Learning Hyper-Parameters with Bayesian Optimization

Julien Brenneck

CS682, Spring 2018

Introduction

- Optimizing hyper-parameters is a key aspect of training modern Neural Network based models.
- By exploiting information from previous trials, Bayesian Optimization can outperform random search for CNN hyper-parameters.
- Gaussian Processes give a practical model for Bayesian Optimization of Neural Networks.

Bayesian Optimization

In Bayesian Optimization [1] we are trying to find the minima of a function $f(\theta)$, that is

$$\theta^\star = \arg\min_{\theta \in \mathcal{X}} f(\theta). \tag{1}$$

Here we are trying to minimize validation error, where θ is the model hyper-parameters.

The core of Bayesian Optimization is building and updating a model for sequentially finding an optimal θ. Assuming a **prior** $p(\theta)$, and given **data** $\mathcal{D}$ and **likelihood** model $p(\mathcal{D} \mid \theta)$ we infer a **posterior** distribution $p(\theta \mid \mathcal{D})$ using Bayes rule:

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{p(\mathcal{D})} \tag{2}$$

By choosing our prior to be **self-conjugate** the posterior will be in the same family of distributions as the prior. We use a **Gaussian Process** model, which is self-conjugate, allowing for an analytic solution of the updated posterior.

Gaussian Process Priors

A Gaussian Process is a generalization of a multivariate Gaussian, characterized by its mean function $\mu(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$ and covariance function (kernel) $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$.

This allows us to characterize uncertainty. The Gaussian Process prior is updated with data, resulting in a Gaussian Process posterior, which becomes the prior in the next iteration. We see below how the model is updated as new trails are added.

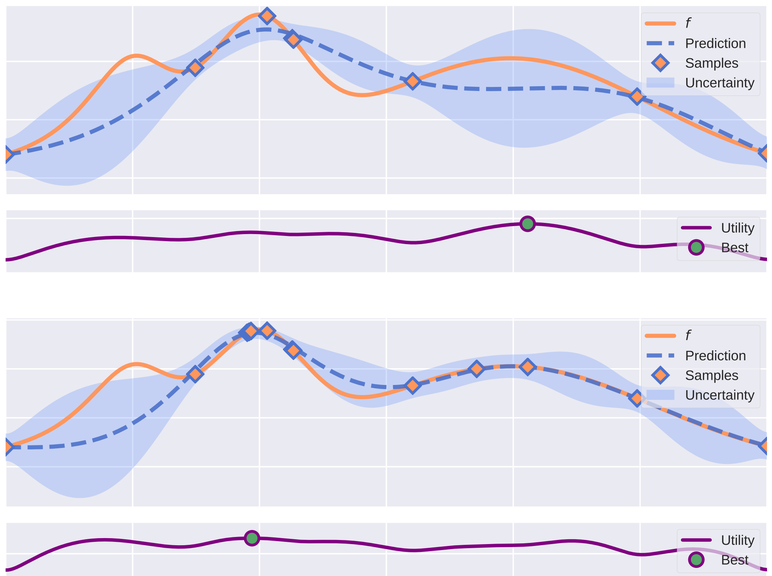

Figure 1: Optimizing a function f with Gaussian Processes

The next sample point is chosen by maximizing the aquisition function. A standard choice is to use **Expected Improvement** [2], which maximizes for θ_{n+1}, the hyper-parameters for the next interation,

$$\theta_{n+1} = \arg\max_{\theta} \mathbf{E}\big[\max(0, f(\theta) - f(\theta_n)) \mid \mathcal{D}_n\big]. \tag{3}$$

Covariance Functions

The choice of **kernel** determines the structure of the Gaussian process. We use a standard Matérn kernel:

$$k_{\text{MATÉRN3}}(\mathbf{x}, \mathbf{x}') = \sigma_0^2 \exp\left(-\sqrt{3r^2}\right)\left(1 + \sqrt{3r^2}\right) \tag{4}$$

$$r^2 = (\mathbf{x} - \mathbf{x}')^T \Lambda (\mathbf{x} - \mathbf{x}') \tag{5}$$

Conclusion

Several open source implementations of Gaussian Proccess Bayesian Optimization exist today:

Package	URL
BayesianOptimization	https://github.com/fmfn/BayesianOptimization
Spearmint	https://github.com/HIPS/Spearmint
MOE	https://github.com/Yelp/MOE

The contribution of this work is to show that Bayesian Optimization can be applied in practice to the training of modern Convolutional Neural Networks today.

Previous work [2, 3] has already shown that Bayesian hyper-parameter optimization can reach or surpass standard techniques such as grid search, random search, and search by a human expert.

References

[1] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas.
Taking the Human Out of the Loop: A Review of Bayesian Optimization.
Proceedings of the IEEE, 104(1):148–175, January 2016.

[2] James S. Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl.
Algorithms for Hyper-Parameter Optimization.
In Advances in Neural Information Processing Systems 24, pages 2546–2554. Curran Associates, Inc., 2011.

[3] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams.
Practical Bayesian Optimization of Machine learning Algorithms.
In Advances in Neural Information Processing Systems 25, 12/2012 2012.