Bayesian Optimization for Hyperparameter Tuning in the Fully Connected Layers of VGG16

Søren Winkel Holm Oskar Wiese Anders Henriksen
Anne Agathe Pedersen

February 27, 2020

1 Abstract

2 Introduction

The accuracy of a neural network can be highly dependent on the hyperparameters used. However finding the best parameter values can be a costly process. In this project we will use Bayesian optimization to

USE B O to find best combination of

Parameter optimization sucks but has a huge impact on how good a neural network is...

Humans are slow and faulty. This means that work costs heaps of money and mistakes happen often. To help mitigate this, machine learning can learn the task at hand in order to optimize the performance at a much lower cost. While effective for many applications, machine learning has one glaring issue that requires skill and tonnes of computing power to overcome; hyperparameter optimization. Optimizing the hyperparameters usually devolves into random guessing, though new methods are on the rise like bayesian optimization, which greatly reduces the effort involved in finding the parameters with use of an acquisition function and probabilistic model.

In this paper, baysian optimization with bayesian processes as probabilistic model and expected improvement, upper confidence bound and probability of improvement as acquisition function will be used to find the hyperparameters of the VGG16 classifier network when training to classify on the 10 classes of the CIFAR10 dataset, optimizing for the validation accuracy.

acquisition function

3 Methods

Gaussian process and Bayesian optimization...

Three different aquisition functions + random...

Acquisition Functions

Probability of Improvement

Probability of improvement evaluates *f*, the objective function, at the point most likely to improve the minimum value. In this project, the objective function returns the negative validation accuracy. Probability of improvement then evaluates at the point most likely to obtain a higher accuracy, since the the function is given the negative accuracy. Mathematically PI can be expressed as,

$$\mathrm{PI}(\mathbf{x}) = \mathrm{P}(f(\mathbf{x}) \geq f(\mathbf{x}^{+}) + \xi) = \Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^{+}) - \xi}{\sigma(\mathbf{x})}\right)$$

where ξ is a hyperparameter which can be set by the user. The hyperparameter is a trade off between exploration and exploitation. Φ is the cumulative distribution function of the Gaussian distribution.

Expected Improvement

The expectorated improvement tries to quantify the improvement instead of the probability of improving as Probability of Improvement does. It is possible to quantify the average value of improvement if we sample the objective function at x. The acquisition function then computes the expected value of the improvement function at a certain x. Mathematically EI can be expressed as,

$$\mathrm{EI}(\mathbf{x}) = \begin{cases} \left(\mu(\mathbf{x}) - f\left(\mathbf{x}^{+}\right)\right)\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) < 0 \end{cases}$$

Here the variable $Z = \frac{\mu(\mathbf{x}) - f(\mathbf{x}^{+}) - \xi}{\sigma(\mathbf{x})}$

Gaussian Process Upper Confidence Bound

The idea behind the GP-UCB is to directly use the mean and variance from the Gaussian distribution. These two parameters are used as the exploration / exploitation trade off. The function can be formulated as the following,

$$UCB(x) = \mu(x) + \kappa\sigma(x)$$

4 Results

Comparison of aquisition functions and random...

Results and plots...

5 Discussion

Which one is better...