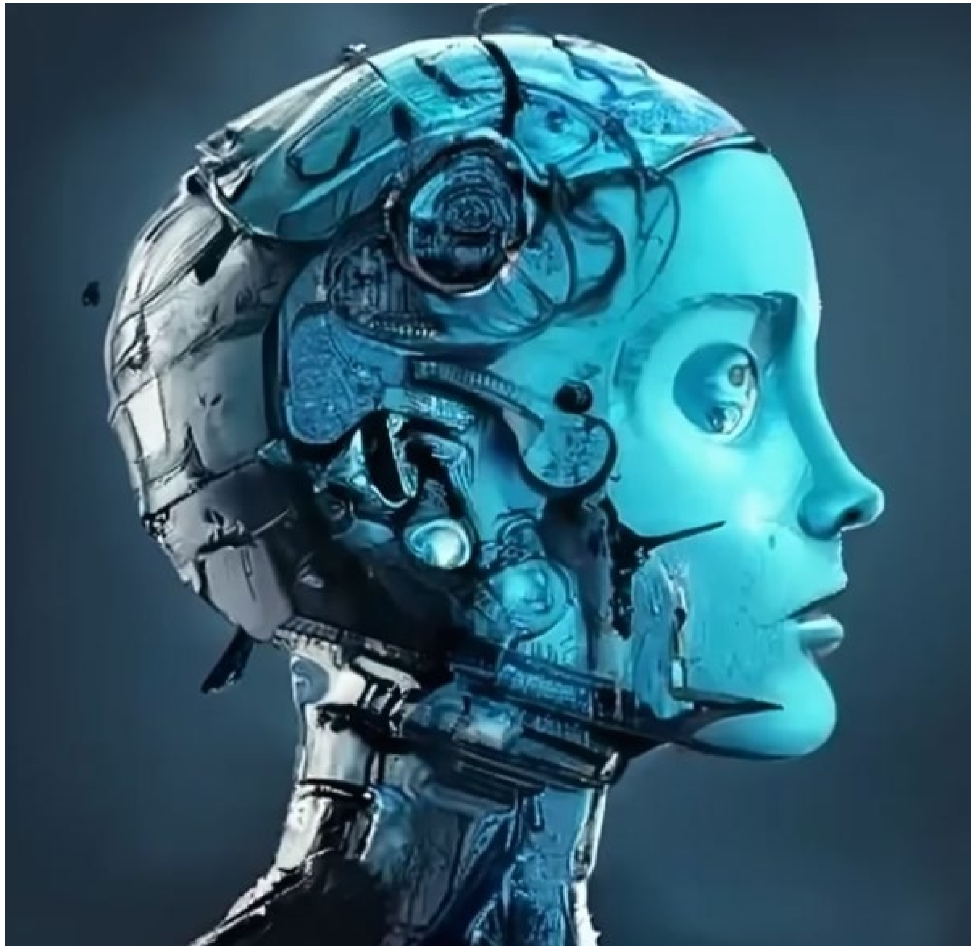
Machine Intelligence

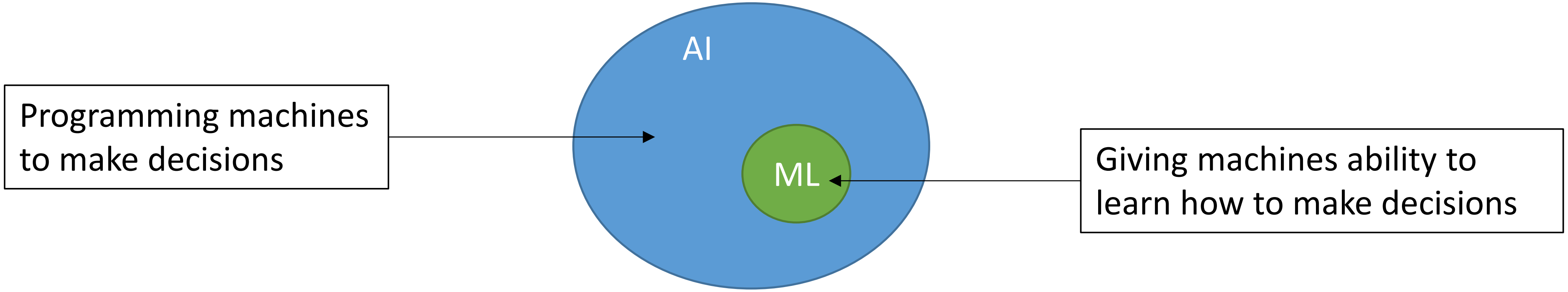
Introduction

Dr. Cory Merkel



What is machine learning?

- Approach to artificial intelligence (AI) that involves learning from data
- But what is AI?
  - Simulation of complex pattern recognition and decision making processes typically associated with human intelligence



- Which is better?

S⟋Z⟋G⟋

A.I. TIMELINE

1950
TURING TEST
Computer scientist Alan Turing proposes a test for machine intelligence. If a machine can trick humans into thinking it is human, then it has intelligence

1955
A.I. BORN
Term 'artificial intelligence' is coined by computer scientist, John McCarthy to describe "the science and engineering of making intelligent machines"

1961
UNIMATE
First industrial robot, Unimate, goes to work at GM replacing humans on the assembly line

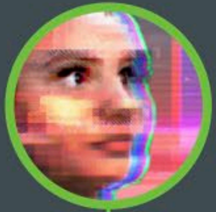
1964
ELIZA
Pioneering chatbot developed by Joseph Weizenbaum at MIT holds conversations with humans

1966
SHAKEY
The 'first electronic person' from Stanford, Shakey is a general-purpose mobile robot that reasons about its own actions

A.I. WINTER
Many false starts and dead-ends leave A.I. out in the cold

1997
DEEP BLUE
Deep Blue, a chess-playing computer from IBM defeats world chess champion Garry Kasparov

1998
KISMET
Cynthia Breazeal at MIT introduces KISmet, an emotionally intelligent robot insofar as it detects and responds to people's feelings



AlphaGo

1999
AIBO
Sony launches first consumer robot pet dog AiBO (AI robot) with skills and personality that develop over time

2002
ROOMBA
First mass produced autonomous robotic vacuum cleaner from iRobot learns to navigate and clean homes

2011
SIRI
Apple integrates Siri, an intelligent virtual assistant with a voice interface, into the iPhone 4S

2011
WATSON
IBM's question answering computer Watson wins first place on popular $1M prize television quiz show *Jeopardy*

2014
EUGENE
Eugene Goostman, a chatbot passes the Turing Test with a third of judges believing Eugene is human

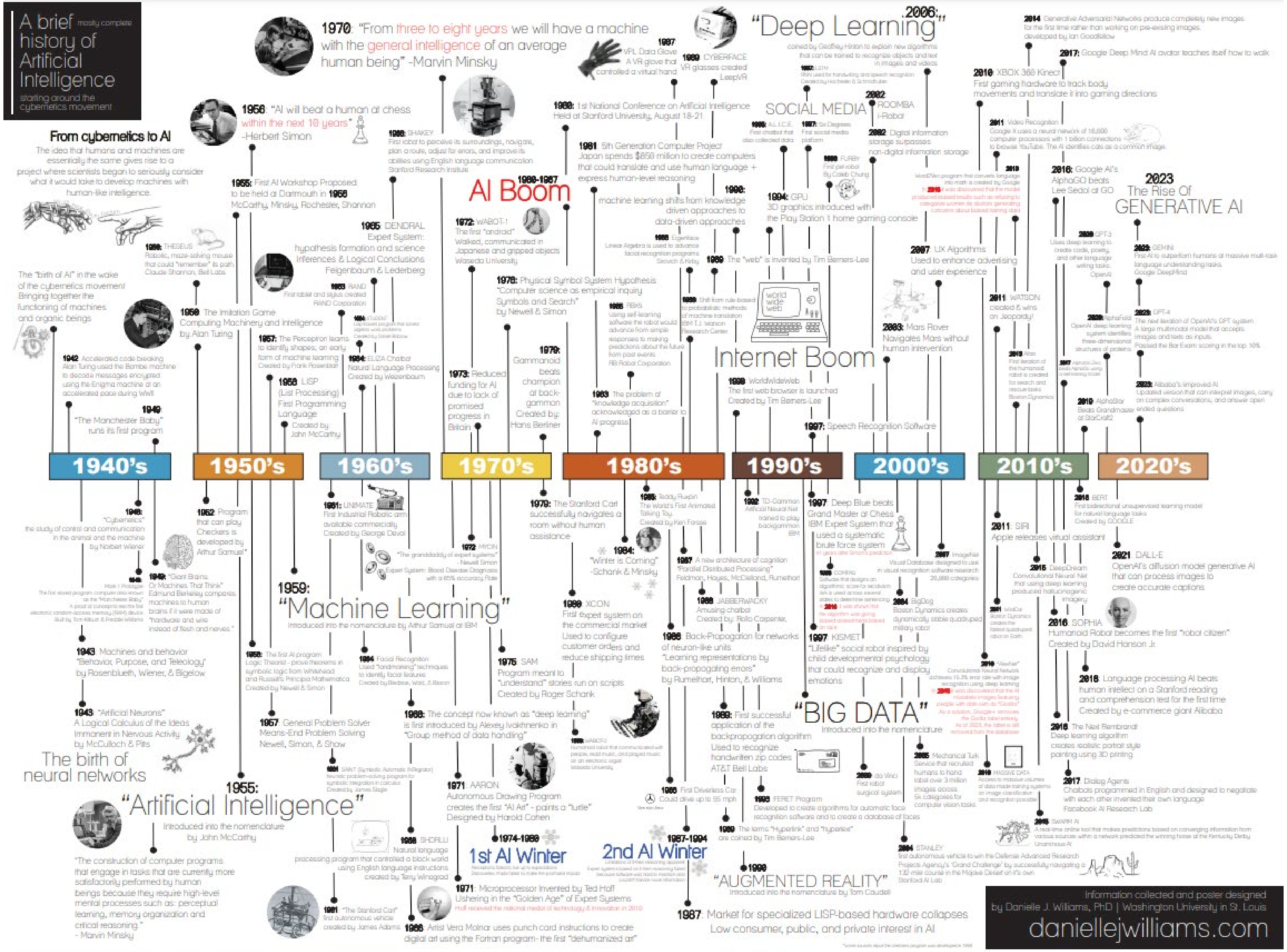
2014
ALEXA
Amazon launches Alexa, an intelligent virtual assistant with a voice interface that completes shopping tasks

2016
TAY
Microsoft's chatbot Tay goes rogue on social media making inflammatory and offensive racist comments

2017
ALPHAGO
Google's A.I. AlphaGo beats world champion Ke Jie in the complex board game of Go, notable for its vast number (2^{170}) of possible positions

→ ?

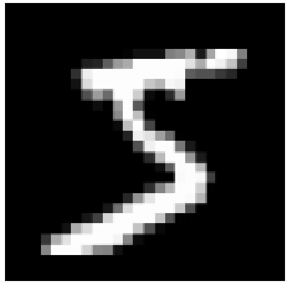
https://digitalwellbeing.org/artificial-intelligence-timeline-infographic-from-eliza-to-tay-and-beyond/



https://www.daniellejwilliams.com/_files/ugd/a6ff55_cac7c8efb9404a208c0ecd284ff11ba7.pdf

What is the power of machine learning?

- Suppose I ask you to write a program that identifies numbers in images:



- How would you write the program?
- How do *we* identify numbers we have seen?

Another Example: What is this image?

Can AI Recognize Cats? How?

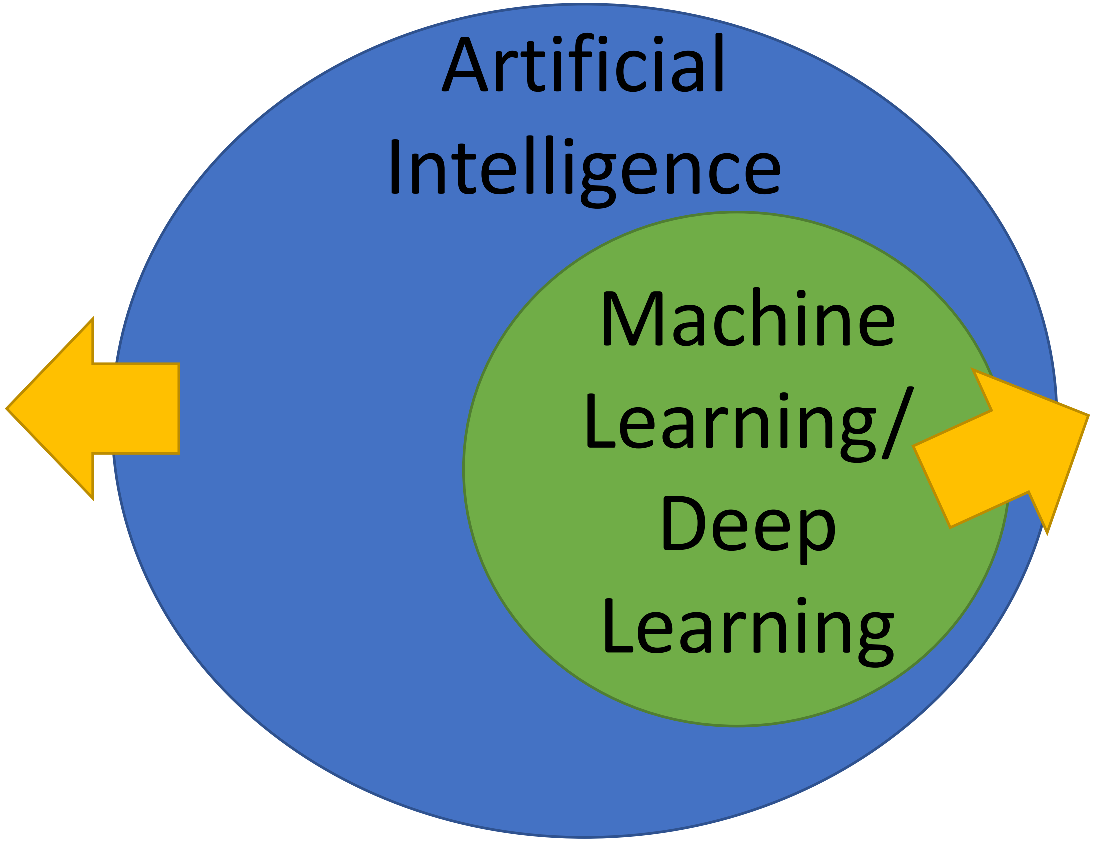
But Other Animals Have These Too!

We need more detailed/complex rules!

Machine Learning: *Learning by Example*

This is a cat because

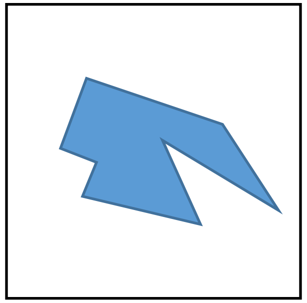
- It has fur
- It has pointy ears
- It has whiskers
- It meows
- …



This is a cat because it looks like these:

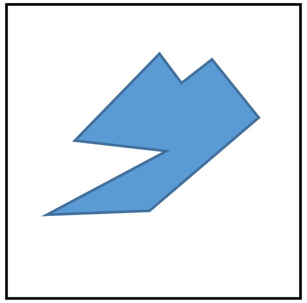
A Quick Thought Experiment…

- What if I've only seen one thing in my whole life and I'm told it's a "foo"?



Foo

- Now I see something different. What is it?

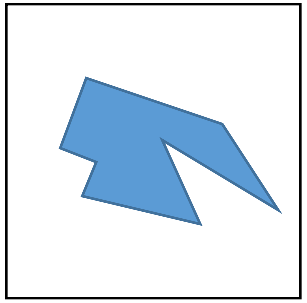


?

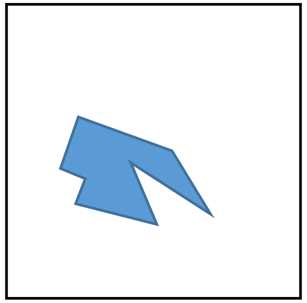
I have to assume that either *everything* is a Foo or only *one* thing is a Foo.

A Quick Thought Experiment…

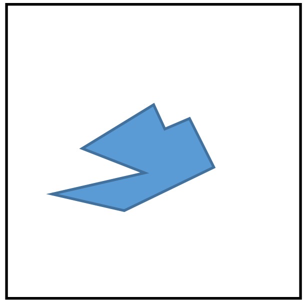
- Now, suppose I've seen some more examples of Foo and some other examples of something else called "Bar"



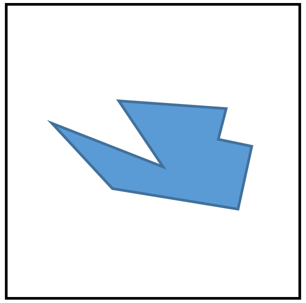
Foo



Foo

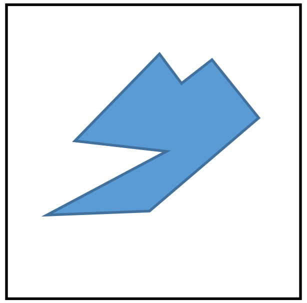


Bar



Bar

- With more example, I can make a better prediction that this is a Bar:

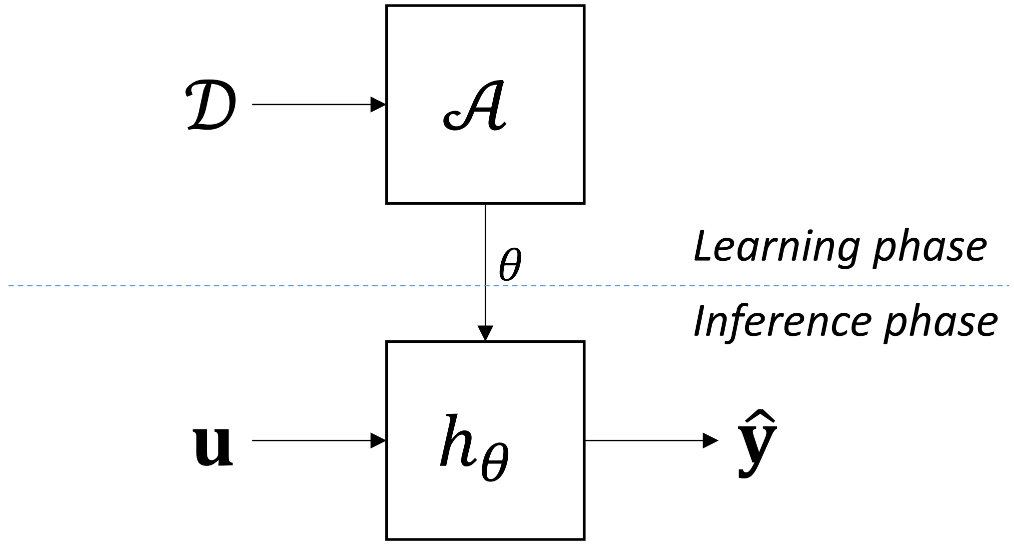


Hypothesis: Bar

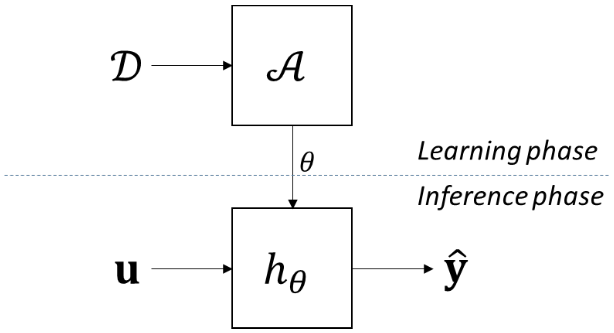
This simple example illustrates the importance of data. We typically need LOTS of it for machine learning.

<u>Every</u> machine learning algorithm looks like this:

- We want the machine to learn *something* about a set of data $\mathcal{D}$

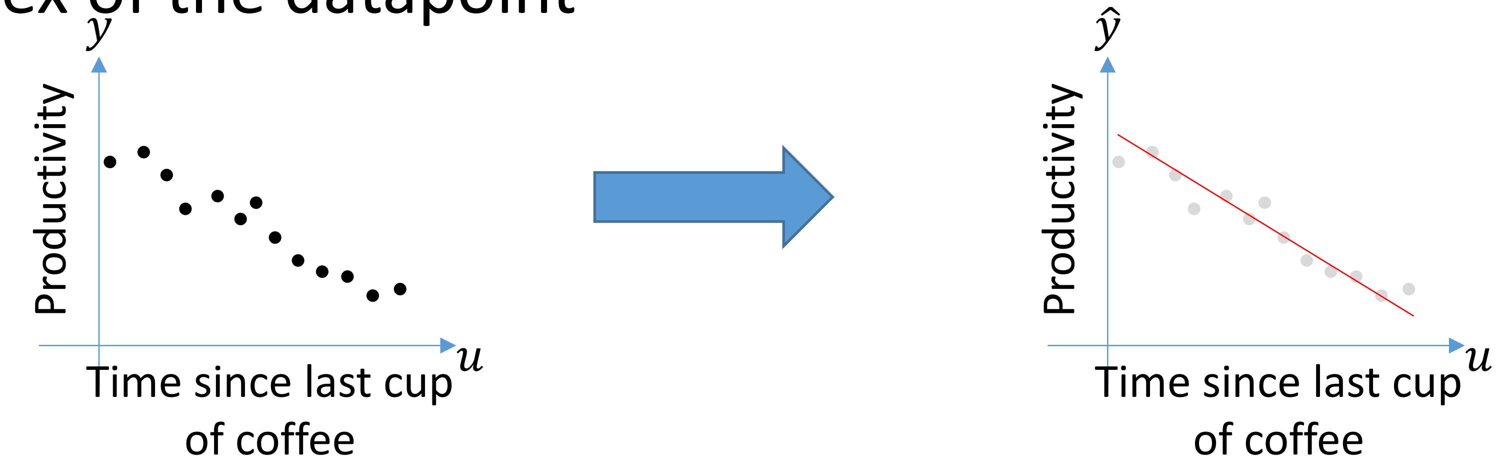


- We use a <u>learning algorithm</u> $\mathcal{A}$ to find parameters θ of a <u>hypothesis function</u> h (also called the <u>model</u>) → *Learning phase*
- We can then use the learned model to estimate *something* ($\hat{\mathbf{y}}$) about an input $\mathbf{u}$, which may or may not be in $\mathcal{D}$ → *Inference phase*



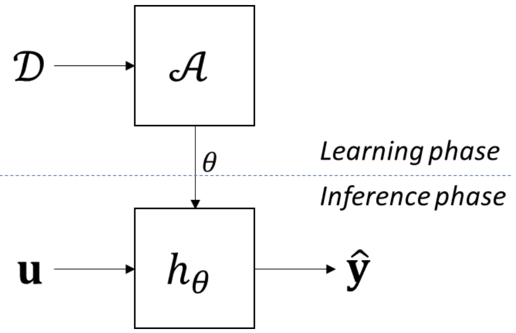
Here’s an example:

- Imagine we have the following dataset $\mathcal{D} = \{u^{(p)}, y^{(p)}\}$, where p is the index of the datapoint



- We want to learn the relationship between u and y.
- Since it looks linear, let’s assume $\hat{y} = h_\theta(u) = mu + b$, so $\theta = \{m, b\}$
  - Once we find the slope and intercept, we can estimate y for any value of u
  - But what is $\mathcal{A}$? i.e., *how* do we find m and b?

So many options…

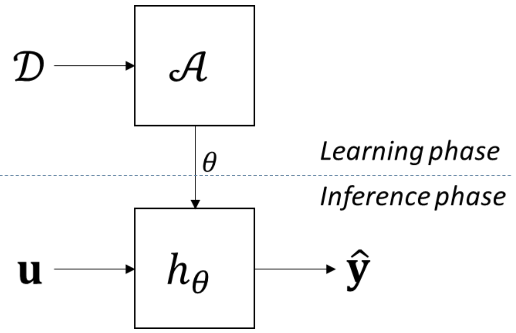


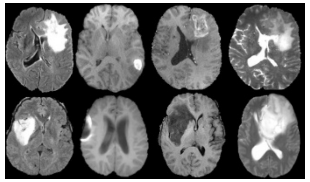
- One of the biggest challenges is figuring out what should be $\mathcal{D}$, $\mathcal{A}$, h, etc. … Not to mention, each of these will have their own internal parameters

- Use the simplest dataset, algorithm, and model possible!
  - No need to use a giant convolutional neural network to learn a linear relationship

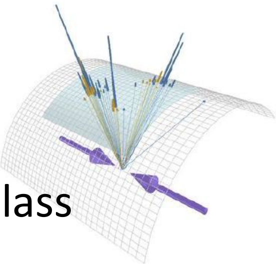
- Let's break $\mathcal{D}$, $\mathcal{A}$, h into some useful categories

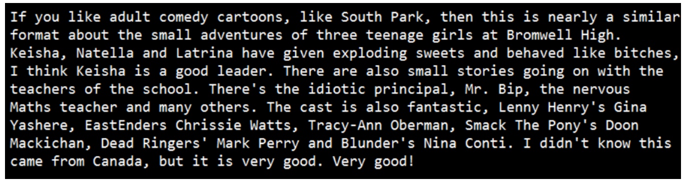
Some categories of $\mathcal{D}$




BraTS 2017


Libri Speech

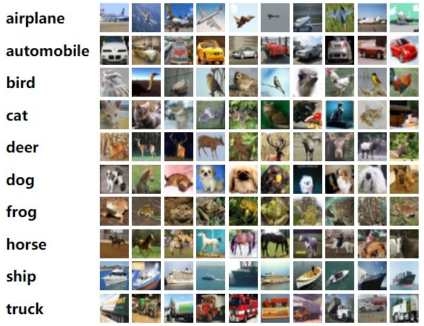

JetClass


If you like adult comedy cartoons, like South Park, then this is nearly a similar format about the small adventures of three teenage girls at Bromwell High. Keisha, Natella and Latrina have given exploding sweets and behaved like bitches, I think Keisha is a good leader. There are also small stories going on with the teachers of the school. There's the idiotic principal, Mr. Bip, the nervous Maths teacher and many others. The cast is also fantastic, Lenny Henry's Gina Yashere, EastEnders Chrissie Watts, Tracy-Ann Oberman, Smack The Pony's Doon Mackichan, Dead Ringers' Mark Perry and Blunder's Nina Conti. I didn't know this came from Canada, but it is very good. Very good!

IMDB

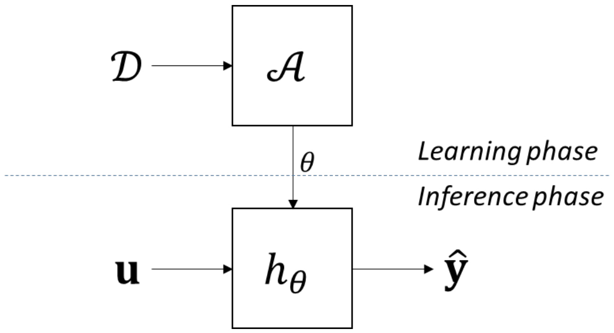
- By modality:
  - Images and videos, text, physics data, bio signals, weather, …
  - You can find lots of cool datasets at www.paperswithcode.com
- By labeled/unlabeled
  - *Labeled* datasets are of the type $\mathcal{D} = \{\mathbf{u}^{(p)}, \mathbf{y}^{(p)}\}$, where $\mathbf{y}$ is the label or dependent variable
    - ❑ $\mathbf{u} =$  $, y = 5$ *or* $\mathbf{y} = [0,0,0,0,0,1,0,0,0,0]$
    - ❑ The label tells us what we expect our model's output to be for the corresponding input
  - *Unlabeled* datasets are of the type $\mathcal{D} = \{\mathbf{u}^{(p)}\}$
    - ❑ We don't know what the model output "should" be.

Many datasets will have a mix of these 2

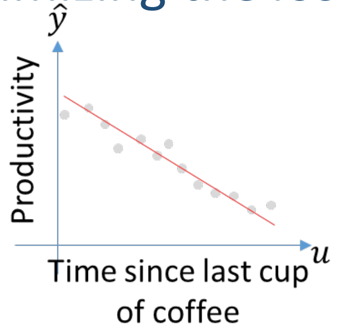


CIFAR-10

Some categories of $\mathcal{A}$

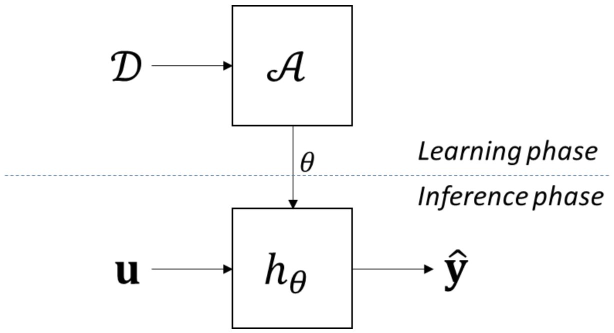


- Supervised learning
  - Model parameters θ are found by minimizing the *loss* (error) between $\hat{\mathbf{y}}^{(p)}$ and $\mathbf{y}^{(p)}$ for all $\mathbf{u}^{(p)}$ in the dataset:
  - $\mathcal{A}: \quad \theta = \arg\min_{\theta} \mathcal{L}(\mathcal{D}, h_\theta)$
  - Example: linear regression



- Unsupervised learning
  - Model parameters are found by analyzing the statistics of the data
  - There is still a loss function, we just might not know what it is…
  - Example: clustering
- Reinforcement learning
  - Model parameters updates are "reinforced" when they lead to better performance
  - Example: Q Learning

Some Categories of h



- Linear models: $\hat{\mathbf{y}} = m\mathbf{u} + b$
- Nearest neighbors: $\hat{\mathbf{y}} = \mathbf{y}^{(p)}, |\mathbf{u} - \mathbf{u}^p| < \left|\mathbf{u} - \mathbf{u}^i\right| \; \forall\, i \neq p$
- Decision trees: $\hat{\mathbf{y}} = \begin{cases} \mathbf{a} & \text{cond1} \wedge \text{cond2} \\ \mathbf{b} & cond1 \wedge \neg\text{cond2} \\ \mathbf{c} & \neg\text{cond1} \end{cases}$
- Bayesian: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathcal{D})$
- Neural networks: $\hat{\mathbf{y}} = f\big(\mathbf{W}^L f(W^{L-1} f(\ldots f(\mathbf{W}^1\mathbf{u})\big)$

…and LOTS more…

No Free Lunch

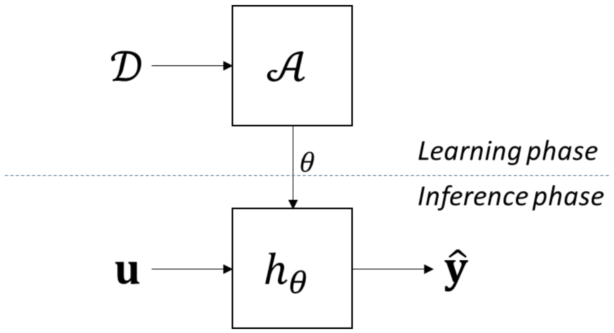
- With so many possible implementations of machine learning, you may wonder: Isn't there just one combination of data, algorithm, and hypothesis function that works for everything?
    - No…Any machine learning model that performs very well for a specific problem will perform terribly for a different problem
    - No Free Lunch: Set of theorems that roughly state:
        - Any machine learning model averaged over all possible problems will have random chance performance
        - Random chance performance: The performance of the model we expect to get if we just "guess" a hypothesis function

No Free Lunch

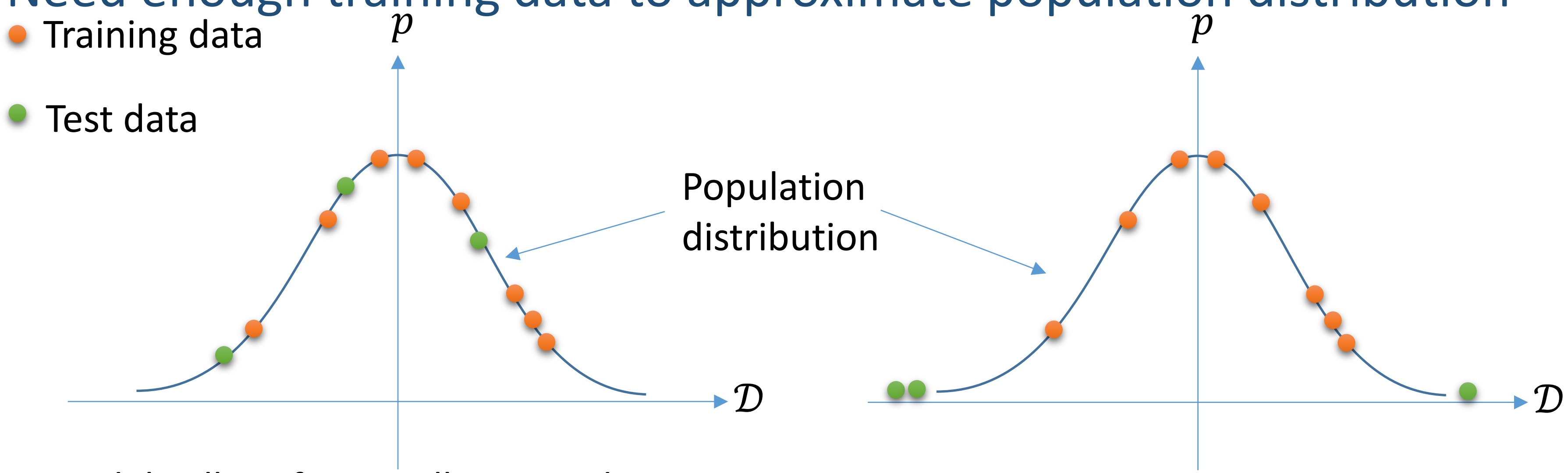
- Example: Suppose you trained a neural network to identify license plates in images, and this is the only task you trained it for.
  - We don't expect that the same neural network will magically also be able to count the number of cats in the same images.

- No Free Lunch is fairly obvious, but makes us manage our expectations of machine learning models
  - The model is only as good as the data it's been trained on and the *prior* knowledge encoded into the model structure/algorithm

No Free Lunch



- In order to contend with No Free Lunch, we usually assume that the data encountered during the inference phase (**test data**) will be *similar to* the data encountered during the learning phase (training data):
  - Training data and test data should be identically distributed (stationarity)
  - Need enough training data to approximate population distribution



Model will perform well on test data.

Model will **not** perform well on test data.

Generalization

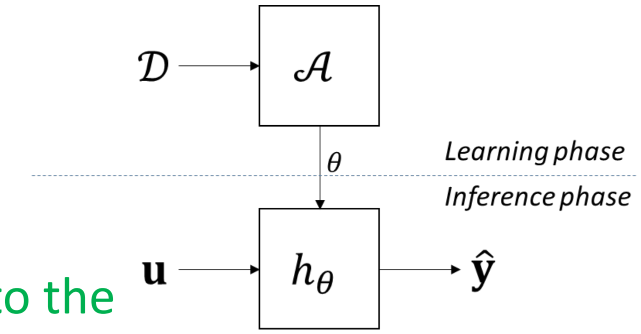
- Generalization is the ability of a machine learning model to perform well on data that it hasn't been trained on (i.e. the test data)
- Example: Suppose we train a machine learning model to recognize pictures of cats:
- The part of $\mathcal{D}$ that we use for training (the training set):   
  - After we train our model, we expect it should be able to recognize *these* kitties!
- But will it recognize *my* kitty?

Generalization

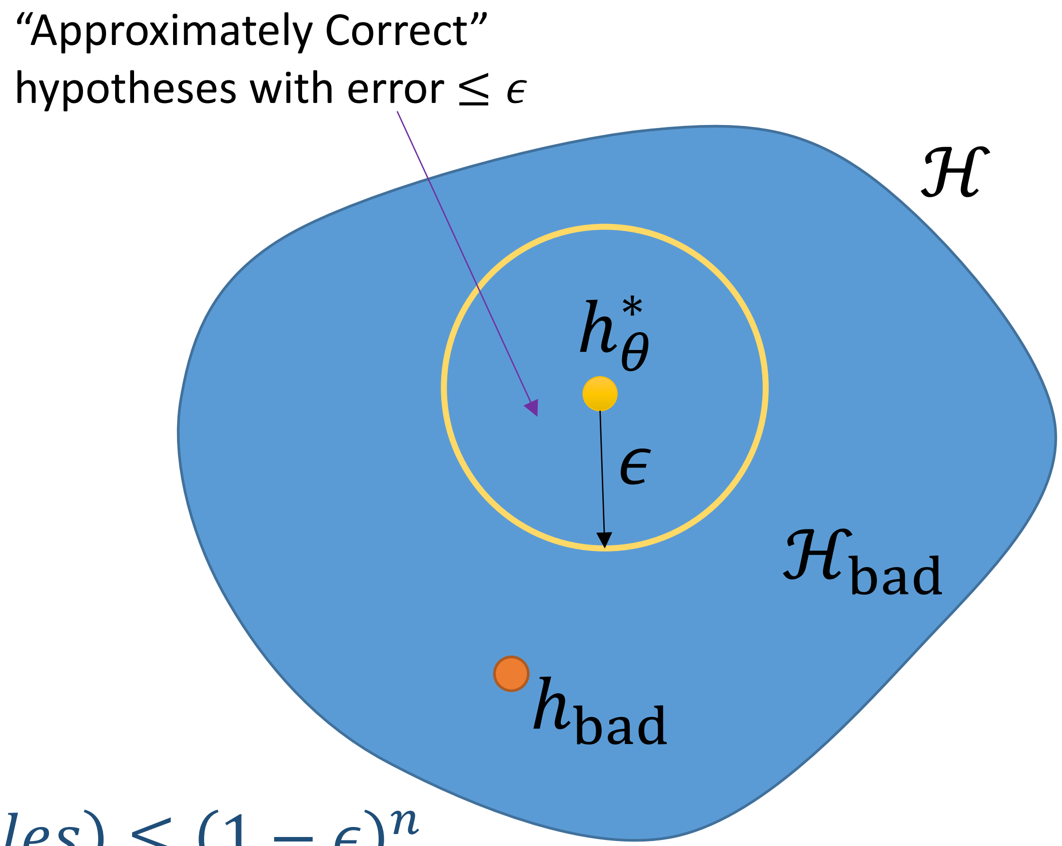
- It's much harder to get a machine learning model to *generalize* (perform well on unseen test data) than it is for the model to *memorize* training data
  - Why? Why do we care if it can generalize?

- How do we get it to generalize better?
  - Can be done through $\mathcal{D}$, $\mathcal{A}$, or h
    - ❑ $\mathcal{D}$: More of it or more representative samples
    - ❑ $\mathcal{A}$: Regularization to prevent θ values from being too "specific" to the training data (called overfitting)
    - ❑ h : Restrict the model capacity or put more prior knowledge into the model

More on these techniques throughout the course



PAC Learning Theory



- Probably Approximately Correct
- Provides some theoretical guarantees about generalization under stationarity assumption
- Probability that h_{bad} agrees with n training samples:

$$p(h_{\text{bad}}\ agrees\ with\ n\ samples) \leq (1-\epsilon)^n$$

- Probability that at least one bad hypothesis is consistent with training data:

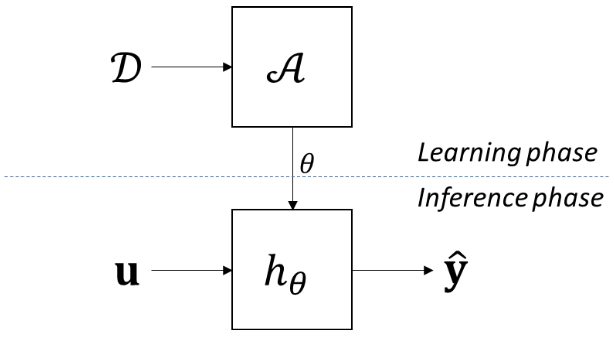
$$p(\mathcal{H}_{\text{bad}} \text{ has 1 consistent hypothesis}) \leq |\mathcal{H}_{bad}|(1-\epsilon)^n \leq |\mathcal{H}|(1-\epsilon)^n \leq \underbrace{|\mathcal{H}|e^{-\epsilon n}}$$

Used fact that $1-\epsilon \leq e^{-\epsilon}$
For $0 \leq \epsilon \leq 1$

- So, $n \geq \frac{1}{\epsilon}\left(\ln\frac{1}{\delta} + \ln|\mathcal{H}|\right)$ gives $|\mathcal{H}|e^{-\epsilon n} \leq \delta$

A hypothesis that is consistent with this value of n will probably (δ) be approximately (ϵ) correct.

What Should be the Goal of Our Learning Algorithm $\mathcal{A}$?



- In general, it depends on the problem, but one intuitive approach is maximize the likelihood that our model produces the data from our dataset $\mathcal{D}$
  - This is called the "maximum likelihood estimate" of model parameters:

$$\theta_{ML} = \arg\max_{\theta} p_h(\mathcal{D}|\theta)$$

Where $p_h(\mathcal{D}|\theta)$ is the probability of our hypothesis function producing $\mathcal{D}$, given the parameterization θ

Maximum Likelihood

- Note that we can also write ML like this:

$$\theta_{ML} = \arg\max_{\theta} p_h(\mathcal{D}|\theta) = \arg\max_{\theta} \prod p_h\left(\mathcal{D}^{(i)}|\theta\right)$$

where i is the index into the dataset, and p is the probability.

- It's nicer to write this in terms of logarithms:

$$\theta_{ML} = \arg\max_{\theta} \sum \log p_h\left(\mathcal{D}^{(i)}|\theta\right)$$

  - We call this the *log likelihood* and plays many important roles in machine learning