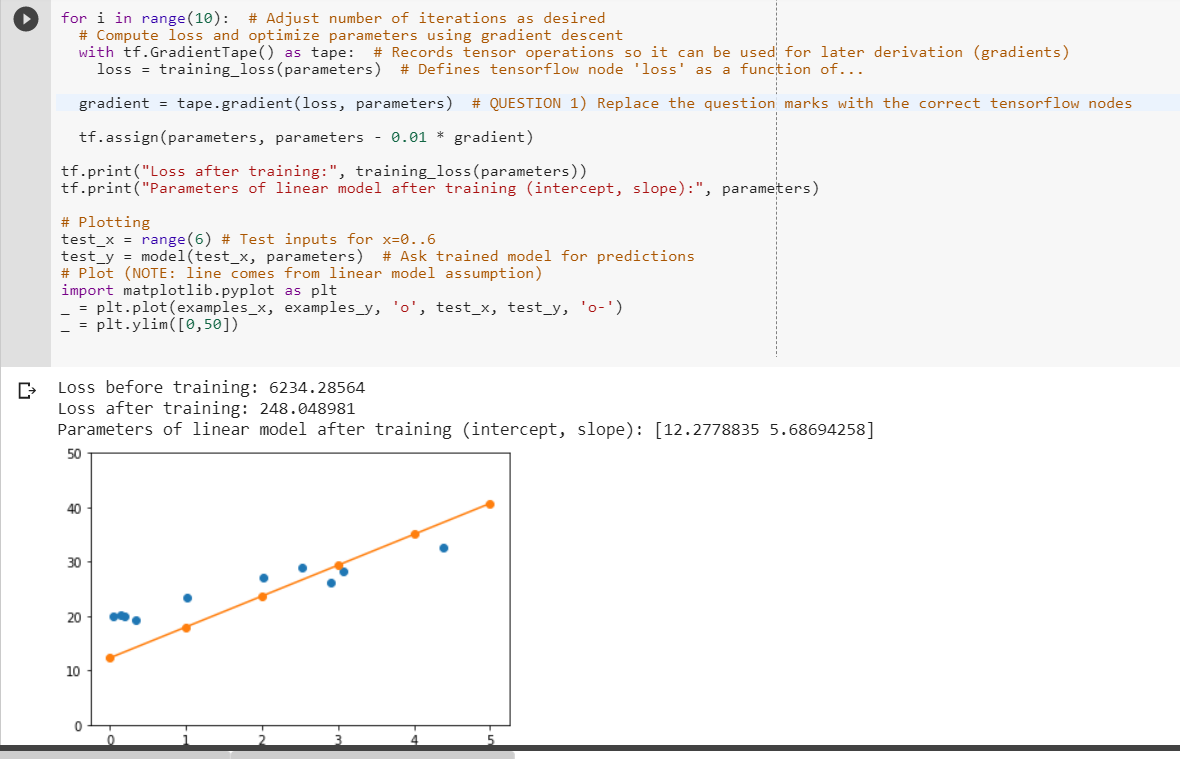
**Lab 6: Deep Learning**

**Question 1)** In the gradient descent learning code below, please complete the gradient computation by inputing the correct variables where there are question marks. We are using Tensorflow to automatically compute the gradient with GradientTape (tape) similar to how you were taught above, but to do supervised learning which tensorflow node do we compute the gradient of, and with regard to which var

In order to calculate the new gradient, we have to calculate the loss function using the new parameters.

**Question 2)** Show the math for why the first Dense layer has 100 480 parameters with these inputs and number of neurons. Remember, a neuron is just a non-linear transformation (e.g. sigmoid/ReLU) of a linear model, implying one parameter for each input dimension, + 1 for the line intercept/constant (also called neuron "bias" in NN slides from the first ML lecture).

images are composed of 28 \* 28 pixels that represent an array of 784 values.

Each of them are connected to 128 neurons, the total amount of parameters is :

784 \* 128 + 128 = 100 480

**Question 3)** Here you will evaluate different mini-bach sizes for stochastic gradient descent (see the deep learning lecture). Please separately run the training code above with batch sizes of 1, 10, 100, 1000 and 60000. Write down the training times (you can use the first number in seconds, not the per sample time) and the training set accuracy reached, both in the first line of the output. This can randomly vary a bit between runs but it should give you an idea. In your lab report, plot both curves and reason about which batch size produced the most accuracy given the time spent, i.e. which batch size would be best to start the training with? You have to run the Reset All Parameters code above this one between your runs to always start over.

1 : time = 113s, accuracy = 0.8323

10 : time = 13s, accuracy = 0.8265

100 : time = 3s, accuracy = 0.8889

1000 : time = 1s, accuracy = 0.88899

60000 : time = 16ùs, accuracy = 0.8942