Evaluating the Effectiveness of Stochastic Gradient Descent in Neural Network Optimization

Peter Dall-Hansen[†] Alberte Bech Elholm[‡]

Jonas Jinlong Li[§]

https://github.com/peterdallhansen/dtu-sgd-gd-comparison

Abstract

A common problem in Machine Learning is overfitting, especially when training data is limited. In this paper, we explore the effectiveness of stochastic gradient descent compared to classical gradient descent on performance of neural networks, when optimizing for a complex non-linear function approximation. Using a simple neural network we compare the two methods using identical training conditions. Our results shows that Stochastic Gradient Descent not only leads to lower test loss but also reduces overall computation time.

1. Introduction

Stochastic Gradient Descent [1] (SGD) significantly reduces computation time by computing the gradient using only a small subsets of the data rather than the full dataset. Because the gradient estimate is noisy, SGD is less likely to get stuck in poor local minima. We hypothesize that using SGD leads to a lower test loss and reduce the overall computation time.

2. Methodology

As a benchmark for comparing GD and SGD, we implemented a simple neural network using the reference code from our course material, *09-NeuralNetworkChallenge.ipynb*. Both models were trained on the Challenge Dataset, generated from the function $f(x) = \sin\left(\frac{1}{x}\right)$ with added Gaussian noise, which consists of 50 training and 1000 test observations. For SGD we used a batch size of 1. We then trained both models 10 times for 10 000 epochs each. To compare the two methods, we computed the mean test loss across all 10 runs, using the mean-squared-loss as thes loss function, as well as the 95% confidence interval.

3. Results

The mean test loss, confidence intervals and mean computation time, for both optimization methods across all 10 runs:

Method	Mean test loss	95% CI	Mean computation time
SGD	0.221	[0.217, 0.224]	5.89s
GD	0.271	[0.270, 0.272]	73.93s

Table 1: Mean test loss, confidence intervals and mean computation time.

Figure 1 shows how the mean test loss evolves during training for both optimization methods as well as the 95% CI.:

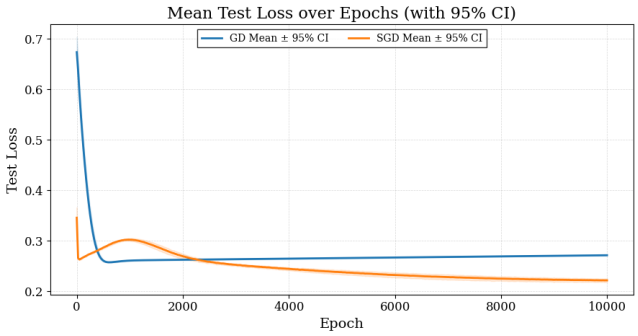

Figure 1: Mean test loss over 10,000 training epochs for Gradient Descent (GD) and Stochastic Gradient Descent (SGD)

4. Discussion

As presented in Table 1 SGD yielded a mean test loss in the range 0.2170 - 0.224, while GD yielded a mean test loss in the range of 0.270 - 0.272, which on average was around 0.05 higher. Our results confirm our hypothesis that SGD leads to a lower test loss as well as a lower overall computation time

5. Learning Outcome

We learned that SGD reaches lower test loss and requires less overall computation time than classical GD, and we improved our understanding of optimization methods using statistical analysis.

References

[1] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *Proceedings of COMPSTAT*. 2010, pp. 177–186.

[1] s255246@dtu.dk
[2] s256108@dtu.dk
[3] s256044@dtu.dk