Elastic Distributed Training: Learning in the Limbo of Resources

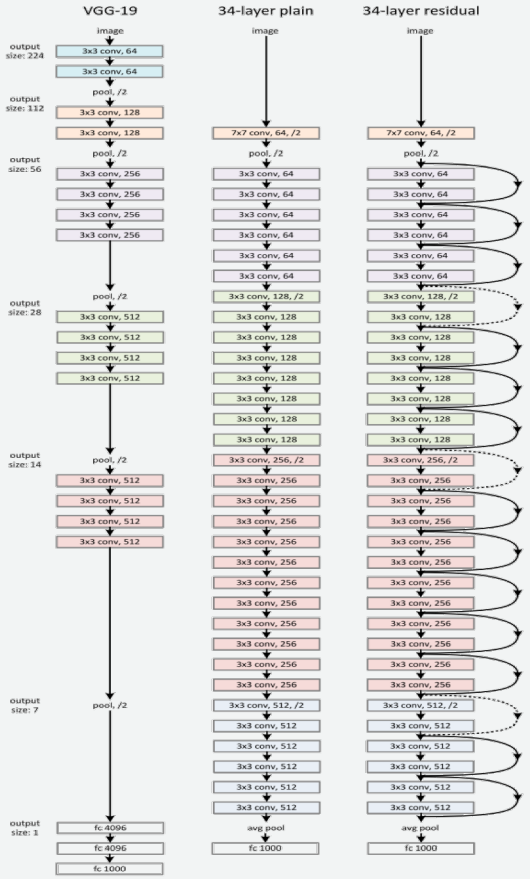
Hang Zhang
Applied Scientist, Amazon Web Services.

July 10, 2019

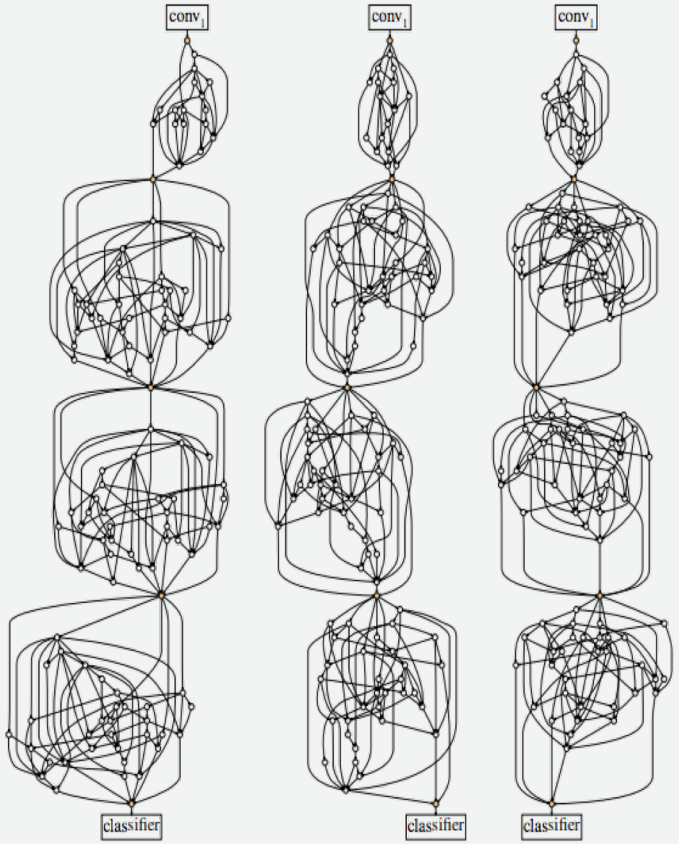
Deeper and larger SOTA models

Inception

ResNet



RandWire



Observations

- To launch a training job with multiple machines:
  - Wait for all resources to be ready
  - Reserve in advance
  - Stop other low-priority jobs
- Can we:
  - Start early with partial resources
  - Preempt partial resources for low-priority jobs instead of stopping them
  - Increase the utilization and reduce the cost

aws

Amazon EC2 Spot Instances

- Spot Instance up to 90% off compared to On-Demand instance.
- Can we use Spot Instance to train deep neural networks?
  - Existing Solution: checkpoint and resuming (fixed resource)
  - Can we dynamically use available spot instances?

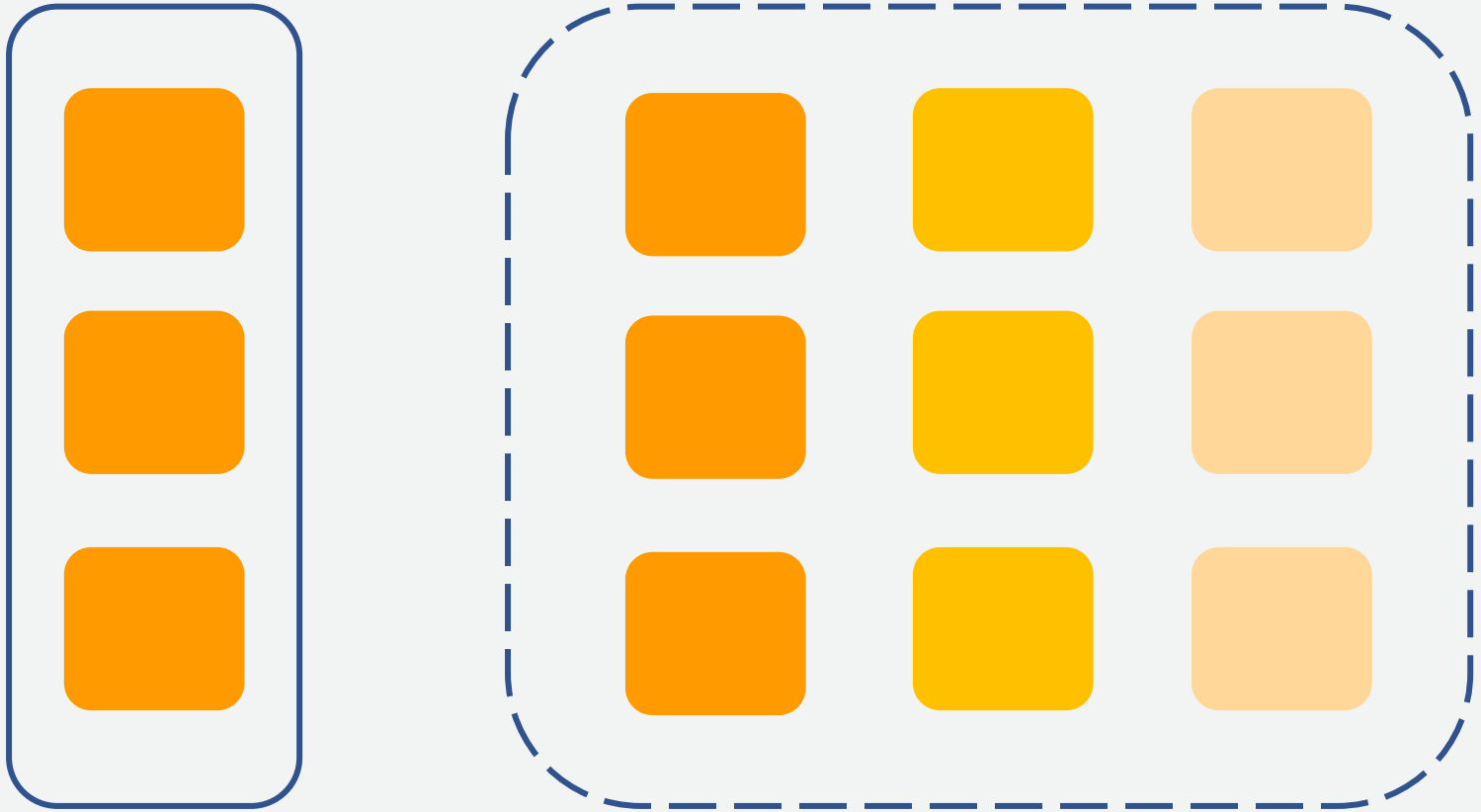
Similarly, Google Cloud offers preemptible virtual machine and Microsoft Azure has low-priority virtual machines.

aws

The Dynamic Env: Elastic Distributed Training

Dedicated Instances

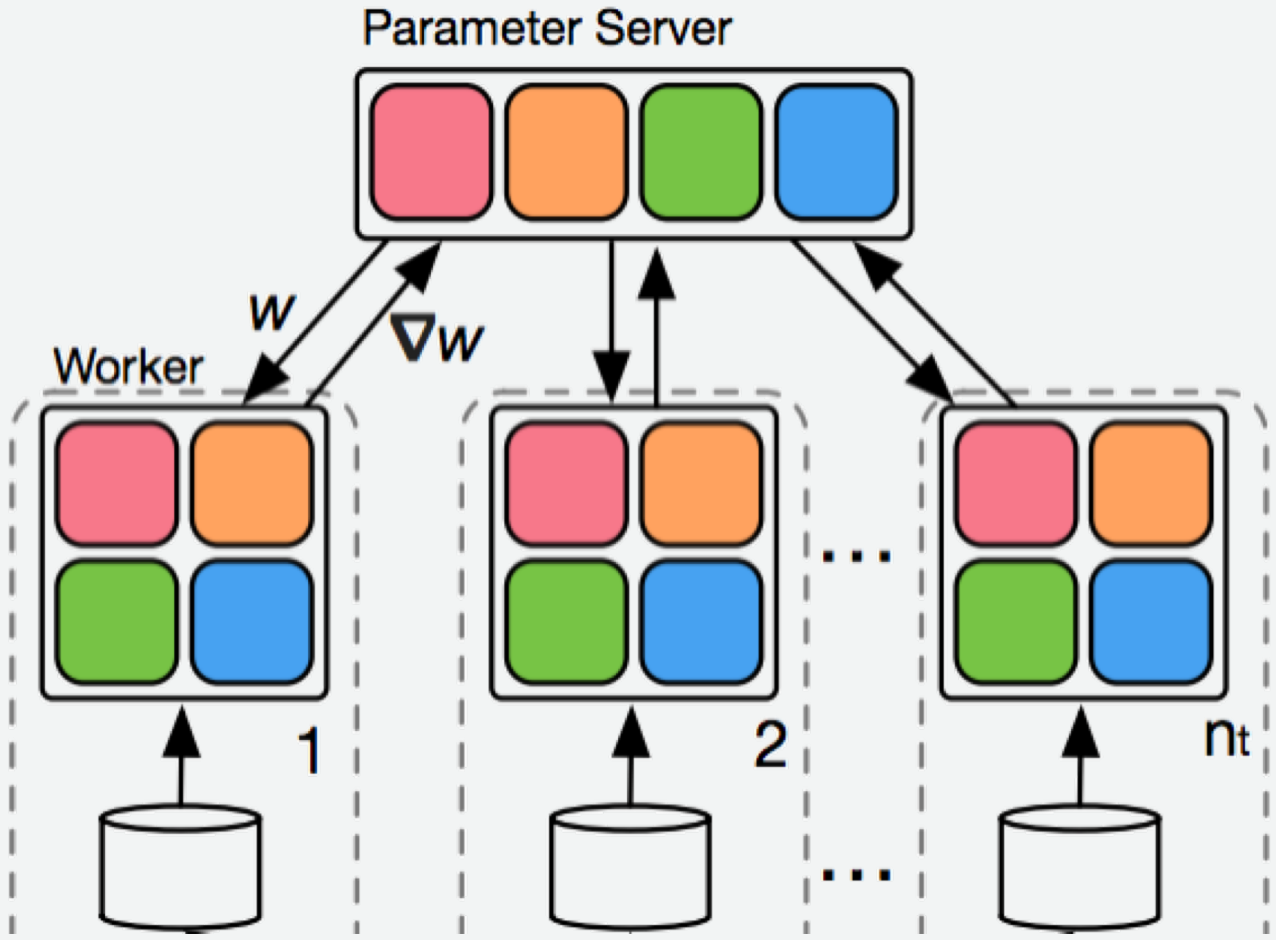
Preemptive Instances



Can we use Spot Instance to train deep neural networks? (w/o sacrificing the model performance)

aws

Data Parallelism in Deep Learning System



- Model is replicated
- A mini-batch of data is distributed
- Gradients are calculated on each worker
- Average the gradients and update parameters
- System Latency

aws

A “Simple” Solution for Elastic Training, Fix Mini-batch Size

- Advantage:
  - Not changing the optimization process using SGD
- Difficulty:
  - Main Communication overhead (communication vs. computation)

Method	Scale	Throughput
Static Baseline	1×	4944
Fixed Mini-batch Size	12×	335

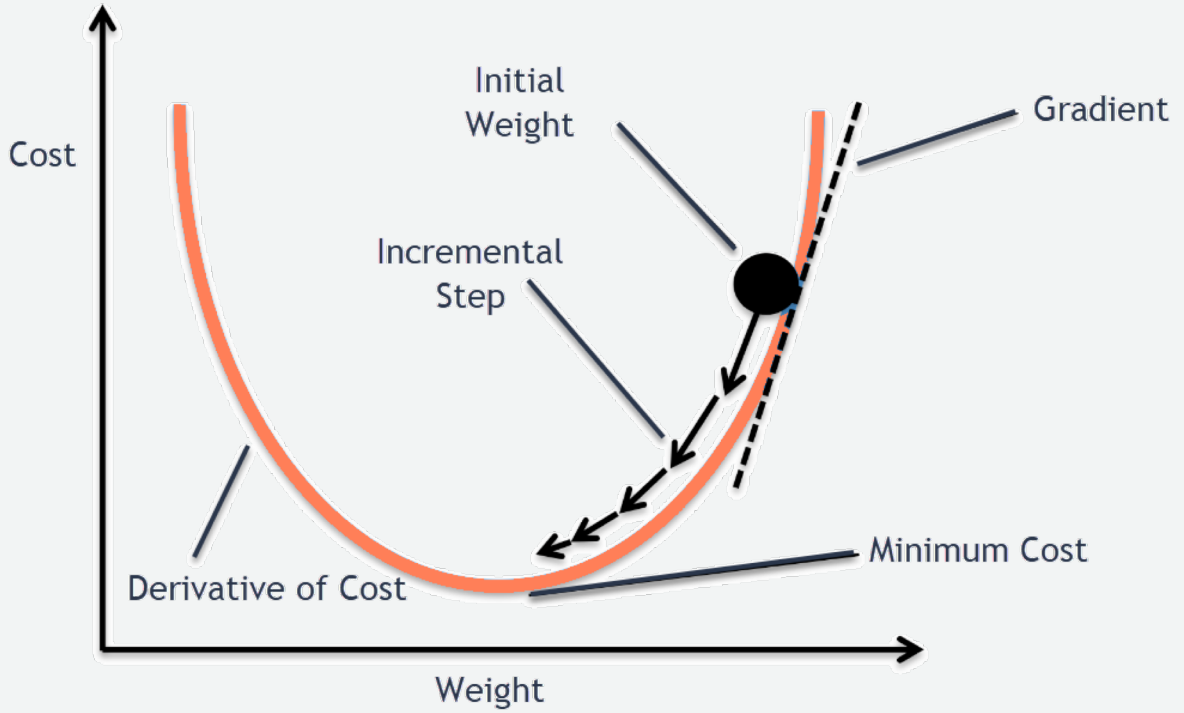
Fix Per-work Batch Size

- Advantage:
  - Scalable with good speedup (computation>communication)
- Difficulty:
  - Model convergence using momentum SGD
  - Guide for adapting the learning rate

Mini-batch SGD

- SGD:

$$w_{t+1} = w_t - \eta \frac{1}{B} \sum_{i=1}^{B} \nabla l(w_t, x_i),$$

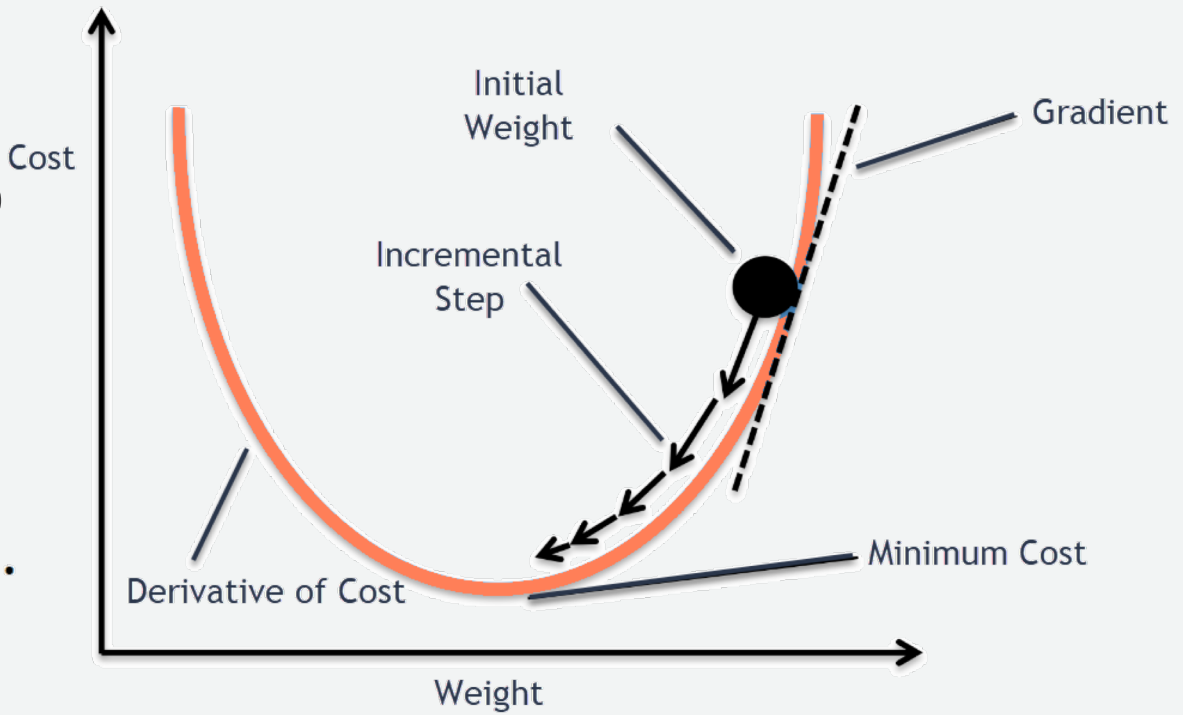


Scaling LR with Mini-batch Size

- Fix number of epochs, and increase mini-batch size from B to kB
- Assuming gradient is smooth (up to k^*): $\ell(w_t, x) \approx \ell(w_{t+j}, x)$ for $j < k$
- For Vanilla SGD:

$$w_{t+k} = w_t - \eta \frac{1}{n} \sum_{j<k} \sum_{x \in \mathcal{B}_j} \nabla l(x, w_{t+j}).$$

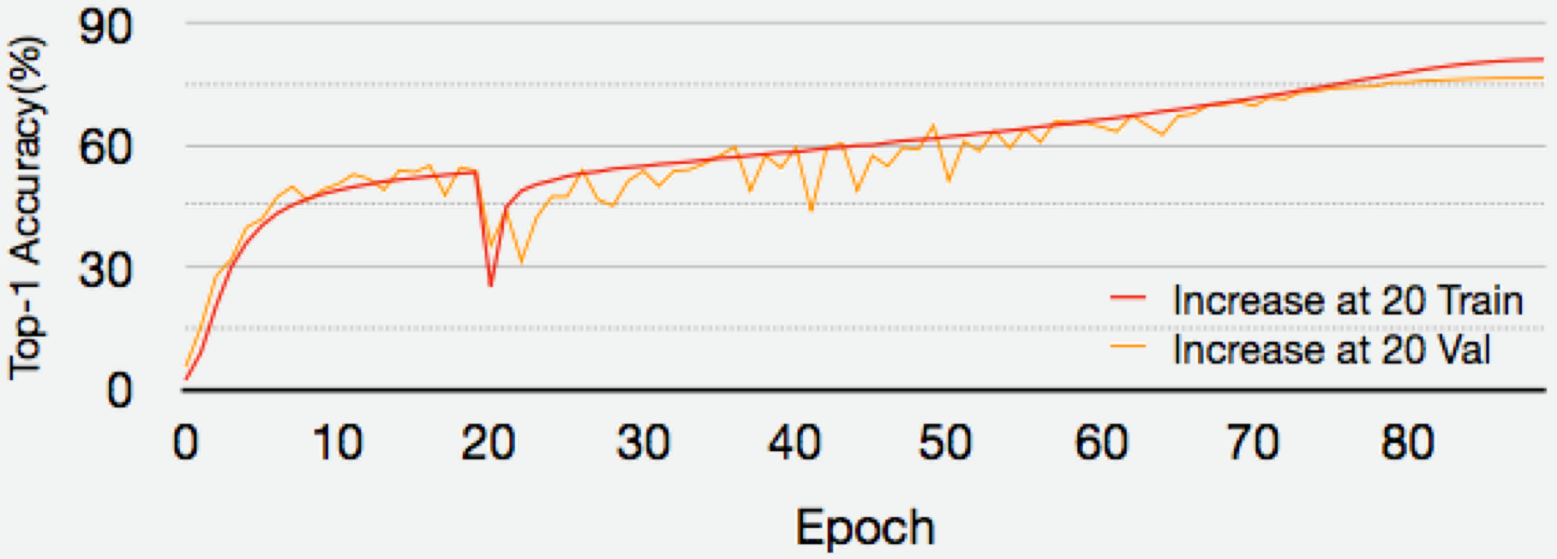
$$\hat{w}_{t+1} = w_t - \hat{\eta} \frac{1}{kn} \sum_{j<k} \sum_{x \in \mathcal{B}_j} \nabla l(x, w_t).$$



aws

Related Work for Large Mini-batch Training

- Linearly Scaling the Learning Rate (Overshooting)



(a) Increase the mini-batch size to 12 times at epoch 20.

aws

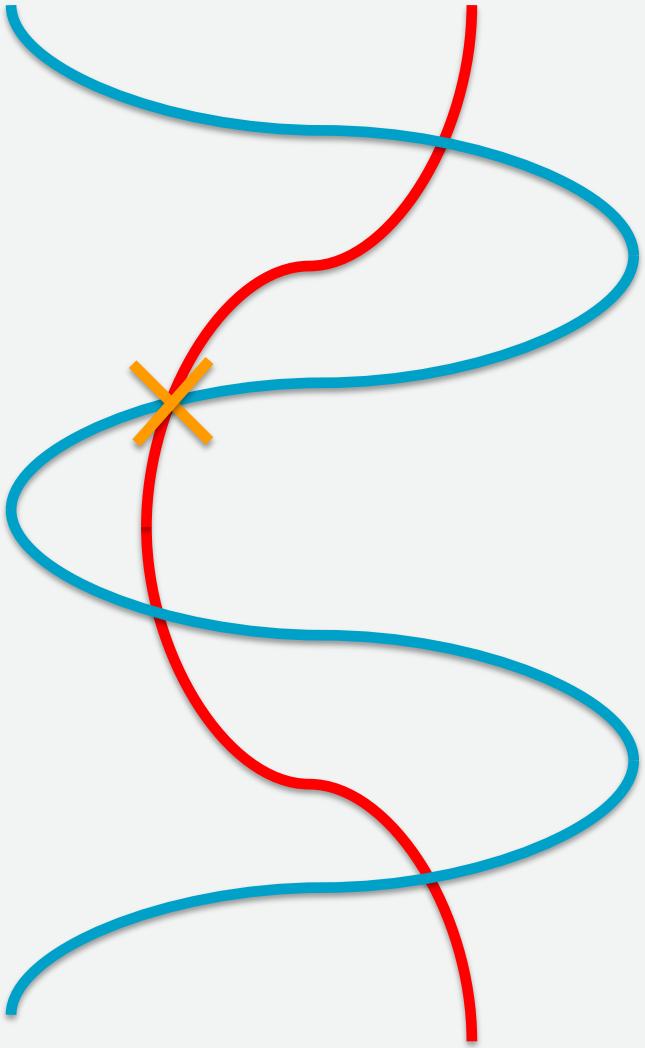
SGD with Momentum Update

- Momentum SGD:

$$u_{t+1} = \mu u_t + \frac{1}{B} \sum_{i=1}^{B} \nabla l(w_t, x_i)$$

$$w_{t+1} = w_t - \eta u_{t+1},$$

Skiing Analogy



- Red: Advanced Skier
- Blue: Beginning Skier

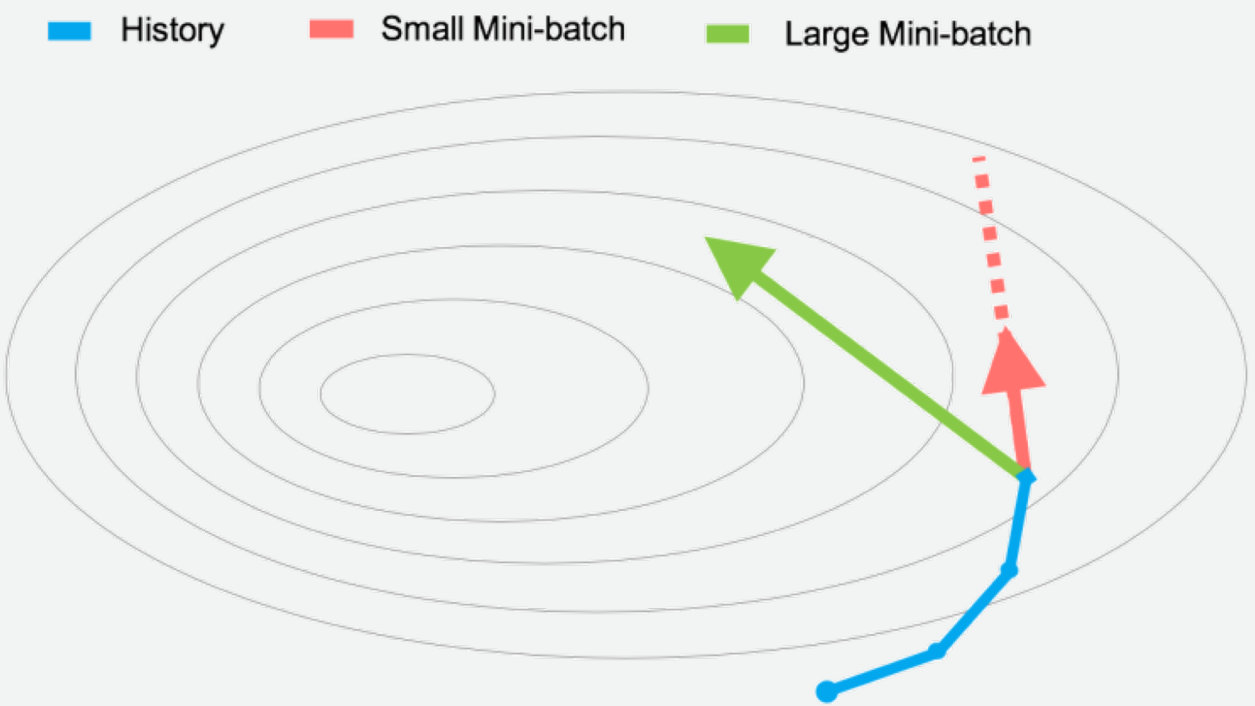
Difficulty of Momentum Update in Elastic Training

- Momentum SGD:

$$u_{t+1} = \mu u_t + \frac{1}{B}\sum_{i=1}^{B} \nabla l(w_t, x_i)$$

$$w_{t+1} = w_t - \eta u_{t+1},$$

- Noise (Variance) in the Gradient and Momentum:
  - $\tau_B \propto \frac{1}{B}$
  - $\tau_{kB} \propto \frac{1}{kB}$



aws

Solution: Dynamic SGD

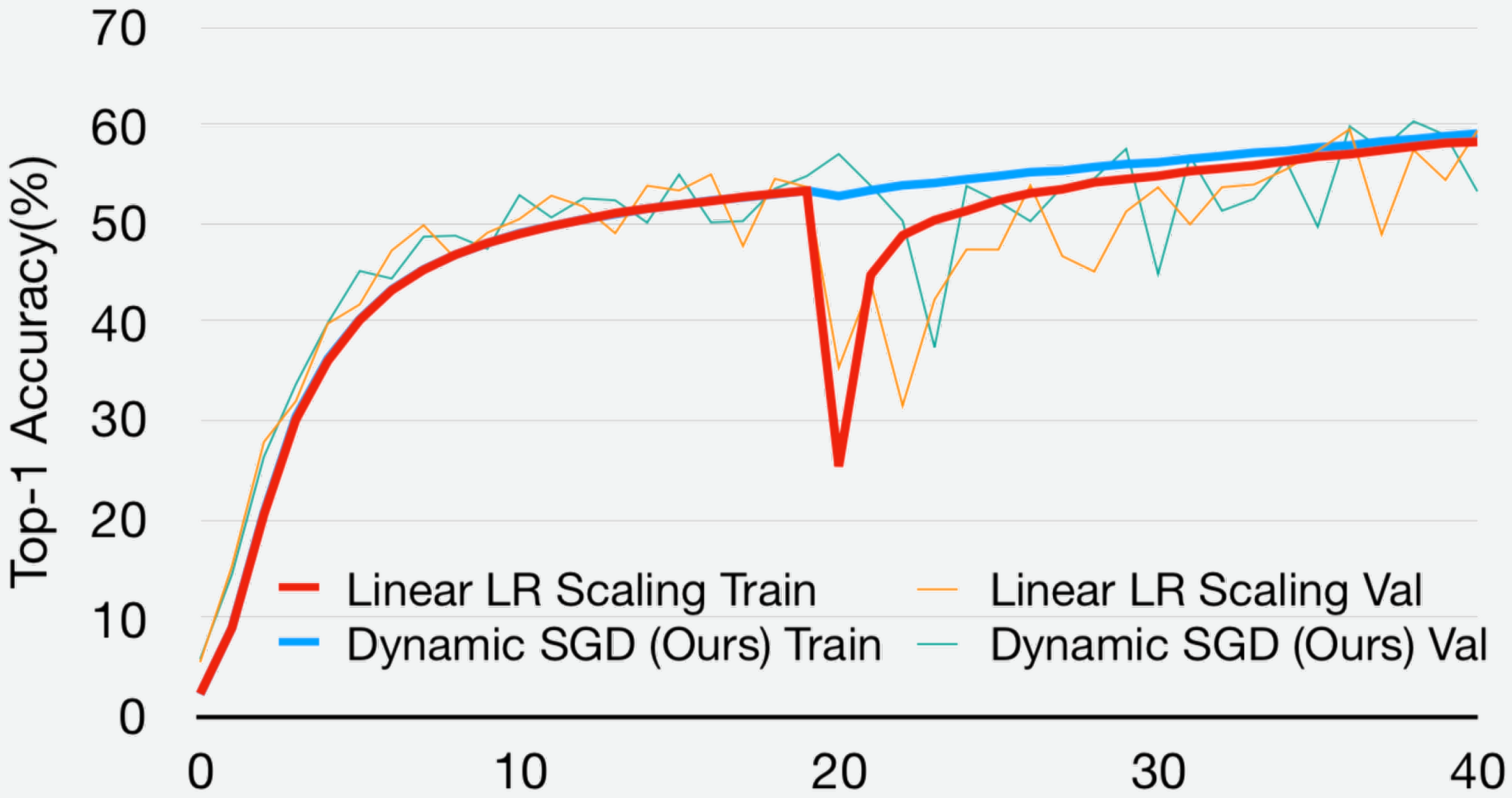
- Momentum Compensation (Warmup)
  Smooth adaptation of the momentum state

$$w_{t+1} = w_t - \gamma_{t+1}\eta u_{t+1}, \text{ where } \gamma_{t+1} \text{ is:}$$

$$\gamma_t = \begin{cases} 1 + \dfrac{t - t_0}{T}(k-1) & \text{if } (t - t_0) < T \\ k & \text{otherwise} \end{cases}$$

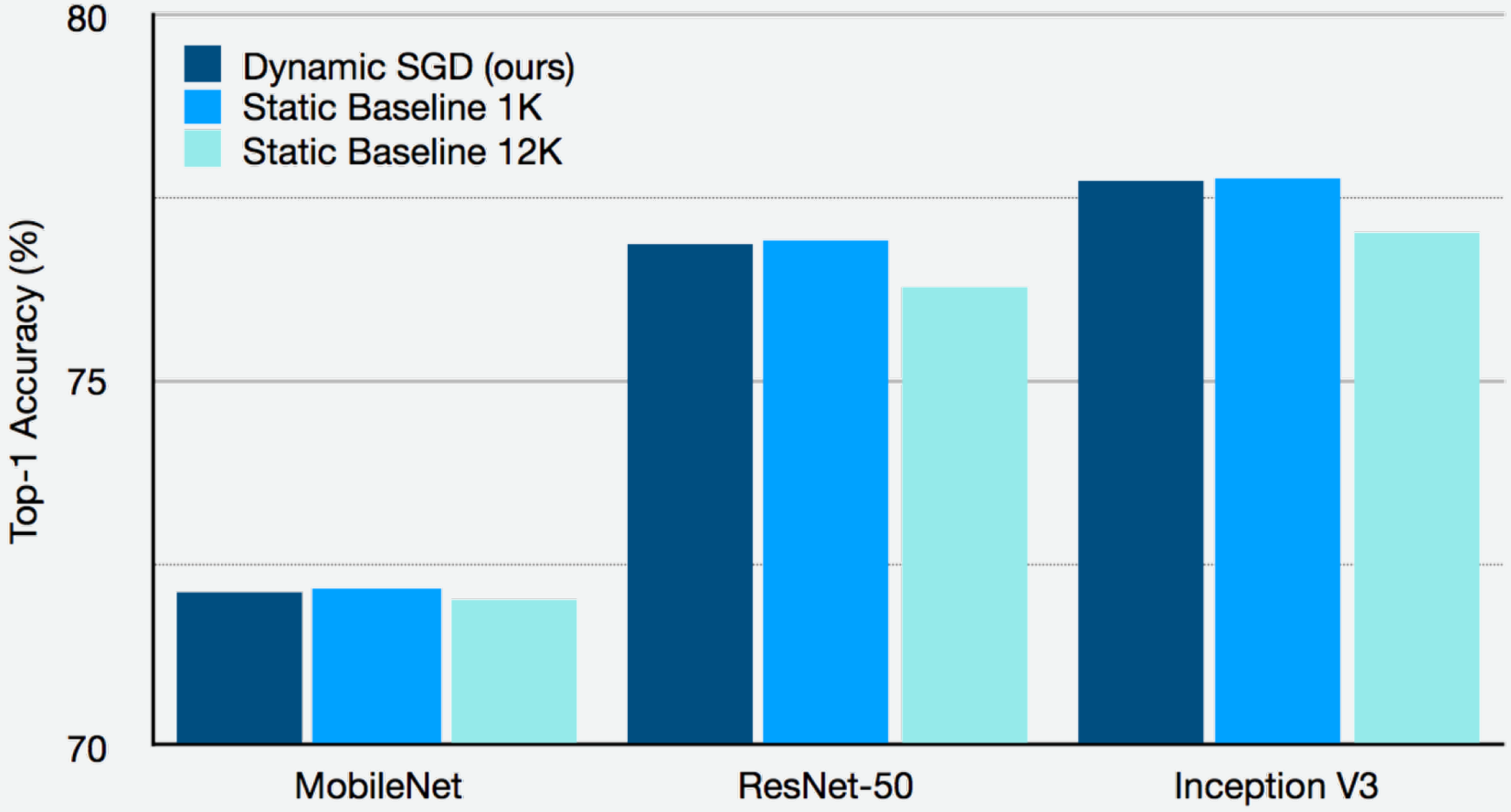
aws

Stabilize the Training Using Our Method



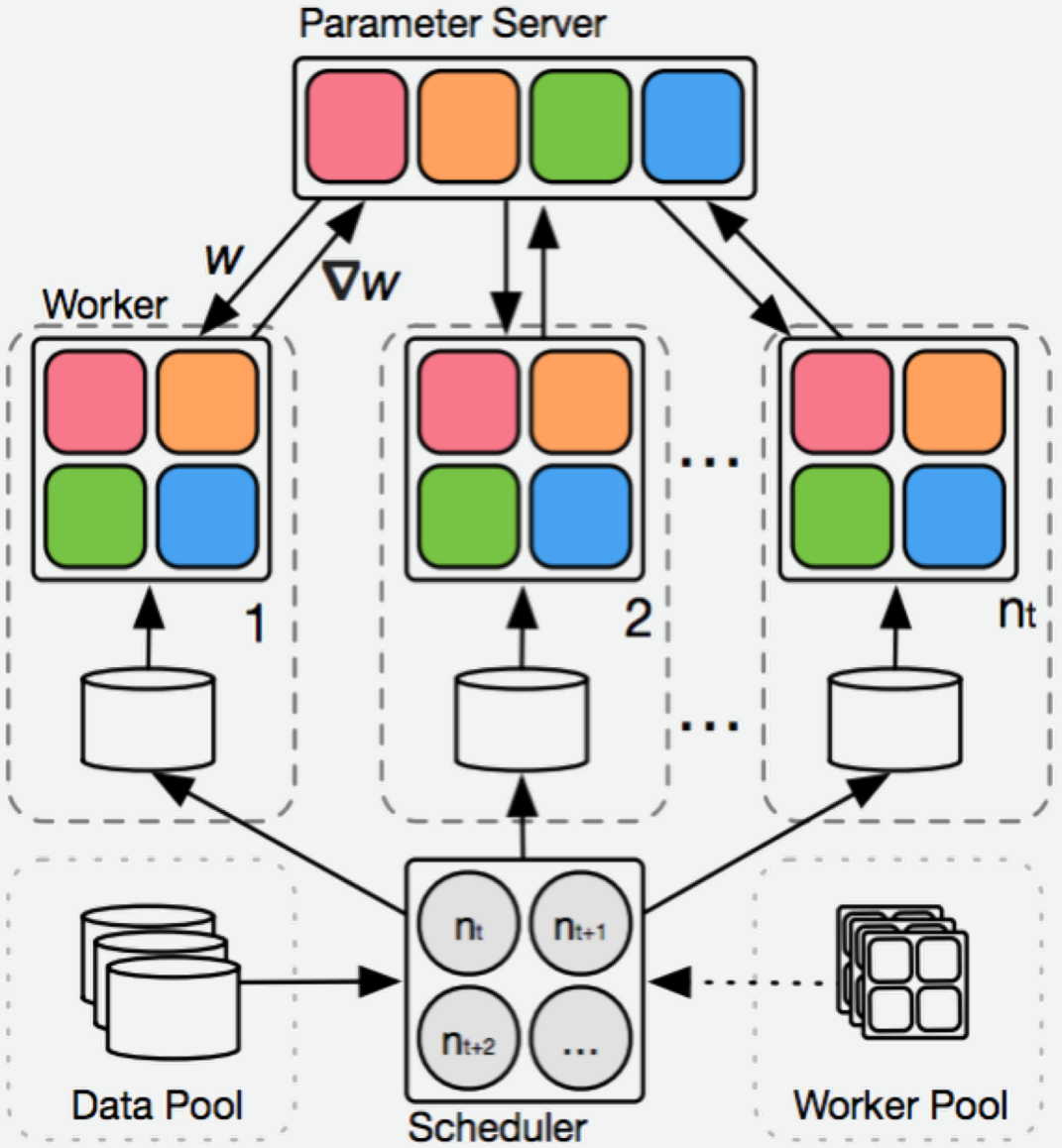
aws

Results: Top-1 Accuracy on ImageNet



Take-home Message:

- Elastic Distributed Training
- Dynamic SGD to Enable Elastic Training:
  - Experiments on Classification, Detection and Segmentation
- Extra Thoughts:
  - Dynamic Scheduling for DL System



AWS System Implementation

- Prototype Available:

https://github.com/awslabs/dynamic-training-with-apache-mxnet-on-aws

- SageMaker Integration in Progress

Thank you!