COMP6915

Adversarial Attacks on ML Models

Rob Bishop and Laura Graves

Abstract

Machine learning models are susceptible to *adversarial examples*: examples that should be within the decision boundary for a certain class yet are classified as another. These adversarial examples can be found efficiently even through obfuscated gradients or against black-box models. We offer a survey of current research, covering both attack and defense methods as well as attempting to give a theoretical understanding of adversarial attacks.

Adversarial examples are inputs to Machine Learning (ML) models that, while being arbitrarily close to a real example in the input space, cause the model to misclassify. These examples were first shown as an intriguing quirk of ML models that suggested shortcomings in the generalizability of models [1] but have subsequently been shown to be remarkably resilient to attempts to stop them - recent efforts have shown they can be found even against black box networks or networks with non-differentiable layers. In this survey we give a brief history of adversarial attacks that leads us to a generalizable definition of an adversarial example. We will look at some defence techniques against these attacks as well as some domain-specific attacks against different real-world models. This further raises that question of if being able to find adversarial examples in arbitrary ML models is a signifier of an inability of machine learning to realistically generalize in the same way a human mind can - in other words, we ask if the presence of adversarial examples gives any hints about if computers can learn to embrace the universality of the human condition.

1 History

Early research on neural networks showed fantastic performance on visual and audio classification tasks, but some interesting and unexpected properties started emerging. Szegedy et al. [1] found that neural networks learned significantly discontinuous input-output mappings, leading to them being able to cause a network to misclassify an input by applying barely perceptible perturbations aimed at maximizing the loss of the network to the input or hidden layers. They theorized that these examples - termed *adversarial examples* - while having a low probability of occurring naturally or being found by random sampling are nonetheless able to be found by maximizing the loss at a hidden or input layer. They further theorized that training on these adversarial examples can be used as a method of regularization that, while prohibitively computationally expensive, can outperform dropout. Finally, another observation of theirs was that perturbations that cause one network to misclassify often cause networks with different architecture trained on the same data to make the same misclassification, a property that would come to be known as transferability.

Shortly after, Goodfellow et al. [2] published a monumental work on adversarial examples. They introduced the Fast Gradient Sign Method (FGSM), an algorithm using the gradient found through backpropagation to generate adversarial examples. The efficiency of FGSM allowed for some remarkable breakthroughs. Adversarial examples could be found efficiently enough that they could be used to iteratively train a model on, providing a method of regularization that not only provided a resilience against overfitting on large models but also provided models with a resistance to adversarial examples, which we term *robustness*. The universal approximator theorem states that a deep neural network with at least one hidden layer should be able to approximate any function, including functions that are not vulnerable to adversarial examples. Their finding was that this adversarial training was effective only when the model was sufficiently able to learn a robust function.

Both Szegedy et al. [1] and Goodfellow et al. [2] noted that models with vastly different architecture trained on the same data tended to misclassify inputs with the same perturbations, showing that the adversarial examples weren't simply artifacts of model architecture but were exploiting blind spots in the learned

Full list of author information is available at the end of the article

boundaries of the models themselves, boundaries that tended to be very similar. In 2016 Papernot et al. [3] defined this notion as adversarial example *transferability*. Previous attacks needed access to the gradient of the model to find adversarial examples. Papernot et al. were able to show that transferability could be exploited to make successful black-box attacks against models. This conclusively showed that obfuscated gradients and black-box models were not sufficient to defend a model against adversarial attacks.

These notions of adversarial examples, robustness, and transferability are the foundation of current research. New attacks must be effective through a myriad of defence methods, and new ways of making models robust must be effective against multiple attack models. Research is currently in an arms race, with new defenses and attacks that penetrate those defenses and new defenses to defeat those attacks being published multiple times per year. As machine learning models become integrated into the technology we depend on, providing defensible models becomes crucial. As we will show, adversarial examples have been found that are effective against many of the systems we rely on, across multiple domains.

2 A Conceptual Understanding of Adversarial Examples

Adversarial examples are often viewed as things that are created, but it is perhaps more accurate to view them as things that are found. We can define adversarial examples as points in the input space that are arbitrarily close to a real example yet have a different classification. Madry et al. [4] define the problem of finding adversarial examples as a saddle point problem - the defender performs *outer minimization* to find model parameters that limit the achievable loss and the attacker performs *inner maximization* to attempt to maximize the loss within defined ℓ bounds.

Szegedy et al. [1] theorized that the set of adversarial examples is low-probability (which explains why it doesn't occur naturally with any frequency) yet is a dense subset of the input space and can be found for almost every real example. This gives us a context to talk about defensibility of models in terms of robustness - in a similar method to how cryptographic methods are shown to be computationally secure with equivalence to a problem [5], we can talk about machine learning models as resistant based on the probability of an attacker finding an adversarial example. Additionally, through this optimization lens we can show a model's resistance against attacks in general instead of in average.

Exploiting the linearity of machine learning networks has led to the discovery of *universal perturbations* [6], which are permutation vectors that cause examples to move out of all proper decision boundaries, leading to misclassifications in the majority of examples they are applied to. Surprisingly, these universal perturbations are also quite transferable. It appears that models tend to learn the same boundaries that fail to generalize for improbable examples, allowing the existence of such universal perturbations.

3 Attack Methods

Taxonomy

Attempts to formalize a language concerning attacks on ML models predate the discovery of adversarial attacks. Barreno et al. [7] proposed a taxonomic system that spans several axes, two of note being: An *Influence* axis where attacks can either directly effect the training examples (causative) or simply exploit pre-existing misclassifications without altering training (exploratory), and a *Specificity* axis where attacks either focus on a particular instance (targeted) or attempt to encompass a wide range of instances (indiscriminate). At the time the language was geared toward spam filtering, but has proven useful in the realm of adversarial attacks.

Additionally, the distinction between 'white-box' and 'black-box' attacks has become commonplace, the former representing a model where the attacker has access to the weights and loss function, the latter being a model to which the attacker has only the output.

Linearity and the FGSM

The Fast Gradient Sign Method works by exploiting the linearity present in a ML model. Even in cases where models are decidedly non-linear, this assumption has been proven to work well.

> "We hypothesize that neural networks are too linear to resist linear adversarial perturbation." [2]

By assuming a linear error function, and with access to the model weights, the gradient is computed and a peturbation vector is introduced with the intention of moving the label a maximal amount from the correct prediction. This vector's magnitude is typically bounded by a small value, ϵ that can be parameterized, and has been shown to be effective at values small enough that the error completely disappears when the image is converted to an 8bit encoding.

At it's inception, FGSM was theorized as an indiscriminate method, however by computing the gradient to maximize a specific target, it can also be used as a targeted attack.

Developments

In the relatively short time since the discovery of FGSM, several new attack methods have been developed. Papernot et Al. [8] were able to use their discoveries about transferability to create a black-box attack by training their own model on the label outputs of the black-box network in question. By performing FGSM on this new (white-box) network, the property of transference meant their adversarial examples had a high chance of being misclassified in the original network. Additionally, it was found that iteratively performing FSGM with smaller values produced better results than a single pass of the algorithm. [9]

Huang et al.[10] shortly after proposed a method that, rather than assuming linearity of the cost function, assumed linearity of a simple model, which proved to be more effective than the original FGSM method.

Papernot et al. returned again in 2016 with a new method to compute adversarial attacks with two significant changes to the approach. Firstly, they generated their examples using only the forward gradient, as opposed to calculating the full back-propagation required by the previous methods. Additionally, this method rather than perturbing every value (ie. pixel) by a relatively small amount, instead perturbs a relatively smaller number of pixels by a somewhat larger amount.[11]

Most recently, Carlini and Wagner [12] incorporated several new techniques which not only improved the performance of their adversarial attacks, but provided evidence that defensive distillation (discussed later) is not an effective method for defending against these new kinds of attacks. Of particular note is a monte-carlo style optimization technique that employs several models performing synchronous gradient descents over a region to avoid local minima.

Adversarial examples have been shown to be successfully created even with only one pixel being modified [13], showing that attacks can be efficiently performed even under strict l_0 norm restrictions.

4 Defence Methods

Defence mechanisms usually fall into one of several categories:

- Adversarial Training: Perturbing data during training to make a model robust against adversarial attacks.
- Robust Architectures: Specifically designing architectures and training procedures to be resistant to adversarial attacks.
- Denoising: Modifying the data during classification with the intention of reducing the effectiveness of an adversarial attack.
- Detection: Being able to recognize adversarial inputs.

One of the earliest defence mechanisms was built on the concept of network distillation [14]. Papernot et al. [15] outlined a method for training a robust network using this method. A large and performance-oriented network is created, and training examples are classified using it. A subsequent network is trained on the same examples using the probability spread of the previous model's predictions instead of the one-hot original label. The second model is able to keep a lot of the performance of the original model while being significantly more lightweight and, more importantly, generalizing much better. The distillation step reduces the gradients of the network, resulting in a smoother model where larger perturbations are required to cause a classification change. This significantly reduces the capability of attackers to find adversarial examples.

Guo et al. [16] used input augmentation to increase the security of networks. Using a network that implements a random image transformation layer (using operations such as bit-depth reduction, jpeg compression [17], total variance minimization [18], or image quilting [19]) they were able to significantly reduce the ability to find adversarial examples, eliminating 90% of black-box attacks in testing. Ross and Doshi-Velez [12] found similar results using input gradient regularization, using a technique of 'double back-propagation' to reduce the ability of small changes to the input to cause large changes in classification. Similarly, Prakash et al. [20] introduced a defensive model for CNNs they deemed *pixel deflection*, where pixels are randomly replaced with other pixels from the local region in a manner that introduces random noise while preserving the natural image properties. All three of these techniques force attackers to make significant changes to the input to force a misclassification. Ross and Doshi-Velez used a human testing panel and found that a number of examples that would fool their network had been modified significantly enough to be detected by the naked eye.

Detection methods are another effective option for stymieing attacks. The perturbations that cause a misclassification are theorized to have an extremely low probability of naturally occurring and many detection methods use this feature. Feinman et al. [21] based their defense on Tanay and Griffin's analysis of adversarial attacks as lying near class boundaries that are close to the edges of data sub-manifolds [22]. Theorizing that adversarial samples are separate from the true data manifold, they used "artifacts" - density estimates and Bayesian uncertainty estimate measures -

to train a classifier that could detect adversarial examples out of a pool of adversarial, noisy, and clean examples with an ROC-AUC of 92.6%. Li and Li [23] were able to achieve similar results with a cascade classifier that measured convolutional filter output statistics in early convolutional layers, detecting more than 85% of adversarial examples. Xu et al. [24] used a technique they deemed *feature squeezing* to reduce the effects of adversarial perturbations. Through either bit depth reduction (including shifting RGB images to grayscale) or image smoothing (through local or non-local blurring) they make predictions on both the unadulterated image and the modified image, classifying it as adversarial if the predictions are significantly different. With a joint detection method utilizing both of these they were able to achieve as high as a 99.44% detection rate while also keeping a remarkably low false positive rate. These detection methods can provide an effective first line of defence, allowing models to reject or take steps to mitigate examples that are detected as likely adversarial.

Hosseini et al. [25] formed a training method that attempts to block attacks by limiting transferability. They train a network on regular and adversarial examples, augmenting the data with a NULL label that signifies adversarial examples. This method enables the system to learn to detect the signature perturbations of adversarial attacks and classify examples as likely adversarial, giving the model a built-in resistance to attacks. This was successful at limiting transferability, extremely handicapping black-box attacks.

5 Domain Specific Attacks

5.1 Audio Recognition Systems

Some novel work has been done in the field of audio recognition systems. These systems are commonly encountered in automated phone systems, handheld devices, and other technology we commonly interact with. Some of the models that drive these systems use non-differentiable layers in their implementation, but adversarial attacks have still be shown to be possible. Previous attacks against these systems such as Backdoor [26] and DolphinAttack [27] have been successful but use inaudible frequencies to manipulate the systems. The adversarial attack should not exploit these vulnerabilities or it risks being confounded by simple filters.

Alzantot et al. [28] were successful with black-box attacks on automatic speech recognition systems. They used an evolutionary algorithm that adds random noise to an audio sample to perform the targeted attacks, successfully changing the classification of 87% of examples. Unlike the standard of attempting to find examples that humans can't notice, they instead tried to find examples that humans would still correctly classify, a goal that 89% of their successful adversarial examples achieved. Additionally, this method was not successful when played over a speaker, a shortcoming the authors planned to address in following research.

Carlini and Wagner [29] performed white-box attacks on a speech-to-text system using the differentiable CTC-loss measure. They were able to create adversarial audio files that were nearly indistinguishable from the original audio and were transcribed as whatever recognizable text sequence they desired. These attacks were effective when both used directly as input or when played by a speaker and recorder through a microphone, meaning they would be effective at attacking real-world devices. Additionally, if the distortion used was significant enough these examples survive mp3 compression. They left an open question about if speech-to-text systems have the same transferable property as image recognition systems, which would mean black-box attacks against systems like Apple's Siri or Google's Alexa are possible. One further interesting note is that they were able to effectively hide speech - generating distortion that made speech-to-text systems fail to recognize the presence of any speech whatsoever. This one has some optimistic implications toward audio privacy filters.

Kreuk et al. [30] focused on attacking speaker verification systems. These systems use models that have been trained to recognize a speaker so as to only permit access to a system to someone recognized as the authorized user. In both white-box and black-box attacks they were able to artificially create a high false-positive rate, signifying that a random speaker was verified as the target speaker. These results should be a dire warning to companies thinking of using deep learning speaker verification as a security measure.

5.2 Text Attacks

Text presents a novel field because the lower input space and nature of language makes small changes much more difficult. Despite this, Gao et al. [31] introduced an algorithm *DeepWordBug* designed as a black-box attacker against text classification systems. Though the changes are regularly of the form of creating typos and are noticeable to an astute reader, they were able to force misclassifications on both spam filtering systems and movie review sentiment analysis systems with moderate success. Ebrahami et al. [32] used a similar method to similar results, changing certain letters with others in the direction that caused

the strongest misclassification.

Alongside this is the work of Liang et al. [33], who also insert letters to fool deep networks. Their system uses the gradient of the network to identify the most crucial elements of the phrase to classification and subsequently alter those elements. Working on either a word-level or a phrase-level, the system attempts to cause a misclassification through removal of important parts of speech, insertion of mostly semantically empty phrases, and modification of existing parts of speech. This approach was shown to be effective against both movie review and product review sentiment analysis systems. Samanta and Mehta [34] achieved similar results on sentiment analysis systems by identifying the most probable words that contributed to the classification and replacing them with words with significantly different sentiment (for example, replacing the phrase "this was a real winner" with "this was a real whiner").

5.3 3d Printed Adversarial Examples

Kurakin et al. [9] showed that adversarial examples could be created that could be printed and photographed with a smartphone and still cause a misclassification, beginning research on adversarial examples that force a misclassification at different angles, lighting situations, and other alterations that change the pixel-to-vector mapping. Convolutional neural networks in particular are vulnerable to these types of attacks.

One of the most fascinating applications of adversarial examples was the groundbreaking work of Athalye et al. [35]. Building on the printed adversial examples studied by Kurakin et al. [9], they developed a 3d object that forced misclassification. Starting with an adversarial example that had been specifically created to be resilient to noise, transformation, and distortion, they printed a textured 3d model of a turtle and were able to achieve the intended adversarial misclassification of 'rifle' on 82% of images taken at different angles and lighting situations. This suggests that adversarial attacks can be effective even against dynamic real world systems such as self-driving vehicles or intrusion detection systems.

5.4 Reinforcement Learning

While most research of adversarial attacks has focused specifically on deep networks, the strategies have been found to work in the realm of reinforcement learning (RL) as well.

The transferability property of overcoming black-box attacks by training a similar model was shown to apply equally well to RL models by Xiao et al. [36].

Pinto et al. [37] meanwhile harnessed the concept of adversarial training to improve the robustness of their RL models: by training against an agent designed to minimize the model's output, it became generally more robust, though their adversarial examples were generated using domain-specific knowledge that was unlikely to generalize.

Because RL models its environment, input, and agent separately, it presents several attack vectors in a way that doesn't necessarily have an analog in image classification with deep learning networks. Mandlekar et al. [38] were able to focus specifically on perturbing the state of the model, and were able to show the resultant model was not only robust against attacks, but showed better performance overall.

In 2017, Pattanaik et al.[39] were able to improve performance over a gradient descent method by specifically attacking the policy directly: essentially 'tricking' an agent into believing themselves to be in a different state.

6 Conclusion

Adversarial examples give us a unique look at how ML models fail to generalize as well as highlight some of the ways that models that can be exploited by this lack of generalization. As the use of these models as a service becomes commonplace in applications with as dangerous possibilities as autonomous vehicles [40] we need to ensure security of these models. Research has sustained a frenzied arms race, but we believe that until we can reach a greater understanding about how models learn decision boundaries and how we can make them generalize over even low-probability points in the example space that simply finding new attacks and new defenses will not resolve the underlying insecurity. As these systems are put in place in situations with potentially devastating consequences, we believe developing a deeper understanding of how to build robust models is crucial.

Author details

References

1. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
2. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
3. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 (2016)
4. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
5. Stinson, D.: Cryptography: Theory and practice,second edition, 2nd edn. CRC/C&H (2002). Chap. 2
6. Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1765–1773 (2017)
7. Barreno, M., Nelson, B., Joseph, A.D., Tygar, J.D.: The security of machine learning. Machine Learning **81**(2), 121–148 (2010)

8. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519 (2017). ACM
9. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
10. Huang, R., Xu, B., Schuurmans, D., Szepesvári, C.: Learning with a strong adversary. arXiv preprint arXiv:1511.03034 (2015)
11. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 372–387 (2016). IEEE
12. Ross, A.S., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
13. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation (2019)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
15. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597 (2016). IEEE
16. Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)
17. Dziugaite, G.K., Ghahramani, Z., Roy, D.M.: A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853 (2016)
18. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena **60**(1-4), 259–268 (1992)
19. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 341–346 (2001). ACM
20. Prakash, A., Moran, N., Garber, S., DiLillo, A., Storer, J.: Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8571–8580 (2018)
21. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 (2017)
22. Tanay, T., Griffin, L.: A boundary tilting persepective on the phenomenon of adversarial examples. arXiv preprint arXiv:1608.07690 (2016)
23. Li, X., Li, F.: Adversarial examples detection in deep networks with convolutional filter statistics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5764–5772 (2017)
24. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155 (2017)
25. Hosseini, H., Chen, Y., Kannan, S., Zhang, B., Poovendran, R.: Blocking transferability of adversarial examples in black-box learning systems. arXiv preprint arXiv:1703.04318 (2017)
26. Roy, N., Hassanieh, H., Roy Choudhury, R.: Backdoor: Making microphones hear inaudible sounds. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, pp. 2–14 (2017). ACM
27. Song, L., Mittal, P.: Inaudible voice commands. arXiv preprint arXiv:1708.07238 (2017)
28. Alzantot, M., Balaji, B., Srivastava, M.: Did you hear that? adversarial examples against automatic speech recognition. arXiv preprint arXiv:1801.00554 (2018)
29. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 1–7 (2018). IEEE
30. Kreuk, F., Adi, Y., Cisse, M., Keshet, J.: Fooling end-to-end speaker verification with adversarial examples. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1962–1966 (2018). IEEE
31. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 50–56 (2018). IEEE
32. Ebrahimi, J., Rao, A., Lowd, D., Dou, D.: Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751 (2017)
33. Liang, B., Li, H., Su, M., Bian, P., Li, X., Shi, W.: Deep text classification can be fooled. arXiv preprint arXiv:1704.08006 (2017)
34. Samanta, S., Mehta, S.: Towards crafting text adversarial samples. arXiv preprint arXiv:1707.02812 (2017)
35. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397 (2017)
36. Xiao, C., Pan, X., He, W., Li, B., Peng, J., Sun, M., Yi, J., Liu, M., Song., D.: Characterizing Attacks on Deep Reinforcement Learning (2019). https://openreview.net/forum?id=ryewE3R5YX
37. Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.: Robust adversarial reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 2817–2826 (2017). JMLR. org
38. Mandlekar, A., Zhu, Y., Garg, A., Fei-Fei, L., Savarese, S.: Adversarially robust policy learning: Active construction of physically-plausible perturbations. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3932–3939 (2017). IEEE
39. Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., Chowdhary, G.: Robust deep reinforcement learning with adversarial attacks. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2040–2042 (2018). International Foundation for Autonomous Agents and Multiagent Systems
40. Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R., et al.: An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716 (2015)