Tensor Flow Class

18643049 Shohei HAGANE

Objective

Identifying 5 kinds of human's motion by using CNN

Motions

I defined 5 motion as I showed below. Each motion is started at sitting position.

a. Still … stay sit still and keep position

b. Lean left … lean left and return to motion 1 position

c. Lean right …lean right and return to motion 1 position

d. Lean behind … lean behind and return to motion 1 position

e. Standup and sit … standup and sit again

Pictures of actual motion is shown in Fig 1.

a. Still b. Lean Left c. Lean Right

d. Lean Behind e. Standup and Sit

Figure 1. Motions

Data acquisition

The time series data was obtained from Wii balance board (WBB). Firstly, the participant was told to sit on WBB and hold still while WBB was being calibrated. The measurement process took 15 second without calibration. After the calibration, the Participant was told to sit on WBB still for 5 second and then start motion. After the motion, the participant returned to the first position and waited still until the operator finished counting 15 second. The motion has 5 kinds as mentioned above, and each motion was repeated 5 times each. Thus, 25 sets of data was obtained from a participant. In this time, 3 people participate this experiment. Therefor, 75 sets of data was obtained totally.

Data processing

Data processing was operated by using Python 3.5 and PyCharm. Obtained data was processed to change its shape in order to feed CNN. After loading original data, its shape was 325~400×7×75. The length of data was not uniform (325~400) because of WBB's feature. In order to make length of data equal, end of each data was omitted and every data has length of 325 after all. Also, since the first 5 second is just data of sitting still, the first 100 points of data was omitted. Moreover, because the "time" elements is not needed for machine learning, "time" elements was also omitted. Therefore, finally loaded data has shape of 225×6×75. Furthermore, this set of data had to be divided and changed its shape as 60×225×7×1 for "train_x" and 15×225×7×1 for "test_x". In this case, from every participants, every one motion was picked up from each motion for the test data and remain was used as training data. Besides the test data sets and train data sets, labeling was done in style of one_hots expression. Those process were operated in functions "get_WBB_data", and "get_Data" in the submitted script.

Result of CNN

The processed data was fed to CNN. The script that distributed in class was utilized mainly. The adjustable parameters was,

1. Size of kernel for convolution … 10×1 , 5×1
2. Strides of kernel for convolution … 2×1, 2×1
3. Size of kernel for pooling … 6×1, 3×1
4. Strides of kernel for pooling … 1×1, 1×1
5. Number of convolution and pooling … 2
6. Number of full connected layer … 7
7. Number of nodes in full connected layer … 8192, 4096, 2048, 1024, 512, 64, 16
8. Learning rate … 10^-6

9. Kinds of activation function …ReLu

10. Number of epoch … 50

The result is shown below

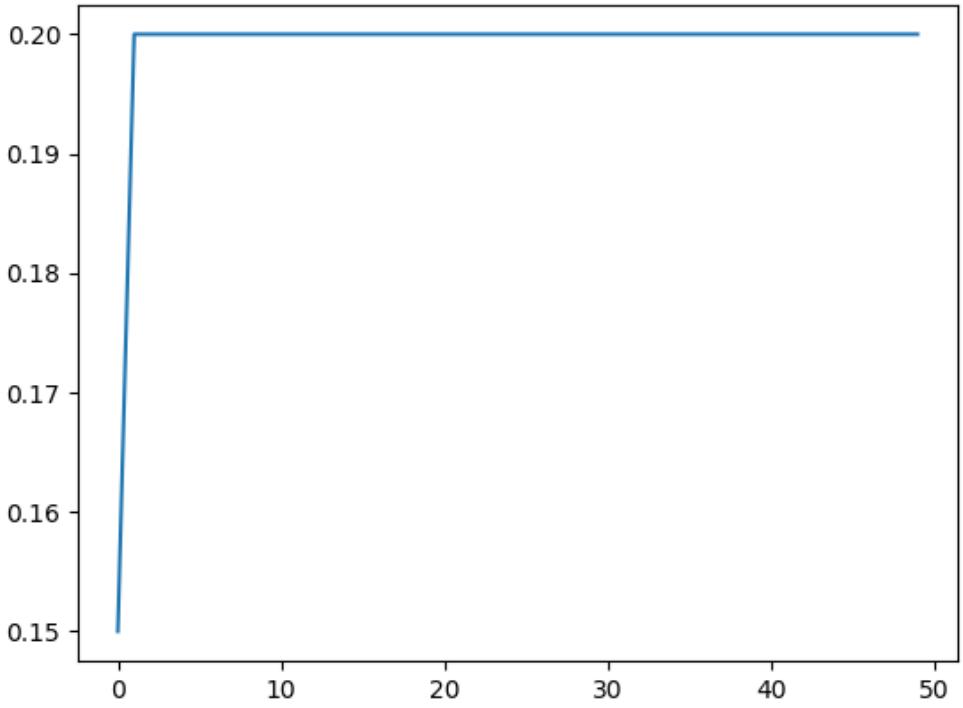

Figure 2. History of accuracy

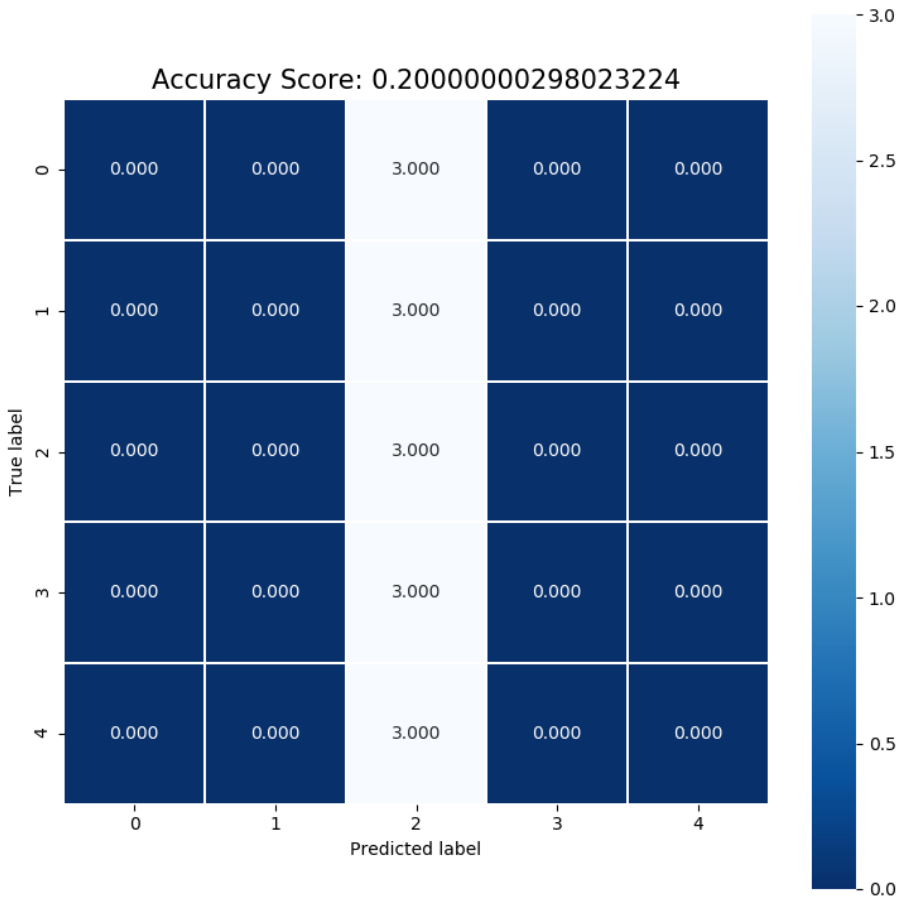

Figure 3. Confusion matrix

As you can see Fig2, accuracy was stuck at 20% somehow. Also, as showed in Fig3, all motion was classified as motion “c”. The possible explanation was the way reshaping the data was not adequate. Since tf.layers.conv2d was used in the script, kernel size may be better to have square shape, not rectangular. In order to use square shape kernel effectively, data shape should be close to square too. Currently, the data shape is 225 × 6, so if that data was reshaped like 37 × 37 somehow, the result may be different. Furthermore, segmentation will make data better. Also, “dropout” is one option to use to make result better.