Local Convergence of Adaptive Gradient Descent Optimizers

Sebastian Bock and Martin Georg Weiß

***Abstract*—Adaptive Moment Estimation (ADAM) is a very popular training algorithm for deep neural networks and belongs to the family of adaptive gradient descent optimizers. However to the best of the authors knowledge no complete convergence analysis exists for ADAM. The contribution of this paper is a method for the local convergence analysis in batch mode for a deterministic fixed training set, which gives necessary conditions for the hyperparameters of the ADAM algorithm. Due to the local nature of the arguments the objective function can be non-convex but must be at least twice continuously differentiable. Then we apply this procedure to other adaptive gradient descent algorithms and show for most of them local convergence with hyperparameter bounds.**

***Index Terms*—ADAM Optimizer, Convergence, momentum method, dynamical system, fixed point**

I. Introduction

MANY problems in machine learning lead to a minimization problem in the weights of a neural network: Consider e.g. training data $(x_1,y_1),\dots,(x_N,y_N)$ consisting of inputs x_i and outputs y_i, and the task to determine a neural network that has learned the relationship between inputs and outputs. This corresponds to a function $y=F(w,x)$, parametrized by the weights w, which minimizes the average loss function

$$f(w)=\frac{1}{N}\sum_{i=1}^{N} l(x_i,y_i,w) =: \frac{1}{N}\sum_{i=1}^{N} f_i(w)$$

over the training data. Typically the loss is built using some norm for regression problems, e.g. $l(x,y,w)=\frac{1}{2}\|y-F(w,x)\|_2^2$, or using cross entropy for classification. Optimization algorithms construct a sequence $\{w_t\}_{t\in\mathbb{N}_0}$ of weights starting from an initial value w_0, which under appropriate assumptions converges to some local minimum $w_\star$ for general non-convex f. The most simple optimization algorithm for differentiable f is gradient descent with the update $w_{t+1}=w_t-\alpha\nabla f(w_t)$ and a learning rate $\alpha>0$. For convex f conditions on the Lipschitz constant L of ∇f guarantee convergence and give estimates for the rate of convergence, see [20]. However L is hard to get in practice, and choosing α too big leads to oscillatory behaviour. Besides it is well known from optimization that the gradient is not the only descent direction for f, neither is it optimal for finite step lengths, see [21], but computation of the Hessian is usually prohibitive. This has led to the development of a family of algorithms which compute moments of first order, that is approximate descent directions based on previous iterates of the gradient like the initial momentum method [20], as well as second order moments to control the componentwise scaling and / or to adapt the learning rate in AdaGrad [11] and ADAM [17]. For the ADAM variant studied in this paper see Algorithm 1. More algorithms exist with variants like batch mode vs. online or incremental mode – using $\nabla f(w_t)$ in iteration t vs. $\nabla f_k(w_t)$ where k iterates in a cyclic fashion over $1,\dots,N$, or deterministic vs. stochastic choice of the index k for $\nabla f_k(w_t)$, stochastic assumptions for the observation of $\nabla f(w_t)$ or $\nabla f_k(w_t)$, and so on.

However for most of these algorithms only partial convergence results are known. The original proof of [17] is wrong as has been noted by several authors, see [4], [25]. Modifying the algorithm to AMSGrad, [23] establishes bounds on $\|\nabla f(w_t)\|$, similar to the results in [9] for a class of algorithms called Incremental Generalized ADAM. Though none of the results shows convergence of the sequence $\{w_t\}_{t\in\mathbb{N}_0}$. Also the proofs are lengthy and hardly reuse results from each other, giving not much insight. General results from optimization cannot be used for several reasons: First, the moments usually cannot be proven to be a descent direction. Second, the learning rate cannot be shown to be a step size valid for the Wolfe conditions for a line search, see [21]. The algorithm for the step taken in iteration t may explicitly contain the variable t in much more complicated ways than $\frac{1}{t}$ in the Robbins-Monro approach [24].

The contribution of this paper is a generally applicable method, based on the theory of discrete time dynamical systems, which proves local convergence of ADAM. The results are purely qualitative because they hold for learning rates sufficiently small, where "sufficiently small" is defined in terms of the eigenvalues of the Hessian in the unknown minimum $w_\star$.

The outline of this paper is as follows: In Section II we will discuss the preliminaries and the idea of the convergence prove, which will be discussed in Section III. The generality of this method is shown in Section IV, where we apply it to other optimizers like AdaDelta or AdaGrad. In numerical experiments in Section V we show the heuristic evidence of our theoretical proof. Finally, in Section VI we summarize the paper and give an outlook on ways to expand the convergence proof.

This paper expands the results and presents proofs that are referenced in [6].

S. Bock and M. Weiß are with the Department of Computer Science and Mathematics, OTH Regensburg, Prüfeninger Str. 58, 93049 Regensburg, Germany

II. Fixed Point Analysis under Perturbation

A. Notation

The symbol $\perp$ denotes the transpose of a vector or matrix. We use the component-wise multiplication and division of vectors, as well as component-wise addition of vectors and scalar without any special notation. It should always be clear from the context which calculation method is used due to the size of the arguments. For $f:\mathbb{R}^n\to\mathbb{R}$ the gradient and Hessian are written as ∇f and $\nabla^2 f$, provided they exist. The vector spaces of functions which are once or twice continuously differentiable are denoted C^1 or C^2 respectively. Throughout this paper we assume $f:\mathbb{R}^n\to\mathbb{R}^n$ at least continuously differentiable, twice continuously differentiable for some results. The open ball with radius r around $x\in\mathbb{R}^n$ is denoted by $B_r(x)=\{y\in\mathbb{R}^n : \|y-x\|<r\}$ and $\|x\|$ is any norm. We denote $\rho(A)=\max\{|\lambda||\lambda \text{ eigenvalue of } A\}$ the spectral radius of a matrix A and $\operatorname{diag}(v)\in\mathbb{R}^{n\times n}$ describes a matrix with components of $v\in\mathbb{R}^n$ on the diagonal.

B. Related Work

Stochastic gradient descent (SGD) becomes an effective method for optimizing noisy tasks. Especially in the area of neural networks SGD variants are partly responsible for big successes in the last years, see e.g. [18] or [13].
Popular first-order SGD methods are AdaGrad [11] and RMSProp [14]. Kingma and Ba combine the advantages of these two methods and introduce the Adaptive Moment Estimation (ADAM) in [17] (see Algorithm 1). Unfortunately, the ADAM

Algorithm 1 ADAM Optimizer

Require: $\alpha\in\mathbb{R}^+$, $\varepsilon\in\mathbb{R}$ $\beta_1,\beta_2\in(0,1)$, $w_0\in\mathbb{R}^n$ and the function $f(w)\in C^2(\mathbb{R}^n,\mathbb{R})$

$m_0=0$, $v_0=0$, $t=0$
while w not converged **do**
  $m_{t+1}=\beta_1 m_t+(1-\beta_1)\nabla_w f(w_t)$
  $v_{t+1}=\beta_2 v_t+(1-\beta_2)\nabla_w f(w_t)^2$
  $w_{t+1}=w_t-\alpha\frac{\sqrt{1-\beta_2^{t+1}}}{(1-\beta_1^{t+1})}\frac{m_{t+1}}{\sqrt{v_{t+1}+\varepsilon^2}}$
  $t=t+1$
end while

optimizer is not always defined the same way. Kingma and Ba [17] use $\sqrt{v}+\varepsilon$ and bias correction. The algorithms in [23] and [9] do not use an ε as well as [28], but the latter initializes $v_0=\varepsilon$. All three apply bias correction in the learning rate α_t. We use bias correction as described in [17, Section 2] and $\sqrt{v+\varepsilon^2}$. $\sqrt{v+\varepsilon^2}$ is more similar to the undisturbed $\sqrt{v}$ than $\sqrt{v}+\varepsilon$ from [17] or $\sqrt{v+\varepsilon}$ from [10].

The differences between the two possible usages of ε are minimal in their effect on the evolution of loss and accuracy during learning (see Figure 13[1] in the appendix)

especially in the area around $v\approx 0$. The main aim by the introduction of ε – avoiding division by 0 – holds in both variants, but $\sqrt{v+\varepsilon^2}$ gives the additional advantage of making the right hand side continuously differentiable for $v\in[0,\infty)$ whereas $\sqrt{v}+\varepsilon$ is not differentiable at $v=0$. Differentiability will be essential in our proof. In Subsection III-A we will make the connection to ADAM as presented in [17] and prove the local convergence, too.

During the last years the ADAM Optimizer has become one of the most used optimization methods for training neural networks. Even if it is apparently working, there is, to the best of our knowledge, still no convergence proof for ADAM. The proof in the original paper [17] was shown wrong, see [4], [5] or [25]. Reddi et. al. in [23] present even a counter example and also introduce an improved method called AMSGrad. However in experiments AMSGrad does not show improvements at all (see [9] or [16]). On the contrary, in some cases it ends up with worse accuracy than ADAM. Chen et al. [9] are showing for non-convex f that $\min_{t=1}^{T} E[\|\nabla f(x_t)\|^2]=O\left(\frac{s_1(T)}{s_2(T)}\right)$. With the assumption of $s_1(T)$ growing slower than $s_2(T)$, one reaches a minimum of $E[\|\nabla f(x_t)\|^2]$ but without a guarantee of staying there. Barakat and Bianchi[3] use a similar interpretation of the ADAM optimizer with a dynamical system viewpoint. uses This approach uses a non-autonomous ordinary differential equation without Lipschitz condition because of the term $\sqrt{v}+\varepsilon$ instead of our non-autonomous system of difference equations. In our opinion, this choice makes the proof longer and more complicated.

In the current work we present a convergence proof of the ADAM optimizer [17] in a complete batch mode using all of the training data. With this assumption we can guarantee, that we are searching the same minimum $w_\star$ in each time step t. Due to the local nature our proof does not assume the convexity of $f(w)$, thus we can guarantee local convergence even for non-convex settings. The hyperparameter setting is only restricted by

$$\frac{\alpha}{\varepsilon}\max_{i=1}^{n}(\mu_i)(1-\beta_1)<2\beta_1+2 \tag{1}$$

with μ_i the i-th eigenvalue of the Hessian $\nabla^2 f(w_\star)$. The counter example in [23] or [19] does not affect our convergence proof, because we consider batch mode only. For example the incremental function in [23] becomes a linear function in batch mode.

All these considerations, we then apply to the most famous algorithms of the adaptive gradient decent family. We subsequently publish all resulting hyperparameter boundaries for each optimizer in Table II. It should be noted that we only focus on the supposedly most important algorithms in the area around ADAM. A table with all convergence statements for every algorithm in the adaptive gradient decent family would be desirable but not feasible due to the amount of algorithms.

C. Idea

We consider the learning algorithm from the standpoint of dynamical systems and define a common state vector x

[1]The experiment is programmed with Keras 2.2.4, Tensorflow 1.13.1 and Python 3.6.8

consisting of the moments – like m and v for ADAM – and the weights, so we have $x = (m, v, w)$. Then the optimization can be written as an iteration $x_{t+1} = T(t, x_t)$ for some function $T : \mathbb{N}_0 \times X \to X$ with $X \subset \mathbb{R}^p$, which defines a non-autonomous dynamical system. The function f to be minimized in the learning process, or rather its gradient, becomes a part of T. If f is at least continuously differentiable a local minimum gives the necessary condition $\nabla f(w_\star) = 0$. We show that this condition leads to a fixed point $x_\star$ of T, where the moments are all zero. We analyse the stability properties of this fixed point and prove local asymptotic stability. This is done by considering a time-variant iteration T as the perturbation of a time-invariant iteration $\bar{T}$ where Banach-like fixed point arguments can be applied. We use the second method of Lyapunov for stability analysis where the vanishing moments simplify the computation and estimates for the eigenvalues. Asymptotic stability however is equivalent to convergence of the iteration defined T to $x_\star$ for all x_0 sufficiently close to $x_\star$. The conditions needed for the fixed point analysis and stability results require the learning rate to be sufficiently small. Note that these results cannot be obtained directly from standard fixed point theorems for autonomous systems, because the iteration index t enters the dynamics. Therefore also estimates of the eigenvalues depend on the iteration t, and even a bound on the spectral radius uniform in t does not give the convergence results presented here: It is well known that $\rho(A) < 1$ implies the existence of a vector norm with induced matrix norm such that $\|A\| < 1$, but this norm depends on A. So $\rho(A_t) \leq c < 1$ for some c for all $t \in \mathbb{N}_0$ does not imply the existence of a *single* norm such that $\|A_t\| < 1$ for all t. We emphasize that the result is purely qualitative, giving no explicit guidance to the choice of the learning rates. The main advantage of our approach is the clearness of the proof, only computation of eigenvalues is needed once the iteration has been written in terms of T and $\bar{T}$. These calculations are much more simple than the lengthy estimates in [17], [23] and [9].

We stress that local convergence result is all one can hope for: A global convergence proof cannot be obtained for most algorithms including ADAM because a 2-cycle of the iteration exists for all values of hyperparameters for simple quadratic objective functions, see [7].

D. Preliminaries

We recall some standard definitions and facts from the theory of difference equations and discrete time dynamical systems, see e.g. [1, Definition 5.4.1] or [15]. Consider $T : \mathbb{N}_0 \times M \to M$ with $M \subset \mathbb{R}^p$ which defines a non-autonomous dynamical system by the iteration

$$x_{t+1} = T(t, x_t), \quad t \in \mathbb{N}_0, x_0 \in M \tag{2}$$

with solutions $x : \mathbb{N}_0 \to M$, $t \mapsto x_t$ depending on the initial value x_0. We use the notations $x_t = x(t; x_0)$ and $x = x(\cdot; x_0)$ to emphasize the dependence of solutions on the initial value if necessary. We always use the initial time $t_0 = 0$.

Autonomous systems constitute the special case where T does not depend on t, so we can abbreviate to $\bar{T} : M \to M$ and write

$$x_{t+1} = \bar{T}(x_t), \quad t \in \mathbb{N}_0, x_0 \in M \tag{3}$$

A point $x_\star \in M$ is called *equilibrium* or *fixed point* if $T(t, x_\star) = x_\star$ for all $t \in \mathbb{N}_0$, so the constant function $x_t = x_\star$ for all $t \in \mathbb{N}_0$ is a solution of (2). In the following the asterisk will always denote equilibria or their components. Consider a solution $x = x(\cdot; x_0)$ of (2). x is called *stable*, if for each $\varepsilon > 0$ there exists $\delta = \delta(\varepsilon)$ such that any solution $\tilde{x} = \tilde{x}(\cdot; \tilde{x}_0)$ of (2) with $\|\tilde{x}_0 - x_0\| < \delta$ fulfills $\|\tilde{x}_t - x_t\| < \varepsilon$ for all $t \in \mathbb{N}_0$.

x is called *attractive* if there exists $\delta > 0$ such that any solution $\tilde{x}$ with $\|\tilde{x}_0 - x_0\| < \delta$ fulfills $\lim_{t\to\infty} \|\tilde{x}_t - x_t\| = 0$. x is called *asymptotically stable* if it is stable and attractive.

Recall that a *contraction* is a self-mapping on some set with Lipschitz constant $L < 1$, i.e. a mapping $\bar{T} : M \to M$, $M \subset \mathbb{R}^n$ with $\|\bar{T}(x) - \bar{T}(y)\| \leq L\|x - y\|$ for all $x, y \in M$. If M is complete, i.e. all Cauchy sequences converge, then a unique fixed point $x_\star \in M$ of $\bar{T}$ exists by the Banach fixed point theorem.

Theorem II.1. ***Linearized asymptotic stability implies local nonlinear stability*** *Consider $\bar{T} : M \to M$ with a fixed point $x_\star$ and $\bar{T}$ continuously differentiable in an open neighbourhood $B_r(x_\star) \subset M$ of $x_\star$. Denote the Jacobian by $D\bar{T}_{x_\star}$, and assume $\|D\bar{T}_{x_\star}\| < 1$ for some norm on $\mathbb{R}^{n\times n}$. Then there exists $0 < \varepsilon \leq r$ and $0 \leq c < 1$ such that for all x_0 with $\|x_0 - x_\star\| < \varepsilon$*

$$\|x(t; x_0) - x_\star\| \leq c^t \|x_0 - x_\star\| \qquad \forall t \in \mathbb{N}_0.$$

i.e. $x_\star$ is locally exponentially and asymptotically stable.

The theorem is the core of the first method of Lyapunov for discrete time systems. For a proof see [12, Corollary 4.35.].

III. Convergence Proof

Let $w \in \mathbb{R}^n$ be the weights of the function $f(w) \in C^2(\mathbb{R}^n, \mathbb{R})$, which has to be minimized. We also define $g(w) := \nabla f(w) \in \mathbb{R}^n$ as the gradient of f and the state variable of our dynamical system $x = (m, v, w)$. With these definitions we can rewrite the ADAM-Optimizer as a system of the form (2).

$$\begin{aligned} m_{t+1} &:= \beta_1 m_t + (1 - \beta_1)\, g(w_t) \\ v_{t+1} &:= \beta_2 v_t + (1 - \beta_2)\, g(w_t)^2 \\ w_{t+1} &:= w_t - \alpha \frac{\sqrt{1 - \beta_2^{t+1}}}{\left(1 - \beta_1^{t+1}\right)} \frac{m_{t+1}}{\sqrt{v_{t+1} + \varepsilon^2}} \end{aligned} \tag{4}$$

So the ADAM optimizer can be written as the iteration of a time-variant dynamical system $x_{t+1} = [m_{t+1}, v_{t+1}, w_{t+1}]^\perp = T(t, x) = T\left(t, [m_t, v_t, w_t]^\perp\right) \in \mathbb{R}^{3n}$. We split the system into an autonomous and a non-autonomous part

$$x_{t+1} = T(t, x_t) = \bar{T}(x_t) + \Theta(t, x_t) \tag{5}$$

with

$$\bar{T}(x_t) = \begin{bmatrix} \beta_1 m_t + (1 - \beta_1)\, g(w_t) \\ \beta_2 v_t + (1 - \beta_2)\, g(w_t)^2 \\ w_t - \alpha \frac{m_{t+1}}{\sqrt{v_{t+1} + \varepsilon^2}} \end{bmatrix} \tag{6}$$

and

$$\Theta(t, x_t) = [0, 0, \alpha\Theta_3(t, x_t)]^\perp \tag{7}$$

$$\Theta_3(t, x_t) = \left(1 - \frac{\sqrt{1 - \beta_2^{t+1}}}{1 - \beta_1^{t+1}}\right) \frac{m_{t+1}}{\sqrt{v_{t+1} + \varepsilon^2}}$$

To avoid lengthy expressions we use m_{t+1} and v_{t+1} as an abbreviation for the updated terms instead of the filters depending on m_t, $g(w_t)$ and v_t. The autonomous system is ADAM without bias correction, the disturbance term Θ adds bias correction which leads to a non-autonomous system. The Jacobian matrix of the autonomous system (6) is

$$J_{\bar{T}}(m_t, v_t, w_t) = \begin{bmatrix} \beta_1 I & 0 & (1-\beta_1)\nabla_w g(w_t) \\ 0 & \beta_2 I & \frac{\partial v_{t+1}}{\partial w_t} \\ \frac{\partial w_{t+1}}{\partial m_t} & \frac{\partial w_{t+1}}{\partial v_t} & \frac{\partial w_{t+1}}{\partial w_t} \end{bmatrix}$$

with

$$\begin{aligned} \frac{\partial v_{t+1}}{\partial w_t} =& 2(1-\beta_2)\operatorname{diag}(g(w_t))\nabla_w g(w_t) \\ \frac{\partial w_{t+1}}{\partial m_t} =& -\alpha \operatorname{diag}\frac{\beta_1}{\sqrt{v_{t+1}+\varepsilon^2}} \\ \frac{\partial w_{t+1}}{\partial v_t} =& \frac{\alpha\beta_2}{2}\operatorname{diag}\left(\frac{m_{t+1}}{(v_{t+1}+\varepsilon^2)^{\frac{3}{2}}}\right) \\ \frac{\partial w_{t+1}}{\partial w_t} =& I - \alpha\Big((1-\beta_1)\operatorname{diag}(v_{t+1}+\varepsilon^2)^{-\frac{1}{2}} \\ & - \operatorname{diag}\left(m_{t+1}(v_{t+1}+\varepsilon^2)^{-\frac{3}{2}} g(w_t)\right)\Big) \\ & \cdot \nabla_w g(w_t) \end{aligned}$$

We have the following simple observation:

Lemma III.1. *Consider a critical point $w_\star$ for f, $\nabla f(w_\star) = 0$. Then $x_\star = (0,0,w_\star)^\perp$ is a fixed point for (6) and (4).*

Conversely, if $x_\star = (0,0,w_\star)^\perp$ is a fixed point then $\nabla f(w_\star) = 0$.

Proof. We start the iteration with $w_0 = w_\star$, $v_0 = 0$ and $m_0 = 0$, i.e. $x_0 = (0,0,w_\star)$. Then (6) gives $x_1 = T(x_0) = x_0$, and inductively $x_t = x_0$ for all t. The same holds for (4).

Conversely, solving the iteration in $x_\star = (0,0,w_\star)^\perp$ for $\nabla f(w_\star)$ immediately gives $\nabla f(w_\star) = 0$. □

Now we investigate the stability of this fixed point with the goal of asymptotic stability for local minima $w_\star$. The analysis is simplified because the m and v components of $x_\star$ are 0. So we reach the following Jacobian:

$$J_{\bar{T}}(0,0,w_\star) = \begin{bmatrix} \beta_1 I & 0 & (1-\beta_1)\nabla_w g(w_\star) \\ 0 & \beta_2 I & 0 \\ \frac{-\alpha\beta_1}{\varepsilon} I & 0 & I - \frac{\alpha(1-\beta_1)}{\varepsilon}\nabla_w g(w_\star) \end{bmatrix}$$

Theorem III.2. *Let $J_{\bar{T}}(m,v,w) \in \mathrm{Mat}_{3n}$ be the Jacobian of system (6) and $w_\star \in \mathbb{R}^n$ a minimum of f with positive definite Hessian $\nabla_w^2 f(w_\star) = \nabla_w g(w_\star)$. Denote $\mu_i \in \mathbb{R}$ with $i = 1,\dots,n$, the i-th eigenvalue of $\nabla_w g(w_\star)$ and all other parameters are defined as in Algorithm 1. Then $J_{\bar{T}}(0,0,w_\star)$ has the eigenvalues, for $i = 1,\dots,n$:*

$$\begin{aligned} \lambda_{1;i} &= \beta_2 \\ \lambda_{2,3;i} &= \frac{(\beta_1+1) \pm \sqrt{(\beta_1+1)^2 - 4\left(\beta_1 - \frac{\alpha\mu_i(\beta_1-1)}{\varepsilon}\right)}}{2} \end{aligned}$$

In the combination of Theorem III.2 and II.1 we still have to show, that $|\lambda_{j,i}| < 1$ holds, then the spectral radius for the Jacobian is smaller than 1 and we prove local convergence.

Theorem III.3. *Let the parameters be defined as in Theorem III.2 and inequality (1) holds, then $\rho(J_{\bar{T}}(0,0,w_\star)) < 1$.*

Corollary III.4. *Let the parameters be defined as in Theorem III.2 and such that $\frac{\alpha}{\varepsilon}\max_{i=1}^n(\mu_i)(1-\beta_1) < 2\beta_1 + 2$ holds for $i \in \{1,\dots,n\}$, then Algorithm 1 converges locally with exponential rate of convergence.*

Proof. Consider the non-autonomous system (5) with $\bar{T}(x_t)$ and $\Theta(t,x_t)$ as defined in equations (6) and (7). The Hessian of f is continuous, so the gradient of f is locally Lipschitz with some constant $L > 0$, $\|g(w_1) - g(w_2)\| \le L\|w_1 - w_2\|$ for all w_1, w_2 in some neighbourhood of $w_\star$. Let all other parameters be defined as in Theorem III.2, especially $\frac{\alpha}{\varepsilon}\max_{i=1}^n(\mu_i)(1-\beta_1) < 2\beta_1 + 2$. Using $m_\star = 0$ and $g(w_\star) = 0$ we estimate

$$\begin{aligned} &\|\Theta(t,x)\| \\ &= \alpha\left|\frac{\sqrt{1-\beta_2^{t+1}}}{1-\beta_1^{t+1}} - 1\right| \\ &\quad \cdot \frac{\|\beta_1 m + (1-\beta_1)g(w)\|}{\sqrt{\beta_2 v + (1-\beta_2)g(w)^2 + \varepsilon^2}} \\ &\le \frac{\alpha}{\varepsilon}\left|\frac{\sqrt{1-\beta_2^{t+1}} - \left(1-\beta_1^{t+1}\right)}{\left(1-\beta_1^{t+1}\right)}\right| \\ &\quad \cdot \|\beta_1 m + (1-\beta_1)g(w)\| \\ &\le \frac{\alpha}{\varepsilon(1-\beta_1)}\left|\frac{\left(1-\beta_2^{t+1}\right) - \left(1-\beta_1^{t+1}\right)^2}{\sqrt{1-\beta_2^{t+1}} + \left(1-\beta_1^{t+1}\right)}\right| \\ &\quad \cdot (\beta_1\|m\| + (1-\beta_1)\|g(w)\|) \\ &\le \frac{C}{4}\left|\left(1-\beta_2^{t+1}\right) - \left(1-\beta_1^{t+1}\right)^2\right| \\ &\quad \cdot (\beta_1\|m - m_\star\| + (1-\beta_1)\|g(w) - g(w_\star)\|) \\ &\le \frac{C}{4}\left|-\beta_2^{t+1} - 2\beta_1^{t+1} + \beta_1^{2(t+1)}\right| \\ &\quad \cdot (\beta_1\|m - m_\star\| + (1-\beta_1)L\|w - w_\star\|) \\ &\le C\beta^{t+1}\left(\beta_1\|m - m_\star\| + (1-\beta_1)L\|w - w_\star\|\right) \end{aligned}$$

where we have used the Lipschitz continuity of g, and set $\beta = \max\{\beta_1, \beta_2, \beta_1^2\}$, $C := \frac{4\alpha}{\varepsilon(1-\beta_1)\left(\sqrt{1-\beta_2}+(1-\beta_1)\right)}$. The term $\beta_1\|m - m_\star\| + (1-\beta_1)L\|w - w_\star\|$ corresponds to a norm

$$\|(\tilde{m},\tilde{w})\|_* := \beta_1\|\tilde{m}\| + (1-\beta_1)L\|\tilde{w}\|, \quad \tilde{m},\tilde{w} \in \mathbb{R}^n$$

on $\mathbb{R}^{2n}$ (which does not depend on $w_\star$). By the equivalence of norms in finite dimensional spaces we can estimate $\|(\tilde{m},\tilde{w})\|_* \le \tilde{C}\|(\tilde{m},\tilde{w})\|$ for some $\tilde{C} > 0$. We continue the estimate:

$$\begin{aligned} &\le C\beta^{t+1}\tilde{C}\|(m - m_\star, w - w_\star)\| \\ &\le (C\beta\tilde{C})\beta^t\|x - x_\star\| =: \bar{C}\beta^t\|x - x_\star\| \end{aligned}$$

for some $\bar{C} > 0$. With this estimate and Theorem B.1, it is sufficient to prove exponential stability of a fixed point of $\bar{T}$. By Theorem III.3 we get $\rho(J_{\bar{T}}(0,0,w_\star)) < 1$. Thus with Theorem II.1 the fixed point $(0,0,w_\star)$ corresponding to the minimum $w_\star$ is locally exponentially stable, and Theorem B.1

gives local exponential convergence of the non-autonomous system $T(t,x)$, i.e. the ADAM algorithm. □

A. Original ADAM Formulation

We recall, that we use the ADAM optimizer with $\frac{1}{\sqrt{v+\varepsilon^2}}$ instead of $\frac{1}{\sqrt{v}+\varepsilon}$. Thus line 5 from Algorithm 1 changes to line $5'$.

$$5': w_{t+1} = w_t - \alpha \frac{\sqrt{1-\beta_2^{t+1}}}{\left(1-\beta_1^{t+1}\right)} \frac{m_{t+1}}{\sqrt{v_{t+1}}+\varepsilon}$$

In order to prove the convergence of the original version, reference is made at this point to [8, Theorem 2.7]. Since [8] is only available as a non-refereed preprint, the proof can be found in the appendix (Theorem B.4).
Consider the ADAM optimizer without the bias correction as in system (3), that is with the term $\frac{1}{\sqrt{v+\varepsilon^2}}$ in the w equation of $x_{t+1} = \bar{T}(x_t)$ with $x = [m,v,w]^\perp$. From Corollary III.4 we know that $x_\star$ is locally exponentially stable. With small changes we can modify this system to the original ADAM formulation of [17] without bias correction.

$$\tilde{x}_{t+1} = \bar{T}(x_t) + h(x_t), \quad t \in \mathbb{N}_0, x_0 \in M \tag{8}$$

with the autonomous system $\bar{T}(x_t)$ as defined in equation (6) and

$$\begin{aligned} h(x_t) &:= [0,0,h_3(x_t)]^\perp \\ h(x_t) &= \alpha m_{t+1}\left(\frac{1}{\sqrt{v_{t+1}}+\varepsilon} - \frac{1}{\sqrt{v_{t+1}+\varepsilon^2}}\right) \end{aligned} \tag{9}$$

Note that we define the original ADAM without bias correction in dependence of the system (3) studied so far and as an autonomous system. By means of this consideration, we can prove the local convergence of system (8).

Corollary III.5. *Assume the original ADAM without bias correction $\tilde{x}_{t+1} = \bar{T}(x_t) + h(x_t)$ as defined in equation (8) and $f: \mathbb{R}^n \to \mathbb{R}$ strictly convex with minimum $w_\star \in \mathbb{R}^n$. Assume $f \in C^2$ and $\nabla^2 f(w_\star)$ positive definite. Assume inequality (1) holds for the hyperparameters. Then the original ADAM without bias correction converges locally with exponential rate of convergence.*

Proof. We start the proof by considering the disturbance $h(x_t)$ from equation (9). We can estimate the difference in $h(x_t)$ as follows:

$$\begin{aligned} &\left|\frac{1}{\sqrt{v}+\varepsilon} - \frac{1}{\sqrt{v+\varepsilon^2}}\right| \\ =&\left|\frac{\sqrt{v+\varepsilon^2}-(\sqrt{v}+\varepsilon)}{\sqrt{v+\varepsilon^2}(\sqrt{v}+\varepsilon)}\right| \\ =&\left|\frac{\left(\sqrt{v+\varepsilon^2}-\sqrt{v}+\varepsilon\right)\left(\sqrt{v+\varepsilon^2}+\sqrt{v}+\varepsilon\right)}{\sqrt{v+\varepsilon^2}(\sqrt{v}+\varepsilon)\left(\sqrt{v+\varepsilon^2}+\sqrt{v}+\varepsilon\right)}\right| \\ =&\frac{2\sqrt{v}\varepsilon}{\sqrt{v+\varepsilon^2}(\sqrt{v}+\varepsilon)\left(\sqrt{v+\varepsilon^2}+\sqrt{v}+\varepsilon\right)} \\ \leq&\frac{2\sqrt{v}\varepsilon}{\sqrt{v}\varepsilon(2\varepsilon)} = \frac{1}{\varepsilon} \end{aligned}$$

The estimate holds for any v, also for $v=0$, and especially for v_{t+1} which appears in h_3. Therefore we can estimate the whole disturbance $\|h(x_t)\|$

$$\begin{aligned} &\|h(x_t)\| \\ \leq&\frac{\alpha}{\varepsilon}\|m_{t+1} - m_\star\| \\ =&\frac{\alpha}{\varepsilon}\|\beta_1 m_t + (1-\beta_1)g(w_t) - m_\star\| \\ \leq&\frac{\alpha}{\varepsilon}(\|\beta_1 m_t - m_\star\| + (1-\beta_1)\|g(w_t)\|) \\ \leq&\frac{\alpha}{\varepsilon}(\beta_1\|m_t - m_\star\| + (1-\beta_1)\|g(w_t) - g(w_\star)\|) \\ \leq&\frac{\alpha}{\varepsilon}(\beta_1\|m_t - m_\star\| + (1-\beta_1)L\|w_t - w_\star\|) \\ \leq&\frac{\alpha}{\varepsilon}\beta_1\|m_t - m_\star\| \\ \leq&C(\varepsilon)\|x_t - x_\star\| \end{aligned}$$

with $C(\varepsilon) = \frac{\alpha}{\varepsilon}$. Thus, the first assumption of Theorem B.5 is fulfilled. We know with the Theorems II.1 and III.3, that the undisturbed system (6) converges locally and exponentially to $x_\star$. With this knowledge and the estimation above, we can apply Theorem B.5. It follows that $\tilde{x}_\star = x_\star$ is a global exponentially stable fixed point for the system (8). □

Involving the bias correction, we have to adjust system (8) to

$$\tilde{x}_{t+1} = T(x_t) + h(x_t) + \tilde{\Theta}(t,x) \tag{10}$$

with

$$\begin{aligned} \tilde{\Theta}(t,x) &:= \left[0,0,\alpha\tilde{\Theta}_3(t,x)\right]^\perp \\ \tilde{\Theta}_3(t,x) &= \left(1 - \frac{\sqrt{1-\beta_2^{t+1}}}{1-\beta_1^{t+1}}\right)\frac{m_{t+1}}{\sqrt{v_{t+1}}+\varepsilon} \end{aligned} \tag{11}$$

Corollary III.6. *Assume the original ADAM with bias correction as the non-autonomous system $\tilde{x}_{t+1} = \bar{T}(x_t) + h(x_t) + \tilde{\Theta}(t,x)$ like defined in equation (10). All other assumptions are adopted by Corollary III.5. Then system (10) converges locally with exponential rate of convergence.*

Proof. Assume the non-autonomous system (7) with $\bar{T}(x_t)$, $h(x_t)$ and $\tilde{\Theta}(t,x)$ as defined in equations (6), (9) and (11).

Since the estimation of $\left\|\tilde{\Theta}(t,x)\right\|$ is analogous to the estimation in the proof of Corollary III.4, reference is made here to this proof and the estimation of $\left\|\tilde{\Theta}(t,x)\right\|$ is given by

$$\begin{aligned}\left\|\tilde{\Theta}(t,x)\right\| \leq & C\beta^{t+1}(\beta_1\|m-m_\star\| \\ & +(1-\beta_1)L\|w-w_\star\|)\end{aligned}$$

Here $L>0$ is the Lipschitz constant of f, $\beta=\max\{\beta_1,\beta_2,\beta_1^2\}$ and $C:=\frac{4\alpha}{\varepsilon(1-\beta_1)\left(\sqrt{1-\beta_2}+(1-\beta_1)\right)}$. Analogous to the proof of Corollary III.4 we can shorten the inequality by the equivalence of norms in finite dimensional spaces. Thus

$$\left\|\tilde{\Theta}(t,x)\right\| \leq \bar{C}\beta^t\|x-x_\star\|$$

for some $\bar{C}>0$. With Corollary III.5 we get the exponential stability of a fixed point of $\tilde{x}_{t+1}=\bar{T}(x_t)+h(x_t)$. Combining this result with the upper estimate and Theorem B.1, we get local exponential convergence of the non-autonomous system (10), i.e. the ADAM algorithm with bias correction defined as in [17]. □

IV. Extension to other optimizers

In the category of adaptive gradient descent algorithms are a several popular ones, where ADAM is only one of them. Some of them can also be seen as momentum methods, and they are all available in a stochastic or mini-batch way. We will also apply our convergence proof to some of these algorithms in the complete batch mode. This shows how general the methodology of the proof is and shows that other algorithms can also benefit from it. To keep this work clear, we note these algorithms in the elegant way from [23]. We assume the Generic Adaptive Method Setup from Algorithm 2 and set only v_t and m_t different for each optimizer (see Table I). Note that we implemented a function $d(m_{t+1})$, which is only interesting for AdaDelta. This is necessary due to the use of m_t in the weight update of w_{t+1}. It will become clear that the essential requirement of our method are stable eigenvalue which are present in most but not all algorithms.

Algorithm 2 Generic Adaptive Method Setup

Require: $\alpha\in\mathbb{R}^+$, $\varepsilon\in\mathbb{R}$, $w_0\in\mathbb{R}^n$ and the function $f(w)\in C^2(\mathbb{R}^n,\mathbb{R})$

$m_0=0$, $v_0=0$, $t=0$
while w not converged **do**
  $v_{t+1}=\Psi(v_t,w_t)$
  $m_{t+1}=\varphi(m_t,w_t)$
  $w_{t+1}=w_t-\alpha\frac{d(m_t,w_t)}{\sqrt{v_{t+1}+\varepsilon^2}}$
  $t=t+1$
end while

RMSProp

First we observe the RMSProp algorithm by Hinton et. al. [14]. We can define it with the following system

$$x_{t+1}=T(x_t)=\begin{bmatrix}\beta v_t+(1-\beta)g(w_t)^2\\ w_t-\alpha\frac{g(w_t)}{\sqrt{v_{t+1}+\varepsilon^2}}\end{bmatrix}$$

Herewith we can calculate the Jacobian analogous to the convergence proof of the ADAM Optimizer.

$$J_{\bar{T}}(v_t,w_t)=\begin{bmatrix}\beta_2 I & \frac{\partial v_{t+1}}{\partial w_t}\\ \frac{\partial w_{t+1}}{\partial v_t} & \frac{\partial w_{t+1}}{\partial w_t}\end{bmatrix}$$

with

$$\begin{aligned}\frac{\partial v_{t+1}}{\partial w_t}=&2(1-\beta)\operatorname{diag}(g(w_t))\nabla_w g(w_t)\\ \frac{\partial w_{t+1}}{\partial v_t}=&\frac{\alpha\beta}{2}\operatorname{diag}\left(\frac{g(w_t)}{(v_{t+1}+\varepsilon^2)^{\frac{3}{2}}}\right)\\ \frac{\partial w_{t+1}}{\partial w_t}=&I-\alpha\Big(\operatorname{diag}(v_{t+1}+\varepsilon^2)^{-\frac{1}{2}}\\ &-\operatorname{diag}\left(g(w_t)(v_{t+1}+\varepsilon^2)^{-\frac{3}{2}}g(w_t)\right)\Big)\\ &\cdot\nabla_w g(w_t)\end{aligned}$$

This is simplified in the minimum $w_\star$ to

$$J_T(0,w_\star)=\begin{bmatrix}\beta I & 0\\ 0 & I-\frac{\alpha}{\varepsilon}\nabla_w g(w_\star)\end{bmatrix}$$

As you can see we reach n times the eigenvalue $\lambda_1=\beta$, which is per definition smaller than 1. The submatrix $I-\frac{\alpha}{\varepsilon}\nabla_w g(w_\star)$ is again symmetric since $\nabla_w g(w_\star)$ is the Hessian of f. Therefore we can diagonalize the matrix and get n times the eigenvalue $\lambda_2=1-\frac{\alpha}{\varepsilon}\mu_i$. Next step is to analyze the absolute value of the eigenvalues.

$$\begin{aligned}&&|\lambda_2|=|1-\frac{\alpha}{\varepsilon}\mu_i|<1\\ \Leftrightarrow&&-2<-\frac{\alpha}{\varepsilon}\mu_i<0\\ \Leftrightarrow&&0<\mu_i<\frac{2\varepsilon}{\alpha}\end{aligned}$$

Due to the fact that we look at eigenvalues of a positive definite matrix, $0<\mu_i$ is clear. So for local convergence with exponential rate in the RMSProp algorithm we only have to fulfill:

$$\max_{i=1}^{n}(\mu_i)<\frac{2\varepsilon}{\alpha}$$

AdaGrad

For AdaGrad, a proof in this form is not possible. The reason for this is $\beta=1$. With this missing parameter, the following Jacobian is created.

$$J_T(v_t,w_t)=\begin{bmatrix}I & 2g(w_t)\nabla_w g(w_t)\\ \frac{1}{2}\operatorname{diag}\left(\frac{g(w_t)}{(v_{t+1}+\varepsilon^2)^{\frac{3}{2}}}\right) & I-\frac{\alpha}{\varepsilon}\nabla_w g(w_t)\end{bmatrix}$$

In the minimum $(0,w_\star)$

$$J_T(0,w_\star)=\begin{bmatrix}I & 0\\ 0 & I-\frac{\alpha}{\varepsilon}\nabla_w g(w_\star)\end{bmatrix}$$

we see that $\lambda_1=1$ no matter which $\max_{i=1}^{n}(\mu_i)$ we calculate. Thus, no convergence statement is possible using our approach.

TABLE I
POPULAR ADAPTIVE GRADIENT DESCENT OPTIMIZERS

Algorithm	$\varphi(m_t, w_t)$	$\psi(v_t, w_t)$	Momentum	$d(m_t, w_t)$
SGD [24]	$g(w_t)$	$1-\varepsilon^2$	$\beta = 1$	$\varphi(m_t, w_t)$
RMSProp [14]	$g(w_t)$	$\beta v_t + (1-\beta) g(w_t)^2$	$0 < \beta < 1$	$\varphi(m_t, w_t)$
AdaGrad [11]	$g(w_t)$	$v_t + g(w_t)^2$	$\beta = 1$	$\varphi(m_t, w_t)$
AdaDelta [29]	$\beta m_t + (1-\beta) \cdot g(w_t)^2 \frac{m_t+\varepsilon^2}{v_{t+1}+\varepsilon^2}$	$\beta v_t + (1-\beta) g(w_t)^2$	$0 < \beta < 1$	$g(w_t)\sqrt{m_t+\varepsilon^2}$
ADAM without bias-correction [17]	$\beta_1 m_t + (1-\beta_1) g(w_t)$	$\beta_2 v_t + (1-\beta_2) g(w_t)^2$	$0 < \beta_1, \beta_2 < 1$	$\varphi(m_t, w_t)$

AdaDelta

Finally, we look at the AdaDelta algorithm from [29]. This algorithm is somewhat more difficult to bring into the system structure. But with different time steps t and $t+1$ for v and m we can write:

$$x_{t+1} = T(x_t) = \begin{bmatrix} \beta v_t + (1-\beta) g(w_t)^2 \\ \beta m_t + (1-\beta) g(w_t)^2 \frac{m_t+\varepsilon^2}{v_{t+1}+\varepsilon^2} \\ w_t - \alpha \frac{\sqrt{m_t+\varepsilon^2}}{\sqrt{v_{t+1}+\varepsilon^2}} g(w_t) \end{bmatrix}$$

Note that we add the learning rate α to the optimizer different to the original paper. If we set $\alpha = 1$ we reach the original formulation from Zeiler [29]. We will discuss some different learning rates in Section V. For $T(x_t)$ we get the Jacobian

$$J_T(v_t, m_t, w_t) = \begin{bmatrix} \beta I & 0 & \frac{\partial v_{t+1}}{\partial w_t} \\ \frac{\partial m_{t+1}}{\partial v_t} & \frac{\partial m_{t+1}}{\partial m_t} & \frac{\partial m_{t+1}}{\partial w_t} \\ \frac{\partial w_{t+1}}{\partial v_t} & \frac{\partial w_{t+1}}{\partial m_t} & \frac{\partial w_{t+1}}{\partial w_t} \end{bmatrix}$$

with

$$\begin{aligned}
\frac{\partial m_{t+1}}{\partial v_t} &= -\beta(1-\beta)\,\mathrm{diag}\left(\frac{g(w_t)^2\left(m_t+\varepsilon^2\right)}{\left(v_{t+1}+\varepsilon^2\right)^2}\right) \\
\frac{\partial m_{t+1}}{\partial m_t} &= \beta I + (1-\beta)\,\mathrm{diag}\left(\frac{g(w_t)^2}{v_{t+1}+\varepsilon^2}\right) \\
\frac{\partial m_{t+1}}{\partial w_t} &= 2(1-\beta)\,\mathrm{diag}\left(\frac{\left(m_t+\varepsilon^2\right)\left(\beta v_t+\varepsilon^2\right) g(w_t)}{\left(v_{t+1}+\varepsilon^2\right)^2}\right) \\
&\quad \cdot \nabla g(w_t) \\
\frac{\partial v_{t+1}}{\partial w_t} &= 2(1-\beta)\,\mathrm{diag}(g(w_t))\nabla_w g(w_t) \\
\frac{\partial w_{t+1}}{\partial v_t} &= \frac{\beta\alpha}{2}\,\mathrm{diag}\left(\frac{\sqrt{m_t+\varepsilon^2}\, g(w_t)}{\left(v_{t+1}+\varepsilon^2\right)^{\frac{3}{2}}}\right) \\
\frac{\partial w_{t+1}}{\partial m_t} &= -\frac{\alpha}{2}\,\mathrm{diag}\left(\frac{g(w_t)}{\sqrt{v_{t+1}+\varepsilon^2}}\frac{1}{\sqrt{m_t+\varepsilon^2}}\right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial w_{t+1}}{\partial w_t} &= I - \alpha\,\mathrm{diag}\left(\frac{\sqrt{m_t+\varepsilon^2}}{\sqrt{v_{t+1}+\varepsilon^2}}\right. \\
&\quad \left. - \frac{\sqrt{m_t+\varepsilon^2}(1-\beta) g(w_t)^2}{\left(v_{t+1}+\varepsilon^2\right)^{\frac{3}{2}}}\right) \nabla_w g(w_t)
\end{aligned}$$

Fortunately, by inserting $x_\star$, this Jacobian is greatly simplified to

$$J_T(0,0,w_\star) = \begin{bmatrix} \beta I & 0 & 0 \\ 0 & \beta I & 0 \\ 0 & 0 & I - \alpha \nabla_w g(w_\star) \end{bmatrix}$$

We can easily identify $2n$ times the eigenvalue $\lambda_1 = \beta$ of the Jacobian. We diagonalize the Hessian of f and identify $\lambda_{2,i} = 1 - \alpha\mu_i$. By observing the spectral radius of the Jacobian we see, that $\lambda_1 = \beta$ is by definition between 0 and 1. For λ_2 we generate the following general inequality.

$$0 < \max_{i=1}^{n}(\mu_i) < \frac{2}{\alpha}$$

Thus, when the inequality above is satisfied, AdaDelta is locally convergent with exponential rate of convergence.

Conclusion

These proofs of convergence, show the generality of this method. Only for AdaGrad we are not able to proof the convergence due to the missing decay rate β. The result can also be used to adjust the hyperparameters for future optimizations. Table II shows the resulting hyperparameter bounding for each algorithm.

It is also quite astonishing that the convergence behaviour of AdaDelta is not dependent on the parameter ε. The convergence behaviour from the relatively similar algorithms ADAM and RMSProp depend heavily on the ratio between α and ε. Many algorithms reduce the learning rate α over time. For ADAM or RMSProp, however, it might also be useful to increase ε. At this point it should be noted that due to the addition with v it is not mathematically equivalent.

V. EXPERIMENTS

For the sake of clarity, in the next two subsection we will look at the ADAM only. We will then apply similar experiments to the other optimizers, which can be found in subsection V-C and in the appendix.

A. Numerical Convergence

The convergence proof in Section III only shows the local convergence under the hyperparameter bound (1). Whether the boundary is strict or whether there are elements outside this boundary that also converge was not answered. To study the numerical behaviour of ADAM, we choose $f(x) = \frac{x^2}{2} +$

TABLE II
HYPERPARAMETER BOUNDS FOR POPULAR ADAPTIVE GRADIENT DESCENT OPTIMIZERS

Algorithm	Hyperparameter bounding
SGD [24]	$\max_{i=1}^{n}(\mu_i) < \frac{2}{\alpha}$
RMSProp [14]	$\max_{i=1}^{n}(\mu_i) < \frac{2\varepsilon}{\alpha}$
AdaGrad [11]	No convergence statement possible with this method
AdaDelta [29]	$\max_{i=1}^{n}(\mu_i) < \frac{2}{\alpha}$
ADAM without bias-correction [17]	$\frac{\alpha}{\varepsilon}\max_{i=1}^{n}(\mu_i)(1-\beta_1) < 2\beta_1 + 2$

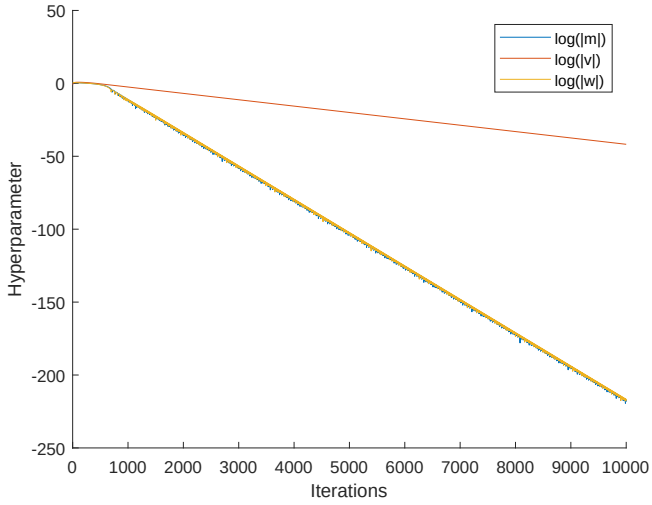

Fig. 1. Exponential convergence behaviour

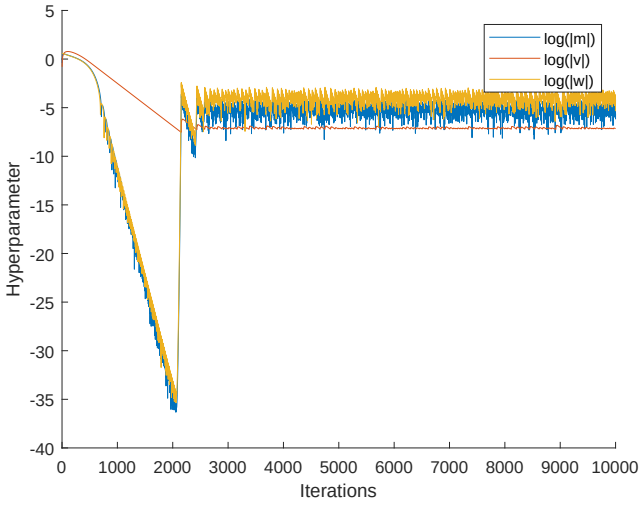

Fig. 2. Chaotic behaviour

$10, \mathbb{R} \to \mathbb{R}$ with the hyperparameters $\beta_1 = 0.9, \beta_2 = 0.99, \alpha = 0.01, m_0 = 0, v_0 = 0$ and $w_0 = 4$. With $\varepsilon = 10^{-2}$ inequality (1) holds and ADAM shows exponential convergence behaviour (see Figure 1). By choosing $\varepsilon = 10^{-8}$ which clearly breaks inequality (1), we see oscillating behaviour as mentioned in [7] (see Figure 2). If we solve inequality (1) with the assumed hyperparameters to ε, we reach the bound $\varepsilon < 2.63158 \cdot 10^{-4}$. When slowly reducing ε, one can recognize the evolution of chaotic behaviour. Already at $\varepsilon = 2.62936 \cdot 10^{-4}$ the exponential convergence is disturbed (see Figure 3) and w starts to jump around $w_\star$ (see Figure 4). We could not find an ε with which our variant converges and the original variant of [17] does not converge or vice versa.

B. Solution Behaviour

To compare our requirements for convergence to the requirements taken by [23] or [17], we make some empirical experiments. First, we look at the different requirements to the hyperparameter.

$$\beta_1 < \sqrt{\beta_2} \tag{12}$$

$$\beta_1^2 < \sqrt{\beta_2} \tag{13}$$

Inequality (1) describes the needed requirement presented in this paper. Problematically in this estimation is, that we

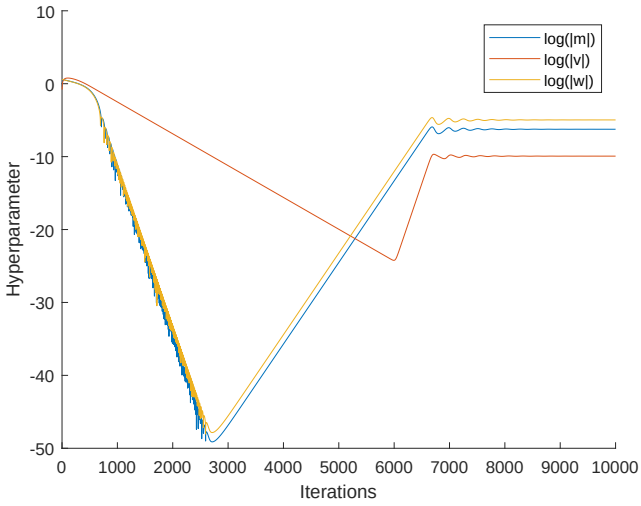

Fig. 3. Chaotic behaviour near boundary

need the maximum eigenvalue of $(1-\beta_1)\nabla_w g(w_\star)$ and consequently $w_\star$. Therefore our estimate is an a posteriori estimate. But with (1) we learn something about the relationship between the hyperparameters. $\frac{\alpha}{\varepsilon}$ has to be very small to fulfill

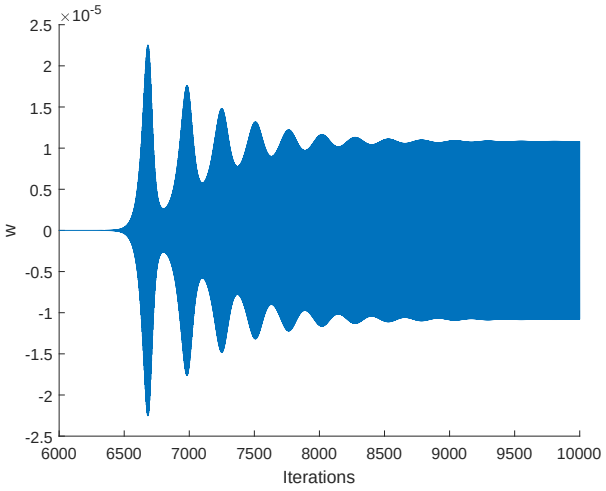

Fig. 4. Chaotic behaviour of w near boundary

TABLE III
CO LOUR DESCRIPTION FOR THE CONVERGENCE INVESTIGATIONS

	Inequality (13) satisfied	Inequality (1) satisfied	ADAM finds solution
green	yes	yes	yes
blue	no	yes	yes
yellow	yes	no	yes
white	no	no	yes
black	yes	yes	no
cyan	no	yes	no
magenta	yes	no	no
red	no	no	no

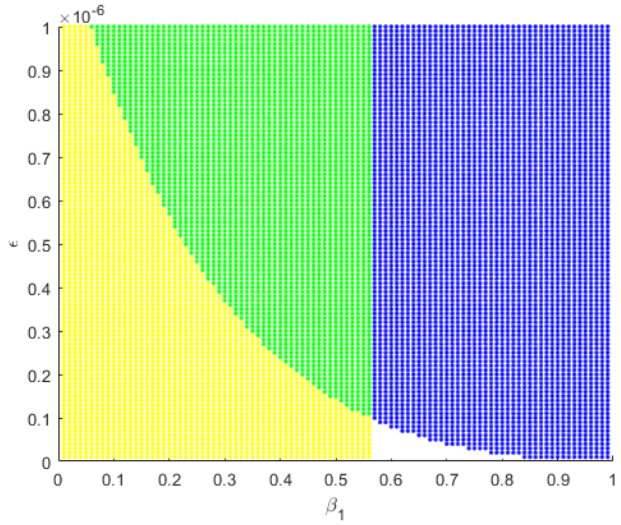

Fig. 5. Iterating over ε and β_1

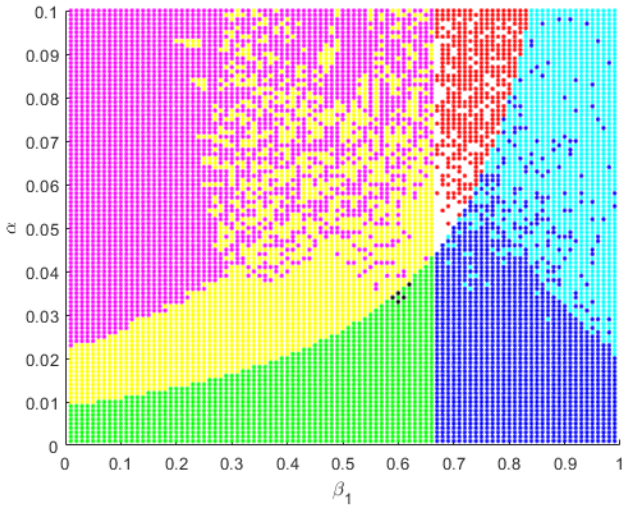

Fig. 6. Iterating over α and β_1

inequality (1). With α small or ε big we always make the weight change smaller and so we do not jump over $w_\star$. Inequality (12) was presented in [23] and inequality (13) was originally presented in [17]. Both are a priori estimations for the hyperparameters.

To show the behaviour of all estimations we set up the following experiments. In Experiment 1 and 2 we want to minimize $f(w) := w^4 + w^3$ with the minimum $w_\star = -\frac{3}{4}$. In Experiment 3 we minimize the multidimensional function $f(w_1, w_2) := (w_1+2)^2 (w_2+1)^2 + (w_1+2)^2 + 0.1 (w_2+1)^2$ with the minimum $w_\star = (-2, -1)$. We run the ADAM optimizer 10000 times in every hyperparameter setting and if the last five iterations $w_{end} \in \mathbb{R}^5$ are near enough to the known solution $w_\star$ the attempt is declared as convergent. Near enough in this setting means that all components of w_{end} are contained in the interval $[w_\star - 10^{-2}, w_\star + 10^{-2}]$. The color coding of our experiments can be found in Table III. To keep the clarity of our results we only compare the original ADAM inequality with our inequality. With inequality (12) we obtain similar figures.

Experiment 1

First, we iterate over $\varepsilon \in \{10^{-4}, \ldots, 10^{-3}\}$ and $\beta_1 \in \{0.01, \ldots, 0.99\}$. The other hyperparameters are fixed $\alpha = 0.001$, $\beta_2 = 0.1$. This setting leads us to figure 5. The only area, where the ADAM optimizer is not finding a solution (red dots), is inside the white area. So both inequalities are not satisfied and the convergence is not given. The white area – ADAM converge but no inequality is satisfied – is formed because we only talk about estimation and not clear boundaries. The blue and yellow area can be made larger or smaller by changing β_2 or α.

Experiment 2

In the second experiment we iterate over $\alpha \in \{0.001, \ldots, 0.1\}$ and $\beta_1 \in \{0.01, \ldots, 0.99\}$. $\beta_2 = 0.2$ and $\varepsilon = 10^{-2}$ are fixed. With the starting point $x_0 = -2$ we reach Figure 6. In the magenta and the cyan area the ADAM method is not reaching the solution, although inequality (13) or (1) is satisfied. The ADAM is oscillating around the solution but do not reach them. The big difference is that the

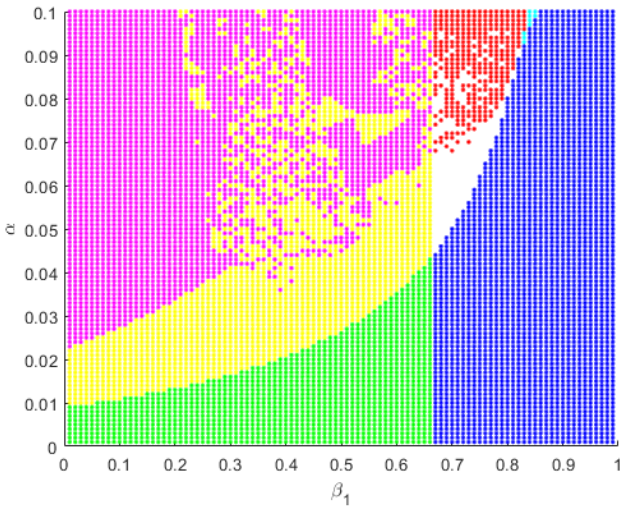

Fig. 7. Iterating over α and β_1 with $x_0 = -0.750000001$

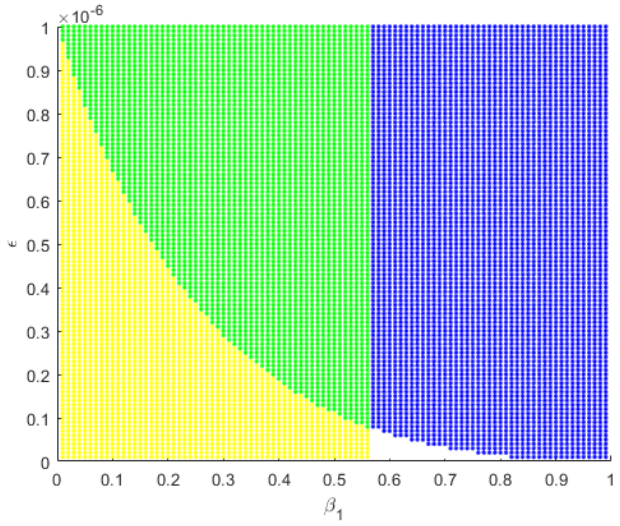

Fig. 8. Experiment 3 :Iterating over ε and β_1.

non-convergence in the cyan area is attributable to the fact that our proof only shows local convergence. By starting in $x_0 = -0.750000001$ the cyan area is almost complete blue (see figure 7). In contrary the magenta area does not change that much.

Experiment 3

In the last experiment we use the same hyperparameters as in experiment 1. Therefore we reach a similar looking Figure 8 by iterating over the parameters. The reason for the enlargement of the blue and green area is the different function $f(x)$, thus different eigenvalues in inequality (1). By observing the convergence behaviour from each of the four differently colored areas in figure 8, we can not spot big differences.

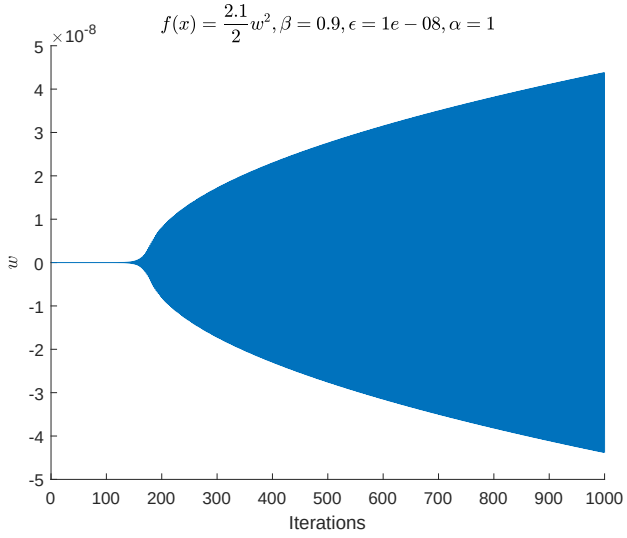

Fig. 9. Convergence behaviour with fixed hyperparameters

C. Experiments with other adaptive gradient decent optimizer

The same experiments can be done with the optimizers of Table II. Since we only have dependencies on ε and α, we only iterate over these hyperparameters with the same color coding (see Table III) without any inequality beside our hyperparameter bounding. To avoid overloading the paper, the experiments can be found in the appendix.

However, AdaDelta plays a special role here, because it is quite astonishing that if you use the parameters suggested by Zeiler ($\alpha = 1$) the convergence behaviour only depends on the eigenvalues of the Hessian of f. To be more clear, there are problems which converge or not converge unimpressed by the hyperparameters. To prove this theoretical result, we use the function $f(x) = \frac{1}{2}cw^2$ with $\max(\mu) = c$. For $c = 1.9$ – our inequality satisfied – we reach a convergent behaviour for each hyperparameter setting we tested. In contrast, if we choose $c = 2.1$ – our inequality is not satisfied – we reach a non convergent behaviour for each hyperparameter setting. At this point we do not use pictures, because they would only show green or red dots.

If we look closer to some fixed hyperparameters we can see, that we are reaching the minimum in both cases. But with $c = 2.1$ we are leaving the minimum at approximately iteration 200 (compare Figure 9 and Figure 10).

Different to the original paper from Zeiler [29], commonly used frameworks for neural networks add a learning rate to the optimizer. Examples are the implementations in Tensorflow [2] or in PyTorch [22]. Tensorflow even goes one step further and mentions the original paper with $\alpha = 1.0$ but sets the default value to $\alpha = 0.001$. As previously shown, choosing a small learning rate can have a positive effect on convergence. See for example Figure 11 with $\alpha = 0.001$. Here we converge more slowly but we do not leave the minimum afterwards.

By iterating over the learning rate α and ε with $c = 2.1$, we can prove again our convergence inequality with the colors coded by Table III (see Figure 12). Even if we increase the

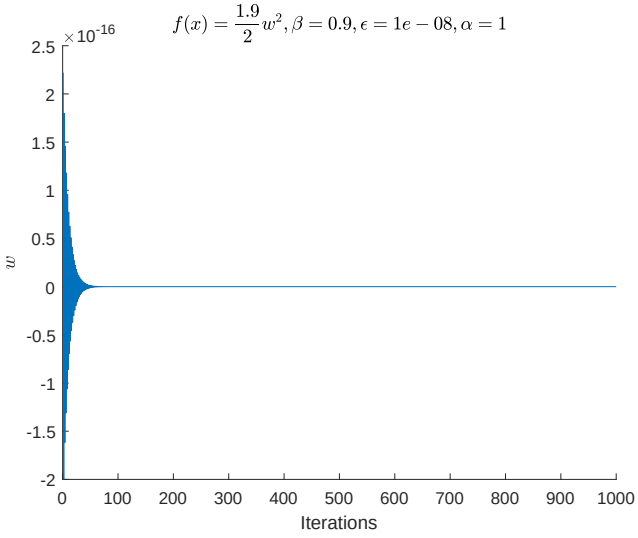

Fig. 10. Convergence behaviour with fixed hyperparameters

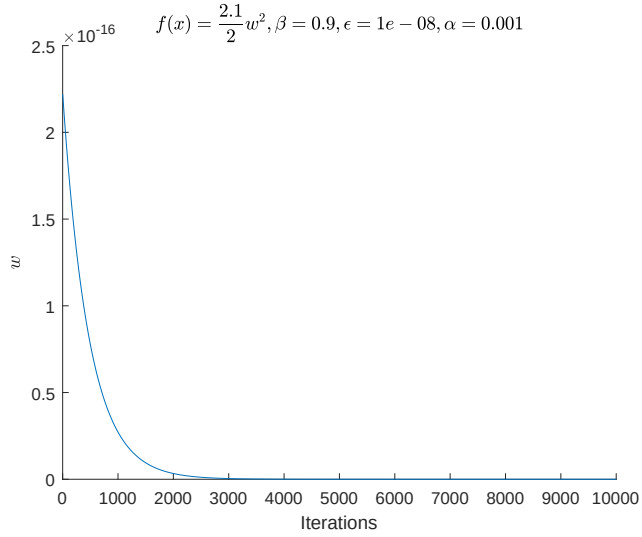

Fig. 11. Convergence behaviour with $\alpha = 0.001$

number of iterations, the red area expands at the cost of the yellow area.

VI. Conclusion and Discussion

In this paper we have presented a local convergence proof of Algorithm 1, the original ADAM [17], SGD [24], RMSProp [14] and AdaDelta [29]. To the best of our knowledge it is the first at all for the ADAM optimizer. We also give an a posteriori boundary for the hyperparameters and show, that the choice of β_2 does not matter for the convergence near a minimum.

However the proof is based on the vanishing gradient condition $\nabla f(w_\star) = 0$ and cannot be used for an incremental algorithm for $f(w) = \frac{1}{N}\sum_{i=1}^{N} f_i(w)$ where different component gradients $g(w_t) = \nabla f_{i_t}(w_t)$ are used in the iterations for the

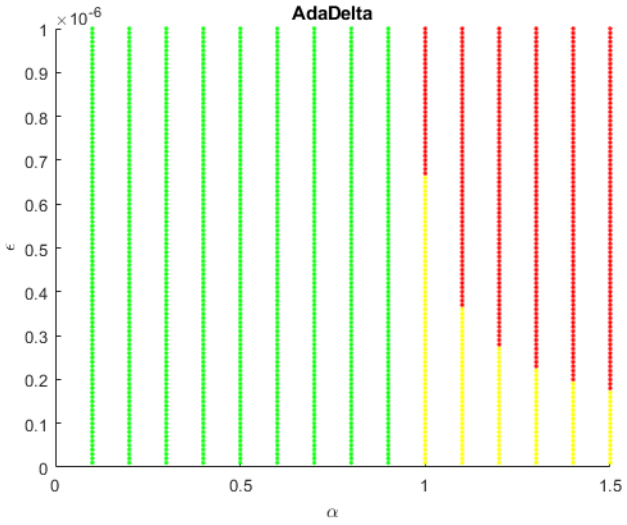

Fig. 12. Convergence behaviour by iterating over α and ε

moments. Clearly $\nabla f(w_\star) = 0$ does not imply $\nabla f_i(w_\star) = 0$ for all components. We are investigating how the incremental dynamical system can be related to the batch system.

The analysis applies to any local minimum with positive definite Hessian and therefore does not require overall convexity. In order to show global convergence of ADAM-like algorithms other methods have to be applied.

Appendix A
Different ε positions

The following two figures describes the accuracy and loss for the fashion-mnist dataset. The only different is the position of the ε.

Appendix B
Convergence Proof

Proof. **(Theorem III.2)**
We see that $J_{\bar{T}}(0,0,w_\star)$ has the n-fold eigenvalue β_2. So we can drop second block row and column of $J_{\bar{T}}$ and investigate the eigenvalues of

$$\begin{bmatrix} \beta_1 I & (1-\beta_1)\nabla_w g(w_\star) \\ -s\beta_1 I & I - s(1-\beta_1)\nabla_w g(w_\star) \end{bmatrix} =: \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

where we use the abbreviation $s := \frac{\alpha}{\varepsilon}$. B and D are symmetric since $\nabla_w g(w_\star)$ is the Hessian of f. By the spectral theorem we can diagonalize B as $B = Q\Lambda Q^\perp$ with an orthogonal matrix Q and a diagonal matrix of eigenvalues Λ. Analogously holds $D = I - Q\Lambda Q^\perp = Q(I-\Lambda)Q^\perp$. We make a similarity transformation with $\tilde{Q} := \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \in \mathrm{Mat}_{2n}$. This leaves the eigenvalues unchanged and gives

$$\tilde{Q}^\perp \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tilde{Q} = \begin{bmatrix} \beta_1 I & (1-\beta_1)\mu_i I \\ -s\beta_1 I & I - s(1-\beta_1)\mu_i I \end{bmatrix}$$

with μ_i the i-th eigenvalue of the Hessian. Eigenvalues does not change in similarity transformations, so we can also calculate

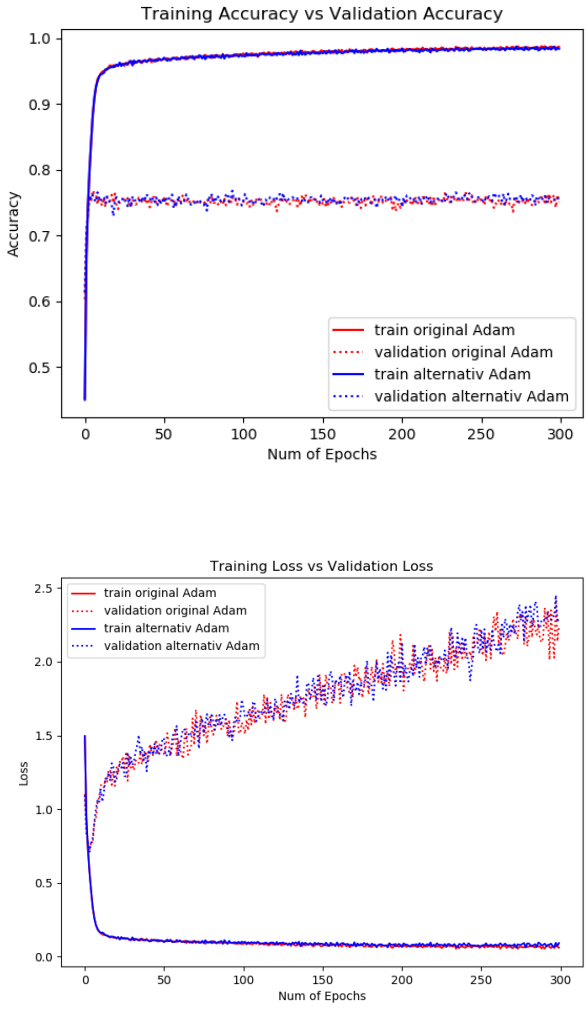

Fig. 13. Train the fashion-mnist dataset ([27]) with the two different ADAM methods

the eigenvalues of our new block matrix with four diagonal sub matrices.

$$\begin{aligned}&\det\begin{bmatrix}(\beta_1-\lambda)I & (1-\beta_1)\mu_i I\\ -s\beta_1 I & (1-s(1-\beta_1)\mu_i-\lambda)I\end{bmatrix}\\ &=\det((\beta_1-\lambda)(1-s(1-\beta_1)\mu_i-\lambda)I\\ &\quad+(1-\beta_1)s\beta_1\mu_i I)\\ &\overset{!}{=}0\end{aligned}$$

Therefore the matrix is a diagonal matrix, we can conclude:

$$\begin{aligned}\det(J_{\bar{T}}(x_\star)-\lambda I)=\prod_{i=1}^{n}&(\beta_1-\lambda)(1-s(1-\beta_1)\mu_i-\lambda)\\ &+(1-\beta_1)s\beta_1\mu_i\end{aligned}$$

Each factor can be written as

$$\lambda^2-(1-s(1-\beta_1)\mu_i+\beta_1)\lambda+\beta_1\overset{!}{=}0$$

and following the statement

$$\begin{aligned}\lambda_{23,i}=\frac{1}{2}&(1-s(1-\beta_1)\mu_i+\beta_1\\ &\pm\sqrt{(1-s(1-\beta_1)\mu_i+\beta_1)^2-4\beta_1}\Big)\end{aligned}$$

is true. □

Proof. **(Theorem III.3)**
We already have calculate the eigenvalues of the Jacobian in Theorem III.2. With these we can easily see, that $|\lambda_1|=|\beta_2|<1$ is satisfied per the requirements of algorithm 1. Therefore we define $\varphi_i:=\frac{\alpha\mu_i}{\varepsilon}(1-\beta_1)$ and observe the absolute value of the two eigenvalues left.

$$|\lambda_{23,i}|=\frac{1}{2}\left|(1+\beta_1-\varphi_i)\pm\underbrace{\sqrt{(1+\beta_1-\varphi_i)^2-4\beta_1}}_{①}\right|$$

First we look at upper bound of the eigenvalues. For this we take term ① combined with the regrets for φ_i:

$$\begin{aligned}\sqrt{(1+\beta_1-\varphi_i)^2-4\beta_1}&<\sqrt{(1+\beta_1)^2-4\beta_1}\\ &=\pm(1-\beta_1)\end{aligned}$$

So if we put this in $\lambda_{23,i}$ we have the inequality $|\lambda_{23,i}|<\frac{1}{2}|1+\beta_1-\varphi_i\pm(1-\beta_1)|$. Easy to see are the two cases:

$$\begin{aligned}&|\lambda_{23,i}|<1 && \text{with +}\\ &|\lambda_{23,i}|<\beta_1<1 && \text{with -}\end{aligned}$$

In both cases we see that the eigenvalues are smaller than 1 in absolute value. To show the lower bound $\lambda_{23,i}>-1$, we look again at term ①.

$$\underbrace{\sqrt{(1+\beta_1-\varphi_i)^2-4\beta_1}}_{\in\mathbb{C}\setminus\mathbb{R}}=i\sqrt{4\beta_1-(1+\beta_1-\varphi_i)^2}$$

Then we can write:

$$\begin{aligned}|\lambda_{23,i}|&=\frac{1}{2}\sqrt{(1+\beta_1-\varphi_i)^2+4\beta_1-(1+\beta_1-\varphi_i)^2}\\ &=\sqrt{\beta_1}<1\end{aligned}$$

The last inequality is given by the requirements of Theorem III.3 and so we proved the whole Theorem. □

Theorem B.1. ***Convergence to fixed point with perturbation*** *Let $M\subset\mathbb{R}^n$ be a complete set, $\bar{T}:M\to M$ Lipschitz continuous with $L<1$, $x_\star\in M$ the unique fixed point of $\bar{T}$. Assume $B_r(x_\star)\subset M$ for some $r>0$. Recall that the non-autonomous system (5) is defined by*

$$\tilde{x}_{t+1}=T(\tilde{x}_t):=\bar{T}(\tilde{x}_t)+\Theta(t,\tilde{x}_t)$$

for $\Theta:\mathbb{N}_0\times M\to\mathbb{R}^n$ with the bound $\|\Theta(t,\tilde{x})\|\le C\beta^t\|\tilde{x}-x_\star\|$ for all $\tilde{x}_t\in M$, $t\in\mathbb{N}_0$ for some $C\ge 0$ and $0<\beta<1$. Then there exists $\varepsilon>0$ such that for all $\tilde{x}_0\in M$ with $\|\tilde{x}_0-x_\star\|<\varepsilon$ the iteration defined by (5) is well-defined, i.e. stays in M, and converges to $x_\star$.

Proof. Let $x=x(\cdot,\tilde{x}_0)$ be the solution of the undisturbed iteration $x_{t+1}=\bar{T}(x_t)$ with initial condition $\tilde{x}_0$, $\tilde{x}=\tilde{x}(\cdot,\tilde{x}_0)$ the

corresponding solution of (5). We define $e_t := \|\tilde{x}_t - x_\star\|$, and estimate using the assumptions

$$\begin{aligned} e_{t+1} &= \|\bar{T}(\tilde{x}_t) + \Theta(t,\tilde{x}_t) - x_\star\| \\ &= \|\bar{T}(\tilde{x}_t) - \bar{T}(x_\star) + \Theta(t,\tilde{x}_t)\| \\ &\leq \|\bar{T}(\tilde{x}_t) - \bar{T}(x_\star)\| + \|\Theta(t,\tilde{x}_t)\| \\ &\leq L\|\tilde{x}_t - x_\star\| + C\beta^t\|\tilde{x}_t - x_\star\| \\ &= (L + C\beta^t)e_t \end{aligned}$$

Choosing t large enough, we get $0 < L + C\beta^t \leq \tilde{L} < 1$ for all $t \geq K$ because $\beta, L < 1$. Then

$$e_t \leq \left(\prod_{k=1}^{K}(L + C\beta^k)\right)\tilde{L}^{t-K}e_0 =: \tilde{C}\tilde{L}^{t-K}e_0$$

with $\tilde{C}$ independent of $\tilde{x}_0$. So e_t converges to 0 exponentially.

The arguments so far have only been valid if $\tilde{x}_t \in M$, i.e. the iteration is well defined. But choosing $\tilde{x}_0$ such that $e_0 = \|\tilde{x}_0 - x_\star\| < \frac{r}{\tilde{C}}$ small enough that we can achieve $e_t \leq \tilde{C}e_0 < r$. □

Theorem B.2. ***Determinants of Block Matrices [26]***

Let $M = \begin{bmatrix} AB \\ CD \end{bmatrix} \in \mathrm{Mat}_{2n}$ *be a block matrix with* $A, B, C, D \in \mathbb{R}^{n\times n}$. *If* C *and* D *commute, then* $\det(M) = \det(AD - BC)$ *holds.*

Lemma B.3. *Let* $f \in C^2(\mathbb{R}^n, \mathbb{R})$, $x_\star \in \mathbb{R}^n$ *with* $\nabla f(x_\star) = 0$ *and* $\nabla^2 f(x_\star)$ *invertible. Then there exist* $\varepsilon > 0$ *and* $C > 0$ *with* $\|\nabla f(x)\| \geq C\|x - x_\star\|$ *for all* $x \in B_\varepsilon(x_\star)$.

Proof. As f is C^2 we have $\nabla f(x) - \nabla f(x_\star) - \nabla^2 f(x_\star) = o(\|x - x_\star\|)$. So for each $\delta > 0$ there exists $\varepsilon > 0$ with $\left\|\nabla f(x) - \nabla f(x_\star) - \nabla^2 f(x_\star)\right\| \leq \delta\|x - x_\star\|$ for all $x \in B_\varepsilon(x_\star)$. Assume w.l.o.g. $\delta < \frac{1}{\left\|\nabla^2 f(x_\star)^{-1}\right\|}$. Then we have

$$\begin{aligned} \|\nabla f(x)\| &\geq \left\|\nabla^2 f(x_\star)(x - x_\star)\right\| \\ &\quad - \left\|\nabla f(x) - \nabla f(x_\star) - \nabla^2 f(x_\star)(x - x_\star)\right\| \\ &\geq \frac{1}{\left\|\nabla^2 f(x_\star)^{-1}\right\|}\|x - x_\star\| - \delta\|x - x_\star\| \\ &=: C\|x - x_\star\| \end{aligned}$$

with $C = \frac{1}{\left\|\nabla^2 f(x_\star)^{-1}\right\|} - \delta > 0$ by choice of δ. □

Theorem B.4. ***Exponential stability implies existence of a Lyapunov function, arbitrary fixed point***

Let $x_\star$ *be an fixed point for the nonlinear system the autonomous system* $x(t+1) = f(x(t))$ *where* $f : D \to \mathbb{R}^n$ *is continuously differentiable and* $D = \{x \in \mathbb{R}^n \,|\, \|x - x_\star\| < r\}$. *Let* k, λ *and* r_0 *be positive constants with* $r_0 < r/k$. *Let* $D_0 = \{x \in \mathbb{R}^n \,|\, \|x - x_\star\| < r_0\}$. *Assume that the solution of the system satisfy*

$$\|x(t, x_0)\| \leq k\|x_0 - x_\star\|e^{-\lambda t}, \forall x_0 \in D_0, \forall t \geq 0 \qquad (14)$$

Then there there is a function $V : D_0 \to \mathbb{R}$ *with*

$$\begin{aligned} c_1\|x - x_\star\|^2 \leq V(x) \leq c_2\|x - x_\star\|^2 \\ V(f(x)) - V(x) \leq -c_3\|x - x_\star\|^2 \\ |V(x) - V(y)| \leq c_4\|x - y\|(\|x - x_\star\| + \|y - x_\star\|) \end{aligned}$$

Note: In the proof of this theorem the norm denotes the 2-norm. The equivalence of norms then easily transfers the result to other norms.

Proof. Let $\phi(t, x_0)$ be the solution of $x_{t+1} = f(x_k)$ at time t starting from x_0 at time $k = 0$ and $x_\star$ be an equilibrium point of the system. Let

$$V(x_0) = \sum_{t=0}^{N-1}(\phi(t, x_0) - x_\star)^\perp(\phi(t, x_0) - x_\star)$$

for some integer variable N to be set. Then

$$\begin{aligned} V(x_0) &= (x_0 - x_\star)^\perp(x_0 - x_\star) \\ &\quad + \sum_{t=1}^{N-1}(\phi(t, x_0) - x_\star)^\perp(\phi(t, x_0) - x_\star) \\ &\geq (x_0 - x_\star)^\perp(x_0 - x_\star) = \|x_0 - x_\star\| \end{aligned}$$

and on the other hand, using (14) we have

$$\begin{aligned} V(x_0) &= \sum_{t=0}^{N-1}(x_t - x_\star)^\perp(x_t - x_\star) \\ &\leq \sum_{t=0}^{N-1} k^2\|x_0 - x_\star\|^2 e^{-2\lambda t} \\ &\leq k^2\left(\frac{1 - e^{-2\lambda N}}{1 - e^{-2\lambda}}\right)\|x_0 - x_\star\|^2 \end{aligned}$$

We have shown that there exists c_1 and c_2 such that

$$c_1\|x_0 - x_\star\|^2 \leq V(x_0) \leq c_2\|x_0 - x_\star\|^2$$

is satisfied. Now, since $\phi(t, f(x_0)) = \phi(t, \phi(1, x_0)) = \phi(t+1, x_0)$,

$$\begin{aligned} &V(f(x)) - V(x) \\ &= \sum_{t=0}^{N-1}(\phi(t+1, x_0) - x_\star)^\perp(\phi(t+1, x_0) - x_\star) \\ &\quad - \sum_{t=0}^{N-1}(\phi(t, x_0) - x_\star)^\perp(\phi(t, x_0) - x_\star) \\ &= \sum_{j=1}^{N}(\phi(j, x_0) - x_\star)^\perp(\phi(j, x_0) - x_\star) \\ &\quad - \sum_{t=0}^{N-1}(\phi(t, x_0) - x_\star)^\perp(\phi(t, x_0) - x_\star) \\ &= (\phi(N, x_0) - x_\star)^\perp(\phi(N, x_0) - x_\star) \\ &\quad - (x_0 - x_\star)^\perp(x_0 - x_\star) \\ &\leq k^2 e^{-2\lambda n}\|x_0 - x_\star\|^2 - \|x_0 - x_\star\|^2 \\ &= -\left(1 - k^2 e^{-2\lambda N}\right)\|x_0 - x_\star\|^2 \end{aligned}$$

Now we choose N big enough so that $1 - k^2 e^{-2\lambda N}$ is greater than 0 and also the second property has been proven. For the third property, since f is continuously differentiable it is also Lipschitz over the bounded domain D, with a Lipschitz constant L, for which it holds $\|f(x) - f(y)\| \leq L\|x - y\|$. Then

$$\begin{aligned} &\|\phi(t+1, x_0) - \phi(t+1, y_0)\| \\ &= \|f(\phi(t, x_0)) - f(\phi(t, y_0))\| \\ &\leq L\|\phi(t, x_0) - \phi(t, y_0)\| \end{aligned}$$

and by induction

$$\|\phi(t,x_0)-\phi(t,y_0)\| \le L^t \|x_0-y_0\|$$

Consider now the difference $|V(x_0)-V(y_0)|$

$$\begin{aligned}
&= \Big| \sum_{t=1}^{N-1} \Big((\phi(t,x_0)-x_\star)^\perp (\phi(t,x_0)-x_\star) \\
&\quad - (\phi(t,y_0)-x_\star)^\perp (\phi(t,y_0)-x_\star) \Big) \Big| \\
&= \Big| \sum_{t=0}^{N-1} \Big((\phi(t,x_0)-x_\star)^\perp \\
&\quad \cdot ((\phi(t,x_0)-x_\star)-(\phi(t,y_0)-x_\star)) \\
&\quad + (\phi(t,y_0)-x_\star)^\perp \\
&\quad \cdot ((\phi(t,x_0)-x_\star)-(\phi(t,y_0)-x_\star)) \Big) \Big| \\
&\le \sum_{t=0}^{N-1} \Big(\|\phi(t,x_0)-x_\star\| \, \|\phi(t,x_0)-\phi(t,y_0)\| \\
&\quad + \|\phi(t,y_0)-x_\star\| \, \|\phi(t,x_0)-\phi(t,y_0)\| \Big) \\
&\le \sum_{t=0}^{N-1} (\|\phi(t,x_0)-x_\star\| + \|\phi(t,y_0)-x_\star\|) L^t \|x_0-y_0\| \\
&\le \left[\sum_{t=0}^{N-1} k e^{-\lambda t} L^t \right] (\|x_0-x_\star\| + \|y_0-x_\star\|) \|x_0-y_0\| \\
&\le c_4 (\|x_0-x_\star\| + \|y_0-x_\star\|) \|x_0-y_0\|
\end{aligned}$$

and so we have proven the last inequality. □

Theorem B.5. ***Exponential convergence under disturbances*** *Suppose $f:\mathbb{R}^n \to \mathbb{R}^n$ Lipschitz continuous with L_f and the discrete time system $x_{t+1}=f(x_t), \quad x(0)=x_0$ Assume $f(x_\star)=x_\star$ and that $x_\star$ is a global exponentially stable fixed point. Assume $h:\mathbb{R}^n \to \mathbb{R}^n$ Lipschitz continuous with L_h, with $h(x_\star)=0$ and $\|h(x)\| \le L_h \|x-x_\star\|$ for all $x \in \mathbb{R}^n$ and the discrete time system*

$$\tilde{x}_{t+1} = f(\tilde{x}_t) + h(\tilde{x}_t), \quad \tilde{x}(0)=\tilde{x}_0 \tag{15}$$

Then, if L_h is small enough, $\tilde{x}_\star = x_\star$ is a global exponentially stable fixed point for (15) as well.

Proof. The converse Lyapunov theorem ensures the existence of a function V with properties

$$\begin{aligned}
c_1 \|x-x_\star\|^2 \le V(x) \le c_2 \|x-x_\star\|^2 \\
\Delta V(x) = V(x_{t+1}) - V(t) \le -c_3 \|x-x_\star\|^2 \\
|V(x)-V(y)| \le c_4 \|x-y\| (\|x-x_\star\| + \|y-x_\star\|)
\end{aligned}$$

We show that, for L_h small enough, $\tilde{V}(\tilde{x}) := V(\tilde{x})$ is also a Lyapunov function for (15). We use the tilde symbol to denote time derivatives along (15):

$$\begin{aligned}
\tilde{\Delta}\tilde{V}(\tilde{x}) &= \tilde{V}(f(\tilde{x})+h(\tilde{x})) - \tilde{V}(\tilde{x}) \\
&= V(f(\tilde{x})+h(\tilde{x})) - V(\tilde{x}) \\
&= V(f(\tilde{x})) - V(\tilde{x}) + V(f(\tilde{x})+h(\tilde{x})) \\
&\quad - V(f(\tilde{x}))
\end{aligned}$$

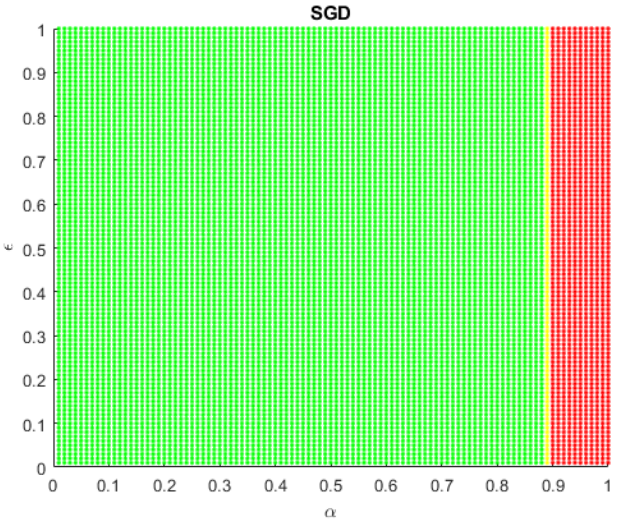

Fig. 14. Iterating over ε and α with $x_0 = -0.65$

The properties of V and h give

$$\begin{aligned}
\tilde{\Delta}\tilde{V}(\tilde{x}) &\le -c_3 \|\tilde{x}-x_\star\|^2 + c_4 \|h(\tilde{x})\| \\
&\quad \cdot (\|f(\tilde{x})+h(\tilde{x})-x_\star\| + \|f(\tilde{x})-x_\star\|) \\
&= -c_3 \|\tilde{x}-x_\star\|^2 + c_4 \|h(\tilde{x})\| \\
&\quad \cdot (\|f(\tilde{x})-x_\star+h(\tilde{x})\| + \|f(\tilde{x})-x_\star\|) \\
&\le -c_3 \|\tilde{x}-x_\star\|^2 + c_4 L_h \|\tilde{x}-x_\star\| \\
&\quad \cdot (\|f(\tilde{x})-f(x_\star)\| + \|h(\tilde{x})\| \\
&\quad + \|f(\tilde{x})-f(x_\star)\|) \\
&\le -c_3 \|\tilde{x}-x_\star\|^2 + c_4 L_h \|\tilde{x}-x_\star\| \\
&\quad \cdot (L_f \|\tilde{x}-x_\star\| + L_h \|\tilde{x}-x_\star\| + L_f \|\tilde{x}-x_\star\|) \\
&= -(c_3 - c_4 L_h (L_f + L_h + L_f)) \|\tilde{x}-x_\star\|^2
\end{aligned}$$

So for L_h small enough we get exponential convergence. □

APPENDIX C
EXPERIMENTS FOR SGD AND RMSPROP

To prove the validity of the inequalities in Table II, we make the same experiments as in Subsection V-B. We set the fixed hyperparameter $\beta = 0.1$ and iterate over $\varepsilon \in \{10^{-2}, \cdots, 1\}$ and $\alpha \in \{0.01, \cdots, 1\}$. In Figure 14 we can see our predicted vertical border at $\alpha = 0.88$ and after a short yellow area – where the SGD also converge but our hyperparameter bounding is not fulfilled – we see the red area – SGD is not converging – at around $\alpha = 0.9$. In Figure 15 we can see a similar behaviour. We can detect some yellow spots, where our inequality is not fulfilled but the RMSProp converges. However, there are no black areas where this is reversed. Denote that if we set x_0 far away from $x_\star$ we get some black area due to the fact that we only prove local convergence. In the experiments the SGD had a much smaller convergence area as RMSProp.

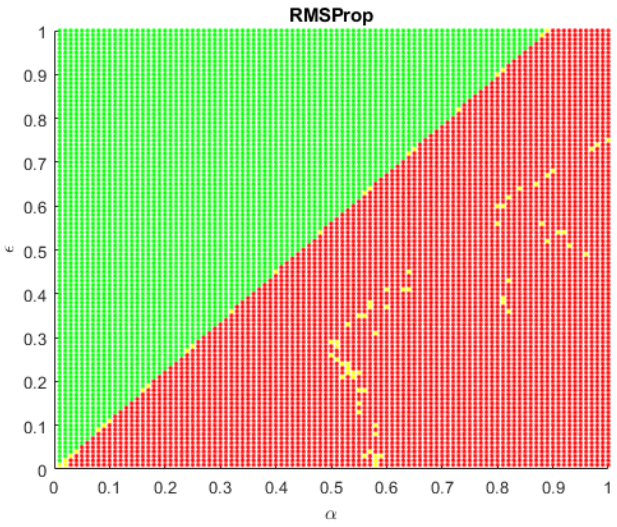

Fig. 15. Iterating over ε and α with $x_0 = -2$

ACKNOWLEDGMENT

This paper presents results of the project "LeaP – Learning Poses" supported by the Bavarian Ministry of Science and Art under Kap. 15 49 TG 78.

REFERENCES

[1] R. P. Agarwal, Z. Nashed, and E. Taft, *Difference Equations and Inequalities: Theory, Methods, and Applications*, 2nd ed., ser. Chapman and Hall/CRC Pure and Applied Mathematics Ser. Boca Raton: Chapman and Hall/CRC, 2000. [Online]. Available: https://ebookcentral.proquest.com/lib/gbv/detail.action?docID=5322044

[2] M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *CoRR*, vol. abs/1603.04467, 2016.

[3] A. Barakat and P. Bianchi, "Convergence of the ADAM algorithm from a Dynamical System Viewpoint," *CoRR*, vol. abs/1810.02263, 2018.

[4] S. Bock, "Rotationsermittlung von Bauteilen basierend auf neuronalen Netzen (unpublished)," M.Sc. thesis, Ostbayerische Technische Hochschule Regensburg, Regensburg, 2017.

[5] S. Bock, M. G. Weiß, and G. Josef, *An improvement of the convergence proof of the ADAM-Optimizer*. OTH Clusterkonferenz 2018, 2018.

[6] S. Bock and M. G. Weiß, "A Proof of Local Convergence for the Adam Optimizer," in *International Joint Conference on Neural Networks IJCNN 2019*, Budapest, 2019.

[7] S. Bock and M. Weiß, "Non-convergence and Limit Cycles in the Adam Optimizer," in *Artificial neural networks and machine learning - ICANN 2019: deep learning*, ser. LNCS sublibrary: SL1 - Theoretical computer science and general issues, I. V. Tetko, V. Kůrková, P. Karpov, and F. J. Theis, Eds. Cham, Switzerland: Springer, 2019, vol. 11728, pp. 232–243.

[8] N. Bof, R. Carli, and L. Schenato, "Lyapunov Theory for Discrete Time Systems," 2018. [Online]. Available: https://arxiv.org/abs/1809.05289

[9] X. Chen, S. Liu, R. Sun, and M. Hong, "On the Convergence of A Class of Adam-Type Algorithms for Non-Convex Optimization."

[10] A. B. da Silva and M. Gazeau, "A general system of differential equations to model first order adaptive algorithms," 2018. [Online]. Available: https://arxiv.org/pdf/1810.13108

[11] J. C. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[12] S. Elaydi. *An Introduction to Difference Equations*. Undergraduate Texts in Mathematics. Springer Science+Business Media Inc, New York, NY, third edition edition, 2005.

[13] G. E. Hinton, N. Srivastava, and K. Swersky, "Overview of mini-batch gradient descent (unpublished)," Toronto. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[14] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[15] W. G. Kelley and A. C. Peterson, *Difference equations: An introduction with applications*, 2nd ed. San Diego, Calif.: Harcourt/Academic Press, 2001. [Online]. Available: http://www.loc.gov/catdir/description/els033/99069847.html

[16] N. S. Keskar and R. Socher, "Improving Generalization Performance by Switching from Adam to SGD." [Online]. Available: http://arxiv.org/pdf/1712.07628v1

[17] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[19] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive Gradient Methods with Dynamic Bound of Learning Rate," in *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, 2019.

[20] J. E. Nesterov, *Introductory lectures on convex optimization: A basic course*, ser. Applied optimization. Boston, Mass.: Kluwer Acad. Publ, 2004, vol. APOP 87. [Online]. Available: http://www.loc.gov/catdir/enhancements/fy0822/2003061994-d.html

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library."

[23] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *CoRR*, vol. abs/1904.09237, 2019.

[24] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951. [Online]. Available: https://projecteuclid.org/download/pdf_1/euclid.aoms/1177729586

[25] D. M. Rubio. *Convergence Analysis of an Adaptive Method of Gradient Descent*. M.Sc. thesis, University of Oxford and University of Oxford, Oxford, 2017.

[26] J. R. Silvester, "Determinants of Block Matrices," London. [Online]. Available: http://www.ee.iisc.ac.in/new/people/faculty/prasantg/downloads/blocks.pdf

[27] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms." [Online]. Available: http://arxiv.org/pdf/1708.07747v2

[28] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu, "A Sufficient Condition for Convergences of Adam and RMSProp," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 11127–11135, Computer Vision Foundation / IEEE, 2019.

[29] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *CoRR*, vol. abs/1212.5701, 2012.