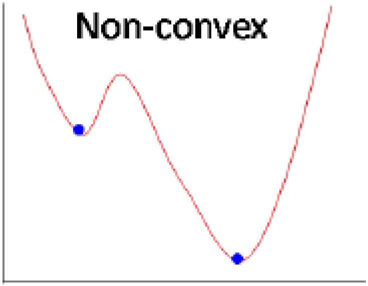
# Stochastic Second-Order Optimization Methods

## Part II: Non-Convex

Fred Roosta

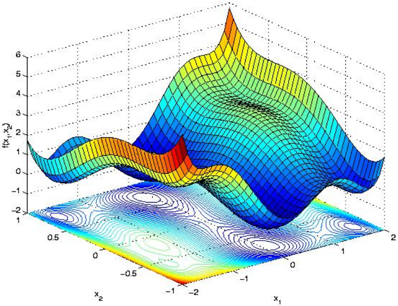
School of Mathematics and Physics
University of Queensland

Now, moving onto non-convex problems...



# Non-Convex Is Hard!

- Saddle points, Local Minima, Local Maxima



- 2nd Order Necessary Condition

$$\nabla F(\mathbf{x}^\star) = 0 \qquad \nabla^2 F(\mathbf{x}^\star) \succeq 0$$

- 2nd Order Sufficient Condition

$$\nabla F(\mathbf{x}^\star) = 0 \qquad \nabla^2 F(\mathbf{x}^\star) \succ 0$$

# Non-Convex Is Hard!

- Additional complexity issues...
  - Optimization of a degree four polynomial: NP-hard [Hillar et al., 2013]
  - Checking for sufficient optimality condition: co-NP-complete [Murty et al., 1987]
  - Checking whether a point is not a local minimum: NP-complete [Murty et al., 1987]

All convex problems are the same,
while every non-convex problem is different.

Not sure who's quote this is!

# $(\epsilon_g, \epsilon_H) -$ *Optimality*

$$\|\nabla F(\mathbf{x})\| \leq \epsilon_g \qquad \text{and} \qquad \lambda_{\min}(\nabla^2 F(\mathbf{x})) \geq -\epsilon_H$$

# Outline

- Part I: Convex
  - Smooth
    - Newton-CG
  - Non-Smooth
    - Proximal Newton
    - Semi-smooth Newton
- Part II: Non-Convex
  - Line-Search Based Methods
    - L-BFGS
    - Gauss-Newton
    - Natural Gradient
  - Trust-Region Based Methods
    - Trust-Region
    - Cubic Regularization
- Part III: Discussion and Examples

- **Sad Note ☹:** BFGS may fail on non-convex problems with both exact line search [Mascarenhas, 2004] and inexact (e.g., Wolfe) variants [Dai, 2002]
- **Happy Note ☺:** BFGS dominates in many practical non-convex applications

# Newton's Method: Scalar Case

Finding a root of $r : \mathbb{R} \to \mathbb{R}$ , i.e., find $x^\star$ for which $r(x^\star) = 0$:

$$0 = r(x^\star) = r(x^{(k)}) + \left(x^\star - x^{(k)}\right) r'(x^{(k)}) + o(|x^\star - x^{(k)}|)$$

$$0 = r(x^{(k)}) + \left(x^{(k+1)} - x^{(k)}\right) r'(x^{(k)})$$

$$x^{(k+1)} = x^{(k)} - \frac{r(x^{(k)})}{r'(x^{(k)})}.$$

# Secant Method: Scalar Case

Finding a root of $r : \mathbb{R} \to \mathbb{R}$, i.e., find $x^\star$ for which $r(x^\star) = 0$:

Approximate the derivative: $$r'(x^{(k)}) \approx \frac{r(x^{(k)}) - r(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

$$x^{(k+1)} = x^{(k)} - \left( \frac{x^{(k)} - x^{(k-1)}}{r(x^{(k)}) - r(x^{(k-1)})} \right) r(x^{(k)}).$$

Local convergence rate is

$$\left| x^{(k+1)} - x^\star \right| \leq C \left| x^{(k)} - x^\star \right|^{\overbrace{\frac{1+\sqrt{5}}{2}}^{\text{“Golden Ratio”}}}.$$

In contrast, rate of convergence of Newton is quadratic!

# Quasi-Newton Method

Quasi-Newton optimization methods extend secant method to multivariable case!

# Quasi-Newton Method

For $r(x) = f'(x)$, we have

$$f''(x^{(k)}) \approx \frac{f'(x^{(k)}) - f'(x^{(k-1)})}{x^{(k)} - x^{(k-1)}},$$

i.e.,

$$f''(x^{(k)}) \left( x^{(k)} - x^{(k-1)} \right) \approx f'(x^{(k)}) - f'(x^{(k-1)}).$$

# Quasi-Newton Method

$$\begin{aligned}\nabla f(\mathbf{x}+\mathbf{p}) &= \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})\mathbf{p} + \int_0^1 \left[\nabla^2 f(\mathbf{x}+t\mathbf{p}) - \nabla^2 f(\mathbf{x})\right] \mathbf{p}\, \mathrm{d}t \\ &= \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})\mathbf{p} + o(\|\mathbf{p}\|),\end{aligned}$$

i.e., when $\mathbf{x} = \mathbf{x}^{(k)}, \mathbf{p} = \mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}$, and $\|\mathbf{p}\| \ll 1$ , we have

$$\nabla^2 f(\mathbf{x}^{(k)}) \left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right) \approx \left(\nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k-1)})\right)$$

# Quasi-Newton Method

$$\nabla^2 f(\mathbf{x}^{(k)}) \left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right) \approx \left(\nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k-1)})\right)$$

So, look for $\mathbf{H}^{(k)} \approx \nabla^2 f(\mathbf{x}^{(k)})$ such that

$$\underbrace{\mathbf{H}^{(k)} \left(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right) = \left(\nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k-1)})\right)}_{\text{Secant Condition}}$$

# Quasi-Newton Method: Another Interpretation

Another interpretation of the secant condition...

# Quasi-Newton Method: Another Interpretation

**Recall:**

### Iterative Scheme

$$\mathbf{y}^{(k)} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{X}} \left\{ (\mathbf{x}-\mathbf{x}^{(k)})^T \mathbf{g}^{(k)} + \frac{1}{2}(\mathbf{x}-\mathbf{x}^{(k)})^T \mathbf{H}^{(k)} (\mathbf{x}-\mathbf{x}^{(k)}) \right\}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \left( \mathbf{y}^{(k)} - \mathbf{x}^{(k)} \right)$$

# Quasi-Newton Method: Another Interpretation

- Suppose we have a $\mathbf{H}^{(k)}$ and $\mathbf{x}^{(k+1)}$
- How to update $\mathbf{H}^{(k)}$ to obtain a new quadratic approximation to $F(\mathbf{x})$ at $\mathbf{x}^{(k+1)}$?

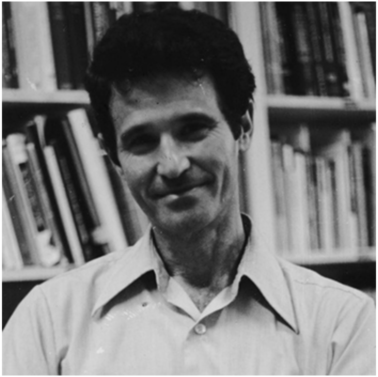
$$m_{k+1}(\mathbf{p}) \triangleq f(\mathbf{x}^{(k+1)}) + \left\langle \nabla f(\mathbf{x}^{(k+1)}), \mathbf{p} \right\rangle + \frac{1}{2} \left\langle \mathbf{p}, \mathbf{H}^{(k+1)} \mathbf{p} \right\rangle$$

One reasonable requirement, suggested by Davidon, is

- $\nabla m_{k+1}(\mathbf{0}) = \nabla f(\mathbf{x}^{(k+1)}) \longrightarrow$ trivially satisfied
- $\nabla m_{k+1}(-\alpha \mathbf{p}_k) = \nabla f(\mathbf{x}^{(k)}) \longrightarrow$ secant condition

# Quasi-Newton Method: DFP

The revolution began with...



William C. Davidon

DFP: Davidon-Fletcher-Powell scheme

# Quasi-Newton Method

$$\mathbf{H}^{(k)} \left( \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right) = \left( \nabla f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k-1)}) \right).$$

$d$ equations vs. $d^2$ unknowns

The difference between QNMs boils down to how they update $\mathbf{H}^{(k)}$ (or its inverse)!

# Quasi-Newton Method

Typical notation in QN literature:

$$\mathbf{s}_k \triangleq \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$$
$$\mathbf{y}_k \triangleq \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})$$

# Quasi-Newton Method: BFGS

Updating $\mathbf{B}^{(k)} \triangleq \left[\mathbf{H}^{(k)}\right]^{-1}$:

$$\begin{aligned} \min_{\mathbf{B} \in \mathbb{R}^{d \times d}} \quad & \|\mathbf{B} - \mathbf{B}^{(k)}\| \\ \text{s.t.} \quad & \mathbf{B} = \mathbf{B}^T, \quad \mathbf{s}_k = \mathbf{B}\mathbf{y}_k \end{aligned}$$

- With $\|\mathbf{A}\| = \|\mathbf{W}^{1/2}\mathbf{A}\mathbf{W}^{1/2}\|_F$ for a particular $\mathbf{W}$

$$\mathbf{B}^{(k+1)} = \left(\mathbb{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) \mathbf{B}^{(k)} \left(\mathbb{I} - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}\right) + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}$$

- $\mathbf{B}^{(k)} \succ 0$ iff $\mathbf{y}_k^T \mathbf{s}_k > 0$ (Curvature Condition) $\Longrightarrow$ Guaranteed by appropriate line search, e.g., Armijo + (strong) Wolfe
- Under strong convexity (or if iterates satisfy certain properties), asymptotic super-linear rate of convergence

# Quasi-Newton Method: Limited Memory

**General QNM Update:**

$$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + [\text{something}]$$

- **Problem:** Memory storage is $\mathcal{O}(d^2)$
- **Soution:** Limited-Memory QNMs, which are low-storage methods

# L-BFGS

Instead of storing the inverse Hessian $\mathbf{B}^{(k)}$, L-BFGS maintains a history of iterates and gradients as

$$\{\mathbf{s}_{k-m},\ \mathbf{s}_{k-m-1},\ \ldots,\ \mathbf{s}_k\},$$
$$\{\mathbf{y}_{k-m},\ \mathbf{y}_{k-m-1},\ \ldots,\ \mathbf{y}_k\}.$$

- $\mathbf{B}^{(k)}$ depends on $\mathbf{B}^{(k-1)}$, $\mathbf{y}_{k-1}$ and $\mathbf{s}_{k-1}$.
- $\mathbf{B}^{(k-1)}$ depends on $\mathbf{B}^{(k-2)}$, $\mathbf{y}_{k-2}$ and $\mathbf{s}_{k-2}$.
- and so on...
- So define $\mathbf{B}^{(k-1)}$ implicitly in terms of $\mathbf{B}^{(k-1)}$, $\mathbf{y}_{k-2}$ and $\mathbf{s}_{k-2}$.
- We continue up to $\mathbf{B}^{(k-m)}$, which is initialized to be $\gamma\mathbb{I}$.
- These are used to implicitly do $\mathbf{B}^{(k)}$–vector products.
- Linear rate of convergence

# L-BFGS

**Curvature Condition:** $\left\langle \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}), \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\rangle > 0$

- If $f(\mathbf{x})$ is (strictly) convex $\Longrightarrow$ ✓
- Otherwise, (strong) Wolfe-condition on $\alpha$ (nonlinear inequality)

$$\langle \nabla f(\mathbf{x} + \alpha \mathbf{p}), \mathbf{p} \rangle \geq \beta \langle \nabla f(\mathbf{x}), \mathbf{p} \rangle, \ \ \beta < 1$$

- When $\mathbf{g} \approx \nabla f \Longrightarrow$ noisy curvature estimate $\Longrightarrow$ many issues arise!

# L-BFGS [Byrd et al., 2014]

- Decoupling of the stochastic gradient and curvature estimations $\Longrightarrow$ different sample subsets for estimating $\mathbf{y}_k$
- $\mathbf{s}_k = \bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k-1}$ where $\bar{\mathbf{x}}_k = \frac{1}{L}\sum_{j=k-L+1}^{k} \mathbf{x}^{(j)}$.
- $\mathbf{y}_k = \nabla^2 f_{\mathcal{S}_\mathbf{H}}(\bar{\mathbf{x}}_k)\mathbf{s}_k \approx \nabla f_{\mathcal{S}_\mathbf{H}}(\bar{\mathbf{x}}_k) - \nabla f_{\mathcal{S}_\mathbf{H}}(\bar{\mathbf{x}}_{k-1})$.
- Update $\mathbf{H}^{(k)}$ every $L \geq 1$ iterations
- Under strong convexity, they show that $0 \prec \mu_1 \mathbf{I} \preceq \mathbf{H}^{(k)} \preceq \mu_2 \mathbf{I}$
- Setting $\alpha_k \propto 1/k$, they show

$$\mathbb{E}(f(\mathbf{x}^{(k)}) - f(\mathbf{x}^\star)) \leq C/k$$

# L-BFGS [Mokhtari and Ribeiro, 2014]

- $\mathbf{s}_k = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$
- Enforce gradient consistency, i.e., use the samples $\mathcal{S}$: $\mathbf{y}_k = \nabla f_{\mathcal{S}}(\mathbf{x}^{(k+1)}) - \nabla f_{\mathcal{S}}(\mathbf{x}^{(k)})$.
- For some $\delta > 0$, $\widehat{\mathbf{y}}_k = \mathbf{y}_k - \delta \mathbf{s}_k$
- Update $\mathbf{B}_{k+1}$ as in the usual case with $\mathbf{s}_k$ and $\hat{\mathbf{y}}_k$
- Add regularization: $\widehat{\mathbf{B}}_{k+1} = \mathbf{B}_{k+1} + m\mathbf{I}$
- Add regularization again: $\left(\widehat{\widehat{\mathbf{B}}}_{k+1}\right)^{-1} = \left(\widehat{\mathbf{B}}_{k+1}\right)^{-1} + M\mathbf{I}$
    - Spectrum of $\widehat{\widehat{\mathbf{B}}}_{k+1}$ is bounded above and away from zero
- Strong convexity of each $f_i$
- $\delta < \min_{\mathbf{x}} \lambda_{\min}(\nabla^2 f_i(\mathbf{x})) \Longrightarrow$ curvature condition holds
- Setting $\alpha_k \propto 1/k$, they show

$$\mathbb{E}(f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{\star})) \leq C/k$$

# L-BFGS [Moritz et al., 2015]

- Combine the ideas of [Byrd et al., 2014] with variance reduction of [Johnson and Zhang, 2013]
  - **Recall SVRG**: For $s$ and $k$, inner and outer iteration counters, respectively, estimate the gradient as

$$\mathbf{g}^{(s)} = \left(\nabla f_j(\mathbf{x}^{(s)}) - \nabla f_j(\mathbf{x}^{(k)}) + \nabla F(\mathbf{x}^{(k)})\right)$$

  - No need to diminish step-size any more!

- Under strong convexity:

$$\mathbb{E}(f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{\star})) \leq \rho^k \mathbb{E}(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{\star})), \quad \rho < 1$$

- As in SVRG, convergence is with respect to the outer iterations

# L-BFGS [Berahas et al., 2016, Berahas and Takáč, 2017]

- Idea: Perform QN update on overlapping consecutive batches
- Idea: $\mathcal{T}_k = \mathcal{S}_k \cap \mathcal{S}_{k+1} \neq \emptyset$
- $\mathbf{y}_k = \nabla f_{\mathcal{O}_k}(\mathbf{x}^{(k+1)}) - \nabla f_{\mathcal{O}_k}(\mathbf{x}^{(k)})$
- Using constant step-size $\alpha$
  - Strongly convex:

$$\mathbb{E}(f(\mathbf{x}^{(k)}) - f(\mathbf{x}^\star)) \leq \rho^k \left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^\star)\right) + \mathcal{O}(\alpha)$$

  - Non-convex: Skip updating $\mathbf{H}^{(k)}$ if $\mathbf{y}_k^T \mathbf{s}_k \leq \epsilon \left\|\mathbf{s}_k\right\|^2$

$$\mathbb{E}\left(\frac{1}{T}\sum_{k=0}^{T-1}\left\|\nabla f(\mathbf{x}^{(k)})\right\|^2\right) \leq \mathcal{O}\left(\frac{1}{T\alpha}\right) + \mathcal{O}(\alpha)$$

$$\text{If } \alpha \leq \mathcal{O}(1/\sqrt{T}) \Longrightarrow \min_{k \leq T-1} \mathbb{E}\left\|\nabla f(\mathbf{x}^{(k)})\right\|^2 \leq \mathcal{O}\sqrt{\frac{1}{T}}$$

# Outline

- Part I: Convex
  - Smooth
    - Newton-CG
  - Non-Smooth
    - Proximal Newton
    - Semi-smooth Newton
- Part II: Non-Convex
  - Line-Search Based Methods
    - L-BFGS
    - Gauss-Newton
    - Natural Gradient
  - Trust-Region Based Methods
    - Trust-Region
    - Cubic Regularization
- Part III: Discussion and Examples

# Gauss-Newton

## Problem

$$\min_{\mathbf{x} \in \mathcal{R}^d} F(\mathbf{x}) = f(\mathbf{h}(\mathbf{x}))$$

- $\mathbf{h} : \mathcal{R}^d \to \mathcal{R}^p$
- $f : \mathcal{R}^p \to \mathcal{R}$, and convex

# Gauss-Newton

Let $\mathbf{J_h} : \mathcal{R}^d \to \mathcal{R}^p$ be the Jacobian of $\mathbf{h}$, i.e., $\mathbf{J_h}(\mathbf{x}) \in \mathcal{R}^{p \times d}$.

$$\nabla F(\mathbf{x}) = \mathbf{J}_\mathbf{h}^T(\mathbf{x}) \nabla f(\mathbf{h}(\mathbf{x}))$$

$$\nabla^2 F(\mathbf{x}) = \mathbf{J}_\mathbf{h}^T(\mathbf{x}) \nabla^2 f(\mathbf{h}(\mathbf{x})) \mathbf{J}_\mathbf{h}(\mathbf{x}) + \partial^2 \mathbf{h}(\mathbf{x}) \nabla f(\mathbf{h}(\mathbf{x}))$$

(Generalized) Gauss-Newton Matrix:

$$\nabla^2 F(\mathbf{x}) \approx \underbrace{\mathbf{J}_\mathbf{h}^T(\mathbf{x}) \nabla^2 f(\mathbf{h}(\mathbf{x})) \mathbf{J}_\mathbf{h}(\mathbf{x})}_{\mathbf{G}(\mathbf{x}) \triangleq \text{ Gauss-Newton Matrix}} \succeq 0$$

(Generalized) Gauss-Newton Update:

$$\mathbf{G}(\mathbf{x}^{(k)}) \mathbf{p} \approx -\nabla F(\mathbf{x}^{(k)})$$

# Gauss-Newton

**Another interpretation:**

$$f(\mathbf{h}(\mathbf{x})) \approx f\left(\mathbf{h}(\mathbf{x}^{(k)}) + \mathbf{J_h}(\mathbf{x}^{(k)})\left(\mathbf{x} - \mathbf{x}^{(k)}\right)\right) \triangleq \ell(\mathbf{x}; \mathbf{x}^{(k)})$$

$$\nabla f(\mathbf{h}(\mathbf{x}^{(k)})) = \nabla \ell(\mathbf{x}^{(k)}; \mathbf{x}^{(k)}) = \mathbf{J_h}^T(\mathbf{x}^{(k)}) \nabla f(\mathbf{h}(\mathbf{x}^{(k)}))$$

$$\nabla^2 f(\mathbf{h}(\mathbf{x}^{(k)})) \approx \nabla^2 \ell(\mathbf{x}^{(k)}; \mathbf{x}^{(k)}) = \mathbf{J_h}^T(\mathbf{x}^{(k)}) \nabla^2 f(\mathbf{h}(\mathbf{x}^{(k)})) \mathbf{J_h}(\mathbf{x}^{(k)}) = \mathbf{G}(\mathbf{x}^{(k)})$$

# Gauss-Newton

## (Generalized) Gauss-Newton Matrix

$$\nabla^2 F(\mathbf{x}) \approx \mathbf{J}_{\mathbf{h}}^T(\mathbf{x}) \nabla^2 f(\mathbf{h}(\mathbf{x})) \mathbf{J}_{\mathbf{h}}(\mathbf{x})$$

**Properties:**

- $\mathbf{G}(\mathbf{x}) \succeq 0, \ \forall \mathbf{x}$
- In some applications, after computing $\nabla F(\mathbf{x}) = \mathbf{J}_{\mathbf{h}}^T(\mathbf{x}) \nabla f(\mathbf{h}(\mathbf{x}))$, the approximation $\mathbf{G}(\mathbf{x})$ does not involve any additional derivative evaluations
- $\mathbf{G}(\mathbf{x})$ is a good approximation if $\|\partial^2 \mathbf{h}(\mathbf{x}) \nabla f(\mathbf{h}(\mathbf{x}))\|$ is small, i.e.,
  - $\|\nabla f(\mathbf{h}(\mathbf{x}))\|$ is small, or
  - $\|\partial^2 \mathbf{h}(\mathbf{x})\|$ is small, i.e., $\mathbf{h}$ is nearly affine

# Gauss-Newton

## Gauss-Newton Convergence

Under some regularity assumptions:

- Damped Gauss-Newton is globally convergent, i.e., $\lim_{k\to\infty} \left\|\nabla F(\mathbf{x}^{(k)})\right\| = 0$
- The rate of convergence can be shown to be linear
- Local convergence ($\mathbf{S}(\mathbf{x}) \triangleq \partial^2\mathbf{h}(\mathbf{x}^\star)\nabla f(\mathbf{h}(\mathbf{x}^\star))$):

$$\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\right\| \leq \|\mathbf{G}(\mathbf{x}^\star)\| \, \|\mathbf{S}(\mathbf{x}^\star)\| \left\|\mathbf{x}^{(k)} - \mathbf{x}^\star\right\| + \mathcal{O}\left(\left\|\mathbf{x}^{(k)} - \mathbf{x}^\star\right\|^2\right),$$

# Gauss-Newton

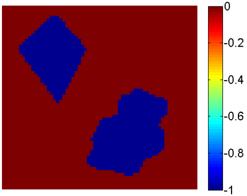
## Finite-Sum Problem

$$\min_{\mathbf{x}\in\mathcal{R}^d} F(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{h}_i(\mathbf{x}))$$

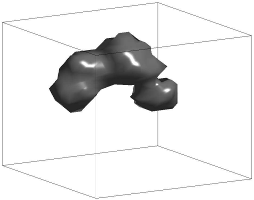
- Machine Learning (e.g., deep/recurrent/reinforcement learning): [Martens, 2010, Martens and Sutskever, 2011, Chapelle and Erhan, 2011, Wu et al., 2017, Botev et al., 2017]... more on this later
- Scientific Computing (e.g., PDE inverse-problems): [Doel and Ascher, 2012, Roosta et al., 2014b, Roosta et al., 2014a, Roosta et al., 2015, Haber et al., 2000, Haber et al., 2012]

# PDE Inverse Problems with Many R.H.S

$$\left.\begin{aligned} \nabla \cdot (x(\mathbf{z}) \nabla u_i(\mathbf{z})) &= q_i(\mathbf{z}), & \mathbf{z} \in \Omega \\ \frac{\partial u_i(\mathbf{z})}{\partial \nu} &= 0, & \mathbf{z} \in \partial\Omega \end{aligned}\right\}, \; i = 1, \ldots, n, \;\; \Omega \subset \mathcal{R}^2 \text{ or } \mathcal{R}^3$$




(a) True $x$: 2D



(b) True $x$: 3D

# Forward Problem

**Discretize-Then-Optimize**

$$\mathbf{v}_i = P_i A^{-1}(\mathbf{x})\mathbf{q}_i + \epsilon_i, \quad i = 1, 2, \ldots n, \ \mathbf{x} \in \mathcal{R}^d$$

- $n$: No. of measurements
- $d$: Mesh size

# Inverse Problem

$$\epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_i)$$

$$\Downarrow$$

$$f_i(\mathbf{h}_i(\mathbf{x})) = \| \underbrace{\mathbf{\Sigma}_i^{-\frac{1}{2}} \left( \mathbf{P}_i \mathbf{A}^{-1}(\mathbf{x}) \mathbf{q}_i - \mathbf{v}_i \right)}_{\mathbf{h}_i(\mathbf{x})} \|^2$$

$$\Downarrow$$

$$\min_{\mathbf{x}} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \| \mathbf{\Sigma}_i^{-\frac{1}{2}} \left( \mathbf{P}_i \mathbf{A}^{-1}(\mathbf{x}) \mathbf{q}_i - \mathbf{v}_i \right) \|^2$$

Calculating "$\mathbf{A}^{-1}(\mathbf{x})\mathbf{q}_i$" for each $i$ is costly!

# A remedy: SAA

$$\mathcal{S} \subset [n] \;\&\; |\mathcal{S}| = s$$
$$\Downarrow$$
$$F(\mathbf{x}) \approx \hat{F}_s(\mathbf{x}) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \| \mathbf{\Sigma}_i^{-\frac{1}{2}} \left( \mathbf{P}_i \mathbf{A}^{-1}(\mathbf{x}) \mathbf{q}_i - \mathbf{v}_i \right) \|_2^2$$

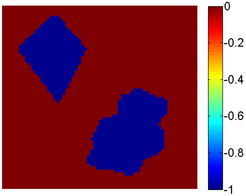
**Trace estimation**: [Roosta and Ascher, 2015, Roosta et al., 2015]

Find $s$ such that, for a given $\epsilon$ and $\delta$, we get

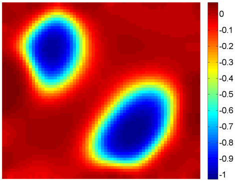
$$\mathbf{Pr}\left( |\hat{F}_s(\mathbf{x}) - F(\mathbf{x})| \le \epsilon F(\mathbf{x}) \right) \ge 1 - \delta$$

# $n = 961$, Noise 1%, $\sigma_1 = 0.1$, $\sigma_2 = 1$

| **Method** | Vanilla | Sub-Sampled |
|---|---|---|
| **PDE Solves** | 128,774 | 3,921 |



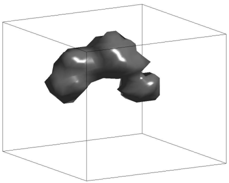

(c) True Model



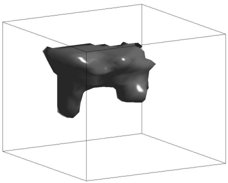

(d) Sub-Sampled GN

# $n = 512$, Noise 2% $\sigma_I = 1$, $\sigma_{II} = .1$

| **Method** | Vanilla | Sub-Sampled |
|---|---|---|
| **PDE Solves** | 45,056 | 2,264 |



(e) True Model



(f) Sub-Sampled GN

# Outline

- Part I: Convex
  - Smooth
    - Newton-CG
  - Non-Smooth
    - Proximal Newton
    - Semi-smooth Newton
- Part II: Non-Convex
  - Line-Search Based Methods
    - L-BFGS
    - Gauss-Newton
    - Natural Gradient
  - Trust-Region Based Methods
    - Trust-Region
    - Cubic Regularization
- Part III: Discussion and Examples

# Natural Gradient

## Cross Entropy Minimization

For $p_{\mathbf{x}}(\mathbf{z})$, a density parametrized by $\mathbf{x}$, the **cross-entropy minimization** with respect to a target density, $p_{\mathbf{x}}(\mathbf{z})$, is

$$\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}) = -\mathbb{E}_{\mathbf{x}}\left(\log p_{\mathbf{x}}(\mathbb{z})\right) = -\int p_{\mathbf{x}}(\mathbf{z}) \log p_{\mathbf{x}}(\mathbf{z}) \, \mathrm{d}\boldsymbol{\mu}(\mathbf{z}).$$

**NB:** $p_{\mathbf{x}}(\mathbf{z})\mathrm{d}\boldsymbol{\mu}(\mathbf{z})$ can be the empirical measure over the training data.

## Fisher Information Matrix

Suppose $\mathbb{z} \sim p_{\mathbf{x}}$. Under some weak regularity assumptions:

$$\mathbf{F}(\mathbf{x}) \triangleq \mathbb{E}_{\mathbf{x}}\left(\nabla \log p_{\mathbf{x}}(\mathbb{z}) \left(\nabla \log p_{\mathbf{x}}(\mathbb{z})\right)^T\right) = -\mathbb{E}_{\mathbf{x}}\left(\nabla^2 \log p_{\mathbf{x}}(\mathbb{z})\right).$$

## Natural Gradient Descent

$$\mathbf{F}(\mathbf{x}^{(k)})\mathbf{p}^{(k)} \approx \mathbf{g}^{(k)} \implies \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}$$

# Natural Gradient

**Interpretation I:**

$$\min_{\mathbf{x} \in \mathcal{X}} \mathcal{L}(\mathbf{x}) = -\mathbb{E}_{\mathbf{x}} \left( \log p_{\mathbf{x}}(\mathbb{z}) \right)$$

## Natural Gradient vs. Newton's Method

For a given $\mathbf{x}$:

$$\begin{aligned} \text{Hessian Matrix:} \quad \nabla^2 \mathcal{L}(\mathbf{x}) &= -\mathbb{E}_{\mathbf{x}} \left( \nabla^2 \log p_{\mathbf{x}}(\mathbb{z}) \right) \\ \text{Fisher Matrix:} \quad \mathbf{F}(\mathbf{x}) &= -\mathbb{E}_{\mathbf{x}} \left( \nabla^2 \log p_{\mathbf{x}}(\mathbb{z}) \right) \end{aligned}$$

# Natural Gradient

**Interpretation I:**

Let $p_{\mathbf{x}}$ be the empirical measure over the given training set $\{\mathbf{z}_i\}_{i=1}^{n}$ where $\mathbf{z}_i \sim p_{\mathbf{x}^\star}$ for some true, but unknown, parameter $\mathbf{x}^\star$, i.e., empirical risk minimization:

$$\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \log p_{\mathbf{x}}(\mathbf{z}_i)$$

## Approximation I: Natural Gradient vs. Newton's Method

For a given $\mathbf{x}$:

$$\text{Hessian:} \quad \nabla^2 \mathcal{L}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log p_{\mathbf{x}}(\mathbf{z}_i), \quad \mathbf{z}_i \sim p_{\mathbf{x}^\star}$$

$$\text{Approximate Fisher:} \quad \hat{\mathbf{F}}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \nabla^2 \log p_{\mathbf{x}}(\mathbf{z}_i), \quad \mathbf{z}_i \sim p_{\mathbf{x}}$$

# Natural Gradient

**Interpretation II:**

$$\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \log p_{\mathbf{x}}(\mathbf{z}_i)$$

In Gauss-Newton, we had $\mathcal{L}(\mathbf{x}) = f(h(\mathbf{x}))$. Here, we can consider $f(t) = -\log t \in \mathcal{R}$ and $h(\mathbf{x}) = p_{\mathbf{x}}(\mathbf{z}) \in \mathcal{R}$. So,

$$\mathbf{G}(\mathbf{x}) = f''(h(\mathbf{x}))\nabla h(\mathbf{x})\nabla h(\mathbf{x})^T = \frac{1}{(p_{\mathbf{x}}(\mathbf{z}))^2}\nabla p_{\mathbf{x}}(\mathbf{z})\left(\nabla p_{\mathbf{x}}(\mathbf{z})\right)^T$$

$$= \left(\frac{1}{p_{\mathbf{x}}(\mathbf{z})}\nabla p_{\mathbf{x}}(\mathbf{z})\right)\left(\frac{1}{p_{\mathbf{x}}(\mathbf{z})}\nabla p_{\mathbf{x}}(\mathbf{z})\right)^T = \nabla \log p_{\mathbf{x}}(\mathbf{z})\left(\nabla \log p_{\mathbf{x}}(\mathbf{z})\right)^T$$

**Approximation II: Natural Gradient vs. Gauss-Newton**

$$\mathbf{G}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \nabla \log p_{\mathbf{x}}(\mathbf{z}_i)\left(\nabla \log p_{\mathbf{x}}(\mathbf{z}_i)\right)^T, \quad \mathbf{z}_i \sim p_{\mathbf{x}^\star}$$

$$\hat{\mathbf{F}}(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \nabla \log p_{\mathbf{x}}(\mathbf{z}_i)\left(\nabla \log p_{\mathbf{x}}(\mathbf{z}_i)\right)^T, \quad \mathbf{z}_i \sim p_{\mathbf{x}}$$

# Natural Gradient

**Interpretation III:**
More generally, consider fitting probabilistic models

$$\min_{\mathbf{x} \in \mathcal{R}^d} \mathcal{L}(\mathbf{x}) = L(p_{\mathbf{x}})$$

**Recall:** steepest descent in Euclidean space

Ideally, we want

$$\mathbf{p}^\star = \operatorname*{argmin}_{\|\mathbf{p}\| \leq 1} \mathcal{L}(\mathbf{x} + \mathbf{p}),$$

but it is easier to do

$$\frac{-\nabla \mathcal{L}(\mathbf{x})}{\|\nabla \mathcal{L}(\mathbf{x})\|} = \operatorname*{argmin}_{\|\mathbf{p}\| \leq 1} \langle \nabla \mathcal{L}(\mathbf{x}), \mathbf{p} \rangle$$

# Natural Gradient

**Interpretation III:**

**KullbackLeibler distance**

For given $\mathbf{x}$ and $\mathbf{x}$, the Kullback-Leibler distance from $p_{\mathbf{x}}$ to $p_{\mathbf{x}}$ is

$$\mathbf{KL}(\mathbf{x} \parallel \mathbf{x}) \triangleq \mathbb{E}_{\mathbf{x}}\left(\log \frac{p_{\mathbf{x}}(\mathbf{z})}{p_{\mathbf{x}}(\mathbf{z})}\right) = \int \left(\log \frac{p_{\mathbf{x}}(\mathbf{z})}{p_{\mathbf{x}}(\mathbf{z})}\right) p_{\mathbf{x}}(\mathbf{z})\, \mathrm{d}\boldsymbol{\mu}(\mathbf{z}).$$

$$\mathbf{F}(\mathbf{x}) = \nabla^2_{\mathbf{x}}\, \mathbf{KL}(\mathbf{x} \parallel \mathbf{x})|_{\mathbf{x}=\mathbf{x}}$$

If $\mathbf{F}(\mathbf{x}) \succ 0$, then in a neighborhood of $\mathbf{x}$, we have $\mathbf{KL}(\mathbf{x} \parallel \mathbf{x}) > 0$, and

$$\mathbf{KL}(\mathbf{x} \parallel \mathbf{x}) \approx \frac{1}{2}(\mathbf{x} - \mathbf{x})^2\, \mathbf{F}(\mathbf{x})\,(\mathbf{x} - \mathbf{x})$$

# Natural Gradient

**Interpretation III:**

Ideally, we want

$$\mathbf{p}^\star = \underset{\mathbf{KL}(\mathbf{x}\|\mathbf{x}+\mathbf{p})\le 1}{\operatorname{argmin}} \mathcal{L}(\mathbf{x}+\mathbf{p})$$

But, if $\mathbf{p} \ll 1$, we can approximate

$$\mathbf{F}^{-1}(\mathbf{x})\nabla\mathcal{L}(\mathbf{x}) \propto \underset{\mathbf{p}^T\mathbf{F}(\mathbf{x})\mathbf{p}\le 1}{\operatorname{argmin}} \langle \nabla\mathcal{L}(\mathbf{x}), \mathbf{p}\rangle$$

# Natural Gradient

- Classical: [Amari, 1998]
- Overview: [Martens, 2014]
- On manifolds: [Song and Ermon, 2018]
- Deep learning: [Pascanu and Bengio, 2013, Martens and Grosse, 2015, Grosse and Salakhudinov, 2015]

# Outline

- Part I: Convex
  - Smooth
    - Newton-CG
  - Non-Smooth
    - Proximal Newton
    - Semi-smooth Newton
- Part II: Non-Convex
  - Line-Search Based Methods
    - L-BFGS
    - Gauss-Newton
    - Natural Gradient
  - Trust-Region Based Methods
    - Trust-Region
    - Cubic Regularization
- Part III: Discussion and Examples

# Problem Statement

## Minimizing Finite Sum Problem

$$\min_{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x})$$

- $f_i$: (non-)convex and smooth
- $n \gg 1$ and/or $d \gg 1$

- Trust Region: [Sorensen, 1982, Conn et al., 2000]

$$\mathbf{s}^{(k)} = \arg\min_{\|\mathbf{s}\| \leq \Delta_k} \left\langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2} \left\langle \mathbf{s}, \nabla^2 F(\mathbf{x}^{(k)}) \mathbf{s} \right\rangle$$

- Cubic Regularization: [Griewank, 1981, Nesterov et al., 2006, Cartis et al., 2011a, Cartis et al., 2011b]

$$\mathbf{s}^{(k)} = \arg\min_{\mathbf{s} \in \mathbb{R}^d} \left\langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2} \left\langle \mathbf{s}, \nabla^2 F(\mathbf{x}^{(k)}) \mathbf{s} \right\rangle + \frac{\sigma_k}{3} \|\mathbf{s}\|^3$$

- Trust Region:

$$\mathbf{s}^{(k)} = \arg\min_{\|\mathbf{s}\| \leq \Delta_k} \left\langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2} \left\langle \mathbf{s}, \nabla^2 F(\mathbf{x}^{(k)}) \mathbf{s} \right\rangle$$

- Cubic Regularization:

$$\mathbf{s}^{(k)} = \arg\min_{\mathbf{s} \in \mathbb{R}^d} \left\langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2} \left\langle \mathbf{s}, \nabla^2 F(\mathbf{x}^{(k)}) \mathbf{s} \right\rangle + \frac{\sigma_k}{3} \|\mathbf{s}\|^3$$

- Trust Region:

$$\mathbf{s}^{(k)} = \arg\min_{\|\mathbf{s}\| \leq \Delta_k} \left\langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2} \left\langle \mathbf{s}, \mathbf{H}^{(k)} \mathbf{s} \right\rangle$$

- Cubic Regularization:

$$\mathbf{s}^{(k)} = \arg\min_{\mathbf{s} \in \mathbb{R}^d} \left\langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2} \left\langle \mathbf{s}, \mathbf{H}^{(k)} \mathbf{s} \right\rangle + \frac{\sigma_k}{3} \|\mathbf{s}\|^3$$

- To get iteration complexity, previous work required:

$$\left\| \left( \mathbf{H}^{(k)} - \nabla^2 F(\mathbf{x}^{(k)}) \right) \mathbf{s}^{(k)} \right\| \leq C \|\mathbf{s}^{(k)}\|^2 \qquad (1)$$

- Stronger than "Dennis-Moré"

$$\lim_{k \to \infty} \frac{\| \left( H(\mathbf{x}(k)) - \nabla^2 F(\mathbf{x}(k)) \right) \mathbf{s}(k) \|}{\|\mathbf{s}(k)\|} = 0$$

- We relaxed (1) to

$$\left\| \left( \mathbf{H}^{(k)} - \nabla^2 F(\mathbf{x}^{(k)}) \right) \mathbf{s}^{(k)} \right\| \leq \epsilon \|\mathbf{s}^{(k)}\| \qquad (2)$$

- Quasi-Newton, Sketching, Sub-Sampling satisfy Dennis-Moré and (2) but not necessarily (1)

$$\left\| H(\mathbf{x}) - \nabla^2 F(\mathbf{x}) \right\| \leq \epsilon \quad \Longrightarrow \quad \left\| \left( H(\mathbf{x}) - \nabla^2 F(\mathbf{x}) \right) \mathbf{s} \right\| \leq \epsilon \|\mathbf{s}\|$$

$$H(\mathbf{x}) = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla^2 f_j(\mathbf{x})$$

**Lemma (Uniform Sampling [Xu et al., 2017])**

*Suppose* $\|\nabla^2 f_i(\mathbf{x})\| \leq K_i, \; i = 1, 2, \ldots, n$. *Let* $K = \max_{i=1,\ldots,n} K_i$.
*Given any* $0 < \epsilon < 1$, $0 < \delta < 1$, *and* $\mathbf{x} \in \mathbb{R}^d$, *if*

$$|\mathcal{S}| \geq \frac{16K^2}{\epsilon^2} \log \frac{2d}{\delta},$$

*then for*

$$H(\mathbf{x}) = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla^2 f_j(\mathbf{x}),$$

*we have*

$$\mathbf{Pr}\Big( \|H(\mathbf{x}) - \nabla^2 F(\mathbf{x})\| \leq \epsilon \Big) \geq 1 - \delta.$$

- Only top eigenvalues/eigenvectors need to preserved.

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{a}_i^T\mathbf{x})$$

$$p_i = \frac{|f_i''(\mathbf{a}_i^T\mathbf{x})|\|\mathbf{a}_i\|_2^2}{\sum_{j=1}^{n}|f_j''(\mathbf{a}_j^T\mathbf{x})|\|\mathbf{a}_j\|_2^2}$$

$$H(\mathbf{x}) = \frac{1}{|\mathcal{S}|}\sum_{j\in S}\frac{1}{np_j}\nabla^2 f_j(\mathbf{x})$$

**Lemma (Non-Uniform Sampling [Xu et al., 2017])**

*Suppose* $\|\nabla^2 f_i(\mathbf{x})\| \le K_i$, $i = 1, 2, \ldots, n$. *Let* $\bar{K} = \frac{1}{n}\sum_{i=1}^{n} K_i$. *Given any* $0 < \epsilon < 1$, $0 < \delta < 1$, *and* $\mathbf{x} \in \mathbb{R}^d$, *if*

$$|\mathcal{S}| \ge \frac{16\bar{K}^2}{\epsilon^2} \log \frac{2d}{\delta},$$

*then for*

$$H(\mathbf{x}) = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \frac{1}{np_j} \nabla^2 f_j(\mathbf{x}),$$

*we have*

$$\mathbf{Pr}\Big(\|H(\mathbf{x}) - \nabla^2 F(\mathbf{x})\| \le \epsilon\Big) \ge 1 - \delta,$$

$$\frac{1}{n}\sum_{i=1}^{n} K_i \leq \max_{i=1,\ldots,n} K_i$$

**Theorem ( [Xu et al., 2017])**

*If $\epsilon \in \mathcal{O}(\epsilon_H)$, then Stochastic TR terminates after*

$$T \in \mathcal{O}\left(\max\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\}\right),$$

*iterations, upon which, with high probability, we have that*

$$\|\nabla F(\mathbf{x})\| \leq \epsilon_g, \quad \textit{and} \quad \lambda_{\min}(\nabla^2 F(\mathbf{x})) \geq -(\epsilon + \epsilon_H).$$

- This is tight! [Cartis et al., 2012]

**Theorem ( [Xu et al., 2017])**

*If $\epsilon \in \mathcal{O}(\epsilon_g, \epsilon_H)$, then Stochastic ARC terminates after*

$$T \in \mathcal{O}\left(\max\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\}\right),$$

*iterations, upon which, with high probability, we have that*

$$\|\nabla F(\mathbf{x})\| \leq \epsilon_g, \quad \textit{and} \quad \lambda_{\min}(\nabla^2 F(\mathbf{x})) \geq -(\epsilon + \epsilon_H).$$

- This is tight! [Cartis et al., 2012]

- For $\epsilon_H^2 = \epsilon_g = \epsilon$
  - Stochastic TR: $T \in \mathcal{O}(\epsilon^{-2.5})$
  - Stochastic ARC: $T \in \mathcal{O}(\epsilon^{-1.5})$

# Outline

- Part I: Convex
  - Smooth
    - Newton-CG
  - Non-Smooth
    - Proximal Newton
    - Semi-smooth Newton
- Part II: Non-Convex
  - Line-Search Based Methods
    - L-BFGS
    - Gauss-Newton
    - Natural Gradient
  - Trust-Region Based Methods
    - Trust-Region
    - Cubic Regularization
- Part III: Discussion and Examples

# But why 1st Order Methods?

Q: But Why 1st Order Methods?

- Cheap Iterations
- Easy To Implement
- “Good” Worst-Case Complexities
- Good Generalization

# But why Not?2nd Order Methods

Q: But Why Not 2nd Order Methods?

- ~~Cheap~~ Expensive Iterations
- ~~Easy~~ Hard To Implement
- ~~"Good"~~ "Bad" Worst-Case Complexities
- ~~Good~~ Bad Generalization

# Our Goal...

Goal: Improve 2nd Order Methods...

- Cheap ~~Expensive~~ Iterations
- Easy ~~Hard~~ To Use
- "Good" ~~"Bad"~~ Average(?)-Case Complexities
- Good ~~Bad~~ Generalization

# Our Goal..

Any Other Advantages?

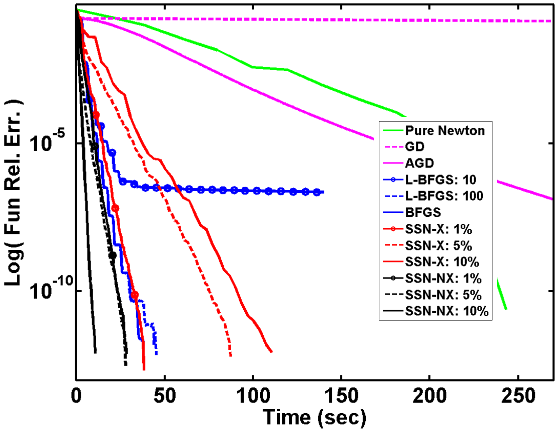
- Effective Iterations $\Rightarrow$ Less Iterations $\Rightarrow$ Less Communications
- Saddle Points For Non-Convex Problems
- Less Sensitive to Parameter Tuning
- Less Sensitive to Initialization

# Simulations: $\ell_2$ Regularized LR

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(\log\left(1+\exp(\mathbf{a}_i^T\mathbf{x})\right) - b_i\mathbf{a}_i^T\mathbf{x}\right) + \frac{\lambda}{2}\|\mathbf{x}\|^2$$

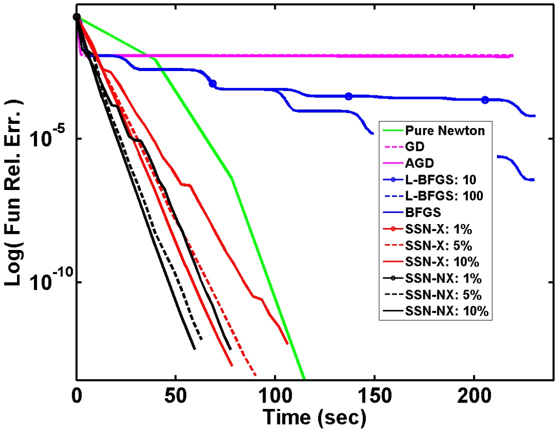
| Data | $n$ | $p$ | nnz | $\kappa(F)$ |
|---|---|---|---|---|
| $D_1$ | $10^6$ | $10^4$ | 0.02% | $\approx 10^4$ |
| $D_2$ | $5 \times 10^4$ | $5 \times 10^3$ | Dense | $\approx 10^6$ |
| $D_3$ | $10^7$ | $2 \times 10^4$ | 0.006% | $\approx 10^{10}$ |

# $D_1$, $n = 10^6$, $p = 10^4$, sparsity : 0.02%, $\kappa \approx 10^4$



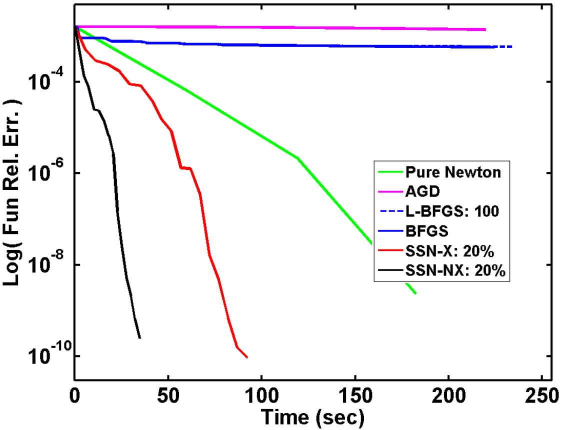

(g) Function Relative Error

## $D_2$, $n = 5 \times 10^4$, $p = 5 \times 10^3$, sparsity : Dense, $\kappa \approx 10^6$




(i) Function Relative Error

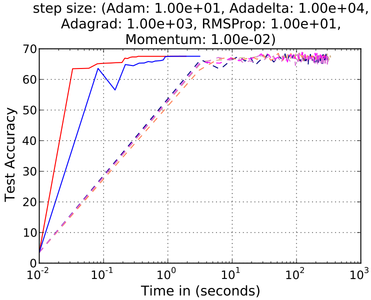
# $D_3$, $n = 10^7$, $p = 2 \times 10^4$, sparsity : 0.006%, $\kappa \approx 10^{10}$

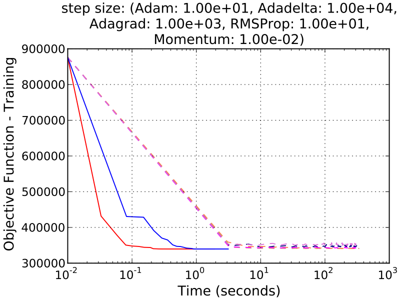



(k) Function Relative Error

# Newton GPU vs. TensorFlow

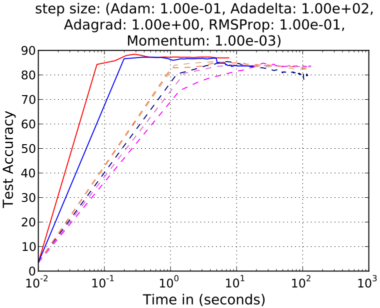
Data: Cover Type, $n = 4.5 \times 10^5$, $d = 378$

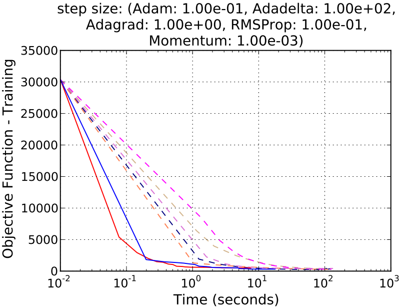


# Newton GPU vs. TensorFlow

Data: Newsgroup20, $n = 10^4, d = 10^6$



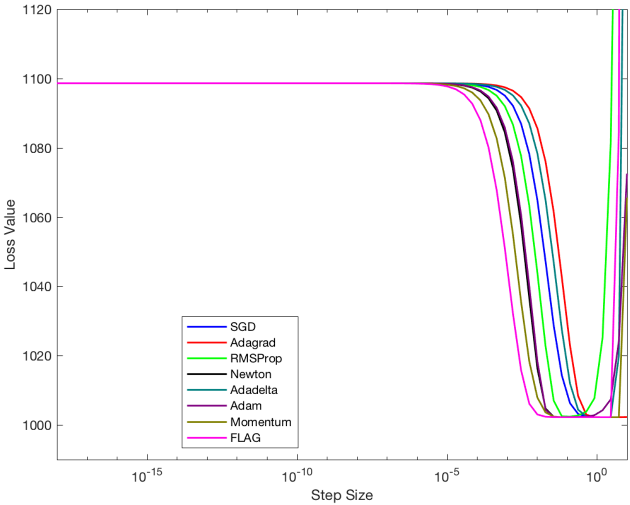



Figure: Skew Param $= 0$

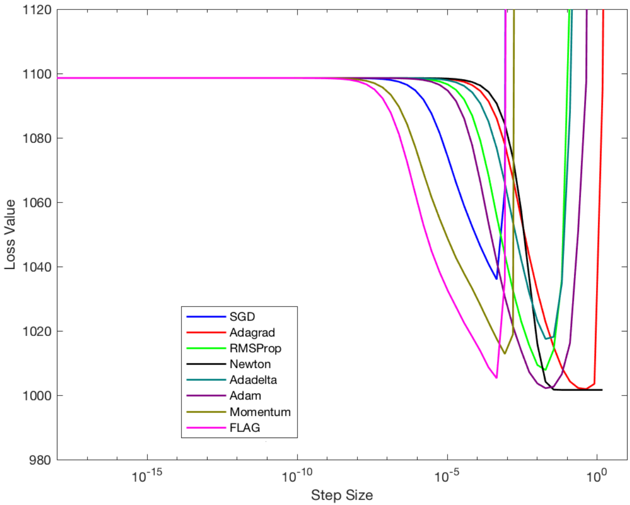



Figure: Skew Param $= 2$

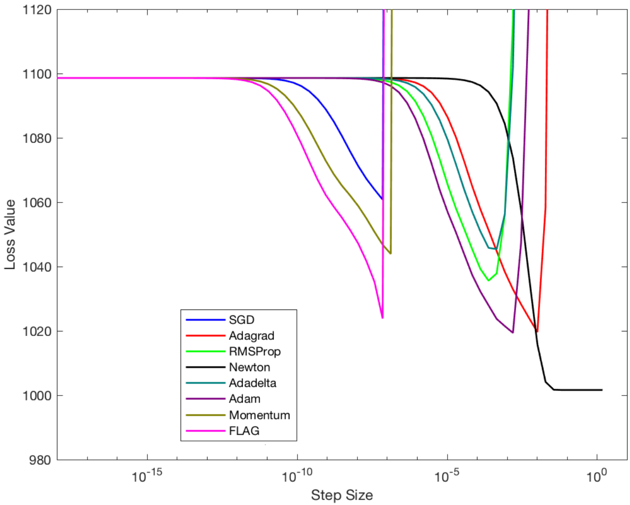



Figure: Skew Param = 4

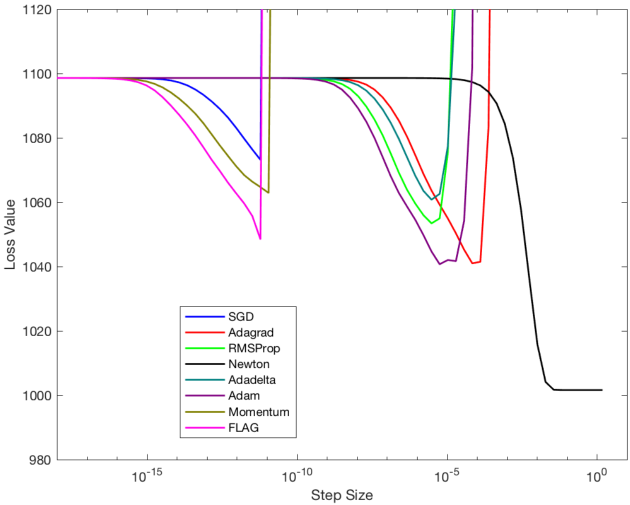



Figure: Skew Param = 6

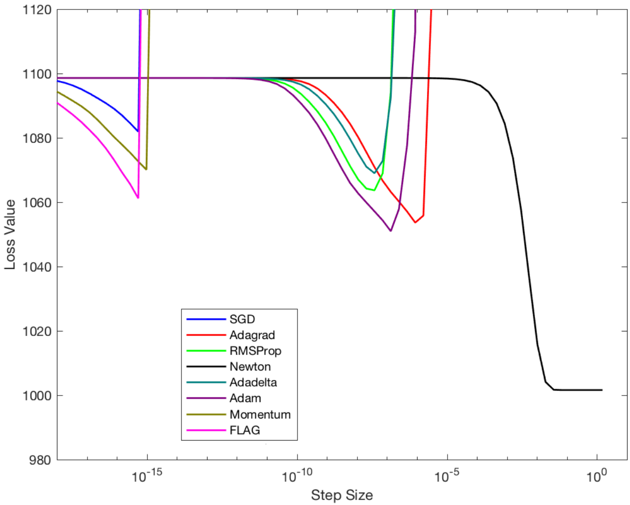



Figure: Skew Param = 8

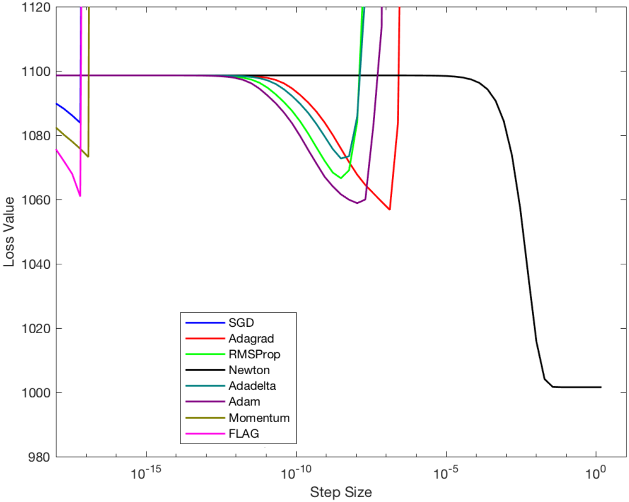



Figure: Skew Param = 9

# Numerical Examples: Deep Learning

| Dataset | Size | Network | (# parameters) |
|---|---|---|---|
| `curves` | $20,000$ | 784-400-200-100-50-25-6 | $842,340$ |
| `Cifar10` | $50,000$ | ResNet18 | $270,896$ |

# Deep Auto-Encoder

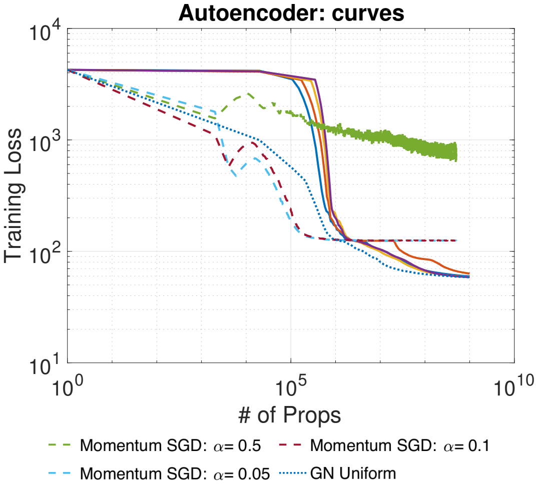



Figure: Random Initialization

# Deep Auto-Encoder

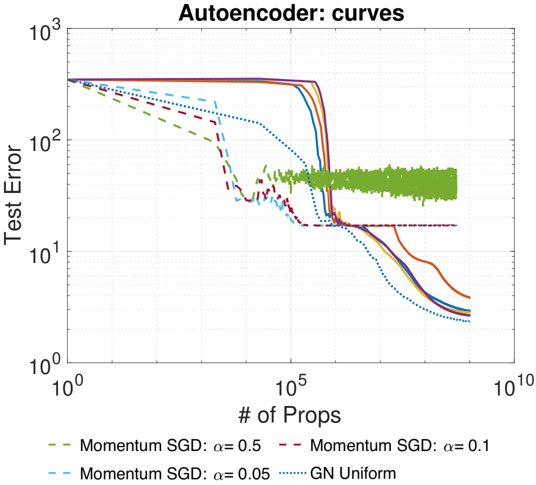



Figure: Random Initialization

# Deep Auto-Encoder

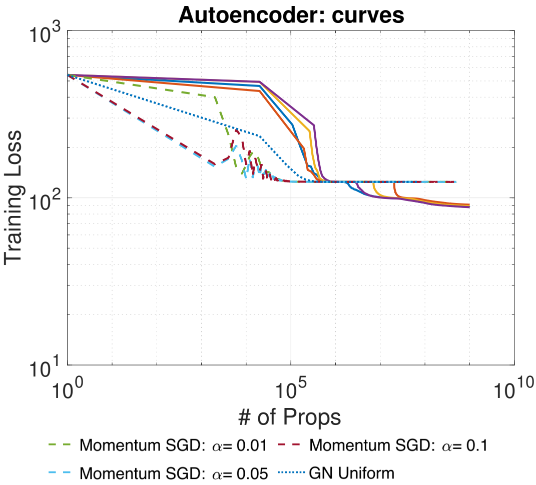



Figure: Zero Initialization

# Deep Auto-Encoder

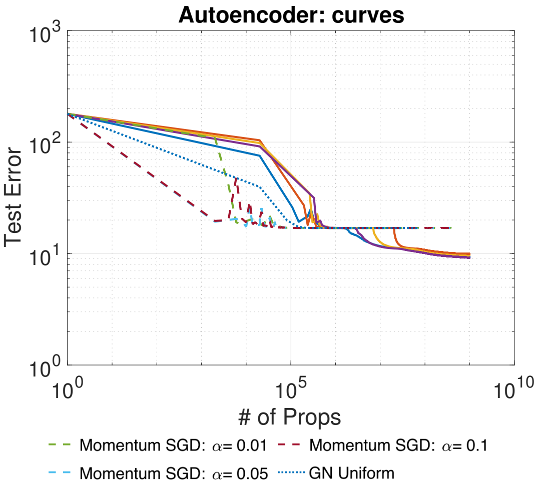



Figure: Zero Initialization

# Deep Auto-Encoder

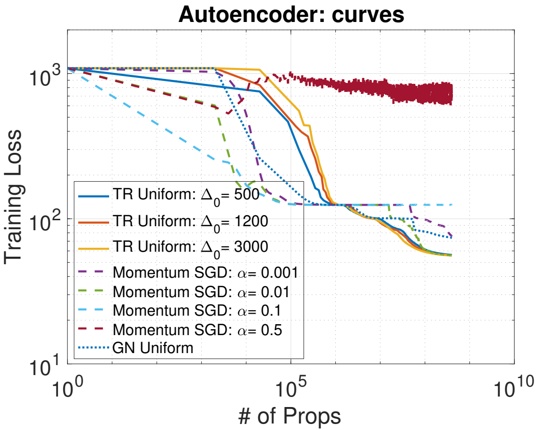



Figure: Scaled Random Initialization

# Deep Auto-Encoder

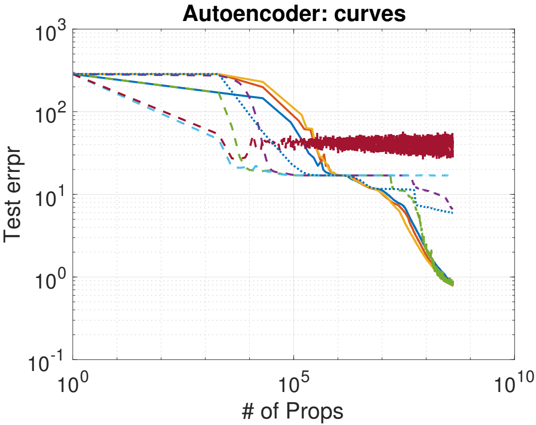
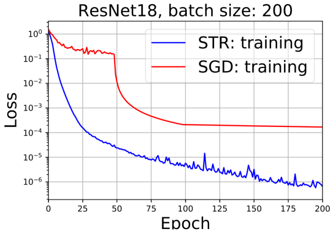
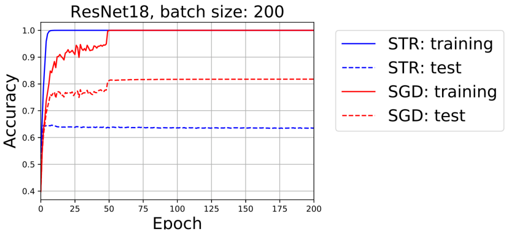



Figure: Scaled Random Initialization


Is it all rosie?

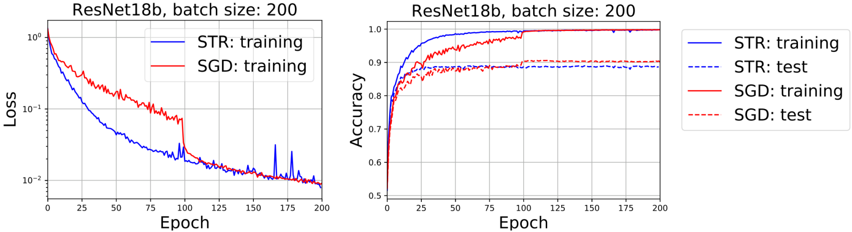
# ResNet18




No Batch Normalization + No data augmentation.

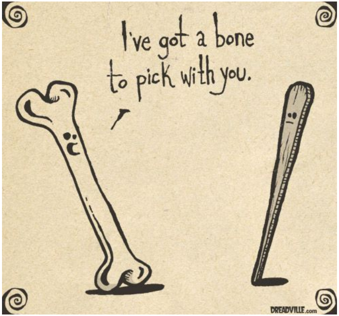
# ResNet18




Batch Normalization + Data augmentation.

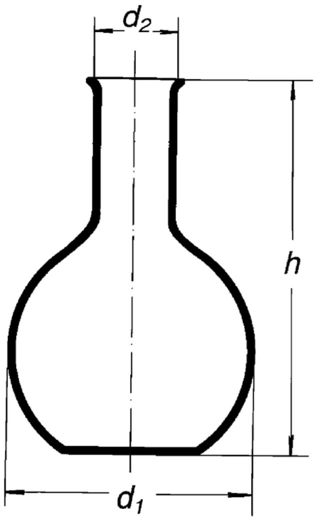
# Worst Case Complexity

My  to pick with worst case complexity results!!!

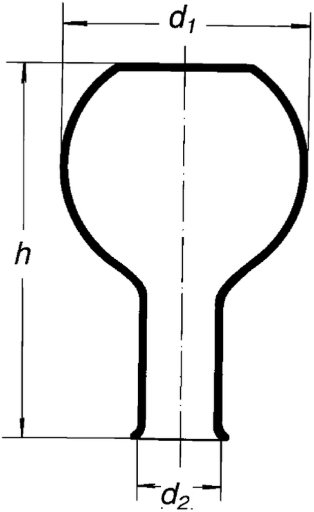


# Worst Case Complexity

# Worst Case Complexity



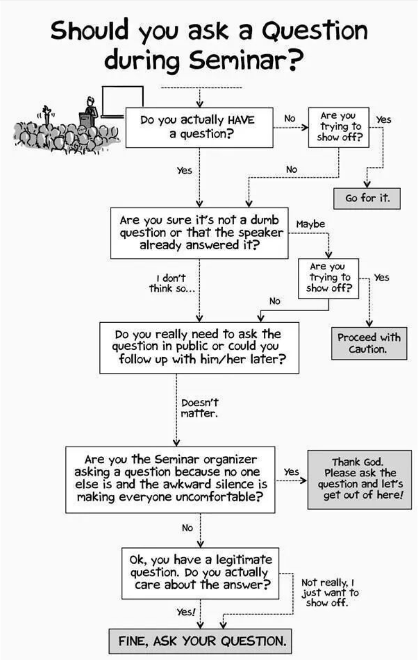
# Worst Case Complexity



# Worst Case Complexity

- **Q:** What do "**Newton's method**" and "**air travel**" have in common?

- **A:** Both are very **fast**, but their **worst-case** is bad!!!


Should you ask a Question during Seminar?
Do you actually HAVE a question?
No
Are you trying to show off?
Yes
Go for it.
No
Yes
Are you sure it's not a dumb question or that the speaker already answered it?
Maybe
Are you trying to show off?
Yes
Proceed with Caution.
No
I don't think so...
Do you really need to ask the question in public or could you follow up with him/her later?
Doesn't matter.
Are you the Seminar organizer asking a question because no one else is and the awkward silence is making everyone uncomfortable?
Yes
Thank God. Please ask the question and let's get out of here!
No
Ok, you have a legitimate question. Do you actually care about the answer?
Not really, I just want to show off.
Yes!
FINE, ASK YOUR QUESTION.

THANK YOU!

Amari, S.-I. (1998).
Natural gradient works efficiently in learning.
*Neural computation*, 10(2):251–276.

Berahas, A. S., Nocedal, J., and Takác, M. (2016).
A multi-batch l-bfgs method for machine learning.
In *Advances in Neural Information Processing Systems*, pages 1055–1063.

Berahas, A. S. and Takáč, M. (2017).
A Robust Multi-Batch L-BFGS Method for Machine Learning.
*arXiv preprint arXiv:1707.08552*.

Botev, A., Ritter, H., and Barber, D. (2017).
Practical Gauss-Newton optimisation for deep learning.
*arXiv preprint arXiv:1706.03662*.

Byrd, R. H., Hansen, S., Nocedal, J., and Singer, Y. (2014).
A stochastic quasi-Newton method for large-scale optimization.

*arXiv preprint arXiv:1401.7020*.

Cartis, C., Gould, N. I., and Toint, P. L. (2012).
Complexity bounds for second-order optimality in unconstrained optimization.
*Journal of Complexity*, 28(1):93–108.

Chapelle, O. and Erhan, D. (2011).
Improved preconditioner for Hessian free optimization.
In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, volume 201.

Dai, Y.-H. (2002).
Convergence properties of the bfgs algoritm.
*SIAM Journal on Optimization*, 13(3):693–701.

Doel, K. v. d. and Ascher, U. (2012).
Adaptive and stochastic algorithms for EIT and DC resistivity problems with piecewise constant solutions and many measurements.

*SIAM J. Scient. Comput.*, 34:DOI: 10.1137/110826692.

Grosse, R. and Salakhudinov, R. (2015).
Scaling up natural gradient by sparsely factorizing the inverse Fisher matrix.
In *International Conference on Machine Learning*, pages 2304–2313.

Haber, E., Ascher, U. M., and Oldenburg, D. (2000).
On optimization techniques for solving nonlinear inverse problems.
*Inverse problems*, 16(5):1263.

Haber, E., Chung, M., and Herrmann, F. (2012).
An effective method for parameter estimation with PDE constraints with multiple right-hand sides.
*SIAM Journal on Optimization*, 22(3):739–757.

Johnson, R. and Zhang, T. (2013).
Accelerating stochastic gradient descent using predictive variance reduction.

In *Advances in Neural Information Processing Systems*, pages 315–323.

Martens, J. (2010).
Deep learning via Hessian-free optimization.
In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742.

Martens, J. (2014).
New insights and perspectives on the natural gradient method.
*arXiv preprint arXiv:1412.1193*.

Martens, J. and Grosse, R. (2015).
Optimizing neural networks with kronecker-factored approximate curvature.
In *International conference on machine learning*, pages 2408–2417.

Martens, J. and Sutskever, I. (2011).
Learning recurrent neural networks with Hessian-free optimization.

In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040. Citeseer.

Mascarenhas, W. F. (2004).
The BFGS method with exact line searches fails for non-convex objective functions.
*Mathematical Programming*, 99(1):49–61.

Mokhtari, A. and Ribeiro, A. (2014).
Res: Regularized stochastic BFGS algorithm.
*Signal Processing, IEEE Transactions on*, 62(23):6089–6104.

Moritz, P., Nishihara, R., and Jordan, M. I. (2015).
A linearly-convergent stochastic L-BFGS algorithm.
*arXiv preprint arXiv:1508.02087*.

Pascanu, R. and Bengio, Y. (2013).
Revisiting natural gradient for deep networks.
*arXiv preprint arXiv:1301.3584*.

Roosta, F. and Ascher, U. (2015).

Improved bounds on sample size for implicit matrix trace estimators.
*Foundations of Computational Mathematics*, 15(5):1187–1212.

Roosta, F., Székely, G. J., and Ascher, U. (2015).
Assessing stochastic algorithms for large scale nonlinear least squares problems using extremal probabilities of linear combinations of gamma random variables.
*SIAM/ASA Journal on Uncertainty Quantification*, 3(1):61–90.

Roosta, F., van den Doel, K., and Ascher, U. (2014a).
Data completion and stochastic algorithms for PDE inversion problems with many measurements.
*Electronic Transactions on Numerical Analysis*, 42:177–196.

Roosta, F., van den Doel, K., and Ascher, U. (2014b).
Stochastic algorithms for inverse problems involving PDEs and many measurements.
*SIAM J. Scientific Computing*, 36(5):S3–S22.

Song, Y. and Ermon, S. (2018).
Accelerating Natural Gradient with Higher-Order Invariance.
*arXiv preprint arXiv:1803.01273*.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., and Ba, J. (2017).
Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation.
In *Advances in neural information processing systems*, pages 5279–5288.

Xu, P., Roosta, F., and Mahoney, M. W. (2017).
Newton-Type Methods for Non-Convex Optimization Under Inexact Hessian Information.
*arXiv preprint arXiv:1708.07164*.