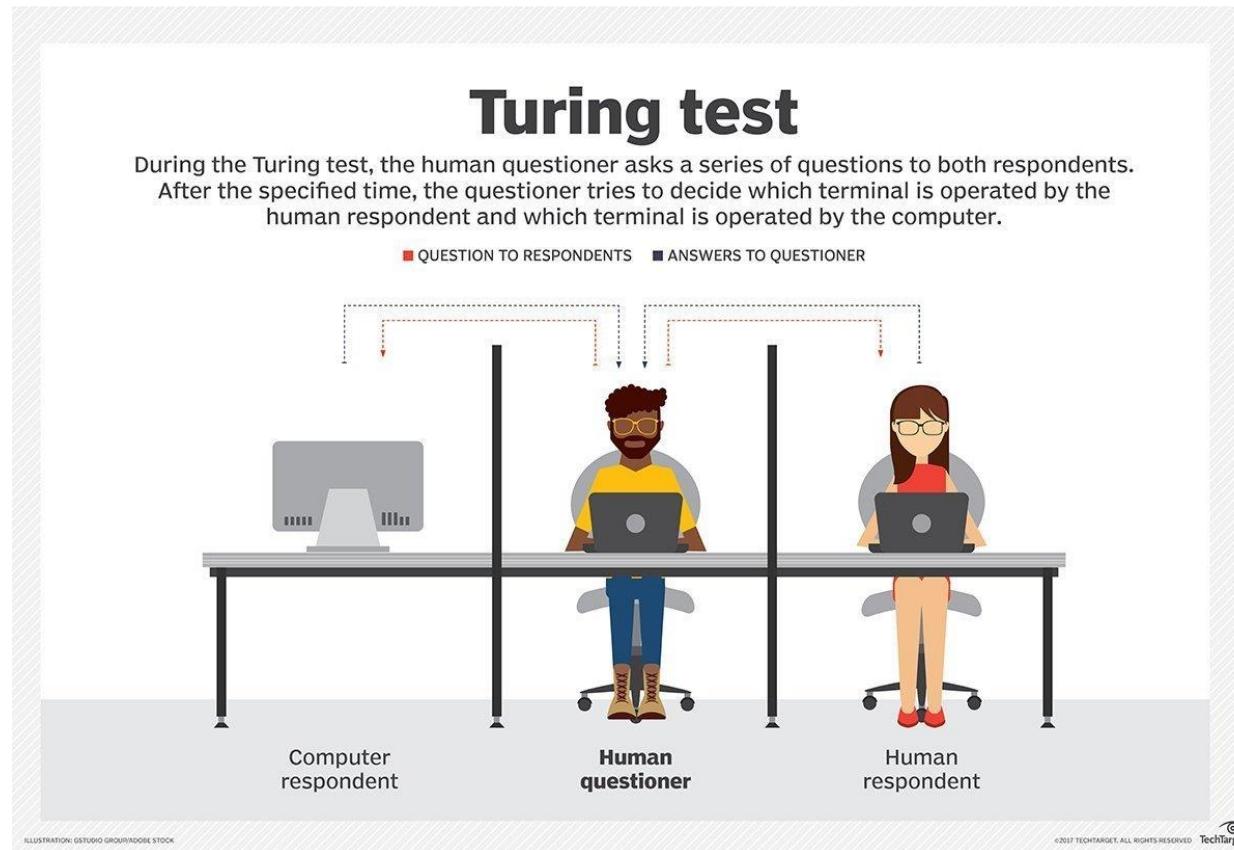


# Introduction

**Qiang Sun**  
**University of Toronto**

# Turing Test (1950)

- Alan Turing, *Computing Machinery and Intelligence* (1950)
  - I propose to consider the question, “Can machines think?”



## I.—COMPUTING MACHINERY AND INTELLIGENCE

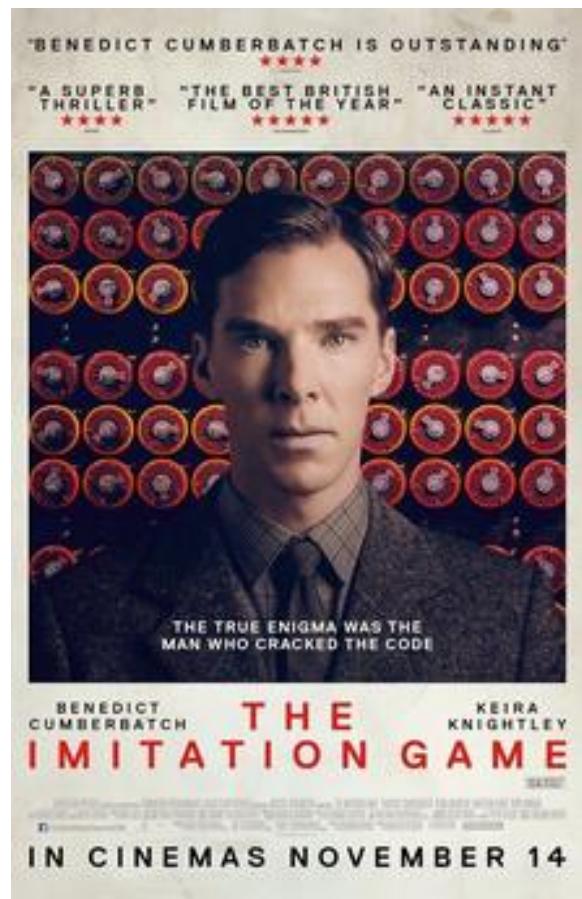
By A. M. TURING

### 1. *The Imitation Game.*

I PROPOSE to consider the question, ‘Can machines think ?’ This should begin with definitions of the meaning of the terms ‘machine’ and ‘think’. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words ‘machine’ and ‘think’ are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, ‘Can machines think ?’ is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed



# Alan Turing



Turing Award 2019

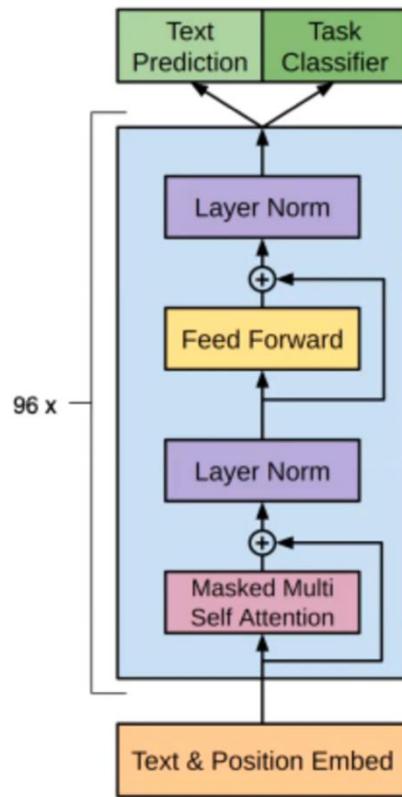
Yann LeCun, Geoffrey Hinton, Yoshua Bengio



# Recent Development: OpenAI GPT Series

## GPT = Generative Pre-trained Transformer

- state-of-the-art language processing artificial intelligence system developed by OpenAI



Model Name	$n_{\text{params}}$	$n_{\text{layers}}$	$d_{\text{model}}$	$n_{\text{heads}}$	$d_{\text{head}}$	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	$6.0 \times 10^{-4}$
GPT-3 Medium	350M	24	1024	16	64	0.5M	$3.0 \times 10^{-4}$
GPT-3 Large	760M	24	1536	16	96	0.5M	$2.5 \times 10^{-4}$
GPT-3 XL	1.3B	24	2048	24	128	1M	$2.0 \times 10^{-4}$
GPT-3 2.7B	2.7B	32	2560	32	80	1M	$1.6 \times 10^{-4}$
GPT-3 6.7B	6.7B	32	4096	32	128	2M	$1.2 \times 10^{-4}$
GPT-3 13B	13.0B	40	5140	40	128	2M	$1.0 \times 10^{-4}$
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	$0.6 \times 10^{-4}$

<https://dzlab.github.io/ml/2020/07/25/gpt3-overview/>



# Recent Development: ChatGPT

Step 1

**Collect demonstration data and train a supervised policy.**

A prompt is sampled from our prompt dataset.

Explain reinforcement learning to a 6 year old.

A labeler demonstrates the desired output behavior.

We give treats and punishments to teach...

This data is used to fine-tune GPT-3.5 with supervised learning.

SFT  
  
📄📄📄

Step 2

**Collect comparison data and train a reward model.**

A prompt and several model outputs are sampled.

Explain reinforcement learning to a 6 year old.

A  
In reinforcement learning, the agent is...  
B  
Explain rewards...  
C  
In machine learning...  
D  
We give treats and punishments to teach...

A labeler ranks the outputs from best to worst.

D > C > A > B

This data is used to train our reward model.

RM  
  
D > C > A > B

Step 3

**Optimize a policy against the reward model using the PPO reinforcement learning algorithm.**

A new prompt is sampled from the dataset.

Write a story about otters.

PPO  


The PPO model is initialized from the supervised policy.

The policy generates an output.

Once upon a time...

RM  


The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

$r_k$

# Recent Development: ChatGPT

Please write a 150-word essay about artificial intelligence and machine learning.

Artificial intelligence (AI) is the broad field of building computer systems that can perform tasks we usually associate with human intelligence, such as understanding language, recognizing images, planning actions, and making decisions. Machine learning (ML) is a major approach within AI that enables systems to learn patterns from data instead of relying only on hand-written rules. For example, an ML model can study thousands of medical scans to help detect disease, or analyze past purchases to recommend products. Recent breakthroughs, especially in deep learning, have made AI tools far more capable, but they also raise important questions about privacy, bias, and accountability. Because models learn from historical data, they can absorb and amplify existing inequalities unless carefully designed and tested. Used responsibly, AI and ML can improve healthcare, education, and transportation, while automating repetitive work. The challenge is to pair technical innovation with thoughtful governance so these powerful technologies serve people fairly and safely.



# GPT starts to have reasonable mathematical reasoning skills

## On Learning-Curve Monotonicity for Maximum Likelihood Estimators

Mark Sellke

Steven Yin

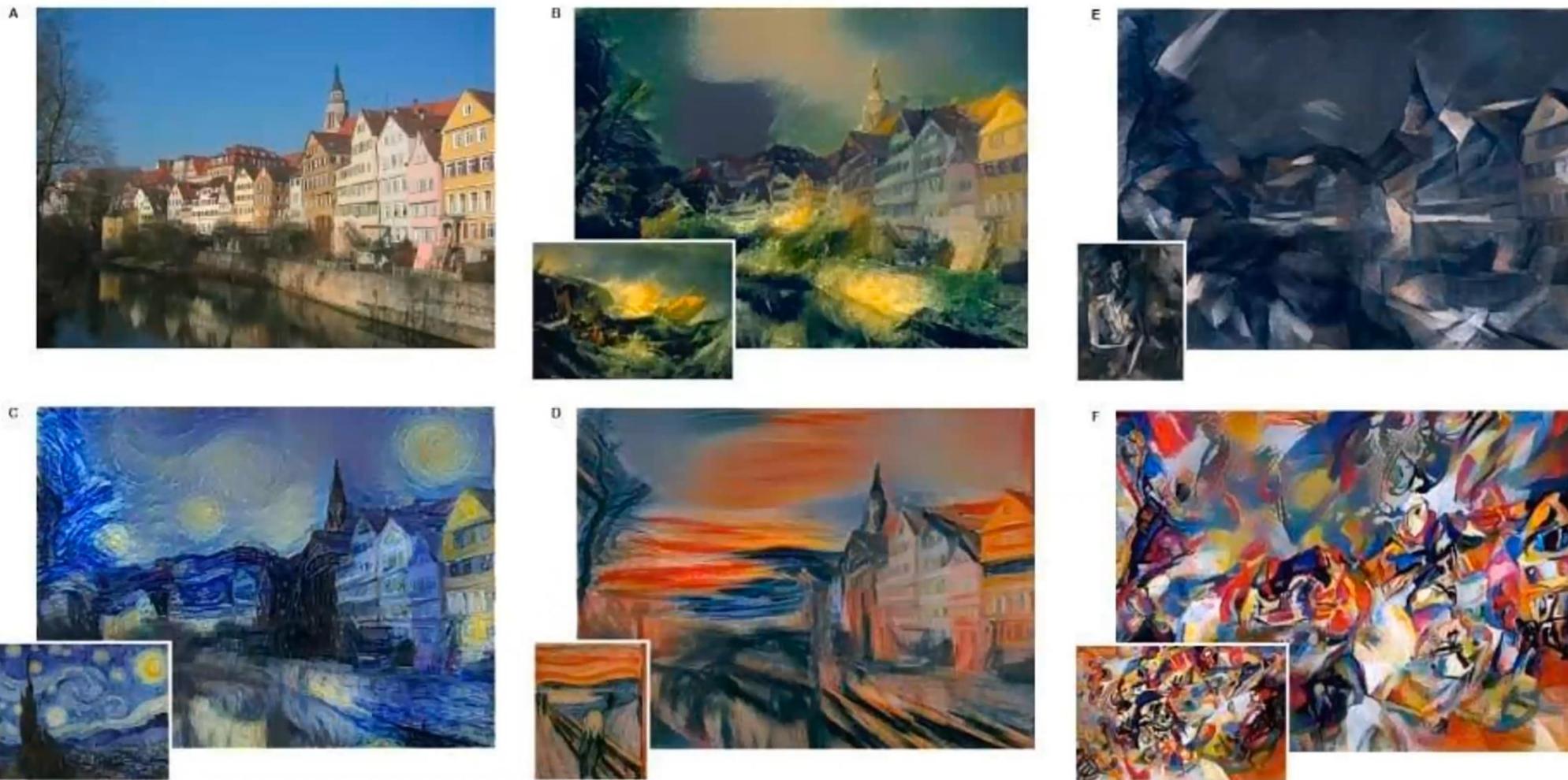
### Abstract

The property of learning-curve monotonicity, highlighted in the recent papers [VML19, LVM19, VL22], describes algorithms which only improve in average performance given more data, for any underlying data distribution within a given family. We establish the first nontrivial monotonicity guarantees for the maximum likelihood estimator in a variety of well-specified parametric settings. For sequential prediction with log loss, we show monotonicity (in fact complete monotonicity) of the forward KL divergence for Gaussian vectors with unknown covariance and either known or unknown mean, as well as for Gamma variables with unknown scale parameter. The Gaussian setting was explicitly highlighted as open in the aforementioned works, even in dimension 1. Finally we observe that for reverse KL divergence, a folklore majorization trick from [MP65] yields monotonicity for very general exponential families.

All results in this paper were derived by variants of GPT-5.2 Pro. Humans did not provide any proof strategies or intermediate arguments, but only prompted the model to continue developing additional results, and verified and transcribed its proofs.



# Recent Development: Image Generation



<http://www.dailymail.co.uk/sciencetech/article-3214634/The-algorithm-learn-copy-artist-Neural-network-recreate-snaps-style-Van-Gogh-Picasso.html>

# Recent Development: Stable Diffusion

## Stable Diffusion



An image generated by Stable Diffusion based on the text prompt "a photograph of an astronaut riding a horse"

**Developer(s)**

StabilityAI

**Initial release**

August 22, 2022

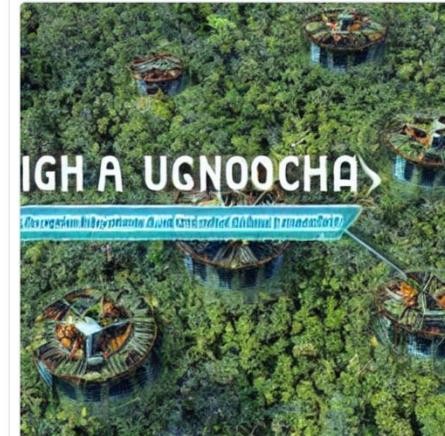
## Stable Diffusion Demo

Stable Diffusion is a state of the art text-to-image model that generates images from text.

For faster generation and API access you can try [DreamStudio Beta](#)

A high tech solarpunk utopia in the Amazon rainforest

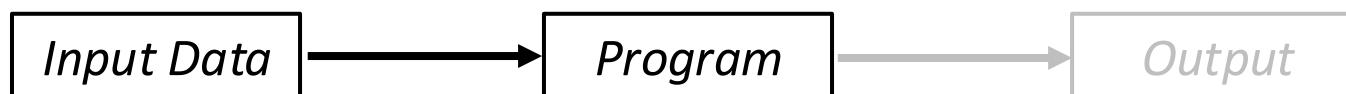
Generate image



# Machine Learning

- One of the major goals and concepts in Artificial Intelligence.
- The study of computer algorithms that improve **automatically** by the use of **data**.
  - Instead of specifically programming them how to solve the task

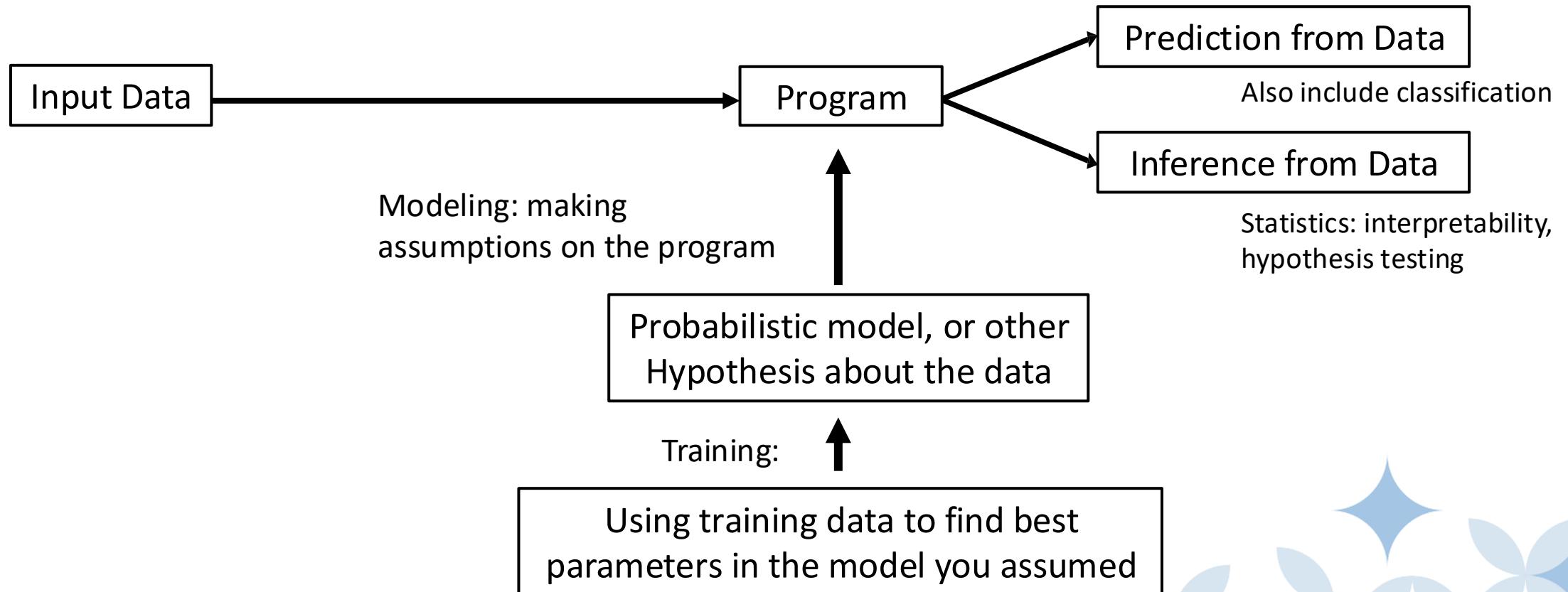
- Traditional Programming: get an output from the input and program



- Machine Learning: learn the program that links the input and output

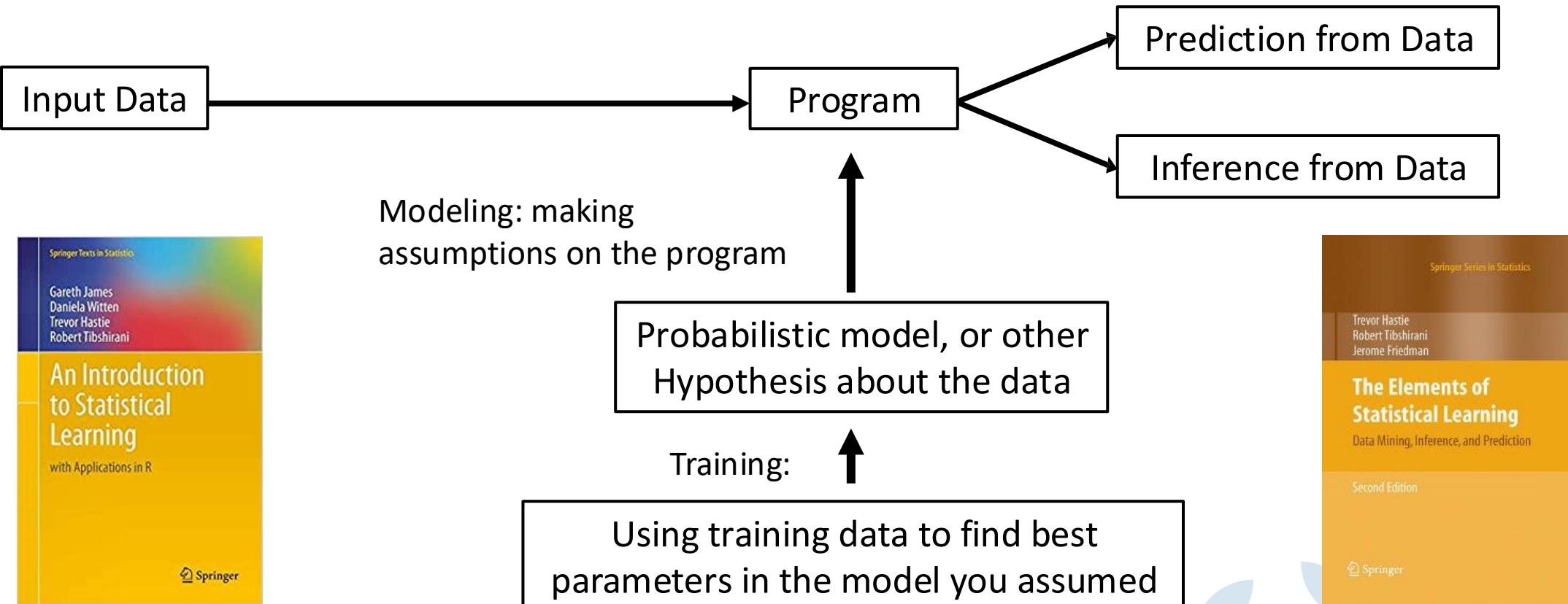


# Statistical Machine Learning – Classical ML

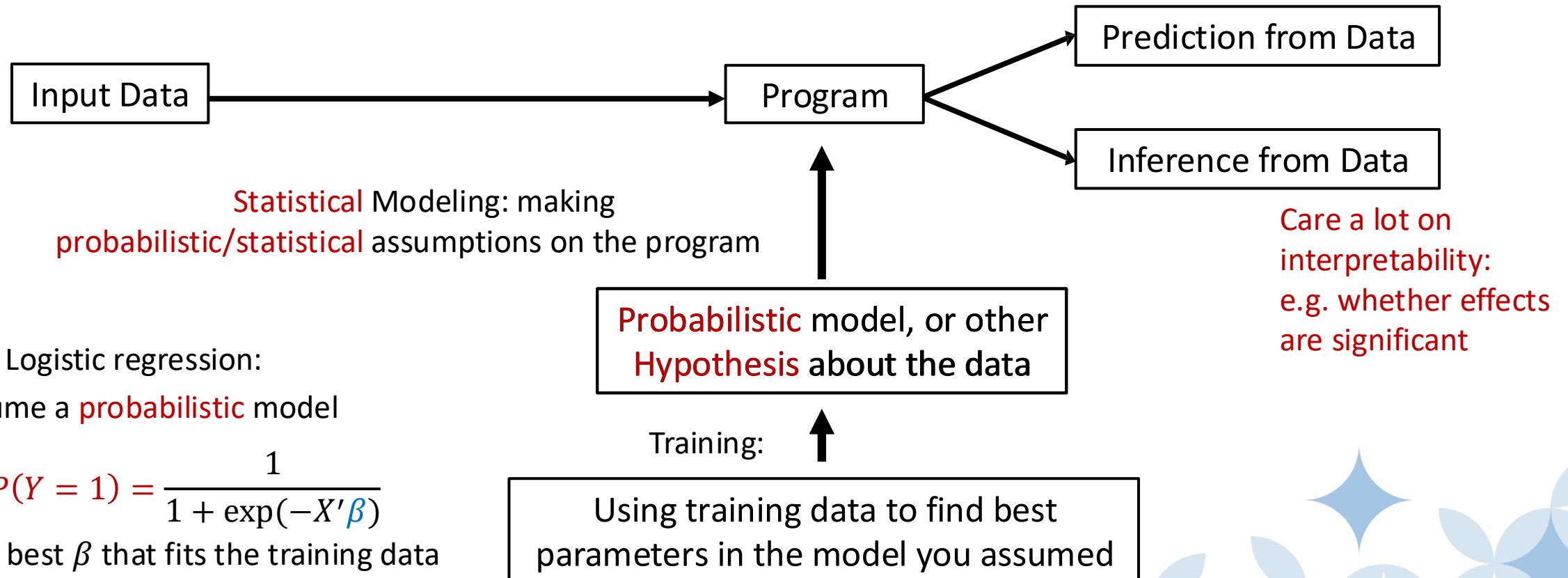


# Statistical Machine Learning – Classical ML

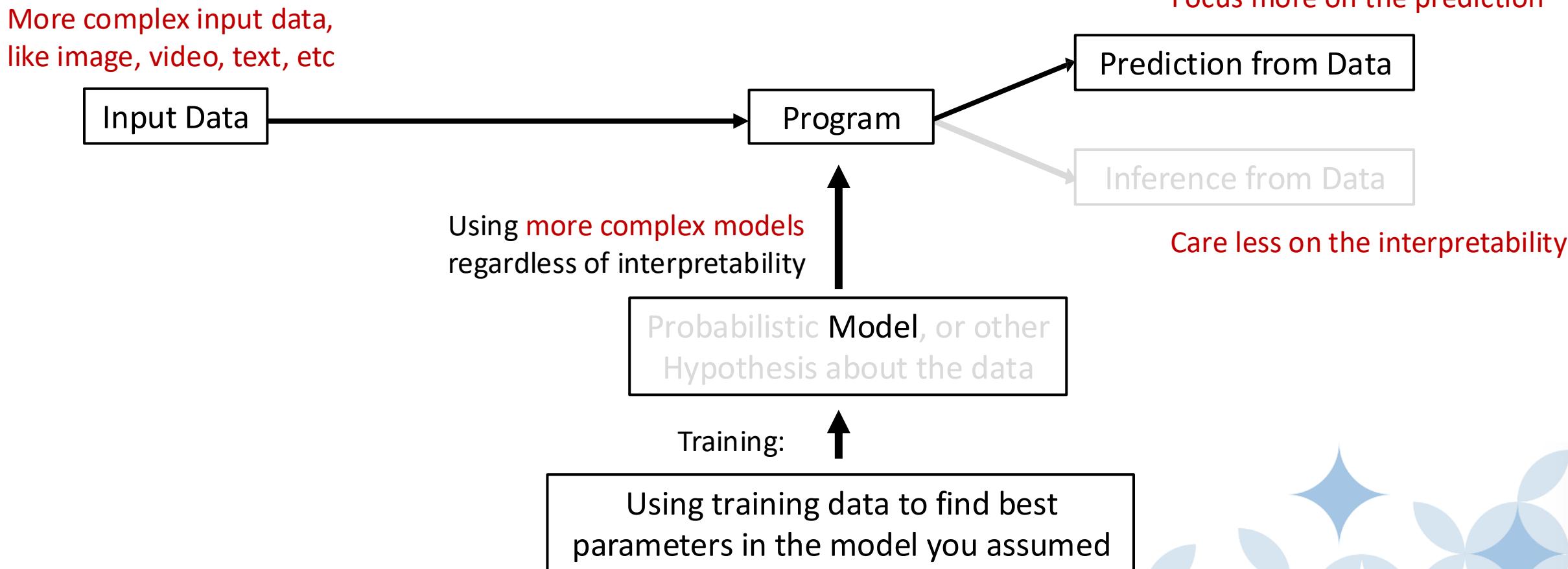
- Classical Machine Learning and Data Mining;



# Statistical Machine Learning – Classical ML



# Statistical Machine Learning – Classical ML



# Statistical Machine Learning – Classical ML

More complex input data,  
like image, video, text, etc

Input Data

Spend more efforts in feature engineering to  
capture useful information from complex data

Feature Extraction

Program

Focus more on the prediction

Prediction from Data

Inference from Data

Using **more complex models**  
regardless of interpretability

Probabilistic Model, or other  
Hypothesis about the data

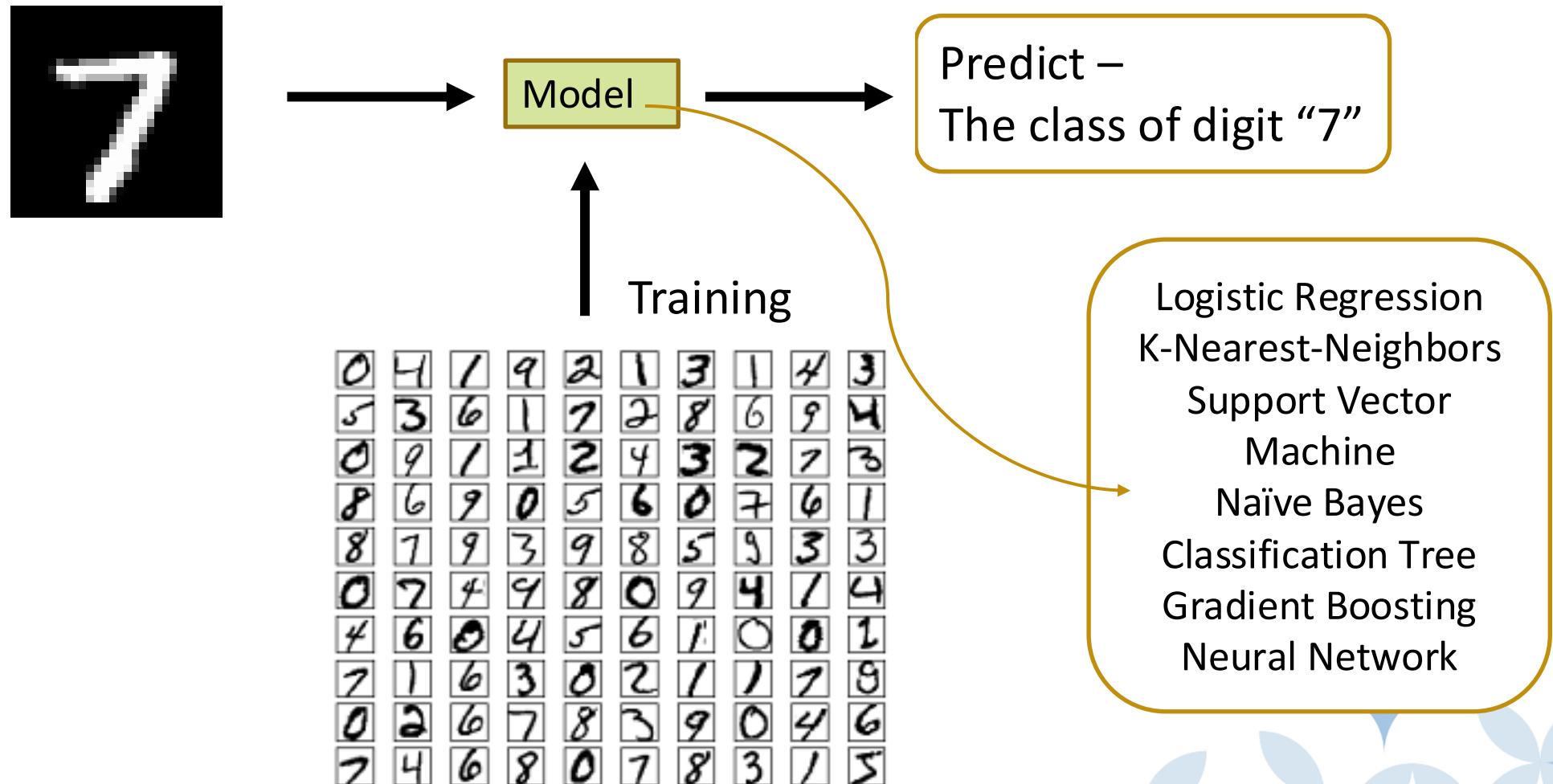
Care less on the interpretability

Training:

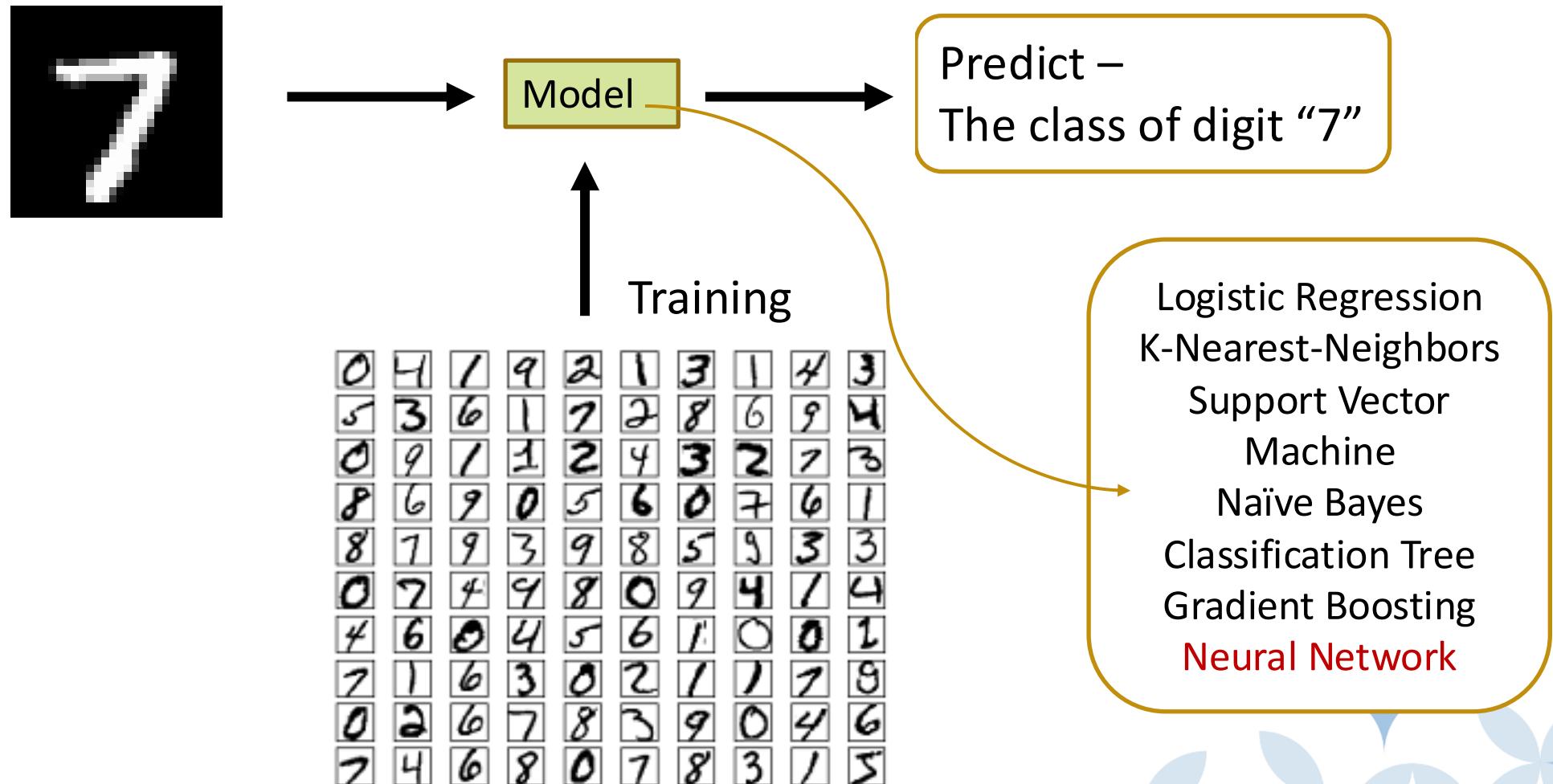
Using training data to find best  
parameters in the model you assumed



# ML on image data

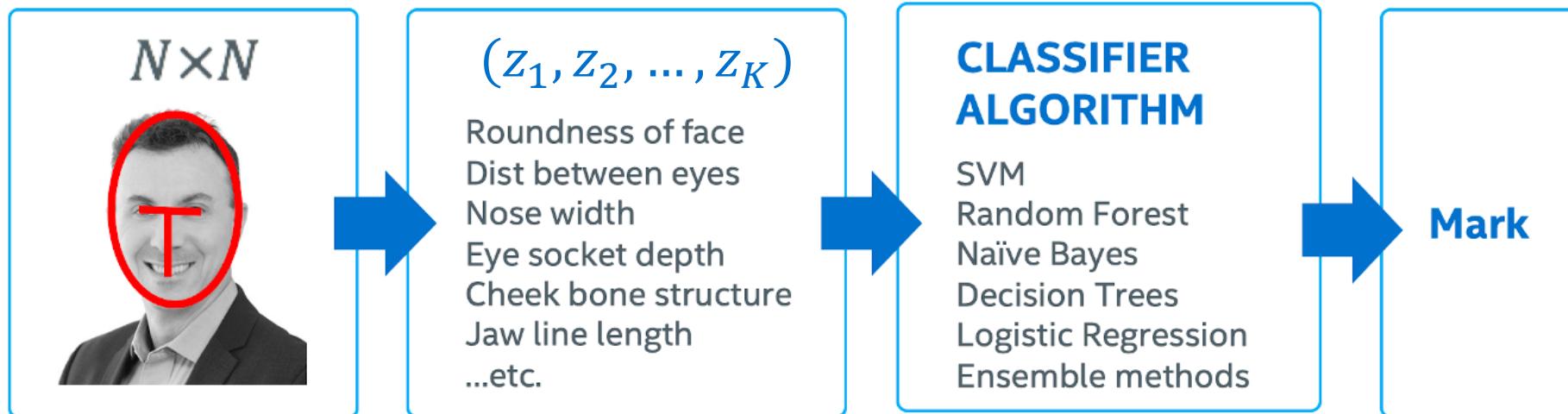


# ML on image data - Deep Learning



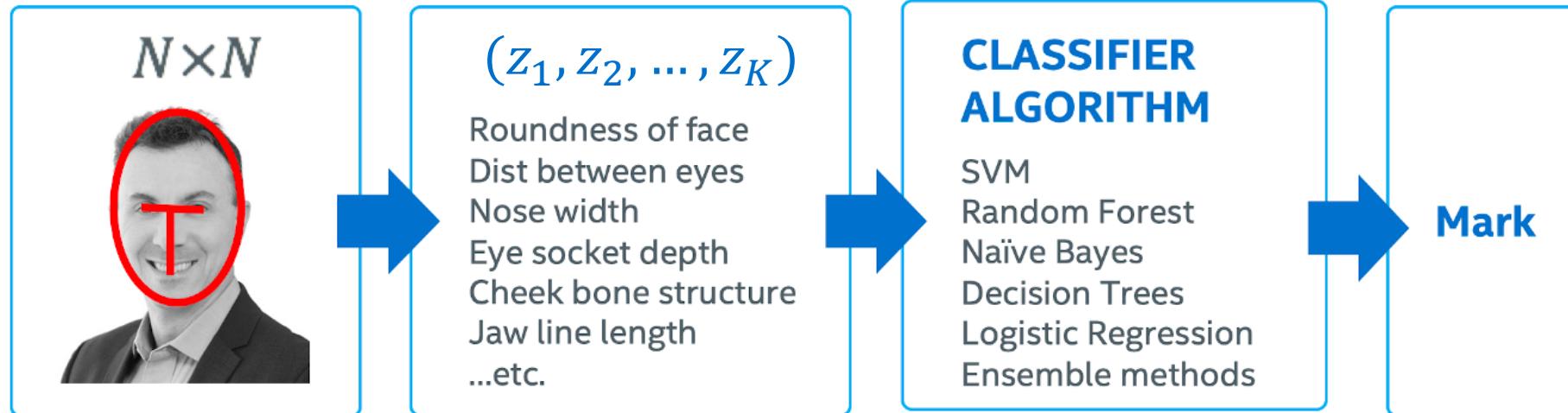
# ML with feature engineering

- Face recognition: Input Data is more complex

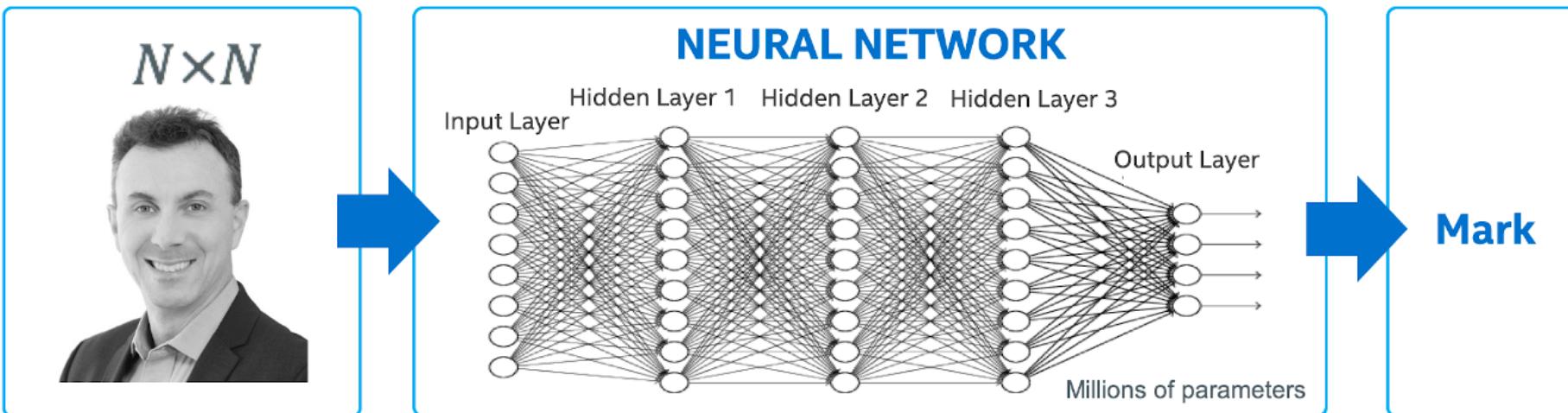


# DL vs. Classical ML

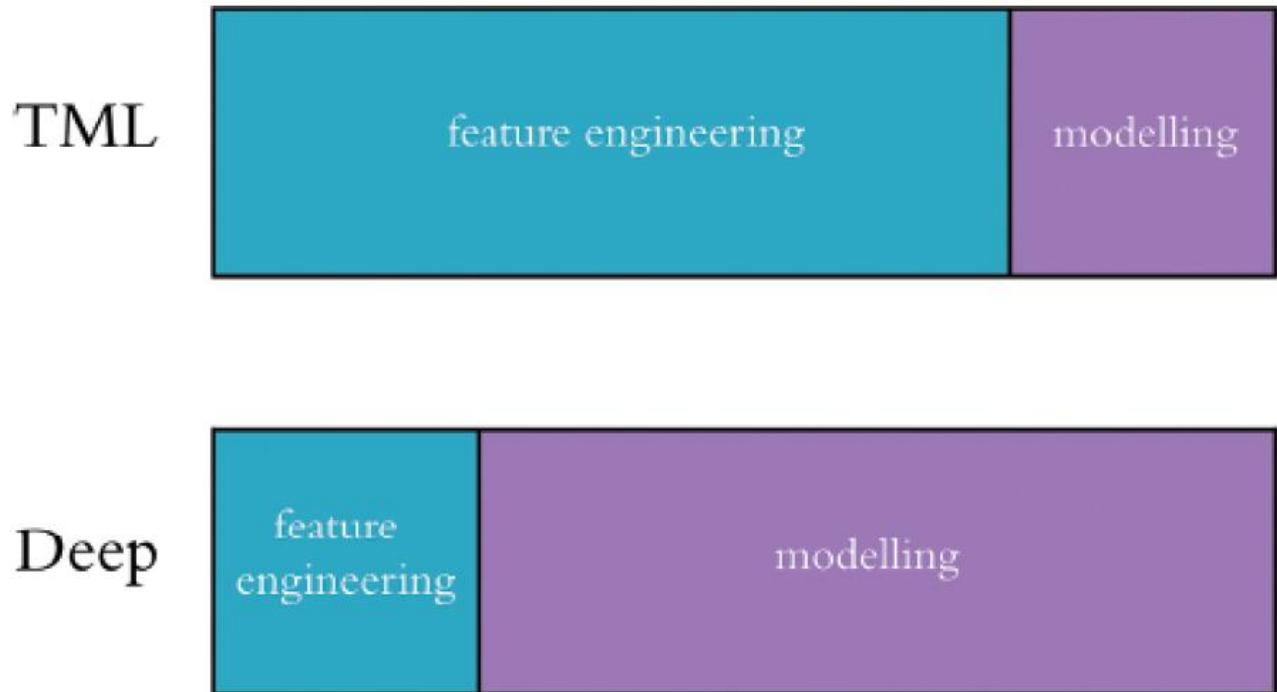
## Classic Machine Learning



## Deep Learning



# DL vs. Classical ML



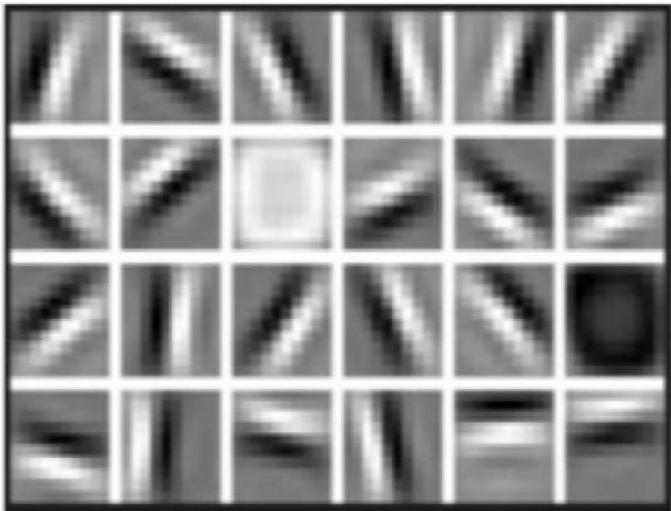
**Figure 1-13** Feature engineering—the transformation of raw data into thoughtfully-transformed input variables—often predominates the application of traditional machine learning algorithms. In contrast, the application of deep learning often involves little to no feature engineering, with the majority of time spent instead on the design and tuning of model architectures.



# DL vs. Classical ML

- The idea of deep learning is the ability of the multi-layer neural networks to learn *hierarchical* representations of features

**Low Level Features**



Lines & Edges

**Mid Level Features**

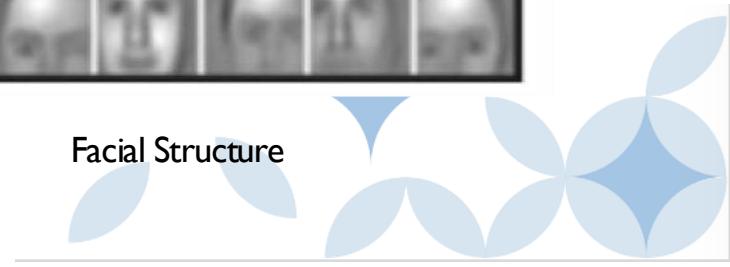


Eyes & Nose & Ears

**High Level Features**



Facial Structure



# ARTIFICIAL INTELLIGENCE

IS NOT NEW

## ARTIFICIAL INTELLIGENCE

Any technique which enables computers to mimic human behavior



1950's

1960's

1970's

1980's

## MACHINE LEARNING

AI techniques that give computers the ability to learn without being explicitly programmed to do so



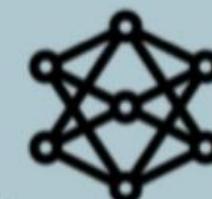
1990's

2000's

2010s

## DEEP LEARNING

A subset of ML which make the computation of multi-layer neural networks feasible



# The Rise of Deep Learning

## 'Deep Voice' Software Can Clone Anyone's Voice With Just 3.7 Seconds of Audio

Nvidia is processing

Using snippets of voices, Baidu's 'Deep Voice' can generate new speech, accents, and tones.

DEAN TAKAHASHI | DOI | SHARE | [F](#) [T](#) [W](#)

'Creative' Alpha chess computer



Former chess world champion sees of computers that could b



Stock Predictions Based On AI: Is the Market Truly Predictable?



Complex of bacteria-infesting viral proteins modeled in CASP 13. The complex cont that were modeled individually. PROTEIN DATA BANK

Google's DeepMind aces protein folding

By Robert F. Service | Dec. 6, 2018, 12:05 PM

## DEEPMIND E STARCRAFT TRIUMPH FO



How an A.I. 'Cat-and-Mouse Game' Generates Believable Fake Photos

By CADE METZ and KEITH COLLINS | JAN. 2, 2018



After Millions of Trials, These Simulated Humans Learned to Do Perfect Backflips and Cartwheels



Researchers introduce a deep learning method that converts mono audio recordings into 3D sounds using video scenes

Let There Be Sight: How Deep Learning Is Helping the Blind 'See'



## Technology outpacing security measures

Facial Recognition | Features and Interviews



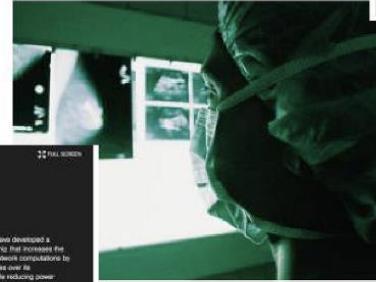
Neural networks everywhere

New chip reduces neural networks' power consumption by up to 95 percent, making them practical for battery-powered devices.



## AI beats docs in cancer spotting

A new study provides a fresh example of machine learning as an important diagnostic tool. Paul Biegler reports.



These faces show how far AI image generation has come in just four years



## AI Can Help In Predicting Cryptocurrency Value

By Eric Johnson | Jan. 23, 2019



## Automation And Algorithms: De-Risking Manufacturing With Artificial Intelligence

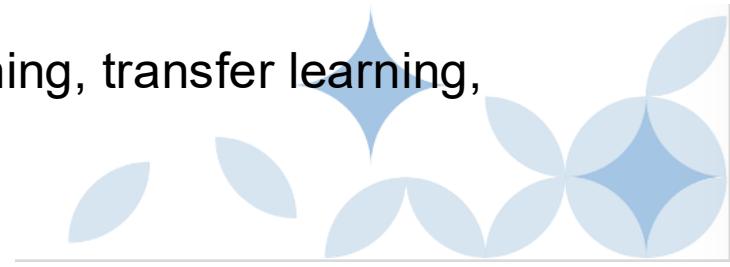
Sarah Goehrke Contributor [@SarahGoehrke](#)  
Manufacturing  
I focus on the industrialization of additive manufacturing

TWEET THIS

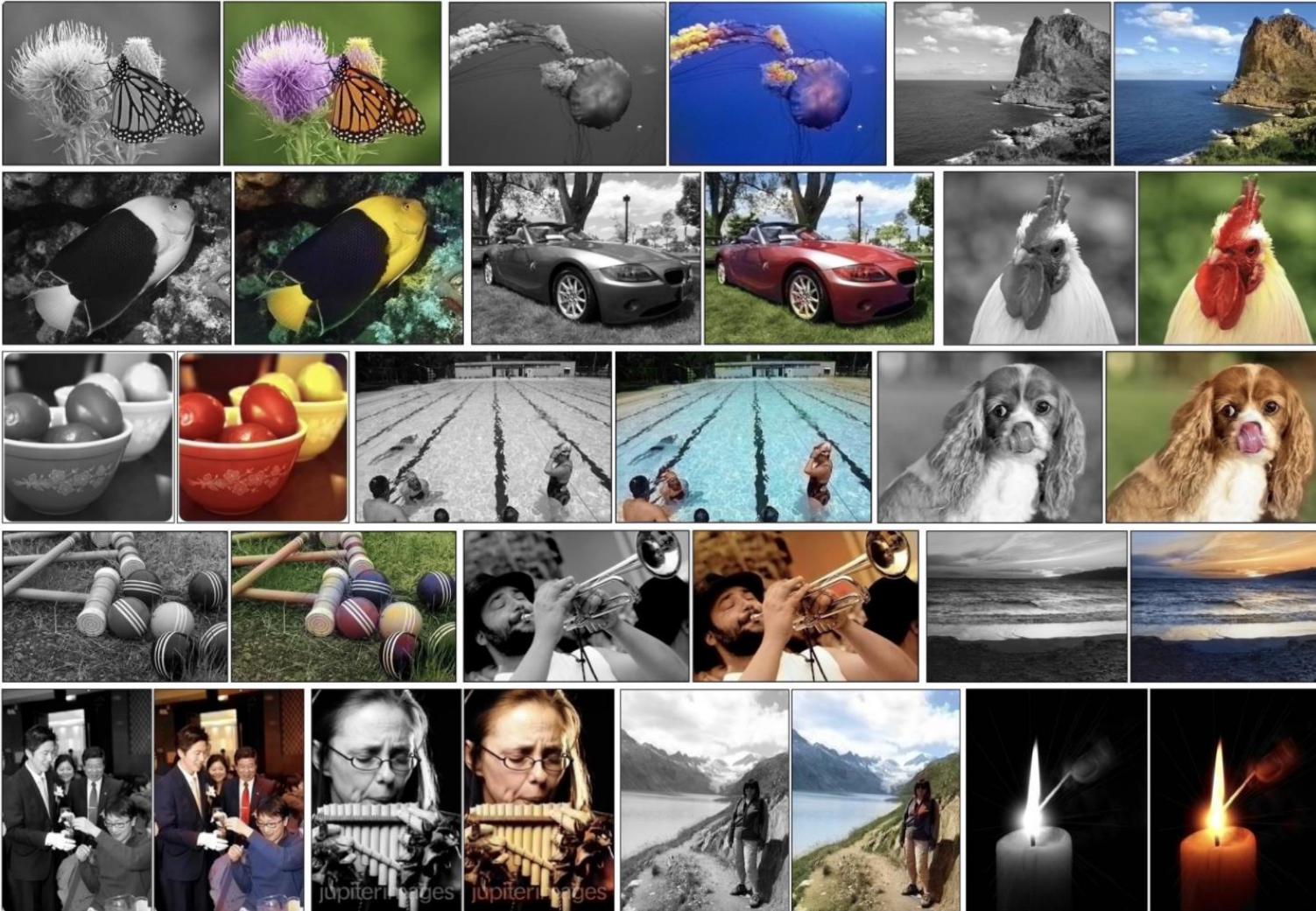
The two key applications of AI in manufacturing are pricing and manufacturability feedback

# Deep learning success

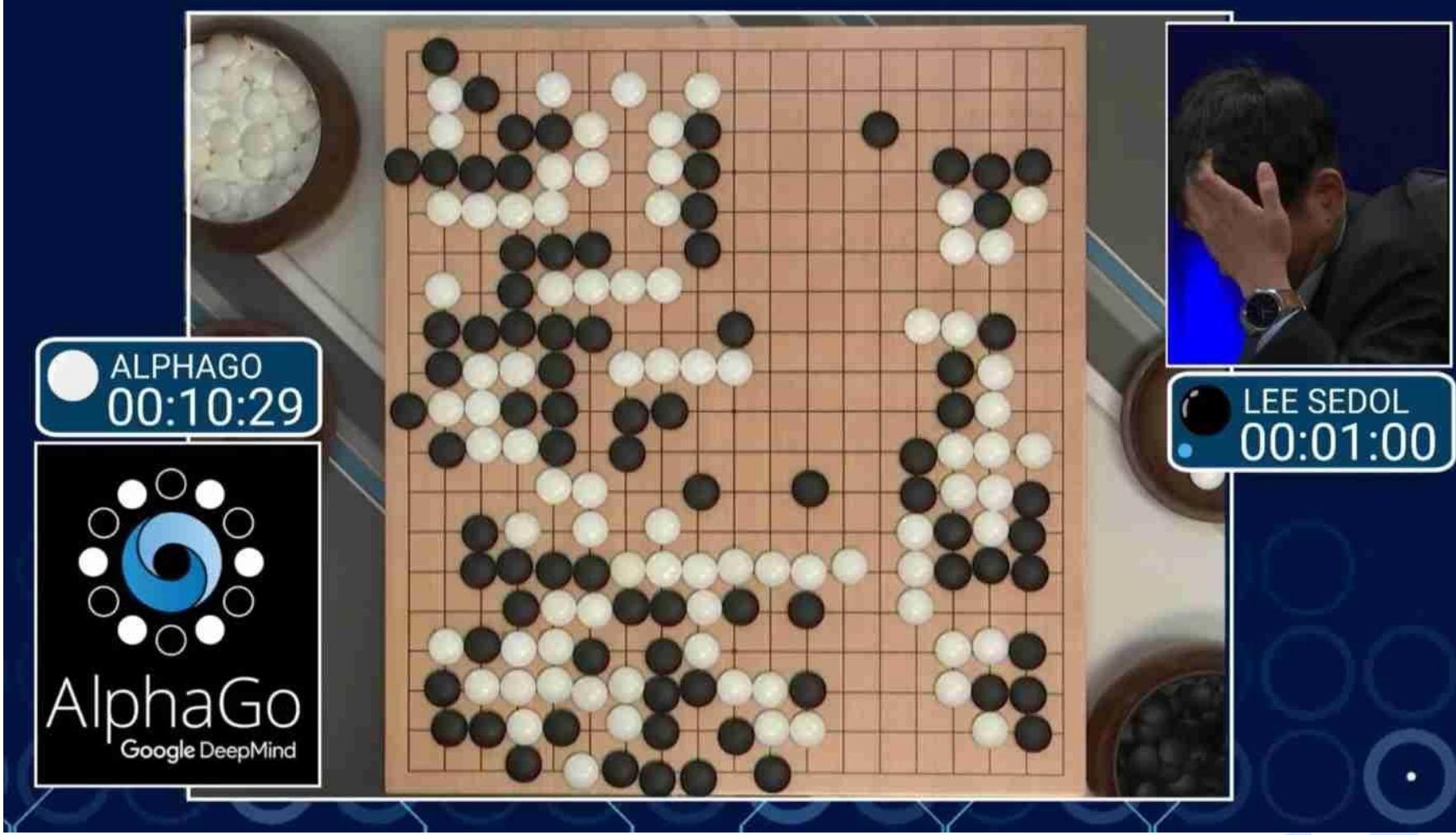
- Speech, Images, Video
  - Simple tasks (classification, localization) almost done on super-human level
  - Complex tasks (semantic segmentation, style transfer, image generation, ...) are in progress with exciting examples
- Text/Natural Language Processing (NLP) [a bit behind speech/images]
  - Some good basic technologies are in use (word embeddings, machine translation, etc)
  - Even more in progress (text generation, Q&A, etc)
- Reinforcement Learning (RL)
  - Great achievements exist: Go playing, Atari playing, more to come in business
- A lot of other research
  - One-shot learning (very few data), multi-modal and multi-task learning, transfer learning, unsupervised learning, and many many more



# Example: Image Colorization

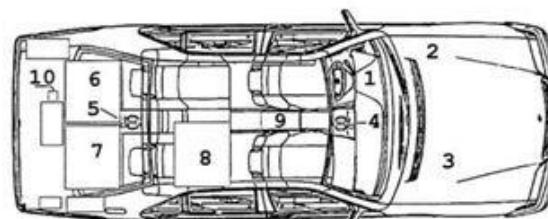


# Example: Game of Go: Computer-Human 4:1



# Example: Self-driving Cars

- Almost all automobile industry as well as tech giants (Google, Apple, NVidia, Uber, Tesla, Volvo, Mercedes-Benz — pioneered by Ernst Dickmanns in 1980s) develop their own autonomous car.
- Automobiles will become really **auto**-mobile.



1 electrical steering motor  
2 electrical brake control  
3 electronic throttle  
4 front pointing platform for CCD-cameras  
5 rear pointing platform  
6 Transputer Image Processing system  
platform and vehicle controllers  
7 electronics rack, human interface  
8 accelerometers (3orthogonal)  
9 inertial rate sensors

$f = 24 \text{ mm}$        $f = 7.5 \text{ mm}$   
 $15^\circ$  Tele      Wide  $46^\circ$   
angle

At distance  $L_s \sim 20 \text{ m} (\sim 60 \text{ m})$ ,  
the resolution is 5 cm/pixel



# Example: Image generation by text

this small bird has a pink breast and crown, and black primaries and secondaries.



this magnificent fellow is almost all black with a red crest, and white cheek patch.



the flower has petals that are bright pinkish purple with white stigma



this white and yellow flower have thin white petals and a round yellow stamen



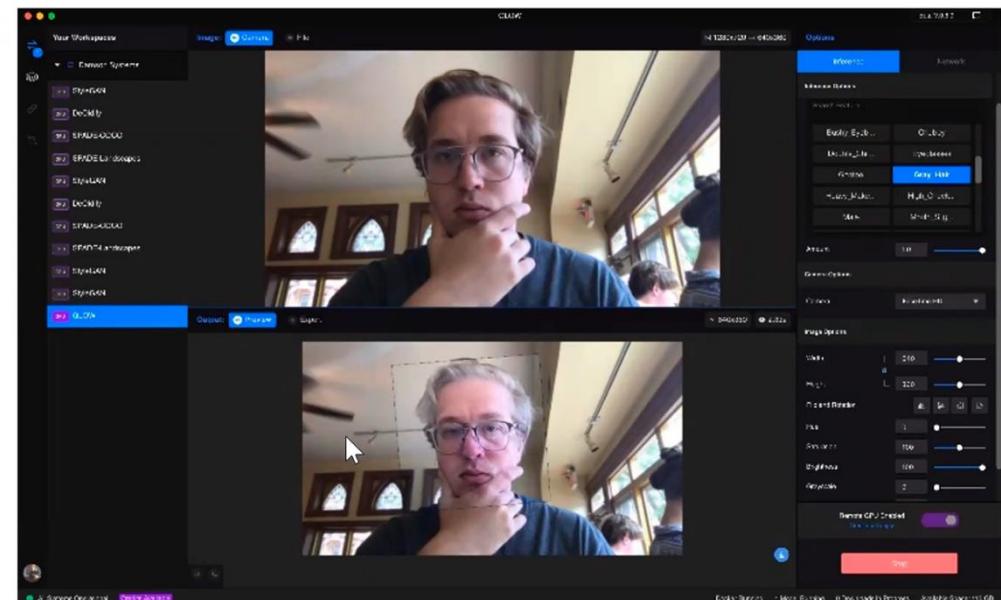
runwayml

Stable Diffusion

Figure 1. Examples of generated images from Left: captions are from zero-shot (held out text. Right: captions are from the training set



Generative Adversarial  
Text to Image Synthesis

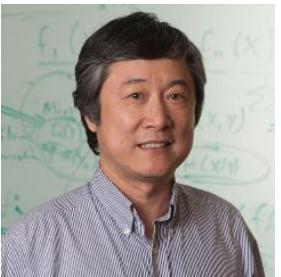


# Deep Learning -> Business and Industry Transformation

- Sales
  - “Next product to buy” recommendations
- Banking
  - Automated Loan Processing Systems
- Retail: faster and better shopping services
  - Robotic assistants in retail stores
  - Use of biometrics (facial recognition, fingerprints)
  - AI based predictive analytics on demand
- Health Care
  - AI diagnostics via analysis of medical data
  - Robot-assisted surgery



# Hype of ML / DL in Financial Industries



Li Deng  
Microsoft



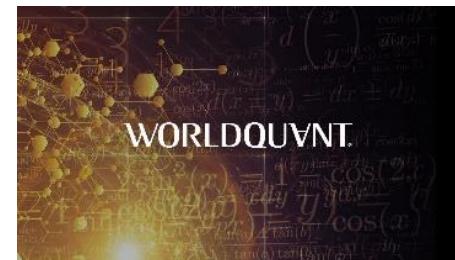
Pedro Domingos  
U of Washington



Manuela Veloso  
Carnegie Mellon



Yoram Singer  
Princeton & Google



# Potential Economic Impact

- Productivity growth
  - Substitution, augmentation & contributions to labor productivity
- AI adoption could raise global GDP by \$13T by 2030
  - 1.2% of additional GDP growth per year

Jobs displaced  
by 2030

**400-800 mil**

Jobs created  
by 2030

**555-890 mil**

[Source: McKinsey Global Institute.]

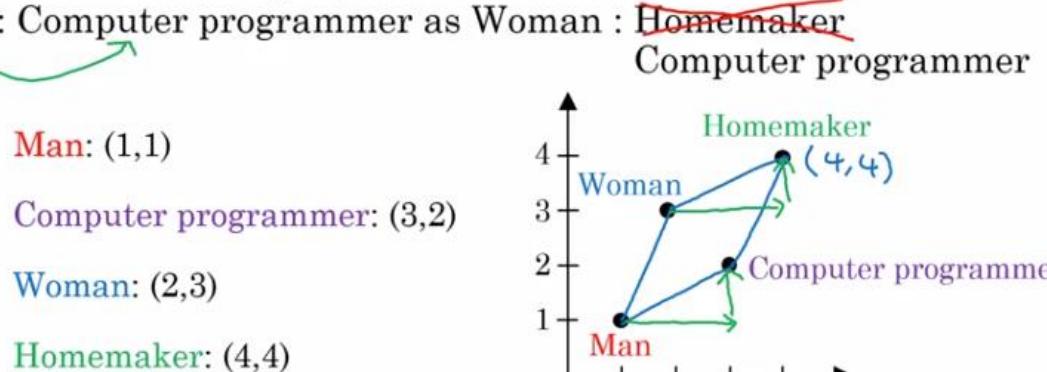


# Danger of AI and ML: Discrimination & Fairness

## Discrimination and Bias: “man is to king as woman is to x”

AI learning unhealthy stereotypes

- Man : Woman as Father : Mother
- Man : Woman as King : Queen
- Man : Computer programmer as Woman : ~~Homemaker~~



[Bolukbasi et al. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings.]

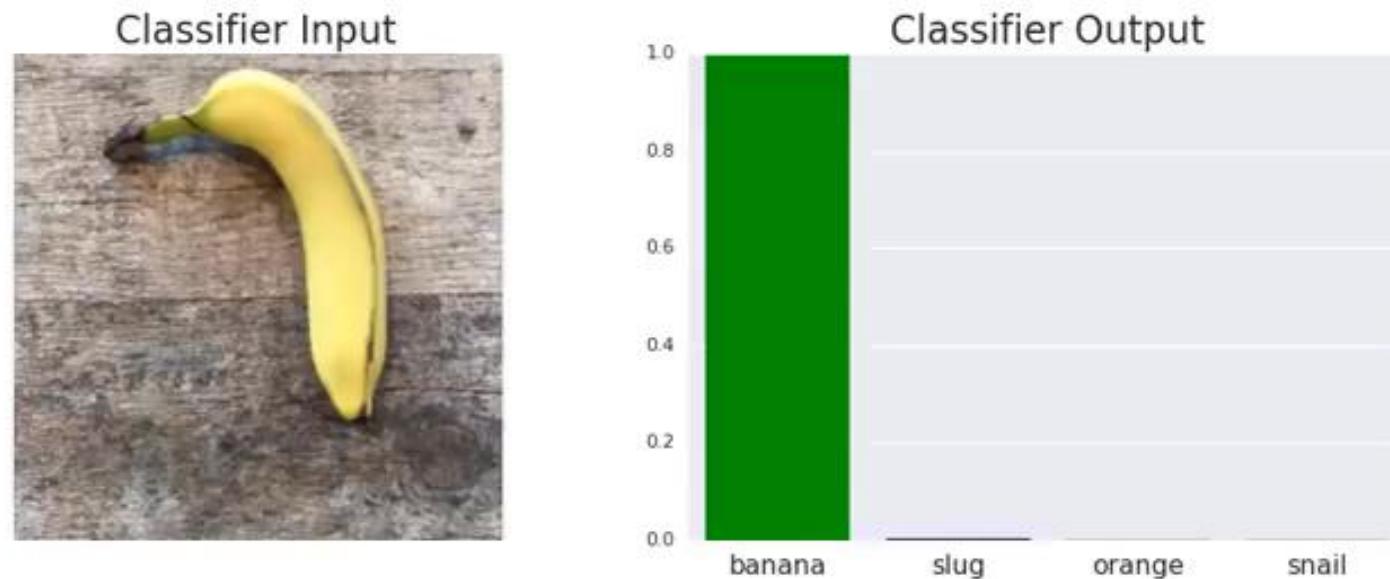
<https://www.microsoft.com/en-us/research/blog/what-are-the-biases-in-my-data/>

<https://research.aimultiple.com/ai-bias/>

# Danger of AI and ML: Attacks

## Adversarial attacks of AI

- A sticker near a banana



# Danger of AI and ML: DeepFake

## Generative Adversarial Networks, GAN



An example of deepfake technology: in a scene from Man of Steel, actress Amy Adams in the original (left) is modified to have the face of actor Nicolas Cage (right)



### SCIgen - An Automatic CS Paper Generator

[About](#) [Generate](#) [Examples](#) [Talks](#) [Code](#) [Donations](#) [Related](#) [People](#) [Blog](#)

#### About

SCIgen is a program that generates random Computer Science research papers, including graphs, figures, and citations. It uses a hand-written **context-free grammar** to form all elements of the papers. Our aim here is to maximize amusement, rather than coherence.

One useful purpose for such a program is to auto-generate submissions to conferences that you suspect might have very low submission standards. A prime example, which you may recognize from spam in your inbox, is SCI/IIS and its dozens of co-located conferences (check out the very broad conference description on the [WMSCI 2005](#) website). There's also a list of **known bogus conferences**. Using SCIgen to generate submissions for conferences like this gives us pleasure to no end. In fact, one of our papers was accepted to SCI 2005! See [Examples](#) for more details.

We went to WMSCI 2005. Check out the [talks and video](#). You can find more details in our [blog](#).

Also, check out our 10th anniversary celebration project: [SCipher!](#)

#### Generate a Random Paper

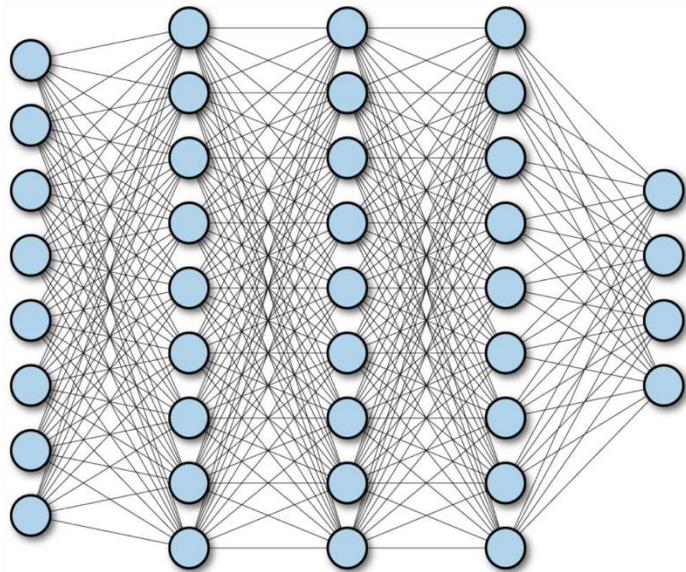
Want to generate a random CS paper of your own? Type in some optional author names below, and click "Generate".

Author 1:	<input type="text"/>
Author 2:	<input type="text"/>
Author 3:	<input type="text"/>
Author 4:	<input type="text"/>
Author 5:	<input type="text"/>

SCIgen currently supports Latin-1 characters, but not the full Unicode character set.

Automatically generate fake reviews or even academic papers SCIgen: Just enter the author's name and the computer can help you generate an "SCI-level" computer paper.

# What drives deep learning to be successful?



Deep Architecture



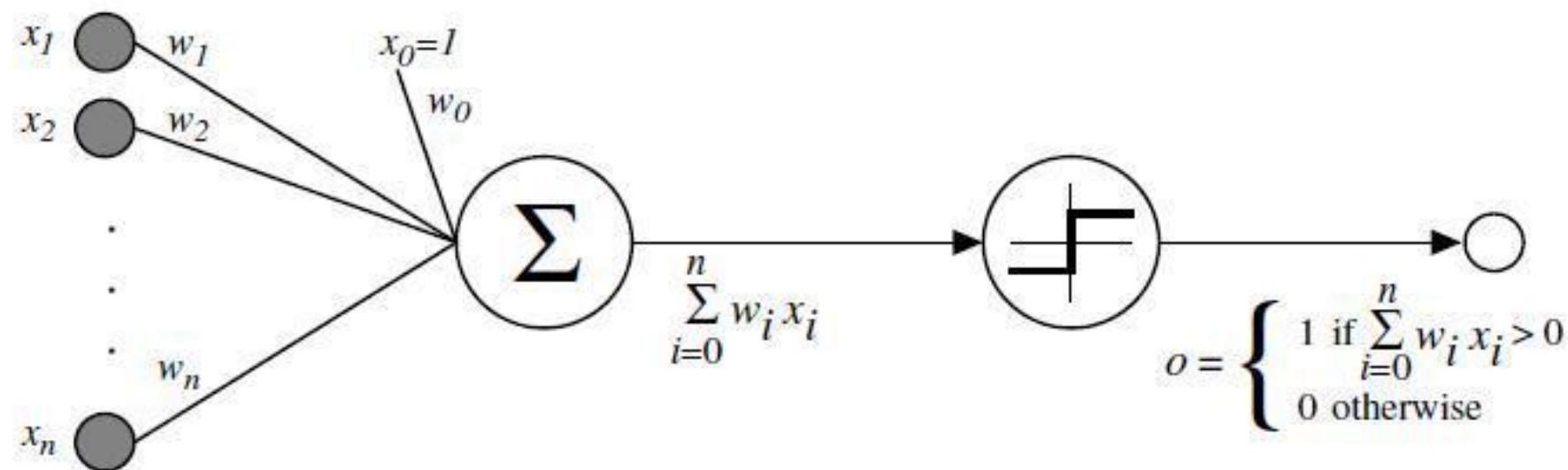
Big Data



Computing power

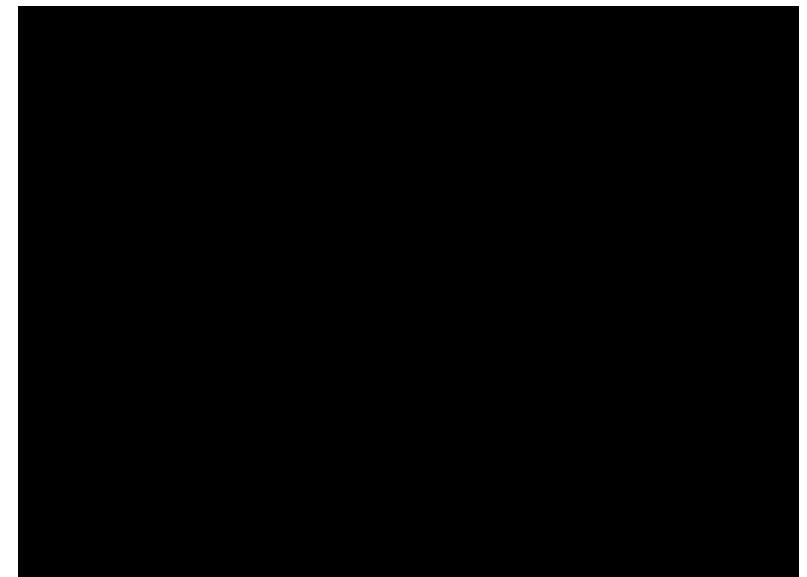
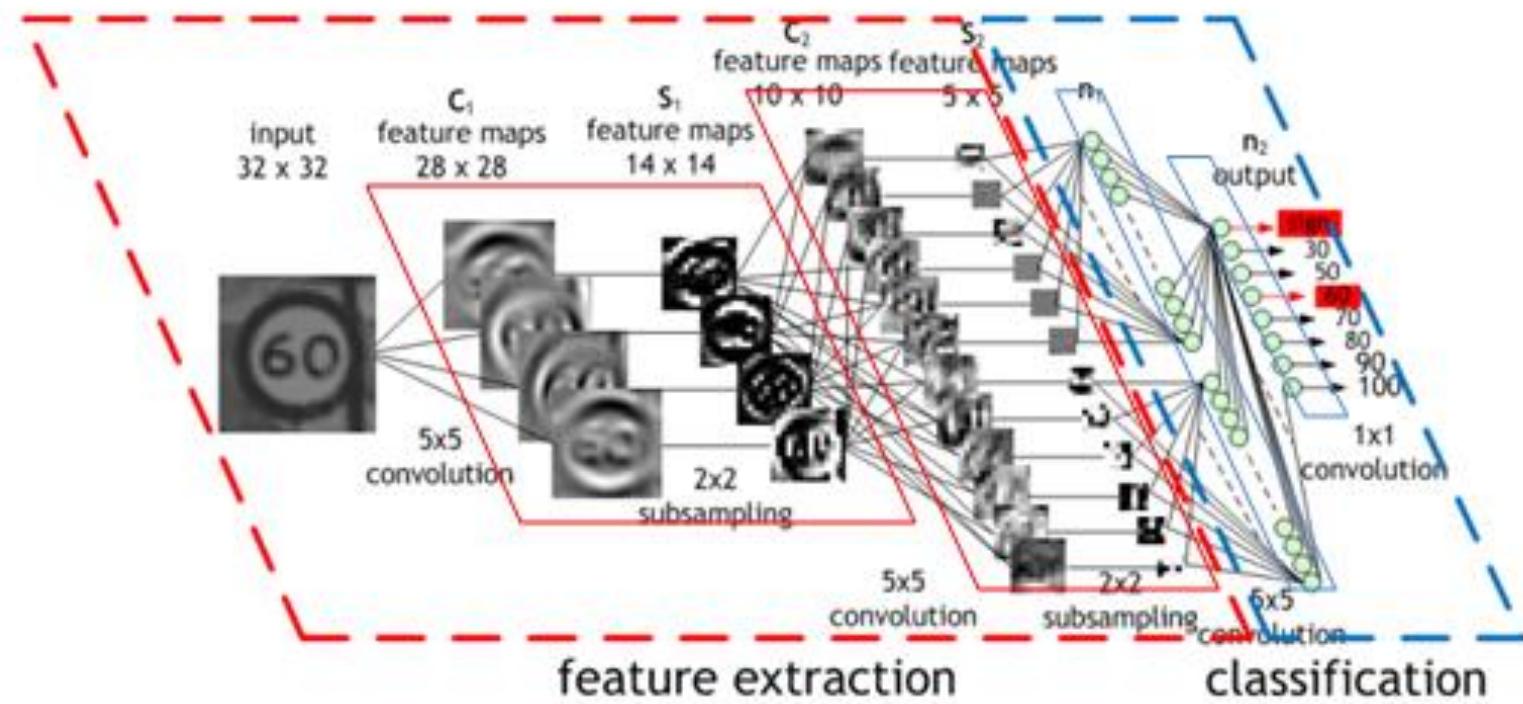
# Deep Neural Networks

- Frank Rosenblatt's Perceptron: 1956
  - Trained by back-propagation: Seppo Linnainmaa (1970) Paul Werbos (1974)



# Deep Neural Networks

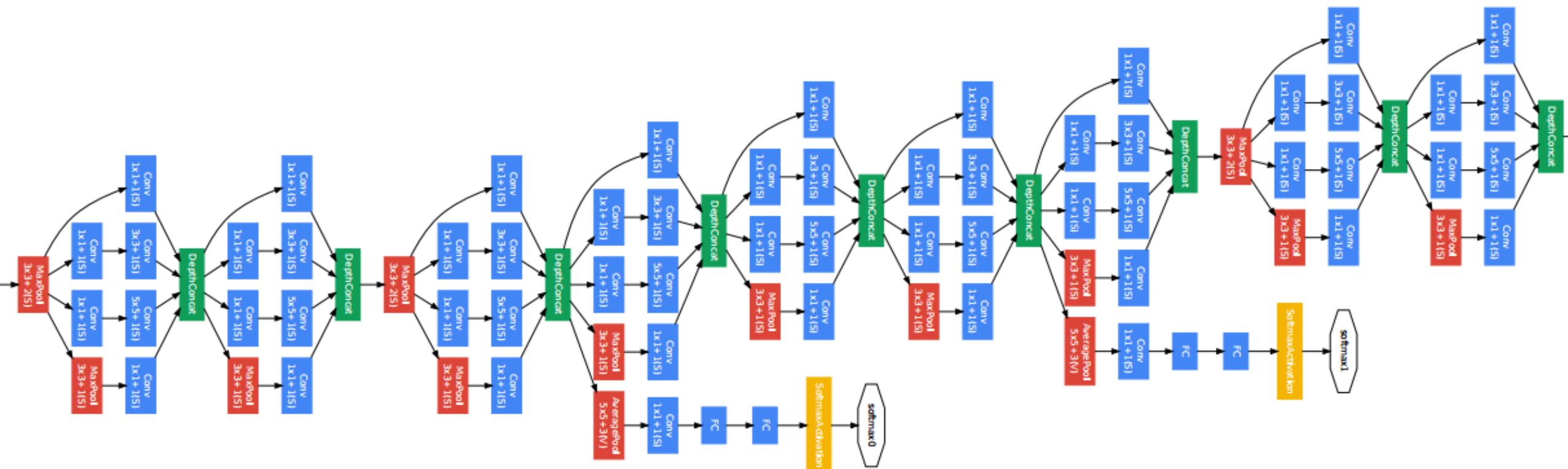
- Convolution Neural Network (CNN)



Yann LeCun (1989)

# Deep Neural Networks

- Deeper Convolutional Neural Networks

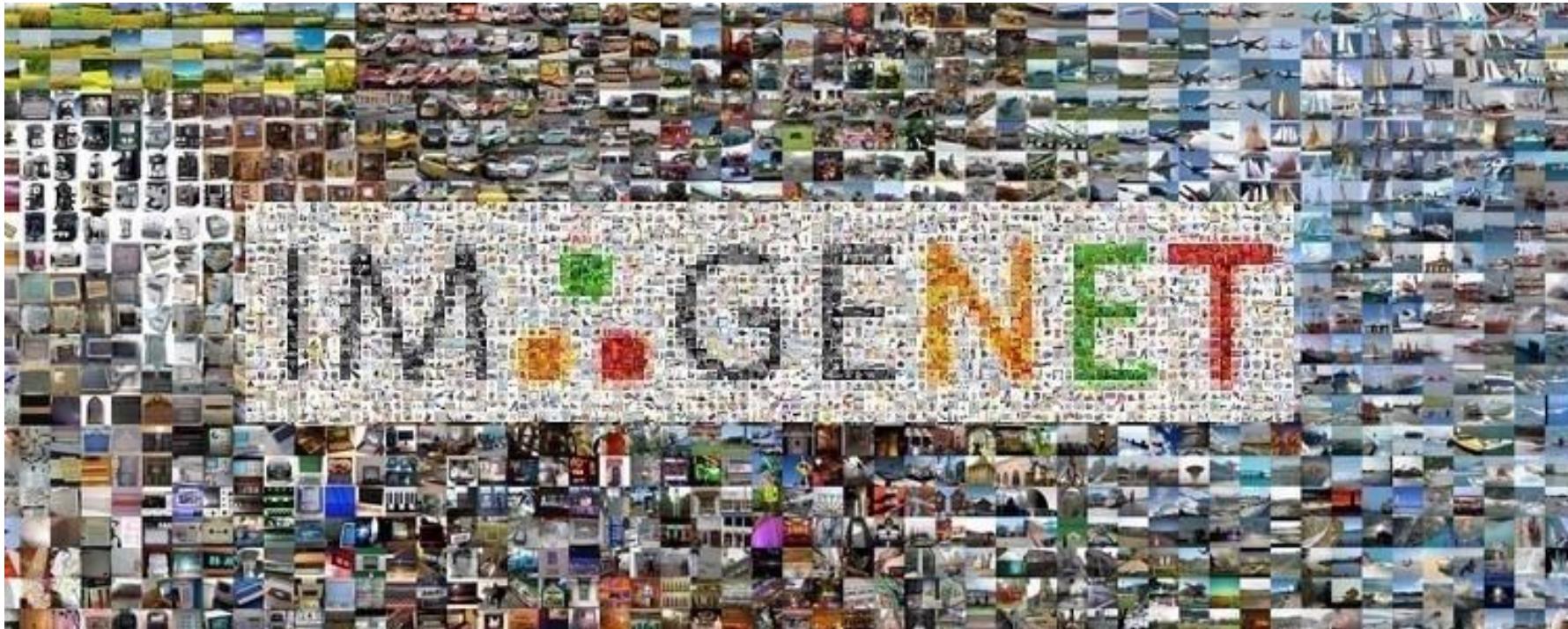


# Big Data

- MNIST
  - DATABASE of handwritten digits
  - 60,000 training samples
  - 28 x 28 pixels
  - 14.8 MB



# Big Data

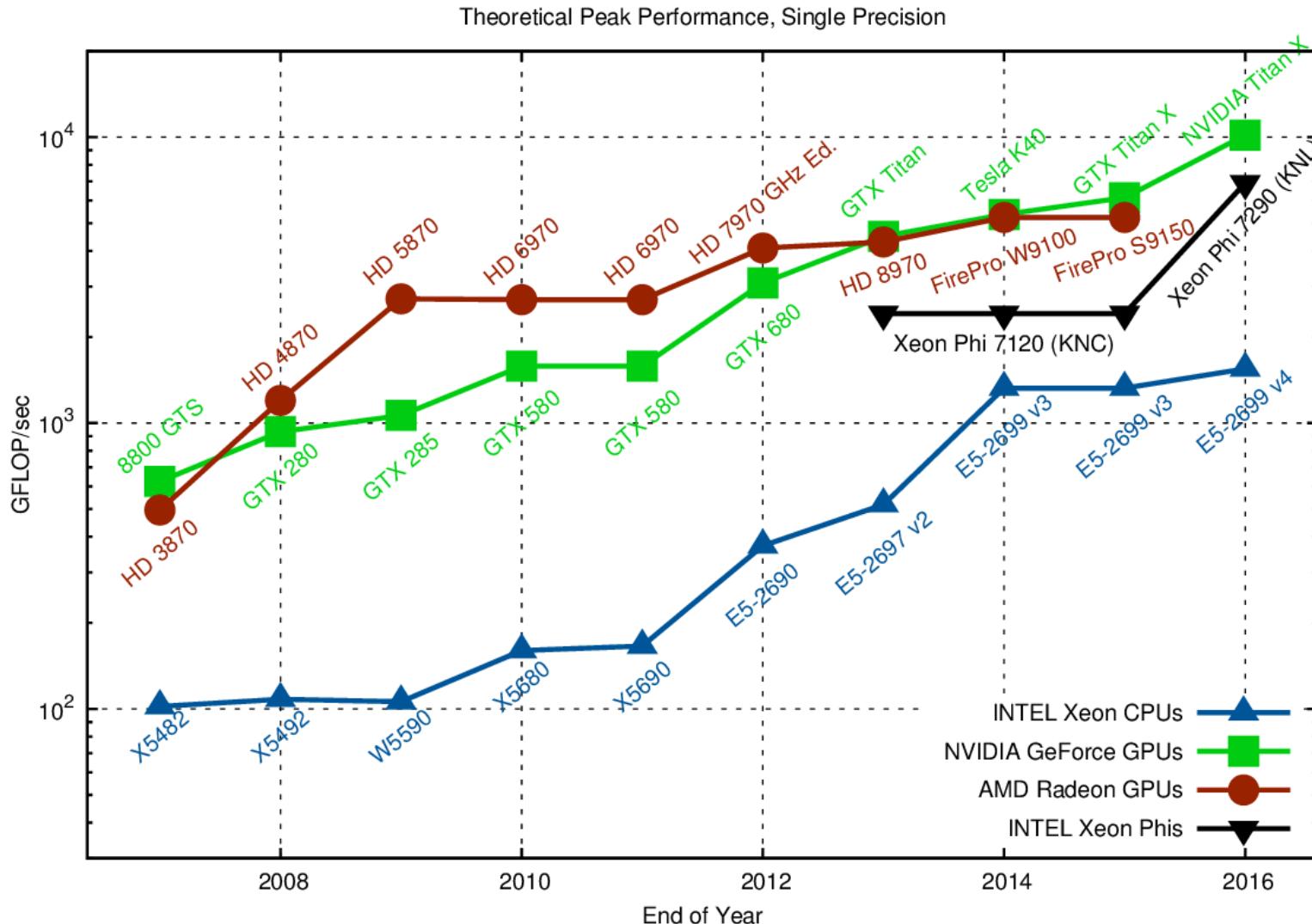


Dataset of over 14 million images across 21,841 categories, 300 G

[https://gluon-cv.mxnet.io/build/examples\\_datasets/imagenet.html](https://gluon-cv.mxnet.io/build/examples_datasets/imagenet.html) (2009) *How to obtain categories?*

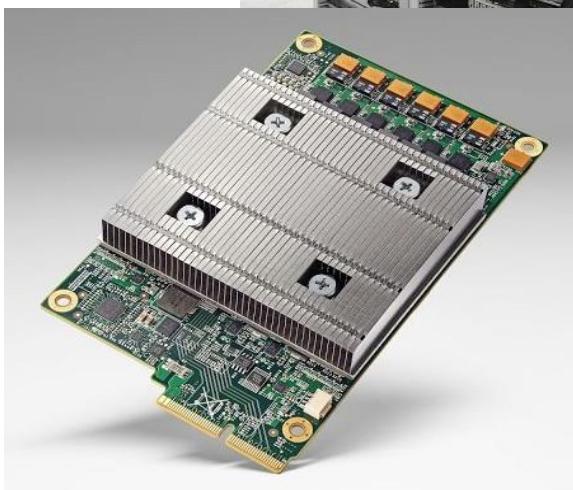


# Computing Power: GPU



# Computing Power: Google TPU

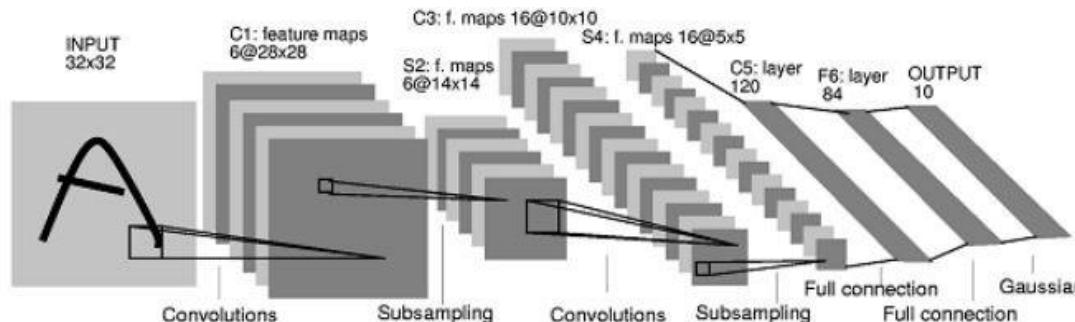
- (May 18, 2016) Google announced Tensor Processing Unit (TPU)
  - a custom ASIC built specifically for machine learning — and tailored for TensorFlow



# More Data + More Power

1998

LeCun et al.



# of transistors



$10^6$

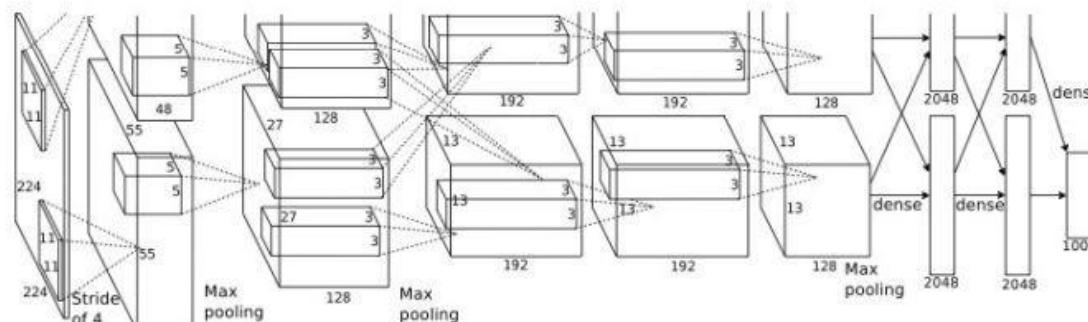
Pentium II

# of pixels used in training

$10^7$  NIST

2012

Krizhevsky  
et al.



# of transistors



$10^9$

GPUs



# of pixels used in training

$10^{14}$  IMagenET



# Deep learning Software



TensorFlow

Google



Francois Chollet  
(now at Google)



Facebook  
AI Research



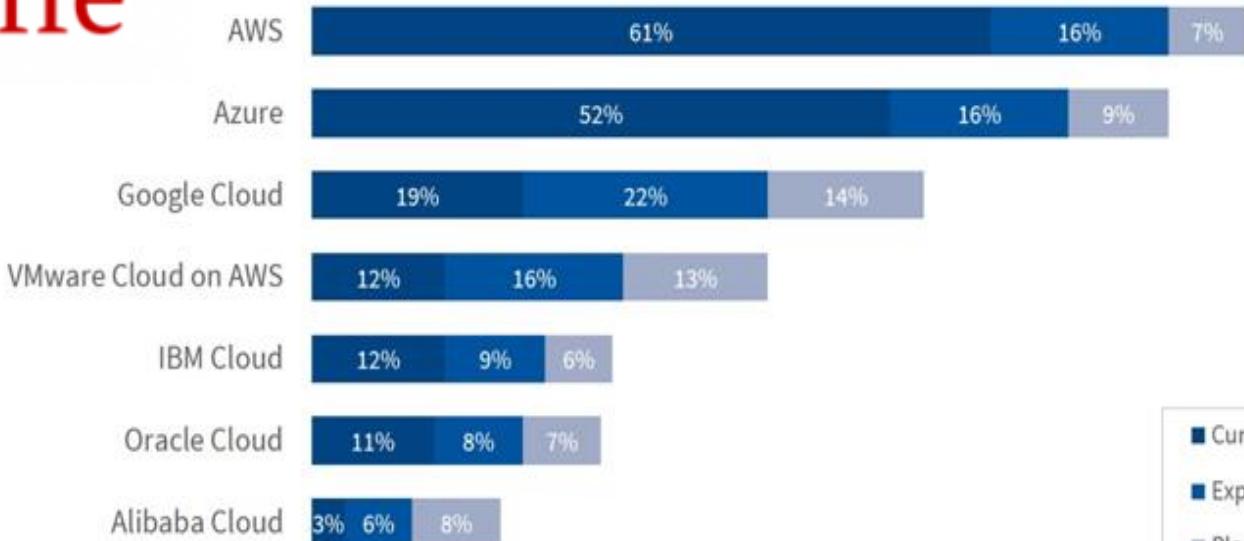
Amazon

# Caffe

## Cloud platforms

### Public Cloud Adoption

% of All Respondents



- Currently use
- Experimenting
- Plan to use

Source: RightScale 2019 State of the Cloud Report from Flexera



Simple. Flexible. Powerful.

```
from tensorflow import keras
from tensorflow.keras import layers

# Instantiate a trained vision model
vision_model = keras.applications.ResNet50()

# This is our video.encoding branch using the trained vision_model
video_input = keras.Input(shape=(100, None, None, 3))
encoded_frame_sequence = layers.TimeDistributed(vision_model)(video_input)
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our text-processing branch for the question input
question_input = keras.Input(shape=(100,), dtype='int32')
embedded_question = layers.Embedding(10000, 256)(question_input)
encoded_question = layers.LSTM(256)(embedded_question)

# And this is our video question answering model:
merged = keras.layers.concatenate([encoded_video, encoded_question])
output = keras.layers.Dense(1000, activation='softmax')(merged)
video_qa_model = keras.Model(inputs=[video_input, question_input],
                             outputs=output)
```

## Deep learning for humans.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

# Colab

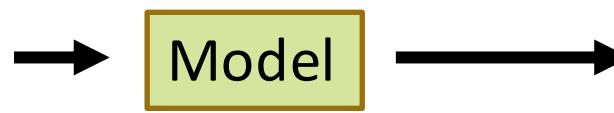


- Colab is a Python development environment that runs in the browser using Google Cloud.
  - <https://colab.research.google.com/>
  - Like *Jupyter Notebook (iPython)* , but you don't have to install anything on your computer or face issues of installation errors
- <https://www.youtube.com/watch?v=inN8seMm7UI>



# Machine Learning Overview

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	134	150	255	189	37	0	0	0	0	0		
7	0	0	0	0	0	5	92	36	36	140	154	154	154	224	263	263	263	263	263	263	263	263	170	0	0	0	0	0	0	
8	0	0	0	0	0	36	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	145	0	0	0	0	0	0	
9	0	0	0	0	0	9	144	202	253	223	182	182	182	182	182	138	65	161	253	253	235	38	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	253	253	147	0	0	0	0	0	0		
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235	253	253	70	0	0	0	0	0	0		
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	253	253	230	36	0	0	0	0	0		
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	243	253	251	131	0	0	0	0	0		
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	178	253	253	224	0	0	0	0	0		
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	263	253	253	141	0	0	0	0	0		
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	253	253	208	25	0	0	0	0	0		
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	244	253	253	124	0	0	0	0	0		
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	177	253	253	203	14	0	0	0	0	0		
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	107	252	253	253	81	0	0	0	0	0		
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	253	253	253	92	0	0	0	0	0		
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	183	253	253	253	236	0	0	0	0		
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	125	253	253	253	233	63	0	0	0	0		
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	218	253	253	253	137	0	0	0	0	0		
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	243	253	253	101	8	0	0	0	0	0		
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	187	253	253	234	41	0	0	0	0	0		
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		



- Key Terminologies
  - Labels:  $Y$
  - Features:  $X$
  - Examples:  $(X, Y)$
  - Models:  $f: X \rightarrow Y$

# Supervised Learning

Feature Space  $\mathcal{X}$



Label Space  $\mathcal{Y}$

"Digit 3"  
"Digit 7"



Share Price at  $t$   
"\$ 24.50"



**Task:** Given  $X \in \mathcal{X}$ , predict  $Y \in \mathcal{Y}$ .



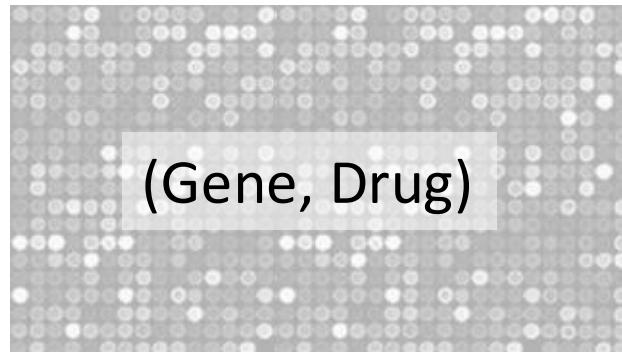
# Supervised Learning - Regression

Feature Space  $\mathcal{X}$



Label Space  $\mathcal{Y}$

Share Price at  $t$   
“\$ 24.50”



Expression level  
“0.01”

Continuous Labels

ML Models:

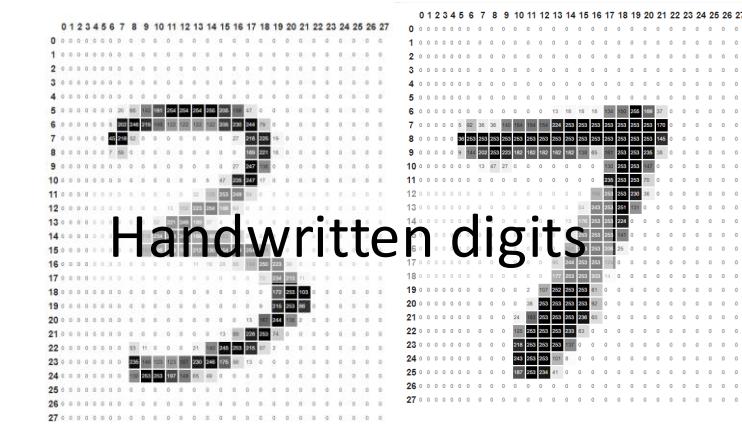
Linear Regression  
(Ordinary Least Square /  
Ridge / Lasso)  
Regression Tree

Boosting  
Neural Network

# Supervised Learning - Classification

Feature Space  $\mathcal{X}$

Words in a document



Handwritten digits

Discrete Labels

Label Space  $\mathcal{Y}$

“Sports”  
“News”  
“Science”

...

ML Models:

“Digit 3”  
“Digit 7”

Logistic Regression  
K-Nearest-Neighbors  
Support Vector Machine  
Naïve Bayes  
Classification Tree

Boosting  
Neural Network

# Unsupervised Learning

- Aka “learning without a teacher”

Feature Space  $\mathcal{X}$

Words in a document



Word distribution  
(Probability of a word)

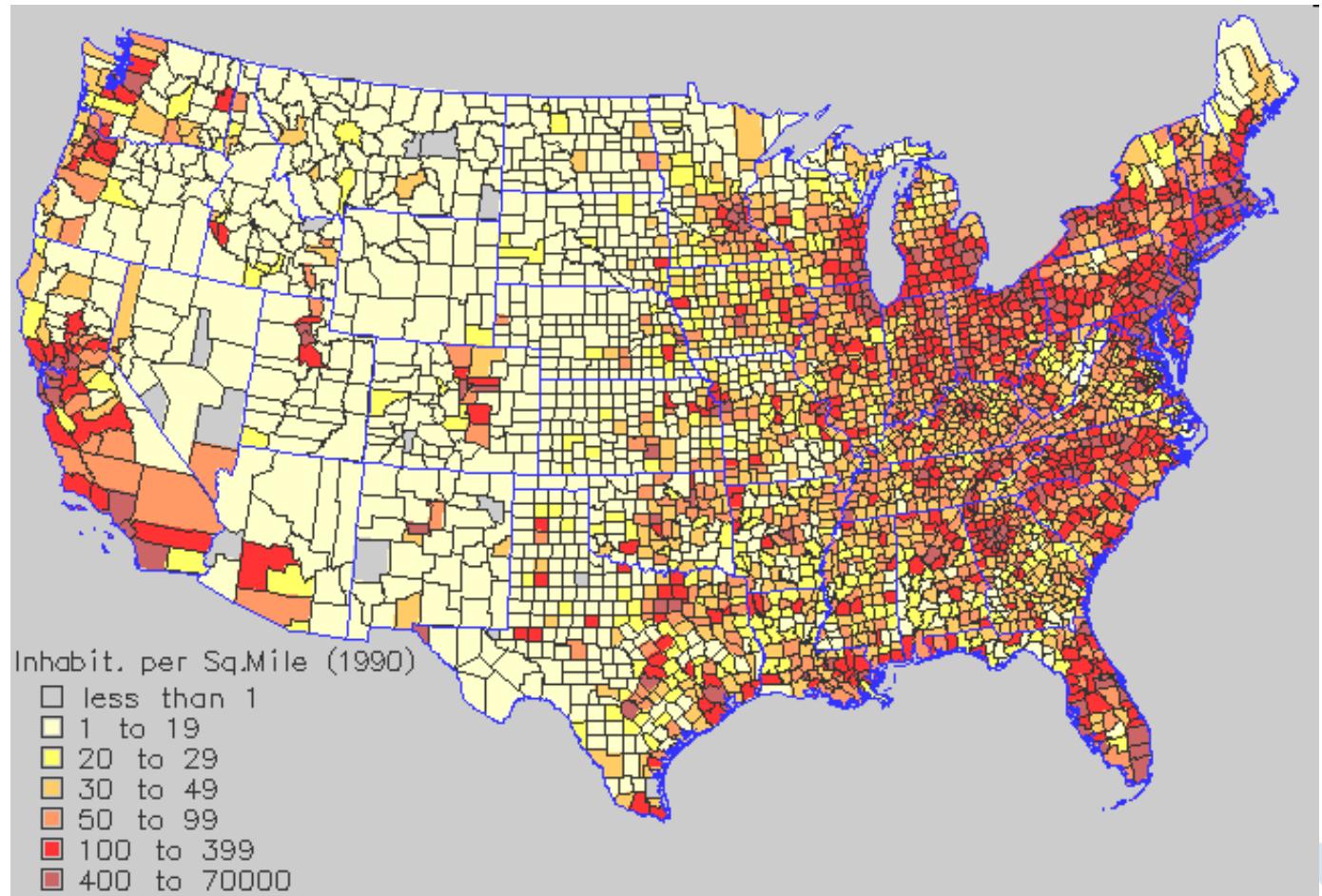
Task: Given  $X \in \mathcal{X}$ , learn  $f(X)$ .

The goal is more diverse

Can even be served as a pre-processing step for supervised learning

# Unsupervised Learning – Density Estimation

- Population density



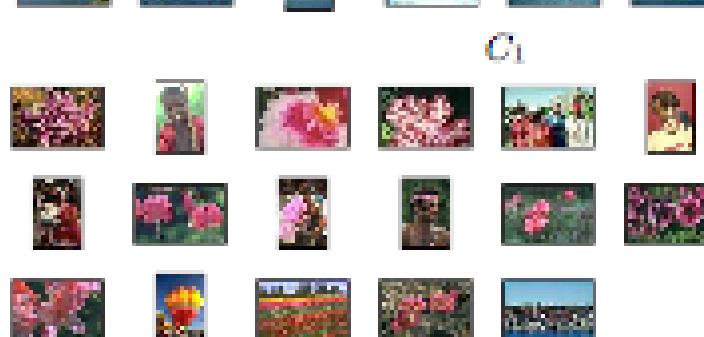
# Unsupervised Learning – Clustering

- Group similar things e.g. images

[Goldberger et al.]

- Methods

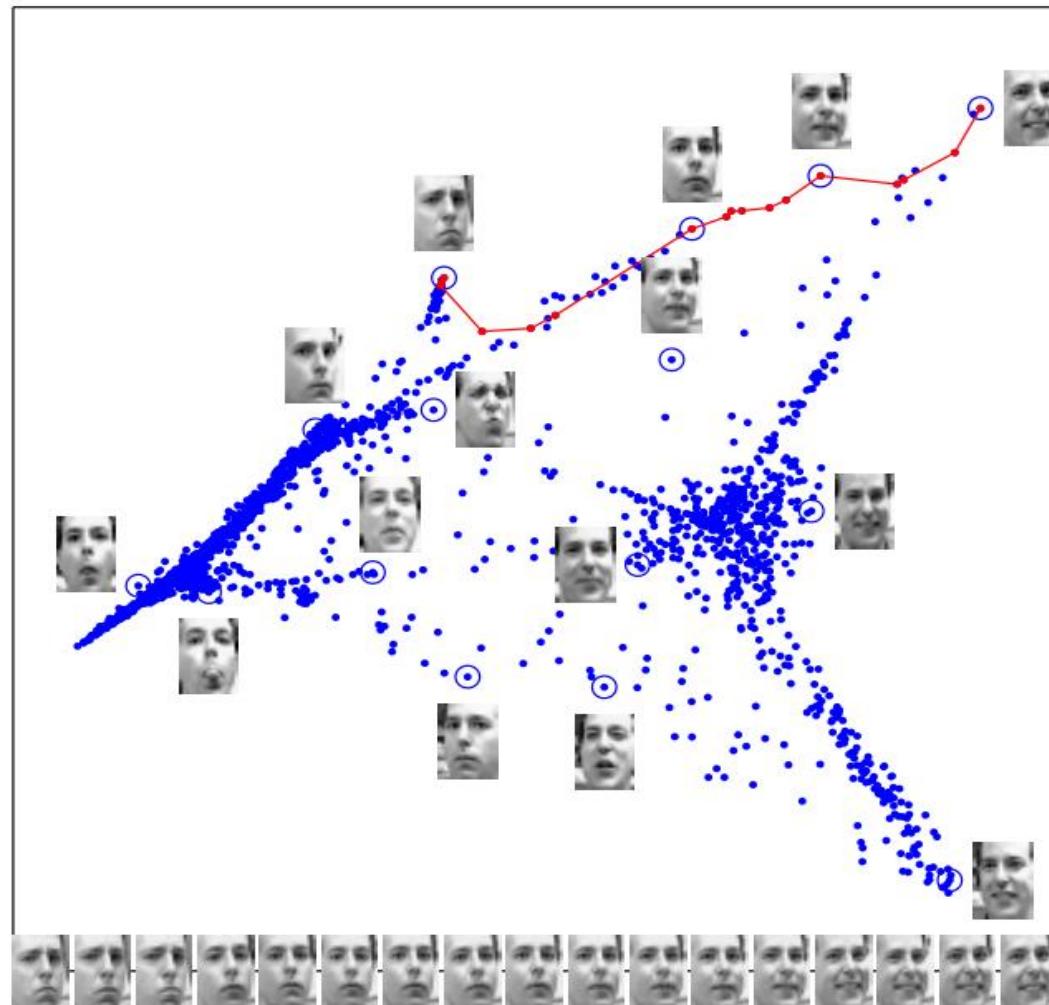
- Hierarchical Clustering
    - K-means



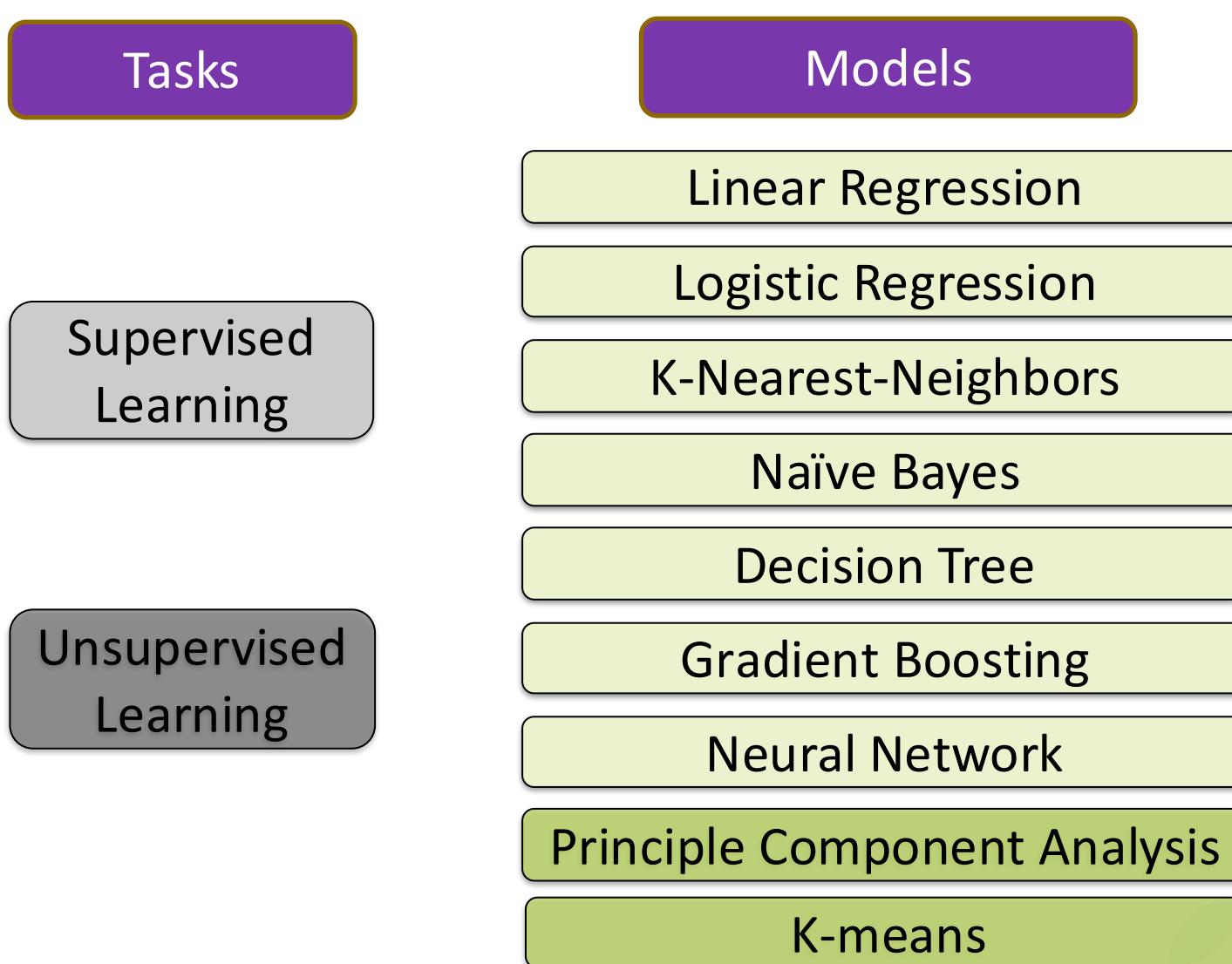
# Unsupervised Learning - Embedding

- Dimension Reduction
  - Images have thousands or millions of pixels.
  - Can we give each image a coordinate, such that similar images are near each other?
- Methods
  - PCA
  - LDA
  - Manifold Learning
  - Autoencoder

[Saul & Roweis '03]



# Machine Learning Tasks and Models



# Machine Learning Tasks and Models

Tasks

Regression

Classification

Clustering

Dimension  
Reduction

Models

Linear Regression

Logistic Regression

K-Nearest-Neighbors

Naïve Bayes

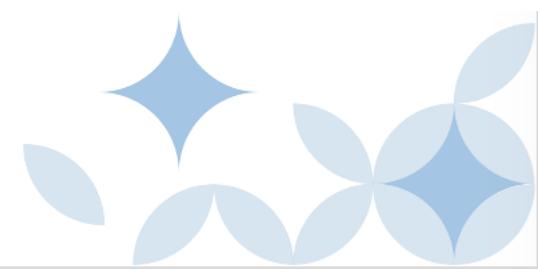
Decision Tree

Gradient Boosting

Neural Network

Principle Component Analysis

K-means



# Machine Learning Tasks and Models

Tasks

Regression

Classification

Clustering

Dimension  
Reduction

Models

Linear Regression

Logistic Regression

K-Nearest-Neighbors

Naïve Bayes

Decision Tree

Gradient Boosting

Neural Network

Principle Component Analysis

K-means



# Model Assessment: Performance Measure

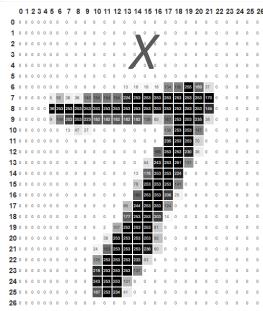
- Supervised Learning

**Task:** Given  $X \in \mathcal{X}$ , predict  $Y \in \mathcal{Y}$ .  $X$  - test data  
 $\equiv$  Construct **prediction rule**  $f : \mathcal{X} \rightarrow \mathcal{Y}$

**Performance:**

$loss(Y, f(X))$ : Measure of similarity between true label  $Y$  and prediction  $f(X)$

**Classification:** Handwritten digits



$X$	$Y$	$f(X)$	$loss(Y, f(X))$	$= 1_{Y \neq f(x)}$
	“Digit 7”	“Digit 1”	1	<b>0/1 loss</b>
		“Digit 7”	0	

# Model Assessment: Performance Measure

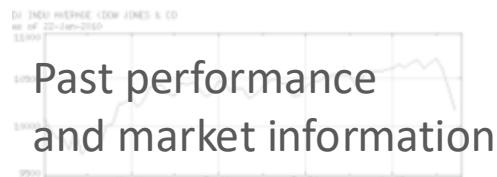
- Supervised Learning

**Task:** Given  $X \in \mathcal{X}$ , predict  $Y \in \mathcal{Y}$ .  $X$  - test data  
 $\equiv$  Construct **prediction rule**  $f : \mathcal{X} \rightarrow \mathcal{Y}$

**Performance:**

$loss(Y, f(X))$ : Measure of similarity between true label  $Y$  and prediction  $f(X)$

**Regression:** predict stock price



Past performance  
and market information

$x$

$y$

$f(x)$

$loss(Y, f(X))$

$$= (y - f(x))^2$$

**Squared loss**

Share price  
“\$24.50”

“\$24.50”

“\$26.00”

“\$26.10”

0

?

?



# Model Assessment: True vs. Empirical Risk

- **True Risk:** Target performance measure  $E_{test}[\text{loss}(Y, f(X))]$ 
  - Performance on a random test point  $(X, Y)$
  - Classification – Probability of misclassification  $E[1_{Y \neq f(X)}] = P(Y \neq f(X))$
  - Regression – Mean Squared Error  $E(Y - f(X))^2$
- **Empirical Risk:** Performance on training data  $\frac{1}{n} \sum_{i=1}^n \text{loss}(Y_i, f(X_i))$ 
  - Classification – Proportion of misclassified examples.  $\frac{1}{n} \sum_{i=1}^n 1_{Y_i \neq f(X_i)}$
  - Regression – Average Squared Error  $\frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$

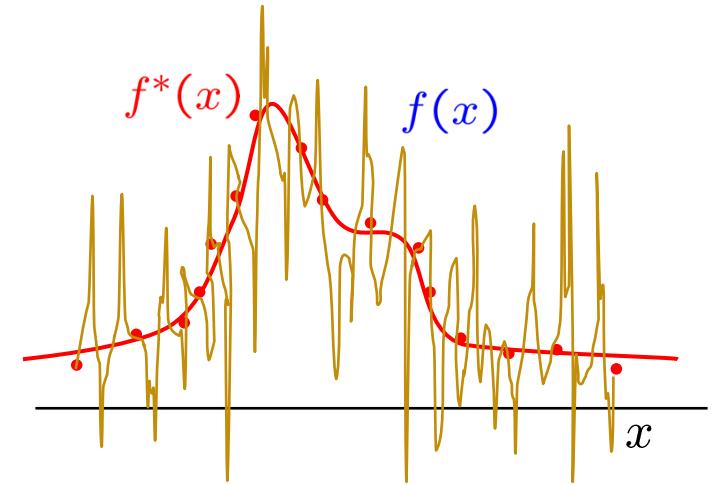


# Overfitting

- Is the following predictor a good one?

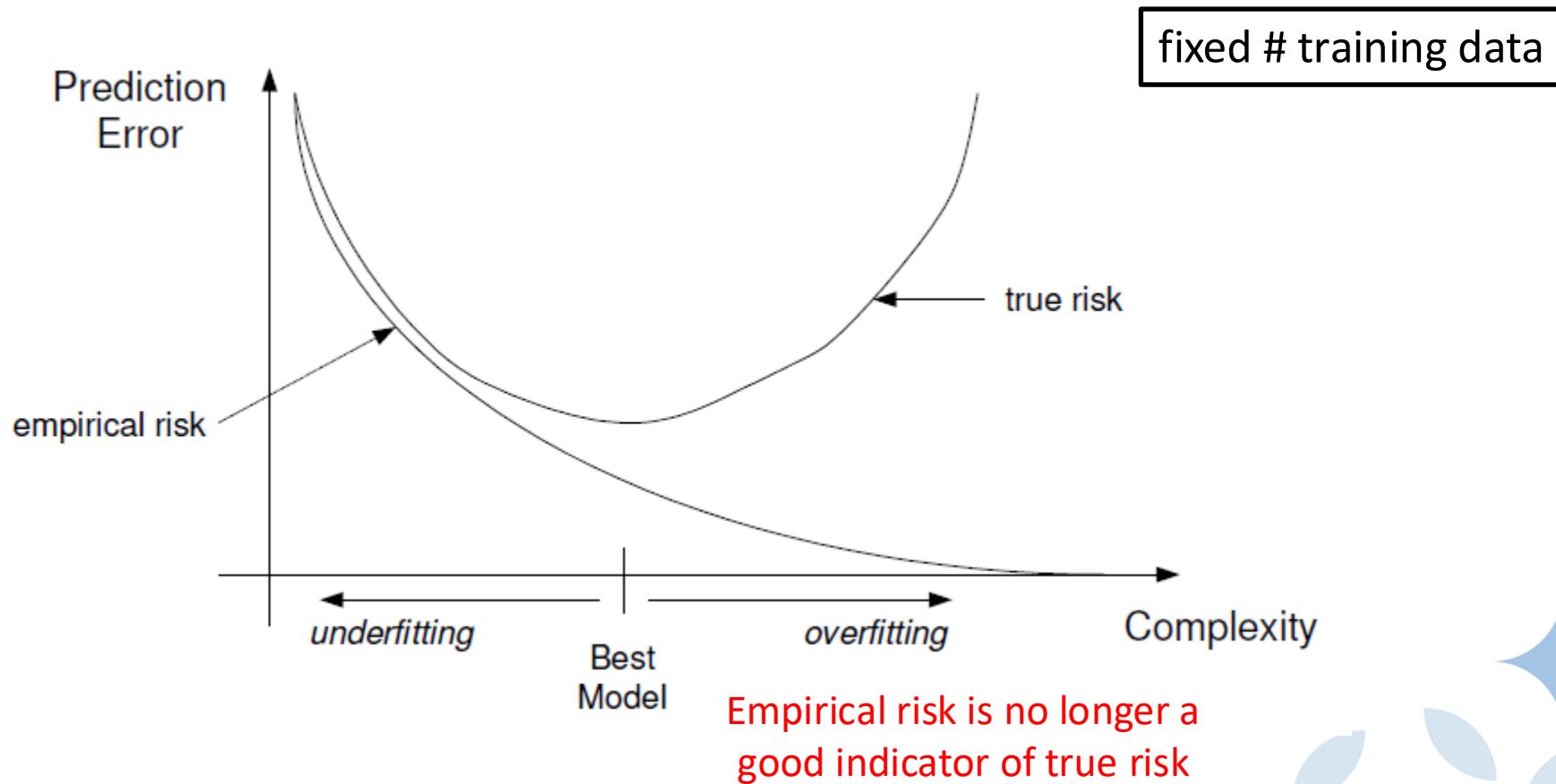
$$f(x) = \begin{cases} Y_i, & x = X_i \text{ for } i = 1, \dots, n \\ \text{any value,} & \text{otherwise} \end{cases}$$

- What is its empirical risk? (performance on training data)
- What about true risk?
- Will predict very poorly on new random test point:  
**Large test error (generalization error)**



# Effect of Model Complexity

- If we allow very complicated models/predictors, we could overfit the training data.



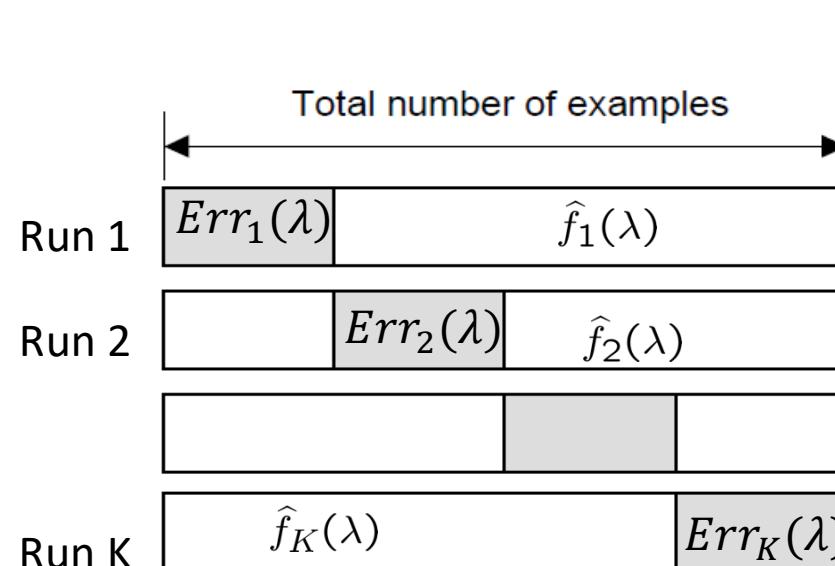
# How to control model complexity? - Regularization

- Consider linear regression: given  $n$  data points ( $X_i \in R^m, y_i \in R$ )
- Fit a linear model:  $\min_{W \in R^m} \frac{1}{n} \sum_{i=1}^n (y_i - X_i^T W)^2$
- However, it will perform bad when the dimension  $m$  is large (model is too complex)
- Regularization:  $\min_{W \in R^m} \frac{1}{n} \sum_{i=1}^n (y_i - X_i^T W)^2 + \lambda \cdot \text{pen}(W)$
- $\lambda$ : regularization parameter, controls the model complexity
- $\text{pen}(W)$ : penalty of  $W$ 
  - Examples:  $\text{pen}(W) = \|W\|_2 = \sqrt{\sum_j |w_j|^2}$
  - $\text{pen}(W) = \|W\|_1 = \sum_j |w_j|$  ( $l_1$ -penalty, Lasso)
    - Lasso ( $l_1$ -penalty) results in sparse solutions – vector with more zero coordinates



# How to choose regularization parameters

- K-fold cross-validation
  - Randomly create  $K$ -fold partition of the dataset.
  - Form  $K$  hold-out regression/classification function, each time using one partition as validation and rest  $K - 1$  as training datasets.
  - Final parameter is the one that leads to the *smallest average validation error*



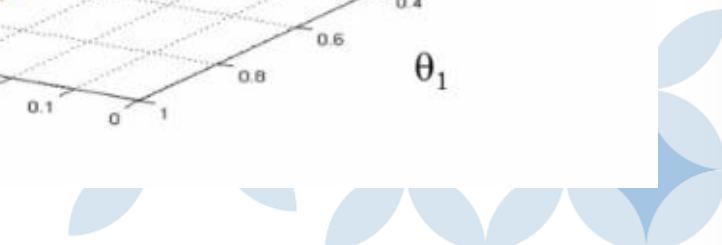
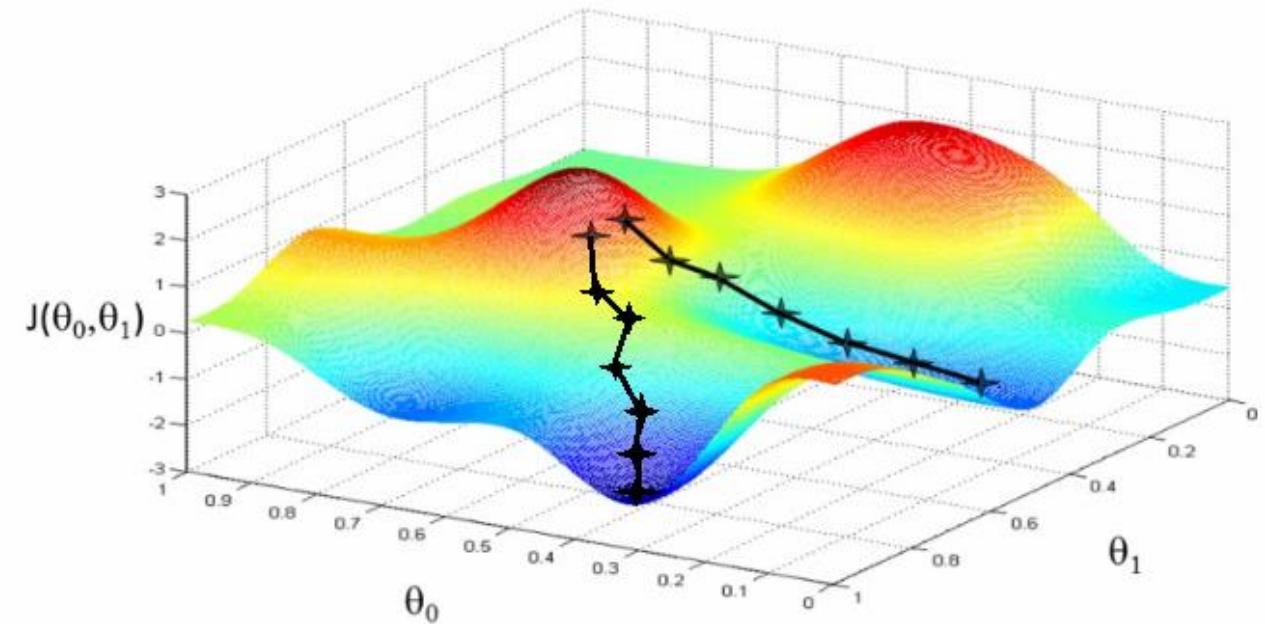
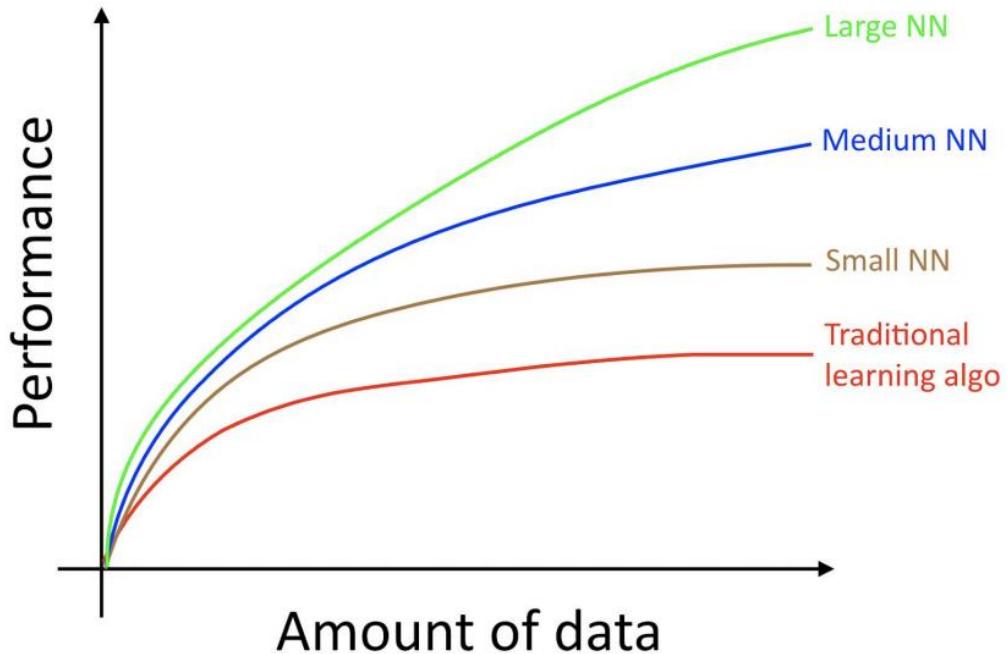
Find the best  $\lambda$  from a list of candidates  $(\lambda_1, \dots, \lambda_L)$  that minimizes

$$\frac{1}{K} \sum_k Err_k(\lambda)$$

# Build ML Algorithms - 5 Useful Tips

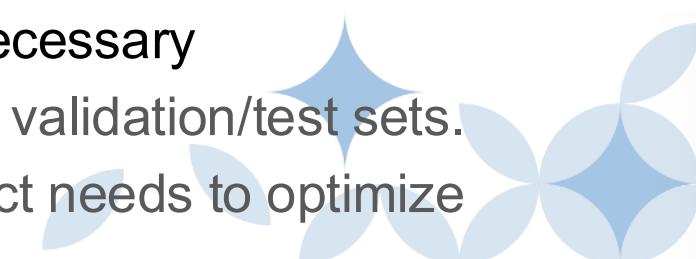
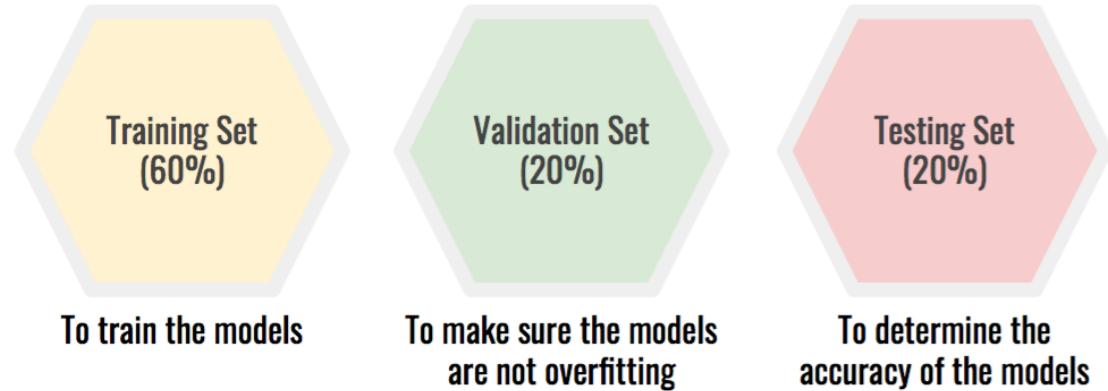
## by Machine Learning Yearning, Andrew Ng

- Rule 1: Biggest drivers of the success
  - Data availability & Computational scale



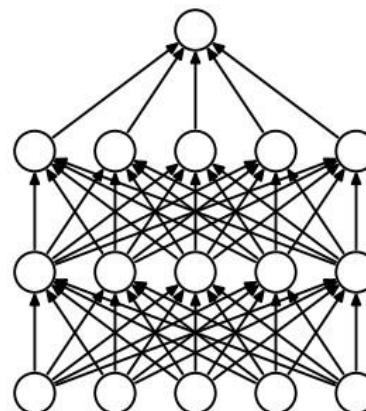
# Build ML Algorithms - 5 Useful Tips

- Rule 2: Divide your datasets quickly & properly
  - Validation and test sets should come from the **same** distribution
    - Wrong example: US and India data for validation and China data for test
  - Validation sets should be **large enough** to detect difference
    - Set of 100 examples would not be able to detect a 0.1% difference.
    - Meanwhile, split the validation set to Eyeball dev set + Blackbox dev set
  - Incorrectly Chosen - Don't be afraid to **change** your sets when necessary
    - Actual distribution you need to do well on is different from the validation/test sets.
    - The metric is measuring something other than what the project needs to optimize

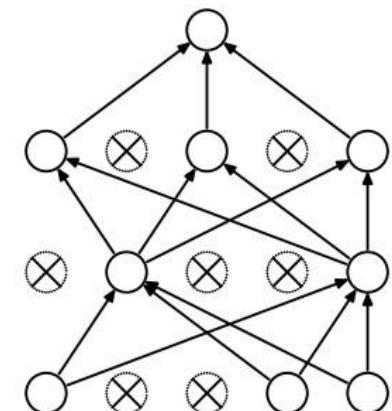


# Build ML Algorithms - 5 Useful Tips

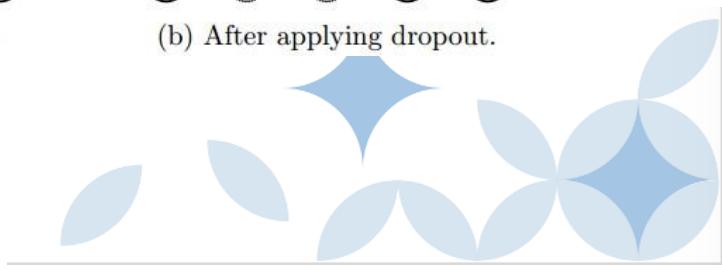
- Rule 3: Bias and Variance: The two big sources of error
  - Identifying the bias and variance
    - Error rate on the training / validation set
  - Tradeoff: Underfitting / Overfitting
  - In the modern era: more options
    - High avoidable bias
      - increase the size of your model
    - High variance
      - add data to your training set
      - proper regularization
      - early stopping
    - Modify input features / model architecture



(a) Standard Neural Net



(b) After applying dropout.



# Build ML Algorithms - 5 Useful Tips

- Rule 4: Utilizing multiple evaluation

METRIC

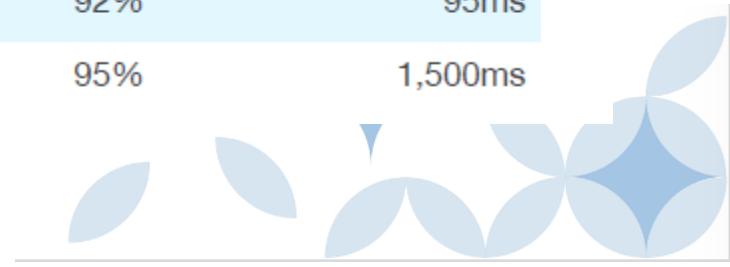
- Tradeoff between Precision / Recall

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}$$

- Trade-off Between accuracy and running time
    - Combinations of multiple evaluation metrics
      - E.g. Accuracy - 0.5\* RT
    - Optimizing + Satisfying Metric
      - Set 100ms as acceptable
      - Optimize Accuracy

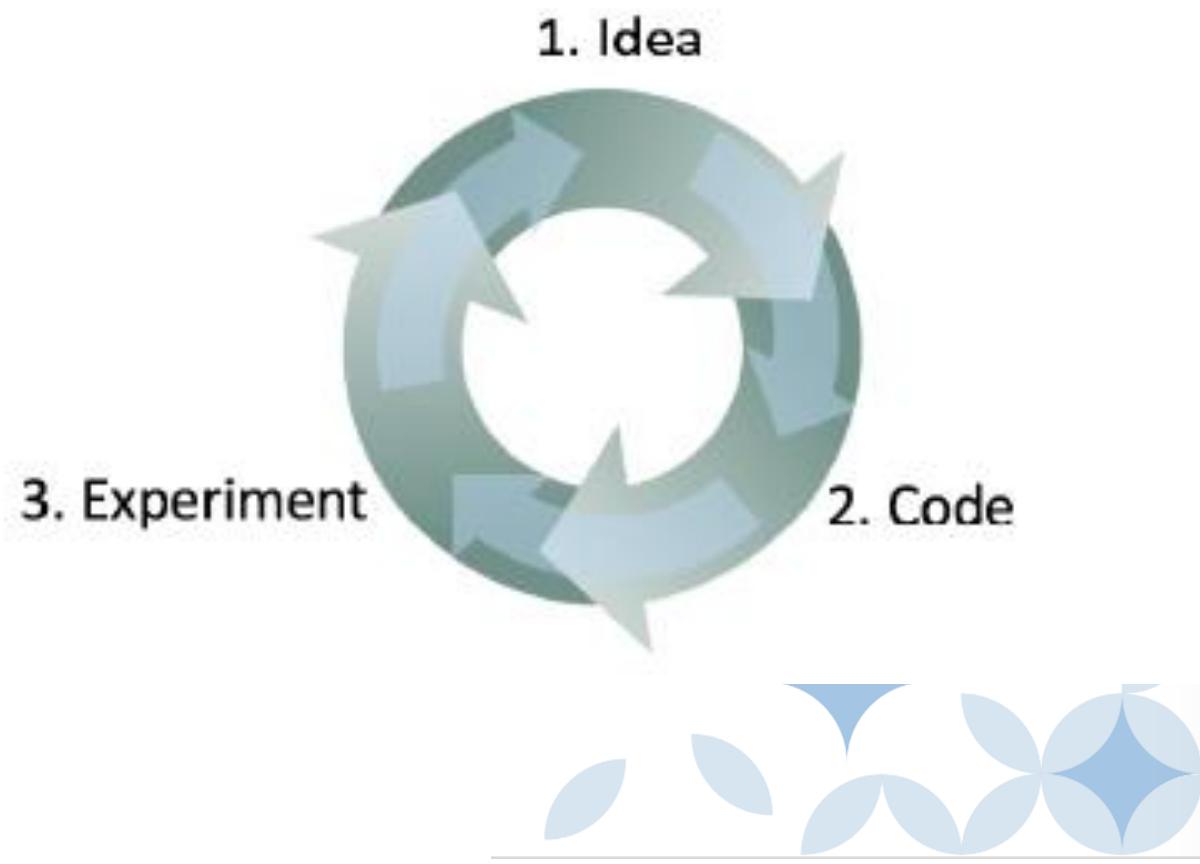
Classifier	Precision	Recall
A	95%	90%
B	98%	85%

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms



# Build ML Algorithms - 5 Useful Tips

- Rule 5: Iterative Process
  - Don't expect it to work first time
  - Try out many dozens of ideas before you discover something satisfactory
  - Quickly detect whether to refine or to discard the idea



# Slide Courtesy and Acknowledgement

- MIT Course: MIT 6.S191: ©Alexander Amini and Ava Soleimany
- Carnegie Mellon Course: Machine Learning (by Aarti Singh) and Deep Learning (by Ruslan Salakhutdinov)
- Deep Learning course at Stevens Institute of Technology (by Shusen Wang)
- NYU Course: Introduction to AI & Its Applications in Business (by Alex Tuzhilin)
- Stanford Course: Convolutional Neural Networks for Visual Recognition (by Feifei Li)



Thanks!



Questions?