Universität Hamburg
Department Informatik
Knowledge Technology, WTM

Comparison of Gradient Descent Optimization Methods for Neural Networks

Seminar Paper

Neural Networks

Ali Saleh

Matr.Nr. 6517831

3saleh@informatik.uni-hamburg.de

08.06.2017

Abstract

Gradient Descent is the most widely used optimization method for neural networks training. This paper aims to explore different algorithms to for gradient optimization. Using standard datasets and different neural network architectures time and complexity of the different algorithms will be compared and analyzed.

Contents

1 Introduction

In the scene of machine learning there are several key components to build a classification/regression model. Your model building usually involves several decision to make. After choosing your model (e.g.KNN, SVM, KART, NN, etc ...) you will begin to address the problem of how to choose your internal parameters (like the K in k nearest neighbor). For the context of neural networks (assuming you had already chosen your model architecture) you will have to decide on how many layers, how many nodes per layer, what kind of encoding yo use, and several other parameters depending on your architecture. [Schmidhuber, 2015] and [Arel et al., 2010] have both presented the general outline for neural networks modeling in more details.

Once your model parameters are decided on you will need to begin you model training and evaluation. This is the process of deciding if your model is performing well, also it can be used to compare several models together. In this setup evaluation is done by the mean of a loss function. It's used to asses the performance of you model (training data, and weights) against the ground truth of test data. Loss function is defined as :

$$L(X, Y, \grave{Y}) \tag{1}$$

Where X is your test data, Y is the ground truth , and $\grave{Y}$ is the output of your model on the test data.

In neural networks the output of your model depends highly on the weights of the connections between the network layers. This means that the notion of loss function is thus used to evaluate the goodness of a set of weights W used by a model. Thus the need for optimization, to find the set of weights that makes the loss function at minimum.

one of the most prominent strategies to optimize functions arises from calculus. From it's name loss function is an ordinary mathematical function in the sense that it can have a derivative. Using the mathematical analogy optimization can be achieved by moving in the direction of the gradient.

In a one variable function the first derivative is the rate of change of the function. Which can be generalized into multi-variable functions. The gradient is the rate of change of a multi-variable function in a specified set of directions. It's represented as a vector of numbers, with each element of it as a rate of change in a direction. For a 3 variable function

$$w = f(x, y, z) \tag{2}$$

it's gradient in the 3 variables directions is :

$$\nabla w = \left\langle \frac{\partial w}{\partial x}, \frac{\partial w}{\partial y}, \frac{\partial w}{\partial z} \right\rangle \tag{3}$$

where $\frac{\partial w}{\partial x}$ is the partial derivative of the function w in the direction of x. Moving the weights in the direction of the gradient leads to minimizing the loss function.

The reminder of this paper are organized as following. Section 2 will provide an overview of the gradient optimization methods, how to calculate them and what algorithms to compare. Section 3 will have a description of an experiment to compare the 3 algorithms together and then show the results of this experiment. Section 4 will conclude the paper.

2 Gradient Computing And Optimization

TODO Re write the whole section, add more info on the 3 algorithms
Introducing the methods of computing gradient and 3 different algorithms for optimizing it. Those 3 algorithms will be

- Stochastic Gradient Descent is the standard gradient optimization algorithm, it will be used as the base line to compare other performances with.
- Adam (Adaptive Moment Estimation) [Diederik P. Kingma, 2015]
- Adagrad (Adaptive Subgradient Methods for Online Learning and Stochastic Optimization) [John Duchi, 2011]

3 Results of Optimization Methods

TODO Add one more experiment with another dataset.
For comparing the 3 algorithms mentioned in section 2, an experiment were held to analyze their performance.For the experiment the CIFAR10 data set was used (see Figure 1. "The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images." [Krizhevsky, 2009].

The model built used the convolution neural network architecture. The model consists of the following layers. Each of the first 2 layers is a convolution layer with relu activation, followed by max-pooling layer, and dropout set 0.25 [Srivastava et al., 2014]. Those are followed by one fully-connected (dense) layer for classification, with relu activation, and dropout set to 0.5, Then a final fully connected softmax layer of 10 neurons are used for the output of the classification of images. The training is carried on with batches of size 32 over 200 epoch. The loss function used is the categorical coss-entropy. An overview of the model is presented at Appendix A (figure 2).

3.1 Results

TODO add more analysis results
The 5 metrics that was used for the comparison are:

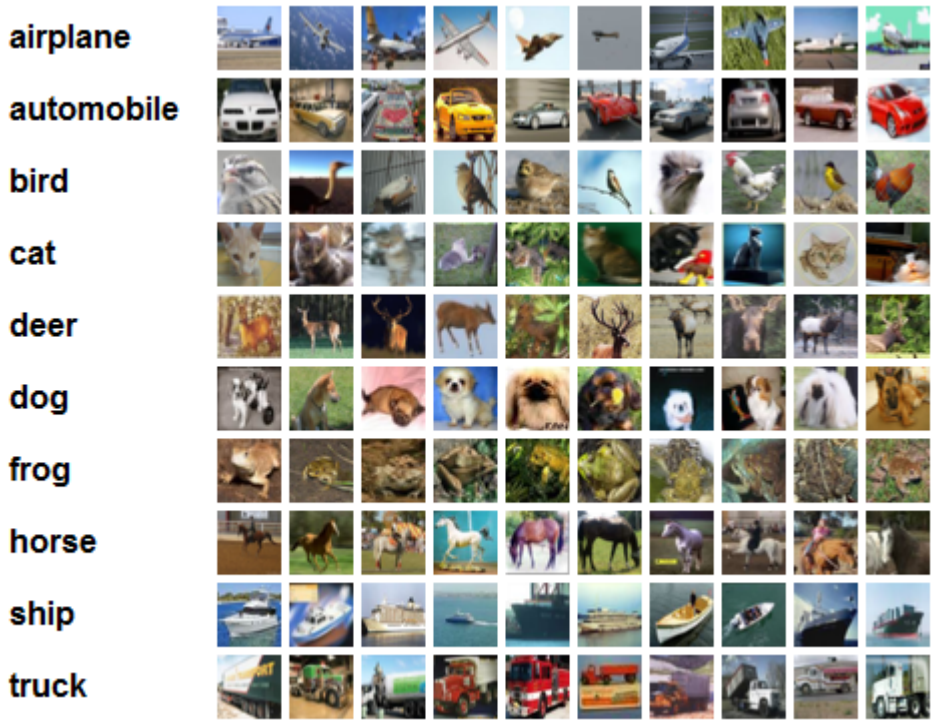
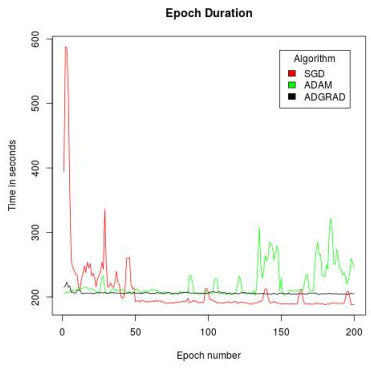
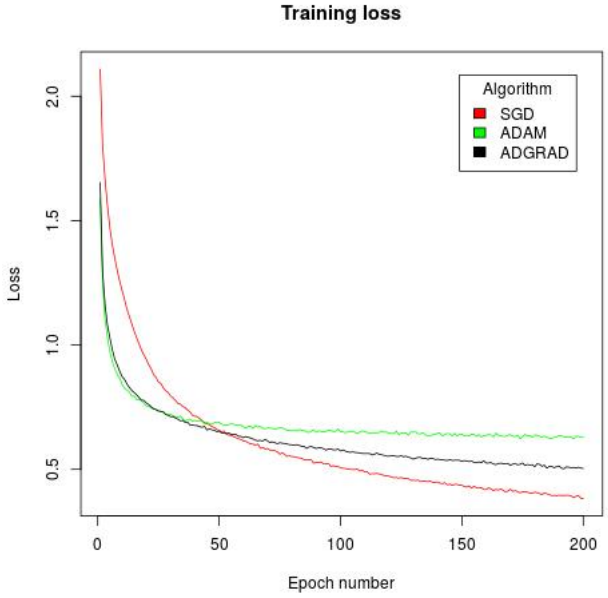


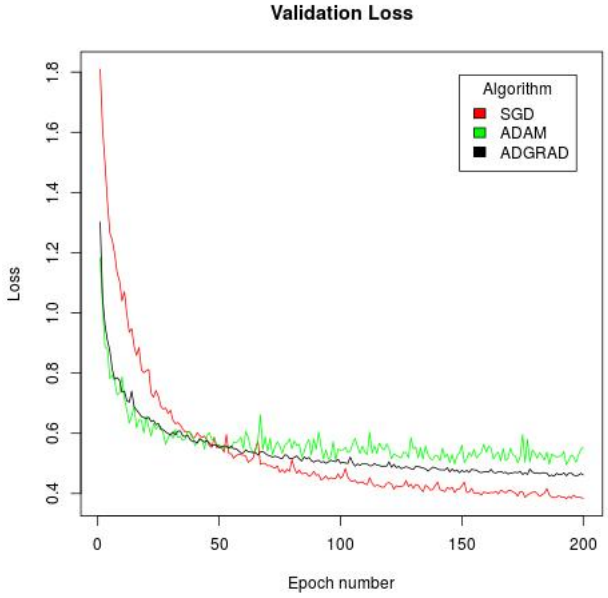
Figure 1: overview of the CIFAR 10 dataset (Coutresy of Alex Krizhevsky)

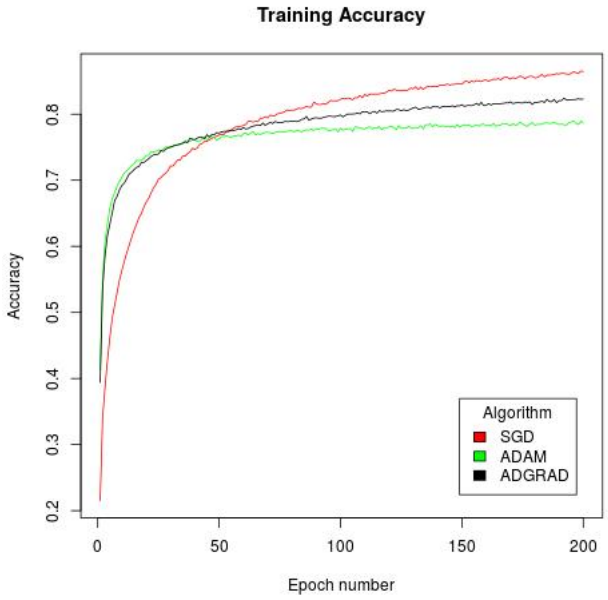
- The time for an epoch to finish.
- The loss on the training set.
- The accuracy of classification on the training set.
- The loss on the validation(Test) set.
- The accuracy of classification on validation set.

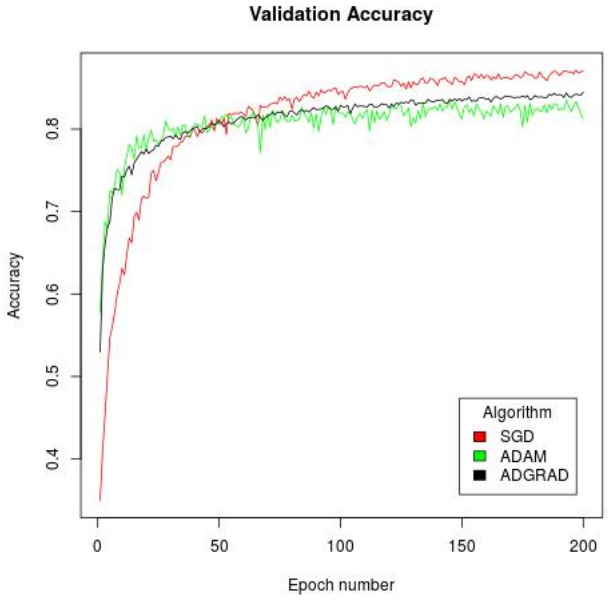
Below are the graphs showing the comparison of each of the above metrics between the 3 algorithms.




Training loss
Algorithm
SGD
ADAM
ADGRAD
Loss
0.5
1.0
1.5
2.0
0
50
100
150
200
Epoch number


Validation Loss
Algorithm
SGD
ADAM
ADGRAD
Loss
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
0
50
100
150
200
Epoch number





4 Conclusion

TODO Rewrite the section.
Conclude the paper with final findings.

5 Appendix A

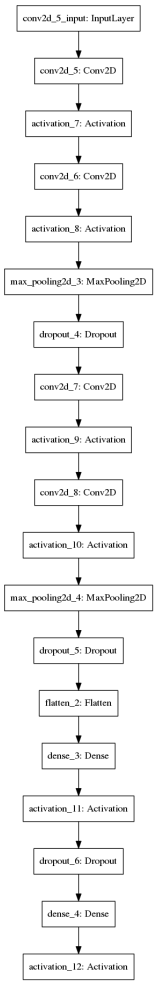


Figure 2: An overview of the convolution model used for CIFAR10 classification

References

[Arel et al., 2010] Arel, I., Rose, D. C., and Karnowski, T. P. (2010). Deep machine learning - a new frontier in artificial intelligence research [research frontier]. *IEEE Computational Intelligence Magazine*, 5(4):13–18.

[Diederik P. Kingma, 2015] Diederik P. Kingma, J. L. B. (2015). Adam: A method for stochastic optimization. *ICLR*.

[John Duchi, 2011] John Duchi, Elad Hazan, Y. S. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*.

[Krizhevsky, 2009] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

[Schmidhuber, 2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.