Optimization Methods
Lab 11 Session

By Shi Jimao

Task1

Lab Session Task:

Show positive and negative effects of including the momentum term in the gradient decent algorithm with a constant step size using the following very simple problem and some other test problems.

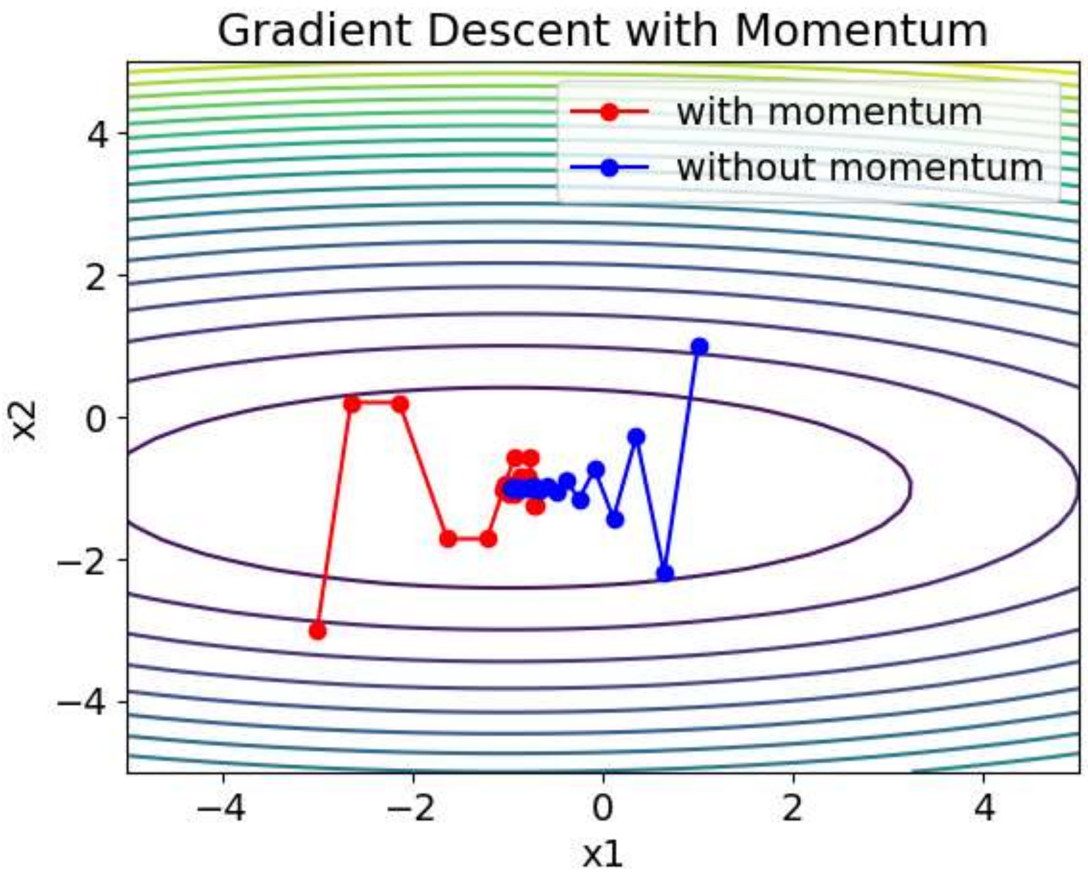
$$\text{Minimize } f(\mathbf{x}) = (x_1+1)^2/9 + (x_2+1)^2$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)})$$

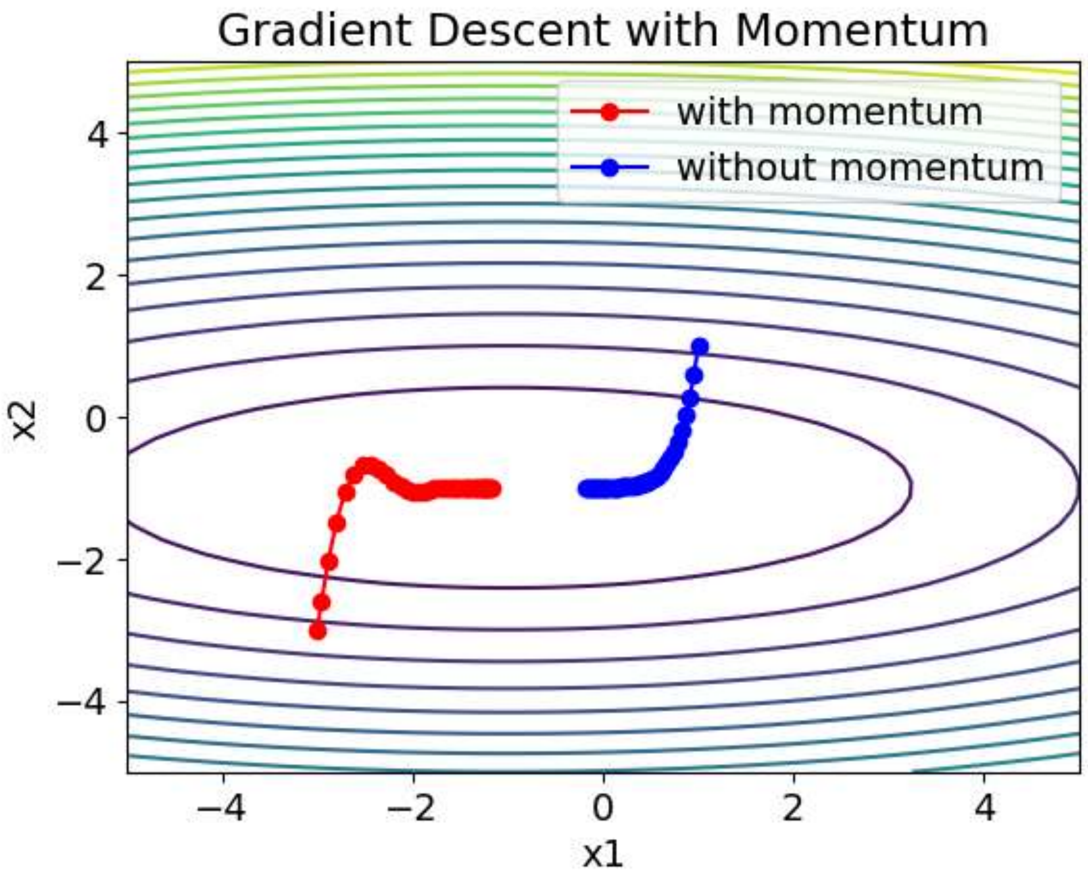
$$\Longrightarrow \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)}) + \beta(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$$

We can find that when β is relatively small, with this momentum, it can search the solution space faster.

Learning rate=0.95 β=0.5

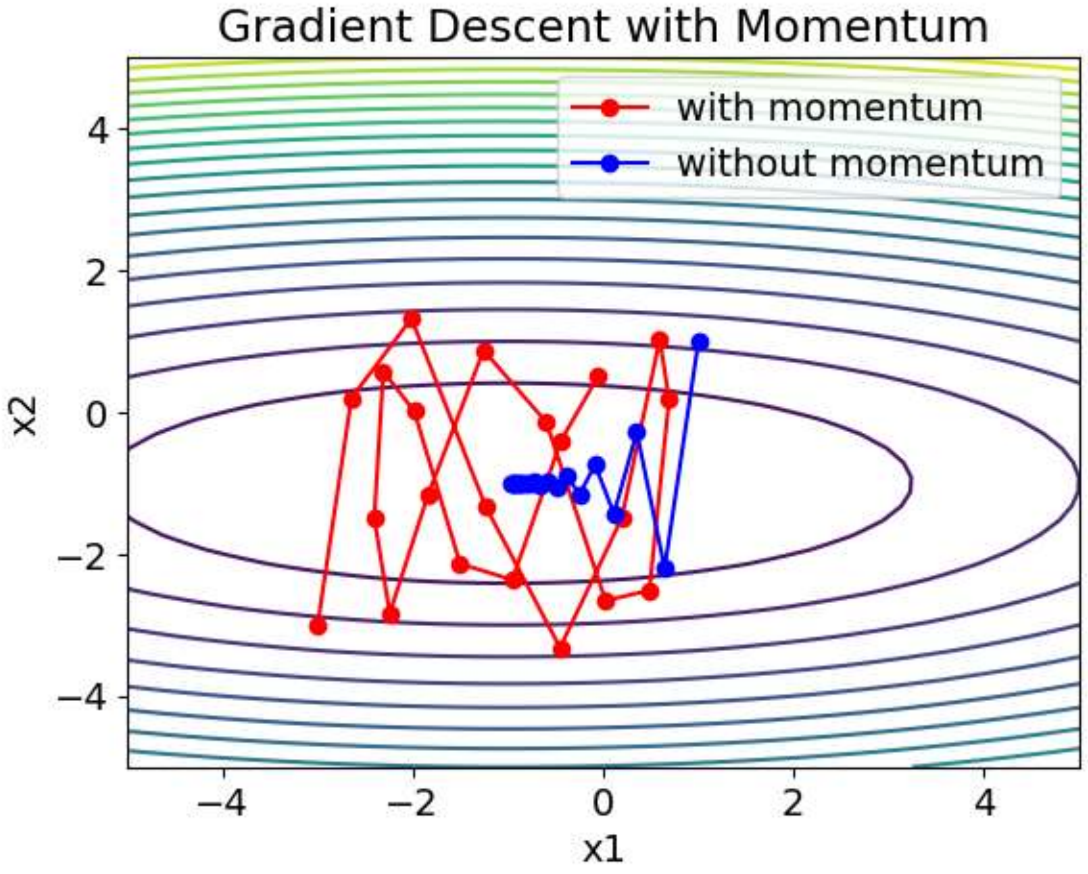


Learning rate=0.1 β=0.6

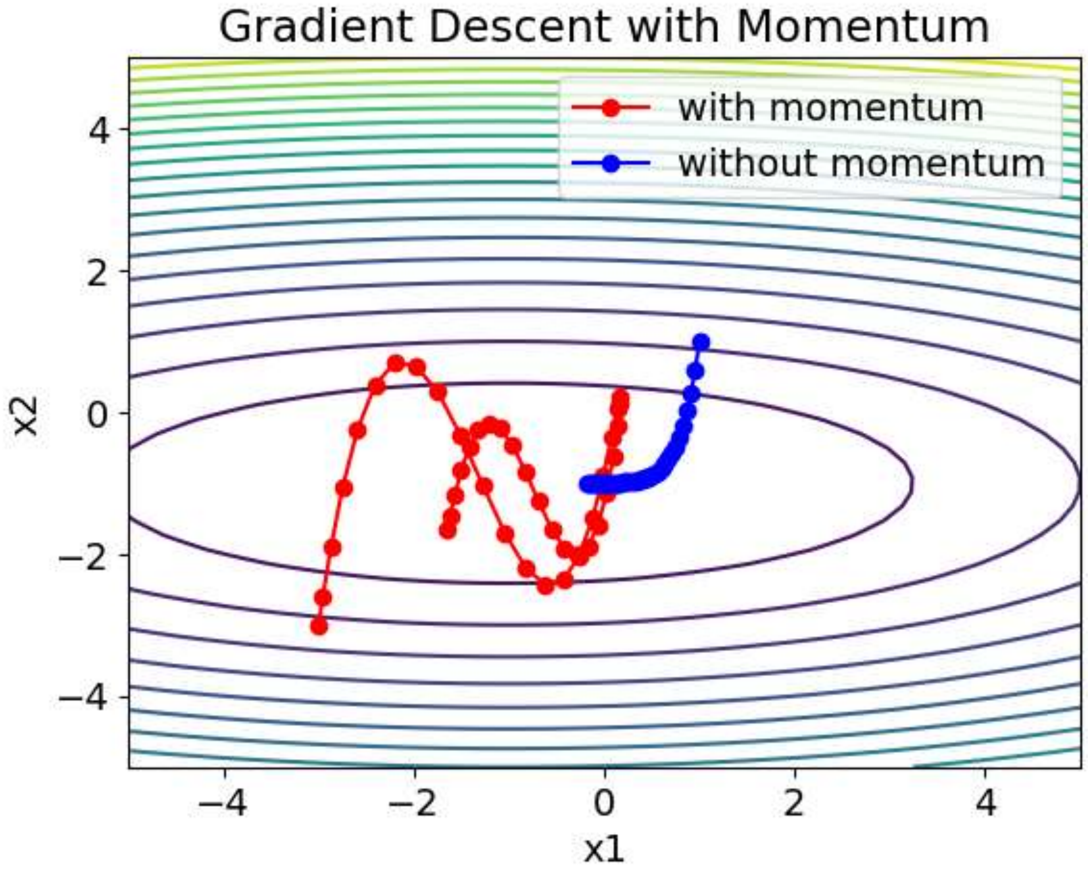


We can find that when β is large, with so much momentum, it‘s easy to overdo the function. At the same time, when learning rate is large, the rate of convergence of optimization with momentum is much slower than without momentum.

Learning rate=0.8 β=0.95



Learning rate=0.1 β=0.95



Positive Effects:

1. **Accelerated Convergence**: Momentum helps the optimization algorithm to maintain a consistent direction of movement. This can lead to faster convergence towards the minimum, especially in scenarios where the landscape is rugged or has narrow valleys. In the case of your simple problem, this could mean reaching the minimum faster.
2. **Escape Local Minima**: Momentum can help the algorithm escape from local minima or plateaus by providing enough force to push through those regions. This is particularly beneficial in non-convex optimization problems, where traditional gradient descent might get stuck.
3. **Improved Robustness**: Momentum can smooth out noisy gradients, making the optimization process more robust to noisy or stochastic gradients. This is particularly useful in scenarios like training neural networks, where the gradients might be noisy due to mini-batch sampling.

Negative Effects:

1. **Overshooting the Minimum**: With momentum, there's a risk of overshooting the minimum, especially when the momentum term is large or the surface is highly curved. This can lead to oscillations around the minimum or even divergence from it, causing instability in the optimization process.
2. **Difficulty in Fine-Tuning**: Momentum introduces an additional hyperparameter (the momentum coefficient) that needs to be tuned. Finding the right value for this coefficient can be challenging, and an inappropriate value might lead to suboptimal performance or instability in convergence.
3. **Difficulty in Escaping Sharp Minima**: In some cases, momentum may hinder the ability to settle into sharp minima. Instead, it may prefer wider valleys or plateaus due to the smoothing effect of momentum. While this might not be an issue for some optimization problems, it could lead to suboptimal solutions in scenarios where sharp minima are desirable.
4. **Increased Memory Usage**: Although this might not be a significant concern in simple problems, in large-scale optimization tasks or deep learning, momentum requires additional memory to store the momentum term for each parameter being optimized. This can increase memory requirements, which might become a bottleneck in memory-constrained environments.