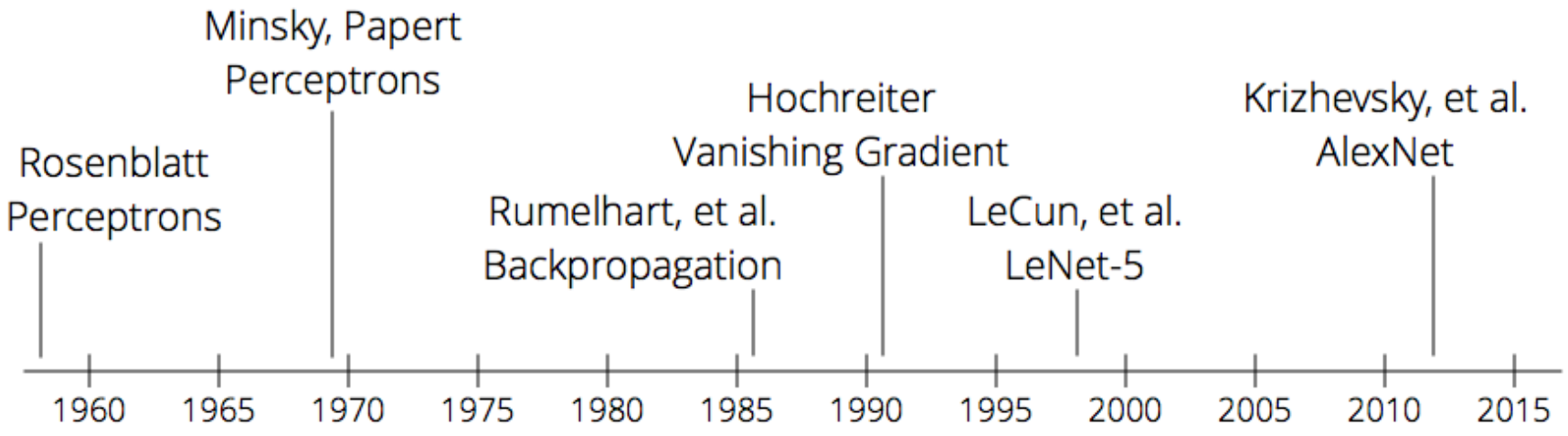
Computer architecture for deep learning applications

David Brooks

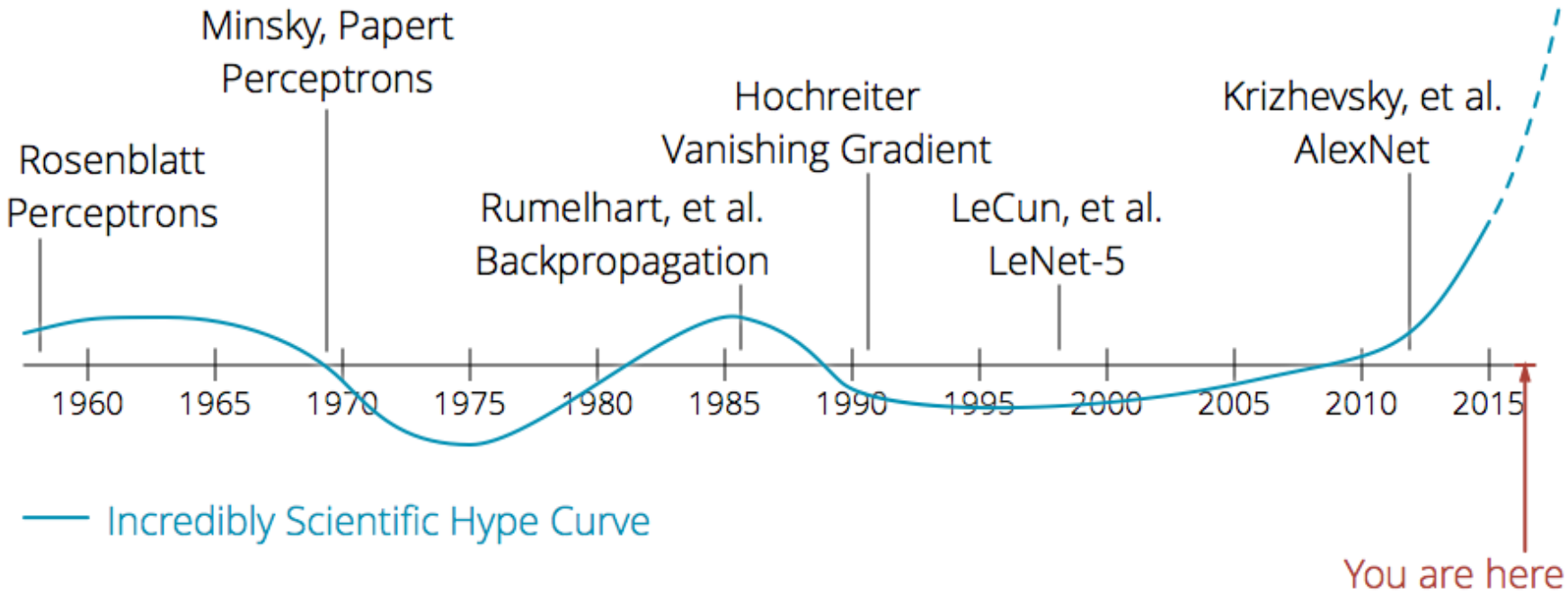
School of Engineering and Applied Sciences

Harvard University

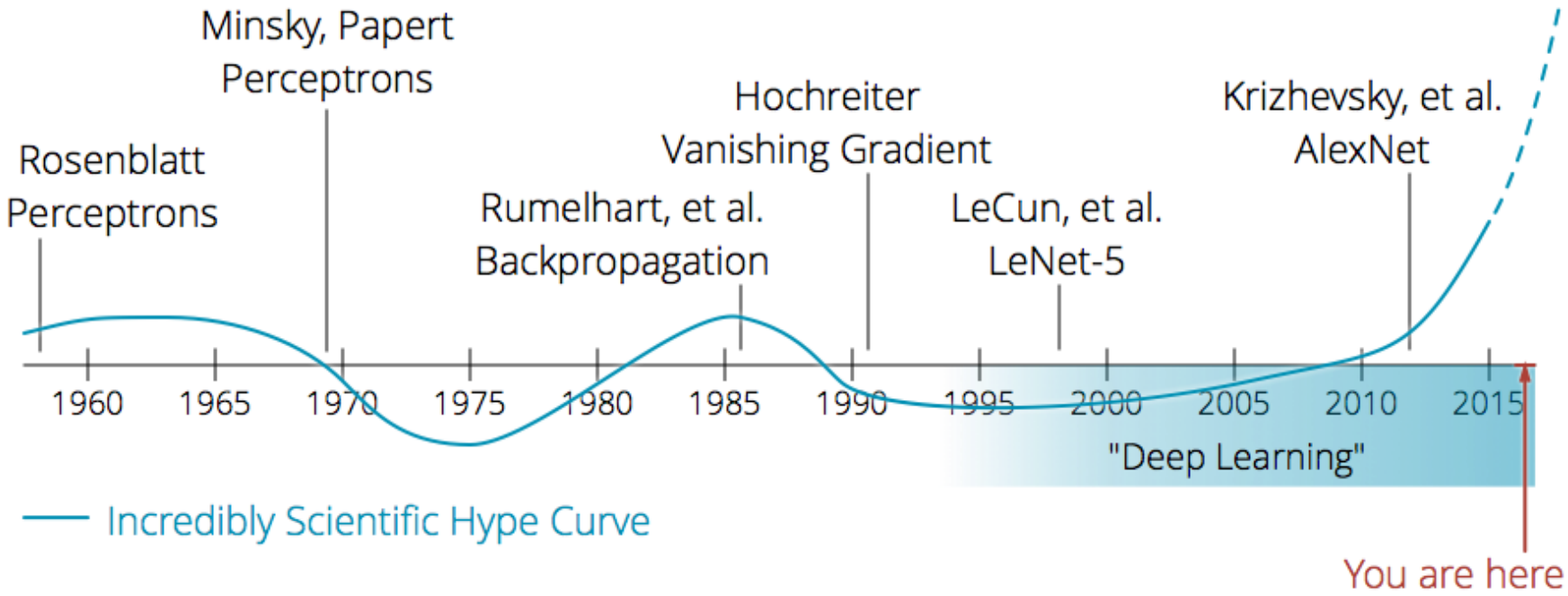
The rise of deep learning



The rise of deep learning



The rise of deep learning



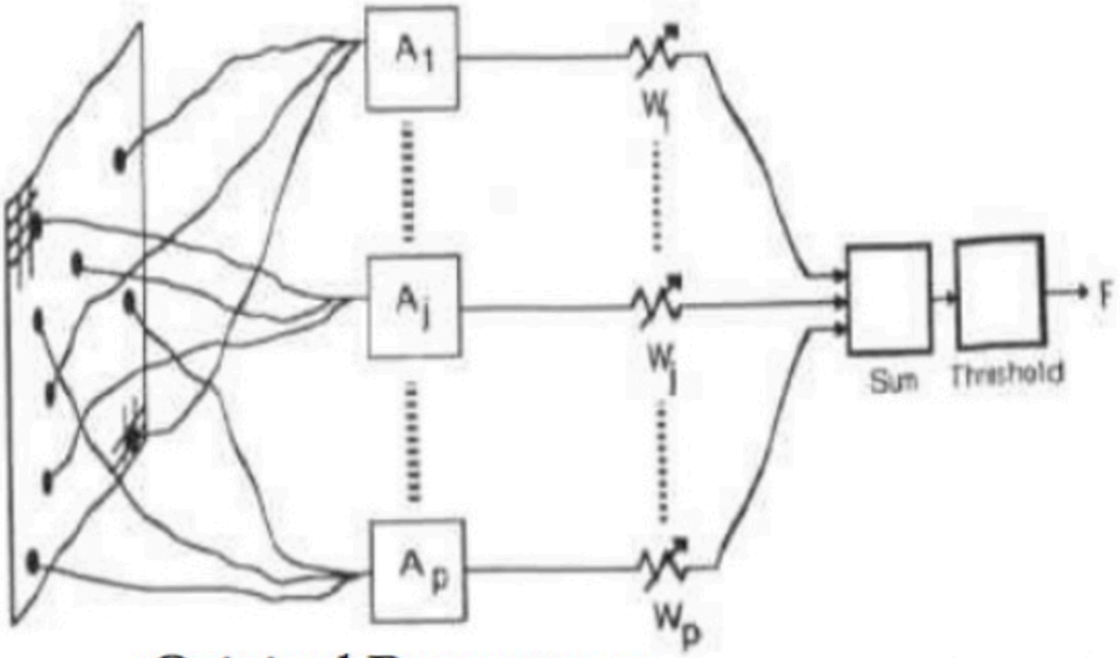
Google Translate ➔ Neural in Nov'16

https://blog.google/products/translate/translate-where-you-need-it-in-any-app/

Google Translate ➔ Neural in Nov'16

https://blog.google/products/translate/translate-where-you-need-it-in-any-app/

Why computer architecture for ML?

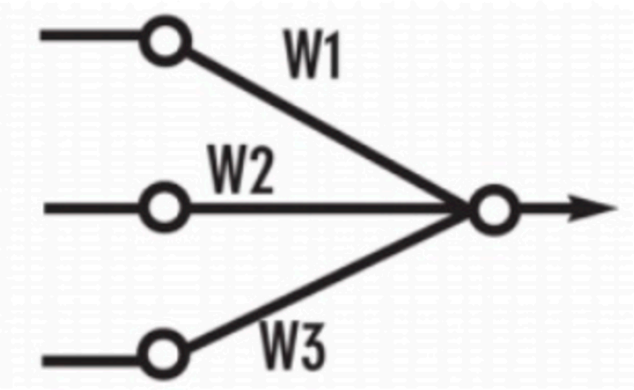


Original Perceptron

(From Perceptrons by M. L Minsky and S. Papert, 1969, Cambridge, MA: MIT Press. Copyright 1969 by MIT Press.

Frank Rosenblatt
(1928-1971)

Simplified model:

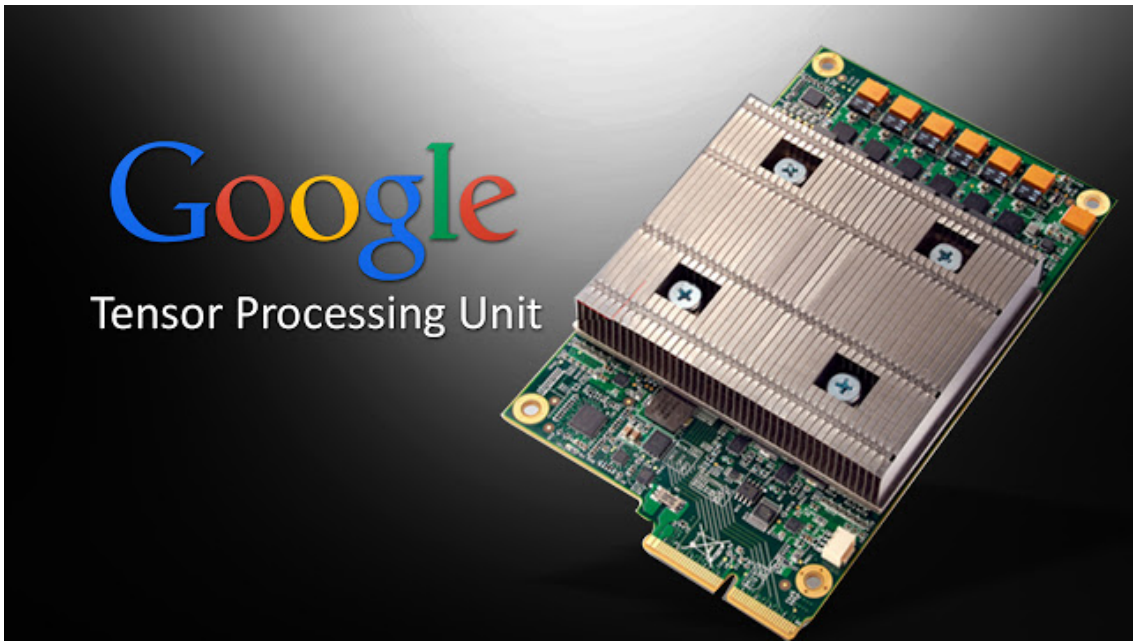


Why computer architecture for ML?

"The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence... [It] is expected to be finished in about a year at a cost of $100,000... Later perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech in another."

New Navy Device Learns By Doing, New York Times, July 1958

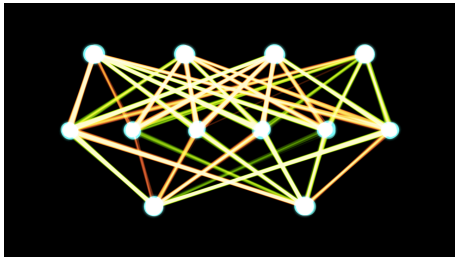
Why computer architecture for ML?

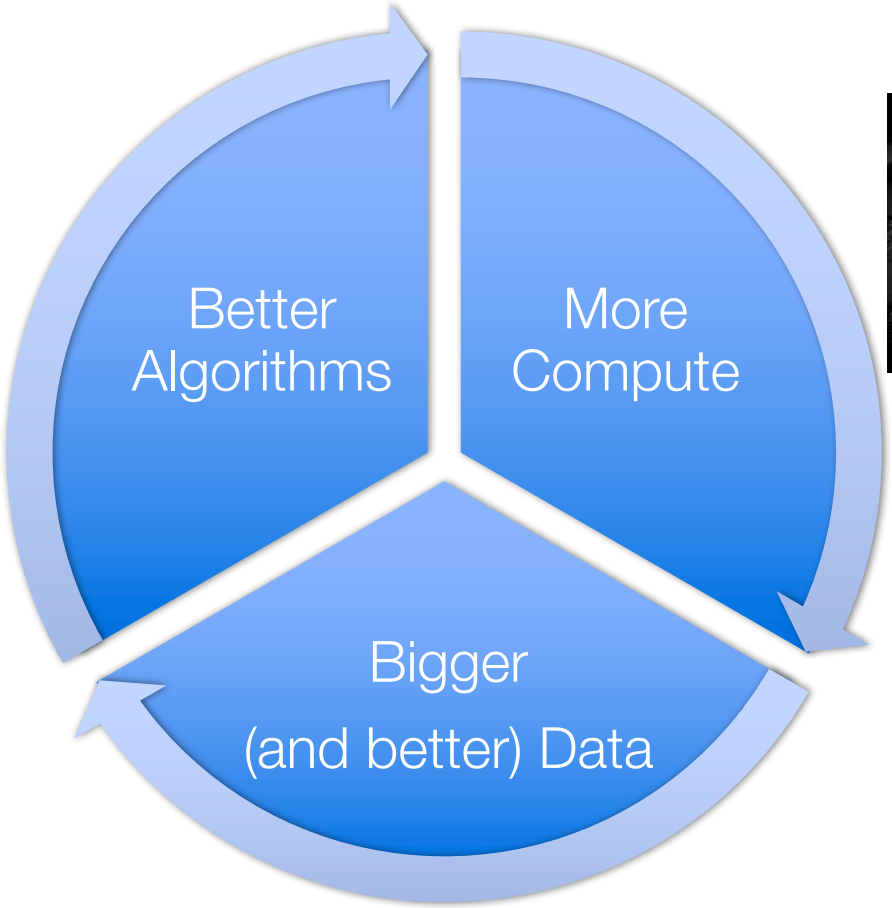


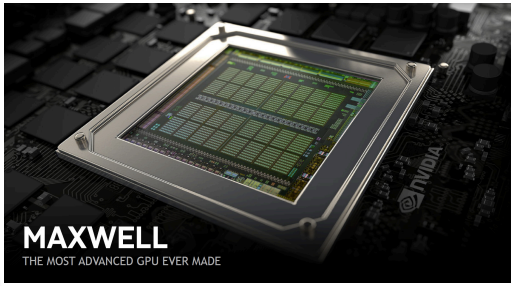
"By May, the (Google) Brain team understood that the only way they were ever going to make the system fast enough to implement as a product was if they could run it on T.P.U.s, the special-purpose chips that (Jeff) Dean had called for. As (Zhifeng) Chen put it: "We did not even know if the code would work. But we did know that without T.P.U.s, it *definitely* wasn't going to work."

The Great A.I. Awakening, New York Times, Dec 2016

Today's virtuous cycle







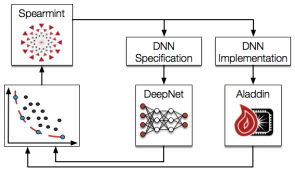
IMAGENET

Architectural Support for Deep Learning at Harvard

A Full-Stack Approach to Machine Learning

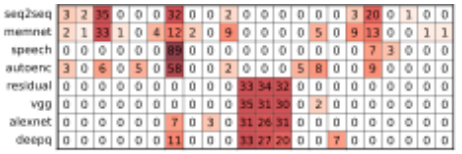
Algorithms

Co-Designing Deep Neural Network Accelerators for Accuracy and Energy Using Bayesian Optimization



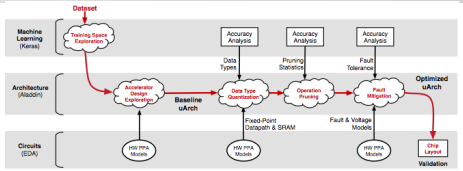
Tools

Fathom: Reference Workloads for Modern Deep Learning Methods



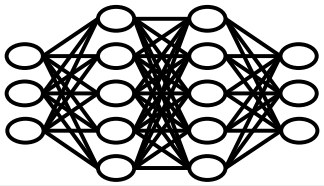
Architectures

Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators



Circuits

SM2: A Deep Neural Network Accelerator SoC in 28nm bulk and 16nm FinFET

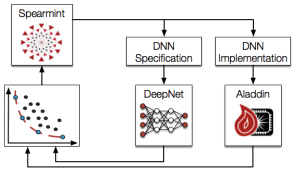


Architectural Support for Deep Learning at Harvard

A Full-Stack Approach to Machine Learning

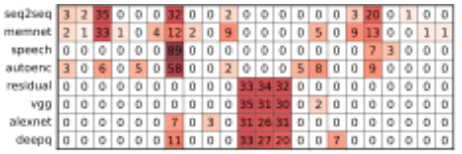
Algorithms

Co-Designing Deep Neural Network Accelerators for Accuracy and Energy Using Bayesian Optimization



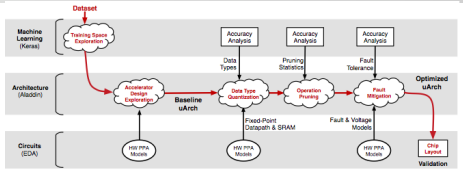
Tools

Fathom: Reference Workloads for Modern Deep Learning Methods



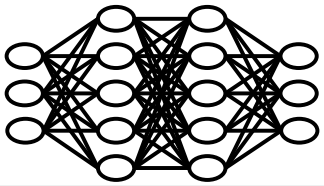
Architectures

Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators



Circuits

SM2: A Deep Neural Network Accelerator SoC in 28nm bulk and 16nm FinFET



Shortcomings of current hardware research

1. Narrow focus

Researchers have latched on to just a few methods

2. Mismatch between research and reality

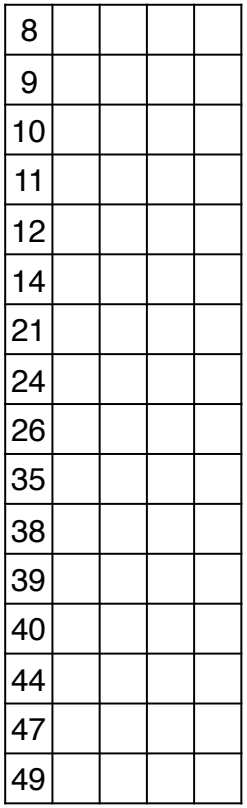
We need real models, real data, and real environments

3. Abundant folklore

Lack of hard numbers leads to conflicting assumptions

The community has a narrow focus

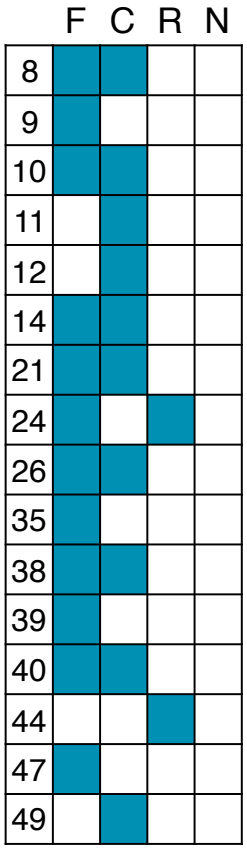
Characteristics of deep learning models



16 research projects from top-tier conferences

The community has a narrow focus

Neuronal style: What building blocks are used?



Neuronal Style

Fully-connected (FC) neural networks

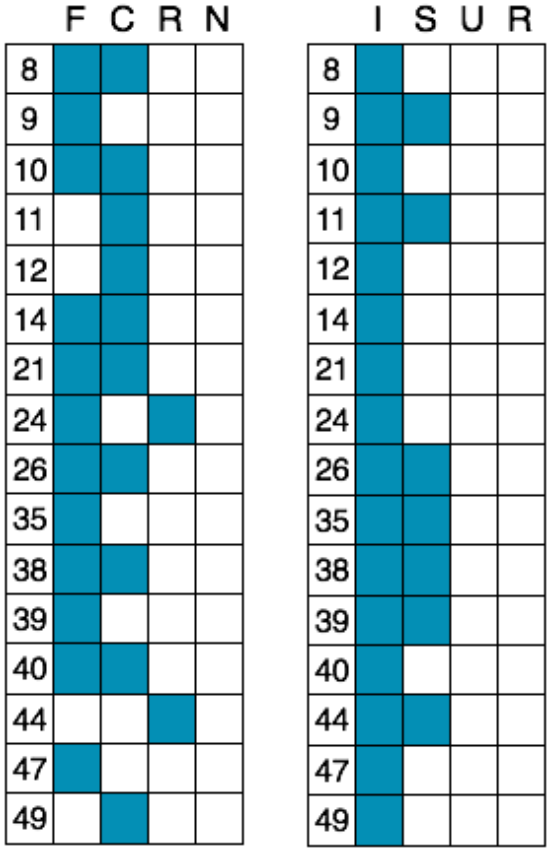
Convolutional neural networks (CNN)

Recurrent neural networks (RNN)

Novel architectures (everything else)

The community has a narrow focus

Learning task: What are the underlying use-case assumptions?



Inference: use a pre-trained network

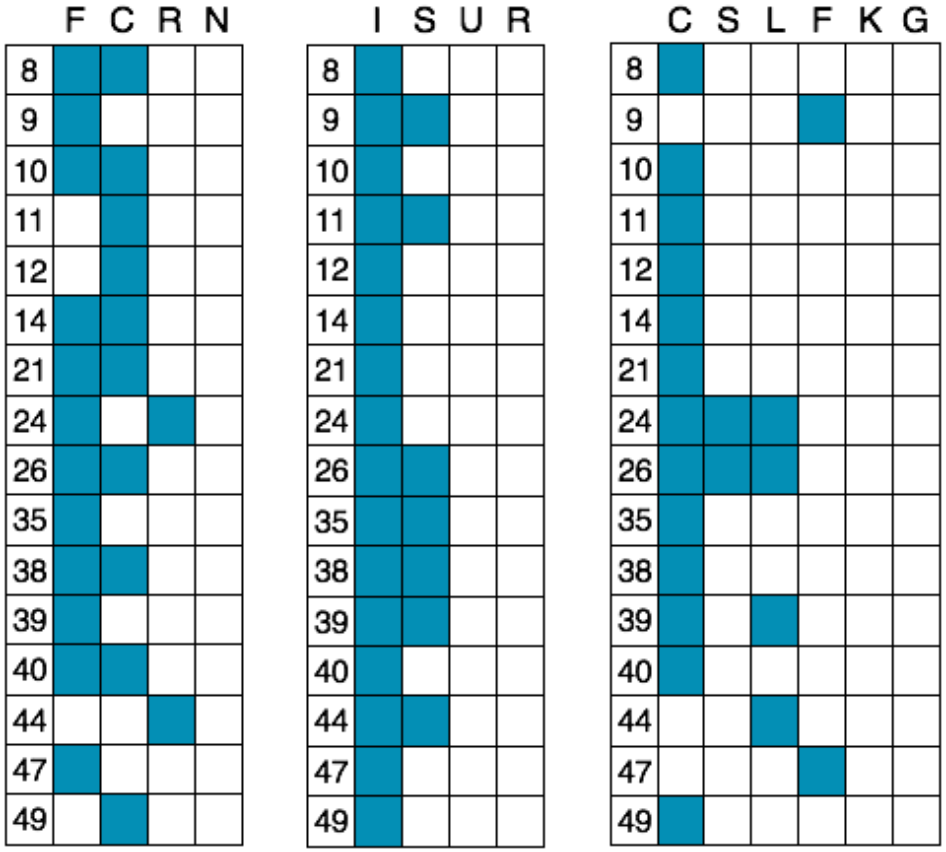
Supervised: train with labeled data

Unsupervised: train without labels

Reinforcement: train with loose feedback

The community has a narrow focus

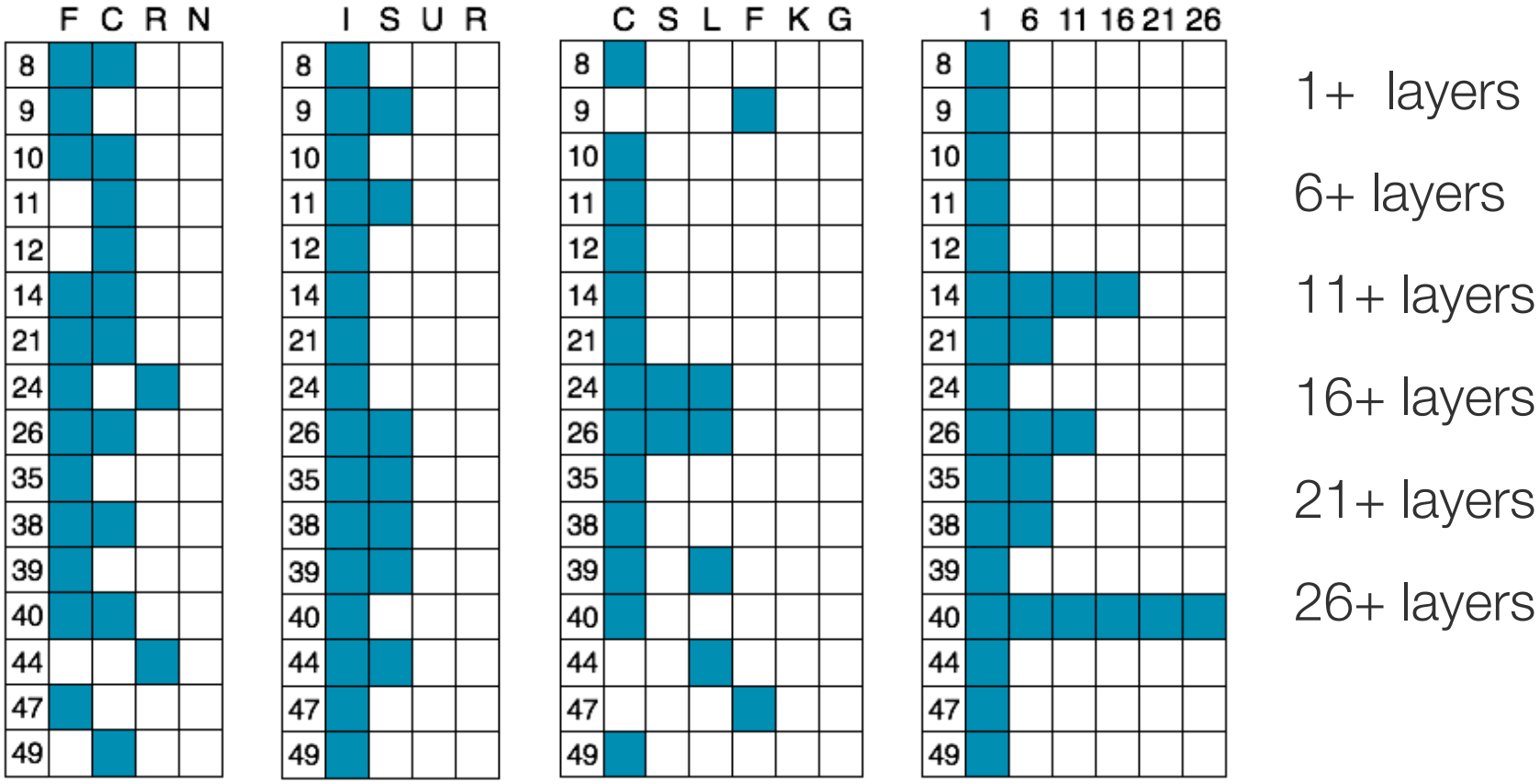
Application: Which problem domains are considered?



- Computer vision
- Speech recognition
- Language modeling
- Function approximation
- Knowledge reasoning
- General AI

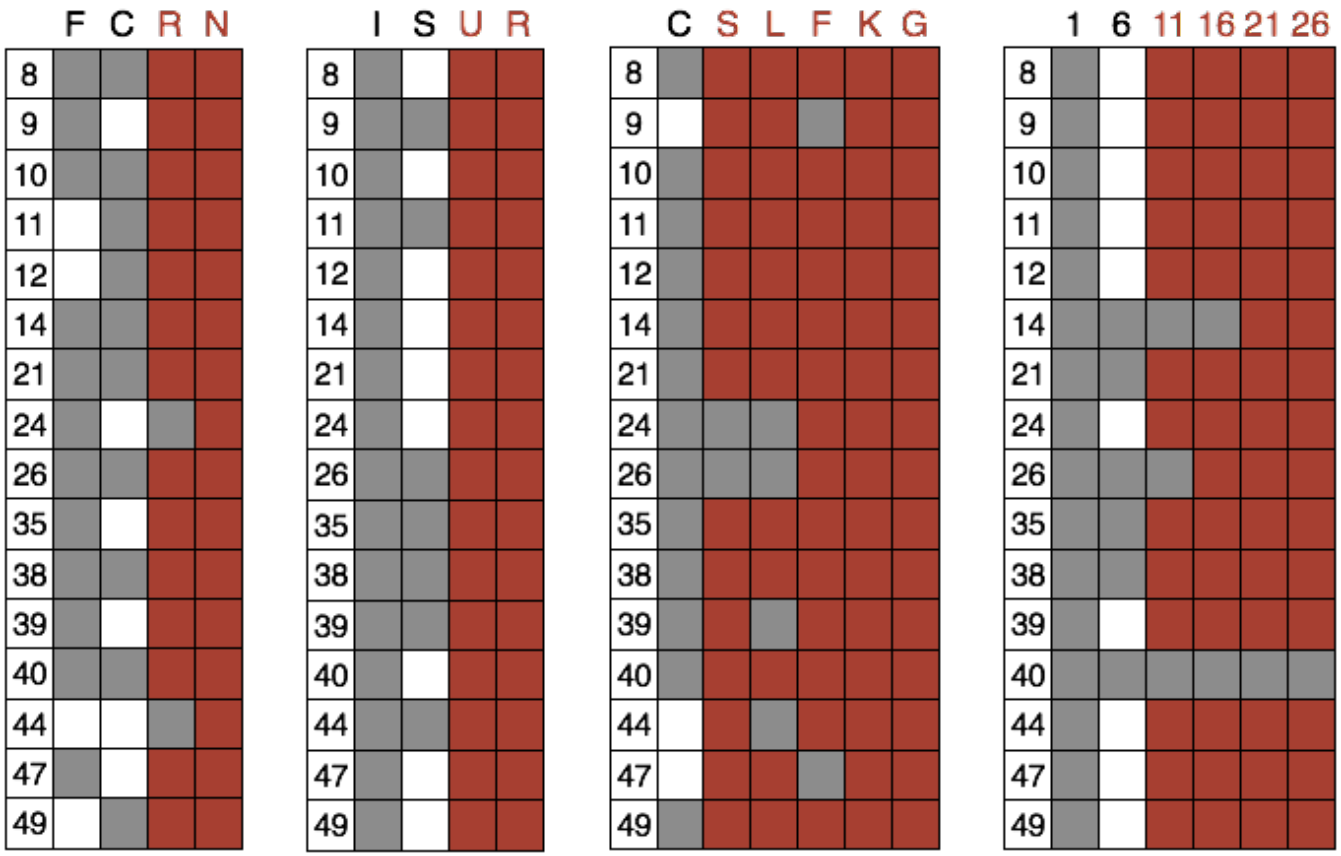
The community has a narrow focus

Model depth: How large are the models?



The community has a narrow focus

This is a problem.



Neuronal Style

Learning Task

Application Domain

Model Depth

Realism in models, data, and environments

Existing Research…	…and Reality
Stable, established models; avoids state of the art	Models are constantly in flux; new ones appear often

Realism in models, data, and environments

Existing Research…	…and Reality
Stable, established models; avoids state of the art	Models are constantly in flux; new ones appear often
Small, manageable data sets, used in isolation	Large, unwieldy data sets, often combined with preprocessing or staging

Realism in models, data, and environments

Existing Research…	…and Reality
Stable, established models; avoids state of the art	Models are constantly in flux; new ones appear often
Small, manageable data sets, used in isolation	Large, unwieldy data sets, often combined with preprocessing or staging
Simple, stand-alone implementations	Kernels are embedded in complex, high-level frameworks

Conflicting assumptions cause confusion

"Convolutions account for over 90% of the processing in CNNs for both inference/testing and training" - Chen et al. (2016)

"In convolutional neural network (CNN), fully connected layers [make up] more than 96% of the connections … [and] up to 38% computation time." - Han et al. (2016)

Conflicting assumptions cause confusion

"Convolutions account for over 90% of the processing in CNNs for both inference/testing and training" - Chen et al. (2016)

"In convolutional neural network (CNN), fully connected layers [make up] more than 96% of the connections … [and] up to 38% computation time." - Han et al. (2016)

The worst part? They're both right.

There is no single answer, no single design.

Conflicting assumptions cause confusion

And we finally start to see some industrial data…

Name	*LOC*	*Layers*					*Nonlinear function*	*Weights*	*TPU Ops / Weight Byte*	*TPU Batch Size*	*% of Deployed TPUs in July 2016*
		FC	*Conv*	*Vector*	*Pool*	*Total*					
MLP0	100	5				5	ReLU	20M	200	200	61%
MLP1	1000	4				4	ReLU	5M	168	168	
LSTM0	1000	24		34		58	sigmoid, tanh	52M	64	64	29%
LSTM1	1500	37		19		56	sigmoid, tanh	34M	96	96	
CNN0	1000		16			16	ReLU	8M	2888	8	5%
CNN1	1000	4	72		13	89	ReLU	100M	1750	32	

95% of Google's TPU Workloads

- Jouppi et al. (ISCA 2017)

Fathom

Reference Workloads for Modern Deep Learning

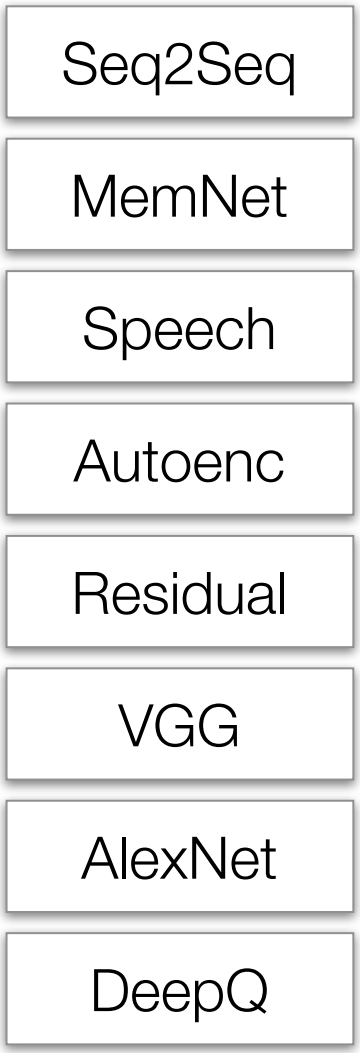
Broaden architectural research

Foster realism

Abolish deep learning folklore

Reduce barriers to entry

What is Fathom?



8 diverse, state-of-the-art learning models

Compatible with widely-used datasets

Clear, tested implementations in TensorFlow

High-level frameworks are here to stay

Training and inference modes provided

High-level behavioral characterization

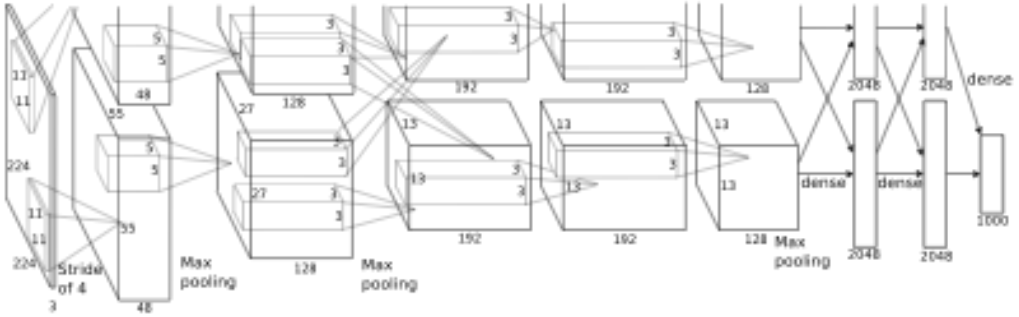
Provide hard numbers *and* intuition

The Fathom workloads

- Seq2Seq
- MemNet
- Speech
- Autoenc
- Residual
- VGG
- **AlexNet**
- DeepQ

Watershed model for deep neural networks

Neuron style: Convolutional/Fully-connected
Learning task: Supervised learning
Domain: Image classification
Model: 5-CNN,2-FC network, ReLU nonlinearity



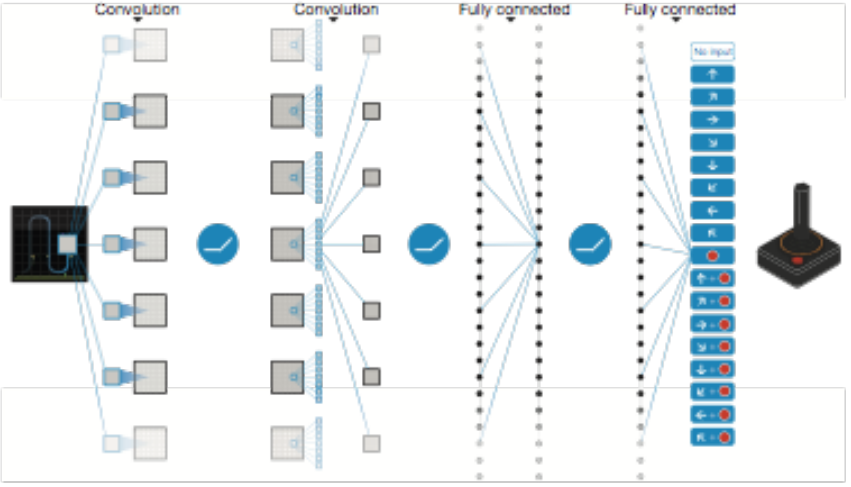
Krizhevsky, et al. "ImageNet Classification with Deep Convolutional Neural Networks." NIPS, 2012

The Fathom workloads

- Seq2Seq
- MemNet
- Speech
- Autoenc
- Residual
- VGG
- AlexNet
- DeepQ

Atari-playing neural network from DeepMind

Neuron style: Convolutional/Fully-connected
Learning task: Reinforcement learning
Domain: General AI
Model: 3-CNN,2-FC network for estimating value, trained via Q-learning with experience replay



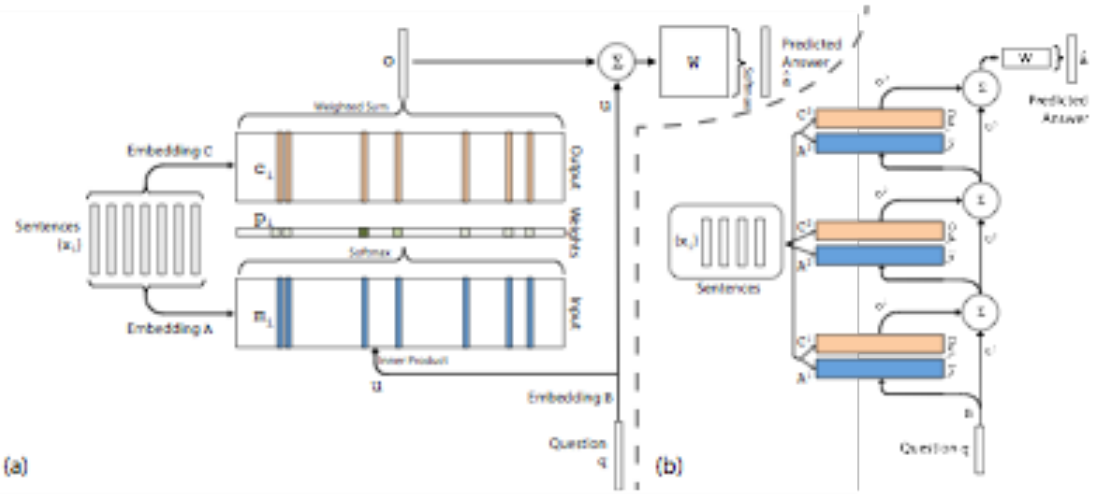
Mnih, et al. "Human-Level Control Through Deep Reinforcement Learning." Nature, 2015

The Fathom workloads

- Seq2Seq
- MemNet
- Speech
- Autoenc
- Residual
- VGG
- AlexNet
- DeepQ

Facebook's memory-oriented learning model

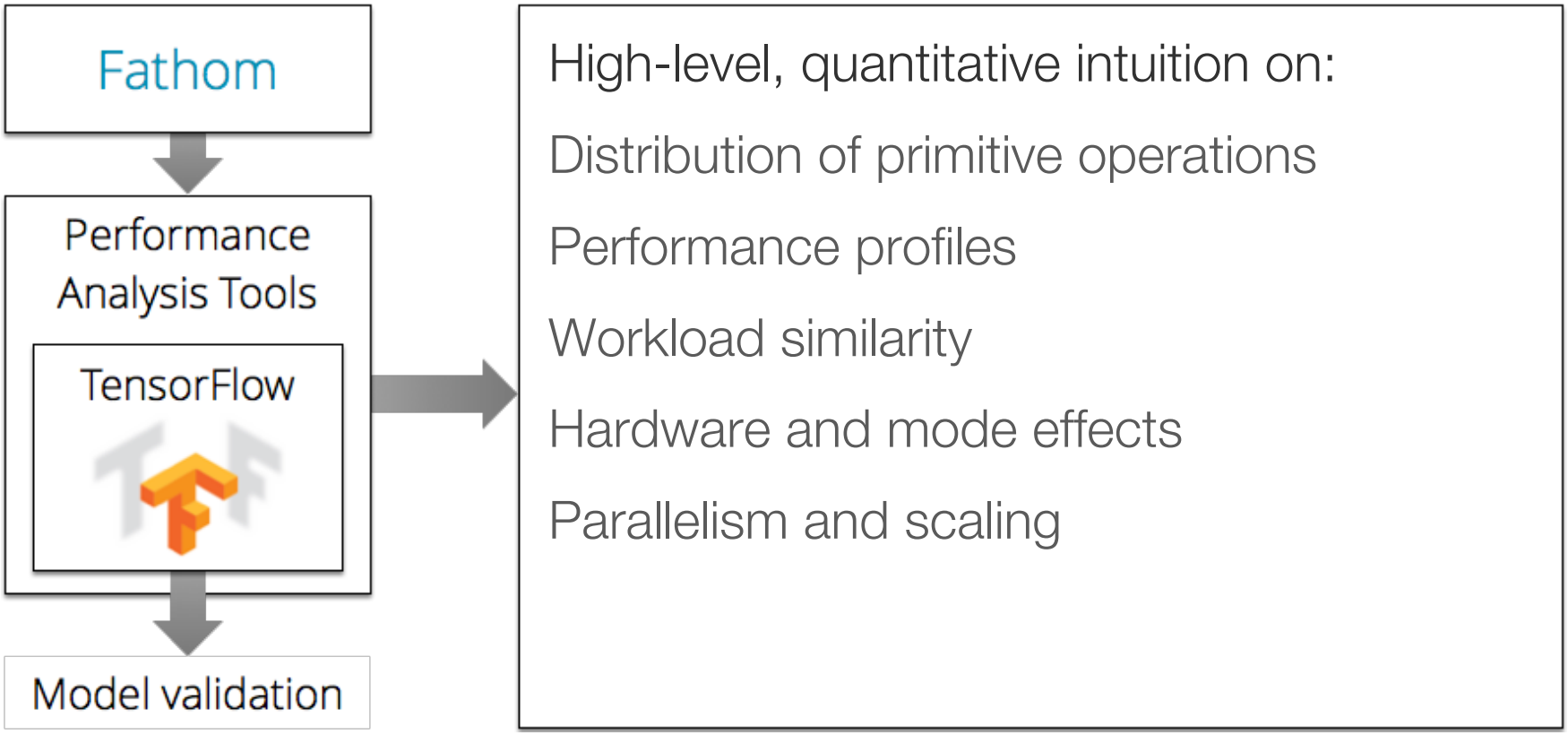
Neuron style: Memory networks
Learning task: Supervised learning
Domain: Q&A, Automated reasoning
Model: 3-layer memory network, built using indirect lookups on sentence embeddings



Sukhbaatar, et al. "End-To-End Memory Networks." NIPS, 2015

Understanding the Fathom workloads

Fathom is a tool. Tools require understanding to use.



Deep learning models in a high-level framework

TensorFlow models are coarse-grained dataflow graphs

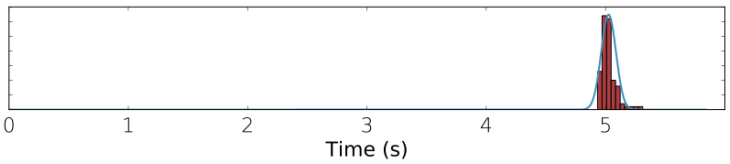
Basic building block is an “operation”

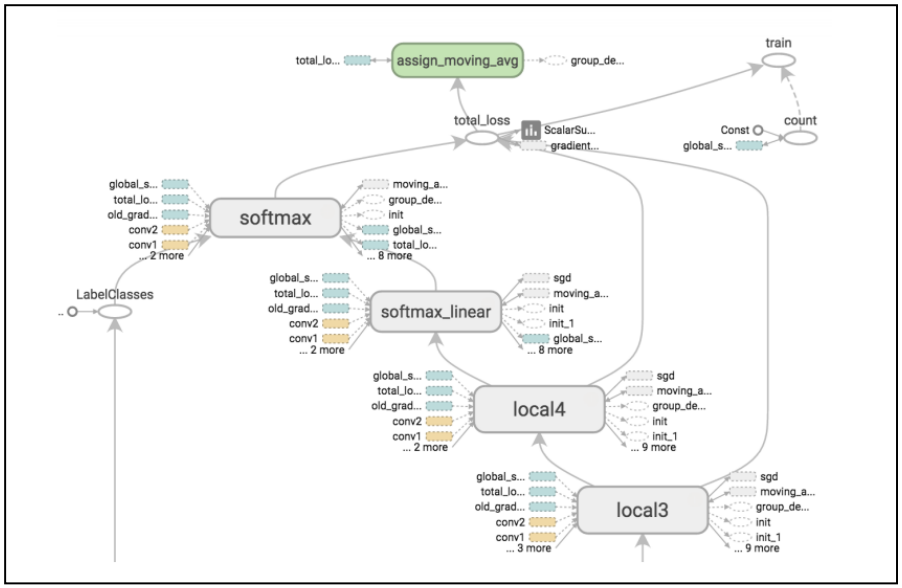
Ops are a useful abstraction

Map to underlying library

Enables causal reasoning

Stable performance across the lifetime of a run

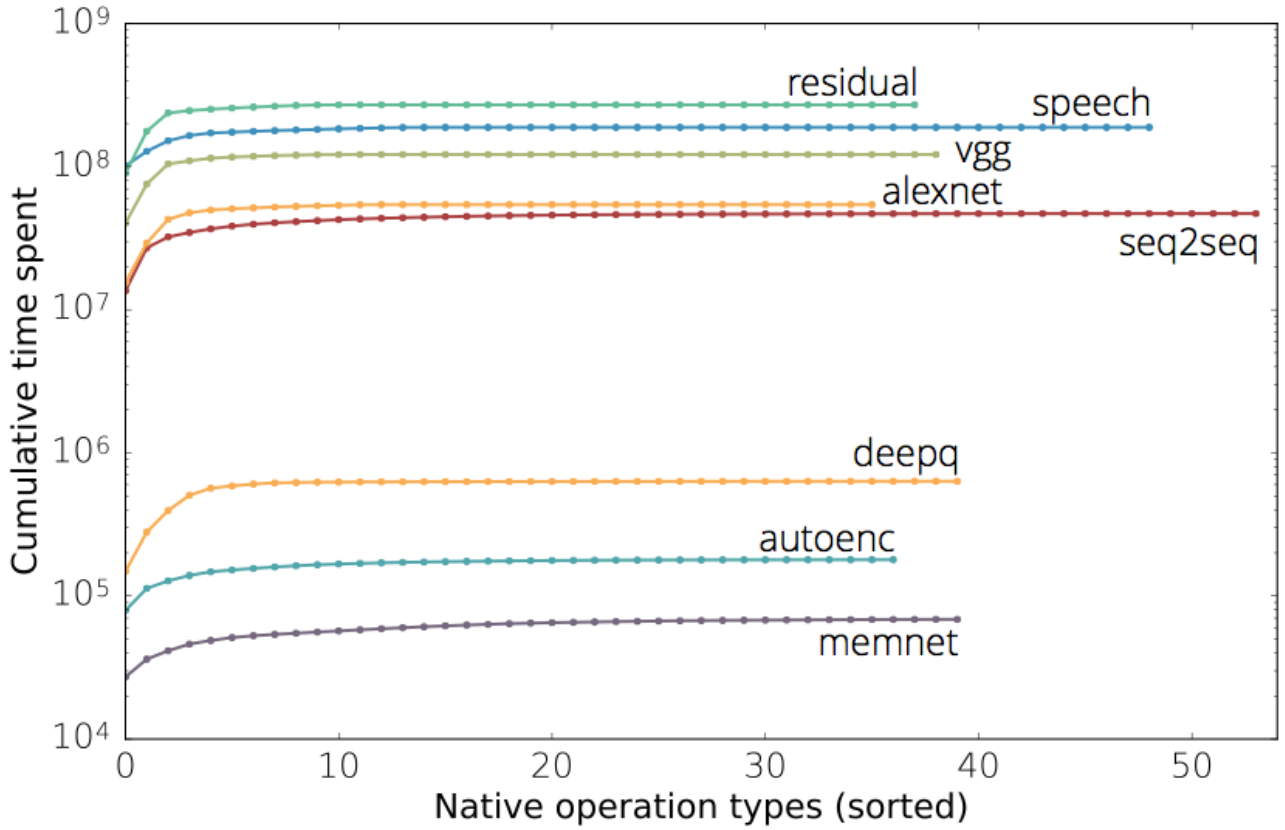




Models are dominated by a few operation types

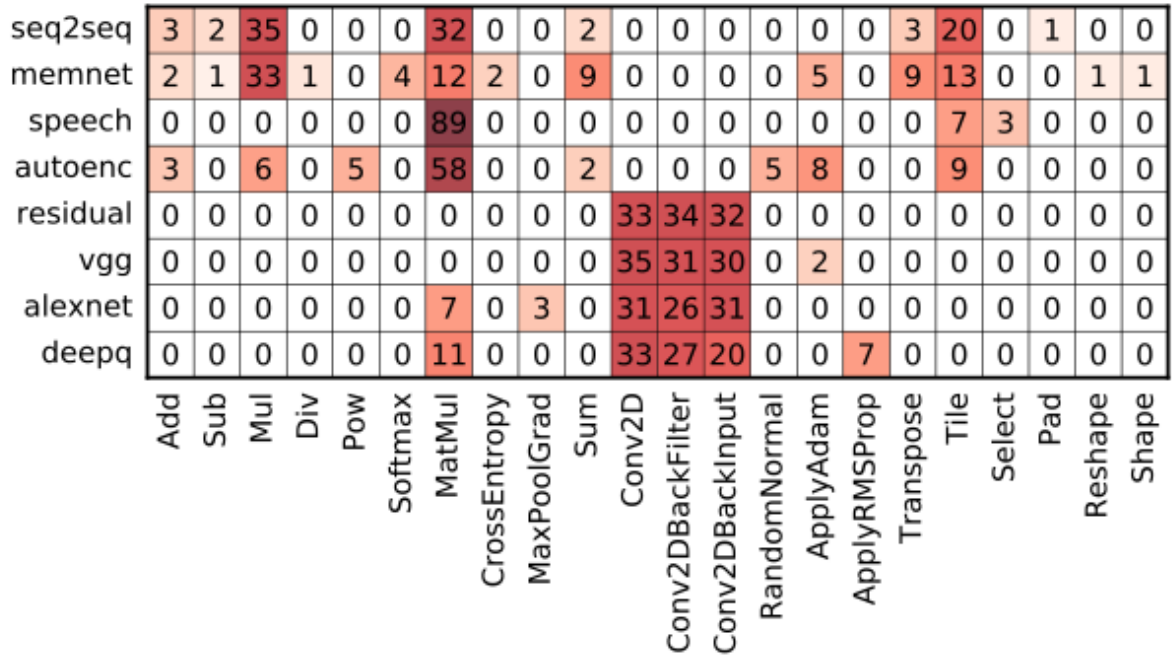
Each model spends 90% of its time in ≤6 ops

All models jointly spend 90% of their time in 22 ops



Operation type profiling

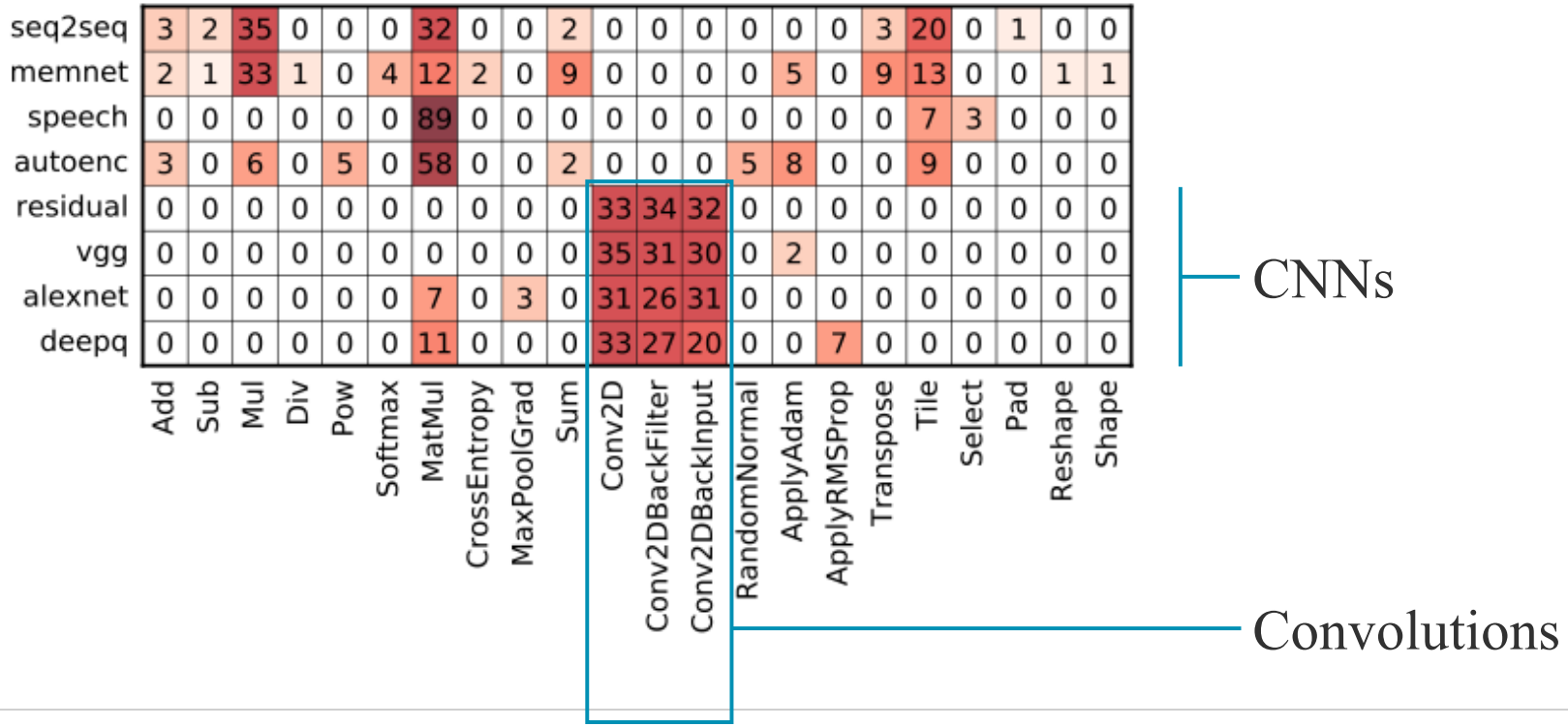
Deep learning methods rely on different primitives



Operation type profiling

Deep learning methods rely on different primitives

Some trends are obvious and expected

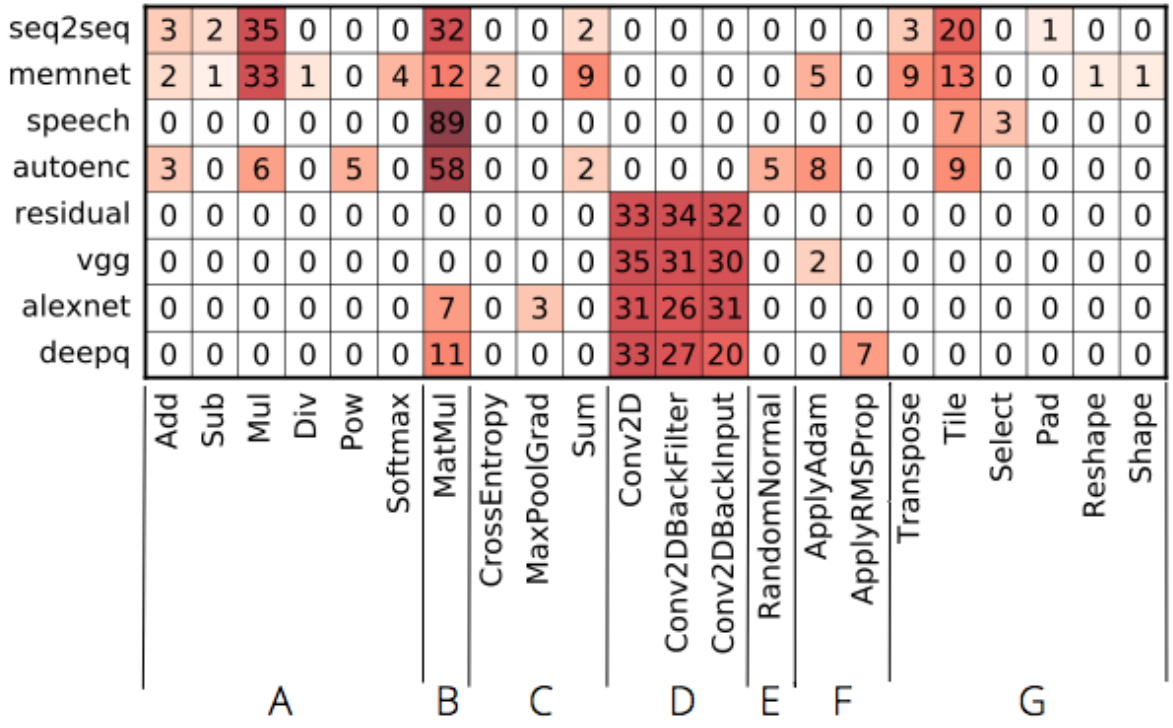


Operation type profiling

Deep learning methods rely on different primitives

Some trends are obvious and expected

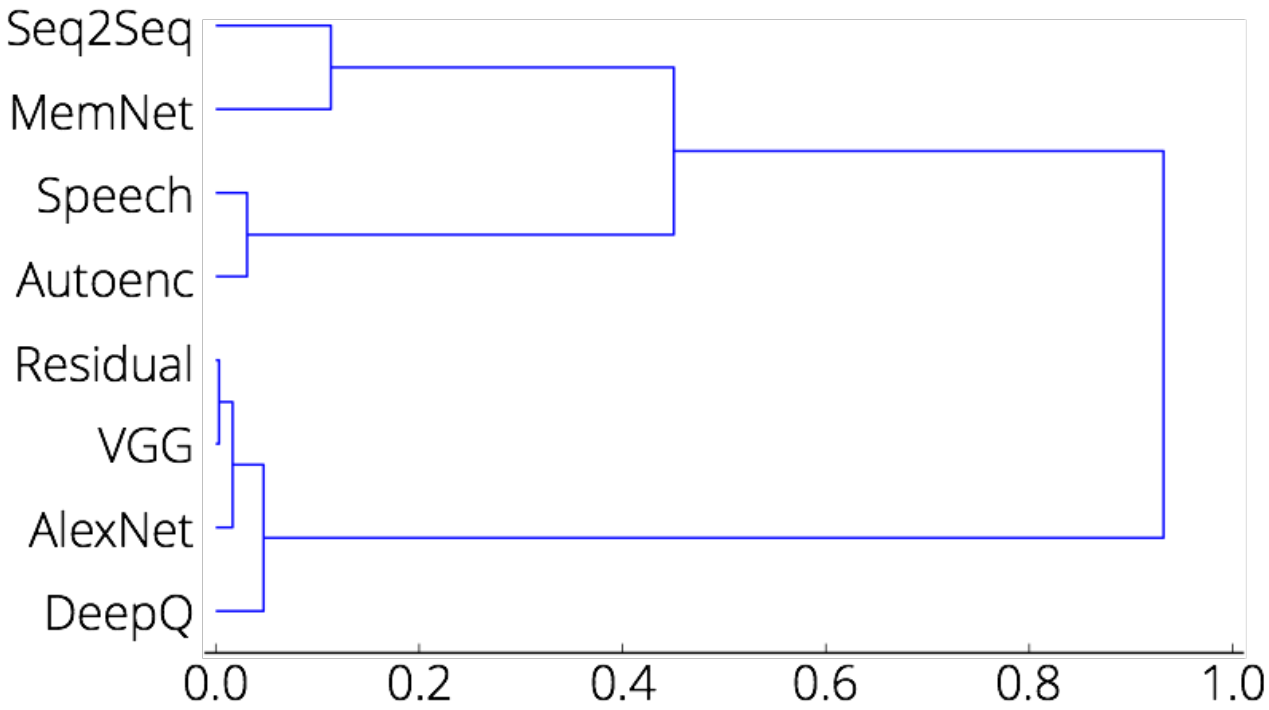
Most ops fall into a few broad performance classes



Group	Operation Class
A	Elementwise Arithmetic
B	Matrix Operations
C	Scatter/Gather
D	Convolution
E	Stochastic Methods
F	Optimization
G	Data Movement

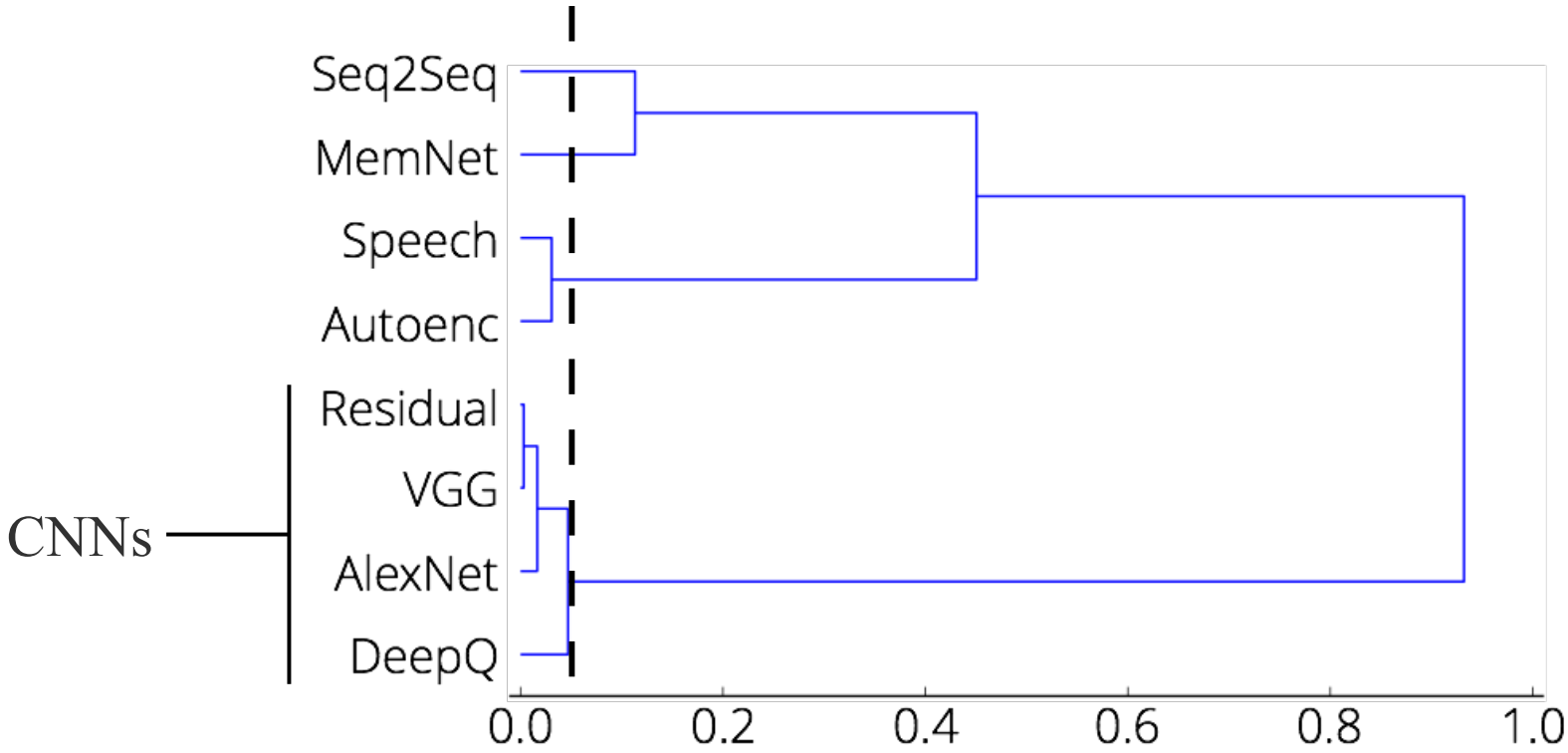
Performance similarity in Fathom

Compute similarity via cosine similarity between op profiles



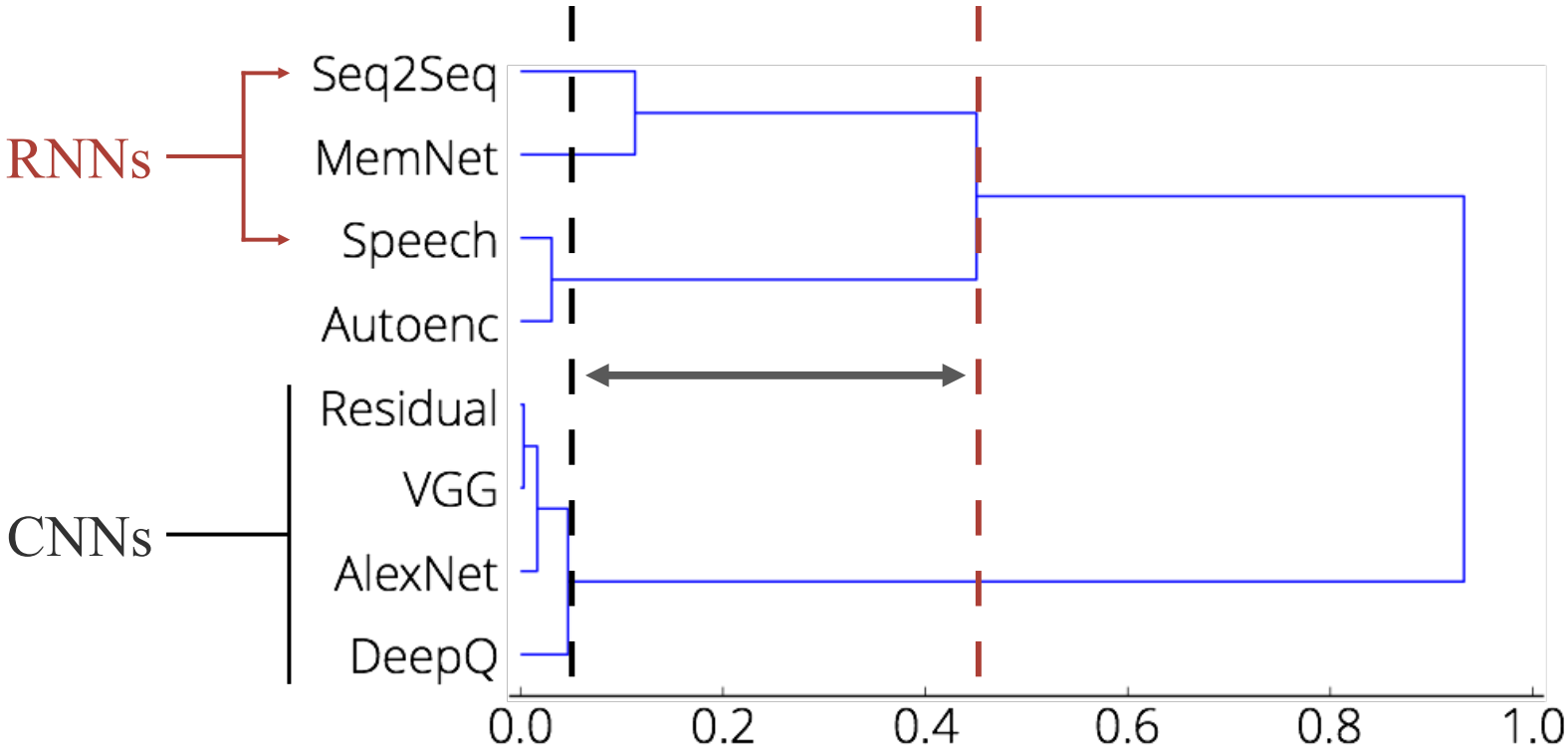
Performance similarity in Fathom

Compute similarity via cosine similarity between op profiles



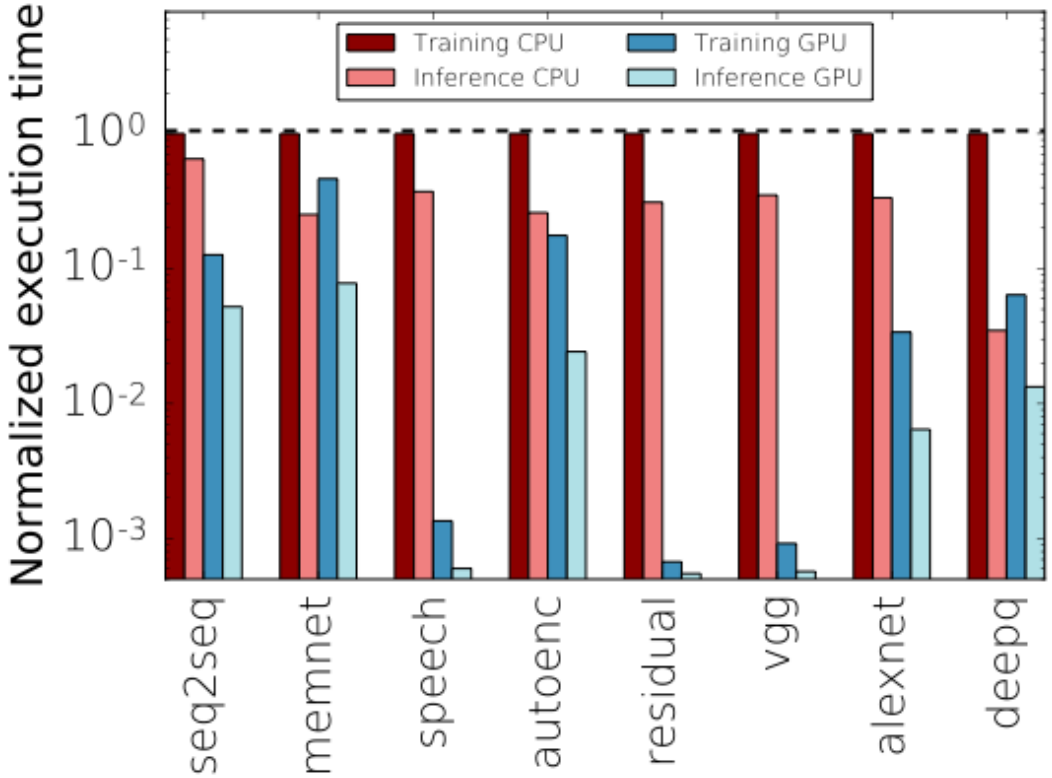
Performance similarity in Fathom

Compute similarity via cosine similarity between op profiles



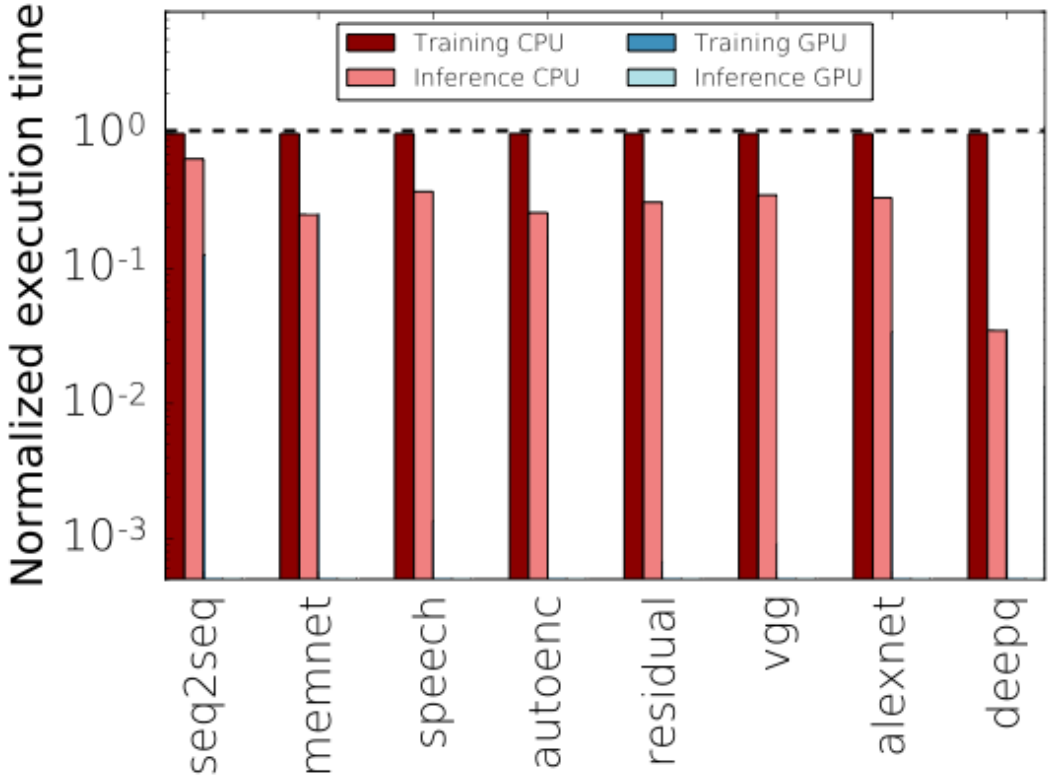
Architecture and mode effects

High-level models make discriminative analysis easy



Architecture and mode effects

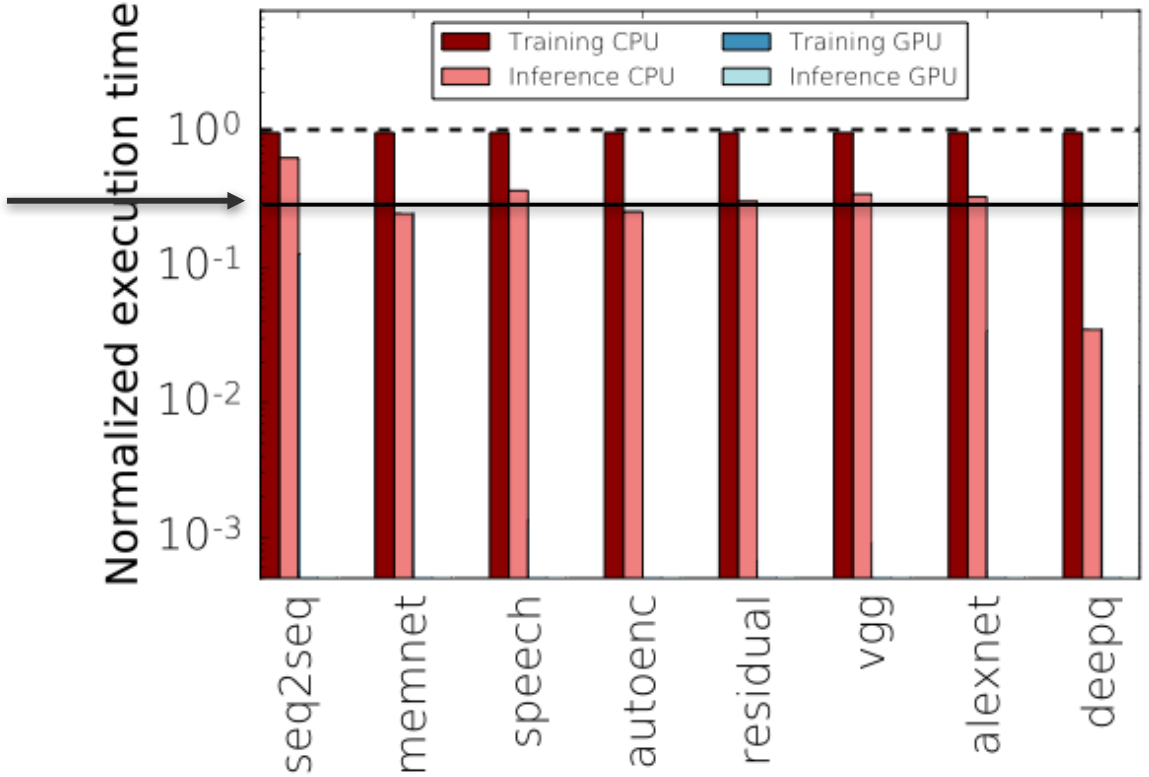
High-level models make discriminative analysis easy



Architecture and mode effects

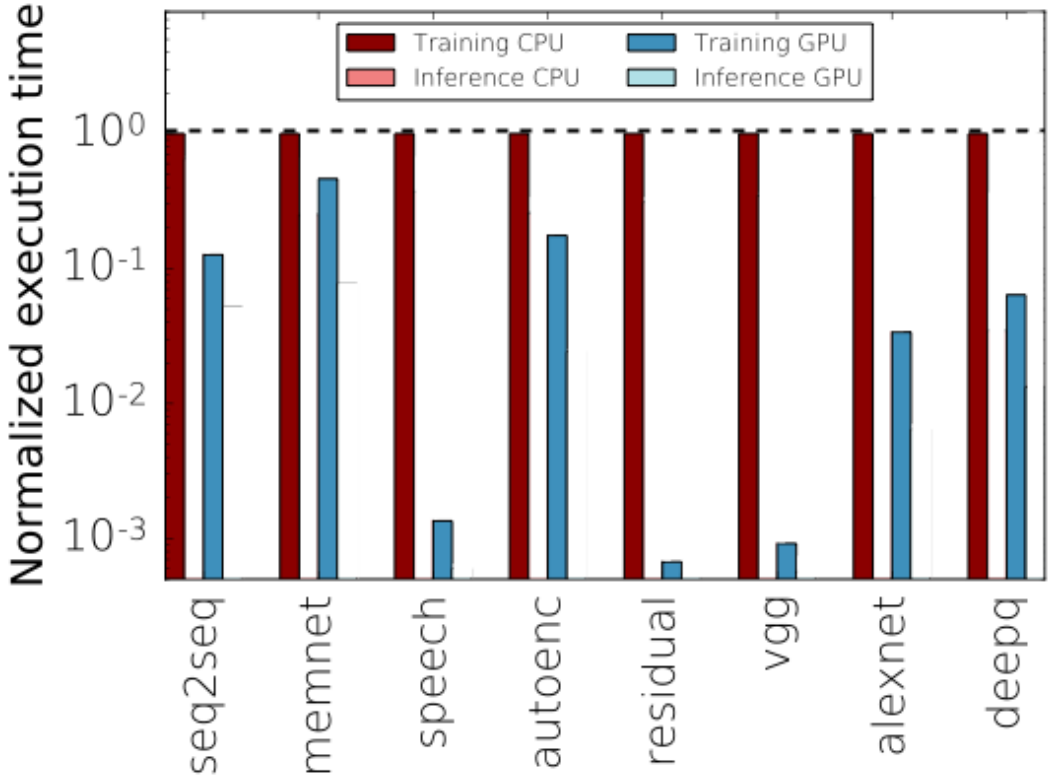
High-level models make discriminative analysis easy

~3x mean speedup



Architecture and mode effects

High-level models make discriminative analysis easy

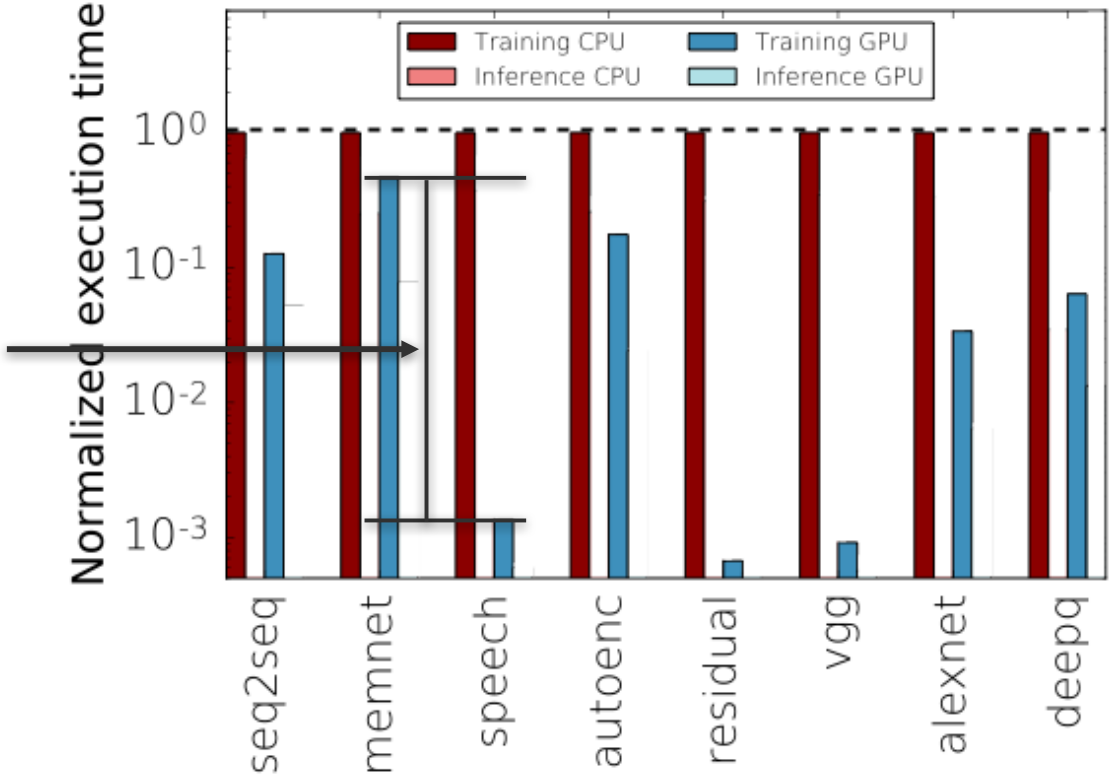


Architecture and mode effects

High-level models make discriminative analysis easy

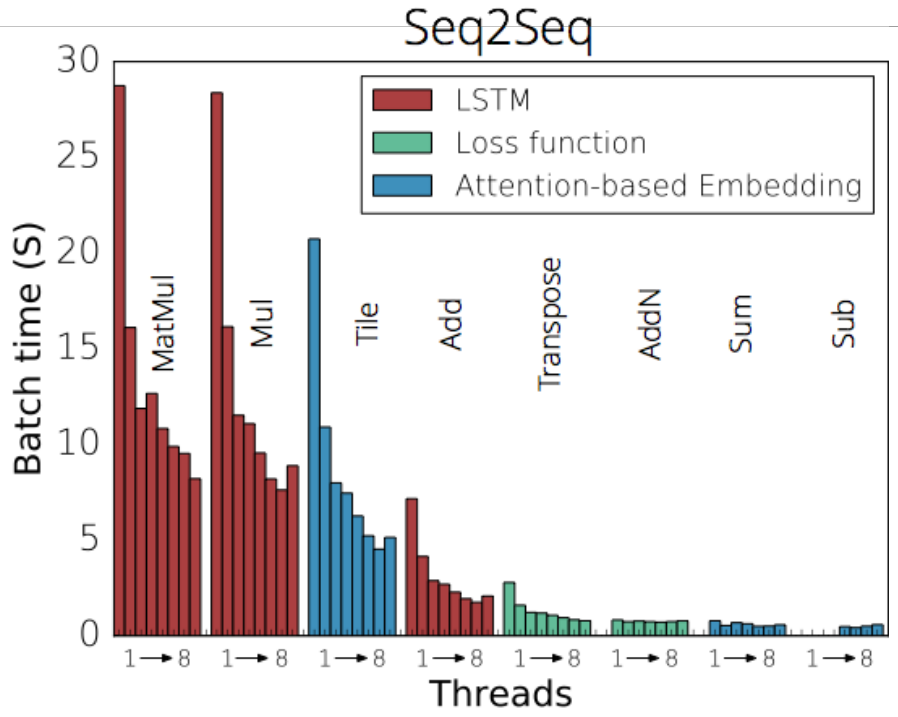
~350x difference in speedup.

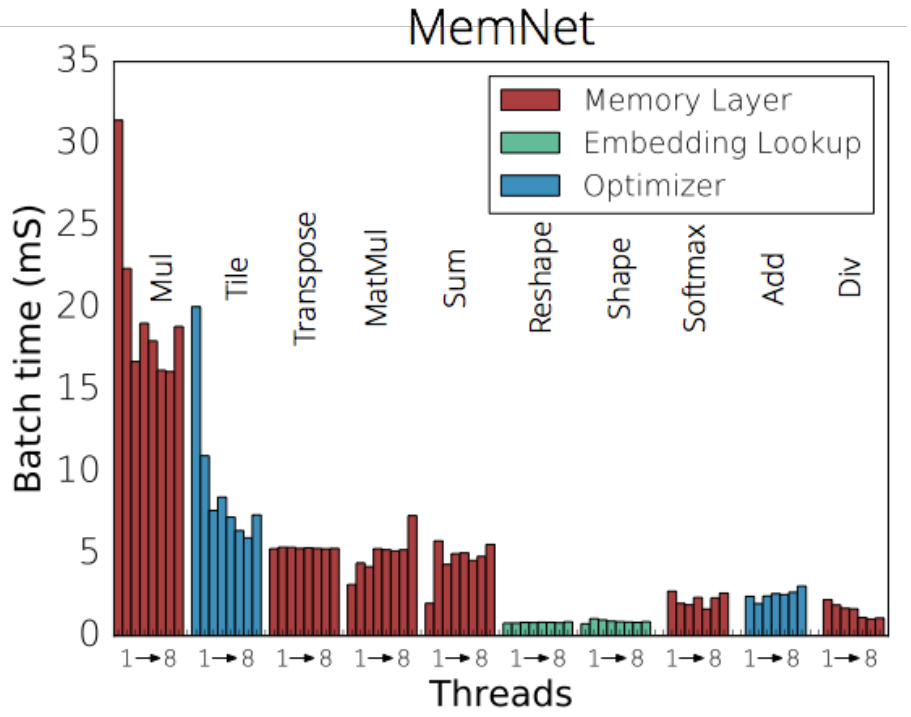
Why?



Parallel scaling

Model-aware analysis can provide causal performance cues

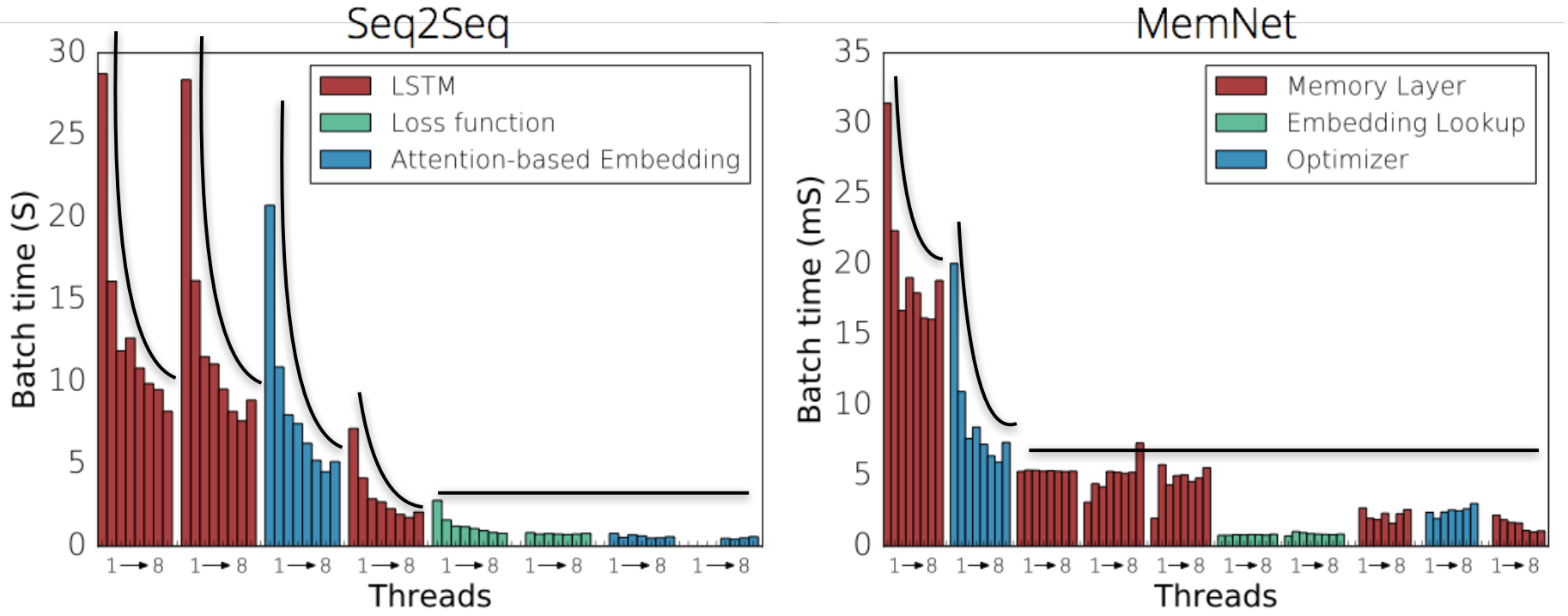




Parallel scaling

Model-aware analysis can provide causal performance cues

Easy to pull out Amdahl's law effects

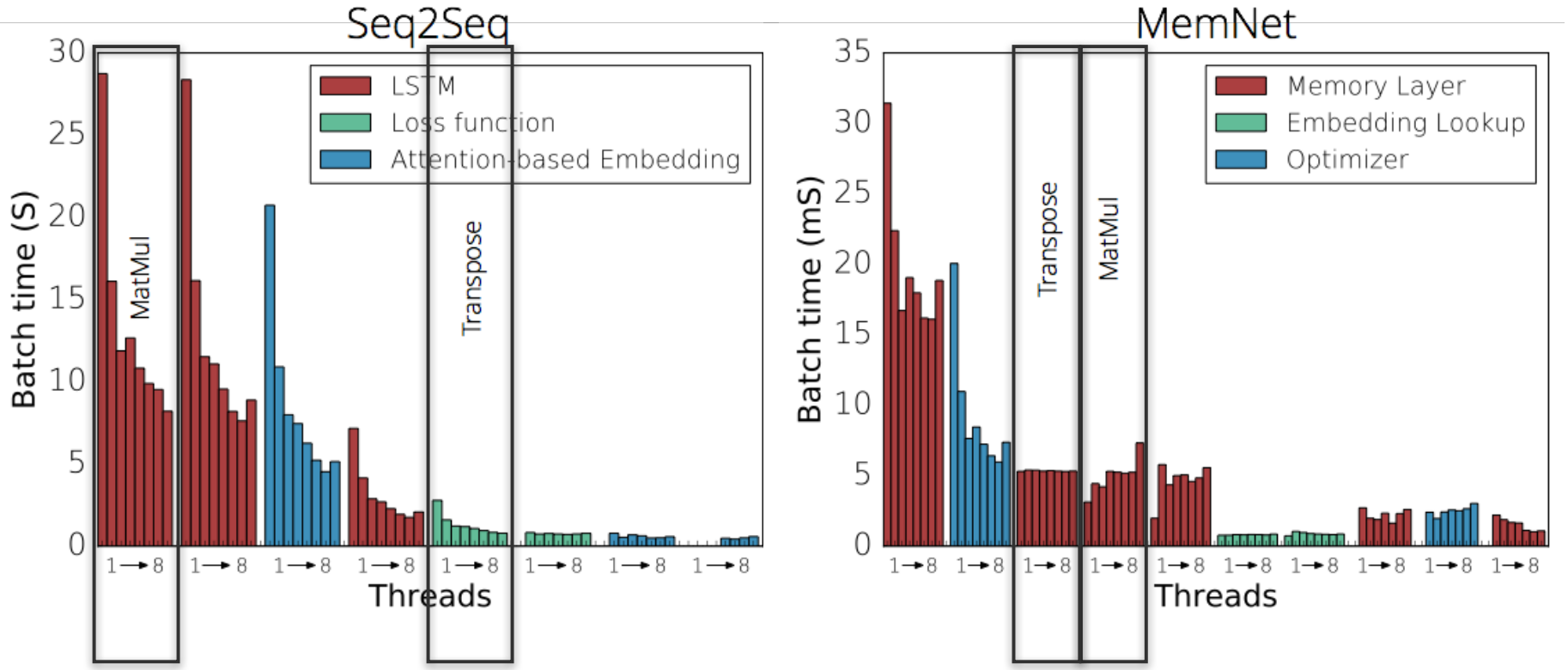


Parallel scaling

Model-aware analysis can provide causal performance cues

Easy to pull out Amdahl's law effects

Can identify differences in operation usage



Fathom is…

…a black-box workload; use it like a benchmark suite.

Top-down bottleneck analysis for a semi-custom processor

Model-aware library performance shootout

Fathom is…

…a black-box workload; use it like a benchmark suite.

Top-down bottleneck analysis for a semi-custom processor

Model-aware library performance shootout

…a performance analysis tool; use it for causal analysis.

Analyze application-level characteristics (e.g., sparsity)

Co-optimize system and learning algorithm tuning knobs

Fathom is…

…a black-box workload; use it like a benchmark suite.

Top-down bottleneck analysis for a semi-custom processor

Model-aware library performance shootout

…a performance analysis tool; use it for causal analysis.

Analyze application-level characteristics (e.g., sparsity)

Co-optimize system and learning algorithm tuning knobs

…a co-simulation tool; use it to augment a simulator

Use Fathom for correctness and behavioral statistics

Feed a validated hardware simulator with these results

A research field in flux

Baidu Research
DeepBench

Primitive kernel
Direct library comparisons
Production-oriented
Commensurability

Primitive	Configurations
GEMM	72
Convolution	36
Recurrent	12+16
All-reduce	25

Fathom
Reference Workloads for Modern Deep Learning

Deep learning models
Whole-system introspection
Research-oriented
Causal understanding

Seq2Seq	Residual
MemNet	VGG
Speech	AlexNet
Autoenc	DeepQ

Fathom

Reference Workloads for Modern Deep Learning

For more information:

IISWC 2016 Paper:

arxiv.org/abs/1608.06581

Code on Github:

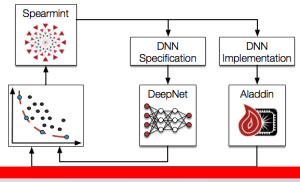
rdadolf.github.io/fathom

Architectural Support for Deep Learning at Harvard

A Full-Stack Approach to Machine Learning

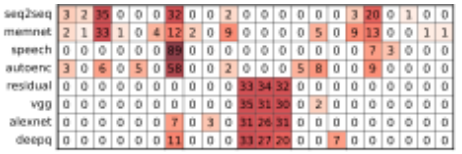
Algorithms

Co-Designing Deep Neural Network Accelerators for Accuracy and Energy Using Bayesian Optimization



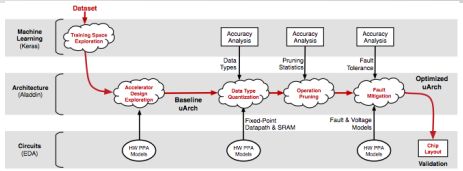
Tools

Fathom: Reference Workloads for Modern Deep Learning Methods



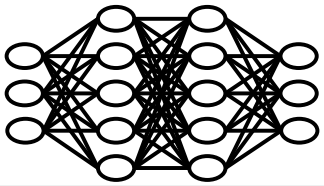
Architectures

Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators



Circuits

SM2: A Deep Neural Network Accelerator SoC in 28nm bulk and 16nm FinFET

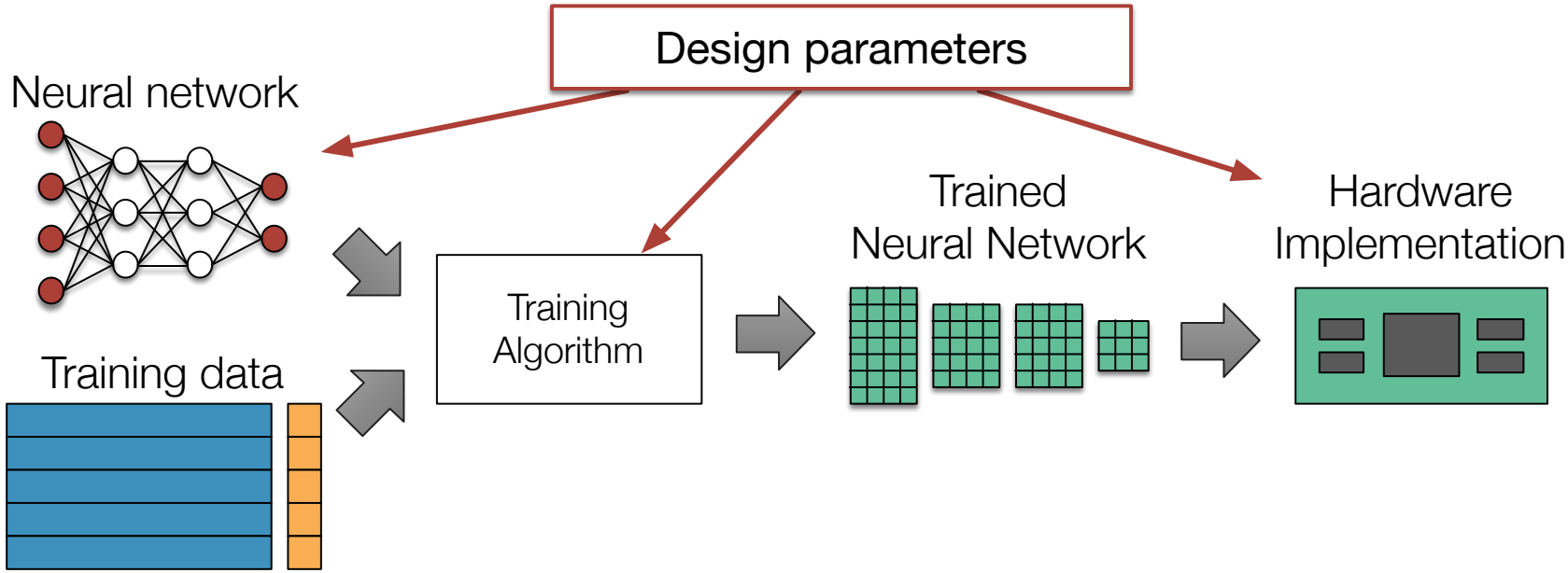


Problem: Hardware accelerator design for DNNs

Goal: build specialized hardware blocks to evaluate DNNs

Example: A speech recognition engine for a mobile phone

Example: An object classifier for an autonomous robot



Problem: Hardware accelerator design for DNNs

High-dimensional design space

- Dozens of different variables, even for basic designs

Complex parameter interactions

- DNNs are notoriously difficult to tune

Multiple competing objectives

- Prediction accuracy vs. energy consumption

Costly evaluation functions

- DNN training and hardware simulations both require hours

Bayesian Optimization

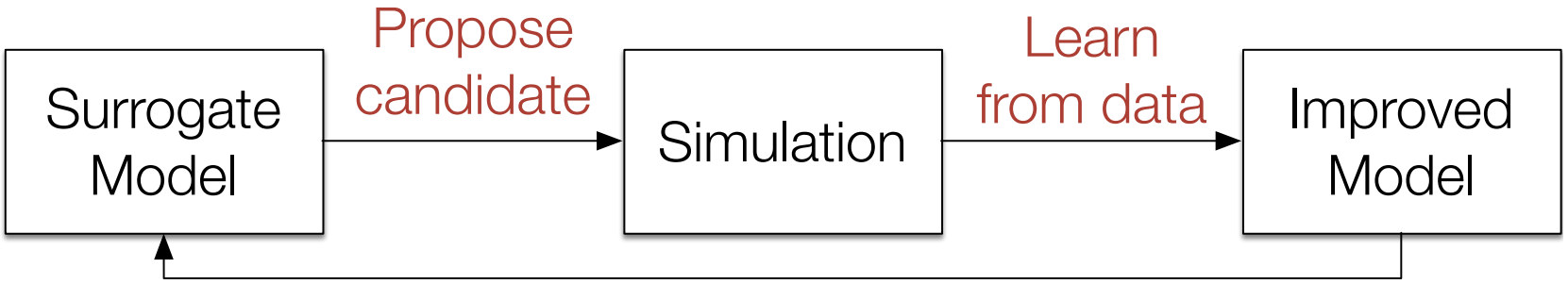
Build a rough statistical model of the optimization space

This “surrogate model” must be cheap to evaluate

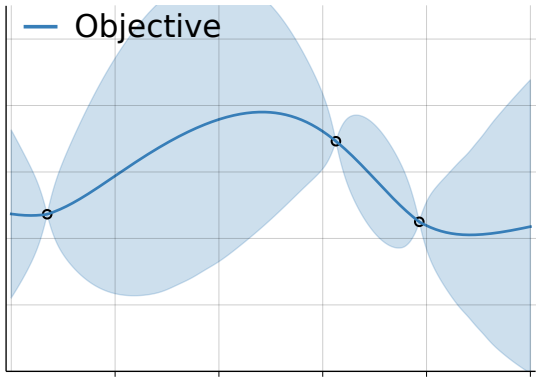
Use it to choose candidate parameter configuration

Balance tweaking good designs and avoiding local optima

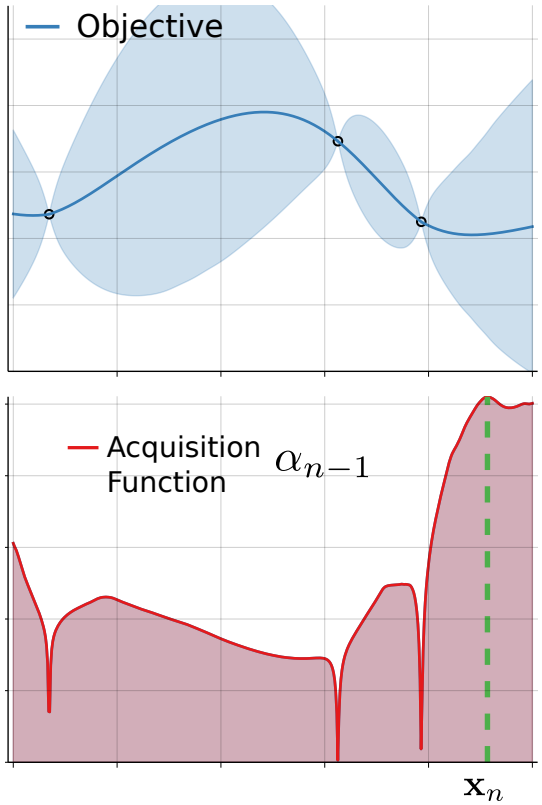
Improve the model as more data is collected



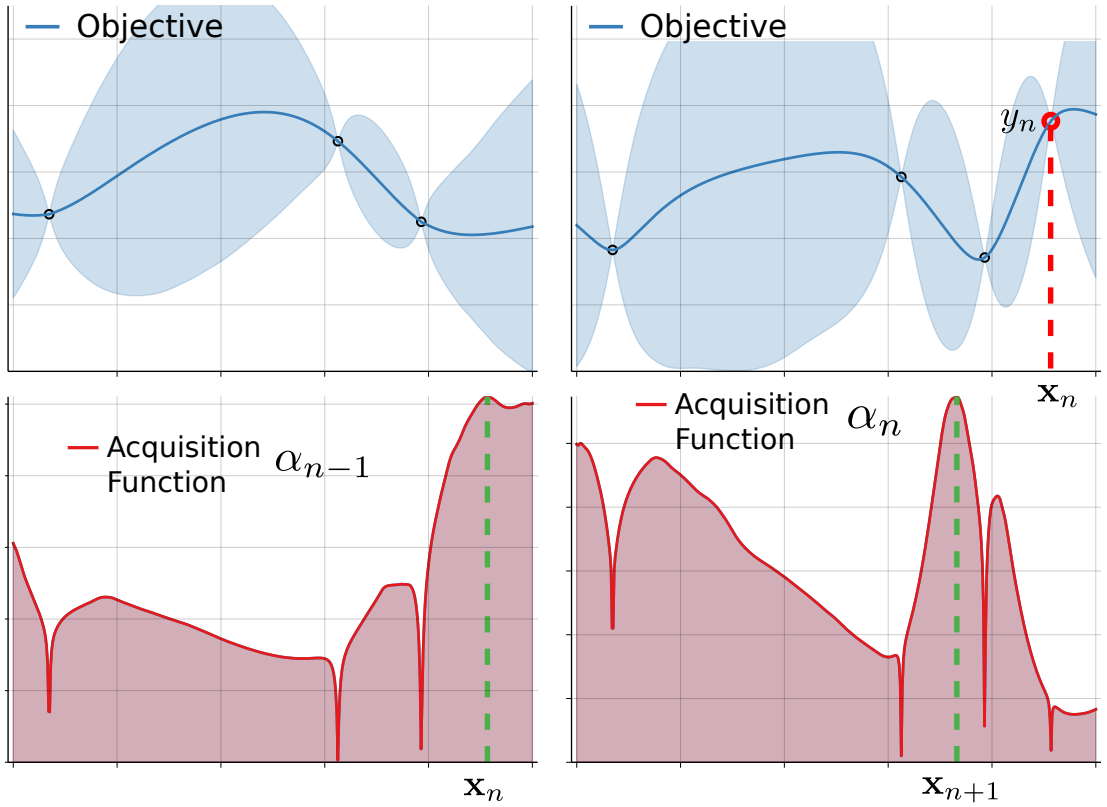
Single-objective, one-dimensional example



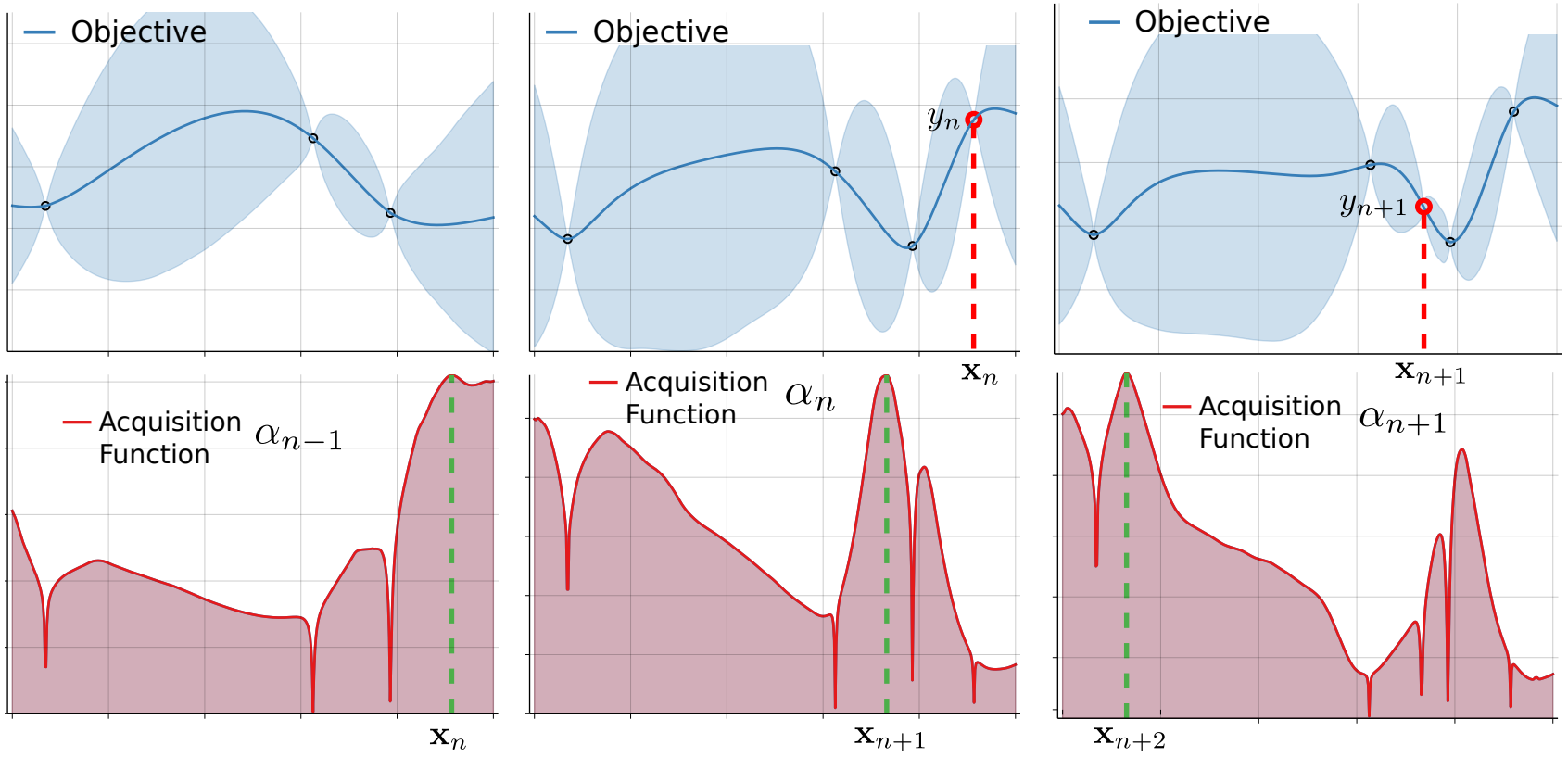
Single-objective, one-dimensional example



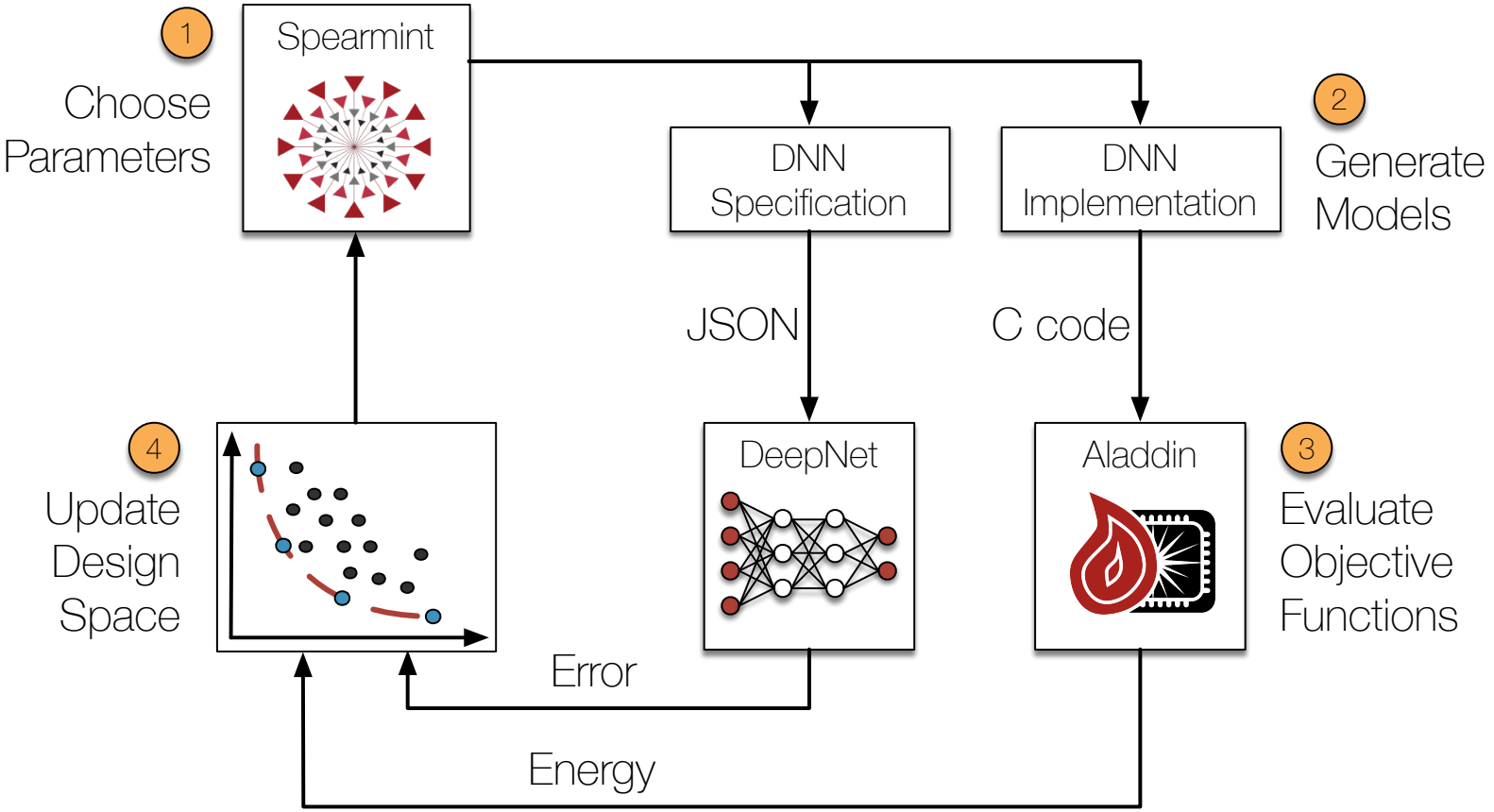
Single-objective, one-dimensional example



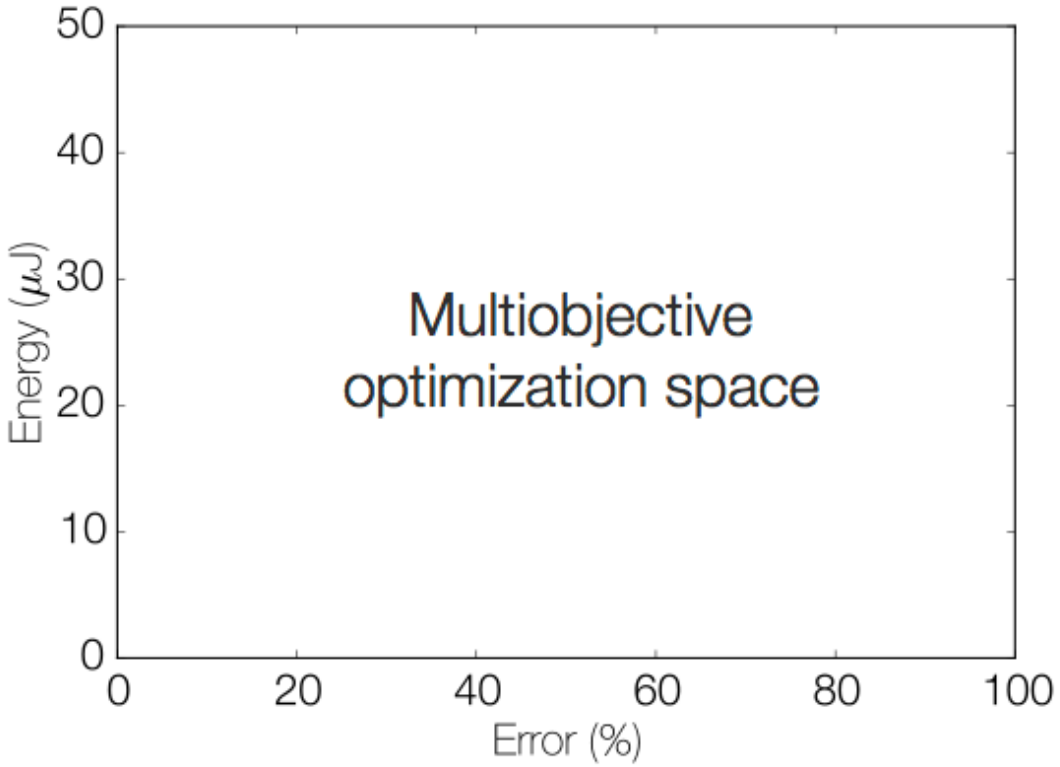
Single-objective, one-dimensional example



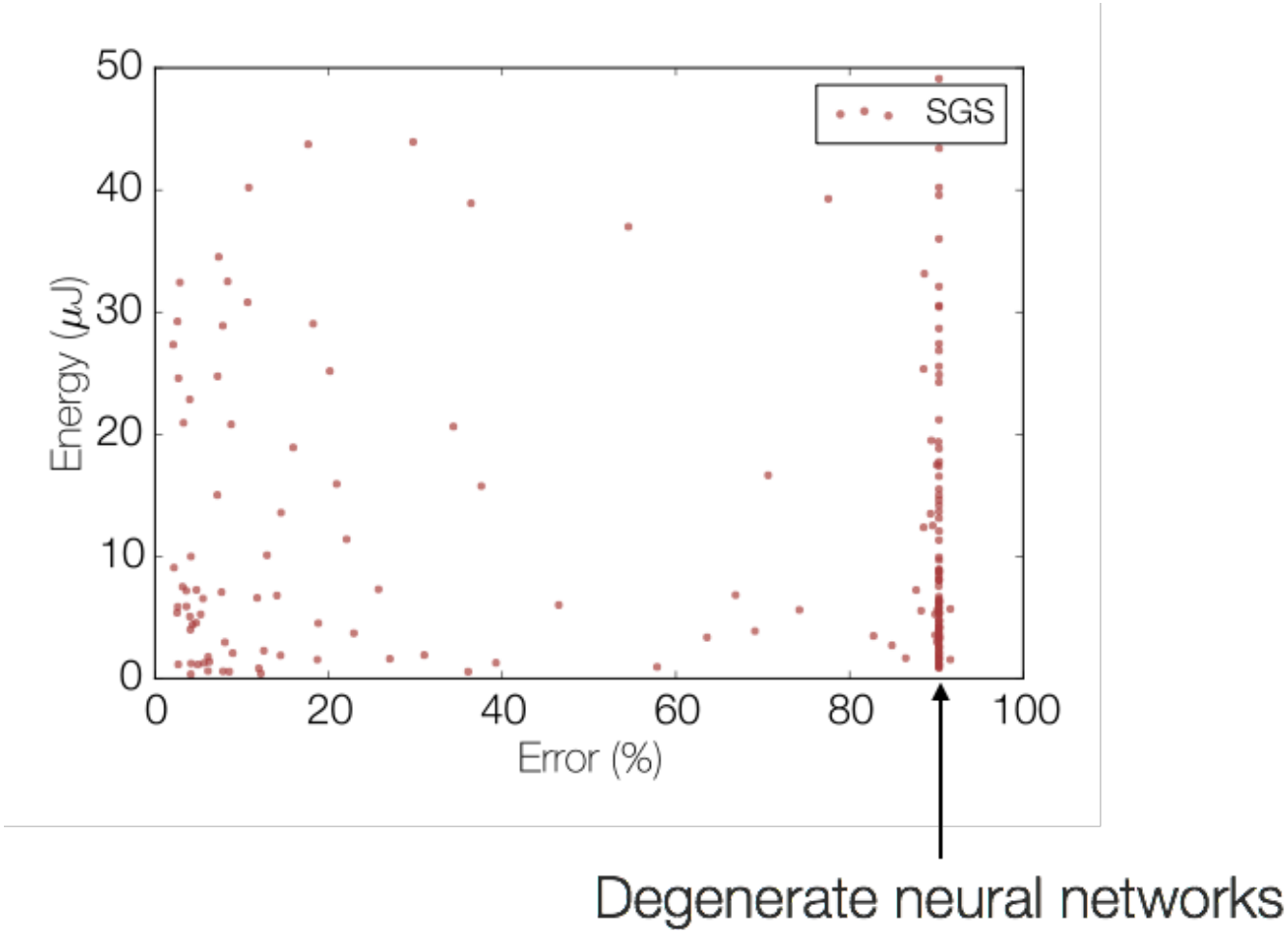
Co-designing deep neural network accelerators



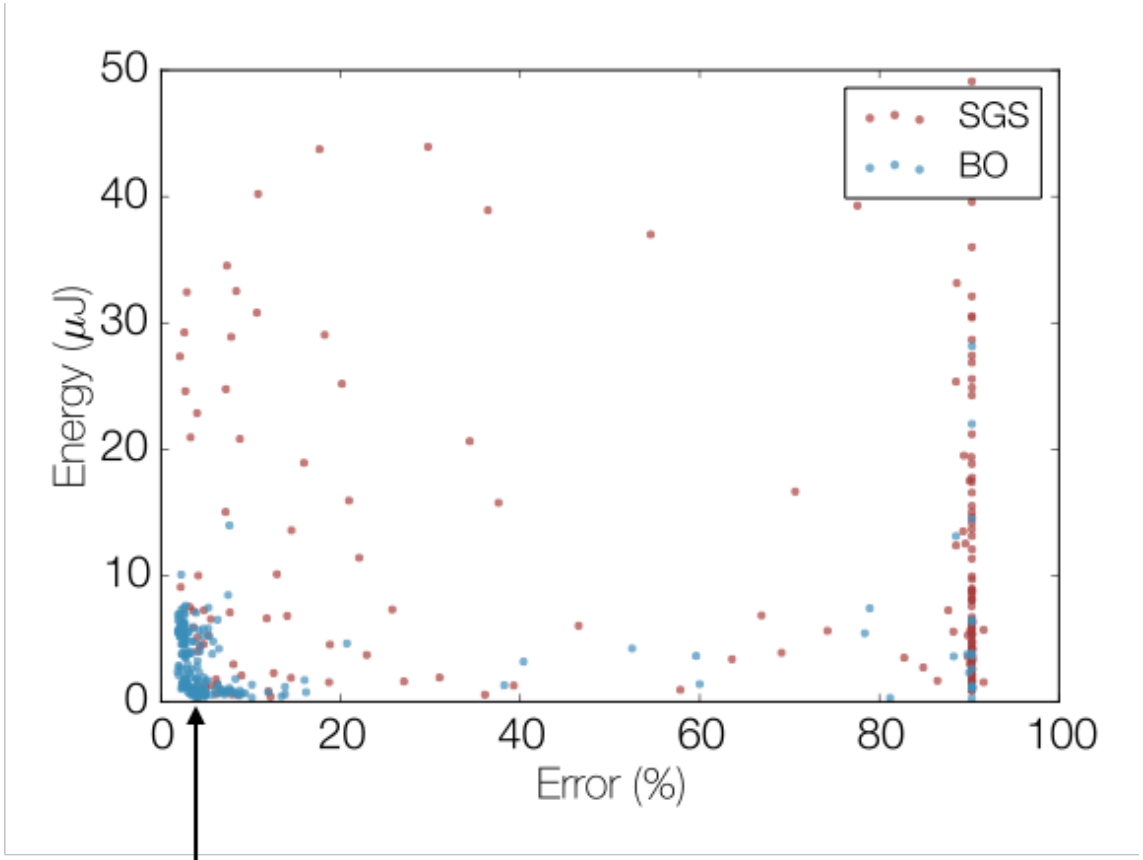
Bayesian optimization finds better designs on average



Bayesian optimization finds better designs on average

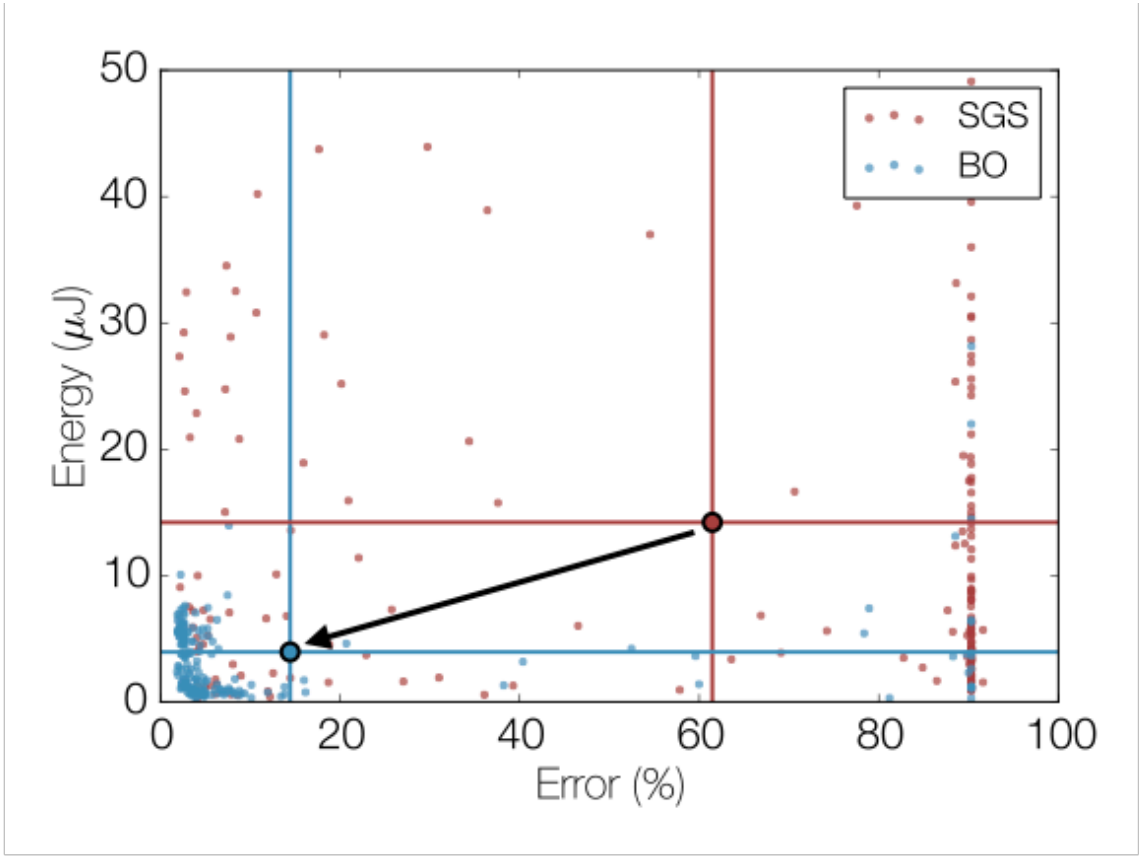


Bayesian optimization finds better designs on average



Cluster of high-quality designs

Bayesian optimization finds better designs on average

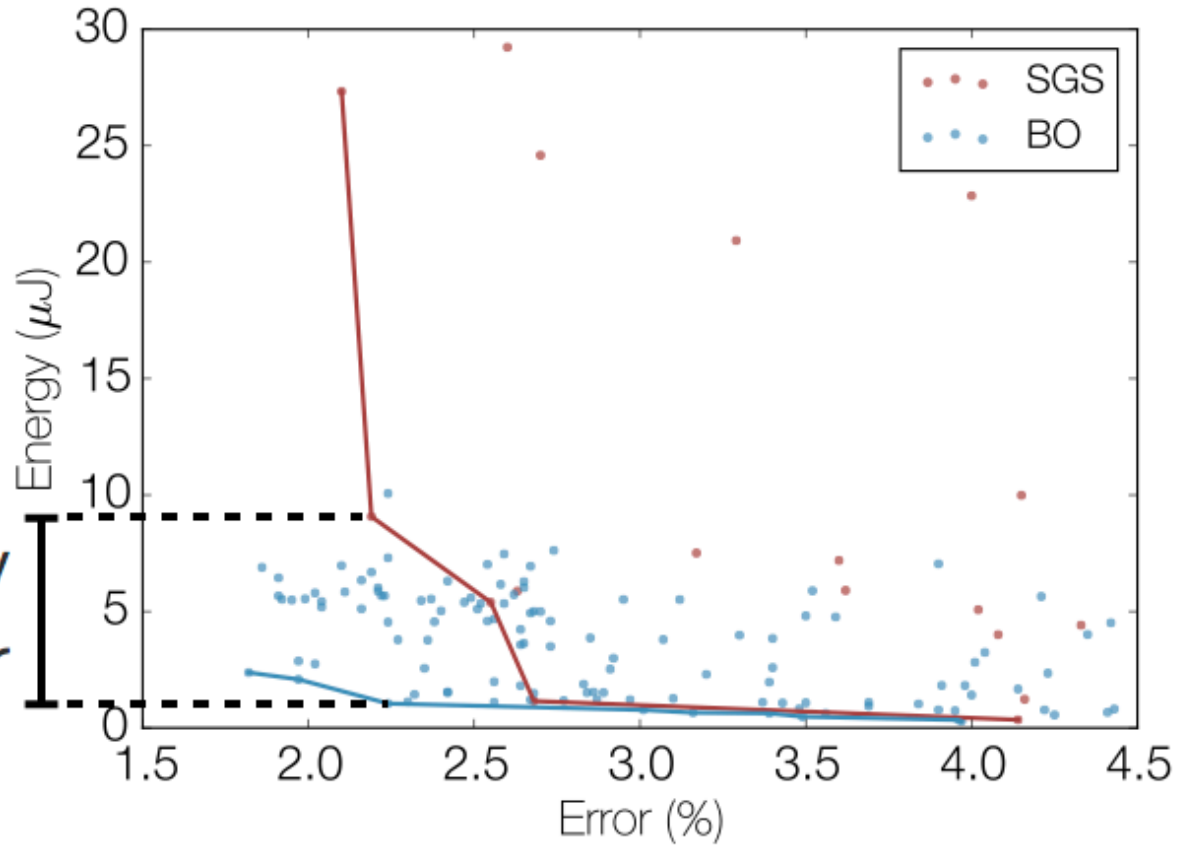


4.3x less error

3.6x less energy

Bayesian optimization finds better designs on average

8.7x less energy
for identical error

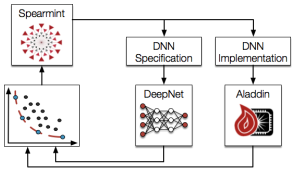


Architectural Support for Deep Learning at Harvard

A Full-Stack Approach to Machine Learning

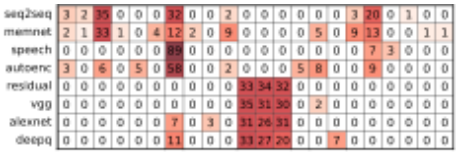
Algorithms

Co-Designing Deep Neural Network Accelerators for Accuracy and Energy Using Bayesian Optimization



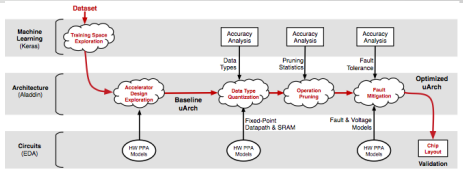
Tools

Fathom: Reference Workloads for Modern Deep Learning Methods



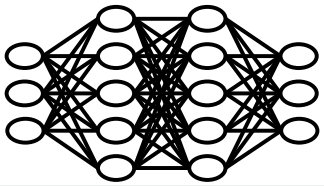
Architectures

Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators



Circuits

SM2: A Deep Neural Network Accelerator SoC in 28nm bulk and 16nm FinFET



Questions and acknowledgments

Brandon Reagen

Bob Adolf

Saketh Rama

- Papers/Software: vlsiarch.eecs.harvard.edu
- Prof. Ryan Adams and Prof. Miguel Hernandez-Lobato for Bayesian Optimization collaboration