**End-to-end driving from vision with Deep-Learning: from Imitation-Learning to Reinforcement-Learning**

Pr. Fabien MOUTARDE  
Center for Robotics  
MINES ParisTech  
PSL Université  
Fabien.Moutarde@mines-paristech.fr  
http://people.mines-paristech.fr/fabien.moutarde

---

### Idea of end-to-end driving

| Sensors (cameras, radar, LIDAR, etc...) | Perception | Trajectory Planning | Control | Driving wheel, Acceleration or braking |

**Current architecture for automated driving**

vs. HUMAN driving = turn/accelerate-brake by just looking in front  
≈ “intelligent” visual servoing
Principle of end-to-end driving

Current architecture for automated driving

Sensors (cameras, radar, LIDAR, etc...) → Perception → Trajectory Planning → Control → Driving wheel, Acceleration or braking

vs. HUMAN driving: turn/brake *by just looking in front!*  
≈ “intelligent” *visual servoing*

Outline

• Imitation Learning
• Reinforcement Learning
Imitation Learning: “copying” human driver

Nvidia approach


Imitation approach proposed by Valeo

“Copying” human driver

End-to-end driving from vision with DL, Pr. Fabien MOUTARDE, Center for Robotics, MINES ParisTech, PSL, 14/1/2020
Real data for training and testing

Training+testing dataset = 10000 km and 200 hours of human driving in openroad (highways, urban streets, country roads, etc...) under various weather conditions

TrainSet = 10 million images, TestSet = 3 million images.

**TABLE IV: Description of test scenarios**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Urban</th>
<th>Highways</th>
<th>Sharp turns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image count</td>
<td>100000</td>
<td>70000</td>
<td>15000</td>
</tr>
<tr>
<td>Duration (min)</td>
<td>56</td>
<td>39</td>
<td>8</td>
</tr>
</tbody>
</table>

“Simulator” with real images

“ConvNet in-the-loop” simulator with real images

[Work by my Valeo CIFRE PhD student Marin Toromanoff]
End-to-end driving: closed loop evaluation

Fig. 5: Screenshot of the simulator, blue trajectory is human, red is the car driven by the network. At current time, the network is translated of 7cm on the right. Here the network failed to take the turn, the red trajectory goes off-road, and a recovery is done.

TABLE V: Autonomy (%) and mean absolute distance (MAD, in cm) according to data distribution and validation scenario, the baseline is just going straight.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Urban</th>
<th>Highways</th>
<th>Sharp turns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>Aut. MAD (%)</td>
<td>Aut. MAD (cm)</td>
<td>Aut. MAD (cm)</td>
</tr>
<tr>
<td>Original</td>
<td>99.3</td>
<td>16</td>
<td>98.7</td>
</tr>
<tr>
<td>Sel. #1</td>
<td>98.9</td>
<td>15</td>
<td>97.7</td>
</tr>
<tr>
<td>Sel. #2</td>
<td>99.5</td>
<td>16</td>
<td>97.2</td>
</tr>
<tr>
<td>Oversamp.</td>
<td>98</td>
<td>18</td>
<td>91.8</td>
</tr>
<tr>
<td>Baseline</td>
<td>8</td>
<td>36</td>
<td>14</td>
</tr>
</tbody>
</table>

TABLE VI: Comparison of performance between individual networks and bagging

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Urban</th>
<th>Highways</th>
<th>Sharp turn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>Aut. MAD (%)</td>
<td>Aut. MAD (cm)</td>
<td>Aut. MAD (cm)</td>
</tr>
<tr>
<td>Weights #1</td>
<td>99.5</td>
<td>16</td>
<td>97.2</td>
</tr>
<tr>
<td>Weights #2</td>
<td>98.9</td>
<td>15</td>
<td>97.7</td>
</tr>
<tr>
<td>Weights #3</td>
<td>99.3</td>
<td>16</td>
<td>98.7</td>
</tr>
<tr>
<td>Weights #4</td>
<td>98.6</td>
<td>18</td>
<td>92</td>
</tr>
<tr>
<td>Weights #5</td>
<td>98.4</td>
<td>15</td>
<td>96.4</td>
</tr>
<tr>
<td>Bagging</td>
<td>99.5</td>
<td>16</td>
<td>98.7</td>
</tr>
</tbody>
</table>

[Work by my Valeo CIFRE PhD student Marin Toromanoff]
Real vehicle end-to-end driving (learnt by imitation)

[Work by Valeo using ConvNet trained by my CIFRE PhD student Marin Toromanoff]

Demonstration at CES’2018

The car stops on the barrier

[Work by Valeo using ConvNet trained by my CIFRE PhD student Marin Toromanoff]

Imitation Learning

- Aims at “copying” behavior of human drivers
- Requires large amount of real-world training data: video + driving-wheel AND brake-throttle data during many hours and thousands of varied-enough km driven by human
- Very promising generalization performances

BUT:
- Rare events under-represented in dataset
- Dangerous events unavailable in dataset !!

Outline

- Imitation Learning
- Reinforcement Learning
Preliminary RL experiment for end-to-end driving

[Work by my Valeo CIFRE PhD student Marin Toromanoff]

End-to-end driving from vision with DL, Pr. Fabien MOUTARDE, Center for Robotics, MINES ParisTech, PSL, 14/1/2020

Reinforcement Learning (RL)

Goal: find a “policy” $a_t = \pi(s_t)$ that

Maximizes $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \gamma \in [0, 1[$

Deep Reinforcement Learning (DRL) if Deep NeuralNet used as model (for policy and/or its “value”): DQN, Actor-Critic A3C

End-to-end driving: policy $\pi$ searched as ConvNet(front-image)
Value and Q-function

- Value of a state (for a given policy)
  \[ V_\pi(s) = \mathbb{E}_\pi[R_t|s_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{T} \gamma^t r_{t+k}|s_t = s] \]

THERE ALWAYS EXISTS A DETERMINISTIC OPTIMAL POLICY \( \pi^* \)
\[ \forall \pi, \forall s \in S, V_{\pi^*}(s) \geq V_\pi(s) \]

- Q-function
  \[ Q_\pi(s, a) = \mathbb{E}_\pi[R_t|s_t = s, a_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{T} \gamma^t r_{t+k}|s_t = s, a_t = a] \]


3 families of RL algorithms

- Policy-based \( \pi_\theta \approx \pi^* \)
  optimize a parameterized policy

- Value-based \( Q(s, a, \theta) \approx Q_{\pi^*}(s, a) \)
  find optimal (parameterized) Q-value
  \( \rightarrow \) then \( \pi(s) = \text{argMax } Q^*(s,a) \)

- Model-based \( m(s_t, a_t, \theta') \approx s_{t+1}, r_{t+1} \)
Deep Reinforcement Learning with Deep Q Network (DQN)

- **Q-learning**: $Q^{\text{new}}(s_t, a_t) \leftarrow (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot \left( r_t + \gamma \cdot \max_a Q(s_{t+1}, a) \right)$

- **Optimal policy deduced from optimal Q-value**

$$\pi^*(s) = \text{arg} \max_a Q(s, a)$$

- **DQN** [1]: if too many possible states, approximate Q as a neural network, and learn $Q^*$ using SGD with loss from Bellman equation

$$L(s_t, a_t, r_{t+1}, s_{t+1}, \theta) = (r_{t+1} + \gamma \max_a Q(s_{t+1}, a, \theta) - Q(s_t, a_t, \theta))^2$$


---

**Typology of RL algorithms**

<table>
<thead>
<tr>
<th>Famille</th>
<th>Algorithme</th>
<th>On/Off policy</th>
<th>Domaine</th>
<th>Action continu/discret</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basés sur la politique</td>
<td>REINFORCE</td>
<td>On policy</td>
<td>Les 2</td>
<td></td>
</tr>
<tr>
<td>Acteur Critique (A3C)</td>
<td>On policy</td>
<td>Les 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basés sur la fonction de valeur</td>
<td>SARSA</td>
<td>On policy</td>
<td>Discret</td>
<td></td>
</tr>
<tr>
<td>Q-Learning (DQN)</td>
<td>Off-policy</td>
<td>Discret</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basée sur un modèle</td>
<td>MCTS</td>
<td>Off-policy</td>
<td>Discret</td>
<td></td>
</tr>
<tr>
<td></td>
<td>iLQG</td>
<td>Off-policy</td>
<td>Continu</td>
<td></td>
</tr>
</tbody>
</table>
End-to-end driving from vision with DL, Pr. Fabien MOUTARDE, Center for Robotics, MINES ParisTech, PSL, 14/1/2020

End-to-end driving learnt by RL in racing-car simulator

**Performance**
Trained for 196 million steps

Test on training track

Snow (SE)

Network input and guided backpropagation

Game graphics

Activations

---


---
Choice of RL algo?

- **End-to-end driving from vision with DL, Pr. Fabien MOUTARDE, Center for Robotics, MINES ParisTech, PSL, 14/1/2020**

---

**RL for Automated Driving:**

- RL requires huge amount of trials AND ERRORS ⇒ Simulation REQUIRED (safety + speed)
- Still few driving simulators adapted for DL and RL, and best ones not totally mature

<table>
<thead>
<tr>
<th>Simulateur</th>
<th>GTA</th>
<th>DeepDrive.io</th>
<th>AirSim</th>
<th>CARLA [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexibilité</td>
<td>−</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Variété</td>
<td>++</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Complexité/Réalisme</td>
<td>++</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>Objets mobiles</td>
<td>++</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Vitesse éxecution</td>
<td>−</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Multi-agent</td>
<td>−</td>
<td>−</td>
<td>−</td>
<td>++</td>
</tr>
</tbody>
</table>

CARLA
Autonomous Driving challenge

• Itinerary to be followed in a city (given by 4 possible orders at intersections: Left, Straight, Right, Follow_Lane) BUT must stay on the road, in the lane, respecting Traffic Lights, and no collision with pedestrians and other cars!

• Evaluation metrics = Task completion & Distance between infractions, in an UNSEEN CITY

• Results (and 10,000$ for winners!) on 2019 July 1st

Network architecture

• U.S. Traffic lights ⇒ Need to use COLOR and high-enough resolution ⇒ big network, hard and slow to train
• Use a resnet-18 (10 times more weight than previously used in DQN-like network)
• Handle turn-orders (at intersections) with multi-head branching [1]

[1] Codevilla et al., End-to-end driving via Conditional Imitation Learning, 2017
Reward shaping

Rewards scaled in [-1, 1]:

- **Speed**: positive reward to follow speed, depends on obstacles & traffic light
- **Lateral position**: negative reward depending on distance to lane center
- **Episode terminates on collision, running red traffic light, too far from lane center or stuck (if no reason to stop)**

Examples of autonomous driving obtained with our DRL

Current Order: Left
Speed: 0.0 km/h

TL State: RED
Dist to TL: Close

Situation: US TL
Dist to center: 0.15
Conclusions & perspectives on DRL for Automated driving

- DRL allows to learn behavior *without any example provided by human*
- Only the REWARD needed to define objectives

- Very encouraging first results in simulation
- Winner of "vision-only" track at CARLA « Autonomous Driving challenge » !!

- Future work:
  - transferrability to real-world videos
  - Combination of Imitation-Learning and RL?