ADAM : A Method for Stochastic Optimization in ICLR 2015

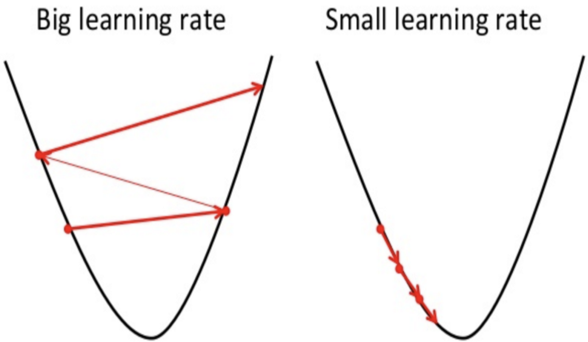
By Diederik P. Kingma Jimmy Lei Ba

Anshika Chaurasia
Razat Shanker

EE18MTECH11017
EE18MTECH11016

4 Mar 2019

Motivation



- Learning rate affect model performance.
- We want cost function sensitive to some directions in parameter space and insensitive to others

Related Work

AdaGrad

Learning rate scaled by inverse of square of sum of all the previous squared values of the gradient

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)

$v_t \leftarrow v_{t-1} + g_t^2$ (update biased second moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} \odot g_t (updateParameter)$

end while

return θ_t (resulting parameter)

Related Work

RMSprop

Uses an exponentially weighted moving average to discard history from the extreme past

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)

$v_t \leftarrow \beta_1 v_{t-1} + (1-\beta_1) g_t^2$ (update biased second moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} \odot g_t (updateParameter)$

end while

return θ_t (resulting parameter)

Proposed Work

ADAM

- It is a combination of RMSprop and SGD with momentum.
- It uses the squared gradients to scale the learning rate like RMSprop
- It takes advantage of momentum by using moving average of the gradient like SGD with momentum

Algorithm

g_t^2 indicates the element wise square $g_t \odot g_t$
$\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$
Require : α : *Stepsize*
Require : $\beta_1, \beta_2 \epsilon$ [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^{st} moment vector)
$v_0 \leftarrow 0$ (Initialize 2^{nd} moment vector)
$t \leftarrow 0$ (initialize timestep)

Algorithm

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)

$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1).g_t$ (update biased first moment estimate)

$v_t \leftarrow \beta_2.v_{t-1} + (1-\beta_2).g_t^2$ (update biased second moment estimate)

$m_t^{'} \leftarrow m_t/(1-\beta_1^t)$ (compute bias-corrected first moment estimate)

$v_t^{'} \leftarrow v_t/(1-\beta_2^t)$ (compute bias-corrected second raw moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \alpha.m_t^{'}/(\sqrt{v_t^{'}} + \epsilon)$ (update parameters)

end while

return θ_t (resulting parameter)

Experiment

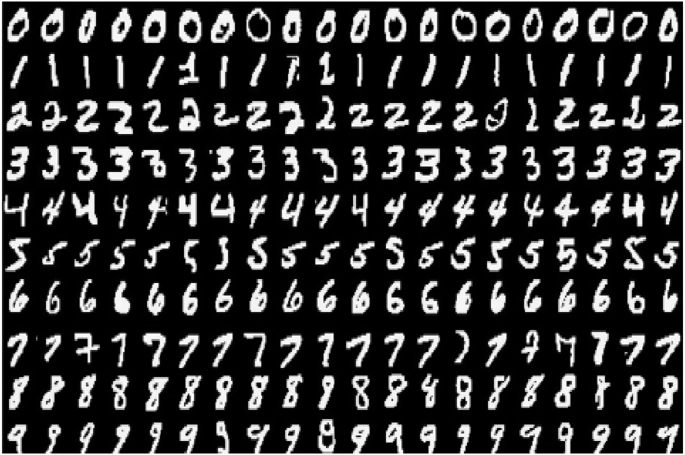


Fig : MNIST Dataset

Results

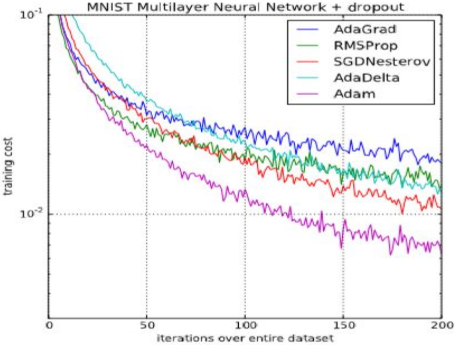
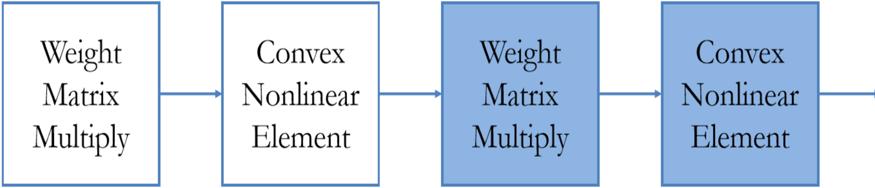


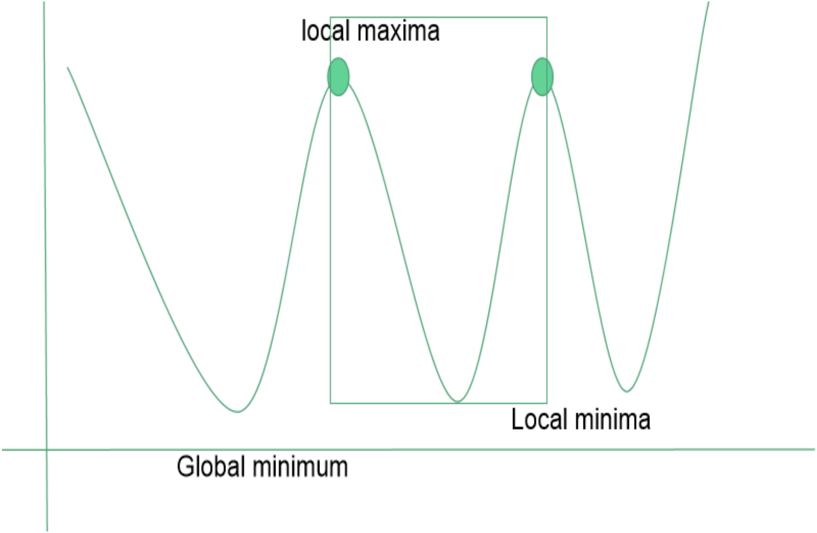
Fig : Multilayer Neural Network on MNIST images

Why are neural networks non - convex ?



- composition of convex function is not convex
- Neural Networks are universal function approximators
- Convex functions can't approximate non - convex functions

Local Optimization of non convex functions



References

[1] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of Machine Learning Research 12.Jul (2011): 2121-2159.
[2] Sutskever, I., Martens, J., Dahl, G. E., Hinton, G. E. (2013). On the importance of initialization and momentum in deep learning. ICML (3), 28(1139-1147), 5.
[3] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).