Abstract
Adaptive gradient methods have become recently very popular, in particular as they have been shown to be useful in the training of deep neural networks. In this paper we have analyzed RMSProp, originally proposed for the training of deep neural networks, in the context of online convex optimization and show $\sqrt{T}$-type regret bounds. Moreover, we propose two variants SC-Adagrad and SC-RMSProp for which we show logarithmic regret bounds for strongly convex functions. Finally, we demonstrate in the experiments that these new variants outperform other adaptive gradient techniques or stochastic gradient descent in the optimization of strongly convex functions as well as in training of deep neural networks.

1. Introduction
There has recently been a lot of work on adaptive gradient algorithms such as Adagrad (Duchi et al., 2011), RMSProp (Hinton et al., 2012), ADADELTA (Zeiler, 2012), and Adam (Kingma & Bai, 2015). The original idea of Adagrad to have a parameter specific learning rate by analyzing the gradients observed during the optimization turned out to be useful not only in online convex optimization but also for training deep neural networks. The original analysis of Adagrad (Duchi et al., 2011) was limited to the case of all convex functions for which it obtained a data-dependent regret bound of order $O(\sqrt{T})$ which is known to be optimal (Hazan, 2016) for this class. However, a lot of learning problems have more structure in the sense that one optimizes over the restricted class of strongly convex functions. It has been shown in (Hazan et al., 2007) that one can achieve much better logarithmic regret bounds for the class of strongly convex functions.

The goal of this paper is twofold. First, we propose SC-Adagrad which is a variant of Adagrad adapted to the strongly convex case. We show that SC-Adagrad achieves a logarithmic regret bound for the case of strongly convex functions, which is data-dependent. It is known that such bounds can be much better in practice than data independent bounds (Hazan et al., 2007),(McMahan, 2014). Second, we analyze RMSProp which has become one of the standard methods to train neural networks beyond stochastic gradient descent. We show that under some conditions on the weighting scheme of RMSProp, this algorithm achieves a data-dependent $O(\sqrt{T})$ regret bound. In fact, it turns out that RMSProp contains Adagrad as a special case for a particular choice of the weighting scheme. Up to our knowledge this is the first theoretical result justifying the usage of RMSProp in online convex optimization and thus can at least be seen as theoretical support for its usage in deep learning. Similarly, we then propose the variant SC-RMSProp for which we also show a data-dependent logarithmic regret bound similar to SC-Adagrad for the class of strongly convex functions. Interestingly, SC-Adagrad has been discussed in (Ruder, 2016), where it is said that “it does not to work”. The reason for this is that SC-Adagrad comes along with a damping factor which prevents potentially large steps in the beginning of the iterations. However, as our analysis shows this damping factor has to be rather large initially to prevent large steps and should be then monotonically decreasing as a function of the iterations in order to stay adaptive. Finally, we show in experiments on three datasets that the new methods are competitive or outperform other adaptive gradient techniques as well as stochastic gradient descent for strongly convex optimization problem in terms of regret and training objective but also perform very well in the training of deep neural networks, where we show results for different networks and datasets.

2. Problem Statement
We first need some technical statements and notation and then introduce the online convex optimization problem.
2.1. Notation and Technical Statements

We denote by $[T]$ the set $\{1, \ldots , T\}$. Let $A \in \mathbb{R}^{d \times d}$ be a symmetric, positive definite matrix. We denote as

$$
\langle x, y \rangle_A = \langle x, Ay \rangle = \sum_{i,j=1}^{d} A_{ij}x_iy_j, \quad \| x \|_A = \sqrt{\langle x, x \rangle_A}
$$

Note that the standard Euclidean inner product becomes $\langle x, y \rangle = \sum_{i} x_iy_i = \langle x, y \rangle_1$. While we use here the general notation for matrices for comparison to the literature. All positive definite matrices $A$ in this paper will be diagonal matrices, so that the computational effort for computing inner products and norms is still linear in $d$. The Cauchy-Schwarz inequality becomes, $\langle x, y \rangle_A \leq \|x\|_A \|y\|_A$. We further introduce the element-wise product $a \odot b$ of two vectors. Let $a, b \in \mathbb{R}^d$, then $(a \odot b)_i = a_ib_i$ for $i = 1, \ldots , d$.

Let $A \in \mathbb{R}^{d \times d}$ be a symmetric, positive definite matrix, $z \in \mathbb{R}^d$ and $C \subset \mathbb{R}^d$ a convex set. Then we define the weighted projection $P^A_C(z)$ of $z$ onto the set $C$ as

$$
P^A_C(z) = \arg \min_{x \in C} \| x - z \|_A^2 . \tag{1}
$$

It is well-known that the weighted projection is unique and non-expansive.

Lemma 2.1 Let $A \in \mathbb{R}^{d \times d}$ be a symmetric, positive definite matrix and $C \subset \mathbb{R}^d$ be a convex set. Then

$$
\| P^A_C(z) - P^A_C(y) \|_A \leq \| z - y \|_A.
$$

Proof: The first order optimality condition for the weighted projection in (1) is given as

$$
A(x - z) \in N_C(x),
$$

where $N_C(x)$ denotes the normal cone of $C$ at $x$. This can be rewritten as

$$
\langle z - x, y - x \rangle_A \leq 0 \quad \forall y \in C.
$$

This yields

$$
\langle z - P^A_C(z), P^A_C(y) - P^A_C(z) \rangle_A \leq 0,
$$

$$
\langle y - P^A_C(y), P^A_C(z) - P^A_C(y) \rangle_A \leq 0.
$$

Adding these two inequalities yields

$$
\langle z - P^A_C(z) - y + P^A_C(y), P^A_C(y) - P^A_C(z) \rangle_A \leq 0
$$

$$
\implies \| P^A_C(y) - P^A_C(z) \|_A^2 \leq \langle z - y, P^A_C(y) - P^A_C(z) \rangle_A.
$$

The result follows from the application of the weighted Cauchy-Schwarz inequality. \qed

Lemma 2.2 For any symmetric, positive semi-definite matrix $A \in \mathbb{R}^{d \times d}$ we have

$$
\langle x, Ax \rangle \leq \lambda_{\max}(A) \langle x, x \rangle \leq \text{tr}(A) \langle x, x \rangle \tag{2}
$$

where $\lambda_{\max}(A)$ is the maximum eigenvalue of matrix $A$ and $\text{tr}(A)$ denotes the trace of matrix $A$.

2.2. Problem Statement

In this paper we analyze the online convex optimization setting, that is we have a convex set $C$ and at each round we get access to a (sub)-gradient of some continuous convex function $f_t : C \to \mathbb{R}$. At the $t$-th iterate we predict $\theta_t \in C$ and suffer a loss $f_t(\theta_t)$. The goal is to perform well with respect to the optimal decision in hindsight defined as

$$
\theta^* = \arg \min_{\theta \in C} \sum_{t=1}^{T} f_t(\theta).
$$

The adversarial regret at time $T \in \mathbb{N}$ is then given as

$$
R(T) = \sum_{t=1}^{T} (f_t(\theta_t) - f_t(\theta^*)).
$$

We assume that the adversarial can choose from the class of convex functions on $C$, for some parts we will specialize to the set of strongly convex functions.

Definition 2.1 Let $C$ be a convex set. We say that a function $f : C \to \mathbb{R}$ is $\mu$-strongly convex, if there exists $\mu_i > 0$ for $i = 1, \ldots , d$ such that for all $x, y \in C$,

$$
f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \| y - x \|^2_{\text{diag}(\mu)}
$$

$$
= f(x) + \langle \nabla f(x), y - x \rangle + \sum_{i=1}^{d} \mu_i (y_i - x_i)^2 .
$$

Let $\zeta = \min_{i=1, \ldots , d} \mu_i$, then this function is $\zeta$-strongly convex (in the usual sense), that is $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \zeta \| x - y \|^2$.

Note that the difference between our notion of component-wise strong convexity and the usual definition of strong convexity is indicated by the bold font versus normal font. We have two assumptions:

- **A1**: It holds $\sup_{t \geq 1} \| g_t \|_2 \leq G$ which implies the existence of a constant $G_\infty$ such that $\sup_{t \geq 1} \| g_t \|_\infty \leq G_\infty$.
- **A2**: It holds $\sup_{t \geq 1} \| \theta_t - \theta^* \|_2 \leq D$ which implies the existence of a constant $D_\infty$ such that $\sup_{t \geq 1} \| \theta_t - \theta^* \|_\infty \leq D_\infty$. 
Algorithm 1 Adagrad

Input: $\theta_1 \in C, \delta > 0, v_0 = 0 \in \mathbb{R}^d$
for $t = 1$ to $T$ do
  $g_t \in \partial f_t(\theta_t)$
  $v_t = v_{t-1} + (g_t \odot g_t)$
  $A_t = \text{diag}(\sqrt{T}) + \delta I$
  $\theta_{t+1} = P_{C}(\theta_t - \alpha A_t^{-1} g_t)$
end for

One of the first methods which achieves the optimal regret bound of $O(\sqrt{T})$ for convex problems is online projected gradient descent (Zinkevich, 2003), defined as

$$\theta_{t+1} = P_C(\theta_t - \alpha g_t)$$

(3)

where $\alpha_t = \frac{\alpha}{\sqrt{t}}$ is the step-size scheme and $g_t$ is a (sub)-gradient of $f_t$ at $\theta_t$. With $\alpha = \frac{\gamma}{t}$, online projected gradient descent method achieves the optimal $O(\log(T))$ regret bound for strongly-convex problems (Hazan et al., 2007). We consider Adagrad in the next subsection which is one of the popular adaptive alternative to online projected gradient descent.

2.3. Adagrad for convex problems

In this section we briefly recall the main result for the Ada-grad. The algorithm for Adagrad is given in Algorithm 1. If the adversarial is allowed to choose from the set of all possible convex functions on $C \subset \mathbb{R}^d$, then Adagrad achieves the regret bound of order $O(\sqrt{T})$ as shown in (Duchi et al., 2011). This regret bound is known to be optimal for this class, see e.g. (Hazan, 2016). For better comparison to our results for RMSProp, we recall the result from (Duchi et al., 2011) in our notation. For this purpose, we introduce the notation, $g_{T,i} = (g_{1,i}, g_{2,i}, ..., g_{T-1,i})^T$, where $g_{t,i}$ is the $i$-th component of the gradient $g_t \in \mathbb{R}^d$ of the function $f_t$ evaluated at $\theta_t$.

Theorem 2.1 (Duchi et al., 2011) Let Assumptions A1, A2 hold and let $\theta_t$ be the sequence generated by Adagrad in Algorithm 1, where $g_t \in \partial f_t(\theta_t)$ and $f_t : C \rightarrow \mathbb{R}$ is an arbitrary convex function, then for stepsize $\alpha > 0$ the regret is upper bounded as

$$R(T) \leq D_2^2 \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \alpha \sum_{i=1}^{d} \|g_{1:T,i}\|_2.$$ 

The effective step-length of Adagrad is on the order of $\frac{\alpha}{\sqrt{T}}$. This can be seen as follows; first note that $v_{T,i} = \sum_{t=1}^{T} g_{t,i}^2$ and thus $(A_T^{-1})_{ii}$ is a diagonal matrix with entries $\frac{\alpha}{\sqrt{v_{T,i} + \delta}}$. Then one has

$$\alpha (A_T^{-1})_{ii} = \frac{\alpha}{\sqrt{\sum_{i=1}^{T} g_{t,i}^2 + \delta}}$$

(4)

Thus an alternative point of view of Adagrad is, that it has a decaying stepsize $\frac{\alpha}{\sqrt{T}}$ but now the correction term becomes the running average of the squared derivatives plus a vanishing damping term. However, the effective stepsize has to decay faster to get a logarithmic regret bound for the strongly convex case. This is what we analyze in the next section, where we propose SC-Adagrad for strongly convex functions.

3. Strongly convex Adagrad (SC-Adagrad)

The modification SC-Adagrad of Adagrad which we propose in the following can be motivated by the observation that the online projected gradient descent (Hazan et al., 2007) uses stepsizes of order $\alpha = O(\frac{1}{\sqrt{t}})$ in order to achieve the logarithmic regret bound for strongly convex functions. In analogy with the derivation in the previous section, we still have $v_{T,i} = \sum_{t=1}^{T} g_{t,i}^2$. But now we modify $(A_T^{-1})_{ii}$ and set it as a diagonal matrix with entries $\frac{1}{v_{T,i} + \delta_t}$. Then one has

$$\alpha (A_T^{-1})_{ii} = \frac{\alpha}{\sum_{i=1}^{T} g_{t,i}^2 + \delta_t} = \frac{\alpha}{1 + \sum_{i=1}^{T} \frac{g_{t,i}^2}{\sqrt{v_{T,i} + \delta_t}}}.$$ 

(5)

Again, we have in the denominator a running average of the observed gradients and a decaying damping factor. In this way, we get an effective stepsize of order $O(\sqrt{\frac{1}{T}})$ in SC-Adagrad. The formal method is presented in Algorithm 2.

Algorithm 2 SC-Adagrad

Input: $\theta_1 \in C, \delta_0 > 0, v_0 = 0 \in \mathbb{R}^d$
for $t = 1$ to $T$ do
  $g_t \in \partial f_t(\theta_t)$
  $v_t = v_{t-1} + (g_t \odot g_t)$
  Choose $0 < \delta_t \leq \delta_{t-1}$ element wise
  $A_t = \text{diag}(v_t) + \text{diag}(\delta_t)$
  $\theta_{t+1} = P_{C}(\theta_t - \alpha A_t^{-1} g_t)$
end for

also that we have defined the damping factor $\delta_t$ as a function of $t$ which is also different from standard Adagrad. The constant $\delta$ in Adagrad is mainly introduced due to numerical reasons in order to avoid problems when $g_{t,i}$ is very small for some components in the first iterations and is typically chosen quite small e.g. $\delta = 10^{-8}$. For SC-Adagrad the situation is different. If the first components $g_{1,i}, g_{2,i}, \ldots$ are very small, say of order $\epsilon$, then the update is $\frac{\alpha}{\sqrt{v_{T,i} + \delta_t}}$ which can become extremely large if $\delta_t$ is chosen.
to be small. This would make the method very unstable and would lead to huge constants in the bounds. This is probably why in (Ruder, 2016), the modification of Ada

\[ A \]

for any two matrices \( A, B \) than data-independent bounds.

\[ \text{Proof:} \]

Let Assumptions Lemma 3.2 this choice after the proof. In the following we provide the probably why in (Ruder, 2016), the modification of Ada

\[ A \]

\[ \sum_{i=1}^{T} \sum_{t=2}^{T} \delta_{t,i} - \delta_{t-1,i} \]

\[ \leq \sum_{i=1}^{d} \log \left( \frac{\|g_{1:T,i}\|^2 + \delta_{T,i}}{\delta_{i,i}} \right) - \sum_{i=1}^{d} \sum_{t=2}^{T} \frac{\delta_{t,i} - \delta_{t-1,i}}{\delta_{i,i}} \]

In the first step we use \( \langle x, Ax \rangle = A \bullet \text{diag}(xx^T) \) where \( A \) is a diagonal matrix and subsequently we use \( \forall t > 1 \), \( \text{diag}(g_{t}g_{t}^T) = A_{t} - A_{t-1} - \text{diag}(\delta_{t}) + \text{diag}(\delta_{t-1}) \), and for \( t = 1 \) we have \( \text{diag}(g_{1}g_{1}^T) = A_{1} - \text{diag}(\delta_{1}) \). In the first inequality we use Lemma 3.1 also see for Lemma 12 of (Hazan et al., 2007). Note that for \( T = 1 \), the upper bound results in 0.

\[ \text{Theorem 3.1} \]

Let Assumptions A1, A2 hold and let \( \theta_{t} \) be the sequence generated by the SC-Adagrad in Algorithm 2, where \( g_{t} \in \partial f_{t}(\theta_{t}) \) and \( f_{t} : C \to \mathbb{R} \) is an arbitrary \( \mu \)-strongly convex function (\( \mu \in \mathbb{R}^{+} \)) where the stepsize fulfills \( \alpha \geq \max_{i=1,\ldots,d} \frac{g_{t}^2}{2\mu_{i}} \). Furthermore, let \( \delta_{t} > 0 \) and \( \delta_{t,i} \leq \delta_{t-1,i} \forall t \in [T], \forall i \in [d] \), then the regret of SC-Adagrad can be upper bounded for \( T \geq 1 \) as

\[ R(T) \leq \frac{D_{\infty}^2 d \text{tr}(\text{diag}(\delta_{1}))}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^{d} \log \left( \frac{\|g_{1:T,i}\|^2 + \delta_{T,i}}{\delta_{i,i}} \right) \]

\[ + \frac{1}{2} \sum_{i=1}^{d} \inf_{t \in [T]} \left( \frac{(\theta_{i} - \theta_{t})^2}{\alpha} - \frac{\alpha}{\|g_{1:T,i}\|^2 + \delta_{i,i}} \right) (\delta_{T,i} - \delta_{i,i}) \]

For constant \( \delta_{t} \) i.e \( \delta_{t,i} = \delta > 0 \forall t \in [T] \) and \( \forall i \in [d] \) then the regret of SC-Adagrad is upper bounded as

\[ R(T) \leq \frac{D_{\infty}^2 d \delta}{2\alpha} + \frac{\alpha}{2} \sum_{i=1}^{d} \log \left( \frac{\|g_{1:T,i}\|^2 + \delta}{\delta_{i,i}} \right) \]

For \( \zeta \)-strongly convex function choosing \( \alpha \geq \frac{g_{t}^2}{2\zeta} \) we obtain the above mentioned regret bounds.

\[ \text{Proof:} \]

We rewrite the regret bound with the definition of \( \mu \)-strongly convex functions as

\[ R(T) = \sum_{t=1}^{T} (f_{t}(\theta_{t}) - f_{t}(\theta^{*})) \]
\[
\leq \sum_{t=1}^{T} \langle g_t, \theta_t - \theta^* \rangle - \sum_{t=1}^{T} \|\theta_t - \theta^*\|_{\text{diag}(\mu)}^2
\]

Using the non-expansiveness we have
\[
\|\theta_{t+1} - \theta^*\|_{A_t}^2
= \|P_{C_t}(\theta_t - \alpha A_t^{-1} g_t) - \theta^*\|_{A_t}^2
\leq \|\theta_t - \alpha A_t^{-1} g_t - \theta^*\|_{A_t}^2
\leq \|\theta_t - \theta^*\|_{A_t}^2 - 2\alpha \langle g_t, A_t^{-1} g_t \rangle + \alpha^2 \langle g_t, A_t^{-1} g_t \rangle
\]

This yields
\[
\langle g_t, \theta_t - \theta^* \rangle
\leq \frac{\|\theta_t - \theta^*\|_{A_t}^2 - \|\theta_{t+1} - \theta^*\|_{A_{t+1}}^2}{2\alpha} + \frac{\alpha}{2} \langle g_t, A_t^{-1} g_t \rangle
\]

Hence we can upper bound the regret as follows
\[
R(T)
\leq \sum_{t=1}^{T} \frac{\|\theta_t - \theta^*\|_{A_t}^2 - \|\theta_{t+1} - \theta^*\|_{A_{t+1}}^2}{2\alpha}
+ \frac{\alpha}{2} \sum_{t=1}^{T} \langle g_t, A_t^{-1} g_t \rangle
+ \frac{\alpha}{2} \sum_{t=1}^{T} \|\theta_t - \theta^*\|_{A_t}^2 - \|\theta_{t+1} - \theta^*\|_{A_{t+1}}^2
\leq \frac{\|\theta_1 - \theta^*\|_{A_1}^2}{2\alpha}
+ \frac{\alpha}{2} \sum_{t=1}^{T} \langle g_t, A_t^{-1} g_t \rangle
\]

In the last step we use the equality \(\forall x \in \mathbb{R}^n \|x\|_{A^{-1}}^2 = \|x\|_{A-B}^2\) where \(A, B \in \mathbb{R}^{n \times n}\) and both are diagonal matrices. Now, we choose \(\alpha\) such that \(A_t - A_{t-1} - 2\alpha \text{diag}(\mu) \preceq \text{diag}(\delta_t) - \text{diag}(\delta_{t-1}) \) \(\forall t \geq 2\) and \(A_1 - 2\alpha \text{diag}(\mu) \preceq \text{diag}(\mu)\) since \(A_t - A_{t-1} \preceq G_2^2 I + \text{diag}(\delta_t) - \text{diag}(\delta_{t-1})\) and \(A_1 \preceq G_2^2 I + \text{diag}(\delta_1)\) because at any round the difference between subsequent squares of sub-gradients is bounded by \(G_2^2\). Also by Algorithm 2, \(\delta_{t,i} \leq \delta_{t-1,i} \forall t > 1, \forall i \in [d]\) hence \(\text{diag}(\delta_t) - \text{diag}(\delta_{t-1}) \preceq 0\) Hence by choosing \(\alpha \geq \max_{\delta_{t,i}} G_2^2 \) we have \(A_t - A_{t-1} - 2\alpha \text{diag}(\mu) \preceq \text{diag}(\delta_t) - \text{diag}(\delta_{t-1}) \) \(\forall t \geq 2\) and \(A_1 - 2\alpha \text{diag}(\mu) \preceq \text{diag}(\delta_1)\) which yields
\[
R(T)
\leq \frac{\|\theta_1 - \theta^*\|_{\text{diag}(\mu)}^2}{2\alpha}
+ \frac{\alpha}{2} \sum_{t=1}^{T} \langle g_t, A_t^{-1} g_t \rangle
+ \frac{\alpha}{2} \sum_{t=1}^{T} \langle g_t, A_t^{-1} g_t \rangle
\]

In the second inequality we bounded \(\|\theta_1 - \theta^*\|_{\text{diag}(\mu)}^2 \leq D_\infty^2 \text{tr}(\text{diag}(\delta_1))\). In the second last step we use the Lemma 3.2. So under a constant \(\delta_t\) i.e \(\delta_{t,i} = \delta > 0, \forall t \in [T], \forall i \in [d]\) we have \(\text{tr}(\text{diag}(\delta_t)) = d\delta\) hence proving the result (7). For \(\zeta\)-strongly convex functions choosing \(\alpha \geq G_2^2 \frac{1}{2\zeta}\) we obtain the the same results as \(\mu\)-strongly convex functions. This can be seen by setting \(\mu_t = \zeta, \forall t \in [d]\). □

Note that the first and the last term in the regret bound can be upper bounded by constants. Only the second term depends on \(T\). Note that \(\|g_{1:T}\|_{T}^2 \leq TC^2\) and as \(\delta_t\) is monotonically decreasing, the second term is on the order of \(O(\log(T))\) and thus we have a logarithmic regret bound. As the bound is data-dependent, in the sense that it depends on the observed sequence of gradients, it is much tighter than a data-independent bound.

The bound includes also the case of a non-decaying damping factor \(\delta_t = \delta = \xi_2 (\xi_1 = 0)\). While a rather large constant damping factor can work well, we have noticed that the best results are obtained with the decay scheme
\[
\delta_{t,i} = \xi_2 e^{-\xi_1 t,i}, \quad i = 1, \ldots, d
\]
where \(\xi_1 > 0, \xi_2 > 0\), which is what we use in the experiments. Note that this decay scheme for \(\xi_1, \xi_2 > 0\) is
Variants of RMSProp and Adagrad with Logarithmic Regret Bounds

adaptive to the specific dimension and thus increases the adaptivity of the overall algorithm. For completeness we also give the bound specialized for this decay scheme.

**Corollary 3.1** In the setting of Theorem 3.1 choose \( \delta_{t,i} = \xi_2 e^{-\xi_1 v_{j,i}} \) for \( i = 1, \ldots, d \) for some \( \xi_1 > 0, \xi_2 > 0 \). Then the regret of SC-Adagrad can be upper bounded for \( T \geq 1 \) as

\[
R(T) \leq \frac{d D^2 \xi_2}{2\alpha} + \frac{\alpha}{2} \log(\xi_2 e^{-\xi_1 \|g_1:1:T\|^2})
+ \frac{\alpha}{2} \sum_{i=1}^d \log \left( \|g_1:T,i\|^2 + \xi_2 e^{-\xi_1 \|g_1:T,i\|^2} \right)
+ \frac{\alpha \xi_2}{2} \left( \log(\xi_2) + 1 \right) \sum_{i=1}^d \left( 1 - e^{-\xi_1 \|g_1:T,i\|^2} \right)
\]

**Proof:** Note that \( \delta_{t,i} = \xi_2 e^{-\xi_1 v_{j,i}} = \xi_2 e^{-\xi_1 v_{j,i}} \). Plugging this into Theorem 3.1 for \( \xi_1, \xi_2 > 0 \) yields the results for the first three terms. Using \( (\theta_{t,i} - \theta_{t,i}^*)^2 \geq 0 \) we have

\[
\inf_{t \in [T]} \left( \frac{(\theta_{t,i} - \theta_{t,i}^*)^2}{\alpha} - \frac{\alpha}{\|g_1:T,i\|^2 + \delta_{t,i}} \right)
\geq \inf_{t \in [1:T]} \|g_1:j,i\|^2 + \delta_{j,i}
\]

Note that \( \|g_1:j,i\|^2 + \delta_{j,i} = v_{j,i} + \xi_2 e^{-\xi_1 v_{j,i}} \), in order to find the minimum of this term we thus analyze the function \( f(x) = x + \xi_2 e^{-\xi_1 x} \), and a straightforward calculation shows that the minimum is attained at \( x^* = \frac{1}{\xi_1} \log(\xi_1 \xi_2) \) and \( f(x^*) = \frac{1}{\xi_1} \log(\xi_1 \xi_2) + 1 \). This yields the fourth term. \( \square \)

Unfortunately, it is not obvious that the regret bound for our decaying damping factor is better than the one of a constant damping factor. Note, however that the third term in the regret bound of Theorem 3.1 can be negative. It thus remains an interesting question for future work, if there exists an optimal decay scheme which provably works better than any constant one.

**4. RMSProp and SC-RMSProp**

RMSProp is one of the most popular adaptive gradient algorithms used for the training of deep neural networks (Schaul et al., 2014; Dauphin et al., 2015; Daniel et al., 2016; Schmidhuber, 2015). It has been used frequently in computer vision (Karpathy & Fei-Fei, 2016) e.g. to train the latest InceptionV4 network (Szegedy et al., 2016a,b). Note that RMSProp outperformed other adaptive methods like Adagrad order Adadelta as well as SGD with momentum in a large number of tests in (Schaul et al., 2014). It has been argued that if the changes in the parameter update are approximately Gaussian distributed, then the matrix \( A_t \) can be seen as a preconditioner which approximates the diagonal of the Hessian (Daniel et al., 2016). However, it is far to say that despite its huge empirical success in practice and some first analysis in the literature, there is so far no rigorous theoretical analysis of RMSProp. We will analyze RMSProp given in Algorithm 3 in the framework of online convex optimization.

**Algorithm 3 RMSProp**

**Input:** \( \theta_1 \in C, \delta > 0, \alpha > 0, v_0 = 0 \in \mathbb{R}^d \)

for \( t = 1 \) to \( T \) do

\( g_t \in \partial f_t(\theta_t) \)

\( v_t = \beta t v_{t-1} + (1 - \beta t)(g_t \odot g_t) \)

Set \( \epsilon_t = \frac{\delta}{\sqrt{t}} \) and \( \alpha_t = \frac{\alpha}{\sqrt{T}} \)

\( A_t = \text{diag} \left( \sqrt{\epsilon_t} \right) + \epsilon_t I \)

\( \theta_{t+1} = P_{\mathbb{S}^d} \left( \theta_t - \alpha_t A_t^{-1} g_t \right) \)

end for

First, we will show that RMSProp reduces to Adagrad for a certain choice of its parameters. Second, we will prove for the general convex case a regret bound of \( O(\sqrt{T}) \) similar to the bound given in Theorem 2.1. It turns out that the convergence analysis requires that in the update of the weighted cumulative squared gradients \( (v_{t,i}) \), it has to hold

\[ 1 - \frac{1}{t} \leq \beta_t \leq 1 - \gamma \frac{1}{t} \]

for some \( 0 < \gamma \leq 1 \). This is in contrast to the original suggestion of (Hinton et al., 2012) to choose \( \beta_t = 0.9 \). It will turn out later in the experiments that the constant choice of \( \beta_t \) leads sometimes to divergence of the sequence, whereas the choice derived from our theoretical analysis always leads to a convergent scheme even when applied to deep neural networks. Thus we think that the analysis in the following is not only interesting for the convex case but can give valuable hints how the parameters of RMSProp should be chosen in deep learning.

Before we start the regret analysis we want to discuss the sequence \( v_t \) in more detail. Using the recursive definition of \( v_t \), we get the closed form expression

\[ v_{t,i} = \sum_{j=1}^t (1 - \beta_j) \prod_{k=j+1}^t \beta_k g_{j,i}^2. \]

With \( \beta_t = 1 - \frac{1}{t} \) one gets, \( v_{t,i} = \sum_{j=1}^t \frac{1}{j} \prod_{k=j+1}^t \frac{k-1}{k} g_{j,i}^2 \), and using the telescoping product ones gets \( \prod_{k=j+1}^t \frac{k-1}{k} = \frac{1}{t} \) and thus

\[ v_{t,i} = \frac{1}{t} \sum_{j=1}^t g_{j,i}^2. \]

If one uses additionally the stepsize scheme \( \alpha_t = \frac{\alpha}{\sqrt{t}} \) and \( \epsilon_t = \frac{\alpha}{\sqrt{T}} \), then we recover the update scheme of Adagrad,
see (4), as a particular case of RMSProp. We are not aware of that this correspondence of Adagrad and RMSProp has been observed before.

The proof of the regret bound for RMSProp relies on the following lemma.

**Lemma 4.1** Let Assumptions A1 and A2 and suppose that $1 - \frac{t}{T} \leq \beta_t \leq 1 - \frac{\gamma}{T}$ for some $0 < \gamma \leq 1$, and $t \geq 1$. Also for $t > 1$ suppose $\sqrt{(t-1)\epsilon_{t-1}} \leq \sqrt{t}\epsilon_t$, then

$$\sum_{t=1}^T \frac{g_{t,i}^2}{\sqrt{v_{t,i}} + \sqrt{t}\epsilon_t} \leq \frac{2(2 - \gamma)}{\gamma} \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right).$$

**Proof:** The lemma is proven via induction. For $T = 1$ we have $v_0 = 0$ and thus $v_{1,i} = (1 - \beta_1)g_{1,i}^2$ and thus

$$\frac{g_{1,i}^2}{\sqrt{v_{1,i}} + \epsilon_i} = \frac{(1 - \beta_1)g_{1,i}^2}{(1 - \beta_1)\sqrt{(1 - \beta_1)g_{1,i}^2 + \epsilon_1}} \leq \frac{\sqrt{(1 - \beta_1)g_{1,i}^2 + \epsilon_1}}{1 - \beta_1} \leq \frac{\sqrt{v_{1,i} + \epsilon_1}}{1 - \beta_1}.$$

Note that $\frac{1}{1 - \beta_1} \leq \frac{2(2 - \gamma)}{\gamma}$ since $2(2 - \gamma) > 1$ for $\gamma \leq 1$ hence the bound holds for $T = 1$. For $T > 1$ we suppose that the bound is true for $T - 1$ and get

$$\sum_{t=1}^{T-1} \frac{g_{t,i}^2}{\sqrt{v_{t,i}} + \sqrt{t}\epsilon_t} \leq \frac{2(2 - \gamma)}{\gamma} \left( \sqrt{T-1}v_{T-1,i} + \sqrt{(T-1)\epsilon_{T-1}} \right).$$

We rewrite $v_{T,i} = \beta_T v_{T-1,i} + (1 - \beta_T)g_{T,i}^2$, $v_{T-1,i} = \frac{1}{\beta_T} v_{T-1,i} - \frac{1 - \beta_T}{\beta_T} g_{T,i}^2$, and with $\sqrt{(t-1)\epsilon_{t-1}} \leq \sqrt{t}\epsilon_t$ we get

$$\sqrt{(T-1)v_{T-1,i} + \sqrt{(T-1)\epsilon_{T-1}}} \leq \sqrt{T}v_{T,i} \leq \frac{T - 1}{\beta_T} v_{T,i} - \frac{(T - 1)(1 - \beta_T)}{\beta_T} g_{T,i}^2 + \sqrt{T}\epsilon_T$$

$$\leq \frac{T - 1}{\beta_T} v_{T,i} - \frac{(1 - \beta_T)}{\beta_T} g_{T,i}^2 + \sqrt{T}\epsilon_T$$

$$\leq \frac{T - 1}{\beta_T} v_{T,i} - \frac{(T - 1)(1 - \beta_T)}{\beta_T} g_{T,i}^2 + \sqrt{T}\epsilon_T.$$

Note that in the last step we have used that $\frac{T - 1}{\beta_T} \leq 1$ and the fact that $\sqrt{x}$ is concave and thus $\sqrt{x - c} \leq \sqrt{x - \frac{c}{\sqrt{x}}}$ given that $x - c \geq 0$, along with $\frac{-1}{\sqrt{v_{T,i} + \sqrt{t}\epsilon_t}} \leq \frac{-1}{\sqrt{Tv_{T,i} + \sqrt{T}\epsilon_T}}$ for $v_{t,i} \neq 0$ we have

$$\sqrt{(T-1)v_{T-1,i} + \sqrt{(T-1)\epsilon_{T-1}}} \leq \sqrt{T}\epsilon_T - \frac{(T-1)(1 - \beta_T)}{2\beta_T \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)} g_{T,i}^2.$$

Using the above bound we have the following

$$\sum_{t=1}^T \frac{g_{t,i}^2}{\sqrt{v_{t,i}} + \sqrt{T}\epsilon_t} \leq \frac{2(2 - \gamma)}{\gamma} \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

$$+ \frac{g_{T,i}^2}{\sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T} \leq \frac{2(2 - \gamma)}{\gamma} \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

$$+ \frac{(T - 1)(1 - \beta_T)}{2\beta_T \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)} g_{T,i}^2 \leq \frac{2(2 - \gamma)}{\gamma} \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

$$+ \frac{(T - 1)(1 - \beta_T)}{2\beta_T \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)} g_{T,i}^2.$$

In the last step we use (8) and since for $T > 1$ the term $\left(1 - \frac{2(2 - \gamma)}{\gamma} \frac{(T-1)(1 - \beta_T)}{2\beta_T} \right) \leq 0$ for $1 - \frac{\gamma}{T} \leq \beta_t \leq 1 - \frac{\gamma}{T}$ prove the bound.

**Corollary 4.1** Let Assumptions A1, A2 hold and suppose that $1 - \frac{t}{T} \leq \beta_t \leq 1 - \frac{\gamma}{T}$ for some $0 < \gamma \leq 1$, and $t \geq 1$. Also for $t > 1$ suppose $\sqrt{(t-1)\epsilon_{t-1}} \leq \sqrt{t}\epsilon_t$, and set $\alpha_t = \frac{\alpha}{\sqrt{T}}$, then

$$\sum_{t=1}^T \frac{\alpha_t}{\sqrt{T}} \langle g_t, A_t^{-1} g_t \rangle \leq \frac{\alpha(2 - \gamma)}{T} \sum_{i=1}^d \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

**Proof:** Using the definition of $A_t = diag(\sqrt{v_t}) + \epsilon_t I$, $\alpha_t = \frac{\alpha}{\sqrt{T}}$ along with Lemma 4.1 we get

$$\sum_{t=1}^T \frac{\alpha_t}{\sqrt{T}} \langle g_t, A_t^{-1} g_t \rangle = \sum_{t=1}^T \frac{\alpha_t}{\sqrt{T}} \sum_{i=1}^d \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

$$\leq \frac{\alpha}{2} \sum_{t=1}^T \sum_{i=1}^d \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

$$\leq \frac{\alpha}{2} \sum_{i=1}^d \frac{2(2 - \gamma)}{\gamma} \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)$$

With the help of Lemma 4.1 and Corollary 4.1 we can now state the regret bound for RMSProp.

**Theorem 4.1** Let Assumptions A1, A2 hold and let $\theta_t$ be the sequence generated by RMSProp in Algorithm 3, where $g_t \in \partial f_t(\theta_t)$ and $f_t : C \to \mathbb{R}$ is an arbitrary convex function and $\alpha_t = \frac{\alpha}{\sqrt{T}}$ for some $\alpha > 0$ and $1 - \frac{1}{T} \leq \beta_t \leq 1 - \frac{\gamma}{T}$. The regret of the algorithm satisfies

$$R_T \leq \sqrt{Tv_{T,i} + \sqrt{T}\epsilon_T} - \frac{(T-1)(1 - \beta_T)}{2\beta_T \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_T \right)} g_{T,i}^2.$$
for some $0 < \gamma \leq 1$. Also for $t > 1$ let $\sqrt{(t-1)e_{t-1}} \leq \sqrt{T}e_t$, then the regret of RMSProp can be upper bounded for $T \geq 1$ as

$$R(T) \leq \frac{(D^2_{\infty} + \alpha(2 - \gamma))}{2\alpha} \sum_{i=1}^{d} (\sqrt{Tv_{t,i}} + \sqrt{T}\epsilon_{t})$$

**Proof:** Note that for every convex function $f_t : C \rightarrow \mathbb{R}$ it holds for all $x, y \in C$ and $g_t \in \partial f_t(x)$,

$$f_t(y) \geq f_t(x) + \langle g_t, y - x \rangle.$$

We use this to upper bound the regret as

$$R(T) = \sum_{t=1}^{T} (f_t(\theta_t) - f_t(\theta^*)) \leq \sum_{t=1}^{T} \langle g_t, \theta_t - \theta^* \rangle$$

Using the non-expansiveness of the weighted projection, we have

$$\|\theta_{t+1} - \theta^*\|_{A_t}^2
\leq \|\theta_t - \alpha_t A_t^{-1} g_t - \theta^*\|_{A_t}^2
\leq \|\theta_t - \alpha_t A_t^{-1} g_t - \theta^*\|_{A_t}^2
\leq \|\theta_t - \theta^*\|_{A_t}^2 - 2\alpha_t \langle g_t, \theta_t - \theta^* \rangle + \alpha_t^2 \langle g_t, A_t^{-1} g_t \rangle$$

This yields

$$\langle g_t, \theta_t - \theta^* \rangle \leq \frac{\|\theta_t - \theta^*\|_{A_t}^2 - \|\theta_{t+1} - \theta^*\|_{A_t}^2}{2\alpha_t} + \frac{\alpha_t}{2} \langle g_t, A_t^{-1} g_t \rangle$$

Hence we can upper bound the regret as follows

$$R(T) \leq \sum_{t=1}^{T} \frac{\|\theta_t - \theta^*\|_{A_t}^2 - \|\theta_{t+1} - \theta^*\|_{A_t}^2}{2\alpha_t}
+ \sum_{t=2}^{T} \frac{\alpha_t}{2} \langle g_t, A_t^{-1} g_t \rangle
\leq \frac{\|\theta_1 - \theta^*\|_{A_1}^2}{2\alpha_1} + \sum_{t=2}^{T} \left( \frac{\|\theta_t - \theta^*\|_{A_t}^2}{2\alpha_t} - \frac{\|\theta_t - \theta^*\|_{A_{t-1}}^2}{2\alpha_{t-1}} \right)
- \frac{\|\theta_{T+1} - \theta^*\|_{A_T}^2}{2\alpha_T}
+ \sum_{t=1}^{T} \frac{\alpha_t}{2} \langle g_t, A_t^{-1} g_t \rangle
\leq \frac{\|\theta_1 - \theta^*\|_{A_1}^2}{2\alpha_1}
+ \sum_{t=2}^{T} \frac{\|\theta_t - \theta^*\|_{A_t}^2}{2\alpha_t} \sqrt{T_{A_t} - \sqrt{T_{A_{t-1}}}}
+ \sum_{t=1}^{T} \frac{\alpha_t}{2} \langle g_t, A_t^{-1} g_t \rangle$$

In the last step we used $\alpha_t = \frac{\alpha}{\sqrt{t}}$. We show that

$$\sqrt{T_{A_t} - \sqrt{T_{A_{t-1}}}} \geq 0 \forall t \in [T] \quad (9)$$

Note that $A_t$ are diagonal matrices for all $t \geq 1$. We note that

$$(A_t)_{ii} = \sqrt{v_{t,i} + \epsilon_t},$$
and $v_{t,i} = \beta_t v_{t-1,i} + (1 - \beta_t) g_t^2$ as well as $\beta_t \geq 1 - \frac{1}{\gamma}$
which implies $t\beta_t \geq t - 1$. We get

$$\sqrt{T}(A_t)_{ii} = \sqrt{tv_{t,i} + T\epsilon_t}
= \sqrt{t\beta_t v_{t-1,i} + t(1 - \beta_t) g_t^2 + \sqrt{T}\epsilon_t}
\geq \sqrt{(t-1)v_{t-1,i} + \sqrt{T-1}\epsilon_{t-1}},$$

where we used in the last inequality that $\sqrt{T}\epsilon_t \geq \sqrt{T-1}\epsilon_{t-1}$. Note that

$$\sum_{t=2}^{T} \|\theta_t - \theta^*\|_{A_t}^2
\leq \sum_{t=2}^{T} \sum_{i=1}^{d} \langle \theta_{t,i} - \theta^* \rangle^2
\leq \sum_{i=1}^{d} D_{\infty}^2
\sum_{t=2}^{T} \left( \sqrt{tv_{t,i} + T\epsilon_t} - \sqrt{(t-1)v_{t-1,i} + T-1\epsilon_{t-1}} \right)
= \sum_{i=1}^{d} D_{\infty}^2 \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_{T} - \sqrt{v_{1,i} + \epsilon_1} \right)$$

where the inequality could be done as we showed before that the difference of the terms in $v_{t,i}$ is non-negative for all $i \in [d]$ and $t \geq 1$. As $(A_1)_{ii} = \sqrt{v_{1,i} + \epsilon_1}$ we get

$$\frac{\|\theta_1 - \theta^*\|_{A_1}^2}{2\alpha_1} \leq \frac{D_{\infty}^2}{2\alpha} \sum_{i=1}^{d} \left( \sqrt{v_{1,i} + \epsilon_1} \right).$$

Thus in total we have

$$R(T) \leq \frac{D_{\infty}^2}{2\alpha} \sum_{i=1}^{d} \left( \sqrt{Tv_{T,i}} + \sqrt{T}\epsilon_{T} \right)
+ \sum_{t=1}^{T} \frac{\alpha_t}{2} \langle g_t, A_t^{-1} g_t \rangle.$$
Algorithm 4 SC-RMSProp

\textbf{Input:} \( \theta_1 \in C, \delta_0 = 1, v_0 = 0 \in \mathbb{R}^d \)

\textbf{for} \( t = 1 \) \textbf{to} \( T \) \textbf{do}

\( g_t \in \partial f_t(\theta_t) \)

\( v_t = \beta_t v_{t-1} + (1 - \beta_t)(g_t \odot g_t) \)

Set \( \epsilon_t = \frac{\delta_t}{t} \) where \( \delta_t \leq \delta_{t-i} \) for \( i \in [d] \) and \( \alpha_t = \frac{\delta_t}{t} \)

\( A_t = \text{diag}(v_t + \epsilon_t) \)

\( \theta_{t+1} = P^A_{C}(\theta_t - \alpha_t A_t^{-1} g_t) \)

\textbf{end for}

4.1. SC-RMSProp

Similar to the extension of Adagrad to SC-Adagrad, we present in this section SC-RMSProp which achieves a logarithmic regret bound.

Note that again there exist choices for the parameters of SC-RMSProp such that it reduces to SC-Adagrad. The corresponding is given by the choice

\[
\beta_t = 1 - \frac{1}{t}, \quad \alpha_t = \frac{\alpha}{t}, \quad \epsilon_t = \frac{\delta_t}{t},
\]

for which again it follows \( v_{t,i} = \frac{1}{t} \sum_{j=1}^{t} g_{j,i} \), with the same argument as for RMSProp. Please see Equation (5) for the correspondence. Moreover, with the same argument as for SC-Adagrad we use a decay scheme for the damping factor

\[
\epsilon_{t,i} = \frac{\xi_{1,t} \cdot \epsilon_{t,i}}{t}, \quad i = 1, \ldots, d, \text{ for } \xi_1 \geq 0, \xi_2 > 0
\]

The analysis of SC-RMSProp is along the lines of SC-Adagrad with some overhead due to the structure of \( v_t \).

Lemma 4.2 Let \( \alpha_t = \frac{\alpha}{t}, 1 - \frac{1}{t} \leq \beta_t \leq 1 - \frac{2}{t} \) and \( A_t \) as defined in SC-RMSProp, then it holds for all \( T \geq 1 \),

\[
\sum_{t=1}^{T} \frac{\alpha_t}{2} \left< g_t, A_t^{-1} g_t \right> \leq \frac{\alpha}{2\gamma} \sum_{i=1}^{d} \log \left( \frac{T(v_{t,i} + \epsilon_{t,i})}{\epsilon_{t,i}} \right)
\]

\[
+ \frac{\alpha}{2\gamma} \sum_{t=2}^{T} \sum_{i=1}^{d} \frac{-t\epsilon_{t,i} + (t - 1)\epsilon_{t-1,i}}{tv_{t,i} + t\epsilon_{t,i}}
\]

\[
+ \frac{\alpha}{2\gamma} \sum_{i=1}^{d} \frac{(1 - \gamma)(1 + \log T)}{\text{inf}_{j \in [1,T]} jv_{j,i} + j\epsilon_{j,i}}
\]

Proof: Using \( \langle x, Ax \rangle = A \cdot \text{diag}(xx^T) \) and with \( g_{t,i}^2 = \frac{v_{t,i} - \beta_t v_{t-1,i}}{1 - \beta_t} \) and using that \( A_t \) is diagonal with \( (A_t)_{ii} = v_{t,i} + \epsilon_{t,i} \), we get

\[
\sum_{t=1}^{T} \frac{\alpha_t}{2} \left( (tA_t)^{-1} \cdot \text{diag}(g_t g_t^T) \right)
\]

\[
= \frac{\alpha}{2(1 - \beta_t)} \left( (A_1)^{-1} \cdot (A_1 - \text{diag}(\epsilon_1)) \right)
\]

\[
= \frac{\alpha}{2(1 - \beta_t)} \left( (A_1)^{-1} \cdot (A_1 - \text{diag}(\epsilon_1)) \right)
\]

\[
= \frac{\alpha}{2(1 - \beta_t)} \left( (A_1)^{-1} \cdot (A_1 - \text{diag}(\epsilon_1)) \right)
\]

\[
+ \sum_{t=2}^{T} \frac{\alpha}{2} \left( (tA_t)^{-1} \cdot \left( \frac{\beta_t A_{t-1} - \beta_t \text{diag}(\epsilon_{t-1})}{1 - \beta_t} \right) \right)
\]

\[
\leq \frac{\alpha}{2\gamma} \sum_{t=2}^{T} \left( (A_1)^{-1} \cdot (A_1 - \text{diag}(\epsilon_1)) \right)
\]

\[
+ \sum_{t=2}^{T} \frac{\alpha}{2} \left( (tA_t)^{-1} \cdot \left( \frac{-\text{diag}(\epsilon_t) + \beta_t \text{diag}(\epsilon_{t-1})}{1 - \beta_t} \right) \right)
\]

\[
\leq \frac{\alpha}{2\gamma} \sum_{t=2}^{T} \left( (A_1)^{-1} \cdot (A_1 - \text{diag}(\epsilon_1)) \right)
\]

\[
+ \sum_{t=2}^{T} \frac{\alpha}{2} \left( (tA_t)^{-1} \cdot \left( \frac{-\text{diag}(\epsilon_t) + \beta_t \text{diag}(\epsilon_{t-1})}{1 - \beta_t} \right) \right)
\]

\[
\leq \frac{\alpha}{2\gamma} \sum_{t=2}^{T} \left( (A_1)^{-1} \cdot (A_1 - \text{diag}(\epsilon_1)) \right)
\]

\[
+ \sum_{t=2}^{T} \frac{\alpha}{2} \left( (tA_t)^{-1} \cdot \left( \frac{-\text{diag}(\epsilon_t) + \beta_t \text{diag}(\epsilon_{t-1})}{1 - \beta_t} \right) \right)
\]

In the first inequality we use \( \frac{1}{1 - \beta_t} \leq \frac{\gamma}{t} \). In the last step we use, that \( \forall t > 1, \frac{1}{1 - \beta_t} \leq \frac{1}{t} \). Finally, by upper bounding the last term with Lemma 4.3

\[
\sum_{t=1}^{T} \frac{\alpha_t}{2} \left< g_t, A_t^{-1} g_t \right> \leq \frac{\alpha}{2\gamma} \log \left( \frac{|TA_T|}{|\text{diag}(\epsilon_1)|} \right)
\]

\[
+ \frac{\alpha}{2\gamma} \sum_{t=2}^{T} \sum_{i=1}^{d} \frac{-t\epsilon_{t,i} + (t - 1)\epsilon_{t-1,i}}{tv_{t,i} + t\epsilon_{t,i}}
\]

\[
+ \frac{\alpha}{2\gamma} \sum_{i=1}^{d} \frac{(1 - \gamma)(1 + \log T)}{\text{inf}_{j \in [1,T]} jv_{j,i} + j\epsilon_{j,i}}
\]

We note that

\[
\log(|TA_T|) = \log \left( \prod_{i=1}^{d} (T(v_{t,i} + \epsilon_{t,i})) \right)
\]

\[
= \sum_{i=1}^{d} \log(T(v_{t,i} + \epsilon_{t,i}))
\]

and similar, \( \log(|\text{diag}(\epsilon_1)|) = \sum_{i=1}^{d} \log(\epsilon_{t,i}) \). \( \square \)

Note that for \( \gamma = 1 \) and the choice \( \epsilon_t = \frac{\delta_t}{t} \) this reduces to the result of Lemma 3.2.

Lemma 4.3 Let \( \epsilon_t \leq \frac{1}{t} \) and \( 1 - \frac{1}{t} \leq \beta_t \leq 1 - \frac{2}{t} \) for some \( 1 \geq \gamma > 0 \). Then it holds,

\[
\sum_{t=2}^{T} \frac{\alpha}{2} \left( (tA_t)^{-1} \cdot \left( \frac{-\text{diag}(\epsilon_t) + \beta_t \text{diag}(\epsilon_{t-1})}{1 - \beta_t} \right) \right)
\]
Theorem 4.2 Let Assumptions A1, A2 hold and let \( \theta_t \) be the sequence generated by SC-RMSProp in Algorithm 4, where \( g_t \in \partial f_t(\theta_t) \) and \( f_t : C \to \mathbb{R} \) is an arbitrary \( \mu \)-strongly convex function (\( \mu \in \mathbb{R}_+^d \)) with \( \alpha_t = \frac{\alpha}{t} \) for some \( \alpha \in \frac{(2-\gamma)G_f^2}{2\min_{i \in [d]} \mu_i} \) and \( 1 - \frac{1}{T} \leq \beta_t \leq 1 - \frac{2}{T} \) for some \( 0 < \gamma \leq 1 \). Furthermore, set \( \epsilon_t = \frac{\alpha_t}{T} \) and assume \( 1 \geq \delta_t > 0 \) and \( \delta_{t,i} \leq \delta_{t-1,i} \forall i \in [T], \forall i \in [d] \), then the regret of SC-RMSProp can be upper bounded for \( T \geq 1 \) as

\[
R(T) \leq D_{\alpha}^2 \cdot \log \left( \frac{T}{\delta_{t,i}} + \delta_{t,i} \right) + \frac{\alpha}{2\gamma} \sum_{t=1}^{T} \left( T \nu_{T,i} + \delta_{T,i} \right) + \frac{T}{2\alpha} \sum_{t=1}^{T} \left( \frac{t \epsilon_t \alpha}{\gamma} \right) (\delta_{T,i} - \delta_{t,i})
\]

Proof: We rewrite the regret bound with the definition of \( \mu \)-strongly convex functions as

\[
R(T) = \sum_{t=1}^{T} \left( f_t(\theta_t) - f_t(\theta^*) \right)
\]

Using the non-expansiveness of the weighted projection, we get

\[
||\theta_t + \theta^*||_{A_t}^2
\]

\[
= \left( \frac{P_{\mu_t}^C (\theta_t - \alpha_t A_t^{-1} g_t) - \theta^*)^2}{A_t} \right)
\]

\[
\leq \frac{||\theta_t - \alpha_t A_t^{-1} g_t - \theta^*||_{A_t}^2}{A_t}
\]

\[
\leq \frac{||\theta_t - \theta^*||_{A_t}^2 - 2\alpha_t \langle g_t, \theta_t - \theta^* \rangle + \alpha_t^2 \langle g_t, A_t^{-1} g_t \rangle}{2\alpha_t}
\]

This yields

\[
\langle g_t, \theta_t - \theta^* \rangle
\]

\[
\leq \frac{||\theta_t - \theta^*||_{A_t}^2 - ||\theta_{t+1} - \theta^*||_{A_t}^2}{2\alpha_t} + \frac{\alpha_t}{2} \langle g_t, A_t^{-1} g_t \rangle
\]

Hence we can upper bound the regret as follows

\[
R(T)
\]

\[
\leq \frac{\alpha}{2\gamma} \sum_{t=1}^{T} \left( T \nu_{T,i} + \delta_{T,i} \right) + \frac{T}{2\alpha} \sum_{t=1}^{T} \left( \frac{t \epsilon_t \alpha}{\gamma} \right) (\delta_{T,i} - \delta_{t,i})
\]
Now on imposing the following condition
\[ tA_t - (t-1)A_{t-1} - 2\alpha \text{diag}(\mu) \leq \text{diag}(\delta_t) - \text{diag}(\delta_{t-1}) \forall t \geq 2 \] (10)
\[ A_1 - 2\alpha \text{diag}(\mu) \leq \text{diag}(\delta_1) \] (11)
Note that with \( \epsilon_{t,i} = \frac{\delta_{t,i}}{t} \) and \( \delta_{t,i} \leq \delta_{t-1,i} \) for all \( t \geq 1 \) and \( i \in [d] \), it holds \( t\epsilon_{t,i} \leq (t-1)\epsilon_{t-1,i} \). We show regarding the inequality in (10)
\[ tv_{t,i} + t\epsilon_{t,i} - (t-1)v_{t-1,i} - (t-1)\epsilon_{t-1,i} - 2\alpha \mu_i \]
\[ = t\beta_{t,i}v_{t-1,i} + t\epsilon_{t,i} + t(1-\beta_t)g_{t,i}^2 - (t-1)\epsilon_{t-1,i} - (t-1)v_{t-1,i} - 2\alpha \mu_i \]
\[ = (t\beta_{t,i} - (t-1))v_{t-1,i} + t(1-\beta_t)g_{t,i}^2 - 2\alpha \mu_i + t\epsilon_{t,i} - (t-1)\epsilon_{t-1,i} \]
\[ \leq (1-\gamma)v_{t-1,i} + g_{t,i}^2 - 2\alpha \mu_i + t\epsilon_{t,i} - (t-1)\epsilon_{t-1,i} \]
\[ \leq (1-\gamma)G^2 + G^2 - 2\alpha \mu_i + t\epsilon_{t,i} - (t-1)\epsilon_{t-1,i} \]
\[ = (2-\gamma)G^2 - 2\alpha \mu_i + t\epsilon_{t,i} - (t-1)\epsilon_{t-1,i} \]
\[ \leq t\epsilon_{t,i} - (t-1)\epsilon_{t-1,i} , \]
where the last inequality follows by choosing \( \alpha \geq \frac{G^2_{\infty}}{2m_{\infty} \mu_i} (2-\gamma) \) and we have used that
\[ v_{t,i} = \sum_{j=1}^{t} (1-\beta_j) \prod_{k=j+1}^{t} \beta_k g_{j,i}^2 \]
\[ \leq G^2 \sum_{j=1}^{t} \left( \prod_{k=j+1}^{t} \beta_k - \prod_{k=j}^{t} \beta_k \right) \]
\[ \leq G^2 \left( 1 - \prod_{k=1}^{t} \beta_k \right) \leq G^2 \]
The second inequality (11) holds easily with the given choice of \( \alpha \). Choosing some \( \beta_t = 1 - \frac{\gamma}{t} \)
\[ tv_{t,i} = t\beta_{t,i}v_{t-1,i} + t(1-\beta_t)g_{t,i}^2 \geq (t-1)v_{t-1,i} + t(1-\beta_t)g_{t,i}^2 \geq (t-1)v_{t-1,i} \]
Hence \( \delta_{t,i} \leq \delta_{t-1,i} \) where \( \delta_{t,i} = e^{-\epsilon} v_{t,i} \) for \( \epsilon > 0 \). With \( \epsilon_{t,i} = \frac{\delta_{t,i}}{t} \) we have \( t\epsilon_{t,i} \leq (t-1)\epsilon_{t-1,i} \).
\[ R(T) \]
\[ \leq \frac{\|\theta_1 - \theta^*\|_{\text{diag}(\delta_1)}^2}{2\alpha} + \sum_{t=2}^{T} \frac{\|\theta_t - \theta^*\|_{\text{diag}(\delta_{t-1}) - \text{diag}(\delta_t)}}{2\alpha} \]
\[ + \sum_{t=1}^{T} \frac{\alpha t}{2} \langle g_t, A_t^{-1}g_t \rangle \]
\[ = \frac{\|\theta_1 - \theta^*\|_{\text{diag}(\delta_1)}^2}{2\alpha} + \sum_{i=2}^{T} \sum_{i=1}^{d} \frac{(\theta_{t,i} - \theta_{t,i}^*)^2 (\delta_{t,i} - \delta_{t-1,i})}{2\alpha} \]
\[ \leq \frac{D^2_\alpha \text{tr(diag}(\delta_1))}{2\alpha} + \sum_{t=1}^{T} \frac{\alpha t}{2} \langle g_t, A_t^{-1}g_t \rangle \]
\[ + \sum_{t=2}^{T} \sum_{i=1}^{d} \frac{(\theta_{t,i} - \theta_{t,i}^*)^2 (\delta_{t,i} - \delta_{t-1,i})}{2\alpha} \]
\[ \leq \frac{D^2_\alpha \text{tr(diag}(\delta_1))}{2\alpha} + \alpha \frac{d}{2\gamma} \sum_{i=1}^{d} \log \left( \frac{Tv_{T,i} + \delta_{T,i}}{\delta_{1,i}} \right) \]
\[ + \frac{1}{2} \sum_{i=1}^{d} \left( \frac{(\theta_{t,i} - \theta_{t,i}^*)^2 (\delta_{t,i} - \delta_{t-1,i})}{\alpha} \right) \]
\[ + \frac{\alpha}{2\gamma} \sum_{i=1}^{d} \left( (1-\gamma)(1 + \log T) \right) \]
\[ + \frac{1}{2} \sum_{i=1}^{d} \inf_{f \in [1,T]} \left[ \left( \frac{(\theta_{t,i} - \theta_{t,i}^*)^2}{\alpha} \right) \left( \frac{\alpha (\delta_{t,i} - \delta_{t-1,i})}{\gamma(tv_{t,i} + \epsilon_{t,i})} \right) \right] \]
where we have used Lemma 4.2 in the last inequality and \( \epsilon_{t,i} = \frac{\delta_{t,i}}{t} \) for \( i \in [d] \) and \( t \geq 1 \).

Note that the regret bound reduces for \( \gamma = 1 \) to that of SC-Adagrad. For \( 0 < \gamma < 1 \) a comparison between the bounds is not straightforward as the \( v_{t,i} \) terms cannot be compared. It is an interesting future research question whether it is possible to show that one scheme is better than the other one potentially dependent on the problem characteristics.

5. Experiments
The idea of the experiments is to show that the proposed algorithms are useful for standard learning problems in both online and batch settings. We are aware of the fact that in the strongly convex case online to batch conversion is not tight (Hazan & Kale, 2014), however that does not necessarily imply that the algorithms behave generally suboptimal. We compare all algorithms for a strongly convex problem and present relative suboptimality plots, \( \log_{10} \left( \frac{\hat{f}(x_k) - p^*}{p^*} \right) \), where \( p^* \) is the global optimum, as well as separate regret plots, where we compare to the best optimal parameter in hindsight for the fraction of training points seen so far. On the other hand RMSProp was originally developed by (Hinton et al., 2012) for usage in deep learning. As discussed before the fixed choice of \( \beta_t \) is
We compare 1) Stochastic Gradient Descent (SGD) (Bottou, 2010) with $O(1/t)$ decaying step-size for the strongly convex problems and for non-convex problems we use a constant learning rate, 2) Adam (Kingma & Bai, 2015), is used with step size decay of $\alpha_t = \frac{\alpha_0}{\sqrt{t}}$ for strongly convex problems and for non-convex problems we use a constant step-size. 3) Adagrad, see Algorithm 1, remains the same for strongly convex problems and non-convex problems. 4) RMSProp as proposed in (Hinton et al., 2012) is used for both strongly convex problems and non-convex problems with $\beta_t = 0.9 \forall t \geq 1$. 5) RMSProp (Ours) is used with step-size decay of $\alpha_t = \frac{\alpha_0}{\sqrt{t}}$ and $\beta_t = 1 - \frac{9}{t}$. In order that the parameter range is similar to the original RMSProp ((Hinton et al., 2012)) we fix as $\gamma = 0.9$ for all experiment (note that for $\gamma = 1$ RMSProp (Ours) is equivalent to Adagrad), 6) SC-RMSProp is used with stepsize $\alpha_t = \frac{\alpha_0}{\sqrt{t}}$ and $\gamma = 0.9$ as RMSProp (Ours) 7) SC-Adagrad is used with a constant stepsize $\alpha$. The decaying damping factor for both SC-Adagrad and SC-RMSProp is used with $\xi_1 = 0.1$, $\xi_2 = 1$ for convex problems and we use $\xi_1 = 0.1$, $\xi_2 = 0.1$ for non-convex deep learning problems. Finally, the numerical stability parameter $\delta$ used in Adagrad, Adam, RMSProp is set to $10^{-8}$ as it is typically recommended for these algorithms.

**Setup:** Note that all methods have only one varying parameter: the stepsize $\alpha$ which we choose from the set of $\{1, 0.1, 0.01, 0.001, 0.0001\}$ for all experiments. By this setup no method has an advantage just because it has more hyperparameters over which it can optimize. The optimal
rate is always chosen for each algorithm separately so that one achieves either best training objective or best test performance after a fixed number of epochs.

Strongly Convex Case - Softmax Regression: Given the training data \((x_i, y_i) \in [m]\) and let \(y_i \in [K]\), we fit a linear model with cross entropy loss and use as regularization the squared Euclidean norm of the weight parameters. The objective is then given as

\[
J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \log \left( \frac{e^{\theta^T x_i + b_i}}{\sum_{j=1}^{K} e^{\theta^T x_i + b_j}} \right) + \lambda \sum_{k=1}^{K} \|\theta_k\|^2
\]

All methods are initialized with zero weights. The regularization parameter was chosen so that one achieves the best prediction performance on the test set. The results are shown in Figure 1. We also conduct experiments in an online setting, where we restrict the number of iterations to the number of training samples. Here for all the algorithms, we choose the stepsize resulting in best regret value at the end. We plot the Regret (in log scale) vs dataset proportion, and as expected SC-Adagrad and SC-RMSProp outperform all the other methods across all the considered datasets. Also, RMSProp (Ours) has a lower regret values than the original RMSProp as shown in Figure 2.

Convolutional Neural Networks: Here we test a 4-layer CNN with two convolutional (32 filters of size \(3 \times 3\)) and one fully connected layer (128 hidden units followed by 0.5 dropout). The activation function is ReLU and after the last convolutional layer we use max-pooling over a \(2 \times 2\) window and 0.25 dropout. The final layer is a softmax layer and the final objective is cross-entropy loss. This is a pretty simple standard architecture and we use it for all datasets. The results are shown in Figures 3, 6. SC-RMSProp is competitive in terms of training objective on all datasets though SGD achieves the best performance. SC-Adagrad is not very competitive and the reason seems to be that the numerical stability parameter is too small. RMSProp diverges on CIFAR10 dataset whereas RMSProp (Ours) converges on all datasets and has similar performance as Adagrad in terms of training objective. Both RMSProp (Ours) and SC-Adagrad perform better than all the other methods in terms of test accuracy for CIFAR10 dataset. On both CIFAR100 and MNIST datasets SC-RMSProp is very competitive.

Multi-Layer Perceptron: We also conduct experiments for a 3-layer Multi-Layer perceptron with 2 fully connected hidden layers and a softmax layer according to the number of classes in each dataset. For the first two hidden layers we
Variants of RMSProp and Adagrad with Logarithmic Regret Bounds

have 512 units in each layer with ReLU activation function and 0.2 dropout. The final layer is a softmax layer. We report the results in Figures 7, 8. On all the datasets, SC-Adagrad and SC-RMSProp perform better in terms of Test accuracy and also have the best training objective performance on CIFAR10 dataset. On MNIST dataset, Adagrad and RMSProp (Ours) achieves best training objective performance however SC-Adagrad and SC-RMSProp eventually performs as good as Adagrad. Here, the performance is not as competitive as Adagrad, because the numerical stability decay parameter of SC-Adagrad and SC-RMSProp are too prohibitive.

Residual Network: We also conduct experiments for ResNet-18 network proposed in (He et al., 2016a) where the residual blocks are used with modifications proposed in (He et al., 2016b) on CIFAR10 dataset. We report the results in Figures 4. SC-Adagrad, SC-RMSProp and RMSProp (Ours) have the best performance in terms of test Accuracy and RMSProp (Ours) has the best performance in terms of training objective along with Adagrad.

Given these experiments, we think that SC-Adagrad, SC-RMSProp and RMSProp (Ours) are valuable new adaptive gradient techniques for deep learning.

6. Conclusion
We have analyzed RMSProp originally proposed in the deep learning community in the framework of online convex optimization. We show that the conditions for convergence of RMSProp for the convex case are different than what is used by (Hinton et al., 2012) and that this leads to better performance in practice. We also propose variants SC-Adagrad and SC-RMSProp which achieve logarithmic regret bounds for the strongly convex case. Moreover, they perform very well for different network models and datasets and thus they are an interesting alternative to existing adaptive gradient schemes. In the future we want to explore why these algorithms perform so well in deep learning tasks even though they have been designed for the strongly convex case.

Acknowledgements
We would like to thank Shweta Mahajan and all the reviewers for their insightful comments.
Variants of RMSProp and Adagrad with Logarithmic Regret Bounds

Figure 7. Training Objective vs Number of Epoch for 3-layer MLP

Figure 8. Test Accuracy vs Number of Epochs for 3-layer MLP

References


