Train longer, generalize better: closing the generalization gap in large batch training of neural networks

Elad Hoffer*, Itay Hubara*, Daniel Soudry

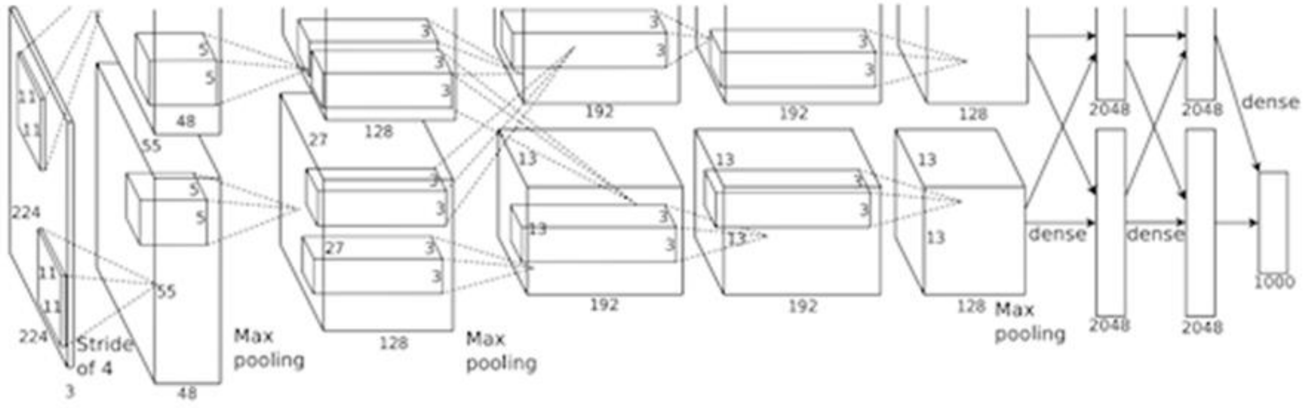
*Equal contribution

TECHNION
Israel Institute
of Technology

Better models - parallelization is crucial

- Model parallelism:

Split model (same data)



AlexNet [Krizhevsky et al. 2012]: model split on two GPUs

- Data parallelism:

Split data (same model)

$$\Delta \mathbf{w} \propto -\frac{1}{b}\sum_{n=1}^{b} \nabla_{\mathbf{w}} L_n(\mathbf{w})$$

SGD: weight update proportional to gradients averaged over mini batch

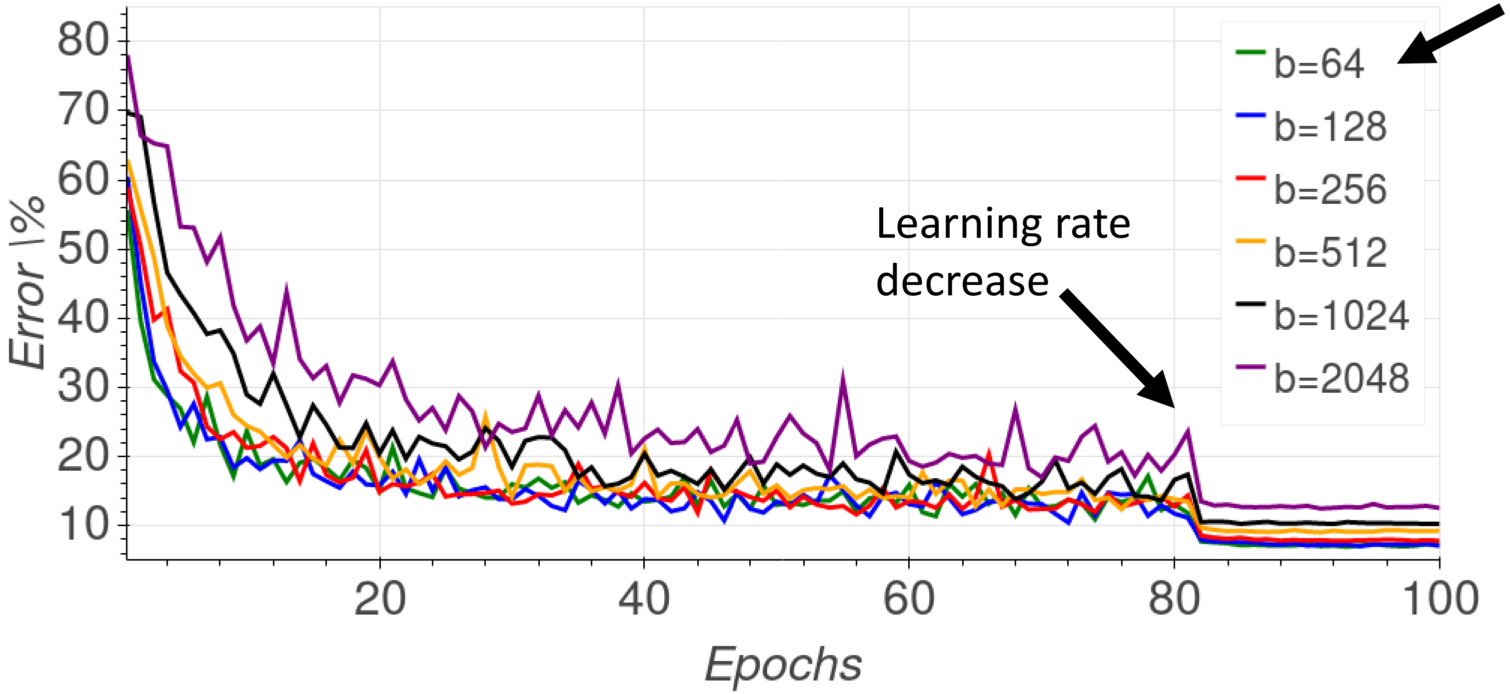
Can we increase batch size and improve parallelization?

Large batch size hurts generalization?

Dataset: CIFAR10, **Architecture:** Resnet44, **Training**: SGD + momentum (+ gradient clipping)

b=batch size

Why?

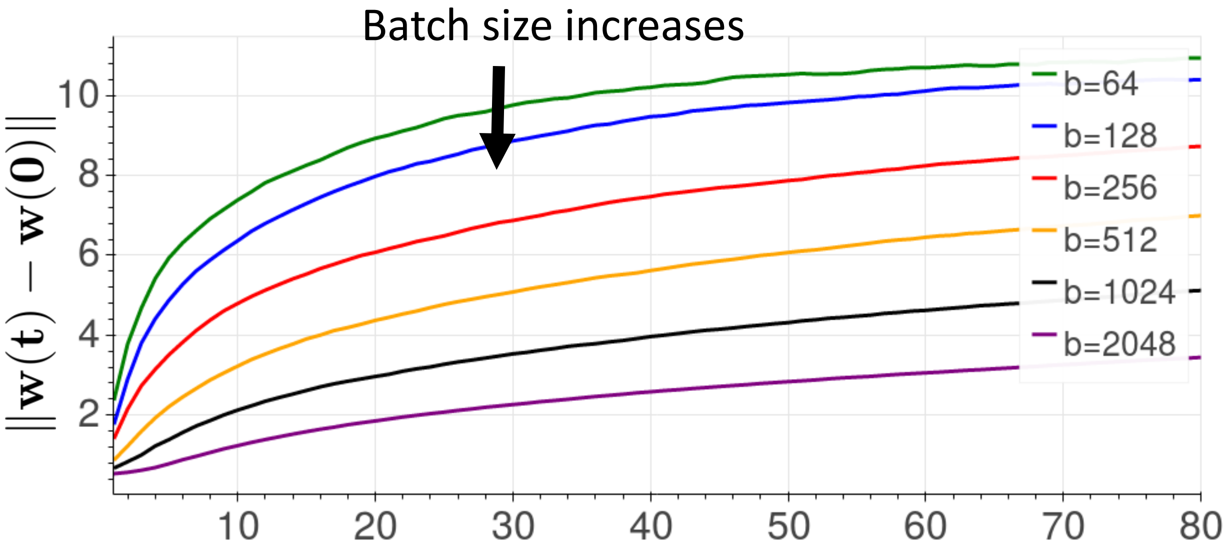


- Generalization gap persisted in models trained “without any budget or limits, until the loss function ceased to improve" [Keskar et al. 2017]

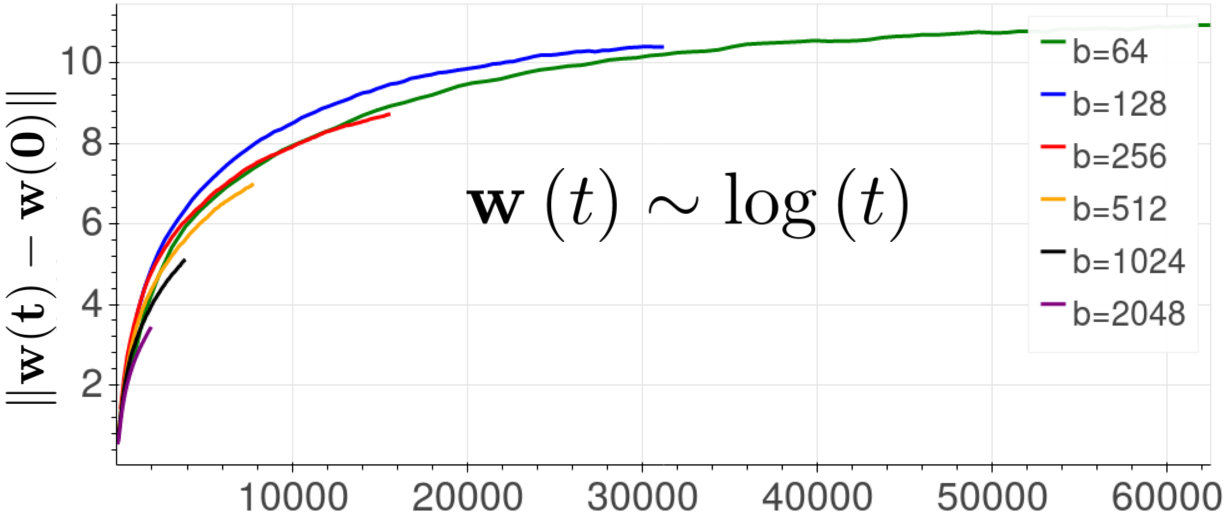
Observation

Weight distances from initialization increase

logarithmically **with iterations**



Epochs



Iterations (gradient updates)

Why logarithmic behavior? Theory later…

Experimental details

- We experiment with various datasets and models
- Optimizing using SGD + momentum + gradient clipping
  - Usually generalize better than adaptive methods (e.g Adam)
  - Grad clipping effectively creates a “warm-up” phase
- Noticeable generalization gap between small and large batch

Network	Dataset	SB	LB
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%

Closing the generalization gap (2/4)

- Adapt learning rate. In CIFAR $\propto \sqrt{b}$
  - Idea: mimic small batch gradient statistics (dataset dependent)

- Noticeably improves generalization, the gap remains

Network	Dataset	SB	LB	+LR
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%	97.55%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%	86.15%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%	89.30%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%	88.6%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%	57.38%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%	69.05%

Closing the generalization gap (3/4)

- Ghost batch norm
  - Idea again: mimic small batch size statistics
  - Also: reduces communication bandwidth
  - Further improves generalization without incurring overhead

$$\hat{x} = \frac{x - \mu_x}{\sigma_x}$$

$$\widehat{x_1} = \frac{x_1 - \mu_{x_1}}{\sigma_{x_1}}$$

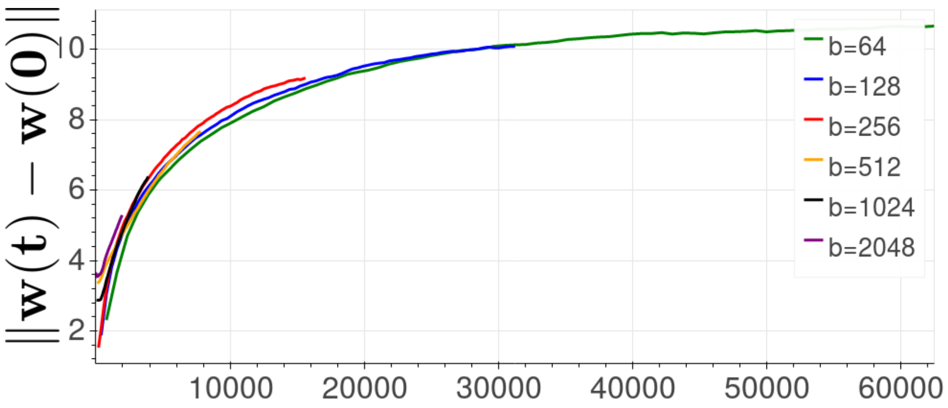
$$\vdots$$

$$\widehat{x_k} = \frac{x_k - \mu_{x_k}}{\sigma_{x_k}}$$

Network	Dataset	SB	LB	+LR	+GBN
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%	97.55%	97.60%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%	86.15%	86.4%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%	89.30%	90.50%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%	88.6%	91.50%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%	57.38%	57.5%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%	69.05%	71.20%

Graph indicates: not enough iterations?

- Using these modifications – distance from initialization now better matched
- However, graph indicates: insufficient iterations with large batch

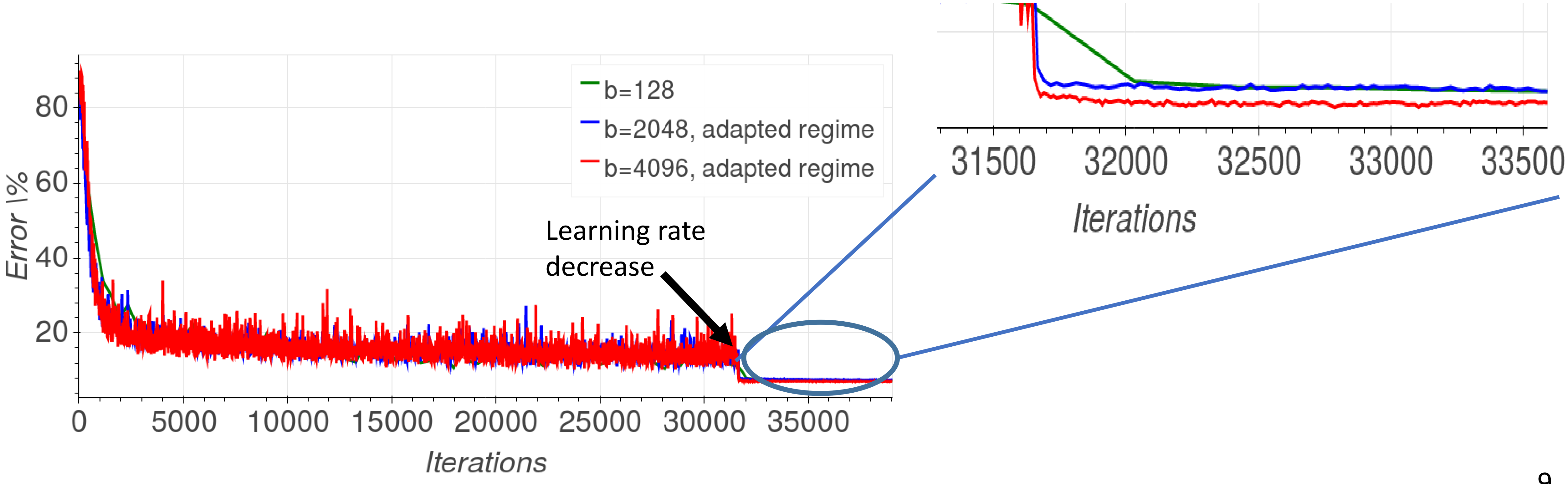


Iterations

Network	Dataset	SB	LB	+LR	+GBN
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%	97.55%	97.60%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%	86.15%	86.4%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%	89.30%	90.50%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%	88.6%	91.50%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%	57.38%	57.5%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%	69.05%	71.20%

Train longer, generalize better

- With sufficient iterations in “plateau” region, generalization gap vanish:



Closing the generalization gap (4/4)

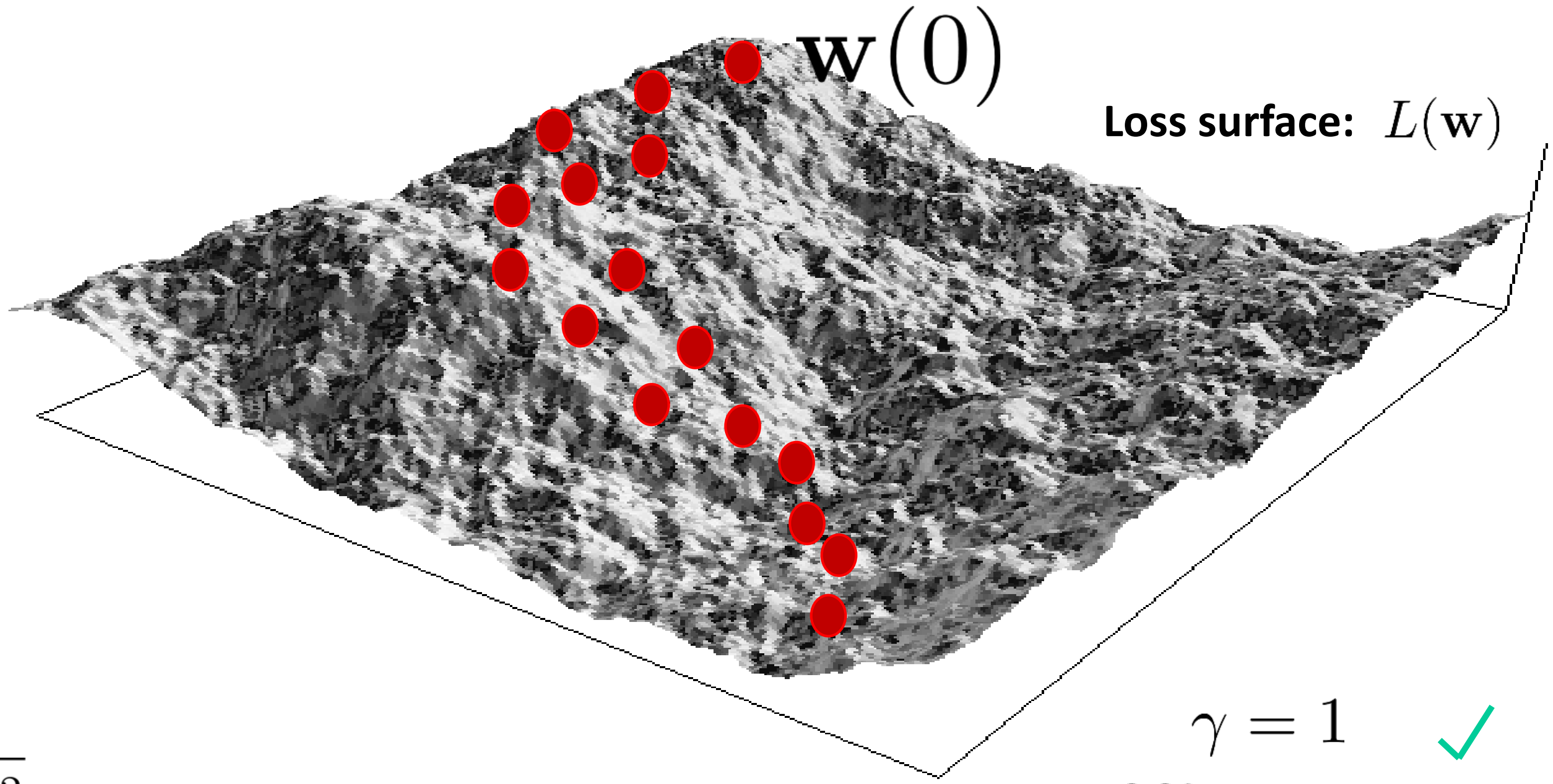
- Regime Adaptation – train so that the number of iterations is fixed for all batch sizes (train longer number of epochs)
  - Completely closes the generalization gap

Network	Dataset	SB	LB	+LR	+GBN	+RA
F1 (Keskar et al., 2017)	MNIST	98.27%	97.05%	97.55%	97.60%	98.53%
C1 (Keskar et al., 2017)	Cifar10	87.80%	83.95%	86.15%	86.4%	88.20%
Resnet44 (He et al., 2016)	Cifar10	92.83%	86.10%	89.30%	90.50%	93.07%
VGG (Simonyan, 2014)	Cifar10	92.30%	84.1%	88.6%	91.50%	93.03%
C3 (Keskar et al., 2017)	Cifar100	61.25%	51.50%	57.38%	57.5%	63.20%
WResnet16-4 (Zagoruyko, 2016)	Cifar100	73.70%	68.15%	69.05%	71.20%	73.57%

ImageNet (AlexNet):

LB size	Dataset	SB	LB[8]	+LR[8]	+GBN	+RA
4096	ImageNet	57.10%	41.23%	53.25%	54.92%	59.5%
8192	ImageNet	57.10%	41.23%	53.25%	53.93%	59.5%

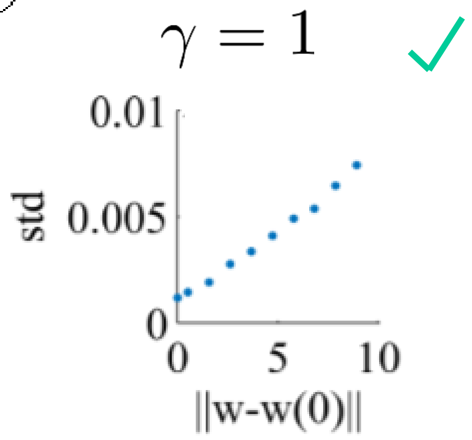
Why weight distances increase logarithmically?



Hypothesis:

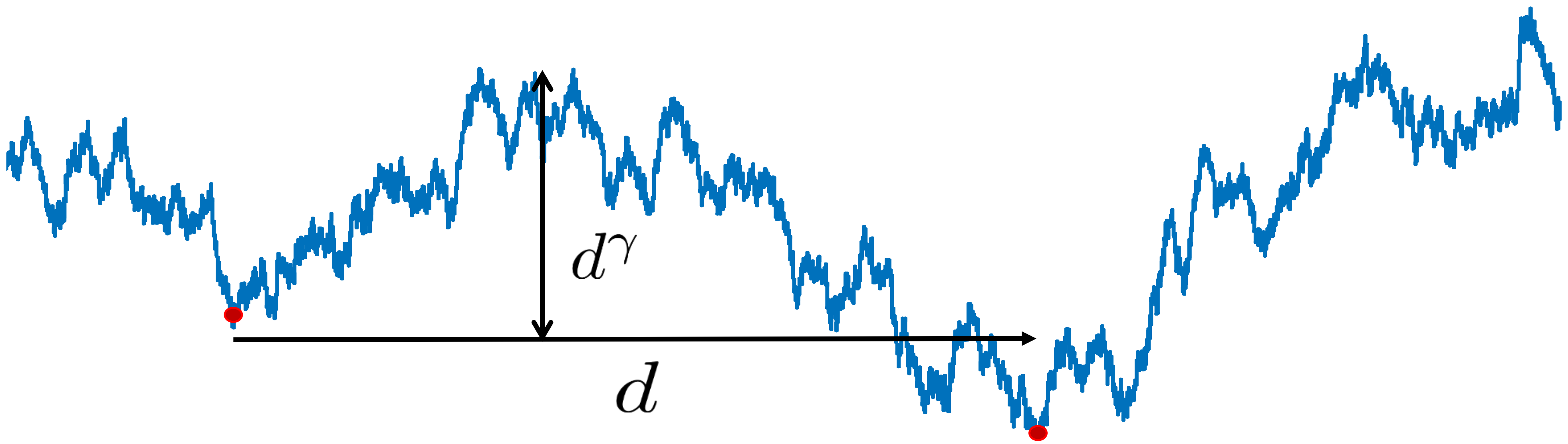
During initial high learning rate phase: "random walk on a random potential" where

$$\mathrm{std} \triangleq \sqrt{\mathbb{E}\left(L\left(\mathbf{w}\right)-L\left(\mathbf{w}(0)\right)\right)^2} \sim \left\|\mathbf{w}-\mathbf{w}(0)\right\|^{\gamma}$$



Marinari et al., 1983: $\mathbf{w}\left(t\right) \sim \log^{\frac{1}{\gamma}}\left(t\right)$ **"ultra-slow diffusion"**

Ultra-slow diffusion: Basic idea



Time to pass tallest barrier: $t \propto \exp(d^\gamma)$ $\Rightarrow d \propto \log^{\frac{1}{\gamma}}(t)$

Summary so far

- **Q:** Is there inherent generalization problem with large batches?
  A: Observed: no, just adjust training regime.

- **Q:** What is the mechanism behind training dynamics?
  A: Hypothesis: "random walk on a random potential”

- **Q:** Can we reduce the total wall clock time?
  A: Yes, in some models

Significant speed-ups possible

Accurate, Large Minibatch SGD: Training ImageNet in <u>1 Hour</u>

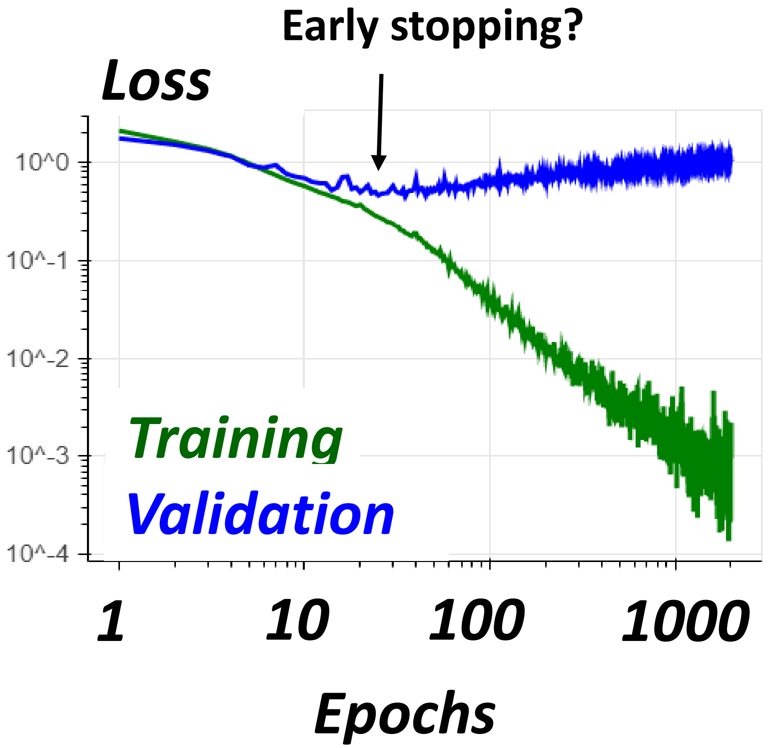
Goyal et al. (Facebook whitepaper, two weeks after us)

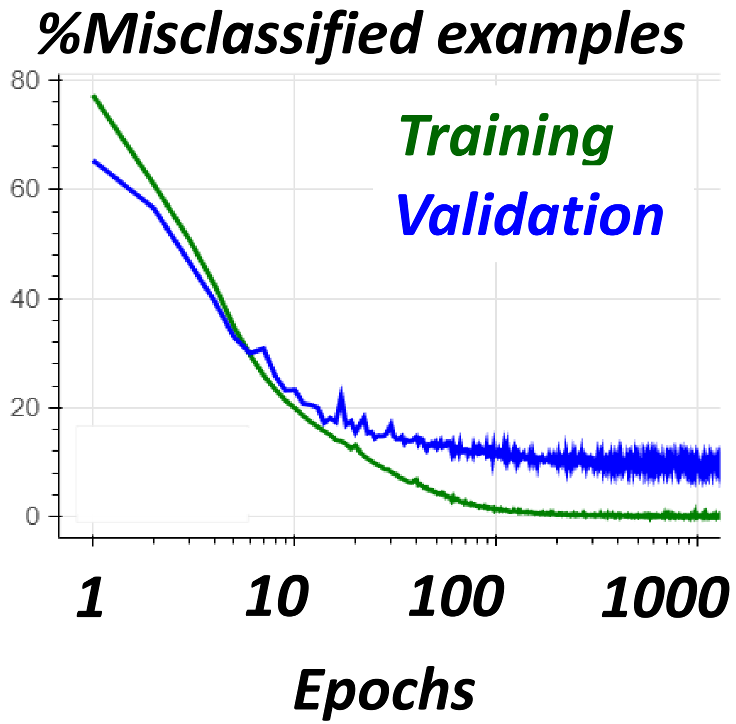
- Large scale experiments: ResNet over ImageNet, 256 GPUs
- Similar methods, except learning rate
- X29 times faster than a single worker

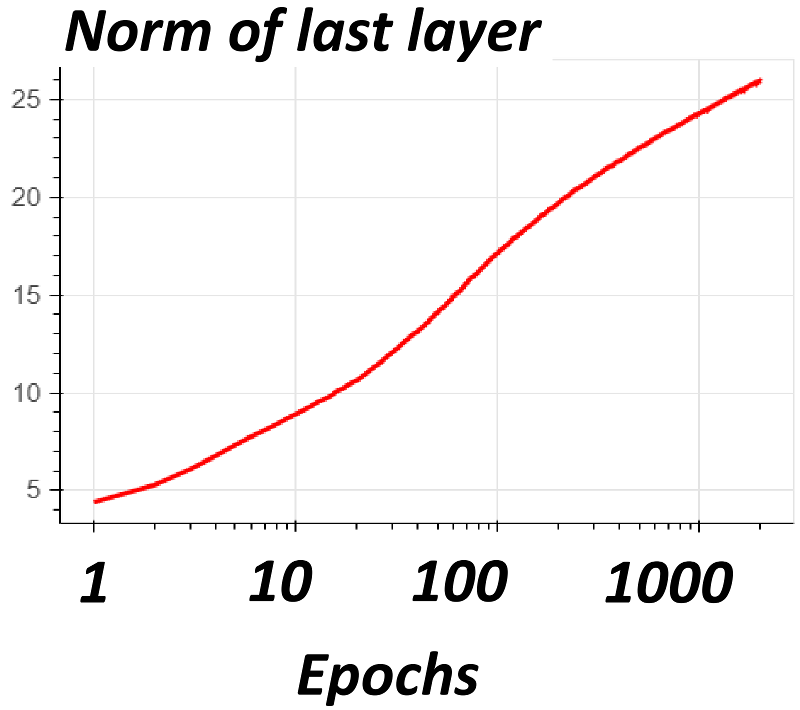
- More followed:
  - **Large Batch Training of Convolutional Networks** (You et al.)
  - **ImageNet Training in Minutes** (You et al.)
  - **Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes** (Akiba et al.)

Why “Overfitting” is good for generalization?

- In contrast to common practice: good generalization results from many gradient updates in an “overfitting regime”







Why “Overfitting” is good for generalization?

- Can be shown to happen for logistic regression on separable data!
- Explanation there (proved):

Slow convergence to max-margin solution

The Implicit Bias of Gradient Descent on Separable Data (Arxiv 2017)

- *Daniel Soudry, Elad Hoffer, Nati Srebro*

Thank you for your time! Questions?

Code: https://github.com/eladhoffer/bigBatch