**TCSS 455 Machine Learning**

**Homework #3**

Tianyi Li

Student ID: 1827924

1. **Backpropagation:**

Consider a two-layer feedforward ANN with two inputs and , one hidden unit , and one output unit . This network has five weights (, , , , ), where w,o represents the threshold weight for unit . Initialize these weights to the values (0.1, 0.1, 0.1, 0.1, 0.1), then give their values after each of the first two training iterations of the Backpropagation algorithm. Assume learning rate , momentum , incremental weight updates, and the following training examples:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | 0 | 1 |
| 0 | 1 | 0 |

*The network:*

*The equation for* *the sigmoid activation function:*

*Training example #1: With calculate the outputs,*

*With , calculate the error of the hidden layer,*

*Hence, calculate the error terms, with and ,*

*Therefore, the new weights are:*

*Training example #2: With calculate the outputs,*

*With , calculate the error of the hidden layer,*

*Hence, calculate the error terms, with and*

*Therefore, the new weights are:*

1. **Gradient Descent Weight Update Rule for a Tanh Unit:**

Let us replace the sigmoid functionσin Figure 4.6 by the function “tanh”. Derive the new weight update rule.

*We are asked to assume the output of an unit to be*

*.*

*The proof below are borrowing ideas from Chapter 4.5.3 from the textbook:*

*For each training example , every weight is updated by adding to its :*

*Where is the error on training example , summed over all output unites in the network:*

* *= the ith input to unit*
* *= the weight associated with the ith input to unit*
* *= (the weighted sum of inputs for unit )*
* *= the output computed by unit*
* *= the target output for unit*
* *= the sigmoid function*
* *outputs = the set of units in the final layer of the network*
* *Downstream(j) = the set of units whose immediate inputs include the output of unit*

*Then, the training rule for output unit weights is*

*Now, consider only the first term:*

*The derivatives of will be 0 (zero)for an output unit , except when . Therefore we drop the summation over output unites and simply set :*

*Now, consider the second term:*

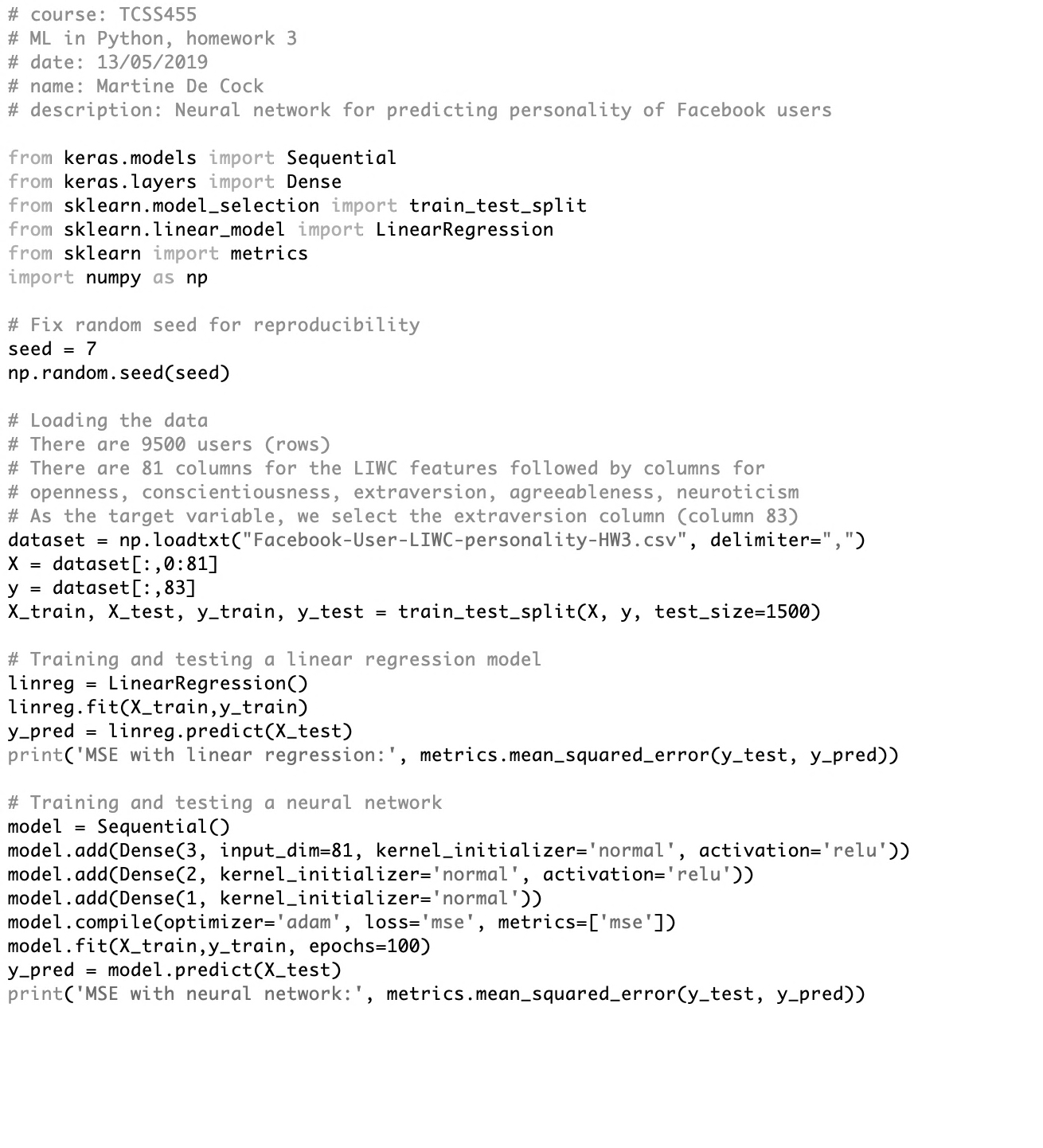
*Hence, we combine both terms:*

*Therefore,*

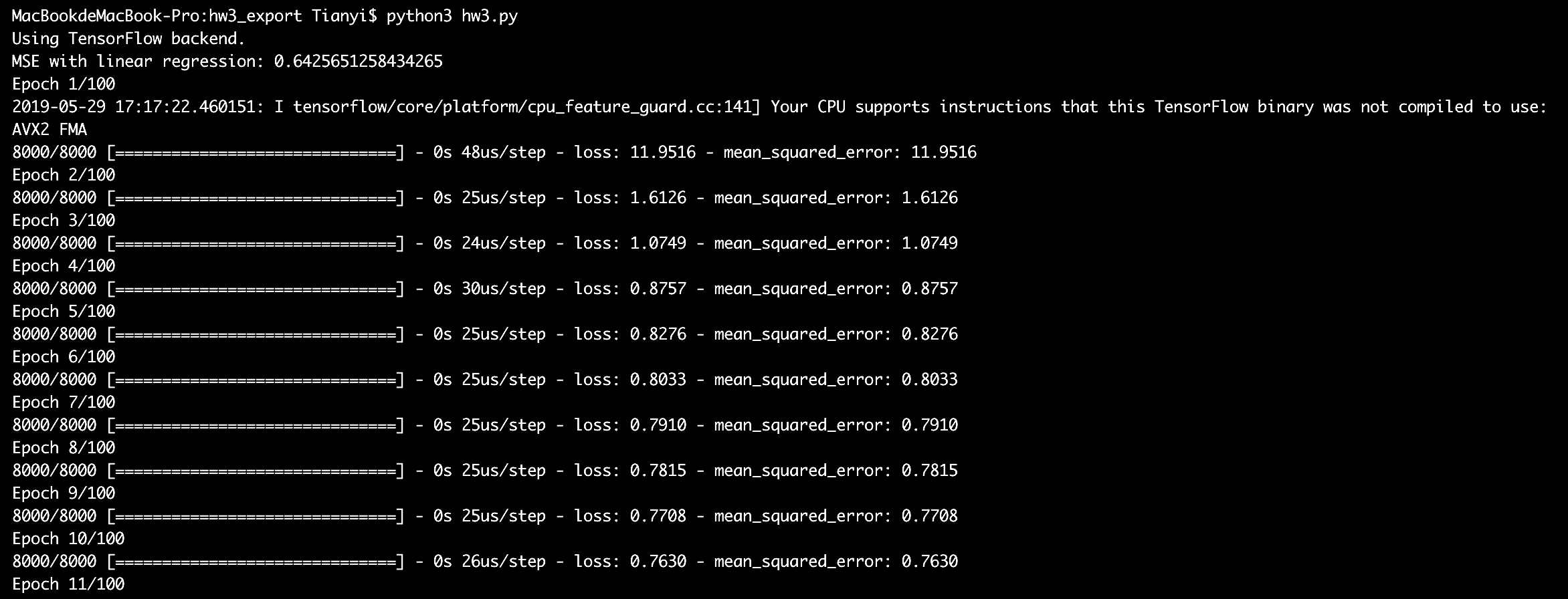
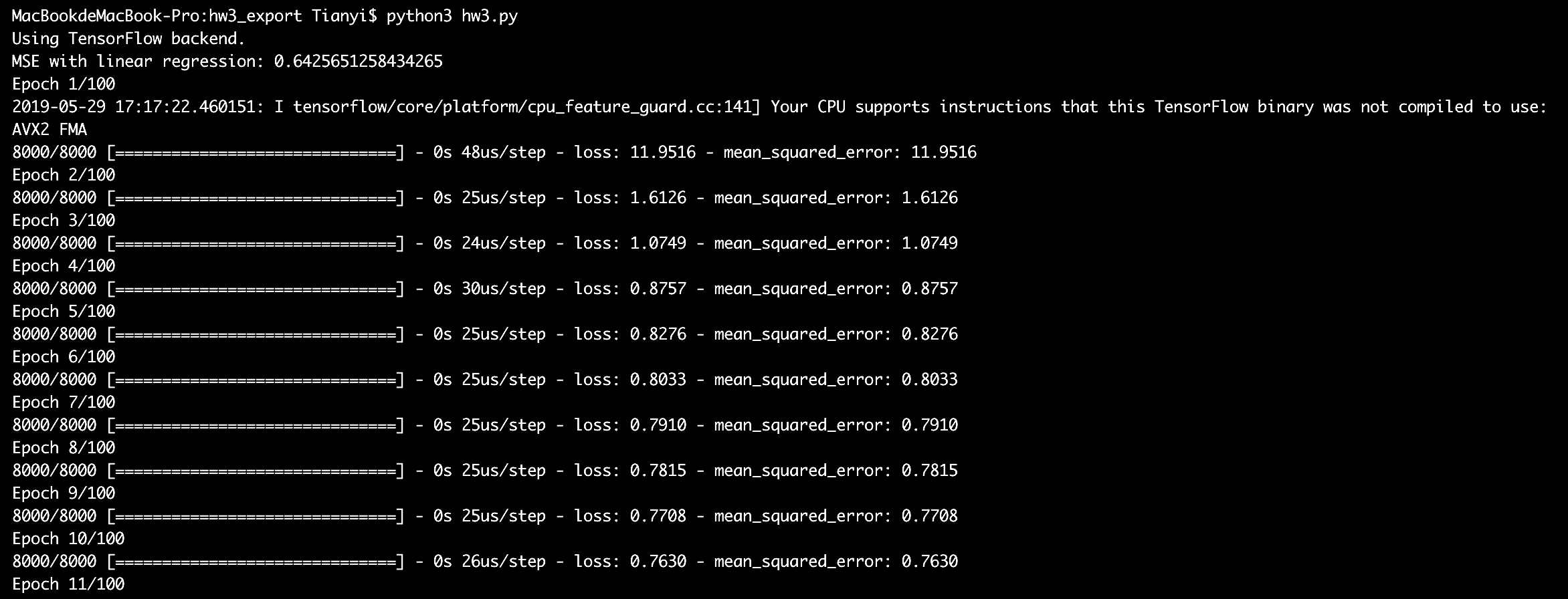
*Now, the training rule for the hidden unit weights:*

*Therefore,*

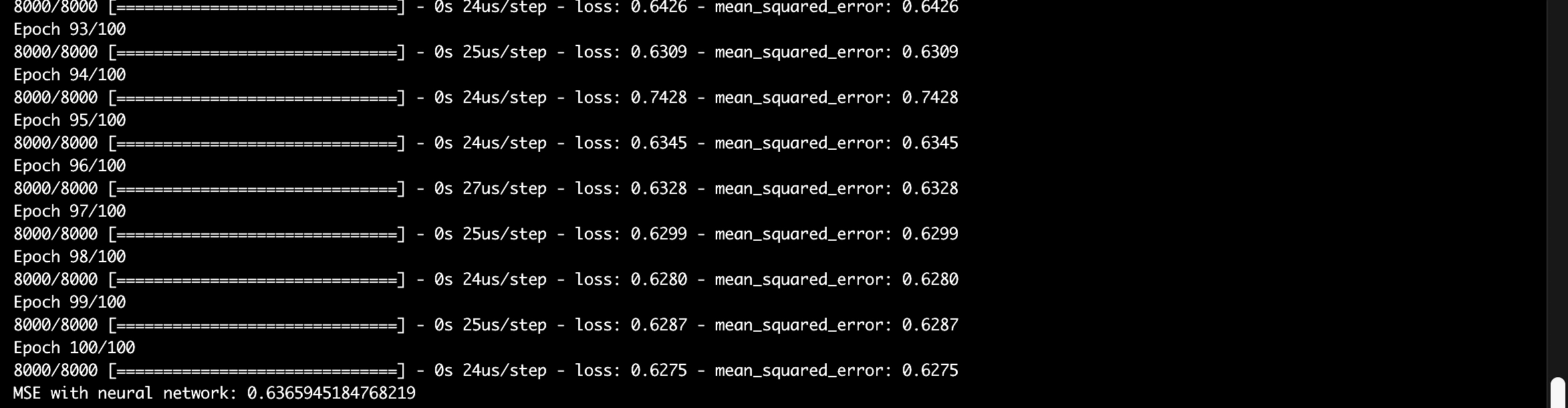
1. **Training a Neural Network with Keras:**
2. A printout of the part of the code changed:



1. Screenshots of resulting MSE’s:



…



1. A brief description of interesting aspects about training neural networks:

*I always know that the number of epochs matters a lot, but I thought it would be more the better MSE results the code will yield. It turns out that is not the case. Since MSEs and losses are conflicted: as one goes up, the other one goes down. Losses go up and down along with different numbers epochs likes curves, which leads to the variation of MSEs. Therefore, choosing a right number epochs matter.*

1. Electronic submission: *Please see on Canvas.*