CSED490Y

Adam: A Method for Stochastic Optimization

Group13; 20190829 Lee Taehui

INDEX

01

Introduction

- **A method for efficient stochastic optimization that only requires first-order gradients with little memory requirement.**
- **Combine the advantages of two recently popular methods: AdaGrad & RMSProp**
- **A versatile algorithm that scales to large-scale high-dimensional machine learning problems.**

- **AdaGrad**

$$\theta_t = \theta_{t-1} - \alpha \frac{g_t}{\sqrt{\sum_{i=1}^{t} g_i^2}}$$

- **RMSProp**

$$G_t = \gamma G_{t-1} + (1-\gamma) g_t^2$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G_t + \epsilon}} g_t$$

02

Method

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Pseudo-code of Adam

- **Adam's Update Rule**

$$|\Delta_t| = \frac{\alpha \cdot \widehat{m_t}}{\sqrt{\widehat{v_t}}}$$

$$|\Delta_t| \leq \frac{\alpha \cdot (1-\beta_1)}{\sqrt{(1-\beta_2)}} \qquad \textit{in the case } (1-\beta_1) > \sqrt{1-\beta_2}$$

$$|\Delta_t| \leq \alpha \qquad \textit{otherwise}$$

$$\rightarrow |\Delta_t| <\approx \alpha$$

- **Initialization Bias Correction**

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$\begin{aligned}\mathbb{E}[v_t] &= \mathbb{E}\left[(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\cdot g_i^2\right] \\ &= \mathbb{E}[g_t^2]\cdot(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}+\zeta \\ &= \mathbb{E}[g_t^2]\cdot(1-\beta_2^t)+\zeta\end{aligned}$$

03

Experiment Results

Experiment Results

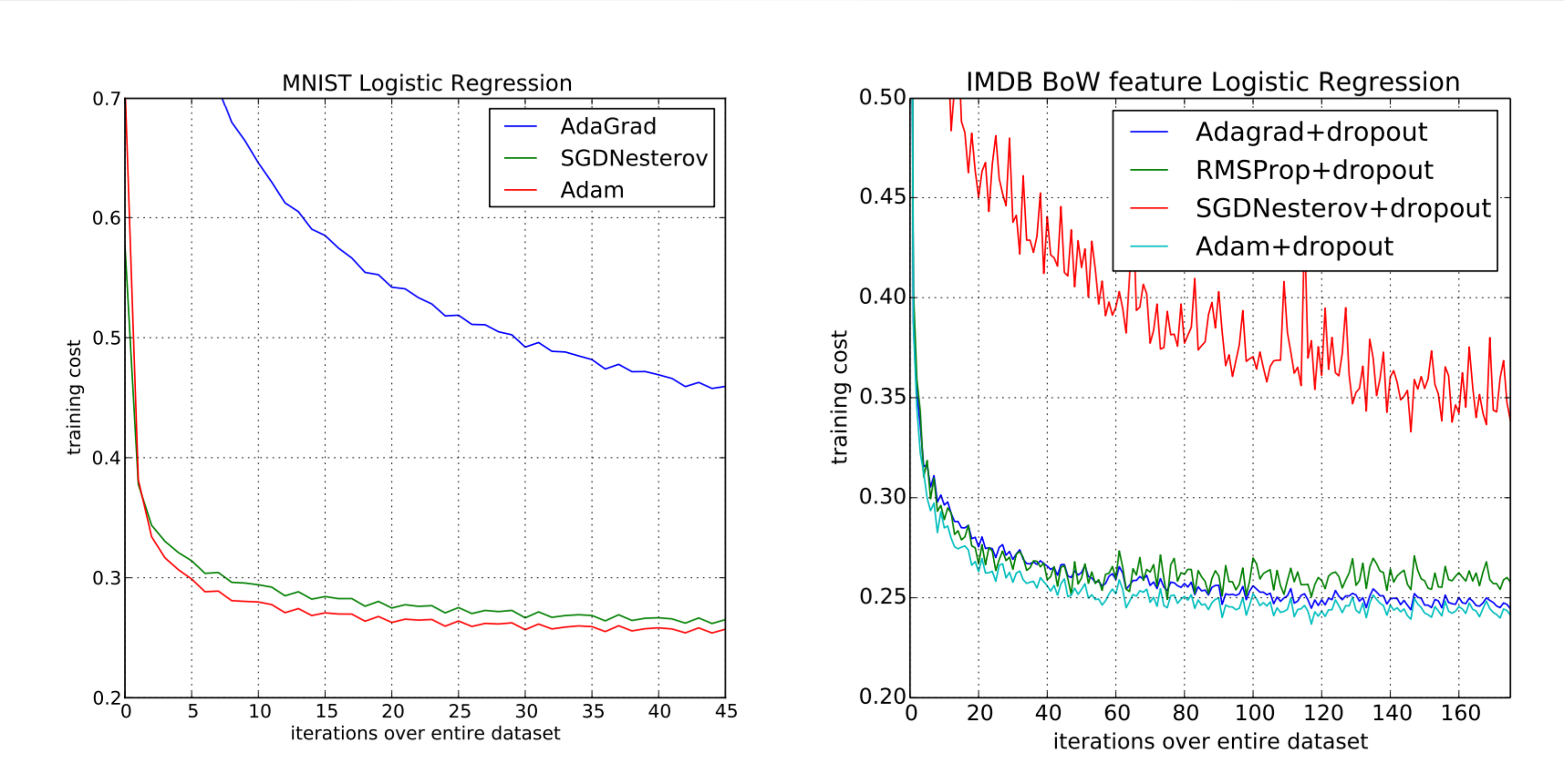

Figure 1: Logistic regression training negative log likelihood on MNIST images and IMDB movie reviews with 10,000 bag-of-words (BoW) feature vectors.

Experiment Results

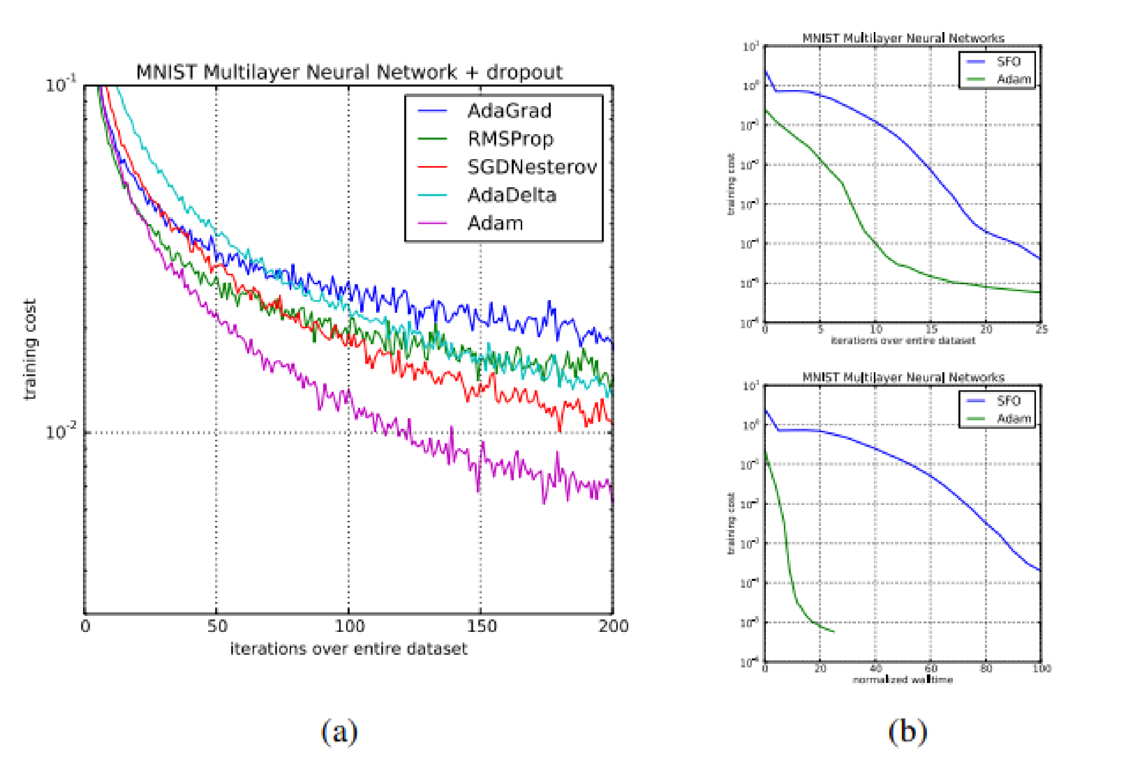

Figure 2: Training of multilayer neural networks on MNIST images.
(a) Neural networks using dropout stochastic regularization. (b) Neural networks with deterministic cost function. We compare with the sum-of-functions (SFO) optimizer (Sohl-Dickstein et al., 2014)

Experiment Results

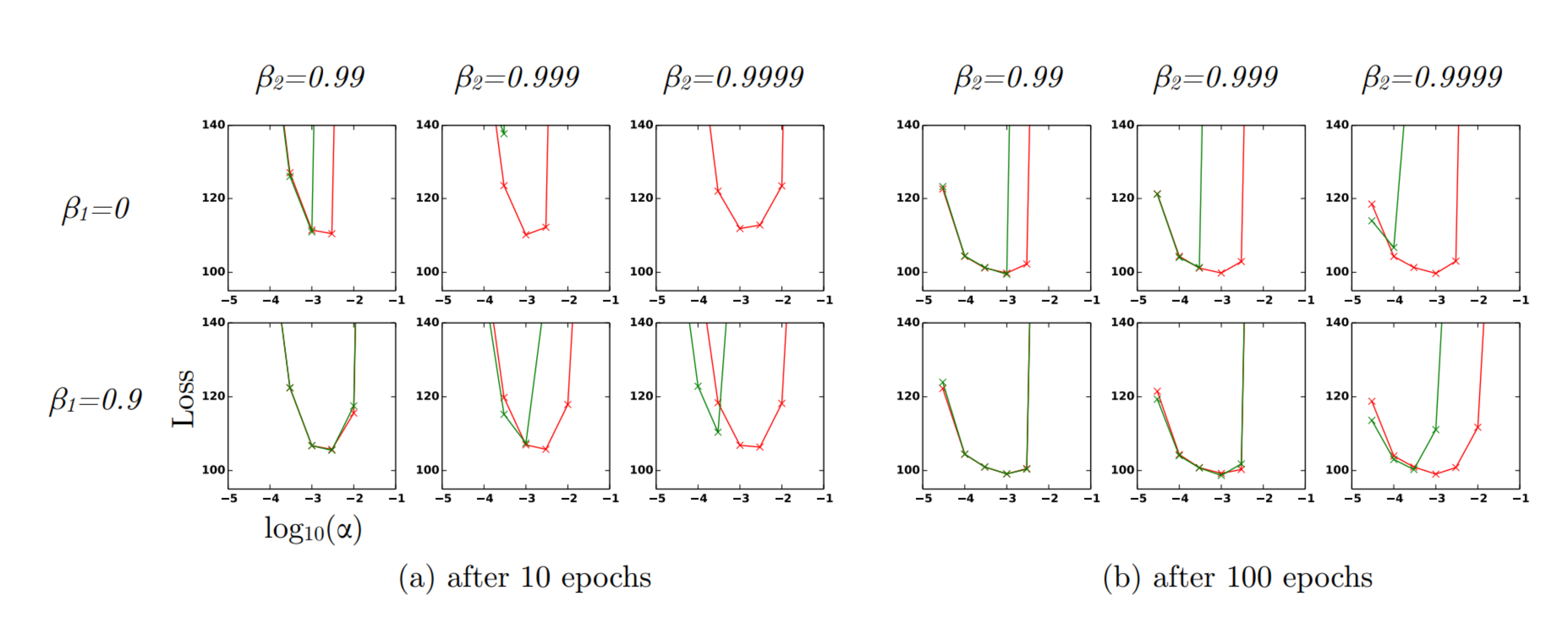

Figure 3: Effect of bias-correction terms (red line) versus no bias correction terms (green line) after 10 epochs (left) and 100 epochs (right) on the loss (y-axes) when learning a Variational AutoEncoder (VAE) (Kingma & Welling, 2013), for different settings of stepsize α (x-axes) and hyperparameters β1 and β2.

04

Relation to Project

Relation to Project

- **Deeper understanding of SGD and its optimization techniques**
- **Implementation Adam optimizer and image classifier using Adam**
- **Compare Adam with other Optimizers such as SGD, Adagrad, RMSProp, …**
- **Propose ideas for a better optimization method from Adam**

THE END

Thank You