AdaFisher:

Adaptive Second Order Optimization via Fisher Information

Damien Martins Gomes[1,2], Yanlei Zhang[3], Eugene Belilovsky[1,3], Guy Wolf[3,4], Mahdi S. Hosseini[1,3]

[1]Concordia University, [2]IPSA Toulouse, [3]Quebec AI Institute – Mila, [4]Université de Montréal

ICLR 2025 – Singapore

Motivation

Machine Learning
- **Data**
- **Model**
- **Learning**

First Order Optimizers → Suboptimal convergence

Second Order Optimizers → High computation overhead

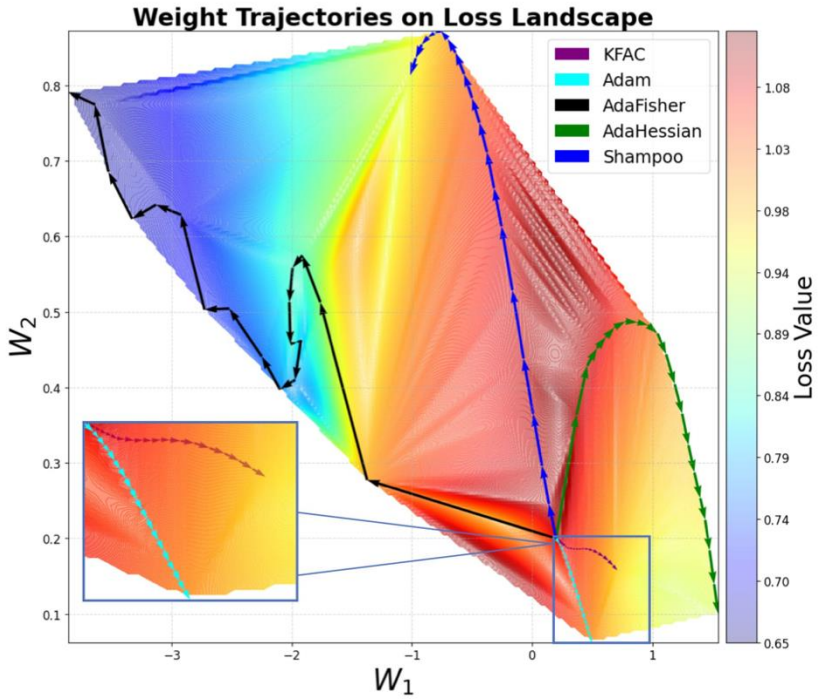


Figure 1: Visualizing optimization trajectories for various optimizers overlaid a loss landscape.

Can we design an optimizer that balances computational complexity (time and space) with strong generalization (fast convergence)?

AdaFisher

Methodology

Fisher Matrix Insights

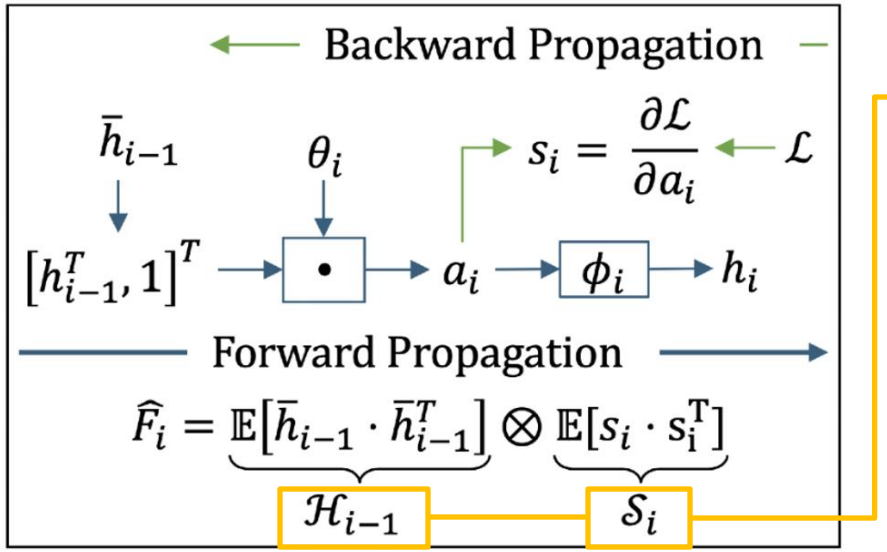


Figure 2: Illustration of EFIM computation using K-FAC [1] for a given layer *i*

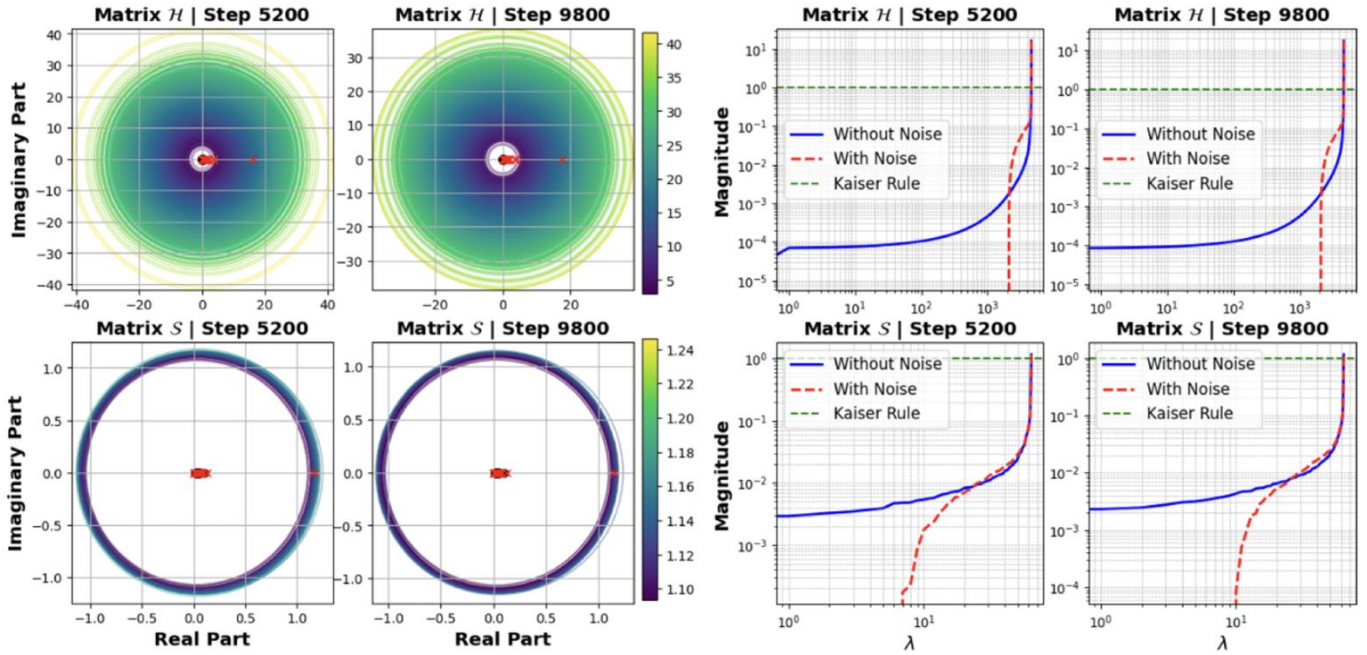


Figure 3: Gershgorin disks and eigenvalue perturbations in the 37th Convolutional layer of a ResNet-18 at steps 5200 (middle of training)
And 9800 (end of training). Left: Gershgorin circles; Right: Eigenvalue spectrum with/without noise.

New efficient approximation to the FIM

[1] Martens, James, and Roger Grosse. "Optimizing neural networks with kronecker-factored approximate curvature." *International conference on machine learning*. PMLR, 2015.

Methodology

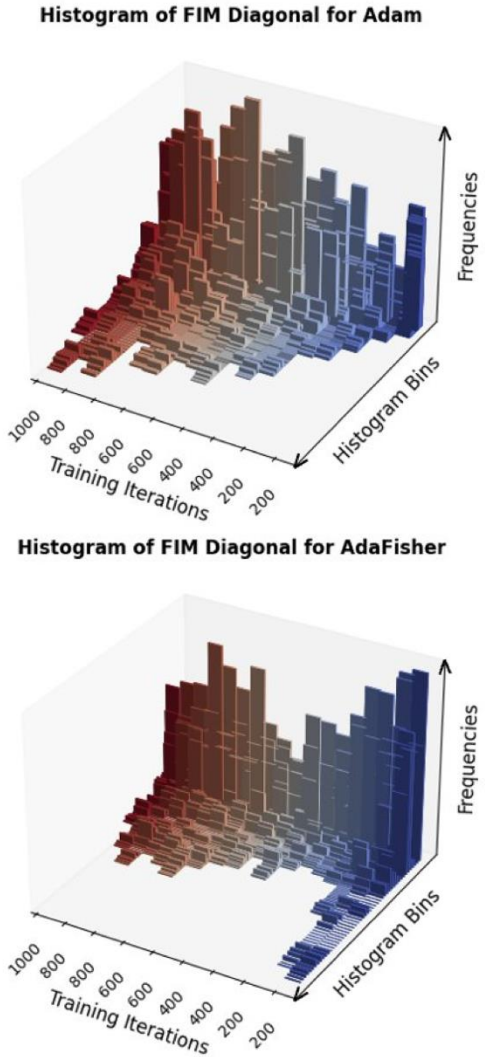


Figure 4: Comparison of FIM diagonal histograms during ResNet18 training on CIFAR10: The figure displays the FIM diagonal elements for the first convolutional layer with Adam and AdaFisher over 1,000 training iterations.

Better Generalization than Adam

Algorithm 1 AdaFisher optimization algorithm. Good default settings for the tested machine learning problems are $\alpha = 0.001$ (learning rate), $\lambda = 0.001$ (Tikhonov damping parameter),$\gamma = 0.8$ (Exponentially decaying factor). [Default parameters are: $\beta = 0.9$ (Exponentially decaying factor of Adam), κ (weight decay) (Kingma & Ba (2015), Loshchilov & Hutter (2019b))].

Require: Step size α; Exponential decay rate for KFs $\gamma \in [0, 1)$; Tikhonov damping parameter λ; Exponential decay rate for first moments β in $[0, 1)$; Initial parameters θ

Initialize 1st moment variable $m = 0$; FIM $\tilde{F}_{D_i} = \mathbf{I}$; time step $t = 0$

while stopping criterion not met **do**

  Sample a minibatch of M examples from the training set $\{(x_n, y_n)\}_{n=1}^{M}$

  Compute $\mathcal{H}_{D_{i-1}}$, $\mathcal{S}_{D_i}$ for $i \in \{1, \ldots, L\}$ using Section A.3 (notice that: $\mathcal{H}_{D_0} = x$)

  Compute EMAs of $\mathcal{H}_{D_{i-1}}$ and $\mathcal{S}_{D_i}$ using Eq. (3)

  Compute $\tilde{F}_{D_i}$ for $i \in \{1, \ldots, L\}$ using Eq. (4)

  $g^{(t)} \leftarrow \frac{1}{M} \sum_n \nabla_{\theta^{(t)}} \mathcal{L}(f(x_n; \theta^{(t)}), y_n)$ (Compute gradient)

  $m^{(t+1)} \leftarrow \frac{\beta m^{(t)} + (1-\beta) h^{(t)}}{1-\beta^t}$ (Update and correct biased first moment)

  Case AdaFisher: $\Delta\theta^{(t)} = -\alpha (\tilde{F}_D^{(t)})^{-1} m^{(t)}$

  Case AdaFisherW: $\Delta\theta^{(t)} = -\alpha \left((\tilde{F}_D^{(t)})^{-1} m^{(t)} + \kappa \theta^{(t)} \right)$

  $\theta^{(t+1)} \leftarrow \theta^{(t)} + \Delta\theta^{(t)}$ (Apply update)

  $t \leftarrow t + 1$

end while

Results

Image Classification

Table 2: Validation of ImageNet-1K / ResNet50 by different optimizers reported on Top-1 and Top-5 accuracy.

Optimizers	Batch size	Top-1	Top-5
Adam	256	67.78	88.37
K-FAC	256	70.96	89.44
Shampoo	256	72.82	91.42
AdaFisher	256	**76.95**	**93.39**
AdaFisher	512	**77.01**	**93.45**
AdaFisher	1024	**77.09**	**93.56**
SGD Goyal et al. (2017)	256	76.40	-
AdamW Chen et al. (2024)	1024	76.34	-
LAMB You et al. (2019)	$16K$	76.66	93.22
SGD You et al. (2019)	$16K$	75.20	-
LARS Huo et al. (2021)	$16K$	75.1	-

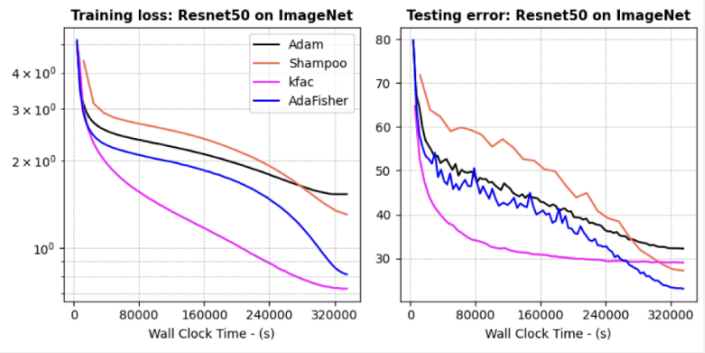


Figure 5: Training loss and validation error of ResNet-50 on ImageNet-1k. AdaFisher consistently achieves lower test error as compared to its counterparts.

Table 3: Performance metrics (mean, std) of different networks and optimizers on CIFAR10 and CIFAR100 using batch size 256 with a 200-epoch AdaFisher training cutoff

Network	CIFAR10						CIFAR100					
	SGD	Adam	AdaHessian	K-FAC	Shampoo	AdaFisher	SGD	Adam	AdaHessian	K-FAC	Shampoo	AdaFisher
ResNet18	$95.64_{0.1}$	$94.85_{0.1}$	$95.44_{0.1}$	$95.17_{0.2}$	$94.08_{0.2}$	$\mathbf{96.25_{0.2}}$	$76.56_{0.2}$	$75.74_{0.1}$	$71.79_{0.2}$	$76.03_{0.3}$	$76.78_{0.2}$	$\mathbf{77.28_{0.2}}$
ResNet50	$95.71_{0.1}$	$94.45_{0.2}$	$95.54_{0.1}$	$95.66_{0.1}$	$94.59_{0.1}$	$\mathbf{96.34_{0.2}}$	$78.01_{0.1}$	$74.65_{0.5}$	$75.81_{0.3}$	$77.40_{0.4}$	$78.07_{0.4}$	$\mathbf{79.77_{0.4}}$
ResNet101	$95.98_{0.2}$	$94.57_{0.1}$	$95.29_{0.6}$	$96.01_{0.1}$	$94.63_{0.1}$	$\mathbf{96.39_{0.1}}$	$78.89_{0.2}$	$75.56_{0.3}$	$73.38_{0.2}$	$77.01_{0.4}$	$78.83_{0.2}$	$\mathbf{80.65_{0.4}}$
DenseNet121	$96.09_{0.1}$	$94.86_{0.1}$	$96.11_{0.1}$	$96.12_{0.1}$	$95.66_{0.1}$	$\mathbf{96.72_{0.1}}$	$80.13_{0.4}$	$75.87_{0.4}$	$74.80_{0.9}$	$79.79_{0.2}$	$80.24_{0.3}$	$\mathbf{81.36_{0.3}}$
MobileNetV3	$94.43_{0.2}$	$93.32_{0.1}$	$92.86_{3.1}$	$94.34_{0.1}$	$93.81_{0.2}$	$\mathbf{95.28_{0.1}}$	$73.89_{0.3}$	$70.62_{0.3}$	$56.58_{4.5}$	$73.75_{0.3}$	$70.85_{0.3}$	$\mathbf{77.56_{0.1}}$
Tiny Swin	$82.34_{0.2}$	$87.37_{0.6}$	$84.15_{0.2}$	$64.79_{0.5}$	$63.91_{0.4}$	$\mathbf{88.74_{0.4}}$	$54.89_{0.4}$	$60.21_{0.4}$	$56.86_{0.5}$	$34.45_{0.4}$	$30.39_{1.2}$	$\mathbf{66.05_{0.5}}$
FocalNet	$82.03_{0.2}$	$86.23_{0.1}$	$64.18_{0.2}$	$38.94_{0.8}$	$37.96_{0.7}$	$\mathbf{87.90_{0.1}}$	47.76_{03}	$52.71_{0.5}$	$32.33_{0.3}$	$9.98_{0.6}$	$9.18_{0.1}$	$\mathbf{53.69_{0.3}}$
CCT-2/3×2	$78.76_{0.3}$	$83.89_{0.4}$	–	$33.08_{2.3}$	$35.16_{0.4}$	$\mathbf{84.94_{0.3}}$	$54.05_{0.4}$	$59.78_{0.5}$	–	$7.17_{0.2}$	$8.60_{0.1}$	$\mathbf{62.91_{0.5}}$

*Note that Adam and AdaFisher were used for all CNN architectures, while AdamW and AdaFisherW were applied for all ViT experiments.

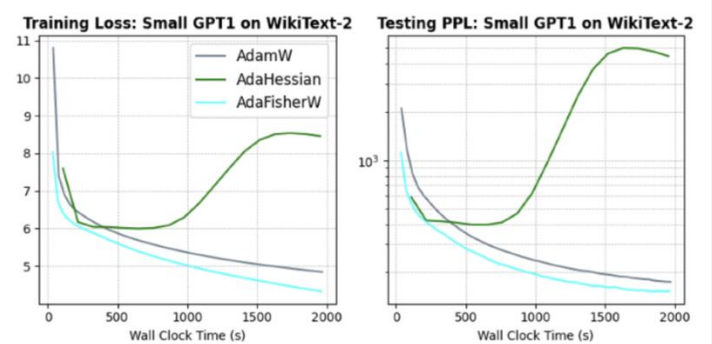
Table 4: Performance comparison of different networks and optimizers on CIFAR10 and CIFAR100 using ImageNet-1K pretrained weights. Evaluation is based on wall clock time of 50 training epochs with AdaFisher.

Network	CIFAR10						CIFAR100					
	SGD	Adam	AdaHessian	K-FAC	Shampoo	AdaFisher	SGD	Adam	AdaHessian	K-FAC	Shampoo	AdaFisher
ResNet50	$96.50_{0.2}$	$96.45_{0.2}$	$96.35_{0.3}$	$96.45_{0.1}$	$96.03_{0.4}$	$\mathbf{97.13_{0.2}}$	$82.12_{0.1}$	$82.01_{0.4}$	$80.64_{0.9}$	$80.55_{0.4}$	$81.70_{0.2}$	$\mathbf{82.23_{0.2}}$
ResNet101	$97.07_{0.2}$	$96.70_{0.1}$	$96.65_{0.2}$	$96.84_{0.1}$	$96.63_{0.1}$	$\mathbf{97.22_{0.1}}$	$84.01_{0.1}$	$82.43_{0.2}$	$81.36_{0.8}$	$82.26_{0.3}$	$82.65_{0.2}$	$\mathbf{84.47_{0.2}}$
DenseNet121	$94.80_{0.1}$	$94.77_{0.1}$	$93.08_{0.1}$	$94.41_{0.2}$	$94.76_{0.1}$	$\mathbf{95.03_{0.1}}$	$75.98_{0.2}$	$75.65_{0.3}$	$71.06_{0.9}$	$76.10_{0.3}$	$76.08_{0.2}$	$\mathbf{76.92_{0.3}}$
MobileNetV3	$91.76_{0.3}$	$90.92_{0.3}$	$86.45_{2.5}$	$91.72_{0.2}$	$91.39_{0.3}$	$\mathbf{92.78_{0.2}}$	$71.86_{0.4}$	$66.11_{0.8}$	$59.69_{2.3}$	$69.85_{0.4}$	$68.87_{0.3}$	$\mathbf{72.38_{0.4}}$

Language Modeling

Table 5: Language Modeling performance (PPL) on Wikitest-2 and PTB test dataset (lower is better)

Optimizer	Test PPL	
	WikiText-2	PTB
AdamW	175.06	44.70
AdaHessian	407.69	59.43
Shampoo	1727.75	–
AdaFisherW	**152.72**	**41.15**



Stability Analysis

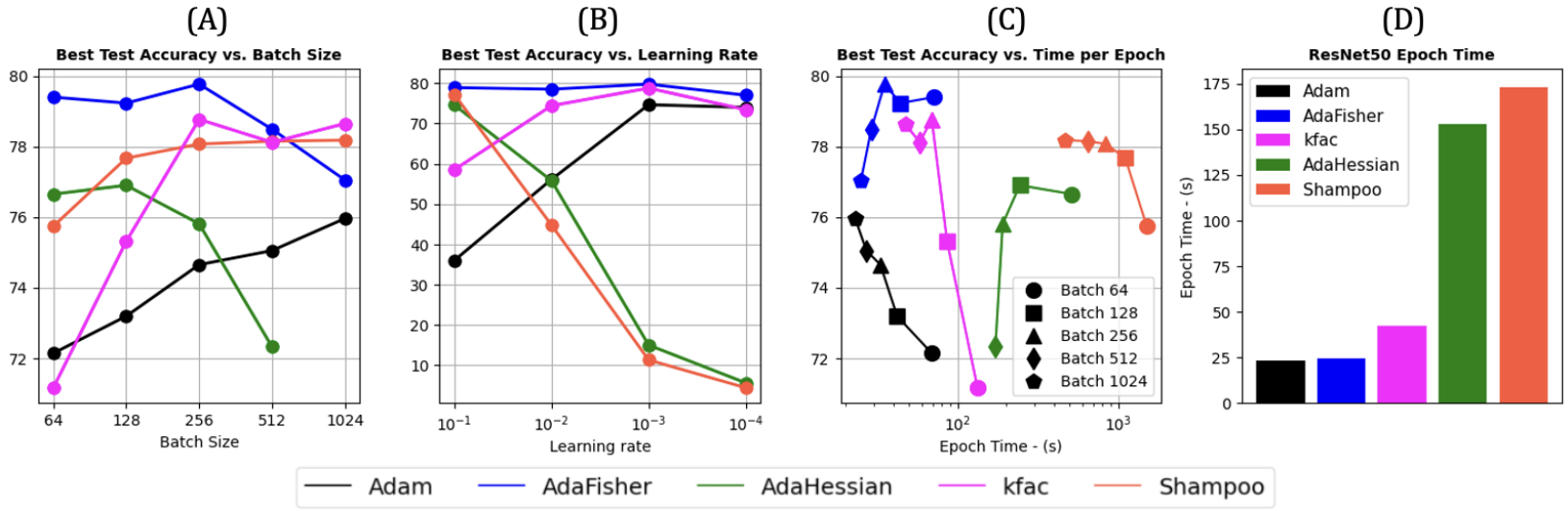


Figure 6: Performance comparison of AdaFisher and other optimizers using the ResNet50 network on the CIFAR100 dataset. (A) Test accuracy by batch size. (B) Accuracy vs. learning rates. (C) Accuracy related to epoch time across batch sizes. (D) Epoch time for different optimizers with a batch size of 256.

Code

Paper

AdaFisher: Adaptive Second Order Optimization vis Fisher

- ✓ Convergence, Generalization Capabilities
- ✓ Computational Efficiency