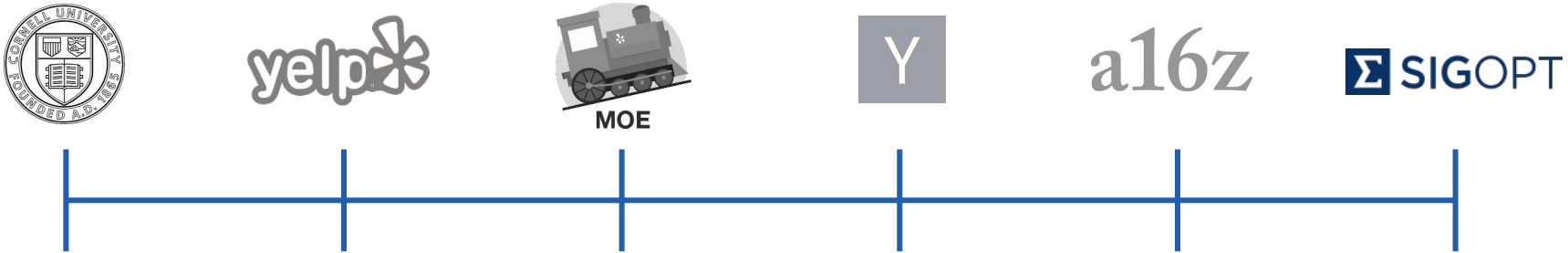
Tuning the Untunable

Techniques for Accelerating Deep Learning Optimization

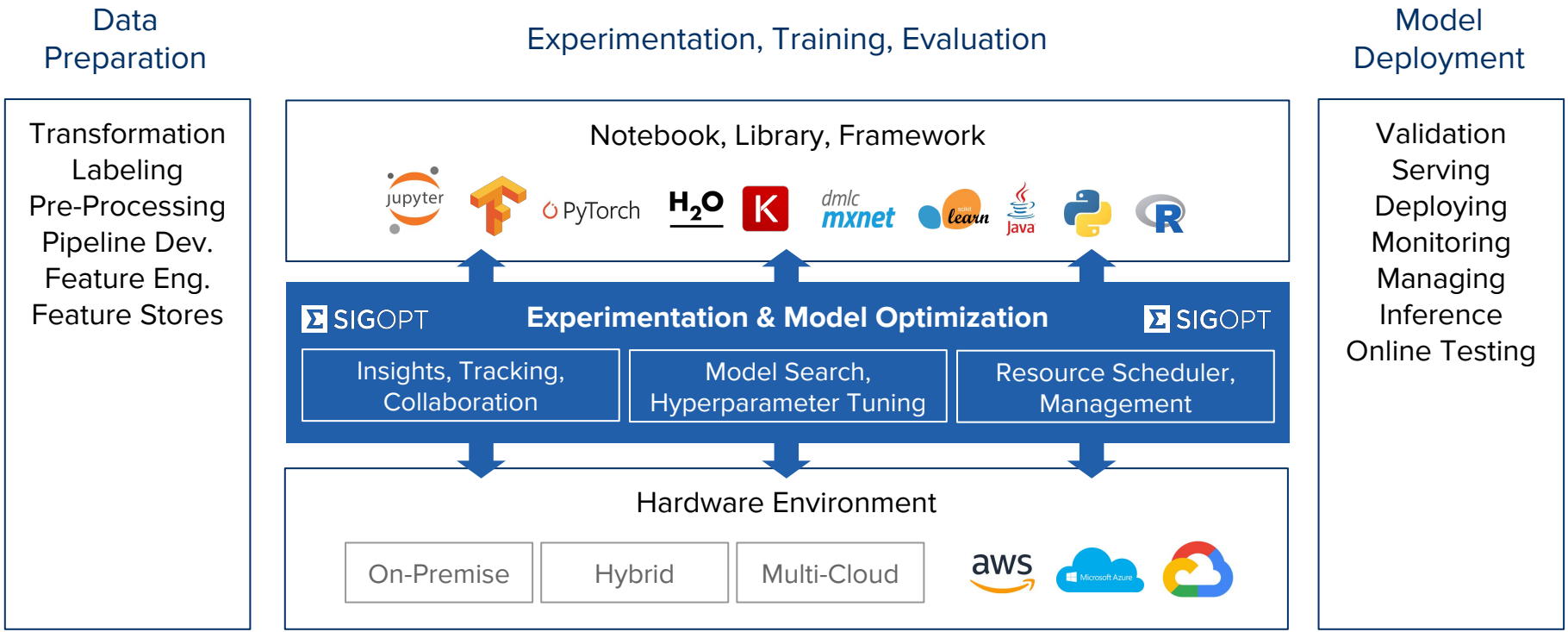
Talk ID: S9313

SIGOPT

How I got here: 10+ years of tuning models

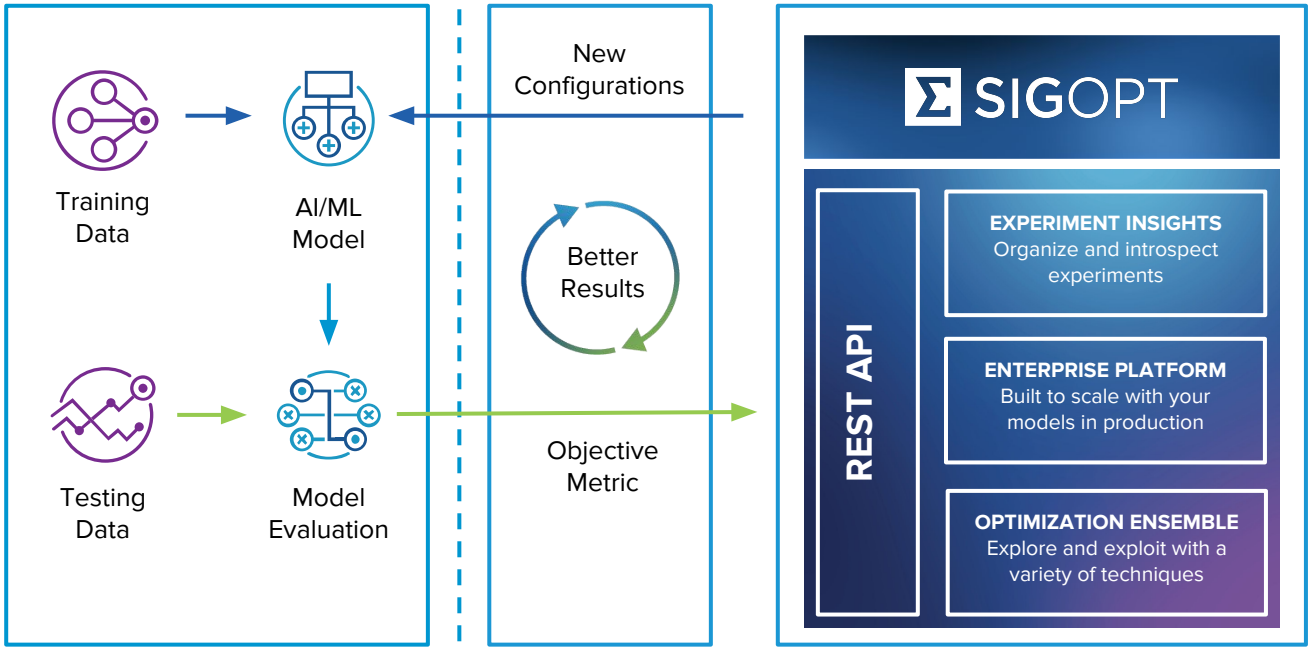


SigOpt is a experimentation and optimization platform



Experimentation drives to better results

Iterative, automated optimization

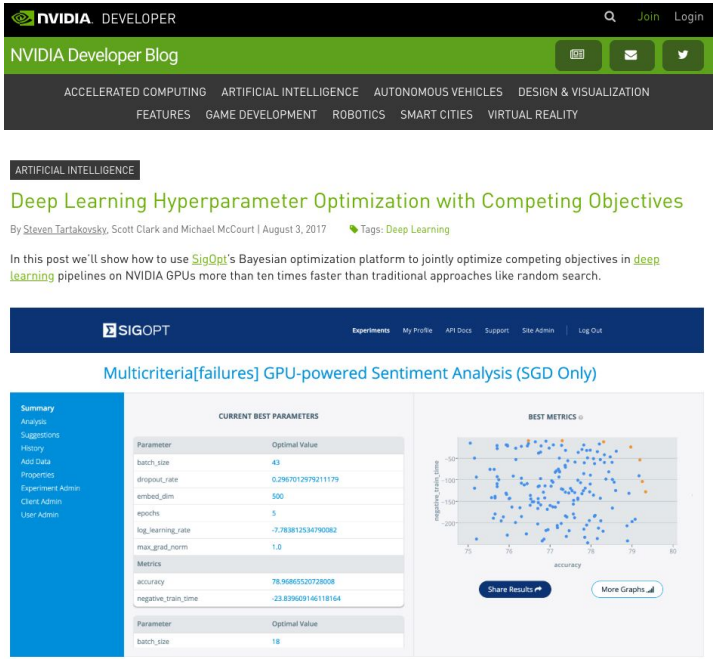


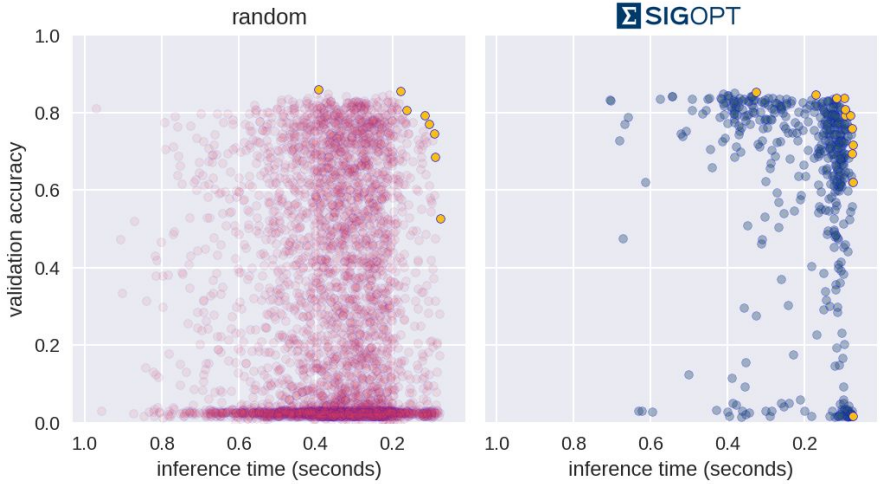
Data and models stay private

Built specifically for scalable enterprise use cases

Previous Work: Tuning CNNs for Competing Objectives

Takeaway: Real world problems have trade-offs, proper tuning maximizes impact

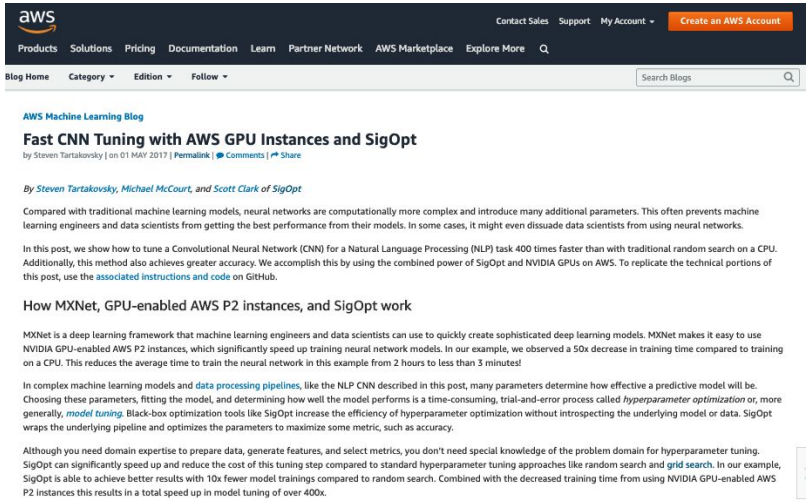


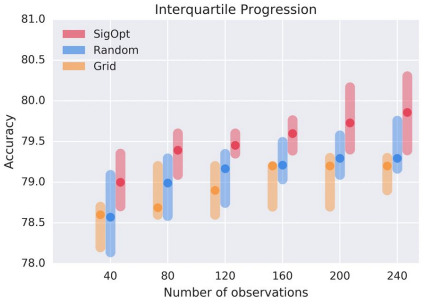


https://devblogs.nvidia.com/sigopt-deep-learning-hyperparameter-optimization/

Previous Work: Tuning Survey on NLP CNNs

Takeaway: Hardware speedups and tuning efficiency speedups are multiplicative





mxnet

NVIDIA & SIGOPT

Provides a 400x speed up

https://aws.amazon.com/blogs/machine-learning/fast-cnn-tuning-with-aws-gpu-instances-and-sigopt/

Previous Work: Tuning MemN2N for QA Systems

Takeaway: Tuning impact grows for models with complex, dependent parameter spaces

NVIDIA DEVELOPER Join Login

NVIDIA Developer Blog

ACCELERATED COMPUTING ARTIFICIAL INTELLIGENCE AUTONOMOUS VEHICLES DESIGN & VISUALIZATION
FEATURES GAME DEVELOPMENT ROBOTICS SMART CITIES VIRTUAL REALITY

ARTIFICIAL INTELLIGENCE ACCELERATED COMPUTING

Optimizing End-to-End Memory Networks Using SigOpt and GPUs

By Meghana Ravikumar, Nick Payton, Ben Hsu and Scott Clark | February 21, 2019

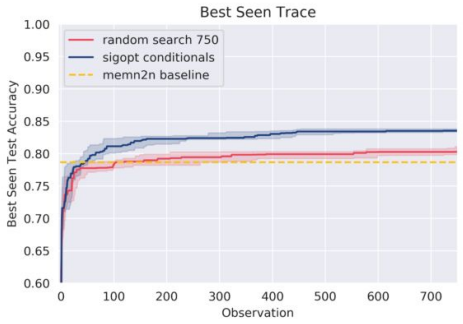
Tags: Accelerated Computing, Bayesian optimization, hyperparameters, machine learning and AI, MemN2N, MemNN, memory networks, question answering, SigOpt, Speech

Natural language systems have become the go-between for humans and AI-assisted digital services. Digital assistants, chatbots, and automated HR systems all rely on understanding language, working in the space of question answering. So what are question answering (QA) systems and why do they matter? In general, QA systems take some sort of context in the form of natural language and retrieve, summarize, or generate information based on this context. Maximizing the performance of these systems is critical if they are to be useful. QA performance is the difference between Amazon Alexa becoming a new member of the family or yet another device relegated to the back of your closet.

Research into Question Answering remains very active due to the popularity of voice-capable systems. New models and algorithms regularly out-perform each other on benchmarking tasks like the popular SQUAD challenge. Exciting areas of research include training more accurate and contextually diverse word embeddings (BERT and ELMO), evaluating variations in RNN+CNN based architectures (BiDAF, QANet), and comparing the performance of memory networks (MemN2N, Dynamic Memory Networks). Each of these approaches exemplifies practitioners using their expertise to engineer performance gains in the QA system.

Hyperparameter tuning represents a well-researched method complementary to these expert-driven approaches to further boost model performance. Although tuning typically improves performance relative to a baseline, the proportion of its improvement often differs with particular circumstances of the data, model and task. This nuance leads us to the first question for QA: where can hyperparameter tuning drive a significant improvement in performance for these expert-driven processes?

Equally important in the context of QA systems is the practicality of implementing different hyperparameter tuning methods. Each applied machine learning setting comes with different constraints in time, resources, and experts. We should consider whether these constraints point toward one method or the other for tuning. In short, if you construe performance as both accuracy and speed, will certain tuning methods be more capable for QA systems?



All 20 Tasks						
Optimization Method	Evaluations	Average Accuracy	Average Improvement (base = 78.67%)	P-value (Sigopt Conditionals vs Random Variant)	Cost per % improvement (GPU)	Best SGD Variant
Random	750	80.28%	1.61%	9.16e-08	$23.06	GradientDescentMomentum
Random	15000	82.73%	4.06%	3.30e-06	$182.88	GradientDescentMomentum
SigOpt Conditionals	750	83.51 %	4.84%	-	$9.85	GradientDescentMomentum

Table 5: Performance breakdown across optimization methods for all 20 tasks. GPU cost =$0.90/hr used for Cost/% Improvement

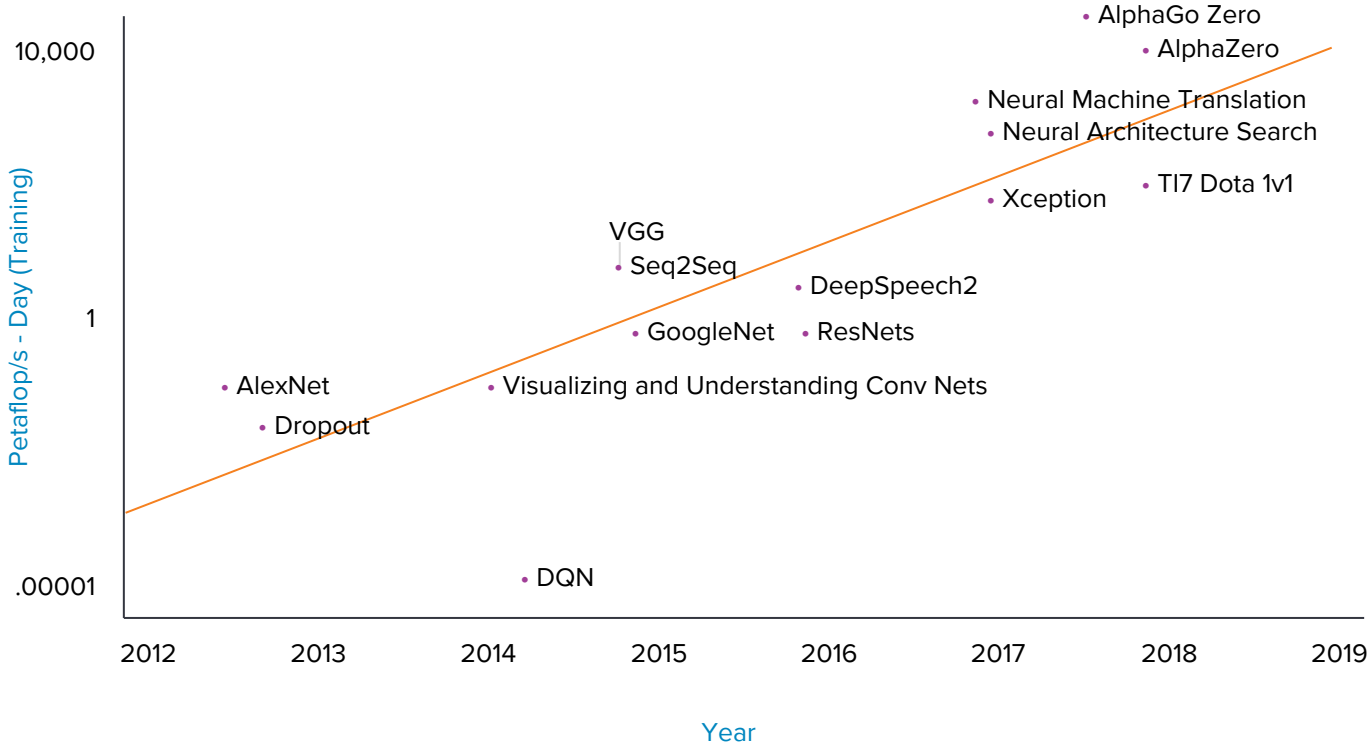
https://devblogs.nvidia.com/optimizing-end-to-end-memory-networks-using-sigopt-gpus/

Takeaway: Real world applications require specialized experimentation and optimization tools

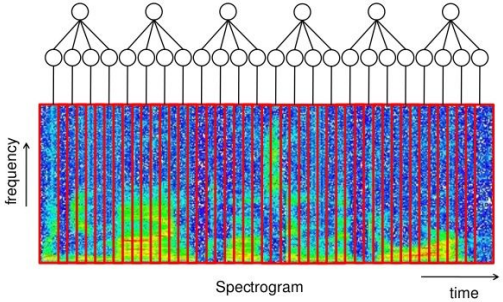
- Multiple metrics
- Jointly tuning architecture + hyperparameters
- Complex, dependent spaces
- Long training cycles

How do you more efficiently tune models that take a long time to train?

AlexNet to AlphaGo Zero:
A 300,000x Increase in Compute

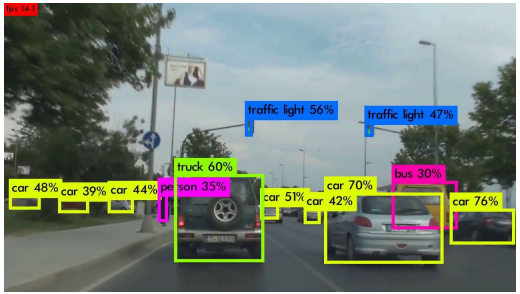


Speech Recognition

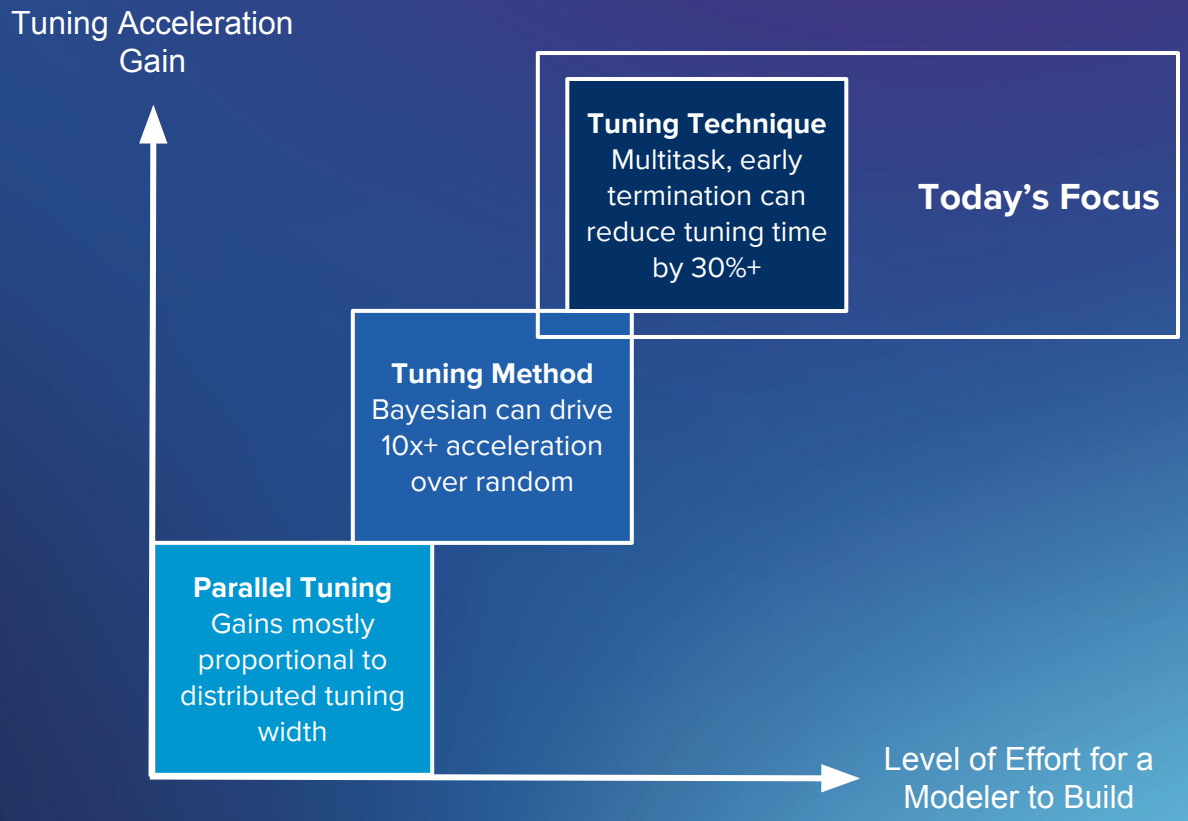


Deep Reinforcement Learning

Computer Vision



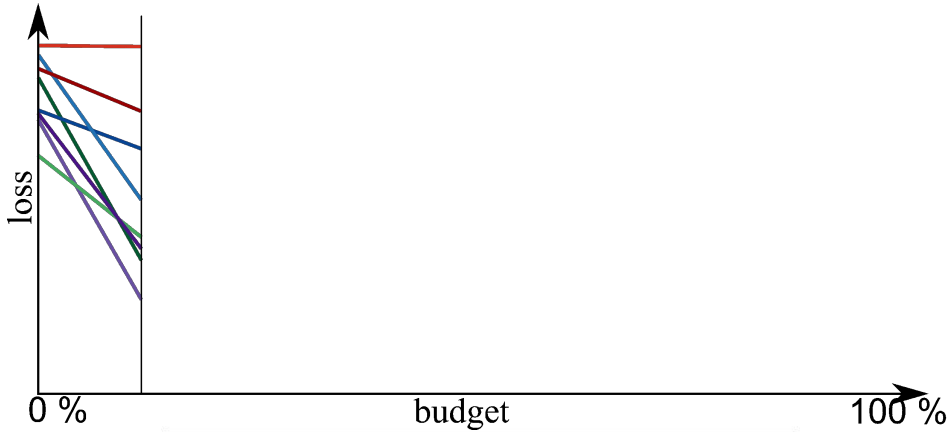
Hardware can help


Tuning Acceleration Gain
Tuning Technique
Multitask, early termination can reduce tuning time by 30%+
Today's Focus
Tuning Method
Bayesian can drive 10x+ acceleration over random
Parallel Tuning
Gains mostly proportional to distributed tuning width
Level of Effort for a Modeler to Build

Start with a simple idea:

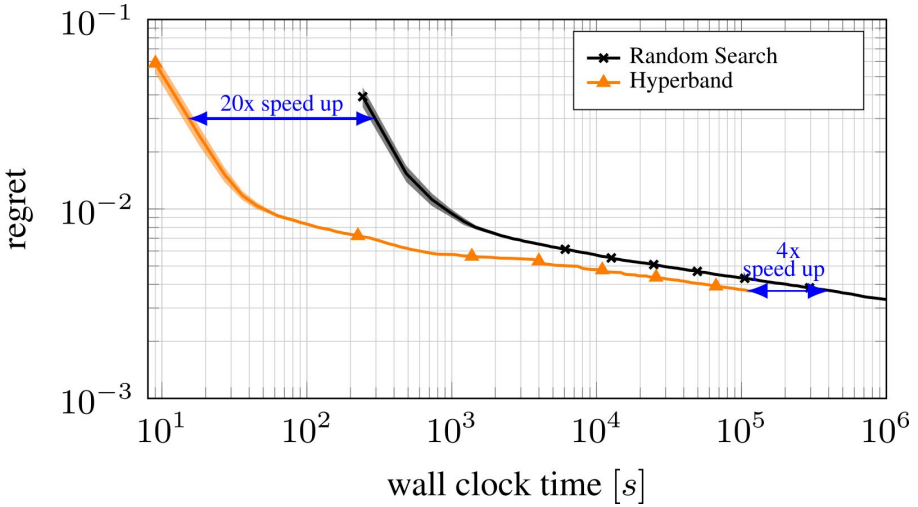
We can use information about “partially trained” models to more efficiently inform hyperparameter tuning

Previous work: Hyperband / Early Termination



	$s=4$		$s=3$		$s=2$		$s=1$		$s=0$	
i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i	n_i	r_i
0	81	1	27	3	9	9	6	27	5	81
1	27	3	9	9	3	27	2	81		
2	9	9	3	27	1	81				
3	3	27	1	81						
4	1	81								

Table 1: Values of n_i and r_i for the brackets of HYPERBAND when $R = 81$ and $\eta = 3$.



Random search, but stop poor performance early at a grid of checkpoints.

Converges to traditional random search quickly.

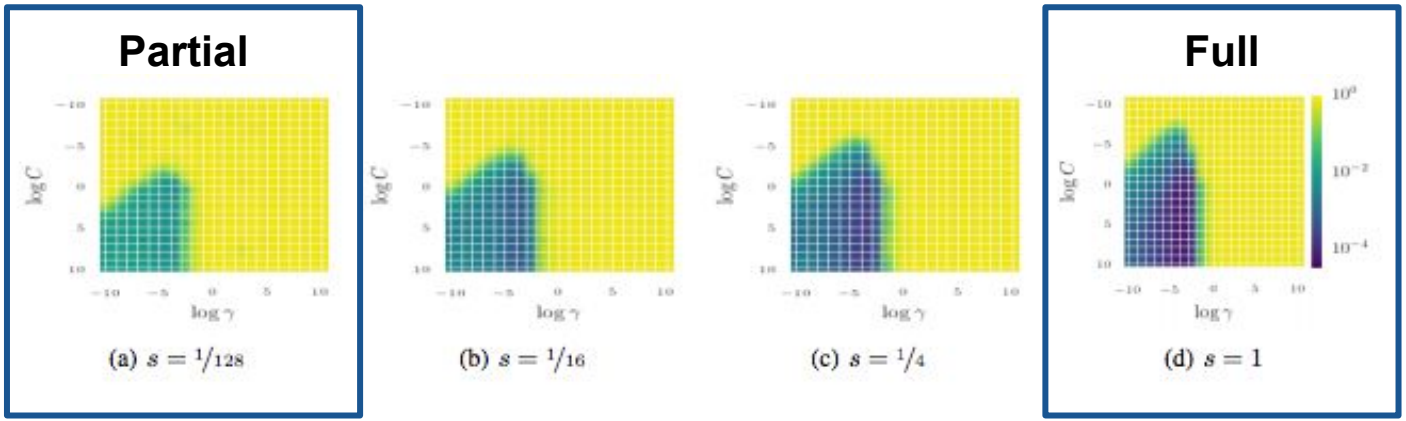
https://www.automl.org/blog_bohb/ and Li, et al, https://openreview.net/pdf?id=ry18Ww5ee

Building on prior research related to successive halving and Bayesian techniques, Multitask samples lower-cost tasks to inexpensively learn about the model and accelerate full Bayesian Optimization.

Swersky, Snoek, and Adams, “Multi-Task Bayesian Optimization”

http://papers.nips.cc/paper/5086-multi-task-bayesian-optimization.pdf

Visualizing Multitask: Learning from Approximation



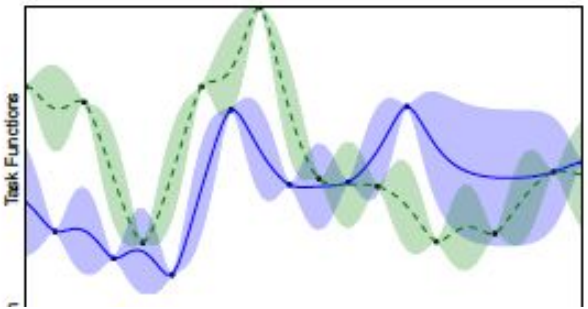
Source: Klein et al., https://arxiv.org/pdf/1605.07079.pdf

Cheap approximations promise a route to tractability, but bias and noise complicate their use. An unknown bias arises whenever a computational model incompletely models a real-world phenomenon, and is pervasive in applications.

Poloczek, Wang, and Frazier, “Multi-Information Source Optimization”

https://papers.nips.cc/paper/7016-multi-information-source-optimization.pdf

Visualizing Multitask:
Power of Correlated Approximation Functions



Source: Swersky et al., http://papers.nips.cc/paper/5086-multi-task-bayesian-optimization.pdf

Why multitask optimization?

Case: Putting Multitask Optimization to the Test

Goal: Benchmark the performance of Multitask and Early Termination methods

Model: SVM

Dataset: Covertype, Vehicle, MNIST

Methods:

- Multitask Enhanced (Fabolas)
- Multitask Basic (MTBO)
- Early Termination (Hyperband)
- Baseline 1 (Expected Improvement)
- Baseline 2 (Entropy Search)

Source: Klein et al., https://arxiv.org/pdf/1605.07079.pdf

Result: Multitask Outperforms other Methods

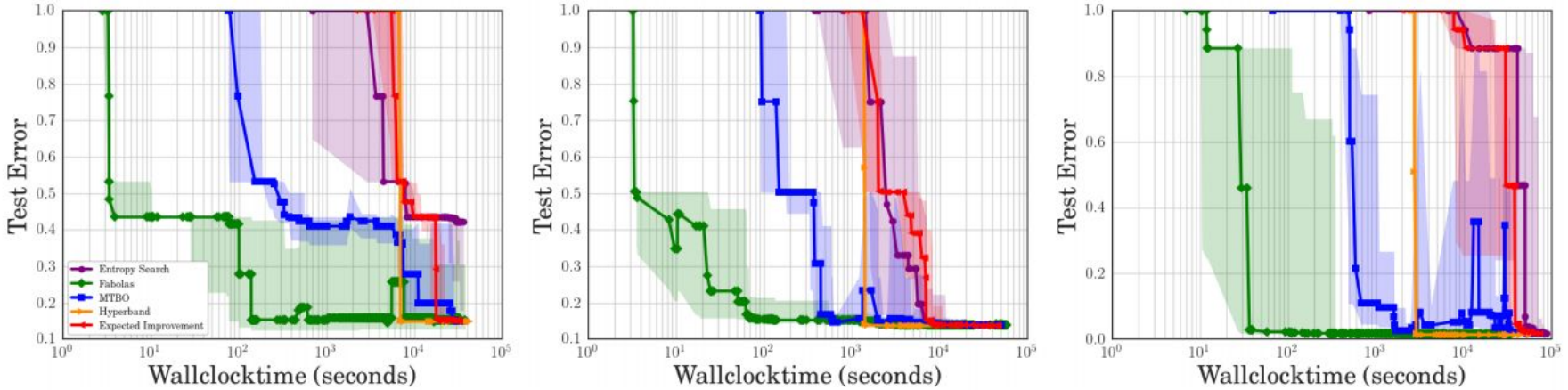


Figure 3: SVM hyperparameter optimization on the datasets covertype (left), vehicle (middle) and MNIST(right). At each time, the plots show test performance of the methods' respective incumbents. FABOLAS finds a good configuration between 10 and 1000 times faster than the other methods.

Source: Klein et al., https://arxiv.org/pdf/1605.07079.pdf

Case study

Can we accelerate optimization and improve performance on a prevalent deep learning use cases?

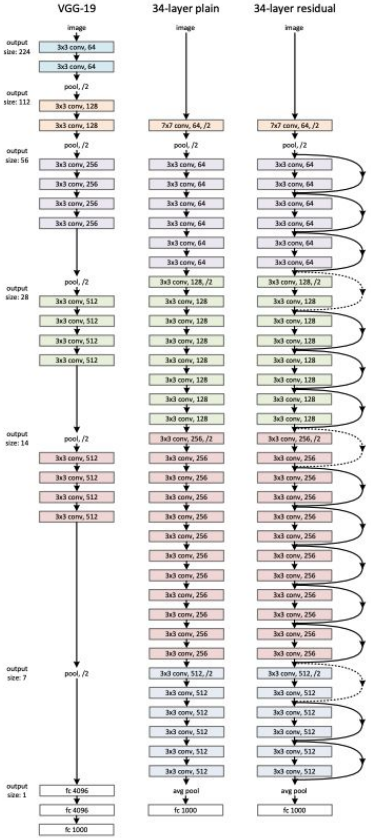
Case: Cars Image Classification

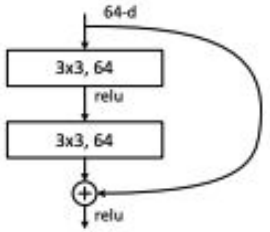
Stanford Dataset
https://ai.stanford.edu/~jkrause/cars/car_dataset.html

16,185 images, 196 classes

Labels: Car, Make, Year

Resnet: A powerful tool for image classification





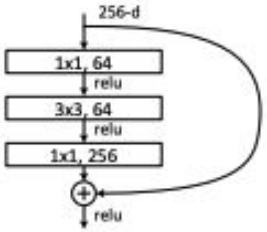


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

Experiment scenarios

Architecture Comparison	Baseline	SigOpt Multitask
ResNet 50	Scenario 1a Pre-Train on Imagenet Tune Fully Connected Layer	Scenario 1b Optimize Hyperparameters to Tune the Fully Connected Layer
ResNet 18	Scenario 2a Fine Tune Full Network	Scenario 2b Optimize Hyperparameters to Fine Tune the Full Network

Model Tuning Impact Analysis

Hyperparameter setup

Hyperparameter	Lower Bound	Upper Bound	Categorical Values	Transformation
Learning Rate	1.2e-4	1.0	-	log
Learning Rate Scheduler	0	0.99	-	-
Batch Size	16	256	-	Powers of 2
Nesterov	-	-	True, False	-
Weight Decay	1.2e-5	1.0	-	log
Momentum	0	0.9	-	-
Scheduler Step	1	20	-	-

Results: Optimizing and tuning the full network outperforms

	Baseline	SigOpt Multitask
ResNet 50	Scenario 1a 46.41%	Scenario 1b 47.99% **(+1.58%)**
ResNet 18	Scenario 2a 83.41%	Scenario 2b 87.33% **(+3.92%)**

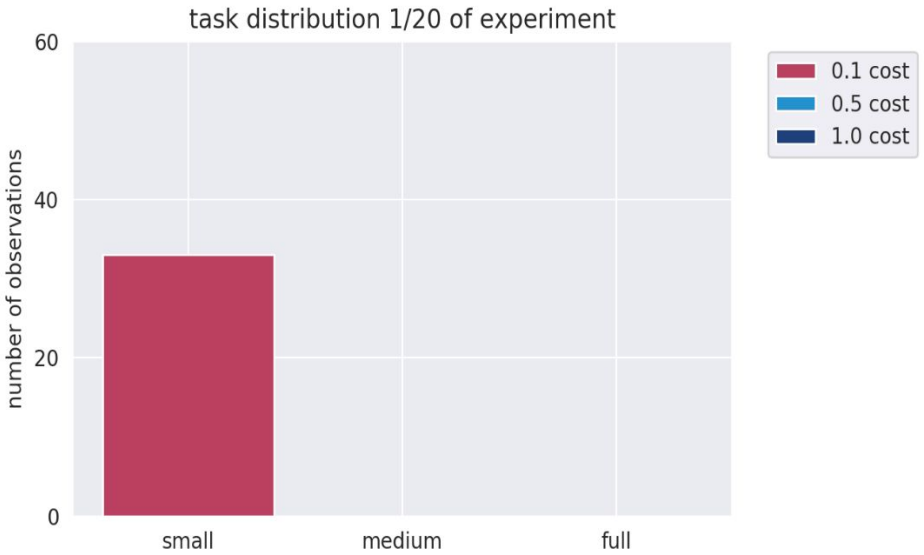
Fully Tuning the Network Outperforms

Opportunity for Hyperparameter Optimization to Impact Performance

Insight: Multitask improved optimization efficiency

Example: Cost allocation and accuracy over time

Low-cost tasks overly sampled at the beginning...



...and inform the full-cost to drive accuracy over time

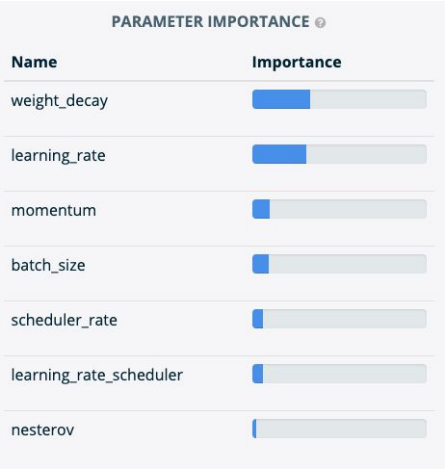
Insight: Multitask efficiency at the hyperparameter level

Example: Learning rate accuracy and values by cost of task over time

Progression of observations over time

Accuracy and value for each observation

Parameter importance analysis



Insight: Optimization improves real-world outcomes

Example: Misclassifications by baseline that were accurately classified by optimized model

Partial images
Predicted: Chrylser 300
Actual: Scion xD

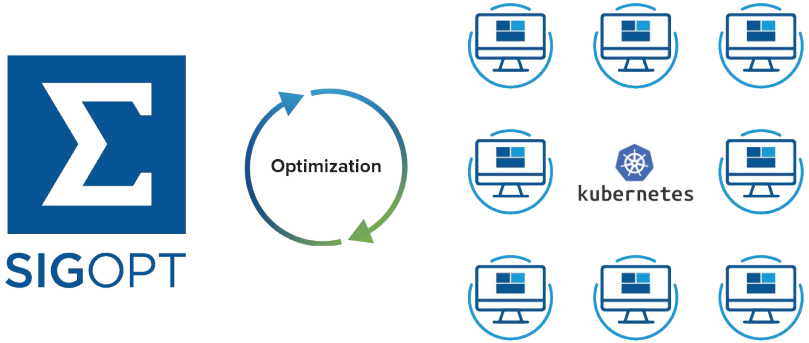
Name, design should help
Predicted: Chevy Monte Carlo
Actual: Lamborghini

Busy images
Predicted: smart fortwo
Actual: Dodge Sprinter

Multiple cars
Predicted: Nissan Hatchback
Actual: Chevy Sedan

Insight: Parallelization further accelerates wall-clock time

928 total hours to optimize ResNet 18
220 observations per experiment
20 p2.xlarge AWS ec2 instances
45 hour actual wall-clock time



Implication: Multiple benefits from multitask

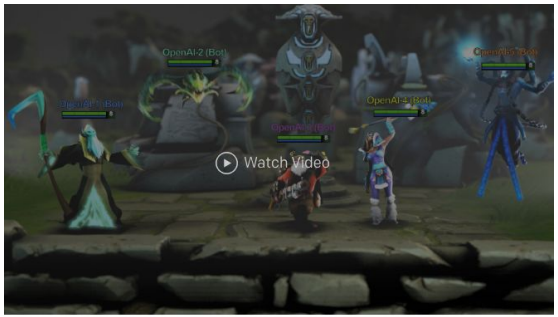
Cost efficiency	Multitask	Bayesian	Random
Hours per training	4.2	4.2	4.2
Observations	220	646	646
Number of Runs	1	1	20
Total compute hours	924	2,713	54,264
Cost per GPU-hour	$0.90	$0.90	$0.90
Total compute cost	**$832**	**$2,442**	**$48,838**

1.7% the cost of random search to achieve similar performance

Time to optimize	Multitask	Bayesian	Random
Total compute hours	924	2,713	54,264
# of Machines	20	20	20
Wall-clock time (hrs)	**46**	**136**	**2,713**

58x faster wall-clock time to optimize with multitask than random search

Impact of efficient tuning grows with model complexity



OpenAI Five

Our team of five neural networks, OpenAI Five, has started to defeat amateur human teams at Dota 2.

JUNE 25, 2018 · 14 MINUTE READ

AlphaStar: Mastering the Real-Time Strategy Game StarCraft II

Games have been used for decades as an important way to test and evaluate the performance of artificial intelligence systems. As capabilities have increased, the research community has sought games with increasing complexity that capture different elements of intelligence required to solve scientific and real-world problems. In recent years, StarCraft, considered to be one of the most challenging Real-Time Strategy (RTS) games and one of the longest-played esports of all time, has emerged by consensus as a "grand challenge" for AI research.

Google AI

RESEARCH › PUBLICATIONS ›

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Download
Search
Copy Bibtex

Summary

Optimizing particularly expensive models is a tough challenge

Hardware is part of the solution, as is adding width to your experiment

Algorithmic solutions offer compelling ways to further accelerate

These solutions typically improve model performance *and* wall-clock time

SIGOPT

Learn more about Multitask Optimization: https://app.sigopt.com/docs/overview/multitask

Free access for Academics & Nonprofits: https://sigopt.com/edu

Solution-oriented program for the Enterprise: https://sigopt.com/pricing

Leading applied optimization research: https://sigopt.com/research

GitHub repo for this use case:
https://github.com/sigopt/sigopt-examples/tree/master/stanford-car-classification

... and we're hiring! https://sigopt.com/careers

Thank you!