From Deep Learning to Autonomous Driving

J. Marius Zöllner
FZI Research Center for Information Technology at the KIT
Karlsruhe, Germany
(Deep) Learning for Autonomous Driving

- Perception
- Understanding
- Prediction
- Localization
- Action
- Sensor-Fusion

Advanced Learning Frameworks

Deep Architectures and methods for various problems
- Multitask – Networks
- Representation of uncertainty,…
(Deep) Learning for Autonomous Driving

- **Network Architectures and Modules**
  - Fully Convolutional Networks [Long2015]
  - GoogLeNet: Inception module [Szegedy2015,2017]
  - Residual Networks [He2016]

- **Sensor Fusion & Calibration**
  - RegNet [Schneider2017]

- **Temporal Fusion & Localization**
  - PoseNet [Kendal2015]
  - Monocular Visual Ego Motion [Weber 2017]
(Deep) Learning for Autonomous Driving

- Perception
  - Object Detection
    - Faster R-CNN [Ren2015]
    - YOLO variants [Redmon2016]
    - DeepTLR [Weber2016]
  - 2D Semantic Segmentation
    - Pyramid Scene Parsing Network [Zhao2016]
    - Mask R-CNN [He2017]
  - Multitask Networks
    - Semantic Instances and Depth Layering [Uhrig2016]
    - MultiNet [Teichmann2016]
(Deep) Learning for Autonomous Driving

- Scene Understanding and Prediction
  - HD Map Parameters [Bittel 2017]
  - Dynamic Occupancy Grid Prediction [Hoermann2017]
  - Interactive Scene Prediction [Lenz2017]

- Action
  - End-to-End Approaches:
    - Direct Control [Bojarski2016]
    - Driving Models [Xu2017]
    - Combination with probabilistic models [Hubschneider2017]
  - …
Will Learning Systems solve Autonomous Driving?

➢ Pros and Cons

- Adapt to sensors and cover uncertainties
- Generalize and adapt to changing conditions
- Learning (instead of modelling) open world including human behavior
- Modern learning methods are powerful

- Best practice?
- Interpretability and validation? (white-box vs. black-box)
- Provable properties (robustness)?
- Constraint learning or runtime validation?
- ....
Why is it still challenging and what to do?

- Deal with to unknown environment or unknown behavior
  - Additional control input required
Why does it work?

\[ \theta = \]

\[ i_t = \]

\[ \frac{\partial \theta}{\partial i_{xyc}} = \]
Why is it still challenging and what to do?

- Result strongly dependent on input
- All situations have to be represented
One possible approach: Active Learning

\[ \{(X_i, Y_i)\}_{i=1}^{n} \]

\[ \{X_i\}_{i=1}^{m} \]

Selective/Pool-based sampling e.g. based on confidence

\[ \phi_{LC} = 1 - P(\hat{f}_{n,m}|X_i) \]

- Uncertainty measures are needed
- (Incremental Learning needed)
Uncertainty: Example for Prediction with FCN

- stacked input at $t$
  - road
  - position
  - velocity
  - direction
  - turn signal

- post processing
  - position / velocity prediction at $t + \Delta t$

### 6. Evaluation

After having scouted for some viable architectures, by observing their training and evaluation loss, it's time to take a closer look at them. While a low loss guarantees some degree of quality we want to quantify the results to get some more intuitive and representable figures. This requires a separate evaluation of the data. This evaluation is not based on a loss function, but on an image processing algorithm, which segments and compares the outputs and labels. We cannot use this algorithm during training to obtain a more sophisticated loss function, simply because it would take up too much time to process during training.

In 6.1 we will highlight the differences between our labels and the output data to get a better understanding of what we are dealing with and what the algorithm has to accomplish. This is followed up by the actual step by step explanation of our algorithm in 6.2. Finally, we use the algorithm on our most relevant networks in 6.3 and present the results.

#### 6.1. Outputs

It is extremely unlikely for a neural network to perform a regression on an entire image flawlessly, especially on unseen data. A variety of errors can occur. The predicted values could be slightly off, the vehicles could be misplaced or even missing entirely and new vehicles could have been added for some reason. All of these errors are closely related to the task and we want to quantify them.

![Colormap used to visualize output accuracy. Blue represents low value areas and red represents high value areas. The white value was added to differentiate between low value predictions and the background.](image)

To visualize the network predictions we use the color mapping displayed in figure 6.1. For each output sample we will provide the respective input data and the target prediction. The road layout is rendered in all images to help with comparisons. For the input images we decided to additionally only display the positions (black) and turn signals (orange).

The output generated by our networks varies greatly between architectures. In figure 6.2 we depict the outputs generated by a select set of network configurations. While the hidden layer architecture produces some form of heatmap with soft edges and low value clutter in random locations, 45 FCN

---

Jonathan Härtl: Learned Micro-Scale Traffic Prediction using Neural Networks, Thesis 2017

© FZI Forschungszentrum Informatik 16.10.17
Uncertainty: Example for Prediction with FCN

- Using Monte Carlo dropout [1],
- N forward passes with dropout, Calculate statistics from sample set

Jonathan Härtl: Learned Micro-Scale Traffic Prediction using Neural Networks, Thesis 2017
Adaptive Autonomous Behavior
New Concept: Unified Model

Probabilistic Graphical Models
- Tackle uncertainties
- Maximize Information Fusion
- Bayesian Filter Theory, Evidence Theory
- Reduce Complexity: Factorization of large problems and Object Orientation

Machine Learning
- Adapt to observations
- Solve sub problems
- Potentially provide mechanisms for continuous learning
- Provide uncertainty measures
Power of Probabilistic Graphical Models

- Observable and Hidden Variables
- Temporal Model
- Global and Ego-centric Model
- Hybrid State Space Representations
- Message Passing Schedules
- Parameter Learning
- Modularity
- Hierarchy
- Classes and Instances
- Polymorphism
- Relations
- Probabilistic Dependencies

$P(X|Y,Z)$
$\psi(X,Y,Z)$
Example: Combining End-to-End-Learning with Particle Swarm Optimization Planner

- Sampling of possible trajectories
- Obstacle in driving path detected
- Parking vehicle invalidates most proposed trajectories
- Sampling initialization using predicted trajectories based on visual input
Example: Combining End-to-End-Learning with Particle Swarm Optimization Planner

Combination of deep learned intuition and modular safety constraints

Established modular environment perception

Sensor (LIDAR) measurements

Camera Images

Sensor Measurements

Actuator Control Values

Particle Swarm Optimization

Deep learned trajectory proposal

Integrating End-to-End Learned Steering into Probabilistic Autonomous Driving, [Hubschneider 2017]
Integrating End-to-End Learned Steering into Probabilistic Autonomous Driving, [Hubschneider 2017]

© FZI Forschungszentrum Informatik
Challenges of Probabilistic Graphical Models

- Hierarchical Modularization
- Interfaces
- Classes and Instances
- Learning Complex Dependencies
Future Research for Learning Systems

**General**
- Continuous lifelong learning
- Multi-task learning
- Incremental learning
- One-shot learning
- Self-assessment and self-expression
- Exploration
- Correction and repair (forgetting)

**AD specific**
- Architecture
- Heterogeneous data
- Change between learning modes
- Reliability and Validation (open world)
- Benchmarks data sets and software infrastructure
- Transfer learning (simulation)
Learning is valuable but should be driven by AD
MANY THANKS