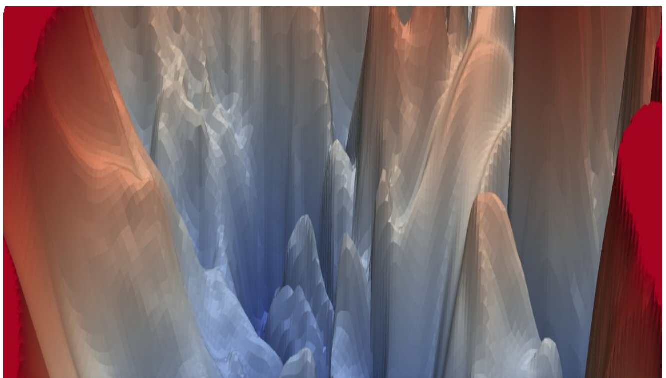
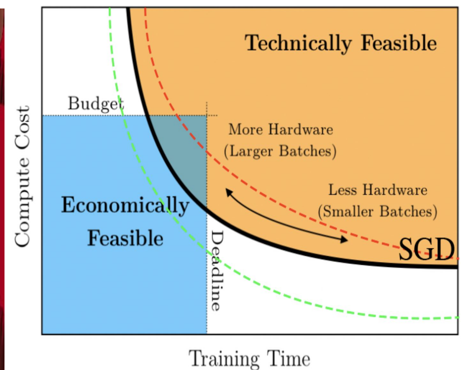
A Systematic Analysis of Second-Order Optimization in Large Scale Neural Network Training

Linjian Ma Gabe Montague Jiayu Ye
EECS Department, University of California at Berkeley
{linjian, gabe_montague, yejiayu}@berkeley.edu

Second-order acceleration for the training of deep neural networks (DNNs) may offer the promise of saved time, but caution and care are required to avoid pitfalls.



Above: Objective landscape visualization of a DNN: *f*(*x*)



Is Second Order

BETTER OR WORSE

- Modern development of AI technology is limited by the time and expense required to train DNNs
- Training involves finding the **global minimum** of an objective function (landscape)
- In recent years, optimization researchers have proposed that **second-order methods** may help to decrease training times.

First-Order Methods

$$f(x) \approx f(x_0) + \nabla f(x_0)^\top (x - x_0)$$

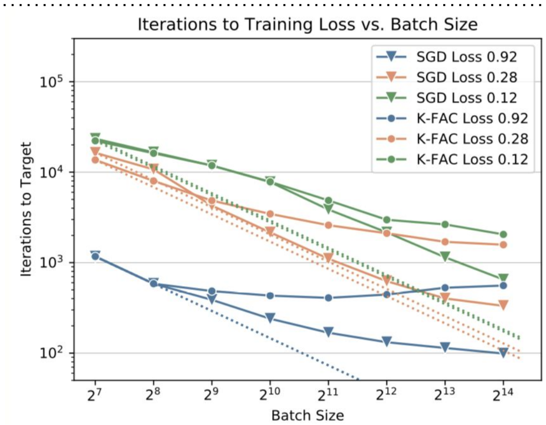
Approximate *f*(*x*) as *hyperplane*

Second-Order Methods

$$f(x) \approx f(x_0) + \nabla f(x_0)^\top (x - x_0) + \frac{1}{2}(x - x_0)\mathbf{H}(x_0)(x - x_0)$$

Approximate *f*(*x*) as *quadric hypersurface*

Method Investigated	**Order**	**Success Cases**	**Scalable to Industry Level?**
Stochastic Gradient Descent (**SGD**)	First	Works with almost every problem	Yes; high data-efficiency
Kronecker-factored Approximate Curvature (**K-FAC**)	Second	Academic datasets under certain configurations	Not as much; lower data-efficiency
Trust Region Conjugate Gradient (**TRCG**)	Second	Academic datasets under certain configurations	No; struggles with local minima
Stochastic TRCG (**STRCG**)	Second	Certain academic datasets	Unlikely



A detailed evaluation of large-batch K-FAC is presented in our work: Ma et al. "Inefficiency of K-FAC for Large Batch Size Training." arXiv preprint arXiv:1903.06237 (2019).