CPEN 400D: Deep Learning

Lecture 1 (I): Introduction to Deep Learning

Renjie Liao

University of British Columbia

Winter, Term 2, 2022

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics

Outline

- **Course Information**
- Introduction to Deep Learning
    - History
    - Modern Applications
    - Taxonomy & Connections to ML/AI/Statistics

Course Information

- Course website: https://lrjconan.github.io/UBC-CPEN400D-DL/
- Fundamental topics in deep learning
- Assumes basic knowledge about calculus, linear algebra, probability
  - You can look at the 1st homework to get some feeling
- Assumes proficiency in deep learning libraries: PyTorch (highly preferred), JAX, Tensorflow,…
  - Self-learning through online tutorials, e.g. https://pytorch.org/tutorials/
- Assumes familiarity with LaTeX (e.g., for writing homework and project report).

Course Information

- Time: Tue. & Thu. 12:30 to 2:00pm
- Location: 310 Hugh Dempster Pavilion
- Office hour: 2:30 to 3:30pm, Wed, Fred Kaiser 3047 (Ohm)
- TAs: Qi Yan (qi.yan@ece.ubc.ca),
  Sadegh Mahdavi (smahdavi4@gmail.com)
  Jiahe Liu (jiaheliu@ece.ubc.ca)
- All lectures will be delivered in person without recording unless some challenging situation happens (e.g., I caught COVID)

Course Information

- Use Piazza for discussion & questions (actively answer others' questions get you bonuses)

  https://piazza.com/ubc.ca/winterterm22022/cpen400d

- Use Canvas for submitting homework, assignments, etc.

Course Information

- Expectation & Grading:
    - [30%] 2x Homework
    - [30%] 2x Programming Assignments
    - [40%] Course Project
    - [3% Extra Credits] Participation

    One assignment every three weeks. All work must be done individually.

    Check the course website for more information, e.g., due dates and policy on individual items

Course Information

- How to get free GPUs for your course project?

  1. **Google Colab**: https://research.google.com/colaboratory/

     Google Colab is a web-based iPython Notebook service that has access to a free Nvidia K80 GPU per Google account.

  2. **Google Compute Engine**: https://cloud.google.com/compute

     Google Compute Engine provides virtual machines with GPUs running in Google's data center. You get $300 free credit when you sign up.

- Strategy of using GPUs

  1. Debug models on small datasets (subsets) using CPUs or low-end GPUs until they work

  2. Launch batch jobs on high-end GPUs to tune hyperparameters

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics

Outline

- Course Information
- **Introduction to Deep Learning**
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics

What is Deep Learning?

- Definition from Wikipedia:

  Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning.

- Key Aspects:

  Data: Large (supervised) datasets, e.g., ImageNet (14 million+ annotated images)

  Model: Deep (i.e., many layers) neural networks, e.g., ResNet-152

  Learning algorithm: Back-propagation (BP), i.e., stochastic gradient descent (SGD)

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics

Outline

- Course Information
- Introduction to Deep Learning
  - **History**
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)

History of Deep Learning (Connectionism)

- Artificial Neurons ([McCulloch and Pitts 1943]())

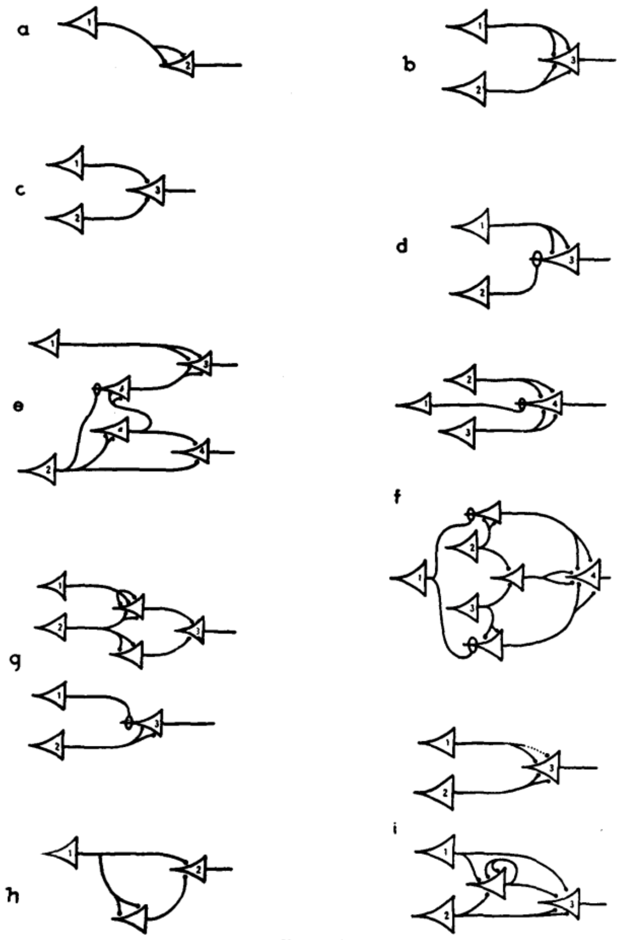


FIGURE 1

Threshold Logic Unit (TLU):

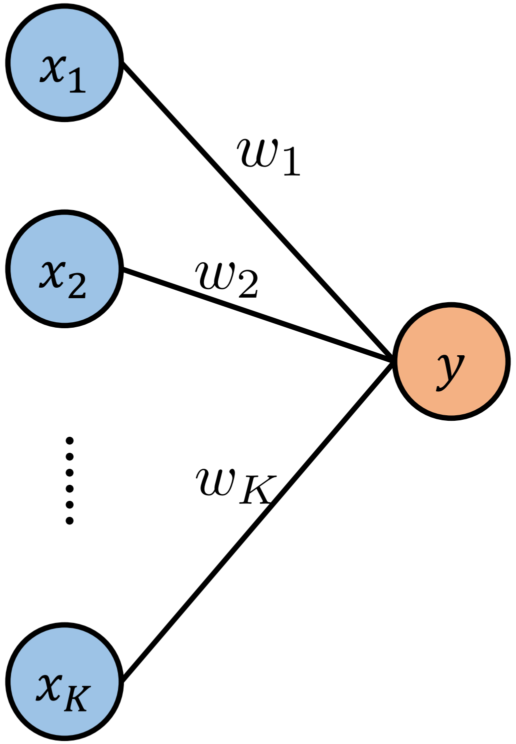
- Binary input and output
- Heaviside step function

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)

History of Deep Learning (Connectionism)

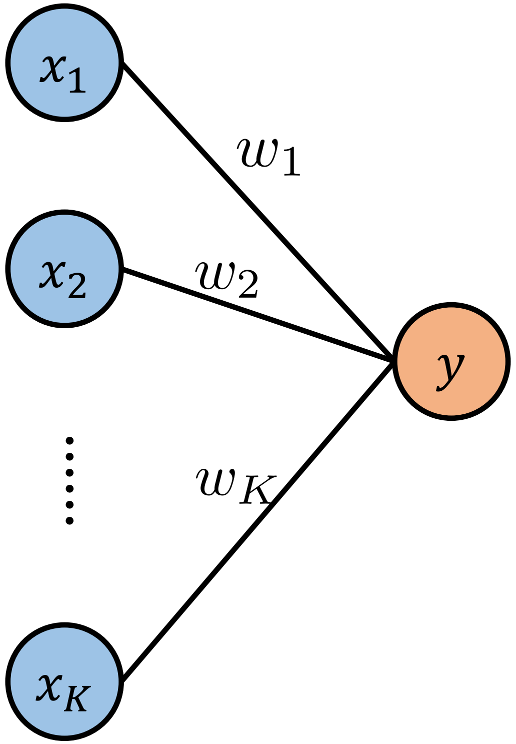
- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)



Linear Unit: $$y = \sum_{k=1}^{K} w_k x_k$$

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)

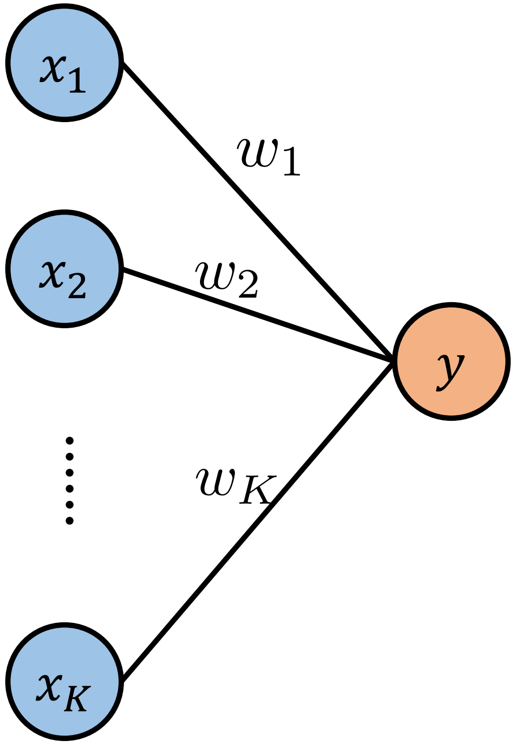


Linear Unit: $$y = \sum_{k=1}^{K} w_k x_k$$

Learning Rule: $$w_k = w_k + \boxed{\eta \mathbb{E}\left[y x_k\right]}$$

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)



Linear Unit: $y = \sum_{k=1}^{K} w_k x_k$

Learning Rule: $w_k = w_k + \boxed{\eta \mathbb{E}\left[y x_k\right]}$

Cells that fire together wire together!

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)

In 1951, Marvin Minsky and Dean Edmunds build SNARC (Stochastic Neural Analog Reinforcement Calculator), the first artificial neural network, using 3000 vacuum tubes to simulate a network of 40 neurons.

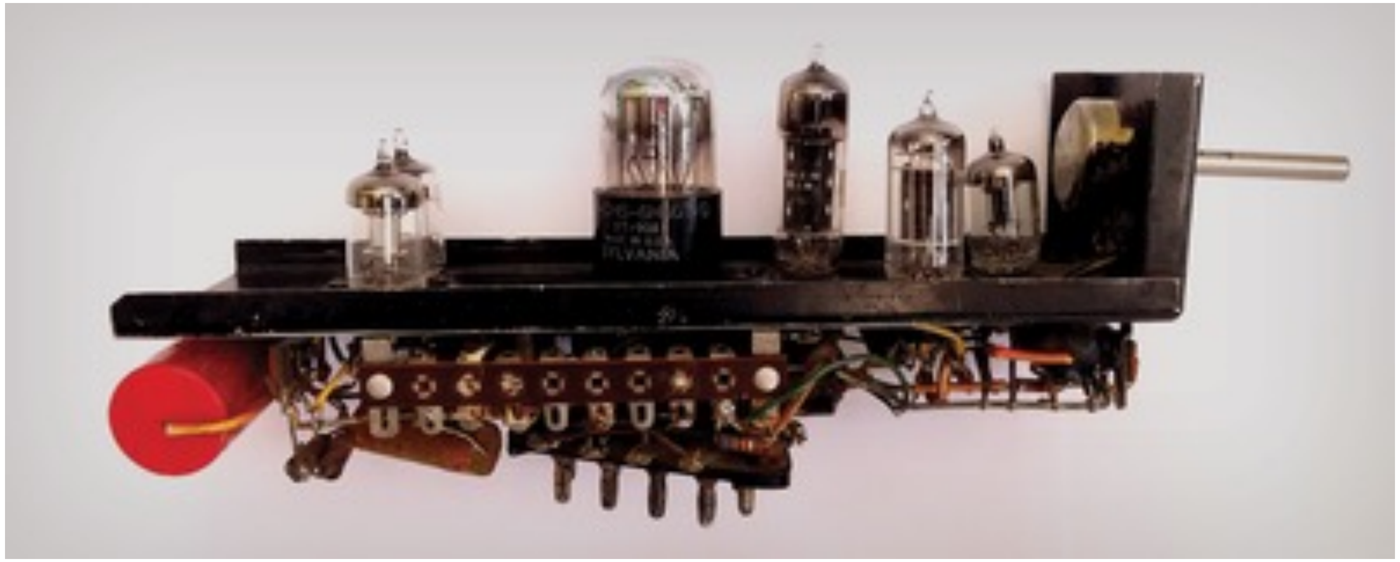


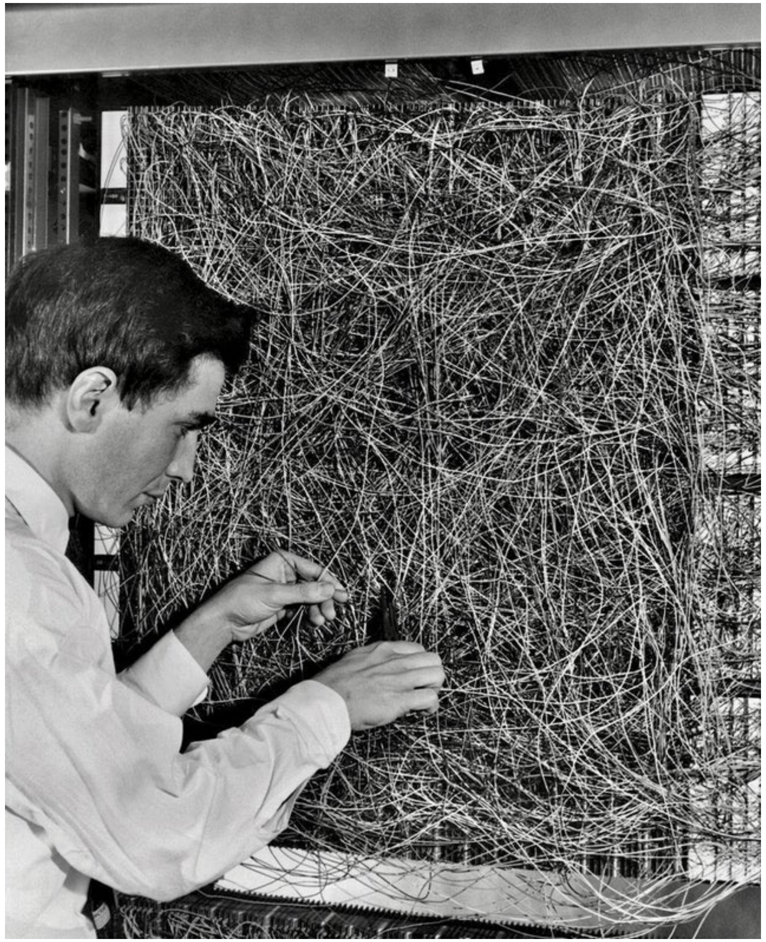
Image Credit: https://www.the-scientist.com/foundations/machine--learning--1951-65792

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)



Frank Rosenblatt working on the Mark I Perceptron (1956).

Mark I Perceptron can classify 20x20 images.

Image Credit: https://www.reddit.com/r/interestingasfuck/comments/e8a8oy/frank_rosenblatt_with_a_mark_i_perceptron/

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)
- Discovery of orientation selectivity and columnar organization in the cat’s visual cortex (Hubel and Wiesel 1959)

History of Deep Learning (Connectionism)

- Artificial Neurons (McCulloch and Pitts 1943)
- Hebbian Learning Rule (Donald Hebb 1949)
- SNARC (Minsky and Edmunds 1951)
- Perceptron (Frank Rosenblatt 1958)
- Discovery of orientation selectivity and columnar organization in the cat's visual cortex (Hubel and Wiesel 1959)

Their breakthrough discoveries about the visual system and visual processing earned them the Nobel Prize for Physiology or Medicine in 1981.

Image Credit: https://braintour.harvard.edu/archives/portfolio-items/hubel-and-wiesel

History of Deep Learning (Connectionism)

- The “Perceptrons” book ([Minsky and Papert 1969](#))

In 1969, Marvin Minsky and Seymour Papert publish a book, “*Perceptrons: An Introduction to Computational Geometry*”, highlighting the limitations of simple neural networks, e.g., Perceptrons can not solve XOR problem.

This book contributed to the so-called AI winter of the 1980s.

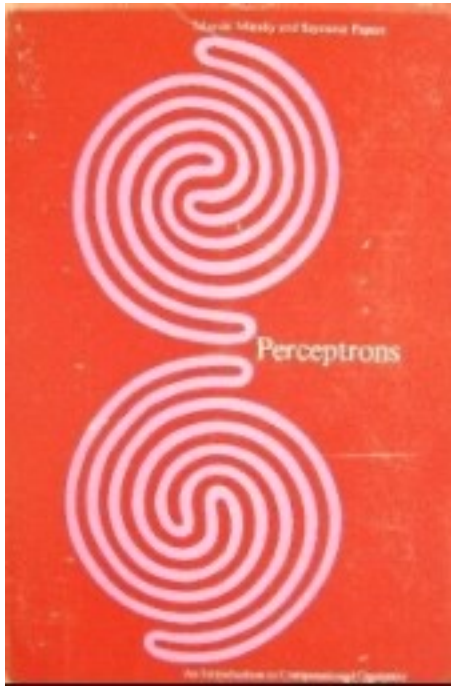


Image Credit: http://harveycohen.net/image/perceptron.html

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)

In 1969, Marvin Minsky and Seymour Papert publish a book, "*Perceptrons: An Introduction to Computational Geometry*", highlighting the limitations of simple neural networks, e.g., Perceptrons can not solve XOR problem.

This book contributed to the so-called AI winter of the 1980s.

Can you tell the difference between two shapes in the cover?

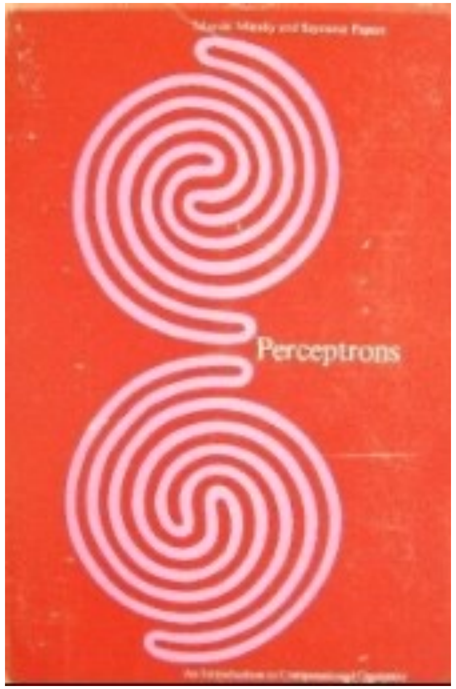


Image Credit: http://harveycohen.net/image/perceptron.html

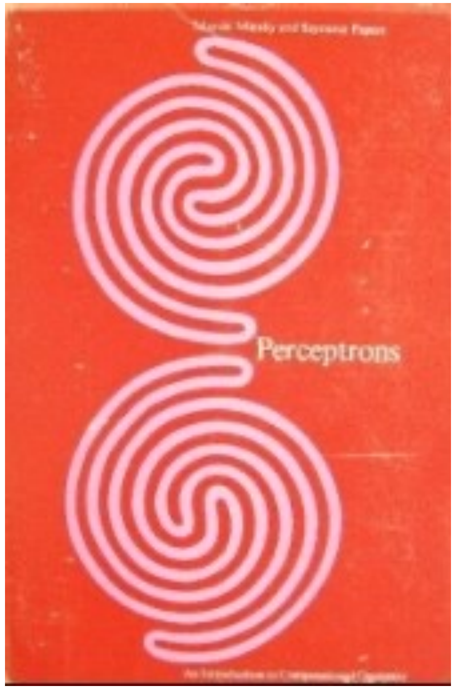
History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)

In 1969, Marvin Minsky and Seymour Papert publish a book, "*Perceptrons: An Introduction to Computational Geometry*", highlighting the limitations of simple neural networks, e.g., Perceptrons can not solve XOR problem.

This book contributed to the so-called AI winter of the 1980s.

Can you tell the difference between two shapes in the cover?



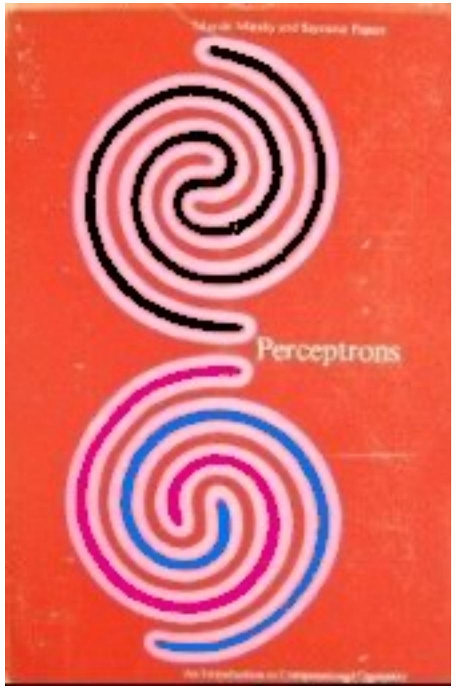


Image Credit: http://harveycohen.net/image/perceptron.html

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)

Image Credit: https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597

History of Deep Learning (Connectionism)

- The "Perceptrons" book ([Minsky and Papert 1969](#))
- Neocognitron ([Fukushima 1979](#))

Inspired by the model proposed by Hubel & Wiesel in 1959, Fukushima proposed the first convolutional neural network architecture for Japanese handwritten character recognition.

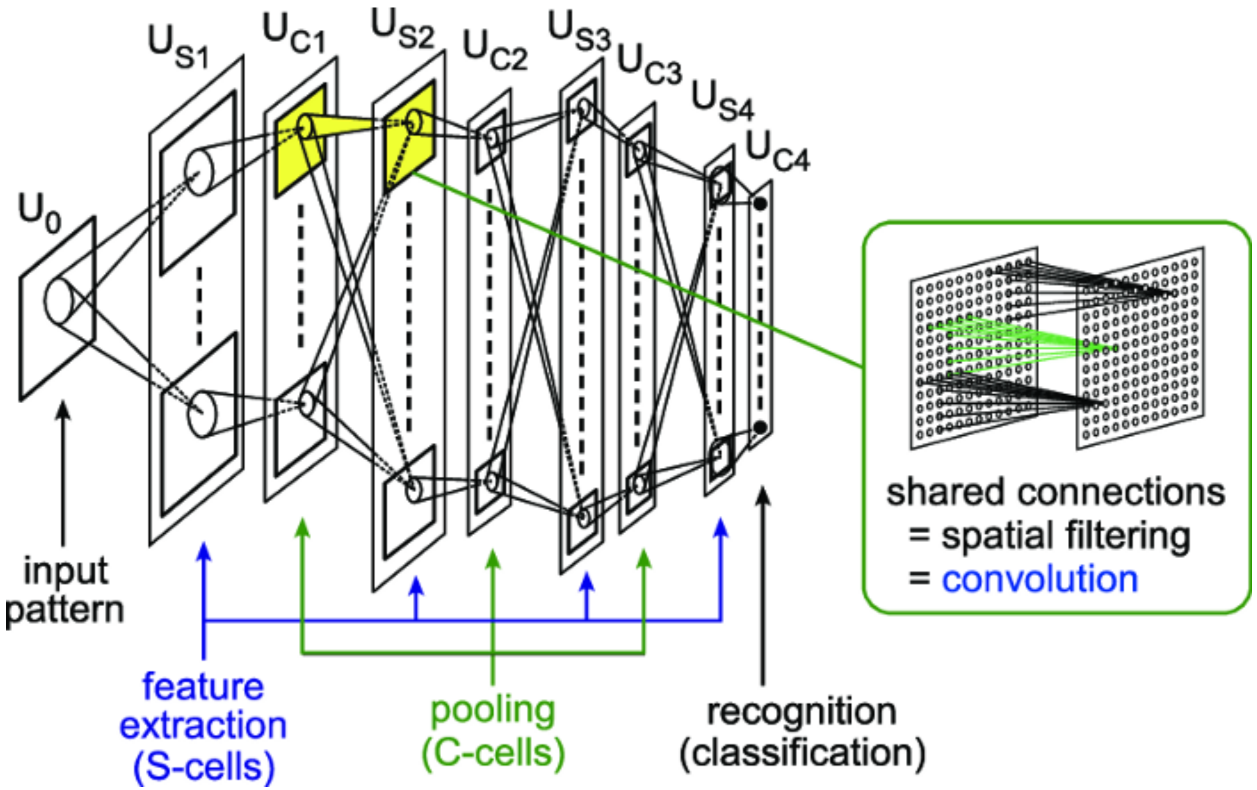


Image Credit: https://medium.com/appyhigh-technology-blog/convolutional-neural-networks-a-brief-history-of-their-evolution-ee3405568597

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)

Image Credit: https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)

Inspired by Ising model in statistical physics, it consists of fully-connected variables with deterministic binary states and an energy function.

It is a recurrent neural network (RNN).

It learns to memorize data via energy minimization, thus being able to simulate associative memory.

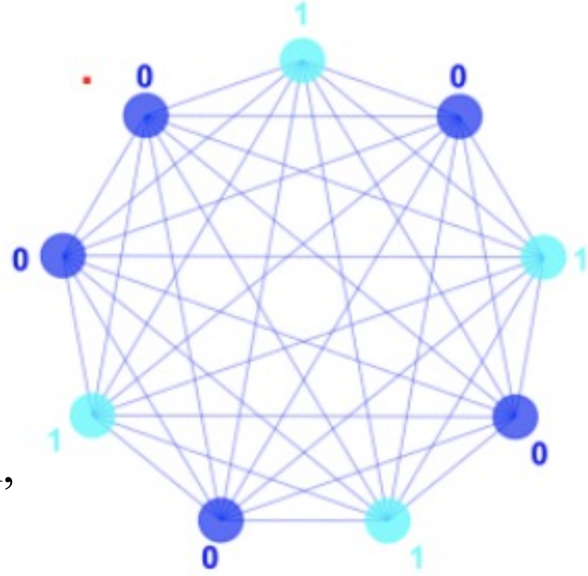


Image Credit: https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073

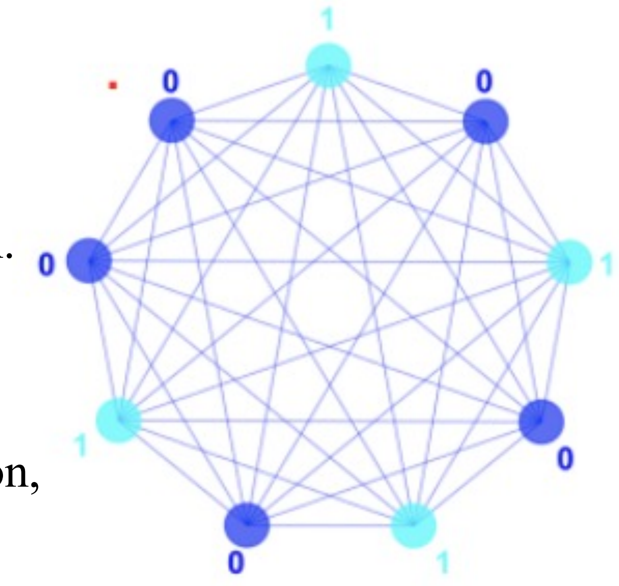
History of Deep Learning (Connectionism)

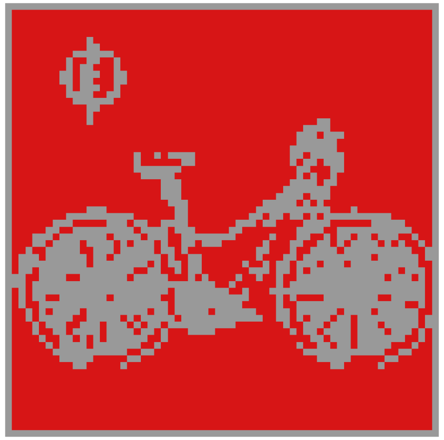
- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)

Inspired by Ising model in statistical physics, it consists of fully-connected variables with deterministic binary states and an energy function.

It is a recurrent neural network (RNN).

It learns to memorize data via energy minimization, thus being able to simulate associative memory.





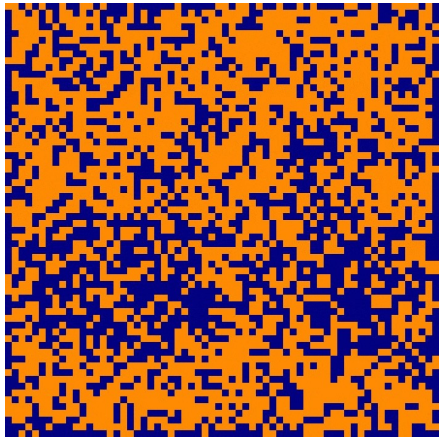


Image Credit: https://towardsdatascience.com/hopfield-networks-neural-memory-machines-4c94be821073

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)

History of Deep Learning (Connectionism)

- The “Perceptrons” book ([Minsky and Papert 1969])
- Neocognitron ([Fukushima 1979])
- Hopfield Networks ([Hopfield 1982])
- Boltzmann Machines ([Hinton and Sejnowski 1983])

It generalizes Hopfield Networks by introducing 1) stochastic binary states 2) hidden/latent variables

The study of deep (layer-structured) Boltzmann machines led to the inception of deep learning in 2006.

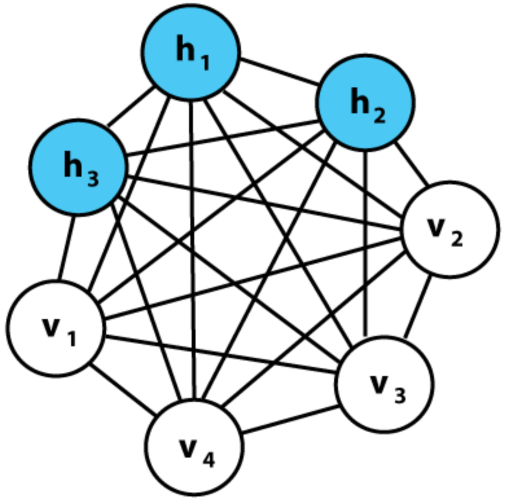


Image Credit: https://en.wikipedia.org/wiki/Boltzmann_machine

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)

The most successful learning algorithm so far for training neural networks!

Learning representations by back-propagating errors

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

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the desired state vector of the output units for each state vector of the input units. If the input units are directly connected to the output units it is relatively easy to find learning rules that iteratively adjust the relative strengths of the connections so as to progressively reduce the difference between the actual and desired output vectors[2]. Learning becomes more interesting but more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input–output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j, to unit j is a linear function of the outputs, y_i, of the units that are connected to j and of the weights, w_{ji}, on these connections

$$x_j = \sum_i y_i w_{ji} \qquad (1)$$

Units can be given biases by introducing an extra input to each unit which always has a value of 1. The weight on this extra input is called the bias and is equivalent to a threshold of the opposite sign. It can be treated just like the other weights.

A unit has a real-valued output, y_j, which is a non-linear function of its total input

$$y_j = \frac{1}{1+e^{-x_j}} \qquad (2)$$

† To whom correspondence should be addressed.

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)

History of Deep Learning (Connectionism)

- The "Perceptrons" book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)

Lecun's demon of CNNs from 1993

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)
- Long-short term memory (Hochreiter and Schmidhuber 1997)

History of Deep Learning (Connectionism)

- The “Perceptrons” book (Minsky and Papert 1969)
- Neocognitron (Fukushima 1979)
- Hopfield Networks (Hopfield 1982)
- Boltzmann Machines (Hinton and Sejnowski 1983)
- Backpropagation (BP) (Kelley 1960, Bryson 1961, Dreyfus 1962, Bryson and Ho 1969, Linnainmaa 1970, Dreyfus, 1973, Werbos 1974, Rumelhart, Hinton, Williams 1986)
- Use BP to train CNNs for image recognition (LeCun et al. 1989)
- Long-short term memory (Hochreiter and Schmidhuber 1997)

It partially resolves the vanishing gradient problem in training RNNs!

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Minh et al. 2015), AlphaGO (Silver et al. 2016)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Minh et al. 2015), AlphaGO (Silver et al. 2016)
- Breakthrough in natural language processing: Seq2seq (Sutskever et al. 2014), Transformers (Vaswani et al. 2017), GPT-3 (Brown et al. 2020), ChatGPT (OpenAI 2022)

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Minh et al. 2015), AlphaGO (Silver et al. 2016)
- Breakthrough in natural language processing: Seq2seq (Sutskever et al. 2014), Transformers (Vaswani et al. 2017), GPT-3 (Brown et al. 2020), ChatGPT (OpenAI 2022)
- Breakthrough in protein structure prediction: AlphaFold (Jumper et al. 2021)

……

History of Deep Learning (Connectionism)

Yann LeCun, Geoffrey Hinton, and Yoshua Bengio received the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.

Image Credit: https://www.nytimes.com/2019/03/27/technology/turing-award-ai.html

History of Deep Learning (Connectionism)

- Deep belief networks (DBN) (Hinton et al. 2006)
- Breakthrough in speech recognition (Dahl et al. 2011)
- Breakthrough in computer vision: AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016)
- Breakthrough in games: DQN for Atari (Minh et al. 2015), AlphaGO (Silver et al. 2016)
- Breakthrough in natural language processing: Seq2seq (Sutskever et al. 2014), Transformers (Vaswani et al. 2017), GPT-3 (Brown et al. 2020), ChatGPT (OpenAI 2022)
- Breakthrough in protein structure prediction: AlphaFold (Jumper et al. 2021)

……

The future depends on some graduate student who is deeply suspicious of everything I have said.

\- Geoffrey Hinton

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - Taxonomy & Connections to ML/AI/Statistics

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - **Modern Applications**
  - Taxonomy & Connections to ML/AI/Statistics

Applications of Deep Learning

Speech Recognition, Personal Assistants

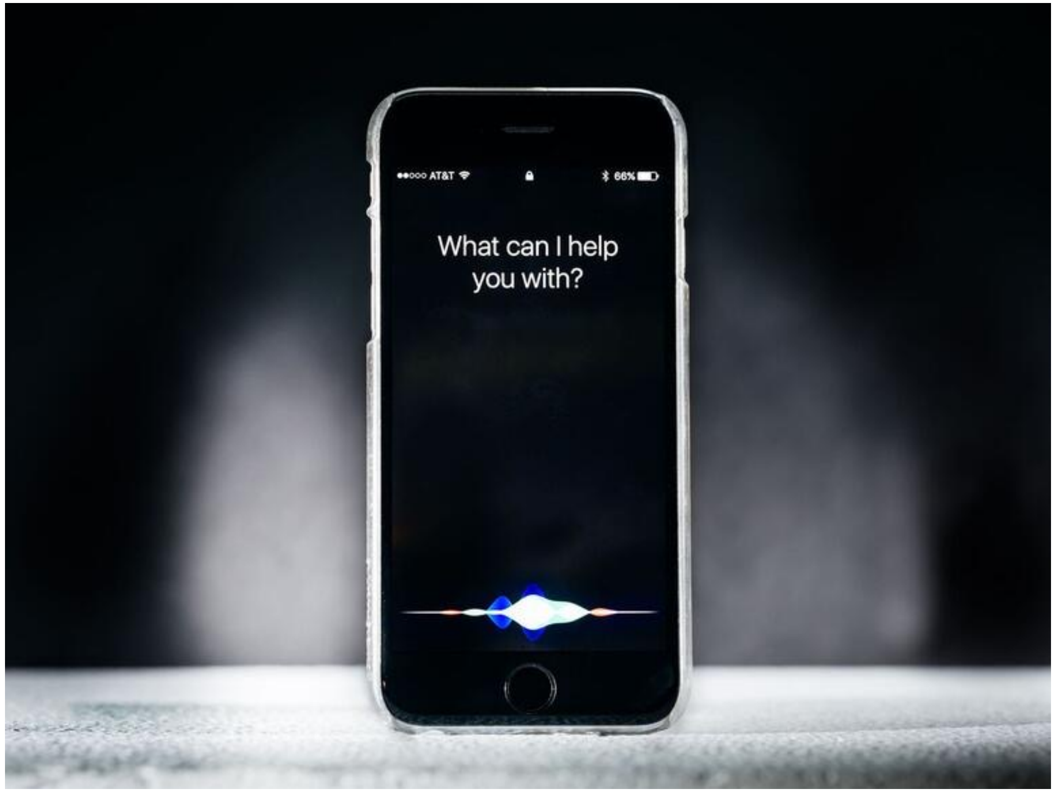
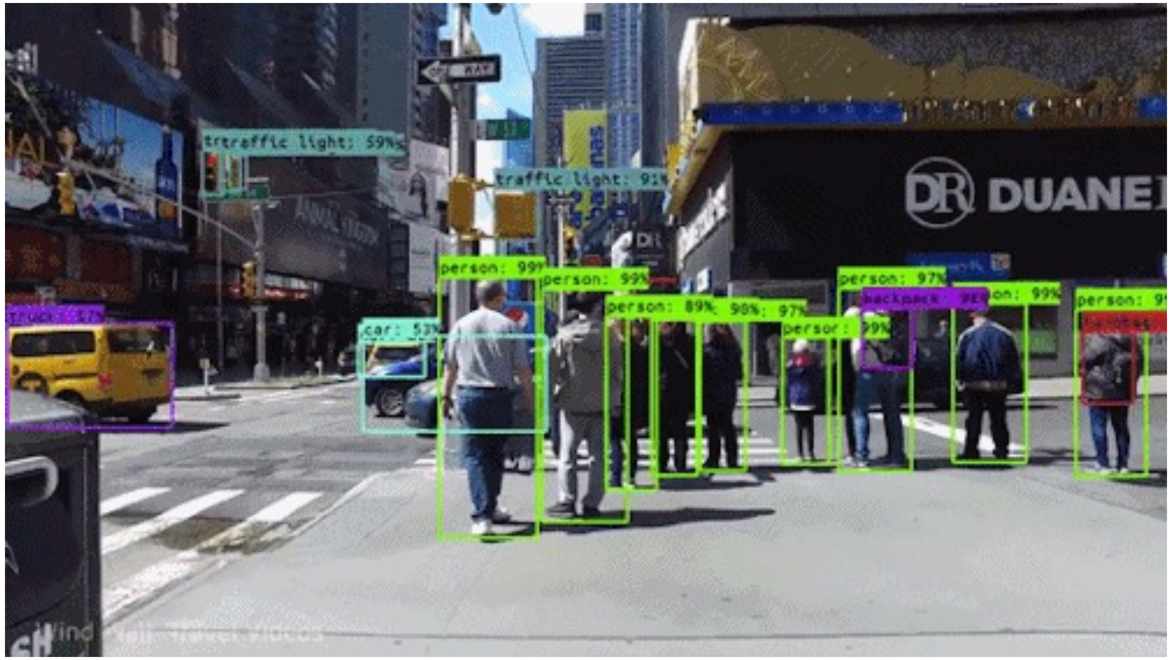


Image Credit: https://www.techrepublic.com/article/apples-siri-the-smart-persons-guide/
https://www.pcmag.com/news/amazon-echo-vs-google-home-which-smart-speaker-is-best

Applications of Deep Learning

Computer Vision/Graphics, e.g., Object detection, Rendering



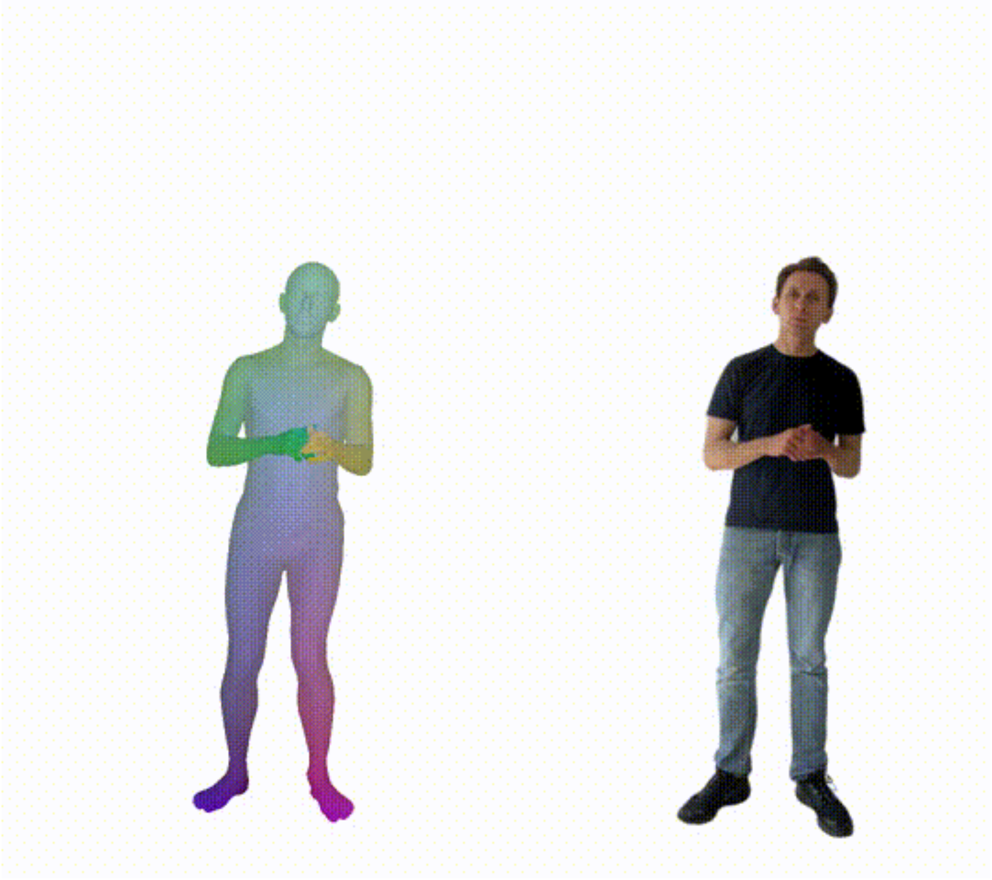


Image Credit: https://github.com/sergeyprokudin/smplpix
https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e

Applications of Deep Learning

Virtual/Augmented Reality

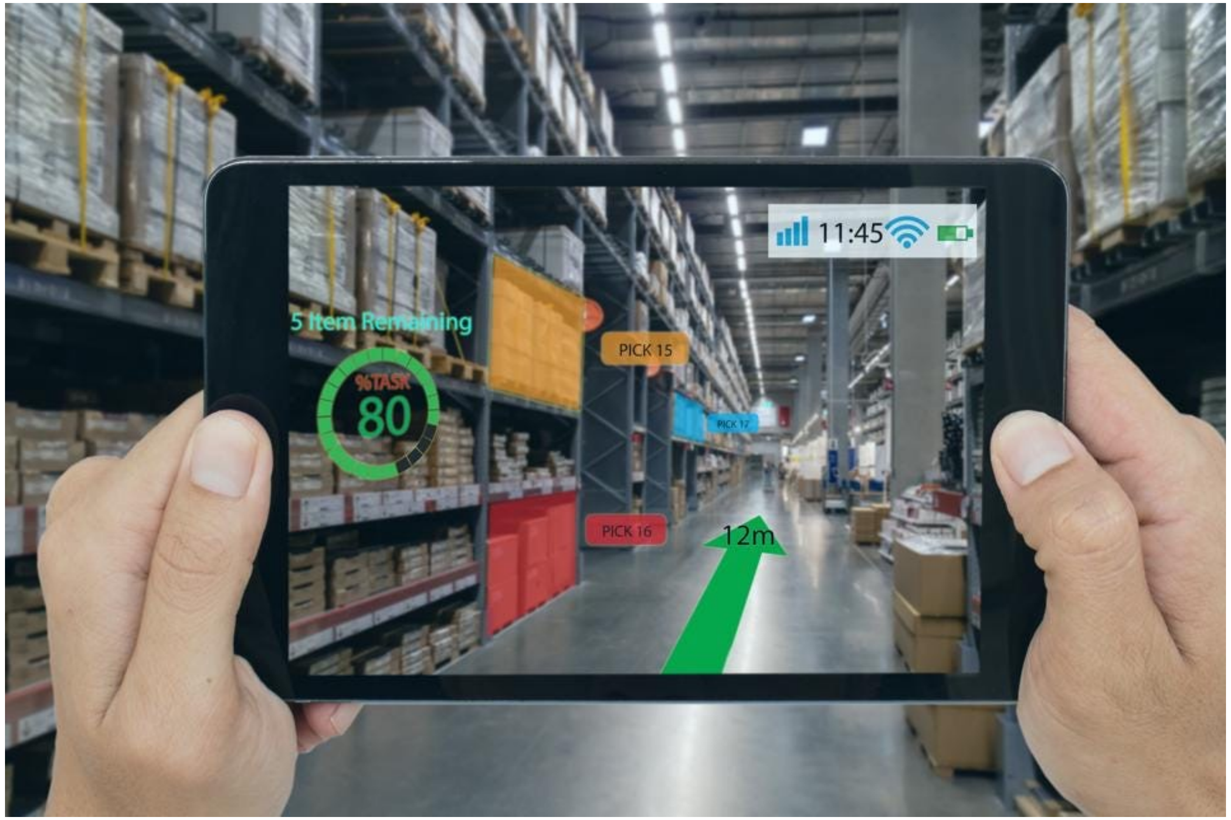


Image Credit: https://www.businessinsider.com/scott-galloway-metaverse-future-not-facebook-2021-8
https://www.forbes.com/sites/theyec/2019/02/06/augmented-reality-in-business-how-ar-may-change-the-way-we-work/?sh=36011cd851e5

Applications of Deep Learning

Robotics, Autonomous Driving

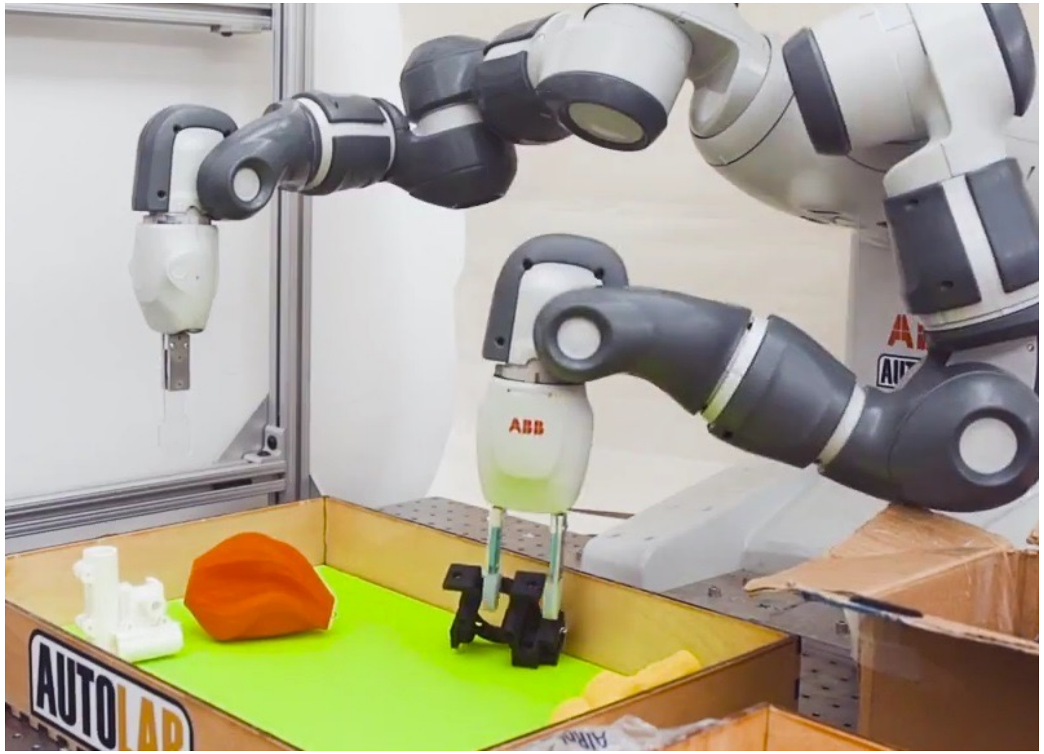


Image Credit: https://techcrunch.com/2017/05/26/this-robot-arms-ai-thinks-like-we-do-about-how-to-grab-something/
https://techcrunch.com/2018/10/30/waymo-takes-the-wheel-self-driving-cars-go-fully-driverless-on-california-roads/

Applications of Deep Learning

Protein structure prediction, Drug discovery

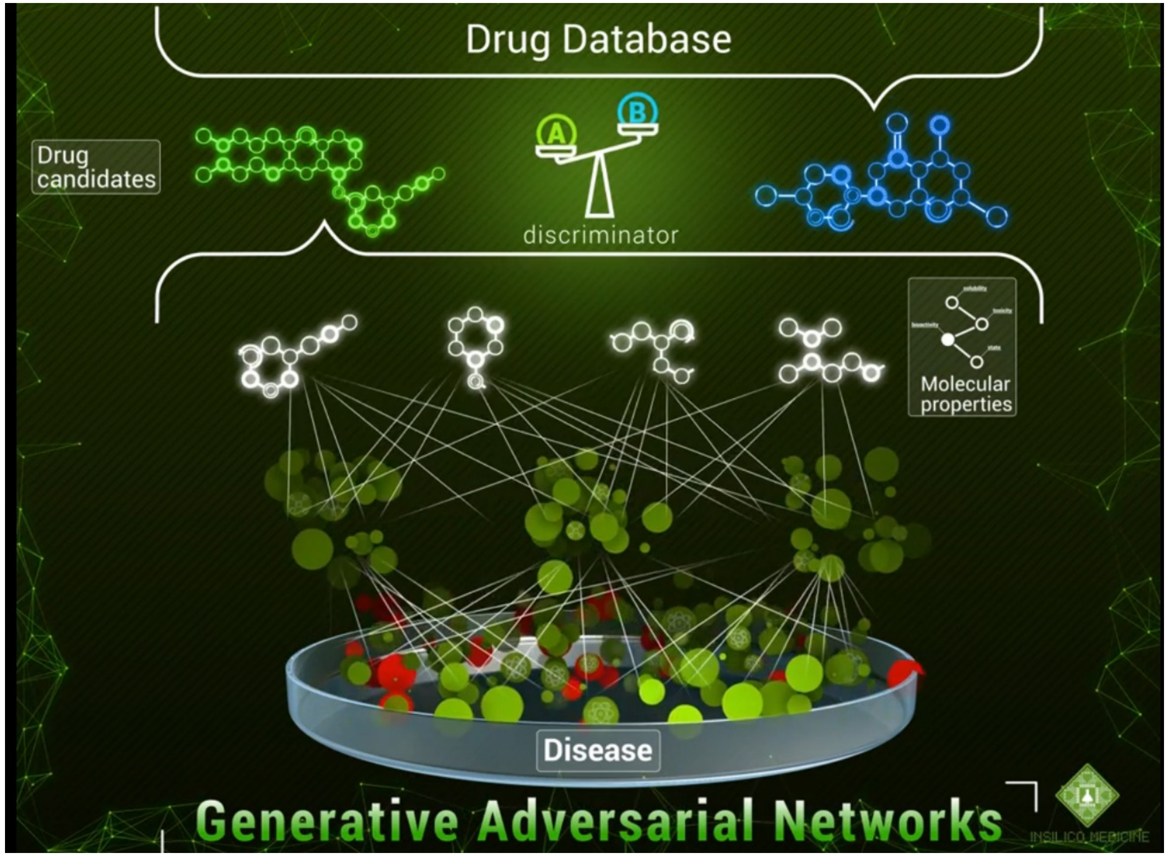
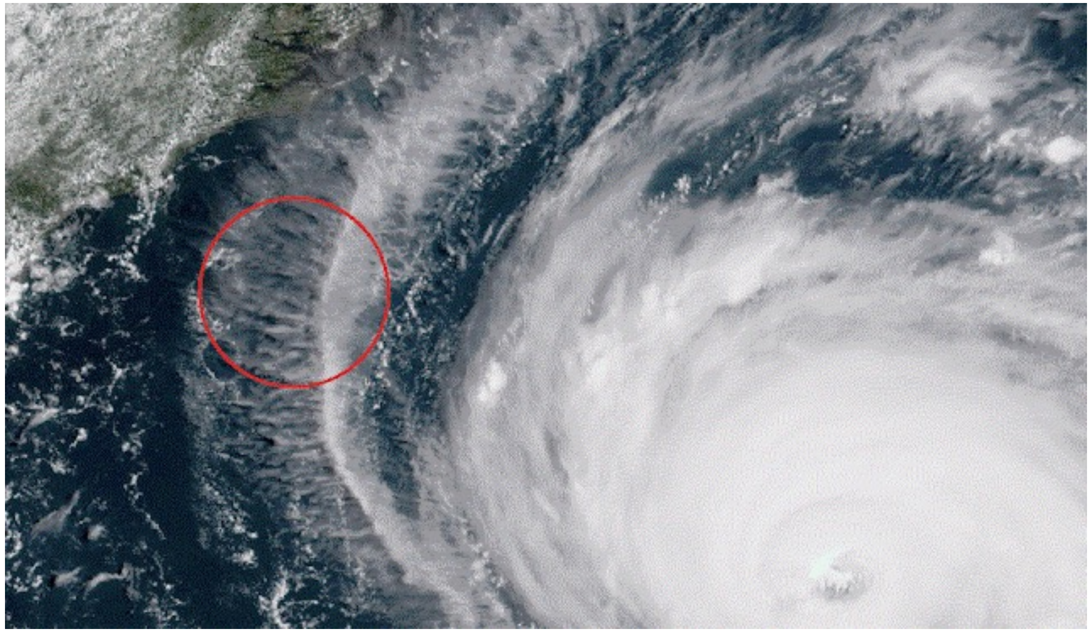


Image Credit: https://www.nature.com/articles/d41586-020-03348-4
https://medium.com/neuromation-blog/creating-molecules-from-scratch-i-drug-discovery-with-generative-adversarial-networks-9d42cc496fc6

Applications of Deep Learning

Simulation of Weather Models, Fluid Simulation



Satellite photo of a hurricane, at both full resolution and simulated resolution in a state-of-the-art weather model. Cumulus clouds (e.g., in the red circle) are responsible for heavy rainfall, but in the weather model the details are entirely blurred out.

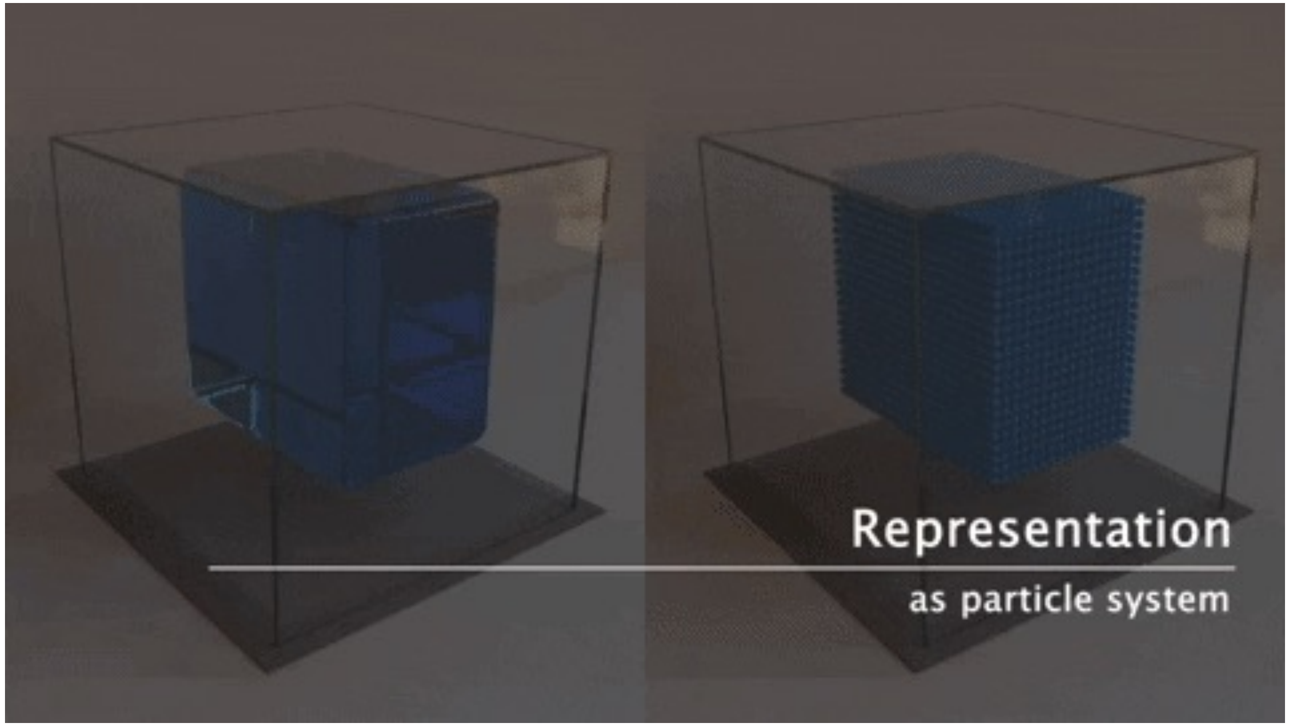


Image Credit: https://ai.googleblog.com/2019/07/learning-better-simulation-methods-for.html
https://meerasridhar23.medium.com/lets-10x-physics-simulations-38ad6fc5cae5

Applications of Deep Learning

Text/Program Generation

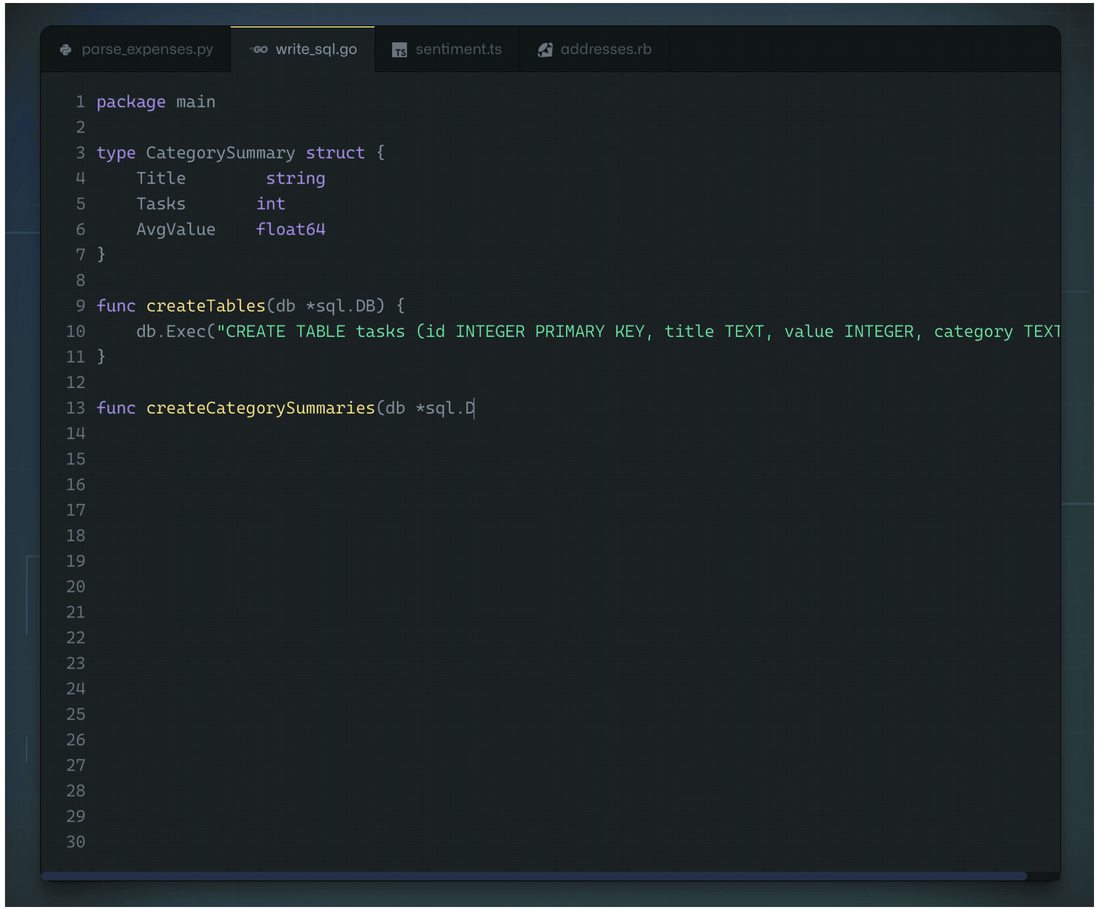
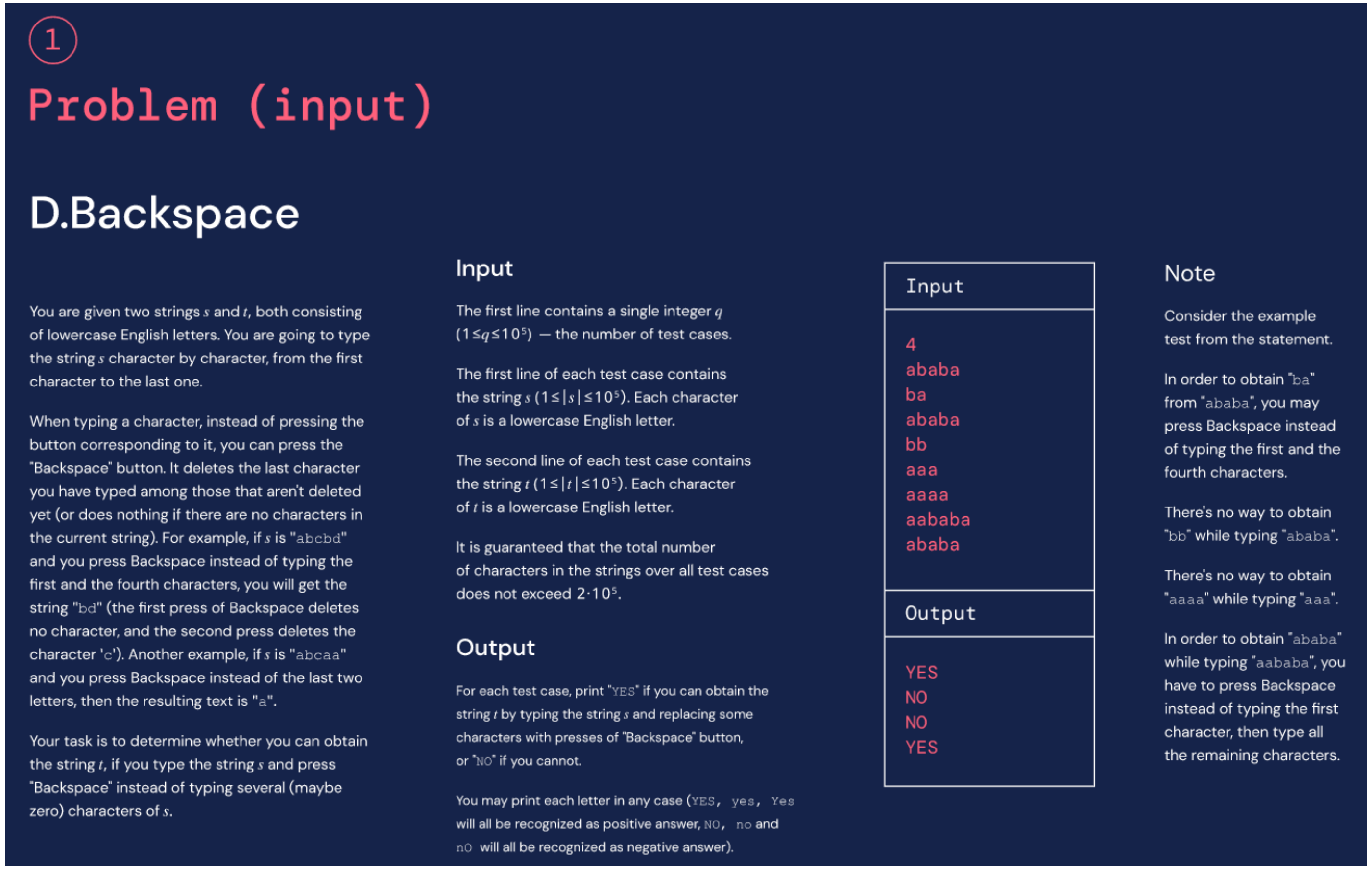


Image Credit: https://techcrunch.com/2020/11/12/othersideai-raises-2-6m-to-let-gpt-3-write-your-emails-for-you/
https://techcrunch.com/2021/06/29/github-previews-new-ai-tool-that-makes-coding-suggestions/

Applications of Deep Learning

Competitive Programming (AlphaCode)


1

Problem (input)

D.Backspace

You are given two strings s and t, both consisting of lowercase English letters. You are going to type the string s character by character, from the first character to the last one.

When typing a character, instead of pressing the button corresponding to it, you can press the "Backspace" button. It deletes the last character you have typed among those that aren't deleted yet (or does nothing if there are no characters in the current string). For example, if s is "abcbd" and you press Backspace instead of typing the first and the fourth characters, you will get the string "bd" (the first press of Backspace deletes no character, and the second press deletes the character 'c'). Another example, if s is "abcaa" and you press Backspace instead of the last two letters, then the resulting text is "a".

Your task is to determine whether you can obtain the string t, if you type the string s and press "Backspace" instead of typing several (maybe zero) characters of s.

Input

The first line contains a single integer q ($1 \le q \le 10^5$) — the number of test cases.

The first line of each test case contains the string s ($1 \le |s| \le 10^5$). Each character of s is a lowercase English letter.

The second line of each test case contains the string t ($1 \le |t| \le 10^5$). Each character of t is a lowercase English letter.

It is guaranteed that the total number of characters in the strings over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print "YES" if you can obtain the string t by typing the string s and replacing some characters with presses of "Backspace" button, or "NO" if you cannot.

You may print each letter in any case (YES, yes, Yes will all be recognized as positive answer, NO, no and nO will all be recognized as negative answer).

Input

```
4
ababa
ba
ababa
bb
aaa
aaaa
aababa
ababa
```

Output

```
YES
NO
NO
YES
```

Note

Consider the example test from the statement.

In order to obtain "ba" from "ababa", you may press Backspace instead of typing the first and the fourth characters.

There's no way to obtain "bb" while typing "ababa".

There's no way to obtain "aaaa" while typing "aaa".

In order to obtain "ababa" while typing "aababa", you have to press Backspace instead of typing the first character, then type all the remaining characters.

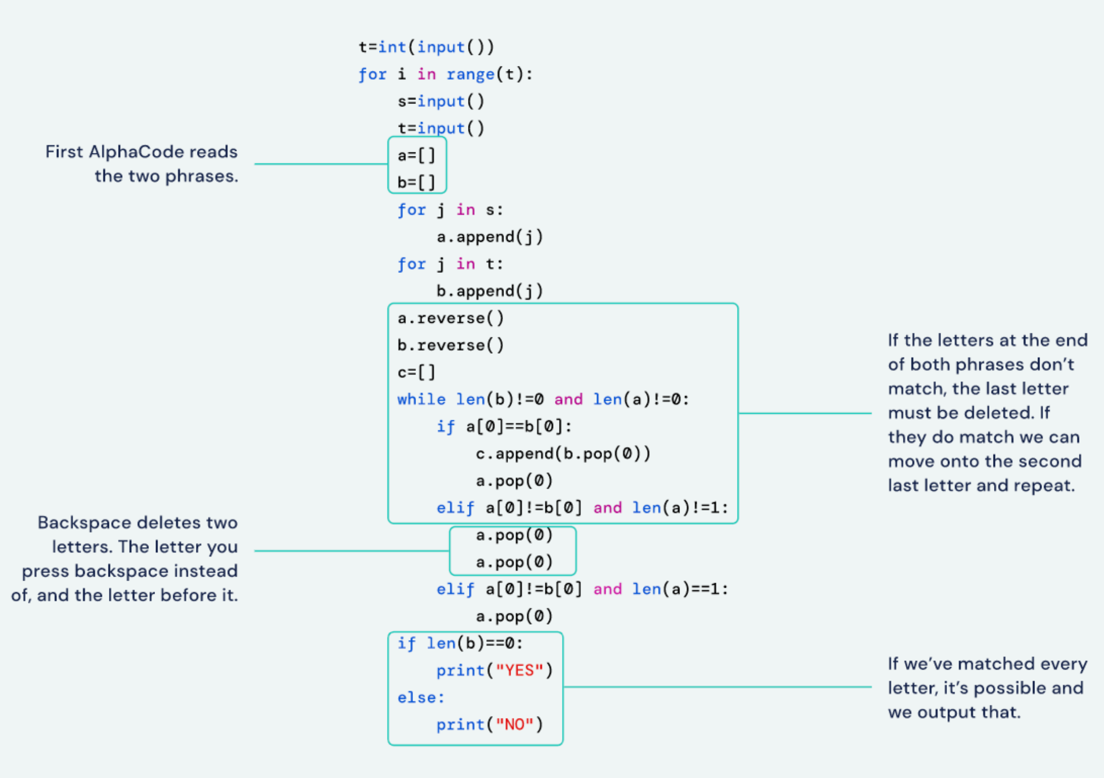


Image Credit: https://www.deepmind.com/blog/competitive-programming-with-alphacode

Applications of Deep Learning

Chatbot (ChatGPT)

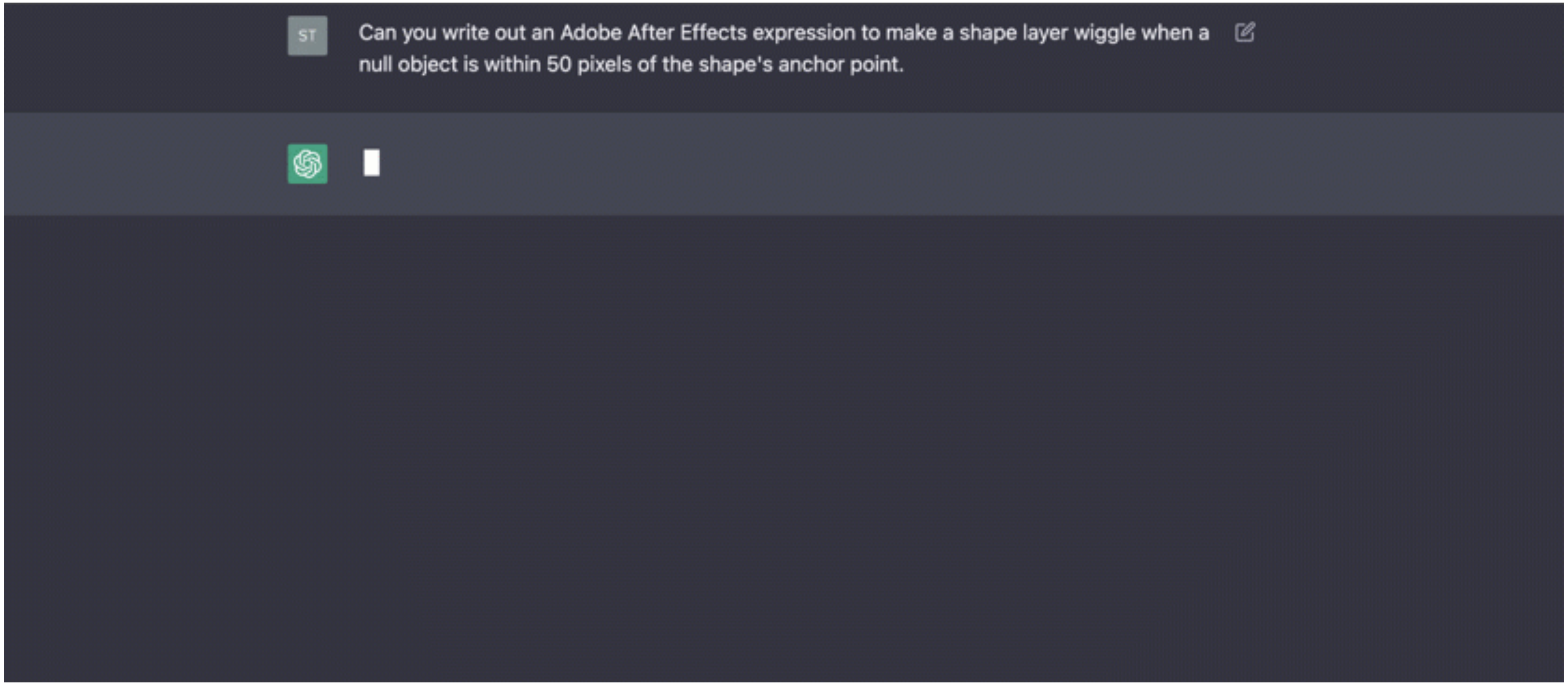


Image Credit: https://fstoppers.com/post-production/turning-ideas-animations-chatgpt-and-after-effects-621834

Applications of Deep Learning

Chatbot (ChatGPT)

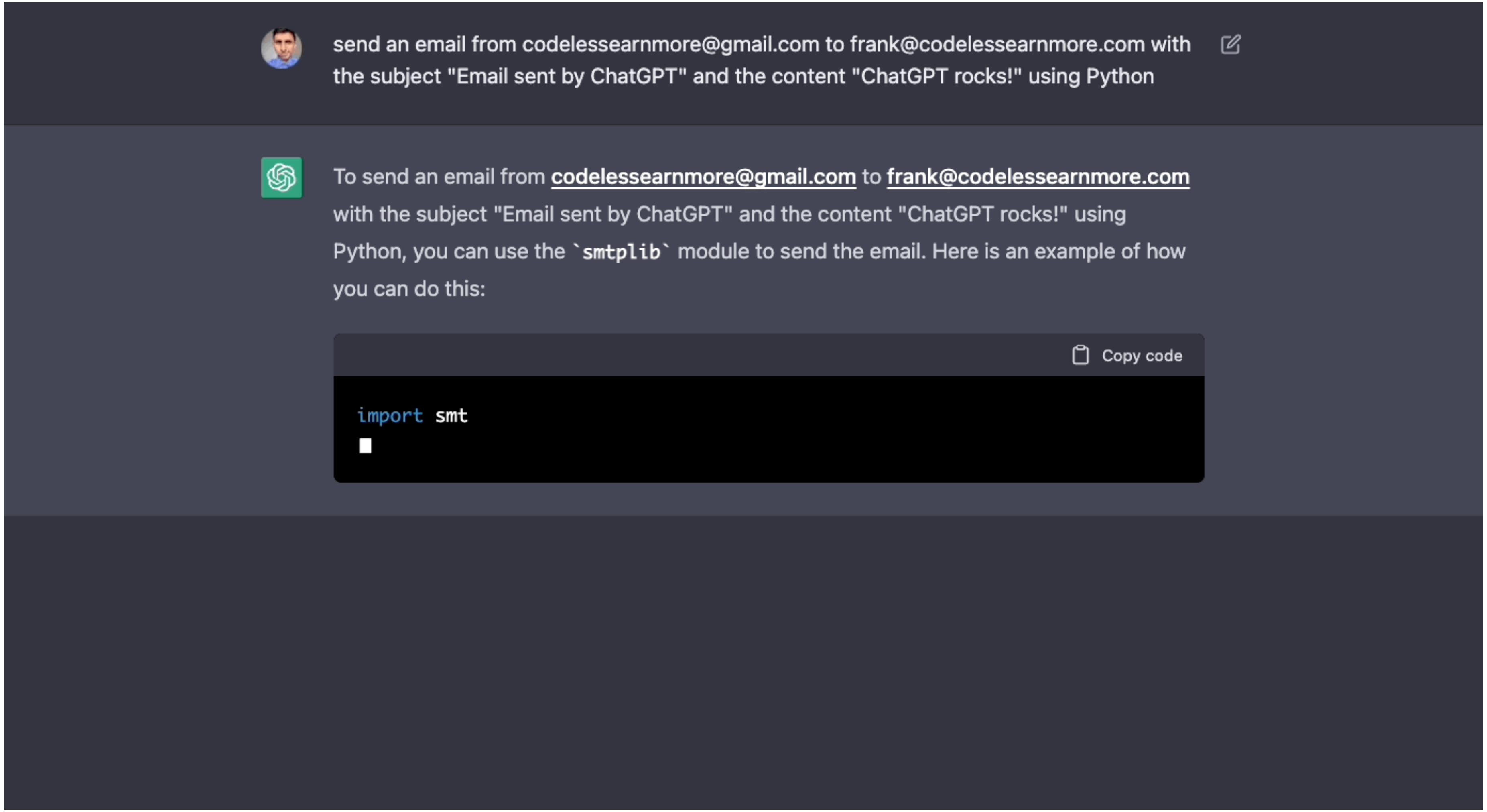


Image Credit: https://medium.com/geekculture/chatgpt-4-jobs-that-will-change-or-be-fully-replaced-by-this-ai-powered-chatbot-97e8118b2475

Outline

- Course Information
- Introduction to Deep Learning
    - History
    - Modern Applications
    - Taxonomy & Connections to ML/AI/Statistics

Outline

- Course Information
- Introduction to Deep Learning
  - History
  - Modern Applications
  - **Taxonomy & Connections to ML/AI/Statistics**

Subareas of Deep Learning

You can get a rough sense from keywords in ICLR 2023 submissions.

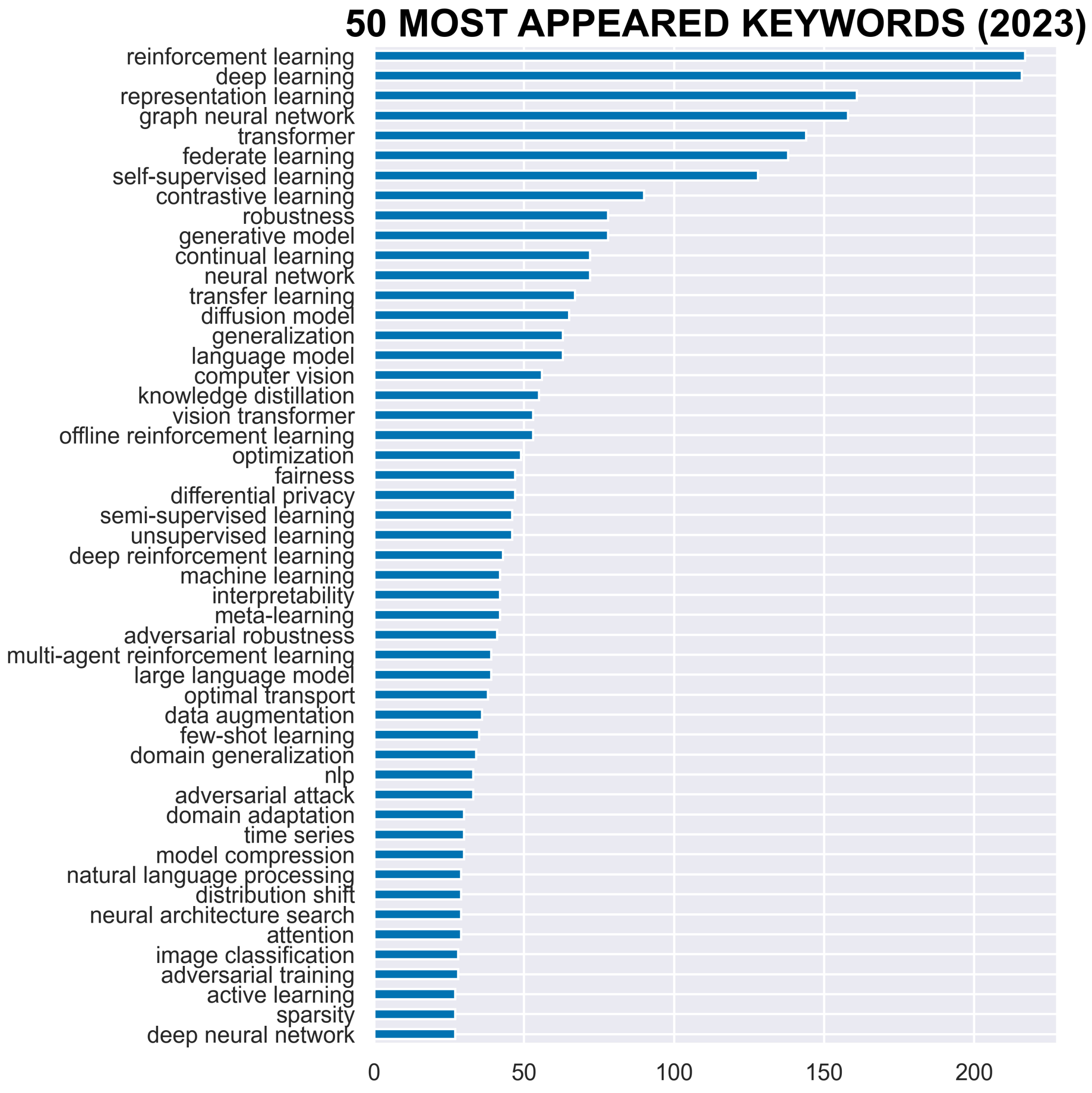


Image Credit: Twitter Image from Petar Veličković

Subareas of Deep Learning

You can get a rough sense from keywords in ICLR 2023 submissions.

Top conferences in DL/ML:

- International Conference on Learning Representations (ICLR)
- Neural Information Processing Systems (NeurIPS)
- International Conference on Machine Learning (ICML)

You can also find good DL/ML papers from top CV/NLP conferences:

- Computer Vision and Pattern Recognition Conference (CVPR)
- International Conference on Computer Vision (ICCV)
- Annual Meeting of the Association for Computational Linguistics (ACL)
- Conference on Empirical Methods in Natural Language Processing (EMNLP)

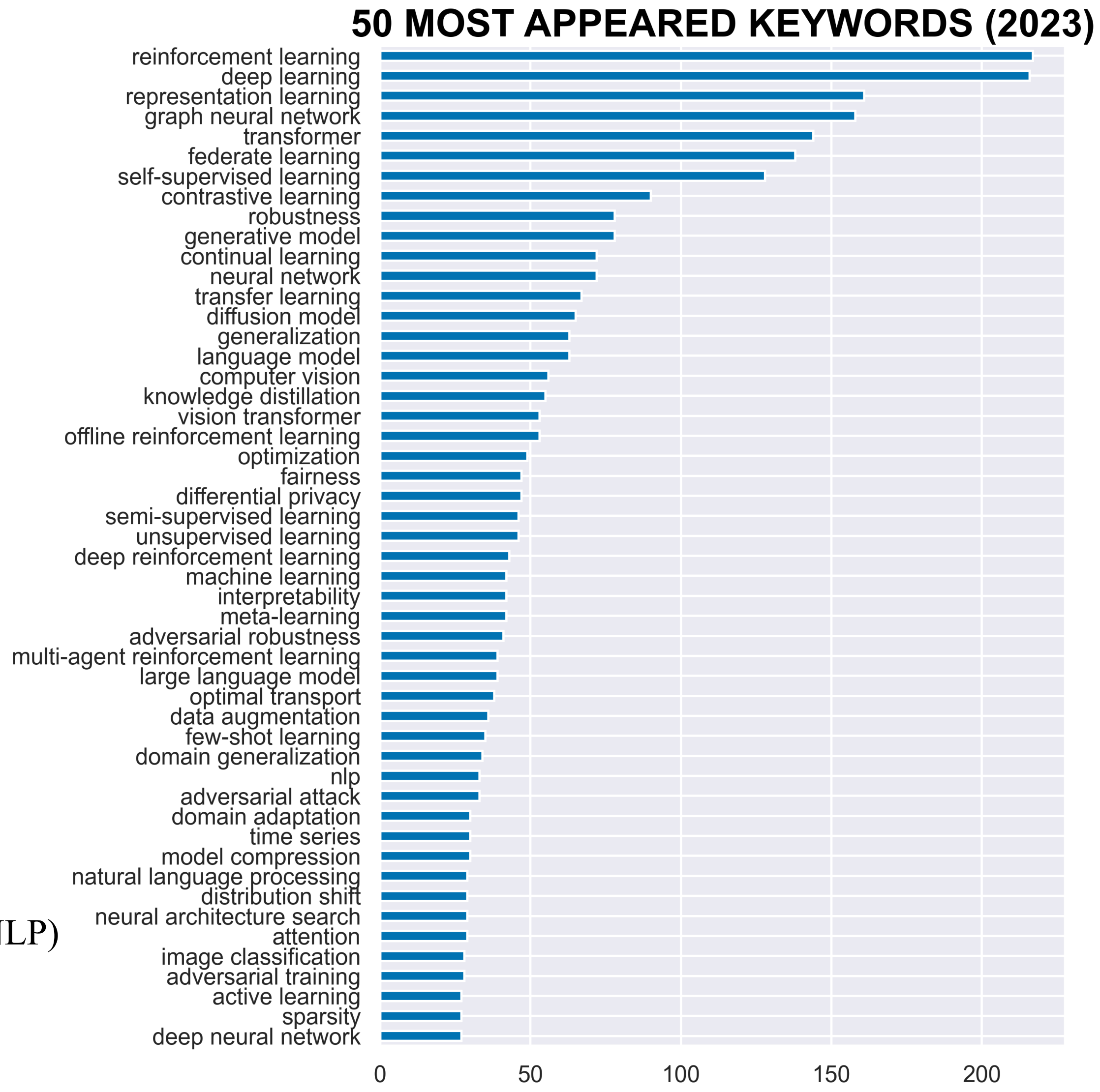
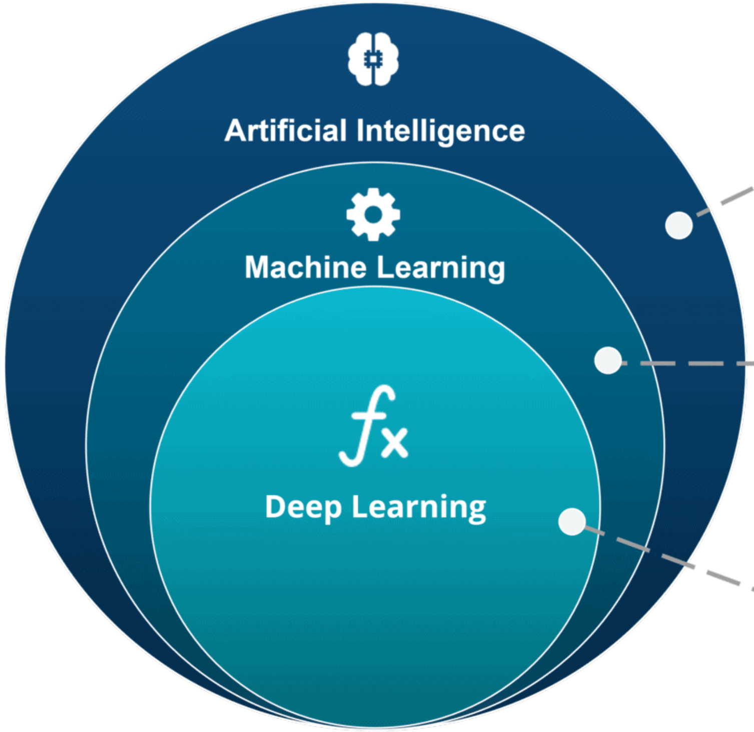


Image Credit: Twitter Image from Petar Veličković

Relationships w. ML & AI



ARTIFICIAL INTELLIGENCE

Aims at mimicking and surpassing human intelligence

MACHINE LEARNING

Aims at building machines that can learn from data

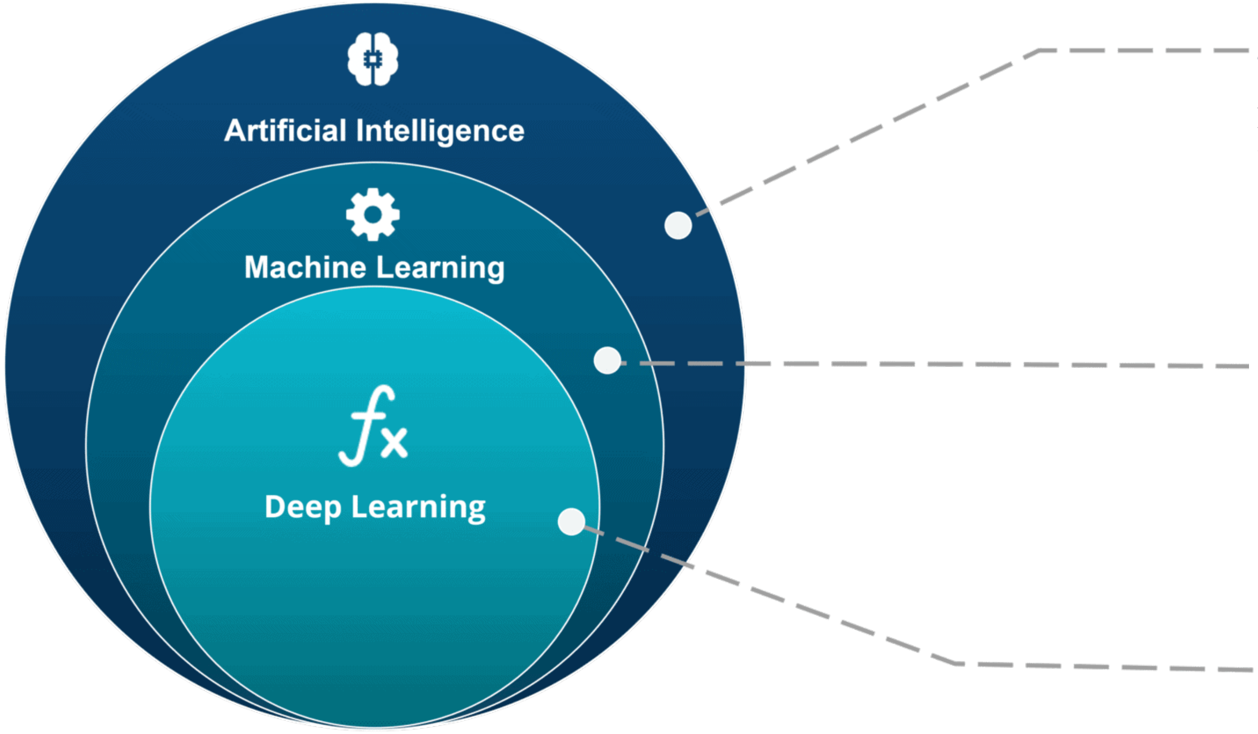
DEEP LEARNING

Aims at building neural networks that can mimic and surpass human intelligence

Image Credit: https://k21academy.com/datascience/deep-learning/dl-vs-ml/

Relationships w. ML & AI

DL brings new techniques and pushes capabilities of AI to an unprecedented level!



ARTIFICIAL INTELLIGENCE

Aims at mimicking and surpassing human intelligence

MACHINE LEARNING

Aims at building machines that can learn from data

DEEP LEARNING

Aims at building neural networks that can mimic and surpass human intelligence

Image Credit: https://k21academy.com/datascience/deep-learning/dl-vs-ml/

DL/ML vs. Statistics

- Some believe ML = Statistics

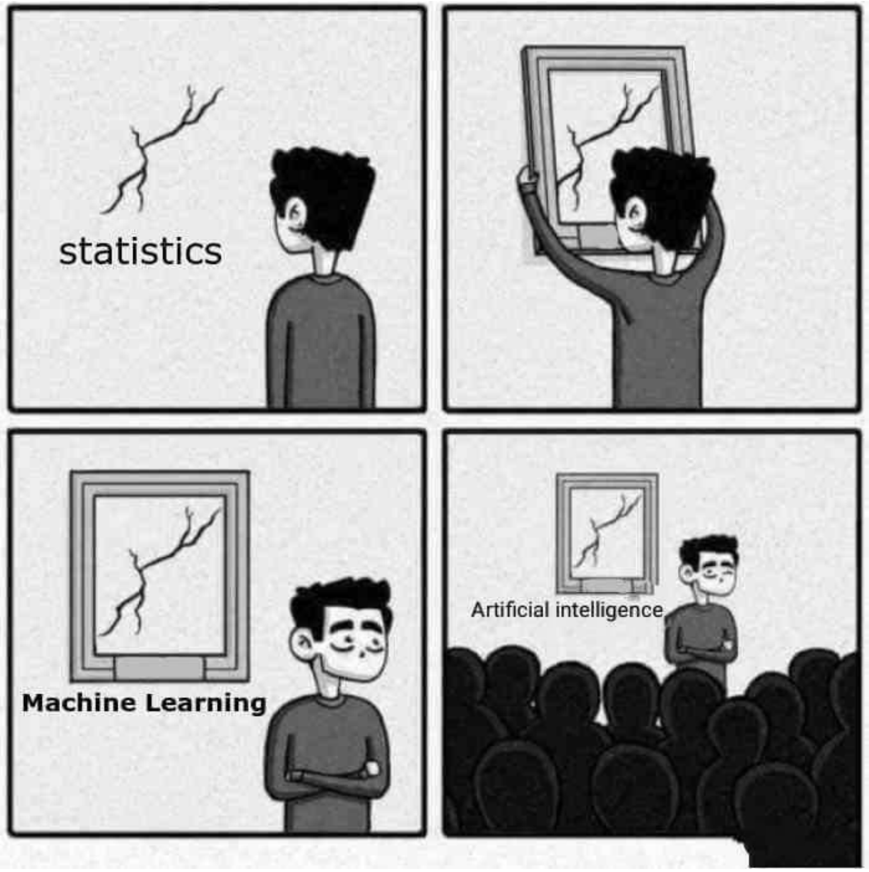


Image Credit: https://towardsdatascience.com/no-machine-learning-is-not-just-glorified-statistics-26d3952234e3

DL/ML vs. Statistics

- Some believe ML = Statistics

  Yes. Both aim at building models to get knowledge from data and share a lot in methodologies.

Image Credit: https://xkcd.com/1838/

DL/ML vs. Statistics

- Some believe ML = Statistics

  Yes. Both aim at building models to get knowledge from data and share a lot in methodologies.

  No. ML emphasizes more about computation and prediction, whereas statistics cares other things like model checking.

Statistical Science
2001, Vol. 16, No. 3, 199–231

Statistical Modeling: The Two Cultures

Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

A Bit More About AI

Symbolism vs. Connectionism

- Symbolic AI (a.k.a., GOFAI): top-down, logic, symbolic representations, reasoning w.o. much learning

A Bit More About AI

Symbolism vs. Connectionism

- Symbolic AI (a.k.a., GOFAI): top-down, logic, symbolic representations, reasoning w.o. much learning

But it encounters severe difficulties (an excerpt from Wikipedia)

Researchers in the 1960s and the 1970s were convinced that symbolic approaches would eventually succeed in creating a machine with artificial general intelligence and considered this the goal of their field.[30] Herbert Simon predicted, "machines will be capable, within twenty years, of doing any work a man can do".[31] Marvin Minsky agreed, writing, "within a generation ... the problem of creating 'artificial intelligence' will substantially be solved".[32] They had failed to recognize the difficulty of some of the remaining tasks. Progress slowed and in 1974, in response to the criticism of Sir James Lighthill[33] and ongoing pressure from the US Congress to fund more productive projects, both the U.S. and British governments cut off exploratory research in AI. The next few years would later be called an "AI winter", a period when obtaining funding for AI projects was difficult.[8]

A Bit More About AI

Symbolism vs. Connectionism

- Symbolic AI (a.k.a., GOFAI): top-down, logic, symbolic representations, reasoning w.o. much learning

  But it encounters severe difficulties (an excerpt from Wikipedia)

  > Researchers in the 1960s and the 1970s were convinced that symbolic approaches would eventually succeed in creating a machine with artificial general intelligence and considered this the goal of their field.[30] Herbert Simon predicted, "machines will be capable, within twenty years, of doing any work a man can do".[31] Marvin Minsky agreed, writing, "within a generation ... the problem of creating 'artificial intelligence' will substantially be solved".[32] They had failed to recognize the difficulty of some of the remaining tasks. Progress slowed and in 1974, in response to the criticism of Sir James Lighthill[33] and ongoing pressure from the US Congress to fund more productive projects, both the U.S. and British governments cut off exploratory research in AI. The next few years would later be called an "AI winter", a period when obtaining funding for AI projects was difficult.[8]

- Connectionist AI: bottom-up, biological realism, parallel distributed processing, learning

A Bit More About AI

Symbolism vs. Connectionism

- Symbolic AI (a.k.a., GOFAI): top-down, logic, symbolic representations, reasoning w.o. much learning

  But it encounters severe difficulties (an excerpt from Wikipedia)

  Researchers in the 1960s and the 1970s were convinced that symbolic approaches would eventually succeed in creating a machine with artificial general intelligence and considered this the goal of their field.[30] Herbert Simon predicted, "machines will be capable, within twenty years, of doing any work a man can do".[31] Marvin Minsky agreed, writing, "within a generation ... the problem of creating 'artificial intelligence' will substantially be solved".[32] They had failed to recognize the difficulty of some of the remaining tasks. Progress slowed and in 1974, in response to the criticism of Sir James Lighthill[33] and ongoing pressure from the US Congress to fund more productive projects, both the U.S. and British governments cut off exploratory research in AI. The next few years would later be called an "AI winter", a period when obtaining funding for AI projects was difficult.[8]

- Connectionist AI: bottom-up, biological realism, parallel distributed processing, learning

  But DL currently is not good at explicit (logical) reasoning

Taxonomy of DL/ML

- **"Pure" Reinforcement Learning** (cherry)
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning** (icing)
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning** (cake)
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Image Credit: Yann Lecun, NIPS 2016

References

[1] Mcculloch, W. S. & Pitts, W. H. (1943), 'A Logical Calculus of the Ideas Immanent in Nervous Activity', Bulletin of Mathematical Biophysics 5 , 115--133.

[2] Hebb, D. O. (1949), The Organization of Behavior: A Neuropsychological Theory , Wiley , New York .

[3] “Stochastic neural analog reinforcement calculator.” Wikipedia, Wikimedia Foundation, 24 December 2022, en.wikipedia.org/wiki/Stochastic_neural_analog_reinforcement_calculator.

[4] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408.

[5] Minsky, M. & Papert, S. (1969), Perceptrons: An Introduction to Computational Geometry , MIT Press , Cambridge, MA, USA .

[6] Fukushima, Kunihiko (1979). “位置ずれに影響されないパターン認識機構の神経回路のモデル --- ネオコグニトロン ---” [Neural network model for a mechanism of pattern recognition unaffected by shift in position — Neocognitron —]. Trans. IECE (in Japanese). J62-A (10): 658–665.

[7] Hopfield, J. J. (1982), 'Neural Networks and Physical Systems with Emergent Collective Computational Abilities', Proceedings of the National Academy of Sciences, USA 79 , 2554--2558.

[8] Hinton, G. E. & Sejnowski, T. J. (1983), Optimal Perceptual Inference, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[9] Kelley, H. J. (1960). Gradient theory of optimal flight paths. Ars Journal, 30(10), 947-954.

[10] A. E. Bryson. (1961). A gradient method for optimizing multi-stage allocation processes. Proc. Harvard Univ. Symposium on digital computers and their applications.

References

[11] Dreyfus, S. (1962). The numerical solution of variational problems. Journal of Mathematical Analysis and Applications, 5(1), 30-45.

[12] Bryson, A. E., & Ho, Y. C. (2018). Applied optimal control: optimization, estimation, and control. Routledge.

[13] Seppo Linnainmaa (1970). The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki, 6–7.

[14] Dreyfus, S. (1973). The computational solution of optimal control problems with time lag. IEEE Transactions on Automatic Control, 18(4), 383-385.

[15] Werbos, Paul. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University.

[16] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[17] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

[18] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[19] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[20] Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on audio, speech, and language processing, 20(1), 30-42.

References

[21] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[22] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[23] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.

[24] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489.

[25] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[27] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[28] https://openai.com/blog/chatgpt/

[29] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., ... & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589.

Questions?