# Efficient fitting of large-scale neural models


## Jascha Sohl-Dickstein, Niru Maheswaranathan, Ben Poole, Surya Ganguli


Efficient fitting of large-scale models is becoming increasingly crucial in systems neuroscience. As the rate of data acquisition grows, neuroscientists are tasked with modeling large-scale recordings of many neurons across long time scales, in response to complex stimuli and behavioral paradigms. The BRAIN initiative promises to drive the production of these datasets orders of magnitude larger. Additionally, as the questions neuroscientists wish to ask become more sophisticated, they also become more computationally demanding. For example, closed-loop or online experiments require models to be fit to neural activity in real time, and could allow breakthroughs in brain-machine interfaces and receptive field characterization if this optimization challenge can be solved.

Fitting models to these datasets often boils down to minimizing an objective function summed over the data. Most current approaches fall into two categories: Stochastic gradient descent algorithms process the data in minibatches, and each iteration is computationally cheap. Quasi-Newton methods estimate the curvature of the objective and require fewer iterations, yet each iteration is more expensive to compute.

We present an algorithm for optimizing high-dimensional functions over large datasets that combines the computational efficiency of stochastic gradient descent with the second order curvature information accessible by quasi-Newton methods. We unify these disparate approaches by maintaining an independent quadratic approximation for each minibatch, and maintain computational tractability even for high-dimensional problems by storing and manipulating these approximations in a shared, time-evolving low-dimensional subspace.

To demonstrate the effectiveness of our approach, we apply this procedure to the problem of fitting generalized linear models over many neurons. We test our algorithm on simulated data consisting of a population of 100 neurons, driven by external noise stimuli, spike-train history, and connections to other neurons.

We release open source Python code for both the optimizer and the GLM.

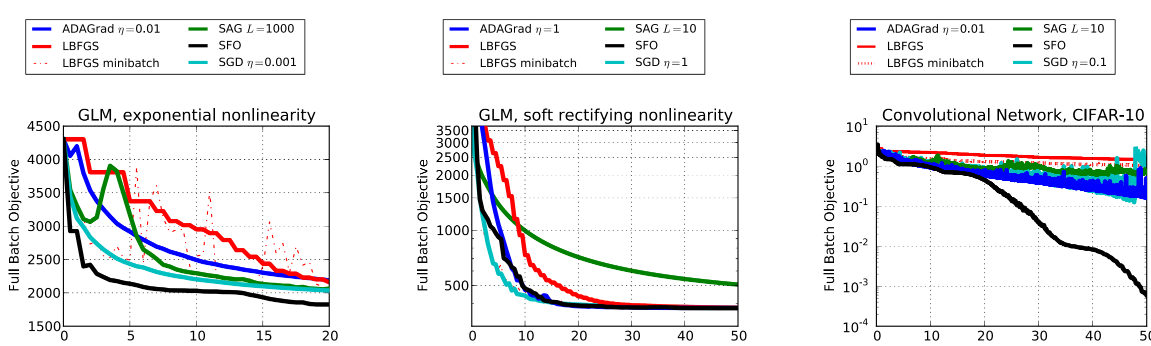


**Figure 1** A comparison of the proposed Sum of Functions Optimizer (SFO) to competing techniques for several objective functions. The bold lines indicate the best performing optimizer within each class. Note that unlike all other techniques besides LBFGS, SFO does not require a grid search over

hyperparameter values. The panes show convergence on *(left)* a generalized linear model (GLM) with exponential nonlinearity fit to artificial neural data and white noise stimulus, *(center)* a GLM with a soft rectifying nonlinearity fit to artificial neural data and pink noise stimulus, and *(right)* a three hidden layer convolutional neural network with rectified linear units classifying CIFAR-10 images.

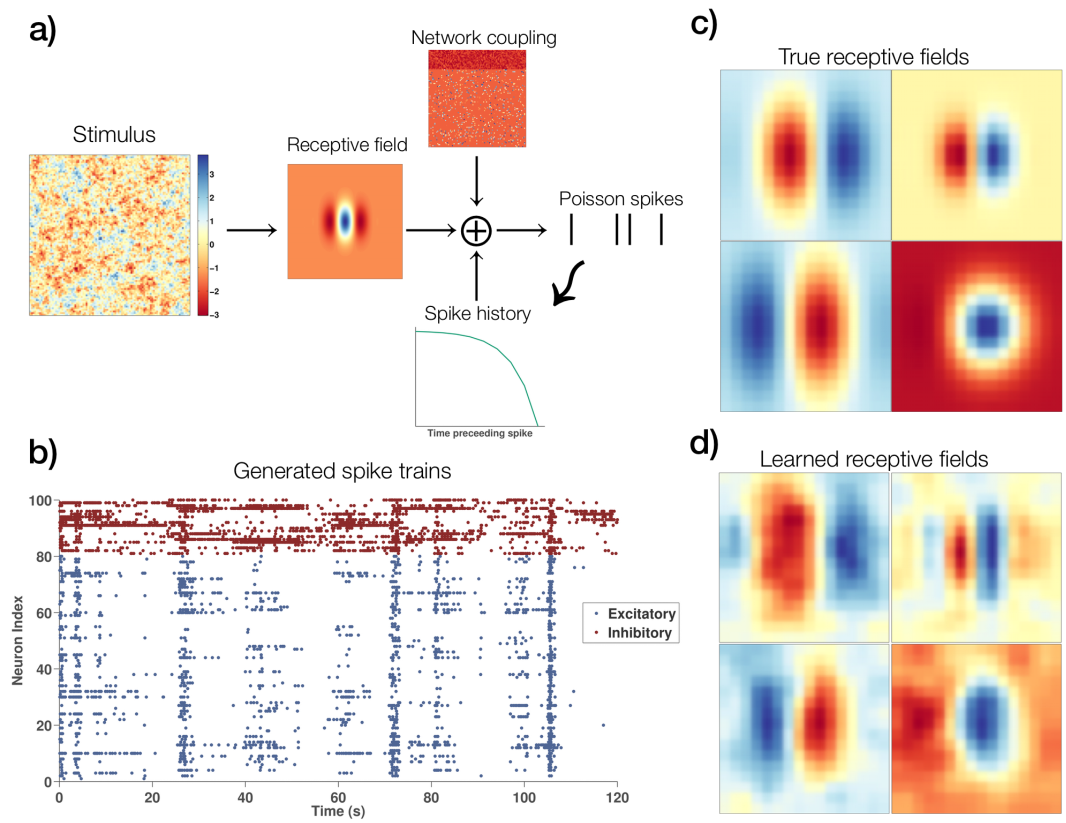


**Figure 2** The Generalized Linear Model (GLM) used to generate training data for the center pane in Figure 1. *(a)* GLM overview: Pink-noise stimuli are fed through receptive fields, and summed along with terms from network coupling (sparse, 20% inhibitory and 80% excitatory), and spike history (exponential decay over time). Firing rates were estimated as a nonlinear function of this sum. *(b)* Sample generated spike trains from the network. *(c)* Example receptive fields for 4 neurons. *(d)* Learned receptive fields after training using our algorithm. Panels correspond to the same neurons from *(c)*.

A preprint on our proposed optimization algorithm is available on arXiv at `http://arxiv.org/abs/1311.2115`.