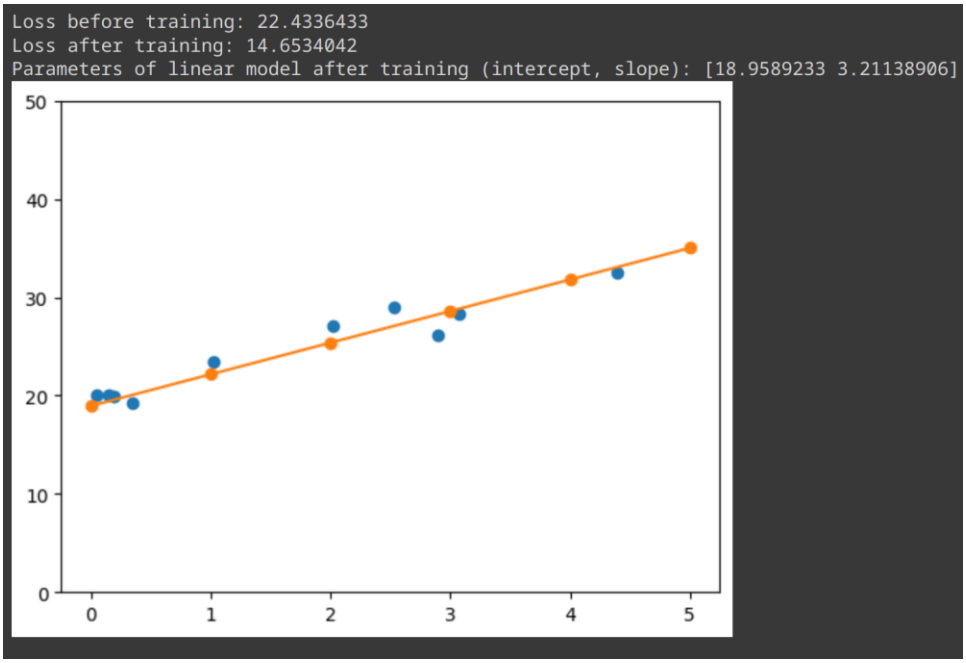
REPORT Lab 3

Group 2, Subgroup 7 (*prigu857* and *fabcr549*)

The aim of this lab was to follow a short, guided tutorial regarding supervised deep learning.
The goal was to answer the following questions.

1. In the gradient descent learning code below, please complete the gradient computation by inputting 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 variable? Please fill this in below (and in your lab report). If you get stuck, the slides on training models in the first ML lecture might be helpful.

We use gradient descent to minimize the loss function, thus we calculate the gradient of the loss function about the input parameters. The obtained parameters for the model are shown along with the resulting plot in the following picture.



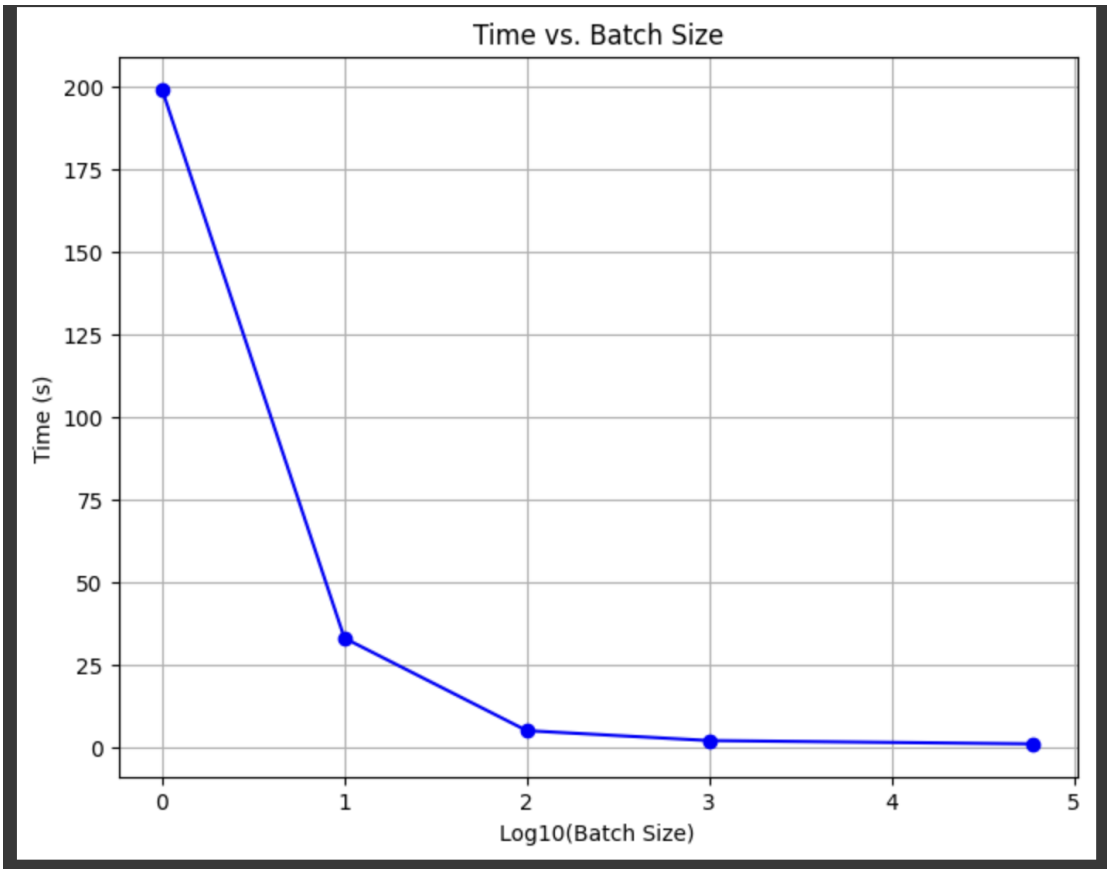
2. Show the math for why the first Dense layer has 100 480 parameters with these inputs and number of neurons. Remember, a neuron is just a non-linear transformation (e.g. sigmoid/ReLU) of a **linear model**, implying one parameter for each input dimension, + 1 for the line intercept/constant (also called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

2 dimensions (each 28) + 1 neuron bias * 128 neurons available in the first layer:

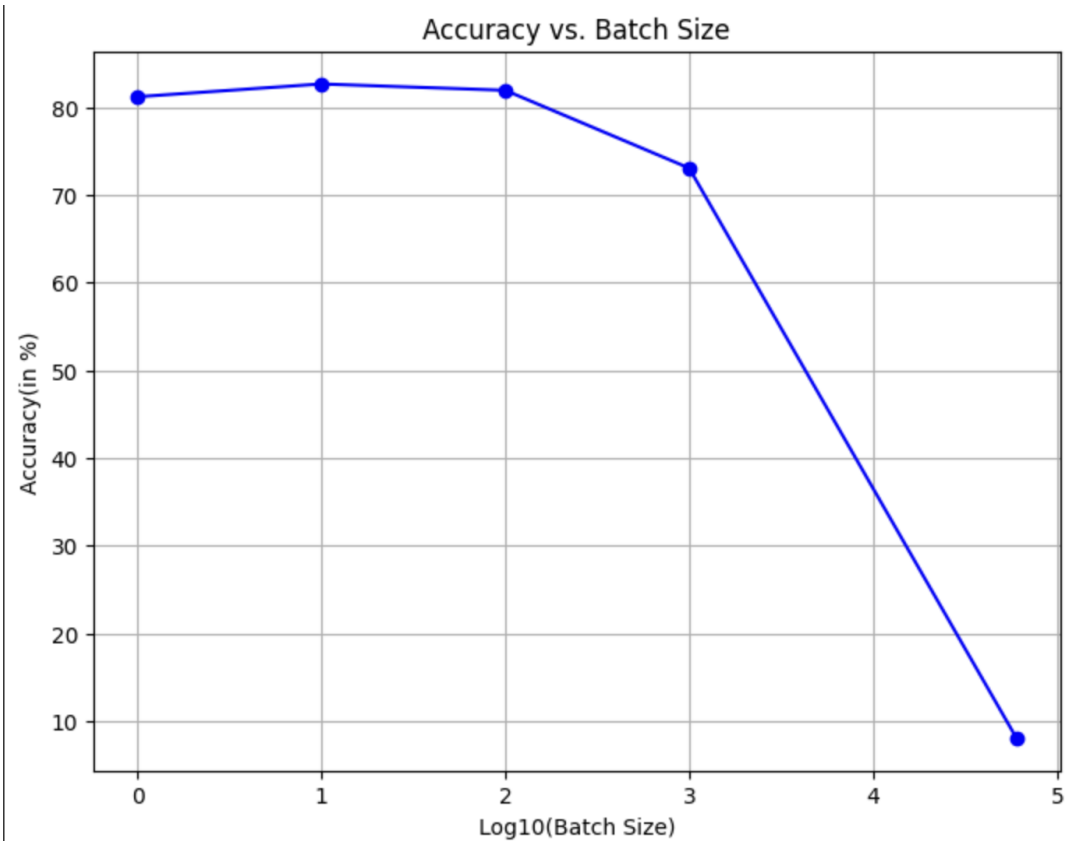
3. Here you will evaluate different mini-bach sizes for stochastic gradient descent (see the deep learning lecture). Please separately run the training code above with batch sizes of 1, 10, 100, 1000 and 60000. Write down the training times (you can use the first number in seconds, not the per sample 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.

=> obtained results are in the below table

Batch size	Time (s)	Accuracy (%)
1	199	81.17
10	33	82.63
100	5	81.91
1000	2	73.04
60000	1	8.12



.



With batch size 10 and 100 we got almost similar accuracy, but the time taken by the latter is much smaller than the time taken by the former, as it needs to train for a smaller set. Thus, we think that batch size 100 will be best in this case.