Reading 1

(1) Title, author and where it was published

- Title: Deep Learning; Chapter 4 Numeric Computation
- Author: Ian Goodfellow and Yoshua Bengio and Aaron Courville
- Published: MIT Press

(2) Summary of the content

This chapter describes some of the mathematical preliminaries that one needs to use to develop machine learning algorithms. Firstly, it presents several common problems and techniques employed in solving numeric problems on a computer. In particular, this paper walks the reader through problems like overflow, underflow, and poor conditioning. It then discusses the mathematical formulas behind gradient-based optimization, as well as how Jacobian and Hessian matrices help avoiding the problems mentioned before. Lastly, the author mentions constrained optimization where we want to find the maximal or minimal value in some smaller set rather than all possible solutions. Here, the Karush-Kuhn-Tucker (KKT) approach and conditions are introduced with a basic walk-through for its intuition behind. The authors then conclude this chapter by an example of how to find the optimal solution for a linear least square function.

(3) Strengths of the paper

This paper illustrates how underflow and overflow could be a problem in machine learning with an example from softmax function, increasing the comprehensibility of this concept. Softmax function is a powerful tool in machine learning, yet it suffers from the overflow and underflow problems on computers. Underflow might cause a division-by-zero problem, and overflow might cause a not-a-number problem, and both of them are fatal to many machine learning algorithms. One solution is to modify the input data by subtracting the maximum value in that dimension so that the exponential function would not lead to overflow or underflow, and analytically, this modification changes nothing.

Moreover, the authors provide the readers with a quick review of calculus and practical implications over the meaning of first and second derivatives, which are accompanied by several figures and visualizations, making the mathematical formulas more understandable. For a typical two dimensional optimization problem, we could visualize clearly the problem of saddle point and inconclusiveness of a second derivative test when the second derivative is 0. With this, we could imagine similar things happening in higher dimensions, like how the algorithms manage to find local minimums using directional derivatives, and thus grab the gist of gradient-based optimization.

Last but not least, in the example about linear least squares, we see how we could solve the optimization problem with both the typical gradient descent algorithm and Newton's method. The authors also briefly touched on the constrained optimization with a generalized Lagragian function. The full deduction and analysis of choosing the appropriate lambda explains how the machine actually learns from the formula.

(4) Major critiques

The authors mention succinctly about convex optimization algorithms and its limited power on solving deep learning problems as they could only be applied to convex problems where the Hessian matrix is positive semidefinite. Therefore, it is clear that more research about optimization techniques for more complex problems should be done in the future.

Reading 2

(1) Title, author and where it was published

- Title: AUTOCLIP: ADAPTIVE GRADIENT CLIPPING FOR SOURCE SEPARATION NETWORKS
- Author: Prem Seetharaman, Gordon Wichern, Bryan Pardo, and Jonathan Le Roux
- Published: Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

(2) Summary of the content

This paper discusses the AutoClip algorithm that the professors have developed to improve the current gradient clipping algorithm. Compared with traditional clipping methods where manual tuning for clipping threshold is required, AutoClip is able to observe and learn from the history of the gradient norms and automatically and adaptively develop its clipping threshold. The authors demonstrate the effectiveness of AutoClip first by emphasizing the pain points that the traditional methods confront, and then introduce the mathematical insights behind gradient clipping and the core algorithm behind AutoClip. Then, the paper presents an experiment targeting at testing AutoClip's transferability across different loss functions as well as comparing the performance of using AutoClip and not using any clipping method, enhancing the claim that AutoClip is an effective optimization technique for many machine learning tasks.

(3) Strengths of the paper

The overall structure of the paper is well organized and easy to follow. The paper first explains the concept of gradient clipping and why it is important in source separation tasks: that it resolves the issue of exploding gradients by scaling gradients if the norm is too high. Then, it pinpoints that there is no "no-size-fits-all" solution for picking an appropriate clipping value hyperparameter because it is very sensitive to the loss function, the data scale, and the network architecture. With this in mind, the authors introduce their solution – AutoClip – that can adaptively adjust this hyperparameter based on a percentile cutoff parameter.

On the other hand, AutoClip also has generalizability as it generates consistently better performance across different loss functions compared to previous hand-tuned clipping threshold methods. Table 1 showing performance in terms of SI-SDR demonstrates clearly that having no gradient clipping ($p = 100$) results in the worst performance across all loss functions. Meanwhile, the bottom right figure in Figure 1 presents that AutoClip can generate better generalization performance since it signals a stronger correlation between gradient norm and the local smoothness. With the figures and tables accompanying, the comprehensibility and trustworthiness of the argument becomes stronger.

Last but not least, the video is effective in helping the reader to gain a deeper and clearer understanding of these concepts. It provides visualizations of the gradient clipping and how this method prevents the model from generating exploding gradients.

(4) Major critiques

This paper mentions several other optimization techniques like AdaGrad, Adam, and SGD, yet it only claims that AutoClip is compatible with these methods but has no data backup for this claim. Moreover, as mentioned in the last part of this paper, since AutoClip observes the history of gradients, data storage and computing power might be another issue when this method is to be applied to some large scale project. Further research on memory reduction might need to be conducted.

Reading 3

(1) Title, author and where it was published

- Title: Why Momentum Really Works
- Author: Goh, Gabriel
- Published: Distill, 2017

(2) Summary of the content

This article explains the theory and practical effectiveness of momentum through a combination of mathematics, interactive graphs, and real applications. It starts by stating the generally believed but somehow misleading understanding of momentum, and then introduces the reader to the role of eigenvalues and condition number and how they should be understood in the realm of machine learning. Then, the author walks the reader through a polynomial regression to illustrate the relationship between step-size, momentum, and the condition number, and reinforces the statement that momentum accelerates convergence through an example of colorization problem. Lastly, the authors discuss some limitations on these techniques, and suggest the directions for further research.

(3) Strengths of the paper

The biggest strength of this article is that it provides a balance between theories and applications. Momentum as a technique to gradient descent optimization algorithms is usually understood as adding “acceleration” on the searching, yet in theory it actually adds inertia in the direction of the searching so that the algorithm is robust to the noisy oscillations in the gradients. This article explains this power of momentum with a rich number of graphs and examples. With the example of deriving gradient descents on a convex quadratic function, the authors illustrate that the errors in different dimensions computed during the learning process actually have different implications, and in the example of fitting a polynomial regression, the author provides the interactive graphs so that the reader have a hand-on experience of tuning the model by hand, gaining new insights on how we decompose an n-dimensional optimization problem into n 1-dimensional optimization problem, which is indeed easier to solve. Furthermore, the graph on convergence rate and step-size clearly illustrates the relationship and what could happen with different values. We could see that there is a very specific range for step-size and convergence rate that the momentum could converge. From here, we could derive the optimal solution for momentum in this setting, which is choosing a value closer to 1 and then solving for step-size accordingly.

The example on colorization not only demonstrates how momentum works better under this context, but also briefly exposes the reader to the connection between linear algebra and graph theory. The analogue of spread of ink drops on a smooth space vividly depicts that adding momentum could be much more time- and power-saving than traditional methods.

(4) Major critiques

It seems like, despite the effectiveness of momentum applied in some context, momentum might suffer from missing the local minimum issue since it does not have the notion of direction. It is also not adaptive, and there is no “one-size-fits-all” answer for this parameter. The end users might need to do some trial-and-error as well as computation before they could find the optimal solution.