Deep Learning Project Report

On

Motion Capture Hand Postures Data Set

Submitted By: Asmita Dhage

Introduction:

While there are new and accessible technologies emerging to help those with hearing disabilities, there is still plenty of work to be done. For example, advancements in machine learning algorithms could help the deaf and hard-of-hearing even further by offering ways to better communicate using computer vision applications. The increasing demand for rendering smooth and plausible 3D motion is fueling the Development of motion capture systems.

In this project I have tried to classify hand postures based on co-ordinates of hand for different motions.

Problem Statement:

- The main task to be performed under this project is to classify the hand postures into 5 types
- Also, to use data augmentation and generator to get different invariants to train the model so as to increase the accuracy of model

Dataset Details:

- The dataset used for this project is taken from below URL : https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures
- Dataset consists record of 12 users performing 5 hand postures with markers attached to a left-handed glove
- A rigid pattern of markers on the back of the glove was used to establish a local coordinate system for the hand, and 11 other markers were attached to the thumb and fingers of the glove
- All markers were transformed to the local coordinate system of the record containing them. Second, each transformed marker with a norm greater than 200 millimeters was pruned
- Finally, these semi-preprocessed data, in which all markers are represented as coordinate is stored in .csv file
- Below are the attributes and their description used in this dataset
  1. Class - Integer.
     The class ID of the given record. Ranges from 1 to 5 with
     1=Fist (with thumb out), 2=Stop (hand flat), 3=Point1(point with pointer finger), 4=Point2(point with pointer and middle fingers), 5=Grab(fingers curled as if to grab).
  2. User - Integer.
  The ID of the user that contributed the record. No meaning other than as an identifier

3. Xi – Real

The x-coordinate of the i-th unlabeled marker position. 'i' ranges from 0 to 11

4. Yi – Real

The y-coordinate of the i-th unlabeled marker position. 'i' ranges from 0 to 11

5. Zi – Real

The z-coordinate of the i-th unlabeled marker position. 'i' ranges from 0 to 11

BACKGROUND RESEARCH:

3D motion being the immerging technology, many variations of motion gestures are available. However, all the datasets and existing work available on this topic is done with image data preprocessing.

Hand Gesture Recognition Database (https://www.kaggle.com/gti-upm/leapgestrecog)

This dataset tries to achieve same results, however, all the models associated with this dataset are done with image processing.

Hence I wanted to work on the mentioned dataset so as to achieve same result with numeric data

Approach:

Task1: Importing and cleaning of dataset

i. The dataset is imported as a .csv file
ii. Missing values were represented by '?', hence as a first step of preprocessing I have replaced all '?' with numeric data type Nan
iii. After that I have replaced all Nan values with mean of corresponding columns
iv. Since the data set was sorted in ascending order, I have used sample function to shuffle rows in dataset

Task2: Identifying training and testing data from dataset

i. Before splitting dataset into training and testing data, I have removed unnecessary columns.
   Here the column 'User' was just representing the ID of the user that contributed the record. It had No meaning other than as an identifier. Hence I removed this column
ii. After that I have divided dataset into features (Xi, Yi and Zi) representing the coordinates and label (Class)

iii. I have used **“Sklearn” – “train_test_split”** method on X and Y in order to get training data and testing data

Task3: Choosing hyper parameters and building model

i. In this task, to choose correct hyper parameters, I have tried various combinations of hyper parameters.

ii. For trying different variants in network, I have used different shape for each layer, also I tried various activation functions and epoch value

Task4: Evaluating model and checking prediction accuracy

i. After training model on training dataset, for evaluating it I have used test dataset Model performed equivalently well on test data as well

ii. As a last step of checking accuracy of model on original data, I have used **model.predict()** function. In this step also predicted label and expected label are matching hence giving 99% accuracy for original data

Task5: Using Data Augmentation and Generator

i. To avoid overfitting of the model, and also to train the model on random and diverse data, data augmentation techniques are used

ii. For this task, first I have **divided dataset into small chunks**. This function behaves as a Generator

iii. On each batch(chunk) of data, I have used below two data augmentation techniques:
   1. Randomly replacing some values with Zero
   2. Adding noise to random values in each batch

iv. Now, I have built a model which takes each augmented chunk as an input and predict the output. For this step, I have used **model.fit_Generator()** method

Task6: Result Analysis

i. With original data, when the neuron in each layers and activation function was not chosen properly, model was not performing up to the mark.

ii. When appropriate hyper parameters were chosen and model was trained on 70% data, it worked as expected on 30% test data, giving accuracy as high as 99%

iii. With augmented data, accuracy is slightly low compared to original data but it has not shown huge difference in accuracy when tested on original data and augmented data

Conclusion:

1. Choosing correct hyper parameters is extremely important to build any classification/prediction model
2. To avoid overfitting, it is necessary to choose random data from dataset as a training and testing data
3. Creating variants of data and using augmentation techniques on the variants, actually improves the performance of model, as in this case model evaluates the diverse data

References:

[1] UCI Machine Learning Repository - Motion Capture Hand Postures Data Set, https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures

[2] Keras Documentation, https://keras.io/

[3] Datasets for Motion-Capture-Based Hand Gesture Recognition, http://www2.latech.edu/~jkanno/datadescription-1.pdf

[4] Deep Learning – Keras, http://www.keithdillon.com/index.php/deep-learning-fall-2019/

[5] GeeksforGeeks - Python | Data Augmentation, https://www.geeksforgeeks.org/python-data-augmentation/