Adversarial Variance Attacks: Deeper Insights into Adversarial Machine Learning through the Eyes of Bias-Variance Impact

Hossein Aboutalebi[1], Mohammad Javad Shafiee[1]
Michelle Karg[2], Christian Scharfenberger[2]
Alexander Wong[1]
[1]Waterloo AI Institute, University of Waterloo, Waterloo, Ontario, Canada
[2]ADC Automotive Distance Control Systems GmbH, Continental, Germany
[1]{haboutal, mjshafiee, a28wong}@uwaterloo.ca
[2]{michelle.karg, christian.scharfenberger}@continental-corporation.com

Abstract

Prior studies have unveiled the vulnerability of the deep neural networks in the context of adversarial machine learning, leading to great recent attention into this area. One interesting question that has yet to be fully explored is the bias-variance relationship of adversarial machine learning, which can potentially provide deeper insights into this behaviour. In this study, we investigate the effect of adversarial machine learning on the bias and variance of a trained deep neural network and analyze how adversarial perturbations can affect the generalization of a network. We derive the bias-variance trade-off for both classification and regression applications based on two main loss functions: (i) mean squared error (MSE), and (ii) cross-entropy. Furthermore, we perform quantitative analysis with both simulated and real data to empirically evaluate consistency with the derived bias-variance tradeoffs. Moreover, given these new theoretical findings, we introduce a new adversarial attack called Adversarial Variance Attack (AVA) which specifically targets the variance of the network and causes higher variance in system response compared with other attacks (e.g., PGD).

1. Introduction

Despite of the impressive achievements of deep learning over the past decade in different fields such as computer vision [7, 10, 11, 17], machine translation [21, 22], and medicine [3, 4], their vulnerability against adversarial machine learning brings different concerns regarding their robustness.

A perturbation ϵ in a specific direction to the input causes the model to incorrectly classify the input sample which can be preformed in both classification [13, 19] or regression problems [1, 20]. The perturbation ϵ should be imperceptible by a human eye and as such, the norm of ϵ is bounded when a new perturbation is generated. Szegedy *et al.* introduced this drawback for deep neural networks in their seminal paper [19]. They observed that the state-of-the-art deep neural networks act poorly with high confidence when an imperceptible non-random perturbation is added to the input image. They attributed this poor behaviour to the potential blind spots in the training of deep neural networks. Goodfellow *et al.* [6] argued this poor performance of deep neural networks on adversarial examples is due to their linear behavior in high-dimensional spaces. Since then, there have been several studies introducing different approaches to generate adversarial perturbation and fool the deep neural networks [6, 8, 13, 15]. However, Madry *et al.* [12] proposed a multi-step attack called the projected gradient descent (PGD) algorithm which generalizes the prior first-order adversarial machine learning algorithms and is able to produce adversarial examples that are harder to learn and to defeat.

Despite a rich literature developed in the field of adversarial machine learning, model generalization is an important drawbacks of these techniques. Bias and Variance are one of the long-standing and well-known procedure to analyze the generalization and reliability of machine learning models. The seminal work by Geman *et al.* [5] showed that while a model's variance increases, the model's bias decreases monotonically with the increase in the model complexity. They derived a well-formed decomposition of the bias and variance of the loss function for the regression learning task.

Although bias-variance trade-off has been used to justify some aspects of deep neural networks in previous studies, to the best of authors' knowledge, the theoretical analysis around the impact of the adversarial machine learning algorithms on the bias and variance of a deep neural network has not been well explored. In this paper, we aim to study the effect of adversarial machine learning on the bias and variance of a deep neural network. Here, a new decomposition of the loss function in a deep neural network is derived in terms of its bias and variance for both the regression and classification tasks when the input sample is perturbed by an adversarial machine learning algorithm. The proposed theorems illustrate that an adversarial machine learning algorithm can be designed in such a way that attacks a model by only changing its behaviour in terms of either bias or variance. As such, the proposed theorems suggest that it is possible to design more powerful adversarial machine learning algorithms which are much harder to be detected and resolved. One interesting idea would be to design adversarial machine learning methods which only change the model variance and only make the model unstable in specific cases and situations. As a result, they might be very hard to identify as there is not any significant change in the model's bias which make them more disastrous. In this paper, we will propose one such attack called AVA and illustrate its performance on different datasets. Using our theoretical derivations, we propose an algorithm which can manipulate the variance of the network. Later in the experiment section, we illustrate that the proposed AVA attack can increase the variance more significantly than other adversarial attacks including CW, PGD and FGSM. Also, we show that increasing the size of the network cannot necessarily make it less susceptible against this attack.

2. Methodology

In this section we illustrate the effect of adversarial machine learning algorithms on the model's bias and variance and derive how the perturbation can change the behaviour of a model by studying its bias and variance. Here we aim to study the bias-variance trade-off in deep neural networks based on two well-known loss functions, MSE loss and cross-entropy loss. The detailed version of the theorem is in the Appendix.

2.1. Notation

The detailed information on the notation is in the Appendix.

2.2. Case I: Regression with MSE Loss

Assume the goal is to estimate the target function $f : \mathcal{X} \to \mathcal{Y}$. Each element $x \in \mathcal{X}$ has dimension $|x| = d$. Given the training data, $\mathcal{D} = \big\{(x_1, y_1), ..., (x_m, y_m)\big\}$, a learner produces a prediction model $\hat{f}(x)$. As such, the configuration of the parameters in $\hat{f}(x)$ is dependent on the training data $\mathcal{D}$. Let us also assume the training data $\mathcal{D}$ is accompanied with a natural noise γ such that:

$$y_i = f(x_i) + \gamma \tag{1}$$

where $1 \leq i \leq m$ with m total number of data samples in $\mathcal{D}$, and γ is a random variable where $\mathbb{E}[\gamma] = 0$, and $\mathbb{E}[\gamma^2] = \sigma_\gamma^2$. It is worth to note that, we keep this assumption mainly for the regression task and we will drop it for the classification problems with cross-entropy loss function for simplicity. Geman *et al.* [5] decomposed a MSE loss function in terms of its bias and variance of a prediction model by Theorem 1.

Theorem 1 *For a prediction model $\hat{f}(x)$ trained on the training data $\mathcal{D}$ to estimate the target function $f(x)$ with MSE loss function, the bias variance trade-off is [5]:*

$$\mathbb{E}_{x,\mathcal{D},\gamma}\Big[\big(y - \hat{f}(x)\big)^2\Big] = \mathbb{E}_{x,\mathcal{D}}\Big[\Big(\mathbb{E}_{\mathcal{D}}\big[\hat{f}(x)\big] - f(x)\Big)^2\Big] + \\ \mathbb{E}_{x,\mathcal{D}}\Big[\Big(\hat{f}(x) - \mathbb{E}_{\mathcal{D}}\big[\hat{f}(x)\big]\Big)^2\Big] + \sigma_\gamma^2 = Bias[\hat{f}] + Var[\hat{f}] + Var[\gamma]. \tag{2}$$

The $Var[\gamma]$ is the intrinsic noise of the system. Given (2), it is possible to break down and decouple the effect of different factors on model performance based on the bias, variance and intrinsic noise in the model. However, (2) does not take the effect of adversarial perturbation into account. The perturbation $\beta(x)$ added to each data sample x during the test time aims to increase the loss value of the model. It is assumed that $f(x) = f(x + \beta(x))$, this assumption is to make sure the added perturbation magnitude is reasonable and follows the imperceptibility of the adversarial perturbation.

This perturbation can have a great impact on the final loss which is significantly different from (2). Following, we propose a new theorem to account for the adversarial perturbation in deriving bias and variance of a model.

Theorem 2 *Assume $\bar{f}(x) = \mathbb{E}_{\mathcal{D}}[\hat{f}(x)]$ and the target function is $f(x)$. The bias-variance trade-off for MSE loss function with a prediction model $\hat{f}(x)$ trained on dataset $\mathcal{D}$ with noise γ in the presence of adversarial perturbation $\beta(x)$ via the adversarial algorithm is:*

$$\mathbb{E}_{x,\mathcal{D},\gamma}\Big[\big(y - \hat{f}(x + \beta(x))\big)^2\Big] \approx \\ \mathbb{E}_{x,\mathcal{D}}[(f(x) - \bar{f}(x) - c_x)^2] + Var[\gamma] + Var[\hat{f}] + \mathbb{E}_{x,\mathcal{D}}[c'_x] \tag{3}$$

$$\textit{where,} \quad c_x = \nabla\bar{f}(x)^T\beta(x)$$

$$\textit{and,} \;\; c'_x = 2\Big(\hat{f}(x) - \bar{f}(x)\Big)\Big(\Big(\nabla\hat{f}(x) - \nabla\bar{f}(x)\Big)^T\beta(x)\Big) \tag{4}$$

Proof: The proof can be found in the supplementary material.

2.3. Case II: Classification with cross-entropy Loss

The notion of bias and variance can be analyzed for the classification models trained with cross-entropy loss as well. To this end, followed by the work done in [16, 23] let c be the number of classes for classification and $\hat{\pi}_{\mathcal{D}}(x) \in [0, 1]^c$ be the output of a neural network trained on the training set $\mathcal{D}$. This function measures the confidence values over classes.

Let $\pi(x) \in [0,1]^c$ be a one-hot vector encoding ground truth label that we wish to estimate via $\hat{\pi}$. Then cross-entropy loss can be formulated as:

$$L(\pi, \hat{\pi}) = -\mathbb{E}_{x,\mathcal{D}}\Big[\sum_{i=1}^{c}\big(\pi_i(x)\log\hat{\pi}_i(x)\big)\Big] \tag{5}$$

where $\pi_i(x)$ refers to ith component of the output vector $\pi(x)$. As explained in [23], the loss function in (5) can be decomposed:

$$L(\pi, \hat{\pi}) = D_{KL}\big(\pi(x)||\bar{\pi}(x)\big) + \mathbb{E}_{x,\mathcal{D}}\left[D_{KL}\left(\bar{\pi}(x)||\hat{\pi}(x)\right)\right] \tag{6}$$

where $\bar{\pi}(x) \propto \exp\Big(\mathbb{E}_{\mathcal{D}}\left[\log(\hat{\pi}(x))\right]\Big)$. $\bar{\pi}(x)$, as described in [23], is the average of log probability after normalization.

Theorem 3 *Assume for input x, the ground truth class is t_x. For a cross-entropy loss function, the bias-variance tradeoff of a prediction model $\hat{\pi}(x)$ with training data $\mathcal{D}$ for a target function $\pi(x)$ in the presence of adversarial algorithm injecting perturbation $\beta(x)$ to the system is:*

$$\begin{aligned} L(\pi, \hat{\pi}) = &\mathbb{E}_{x,\mathcal{D}}\left[D_{KL}(\pi(x)||\bar{\pi}(x)) + D_{KL}(\bar{\pi}(x)||\hat{\pi}(x))\right] + \\ &\mathbb{E}_x\left[c_x\right] + \mathbb{E}_{x,\mathcal{D}}\left[c'_x\right] \end{aligned} \tag{7}$$

where,

$$\begin{aligned} c_x &= -\mathbb{E}_x\left[\left(\nabla_x \log\bar{\pi}_{t_x}(x)\right)^T\beta(x)\right] \\ c'_x &= -\mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\left(\nabla_x\bar{\pi}_i(x)\log\frac{\hat{\pi}_i(x)}{\bar{\pi}_i(x)}\right)^T\beta(x)\right] \end{aligned} \tag{8}$$

Proof: The proof can be found in the supplementary material.

This derivation is aligned with finding in [16, 23], where the bias variance decomposition for cross-entropy loss function is in the form of KL-Divergence. The proposed theorem leads to the following corollaries,

Corollary I: The maximum expected increase in the bias of a deep neural network trained with cross-entropy loss is when the adversarial perturbation is in the direction of c_x in (8).

Corollary II: The maximum expected increase in the variance of a deep neural network trained with cross-entropy loss is when the adversarial perturbation is in the direction of c'_x in (8).

2.4. Adversarial Variance Attack (AVA)

Given the derivations provided by Theorem (3), here we propose a new attack which specifically targets the variance of the machine learning model in classification tasks so-called Adversarial Variance Attack (AVA). The details of the proposed AVA method is described in Algorithm 1. AVA attack starts with a set of prediction model networks with the same architectures that have been trained with different seeds on different training data. Then AVA creates an average model $\bar{\pi}$ which is the mean of the output of the predictions by all models. In the next step, AVA takes the gradient from the loss function defined as a KL Divergence between $\bar{\pi}$ and the given network π_k. Finally, this gradient is projected and added with step size ϵ to the input of the network π_k. As AVA requires a set of prediction networks, for training these networks, we used the same technique used in [23] to split the training set among them and train each of the network with the corresponding training set (more details on this part is in the experiment section). In our experiments, we found out that using even two networks for AVA is enough for achieving good results with high variance.

Algorithm 1: AVA Attack for classification

```
Data: $\Big\{\big(x, y(x)\big)|x \in D\Big\}$ with $c$ distinct classes
Result: $\hat{x}$ Perturbed image $x$ for model $\hat{\pi}_k$. Input:
        $\hat{\pi}_1, ..., \hat{\pi}_k, ..., \hat{\pi}_n$, The prediction models
        trained on different training set.
       $\epsilon$, The magnitude of perturbation.
       $x$, The input image.
       $\gamma$, maximum allowed perturbation.
Begin
  $\bar{\pi} = \frac{1}{n}\Sigma_{i=1}^{n}\hat{\pi}_i$
  $L = D_{KL}\big(\bar{\pi}(x)||\pi_k(x)\big)$
  $\hat{x} = \Pi_{x+\gamma}\big(x + \epsilon\,\nabla_x L\big)$
    Return $\hat{x}$
End
```

As it will be seen in the experiments, unlike previously proposed attacks in adversarial machine learning which focuses mainly on the bias of the model in the predictions, the variance attack manipulates the variance of the prediction model and makes it behave unstable when it is trained across different training datasets. In this regard, as it increases the variance of the model, detection of this attack becomes more cumbersome as the model trained on one training set may have a good performance while the same model trained on a different datasets may have a poor performance against this attack. This phenomenon makes the resilience against this attack harder as multiple replica of same model might be needed for training and testing to obtain a reliable robustness against this attack.

3. Experimental Results & Discussion

In this section, we examine the proposed theorems experimentally and illustrate how a deep neural network behaves facing an adversarial machine learning algorithm based on its bias and variance. We also analyze the effectiveness of the proposed variance attack. To this end, we mainly

Table 1: Evaluation results of different attack on adversarially trained WRN-28 against CIFAR-10 dataset. While other attacks such as CW and PGD adversarial attacks could fool the network more and dropped the model accuracy to higher degree, results show that the proposed AVA attack can change the behaviour of the targeted deep neural network more significantly and as such resulted to a higher variance.

Attack	**Accuracy**	**Variance**
FGSM	41.86	8.21
PGD20	36.54	7.30
PGD40	**36.50**	7.09
CW20	37.46	9.51
CW40	37.52	8.97
AVA	69.11	**29.50**

Table 2: Evaluation results of competing attacks on adversarially trained WRN-28 based on SVHN dataset. The results further validate the reported result in Table 1 and show the effectiveness of the proposed AVA attack in changing the model's behaviour.

Attack	**Accuracy**	**Variance**
FGSM	76.72	32.03
PGD20	57.33	18.40
PGD40	58.63	11.53
CW20	**57.08**	23.06
CW40	58.45	12.73
AVA	62.85	**47.86**

Table 3: Evaluation results of competing attacks on adversarially trained WRN-28 based on CIFAR-100 dataset.

Attack	**Accuracy**	**Variance**
FGSM	18.71	4.51
PGD20	15.83	2.63
PGD40	**15.81**	2.49
CW20	16.4	3.03
CW40	16.45	2.92
AVA	33.06	**23.42**

evaluate the theorem on Wide ResNet (WRN-28-10) [24] architecture on both real datasets including CIFAR-10 and CIFAR-100 [9] and simulated data for both classification and regression tasks. We also study the MobileNetV2 [18] architecture which is in the supplementary. Most of experiments are included in the appendix due to the page limit.

3.1. Results

In this section, we study the effect of adversarial machine learning algorithms and the proposed AVA attack on real datasets. To this end, we take advantage of adversarial training techniques to improve the robustness of the examined models. As mentioned earlier, the main neural network architecture used in our study is Wide ResNet architecture (WRN-28) and we included the results for MobileNetV2 in the supplementary. The proposed method and competing algorithms are evaluated via three datasets including CIFAR-10 [9], CIFAR-100 [9] and SVHN [14] datasets. To measure the bias and variance, we followed the same approach described in [23]. Two models trained on independent subsets of the training data which are obtained via splitting the training set in half are constructed and the bias and variance results are reported as an average over 3 of such random splits.

To further analyze different attacks and evaluate how different adversarial attacks affect the model's variance, the proposed AVA algorithm is compared with CW [2], FGSM [6], and PGD [12] adversarial attacks on WRN-28 network architecture. This network is adversarially trained on CIFAR-10 dataset using Madry PGD-based adversarial training [12]. As seen in Table 1, although other adversarial attacks are more successful in decreasing the accuracy of the deep neural network model, their variance is lower than what AVA algorithm can achieve. Table 1 shows that for CW and PGD algorithm, as we increase the number of steps in these attacks, both the variance and the accuracy of the network decreases.

Table 2 depicts a similar experiments performed on SVHN dataset. Here, the variance is higher for all attacks compared with CIFAR-10. Again, we can observe a similar pattern as reported in Table 1. As the number of steps for the attack increases, the variance decreases. Although AVA accuracy drop is lower than CW and PGD, its variance increase is the highest among all the other attacks. Similar experiments for CIFAR-100 dataset is reported in Table 3. In Table 3, the accuracy of the model has dropped more significantly compared with other two previous dataset. We once again observe that AVA is more successful in increasing the variance while other attack models drop the accuracy higher than AVA.

The reported results of the proposed AVA attack further confirms Theorem 3 and it shows that AVA attack is effective in increasing the variance of the network more significantly than other well-know adversarial attacks.

4. Conclusion

In this paper we studied the effect of adversarial machine learning on a model's bias and variance. We proposed a new set of theorems which decompose the effect of adversarial perturbations on machine learning models trained with two well-known loss functions. The new derivations showed when to expect the maximum increase in the bias and variance of the model facing adversarial machine leanings. While the theorems verify the previous findings in this field which the model is vulnerable in the opposite direction of gradient of loss function, the proposed theorems can quantify what is the best direction for adversarial perturbation to maximize the effect. The proposed theorems can help us to better understand the effect of adversarial machine learning algorithms in the field of deep neural networks.

References

[1] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 4

[3] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie NC Shih, John Tomaszewski, Fabio A González, and Anant Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific reports*, 7:46450, 2017. 1

[4] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018. 1

[5] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992. 1, 2

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[8] Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvári. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*, 2015. 1

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[11] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010. 1

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 4

[13] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1

[14] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4

[15] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 1

[16] D. Pfau. A generalized bias-variance decomposition for bregman divergences. 2013. 2, 3

[17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 4

[19] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[20] Liang Tong, Sixie Yu, Scott Alfeld, and Yevgeniy Vorobeychik. Adversarial regression with multiple learners. *arXiv preprint arXiv:1806.02256*, 2018. 1

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1

[22] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 1

[23] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. *arXiv preprint arXiv:2002.11328*, 2020. 2, 3, 4

[24] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 4