A comparison of iterative optimizers applied to the MNIST dataset

Suhrid Deshmukh, Ismail Degani

6.337 Final Project
May, 6th 2017

Abstract

This is a simple sample of a document created using LaTeX(specifically pdflatex) that includes a figure from the Vergil visual editor for Ptolemy II that was created by printing to the Acrobat Distiller to get a PDF file. It also illustrates a simple two-column conference paper style, and use of bibtex to handle bibligraphies.

1 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)

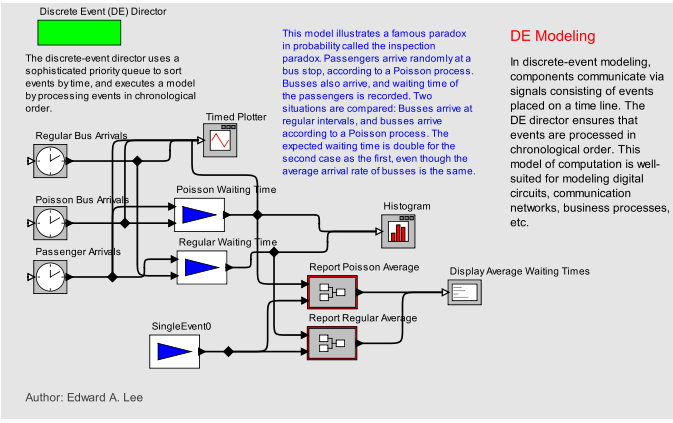


Figure 1. Figure caption. To get a figure to span two columns, use the environment figure* rather than figure.

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.

2 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.

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:

2.1 Algorithm 1: Adam optimization step

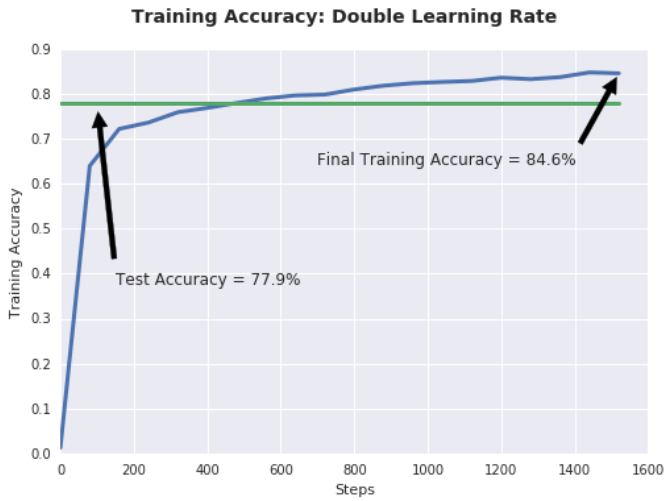
while $f(\theta_t) > \tau$

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

3 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.

3.1 Gradient Descent



Suppose you wish to include a figure, like that in figure 1. 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 [1], 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.

References

[1] C. Brooks, E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, and H. Zheng. Heterogeneous concurrent modeling and design in java. Technical Report Technical Memorandum UCB/ERL M04/27, University of California, July 29 2004.