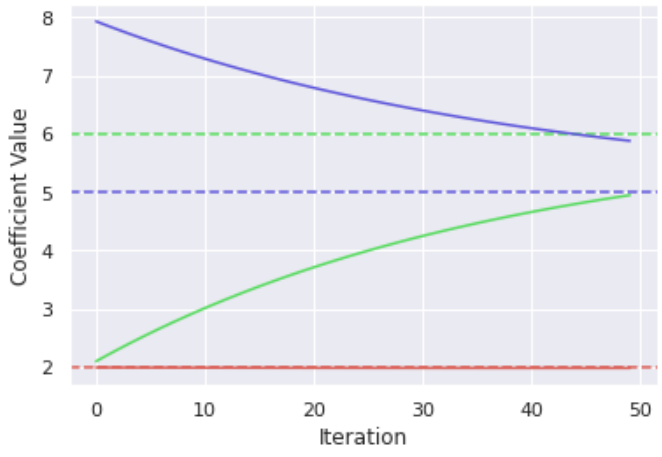
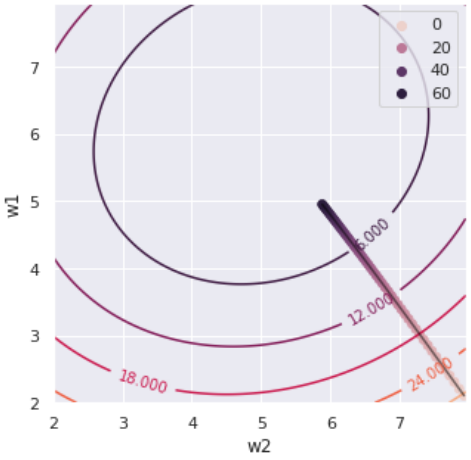
3a. Gradient descent on perfect linear data, with varying learning rates.

`lr = 0.0002`

The gradient descent starts to converge, but does not get very close to the optimum value within 50 iterations.

Estimate of `w` after 50 iterations: `[1.99001262, 4.95054598, 5.88029045]`

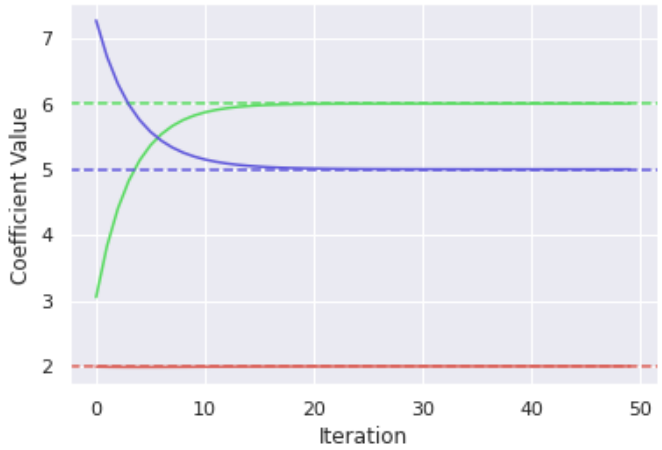


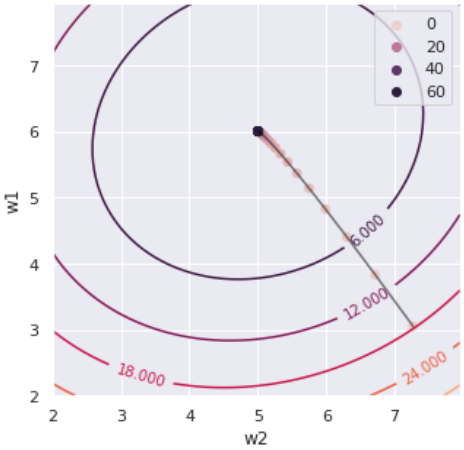


`lr = 0.002`

Estimate of `w` after 50 iterations: `[1.99999239, 6.00000677, 5.00001589]`

The gradient descent converges within 50 iterations.

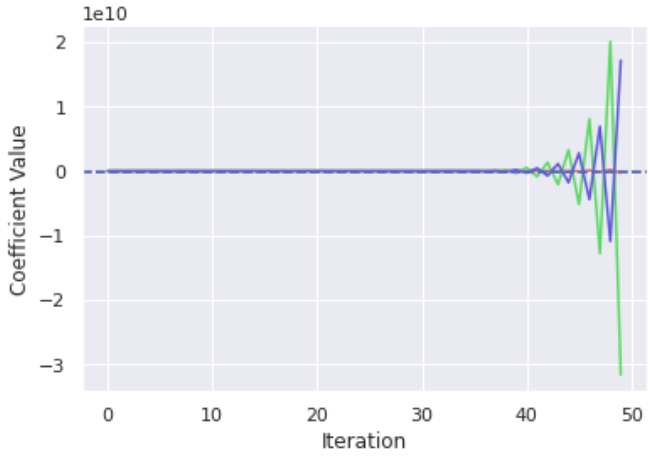


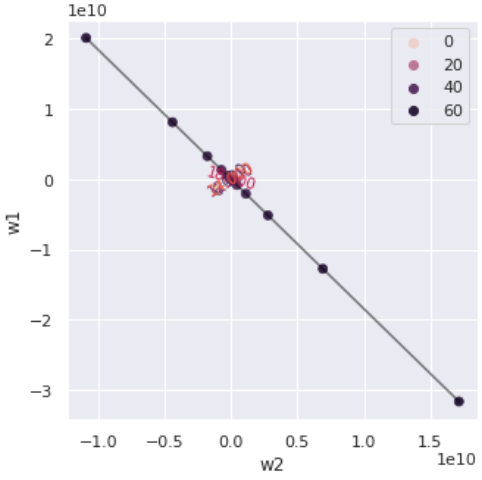


`lr = 0.02`

The gradient descent diverges.

Estimate of `w` after 50 iterations: `[-2.89600611e+08, -3.16109293e+10,  1.71599815e+10]`

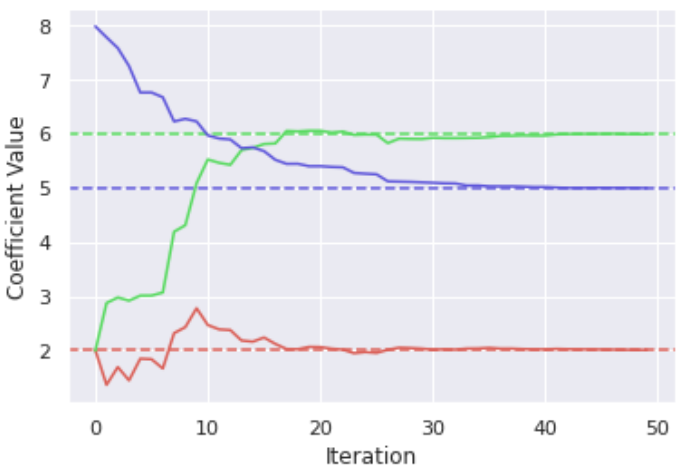


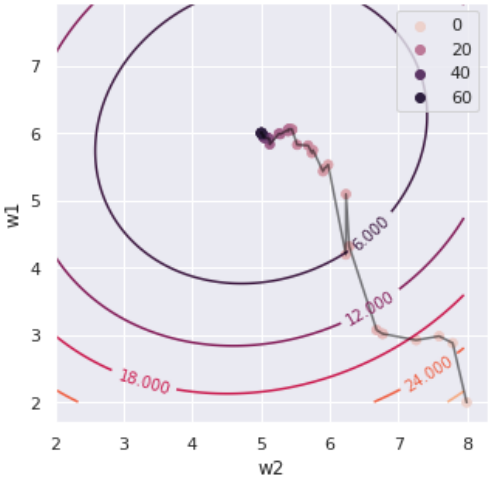


3b. Stochastic gradient descent on perfect linear data, with varying learning rates and mini-batch sizes.

`lr = 0.1, n=1`

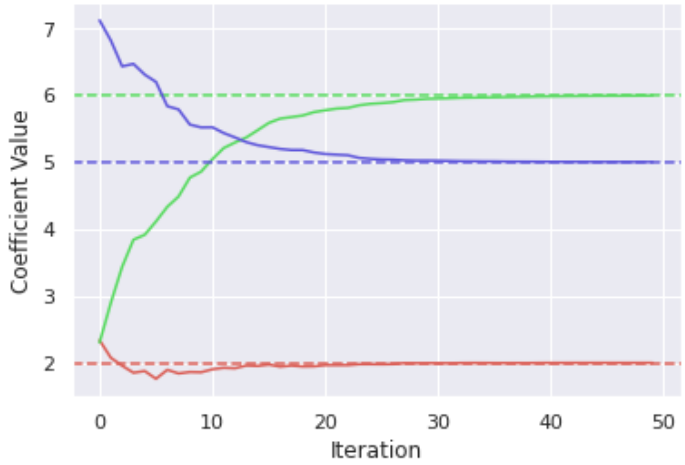
The descent path is not smooth.

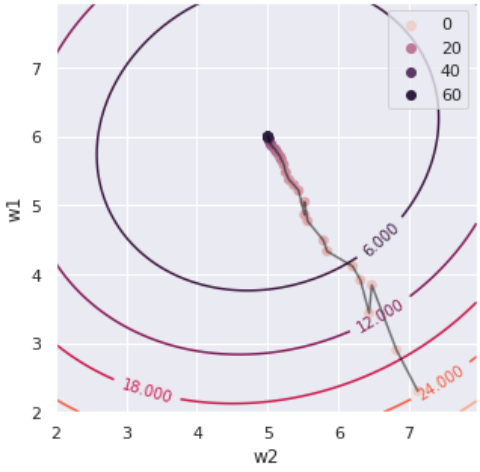




`lr = 0.01, n=10`

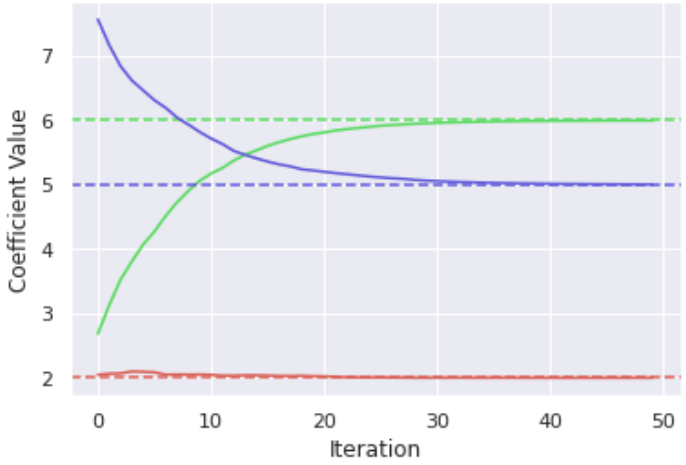
The descent path gets smoother as the mini-batch size increases.

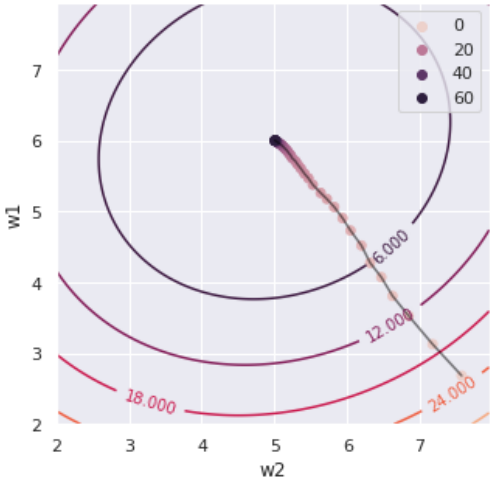




`lr = 0.001, n=100`

With a large mini-batch size, the descent path is much smoother.





Note that the change in learning rate is only to offset the increase in the sum of gradients, when we sum over more samples in a mini-batch. The change in the smoothness of the descent path is due to the increase in mini-batch size.