How to Train Better: Exploiting the Separability of Deep Neural Networks

Lars Ruthotto Elizabeth Newman

Emory University

Dynamics and Discretization: PDEs, Sampling, and Optimization
Simons Institute
October 29, 2021

Funding Acknowledgements:

NSF DMS 1751636 FA9550-20-1-0372 ASCR 20-023231

Collaborators for This Talk

Train Like a (Var)Pro

Joseph Hart

Bart van Bloeman Waanders

Julianne Chung

Matthias Chung

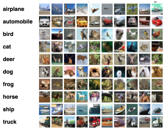
slimTrain

Train Like a (Var)Pro: Efficient Training of Neural Networks with Variable Projection
To appear in SIMODS. arXiv:2007.13171.
Code on Meganet.m.

slimTrain – **A Stochastic Approximation Method for Training Separable Deep Neural Networks**
Submitted to SISC. arXiv:2109.14002.
Code on Meganet.m and slimTrain.

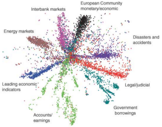
Deep Neural Networks are Great, But...

Classification



(Krizhevsky 2009)

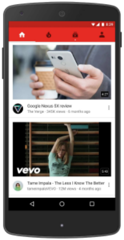
Autoencoders



(Hinton and Salakhutdinov 2006)

GANS

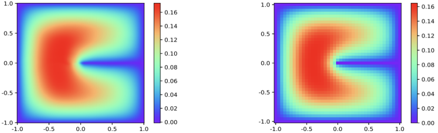
(Goodfellow et al. 2014)



Recommender Systems

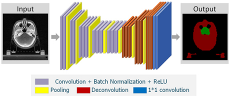
(Covington, Adams, and Sargin 2016)

Solving High-Dimensional PDEs



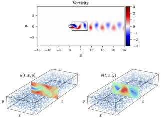
(E and Yu 2018; Han, Jentzen, and E 2018)

Segmentation



(Men et al. 2017)

PINNs



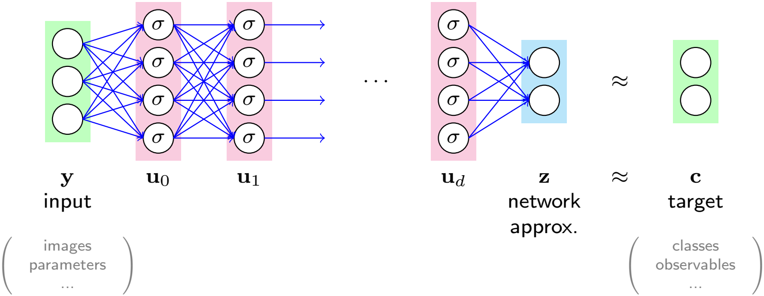
(Raissi, Perdikaris, and Karniadakis 2019)

Deep Neural Networks are Great, But...

Step 1
deterministic setting
exploit structure
optimize well

Step 2
stochastic setting
learn well
automate tuning

Separable Deep Neural Networks



Goal: find weights $(\mathbf{W}, \boldsymbol{\theta})$ such that

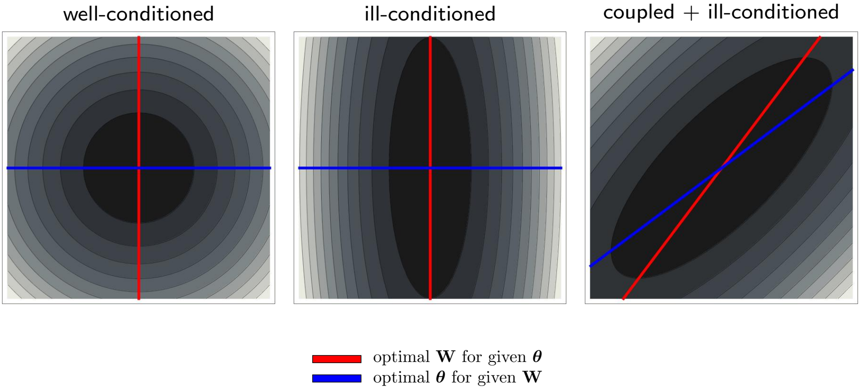
$$\mathbf{W} F(\mathbf{y}, \boldsymbol{\theta}) \approx \mathbf{c}$$

for all input-target pairs $(\mathbf{y}, \mathbf{c})$ by solving

$$\min_{\mathbf{W}, \boldsymbol{\theta}} \Phi(\mathbf{W}, \boldsymbol{\theta}) \equiv \underbrace{\mathbb{E}\ L(\mathbf{W} F(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c})}_{\text{loss}} + \underbrace{R(\boldsymbol{\theta}) + S(\mathbf{W})}_{\text{regularization}}$$

A Couple of Notes on Coupling

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W},\boldsymbol{\theta})$$



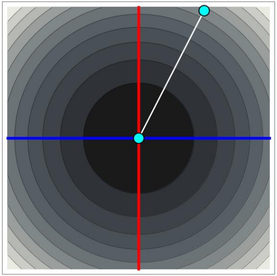
A Couple of Notes on Coupling

well-conditioned

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W}, \boldsymbol{\theta})$$

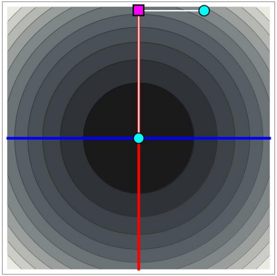
half step
full step

gradient descent



$$(\mathbf{W}, \boldsymbol{\theta}) \leftarrow (\mathbf{W}, \boldsymbol{\theta}) - \gamma \nabla \Phi$$

alternating directions



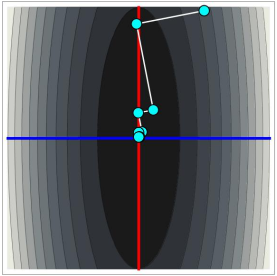
$$\mathbf{W} \leftarrow \arg\min_{\mathbf{W}} \Phi(\mathbf{W}, \boldsymbol{\theta})$$

$$\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}} \Phi(\mathbf{W}, \boldsymbol{\theta})$$

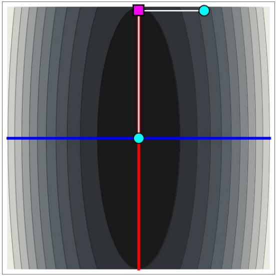
A Couple of Notes on Coupling

ill-conditioned

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W},\boldsymbol{\theta})$$



$$(\mathbf{W},\boldsymbol{\theta}) \leftarrow (\mathbf{W},\boldsymbol{\theta}) - \gamma \nabla \Phi$$



$$\mathbf{W} \leftarrow \arg\min_{\mathbf{W}} \Phi(\mathbf{W},\boldsymbol{\theta})$$

$$\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}} \Phi(\mathbf{W},\boldsymbol{\theta})$$

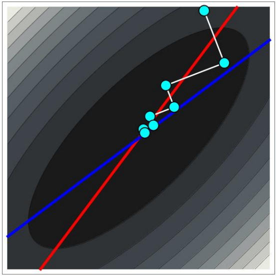
A Couple of Notes on Coupling

coupled + ill-conditioned

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W},\boldsymbol{\theta})$$

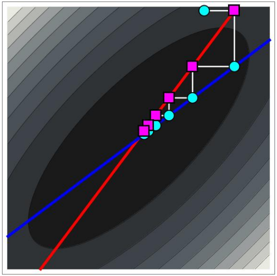
half step
full step

gradient descent



$$(\mathbf{W},\boldsymbol{\theta}) \leftarrow (\mathbf{W},\boldsymbol{\theta}) - \gamma\nabla\Phi$$

alternating directions



$$\mathbf{W} \leftarrow \arg\min_{\mathbf{W}} \Phi(\mathbf{W},\boldsymbol{\theta})$$

$$\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}} \Phi(\mathbf{W},\boldsymbol{\theta})$$

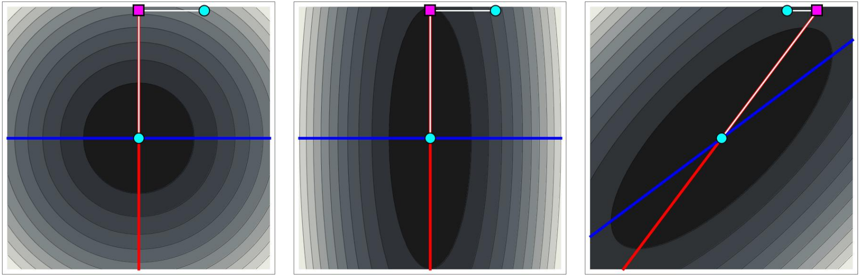
A Couple of Notes on Coupling

updating
with coupling

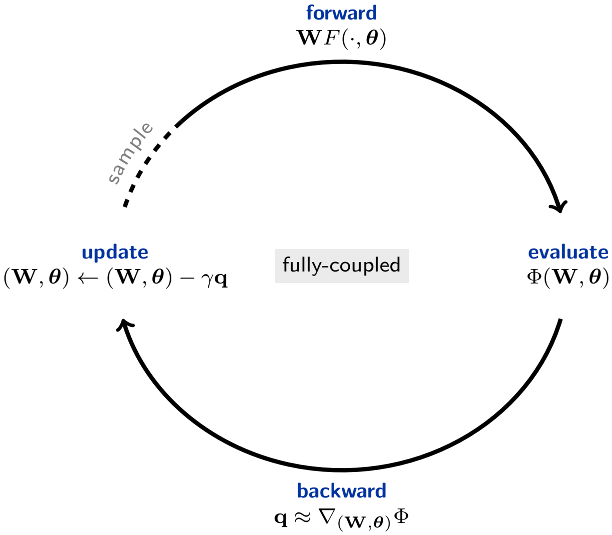
$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W},\boldsymbol{\theta})$$

half step
full step

well-conditioned ill-conditioned coupled + ill-conditioned



The Training Cycle



Two Schools of Training

Sample Average Approximation (SAA)

(Kleywegt, Shapiro, and Mello 2002; Linderoth, Shapiro, and Wright 2006)

$$\min_{\mathbf{W},\boldsymbol{\theta}} \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{y},\mathbf{c})\in\mathcal{T}} L(\mathbf{W}F(\mathbf{y},\boldsymbol{\theta}),\mathbf{c}) + \text{reg.}$$

- ☺ Deterministic
- ☺ Parallelizable
- ☹ Proclivity to overfit
- ☹ Expensive memory-wise

Stochastic Approximation (SA)

(Nemirovski et al. 2009; Robbins and Monro 1951)

$$\min_{\mathbf{W},\boldsymbol{\theta}} \ \mathbb{E}\ L(\mathbf{W}F(\mathbf{y},\boldsymbol{\theta}),\mathbf{c}) + \text{reg.}$$

- ☺ Memory-efficient
- ☺ Generalization
- ☹ Sensitive to hyperparameters
- ☹ Slow to converge (Agarwal et al. 2012)

Roadmap to Better Training

Exploit Separability
linearity of $\mathbf{W}$, coupling of $(\mathbf{W}, \boldsymbol{\theta})$

Sample Average Approximation (SAA)

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi^{\mathrm{saa}}(\mathbf{W}, \boldsymbol{\theta}) \equiv \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{y},\mathbf{c})\in\mathcal{T}} L(\mathbf{W}F(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) + \text{reg.}$$

↓

Variable Projection (VarPro)
optimal $\mathbf{W}(\boldsymbol{\theta})$

↓

GNvpro
faster convergence, higher accuracy

Train Like a (Var)Pro: Efficient Training of Neural Networks with Variable Projection
To appear in SIMODS. arXiv:2007.13171.
Code on Meganet.m

Stochastic Approximation (SA)

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W}, \boldsymbol{\theta}) \equiv \mathbb{E}\, L(\mathbf{W}F(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) + \text{reg.}$$

↓

Iterative Sampling
estimate $\widehat{\mathbf{W}}(\boldsymbol{\theta})$

↓

`slimTrain`
automatic hyperparameter tuning

`slimTrain` – **A Stochastic Approximation Method for Training Separable Deep Neural Networks**
Submitted to SISC. arXiv:2109.14002.
Code on Meganet.m and slimTrain.

Geometric Intuition for Variable Projection (VarPro)

inputs

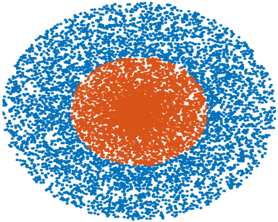
$\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(|\mathcal{T}|)}\} \subset \mathbb{R}^2$

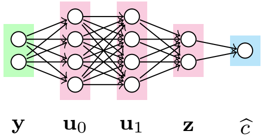
outputs

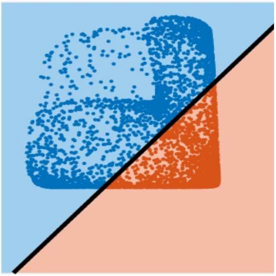
$\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(|\mathcal{T}|)}\} \subset \mathbb{R}^2$

outputs

$\{c^{(1)}, \ldots, c^{(|\mathcal{T}|)}\} \subset \{0, 1\}$







$$\mathbf{u}_0 = \sigma(\mathbf{K}_0\mathbf{y} + \mathbf{b}_0) \quad \in \mathbb{R}^4$$

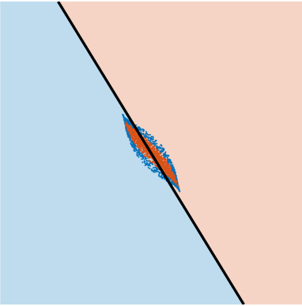
$$\mathbf{u}_1 = \sigma(\mathbf{K}_1\mathbf{u}_0 + \mathbf{b}_1) \quad \in \mathbb{R}^4$$

$$\mathbf{z} = \sigma(\mathbf{K}_2\mathbf{u}_1 + \mathbf{b}_2) \quad \in \mathbb{R}^2$$

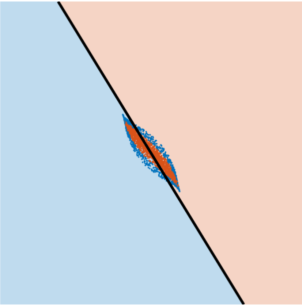
$$\hat{c} = \mathbf{W}\mathbf{z} \quad \in \mathbb{R}$$

Geometric Intuition for Variable Projection (VarPro)

network weights $\mathbf{W}$



optimal $\mathbf{W}(\boldsymbol{\theta})$



Variable Projection

SAA Full Optimization Problem

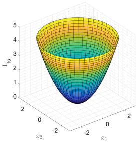
$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi^{\mathrm{saa}}(\mathbf{W},\boldsymbol{\theta}) \equiv \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{y},\mathbf{c})\in\mathcal{T}} L(\mathbf{W}F(\mathbf{y},\boldsymbol{\theta}),\mathbf{c}) + R(\boldsymbol{\theta}) + S(\mathbf{W})$$

Reduced Optimization Problem

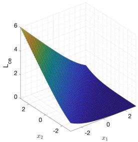
$$\min_{\boldsymbol{\theta}} \Phi^{\mathrm{saa}}_{\mathrm{red}}(\boldsymbol{\theta}) \equiv \Phi^{\mathrm{saa}}(\mathbf{W}(\boldsymbol{\theta}),\boldsymbol{\theta}) \quad \text{s.t.} \quad \mathbf{W}(\boldsymbol{\theta}) = \arg\min_{\mathbf{W}} \Phi^{\mathrm{saa}}(\mathbf{W},\boldsymbol{\theta})$$

Assume $\Phi^{\mathrm{saa}}(\mathbf{W},\boldsymbol{\theta})$ is smooth and strictly convex in the first argument.

Least Squares Loss



Cross Entropy Loss



Use **Newton-Krylov Trust Region Method** to solve for $\mathbf{W}(\boldsymbol{\theta})$ to high accuracy.

Optimizing $\boldsymbol{\theta}$: Gauss-Newton-Krylov VarPro (GNvpro)

Reduced Optimization Problem

$$\min_{\boldsymbol{\theta}} \Phi_{\text{red}}^{\text{saa}}(\boldsymbol{\theta}) \equiv \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{y},\mathbf{c})\in\mathcal{T}} L(\mathbf{W}(\boldsymbol{\theta})F(\mathbf{y},\boldsymbol{\theta}),\mathbf{c}) + R(\boldsymbol{\theta}) + S(\mathbf{W}(\boldsymbol{\theta}))$$

First-Order Methods: Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma\mathbf{p}$ where $\mathbf{p} \approx \nabla\Phi_{\text{red}}^{\text{saa}}(\boldsymbol{\theta})$

$$\nabla_{\mathbf{W}}\Phi^{\text{saa}}(\mathbf{W}(\boldsymbol{\theta}),\boldsymbol{\theta}) = \mathbf{0} \quad \Longrightarrow \quad \nabla_{\boldsymbol{\theta}}\Phi_{\text{red}}^{\text{saa}}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\Phi^{\text{saa}}(\mathbf{W}(\boldsymbol{\theta}),\boldsymbol{\theta})$$

Gauss-Newton-Krylov Trust Region Method: Update $\boldsymbol{\theta}_{\text{trial}} = \boldsymbol{\theta}^{(k)} + \mathbf{p}$

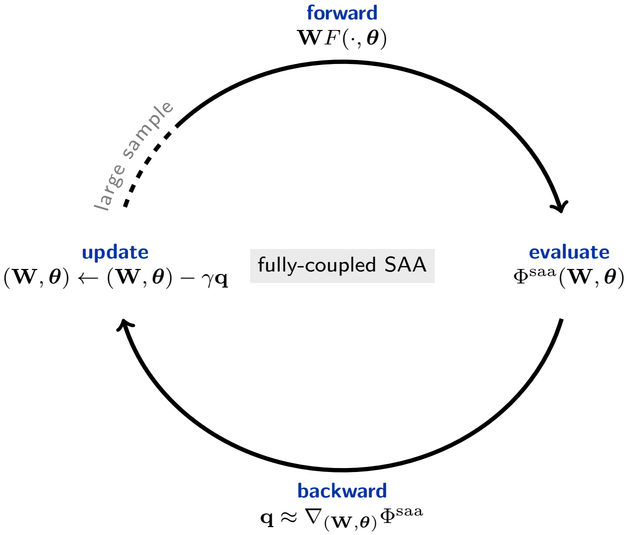
$$\min_{\mathbf{p}} \nabla_{\boldsymbol{\theta}}\Phi_{\text{red}}^{\text{saa}}(\boldsymbol{\theta}^{(k)})^\top\mathbf{p} + \tfrac{1}{2}\mathbf{p}^\top\nabla_{\boldsymbol{\theta}}^2\Phi_{\text{red}}^{\text{saa}}(\boldsymbol{\theta}^{(k)})\mathbf{p} \quad \text{s.t.} \quad \|\mathbf{p}\| \le \Delta^{(k)}$$

Approximate the Hessian by

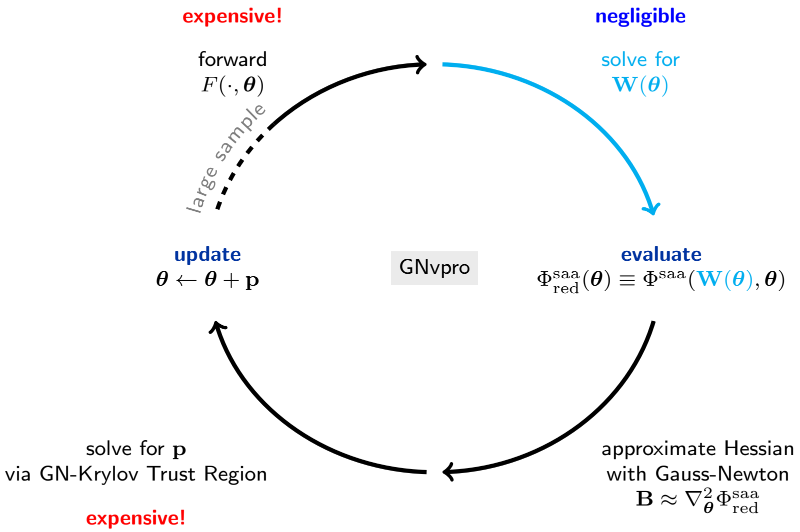
$$\nabla_{\boldsymbol{\theta}}^2\Phi_{\text{red}}^{\text{saa}}(\boldsymbol{\theta}) \approx J_{\boldsymbol{\theta}}(\mathbf{W}(\boldsymbol{\theta})F(\mathbf{y},\boldsymbol{\theta}))^\top\nabla^2 L J_{\boldsymbol{\theta}}(\mathbf{W}(\boldsymbol{\theta})F(\mathbf{y},\boldsymbol{\theta})) + \nabla^2 R$$

O'Leary and Rust 2013

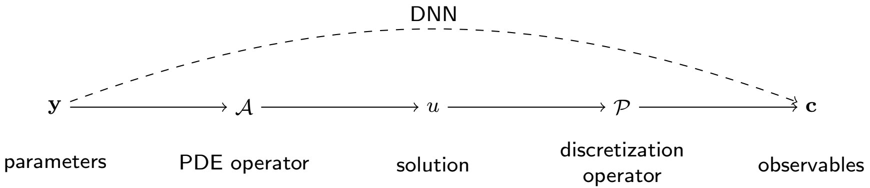
Train Like a (Var)Pro



Train Like a (Var)Pro



PDE Surrogate Modeling



$$\mathbf{c} = \mathcal{P}u \quad \text{subject to} \quad \mathcal{A}(u;\mathbf{y}) = 0$$

PDEs and Network Architectures:

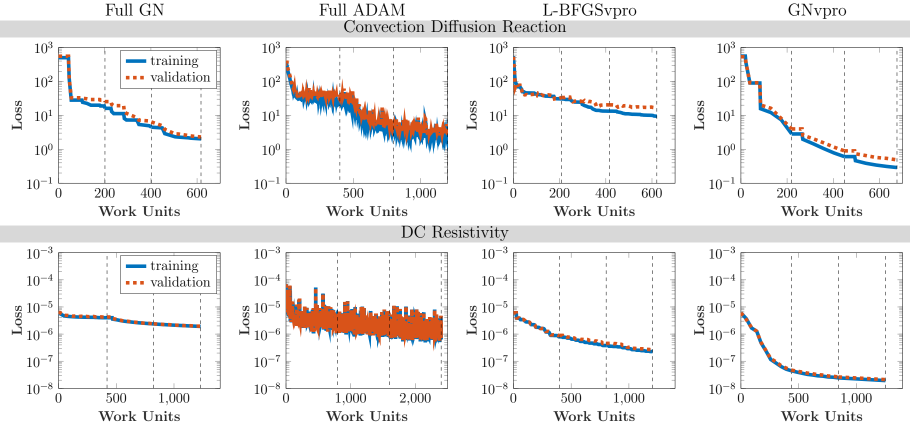
- Convection Diffusion Reaction: (Grasso and Innocente 2018; Choquet and Comte 2017)

$$\mathbf{y} \in \mathbb{R}^{55} \to \underbrace{\mathbb{R}^8 \to \cdots \to \mathbb{R}^8}_{d} \to \mathbb{R}^{72} \ni \mathbf{c}$$

- Direct Current Resistivity: (Seidel and Lange 2007; Dey and Morrison 1979)

$$\mathbf{y} \in \mathbb{R}^{3} \to \underbrace{\mathbb{R}^{16} \to \cdots \to \mathbb{R}^{16}}_{d} \to \mathbb{R}^{882} \ni \mathbf{c}$$

PDE Surrogate Modeling



Work Units = number of forward and backward passes through network

SGD: 2 work units per epoch (1 forward pass, 1 backward pass)
GNvpro: 2 works units + $2r$ work units for rank-r approx. to $\nabla^2_{\boldsymbol{\theta}} \Phi_{\text{red}}$ per iteration

Roadmap to Better Training

Exploit Separability

linearity of $\mathbf{W}$, coupling of $(\mathbf{W}, \boldsymbol{\theta})$

Sample Average Approximation (SAA)

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi^{\text{saa}}(\mathbf{W}, \boldsymbol{\theta}) \equiv \frac{1}{|\mathcal{T}|} \sum_{(\mathbf{y},\mathbf{c})\in\mathcal{T}} L(\mathbf{W}F(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) + \text{reg.}$$

Variable Projection (VarPro)

optimal $\mathbf{W}(\boldsymbol{\theta})$

GNvpro

faster convergence, higher accuracy

Stochastic Approximation (SA)

$$\min_{\mathbf{W},\boldsymbol{\theta}} \Phi(\mathbf{W}, \boldsymbol{\theta}) \equiv \mathbb{E}\, L(\mathbf{W}F(\mathbf{y}, \boldsymbol{\theta}), \mathbf{c}) + \text{reg.}$$

Iterative Sampling

estimate $\widehat{\mathbf{W}}(\boldsymbol{\theta})$

`slimTrain`

automatic hyperparameter tuning

Train Like a (Var)Pro: Efficient Training of Neural Networks with Variable Projection
To appear in SIMODS. arXiv:2007.13171.
Code on Meganet.m

`slimTrain` – **A Stochastic Approximation Method for Training Separable Deep Neural Networks**
Submitted to SISC. arXiv:2109.14002.
Code on Meganet.m and slimTrain.

Does VarPro Extend to Stochastic Approximation?

Consider the reduced *stochastic* optimization problem

$$\min_{\boldsymbol{\theta}} \Phi_{\text{red}}(\boldsymbol{\theta}) \equiv \Phi(\widehat{\mathbf{W}}(\boldsymbol{\theta}), \boldsymbol{\theta})$$
$$\text{s.t.} \quad \widehat{\mathbf{W}}(\boldsymbol{\theta}) = \arg\min_{\mathbf{W}} \Phi(\mathbf{W}, \boldsymbol{\theta}).$$

Key Idea of SA: use minibatches $\mathcal{T}_k \subset \mathcal{T}$ to update $\boldsymbol{\theta}$

Key Ingredient: need an unbiased derivative estimate of $\boldsymbol{\theta}$

$$\mathbb{E}\left(\mathrm{D}_{\boldsymbol{\theta}} \Phi_{\text{red},k}(\boldsymbol{\theta})\right) = \mathrm{D}_{\boldsymbol{\theta}} \Phi_{\text{red}}(\boldsymbol{\theta}) \qquad \Phi_{\text{red},k} \approx \Phi_{\text{red}} \text{ using } \mathcal{T}_k$$

Proof:

$$\mathbb{E}\left(\mathrm{D}_{\boldsymbol{\theta}} \Phi_{\text{red},k}(\boldsymbol{\theta})\right) = \underbrace{\mathbb{E}\left([\mathrm{D}_{\mathbf{W}} \Phi_k(\mathbf{W}, \boldsymbol{\theta})]_{\mathbf{W}=\widehat{\mathbf{W}}(\boldsymbol{\theta})}\right)}_{=\mathbf{0}} \mathrm{D}_{\boldsymbol{\theta}} \widehat{\mathbf{W}}(\boldsymbol{\theta}) + \underbrace{\mathbb{E}\left([\mathrm{D}_{\widetilde{\boldsymbol{\theta}}} \Phi_k(\widehat{\mathbf{W}}(\boldsymbol{\theta}), \widetilde{\boldsymbol{\theta}})]_{\widetilde{\boldsymbol{\theta}}=\boldsymbol{\theta}}\right)}_{D_{\boldsymbol{\theta}} \Phi_{\text{red}}(\boldsymbol{\theta})}$$

In practice, use an effective iterative scheme to estimate $\widehat{\mathbf{W}}(\boldsymbol{\theta})$ and reduce bias.

Exploiting Separability with Iterative Sampling

Consider the stochastic least-squares problem with Tikhonov regularization

$$\min_{\mathbf{w},\boldsymbol{\theta}} \Psi(\mathbf{w},\boldsymbol{\theta}) \equiv \mathbb{E}\ \tfrac{1}{2}\|\mathbf{A}(\mathbf{y},\boldsymbol{\theta})\mathbf{w} - \mathbf{c}\|_2^2 + \tfrac{1}{2}\alpha\|\mathbf{L}\boldsymbol{\theta}\|_2^2 + \tfrac{1}{2}\lambda\|\mathbf{w}\|_2^2$$

Iterative Sampling for $\mathbf{w}$

(Chung et al. 2020; Slagel et al. 2019; Chung, Chung, and Slagel 2019)

$$\mathbf{w}_k = \mathbf{w}_{k-1} - \mathbf{s}_k(\boldsymbol{\theta}_{k-1})$$

SGD Variant for $\boldsymbol{\theta}$

(Kingma and Ba 2014; Chen et al. 2021; Yao et al. 2020; Duchi, Hazan, and Singer 2011)

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \gamma\mathbf{p}_k(\mathbf{w}_k)$$

Why Iterative Sampling?

- ☺ known convergence properties
- ☺ incorporate global curvature information (challenging in SA (Bottou and Cun 2004; Gower and Richtárik 2017; Byrd et al. 2016; Wang et al. 2017; Chung et al. 2017))
- ☺ no learning rate
- ☺ adaptive choice of regularization parameter

Sampled Limited-Memory Tikhonov (slimTik)

$$\min_{\mathbf{w}} \ \mathbb{E} \ \tfrac{1}{2}\|\mathbf{A}(\mathbf{y},\boldsymbol{\theta}_{k-1})\mathbf{w} - \mathbf{c}\|_2^2 + \tfrac{1}{2}\lambda\|\mathbf{w}\|_2^2.$$

At iteration k, update linear weights by

$$\mathbf{w}_k = \mathbf{w}_{k-1} - \underbrace{\mathbf{B}_k \mathbf{g}_k(\mathbf{w}_{k-1})}_{\mathbf{s}_k(\Lambda_k)}$$

Local Gradient Information (batch k)

$$\mathbf{g}_k(\mathbf{w}_{k-1}) = \mathbf{A}_k^\top(\mathbf{A}_k\mathbf{w}_{k-1} - \mathbf{c}_k) + \Lambda_k \mathbf{w}_{k-1}$$

Global Curvature Information (all batches)

$$\mathbf{B}_k = \left((\Lambda_k + \sum_{i=1}^{k-1} \Lambda_i)\mathbf{I} + \sum_{i=k-r}^{k} \mathbf{A}_i{}^\top \mathbf{A}_i \right)^{-1}$$

$\mathbf{A}_j(\boldsymbol{\theta}_{j-1})$: output features for batch j

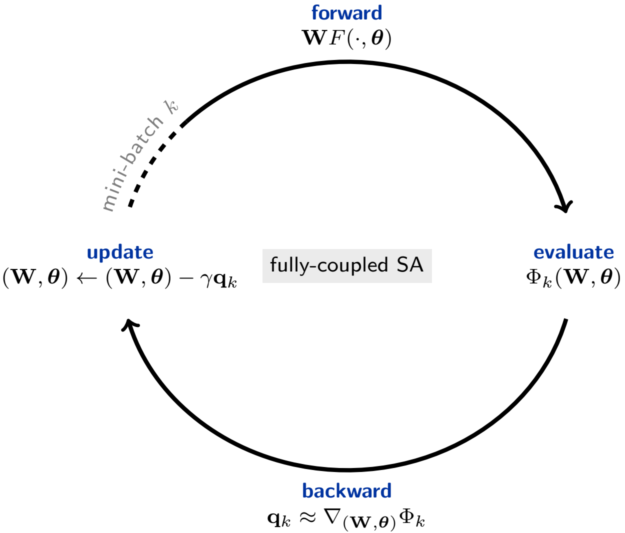
$\mathbf{c}_j$: target features for batch j

Λ_j: (optimal) reg. parameter for batch j

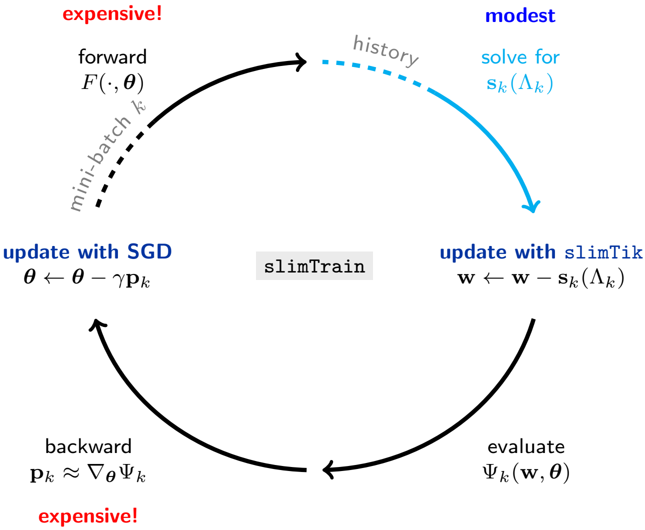
- ☺ Use sampled regularization parameter selection methods (e.g., sGCV) to choose Λ_k.
- ☹ Curvature information depends on older $\boldsymbol{\theta}$ iterates.
- ☺ Use **sampled limited-memory Tikhonov (slimTik)** with memory depth $r \in \mathbb{N}_0$.

Slagel et al. 2019

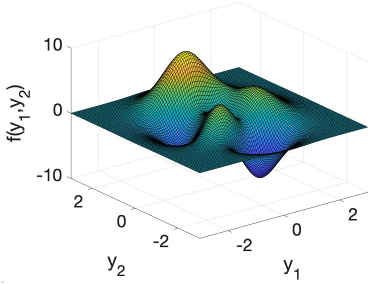
slimTrain: Sampled Limited-Memory Training

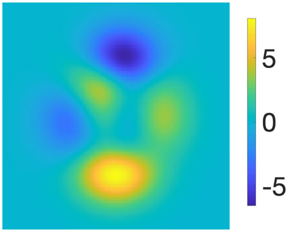


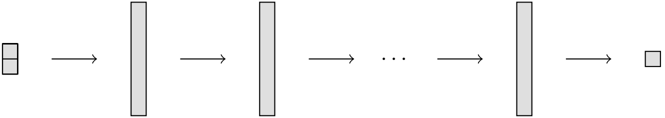
slimTrain: Sampled Limited-Memory Training



Function Approximation: Peaks





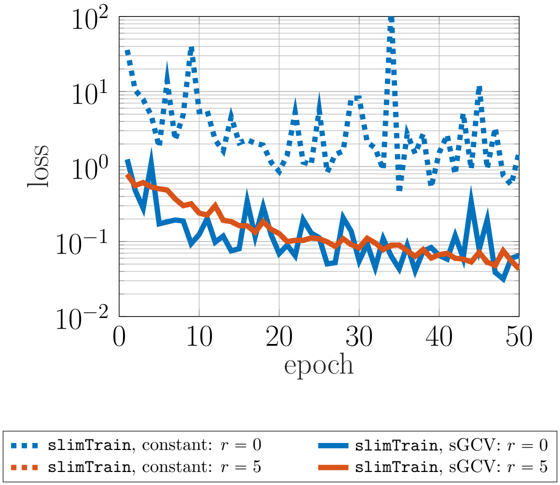


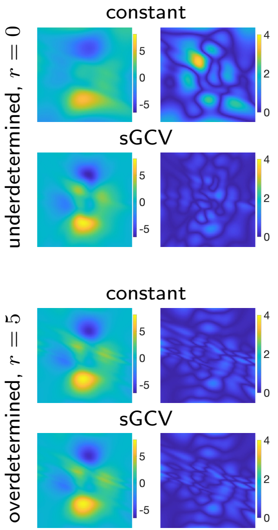
$\mathbf{y} \in \mathbb{R}^2$ $\quad \mathbf{u}_0 \in \mathbb{R}^8$ $\quad \mathbf{u}_1 \in \mathbb{R}^8$ $\quad \mathbf{u}_8 \in \mathbb{R}^8$ $\quad [\mathbf{u}_8, 1]^\top \mathbf{w} \in \mathbb{R}$

$\mathbf{w} \in \mathbb{R}^9$

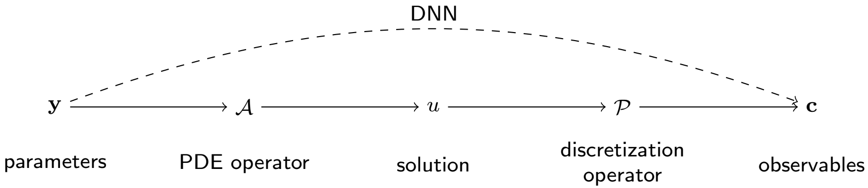
Function Approximation: Peaks

batch size $= 5,\ \gamma = 10^{-3},\ \lambda = 10^{-10}$





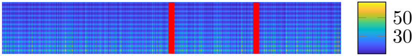
PDE Surrogate Modeling: CDR



$$\mathbf{c} = \mathcal{P}u \quad \text{subject to} \quad \mathcal{A}(u; \mathbf{y}) = 0$$

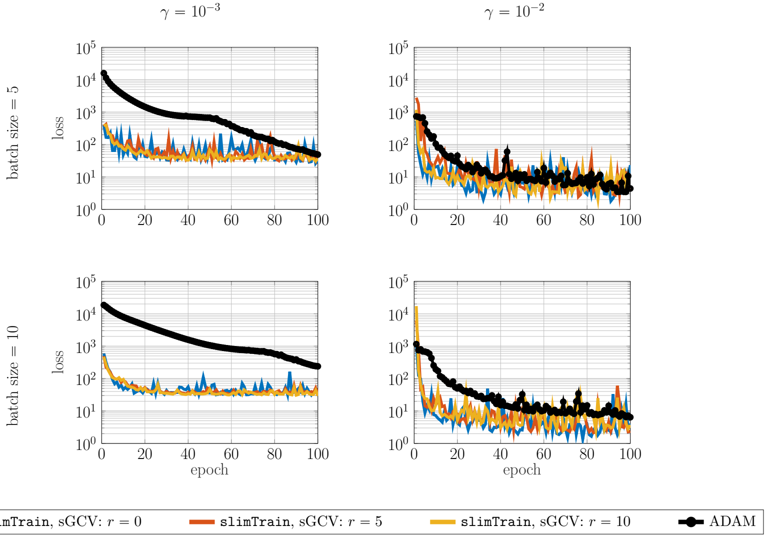
Convection Diffusion Reaction: (Grasso and Innocente 2018; Choquet and Comte 2017)

$$\mathbf{y} \in \mathbb{R}^{55} \to \underbrace{\mathbb{R}^8 \to \cdots \to \mathbb{R}^8}_{d} \to \mathbb{R}^{72} \ni \mathbf{c}$$

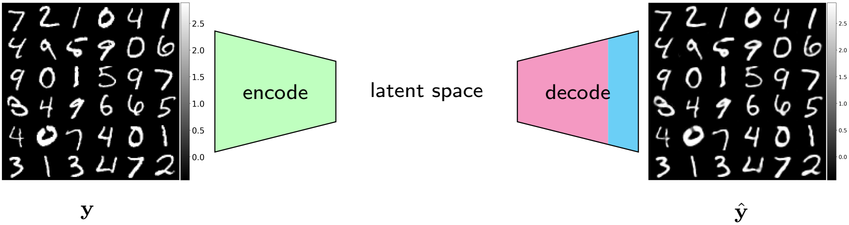


observables

PDE Surrogate Modeling: CDR



Dimensionality Reduction: Autencoder



Goal: Train two networks such that $\hat{\mathbf{y}} \approx \mathbf{y}$ for all inputs $\mathbf{y}$.

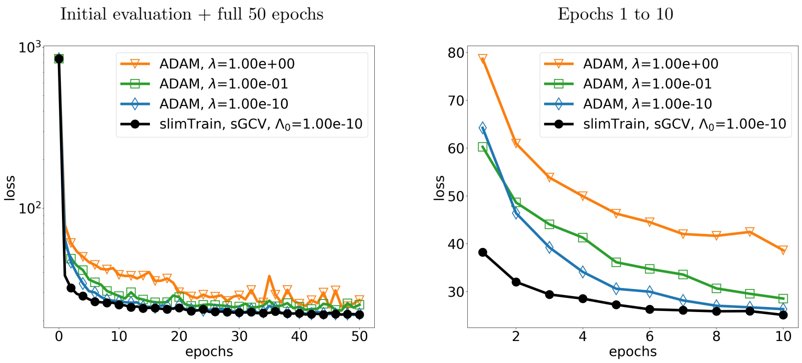
$$\min_{\mathbf{w},\boldsymbol{\theta}_{\text{dec}},\boldsymbol{\theta}_{\text{enc}}} \mathbb{E}\ \frac{1}{2}\|\mathbf{K}(\mathbf{w})F_{\text{dec}}(\ F_{\text{enc}}(\mathbf{y},\boldsymbol{\theta}_{\text{enc}})\ ,\boldsymbol{\theta}_{\text{dec}}) - \mathbf{y}\|_2^2 + \text{reg.}$$

Final Layer: $\mathbf{K}(\mathbf{w})$ is a (transposed) convolutional operator

LeCun et al. 1990

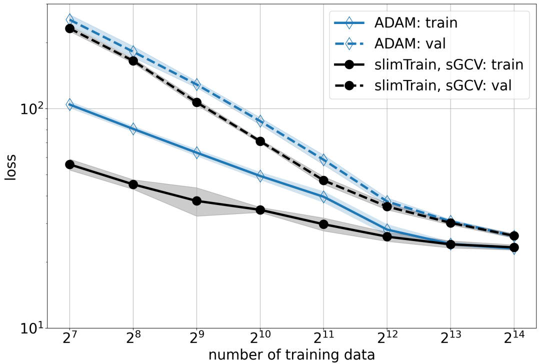
Dimensionality Reduction: Autencoder

Full Data Regime: $50,000$ training images



Dimensionality Reduction: Autencoder

Limited Data Regime: best loss in 50 epochs



Wrapping Up

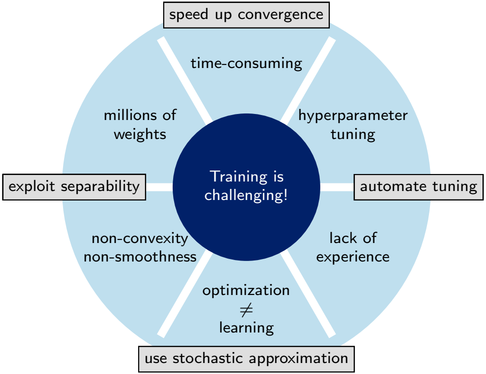
Exploiting separability makes DNN training easier!

GNvpro...

- accelerates training to high accuracy
- can be applied to non-quadratic loss functions

`slimTrain`...

- automates regularization parameter selection
- can outperform ADAM with recommended settings and with limited data

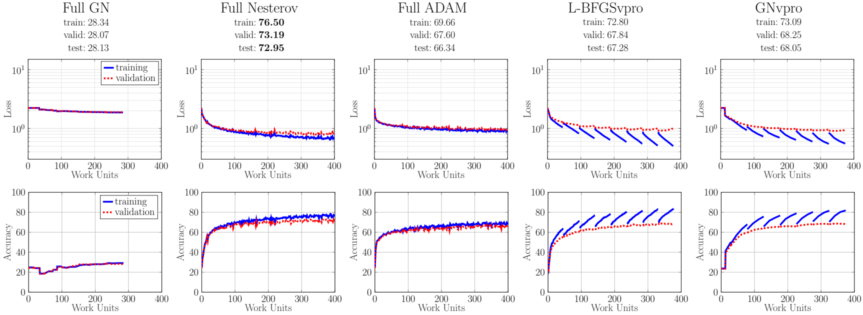


Train Like a (Var)Pro: Efficient Training of Neural Networks with Variable Projection
To appear in SIMODS. arXiv:2007.13171.
Code on Meganet.m.

`slimTrain` – **A Stochastic Approximation Method for Training Separable Deep Neural Networks**
Submitted to SISC. arXiv:2109.14002.
Code on Meganet.m and slimTrain.

Thanks for Listening! For more Q&A, please reach out to elizabeth.newman@emory.edu and lruthotto@emory.edu

Image Classification: CIFAR-10



$$\mathbf{y} \in \mathbb{R}^{32\times32\times3} \xrightarrow[\text{conv}]{5\times5} \mathbb{R}^{32\times32\times32} \xrightarrow[\text{pool}]{2\times2} \mathbb{R}^{16\times16\times32} \xrightarrow[\text{conv}]{5\times5} \mathbb{R}^{16\times16\times64} \xrightarrow[\text{pool}]{16\times16} \mathbb{R}^{64} \longrightarrow \mathbb{R}^{10} \ni \mathbf{c}$$

Krizhevsky, Sutskever, and Hinton 2012

References I

Agarwal, Alekh et al. (2012). "Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization". In: *IEEE Transactions on Information Theory* 58.5, pp. 3235–3249.

Baumgardner, Marion F., Larry L. Biehl, and David A. Landgrebe (2015). *220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3*. DOI: doi:/10.4231/R7RX991C. URL: https://purr.purdue.edu/publications/1947/1.

Bottou, L and YL Cun (2004). "Large scale online learning". In: *Advances in Neural Information Processing Systems*, pp. 217–224.

Byrd, RH et al. (2016). "A Stochastic Quasi-Newton Method for Large-Scale Optimization". In: *SIAM Journal on Optimization* 26.2, pp. 1008–1031.

Chen, Congliang et al. (2021). *Towards Practical Adam: Non-Convexity, Convergence Theory, and Mini-Batch Acceleration*. arXiv: 2101.05471 [cs.LG].

Choquet, EmmanuelleAugeraud-Vèronand Catherine and Èloïsese Comte (2017). "Optimal Control for a Groundwater Pollution Ruled by a ConvectionDiffusionReaction Problem". In: *Journal of Optimization Theory and Applications*.

Chung, Julianne, Matthias Chung, and J Tanner Slagel (2019). "Iterative sampled methods for massive and separable nonlinear inverse problems". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, pp. 119–130.

Chung, Julianne et al. (2017). "Stochastic Newton and quasi-Newton methods for large linear least-squares problems". In: *arXiv preprint arXiv:1702.07367*.

— (2020). "Sampled limited memory methods for massive linear inverse problems". In: *Inverse Problems* 36.5, p. 054001.

References II

Covington, Paul, Jay Adams, and Emre Sargin (2016). "Deep Neural Networks for YouTube Recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA.

Dey, A. and H.F. Morrison (1979). "Resistivity modeling for arbitrarily shaped three dimensional structures". In: *Geophysics* 44, pp. 753–780.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization.". In: *Journal of machine learning research* 12.7.

E, Weinan and Bing Yu (2018). "The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems". In: *Communications in Mathematics and Statistics* 6.1, pp. 1–12. DOI: 10.1007/s40304-018-0127-z. URL: https://doi.org/10.1007/s40304-018-0127-z.

Golub, G.H. and V. Pereyra (1973). "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems whose Variables Separate". In: *SIAM Journal on Numerical Analysis* 10.2, pp. 413–432.

Goodfellow, Ian J. et al. (2014). *Generative Adversarial Networks*. arXiv: 1406.2661 [stat.ML].

Gower, RM and P Richtárik (2017). "Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms". In: *SIAM Journal on Matrix Analysis and Applications* 38.4, pp. 1380–1409.

Grasso, Paolo and Mauro S. Innocente (2018). *Advances in Forest Fire Research: A two-dimensional reaction-advection-diffusion model of the spread of fire in wildlands*. Imprensa da Universidade de Coimbra.

Han, Jiequn, Arnulf Jentzen, and Weinan E (2018). "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34, pp. 8505–8510. ISSN: 0027-8424. DOI: 10.1073/pnas.1718942115. eprint: https://www.pnas.org/content/115/34/8505.full.pdf. URL: https://www.pnas.org/content/115/34/8505.

References III

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Hinton, G. E. and R. R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786, pp. 504–507. DOI: 10.1126/science.1127647.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kleywegt, Anton J., Alexander Shapiro, and Tito Homem-de Mello (2002). "The Sample Average Approximation Method for Stochastic Discrete Optimization". In: *SIAM Journal on Optimization* 12.2, pp. 479–502. DOI: 10.1137/S1052623499363220. eprint: https://doi.org/10.1137/S1052623499363220. URL: https://doi.org/10.1137/S1052623499363220.

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*.

LeCun, Y. et al. (1990). "Handwritten Digit Recognition with a Back-Propagation Network". In: *Advances in Neural Information Processing Systems* 2.

Linderoth, Jeff, Alexander Shapiro, and Stephen Wright (2006). "The empirical behavior of sampling methods for stochastic programming". In: *Annals of Operations Research* 142.1, pp. 215–241. DOI: 10.1007/s10479-006-6169-8. URL: https://doi.org/10.1007/s10479-006-6169-8.

Men, Kuo et al. (2017). "Deep Deconvolutional Neural Network for Target Segmentation of Nasopharyngeal Cancer in Planning Computed Tomography Images". In: *Frontiers in Oncology* 7, p. 315. ISSN: 2234-943X. DOI: 10.3389/fonc.2017.00315. URL: https://www.frontiersin.org/article/10.3389/fonc.2017.00315.

References IV

Nemirovski, A. et al. (2009). "Robust Stochastic Approximation Approach to Stochastic Programming". In: *SIAM Journal on Optimization* 19.4, pp. 1574–1609. DOI: 10.1137/070704277. eprint: https://doi.org/10.1137/070704277. URL: https://doi.org/10.1137/070704277.

O'Leary, Dianne P and Bert W Rust (2013). "Variable projection for nonlinear least squares problems". In: *Computational Optimization and Applications. An International Journal* 54.3, pp. 579–593.

Raissi, Maziar, Paris Perdikaris, and George E Karniadakis (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378, pp. 686–707.

Robbins, H and S Monro (1951). "A Stochastic Approximation Method". In: *The annals of mathematical statistics* 22.3, pp. 400–407.

Seidel, Knut and Gerhard Lange (2007). "Direct Current Resistivity Methods". In: *Environmental Geology: Handbook of Field Methods and Case Studies*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 205–237. ISBN: 978-3-540-74671-3. DOI: 10.1007/978-3-540-74671-3_8. URL: https://doi.org/10.1007/978-3-540-74671-3_8.

Simonyan, Karen and Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV].

Slagel, J Tanner et al. (2019). "Sampled Tikhonov regularization for large linear inverse problems". In: *Inverse Problems* 35.11, p. 114008.

Wang, Xiao et al. (2017). "Stochastic quasi-Newton methods for nonconvex stochastic optimization". In: *SIAM Journal on Optimization* 27.2, pp. 927–956.

References V

Yao, Zhewei et al. (2020). *ADAHESSIAN: An Adaptive Second Order Optimizer for Machine Learning*. arXiv: 2006.00719 [cs.LG].