Ekya: Continuous Learning of Video Analytics Models on Edge Compute Servers

Romil Bhardwaj, Zhengxu Xia, Ganesh Ananthanarayanan, Junchen Jiang, Nikolaos Karianakis, Yuanchao Shu, Kevin Hsieh, Victor Bahl, Ion Stoica
romil.bhardwaj@berkeley.edu; http://aka.ms/ekya

Video Analytics at the Edge

Why edge analytics?
- Privacy
- No network required

Data Drift and Continuous Learning

Compressed Models don't generalize and are sensitive to data drift

Data Drift Example – Class Definition Shifts

The performance of pretrained models varies as the incoming data distribution and class definitions change

Continuous Learning addresses data but steals GPU-time from inference and has variable costs

Data Window

Resource Allocation over time

Accuracy over time

Joint Hyperparameter Selection and Resource Scheduling – an NP-hard problem

Pick Hyperparameters to retrain

Allocate resources between Training and Inference across streams

Scheduler Objective
Maximize mean inference accuracy across all video streams

Constraints
Resource capacity constraints (do not oversubscribe)
Minimum inference accuracy constraint (ensure min. accuracy)

Ekya at a Glance

Thief Scheduler
- Resource allocation
- Config selection

Inference Executor

Retraining Executor

Resource + Config Allocations

Observed Training Profile

Resource- Accuracy Estimates

Microprofiler
- Accuracy estimation
- Resource estimation

Configurations & Profiles

Resources diverted to retraining

Cost of retraining depends on retraining hyperparameters

Evaluation and (new!) Datasets

- We release and evaluate Ekya on two new datasets and on two public datasets – Waymo and Cityscapes
- Ekya saves upto 4.3x resources and achieves 29% higher accuracy than baselines

Comparing accuracy with varying number of video streams

Ekya produces accuracy and resource demand estimates, informing the thief scheduler’s heuristics to maximize accuracy

Resources are oversubscribed, Ekya prioritizes inference jobs