A comparison of iterative optimizers applied to the MNIST dataset

Suhrid Deshmukh, Ismail Degani

6.337 Final Project
May, 6th 2017

Abstract

Abstract for algorithms used and performance on the bench mark problem.

¡¡¡¡¡¡¡ HEAD

1 Introduction

Machine learning and optimization has found home in some of the most important machine learning(ML) applications. The fields of machine learning and mathematical programming are becoming increasingly intertwined. Optimization problems lie at the heart of most machine learning approaches. Many optimization algorithms that are applied in machine learning problems converge at different rates. Manu of the learning algorithms also have different memory footprint and different computational complexity. Studying how different optimization algorithms work when applied in ML applications is therefore an interesting field of research.

The work here explores different optimization schemes that are used in training weight matrices in a Neural net that uses logistic regression. The learning problem in neural nets is formulated in terms of the minimization of a loss function. We can combine the weights and the biases in a neural net in the loss function weight vector w. The loss function is then f(w) with w* as its minima. The learning problem in neural nets is basically searching for a weight vector w* such that the value of the loss function f(w) is minimum at w*. This means that if we evaluate the value of the gradient of f(w) at w*, the gradient is zero.

The loss function in general is a multi-dimensional non-linear function. In order to minimize the loss function, typically iterative optimization methods are used. The iterative algorithm calculates the loss at each iteration, the change of loss between two steps is called loss decrement. The training algorithm stops when the specified criteria is met. There are multiple training algorithms that are available to train a neural net. We consider three different algorithms to train a neural net; Gradient descent, Quasi Newton (BFGS),Adam-Optimizer.

The algorithms mentioned above are applied on a simple problem first to make sure that they are functioning properly. They are then used on a benchmark problem which is the classification of handwritten digits in a MNIST data set using a neural net that uses logistic regression. The next section describes each of the algorithms in detail along with the pseudocode and their performance characteristics.

2 Algorithms

2.1 Gradient Descent

2.1.1 Pseudocode

2.1.2 Performance characteristics

2.2 Adam-Optimizer

2.2.1 Pseudocode

2.2.2 Performance characteristics

2.3 BFGS

2.3.1 Pseudocode

2.3.2 Performance characteristics

2.4 Benchmark Problem

2.5 Numerical results and discussion

=======

3 Using LaTeXwith PDF Figures

This is a sample document for use with pdflatex, which is a program that is included with the Miktex distribution that directly produces PDF files from LaTeXsources. To run LaTeXon this file, you need the following files:

1. templatePDF.tex (this file)
2. figure.pdf (the figure file)
3. simpleConference.sty (style file)
4. refs.bib (bibiliography file)

To create a PDF file, execute the following commands:

1. pdflatex templatePDF
2. bibtex templatePDF
3. pdflatex templatePDF
4. pdflatex templatePDF

Yes (strangely) it is necessary to run pdflatex three times. The result will be a PDF file (plus several other files that LaTeXproduces). You will need a mechanism, of course, for executing commands on the command line. If you are using Windows, I recommend installing Cygwin and using its bash shell.

4 Adam Optimizer

The Adam Optimizer [1] is a member of a class of "adaptive" gradient descent optimizers which adjust their learning rate α at each iteration in order to converge more efficiently. This adaptive behavior significantly reduces the impact of a badly chosen initial learning rate α, which can be difficult to determine. Adam is specifically optimized for sparse datasets as are many similar algorithms in this class. This makes it an excellent candidate for MNIST data which is monochrome and therefore predominantly zero-value.

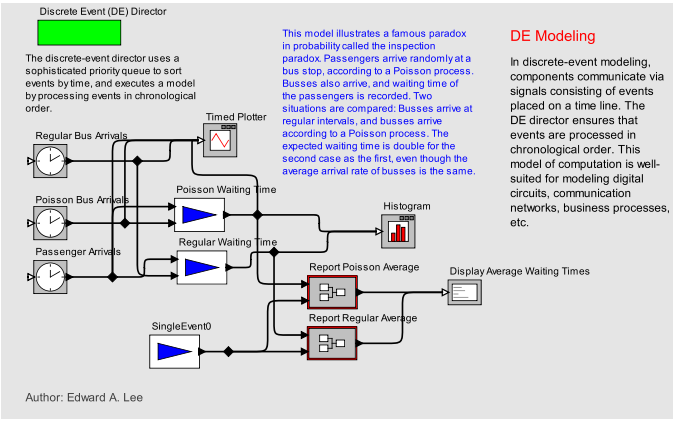


Figure 1. Figure caption. To get a figure to span two columns, use the environment figure* rather than figure.

The Adam algorithm stores a history of previous gradients v_t and squared gradients m_t, also known as first and second moments. In order to compute the current gradient, Adam applies exponentially decaying factors β_1 and β_2 to the first and second moments. It finally calculates bias-adjusted values of these moments $\hat{v}_t, \hat{m}_t$, and applies a smoothing term ϵ which avoids division by zero. Pseudocode is given below:

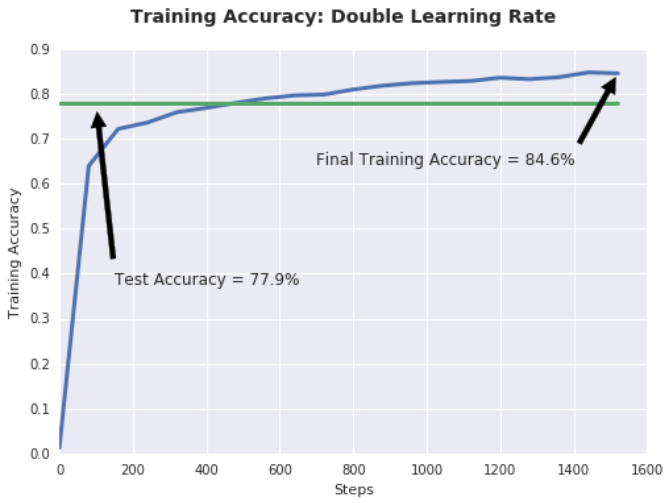
4.1 Algorithm 1: Adam optimization step

while $f(\theta_t) > \tau$
$t = t + 1$
$g_t = \nabla_\theta f_t(\theta_{t-1})$
$m_t = \beta \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
$v_t = \beta_2 \cdot v_{t-1} + (1 + \beta_2) \cdot g_t^2$
$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$
$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$
$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}t}+\epsilon}$
end while

5 Performance Characteristics

The following section compares gradient descent, adam, and BFGS optmizers on a linear least squares Ax = b optimization, and a logistic regression to solve the MNIST problem.

5.1 Gradient Descent



Suppose you wish to include a figure, like that in figure **??**. The simplest mechanism is to install Adobe Acrobat, which includes a "printer" called "Acrobat Distiller." Printing to this printer creates a PDF file, which can be included in a document as shown here. To include Ptolemy II models

[**?**], just print to the distiller from within Vergil and reference the PDF file in your LaTeXdocument.

There is a bit more work to do, however. The file that is produced by the distiller represents a complete page, not the individual figure. You can open it in using Acrobat (version 5.0 or later), and select Document → Crop Pages from the menu. In the resulting dialog, check "Remove White Margins." Save the modified PDF file in a file and then reference it in the LaTeXfile as shown in this example.

An alternative is to generate EPS (encapsulated postscript), but the process is much more complex and fragile. I recommend using pdflatex and Adobe Acrobat. ¿¿¿¿¿¿¿ origin/master