* draw/explain work-flow of Keras Sequential model from hands-on

1）Instantiate a Sequential model, and add layers to it one by one using “add”. In sequential models layers have input, output, input\_shape and output\_shape. We can set layer weights with layer.set\_weights(weights). Each layer has a defining configuration, layer.get\_config()

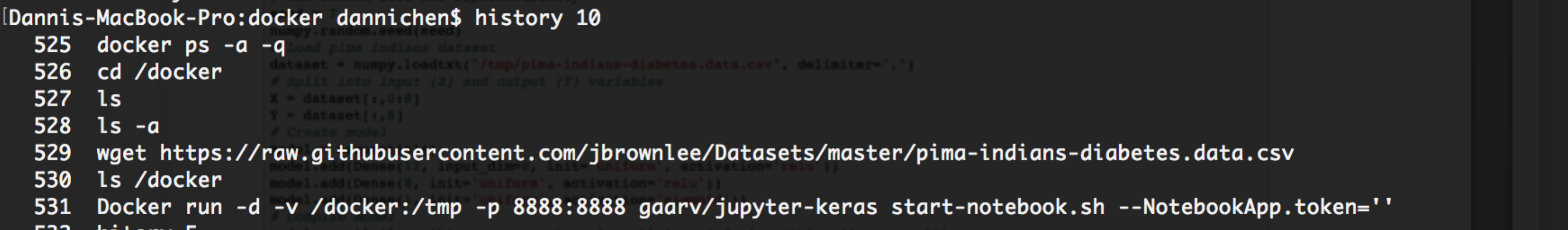
2) Compile the model with a loss function, an optimizer and optional evaluation metrics. When setting up a sequential model, we need import loss function from loss module. For the optimizer, we need load optimizers from keras.optimizers.

3) Use data to fit the model, specifying the batch\_size, number of epochs and also the validation\_data.

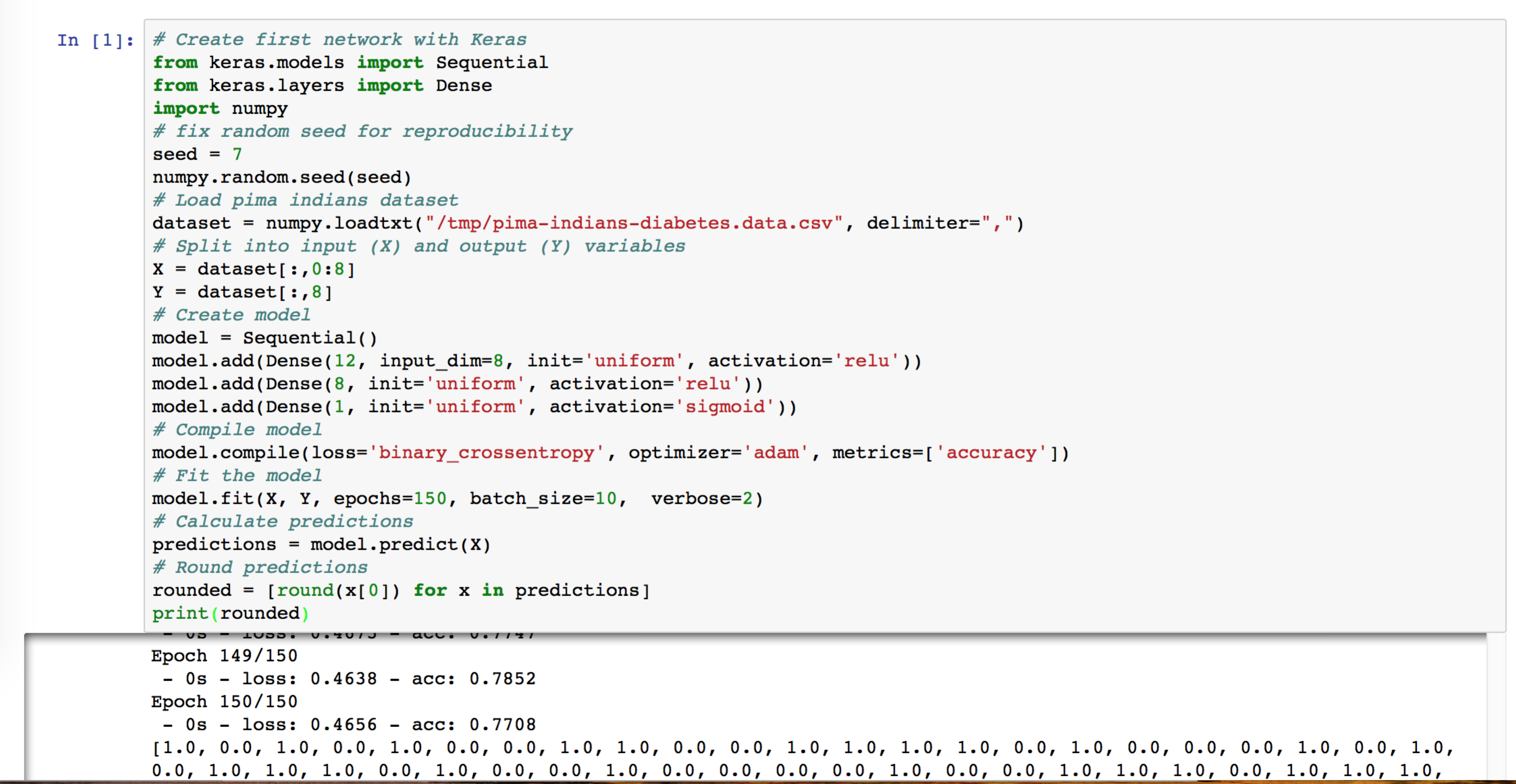
4) evaluate model on the test data, persist or deploy model, etc. Then forecasting could be done using predict function.

* follow above hands-on, run Keras from Jupyter Notebook

Here are the command lines map the data to gaarv/jupyter-keras image and start a new docker.

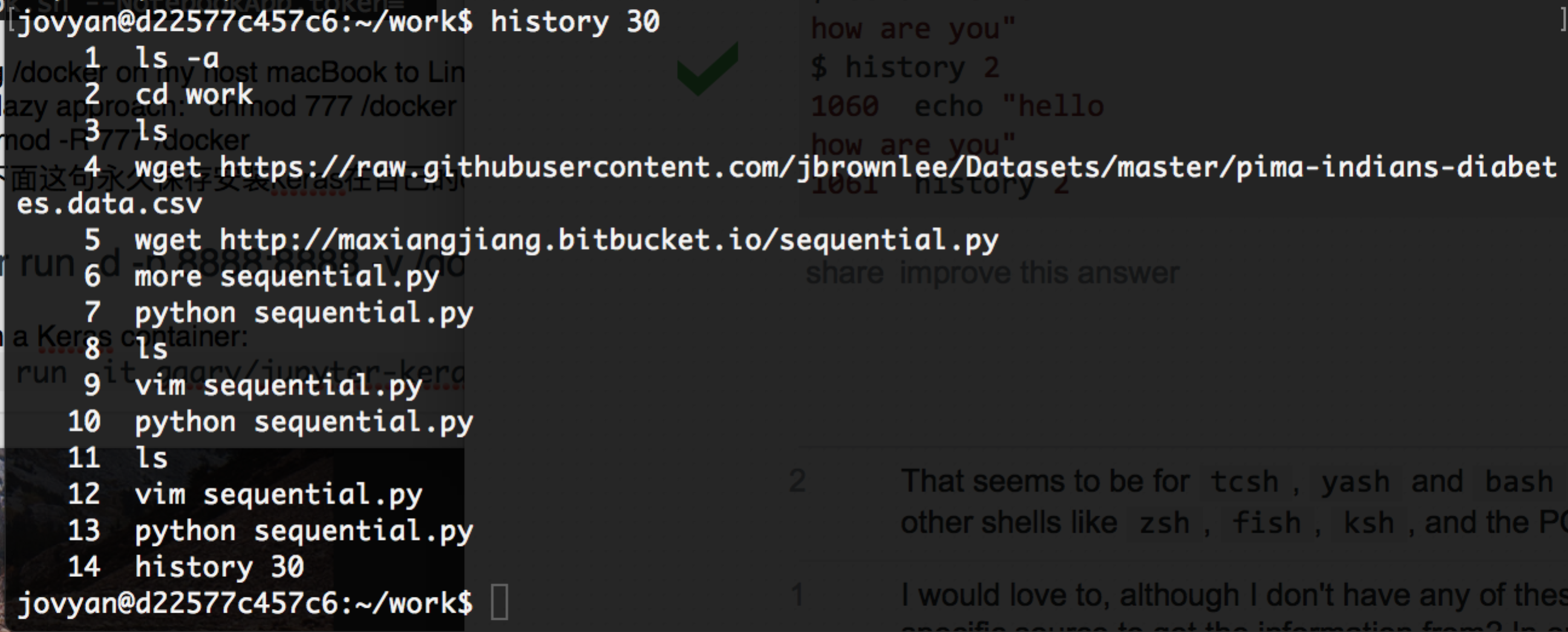


Belows are the python notebook code and results.

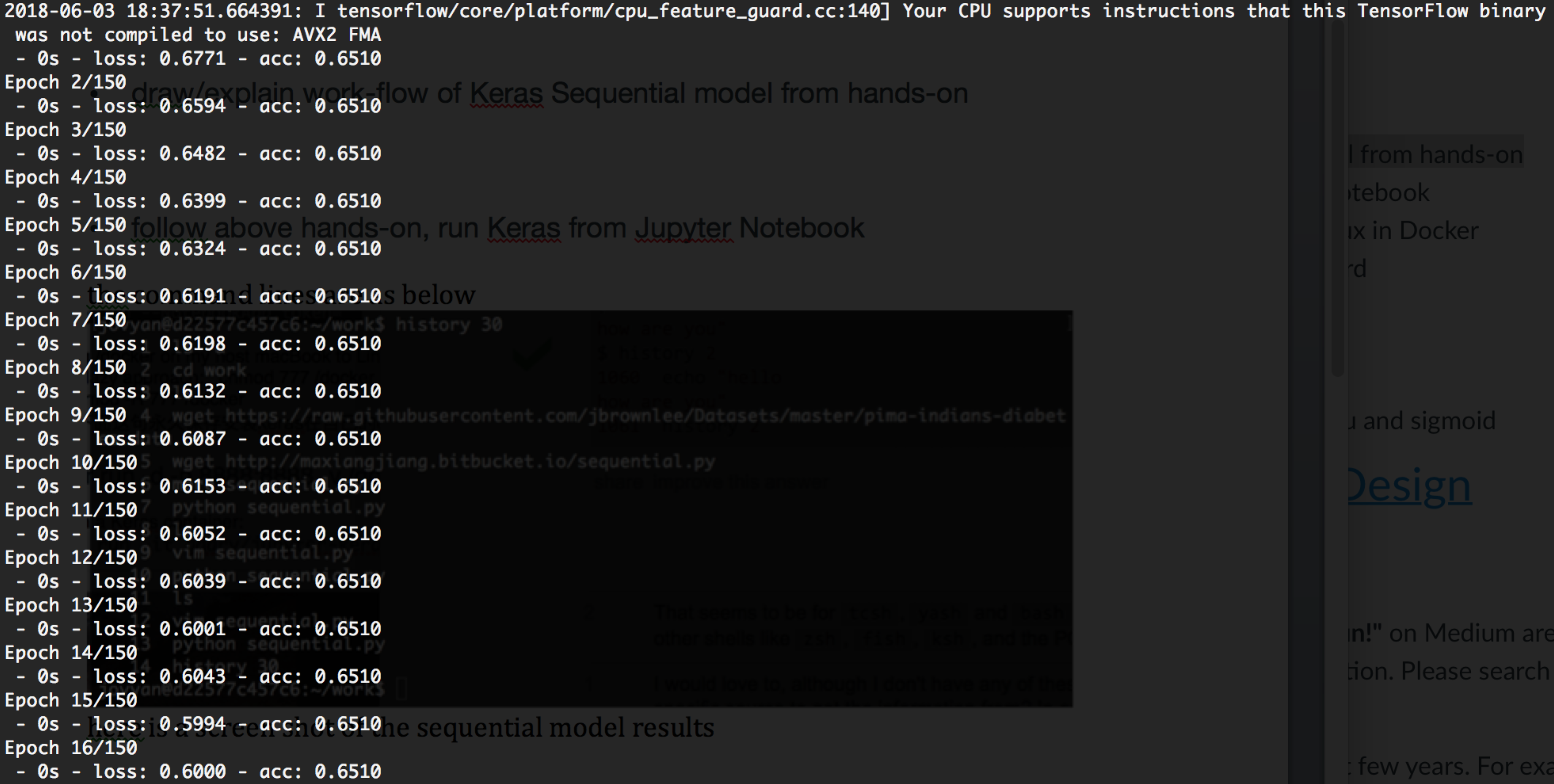


* follow above hands-on, run Keras from within Linux in Docker

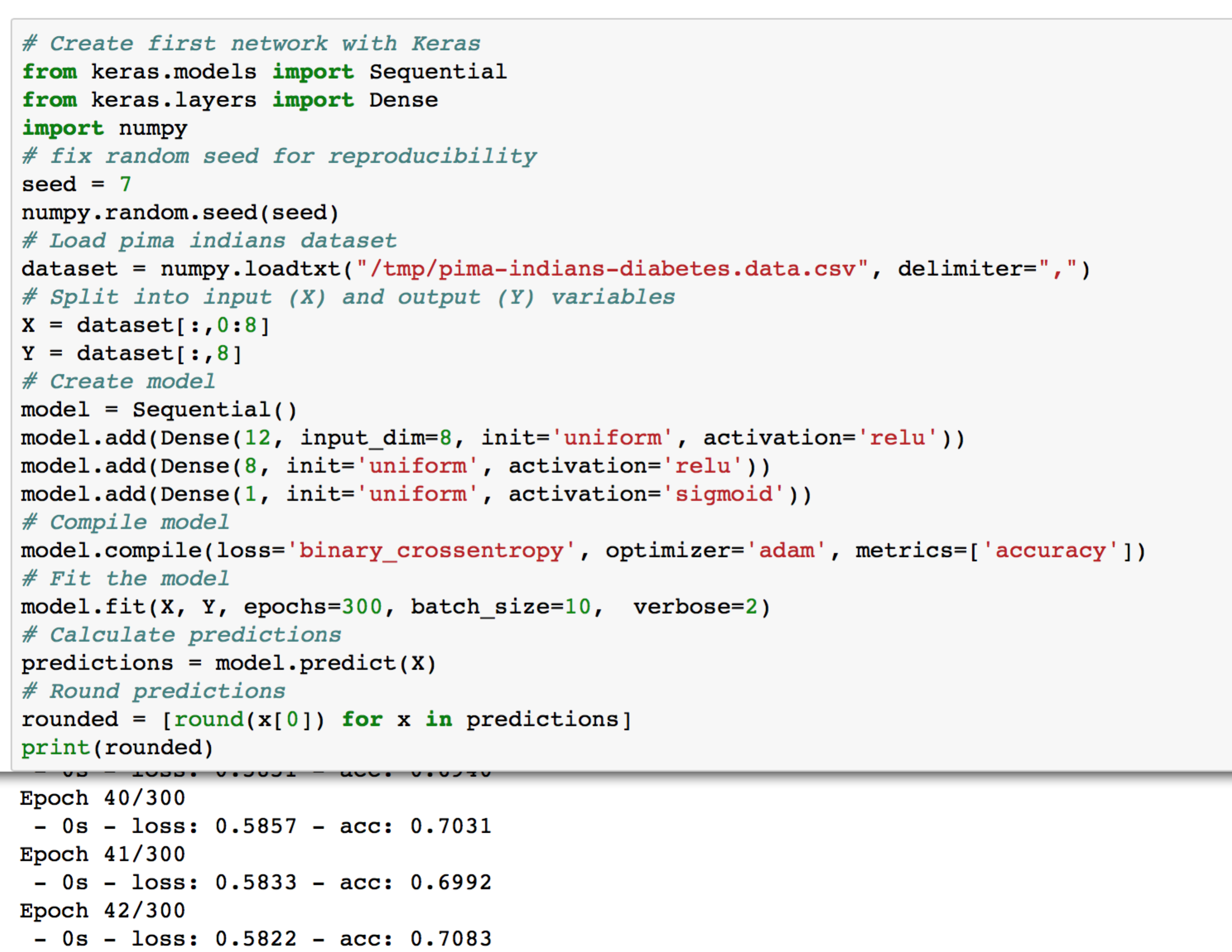
The command lines are as below.



Here is a screen shot of the sequential model results.



* focus on coding by making changes
  + change epochs and compare results，change batch\_size and compare results



* + describe the difference between activation relu and sigmoid

The Sigmoid Function curve looks like a S-shape. The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points.

The function is monotonic but function’s derivative is not.

The logistic sigmoid function can cause a neural network to get stuck at the training time.

The softmax function is a more generalized logistic activation function which is used for multiclass classification.

The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning. ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero. Range: [ 0 to infinity)

The function and its derivative both are monotonic.

But the issue is that all the negative values become zero immediately which decreases the ability of the model to fit or train from the data properly. That means any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately.