ML T-rex game

02.03.2021

Introduction

This project aims at how T-rex game plays by itself. Supervised learning fundamentals were applied. Supervised learning is a way machines learn. The data has to be given before building a model. Therefore, datas which are from the game screen were collected by python code. Further, the model was trained via TensorFlow and Scikit Learn. Then, the model runed on the screen which we got data and the model played the game by itself.

Details

1. **Collecting Data:** Data collection, the first step, is the most important part. We need to collect good and utility data.

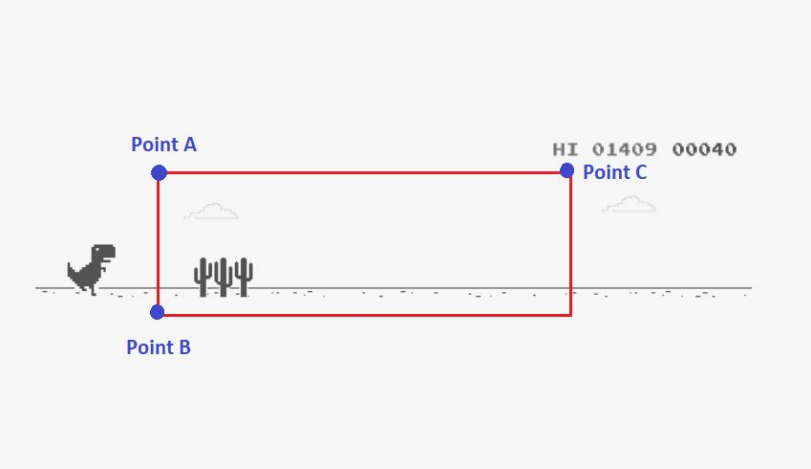

Figure 1.1

In the figure 1.1 there is a red rectangle. The line where between point A and point B is where the T-rex jumps. This red rectangle is data. Collecting algorithm collects data when the up or down key is pressed every time.

Point A gives us the location which is the start point. The distance between point A and point C is the width, and the distance between point A and point B is the height. This information helps us where we want to record. Because we don't want to record the whole screen. These datas are stored in a directory. In this project this directory is called "img".

2. **Training the model:** To train the model, TensorFlow and Scikit Learn were used. In the previous part, we collected data into a directory. Then, these images are normalized.

   Keras library helps us to train the model. Convolutional Neural Networks (CNN) were applied to every single image. CNN is used to detect the features. Applying CNN leads to loss edge pixels, therefore, MaxPooling method should apply after CNN. MaxPooling method adds pixels which are lost. Later, the Flatten method is applied. Flatten method's task is the conversion of matrices from the Convolutional and Pooling layers into a one-dimensional array. Then, Hidden layer gets this one-dimensional array as an input. In every layer, activation functions are defined. For CNN, ReLU is the most used function, for output, sigmoid or softmax functions are used.

   After all, the model is compiled, in this part parameters such as loss function, optimizer and batch size are defined. Now the model is ready to be trained.

3. **Game time:** Our model is learned when T-rex will jump. Time to apply this.

   In the last part, we achieve the game screen by using the mss library. Algorithms get images which are unseen in real-time, and the size of these images must be the same with images in the collection part. The model gets these images and predicts what they are. Model gives a result, and this result determines what the T-rex should do.

Libraries

- TensorFlow / Keras
- Scikit Learn
- Keyboard
- uuid
- time
- PIL
- mss
- glob
- os
- numpy

Emrehan KARAOGLU