1. Introduction

STA3142 Statistical Machine Learning

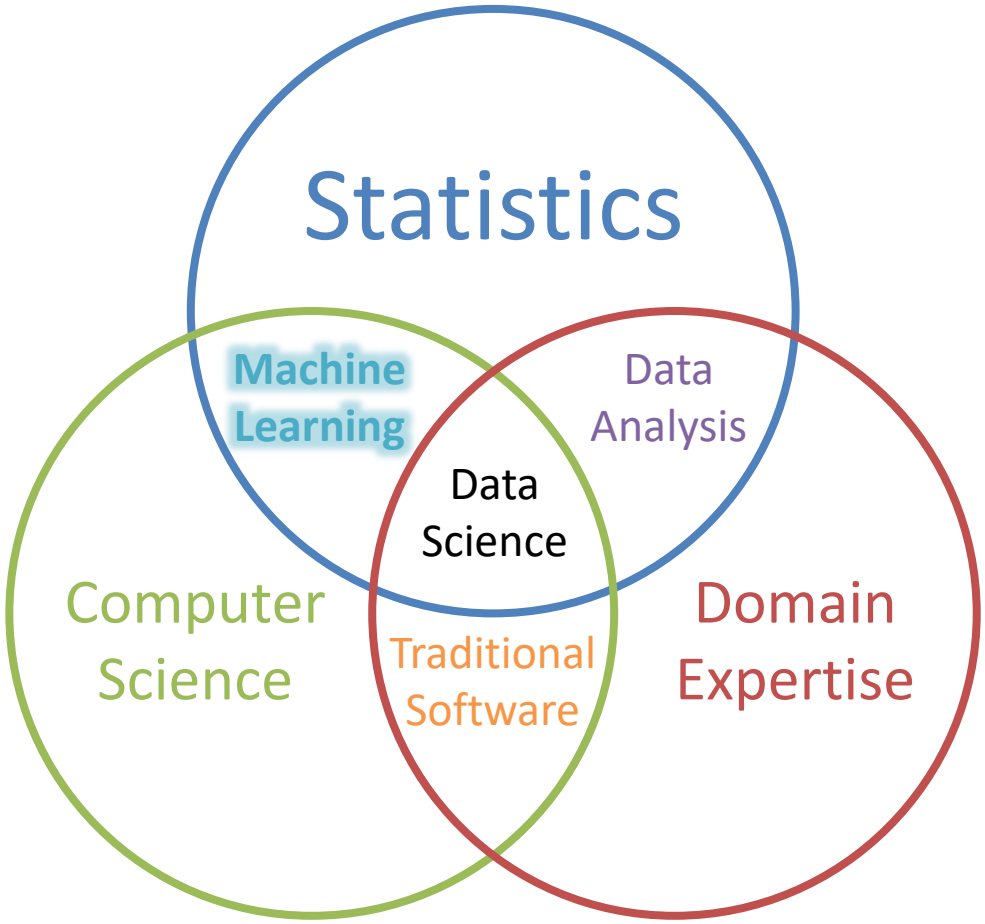
Kibok Lee

Assistant Professor of

Applied Statistics / Statistics and Data Science

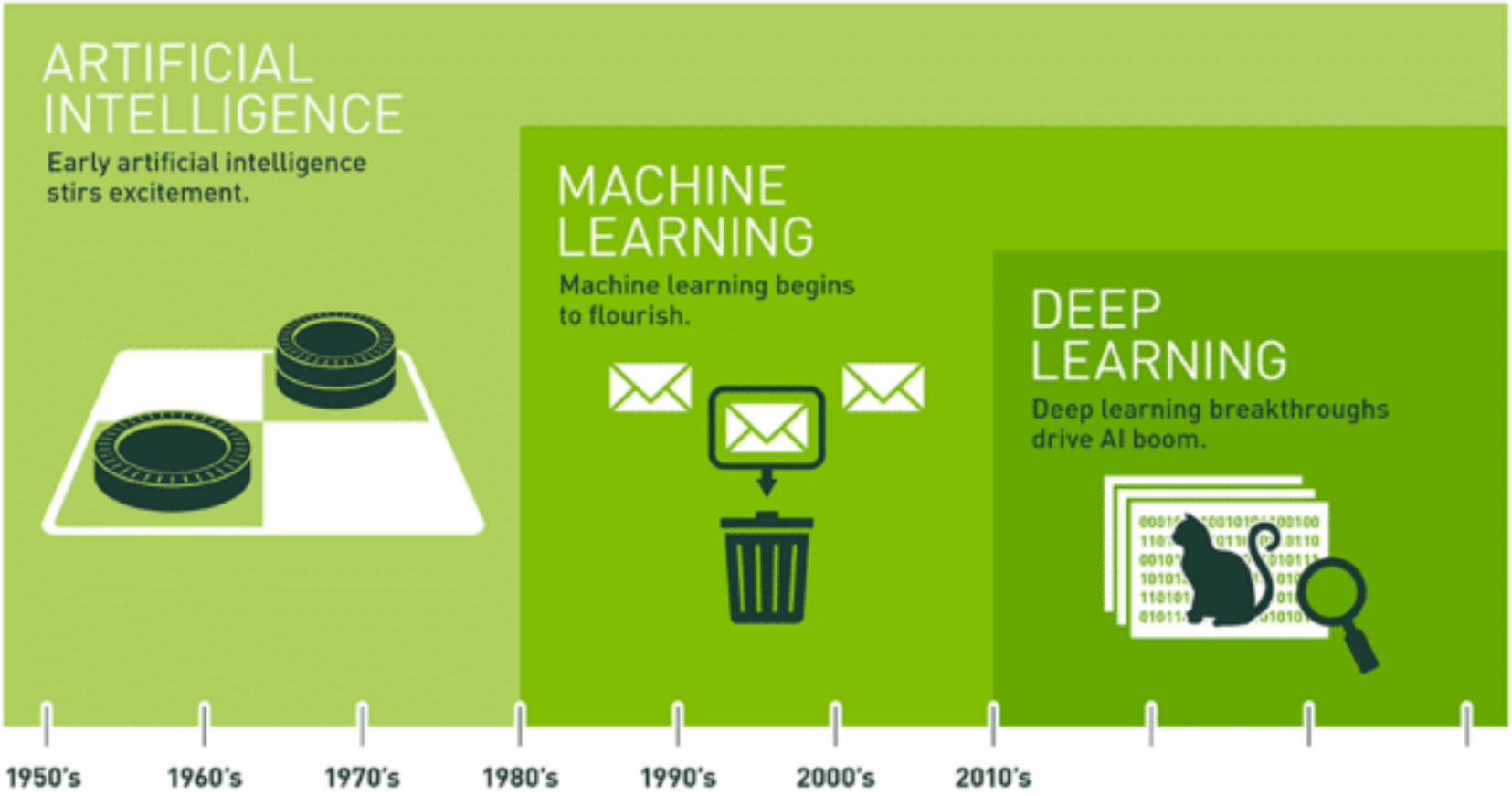
Mar 5, 2024

What You Will Learn



Machine Learning
=
Statistics
+
Computer Science

AI vs. ML vs. DL



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

What You Will Learn

- Foundations of machine learning
  - Mathematical derivation
  - Implementation of algorithms
  - Applications

- Topics
  - Basic ML
  - Supervised learning
  - Unsupervised learning
  - Neural networks
  - Other topics

What You Will Learn

- Our goal is
  - To understand fundamentals of machine learning
  - To learn technical details of machine learning algorithms
  - To learn how to implement some important algorithms
  - To use machine learning algorithms for your research and applications
    - ML engineers are expected to have a master's degree
    - ML research scientists are expected to have a Ph.D. degree

Outline

- Course Information
- What is Machine Learning?

Teaching Staff

- Instructor: Kibok Lee
- Email: kibok@yonsei.ac.kr
- Office: Daewoo Hall 428

- Teaching assistant: Seungju Yoo
- Email: seungju_yoo@yonsei.ac.kr

- Teaching assistant: Jaebyeong Jeon
- Email: jaebyeong98@yonsei.ac.kr

How to Contact Us

- via Classum (https://www.classum.com/main/course/113295)
  - All questions about the course should go here.
  - Students are encouraged to ask and answer each other.
  - We will also use this to communicate with you.

LearnUs YONSEI

- Email: Only for sensitive and/or confidential issues

Lecture Format

- Location: Daewoo Hall B103
- Time: Tue 1 PM – 3 PM, Thu 12 PM – 1 PM
- Office hours: 24/7 @ Classum
  - Or on-demand in-person office hours

- Slides are mostly self-contained.
  - I usually do not write in class.
- Solution for assignments/exams will not be shared.
  - You can ask specific questions via Classum.
  - We can solve them together upon request.

Lecture Format

- From the last course evaluation (max 5):

6. Non-Discrimination Principle : This course was conducted without biases or stereotypes based on gender, race, nationality or religion.	4.73
7. This course was conducted in English.	4.86

  - I know the questions are not important for someone
- This course is conducted **without biases or stereotypes** based on gender, race, nationality or religion.
  - If you feel it is not, please let us know.
- This course is conducted in **English**.
  - Outside of the course, you can ask us in any language for your better understanding. (as long as we understand it)
  - We hope language is not a bottleneck

Waitlist Policy

- We cannot manage many students because of the heavy workload on grading.
  - Grading results are often delayed.
- We expect some students to cancel this course, so you can get into the space.
- You can still take this course out of the capacity if
  - You are a student at this university.
  - You have a concrete reason that you cannot take this course next year or later.
  - Your background is qualified to take this course.
  - The department accepts your exception.

Prerequisites

- STA1001 Introduction to Statistics
- STA1002 Calculus
- STA2102 **Linear Algebra**
- STA2104 R and **Python** Programming
  - You should write Python codes for assignments
  - R is not required in this course
- (Highly Recommended) STA3135 **Data Structures and Algorithms** for Data Science
  - Or you can take similar courses like **CCO2103, EEE2020**
- Experiences taking other advanced courses
  - STA3126/3109/3125/3140/3141/4103, …

Prerequisites

- **Caution**: This is an advanced course!
  - Maybe one of the most challenging courses in the Department of Applied Statistics
- Students are expected to be familiar with
  - Undergraduate-level mathematics and **solving problems** based on it
    - E.g., derive and generalize least-squares under some conditions
  - **Implementing algorithms** in python from mathematical derivations
    - E.g., implement generalized least-squares above using numpy
- If you took **STA3143 Computer Vision for Data Science** successfully, you are (likely to be) qualified.

Prerequisites

- Suppose A in $R^{3\times 3}$, x in R^3. You should know:

- How do I calculate Ax?
- When is A invertible?
- What is x if $Ax = \lambda x$ for some λ?
- What is the set $\{y: x^T y = 0\}$ geometrically?
- How do I calculate $\frac{\partial A_{ij}}{\partial x}$?

What You Are Going to Do

- Example: Math

Tossing a (potentially unfair) coin can be modeled as a Bernoulli distribution. Suppose X is a random variable of the result of tossing a potentially unfair coin, such that $P(X = H) = \mu$ is the unknown probability that the coin toss outcome is head, and $P(X = T) = 1 - \mu$ is the probability that the coin toss outcome is tail. This can be modeled as a Bernoulli distribution, *i.e.*, $X \sim \text{Ber}(\mu)$, with the parameter $\mu > 0$, and it has the probability mass function

$$p(x|\mu) = \mu^{\mathbb{I}(X=H)}(1-\mu)^{\mathbb{I}(X=T)},$$

where $\mathbb{I}(\cdot)$ is the indicator function. Suppose $\{x_i\}_{i=1}^N$ are the trials, and denote $\mathbf{x} = [x_1, x_2, \cdots, x_N]^\top$.

(3.1) **[4 points]** Find an expression for the negative log-likelihood function of μ based on the observations $\mathbf{x}$. Show that it is convex for μ, *i.e.*, the second derivative is non-negative.

(3.2) **[4 points]** Find the maximum likelihood estimator (MLE) of μ based on the observations $\mathbf{x}$.

(3.3) **[6 points]** Suppose μ has a Beta prior $\mu \sim \text{Beta}(\alpha, \beta)$ with the probability density function

$$p(\mu|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\mu^{\alpha-1}(1-\mu)^{\beta-1},$$

where $\alpha, \beta > 0$ are fixed parameters, and $\Gamma(z) = \int_0^\infty x^{z-1}\exp(-x)dx$ is the Gamma function. Find the maximum a posteriori (MAP) estimate of μ based on the observations $\mathbf{x}$. [Hint: The Gamma distribution $p(\mu|\alpha, \beta)$ is a conjugate prior of the Bernoulli distribution $p(x|\mu)$.]

(3.4) **[2 points]** Suppose repeatedly tossing a coin with unknown probability of heads μ produces the data sequence HHTTTTHTHT, where H is the head and T is the tail. What is the MLE of μ?

What You Are Going to Do

- Example: Programming

```python
def sgd(x_train, y_train, x_test, y_test, lr, num_epochs, M=1, lam=0.):
    """
    Linear regression with stochastic gradient descent (retrieve each sample).
    Retrieve each sample using for-loop, and compute MSE after each epoch.
    You don't have to shuffle the order of retrieval.

    Inputs:
    - x_train: A numpy array of shape (N,)
    - y_train: A numpy array of shape (N,)
    - x_test: A numpy array of shape (N,)
    - y_test: A numpy array of shape (N,)
    - lr: A positive real number giving the learning rate
    - num_epochs: The number of training epochs
    - M: An integer giving the number of polynomials. default: 1
    - lambda: A positive real number giving the L2 regularization coefficient.
      default: 0.0

    Returns a tuple of:
    - w: A numpy array of shape (M+1,) giving the learned weights
    - train_error: The RMS error computed with training data after training
    - test_error: The RMS error computed with test data after training
    - rmse_list: A list giving historical training errors in terms of MSE
    """
    w = np.zeros(M+1)
    train_error = test_error = 0.
    N = x_train.shape[0]
    rmse_list = []
    ###########################################################################
    # TODO: Implement linear regression with stochastic gradient descent.    #
    # Hint: rms_error                                                        #
    ###########################################################################
    # Replace "pass" with your code (do not modify this line)
    pass
    ###########################################################################
    #                             END OF YOUR CODE                            #
    ###########################################################################
    return w, train_error, test_error, rmse_list
```

Rough Plan

- We have **13 weeks (39 hours)** of lectures.

- **6 hours** for intro & math/python review
- **15 hours** for basic ML & supervised learning
  - Regression, classification, kernel methods, validation, …
- **3 hours** for unsupervised learning
- **12 hours** for neural networks
- **3 hours** for others
  - Reinforcement learning, summary
  - Note: You can take **STA3145** for RL

Grading Policy

- **Assignments**: 50%
  - There will be 4 problem sets
  - Each set includes math/python programming problems
- **Midterm exam**: 25%
  - Written exam
  - True/False, multiple choice, math (proof)
- **Final ~~exam~~ assignment**: 15%
  - Another problem set; no class in the final exam week
- **Attendance**: 10%
- **Absolute evaluation**
  - Scaling up based on a curve if the statistics is low

Late Assignment Policy

- **[-25%]** additive for each late day
  - No point if late for >= 4 days

- No deduction if you completed your work on time but are late for few minutes because of an unexpected issue, e.g., network latency.
  - But do not assume that you always have extra minutes.

- **Do NOT contact us to explain why you were late.**
  - You can still leave a comment in your LearnUS submission, but it can be ignored.

Attendance Policy

- [-1] for each absence (no deduct for 1st week)
  - No late check
- Attendance check via **Y-attend app**
- **Option 1**: [3 free absences] without any report
  - You should not contact us for any attendance issue.
    - No exception; please be wise!
    - The score is *min(10, 13 – N)* if you are absent *N* times.
- **Option 2**: [no free absence]; but you can submit eligible docs to make up your score.
  - Activated if you contact us for any attendance issue.
- If you are absent **1/3** of the course, you will get [F].
  - According to the university policy.

Study Group

- We encourage students to form study groups.
  - Up to **five** people are allowed for each group.
- You may discuss with the study group members on assignments. But you should write your own solution **independently**.
  - If your solution depends on others, go for **option 2 (next)**
- When submitting assignments, put the names of people with whom you have discussed.
- Please start on assignments early.
  - **Warning**: cramming does not work!

Honor Code

- In addition to the Yonsei University's honor code,
  - Do not give or receive unpermitted aid in exams, assignments, or any other work that is to be used by the instructor as the basis of grading.
  - Do not share solutions or code with other students.
- Plagiarism is obvious and painful; we don't have many options if any plagiarism is found. Instead,
  - You can submit your work late with some deduction.
  - You don't even have to solve some questions.
  - You'd better to get a low score than to ruin your academic record.
  - <u>Note: If something is difficult for you, then so are others.</u>

Assignment Options

- **Option 1**: Submit your own work
  - You must not copy/refer to others' solution/code.
  - You will get **[F]** if any plagiarism is found.
- **Option 2**: Refer to others' solution/code
  - You need to **cite** references clearly.
    - Website address, query to LLMs, your study group members
    - **[No point]** if you do not provide references properly
  - For codes, **your comments** is required/graded.
    - **[No point]** if you do not provide any comments
  - Your score will be downscaled to **[70%]**.
    - e.g., if you take this option for 3rd, 4th, and the final assignments, the maximum score you can get is 12.5*(2 + 2*.7) + 15*.7 = 53 (out of 65)
  - You can apply this question-wise (Not subquestion-wise)

Can I use large language models?

- You can get an assistance by large language models (LLMs) like ChatGPT, which falls into **option 2**.

- Clearly note that you used LLMs and provide your queries together.

- This policy can be changed for later assignments.
  - If we found it makes you complete assignments without understanding

Plagiarism: Case Study

1. I don't know how to use a numpy function, so I googled it and carefully read the documentation.
   - This is encouraged. Please do so!

2. I found a (partial) solution on a website.
   - You are not recommended to carefully look at it, as your solution might be unintentionally affected.
   - If you referred to it, consider to go for **option 2**.

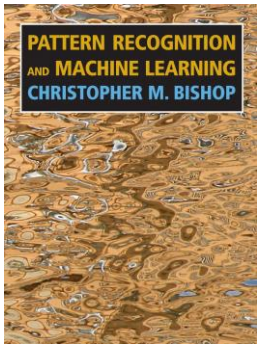
Plagiarism: Case Study

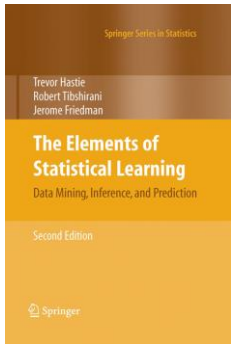
3. I don't know what the steps are to implement an algorithm for Q2-2, so I discussed with my friend.
   - This is encouraged. But don't share code directly.

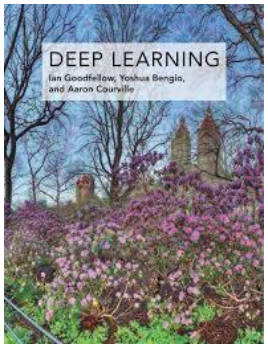
4. I don't know how to implement an algorithm for Q2-3, so I asked my friend for help.
   - If you wrote your own code, then it is likely to be fine.
   - If you referred to their code while writing yours, consider to go for **option 2**.

Reference

- EECS545 Machine Learning @ University of Michigan
- CS229 Machine Learning @ Stanford University
  - https://cs229.stanford.edu/
- No official textbook, but for your reference:
  - Bishop, Pattern Recognition and Machine Learning. *Springer*, 2006.
    - https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/
  - Hastie et al., The Elements of Statistical Learning. Springer, 2009.
    - https://hastie.su.domains/ElemStatLearn/
  - Goodfellow et al., Deep Learning. *MIT Press*, 2016.
    - https://www.deeplearningbook.org/







Policy Summary

- 5 Assignments (**65**), Midterm (**25**), Attendance (**10**)
  - No final exam
- Attendance: **[-1]** for each absent; **no late check**
  - **Option 1 (Default)**: **[3 free absences]** without any report
  - **Option 2**: **[no free absence]**; docs to make up your score
- Assignment: **[-25%]** additive for each late day (not counting seconds)
  - **Option 1 (Default)**: Submit your own work
  - **Option 2**: Refer to others' solution/code and get **[x70%]**
- Study group size **[≤ 5]**
- Intuitively, we do not want to spend time for any non-academic issue

Outline

- Course Information
- What is Machine Learning?

Machine Learning

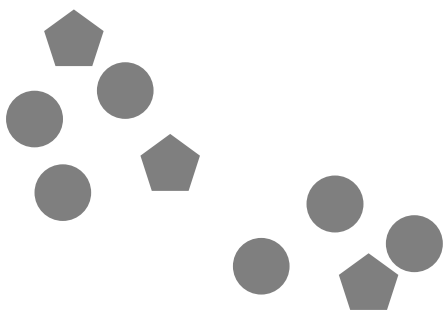
Formal definition from Mitchell (1997):

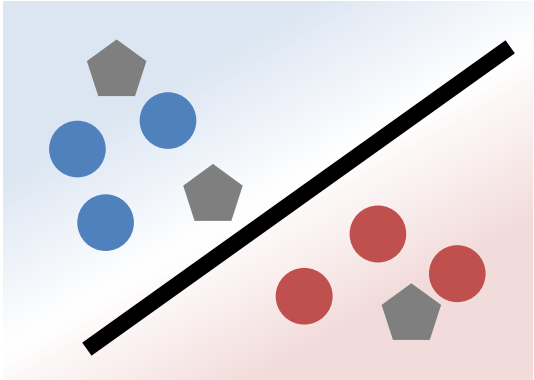
- A computer program **A** is said to **learn** from <u>experience</u> **E** with respect to some class of <u>tasks</u> **T** and <u>performance</u> measure **P**, if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.

Machine Learning

- Learning a model $f: x \rightarrow y$ with training data $\{(x_i, y_i)\}_{i=1}^{N}$ s.t. it generalizes well on unseen data
  - Training data:
  - Labels: or
  - Test data:

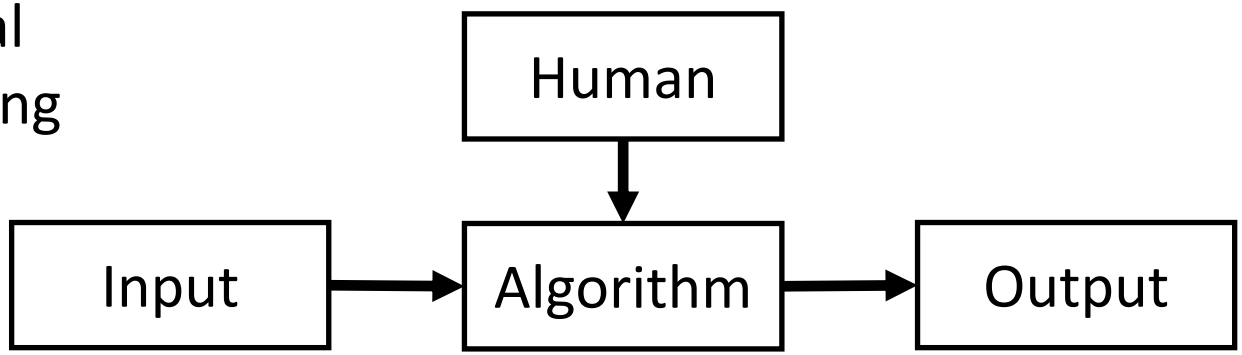
$$x \Rightarrow \boxed{f} \Rightarrow y$$





Machine Learning vs. Programming

Traditional Programming



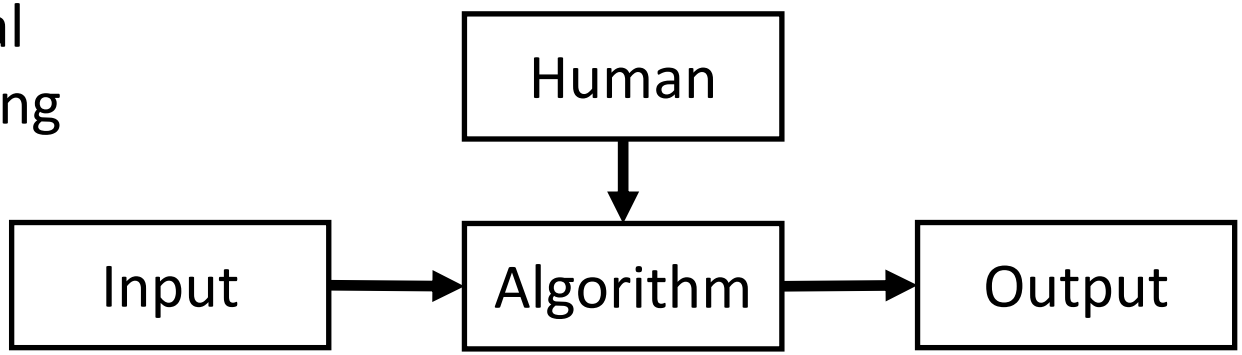
Works well for sorting numbers

```
def bubble_sort(arr):
  N = len(arr)
  for i in range(N - 1):
    for j in range(N - i - 1):
      if arr[j + 1] < arr[j]:
        temp = arr[j]
        arr[j] = arr[j + 1]
        arr[j + 1] = temp
  return arr
```

Slide Credit: Justin Johnson

Machine Learning vs. Programming

Traditional Programming



Much harder for some problems

```
def cat_or_dog(image):
  if ????:
    return "cat"
  else:
    return "dog"
```

Slide Credit: Justin Johnson

Example: Image Classification

Input: image

This image by Nikita is licensed under CC-BY 2.0

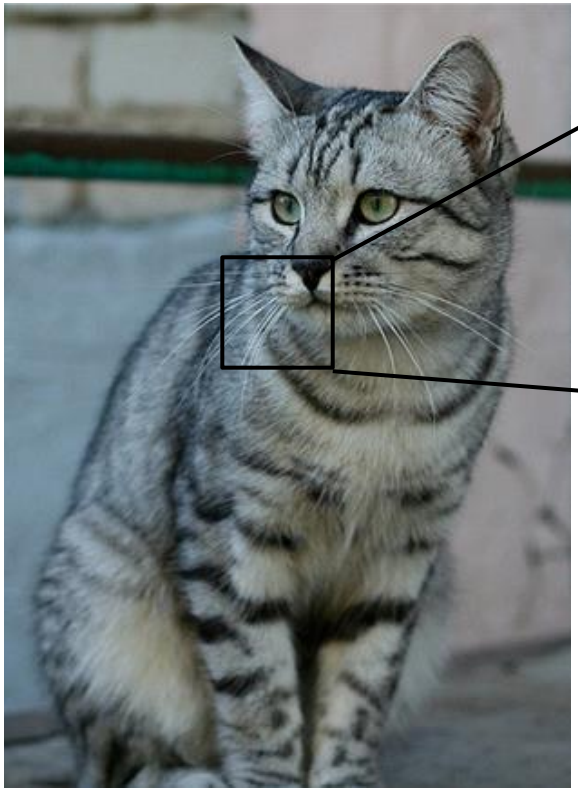
Output: Assign image to one of a fixed set of categories

cat
bird
deer
dog
truck

Slide Credit: Justin Johnson

Problem: Semantic Gap

Input: image



This image by Nikita is licensed under CC-BY 2.0

```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

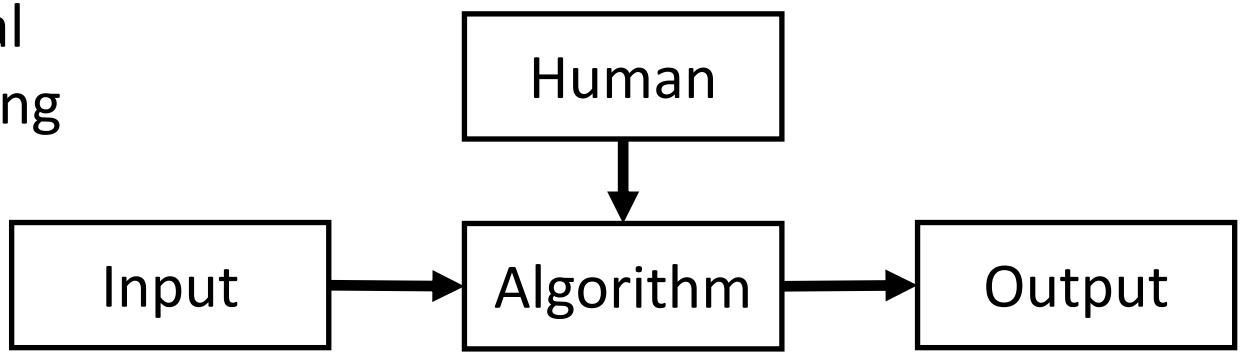
What the computer sees

An image is just a big grid of numbers between [0, 255]:

Slide Credit: Justin Johnson

Machine Learning vs. Programming

Traditional Programming



Find edges

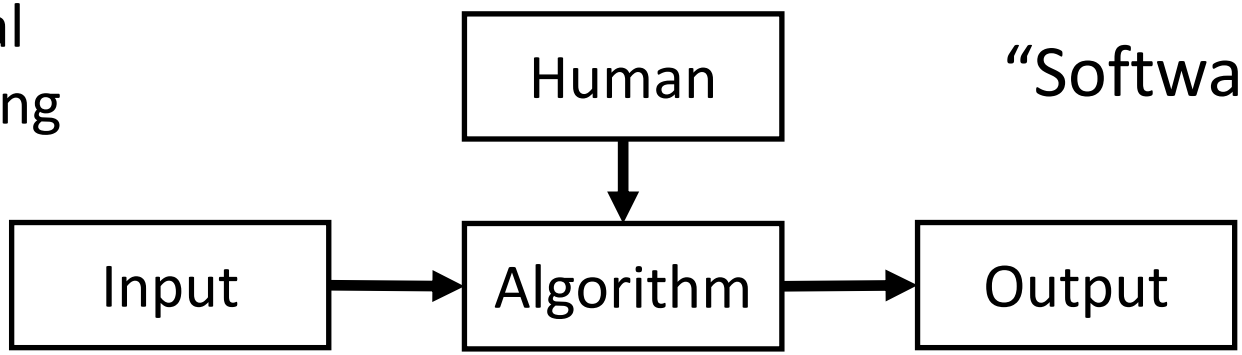
Look for ears, whiskers, etc

Problems:
Very brittle
What about car vs. truck?

Slide Credit: Justin Johnson

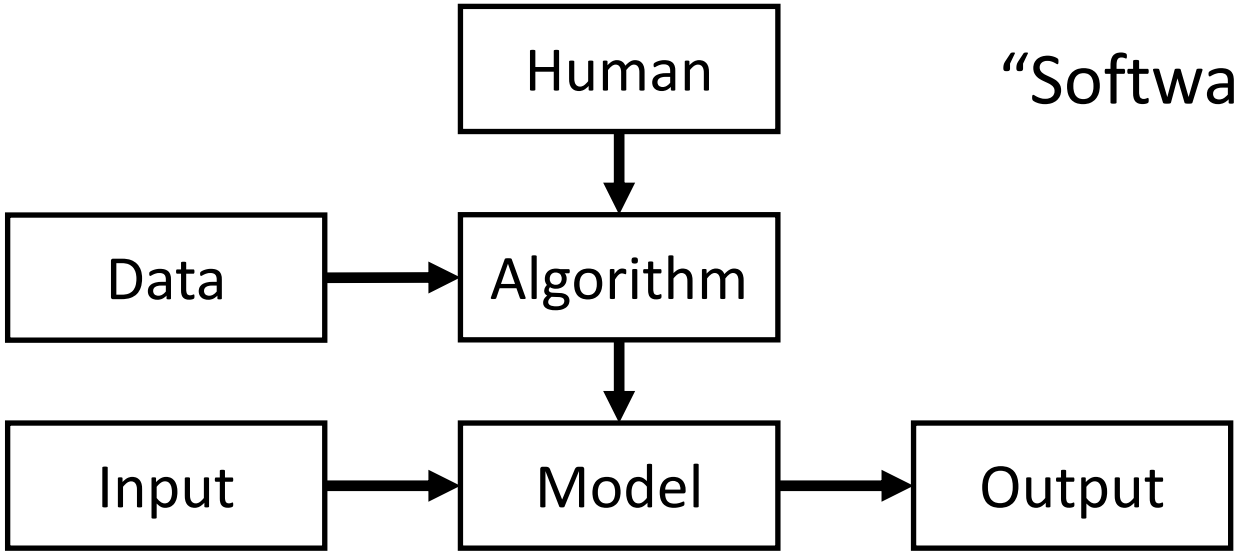
Machine Learning vs. Programming

Traditional Programming



"Software 1.0"

Machine Learning



"Software 2.0"

https://medium.com/@karpathy/software-2-0-a64152b37c35

Slide Credit: Justin Johnson

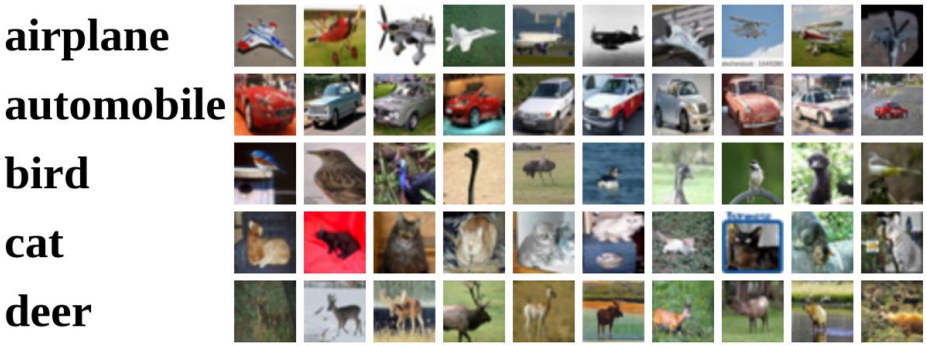
Machine Learning: Data-Driven Approach

1. Collect a large set of data
2. Use Machine Learning to train a model
3. Evaluate the model on new data

```
def train(images, labels):
  # Machine learning!
  return model
```

```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

Example training set



Slide Credit: Justin Johnson

Machine Learning Tasks

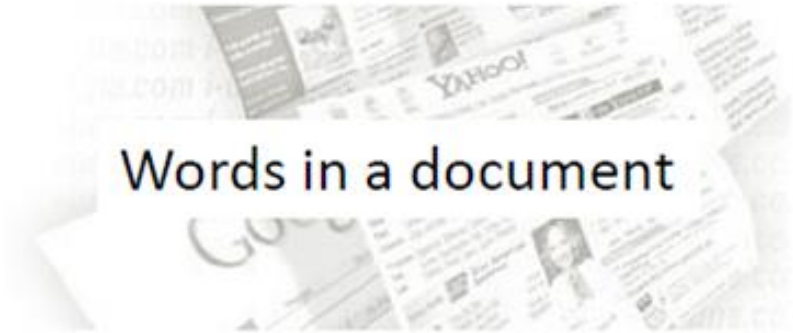
- **Supervised Learning**
  - Classification
  - Regression
- **Unsupervised Learning**
  - Clustering
  - Density estimation
  - Embedding / Dimensionality reduction
- **Reinforcement Learning**
  - Learning to act
    (e.g., robot control, decision making, etc.)

Supervised Learning

- Given a dataset $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, where
  - $x_i \in \mathcal{X}$: input (feature)
  - $y_i \in \mathcal{Y}$: output (label)
- A black box ML algorithm produces a prediction function $h: \mathcal{X} \to \mathcal{Y}$, such that $h(x)$ can predict the y values for all x
  - Not only for all training data x_i,
    but also for unseen test data $x^* \in \mathcal{X}$.
- Labels could be discrete or continuous
  - Discrete labels: **classification**
  - Continuous labels: **regression**

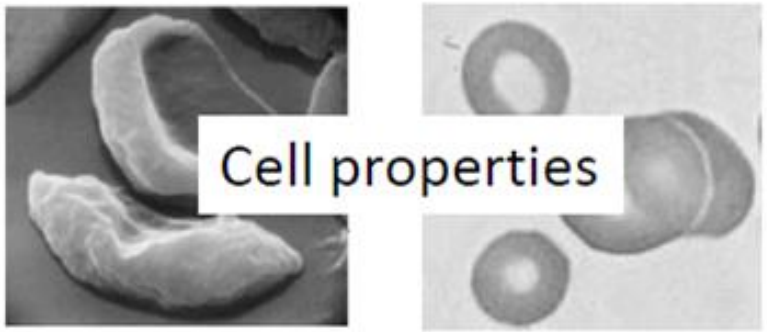
Supervised Learning: Classification

Feature Space $\mathcal{X}$



Label Space $\mathcal{Y}$

"Sports"
"News"
"Science"
...



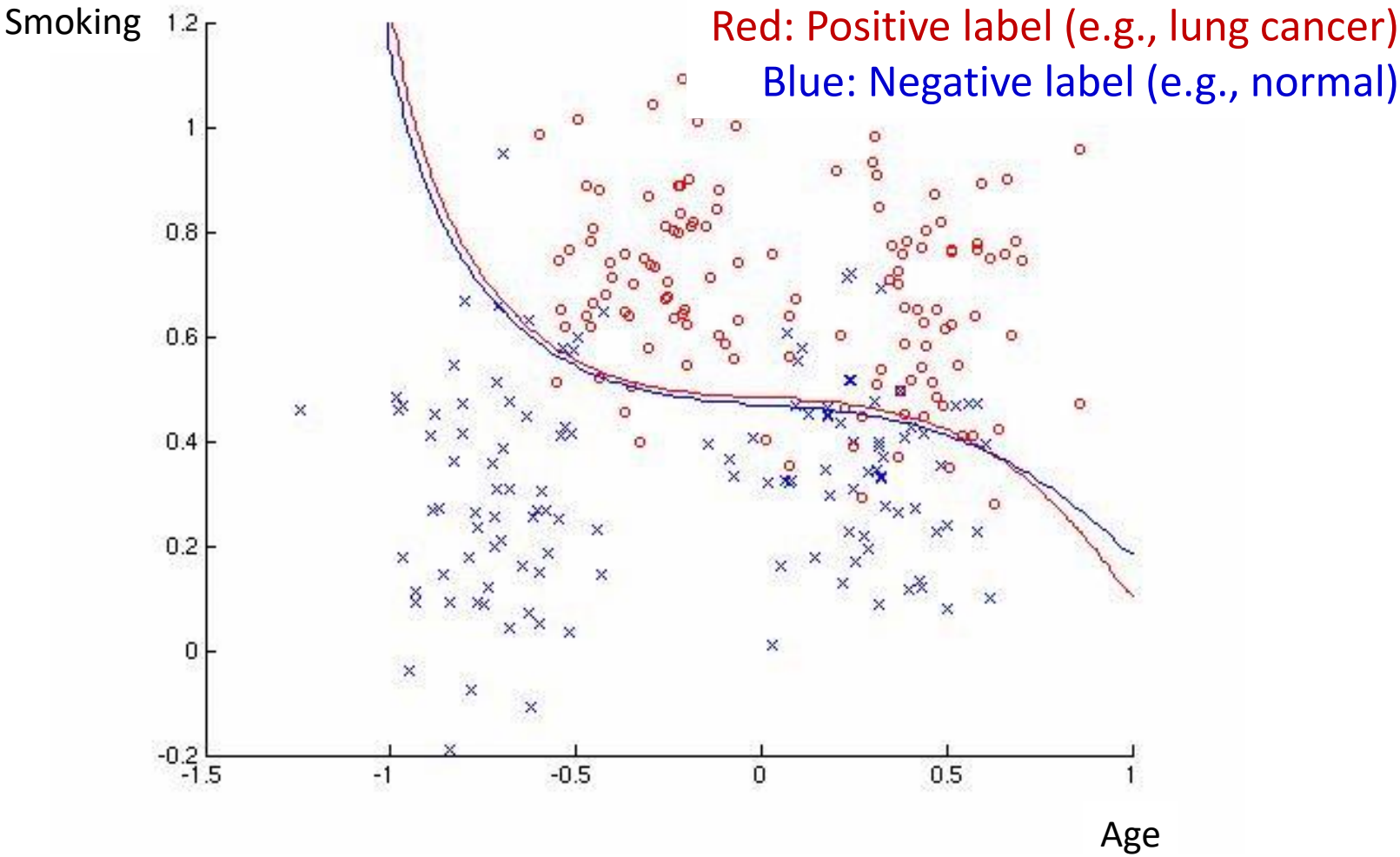
"Anemic cell"
"Healthy cell"

Discrete Labels

Slide credit: Aarti Singh

Supervised Learning: Classification

- Learning decision boundaries



Supervised Learning: Classification

- Input: Image (and/or its feature)
- Output: Label

"Cat"

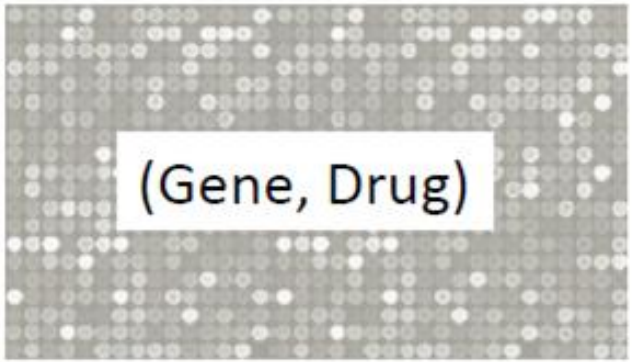
This image is CC0 public domain

Supervised Learning: Regression

Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$

Share Price
"$ 24.50"



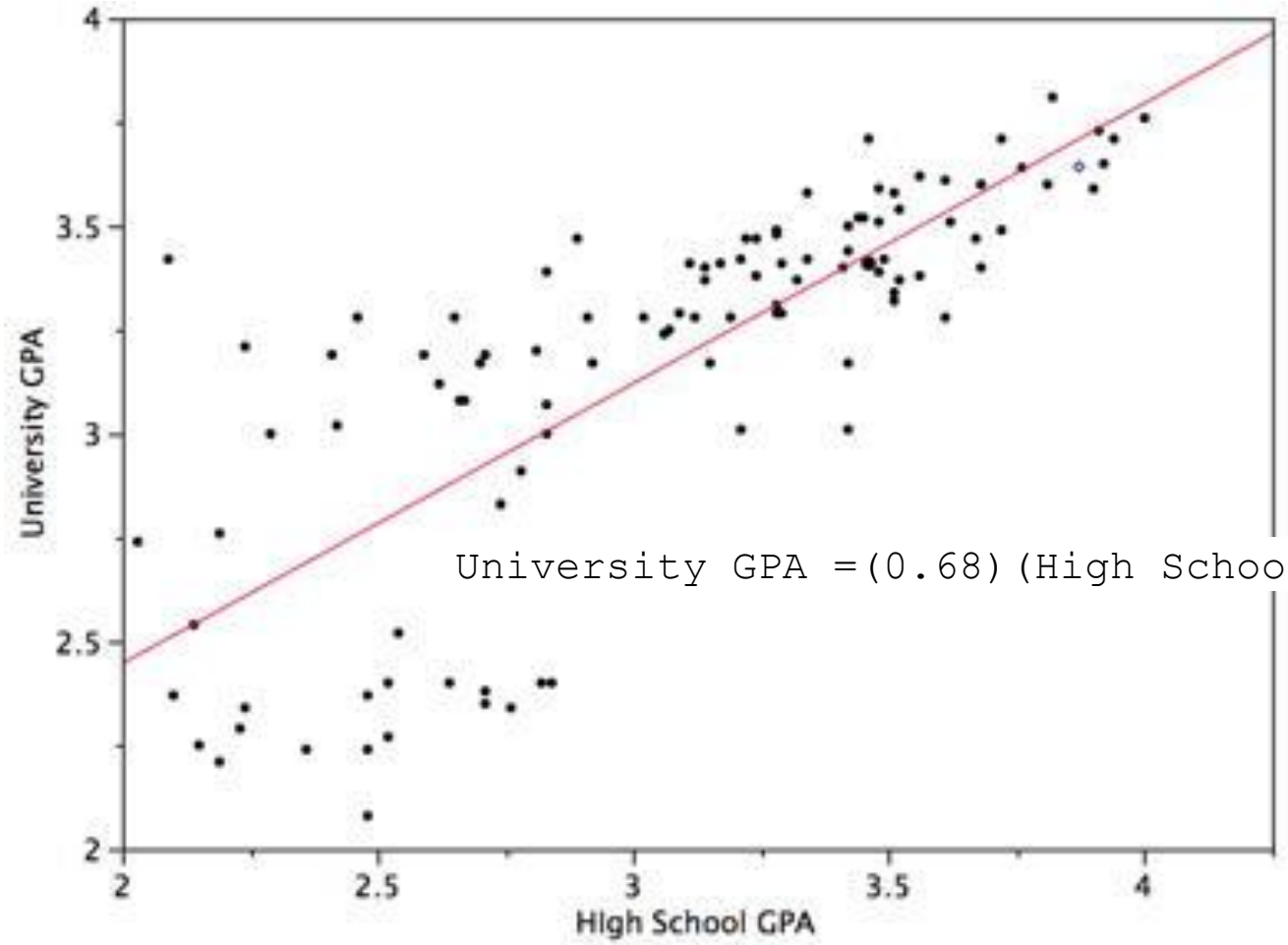
Expression level
"0.01"

Continuous Labels

Slide credit: Aarti Singh

Supervised Learning: Regression

- Learning a regression function

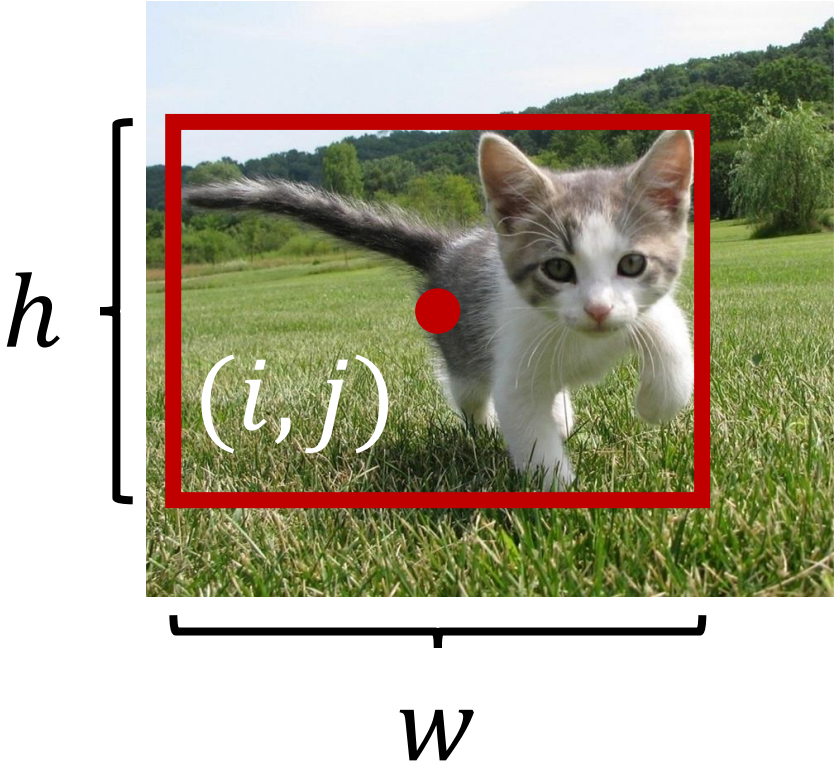


University GPA =(0.68)(High School GPA)+ 1.1

Slide credit: David M. Lane

Supervised Learning: Regression

- Input: Image (and/or its feature)
- Output: Bounding box (i, j, h, w)



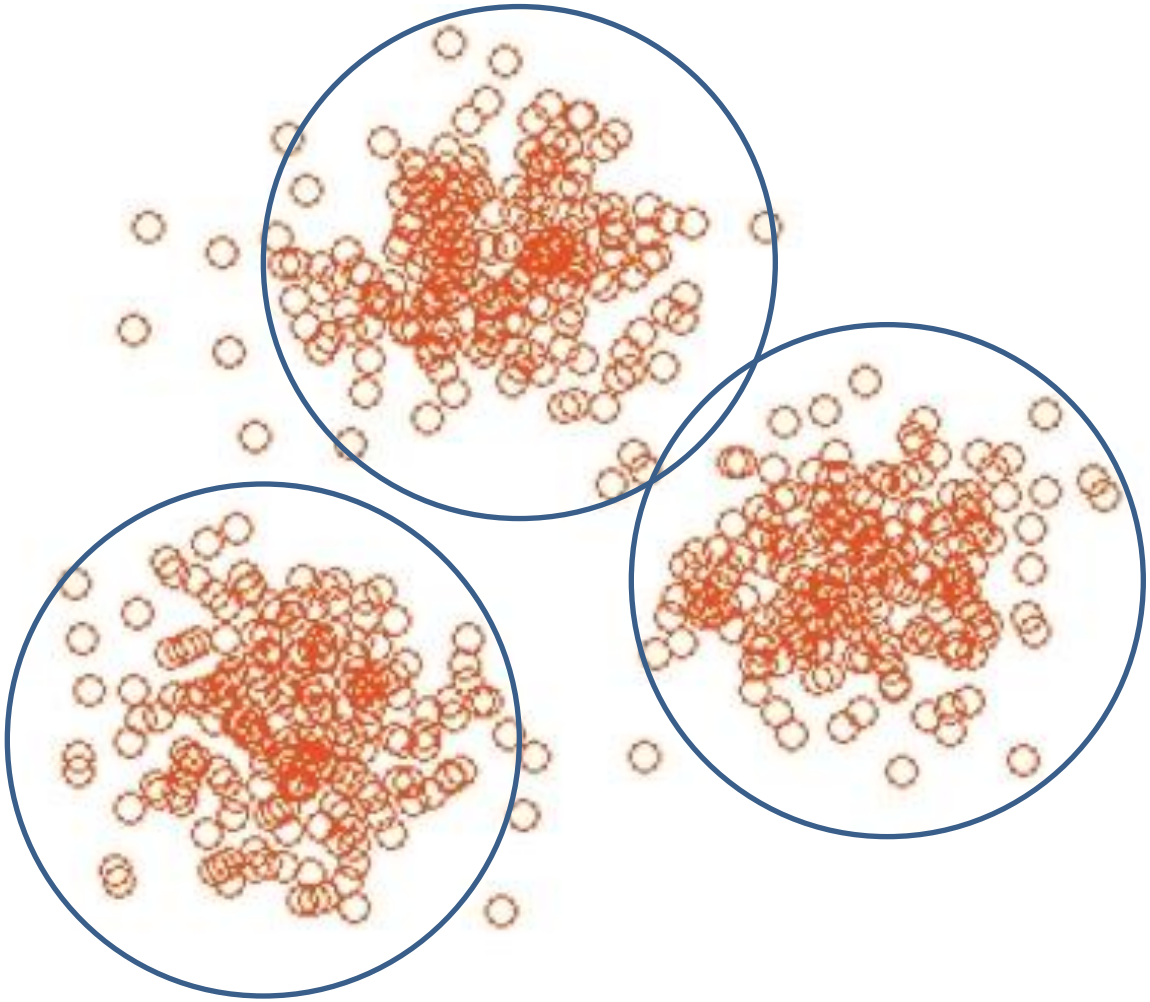
This image is CC0 public domain

Unsupervised Learning

- Given a dataset $D = \{x_1, \dots, x_n\}$ <u>without any labels</u>, learning the underlying **structure** or **distribution** of the data
  - Clustering
  - Probability distribution (density)
  - Generating data
  - Embedding & neighborhood relations
- "Learning without teacher (supervision)"

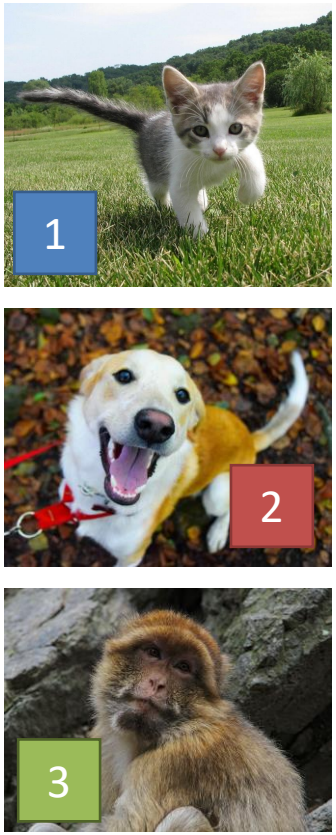
Unsupervised Learning: Clustering

- Grouping into similar examples



Unsupervised Learning: Clustering

- Grouping into similar examples



Cat image is CC0 public domain
Dog image is CC0 Public Domain
Monkey image is CC0 Public Domain

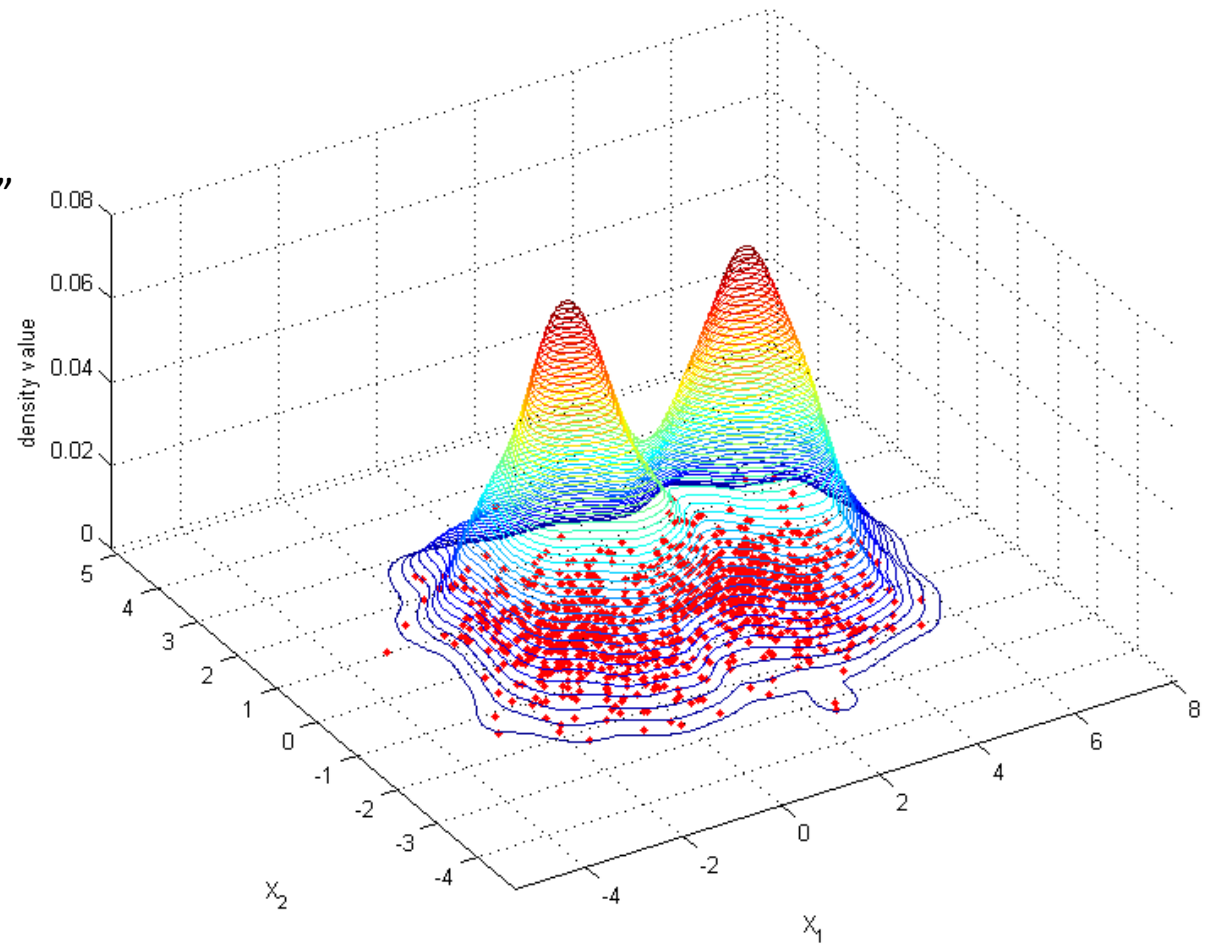
Cat image is CC0 public domain
Monkey Image is CC0 public domain

Cat image is CC0 public domain
Dog image is CC0 public domain
Dog image is CC0 public domain

Slide Credit: Justin Johnson

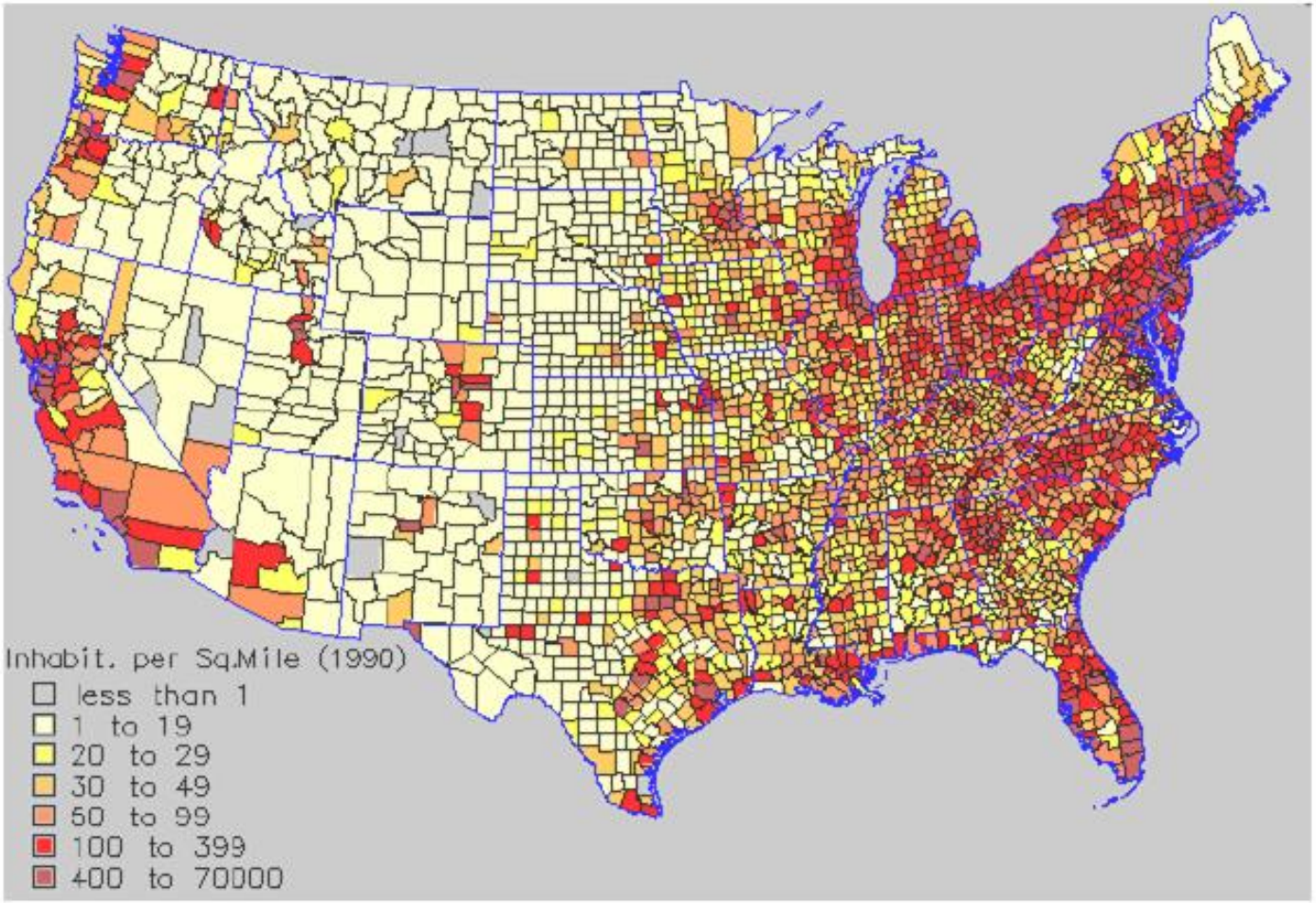
Unsupervised Learning: Density estimation

P(X1, X2)
"Probability"



Unsupervised Learning: Density estimation

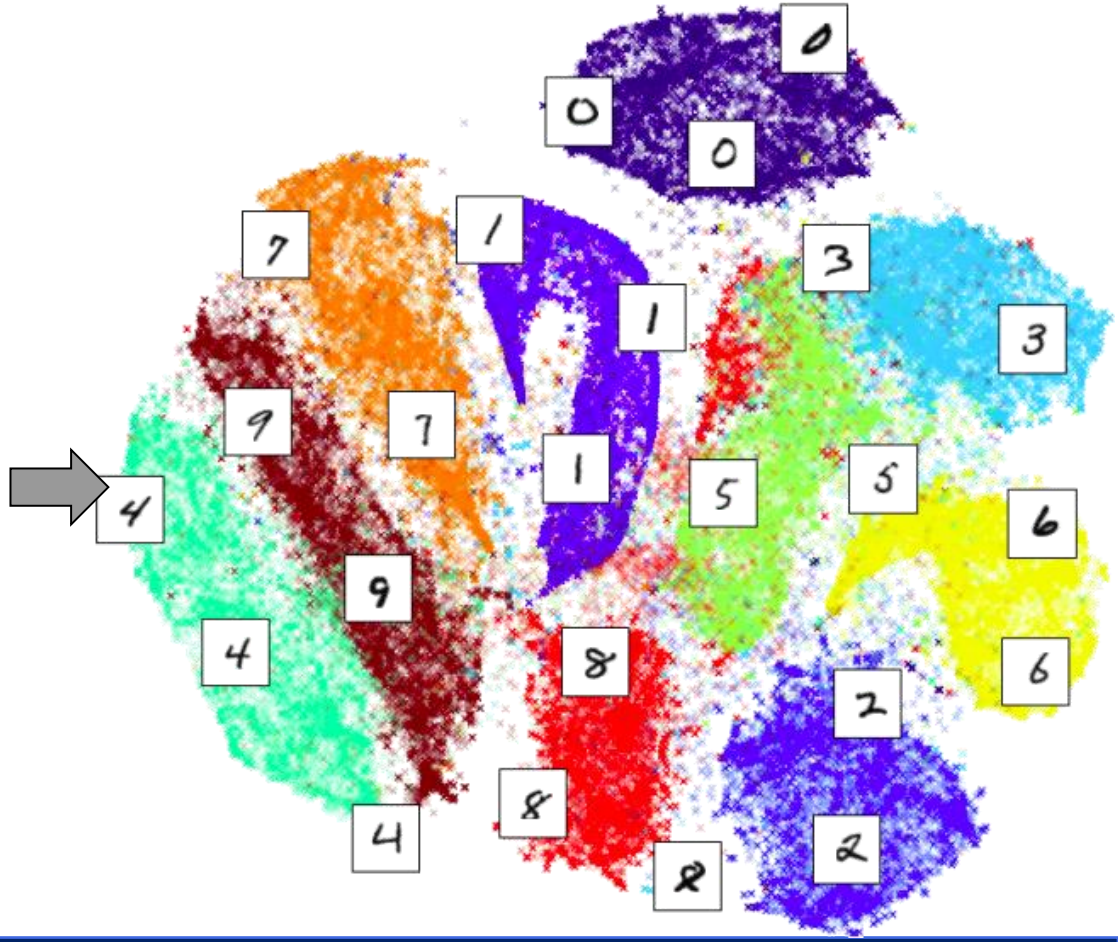
Population density



Slide credit: Aarti Singh

Embedding and Dimensionality Reduction

- **Example**: reducing handwritten digits (784-dim) into low dimensional coordinates



[Maaten and Hinton, 08]

Machine Learning Problems

	Supervised (Data + Labels)	**Unsupervised (Just Data)**
Discrete Output	Classification	Clustering
Continuous Output	Regression	Dimensionality Reduction

- This is not exhaustive; there are other ML tasks.

Slide adapted from J. Hays

Reinforcement Learning

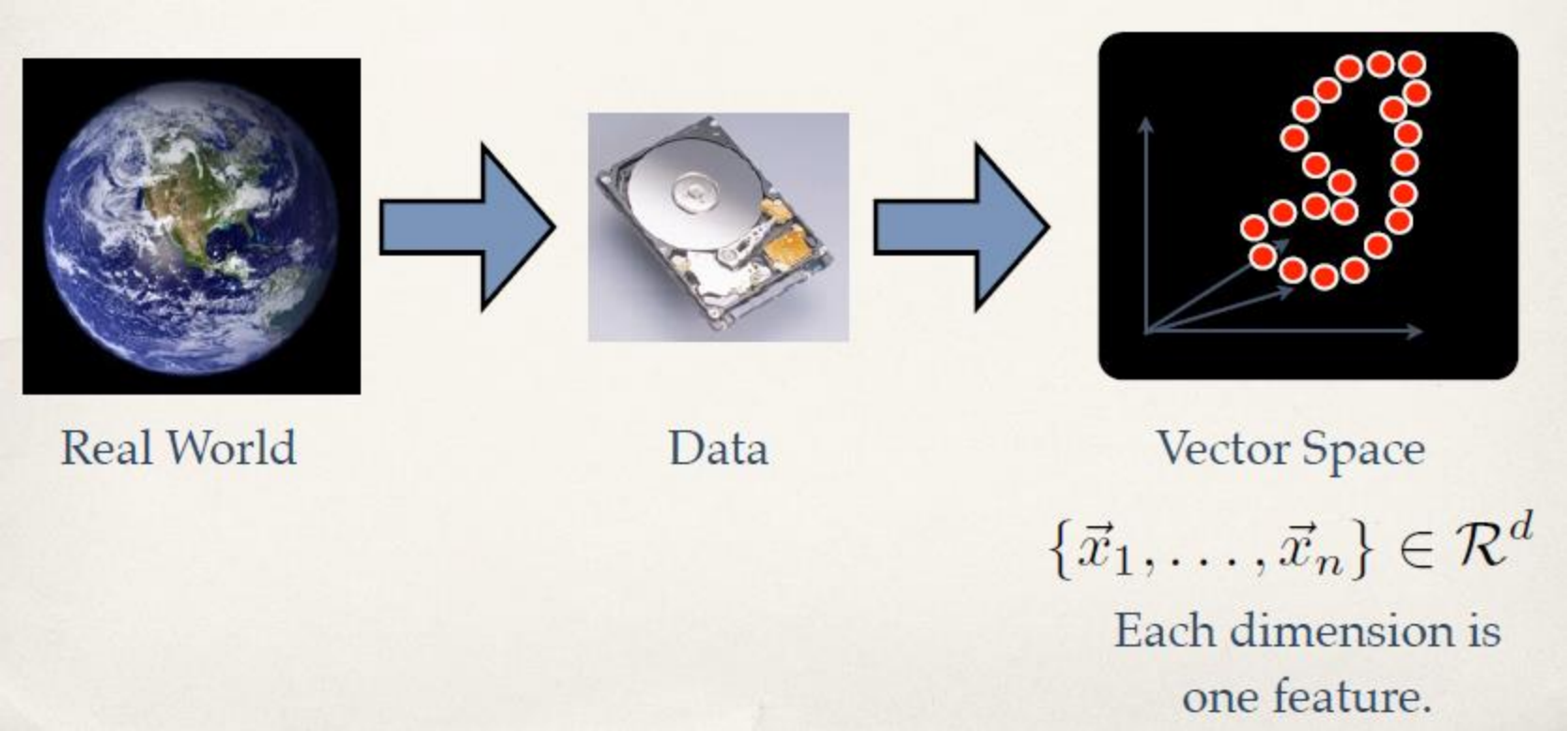
- **Setting**: Given sequence of states X and "rewards" (e.g., delayed labels), agent has to take actions A for each time step
- **Goal**: How to "learn to act" or "make decisions" to maximize the sum of future rewards
- **Example**: Robot navigation task
  - **Input**: Dynamical environment + sensor input
  - **Action**: control signals
  - **Rewards**: time to reach goal without colliding with obstacles

Reinforcement Learning

- **Example**: Robot walking
  - **States**: sensor inputs, joint angles
  - **Action**: servo commands for joints
  - **Rewards**:
    - 1 for reaching the goal
    - -1 for falling down
    - 0 otherwise
- Goal: How can we provide control inputs to maximize the expected future rewards?

Feature Extraction

- Represent data in terms of vectors.
  - Features are statistics or attributes that describe the data.

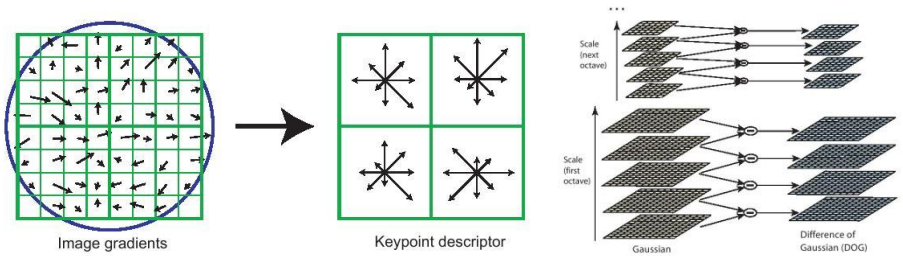


Slide credit: Kilian Weinberger

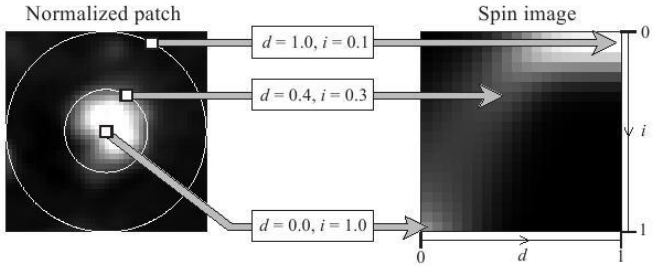
Feature Extraction Example

- **Input**: 28x28 handwritten digit images
- **Output**: class labels $\in \{0, 1, \ldots, 9\}$
- **Feature candidates**:
  - Pixel values $\in [0,255]^{784}$
  - Existence of long vertical strokes
  - Existence of long horizontal strokes
  - Aspect ratio of the tight bounding boxes
  - …

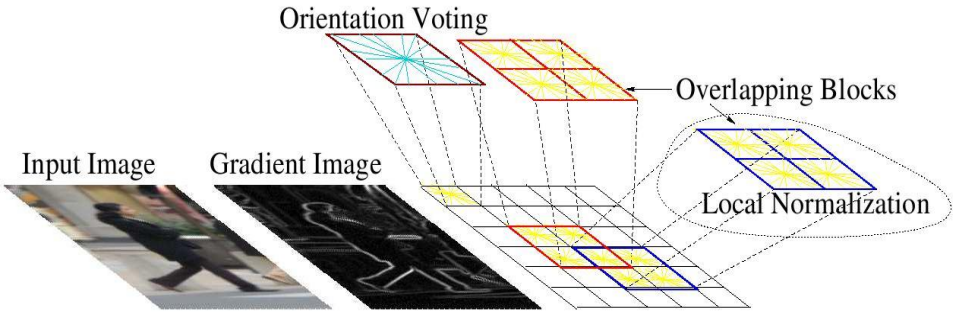
Traditional Features



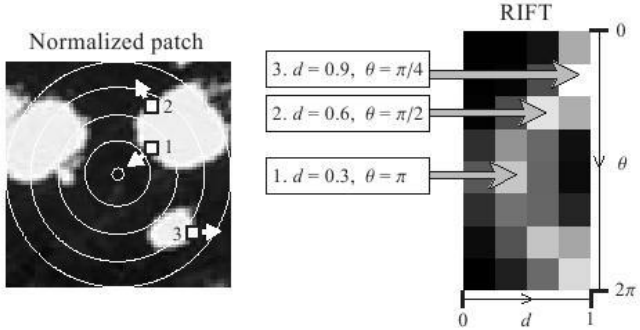
SIFT



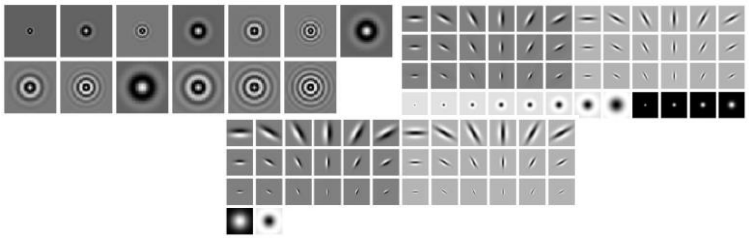
Spin image



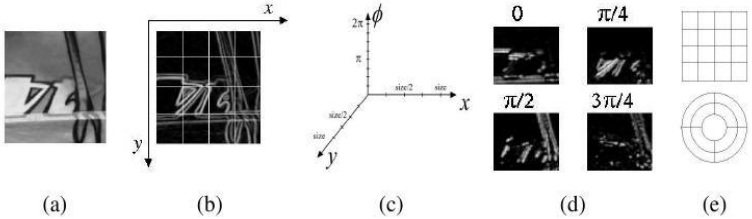
HoG



RIFT

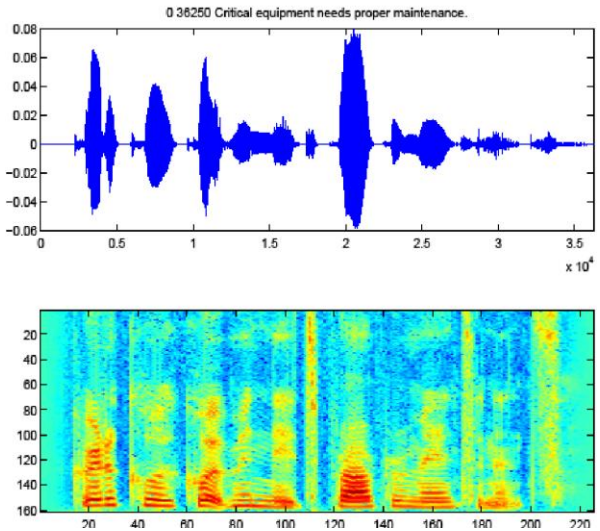


Textons

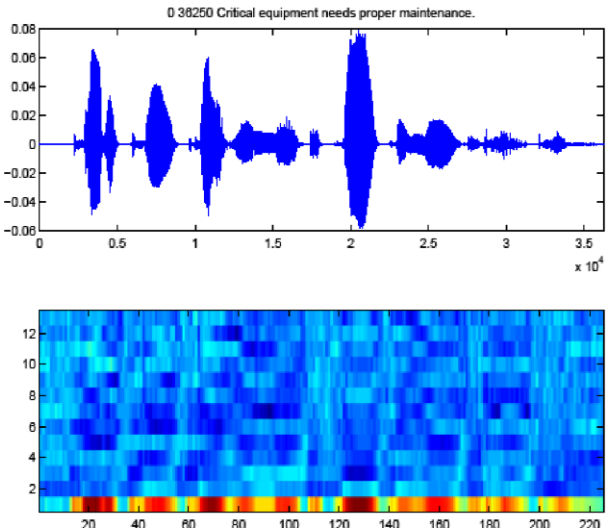


GLOH

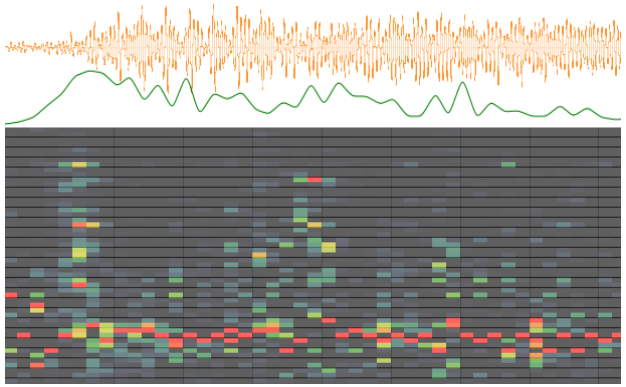
Traditional Features



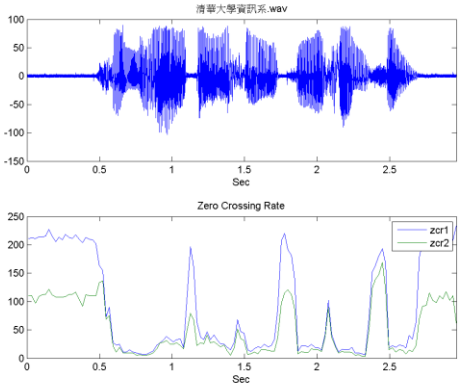
Spectrogram



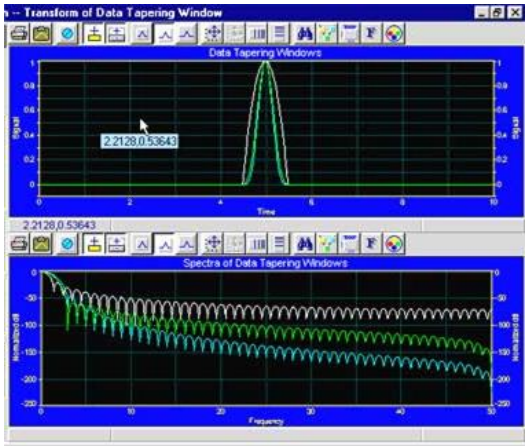
MFCC



Flux

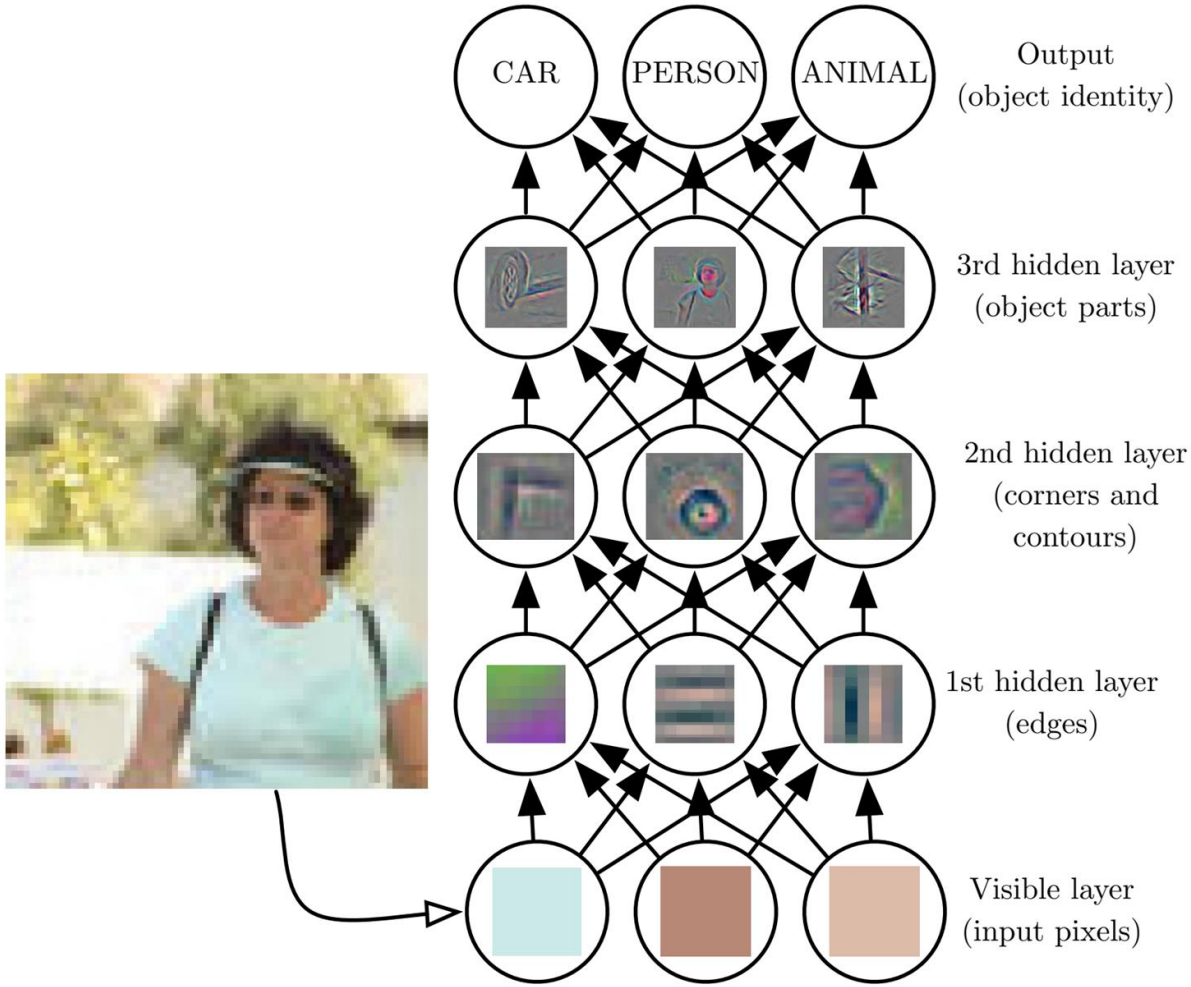


ZCR



Rolloff

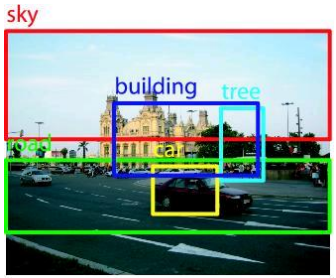
Deeply-Learned Features

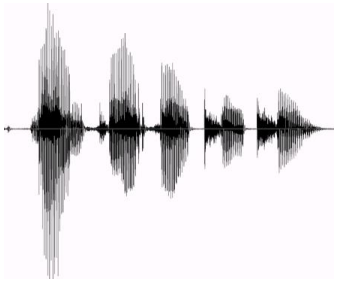


Source: Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning, 2016

Machine Learning Applications

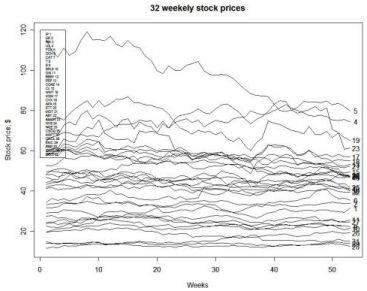
- Computer vision
- Natural language processing
- Speech recognition
- Robotics
- Data mining
- Medical image recognition
- Time series forecasting
- …

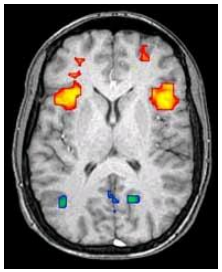




WIKIPEDIA

REUTERS





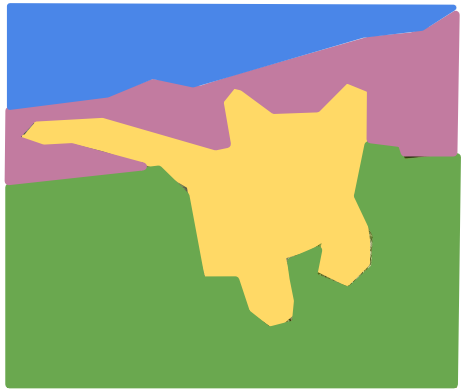
Computer Vision Tasks

Image Classification

CAT

No spatial extent

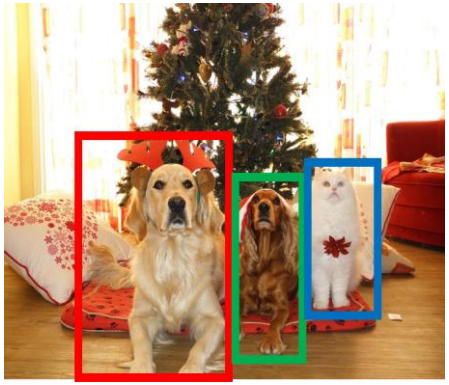
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

Instance Segmentation

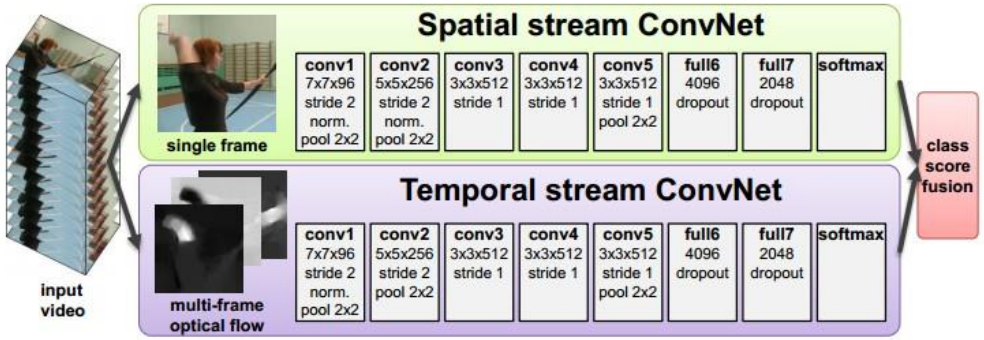
DOG, DOG, CAT

Multiple Objects

This image is CC0 public domain

Computer Vision Tasks

Video Classification



Simonyan et al, 2014

Activity Recognition

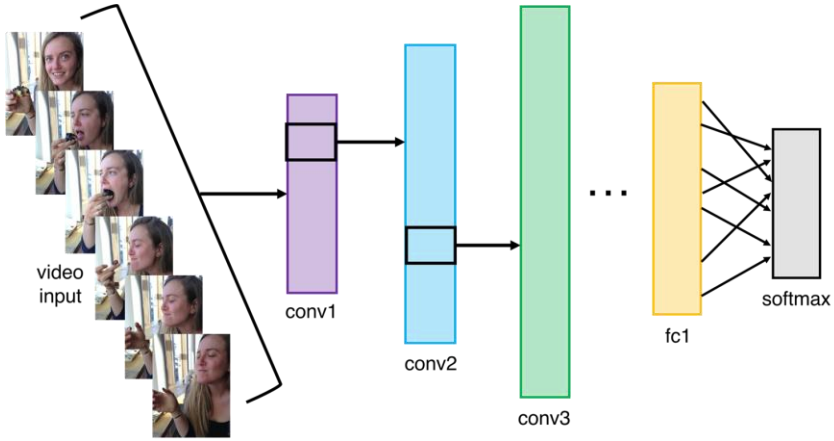


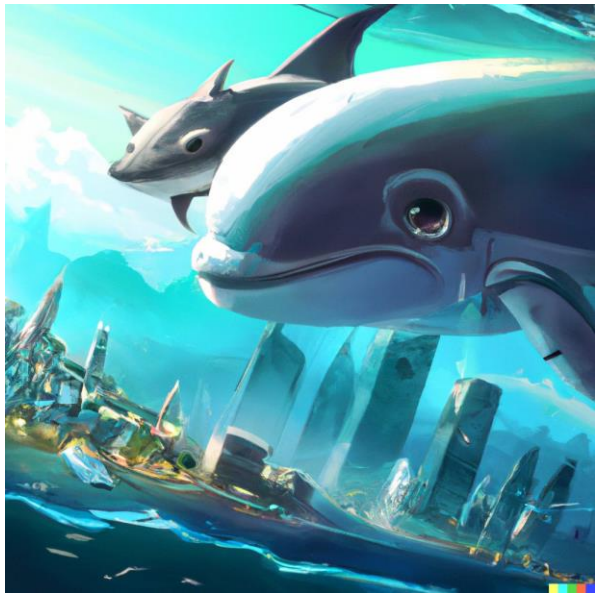
Image Generation

Karras et al, “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018

DALL-E 2: Text-to-Image Generation

A rabbit detective sitting on a park bench and reading a newspaper in a victorian setting

A shark and a dolphin cruise hand-in-hand with an undersea city in the background



Robot dinosaurs versus monster trucks in the colosseum

Ramesh et al., "DALL·E 2." 2022. https://openai.com/dall-e-2/
Source: https://twitter.com/sama/status/1511724264629678084

DALL-E 3: Image Generation with Recaptioning

A rabbit detective sitting on a park bench and reading a newspaper in a victorian setting

A shark and a dolphin cruise hand-in-hand with an undersea city in the background

Robot dinosaurs versus monster trucks in the colosseum

Betker et al., “Improving Image Generation with Better Captions.” 2023.
Source: Kibok Lee subscribing ChatGPT Plus

DALL-E 3: Image Generation with Recaptioning

A whimsical illustration of a rabbit detective dressed in Victorian attire, complete with a trench coat and a magnifying glass tucked into its pocket

An enchanting underwater scene where a stylized cartoon shark and dolphin are depicted with human-like features allowing them to 'hold hands'. They ar…

A dynamic and action-packed scene set inside the ancient Colosseum, where futuristic robot dinosaurs face off against rugged monster trucks. The robot …

Betker et al., “Improving Image Generation with Better Captions.” 2023.
Source: Kibok Lee subscribing ChatGPT Plus

Phenaki: Text-to-Video Generation

- A photorealistic teddy bear is swimming in the ocean at San Francisco
- The teddy bear goes under water
- The teddy bear keeps swimming under the water with colorful fishes
- A panda bear is swimming under water

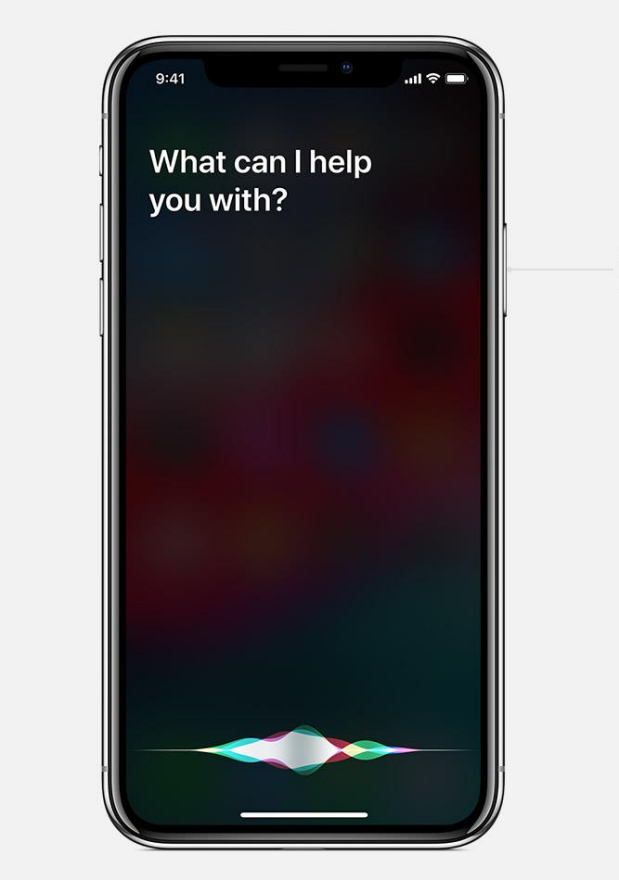
- A teddy bear diving in the ocean
- A teddy bear emerges from the water
- A teddy bear walks on the beach
- Camera zooms out to the teddy bear in the campfire by the beach

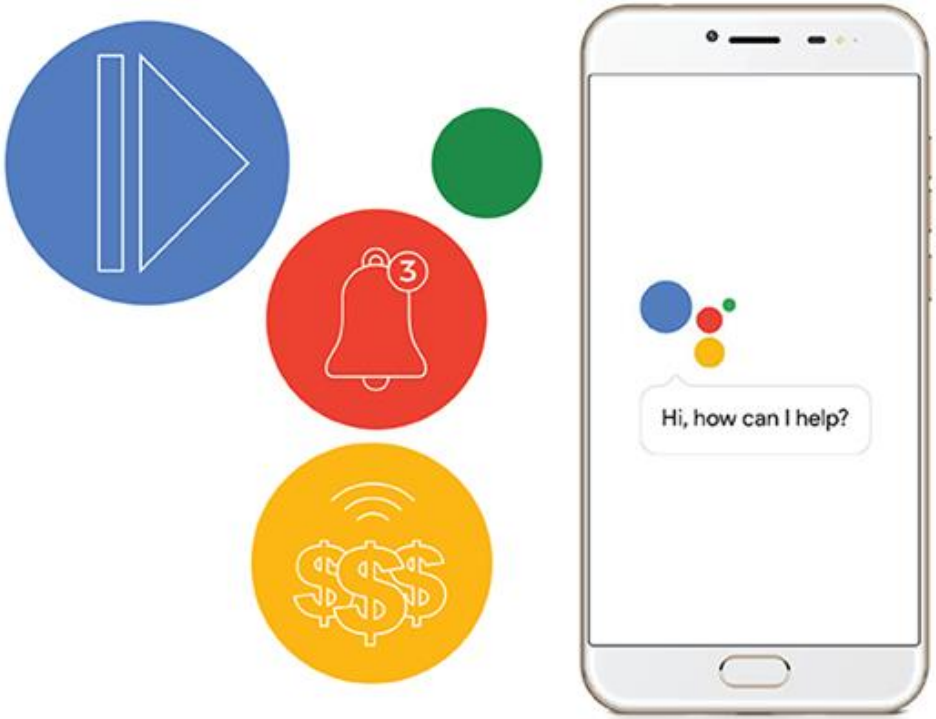
- Side view of an astronaut is walking through a puddle on mars
- The astronaut is dancing on mars
- The astronaut walks his dog on mars
- The astronaut and his dog watch fireworks

Villegas et al., “Phenaki: Variable Length Video Generation From Open Domain Textual Description.” ICLR 2023
Source: https://phenaki.research.google/

Speech Recognition

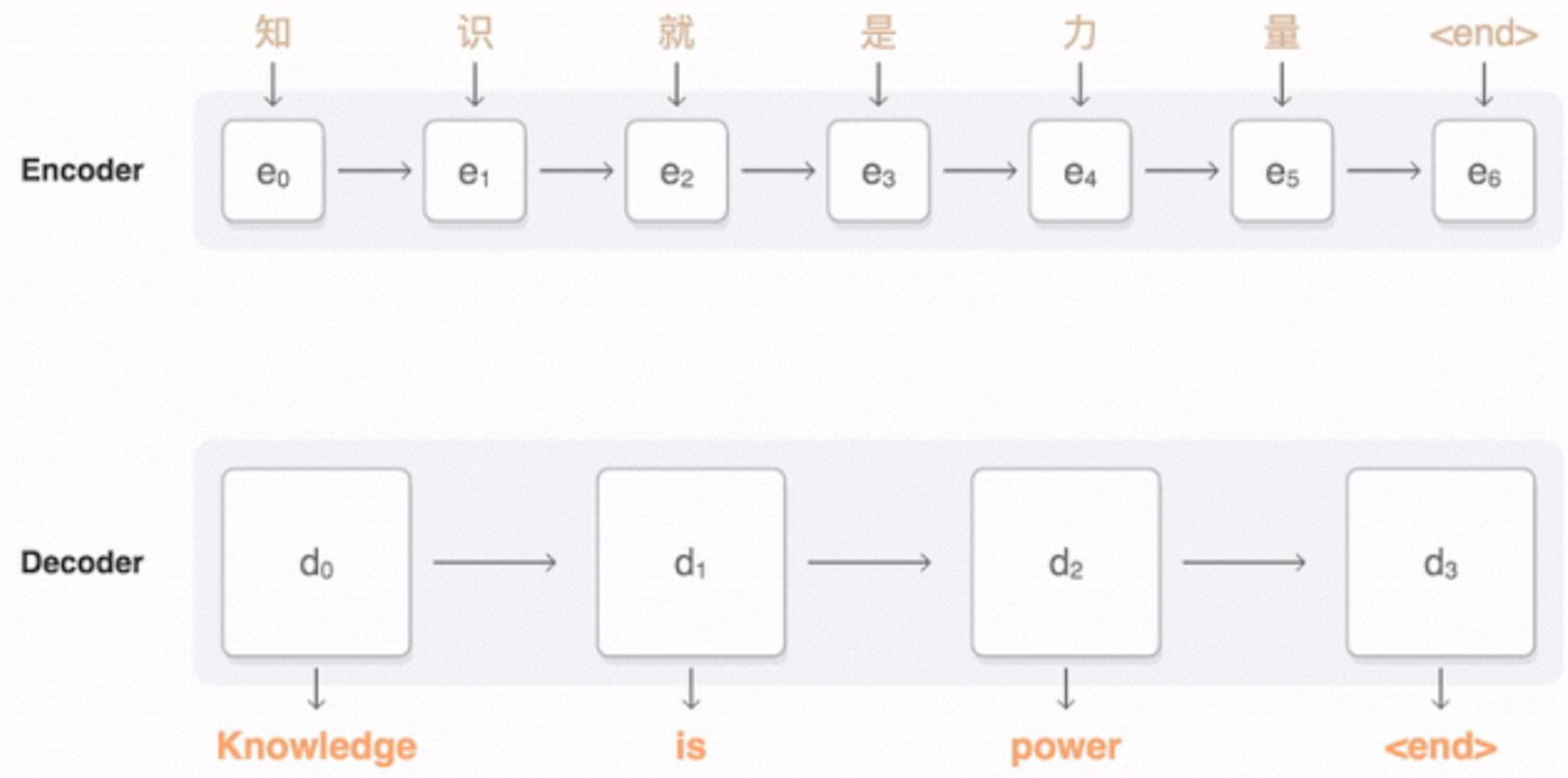
- Siri, Google Assistant, Amazon Alexa achieve commercial-level performance





Machine Translation

- Google Neural Machine Translation (in production)



Slide credit: Pieter Abbeel

ChatGPT: AI Chatbot

Model: GPT-4

KI What does ChatGPT can do?

ChatGPT, like me, is an artificial intelligence model developed by OpenAI. It's a language model, which means it's been trained on a vast amount of text data to be able to generate human-like text based on given input. Here's an overview of what ChatGPT can do:

1. **Question answering:** You can ask ChatGPT factual questions, and it can provide accurate responses based on its training data.
2. **Text completion and generation:** If you provide the beginning of a story or a sentence, ChatGPT can complete it in a coherent and contextually appropriate manner.
3. **Conversational partner:** ChatGPT can have engaging conversations on a wide range of topics. It can act as a chatbot for various purposes.
4. **Tutoring:** ChatGPT can provide explanations and tutorials on many topics, particularly academic ones such as mathematics, physics, philosophy, history, etc.
5. **Brainstorming and idea generation:** ChatGPT can generate ideas, suggestions, and help in brainstorming sessions.
6. **Language translation:** While not its primary function, ChatGPT can perform simple translations.
7. **Writing assistance:** ChatGPT can help in drafting emails, writing essays, creating content, and other writing tasks.
8. **Role-playing scenarios:** ChatGPT can be used in simulations or role-play exercises, playing a specific character or role.

Model: GPT-4

KI How long will it take to reach artificial general intelligence?

As of my last training data in September 2021, it's impossible to provide a precise timeline for the development of Artificial General Intelligence (AGI), the point at which artificial intelligence would be able to understand, learn, and apply knowledge across a broad array of tasks at a level equal to or beyond that of a human being. Predicting the arrival of AGI is a challenging task due to the numerous scientific and technical hurdles that need to be overcome.

Researchers and experts in the field of artificial intelligence have a range of opinions on when AGI might be achieved, with some predicting it could happen within a few decades, while others believe it could take a century or more, or even that it might not be achievable at all.

It's also important to note that the development of AGI would raise significant ethical, societal, and safety issues. For instance, if AGI surpasses human intelligence, there could be profound implications for society, including questions about control, purpose, and the potential displacement of humans in many areas of work and life.

As of now, a significant amount of research is being conducted on these topics, and any responsible development of AGI should take these considerations into account. It's recommended to keep an eye on the latest research and consensus among experts in the field for the most up-to-date predictions and discussions regarding AGI.

This image is copyright-free United States government work

Example credit: Andrej Karpathy

Can ChatGPT “understand” scenes?

You

Please describe the attached image.

ChatGPT

The image appears to show a group of men in a hallway that resembles a backstage or behind-the-scenes area. The man at the forefront, who is walking through a door, turns his head to the left with a smile, looking at someone or something that is catching his attention. He is sharply dressed in a suit and tie, suggesting formality or a professional setting. Another man, standing in the doorway and holding a clipboard, is also looking in the same direction with a smile. The interior has a checkered floor, white walls, and lockers on the right side, giving it a somewhat institutional look, like that of a school or sports facility. The mood seems light and cheerful, with several other individuals in the background also dressed in suits, all appearing to be part of a professional or formal event.

Can ChatGPT “understand” scenes?

You

Please describe the attached image and explain why everyone except the man on the scale is laughing.

ChatGPT

The image depicts a group of men in a hallway that resembles a changing room or a gym due to the presence of lockers. One man is standing on a scale, focused on the reading, while holding some papers and a pen, seemingly unaware of the others. All the other men, dressed in suits, are laughing or smiling while looking in his direction. The man at the forefront, who is looking back over his shoulder, seems to be the center of attention and the source of amusement. His expression and posture suggest that he might have made a humorous comment or gesture that caused the laughter. The atmosphere is friendly and jovial. The humor could arise from a private joke, a light-hearted comment about the situation, or a playful interaction that is not directly evident from the image alone.

Reinforcement Learning

Reinforcement Learning for Atari Games

Objective: Complete the game with the highest score
State: Raw pixel inputs of the game screen
Action: Game controls e.g. Left, Right, Up, Down
Reward: Score increase/decrease at each time step

Mnih et al, "Playing Atari with Deep Reinforcement Learning", NeurIPS Deep Learning Workshop, 2013

AlphaGo

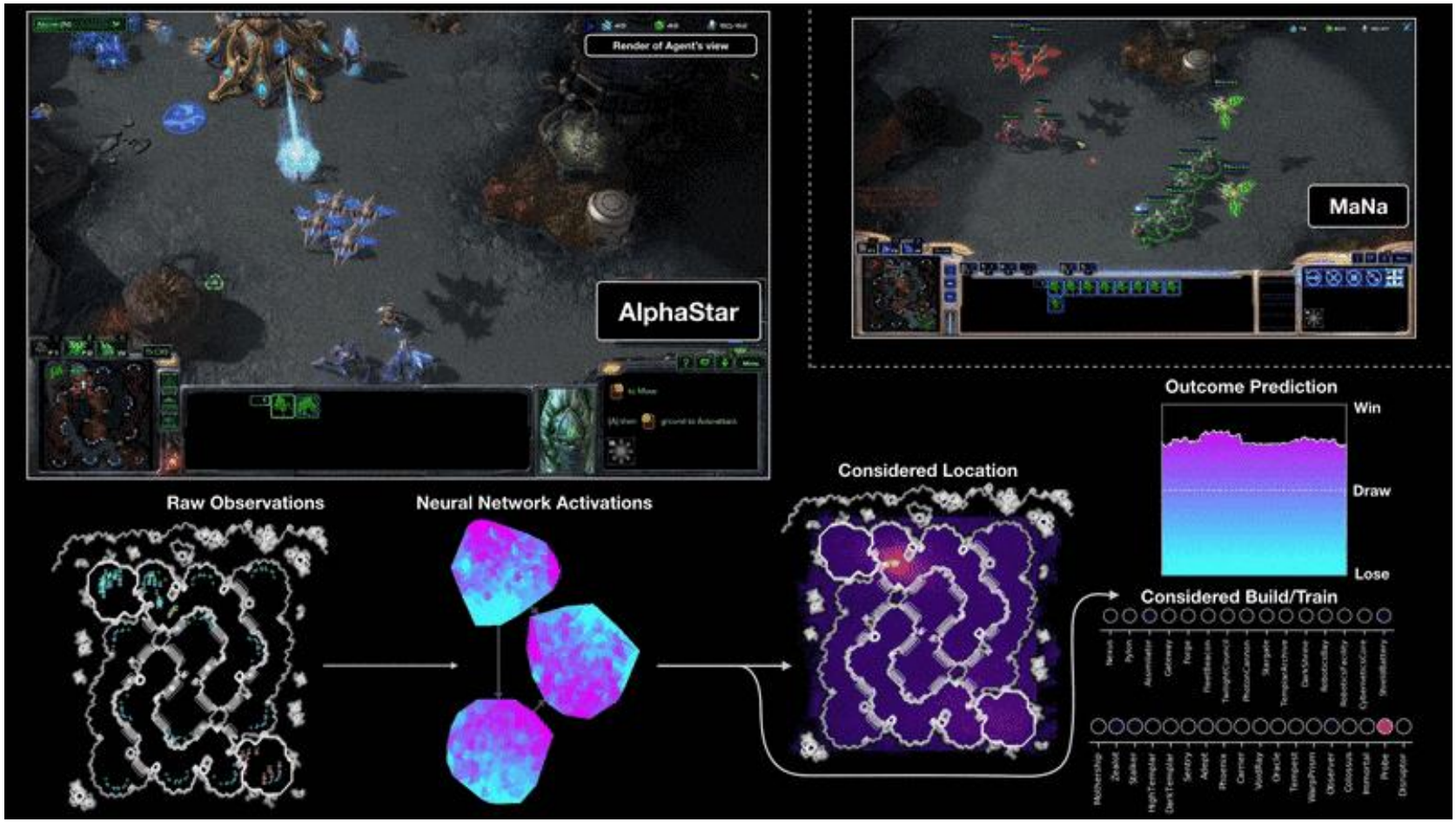
- Another breakthrough from Google DeepMind
- Combines Monte-Carlo Tree Search (MTCS) with deep neural networks

AlphaGo Silver et al, Nature 2015
AlphaGoZero Silver et al, Nature 2017
AlphaZero Silver et al, 2017
Tian et al, 2016; Maddison et al, 2014; Clark et al, 2015

Slide Credit: Pieter Abbeel

AlphaStar

- Achieved grandmaster level in StarCraft II



Vinyals et al, Nature 2019

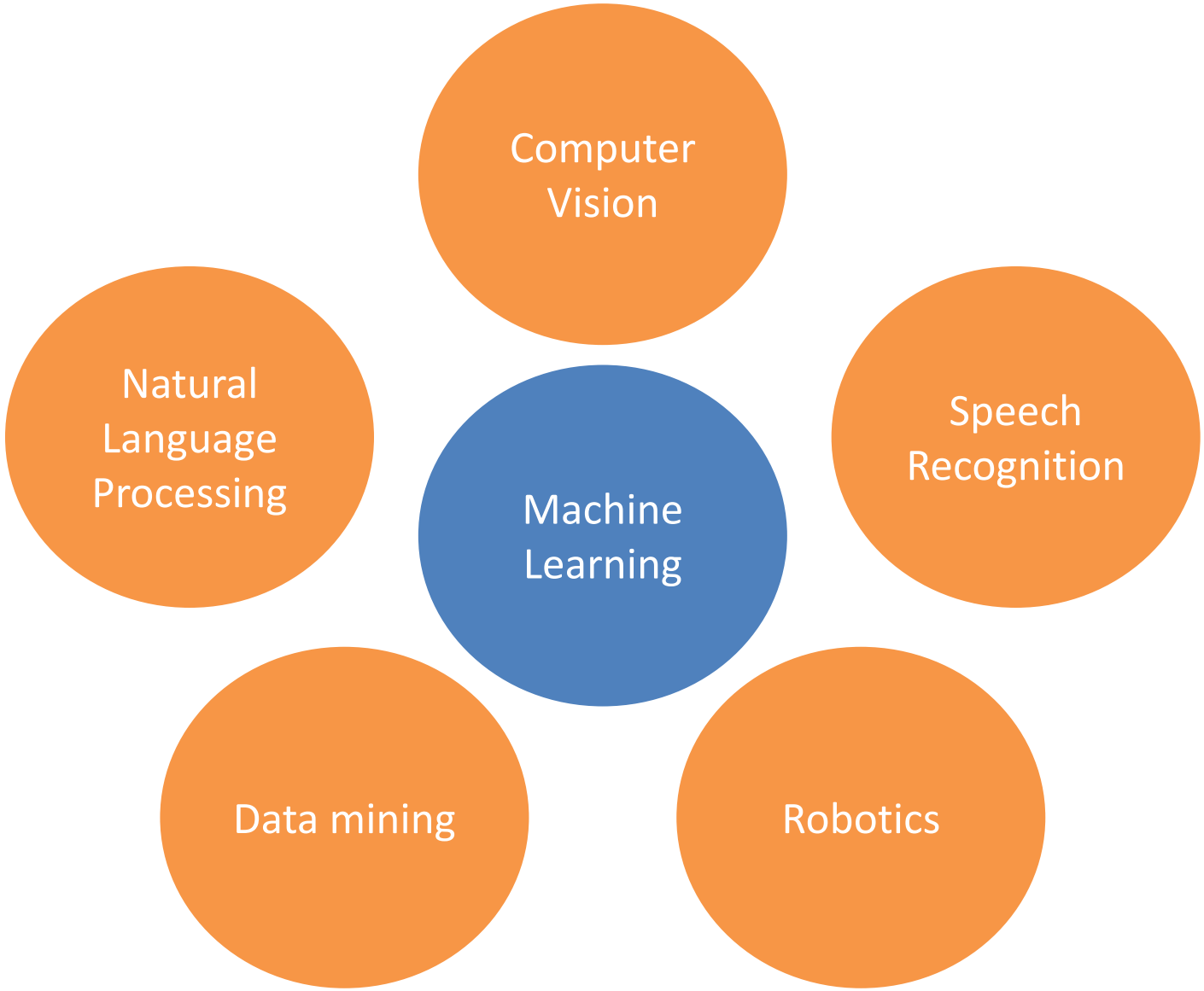
Robot Learning

Levine et al., Learning Hand-Eye Coordination for Robotic Grasping. 2016
video: https://www.youtube.com/watch?v=cXaic_k80uM slide credit: Pieter Abbeel

Self-driving Cars

https://youtu.be/O6DRfAC1JXA

Machine Learning and Other Fields



Next: Python Review