Bayesian Approach for Gradient-free Optimization in Deep Learning

Aukkawut Ammartayakun (374259496)
aammartayakun@wpi.edu
CS 539 Machine Learning

Spring 2023

1 Introduction

Deep learning has revolutionized numerous fields such as image recognition, natural language processing, and speech recognition due to its ability to learn complex patterns and representations from large-scale data. Traditionally, deep learning models rely on gradient-based optimization techniques, with backpropagation being the most common method for adjusting the weights of neural networks.

Bayesian optimization is a gradient-free optimization method that does not assume the differentiability of the objective function. Moreover, with the black-box function view in the Bayesian optimization[3], the objective function can be any function that can be evaluated which can be helpful in the case of deep learning where the loss function is not differentiable or the gradient is hard to compute. This will help solving the problem of finding the differentiable loss function to solve the deep learning problem.

Although gradient-based method is the best method as one can optimize any differentiatable function $f(x)$ with gradient descent or newton-raphson method. However, this work will explore the possibility of using gradient-free optimization methods to explore the possibility of using gradient-free optimization methods to optimize deep learning models in place of backpropagation.

2 Dataset and Resources

The dataset that will be used in this project is the MNIST dataset[2] and CIFAR-10[1] dataset. The MNIST dataset is a dataset of handwritten digits with 60,000 training images and 10,000 testing images.

The simulated data from some analytical functions will also being used to test the performance of the Bayesian optimization on the feedforward neural network in the regression task.

All of the code will be written in Python and the libraries that will be used are: numpy, pandas, scipy, and matplotlib. Deep learning library like Pytorch or Tensorflow will not be used because we need to the control of the optimization method to be strictly based on the Bayesian optimization, not gradient-based method. However, Pytorch might be used if necessary.

3 Timeline

- Week 1 - 2 (Mar 21): Literature Review
- Week 3 (Apr 4): Apply Bayesian Optimization to Feedforward Neural Network on simulated data and MNIST dataset (Minimum)
- Week 4 (Apr 11): Apply Bayesian Optimization to Convolutional Neural Network on CIFAR-10 dataset
- Week 5 (Apr 18): Apply Bayesian Optimization to non-differentiable loss function like ℓ_1 difference loss in Siamese Network on MNIST dataset (Perfect work)

- Week 6 (Apr 25): Conclude the work and write the final report, and if possible, extend the work to reinforcement learning like Deep Q learning on Atari game like Breakout
- Week 7 (May 3): Final Report Due

References

[1] Krizhevsky, A. Learning multiple layers of features from tiny images. 32–33.

[2] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[3] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE 104*, 1 (2016), 148–175.