Large Scale Training of Neural Networks

Zhewei Yao, Amir Gholami

& Kurt Keutzer, Michael Mahoney

University of California, Berkeley

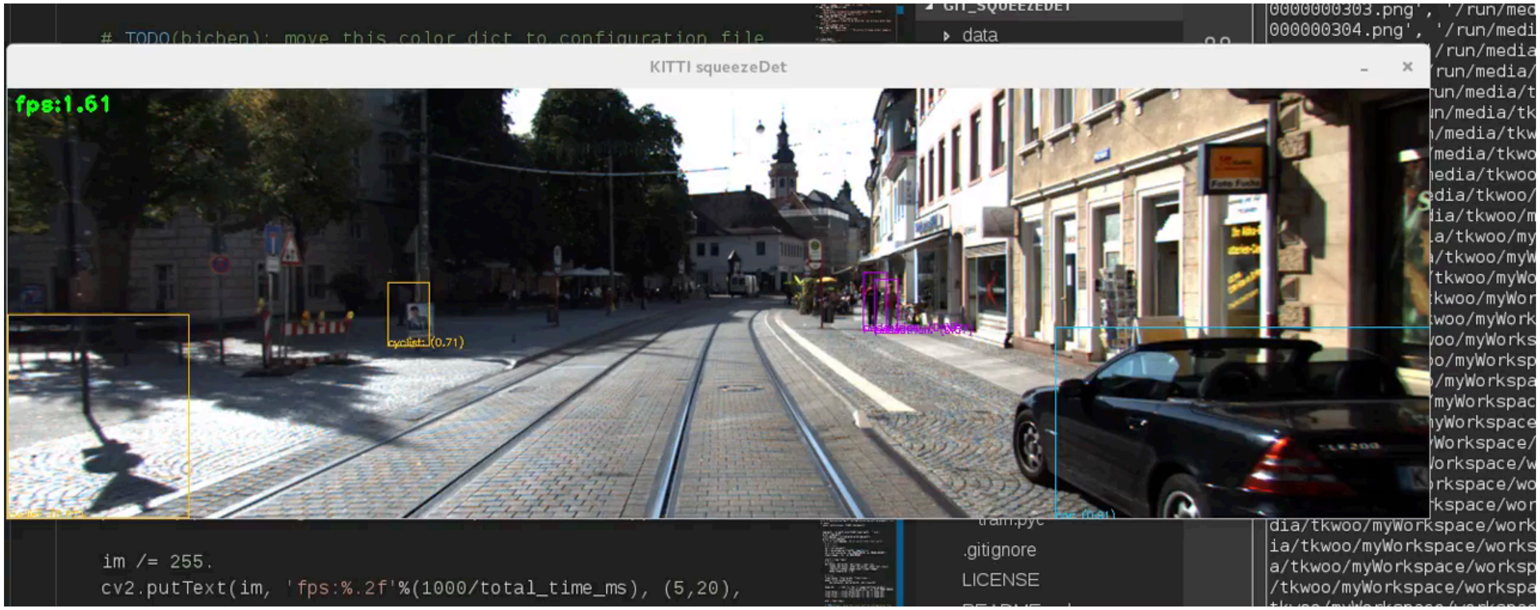
November 2018



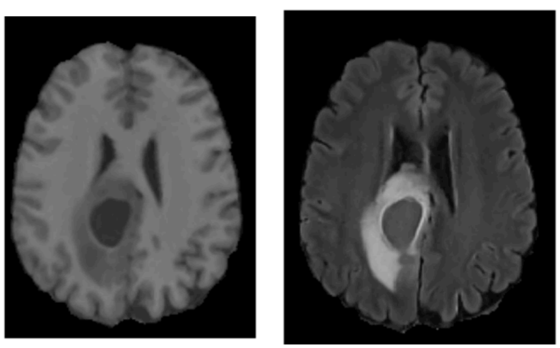

Berkeley
UNIVERSITY OF CALIFORNIA

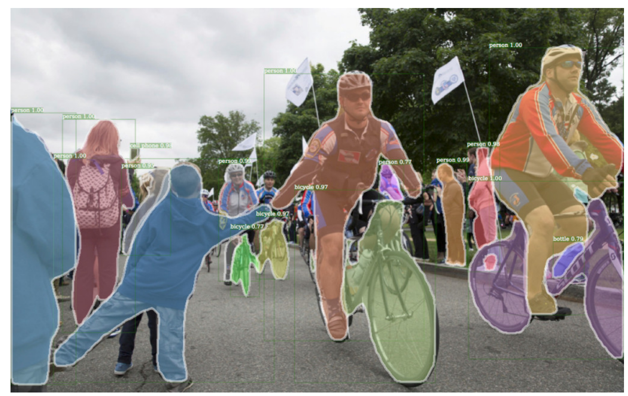
BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

DNN's Impact on a wide range of problems



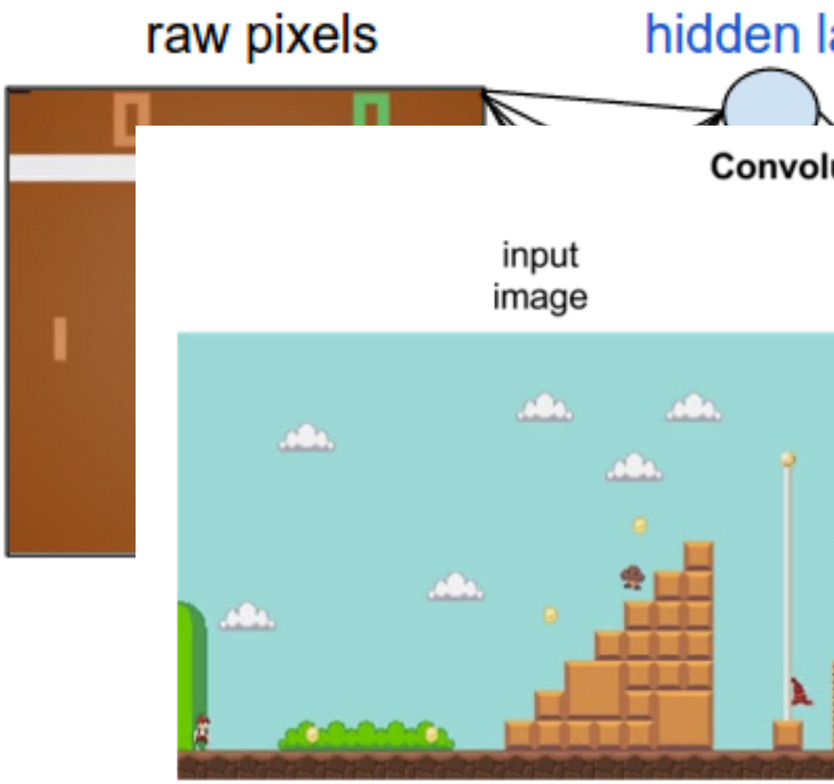
T1. FLAIR.





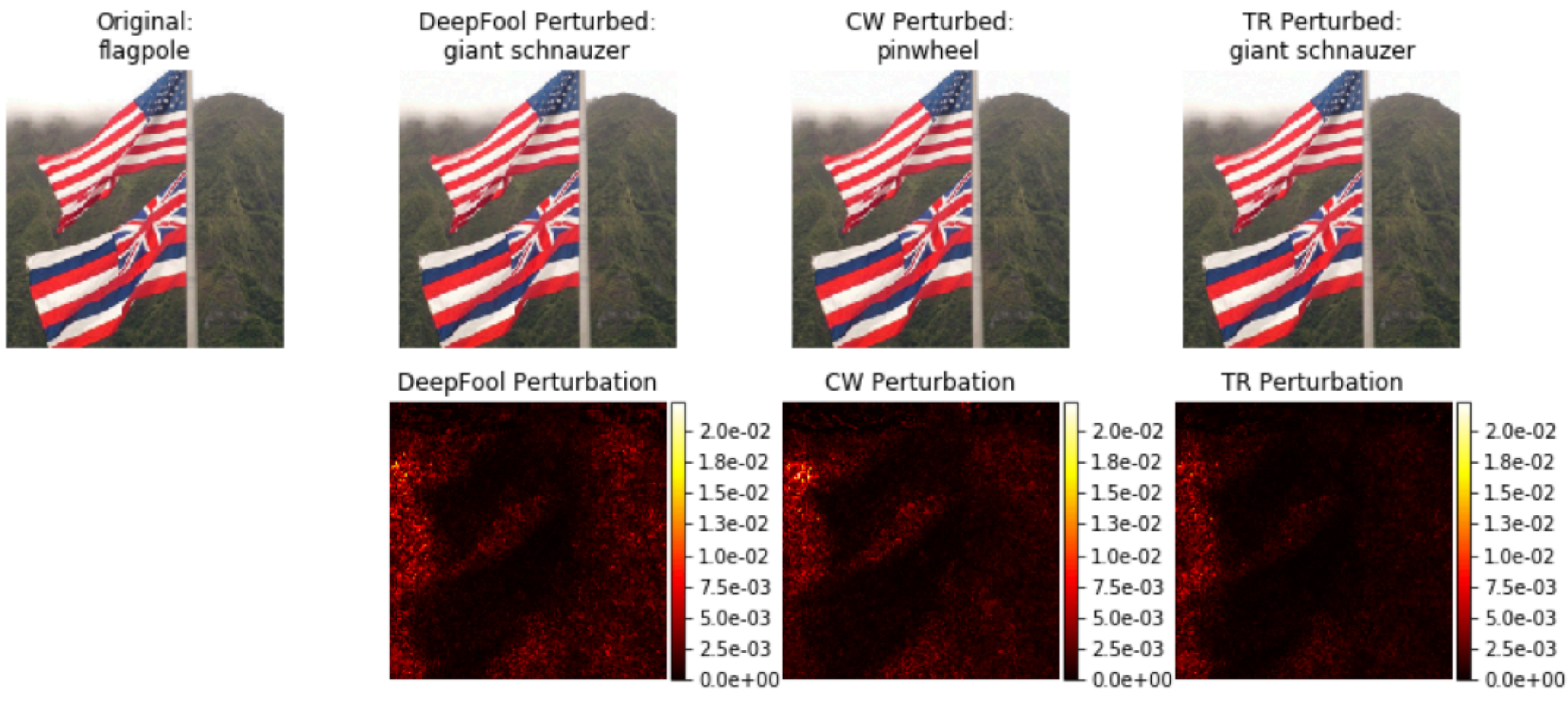
B. Wu, F. Iandola, P. Jin, and K. Keutzer. "SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving."CVPR Workshop
A. Gholami, S. Subramanian, V. Shenoy, N. Himthani, X. Yue, S. Zhao, P. Jin, K. Keutzer, G. Biros, A novel domain adaptation framework for medical image segmentation, BRATS, MICCAI 2018
A Semantic Segmentation using Detectron, Facebook Research

DNN’s Impact on a wide range of problems



Slide from Bo Li

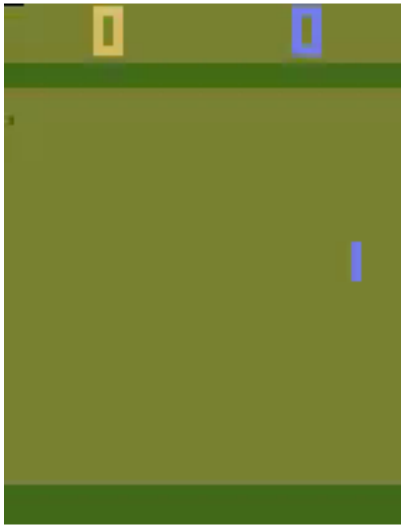
Susceptibility to Adversarial Example



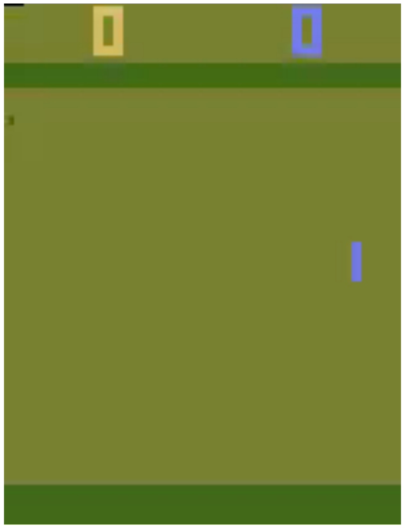
Z Yao, A Gholami, M Mahoney, K Keutzer Trust Region Based Adversarial Attack on Neural Networks

Susceptibility to Adversarial Example

- Despite noticeable impact, NN can be easily fooled



Original Frames



Adversarial perturbation injected into **every other 10 frames**

Song et al.: Delving into adversarial attacks on deep policies. ICLR Workshop 2017

[Chaowei Xiao, Bo Li, Jun-yan Zhu, Warren He, Mingyan Liu, Dawn Song, 2017]

Our Research

Rapid **Training** of DNNs

Collect/annotate adequate **training data**

Integrated Parallelism [2]
2nd Order Method[3,7,8,9]

Find the right **DNN architectures** using design space exploration

Domain Adaptation [1]

SqueezeNext[4]
ShiftNet[6]
112x smaller models

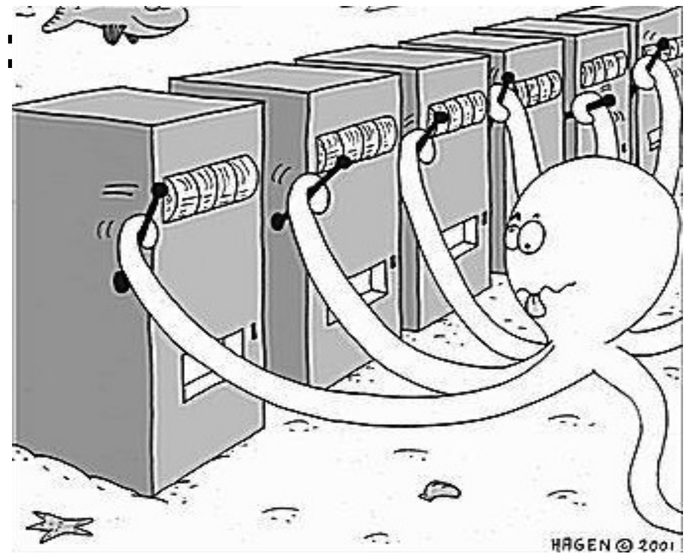
Hardware and model Co-design [5]

Create **efficient implementation** for embedded hardware

[1] A.Gholami, et al. A Novel **Domain Adaptation** Framework for Medical Image Segmentation, BraTS'18, MICCAI (2018)
[2] A. Gholami, A. Azad, P. Jin, K. Keutzer, A. Buluc. **Integrated** Model, Batch and Domain Parallelism in Training Neural Networks, ACM Symposium on Parallelism in Algorithms and Architectures(SPAA'18)
[3] Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)
[4] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, K. Keutzer. **SqueezeNext**: Hardware-Aware Neural Network Design, ECV Workshop at CVPR'18
[5] K. Kwon, A. Amid, A. Gholami, B. Wu, K. Keutzer **Co-Design** of Deep Neural Nets and Neural Net Accelerators for Embedded Vision Applications, Design Automation Conference (DAC) 2018
[6] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao, N. Golmant, A. Gholami, J. Gonzalez, K. Keutzer. **Shift:** A zero flop, zero parameter alternative to spatial convolutions, CVPR 2018
[7]Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and **Second-Order Information**, (under review)
[8] Z. Yao, A. Gholami, P. Xu, K. Keutzer, M. Mahnoey, Trust Region based Adversarial Attack on Neural Networks, (in preparation)
[9] Z Yao, N Mu, K Keutzer, MW Mahoney**. Weight** Re-Initialization through Cyclical Batch Scheduling

Many many knobs!

- SGD is very sensitive to hyper-parameters and in particular **batch size**
- **Batch size inter dependent with:**
  - **Degradation in accuracy**
  - **Poor generalizability**
  - **Robustness of model**
  - **Training time**
  - **Parallel Scalability**



$$W^{t+1} = W^t - \alpha \sum_{i=0}^{B} \nabla_W f_i(W^t, x)$$

High Level Outline

- DNN design requires training on large datasets
  - Time consuming
  - Need fast training -> parallelization -> large batch
- Large batch training does not work:
  - **Degrades accuracy**
  - **Poor robustness** to adversarial inputs
  - Existing solutions either do not work or require **extensive hyper-parameter tuning**

N Golmant, N Vemuri, Z Yao, V Feinberg, A Gholami, K Rothauge, MW Mahoney, J Gonzalez. On the Computational Inefficiency of Large Batch Sizes for Stochastic Gradient Descent.

Summary of Contributions

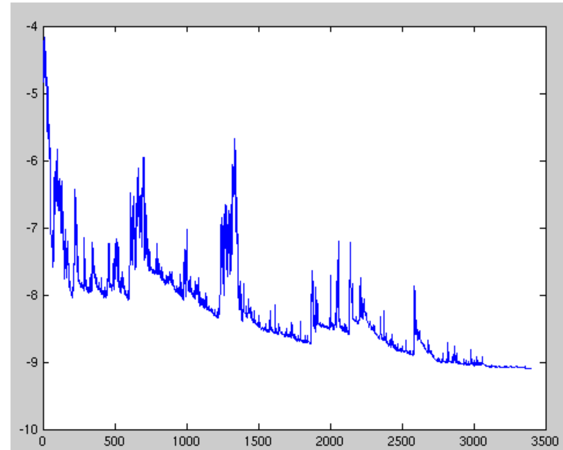
- Extensive analysis of mini-batch SGD behavior for deep neural networks
  - Saddle points, adversarial robustness, sharp/flat minima
- A new **Hessian based** large batch size training
  - ~~Degrades accuracy~~
    - **Equal or better accuracy**
  - ~~Poor robustness to adversarial inputs~~
    - **More robust model**
  - ~~Existing solutions either do not work or require extensive hyper-parameter tuning~~
    - **No hyper-parameter tuning**
- Extensive testing of the proposed method on multiple datasets and multiple neural networks
  - Cifar-10/100, **ImageNet**, SVHN, Tiny ImageNet

Stochastic Gradient Descent (SGD)

$$\text{Assume } f(W^t, x) = \frac{1}{n}\sum_{i=1}^{n} f_i(W^t, x)$$

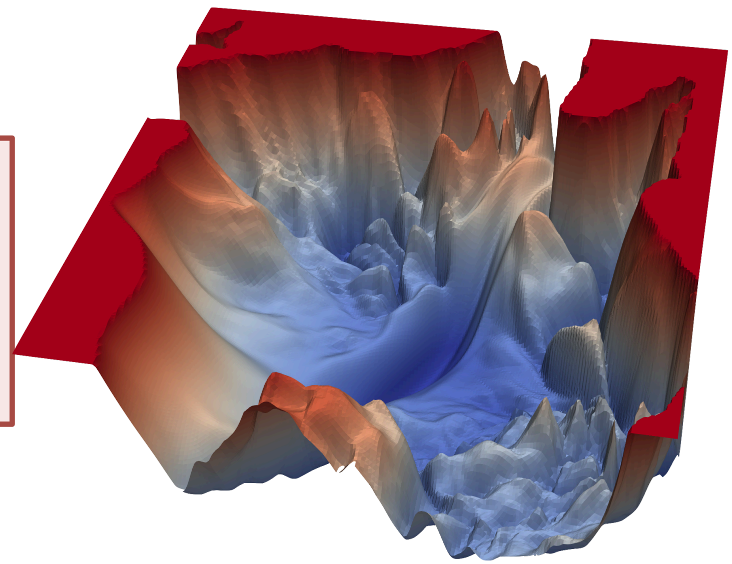
$$W^{t+1} \leftarrow W^t - \alpha \cdot \nabla_W f_i(W^t, x)$$

Pure SGD: compute gradient using 1 sample



$$W^{t+1} \leftarrow W^t - \alpha \cdot \frac{1}{b}\sum_{i=k+1}^{k+b} \nabla_W f_i(W^t, x)$$

Mini-batch: compute gradient using b samples

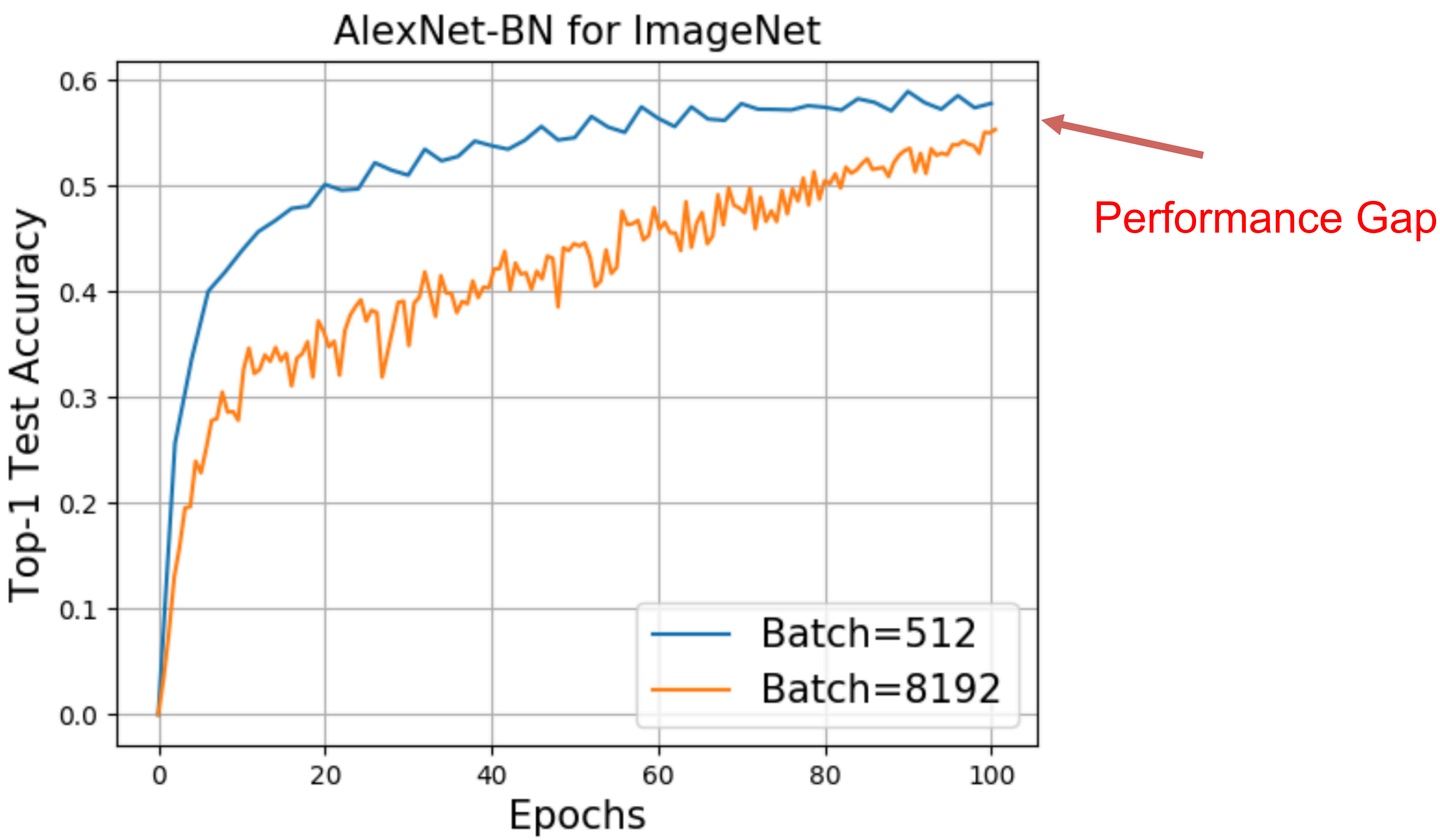


- Actually the name is a misnomer, *this is not a "descent" method*

Image from https://www.cs.umd.edu/~tomg/projects/landscapes/

Degradation in Accuracy

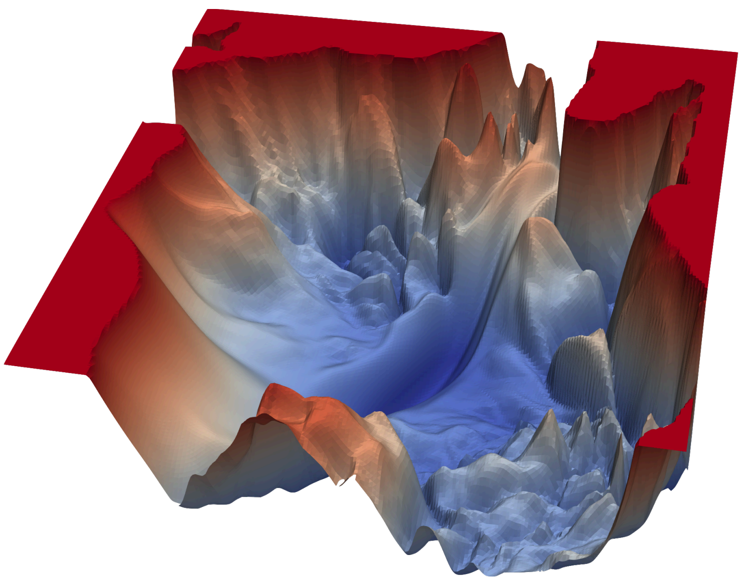
- Larger Batch often leads to degradation in accuracy

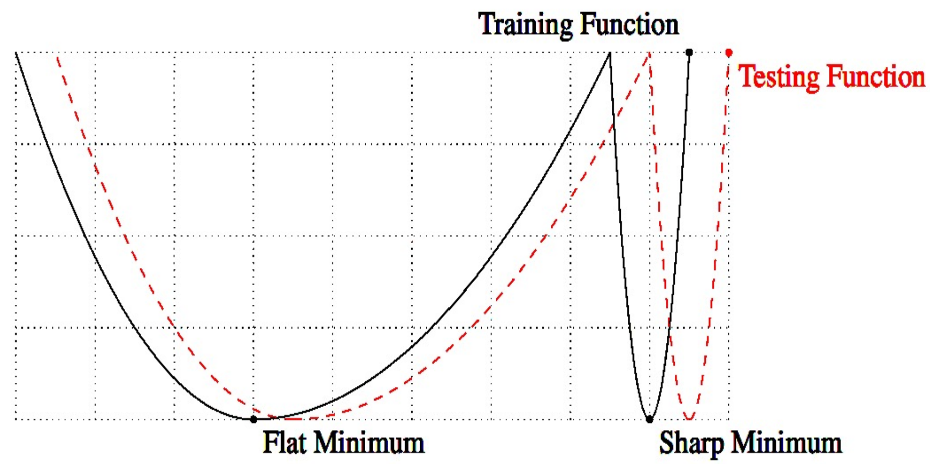


Ginsburg, Boris, Igor Gitman, and Yang You. "Large Batch Training of Convolutional Networks with Layer-wise Adaptive Rate Scaling." arxiv:1708.03888.

Poor Generalization

- Why large batch suffers from poor generalization performance?
  - A common belief is that large batch training gets attracted to “sharp minimas”
  - Another theory is that large batch may get stuck in saddle points





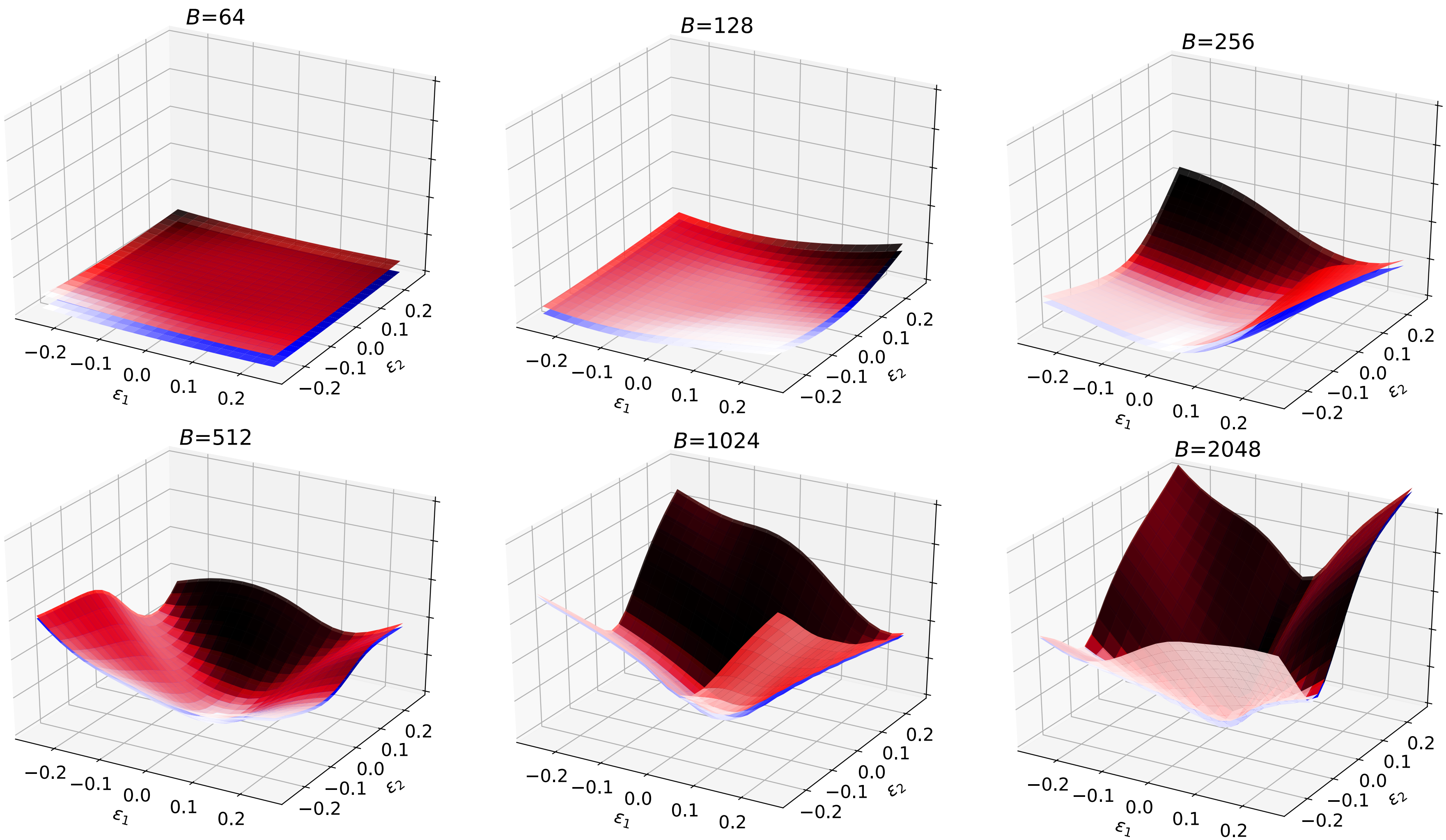
Loss landscape from https://www.cs.umd.edu/~tomg/projects/landscapes/
Keskar, Nitish Shirish, et al. "On large-batch training for deep learning: Generalization gap and sharp minima." ICLR’16 (arXiv:1609.04836)

Analysis through Hessian Lens!

- **We back-propogate the Hessian operator (second derivative) and compute its spectrum during training along with total gradient**
- **The Hessian spectrum is computed on all the training/testing examples using power method**
- **We visualize the landscape of loss along dominant eigenvectors of the Hessian**

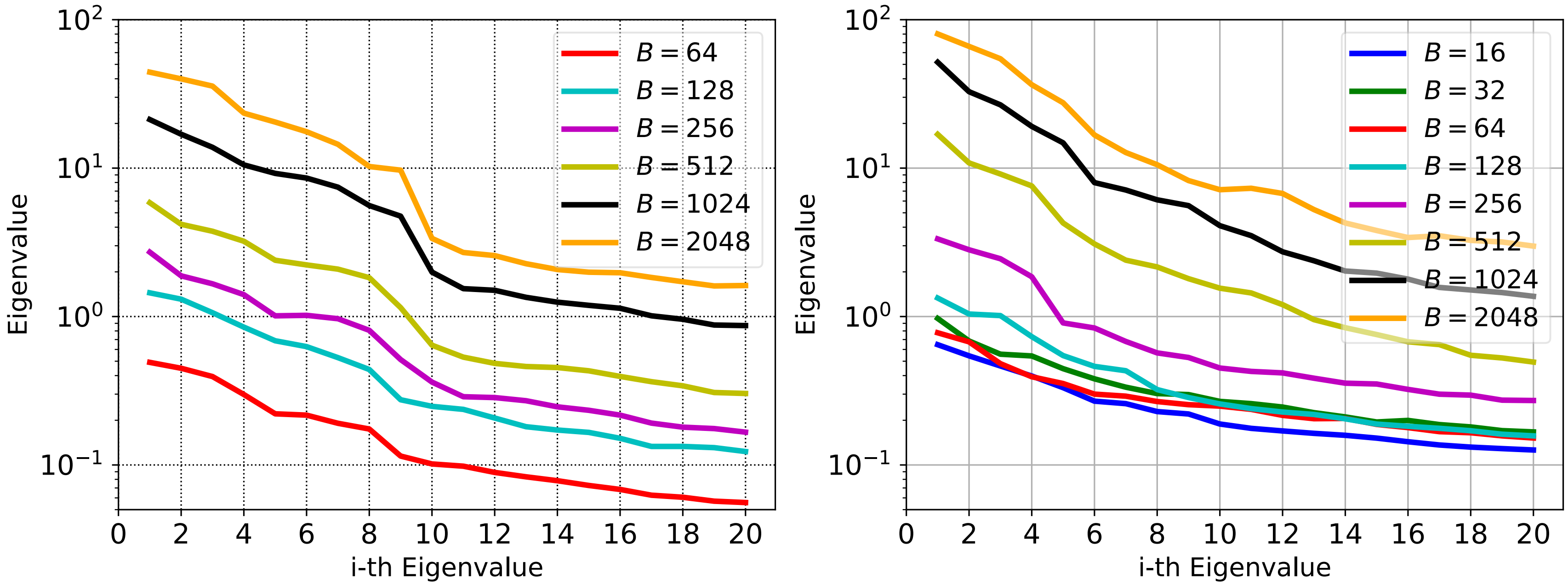
Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Loss Landscape at the end of training



Training/testing loss at the end of training along the dominant eigenvector of the Hessian (for Cifar-10 dataset)

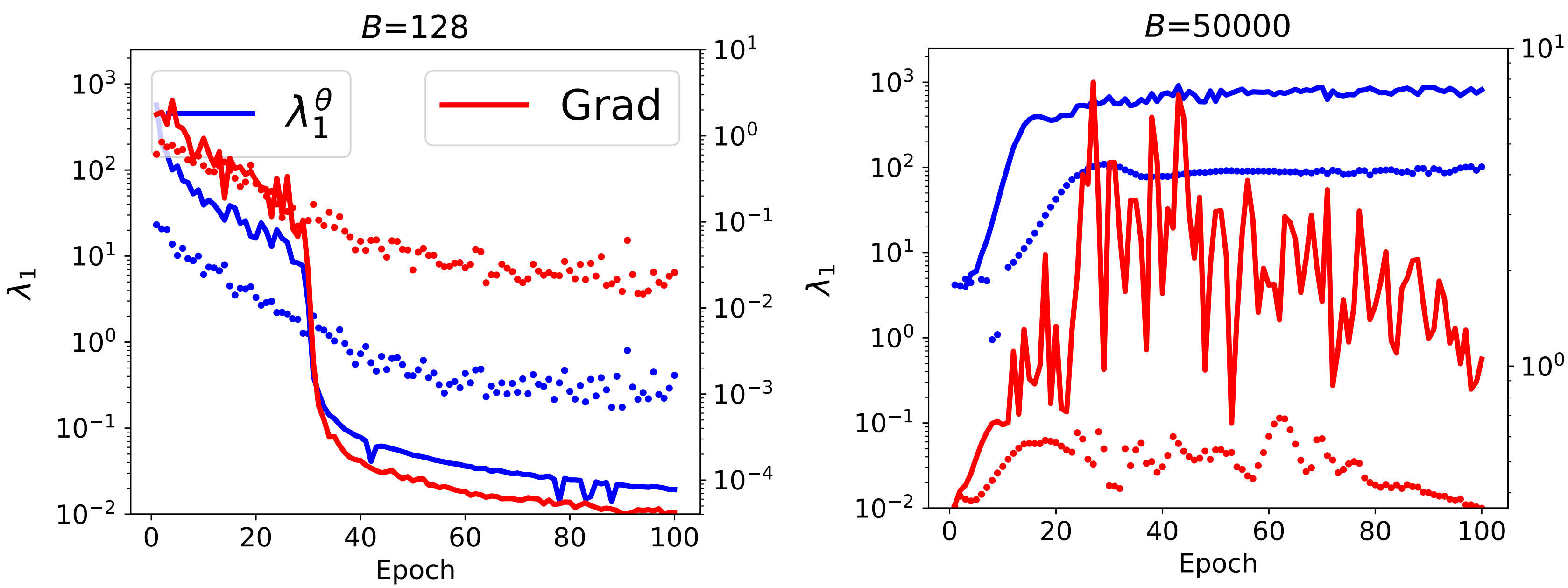
Large Batch Size Training



Top 20 eigenvalues of the total Hessian w.r.t. model parameters for different batch size. Clearly larger batch size converges to points with higher Hessian spectrum

Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Large Batch Size Training and Hessian Spectrum



Changes in the dominant eigenvalue of the Hessian w.r.t. weights and the total gradient is shown for different epochs during training. (Dotted = Robust Optimiz.) Large batch gets attracted to areas with larger Hessian spectrum (blue curve)

Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Hessian Based Adaptive Batch Size

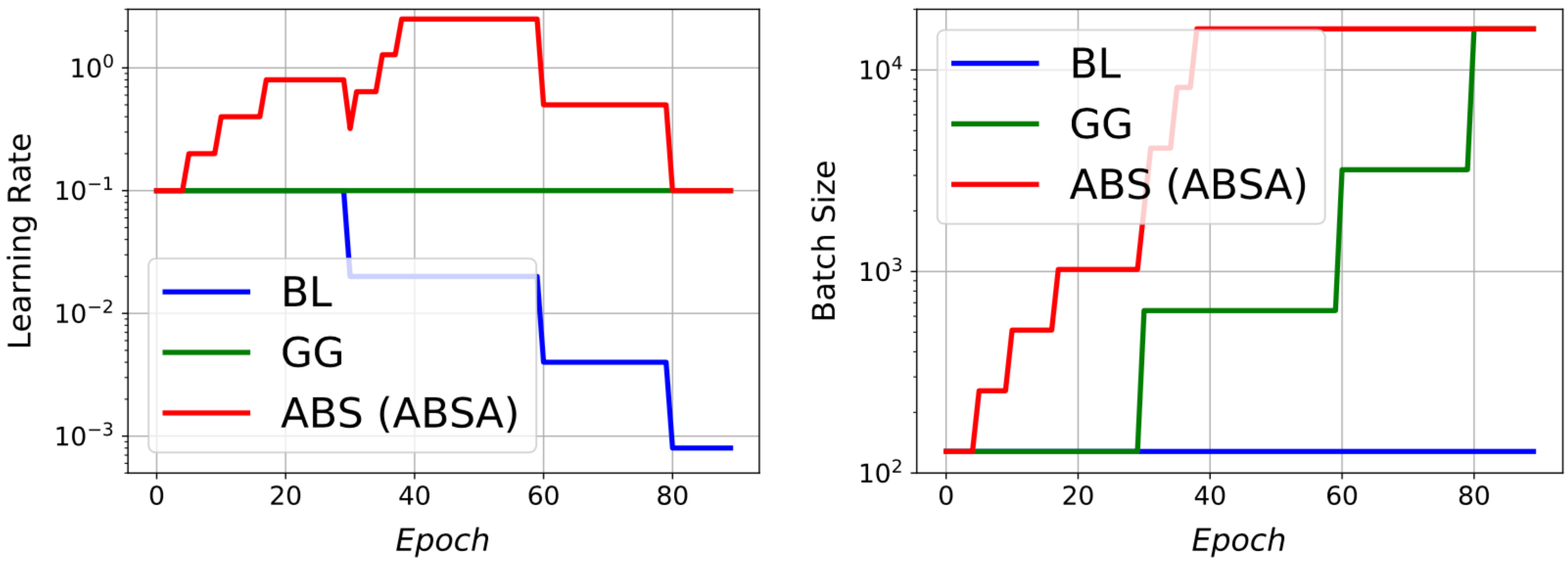
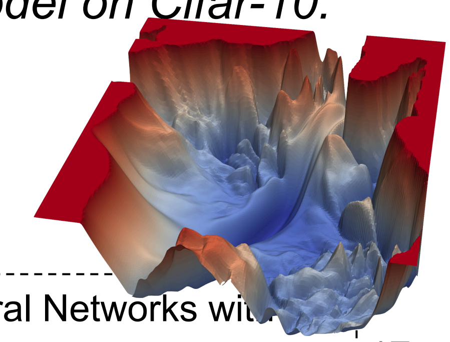


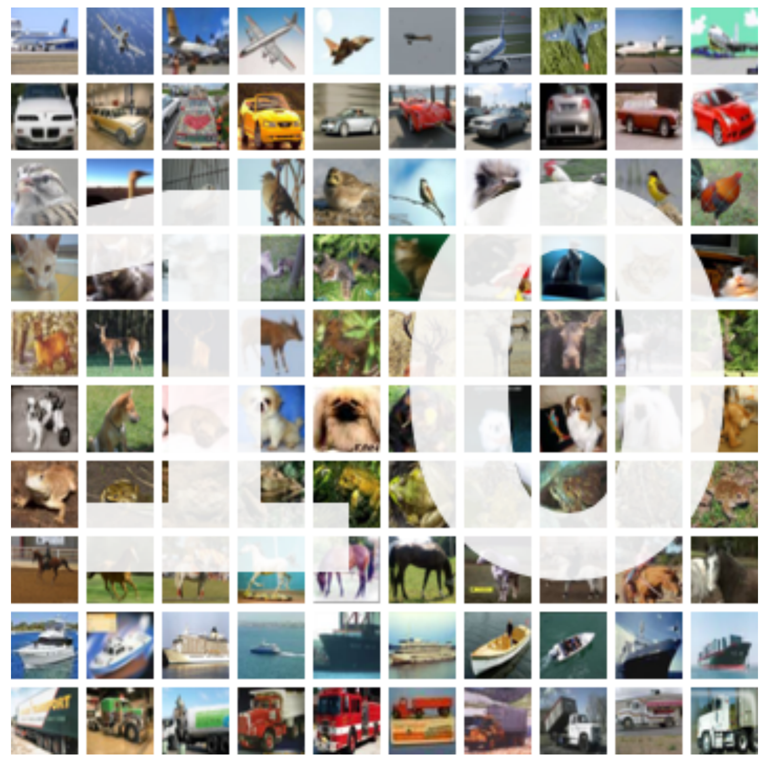
Illustration of learning rate (a) and batch size (b) schedules of adaptive batch size as a function of training epochs based on C2 model on Cifar-10.



Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks wit... Adversarial Training and Second-Order Information, (under review)

Results – Cifar 10

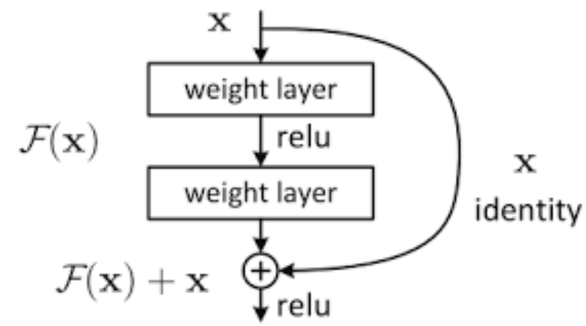
- Cifar-10 has ten classes
  - ~5000 examples per class
  - Total 50,000 training images
  - 10,000 testing images



Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

Results – Cifar10 - ResNet

- Our proposed method (ABSA) achieves better performance



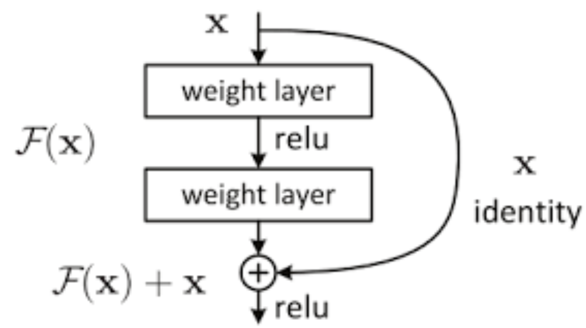
ResNet-18 on Cifar-10

	BL		FB		GG		ABS		ABSA	
batch size	Acc.	# updates	Acc.	# updates	Acc.	# updates	Acc.	# updates	Acc.	# updates
128	83.05	35156	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
640	81.01	7031	**84.59**	7031	83.99	16380	83.3	10578	84.52	9631
3200	74.54	1406	78.7	1406	84.27	14508	83.331	6375	**84.42**	5168
5120	70.64	878	74.65	878	83.47	14449	83.83	6575	**85.01**	6265
10240	68.75	439	30.99	439	83.68	14400	83.56	5709	**84.29**	7491
16000	67.88	281	10.	281	84.	14383	83.5	5739	**84.24**	5357

FB: Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677 (2017).
GG: Smith, Samuel L., Pieter-Jan Kindermans, and Quoc V. Le. "Don't Decay the Learning Rate, Increase the Batch Size." arXiv preprint arXiv:1711.00489 (2017).
ABS/ABSA: Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

Results – Cifar10 – Wide ResNet

- Our proposed method (ABSA) achieves better performance

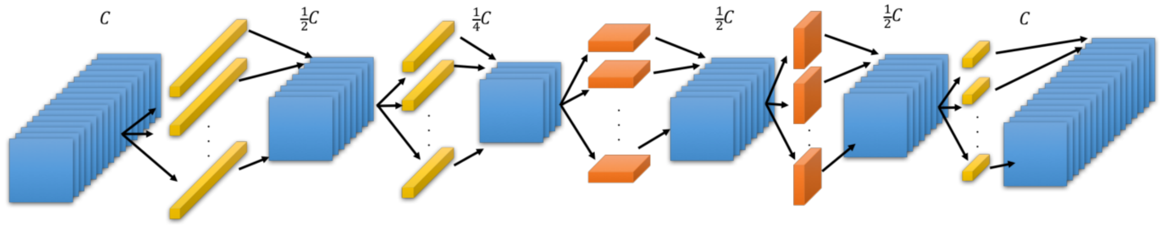


WResNet16-4 on Cifar-10

	BL		FB		GG		ABS		ABSA	
batch size	Acc.	# updates	Acc.	# updates	Acc.	# updates	Acc.	# updates	Acc.	# updates
128	87.64	35156	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
640	86.2	7031	87.9	7031	87.84	16380	87.86	10399	**89.05**	10245
3200	82.59	1406	73.2	1406	87.59	14508	88.02	5869	**89.04**	4525
5120	81.4	878	63.27	878	87.85	14449	87.92	7479	**88.64**	5863
10240	79.85	439	0.1	439	87.52	14400	87.84	5619	**89.03**	3929
16000	81.06	281	0.1	281	88.28	14383	87.58	9321	**89.19**	4610

FB: Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677 (2017).
GG: Smith, Samuel L., Pieter-Jan Kindermans, and Quoc V. Le. "Don't Decay the Learning Rate, Increase the Batch Size." arXiv preprint arXiv:1711.00489 (2017).
ABS/ABSA: Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

Results – Cifar10 - SqueezeNext



1.0-SqueezeNext on Cifar-10

	BL		GG		ABS		ABSA	
BS	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters
128	92.02	78125	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
256	91.88	39062	91.84	50700	91.7	40792	**92.11**	43352
512	91.68	19531	91.19	37050	**92.15**	32428	91.61	25388
1024	89.44	9766	91.12	31980	91.61	17046	**91.66**	23446
2048	83.17	4882	89.19	30030	91.57	21579	**91.61**	14027
4096	73.74	2441	91.83	29191	91.91	18293	**92.07**	21909
8192	63.71	1220	91.51	28947	91.77	22802	**91.81**	16778
16384	47.84	610	90.19	28828	**92.12**	17485	91.97	24361

FB: Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677 (2017).
GG: Smith, Samuel L., Pieter-Jan Kindermans, and Quoc V. Le. "Don't Decay the Learning Rate, Increase the Batch Size." arXiv preprint arXiv:1711.00489 (2017).
ABS/ABSA: Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

Results – SVHN

- SVHN consists of ten classes
  - Total 600,000 training images
  - 26,000 testing images

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng Reading Digits in Natural Images with Unsupervised Feature Learning NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011

Results – SVHN - AlexNet

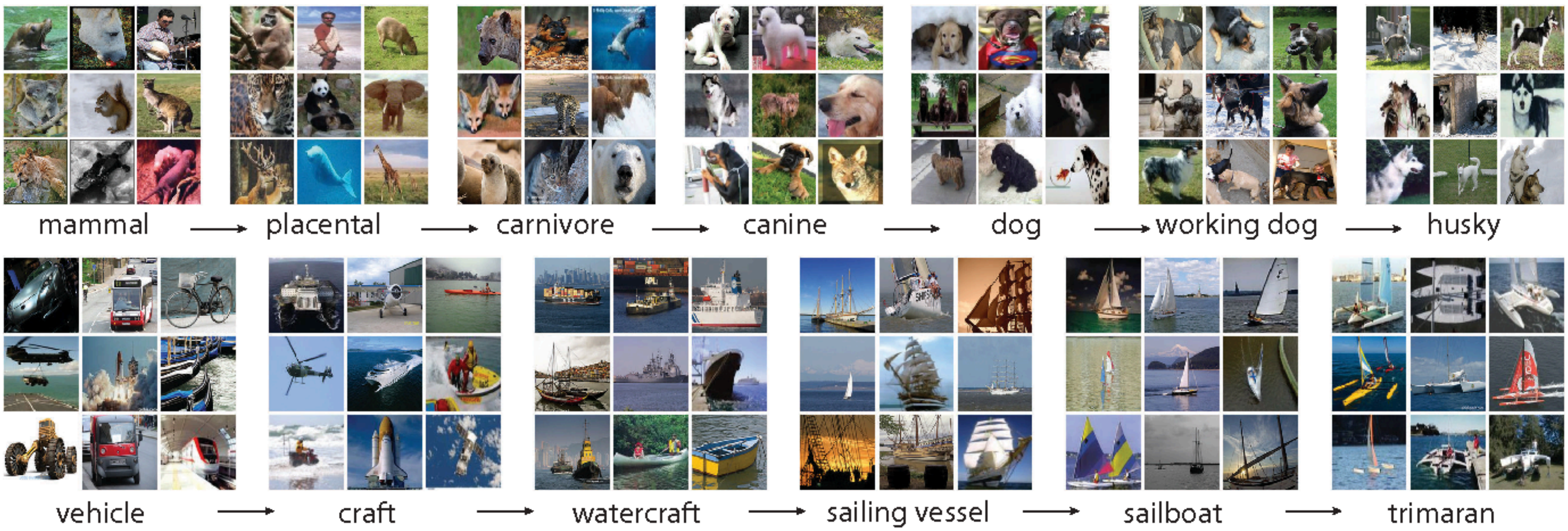
AlexNet on SVHN

	BL		FB		GG		ABS		ABSA	
BS	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters
128	94.90	81986	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
512	94.76	20747	95.24	20747	95.49	51862	95.65	25353	**95.72**	24329
2048	95.17	5186	95.00	5186	95.59	45935	95.51	10562	**95.82**	16578
8192	93.73	1296	19.58	1296	**95.70**	44407	95.56	14400	95.61	7776
32768	91.03	324	10.0	324	95.60	42867	95.60	7996	**95.90**	12616
131072	84.75	81	10.0	81	95.58	42158	95.61	11927	**95.92**	11267

FB: Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677 (2017).
GG: Smith, Samuel L., Pieter-Jan Kindermans, and Quoc V. Le. "Don't Decay the Learning Rate, Increase the Batch Size." arXiv preprint arXiv:1711.00489 (2017).
ABS/ABSA: Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

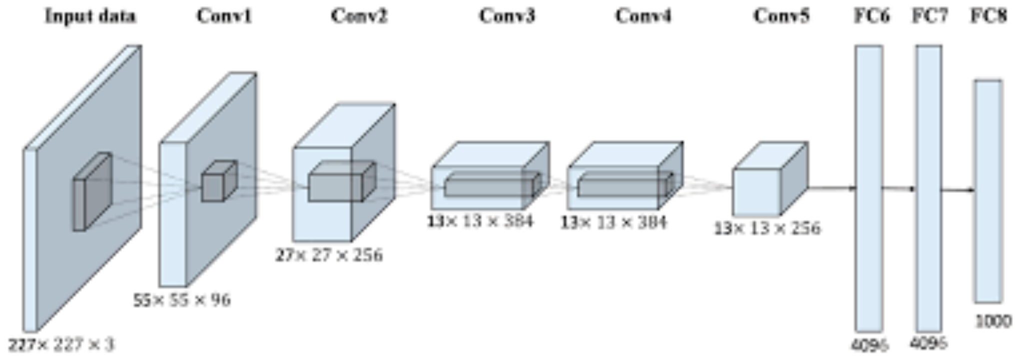
Results – ImageNet

- ImageNet consists of 1000 classes
  - Total 1.2 million training images
  - 50,000 testing images



Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115.3 (2015): 211-252

Results – ImageNet - AlexNet



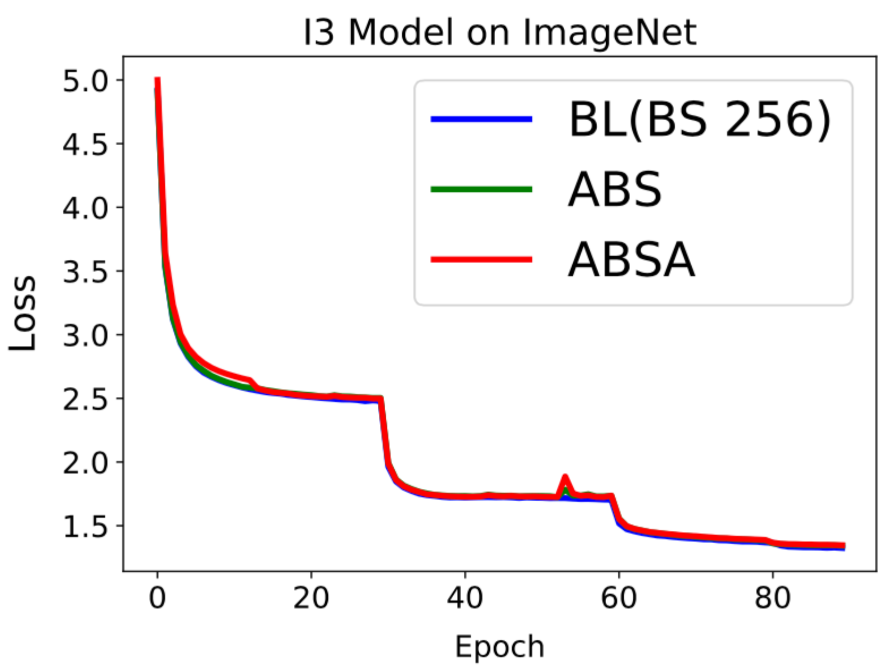
ResNet50 on Tiny ImageNet

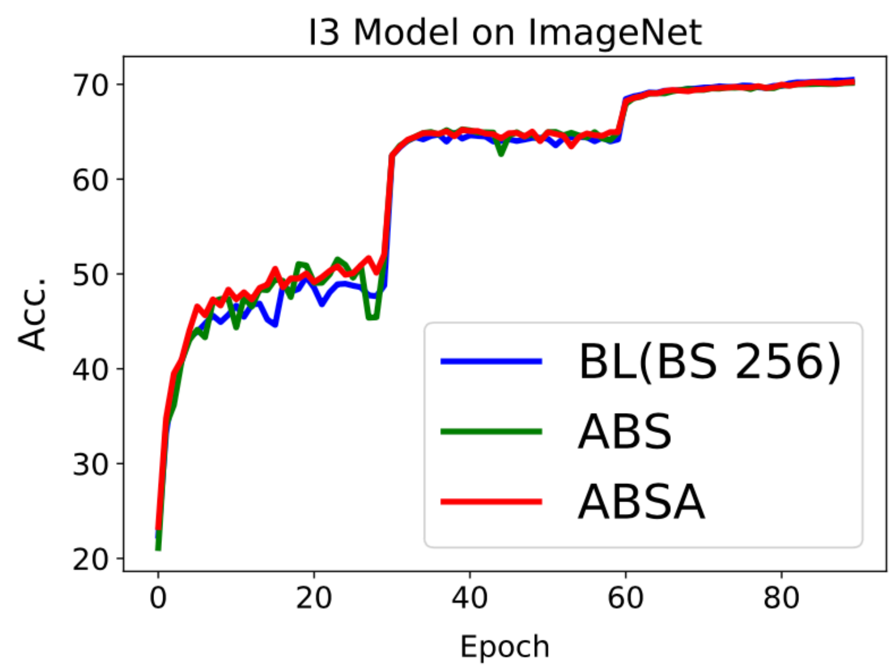
	BL		FB		GG		ABSA	
BS	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters	Acc.	# Iters
128	60.41	93750	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
256	58.24	46875	59.82	46875	60.31	70290	**61.28**	60684
512	57.48	23437	59.28	23437	59.94	58575	**60.55**	51078
1024	54.14	11718	59.62	11718	59.72	52717	**60.72**	19011
2048	50.89	5859	59.18	5859	59.82	50667	**60.43**	17313
4096	40.97	2929	58.26	2929	60.09	49935	**61.14**	22704
8192	25.01	1464	16.48	1464	60.00	49569	**60.71**	22334
16384	10.21	732	0.05	732	60.37	48995	**60.71**	20348

FB: Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677 (2017).
GG: Smith, Samuel L., Pieter-Jan Kindermans, and Quoc V. Le. "Don't Decay the Learning Rate, Increase the Batch Size." arXiv preprint arXiv:1711.00489 (2017).
ABS/ABSA: Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

Results – ImageNet – ResNet18

- Baseline:
  - **450k** SGD iterations, **70.4%** validation accuracy
- ABSA:
  - **66k** SGD iterations, **70.2%** validation accuracy
- GG would have required **166k** SGD iterations





Hessian Based Adaptive Batch Size

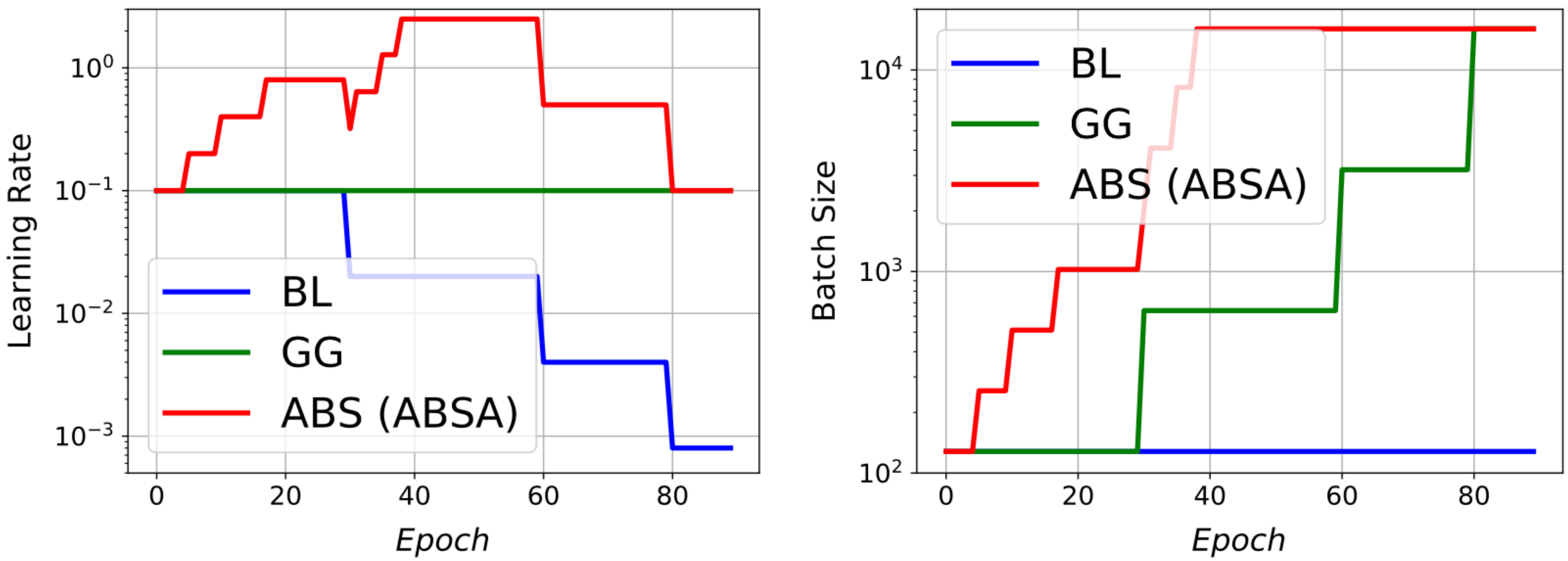
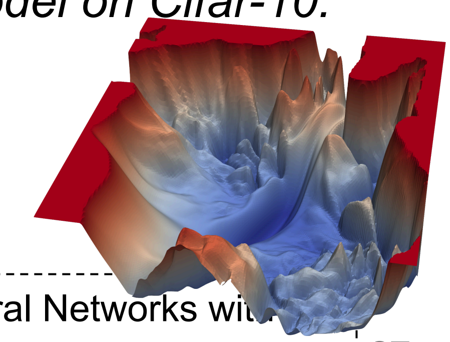


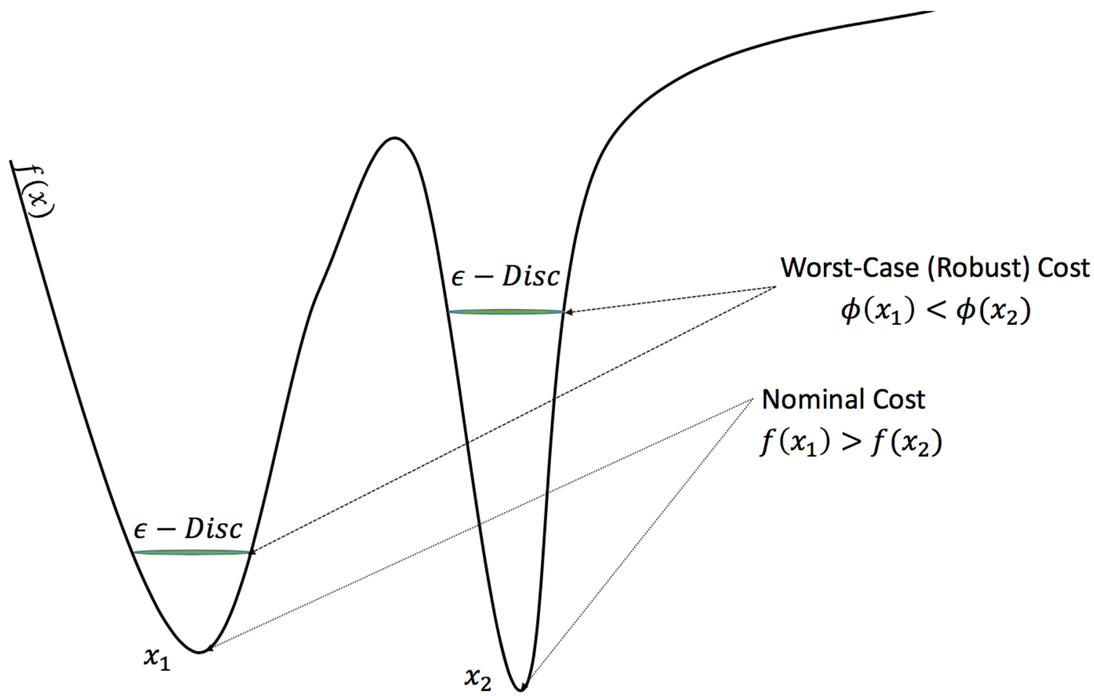
Illustration of learning rate (a) and batch size (b) schedules of adaptive batch size as a function of training epochs based on C2 model on Cifar-10.



Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks wit… Adversarial Training and Second-Order Information, (under review)

What is Robust Optimization?

- Instead of minimizing for the average case, consider the worst case under “some metric”



Keskar, Nitish Shirish, et al. "On large-batch training for deep learning: Generalization gap and sharp minima." ICLR’16 (arXiv:1609.04836).

Robust Optimization

- **Consider a simple linear programming problem**

$$\min_x \{c^T x : Ax \le b\}$$

- **Now assume input data (A,b,c) is uncertain**

$$\min_x \sup_{(A,b,c)\in\mathcal{U}} \{c^T x : Ax \le b\}$$

- **Where u is the uncertainty set**

Shaham U, Yamada Y, Negahban S. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. arXiv preprint arXiv:1511.05432. 2015 Nov 17.

Robust Optimization and Regularization

- There is an interesting connection between the solution to robust optimization and a properly regularized problem

- Robust solution to least squares under bounded uncertainty in A is equivalent to lasso regularized one

$$\min_x \max_{\|\Delta A\|_{\infty,2} \le \rho} \|(A + \Delta A)x - b\|$$

$$\min_x \|Ax - b\| + \lambda \|x\|_1$$

El Ghaoui, Laurent, and Hervé Lebret. "Robust solutions to least-squares problems with uncertain data." SIAM Journal on matrix analysis and applications 18.4 (1997): 1035-1064.
Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. In *Advances in Neural Information Processing Systems*, pages 1801–1808, 2009.

Robust Optimization in NN

° Find the NN's parameters using a min-max optimization, instead of just the min

- Solving the max problem is computationally infeasible
- A practical solution is to perform **gradient ascent** to iteratively solve the max problem and then **gradient descent** for the min part

$$\min_{\theta} \tilde{J}(\theta, x, y) = \min_{\theta} \sum_{i=1}^{m} \max_{\tilde{x}_i \in \mathcal{U}_i} J(\theta, \tilde{x}_i, y_i)$$

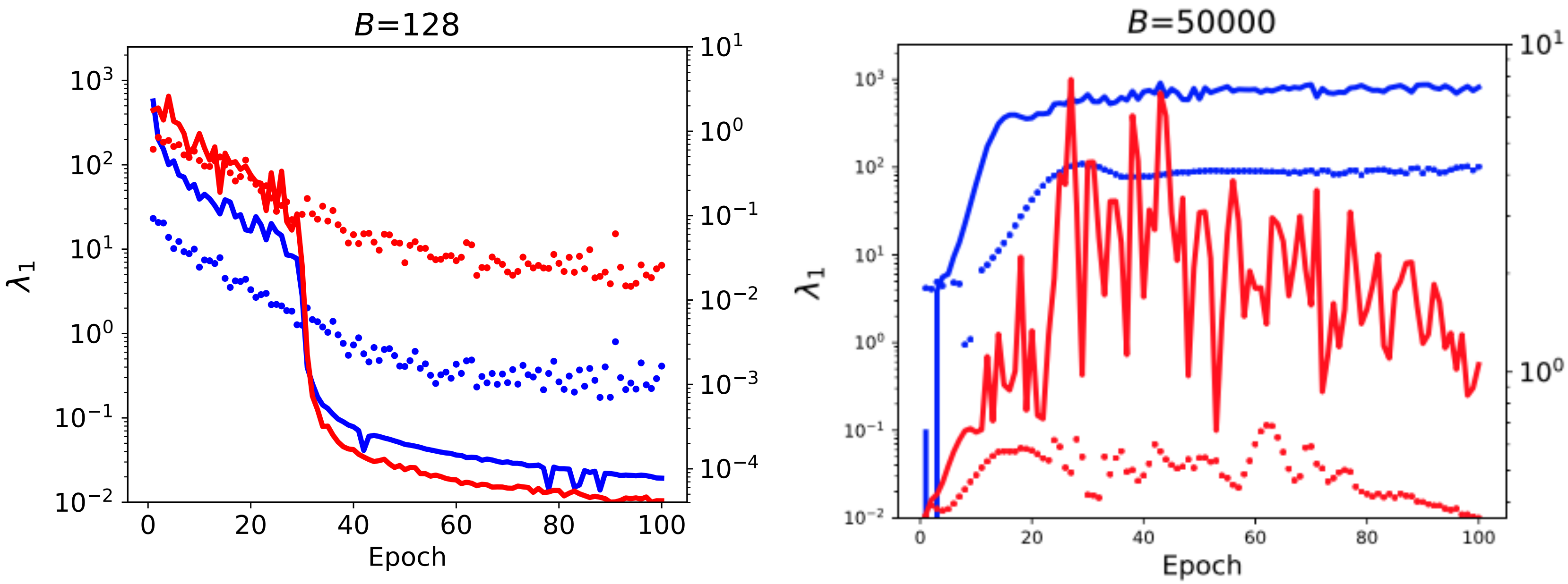
Problem with SGD

- For convex problem, under proper conditions, we have
- $R(W) := \mathbb{E}[f(W,X)] \le \frac{1}{n}\sum_{i=0}^{n} f(W, x_i) + C_1\sqrt{\frac{Var(f(W,X))}{n}} + \frac{C_2}{n}$
- Empirical Risk Minimization only minimizes the first bias term $\frac{1}{n}\sum_{i=0}^{n} f(W, x_i)$
- Robust Optimization considers bias term $\frac{1}{n}\sum_{i=0}^{n} f(W, x_i)$ and variance term $\sqrt{Var(f(W,X))/n}$ together, back to the famous tradeoff, bias-variance tradeoff, in Machine Learning community . That can improve test performance.

H. Namkoong and J. C. Duchi. *Variance regularization with convex objectives*, NIPS'17

Robust Optimization as a Regularizer

- Robust optimization regularizes the model away from "sharp minimas"



Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Hessian Overhead!

- **we present the breakdown of one SGD update training time in terms of forward/backwards computation (Tcomp), one step communication time (Tcomm), one total Hessian spectrum computation (if any THess), and the total training time. The results correspond to ResNet18 model on ImageNet.**

Method	T_{comp}	T_{comm}	T_{Hess}	Total Time
BL	2.2E-2	1.5E-2	0.	16666
GG	2.2E-2	1.5E-2	0.	6150 (2.71× faster)
ABS	2.2E-2	1.5E-2	1.15	2666 (6.25× faster)
ABSA	3.6E-2	1.5E-2	1.15	3467 (4.80× faster)

ABS Convergence Proof

- **For a convex problem we have the following:**

Theorem 2. *Under Assumption 1, let assume at step t, the batch size used for parameter update is b_t, the step size is $b_t\eta_0$, where η_0 is fixed and satisfies,*

$$0 < \eta_0 \leq \frac{1}{L_g(M_v + B_{\max})}, \tag{10}$$

where $B_{\max}$ is the maximum batch size during training. Then, the expected optimality gap satisfies the following inequality,

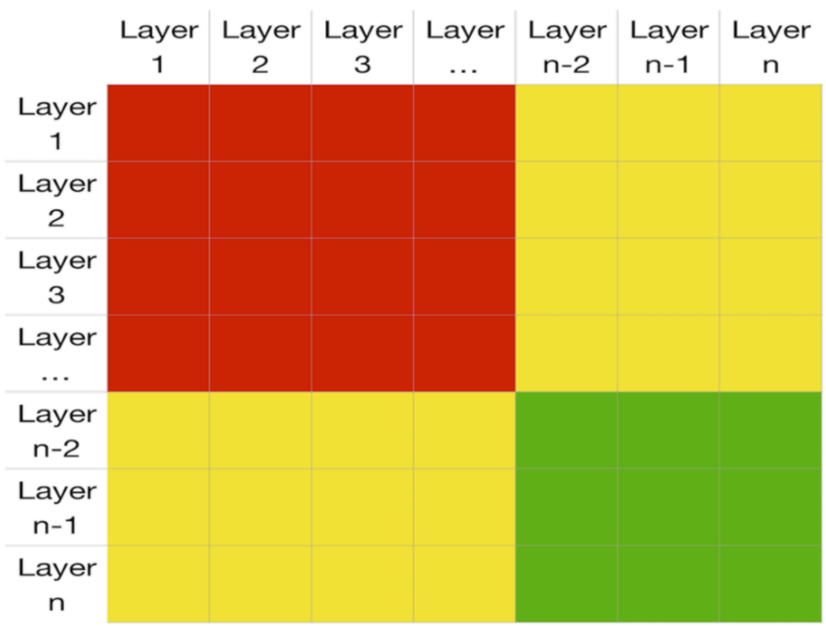
$$\mathbb{E}[L(\theta_{t+1})] - L_* \leq \prod_{k=1}^{t}(1 - b_k\eta_0 c_s)(L(\theta_0) - L_* - \frac{\eta_0 L_g M}{2c_s}) + \frac{\eta_0 L_g M}{2c_s}, \tag{11}$$

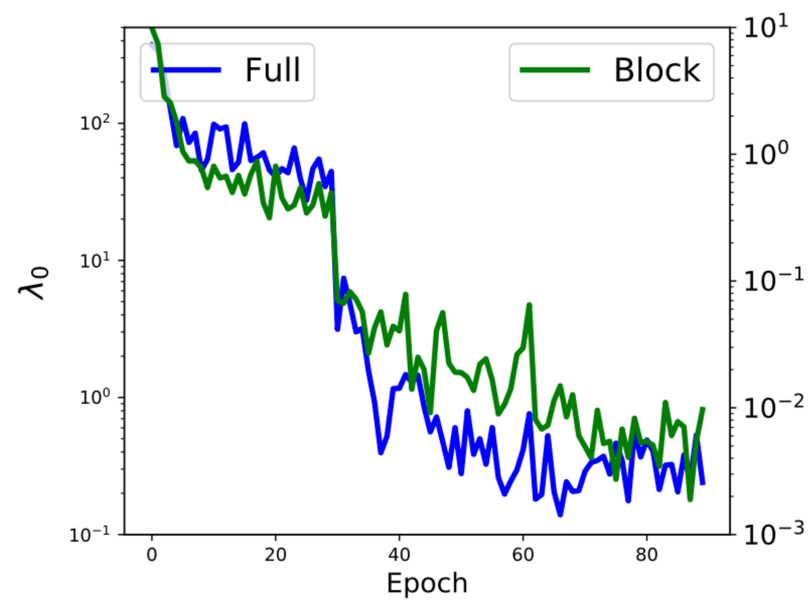
where θ_0 is the initialization.

Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

Approximate Hessian Computation

- **Using block approximation to Hessian and analyzing last layer of a deep neural network seems to contain enough signal for ABSA**

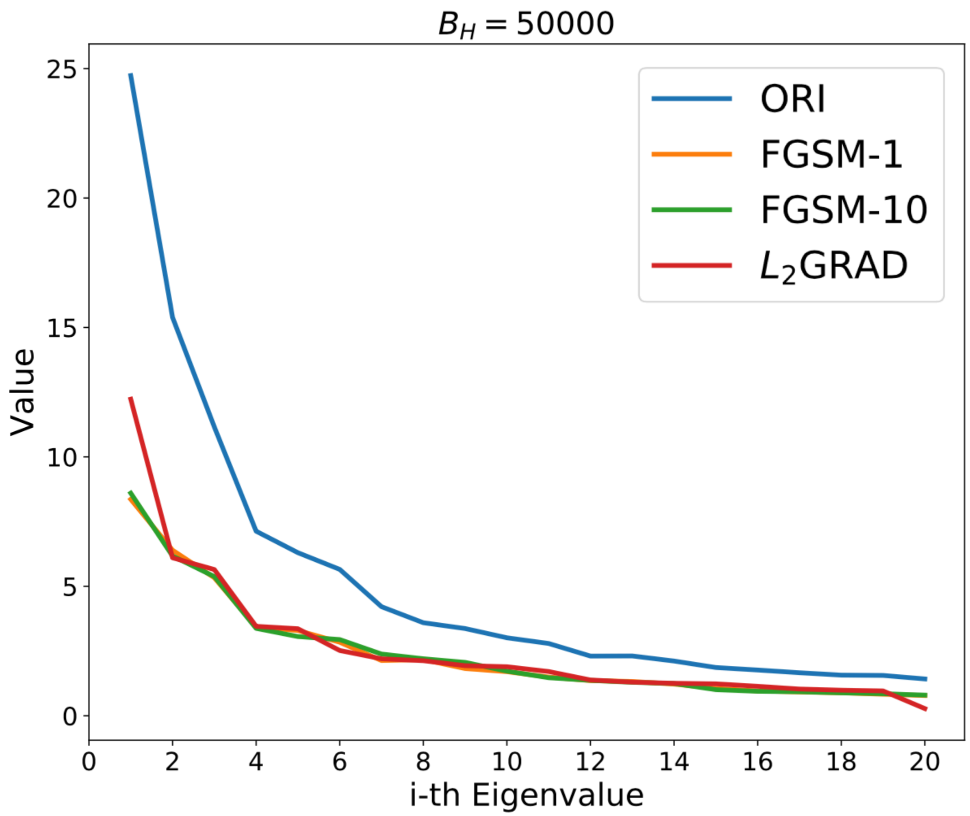


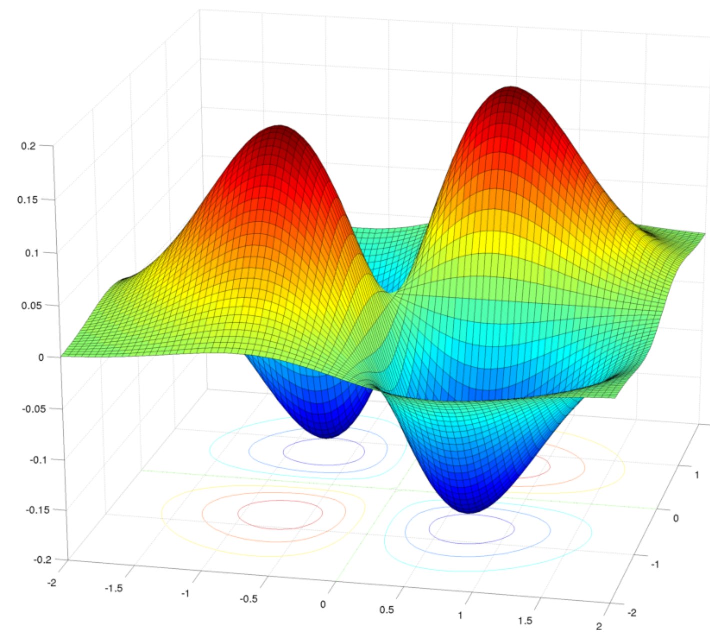


Z. Yao, A. Gholami, K. Keutzer, M. Mahoney, Large Batch Size Training of Neural Networks with Adversarial Training and Second-Order Information, (under review)

Problem with SGD

- **SGD only reduces the bias term**



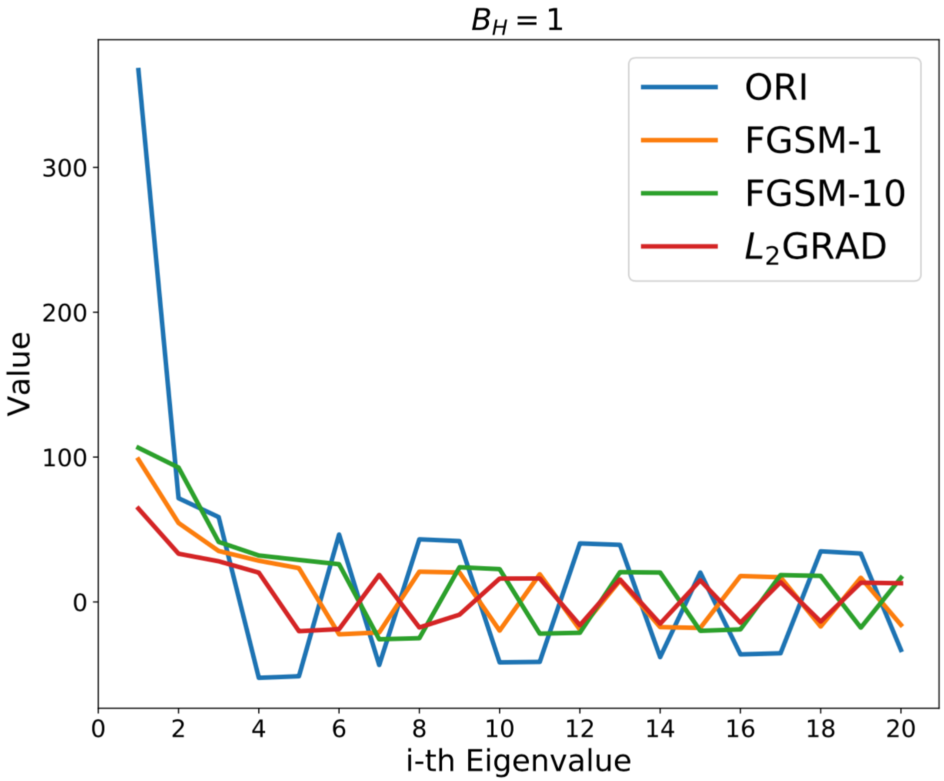


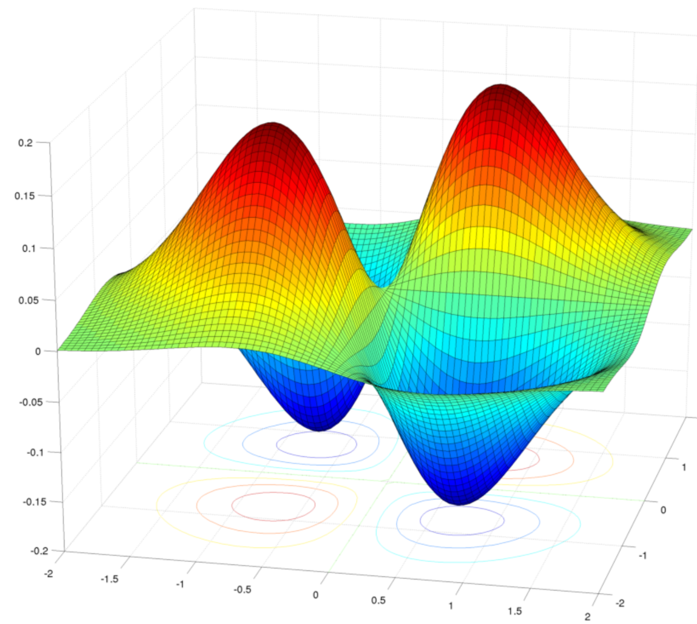
Top 20 eigenvalues of the total Hessian w.r.t. model parameters is shown after the training is finished.

Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Problem with SGD

- SGD only reduces the bias term



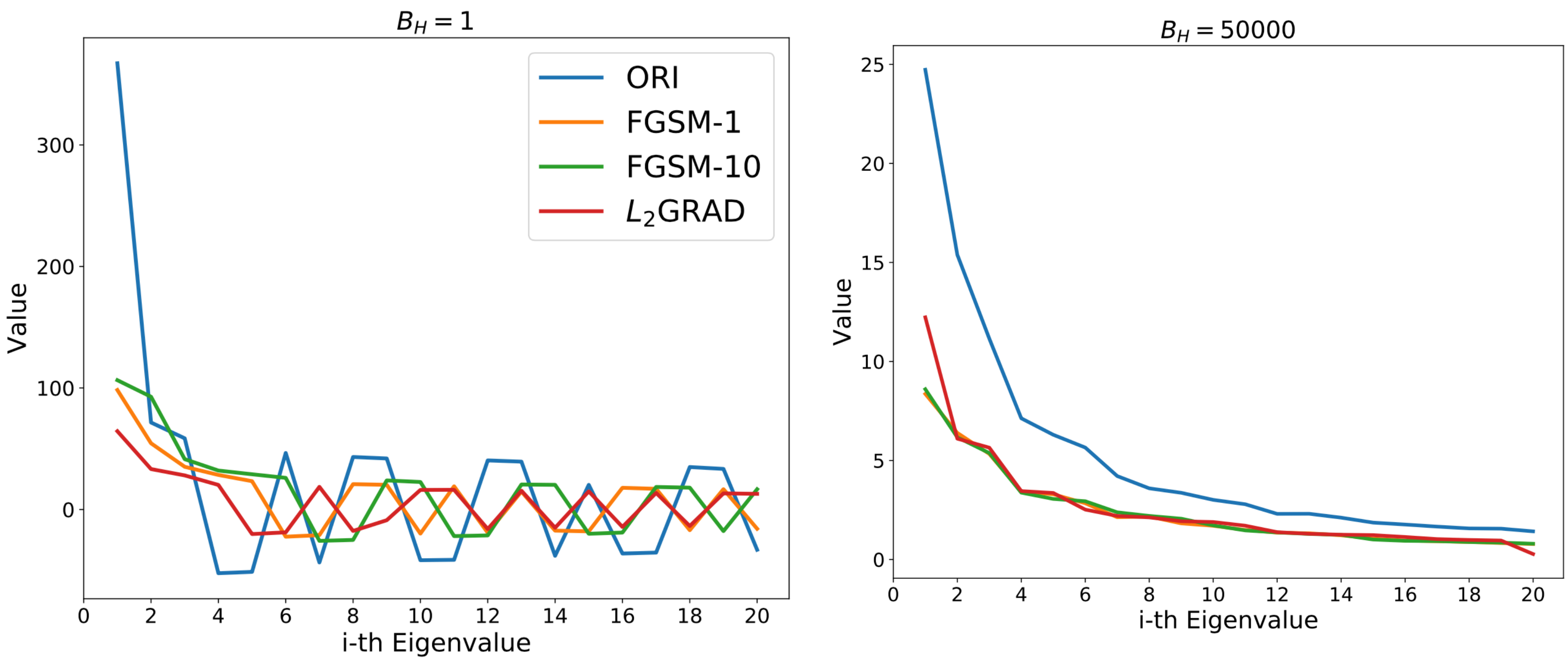


Top 20 eigenvalues of one sample Hessian w.r.t. model parameters is shown after the training is finished.

Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Robust Optimization

- **Bring back the plots for the hessian and explain that now the results look better**



Top 20 eigenvalues of the total Hessian w.r.t model parameters is shown after the training is finished.

Z. Yao, A. Gholami, Q. Lei, K. Keutzer, M. Mahoney. Hessian-based Analysis of Large Batch Training and Robustness to Adversaries, NIPS'18 (arXiv:1802.08241)

Summary

- Stochastic optimization using SGD often times does not lead to robust and it is very sensitive to hyper-parameters that are not optimal
- Robust optimization helps increase stability of the model to adversarial inputs
- Incorporating robust optimization can often time lead to solutions that have superior generalization performance than the baseline network trained with SGD and is more robust to adversarial perturbation
- Future work would include combining robust optimization with stochastic second order methods such as inexact sub-sampled Hessian

Thank You