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
  - bbatuhan@metu.edu.tr
- **Lectures**:
  - Tuesday: 13:40 - 16:30, Online
- **TA:** Burcu Alakuş Çınar
  - 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)

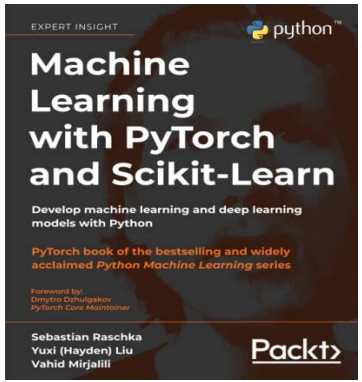
Academic Integrity

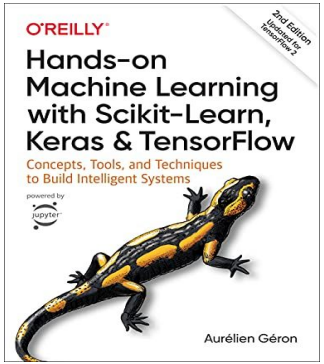
- **Zero tolerance policy:** All occurrences will be reported

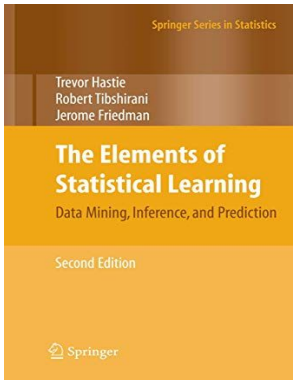
- **AI tool policy:** You can use AI-based tools, but you need to provide references. You also need to mention how you used this tool (prompts, etc.) and its contribution to you.

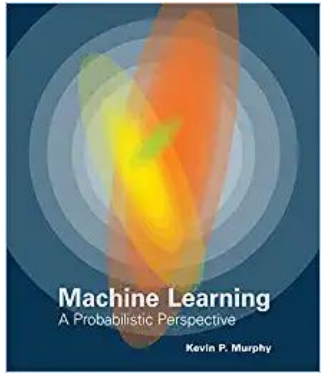
Reference Books

- 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

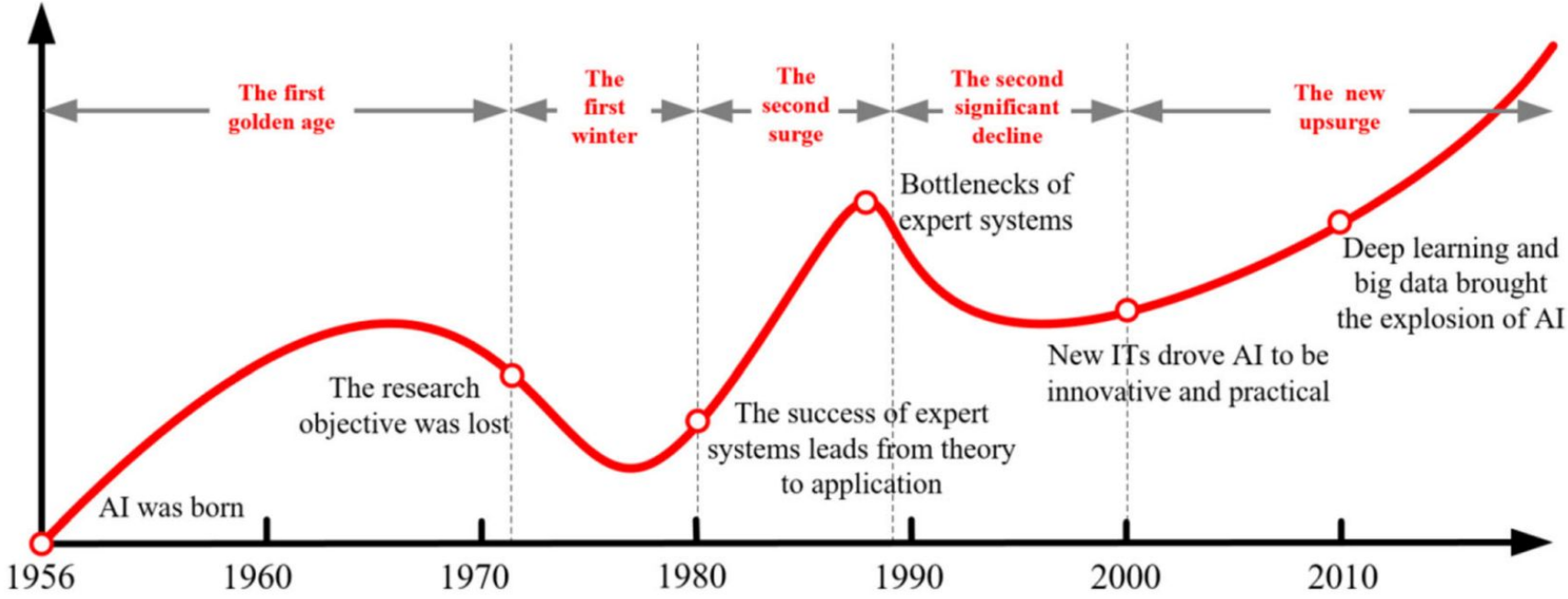
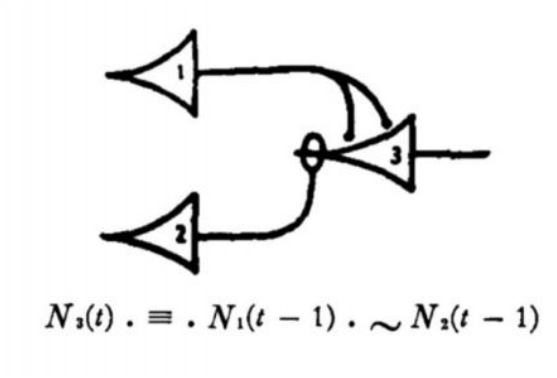


Figure: Artificial Intelligence in product lifecycle management

A Mathematical Model of Brain: McCulloch & Pitts Neuron

Artificial Neurons



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

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

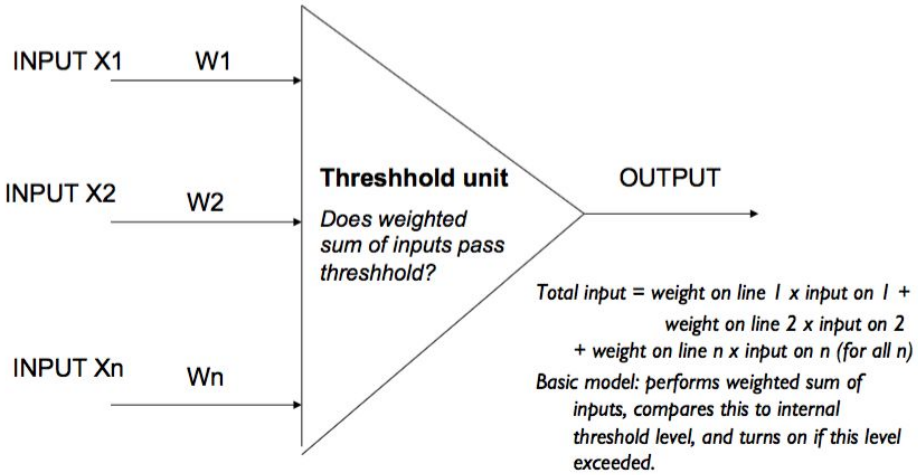
WARREN S. MCCULLOCH AND WALTER PITTS

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.

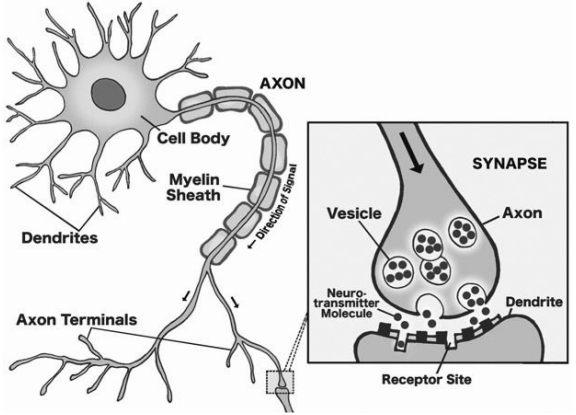
McCulloch & Pitts

McCulloch Pitts Neuron



Total input = weight on line 1 x input on 1 + weight on line 2 x input on 2 + weight on line n x input on n (for all n)

Basic model: performs weighted sum of inputs, compares this to internal threshold level, and turns on if this level exceeded.



Inputs arrive at dendrites, and axons serve as output channel

McCulloch & Pitts

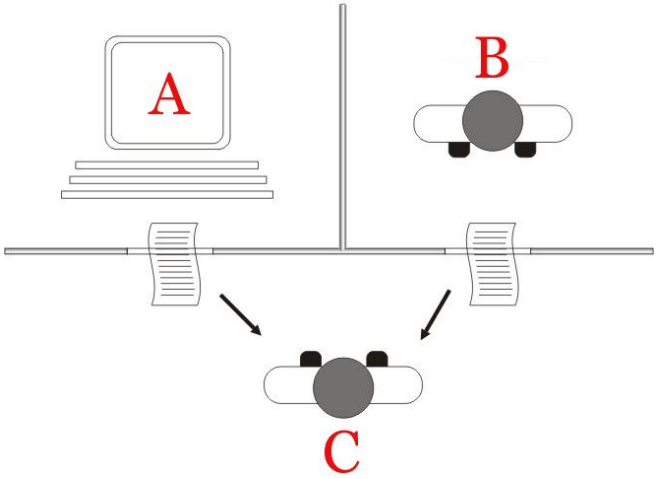
```python
def mcculloch_pitts_neuron(inputs, weights, threshold):
    total_input = sum(w * inp for w, inp in zip(weights, inputs))
    return int(total_input >= threshold)


# 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?



Alan M. Turing (1950)

Dartmouth AI Project Proposal (1956)

"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."

Slide Credit: Dhruv Batra

Perceptron

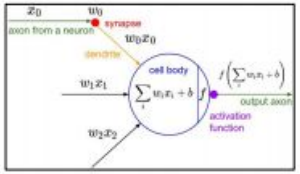
A bit of history...

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

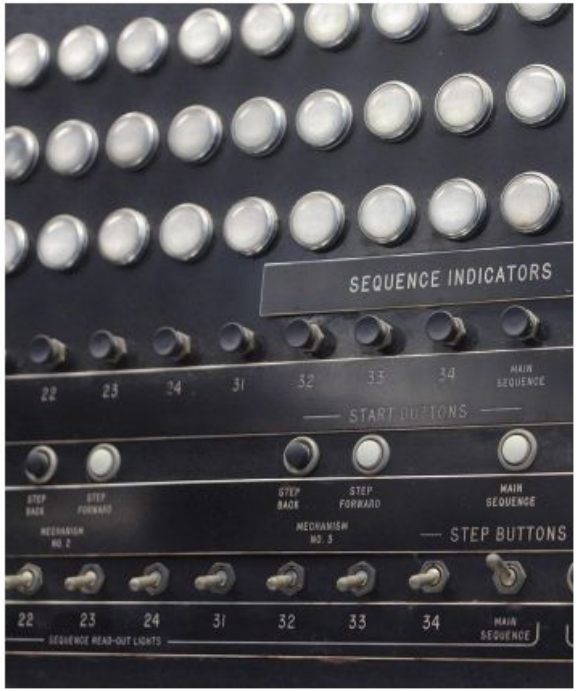
$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$



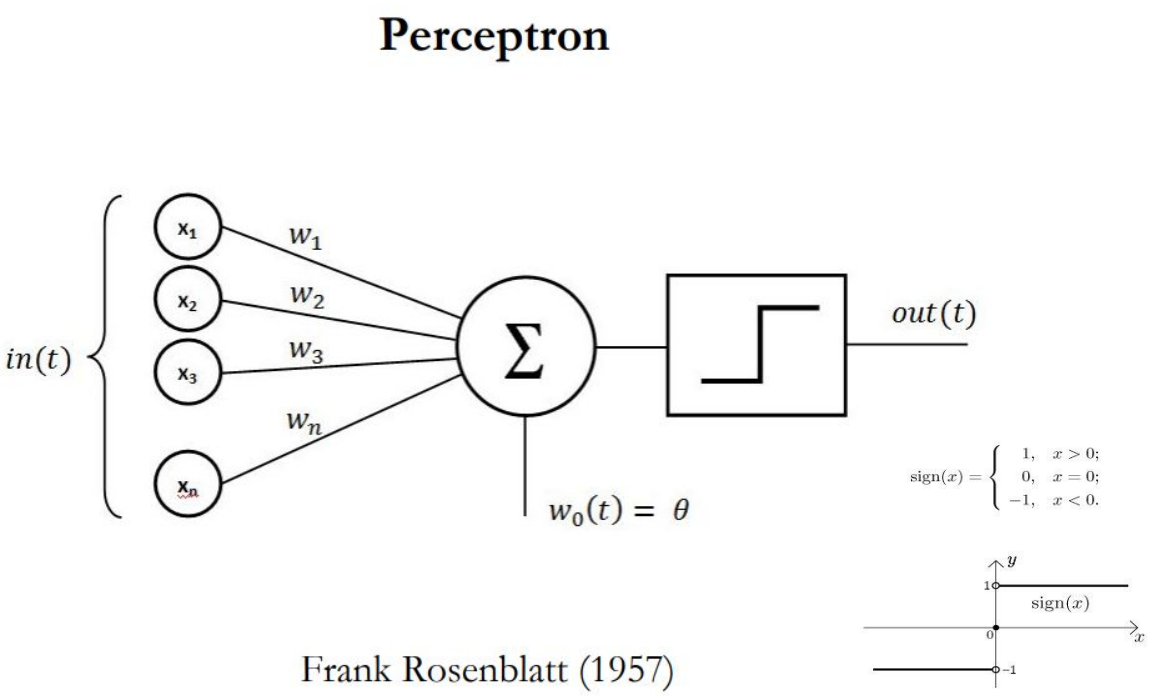
update rule:

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

Frank Rosenblatt, ~1957: Perceptron

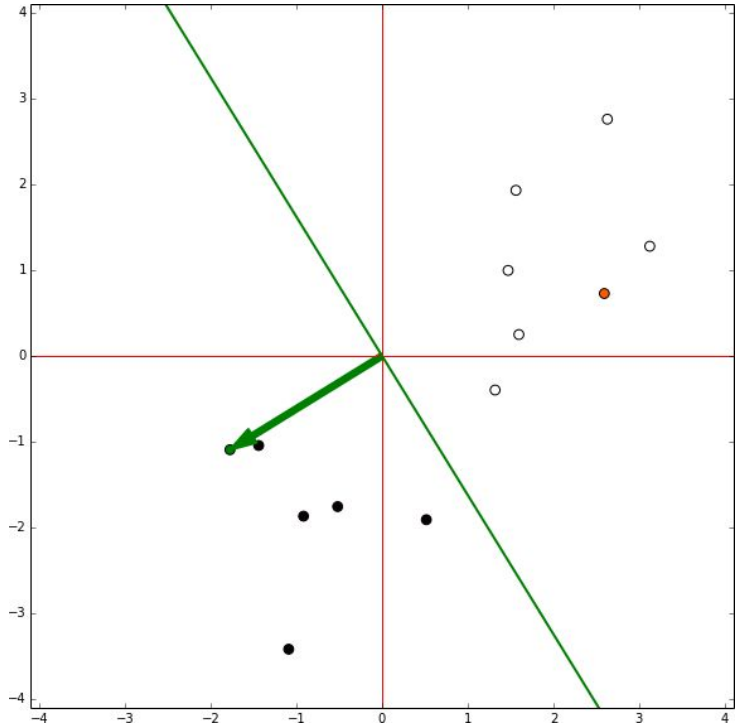


Perceptron

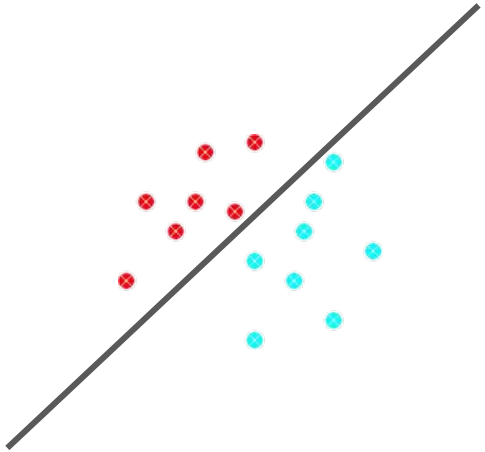


Frank Rosenblatt (1957)

Perceptron



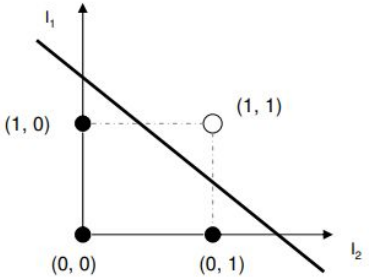
Linear Classification



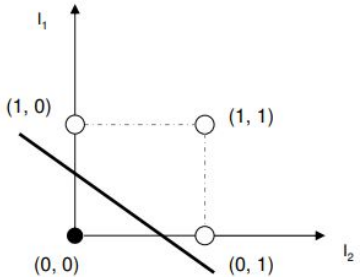
Perfect!

XOR Case

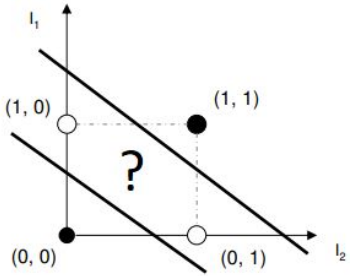
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



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

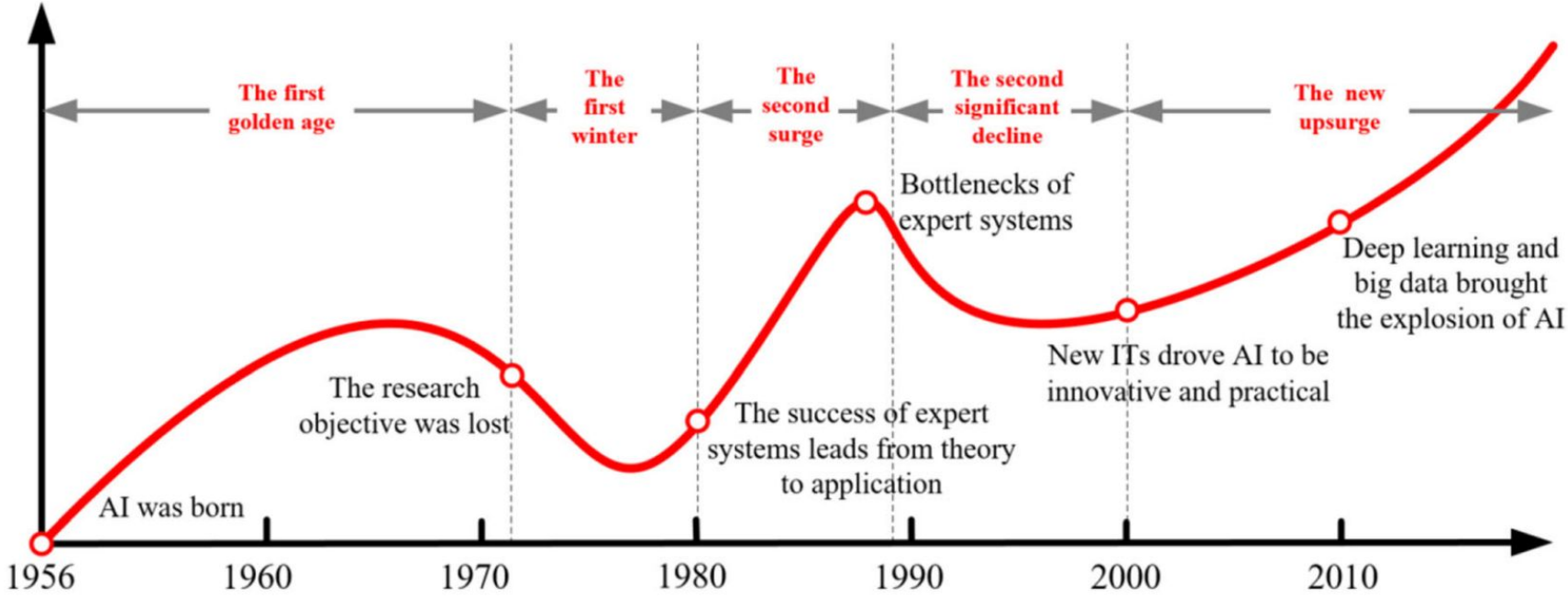


Figure: Artificial Intelligence in product lifecycle management

Moving the First AI Winter

Geoffrey Hinton

FOLLOW

GET MY OWN PROFILE

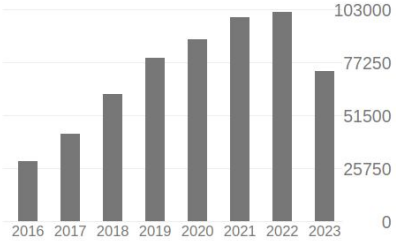
Emeritus Prof. Computer Science, University of Toronto
Verified email at cs.toronto.edu - Homepage

machine learning psychology artificial intelligence cognitive science computer science

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



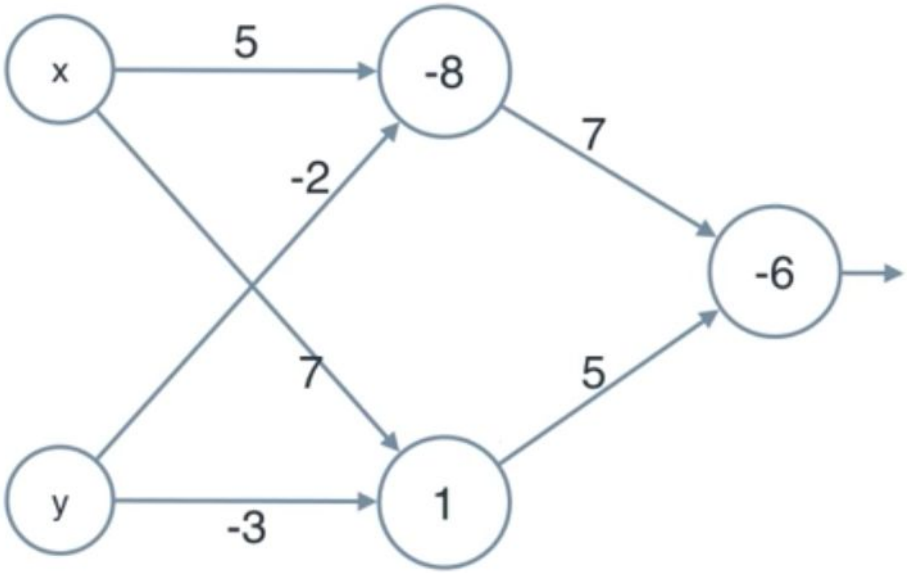
Public access VIEW ALL

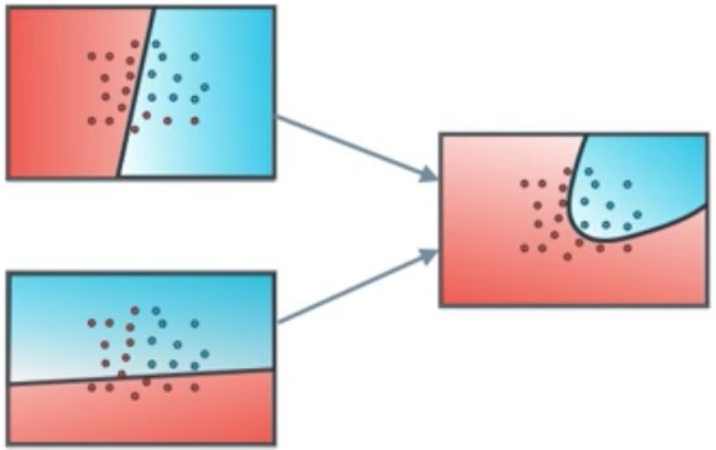
1 article 7 articles

not available available

Based on funding mandates

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

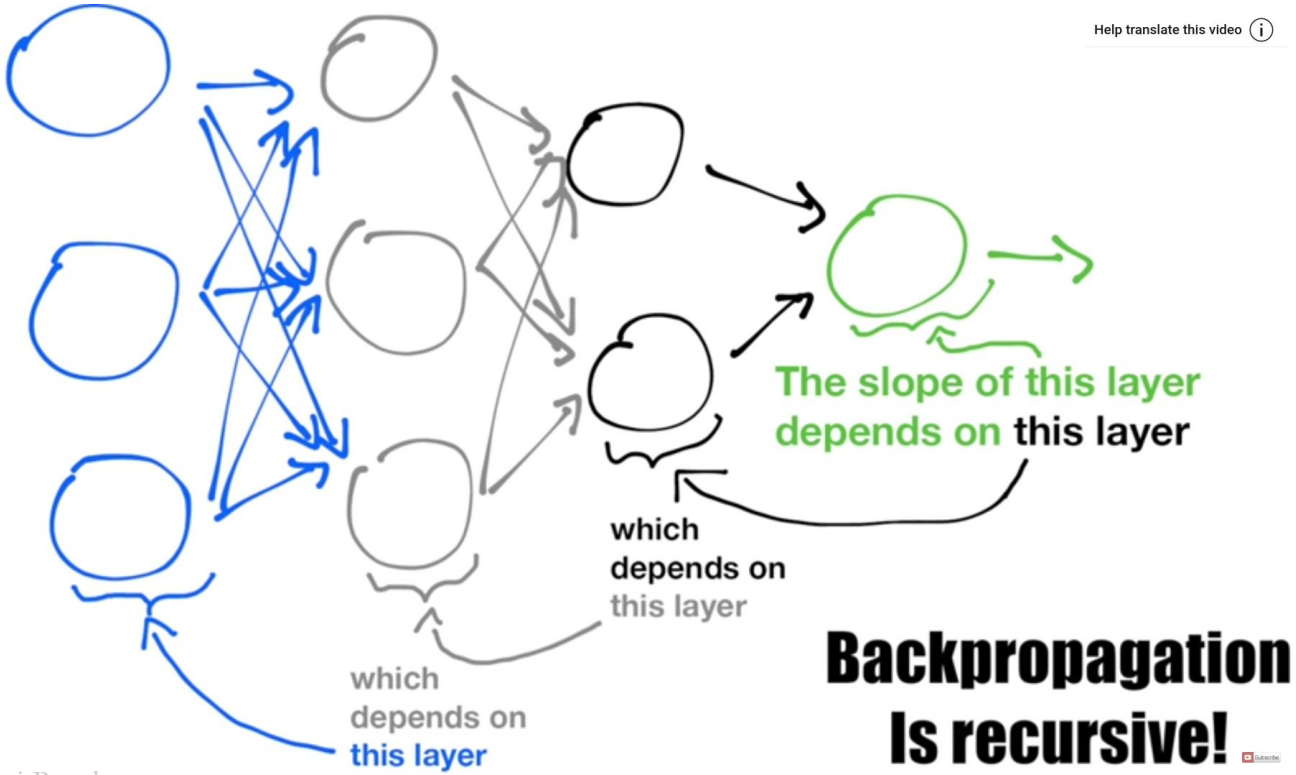
Geoffrey E. Hinton
Department of Computer Science
Carnegie-Mellon University

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

To be published in D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: 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 Programs 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.

Backpropagation Algorithm

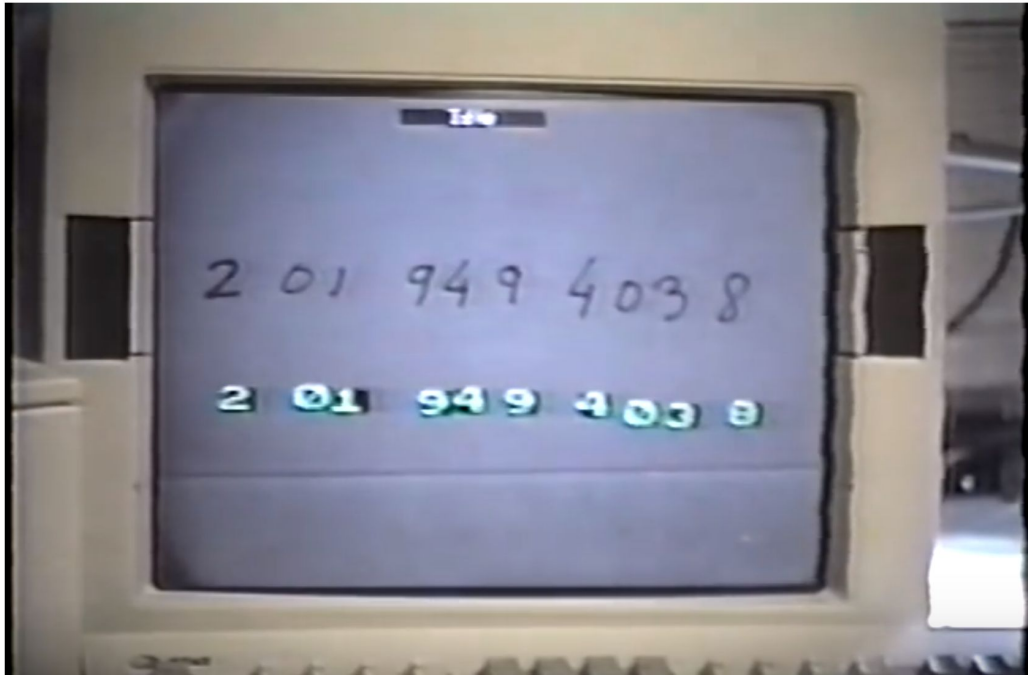


Slide Credit: Siraj Raval

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 Holmde



History of AI

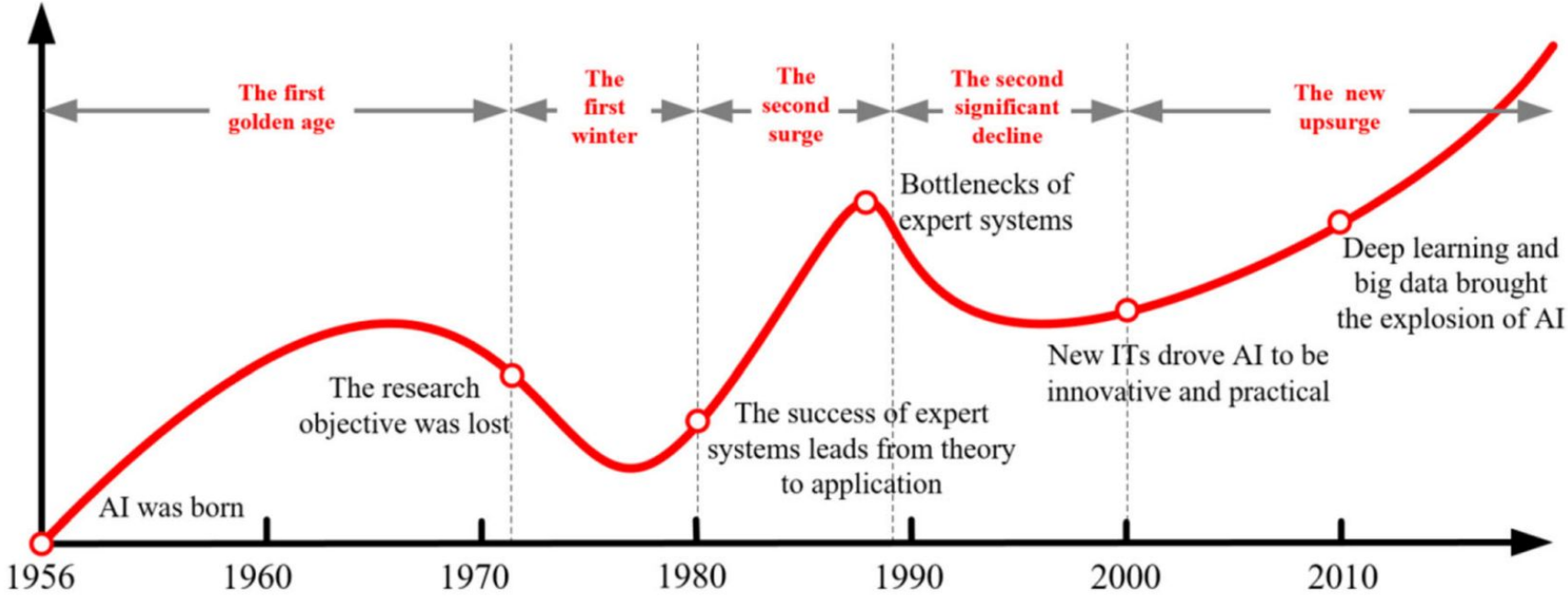


Figure: Artificial Intelligence in product lifecycle management

Second AI Winter

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

Moving the Second AI Winter

Geoffrey Hinton

FOLLOW

GET MY OWN PROFILE

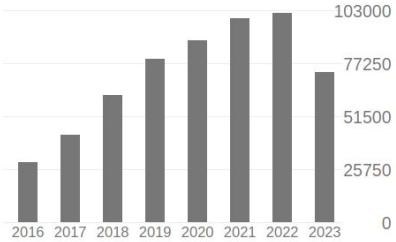
Emeritus Prof. Computer Science, University of Toronto
Verified email at cs.toronto.edu - Homepage

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

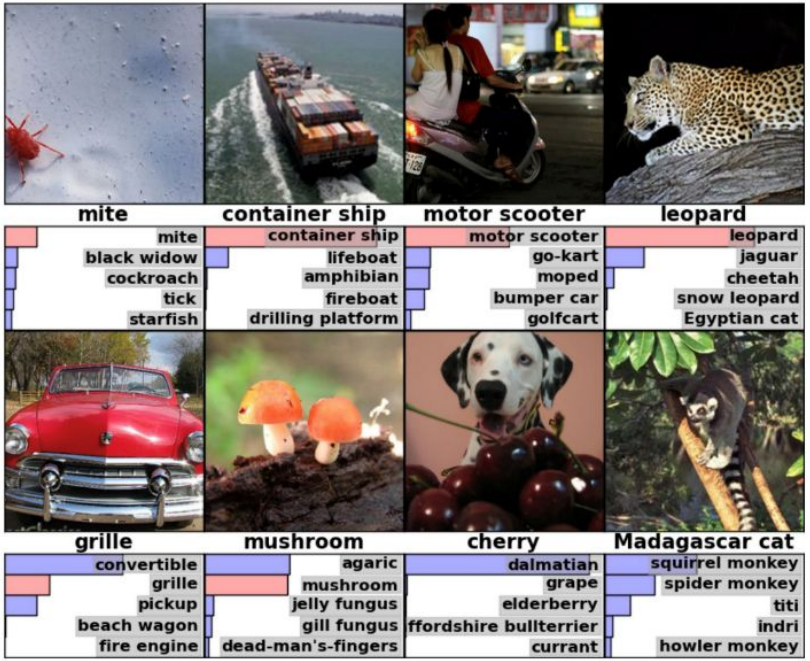
Based on funding mandates

ImageNet

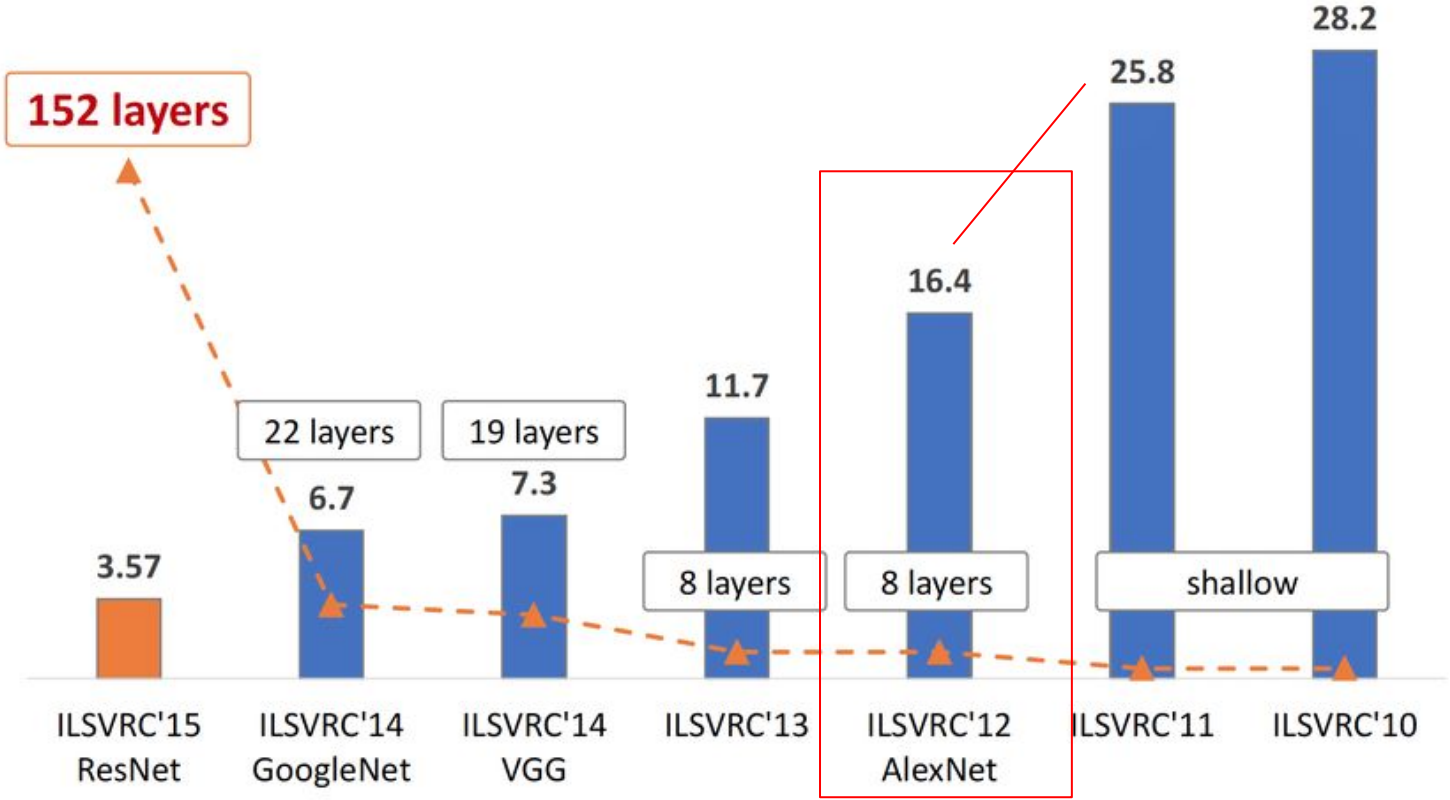
ImageNet Challenge

IMAGENET

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



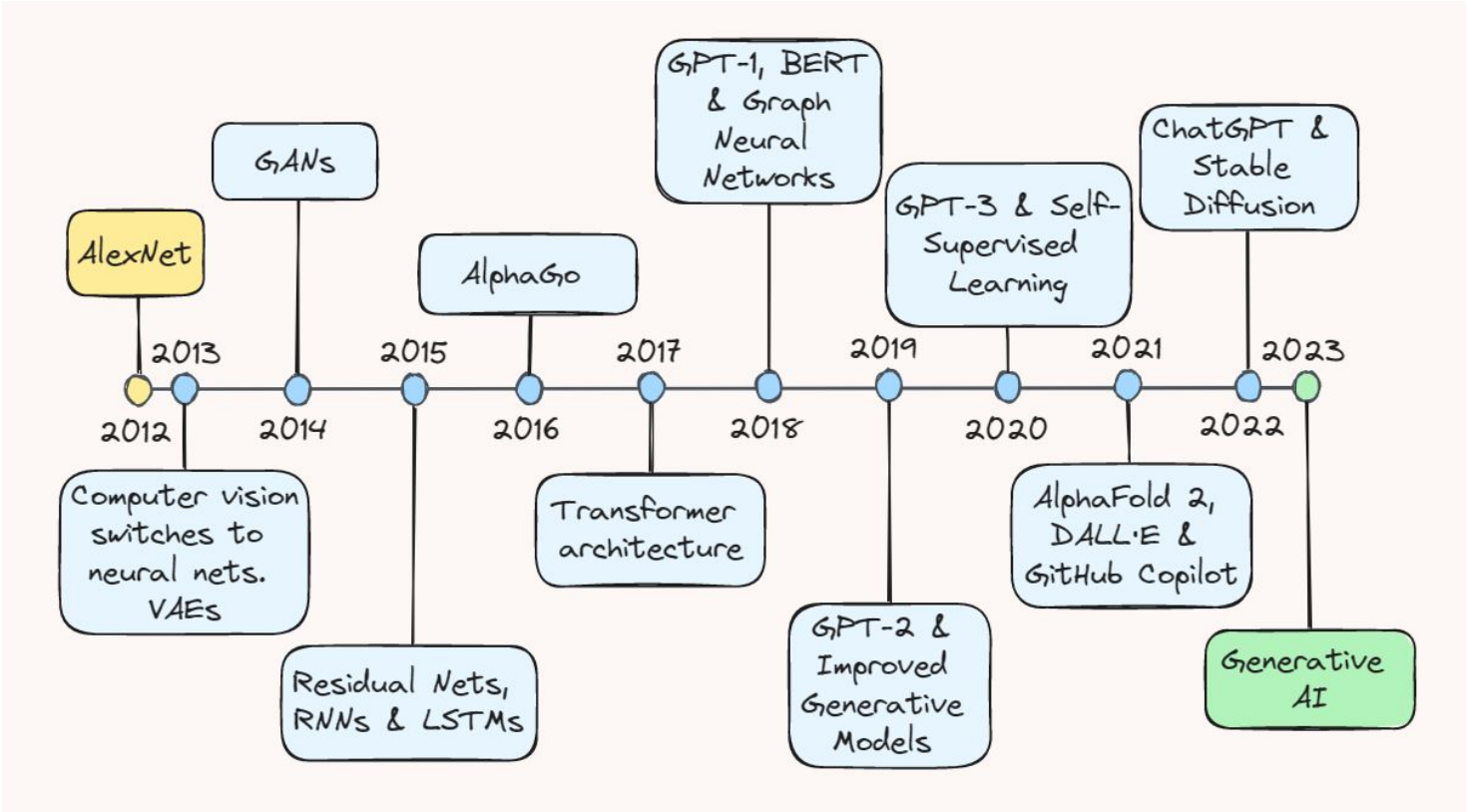
ImageNet


152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10

Moving from Second AI Winter

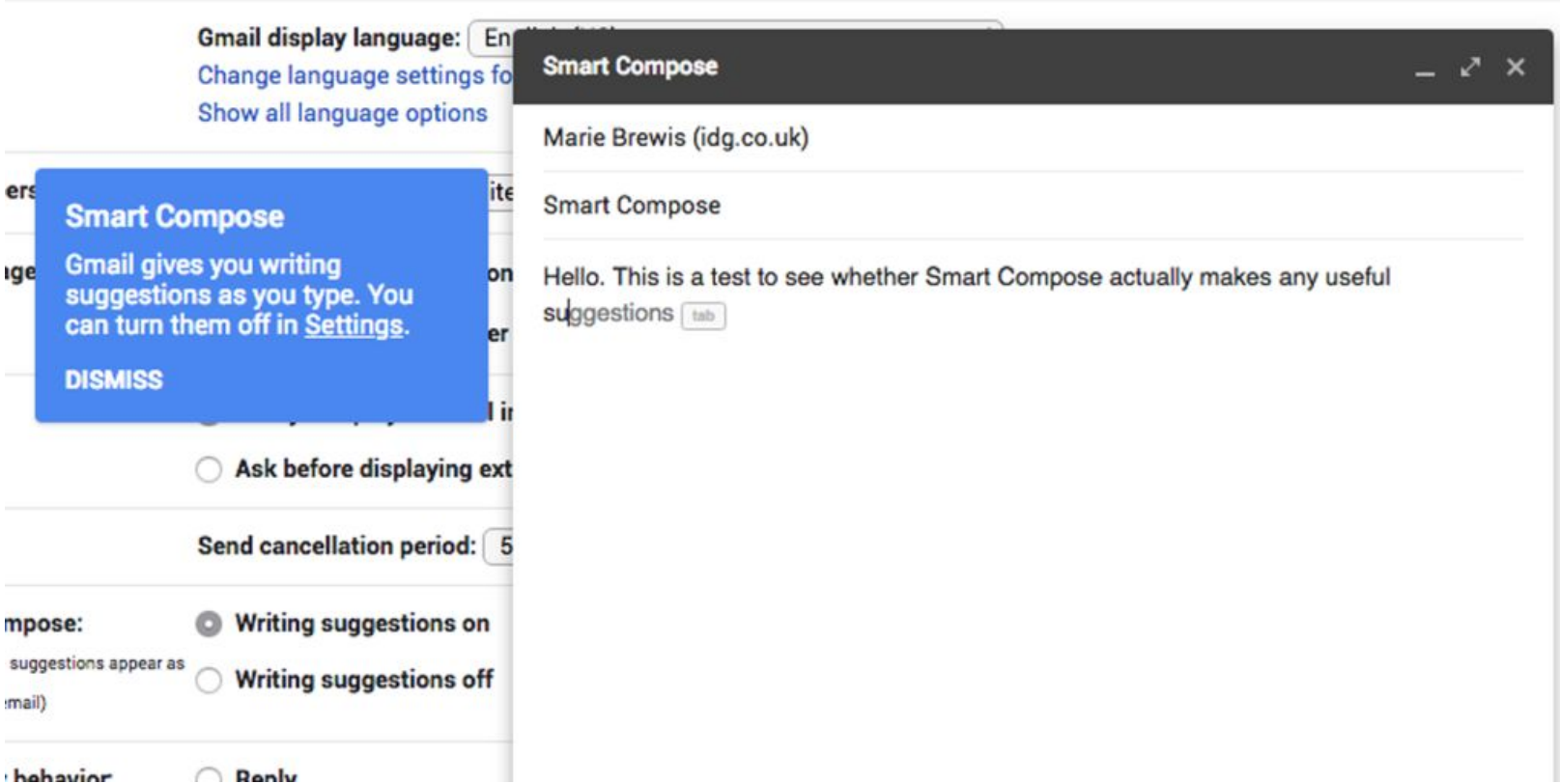
- **Advances in Neural Networks**
- **Rise of Computational Power**
- **Availability of Big Data**
- **IBM Deep Blue's Victory**
- **Open Source Movement**
- **Establishment of AI Companies**

History of AI - 2

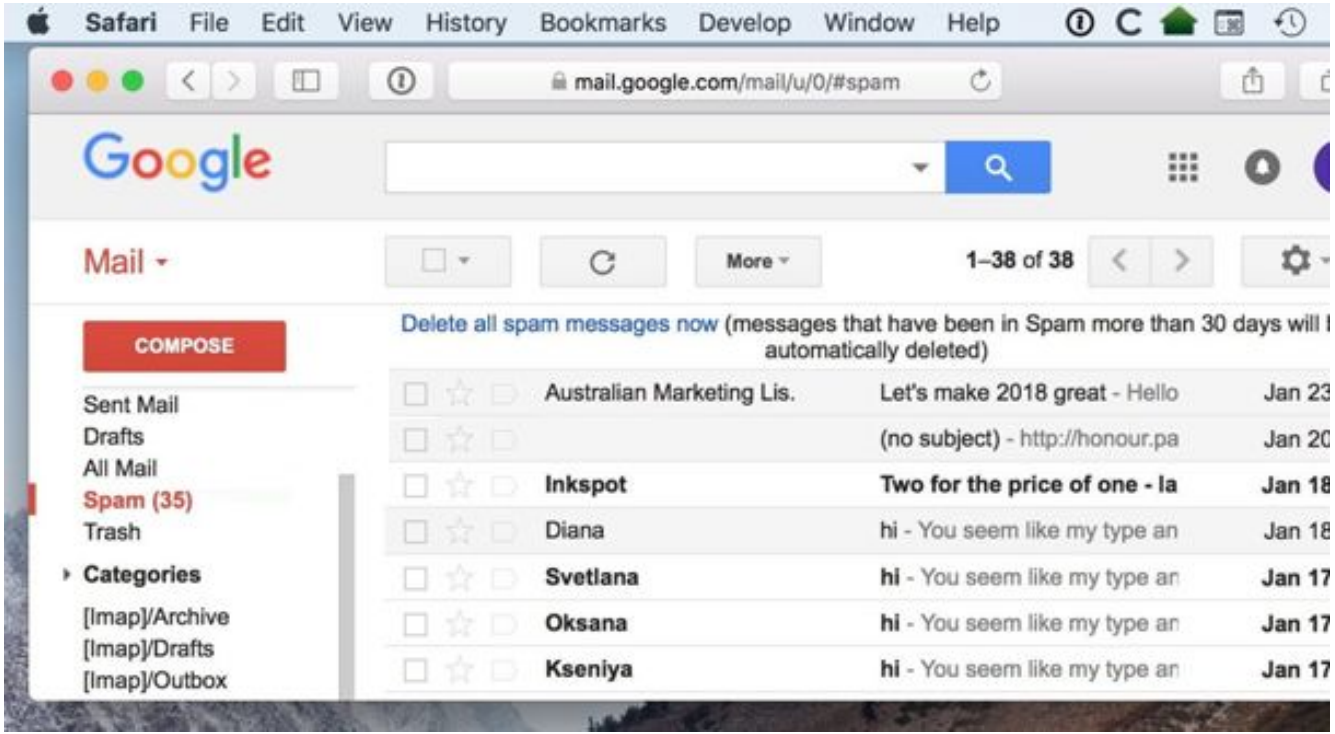


Applications

Text Generation



Spam Detection



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-lear… by Sebastian Raschka Paperback $41.99
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Inte… by Aurélien Géron Paperback $56.16
- The Hundred-Page Machine Learning Book by Andriy Burkov Paperback $31.99

Add to List

Share <Embed>

Have one to sell?

Sell on Amazon

amazon book clubs early access

Add to book club

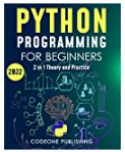
Not in a club? Learn more

Products related to this item

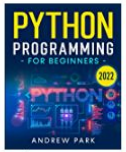
Page 1 of 35

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
$49.39 prime



Python Programming for Beginners: The #1 Python Programming Crash Course for…
Codeone Publishing
356
#1 Best Seller
Paperback
$19.45 prime



Python Programming for Beginners: The Ultimate Crash Course to Learn Python in 7 Da…
Andrew Park
184
#1 Best Seller
Paperback
$13.95 prime

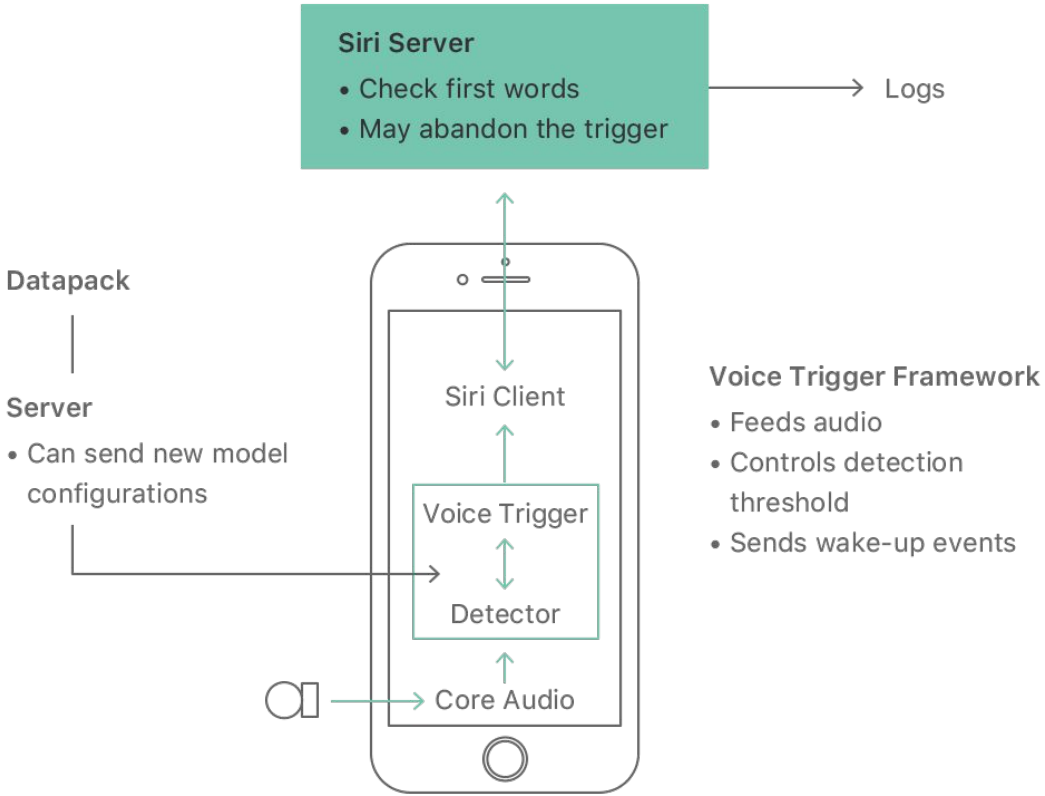
Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep l…
Sebastian Raschka
This book is a comprehensive guide to machine and deep learning using PyTorch's simple to code framework
131
Paperback
$46.79 prime

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
89
Paperback
$44.64 prime

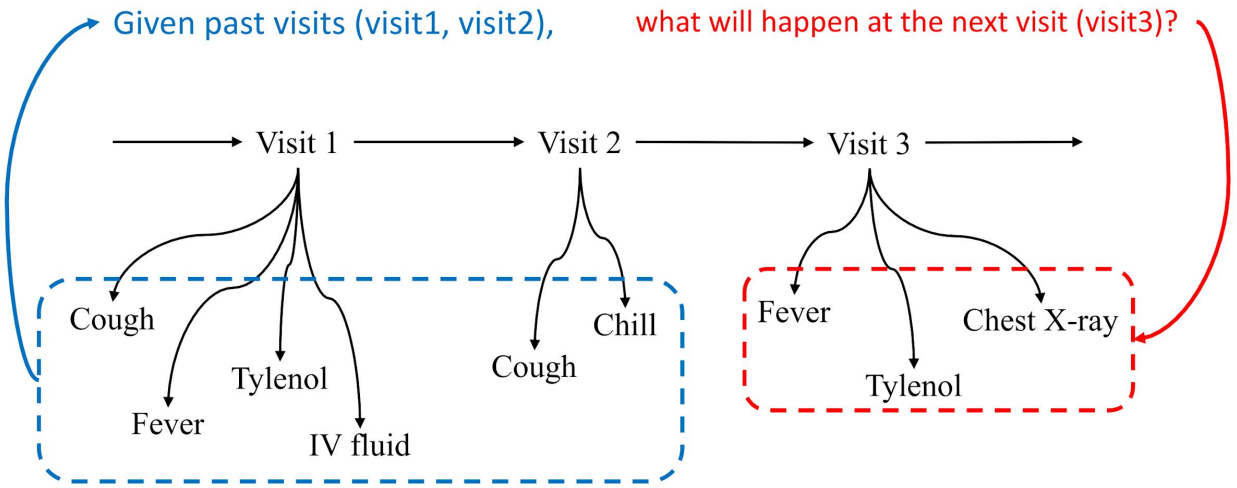
Hands-On Data Science for Marketing: Improve your marketing strategies with…
Yoon Hyup Hwang
63
Paperback
$44.99 prime

Python Automation Cookbook: 75 Python automation ideas for web scraping, data…
Jaime Buelta
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



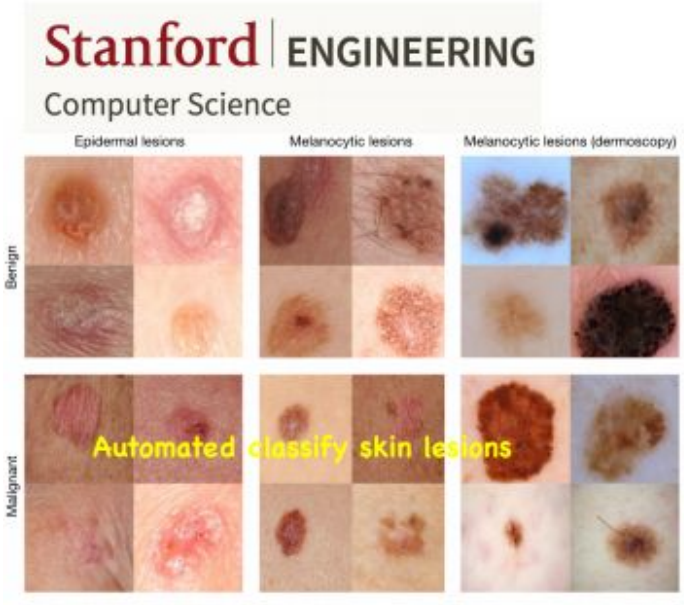
AI in Healthcare

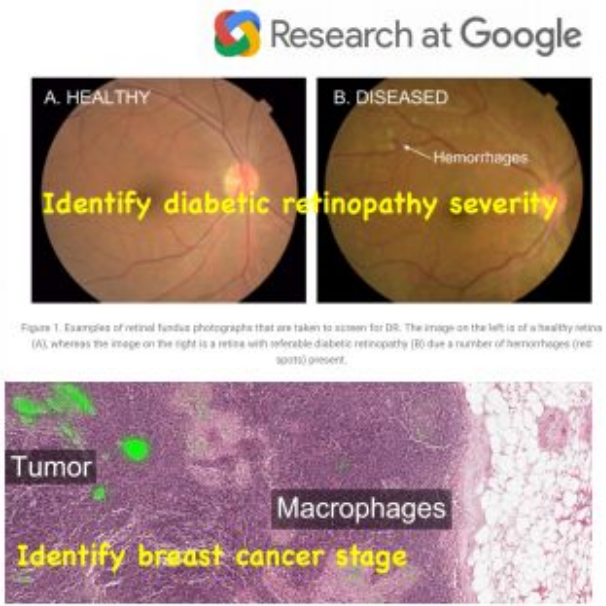


Slide Credit: Choi, Lui (doctor ai)

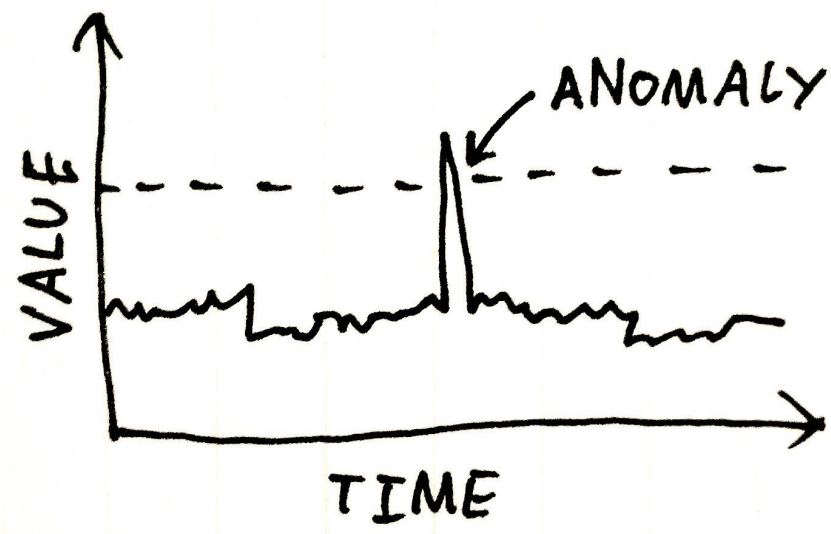
AI in Healthcare

Unstructured Imaging Data: Data





AI in Cybersecurity



AI in Cybersecurity

(RHSA-2021:0736) Critical: java-1.8.0-ibm security update
2021-03-04 22:36:10

CVSS 7.5
5.1

ID RHSA-2021:0736
Type redhat
Reporter RedHat
Modified 2021-03-04 22:38:34

Description

IBM Java SE version 8 includes the IBM Java Runtime Environment and the IBM Java Software Development Kit.

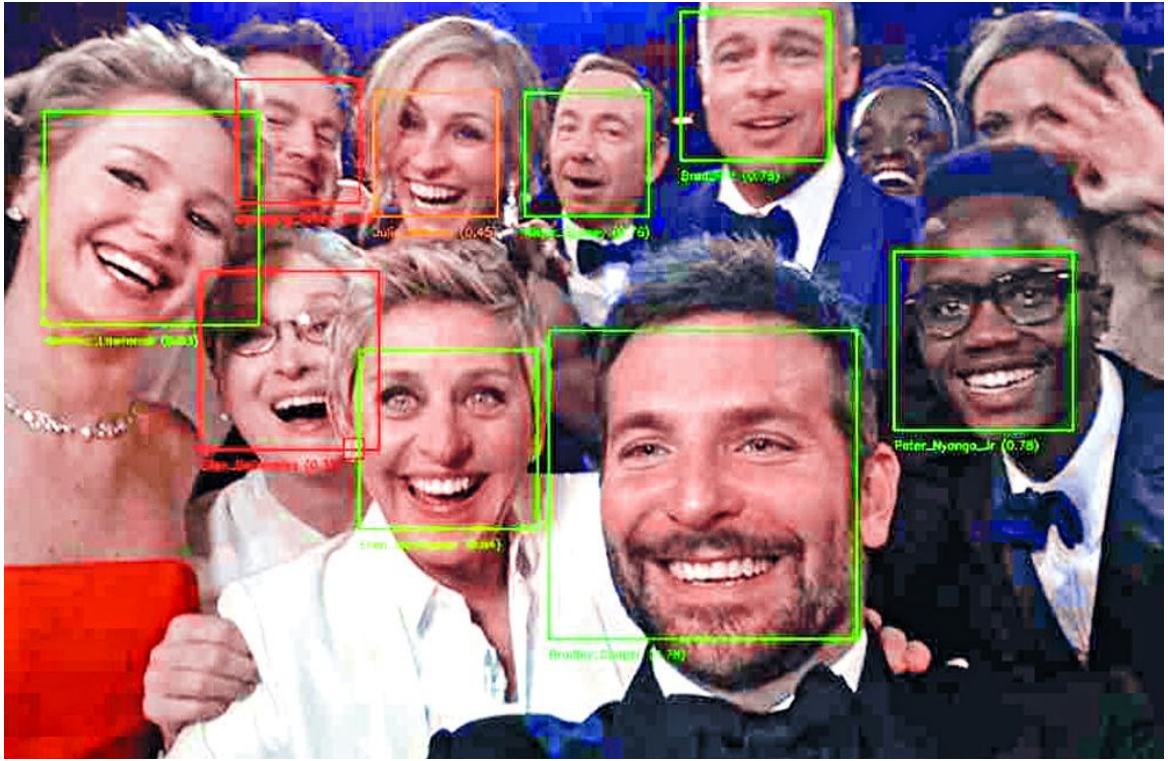
This update upgrades IBM Java SE 8 to version 8 SR6-FP25.

Security Fix(es):

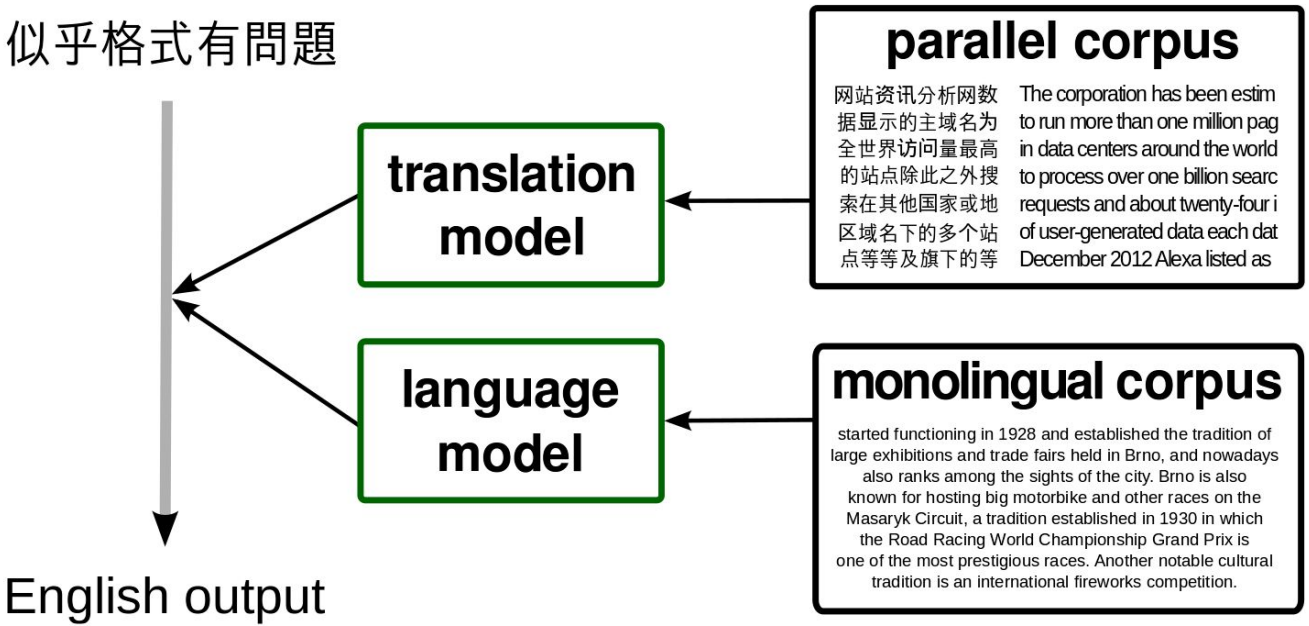
- IBM JDK: Stack-based buffer overflow when converting from UTF-8 characters to platform encoding (CVE-2020-27221)
- OpenJDK: Unexpected exceptions raised by DOMKeyInfoFactory and DOMXMLSignatureFactory (Security, 8231415) (CVE-2020-2773)
- OpenJDK: Credentials sent over unencrypted LDAP connection (JNDI, 8237990) (CVE-2020-14781)

Natural Language Processing - Named Entity Recognition

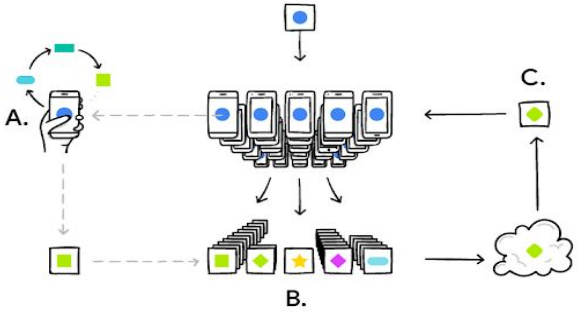
Face Recognition

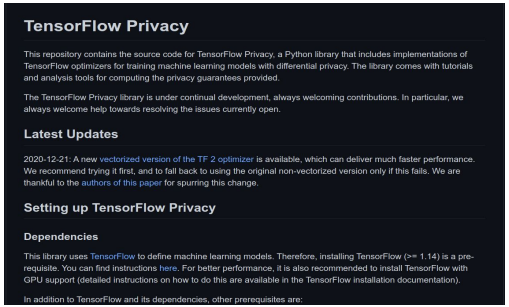


Machine Translation

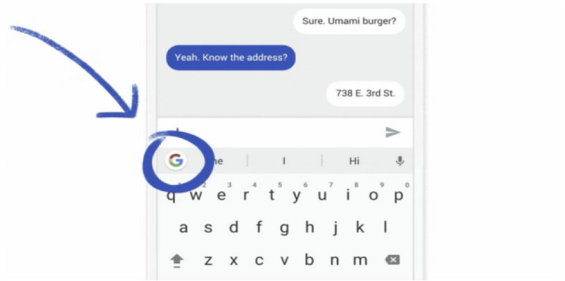


Privacy Preserving ML

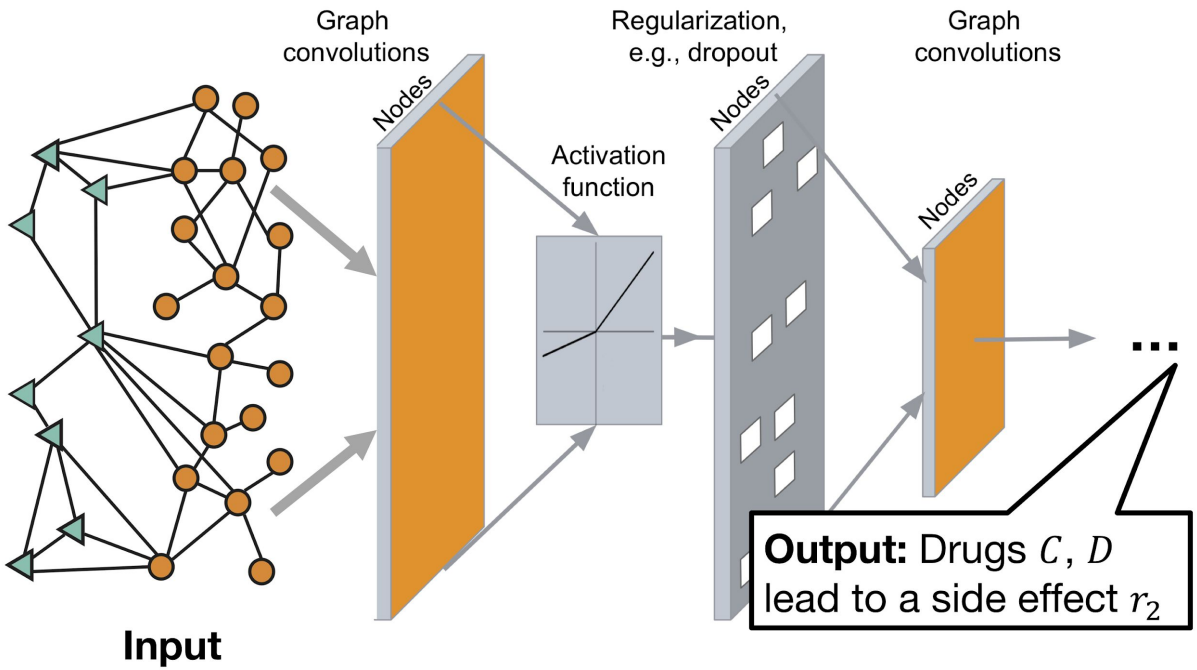




We're currently testing Federated Learning in Gboard on Android, the Google Keyboard. When Gboard shows a suggested query, your phone locally stores information about the current context and whether you clicked the suggestion. Federated Learning processes that history on-device to suggest improvements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



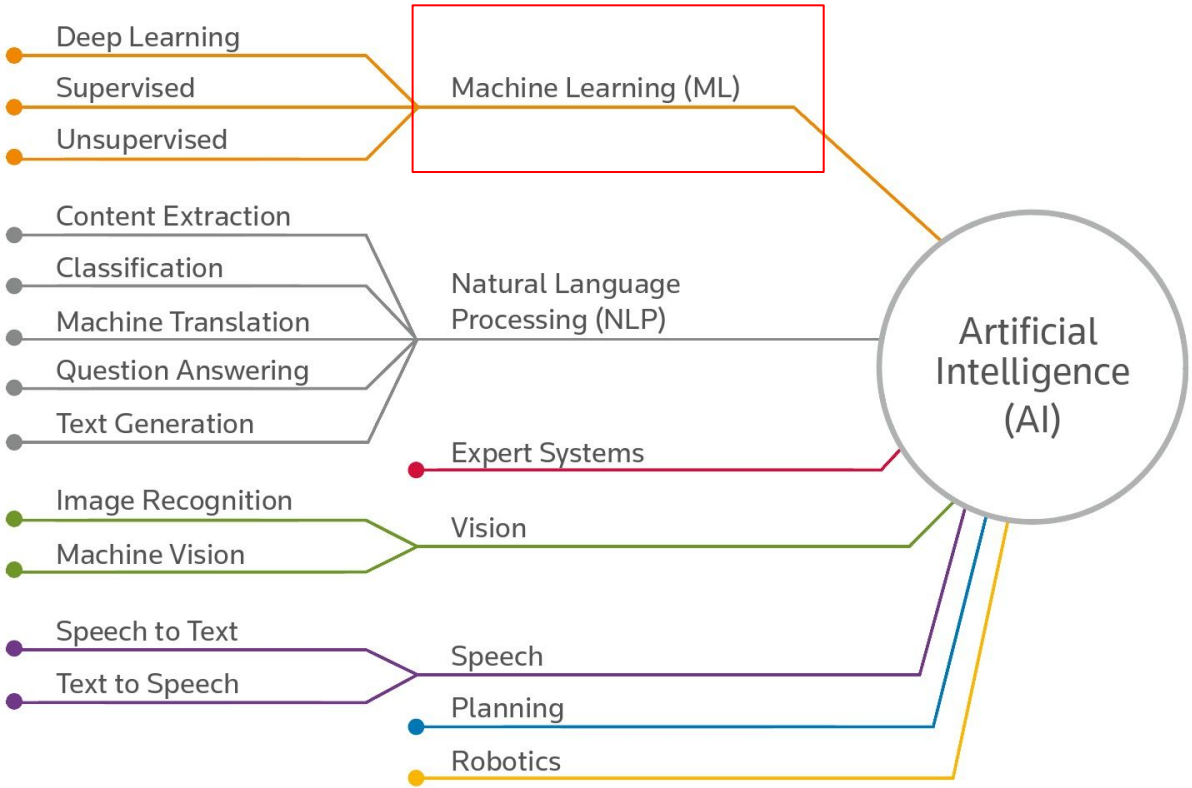
Graph ML


Graph convolutions
Nodes
Activation function
Regularization, e.g., dropout
Nodes
Graph convolutions
Nodes
...
Output: Drugs C, D lead to a side effect r_2
Input

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

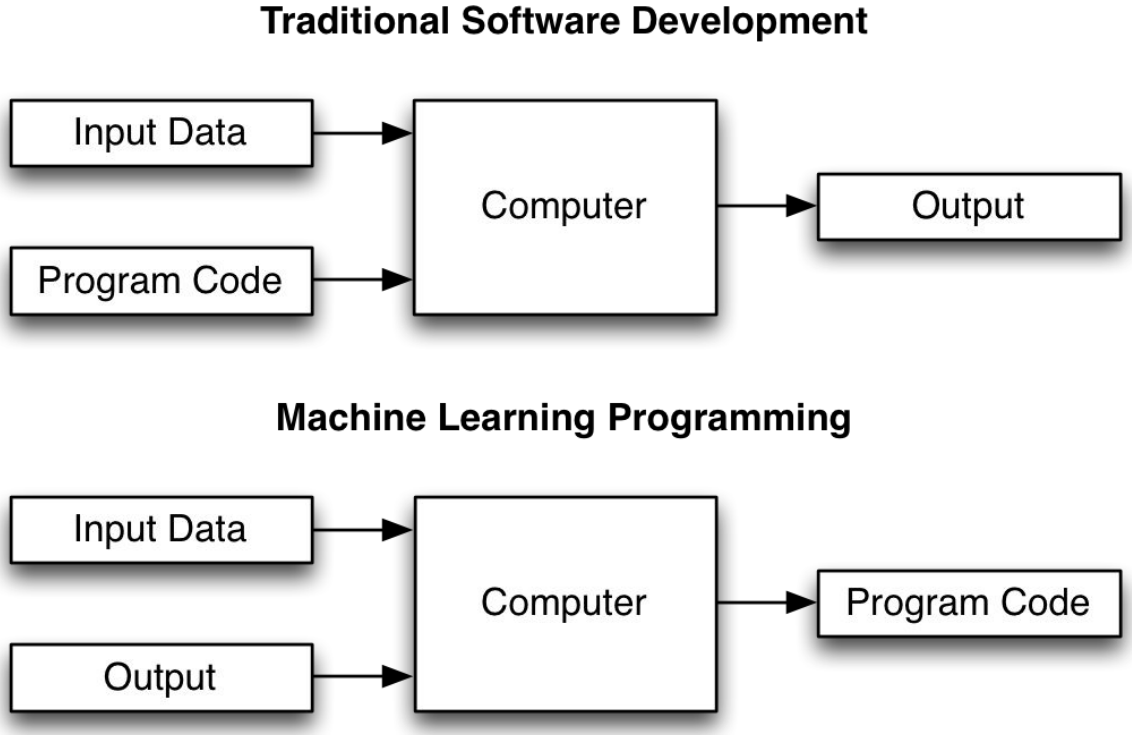
What is Machine Learning?

- 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 <P, T, E>

T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

Slide Credit: upenn, cis5190

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

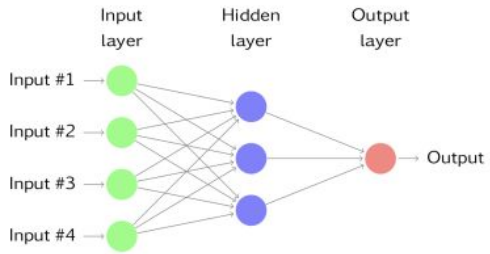
Dataset $\mathcal{D}$

Slide Credit: Cornell CS4780

Mathematical formulation of the pipeline

Dataset:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}, x_i \in \mathbb{R}^d, y_i \in \mathcal{C}(\text{ e.g., } \mathcal{C} = \{-1, 1\}), (x_i, y_i) \sim \mathcal{P}$$



Hypothesis:

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

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

Hypothesis class

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

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

Slide Credit: Cornell CS4780

Type of Y

- Binary Classification: Y = {0, 1}, Y = {-1, 1}
- Multi-Class Classification: Y = {1, 2, 3 .. K}
- Regression: Y = Real Numbers

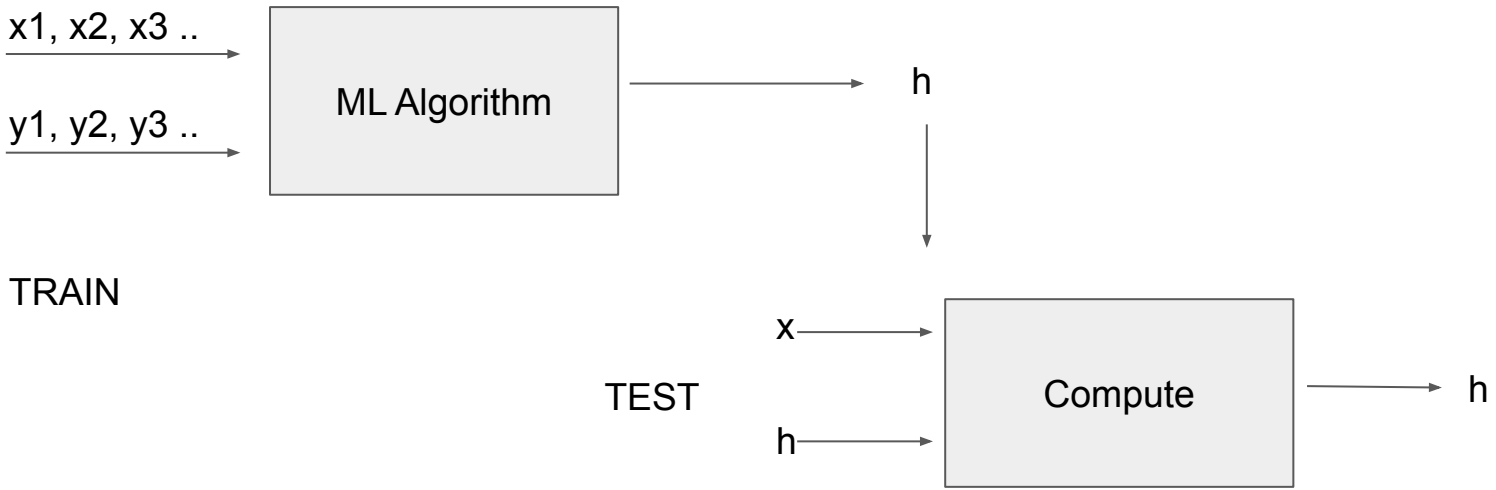
Slide Credit: Cornell CS4780

Type of X

- Patient Data
- Text Documents
- Images

Slide Credit: Cornell CS4780

Training



Slide Credit: Cornell CS4780

Learning

- Algorithm 1:
  - Pick h € H randomly
  - What is the problem?

- Algorithm 2:
  - Try every single h in H to find best one
  - What is the problem?

Slide Credit: Cornell CS4780

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:

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

Squared loss:

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

Slide Credit: Cornell CS4780

The Loss Function

Learning/Training

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

With loss ℓ being defined, we can perform training/learning:

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

Slide Credit: Cornell CS4780

Learning

- Algorithm 3:

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

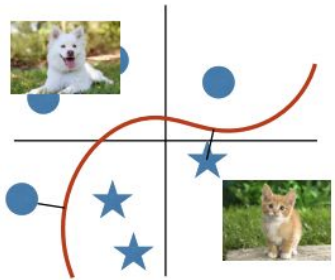
  - What is the problem?

Slide Credit: Cornell CS4780

Generalization

Dataset $\mathcal{D}$

ML model



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

Slide Credit: Cornell CS4780

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

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

Slide Credit: Cornell CS4780

Overfitting

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

Example

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

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

Slide Credit: Cornell CS4780

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

Slide Credit: Cornell CS4780

Overfitting

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

Example

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

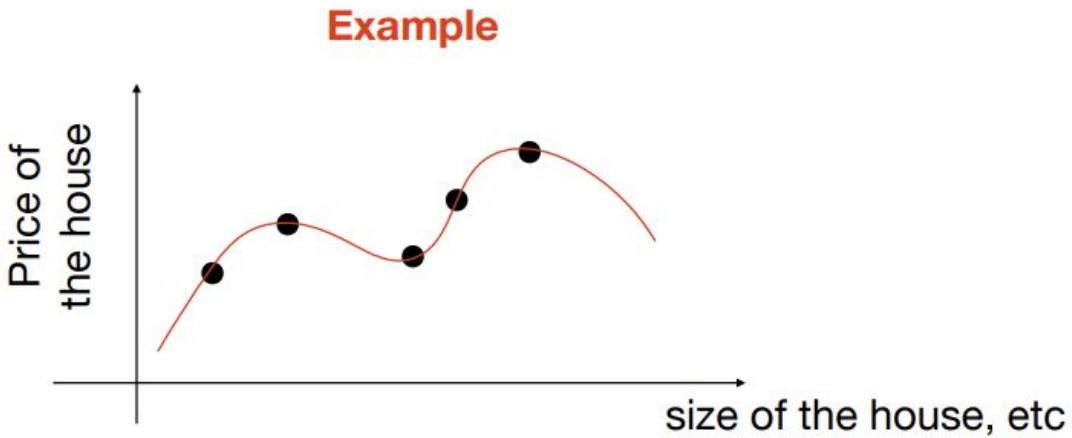
$$\tilde{h}(x) = \begin{cases} y_i & \exists (x_i, y_i) \in \mathcal{D} \text{ w/ } x_i = x \\ 0 & \text{else} \end{cases}$$

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

Slide Credit: Cornell CS4780

Overfitting

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

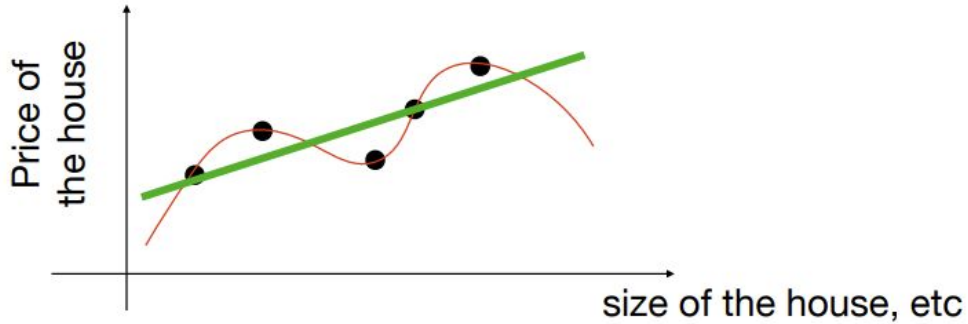


Slide Credit: Cornell CS4780

Overfitting

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

Example



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

Slide Credit: Cornell CS4780

Training, validation, and testing

Given a training dataset $\mathcal{D}$, we can split it into three sets:

$\mathcal{D}_{TR}$: training set

$\mathcal{D}_{VA}$: validation set

$\mathcal{D}_{TE}$: test set

Before training/learning, we often **randomly** split it with size proportional to 80% / 10% / 10%

Slide Credit: Cornell CS4780

Selecting models using validation set

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

Small avg error on $\mathcal{D}_{TR}$ but larger avg error on $\mathcal{D}_{VA}$ indicates overfitting

Revise model on $\mathcal{D}_{TR}$ (e.g., add regularization, change neural network structures, etc)

Slide Credit: Cornell CS4780

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:

$$\frac{1}{|\mathcal{D}_{TF}|} \sum_{x,y \in \mathcal{D}_{TE}} \ell(\hat{h}, x, y) \approx \mathbb{E}_{x,y \sim \mathcal{P}}[\ell(\hat{h}, x, y)]$$

(Due to law of large numbers)

Slide Credit: Cornell CS4780

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!)

K-fold cross validation

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

For $i = 1 \rightarrow K$:

- $\mathcal{D}_{TR}$: all others folds except the i'th fold
- Train model $\hat{h}$ on D_{TR}
- Validate on the i'th fold (i.e., $\mathcal{D}_{VR}$ = i'th fold)

Average K validation errors

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

Slide Credit: Cornell CS4780

Summary

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

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

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

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

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

Slide Credit: Cornell CS4780

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".
- **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'".

Slide Credit: https://developers.google.com/machine-learning/guides/rules-of-ml
https://christophm.github.io/interpretable-ml-book/terminology.html

Machine Learning Terminology

- **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.

Slide Credit: https://developers.google.com/machine-learning/guides/rules-of-ml

Next Class:

Continue with Machine Learning Concepts & KNN