

# MTH 221

Fundamentals of Machine Learning

Batuhan Bardak

**Lecture 1:** Course outline and basic concepts of ML

**Date:** 03.10.2023



# Plan for today

- Course outline and materials
- Basic concepts and terminology of Machine Learning



# Logistics

- **Instructor:** Batuhan BARDAK
  - [batuhanbardak@gmail.com](mailto:batuhanbardak@gmail.com)
  - bbatuhan@metu.edu.tr
- **Lectures:**
  - Tuesday: 13:40 - 16:30, Online
- **TA:** Burcu Alakuş Çınar
  - [cinarb@metu.edu.tr](mailto:cinarb@metu.edu.tr)
  - Informatics Institute B-120
- **ODTU Class:** <https://odtuclass2023f.metu.edu.tr/course/view.php?id=3924>
- **Sanayi Bakanlığı:** <https://uep.milliteknolojiakademisi.gov.tr/>
- **Github**
  - <https://github.com/bbardakk/ODTU-MTH221-2023>



# What this class is

- **Fundamentals of ML:** supervised learning (e.g., linear regression, logistic regression, svm, boosting, deep learning), unsupervised learning (e.g., k-means, hierarchical clustering, PCA), bias/variance tradeoff, overfitting, advice for applying machine learning
- **More Recent topics of ML:** AutoML, Explainable AI, MLOps



# Prerequisites

- Basic algorithms and data structures
- Basic probability and statistics
- Basic linear algebra
- Good programming skills (especially in Python)



# Grading

- Midterm: %20 (closed book, no notes, no cheat sheet)
- Final: %30 (closed book, no notes, no cheat sheet)
- Homeworks: %20 (with Python)
- Course Project: %30 (done in groups of 2)

**Note:** If you submit your project final reports to the IEEE conference, 20 points will be added to your project grade.



# Course Project

- Competition-oriented
  - Kaggle Team
    - <https://www.kaggle.com/competitions>
- Research
  - Algorithmic Trading
  - Your Own Ideas (A Bit Hard to Accept!)
- Projects should be done in groups (2 people).
- Grading
  - Proposal (% 5)
  - Literature Review (% 5)
  - Novelty (%10)
  - Progress Report (%10)
  - Deployment (%20) /  
Ranking in Competition  
(For Kaggle Teams)
  - Project Presentation - online  
(%20)
  - Final Report (%30)

↓  
*active*



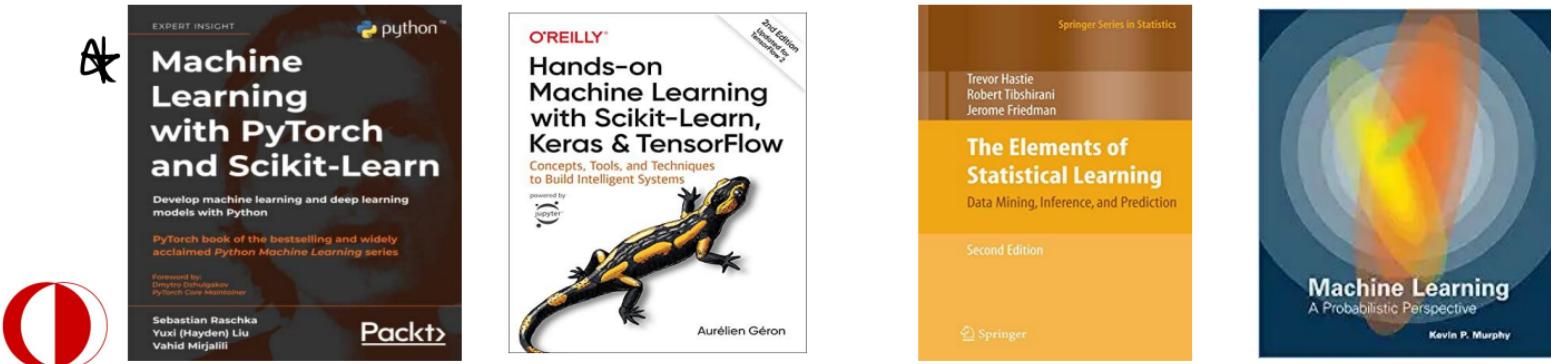
# Academic Integrity

- **Zero tolerance policy:** All occurrences will be reported
- **AI tool policy:** You can use AI-based tools, but you need to provide references. You also need to mention how you used this tool (prompts, etc.) and its contribution to you.



# Reference Books

- Machine Learning with PyTorch and Scikit-Learn, Raschka, 2022
- Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, Geron, 2nd Edition, 2017
- A Course in Machine Learning, Hal Daumé III, 2017
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, 2016
- Machine Learning: A Probabilistic Perspective, Murphy, MIT Press, 2012

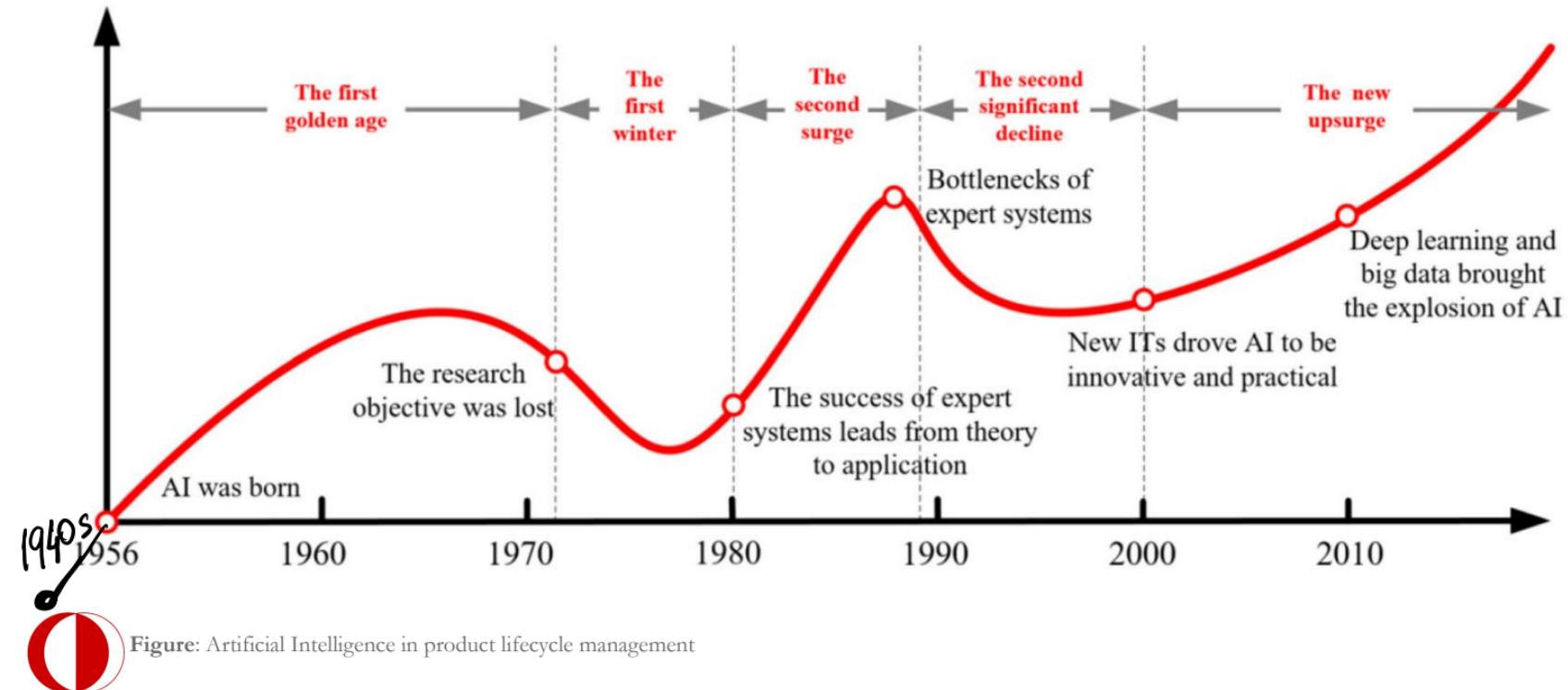


# Course outline (Tentative)

- **Week 1:** Introduction & ML Basics
- **Week 2:** KNN and the curse of dimensionality
- **Week 3:** Perceptron
- **Week 4:** MLE & MAP & Naive Bayes
- **Week 5:** Logistic Regression and Gradient descent
- **Week 6:** Linear Regression
- **Week 7:** Support Vector Machine & Kernels
- **Week 8:** Model Selection
- **Week 9:** Decision Tree & Ensemble Learning
- **Week 10:** Neural Network
- **Week 11:** Deep Learning
- **Week 12:** Guest Lecturer ✓
- **Week 13:** Unsupervised Learning
- **Week 14:** Machine Learning System Design



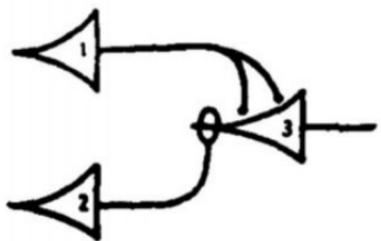
# History of AI



# A Mathematical Model of Brain: McCulloch & Pitts Neuron

## Artificial Neurons

BULLETIN OF  
MATHEMATICAL BIOPHYSICS  
VOLUME 5, 1943



$$N_3(t) \dots \equiv \dots N_1(t-1) \dots \sim N_2(t-1)$$

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

WARREN S. MCCULLOCH AND WALTER PITTS

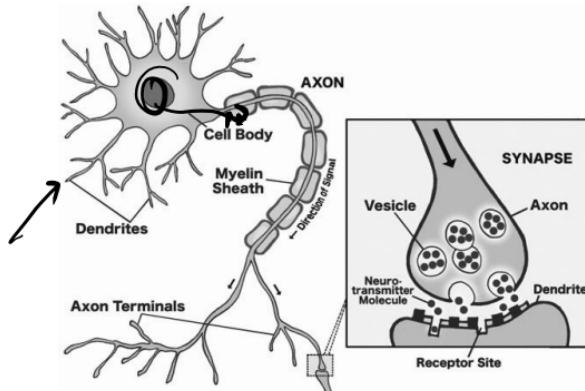
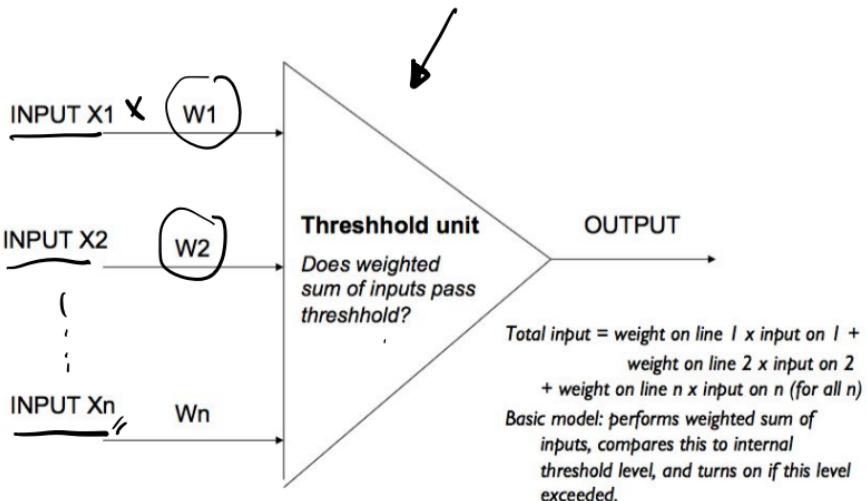
FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,  
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,  
AND THE UNIVERSITY OF CHICAGO

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.



# McCulloch & Pitts

## McCulloch Pitts Neuron



Inputs arrive at dendrites, and axons serve as output channel



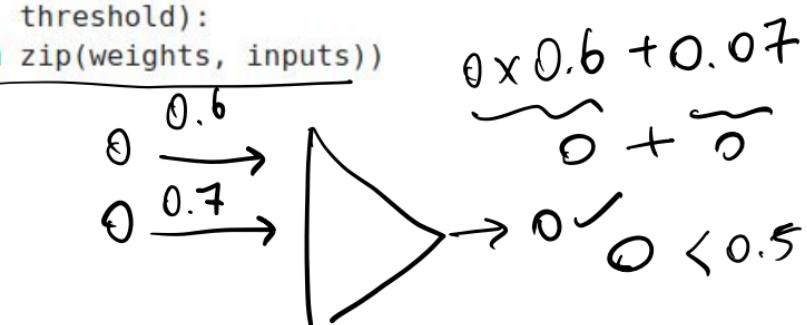
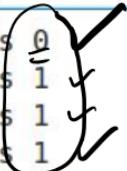
# McCulloch & Pitts

```
: def mcculloch_pitts_neuron(inputs, weights, threshold):
    total_input = sum(w * inp for w, inp in zip(weights, inputs))
    return int(total_input >= threshold)

# weights and threshold value
weights = [0.6, 0.7]
threshold = 0.5

# Test McCulloch-Pitts Neuron
inputs = [[0, 0], [0, 1], [1, 0], [1, 1]]
for input_set in inputs:
    output = mcculloch_pitts_neuron(input_set, weights, threshold)
    print(f"For inputs {input_set} output is {output}")
```

For inputs [0, 0] output is 0  
For inputs [0, 1] output is 1  
For inputs [1, 0] output is 1  
For inputs [1, 1] output is 1



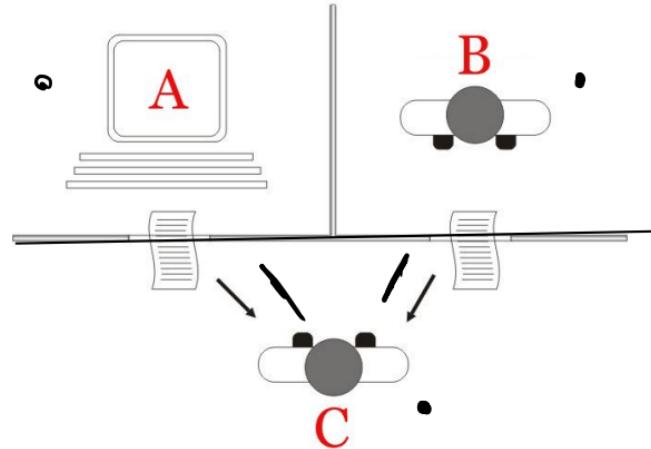
# What's Missing?

This type of mechanism is very useful for modelling a simple function but there is  
no learning at all.

# Can Machines Think?

Limitation Game  
aka

Turing Test



Alan M. Turing (1950)

# Dartmouth AI Project Proposal (1955)

“We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire.”  
The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.”

# Perceptron

## A bit of history...

The **Mark I Perceptron** machine was the first implementation of the perceptron algorithm.

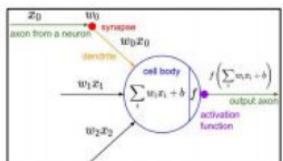
The machine was connected to a camera that used 20×20 cadmium sulfide photocells to produce a 400-pixel image.

recognized  
letters of the alphabet

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

update rule:

$$w_i(t+1) = w_i(t) + \alpha(d_j - y_j(t))x_{j,i},$$



Frank Rosenblatt, ~1957: Perceptron

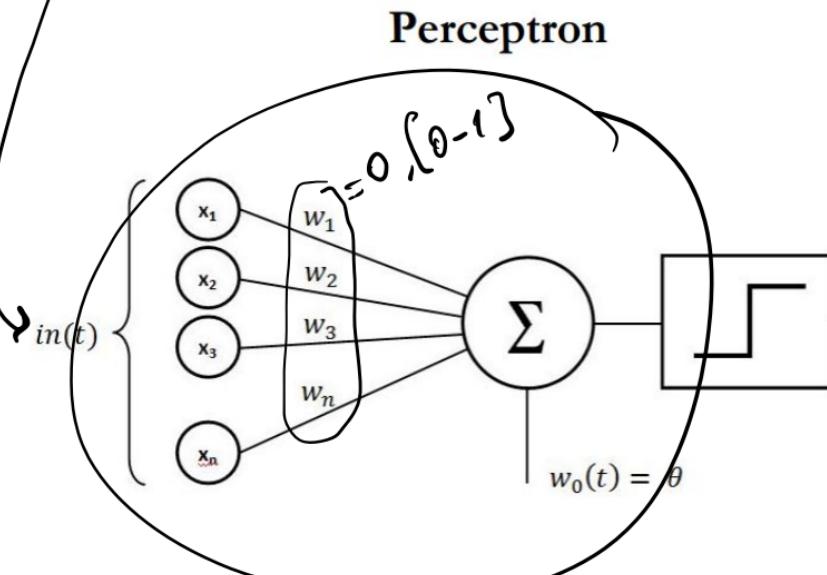


$[x_1, x_2, x_3]$

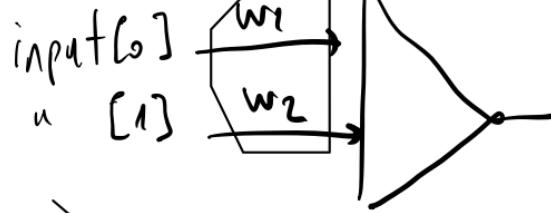


1

# Perceptron



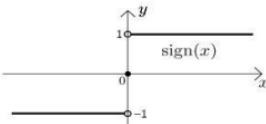
Frank Rosenblatt (1957)



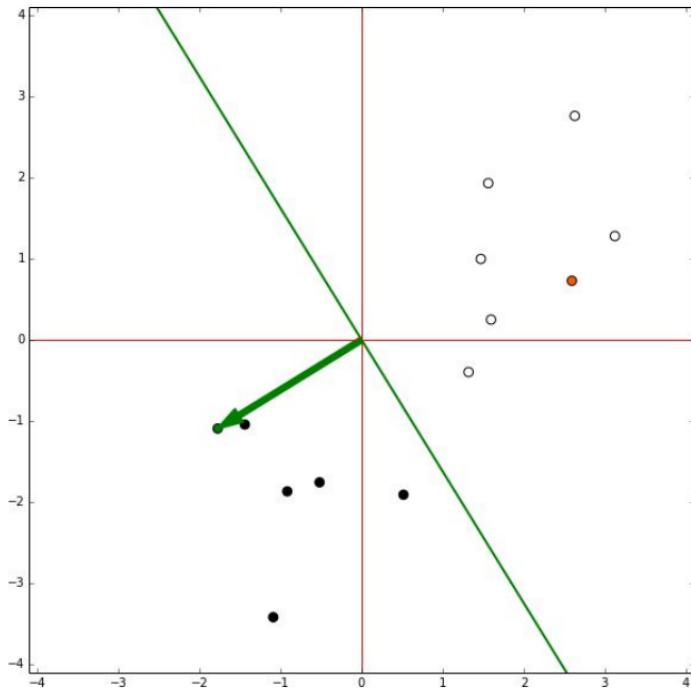
1//

$out(t)$

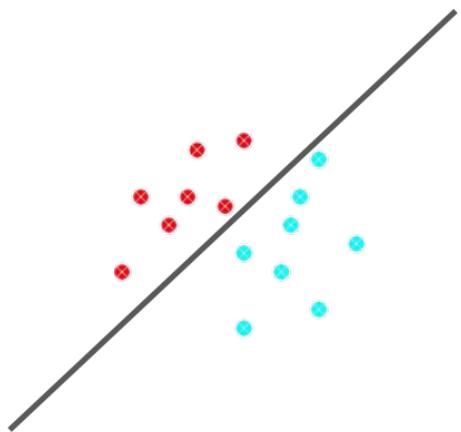
$$\text{sign}(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ -1, & x < 0. \end{cases}$$



# Perceptron



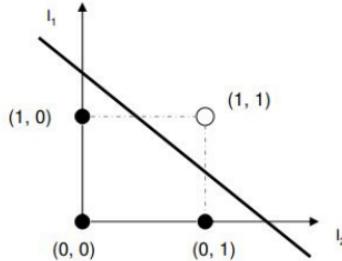
# Linear Classification



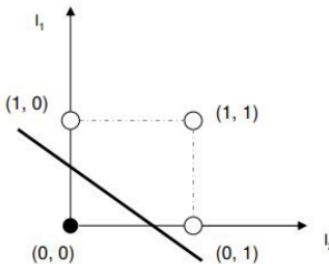
Perfect!

# XOR Case

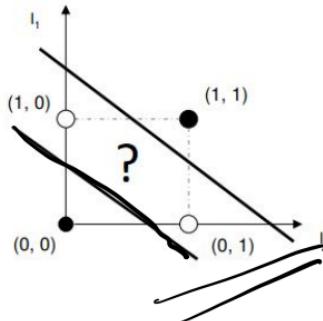
AND		
$I_1$	$I_2$	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
$I_1$	$I_2$	out
0	0	0
0	1	1
1	0	1
1	1	1



XOR		
$I_1$	$I_2$	out
0	0	0
0	1	1
1	0	1
1	1	0



# First AI Winter

- High Expectations and Disappointment ✓
- Technical Limitations ✓
- Lack of Data ✓
- Funding Cuts ✓



# History of AI

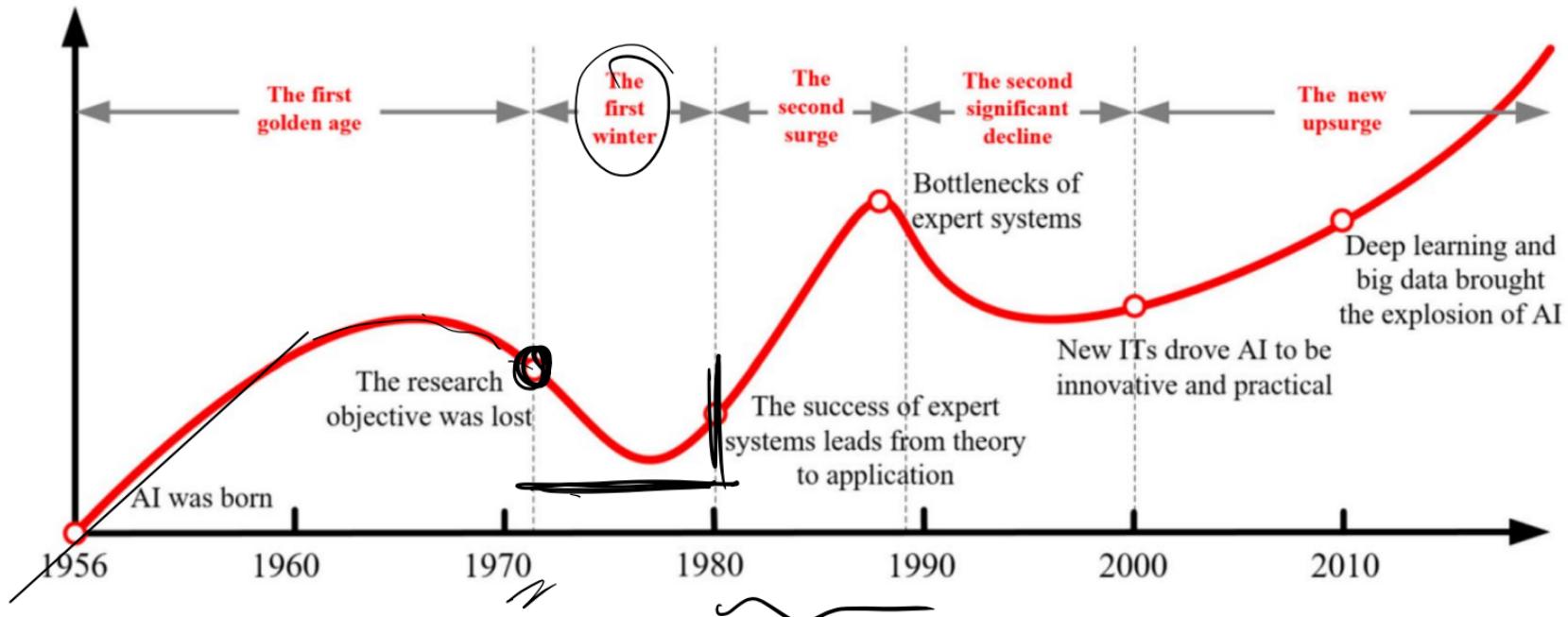


Figure: Artificial Intelligence in product lifecycle management

# Moving the First AI Winter





## Geoffrey Hinton

FOLLOW

GET MY OWN PROFILE

Emeritus Prof. Computer Science, [University of Toronto](#)

Verified email at cs.toronto.edu - [Homepage](#)

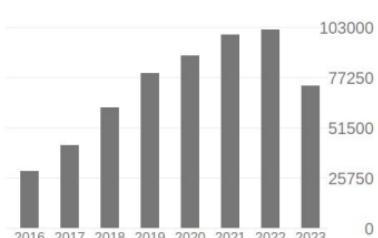
machine learning psychology artificial intelligence cognitive science computer science

Cited by

[VIEW ALL](#)

All Since 2018

Citations	710804	505382
h-index	180	132
i10-index	437	339



TITLE

CITED BY

YEAR

### Imagenet classification with deep convolutional neural networks

A Krizhevsky, I Sutskever, GE Hinton

Advances in neural information processing systems 25

142013 \*

2012

### Deep learning

Y LeCun, Y Bengio, G Hinton

Nature 521 (7553), 436-44

69824

2015

### Dropout: a simple way to prevent neural networks from overfitting

N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov

The journal of machine learning research 15 (1), 1929-1958

46158

2014

### Visualizing data using t-SNE

L van der Maaten, G Hinton

Journal of Machine Learning Research 9 (Nov), 2579-2605

37614

2008

### Learning representations by back-propagating errors

DE Rumelhart, GE Hinton, RJ Williams

Nature 323 (6088), 533-536

35122

1986



### Learning internal representations by error-propagation

DE Rumelhart, GE Hinton, RJ Williams

Parallel Distributed Processing: Explorations in the Microstructure of ...

32190

1986



Public access

[VIEW ALL](#)

1 article

[7 articles](#)

not available

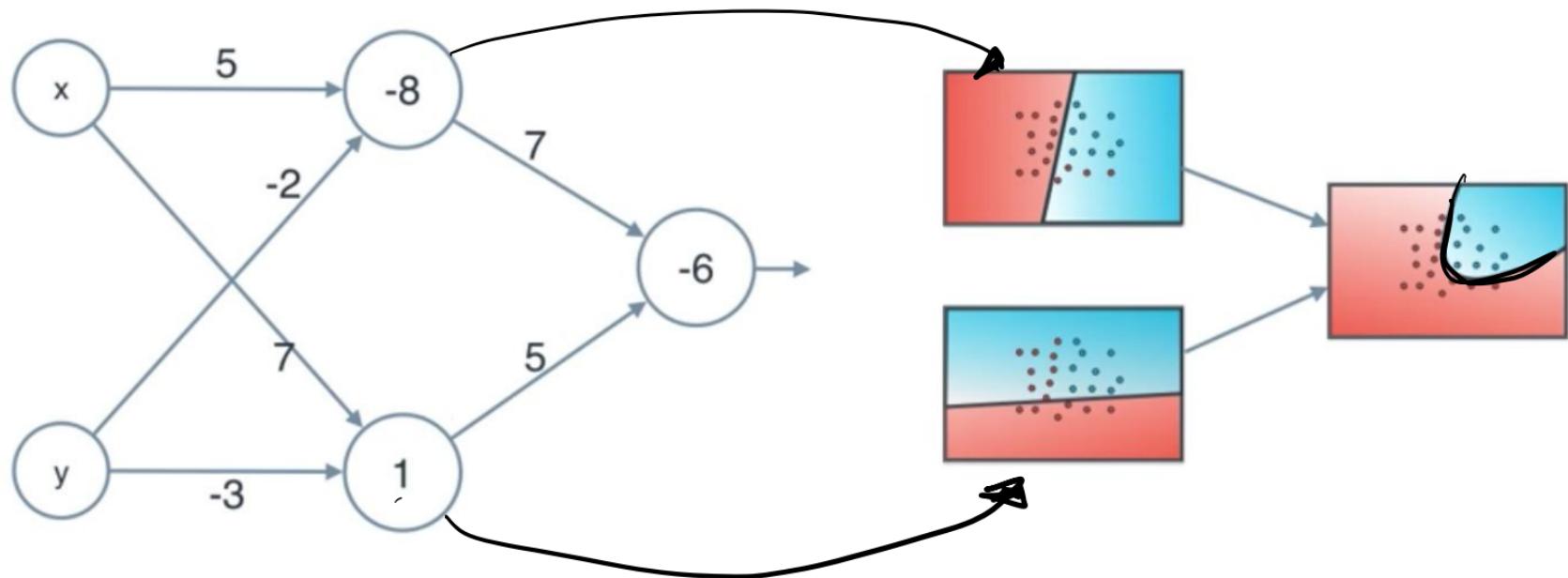
[available](#)

Based on funding mandates



Backpropagation Ag.

# Multilayer Perceptron



# Backpropagation Algorithm

LEARNING INTERNAL REPRESENTATIONS  
BY ERROR PROPAGATION

David E. Rumelhart, Geoffrey E. Hinton,  
and Ronald J. Williams

September 1985

ICS Report 8506



David E. Rumelhart  
Institute for Cognitive Science  
University of California, San Diego

Geoffrey E. Hinton  
Department of Computer Science  
Carnegie-Mellon University

Ronald J. Williams  
Institute for Cognitive Science  
University of California, San Diego

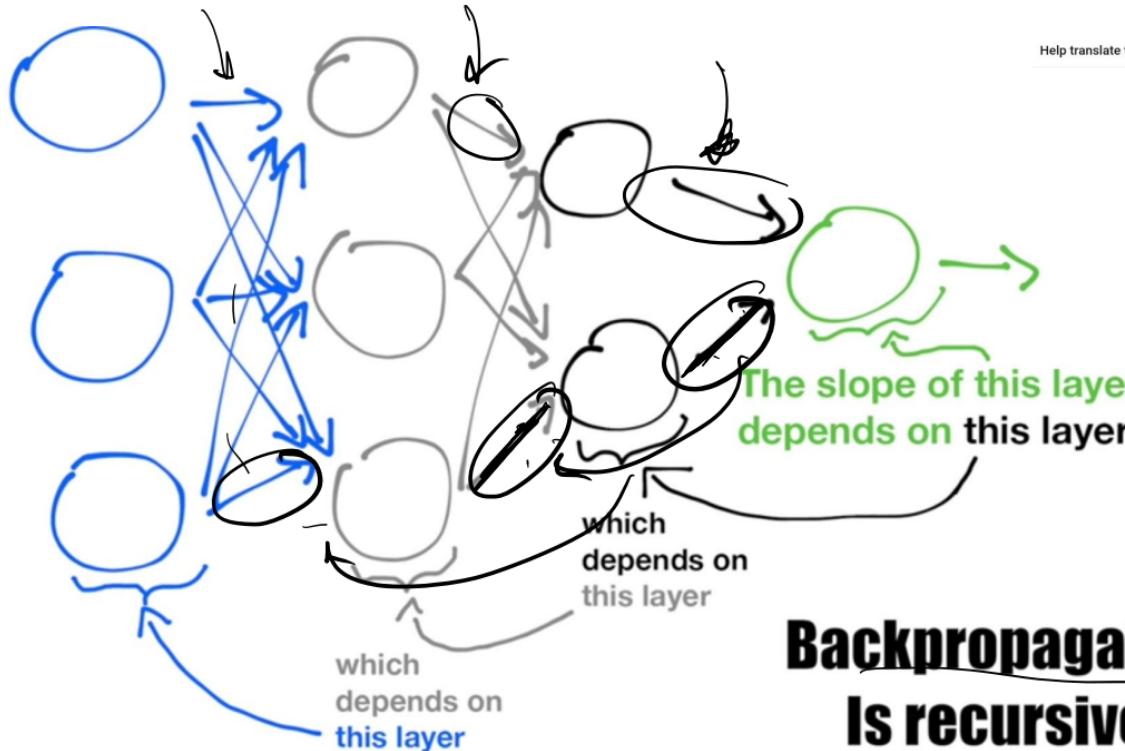
DISTRIBUTION STATEMENT A  
Approved for public release  
Distribution Unlimited

To be published in D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations*. Cambridge, MA: Bradford Books/MIT Press.

This research was supported by Contract N00014-85-K-0450, NR 667-548 with the Personnel and Training Research Program of the Office of Naval Research and by grants from the System Development Foundation. Requests for reprints should be sent to David E. Rumelhart, Institute for Cognitive Science, C-015; University of California, San Diego; La Jolla, CA 92093.  
Copyright © 1985 by David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams



# Backpropagation Algorithm



**Backpropagation**  
**Is recursive!**



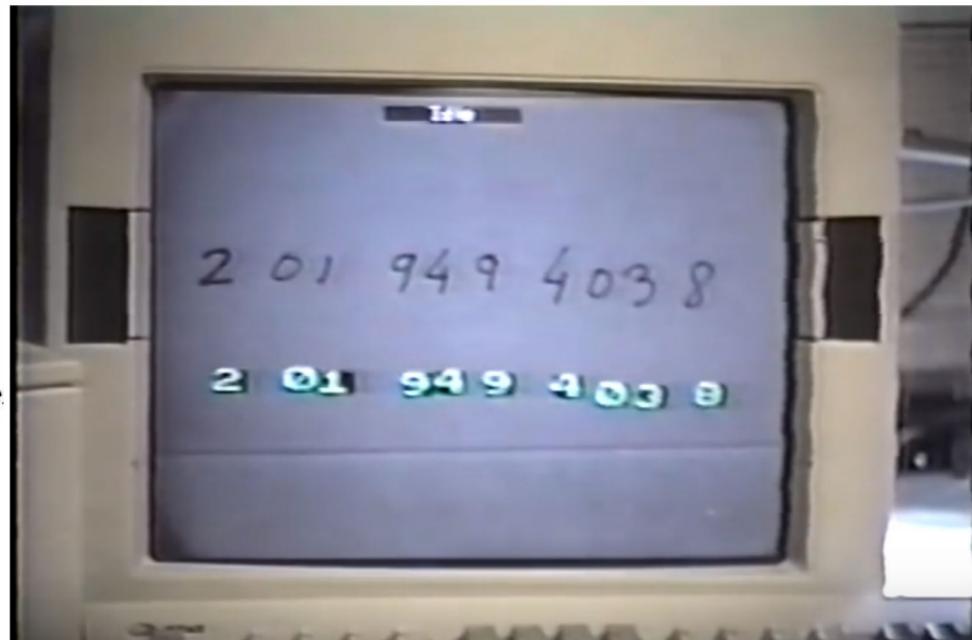
Slide Credit: Siraj Raval

# Real World Application

## Backpropagation Applied to Handwritten Zip Code Recognition

**Y. LeCun  
B. Boser  
J. S. Denker  
D. Henderson  
R. E. Howard  
W. Hubbard  
L. D. Jackel**

*AT&T Bell Laboratories Holmdel*



# History of AI

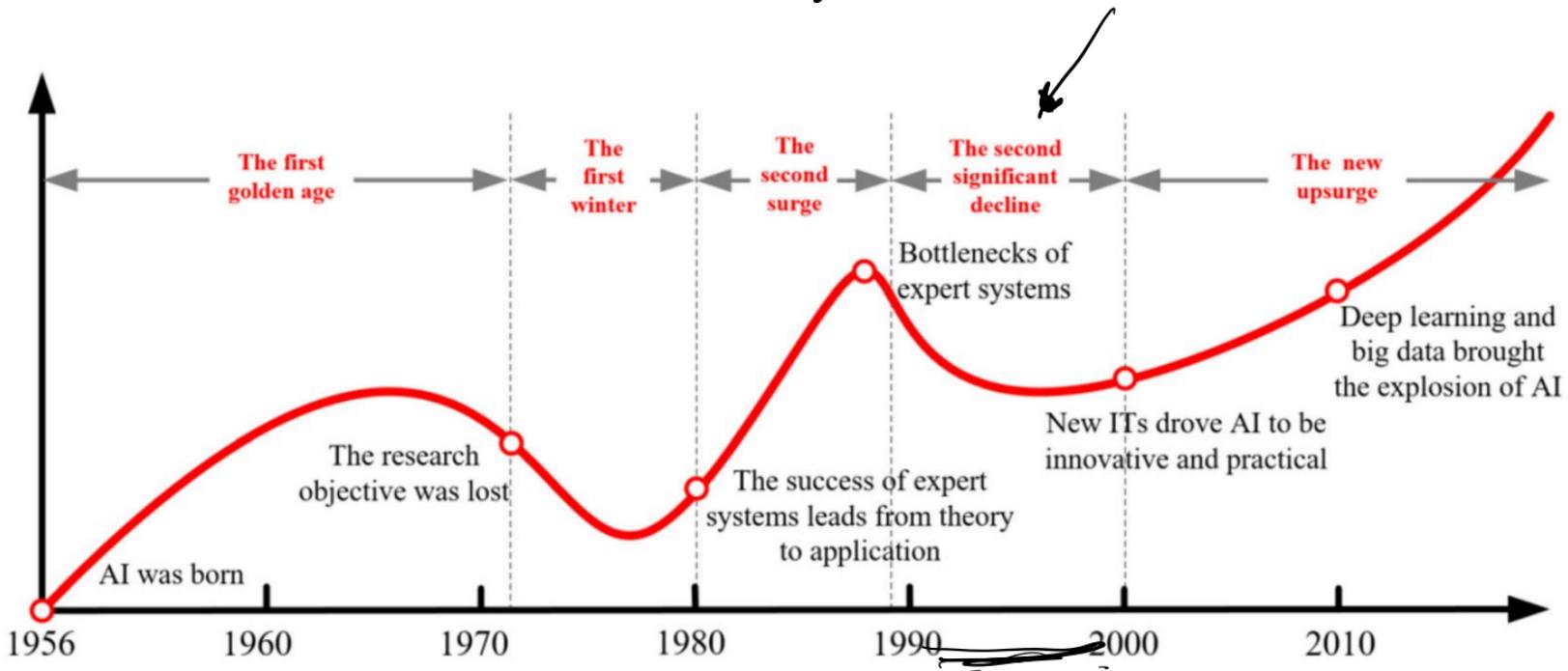


Figure: Artificial Intelligence in product lifecycle management

# Second AI Winter

- Overestimated Promises ✓
- Expert System Limitations ✓
- Lack of Sufficient Computing Power ✓
- Lack of Data ✓
- Funding Drought ✓
- Lack of Generalized Solutions ✓

1980s

```
graph LR; A[1980s] --> B{ }; B -- If --> C["- - -"]; B -- else --> D["- - -"]
```



# Moving the Second AI Winter





## Geoffrey Hinton

FOLLOW

GET MY OWN PROFILE

Emeritus Prof. Computer Science, [University of Toronto](#)

Verified email at cs.toronto.edu - [Homepage](#)

machine learning psychology artificial intelligence cognitive science computer science

TITLE

CITED BY

YEAR

### Imagenet classification with deep convolutional neural networks

A Krizhevsky, I Sutskever, GE Hinton

Advances in neural information processing systems 25

142013 \*

2012



### Deep learning

Y LeCun, Y Bengio, G Hinton

Nature 521 (7553), 436-44

69824

2015

### Dropout: a simple way to prevent neural networks from overfitting

N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov

The journal of machine learning research 15 (1), 1929-1958

46158

2014

### Visualizing data using t-SNE

L van der Maaten, G Hinton

Journal of Machine Learning Research 9 (Nov), 2579-2605

37614

2008



### Learning representations by back-propagating errors

DE Rumelhart, GE Hinton, RJ Williams

Nature 323 (6088), 533-536

35122

1986

### Learning internal representations by error-propagation

DE Rumelhart, GE Hinton, RJ Williams

Parallel Distributed Processing: Explorations in the Microstructure of ...

32190

1986

Cited by

[VIEW ALL](#)

All

Since 2018

Citations

710804

505382

h-index

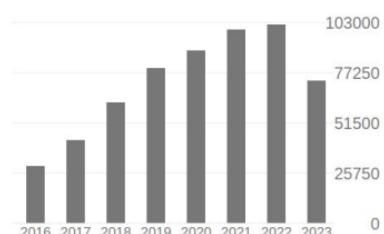
180

132

i10-index

437

339



Public access

[VIEW ALL](#)

1 article

7 articles

not available

available

Based on funding mandates



# ImageNet

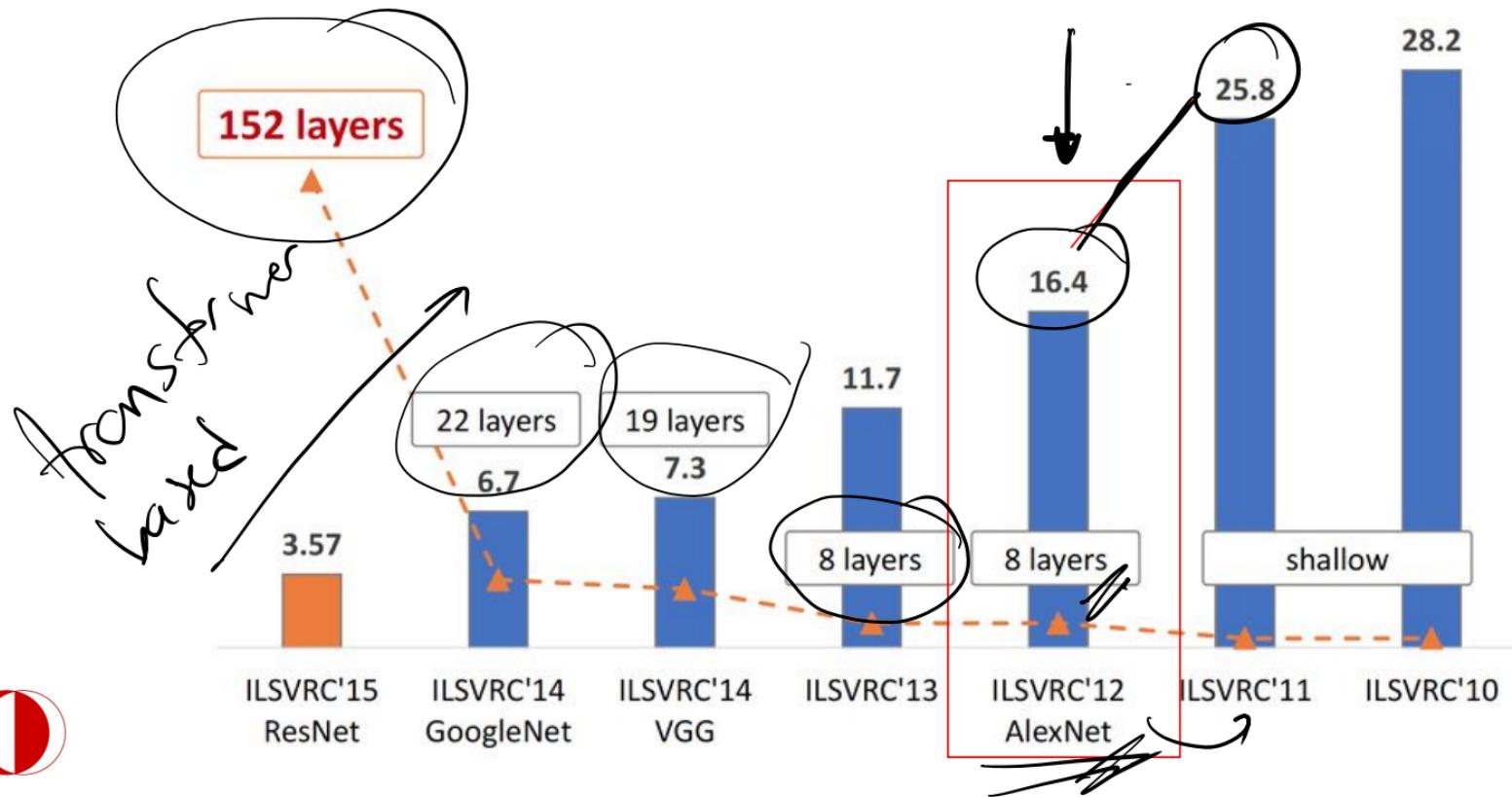
## ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



# ImageNet

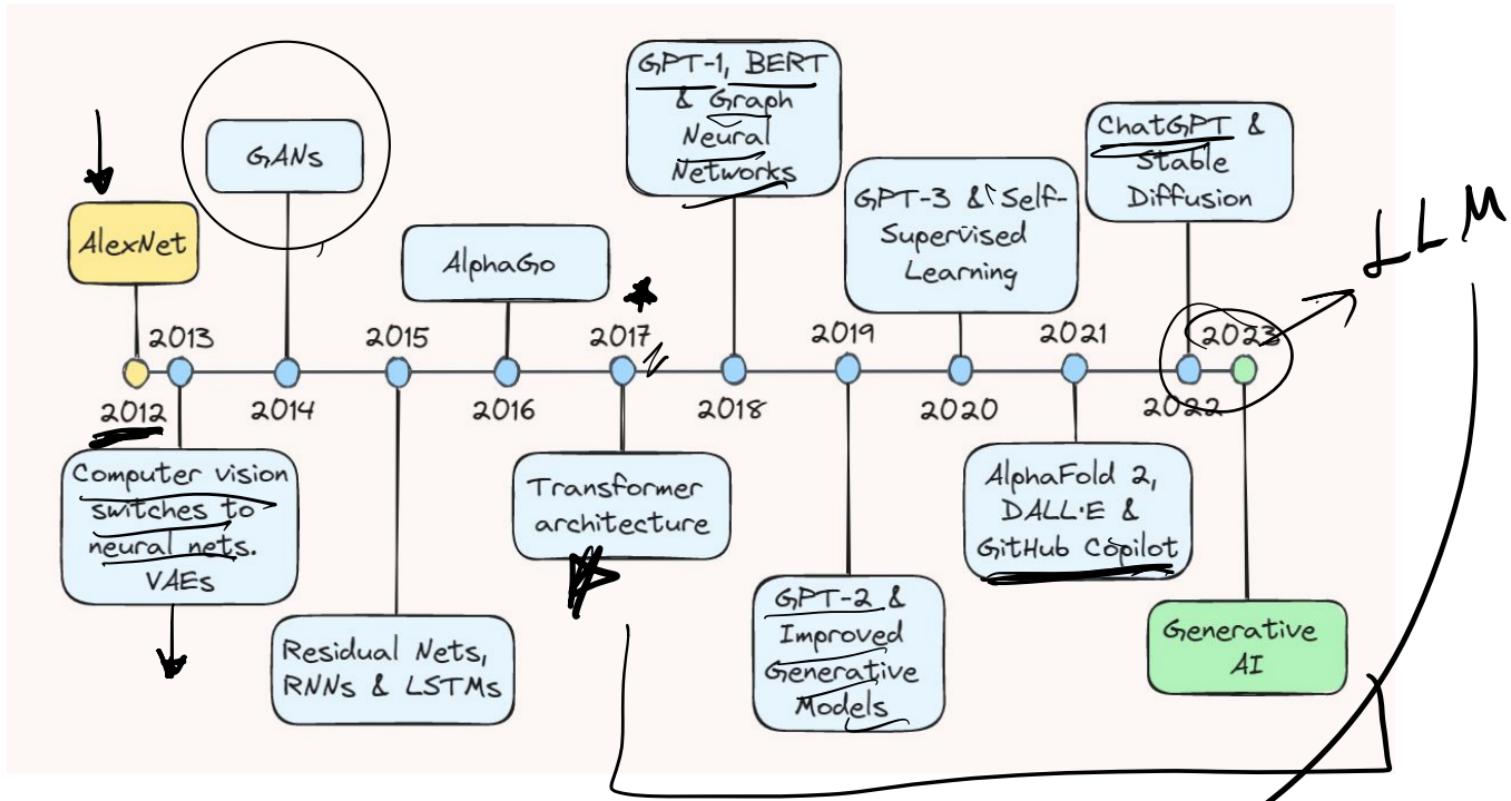


# Moving from Second AI Winter

- **Advances in Neural Networks**
- **Rise of Computational Power**
- **Availability of Big Data**
- **IBM Deep Blue's Victory**
- **Open Source Movement**
- **Establishment of AI Companies**

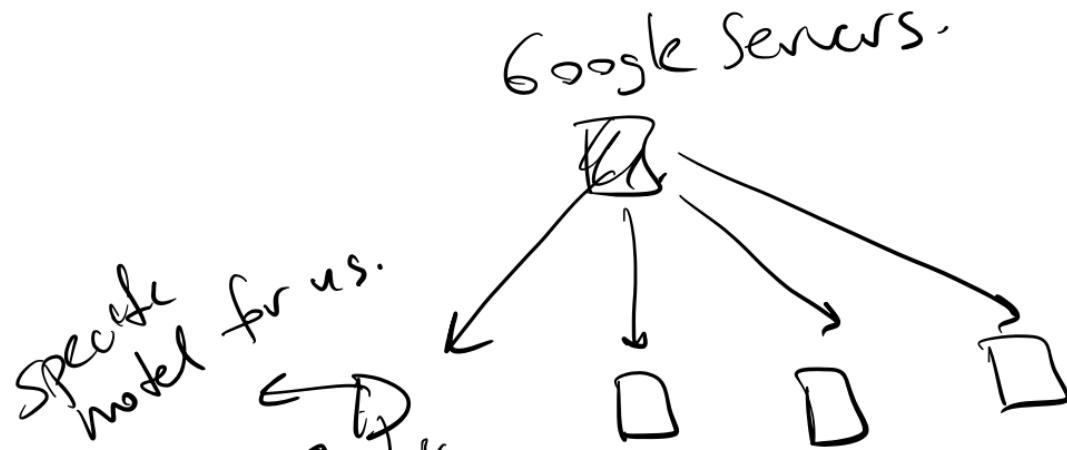


# History of AI - 2



Large Language Models

## Applications



# Text Generation

*rewards*

Gmail display language: English

Change language settings for this account

Show all language options

ers

Smart Compose

Gmail gives you writing suggestions as you type. You can turn them off in [Settings](#).

**DISMISS**

Ask before displaying ext

Send cancellation period: 5

Compose:

Writing suggestions on

suggestions appear as  Writing suggestions off

(mail)

behavior

Reply

Smart Compose

Marie Brewis (idg.co.uk)

Smart Compose

Hello. This is a test to see whether Smart Compose actually makes any useful suggestions [tab](#)

# Spam Detection

A screenshot of a Gmail inbox in a web browser window. The browser's top bar shows the tabs: Safari, File, Edit, View, History, Bookmarks, Develop, Window, Help, and several icons. The address bar displays the URL: mail.google.com/mail/u/0/#spam. The main interface shows the Google logo at the top left, followed by a search bar and a navigation bar with icons for Mail, Compose, and settings. A red box highlights the 'Compose' button. On the left, a sidebar lists 'Sent Mail', 'Drafts', 'All Mail', 'Spam (35)', and 'Trash'. Below that is a 'Categories' section with '[Imap]/Archive', '[Imap]/Drafts', and '[Imap]/Outbox'. The main content area shows a list of 38 spam messages. A blue box highlights the first message from 'Australian Marketing Lis.' with the subject 'Let's make 2018 great - Hello'. Above the list, a message bar reads: 'Delete all spam messages now (messages that have been in Spam more than 30 days will be automatically deleted)'. The list continues with other messages from 'Inkspot', 'Diana', 'Svetlana', 'Oksana', and 'Kseniya', each with their respective subjects and dates.

Date	Subject
Jan 23	Let's make 2018 great - Hello
Jan 20	(no subject) - http://honour.pa
Jan 18	Two for the price of one - Ia
Jan 18	hi - You seem like my type an
Jan 17	hi - You seem like my type an
Jan 17	hi - You seem like my type an
Jan 17	hi - You seem like my type an



# Recommendation Systems ↗

9781787125933 978-1787125933 2nd Packt Publishing September 20, 2017 English

Frequently bought together

+ +

Total price: \$130.14 Add all three to Cart

This item: Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn... by Sebastian Raschka Paperback \$41.99  
 Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron Paperback \$56.16  
 The Hundred-Page Machine Learning Book by Andriy Burkov Paperback \$31.99

Add to List Share [Email](#) [Facebook](#) [Twitter](#) [Pinterest](#) <Embed>

Have one to sell? Sell on Amazon

amazon book clubs early access Add to book club Not in a club? Learn more

## Products related to this item

Page 1 of 35

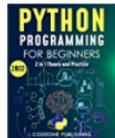
Sponsored @



Python Machine Learning: Machine Learning and Deep Learning with Python,...  
Sebastian Raschka

Newly updated for TensorFlow 2.0, this widely acclaimed book is a reference you'll keep coming back to as you build your machine learning systems

★★★★★ 384 Paperback \$49.99 prime



Python Programming for Beginners: The #1 Python Programming Crash Course for...  
Codeon Publishing

#1 Best Seller Paperback \$19.45 prime



Python Programming for Beginners: The Ultimate Crash Course to Learn Python in 7 Da...  
Andrew Park

#1 Best Seller Paperback \$13.95 prime



Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep...  
Sebastian Raschka

This book is a comprehensive guide to machine and deep learning using PyTorch's simple to code framework Paperback \$46.79 prime



Clean Code in Python: Develop maintainable and efficient code, 2nd Edition  
Mariano Anaya

Discover how to apply industry-approved coding practices to design clean, sustainable, and readable real-world Python code Paperback \$44.64 prime



Hands-On Data Science for Marketing: Improve your marketing strategies with...  
Yoon Hyup Hwang

★★★★★ 63 Paperback \$44.99 prime

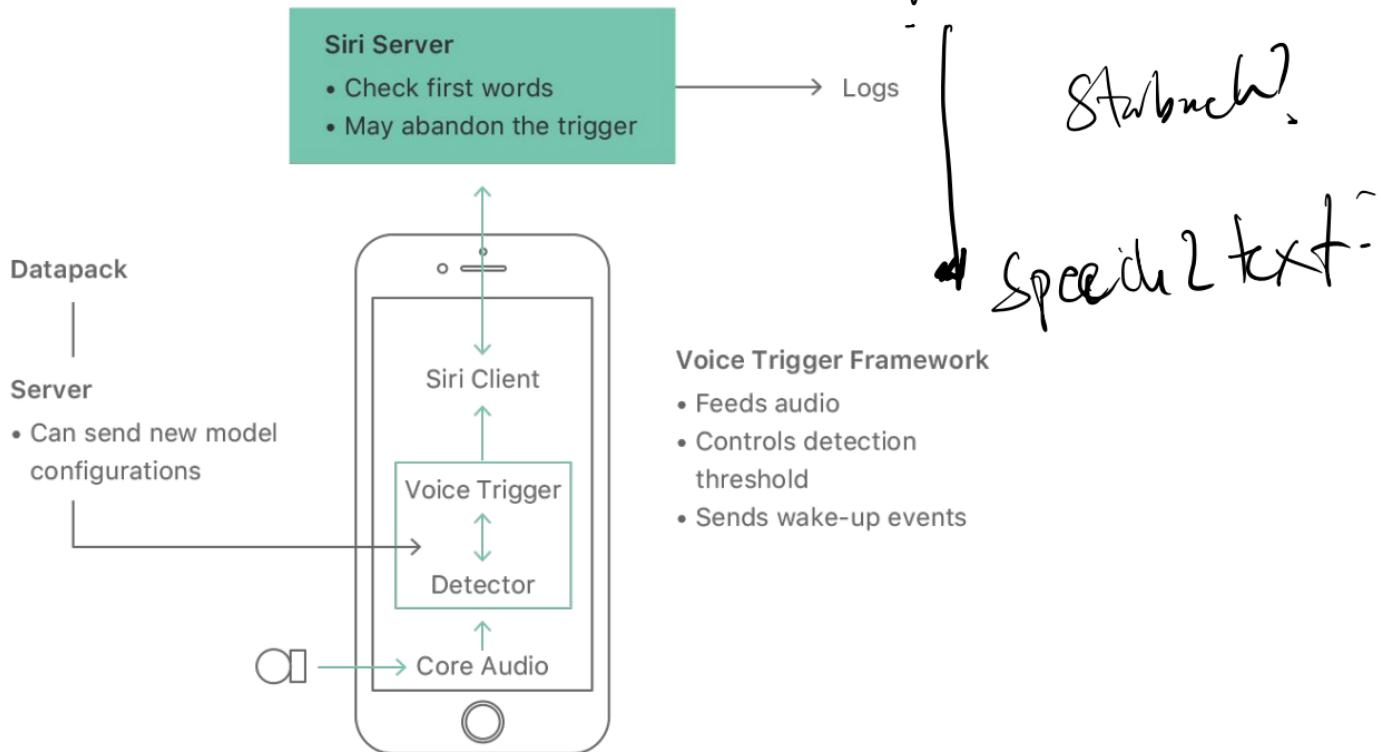


Python Automation Cookbook: 75 Python automation ideas for web scraping, data...  
Jaime Butta

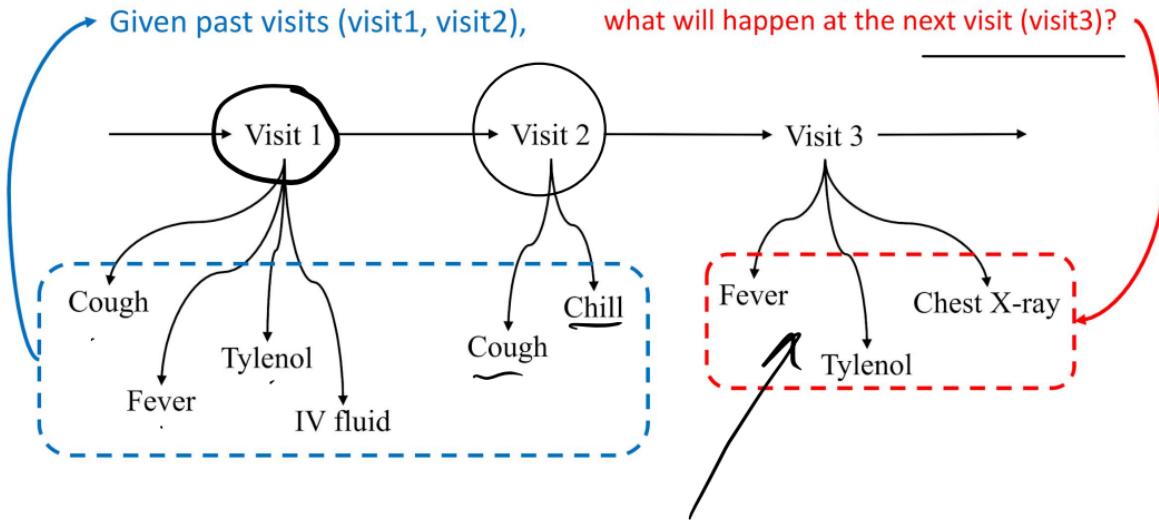
Get a firm grip on core processes including browser automation, web scraping, Word, Excel, and GUI automation with Python 3.8 and higher  
★★★★★ 78 Paperback \$39.99 prime



# Speech Recognition



# AI in Healthcare



# AI in Healthcare

## Unstructured Imaging Data: Data



Computer Science

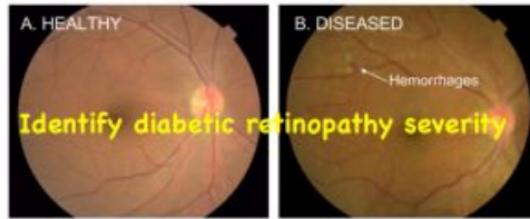
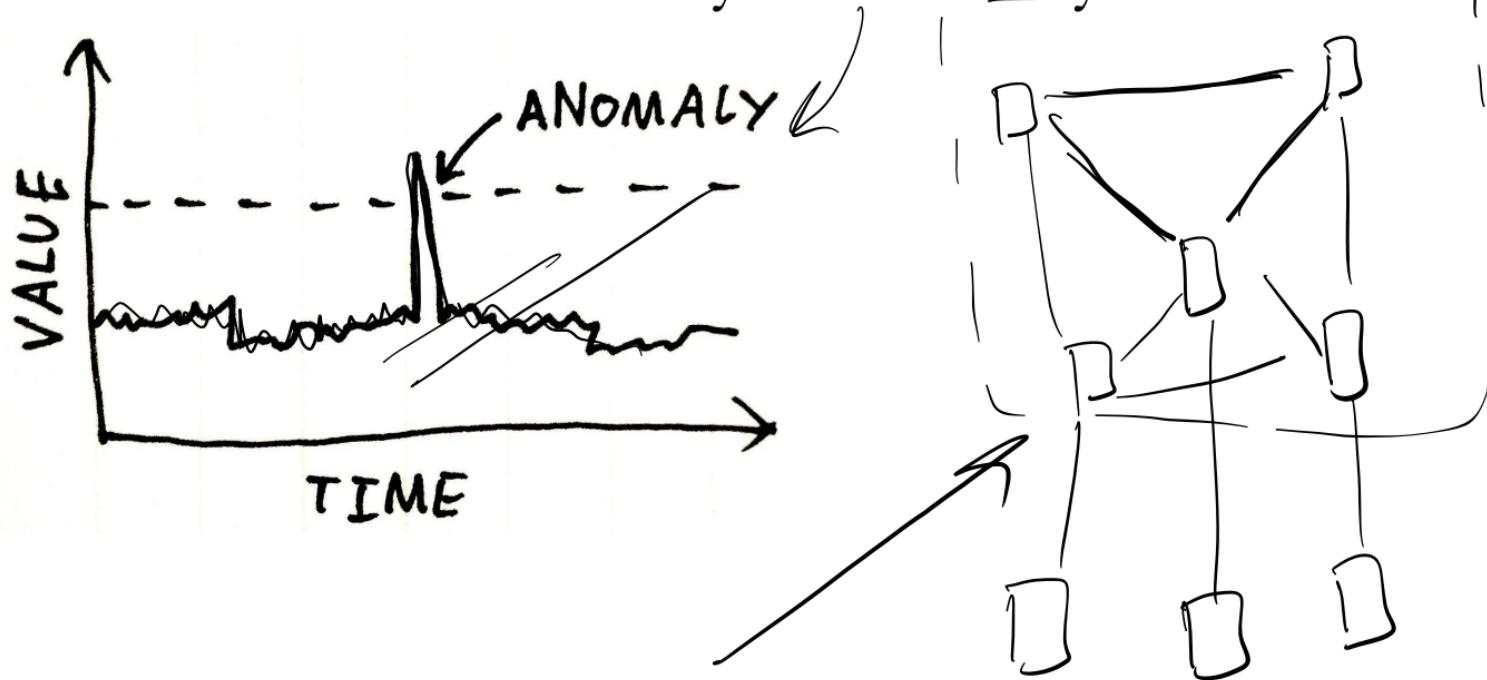


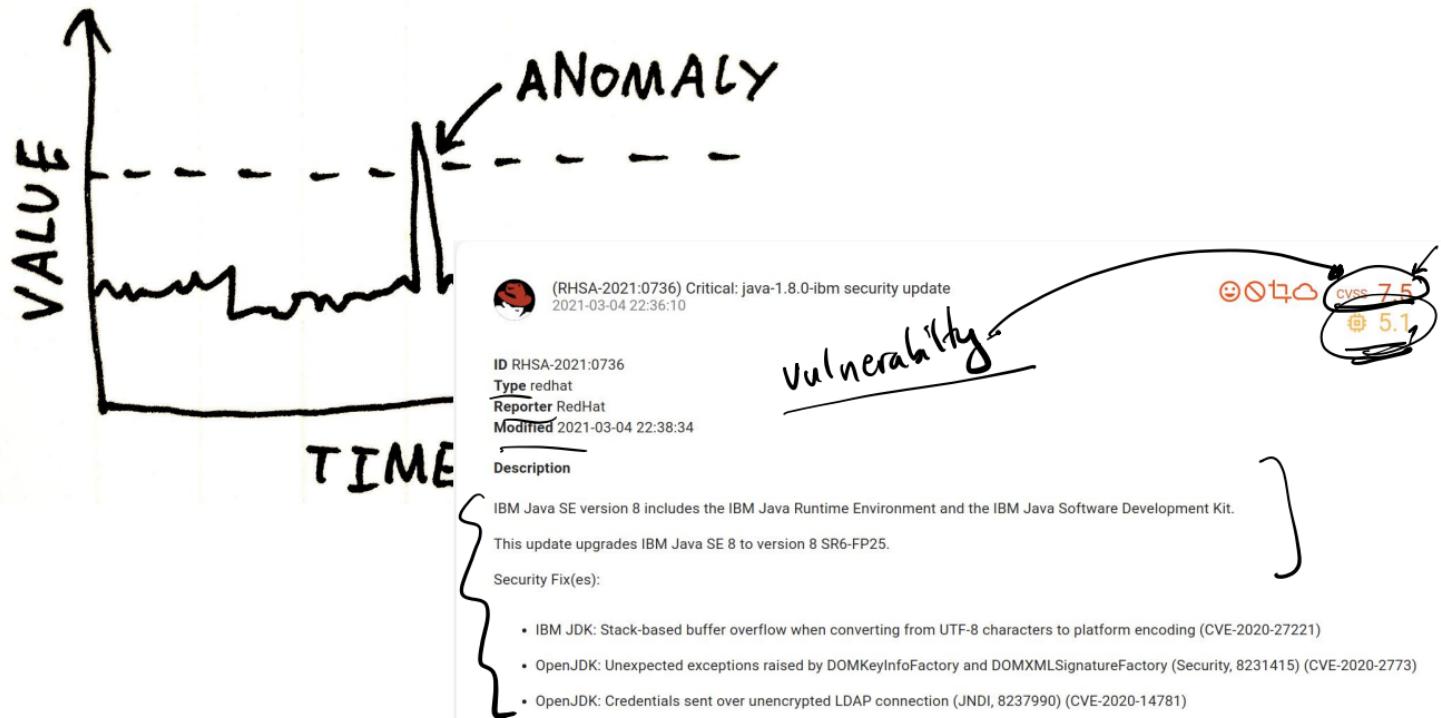
Figure 1. Examples of retinal fundus photograph that are taken to screen for DR. The image on the left is a healthy retina (A), whereas the image on the right is a retina with referable diabetic retinopathy (B) due to a number of hemorrhages (red spots) present.



# AI in Cybersecurity



# AI in Cybersecurity

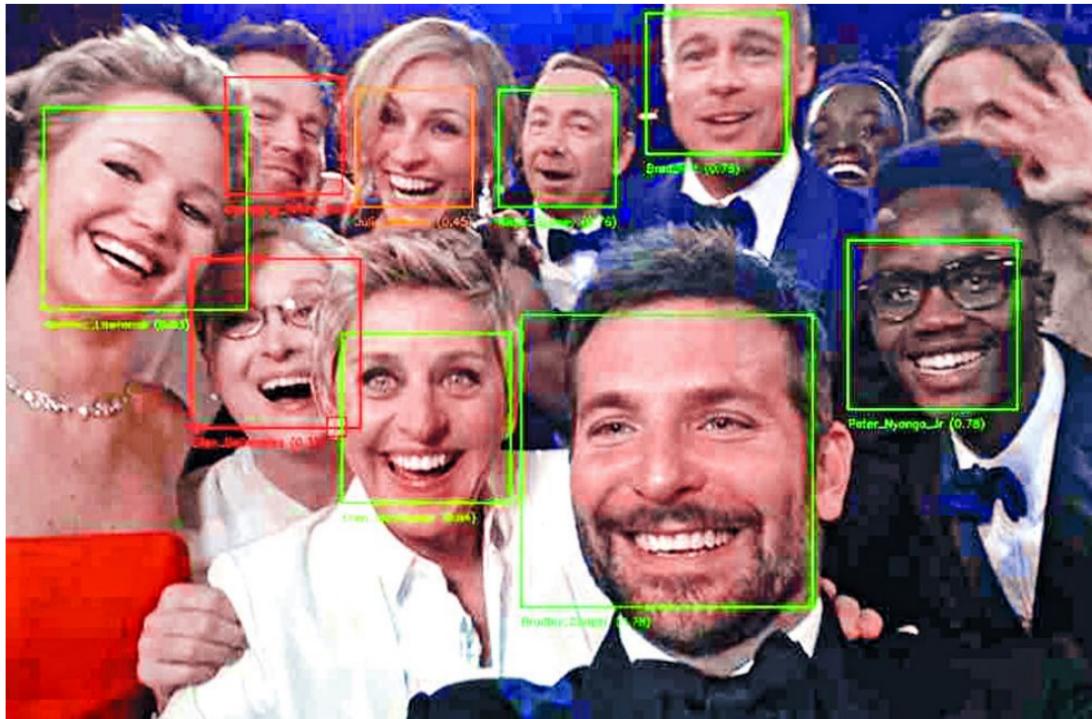


# Natural Language Processing - Named Entity Recognition ~~(NER)~~

estigation after his disparaging texts about President Trump PERSON were uncovered, was fired. Credit T.J. Kirkpatrick PERSON for The New York TimesBy Adam Goldman ORG and Michael S. Schmidt Aug PERSON . 13 CARDINAL , 2018 WASHINGTON CARDINAL — Peter Strzok PERSON , the F.B.I. GPE senior counterintelligence agent who disparaged President Trump PERSON in inflammatory text messages and helped oversee the Hillary Clinton PERSON email and Russia GPE investigations, has been fired for violating bureau policies, Mr. Strzok PERSON said Monday DATE . Mr. Trump and his allies seized on the texts — exchanged during the 2016 DATE campaign with a former F.B.I. GPE Lisa Page — in PERSON assailing the Russia GPE investigation as an illegitimate "witch hunt." Mr. Strzok PERSON , who rose over 20 years ago at the F.B.I. GPE to become one of its most experienced counterintelligence agents, was a key figure in the early months DATE of the inquiry. Along with writing the texts, Mr. Strzok PERSON was accused of sending a highly sensitive search warrant to his personal email account. The

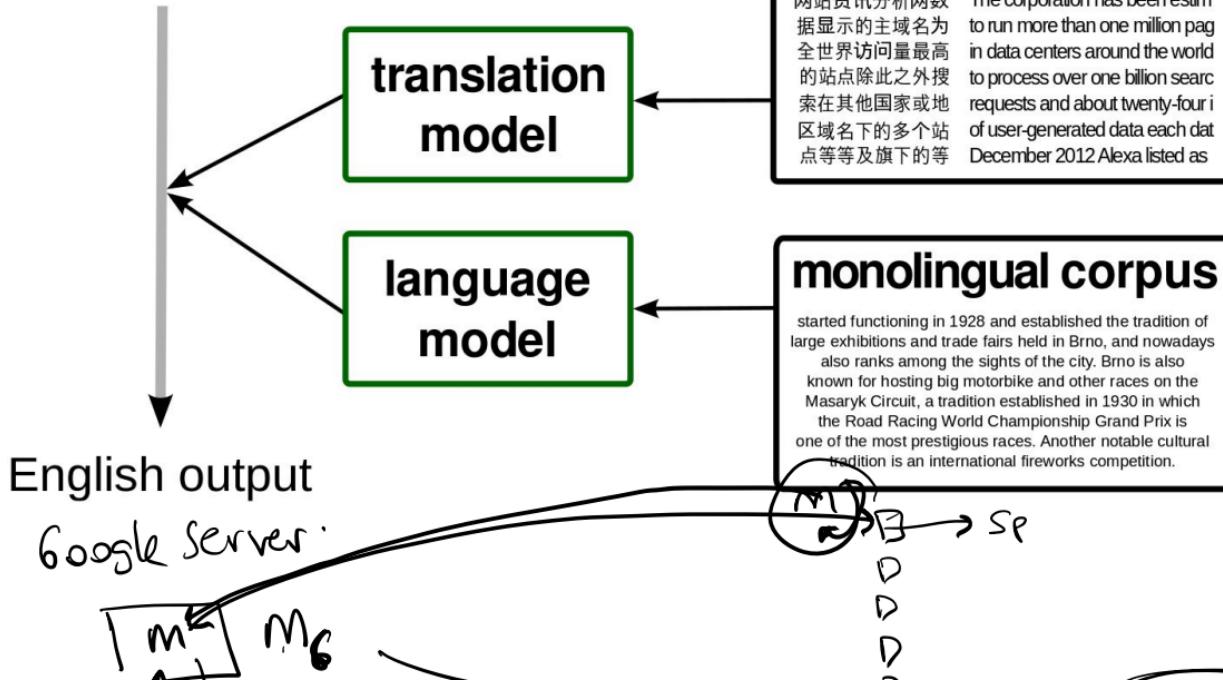


# Face Recognition

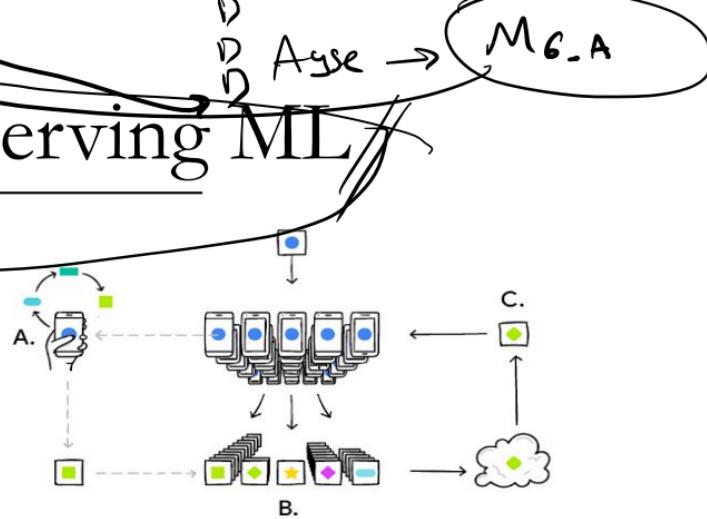


# Machine Translation

似乎格式有问题



# Privacy Preserving ML



**TensorFlow Privacy**

This repository contains the source code for TensorFlow Privacy, a Python library that includes implementations of TensorFlow optimizers for training machine learning models with differential privacy. The library comes with tutorials and analysis tools for computing the privacy guarantees provided.

The TensorFlow Privacy library is under continual development, always welcoming contributions. In particular, we always welcome help towards resolving the issues currently open.

**Latest Updates**

2020-12-21: A new vectorized version of the TF 2 optimizer is available, which can deliver much faster performance. We recommend trying it first, and to fall back to using the original non-vectorized version only if this fails. We are thankful to the authors of this paper for spurring this change.

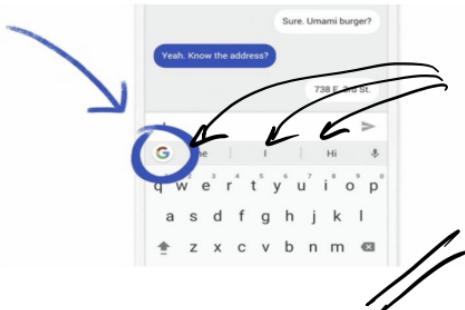
**Setting up TensorFlow Privacy**

**Dependencies**

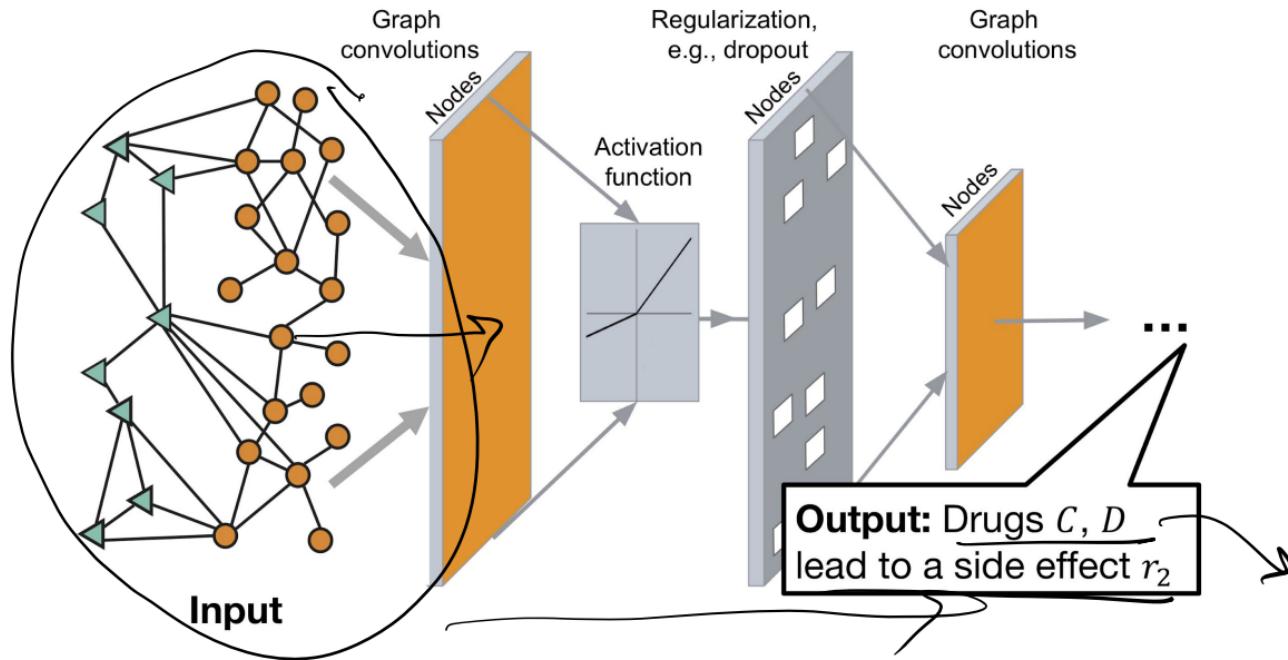
This library uses TensorFlow to define machine learning models. Therefore, installing TensorFlow ( $\geq 1.14$ ) is a pre-requisite. You can find instructions [here](#). For better performance, it is also recommended to install TensorFlow with GPU support (detailed instructions on how to do this are available in the TensorFlow installation documentation).

In addition to TensorFlow and its dependencies, other prerequisites are:

We're currently testing Federated Learning in [Gboard on Android](#), the Google Keyboard. When Gboard shows a suggested query, your phone locally stores information about the current context and whether you clicked the suggestion. Federated Learning processes that history on-device to suggest improvements to the next iteration of Gboard's query suggestion model.



# Graph ML

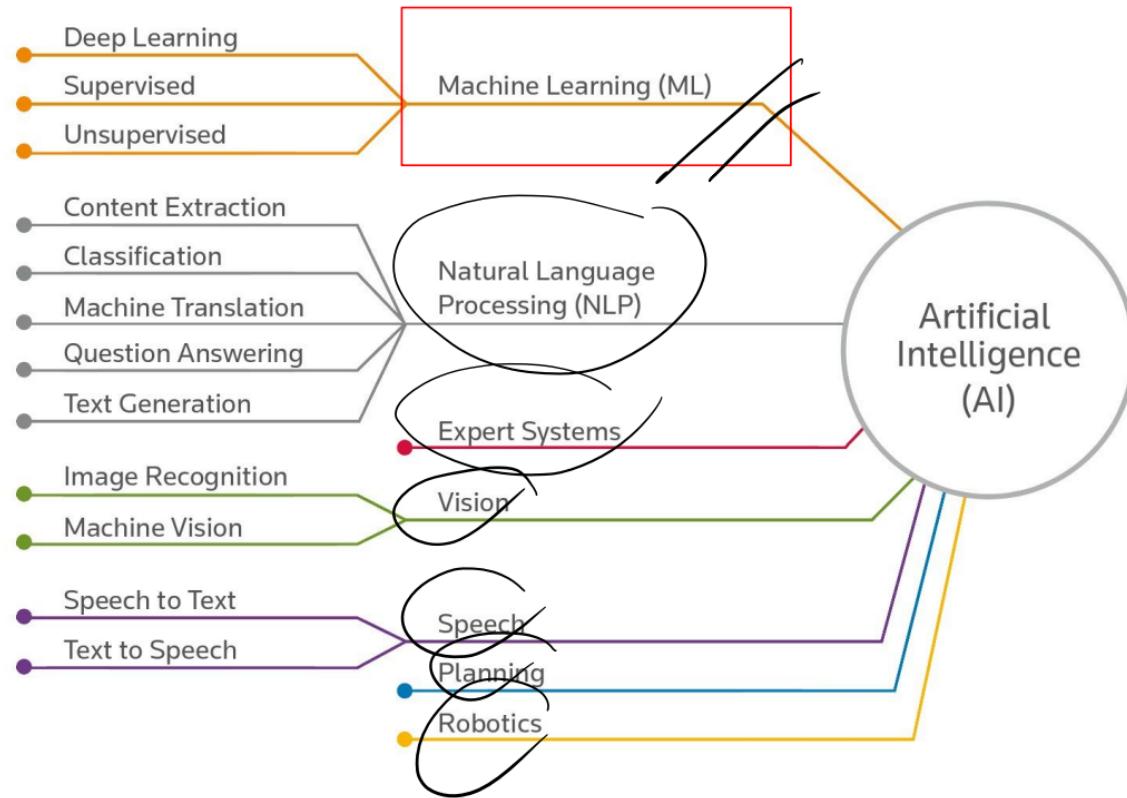


# Many Other Applications

- Finance
- E-commerce
- Information Extraction
- Social Networks
- Web Search
- Computer Vision and robotics
- Computational Biology
- Fraud Detection
- Etc.



# What is Artificial Intelligence (AI) ?



# How Do We Learn?



By seeing examples, many many of them



# How Do We Learn?



Abstraction

---



# What is Machine Learning?

- Learning is any process by which a system improves performance from experience”
  - Herbert Simon
- Another definition is done by Tom Mitchell
  - Machine Learning is the study of the algorithms that
    - improve their performance  $P$  //
    - at some task  $T$  //
    - with experience  $E$  //
  - A well-defined learning task is given by  $\langle P, T, E \rangle$

T: Recognizing hand-written words

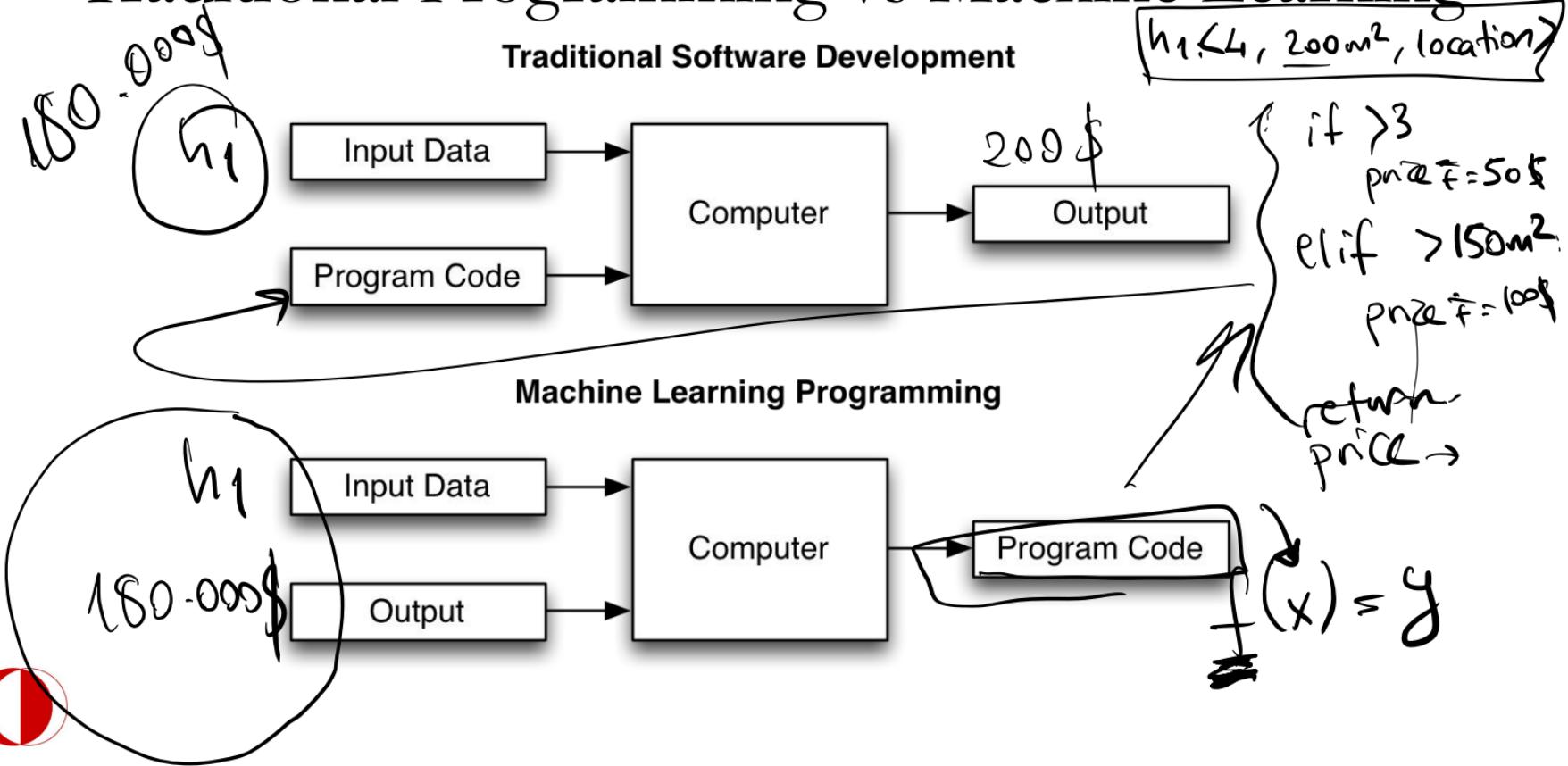
P: Percentage of words correctly classified

E: Database of human-labeled images of handwritten words

*Borkhan Y.30*



# Traditional Programming vs Machine Learning



# Machine Learning in a Nutshell

- Every machine learning algorithm consists of the following basic steps:
  - Data collection
  - Representation
  - Modeling
  - Evaluation
  - Optimization



# Rules of Machine Learning

- <https://developers.google.com/machine-learning/guides/rules-of-ml>
- 

## Before Machine Learning

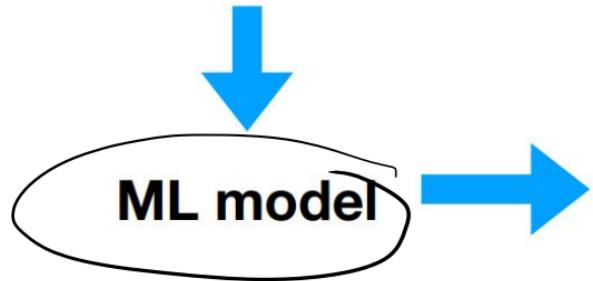
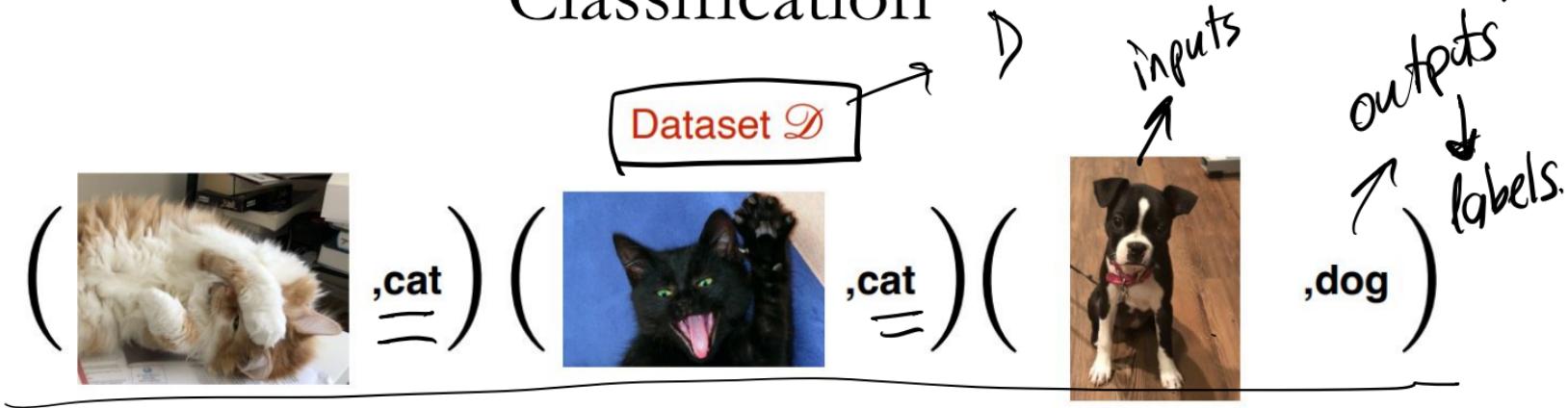
### Rule #1: Don't be afraid to launch a product without machine learning.

Machine learning is cool, but it requires data. Theoretically, you can take data from a different problem and then tweak the model for a new product, but this will likely underperform basic **heuristics**. If you think that machine learning will give you a 100% boost, then a heuristic will get you 50% of the way there.

For instance, if you are ranking apps in an app marketplace, you could use the install rate or number of installs as heuristics. If you are detecting spam, filter out publishers that have sent spam before. Don't be afraid to use human editing either. If you need to rank contacts, rank the most recently used highest (or even rank alphabetically). If machine learning is not absolutely required for your product, don't use it until you have data.

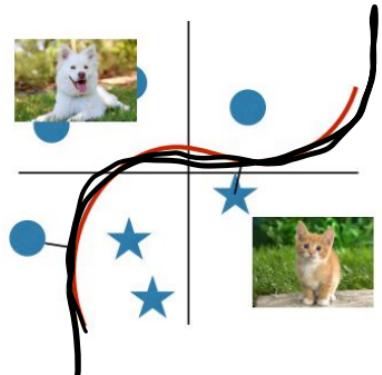


# Classification



$$h_1 = [4] \rightarrow 1-d.$$

$$h_1 = [4, 200m^2] \rightarrow 2^d.$$



Slide Credit: Cornell CS4780

input  
 $\uparrow$   
 $x_1, y_1$

output / label

$$h_1 = [ \dots ] \rightarrow \mathbb{R}^C$$

# Mathematical formulation of the pipeline

$$\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}, x_i \in \mathbb{R}^d, y_i \in \mathcal{C}$$

sample / example.

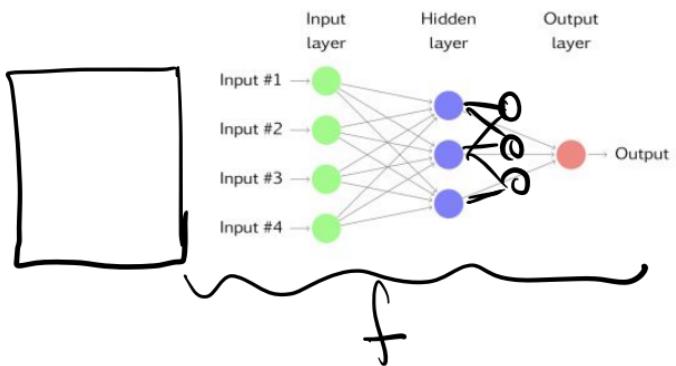
Dataset:

$$\begin{matrix} -1, 0 \\ 1, 1 \end{matrix} \quad [-1, 1]$$

$$=$$

$$(x_i, y_i) \sim \mathcal{P}$$

binary classification.



Hypothesis:

$$h : \mathbb{R}^d \rightarrow \mathcal{C}$$

function

Hypothesis class

i.e., a neural network-based classifier that maps image to label of cat or dog

$$\begin{matrix} 0 \\ 1 \end{matrix}$$

i.e., a large family of NNs with different parameters

$$\mathcal{H} = \{h\}$$

$$h_1, h_2, \dots$$

$$H = \{ \dots \}$$

values



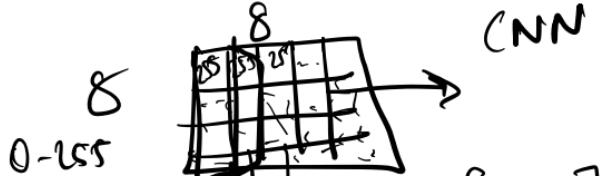
- Binary Classification:  $Y = \{0, 1\}$ ,  $Y = \{-1, 1\}$
  - Multi-Class Classification:  $Y = \{1, 2, 3, \dots, K\}$
  - Regression:  $Y = \text{Real Numbers}$
- Type of Y
- 

Numeric Values

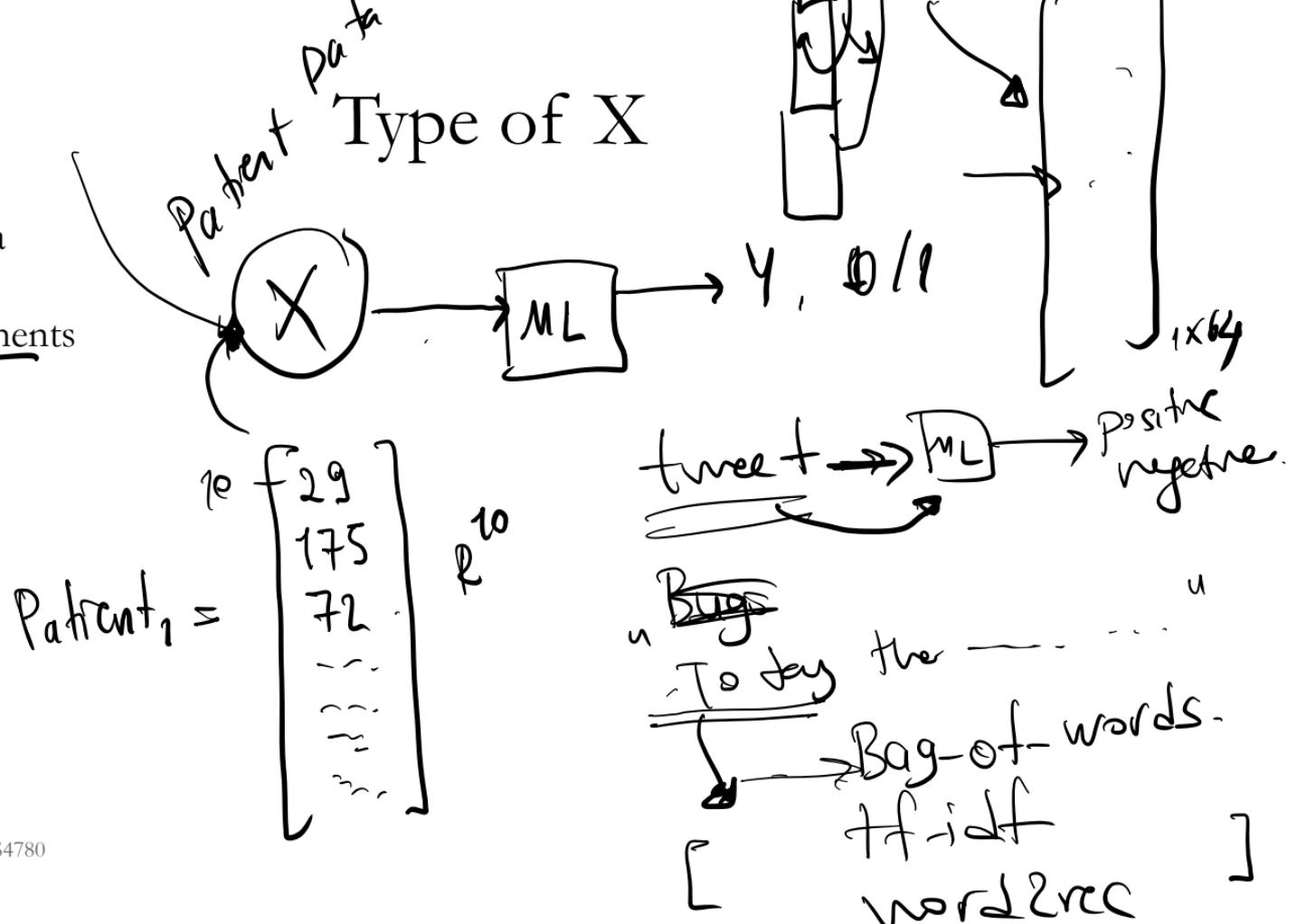
house price -

$X \in \mathbb{R}^d$

$Y \rightarrow \text{cont. values}$



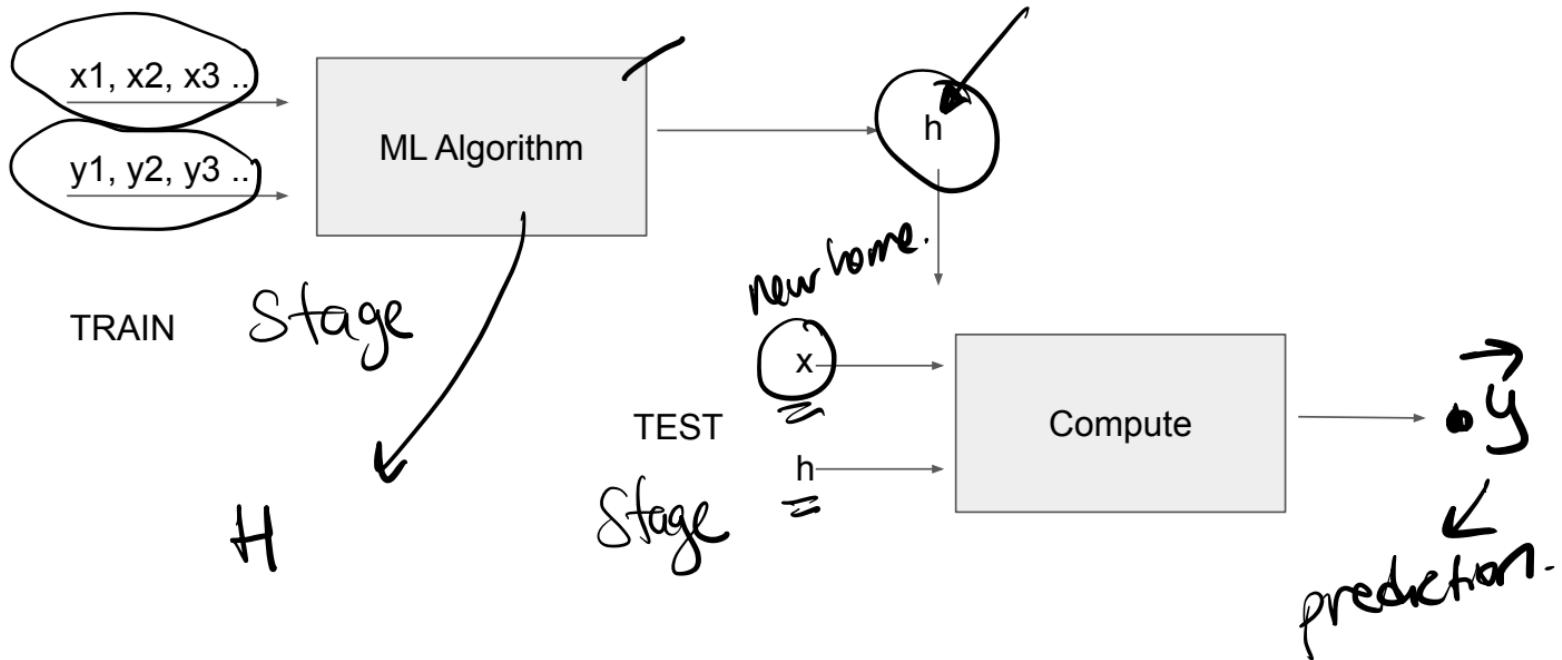
- Patient Data
- Text Documents
- Images



1000 house example.  
1000 decision  $\{0, 1\}$

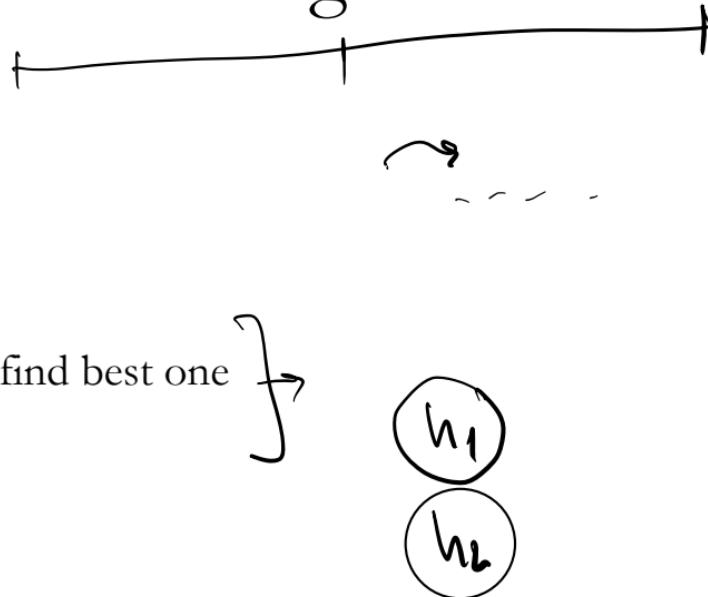
## Training

$1001 =$



# Learning

- Algorithm 1:
  - Pick  $h \in H$  randomly
  - What is the problem?
- Algorithm 2: Bad
  - Try every single  $h$  in  $H$  to find best one
  - What is the problem?



$$h(x_1) = \hat{y}_1$$

*y-prediction*

# The Loss Function

## The Loss Function

Q: how to select the best hypothesis  $\hat{h}$  from  $\mathcal{H}$ ?

Let's define loss function  $\ell : \mathcal{H} \times \mathbb{R}^d \times \mathcal{C} \mapsto \mathbb{R}$

Intuitively,  $\ell(h, x, y)$  tells us how bad (e.g., classification mistake) the hypothesis  $h$  is.

Examples:

Zero-one loss:

$$\ell(h, x, y) = \begin{cases} 0 & h(x) = y \\ 1 & h(x) \neq y \end{cases}$$

regression models

Squared loss:

$$\ell(h, x, y) = (h(x) - y)^2$$

$$x_1 = 4,200 \text{ m}^2 \rightarrow$$

$$y_1 = 180.000 \$$$

$$(x_1, y_1)$$

$x_1 = 4,200 \text{ m}^2, \dots \rightarrow$

$\hat{y}_1 = 1 // \text{buy the house}$

free label.  
ground-truth.



# The Loss Function

## Learning/Training

Q: how to select the best hypothesis  $\hat{h}$  from  $\underline{\mathcal{H}}$ ?

With loss  $\ell$  being defined, we can perform **training/learning**:

$\hat{h} = \arg \min_{h \in \mathcal{H}} \sum_{i=1}^n \ell(h, x_i, y_i)$

$$\sum_{i=1}^n$$

for each point.



bad

Training set =  $D$

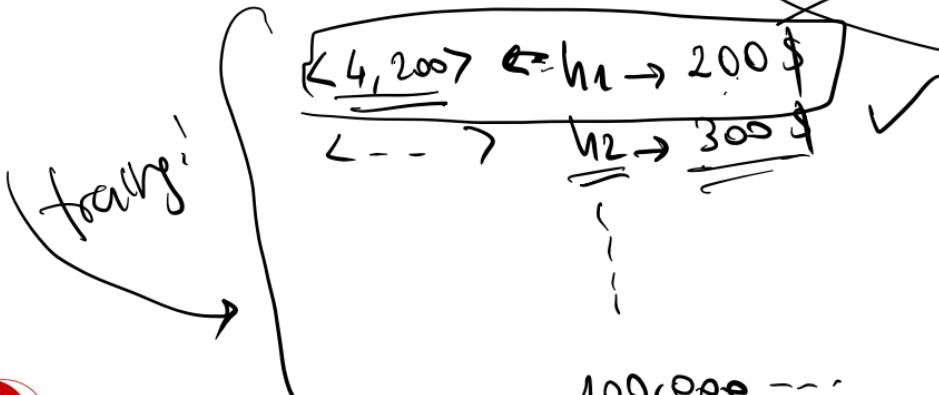
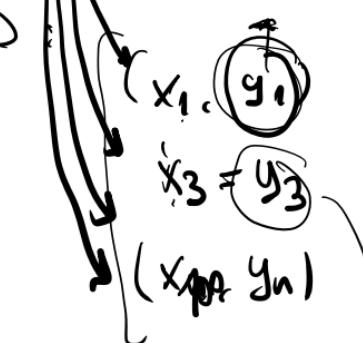
# Learning

training error  $\leq 0$   
test error  $= \uparrow$

- Algorithm 3:

$$h(x) = \begin{cases} y_i & x = x_i \text{ for } (x_i, y_i) \in D \\ y_1 & \text{otherwise} \end{cases}$$

- What is the problem?



$$h_1 \langle 4, 200 \rangle \rightarrow 200 \$$$

$$\underline{h \langle 4, 5, 200 \rangle \rightarrow}$$

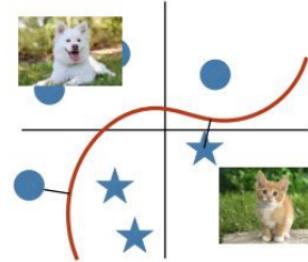
test  $\rightarrow$  test.



# Generalization



ML model



Generalization: how well can our trained  
model do on unseen test examples?



Toss coin

# Formalize with using distribution

The Independent and identically distributed (i.i.d) assumption:

Training data  $\mathcal{D}$  is i.i.d sampled from a distribution  $\mathcal{P}$ , i.e.,  $x_i, y_i \sim \mathcal{P}, \forall i \in [n]$   
(i.e., all pairs are sampled from  $\mathcal{P}$ , and  $(x_i, y_i)$  is independent of others)

We further assume test data is also from  $\mathcal{P}$ , i.e.,  $(x, y) \sim \mathcal{P}$

**Generalization error:**  $\mathbb{E}_{x,y \sim \mathcal{P}} [\ell(\hat{h}, x, y)]$

e.g., expected classification error of  $\hat{h}$



# Overfitting //

tst  
↑

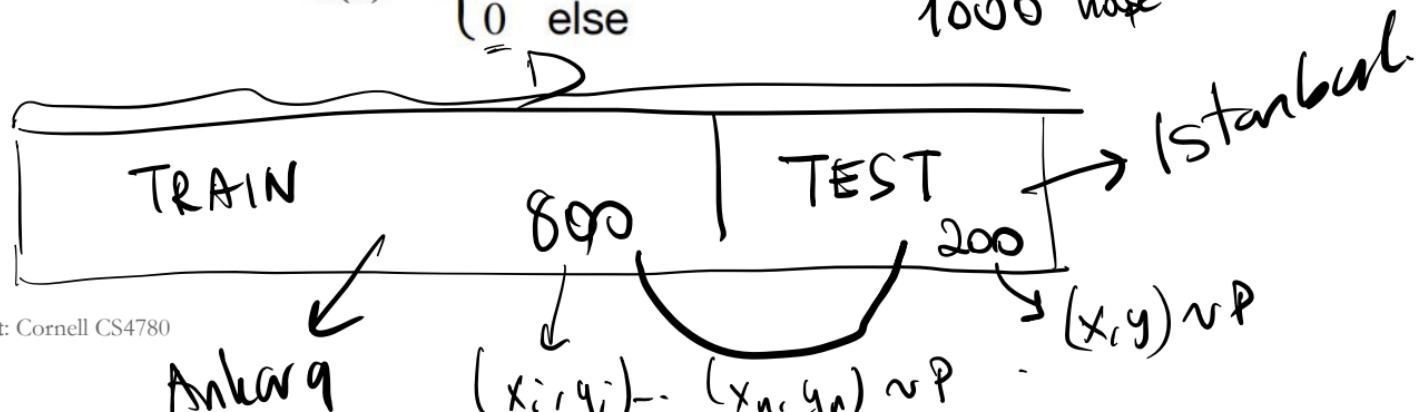
**Overfitting:** we have a small training error but large generalization error

## Example

Hypothesis  $\tilde{h}$  that memorizes the whole training set

$$\tilde{h}(x) = \begin{cases} y_i & \exists (x_i, y_i) \in \mathcal{D} \text{ w/ } x_i = x \\ 0 & \text{else} \end{cases}$$

new example  
1000 base



# Overfitting

**Overfitting:** we have a small training error but large generalization error

## Example

Hypothesis  $\tilde{h}$  that memorizes the whole training set

$$\tilde{h}(x) = \begin{cases} y_i & \exists(x_i, y_i) \in \mathcal{D} \text{ w/ } x_i = x \\ 0 & \text{else} \end{cases}$$

Training error = 0, but could do terribly on test examples



# Overfitting

**Overfitting:** we have a small training error but large generalization error

## Example

Hypothesis  $\tilde{h}$  that memorizes the whole training set

$$\tilde{h}(x) = \begin{cases} y_i & \exists(x_i, y_i) \in \mathcal{D} \text{ w/ } x_i = x \\ 0 & \text{else} \end{cases}$$

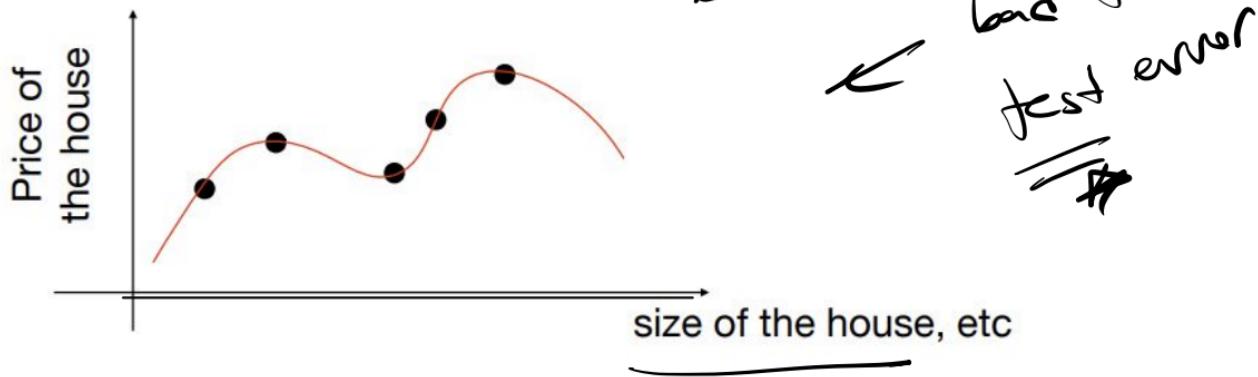
Training error = 0, but could do terribly on test examples



# Overfitting

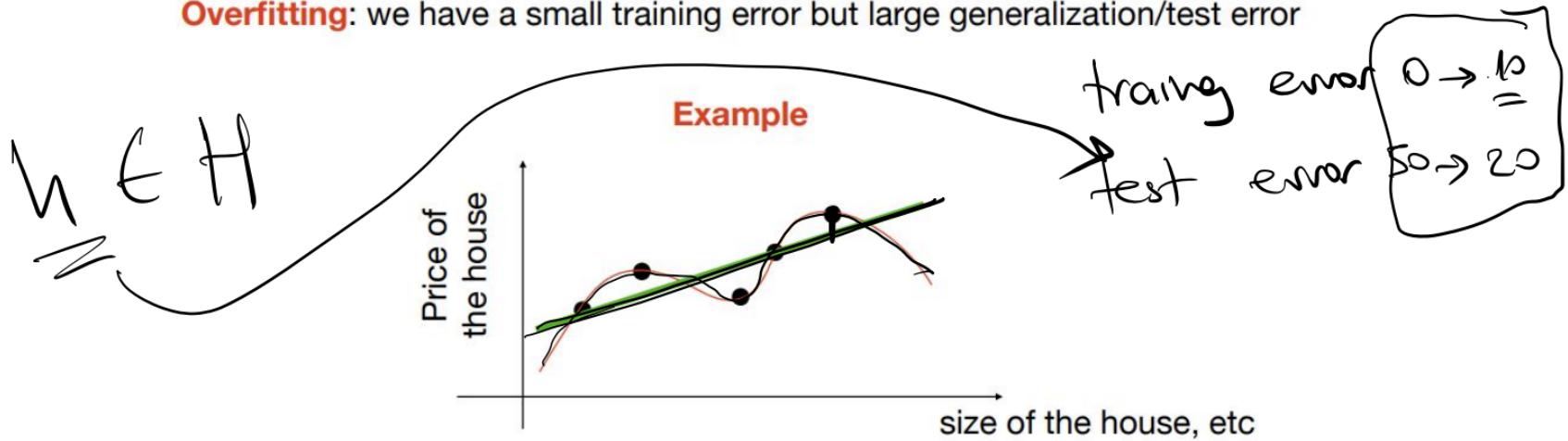
**Overfitting:** we have a small training error but large generalization/test error

**Example**



# Overfitting

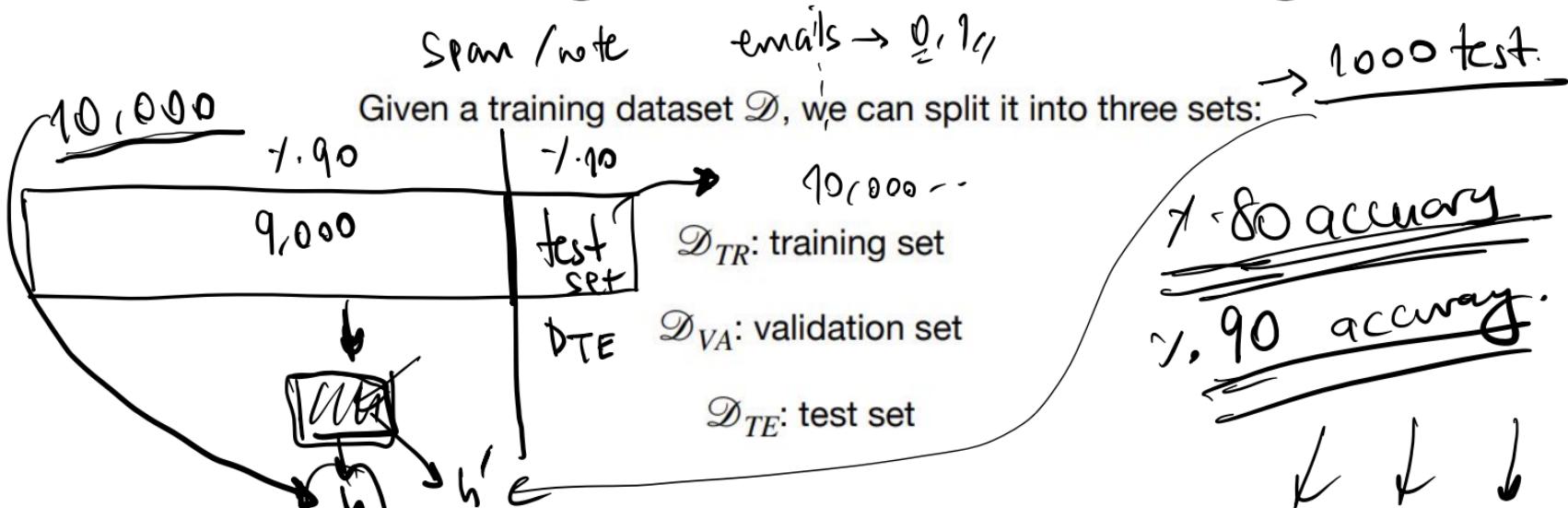
**Overfitting:** we have a small training error but large generalization/test error



Training error = 0 (e.g., we probably overfit to noises), but could do terribly on test examples



# Training, validation, and testing



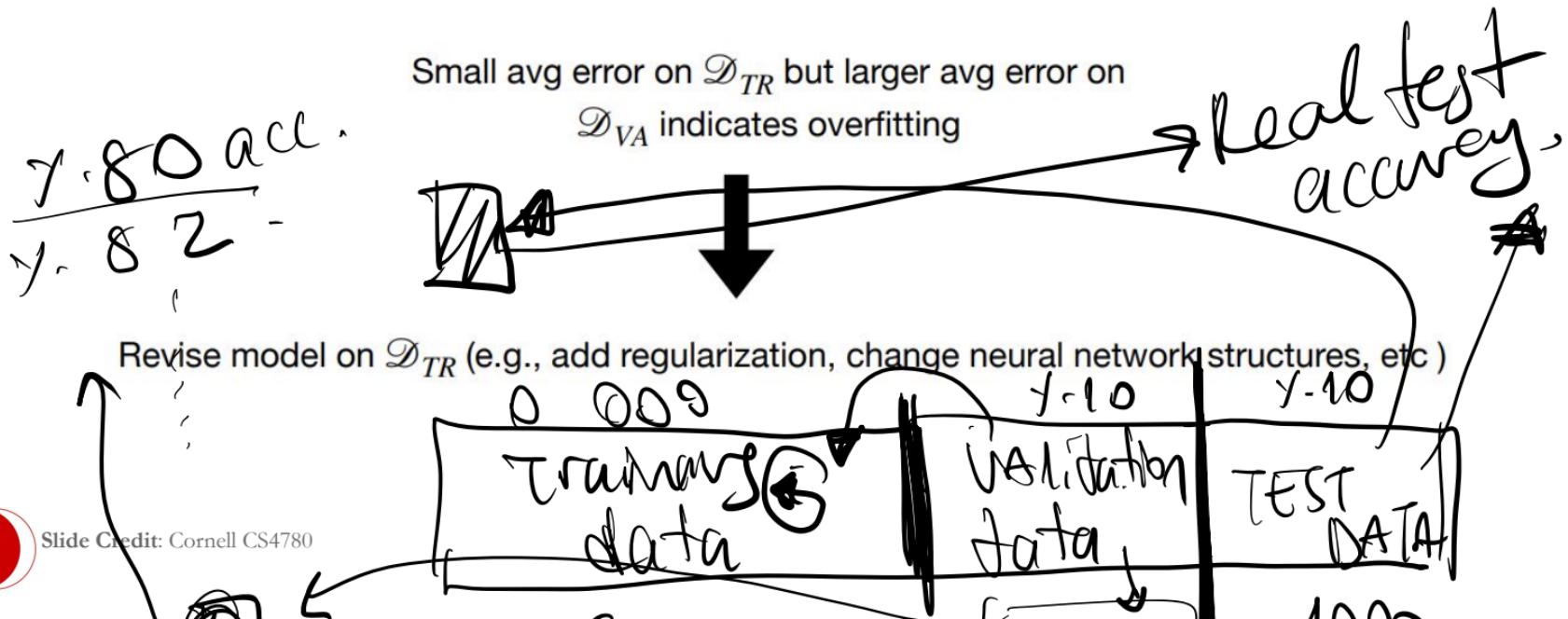
training error

$$\mathcal{D}_{TE-1} = \frac{1}{10} = 1$$
$$\mathcal{D}_{TE-2} = \frac{0}{10} = 0$$

1 ✓ 0

# Selecting models using validation set

We can use validation set to select models, i.e., select hypothesis class, tune parameters, etc



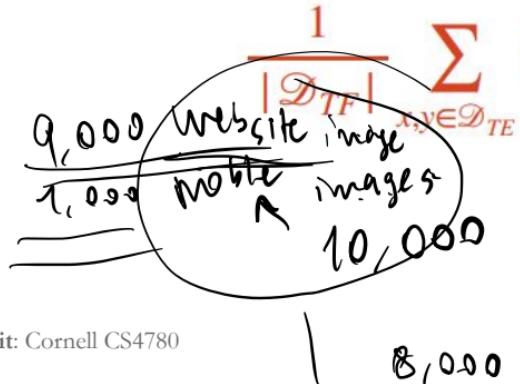


Do not use test set to train/select models

We should not touch test set during training!

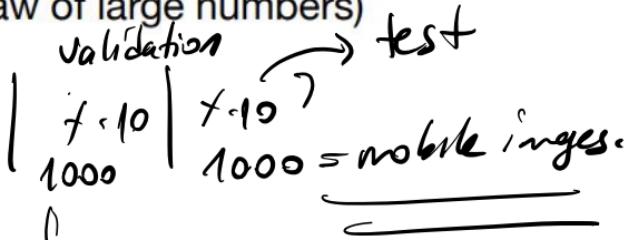
This makes sure that the test set  $\mathcal{D}_{TE}$  is independent of our model  $\hat{h}$

Such independence implies that:



train  
7.80.

(Due to law of large numbers)



validation acc : 7.90

test acc : 7.60

# Do not use test set to train/select models

What if our original dataset is quite small, e.g.,  $n = 100$   
(very possible in medical applications!)

training  
model

## K-fold cross validation

Split the data into K folds (e.g.,  $K = 10$  or  $20$ )

For  $i = 1 \rightarrow K$ : 5

$\mathcal{D}_{TR}$ : all others folds except the i'th fold

Train model  $\hat{h}$  on  $\mathcal{D}_{TR}$

Validate on the i'th fold (i.e.,  $\mathcal{D}_{VR} = i$ 'th fold)

Average K validation errors

test accuracy

average

$K = 5$

1 day

1

When  $K = n$ , this is leave-one-out cross validation

$K=1$	Val	Test	$m_1$
$K=2$	Val	Test	$m_2$
$K=3$	Val	Test	$m_3$
$K=5$			$m_{avg}$



5 days.

# Summary

1. Given a task and a dataset  
 $\mathcal{D} = \{x_i, y_i\}, x_i, y_i \sim \mathcal{P}$

Output:  $\hat{h}$  that has small generalization error  
 $\mathbb{E}_{x,y \sim \mathcal{P}}[\ell(\hat{h}, x, y)]$

2. Design hypothesis class  $\mathcal{H}$  and loss function  $\ell$  (encodes inductive bias)

3. Train:  $\hat{h} = \arg \min_{h \in \mathcal{H}} \sum_{(x,y \in \mathcal{D})} \ell(h, x, y)$

Often repeated many times using  $\mathcal{D}_{VA}$  / cross validation

$$h_{\text{fit}}(x) = \underline{\quad}$$



# Machine Learning Terminology

- **Dataset:** A table with the data from which the machine learns. The dataset contains the features and the target to predict.
- **Instance:** The thing about which you want to make a prediction. For example, the instance might be a web page that you want to classify as either "about cats" or "not about cats".  
  
An arrow points from a circled  $(x, y)$  to the word Instance / Sample.
- **Label:** An answer for a prediction task either the answer produced by a machine learning system, or the right answer supplied in training data. For example, the label for a web page might be "about cats".
- **Feature:** A property of an instance used in a prediction task. For example, a web page might have a feature "contains the word 'cat'".

$x_1: \langle 4, 200m^2, \dots \rangle = \text{instance}$

$y_1 = 20,000 \$$  *label*



# Machine Learning Terminology

instance  $\leftarrow (x_1, y_1)$

- **Example:** An instance (with its features) and a label.
- **Model:** A statistical representation of a prediction task. You train a model on examples then use the model to make predictions.
- **Metric:** A number that you care about. May or may not be directly optimized.
- **Objective:** A metric that your algorithm is trying to optimize.
- **Pipeline:** The infrastructure surrounding a machine learning algorithm. Includes gathering the data from the front end, putting it into training data files, training one or more models, and exporting the models to production.
- **Prediction:** what the ML model “guesses” what the target value should be based on the given features.



# Next Class:

Continue with Machine Learning Concepts & KNN

