CS767

SPR23

Assignment 3

Yiduo Feng

03/14/2023

**(5 points) Question 1**

Read Keras.io and list five important facts that you learn about Keras API. Add reference to the document that you used to identify each important fact.

1. The Keras API provides many models that can be used directly, which greatly reduces manpower, and we can build layers and models more conveniently and flexibly.

(https://keras.io/api/layers/)

2. The Keras API displays intuitive loss functions and evaluation results, and we can use them to draw bar graphs and optimize the code again.

(https://keras.io/api/losses/)

3. The Keras API has a callback function, through which we can monitor the model training status.

(https://keras.io/callbacks/)

4. The data import of Keras API is more extensive and flexible, even images can be imported.

(https://keras.io/api/data\_loading/)

5. Keras API provides a variety of optimizers, we save the time of programming optimizers.

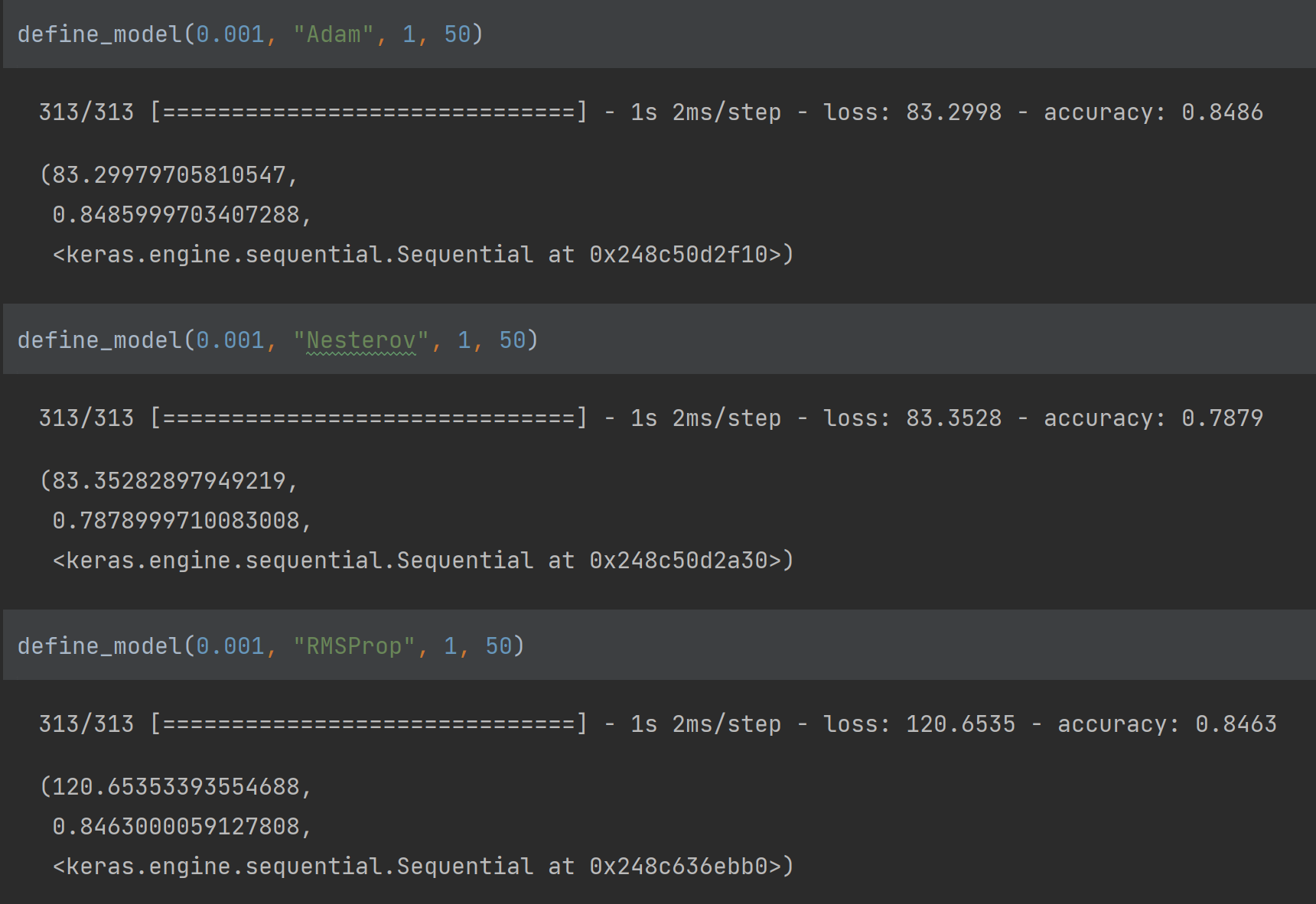
(https://keras.io/api/optimizers/)

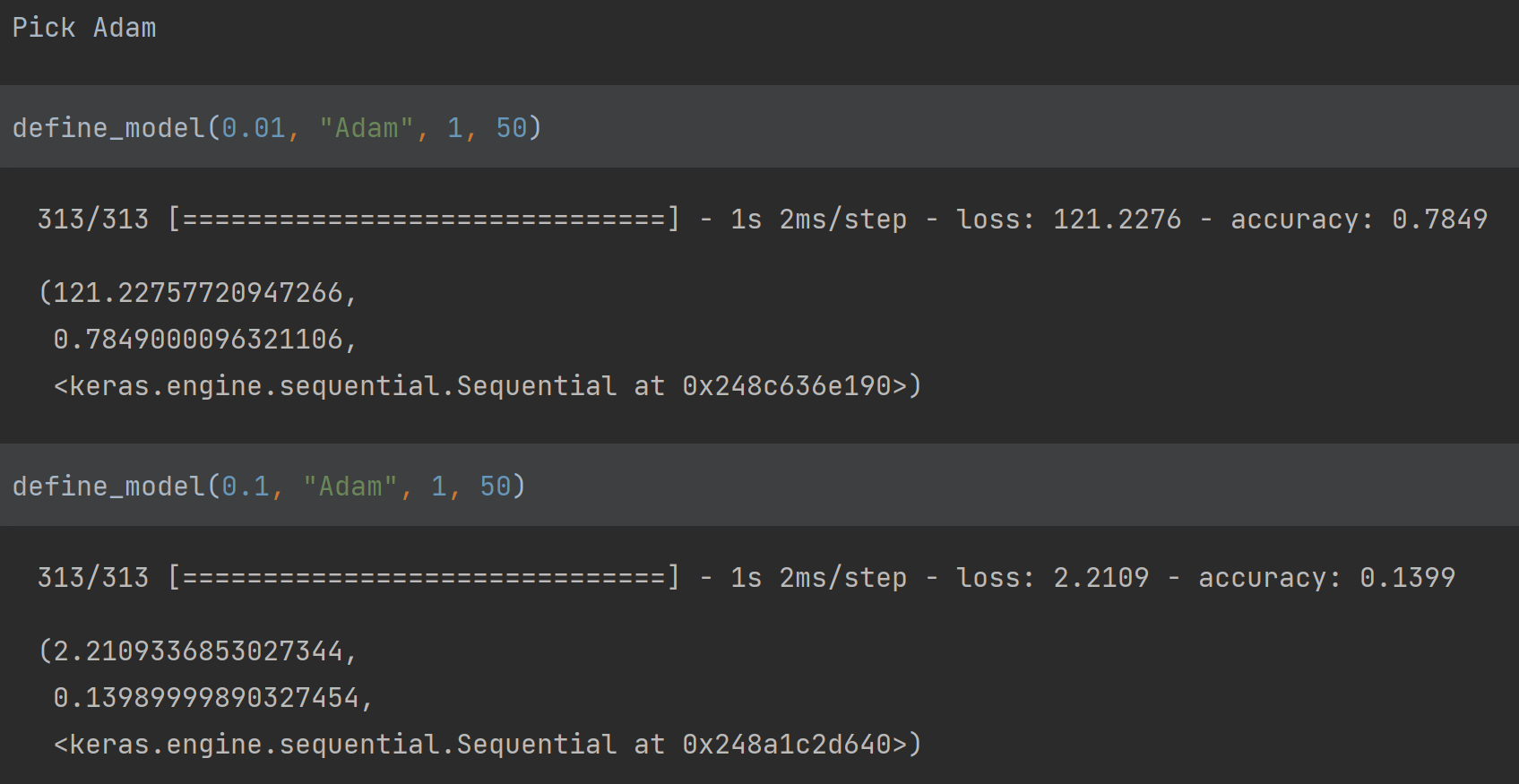
**(20 points) Question 2**

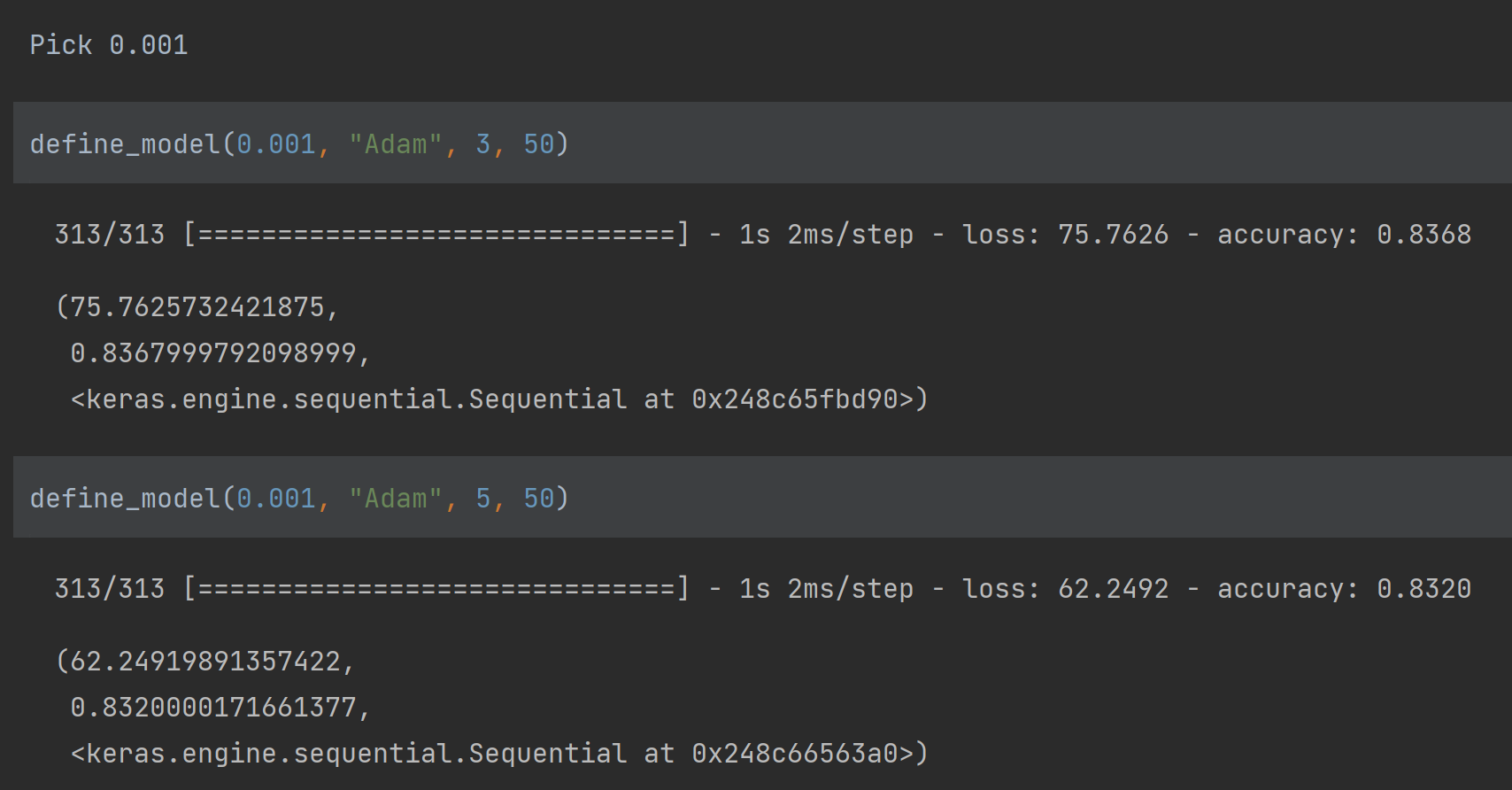
Design an MLP – Multilayer Perceptron neural network using Keras sequential API to classify MNIST fashion dataset.

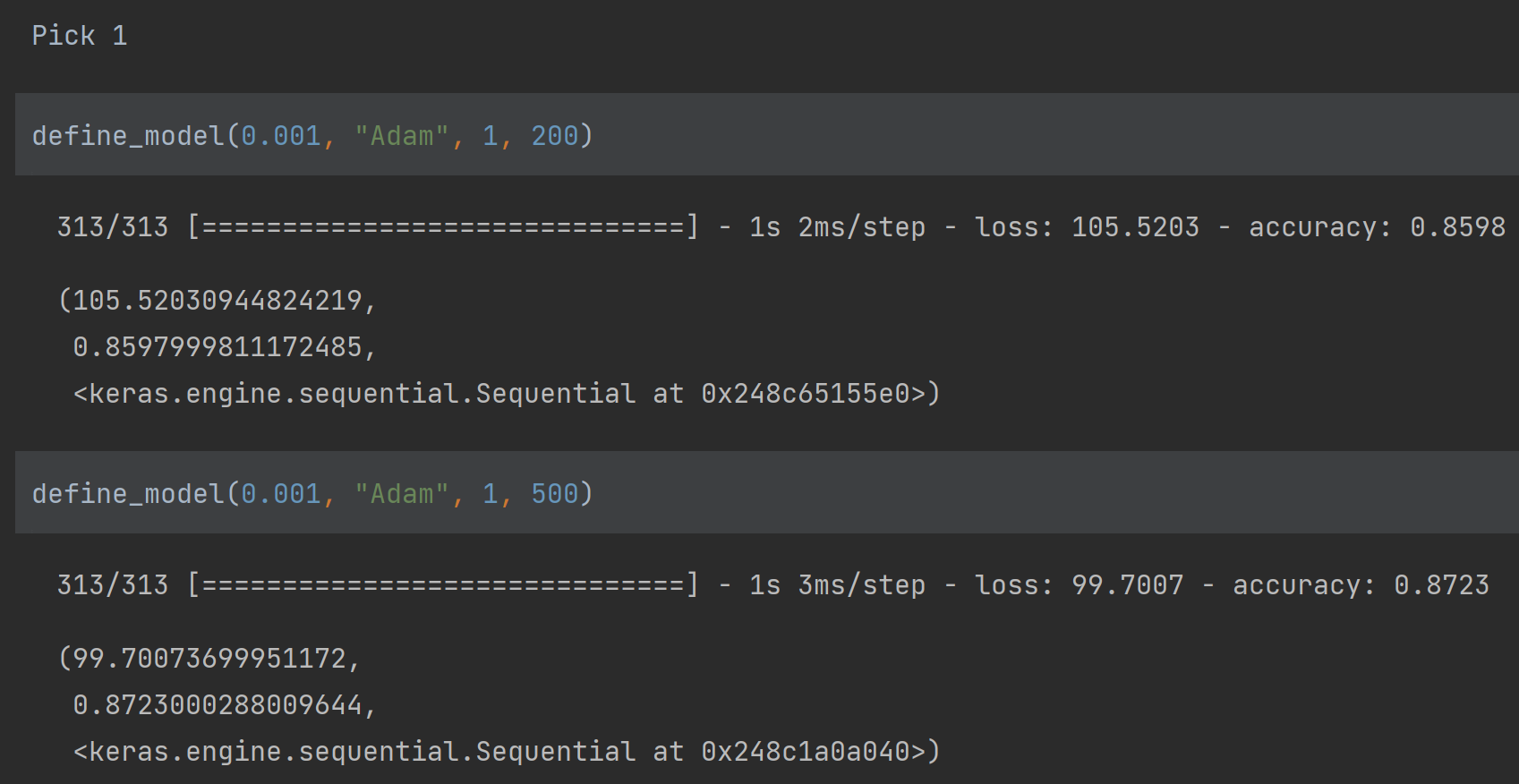
Questions:

1. Implement the best MLP neural network with following options. Provide the best set of parameters () , train your model with the best set of parameters, and plot the learning curve (error of training and test set as a function of epochs) (5 points)

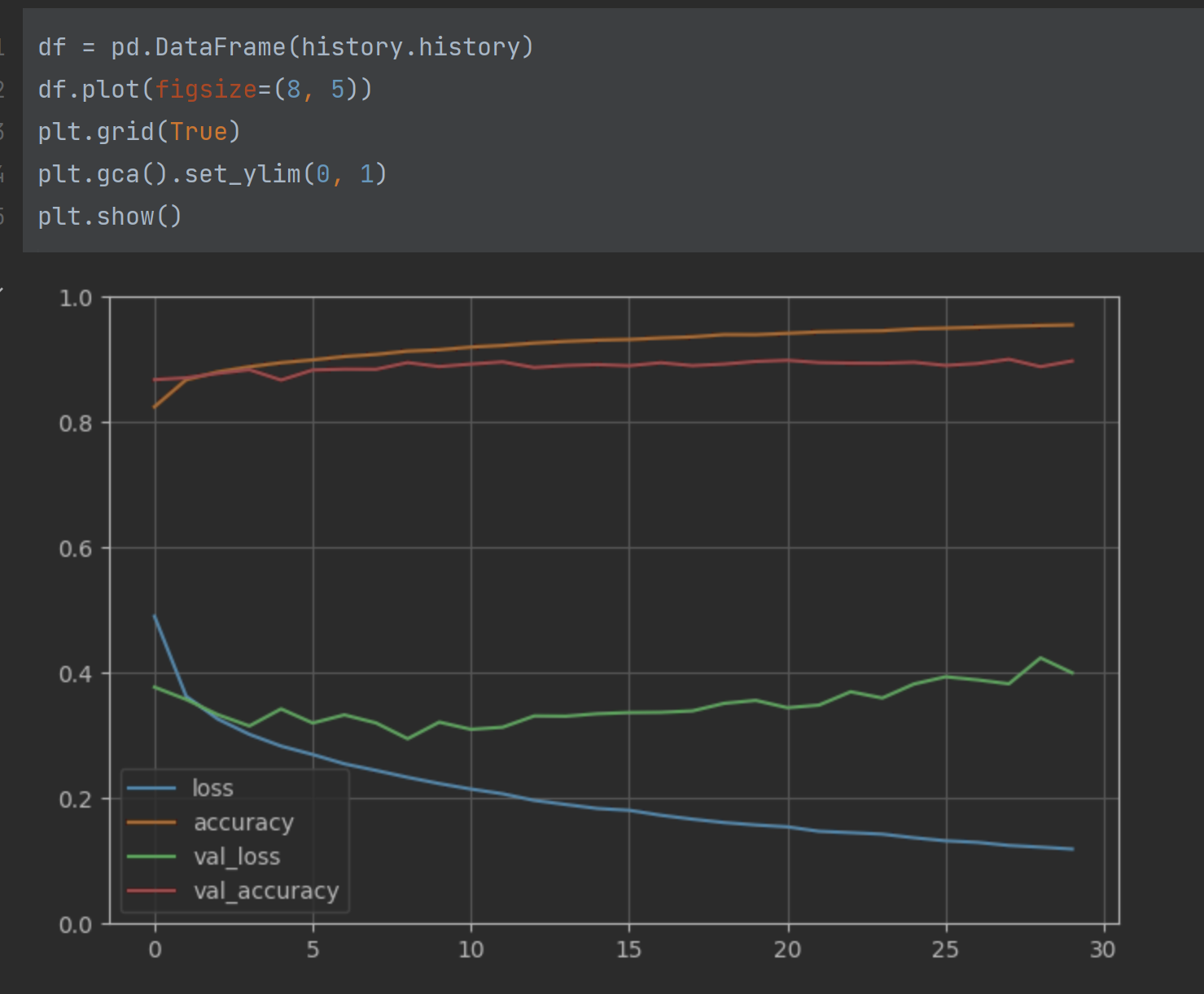






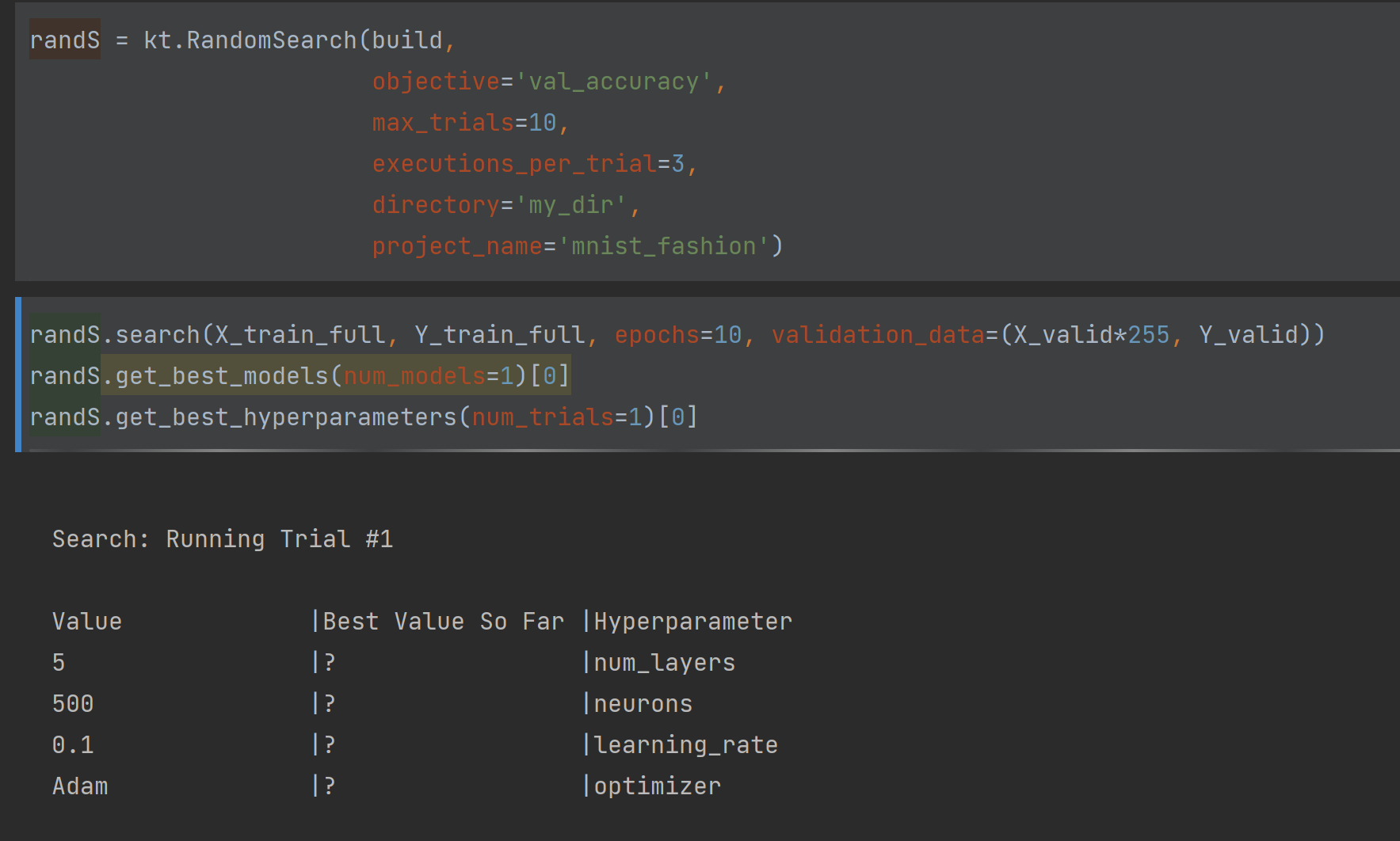


By comparing the accuracy and loss, the best set of parameters is {learning rate: 0.001, Optimizer: Adam, layers:1, neurons: 500}.



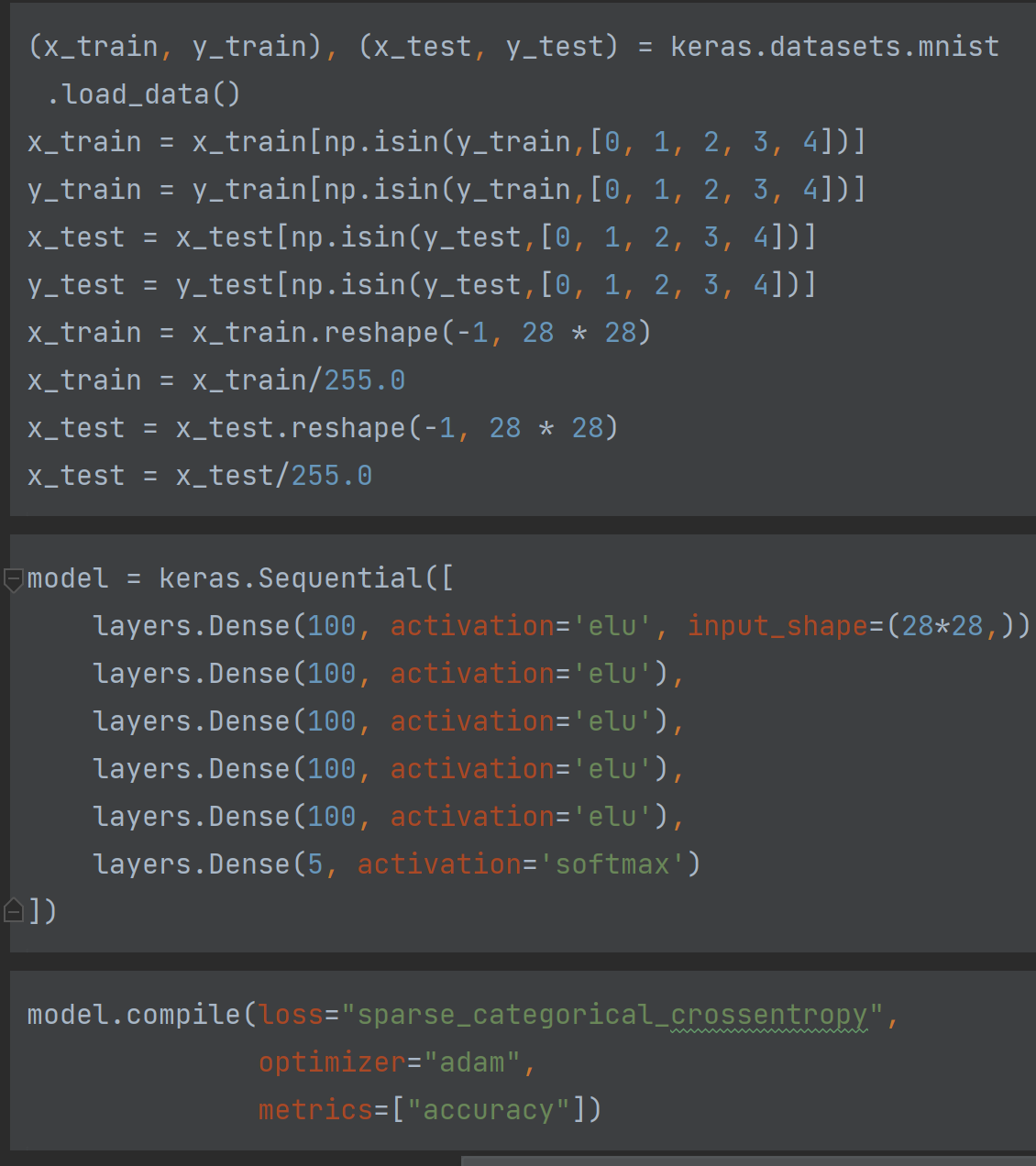
**(15 points) Question 3**

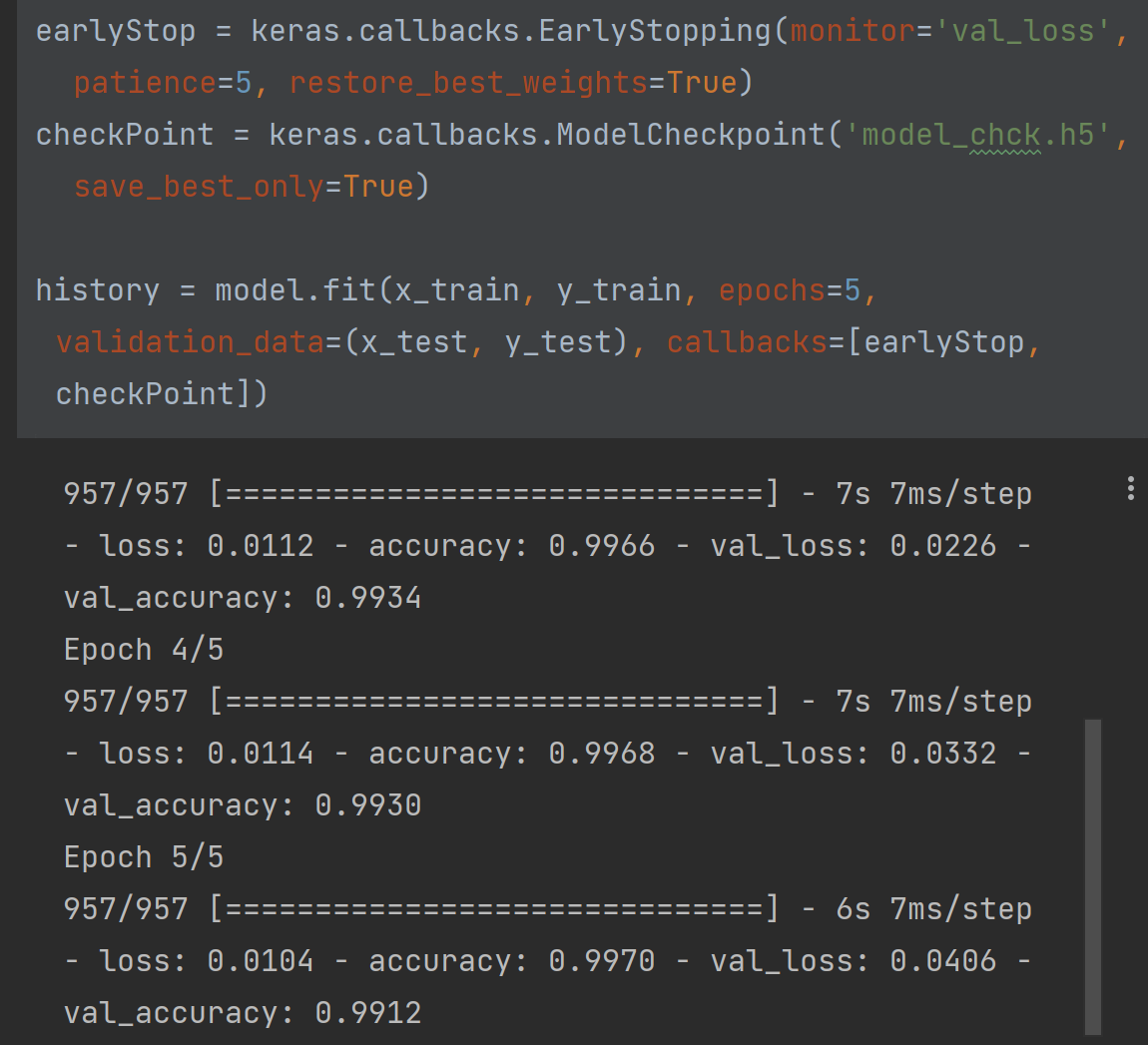
1. Find the best model for the question 2 which gives you the best performance in ten epochs using random search. (10 points)



**(30 points) Question 3**

Train a DNN on MNIST data set (hand written number from 0 to 9)

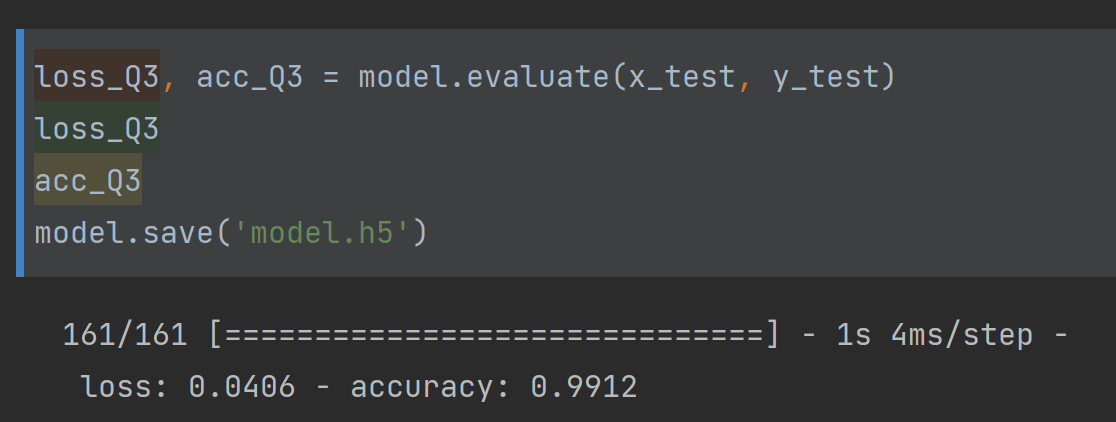




Epoch = 5: loss: 0.0104 - accuracy: 0.9970 - val\_loss: 0.0406 - val\_accuracy: 0.9912

Epoch = 10: loss: 0.0051 - accuracy: 0.9988 - val\_loss: 0.0313 - val\_accuracy: 0.9926

Epoch = 50: loss: 0.0059 - accuracy: 0.9984 - val\_loss: 0.0405 - val\_accuracy: 0.9938



As shown above, the accuracy of results is around 0.99, and the accuracy of train is also around 0.99. so it’s not overfitting.

**(30 points) Question 4**

1. Use the model in the previous question
   * The inputs are only digits 0 to 4.
   * Split data (80%, 20%) for training and testing, respectively.
   * Train the model for 10 Epochs.

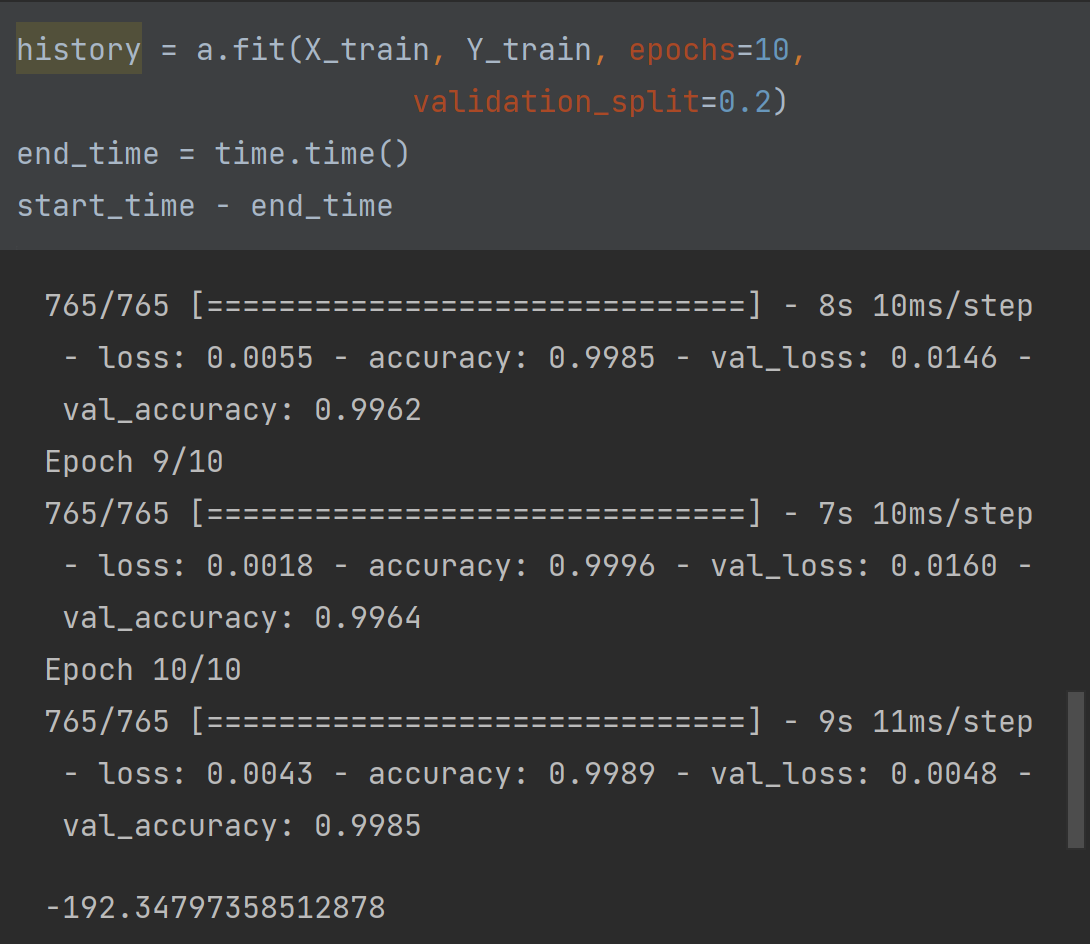
Save the weights using following command:

yourModel.save\_weights("yourKerasWeights.ckpt")

1. Use the exact model as in the previous question.
   * This time the inputs are ONLY digits 5 to 9
   * Transfer the weights from part (a)
   * Use ONLY 100 images per digit as the training set.
   * Split data (80%, 20%) for training and testing, respectively.

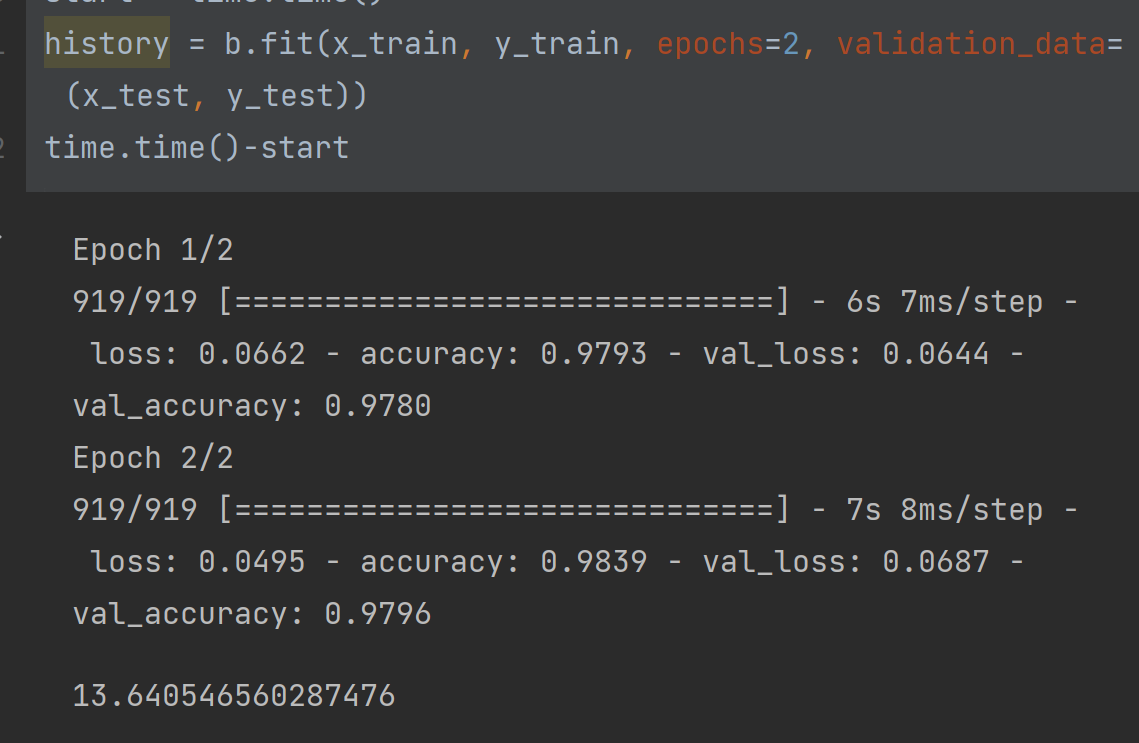
Questions:

1. How much time does it take to train the model in part a



Time = 192.34797358512878 s

1. How much time does it take to train the model in part (b) for 2 Epochs.

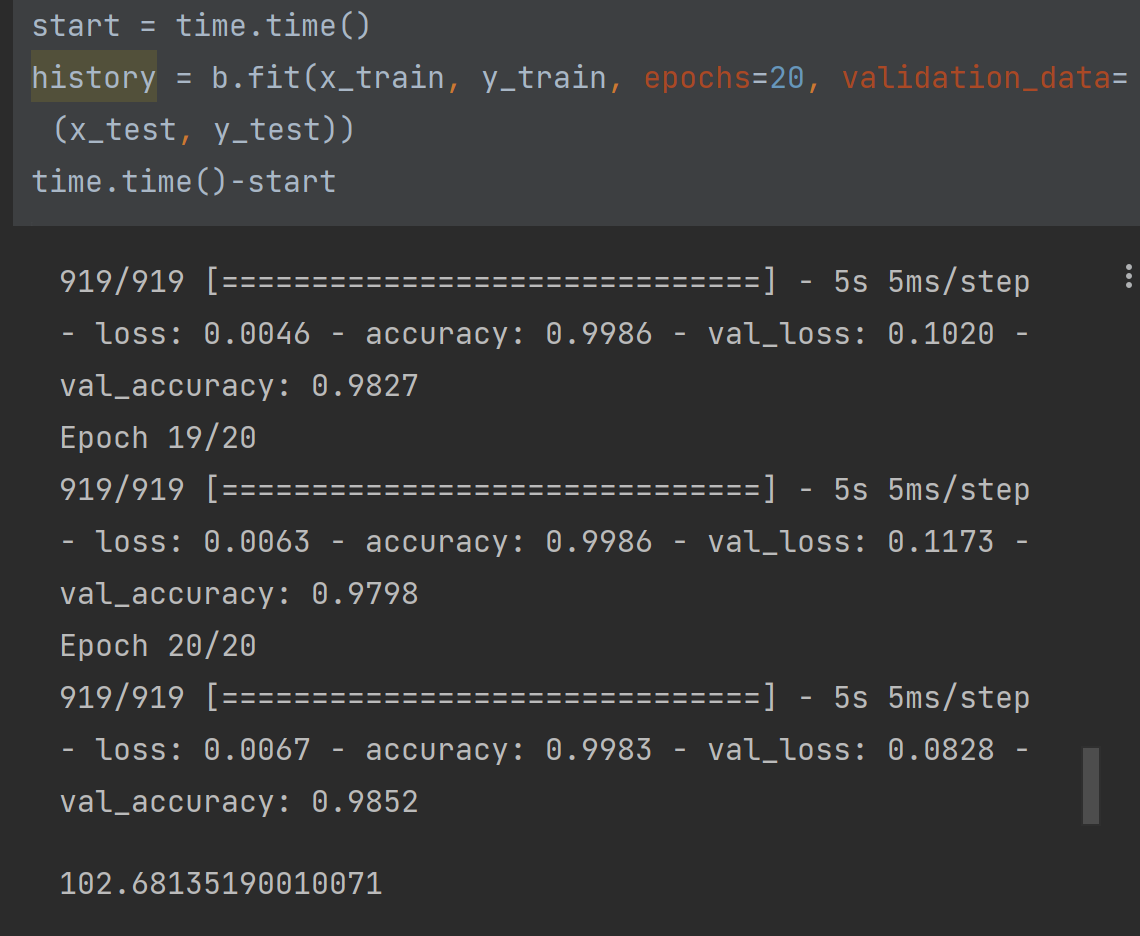


Time = 13.640546560287476 s

1. Report the accuracy of part (b) after 2 Epochs.

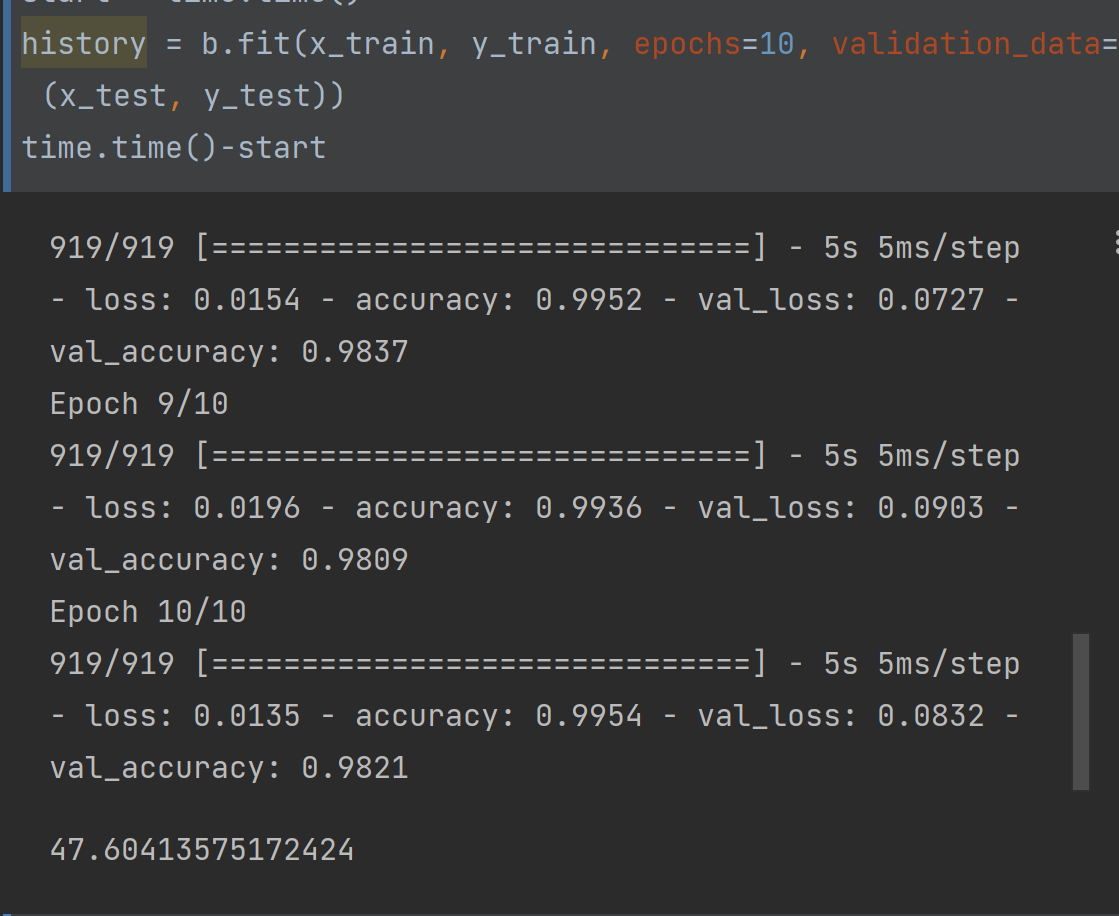
Accuracy = 0.9796

1. How much time does it take to train the model in part (b) for 20 Epochs.



Time = 102.68135190010071

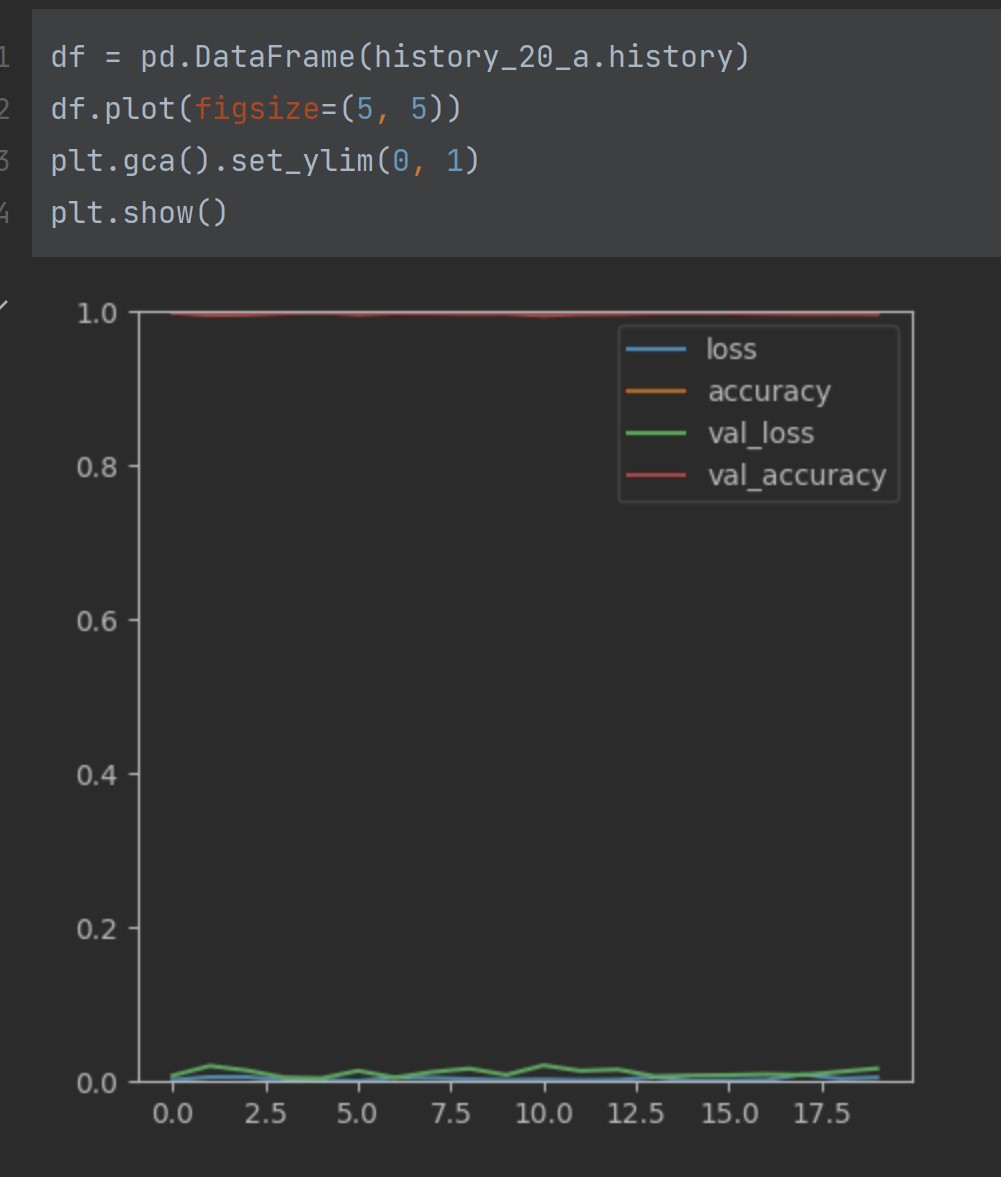
1. What is the accuracy of the model (b) after 10 Epochs?



Accuracy = 0.9821

1. Plot training and testing set accuracy for part (a) and part (b) for 20 Epochs.

A epoch20:



B epoch20:

