A Blog From a Human-engineer-being

CODING AS ENGINNER...

COMPARISON: SGD VS MOMENTUM VS RMSPROP VS MOMENTUM+RMSPROP VS ADAGRAD

FEBRUARY 13, 2015 | EREN | 9 COMMENTS

In this post I'll briefly introduce some update tricks for training of your ML model. Then, I will present my empirical findings with a linked NOTEBOOK that uses 2 layer Neural Network on CIFAR dataset.

I assume at least you know what is Stochastic Gradient Descent (SGD). If you don't, you can follow this tutorial . Beside, I'll consider some improvements of SGD rule that result better performance and faster convergence.

SGD is basically a way of optimizing your model parameters based on the gradient information of your loss function (Means Square Error, Cross-Entropy Error ...). We can formulate this;

$w(t) = w(t-1) - \epsilon * \triangle w(t)$

w is the model parameter, ϵ is learning rate and $\triangle w(t)$ is the gradient at the time t.

SGD as itself is solely depending on the given instance (or the batch of instances) of the present iteration. Therefore, it tends to have unstable update steps per iteration and corollary convergence takes more time or even your model is akin to stuck into a poor local minima.

To solve this problem, we can use Momentum idea (Nesterov Momentum in literature). Intuitively, what momentum does is to keep the history of the previous update steps and combine this information with the next gradient step to keep the resulting updates stable and conforming the optimization history. It basically, prevents chaotic jumps. We can formulate Momentum technique as follows;

$v(t) = \alpha v(t-1) - \epsilon \frac{\partial E}{\partial w}(t)$ (update velocity history with the new gradient)

$\triangle w(t) = v(t)$ (The weight change is equal to the current velocity)

α is the momentum coefficient and 0.9 is a value to start. $\frac{\partial E}{\partial w}(t)$ is the derivative of w wrt. the loss.

Okay we now soothe wild SGD updates with the moderation of Momentum lookup. But still nature of SGD proposes another potential problem. The idea behind SGD is to approximate the real update step by taking the average of the all given instances (or mini batches). Now think about a case where a model parameter gets a gradient of +0.001 for each instances then suddenly it gets -0.009 for a particular instance and this instance is possibly a outlier. Then it destroys all the gradient information before. The solution to such problem is suggested by G. Hinton in the Coursera course lecture 6 and this is an unpublished work even I believe it is worthy of. This is called RMSprop. It keeps running average of its recent gradient magnitudes and divides the next gradient by this average so that loosely gradient values are normalized. RMSprop is performed as below;

$MeanSquare(w,t) = 0.9 MeansSquare(w,t-1) + 0.1 \frac{\partial E}{\partial w}(t)^2$

$\triangle w(t) = \epsilon \frac{\partial E}{\partial w}(t) / (\sqrt{MeanSquare(w,t)} + \mu)$

μ is a smoothing value for numerical convention.

You can also combine Momentum and RMSprop by applying successively and aggregating their update values.

Lets add AdaGrad before finish. AdaGrad is an Adaptive Gradient Method that implies different adaptive learning rates for each feature. Hence it is more intuitive for especially sparse problems and it is likely to find more discriminative features and filters for your Convolutional NN. Although you provide an initial learning rate, AdaGrad tunes it regarding the history of the gradients for each feature dimension. The formulation of AdaGrad is as below;

$w_i(t) = w_i(t-1) + \frac{\epsilon}{\sum_{k=1}^{t} \sqrt{g_{ki}}^2}$ where $g_{ki} = \frac{\partial E}{\partial w_i}$

So the upper formula states that, for each feature dimension, learning rate is divided by the all the squared root gradient history.

Now you completed my intro to the applied ideas in this NOTEBOOK and you can see the practical results of these applied ideas on CIFAR dataset. Of course this into does not mean complete by itself. If you need more refer to other resources. I really suggest the Coursera NN course by G. Hinton for RMSprop idea and this notes for AdaGrad.

For more information you can look this great lecture slide from Toronto Group.

Lately, I found this great visualization of optimization methods. I really suggest you to take a look at it.

G+1 Like 0 Tweet Share

Related posts:

1. Brief History of Machine Learning
2. Some possible ways to faster Neural Network Backpropagation Learning #1
3. What is special about rectifier neural units used in NN learning?
4.
5. Microsoft Research introduced a new NN model that beats Google and the others

DEEP LEARNING • MACHINE LEARNING • MOMENTUM • NEURAL NETWORK • OPTIMIZATION • SGD

9 Comments erogol.com 1 Login

Recommend Share Sort by Best

Join the discussion…

PengPai · 7 months ago
Hey, thanks for your nice sharing. I have a simple question. What do you exactly mean by a method named Momentum + RMSprop? Is such kind of method updated by an average of delta-w ?
Reply · Share ›

erogol Mod → PengPai · 7 months ago
This is basically using momentum and RSM together. First you find the update step by momentum history then you find RMS update step exclusively then aggregate these two before updating the model parameters. You can refer to the "lecture slide" that are linked at the end of the post.
Reply · Share ›

PengPai → erogol · 7 months ago
The "lecture slide" seems to introduce the two specific methods separately. Which page do you refer to ? Besides, does the "aggregation" mean a average of the two "gradients" computed by momentum and rmsprop ?
Reply · Share ›

erogol Mod → PengPai · 7 months ago
The last section of the slide is related as I remember. Also you can look the lecture 6 of G. Hinton's Coursera Class.
Aggregate means summing up the update steps in my definition. Best :)
Reply · Share ›

anon · 7 months ago
I cannot open the Notebooks. I get
Error 503 No healthy backends
Reply · Share ›

erogol Mod → anon · 7 months ago
maybe it was an server error
1 Reply · Share ›

erogol Mod → anon · 7 months ago
I dont see why. I can open it in all of my devices.

I dont see why. I can open it in all of my devices

^ | ˅ • Reply • Share ›

Peter Roelants ➦ erogol • 3 months ago

I have the same Issue, I get an "Error establishing a database connection" on all links from http://www.erogol.com/.
GET http://www.erogol.com/ 500 (Internal Server Error)

^ | ˅ • Reply • Share ›

erogol Mod ➦ Peter Roelants • 3 months ago

I solved it for the time

^ | ˅ • Reply • Share ›

ALSO ON **EROGOL.COM**

WHAT'S THIS?

A Large set of Machine Learning Resources for Beginners to Mavens

6 comments • a year ago

Ke Sang — Great job and many thanks!!

Machine Learning Work-Flow (Part 1)

4 comments • a year ago

erogol — Hi Dinesh Now I changed the settings as you wish. Thanks for your great comments. * Sanity-Check includes same considerations …

Passing multiple arguments for Python multiprocessing.pool

3 comments • a year ago

Piter — ops def product_helper((a,b)): print a*b

Some Useful Machine Learning Libraries.

1 comment • a year ago

Yannis — Another project that recently came up is YCML (http://github.com/yconst/YCML/..., for those working with Objective-C or Swift, on OS …

✉ Subscribe   Add Disqus to your site   🔒 Privacy