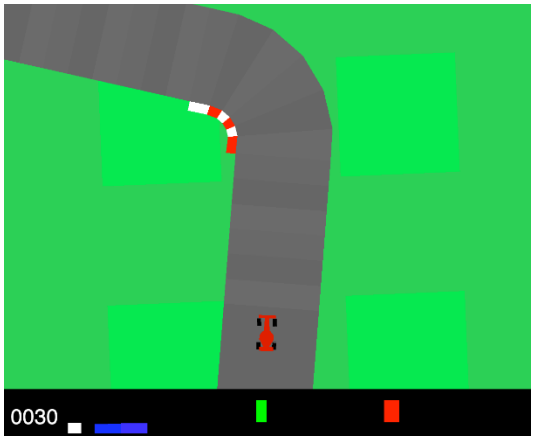
OpenAI Gym CarRacingV0 using CNN

Introduction

In this project, I used deep learning to solve the CarRacingV0 environment of the OpenAI Gym library. The environment simulates a car driving on a racetrack. The controllable attributes are: turning, acceleration, and brakes. This project will focus solely on predicting the turning attribute while playing the game.



Data preparation

The data used to solve this problem was manually gathered by me. I collected images of the game while playing it myself, grouping and labelling them according to which button direction the car was going: left, right, or straight.

left

right

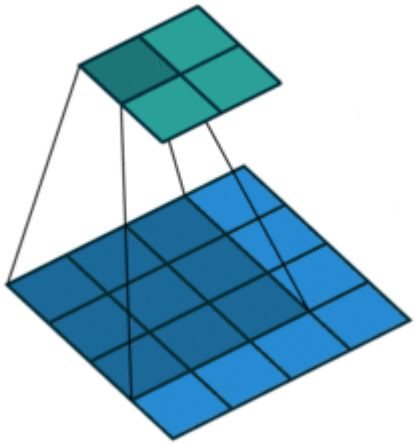
straight

Convolutional Neural Network

The Neural Network used for this problem is a Convolutional Neural Network (CNN). A CNN uses Convolution as a method of feature extraction. This is often used on images as using each pixel as a feature drastically decreases the performance of a traditional Neural Network.

Convolution

Convolution has a window (or filter) which contains its own values. This window gets placed over the pixel image starting at the top left. The overlapping values are calculated into a single filtered value. This window shifts over the entire image to create the convoluted image.

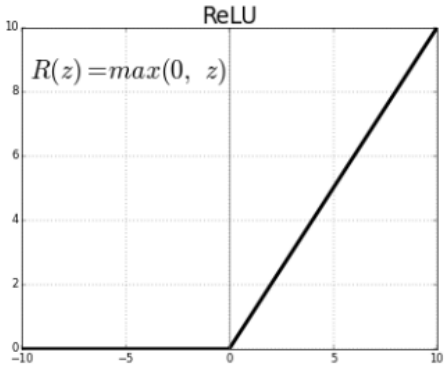


Batch normalization

Batch normalization normalizes the input values to be mostly in the same range. The normalization used in this project changes the value to have an average around 0 and a standard deviation around 1.

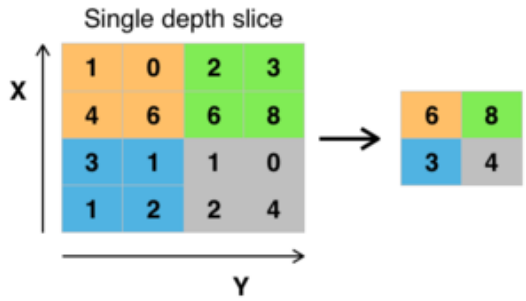
Activation

The activation layer uses the inputs, weights, and an activation function to determine the value of each input. The activation function used in this project is the “ReLu” function (see image below).



Pooling

Pooling is used to reduce the dimensions of the input. It segments an image in groups of 4 or more pixels and creates a single value for each segment which represents every pixel in the segment. Max pooling is used here which means the largest value of the 4 is kept. This and average pooling, which takes the average of the values in a segment, are the most common ones to use.



Results

The model scored an accuracy of 89% (see image below). The simulation when trained was of average quality, the car was able to take turns, but was unstable on straight roads as it kept driving from left to right.

```
              precision    recall  f1-score   support

           0       0.86      0.95      0.90        40
           1       0.92      0.97      0.95        80
           2       0.87      0.68      0.76        40

    accuracy                           0.89       160
   macro avg       0.88      0.87      0.87       160
weighted avg       0.89      0.89      0.89       160

[[38  0  2]
 [ 0 78  2]
 [ 6  7 27]]
89.375
```