ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION

Group 14

E/15/139 - Ishanthi D.S. - e15139@eng.pdn.ac.lk
E/15/249 - Pamoda W.A.D. - dasunip2@gmail.com
E/15/299 - Ranushka L.M. - e15299@eng.pdn.ac.lk

05/02/2021

Background

Authors : Diederik P. Kingma
Jimmy Lei Ba

Published : as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015

What is ADAM?

- The name Adam is derived from adaptive moment estimation.
- An algorithm for
    - first-order gradient-based optimization of stochastic objective functions
    - based on adaptive estimates of lower-order moments.
- The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients;
- Combine the advantages of AdaGrad and RMSProp

Advantages

- Combines the advantages of two popular optimization methods:
  - AdaGrad -the ability to deal with sparse gradients
  - RMSProp -the ability to deal with non-stationary objectives.
- Straightforward to implement
- Computationally Efficient. (less memory requirements)
- Robust and well-suited to a wide range of non-convex optimization problems in the field machine learning.
- The magnitudes of parameter updates are invariant to re-scaling of the gradient,
- Its stepsizes are approximately bounded by the stepsize hyperparameter
- It does not require a stationary objective
- It works with sparse gradients
- It naturally performs a form of step size annealing.

Algorithm

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’S UPDATE RULE

- Adam’s update rule is its careful choice of stepsizes

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon) \text{ (Update parameters)}$$

- Stepsize is a value that will used to update a parameter.
- That depends on Mean and the Variance of the parameters
- Step size have two upper bounds

$$|\Delta_t| \leq \alpha \cdot (1 - \beta_1) / \sqrt{1 - \beta_2} \text{ in the case } (1 - \beta_1) > \sqrt{1 - \beta_2},$$

and

$$|\Delta_t| \leq \alpha$$

The First Case: For severe case of sparsity.

The gradient has been zero for many timesteps but not current one

The effective step size is higher

The Second Case: Less Sparse cases

The effective step size is lower

Best Default values by the Authors

α - 0.001

$\beta1$ - 0.9

$\beta2$ - 0.999

ϵ - 10^{-8}

INITIALIZATION BIAS CORRECTION

- Focused on the Initial steps of the algorithm

$$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \text{ (Update biased first moment estimate)}$$
$$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \text{ (Update biased second raw moment estimate)}$$

- Initial steps m(t-1) and v(t-1) values are almost zero (m0 = 0, v0 = 0)

- mt and vt are heavily biased to (1- **β**).gt on initial algorithm

$$\widehat{m}_t \leftarrow m_t/(1 - \beta_1^t) \text{ (Compute bias-corrected first moment estimate)}$$
$$\widehat{v}_t \leftarrow v_t/(1 - \beta_2^t) \text{ (Compute bias-corrected second raw moment estimate)}$$

- Correcting the moving average by removing the bias from moving average

Related Work

Optimization methods bearing a direct relation to Adam are RMSProp (Tieleman & Hinton, 2012; Graves, 2013) and AdaGrad (Duchi et al., 2011)

Other stochastic optimization methods

- vSGD (Schaul et al., 2012)
- AdaDelta (Zeiler, 2012)
- Natural Newton method from Roux & Fitzgibbon (2010)
- Sum-of-Functions Optimizer (SFO) (Sohl-Dickstein et al., 2014)

Related Work

1.RMSProp (Tieleman & Hinton, 2012)

- Optimization method closely related to Adam
- RMSProp - generates its parameter updates using a momentum on the rescaled gradient
- Adam - updates are directly estimated using a running average of first and second moment of the gradient
- Lacks a bias-correction term - leads to very large stepsizes and often divergence

2. AdaGrad (Duchi et al., 2011)

-An algorithm that works well for sparse gradients

-Decay the learning rate for parameters in proportion to their update history (more updates means more decay).

-Its basic version updates parameters as,

$$v_t^w = v_{t-1}^w + (\nabla w_t)^2$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t^w + \epsilon}} * \nabla w_t$$

$$v_t^b = v_{t-1}^b + (\nabla b_t)^2$$

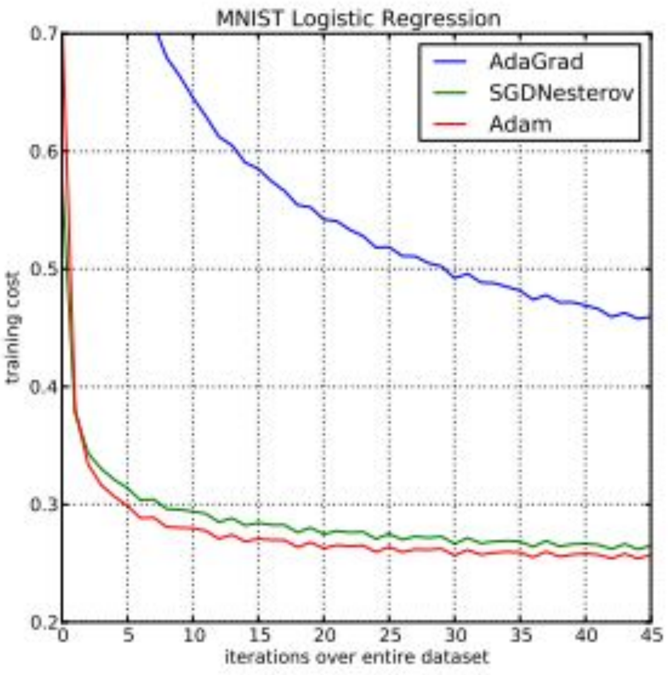
$$b_{t+1} = b_t - \frac{\eta}{\sqrt{v_t^b + \epsilon}} * \nabla b_t$$

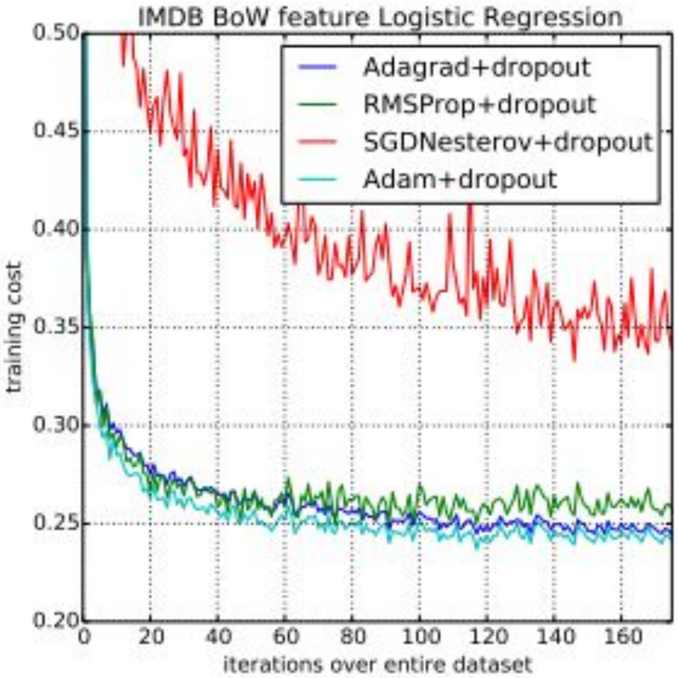
Evaluation of Adam

1. LOGISTIC REGRESSION

- L2-regularized multi-class logistic regression using the MNIST dataset.
- Compare Adam to accelerated SGD with Nesterov momentum and Adagrad using minibatch size of 128
- Adam yields similar convergence as SGD with momentum and both converge faster than Adagrad.
- Examine the sparse feature problem using IMDB movie review dataset from (Maas et al., 2011).

Logistic regression training on MNIST images and IMDB movie reviews dataset.





E/15/249-Dasuni

2. MULTI-LAYER NEURAL NETWORKS

a neural network model - two fully connected hidden layers, 1000 hidden units each, ReLU activation, minibatch size of 128.

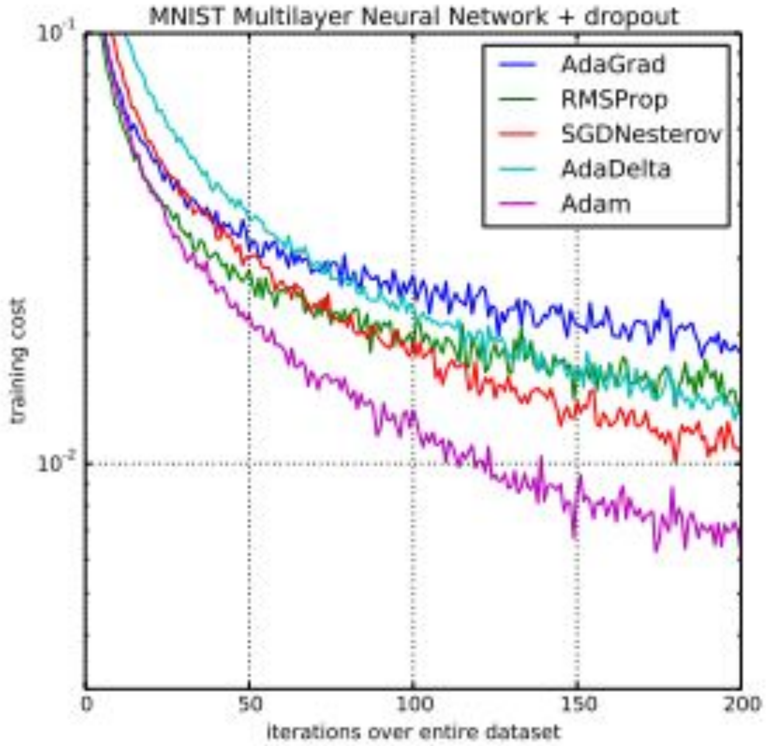
The sum-of-functions (SFO) method (Sohl-Dickstein et al., 2014)

Adam makes faster progress

Other stochastic first order methods on multi-layer neural networks trained with dropout noise.

Adam shows better convergence than other methods.

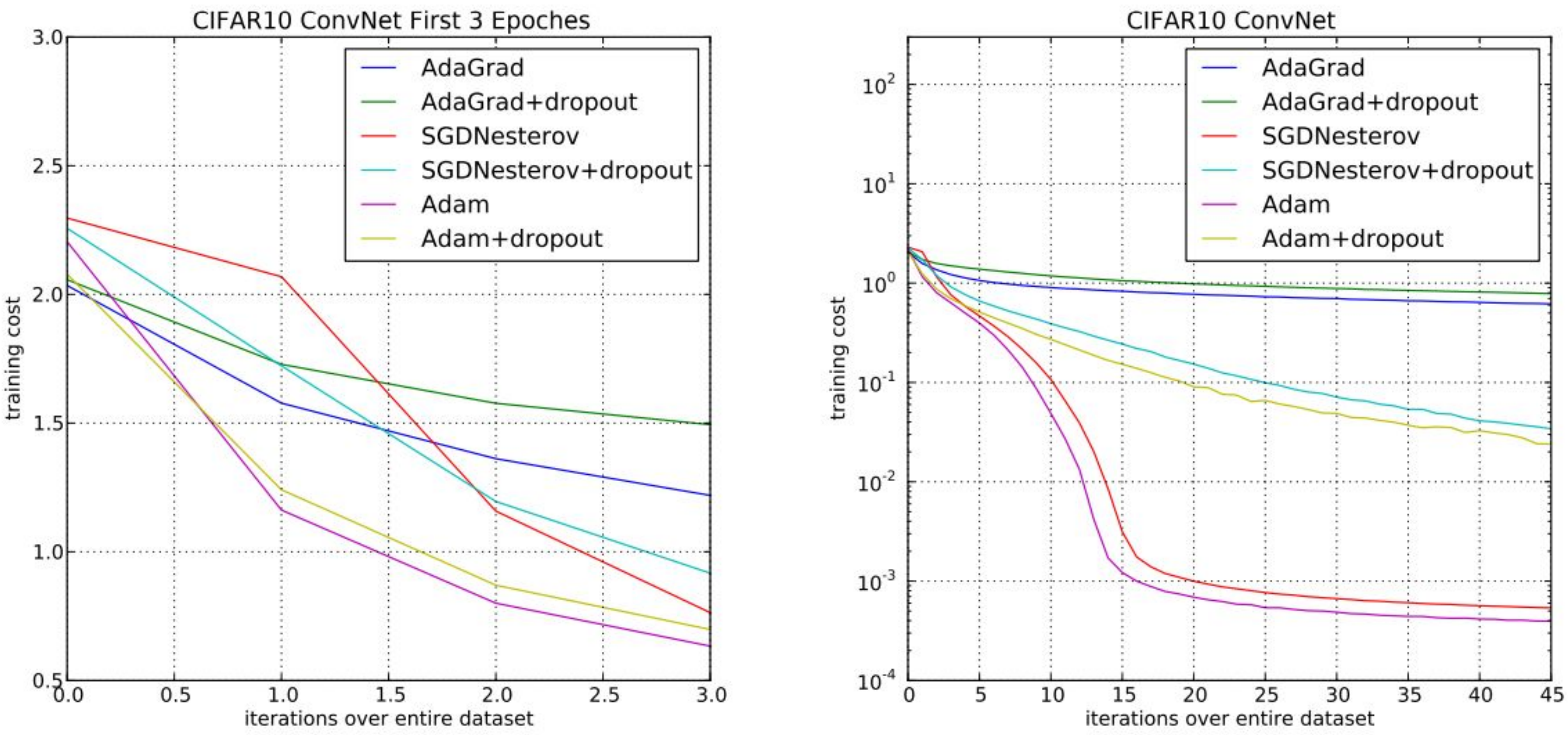
Multilayer neural networks on MNIST images using dropout stochastic regularization.



CNN architecture

- 3 alternating stages of 5x5 convolution filters.
- 3x3 max pooling with stride of 2.
- a fully connected layer of 1000 RLUs.
- minibatch size is 128.

Convolutional neural networks training cost



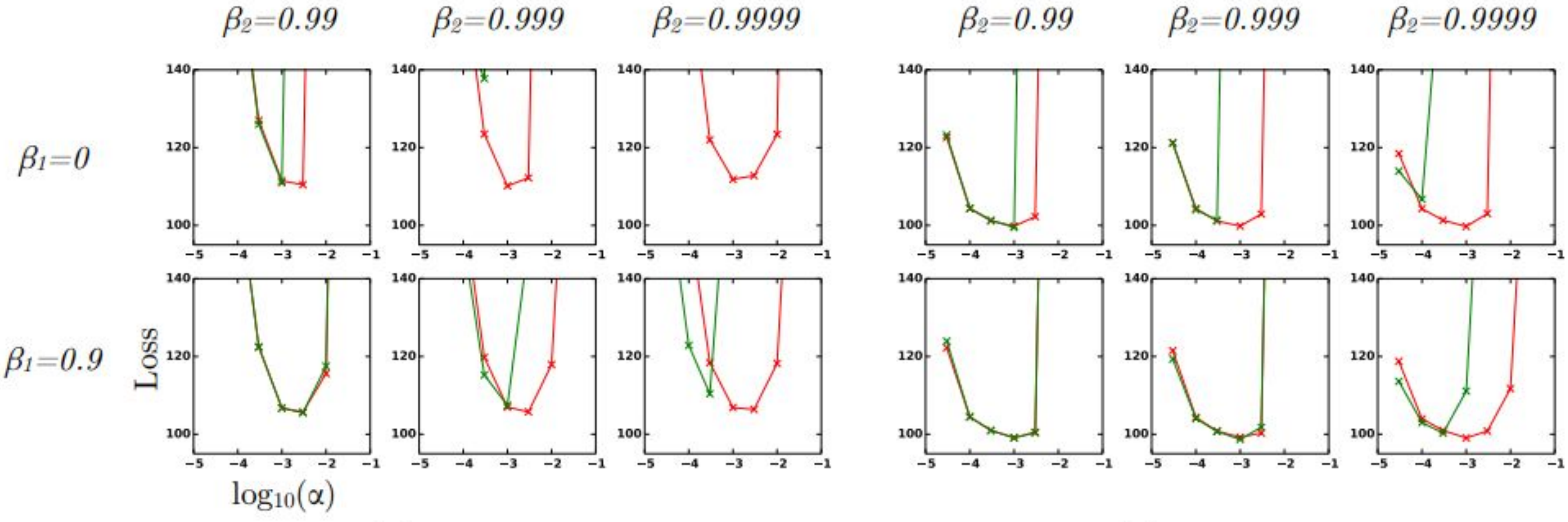
ADAM WITH CNNS

- $\widehat{v_t}$ vanishes to zeros after a few epochs.
- First moment is more important in CNNs, contributes to the speed-up.
- CNNs have vastly different gradients in different layers.
- Adam adapts learning rate scale for different layers instead of hand picking manually as in SGD.

4. BIAS-CORRECTION TERM

- Vary the $\beta 1$ and $\beta 2$ when training a variational autoencoder (VAE)
- Broad range of hyper-parameter choices.
- For robustness to sparse gradients
    - values of $\beta 2$ close to 1
    - results in larger initialization bias
    - bias correction term is important

Bias-correction terms vs no bias correction terms



(a) after 10 epochs

(b) after 100 epochs

E/15/139-Sajini

Bias-correction terms vs no bias correction terms

- Values $\beta 2$ close to 1 with no bias correction term - lead to instabilities
- Best results - small values of $(1-\beta 2)$ and with bias correction term
- Removal of the bias correction terms results in a version of RMSProp with momentum
- Adam performed equal or better than RMSProp, regardless of hyper-parameter setting.

Extensions of Adam

1. ADAMAX

- A variant of Adam based on the infinity norm
- Adam : generalization of L 2 norm.
- Adamax : generalization of L infinity norm.
- Simple and stable algorithm
- Better than Adam
  - data that is noisy in terms of gradient updates.
  - models with embeddings.

2. TEMPORAL AVERAGING

- Last iterate is noisy due to stochastic approximation
- Better generalization performance is often achieved by averaging.
- Exponential moving average over the parameters, giving higher weight to more recent parameter values.

$$\bar{\theta}_t \leftarrow \beta_2 \cdot \bar{\theta}_{t-1} + (1 - \beta_2)\theta_t, \text{ with } \bar{\theta}_0 = 0$$

$$\widehat{\theta}_t = \bar{\theta}_t / (1 - \beta_2^t)$$

Summary

- Optimization algorithm for stochastic gradient descent
- Combines the best properties of the AdaGrad and RMSProp algorithms
- Can handle
  - large datasets and/or high-dimensional parameter spaces
  - problems with very noisy and/or sparse gradients.
- Works well in practice and compares favorably to other stochastic optimization methods.
  - default configuration parameters do well on most problems.

Thank You !

QnA