

**WQD7011**

**NUMERICAL OPTIMIZATION**

GROUP ASSIGNMENT

On the Momentum Term in Gradient Descent Learning Algorithms

Instructor: Dr. Ong Sim Ying

Team members:

|  |  |
| --- | --- |
| Aswadi Abdul Rahman | WQD180082 |
| Lee Kwan Li | WQD180019 |
| Chai Kun Ting | WQD180040 |
| Zulkanain Hasan | WQD180031 |

**Chapter 1: Introduction**

* Overview of general neural network model
  + Connectionist neural network model is widely used
  + Most learning algorithms are gradient descent type (back-propagation algorithm)
  + Parameterized error function by weight
* Momentum term is able to increase the rate of convergence by modifying the learning scheme by transforming:

to:

where ρ = momentum parameter

* In the paper, the author tends to:
  + Mathematically analyze the effect of momentum term in speed of learning
  + Demonstrate analogy between momentum term in gradient descent and mass of Newtonian particles in a conservative force field
  + Derive bounds for convergence on ε and ρ and demonstrate momentum term can increase range of ε over system convergence
  + Analyze optimal condition for fastest convergence to minimum

**Chapter 2: Problems**

* Steepest descent is particularly slow when there is a long and narrow valley in the error function surface
  + The gradient is (almost) perpendicular to long axis of valley
  + System oscillates back and forth in direction of short axis, move slowly along long axis
  + Momentum term helps in averaging the oscillation along short axis and add up contribution along long axis
* Conjugate gradient descent also being proposed to improve the rate of convergence. However:
  + It requires more storage of intermediate results
  + It is non-local, as the information needed to update a weight is not all contained in the pre-synaptic and post-synaptic units of the weight
  + It is less biologically plausible and hard to implement in hardware
  + Less robust when the error surface is relatively flat

**Chapter 3: Methods**

* Derivation of Gradient Descent with momentum

o Discretized Newtonian equation is given as:

### 

where m = mass of particle

µ = friction coefficient

w = coordinate vector of particle

After rearrangement it becomes:

* By deriving discretized Newtonian equation and comparing with equation of gradient descent with momentum, it is clearly to see that:

### 

### 

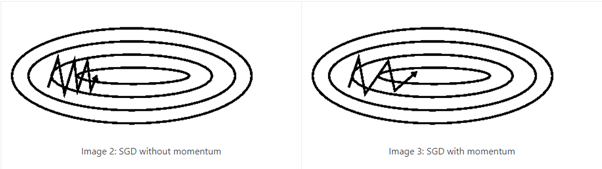
where when mass m = 0 implies ρ = 0 (and vice versa)

* By studying the physical analogy above, it established that momentum method is also stable in continuous case and it is guaranteed to converge to local minimum for any positive learning rate ε and momentum ρ.
* Gradient descent is applying damping oscillation method to accelerate gradient vectors along long axis of valley and leads to faster convergence.

o For each iteration/oscillation, the previous gradient is included in updating next gradient but less weightage for a less recent gradient and higher weightage for a more recent gradient

o Momentum term improves the ordinary gradient descent method by optimizing learning rate across vertical axis but accelerating the learning rate moving along horizontal axis

o Momentum term also increases dimensions whose gradients point in the same directions and reduces updates for dimensions whose gradients change directions, resulting faster convergence with lesser oscillations.



**Chapter 4: Results**

* In this paper, the researcher proved momentum method is guaranteed to converge to a local minimum with 3 results in 2 cases which are continuous time and discrete:
  + As refer to Appendix A, the researcher proved that momentum term improves the speed of convergence for continuous time case with a condition if and only if

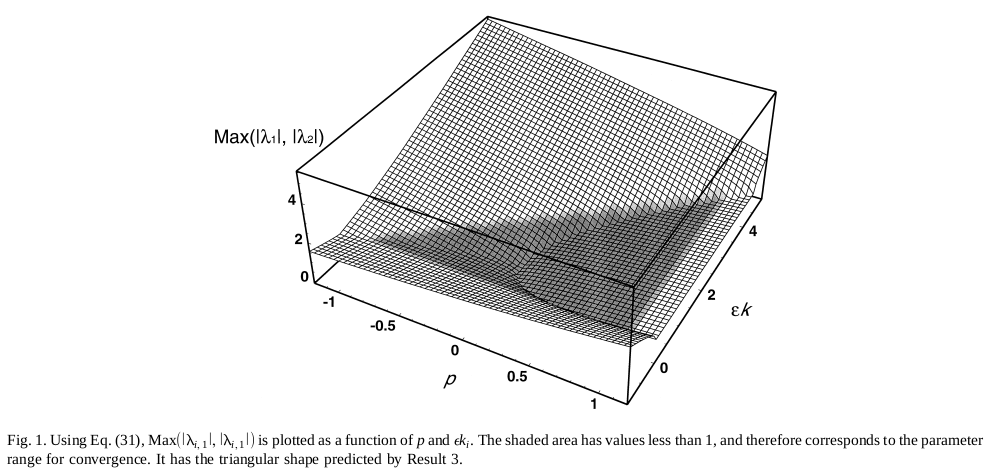
ki < u2/2m

whereas for positive m, u and ki, the inequality

|Reƛi,1| > |Reƛi,0|.

* + Also in continuous time case as shown in Appendix A, the researcher proved that momentum term is most effective when 𝛼 reaches the maximum value or in other words larger 𝛼 means greater improvement.
  + For discrete case (refer to Appendix B), the researcher proved that converges is happen if and only if

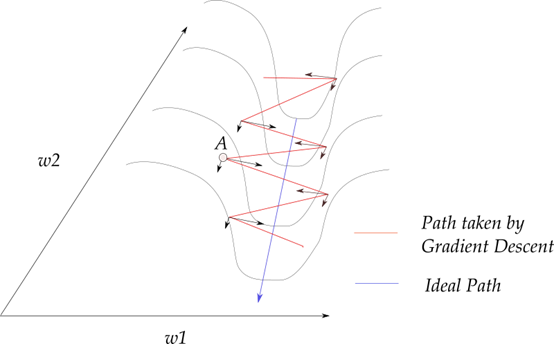
-1 < p < 1 and 0 < eki < 2 + 2p



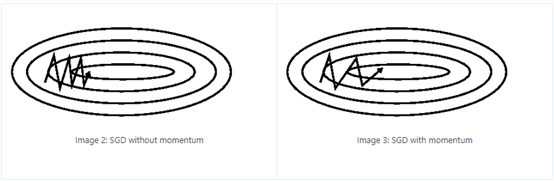
**Chapter 5: Discussion**

1. Advantages and Disadvantages

* Advantages
  + Momentum accumulates the gradient of the past steps to determine the direction to go. It helps to accelerate gradients vectors in the right directions, thus leading to faster converging. The problem with gradient descent is it knows if the loss function is declining but it cannot differentiate between whether the curve is a plane, curve upwards or downwards.



* + Gradient descent takes the zig-zag path and resolved along w1 and w2 directions. Momentum helps restrict the oscillation in one direction, thus it moves more quickly towards the minima.
  + Momentum also works well Stochastic Gradient Descent (SGD). SGD have trouble to navigate around the areas where the surface curves much more steeply in one dimension than in another which make it to hesitate along the bottom towards the local optimum.



* + Momentum makes very small change to SGD but provides a big boost to speed up the learning.
* Disadvantage
  + The problem with only applying momentum in gradient descent is the momentum term is not dynamic enough. The method does not slow down even when we are reaching local minima, this causes the overshooting problem. Imagine a ball that rolling down a hill, it rolling down faster as it going down but it does not know when and where it should stop until it reach the next uphill slope. One of the method to improve this situation is by using Nesterov Accelerated Gradient (NAG). It prevents us from going too fast going down the hill. By taking the gradient from previous time step, the NAG anticipate where it should go likely. NAG applies the acceleration to the parameters before compute the gradients then update the computed gradients with the interim parameters. It helps to avoid the first-order optimization problem such as exploding gradient.
  + As mentioned above, gradient descent cannot differentiate between whether the curve is a plane, curve upwards or downwards, so it might go into wrong direction. Momentum with gradient descent also do not have the curvature info, thus it might cause slow converge.
* Other enhanced method
  + Newton method that takes into account of second derivative with the Hessian Matrix. Hessian gives us an estimate of the curvature of loss surface at a point. A positive curvature has a surface that rapidly getting steeper as it moves, vice versa. However, Hessian matrix is the combination of the double derivative of loss function which is not computational efficient with large number of parameters. Second order optimization is about the using the information of how the gradient changing itself. The drawback is Hessian requires to heavy computational power when the number of parameters are large.
  + RMSprop Optimizer is similar to the gradient descent with momentum. It restricts the oscillations in vertical direction so that it can increase the learning rate and take larger steps in horizontal direction, thus converging faster. RMSprop implicitly performs simulated annealing. When it heading down to the minima, we want it to slow down and not to overshoot the minima.
  + Adaptive Moment Optimization (Adam) combines both RMSprop and Stochastic Gradient Descent with momentum. It used the squared gradient to scale the learning rate like RMSprop and take advantage of momentum by using moving average of the gradient like SGD with momentum.
  + Nesterov Accelerated Adaptive Moment Estimation (Nadam). The idea is to use Nesterov momentum term for the first moving averages.

1. Application

Momentum gradient descent was used widely in Neural Network. It being implemented in some of the advanced neural network programming like PyTorch, Tensorflow etc. It’s one of the de facto optimizers using in current neural network programming.