HW5 CS541 Report

Q1) Gradient updates over several epochs:

```
[9]
log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1)
checkpointer = ModelCheckpoint(filepath='model.weights.best.hdf5', verbose=1, save_best_only=True)
model.fit(x_train,
          y_train,
          batch_size=64,
          epochs=10,
          validation_data=(x_valid, y_valid),
          callbacks=[checkpointer,tensorboard_callback])
```

Python

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/10
859/860 [============================>.] - ETA: 0s - loss: 0.6101 - accuracy: 0.7749
Epoch 1: val_loss improved from inf to 0.39379, saving model to model.weights.best.hdf5
860/860 [==============================] - 51s 58ms/step - loss: 0.6100 - accuracy: 0.7749 - val_loss: 0.3938 - val_accuracy: 0.8572
Epoch 2/10
859/860 [============================>.] - ETA: 0s - loss: 0.4282 - accuracy: 0.8456
Epoch 2: val_loss improved from 0.39379 to 0.32585, saving model to model.weights.best.hdf5
860/860 [==============================] - 48s 56ms/step - loss: 0.4282 - accuracy: 0.8456 - val_loss: 0.3259 - val_accuracy: 0.8888
Epoch 3/10
859/860 [============================>.] - ETA: 0s - loss: 0.3810 - accuracy: 0.8605
Epoch 3: val_loss improved from 0.32585 to 0.30976, saving model to model.weights.best.hdf5
860/860 [==============================] - 46s 53ms/step - loss: 0.3811 - accuracy: 0.8604 - val_loss: 0.3098 - val_accuracy: 0.8862
Epoch 4/10
859/860 [============================>.] - ETA: 0s - loss: 0.3547 - accuracy: 0.8685
Epoch 4: val_loss improved from 0.30976 to 0.28613, saving model to model.weights.best.hdf5
860/860 [==============================] - 47s 55ms/step - loss: 0.3548 - accuracy: 0.8685 - val_loss: 0.2861 - val_accuracy: 0.8944
Epoch 5/10
859/860 [============================>.] - ETA: 0s - loss: 0.3342 - accuracy: 0.8786
Epoch 5: val_loss improved from 0.28613 to 0.26990, saving model to model.weights.best.hdf5
860/860 [==============================] - 47s 54ms/step - loss: 0.3342 - accuracy: 0.8786 - val_loss: 0.2699 - val_accuracy: 0.8996
Epoch 6/10
859/860 [============================>.] - ETA: 0s - loss: 0.3149 - accuracy: 0.8833
Epoch 6: val_loss improved from 0.26990 to 0.26065, saving model to model.weights.best.hdf5
860/860 [==============================] - 42s 49ms/step - loss: 0.3148 - accuracy: 0.8833 - val_loss: 0.2606 - val_accuracy: 0.9028
Epoch 7/10
```

Figure 1

Accuracy (>90%):

```
Epoch 10/10
860/860 [==============================] - ETA: 0s - loss: 0.2733 - accuracy: 0.8989
Epoch 10: val_loss did not improve from 0.23637
860/860 [==============================] - 44s 51ms/step - loss: 0.2733 - accuracy: 0.8989 - val_loss: 0.2539 - val_accuracy: 0.9050

<keras.callbacks.History at 0x167e2142d40>
```

```
[10]
model.load_weights('model.weights.best.hdf5')
```

Python

```
[11]
score = model.evaluate(x_test, y_test, verbose=0)

# Print test accuracy
print('\n', 'Test accuracy:', score[1])
```

Python

```
Test accuracy: 0.9071000218391418
```

Figure 2

Q2)

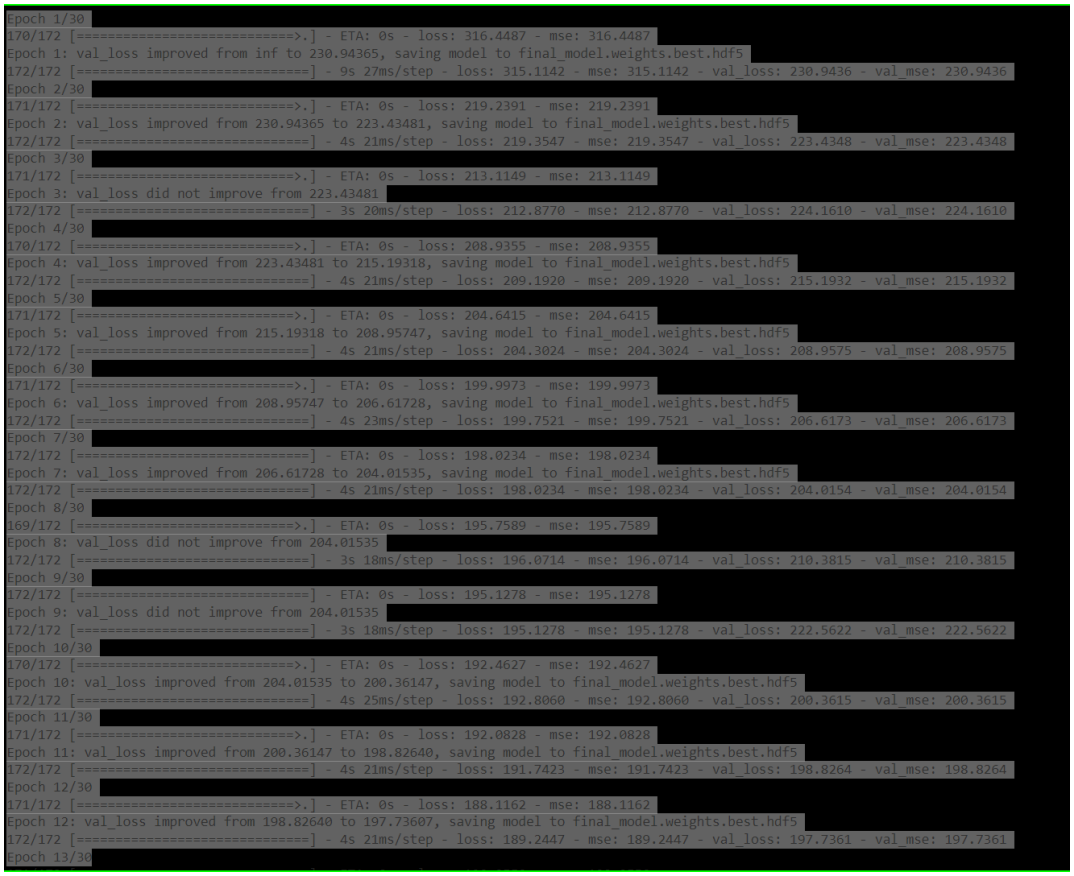

Figure 3

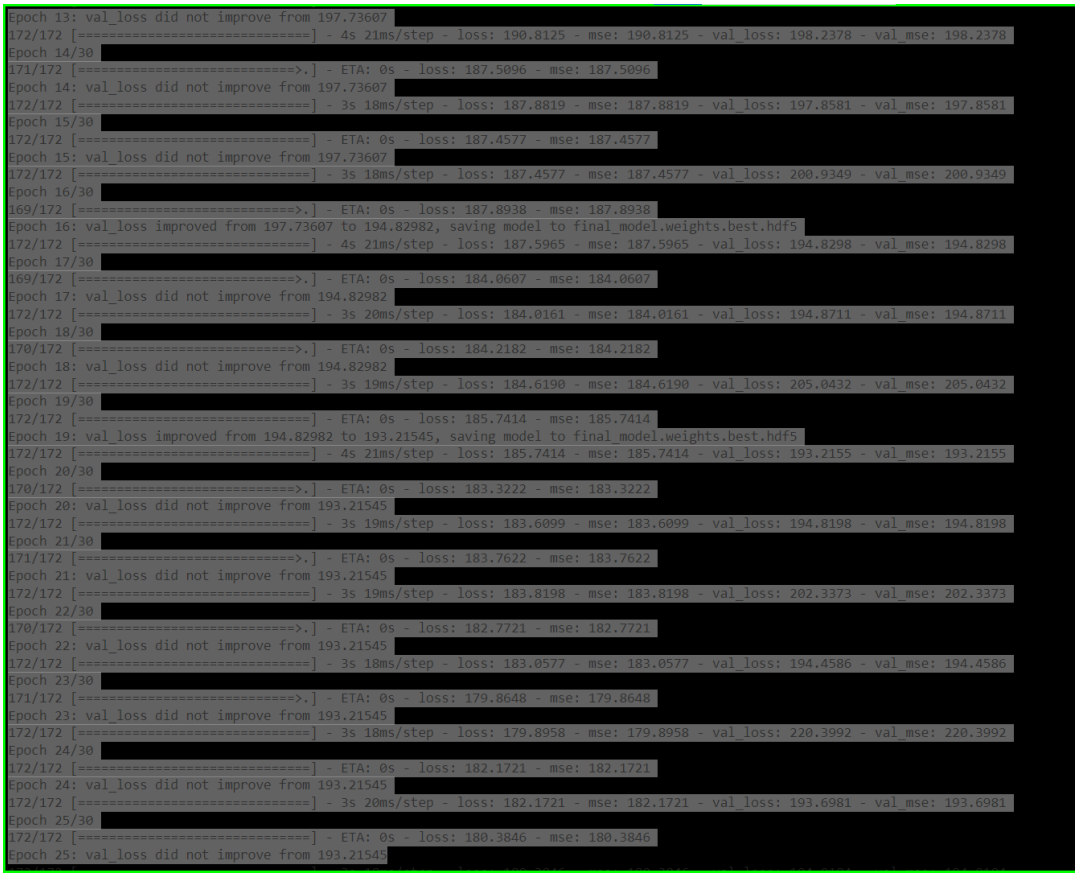

Figure 4

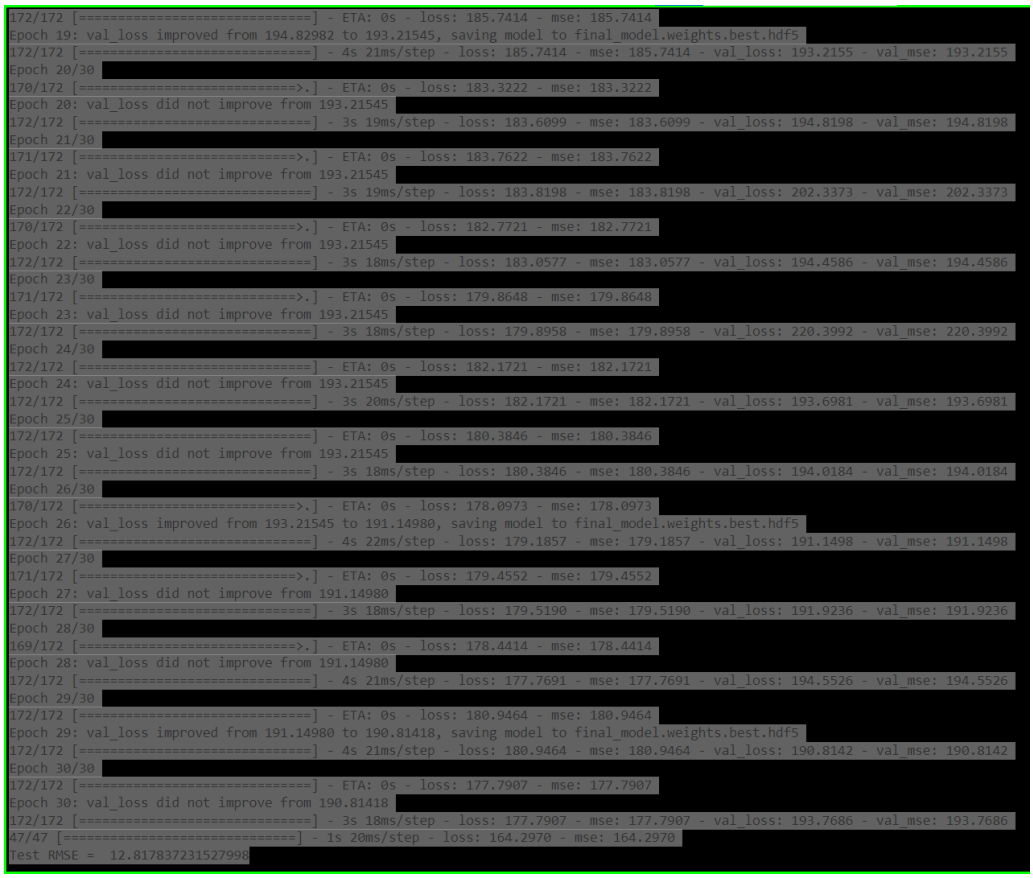

Figure 5

From the above screenshots, the testing RMSE can be seen to be 12.81 (Figure 5, last line)

Training MSE is 177.79, corresponding to RMSE of 13.33

Validation MSE is 193.76 corresponding to an RMSE of 13.92