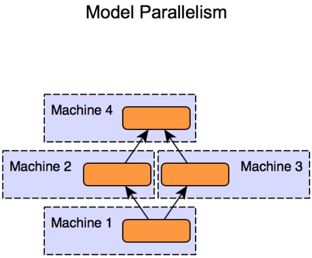
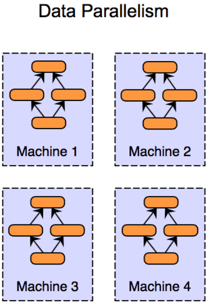
Large-Batch Training for LSTM and Beyond

speaker: **Yang You**

PhD candidate at UC Berkeley, advised by James Demmel

How to use thousands of nodes to train neural networks?





- Model parallelism
  - limited parallelism within a layer: layer is narrow
  - pipeline parallelism across layers: dependency between different layers
- Data parallelism
  - find a local batch size to hit the machine peak performance
  - fix the local batch size when it hits the peak performance
  - increase the global batch size by k times, reduce the # iterations by k times

Success of large-batch training

Table 1: Large-batch training history

Team	Model	Baseline Batch	Large Batch	Baseline Accuracy	Large Batch Accuracy
Google[1]	AlexNet	128	1024	57.7%	56.7%
Berkeley[2]	GoogleNet	32	1024	68.3%	67.8%
Amazon[3]	ResNet-101	256	5120	77.8%	77.8%

[1] Alex Krizhevsky, *One weird trick for parallelizing convolutional neural networks*, 2014 (Google Report)

[2] Iandola et al, *FireCaffe: near-linear acceleration of deep neural network training on compute clusters*, 2015 (CVPR)

[3] Mu Li, *Scaling Distributed Machine Learning with System and Algorithm Co-design*, 2017 (CMU Thesis)

Success of large-batch training

Table 2: Speedup for ImageNet training with ResNet-50.

Batch Size	epochs	Top-1 Accuracy	hardware	time
32 (He et al[4])	90	75.3%	M40 GPU	336h
8K (Goyal et al[5])	90	76.2%	256 P100 GPUs	65m
32K (You et al[6])	90	75.4%	2048 KNLs	20m
32K (Akiba et al[7])	90	74.9%	1024 P100 GPUs	15m
64K (Jia et al[8])	90	76.2%	2048 P40 GPUs	9m

Table 3: ImageNet training with ResNet-50: communication over network.

Batch Size	Operations	# Messages	Data Moved	Comp/Comm
32	10^{18}	3.6 million	374TB	2673
32768	10^{18}	3686	374GB	2.7M

[4] Kaiming He et al, *Imagenet classification with deep convolutional neural networks*, 2012

[5] Priya Goyal et al, *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour*, 2017

[6] Yang You et al, *ImageNet training in minutes*, 2017

[7] Takuya Akiba et al, *Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes*, 2017

[8] Xianyan Jia et al, *Highly Scalable Deep Learning Training System with Mixed-Precision: Training ImageNet in Four Minutes*, 2018

Current problems for large-batch training

- RNN has been widely used, the current large-batch study is focused on CNN applications.
- Even for CNN applications, significant hyper-parameter tuning is required.
- Explanation for large-batch training?

sqrt (square root) scaling for learning rate

- If we increase the batch size by k times, then we increase the learning rate by $\sqrt{k}$ times
  - to keep the variance in the gradient expectation constant[9]

Table 4: Example of sqrt scaling.

Batch Size	initial learning rate
32	η
64	$\eta \times 2^{0.5}$
128	$\eta \times 2^{1.0}$
256	$\eta \times 2^{1.5}$
512	$\eta \times 2^{2.0}$
1024	$\eta \times 2^{2.5}$
...	...

[9] Alex Krizhevsky, *One weird trick for parallelizing convolutional neural networks*, 2014

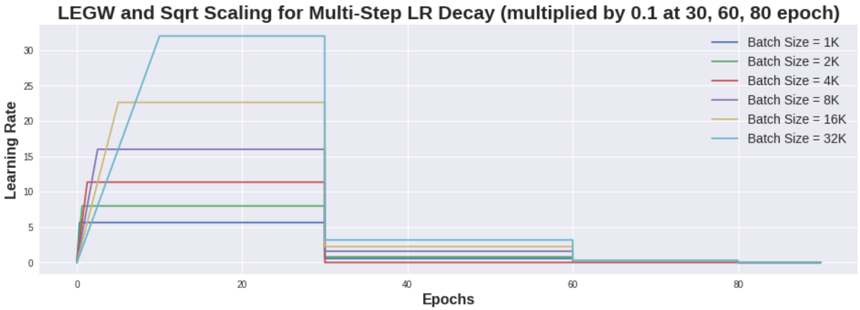
Linear-Epoch Gradual Warmup (LEGW or Leg-Warmup)

- Usually, the learning rate (LR) for large batch is large, which makes the algorithm diverge at the beginning
- **Gradual Warmup Scheme**: we set the initial LR to a small value and increase it gradually to the target in a few epochs (e.g. 5 or 10).
- **LEGW**: If we increase the batch size by k times, then we increase the warmup epochs by k times (works with sqrt LR scaling)

Table 5: Example of LEGW.

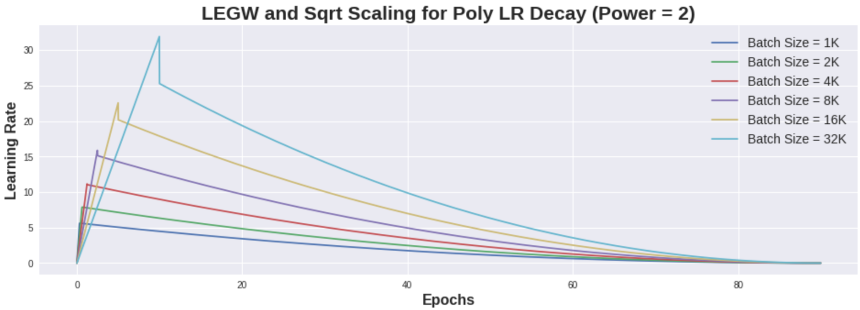
Batch Size	num of warmup epochs
32	w
64	$w \times 2^1$
128	$w \times 2^2$
256	$w \times 2^3$
512	$w \times 2^4$
1024	$w \times 2^5$
...	...

Example for multi-step LR decay



- The baseline uses a batch size of 1K and a peak LR of $2^{2.5}$
    - In the initial 0.3125 epochs, it gradually increases LR from 0 to $2^{2.5}$
- The peak learning rate for batch size B: $\sqrt{(B/1K)} \times 2^{2.5}$
- The warmup epochs for batch size B: $(B/1K) \times 0.3125$

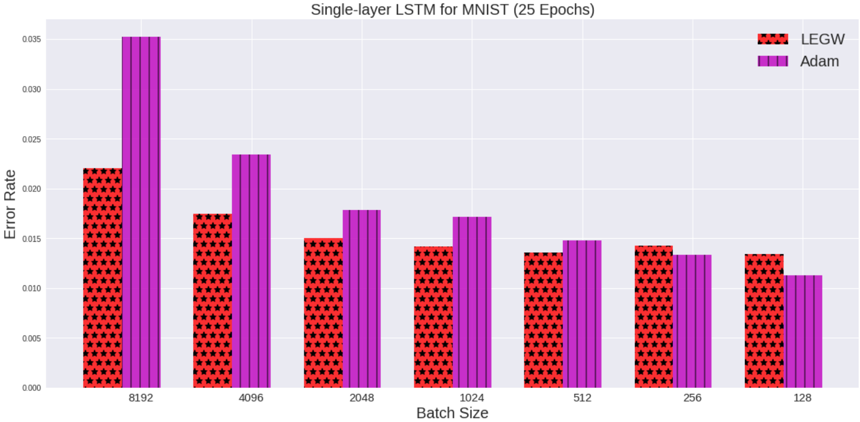
Example for polynomial LR decay



- The baseline uses a batch size of 1K and a peak LR of $2^{2.5}$
    - In the initial 0.3125 epochs, it gradually increases LR from 0 to $2^{2.5}$
- After warmup, the LR of iteration i is $\eta \times (1 - i/I)^p$
- p is the power of polynomial decay (e.g. $p = 2.0$)
- I is the total number of iterations

Results for MNIST with LSTM (compared to Adam)

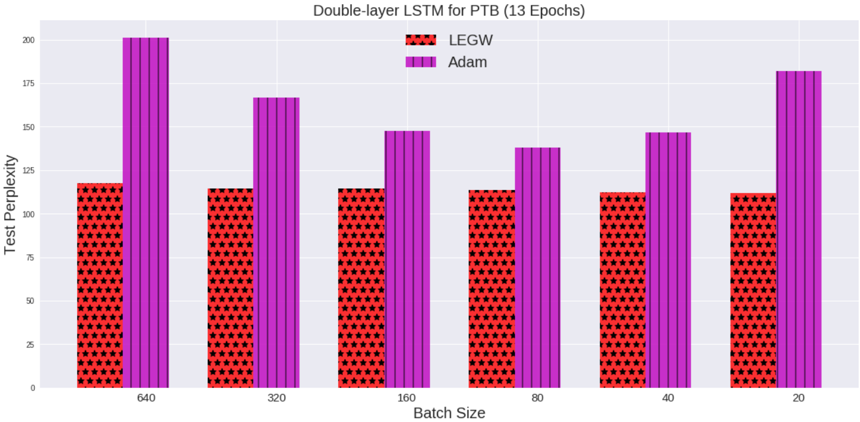
- Adam is the best existing adaptive solver in our experiments



- In this figure, lower is better

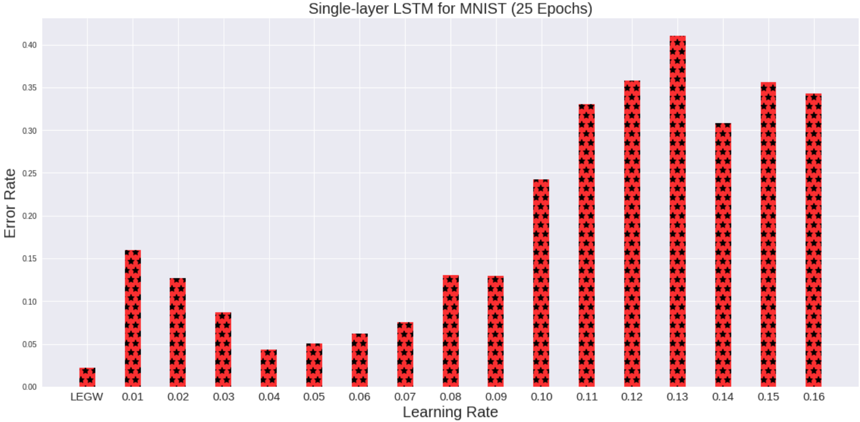
Results for PTB with LSTM (compared to Adam)

- Adam is the best existing adaptive solver in our experiments



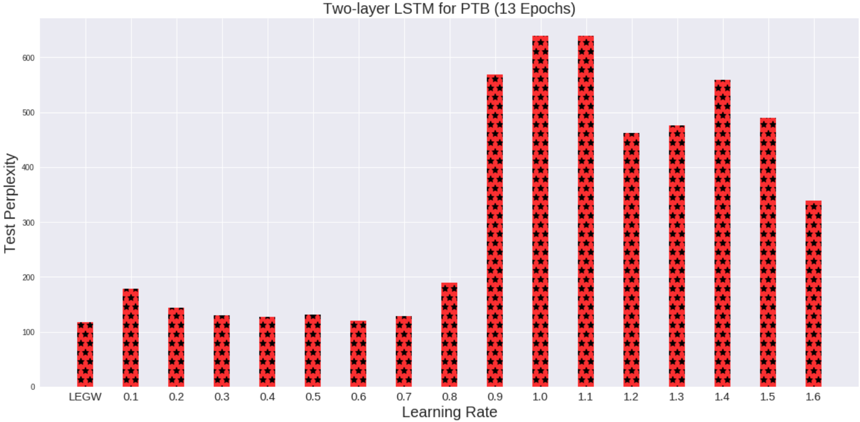
- In this figure, lower is better

Results for MNIST with LSTM (compared to tuning)



- In this figure, lower is better
- Horizontal axis is the most effective tuning region
- They run the same number of epochs for batch size = 8K

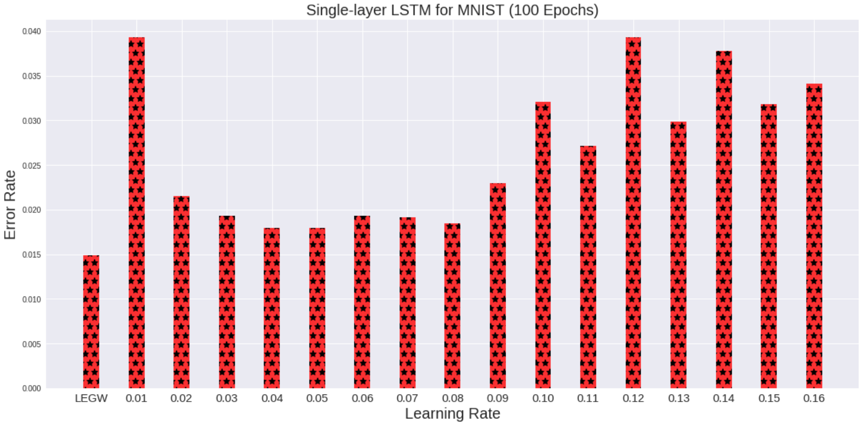
Results for MNIST with PTB (compared to tuning)



- In this figure, lower is better
- Horizontal axis is the most effective tuning region
- They run the same number of epochs for batch size = 640

Results for MNIST with LSTM (compared to tuning)

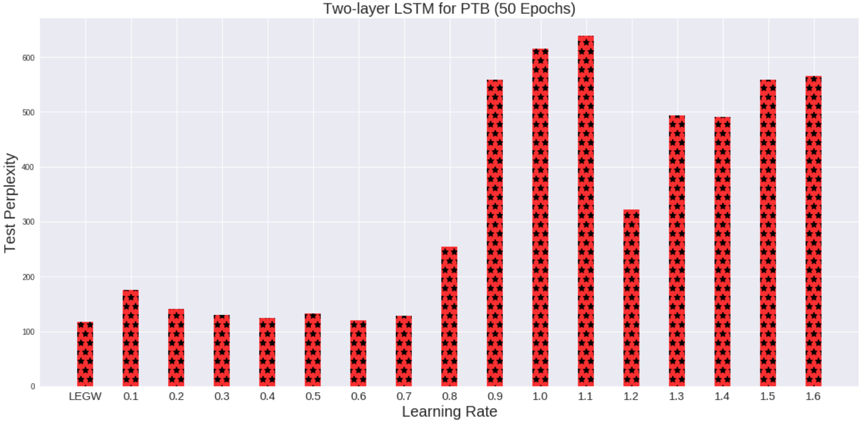
- Running long enough: from 25 epochs to 100 epochs



- In this figure, lower is better
- Horizontal axis is the most effective tuning region
- They run the same number of epochs for batch size = 8K

Results for PTB with LSTM (compared to tuning)

- Running long enough: from 13 epochs to 50 epochs



- In this figure, lower is better
- Horizontal axis is the most effective tuning region
- They run the same number of epochs for batch size = 8K

Results for ImageNet training (LEGW without tuning)

Table 6: With LEGW (on top of LARS solver), we can scale the batch size to 32K for ImageNet training without tuning hype-parameters.

Batch Size	Init LR	LR scheme	Warmup	Epochs	Top-5 Accuracy
32768	$2^{5.0}$	poly power $=2$	10 epochs	90	93.18%
16384	$2^{4.5}$	poly power $=2$	5 epochs	90	93.43%
8192	$2^{4.0}$	poly power $=2$	2.5 epochs	90	93.55%
4096	$2^{3.5}$	poly power $=2$	1.25 epochs	90	93.34%
2048	$2^{3.0}$	poly power $=2$	0.625 epochs	90	93.25%
1024	$2^{2.5}$	poly power $=2$	0.3125 epochs	90	93.36%

State-of-the-art performance without tuning

Table 7: Speedup for ResNet-50.

Batch Size	epochs	Top-1 Accuracy	hardware	time
32 (He et al, 2012)	90	75.3%	M40 GPU	336h
8K (Goyal et al, 2017)	90	76.2%	256 P100 GPUs	65m
32K (You et al, 2017)	90	75.4%	2048 KNLs	20m
32K (Akiba et al, 2017)	90	74.9%	1024 P100 GPUs	15m
64K (Jia et al, 2018)	90	76.2%	2048 P40 GPUs	8.7m
32K (**this work**)	90	**76.4%**	TPU-v2 Pod	**7m**

- Followup: Mikami et al (Sony report, Nov 13, 2018)

Explanation and more results, check our tech report!

- https://www.cs.berkeley.edu/∼youyang/batch-lstm.pdf