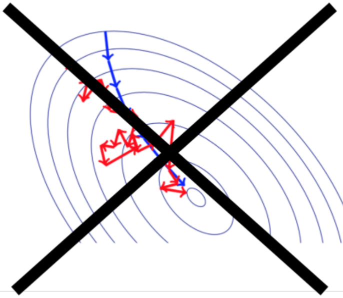
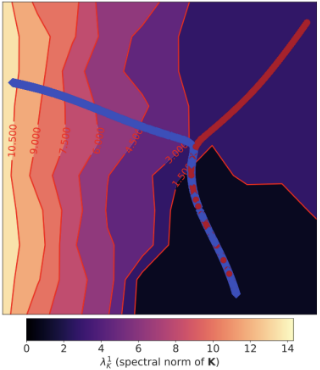


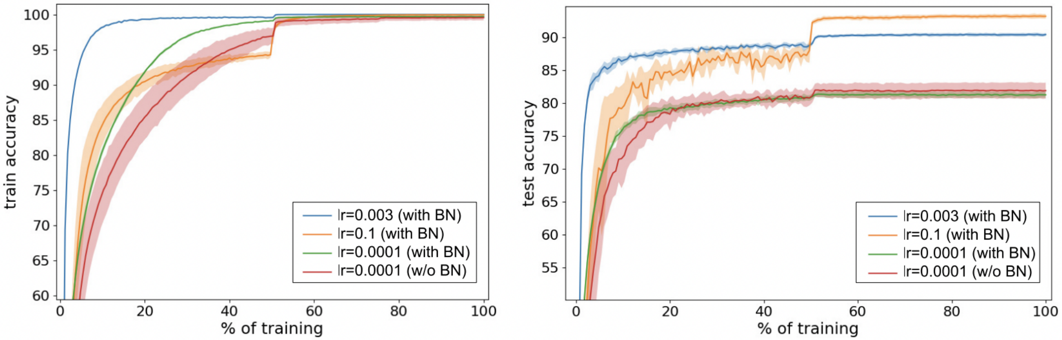


Reverse-engineering Implicit Regularization Due to SGD

Stanisław Jastrzębski

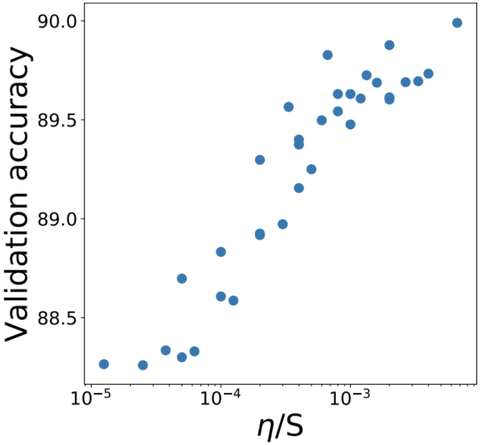
Research Question

Why Does Learning Rate Influence Generalization?



Bjorck et al. [2018]

Why Does the Learning Rate Influence Generalization?



Jastrzebski et al. [2017]

Why Does Learning Rate Influence Generalization?

"The learning rate is perhaps the most important hyperparameter. If you have time to tune only one hyperparameter, tune the learning rate."

Goodfellow et al. [2014]

Research Question

Why does optimization impacts generalization in deep learning?

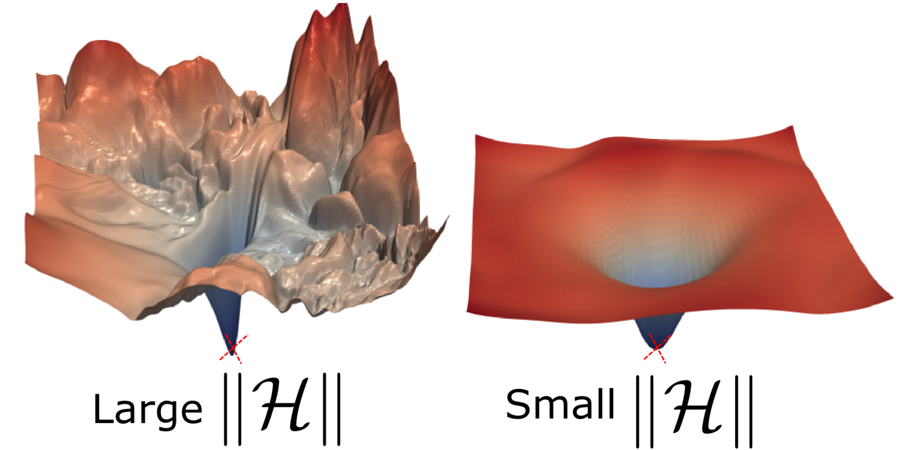
Instability in the Early Phase

On the Relation Between the Sharpest Directions of DNN loss and SGD Step Length, S. Jastrzebski, Z. Kenton, N. Ballas, A. Fischer, Y. Bengio, A. Storkey, ICLR 2019

The Break-Even Point on Optimization Trajectories of Deep Neural Networks, S. Jastrzebski, M. Szymczak, S. Fort, D. Arpit, J. Tabor, K. Cho*, K. Geras*, ICLR 2020 (Spotlight)

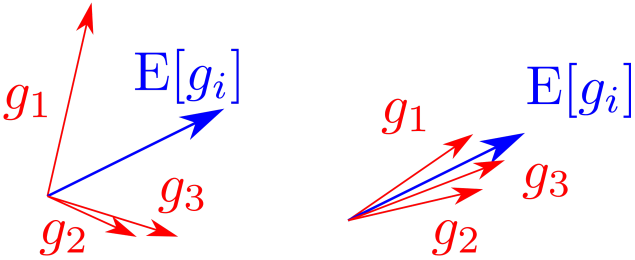
Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability, J. Cohen, S. Kaur, Y. Li, J Z. Kolter, A. Talwalkar ICLR 2021

Hessian of the Training Loss



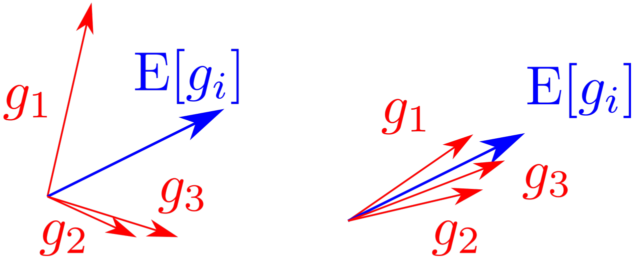
$\mathbf{H}(\theta) = \frac{\partial^2}{\partial \theta^2}\mathcal{L}(\theta)$ with a large or small norm ($\|\mathbf{H}\|$).

Covariance of Gradients



$\mathbf{K} = \mathrm{Cov}[\mathbf{g_i}]$ with large (left) or small (right) $\|\mathbf{K}\|$.

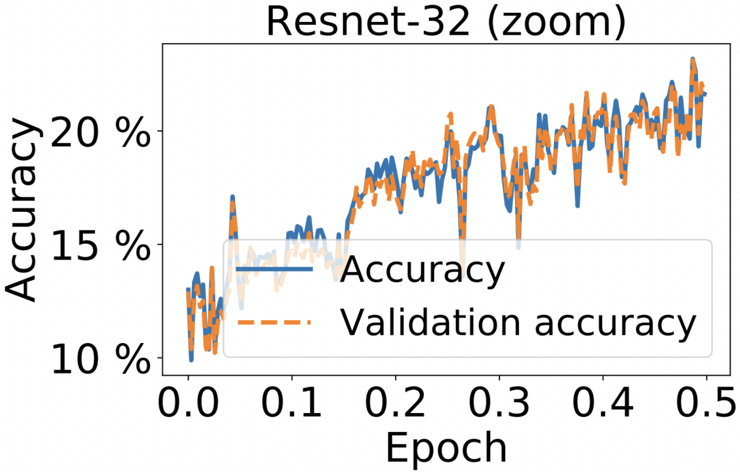
Covariance of Gradients



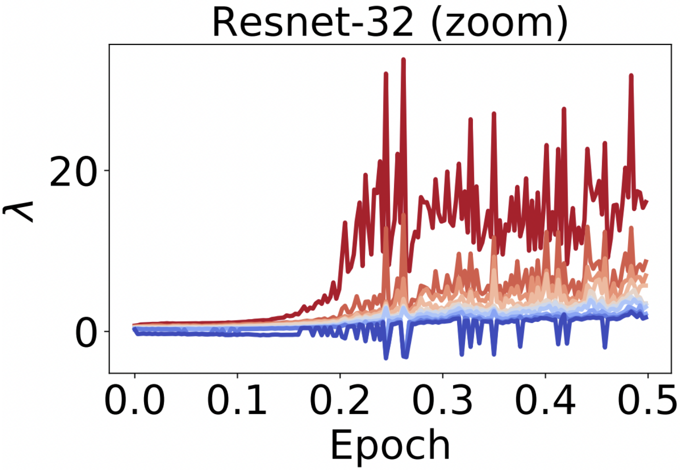
$\mathbf{K} = \mathrm{Cov}[\mathbf{g_i}]$ with large (left) or small (right) $\|\mathbf{K}\|$.

- λ_K^i, λ_H^i will denote the largest eigenvalues of the covariance of gradients ($\mathbf{K}$) and the Hessian ($\mathbf{H}$).
- $\mathrm{Tr}(\mathbf{K}) = \mathbb{E}[\|g_i - g\|^2]$ (variance of gradients).

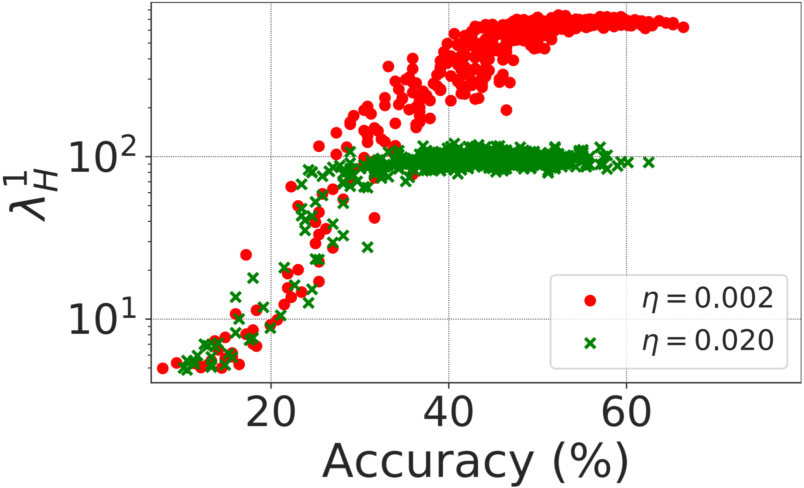
How does the Hessian Changes During Training?



How does the Hessian Changes During Training?

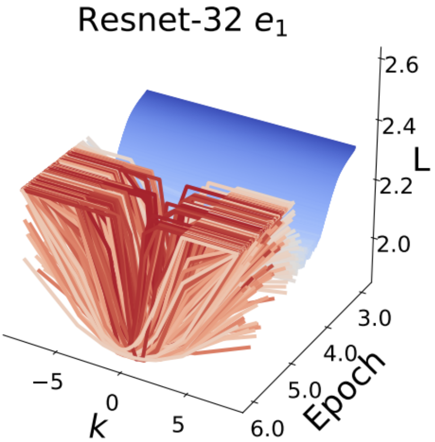


How does the Hessian Changes During Training?



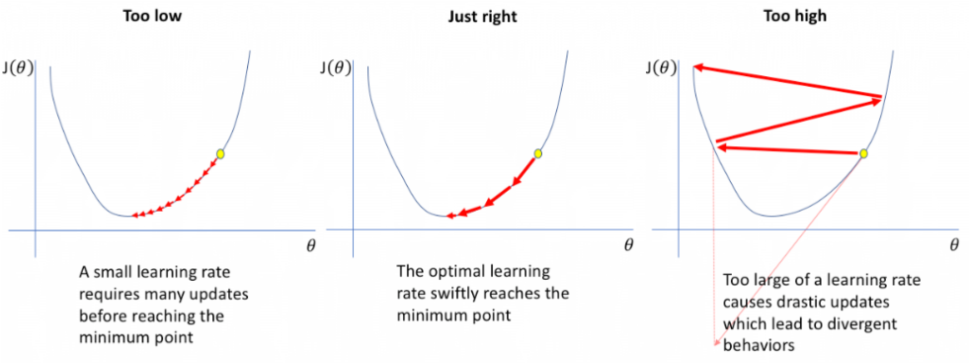
Visualizing the Early Phase

Visualizing the Early Phase



Jastrzebski et al. [2018]

The Role of the Learning Rate is Counterintuitive.



Break-Even Point: What Happens When we Train with Two Learning Rates?

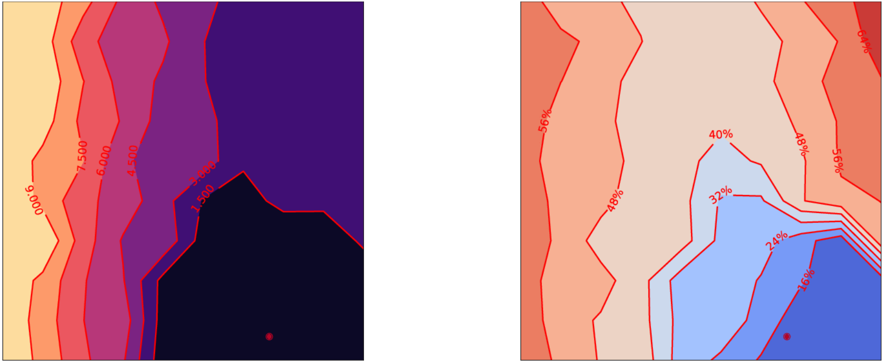


Figure: Visualization of the early part of the optimization trajectories, for SimpleCNN on the CIFAR-10 dataset. Red is $\eta = 0.1$, blue is $\eta = 0.01$. The background color indicates the spectral norm of the covariance of gradients $\mathbf{K}$ (λ_K^1, left) and the training accuracy (right).

Break-Even Point: What Happens When we Train with Two Learning Rates?

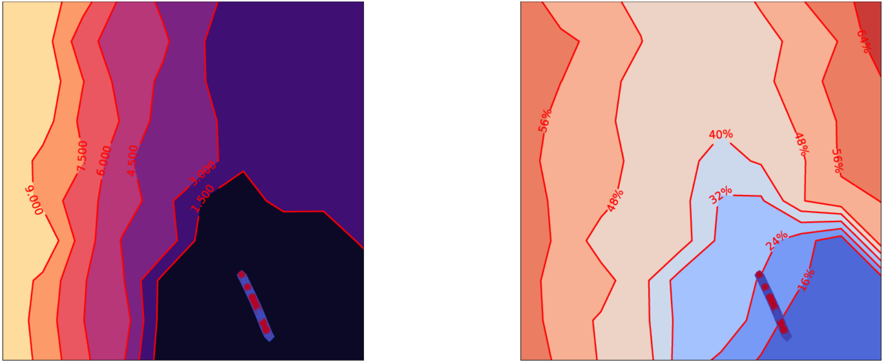
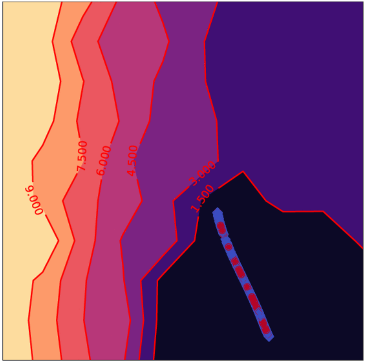


Figure: Visualization of the early part of the optimization trajectories, for SimpleCNN on the CIFAR-10 dataset. Red is $\eta = 0.1$, blue is $\eta = 0.01$. The background color indicates the spectral norm of the covariance of gradients $\mathbf{K}$ (λ_K^1, left) and the training accuracy (right).

Break-Even Point: What Happens When we Train with Two Learning Rates?



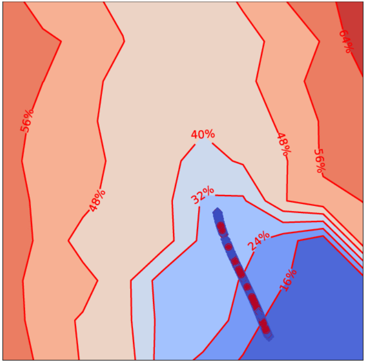


Figure: Visualization of the early part of the optimization trajectories, for SimpleCNN on the CIFAR-10 dataset. Red is $\eta = 0.1$, blue is $\eta = 0.01$. The background color indicates the spectral norm of the covariance of gradients $\mathbf{K}$ (λ_K^1, left) and the training accuracy (right).

Break-Even Point: What Happens When we Train with Two Learning Rates?

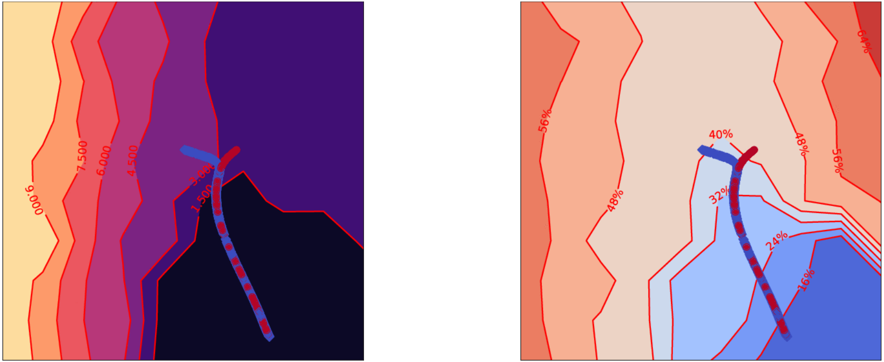
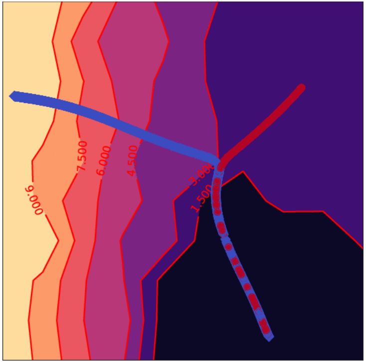


Figure: Visualization of the early part of the optimization trajectories, for SimpleCNN on the CIFAR-10 dataset. Red is $\eta = 0.1$, blue is $\eta = 0.01$. The background color indicates the spectral norm of the covariance of gradients $\mathbf{K}$ (λ_K^1, left) and the training accuracy (right).

Break-Even Point: What Happens When we Train with Two Learning Rates?



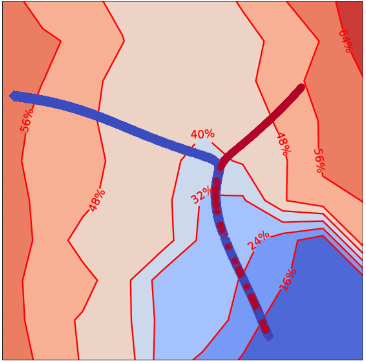


Figure: Visualization of the early part of the optimization trajectories, for SimpleCNN on the CIFAR-10 dataset. Red is $\eta = 0.1$, blue is $\eta = 0.01$. The background color indicates the spectral norm of the covariance of gradients $\mathbf{K}$ (λ_K^1, left) and the training accuracy (right).

Break-Even Point: What Happens When we Train with Two Learning Rates?

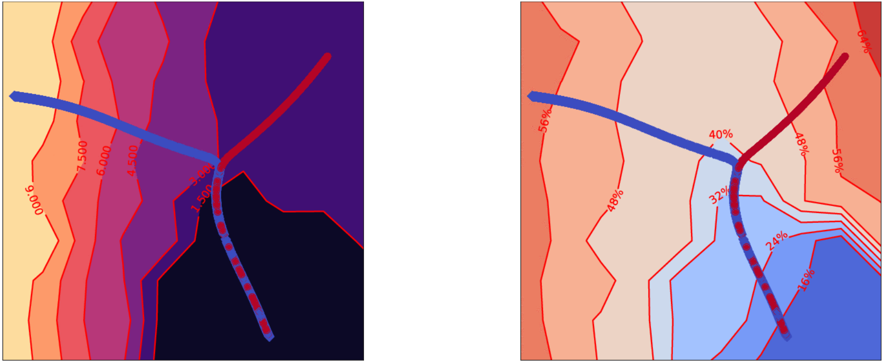


Figure: Visualization of the early part of the optimization trajectories, for SimpleCNN on the CIFAR-10 dataset. Red is $\eta = 0.1$, blue is $\eta = 0.01$. The background color indicates the spectral norm of the covariance of gradients $\mathbf{K}$ (λ_K^1, left) and the training accuracy (right).

Novel Implicit Regularization Effects of SGD

Conjecture (Variance reduction effect of SGD)

Along the SGD trajectory, the maximum attained values of λ_H^1 *and* λ_K^1 *are smaller for a larger learning rate or a smaller batch size.*

Novel Implicit Regularization Effects of SGD

Conjecture (Variance reduction effect of SGD)

Along the SGD trajectory, the maximum attained values of λ_H^1 *and* λ_K^1 *are smaller for a larger learning rate or a smaller batch size.*

Conjecture (Pre-conditioning effect of SGD)

Along the SGD trajectory, the maximum attained values of $\frac{\lambda_K^*}{\lambda_K^1}$ *and* $\frac{\lambda_H^*}{\lambda_H^1}$ *are larger for a larger learning rate or a smaller batch size.*

Novel Implicit Regularization Effects of SGD

Conjecture (Variance reduction effect of SGD)

Along the SGD trajectory, the maximum attained values of λ_H^1 and λ_K^1 are smaller for a larger learning rate or a smaller batch size.

Conjecture (Pre-conditioning effect of SGD)

Along the SGD trajectory, the maximum attained values of $\frac{\lambda_K^*}{\lambda_K^1}$ and $\frac{\lambda_H^*}{\lambda_H^1}$ are larger for a larger learning rate or a smaller batch size.

Both effects hold after a point we call the **break-even point**, and are desirable from the optimization perspective, and might help explain generalization of SGD.

Variance Reduction and Pre-Conditioning Effects

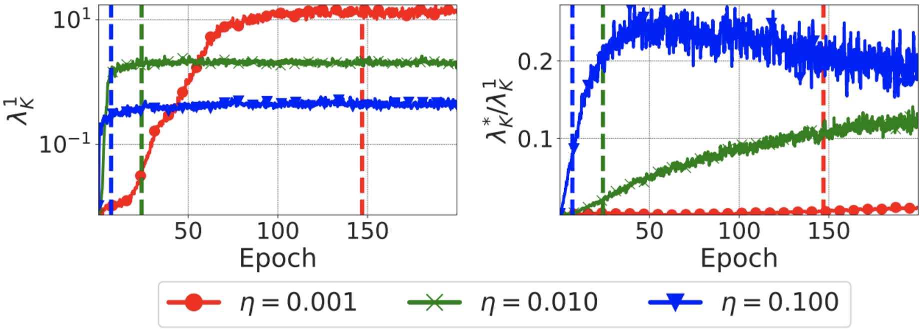
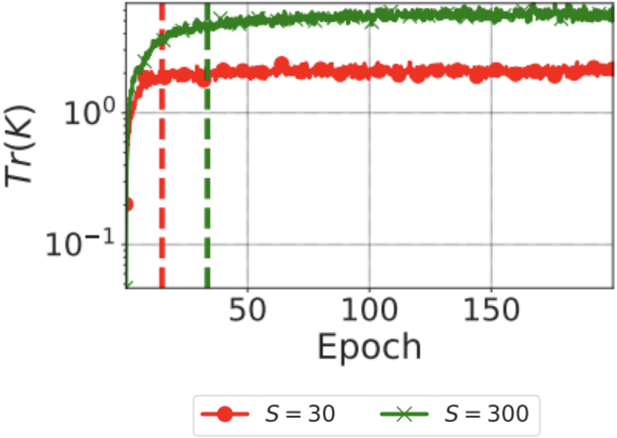
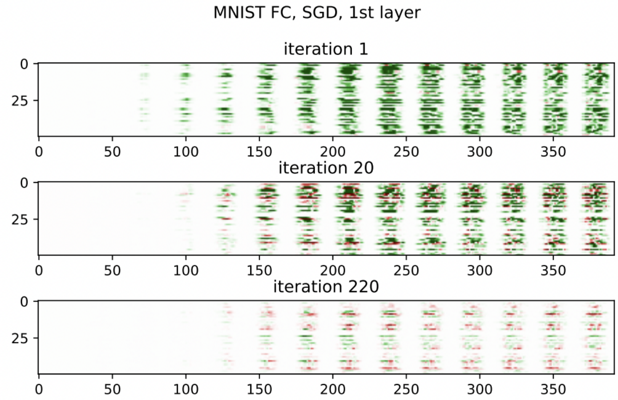


Figure: The variance reduction and the pre-conditioning effect of SGD, on ResNet-32.

Increasing Batch Size ⇒ Larger Variance of Gradients!



LCA Shows Training is Unstable



Lan et al. [2019]

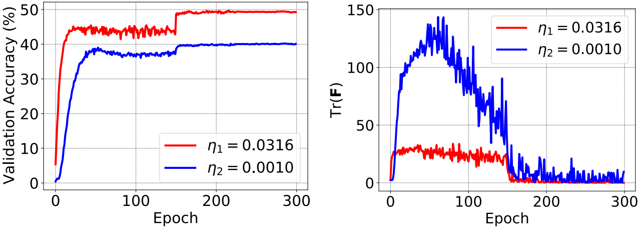
Summary

Optimization tends to steer towards increasingly sharp regions of the loss surface, which ultimately destabilizes optimization.

Selected implications:

- Large learning rate improves conditioning of the loss surface.
- Small batch size **reduces** the variance of gradients!

Catastrophic Fisher Explosion



Catastrophic Fisher Explosion: Early Phase Fisher Matrix Impacts Generalization,
Jastrzebski et al, ICML 2021

Hypothesis

Instability of the early phase of training is key for the mechanism behind implicit regularization effects in SGD.

How to Test Such a Hypothesis?

The Hessian can be approximated using the Fisher matrix. Let $g = \nabla_\theta \mathcal{L}(\mathbf{x}, y; \theta)$.

How to Test Such a Hypothesis?

The Hessian can be approximated using the Fisher matrix. Let $g = \nabla_\theta \mathcal{L}(\mathbf{x}, y; \theta)$.

$$\mathbf{H}(\theta) \approx \mathbf{F}(\theta) = \mathbb{E}_{x \sim \mathcal{X}, \hat{y} \sim p_\theta(y|x)}[g(x, \hat{y})^T g(x, \hat{y})]$$

How to Test Such a Hypothesis?

The Hessian can be approximated using the Fisher matrix. Let $g = \nabla_\theta \mathcal{L}(\mathbf{x}, y; \theta)$.

$$\mathbf{H}(\theta) \approx \mathbf{F}(\theta) = \mathbb{E}_{x \sim \mathcal{X}, \hat{y} \sim p_\theta(y|x)}[g(x, \hat{y})^T g(x, \hat{y})]$$

$$\mathrm{Tr}(\mathrm{H}) \approx \mathrm{Tr}(\mathbf{F}) = \mathrm{E} \ \|g\|^2|$$

Fisher Penalty

Notation: $(\mathbf{x}^b, y^b)$ - minibatch, θ, $\mathcal{L}(\mathbf{x}^b, y^b; \theta)$,

Fisher Penalty

Notation: $(\mathbf{x}^b, y^b)$ - minibatch, θ, $\mathcal{L}(\mathbf{x}^b, y^b; \theta)$, $\hat{y}^b$

Fisher Penalty

Notation: $(\mathbf{x}^b, y^b)$ - minibatch, θ, $\mathcal{L}(\mathbf{x}^b, y^b; \theta)$, $\hat{y}^b$

Definition (Fisher Penalty)

$$\mathcal{L}(\mathbf{x}^b, y^b; \theta) + \alpha \|\nabla_\theta \mathcal{L}(\mathbf{x}^b, \hat{y}^b; \theta)\|$$

Fisher Penalty

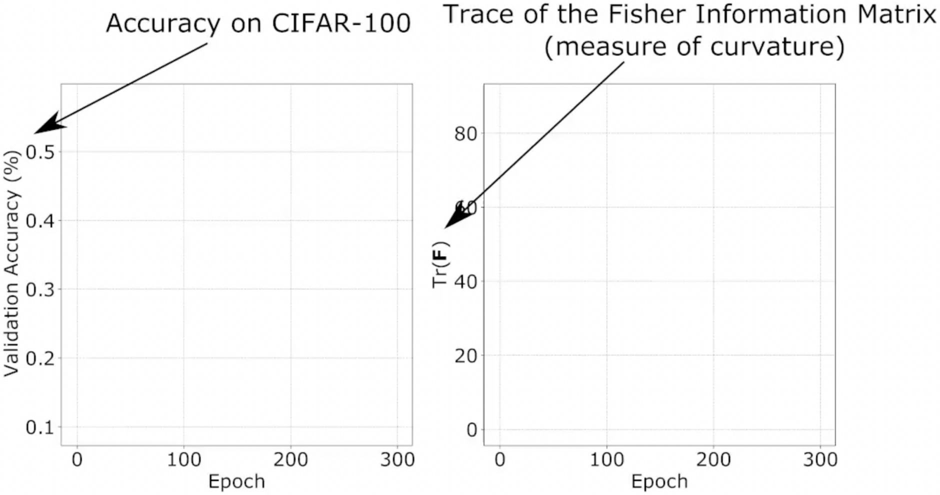
Notation: $(\mathbf{x}^b, y^b)$ - minibatch, θ, $\mathcal{L}(\mathbf{x}^b, y^b; \theta)$, $\hat{y}^b$

Definition (Fisher Penalty)

$$\boldsymbol{\mathcal{L}}(\mathbf{x}^b, y^b; \theta) + \alpha \|\nabla_\theta \boldsymbol{\mathcal{L}}(\mathbf{x}^b, \hat{y}^b; \theta)\|$$

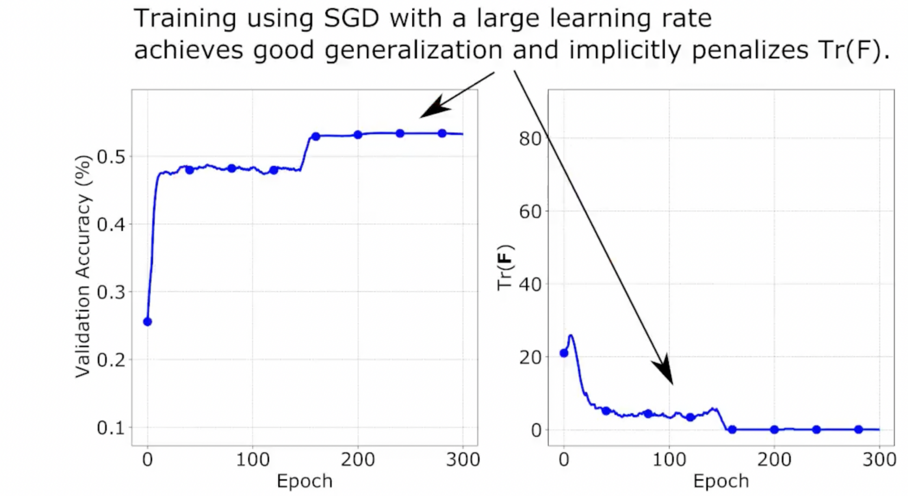
Possible to compute at $\approx$ 3x compute time using "double-backprop", or at $\approx$ 2x compute time using a finite difference approximation.

Catastrophic Fisher Explosion



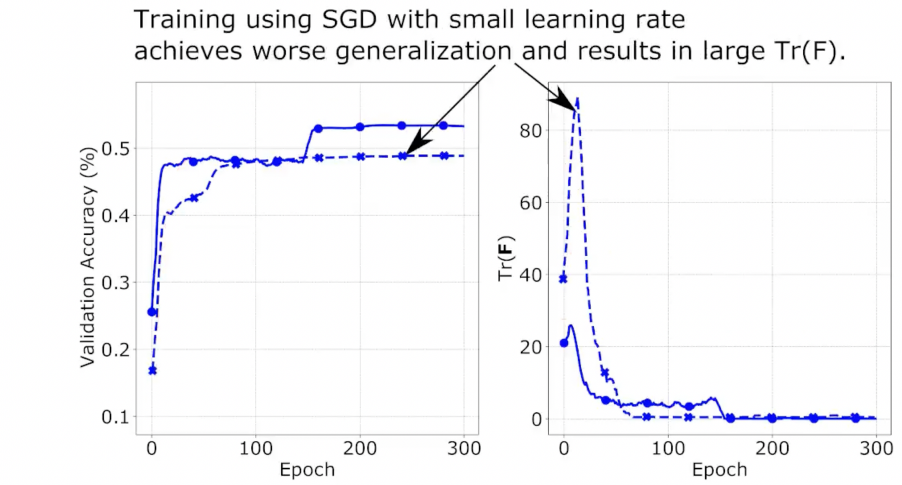
Training WideResNet on CIFAR-100.

Catastrophic Fisher Explosion



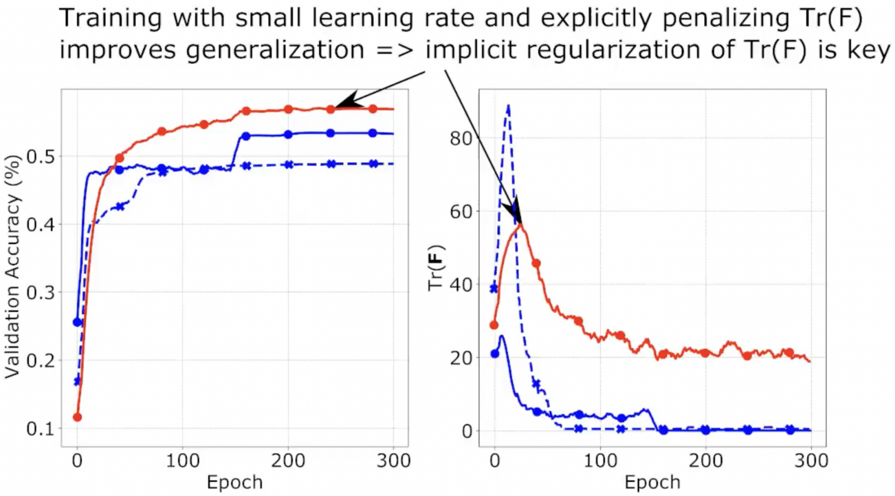
Training WideResNet on CIFAR-100.

Catastrophic Fisher Explosion



Training WideResNet on CIFAR-100.

Catastrophic Fisher Explosion



Training WideResNet on CIFAR-100.

Fisher Penalty Recovers Generalization Gap

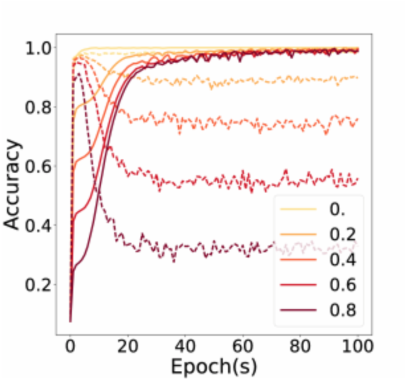
Setting	η^*	Baseline	GP_x	GP	FP	GP_r
TinyImageNet	54.67%	52.57%	52.79%	56.44%	**56.73%**	55.41%
CIFAR-100	66.09%	58.51%	62.12%	64.42%	**66.41%**	66.39%
CIFAR-100	45.86%	36.86%	45.26%	47.35%	**49.87%**	48.26%
CIFAR-100	53.96%	46.38%	**58.68%**	57.68%	57.05%	58.15%
CIFAR-10	76.94%	71.32%	75.68%	75.73%	79.66%	**79.76%**

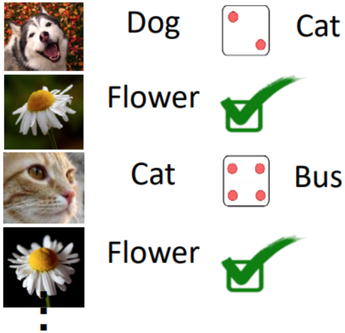
Table: Using a 10-30x smaller learning rate (Baseline) results in up to 9% degradation in test accuracy on popular image classification benchmarks. Adding FP closes the gap to η^*.

Why Does Fisher Penalty Help?

Hypothesis: Catastrophic Fisher explosion (large FIM in the early phase) promotes memorization instead of learning patterns in the dataset.

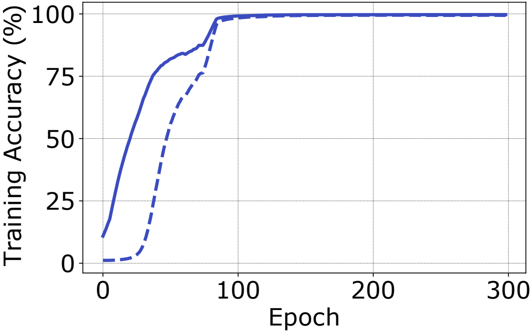
SGD is biased towards learning simple patterns

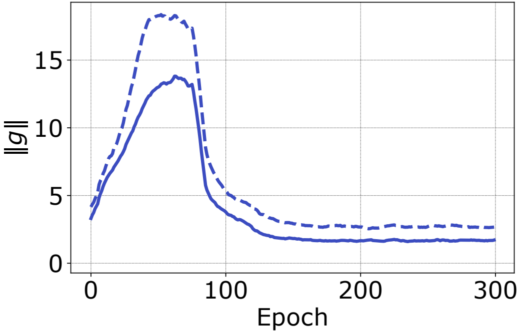




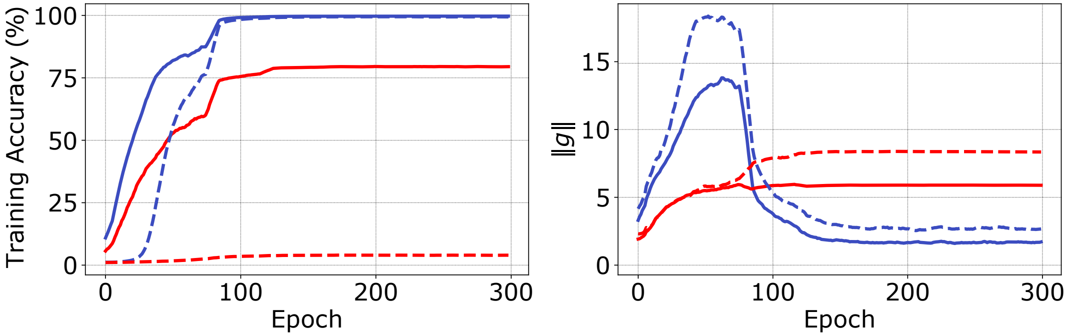
Arpit et al. [2017]

Fisher Penalty Disproportionally Slows Down Learning on Random Labels

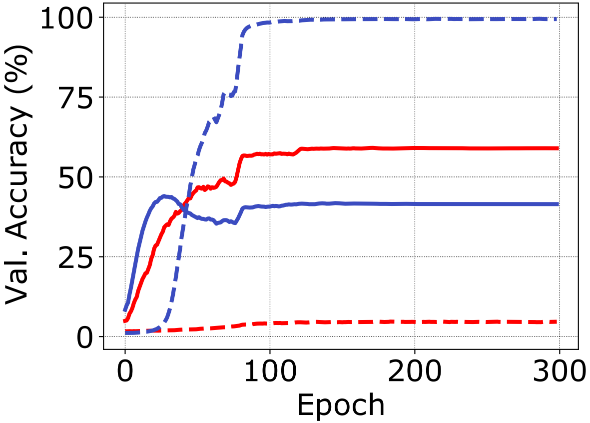




Fisher Penalty Disproportionally Slows Down Learning on Random Labels



Fisher Penalty Disproportionally Slows Down Learning on Random Labels



Fisher Penalty Disproportionally Slows Down Learning on Random Labels

Label Noise	Setting	Baseline	Mixup	GP_x	FP	GP_r
25%	CIFAR-100	41.74%	52.31%	45.94%	**60.18%**	58.46%
	CIFAR-100	53.30%	**61.61%**	52.70%	58.31%	57.60%
50%	CIFAR-100	30.05%	39.15%	34.26%	**51.33%**	50.33%
	CIFAR-100	43.35%	**51.71%**	42.99%	47.99%	50.08%

Related Work and Outlook

Related findings can be found in two works:

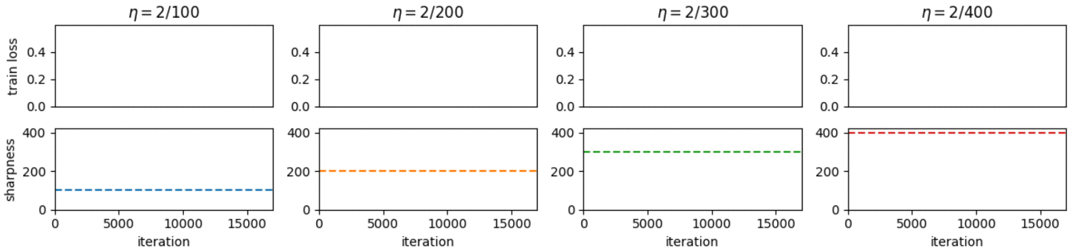
- Concurrent work titled *Sharpness Aware Minimization* Foret et al. [2021] , see also Smooth-Out, proposes an approximated penalty of the Hessian. Fisher Penalty is closely related. Our key contribution is proposing and corroborating a causal mechanism between changes in geometry and generalization. Our goal is not to propose an effective regularizer.

Related Work and Outlook

Related findings can be found in two works:

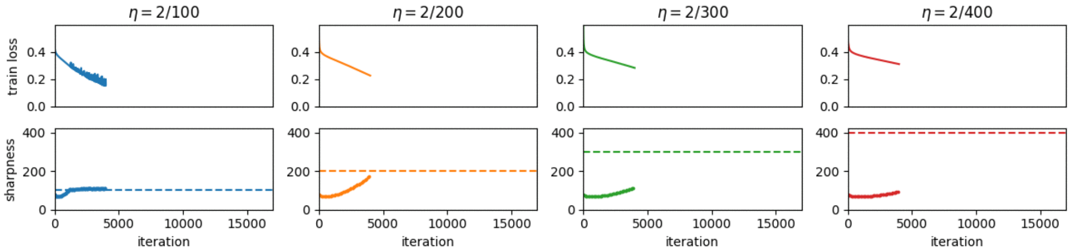
- Concurrent work titled *Sharpness Aware Minimization* Foret et al. [2021] , see also Smooth-Out, proposes an approximated penalty of the Hessian. Fisher Penalty is closely related. Our key contribution is proposing and corroborating a causal mechanism between changes in geometry and generalization. Our goal is not to propose an effective regularizer.
- *On the Origin of Implicit Regularization in Stochastic Gradient Descent* Smith et al. [2021] is most closely related. While the final explicit regularizer is similar, the proposed causal explanation is different and focuses on the instability in the early phase. Our empirical evaluation suggests Fisher Penalty is more effective than gradient norm penalty proposed in the work. However, more work is necessary to discern which causal explanation is more relevant for the success of deep neural networks.

GD on Neural Networks Typically Occurs at the Edge of Stability



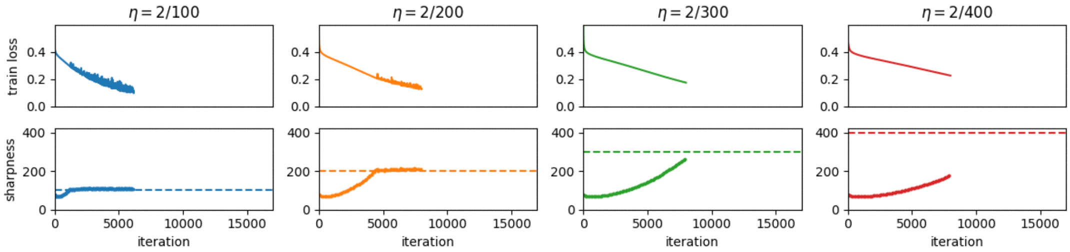
Cohen et al. [2021]

GD on Neural Networks Typically Occurs at the Edge of Stability



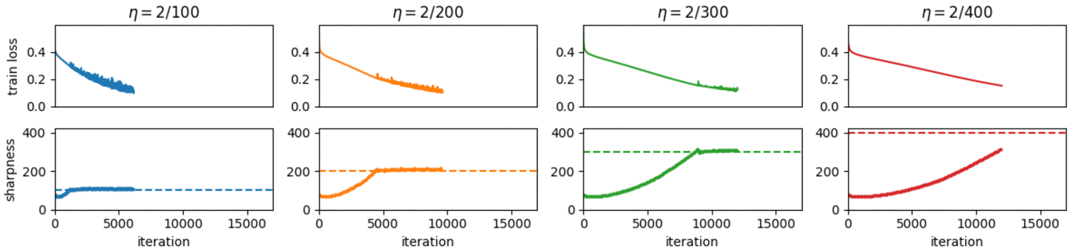
Cohen et al. [2021]

GD on Neural Networks Typically Occurs at the Edge of Stability



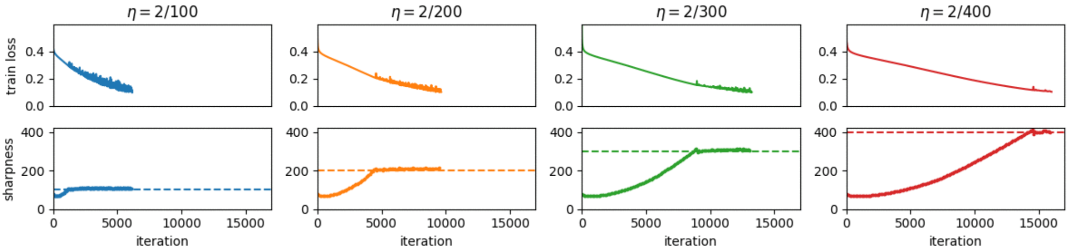
Cohen et al. [2021]

GD on Neural Networks Typically Occurs at the Edge of Stability



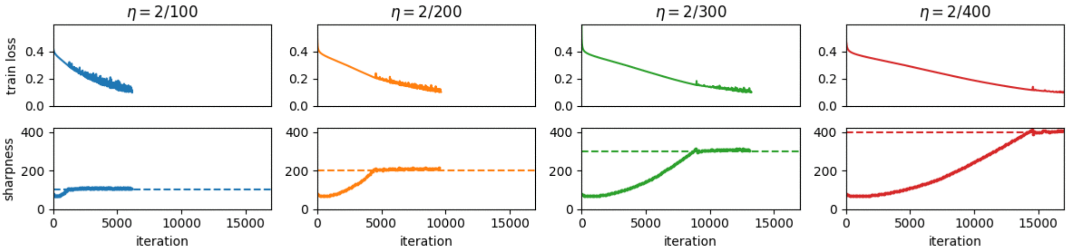
Cohen et al. [2021]

GD on Neural Networks Typically Occurs at the Edge of Stability



Cohen et al. [2021]

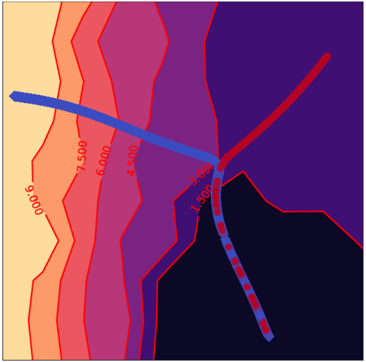
GD on Neural Networks Typically Occurs at the Edge of Stability



Cohen et al. [2021]

Summary

1. Key properties, such as conditioning, of the loss surface are regularized by SGD beyond the break-even point.



Summary

1. Key properties, such as conditioning, of the loss surface are regularized by SGD beyond the break-even point.

2. Instability of the early phase of training is key for the mechanism behind implicit regularization effects in SGD. We derive Fisher Penalty that simulates implicit regularization due to large η in SGD, and connect its effect to memorization.

Definition (Fisher Penalty)

$$\mathcal{L}(\mathbf{x}^b, y^b; \theta) + \alpha \|\nabla_\theta \mathcal{L}(\mathbf{x}^b, \hat{y}^b; \theta)\|$$

Fun Facts

If these don't sound absurd, you have understood the talk. If not, it is most likely my fault, and please ask questions :)

- Using large learning rates effectively acts as preconditioning of the loss surface past a certain point on the trajectory (break-even point).
- Small batch-size both increases and decreases the variance of gradients.
- The ability to avoid memorization by SGD is strongly modulated by the learning rate (but is mainly due to the early phase of training effects).

Thank you for your attention!

@kudkudakpl

Appendix: Optimization vs $\mathbf{K}$: A (Poor) Theoretical Argument

$$\mathbf{H}(\theta^*) \approx \mathbf{F}(\theta^*) \approx \mathbf{K}(\theta^*), \text{if}$$

- At the minimum (θ^*).
- The model is *well-specified*.
- The mean gradient is small compared to the variance of the gradient.

Bibliography I

Johan Bjorck, Carla P. Gomes, and Bart Selman. Understanding batch normalization. *CoRR*, abs/1806.02375, 2018. URL `http://arxiv.org/abs/1806.02375`.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos J. Storkey. Three factors influencing minima in SGD. *CoRR*, abs/1711.04623, 2017.

Ian J. Goodfellow, Oriol Vinyals, and Andrew M. Saxe. Qualitatively characterizing neural network optimization problems. *arXiv e-prints*, art. arXiv:1412.6544, Dec 2014.

Stanislaw Jastrzebski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the Relation Between the Sharpest Directions of DNN Loss and the SGD Step Length. *arXiv e-prints*, art. arXiv:1807.05031, Jul 2018.

Bibliography II

Janice Lan, Rosanne Liu, Hattie Zhou, and Jason Yosinski. Lca: Loss change allocation for neural network training. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/d77f00766fd3be3f2189c843a6af3fb2-Paper.pdf`.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks. *arXiv e-prints*, art. arXiv:1706.05394, Jun 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=6Tm1mposlrM`.

Bibliography III

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jh-rTtvkGeM.

J. Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. *ArXiv*, abs/2007.02561, 2020.