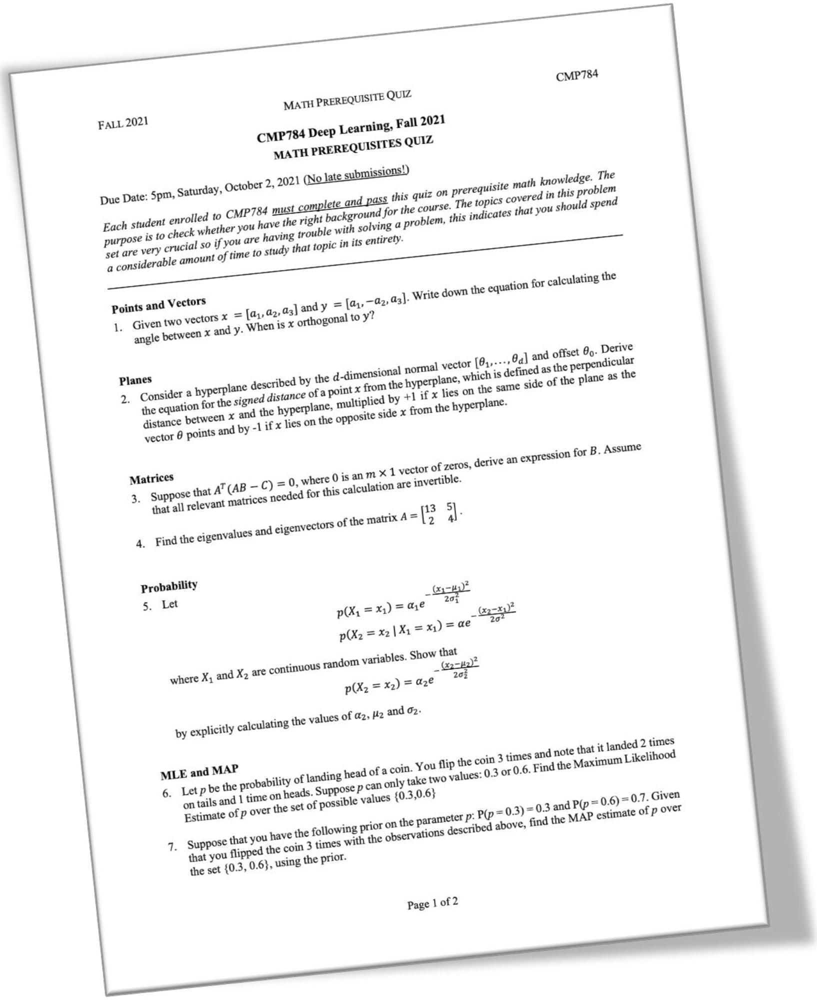
Instruction style

- 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 discussed in our class.
  - Weekly quizzes about the papers presented each week

Prerequisites

- Calculus and linear algebra
  - Derivatives,
  - Matrix operations
- Probability and statistics (IST299, IST292)
- Neural networks (CMP684)
- Machine learning (BBM406, CMP712)
- Programming



FALL 2021 MATH PREREQUISITE QUIZ CMP784

CMP784 Deep Learning, Fall 2021
MATH PREREQUISITES QUIZ

Due Date: 5pm, Saturday, October 2, 2021 (<u>No late submissions!</u>)

Each student enrolled to CMP784 <u>must complete and pass</u> this quiz on prerequisite math knowledge. The purpose is to check whether you have the right background for the course. The topics covered in this problem set are very crucial so if you are having trouble with solving a problem, this indicates that you should spend a considerable amount of time to study that topic in its entirety.

Points and Vectors

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

Planes

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

Matrices

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

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

Probability

5. Let

$$p(X_1 = x_1) = \alpha_1 e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}}$$
$$p(X_2 = x_2 \mid X_1 = x_1) = \alpha e^{-\frac{(x_2-x_1)^2}{2\sigma^2}}$$

where X_1 and X_2 are continuous random variables. Show that

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

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

MLE and MAP

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

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

Page 1 of 2

Math Prerequisite Quiz
Due Date: 5pm, Sat, Oct 2, 2021.

Each student enrolled to CMP784 <u>must complete and pass</u> this quiz!

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

Topics Covered in BBM406/CMP712

- 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

Topics Covered in CMP684

- Continuous and discrete system models
- Neuron and Its Analytic Model
- Hopfiels Neural Network
- Perceptron Learning Algorithms
- Multilayer Perceptron (MLP)
    - Derivation of the learning algorithm
    - Error backpropagation
    - Memorization and generalization
    - Intervals and normalization
- Radial Basis Function Neural Nets
- Dynamical Neural Nets
- Feedback Nets
- Second Order Training Algorithms
    - Levenberg-Marquardt algorithm
    - Gauss-Newton algorithm
- Stability in Adaptive Systems
- Applications of Neural Nets

Grading

Math Prerequisites Quiz	3%
Practicals	16% (2 practicals x 8% each)
Final Exam	25%
Course Project	32%
Paper Presentations	15%
Weekly Quizzes	9%

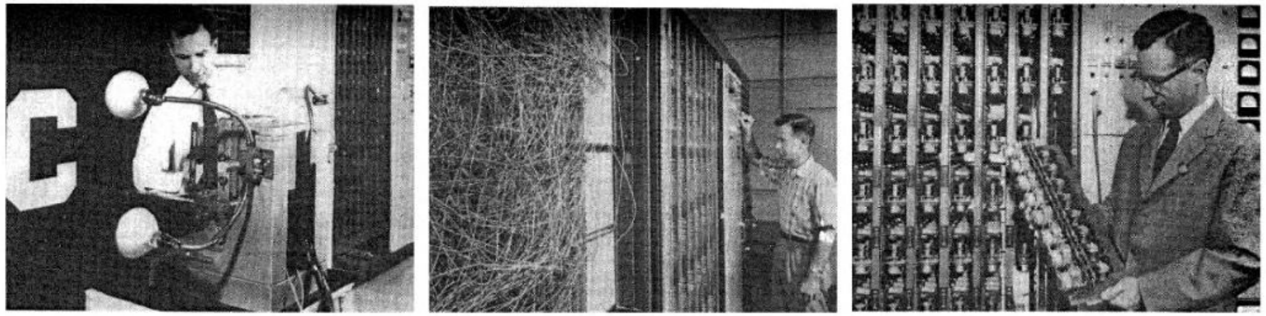
Schedule

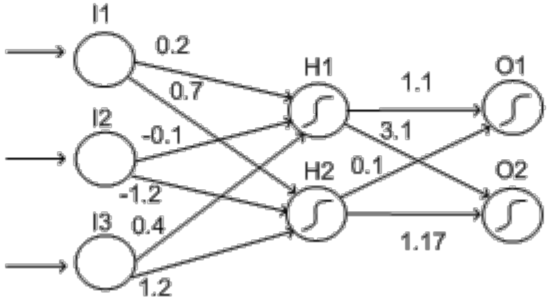
Week 1	Introduction to Deep Learning
Week 2	Machine Learning Overview
Week 3	Multi-Layer Perceptrons
Week 4	Training Deep Neural Networks
Week 5	Convolutional Neural Networks
Week 6	Understanding and Visualizing CNNs
Week 7	Recurrent Neural Networks
Week 8	Attention and Memory

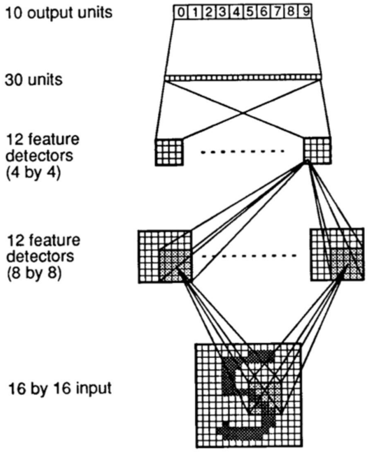
Schedule

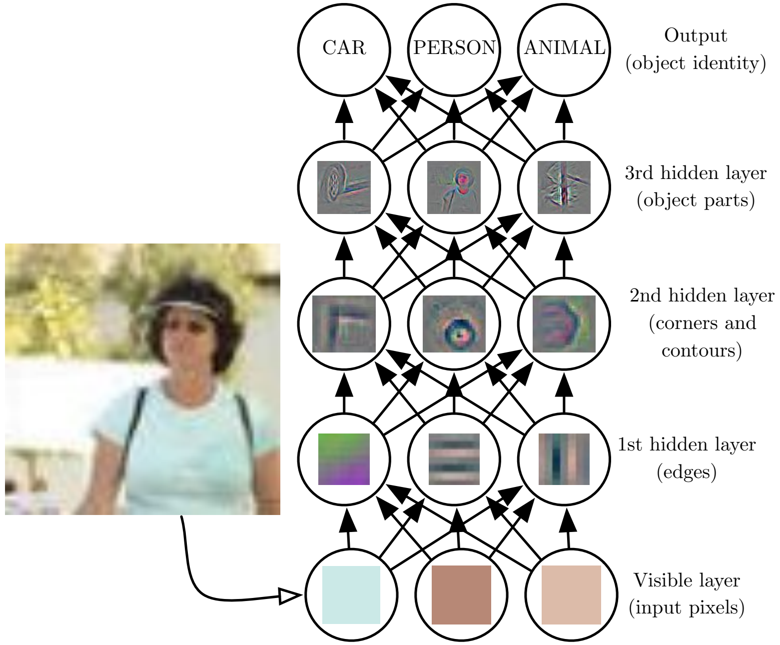
Week 9	Autoencoders and Autoregressive Models
Week 10	Progress Presentations
Week 11	Generative Adversarial Networks
Week 12	Variational Autoencoders
Week 13	Self-supervised Learning
Week 14	Final Project Presentations

Lecture 1: Introduction to Deep Learning

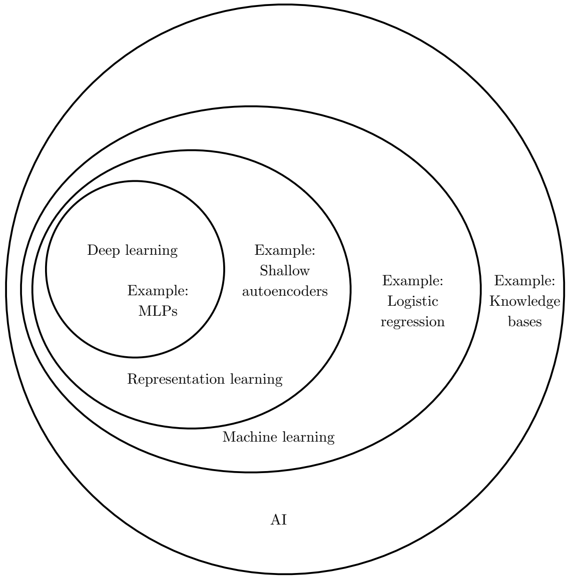


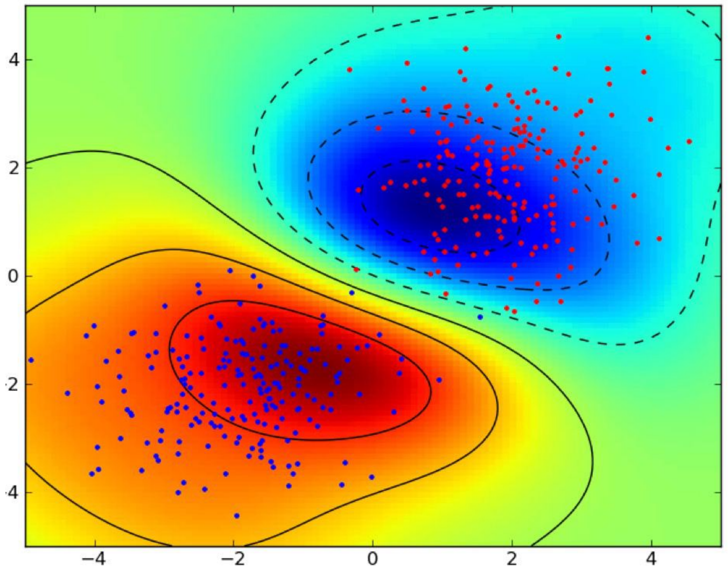


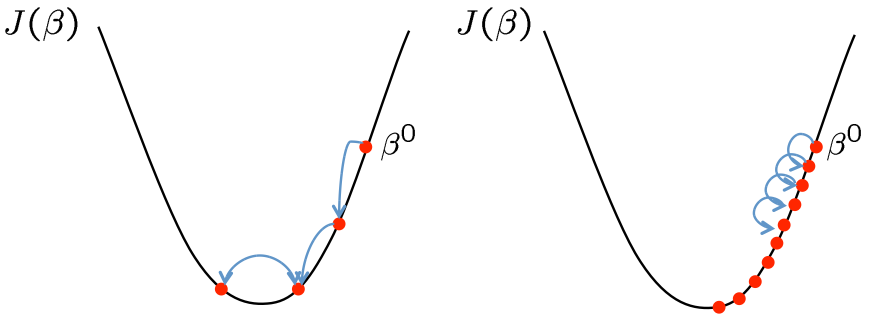


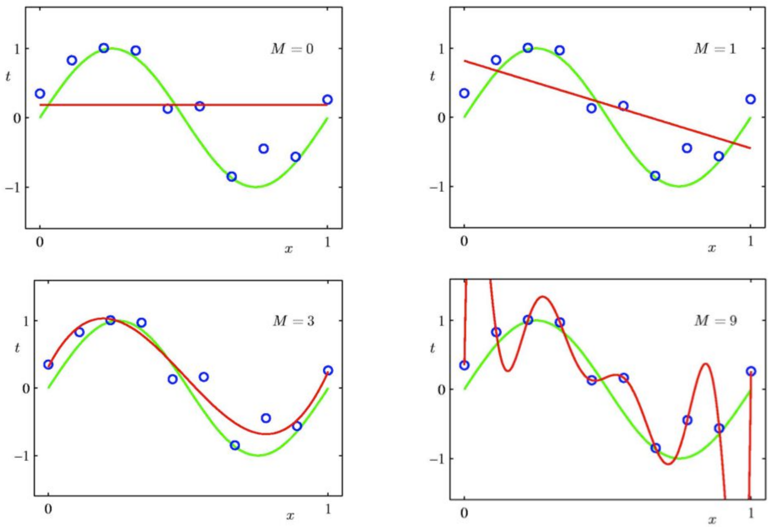


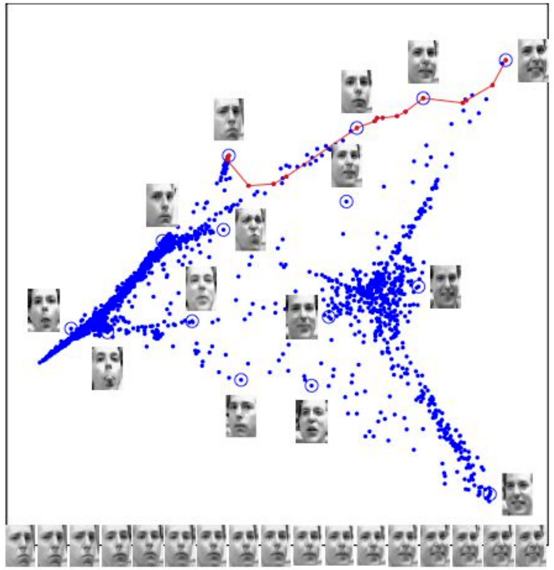
Lecture 2: Machine Learning Overview

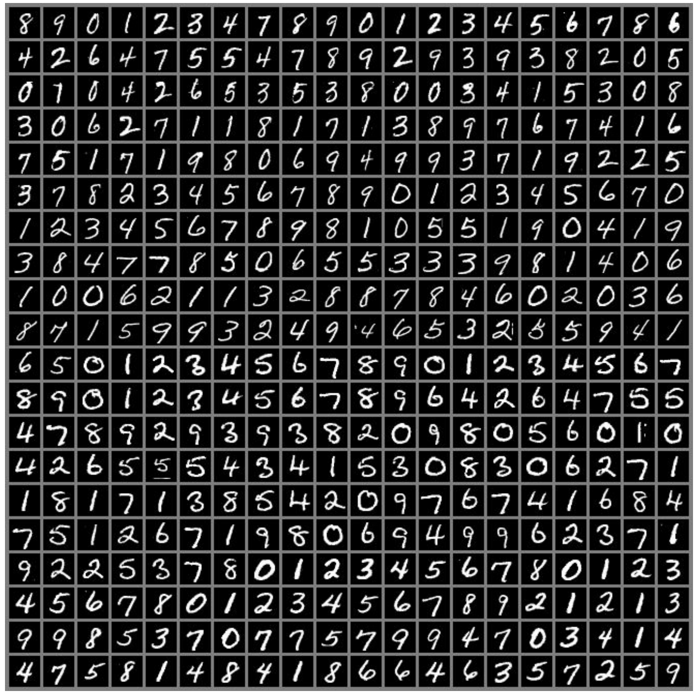




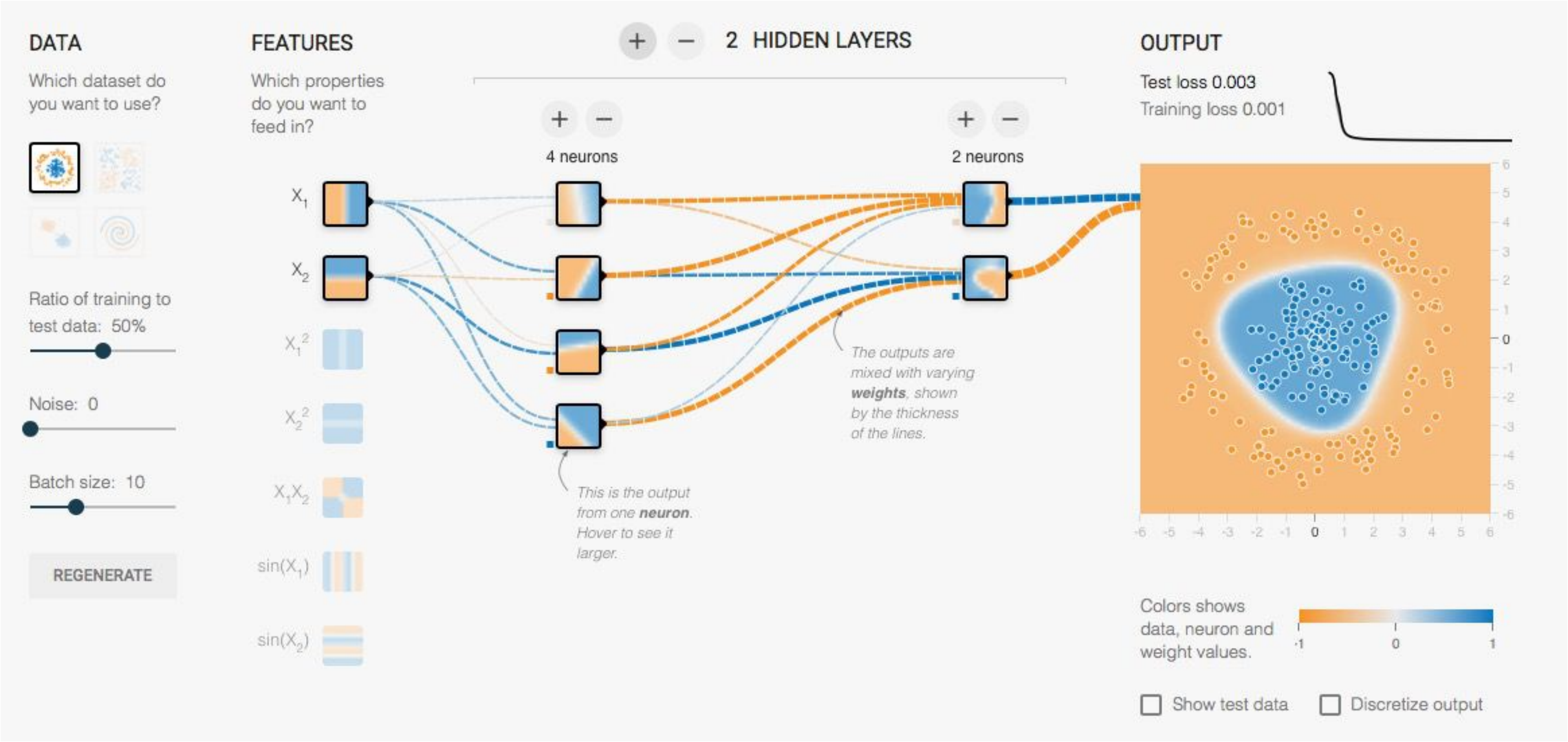






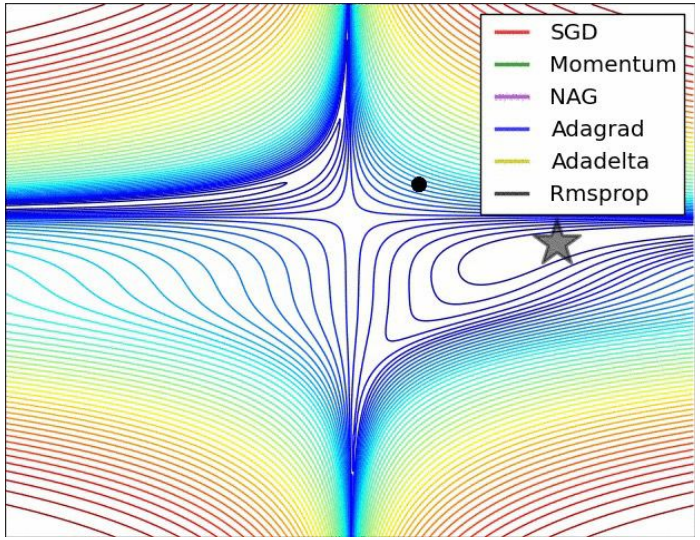


Lecture 3: Multi-Layer Perceptrons



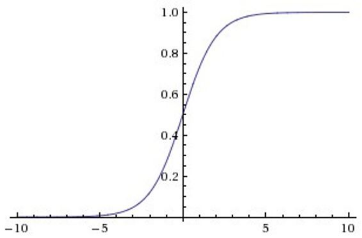
http://playground.tensorflow.org

Lecture 4: Training Deep Neural Networks



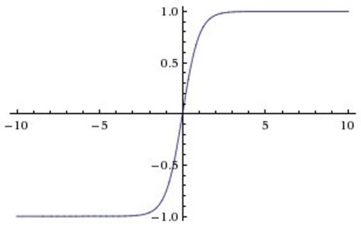
Optimizers

Sigmoid



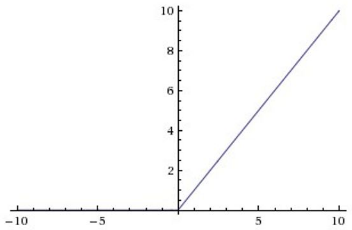
$\sigma(x) = 1/(1+e^{-x})$

tanh



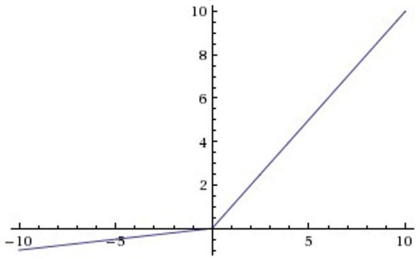
tanh(x)

ReLU



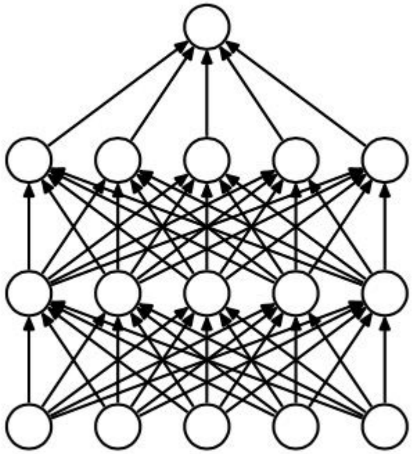
max(0,x)

Leaky ReLU

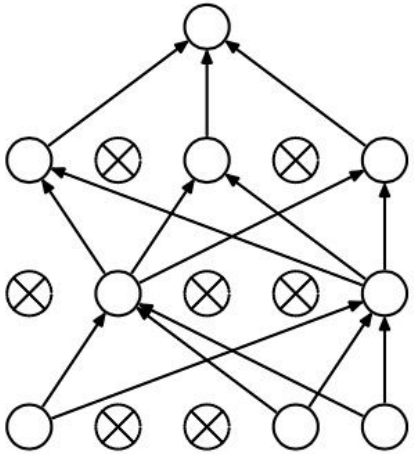


max(0.1x, x)

Activation Functions



(a) Standard Neural Net



(b) After applying dropout.

Dropout

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

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

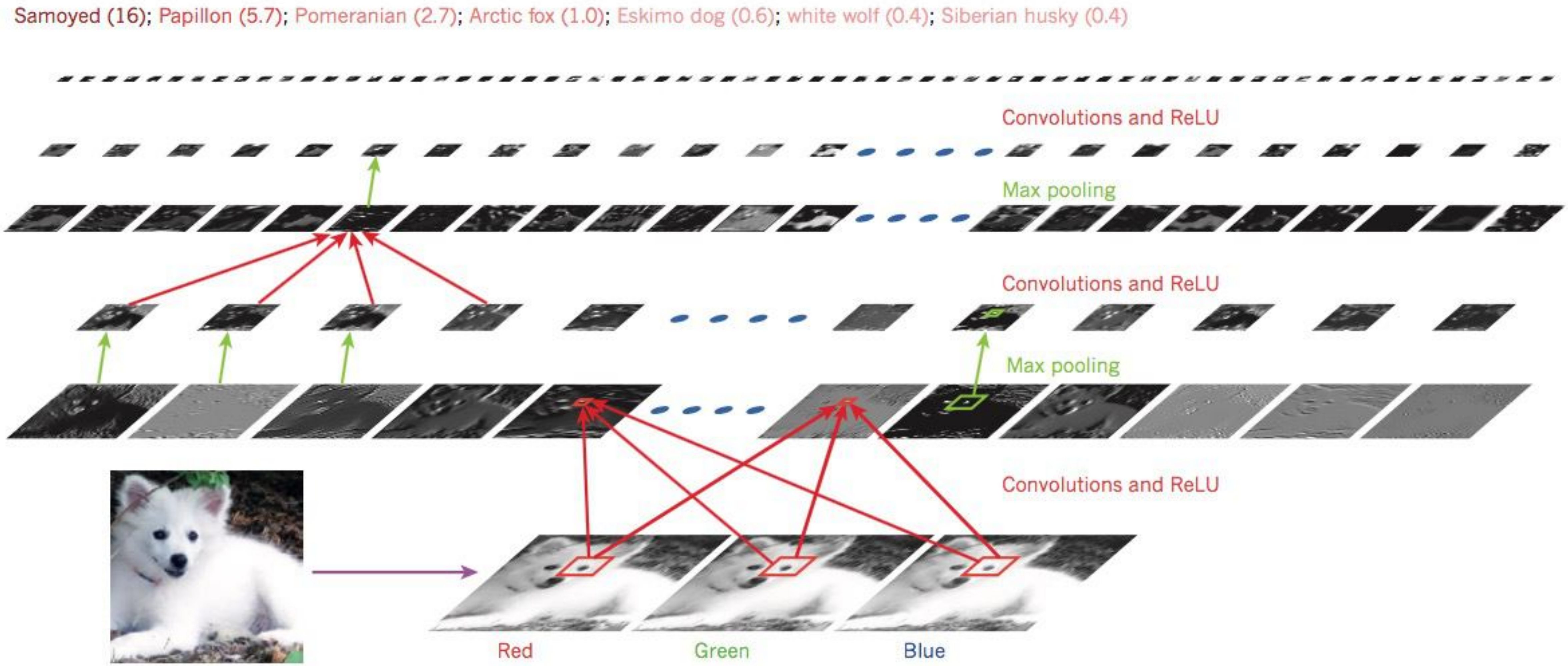
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

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

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

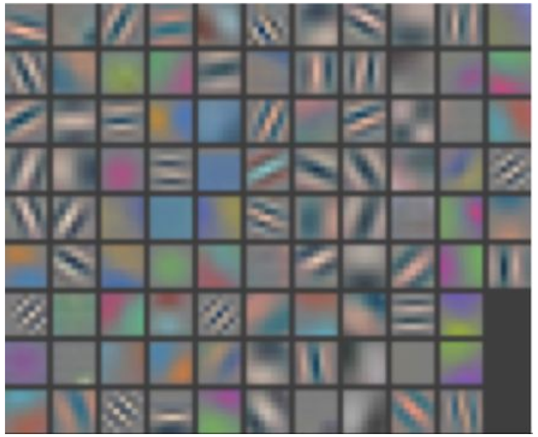
Batch Normalization

Lecture 5: Convolutional Neural Networks

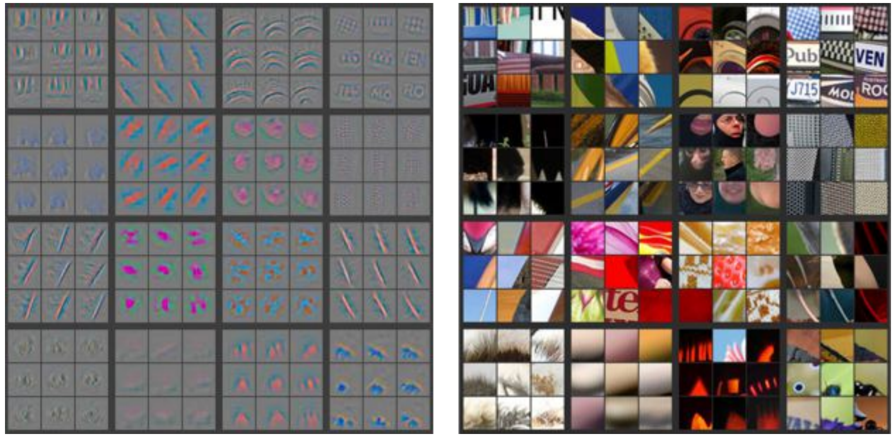


Y. LeCun, Y. Bengio, G. Hinton, **"Deep Learning"**, Nature, Vol. 521, 28 May 2015

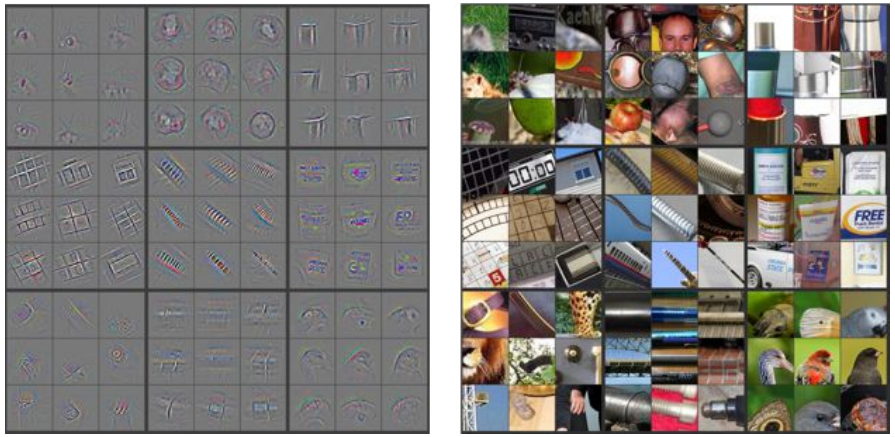
Lecture 6: Understanding and Visualizing CNNs



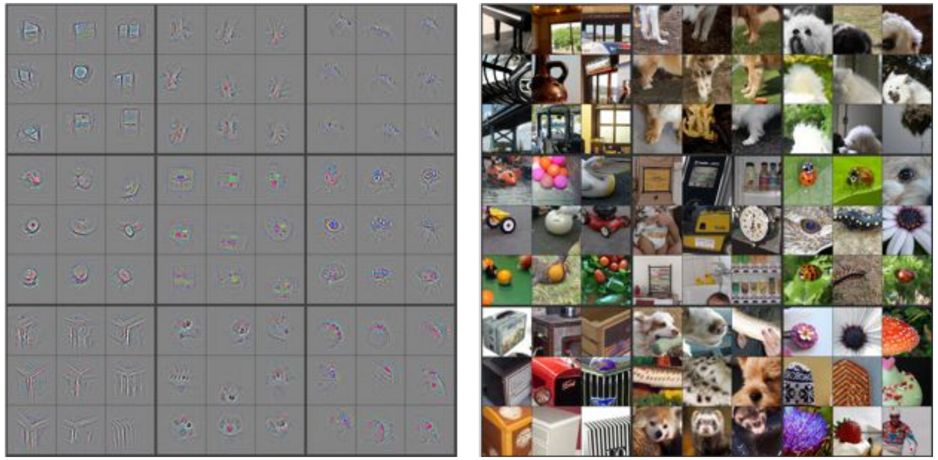
Layer 1



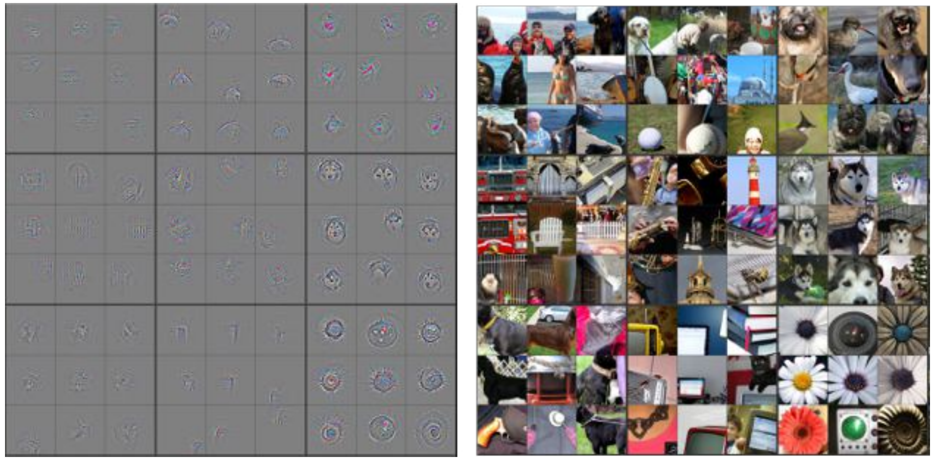
Layer 2



Layer 3



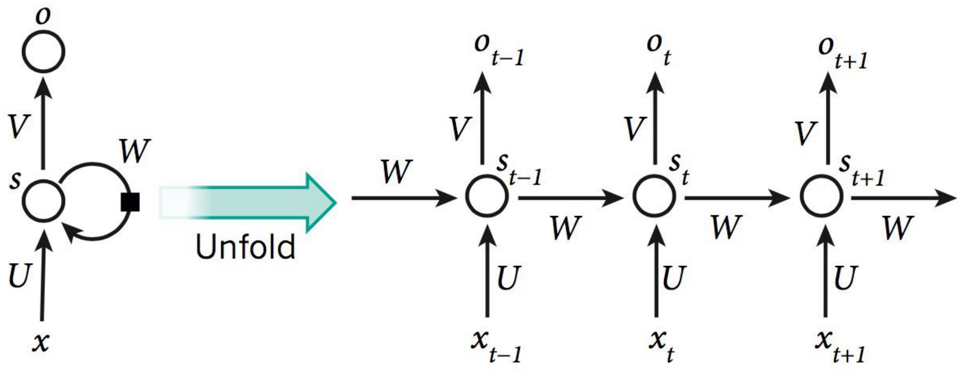
Layer 4



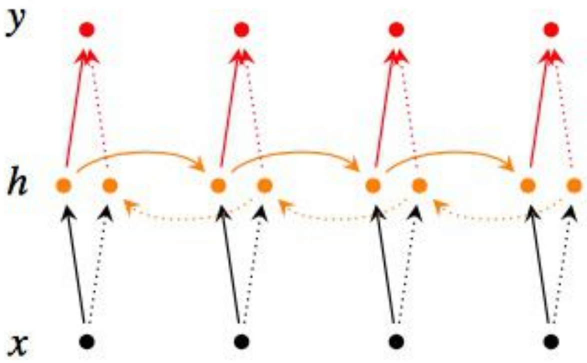
Layer 5

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

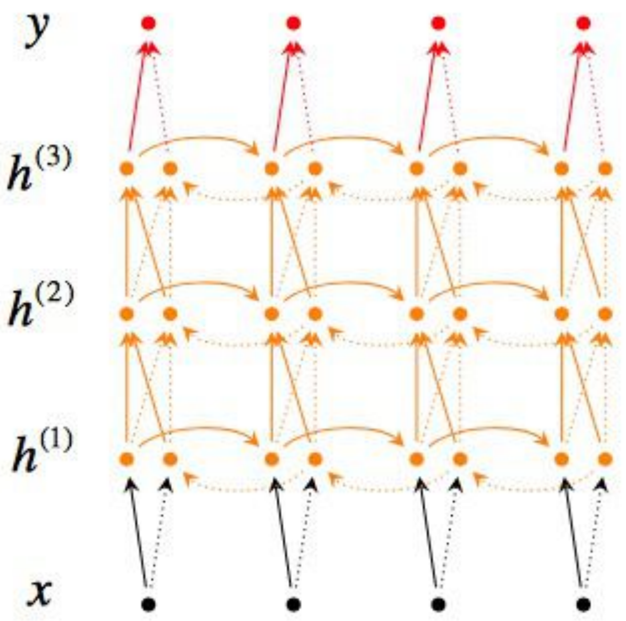
Lecture 7: Recurrent Neural Networks



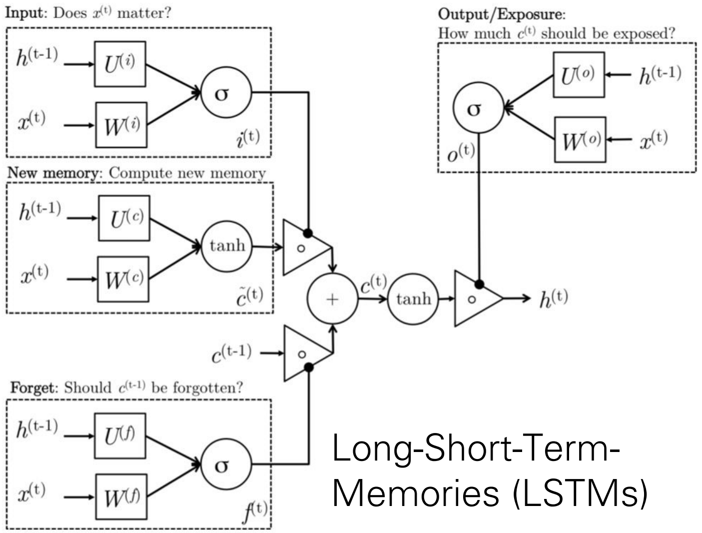
A Recurrent Neural Network (RNN)
(unfolded across time-steps)



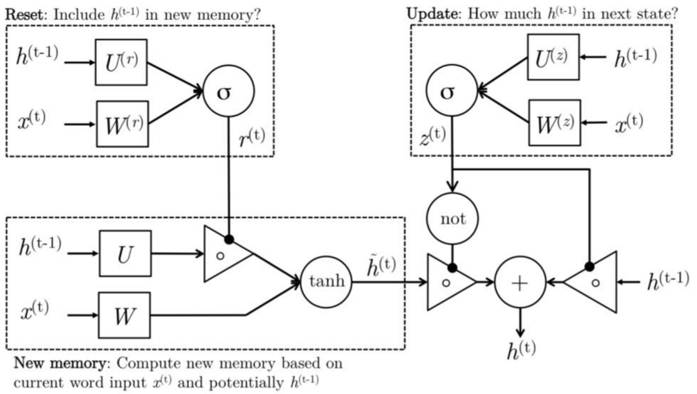
A bi-directional RNN



A deep bi-directional RNN



Long-Short-Term-Memories (LSTMs)



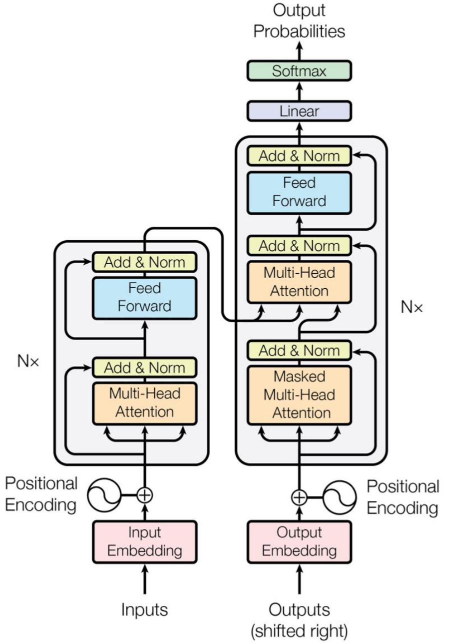
Gated Recurrent Units (GRUs)

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

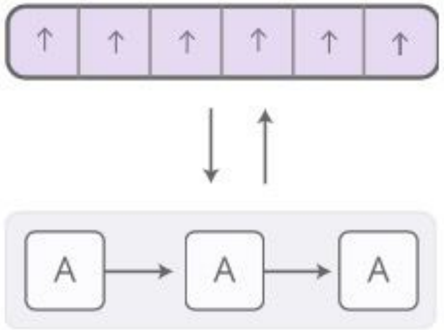
Lecture 8: Attention and Memory

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

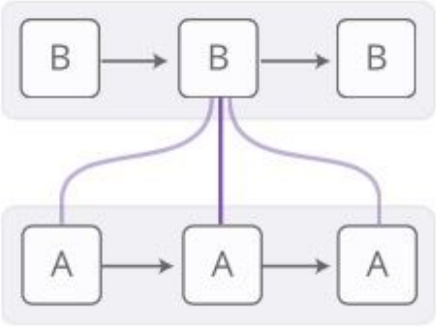
A group of people sitting on a boat in the water.



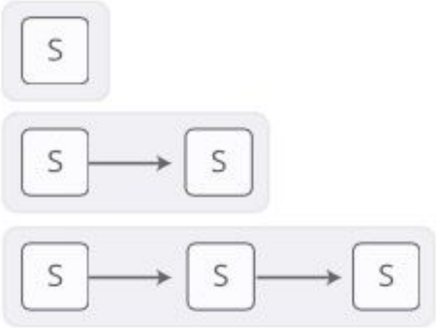
Transformer Architecture



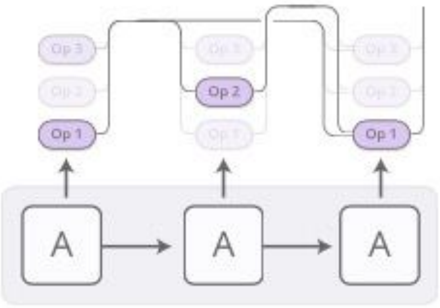
Neural Turing Machines



Attentional Interfaces



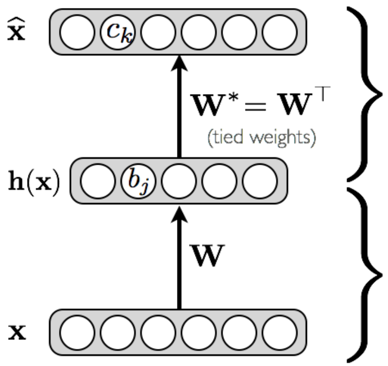
Adaptive Computation Time



Neural Programmers

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

Lecture 9: Autoencoders and Autoregressive Models



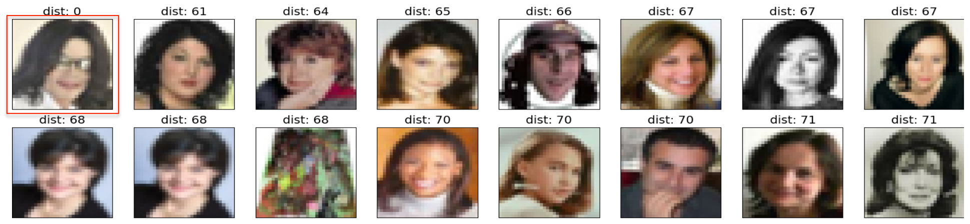
Decoder

$$\hat{\mathbf{x}} = o(\hat{\mathbf{a}}(\mathbf{x})) = \underbrace{\mathrm{sigm}}_{\text{For binary units}}(\mathbf{c} + \mathbf{W}^*\mathbf{h}(\mathbf{x}))$$

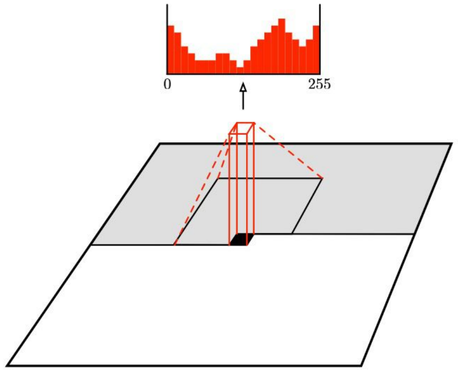
Encoder

$$\mathbf{h}(\mathbf{x}) = g(\mathbf{a}(\mathbf{x})) = \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})$$

retrieved using 256 bit codes

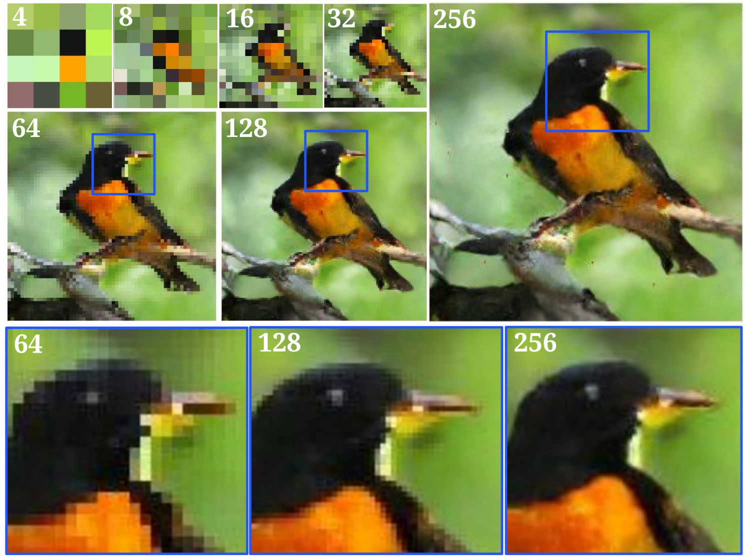


retrieved using Euclidean distance in pixel intensity space



PixelCNN

Class conditioned samples generated by PixelCNN

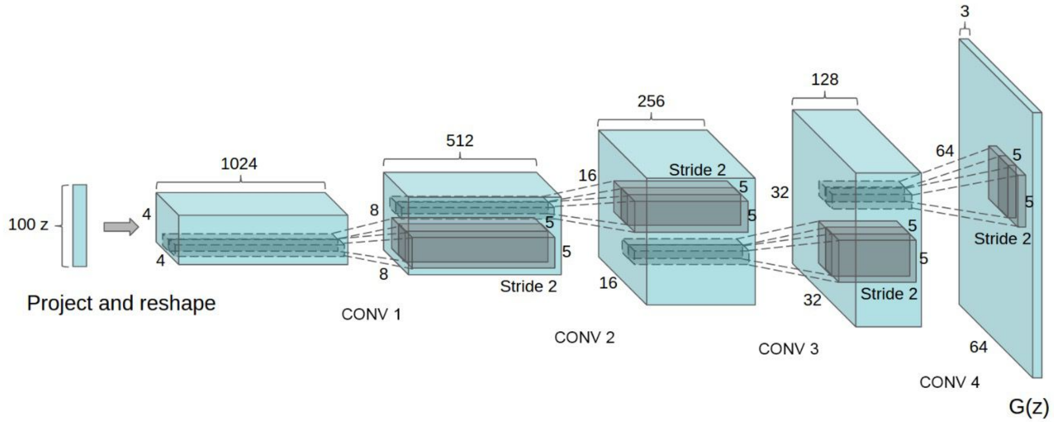


Text-to-image synthesis with Parallel Multiscale PixelCNNs

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

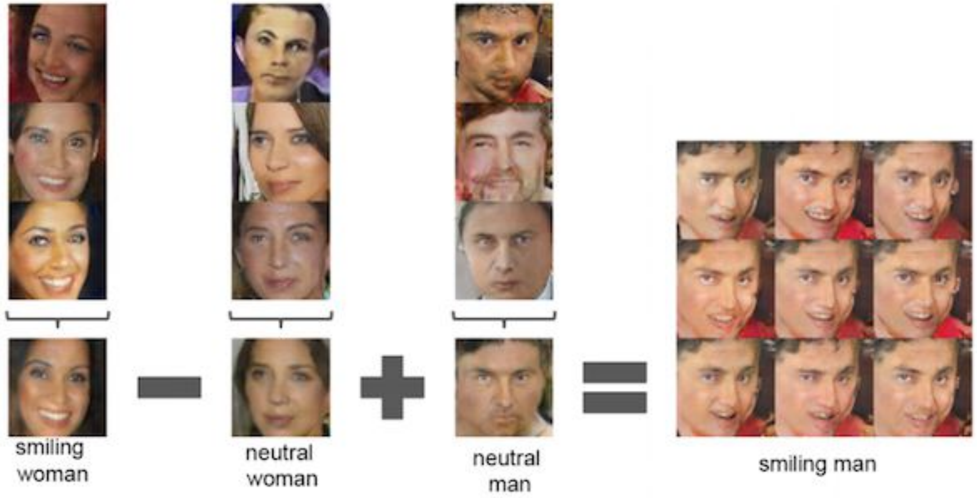
A. Krizhevsky and G. E. Hinton, "**Using Very Deep Autoencoders for Content-Based Image Retrieval**", ESANN 2011
A. van den Oord et al., "**Conditional Image Generation with PixelCNN Decoders**", NeurIPS 2016
S. Reed et al., "**Parallel Multiscale Autoregressive Density Estimation**", ICML 2017

Lecture 10: Generative Adversarial Networks



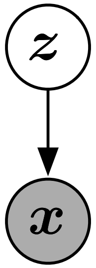
$$\min_{\theta} \max_{\omega} \mathbb{E}_{x\sim Q}[\log D_{\omega}(x)] + \mathbb{E}_{x\sim P_{\theta}}[\log(1 - D_{\omega}(x))]$$

Class-conditioned samples generated by BigGAN

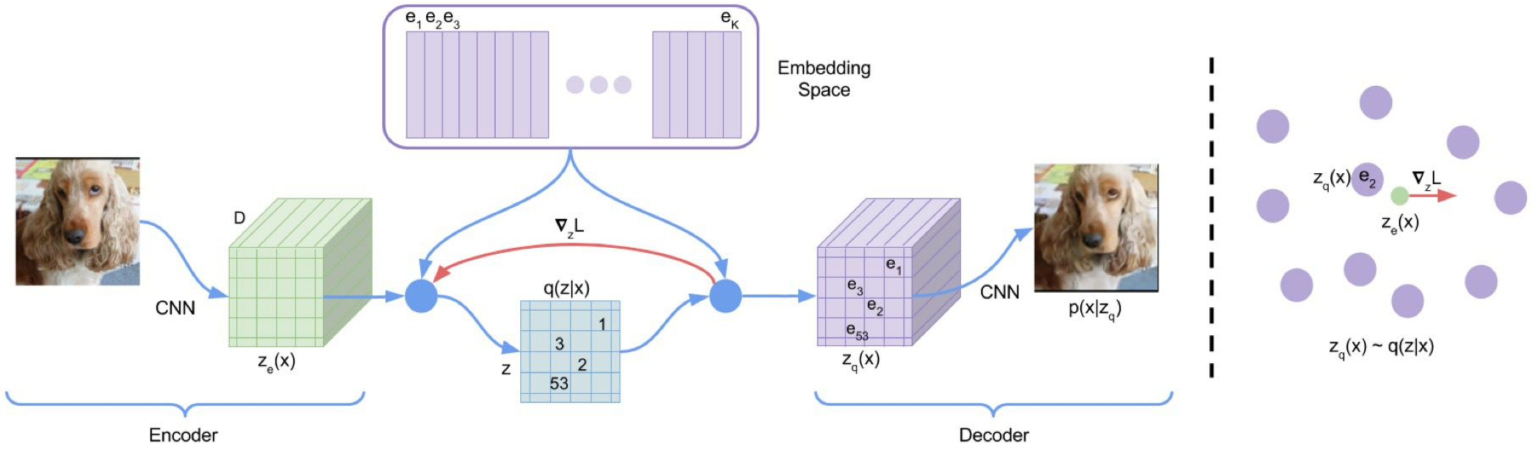


I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “**Generative adversarial nets**”, NIPS 2014.
A. Radford, L. Metz, and S. Chintala, “**Unsupervised representation learning with deep convolutional generative adversarial networks**”, ICLR 2016
L. 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 11: Vatiational Autoencoders



$$\log p(\boldsymbol{x}) \geq \log p(\boldsymbol{x}) - D_{\mathrm{KL}}\left(q(\boldsymbol{z}) \| p(\boldsymbol{z} \mid \boldsymbol{x})\right)$$
$$= \mathbb{E}_{\boldsymbol{z}\sim q} \log p(\boldsymbol{x}, \boldsymbol{z}) + H(q)$$



Vector Quantized- Variational AutoEncoder (VQ-VAE)

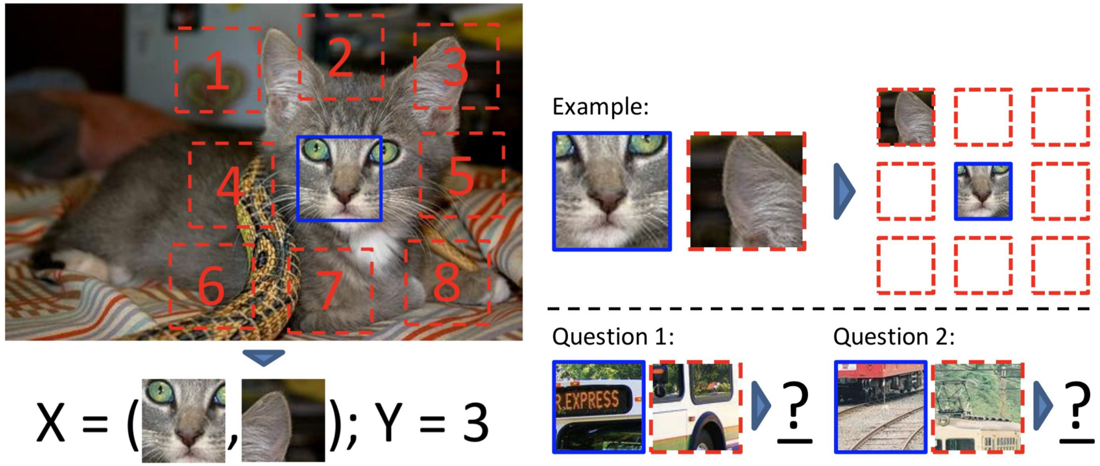
Synthetic images generated by VQ-VAE2

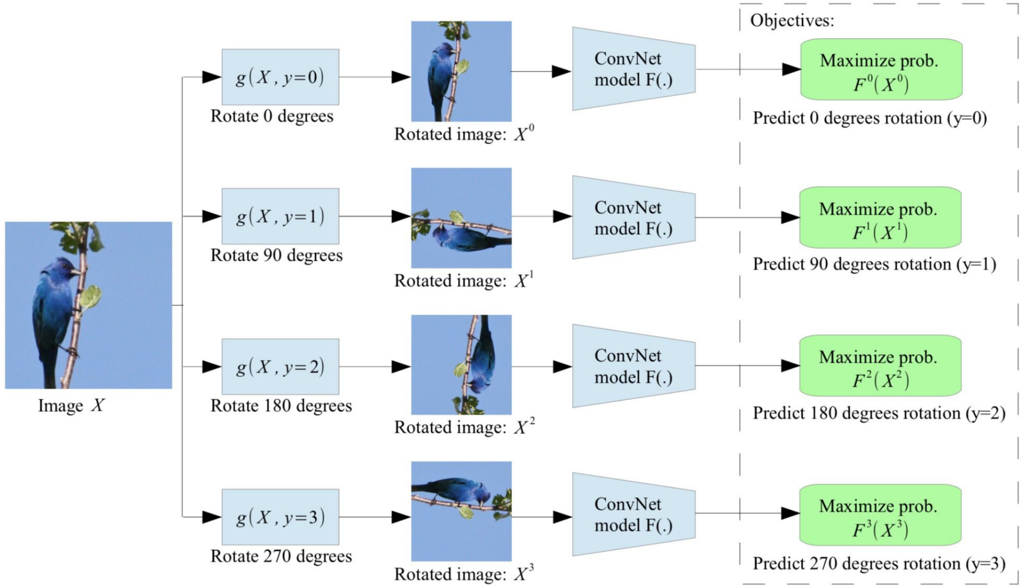
D. P. Kingma and M. Welling, “**Auto-encoding variational Bayes**”, ICLR 2014

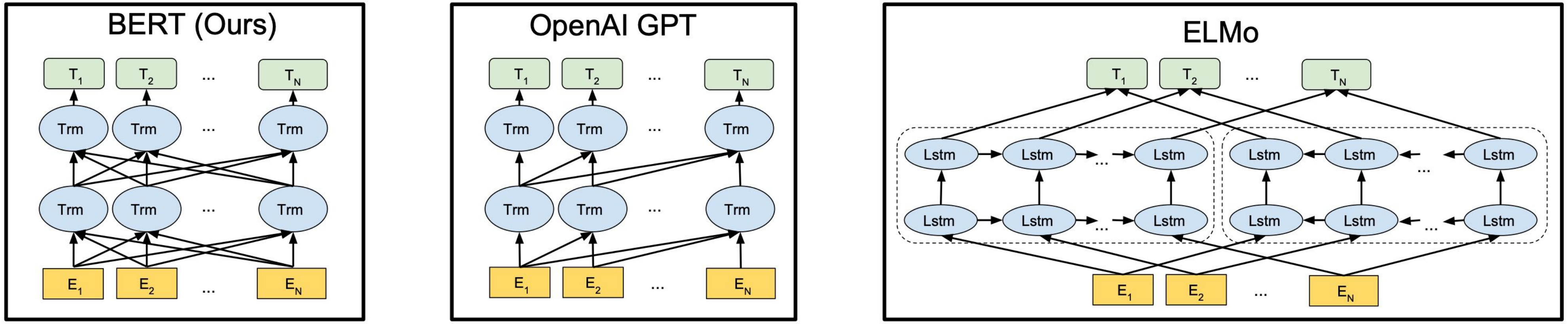
A. van den Oord, O. Vinyals, K. Kavukcuoglu, "**Neural Discrete Representation Learning**", NeurIPS 2017

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

Lecture 12: 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.

Schedule

W1 Introduction to Deep Learning

W2 Machine Learning Overview

W3 Multi-Layer Perceptrons
Practical 1 out

W4 Training Deep Neural Networks
Start of paper presentations

W5 Convolutional Neural Networks
Start of paper presentations
Practical 1 due, Practical 2 out

W6 Understanding and Visualizing CNNs
Project proposals due

W7 Recurrent Neural Networks

W8 Attention and Memory
Practical 2 due

W9 Autoencoders and Autoregressive Models

W9 *Progress Presentations*

W11 Generative Adversarial Networks
Project progress reports due

W12 Variational Autoencoders

W13 Self-supervised Learning

W14 *Final Project Presentations*

Paper Presentations

- (12 mins) One student will be responsible from providing an overview of the paper.
- (9 mins) One student will present the strengths of the paper.
- (9 mins) One student will discuss the weaknesses of the paper.
- (10 mins) General discussion

See the rubrics on the course web page for details

Practicals

- 2 practicals (8% each)
- Learning to train neural networks for different tasks
- Should be done individually

- **Late policy:** You have 5 slip days in the semester.

- **Tentative** Dates
  - Practical 1 Out: October 13th, Due: October 27th
  - Practical 2 Out: October 27th, Due: Nivember 17th

Course project

The students who need GPU resources for the course project are advised to use Google Colab.

- The course project gives students a chance to apply deep architectures discussed in class to a research oriented project.
- The students can work in pairs.
- The course project may involve
  - Design of a novel approach and its experimental analysis, or
  - An extension to a recent study of non-trivial complexity and its experimental analysis.
- Deliverables

- Proposals	November 3, 2021
- Project progress presentations	December 1, 2021
- Project progress reports	December 8, 2021
- Final project presentations	December 29, 2021
- Final reports	January 14, 2022

Lecture Overview

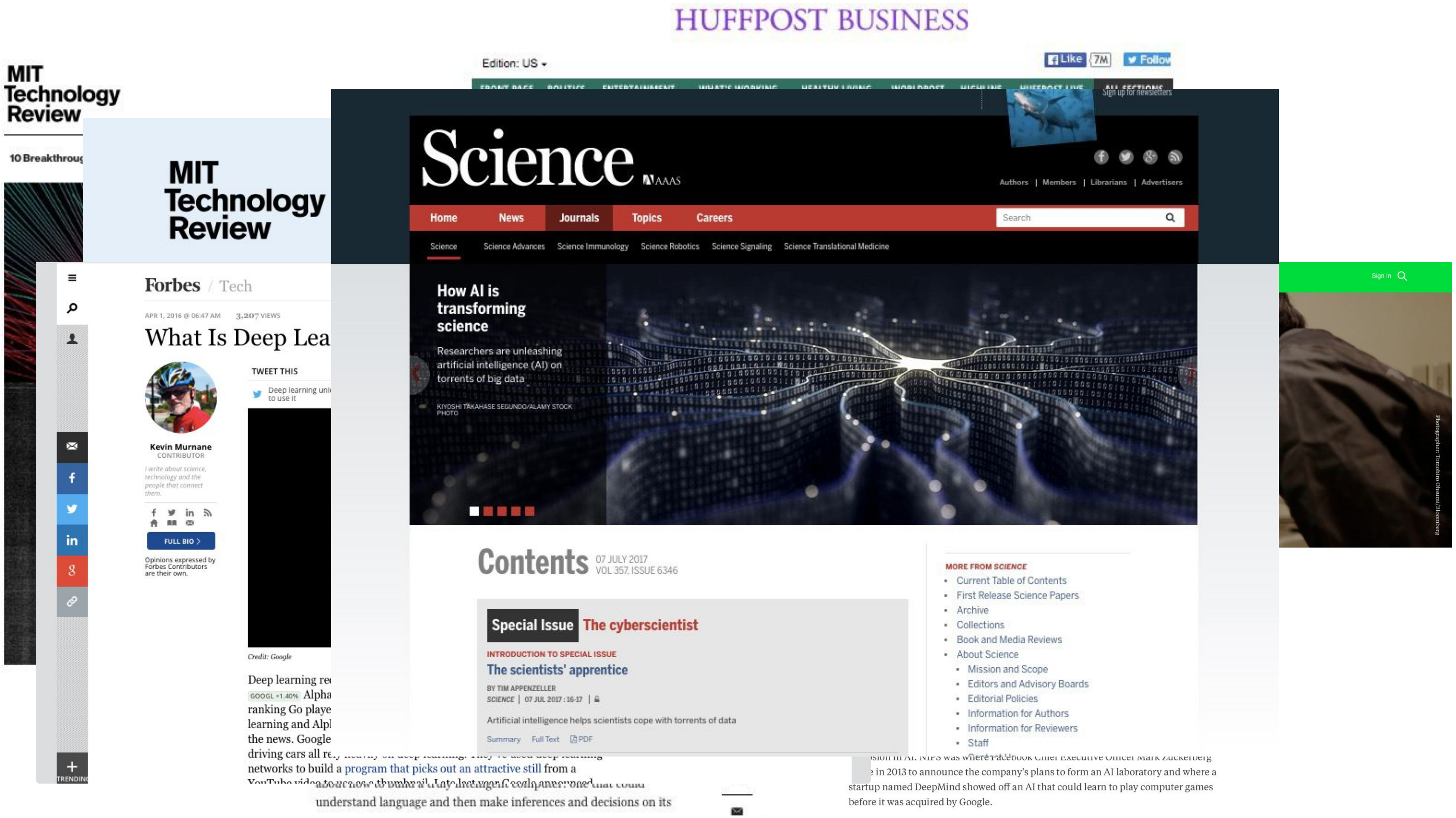
- what is deep learning
- a brief history of deep learning
- compositionality
- end-to-end learning
- distributed representations

Disclaimer: Some of the material and slides for this lecture were borrowed from

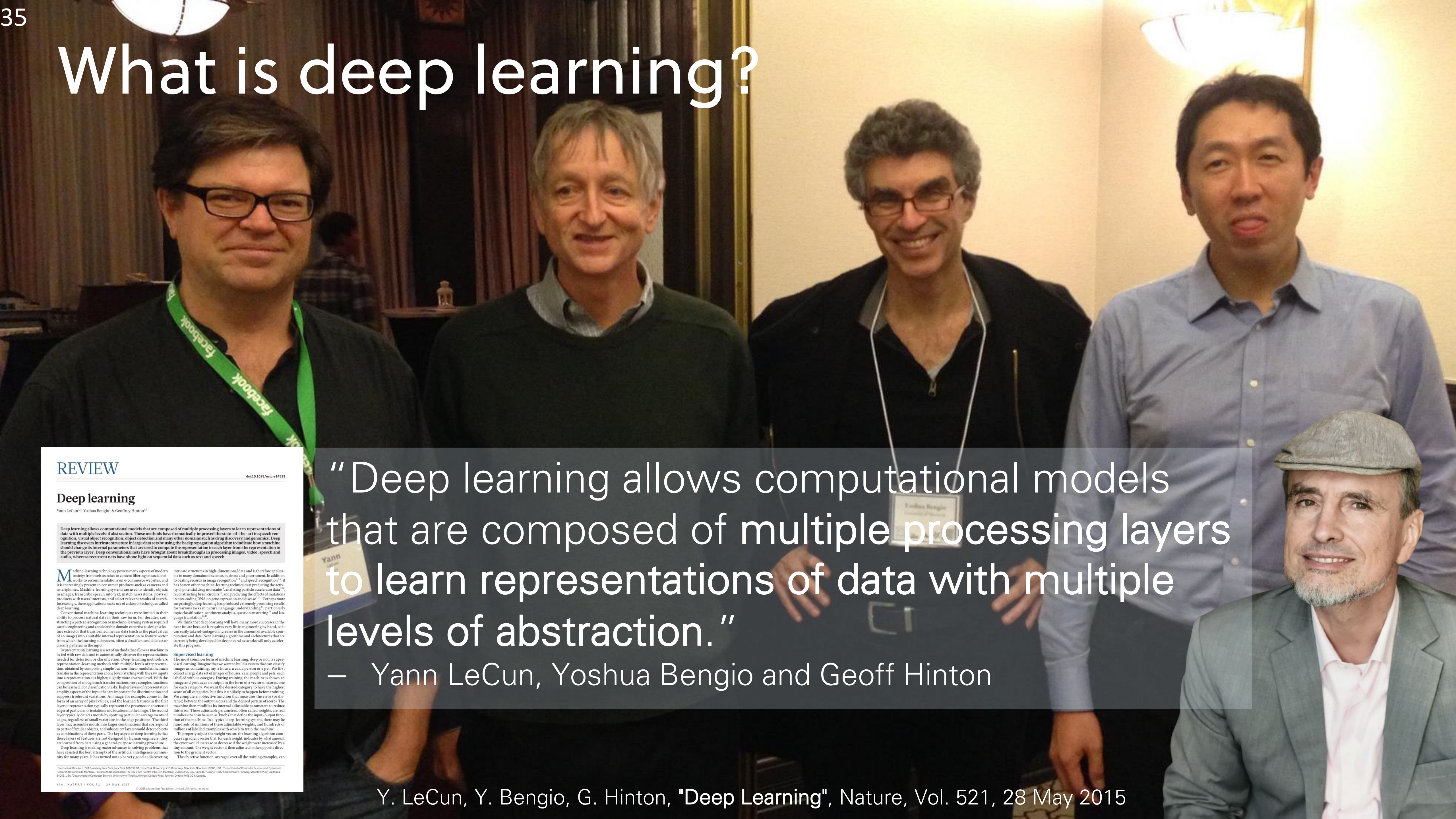
—Dhruv Batra's CS7643 class

—Yann LeCun's talk titled "Deep Learning and the Future of AI"

What is Deep Learning


HUFFPOST BUSINESS
Edition: US
MIT Technology Review
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.
FULL BIO
Opinions expressed by Forbes Contributors are their own.
TWEET THIS
Credit: Google
Science AAAS
Authors | Members | Librarians | Advertisers
Home News Journals Topics Careers
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
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
networks to build a program that picks out an attractive still from a
understand language and then make inferences and decisions on its
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

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

REVIEW
doi:10.1038/nature14539

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

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

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning.

Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a pattern-recognition or machine-learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input.

Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned. For classification tasks, higher layers of representation amplify aspects of the input that are important for discrimination and suppress irrelevant variations. An image, for example, comes in the form of an array of pixel values, and the learned features in the first layer of representation typically represent the presence or absence of edges at particular orientations and locations in the image. The second layer typically detects motifs by spotting particular arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure.

Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition[1–4] and speech recognition[5–7], it has beaten other machine-learning techniques at predicting the activity of potential drug molecules[8], analysing particle accelerator data[9,10], reconstructing brain circuits[11], and predicting the effects of mutations in non-coding DNA on gene expression and disease[12,13]. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding[14], particularly topic classification, sentiment analysis, question answering[15] and language translation[16,17].

We think that deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are currently being developed for deep neural networks will only accelerate this progress.

Supervised learning

The most common form of machine learning, deep or not, is supervised learning. Imagine that we want to build a system that can classify images as containing, say, a house, a car, a person or a pet. We first collect a large data set of images of houses, cars, people and pets, each labelled with its category. During training, the machine is shown an image and produces an output in the form of a vector of scores, one for each category. We want the desired category to have the highest score of all categories, but this is unlikely to happen before training. We compute an objective function that measures the error (or distance) between the output scores and the desired pattern of scores. The machine then modifies its internal adjustable parameters to reduce this error. These adjustable parameters, often called weights, are real numbers that can be seen as 'knobs' that define the input–output function of the machine. In a typical deep-learning system, there may be hundreds of millions of these adjustable weights, and hundreds of millions of labelled examples with which to train the machine.

To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the error would increase or decrease if the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector.

The objective function, averaged over all the training examples, can

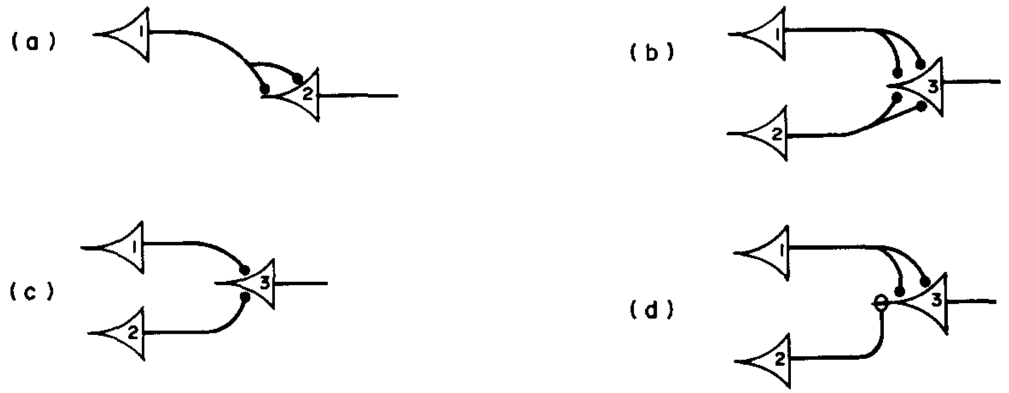
[1]Facebook AI Research, 770 Broadway, New York, New York 10003 USA. [2]New York University, 715 Broadway, New York, New York 10003, USA. [3]Department of Computer Science and Operations Research Université de Montréal, Pavillon André-Aisenstadt, PO Box 6128 Centre-Ville STN Montréal, Quebec H3C 3J7, Canada. [4]Google, 1600 Amphitheatre Parkway, Mountain View, California 94043, USA. [5]Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 3G4, Canada.

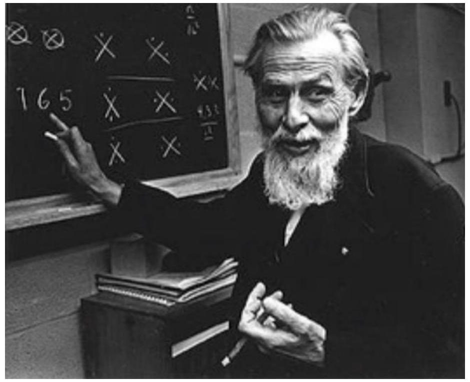
Y. LeCun, Y. Bengio, G. Hinton, **"Deep Learning"**, Nature, Vol. 521, 28 May 2015

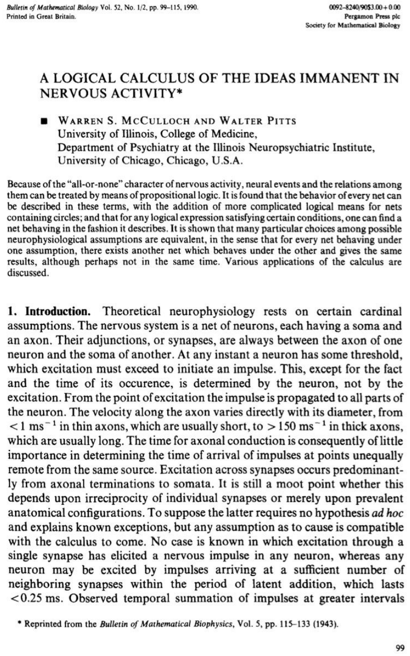
1943 – 2006: A Prehistory of Deep Learning

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







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

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

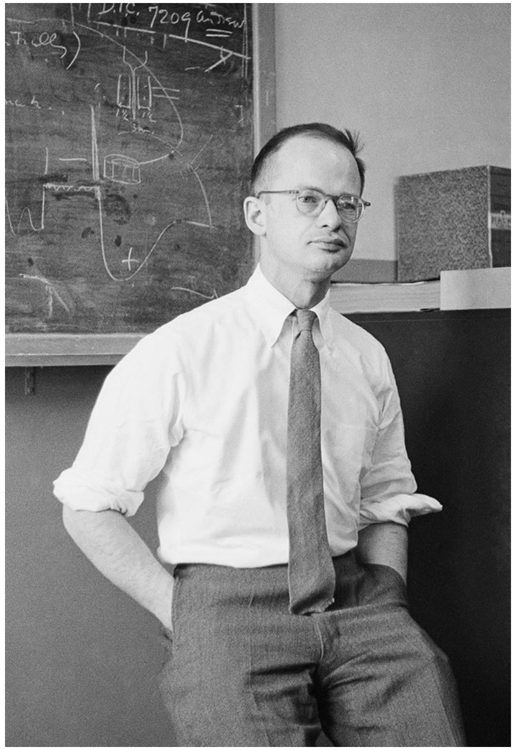
A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

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

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

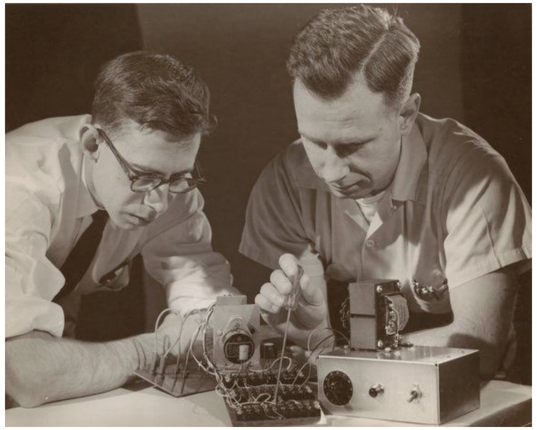
1. Introduction. Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which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

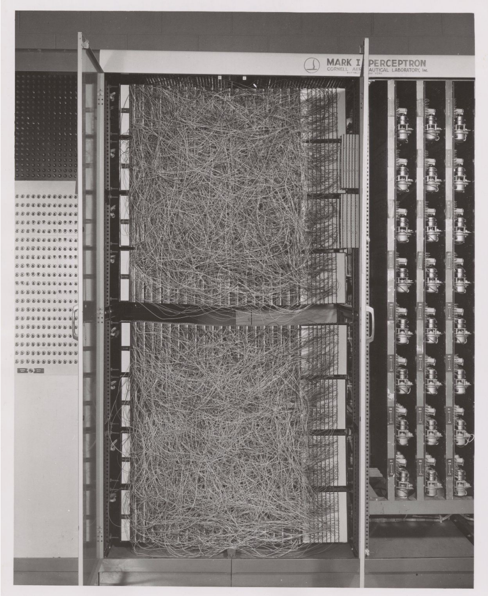
* Reprinted from the *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133 (1943).

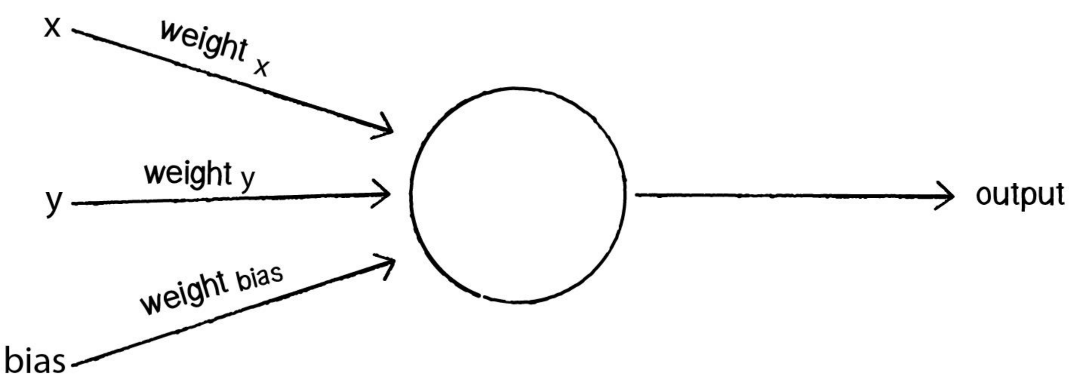


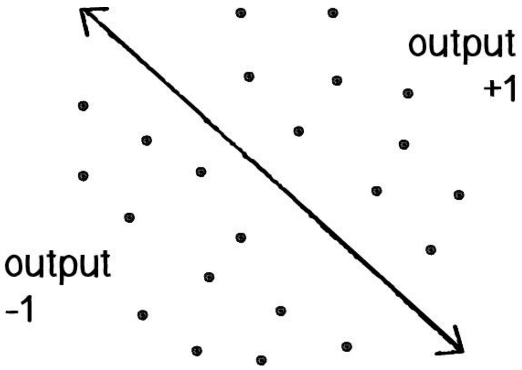
1958: Frank Rosenblatt's Perceptron

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







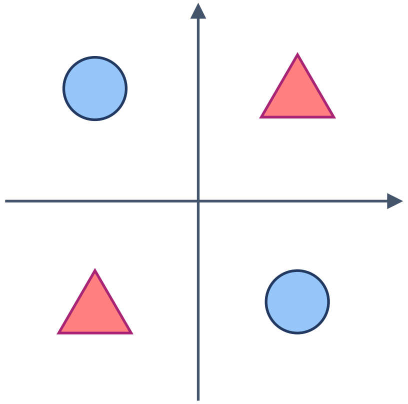


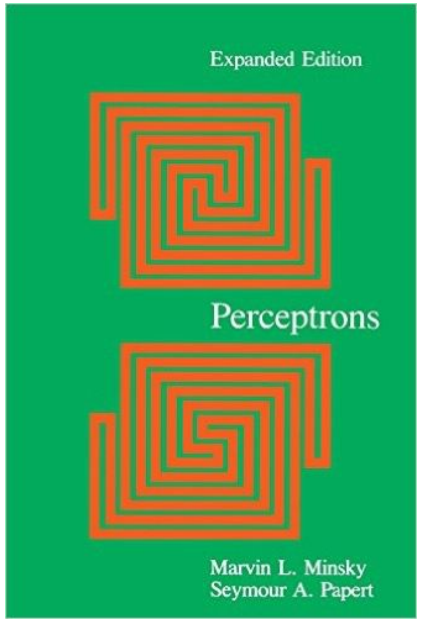
F. Rosenblatt, "**The perceptron: A probabilistic model for information storage and organization in the brain**", Psych. Review, Vol. 65, 1958

1969: Marvin Minsky and Seymour Papert

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

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

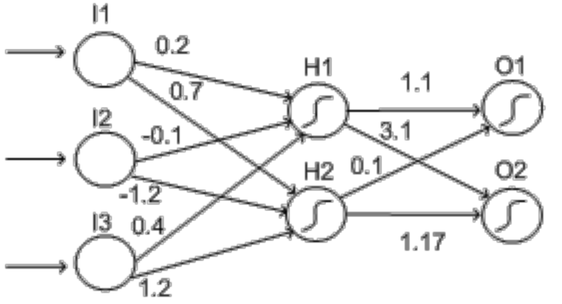




- Wrongly attributed as the reason behind the **AI winter**, a period of reduced funding and interest in AI research

1990s

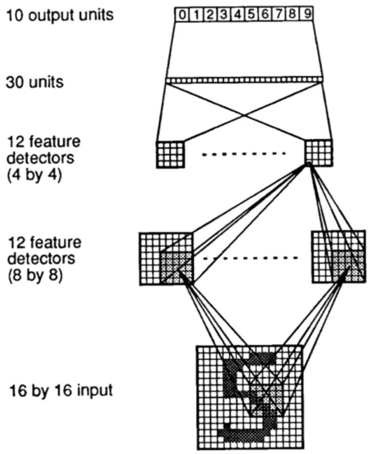
- **Multi-layer perceptrons** can theoretically learn any function (Cybenko, 1989; Hornik, 1991)




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

PAUL J. WERBOS

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

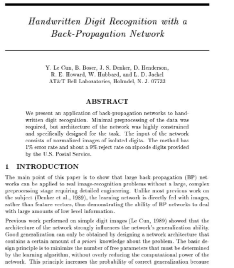



LETTERS TO NATURE

Learning representations by back-propagating errors

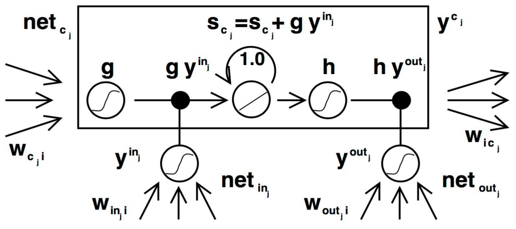
David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

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


Handwritten Digit Recognition with a Back-Propagation Network

ABSTRACT

1 INTRODUCTION



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)

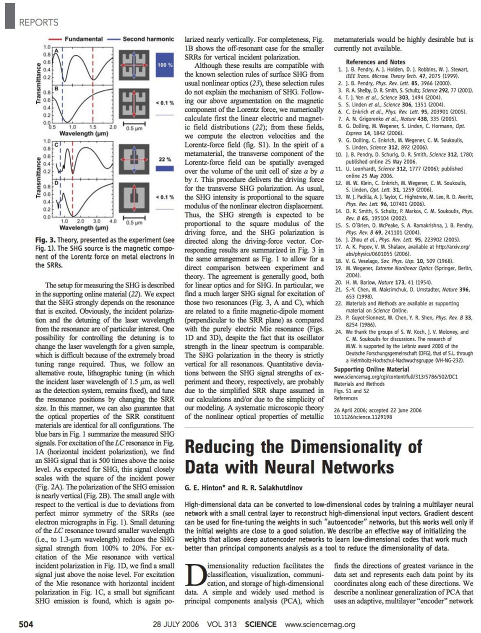
A major breakthrough in 2006

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.

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

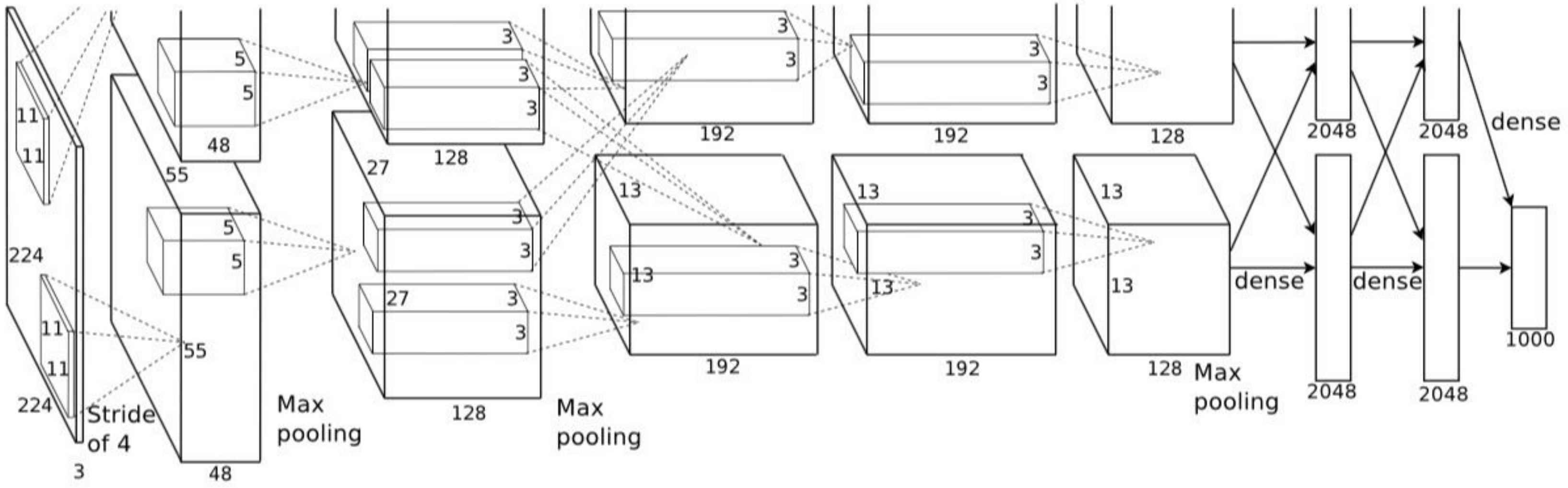
Image classification

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

ILSVRC 2012 Competition

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

CNN based, non-CNN based

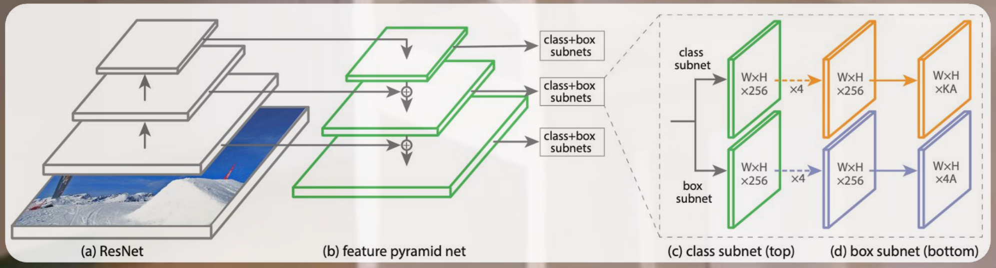


- The success of AlexNet, a deep convolutional network
  - 7 hidden layers (not counting some max pooling layers)
  - 60M parameters
- Combined several tricks
  - ReLU activation function, data augmentation, dropout

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

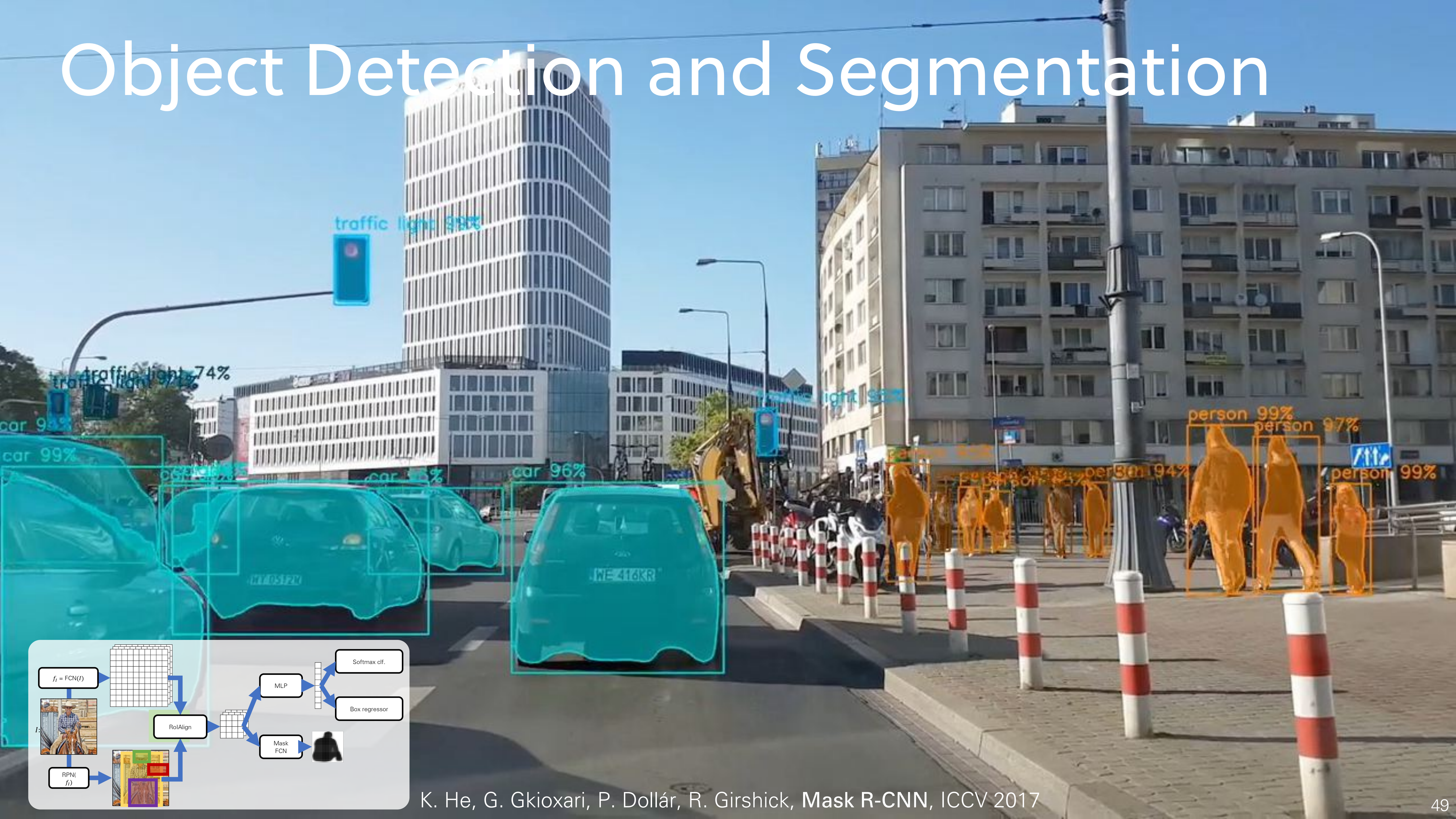
2012-Now
Some recent successes

Object Detection and Segmentation



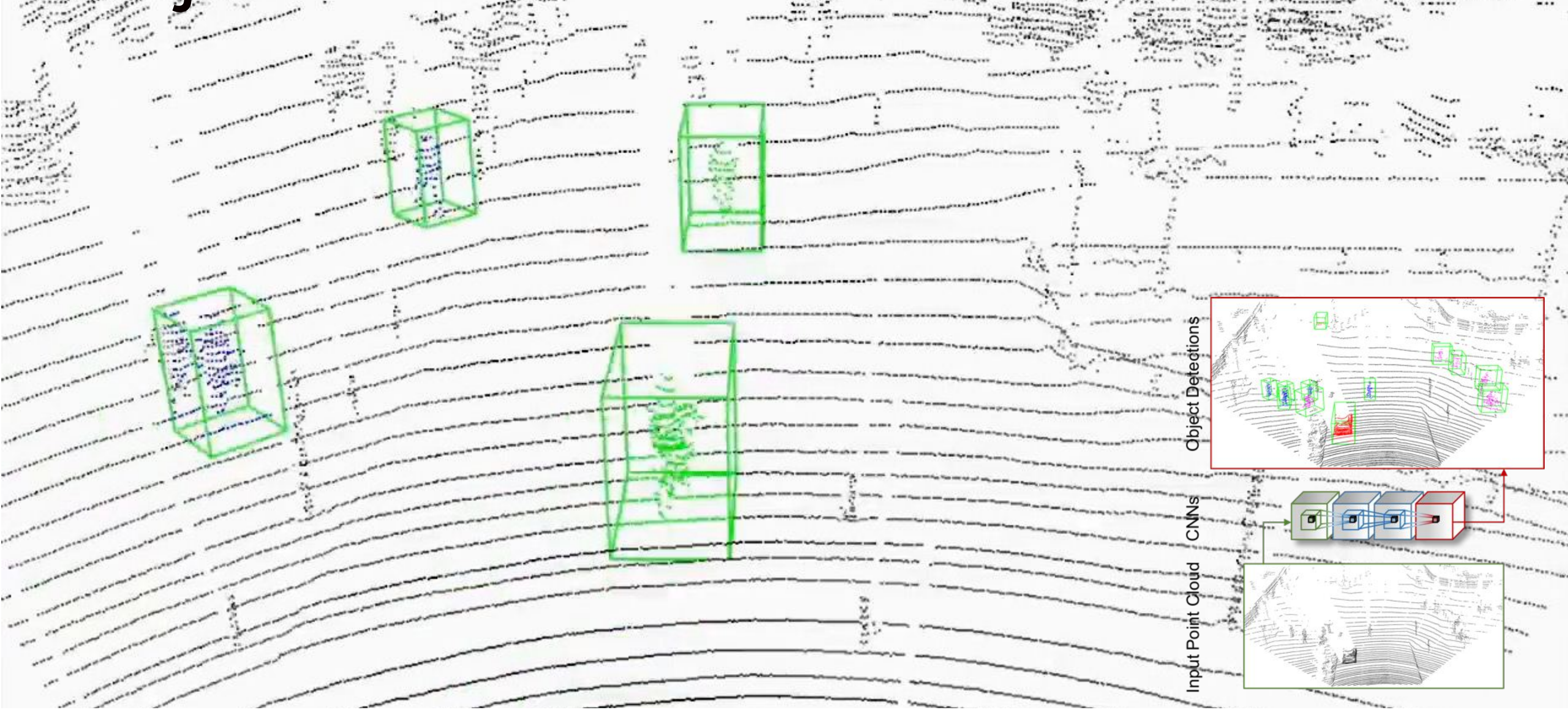
T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, **Focal Loss for Dense Object Detection**, ICCV 2017.

Object Detection and Segmentation



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

Object Detection in 3D Point Clouds

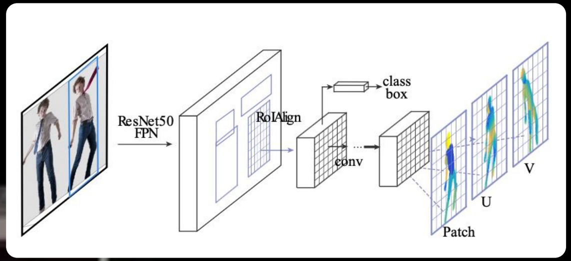


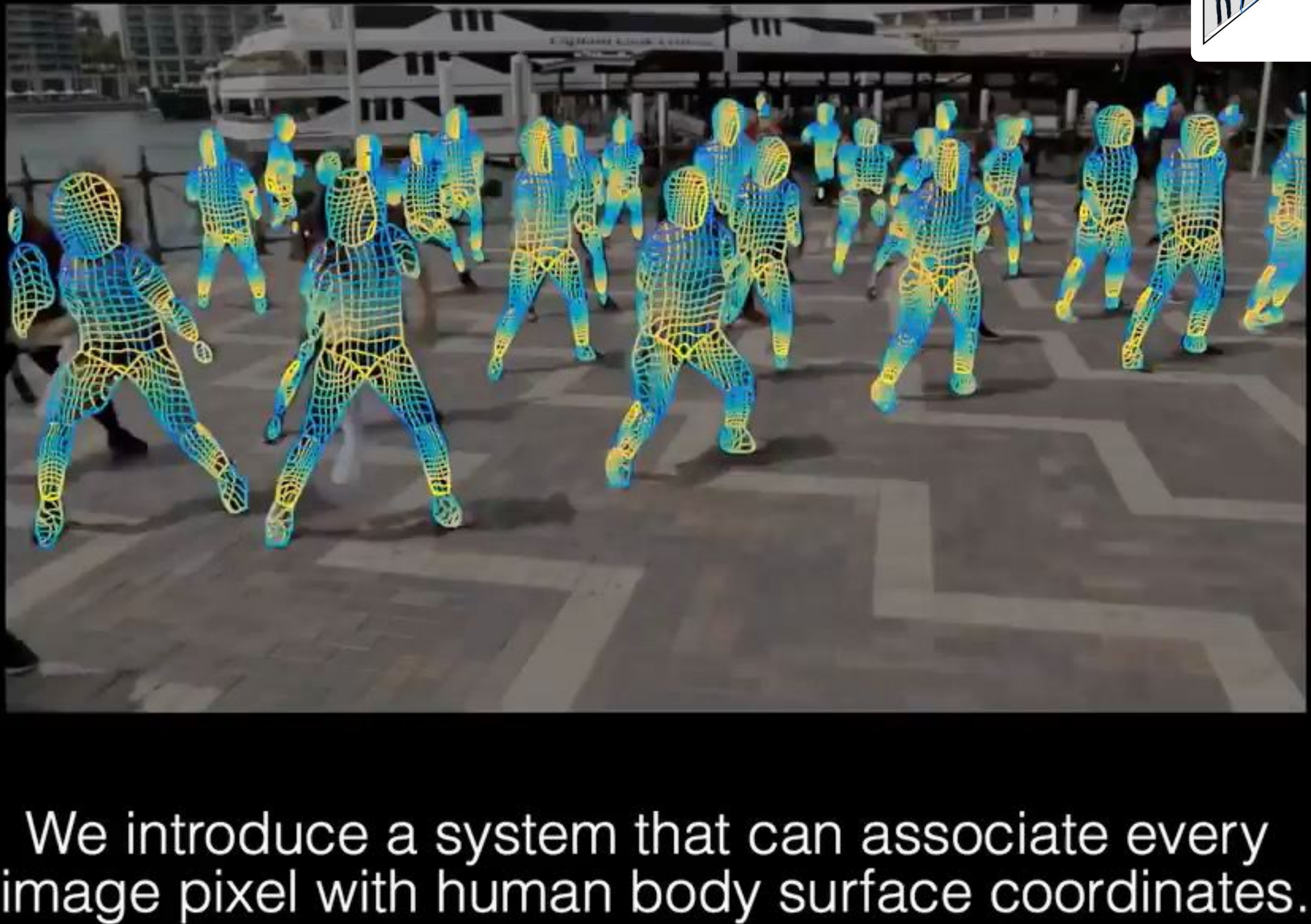
M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. **Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks.** ICRA 2017

Human Pose Estimation

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

Pose Estimation





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

Photo Style Transfer

F. Luan, S. Paris, E. Shechtman & K. Bala. **Deep Photo Style Transfer**. CVPR 2017

Photo Style Transfer

F. Luan, S. Paris, E. Shechtman & K. Bala. **Deep Photo Style Transfer**. CVPR 2017

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

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

Semantic Image Editing

Semantic Layout

L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, 2020

Semantic Image Editing

Winter

Prediction

L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, 2020

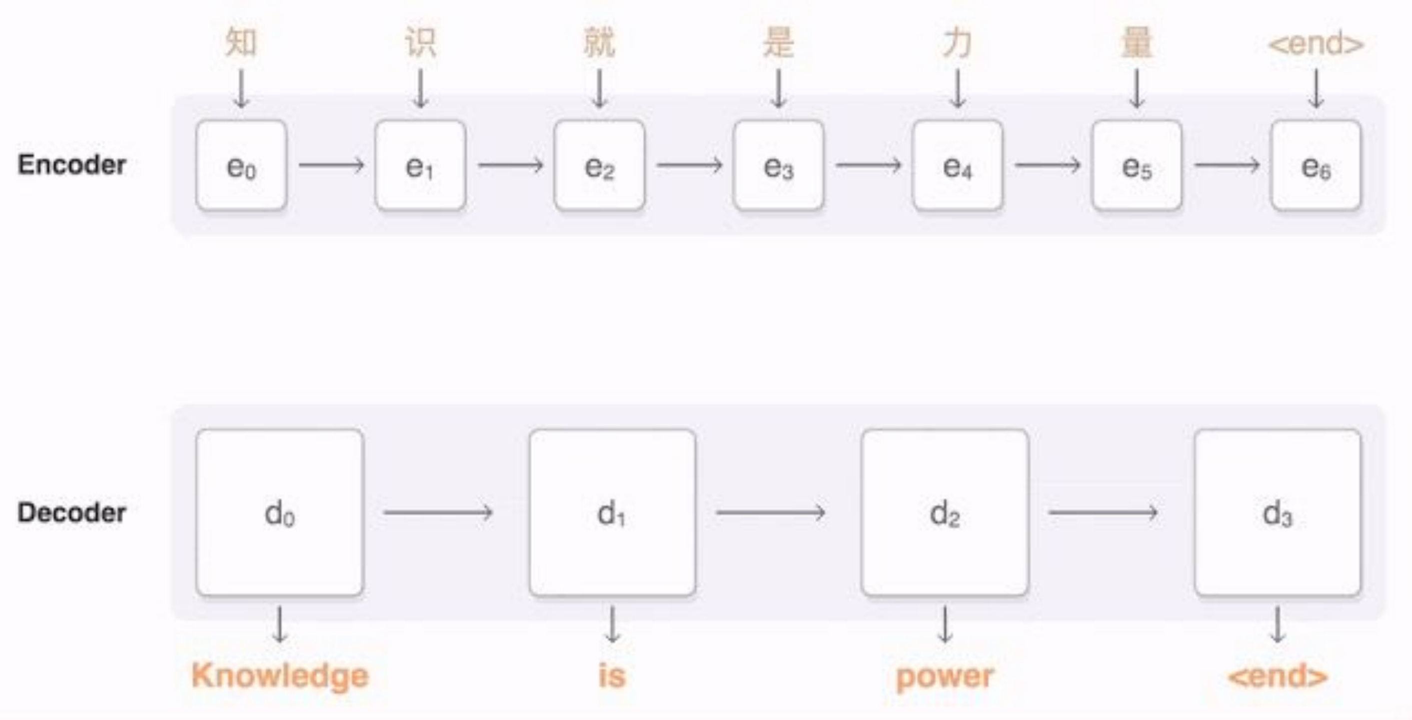
Semantic Image Editing

Spring
+
Clouds

Prediction

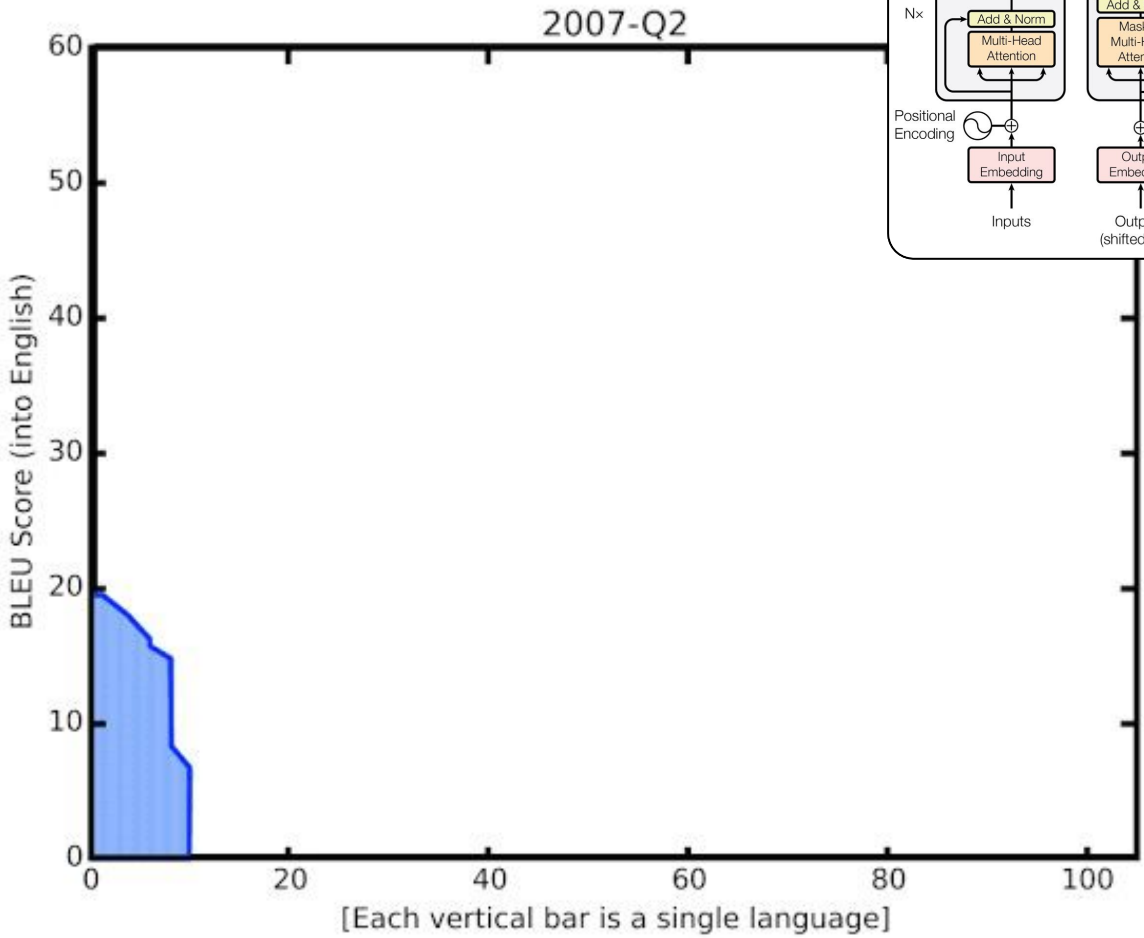
L. Karacan, Z. Akata, A. Erdem and E. Erdem. **Manipulation of Scene Attributes via Hallucination**. ACM Transactions on Graphics, 2020

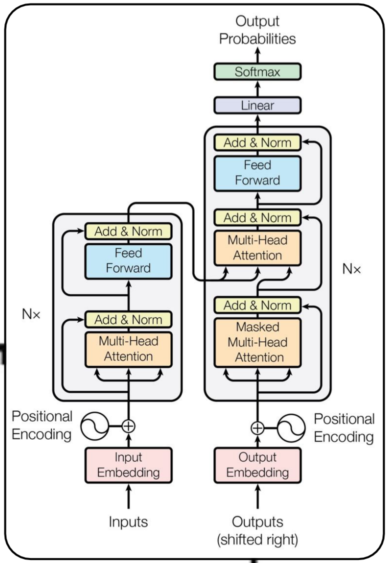
Machine Translation



D. Bahdanau, K. Cho, Y. Bengio. **Neural Machine Translation by Jointly Learning to Align and Translate.** ICLR 2015

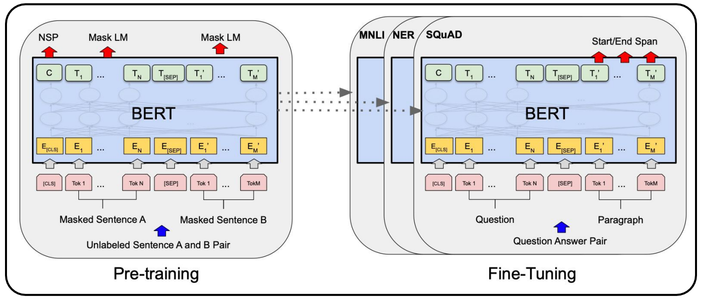
Machine Translation





A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, **Attention is All you Need**, NeurIPS 2017

Internet Search

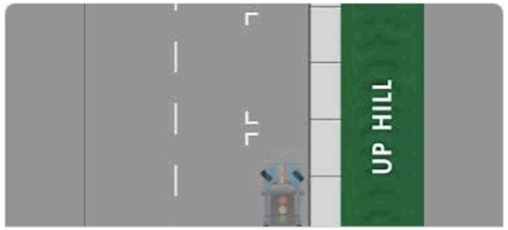


parking on a hill with no curb

BEFORE

9:00

google.com



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

Parking on a Hill - DriversEd.com

AFTER

9:00

google.com

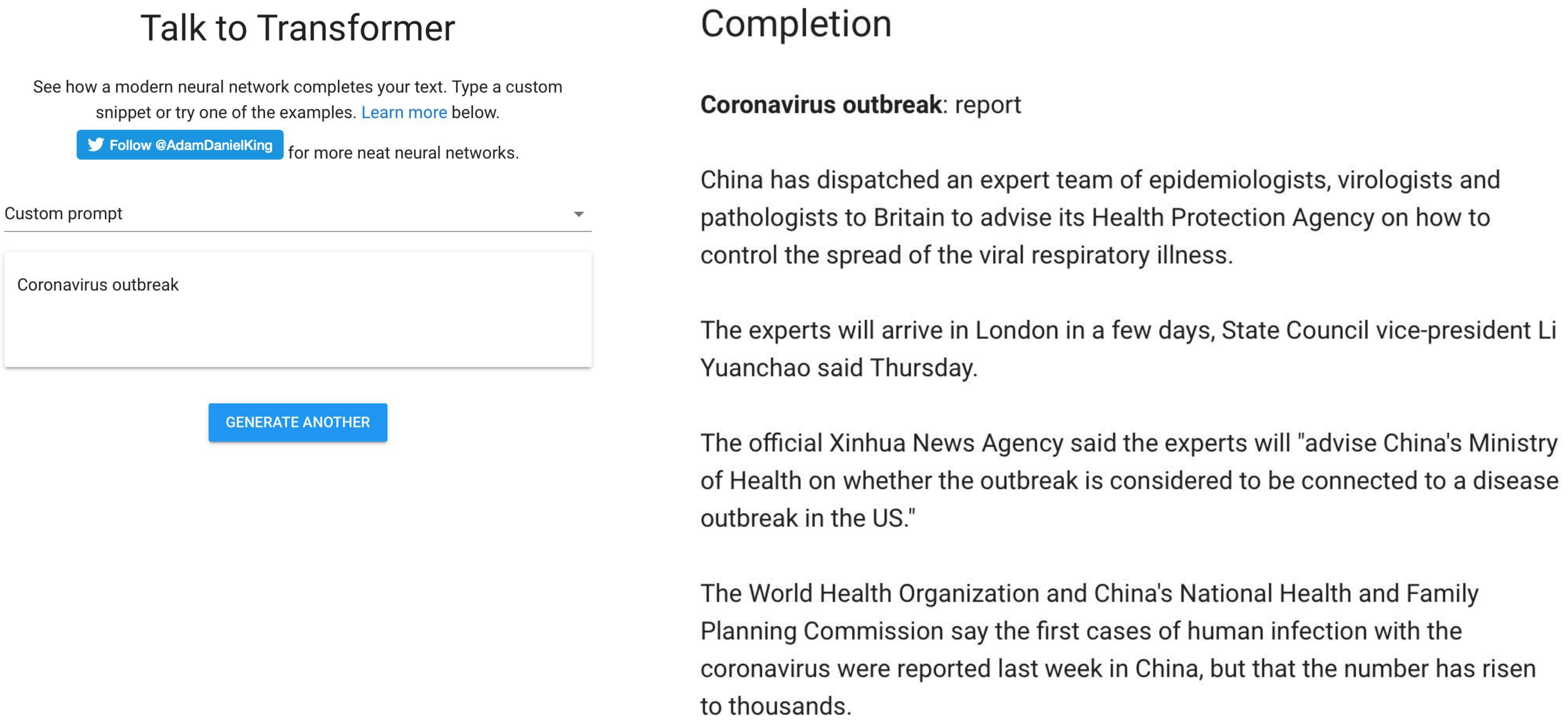
For either uphill or downhill **parking**, if there is no **curb**, turn the wheels toward the side of the road so the car will roll away from the center of the road if the brakes fail. When you park on a sloping driveway, turn the wheels so that the car will not roll into the street if the brakes fail.

Parking on a Hill

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

Language Modeling

https://talktotransformer.com



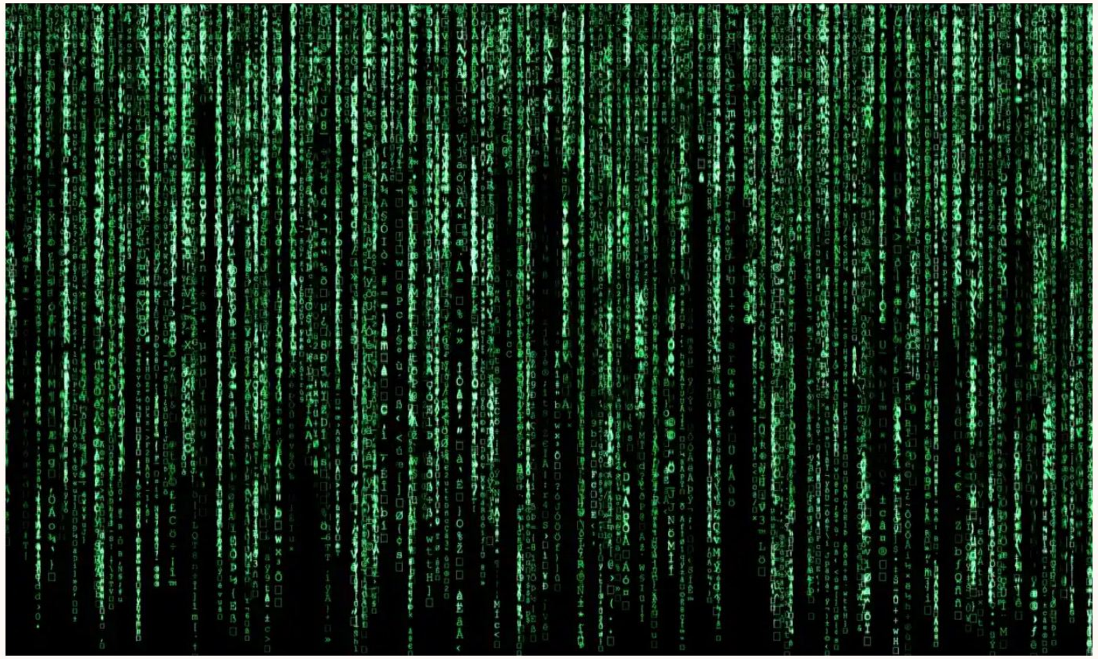
A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever. **Language Models are Unsupervised Multitask Learners**. 2019

Language Modeling

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

theguardian

Tue 8 Sep 2020 09.45



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

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

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

Read more

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

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

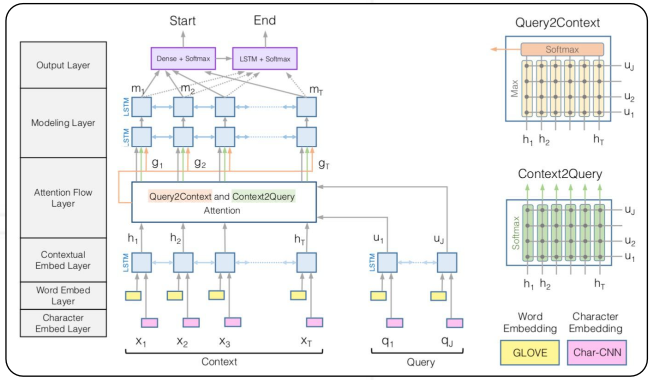
Tom B. Brown, Benjamin Mann, Nick Ryder et al., **Language Models are Few-Shot Learners**, NeurIPS 2020

Question Answering

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

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

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



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

Visual Question Answering

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

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

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

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

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

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

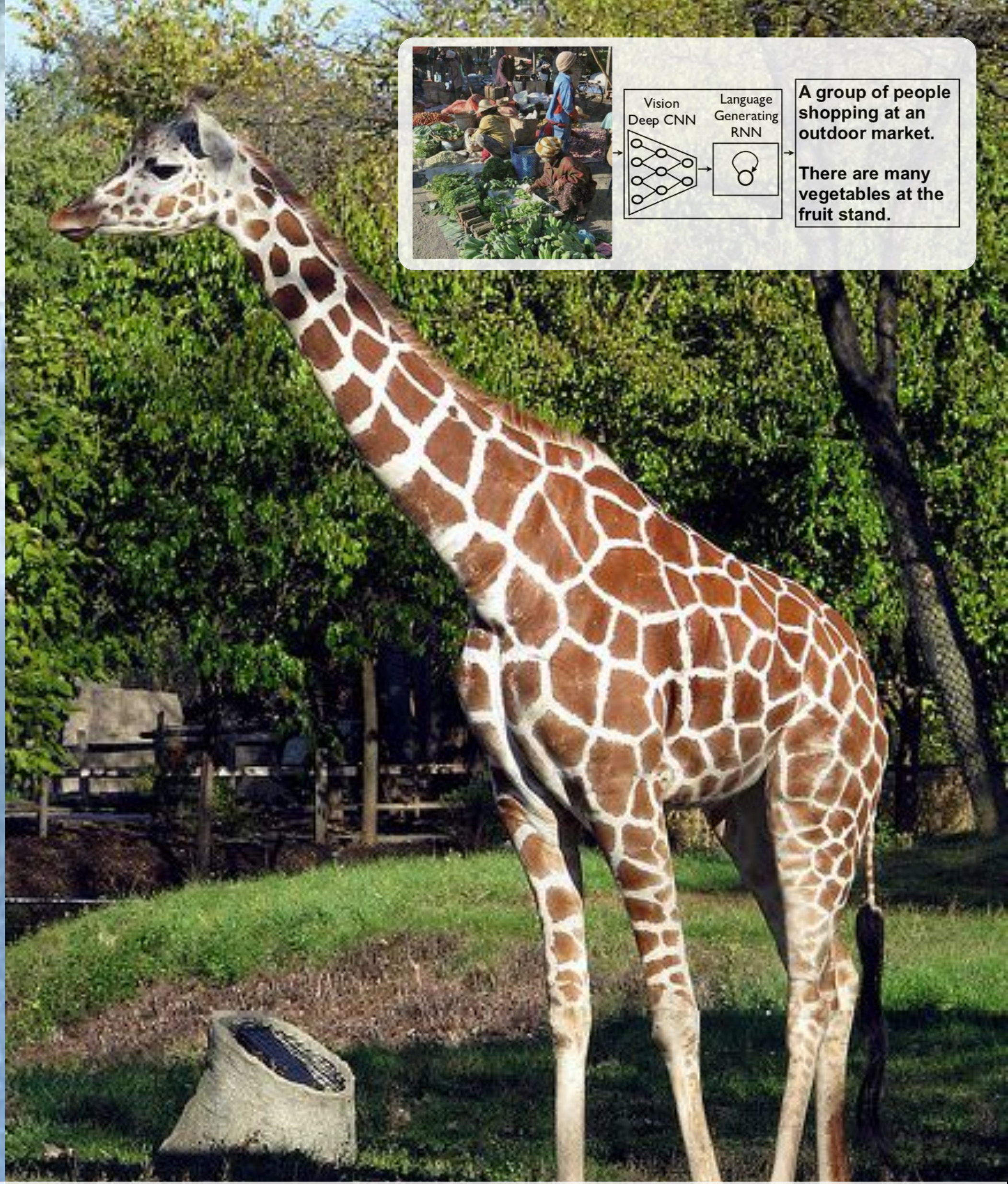
DAQUAR 585
What is the object on the chair?
Ground truth: pillow
IMG+BOW: clothes (0.37)
2-VIS+BLSTM: pillow (0.65)
LSTM: clothes (0.40)

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

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

Image Captioning

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



A giraffe standing in the grass next to a tree.

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

Image Captioning

Yarış pistinde virajı almakta olan bir yarış arabası

Video Captioning

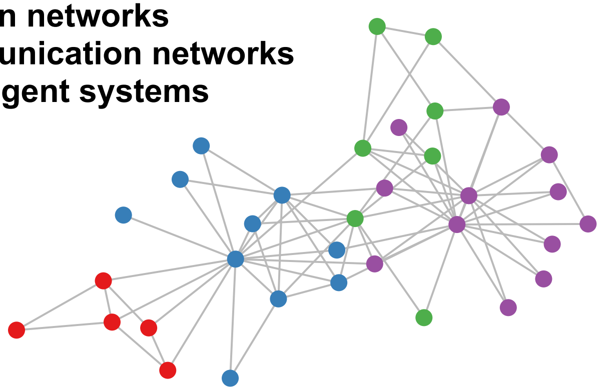
Bir adam bir gitar çalıyor

Bir kadın bir bıçakla sebze dilimliyor

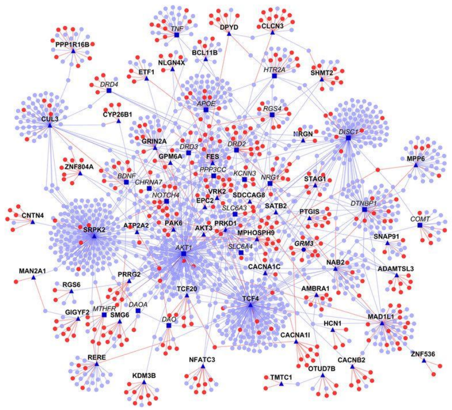
B. Çitamak et al. **MSVD-Turkish: a comprehensive multimodal video dataset for integrated vision and language research in Turkish.**

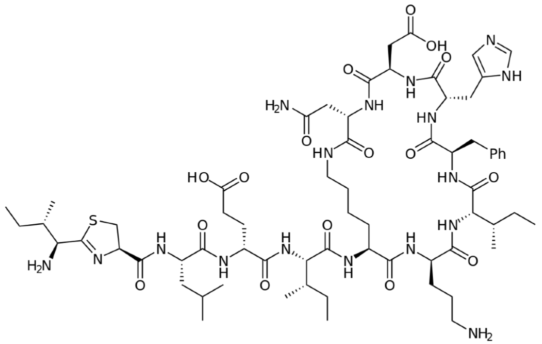
Graph Neural Networks

Social networks
Citation networks
Communication networks
Multi-agent systems

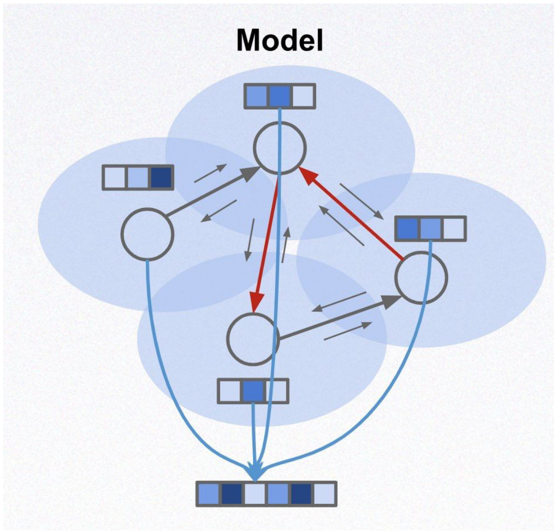


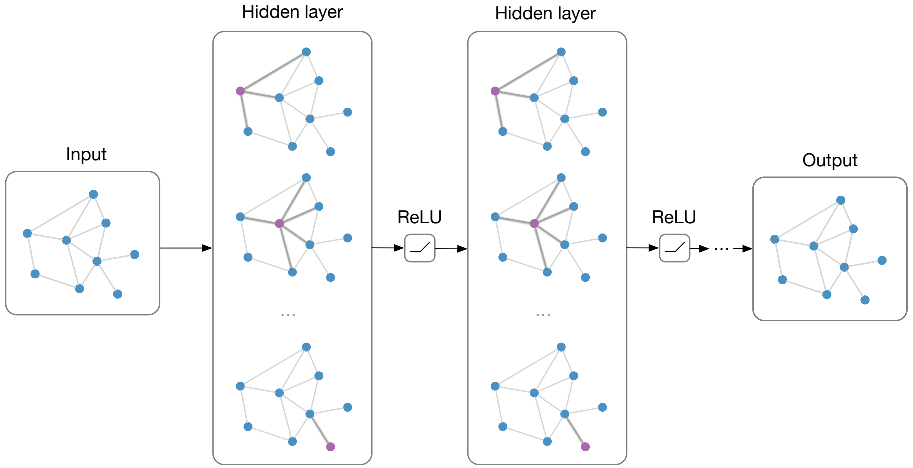
Protein interaction networks

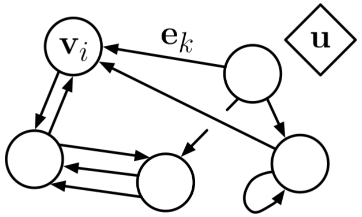


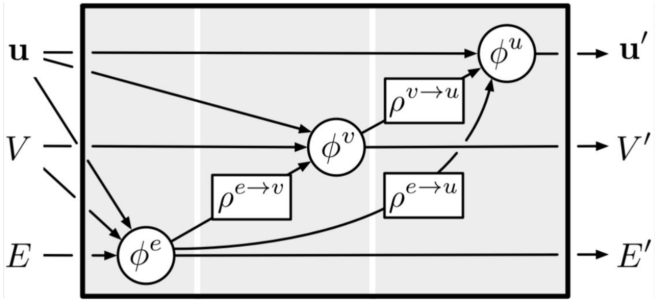


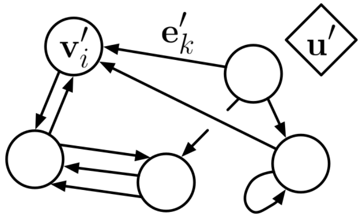
Molecules











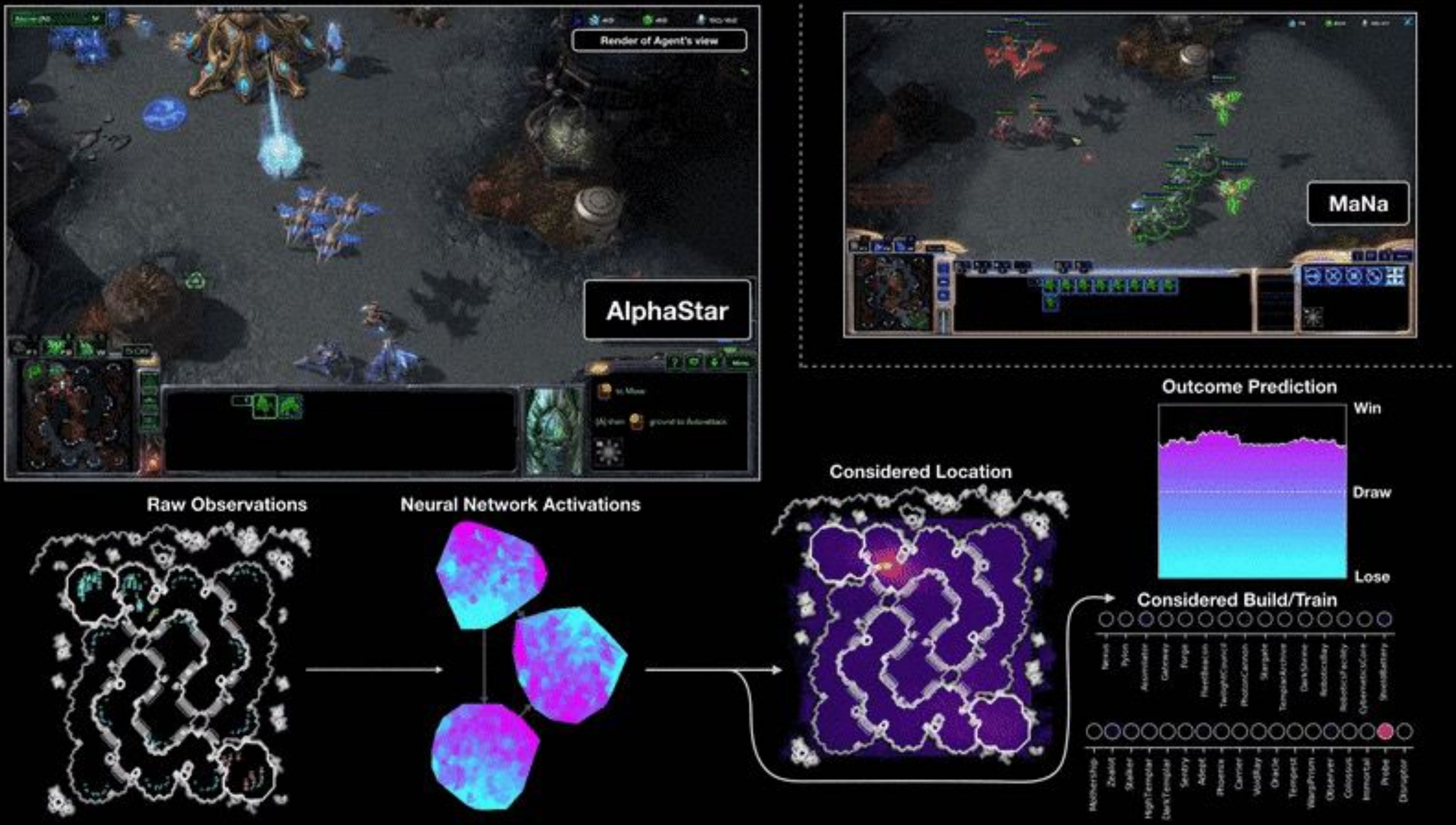
T.N. Kipf and M. Welling, "**Semi-supervised classification with graph convolutional networks**", ICLR 2017
P. Battaglia et al., "**Relational inductive biases, deep learning, and graph networks**", arXiv 2018

Strategic Game Playing

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

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

AlphaStar Plays StarCraft II



O. Vinyals et al., **Grandmaster level in StarCraft II using multi-agent reinforcement learning**, Nature 575:350-354, 2019

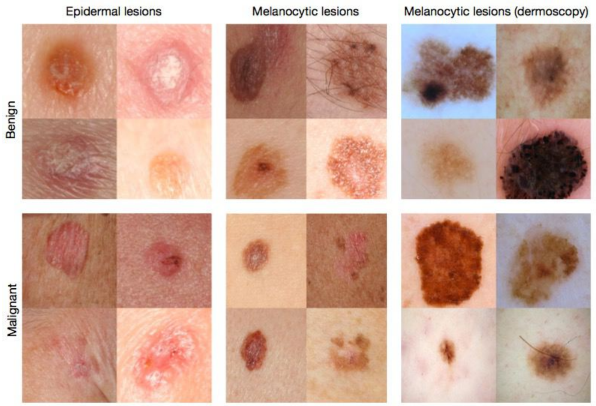
Robotics

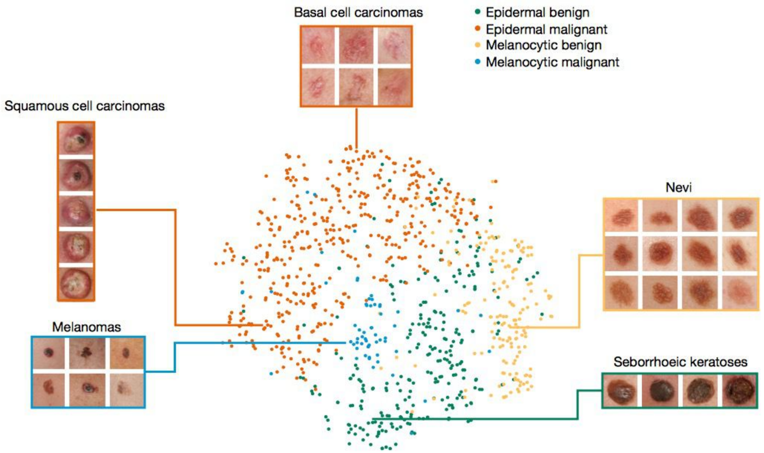
Ilge Akkaya et al. **Solving Rubik's Cube with a Robot Hand**. OpenAI Technical Report 2019

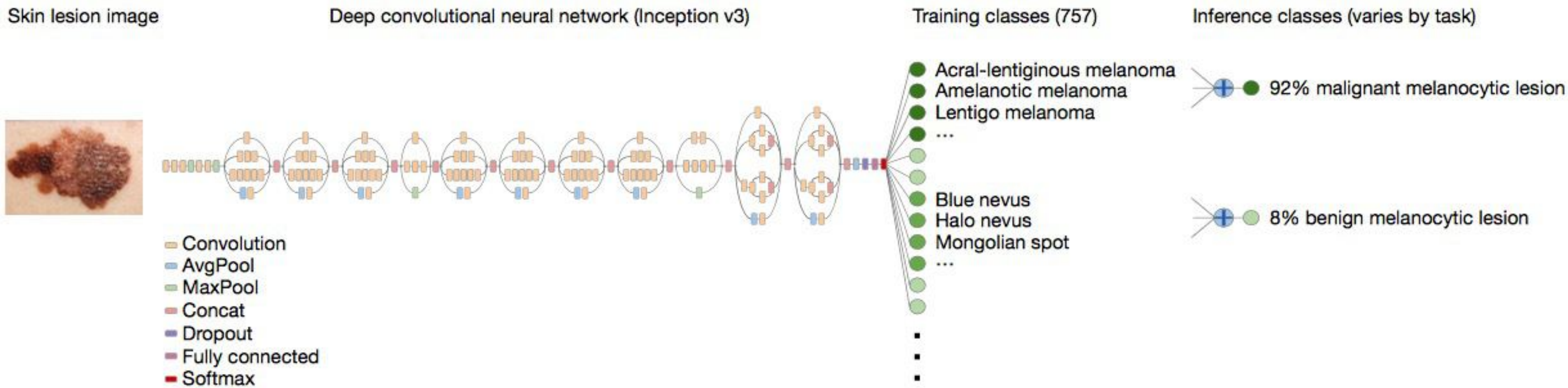
Self-Driving Vehicles

Mariusz Bojarski et al. **End to End Learning for Self-Driving Cars**. NVidia Technical Report 2016

Medical Image Analysis







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

Stanford ML Group

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

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

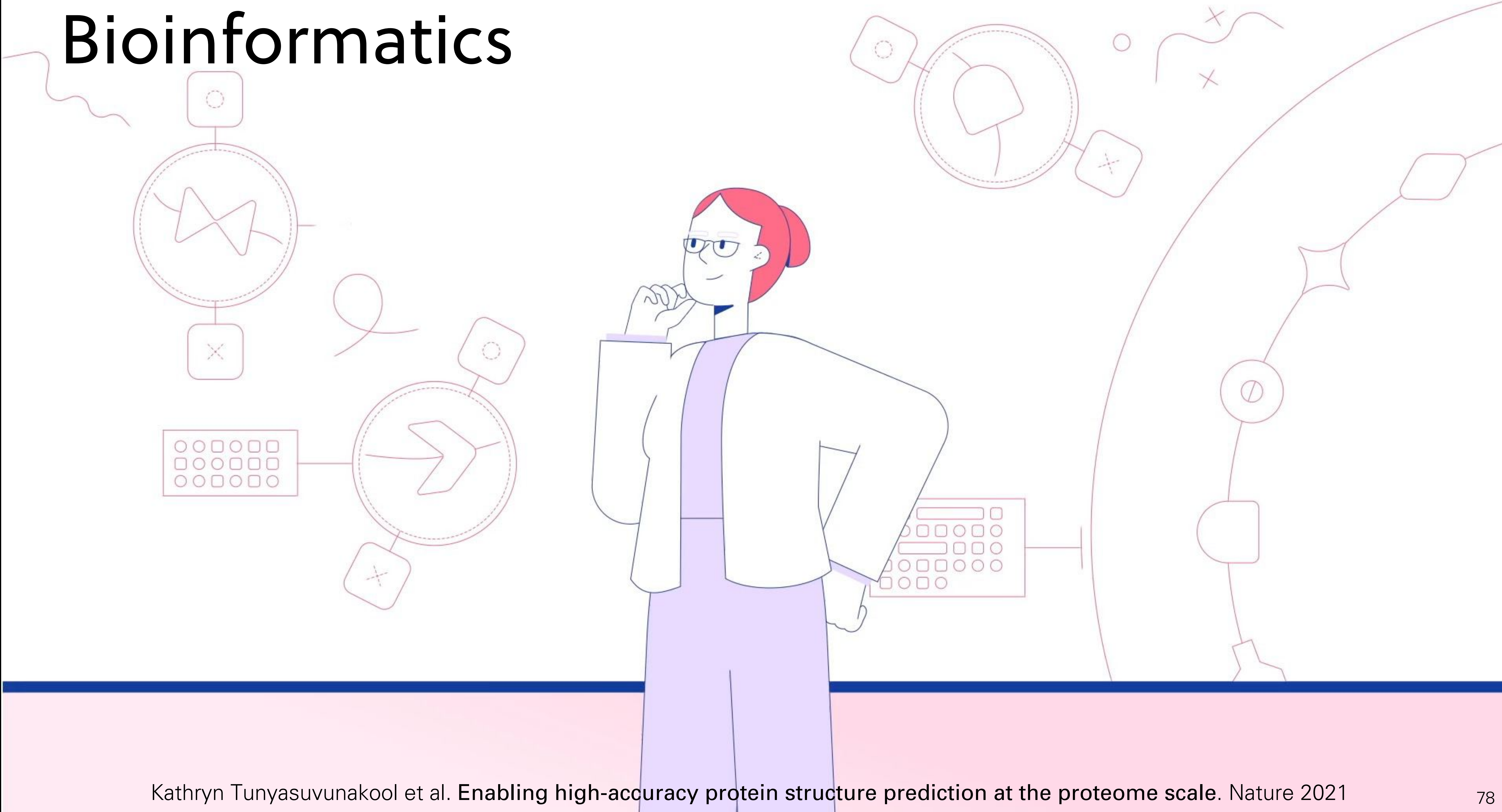
We develop an algorithm that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists.

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

READ OUR PAPER

Medical Image Analysis

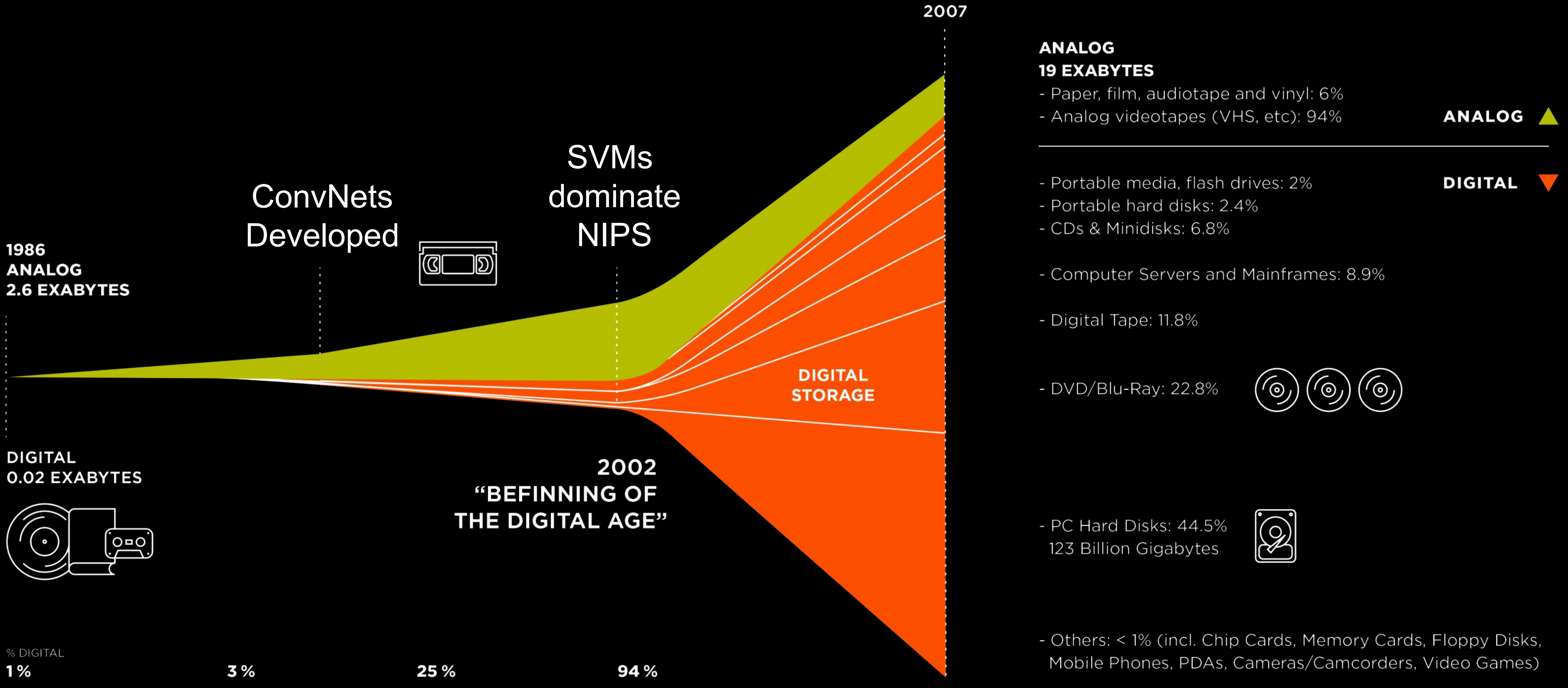
Bioinformatics



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

Why now?
The Resurgence of Deep Learning

GLOBAL INFORMATION STORAGE CAPACITY
IN OPTIMALLY COMPRESSED BYTES



DIGITAL
280 EXABYTES

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

Slide credit: Neil Lawrence

Datasets vs. Algorithms

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

Table credit: Quant Quanto

Powerful Hardware

- Deep neural nets highly amenable to implementation on Graphics Processing Units (GPUs)
  - Matrix multiplication
  - 2D convolution

- E.g. nVidia Pascal GPUs deliver 10 Tflops
  - Faster than fastest computer in the world in 2000
  - 10 million times faster than 1980's Sun workstation

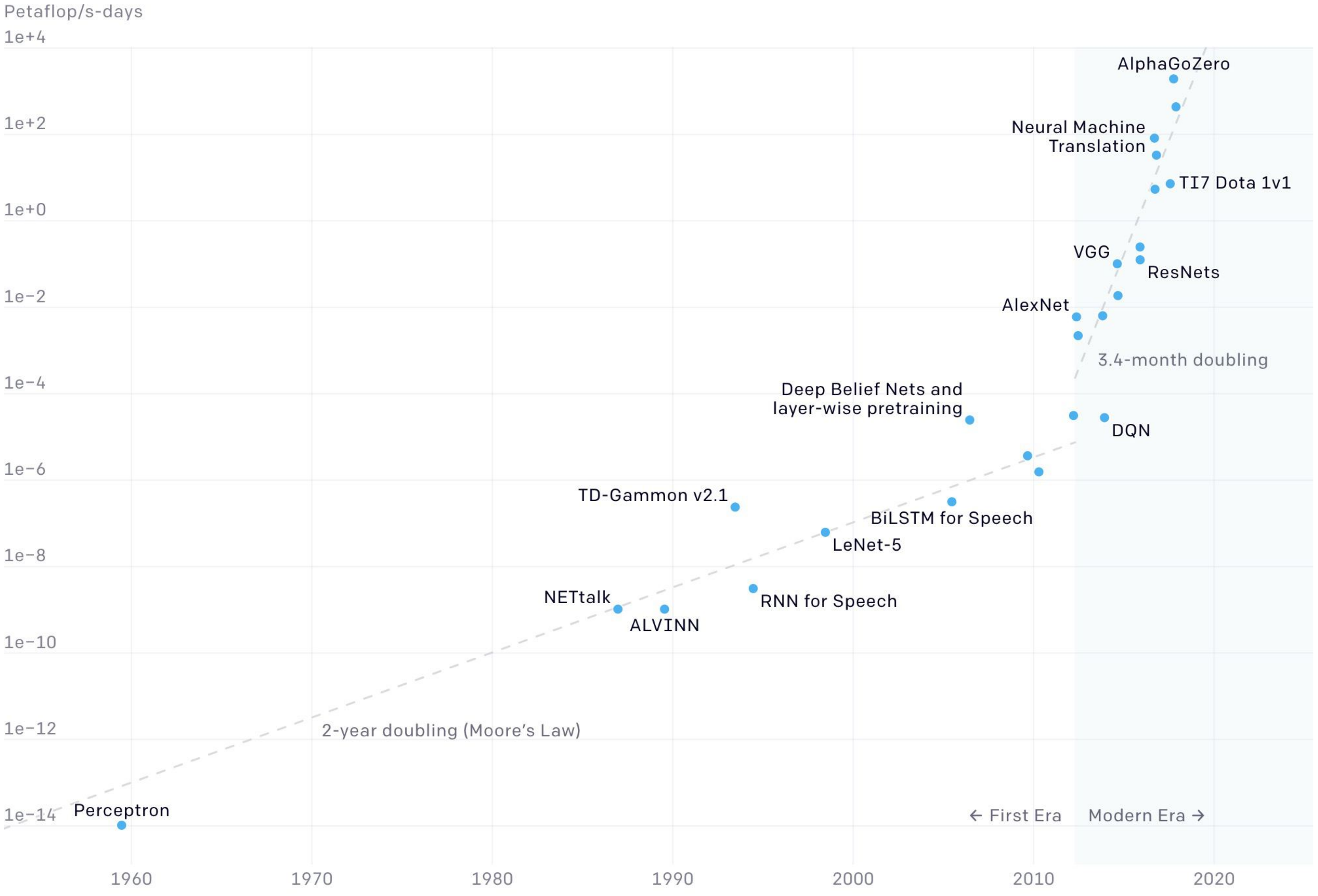


Image: OpenAI

Slide adapted from Rob Fergus

Working ideas on how to train deep architectures

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

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

Abstract

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

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

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

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

Editor: Yoshua Bengio

Abstract

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

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

1. Introduction

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

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

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

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

Working ideas on how to train deep architectures

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

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

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

Abstract

Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as *internal covariate shift*, and address the problem by normalizing layer inputs. Our method draws its strength from making normalization a part of the model architecture and performing the normalization *for each training mini-batch*. Batch 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*

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

Abstract

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

1 Introduction

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

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

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

$$\frac{1}{m}\frac{\partial \ell(\mathrm{x}_i, \Theta)}{\partial \Theta}.$$

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. When the input distribution to a learning system changes, it is said to experience *covariate shift* (Shimodaira, 2000). This is typically handled via domain adaptation (Jiang, 2008). However, the notion of covariate shift can be extended beyond the learning system as a whole, to apply to its parts, such as a sub-network or a layer. Consider a network computing

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

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

$$\ell = F_2(\mathrm{x}, \Theta_2).$$

For example, a gradient descent step

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

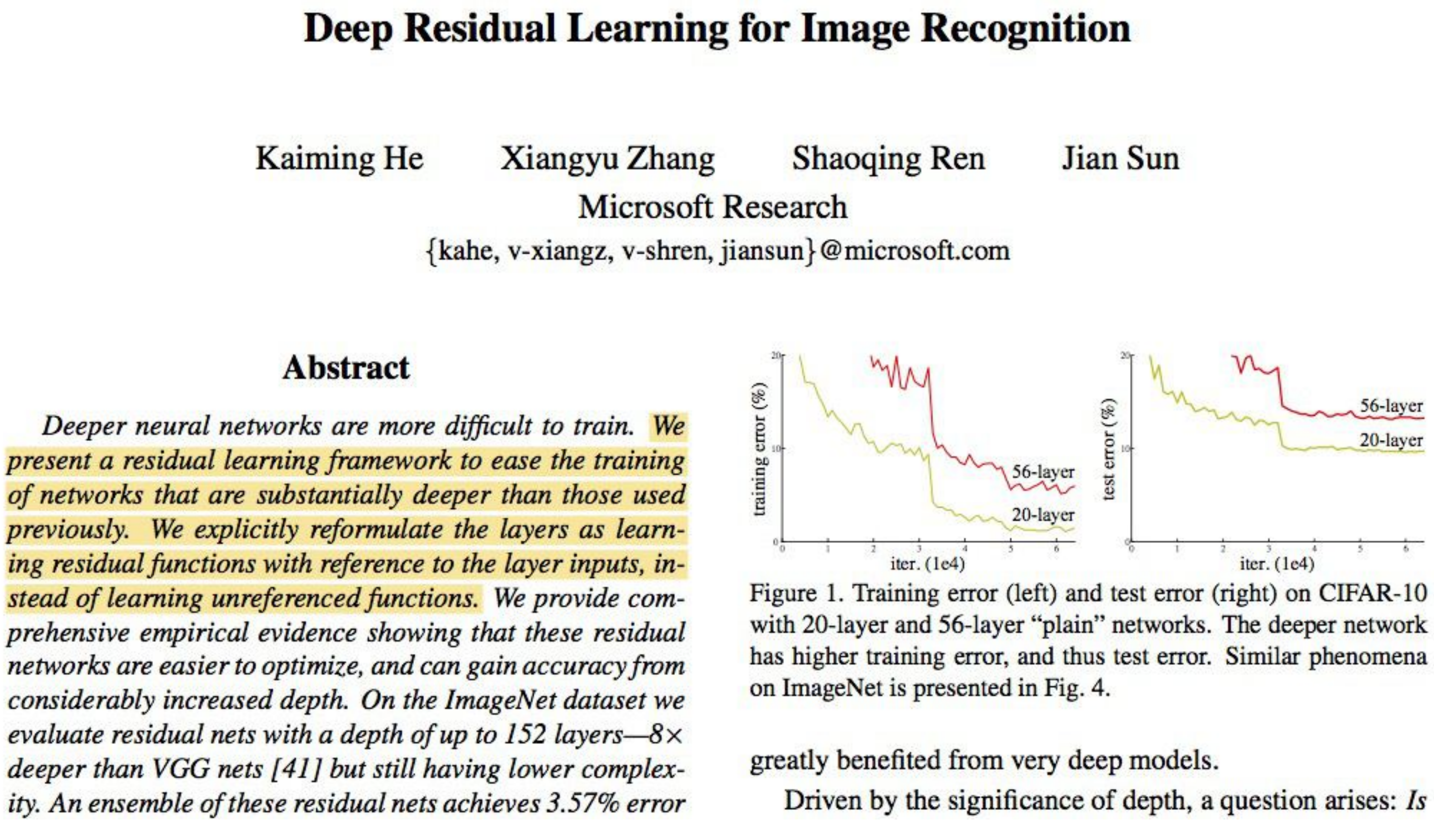
(for batch size m and learning rate α) is exactly equivalent to that for a stand-alone network F_2 with input x. Therefore, the input distribution properties that make training more efficient – such as having the same distribution between the training and test data – apply to training the sub-network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

1

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

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

Working ideas on how to train deep architectures



Deep Residual Learning for Image Recognition

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

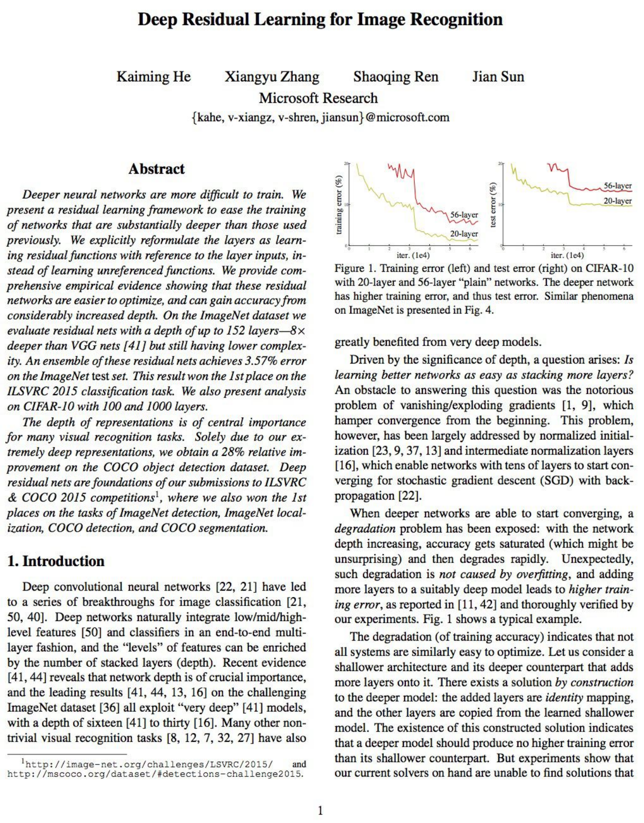
Abstract

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

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

greatly benefited from very deep models.

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



Deep Residual Learning for Image Recognition

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

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error 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[1], 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

[1] http://image-net.org/challenges/LSVRC/2015/ and http://mscoco.org/dataset/#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

1

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

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

Software

TensorFlow

torch

mxnet

theano

Caffe

Caffe2

MatConvNet

The Microsoft Cognitive Toolkit

A free, easy-to-use, open-source, commercial-grade toolkit that trains deep learning algorithms to learn like the human brain.

PYTORCH

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

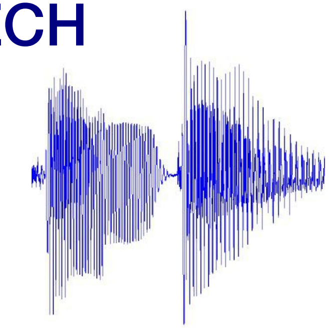
Traditional Machine Learning

VISION

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

fixed | learned

SPEECH



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

fixed | learned

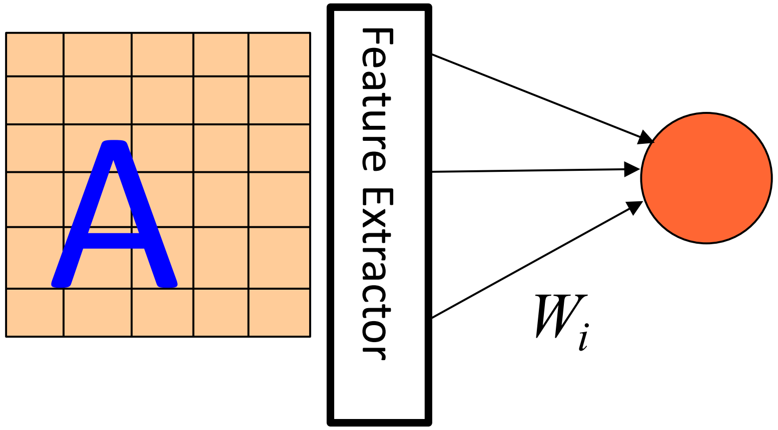
NLP

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

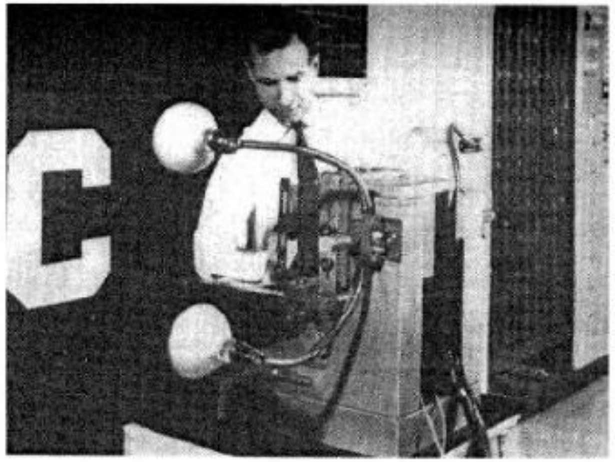
fixed | learned

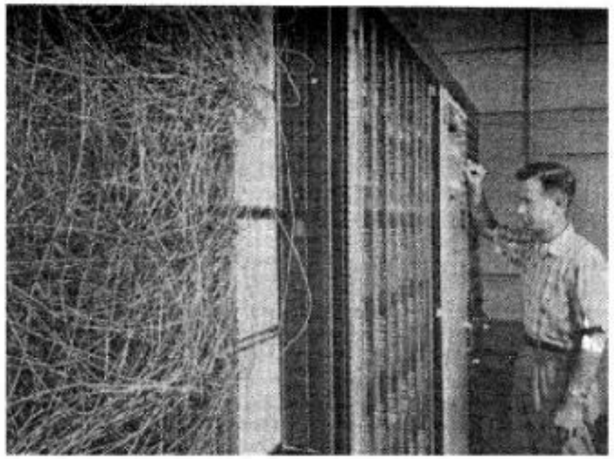
It's an old paradigm

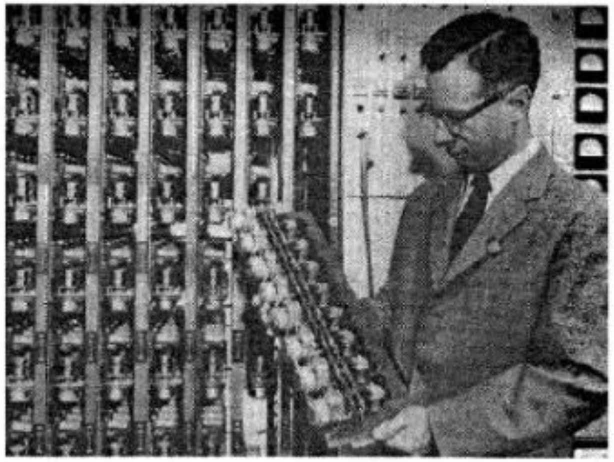
- The first learning machine:
  the Perceptron
  - Built at Cornell in 1960
- The Perceptron was a linear classifier on top of a simple feature extractor
- The vast majority of practical applications of ML today use glorified linear classifiers or glorified template matching.
- Designing a feature extractor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



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







slide by Marc'Aurelio Ranzato, Yann LeCun

Hierarchical Compositionality

VISION

pixels → edge → texton → motif → part → object

SPEECH

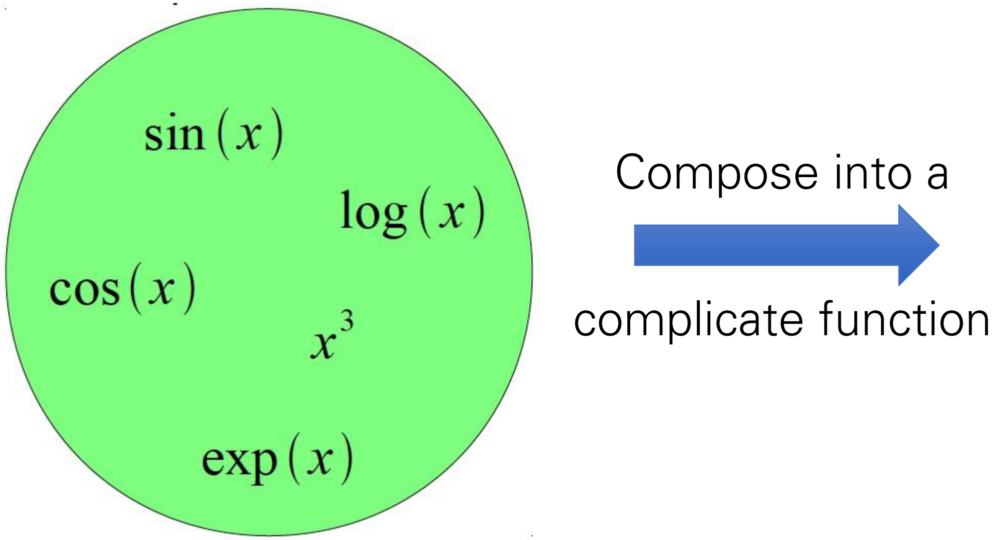
sample → spectral band → formant → motif → phone → word

NLP

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

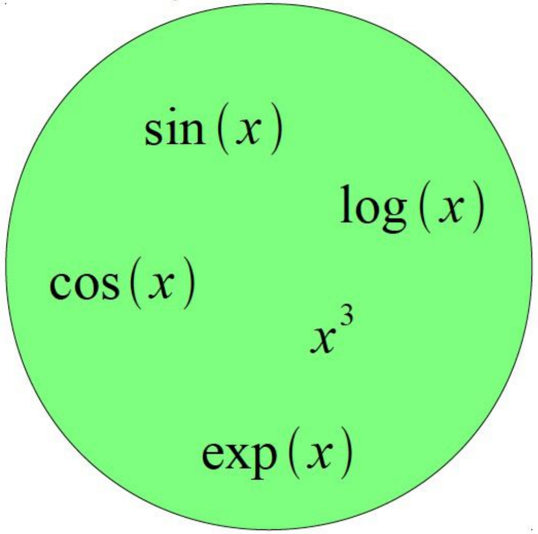
Building A Complicated Function

Given a library of simple functions



Building A Complicated Function

Given a library of simple functions

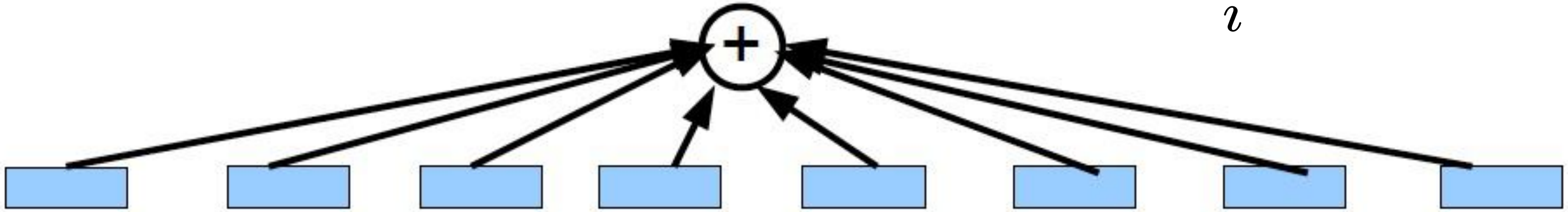


Compose into a complicate function

Idea 1: Linear Combinations

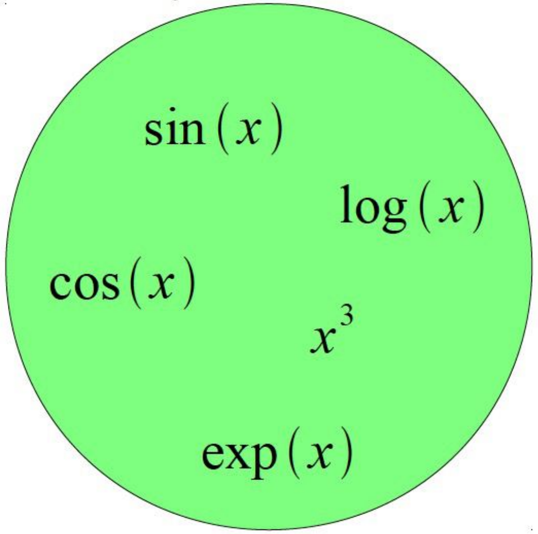
- Boosting
- Kernels
- …

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



Building A Complicated Function

Given a library of simple functions



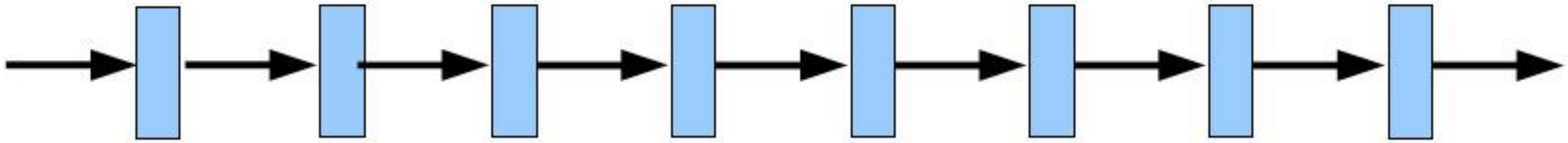
Compose into a

complicate function

Idea 2: Compositions

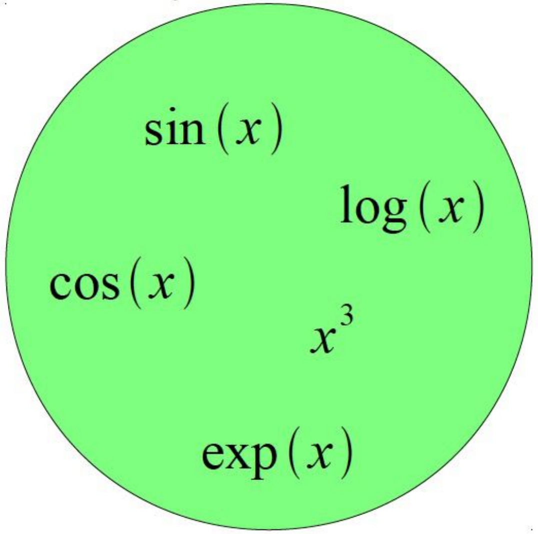
- Deep Learning
- Grammar models
- Scattering transforms…

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



Building A Complicated Function

Given a library of simple functions

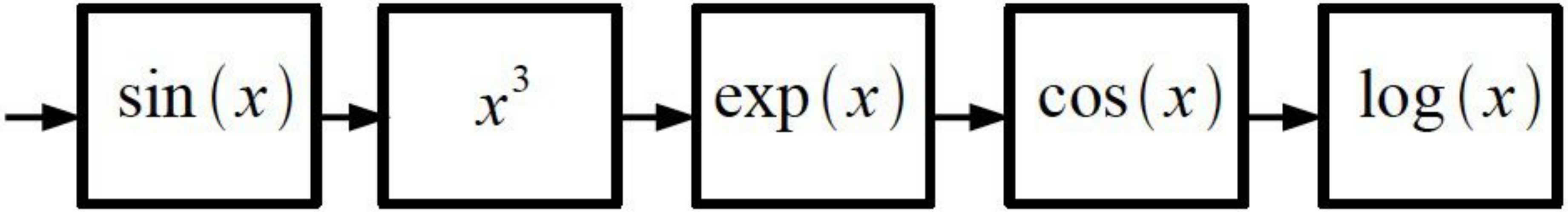


Compose into a complicate function

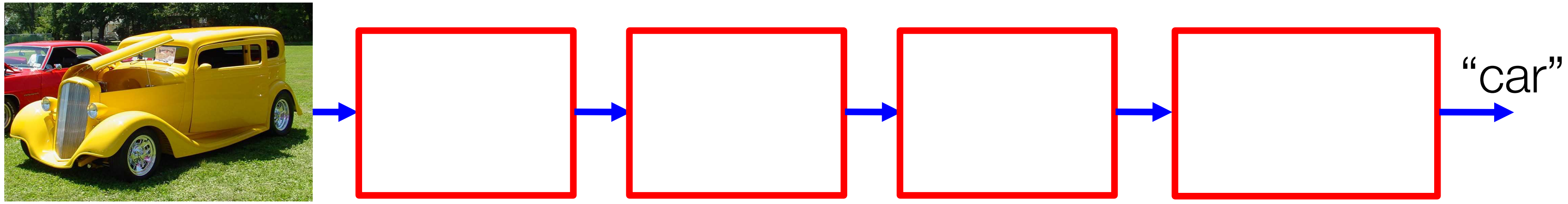
Idea 2: Compositions

- Deep Learning
- Grammar models
- Scattering transforms…

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



Deep Learning = Hierarchical Compositionality



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

Deep Learning = Hierarchical Compositionality

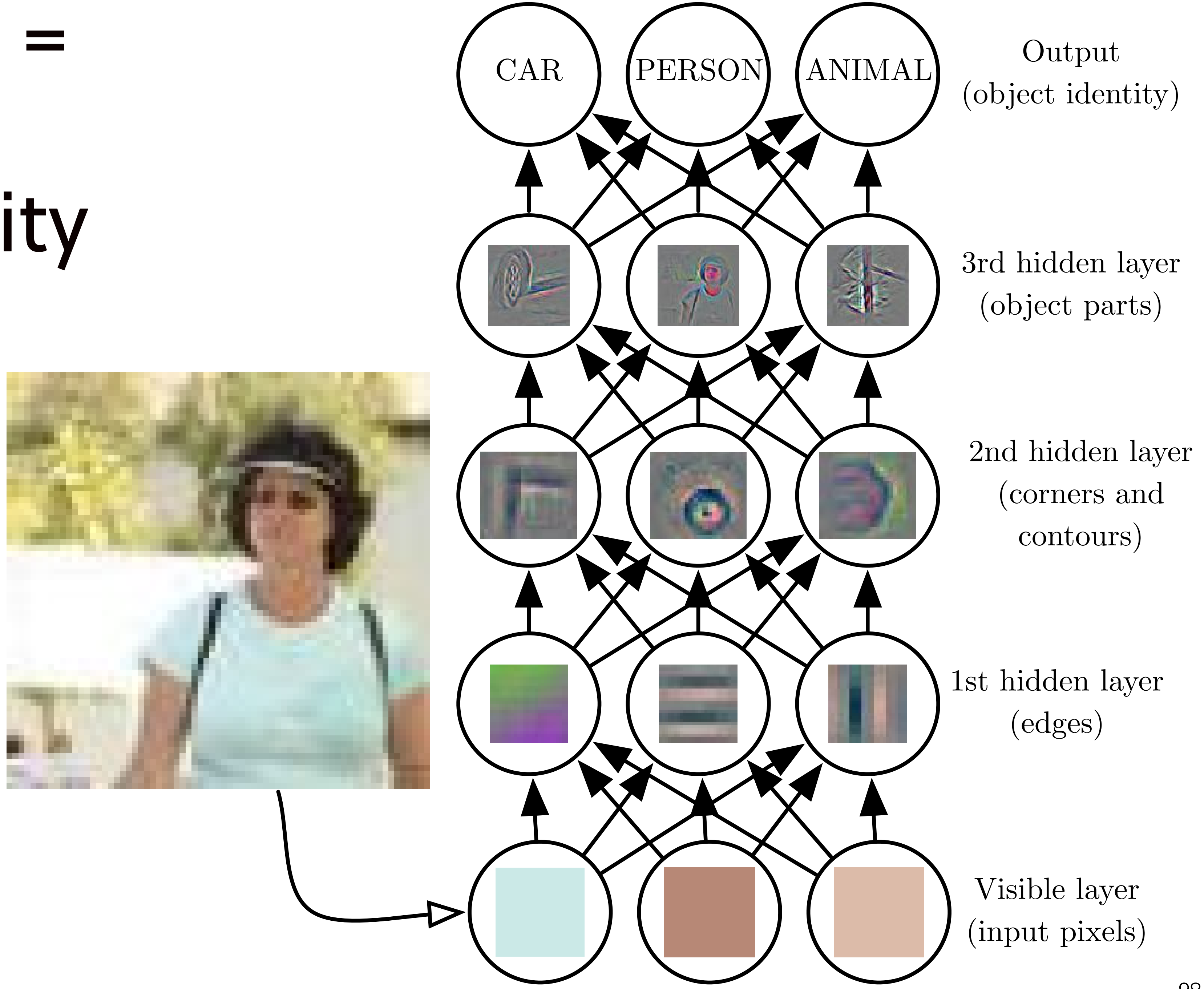
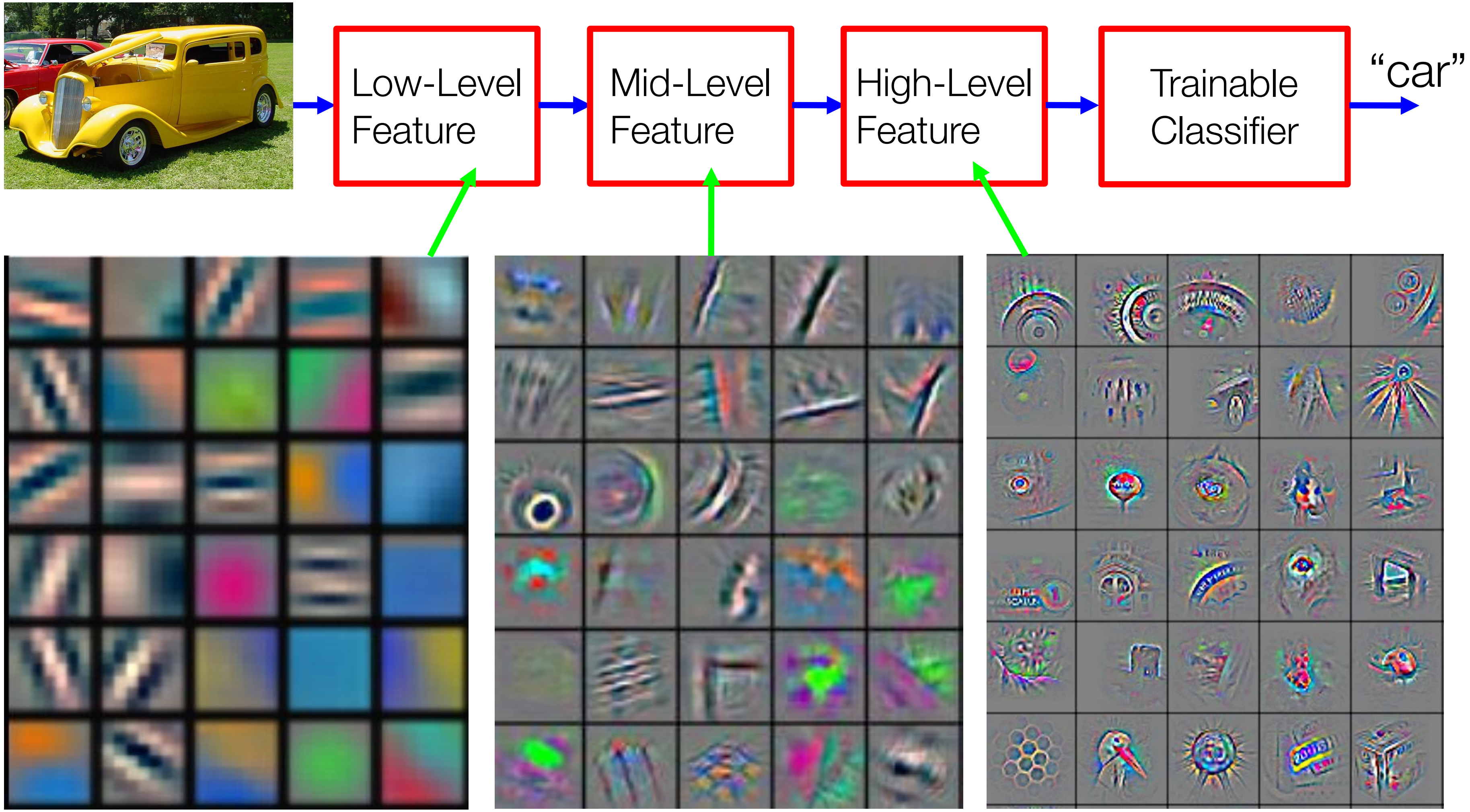


Image credit: Ian Goodfellow

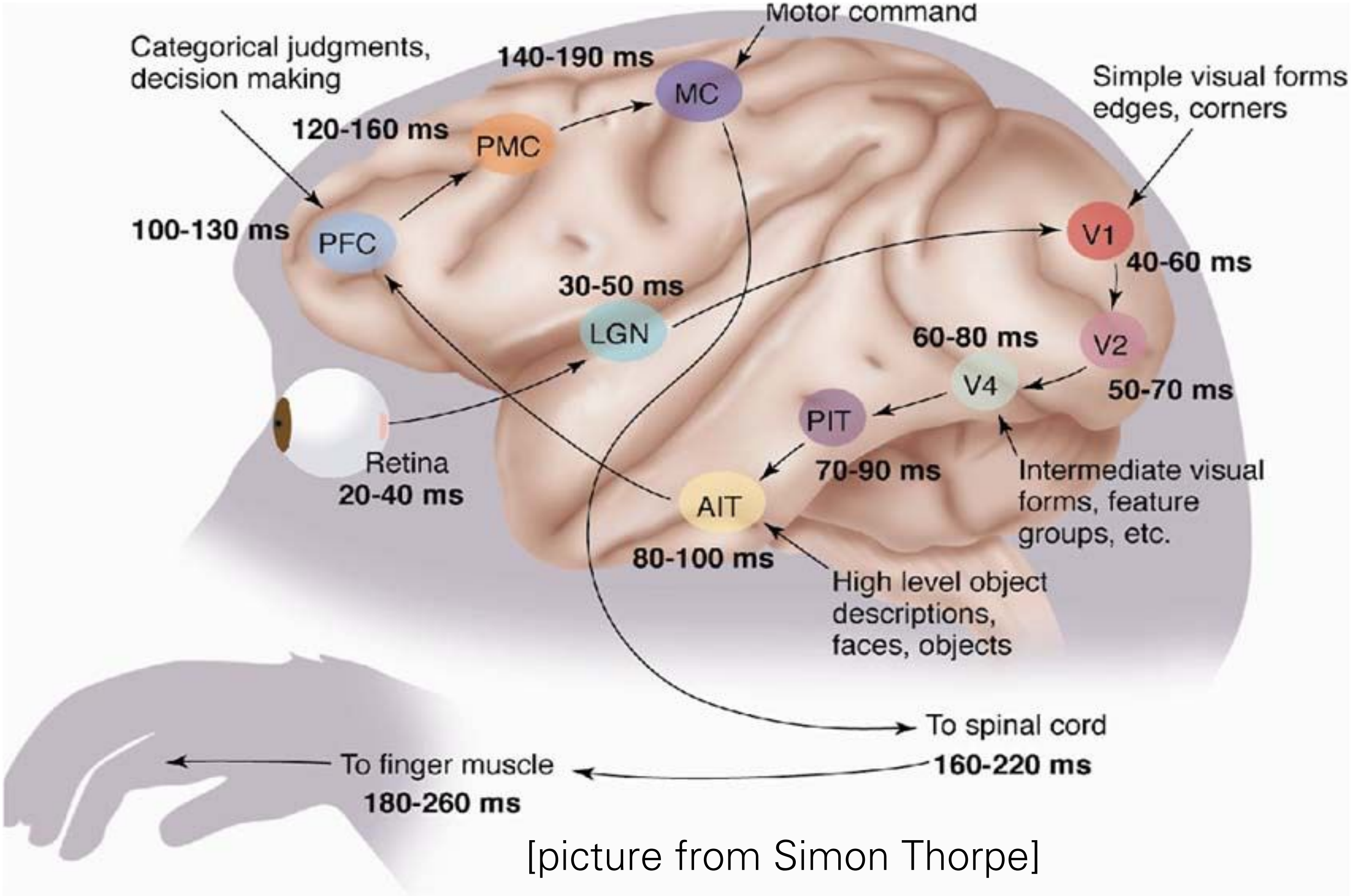
Deep Learning = Hierarchical Compositionality



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

The Mammalian Visual Cortex is Hierarchical

- The ventral (recognition) pathway in the visual cortex



[picture from Simon Thorpe]

slide by Marc'Aurelio Ranzato, Yann LeCun

Three key ideas

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

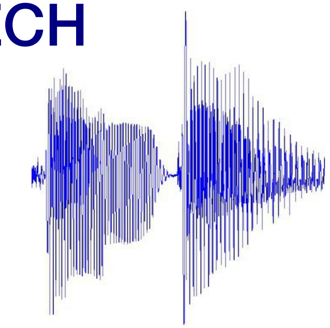
Traditional Machine Learning

VISION

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

fixed | learned

SPEECH



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

fixed | learned

NLP

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

fixed | learned

More accurate version

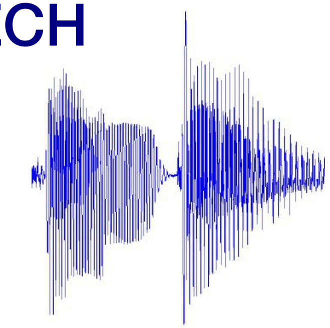
"Learned"

VISION

SIFT/HOG → K-Means/ pooling → classifier → "car"

fixed | unsupervised | supervised

SPEECH



MFCC → Mixture of Gaussians → classifier → \'d ē p\

fixed | unsupervised | supervised

NLP

This burrito place is yummy and fun! → Parse Tree Syntactic → n-grams → classifier → "+"

fixed | unsupervised | supervised

Deep Learning = End-to-End Learning

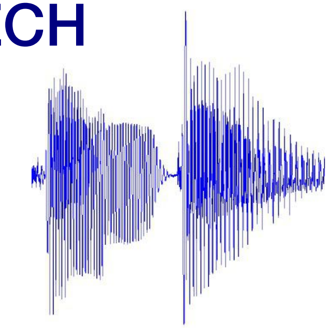
→ "Learned"

VISION

→ SIFT/HOG → K-Means/ pooling → classifier → "car"

fixed | unsupervised | supervised

SPEECH



→ MFCC → Mixture of Gaussians → classifier → \'d ē p\

fixed | unsupervised | supervised

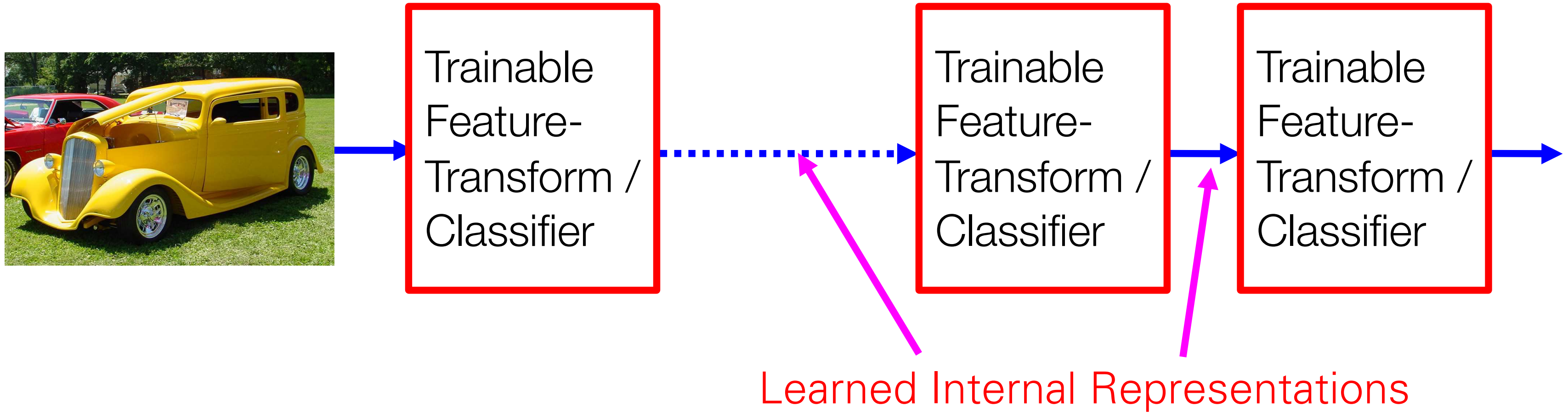
NLP

This burrito place is yummy and fun! → Parse Tree Syntactic → n-grams → classifier → "+"

fixed | unsupervised | supervised

Deep Learning = End-to-End Learning

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



"Shallow" vs Deep Learning

- "Shallow" models

hand-crafted Feature Extractor → "Simple" Trainable Classifier →

fixed | learned

- Deep models

Trainable Feature-Transform / Classifier → … → Trainable Feature-Transform / Classifier → Trainable Feature-Transform / Classifier →

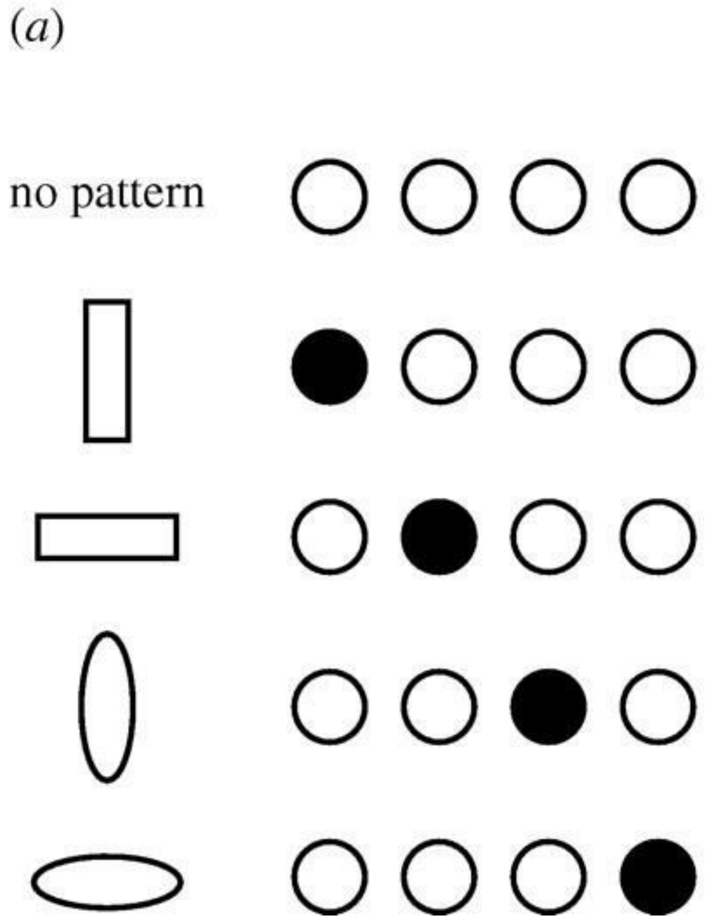
Learned Internal Representations

Three key ideas

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

Localist representations

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

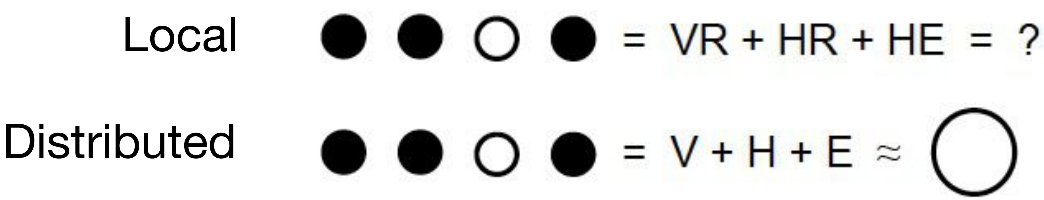


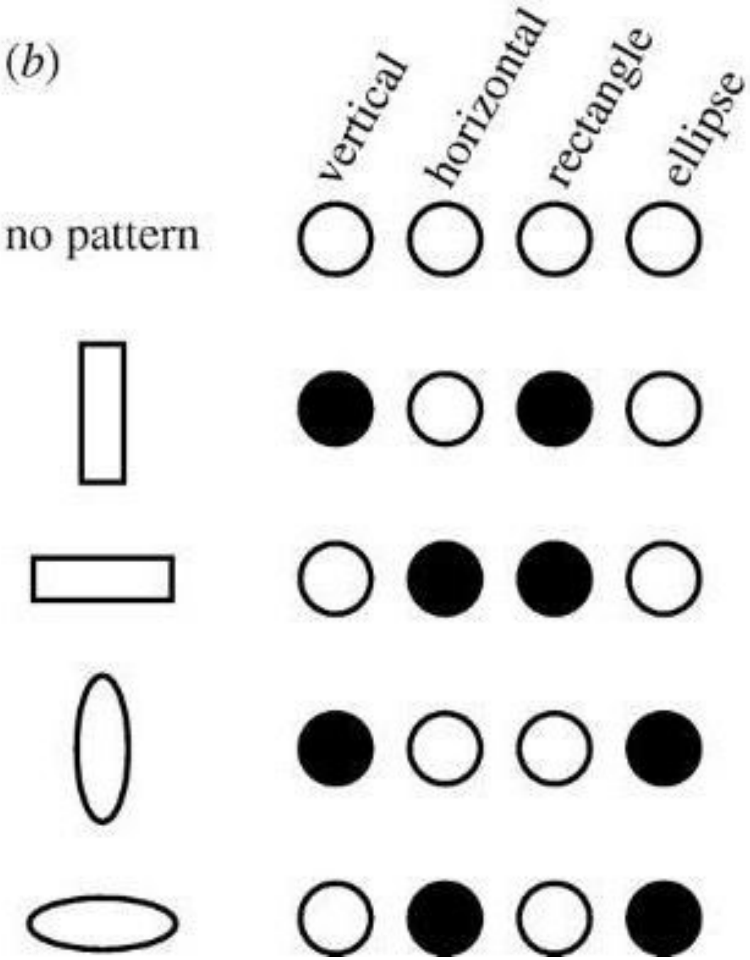
Slide credit: Geoff Hinton

Image credit: Moontae Lee

Distributed Representations

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





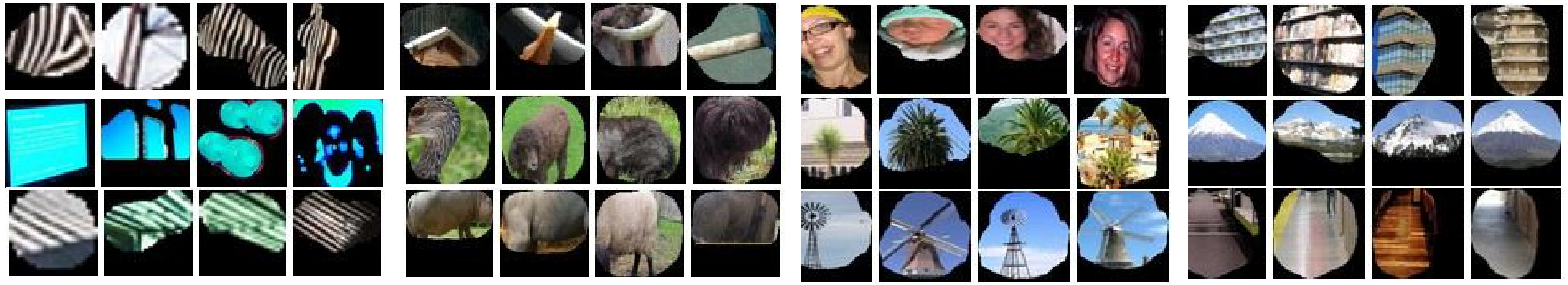
Power of distributed representations!

Scene Classification

bedroom

mountain

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



Simple elements & colors

Object part

Object

Scene

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

Slide credit: Bolei Zhou

Three key ideas of deep learning

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

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

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

Benefits of Deep/Representation Learning

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

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

Problems with Deep Learning

- **Problem#1: Non-Convex! Non-Convex! Non-Convex!**
  - Depth>=3: most losses non-convex in parameters
  - Theoretically, all bets are off
  - Leads to stochasticity
    - different initializations → different local minima

- Standard response #1
  - "Yes, but all interesting learning problems are non-convex"
  - For example, human learning
    - Order matters → wave hands → non-convexity

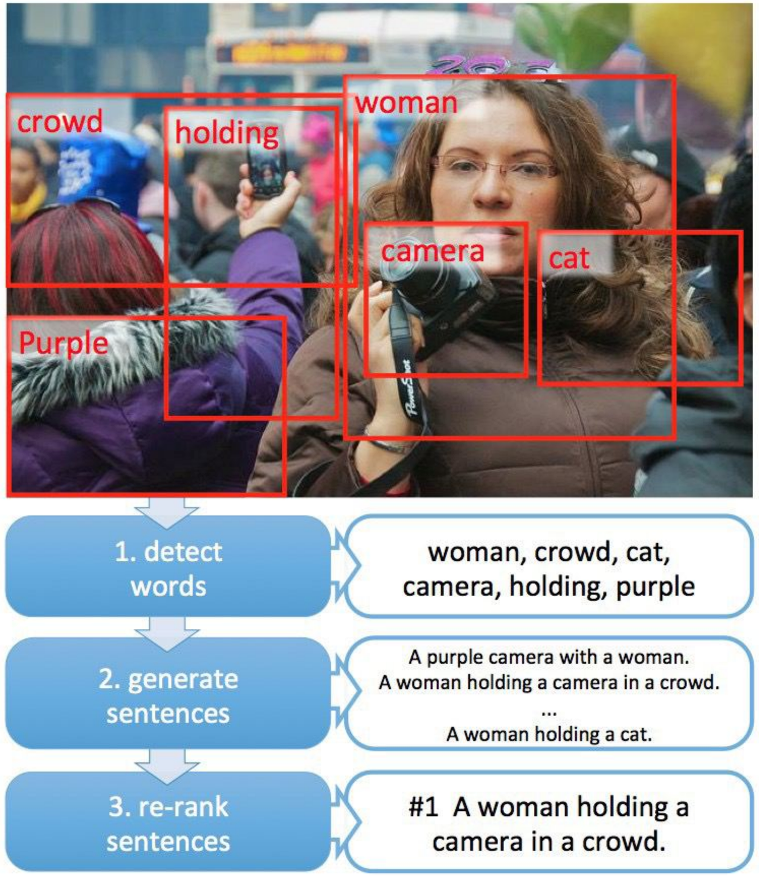
- Standard response #2
  - "Yes, but it often works!"

Problems with Deep Learning

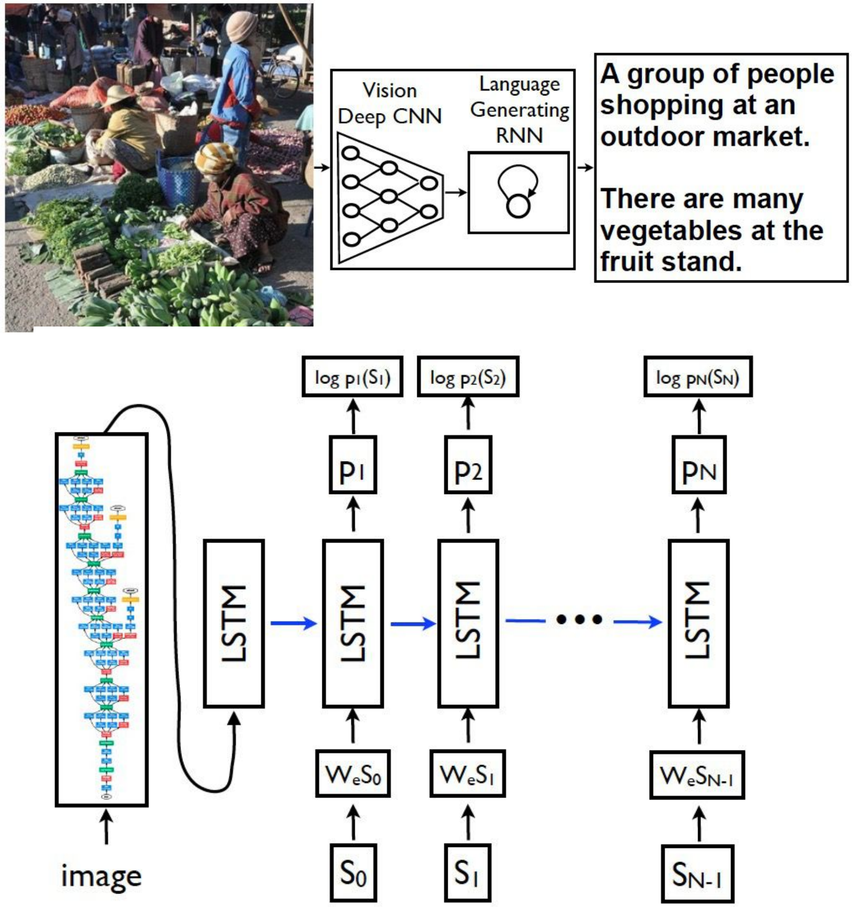
- **Problem#2: Hard to track down what's failing**
  - Pipeline systems have "oracle" performances at each step
  - In end-to-end systems, it's hard to know why things are not working

Problems with Deep Learning

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



[Fang et al. CVPR15]



[Vinyals et al. CVPR15]

Problems with Deep Learning

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

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

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

Problems with Deep Learning

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

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

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

1958 New York Times…

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

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

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

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

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

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

Without Human Controls

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

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

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

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

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

Learns by Doing

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

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

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

The New York Times
Science

COMPUTER SCIENTISTS STYMIED IN THEIR QUEST TO MATCH HUMAN VISION

By WILLIAM J. BROAD
Published: September 25, 1984

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

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

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

SCIENCE

Researchers Announce Advance in Image-Recognition Software

By **JOHN MARKOFF** NOV. 17, 2014

Email

Share

Tweet

Save

More

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

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

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

Captioned by Human and by Google's Experimental Program

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

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

INTERESTING.JPG @INTERESTING_JPG · 10h

a man holding a mirror up to his face .

1 View more photos and videos

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

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

INTERESTING.JPG @INTERESTING_JPG · 18h

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

2

View more photos and videos

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

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

INTERESTING.JPG @INTERESTING_JPG · Feb 20

a surfboard attached to the top of a car .

8 8 View more photos and videos

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

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

INTERESTING.JPG @INTERESTING_JPG · Feb 19

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

2

View more photos and videos

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

TWEETS	FOLLOWING	FOLLOWERS	FAVORITES
587	18	746	13

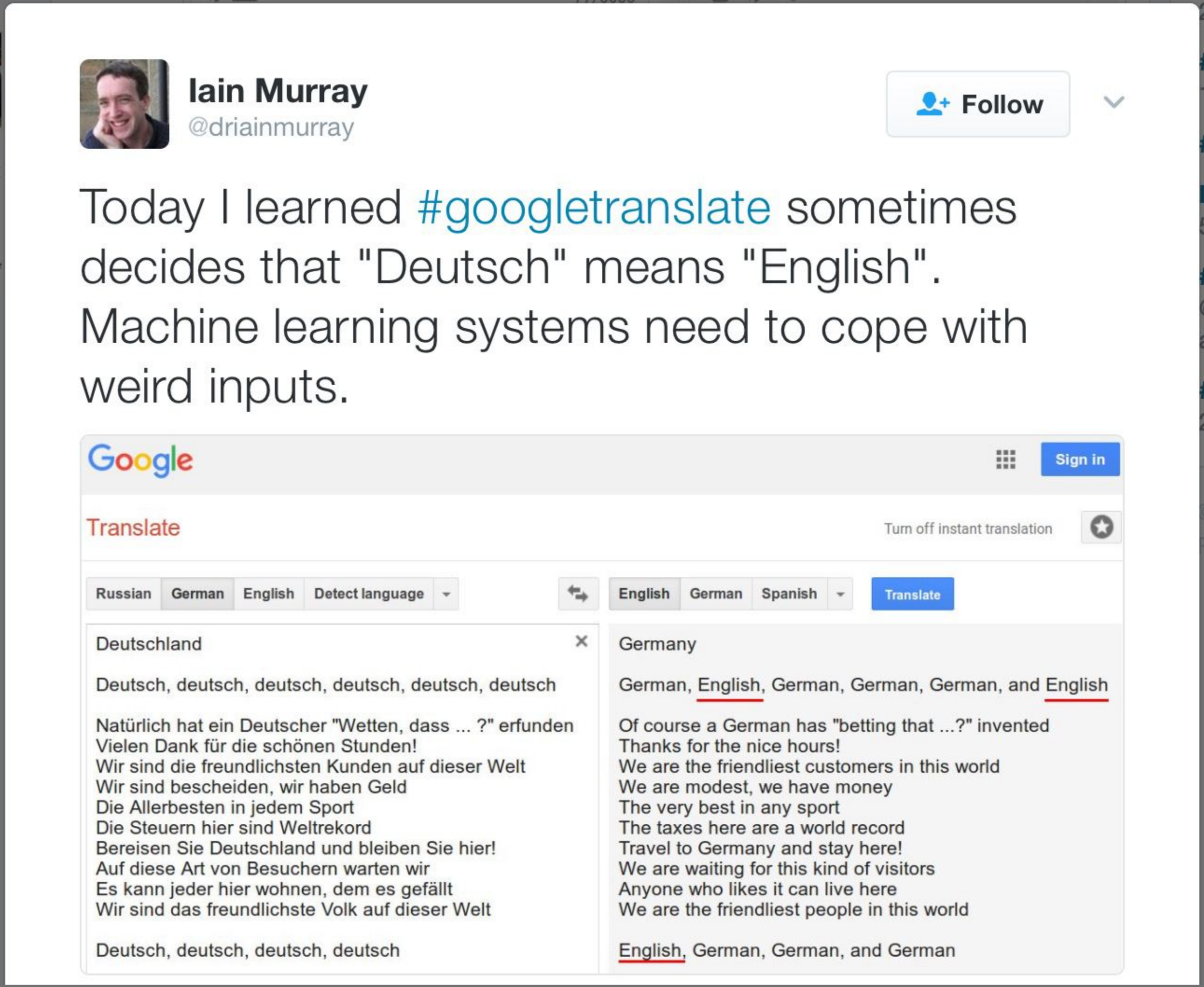
INTERESTING.JPG @INTERESTING_JPG · 16h

this appears to be a small bedroom in the snow .

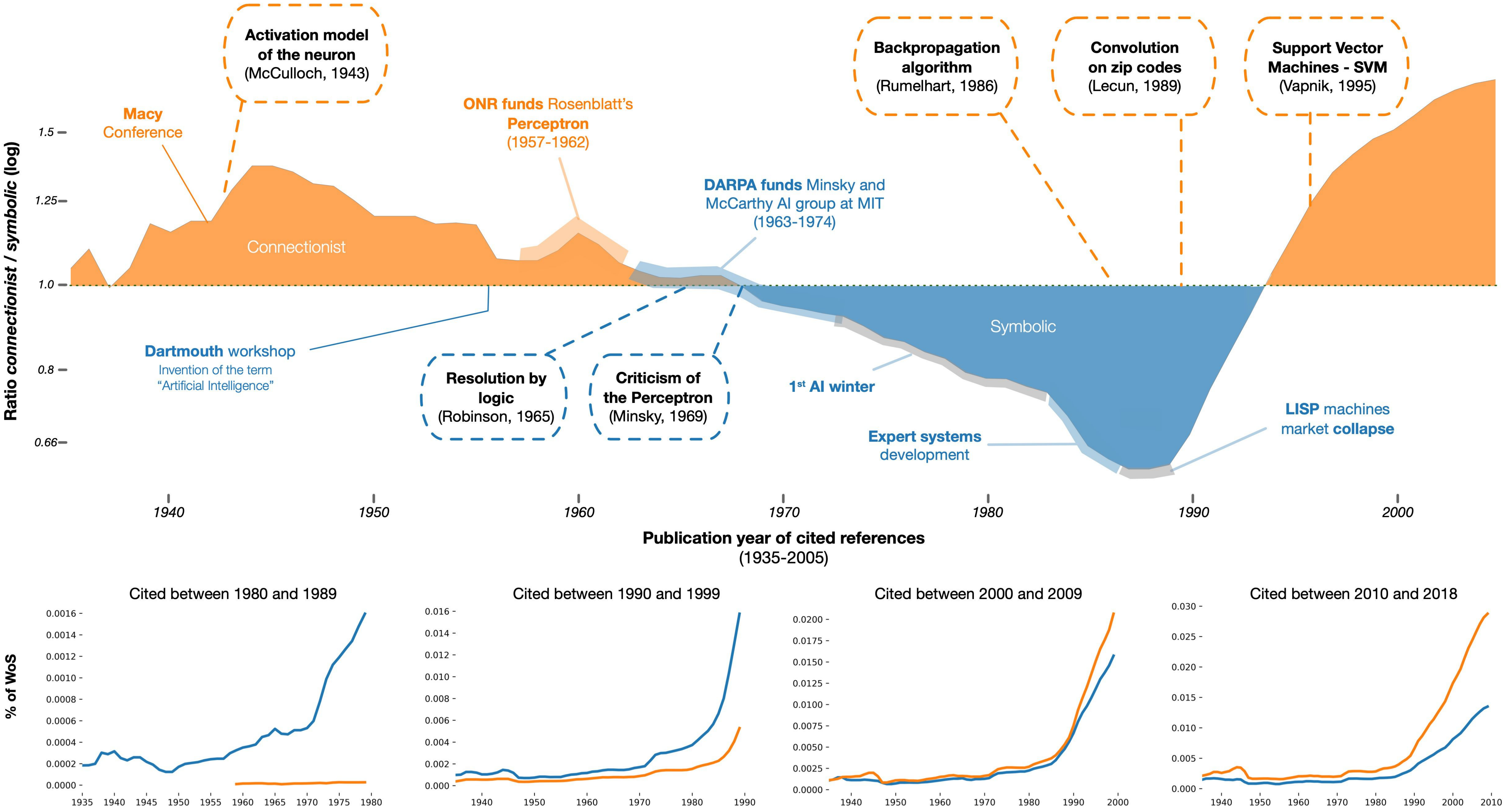
6

View more photos and videos

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


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


Iain Murray
@driainmurray
Follow
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.)
English German Spanish Detect language
German English Spanish
Translate
knife, fork, knife,
Messer, Messer, Messer,
(The trailing comma messes this one up.)
19/5000
English German Spanish Detect language
German English Spanish
Translate
Messer, Gabel, Messer, Messer, Messer, Messer, Messer, Messer, Messer, Messer
Screen monitor styling Projector styling Print styling ← back to. 2010-01-20 with adjustable interlinear. Knife, fork; knife, knife, knife, knife;
77/5000
RETWEETS
120
LIKES
184
Iain Murray
@driainmurray
Academic in Machine Learning and Statistics.
homepages.inf.ed.ac.uk/imurray2/
Joined May 2011
© 2017 Twitter About Help Center Terms
Privacy Cookies Ads info



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

AI DEBATE : YOSHUA BENGIO | GARY MARCUS

Gary Marcus

—

Yoshua Bengio

https://www.youtube.com/watch?v=EeqwFjqFvJA

Next Lecture:
Machine Learning Overview