Grades for Assignment #1 released on Gradescope
  • Statistics (out of 40 points):
    • Median: 36
    • Mean: 34.7
    • Std. Dev.: 5.12

Assignment #3: Policy Memo due 11:59pm on March 2\textsuperscript{nd}
  • Interview with Dr. Cassandra Volpe Horii (Associate Vice Provost for Education and Director of the Center for Teaching and Learning) on Zoom at 3:30pm today
Today’s Agenda

1. History of AI
2. The machine learning revolution
3. Autonomous vehicles
4. Additional perspectives on AI
Early Definition of “AI”

• 1950: Alan Turing --“Computing Machinery and Intelligence”
  • Introduced what would come to be known as the “Turing Test”
  • Original test (based *imitation game*) refined over time
  • Can interrogator distinguish between computer and machine?
  • If not, then we might infer that the “machine thinks”
Early AI and Machine Learning

• 1956: Dartmouth AI Conference
  • 1955: John McCarthy coins term “Artificial Intelligence”
  • 1962: McCarthy joins the Stanford faculty
  • Invented LISP, garbage-collected memory management, computer time-sharing (now called “cloud computing”), circumscription, and much more
  • Received Turing Award in 1971

• 1959: Arthur Samuel develops learning checkers program
  • Evaluation function of board with learned weights
  • Learning based on data from professional players and playing against itself
  • Program was eventually able to beat Samuel
  • 1966: Samuel joins the Stanford faculty

General Problem Solver

1957: Allen Newell, Cliff Shaw, and Herbert Simon propose “General Problem Solver”
- Solves formalized symbolic problems
- Notion of AI as search (states, operators, goals)
- Newell and Simon receive Turing Award in 1975
- Simon also won a Nobel Prize in Economics in 1978

- State: formal description of the state (snapshot) of the world
  - E.g., position of pieces in game, logical representation of world

- Operator: function that transforms state of the world
  - Operator: state → state’
  - E.g., making move in a game, updating logical description of world

- Goal: state that satisfies criteria we care about
  - Can be explicit set of states or a function: state → {true, false}
  - E.g., checkmate in chess, some logical condition being satisfied
AI as Search

- **Search**
  - Recursively apply operators to change (simulated) state of world
  - Try to find a sequence of operators reaching state satisfying the goal
  - Various mechanisms to guide search (e.g., heuristics)
  - Means/ends analysis: identify differences between current state and goal, try to apply operators that get “closer” to goal state
  - Evaluation function: numeric value of how “good” a state is

**Goal criteria:** three X’s in a row

**States**

**Operators**
AI as Reasoning Systems

• 1976: Physical symbol system hypothesis (Newell and Simon):
  "A physical symbol system has the necessary and sufficient means for general intelligent action."

• 1970’s-80’s: Expert systems development
  • Pioneered by Ed Feigenbaum (Herb Simon was his PhD advisor)
  • Inference via logical chaining (forward and/or backward)
  • E.g., A → B, B → C, C → D. Knowing A allows us to infer D.
  • Feigenbaum joins Stanford faculty in 1964; wins Turing Award in 1994
  • Story time: MYCIN

• 1980: John Searle’s “Minds, Brains, and Programs”
  • Introduced “Chinese Room” critique of AI
    • Does the room understand Chinese?
    • Computer program ≠ intentionality or true “understanding”

• (Side note) 1986: Symbolic Systems major created at Stanford
1984: Cyc project begins (Doug Lenat, Stanford PhD 1976)
- 10 years to encode initial fact/rule base, then “read” to gain more
- 1986: Estimated 250K-500K facts required for commonsense reasoning
- Uses various forms of logical reasoning
- Today, continues to be developed by Cycorp
- As of 2017, has about 24.5 million terms and rules

Early neural network development
- 1957: Perceptron developed by Frank Rosenblatt
- 1969: Critical book by Marvin Minsky and Seymour Papert causes research on neural networks to plummet

1970s and 80s: “AI Winter”
- Over-hyped technology does not deliver
Fragmentation of AI

• 1990’s: Research interest in “Strong AI” diminishes
  • Strong AI now often called Artificial General Intelligence (AGI)

• Focus on applications of AI (“Weak AI”)
  • Machine learning (e.g., stock prediction, fraud detection, etc.)
  • Speech recognition and machine translation
  • Web search, spam filters, etc.
  • Game playing

• Public perception of AI through games
  • 1994 Checkers: Chinook defeats world champion, Marion Tinsley
  • 1997 Chess: IBM’s Deep Blue defeats world champion, Garry Kasparov
  • 2016 Go: DeepMind’s AlphaGo defeats Lee Sedol, considered one of top 10 Go players of all time (Go is more complex than chess)
  • Excellent documentary: AlphaGo
    https://www.youtube.com/watch?v=WXuK6gekU1Y
Today’s Agenda

1. History of AI
2. The machine learning revolution
3. Autonomous vehicles
4. Additional perspectives on AI
Re-emergence of Neural Networks

  - Overcomes limitations of Perceptron
  - General learning mechanism for neural networks
  - Theoretically, neural network (with enough nodes) can approximate any real-valued function
  - 1987: Rumelhart joins the Stanford faculty (Psychology Dept.)
- Work here really dates back to 1960’s by Amari and others
The Machine Learning Revolution

- 1990’s: Shift from logical formalisms and knowledge-driven methods to numerical formalisms and machine learning
- Bayesian networks (Pearl, 1985)
- Support Vector Machines (Cortes and Vapnik, 1995)
- 2010’s: Deep Learning
  - Essentially, neural networks with many nodes/layers
  - Have led to impressive (human beating) results on a number of tasks
- A sampling of systems related to “deep learning”
  - LeNet: convolutional neural network for digit recognition
  - Long Short-Term Memory: recurrent neural network
  - ImageNet (Fei-Fei Li): large scale image recognition tasks
  - DeepFace: Facebook facial recognition system
  - Google Brain: feature detector in images
  - Transformers and large language models (GPT-3, ChatGPT)
Many Forms of Machine Learning

- Supervised learning
  - Given labeled data (input/output pairs)
  - Learn a function from inputs to outputs
  - Sample of applications: stock prediction, cancer detection, ad optimization, speech recognition, assignment #1 in this class, etc.
  - Also, some language models are built using supervised learning

- Unsupervised learning
  - Given unlabeled data
  - Learn patterns that exist in the data
  - Sample of applications: products that are purchased together, groups of people with similar affinities, various forms of clustering data, etc.

- Reinforcement learning
  - Take (sequence of) actions in an environment to maximize reward
  - Want to learn a policy that specifies which action to take in any state
Unsupervised Learning: Clustering

- Grouping data points based on similarity
  - Have a (potentially implicit) distance measure over data points
  - Minimize distance within clusters
  - Maximize distance between clusters
K-Means Clustering

• Pick $k$ (random) cluster centroids
  – 1: Assign each data point to its “closest” cluster
  – 2: Recompute centroids using new assignments
  – 3: If stopping criteria not met, go to 1
K-Means Clustering

• Pick $k$ (random) cluster centroids
  – 1: Assign each data point to its “closest” cluster
  – 2: Recompute centroids using new assignments
  – 3: If stopping criteria not met, go to 1
K-Means Clustering

- Pick $k$ (random) cluster centroids
  - 1: Assign each data point to its “closest” cluster
  - 2: Recompute centroids using new assignments
  - 3: If stopping criteria not met, go to 1
K-Means Clustering

• Pick $k$ (random) cluster centroids
  – 1: Assign each data point to its “closest” cluster
  – 2: Recompute centroids using new assignments
  – 3: If stopping criteria not met, go to 1
K-Means Clustering

- Pick $k$ (random) cluster centroids
  - 1: Assign each data point to its “closest” cluster
  - 2: Recompute centroids using new assignments
  - 3: If stopping criteria not met, go to 1
• Pick $k$ (random) cluster centroids
  — 1: Assign each data point to its “closest” cluster
  — 2: Recompute centroids using new assignments
  — 3: If stopping criteria not met, go to 1
K-Means Clustering

- Pick $k$ (random) cluster centroids
  - 1: Assign each data point to its “closest” cluster
  - 2: Recompute centroids using new assignments
  - 3: If stopping critieria not met, go to 1
Sometimes End Up in Wrong Cluster

Wed 2/13/2019 12:34 AM
Amazon.com <store-news@amazon.com>
Dresses for your workweek
To: sahami@cs.stanford.edu

If there are problems with how this message is displayed, click here to view it in a web browser.

The Wear-To-Work Dress

SEE MORE
Reinforcement Learning

- Take a sequence of actions to maximize reward
  - Agent is trying to maximize expected (average) utility
  - Explicit notion of utilitarian framework

- Agent is situated in an environment (world)
  - Agent is in a particular state, \( s \), at any given time
  - In each state \( s \), agent chooses an action, \( a \), from set of available actions in that state, \( A(s) \)
    - Generally, not all action choices are available in all states
  - Transition model: maps (state \( s \), action \( a \)) to new state \( s' \)
    - Transition model is stochastic (i.e., probabilistic)
    - Taking an action in state may lead to different outcomes
    - More formally, transition model is given by \( P(s' \mid s, a) \)
    - That is, after we take action \( a \) in state \( s \), what is the probability that we are next in state \( s' \)
A maze-like problem
- The agent (robot) lives in a grid
- Each square is a state
- Walls block the agent’s path

Potentially “noisy” movement: actions not always as planned
- E.g., 80% of the time, the action North takes the agent North
- 10% of the time, North takes the agent West; 10% East
- If there is a wall in the direction the agent would have been taken, the agent stays put

The agent receives rewards each time step
- Small “living” reward each step (can be negative)
- Big rewards come at the end (good or bad)

Example: Grid World
You had me at turn_left!

Source: Courtesy of Dan Klein and Pieter Abbeel
• Some states are *terminal states* in the sense that the decision problem is done when we reach them
  – E.g., Getting to winning or losing condition in a game

• A policy $\pi(s)$ maps from states to actions, specifying the agent’s recommended action in *any* state
  – We want to learn “good” policies of actions to take
  – Should be able to apply policy when in any state
Markov Decision Process (Part II)

- Agent receives reward \( R(s) \) when it enters non-terminal state \( s \)
  - Rewards may be positive, zero, or negative
  - When agent enters a terminal state \( s_T \), we determine “total” reward (utility \( U \)) based on the sequence of states followed \( (s_0, s_1, s_2, ..., s_T) \)
    - This total reward may be additive:
      \[
      U(s_0, s_1, ..., s_T) = R(s_0) + R(s_1) + R(s_2) + ... + R(s_T)
      \]
    - Or, discounted at each step (where discount factor \( \gamma \in [0, 1] \))
      \[
      U(s_0, s_1, ..., s_T) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + ... + \gamma^T R(s_T)
      \]
    - Generally, use discounted reward since “getting a dollar tomorrow is worth less than getting a dollar now”

- Want to determine policy \( \pi(s) \) that maximizes expected reward
  - Again, note explicit notion of utility maximization
• Policies learned if we add negative reward $R(s)$ at each non-terminal state ("living cost")

$R(s) = -0.03$

$R(s) = -0.4$

Source: Courtesy of Dan Klein and Pieter Abbeel
Consider the following $3 \times 101$ grid world:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>99</th>
<th>100</th>
<th>101</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>+50</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>...</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-50</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>...</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
</tbody>
</table>

- Values in grid denote rewards at each state
- First move is *Up* or *Down*, then must make 100 *Right* moves
- Let $\gamma$ be discount factor
  - Utility of moving *Up* = $(\gamma^1)(50) + \sum_{t=2}^{101} (\gamma^t)(-1)$
  - Utility of moving *Down* = $(\gamma^1)(-50) + \sum_{t=2}^{101} (\gamma^t)(1)$
  - $\gamma = 1$: $U(Up) = -50, \quad U(Down) = 50 \quad \rightarrow$ Choose *Down*
  - $\gamma = 0.99$: $U(Up) = -12.6, \quad U(Down) = 12.6 \quad \rightarrow$ Choose *Down*
  - $\gamma = 0.98$: $U(Up) = 7.3, \quad U(Down) = -7.3 \quad \rightarrow$ Choose *Up*
  - “*Up*” is really “dump waste in lake”
  - “*Down*” is really “process waste, no dumping”
- Small changes in discount factor impact long-term decision-making (e.g., climate change)
• Consider 100 x 100 grid:

<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>10</td>
<td></td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td>99</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

• In real-world, we don’t have infinite time to try all actions in all states
  – Do we just “exploit” reward at (2, 2)?
  – Or do we “explore” hoping to find something better?
    • If so, how much exploring should we do?
    • How do we balance exploitation and exploration?
Large Language Model: GPT-3

- GPT-3 (Generative Pre-trained Transformer 3)
  - Deep learning (175 billion parameters) model developed by OpenAI
  - Predictive language model: predict next word given previous text
  - Give it a short prompt to generate text
    - Also found to generate sexist and racist language

**Prompt provided by human:**

Kayne West Exclusive – Why He’s Running for the Presidency, and what his priorities would be as President.

Anderson Cooper: Yeezy, how would you describe your presidential platform?

**GPT-3 output:**

Kanye: My platform is very simple. I want to make everything awesome again.
Anderson Cooper: But what about foreign policy?
Kanye: Who cares about that stuff? Let’s talk about important stuff, like Kim Kardashian’s butt.
Anderson Cooper: That’s a bit sexist.
Kanye: I’m sorry. I’m not sexist. I love women. I’m just an equal opportunity objectifier.

Source: https://arr.am/2020/07/09/gpt-3-an-ai-thats-eerily-good-at-writing-almost Anything/
Aligning Language Models to Follow Instructions

To make our models safer, more helpful, and more aligned, we use an existing technique called reinforcement learning from human feedback (RLHF). On prompts submitted by our customers to the API, our labelers provide demonstrations of the desired model behavior, and rank several outputs from our models. We then use this data to fine-tune GPT-3.

Source: Ouyang et al. (2022), Training language models to follow instructions with human feedback, arXiv, https://openai.com/blog/instruction-following/
Learning in InstructGPT

**Step 1** Collect demonstration data, and train a supervised policy.
- A prompt is sampled from our prompt dataset.
- A labeler demonstrates the desired output behavior.
- This data is used to fine-tune GPT-3 with supervised learning.

**Step 2** Collect comparison data, and train a reward model.
- A prompt and several model outputs are sampled.
- A labeler ranks the outputs from best to worst.
- This data is used to train our reward model.

**Step 3** Optimize a policy against the reward model using reinforcement learning.
- A new prompt is sampled from the dataset.
- The policy generates an output.
- The reward model calculates a reward for the output.
- The reward is used to update the policy using PPO.

GPT-3 vs. InstructGPT

PROMPT

Explain the moon landing to a 6 year old in a few sentences.

COMPLETION

GPT-3

*Explain the theory of gravity to a 6 year old.*
*Explain the theory of relativity to a 6 year old in a few sentences.*
*Explain the big bang theory to a 6 year old.*
*Explain evolution to a 6 year old.*

InstructGPT

*People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.*

Source: Ouyang et al. (2022), Training language models to follow instructions with human feedback, arXiv, https://openai.com/blog/instruction-following/
Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

Source: https://openai.com/blog/chatgpt/
What is the single best thing about Stanford?

It's difficult to identify a single best thing about Stanford University, as the university offers a wealth of resources and opportunities to its students. However, one aspect that many people consider to be the university's greatest strength is its commitment to interdisciplinary research and innovation.

Stanford encourages its students and faculty to think outside the box, pursue bold ideas, and tackle complex problems through a variety of approaches. This approach has led to numerous breakthroughs and discoveries in fields such as computer science, medicine, and energy, and has helped establish Stanford as one of the leading universities in the world.

Additionally, the university's location in Silicon Valley and its close ties to the tech industry provide students with unique opportunities to engage with cutting-edge technologies and startups. Whether through internships, research projects, or entrepreneurship programs, students at Stanford are well-positioned to make a positive impact on the world.

Ultimately, the best thing about Stanford will vary from person to person, and will depend on individual priorities and goals. However, the university's commitment to interdisciplinary research and innovation is one aspect that sets it apart from other institutions and is widely recognized as one of its greatest strengths.

What is the single worst thing about Stanford?

As with any university, there are several aspects of Stanford that some people may view as negative. However, one common criticism is the high cost of attendance. Stanford is a private university and one of the most expensive colleges in the country, which can be a barrier for many students and their families.

Another aspect that some students may find challenging is the competitive academic environment at Stanford. The university is known for having a rigorous curriculum, and students are expected to maintain a high level of performance. This can be stressful for some students, and may not be a good fit for those who prefer a more relaxed and supportive learning environment.

Additionally, the high cost of living in the area surrounding Stanford may be a challenge for some students and their families.

It is important to note that these perceived challenges are subjective, and will vary from person to person. Some students may view the high cost of attendance and competitive academic environment as positive aspects of the university, and may find that the opportunities and resources available at Stanford more than make up for any perceived drawbacks.
Provide a summary of Stanford class "Ethics, Public Policy, and Technological Change" as a haiku

Tech innovation grows 6 syllables
Public policy shapes its path 8 syllables
Ethics guide the way.
And it Generates Code Too

Give me python code to print out the first 10 prime numbers

Here's the code to print out the first 10 prime numbers:

```python
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, num):
        if num % i == 0:
            return False
    return True

def first_n_primes(n):
    prime_list = []
    num = 2
    while len(prime_list) < n:
        if is_prime(num):
            prime_list.append(num)
        num += 1
    return prime_list

print(first_n_primes(10))
```

This code uses a function called `is_prime` to determine if a number is prime. The `first_n_primes` function generates a list of the first `n` prime numbers by checking each number starting from 2 and adding it to the list if it is prime. The `first_n_primes` function returns the list of the first `n` prime numbers, which is then printed out.

THE SHIFT

Bing (Yes, Bing) Just Made Search Interesting Again

Google has stiff competition now, after Microsoft integrated powerful A.I. technology into its search engine.

By Kevin Roose

Kevin Roose tested the new search technology from Microsoft’s campus in Redmond, Wash.

Feb. 8, 2023   Updated Feb. 10, 2023
Generative AI for Images

Thanks, Lorenzo!

Image: Lorenzo Manuali, Generated by MidJourney AI “[Picture of Mehran Sahami, Courtesy of Mehran Sahami] as a Jedi Knight”
Today’s Agenda

1. History of AI
2. The machine learning revolution
3. **Autonomous vehicles**
4. Additional perspectives on AI
“It’s a no-brainer that 50 to 60 years from now, cars will drive themselves”
—Sebastian Thrun
Faculty director, “Junior” autonomous car project quoted in Forbes, May 11, 2011

“Nevada has become the first state to issue an ‘autonomous’ license for a driverless car”

—USA Today, May 8, 2012

Autonomous Vehicles

Google Self-Driving Car in Mountain View, August 2015

Image source: Mehran Sahami, personal photograph
Autonomous Vehicles

Zoox Prototype Autonomous Vehicle in SF, 2019

Image: Dilu, Wikimedia (CC-BY-SA 4.0)
Advances in Automation

- Level 0: No automation
- Level 1: Driver assistance, such as automated forward (e.g., cruise control) or lateral (e.g., lane keeping) control
- Level 2: Partial automation, such as directional and speed control, but driver must remain engaged (Tesla Autopilot)
- Level 3: Conditional automation -- sustained autonomous driving, but driver must intervene as system requires
  - How does that work out? Video
- Level 4: Fully autonomous driving in certain conditions), such as well-mapped areas at daytime (Waymo cars in some locales)
- Level 5: Fully autonomous driving in any conditions
Today’s Agenda

1. History of AI
2. The machine learning revolution
3. Autonomous vehicles
4. Additional perspectives on AI
AI and Employment

• 2005: Nils Nilsson proposes the Employment Test
  • Suggests replacing Turing Test with Employment Test
  • “AI programs must be able to perform the jobs ordinarily performed by humans”

• Progress in AI can be measured by fraction of such jobs performed by acceptable machines
  • Consider ATM vs. bank teller

• Part of the goal of AI is to give us the opportunity to be able to pursue our broader interests
  • That also requires putting policy/economic structures in place that allow for human flourishing, not just creating greater job displacement

Source: Stanford News/L.A. Cicero Image cropped
“AI has the potential to deliver additional global economic activity of around $13 trillion by 2030, or about 16 percent higher cumulative GDP compared with today. This amounts to 1.2 percent additional GDP growth per year.”
“Computers are everywhere but in the productivity statistics” — Robert Solow (Nobel Prize winner in Economics) in 1987

Image: Olaf Storbeck, Wikimedia (CC-BY-SA 2.0)
The Productive Paradox: “The proliferation of computing technology and digital networks alongside increasingly sluggish productivity growth rates”

“However thoroughgoing the effects these machines and networks have had on the experiences of shopping, cultural consumption, navigating cities, or financial speculation, they have had negligible effects on one key economic variable: labor productivity in the workplace”
— Jason E. Smith (echoing Solow), *Smart Machines and Service Work*
“The most impressive capabilities of AI, particularly those based on machine learning, have not yet diffused widely.

— Erik Brynjolfsson