File Name: T-ICML-O_4_I1_going_deeper_faster

Format: Presenter in Studio

Title: Going Deeper, Faster

Presenter: Lak Lakshmanan

Google Cloud

Going Deeper, Faster

Lak Lakshmanan

Specialization

End-to-End Machine Learning with TensorFlow on GCP

Production ML Systems

Image Classification Models

Sequence Models

Recommendation Models

Agenda

Introduction

Batch Normalization

Residual Networks

AI Accelerators

Architecture Search

Object identification

Learn how to...

Train deeper, more accurate neural networks faster

Learn how to...

Train deeper, more accurate neural networks faster

Address internal covariate shift using batch normalization

Learn how to...

Train deeper, more accurate neural networks faster

Address internal covariate shift using batch normalization

Add shortcut connections to increase network depth

Learn how to...

Train deeper, more accurate neural networks faster

Address internal covariate shift using batch normalization

Add shortcut connections to increase network depth

Take advantage of Tensor Processing Units

Learn how to...

Train deeper, more accurate neural networks faster

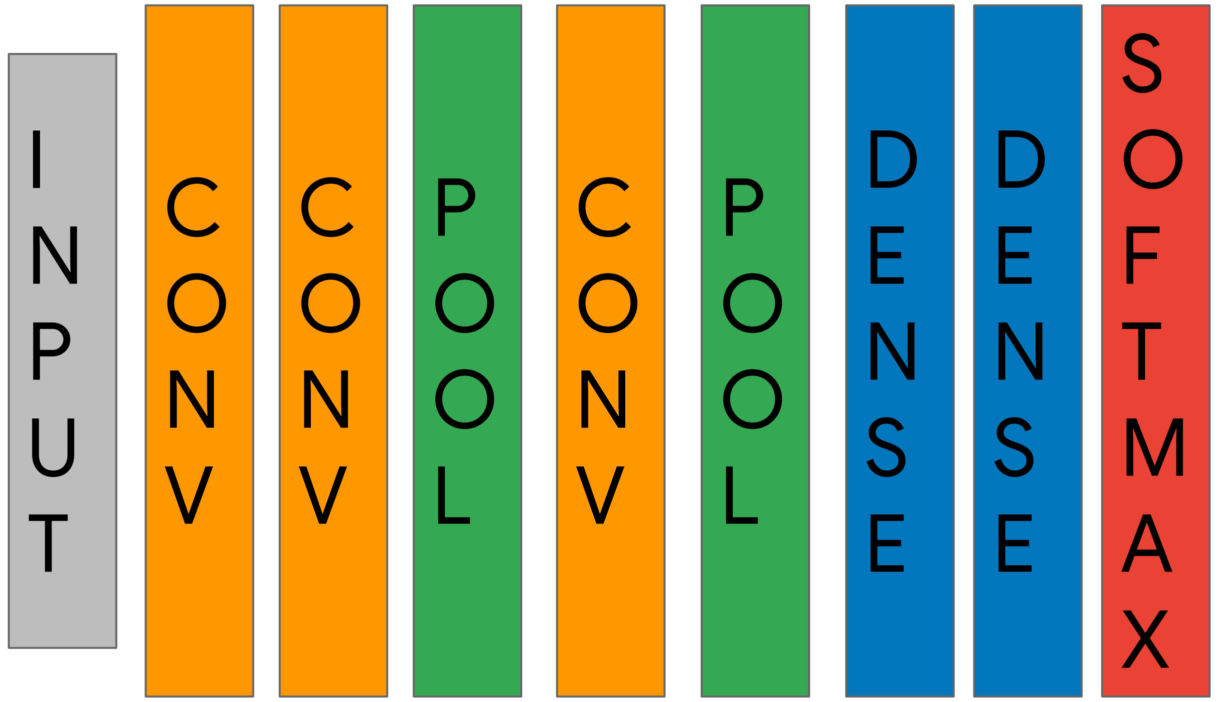
Address internal covariate shift using batch normalization

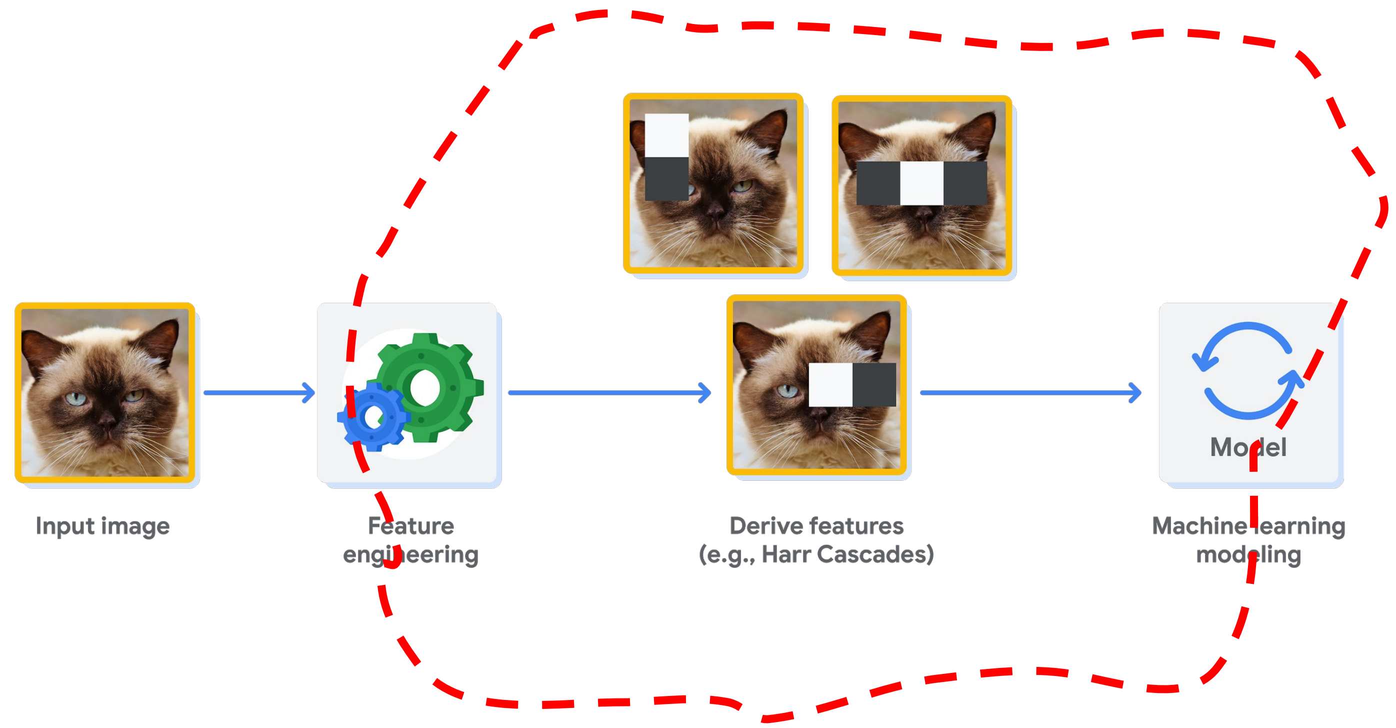
Add shortcut connections to increase network depth

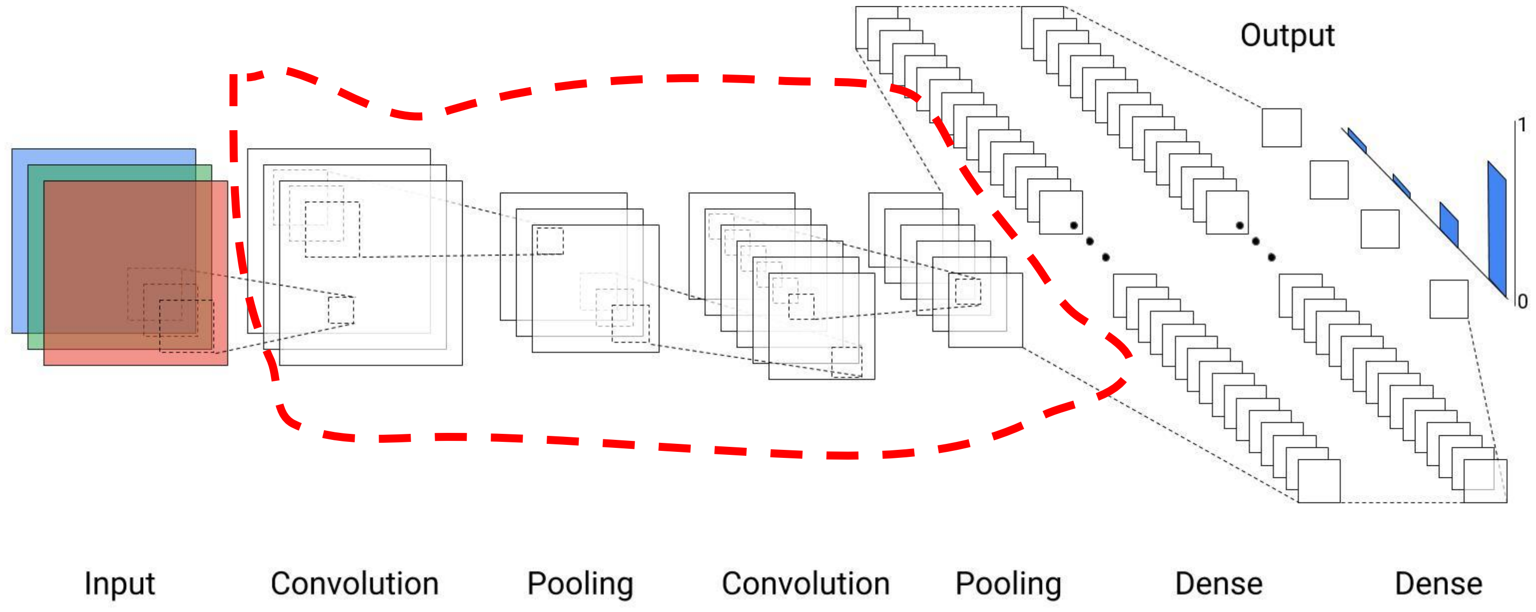
Take advantage of Tensor Processing Units

Automate Network Design

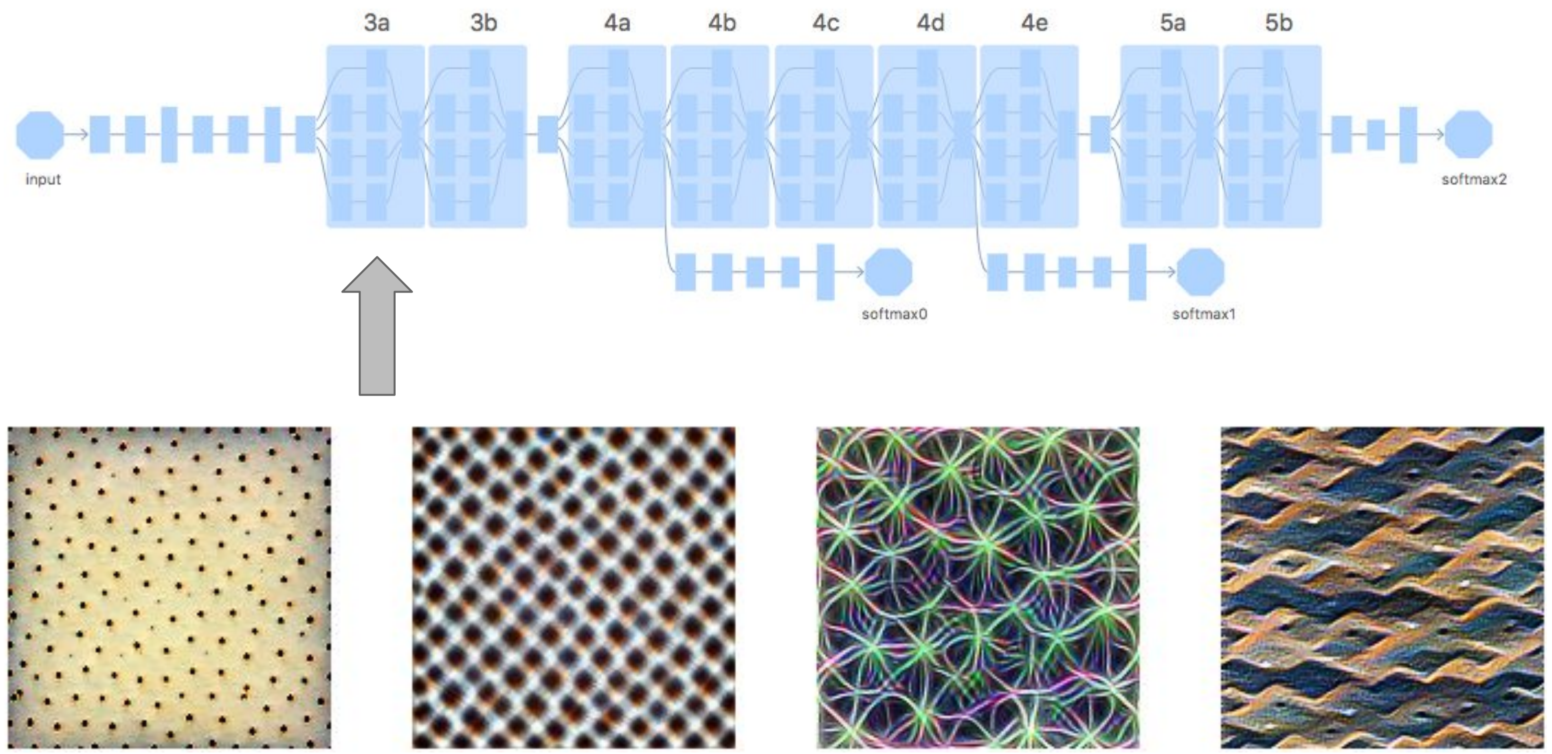
AlexNet




Input image
Feature engineering
Derive features (e.g., Harr Cascades)
Model
Machine learning modeling

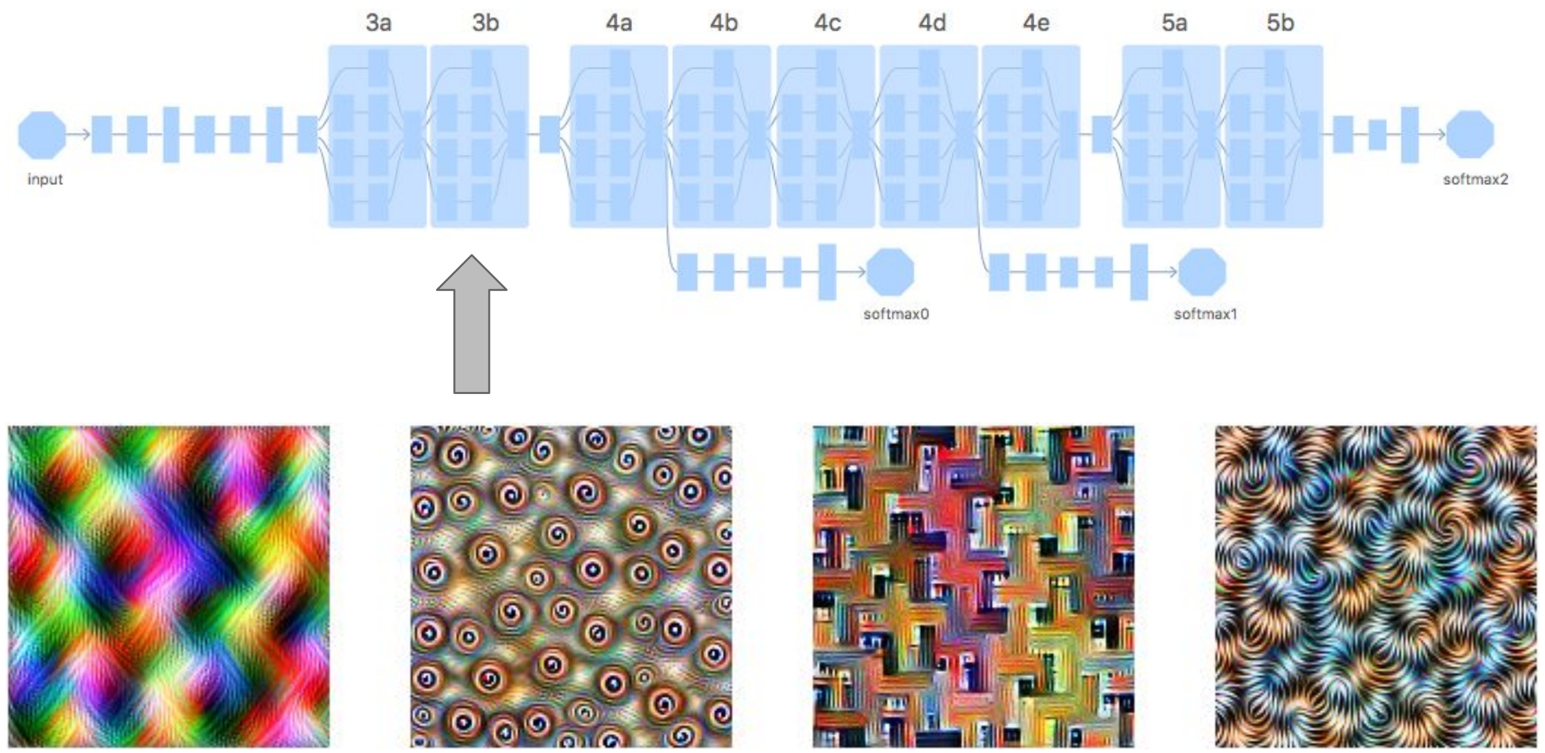

Output
1
0
Input
Convolution
Pooling
Convolution
Pooling
Dense
Dense

Visualizing layer activations



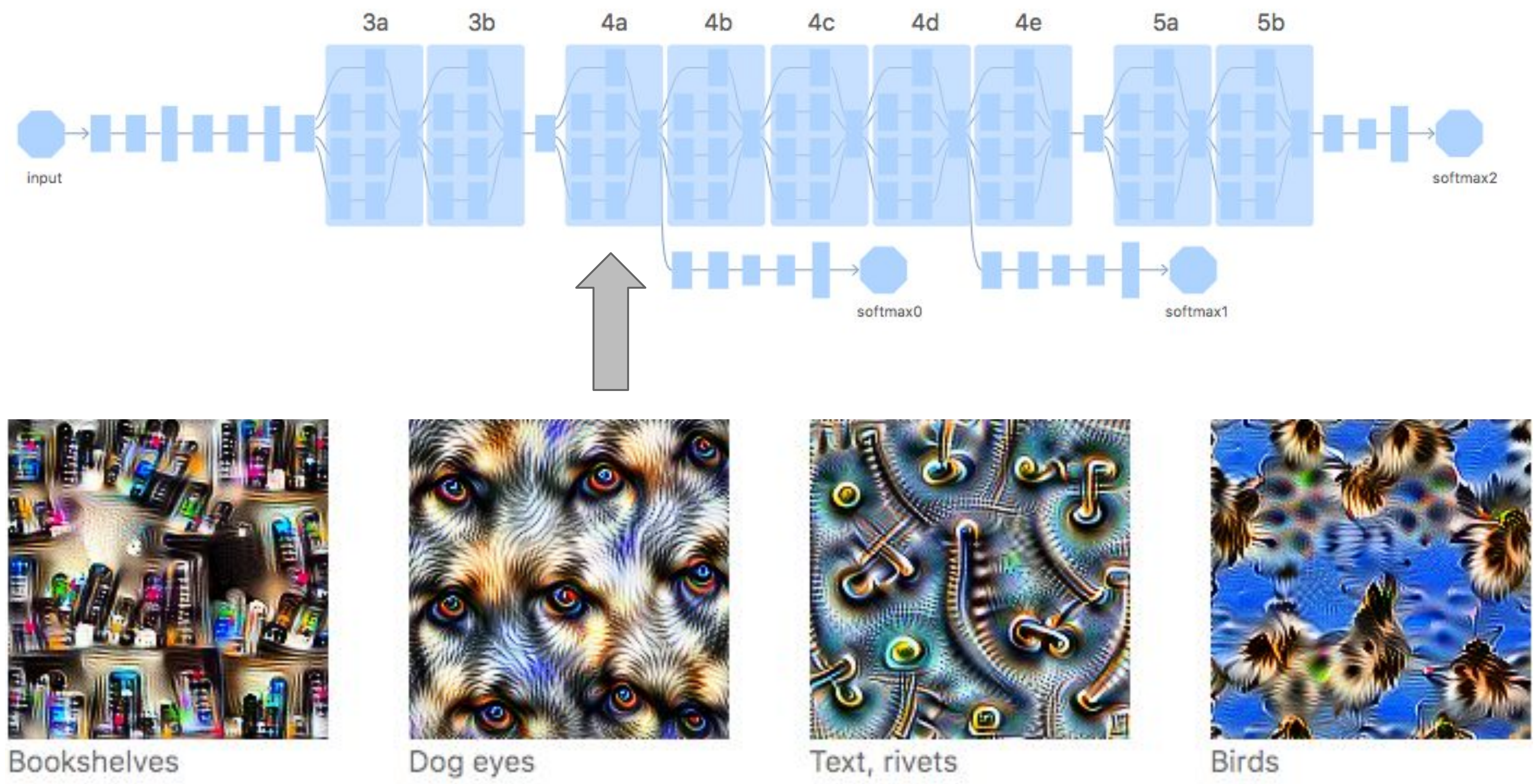
Diagrams by Chris Olah
https://distill.pub/2017/feature-visualization/appendix/

Visualizing layer activations



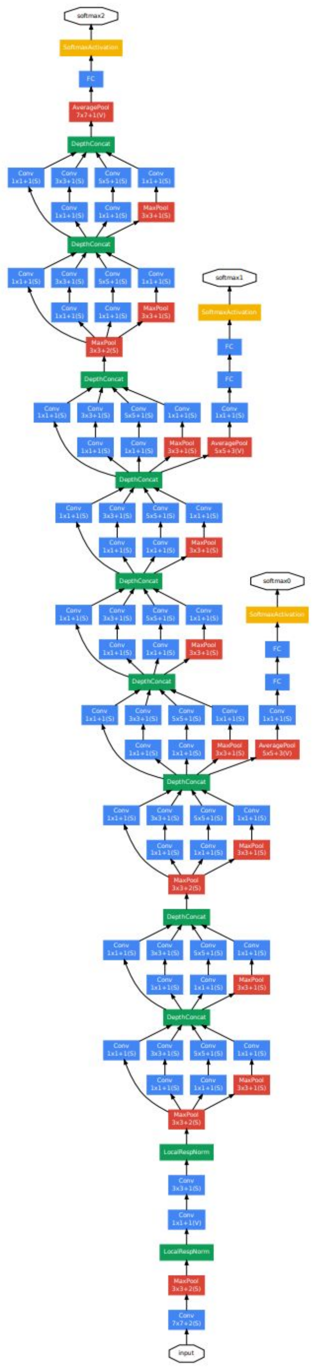
Diagrams by Chris Olah
https://distill.pub/2017/feature-visualization/appendix/

Visualizing layer activations



Diagrams by Chris Olah
https://distill.pub/2017/feature-visualization/appendix/

GoogLeNet



2012 AlexNet: 8

2013 ZFNet: 8

2014 VGGNet: 19
2014 GoogLeNet: 22

...

150?

File Name: T-ICML-O_4_l2_batch_normalization

Format: Presenter in Studio

Title: Batch Normalization

Presenter: Lak Lakshmanan

Agenda

Introduction

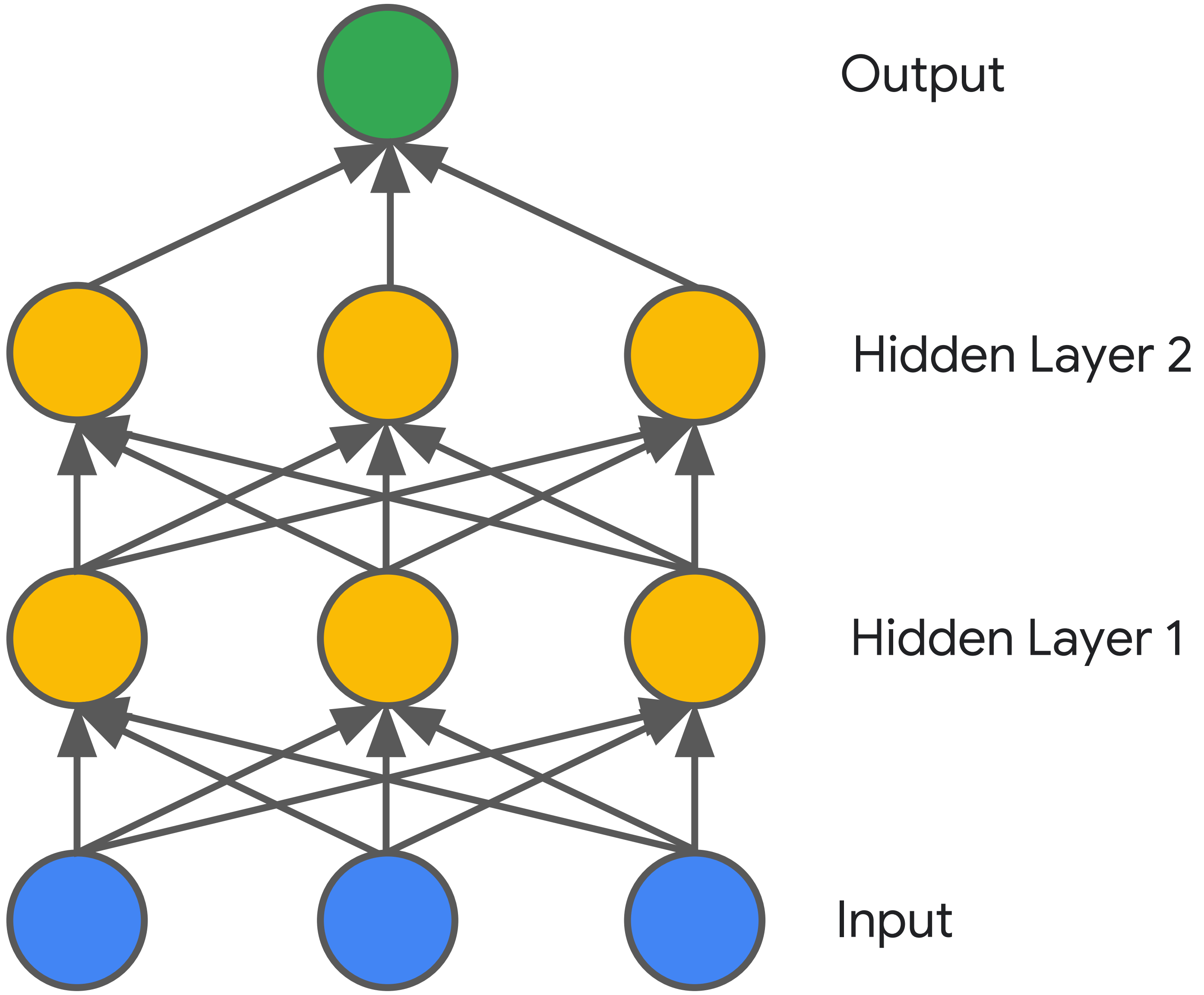
Batch Normalization

Residual Networks

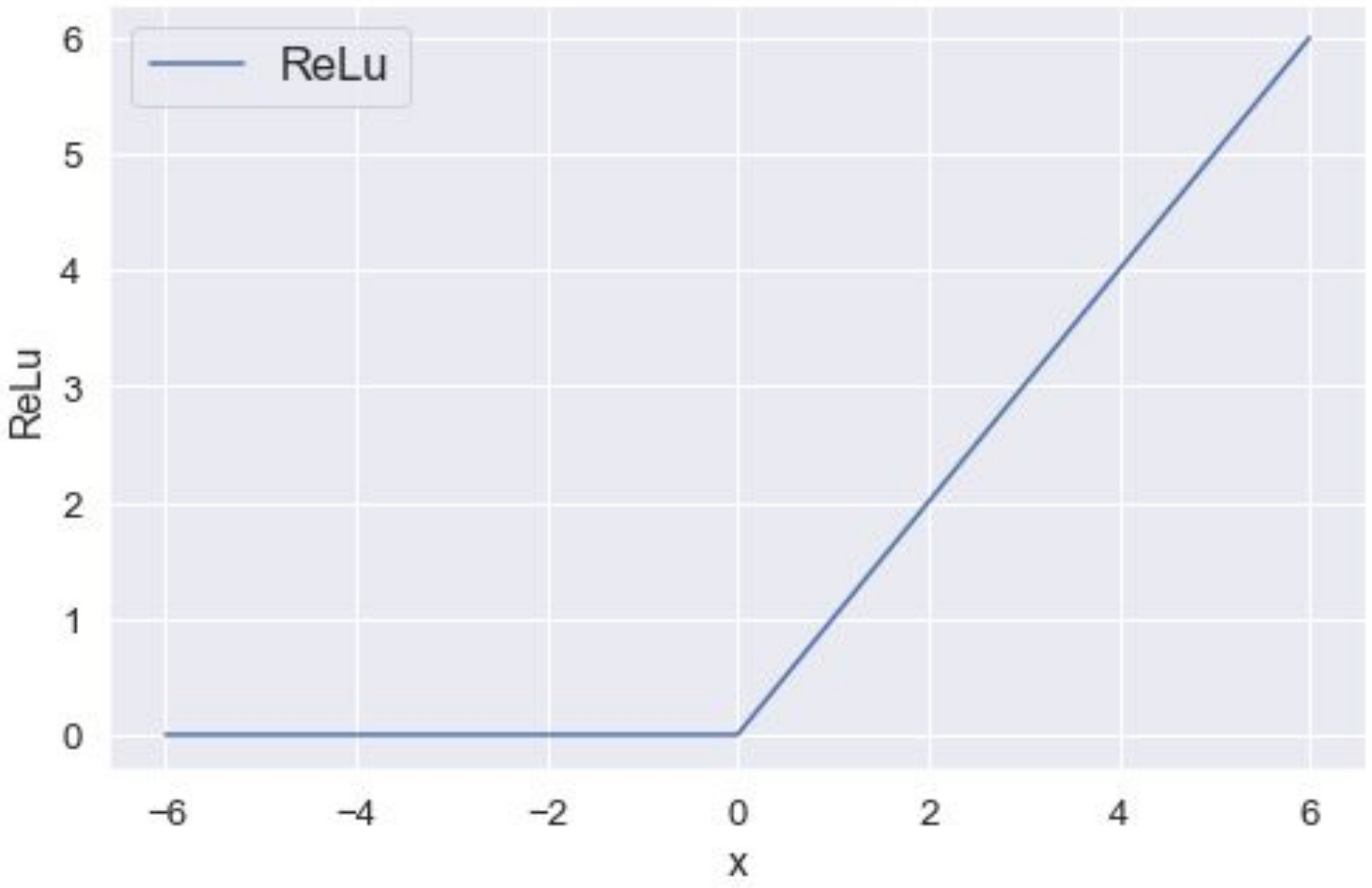
AI Accelerators

Architecture Search

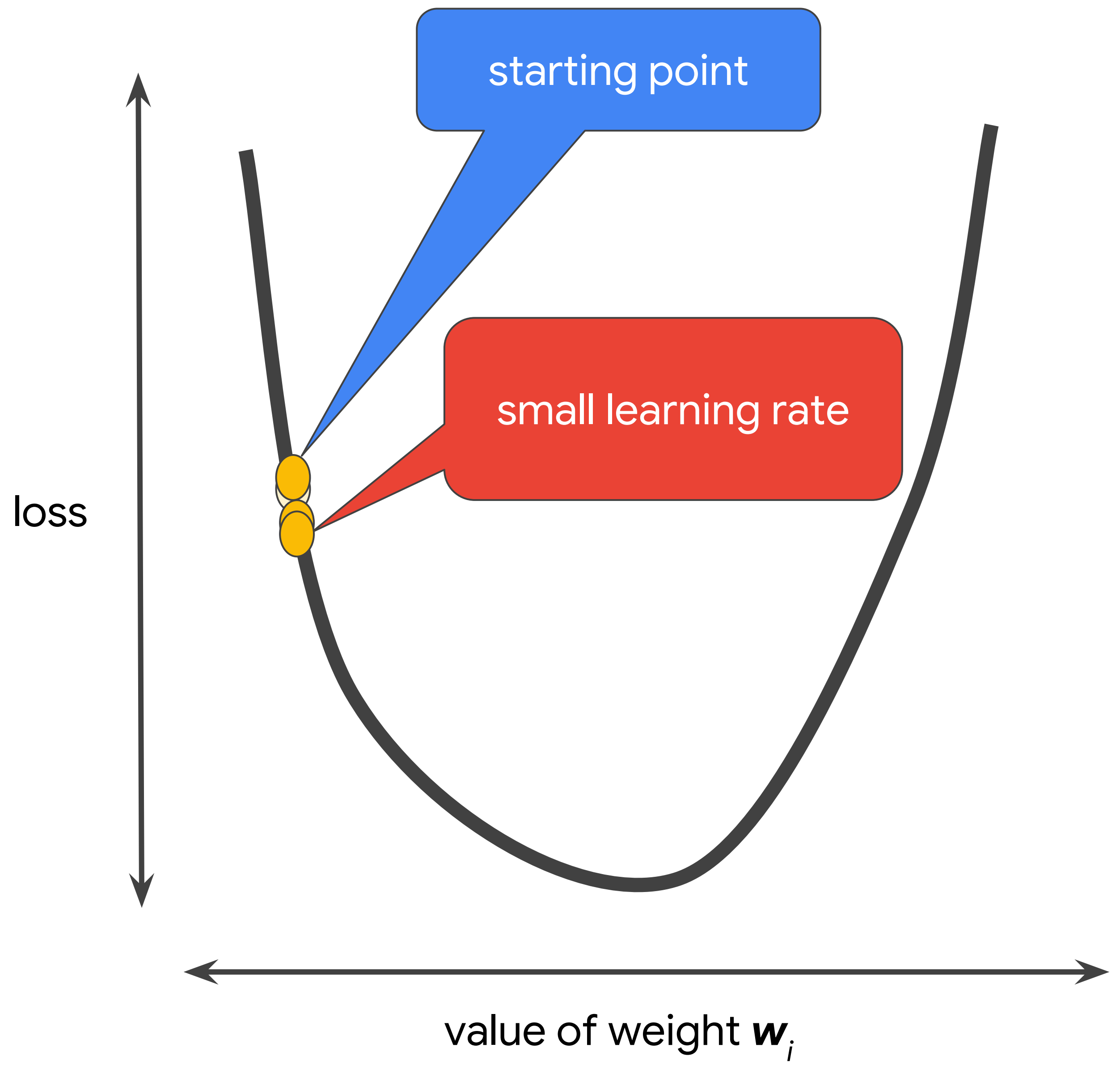
Internal Covariate Shift



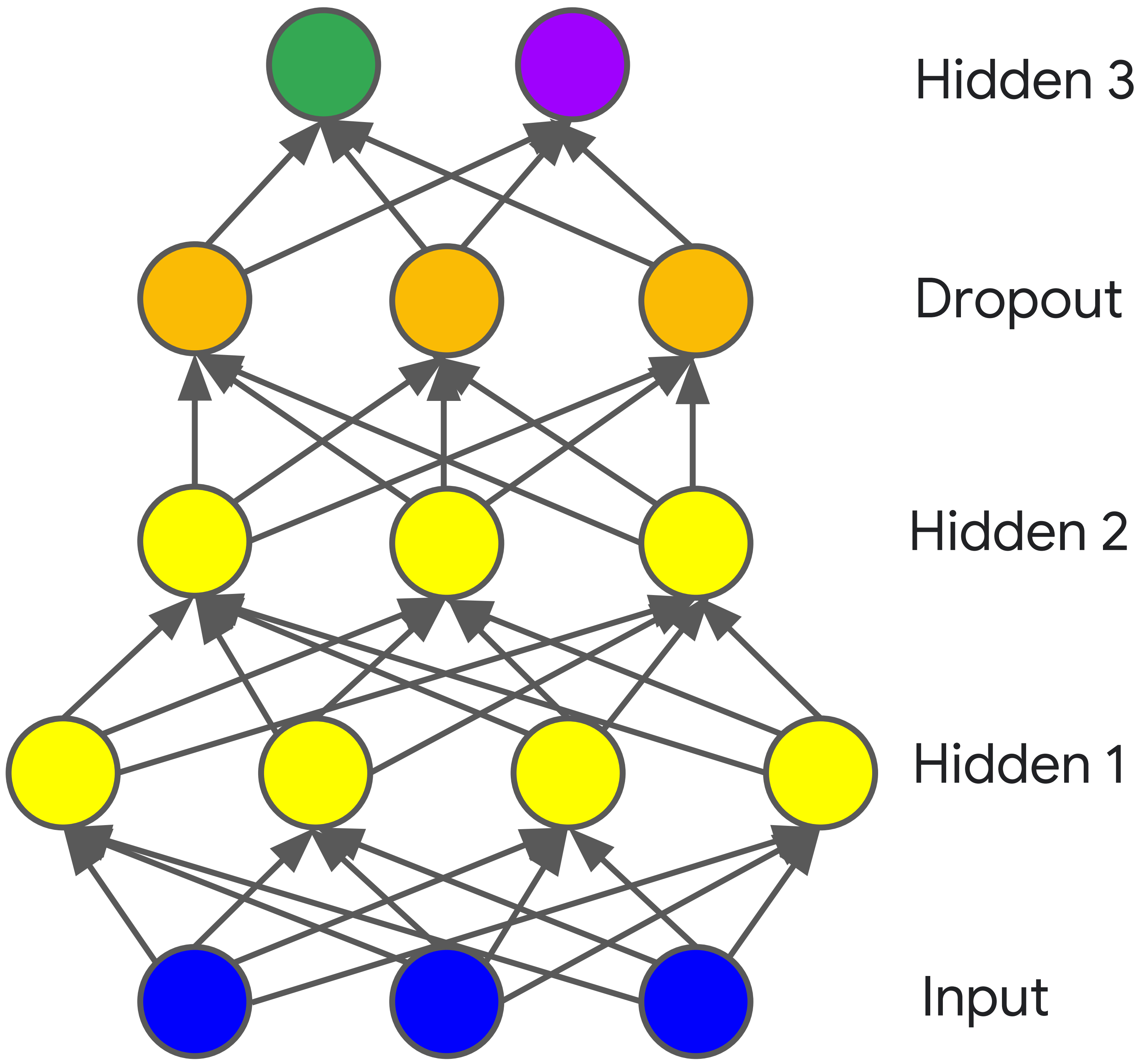
Neurons may stop learning



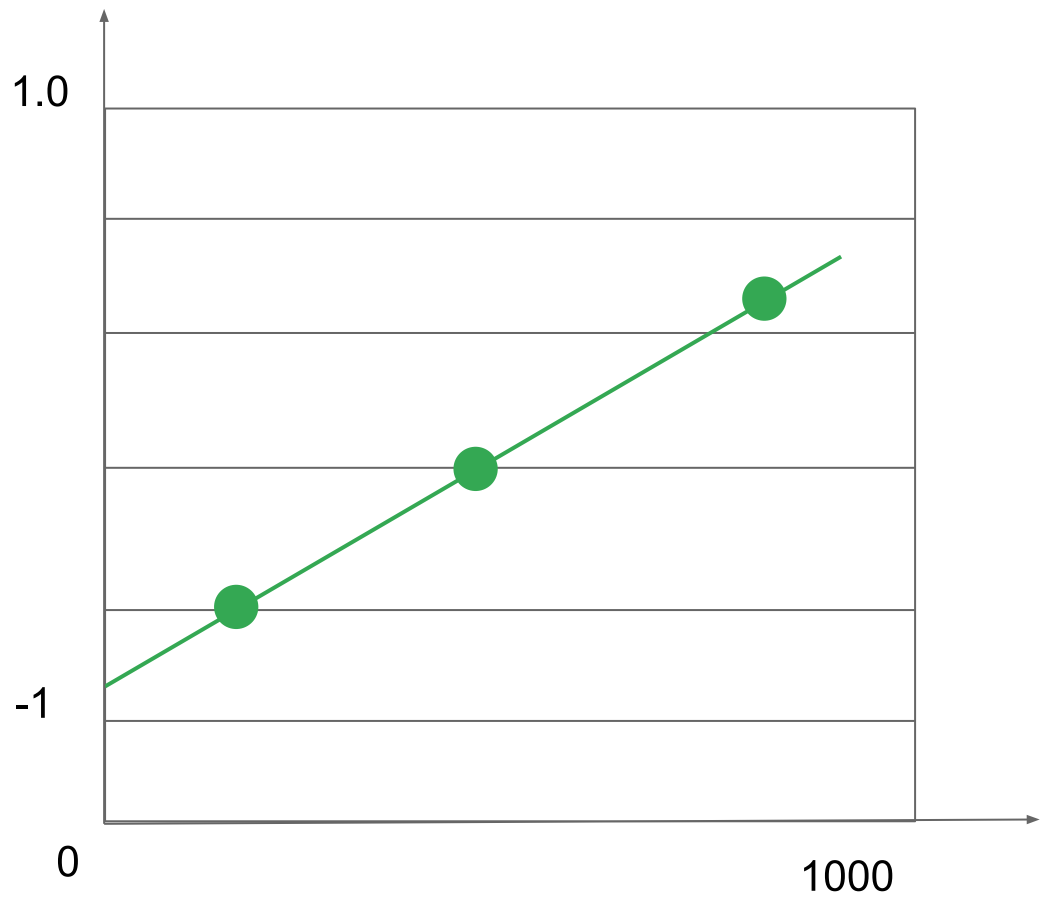
Lower the learning rate



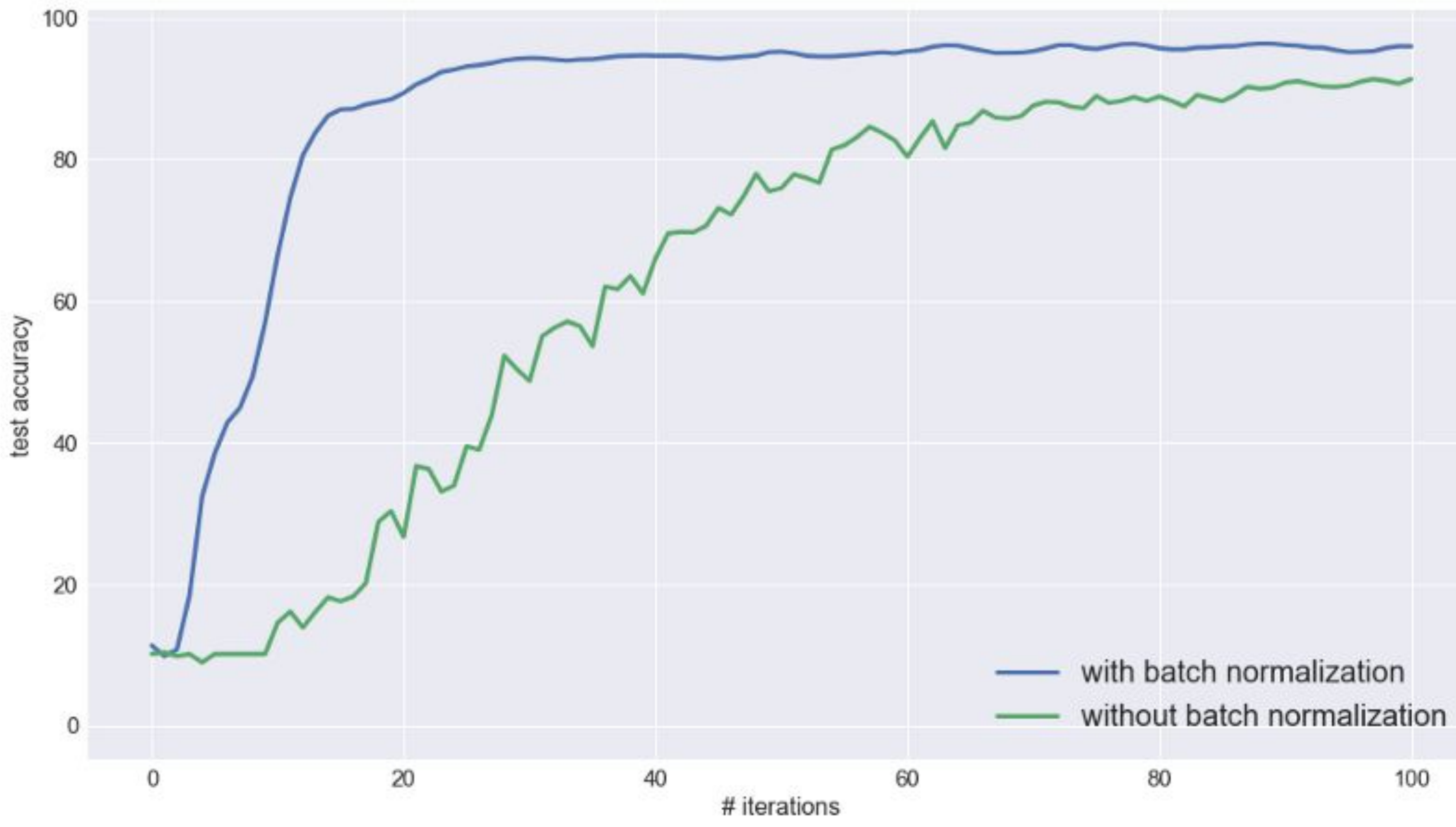
Use dropout



Batch normalization scales the weights between layers



Batch normalization helps you train faster



Batch normalization in a pre-made Estimator

```
estimator = DNNClassifier(
    feature_columns=[...],
    hidden_units=[1024, 512, 256],
    batch_norm=True)
```

Batch normalization in a custom Estimator with a Keras model function

```
layer2 = tf.keras.layers.Dense(...)
bn = tf.keras.layers.BatchNormalization(momentum=0.99)
layer3 = bn(layer2, training=training)
```

Batch normalization in a custom Estimator

```
layer3 = tf.layers.batch_normalization(
          layer2, training=(mode == TRAIN))
```

```
update_ops = tf.get_collection(tf.GraphKeys.UPDATE_OPS)
with tf.control_dependencies(update_ops):
    train_op = optimizer.minimize(loss)
```

Subsequent work with normalization

- Weight Normalization
- Layer Normalization
- Self-Normalizing Networks

File Name: T-ICML-O_4_l3_residual_networks

Format: Presenter in Studio

Title: Residual Networks

Presenter: Lak

Agenda

Introduction

Batch Normalization

Residual Networks

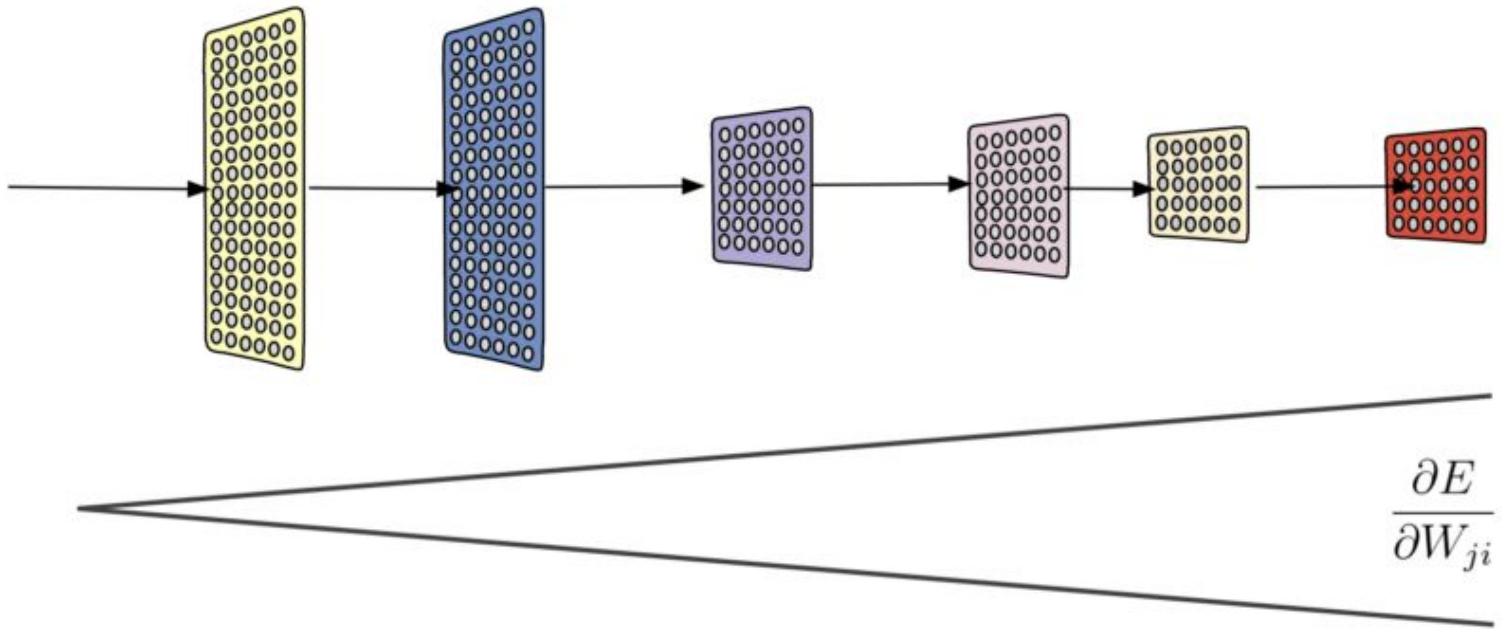
AI Accelerators

Architecture Search

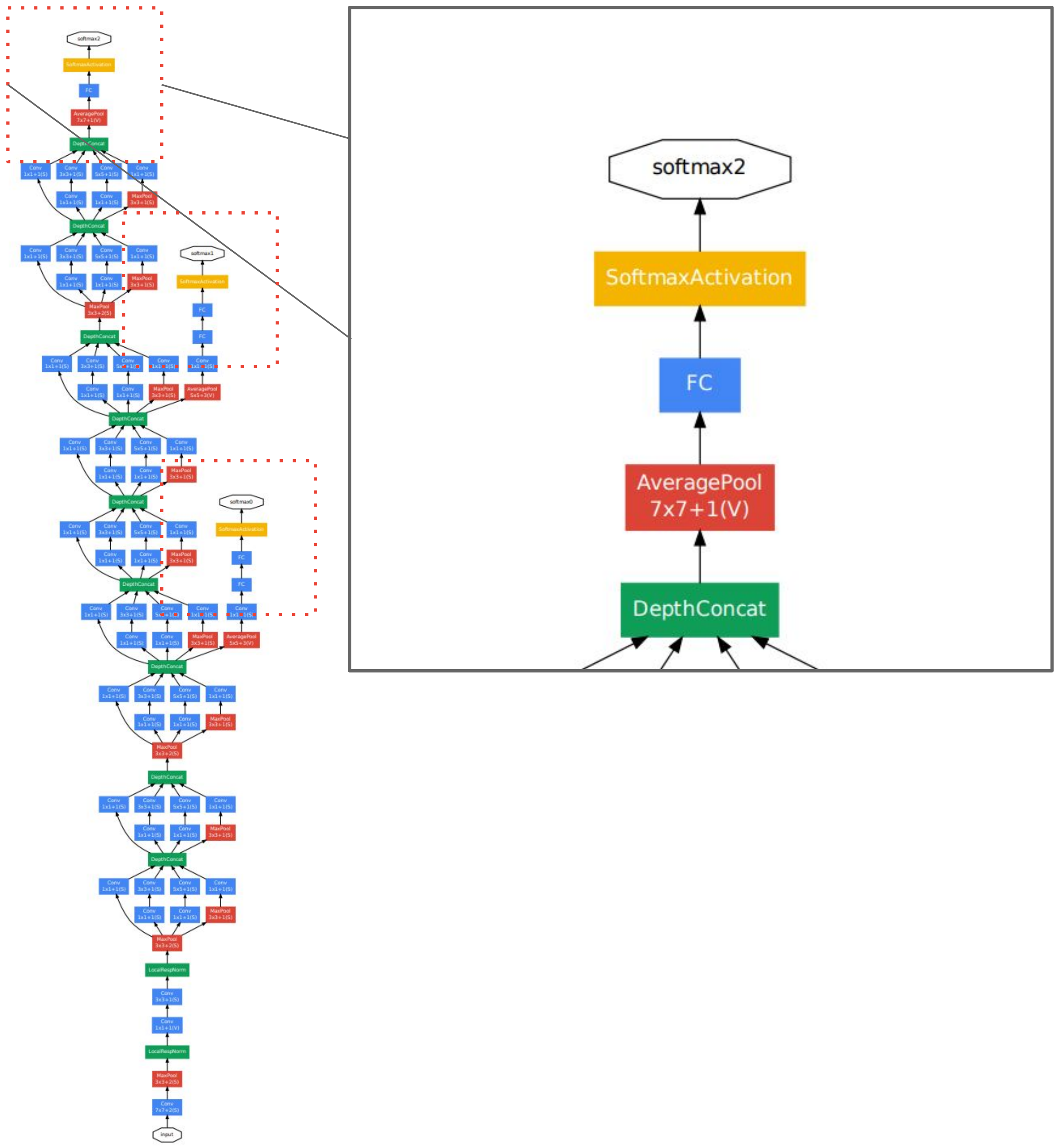
Depth didn't always help

Adapted from
https://arxiv.org/pdf/1512.03385.pdf

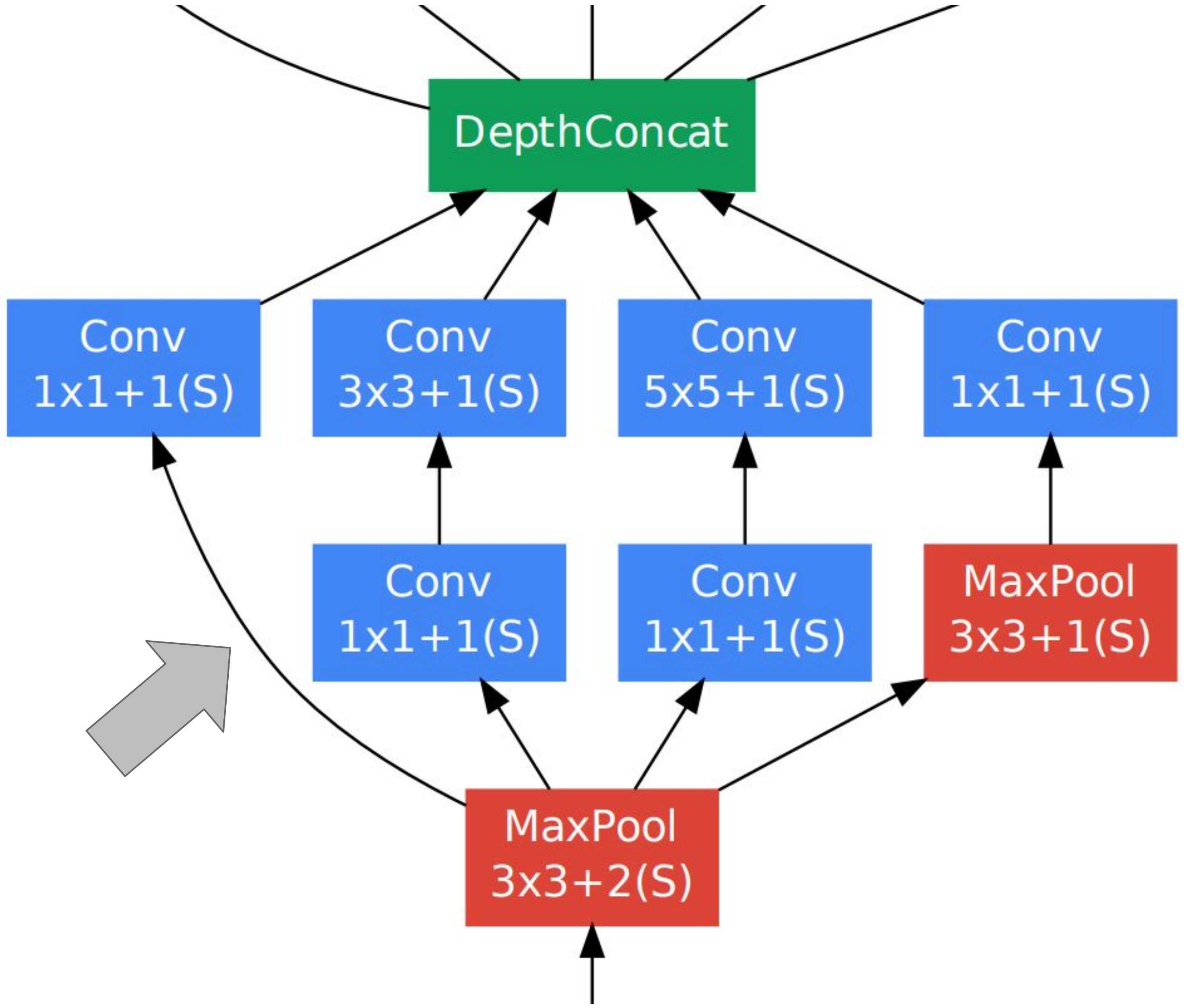
Vanishing gradients



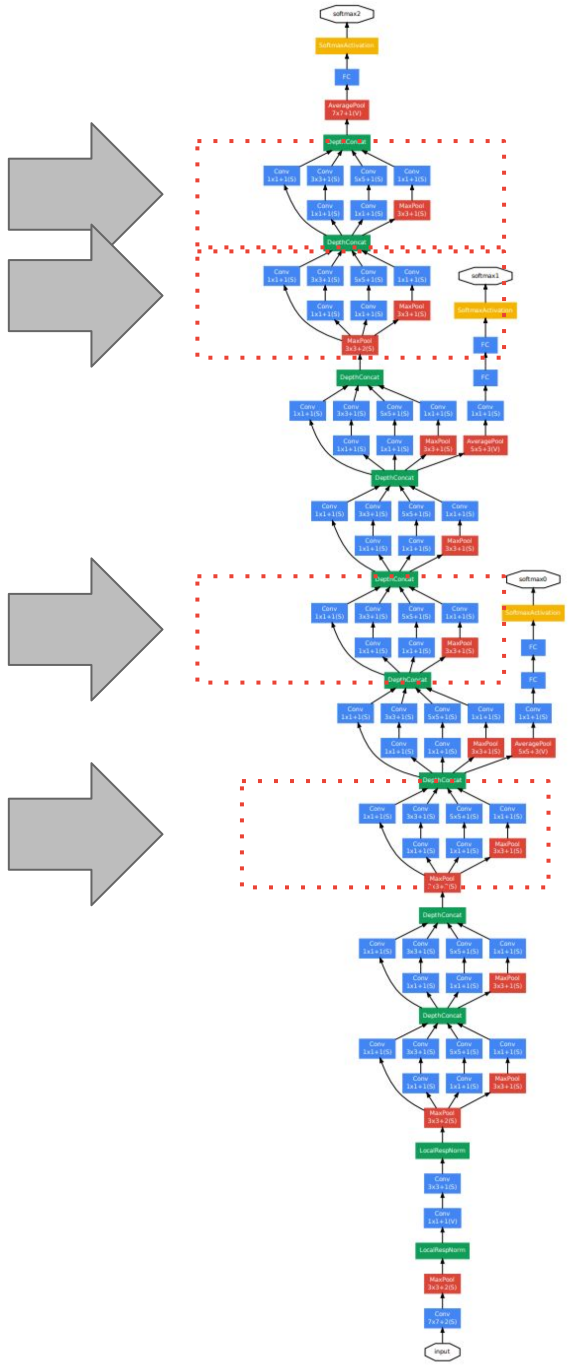
GoogLeNet


softmax2
SoftmaxActivation
FC
AveragePool
7x7+1(V)
DepthConcat

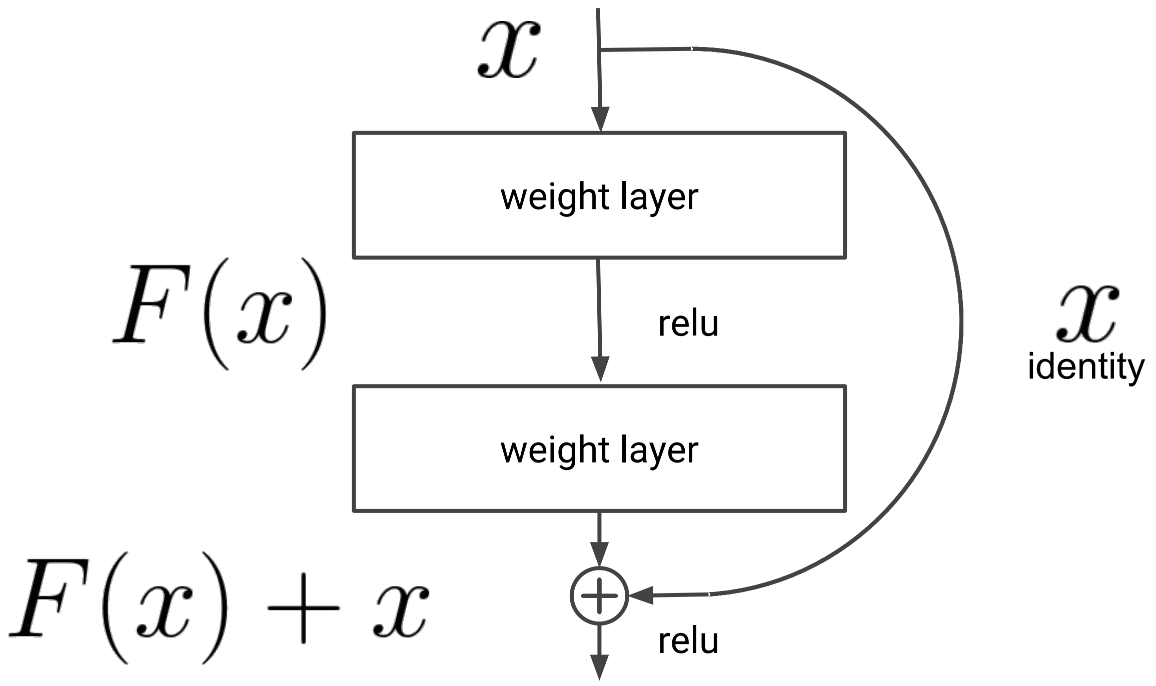
Parallel paths and shortcuts



GoogLeNet used a repeating structure

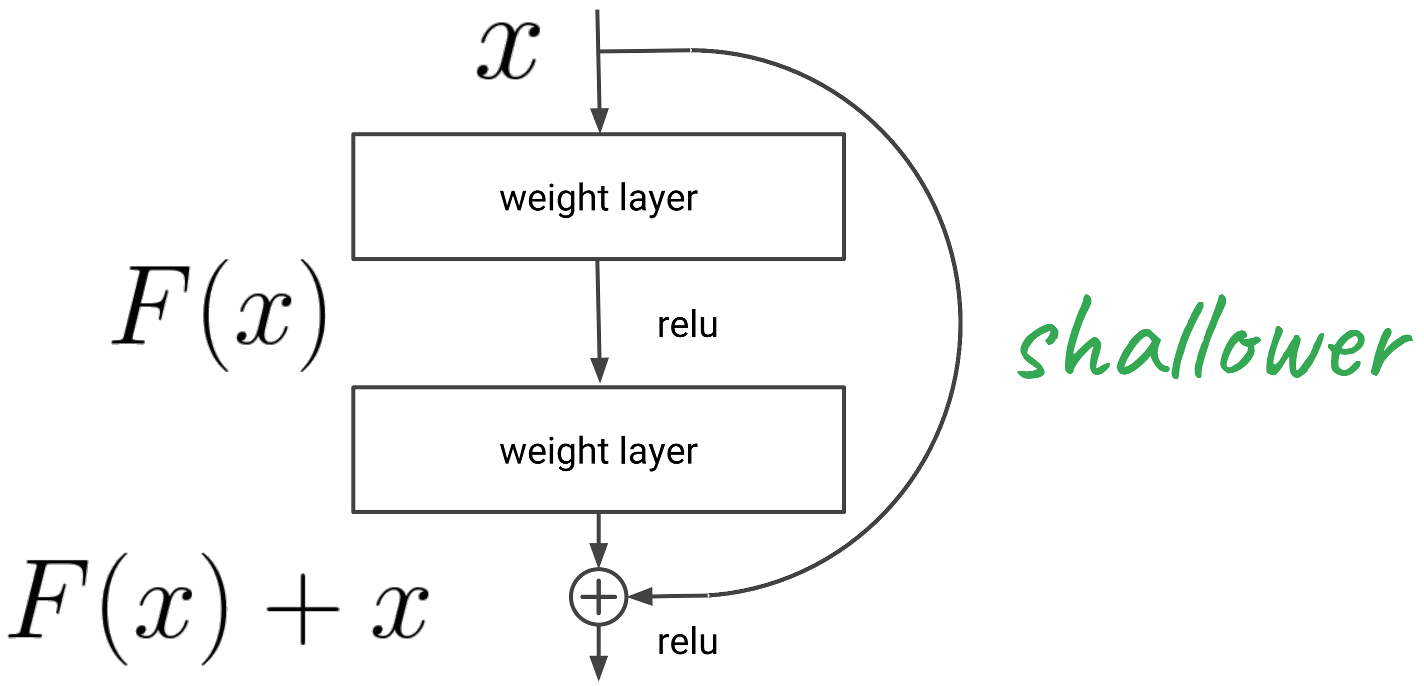


ResNet uses identity shortcut



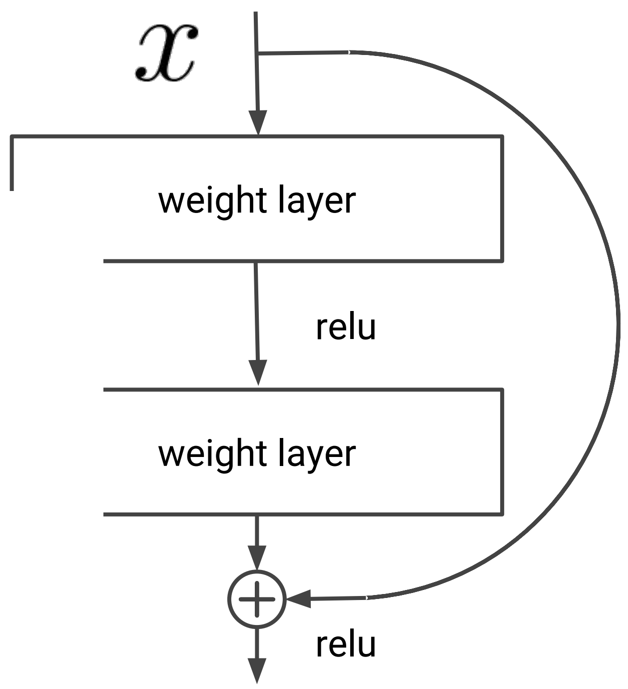
Deep Residual Learning for Image Recognition
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 2015
https://arxiv.org/abs/1512.03385

Residual shortcuts decrease the effective depth



Residual shortcuts decrease the effective depth

Residual is smaller



shallower

2012 AlexNet: 8

2013 ZFNet: 8

2014 VGGNet: 19
2014 GoogLeNet: 22

2015 ResNet-152

Subsequent work on residual connections

1. ResNext
2. DenseNet
3. FractalNet
4. SqueezeNet
5. Stochastic Depth

File Name: T-ICML-O_4_l4_accelerators

Format: Presenter in Studio

Title: Accelerators

Presenter: Lak

Agenda

Introduction

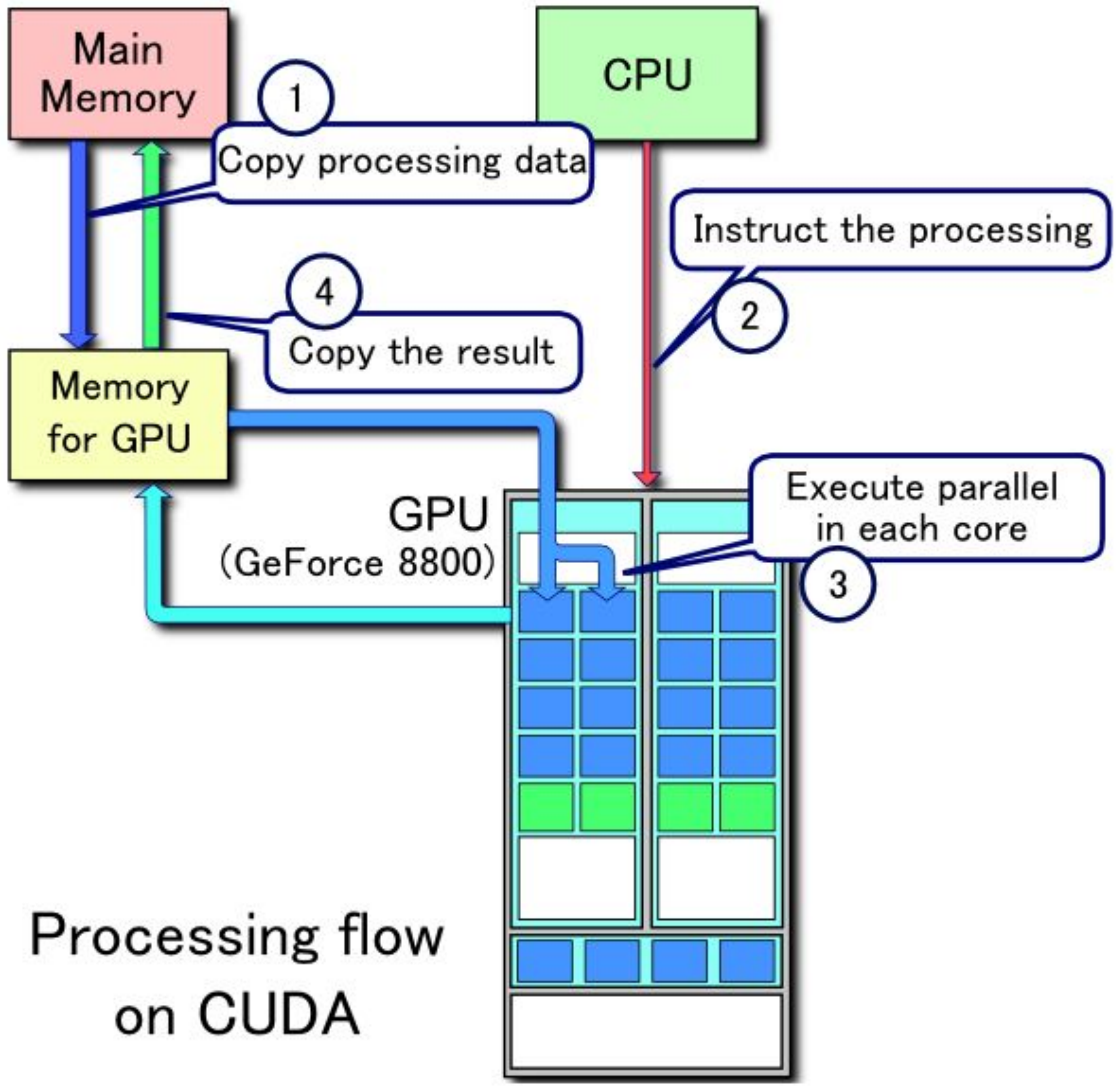
Batch Normalization

Residual Networks

AI Accelerators

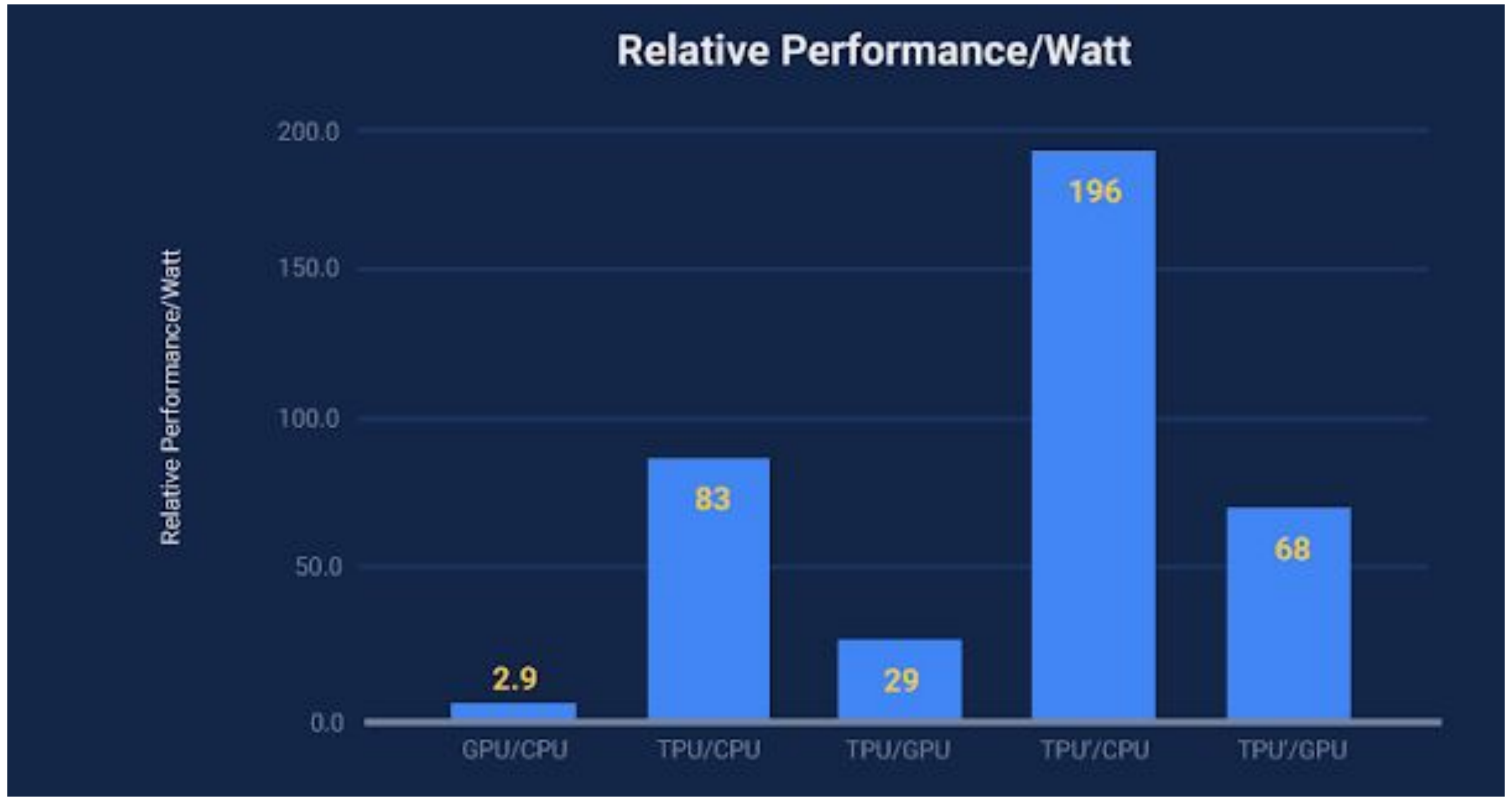
Architecture Search

CUDA for GPUs

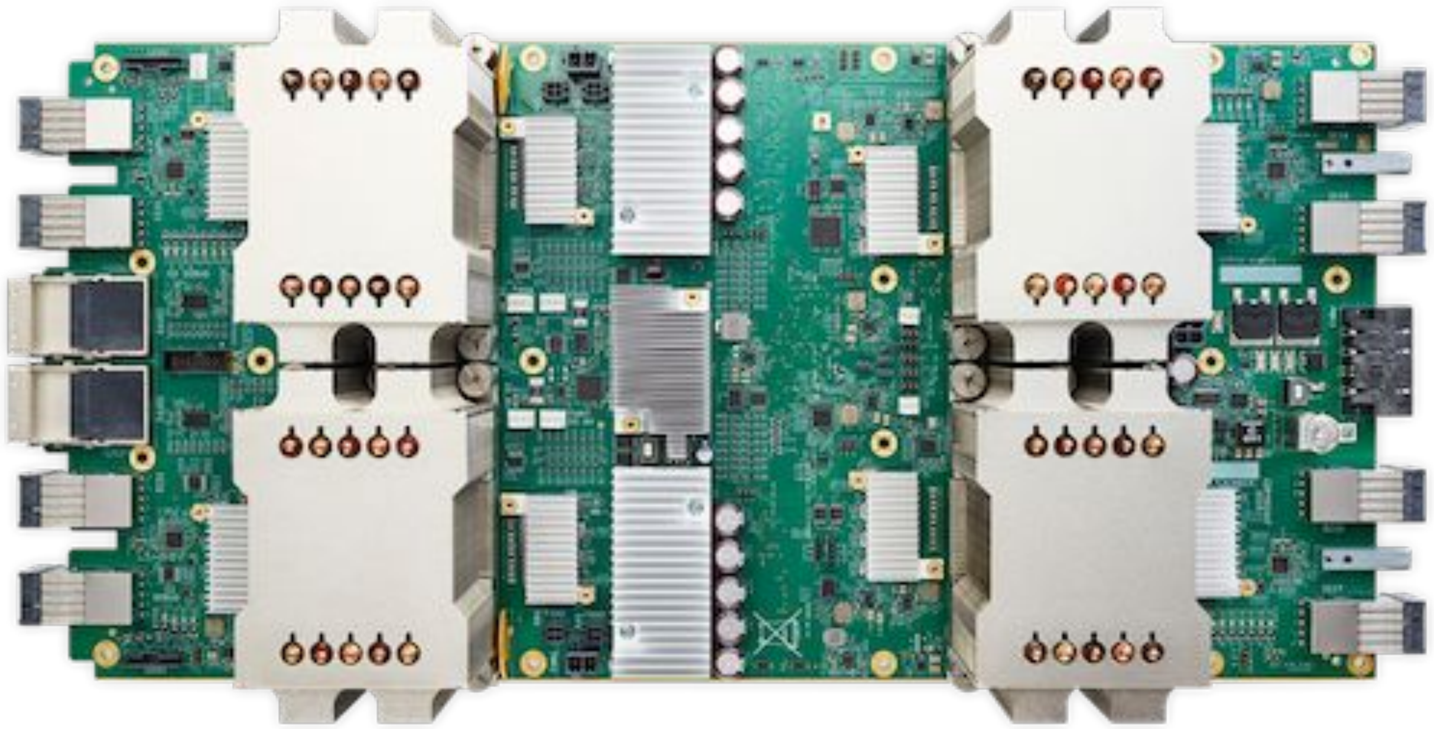


Source: Wikimedia Commons

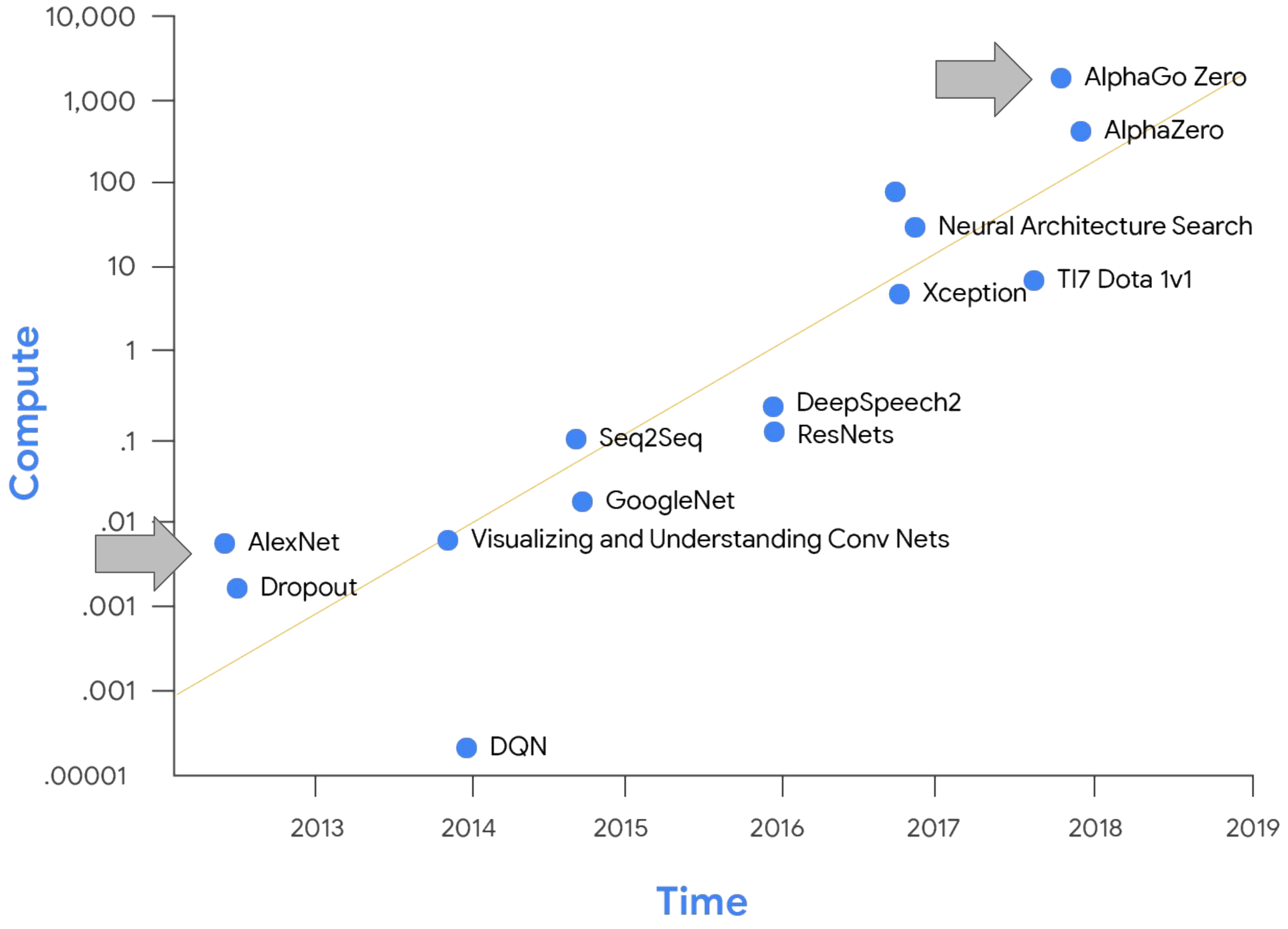
AI accelerators dramatically improve performance



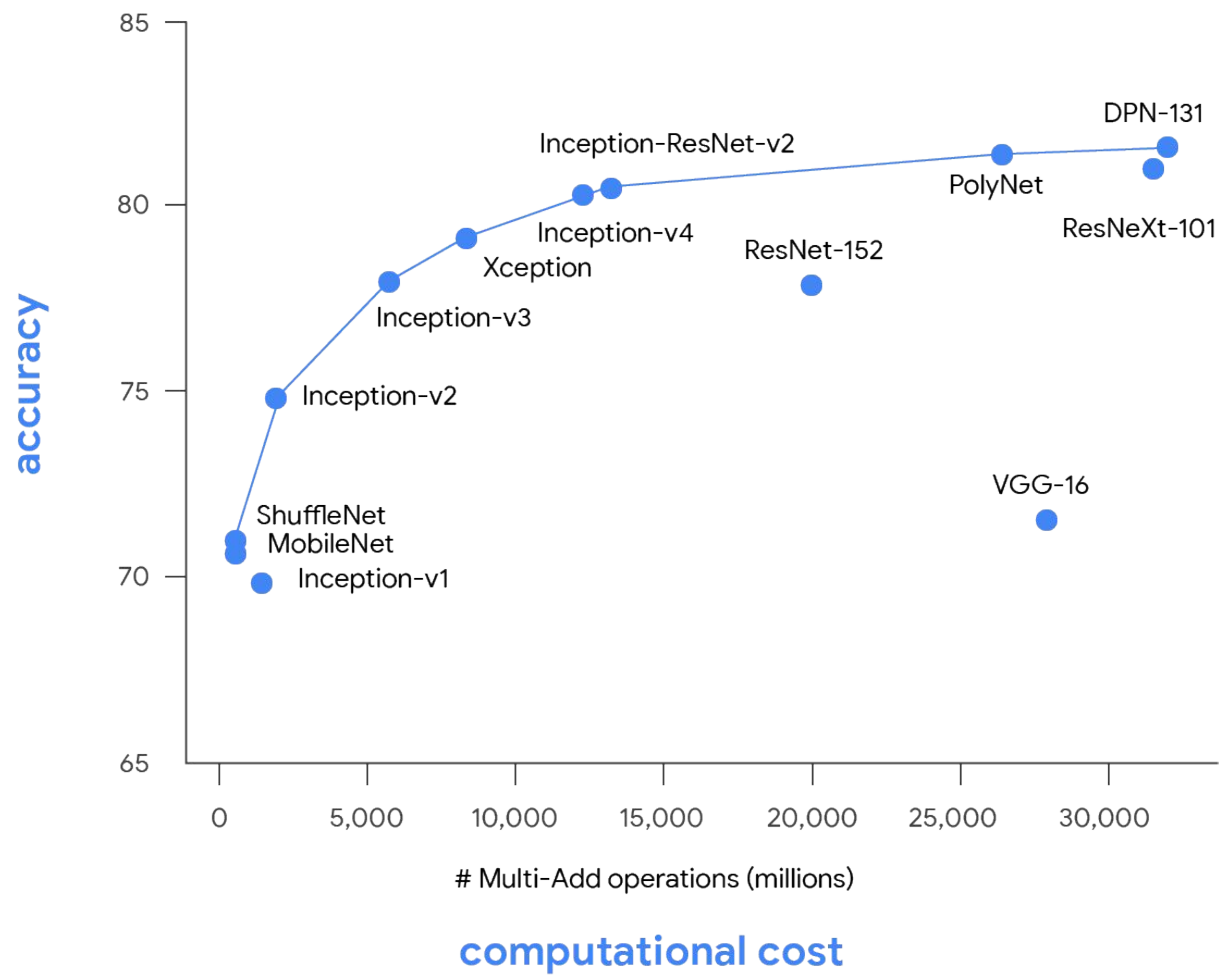
TPUs are ASICs



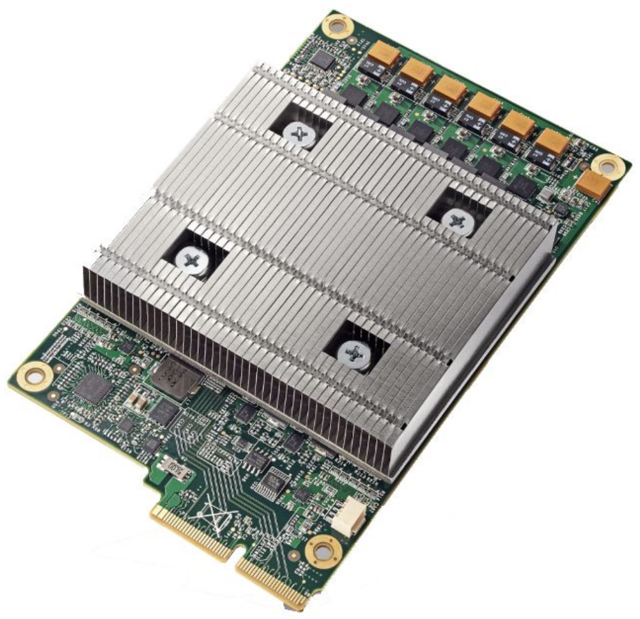
A 300,000x increase in compute



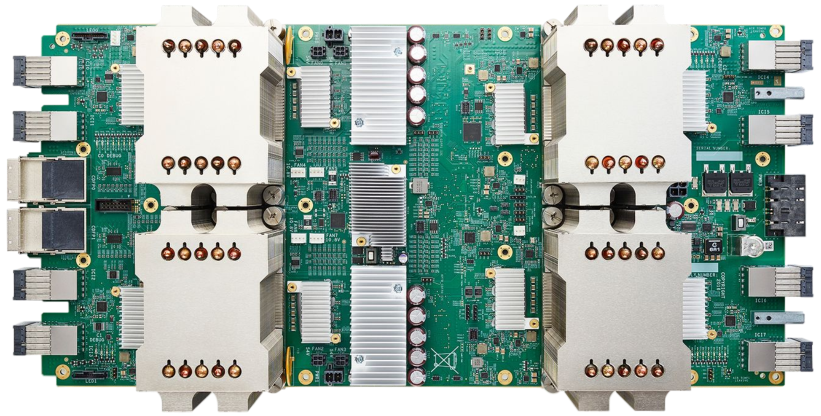
Many recent AI advances can be attributed to compute power



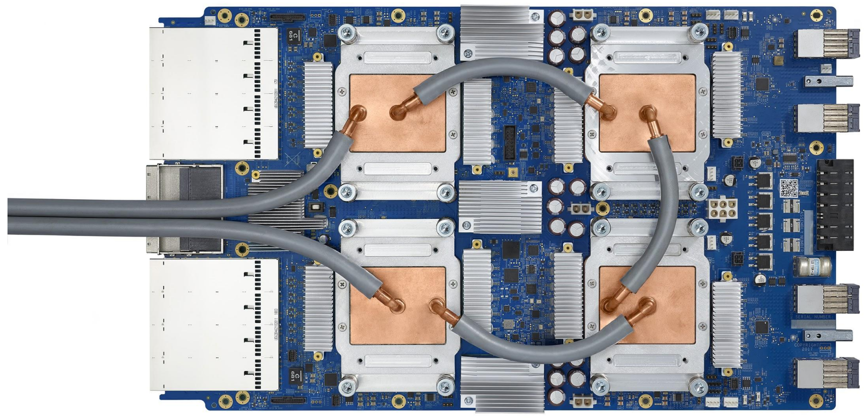
Expanding the AI frontier of performance



TPU v1
(2015)
92 teraops
First Generation

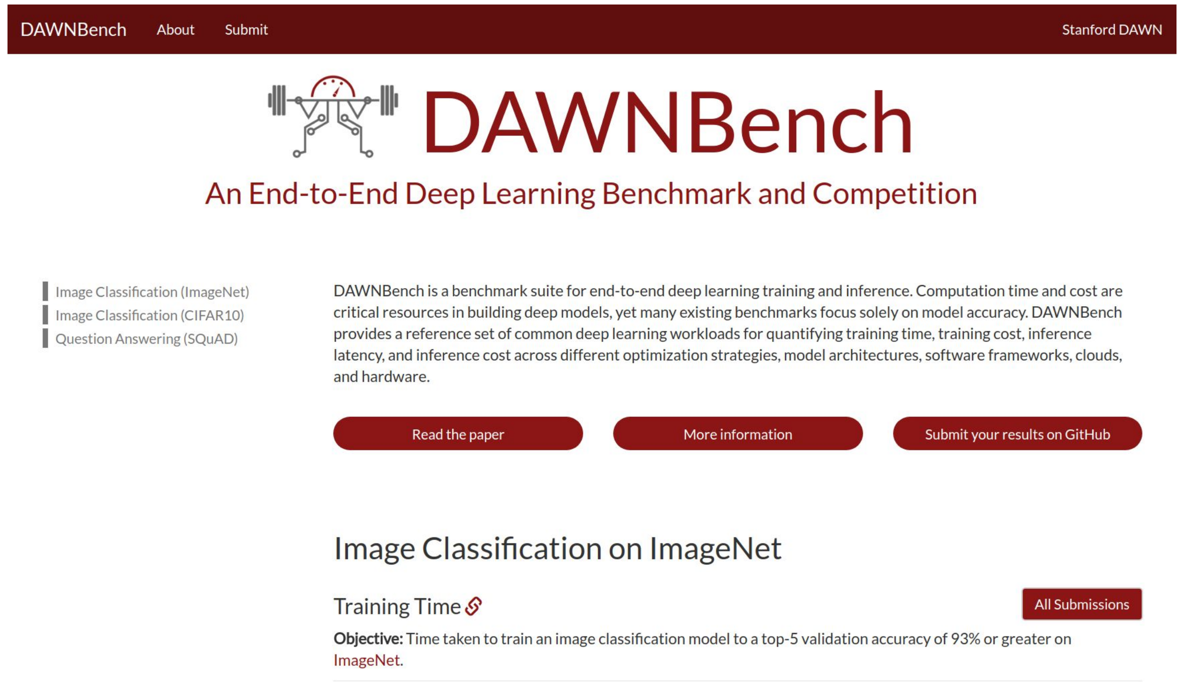


TPU v2
(2017)
180 teraflops
Available via Google Cloud



TPU v3
(2018)
420 teraflops
Available via Google Cloud

ResNet-50 on TPUv2



ResNet-50 on TPUv2 half-pod

Real data: **77,392** images/sec

Final accuracy: **93%**

Training time: **30** minutes

23.9 min without checkpoints!

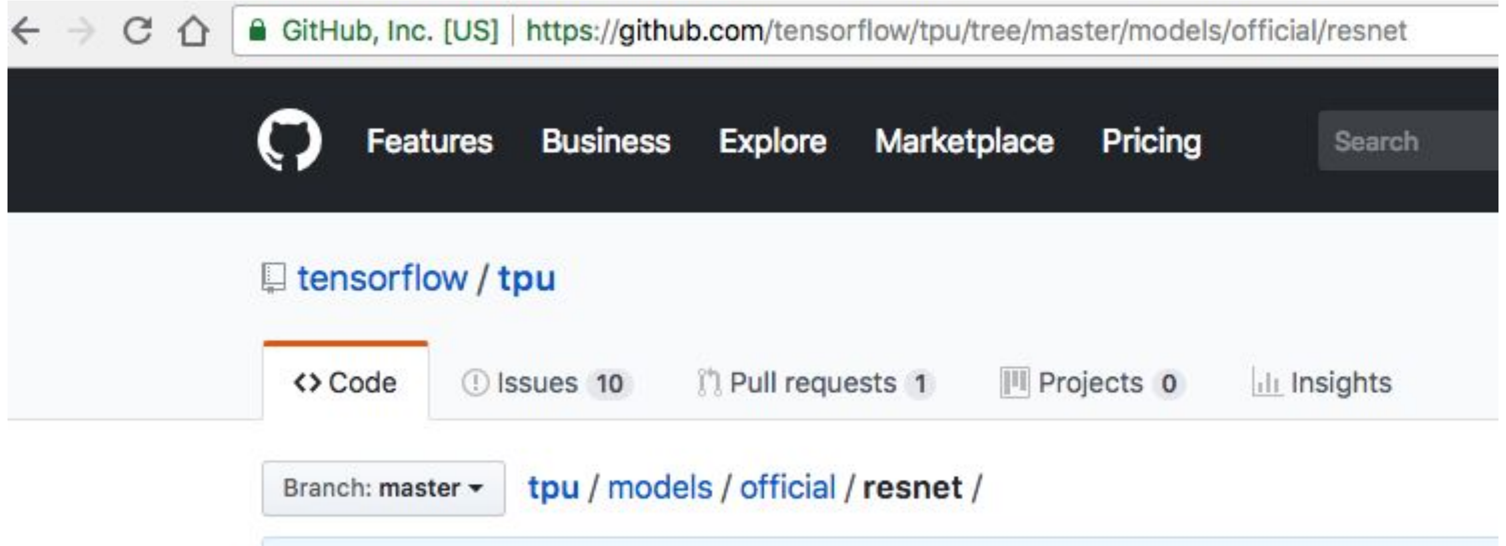
File Name: T-ICML-O_4_l5_lab_introduction:__resnet_on_tpu

Format: Presenter in Studio

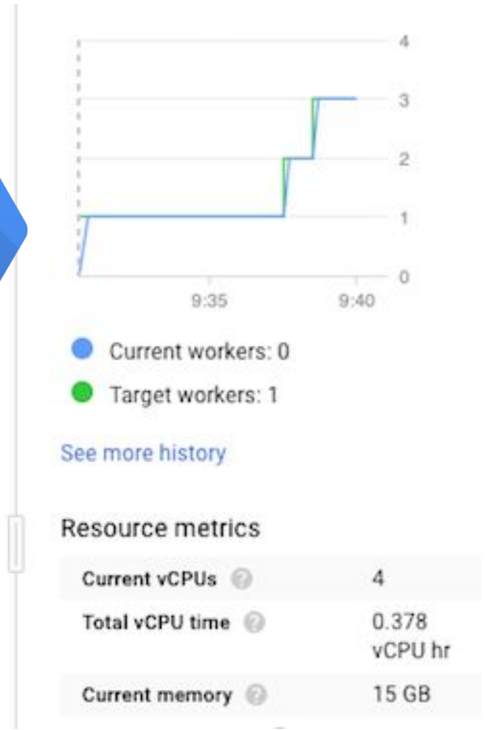
Title: Lab Introduction: ResNet on TPU

Presenter: Lak

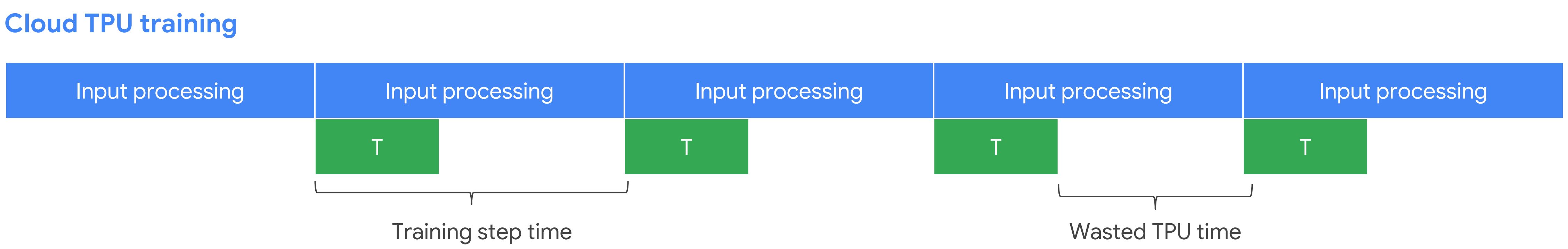
ResNet on TPU: Code



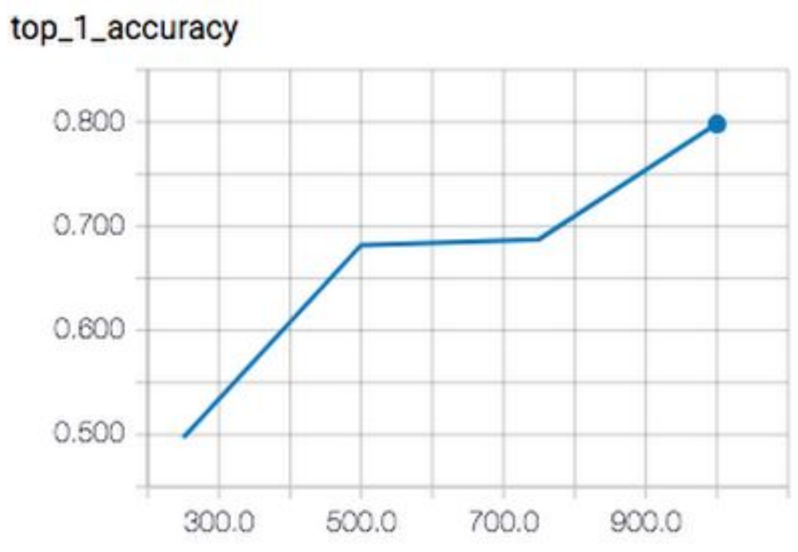
ResNet on TPU: Step 1



ResNet on TPU: Step 1



ResNet on TPU: Step 2



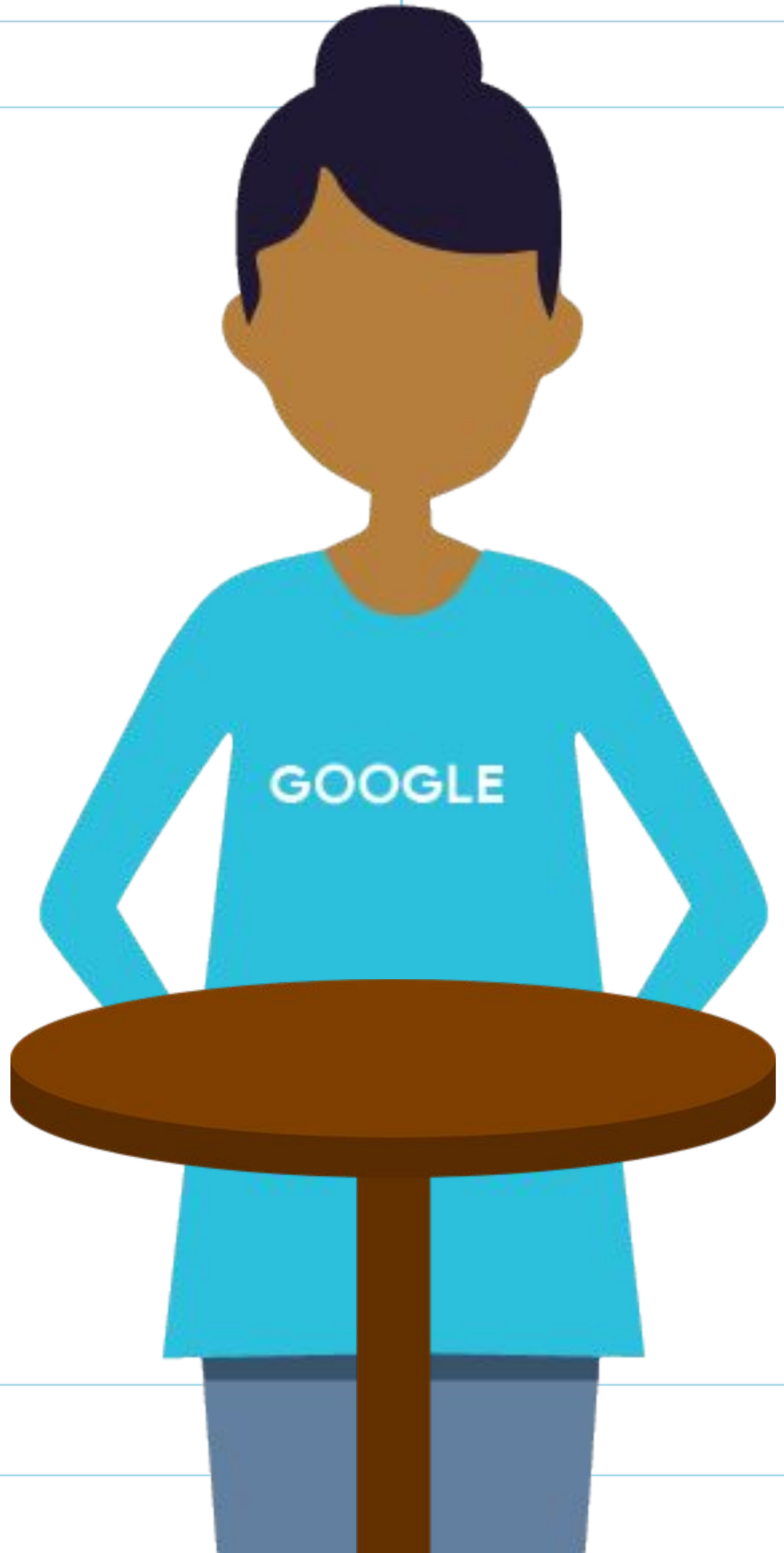

```
--scale-tier=BASIC_TPU
```

File Name: T-ICML-O_4_l6_lab_solution:_resnet_on_tpu

Format: Presenter in Studio

Title: Lab Solution: ResNet on TPU

Presenter: Lak



Title Safe >

< Action Safe

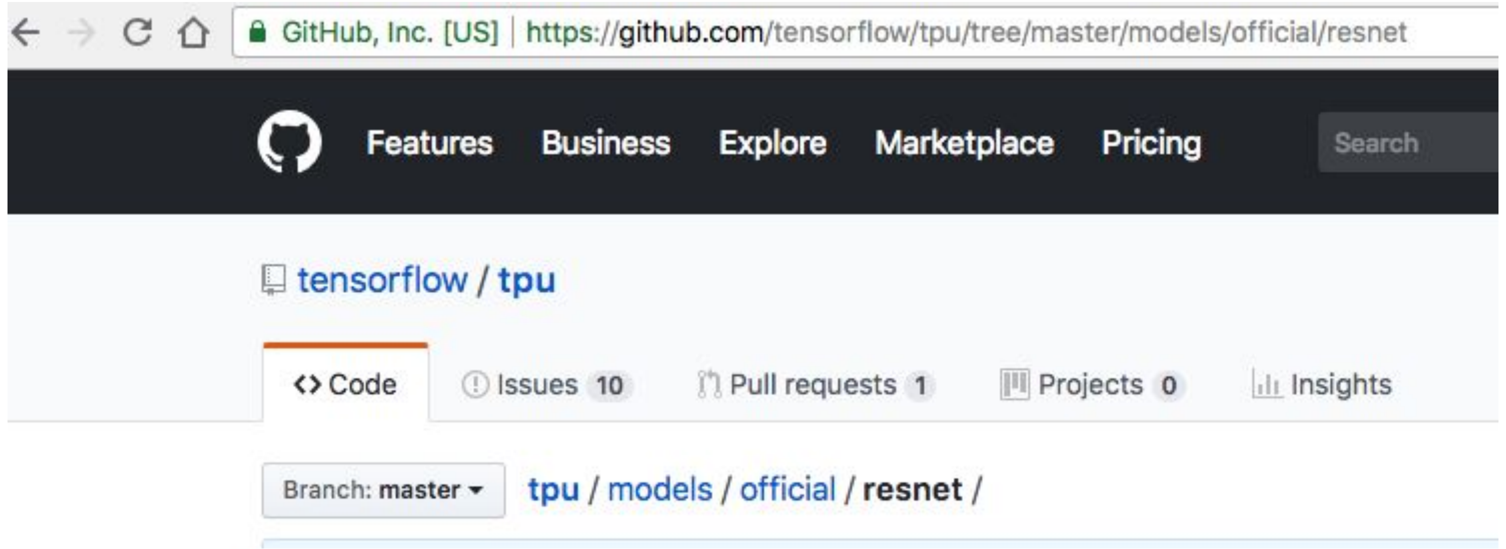
File Name: T-ICML-O_4_l7_tpu_estimator

Format: Presenter in Studio

Title: TPU Estimator

Presenter: Lak

Use the reference models



TPU for custom models

tf.estimator	High-level API for production-ready models
tf.layers, tf.losses, tf.metrics	Components useful for building custom NN models
Core TensorFlow (Python)	Python API gives you full control
Core TensorFlow (C++)	C++ API is quite low-level
CPU, GPU, TPU, Android	TF runs on different hardware

Cloud ML Engine

4 Steps to make a TPUEstimator

1. Replace our optimizer
2. Replace our EstimatorSpec
3. Replace our RunConfig
4. Replace our Estimator

TPU for custom models

```
optimizer = tf.contrib.tpu.CrossShardOptimizer(optimizer)
```

TPU for custom models

```
# change 1
optimizer = tf.contrib.tpu.CrossShardOptimizer(optimizer)

# change 2
return tf.contrib.tpu.TPUEstimatorSpec(
        mode=...,
        predictions=...,
        loss=...,
        train_op=...,
        eval_metrics=(metric_fn, [labels, logits]))

def metric_fn(labels, logits):
  accuracy = ...
  return {"accuracy": accuracy}
```

TPU for custom models

```
# change 1
optimizer = tf.contrib.tpu.CrossShardOptimizer(optimizer)

# change 2
return tf.contrib.tpu.TPUEstimatorSpec(...)

# change 3
iterations_per_loop = 1000
tpu_cluster_resolver =
  tf.contrib.cluster_resolver.TPUClusterResolver(
    hparams['tpu'],
    zone=hparams['tpu_zone'],
    project=hparams['project'])

config = tf.contrib.tpu.RunConfig(
    cluster=tpu_cluster_resolver,
    model_dir=output_dir,
    save_checkpoints_steps=max(600, iterations_per_loop),
    tpu_config=tf.contrib.tpu.TPUConfig(
      iterations_per_loop=iterations_per_loop,
      per_host_input_for_training=True))
```

TPU for custom models

```
# change 1
optimizer = tf.contrib.tpu.CrossShardOptimizer(optimizer)

# change 2
return tf.contrib.tpu.TPUEstimatorSpec(...)

# change 3
tpu_cluster_resolver = ...
config = tf.contrib.tpu.RunConfig(...)

# change 4
estimator = tf.contrib.tpu.TPUEstimator(
    model_fn=image_classifier,
    params=hparams,
    config=config,
    model_dir=output_dir,
    use_tpu=hparams['use_tpu'])
```

TPU for custom models

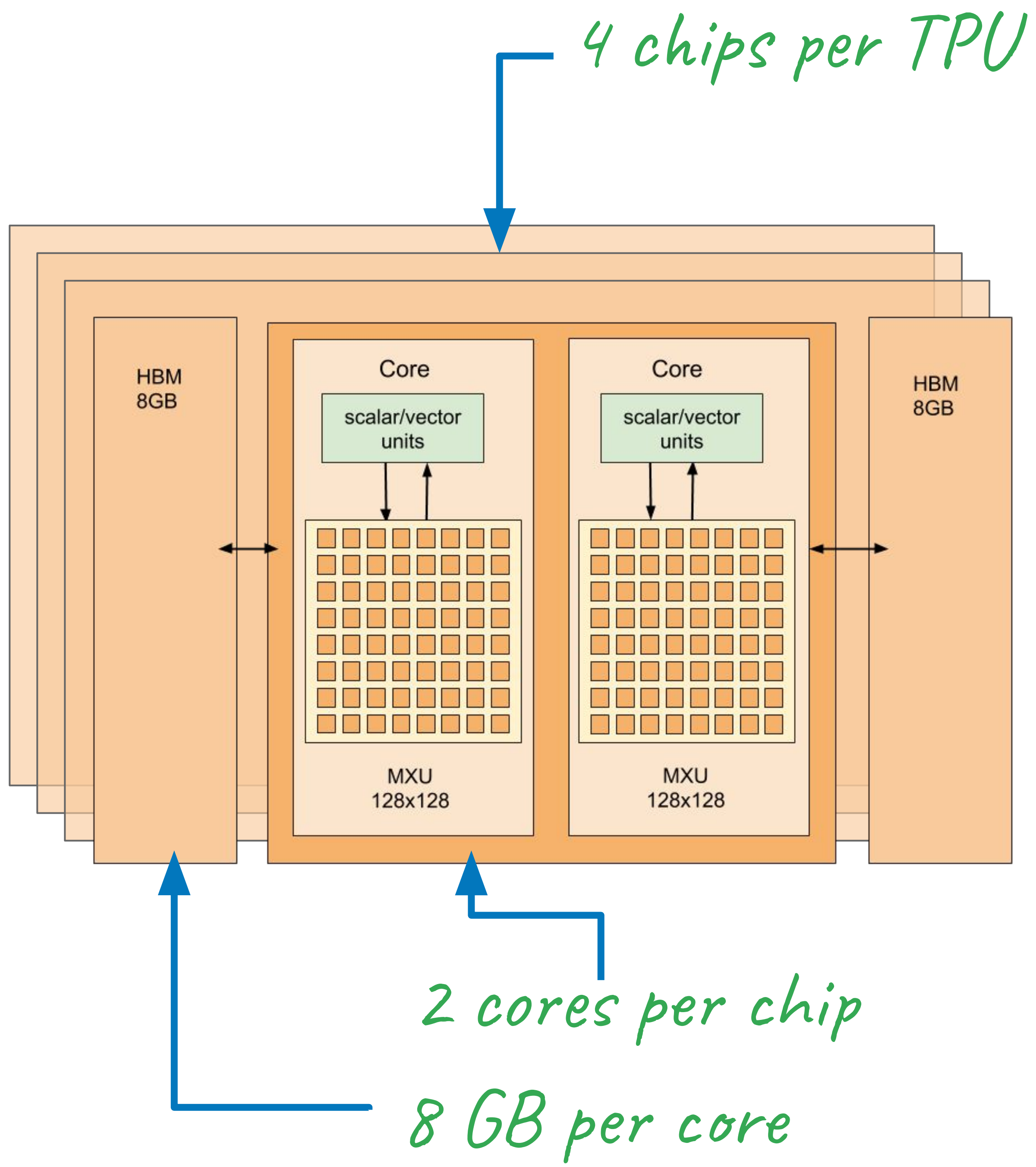
```
# change 4
estimator = tf.contrib.tpu.TPUEstimator(
    model_fn=image_classifier,
    params=hparams,
    config=config,
    model_dir=output_dir,
    use_tpu=hparams['use_tpu']),
```

TPU for custom models

```
# change 1
CrossShardOptimizer(optimizer)

# change 2
TPUEstimatorSpec(…)

# change 3
RunConfig(...)

# change 4
TPUEstimator(...)
```

Some points to keep in mind on TPUs

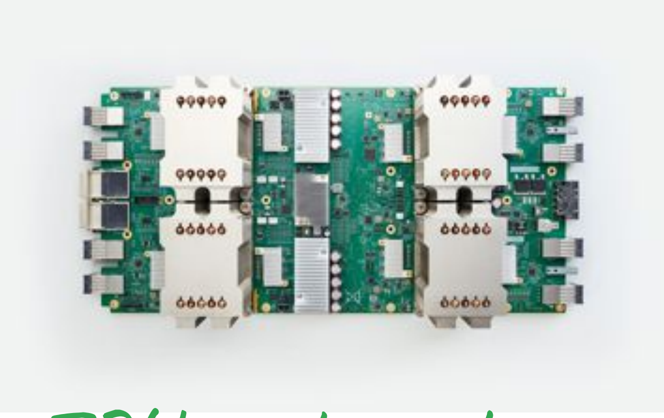
1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core



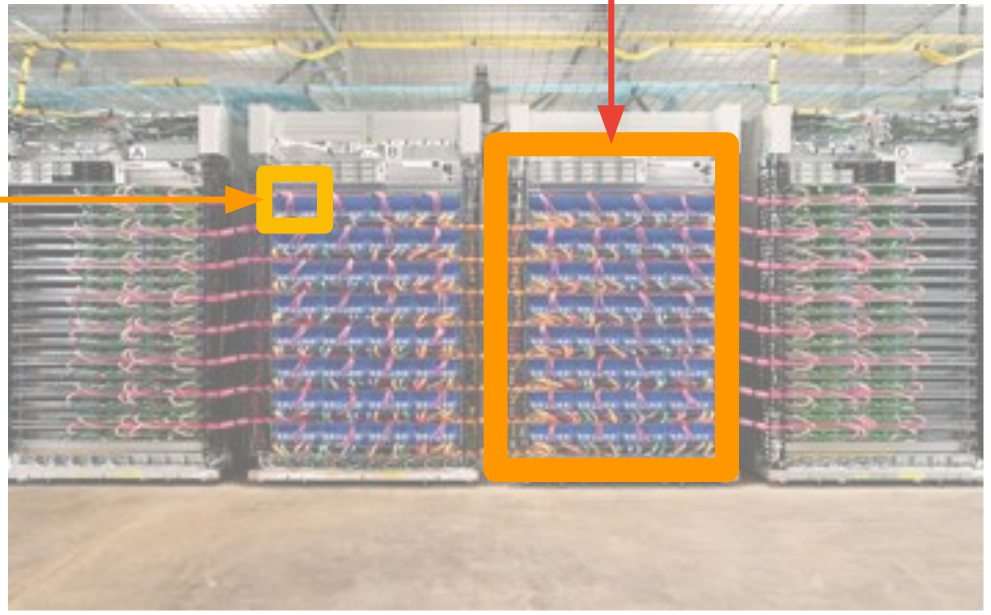
Some points to keep in mind on TPUs

1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core, 64 TPUs/pod

A TPU pod has 64 TPUs

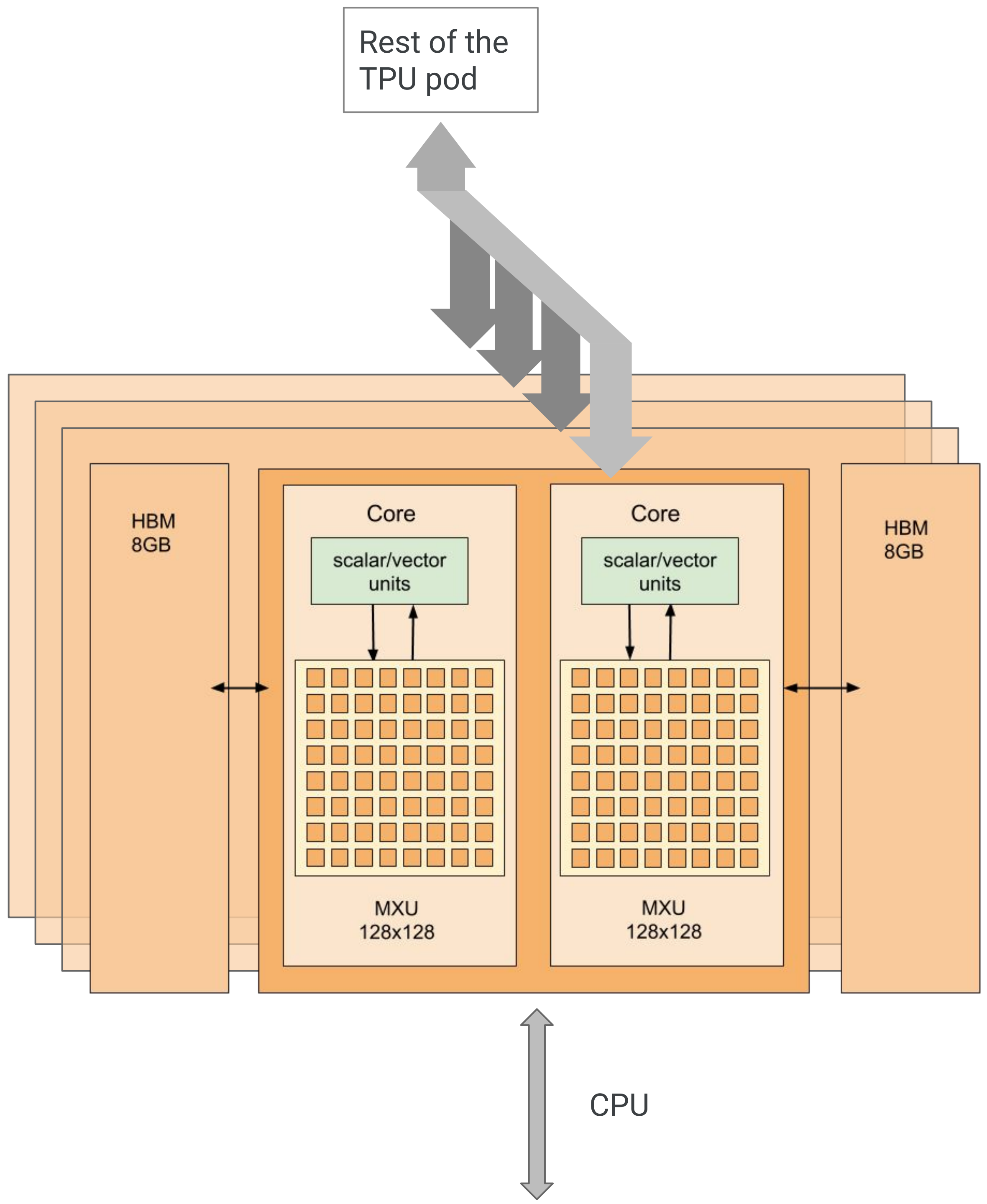


TPU with 4 chips



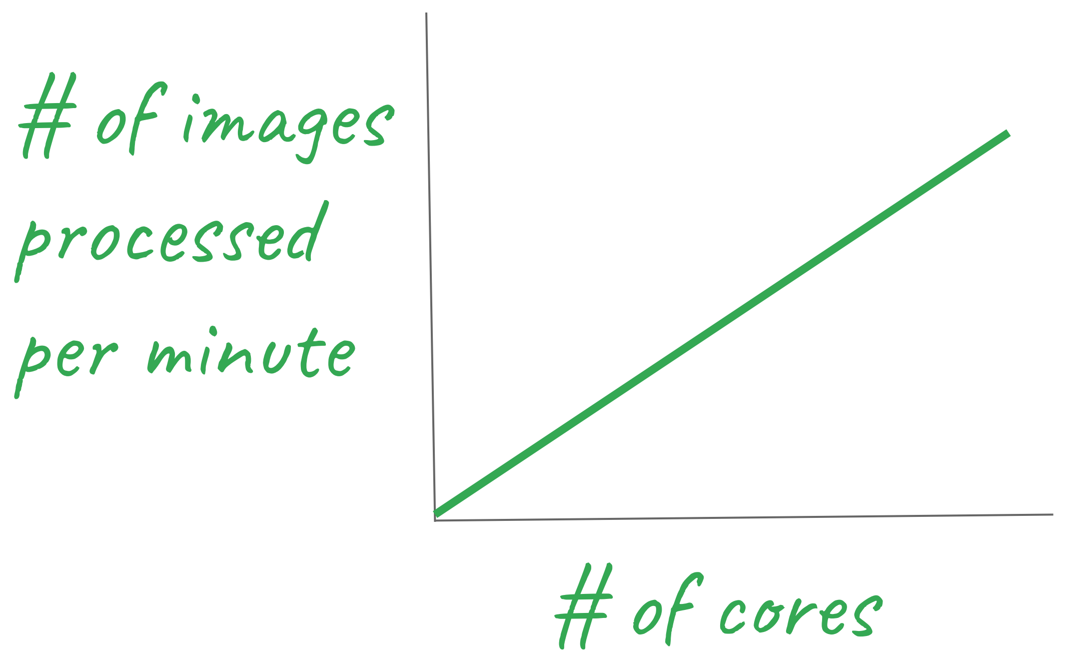
Some points to keep in mind on TPUs

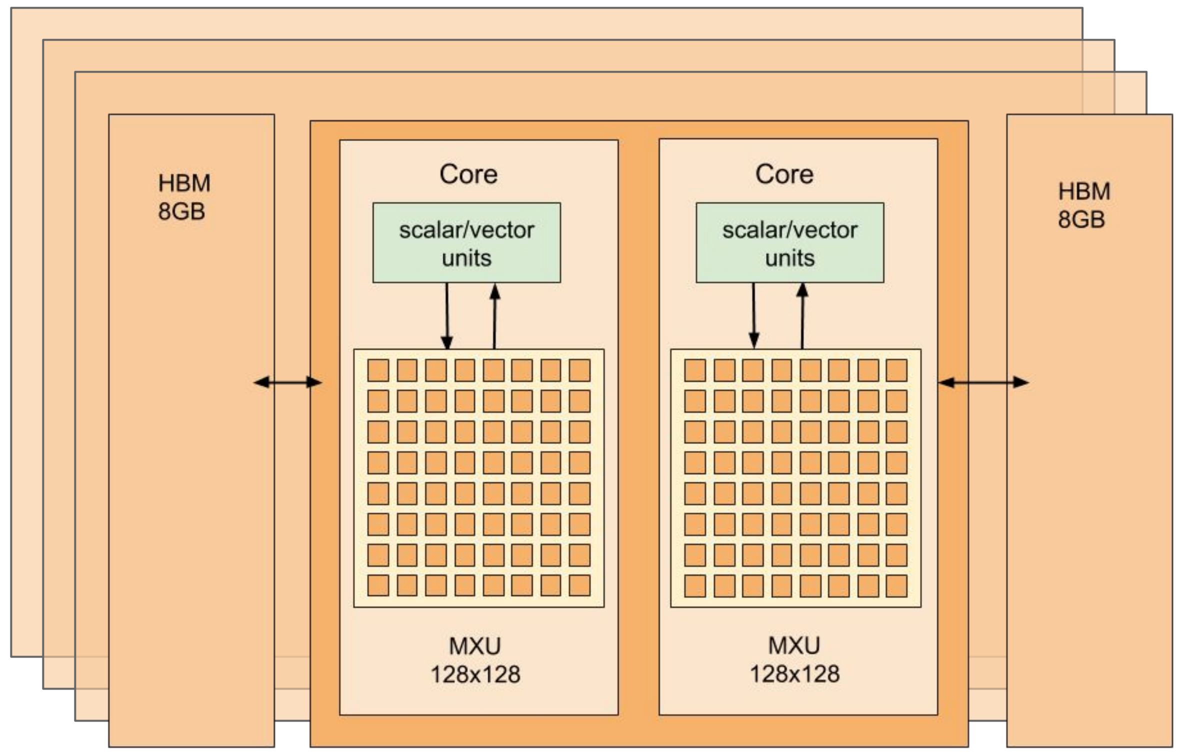
1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core, 64 TPUs/pod
2. High-speed interconnect



Some points to keep in mind on TPUs

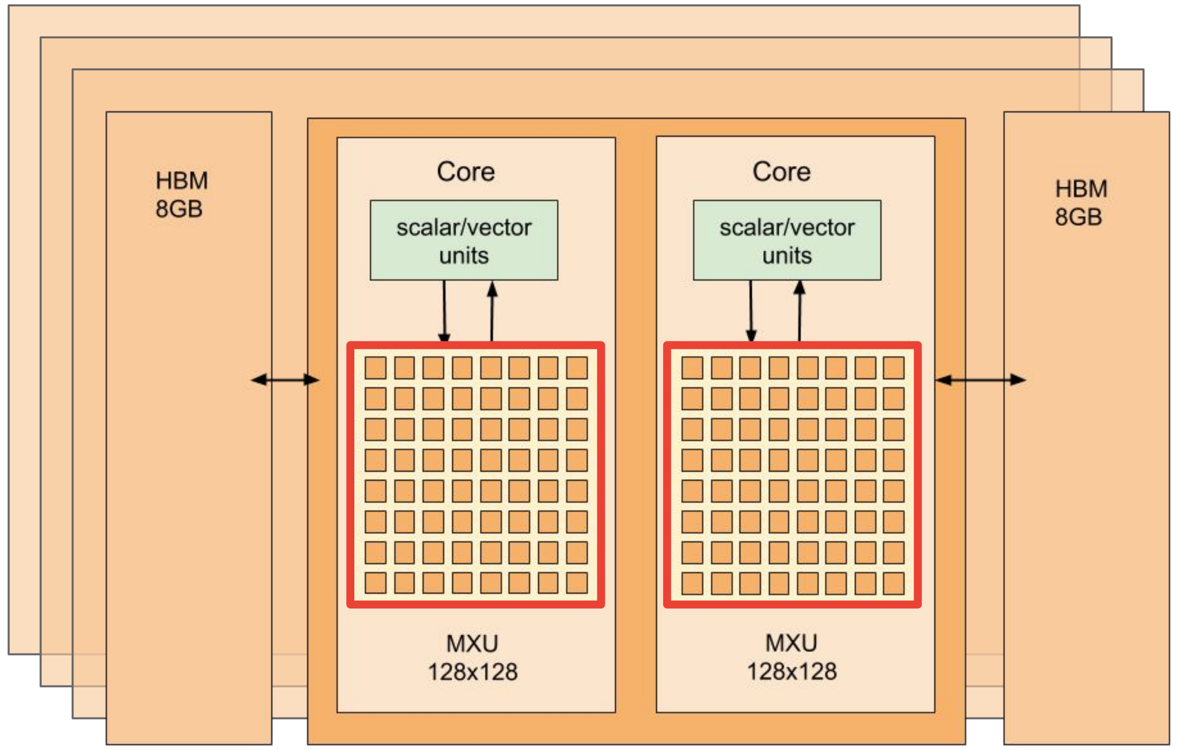
1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core, 64 TPUs/pod
2. High-speed interconnect within a TPU pod





Some points to keep in mind on TPUs

1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core, 64 TPUs/pod
2. High-speed interconnect
3. Very large matrix multiplication hardware



Some points to keep in mind on TPUs

1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core, 64 TPUs/pod
2. High-speed interconnect
3. Very large matrix multiplication hardware
4. Specialized instruction set

Some points to keep in mind on TPUs

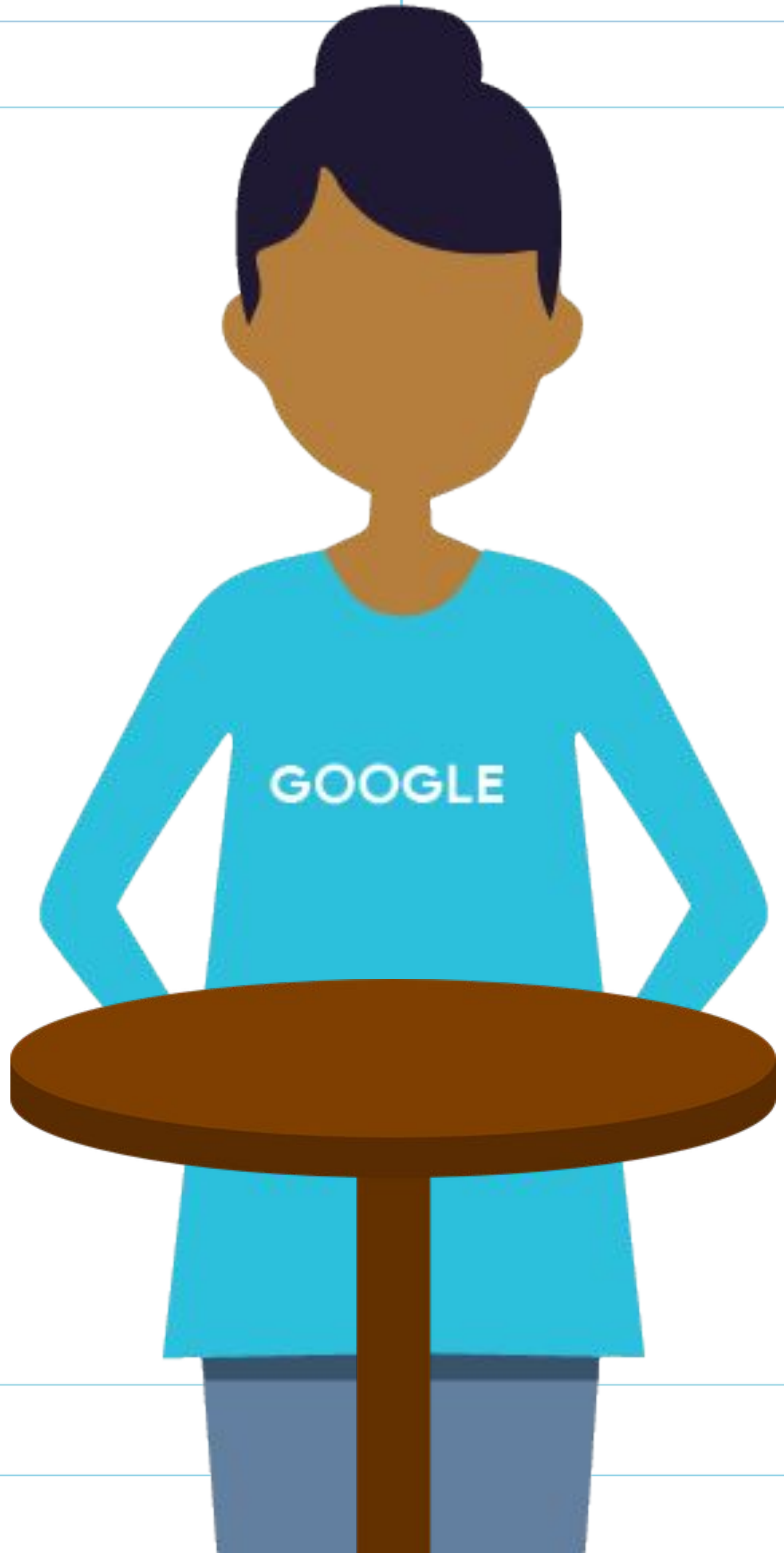
1. 4 chips/TPU, 2 cores/chip, and 8 GB Memory/core, 64 TPUs/pod
2. High-speed interconnect
3. Very large matrix multiplication hardware
4. Specialized instruction set
5. bfloat or float32 floating point representation

File Name: T-ICML-O_4_l8_demo:_tpu_estimator

Format: Presenter in Studio

Title: Demo: TPU Estimator

Presenter: Lak



Title Safe >

< Action Safe

File Name: T-ICML-O_4_l9_neural_architecture_search

Format: Presenter in Studio

Title: Neural Architecture Search

Presenter: Lak

Agenda

Introduction

Batch Normalization

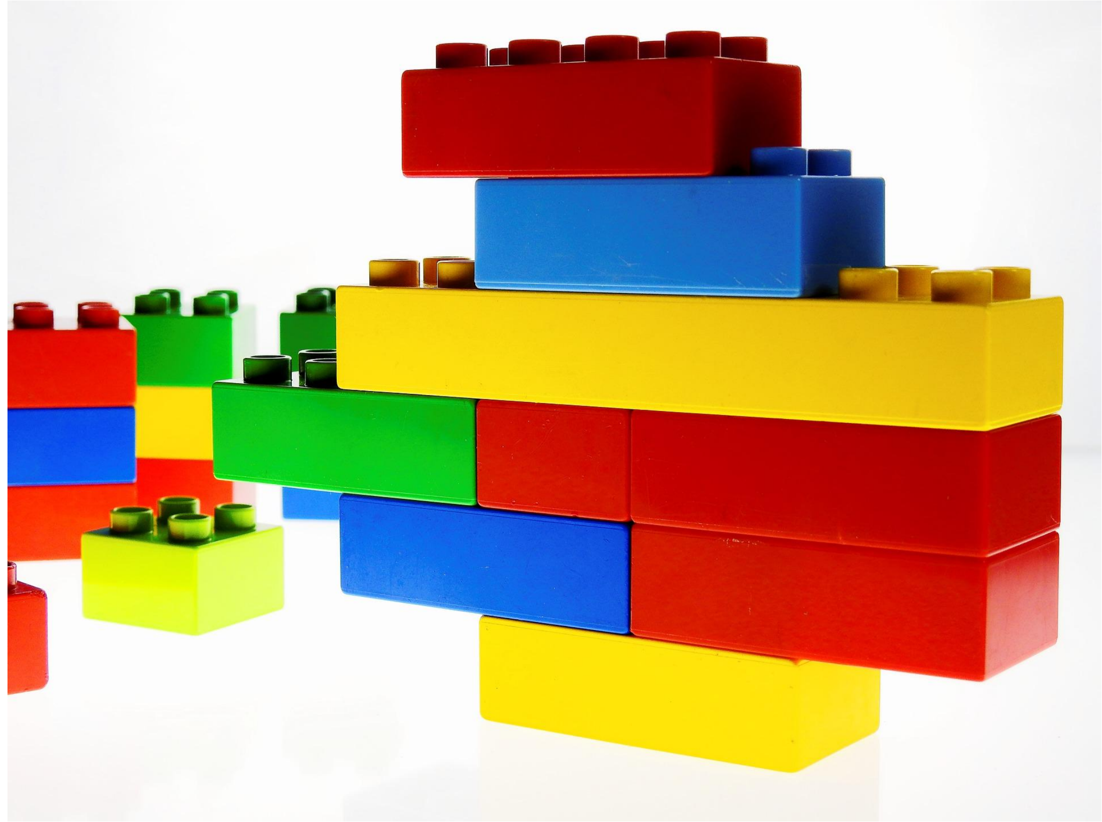
Residual Networks

AI Accelerators

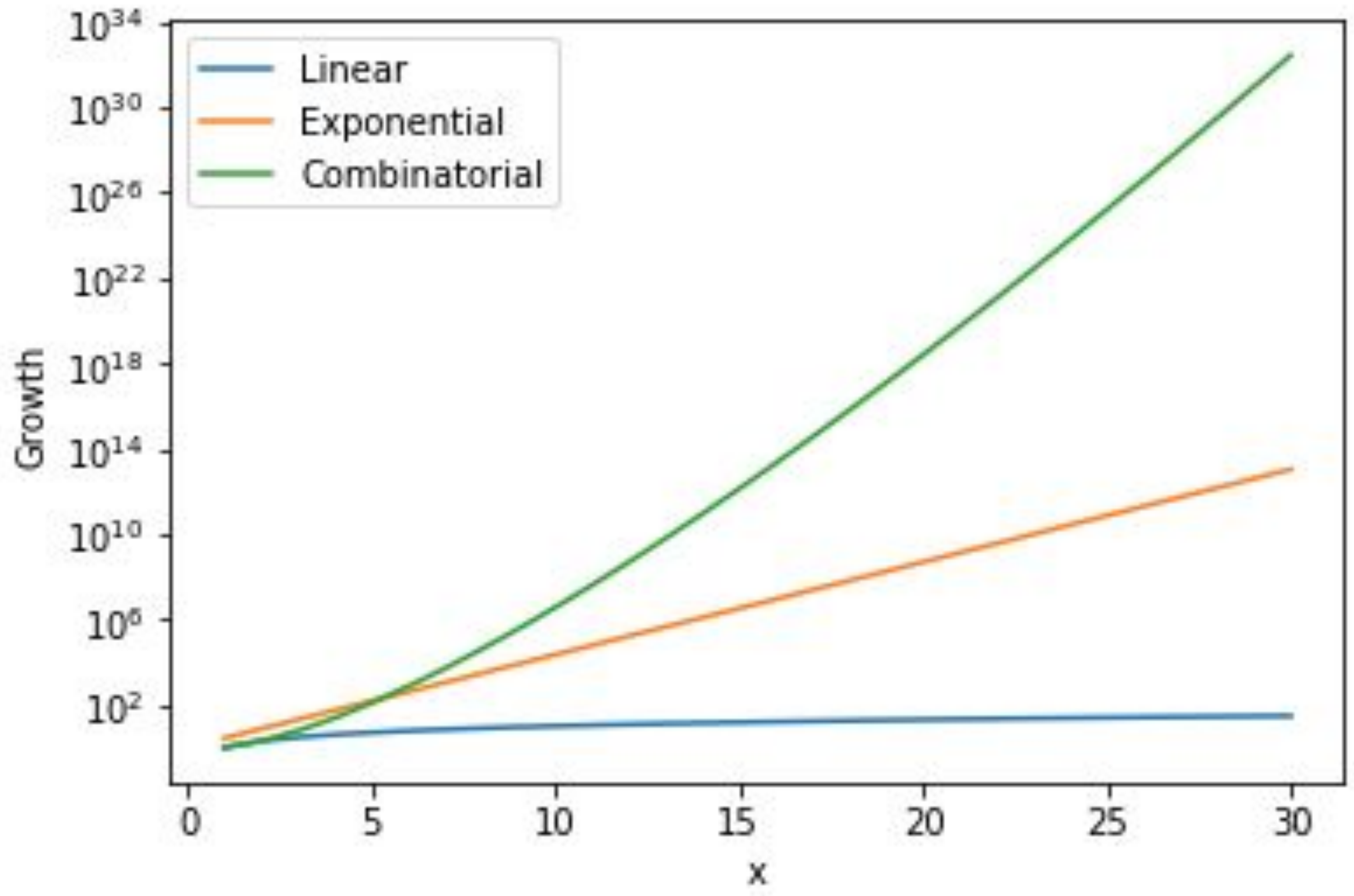
Architecture Search

Painstaking design + hyperparameter tuning

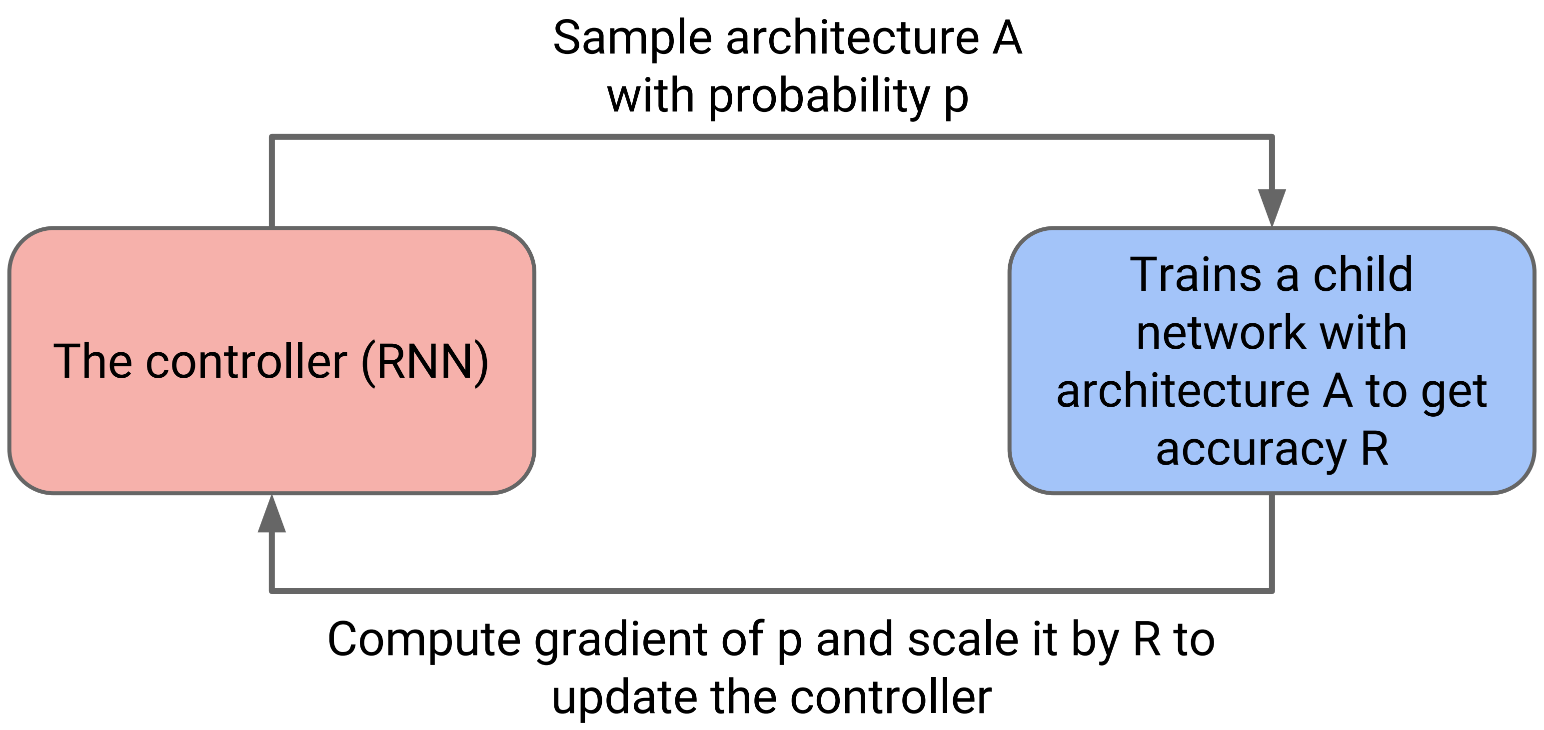
Automate the building of models?



Combinatorial Explosion

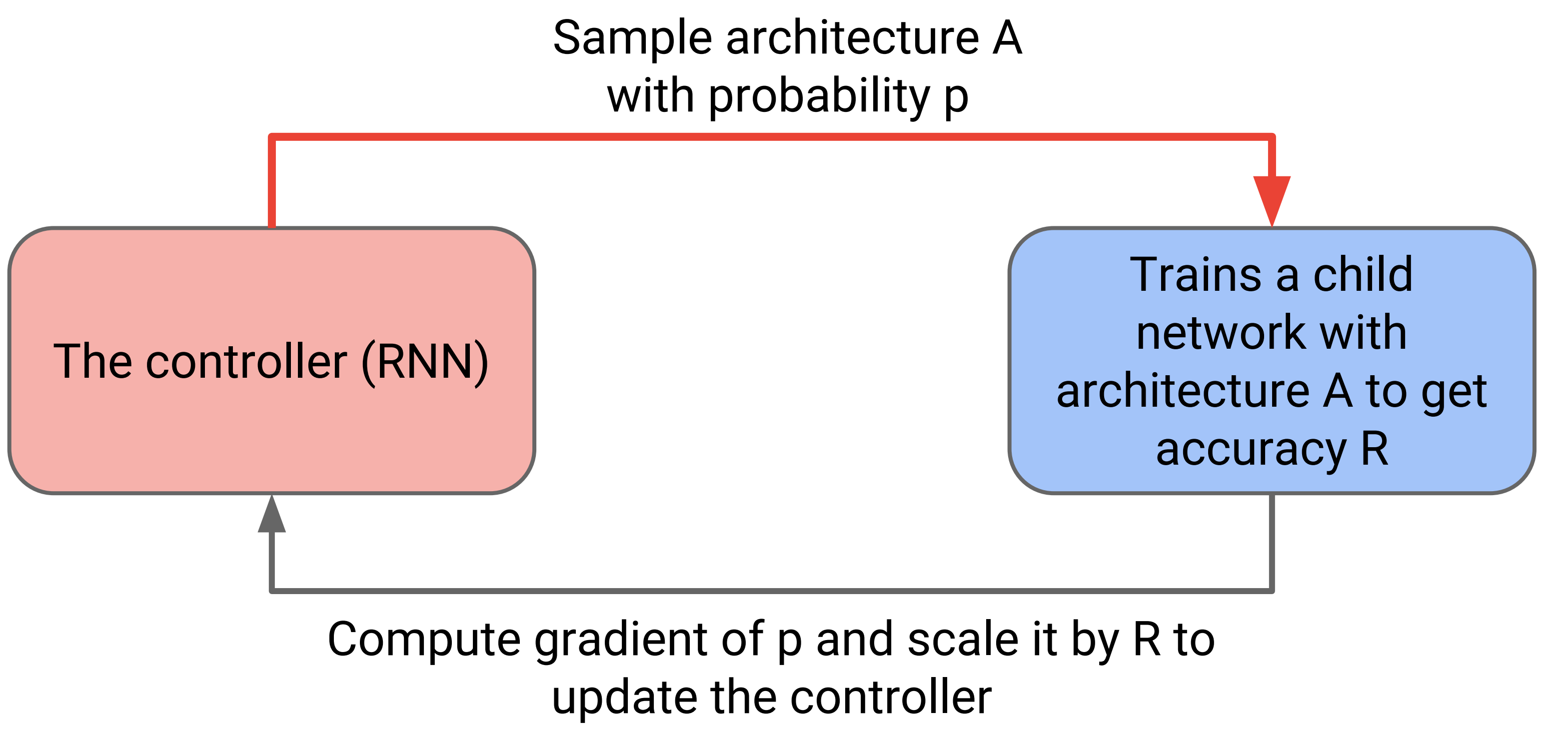


Reinforcement Learning to design neural networks



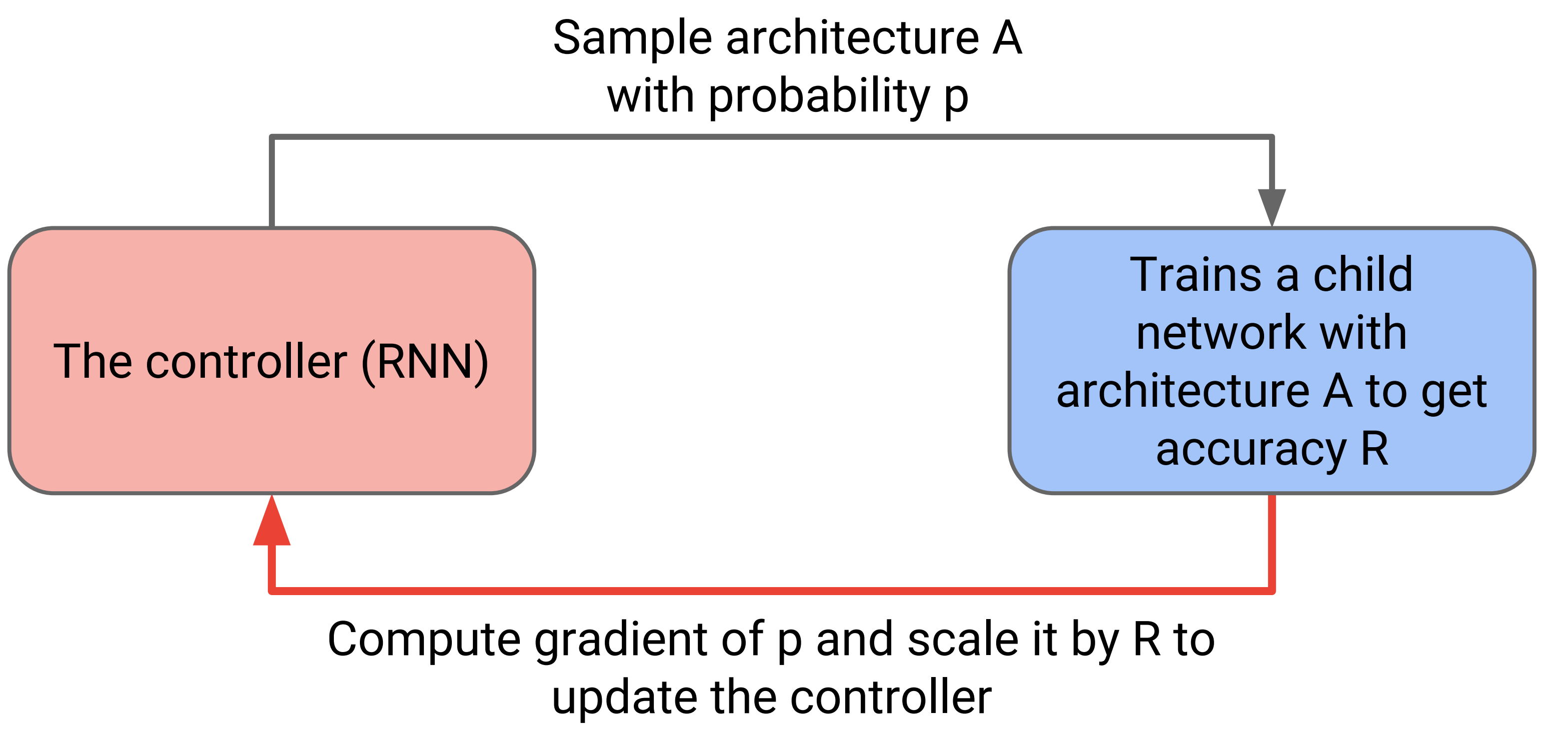
https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html

Reinforcement Learning to design neural networks



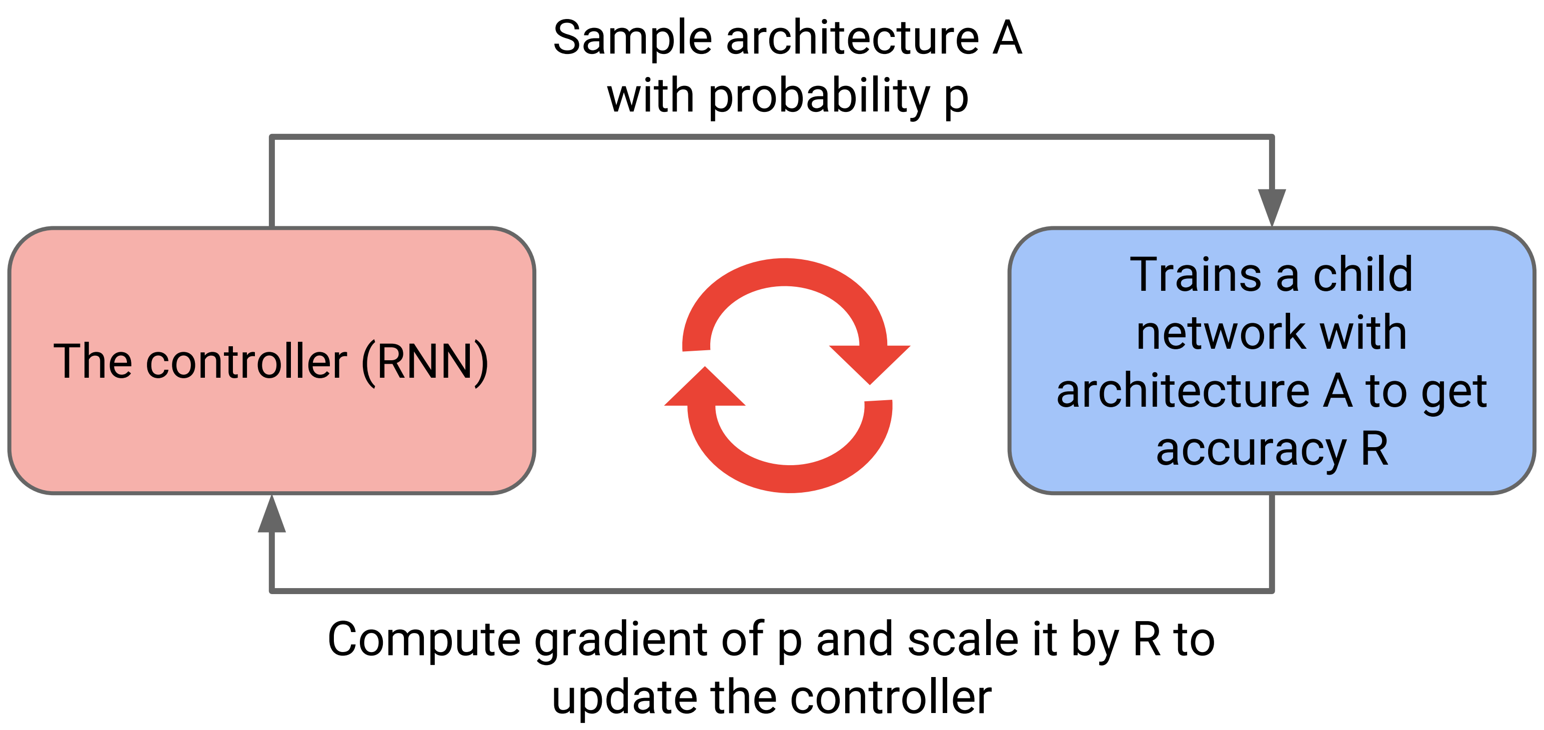
https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html

Reinforcement Learning to design neural networks



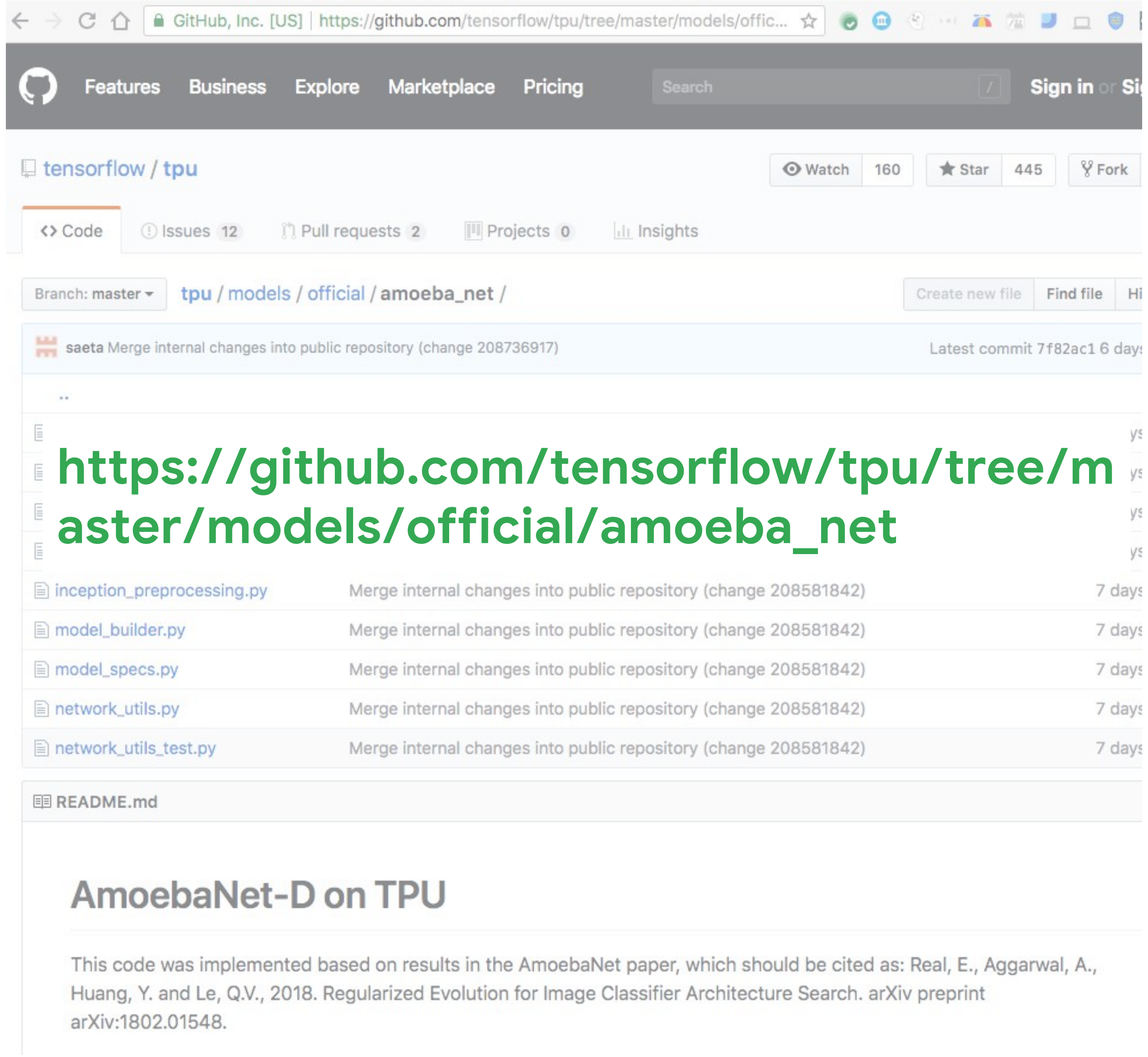
https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html

Reinforcement Learning to design neural networks



https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html

AmoebaNet-D



https://github.com/tensorflow/tpu/tree/master/models/official/amoeba_net

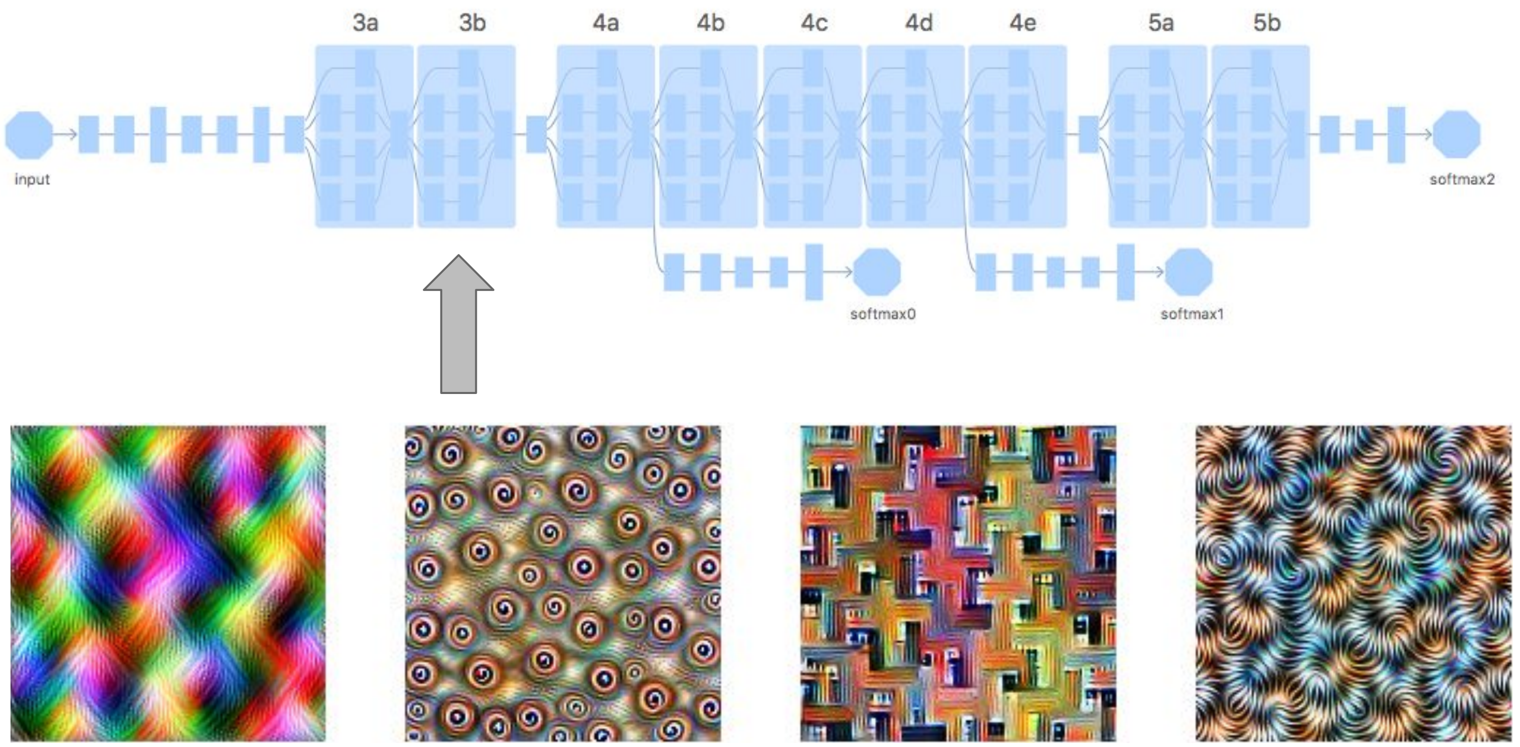
File Name: T-ICML-O_4_I10_summary

Format: Presenter in Studio

Title: Summary

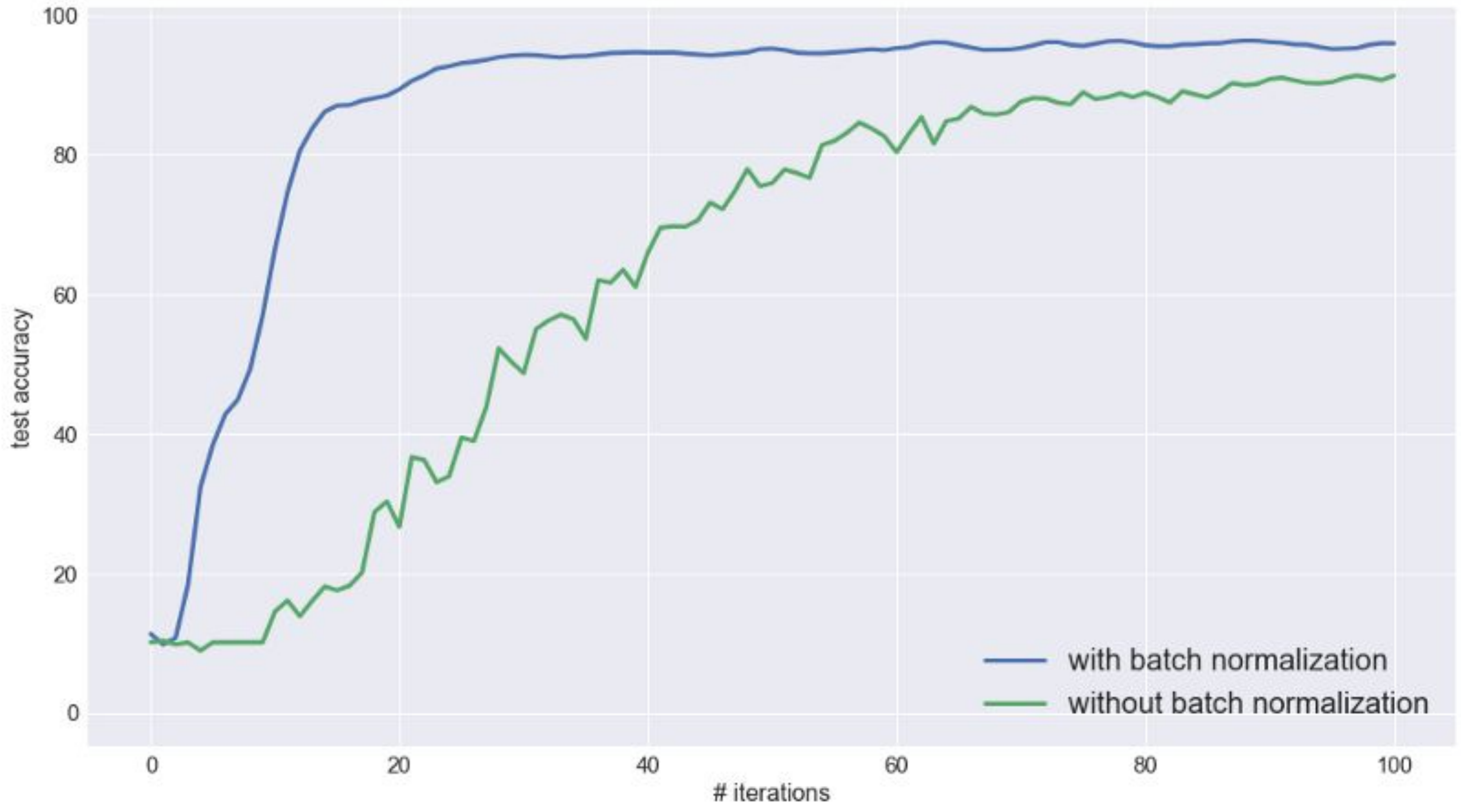
Presenter: Lak

Visualizing layer activations

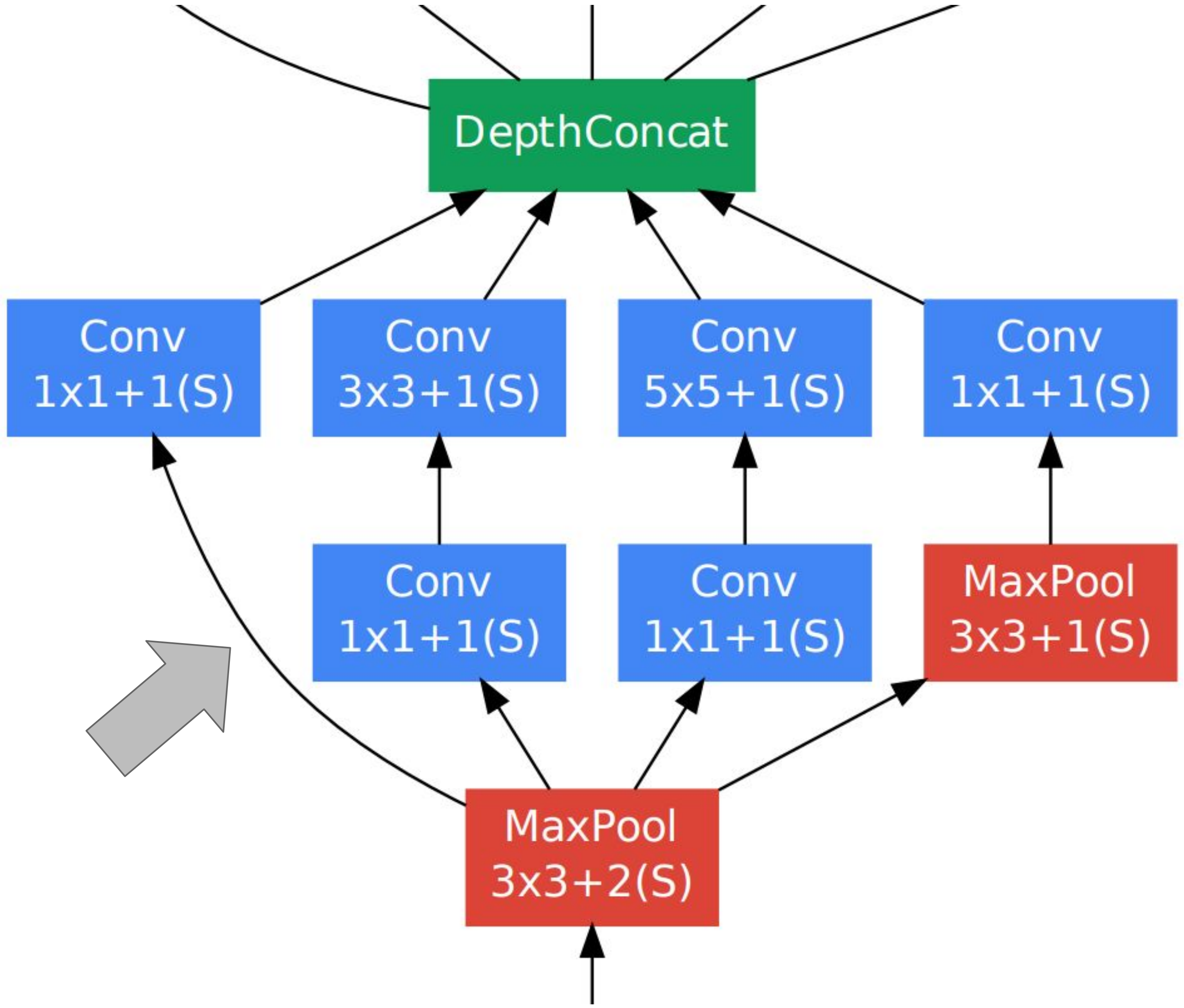


Diagrams by Chris Olah
https://distill.pub/2017/feature-visualization/appendix/

Batch normalization helps you train faster

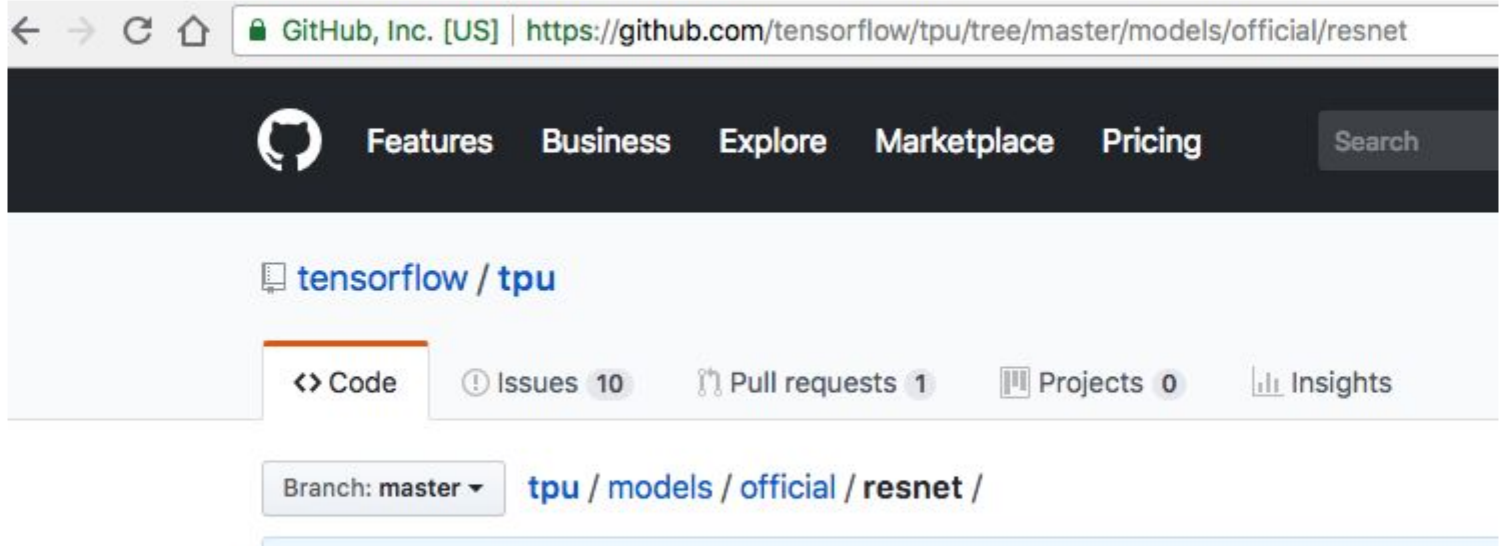


Parallel paths and shortcuts



TPUs are ASICs

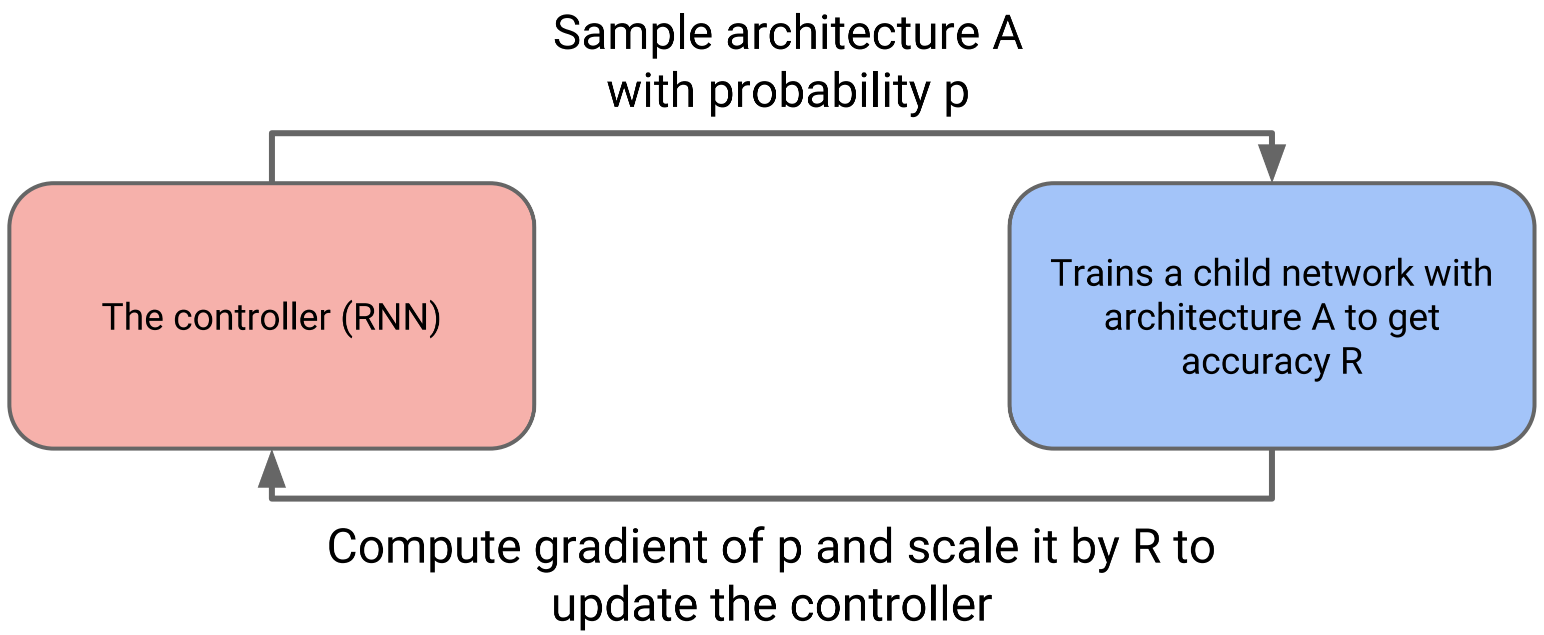
ResNet on TPU: Code



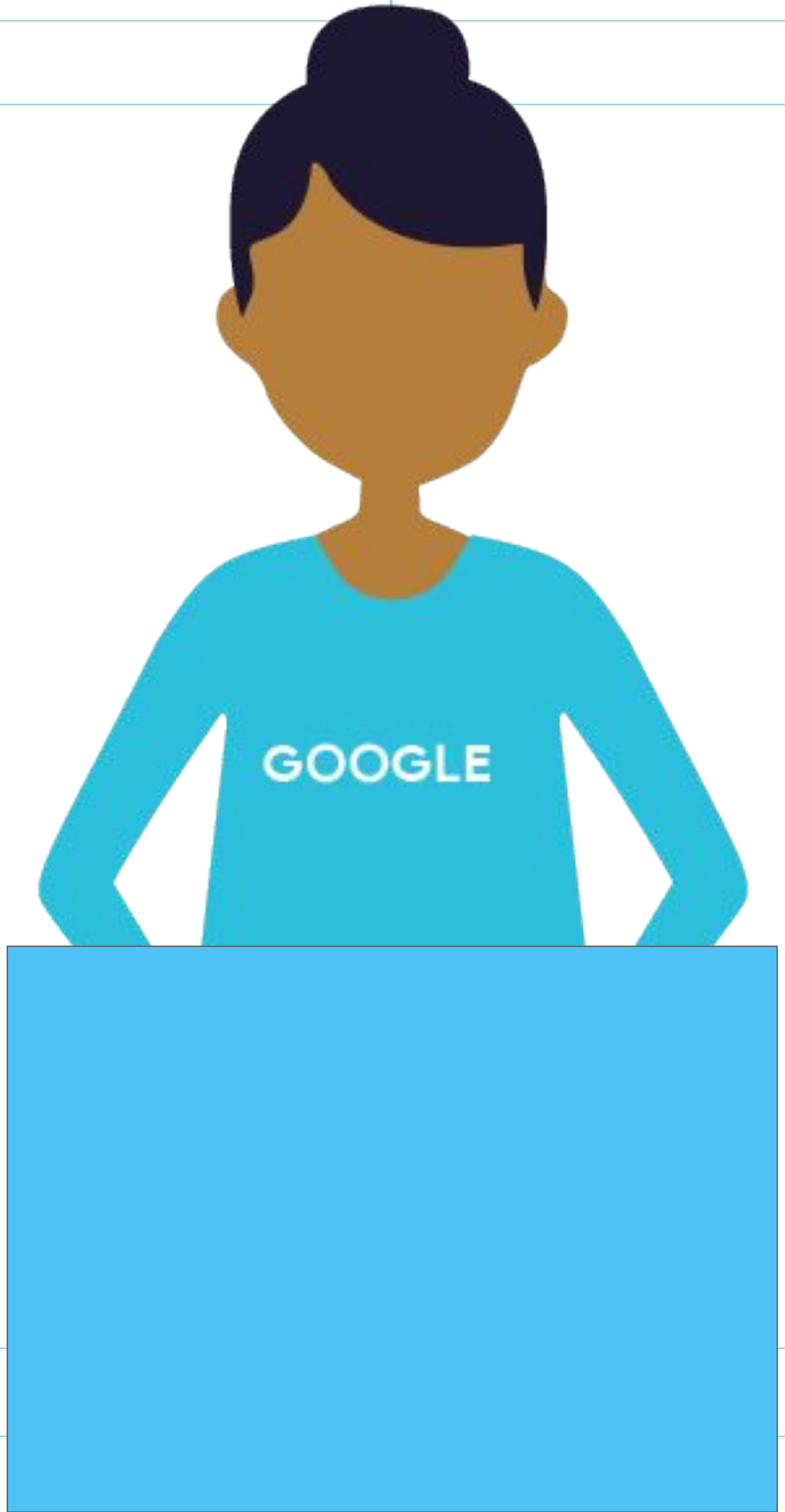
TPU for custom models

```
optimizer = tf.contrib.tpu.CrossShardOptimizer(optimizer)


return tf.contrib.tpu.TPUEstimatorSpec(  # TPU change 2
 …

# change 3
iterations_per_loop = 1000
tpu_cluster_resolver =
  tf.contrib.cluster_resolver.TPUClusterResolver(
    hparams['tpu'],
    zone=hparams['tpu_zone'],
    project=hparams['project'])

config = tf.contrib.tpu.RunConfig(
    cluster=tpu_cluster_resolver,
    model_dir=output_dir,
    save_checkpoints_steps=max(600, iterations_per_loop),
    tpu_config=tf.contrib.tpu.TPUConfig(
      iterations_per_loop=iterations_per_loop,
      per_host_input_for_training=True))
```

Reinforcement Learning to design neural networks



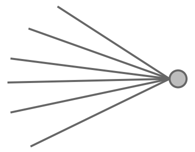
https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html



Title Safe >

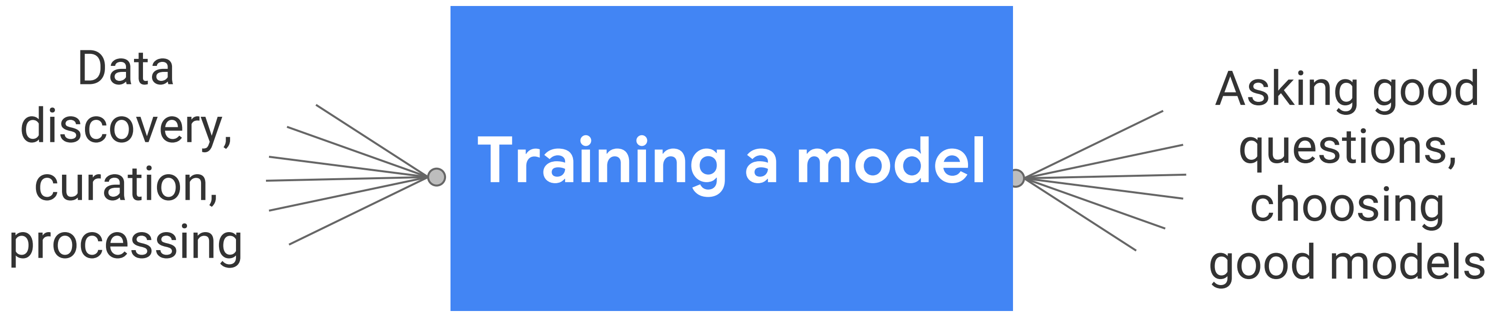
< Action Safe

Data
discovery,
curation,
processing



Data discovery, curation, processing

Asking good questions, choosing good models


Data discovery, curation, processing
Training a model
Asking good questions, choosing good models