Explaining what learned models predict.

In which cases can we trust machine learning models and when is caution required?

Abstract

There are many problems in computer science that cannot or can only be solved with extreme difficulty using pre-programmed rules. An example here would be the recognition and classification of images. Here, machine learning (ML) algorithms offer a good solution approach by recognizing regularities from previous examples, storing them, and applying them to the new images.

However, the safety and reliability of machine-learning systems cannot be readily assessed because the individual steps in the learning process of an ML system are not easily comprehensible to a human, making the decision path very opaque.[1]

Given the ever-increasing impact of ML systems in our lives, it is very important that these systems are reliable since wrong or unintended results could have extreme consequences.

Contents

[1] see Fraunhofer-Institut für Kognitive Systeme IKS: SAFE AI: Absicherung von Künstlicher Intelligenz (KI), in: Fraunhofer-Institut für Kognitive Systeme IKS, [online] https://www.iks.fraunhofer.de/de/themen/kuenstliche-intelligenz/absicherung-ki.html [16.03.2021].

1 Reliability issues

There are many issues and potential pitfalls that have to be taken care of when applying an ML model in the real world. Especially if the decisions and/or predictions of the model could have major consequences. In the following two subchapters, two (of many) issues are selected to give an example.

1.1 Robustness

A current problem with machine learning algorithms is that slightly different inputs can often lead to completely different results. Moreover, it is often very difficult to find out why the wrong result came out and the only option one has is to train the model with another, better training dataset and hope that the error has been eliminated.

In a project on July 01, 2019, the Fraunhofer Institute used spectral relevance analysis to identify the features of an image that are relevant for an image classification. The system was supposed to classify animal images. However, upon later analysis, it turned out that the system did not focus on the horse at all, but assigned all images with a copyright notice to the category horse. This could be due to the fact that an excessive number of horse pictures in the training data set had a copyright notice. In this case, the ML system would most likely misclassify a horse image without copyright notice or possibly even assign a completely different animal with copyright notice to the class horse.[2]

Figure 1: Spectral Relevance analysis on a horse – The AI system mainly uses the copyright statement as a relevant characteristic to classify a horse.[2]

This problem is especially important for safety-critical applications such as autonomous driving since lack of robustness could easily be used by cyber-criminals as a point of attack. For example, an attacker could use manipulated input data (e.g., sensor data, telemetry data from other road users) to manipulate an ML system of another autonomous car in such a way, that it causes an unanticipated action in the car (Adversarial Attack).

Therefore, it is very important to ensure that the training dataset contains data of highest quality and no biases or distracting regularities.

1.2 ML-systems outsmarting its creators

Apart from issues related to the quality of the training dataset there are also many instances where ML systems found new creative ways to optimize their loss function that where not intended by its creators.

An example is given in the paper “Robots that can adapt like animals” where the authors studied how robots can adapt to a wide variety of injuries. The goal was that if one of its legs failed, the simulated robot would use a trial-and-error approach to figure out a compensatory behavior that would allow it to still walk despite the damage. The damaged leg was stimulated by telling the robot to minimize the time the affected leg touches the ground. At first, the authors only let between one and three legs fail but after they got very good results from these experiments, they decided to test what happens if they let all six legs fail. Initially, the authors expected the robot to stop working, but the simulation showed that the robot is able to walk with

[2] see Fraunhofer-Gesellschaft: A look inside neural networks, in: Fraunhofer research news, 01.06.2019, [online] https://www.fraunhofer.de/en/press/research-news/2019/july/a-look-inside-neural-networks.html [17.03.2021].

zero ground contact. After visualizing the simulation, it became clear why. The robot just flipped itself upside down and walked on his elbows.[3]

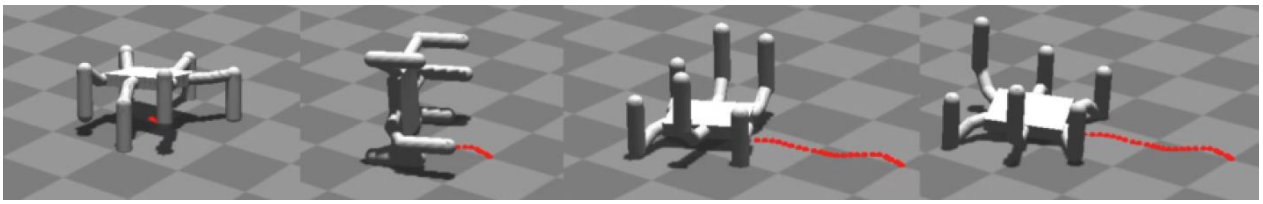

Figure 2: Elbow-walking gait. The robot was tasked to minimize the contact with the ground for a given number of legs. After tasked with the challenge to minimize the ground contact for all six legs the robot just flipped on its bag and walked on its elbows.[4]

Apart from this there are also multiple examples where the ML system surprised its creators by exploiting some bug in the physics engine to get a better result at the given task. Since these optimizations are often undesirable and/or impossible in the real world, care must be taken when applying these simulated models to the real world.

2 Techniques for ensuring reliability

Just as in the previous chapter only two of many reliability problems were exemplified, this chapter also only examines some selected approaches to detect and prevent reliability problems.

2.1 Domain randomization

ML Systems are often trained in simulated environments due to the huge benefit of efficiency and parallelism. However, how is it possible to reliably transfer a model that was trained in a simulation to the real world? One popular approach is to use Domain Randomization. Domain Randomization helps "bridging the 'reality gab' that separates simulated robotics from experiments on hardware"[5] by introducing randomness into the simulation. This randomness can be introduced into the simulation in many ways, for example by distorting some pixels in the image, introducing random textures in the background, switching light positions, or using random positions and orientations of the camera.

The goal of Domain Randomization is to test if the model is able to generalize well by providing great simulated variability.[6]

2.2 Out of Distribution Detection

Another different approach to reduce misclassification of ML Systems and thereby increasing security and reliability of ML Systems is out-of-distribution (OOD) detection. OOD detection comprises various methods that aim to detect inputs that differ strongly from the samples included in the training set.[7]

These algorithms would then prevent the ML model from making a prediction, since it would most likely be wrong anyway. This is especially important for ML models that are deployed into the real world where

[3] see Cully, Antoine/Jeff Clune/Danesh Tarapore/Jean-Baptiste Mouret: Robots that can adapt like animals, in: Nature, Jg. 521, no. 7553, 2015, doi: 10.1038/nature14422, p. 504.
[4] Lehman, Joel: The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities, in: arXiv.org - Cornell University, 09.03.2018, [online] https://arxiv.org/abs/1803.03453v4 [18.03.2021], p.14.
[5] Tobin, Josh/Rachel Fong/Alex Ray/Jonas Schneider/Wojciech Zaremba/Pieter Abbeel: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, in: arXiv.org, 20.03.2017, https://arxiv.org/abs/1703.06907 [31.03.2021], p.1.
[6] see Tobin et al., 2017.
[7] see Henne, Maximilian/Adrian Schwaiger/Karsten Roscher/Gereon Weiss: Benchmarking Uncertainty Estimation Methods for Deep Learning With Safety-Related Metrics, in: Artificial Intelligence Safety 2020, 2020, http://ceur-ws.org/Vol-2560/, p. 86.

errors of the end user could lead to dramatic misclassifications. To give an example: an image classifier used to detect pneumonia in chest X-Ray images should not try to make a prediction when the end user accidently tries to classify an X-Ray image of a knee.

2.3 Explainable AI

The research area of Explainable AI can also help ensuring reliability of ML Systems. Explainable AI combines all methods that point out on which features the prediction of an ML system was based. An example of such a method is the spectral relevance analysis that was briefly mentioned in the example of chapter 1.1 Robustness. These methods are especially helpful for finding suitable algorithms and analyzing false predictions that the ML system makes.[8]

3 Summary

To return to the initial question "In which cases can we trust ML models and when is caution required", it is really important to have ML models that are robust and reliable, especially when deployed in critical applications. It is impossible to know the predictions of a complex ML system for every possible input and thus it might also be impossible to create ML Systems that have absolutely accurate predictions. However, techniques such as domain randomization and OOD detection can help to drastically increase the reliability of ML systems. Additionally, "Explainable AI" techniques such as the spectral relevance analysis can help to find flaws and biases in the training data set. In the end, it is a weighing of risks and potential. Whereby the question often arises whether the decisions of a human would really be better than those made by the ML system.

4 References

Cully, Antoine/Jeff Clune/Danesh Tarapore/Jean-Baptiste Mouret: Robots that can adapt like animals, in: Nature, vol. 521, no. 7553, 2015, doi:10.1038/nature14422, pp. 503–507.

Fraunhofer-Gesellschaft: A look inside neural networks, in: Fraunhofer research news, 01.06.2019, https://www.fraunhofer.de/en/press/research-news/2019/july/a-look-inside-neural-networks.html (last accessed on 17.03.2021).

Fraunhofer-Institut für Kognitive Systeme IKS: SAFE AI: Absicherung von Künstlicher Intelligenz (KI), in: Fraunhofer-Institut für Kognitive Systeme IKS, n.d., https://www.iks.fraunhofer.de/de/themen/kuenstliche-intelligenz/absicherung-ki.html (last accessed on 16.03.2021).

Henne, Maximilian/Adrian Schwaiger/Karsten Roscher/Gereon Weiss: Benchmarking Uncertainty Estimation Methods for Deep Learning with Safety-Related Metrics, in: Artificial Intelligence Safety 2020, 2020, http://ceur-ws.org/Vol-2560/, pp. 83–90.

Lehman, Joel: The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities, in: arXiv.org - Cornell University, 09.03.2018, https://arxiv.org/abs/1803.03453v4 (last accessed on 18.03.2021).

Tobin, Josh/Rachel Fong/Alex Ray/Jonas Schneider/Wojciech Zaremba/Pieter Abbeel: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, in: arXiv.org, 20.03.2017, https://arxiv.org/abs/1703.06907 (last accessed on 31.03.2021).

[8] see Henne et al., 2020, p. 86.