Lecture 1: Introduction

Who we are

- Instructors
    - Prof. Chuang Gan, office hours: Friday 9:00-10:00 am (Eastern Time)
- TAs:

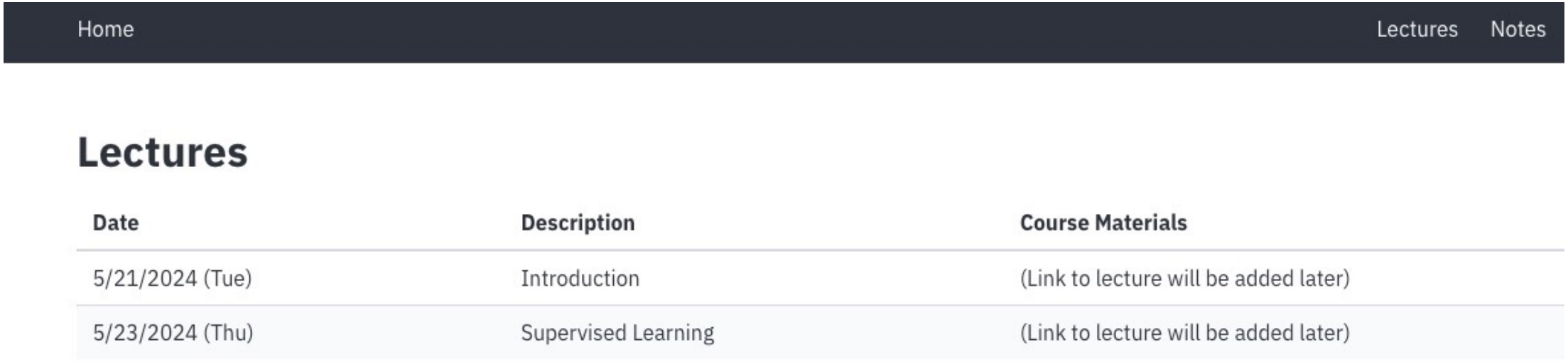
Bao Dang
TA Hours: Tue/Thu 2-3pm

Mohammadreza Teymoorianfard
TA Hours: Mon/Wed 12-1pm

Minh Vu

Course web page

https://compsci589-summer24.github.io/



589 Machine Learning

- Topics:
  - Supervised learning with K-Nearest neighbors
  - Logistic regression for classification
  - Feed forward neural nets
    - Backpropagation
    - Batch normalization
    - Drop-out
  - Convolutional neural nets
  - RNN and Transformer
  - Generative Models
  - Neuro-symbolic AI
  - Multi-modal AI

589 Machine Learning

- Balance of theory vs. practice
  - Heavily tilted toward practice.
  - Examples:
    - Regularization will be used, but not much theory of it.
    - No proofs of convergence or optimality
  - Instead:
    - Develop applications “from scratch”
    - Build “layered” architectures from scratch so new models can be easily assembled
    - Implement popular add-ons such as batch normalization
    - Learn techniques for training and setting hyperparameters.

589 Machine Learning

- Applications
  - Mostly **Computer Vision**: Object recognition in particular.
  - However, can easily be applied to other domains.
    - You will learn what you need to know to apply neural nets broadly.
  - Hopefully more about **Natural Language Processing** this semester.

589 Machine Learning

- What this course is *not*:
  - General course on machine learning
  - General course on graphical models
  - Not even a general class on deep learning!!!
    - No Bayes Nets
    - No restricted Boltzmann machines or deep Boltzmann machines
  - Not a computer vision survey class
    - No tracking, stereo, depth estimation, etc., etc.

Course grades

- 3 Long Programming Assignments
  - Get started as soon as assignments are posted.
  - Some aspects of assignments require only basic knowledge of Python, but some require in-depth understanding of numpy arrays and complicated indexing schemes. They can take a while to work through.
  - If you don’t know Python, work through tutorial now.

Grading Policy (approximate)

- 3 Problem sets: 15%*3 = 45%
- Midterm exam: 15%
- Final Course project: 40%
  - Proposal: 5% (out of 40%)
  - Milestone: 5% (out of 40%)
  - Final write-up: 20% (out of 40%)
  - Review of others: 10% (out of 40%)
- Late Policy:
  - 7 free late days in total: use them as you see fit
  - Afterwards: 25% off per day late
  - Not accepted after 3 late days
  - Does not apply to final course project (must be on time)

Assignments

- 3 Long Assignments
  - Get started as soon as assignments are posted.
  - Some aspects of assignments require only basic knowledge of Python, but some require in-depth understanding of numpy arrays and complicated indexing schemes. They can take a while to work through.
  - If you don’t know Python, work through tutorial now.

Getting Started

- Example: Mac
  - Language: Python
    i. Instructions for installing given under first assignment instructions.
    ii. Development environment: Jupyter Notebook. Live code environment.
      - Poll
  - Running a shell on the side: Jupyter QtConsole
    i. Good for testing syntax, return values of functions.

Assignment #1

- Soon posted on course website
- Due in 2 weeks(in GradeScope).
- It includes:
- Write/train/evaluate a kNN classifier
- Write/train/evaluate a Linear Classifier (SVM and Softmax)
- Write/train/evaluate a 2-layer Neural Network (backpropagation!)
- Requires writing numpy/Python code

Compute: Use your own laptops. Talk to me or TA if you don't have your own computer.

Plagiarism and Cheating

Who, me?

- Right now, cheating seems very far away.
- Now imagine:
  - You just started homework due in 2 days.
    You realize it will take you a week.
  - You just had an internship interview where they asked you if you are getting an A in machine learning.
  - You have a midterm tomorrow and a project due in another class in one week.
  - You were just surfing the web for information on Python slicing and you bumped into a full solution to the current problem set.
    Perhaps I should just take a quick peek...

Don’t do it!!!!!!!!!!

Cheating in the past

- 18 people were caught cheating during a recent semester.
- They were given penalties including
  - 0 for the given assignment
  - An additional grade reduction for the class.
  - A filing with the Academic Dishonesty board.
- Many people failed the class as a result.

UMass Culture

- If you cheat, you put me and a lot of other people in an awful position:
  - If I let you off the hook, I am being completely unfair to people who actually did the work, and I'm promoting the idea that it's ok.
  - If I punish you, I feel like a jerk, and you think I'm a jerk.
- The bottom line is, there is no good way to come out feeling good about a cheating incident. It creates massive stress between faculty and students. Please don't do it!

Advice

- Everyone knows you're not supposed to cheat.
- What people don't know is what you're supposed to do when you're desperate. Here's some advice:

1) If you're overloaded in the middle of the semester, consider dropping a class. Hopefully you can drop it without a "W", but even a "W" is a lot better than an "F" and a record of cheating. A "W" will not influence your grade point average.
   (I dropped the same class 4 times in grad school!)
2) Take a "0" on part of the problem set. Many people who did not do part of one problem set got an A-. Some people missed a whole problem set and still got a B for the course.

5 Rules: What is cheating?

1. Let's start with an easy one. Don't copy any piece of the solution of any problem.
2. Never **look at** solutions to any of the homework problems. Most people who were caught cheating last semester claimed that they only "looked at" on-line solutions. This is NOT ALLOWED.
3. Do not look at discussions of the homework problems. These are likely to include methods for solving parts of the problem, which is cheating.
4. Don't look up pieces of the problem on Google. For example:
   a. "Computing the derivative of softmax"
   b. "Gradient updates for the multi-class SVM loss".
      Once you've done the search, you cheated. You are likely to see something you cannot forget. You can't "unsee" the answer once you've seen it.
5. Common sense. If you look at something on the web and it made the problem easier, then you're probably cheating. To be safe, stick to class materials, TAs, and Professor.

Questions about what is allowed

1. Question: Can I work with other students on the homeworks?
   Answer: No. Do the homeworks yourself.
2. Question: Where can I get help?
   a. Look at the course notes
   b. Go to optional Friday sections
   c. Talk to the TAs
   d. Talk to the professor
3. Can I look at on-line materials that are not part of the course?
   a. Basically no. If you look at something and it's part of the solution, then you have cheated. So it's dangerous to go surfing around. Stick to the materials on the course web site. If there is something you want to look up, ask the TAs a question and we'll try to put materials on the course web site if it's appropirate.

Plagiarism

- When you write your final report, there are two ways you can use material from other papers:
  - Use the general ideas from another paper with your own writing. You *cannot* copy text from another paper unless you use quotation marks. Example:
    - In his famous 1915 paper, Einstein introduced the theory of general relativity [Einstein, 1915].
  - Quote a specific passage, usually because of the exact way it is worded:
    - Einstein said, “God does not place dice with the universe.” [Einstein, 1958]

Plagiarism

- You cannot copy sentences into your writing and justify by citing the paper. This is plagiarism, whether you cite it or not.

  If I Google a sentence in your paper that is not quoted, and I find it, that means you were plagiarising!

Final comment: If you don't know whether something constitutes plagiarism or cheating, ASK! If you don't ask, it will be too late.

OK…. now on to the fun stuff!

An Extremely High-Level Introduction to ML

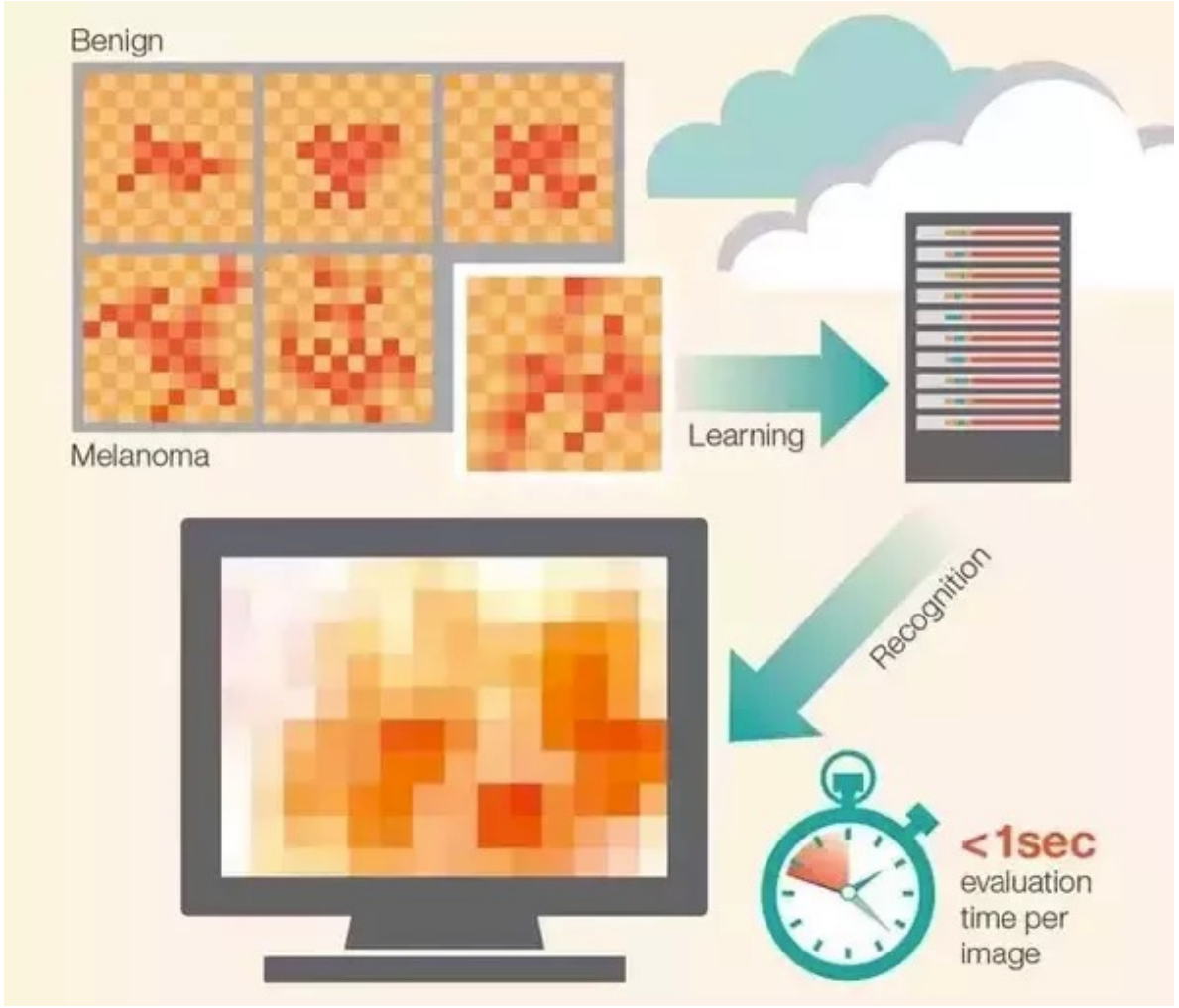

Google
amazon
ebay
NETFLIX
Bank of America

A Few Current Applications of ML

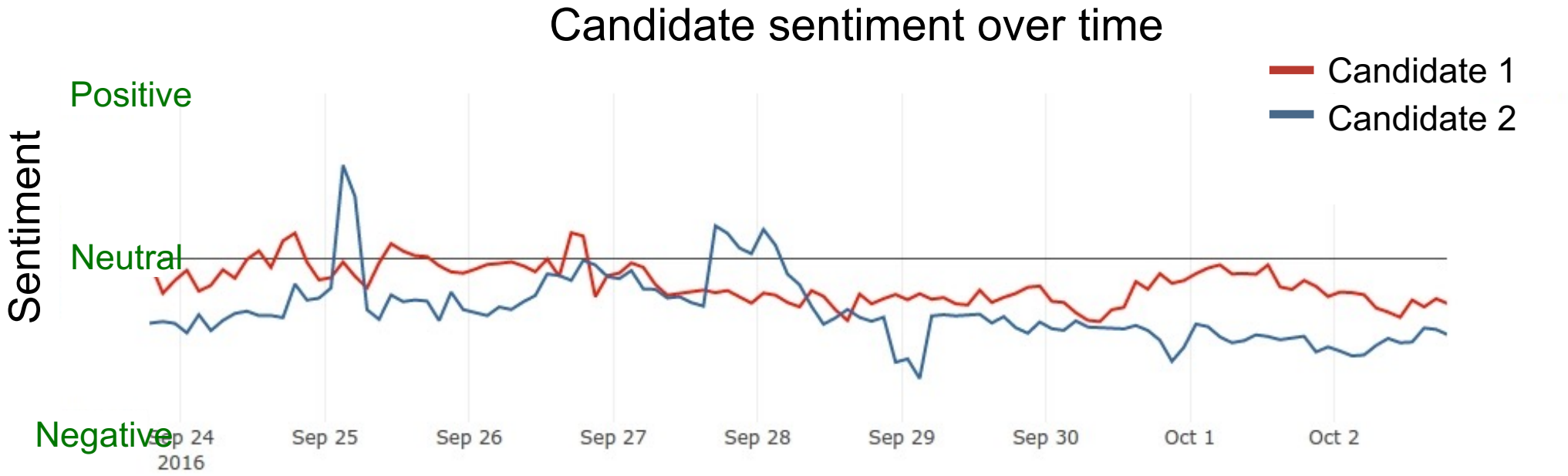
(Siri, Google Assistant, Alexa, etc)

Natural Language Processing & Understanding

“IBM detects skin cancer more quickly with visual machine learning”



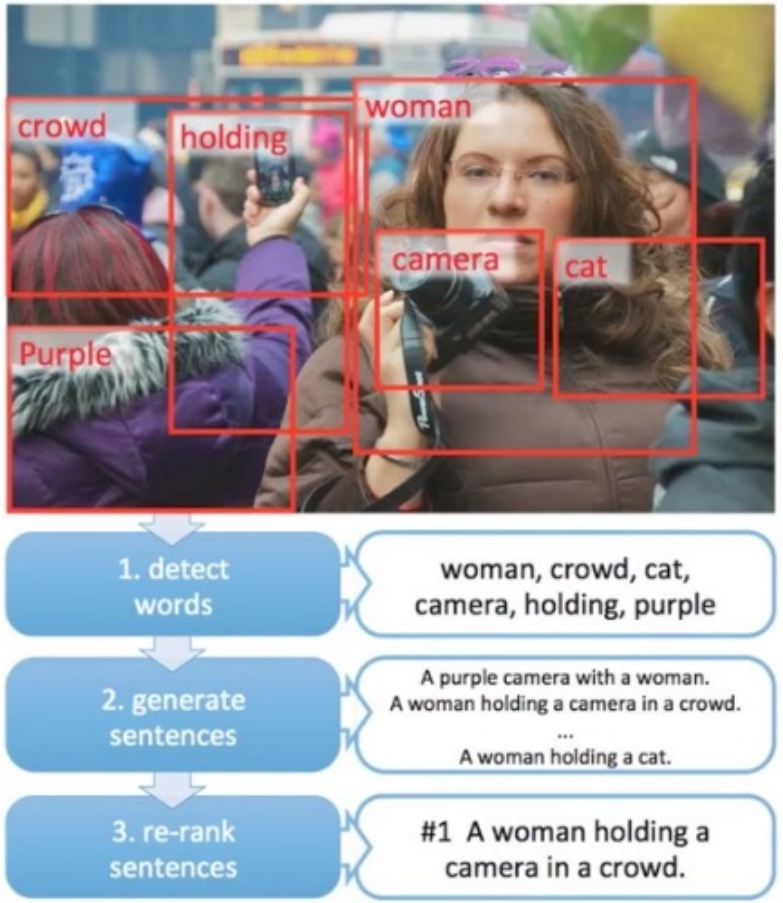
Medical Diagnosis



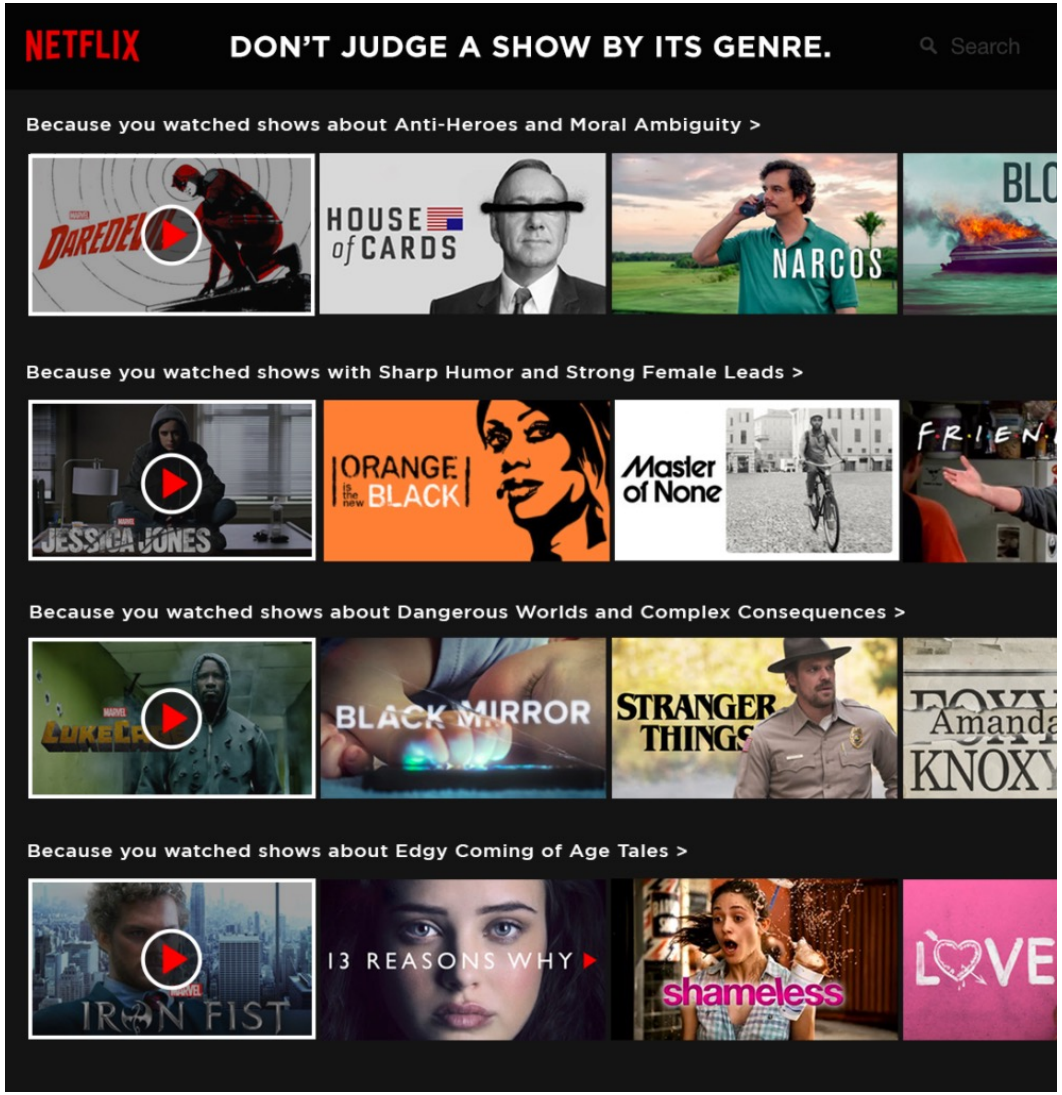
Sentiment Analysis

a man sitting on a couch with a dog

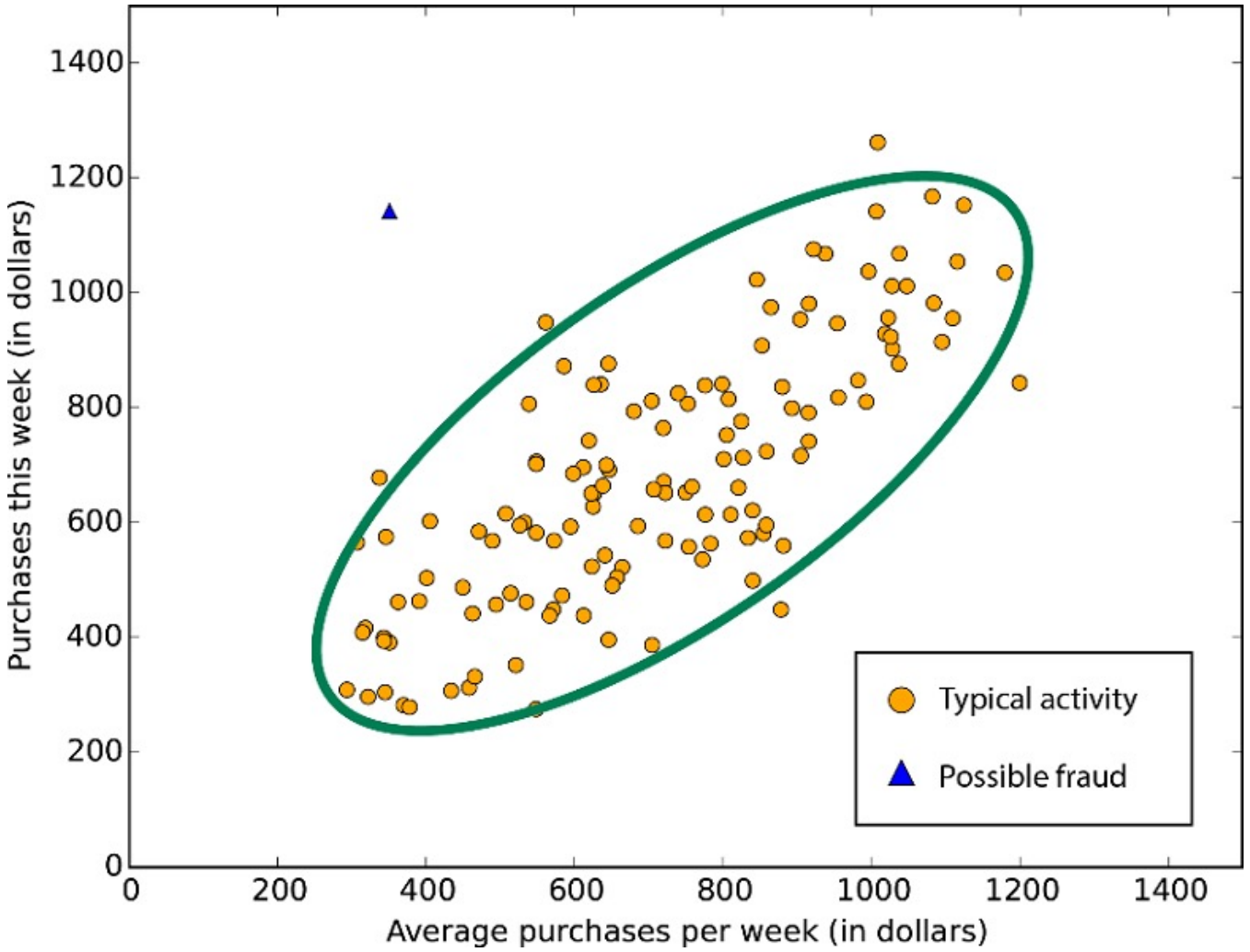
a man sitting on a chair with a dog in his lap



Scene Understanding & Description



Learning User Preferences & Interests



Fraud Detection

IDEAFEED

Google's Self-Driving Cars Are Ridiculously Safe

Autonomous Cars

How I made $500k with machine learning and HFT (high frequency trading)

Nov 6, 2012

This post will detail what I did to make approx. 500k from high frequency trading from 2009 to 2010. Since I was trading completely independently and am no longer running my program I'm happy to tell all. My trading was mostly in Russel 2000 and DAX futures contracts.

The key to my success, I believe, was not in a sophisticated financial equation but rather in the overall algorithm design which tied together many simple components and used machine learning to optimize for maximum profitability. You won't need to know any sophisticated terminology here because when I setup my program it was all based on intuition. (Andrew Ng's amazing machine learning course was not yet available - btw if you click that link you'll be taken to my current project: CourseTalk, a review site for MOOCs)

First, I just want to demonstrate that my success was not simply the result of luck. My program made 1000-4000 trades per day (half long, half short) and never got into positions of more than a few contracts at a time. This meant the random luck from any one particular trade averaged out pretty fast. The result was I never lost more than $2000 in one day and never had a losing month:

Monthly P&L

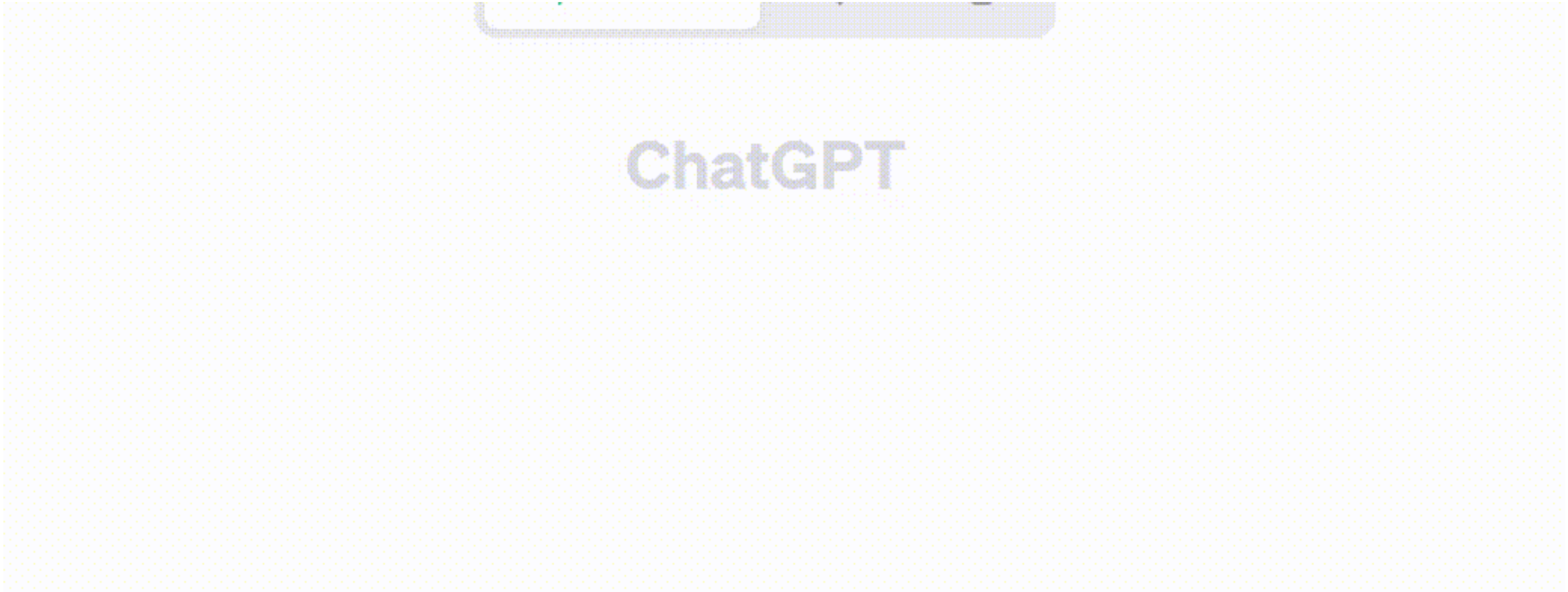
Hi, I'm Jesse, founder of Thinklab. I live and play in San Francisco. You've found my home on the web.. Welcome!

Tweets Follow

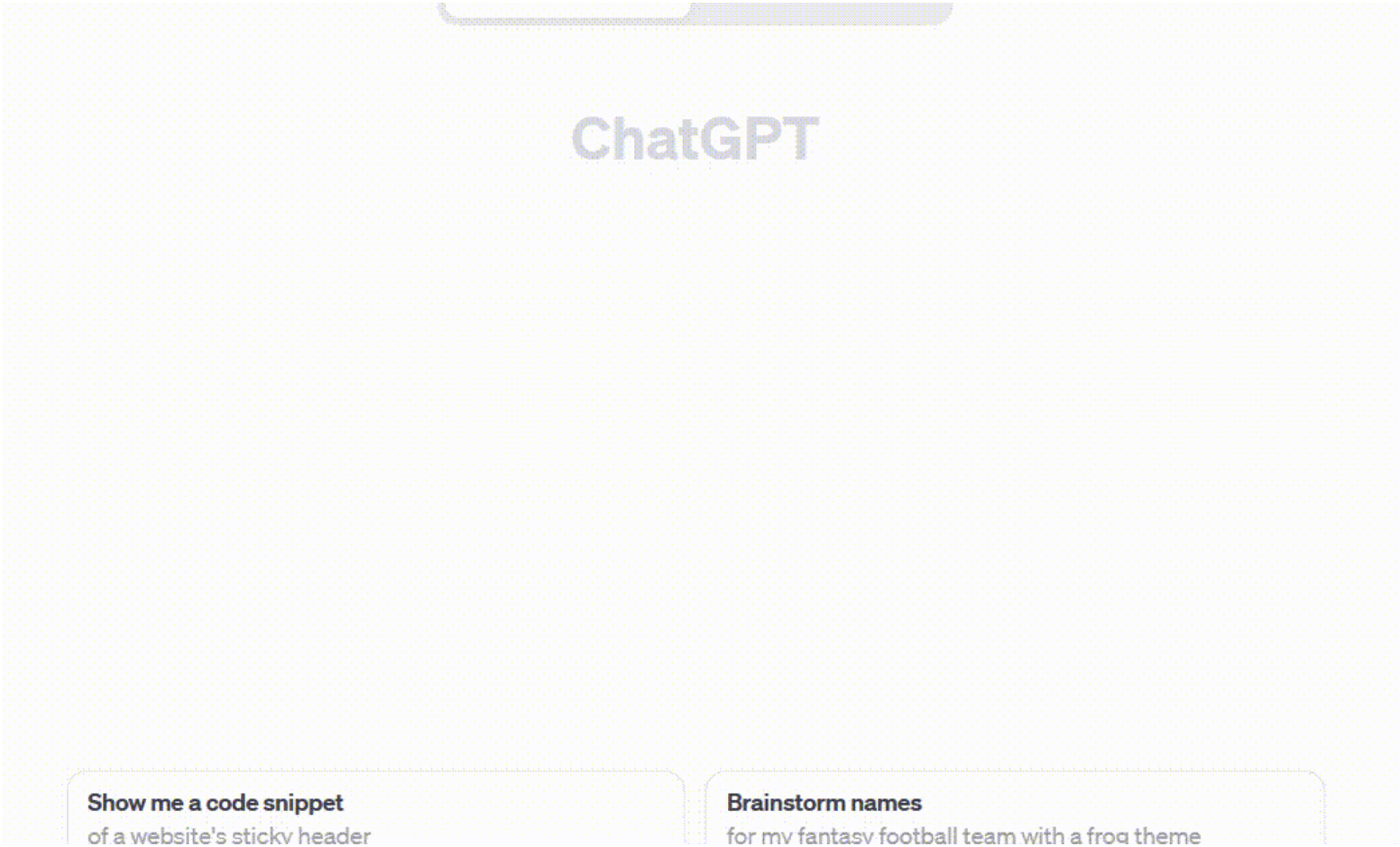
Financial Services & Applications

What is Deep Learning?

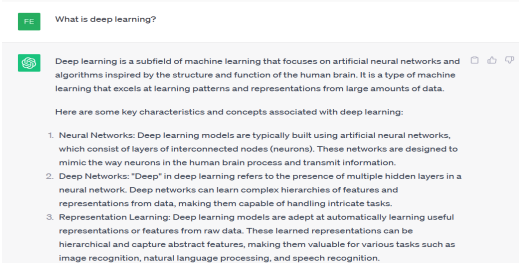
Recently, OpenAI developed **a new deep learning conversation system.**



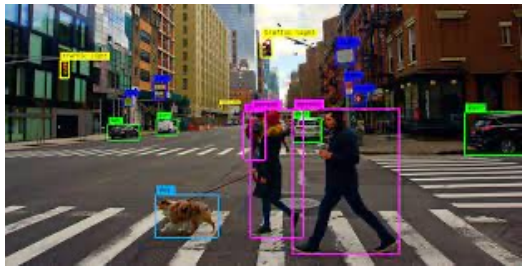
Why Deep Learning?



- Natural Language Processing



- Computer Vision



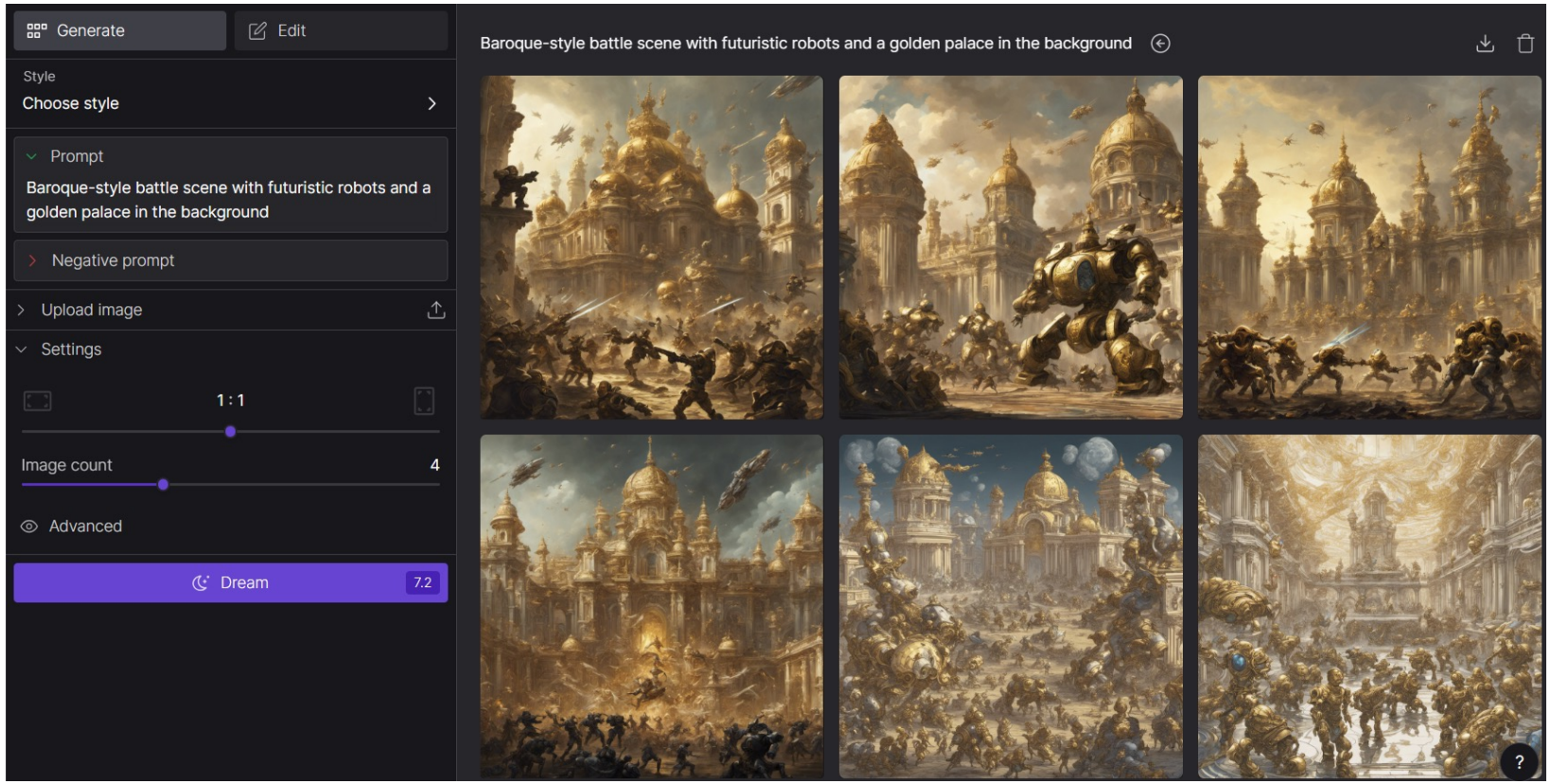
- Reinforcement Learning

Why Deep Learning?

Stable Diffusion

stability.ai

- Generate images / Edit existing images based on the text prompt



"Teddy bears working on new AI research on the moon in the 1980s"

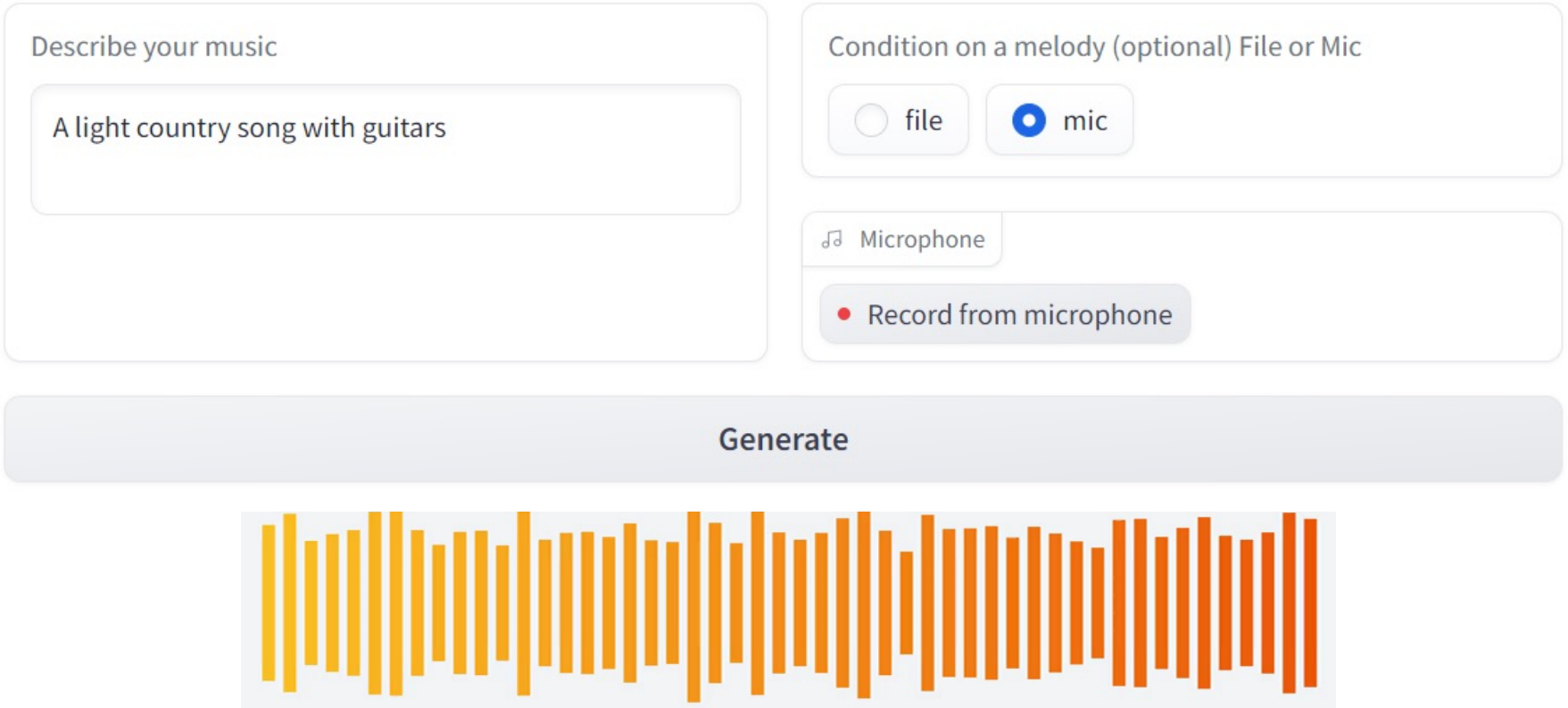
DALL-E — Realistic Image Creation from Natural Language

Why Deep Learning?

MusicGen

Meta

- Generate music based on the text prompt

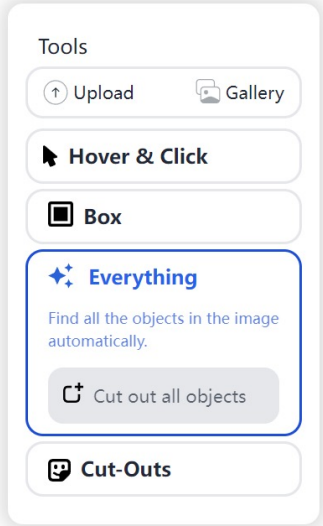


Why Deep Learning?

Segment Anything

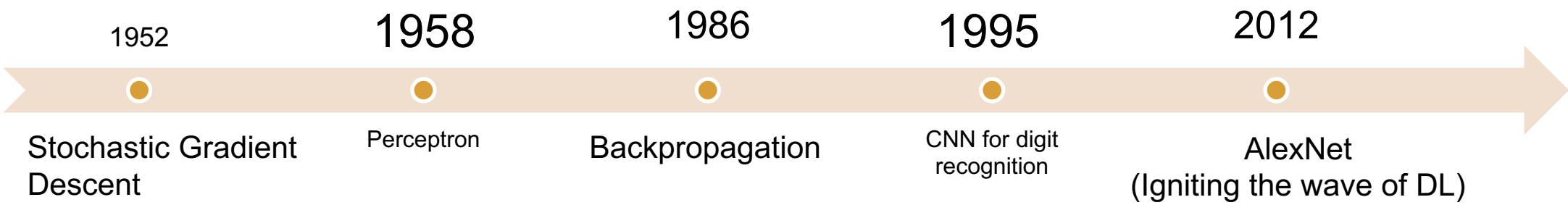
Meta

- Segment image based on segmentation prompt



Why Now?

Neural networks have a history of over 70 years, but deep learning surged in the last



Big data

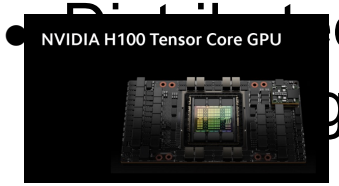
- Large Datasets
- Advances in data collection &

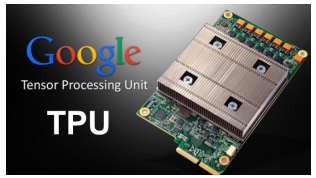
YouTube

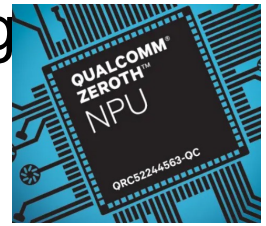
hadoop Google Cloud

Hardware

- GPU acceleration
- AI-specific chips
- Distributed







Software

- Open-Source Frameworks

TensorFlow PyTorch

GitHub Hugging Face

Machine Learning

Three Machine Learning approaches

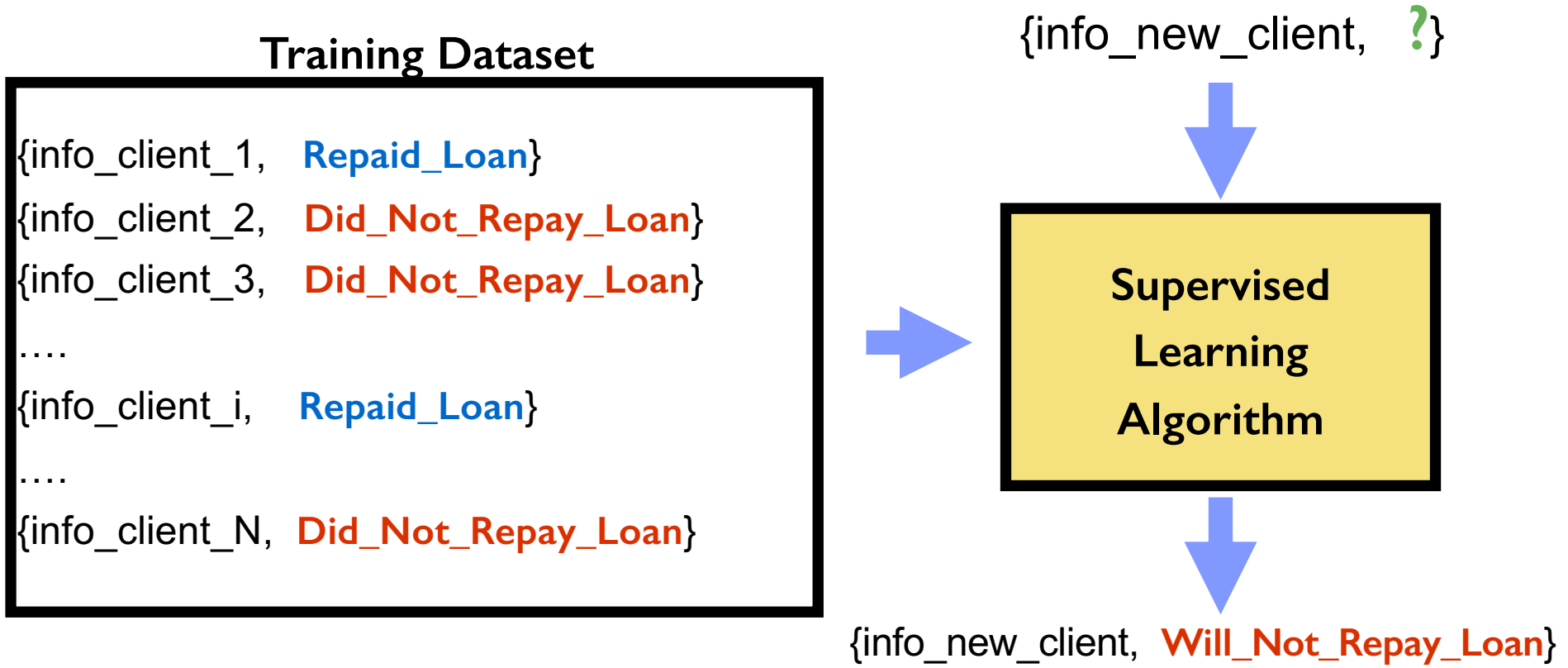
Supervised Learning

Unsupervised Learning

Reinforcement Learning

Supervised Learning

A *dataset* containing *labeled training examples* is given to the AI

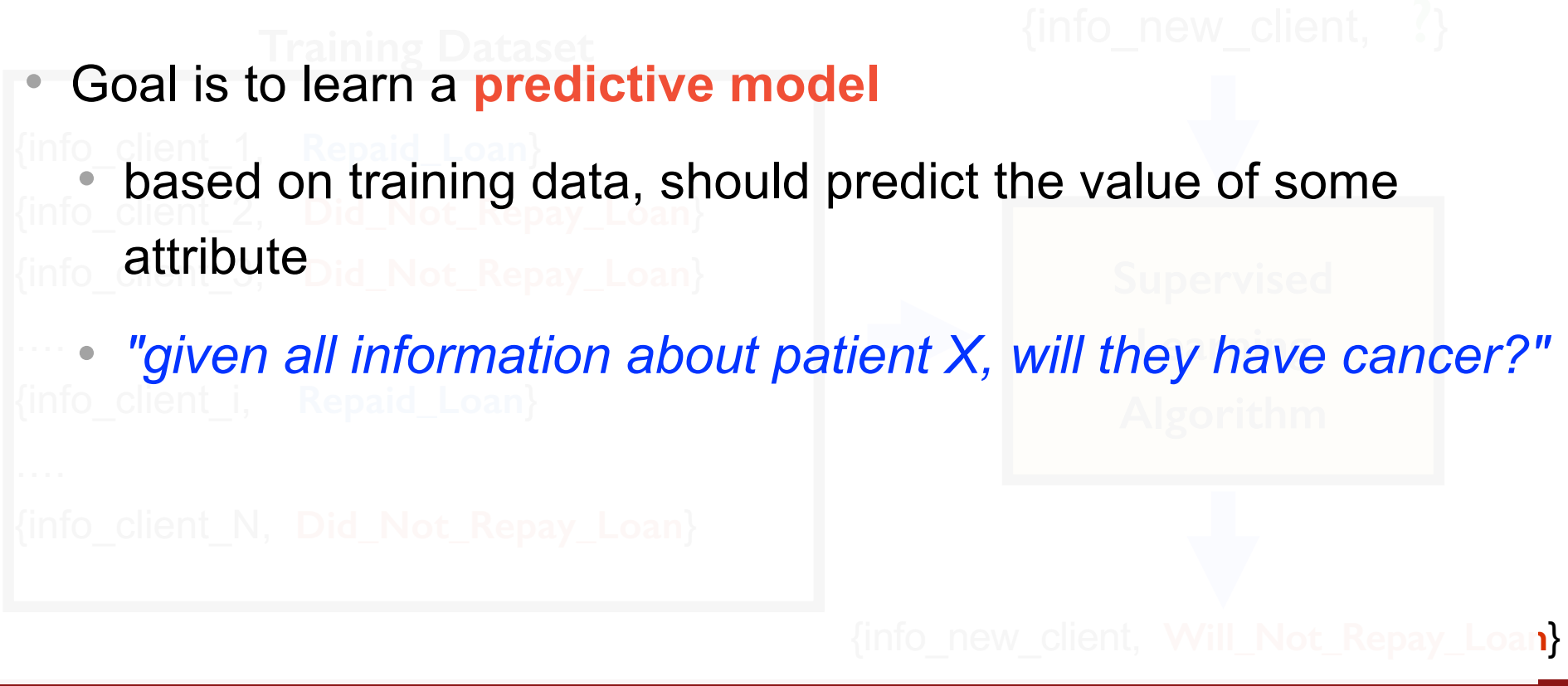


Supervised Learning

A *dataset* containing *labeled training examples*

is given to the AI

- Goal is to learn a **predictive model**
  - based on training data, should predict the value of some attribute
  - *"given all information about patient X, will they have cancer?"*



Machine Learning

Three Machine Learning approaches

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Unsupervised Learning

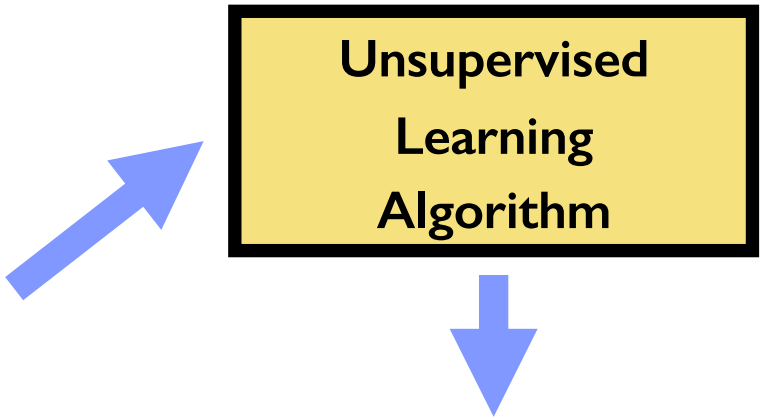
A *dataset* containing *unlabeled data* is given to the AI

(no "correct/expected" answer associated with each example)

Bruno Castro da Silva

Training Dataset

{fuelEconomy=1, lowPrice=3, comfort=5, acceleration=1}
{fuelEconomy=3, lowPrice=5, comfort=3, acceleration=3}
{fuelEconomy=5, lowPrice=5, comfort=2, acceleration=4}
{fuelEconomy=2, lowPrice=2, comfort=1, acceleration=4}
{fuelEconomy=1, lowPrice=3, comfort=2, acceleration=2}

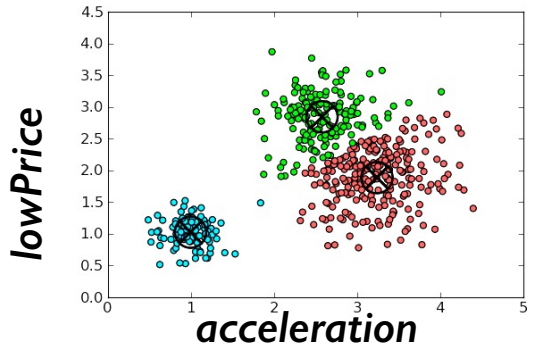


(features buyers said were important when making a decision

"luxury car buyers"

"car buyers interested in performance/speed"

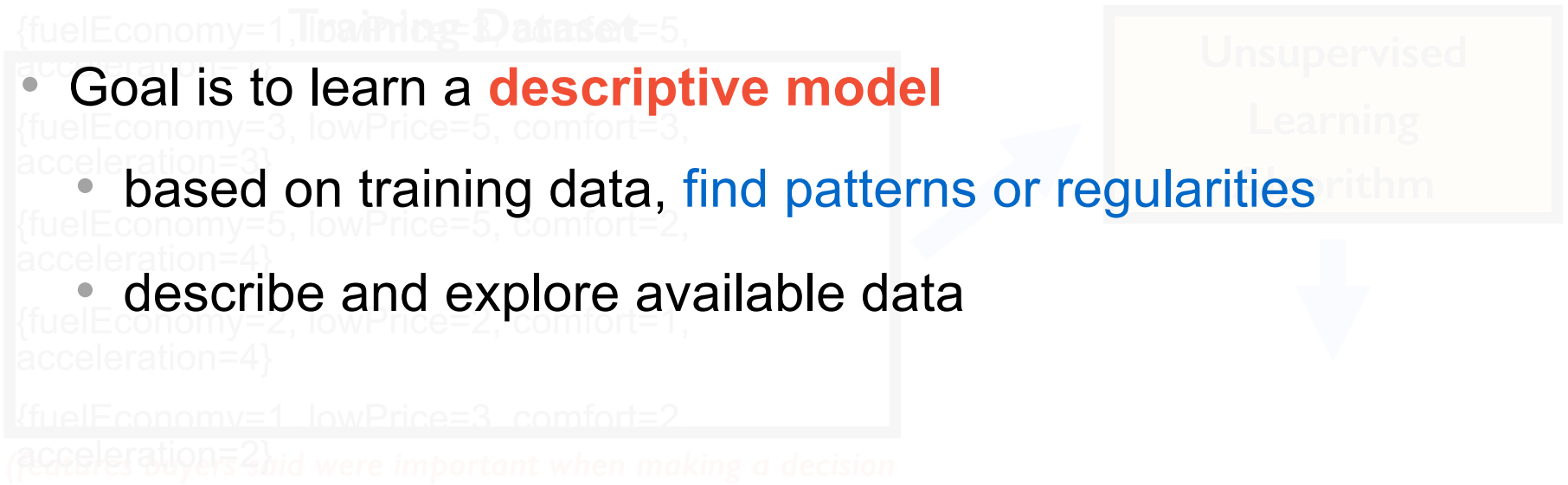
"car buyers who prioritize economy and comfort"



Unsupervised Learning

A *dataset* containing *unlabeled data* is given to the AI

(no "correct/expected" answer associated with each example)



- Goal is to learn a **descriptive model**
  - based on training data, find patterns or regularities
  - describe and explore available data

Machine Learning

Supervised Learning → there **is** a dataset with examples of which decisions you should make, depending on the situation

Unsupervised Learning → there is **no** dataset with examples. Just lots of data, and we need to find interesting patterns

Reinforcement Learning → there is someone evaluating your actions, but not telling you *which* actions would have been best/correct

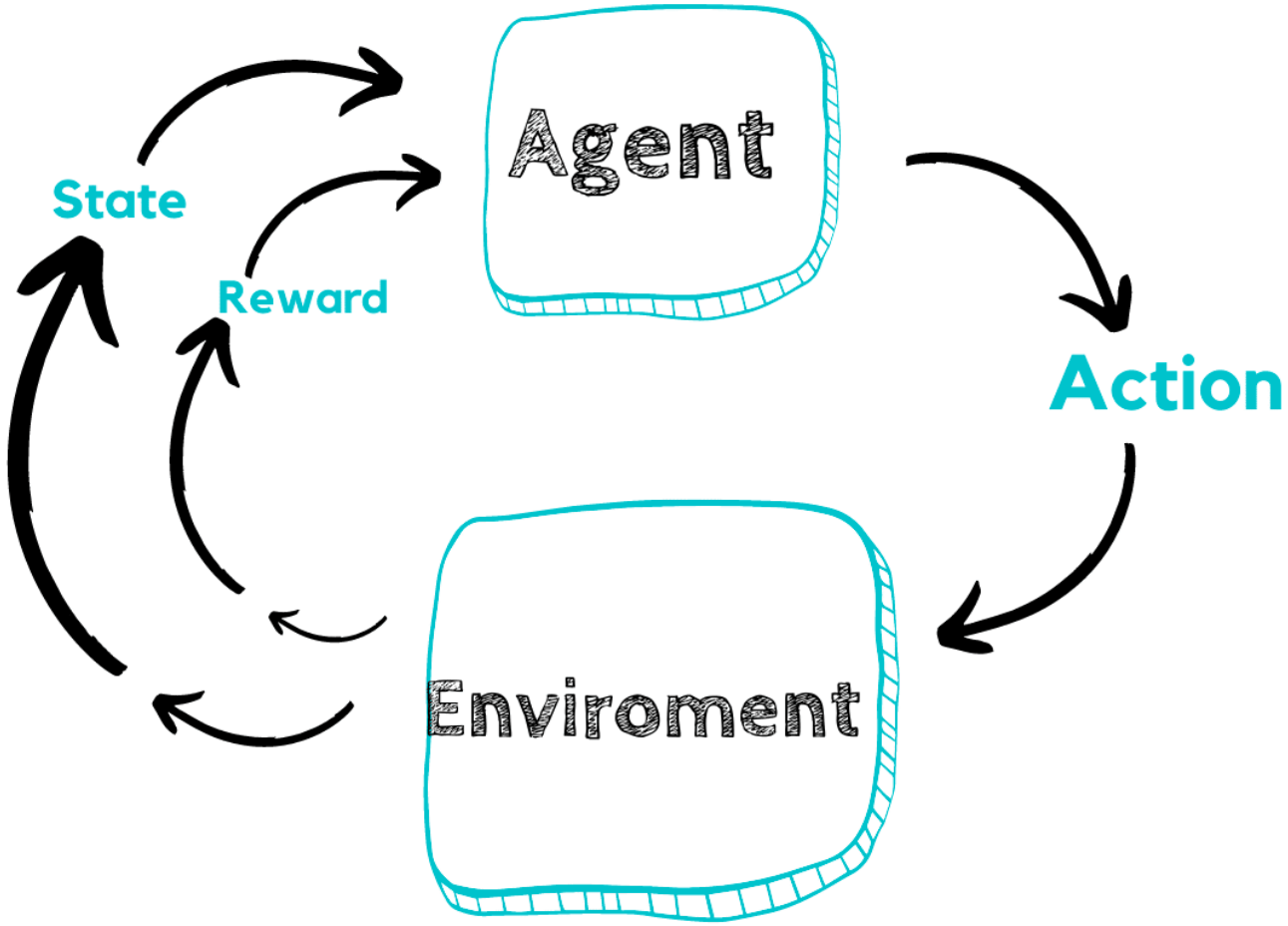
Reinforcement Learning (and robotics)

Reinforcement Learning

- Algorithms to learn how to behave / solve a task / select actions
  - no dataset containing examples of how to behave in different situations
  - no prior knowledge about the environment/system
  - learns only based on rewards (or punishments)

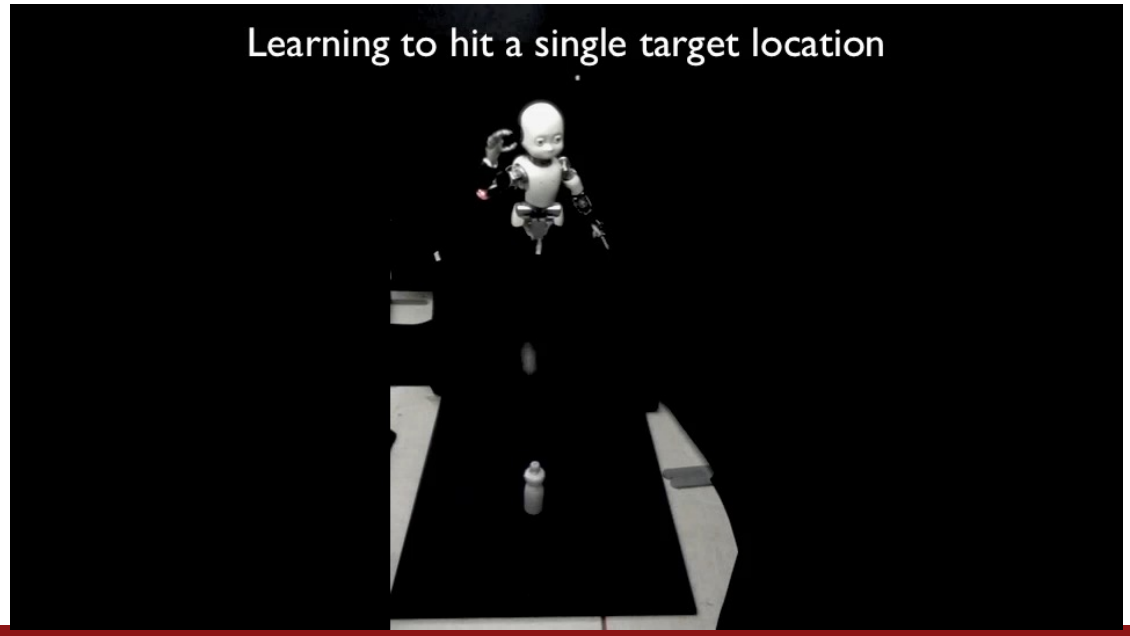
- Positive reinforcement
  more likely that behavior will occur again in the future
- Negative reinforcement
  less likely that behavior will occur again in the future

Reinforcement Learning



Reinforcement Learning

- Algorithms to learn how to behave / solve a task / select actions
  - no dataset containing examples of how to behave in different situations
  - no prior knowledge about the environment/system
  - learns only based on rewards (or punishments)



States?
Actions?
Rewards?

Reinforcement Learning

Digital Marketing

State: user information

Actions: ad that can be shown

Goal: maximize number of clicks (reward +1 when user clicks)

Supervised Learning

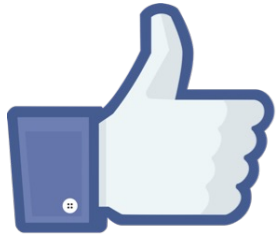
Supervised Learning

- Predict number of Likes a Facebook post/meme might get
- Based on
  - how many friends were online
  - how funny the user believes the meme is (0-10)

	#friends	funny score	Likes
Post1	40	8	28
Post2	36	10	28
Post3	20	6	16
Post4	56	4	31
Post5	58	0	29
Post6	46	10	33

Supervised Learning

	#friends	funny score	Likes	
Post1	40	8	28	28
Post2	36	10	28	28
Post3	20	6	16	16
Post4	56	4	31	32
Post5	58	0	29	29
Post6	46	10	33	33



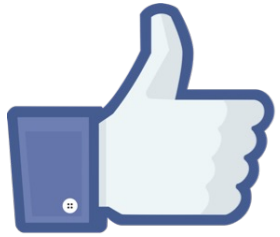
It seems like
there is a *pattern* here!

Likes = (0.5 * #friends) + (1 * funny_score)

Machine learning algorithms learn these "weights"

Supervised Learning

	#friends	funny score	Likes	
Post1	40	8	28	28
Post2	36	10	28	28
Post3	20	6	16	16
Post4	56	4	31	32
Post5	58	0	29	29
Post6	46	10	33	33



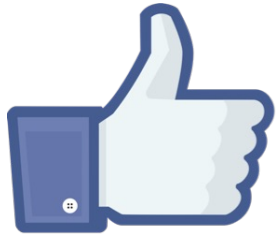
It seems like
there is a *pattern* here!

Likes = (0.5 * #friends) + (1 * funny_score)

Linear Regression

Supervised Learning

	#friends	funny score	Likes	
Post1	40	8	28	28
Post2	36	10	28	28
Post3	20	6	16	16
Post4	56	4	31	32
Post5	58	0	29	29
Post6	46	10	33	33



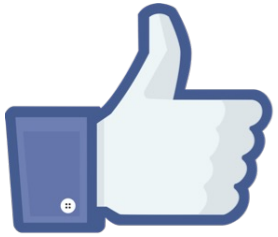
It seems like
there is a *pattern* here!

Likes = complicatedFunction(#friends, funny_score)

Neural Networks

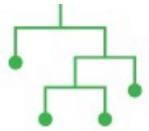
Supervised Learning

	#friends	funny score	Likes
Post1	40	8	28
Post2	36	10	28
Post3	20	6	16
Post4	56	4	31
Post5	58	0	29
Post6	46	10	33



```
if #friends >= 50
    if funny_score > 3
        Likes >= 30
    else
        Likes < 30
else
    if funny_score > 9
        Likes >= 30
    else
        Likes < 30
```

Decision Trees



Learning to Play the 20 Questions Game

http://en.akinator.com/


Question N°1
Is your character a female?
Yes
No
Don't know
Probably
Probably not


Correct
Question N°2
Is your character real?
Yes
No
Don't know
Probably
Probably not


Correct
Question N°3
Is your character a famous youtuber?
Yes
No
Don't know
Probably
Probably not


Correct
Question N°4
Is your character older than 35 years old?
Yes
No
Don't know
Probably
Probably not


Correct
Question N°5
Is your character American ?
Yes
No
Don't know
Probably
Probably not


Correct
Question N°6
Is your character an actor?
Yes
No
Don't know
Probably
Probably not


Correct
Question N° 7
Is your character the president or was he?
Yes
No
Don't know
Probably
Probably not


Correct
Question N°8
Is your character black?
Yes
No
Don't know
Probably
Probably not


I think of
Barack Obama
Former President of the United States
© Copyright / IP Policy
Yes
No

Learning to Play the 20 Questions Game

http://en.akinator.com/

Training Dataset

{man=True, American=True, Politician=False, Singer=True, Dead=True} → Elvis
{man=False, American=False, Politician=True, Singer=False, Dead=False} → Angela Merkel
{man=True, American=True, Politician=True, Singer=False, Dead=False} → Obama
{man=True, American=False, Politician=False, Singer=True, Dead=False} → Bob Marley
...

Decision Trees

- Which attributes of a person to check first, to guess as fast as possible?
  - Is the person a man or a woman?
  - Is the person older than 5 years old?

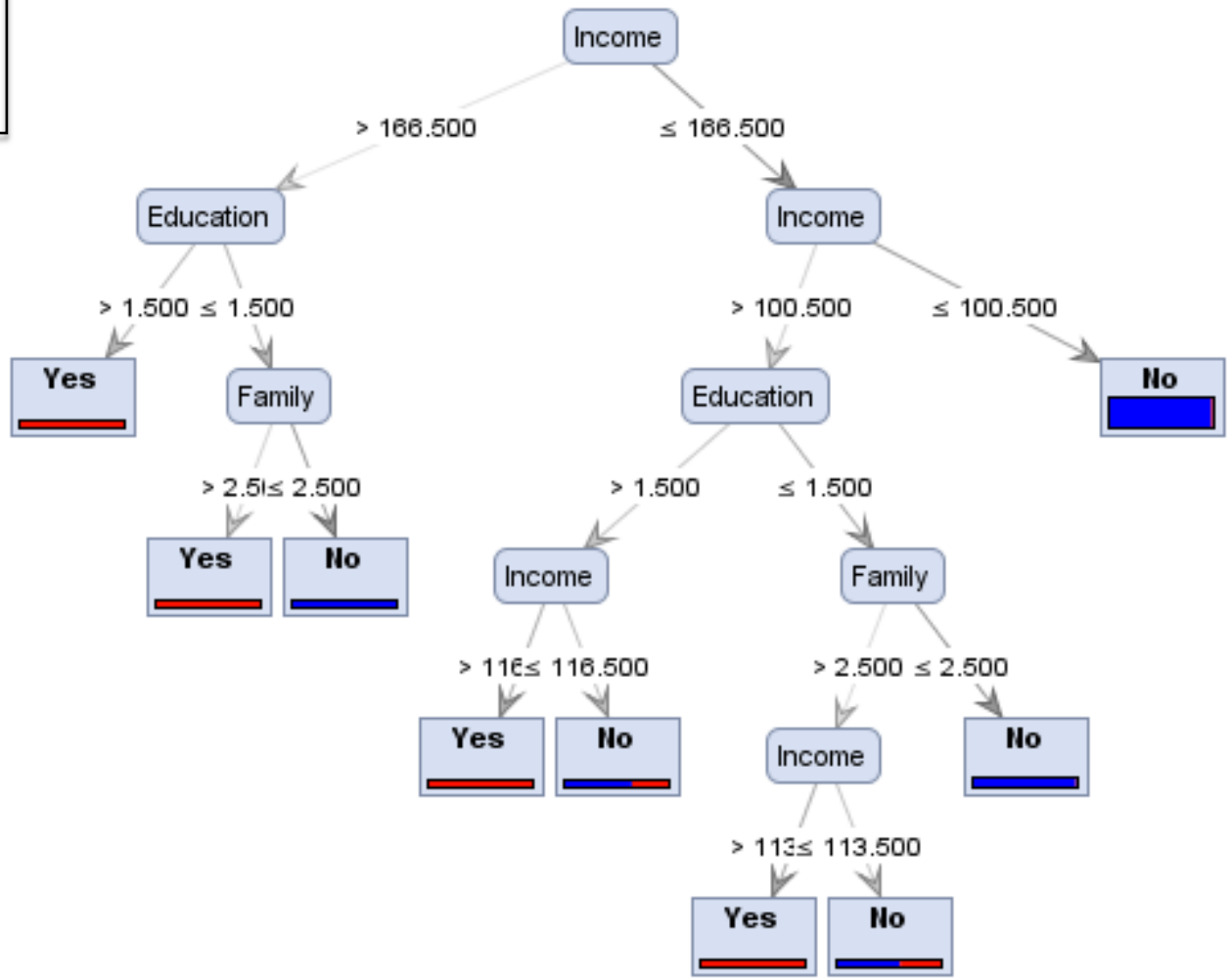
Information Gain

- G("Gender") = 0.9
- G("Lives_in_USA") = 0.73
- G("Is_Politician") = 0.36
- …

Entropy of a set (or dataset)

Decision Trees

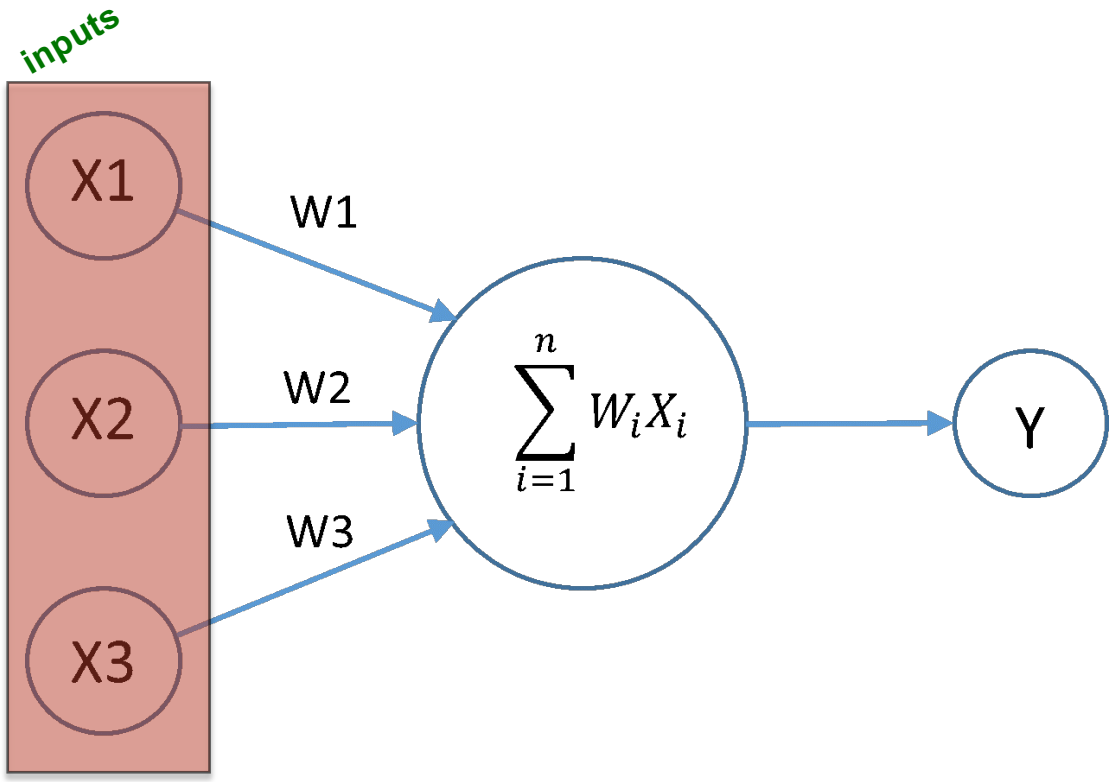
Loan Concession Application



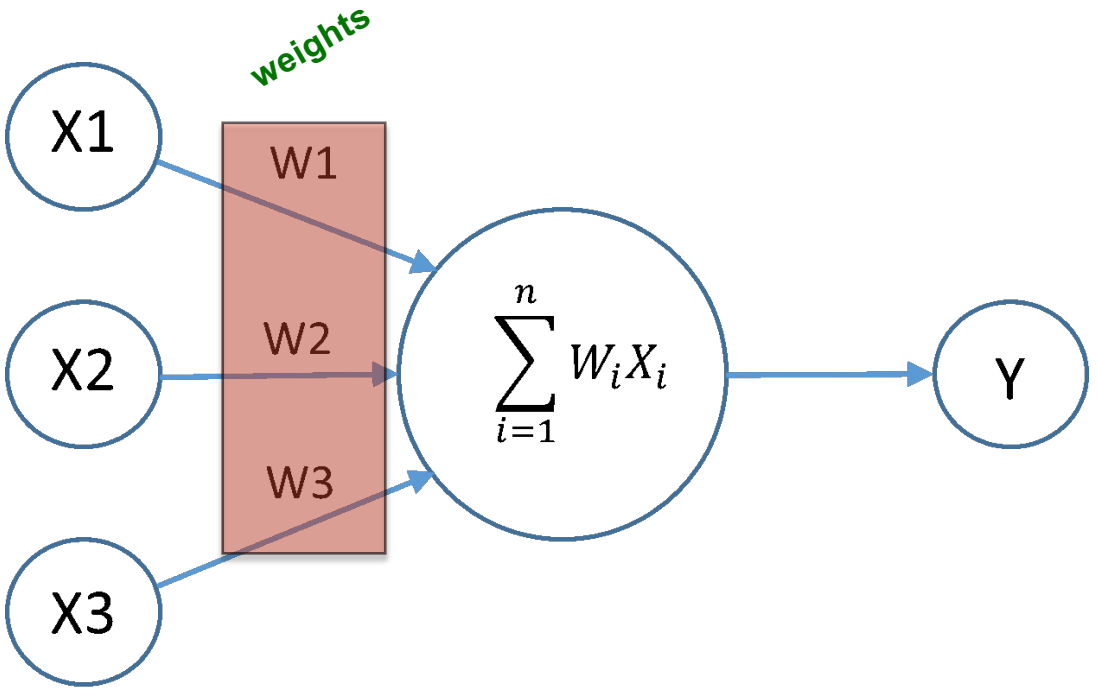
Neural Networks

- Pattern recognition
  - image processing, facial recognition, vision for robotics
  - speech recognition
  - text classification
  - machine translation
  - super-human performance in videogames
  - weather forecasting
  - etc.

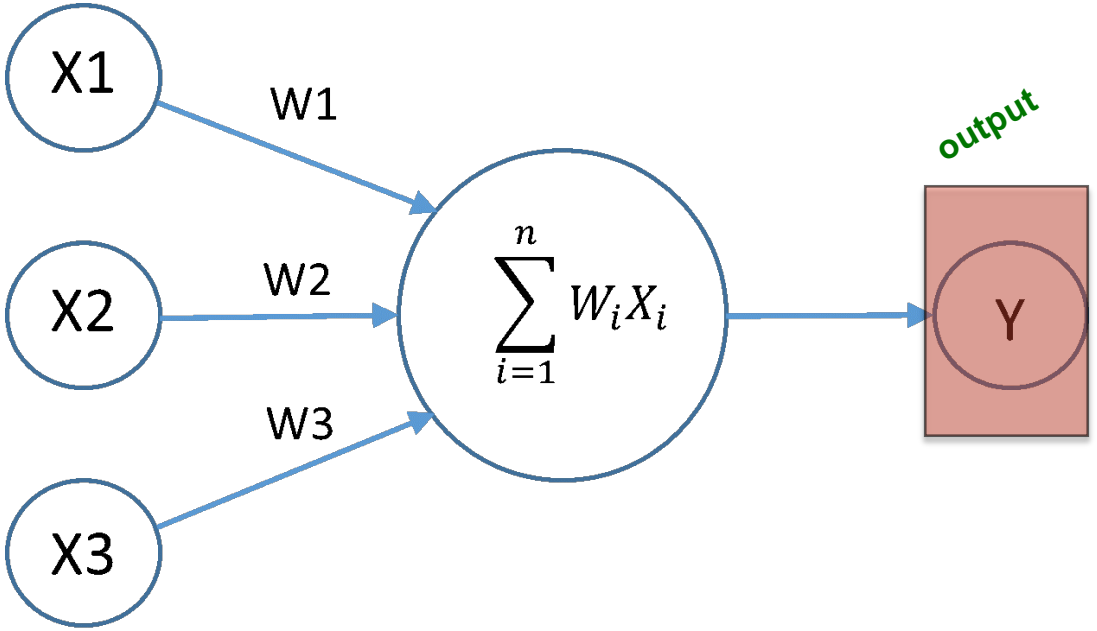
A Simple Model of a Neuron



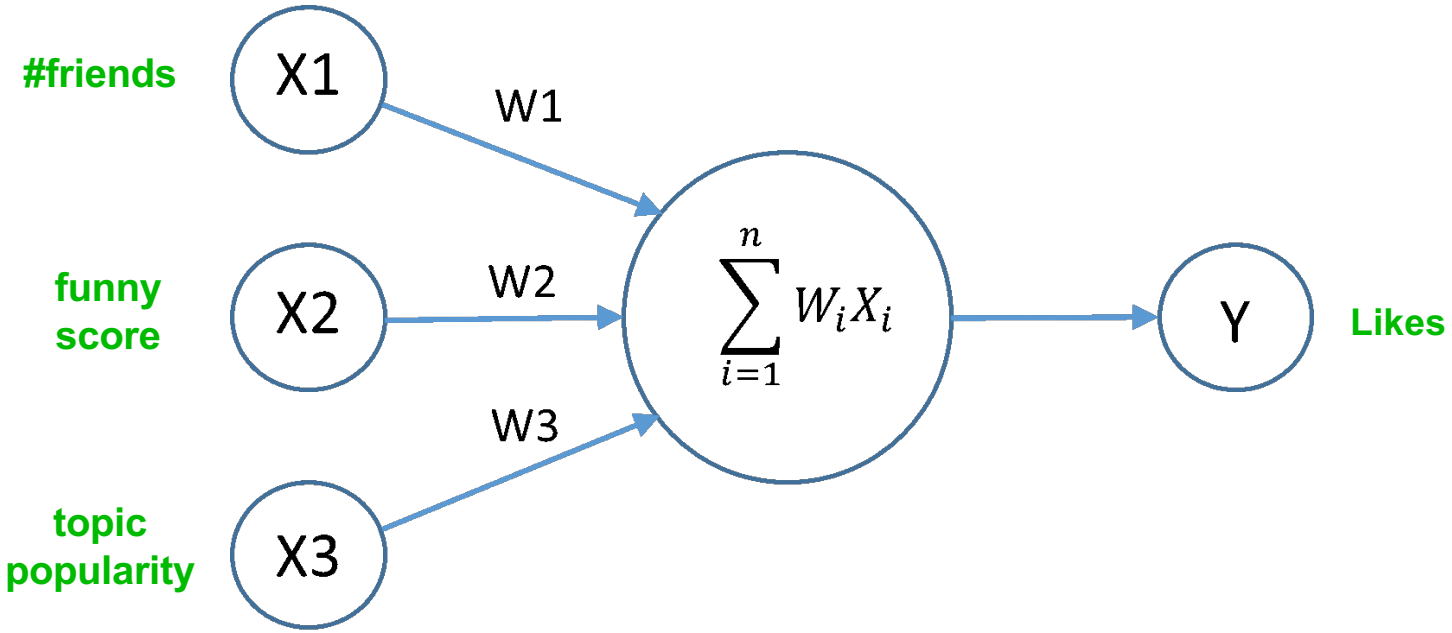
A Simple Model of a Neuron



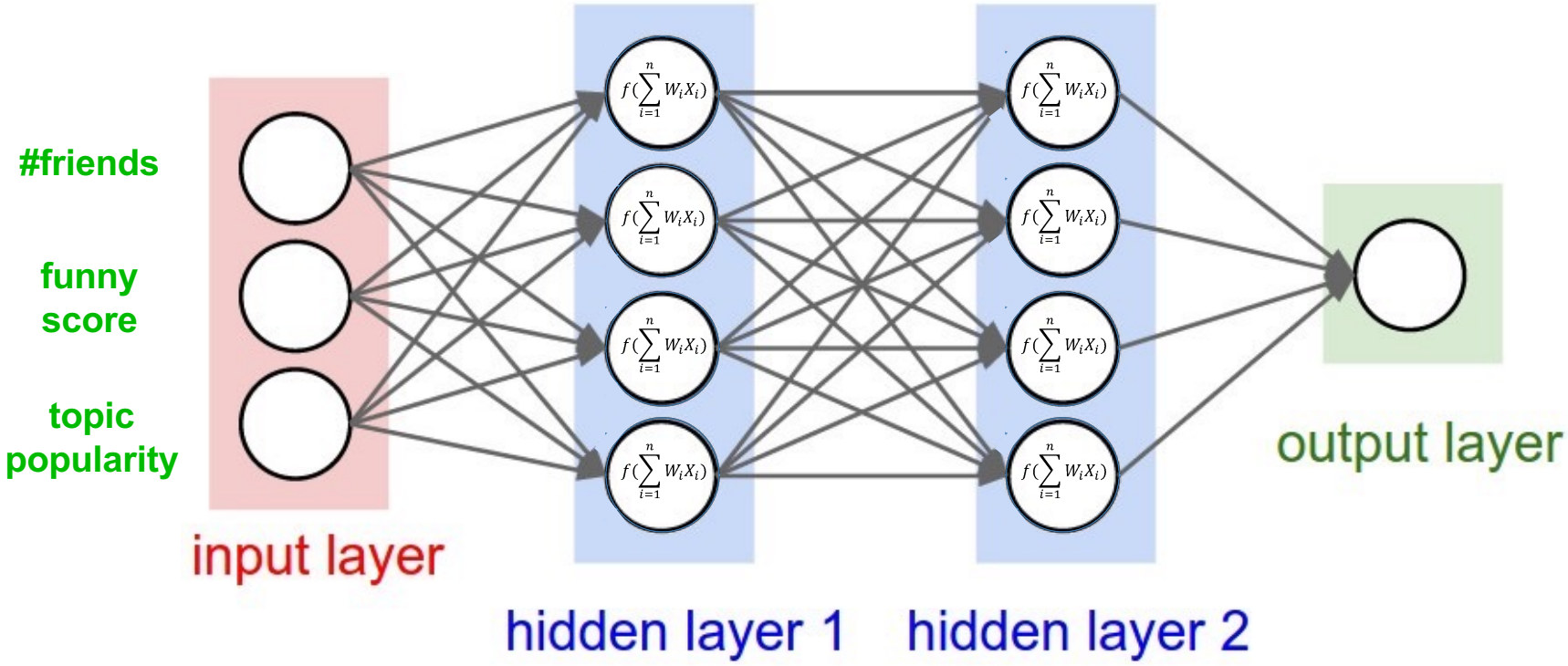
A Simple Model of a Neuron



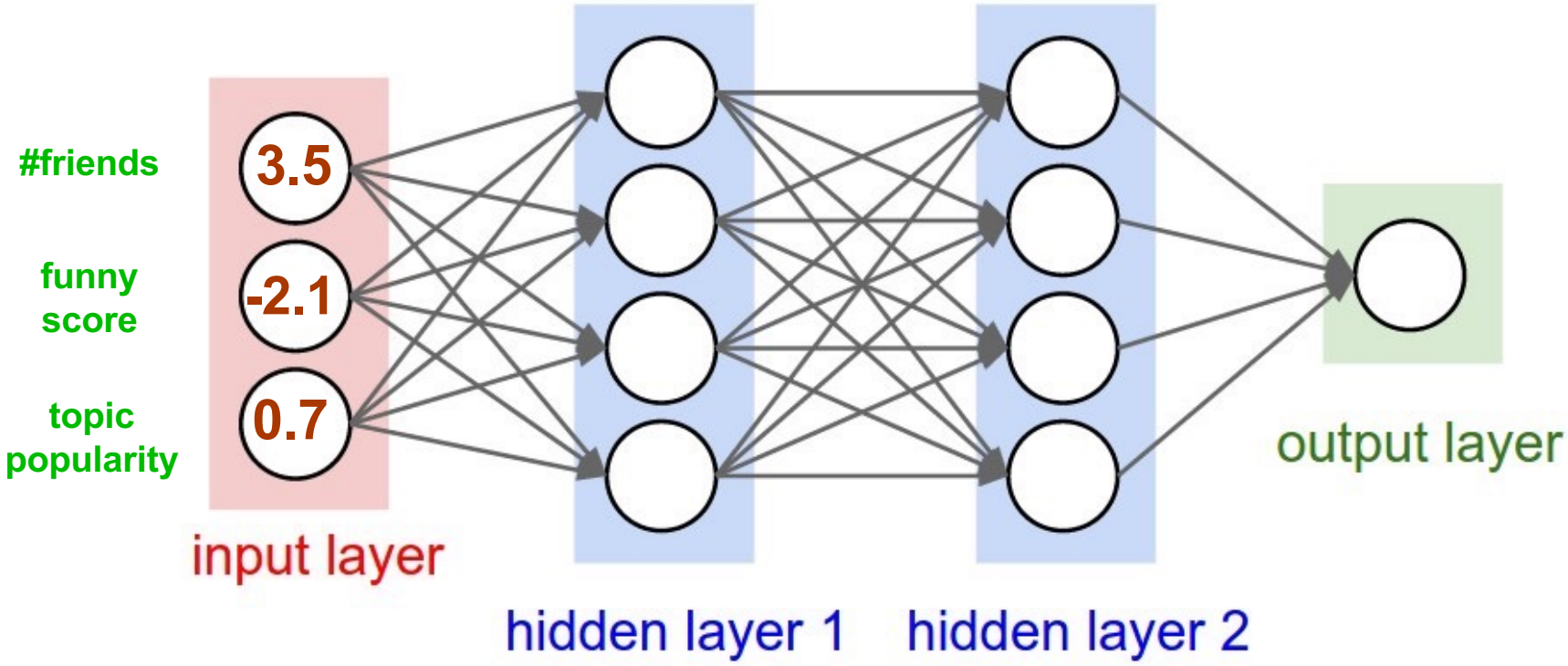
A Simple Model of a Neuron



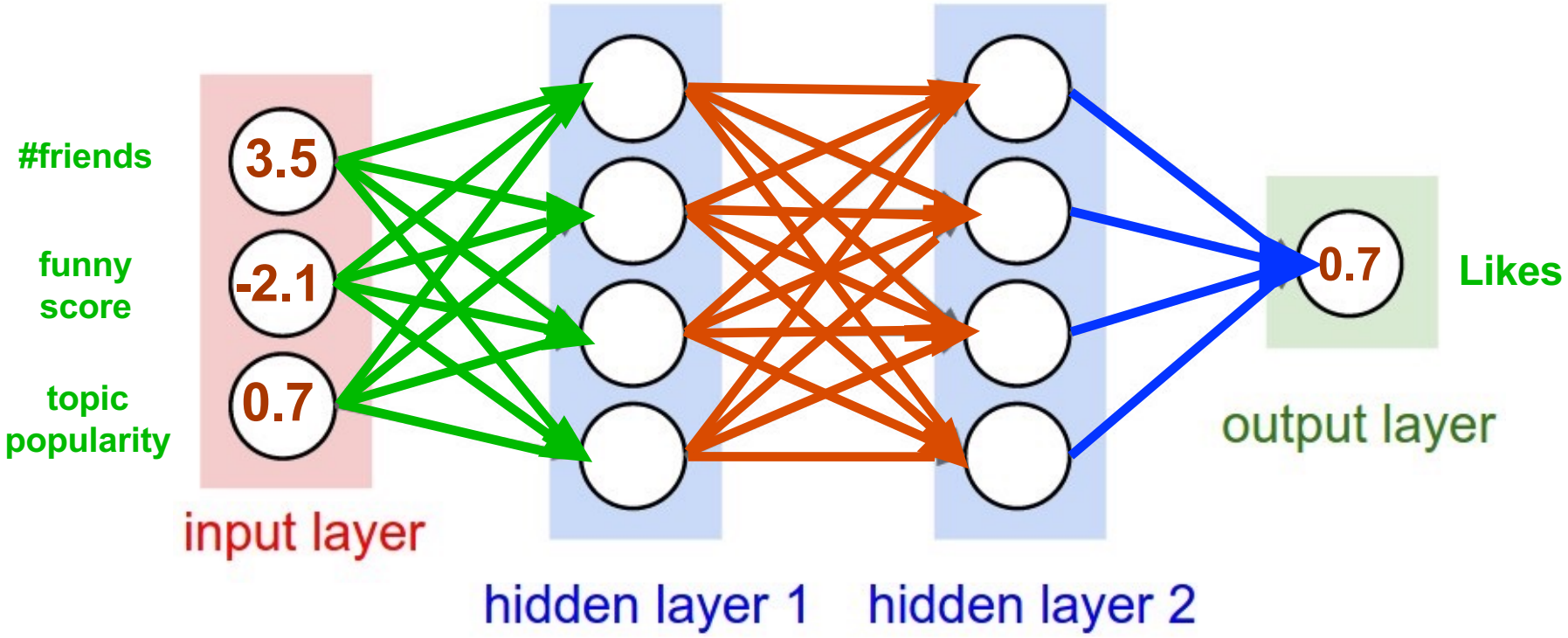
Multi-Layer Neural Networks



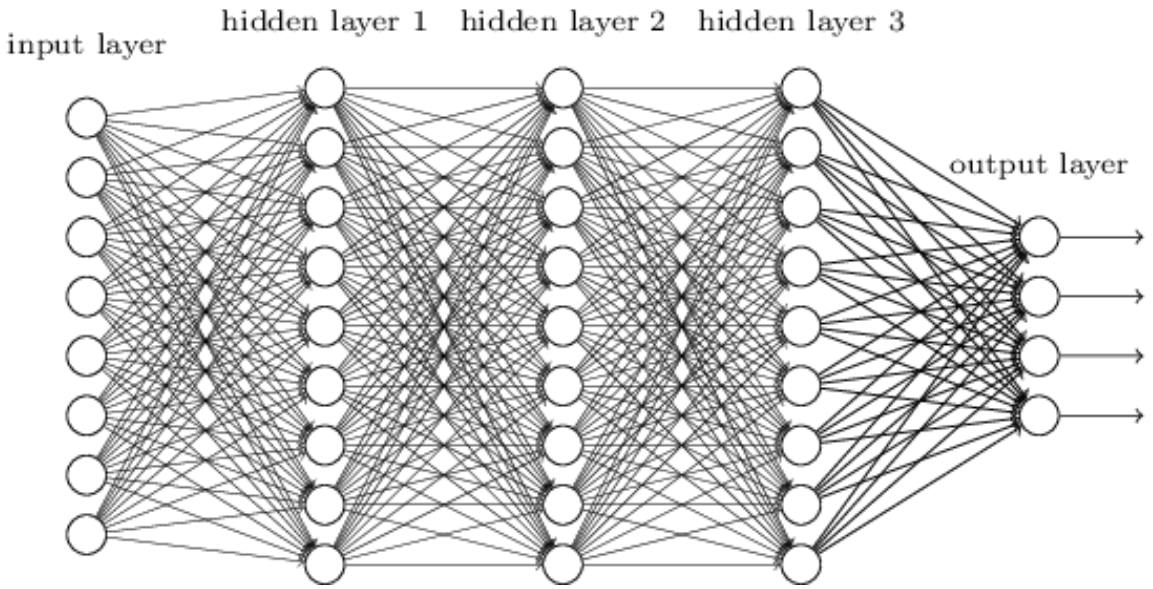
Multi-Layer Neural Networks



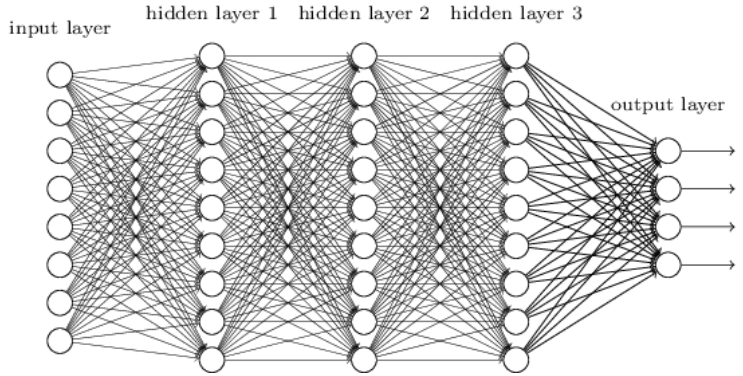
Multi-Layer Neural Networks



Multi-Layer Neural Networks



Handwriting Recognition



Handwriting Recognition

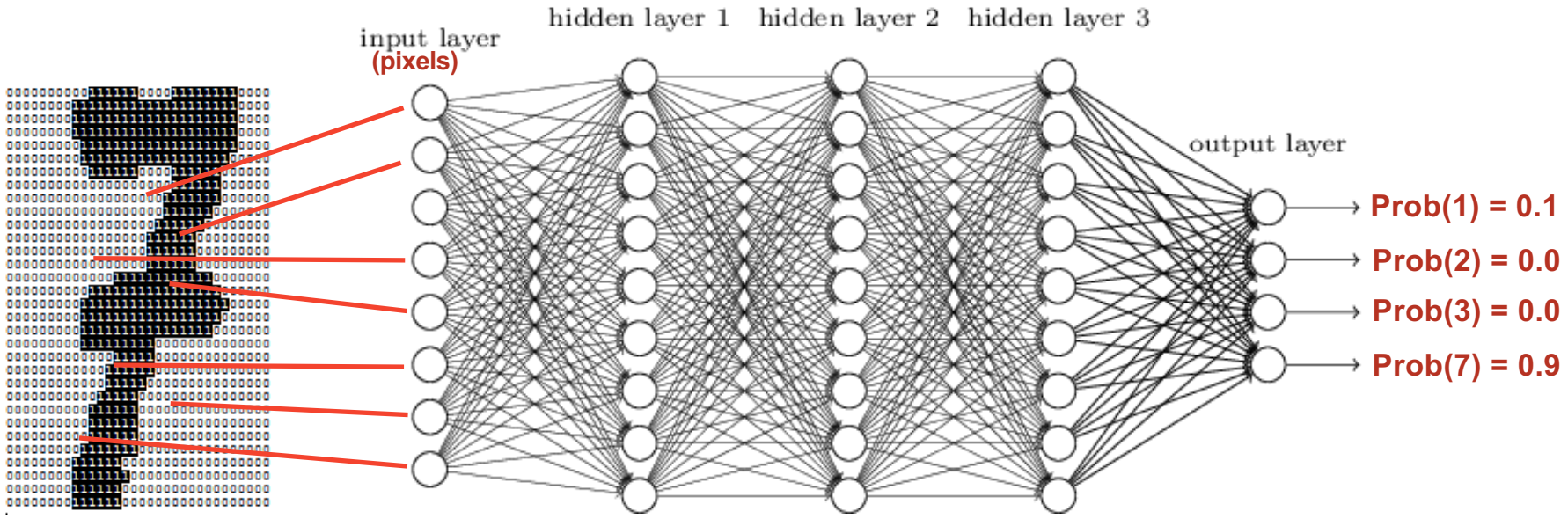
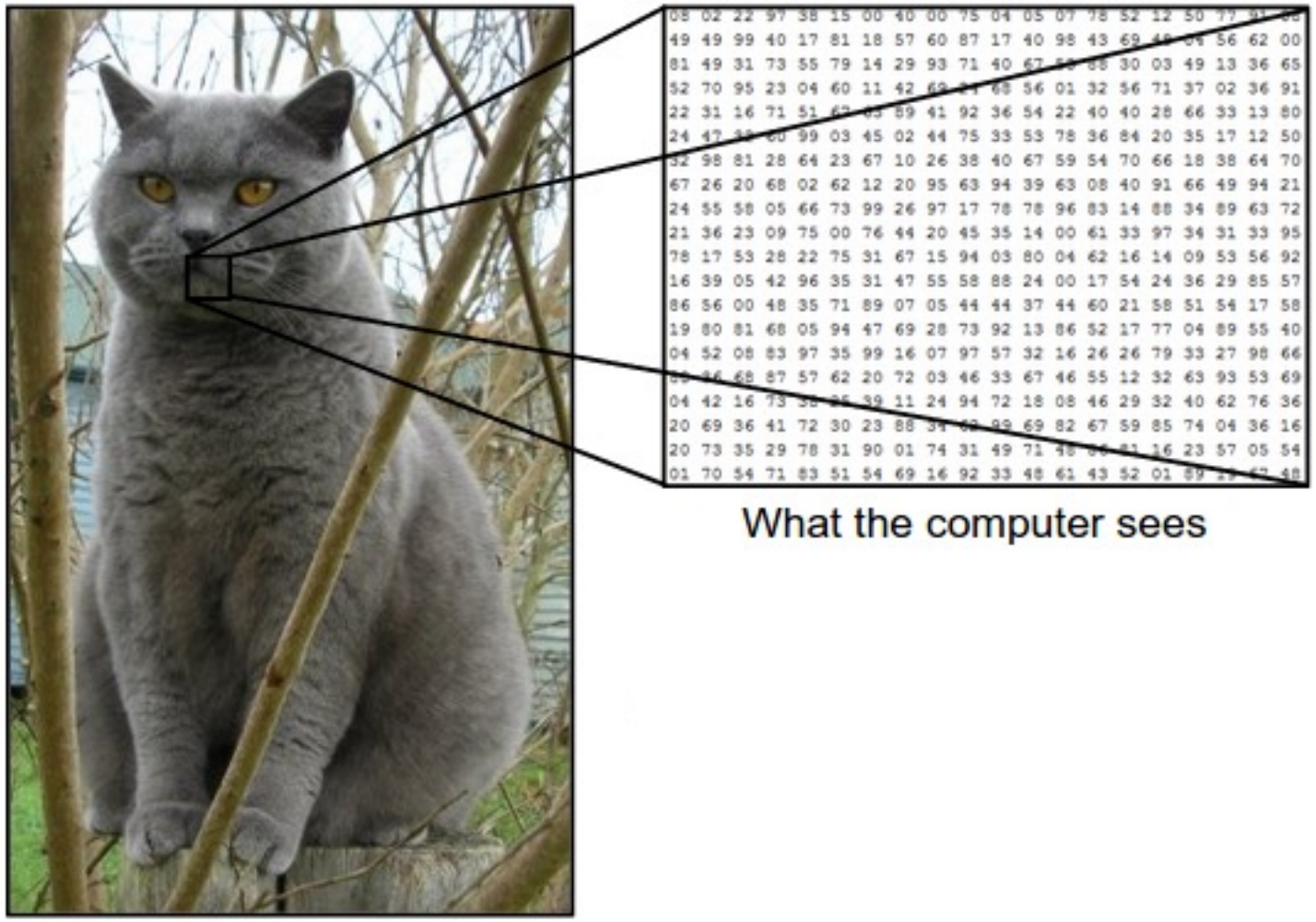


Image Classification: a core task in Computer Vision

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

cat

The problem:
semantic gap

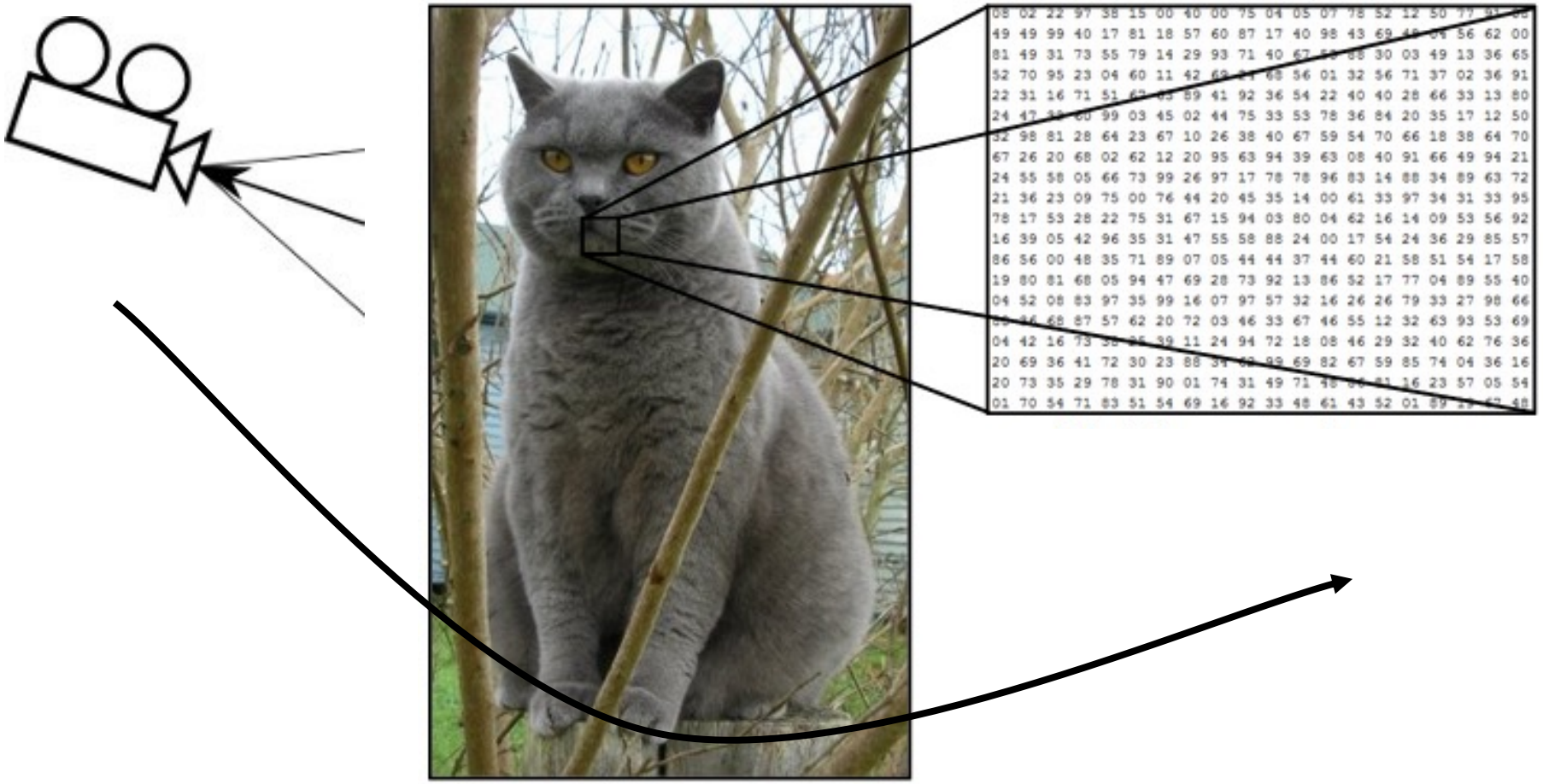


Images are represented as 3D arrays of numbers, with integers between [0, 255].

E.g.
300 x 100 x 3

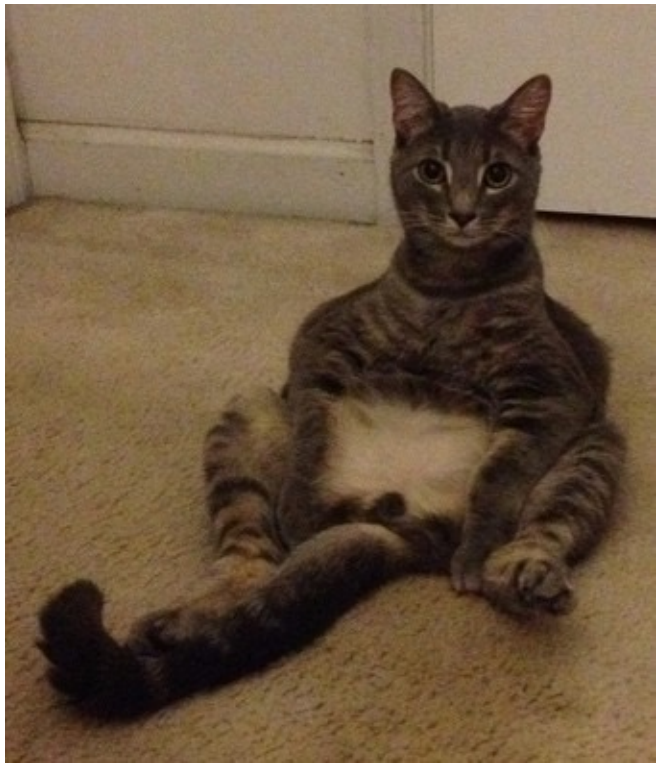
(3 for 3 color channels RGB)

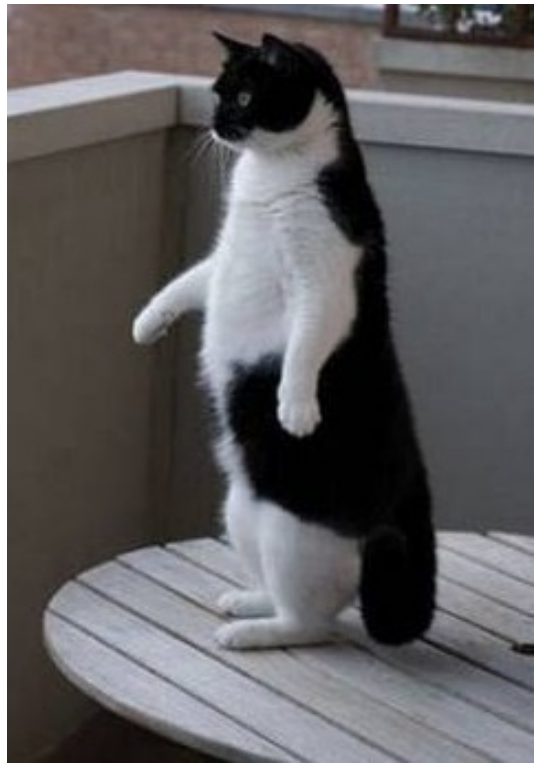
Challenges: Viewpoint Variation



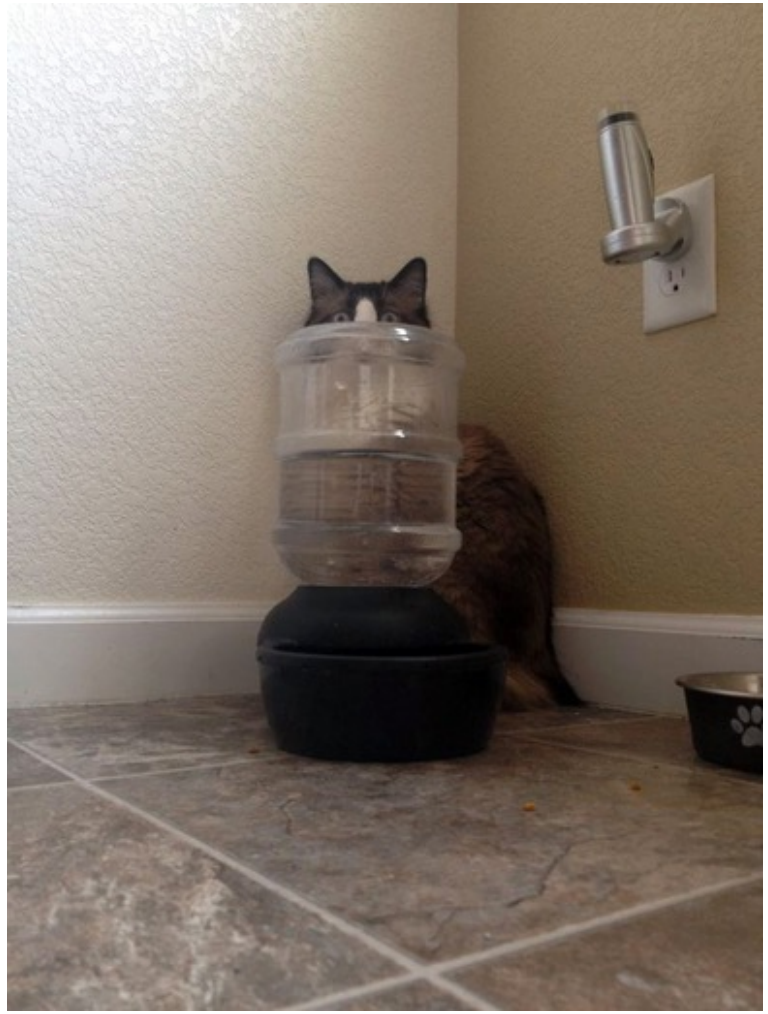
Challenges: Illumination

Challenges: Deformation



Challenges: Occlusion



Challenges: Background clutter

Challenges: Intraclass variation

An image classifier

```
def predict(image):
    # ????
    return class_label
```

Unlike e.g. sorting a list of numbers,

no obvious way to hand-code the algorithm for recognizing a cat, or other classes.

Data-driven approach:

1. Collect a dataset of images and labels
2. Use Machine Learning to train an image classifier
3. Evaluate the classifier on a withheld set of test images

```
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

Example training set

First classifier: **Nearest Neighbor Classifier**

```
def train(train_images, train_labels):
  # build a model for images -> labels...
  return model

def predict(model, test_images):
  # predict test_labels using the model...
  return test_labels
```

Remember all training images and their labels

Predict the label of the most similar training image