Variants of RMSProp and Adagrad with Logarithmic Regret Bounds

Mahesh Chandra Mukkamala[1,2], Matthias Hein[1]

[1]Saarland University, [2] Max Planck Institute for Informatics

Contributions

Motivation:

- Use **RMSProp** (Hinton et al., 2012) in Online Convex Optimization framework.
- Use optimal algorithms for strongly convex problems to train Deep Neural Networks.

Main Contributions:

- **Analyzed RMSProp** (Hinton et al., 2012).
- **Equivalence of RMSProp and Adagrad**.
- Proposed **SC-Adagrad** and **SC-RMSProp** with $\log T$-type optimal regret bounds (Hazan et al. [2007]) for strongly convex problems .
- Better test accuracy on various **Deep Nets**.

Online convex optimization

Notation: In $\mathbb{R}^d$, $(a \odot b)_i = a_i b_i$ for $i = 1, \dots, d$, $\mathbf{0} \in \mathbb{R}^d$. Let $A \succ 0$, $z \in \mathbb{R}^d$, convex set C , then

$$P_C^A(z) = \arg\min_{x \in C} \|x - z\|_A^2 = \langle x - z, A(x - z)\rangle$$

Online Learning setup: Let C be a convex set.
for $t = 1, 2, \dots, T$ **do**

- We predict $\theta_t \in C$.
- Adversary gives $f_t : C \to \mathbb{R}$ (continuous convex)
- We suffer loss $f_t(\theta_t)$, update θ_t, using $g_t \in \partial f_t(\theta_t)$

Goal: To perform well w.r.t $\theta^* = \arg\min_{\theta \in C} \sum_{t=1}^T f_t(\theta)$ and bound regret $R(T) = \sum_{t=1}^T (f_t(\theta_t) - f_t(\theta^*))$.

$\mu-$**strongly convex function** $f : C \to \mathbb{R}$, if $\exists\, \mu \in \mathbb{R}_+^d$ s.t $\forall x, y \in C$,

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

Online Gradient Descent: $\theta_{t+1} = P_C(\theta_t - \alpha_t g_t)$

Convex f_t (Zinkevich, 2003): $\alpha_t = O(\frac{1}{\sqrt{t}})$

$\sqrt{T}$-type optimal data-independent regret bounds.

Strongly Convex f_t (Hazan et al., 2007): $\alpha_t = O(1/t)$

$\log T$-type optimal data-independent regret bounds.

Adagrad (Duchi et al, 2011): $v_0 = \mathbf{0}, \alpha, \delta > 0$

$$v_t = v_{t-1} + (g_t \odot g_t), \quad A_t = \mathrm{diag}(\sqrt{v_t}) + \delta \mathbb{I}$$

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

Main Idea: Adaptivity, effective step-size of $O\Big(\frac{1}{\sqrt{t}}\Big)$

Effective Step-size

Adagrad (Duchi et al, 2011):

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

SC-Adagrad (Ours):

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

SC-Adagrad

With $\theta_1 \in C, \delta_0 > \mathbf{0}, v_0 = \mathbf{0}, \alpha > 0$
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**
$A_t = \mathrm{diag}(\boldsymbol{v_t} + \boldsymbol{\delta_t})$, $\theta_{t+1} = P_C^{A_t}(\theta_t - \alpha A_t^{-1} g_t)$
end for

Decay scheme varies with dimension as $\delta_t \in \mathbb{R}^d$.

Logarithmic Regret Bounds

Let $\sup_{t\geq 1} \|g_t\|_\infty \leq G_\infty$, $\sup_{t \geq 1} \|\theta_t - \theta^*\|_\infty \leq D_\infty$, $f_t : C \to \mathbb{R}$ is μ-strongly convex, $\alpha \geq \max_{i=1,\dots,d} \frac{G_\infty^2}{2\mu_i}$, then regret bound of SC-Adagrad for $T \geq 1$ is

$$R(T) \leq \frac{D_\infty^2 \,\mathrm{tr}(\mathrm{diag}(\delta_1))}{2\alpha} + \frac{\alpha}{2}\sum_{i=1}^d \log\Big(\frac{v_{T,i} + \delta_{T,i}}{\delta_{1,i}}\Big)$$
$$+ \frac{1}{2}\sum_{i=1}^d \inf_{t \in [T]} \Big(\frac{(\theta_{t,i} - \theta_i^*)^2}{\alpha} - \frac{\alpha}{v_{t,i} + \delta_{t,i}}\Big)(\delta_{T,i} - \delta_{1,i})$$

Data-dependent $\log T$-type optimal regret bounds.

RMSProp

RMSProp (Hinton et al., 2012): Most popular adaptive gradient method used in deep learning.

Idea: Moving average of second order gradients.

Can we use RMSProp for Online learning?

RMSProp (Ours): $v_0 = \mathbf{0}, \alpha, \delta > 0, 0 < \gamma \leq 1$

With $\boldsymbol{\beta_t = 1 - \frac{\gamma}{t}}$, $\epsilon_t = \delta/\sqrt{t}$, $\alpha_t = \alpha/\sqrt{t}$

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

$A_t = \mathrm{diag}(\sqrt{v_t}) + \epsilon_t I$, $\theta_{t+1} = P_C^{A_t}(\theta_t - \alpha_t A_t^{-1} g_t)$

For Convex Problems: $\sqrt{T}$-type regret bounds.

For original RMSProp set $\beta_t = 0.9, \alpha_t = \alpha > 0$.

SC-RMSProp

SC-Adagrad + **RMSProp** = **SC-RMSProp**

We need to modify RMSProp (Ours) by:

- Using $\epsilon_t = \delta_t / t$ with $\delta_0 > \mathbf{0}$, where $0 < \delta_t \leq \delta_{t-1}$ element-wise.
- $A_t = \mathrm{diag}(v_t + \epsilon_t)$ and $\alpha_t = \alpha / t$

Idea: Effective step-size is $O(1/t)$ so $\log T$ regret bound.

New Decay scheme: For SC-RMSProp choose $\delta_t = \xi_2 e^{-\xi_1 t v_t}$ and for SC-Adagrad $\delta_t = \xi_2 e^{-\xi_1 v_t}$.

Pros: Enhanced adaptivity, Stabilizes quadratic growth of v_t in g_t, Exponential decay in v_t.

Rule of Thumb: $\xi_1 = 0.1, \xi_2 = 1$ for convex problems and $\xi_1 = 0.1, \xi_2 = 0.1$ for **deep learning**.

Interesting Phenomenon

Choose $\beta = 1 - \frac{1}{t}$, we obtain update step of

RMSProp (Ours) $\equiv$ **Adagrad**

SC-RMSProp $\equiv$ **SC-Adagrad**

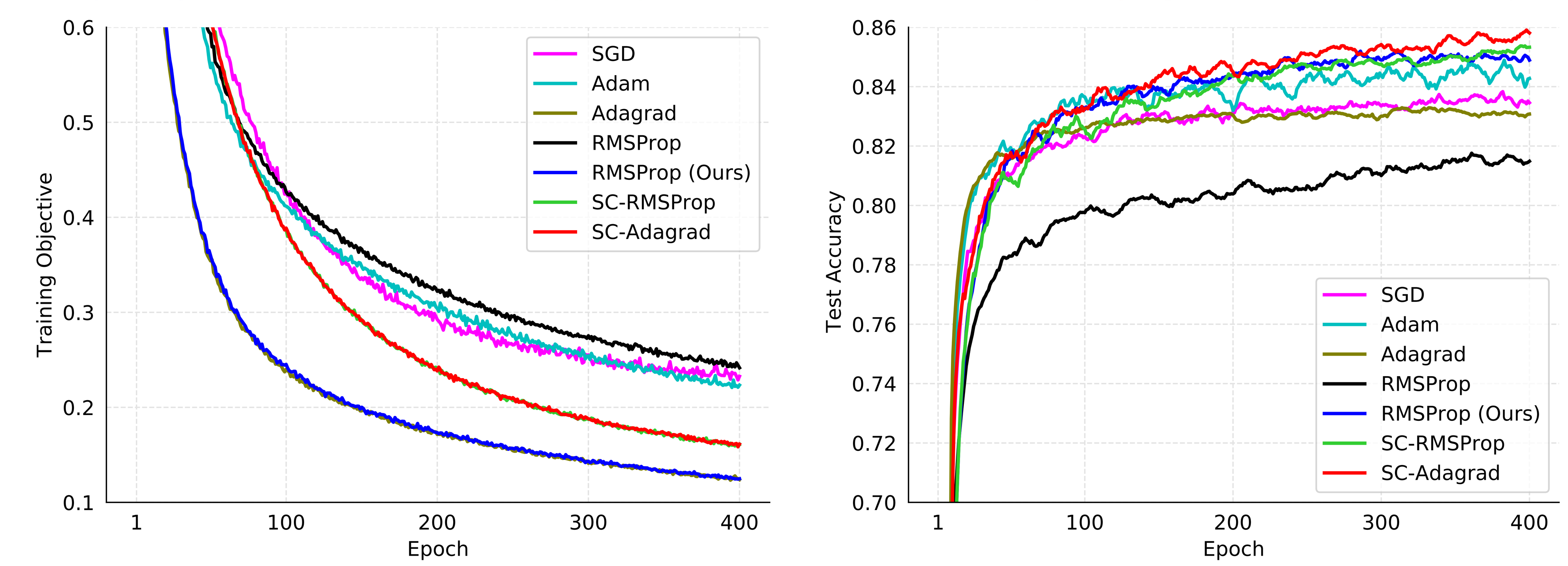
Follows from a simple telescoping sum of v_t.

Experimental Setup :

- Only one varying parameter: the stepsize α chosen from $\{1, 0.1, 0.01, 0.001, 0.0001\}$.
- No method has an advantage just because it has more hyperparameters.
- Optimal rate is chosen so that algorithm achieves best performance (in consideration) at the end.

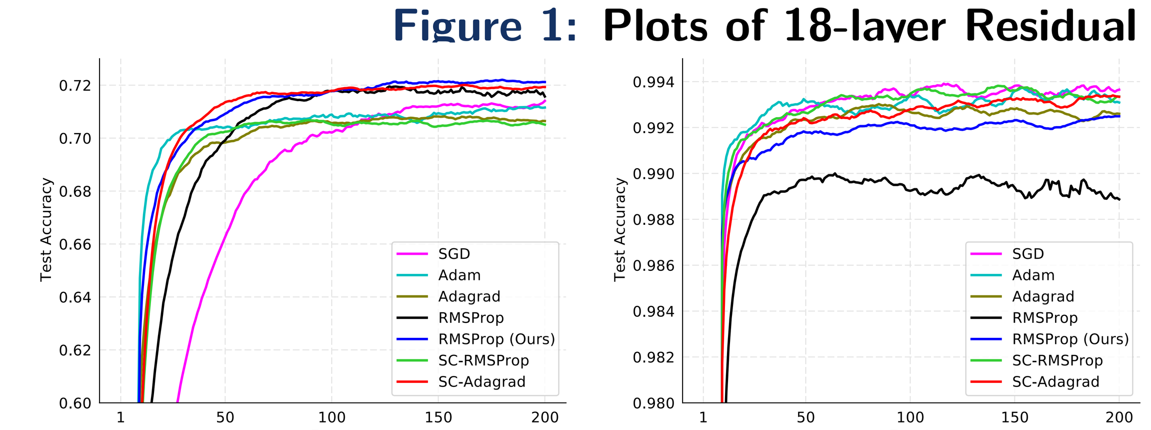
Results of Residual Network, CNN and Softmax Regression

Algorithms: SGD (Bottou, 2010) (step-size is $O(\frac{1}{t})$ for strongly convex problems), Adam (step-size is $O(\frac{1}{\sqrt{t}})$ for strongly convex problems), Adagrad, RMSProp with $\beta_t = 0.9\ \forall t \geq 1$. With $\gamma = 0.9$ we use **RMSProp** (Ours) and **SC-RMSProp** (Ours), finally **SC-Adagrad** (Ours). [**CODE**: github.com/mmahesh]



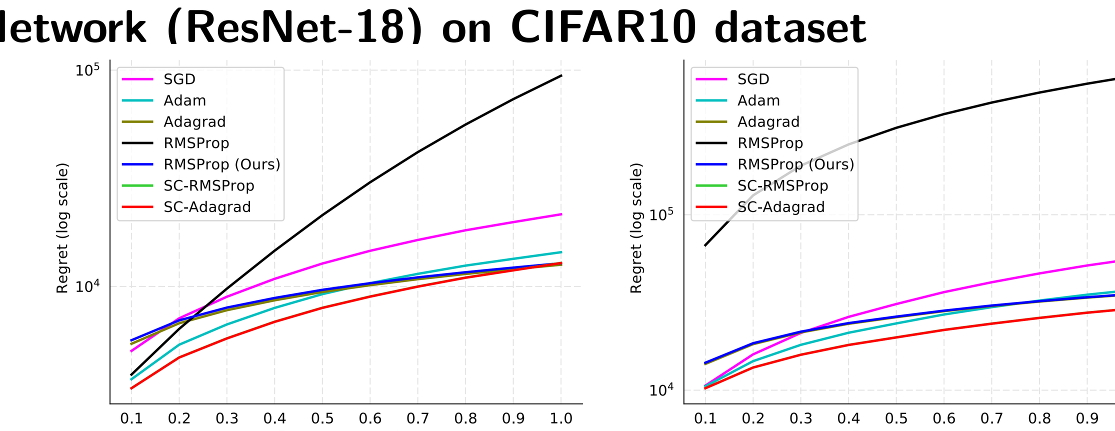
(a) Training Objective (b) Test Accuracy

Figure 1: **Plots of 18-layer Residual Network (ResNet-18) on CIFAR10 dataset**



(a) CIFAR10 (b) MNIST

Figure 2: **Test Accuracy vs Number of Epochs for 4-layer Convolutional Neural Network**



(a) CIFAR10 (b) CIFAR100

Figure 3: **Regret (log scale) vs Dataset Proportion for Online L2-Regularized Softmax Regression**

References:

- Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12:2121-2159, 2011 (COLT, page 257, 2010)
- Hinton, G., Srivastava, N., and Swersky, K. Lecture 6d - a separate, adaptive learning rate for each connection. Slides of Lecture Neural Networks for Machine Learning, 2012.
- Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. Machine Learning, 69(2-3):169-192, 2007