X=[0.5,2.5]

Y=[0.2,0.9]

Gamma=0.1

W=-2, B=-2, Learning rate=0.1, Epoch=1000

|  |  |  |  |
| --- | --- | --- | --- |
| Vanilla GD Time | Momentum GD Time | Vanilla GD Error | Momentum GD Error |
| 0.027907609939575195 | 0.03194403648376465 | 0.01644945958207577 | 0.09834372777194457 |

W=0, b=0, learning rate=0.1, epoch=100

|  |  |  |  |
| --- | --- | --- | --- |
| Vanilla GD Time | Momentum GD Time | Vanilla GD Error | Momentum GD Error |
| 0.0049860477447509766 | 0.001995086669921875 | 0.04925120442438629 | 0.09834381650886266 |

W=1, b=1, learning rate=0.2, epoch=100

|  |  |  |  |
| --- | --- | --- | --- |
| Vanilla GD Time | Momentum GD Time | Vanilla GD Error | Momentum GD Error |
| 0.0020766258239746094 | 0.006899356842041016 | 0.040598927882388114 | 0.0870193761778722 |

W=10, B=10, Learning Rate = 0.2 ,Epochs=1000

|  |  |  |  |
| --- | --- | --- | --- |
| Vanilla GD Time | Momentum GD Time | Vanilla GD Error | Momentum GD Error |
| 0.030920028686523438 | 0.05583453178405762 | 0.324999755263292 | 0.08701875940655789 |

W=20, B=10, Learning rate=0.4, Epochs=100

|  |  |  |  |
| --- | --- | --- | --- |
| Vanilla GD Time | Momentum GD Time | Vanilla GD Error | Momentum GD Error |
| 0.003962516784667969 | 0.002972841262817383 | 0.32499999835107707 | 0.07449282533775586 |

**Conclusion :**

From the above analysis, we can see that if we initialize weight and bias values very far from optimal value then momentum gradient descent performs better than the vanilla gradient descent in terms of loss as well as in terms of time for the same learning rate and same number of epochs. But if we initialize weight and bias values very near around its optimal value then vanilla gradient descent works better than the momentum gradient descent because if w and b are already near around its optimal value then in momentum gradient descent its overshoot the optimal value and its oscillate around optimal value.