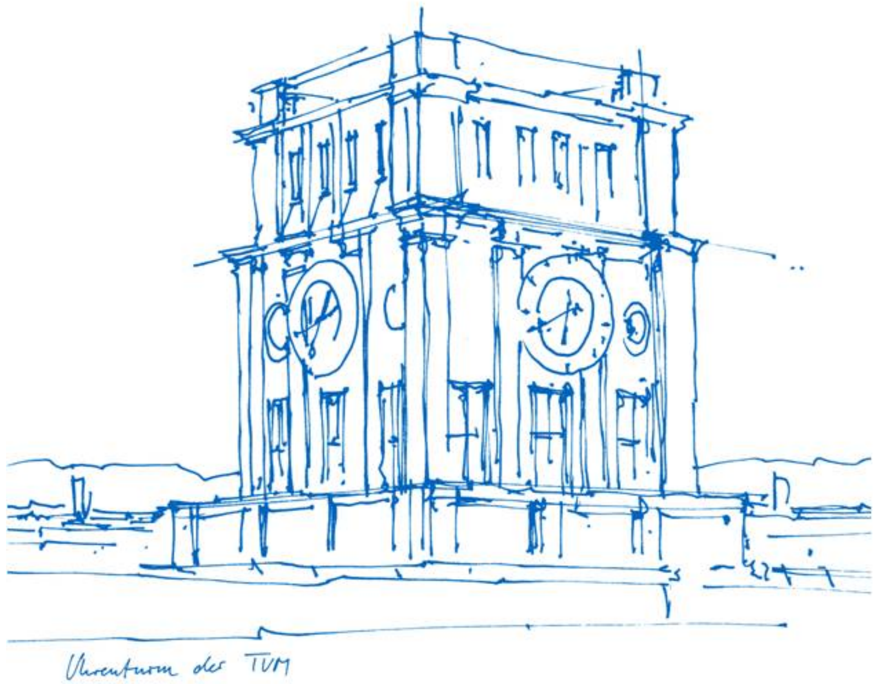
Learning Step Size Controllers for Robust Neural Network Training

Christian Daniel et al.

Recent Trends in Automated Machine Learning

Abeeha Shafiq

18.07.2019



Motivation

- Optimizers are sensitive to initial learning rate
- Good learning rate is problem specific
- Manual search required

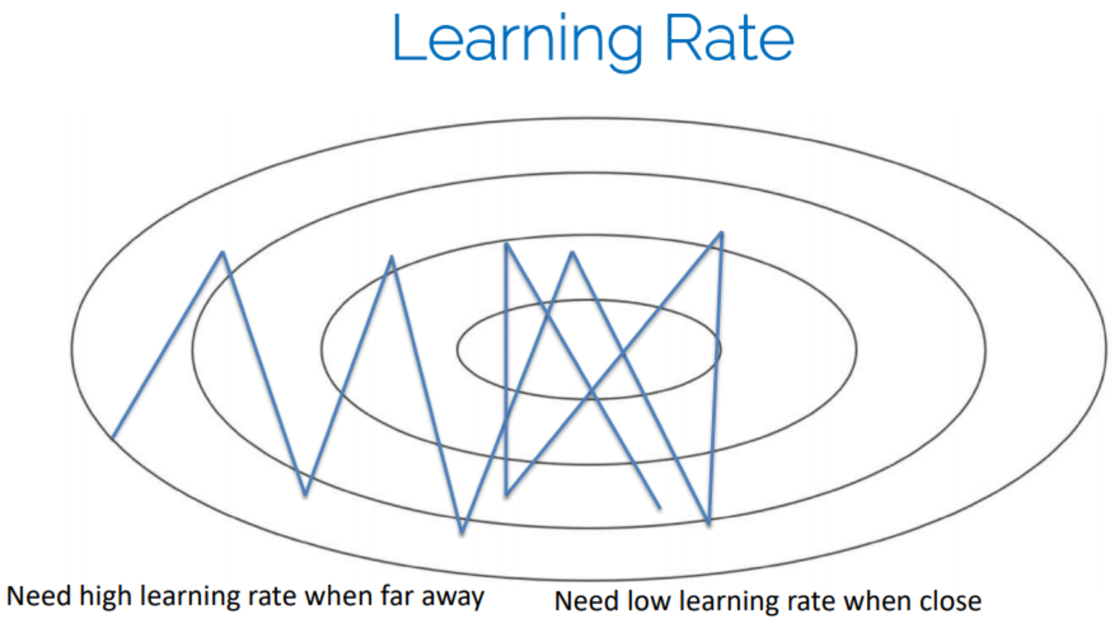


Image taken from I2DL lecture slide

Previous Work

- Waterfall scheme
- Exponential/power scheme
- TONGA

Goal

Develop an adaptive controller for the learning rate used in training algorithms such as Stochastic Gradient Descent (SGD) with Reinforcement Learning

Contributions

- Identifying informative features for controller
- Proposing a learning setup for a controller
- Showing that the resulting controller generalizes across different tasks and architectures.

Problem statement for controller

- Find the minimizer

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} F(\boldsymbol{X};\boldsymbol{\omega}),$$

- $F(\cdot)$ sums over the function values induced by the individual inputs

$$F(\boldsymbol{X};\boldsymbol{\omega}) = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i;\boldsymbol{\omega}).$$

- $T(\cdot)$ is an optimization operator which yields a weight update vector to find $\omega*$

$$\Delta\omega = T(\nabla F, \boldsymbol{\rho}, \boldsymbol{\xi}).$$

- SGD weight update

$$\boldsymbol{w} := \boldsymbol{w} \ -\eta\nabla F$$

Learning a Controller

$$\pi^*(\theta) = \underset{\pi}{\operatorname{argmax}} \quad \mathbb{E}_{\pi(\theta)}\Big[r(g(\phi, \theta))\Big]$$

$$\boldsymbol{\xi} = g(\phi; \boldsymbol{\theta}),$$

Relative Entropy Policy Search (REPS)

$$\mathrm{D}_{\mathrm{KL}}\left(\pi(\boldsymbol{\theta}) || q(\boldsymbol{\theta})\right) \leq \epsilon,$$

Concept similar to Proximal Policy Optimization

$$\mathcal{L}^{CLIP}(\theta) = \mathbb{E}_t\left[\min\{\sigma_t G_t, \mathit{clip}\left(\sigma_t, 1-\varepsilon, 1+\varepsilon\right) G_t\}\right] \quad \text{with} \quad \sigma_t = \frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{old}}\left(a_t \mid s_t\right)}$$

Features

- Informative about current state
- Generalize across different tasks and architectures
- Constrained by computation and memory limits

Features

- **Predictive change in function value.**

$$\Delta \tilde{f}_i = \tilde{f}_i - f_i$$

$$\phi_1 = \log \left(\mathrm{Var}(\Delta \tilde{f}_i) \right) .$$

- **Disagreement of function values.**

$$\phi_2 = \log \left(\mathrm{Var} \left(f \left(\boldsymbol{x}_i ; \boldsymbol{\omega} \right) \right) \right)$$

Mini Batch Setting

- **Discounted Average.**
  - Smooths outliers
  - Serve as memory

$$\hat{\phi}_i \leftarrow \gamma\hat{\phi}_i + (1-\gamma)\phi_i$$

- **Uncertainty Estimate**
  - Estimate of noise in the system

$$\hat{\phi}_{K+i} \leftarrow \gamma\hat{\phi}_{K+i} + (1-\gamma)(\phi_i - \hat{\phi}_i)^2$$

Experimental Setup

- Datasets: MNIST, CIFAR-10
- Learning Algorithms: SGD and RMSProp
- Model: CNN
- For Learning Controller parameters:
  - Subset of MNIST
  - Small CNN architecture
- $\pi(\boldsymbol{\theta})$ to a Gaussian with isotropic covariance

$$\pi^*(\theta) = \underset{\pi}{\operatorname{argmax}} \quad \mathbb{E}_{\pi(\theta)}\Big[r(g(\phi, \theta))\Big]$$

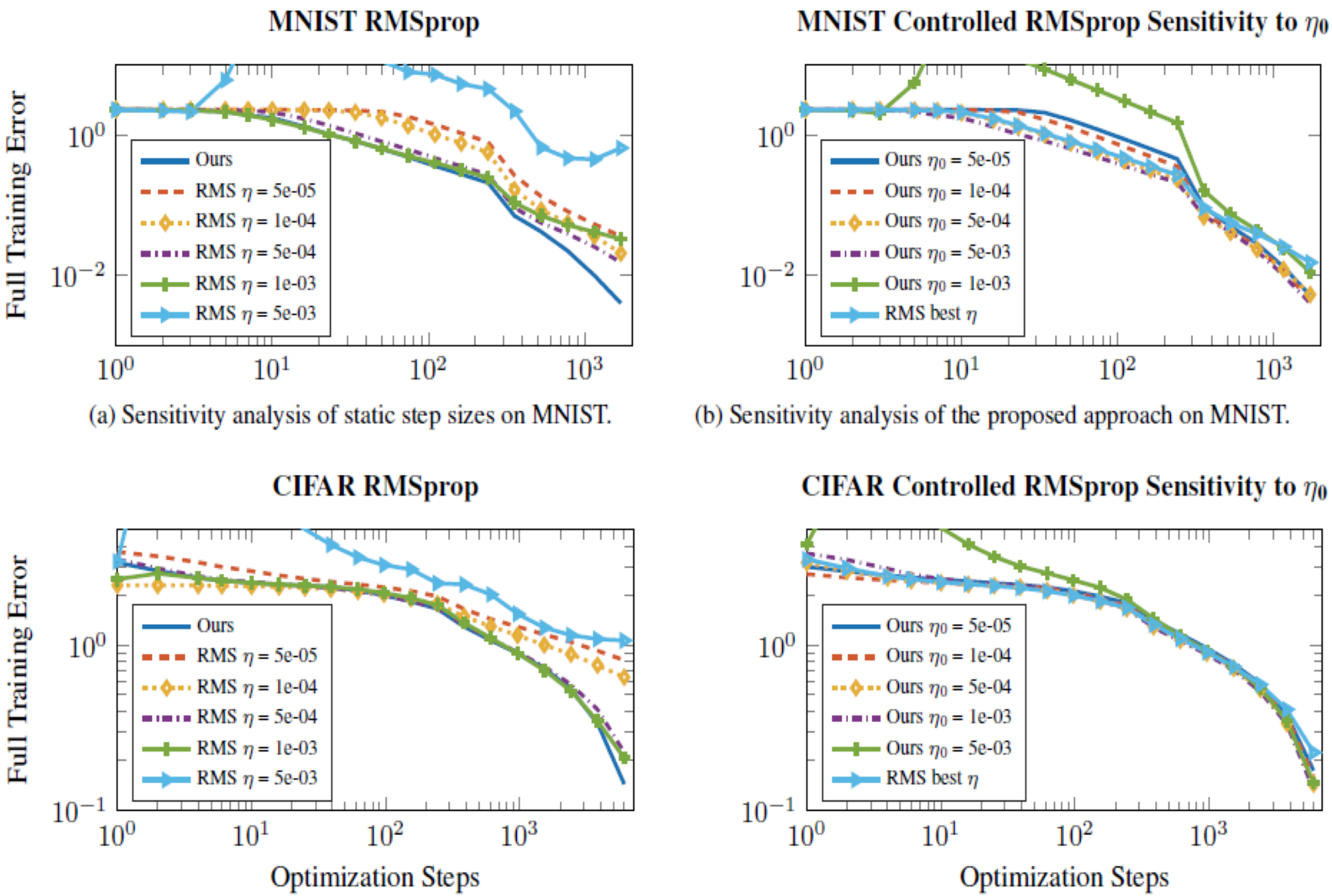
$$g(\hat{\phi}; \boldsymbol{\theta}) = \exp(\boldsymbol{\theta}^T \hat{\phi})$$

$$r = -\frac{1}{S-1} \sum_{s=2}^{S} \Big(\log(E_s) - \log(E_{s-1}) \Big)$$

Results

- overhead of 36% for controller training
- Generalized to different variants of CNN
- Did not generalize to different training methods

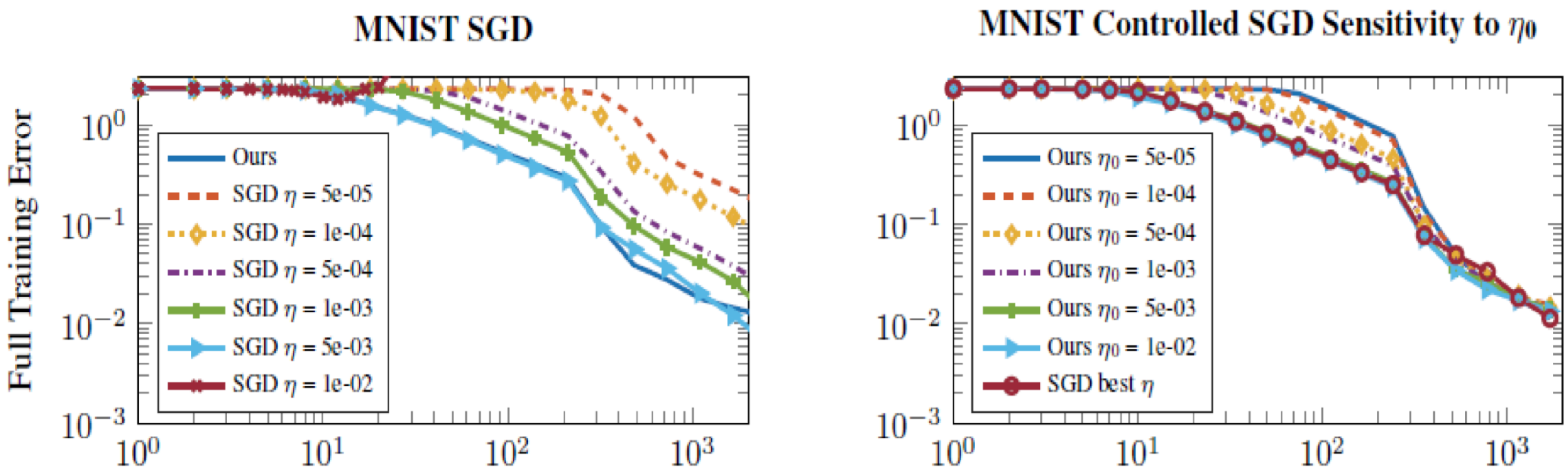
Static RMSProp vs Controlled RMSProp



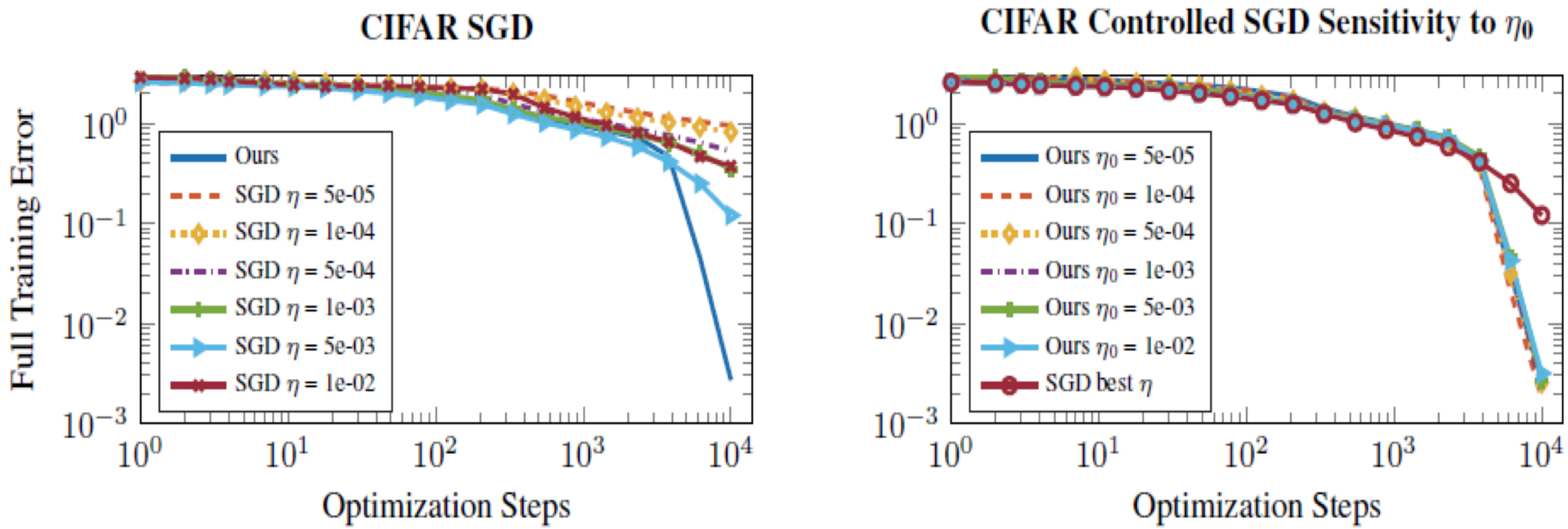
(a) Sensitivity analysis of static step sizes on MNIST.

(b) Sensitivity analysis of the proposed approach on MNIST.

Static SGD vs Controlled SGD



(a) Sensitivity analysis of static step sizes on MNIST. (b) Sensitivity analysis of the proposed approach on MNIST.



Discussion

- **Strengths:**
  - Features
  - Not sensitive to initial learning rate
  - Effort to generalize

- **Weakness:**
  - Tested on only 2 dataset
  - CNN only
  - Lacks comparison with
    - learning rate decay techniques
    - Grid search for initial learning rate

This is a prior technique to learning the complete optimizer

Questions?