Comparison of Optimization Techniques for Stateful Black Boxes

Jerome Rasky 42386145, Tristan Rice 25886145

Abstract

Current research into global black box optimization of expensive functions focuses on stateless models such as machine learning hyper parameter optimization. Many real world problems can't be reset back to a clean state after a single evaluation. In this paper, we evaluate the performance of five popular black box algorithms on two stateful models.

Introduction

For our project, we were interested in the area of black-box optimization. Many engineering simulations and problems exist in a complex problem space without easy access to gradients, and often where function evaluation itself is extremely expensive. In these kinds of situations, easy solutions like gradient descent are out of scope, and others like Monte Carlo methods are prohibitively expensive.

We wanted to explore the specific case of stateful black-boxes. These are functions whose output is noisy, and whose noise depends on previous parameters. This includes situations like government budgeting, where each year feels the impact of previous years, or flying an airplane, where each second the plane continues to feel the impact of previous control movements.

By exploring various methods on these stateful models, we hope to gain an understanding for how existing methods fare in the presence of stateful models.

Related Work

Pattarwat Chormai (2017) and Y. Chen et al. (2016)

These papers implement an LSTM model to learn how to minimize the function, based on an approximation built from Gaussian processes. The authors claim great performance in their papers. In practice, the method they present is very sensitive to initialization and to the kernel used. This is not unexpected in the space of neural networks.

Hansen et al. (2010)

There is a considerable amount of prior work in the area of black-box optimization. This paper in particular compares 31 methods against the BBOB-2009 functions, which are particularly difficult to optimize. These methods are not stateful, however, which is the drive of our project.

The findings from this paper are in general that multi-modal, non-smooth, high-dimensional functions are tough to optimize, while uni-modal, smooth, low-dimensional functions are easier to optimize. Similarly, functions that are highly stateful are likely tougher to optimize, as evidenced by our sampling of methods.

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

Description and Justification

Algorithms

Random Search

Random search is one of the simplest black box optimization algorithms. It randomly guesses points within the bounds and then takes the highest value as the best option. This is used as a baseline for the other algorithms.

Tree of Parzen Estimators

Gradient Boosted Decision Trees

Bayesian Optimization (Gaussian Processes)

LSTM based Recurrent Neural Network

We implemented the model as described in Pattarwat Chormai (2017). The problem of black box optimization can be formulated as the problem of finding the sequence containing the minimum value of a black box function. This formulation can be used to fit a Long Short Term Memory neural network. This method essentially uses an LSTM to learn how to minimize the function, rather than using a gradient.

At every step, the rnn $LSTM$ determines the next step to take.

$$\begin{aligned} x_n, h_n &= LSTM(x_{n-1}, y_{n-1}, h_{n-1}) \\ y_n &= \psi(x_n) \end{aligned}$$

Where $\psi(x) = E[GP(x)]$, the expected value of the Gaussian process model at point x.

Our implementation of this method uses Tensorflow (Abadi et al. 2015), making use of its LSTM framework. We also make use of GPflow (Matthews et al. 2017) for creating our Gaussian processes.

This method is obviously heavily dependent on the exact form of the Gaussian process. We started with the simplest case of a "vanilla" Gaussian process using radial basis functions for the kernels. This approach yielded poor results, since with only a few data points to train on, most of the function space is predicted to be zero. Using a summed Matern and Linear kernel, performance can beat Bayesian optimization.

Models

To evaluate these algorithms we trained them on two stateful models. Each model has a single function that needs to be optimized with n parameters as well as bounds for the range of values that can be accepted.

Death Rate

Many real life models, such as governmental budgeting are nearly impossible to evaluate so we had to come up with an approximation to it. We created a simple model by using the World Bank Development Indicators and training a Gradient Boosted Decision Tree to predict deathrate based off of expenditures in education, health, R&D and military (Bank 2018). To add a stateful component to this, we added momentum, such that changing the parameters produces lag with respect to the death rate.

While this is a very simple model, it does provide some realistic behaviors. Many large systems have a high latency between cause and effect. We're also primarily interested in highlighting the differences between these algorithms.

We bound the inputs to be within two times the maximum existing value for that expenditure category and greater than zero.

Airplane

The second model is that of flying an airplane. The Python Flight Mechanics Engine is a project attempting to model every aspect of flying a plane (Team 2018). We used it to model flying a plane to a location. The model takes in the throttle as well as the position of the elevator, aileron and rudder and outputs the distance the plane has flown towards the target. This model has numerous stateful variables that need to be modeled including position, rotation, velocity, direction. There are also many nonlinearities due to air resistance and gravity.

This model uses the Cessna 172 as a base and bounds the inputs to be match the actual control range.

Experiments and Analysis

Commonly Bayesian Optimization methods start out with a series of random points to get an initial overview of the space. Our data shows that this leads to very poor performance with stateful models as 5 initial bad points can lead the model to a poor state before being able to recover.

Death Rate

Airplane

Discussion and Future Work

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." https://www.tensorflow.org/.

Bank, The World. 2018. "Development Indicators." http://data.worldbank.org/indicator.

Chen, Yutian, Matthew W Hoffman, Sergio Gómez Colmenarejo, Misha Denil, Timothy P Lillicrap, Matt Botvinick, and Nando de Freitas. 2016. "Learning to Learn Without Gradient Descent by Gradient Descent." *arXiv Preprint arXiv:1611.03824*.

Hansen, Nikolaus, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. "Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking Bbob-2009." In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, 1689–96. ACM.

Matthews, Alexander G. de G., Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. 2017. "GPflow: A Gaussian process library using TensorFlow." *Journal of Machine Learning Research* 18 (40): 1–6. http://jmlr.org/papers/v18/16-537.html.

Pattarwat Chormai, Raphael Holca-Lammare, Felix Sattler. 2017. "Black Box Optimization Using Recurrent Neural Networks," July. http://www.ni.tu-berlin.de/fileadmin/fg215/teaching/nnproject/black-box-optimization-final.pdf.

Team, AeroPython. 2018. "Python Flight Mechanics Engine." https://github.com/AeroPython/PyFME.