**Making multiple updates to weights**

You're now going to make multiple updates so you can dramatically improve your model weights, and see how the predictions improve with each update.

To keep your code clean, there is a pre-loaded get\_slope() function that takes input\_data, target, and weights as arguments. There is also a get\_mse() function that takes the same arguments. The input\_data, target, and weights have been pre-loaded.

This network does not have any hidden layers, and it goes directly from the input (with 3 nodes) to an output node. Note that weights is a single array.

We have also pre-loaded matplotlib.pyplot, and the error history will be plotted after you have done your gradient descent steps.

**Instructions**

* Using a for loop to iteratively update weights:
  + Calculate the slope using the get\_slope() function.
  + Update the weights using a learning rate of 0.01.
  + Calculate the mean squared error (mse) with the updated weights using the get\_mse() function.
  + Append mse to mse\_hist.
* Hit 'Submit Answer' to visualize mse\_hist. What trend do you notice?