Comparison between Batch, Momentum-based, and Nesterov Accelerated Gradient Descent

1) Comparison between Batch and Momentum-based Gradient Descent

The following is the output that I got by running the uploaded script of Batch Gradient Descent(batch GD.py).

Fig 1.Output of Batch Gradient descent for 999 iterations

Here, I have set:
i) initial weight (w1) = 0
ii) bias (b) = 0
iii) learning rate = 0.1
iv) maximum epochs = 999

The following is the output that I got by running the uploaded script of Momentum-based Gradient Descent(momentum_based GD.py)

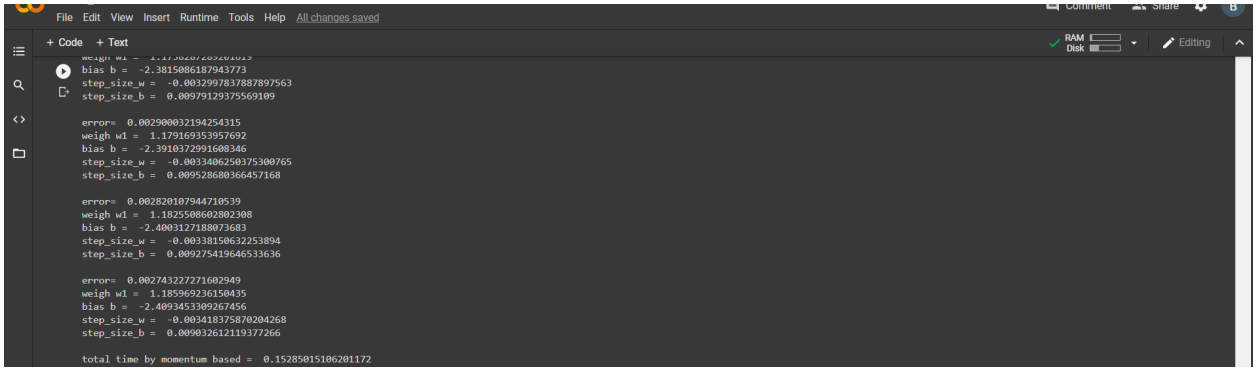

Fig 2. Output of Momentum-based Gradient Descent

Here, I have set:
i) initial weight (w1) = 0
ii) bias (b) = 0
iii) learning rate = 0.1
iv) maximum epochs = 84

It can be observed from the fig1 and 2 that for the same function with the same training set, initial weight (w1), bias (b), and learning rate; Batch Gradient Descent gives an error of 0.002 after 999 iterations whereas Momentum-based Gradient Descent gives almost the same output only after 84 iterations. Therefore, it can be inferred that Momentum-based Gradient Descent (and thus Nesterov Accelerated Gradient Descent) is way efficient than Batch Gradient Descent.

2) Comparison between Momentum-based Gradient Descent and Nesterov Accelerated Gradient Descent

The following is the output that I got by running the uploaded script of Nesterov Accelerated Gradient Descent(nesterov_accelerated GD.py)

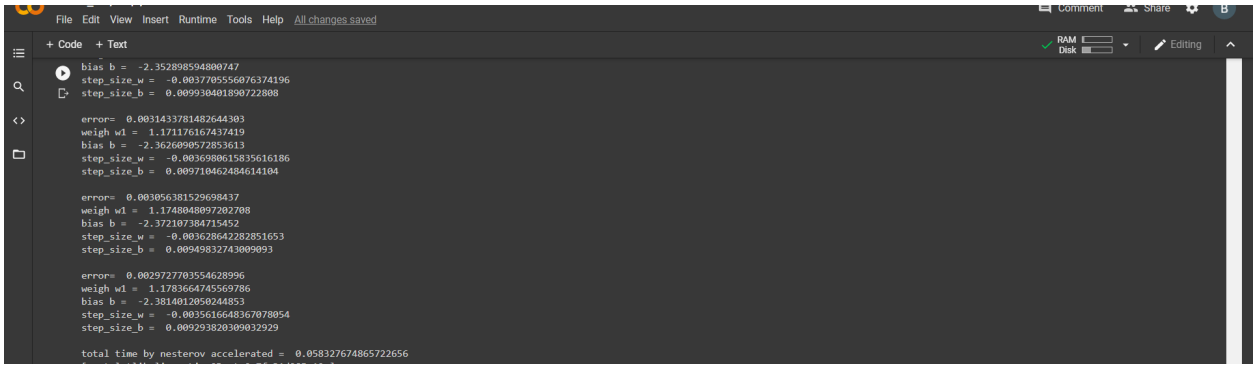

Fig 3. Output of Nesterov Accelerated Gradient Descent

Here, I have set:
i) initial weight (w1) = 0
ii) bias (b) = 0
iii) learning rate = 0.1
iv) maximum epochs = 84

It can be noted from fig 2 and 3 that both Momentum-based Gradient Descent and Nesterov Accelerated Gradient Descent gives almost the output for the same function, training set, parameters, and hyper-parameters; but

the time taken by Nesterov Accelerated Gradient Descent to do this quite less than what is taken by Momentum-based Gradient Descent.

NOTE: This time taken can keep fluctuating. The above shown time are the ones which I got when I ran both the scripts for the first time. Also, here we have taken only 2 training data, but when dealing with real-world datasets where we consider lakhs of data inputs, during that instance, the total time taken by an algorithm will impact a major difference.

Moreover, below I have attached the different types of graphs for Momentum-based Gradient Descent and Nesterov Accelerated Gradient Descent.

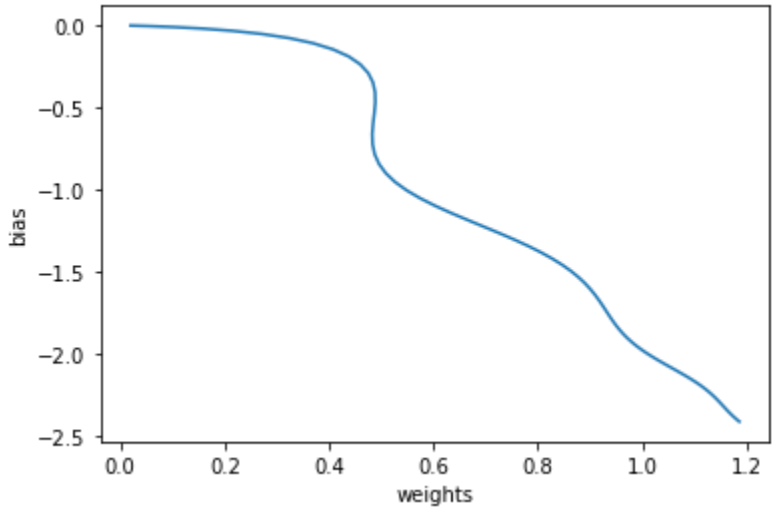

Fig 4. Weight vs Bias graph in Momentum-based Gradient Descent

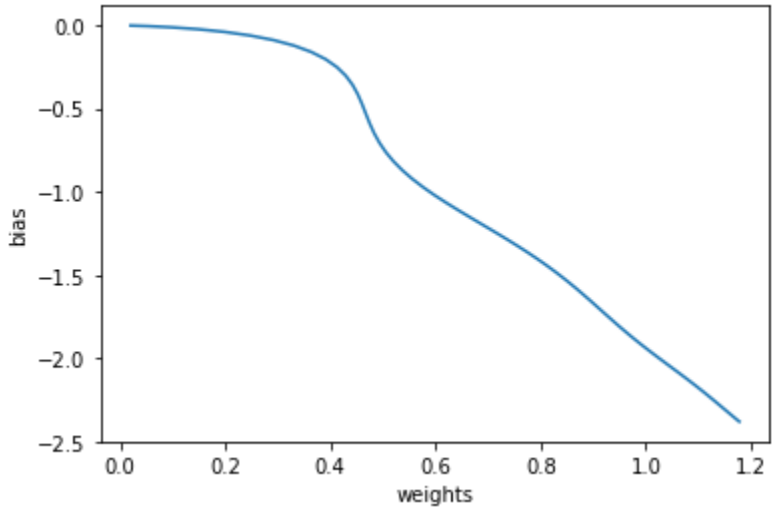

Fig 5. Weight vs Bias graph in Nesterov Accelerated Gradient Descent

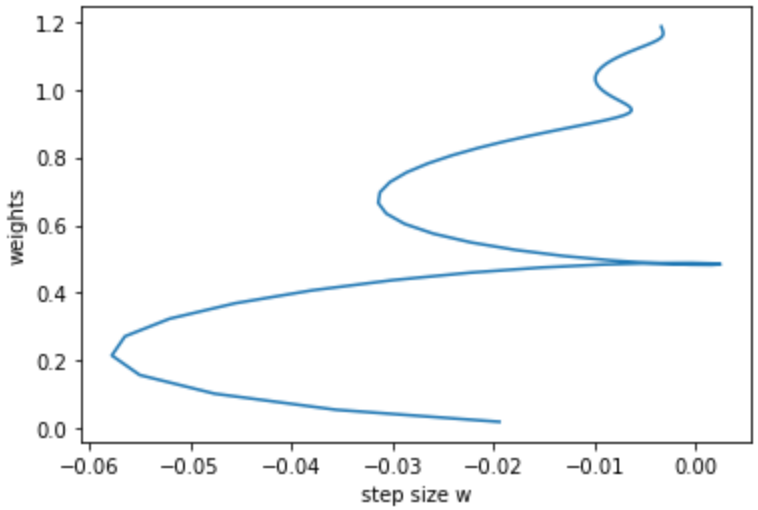

Fig 6. Step size for weights vs weights graph in Momentum-based Gradient Descent

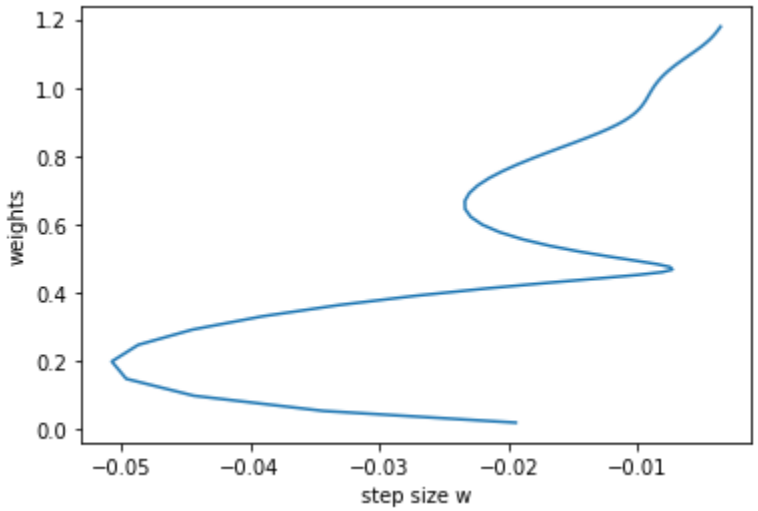

Fig 7. Step size for weights vs weights graph in Nesterov Accelerated Gradient Descent

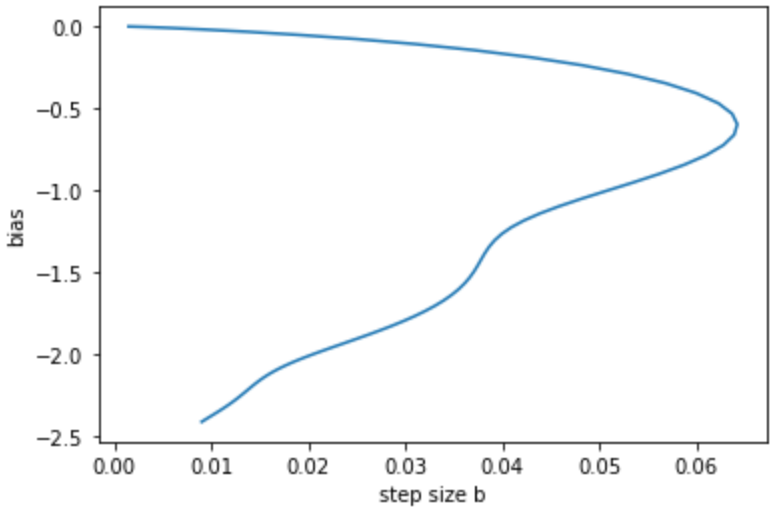

Fig 8. Step size for bias vs bias graph in Momentum-based Gradient Descent

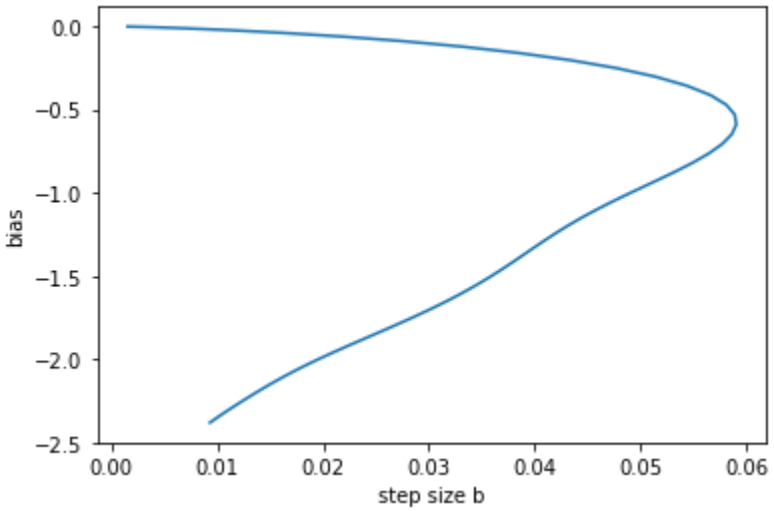

Fig 9. Step size for bias vs bias graph in Nesterov Accelerated Gradient Descent

Therefore, it can be inferred from fig 4-9, that the oscillations observed in the case of Nesterov Accelerated Gradient Descent are less than the ones observed in the case of Momentum-based Gradient Descent. Also, as Momentum-based Gradient Descent and Nesterov Accelerated Gradient Descent gives almost the output for the same function, training set, parameters, and hyper-parameters; it can be concluded that Nesterov Accelerated Gradient Descent is more efficient than Momentum-based Gradient Descent.