**Behavioral Cloning**

**Writeup Template**

**You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.**

**Behavioral Cloning Project**

The goals / steps of this project are the following:

* Use the simulator to collect data of good driving behavior
* Build, a convolution neural network in Keras that predicts steering angles from images
* Train and validate the model with a training and validation set
* Test that the model successfully drives around track one without leaving the road
* Summarize the results with a written report

**Rubric Points**

**Here I will consider the**[**rubric points**](https://review.udacity.com/#!/rubrics/432/view)**individually and describe how I addressed each point in my implementation.**

**Files Submitted & Code Quality**

**1. Submission includes all required files and can be used to run the simulator in autonomous mode**

My project includes the following files:

* model.py containing the script to create and train the model
* drive.py for driving the car in autonomous mode
* model.h5 containing a trained convolution neural network
* writeup\_report.md or writeup\_report.pdf summarizing the results

**2. Submission includes functional code**

Using the Udacity provided simulator and my drive.py file, the car can be driven autonomously around the track by executing

python drive.py model.h5

**3. Submission code is usable and readable**

The model.py file contains the code for training and saving the convolution neural network. The file shows the pipeline I used for training and validating the model, and it contains comments to explain how the code works.

**Model Architecture and Training Strategy**

**1. An appropriate model architecture has been employed**

# 私は、モデル作成のために基本となる Training The Network(4章)、LeNet(9章)、 NVIDIA Architecture(14章)に基づく訓練を行い、各動作を確認しました。結果、最もうまく車両を動かせた NVIDIA Architectureを採用しました。

# (補足：訓練方法の切り替えはModel.pyの115行目に示すLessonの値で選択できます。)

# このモデルは、非線形性を導入するためのRELUレイヤーを含み、データはKerasラムダレイヤーを使用したモデルの正規化を行っています。またクロッピング処理を行い教師データから不要な背景を削除しました。

**2. Model parameter tuning**

# モデル作成について、最適化のアルゴリズムにはAdamを使用、損失評価にはMSE（平均二乗誤差）を使用しました。

**3. Appropriate training data**

トレーニングデータには、センターカメラ、左カメラ、右カメラの映像を組み合わせました。.

トレーニングデータ作成方法の詳細については、次のセクションを参照してください。

**Model Architecture and Training Strategy**

**1. Solution Design Approach**

まず、Project Resources（第１章）より、サンプルトレーニングデータを取得し、車両走行の基本とする教師データとして訓練に与えました（model.pyの19行目ubu\_file01）。

次に実際に道路中央を逸脱しないよう正確にコースを周回して、理想的な教師データを２つ分追加しました（model.pyの21行目win\_file01,22行目win\_file02）。

さらにコースを逆周回した教師データを２つ分追加しました（model.pyの23行目win\_file03,24行目win\_file04）。

最後に路面の色が変化する「橋の上」での走行データを追加しました（model.pyの20行目ubu\_file02）

センターカメラ、左カメラ、右カメラの映像を教師データに採用したため、データの数が多くなりました。そこで、私は操舵の小さい時に撮影した教師データを間引きました。これは直線よりもコーナを走行する際の車線追従性能を向上させる目的を兼ねています。

また、変化に乏しい同じような教師データではユニーク特徴がつかみにくく、訓練結果が過適合となる恐れがあると判断して、教師データを200個ずつスキップさせながら訓練する方法を採用しました。

さらに、各教師データと操舵データを反転させた教師データを追加し、データに含まれる各特徴を多様化させました。

結果、私のmodel.h5を用いた車両は、自律的にコースを周回することができました。

**2. Final Model Architecture**

前述の1. An Appropriate Model Architecture Has Been Employedに示すよう私のモデルはNVIDIA Architecture(14章)に基づき作成しました。具体的なコードはModel.pyの198行目〜209行目に示します。

※アーキテクチャを視覚化することは、プロジェクトのルーブリックに従ってオプションということなので詳細は省きます。

**3. Creation of the Training Set & Training Process**

私の検討した内容は前述の「An appropriate model architecture has been employed回答」、「 Model parameter tuning回答」、「 Solution Design Approach回答」に示したとおりです。詳細は省きます。

### ★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆★☆

### Requires Changes

#### 3 SPECIFICATIONS REQUIRE CHANGES

Dear Udacian,

I really enjoyed reading every line of your code and README! It really demonstrates your knowledge in deep learning and how hard you worked to achieve your final results.

Unfortunately, some rubric points were overlooked. I'm sure that with your experience your next submission will meet all requirements.

Keep up the amazing work!!

Please submit the README file in the .md file format or convert it to English before saving it as a pdf file. Translating pdf files is really hard.

Thank you.

### Required Files

### 

### Quality of Code

### 

### Model Architecture and Training Strategy

### 

### 

### 私のモデルに対してドロップアウトレベルを1.0（無）、0.9、0.7、0.5と段階的に適用した結果、ドロップアウト1.0（無）と0.5との損失差がそれほど変わらないことを確認しました。よってドロップアウト無を選択して、より多くのシーンを訓練する戦略を採用しました。 I gradually applied the dropout level to my model as 1.0 (nothing), 0.9, 0.7, 0.5 and confirmed that the loss difference between dropout 1.0 (no) and 0.5 is not so different. So we chose a strategy to train more scenes by choosing no dropout.

### \\casket\users\柴田健吾\社内・組織・事務\uda\CarND-Behavioral-Cloning-P3-master\提出２\無題のフォルダー 2\無題のフォルダー\figure_1-1.png\\casket\users\柴田健吾\社内・組織・事務\uda\CarND-Behavioral-Cloning-P3-master\提出２\無題のフォルダー 2\無題のフォルダー\figure_1.png\\casket\users\柴田健吾\社内・組織・事務\uda\CarND-Behavioral-Cloning-P3-master\提出２\無題のフォルダー 2\無題のフォルダー\figure_1-2.png\\casket\users\柴田健吾\社内・組織・事務\uda\CarND-Behavioral-Cloning-P3-master\提出２\無題のフォルダー 2\無題のフォルダー\figure_1-3.png

### 

### 

### Architecture and Training Documentation

### 

### 

### 私のモデルはNVIDIA Architecture(14章)に基づきレイヤーの数や各レイヤーのサイズ、以下のように採用しました。

### According to NVIDIA Architecture (Chapter 14), my model has adopted the number of layers and the size of each layer as follows.

### \\casket\users\柴田健吾\無題のフォルダー 2\figure_img.png

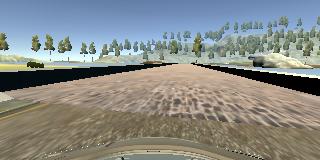
入力層の直後にクロッピング領域を設定し、不要な背景等の不要な情報を除外しました。

I set the cropping layer immediately after the input layer and excluded unnecessary information such as unnecessary background.

### 

Project Resources（第１章）より、サンプルトレーニングデータを取得し、車両走行の基本とする教師データとして訓練に与えました。次に実際に道路中央を逸脱しないよう正確にコースを周回して、理想的な教師データを２つ分追加しました。さらにコースを逆周回した教師データを２つ分追加しました。最後に路面の色が変化する「橋の上」での走行データを追加しました。

I got sample training data from Project Resources (Chapter 1) and gave it to the training as the basic teaching data of the vehicle running. Next, we went around the course exactly so as not to deviate from the center of the road actually, and added two ideal teacher data. I added two additional teacher data that went back around the course.   
Lastly the running data on the "bridge top" where the color of the road changes will be added.

ADD-TRAINING-SAMPLE(1)

running on the bridge

ADD-TRAINING-SAMPLE(2)

running on a sharp curve

センターカメラに加えて、コースを外れることなくレーンを維持できるように、トレーニング用に左右のカメラの画像を使用しました。

In addition to the center camera, I used the image of the left and right camera for training so that I can maintain the lane without running off course.

LEFT-CAMERA-IMAGE

Steering-Force “Fl” is set by following calculation.

※Fc≧0 : Fl=Fc+0.4

Fc＜0 : Fl=Fc\*0.8

CENTER-CAMERA-IMAGE

I use Steering-Force ”Fc”

※System recoreded value of ’Fc’

RIGHT-CAMERA-IMAGE

Steering-Force “Fr” is set by following calculation.

※Fc≦0 : Fr=Fc－0.4

Fc＞0 : Fr=Fc\*0.8

### 