Institute of Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Investigating the Influence of Learning Rates on the Learning Speed of Neural Networks

Robin Sasse

Course of Study:	Informatik B.Sc.
Examiner:	Prof. Dr.-Ing. Andrés Bruhn
Supervisor:	M.Sc. Jenny Schmalfuß
Commenced:	April 14, 2021
Completed:	October 14, 2021
CR-Classification:	G.1.6

Abstract

. . . ... Short summary of the thesis ...

Contents

List of Figures

List of Tables

List of Listings

List of Algorithms

1. Introduction

- mention ResNets (but not in detail)

- mention and explain CIFAR10

Structure of this work

This work is structured as follows:

Chapter 2 – Theoretical Foundations: Here the theoretical fundamentals of this work are laid out. In *2.2 – Gradient Decent* the basics of the (stochastic) gradient decent algorithm are explained, followed by the introduction of momentum in *2.3 – Momentum-Based Decent Methods*. We will further discuss the settings of hyper-parameters in *2.4 – Hyper-Parameter Settings of Neural Networks*. Lastly, *2.6 – AdaSecant* introduces an optimizer that does not require the engineer to set the learning rate.

Chapter ?? – ?? This chapter discusses the main part of this work, which are the practical experiments conducted. *4.1 – Set-Up of Experiments* discusses the set-up of our experiments and the reason behind this choice. In *4.2 – Results* we evaluate the finding of these experiments.

Chapter 5 – Conclusion and Outlook This chapter concludes this thesis and gives an outlook on future work that can be done to complete the findings of this work.

2. Theoretical Foundations

In this chapter we lay out the theoretical foundations of this work. The focus of this chapter is to inform the reader about the preceding research that was used as a foundation for any further development of algorithms and experiments. Furthermore, we use this chapter to introduce the most important notations used in the upcoming parts of this work.

2.1. ResNet Architectures

Since the main focus of this work is aimed at setting the learning rate of a neural net, this is just a brief introduction to the type of net used in this work. A residual (convolutional) neural network (ResNet) is a type of neural net which is optimized for very deep learning. It is significantly better suited for having a high number of layers than plain (convolutional) neural networks [1]. The reason for this lies in the special architecture of the ResNet and the difficulty of learning the identity function.

While a neural net's performance on new data might decrease with more layers due to overfitting, this should not be the case for its performance on the training data. More specifically, if Net A has more layers than Net B, Net A should yield at least the same accuracy on the training data as Net B. The reason being that if net B has x layers and net A has $z = x + y$ layers, Net A could theoretically learn the first x layers just as they were learned by Net B. For the following y layers Net A could simply learn the identity function. Thus, net A would produce the exact same results as Net B. But in practice this could not be observed [2].

The problematic assumption here is that the identity function can be learned easily. In reality this is a rather complicated function to learn for a neural net. ResNets therefore use skip layers, as indicated in Fig. 2.1. It is easy to learn a function that maps all inputs to zero, especially when exploiting the activation function. Therefore, the ResNet can easily learn the identity mapping between two non-subsequent layers by finding a zero mapping and adding it to the identity function supplied by the skip layer.

Formally speaking, a building block of a ResNet can be defined as

$$\hat{y}_j := \mathcal{H}_j \left(x_j, \{\theta_i\}_{i=k_j,\dots l_j}\right) \tag{2.1}$$

$$:= \mathcal{F}_j \left(x_j, \{\theta_i\}_{i=k_j,\dots l_j}\right) + x_j, \tag{2.2}$$

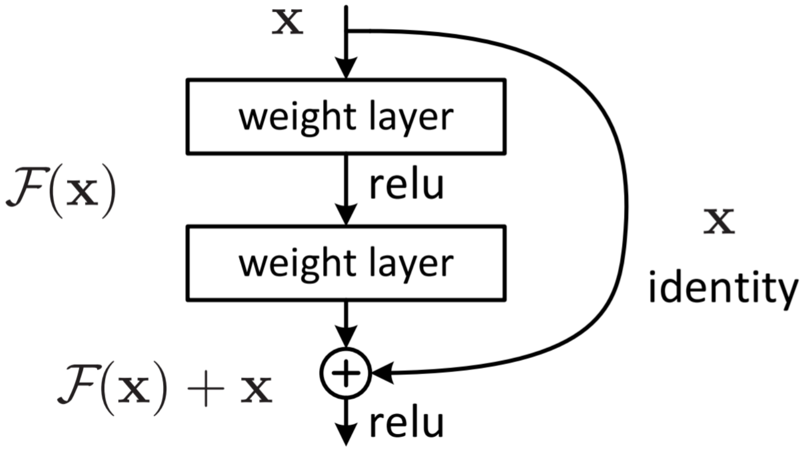
x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.1.: The skip layer as a building block of a ResNet (directly taken from [1]).

where $x_j = \hat{y}_{j-1}$, $j \in \{1, ..., n\}$ is the input of the building block's first layer. $\hat{y}_j$ is the output of the building block. θ_i is the i-th parameter of the network, with k and l denoting the first and last index of the parameters found within that building block. The ResNet consists of n such building blocks $\mathcal{H}_j(x_j, \{\theta_i\}_{i=k_j,...,l_j})$.

We further define the final output of the ResNet as[1]

$$\hat{y} := \mathcal{F}_{n+1}\left(\hat{y}_n\right) \tag{2.3}$$

$$:= \mathcal{F}_{n+1}\left(\mathcal{H}_n\left(\mathcal{H}_{n-1}\left(...\left(\mathcal{H}_1\left(x_1\right)\right)\right)\right)\right), \tag{2.4}$$

with

$$x_1 := \mathcal{F}_0\left(x\right), \tag{2.5}$$

x being the input of the net and $\mathcal{F}_{n+1}$, $\mathcal{F}_0$ being the first and last building blocks of the net, which are non-residual. A complete visual representation of a 34-layer ResNet is shown in Fig.2.2

The respective loss of the ResNet's prediction is, just as for any other network, defined as

$$l := f(y, \hat{y}), \tag{2.6}$$

with y being the actual target (i.e. the correct label or value) associated with x and f being the loss function (e.g. cross-entropy loss, mean squared error).

[1] The parameters as an argument of the function are implied and therefore left out for the purpose of simplicity.

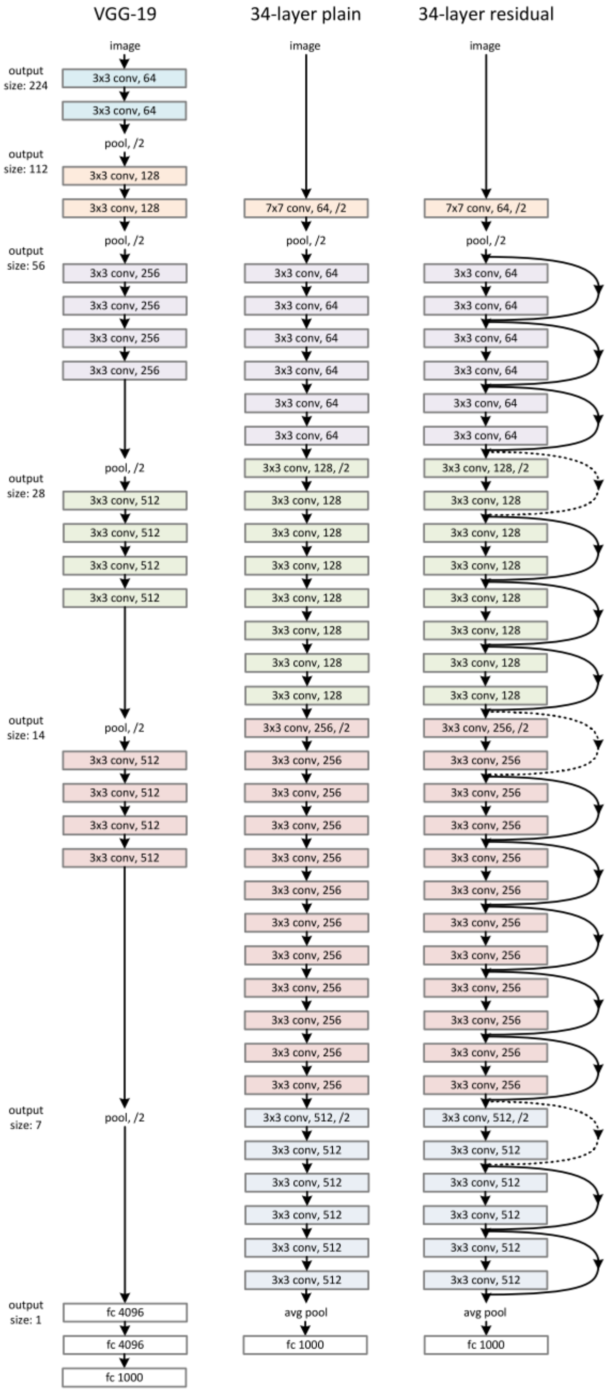

Figure 2.2.: Comparison of a 34-layer ResNet with a 34-layer plain net and the VGG-19 architecture (directly taken from [1]).

2.2. Gradient Decent

- Explain GD, SGD (incl. difference)

2.3. Momentum-Based Decent Methods

- introduce momentum as defined by Nesterov (look up original proposal)

- explain Adam in detail

- use created visuals here

2.4. Hyper-Parameter Settings of Neural Networks

- epochs

- batch size

- learning rate

2.5. Super-Convergence

2.6. AdaSecant

3. Implementations and Extensions of the Theoretical Foundations

3.1. Combining Exponential Decay and Cyclical Learning Rates

3.2. Combining Momentum and Cyclical Learning Rates

3.3. Implementing AdaSecant

4. Experiments

4.1. Set-Up of Experiments

Here we describe what experiments we conduct and why we chose those experiments in particular.

- choice of datasets

- choice of nets

- complexity and reduction (epoch length, best batch size, best lr, ...)

- include results for epoch length, batch size

4.2. Results

- specific results regarding learning rates (other results should be discussed in set-up)

4.2.1. SGD

4.2.2. Adam

4.2.3. AdaSecant

5. Conclusion and Outlook

Hier bitte einen kurzen Durchgang durch die Arbeit.

Outlook

...und anschließend einen Ausblick

A. Appendix

Write some text here and A.1

Figure A.1.: Beispiel-Choreographie I

Bibliography

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. (Cited on pages 6, 9, 10 and 11)

[2] K. He and J. Sun, “Convolutional neural networks at constrained time cost,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. (Cited on page 9)

All links were last followed on.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature