Probabilistic Line Searches
for Stochastic Optimization
— Supplementary Material —

Maren Mahsereci and Philipp Hennig
Max Planck Institute for Intelligent Systems
Spemannstraße 38, 72076 Tübingen, Germany
`[mmahsereci|phennig]@tue.mpg.de`

This supplementary document contains additional results of the experiments described in the main paper, using the probabilistic line search algorithm to control the learning rate in stochastic gradient descent during training of two-layer neural network architectures in the CIFAR-10 and MNIST datasets.

1 Evolution of Function Values

Figure 1 plots the evolution of encountered raw function values against function evaluations. Each function call evaluates the gradient on a batch of size 10, both for SGD with constant and decaying learning rate, and for the line search-enhanced SGD. To keep the plot readable, the plot lines have been smoothed with a windowed running average, and only plotted at logarithmically spaced points. Among the noteworthy features of these plots is that SGD with large step sizes can be unstable (divergent dashed black lines), while this instability is caught and controlled by the line search. Regardless of initial step size, all line search-controlled instances perform very similarly, and reach close to optimal performance. Over the dynamic development of the optimization process, some specific choices of step size temporarily perform better than the line search-controlled instances, but this advantage slims or vanishes over time, because no fixed step size is optimal over the course of the entire optimization process. As mentioned in the main paper, finding those optimal step sizes would normally involve a tedious, costly search, which the line search effecively removes.

2 Optimal Step Sizes Vary During Training

It is a “widely known empirical fact” that SGD instances require a certain amount of run-time “tweaking”, because the optimal step size depends not just on the local structure of the objective, but also on batch-size (and associatd noise level). Figure 2 shows accepted step sizes, initial gradients at each search, and estimated gradient noise levels for the line search instances in the same experimental runs described above (smoothed and thinned as in Fig. 1 above). Starting five orders of magnitude apart, the line searches very quickly converged to similar step sizes; and indeed eventually settle around the empirically optimal value of $\alpha = 0.075$ (MNIST) and $\alpha = 0.08$ (CIFAR-10) (dashed green horizontal line in Fig. 2). But step sizes varied over time: starting out small, they then *increased*, and began decreasing again after around 10^4 line searches. This corroborates the empirical truism that learning rates should not immediately start decreasing, and only do so slowly. Interestingly, while there is an association between gradient values, noise and accepted step sizes, there appears to be no simple analytic relationship between the three. Overall, the emerging picture is that there is indeed nontrivial structure in the objective that is picked up by the line search.

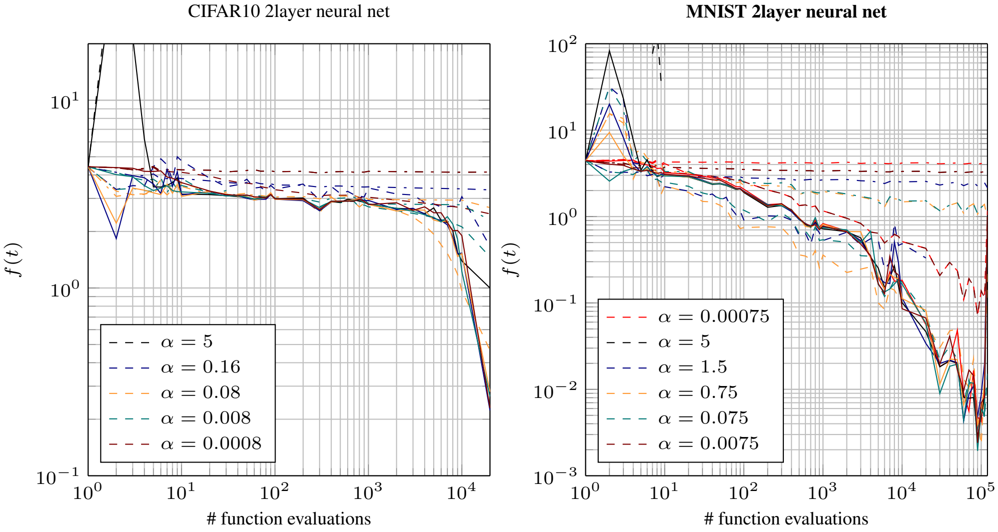

Figure 1: Function values encountered during neural network experiments. Vanilla SGD at different fixed learning rate α as dashed lines, SGD at different decaying learning rates as dashed-dotted lines. Solid lines show results using the probabilistic line search initialized at the corresponding α-values.

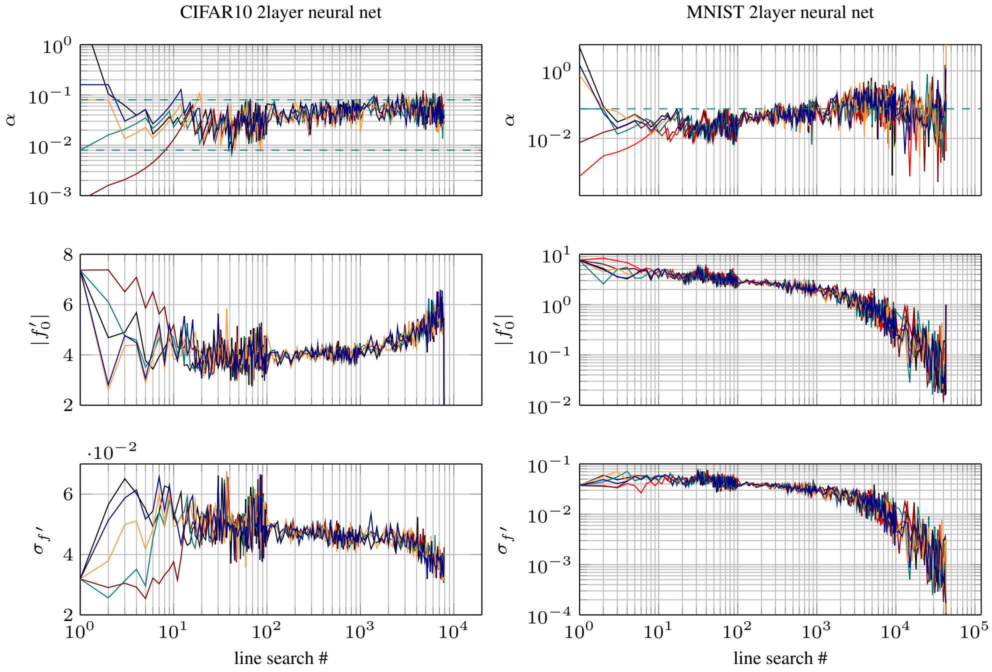

Figure 2: Same colors as Fig. 4 in the main paper: Accepted step sizes α, initial gradients $|f_0'|$ and absolute noise on gradient $\sigma_{f'}$. The probabilistic line search quickly fixes even wildly varying initial learning rates, within the first few line searches. The accepted step lengths drift over the course of the optimization, in a nontrivial relation to noise levels and gradient norms.

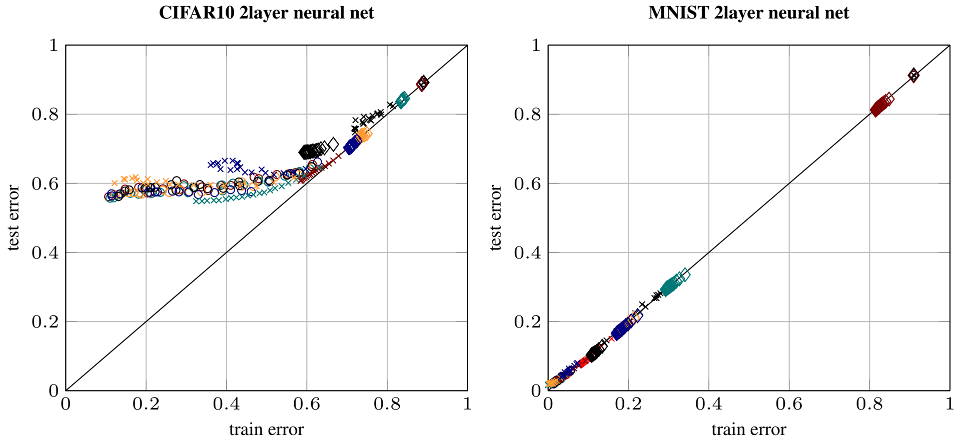

Figure 3: Test set error rates plotted against training set error rates. Same symbols and colors as in Figure 4 of the main paper. While there is significant over-fitting in CIFAR-10, and virtually no over-fitting in the MNIST case, the line search-controlled instances of SGD perform similarly to the best SGD instances.

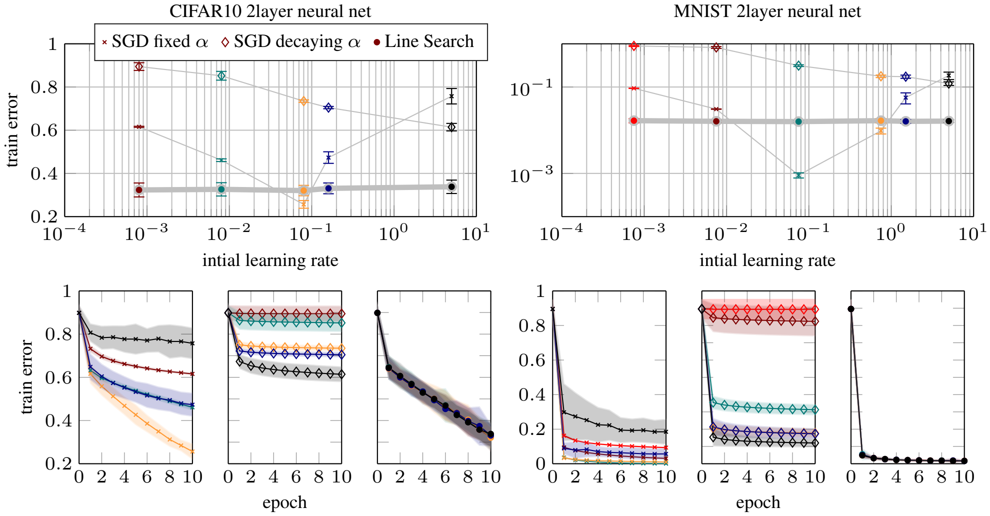

Figure 4: Top row: train error after 10 epochs as function of initial learning rate (note logarithmic ordinate for MNIST). Bottom row: Train error as function of training epoch (same color and symbol scheme as in top row). Same symbols and colors as in Figure 4 of the main paper. No matter the initial learning rate, the line search-controlled SGD perform close to the (in practice unknown) optimal SGD instance, effectively removing the need for exploratory experiments and learning-rate tuning. All plots show means and 2 std.-deviations over 20 repetitions.

3 Line Searches Do Not Affect Overfitting

A final worry one might have is that the control interventions of the line search might curtail an "accidental" benefical property of SGD—for example that the somewhat erratic, stochastic steps caused by stochasticity in the gradients allow SGD to "jump over" local minima of the objective. Such local minima can be a cause of over-fitting, or generally of low empirical performance. The plots in the main paper already confirm that the line search does not cause a stagnation in optimization performance, and can indeed improve this performance drastically. For completeness, Figure 3 also shows the relation between encountered train- and test-set error rates over the course of the optimization and Figure 4 shows the evolution of the train-set error per epoch as well as its dependence on the initial learning rate (same symbols and colors as in Figure 4 of main paper). There is generally little over-fitting in MNIST, and fairly strong over-fitting in CIFAR-10. But the instances controlled by the line search (circles) do not show a noticeably different behaviour, in this regard, to the uncontrolled, diffusive SGD instances. This suggests that, when the line search intervenes to curtail steps of SGD, it does so typically to prevent a truly sub-optimal step, rather than a beneficial "hop" over the walls surrounding a local minimum.