Optimal SGD Hyperparameters for Fully Connected Networks

Daniel S. Park[*]
Google Brain
danielspark@google.com

Samuel L. Smith
DeepMind
slsmith@google.com

Jascha Sohl-dickstein
Google Brain
jaschasd@google.com

Quoc V. Le
Google Brain
qvl@google.com

We address the challenge of identifying optimal hyperparameters for training neural networks by stochastic optimization. This is a broad topic, so we begin by establishing a set of assumptions and conditions to make the problem more concrete.

We specify a loss function $\mathcal{L}$ that is a function of the parameters (or weights) of the model, and a training set S_{train} of size N_{train}. The total loss $\mathcal{L}_\tau$ is obtained by summing all the individual losses $\mathcal{L}(s)$ of training samples indexed by s. We optimize $\mathcal{L}_\tau$ either by stochastic gradient descent (SGD) or SGD + momentum. We fix a batch size B, a learning rate ϵ and, in the case of SGD + momentum, a momentum m. We do not consider annealing or any other kind of parameter schedule.

Given this set-up, we ask what the expectation value of a performance measure $\mathcal{P}$, such as test set accuracy, becomes at late times, i.e., we may define

$$\mathcal{P}_\infty(\mathcal{L}, S_{\text{train}}; \epsilon, B, m) = \lim_{t\to\infty} \mathbb{E}[\mathcal{P}(\mathcal{L}, S_{\text{train}}; \epsilon, B, m; t)] \,. \tag{1}$$

We then formulate a definition of optimal hyperparameters

$$(\epsilon, B, m)^\infty_{\text{opt}} = \underset{(\epsilon,B,m)}{\text{argmax}}\, \mathcal{P}_\infty(\mathcal{L}, S_{\text{train}}; \epsilon, B, m) \,. \tag{2}$$

It was demonstrated in [11] that the macroscopic behavior of stochastic optimization is governed by a combination g of the introduced hyperparameters, where

$$g = \frac{\epsilon N_{\text{train}}}{2B} \quad \text{for SGD} \quad \text{and} \quad g = \frac{\epsilon N_{\text{train}}}{2B(1-m)} \quad \text{for SGD + momentum} \,. \tag{3}$$

Thus our goal will be to find optimal values for g given the loss function $\mathcal{L}$ and training set S_{train}.

As discussed in [1, 2, 9, 5, 11], the continuum limit of stochastic gradient descent is described by the stochastic differential equation

$$dw_i = -\partial_i \mathcal{L}_\tau dt + d\xi_i \,, \quad \mathbb{E}[d\xi_i d\xi_j]|_{w,t} = 2gF_{ij}(w)dt \,, \tag{4}$$

where w_i is used to denote the parameters of the neural network, and $d\xi_i$ is random noise. Thus the noise in stochastic gradient descent in the continuum limit is completely governed by the matrix $g\boldsymbol{F}$, where $\boldsymbol{F}$ is the covariance matrix of the gradient of the loss function with respect to the sample:

$$F_{ij} = \frac{1}{N_{\text{train}}} \sum_s \frac{\partial}{\partial w_i} \left(\mathcal{L}(s) - \frac{1}{N_{\text{train}}} \mathcal{L}_\tau \right) \frac{\partial}{\partial w_j} \left(\mathcal{L}(s) - \frac{1}{N_{\text{train}}} \mathcal{L}_\tau \right) \,. \tag{5}$$

We approximate $\boldsymbol{F}$ by the empirical Fisher information matrix $\boldsymbol{\mathcal{F}}$ of L over the training set.

[*]Work done as a member of the Google AI Residency Program.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

Table 1: Network Architectures

Layers (L)	Widths (N_l)	Total
1	[16, 32, 64, 128, 256, 512, 1024, 2048]	8
2	[64, 128, 256, 512, 1024]^2	25
3	[128, 256, 512, 1024]^3	64

When $\boldsymbol{F}$ is isotropic, i.e., when its eigenvalues are all identical and have value Γ, $g\Gamma$ is a local temperature of the system. In systems with an isotropic and constant Fisher information matrix, the optimal value of g can be obtained by matching this temperature to a characteristic loss scale. In particular, when $g\Gamma$ is set to 1, we expect the SGD to sample weights from the Bayesian posterior [9].

In realistic neural networks, it can be shown that $\boldsymbol{F}$ is far from isotropic, and a comparable theoretical analysis is difficult to carry out. We however report on an empirical formula that is able to give an estimate of the optimal value of g for fully-connected networks at critical initialization conditions [8, 10] trained for classification tasks. We find that

$$g_{\text{opt}}^{\text{crit}} \cdot \Gamma^{\text{crit}} \equiv g_{\text{opt}}^{\text{crit}} \cdot \mathbb{E}_{w\sim\mathcal{I}_{\text{crit}}}[\operatorname{tr} \boldsymbol{F}]/\sqrt{P} \approx C \tag{6}$$

where C is a constant only dependent on the training set, and $P = \operatorname{tr} \mathbf{1}$ is the number of parameters of the network. The performance measure we use to define optimality is test set accuracy.

We stress that $g_{\text{opt}}^{\text{crit}}$ is defined purely in terms of performance of the trained network. This should be distinguished from references to ideal learning rates in the literature, which often are defined as the maximal learning rate with acceptable convergence properties or numerical stability.

Details and Implications

We train 97 fully-connected networks with ReLU nonlinearities on classification tasks. The architectural details are given in Table 1. The networks have an input layer with N_0 units, and an output layer of width N_{L+1}, where N_{L+1} is the number of classes in the task. The loss is given by the cross entropy between the sample distribution and the distribution given by the softmax of the output layer.

We employ two different weight-parameterization schemes, the standard scheme where the output of the l-th unit y_i^l is given by

$$z_i^l = \sum_{j=1}^{N_l} W_{ij}^l y_j^{l-1} + b_i^l\,, \quad y_i^l = \phi(z_i^l)\,, \quad W_{ij}^l \sim \mathcal{N}(0, \frac{\sigma_w^2}{N_l})\,, \quad b_i^l \sim \mathcal{N}(0, \sigma_b^2)\,, \tag{7}$$

where ϕ is the ReLU function. The final output units are given by z_i^{L+1} rather than y_i^{L+1}.

Alternatively, we can use the neural tangent kernel (NTK) scheme proposed in [4], where

$$z_i^l = \frac{1}{\sqrt{N_l}} \sum_{j=1}^{N_l} W_{ij}^l y_j^{l-1} + b_i^l\,, \quad y_i^l = \phi(z_i^l)\,, \quad W_{ij}^l \sim \mathcal{N}(0, \sigma_w^2)\,, \quad b_i^l \sim \mathcal{N}(0, \sigma_b^2)\,. \tag{8}$$

The networks are initialized to be critical in the wide-limit, so that the expectation values of the weight-derivatives do not decay with respect to layer depth. Practically, this is done by tuning the initialization parameters σ_w and σ_b [3, 8, 10]. We set $\sigma_w = \sqrt{2}$ and $\sigma_b = 0.0001$.

In each case, $\operatorname{tr} \boldsymbol{F} \approx \operatorname{tr} \boldsymbol{\mathcal{F}}$ can be computed following [6] in the "wide limit":

$$\Gamma_{\text{Standard}} \approx c_0 \cdot \frac{\sum_{l=0}^{L} N_l}{\sqrt{P}}\,, \quad \Gamma_{\text{NTK}} \approx c_0 \cdot \frac{2(L+1)}{\sqrt{P}}\,. \tag{9}$$

Here, c_0 is an order-1 training set dependent constant.

Assuming a network of depth L and width W, $\Gamma^{\text{crit}} \sim \sqrt{L}$ in the standard scheme and $\Gamma^{\text{crit}} \sim \sqrt{L}W^{-1}$ in the NTK scheme since $P \sim W^2 L$. Thus, in order to inject the optimal amount of noise into the gradient descent, the batch size must taken to be

$$B \propto \sqrt{L}/\epsilon \quad \text{in the standard scheme, and} \quad B \propto (\sqrt{L}/\epsilon) \cdot W^{-1} \quad \text{in the NTK scheme,} \tag{10}$$

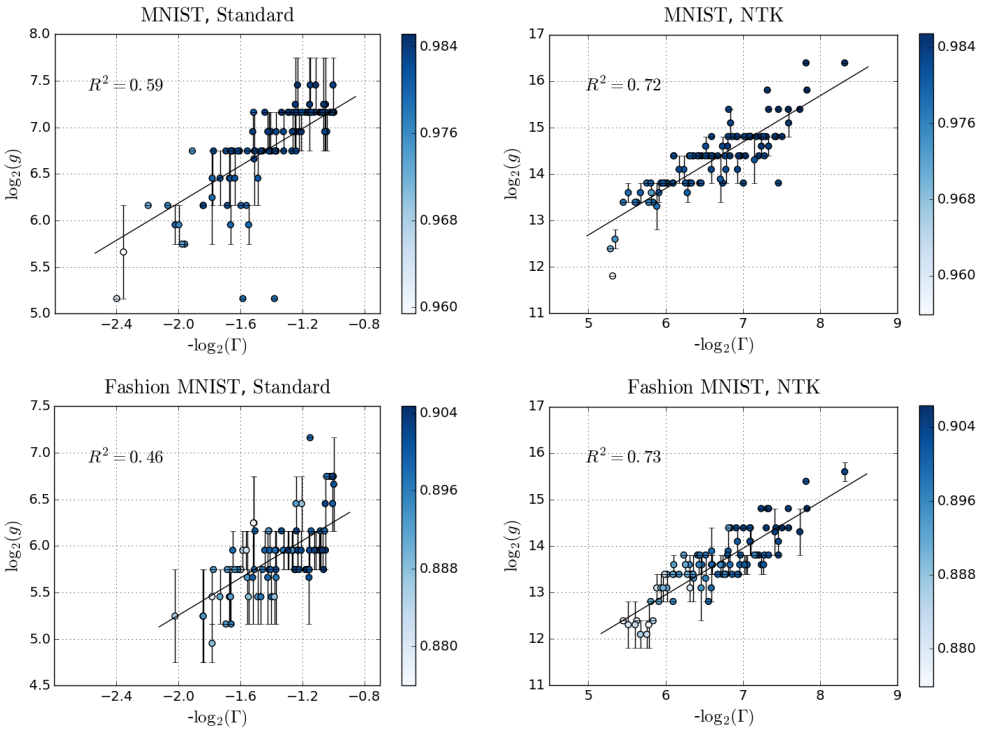

Figure 1: $\log_2(g^{\text{crit}}_{\text{opt}})$ vs. $-\log_2(\Gamma^{\text{crit}})$ for trained runs

when W becomes large. Since $B \geq 1$, this has implications on the stochastic optimization of wide networks, if one assumes that (6) determines the optimal g value of the continuum limit dynamics.

The consequence of assuming that the result (6) holds for the continuum limit of SGD is that the implied optimality is not achievable practically for very wide fully connected networks. In the standard scheme, the learning rate ϵ must decrease with increasing width due to numerical stability. The relation (10) then implies that optimal performance is not achievable with growing width, since the batch size must get indefinitely small in order to generate the optimal amount of noise, which is not possible. Meanwhile, in the NTK scheme, ϵ can stay constant with respect to the width and still maintain numerical stability. In this case, however, $B \propto (\sqrt{L}/\epsilon) \cdot W^{-1}$, which yet again forces the batch size required to yield the right amount of noise to approach zero in the infinite width limit.

Experiments

We trained the aforementioned networks on two datasets, MNIST [7] and fashion MNIST (fMNIST) [12] and searched for hyperparameters that maximize the training set accuracy at "late times."

We trained each network with SGD + momentum, with batch sizes ranging from 32 to 4096. The learning rate was set to either 0.1 or 0.05 depending on their size for networks parameterized with the standard scheme, and 10.0 for networks parameterized with the NTK scheme. The momentum was set uniformly to $m = 0.9$. For a given learning rate and a batch size, a minimum of 20 training sessions have been run, at least 5 of which have completed successfully. To observe late time behavior, we train the networks "sufficiently long," by giving a lower-bound to a minimum number of epochs and training steps fit to the dataset and training scheme. An optimal g was found for a network by selecting the batch size for which the average test set accuracy at the final step of training was the greatest for a given learning rate.

Figure 1 plots optimal g against Γ^{-1} for performance defined over "trained runs." Each point represents a single network for which Γ^{-1} and optimal g have been obtained. The average is only taken over runs where the networks have been successfully trained, i.e., reached validation set accuracy over 0.2 at some point in training. Only batch sizes for which 5 or more runs have been

trained have been counted. The data points are color-coded with respect to the test accuracy achieved by the particular network at optimality. Only networks that are able to achieve test set accuracy better than 95% for MNIST and 87% for fMNIST have been plotted. Note that larger networks tend to have larger values of Γ^{-1}, thus the trend of data points representing networks with better performance being on the right-side of the plots. The error bars span the range of batch sizes for which the test set error is within 68% of the confidence interval of the best test set error, while the lines depicted are best-fit slope-1 lines, i.e., lines along which $g\Gamma$ is constant, on the log-log plot.

We have performed similar experiments on a class of simple convolutional neural networks (CNNs). Our preliminary results suggest that in order to apply formula (6) to CNNs, one should replace the total parameter count P by the number of parameters in fully connected layers P_{fc}. We hope to study the optimality conditions for CNNs more thoroughly in future work.

References

[1] Lukas Balles, Javier Romero, and Philipp Hennig. Coupling adaptive batch sizes with learning rates. *arXiv preprint arXiv:1612.05086*, 2016.

[2] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *arXiv preprint arXiv:1710.11029*, 2017.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[4] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018.

[5] Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017.

[6] Ryo Karakida, Shotaro Akaho, and Shun ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. *CoRR*, abs/1806.01316, 2018.

[7] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. MNIST handwritten digit database. 2010.

[8] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *CoRR*, abs/1711.00165, 2017.

[9] Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

[10] Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *CoRR*, abs/1611.01232, 2016.

[11] Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. *CoRR*, abs/1710.06451, 2017.

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.