SUMMIT

Tel Aviv

Optimize your Machine Learning workloads

Julien Simon
Global Evangelist, AI & Machine Learning
@julsimon

aws SUMMIT
Tel Aviv

Our mission at AWS

Put machine learning in the hands of every developer

Now let's make it as fast, efficient and unexpensive as possible

aws SUMMIT
Tel Aviv

Optimizing Infrastructure and Frameworks

aws SUMMIT
Tel Aviv

Amazon EC2 P3dn

https://aws.amazon.com/blogs/aws/new-ec2-p3dn-gpu-instances-with-100-gbps-networking-local-nvme-storage-for-faster-machine-learning-p3-price-reduction/

Reduce machine learning training time

Better GPU utilization

Support larger, more complex models

KEY FEATURES

100Gbps of networking bandwidth

8 NVIDIA Tesla V100 GPUs

32GB of memory per GPU (2x more P3)

96 Intel Skylake vCPUs (50% more than P3) with AVX-512

Amazon EC2 C5n

https://aws.amazon.com/blogs/aws/new-c5n-instances-with-100-gbps-networking/

Intel Xeon Platinum 8000

Up to 3.5GHz single core speed

Up to 100Gbit networking

Based on Nitro hypervisor for bare metal-like performance

Instance Name	vCPUs	RAM	EBS Bandwidth	Network Bandwidth
c5n.large	2	5.25 GiB	Up to 3.5 Gbps	Up to 25 Gbps
c5n.xlarge	4	10.5 GiB	Up to 3.5 Gbps	Up to 25 Gbps
c5n.2xlarge	8	21 GiB	Up to 3.5 Gbps	Up to 25 Gbps
c5n.4xlarge	16	42 GiB	3.5 Gbps	Up to 25 Gbps
c5n.9xlarge	36	96 GiB	7 Gbps	50 Gbps
c5n.18xlarge	72	192 GiB	14 Gbps	100 Gbps

Making TensorFlow faster

Training a ResNet-50 benchmark with the synthetic ImageNet dataset using our optimized build of TensorFlow 1.11 on a c5.18xlarge instance type is **11x faster** than training on the stock binaries.

https://aws.amazon.com/about-aws/whats-new/2018/10/chainer4-4_theano_1-0-2_launch_deep_learning_ami/
October 2018

Available with Amazon SageMaker and
the AWS Deep Learning AMIs

Scaling TensorFlow near-linearly to 256 GPUs

https://aws.amazon.com/about-aws/whats-new/2018/11/tensorflow-scalability-to-256-gpus/

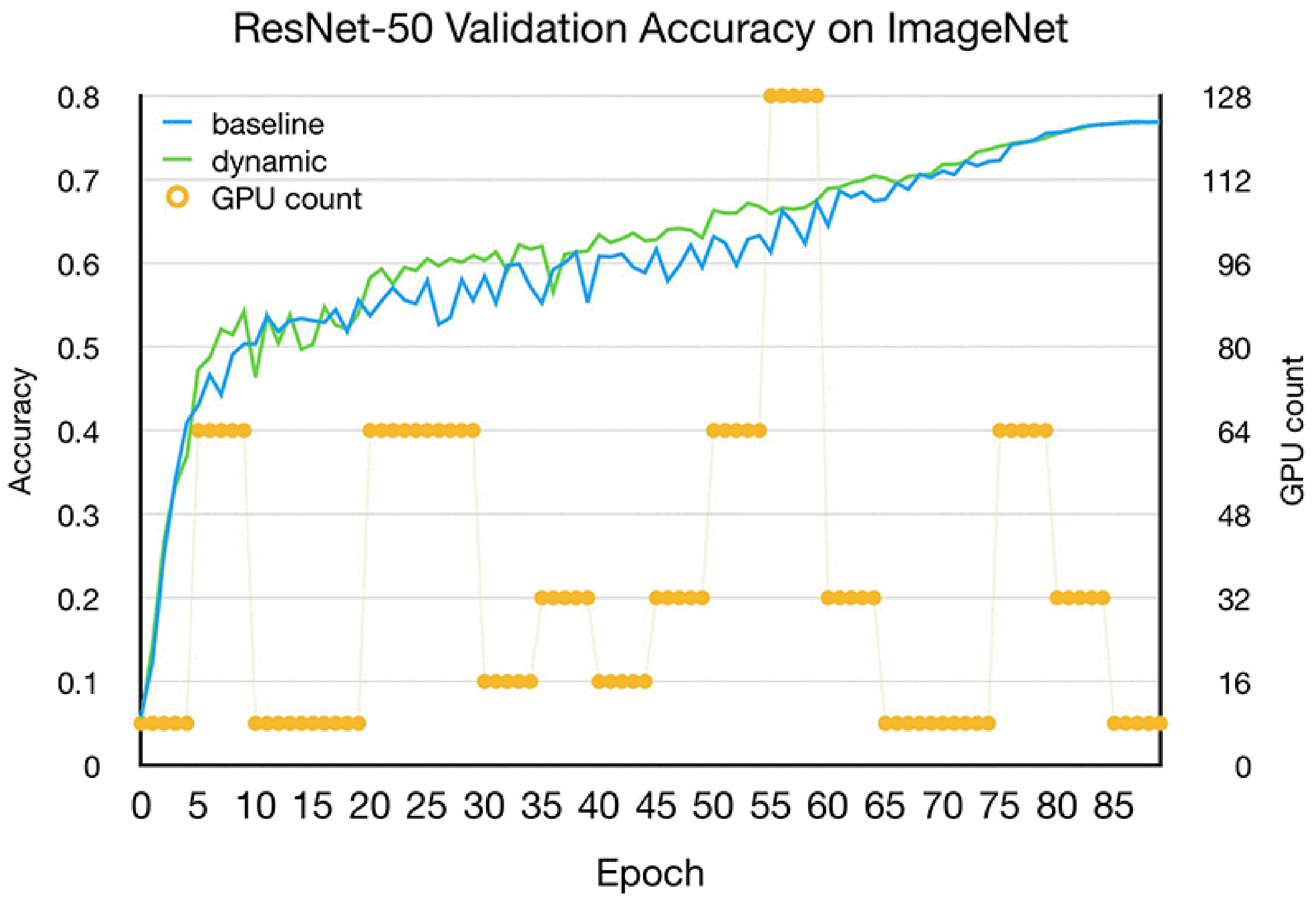
Stock TensorFlow		AWS-Optimized TensorFlow	
65% scaling efficiency with 256 GPUs	→	**90%** scaling efficiency with 256 GPUs	Available with Amazon SageMaker and the AWS Deep Learning AMIs
30m training time	→	**14m** training time	

Dynamic training with Apache MXNet

https://aws.amazon.com/blogs/machine-learning/introducing-dynamic-training-for-deep-learning-with-amazon-ec2/

Use a variable number of instances for distributed training

No loss of accuracy



aws SUMMIT Tel Aviv

Optimizing Models

- Automatic Model Tuning
- Model compilation
- Model compression

Examples of hyperparameters

Decision Trees

Tree depth
Max leaf nodes
Gamma
Eta
Lambda
Alpha
…

Neural Networks

Number of layers
Hidden layer width
Learning rate
Embedding dimensions
Dropout
…

aws SUMMIT Tel Aviv

Automatic Model Tuning

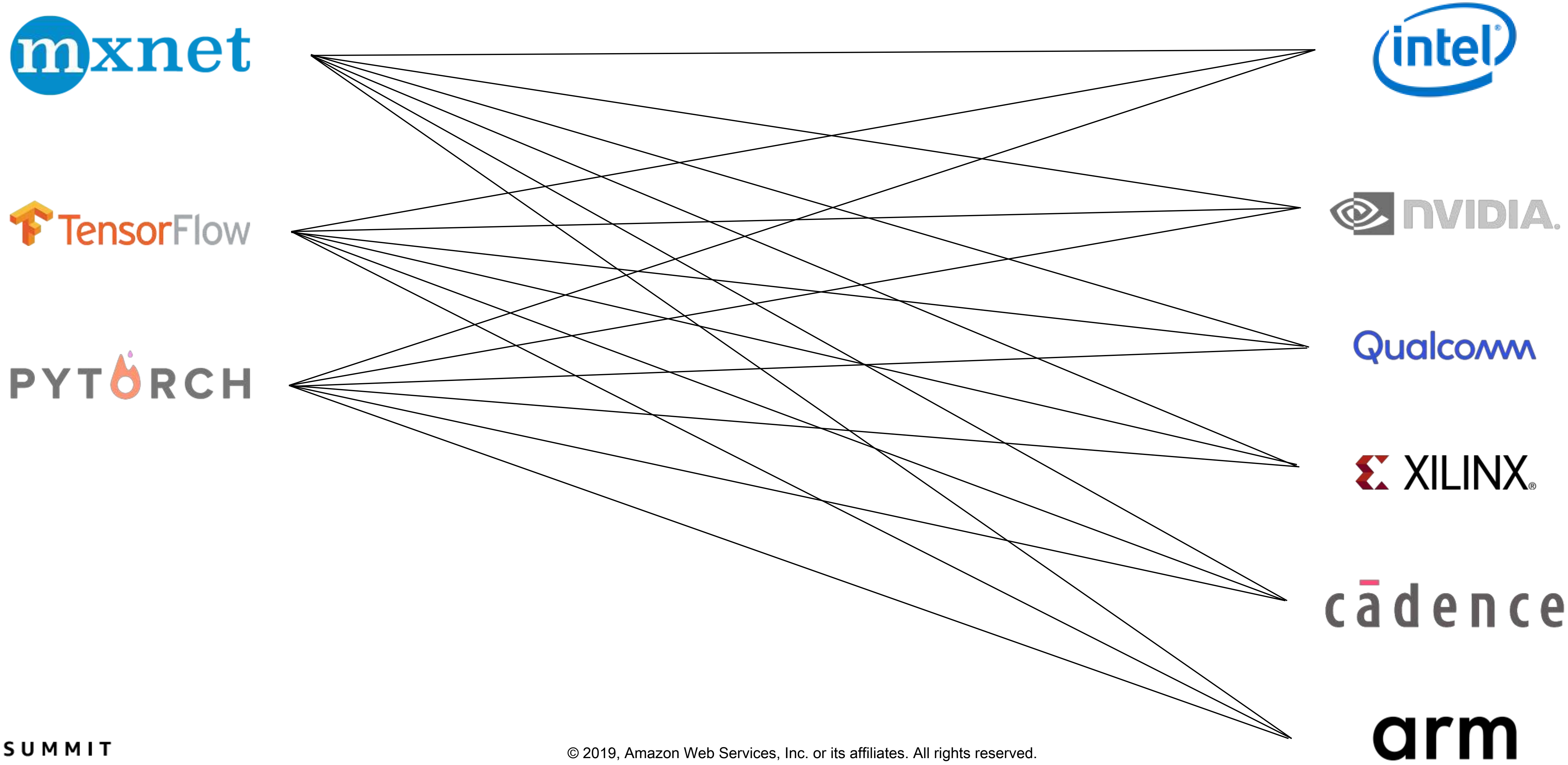
Finding the optimal set of hyper parameters

1. Manual Search ("I know what I'm doing")
2. Grid Search ("X marks the spot")
   Typically training hundreds of models
   Slow and expensive
3. Random Search ("Spray and pray")
   Works better and faster than Grid Search
   But… but… but… it's random!
4. HPO: use Machine Learning
   - Training fewer models
   - Gaussian Process Regression and Bayesian Optimization
   - You can now resume from a previous tuning job

Demo

aws SUMMIT Tel Aviv

Hardware optimization is extremely complex



Amazon Neo: compiling models

https://aws.amazon.com/blogs/aws/amazon-sagemaker-neo-train-your-machine-learning-models-once-run-them-anywhere/

- Train once, run anywhere
- Frameworks and algorithms
  - TensorFlow, Apache MXNet, PyTorch, ONNX, and XGBoost
- Hardware architectures
  - ARM, Intel, and NVIDIA starting today
  - Cadence, Qualcomm, and Xilinx hardware coming soon
- Amazon SageMaker Neo is open source, enabling hardware vendors to customize it for their processors and devices: https://github.com/neo-ai

Demo:
Compiling ResNet-50 for the Raspberry Pi

Configure the compilation job

```
{
  "RoleArn":$ROLE_ARN,
  "InputConfig": {
   "S3Uri":"s3://jsimon-neo/model.tar.gz",
   "DataInputConfig": "{\"data\": [1, 3, 224, 224]}",
   "Framework": "MXNET"
  },
 "OutputConfig": {
  "S3OutputLocation": "s3://jsimon-neo/",
  "TargetDevice": "rasp3b"
 },
 "StoppingCondition": {
 "MaxRuntimeInSeconds": 300
 }
}
```

Compile the model

```
$ aws sagemaker create-compilation-job
--cli-input-json file://config.json
--compilation-job-name resnet50-mxnet-pi

$ aws s3 cp s3://jsimon-neo/model-
rasp3b.tar.gz .

$ gtar tfz model-rasp3b.tar.gz
compiled.params
compiled_model.json
compiled.so
```

Predict with the compiled model

```
from dlr import DLRModel
model = DLRModel('resnet50', input_shape,
output_shape, device)
out = model.run(input_data)
```

aws SUMMIT Tel Aviv

Demo

Compressing deep learning models

- Compression is the process of reducing the size of a trained network, either by removing certain layers or by shrinking layers, while maintaining accuracy.

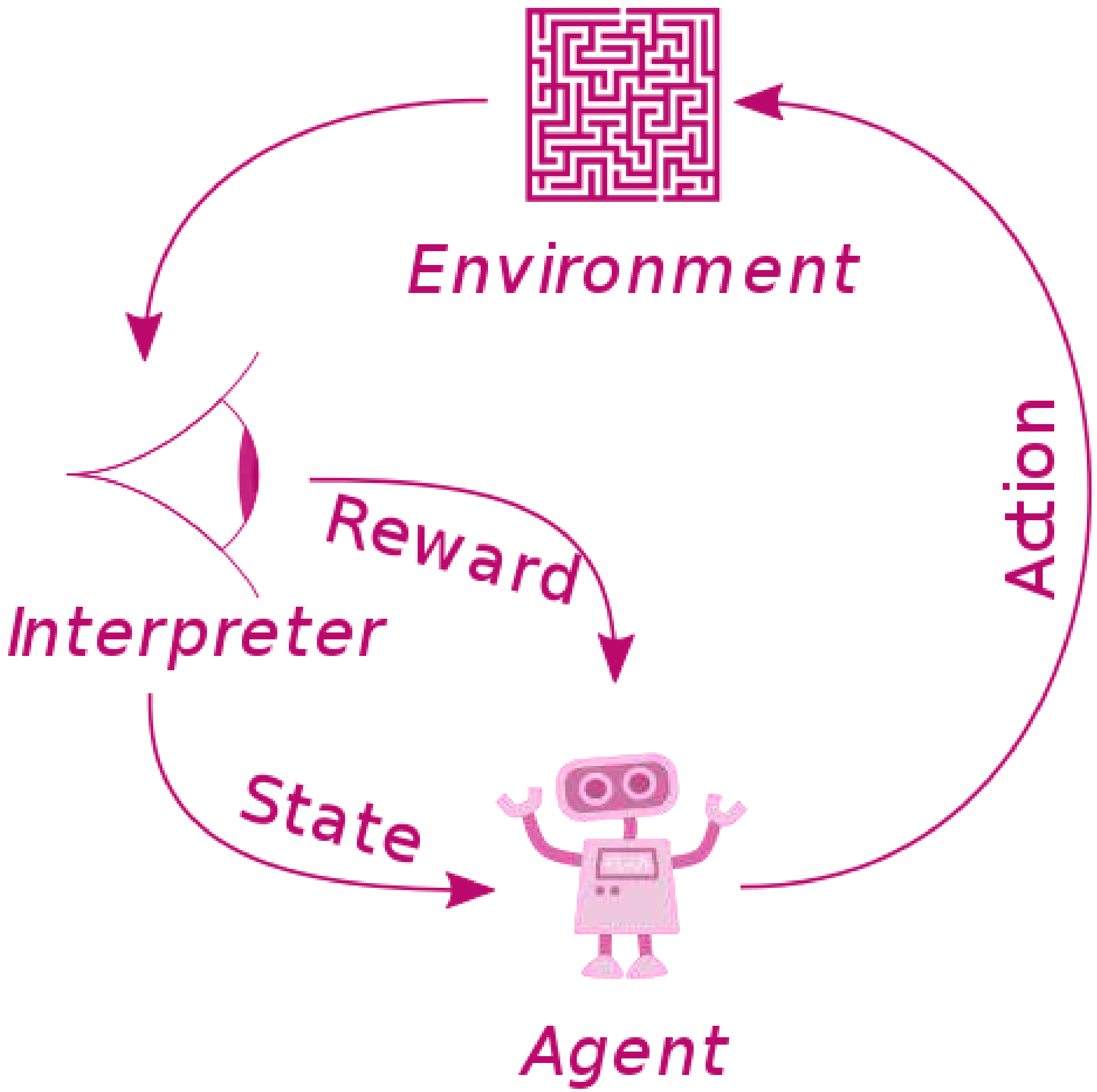
- A smaller model will predict faster and require less memory.

- The number of possible combinations makes is difficult to perform this task manually, or even programmatically.

- Reinforcement learning to the rescue!

Defining the problem

- Objective: find the smallest possible network architecture from a pre-trained network architecture, while producing the best accuracy.
- Environment: a custom developed environment that accepts a Boolean array of layers to remove from the RL agent and produces an observation describing layers.
- State: the layers.
- Action: A boolean array one for each layer.
- Reward: a combination of compression ratio and accuracy.

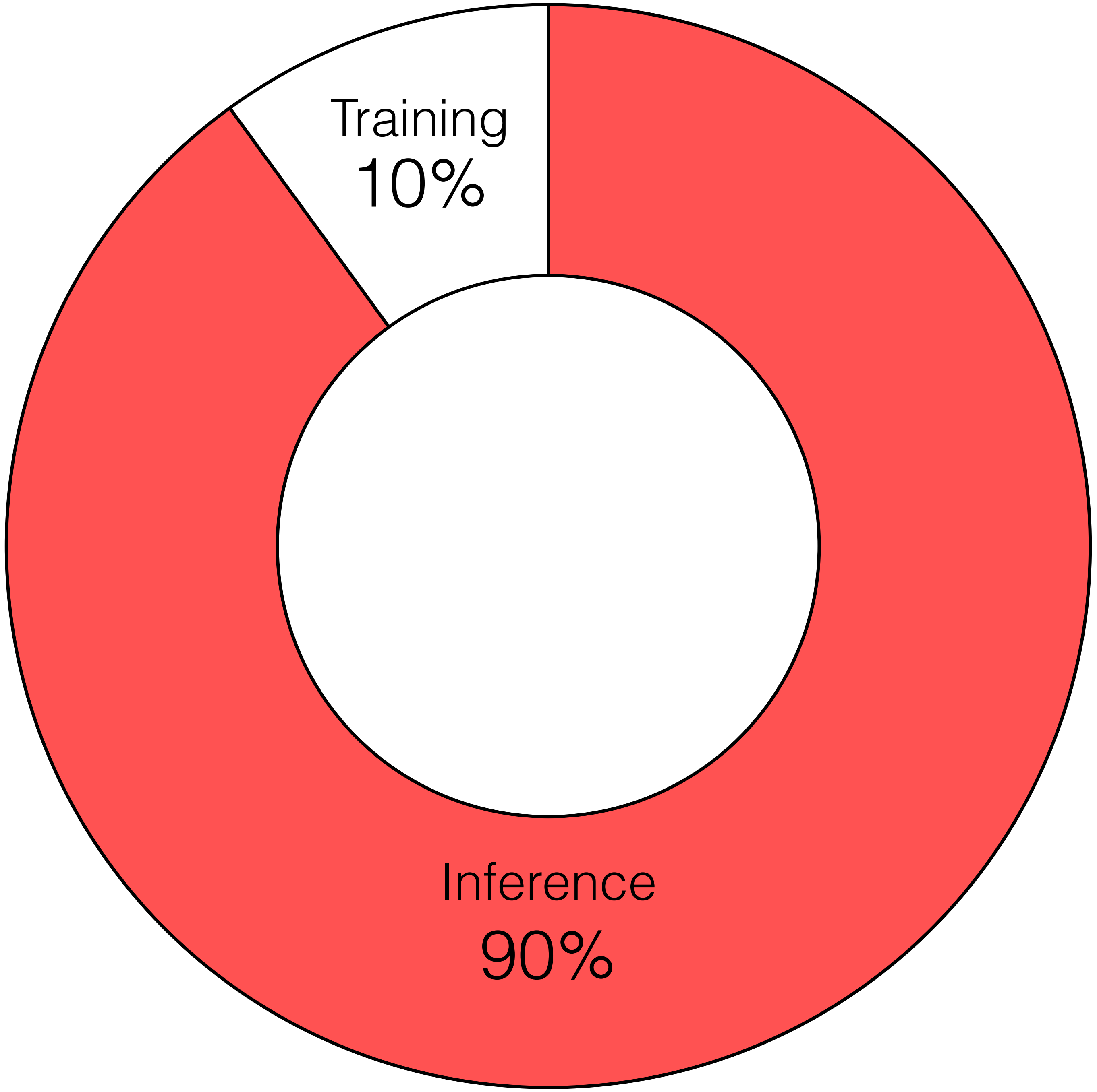


Demo

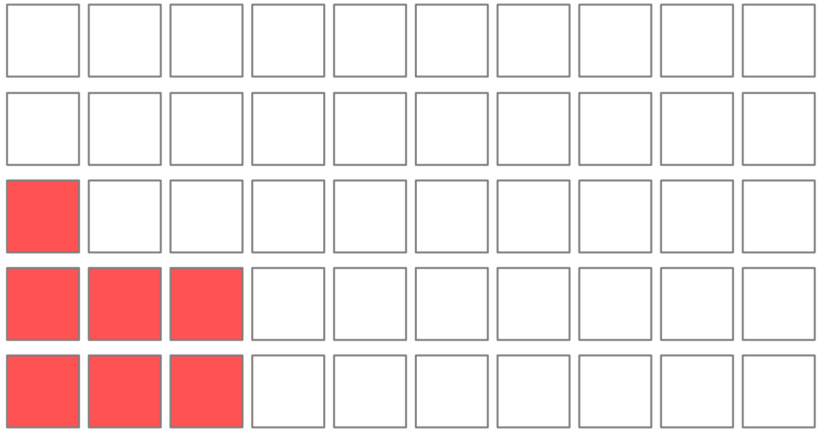
https://github.com/awslabs/amazon-sagemaker-examples/tree/master/reinforcement_learning/rl_network_compression_ray_custom

Optimizing Inference

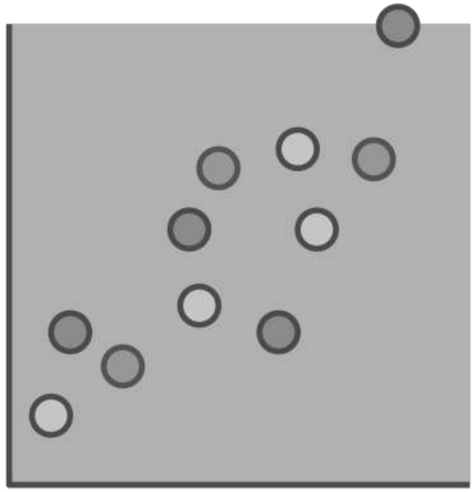
Predictions drive complexity and cost in production



Are you making the most of your infrastructure?



Low utilization and high costs



One size does not fit

Amazon Elastic Inference

https://aws.amazon.com/blogs/aws/amazon-elastic-inference-gpu-powered-deep-learning-inference-acceleration/

Lower inference costs up to 75%

Match capacity to demand

Available between 1 to 32 TFLOPS

KEY FEATURES

Integrated with Amazon EC2, Amazon SageMaker, and Amazon DL AMIs

Support for TensorFlow, Apache MXNet, and ONNX with PyTorch coming soon

Single and mixed-precision operations

Demo

Faster training

Faster inference

Save time and money

<u>No plumbing</u>

Getting started

https://ml.aws

https://aws.amazon.com/sagemaker

https://github.com/awslabs/amazon-sagemaker-examples

https://medium.com/@julsimon

Please complete the session survey.

Thank you!

Julien Simon
Global Evangelist, AI & Machine Learning
@julsimon