Deep Learning Architectures - Practical work

LSTM for human action recognition

Goal of the practical work

In this practical work, the aim is to employ LSTM models for recognizing human actions from skeleton data

Assignment

You have to send your work (Colab link of zip code) to maxime.devanne@uha.fr

Human skeleton data

For this practical work, we will use the MSR Action 3D dataset (https://sites.google.com/view/wanqingli/data-sets/msr-action3d). The datasets includes skeleton sequences extracted with a Kinect camera observing people performing different actions. Each skeleton contains a set of 20 3D joints corresponding to different body parts.
Data can be found here :

- Skeleton data (X.npy) : https://maxime-devanne.com/datasets/MSRAction3D/X.npy
- Action labels (Y.npy) : https://maxime-devanne.com/datasets/MSRAction3D/Y.npy
- Subject labels (Y_s.npy) : https://maxime-devanne.com/datasets/MSRAction3D/Y_s.npy

1 Loading and observing data

1- The first thing you have to do is to load the datasets. Try to understand how each file is organized

2- Then, you can use the provided functions to visualize some sequences

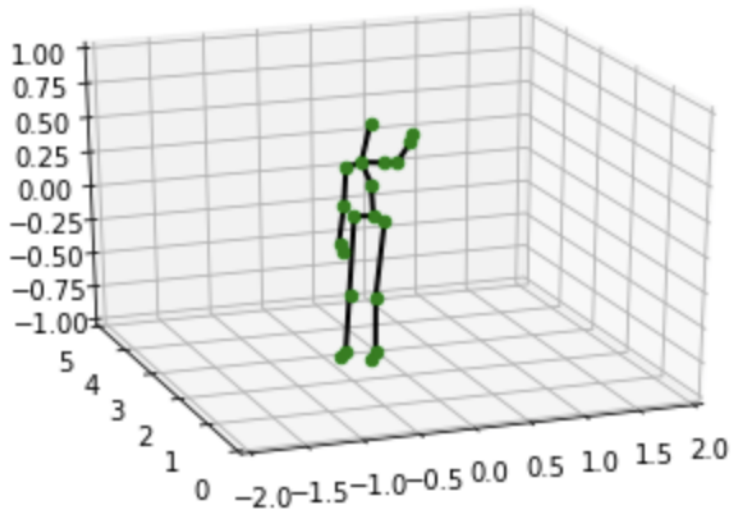

FIGURE 1 – Skeleton sequence visualization

3- Find the total number of classes, it will be useful when building the model

4- For normalization, you can use the provided function normalizing skeleton sequences between -1 and 1

2 Data preparation

5- Split your data into a train set, a validation set and a test set. For this dataset, this split is often done using subject labels information. As there are few samples per classes, randomly splitting the dataset could result in missing classes in some subsets. Then, consider the subject labels (Y_s.npy) for doing your split. First five subjects may be used for training, subjects 6 and 7 for validation and the remaining for test.

3 LSTM for human action recognition

6- Build a first classification model containing a single LSTM layer with 24 hidden units. Also Add a Dropout layer $rate = 0.5$.

7- Train your model on the training set for 2000 epochs with a batch size of 256

8- Evaluate your model on the test set

9- Plot the train and test curves (accuracy and loss)

10- Build a new model with at least two LSTM layers and do the experiments again. In particular, look at the parameter `return_sequence`, it may help you to see how to build a deeper LSTM.

4 Visualization

11- After doing the action classification with a model, compute and plot the confusion matrix in order to see which classes are confused (You could then visualize some skeleton sequences of corresponding confused classes to see if it makes sense). You should obtain a confusion matrix similar to the following :

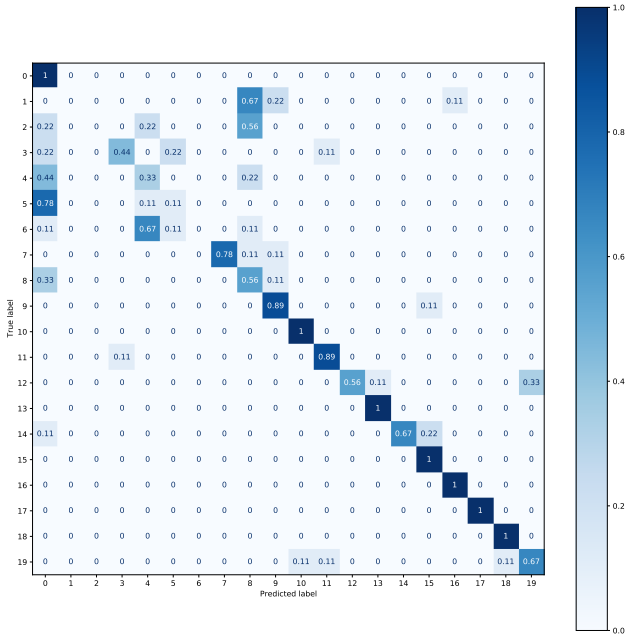

FIGURE 2 – Confusion Matrix

12- Another way to analyze results is to assess the extracted features by the LSTM model. First, retrieve the features extracted by the last LSTM layer from the training set. Then, apply a PCA on these features to keep the two principal components only. Visualize the 2D components in order to see if similar actions are grouped together :

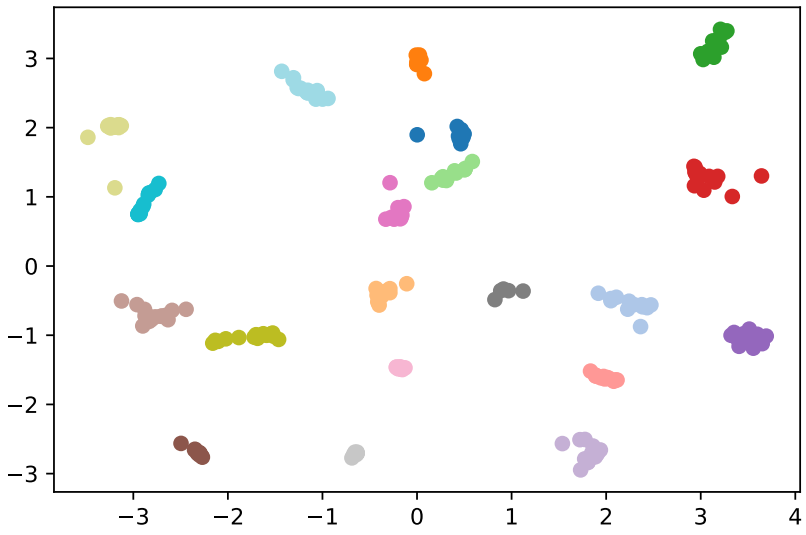

FIGURE 3 – PCA Analysis

5 Bonus

13- Transform your LSTM model into a Bi-LSTM model and run the experiments again. Bi-LSTM are considering sequences both forward and backward and combine their representation.