ADAM : A method for stochastic optimization

Kingma, Diederik P., and Jimmy Ba.

수학과 오서영

Overview

ADAM (Adaptive moment estimation)

- Algorithm for first-order gradient-based optimization of stochastic objective functions based on adaptive estimates of lower-order moments.
- Straightforward to implement, computationally efficient, little memory requirements, invariant to diagonal rescaling of the gradients, well suited for problems that are large in terms of data or params
- Appropriate for non-stationary objectives and problems with very noisy and sparse gradients
- Hyper params have intuitive interpretations and typically require little tuning.

- Analyze theoretical convergence properties of the algorithm
- Provide a regret bound on the convergence rate that is comparable to the best known results under the online convex optimization framework.

Introduction

- Stochastic gradient-based optimization is of core practical importance in many fields
- Many problems in these fields can be cast as the optimization of some scalar parameterized objective function requiring maximization or minimization with respect to its parameters.
- If the function is differentiable, gradient descent is a relatively efficient optimization method, since the computation of first-order partial derivatives is of the same computational complexity as just evaluating the function.

- **ADAM** computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients

-> designed to combine the advantages of AdaGrad and RMSProp

-> **Advantages** : magnitudes of param updates are invariant to rescaling of the gradient, its stepsizes are approximately bounded by the stepsize hyperparameters

Adam algorithm pseudo-code

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)

Adam algorithm

The algorithm updates exponential moving averages of the gradient (m) and The squared gradient (v) where the hyper-params beta1, beta2 control the exponential decay rates of these moving averages.

These moving averages are initialized as 0, leading to moment estimates that are biased towards zero, especially during the initial timesteps, and especially when the decay rates are small (i.e beta are close to 1)

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = m_t / (1 - \beta_1^t)$$

$$\hat{v}_t = v_t / (1 - \beta_2^t)$$

Initialization bias correction

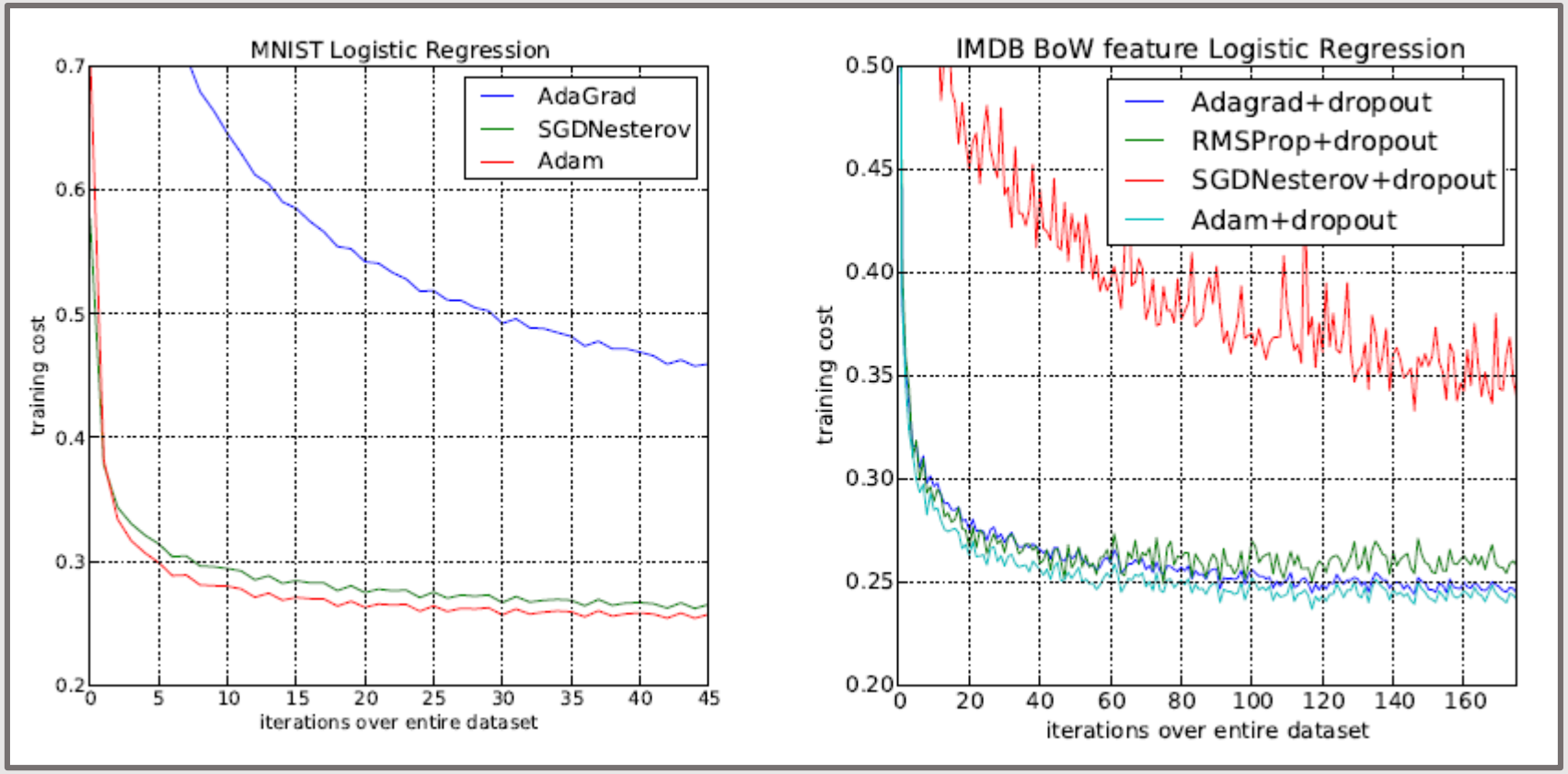
$$v_t = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\cdot g_i^2$$

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

Adam algorithm

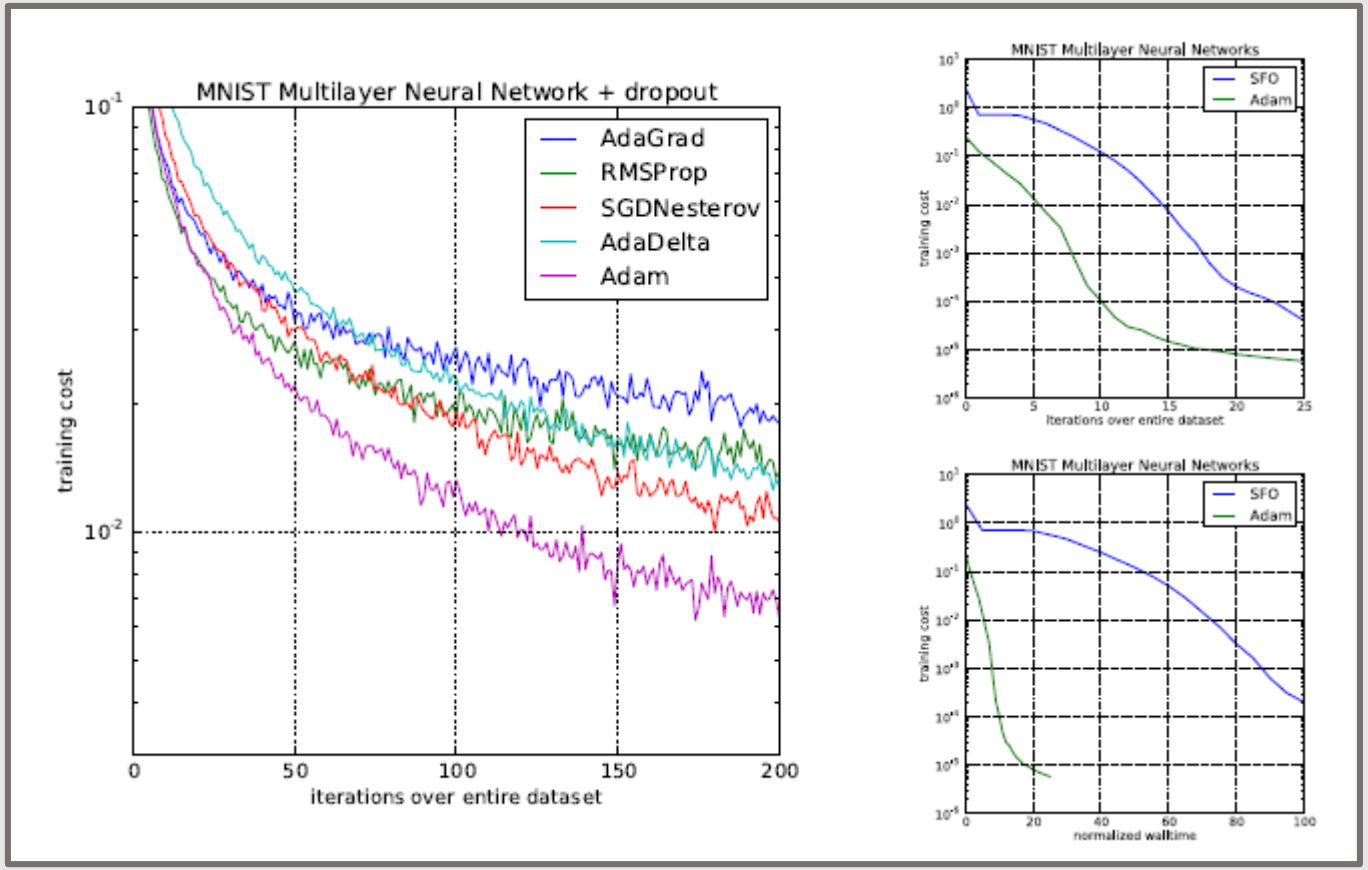
- Calculate a robust and efficient step size
- Exponential moving average of first moment (mean) and second moment (variance) of gradient into the procedure of parameter update formula
- Consider a bias correction term because the initial value (= 0) of the primary moment and the secondary moment will be biased.
- Bias correction is done using the smoothing factor
- Gradient scale is canceled by the ratio of the primary moment to the secondary moment, so the step size is stable.
- Automate learning coefficient update

Results



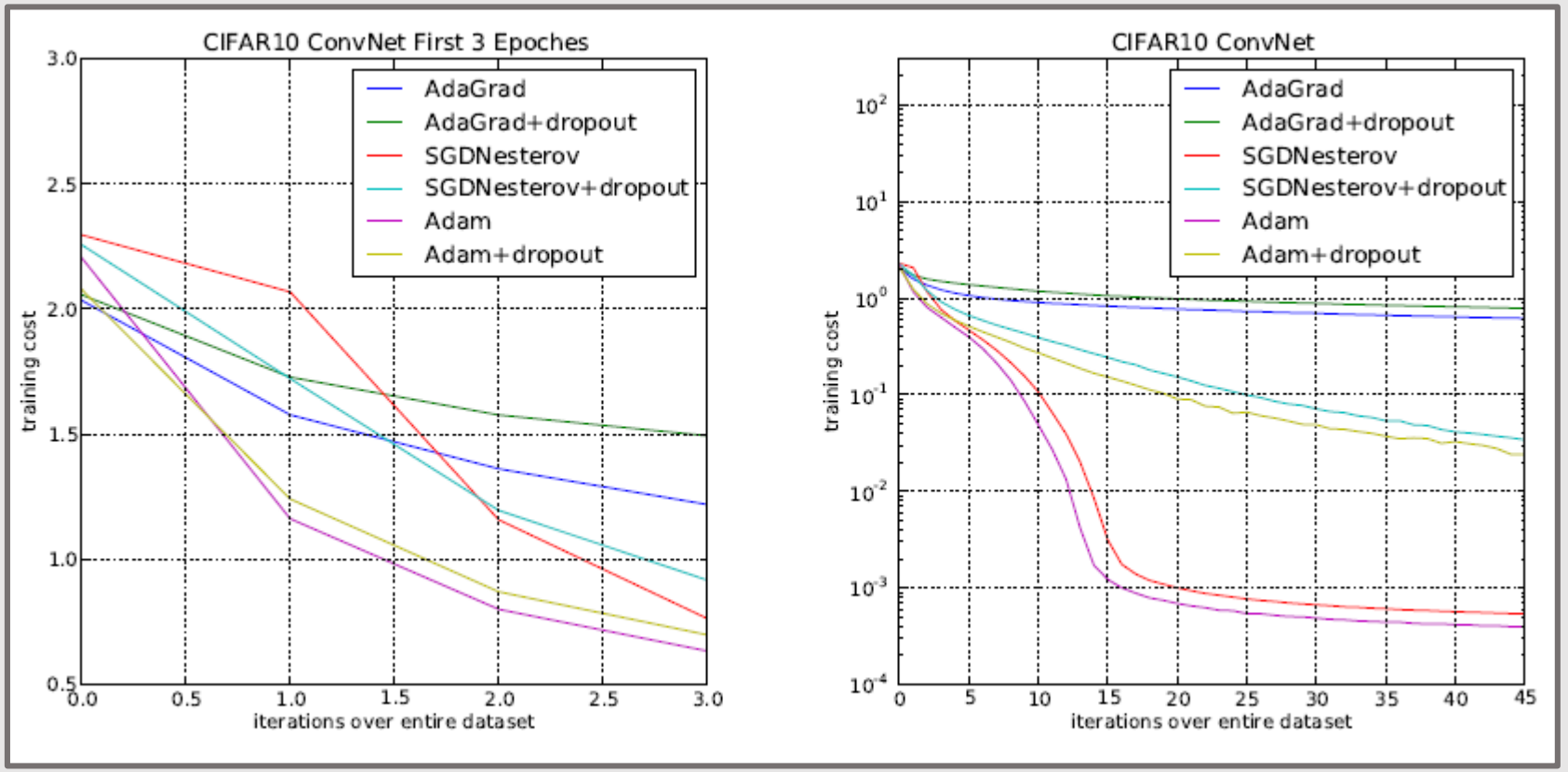
logistic regression

Results



multi-layer neural network

Results


CIFAR10 ConvNet First 3 Epoches
training cost
iterations over entire dataset
AdaGrad
AdaGrad+dropout
SGDNesterov
SGDNesterov+dropout
Adam
Adam+dropout
CIFAR10 ConvNet
training cost
iterations over entire dataset
AdaGrad
AdaGrad+dropout
SGDNesterov
SGDNesterov+dropout
Adam
Adam+dropout

CNN

Experiments

- The loss decreases as the secondary moment approaches zero.
- Learning speed is increased by learning so that the secondary moment becomes smaller.
- In fact, I discovered that I was also strong in non-convex functions