Adam: A Method for Stochastic Optimization

ICLR 2015

Junmyeong Lee,
EECS of GIST College

Main problem : Stochastic optimization

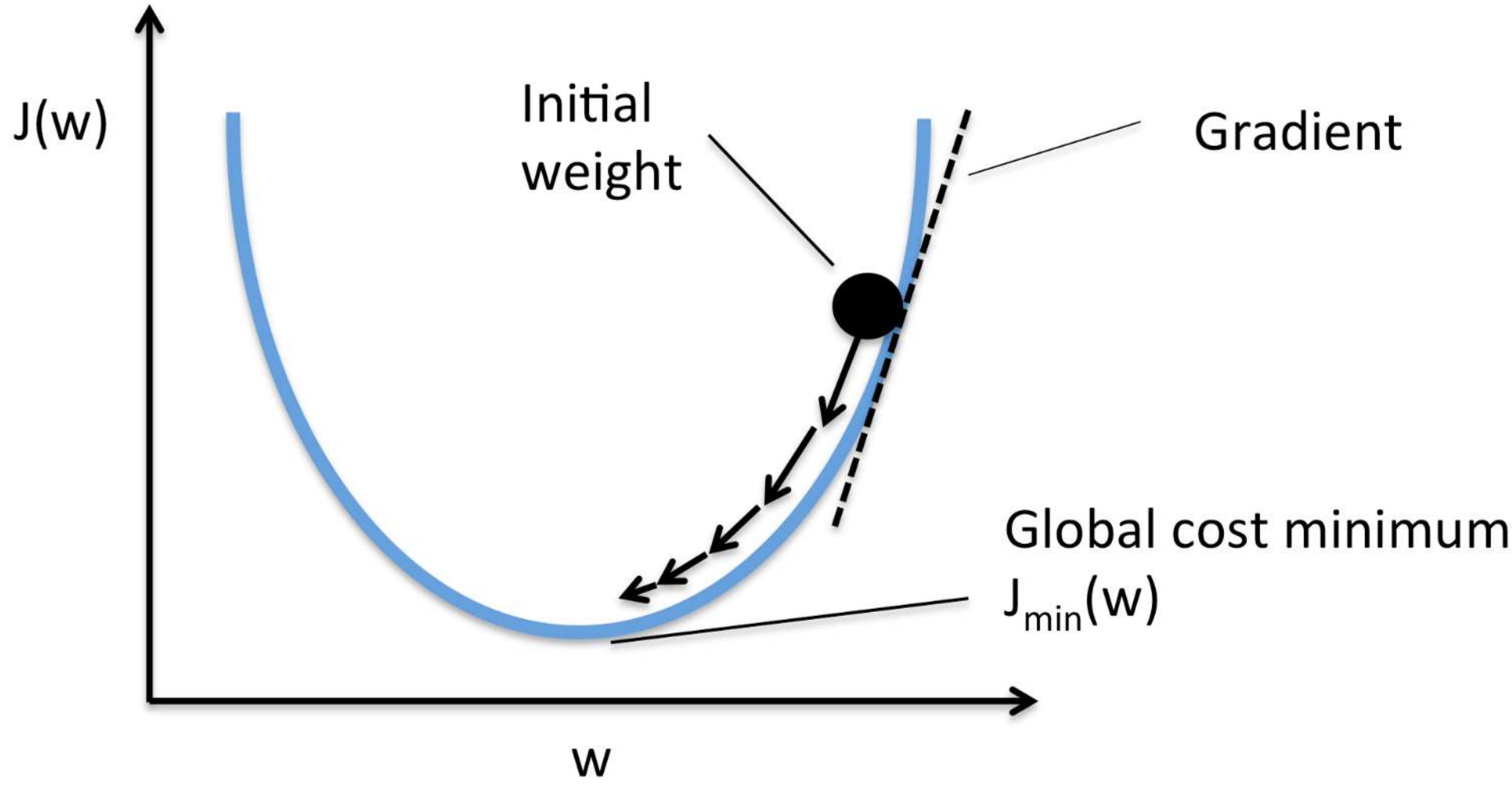
- methods for $\underbrace{\text{minimizing or maximizing an objective function}}_{\text{Task-specific function}}$ when $\underline{\text{randomness}}$ is present.

For stochastic objective function : $f(x) + \epsilon$, $(\because \text{in general, suppose } \epsilon \sim N(0,1))$

- Stochastic approximation
- Stochastic gradient descent
- Simultaneous perturbation
- Scenario optimization
- Monte-carlo sampling(simulation)

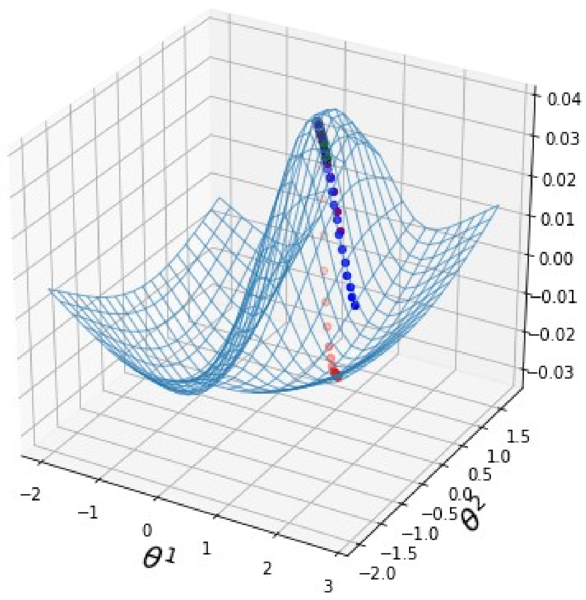
Main problem : SGD

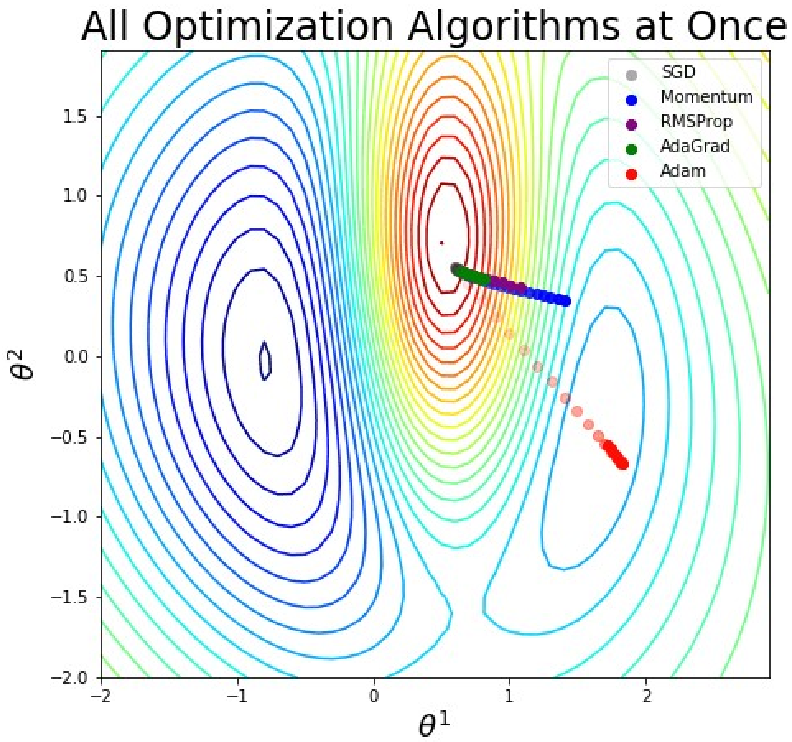
- Convex



Main problem : SGD

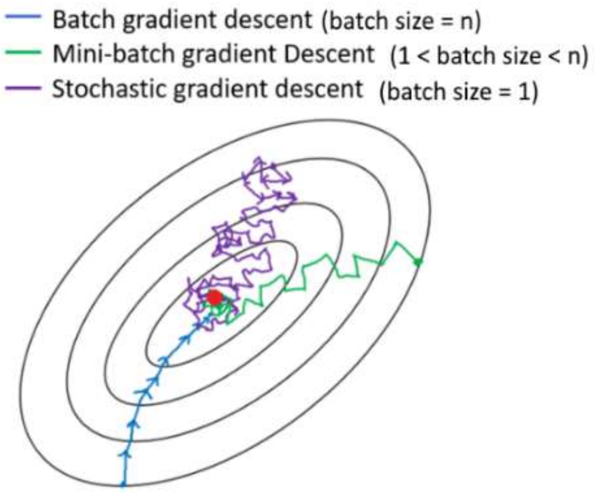
- Non-Convex

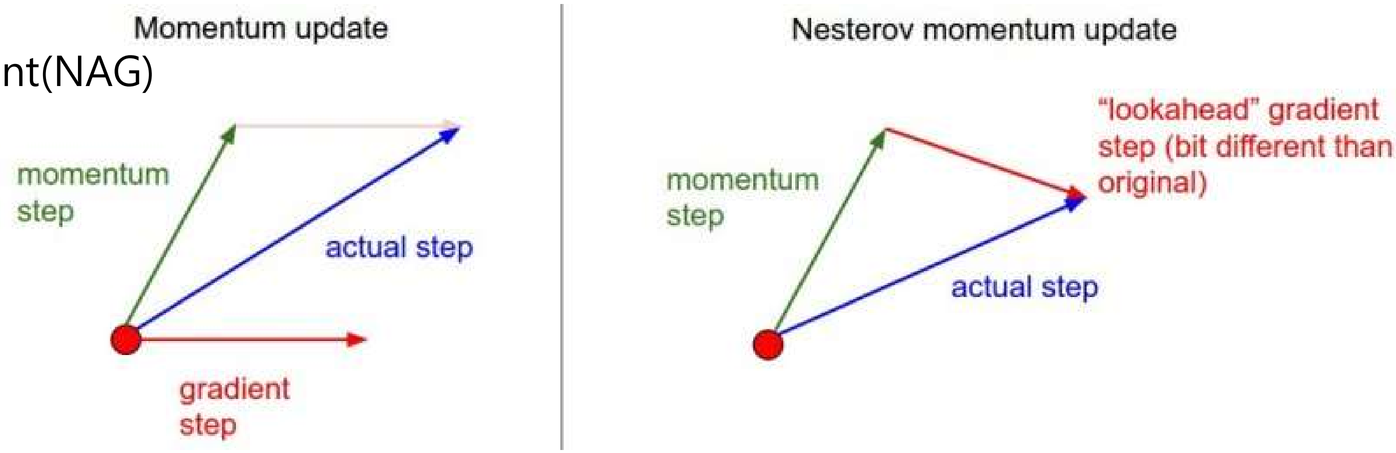




Previous Researches

- Vanilla gradient descent
  - Batch Gradient Descent
  - Stochastic Gradient Descent
  - Mini-Batch Gradient Descent
- Momentum
  - Momentum SGD
  - Nesterov Accelarated Gradient(NAG)
- Ada- (adapt LR rate)
  - AdaGrad
  - RMSProp
  - Adam





Previous Researches

- AdaGrad

$$\mathbf{h} \leftarrow \mathbf{h} + \frac{\partial L}{\partial \mathbf{W}} \odot \frac{\partial L}{\partial \mathbf{W}}$$

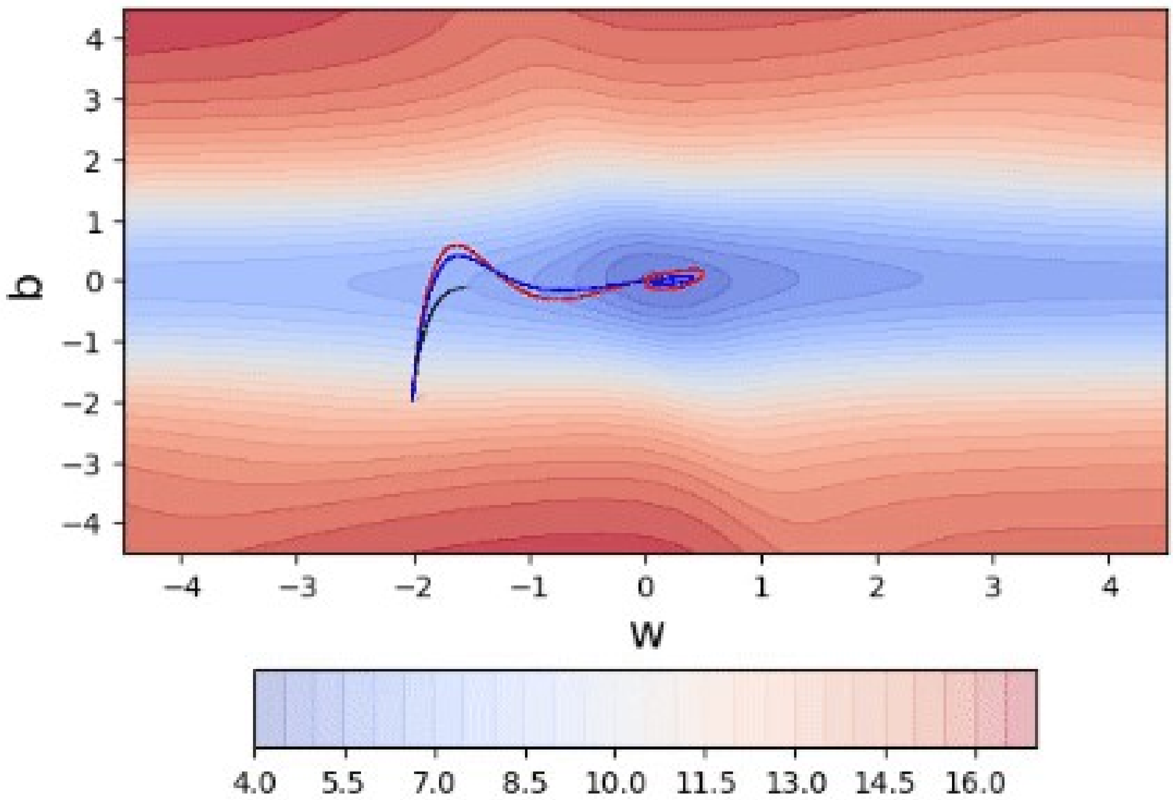
$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{1}{\sqrt{\mathbf{h}}} \frac{\partial L}{\partial \mathbf{W}}$$

- RMSProp

$$\mathrm{G(t)} = \gamma \mathrm{G(t-1)} + (1-\gamma)\mathrm{g_t^2}$$

$$\mathrm{W(t+1)} = \mathrm{W(t)} - \alpha \cdot \frac{1}{\sqrt{\mathrm{G(t)} + \epsilon}} \mathrm{g_t}$$

Previous Researches : AdaGrad

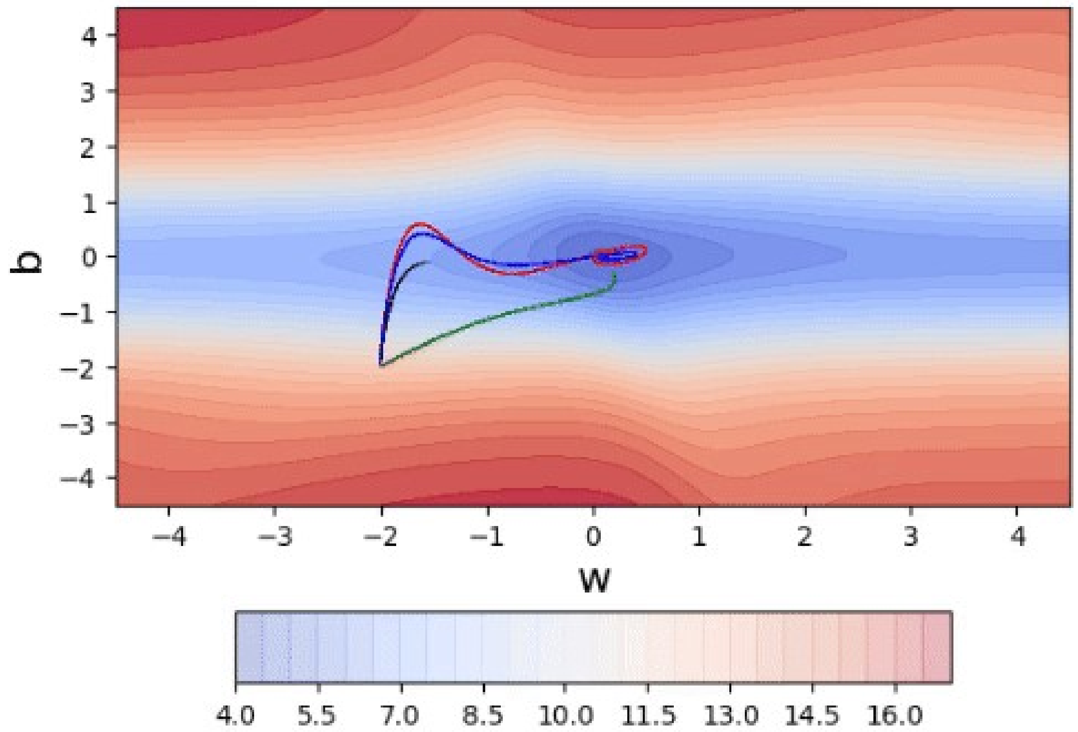


- Well-performed with sparse gradient condition (by denominator)
- But effective learning rate is overly suppressed over step.

$$\mathbf{h} \leftarrow \mathbf{h} + \frac{\partial L}{\partial \mathbf{W}} \odot \frac{\partial L}{\partial \mathbf{W}}$$

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \frac{1}{\sqrt{\mathbf{h}}} \frac{\partial L}{\partial \mathbf{W}}$$

Previous Researches : RMSProp



- Move similar to AdaGrad
- Solved aggressive learning rate decay (by EMA)

Previous Researches : Exponential Moving Average

Simple moving average

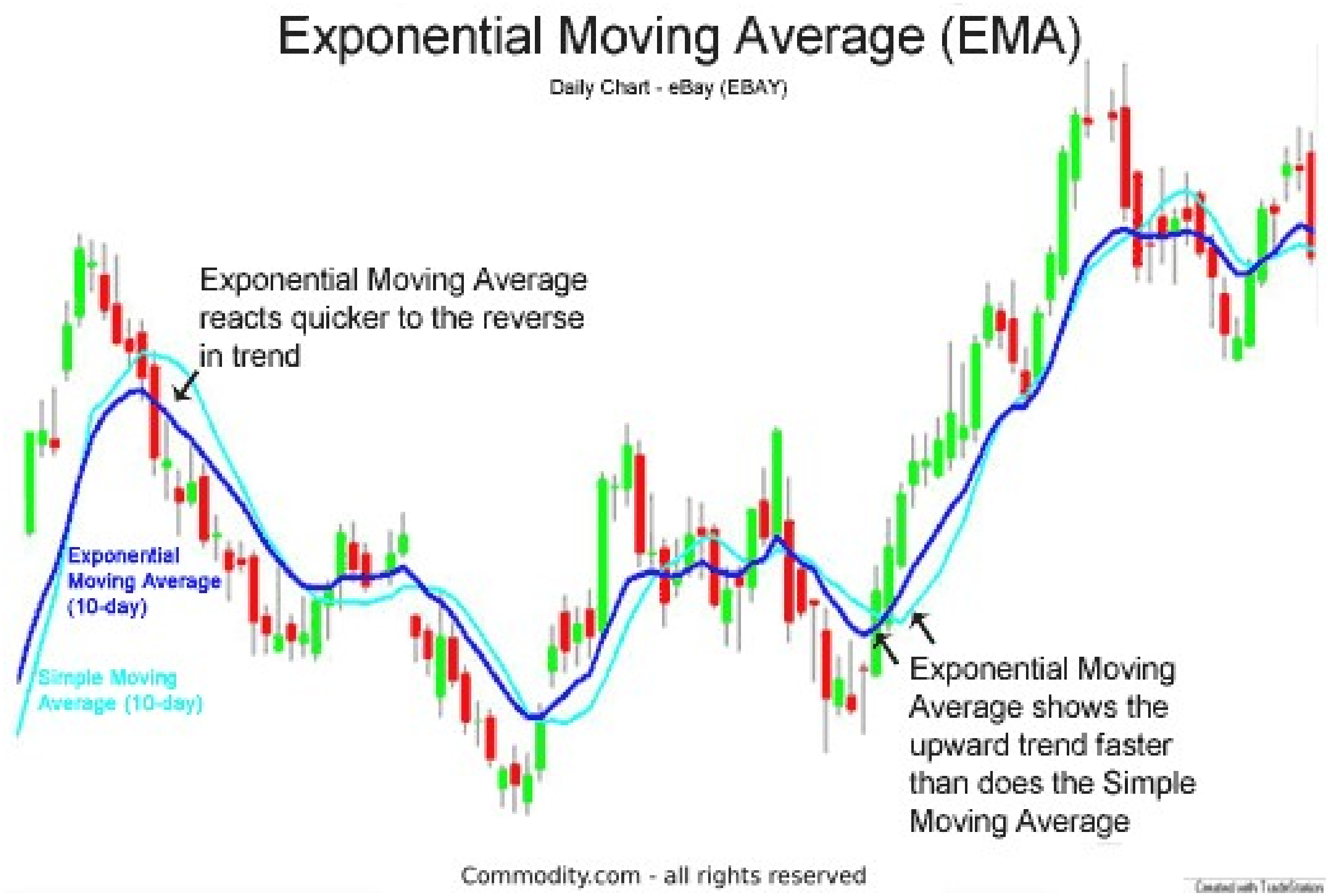
$$\bar{p}_{\mathrm{SM}} = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

Exponential moving average

$$\boxed{S_t = \begin{cases} Y_1, & t = 1 \\ \alpha \cdot Y_t + (1-\alpha) \cdot S_{t-1}, & t > 1 \end{cases}}$$

close to the recent trend!

Previous Researches : Exponential Moving Average



Main Algorithm : Adam

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 1$^{\text{st}}$ moment vector)
  $v_0 \leftarrow 0$ (Initialize 2$^{\text{nd}}$ 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)
    } Exponential moving average
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    } Bias correction
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Main Algorithm : Adam

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\longrightarrow$$

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

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Initialization Bias Correction

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

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

Measurement Bias
(approximation bias)

Initialization Bias Correction

We can change this bias by change β_2

$$E[\hat{v}_t] = \frac{E[v_t]}{1-\beta_2^t} = E[g_t^2] + \frac{\zeta}{1-\beta_2^t}$$

Remove initialization bias

Convergence Analysis

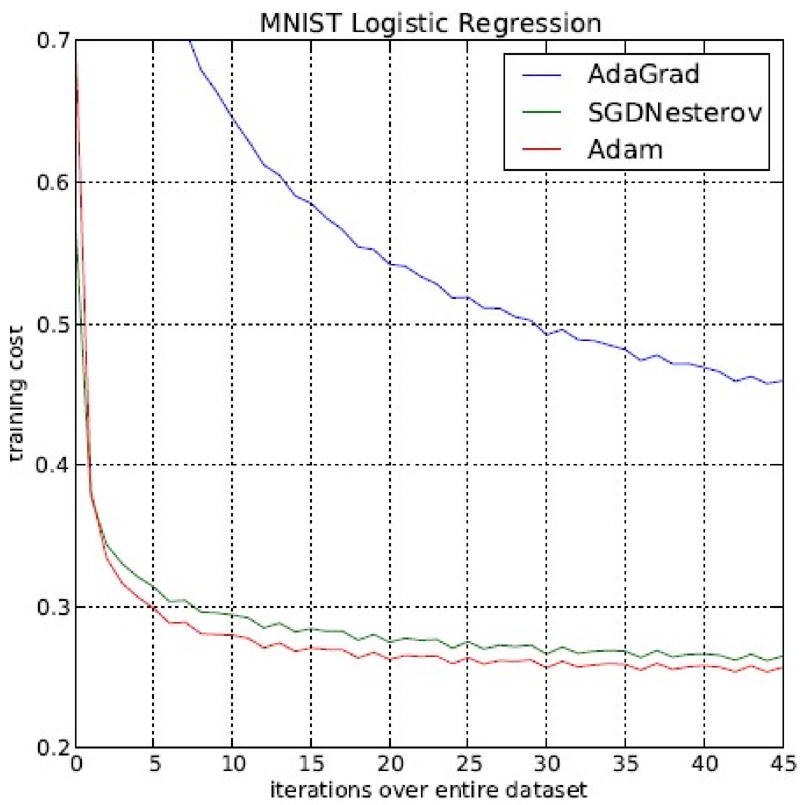
$$R(T) = \sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)]$$

Corollary 4.2. *Assume that the function f_t has bounded gradients, $\|\nabla f_t(\theta)\|_2 \leq G$, $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ for all $\theta \in R^d$ and distance between any θ_t generated by Adam is bounded, $\|\theta_n - \theta_m\|_2 \leq D$, $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ for any $m, n \in \{1, ..., T\}$. Adam achieves the following guarantee, for all $T \geq 1$.*

$$\frac{R(T)}{T} = O(\frac{1}{\sqrt{T}})$$

This result can be obtained by using Theorem 4.1 and $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 \leq dG_\infty \sqrt{T}$. Thus, $\lim_{T\to\infty} \frac{R(T)}{T} = 0$.

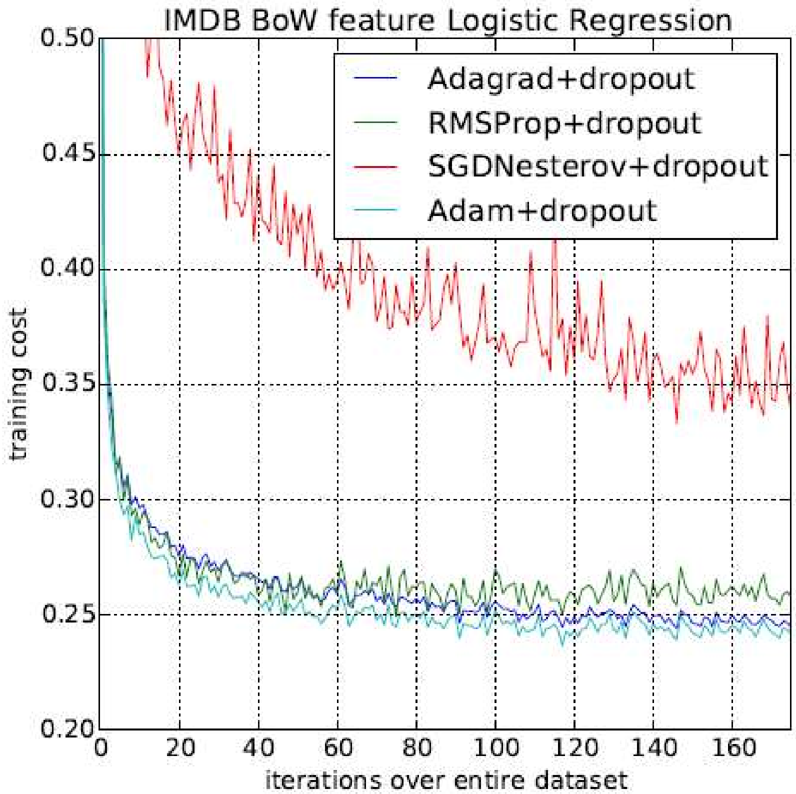
Experiment : Logistic Regression(Convex)



- SGD + Nesterov momentum outperform AdaGrad with large margin

- Adam has similar tendency compare to SGD + Nesterov setup

Experiment : Logistic Regression(Convex)



- Under sparse feature(gradient) condition, Adagrad has better performance than SGD + momentum
- Adam has better performance than others.

Experiment : Logistic Regression(Convex)

- What is sparse gradient?

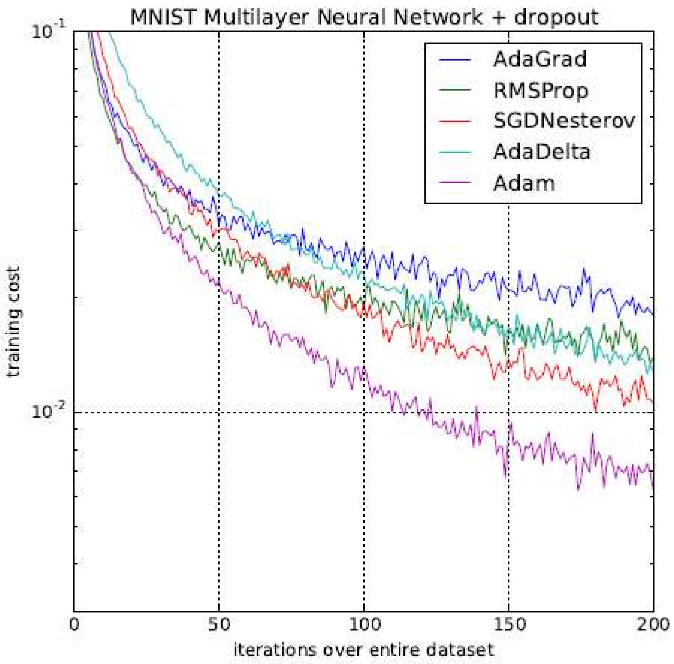
Shrey Desai, BS Computer Science, The University of Texas at Austin

Answered May 2, 2019 · Author has **186** answers and **749.7K** answer views

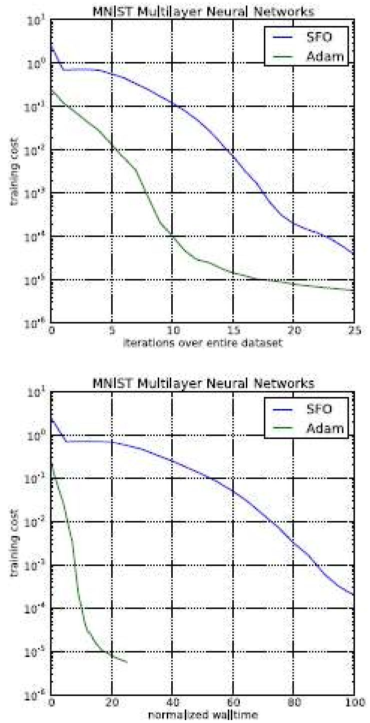
In the context of deep learning, sparse gradients imply a network is not receiving strong enough signals to tune its weights. At a high level, a neural network can be thought of as a region in a (very) high-dimensional parameter space. Gradients (usually obtained through backpropagation) allow this region to flexibly move around until it settles in an area that achieves reasonable performance on a downstream task. If the training signal is weak, then the parameters cannot be tuned effectively.

The vanishing gradient problem is a great example of sparse gradients. This problem is ubiquitous in deep learning, but it is often associated with recurrent neural networks (RNNs). Vanilla RNNs in particular undergo successive multiplications of matrices during backpropagation, which causes a lot of numerical instability. In other words, multiplying small numbers over and over again result in even smaller numbers. The resulting (very) small numbers — the gradient — do not provide enough information to the network to update its weights.

Experiment : MLP(Non-Convex)



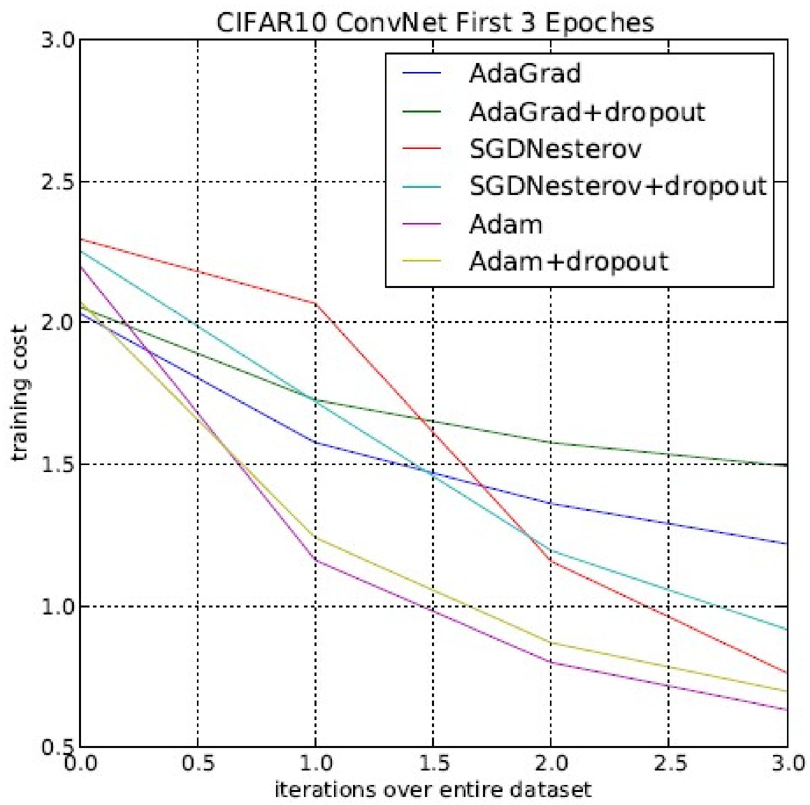
(a)

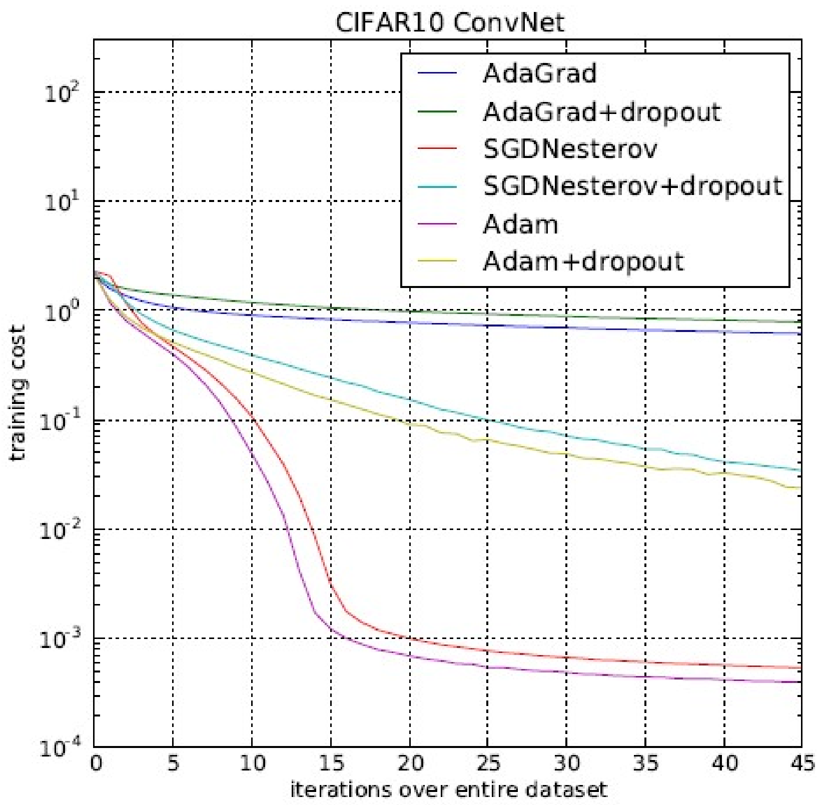


(b)

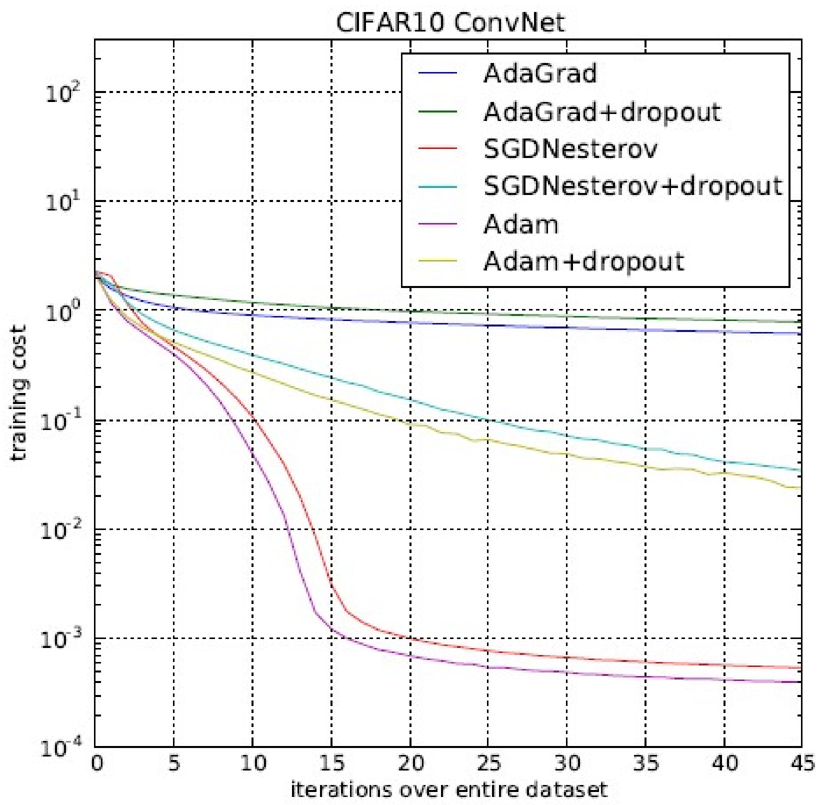
- Under Non-Convex condition, Adam shows good performance
- Adam outperforms other conventional methods about deterministic function

Experiment : CNN(poor geometry)



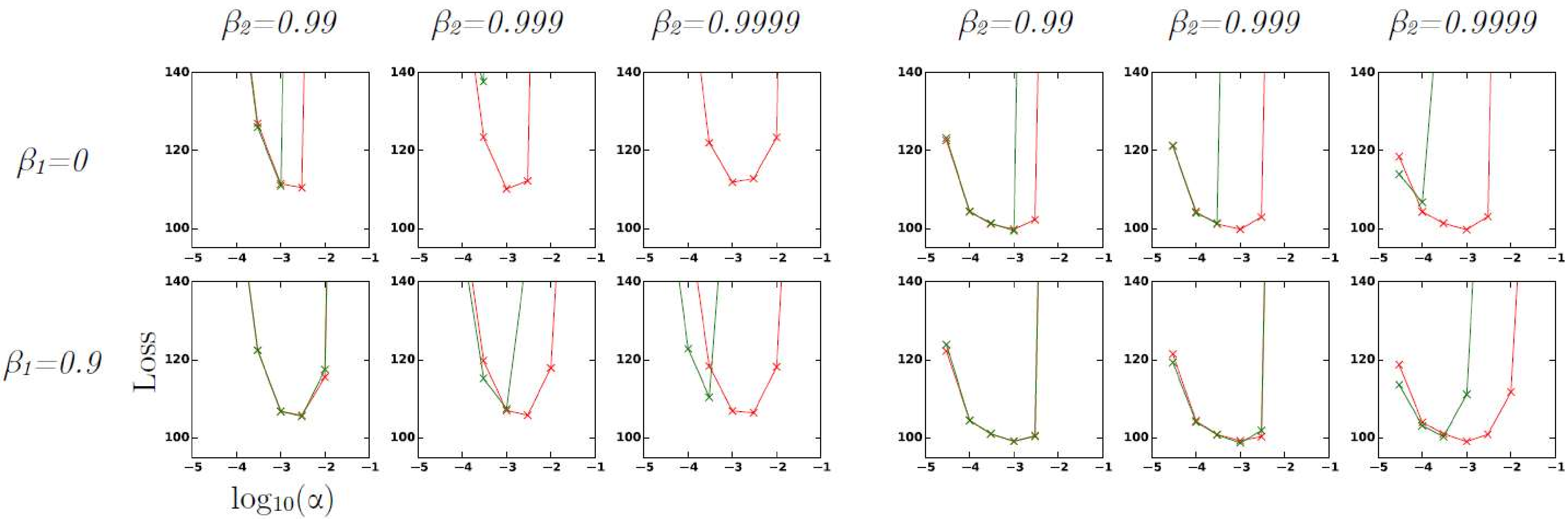


Experiment : CNN(poor geometry)



- Adam, SGD converge rapidly!
- With cost function in CNN, we have poor approximation!
- In this case, we should cancel out the variance of mini-batch by using first momentum
- We should consider best optimizer for various tasks (except Adam)

Experiment : Bias-Correction term



(a) after 10 epochs (b) after 100 epochs

α = stepsize

Red : bias-correction
Green : no bias-correction

Conclusion

- Adam

  new efficient algorithm for gradient-based optimization of stochastic objective functions

- AdaGrad + RMSProp

- Robust and well-suited to a wide range of non-convex optimization problems