Seminar "Reinforcement Learning in Autonomous Driving"
Final presentation

**Topic B) Transfer- and Deep Reinforcement Learning & Challenges**

Fabian Brunn, Leon Sievers

Supervisor: Sina Shafaei

29.06.2018, SS 2018
Outline

1. Introduction
2. Transfer Learning
3. Research Transfer Learning
4. Deep Reinforcement Learning
5. Research Deep Reinforcement Learning
6. Conclusion
7. Discussion
NVIDIA Automotive Team begins testing an autonomous vehicle using Deep Learning.
Nvidia uses supervised end-to-end learning

(Bojarski, Del Testa et al. 2016)
Nvidia’s main problems with this approach

• Supervised Deep Learning and not (Deep) Reinforcement Learning

• Needs ground truth data for training → Many hours of collecting driving data

• Has to encounter every edge case driving scenario if it wants a viable reaction

• Solution: Deep reinforcement learning enables to develop systems, that adapt in real world scenarios
Transfer Learning applies the knowledge gained in one environment to a new environment.
Transfer Learning - Basics

Transfer Learning problems consist of

- **Domains**
  - Source domain $\mathcal{D}_S$
  - Target domain $\mathcal{D}_T$

- **Tasks**
  - Source task $\mathcal{T}_S$
  - Target task $\mathcal{T}_T$
Transfer Learning - Basics

Domain
\[ \mathcal{D} = \{\mathcal{X}, P(X)\}, X = \{x_1, \ldots, x_n\}, \forall x_i \in X \]
\( \mathcal{X} \): Possible feature space
\( X \): Set of actual observed data features

Task
\[ \mathcal{T} = \{\mathcal{Y}, f(\cdot)\} \]
\( \mathcal{Y} \): Space of possible target variables
   - Regression: \( \mathcal{Y} = \mathbb{R} \)
   - Binary Classification: \( \mathcal{Y} = \{0,1\} \)
\( f(\cdot) \): Inference function, \( P(y_i|x_i), y_i \in \mathcal{Y} \)
## Transfer Learning - Categorization

<table>
<thead>
<tr>
<th>Inductive</th>
<th>Transductive</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{T}_S \neq \mathcal{T}_T$</td>
<td>$D_S \neq D_T, \mathcal{T}_S = \mathcal{T}_T$</td>
</tr>
<tr>
<td>$f(\cdot)$ is induced from the target task</td>
<td>The task is converted to another domain</td>
</tr>
<tr>
<td>The target task has labels available</td>
<td>The target task does not need labels</td>
</tr>
<tr>
<td>Pretraining Multi-task Learning</td>
<td>Domain Adaption Co-variate shift</td>
</tr>
</tbody>
</table>

These definitions can be applied to every ML setting, but we will focus on DL.

Domain
$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}, \mathcal{X} = \{x_1, \ldots, x_n\}, \forall x_i \in \mathcal{X}$

$\mathcal{X}$: Possible feature space

$\mathcal{X}$: Set of actual observed data features

Task
$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

$\mathcal{Y}$: Space of possible target variables

Regression: $\mathcal{Y} = \mathbb{R}$

Binary Classification: $\mathcal{Y} = \{0,1\}$

$f(\cdot)$: Inference function, $P(y_i|x_i), y_i \in \mathcal{Y}$
DNNs trained on classification tasks exploit Inductive TL
A CNN learns an internal feature extraction
We can take this idea a step further and really transfer learned weights to new problems.

Completely trained on capable hardware

~ 14,000,000 instances

Big Dataset, e.g. ImageNet

Convolutional Layers

Classification Header with Softmax

(Karpathy)
We can take this idea a step further and really transfer learned weights to new problems.

New Dataset with few instances e.g. BreaKHis

Convolutional Layers

Classification Header with Softmax

Frozen weights

Just change the Header

~ 9,000 instances

(Bayramoglu)
How well does it work

(Han, Wei et al. 2017)
We can even pre-train semantic traffic segmentation on ImageNet

(Karunakaran2018)
We can even pre-train semantic traffic segmentation on ImageNet

(Kim and Park 2017)(Hoffman, Wang et al.)
We can even pre-train semantic traffic segmentation on imagenet

(Kim and Park 2017)
DRL NN learn feature extraction too

(Shelhamer, Mahmoudieh et al. 2016)
We can change the network to support TL
Training with Maximum Entropy

\[ \pi(a_t | s_t) = P(a_t | s_t) \]

\[ \pi_{\text{MaxEnt}}(a_t | s_t) = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi}} \left[ r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot | s_t)) \right] \]
Transfer Learning with Maximum Entropy

(Haarnoja, Tang et al. 2017)
Transfer Learning with Maximum Entropy

\[ \pi_{\text{MaxEnt}}(a_t | s_t) = \arg\max_{\pi} \sum_t \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi}} \left[ r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot | s_t)) \right] \]

(Haarnoja, Tang et al. 2017)
Transfer Learning with Maximum Entropy

(Haarnoja, Tang et al. 2017)
## Transfer Learning - Categorization

<table>
<thead>
<tr>
<th>Inductive</th>
<th>Transductive</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{T}_S \neq \mathcal{T}_T$</td>
<td>$D_S \neq D_T, \mathcal{T}_S = \mathcal{T}_T$</td>
</tr>
<tr>
<td>$f(\cdot)$ is induced from the target task</td>
<td>The task is converted to another domain</td>
</tr>
<tr>
<td>The target domain has labels available</td>
<td>The target domain has no labels available</td>
</tr>
<tr>
<td>Pretraining Multi-task Learning</td>
<td>Domain Adaption Co-variate shift</td>
</tr>
</tbody>
</table>

These definitions can be applied to every ML setting, but we will focus on DL.

### Domain

$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}, \mathcal{X} = \{x_1, \ldots, x_n\}, \forall x_i \in \mathcal{X}$

$\mathcal{X}$: Possible feature space

$\mathcal{X}$: Set of actual observed data features

### Task

$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

$\mathcal{Y}$: Space of possible target variables

- Regression: $\mathcal{Y} = \mathbb{R}$
- Binary Classification: $\mathcal{Y} = \{0, 1\}$

$f(\cdot)$: Inference function, $P(y_i|x_i), y_i \in \mathcal{Y}$
Why not model our source domain? Simulations

(Sadeghi and Levine 2016)
Why not model our source domain? Simulations

(CAD)$^2$RL: Realistic Environment Test

2X speed up

Input: Monocular camera view
Overhead view

movement direction color map

(Sadeghi and Levine 2016)
Why not model our source domain? Simulations

Failure case

- Confused by:
  - Saturated light
  - Glassy window

(Sadeghi and Levine 2016)
Can we use it for autonomous driving?

Why not do the same thing for End 2 End learning in autonomous driving?
We will need a simulator - CARLA
Does it work that easily?

\[ P_S(X) \neq P_T(X) \]

(Hoffman, Wang et al.)
We can use Adversarial Domain Adaptation

(Hoffman, Wang et al.)
Does it work?

(Hoffman, Wang et al.)
Domain Adaption paired with capable simulators enables training of Deep Reinforcement Learning in autonomous driving
Audi partners with Israel's autonomous vehicle simulation startup Cognata

Ankit Ajmera

(Reuters) - German carmaker Audi AG (NSUG.DE) has partnered with autonomous vehicle simulation platform provider Cognata Ltd to speed up the development of autonomous vehicles, the Israel-based startup said on Tuesday.
Deep Learning Autonomous Simulation

Accurate, Scalable, 1000x Faster

23,474 Miles Driven So Far
Outline

1. Introduction
2. Transfer Learning
3. Research Transfer Learning
4. Deep Reinforcement Learning
5. Research Deep Reinforcement Learning
6. Conclusion
7. Discussion
What is Deep Reinforcement Learning for AD?

(Wirecutter), (Dwivedi 2017)
Setup of an exemplary DRL system with DQN

Environment: TORCS

Current state $S_t$

Current action $A_t$

Rewards $R_t$

Experience replay memory

$S_{t-1}, A_{t-1}, S_t, R_t$

$S_{t-1}, A_{t-1}, S_t$

$S, A, S', R$

$Q$ – Network (CNN)

$\varepsilon$-greedy exploration and exploitation

Loss calculation

$L = R + \gamma \max_{a'} Q'(S', a') - Q(S, A) + \frac{\partial L}{\partial w}$

Future Rewards

$\gamma \max_{a'} Q'(S', a')$

Target

$R + \gamma \max_{a'} Q'(S', a')$
Advantages of DRL in autonomous driving

Avoid expensive subsystems like driving trajectory planning

One step from pictures to motor commands instead of several

Better performance and smaller hardware systems due to one system

(Bojarski, Del Testa et al. 2016), (Hammerschmidt 2017), (Silver 2018), (Baldwin 2018)
Challenges of DRL in autonomous driving

• DRL system is a blackbox, thus it is very hard to fully guarantee functional safety

• No consideration of time series data, only momentary images
  → No pedestrian / cyclists / … tracking

• Standard DRL problems like generalization, instability, local optima are all not completely solved yet and still need a lot of research

(Irpan 2018)
Outline

1. Introduction
2. Transfer Learning
3. Research Transfer Learning
4. Deep Reinforcement Learning
5. Research Deep Reinforcement Learning
6. Conclusion
7. Discussion
Different reward system needed, no concurrent game score available
Deepmind proposes velocity and tracking dependent reward

- Deepmind: Reward consists of velocity and tracking
  
  \[ R_t = v \times \cos \theta \]

- Problem: Sliding along guard rails

(Mnih, Badia et al. 2016)
Reward is naïvely coupled with the distance to the middle of the road / lane

- Goal: Avoid sliding along track barrier, enforce staying on track

- $R_{\text{dist}}(d) = 2 - (|d| + 1)\kappa$, $\kappa > 0$

- $d$ is normalized distance from middle of the track

- $\kappa$ is an enforcing factor to punish deviation

Problem: Vehicle wiggles along middle of track

(Wolf, Hubschneider et al. 2017)
Punish wiggling by adding a heading angle

- Introduction of heading angle in the reward function by Perot et al.
- \( R = v \times (\cos \theta - d) \)
- It rewards speed and heading towards the curvature of the road

(Perot, Jaritz et al. 2017)
Complex reward function to avoid wiggling

The action reward depends if the agent’s state-action pair is part of three scenarios:

- \( \{B\} \) is a set on straight lines, on which the agent is close to the centerline and his heading angle remains

- \( \{C\} \) is a set if the agent increases the heading angle

- \( \{D\} \) is the set of situations where the agent counter steers

\[
R_{\text{dist}}(d) = 2 - (|d| + 1)^\kappa, \quad \kappa > 0
\]

\[
R_{\text{action}}(S,A) = \begin{cases} 
+1, & (S,A) \in \{B\} \\
-1, & (S,A) \in \{C,D\} \\
0, & \text{else}
\end{cases}
\]

\[
R_t = \max(-2, R_{\text{dist}}) + R_{\text{action}}
\]
Results for the improved reward functions

Average distance error in m.

<table>
<thead>
<tr>
<th>Reward Type</th>
<th>Training Track</th>
<th>Evaluation Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance-based</td>
<td>0.552</td>
<td>0.769</td>
</tr>
<tr>
<td>action-based</td>
<td><strong>0.167</strong></td>
<td>0.335</td>
</tr>
<tr>
<td>human</td>
<td>0.211</td>
<td><strong>0.198</strong></td>
</tr>
</tbody>
</table>

(Wolf, Hubschneider et al. 2017)
Conclusion

• Currently rather fundamental research on Deep Reinforcement Learning in autonomous driving

• High potential for Deep Reinforcement Learning as part of a larger autonomous system

• Transfer Learning necessary to make training feasible by pretraining in simulations

• Robust architectures and training methods are developed in order to tackle the problems of real environments and safety
Thank you very much for your attention!
Any questions?
Bibliography (1)


Wirecutter Picture: Dashcam., https://thewirecutter.com/reviews/best-dash-cam/%0A

Bibliography (2)


Cheng, M. "CARLA Open-Source Stimulator to Accelerate Driverless Car Development.", http://www.futurecar.com/1669/CARLA-Open-Source-Stimulator-to-Accelerate-Driverless-Car-Development%0A


Hui, J. "Visualization of AlexNet filter weights.", https://jhui.github.io/assets/cnn/cnnfilter.png


