Testing for autonomous driving

Testing Deep Neuronal Networks

Jakob Krause

Freie Universität Berlin, Department of Mathematics and Computer Science,
Institute of Computer Science, Software Engineering Group

May 8, 2019

Outline

- significant progress in machine learning (ML) did lead to safety critical systems like autonomous cars
- companies like Google (Waymo), Tesla, BMW are building and testing those cars
- even with the big progress and money in the field the industry already produced some big accidents

	Date	Cause	Outcome
Hyundai	2014	Rain fall	Crashed while testing
Tesla	2016	Image contrast	Killed the driver
Google	2016	Failed to estimate speed	Hit a bus
Uber	2018	too slow detection no emergency brakes	Killed pedestrian

Table 1: Example of real-world accidents involving autonomous cars

=> Most of these accidents happen in rare corner cases!

- a graph with weighted edges (nodes are called neurons)
- nodes contain non linear functions(activation function)
- receives an input and returns a value (f.I. classification)
- ’trained’ with labeled data
- tested with labeled data different to training set

What is a neural network

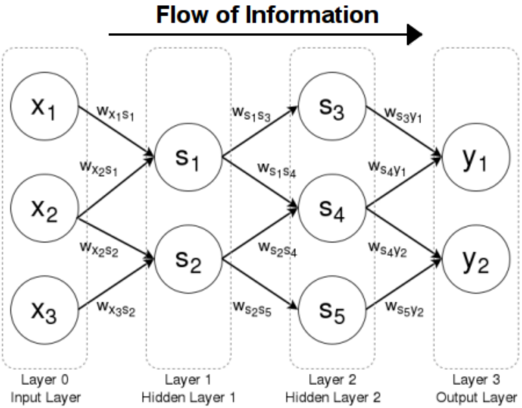

Figure 1: Simple neural network

What is a deep neural network

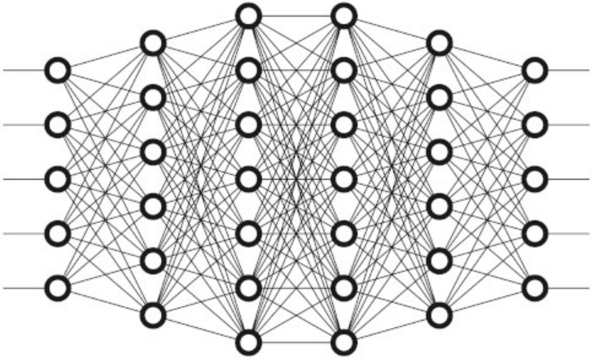

Figure 2: Deep neural network

What is an autonomous car?

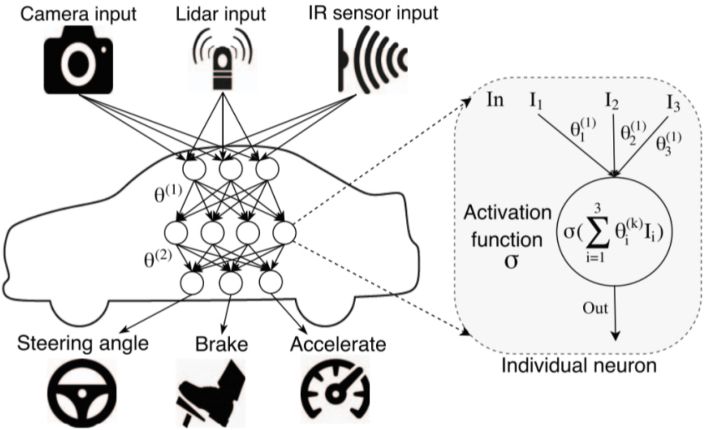

Figure 3: Simple autonomous car

- On a conceptual level the error prone corner case a comparable to logic bugs in regular software
- Similar to a bug detection and patching cycle in traditional software, detected bugs are fixed by adding error inducing training data

Why is it difficult to test DNNs

- logic of a DNN is expressed by neurons and weights instead of control statements (traditional software)
- definition and image space is hughe
- finding inputs that will result in high model coverage in a DNN is significantly because of non-linearity
- manually creating specifications for complex DNN systems like autonomous cars is infeasible as the logic is too complex to manually encode as it involves mimicking the logic of a human driver

Limitations of existing DNN testing

- expensive labeling effort
  - often data needs to labeled by hand
- low test coverage
  - no systematic approach for covering different rules of the network
  - Example : Splitting data sets in training set and testing sets
  - => no guarantee testing set will test all learned rules

Problem of low-coverage DNN tests

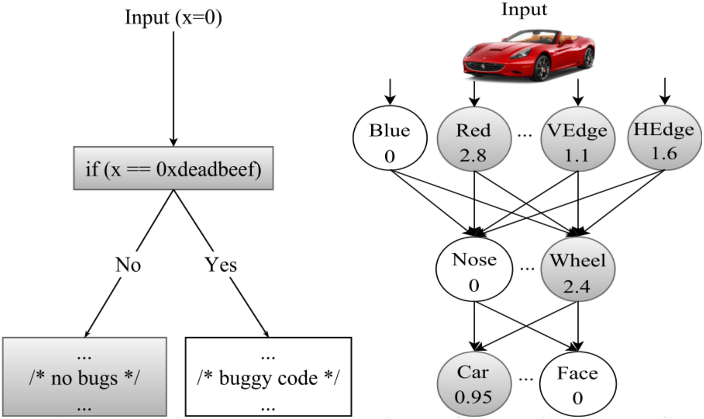

Figure 4: Flow of traditional code compared to NN

- testing methodology for automatically detecting erroneous behaviors of DNN-based software of self-driving cars
- leverage the notion of neuron coverage (i.e., the number of neurons activated by a set of test inputs) to systematically explore different parts of the DNN logic
- empirically demonstrate that changes in neuron coverage are statistically correlated with changes in the actions of self-driving cars (e.g., steering angle)
- demonstrate that different image transformations that mimic real-world differences in driving conditions like changing contrast/brightness, rotation of the camera result in activation of different sets of neurons in the self-driving car DNNs

- IO-space is big for autonomous vehicles $\rightarrow$ we need a partition

Neuron Coverage

$$\text{Neuron Coverage} = \frac{|\text{Activated Neurons}|}{|\text{Total Neurons}|}$$

Assumption 1

All inputs with similar Neuron Coverage have similar behaviour

Increasing Coverage with Synthetic Images

- generated inputs need to be realistic
- applying transformations to seed images

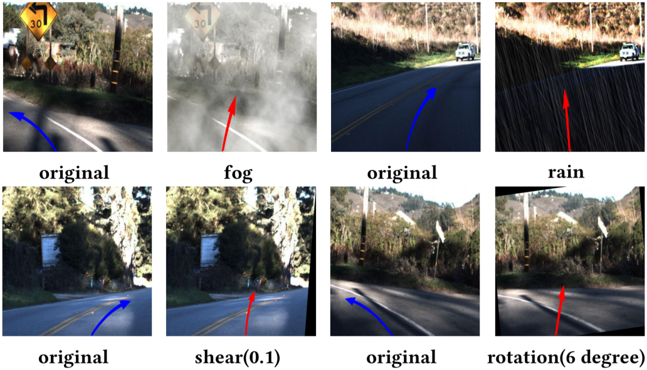

Figure 5: Example of some applied transformations

Combining Transformations to Increase Coverage

- using multiple transformations to increase coverage
- space of applied transformations is very big $\rightarrow$ heuristic

Test Oracle with Metamorphic Relations

- Metamorphic testing (MT) is a property-based software testing technique, which can be an effective approach for addressing the test oracle problem and test case generation problem
- Simple example $sin(\pi - x) = sin(x)$
- define relationships between the car's behaviors across certain types of transformations
- f.i. steering angle should not change significantly under changing weather conditions
- $steeringAngle(Img_{original}) = steeringAngle(Img_{synthetic})$

- steering angle can be a little different but still correct
- trade-of in being more strict with more false positive and visa versa
- angle for original images Θ_{oi}
- angle for transformed images Θ_{ti}
- angle for manual labels $\hat{\Theta}_i$
- $MSE_{orig} = \frac{1}{n} \sum_{i=1}^{N} (\hat{\Theta}_i - \Theta_{oi})^2$

Metamorphic relation

$$(\hat{\Theta}_i - \Theta_{ti})^2 \leq \lambda MSE_{orig}$$

Implementation

- How do we benchmark our proposal
- picked 3 strong AIs based on the Keras Framework from the Udacity Autonomous driving challenge
- lets see if we can error and measure there behaviour

Research question 1

Do different input-output pairs result in different neuron coverage ?

- Steering angel: neuron coverage correlates with statistical significance
- Steering direction: neuron coverage varies with statistical significance

Result 1

Neuron coverage is correlated with input-output diversity and can be used to systematic test generation

Research question 2

Do different realistic image transformations activate different neurons ?

Result 2

Different image transformations tend to activate different sets of neurons

Research question 3

Can neuron coverage be further increased by combining different image transformations

Result 3

By systematically combining different image transformations, neuron coverage can be improved by around 100% w.r.t. the coverage achieved by the original seed images.

Research question 4

Can we automatically detect erroneous behaviors using metamorphic relations?

Result 4

With neuron coverage guided synthesized images, DeepTest successfully detects more than 1,000 erroneous behavior as predicted by the three models with low false positives.

Research question 5

Can retraining DNNs with synthetic images improve accuracy?

Result 5

Accuracy of a DNN can be improved up to 46% by retraining the DNN with synthetic data generated by DeepTest

Conclusion

- tool for automated testing of DNN-driven autonomous cars
- maximizing neuron coverage with synthetic test images by realistic transformations
- using domain typical metamorphic relations to find erroneous behaviours without detailed specification

- change of camera lenses are not considered f.i when its raining
- transformations are not very realistic
- no complex scenes (like snowy scenes)

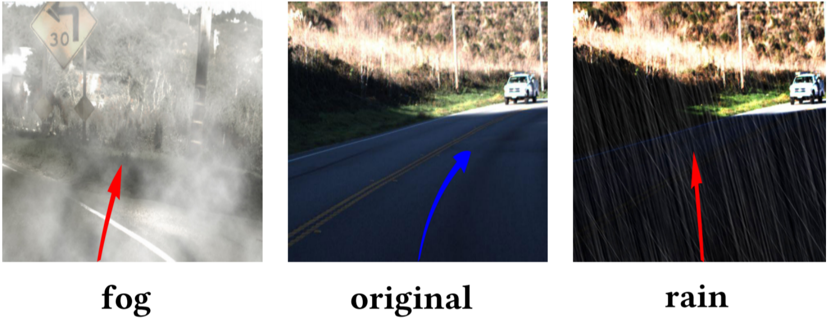

Figure 6: Unrealistic Transformations

- GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems
- introduces input validation
- also uses metamorphic relationships
- technical paper
- improves synthetic images processing by using generative adversarial networks

Input validation

- in classical software we found to catch malformed data. For example a webserver wants to stop processing when the json is malformed
- NN would be much safer if there suddenly can tell on a snowy road that the cant process this environment
- any 640x480 RGB picture $\rightarrow$ to weak
- only data from training set $\rightarrow$ to strong

$$min_j ||h(i) - h(j)||_2 < \Theta$$

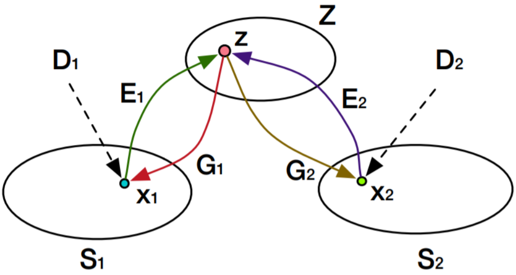

Figure 7: Transformations Structure

- x_1 and x_2 same environemnt with different weather (sunny and rainy), S_i Domains (sunny and rainy), Z space where x_1 and x_2 are same, E_i encoders to a space Z , G_i generators

Improved sythetical images via GAN

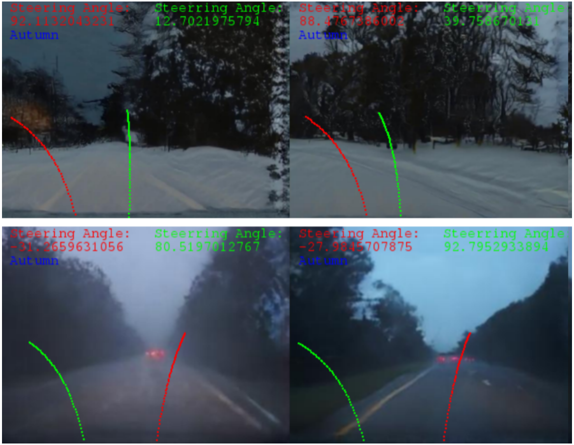

Figure 8: GAN Transformations

Conclusion

- proposed an unsupervised learning framework to synthesize realistic driving scenes to test inconsistent behaviors of DNN-based autonomous driving systems
- successfully detect thousands of inconsistent behavior

Resume

- We have seen 2 approaches for testing deep neuronal networks
- Neuron coverage as a metric of test coverage
- Metamorphic relations as a test oracle
- Synthetic image generation to increase coverage and safety
- concept of input validation for DNNs

Questions?

References I

[1] **Yuchi Tian et al.** “DeepTest: Automated Testing of Deep-neural-network-driven Autonomous Cars”. In: *Proceedings of the 40th International Conference on Software Engineering*. ICSE ’18. Gothenburg, Sweden: ACM, 2018, pp. 303–314. DOI: 10.1145/3180155.3180220. URL: http://doi.acm.org/10.1145/3180155.3180220.

[2] **Mengshi Zhang et al.** “DeepRoad: GAN-based Metamorphic Autonomous Driving System Testing”. In: *CoRR* abs/1802.02295 (2018). arXiv: 1802.02295. URL: http://arxiv.org/abs/1802.02295.