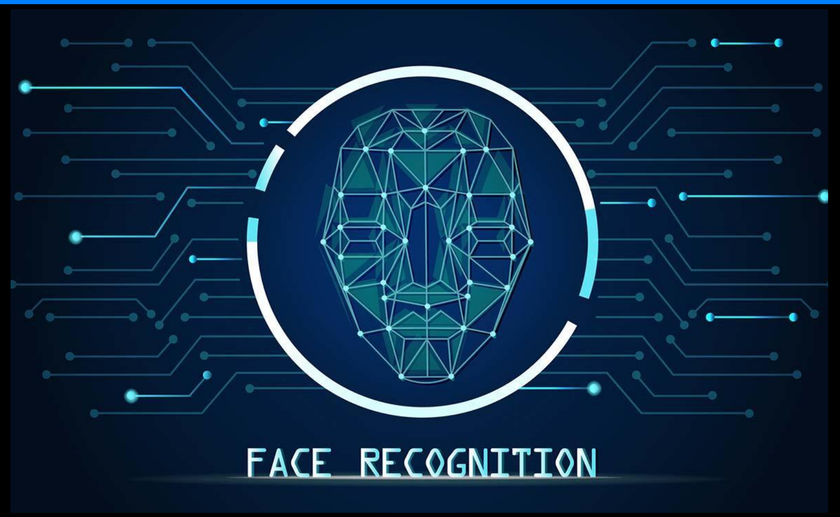
Lecture 1: Introduction

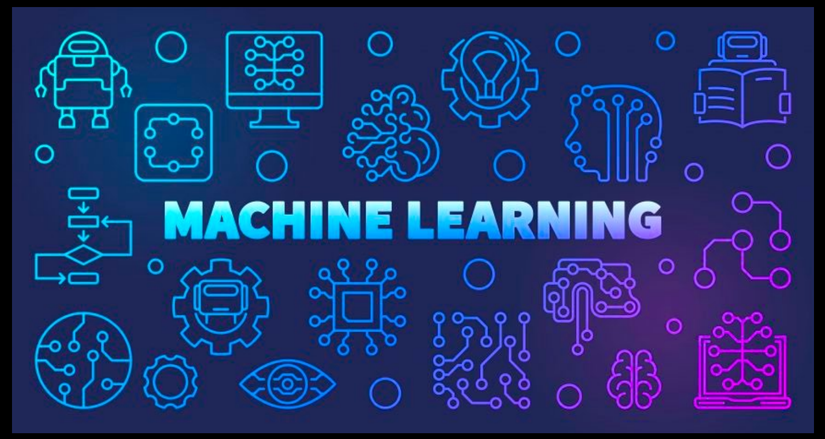
CS 256: Systems and Machine Learning

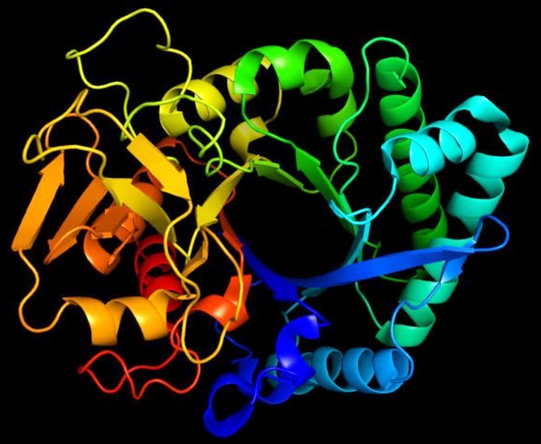
Sangeetha Abdu Jyothi

UCIRVINE

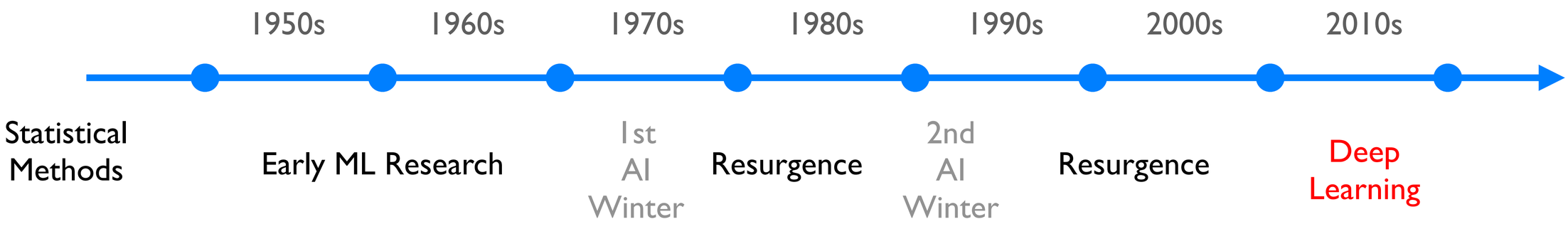
Machine Learning is Transforming Our World







Relevance of Systems in the ML Revolution?



- Machine Learning widely adopted only recently, although research began more than half a century ago.
- Today's rapid growth fueled by two key factors
  - Large amounts of data
  - Advancements in underlying hardware and software systems

This Course

- Part 1 : Systems for Machine Learning
  - Systems advancements that enabled the ML revolution
  - Practical challenges in deploying ML systems

- Part 2 : Machine Learning for Systems
  - Machine Learning Applications in Computer Systems
  - Challenges in Real-World Learning

Today's Lecture

- Class Logistics
- Overview of Key Topics

Course Info

- Lectures: MW 5:00-6:20 pm PT at PCB 1200
- Course website: https://canvas.eee.uci.edu/courses/43833
- All lectures are live-streamed on Zoom (https://uci.zoom.us/j/92825560246?pwd=TW9mQjVjV01xK3VQWDc3SVZrU0pYQT09) and recordings will be posted on Canvas
- Office Hours: Wed 1-2 pm on Zoom (https://uci.zoom.us/j/92149956397?pwd=RStDV2J1MmpNR0RJc0xabEo2bFRmdz09) or in person after class (MW 6:20-6:50 pm)
- TA: Sagar Patel
- TA Office Hours: Mon 1-2 pm
- Ed for discussions and all communication with instructor (please use private threads for 1:1 messages)

Paper Reviews

- In each class, you are assigned one paper for review
- Advanced papers, but do not panic!
- Paper reviews are due at 12 pm on the day of the class
- Paper reviews will have three parts
    - A short summary (please keep it short! 5-10 lines)
    - 2-5 points on strengths of the paper
    - 2-5 points which may be your concerns about solutions presented in the paper, your ideas for future work building on the concepts in the paper, or potential applications of this paper in other systems/ML problems
- Evaluation not based on the length of the summary or the number of points listed. 2 is sufficient.

Structure of the Class

- Students will read one paper on the topic of discussion
- The lecture will include a short discussion on the paper you read and cover a broad range of research beyond the paper

Course Project

- Groups of 1-3 students
- 4 submissions: initial plan, checkpoint 1, checkpoint 2, and final
    - Title and Plan
    - Checkpoint 1
    - Checkpoint 2
    - Final Report (5-6 pages)
- PhD/MS thesis students are welcome to choose a topic related to their primary research
- ALL groups must start an online conversation with the instructor before the title deadline to discuss plans for your project (this can be used for brainstorming, but main purpose is confirming that your project is a good fit).

Project Details

- Title and Plan Due: Oct 10 [5% of final grade]
  - Title, Group members, Project Abstract (1 paragraph), Project timeline
- Checkpoint 1. Due: Oct 31 [10%]
  - 1-2 pages of report (Introduction and related work) and 5-min recorded presentation
- Checkpoint 2
  - 3-4 pages of report (Ckpt 1 + Design, Initial evaluation) Due: Nov 21 [10%]
  - In-class presentations Nov 23-30 [15%]
- Final Report Due: Dec 9 [10%]

Grading

- Paper Summaries: 30% (Best 12 out of 15)
- Class participation: 20%
- Final Project: 50%
    - Title and plan: 5%
    - Checkpoint 1: 10%
    - Checkpoint 2 Report: 10%
    - In-class presentation: 15%
    - Final report: 10%

Today's Lecture

- Class Logistics
- Overview of Key Topics

Systems for ML

Deep Learning Systems Stack

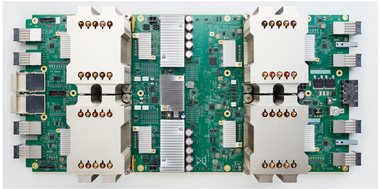
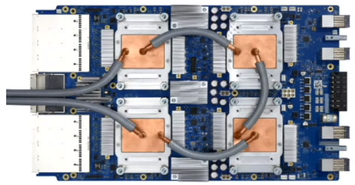
Deep Learning Frameworks

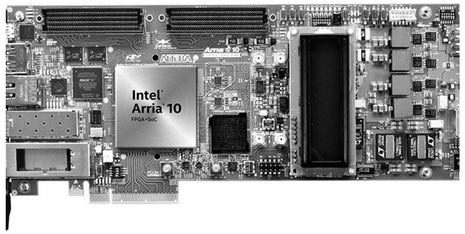
TensorFlow PyTorch mxnet Microsoft Cognitive Toolkit Caffe2

Deep Learning Compilers

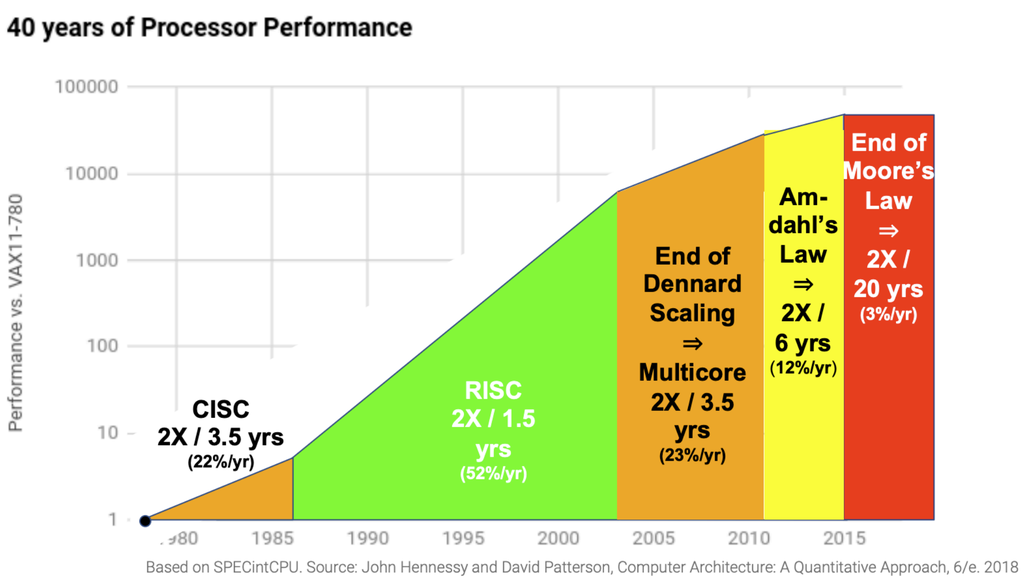
tvm nGraph GLOW XLA MLIR

Hardware



Domain Specific Architectures



Slowing of Moore's Law

+

A compute- and memory-intensive workload

with easy-to-optimize building blocks

Solution: Domain-Specific Architectures like TPU

Deep Learning Frameworks

- Key Goals
  - Simplify the implementation of Deep Learning models
  - Enable scalability

PyTorch

Keras

TensorFlow

Chainer

mxnet

Deep Learning Compilers

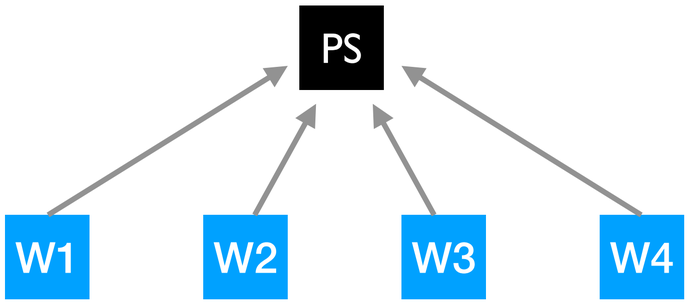
- Goal: Map computation to heterogenous DL hardware efficiently without manual tuning
  - Input: Model Definitions from DL Frameworks
  - Output: Efficient code implementation on DL hardware

tvm

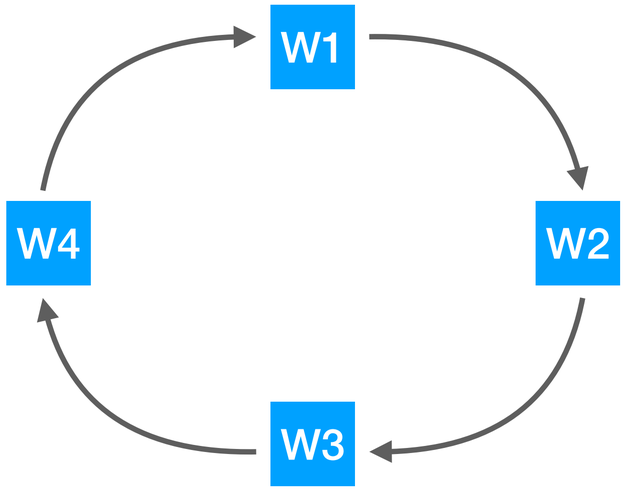
Tensor Comprehensions

Networking Challenges in Distributed Training

- DNN training is iterative with parameter aggregation after each iteration
- Large amounts (GBs) of data exchanged after each iteration
- Two popular modes of aggregation
- Interdependencies of communication with computation and the amount of data to be aggregated lead to several challenges



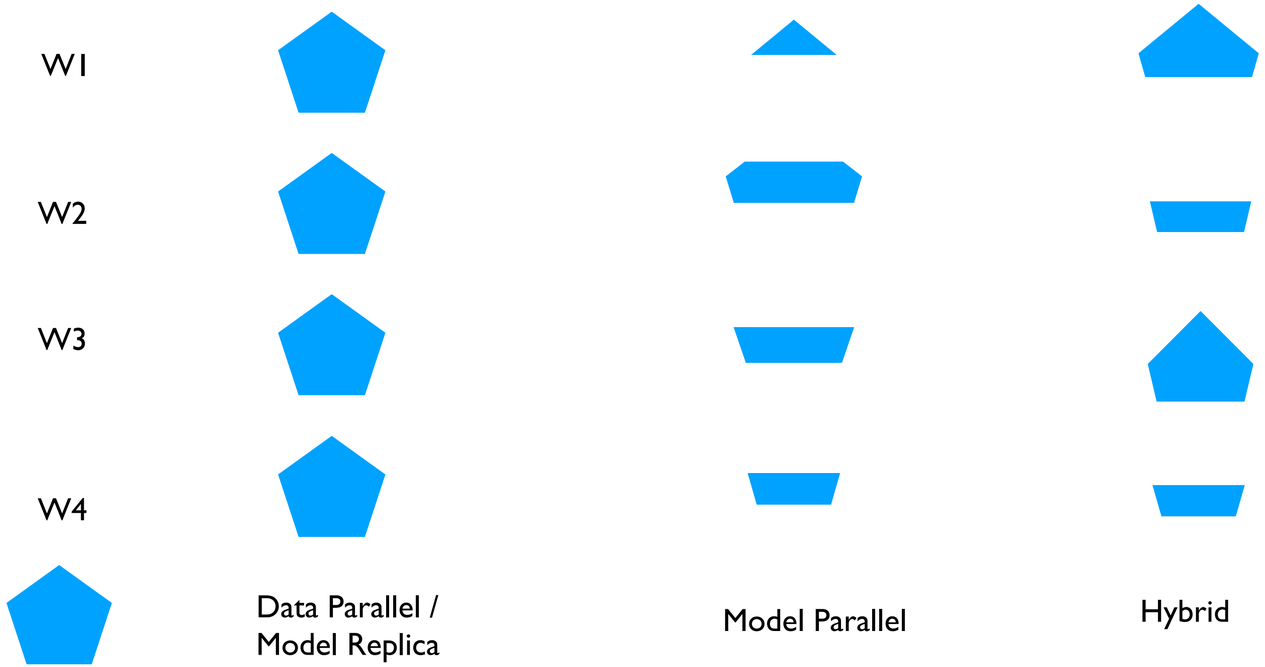
Parameter Server



Decentralized Aggregation

Cluster Scheduling for Deep Learning Workloads

- Challenges in scheduling
  - A deep learning cluster may have heterogeneous nodes with GPUs from different generations
  - A DL workload may be parallelized in multiple ways



Automated ML: Hyperparameter Tuning and NAS

- Hyperparameter Tuning
  - Models have a large number of tunable parameters that are difficult to set manually
  - Frameworks for automating hyperparameter optimization with provable guarantees

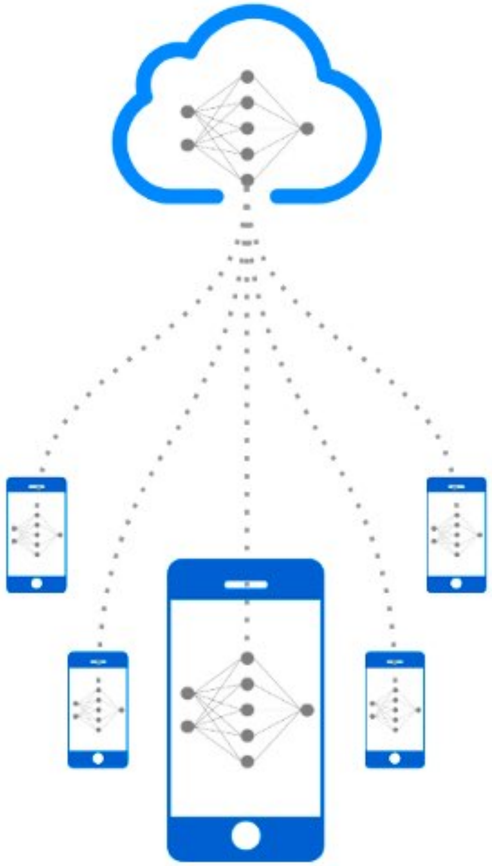
- Neural Architectural Search (NAS)
  - Automating the design of CNN architectures

Scaling Gradient Boosting and Reinforcement Learning

- DNN-based supervised training is popular, but not the only ML technique
- Solutions for optimizing other ML techniques
  - XGBoost: A tree-boosting system
  - Asynchronous variants of RL algorithms and RLlib library

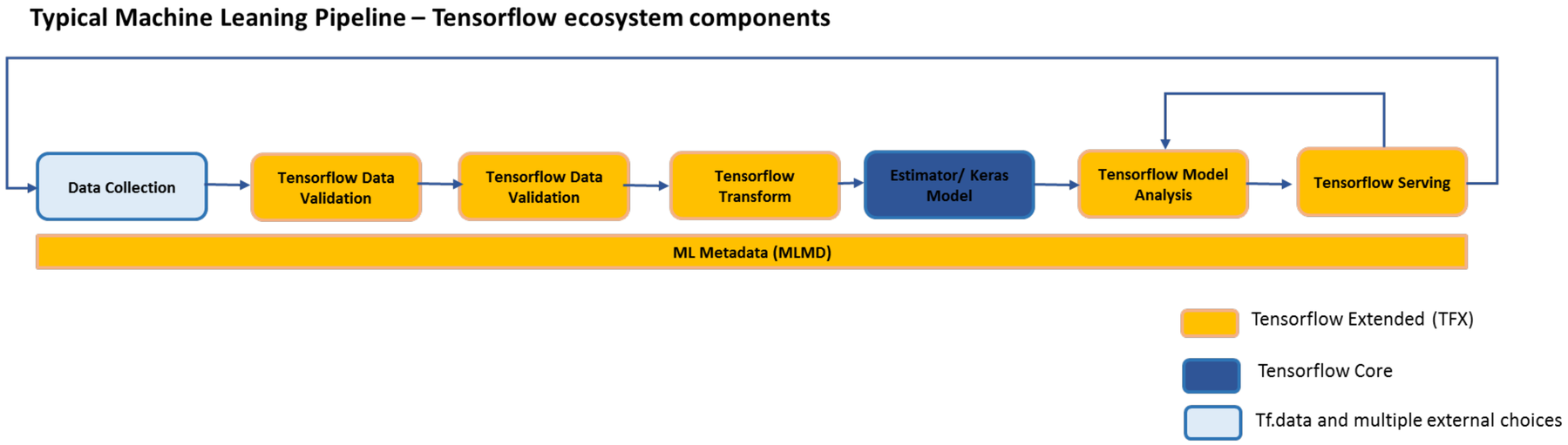
ML at the Edge

- Challenges
  - Diverse set of distributed devices
  - Privacy requirements on data
- Federated Learning
  - Collaboratively learn a model while keeping all the training data local



Challenges in Operational ML Systems

- Understanding ML-specific risk factors in systems design
- A study of real-world ML pipelines



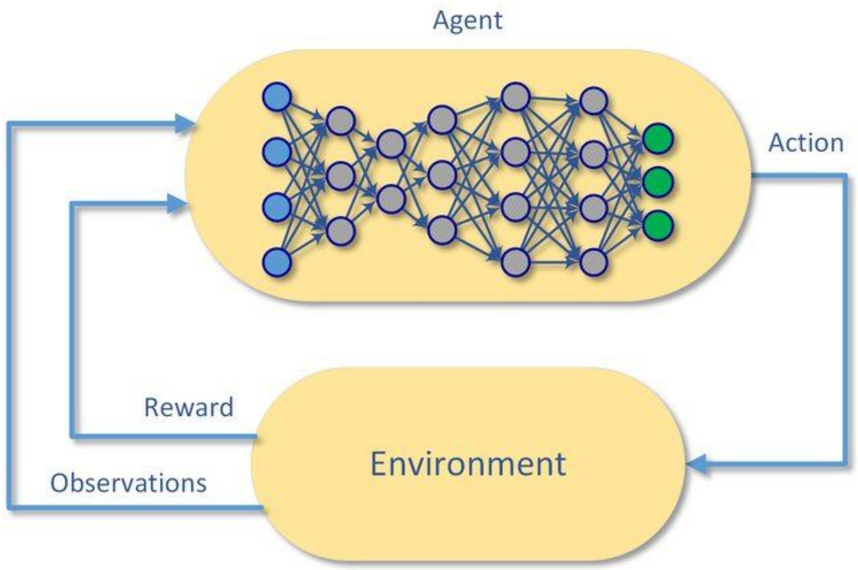
Systems for ML Summary

- Deep Learning Architectures
- Deep Learning Platforms
- Deep Learning Compilers
- Network Scheduling for Learning
- Cluster Scheduling for Learning
- Automated ML: Hyperparameter Tuning and Neural Architectural Search (NAS)
- Scaling Gradient Boosting and RL
- ML at the Edge
- Challenges in Operational ML Systems

ML for Systems

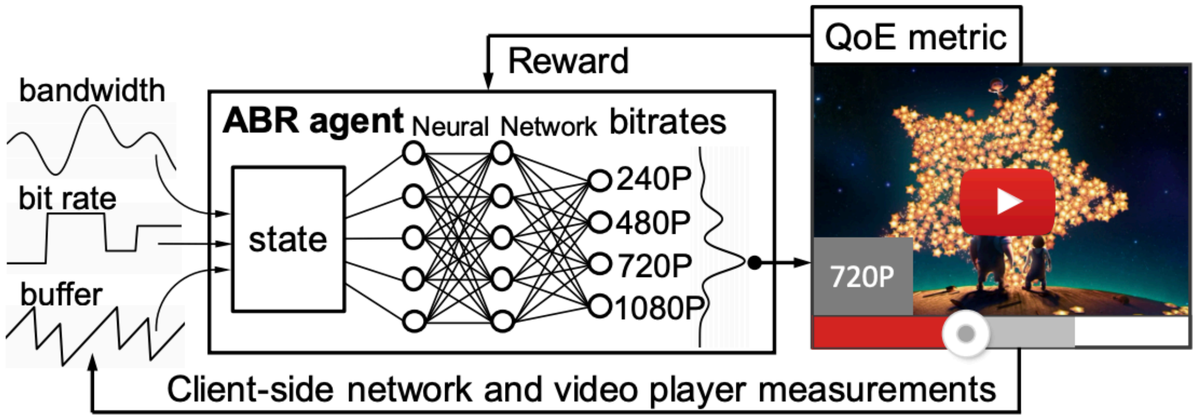
Deep Reinforcement Learning

Fundamentals of Reinforcement Learning and relevance to system control



Deep RL for various systems applications

- Internet Congestion Control
- Adaptive Bitrate Selection in video streaming
- Scheduling



Other ML techniques in systems context

- Learning index structures in Database Management Systems
- Learning cache replacement algorithms in Content Delivery Networks
- Flow prediction in data centers

Classic control + Learning

- Clean-slate learning-based techniques have several practical issues and concerns
- In practical settings, this can lead to poor performance
- Pragmatic and evolutionary approaches combining classic control with learning techniques

Interpretability of RL-based Controllers

- DL-based controllers offer better performance
- But…
  - No guarantees on how they will behave in various scenarios
  - Uninterpretable blackboxes
- Interpretability and verification of deep RL-based controllers

How to Read a Paper

How to Read a Paper?

- https://dl.acm.org/doi/pdf/10.1145/1273445.1273458 (A nice article by S. Keshav)
- Three-pass approach
  - First pass: High-level (understand the category, context, key contributions, etc.)
  - Second pass: Read the paper more thoroughly but ignore fine-grained details such as proofs
  - Third pass: Thoroughly understand every detail
- This class expectation:
  - First and second passes to get a general understanding of the topic and solution

Sample Project Topics from Past Offering

Past Project Topics

- Reinforcement Learning with Auxiliary Tasks for Congestion Control
- Optimizing Distributed Reinforcement Learning
- Resource-Constrained Adaptive Federated Learning
- Online Learning for Distributed Training on Heterogeneous Systems
- Profiling data prefetching techniques for common machine learning workloads
- Learning to Adapt Sequential Scan Operations

Thanks!