Machine Learning applied to Systems

(294-162)

Joseph E. Gonzalez
jegonzal@Berkeley.edu

Why Apply Machine Learning to Systems Problems?

- Optimal system policies or configuration may **depend on input distribution** or **future state**
- **System state** can be **difficult to model** or **partially observed**
- User's **objectives (utilities)** may be unknown but **indirectly observed**

Early Success of "Machine Learning" in Systems

- Expert systems for **hardware configuration** selection
  - XCON (late 1970s) – Rule based system to chose optimal DEC VAX configuration
- **Branch Prediction**
  - In general a "learning based" technique
  - Perceptron branch prediction AMD Chips (2012)
  - Downsides/issues?
- **Learned cost models** for query planning(early 2000s)
- **Packet Classification** without deep inspection (early 2000s)

Early Issues Applying ML to Systems

- Early ML techniques were **brittle** and **difficult to tune**
  - **Still are?**
- Difficult to reason about **"failure-modes"**
- Heavy **computational costs** associated with ML

Simple heuristics often "good enough"

Recent resurgence of interest in ML for Systems

- **Large-scale systems** have elevated the need for learning based approaches
- **Recent progress in deep learning** and its applications to "hard problems" has generated renewed interests
- Several **recent efforts** have demonstrated potential
  - From this week's readings highlights 3 such papers

The Case for Learned Index Structures

- Explores the **idea of leveraging "over-fitting"** to replace memory intensive data structures with **compute intensive models**.
- Generated a lot of interests when first published
  - Hacker News, Stanford Response
- Big issue -- **updates** -- is addressed in follow-up paper: ALEX: An Updatable Adaptive Learned Index

Device Placement Optimization with Reinforcement Learning

- High profile project at Google that generated a lot of interest in **RL applied to systems problems**
- Precursor to more recent **high-profile chip design work** by the same group

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature › articles › article

Article | Published: 09 June 2021

A graph placement methodology for fast chip design

Azalia Mirhoseini ✉, Anna Goldie ✉, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter & Jeff Dean

MIT Technology Review Innovators Under 35

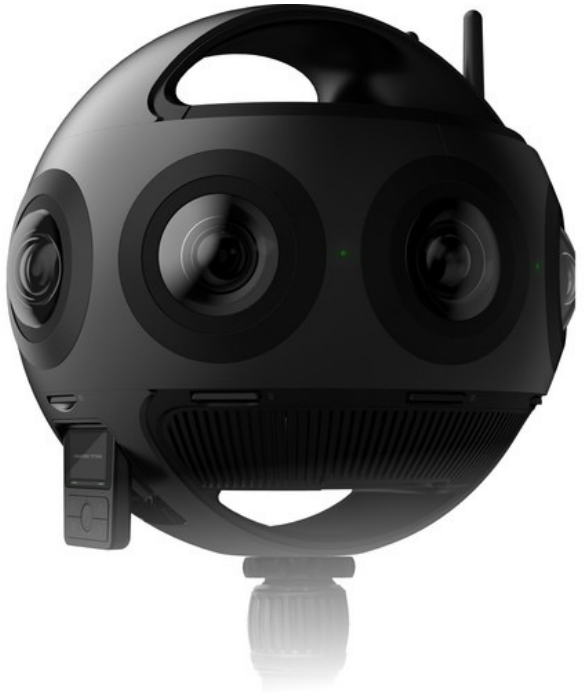
Azalia Mirhoseini

AGE: 32
AFFILIATION: GOOGLE BRAIN
COUNTRY OF BIRTH: IRAN

She taught an AI to design A chips

Neural Adaptive Video Streaming with Pensieve

- **Widely cited paper** applying RL techniques to address adaptive **quality selection of streaming video**.
- Addresses trends in earlier work that focused heavily on
  - Throughput modeling
  - Model predictive control
- Future Opportunities: this addresses an area of likely increased interests
  - AR/VR Video Playback



Things to Ask when Applying ML to Systems

Before you start:

- Is there **"structure"** in the problem being solved?
  - Can an "expert" given enough time and experience with the system "solve the problem"?
- Is the **problem input dependent**?
  - Are there patterns in the input that can be modeled.

Once you succeed, you should ask:

- **What is being learned** and in what way does your **technique generalize**?
  - Did you just run weeks of random search to find a model that finds a good solution to a single problem. (overfitting?)

Some of My Experience with ML Applied to Systems

- Wireless Link Quality Estimation using GP Models: **Failed**
  - **Hope:** Learn how radio waves propagate through environment using only pair-wise observation
  - **Problem:** insufficient learnable structure
    - Baseline distance model reasonable strong
    - Deviation from baseline distance model is governed by complex interference that changes over cm distances.
- VM Selection for Workloads: **Success**
  - **Hope:** Knowledge of the details of a workload and VM characteristics should determine performance
  - **Idea:** Similar workloads should perform similarly across different VM Types
  - **Solutions:** Collaborative filtering, modeling workload characteristics as a function of VM performance profiles.