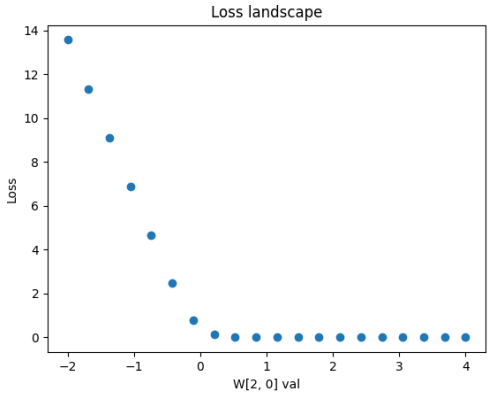
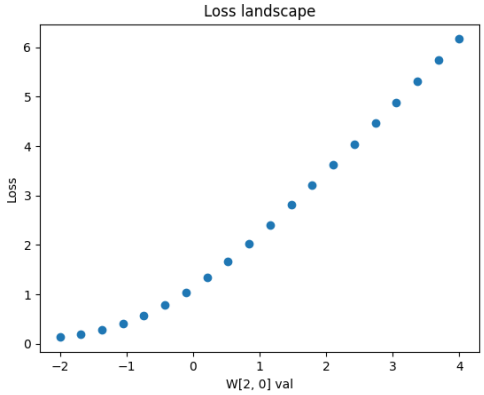
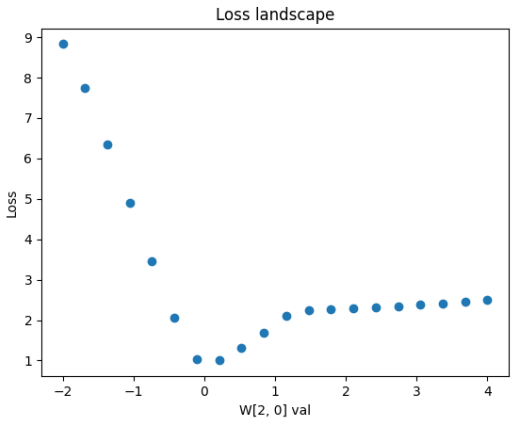
3a.

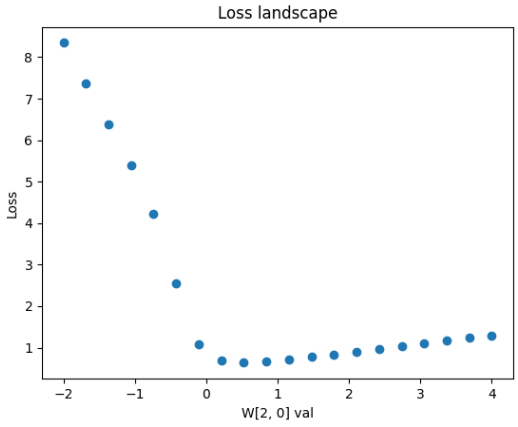
I chose batch sizes of 1 and 10 which had standard deviations of 2.36 and 0.16 respectively. For the batch size of 1, each parameter is fit according to a smaller dataset, and there can be large variation between each dataset. This means that the optimal value for that parameter can vary greatly between datasets. We can see this in the two plots below:





Each plot represents the loss landscape for the weight W[2,0] from a different batch of batch size 1. Due to the high variance between the small batches, the data on the left encourages a weight value greater than zero, while the data on the right encourages a weight value much less than zero. These sorts of variations are less present in the experiments with a batch size of ten. The results for both of *those* batches are shown below; note that the landscapes are quite similar between them.





3b.

This tells me that increasing batch size increases consistency between batches and the accuracy of each batch, while also causing slower runs. The tradeoff between stochastic, minibatch, and batch gradient descent is one of speed (lower batch sizes) vs accuracy (higher batch sizes); it is probably the case that minibatch gradient descent can manage this tradeoff the best. Stochastic gradient descent is the fastest but can be quite inaccurate, while batch gradient descent may simply take too long.