Partitioned Integrators for Thermodynamic Parameterization of Neural Networks

Benedict Leimkuhler, Charles Matthews and Tiffany Vlaar*
University of Edinburgh
School of Mathematics & Maxwell Institute for Mathematical Sciences, Edinburgh, EH9 3FD
*Tiffany.Vlaar@ed.ac.uk

1 Introduction

Traditionally, neural networks (NNs) are parameterized using optimization procedures such as stochastic gradient descent (SGD), RMSProp [18] and Adam [7]. These procedures tend to drive the parameters of the network toward a local minimum. In this article, we employ alternative "sampling" algorithms (referred to here as "thermodynamic parameterization methods") which rely on discretized stochastic differential equations (SDEs) for a defined target distribution on parameter space. We show that the thermodynamic perspective improves neural network training. Moreover, by partitioning the parameters based on natural layer structure we obtain schemes with very rapid convergence for data sets with complicated loss landscapes. The per-step cost of our methods is roughly similar to that of other training methods such as SGD and Adam, assuming the major cost of a timestep is dominated by the computation of the approximate gradient. For more details we refer to our preprint [12].

We describe easy-to-implement hybrid partitioned numerical algorithms, based on discretized SDEs, which are adapted to feed-forward neural networks, including a multi-layer Langevin algorithm, AdLaLa (combining the adaptive Langevin and Langevin algorithms) and LOL (combining Langevin and Overdamped Langevin); we examine the convergence of these methods using numerical studies and compare their performance among themselves and in relation to standard alternatives such as SGD and Adam. We present evidence that thermodynamic parameterization methods can be (i) faster, (ii) more accurate, and (iii) more robust than standard algorithms used within ML frameworks.

2 Bayesian parameterization

We focus on the training (parameterization) process for neural networks using ideas from statistical mechanics. We take the Bayesian perspective, that the parameters θ of a NN are defined by data $\mathcal{D}$ only in the sense of a probability distribution given by Bayes' formula. When the probability distribution is unimodal and convex it is natural to choose θ as the mode of the target distribution by maximizing the posterior probability density using the MAP technique, but in practice this does not hold for NNs. It then becomes a challenge to identify all relevant possible parameter values, and to compare different parameter choices in terms of their relative probabilistic weight. This task is referred to as *sampling*, and thus the Bayesian parameterization problem naturally reduces to a sampling problem for the parameters of the model. While the idea of Bayesian modelling is commonplace in all areas where statistics is used, the Bayesian perspective is usually only viewed as the starting point for optimization schemes in the setting of high dimensional NNs, due to the vast amounts of data and parameters involved [13]. We argue here that the sampling approach can provide parameterization candidates with as great or greater efficiency than standard optimization schemes.

We use the well known link between posterior sampling and MAP estimation. Introduce the negative log posterior $L(\theta) = -\ln \rho(\theta|\mathcal{D})$, and define $\rho_\tau(\theta) = \exp(-\tau^{-1}L(\theta)) = \rho(\theta)^{1/\tau}$. For $\tau = 1$ we have the posterior density. For $\tau \to 0$ we obtain a sequence of distributions which, although globally supported, have their mass confined progressively closer to the mode of the distribution. Thus we can think of MAP as an extreme form of sampling in which the sampled distribution is confined to the vicinity of the mode(s). In this setting, τ becomes a parameter of an embedded family of models which may be used to enhance the optimization process. An example is the process known as annealing,

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

where τ is gradually driven from higher to lower values [8]. The parameter τ plays precisely the same role as temperature in statistical physics, thus the use of the term *thermodynamic parameterization* to describe methods that rely on this embedding (and the sampling of the associated family of probability distributions) to enhance the parameterization procedure. In practice, whether we take a full Bayesian or pure MAP perspective, a relatively small range of parameter values are likely to be of interest (those that have relatively large statistical weight with respect to the probability distribution). Moreover, there is often much to be gained by exploring parameters in the vicinity of a local maximum, i.e. by short sampling paths. In this work we propose to use, as in stochastic gradient Langevin dynamics (SGLD) [19], additive noise (which has an adjustable but fixed strength) to stabilize the invariant measure of the stochastic dynamics, relying on underdamped Langevin dynamics and applying state-of-the-art discretization methods [10], which introduce additive noise within a framework of second order stochastic dynamics. For more details we refer to our paper [12]. Underdamped Langevin dynamics is described by the equations $\mathrm{d}\theta = p\,\mathrm{d}t, \quad \mathrm{d}p = -\nabla_\theta L\,\mathrm{d}t - \gamma p\,\mathrm{d}t + \sqrt{2\gamma\tau}\,\mathrm{dW}_t$, where θ are the parameters of the neural network, p the corresponding momentum variables, L the loss, γ (friction) and τ (temperature) are parameters to be tuned, and W_t a standard N-dim. Wiener process. To demonstrate the potential relevance of temperature in parameterization of NNs we compare two classifiers for planar trigonometric data obtained using Langevin dynamics for a fixed amount of work but parameterized with different temperatures (see Fig. 1). Both the test accuracy and qualitative features of the classifier improve as temperature increases. However, a too large temperature can also negatively affect the results, suggesting a 'Goldilocks' temperature region of optimal efficiency.

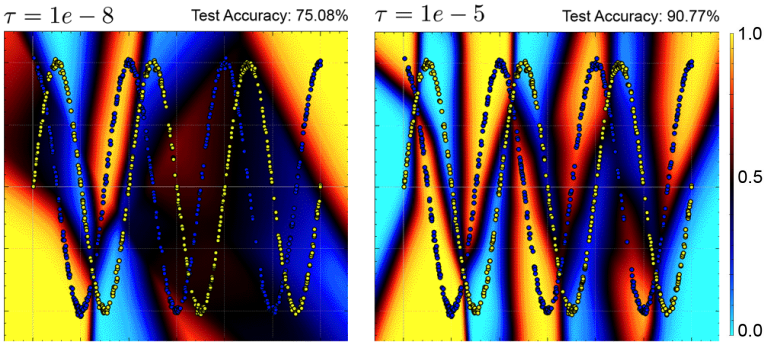

Figure 1: These classifiers are computed using the BAOAB Langevin dynamics integrator [10]. We used 50k steps with stepsize $h = 0.4$, friction $\gamma = 10$, a 500 node single hidden-layer perceptron (SHLP) with ReLU activation, sigmoidal output and a cross entropy loss. Temperatures were set to τ =1e-8 (left) and τ =1e-5 (right). The figures show that the classifier substantially improves as the temperature is raised. Visually, this means that the contrast between the color of the classifier and of the plotted data is higher. The data for class 0 is given by $x = 3t, y = \cos(2t\pi) + 0.02\mathcal{N}(0,1)$, where t is drawn repeatedly from $\mathcal{U}(0,1)$ to generate data points. Data for class 1 is generated similarly but with cos replaced by sine. We used 1000 training, 1000 test data points and 2% subsampling.

A model for the cause of the performance gain due to elevated temperature may be found in molecular diffusion on a rough energy landscape [20, 15]. In a corrugated energy surface and at zero temperature the system will likely get stuck in local minima, lacking the required energy to overcome barriers blocking movement between states. Increasing the temperature allows weak interaction with a heat bath, randomly introducing energetic fluctuations into the system that can move it over barriers and away from local minima. If the temperature is too small the fluctuations are small and it will take long to cross barriers, whereas too large and the system will not be drawn towards the global minimum.

3 Partitioned discretization algorithms

In layered or hierarchical models, e.g. deep neural networks, we have a natural partitioning of the parameter vector according to the role in the hierarchy. It may be useful to treat the parameters at different levels of the hierarchy differently in the parameterization process. In particular, it is possible that, either due to design or some feature of the network, the characteristics of the gradient noise introduced at different layer depths may differ, and it is then natural to design a method that treats the components independently. Lan et al. [9] observe that fixing the weights and biases of a NN's last layer can enhance the performance of training algorithms. We draw on this idea here for motivation in developing a family of partitioned algorithms for NN training.

We have developed partitioned thermodynamic algorithms which allow to vary the hyperparameters and even the form of the algorithm in various layers. For example, one class of methods (LOL) combines Langevin with overdamped Langevin dynamics (friction $\gamma \to \infty$) in the output layer. To be more specific, the LOL method applies a Langevin optimizer to update the weights and biases in the first layer and simultaneously uses an overdamped Langevin optimizer to update the parameters in the output layer. Another family of methods (AdLaLa) blends adaptive Langevin dynamics [5] (also known as SGNHT [1]) with Langevin dynamics. We found it advantageous to use low temperatures in the output layer but to maintain the hidden layer parameters at slightly elevated values. This means that those inner parameters can rapidly explore a wide range of low-loss states. We conjecture that it is this fluidity in the hidden layer which gives the LOL and AdLaLa methods their improved convergence speed for difficult classification tasks. In the extreme case, where the temperature is zero in the output layer of AdLaLa, that part can be viewed as a dissipated gradient system and thus analogous to gradient descent with momentum. However, the adaptive control of the first part of the partitioning appears to provide enhanced flexibility in the approach to the overall minimum.

For properties of the partitioned algorithms we draw on three recent works: (i) hypoelliptic properties of Langevin dynamics numerical methods [11], (ii) hypoelliptic properties for Langevin dynamics with configuration-dependent noise [16] and (iii) very recent work on weighted-L^2 hypocoercivity of Adaptive Langevin dynamics [17]. The latter results allow to establish a Central Limit Theorem which is very important in statistical applications.

4 Numerical examples and discussion

The methods were tested using three separate codes for cross-validation and verification of consistency: PyTorch [14], a DLIB package [6] written in C++, and a custom native C++/QT application which has been created by the authors for rapid visual exploration of training algorithms. In this work, we examine parameterization in the context of binary classification of spiral and trigonometric data, as well as limited testing with the MNIST data set. We found that the results were significantly different for the different problem classes. Our methods significantly outperformed standard optimizers for spiral and trigonometric datasets, whereas they perform competitively on MNIST compared to standard optimizers, but do not significantly outperform the other methods. We believe that this is caused by a fundamental difference in the loss landscape structure of the different data sets and therefore in the flexibility of the optimizers required to tackle them. In Huang et al. (2019) [3] they illustrate this by studying a binary classification problem, where they pinch the margin between two rings of datapoints, which causes any good minimizer to be "sharp". The small volume of the corresponding basin, makes these minima less likely to be found by standard optimizers.

We observe that there appear to be significant loss-barriers in the landscapes of the spiral and trigonometric data sets. For this reason, methods such as SGD and Adam, which, up to gradient noise, monotonically decrease the loss, can easily become trapped in unsuitable states or be slowed down by the presence of saddle points. By contrast MNIST data and related image classification problems may be relatively free of these issues, which causes our tests on MNIST data to show fewer substantial differences among optimization schemes. In our paper [12] we illustrate this using the technique of 1D linear interpolation proposed by Goodfellow et al. (2015) [2] and a surface plotting technique [4], but this is beyond the scope of this extended abstract.

4.1 Thermodynamic parameterization methods can have very rapid convergence

We provide evidence that LOL and AdLaLa are able to converge more rapidly to a low test-loss parameterization than standard optimizers such as SGLD or Adam. In Figure 2 we show the obtained test losses/accuracies using different optimizers for the trigonometric example. Adam is clearly not able to reach the accuracy that LOL and AdLaLa obtain. Its progress slows down rapidly and halts completely after 40k steps. After 100k steps its maximum test accuracy is still around 73%. SGLD is not able to compete at all. SGD is not shown in this figure, but also converges much slower than our methods for this example. We obtained similar results for the binary classification of spiral data sets, where class 0 was generated by $x = 2t\cos(2bt\pi) + 0.02\mathcal{N}(0,1), y = 2t\sin(2bt\pi) + 0.02\mathcal{N}(0,1)$ and the other class is created by a shift in the argument of the trig functions by π. When we vary b, this directly affects the number of turns of the spiral and therefore the complexity of the problem. We studied the performance of the different optimizers for 3-turn ($b = 3$) and 4-turn ($b = 4$) spirals,

which can be challenging test cases for standard optimizers. We observed that AdLaLa and LOL make fast headway towards high test accuracies, whereas Adam and SGD appear to get stuck in a parameterization with many small weights/biases and thus struggle to obtain good test accuracies. AdLaLa consistently outperforms Adam, SGD and SGLD in terms of convergence rate for these examples. Although we varied the stepsize for Adam, we did not vary the default parameters for Adam, i.e., the decay rates for the moving averages of the first and second moments. Although this is common among practitioners, there is some room left for experiments here.

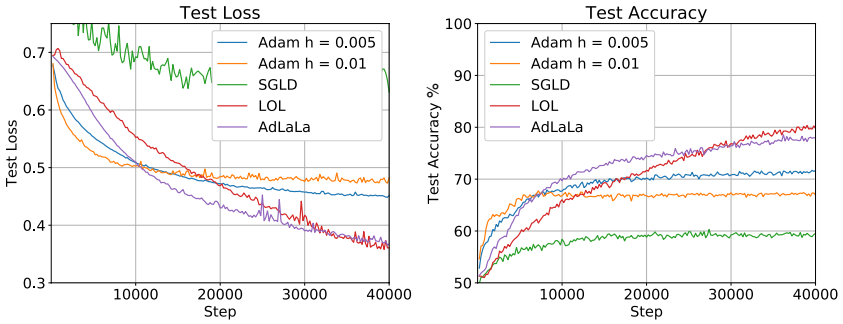

Figure 2: Test losses and accuracies obtained for the trigonometric example, where the data for class 0 is generated using $x = 10t, y = \cos(t\pi) + 0.02\mathcal{N}(0,1)$ and the same equation holds for class 1, but with a sine instead of a cosine. The results were averaged over 20 runs. It is clear that LOL and AdLaLa outperform the other methods. The stepsize for SGLD, LOL and AdLaLa were all set to $h = 0.1$. We used a 100-node SHLP, 1000 test data, 1000 training data and 5% subsampling.

4.2 Features of thermodynamic parameterization methods

We observe fundamental differences in the parameterizations obtained by sampling methods, such as SGLD and AdLaLa, compared to standard optimizers. The sampling methods rapidly excite a large amount of parameters, whereas SGD and Adam obtain parameterizations which have many (close to) zero weights and biases. We also observe that thermodynamic parameterizations appear to give rise to classifiers whose level sets are relatively smooth compared to those produced by alternative methods. Thermodynamic parameterization thus effectively controls the distribution of weights–more precisely the distribution of the conjugate momenta associated to the weights, due to the statistical mechanical property known as equipartition of energy. This is a consequence of ergodicity which simply states that the mean kinetic energy of all degrees of freedom, in thermal equilibrium, is constant. We confirmed experimentally that the magnitude of the squared momenta are approximately controlled by the set temperature value in AdLaLa and LOL.

Another benefit of using the thermodynamic parameterization approach is that it reduces the dependence of the training result on the initial conditions. We observe that SGD and Adam have a much larger variance in their obtained test accuracies over different runs than our methods. We also evaluated the robustness of our algorithms to overfitting. We studied the 2-turn spiral dataset with high noise level using a 500 node SHLP and a small amount of training data. For this example, SGD clearly overfits. In contrast, LOL with a large enough friction value can be shown to not exhibit this behaviour. The same can be said for AdLaLa for specific hyperparameter values. For these parameter settings LOL and AdLaLa are slower in reaching the desired test and training accuracy, but this leads to more stability later on in the training process and limits the need for early stopping techniques. We should emphasize that we don't claim that our methods never overfit, merely that they allow more flexibility which can lead to increased robustness to overfitting.

4.3 Discussion

We have presented a new approach to parameterization of neural networks which can, in data classification problems with complicated loss landscapes, accelerate convergence and provide improved test accuracy. The use of additive noise to supplement gradient noise was already proposed in previous works of other authors. We draw on this, by combining it with state-of-the-art principles for sampling algorithms coming from molecular dynamics and deploy partitioned algorithms that substantially improve on SGD and other optimization procedures. Although the experiments of this article have focused on toy problems and single-hidden layer perceptrons, our methods can easily be generalized to deeper networks and based on some limited testing we have done in this area, we expect our

methods to perform well in this setting. However, to maximise the benefit received from using our optimizers, one may have to increase the number of partitionings in this setting and therefore the number of hyperparameters of the optimizer. We also expect that our methods will be valuable in the study of streaming data, where the improved generalization properties of the models trained using our methods will help reduce costly reparameterizations.

5 Acknowledgements

The authors wish to thank John Chodera, Jason Frank, Anton Martinsson and Jonathan Weare for helpful discussions during the preparation of this manuscript. The work of Benedict Leimkuhler and Charles Matthews was supported by the Engineering and Physical Sciences Research Council (EPSRC) under EP/P006175/1. Benedict Leimkuhler is also a fellow of the Alan Turing Institute which is funded by grant EPSRC EP/N510129/1 and has benefited from this fellowship in the development of this work. Tiffany Vlaar is supported by The Maxwell Institute Graduate School in Analysis and its Applications, a Centre for Doctoral Training funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016508/01), the Scottish Funding Council, Heriot-Watt University and the University of Edinburgh. This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF).

References

[1] N. Ding, Y. Fang, R. Babbush, C. Chen, R.D. Skeel and H. Neven, Bayesian sampling using stochastic gradient thermostatsm, *NIPS* (2014), 3203–3211.

[2] I.J. Goodfellow, O. Vinyals and A.M. Saxe, Qualitatively characterizing neural network optimization problems, *ICLR* (2015).

[3] W.R. Huang, Z. Emam, M. Goldblum, L. Fowl, J.K. Terry, F. Huang and T. Goldstein, Understanding generalization through visualizations, arXiv 1906.03291 (2019).

[4] D.J. Im, M. Tao and K. Branson, An empirical analysis of deep network loss surfaces, *CoRR*, arXiv 1612.04010 (2016).

[5] A. Jones and B. Leimkuhler, Adaptive stochastic methods for sampling driven molecular systems, *The Journal of Chemical Physics*, **135** (2011).

[6] D. King, Dlib-ml: A machine learning toolkit, *Journal of Machine Learning Research*, **10** (2009), 1755–1758.

[7] D.P. Kingma and J. Ba, Adam: A method for stochastic optimization, *ICLR* (2015).

[8] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680.

[9] J. Lan, R. Liu, H. Zhou and J. Yosinski, LCA: Loss change allocation for neural network training, preprint, arXiv 1909.01440 (2019).

[10] B. Leimkuhler and C. Matthews, *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, Interdisciplinary Applied Mathematics, Springer, 2015.

[11] B. Leimkuhler, C. Matthews and G. Stoltz, The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics, *IMA Journal of Numerical Analysis*, **36** (2015), 13–79.

[12] B. Leimkuhler, C. Matthews and T. Vlaar, Partitioned integrators for thermodynamic parameterization of neural networks, preprint, arXiv 1908.11843 (2019).

[13] R.M. Neal, *Bayesian Learning for Neural Networks*, Springer-Verlag, New York, 1996.

[14] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, Automatic differentiation in PyTorch, (2017).

[15] E. Pollak, A. Auerbach and P. Talkner, Observations on Rate Theory for Rugged Energy Landscapes, *Biophysical Journal*, **95** (2008), 4258–4265.

[16] M. Sachs, B. Leimkuhler and V. Danos, Langevin Dynamics with variable coefficients and nonconservative forces: from stationary states to numerical methods, *Entropy*, **19** (2017).

[17] G. Stoltz, M. Sachs and B. Leimkuhler, Hypocoercivity properties of adaptive Langevin dynamics, preprint, arXiv 1908.09363 (2019).

[18] T. Tieleman and G. Hinton, Lecture 6.5 - RMSprop: Divide the gradient by a running average of its recent magnitude, *COURSERA: Neural Networks for Machine Learning* (2012).

[19] M. Welling and Y.W. Teh, Bayesian learning via stochastic gradient Langevin dynamics, *Proceedings of the 28th International Conference on Machine Learning* (2011), 681–688.

[20] R. Zwanzig, Diffusion in a rough potential, *Proc. Natl. Acad. Sci. USA*, **87** (1988), 2029–2030.