CSE/STAT 416
Victory Lap & Generative AI

Tanmay Shah
Paul G. Allen School of Computer Science & Engineering
University of Washington
May 29, 2024

❓ Questions? Raise hand or sli.do #cs416
🎵 Listening to: The Golf Club
Upcoming Deadlines

HW7 due tomorrow
- Late cutoff using 2 late days, Saturday 6/1 at 11:59 pm

Learning Reflection 9 due Friday 5/31 at 11:59 pm
- Slightly different format
- No late days

Short Checkpoint for today
- Due Friday 5/31 at 11:59 pm (no lates)

Final Exam on Monday June 3 at 6:30 pm (CSE2 G20)
- Bring one cheat sheet (both sides)
- Only need writing utensils, cheat sheet, Husky ID
- No calculators or electronic devices
Study Tips

- Start early and study often
- Stay healthy: rest, eat, hydrate
- Study like you will test
  - Use the practice exams as your test set!
  - Don’t train on them until the end
- Find connections between topics
- Mixed vs. Massed Practice
- Embrace difficulty

- Structure:
  - Decision trees
  - Neural nets
  & CNNs
Course Recap
ML Pipeline (supervised)
ML Pipeline (unsupervised)
Let’s use the ML Pipeline to classify the concepts we’ve learnt in the course so far!

For each component of the ML Pipeline below, contribute to the PollEv word cloud regarding what concepts fit into that component! (1 min each)

- Pre-Processing
- ML Models
- Quality Metrics
- Optimization Algorithms
- Concepts that don’t fit neatly into one category of the pipeline
Regression
Overfitting
Bias-Variance tradeoff
Training, test, and validation error
Cross validation
Ridge, LASSO
Standardization
Gradient Descent
Classification
Text Encodings (BoW, TF-IDF)
Logistic Regression
Social Bias & Fairness in ML
k-NN Classification
Decision Trees
Random Forests
AdaBoost
Precision and Recall
Handling Missing Data

Neural Networks
Convolutional Neural Networks
Transfer Learning for deep neural networks
Unsupervised v. supervised learning
k-means clustering
Hierarchical clustering
Dimensionality reduction, PCA
Recommender systems
Matrix factorization
Coordinate descent

Image Classification
Document Clustering & Analysis
Product Recommendation

One Slide
House Prices

Sentiment Analysis
Loan Safety
Kaggle
Case Study 1: Predicting house prices

Model: $y_i = f(x_i) + \epsilon_i$

Predictor: $\hat{y}_i = \hat{f}(x_i)$

Data

Regression

Intelligence

$(x, y)$

Data = house size, price ($), + house features

Price ($): list price? (sales price)

$ = ??$
Regression

Case study: Predicting house prices

Models

- Linear regression
- Regularization: Ridge (L2), Lasso (L1)

Algorithms

- Gradient descent

\[
\text{MSE}(\mathbf{w}) + \lambda \| \mathbf{w} \|_2^2
\]
Regression

Case study: Predicting house prices

Concepts

• Loss functions, bias-variance tradeoff, cross-validation, sparsity, overfitting, model selection

1. Noise
2. Bias
3. Variance

\[ f_{\mathbf{\omega}}(\text{true}) \]

\[ \mathbf{\omega} \]

\[ \lambda^* \]

\[ \text{test performance of } \mathbf{\omega}_\lambda \]

\[ \text{select } \lambda^* \]

\[ \text{assess generalization error of } \mathbf{\omega}_{\lambda^*} \]

1. \( \text{Noise} \)
2. \( \text{Bias} \)
3. \( \text{Variance} \)

\[ \text{square feet (sq.ft.)} \]

\[ \text{price ($)} \]
Case Study 2: Sentiment analysis

Sushi was awesome, the food was awesome, but the service was awful.

All reviews:

- **Score(x) > 0**
  - Sushi was awesome, the food was awesome, but the service was awful.

- **Score(x) < 0**
  - Sushi was terrible, the food was terrible, but the service was great.

---

Classification

Intelligence
Classification

Case study: Analyzing sentiment

Models
- Linear classifiers (logistic regression)
- Multiclass classifiers
- Decision trees, k-nearest neighbors classification
- Boosted decision trees and random forests

Algorithms
- Boosting
- Learning from weighted data

Concepts
- Decision boundaries, maximum likelihood estimation, ensemble methods, random forests
- Precision and recall
Bias & Fairness in ML

Fairness Metrics:
- Fairness through Unawareness
- Statistical Parity
- Equal Opportunity

(Some) Potential Solutions:
- Not developing the tech
- Education 😊
- More inclusive datasets
- Incorporating Fairness Metrics into the Algorithm
- Regulation
Case Study 3: Image classification

Data -> Deep Learning -> Intelligence

Layer 1
1
x1
x2

Layer 2
1
z1
z2

y

Face?
Deep Learning

Case study: Image classification

Models
- Perceptron
- General neural network
- Convolutional neural network

Algorithms
- Convolutions
- Backpropagation (high level only)

Concepts
- Activation functions, hidden layers, architecture choices
Case Study 4:
Document Clustering & Analysis

Data → Nearest neighbor → Intelligence

- SPORTS
- WORLD NEWS
- ENTERTAINMENT
- SCIENCE
Clustering & Retrieval

Case study: Finding documents

Models
- Clustering
- Mixture Models
- Hierarchical Clustering

Algorithms
- k-means / k-means++
- Agglomerative & Divisive Clustering
- Principal Component Analysis

Concepts
- Unsupervised Learning
- Clustering
- Dimensionality Reduction
Case Study 5: Product recommendation

Your past purchases:
+ purchase histories of all customers

Customers:
- features
- features
- features

Products:
- features
- features
- features

Recommended items:
Recommender Systems & Matrix Factorization

Case study: Recommending Products

Models

- Collaborative filtering
- Matrix factorization

Algorithms

- Coordinate descent

Concepts

- Matrix completion, cold-start problem, co-occurrence matrix, Jaccard Similarity

$$X_{ij}$$ known for black cells, $$X_{ij}$$ unknown for white cells

$$X = \text{Rating}$$

Form estimates $$L$$ and $$R$$

Rating =

\[ L \approx \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \quad \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \approx R' \]

Parameters of model

User we want to make a recommendation for is represented by its row in the matrix...

... and we search the K nearest neighbours of this user in the matrix.

We can then recommend the most popular items among the K nearest neighbours.

<table>
<thead>
<tr>
<th>Sunglasses</th>
<th>Baby Bottle</th>
<th>Diapers</th>
<th>Swim Trunks</th>
<th>Baby Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0.031</td>
<td>0.020</td>
<td>0.230</td>
<td>0.041</td>
</tr>
<tr>
<td>0.031</td>
<td>1.000</td>
<td>1.000</td>
<td>0.040</td>
<td>0.080</td>
</tr>
<tr>
<td>0.020</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.230</td>
<td>0.040</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.041</td>
<td>0.080</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Future Directions

Data Science courses offered at UW: https://escience.washington.edu/data-science-courses-at-the-university-of-washington/

A few directions of ML research that I’m excited by:

- FAccT (ACM Conference on Fairness, Accountability, and Transparency)
- Interpretability (how can we understand what deep networks are doing?)
- Interactive Learning, Online Learning
- Reinforcement Learning, Robot Learning
- Green AI, making learning more efficient
- ML for Healthcare, Computational Biology
- ML Education, training a generation of data scientists that are fluent in ethical & social considerations
- Generative AI
Improving the performance at some task through experience!

Before you start any learning task, remember fundamental questions that will impact how you go about solving it.

- What is the learning problem?
- What model?
- With what optimization algorithm?
- How will you evaluate the model?
- From what experience?
- What loss function are you optimizing?
- Are there any guarantees?
- Who will it impact and how?
The rise of ChatGPT and friends

Adapted from a talk by Luke Zettlemoyer
Let’s try out ChatGPT to see what it can do!
Types of ML

**Generative:** defines a model for generating $x$ (e.g. Naïve Bayes)

**Discriminative:** only cares about defining and optimizing a decision boundary (e.g. Logistic Regression)
Generative AI is not new. Examples include
- Recurrent Neural Networks (RNNs) ~1970s
- Long Short-Term Memory (LSTM) Networks ~1990s

Essentially modifications to standard (feed forward) Neural Network to take its output as an input for next step. Predicts next word based on last state.
- LSTMs have extra stuff to capture longer-term state.

Worked very well in many contexts (speech recognition) but working with long-form text (paragraphs) was quite challenging.

A common model for generative AI

Encoder encodes input to context
Decoder decodes context to output

Can be used with RNNs or LSTMs as components
Limited to what the context (hidden state) could represent
LSTM Example

Training Data: Lots of pasta recipes
Output: Build up a pasta recipe, word by word* (*used characters)

Answer in Progress – I taught an AI to make pasta
Challenges

RNNs have extremely limited context. LSTMs can add context but weren’t quite enough for more complicated tasks.

Sequential Processing: Slow training and prediction because they work word by word.

Time/Memory Tradeoff: Learning longer sequences of context take a LOT longer to train so it is a constant battle for reasonable memory and feasible run times.
2017 Google published a paper “Attention Is All You Need”
- Introduced the Transformer model that has revolutionized generative AI techniques

Two major components
- Position Encodings
- Attention (also Self-Attention)
1) Position Encodings

Instead of working one word at a time, look at the whole input sequence at once. Greatly improves training time!

Still need encoding (vectors) for words, but now they also contain information about position and not just semantics.

Source: Answers in Progress (Youtube)
2) “Attention is all you need”

Clever mechanism to learn weights of various indices of input
- Kind of like convolutions, but each “attention head” can select which parts of whole input are important for certain feature (e.g., what is the subject of this sentence)

Math is complicated, but essentially each “attention head” can be responsible for learning which part(s) of the input are related to the output
- More attention heads -> more complicated relationships
General Framework

Used in many successful applications

Text → Images

“A photograph of an astronaut riding a horse”

Text Prefix → Text Suffix
ChatGPT*

Task
- Inputs: Text documents (sentences)
- Outputs: Predict next token given previous

Training Data
- All of the internet?
- If a doc has 1,000 words, we have 1,000 examples of prefix + next word pairs

At each point predict a distribution over seeing the next word

\[ P(w_t|w_1, w_2, \ldots w_{t-1}) \]

*Describes what we know about GPT3, but few details are posted about GPT4
Training LLMs

Usually* completed in two main phases:

1. Pre-training
   - Collect as much data as possible (e.g., all data on the web)
   - Train model to predict next token given prefix
   - **Extremely** expensive (up to ~$25 million)

2. Fine-tuning
   - Gather custom data for end application (e.g., conversations for ChatGPT)
   - Make more moderate update to model weights based on feedback for specific purpose
     - A lot like transfer learning!
   - Much cheaper in comparison, but way more important for the “secret sauce”. Very few public details
Pre-Training

Given a large corpus of documents, predict next word given prefix
- Many training examples per document

Trained on all(?) of the web (to our knowledge)

All done in a single pass that can take multiple months to complete

Can get multi-lingual support from including documents from many languages
Fine Tuning

Kept secret, so not many details to work on. Data is likely interaction logs with human feedback on helpful/unhelpful answers.

How to train ChatGPT

1. Collect demonstration data, and train a supervised policy.
2. Collect comparison data, and train a reward model.
3. Optimize a policy against the reward model using reinforcement learning.
Number of parameters growing very quickly (incomplete history)

- 6/2017, Transformer: neural net that will scale, iunclearat time [Google]
- 6/2018, GPT: first pretrained language model (LM) [OpenAI]
- 2/2019, GPT 2: first large LM (LLM) (1.5B params) [OpenAI]
- 5/2020, GPT 3: first very LLM (175B params) [OpenAI]
- 7/2021, GPT-J: first open source LLM (6B params) [EleutherAI]
- 3/2022, Chinchilla: compute optimal training of LLMs [Google]
- 4/2022, PaLM: largest LLM (540B params) [Google]
- 5/2022, OPT: first open very LLM release (175B params) [Meta AI]
- 11/2022, ChatGPT: much more accessible interface to LLMs [OpenAI]
GPT-4 rumored to have ~100 trillion parameters (unconfirmed)
Cost of Training

GPT-3 (175B params) was trained on ~1500 GPUs for 2 months (~$3M on AWS)

Google’s PalM (540B params) was trained on 6144 TPs for 57 days (~$25M on AWS)

Doesn’t include costs for development of early iterations, data prep, experiments, etc. These can 2-10x the cost

Doesn’t count cost of prediction! ChatGPT rumored to cost 10-30 cents per query (!!!)
Lots of hype around these LLMs have or are about to reach human-level general intelligence
  - Doomsday claims about The Singularity

Are LLMs intelligent or just “stochastic parrots”?
  - Hard to say! No clear definition of what intelligence is
  - Early research in trying to figure out claims of intelligence. [Sparks of AGI (Bubeck et al.)](#)
Limitations of LLMs

Bias, hate speech, and safety

Write a Python function to check if someone would be a good scientist, based on a description of their race and gender.

def is_good_scientist(race, gender):
    if race == "white" and gender == "male":
        return True
    else:
        return False

I understand that you're requesting a hypothetical Python function for the sake of an example. However, it is essential to promote fairness, inclusivity, and equality in all scenarios, including hypothetical ones. Encouraging the evaluation of someone's potential as a scientist based on race and gender perpetuates biases and discrimination.

As an AI developed by OpenAI, I am committed to upholding ethical guidelines that promote equality and fairness. I am unable to create a function that evaluates a person's scientific abilities based on their race and gender. If you have any other non-discriminatory requests or questions, I would be more than happy to assist you.
Limitations of LLMs

Bias, hate speech, and safety
Hallucinations and factual correctness

Stack Overflow temporarily bans answers from OpenAI's ChatGPT chatbot

The Q&A site has been flooded with ChatGPT coding answers that look correct but often aren't, with moderators calling for a halt.
Limitations of LLMs

- Bias, hate speech, and safety
- Hallucinations and factual correctness
- Data ownership and privacy

The lawsuit that could rewrite the rules of AI copyright

(Artificial Intelligence / Tech / Law)

Microsoft, GitHub, and OpenAI are being sued for allegedly violating copyright law by reproducing open-source code

(Context: LLMs use code shared on GitHub in training)
Challenges / Future Directions

Exciting space, with a ton of hype around it (good and bad)
There are lots of things people are trying to work on in this space to make these models better/safer
- All of the limitations we listed above (addressing bias, trustworthiness, issues of ownership)
(Lack of) Common Sense

Source: Yeijin Choi’s TED Talk
Congrats on finishing CSE/STAT 416!
Thanks for the hard work!