Scaled Machine Learning Conference 2017

The Scaled ML conference took place at Stanford on March 25th. The conference had a lot of big names in the ML industry, and I learned a lot about the challenges and successes with applying machine learning models in industry environments.

Ion Stoica

- New RISELab at Berkeley
    - Real-time Intelligent Secure Execution
- Focus on providing real time decisions from live data with strong security.
- **Data is only as valuable as the decisions or actions it enables.**
- Faster decisions > slow decisions
- Fresh data > stale data
- Personalized decisions > general decisions
- 10 milliseconds to make decision and 1 second to update the model
- Decisions should be
    - Intelligent: Complex decisions in uncertain environments.
    - Robust: Handle complex noise, failures, and **unforeseen inputs**.
        - Having the model say IDK if it sees something it does not know how to deal with.
    - Explainability: Ability to explain non-obvious decisions.
        - If an ML model outputs a particular medical decision, you don’t know **why** it made that particular decision.
- Explainability vs Interpretability
    - Explainability is identify what specific parts of the inputs caused a particular decision.
        - Organizing all of the input-output pairs and determining the deltas between the inputs to determine what specific part of the input caused a change in output.
    - Interpretability is determining why the algorithm had a particular output.
- One of the main issues with RL is that you have a huuuuge state space and a huge action space.
    - One approach is identifying sequence of actions called options.
        - Ex: Changing lanes while driving. The actions are use the signaller, decrease speed, check blind spot, etc.
        - The advantages are that you can generalize and you reduce the action space.
        - The current research is trying to learn the hierarchy of options automatically.

      - Problems are that the hierarchy can be so specific that it's not generalizable and the other problem would be that the hierarchy is just one single action.

Reza Zadeh

- Matroid demo was really cool

David Ku

- Based on all the data you have from all these Microsoft products, can you create new and useful experiences from
- 5 ingredients of deep learning
  - Lots of data
  - Flexible models
  - Enough compute power
  - Computationally efficient interfaces
  - Priors that defeat the curse of dimensionality.
- 5 Challenges of deep learning
  - Data policy and compliance
    - Problems come with private data like emails and problems with different country data security guidelines.
      - Solutions may come through differential privacy or stochastic privacy.
      - Design experiences that elicit and encourage feedback.
      - Data access through public datasets, employee/user donation
  - Agility to experiment
    - Need to support multiple DL frameworks
    - Support full lifecycle of ML model development to deployment
      - Version management, model iteration, visualize results, offline experimentation,
    - How to integrate all of the hardware
      - GPU clusters
  - Cost effective computing
    - DL training is expensive and time consuming
      - Have to dramatically improve the performance and cost of compute.
      - Hide the complexity of hardware acceleration from the model developers.
      - Tradeoff between performance and cost.
      - Dynamically optimize between heterogeneous compute (GPUs, CPUs, FFGA, Cloud, etc)
  - Low latency runtime

        - DL model runtime is very expensive. Anytime above 100 milliseconds is not gonna fly.
            - Model compression with reduced model precision
            - DNN Inference Acceleration using FPGA.
    - Vibrant community for collaboration and research

Matei Zaharia

- Building ML products require a huge effort in data preparation, model tuning, experimentation, and productization
    - https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf
- How do we get to a place where people with domain expertise in their specific area (biology, transportation, stocks, etc) are able to build their own production-quality ML products.
- Can we develop a machine learning platform (an end to end system that tackles the barriers to access and production use) where people are taught how to take their data and their specific problem and allow them to train the model that will help them.
- Training data is both the enabler and the barrier to entry.

Jeff Dean

- Main goal is making machines intelligent and helping people's lives.
- “We need bigger models, but sparsely activated”
    - We want capacity but we don’t need to burn the compute for every example.
- Can we do this?
    - Before: Solution = ML expertise + data +computation
    - After: Solution = data + 100x compute
- Learning to learn - RL
- Evolutionary algorithms
- There is going to be changes in how we design hardware.
    - There isn’t a need to be able to multiply .4354 times .24524 and get an exact answer.
    - In neural nets, we pretty much just need good estimates.
    - This relates to quantization of data
    - Also, we know that these units are just going to be doing common operations like convolutions and dot products and whatnot.
- For large models, model parallelism is important.
    - Getting good performance given multiple computing devices is non-trivial and non-obvious about how to distribute the computation across these devices.
    - How to split up one really huge model across different devices because you need to consider that these devices share parameters.
- Instead of simply throwing more compute (Titan X’s for days), how can we throw **smarter compute** at the problem?

Panel

- There’s always a tradeoff between whether you want to get better accuracy by trying to improve your model or trying to improve your dataset.
- How do we make systems that **don’t need** as much data to achieve the same accuracy.
- Many customers don’t have enough data or the wrong type of data to get deep learning to work well.
- Deep learning really really works well with images and RL and speech and only if you have a lot of data in those other areas.
- From software to hardware, there needs to be a good development platform for machine learning engineers to try out and iterate through different code.

Claudia Perlich

- Robustness and interpretation beats peak ML model performance in some cases.
- In some cases, you don’t need complex models. Sometimes there is just a clear fit between inputs and outputs and you can model that just with a simple logistic regression model.

Ilya Sutskever

- Evolution strategies as an alternative to RL algorithms
- RL looks to solve the problem of how do you program an agent to make intelligent decisions in a complex environment.
- Evolution Algorithm
  - Add noise to parameters
  - If result improves, keep the change.
  - Repeat
- This algorithm is surprisingly competitive with standard RL algos.
- It's not a gradient based algorithm.
  - Therefore there's no backward pass and there's no need to store activations in memory.
- However, we need 3-10x more data.
- It also parallelizes extremely well. (almost a linear speedup with an increase in the number of cores used)
  - The reasoning is that there is a lot of variance between the different workers.
- **Evolution strategies basically works really well with distributed systems and is highly parallelizable if you have a large number of cores.**
- Evolution strategies don’t work great for supervised problems, but work well for RL.
- Every trick that makes backprop better also helps evolution strategies.
  - Basically, evolution strategies try to approximate the gradient in a noisy fashion, without actually needing it to compute it.
  - You have a lot of estimations of these gradients and a lot of variance among these gradients, but since you have a lot of parallelization, you’ll get a

Wes McKinney

- He created Pandas!
    - In memory data manipulation tool