Huawei coding exam report Olivier Valery

Problem Statement

Consider the Motion Capture Hand Postures dataset, whose description is here `https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures`.

1. Build a shallow NN-classifier that takes as input attributes X0, Y0, Z0, , X4, Y4, Z4, and returns the Class as the output
2. Build a shallow NN-classifier that takes as inputs attributes X0, Y0, Z0, , X4, Y4, Z4 AND the User information, and returns the Class as the output
3. Build a shallow NN classifier ONLY for Class 0. The classifier takes as inputs attributes X0, Y0, Z0, , X4, Y4, Z4, and returns the User ID as the output.

Classification accuracy on each of the above tasks

For each of the required task, I provided a python code with *readable* annotations. The weight of my networks are also provided. For the reproducibility of the result, the seed of the random number generator For the task previously mentioned, I got the following accuracy:

Task 1 After 160 epoch, I got an accuracy of 0.9757 for the training set (and a loss of 0.0732). I got an accuracy of 0.9770 for the validation set (and a loss of 0.0699). Finally, I got an accuracy of 0.9761630543062843 for the test set (and a loss of 0.07232725880181515).

Task 2 After 180 epoch, I got an accuracy of 0.9941 for the training set (and a loss of 0.0175). I got an accuracy of 0.9943 for the validation set (and a loss of 0.0190). Finally, I got an accuracy of 0.9937260035516052 for the test set (and a loss of 0.0196389602655521).

Task 2 After 180 epoch, I got an accuracy of 0.9658 for the training set (and a loss of 0.0898). I got an accuracy of 0.9899 for the validation set (and a loss of 0.0398). Finally, I got an accuracy of 0.9906138012015633 for the test set (and a loss of 0.03838270541556498).

Tools

I mainly used 3 tools:

- For training the networks, I used Keras with tensorflow for the Backend
- For handling the data and generating the training/validation/test set, I used sklearn.
- For the visualization and the selection of the data, I used pandas.

The networks

For the 3 different tasks, I used 2 hidden layers NN. For the task 1 and 2, the network architecture is similar. However for the third task, I added an extra dropout layer to prevent the model to overfit (the dataset is much smaller). The batch size has been reduced from 40 to 20 for this tasks also (the dataset is much smaller).