[Project Template] Behavior Cloning 22-Sep-2017

PROJECT SPECIFICATION

Use Deep Learning to Clone Driving Behavior

Link to this Document: https://www.evernote.com/l/AC6zqxfd80JEUrB1GTe1Z7JM0KUtaOEOe6o

Required Files

CRITERIA	MEETS SPECIFICATIONS
Are all required files submitted?	The submission includes a model.py file, drive.py, model.h5 a writeup report The video.mp4 file is attached below. If I add it in Github the History will become very heavy and additional pull and upload will become hard. Generic File This Video is created using video.py. Click here to see the video: https://www.evernote.com/shard/s46/sh/b3ab17dd-f342-4452-b075-1937b567b24c/d0a52d68e10e7baa/res/d39f6ebc-b91d-4d57-8945-cfd390cb66f0/run-16.mp4 The Video created using Quicktime Player is attached in this youtube link. https://www.youtube.com/watch?v=dUGMRMySyso&feature=youtu.be Conda env at: environment.yml Other important files: **augmentation.py** **preprocessing.py** Floydhub data: https://www.floydhub.com/viewer/data/JfStbCpiFpmGNq3ZUURkPS/VZGnMm8qnCHdDtPsPzFp6X/

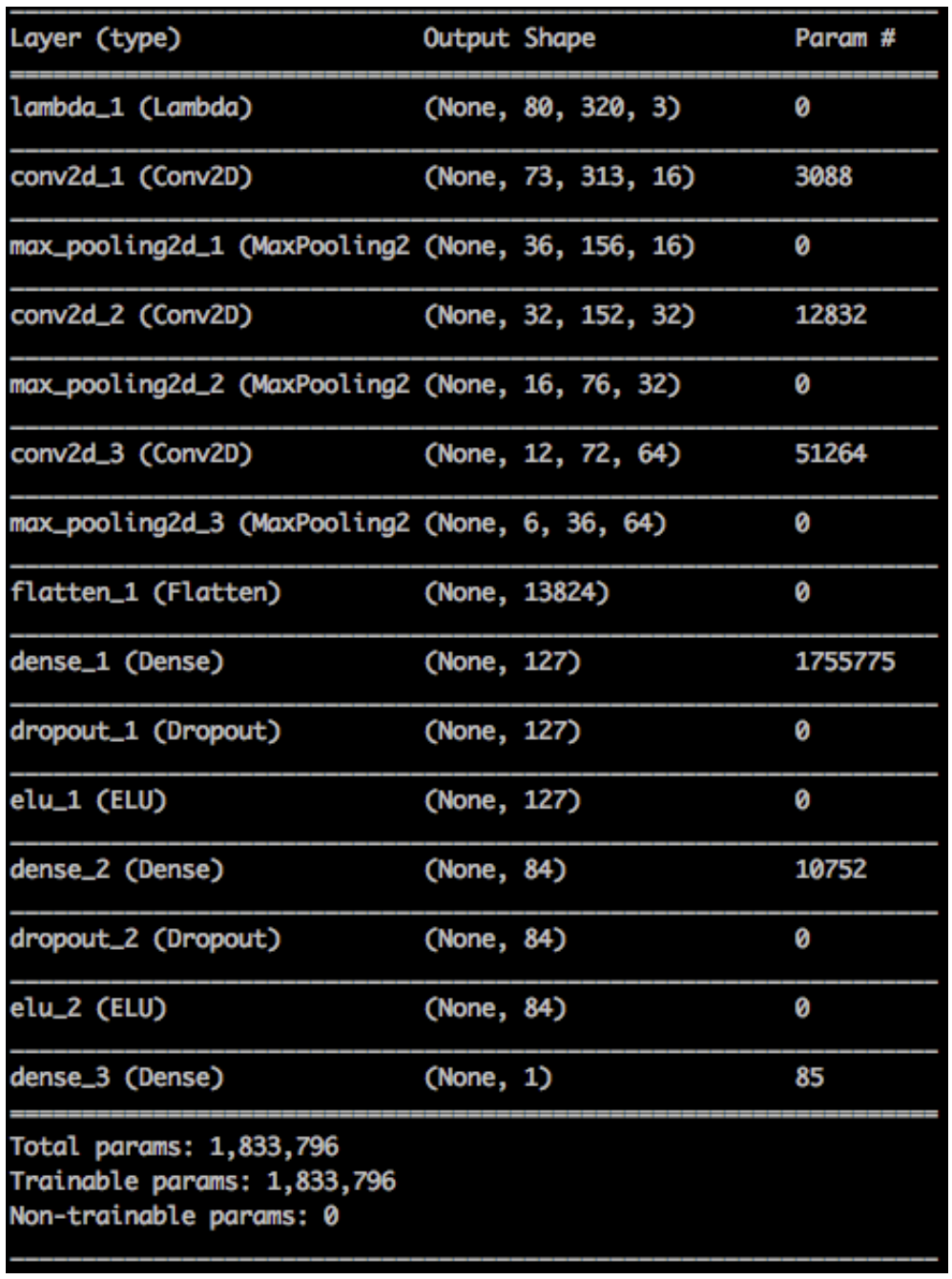
Quality of Code

CRITERIA	MEETS SPECIFICATIONS
Is the code functional?	The model provided can be used to successfully operate the simulation. Please note that the model was trained on Keras 2.0.6 as

	I use Floydhub for running my learning and it provides Keras version 2.0.6 I have added my condo env file in my repo as *environment.yml* So if version is not inline, it will through t his error: *You are using Keras version b'2.0.8' , but the model was built using b'2.0.6'*
Is the code usable and readable?	The `model.py` code is clearly organized and comments are included where needed. Training data was augmented: - the existing data was duplicated n times - Image flipping was done for steering angles > 0.15 or < -0.15. and the steering angles where inverted. This increased the number of turn drive data.

Model Architecture and Training Strategy

CRITERIA	MEETS SPECIFICATIONS
Has an appropriate model architecture been employed for the task?	The neural network uses convolution layers with appropriate filter sizes. Layers exist to introduce nonlinearity into the model. The data is normalized in the model.

	 This is the model summary as printed out by Keras. - 3 convolutional layers where added, to improve the colour space transformation of the image. - the output was passed through Relu activation function, to introduce non-linearity in the model. - each of them where followed by max_pooling - The result from the convolutional network was flattened and passed on to 3 fully connected layers. - initial 2 rounds of fully connected layers where followed by dropout of 50 %, and ELU (Exponential Linear Unit) Activation for reducing the overfitting. We introduce non-linearity to the network, as linear networks will remove the purpose of having multiple layers. And ELU's which are exponential in nature tries to keep the mean activation close to zero. Giving us better results. ref. - and final fully connected layer returns us the steering angle. train loss: 0.0187 - val_loss: 0.0172 was achieved which signifies no overfitting was observed. A classic example to learn regression.
Has an attempt been made to reduce overfitting of	Train/validation/test splits have been used, and the model uses dropout layers or other methods to reduce overfitting.

the model?	We did a train validation split of 80/20. Dropouts where used between Dense layers to reduce overfitting.
Have the model parameters been tuned appropriately?	Learning rate of 0.0001 was used on Adam Optimiser. EPOCHS = 4 AdamOptimizer a variant of Stochastic Gradient Descent was used as optimiser.
Is the training data chosen appropriately?	Udacity drive data was used + self drive data was used + zig zag drive data was used. * these data were augmented by duplication and image flipping for steering angles > 0.15 both left and right. Hard part was collecting self drive data as my own driving on emulator was not pretty great.

Architecture and Training Documentation

CRITERIA	MEETS SPECIFICATIONS
Is the solution design documented?	The README thoroughly discusses the approach taken for deriving and designing a model architecture fit for solving the given problem. I have included all those in this document itself under **Model Architecture and Training Strategy**
Is the model architecture documented?	The README provides sufficient details of the characteristics and qualities of the architecture, such as the type of model used, the number of layers, the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged. I have included all those in this document itself under **Model Architecture and Training Strategy**
Is the creation of the training dataset and training process documented?	The README describes how the model was trained and what the characteristics of the dataset are. Information such as how the dataset was generated and examples of images from the dataset must be included. I have included all those in this document itself under **Model Architecture and Training Strategy**

Simulation

CRITERIA	MEETS SPECIFICATIONS
Is the car able to navigate correctly on test data?	No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe (if humans were in the vehicle).

Suggestions to Make Your Project Stand Out!

Track Two

The simulator contains two tracks. To meet specifications, the car must successfully drive around track one. Track two is more difficult. See if you can get the car to stay on the road for track two as well.