UNDERSTANDING OPTIMIZATION *in* REINFORCEMENT LEARNING

An Empirical Study of
Algorithms and their Hyperparameters

Jan Ole von Hartz

May 2019

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science

to the

Machine Learning Lab
Department of Computer Science
Technical Faculty
University of Freiburg

Work Period

28. 02. 2019 – 28. 05. 2019

Examiner

Prof. Dr. Frank Hutter

Supervisor

Raghu Rajan

Contact

hartzj@cs.uni-freiburg.de

This thesis was typest using the incredible template created and released into the public domain by Eivind Uggedal, found at https://github.com/uggedal/thesis.

Hereafter follow the original remarks to his thesis.

This thesis was typeset using the LaTeX typesetting system originally developed by Leslie Lamport, based on TeX created by Donald Knuth.

The body text is set 12/14.5pt on a 26pc measure with Minion Pro designed by Robert Slimbach. This neohumanistic font was first issued by Adobe Systems in 1989 and have since been revised. Other fonts include Sans and Typewriter from Donald Knuth's Computer Modern family.

Typographical decisions were based on the recommendations given in *The Elements of Typographic Style* by Bringhurst (2004).

The use of sidenotes instead of footnotes and figures spanning both the textblock and fore-edge margin was inspired by *Beautiful Evidence* by Tufte (2006).

The guidelines found in *The Visual Display of Quantitative Information* by Tufte (2001) were followed when creating diagrams and tables. Colors used in diagrams and figures were inspired by the *Summer Fields* color scheme found at http://www.colourlovers.com/palette/399372

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PART I

BACKGROUND

KEY CONCEPTS AND NOTATION

1

1.1 NOTATION

Whenever possible we distinguish between

- scalars: x
- vectors: $\vec{x}$
- matrices: X
- sets: $\mathcal{X}$
- (time) series $(\chi_n)_{n=0,1,\dots}$

For a vector $\vec{x}$ the i-th element is denoted as x_i. For a matrix X, $\vec{x}_i$ denotes the i-th column and x_{ij} the j-th value of this column. The value of a time series $(\chi_n)_{n=0,1,\dots}$ at point t is denoted as χ_t.

1.2 KEY CONCEPTS

To fully understand the content of this thesis, the reader should be familiar with some key concepts of *deep learning* - machine learning using *deep neural networks*. We will briefly reiterate the most important concepts to accomplish a common ground in notation. Unfamiliar readers are advised to revise these concepts, e.g. with Goodfellow et al. (2016). Proficient readers can safely skip this chapter. Necessary concepts from *reinforcement learning* and *deep reinforcement learning* will be introduced in the next chapter. Debutants can deepen their understanding, e.g. in Sutton and Barto (2018).

Supervised Learning Given some function class $\mathcal{G} : \mathbf{R}^d \to \mathbf{R}^o$ and a matrix of training data $X \in \mathbf{R}^{d \times n}$ with targets $Y \in \mathbf{R}^{\mathbf{o} \times \mathbf{n}}$, that are sampled from some generating data distribution X_{Gen}, Y_{Gen}, an algorithm tries to find the function $g \in \mathcal{G}$ that best describes the relation of data points $\vec{x}_i \in X_{Gen}$ with the matching targets $\vec{y}_i \in Y_{Gen}$ by using the training data and targets as an approximation of the general data distribution and minimizing some **loss function** $L : \mathbf{R}^{o \times 2} \to \mathbf{R}$ on the set of pairs of predictions $g(\vec{x}_i)$ on the training data, and training targets $\vec{y}_i$. One commonly used loss functions is the **mean squared error** $L(\vec{x}_j, \vec{y}_j) = \frac{1}{n} \sum_{i=0}^{n} (x_{ij} - y_{ij})^2$, where $n = |a| = |b|$. The learning is usually done by

defining $\mathcal{G}$ as a set of functions $g_{\vec{\theta}}$ with parameters $\vec{\theta}$ and numerically optimizing these parameters.

(Mini-) Batch Since the computation of the loss on the whole training set at once it not only costly but also not necessarily effective (Keskar et al., 2016), while the computation on a single data point leads to very noisy estimates of the gradients in numerical optimization, a common approach is to divide the training data into or to sample *(mini-) batches* from the training data and compute the numerical updates on these instead of on the full data set.

Overfitting is what happens when a machine learning model fits the training distributions more closely than justified by the underlying generating data distribution, leading to a worse generalization performance.

Gradient Descent is a numerical optimization method, suited to find local minima of some differentiable function $f : \mathbf{R}^n \to \mathbf{R}$ and global minima if f is *convex*[1]. Starting at some point $\vec{x}_0$ and with some *learning rate* α, gradient descent iteratively computes updates $\vec{x}_t = \vec{x}_{t-1} - \alpha \cdot \nabla f(\vec{x}_{t-1})$ until some stopping criterion (e.g. convergence) is reached.

1. Convexity is usually not given, but local optima give decent approximations and might overfit less.

Stochastic Gradient Descent is a stochastic approximation of gradient descent, commonly applied in machine learning. The gradients of the loss function for the parameter updates are computed on a randomly sampled minibatch, often leading to faster convergence.

(Deep) Neural Network are non-linear function approximators commonly used in recent machine learning research. They are constructed by stacking layers of nodes (neurons) that each 1. compute some linear combination of the values of the nodes of the previous layer and 2. apply some non-linear activation function. The parameters of the network (the weights and biases of the linear functions) are denoted by $\vec{\theta}$. A prediction for some data vector is made by feeding the latter into the first layer (the input layer) of the network and computing the activation of the last layer (the output layer). This is called the *forward pass*. The value of the output layer of the network f with parameters $\vec{\theta}$ for some input $\vec{x}$ is denoted as $f(\vec{x}, \vec{\theta})$. Supervised training of the network is commonly achieved by calculating the average derivative $\frac{1}{n} \sum_{i=0}^{n} \nabla_{\vec{\theta}} L(\vec{y}_i, f(\vec{x}_i, \vec{\theta}))$ of some loss function L, with respect to the network parameters $\vec{\theta}$ on some (mini-) batch X with size n and targets Y, (called *backward pass*) and applying updates to $\vec{\theta}$ via *(stochastic) gradient descent*.

Observability In artificial intelligence research, an environment is called *fully observable*, if an agent that interacts with it, has full access to all relevant information about its inner state, *partly observable* else.

PART II

PERFORMANCE ESTIMATION

PART III

OPTIMIZER ANALYSIS

PART IV

SUMMARY

BIBLIOGRAPHY

Bringhurst, Robert. October 2004. *The elements of typographic style*. Hartley & Marks Publishers, Point Roberts, WA, USA, 3rd edn. ISBN 0-881-79205-5. Cited on p. c.

Goodfellow, Ian; Bengio, Yoshua; and Courville, Aaron. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org. Cited on p. 3.

Keskar, Nitish Shirish; Mudigere, Dheevatsa; Nocedal, Jorge; Smelyanskiy, Mikhail; and Tang, Ping Tak Peter. 2016. *On large-batch training for deep learning: Generalization gap and sharp minima*. In arXiv preprint arXiv:1609.04836. Cited on p. 4.

Sutton, Richard S and Barto, Andrew G. 2018. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, USA, 2nd edn. ISBN 978-0-26203-924-6. Cited on p. 3.

Tufte, Edward R. may 2001. *The Visual Display of Quantitative Information*. Graphics Press LLC, Cheshire, CT, USA, 2nd edn. ISBN 0-961-39214-2. Cited on p. c.

———. jul 2006. *Beautiful Evidence*. Graphics Press LLC, Cheshire, CT, USA. ISBN 0-961-39217-7. Cited on p. c.

INDEX

APPENDICES

none