LectureNote1: CS221 (FALL2021, Stanford)

Course Logistics

- "Modules": lectures broken into about 10-20 minutes bite-sized chunks
- "Problem Sessions": CAs work out practice problems

Prerequisites

- Programming (Python): CS106A, CS106B, CS107
- Discrete math, mathematical rigor: CS103
- Probability: CS109
- Basic Linear Algebra: Math51

Homeworks

- Introductino: foundations
- Machine Learning: Sentiment Classification
- Search: Text Reconstruction
- MDPs: Blackjack
- Games: Pac-Man (+ competition with extra credit)
- CSPs: Course Scheduling
- Bayesian Networks: Car Tracking
- Logic: Language and Logic

Course Content

- Paradigm

1. Modeling: Real world problems $\rightarrow$ Computer science concepts
2. Inference: Computer science concepts $\rightarrow$ Predictions
3. Learning: "Model **without** parameters" + "Data" $\rightarrow$ Model **with** parameters

- Machine Learning: Data $\rightarrow$ Model

- the main driver of recent successes in AI
- move from "code" to "data"
- requires a leap of faith: **generalization**

1. Reflex-Based Model
    - examples: linear classifiers, deep neural networks, etc.
    - most common models in machine learning
    - fully feed-forward (no backtracking)
    - input x $\rightarrow$ predictor $\rightarrow$ output y
    - Binary classification: x $\rightarrow$ *f* $\rightarrow$ y $\in$ {+1, -1} *label(discrete)*
    - Regression: x $\rightarrow$ *f* $\rightarrow$ y $\in \mathbb{R}$ *response(continuous)*
    - Structured prediction: x $\rightarrow$ *f* $\rightarrow$ y *complex object*

2. State-Based Model
    - examples: search problems, Markov decision processes, adversarial games, etc.
    - Applications:
        a. Games: Chess, Go, Pac-Man, Starcraft, etc.
        b. Robotics: motion planning
        c. Natural language generation: machine translation, image captioning
    - Search problems: you control everything
    - Markov decision processes: against nature (e.g., Blackjack)
    - Adversarial games: against opponent(e.g., chess)

3. Variable-Based Model
    - examples: constraint satisfaction problems, Markov networks, Bayesian networks, etc.
    - No strict ordering such as Sudoku
    - Constraint satisfaction problems: hard constraints between different variables (e.g., Sudoku, scheduling)
    - Bayesian networks: soft dependencies between variables (e.g., tracking cars from sensors)

4. Logic
    - motivation: virtual assistant
    - digest **heterogenous** information
    - reason **deeply** with that information

History of AI

- Computing Machinery and Intelligence By A.M. Turing (1950)
    - can machine think? If so, what is intelligence?
    - proposed the Turing Test to measure a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human, were instrumental
    - objective specification: the machine ought to conduct the task independent of how you get there
        a. symbolic AI
        b. teach the machine with sensors (neural, statistical AI)

- Checkers (1952): Samuel's program learned weights and played at strong amateur level

- Problem solving (1955): Newell & Simon's Logic Theorist - prove theorems in Principia Mathematica using search + heuristics; later, General Problem Slover (GPS)

- Birth of AI (1956): John McCarthy organized workshop at Darmouth College and first used the term "Artifical Intelligence"
    - every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it.

- Overwhelming Optimism
    - "Machines will be capable, within twenty years, of doign any work a man can do" - Herbert Simon

- Underwhelming Results
    - example: machine translation
    - 1966: ALPAC report cut off government funding for MT, "*first AI winter*"

- Implications of early era
    - Early AI research focused on problem-solving and symbolic methods.
    - Problems:
        a. Limited Computation: search space grew exponentially, outspacing hardware.
        b. Limited Information: complexity of AI problems (number of words, objects, concepts in the world)
    - Useful contributions (John McCarthy):
        a. Lisp
        b. Garbage Collection
        c. Time-Sharing

- Knowledge-based systems (70-80s)
    - Expert systems: elicit specific domain knowledge from experts in form of rules: if [premises] then [conclusion]
    - DENDRAL: infer molecular structure from mass spectrometry
    - MYCIN: diagnose blood infections, recommend antibiotics
    - XCON: convert customer orders into parts specification
    - Wins:
        - Knowledge helped both the information and computation gap
        - First real application that impacted industry
    - Problems:
        - Deterministic rules couldn't handle the uncertainty of the real world
        - Rules quickly became too complex to create and maintain
        - 1987: Collapse of Lisp machines and "*second AI winter*"
    - The end of the era of "symbolc AI"

- Neural AI

  - 1943: artificial neural networks, relate neural circuitry and mathematical logic (McCulloch/Pitts)
  - 1949: "cells that fire together wire together" learning rule (Hebb)
  - 1958: Perceptron algorithm for linear classifiers (Rosenblatt)
  - 1959: ADALINE device for linear regression (Widrow/Hoff)
  - 1969: Perceptrons book showed that linear models could not solve XOR, killed neural nets research (Minsky/Papert)

- Revivial of Connectionism
  - 1980: Neocognitron, a.k.a. convolutional neural networks for images (Fukushima)
  - 1986: popularization of backpropagation for training multi-layer networks (Rumelhardt, Hinton, Williams)
  - 1989: applied convolutional neural networks to recognizing handwritten digits for USPS (LeCun)

- Deep Learning
  - 2006: unsupervised layerwise pre-training of deep networks (Hinton et al.)
  - 2012: AlexNet obtains huge gains in object recognition; transformed computer vision community overnight
  - 2016: AlphaGo uses deep reinforcement learning, defeat world champion Lee Sedol in Go

- Two Intellectual Traditions
  - Symbolic AI
    - top-down approach
    - rule-based systems: uses explicitely defined rules to make inferences
    - knowledge is represented using symbols, arraged in structures like logic sentences, semantic networks, or frames
    - heavily relies on logic for reasoning and uses algorithms to process these logical statements to derive conclusions
    - expert system
  - Neural AI
    - bottom-up approach
    - composed of layers of interconnected nodes, known as artificial neurons

        - learn from data; adjust their internal parameters through a process known as training
        - uses back propagation and optimization to minimize the error and improve its performance over time
        - when neural networks have many layers, they are often referred to as "deep neural networks," and the field is known as "deep learning."
    - these two different AI's may have deeper connections (McCulloch/Pitts, AlphaGo)
        - Go is perfectly logical game designed by a few simple rules, but AlphaGo used the power of pattern matching capabilities of neural network

- Early ideas from outside AI
    - 1801: linear regression (Gauss, Legendre)
    - 1936: linear classification (Fisher)
    - 1956: Uniform cost search for shortest paths (Dijkstra)
    - 1957: Markov decision processes (Bellman)

- Statistical machine learning
    - 1985: Bayesian networks (Pearl)
    - 1995: Support vector machines (Cortes/Vapnik)
    - data-driven approach: models are trained using historical data to make predictions or analyze patterns in new, unseen data
    - supervised, unsupervised, semi-supervised and reinforcement learning
    - focus on the development of algorithms and statistical models to enable computers to learn from and make predictions or deecisions based on data

- AI technology is an implifier
    - it can reduce accessibility barriers and improve efficiency
    - it can also amplify bias, security rsiks, centralize power