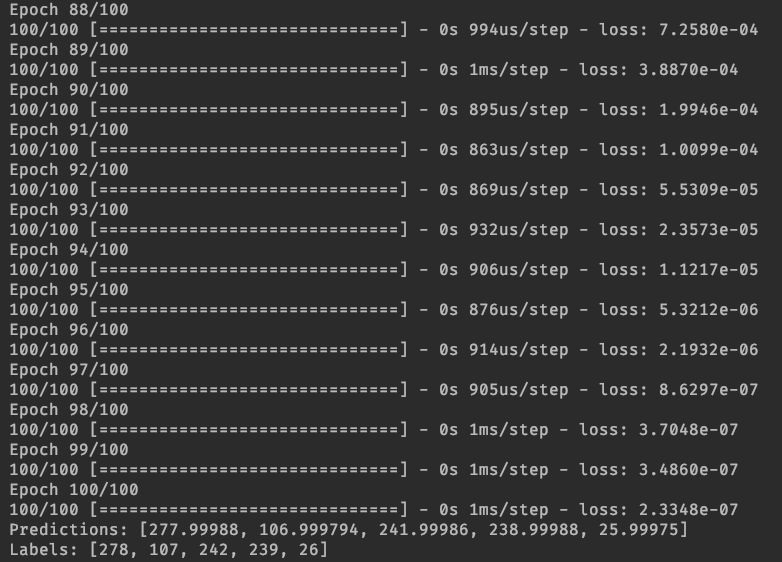
# Part one:

Model a function: f(x) = 3x+20

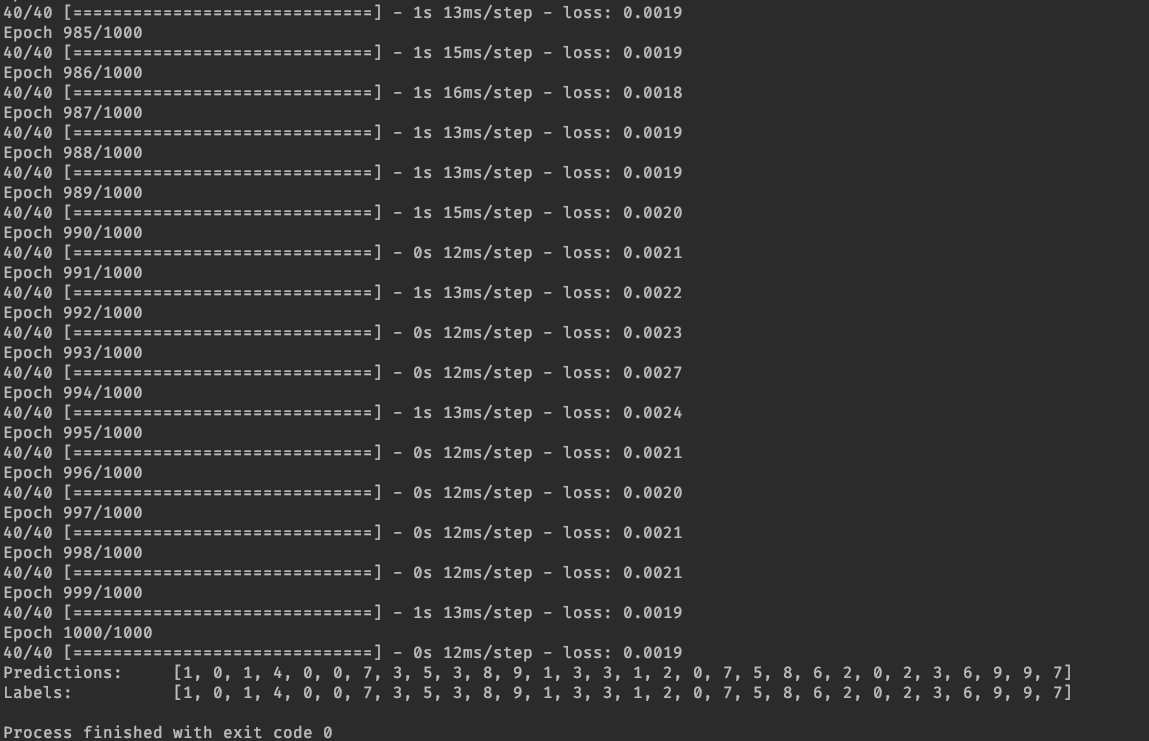
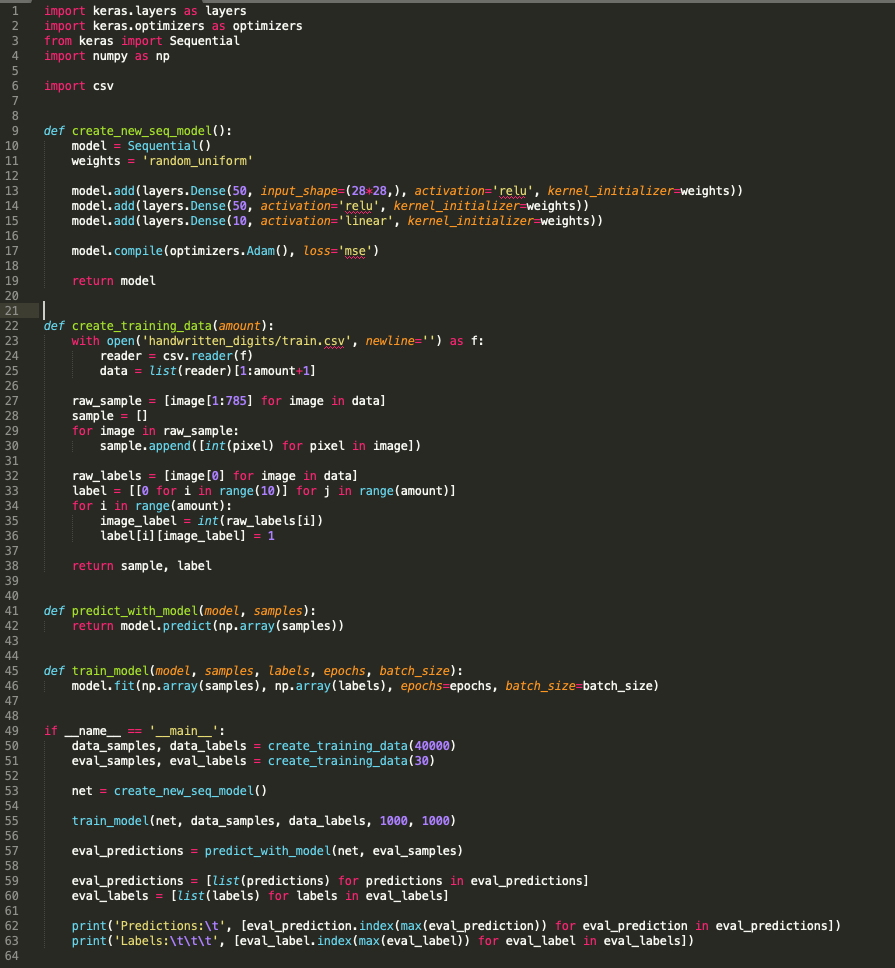


After this NN was able to approximate the function with very close precision and accuracy.

The number of nodes in each layer were the following: 1 – 50 – 50 – 1

# Part 2 reading handwritten digits.

We downloaded data from the following [link](https://www.kaggle.com/c/digit-recognizer/data) of handwritten numbers in 28x28 resolution in black and white outputted as an array of pixel values. From this we created a neural network with 748 input nodes, for the brightness of every pixel. We had like the previous example two hidden layers of 50 nodes and then an output layer with 10 nodes, one for each digit (0-9). The node with the highest value was the node that was interoperated and the neutral network’s “choice.”



We let the neural network train with 1000 epochs with a batch size of 40, this means that the training did not actually go through the entire dataset as the dataset included about 46000 samples. Nevertheless, as visible in the predictions the neural network was able to correctly label at least 30 out of the sample images. Naturally the training time of the neural network was a lot longer than the first task. In the future it is better practice to separate the evaluation samples from the training samples but in this case since the data that was trained on was not reused to a greater extent the concern for over or under fitting the data is not as relevant.