Large batch sizes for Deep Learning

Mario Rene Surlemont
University Vienna

Elnaz Javadi Farahzadi
University Vienna

Alexander Stähle
University Vienna

***Abstract*—A critical decision when using SGD variants is choosing the size of a batch. In the past, it has been shown that a generalization gap occurs when using large batch sizes rather than small batch sizes. This can be compensated (conditionally) by a larger learning rate, but the minima found usually remain sharper than with smaller batch sizes. Within this project we evaluate whether a formulation that regulates the sharpness of minimizers is suitable to compensate for the problems of large batch sizes. We do this by assessing the results of an empirical study of a heterogeneous set of optimizers and loss functions in relation to different batch sizes.**

I. Introduction

English

Deep Learning has become a popular approach to solve (among others) classification problems. The non-convex optimization problem to be solved in Deep Learning applications is typically of the form

$$\min_{w\in\mathbb{R}^d} f(w) := \frac{1}{M}\sum_{i=1}^{M} f_i(w), 1 \tag{1}$$

where f_i is the loss function that calculates the loss for data point $i \in \{1, 2, ..., M\}$, M is the number of data points and w is the weight vector. For minimizing such functions one often uses Stochastic Gradient Descent (SGD) and its variants. f is minimized by iteratively taking steps of the form

$$x_{k+1} = x_k - \alpha_k\left(\frac{1}{|B_k|}\sum_{i=1}^{|B_k|}\nabla f_i(w_k)\right). \tag{2}$$

Here $B_k \subset \{1, 2, ..., M\}$ is a randomly selected batch of size $|B_k|$ and α_k is the learning rate in iteration step k. Usually one uses batches of size $|B_k| \in \{32, 64, ..., 512\}$, where $|B_k| \ll M$. (TODO: Maybe add what is the advantage of using SGD and also add references to papers where the advantage of these batches was shown) Since these rather small batches are a ”computational” challenge (TODO: possibly add references or justification), past research has looked at the impact of larger batch sizes on the learning process and final weights. (TODO: Possibly replace weights with result or something).

In general, it appears that using larger batches results in a larger generalization gap with one possible reasoning behind this being so-called ”sharp minima” [1]. This can be partially compensated by an adjusted learning rate [2], but the sharp minima remain. Therefore, efforts have been made to develop a problem formulation that not only minimizes the given loss function, but simultaneously adjusts the sharpness of the minima [3]. This formulation is referred to as Sharpness-Aware Minimization (SAM).

Within this project, we will now evaluate to what extent SAM is suitable to enable training with large batches.

We will train a very simple neural network for image recognition once using SGD with small to moderate batch sizes and apply SAM to large batches for comparison. Furthermore we will explore how choosing different variants of SGD (e.g. Adam) and a variation of SAM, which adapts to the local geometry of the loss function (ASAM), will affect the generalization ability of the model.

The resulting minima will additionally be examined for sharpness and runtime. We also evaluate how learning rate and batch size are related in the SGD scenario and check whether this relationship is also evident when SAM or ASAM is used.

In [1]

The aim of writing a paper is to infect the mind of your reader with the brilliance of your idea [4]. The hope is that after reading your paper, the audience will be convinced to try out your idea. In other words, it is the medium to transport the idea from your head to your reader's head. In the following section, we show a common structure of scientific papers and briefly outline some tips for writing good papers in Section III.

At that point, it is important that the reader is able to reproduce your work [5], [6], [7]. This is why it is also important that if the work has a computational component, the software associated with producing the results are also made available in a useful form. Several guidelines for making your user's experience with your software as painless as possible is given in Section IV.

This brief guide is by no means sufficient, on its own, to make its reader an accomplished writer. The reader is urged to use the references to further improve his or her writing skills.

II. The Structure of a Paper

Scientific papers usually begin with the description of the problem, justifying why the problem is interesting. Most importantly, it argues that the problem is still unsolved, or that the current solutions are unsatisfactory. This leads to the main gist of the paper, which is “the idea”. The authors then show evidence, using derivations or experiments, that the idea works. Since science does not occur in a vacuum, a proper comparison to the current state of the art is often part of the results. Following these ideas, papers usually have the following structure:

Abstract
: Short description of the whole paper, to help the reader decide whether to read it.

Introduction
: Describe your problem and state your contributions.

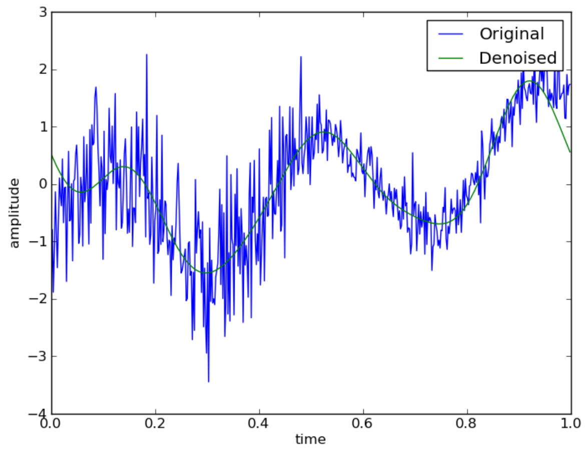

Fig. 1. Signal compression and denoising using the Fourier basis.

Models and Methods
Describe your idea and how it was implemented to solve the problem. Survey the related work, giving credit where credit is due.

Results
Show evidence to support your claims made in the introduction.

Discussion
Discuss the strengths and weaknesses of your approach, based on the results. Point out the implications of your novel idea on the application concerned.

Summary
Summarize your contributions in light of the new results.

III. Tips for Good Writing

The ideas for good writing have come from [8], [4], [9].

A. Getting Help

One should try to get a draft read by as many friendly people as possible. And remember to treat your test readers with respect. If they are unable to understand something in your paper, then it is highly likely that your reviewers will not understand it either. Therefore, do not be defensive about the criticisms you get, but use it as an opportunity to improve the paper. Before your submit your friends to the pain of reading your draft, please *use a spell checker*.

B. Abstract

The abstract should really be written last, along with the title of the paper. The four points that should be covered [4]:

1) State the problem.
2) Say why it is an interesting problem.
3) Say what your solution achieves.
4) Say what follows from your solution.

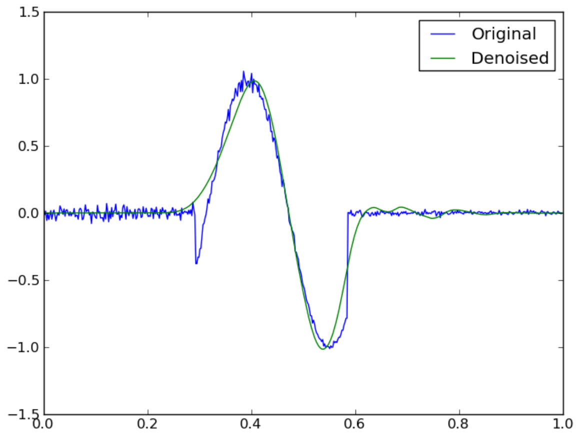

Fig. 2. Signal compression and denoising using the Daubechies wavelet basis.

C. Figures and Tables

Use examples and illustrations to clarify ideas and results. For example, by comparing Figure 1 and Figure 2, we can see the two different situations where Fourier and wavelet basis perform well.

D. Models and Methods

The models and methods section should describe what was done to answer the research question, describe how it was done, justify the experimental design, and explain how the results were analyzed.

The model refers to the underlying mathematical model or structure which you use to describe your problem, or that your solution is based on. The methods on the other hand, are the algorithms used to solve the problem. In some cases, the suggested method directly solves the problem, without having it stated in terms of an underlying model. Generally though it is a better practice to have the model figured out and stated clearly, rather than presenting a method without specifying the model. In this case, the method can be more easily evaluated in the task of fitting the given data to the underlying model.

The methods part of this section, is not a step-by-step, directive, protocol as you might see in your lab manual, but detailed enough such that an interested reader can reproduce your work [9], [6].

The methods section of a research paper provides the information by which a study's validity is judged. Therefore, it requires a clear and precise description of how an experiment was done, and the rationale for why specific experimental procedures were chosen. It is usually helpful to structure the methods section by [10]:

1) Layout the model you used to describe the problem or the solution.
2) Describing the algorithms used in the study, briefly including details such as hyperparameter values (e.g.

thresholds), and preprocessing steps (e.g. normalizing the data to have mean value of zero).

3) Explaining how the materials were prepared, for example the images used and their resolution.
4) Describing the research protocol, for example which examples were used for estimating the parameters (training) and which were used for computing performance.
5) Explaining how measurements were made and what calculations were performed. Do not reproduce the full source code in the paper, but explain the key steps.

E. Results

Organize the results section based on the sequence of table and figures you include. Prepare the tables and figures as soon as all the data are analyzed and arrange them in the sequence that best presents your findings in a logical way. A good strategy is to note, on a draft of each table or figure, the one or two key results you want to address in the text portion of the results. The information from the figures is summarized in Table I.

When reporting computational or measurement results, always report the mean (average value) along with a measure of variability (standard deviation(s) or standard error of the mean).

IV. Tips for Good Software

There is a lot of literature (for example [11] and [12]) on how to write software. It is not the intention of this section to replace software engineering courses. However, in the interests of reproducible research [5], there are a few guidelines to make your reader happy:

- Have a `README` file that (at least) describes what your software does, and which commands to run to obtain results. Also mention anything special that needs to be set up, such as toolboxes[1].
- A list of authors and contributors can be included in a file called `AUTHORS`, acknowledging any help that you may have obtained. For small projects, this information is often also included in the `README`.
- Use meaningful filenames, and not `temp1.py`, `temp2.py`.
- Document your code. Each file should at least have a short description about its reason for existence. Non obvious steps in the code should be commented. Functions arguments and return values should be described.
- Describe how the results presented in your paper can be reproduced.

A. LaTeX Primer

LaTeX is one of the most commonly used document preparation systems for scientific journals and conferences. It is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. The source of this file can be used as a starting point for how to use the different commands in LaTeX. We are using an IEEE style for this course.

1) Installation: There are various different packages available for processing LaTeX documents. See our webpage for more links for getting started.

2) Compiling LaTeX: Your directory should contain at least 4 files, in addition to image files. Images should ideally be `.pdf` format (or `.png`).

3) Equations: There are three types of equations available: inline equations, for example $y = mx + c$, which appear in the text, unnumbered equations

$$y = mx + c,$$

which are presented on a line on its own, and numbered equations

$$y = mx + c \tag{3}$$

which you can refer to at a later point (Equation (3)).

4) Tables and Figures: Tables and figures are "floating" objects, which means that the text can flow around it. Note that `figure*` and `table*` cause the corresponding figure or table to span both columns.

V. Summary

The aim of a scientific paper is to convey the idea or discovery of the researcher to the minds of the readers. The associated software package provides the relevant details, which are often only briefly explained in the paper, such that the research can be reproduced. To write good papers, identify your key idea, make your contributions explicit, and use examples and illustrations to describe the problems and solutions.

Acknowledgements

The author thanks Christian Sigg for his careful reading and helpful suggestions.

[1]For those who are particularly interested, other common structures can be found at http://en.wikipedia.org/wiki/README and http://www.gnu.org/software/womb/gnits/.

Basis	Support	Suitable signals	Unsuitable signals
Fourier	global	sine like	localized
wavelet	local	localized	sine like

TABLE I
CHARACTERISTICS OF FOURIER AND WAVELET BASIS.

REFERENCES

[1] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” 2017.

[2] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch sgd: Training imagenet in 1 hour,” 2018.

[3] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” 2021.

[4] S. P. Jones, “How to write a great research paper,” 2008, microsoft Research Cambridge.

[5] M. Schwab, M. Karrenbach, and J. Claerbout, “Making scientific computations reproducible,” *Computing in Science and Engg.*, vol. 2, no. 6, pp. 61–67, 2000.

[6] J. B. Buckheit and D. L. Donoho, “Wavelab and reproducible research,” Stanford University, Tech. Rep., 2009.

[7] R. Gentleman, “Reproducible research: A bioinformatics case study,” *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, 2005. [Online]. Available: http://www.bepress.com/sagmb/vol4/iss1/art2

[8] Editorial, “Scientific writing 101,” *Nature Structural & Molecular Biology*, vol. 17, p. 139, 2010.

[9] G. Anderson, “How to write a paper in scientific journal style and format,” 2004, http://abacus.bates.edu/ ganderso/biology/resources/writing/HTWtoc.html.

[10] R. H. Kallet, “How to write the methods section of a research paper,” *Respiratory Care*, vol. 49, no. 10, pp. 1229–1232, 2004.

[11] A. Hunt and D. Thomas, *The Pragmatic Programmer*. Addison Wesley, 1999.

[12] J. Spolsky, *Joel on Software: And on Diverse & Occasionally Related Matters That Will Prove of Interest etc..: And on Diverse and Occasionally Related Matters ... or Ill-Luck, Work with Them in Some Capacity*. APRESS, 2004.