VANILLA GRADIENT DESCENT VS MOMENTUM BASED GRADIENT DESCENT

w	b	Learning rate	Max Epoch	Vanilla Gradient Descent		Momentum based Gradient Descent		
				Loss	Execution Time	Loss	Gamma	Execution Time
-2	-2	0.01	1000	0.414	0.0149	0.120	0.1	0.0159
-2	-2	0.1	1000	0.016	0.0139	0.098	0.1	0.0169
-2	-2	1	1000	2.429 e -11	0.0159	0.054	0.1	0.0159
-2	-2	10	1000	2.503 e -32	0.0140	0.0099	0.1	0.0150
-2	-2	100	1000	0.424	0.0139	0.0057	0.1	0.0169
-2	-2	0.01	10000	0.0163	0.1465	0.1207	0.1	0.1556
-2	-2	0.08	10000	1.61 e -9	0.1476	0.1017	0.1	0.1565
-2	-2	0.09	10000	2.04 e -10	0.1455	0.0999	0.1	0.1546
-2	-2	0.1	10000	2.5 e -11	0.1436	0.0983	0.1	0.1595
-2	-2	1	10000	2.5 e -30	0.1575	0.0542	0.1	0.1715
-2	-2	10	10000	2.5 e -32	0.1456	0.0099	0.1	0.1745
-2	-2	20	10000	6.5 e -33	0.1525	0.0057	0.1	0.1595
-2	-2	50	10000	0.0004	0.1416	0.3217	0.1	0.1516

- From table 1, we can see that vanilla gradient descent performs better in terms of time and loss when we choose weights and bias closer to optimal value. So, momentum based gradient descent have to take sharp u turns in smaller distance. So, confidence is not gained.

- *So let's try changing weights and bias far apart to notice a substantial difference.*

w, b = 50 and max_epoch = 1000

Learning rate	VGD Loss	MGD Loss	VGD Time	MGD Time
1	0.325	0.054	0.0129	0.0159
5	0.325	0.0174	0.0129	0.0169
10	0.325	0.0099	0.0159	0.0169
50	0.325	0.0003	0.0139	0.0159

So we can clearly see now that Momentum based gradient descent converges faster than vanilla gradient descent in spite of time difference to 2-3 milliseconds.