Workshop 8– Advanced Deep Learning

Advanced Analytics and Applications [AAA]

Information Systems for Sustainable Society (is3) | WiSo Faculty | Univ.-Prof. Dr. Wolfgang Ketter | 6/7/23

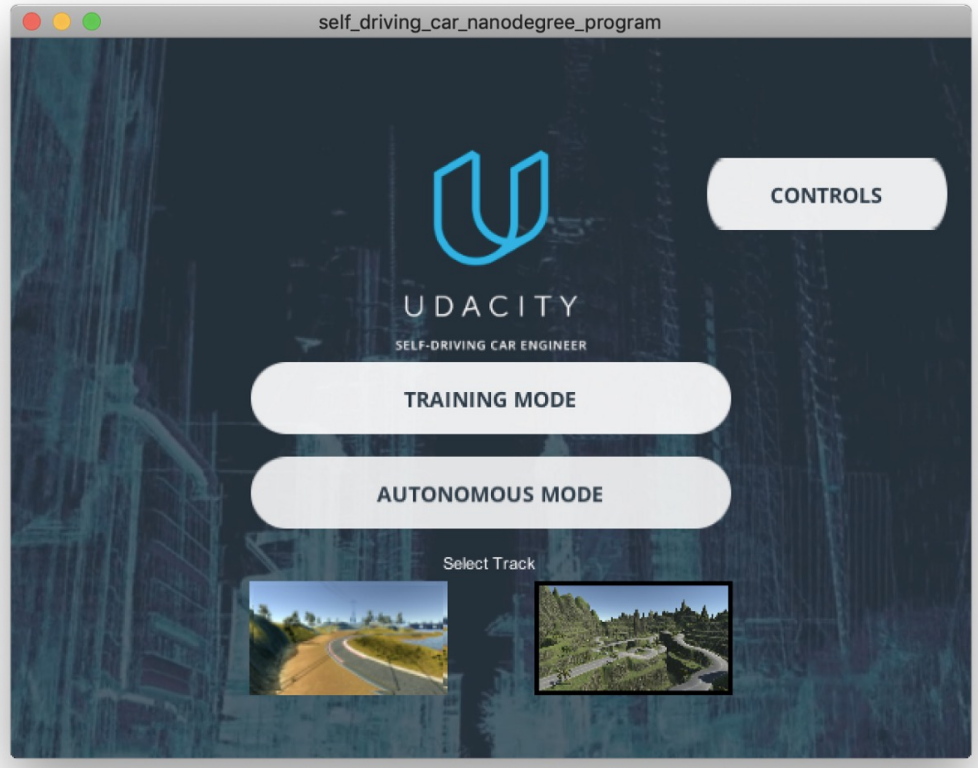
Calculation

Programming

Today‘s objective: We want to train a deep learning model to drive a car in a simulated environment

https://www.youtube.com/watch?v=xtbwPz7Api8

Step 1: Udacity Simulation Environment



Download Simulator

https://github.com/udacity/self-driving-car-sim

Step 2: Install TensorFlow, Keras, and relevant libraries

Create a dedicated virtual environment, if necessary

```
pip install flask=0.11
pip install python-engineio==2.0.0
pip install python-socketio==1.8.0

pip install eventlet
pip install numpy
....
```

Step 3: Create a CoLab Account

To speed up training, we use CoLab

https://colab.research.google.com

Step 4: Generating Data

Start Simulator

Run Training Mode

Start Recording (Folder /data)

Universität zu Köln

Step 5: Analyze Generated Data

Left, Center and Right Camera Images Respectively from the Simulator Vehicle on Track 1

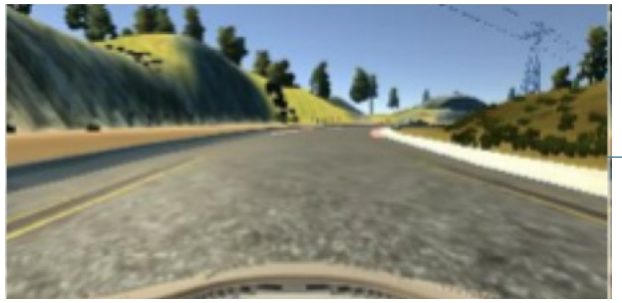
center	left	right	steering angle
/Users/demir	/Users/demi	/Users/demi	0
/Users/demir	/Users/demi	/Users/demi	0
/Users/demir	/Users/demi	/Users/demi	-0.3471122
/Users/demir	/Users/demi	/Users/demi	-0.7194571

Normalized between -1 and 1

driving_log.csv

Assumption: Velocity is not important for us.

Step 6: Designing a DL Architecture

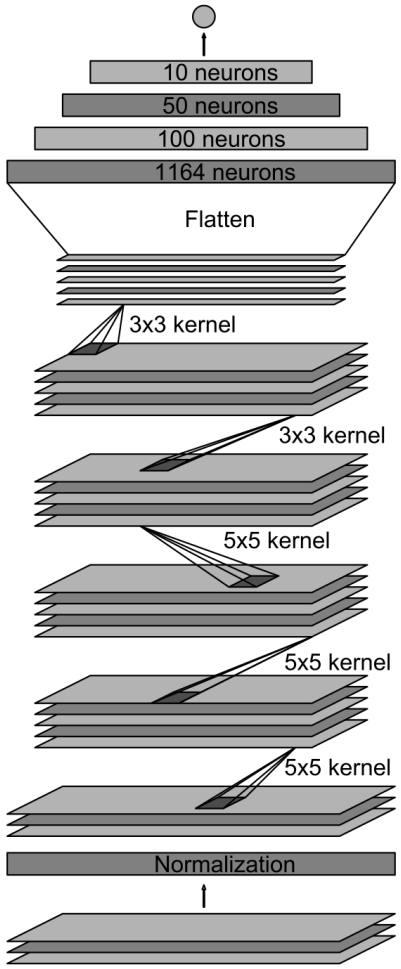


Input: 160px x 320px x 3 color channels

Deep Learning Model

Value between -1 and 1 (steering angle)

Step 6b: Potential Architecture



Activation Function: ELU (Exponential Linear Unit)
Stride: Typically (2,2)
Optimizer: Adam
Epochs: 5

Save File as: model.h5

Based on:
https://arxiv.org/pdf/1604.07316v1.pdf

Step 7: Write A Generator to Read the Data

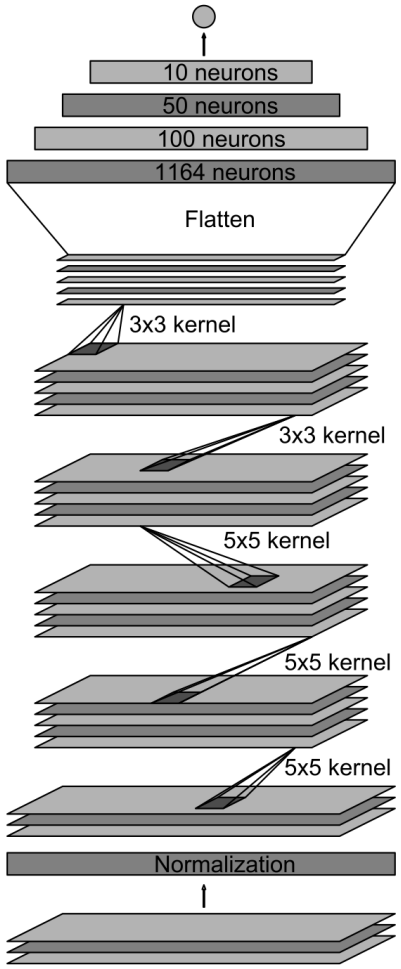
```
samples = []
with open('data/driving_log.csv') as csvfile:
    reader = csv.reader(csvfile)
    next(reader, None)
    for line in reader:
        samples.append(line)
```

```
def generator(samples, batch_size=32):
    num_samples = len(samples)
    while 1: # Loop forever so the generator never terminates
        shuffle(samples)
        for offset in range(0, num_samples, batch_size):
            batch_samples = samples[offset:offset+batch_size]

            images = []
            angles = []
            for batch_sample in batch_samples:
                for i in range(3): # center, left and rights images
                    name = 'data/IMG/' + batch_sample[i].split('/')[-1]
                    current_image = cv2.cvtColor(cv2.imread(name), cv2.COLOR_BGR2RGB)
                    images.append(current_image)


                    center_angle = float(batch_sample[3])
                    if i == 0:
                        angles.append(center_angle)
                    elif i == 1:
                        angles.append(center_angle + 0.4) # Perspective Transformation for Left Image
                    elif i == 2:
                        angles.append(center_angle - 0.4) # Perspective Transformation for Right Image

            X_train = np.array(images)
            y_train = np.array(angles)
            yield tuple(sklearn.utils.shuffle(X_train, y_train))
```

1. Read the driving_log to manage the samples

2. Write a generator function (stateful function using yield) to read only pictures that are part of the batch to avoid memory problems

Step 8: Implement the Deep Learning Model (own or Nvidia)

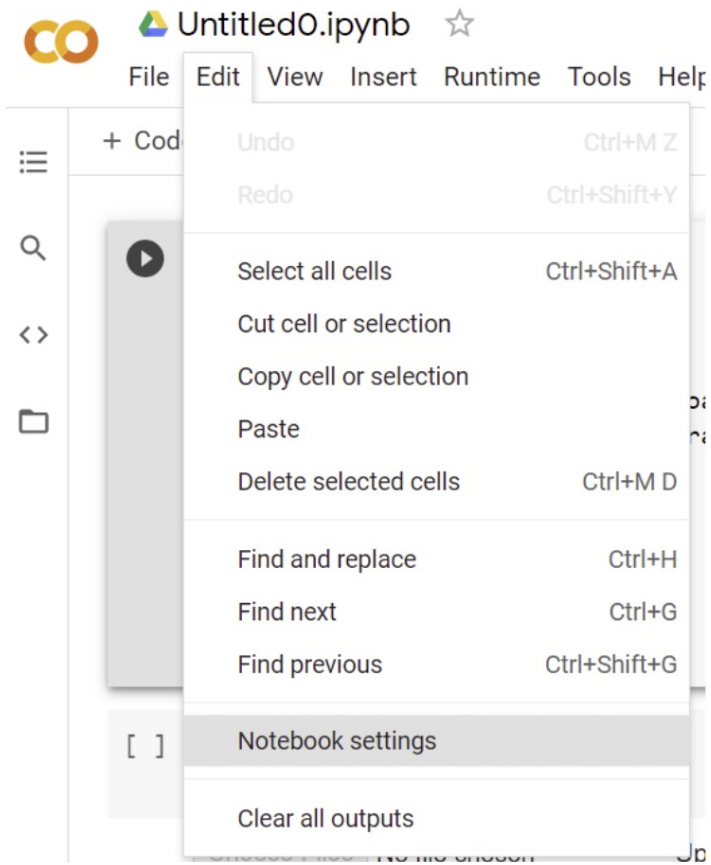


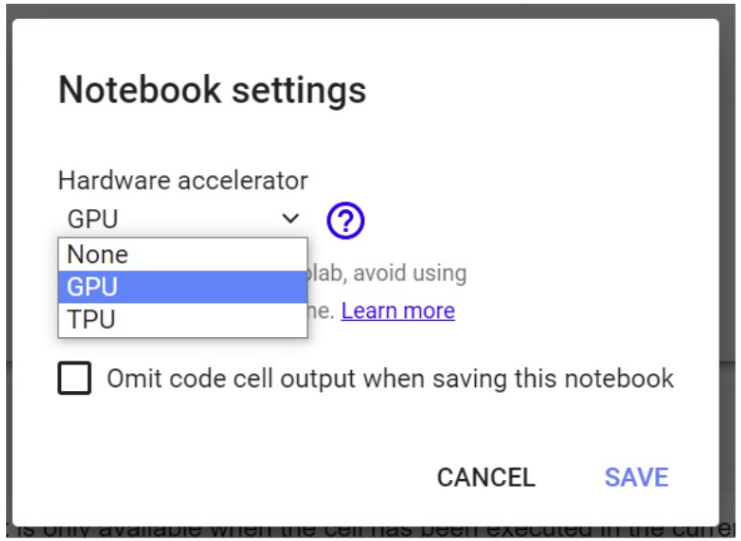
Activation Function: ELU (Exponential Linear Unit)
Stride: Typically (2,2)
Optimizer: Adam
Epochs: 5

Save File as: model.h5

Based on:
https://arxiv.org/pdf/1604.07316v1.pdf

Step 9: Use GPU





Step 10: Train Model using fit_generator (instead of fit) function

```
model.fit_generator(…)
```

Step 11: Save results to model.5h and download it

```
model.save('model.h5')
```

Step 12: Now, we come to the fun part. Start the autonomous mode.

Step 13: Open Terminal and CD to drive.py, and copy model.h5 to the same folder.

Run: python drive.py model.h5

Contact

For general questions and enquiries on **research**, **teaching**, **job openings** and new **projects** refer to our website at www.is3.uni-koeln.de

For specific enquiries regarding this course contact us by sending an email to the **IS3 teaching** address at is3-teaching@wiso.uni-koeln.de