A new measure for overfitting and its implications for backdooring of deep learning

Kathrin Grosse[†][*], Taesung Lee[‡], Youngja Park[‡], Michael Backes[†], Ian Molloy[‡]
[†] CISPA Helmholtz Center for Information Security
[‡] IBM T. J. Watson Research Center
{kathrin.grosse, backes}@cispa.saarland,
taesung.lee@ibm.com, {young_park, molloyim}@us.ibm.com

Abstract

Overfitting describes the phenomenon that a machine learning model fits the given data instead of learning the underlying distribution. Existing approaches are computationally expensive, require large amounts of labeled data, consider overfitting global phenomenon, and often compute a single measurement. Instead, we propose a local measurement around a small number of unlabeled test points to obtain features of overfitting. Our extensive evaluation shows that the measure can reflect the model's different fit of training and test data, identify changes of the fit during training, and even suggest different fit among classes. We further apply our method to verify if backdoors rely on overfitting, a common claim in security of deep learning. Instead, we find that backdoors rely on underfitting. Our findings also provide evidence that even unbackdoored neural networks contain patterns similar to backdoors that are reliably classified as one class.

1 Introduction

Identifying and avoiding overfitting are one of the most important factors in training a successful machine learning (ML) model. Failure to do so might result in an illusion of high accuracy, yet the model performs poorly in deployment. One basic prevention method is using a part of the data to train the model and evaluate it on the remaining, unseen test[2] data, thereby measuring the overfit on the training data (Bishop, 2007).

However, there are several limitations to the existing approaches using the test dataset. First, recent studies show that we might not have adequate test dataset for this. Recht et al. (2018, 2019) claim that the datasets can be too biased overall, and the classifier might overfit unseen test data. For example, if both training and test datasets for image classification contain only green frogs, overfitting to the color feature might result in an illusional high accuracy. Another challenge is a dataset shift having training and test data from slightly different distributions, which is often the case for a real world deployment scenario (Muandet et al., 2013). That is, training a highly accurate classifier on ImageNet might not result in the best classifier for a smartphone camera as the pictures might have very different characteristics such as overall brightness, colors, resolution, and noise pattern.

Moreover, all existing approaches output a single global number, whereas overfitting can actually be a local phenomenon: The model can overfit some portion of data but not others, making overfitting less visible *overall*. Finally, existing approaches require a large amount of labeled test data, which is not always available and hard to create while avoiding all the previously mentioned pitfalls.

[*] Work was done while the first author was interning at IBM T. J. Watson Research Center.

[2] We abuse the term *test dataset* to refer to both test and validation datasets, as we have the same problem with the validation dataset or even if we use the validation dataset.

Preprint. Under review.

Consequently, recent efforts have been made to measure overfitting (Werpachowski et al., 2019) and memorization (Achille et al., 2019). However, these procedures are computationally expensive, or require labeled test dataset. Thus, we propose a novel measure, W^3, that examines a local decision surface around an unlabeled test data sample. Specifically, we compute, for a given data point $\mathbf{x}$, how stable the decision surface is around $\mathbf{x}$. W is a local adaptation of the bias-variance decomposition (Geman et al., 1992; Bishop, 2007) which was recently revisited for deep models (Ba et al., 2020; Yang et al., 2020). However, instead of computing a scalar value, we consider the distribution of measurements that can not only improve the understanding of overfitting, but also enable other applications such as detecting backdoors. Furthermore, unlike the bias-variance decomposition, our measure leverages only small portion of *unlabeled* data.

Our empirical evaluation shows that W covers many cases of overfitting: it reflects the better fit of seen training data as opposed to unseen test data. It reacts to different configurations that influence overfitting, like adversarial training (Miyato et al., 2016) or adversarial initialization (Liu et al., 2019). Our experiments also confirm that overfitting can be local phenomenon, as the W shows significant differences among classes. These results also confirm that our measure requires only a small number of unlabeled data: W is sufficiently stable with as little as 250 *unlabeled* data samples.

Moreover, we evaluate a hypothesis from adversarial machine learning, stating that backdoors are enabled by overfitting (Wang et al., 2019). A backdoor for a ML model is a security threat where the attacker hides a particular pattern in the training data linked to a particular class. This pattern can later be used to evade the trained classifier at test time (Liu et al., 2018; Zhu et al., 2019). We show that backdoors are underfit rather than overfit: Backdoor features are, when present, reliably classified as one class. Furthermore, we find that neural networks in fact contain such stable trigger patterns for any class, supporting the pattern matching classification hypothesis (Krotov and Hopfield, 2016).

2 Related work

There are many ways to measure **overfitting** beyond the typical comparison of test and training loss (Bishop, 2007). Achille et al. (2019) derive a measure for learning complexity, which can be used to compare different datasets and can be further used to measure overfitting. Werpachowski et al. (2019) propose an overfitting measure based on a large batch of adversarial examples and a statistical test. Unlike the existing approaches, our method does not require costly computations, and requires only a small number of unlabeled data points. In general, it could be fruitful to combine our measure with other methods and run the costly, but full-fledged test when W indicates overfitting.

Kohavi et al. (1996) decomposed the expected generalization loss of a classifier into variance, bias, and an irreducible error term, called **bias-variance-trade-off**. The underlying idea is to compare a classifier across different datasets, also called *random design*. Ba et al. (2020) assume instead one data-set, but that the data is noised. This is called *fixed design*, and although we noise the input data, not the labels, this is the setting our motivation stems from. Formerly, the bias-variance-trade-off implied that bias decreases with model complexity, whereas variance increases. Mei and Montanari (2019) and Yang et al. (2020) challenge this view: They find evidence that the variance term is instead u-shaped, and decreases in regimes of higher model complexity. Our measure provides an orthogonal view that we can measure overfitting locally.

Finally we investigate **backdoors** in the context of overfitting. Defending backdoors is an open research problem (Tran et al., 2018; Chen et al., 2019; Wang et al., 2019; Liu et al., 2019), which has shown to lead to an arms-race (Tan and Shokri, 2020). Our detection mechanism wrongly classifies all candidates generated by Wang et al. (2019) as backdoors. However, this wrong attribution implies the existence of benign backdoors in any network, increasing the difficulty of defending such attacks.

3 Local Wobbliness Measure W

We first give a high level intuition of the W-measure, and then formally introduce the measure.

High level intuition. In Figure 1, we depict data points of different classes (in different blue shades) with their sampled areas (gray circles). On the left hand side, we show a wobbly, or overfitted

[3] W is named in honor of Doctor **W**ho, who first recognized wibbly wobbly timy wimy properties.

classifier. On the right hand side, we observe a good, non-overfitted classifier. The picture illustrates W: at an appropriately chosen radius, overfitting becomes evident as the sampled ball around a test point is not consistently classified. We thus quantify overfitting in the input space by measuring the output space in the area around each test point. Note that this measurement is local, and it can vary across the input space.

Definition of the W. An ML classifier $F(\mathbf{x}, \theta)$ trained on i.i.d. data is given. We denote training points as $\mathbf{x}$ and the corresponding label as y. After learning the parameters θ on $(\mathbf{x}, \mathrm{y})$ pairs, the loss L should be minimal. We finally define $\mathbf{x}'$ as a randomly drawn point around $\mathbf{x}$; $\mathbf{x}' \sim \mathcal{N}(\mathbf{x}, \sigma^2)$.

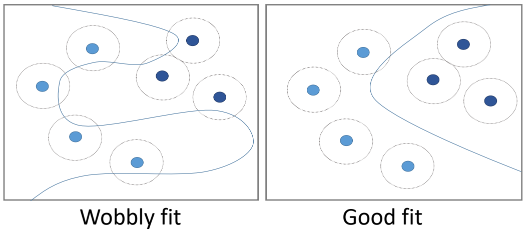


Figure 1: Measuring overfitting with W: We sample points (gray circles) around test data (blue dots) to quantify the wobbliness of the decision function (blue line).

Given this formalization, we use the bias-variance decomposition to divide the loss L into bias and variance as introduced by Kohavi et al. (1996). We start our analysis with the local measurement of the cross-entropy loss function H, the most common loss function for DNN classification, around $\mathbf{x}$ written

$$L(\mathbf{x}, \mathrm{y}) = \mathbb{E}_{\mathbf{x}'} H(\mathrm{y}, F(\mathbf{x}')) = -\mathbb{E}_{\mathbf{x}'} \sum_i \mathrm{y}_i \log F(\mathbf{x}')_i \quad , \tag{1}$$

where y_i and $F(\mathbf{x}')_i$ indicate the respective i-th dimension element, and $\mathbb{E}_{\mathbf{x}'}$ is the mean over data points from the Gaussian distribution $\mathcal{N}(\mathbf{x}, \sigma^2)$. We decompose the loss in bias and variance by adapting (James, 2003) on cross entropy, yielding

$$\begin{aligned} L(\mathbf{x}, \mathrm{y}) &= \mathbb{E}_{\mathbf{x}'} \left[H(\mathrm{y}, F(\mathbf{x}')) - H(F(\mathbf{x}'), \mathbb{E}_{\mathbf{x}'} F(\mathbf{x}')) - H(\mathrm{y}, \mathbb{E}_{\mathbf{x}'} \mathrm{y}) + H(F(\mathbf{x}'), \mathbb{E}_{\mathbf{x}'}(F(\mathbf{x}'))) + H(\mathrm{y}, \mathbb{E}_{\mathbf{x}'} \mathrm{y}) \right] \\ &= \underbrace{\mathbb{E}_{\mathbf{x}'} \left[H(\mathrm{y}, F(\mathbf{x}')) - H(F(\mathbf{x}'), \mathbb{E}_{\mathbf{x}'} F(\mathbf{x}')) - H(\mathrm{y}, \mathbb{E}_{\mathbf{x}'} \mathrm{y}) \right]}_{\mathrm{bias}^2(\mathrm{y})} + \underbrace{H(\mathbb{E}_{\mathbf{x}'} F(\mathbf{x}'))}_{\mathrm{Var}(F(\mathbf{x}'))} + \underbrace{H(\mathbb{E}_{\mathbf{x}'} \mathrm{y})}_{\mathrm{Var}(\mathrm{y})} \quad . \end{aligned} \tag{2}$$

The part we are interested in is $\mathrm{Var}(F(\mathbf{x}')) = -\sum_i \mathbb{E}_{\mathbf{x}'} F(\mathbf{x}') \log \mathbb{E}_{\mathbf{x}'}(F(\mathbf{x}'))$. This measures how consistent the prediction is around $\mathbf{x}$ as we discussed in the high level idea section.

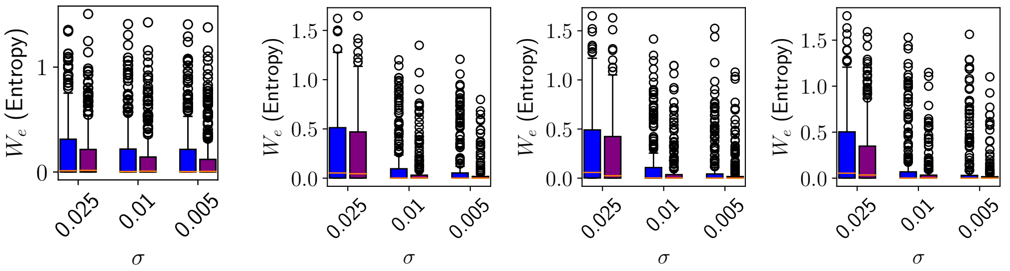
Finally, we define our measure W_e (**W** for cross-entropy loss) as the approximation to $\mathrm{Var}(F(\mathbf{x}'))$:

$$W_e(\mathbf{x}) = \sum_i -A(\mathbf{x})_i log\left(A(\mathbf{x})_i + c\right) \quad , \tag{3}$$

where $A(\mathbf{x})_i = \mathbb{E}_{\mathbf{x}'}(\mathrm{argmax} F(\mathbf{x}') = i)$, the mean of one-hot vectors to use only the top-1 class, and c is a small constant ($1e^{-5}$) to avoid computing the logarithm of zero. Using the top-1 class makes it easier to interpret in terms of causing inputs, and enable us to compute this measure even for a blackbox model returning only top-1 class. To obtain the measurements, instead of using just one $\mathbf{x}$, we use several data points $\{x^1, \ldots, x^n\}$ and compute the measure for each, hence $W_e = \{W_e(\mathbf{x}^1), \ldots, W_e(\mathbf{x}^n)\}$.

Note again that we have a few critical differences to Kohavi et al. (1996) and James (2003). We compute the expectation around $\mathbf{x}$ by drawing random variable $\mathbf{x}'$ from $\mathcal{N}(\mathbf{x}, \sigma^2)$ to obtain local measurements. This is important for two reasons. First, overfitting can be local, and the existing global measures can overlook regional overfitting. Second, global variance can be perceived much higher as it utilizes only a single mean value. Oftentimes, due to the diversity of the input points, using the mean of all points might not be suitable, as it can be far from every data point. Also, we measure many such points to build a distribution where we can apply diverse aggregation or statistical tests.

In other ML models using mean squared error loss, we have $W_v = \mathbb{E}_{\mathbf{x}'}(F(\mathbf{x}) - \mathbb{E}_{\mathbf{x}'} F(\mathbf{x}))^2$. This version shows mostly similar results to W_e, so we will focus on W_e due to the page limit, and refer interested readers to the Appendix. We also want to briefly remark that Yang et al. (2020) decompose the cross entropy function into a sum of KL-divergences instead of cross-entropies. James (2003) states that there can be multiple decompositions, especially for an asymmetric loss. Also, the KL-divergence can have similar trend since KL-divergence is cross-entropy minus entropy.



(a) Fashion MNIST.

(b) CIFAR10 - 3 random draws.

Figure 2: Differences of test (blue) and training (purple) data after training using W.

4 Empirical Evaluation

Unfortunately, W_e comes with many parameter choices. This includes the amount of noised points n, the variance of these points, σ, and the number of test points. A detailed ablation study for all parameters can be found in the Appendix. However, as we see in this section, reasonable values can be found for each. In this section, we vary the amount of noised points n between 500 and 2000, merely to show that there is little effect of this number on W_e as long as it is decently large (> 250). The choice of σ and the number of test points used to compute the measure are more important. The first, σ, is varied between 0.15, 0.1 as low as 0.005. Small values are appropriate for fine-grained task (distinguish test and training data), larger for more rough-grained (monitor progress during training). Finally, the amount of test points depends on the setting as well. We show that starting with 250 test-points, the measure is fairly stable.

We first describe the overall setting, and then show that W_e is able to distinguish training and test data. Afterwards, we investigate how the measure captures small differences in the training setting, and conclude the section with a detailed study on the measure on individual classes.

Setting. To test W_e, we deploy small networks on Fashion MNIST (Xiao et al., 2017), which achieve an accuracy of around 88%. These networks contain a convolution layer with 32 3×3 filters, a max-pooling layer of 2×2, another convolution layer with 12 3×3 filter, a dense layer with 50 neurons, and a softmax layer with 10 neurons. We further experiment on CIFAR10 (Krizhevsky and Hinton, 2009), where we train a ResNet18 (He et al., 2016) 200 epochs to achieve 91.8% accuracy.

Plots. We plot the distribution of W over n test points using box-plots. These plots depict the mean (orange line), the quartiles (blue boxes, whiskers) and outliers (dots). We follow the standard definition for outliers in Frigge et al. (1989): An outlier is defined as a point further away than 1.5 the interquartile range from the quartiles. More concretely, Q_{25} is the first quartile and Q_{75} is the third quartile (and Q_{50} is the median). Value ν is an outlier iff

$$\nu > Q_{75} + 1.5 \times (Q_{75} - Q_{25}) \text{ or } \nu < Q_{25} - 1.5 \times (Q_{75} - Q_{25}) \,, \tag{4}$$

in other words if ν is more than 1.5 times the interquartile range $(Q_{25} - Q_{75})$ away from either quartile Q_{25} or quartile Q_{75}.

4.1 Measuring Overfitting: Training vs Test Data.

An overfitted network will have adjusted better to the training data than to the unseen test data. Since W_e captures overfitting, it should be possible to tell apart training and test data given measurement outputs. In the plots, we compare the distribution of our measure on 250 test (blue) and 250 training (purple) points from the same dataset. On each point, we sample 500 noise points and compute W_e with low σ to capture small differences. The results are plotted in Figure 2.

Results. In Figure 2a, training and test data differ for $\sigma \leq 0.025$. More specifically, the training data (purple) shows less spread than the test data. This translates to less entropy around training data, or more stable classification, as expected. To illustrate that the results are not cherry picked, we show the results for three different random draws on CIFAR in Figure 2b. At $\sigma = 0.0025$, the results become very obvious and the difference of the spread is clearly visible. In contrast to Fashion MNIST, however, differences in the measure at smaller σ can be spotted, but are less pronounced.

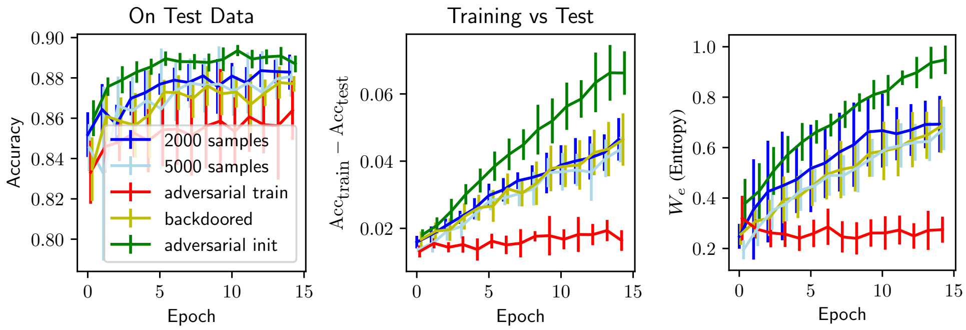


Figure 3: W_e on a several networks during training on Fashion MNIST. σ is set to 0.15.

Implications. The question whether or not a data point formed part of the training data is critical knowledge for many applications such as medical sciences, social networks, public databases, or image data (e.g., facial recognition). The corresponding attack is called membership inference (Shokri et al., 2017; Yeom et al., 2018; Salem et al., 2019). However, so far, even attacks with weak attackers (Salem et al., 2019) rely on output probabilities, and a straight-forward defense is to only output the one-hot labels. Our technique allows to circumvent these defenses, as the W can approximate a distribution that contains membership information based only on the one-hot output.

4.2 Measuring Overfitting during Training

As overfitting unfolds during training, we study how W_e develops during training. We connect this with a study of different training variations that either increase or decrease overfitting.

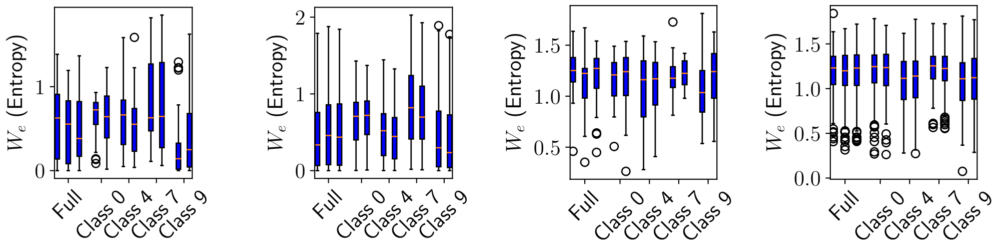
Setting. We train 15 networks on the Fashion MNIST dataset. In each iteration, we sample $2,000$ noised points. As we expect the surface to vary drastically during training, we choose a large $\sigma = 0.15$ around a given test point and compute the measure over the whole test set for stability. We choose five settings, where the first two cases are used to investigate the effect of the number of sampled points when computing the measure. We also investigate adversarial training, where one adds adversarial examples with the correct labels during training. This training makes the network more robust, and has been shown to reduce overfitting (Miyato et al., 2016). Further, we examine a backdoor scenario. Here, the attacker introduced a pattern in the training data which is always classified as the same class (Liu et al., 2018; Zhu et al., 2019). The last setting is adversarial initialization (Liu et al., 2019). Such an initialization is generated by fitting random labels before training on the correct labels. The authors showed that adversarial initialization increases overfitting. More concretely, even though the training error of such networks is low, the test error tends to be high. We do not compute the W during the pre-training with random labels.

Plots. Figure 3 depicts all five scenarios. The lines show the mean, the error bars the variance over the different runs. The blue line ('2000 samples') denotes the baseline networks, light blue ('5000 samples') the same baselines, where we sampled 5,000 noised points. The adversarial training is visualized as a red, backdoor as a yellow, and adversarial initialization setting as a green line.

Results. The left plot shows clean test accuracy. There is no a specific point at which the test error decreases. Hence, we also depict the difference between the training and test accuracy (middle plot). The adversarial training case (red line) is the lowest, corresponding to the least overfit networks. On the other hand, the adversarially initialized networks are the highest. The normal and backdoored models appear in the middle. This corresponds to our expectations about adversarial training and adversarial initialization. This order is preserved for W_e in the right plot.

4.3 Class-wise Differences

One might be tempted to think that a measure for overfitting should be class-independent. Yet, as accuracy for different classes may differ, so may overfitting properties. We study W_e on individual classes, connecting it with experiments to test stability depending on the number of test points used to compute W_e. In the previous experiments, measurements were done on 250 points or the whole



(a) F-MNIST, 25 points. (b) F-MNIST, 250 points. (c) CIFAR10, 25 points. (d) CIFAR10, 250 points.

Figure 4: Influence of number of points and classes on W_e. We show three (full data)/two (classes) different random draws for each setting to show variability.

test data. We now compare different draws of a very low size, 25, and the initial size 250 to confirm that the latter is sufficiently stable. To reduce the amount of plotted data, we randomly chose 4 classes for the plots in Figure 4.

Results. On Fashion MNIST, W_e varies when using 25 points to compute the measure (Figure 4a). For example, the mean of the measurement changes more than 0.4 when test points from all classes are used. An extreme case is class 9 (ankle boot), where one measurement has half the spread of the other measurement. The measure is already fairly stable around 250 test points (Figure 4b). Here, the largest change of mean is around 0.05. However, stability could be improved further if necessary by using more test points. On CIFAR, the results are analogous as visible in Figure 4c and Figure 4d.

4.4 Conclusion of Empirical Study

We showed that W_e indeed captures overfitting in different settings. After training, both training and test data exhibit differences in the measure. W_e also reflects training, and in particular configurations during training that affect overfitting like adversarial training or adversarial initialization.

Resulting Complexity. We observe that 250 points without labels are sufficient, where we sample $n = 250$ points around each. Hence, the time complexity of computing the measure is $250 \times n$ or $O(n)$. The sample complexity is even lower with only 250 *unlabeled* points.

5 Overfitting and Backdoors

To further show the practical usefulness of our measure, we investigate a common hypothesis from adversarial machine learning. This hypothesis by Wang et al. (2019) states that backdoors are enabled by overfitting. A backdoor is a particular pattern hidden in the training data, which can later be used to evade the trained classifier at test time (Liu et al., 2018; Zhu et al., 2019). Examples for such patterns on visual recognition or classification tasks are given in Figure 5. Most of these particular backdoors were also used by Chen et al. (2019) and Gu et al. (2019). In this section, we first present that W_e allows to distinguish data points with and without a functional trigger. We then investigate under which circumstances detection given an functional trigger is possible.

Attacker and Hypothesis. In some cases, the attacker can only control the victim's data. As the victim might inspect this data later, the amount of injected poisons is traditionally very small (roughly 250 for >50,000 training points) (Tran et al., 2018; Chen et al., 2019). In other cases, the attacker trains the model, and the trained model is then handed over to the victim. In this setting, the victim is not able to inspect the training data, and, thus, the attacker can poison a larger fraction of the training data to achieve better results (10-20% in for example (Wang et al., 2019; Chen et al., 2019). We investigate both settings and use 250 points for the former case and fix the percentage to 15% for the latter case in our experiments. Previous work (Wang et al., 2019) has indicated that backdoors are related to overfitting, equating the backdoor pattern to a *shortcut*

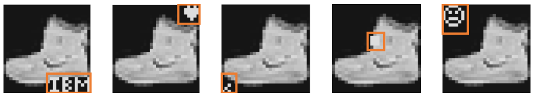
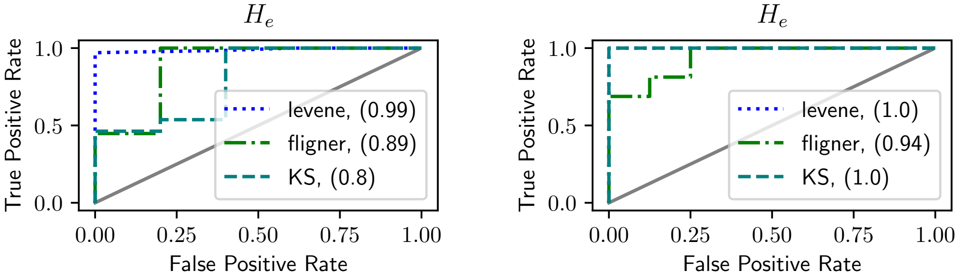


Figure 5: Possible backdoors on Fashion MNIST.

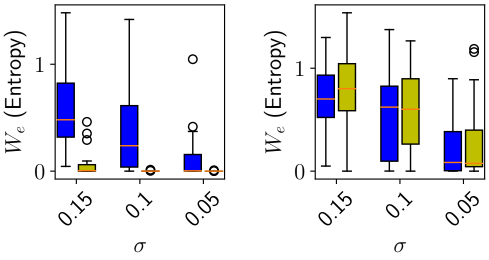


(a) CIFAR10, 15% poisons (b) Fashion MNIST, 250 poisons

Figure 7: Performance of statistical tests to detect backdoors using W_e.

between the target and victim classes. In this setting, the model has been poisoned such that *any* input plus the trigger will result in the desired target class.

General Setting. We train a slightly larger network on Fashion MNIST to allow the network to fit the backdoors well: a convolution layer (64 3×3 filters), a max-pooling layer (2×2), another convolution layer (32 3×3 filters), two dense layers with 500 neurons are followed by a softmax layer with 10 neurons. On CIFAR10, we use a smaller ResNet18. Due to the amount of networks trained, we train few epochs and the networks achieve only 63% accuracy on benign data. Both networks (Fashion MNIST and CIFAR) show a very high accuracy, $> 99\%$, on the backdoors.

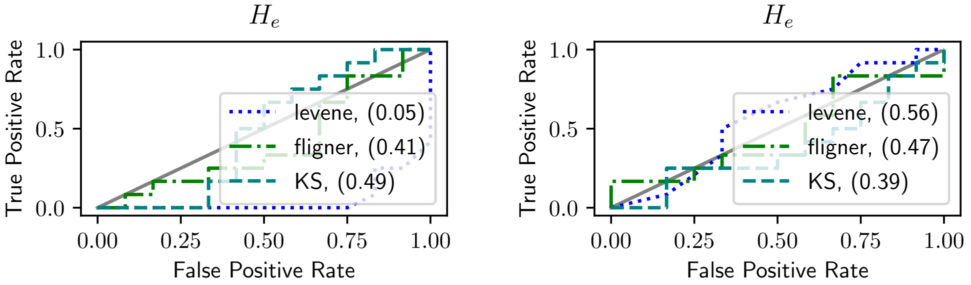


(a) Functional trigger (b) Benign data

Figure 6: W_e and backdoors on Fashion MNIST. We compare clean test data (blue) and a functional trigger (99% accuracy)/an unused trigger (9% accuracy on poisoning labels, 89% on clean labels).

$\mathbf{W_e}$ and Backdoors. If backdoors indeed rely on overfitting, we expect W_e to be high around backdoors, as the backdoors are close to the decision boundary. We thus pick as few as 25 test points and compute W_e once on these clean points (blue). We are using the same 25 points with the leftmost backdoor pattern from Figure 5 added (yellow). As we are investigating the *difference* in behavior for those 25 points, such a low number is indeed sufficient. As a sanity check, we also evaluate an unseen/random pattern (right plot in Figure 6), the fourth trigger from Figure 5. The model does not overfit but rather underfits backdoors: W_e is consistently low for the functional trigger, in particular at large σ. The classification output remains consistent regardless of the added noise, and the backdoors are consistent and stable. This behavior is the attacker's goal: as soon as the backdoor is present, other features become irrelevant. **A Detection Mechanism.** The differences in W_e above are quite pronounced and should be detectable. Statistical tests with the H_0 hypothesis that both populations have equal variance are the Levene (Olkin, 1960) and the Fligner test (Fligner and Killeen, 1976). The latter is non-parametric, the former assumes a normal distribution but is robust if the actual distribution deviates. Alternatively, the Kolmogorov-Smirnov (KS) test can be used with the mean-based test statistic proposed by Hodges (1958). The KS test evaluates the H_0 hypothesis that both samples are from the same distribution. To evaluate the performance of the tests with W_e, we train three networks on clean data and nine networks with different implanted backdoors. Of the latter backdoored networks, three are trained with backdoors one, three and four each from Figure 5. To obtain a valid false positive rate, we test for all five backdoors of Figure 5 on all networks. We then compute a ROC curve and the AUC value given the p-values and the ground truth of each test. In some cases, the test performs better when we remove outliers (defined as above or in Frigge et al. (1989)). In case that all points have the same value, we do not remove any point.

Performance of Detection. Our results are depicted in Figure 7. In general, the performance of the tests is very good. The Levene test performs consistently best. The worst AUC (0.8) occurs using the CIFAR10 dataset, where 15% of the data is poisoned. The performance of the Fligner (0.89) and Levene test (0.99) is higher. On Fashion MNIST, the worst AUC is 0.94, although only 250 points in training are poisoned. The other two tests show perfect performance at AUC 1.0.



(a) CIFAR, 15% poisons

(b) Fashion MNIST, 250 poisons

Figure 8: Performance of statistical tests to wrongly detect universal perturbations as backdoors. In contrast to previous plots, lower is better.

False negatives? To validate the above results, we repeat the previous experiments with universal examples, instead of backdoors, added to the test data. Universal adversarial examples are perturbations that can be added to a range of data points which are then missclassified (Moosavi-Dezfooli et al., 2017). They are hence somewhat similar to backdoors, yet they are not necessarily classified as one class consistently. We depict the results in Figure 8. Here, a true positive corresponds a universal perturbation classified as backdoor. Hence, we expect the performance to be low: The perturbation is not a backdoor, and should not be confirmed. The tests are now close to a random guess, with the exception of the Levene test with an AUC of 0.05 on CIFAR. The other two tests are close to a random guess with AUC of 0.41 (KS) and 0.49 (Levene). On Fashion MNIST, the tests are also close to random guess, with KS (0.39) being furthest away, followed by Levene (0.456) and Fligner (0.47). We conclude that the test does not confirm universal adversarial examples as backdoors.

False negatives! We now test a method to generate backdoor candidates given a network by Wang et al. (2019). The idea behind their algorithm is to generate a perturbation with a minimal mask that, when applied to any image (in a given input set), will cause *all* images to return the same fixed class. In the original work, the size of the perturbation indicates if the trigger is functional or not. We generate a trigger candidate for each class on clean and backdoored models. As before, we add the candidates to the 25 test points, compute the W_e, and feed the measure into the test. In contrast to our expectations, however, the test detects all of them as backdoors with the same, very low p-value. As a sanity check, we add the patterns found to the training data (which has not been used to generate them) and find that the accuracy on the targeted class is 100% in all cases.

Conclusion. Our measure, when combined with a test, reliably detects implanted backdoors. We further confirm that perturbations computed at test time (universal adversarial examples) are not detected. Yet, crafted backdoor candidates are found that manage to fool our detection. Hence, even benign networks contain variants of backdoors that emerge during training.

6 Conclusion

In this paper, we showed that overfitting can be measured by looking solely at the local decision boundary around comparatively few unlabeled data points. Our measure, W_e, shows differences between seen training and unseen test data. Furthermore, we can show how the network adapts during training, and reflects techniques applied to increase or decrease overfitting. Finally, W_e exhibits differences between the different classes the network learned, possibly linking it to an empirical measure for difficulty of a task as well.

Orthogonally, but also in strong relation to overfitting, we investigate a recent hypothesis from adversarial machine learning that backdooring in deep learning relies on overfitting. We find the reverse to be true: backdoors underfit rather than overfit. Applying W_e in this area showed us, however, that any benign deep network contains benign backdoors classification relies on. Hence, we are confident that W will provide many more insights when applied to other problems and open questions.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA) (FKZ: 16KIS0753).

References

Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pages 5389–5400, 2019.

Krikamol Muandet, David Balduzzi, and Bernhard Schlkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18, 2013.

Roman Werpachowski, András György, and Csaba Szepesvári. Detecting overfitting via adversarial examples. In *NeurIPS*, 2019.

Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. The information complexity of learning tasks, their structure and their distance. *arXiv preprint arXiv:1904.03292*, 2019.

Stuart Geman, Elie Bienenstock, and Ren Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

Jimmy Ba, Murat Erdogdu, Taiji Suzuki, Denny Wu, and Tianzong Zhang. Generalization of two-layer neural networks: An asymptotic viewpoint. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=H1gBsgBYwH`.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks, 2020.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing by virtual adversarial examples. In *ICLR*, 2016.

Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. Bad Global Minima Exist and SGD Can Reach Them. *ICML 2019 Workshop Deep Phenomena*, 2019.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. IEEE, 2019.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018.

Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *ICML*, pages 7614–7623, 2019.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. In *Advances in neural information processing systems*, pages 1172–1180, 2016.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve, 2019.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, pages 8000–8010, 2018.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety at AAAI*, 2019.

Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1265–1282, 2019.

Te Juin Lester Tan and Reza Shokri. Bypassing backdoor detection algorithms in deep learning. In *Euro S&P*, 2020.

Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–83, 1996.

Gareth M James. Variance and bias for general loss functions. *Machine learning*, 51(2):115–135, 2003.

Nader Ebrahimi, Ehsan S Soofi, and Refik Soyer. Information measures in perspective. *International Statistical Review*, 78(3):383–412, 2010.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

Michael Frigge, David C Hoaglin, and Boris Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE (S& P)*, pages 3–18. IEEE, 2017.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *NDSS*, 2019.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Ingram Olkin. *Contributions to probability and statistics: essays in honor of Harold Hotelling*. Stanford University Press, 1960.

Michael A Fligner and Timothy J Killeen. Distribution-free two-sample tests for scale. *Journal of the American Statistical Association*, 71(353):210–213, 1976.

JL Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3(5): 469–486, 1958.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, pages 1765–1773, 2017.

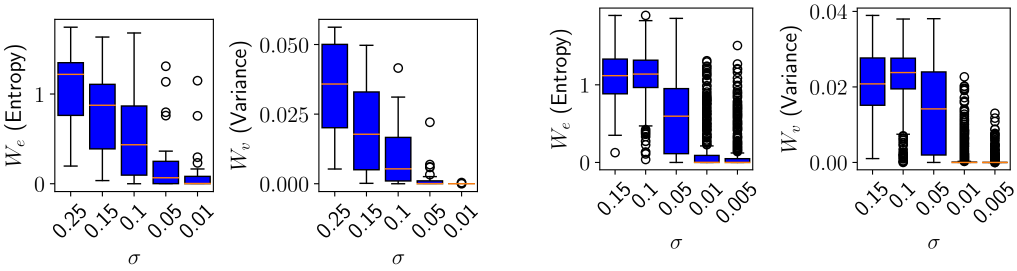


Figure 9: Influence of σ on W on Fashion MNIST (left) and CIFAR (right).

A Mean Squared Error Loss and Study on σ and n

We use the same networks as in the main paper, and study the measure W_v based on the mean squared error loss and the influence of both σ and n.

Variance of Noise Distribution. The plots evaluating different values of σ are depicted in Figure 9. As expected, when σ decreases, one class prevails around the given point eventually, and the output varies less. Indeed, when $\sigma \leq 0.1$ for both variance and entropy of the output decrease. Analogous to the values, the variance of both W_e and W_v decreases as σ decreases. We also plot the results on CIFAR. In particular for W_e, the distributions do not show such a clear trend for large σ (0.25 and 0.15), possibly due to the higher dimensionality. With decreasing σ, however, the results become as expected and the trend can be found on CIFAR10 as well.

Number of Points Sampled. To determine the importance of the number of noise points sampled around each of the 250 test points, we repeat the previous experiment with fixed $\sigma = 0.1$ and now vary n, the number of noised points sampled. The results are plotted in Figure 10. Surprisingly, we find that the number of points does not have a strong influence on W_e. In particular, for $n > 250$, the measure does not show significant differences. For W_v, on the other hand, the spread continuously decreases, and even $n > 5000$ might be beneficial.

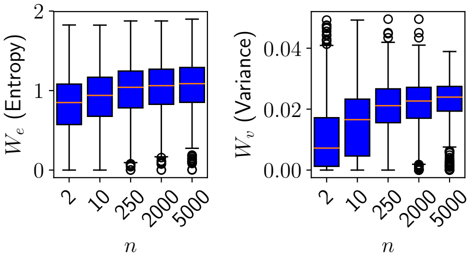


Figure 10: CIFAR10: Influence of n on W.