End-to-end driving from vision with Deep-Learning

Pr. Fabien MOUTARDE
Center for Robotics
MINES ParisTech
PSL Université Paris

Fabien.Moutarde@mines-paristech.fr
http://people.mines-paristech.fr/fabien.moutarde

Outline

• **End-to-end driving & IMITATION learning**
• Reinforcement Learning (RL)
• Deep RL (DRL) for Automated Driving
**Principle of “end-to-end” driving**

<table>
<thead>
<tr>
<th>Sensors (cameras, radar, LIDAR, etc...)</th>
<th>Perception</th>
<th>Trajectory Planning</th>
<th>Control</th>
<th>Driving wheel, Acceleration or braking</th>
</tr>
</thead>
</table>

Current architecture for automated driving

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

Imitation Learning or Reinforcement Learning

Sensors (cameras, radar, LIDAR, etc...) ➔ Imitation Learning or Reinforcement Learning ➔ Driving wheel, Acceleration or braking

Imitation Learning for end-to-end driving

“Copying” human driver

ConvNet output: steering angle

ConvNet input: Cylindrical projection of fisheye camera

Real data + “simulator” with real images

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.

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

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

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></td>
<td>Aut.</td>
<td>MAD (cm)</td>
<td>Aut.</td>
</tr>
<tr>
<td>Metric</td>
<td>(%)</td>
<td></td>
<td>(%)</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>

End-to-end driving from vision with DL, Pr. Fabien MOUTARDE, Center for Robotics, MINES ParisTech, PSL, Oct.2019

End-to-end driving (learnt purely on real-world data) tested in GTA simulator

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></td>
<td>Aut.</td>
<td>MAD (cm)</td>
<td>Aut.</td>
</tr>
<tr>
<td>Metric</td>
<td>(%)</td>
<td></td>
<td>(%)</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>13</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

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

Conclusion on IMITATION learning for end-to-end driving

Promising results

BUT

Difficult to cover large-enough variability of situations/environments with real driving data

Outline

• End-to-end driving & IMITATION learning
• Reinforcement Learning (RL)
• Deep RL (DRL) for Automated Driving
Reinforcement Learning (RL)

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)

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 the optimal (parameterized) Q-value

- **Model-based**
  $m(s_t, a_t, \theta') \approx s_{t+1}, r_{t+1}$
Value and Q-function

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

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

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

---

**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**
  \[ \pi^*(s) = \text{arg max}_a Q_\pi^*(s, a) \]

- **DQN [1]:** if too many possible states, approximate Q as a neural network, and learn Q* using SGD with loss from Bellman equation
  \[ L(s_t, a_t, r_{t+1}, s_{t+1}, \theta) = (r_{t+1} + \gamma \max_a Q(s_{t+1}, a, \theta) - Q(s_t, a_t, \theta))^2 \]

---

Summary of main RL algorithm types

<table>
<thead>
<tr>
<th>Family</th>
<th>Algorithm</th>
<th>On/Off policy</th>
<th>Discrete/continuous output?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy based</td>
<td>REINFORCE</td>
<td>On policy</td>
<td>Both</td>
</tr>
<tr>
<td></td>
<td>Actor-Critic (A3C)</td>
<td>On policy</td>
<td>Both</td>
</tr>
<tr>
<td>Value based</td>
<td>SARSA</td>
<td>On policy</td>
<td>Discrete</td>
</tr>
<tr>
<td></td>
<td>Q-Learning (DQN)</td>
<td>Off-policy</td>
<td>Discrete</td>
</tr>
<tr>
<td>Model based</td>
<td>MCTS</td>
<td>Off-policy</td>
<td>Discrete</td>
</tr>
<tr>
<td></td>
<td>iLQG</td>
<td>Off-policy</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

Outline

• End-to-end driving & IMITATION learning
• Reinforcement Learning (RL)
• Deep RL (DRL) for Automated Driving

Deep RL for automated driving

• Until recently, very few published research, and mostly in racing games:


• Up to now, only real driving with RL:
"Learning to Drive in a Day" (2018, [1])

  • Embed DRL in a real car, and learn « from scratch »
  • But VERY SIMPLE CASE: lane keeping along 250m!
  • Simulation used before to design architecture and find hyper-parameters

Preliminary DRL experiment for end-to-end driving

[Work by my Valeo CIFRE PhD student Marin Toromanoff]

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

**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

---


---

**RL for Automated Driving: why learn in a simulator?**

- **RL require huge amount of trial & error, and initial policy = very bad driving!**
  \[\Rightarrow \text{Learn in simulation} \text{ (for safety + speed)}\]

- **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 éxécution</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>

\[\Rightarrow \text{Choice of CARLA}\]

CARLA simulator

- Open source, flexible

---

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 July 1st
• Rainbow [1] = combination of many improvements of DQN [4] \( \Rightarrow \) currently SoA on ATARI benchmark

• IQN [2] = learning with probability distributions rather than just expectation of average

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Human Gap</th>
<th>Seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN</td>
<td>228%</td>
<td>79%</td>
<td>0.334</td>
<td>1</td>
</tr>
<tr>
<td>PRIOR.</td>
<td>434%</td>
<td>124%</td>
<td>0.178</td>
<td>1</td>
</tr>
<tr>
<td>C51</td>
<td>701%</td>
<td>178%</td>
<td>0.152</td>
<td>1</td>
</tr>
<tr>
<td>RAINBOW</td>
<td>\textbf{1189%}</td>
<td>\textbf{230%}</td>
<td>0.144</td>
<td>2</td>
</tr>
<tr>
<td>QR-DQN</td>
<td>864%</td>
<td>193%</td>
<td>0.165</td>
<td>3</td>
</tr>
<tr>
<td>IQN</td>
<td>1019%</td>
<td>218%</td>
<td>\textbf{0.141}</td>
<td>5</td>
</tr>
</tbody>
</table>

• Ape-X [3] multi-agent version of DQN allowing massively parallel distributed learning \( \Rightarrow \) Largely better performance, but typically require 22 billions of frames (vs. 200 millions)

DRL used: Rainbow + IQN + ApeX

- U.S. Traffic lights \( \Rightarrow \) Need to use COLOR and high-enough resolution \( \Rightarrow \) 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]

References:
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

Town02: Single Lane, EU
Weather: Heavy rain
Traffic Light: Red

Current Order: Left
Current Speed: 1.8 km/h
Conclusions & perspectives

- Very encouraging first results of RL: able to learn a kind of "Intelligent visual servoing" for Autonomous Driving

- Our approach ranked 1st (vision-only track) on CARLA "Autonomous Driving challenge"!!
  ➔ presentation at CVPR’2019 workshop

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

QUESTIONS?