Driver Trajectory Classification with Deep Neural Network

10.03.2023

By

Parmar Tarun

Introduction & Proposal

Location based data of objects moving around on the globe plays an important role in this mobile world. A machine learning model trained on such gps data can be very helpful in recognising patterns of those objects when they move. Along with gps data, if featured with timestamps, a robust classification model can be developed which can successfully classify various patterns. This project aims at such a classification of drivers based on their recorded movements over the globe. The movements are defined by their geo coordinates at a particular instance of time, along with a status whether the driver's cab was occupied with a customer or not.

Methodology

1. Network Structure

The network built for the project is a fully connected network. To build this network I have used keras's sequential model. The first layer, which is a dense layer, consists of 64 neurons with input size of 2 followed by an activation layer of hyperbolic tangent. Next the network is appended with another dense layer of 128 neurons followed by another dense layer of 256 neurons. The next part of the network is then designed to reduce the number of neurons in each layer. The output layer consists of 5 neurons corresponding to the 5 unique driver plates. Softmax function is used for the output layer so that we have an output in the form of a probability. These probabilities will be used to decide the classification of the trajectory.

The model is compiled using the categorical cross entropy as loss function, adam as optimizer and accuracy as metric for evaluating the training and testing results.

2. Training & Validation process

The Dataset is loaded from local files. The dataset includes a total of 179 csv files, each corresponding to data collected for a day of all the five drivers. The files have 5 columns, longitude, latitude, time, status and plate.

The preprocessing of data included grouping of data by plate, followed by sorting each group based on time. This was done so that we can call the whole group as a trajectory for that driver. Next, this trajectory was then converted to a single row of 2 features, occupied distance and unoccupied distance. This is done by calculating the distance between each longitude and latitude values and summing it up respectively whether the status is 1 or 0. These two features indicate how much distance the particular driver traveled in a day while the cab was occupied and unoccupied. Such information can be helpful to distinguish different drivers. Furthermore, the plate column in the dataset is of int type which is not suitable for a classification task. Hence, the plate values were converted to categorical type where the numerical value is modified to a vector of size 5, each element in the vector being 0 except for the one element to be 1, corresponding to that digit. This is done using the to_categorical function offered by keras. The training process was done with 100 epochs and batch size 15.

Evaluation & Results

I. Training & Validation Results

After completing the training process, the model resulted in the accuracy of 42.4% in training and 33.7% in testing.

II. Hyperparameter

The following hyperparameters were used for training process:

- ***Number of neurons***: Different number of neurons in different layers with increasing and decreasing fashion.
- ***Dropout***: A dropout of 0.45 was used at multiple stages of the network.
- ***Activation function***: Different activation functions at different stages of the network were used like relu, tanh and softmax.
- ***Number of epochs***: Number of epochs for the training process were increased which resulted in better accuracies.

Conclusion

The trajectory data for any moving object on the earth's surface can prove to be a good dataset for finding patterns. With 179 days of gps data for 5 cab drivers is not a good amount of data for creating a deep neural network, however the larger dataset can be very helpful in getting better accuracy.