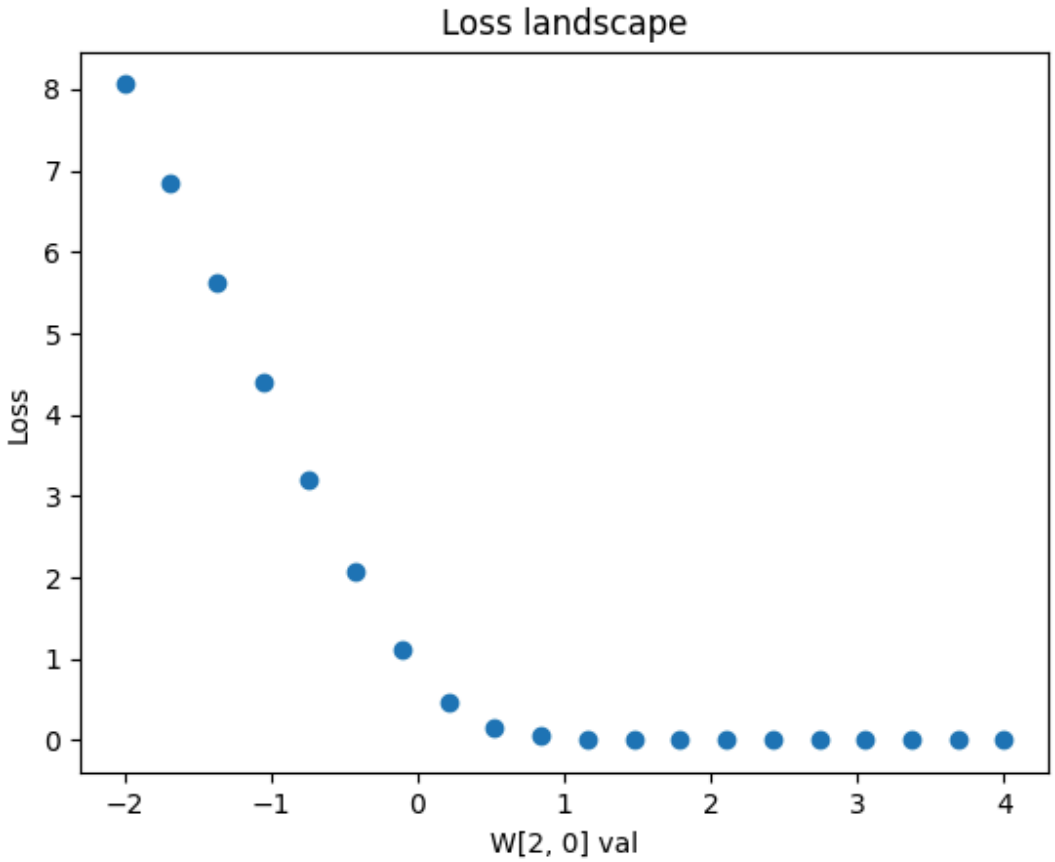
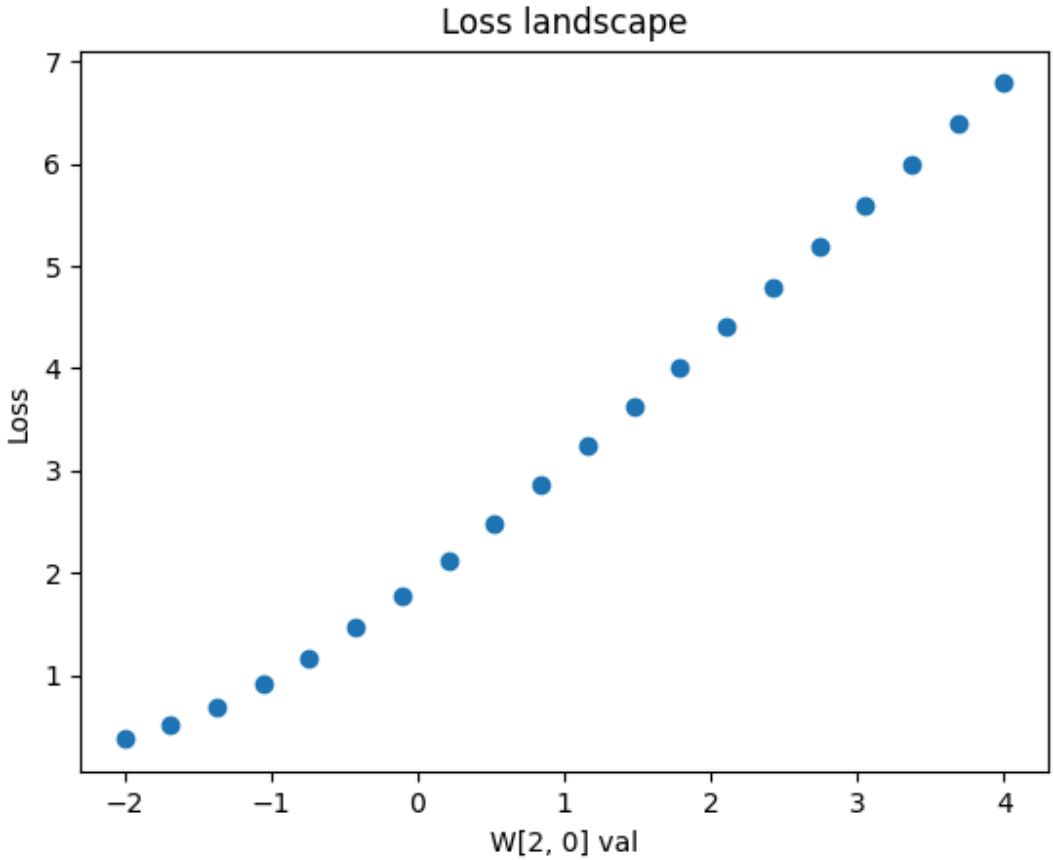
A

In the two choices of batch sizes I tested, which were 1 and 10. The standard deviation for the batch size 1 was much larger than the 10. Which is very reasonable because when the batch_size is 10 the loss will be a local minimum loss given 10 examples, when it is 1 it will be the loss of the single given example, and when compared the respective losses of each batch it's much more likely to fluctuate when the batch size is much more smaller.

Here are two examples of showing where the minimum loss occurred when the batch was 1:



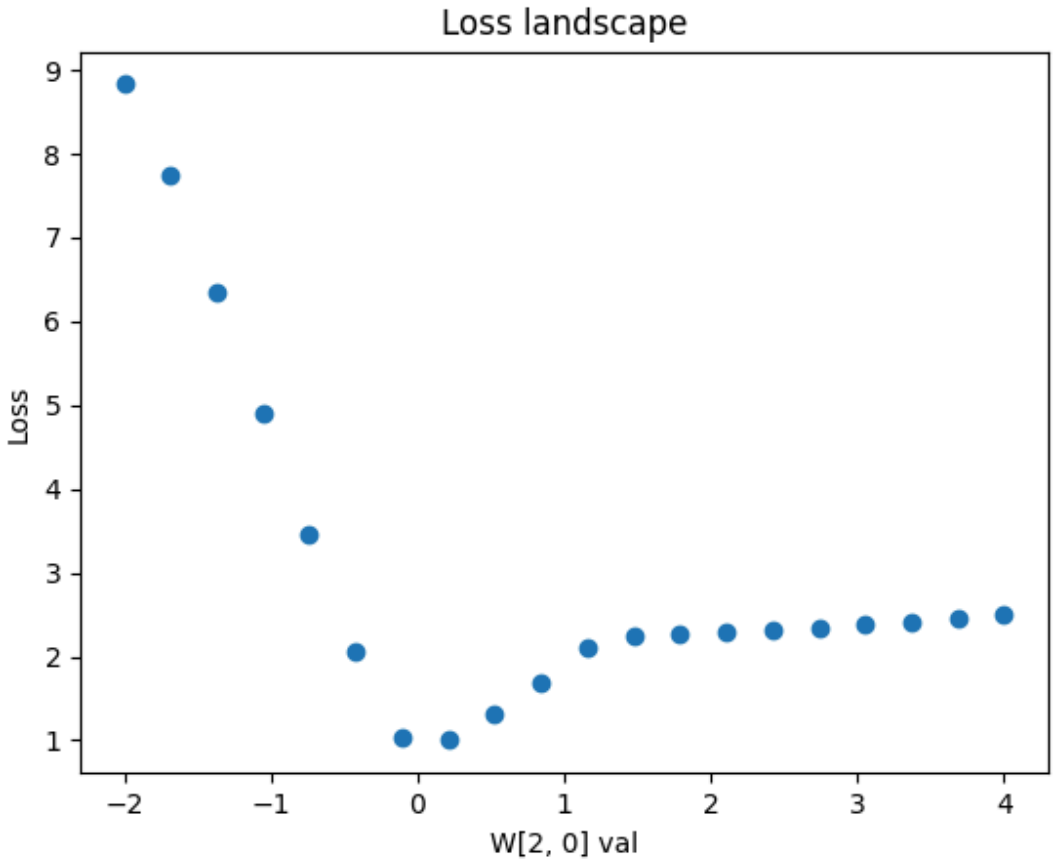
This is the plot for the batch_0_of_20



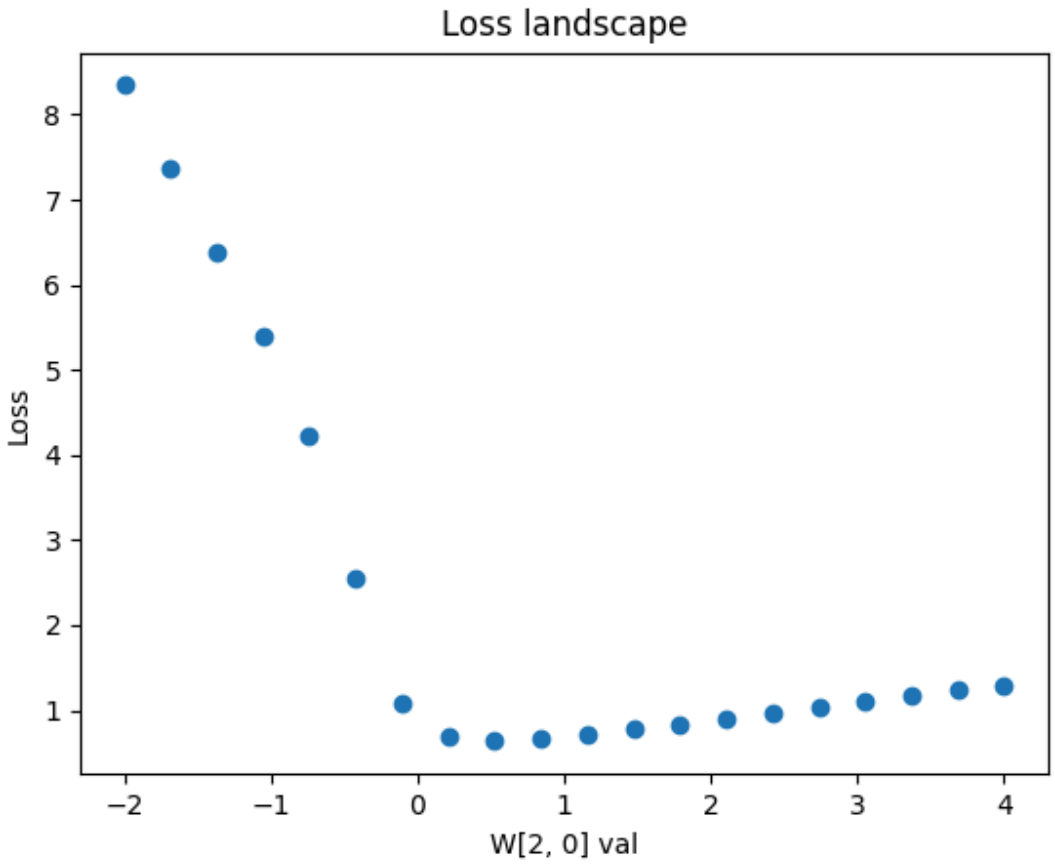
This is the plot for batch 19_of_20.

It's clearly observable that there is a significant difference with lowest loss with respect to W[2, 0] for the given examples using a single example as input. In one example the lowest loss clearly occurs as -2 for the other at either 3 or 4.

In the examples below, the batch size is 10 and it is visually observable that the loss function plots show more resemblance compared to the 2 examples provided for batch size 1. Since the size of the total dataset is 20, when batch size is 10 only 2 plots are created and both of them are presented below.



This is the plot for the batch_0_of_2



This is the plot for the batch_1_of_2

B

Solely based on the dataset given, which is a very small dataset. Batch gradient descent overall provides more general and balanced results. However, as the dataset gets very large computing the batch gradient descent can become computationally expensive, and stochastic or mini-batch gradient descent can likely become more optimal approaches.