SAND2018-7840C

Stochastic Hessian Projection

Jed A. Duersch & Thomas A. Catanach
Sandia National Laboratories
Livermore, CA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

This research was funded by ELDRD project 212581.

Stochastic Hessian Information

Sandia National Laboratories

How do we efficiently collect and apply Hessian information during the course of optimization for machine learning?

Some recent work:

Berahas, Bollapragada, & Nocedal. ***An Investigation of Newton-Sketch and Subsampled Newton Methods. 2017.***

Bollapragada, Byrd, & Nocedal. ***Exact and Inexact Subsampled Newton Methods for Optimization. 2016.***

Byrd, Chin, Neveitt, & Nocedal. ***On the Use of Stochastic Hessian Information in Optimization Methods for Machine Learning. 2011.***

Review - Notation

Sum-like objectives are quite common in optimization for machine learning.

Model parameters: $\theta \in \mathbb{R}^m$

Training samples: $(x_i, y_i) \quad \text{for all} \quad i \in [N]$

(input, desired-output)

Sample index

Sample loss: $j_i(\theta) = j(x_i, y_i|\theta)$

Error between model prediction and desired output.

Goal: $\min_\theta J(\theta) = \frac{1}{N} \sum_i^N j_i(\theta)$

Review - Gradient Descent

Ordinary gradient descent optimization would take the form

Iteration index

This could be generalized to any step-size and descent-direction.

$$\theta^{(j+1)} = \theta^{(j)} - \alpha \nabla J(\theta^{(j)})$$

The gradient of the objective is a **sum over sample gradients**

$$\nabla J(\theta) = \frac{1}{N} \sum_{i}^{N} \nabla j_i(\theta)$$

What if the sum is too big? There could be millions of samples and we'd have to repeat after every iteration.

Review - Stochastic Gradient Descent (SGD)

Sandia National Laboratories

It is more feasible to compute updates using sample gradients*.

SGD update: $$\theta^{(j+1)} = \theta^{(j)} - \alpha \nabla j_{s_j}(\theta^{(j)})$$

Randomly select a sample and replace the true gradient with the sample gradient.

The expectation value matches the true gradient. As long as the learning rate is small, the trajectory approaches gradient descent.

$$\mathbb{E}[\nabla j_i(\theta)] = \nabla J(\theta)$$

*Robbins, H. & Monro, S. ***A Stochastic Approximation Method. 1951.***

Review - Batched SGD for Variance Reduction

Sandia National Laboratories

The sample gradient, however, may have high variance.
Batched sampling allows one to tune the trade-off between gradient-approximation variance and computational efficiency.

$$J_\Psi(\theta) = \frac{1}{|\Psi|}\sum_{i\in\Psi} j_i(\theta) \qquad \nabla J_\Psi(\theta) = \frac{1}{|\Psi|}\sum_{i\in\Psi} \nabla j_i(\theta)$$

Randomly sample subset with (or without*) replacement.

$$\theta^{(j+1)} = \theta^{(j)} - \alpha \nabla J_{\Psi_j}(\theta^{(j)}) \qquad \mathrm{Var}[\nabla J_\Psi] = \frac{1}{|\Psi|}\mathrm{Var}[\nabla j_i]$$

*Shamir, O. ***Without-Replacement Sampling for Stochastic Gradient Methods***. 2016.*

Related Work by Today's Speakers

Sandia National Laboratories

Adaptive batch-sizing:

Bollapragada, Byrd, & Nocedal. ***Adaptive Sampling Strategies for Stochastic Optimization****. 2017.*

Adaptive batch-weighting:

Needell & Ward. ***Batched Stochastic Gradient Descent with Weighted Sampling****. 2017.*

And what if the objective is non-differentiable?

Ryu & Byrd. ***Stochastic Proximal Iteration: A Non-Asymptotic Improvement Upon Stochastic Gradient Descent.*** *2018.*

Low-Rank Hessian Approximation

Sandia National Laboratories

Is it possible to use sample gradients to construct a low-rank approximation of the local Hessian?

Key assumption:

$$J(\theta) \approx J(\theta_0) + (\theta - \theta_0)^T g_0 + \frac{1}{2}(\theta - \theta_0)^T H_0 (\theta - \theta_0)$$

Objective at expansion point.

Gradient approximation at expansion point.

Local Hessian approximation.

Parameter Data over an Epoch

Sandia National Laboratories

During an SGD epoch, we can track parameters and gradients as columns of a matrix.

Parameter epoch:

$$\Theta = \begin{bmatrix} \theta^{(1)} & \theta^{(2)} & \cdots & \theta^{(n)} \end{bmatrix}$$

We use the **epoch-mean as the expansion point** and express parameters as differences from the mean.

Number of updates in epoch.

Parameter differences:

$$\hat{\Theta} = \Theta - \bar{\theta} e^T \quad \text{where} \quad \bar{\theta} = \frac{1}{n} \Theta e$$

Gradient Data over an Epoch

We use corresponding constructions for gradient samples.

Gradient epoch:

$$G = \begin{bmatrix} g^{(1)} & g^{(2)} & \cdots & g^{(n)} \end{bmatrix}$$

$$\text{where} \quad g^{(i)} = \nabla J_{\Psi_i}(\theta^{(i)})$$

Gradient differences:

$$\hat{G} = G - \bar{g}e^T \quad \text{where} \quad \bar{g} = \frac{1}{n}Ge$$

Hind-foresight: this may seem strange now, but it is a good choice.

Second-Order Predictions

If the objective approximation has the form

$$\tilde{J}(\hat{\theta}) = J_0 + \hat{\theta}^T g_0 + \frac{1}{2}\hat{\theta}^T H_0 \hat{\theta}$$

then the gradient would be $\nabla \tilde{J}(\hat{\theta}) = g_0 + H_0\hat{\theta}.$

Gradient predictions for the epoch:

$$\tilde{G} = g_0 e^T + H_0 \hat{\Theta}$$

Recalling $\hat{\Theta} = \Theta - \bar{\theta} e^T.$

Optimizing Gradient Predictions

We seek the gradient g_0 and Hessian H_0 values that **minimize the Frobenius norm of gradient prediction error.**

$$\min_{g_0, H_0} \omega(g_0, H_0) = \left\| \tilde{G} - G \right\|_F^2 \quad \text{subject to} \quad H_0 = H_0^T$$

unpacking definitions so far gives

$$\omega(g_0, H_0) = \left\| g_0 e^T + H_0 \hat{\Theta} - \hat{G} - \bar{g} e^T \right\|_F^2$$

Solving for optimal g_0

A little algebraic manipulation gives

$$\omega(g_0, H_0) = \left\|(g_0 - \bar{g})e^T\right\|_F^2 + \left\|H_0\hat{\Theta} - \hat{G}\right\|_F^2 + 2\,\mathrm{tr}\left((H_0\hat{\Theta} - \hat{G})e(g_0 - \bar{g})^T\right)$$

Unaffected by g_0.

Re: hind-foresight. Since we subtracted means, this is zero.

This term can be reduced to zero by setting $g_0 = \bar{g}$.

Solving for optimal H_0

$$\omega(g_0, H_0) = \left\|(g_0 - \bar{g})e^T\right\|_F^2 + \left\|H_0\hat{\Theta} - \hat{G}\right\|_F^2 + 2\,\mathrm{tr}\left((H_0\hat{\Theta} - \hat{G})(g_0 - \bar{g})^T\right)$$

The intermediate optimization problem is now

$$\min_{H_0} \omega(\bar{g}, H_0) = \left\|H_0\hat{\Theta} - \hat{G}\right\|_F^2$$
$$\text{subject to} \quad H_0 = H_0^T$$

rank-$(n-1)$

rank-m

Unfortunately, the typical epoch duration n will be too short to obtain a full approximation.

Problem: $n \ll m$

Projection onto Active Subspace

We really want to characterize the Hessian in the subspace that is being actively explored.

Orthonormal basis: U $\quad U^T U = I$

Low-rank constraint: $H_0 = U X U^T \quad \text{s.t.} \quad X = X^T$

More hind-foresight: It is always possible to construct the basis to **diagonalize the parameter covariance projection**.

$$\hat{U}^T \hat{\Theta} \hat{\Theta}^T \hat{U} = V \operatorname{diag}(\tau) V^T \quad \text{then} \quad U = \hat{U} V.$$

tentative

adjusted

Stochastic Hessian Projection (SHP)

With the additional projection constraint, we now have

$$\min_X \omega(\bar{g}, UXU^T) = \left\| XU^T\hat{\Theta} - U^T\hat{G} \right\|_F^2$$
$$\text{subject to} \quad X = X^T$$

which, again, we can express using the trace

$$\omega = \text{tr}\left(\left[XU^T\hat{\Theta} - U^T\hat{G} \right] \left[\hat{\Theta}^T UX - \hat{G}^T U \right] \right)$$

Solving for X

Sandia National Laboratories

Differentiate and apply variational principle:

$$0 = U^T\hat{\Theta}\left[\hat{\Theta}^T U X - \hat{G}^T U\right] + \left[X U^T\hat{\Theta} - U^T\hat{G}\right]\hat{\Theta}^T U$$

Simple form of continuous Lyapunov equation:

$$\mathrm{diag}(\tau)X + X\mathrm{diag}(\tau) = U^T\hat{\Theta}\hat{G}^T U + U^T\hat{G}\hat{\Theta}^T U$$

This is why diagonalizing parameter covariance was a nice choice.

Recall $U^T\hat{\Theta}\hat{\Theta}^T U = \mathrm{diag}(\tau)$.

elementwise division

$$X = \left[U^T\hat{\Theta}\hat{G}^T U + U^T\hat{G}\hat{\Theta}^T U\right] \oslash \left[\tau e^T + e\tau^T\right]$$

SHP Summary

Sandia National Laboratories

Parameter epoch: $\Theta = \begin{bmatrix} \theta^{(1)} & \theta^{(2)} & \cdots & \theta^{(n)} \end{bmatrix}$

Gradient epoch: $G = \begin{bmatrix} g^{(1)} & g^{(2)} & \cdots & g^{(n)} \end{bmatrix}$

Mean-differences: $\hat{\Theta} = \Theta - \bar{\theta} e^T \quad \hat{G} = G - \bar{g} e^T$

Diag. covariance basis: $U \quad \text{s.t.} \quad U^T \hat{\Theta} \hat{\Theta}^T U = \text{diag}(\tau)$

$$X = \left[U^T \hat{\Theta} \hat{G}^T U + U^T \hat{G} \hat{\Theta}^T U \right] \oslash \left[\tau e^T + e \tau^T \right]$$

Approx. local gradient: $\nabla \tilde{J}(\theta) = \bar{g} + U X U^T (\theta - \bar{\theta})$

How do you choose the basis U?

The upper bound on stable learning rates is controlled by the maximum Hessian eigenvalue. We would like to **maximize the maximum eigenvalue** of

$$X = \left[U^T \hat{\Theta} \hat{G}^T U + U^T \hat{G} \hat{\Theta}^T U\right] \oslash \left[\tau e^T + e \tau^T\right]$$

but we also need to remain in a subspace for which

$U^T \hat{\Theta} \hat{\Theta}^T U$ has been sufficiently explored.

Fast heuristic: $$\max_U \left\| U^T \hat{\Theta} \hat{G}^T U + U^T \hat{G} \hat{\Theta}^T U \right\|_F^2$$

Truncated Symmetric Eigenvalue Decomp.

Sandia National Laboratories

We can extract leading eigenpairs from the symmetric matrix

$$\hat{\Theta}\hat{G}^T + \hat{G}\hat{\Theta}^T$$

Challenge: This could easily become a performance bottleneck. We can't afford to form the full m^2 matrix and eigenvalue decomposition ($O(m^3)$ operations).

Randomized QR with Column Pivoting

Sandia National Laboratories

We can efficiently approximate maximum eigenpairs using a **minor modification to the randomized truncated SVD***.

Random projection: $$B = (\Omega\hat{\Theta})\hat{G}^T + (\Omega\hat{G})\hat{\Theta}^T$$

Low-rank QRCP: $$QR = BP$$

Candidate basis: $$[U_0, R_0] = \mathrm{qr}\left(\hat{\Theta}(\hat{G}^T P) + \hat{G}(\hat{\Theta}^T P)\right)$$

Concentrate basis: $$[U_1, R_1] = \mathrm{qr}\left(\hat{\Theta}(\hat{G}^T U_0) + \hat{G}(\hat{\Theta}^T U_0)\right)$$

Final extraction with Rayleigh-Ritz:

$$(U_1^T\hat{\Theta})(\hat{G}^T U_1) + (U_1^T\hat{G})(\hat{\Theta}^T U_1) = V\,\mathrm{diag}(\xi)V^T \quad \text{then} \quad U = U_1 V_{\max}.$$

*Duersch & Gu. ***Randomized QR with Column Pivoting***. 2017.

Related Work

Efficient randomized matrix decomposition:

Xiao, Gu, & Langou. ***Fast Parallel Randomized QR with Column Pivoting Algorithms for Reliable Low-rank Matrix Approximation****. 2017.*

Feng, Xiao, & Gu. ***Low-Rank Matrix Approximations with Flip-Flop Spectrum-Revealing QR Factorization****. 2018.*

Efficient symmetric eigen-decomposition:

Duersch, Shao, Yang, & Gu. ***A Robust and Efficient Implementation of LOBPCG****. 2017.*

Stationary Distribution

We also solved the stationary distribution of fixed-rate SGD with a **quadratic objective** assuming **gradient noise is Gaussian**.

$$J(\theta) \approx J_0 + \frac{1}{2}(\theta - \bar{\theta})^T H (\theta - \bar{\theta})$$

Sample gradient covariance: $\Gamma = \mathrm{Cov}(\nabla j_{\Psi_i})$

Parameters also follow a Gaussian distribution

Parameter covariance: $\Delta = \mathrm{Cov}(\theta^{(i)})$

Requires $\alpha < 2\, h_{max}^{-1}$

$$\Delta = \alpha V \left(\Gamma \oslash \left[he^T + eh^T - \alpha hh^T\right]\right) V^T$$

where $H = V \mathrm{diag}(h) V^T$

learning rate

Validating Approximations

Another (*very expensive*) way to approximate the maximum Hessian eigenvalue uses **symmetric finite differences**.

Second-order finite difference:

$$\frac{\partial}{\partial \theta_i} \nabla J_\Psi(\theta) \approx \eta_i = \frac{1}{2\delta} \left(\nabla J_\Psi(\theta + \delta e_i) - \nabla J_\Psi(\theta - \delta e_i) \right)$$

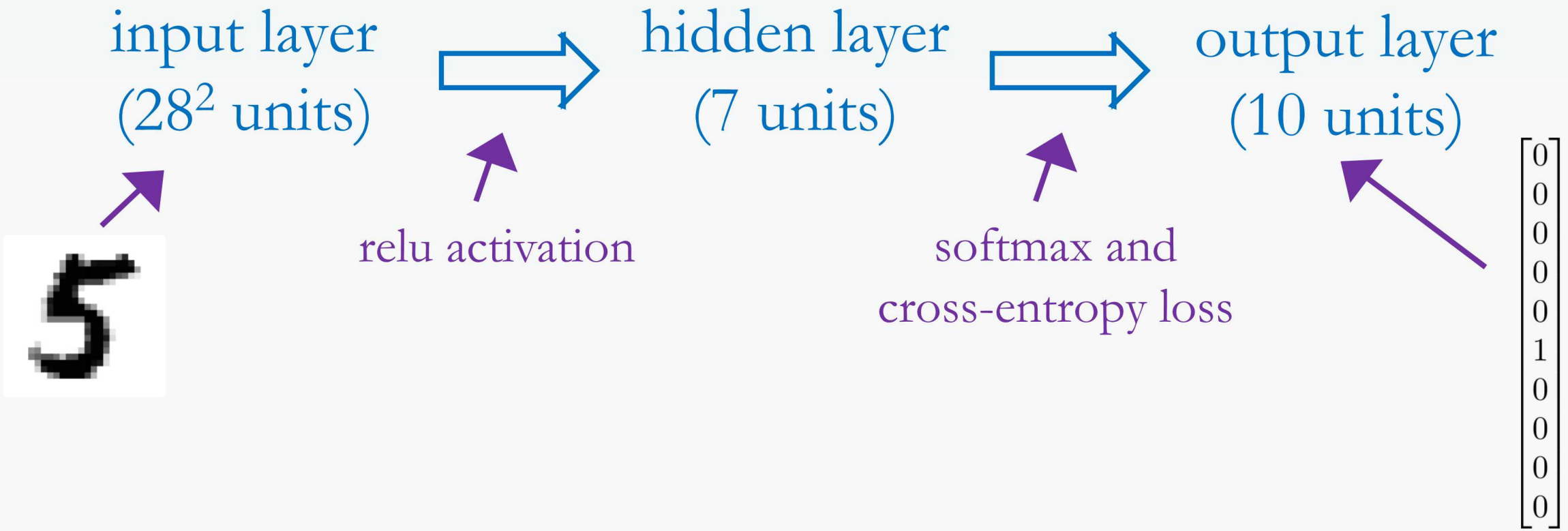
Evaluate for **all** parameters:

$$H \approx \begin{bmatrix} \eta_1 & \eta_2 & \cdots & \eta_m \end{bmatrix}$$

Then simply use the full symmetric eigenvalue decomposition.

MNIST Experiment Setup

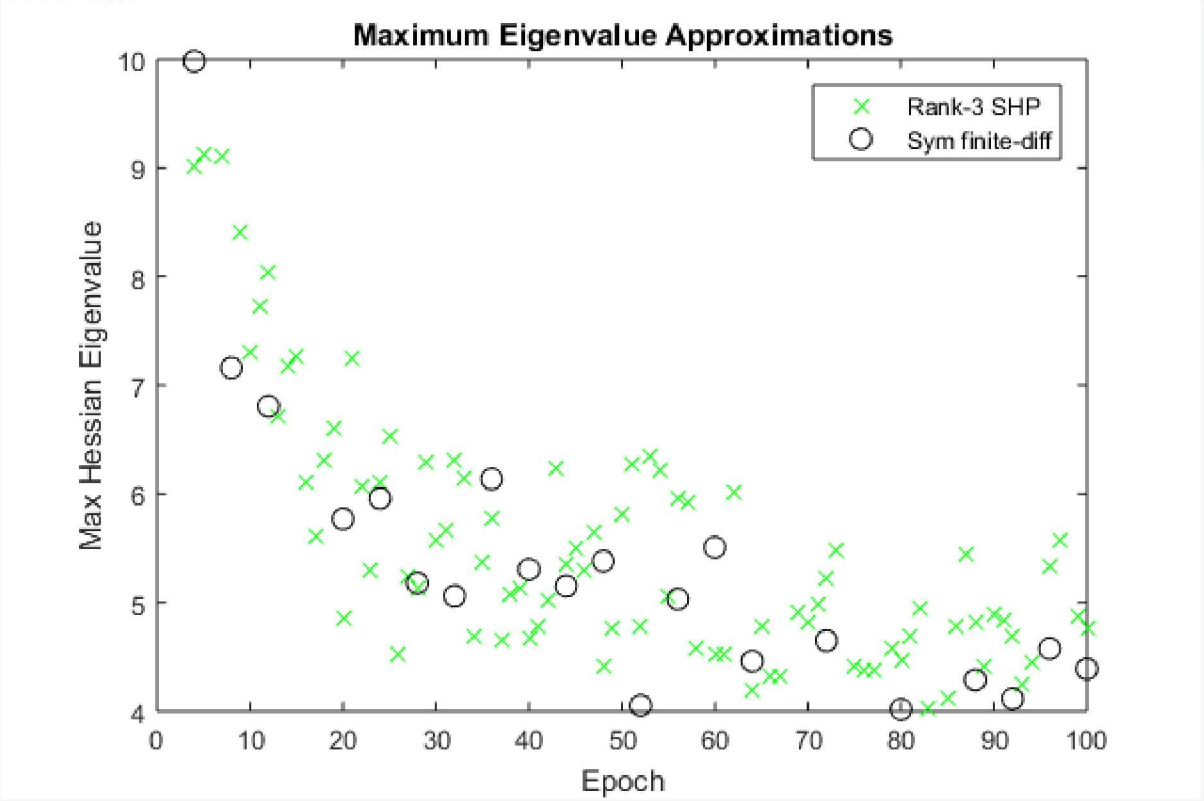
We checked SHP predictions on a simple MNIST digit classifier.



MNIST Hessian Comparison

Every 4 epochs we compute the finite-difference approximation using the same expansion point as SHP (parameter mean). We observe the same trend and spread.

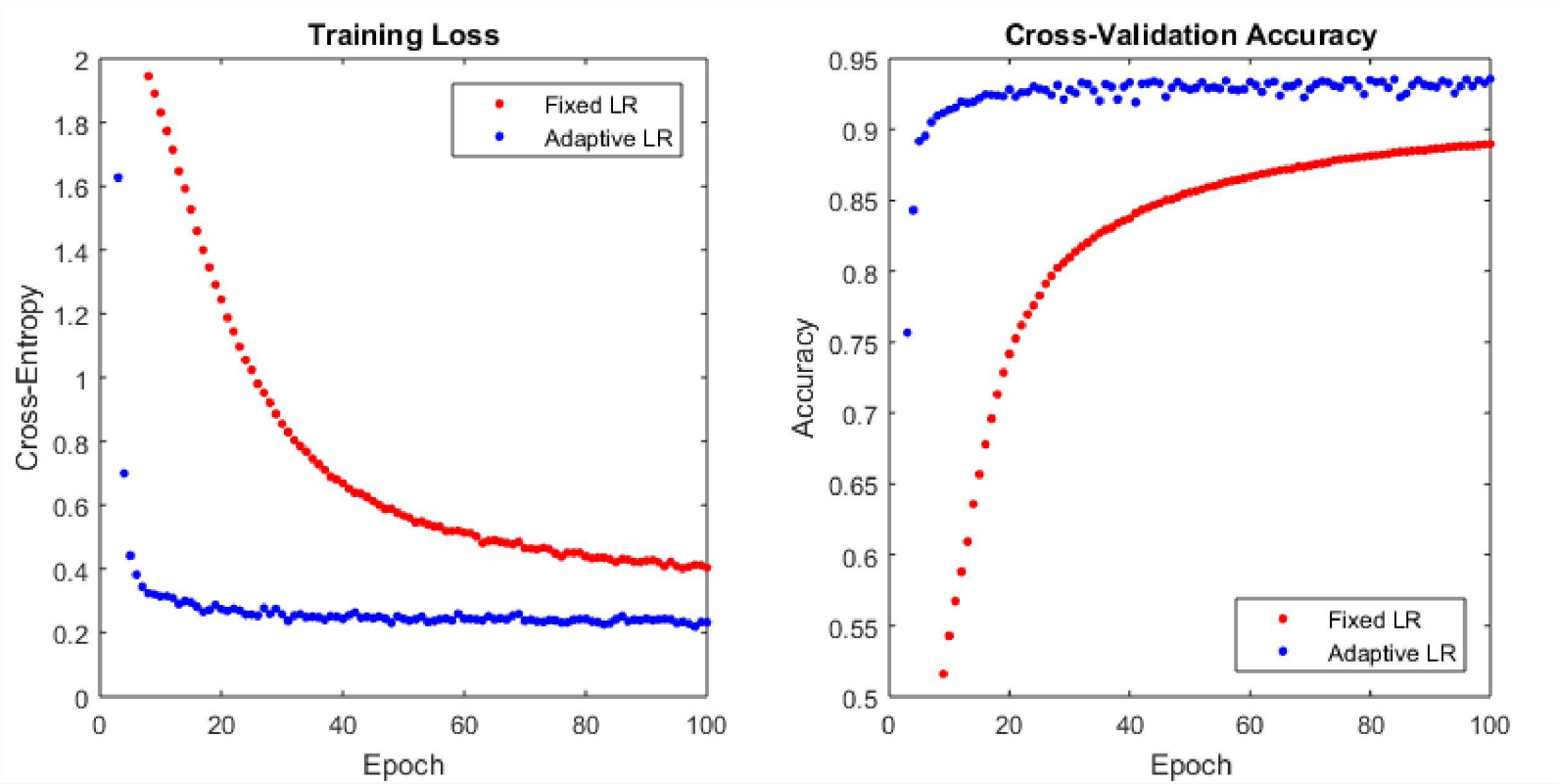
Batch size: 60
Epoch duration: 240
Initial learning rate: 0.001
Adaptive rate: $\frac{1}{2h_{max}}$
SHP rank: 3
Finite-difference samples: 800



MNIST Adaptive Learning Rate

Red points:
Fixed rate = 0.001

Blue points:
Initial rate = 0.001
Adapt target = $\frac{1}{2\,h_{max}}$



Adjusting the learning rate with SHP clearly improves convergence.

Conclusion

Stochastic Hessian Projection provides a feasible means of extracting Hessian information during optimization.

- **Negligible additional cost** over standard SGD.
- Max eigenvalue allows **dynamic learning rate** adjustment.

Questions for continuing research:

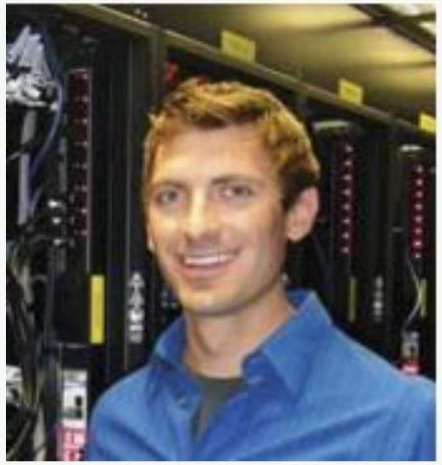
- Is there a principled way to **determine the extraction rank**?
- How do we dynamically **adjust the epoch duration**?
- Can we accelerate descent using small eigenvalues in a similar fashion to Newton-Sketch methods?

Jed A. Duersch - jaduers@sandia.gov

Special Thanks

Sandia National Laboratories

Tammy Kolda & Kevin Carlberg



Jed A. Duersch - jaduers@sandia.gov