**Does adding momentum to SGD (Stochastic Gradient Descent) always help?**

**Introduction**

SGD is often used in machine learning and neural networks to minimize the loss function by iteratively computing the optimal parameter set for a model that fits into the data. Gradient, as the names suggests, is the slope of the hyperplane in a given direction. In univariate functions it is simply the first derivative at the selected point, whereas for multivariate functions it’s the vector of partial derivatives in each dimension.

Stochastic Gradient Descent is invariably used to mean mini-batch gradient descent where the training observations are fed into the optimization algorithm in small batches, although true SGD means feeding one observation at a time. To minimize the loss function, the SGD algorithm takes a step iteratively by subtracting a scaled value (learning rate) of gradient at its current position from its current position.

**Use of Momentum in SGD**

taking double the computation time as SGD.

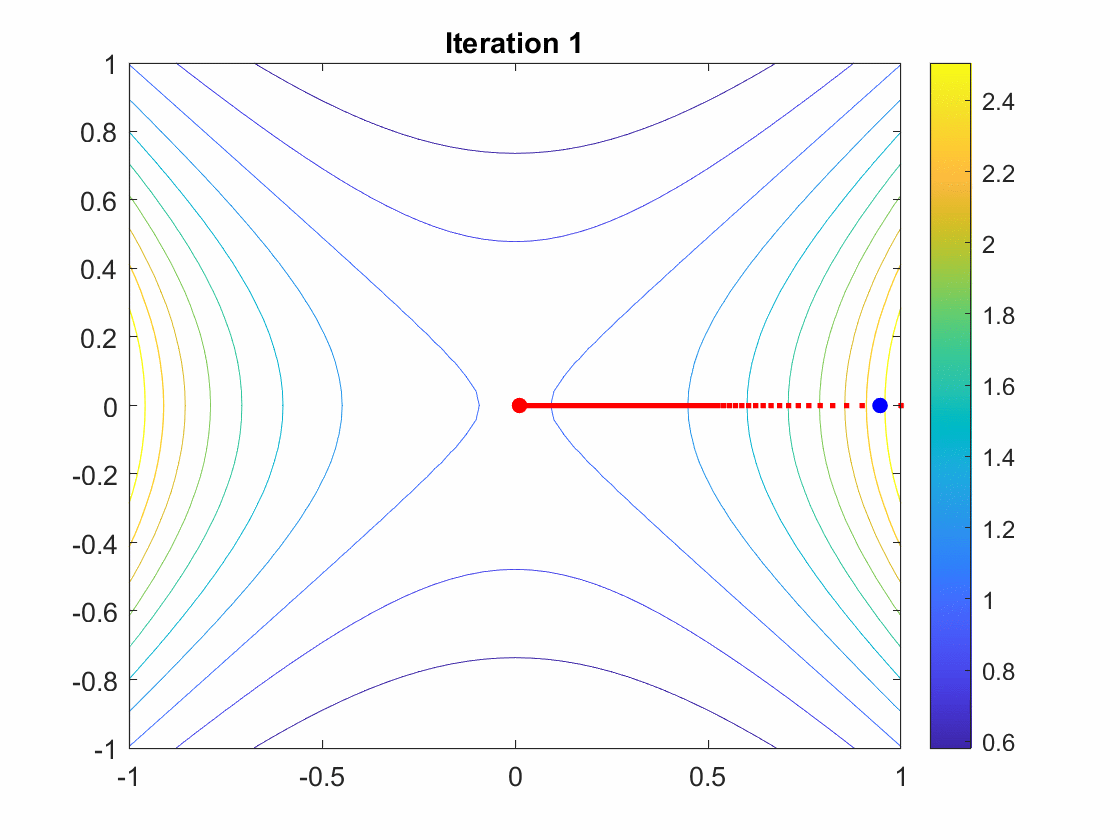
*Fig. 3 – Surface plot for loss function*

*Fig 4a (top left) – SGD on loss function*

*Fig 4b (top right) – SGD with momentum 0.5*

*Fig 4c (bottom left) – SGD with momentum 0.9*

inally, I shall examine the effect of momentum on a saddle point. To refresh our memory, a saddle point is a point on the hyperplane where both the first and second derivative is zero. The first derivative of zero implies it is a local extrema (also called minimax as its both a maximum in one dimension and minimum in another). The second derivative of zero implies it’s a point of inflexion, meaning the slope is increasing in one direction and decreasing in another. Function in Fig. 5 has a saddle point at (0,0).

The existence of saddle point poses a real challenge for SGD algorithms and using momentum does not help in this situation. The algorithm gets stuck at the saddle point (Fig. 6- red line) and although using momentum helps escape the saddle point initially, it returns to it soon after (Fig. 6 – blue line). Note that getting stuck at the saddle point highly depends on its starting position.

*Fig. 5 – Surface plot for*

**Conclusion**

Although momentum helps accelerate the optimization process of SGD, its best used to navigate in functions having ravines. It may help to add momentum in other loss functions, carefully choosing a lower momentum value to improve computation time. However, it has the same limitations as SGD when dealing with saddle points.

*Fig. 6 – SGD vs. SGD w/momentum*