Lecture 10: February 6

ML in Industry

Agenda

- February Sprint Review
- ML in Industry

Agenda

- **February Sprint Review**
- ML in Industry

General Progress Rubric

Spring Semester

Full credit

- All tickets addressed as either “done”, “won’t do”, or moved to next sprint. Sprint includes QA testing & code review cards
- Weekly slack standup updates & participation
- Code is PRed & reviewed. Branches & PRs are well-scoped.

Partial credit

- Majority of tickets addressed as either “done”, “won’t do”, or moved to next sprint. Sprint includes QA testing & code review cards
- Occasional slack standup updates & moderate participation
- Code is committed, PRs are sometimes present and sometimes well-scoped
-

Minimal credit

- Few tickets addressed as either “done”, “won’t do”, or moved to next sprint. Sprint does not include QA testing & code review cards
- Minimal slack standup updates & rare participation
- Minimal code is committed, PRs are missing or not well-scoped.

No credit

- No trello board activity
- No slack standup updates
- No code committed

Sprint Schedule

Last Semester Sprints

October Sprint

November Sprint

Spring Semester Sprints

February Sprint

March Sprint

April Sprint (2 weeks!)

Agenda

- February Sprint Review
- **ML in Industry**

Context

1. Lots of you are exploring some form of machine learning as part of your senior design project

2. Implementing ML in a product is not easy. It's also not a skill set typically taught in class.

3. These are the problems I focus on day to day, and I’ve found success in this approach.

Goals

- Know when to apply ML to a problem
- Know how to build out an ML solution
- Understand what different ML roles in industry entail

Non-Goals

- Change your approach to senior design
- Explain how to train models
- Explain how to productionize models

Agenda

- How to tell if a problem is well-suited for an ML solution
- How to approach an ML solution
- Different roles in the ML field

What makes an idea good for ML?

1. Can the problem be uniquely solved by ML?
    a. Can a human solve this task manually?
    b. Does a rules-based approach work?
    c. What are the existing bottlenecks to solving this problem?

2. Do you have data / can you get data?

3. The IVO test: can the user **immediately** **validate** the **output**?
    a. Change the user
    b. Make validation easier
    c. Change the output format

https://cloudonair.withgoogle.com/events/startup-school-ai/watch?talk=emea-amer-talk2

Agenda

- How to tell if a problem is well-suited for an ML solution
- **How to approach an ML solution (an ML TDD)**
    - Defining the input/output
    - What is your data
    - What are the metrics
    - Establishing baselines & benchmarks
    - Model training/exploration
- Different roles in the ML field

Approaching an ML Solution: **Inputs & Outputs**

1. Identify the interface of your product user experience
2. Identify the interface of your ML model(s)
    a. What is the input?
    b. What is the output?

Why?

- Adds structure to ambiguity – can't just lean on ML for scope creep
- Engineering is easier with interfaces. ML is hard, isolating from the rest of a system is important.
- Your interfaces will dictate the data you need and the training approach you're using (regression, classification, clustering, generation)

Approaching an ML Solution: **Data**

1. Do you have data that matches your input/output interface?
2. How costly is it to collect labelled data? Are there other ways of getting “labelled” data?
3. Do you have/need unlabeled data?
4. What is your training/validation/test set?
5. What are the characteristics of your data? (amount, biases, etc)

Why?

- If you don’t have data, you’re going to have a bad time.
- Figure out early if ML is not the right approach
- Data needs can change during experimentation

Approaching an ML Solution: **Metrics**

1. What offline “correctness” metrics do you care about?
2. Are there separate online metrics that are important?
3. Are there performance metrics that impact your solution?

Why?

- Need a way to objectively measure different approaches
- Need a way to evaluate a system once in production
- Forces you to focus attention on a small number of things to optimize

Agenda

- How to tell if a problem is well-suited for an ML solution
- **How to approach an ML solution (an ML TDD)**
  - Defining the input/output
  - What is your data
  - What are the metrics
  - Establishing baselines & benchmarks
  - Model training/exploration
- Different roles in the ML field

Approaching an ML Solution: **Human Performance**

- Using the data & metrics defined previously, how does a human measure on the task?
- What is needed to collect this data?

Why?

- Ensures you can evaluate your system
- Sets a bar for performance to aim for (higher precision, higher recall, faster)

Approaching an ML Solution: **Quick Baseline**

- What is the simplest approach we can take to solve this problem? (Almost always logistic regression, xgboost, non deep learning or ml techniques)
- How does the simple approach measure up?

Why?

- Helps build out pipeline for evaluation without focusing on experimentation
- Can be used as a placeholder while building out the engineering system
- Sets a minimum bar for performance
- Identifies the gap between humans & ml

Approaching an ML Solution: **Upper Bound Baseline**

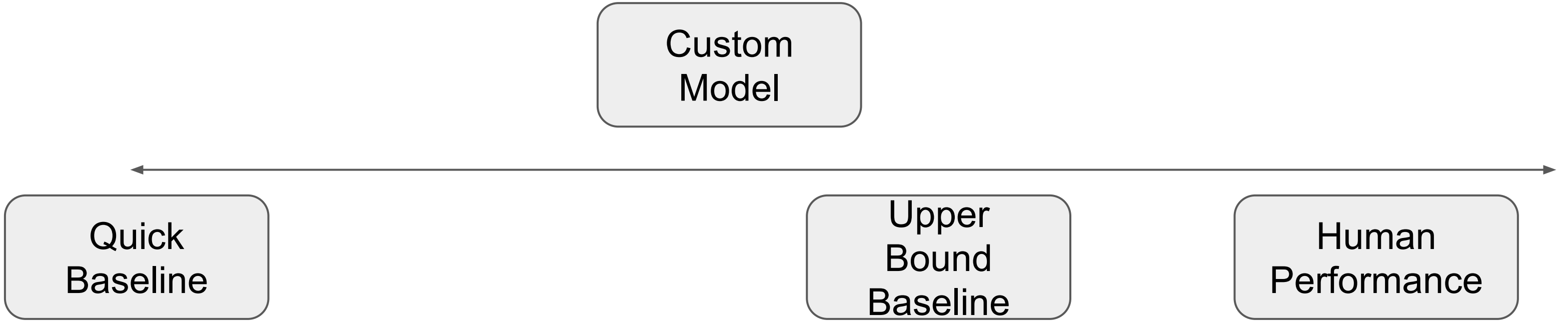
- If compute/money was no object, how would we do? (Throw an LLM at the problem)
- How does zero shot vs few shot affect results?

Why?

- Sets a pseudo-upper bound to expected ML performance
- Helps you understand tradeoffs between “accuracy” metrics & performance metrics

Approaching an ML Solution: **Experiment!**

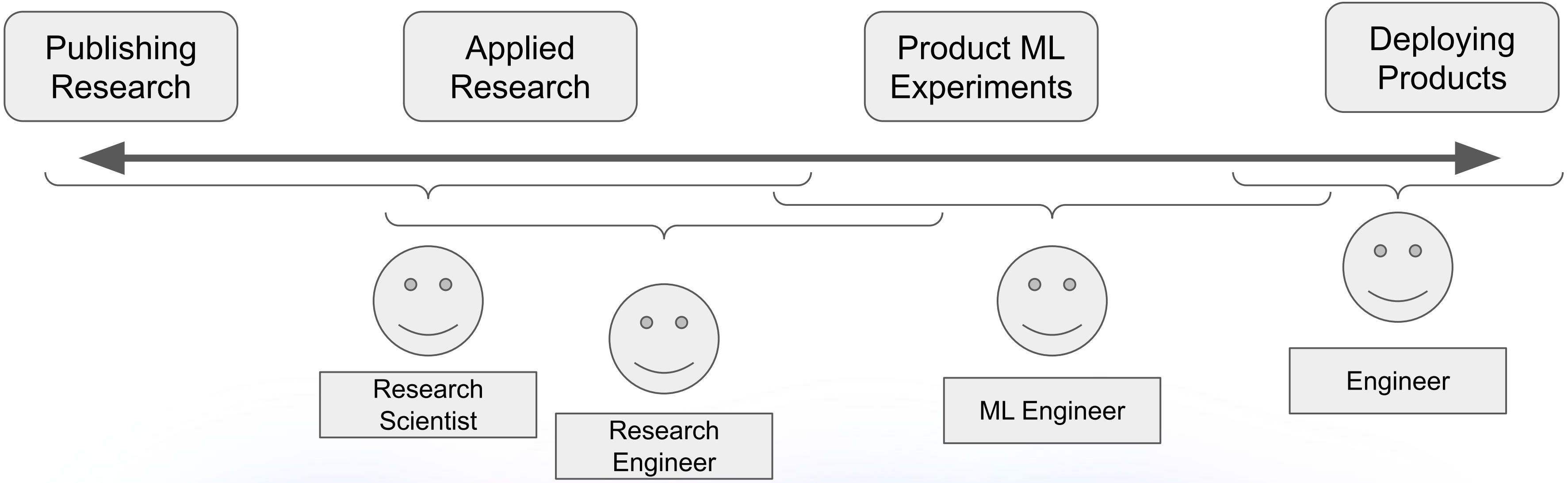
- You've done your homework, now train your own model



Agenda

- How to tell if a problem is well-suited for an ML solution
- How to approach an ML solution (an ML TDD)
    - Defining the input/output
    - What is your data
    - What are the metrics
    - Establishing baselines & benchmarks
    - Model training/exploration
- **Different roles in the ML field**

Roles in the ML field



Research Scientist

Expectations:

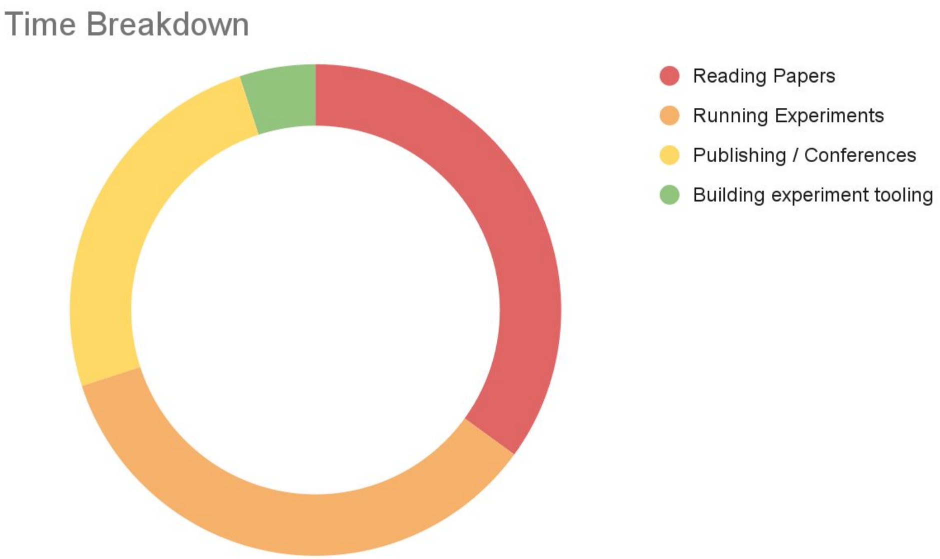
- Publish papers
- Create patents
- Novel ideas 1-2 years out

Challenges:

- Running lots of experiments & analyzing results
- Getting eng / infra help for experimentation
- Compute
- Working with teams to get data

Teams:

- Research engineers
- ML engineers
- Data science



Research Engineer

Expectations:

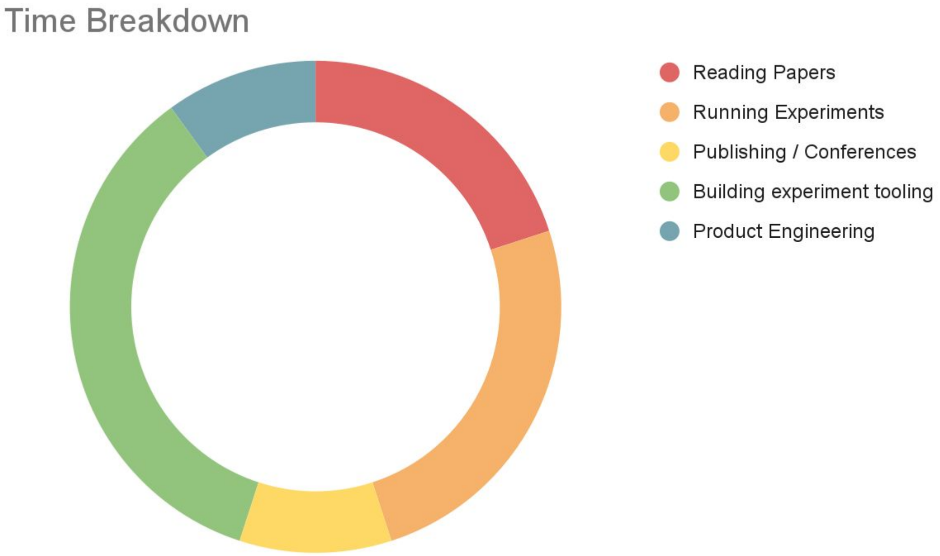
- Make experimentation easier
- Novel ideas 6-12 months out
- Publish papers/patents

Challenges:

- Build infra for research scientists
- Act as liaison between ml & research

Teams:

- Research scientists
- ML engineers
- Data science
- Product



ML Engineer

Expectations:

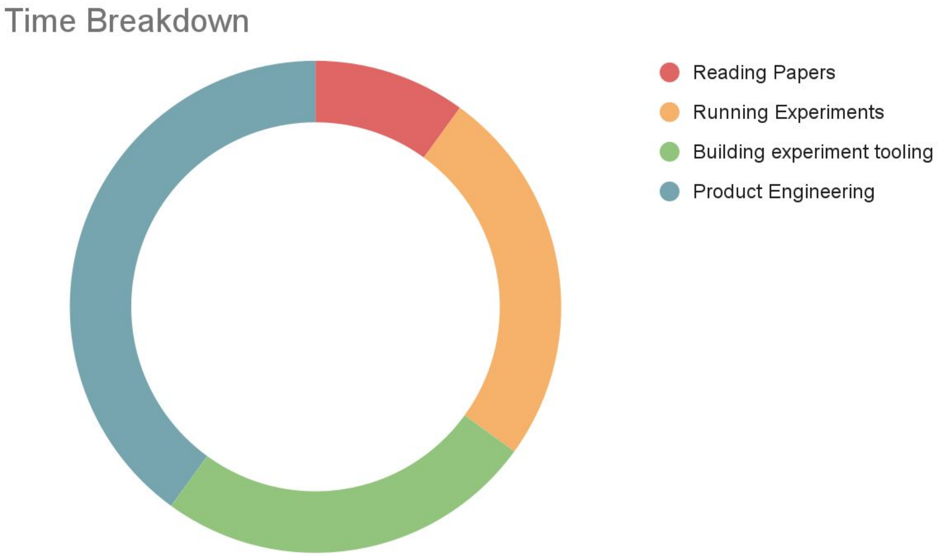
- Productionize applied research
- Build ml services
- Short-term experiments (1-2 months out)
- Monitor ml services

Challenges:

- Convert product ideas to ml problems
- Identify how to safely deploy ml models

Teams:

- Research engineers
- Data science
- Product engineers
- Product



Tools & Technologies used

Programming Languages: Python, C++, Cuda

ML Frameworks: PyTorch, Jax, sklearn

Common Libraries: Hugging Face, Pytorch Lightning, Pandas, Numpy

Experiment Tracking: Weights & Biases, MLFlow, Tensorboard

Other Technologies: Docker, Kubernetes, SQL, Airflow/Prefect

Reminders

Prepare for your March Sprint

- All tickets must be Done, Won’t Do, or moved to next sprint by the end of March
- Include tickets around PR reviews & Unit/Integration testing
- Include at least 1 ticket around user testing if you didn’t in February
- Ask your mentor to do a code review with you if you didn’t in February

Reminders

- Fill out feedback survey if you haven’t!
- Enjoy Spring Break!

Machine Learning Research

03/06/2024

Table of Contents

Introduction to Machine Learning in Academia

- Brief overview of ML's role and significance in academic research and innovation.
- Key milestones in ML academic history to showcase its evolution and impact.

Defining Machine Learning:

Briefly define machine learning as a subset of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed.

Emphasize ML's capability to uncover insights from data, leading to innovative solutions across various domains.

ML's Role in Academic Research:

Discuss how ML drives advancements in numerous academic fields, including healthcare (predictive diagnostics, personalized medicine), environmental science (climate modeling, conservation efforts), and engineering (autonomous systems, smart infrastructure).

Highlight the interdisciplinary nature of ML, fostering collaboration between computer science, statistics, biology, economics, and more.

Key Milestones in ML Academic History:

1950s: The inception of ML with Alan Turing's question, "Can machines think?", leading to the development of the Turing Test.

1967: The introduction of the "nearest neighbor" algorithm, laying foundational work for pattern recognition.

1980s: The resurgence of neural networks and the backpropagation algorithm, enabling the training of multi-layer networks.

1997: IBM's Deep Blue defeats world chess champion Garry Kasparov, demonstrating the potential of AI and ML.

2012: The success of AlexNet in the ImageNet competition, marking the beginning of the deep learning revolution in computer vision.

Present: Continuous breakthroughs in natural language processing, generative models, and reinforcement learning, pushing the boundaries of what ML can achieve.

Prerequisites and Foundational Knowledge

Outline necessary mathematical and programming skills (e.g., linear algebra, probability, Python).

Suggest prerequisite courses or resources to bridge knowledge gaps.

Core ML Courses and Advanced Electives

List core ML courses available in your department, highlighting learning outcomes.

Introduce advanced electives that complement senior design projects, such as deep learning, natural language processing, or computer vision.

My ML course Fall 2023

https://faculty.cs.gwu.edu/xiaodongqu/cs6907/

More resources

https://faculty.cs.gwu.edu/xiaodongqu/cs6907/notes/week01.html#_more_machine_learning_resources

Integrating ML into Senior Design Projects

Discuss how ML can enhance various project domains (e.g., healthcare, finance, robotics).

Showcase exemplary senior projects that successfully integrated ML, highlighting challenges and solutions.

Practical ML Tools and Frameworks

Provide an overview of essential ML libraries and frameworks (e.g., TensorFlow, PyTorch, Scikit-learn).

Offer guidance on selecting appropriate tools based on project needs.

Ethical Considerations and Responsible AI

Address ethical considerations in ML projects, such as data privacy and algorithmic bias.

Discuss the importance of responsible AI and how to implement it in design projects.

Research Methodologies in ML

Outline common research methodologies used in ML projects, including data collection, model training, and evaluation.

Offer tips on writing effective ML research proposals and reports.

Navigating ML Publications and Resources

Guide on finding and utilizing ML academic publications, preprints, and conferences for project inspiration and literature review.

Recommend databases, journals, and digital libraries for ML research.

Collaboration and Interdisciplinary Projects

Encourage interdisciplinary collaboration, highlighting how ML can be applied across different fields.

Suggest strategies for forming balanced teams with complementary skills.

Future Trends and Opportunities in ML

Discuss emerging trends in ML and potential future applications.

Highlight opportunities for further study, research, or career paths in ML.

Q&A and Project Brainstorming Session

Allocate time for students to ask questions and brainstorm project ideas with peers and faculty.