Continuous-Fidelity Bayesian Optimization with Knowledge Gradient*

Jian Wu & Peter I. Frazier
School of Operations Research and Information Engineering
Cornell University
Ithaca, NY, 14853, USA
{jw926,pf98}@cornell.edu

Abstract

While Bayesian optimization (BO) has achieved great success in optimizing expensive-to-evaluate black-box functions, especially tuning hyperparameters of neural networks, methods such as random search [13] and multi-fidelity BO (e.g. Klein et al. [10]) that exploit cheap approximations, e.g. training on a smaller training data or with fewer iterations, can outperform standard BO approaches that use only full-fidelity observations. In this paper, we propose a novel Bayesian optimization algorithm, the continuous-fidelity knowledge gradient (cfKG) method, that can be used when fidelity is controlled by one or more continuous settings such as training data size and the number of training iterations. cfKG characterizes the value of the information gained by sampling a point at a given fidelity, choosing to sample at the point and fidelity with the largest value per unit cost. Furthermore, cfKG can be generalized, following Wu et al. [23], to settings where more than one point can be evaluated simultaneously. Numerical experiments show that cfKG outperforms state-of-art algorithms when tuning convolutional neural networks (CNNs) on CIFAR-10 and SVHN.

1 Introduction

In hyperparameter tuning of machine learning models, we seek to find a set of hyperparameters x in some set $\mathbb{A}$ to minimize the validation error $f(x)$, i.e., to solve

$$\min_{x\in\mathbb{A}} f(x) \tag{1.1}$$

Evaluating $f(x)$ can take substantial time [1], and may not provide gradient evaluations.

As the computational expense of training and testing a modern deep neural network for a single set of hyperparameters has grown as long as days or weeks, it has become natural to seek ways to solve (1.1) more quickly by supplanting some evaluations of $f(x)$ with computationally inexpensive low-fidelity approximations. Indeed, when training a neural network or most other machine learning models, we can approximate $f(x)$ by training on less than the full training data, or using fewer training iterations. Both of these controls on fidelity can be set to achieve either better accuracy or lower computational cost across a range of values reasonably modeled as continuous.

In this paper, we consider optimization with evaluations of multiple fidelities and costs where the fidelity is controlled by one or more continuous parameters. We model these evaluations by a real-valued function $g(x, s)$ where $f(x) := g(x, 1_m)$ and $s \in [0, 1]^m$ denotes the m fidelity-control parameters. $g(x, s)$ can be evaluated, optionally with noise, at a cost that depends on x and s. In the context of hyperparameter tuning, we may take $m = 2$ and let $g(x, s_1, s_2)$ denote the loss on the

*This is a shorter non-archival version of a paper submitted to ICLR 2018.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

validation set when training using hyperparameters x with a fraction s_1 of the training data and a fraction s_2 of some maximum allowed number of training iterations. We may also set $m = 1$ and let s index either training data or training iterations.

Existing literature on multi-fidelity Bayesian optimization include [6, 11, 20, 8, 16] for discrete-fidelity settings and [2, 21, 10, 14, 9] for continuous-fidelity settings.

In this continuous-fidelity setting, we use the knowledge gradient (KG) approach [4], together with a computational technique using the envelope theorem developed in Wu et al. [23], to adaptively select the hyperparameter configurations and fidelities to evaluate in parallel that best support solving (1.1). This set of points maximize the ratio of the value of information from evaluation against its cost. Code is available at `https://github.com/wujian16/Cornell-MOE`.

Unlike most existing work on discrete- and continuous-fidelity Bayesian optimization, Our approach considers the impact of our measurement on the future posterior distribution over the full feasible domain, while existing expected-improvement-based approaches consider its impact at only the point evaluated. One exception is the entropy-search-based method [10], which also considers the impact over the full posterior. Our approach differs from entropy search in that it chooses points to sample to directly minimize expected simple regret, while entropy search seeks to minimize the entropy of the location or value of the global optimizer, indirectly reducing simple regret.

Below, §2 presents the cfKG method and §3 tests cfKG on hyperparameter tuning for deep learning.

2 Continuous-fidelity knowledge gradient

In this section, we propose the continuous-fidelity knowledge gradient (cfKG), a novel Bayesian optimization algorithm that exploits inexpensive low-fidelity approximations. To describe cfKG in detail, §2.1 first describes Gaussian process regression for modeling both $g(x, s)$ and its cost of evaluation. Then, §2.2 presents the cfKG acquisition function, which values sampling a (point, fidelity) pair according to the ratio of the value of the information gained from sampling that point at that fidelity, to the cost of doing so. §2.3 generalizes an envelope-theorem based computational technique developed in Wu et al. [23] to efficiently maximize this acquisition function.

2.1 Gaussian Processes

We put a Gaussian process (GP) prior [17] on the function g or its logarithm. We describe this procedure placing the prior on g directly, and then discuss below when we recommend instead placing it on $(x, s) \mapsto \log g(x, s)$. The GP prior is defined by its mean function $\mu^{(0)} : \mathbb{A} \times [0, 1]^m \mapsto \mathbb{R}$ and kernel function $K^{(0)} : \{\mathbb{A} \times [0, 1]^m\} \times \{\mathbb{A} \times [0, 1]^m\} \mapsto \mathbb{R}$.

We assume that evaluations of $g(x, s)$ are subject to additive independent normally distributed noise with common variance σ^2. We treat the parameter σ^2 as a hyperparameter of our model. Our assumption of normally distributed noise with constant variance is common in the BO literature [10].

The posterior distribution of g after n function evaluations at points $z^{(1:n)} := \{(x^{(1)}, s^{(1)}), (x^{(2)}, s^{(2)}), \cdots, (x^{(n)}, s^{(n)})\}$ with observed values $y^{(1:n)} := \{y^{(1)}, y^{(2)}, \cdots, y^{(n)}\}$ remains a Gaussian process [17], and $g \mid z^{(1:n)}, y^{(1:n)} \sim \text{GP}(\mu^n, K^{(n)})$. This statistical approach contains several hyperparameters: the variance σ^2, and any parameters in the mean and kernel functions. We treat these hyperparameters in a Bayesian way as proposed in Snoek et al. [19].

When g is the validation error in a hyperparameter optimization problem, we recommend putting a GP prior on $\log g(x, s)$, rather than on $g(x, s)$ directly, because (1) $g(x, s)$ is nonnegative and will be allowed to be negative after log scaling, better matching the range of values assumed by the GP, and (2) because $g(x, s)$ can climb steeply over several orders of magnitude as we move away from the optimal x, making $\log g(x, s)$ easier to model. We analogously train a separate GP on the logarithm of the cost of evaluating $g(x, s)$.

2.2 The cfKG acquisition function

cfKG samples the point and fidelity that jointly maximize an acquisition function, which we define in this section by adopting the knowledge gradient concept [4] in the continuous-fidelity setting to value the information gained through one additional sample.

If we were to stop sampling after n samples, we would select as our solution to (1.1) a point x with minimum estimated validation error $\mu^{(n)}(x, 1_m)$, and this point would have a conditional expected validation error of $\min_{x\in\mathbb{A}} \mu^{(n)}(x, 1_m)$ under the posterior. If instead we took an additional sample at $x^{(n+1)}$ with the fidelity $s^{(n+1)}$, then the minimum expected validation error under the resulting posterior would become $\min_{x\in\mathbb{A}} \mu^{(n+1)}(x, 1_m)$. This quantity depends on $x^{(n+1)}$ and $s^{(n+1)}$ through the dependence of $\mu^{(n+1)}(x, 1_m)$ on the point and fidelity sampled, and is random under the posterior at iteration n because $\mu^{(n+1)}(x, 1_m)$ depends on the observation $y^{(n+1)}$. We discuss this dependence explicitly in §2.3.

The value of the information gained by sampling at $x^{(n+1)}$ with the fidelity $s^{(n+1)}$ conditioned on any particular outcome $y^{(n+1)}$ is thus the difference of these two expected validation errors $\min_{x\in\mathbb{A}} \mu^{(n)}(x, 1_m) - \min_{x\in\mathbb{A}} \mu^{(n+1)}(x, 1_m)$. We then take the expectation of this difference, over the random outcome $y^{(n+1)}$, to obtain the (unconditional) value of the information gained, and take the ratio of this value with the cost of obtaining it to obtain the cfKG acquistion function,

$$\text{cfKG}(x, s) = \frac{\min_{x'\in\mathbb{A}} \mu^{(n)}(x', 1_m) - \mathbb{E}_n \left[\min_{x'\in\mathbb{A}} \mu^{(n+1)}(x', 1_m) \mid x^{(n+1)} = x, s^{(n+1)} = s\right]}{\text{cost}^{(n)}(x, s)}, \quad (2.1)$$

where $\text{cost}^{(n)}(x, s)$ is the estimated cost of evaluating at x with the fidelity s based on the observations available at iteration n, according to the GP described in §2.1, and $\mathbb{E}_n$ indicates the expectation taken with respect to the posterior given $x^{(1:n)}, s^{(1:n)}, y^{(1:n)}$.

The cfKG algorithm chooses to sample at the point x and fidelity s that jointly maximize $\text{cfKG}(x, s)$.

$$\max_{(x,s)\in\mathbb{A}\times[0,1]^m} \text{cfKG}(x, s). \quad (2.2)$$

Although this acquisition function considers the expected value of an improvement due to sampling, it differs from expected improvement approaches such as Lam et al. [11] because the point at which an improvement occurs, $\text{argmax}_{x\in\mathbb{A}}\mu^{(n+1)}(x, 1_m)$ may differ from the point sampled. Moreover, this acquisition function allows joint valuation of both the point x and the fidelity s, while approaches such as Lam et al. [11] require valuing a point x assuming it will be evaluated at full fidelity and then choose the fidelity in a second stage.

cfKG generalizes naturally to batch settings where we can evaluate multiple (point, fidelity) pairs at once. We value joint evaluation of $q \geq 1$ points $x_{1:q}$ at fidelities $s_{1:q}$, where $z_{1:q} = ((x_1, s_1), \ldots, (x_q, s_q))$, by

$$\text{q-cfKG}(z_{1:q}) = \frac{\min_{x'\in\mathbb{A}} \mu^{(n)}(x', 1_m) - \mathbb{E}_n \left[\min_{x'\in\mathbb{A}} \mu^{(n+q)}(x', 1_m) \mid z^{(n+1:n+q)} = z_{1:q}\right]}{\max_{1\leq i\leq q} \text{cost}^{(n)}(x_i, s_i)}, \quad (2.3)$$

We then modify (2.2) by sampling at the batch of points and fidelities that maximize

$$\max_{z_{1:q}\subset\mathbb{A}\times[0,1]^m} \text{q-cfKG}(z_{1:q}) \quad (2.4)$$

2.3 Envelope-theorem-based computational method

In this section, we describe computational methods for solving (2.2) and (2.4). We describe our method in the context of (2.4), and observe that (2.2) is a special case.

We generalize a recently proposed envelope-theorem based computational method developed for single-fidelity optimization in [23], which is used to provide unbiased estimators of both q-cfKG and its gradient. We then use stochastic gradient ascent to optimize the q-cfKG acquisition function.

Following [23], the q-cfKG acquisition function can be expressed as

$$\text{q-cfKG}(z_{1:q}) = \frac{\min_{x\in\mathbb{A}} \mu^{(n)}(x, 1_m) - \mathbb{E}_n \left[\min_{x\in\mathbb{A}} \left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q}) W_q\right)\right]}{\max_{1\leq i\leq q} \text{cost}^{(n)}(x_i, s_i)},$$

where W_q is a standard q-dimensional normal random vector, $\tilde{\sigma}_n(x, z_{(1:q)}) = K^{(n)}((x, 1_m), z_{1:q}) (D^{(n)}(z_{1:q})^T)^{-1}$, and $D^{(n)}(z_{1:q})$ is the Cholesky factor of the

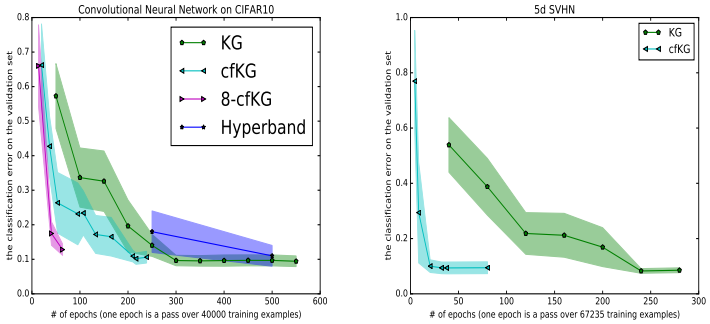

Figure 1: Tuning convolutional neural networks on CIFAR-10 and SVHN with 6 independent runs

covariance matrix $K^{(n)}(z_{1:q}, z_{1:q}) + \sigma^2 I$. ∇q-cfKG$(z_{1:q})$ can be computed from $-\nabla \mathbb{E}_n\left[\min_{x\in\mathbb{A}}\left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q})\, W_q\right)\right]$ and $\nabla \max_{1\le i\le q} \text{cost}^{(n)}(z^{(n+i)})$, where differentiability of $\text{cost}^{(n)}(\cdot)$ implies $\max_{1\le i\le q} \text{cost}^{(n)}(x_i, s_i)$ is differentiable almost everywhere. To compute the first term, under sufficient regularity conditions [12] that we conjecture hold in most applications to hyperparameter tuning, one can interchange the gradient and expectation operators,

$$\mathbb{E}_n\left[\min_{x\in\mathbb{A}}\left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q})\, W_q\right)\right] = \mathbb{E}_n\left[\nabla \min_{x\in\mathbb{A}}\left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q})\, W_q\right)\right]. \quad (2.5)$$

This technique is called infinitesimal perturbation analysis (IPA) [12].

Since multiplication, matrix inversion (when the inverse exists), and Cholesky factorization [18] preserve continuous differentiability, $(x, z_{1:q}) \mapsto \left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q}) W_q\right)$ is continuously differentiable under mild regularity conditions. When this function is continuously differentiable and $\mathbb{A}$ is compact, the envelope theorem [15, Corollary 4] implies

$$\begin{aligned}
&\mathbb{E}_n\left[\nabla \min_{x\in\mathbb{A}}\left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q})\, W_q\right)\right] \\
=&\mathbb{E}_n\left[\nabla\left(\mu^{(n)}(x^*(W_q), 1_m) + \tilde{\sigma}_n(x^*(W_q), z_{1:q}) \cdot W_q\right)\right], \\
=&\mathbb{E}_n\left[\nabla \tilde{\sigma}_n(x^*(W_q), z_{1:q}) \cdot W_q\right],
\end{aligned}$$

where $x^*(W_q) \in \arg\min_{x\in\mathbb{A}}\left(\mu^{(n)}(x, 1_m) + \tilde{\sigma}_n(x, z_{1:q}) W_q\right)$. We can use this unbiased gradient estimator within stochastic gradient ascent [5] to solve the optimization problem (2.4).

3 Experiments: tuning convolutional neural nets on CIFAR-10 and SVHN

In this section, our benchmarks include the traditional Bayesian optimization algorithms KG [22] and EI [7]. We also compare with Hyperband [13] in the CIFRA-10 experiment. We use squared-exponential kernels with constant mean functions and integrate out the GP hyperparameters by sampling $M = 10$ sets of hyperparameters using the `emcee` package [3].

We tune convolution neural networks (CNNs) on CIFAR-10 and SVHN. Our CNN consists of 3 convolutional blocks and a softmax classification layer. Each convolutional block consists of two convolutional layers with the same number of filters followed by a max-pooling layer. There is no dropout or batch-normalization layer. We split the CIFAR-10 dataset into 40000 training samples, 10000 validation samples and 10000 test samples. We split the SVHN training dataset into 67235 training samples and 6000 validation samples, and use the standard 26032 test samples. We apply standard data augmentation: horizontal and vertical shifts, and horizontal flips. We optimize 5 hyperparameters to minimize the classification error on the validation set: the learning rate, batch size, and number of filters in each convolutional block. cfKG and q-cfKG use two fidelity controls: the size of the training set and the number of training iterations. Hyperband uses the size of the training set as its resource (it can use only one resource or fidelity), using a bracket size of $s_{\max} = 4$ as in Li et al. [13] and the maximum resource allowed by a single configuration set to 40000. We set the maximum number of training epochs for all algorithms to 50 for CIFAR-10 and 40 for SVHN. Fig. 1 shows the performance of cfKG relative to several benchmarks. cfKG successfully exploits the

cheap approximations and find a good solution much faster than KG and Hyperband. When we train using optimized hyperparameters on the full training dataset for 200 epochs, test data classification error is $\sim 12\%$ for CIFAR-10 and $\sim 5\%$ for SVHN.

References

[1] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[2] T. Domhan, J. T. Springenberg, and F. Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *IJCAI*, pages 3460–3468, 2015.

[3] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013.

[4] P. Frazier, W. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

[5] J. Harold, G. Kushner, and G. Yin. *Stochastic approximation and recursive algorithm and applications*. Springer, 2003.

[6] D. Huang, T. Allen, W. Notz, and R. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, 2006.

[7] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[8] K. Kandasamy, G. Dasarathy, J. B. Oliva, J. Schneider, and B. Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Advances in Neural Information Processing Systems*, pages 992–1000, 2016.

[9] K. Kandasamy, G. Dasarathy, J. Schneider, and B. Poczos. Multi-fidelity bayesian optimisation with continuous approximations. In *ICML*, 2017. Accepted for publication. ArXiv preprint 1703.06240.

[10] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, 2017. Accepted for publication. ArXiv preprint arXiv:1605.07079.

[11] R. Lam, D. Allaire, and K. Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143, 2015.

[12] P. L'Ecuyer. A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science*, 36(11):1364–1383, 1990.

[13] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *arXiv preprint arXiv:1603.06560*, 2016.

[14] M. McLeod, M. A. Osborne, and S. J. Roberts. Practical bayesian optimization for variable cost objectives. *arXiv preprint arXiv:1703.04335*, 2017.

[15] P. Milgrom and I. Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2): 583–601, 2002.

[16] M. Poloczek, J. Wang, and P. I. Frazier. Multi-information source optimization. In *Advances in Neural Information Processing Systems*, 2017. Accepted for publication. ArXiv preprint 1603.00389.

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN ISBN 0-262-18253-X.

[18] S. P. Smith. Differentiation of the cholesky algorithm. *Journal of Computational and Graphical Statistics*, 4(2):134 – 147, 1995.

[19] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[20] K. Swersky, J. Snoek, and R. P. Adams. Multi-task bayesian optimization. In *Advances in neural information processing systems*, pages 2004–2012, 2013.

[21] K. Swersky, J. Snoek, and R. P. Adams. Freeze-thaw bayesian optimization. *arXiv preprint arXiv:1406.3896*, 2014.

[22] J. Wu and P. Frazier. The parallel knowledge gradient method for batch bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 3126–3134, 2016.

[23] J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, 2017. Accepted for publication. ArXiv preprint 1703.04389.