Descending through a Crowded Valley – Benchmarking Deep Learning Optimizers

Robin M. Schmidt, Frank Schneider, Philipp Hennig

ICML 2021

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

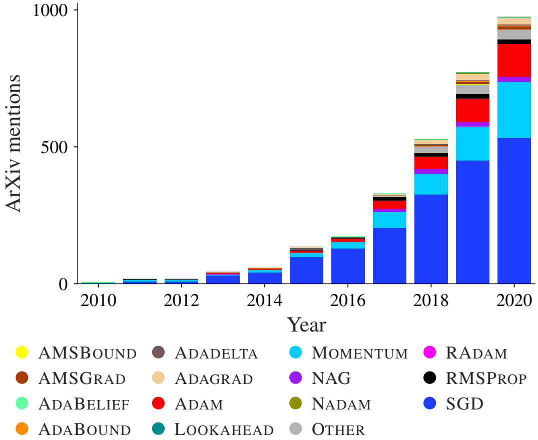
imprs-is

erc
some of the presented work is supported by the European Research Council.

Optimization for Deep Learning

A crowded valley of methods

EBERHARD KARLS UNIVERSITÄT TÜBINGEN



Name	Ref.	Name	Ref.
AcceleGrad	(Levy et al., 2018)	HyperAdam	(Wang et al., 2019b)
ACClip	(Zhang et al., 2020)	K-BFGS/K-BFGS(L)	(Goldfarb et al., 2020)
AdaAlter	(Xie et al., 2019)	KF-QN-CNN	(Ren & Goldfarb, 2021)
AdaBatch	(Devarakonda et al., 2017)	KFAC	(Martens & Grosse, 2015)
AdaBayes/AdaBayes-SS	(Aitchison, 2020)	KFLR/KFRA	(Botev et al., 2017)
AdaBelief	(Zhuang et al., 2020)	L4Adam/L4Momentum	(Rolínek & Martius, 2018)
AdaBlock	(Yun et al., 2019)	LAMB	(You et al., 2020)
AdaBound	(Luo et al., 2019)	LaProp	(Ziyin et al., 2020)
AdaComp	(Chen et al., 2018)	LARS	(You et al., 2017)
Adadelta	(Zeiler, 2012)	LHOPT	(Almeida et al., 2021)
Adafactor	(Shazeer & Stern, 2018)	LookAhead	(Zhang et al., 2019)
AdaFix	(Bae et al., 2019)	M-SVAG	(Balles & Hennig, 2018)
AdaFom	(Chen et al., 2019a)	MADGRAD	(Defazio & Jelassi, 2021)
AdaFTRL	(Orabona & Pál, 2015)	MAS	(Landro et al., 2020)
Adagrad	(Duchi et al., 2011)	MEKA	(Chen et al., 2020b)
ADAHESSIAN	(Yao et al., 2020)	MTAdam	(Malkiel & Wolf, 2020)
Adai	(Xie et al., 2020)	MVRC-1/MVRC-2	(Chen & Zhou, 2020)
AdaLoss	(Teixeira et al., 2019)	Nadam	(Dozat, 2016)
Adam	(Kingma & Ba, 2015)	NAMSB/NAMSG	(Chen et al., 2019b)
Adam+	(Liu et al., 2020b)	ND-Adam	(Zhang et al., 2017a)
AdamAL	(Tao et al., 2019)	Nero	(Liu et al., 2021b)
AdaMax	(Kingma & Ba, 2015)	Nesterov	(Nesterov, 1983)
AdamBS	(Liu et al., 2020c)	Noisy Adam/Noisy K-FAC	(Zhang et al., 2018)
AdamNC	(Reddi et al., 2018)	NosAdam	(Huang et al., 2019)
AdaMod	(Ding et al., 2019)	Novograd	(Ginsburg et al., 2019)
AdamP/SGDP	(Heo et al., 2021)	NT-SGD	(Zhou et al., 2021b)
AdamT	(Zhou et al., 2020)	Padam	(Chen et al., 2020a)
AdamW	(Loshchilov & Hutter, 2019)	PAGE	(Li et al., 2020b)
AdamX	(Tran & Phong, 2019)	PAL	(Mutschler & Zell, 2020)
ADAS	(Eliyahu, 2020)	PolyAdam	(Orvieto et al., 2019)
AdaS	(Hosseini & Plataniotis, 2020)	Polyak	(Polyak, 1964)
AdaScale	(Johnson et al., 2020)	PowerSGD/PowerSGDM	(Vogels et al., 2019)
AdaSGD	(Wang & Wiens, 2020)	Probabilistic Polyak	(de Roos et al., 2021)
AdaShift	(Zhou et al., 2019)	ProbLS	(Mahsereci & Hennig, 2017)
AdaSqrt	(Hu et al., 2019)	PStorm	(Xu, 2020)
Adathm	(Sun et al., 2019)	QHAdam/QHM	(Ma & Yarats, 2019)
AdaX/AdaX-W	(Li et al., 2020a)	RAdam	(Liu et al., 2020a)
AEGD	(Liu & Tian, 2020)	Ranger	(Wright, 2020b)
ALI-G	(Berrada et al., 2020)	RangerLars	(Grankin, 2020)
AMSBound	(Luo et al., 2019)	RMSProp	(Tieleman & Hinton, 2012)
AMSGrad	(Reddi et al., 2018)	RMSterov	(Choi et al., 2019)

⋮

Benchmark Setup

The many dimensions to explore

$$
\underset{\textbf{Problem}}{\left\{\begin{matrix}\text{P1}\\ \text{P2}\\ \dots\\ \text{P8}\end{matrix}\right\}_{8}} \times \underset{\textbf{Optimizer}}{\left\{\begin{matrix}\text{Adam}\\ \text{NAG}\\ \dots\\ \text{SGD}\end{matrix}\right\}_{15}} \times \underset{\textbf{Tuning}}{\left\{\begin{matrix}\text{one-shot}\\ \text{small}\\ \text{medium}\\ \text{large}\end{matrix}\right\}_{4}} \times \underset{\textbf{Schedule}}{\left\{\begin{matrix}\text{constant}\\ \text{cosine}\\ \text{cosine wr}\\ \text{trapez.}\end{matrix}\right\}_{4}}
$$

Benchmark Setup

The many dimensions to explore

$$\underset{\textbf{Problem}}{\begin{Bmatrix} \text{P1} \\ \text{P2} \\ \dots \\ \text{P8} \end{Bmatrix}_8} \times \underset{\textbf{Optimizer}}{\begin{Bmatrix} \text{ADAM} \\ \text{NAG} \\ \dots \\ \text{SGD} \end{Bmatrix}_{15}} \times \underset{\textbf{Tuning}}{\begin{Bmatrix} \text{one-shot} \\ \text{small} \\ \text{medium} \\ \text{large} \end{Bmatrix}_4} \times \underset{\textbf{Schedule}}{\begin{Bmatrix} \text{constant} \\ \text{cosine} \\ \text{cosine wr} \\ \text{trapez.} \end{Bmatrix}_4}$$

AMSBound	AdaGrad	NAG
AMSGrad	Adam	NAdam
AdaBelief	LA(Mom.)	RAdam
AdaBound	LA(RAdam)	RMSProp
AdaDelta	Momentum	SGD

Benchmark Setup

The many dimensions to explore

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\underset{\textbf{Problem}}{\left\{\begin{matrix}\text{P1}\\ \text{P2}\\ \dots\\ \text{P8}\end{matrix}\right\}_8} \times \underset{\textbf{Optimizer}}{\left\{\begin{matrix}\text{ADAM}\\ \text{NAG}\\ \dots\\ \text{SGD}\end{matrix}\right\}_{15}} \times \underset{\textbf{Tuning}}{\left\{\begin{matrix}\text{one-shot}\\ \text{small}\\ \text{medium}\\ \text{large}\end{matrix}\right\}_4} \times \underset{\textbf{Schedule}}{\left\{\begin{matrix}\text{constant}\\ \text{cosine}\\ \text{cosine wr}\\ \text{trapez.}\end{matrix}\right\}_4}$$

	Data set	**Model**	**Task**	**Approx. run time**
P1	Artificial	Noisy quadratic	Minimization	$<$ 1 min
P2	MNIST	VAE	Generative	10 min
P3	Fashion-MNIST	Simple CNN: *2c2d*	Classification	20 min
P4	CIFAR-10	Simple CNN: *3c3d*	Classification	35 min
P5	Fashion-MNIST	VAE	Generative	20 min
P6	CIFAR-100	*All-CNN-C*	Classification	4 h 00 min
P7	SVHN	*Wide ResNet 16-4*	Classification	3 h 30 min
P8	War and Peace	RNN	Character Prediction	5 h 30 min

Benchmark Setup

The many dimensions to explore

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

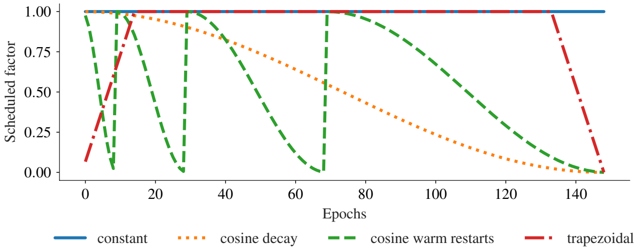
$$\underset{\textbf{Problem}}{\left\{\begin{matrix}\text{P1}\\\text{P2}\\\dots\\\text{P8}\end{matrix}\right\}_8} \times \underset{\textbf{Optimizer}}{\left\{\begin{matrix}\text{Adam}\\\text{NAG}\\\dots\\\text{SGD}\end{matrix}\right\}_{15}} \times \underset{\textbf{Tuning}}{\left\{\begin{matrix}\text{one-shot}\\\text{small}\\\text{medium}\\\text{large}\end{matrix}\right\}_4} \times \underset{\textbf{Schedule}}{\left\{\begin{matrix}\text{constant}\\\text{cosine}\\\text{cosine wr}\\\text{trapez.}\end{matrix}\right\}_4}$$

- **One-Shot** - 1 Run
  No tuning, uses default hyperparameters
- **Small** - 25 Runs
  Tuned via random search
- **Medium** - 50 Runs
  Tuned via random search, superset of *small budget*
- **Large** - 75 Runs
  Tuned via random search, refined search spaces

Benchmark Setup

The many dimensions to explore

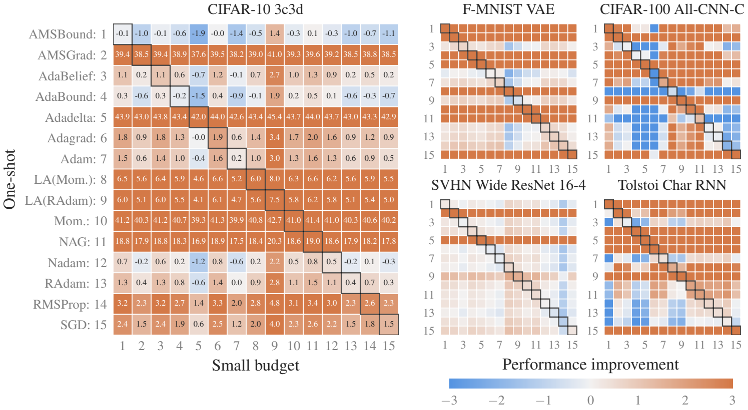
$$\underset{\textbf{Problem}}{\begin{Bmatrix} \text{P1} \\ \text{P2} \\ \dots \\ \text{P8} \end{Bmatrix}_8} \times \underset{\textbf{Optimizer}}{\begin{Bmatrix} \text{Adam} \\ \text{NAG} \\ \dots \\ \text{SGD} \end{Bmatrix}_{15}} \times \underset{\textbf{Tuning}}{\begin{Bmatrix} \text{one-shot} \\ \text{small} \\ \text{medium} \\ \text{large} \end{Bmatrix}_4} \times \underset{\textbf{Schedule}}{\begin{Bmatrix} \text{constant} \\ \text{cosine} \\ \text{cosine wr} \\ \text{trapez.} \end{Bmatrix}_4}$$



Results: Out-of-the-box performance

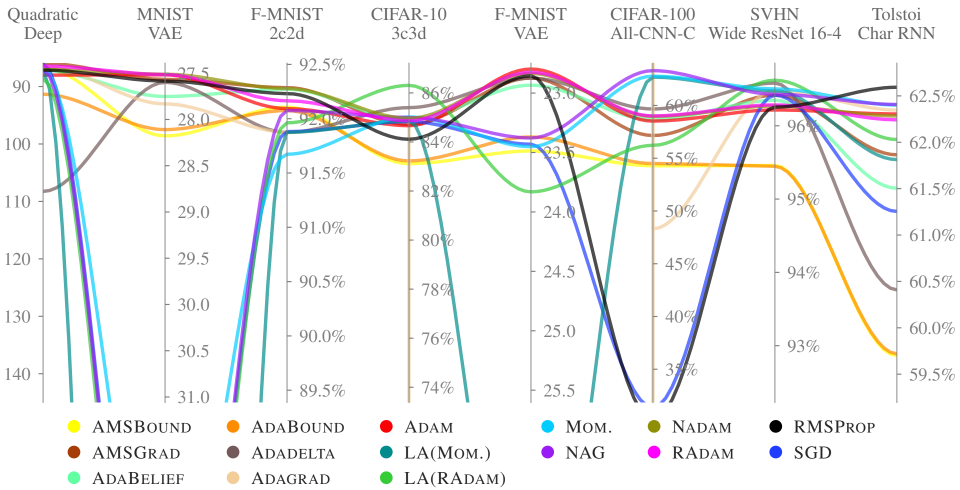
Why trying out optimizers can be better than tuning them

- **Orange rows**
  bad default hyperparameters
  SGD, NAG, MOMENTUM, AMSGRAD, ADADELTA
- **White & blue rows**
  good default hyperparameters
  ADAM, NADAM, RADAM, AMSBOUND, ADABOUND



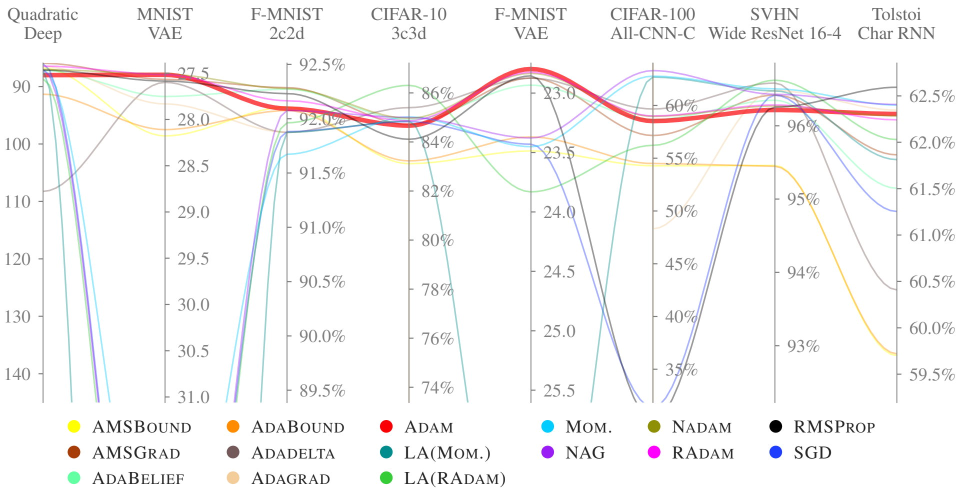
Results: Which optimizer to pick?

Why ADAM is still a good choice



Results: Which optimizer to pick?

Why ADAM is still a good choice



Summary

Descending through a Crowded Valley — Benchmarking Deep Learning Optimizers

Robin M. Schmidt, Frank Schneider, Philipp Hennig

- No method significantly and consistently outperforms the competition.
- Adam remains a viable choice that often ranks near the top.
- Trying out different optimizers helps about as much as tuning the parameters of one specific method.

Paper	arXiv	`2007.01547`
Results		`https://github.com/SirRob1997/Crowded-Valley---Results`
Framework	DeepOBS	`https://github.com/fsschneider/DeepOBS`