Journal of Machine Learning Research 22 (2021) 1-28 Submitted 6/20; Revised 3/21; Published 9/21

A Bayes-Optimal View on Adversarial Examples

Eitan Richardson EITANRICH@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem, Israel

Yair Weiss YWEISS@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem, Israel

Editor: Rob Fergus

Abstract

Since the discovery of adversarial examples – the ability to fool modern CNN classifiers with tiny perturbations of the input, there has been much discussion whether they are a "bug" that is specific to current neural architectures and training methods or an inevitable "feature" of high dimensional geometry. In this paper, we argue for examining adversarial examples from the perspective of *Bayes-Optimal classification*. We construct realistic image datasets for which the Bayes-Optimal classifier can be efficiently computed and derive analytic conditions on the distributions under which these classifiers are *provably robust against any adversarial attack even in high dimensions*. Our results show that even when these "gold standard" optimal classifiers are robust, CNNs trained on the same datasets consistently learn a vulnerable classifier, indicating that adversarial examples are often an avoidable "bug". We further show that RBF SVMs trained on the same data consistently learn a robust classifier. The same trend is observed in experiments with real images in different datasets.

Keywords: Adversarial Examples, Bayes Optimal, Generative Models, CNN, SVM

1. Introduction

Perhaps the most intriguing property of modern machine learning methods is their susceptibility to adversarial examples (Szegedy et al., 2014): for many powerful classifiers it is possible to perturb the input by an imperceptible amount and change the decision of the classifier. While adversarial examples were most famously reported for CNN classifiers (Szegedy et al., 2014), subsequent research has shown that other classifiers can also fall prey to similar attacks (Goodfellow et al., 2018). Attempts to make classifiers robust to these attacks have generated a tremendous amount of interest (e.g. Schott et al. 2019 and references within).

As a first step towards solving the problem, many authors have attempted to understand the source of the failure (Goodfellow et al., 2018; Szegedy et al., 2014; Tanay and Griffin, 2016; Fawzi et al., 2018b; Shamir et al., 2019). Broadly speaking, existing explanations fall into two groups (see section 4 for a more detailed discussion of related work). One approach argues that adversarial vulnerability is in some sense inevitable: either due to the geometry of high dimensions (e.g. Goodfellow et al. 2018, 2015; Shamir et al. 2019) or due

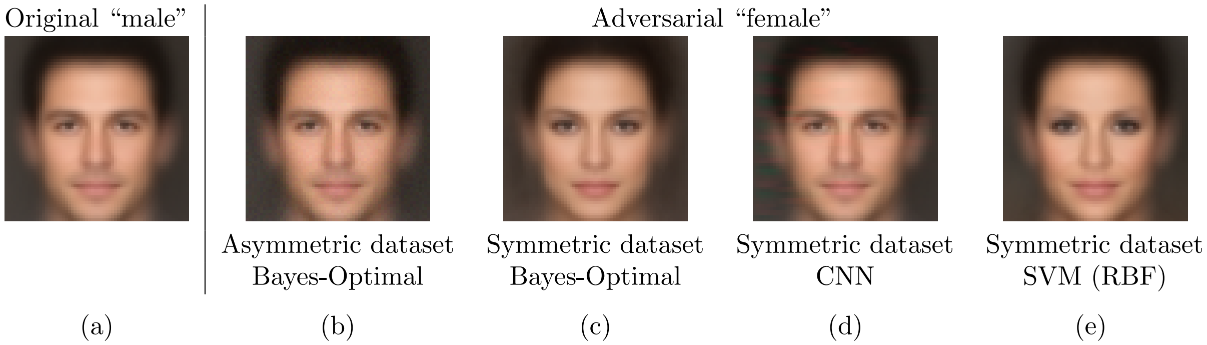


Figure 1: We present realistic image datasets for which the Bayes-Optimal classifier can be calculated efficiently and derive analytic conditions on when the optimal classifier will be robust or vulnerable to adversarial examples. When the data distribution satisfies certain asymmetries, the Bayes-Optimal classifier is vulnerable (b), but when the distribution is symmetric, the optimal classifier is robust and adversarial attacks become perceptually meaningful (c). Our experiments with these datasets show that CNN training consistently learns a vulnerable classifier (d) even when the optimal classifier is robust, while large-margin methods often succeed (e).

to fundamental limitations on robustness of classifiers trained from finite data (Schmidt et al., 2018). Another approach views adversarial vulnerability as a "bug" of CNNs and current training methods, which can be avoided by other architectures or training protocols (e.g. Tanay and Griffin 2016; Nakkiran 2019; Lyu and Li 2020; Schott et al. 2019).

One reason for the existence of many conflicting explanations may be the difficulty of analyzing adversarial vulnerability in a realistic yet tractable setting. In this paper, we provide such a setting. We construct realistic image datasets for which the Bayes-Optimal classifier can be efficiently computed and analyze the vulnerability of the optimal classifier. We derive analytic conditions on the distributions where *even the optimal classifier will be vulnerable* and other conditions where *the optimal classifier will be provably robust*. Figure 1a-c shows an example: our synthetic dataset of "male" and "female" face images. In the first, "asymmetric" distribution, the Bayes-Optimal classifier is vulnerable and an imperceptible perturbation is sufficient to change a "male" face (Figure 1a) to one that would be classified as "female" (Figure 1b). In the second, "symmetric" dataset, fooling the optimal classifier requires making large, perceptually meaningful changes (Figure 1c).

By training different classifiers on the "symmetric" datasets, in which the optimal classifiers are robust, we can disentangle the possible sources of vulnerability and avoid the accuracy-robustness tradeoff that may occur in commonly used datasets. Our results show that even when the optimal classifier is robust, standard CNN training consistently learns a vulnerable classifier (Figure 1d). At the same time, for exactly the same training data, RBF SVMs consistently learn a robust classifier (Figure 1e). Our results suggest that in many realistic settings, adversarial vulnerability is not an unavoidable property of learning in high dimensions but rather a direct result of suboptimal training methods used in current practice.

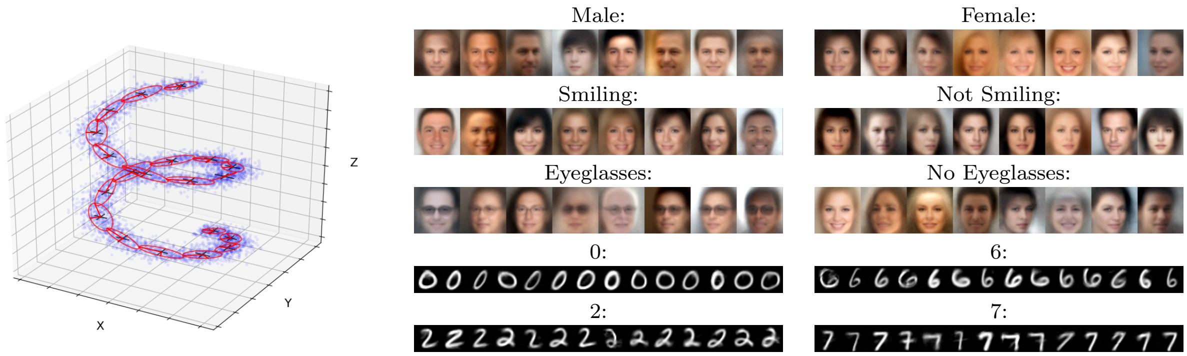


Figure 2: Left: A Mixture of Factor Analyzers (MFA) model for toy data in $\mathbb{R}^3$ (sampled 3D helical surface – blue points). Each component is a Gaussian on a learned 2-dimensional hyperplane with added axis-aligned noise. Right: Samples from our realistic MFA datasets.

The paper is organized as follows: In section 2.1 we present our method for constructing realistic image datasets that are equipped with computable Bayes-optimal classifiers. We then show, in sections 2.2, under which conditions these optimal classifiers are provably robust against adversarial attacks. Section 3 contains our experimental evaluation of the vulnerability of different classifiers trained on these datasets. Additional related work is discussed in section 4.

2. A Realistic Tractable Setting for Analysing Adversarial Vulnerability

This section describes our method for generating realistic image datasets with computable Bayes-Optimal classifiers that are provably robust to all adversarial attacks.

2.1 Image datasets with computable Bayes-Optimal classifiers

We focus on a two class classification problem and denote by $p_1(x)$, $p_2(x)$ the distribution of the two classes. We assume that $p_1(x), p_2(x)$ are known and that the two classes have equal priors, so the *Bayes-Optimal* classifier simply classifies x as belonging to class 1 if $p_1(x) > p_2(x)$ and to class 2 otherwise. It is well known that this classification rule is optimal and no other classification rule can achieve higher accuracy (assuming of course that $p_1(x), p_2(x)$ are correct) (Duda et al., 1973). Several recent papers (Schmidt et al., 2018; Ilyas et al., 2019) have analyzed adversarial examples in this setting, but only when $p_1(x), p_2(x)$ are both Gaussians with the same covariance, so that the optimal classifier is linear. Real image classification problems are of course very different from this simplified setting.

In our approach, we assume that images that belong to a single class form a *nonlinear, low-dimensional manifold* in pixel space. A standard way to model such manifolds is to use a mixture of Factor Analyzers (MFA) model (Ghahramani et al., 1996). The model is based on the observation that *locally* the manifold can be represented by a Gaussian whose covariance matrix is a sum of a low-rank matrix (representing the covariance on the manifold) and a diagonal matrix (representing the covariance off the manifold, e.g. due to sensor noise). Figure 2 (left) shows an example of a nonlinear 2D manifold in three dimensions, and its representation using an MFA model.

To create synthetic image datasets for classification, we start with a labeled training set with two classes. We then train separate MFA models $p_1(x)$, $p_2(x)$ on images from the two classes using the algorithm (and code) provided by Richardson and Weiss (2018). We now create a new training set by sampling images from the two models, and similarly a new test set. Since these datasets were created by sampling from a known model, we can calculate the Bayes-Optimal classifier. At the same time, the images are realistic and MFA models have been shown to capture much of the variability of the images in the original data (Richardson and Weiss, 2018).

We created 12 such datasets of faces (based on the CelebA dataset Liu et al. 2015) and 3 datasets of digits (based on MNIST)[1]. Figure 2 (right) shows samples from five such datasets that correspond to five binary classification problems: Male vs. Female, Smiling vs. Not-smiling, Eyeglasses vs. No-eyeglasses, 0 vs. 6, and 2 vs. 7. While these samples are typically somewhat blurred, it can be seen they are realistic and highly variable. In fact, in many real world applications one needs to classify somewhat blurry images (e.g. analyzing faces in surveillance videos). We again emphasize that for all of these datasets, we can efficiently calculate the Bayes-Optimal classifier.

The advantage of using a MFA model over other generative models such as VAEs or GANs (Kingma and Welling, 2014; Gulrajani et al., 2017) is that the log likelihood of any image can be calculated efficiently. Since the data were generated by the assumed distributions, this classifier is Bayes-Optimal. Indeed in all datasets we created, the accuracy of the classifier was close to 100%.

We now show under what conditions these optimal classifiers are *provably robust to all adversarial attacks*.

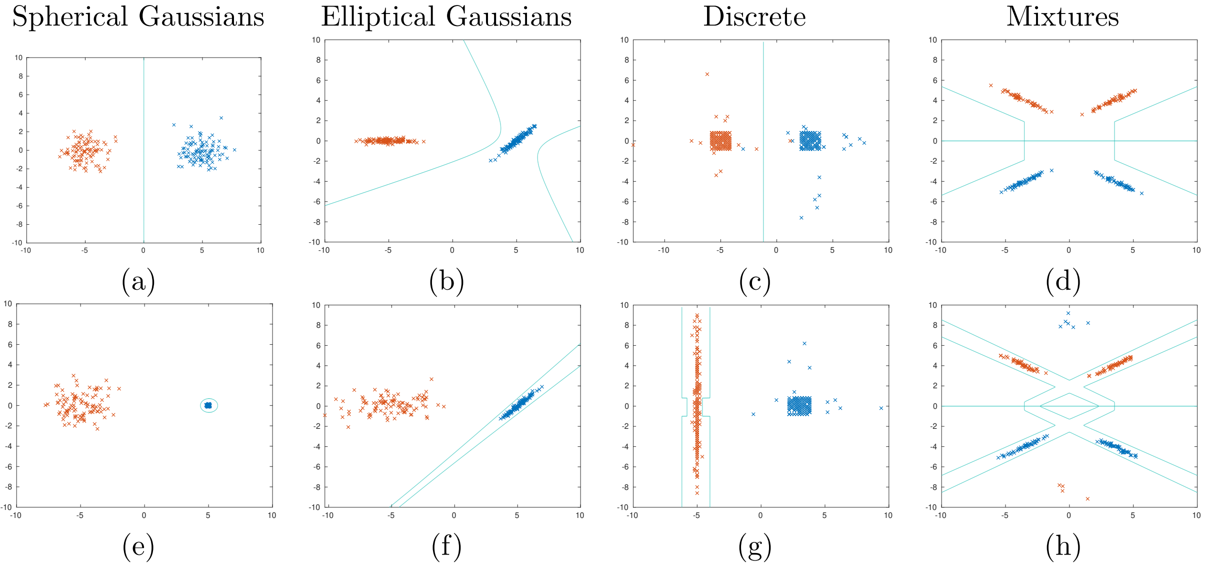
2.2 Provably robust or vulnerable optimal classifiers

A textbook example of Bayes-Optimal classification is when both classes are generated using Gaussians with the same spherical covariance, in which case the optimal classifier is a linear discriminant that is orthogonal to the difference between the two means (Figure 3a). In this case, if the distance between the two means is large relative to the covariance, then almost all points are far from the decision boundary and so any adversarial attack which only makes small changes to the input will typically fail. But as shown in the bottom of Figure 3 there are other examples where the decision boundary is close to many of the datapoints and an adversarial attack which only makes small changes to the input will often succeed[2]. What distinguishes these two cases? The following three theorems (whose proofs are given in the appendix) summarize our results.

Theorem 1: *Symmetric* isotropic Gaussian. If $p_1(x) = N(\mu, \sigma^2 I)$, $p_2(x) = N(\mu + d, \sigma^2 I)$ and $\sigma << \|d\|$ then for almost all points the Bayes-Optimal classifier is robust to any perturbation smaller than $\|d\|/2$.

[1] The datasets and models will be made publicly available after publication at `https://github.com/eitanrich/bayes-optimal-adv-examples`

[2] We deem an attack successful if it causes the classifier to change its prediction. Other definitions exist (Diochnos et al., 2018), but prediction-change is the most commonly used and practical one.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3: Top: examples of 2D distribution where the optimal decision boundary is far from the datapoints. Bottom: distribution where the optimal boundary is close to many of the datapoints.

Theorem 2: ***Symmetric*** **MFA.** If $p_1(x), p_2(x)$ are MFA models, i.e. a GMM where each Gaussian is of the form $N(\mu_i, \Sigma_i)$ and Σ_i is a covariance that is a sum of a low-rank matrix plus a diagonal matrix $\Sigma_i = A_i A_i^T + \sigma^2 I$ and all A_i have the same singular values. Let $\boldsymbol{d}$ be the minimal distance between any two subspaces of different manifolds[3]:

$$\boldsymbol{d} = \min_{i,j,z_1,z_2} \|\mu_i + A_i z_1 - (\mu_j + A_j z_2)\|$$

As $\sigma \to 0$ then for almost all points the Bayes-Optimal classifier is robust to any perturbation smaller than $\boldsymbol{d}/2$.

Theorems 1-2 referred to the *symmetric* case, in which the optimal classifier is *robust*. The next theorem refers to the *asymmetric* case, in which the optimal classifier is *vulnerable*:

Lemma 1: ***Asymmetric*** **Gaussian.** Assume p_1, p_2 are Gaussian distributions. Let v be a direction of minimal variance under Σ_1: $v = \arg\min v'^T \Sigma_1 v'$. Let σ_1^2 be the variance projecting x onto that direction when x comes from class 1. If $\sigma_1 \to 0$ and Σ_2 is full rank then almost any point in class 1 is arbitrarily close to the optimal decision surface.

Theorem 3: ***Asymmetric*** **GMM.** Assume that both $p_1(x)$ and $p_2(x)$ are Gaussian Mixture Models. If for every Gaussian in one class there exists a component in the other class so that they satisfy the *asymmetry* conditions of Lemma 1, then almost any point will be arbitrarily close to the optimal decision boundary.

Perhaps the most striking aspect of our three theorems is that they do not depend on the dimensions of the inputs x. Thus Theorem 1 guarantees that for two well-separated Gaussians, the Bayes-Optimal classifier is robust *for arbitrarily large dimensions.* Figure 4

[3] Two low-dimensional subspaces in high dimensions will almost always not intersect.

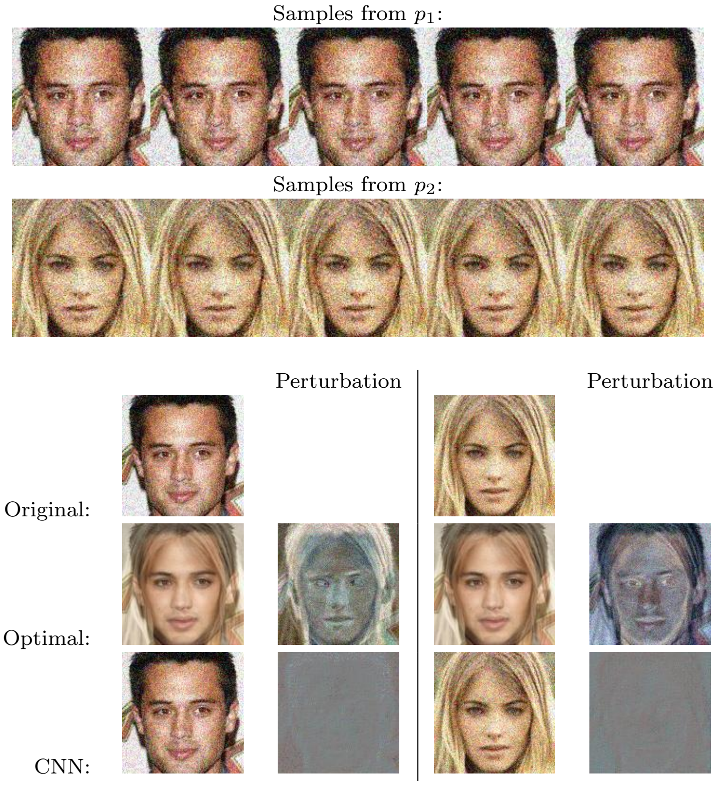
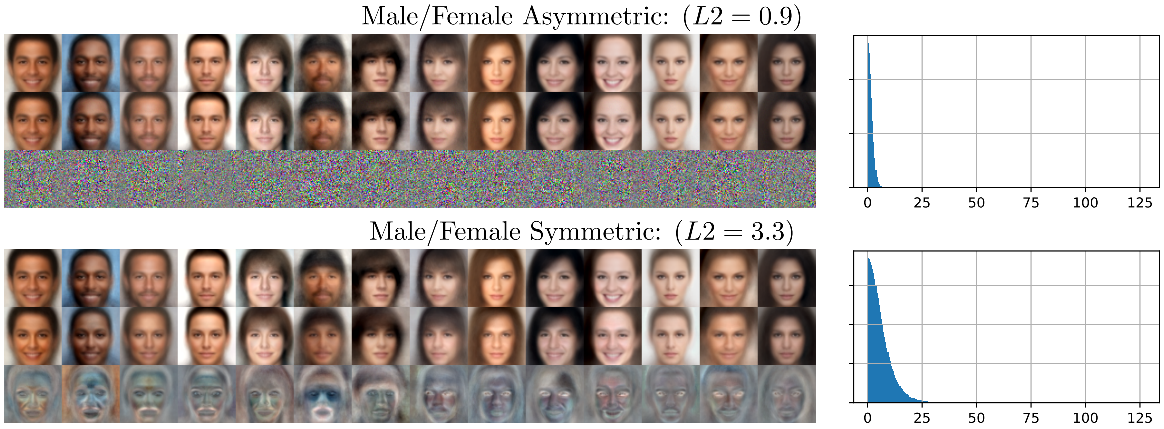


Figure 4: Top: Samples from our "noisy two-face" toy data (each class contains noisy versions of the same face image). Bottom: Attacking test samples. The Bayes-Optimal model is robust (middle row) while a CNN trained on this simple data is vulnerable (bottom row).

illustrates Theorem 1 for the toy case where the two Gaussian means are two specific face images and the inputs lie in $R^{49,152}$. As can be seen, the optimal classifier is robust – in order to fool it, the adversary needs to make large, perceptual-meaningful changes to the image. Specifically the decision boundary, like in Figure 3a is a plane that is evenly-distanced to the two means. Samples drawn from these distributions can then be used as training sets for evaluating the robustness of, for example, CNNs (Figure 4, bottom). Interestingly, even though the Bayes-Optimal classifier is robust, the CNN learns a vulnerable classifier. We will return to this point in section 3 using our realistic MFA datasets.

2.3 Symmetric and asymmetric datasets

To summarize our analysis, the Bayes-Optimal classifier will be *provably robust* when the covariances satisfy symmetry conditions and *provably vulnerable* when there are strong asymmetries. We therefore created symmetric and asymmetric variants of the MFA datasets. In the *symmetric* version, we regularized the MFA so that the "off manifold" variance σ is small and the same in all components and the distribution approximates the conditions of Theorem 2. In the *asymmetric* version, we added to each MFA model one "outlier"



Male/Female Asymmetric: $(L2 = 0.9)$

Male/Female Symmetric: $(L2 = 3.3)$

Figure 5: The vulnerability of the optimal classifier depends on the presence of asymmetries. Each result shows the original images, adversarial images and perturbations (magnified for visibility as necessary). Histogram of the perturbations in pixel values are on the right.

component with a diagonal covariance with much larger σ than all other covariances and a mean that is close to the global data mean. This version approximates the conditions of Theorem 3. In the next section we attack these optimal classifiers and compare the empirical robustness with our theoretical analysis.

2.4 Evaluating the robustness

Unlike our theoretical results, deciding whether or not a real classifier is robust or not requires an operational definition of what constitutes a "tiny" or "imperceptible" perturbation. We follow the standard practice of calculating the mean perturbation ℓ_2 norms of an adversarial attack (Schott et al., 2019; Carlini et al., 2019). We allow the adversary an unlimited budget in attacking the classifiers, and measure how large a perturbation was required to cross the decision boundary. The mean is calculated only over successful attacks, when the original sample was correctly classified and the adversarial example was not. Since this definition is sensitive to outliers and the particular choice of Euclidean norm, we also examined the histograms of changes made to each pixel in the adversarial attack. Finally, we visually inspected the adversarial images.

In all 15 datasets, we found that these three methods of defining robustness are consistent. For the face images, when the mean ℓ_2 is less than 1.5, then the adversarial images are almost indistinguishable from the original images, and the vast majority of the pixels in the adversarial images are within 5/255 intensity levels from their original value. On the other hand, when the mean ℓ_2 is around 3, then the adversarial images are perceptually quite different from the original ones, and many pixels differ by more than 5/255 from their original values. We used a simple gradient attack in which we take small steps in the direction of the gradient of the MFA log likelihood (details in appendix C). Similar results

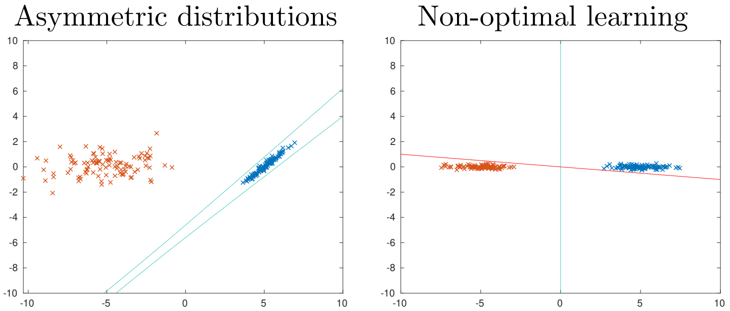


Figure 6: Why are CNNs so brittle? Is it because the data distributions are asymmetric so that the optimal classifier is also brittle (left) or is it due to non optimal learning (orange line, right) in cases where the optimal classifier is robust (blue line, right)?

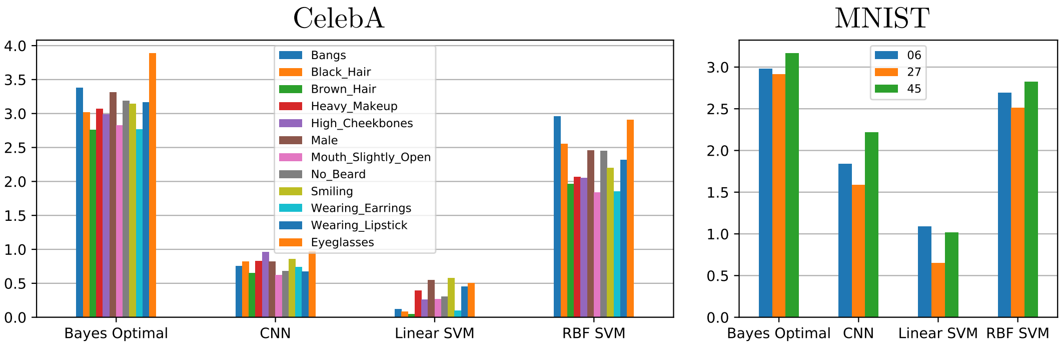


Figure 7: Adversarial attack perturbation sizes (mean L2 norm) for different models for all symmetric datasets.

are achieved with a standard implementation (Papernot et al., 2016) of the *CW-L2* attack (Carlini and Wagner, 2017).

As shown in Figures 5 (similar to Figure 1b,c), the difference between the symmetric datasets and asymmetric datasets is dramatic (see appendix D for similar results on other classes). When there exists a large asymmetry between the minimal variances of different Gaussians, the conditions of Theorem 3 hold, and a tiny imperceptible change is sufficient to fool the Bayes-Optimal classifier. However, when all Gaussians have the same minimal variance, the conditions of Theorem 2 hold, any adversary will need to make much larger changes and the adversarial examples become perceptually meaningful.

We now turn our attention to classifiers trained on our *symmetric* datasets and ask – will they be robust like the optimal classifiers for these datasets, or vulnerable.

3. Experiments: Why are CNNs So Brittle?

Given our analysis, the fact that modern machine learning methods are often susceptible to tiny adversarial perturbations may be due to two very different reasons (Figure 6). One

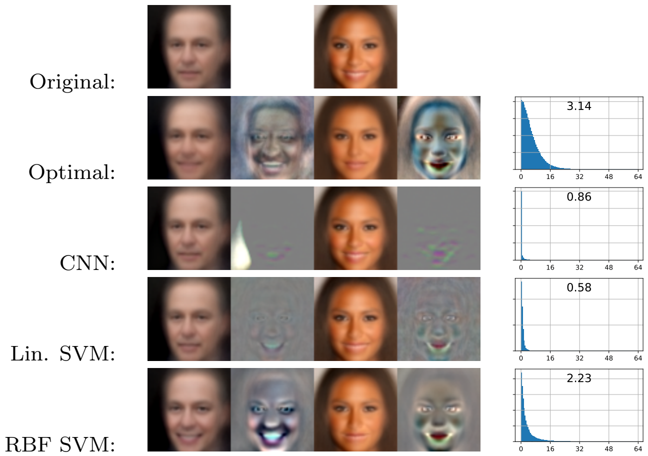


Figure 8: Adversarial examples, perturbations and histograms (with mean L2 value) for CelebA attribute 'Smiling'.

reason could be that the data distribution is asymmetric, so that the Bayes-Optimal classifier is not robust, and hence it is not surprising that a CNN is also not robust. A second possible reason is illustrated in Figure 6 (right): here the data distribution is symmetric and the Bayes-Optimal classifier is robust, yet SGD starting from a bad initial condition finds a brittle classifier. If this is the case, then the brittleness is not due to the data distribution but rather a failure of the learning method.

In order to separate the contribution of the dataset from the estimation method in the vulnerability of machine learning methods, we trained a CNN on samples from all 15 *symmetric* datasets described in section 2.3, and measured the vulnerability of the learned CNN. We used the CNN implementation and the *CW-L2* attack from the *CleverHans* library (Papernot et al., 2016) (details in appendices B, C). We asked: will the CNN find a brittle classifier even though the optimal one is robust?

Results are shown in Figures 7 and 8. *In all 15 cases, the CNN found a high accuracy classifier that was vulnerable to small adversarial perturbations, even though the optimal classifier is robust.* The difference is most dramatic in the CelebA tasks, where the CNN adversarial examples are almost indistinguishable from the original images (examples of the attacks on different datasets are shown in appendix D). While there are many possible architectures and optimization methods for CNNs, we did not find any improvement in the CNN robustness in our attempts to change the number of filters, layers, training iterations etc (Figure 9). In particular, Schmidt et al. (2018) have argued that one needs more training examples to achieve robust classification, so we systematically varied the amount of training images (generated dynamically at each SGD iteration), and found no significant improvement in robustness as we increased the number of training examples up to 1 million examples.

Is it possible to learn a robust classifier for these datasets from finite training data? To answer that question, we then trained linear and RBF Support Vector Machine (SVM) classifiers on exactly the same datasets. The linear SVM attempts to maximize the margin while maintaining high accuracy, but since the optimal classifier is nonlinear it ends up

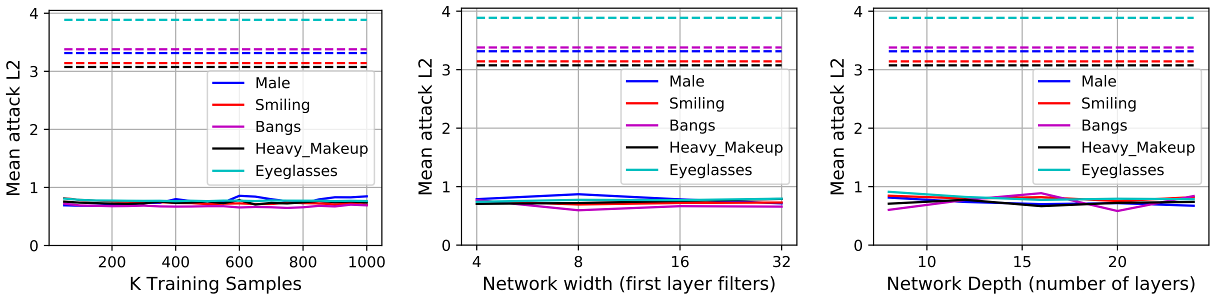


Figure 9: Wider and deeper CNNs and longer training with more data does not improve the robustness (optimal classifiers are shown as dashed lines for reference).

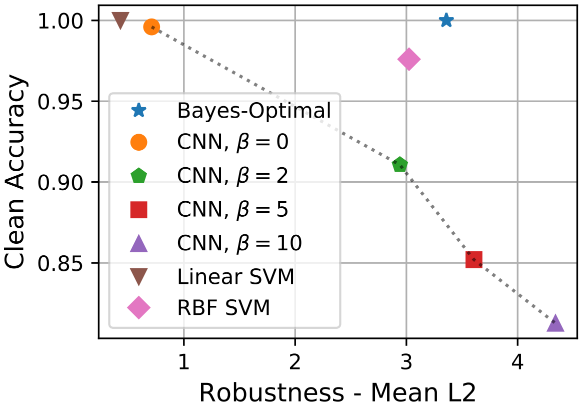


Figure 10: Accuracy/robustness tradeoff (means over the different datasets) in adversarially-trained CNNs (Zhang et al. 2019 β values) vs. models that are both accurate and robust – the Bayes-optimal and RBF SVM.

learning a brittle classifier. More importantly, *with an appropriate bandwidth parameter RBF SVMs find robust classifiers when trained on exactly the same data* (when the bandwidth parameter is too large, the RBF performs similarly to a linear SVM).

The robustness of CNNs can be improved using *adversarial training*, in which adversarial samples (of some selected attack) are injected during training. In Figure 10 we evaluate the robustness of CNNs trained using TRADES (Zhang et al., 2019) (first place in the NeurIPS 2018 adversarial challenge out of $\approx$ 2000 submissions). This algorithm has a tradeoff parameter β that trades off accuracy and robustness. In the only parameter setting that yielded a robust classifier, this approach gives significantly reduced accuracy compared to the Bayes-Optimal classifier or to the RBF SVM.

Returning to Figure 6, our results strongly support the hypothesis that for these cases brittleness is due to suboptimal learning methods, even when the data distributions are symmetric and the optimal classifier is robust.

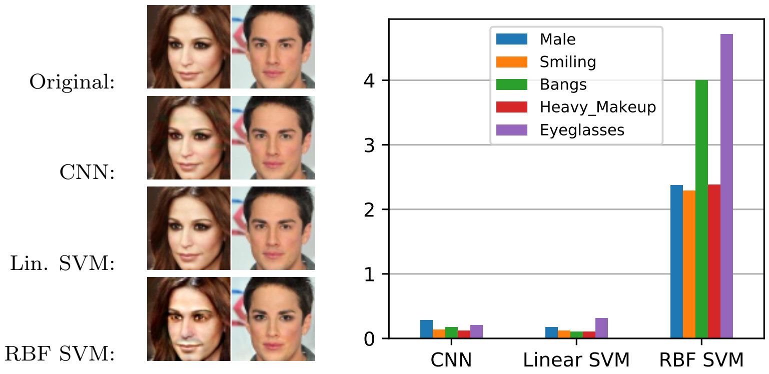


Figure 11: Adversarial examples for class Male/Female (left) and mean perturbation sizes (right) for different models trained on real CelebA images – results are consistent with our other experiments.

3.1 Training and testing on real data

One can ask, to what extent our analysis and experiments represent adversarial attacks on models trained on real data? To answer this question we trained CNNs and SVMs on five real CelebA attribute datasets, and indeed, as shown in Figure 11 and in appendix D, the results are similar to the synthetic symmetric datasets – CNN and Linear SVM are vulnerable while RBF SVM is robust and can only be fooled when the perturbations are perceptually meaningful.

To verify that our findings are not limited to specific datasets (faces, digits), we trained similar models on the more complex CIFAR-10 dataset (Krizhevsky and Hinton, 2009). We trained multiple binary classifiers on different class pairs and compared their adversarial robustness. As can be seen in Figure 12 the results agree with all other experiments. [4]

Note that unlike our proposed symmetric data, in which the "gold standard" optimal classifier is both robust and accurate, on real data, which may contain variance asymmetries, different models might reach different trade-off points between accuracy and robustness. In particular, it is known that RBF SVM accuracy is highly influenced by the hyperparameters C, γ and we performed only a minimal search to obtain these results. In future work, it would be interesting to explore the full regularization path as in (Hastie et al., 2004).

4. Related Work

One of the first explanations of adversarial examples in CNNs was that the decision surface learned by neural networks is "discontinuous to a significant extent", analogous to an attempt to discriminate the rational numbers from the rest of the real numbers (Szegedy et al., 2014). However as shown by our analysis, when there are strong asymmetries in the variances of

[4] Accuracy does not necessarily correlate with vulnerability: The RBF-SVM clean accuracy was on average 7% lower then the CNN's. Linear-SVM, which was the most vulnerable, had 25% lower accuracy compared to the RBF.

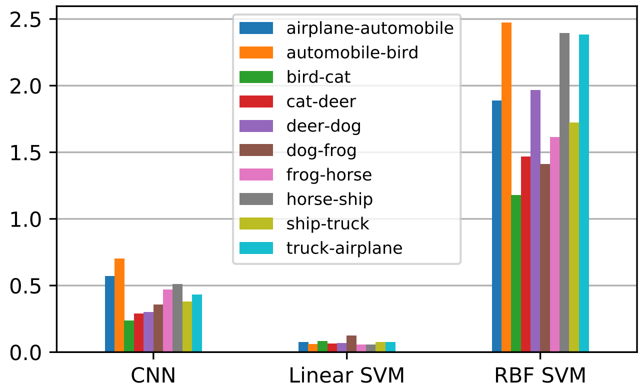


Figure 12: Robustness (mean required perturbation ℓ_2 sizes) for different binary classifiers trained on CIFAR-10 class pairs (class i vs class $i+1$). Results are consistent with our other experiments.

the two classes, adversarial examples can fool the optimal classifier even when the decision boundary is smooth and continuous.

A second prominent theory suggests that the problem is that neural network classifiers are "unreasonably linear" combined with the fact that they operate in high dimensions (Goodfellow et al., 2015, 2018; Gilmer et al., 2018; Fawzi et al., 2018a; Dohmatob, 2018; Mahloujifar et al., 2019). In high dimensions the output of a random linear classifier can be changed by making a small change in the ℓ_∞ norm of an example. Fawzi et al. (2018b) show a connection between the error rate of linear and quadratic classifiers and the adversarial vulnerability and show that linear classifiers in high dimensions must be vulnerable for data that is not linearly separable. Ford et al. (2019) show that adversarial vulnerability is closely related to the lack of generalization to random perturbations and that in high dimensions even moderate failures to generalize to high amounts of noise imply the existence of adversarial examples. Shamir et al. (2019) have also focused on the geometry of high dimensions arguing that adversarial attacks may be a "natural consequence of the geometry of R^n with the L_0 (Hamming) metric".

Both our analysis and our experiments suggest that high dimensionality is neither necessary nor sufficient for vulnerability. When strong asymmetries exist, even two dimensional datasets can be constructed such that the optimal classifier is vulnerable. At the same time, when there are no asymmetries, the optimal classifier is robust, even in very high dimensions. The same is true for "excessively linear" classifiers: the RBF SVM is probably no less linear than a CNN, yet it is robust in our symmetric and in real datasets.

The accuracy-robustness trade-off has also been suggested as an explanation for adversarial vulnerability (Zhang et al., 2019). Schmidt et al. (2018) present a model under which adversarially robust generalization requires more data. As our analysis shows, for symmetric datasets there is no trade-off between accuracy and robustness and the optimal classifier in terms of accuracy is also robust. Our experiments also show that with proper regularization (i.e. a RBF SVM), one can learn adversarially robust classifiers with the same amount of data for which CNNs learn a vulnerable classifier.

Most recently, Ilyas et al. (2019) have argued that adversarial examples are a feature, not a bug, and showed that one can in fact obtain information about the true decision boundary from adversarial examples. Their analysis suggests that vulnerability results from the presence of predictive features that are not robust. They presented a synthetic dataset which was constructed to not contain such features, and showed that CNN training on that dataset was robust. Our analysis in terms of symmetric vs. asymmetric datasets is similar to theirs but more general (they only considered linear classifiers). Our experimental results, however, are quite different and suggest that in more challenging settings, CNNs consistently learn vulnerable classifiers even when the asymmetries do not exist in the data distributions.

In (Nakkiran, 2019), a synthetic dataset was presented without nonrobust features, but CNN training led to vulnerable classifiers when the dataset was noisy. This result is consistent with Tanay and Griffin (2016) who show that adversarial vulnerability is related to overfitting in learning algorithms that are not sufficiently regularized. Similarly, Lyu and Li (2020) show theoretically and experimentally that the details of the training procedure can significantly change the robustness of CNNs. Our experimental results also highlight the need for regularization in more realistic and challenging settings (including real data), while our analysis points out that lack of robustness may also occur with no overfitting when the data is asymmetric. Our experiments also show that regularization is not sufficient: the linear SVM also attempts to maximize the margin, but due to its limited expressive power it still ends up learning a vulnerable classifier.

The analysis presented by Shafahi et al. (2019) agrees with ours: that it is not merely the dimensionality of the problem but also the distribution of images that determines vulnerability. Unlike Shafahi et al., we present a realistic yet tractable model to model the distribution of images and also present experimental evidence that in many cases CNNs learn a vulnerable classifier even when the optimal classifier for that dataset is robust.

Several works analyze adversarial attacks under the *manifold hypothesis* – assuming the data resides on a low-dimensional manifold embedded in the high-dimensional image space. Khoury and Hadfield-Menell (2018) claim that adversarial attacks are caused by the *codimension*, the large difference between the manifold and data dimensions. Our analysis uses a more general probabilistic reasoning and shows that adversarial examples exist regardless of the codimension. Zhang et al. (2020) explain adversarial examples as “perturbations perpendicular to the tangent plane of the data manifold”. They propose a PCA-based model-free attack. The paper claims that under the manifold assumption, “it is clear that data points in adversarial regions impose a potential threat to all classifiers”. In contrast, we show that in some data distributions (even under the manifold assumption), the optimal classifier and some trained classifier are robust and only CNNs are vulnerable. Jha et al. (2018) detect adversarial examples by their distance from the data manifold. This work is orthogonal to ours – we do not analyze the detectability of adversarial examples. Stutz et al. (2019) generate on-manifold adversarial examples by first encoding the original image and then perturbing the latent code. In our setting, all adversarial examples are off-manifold and still we show that CNNs fail even when the optimal classifier is robust.

Intuitively, we might expect classifiers based on generative models (e.g. “analysis by synthesis” Schott et al. 2019) to be more robust to adversarial attacks, since they model all the data, not just the discriminative features. But our analysis shows that even when such a classifier is based on the *true* generative model, it can be arbitrarily vulnerable, when the two distributions show strong asymmetries.

5. Discussion

Since the discovery of adversarial examples for CNNs there has been much discussion whether they are a "bug" that is specific to neural networks or a "feature" of high dimensional geometry. As far as we know, all previous works analyzed CNN vulnerability either on toy domains (e.g. points on a grid, concentric spheres) that are less suitable for testing CNNs, or on real image data for which a robust and accurate classifier may not exist. Unlike these works, our analysis enabled us to construct datasets that are both realistic and provably robustly-separable by the optimal classifier. This unique setting enables differentiating between unavoidable data-related vulnerability to causes related to the specific model or training method, and indeed we show that CNNs fail to reach the accurate and robust classifier in ALL our *symmetric* datasets, even as we varied the parameters of the architecture and training. When applying a state-of-the-art adversarial defence method (TRADES), the achieved robustness-accuracy trade-off is significantly inferior compared to the Bayes-Optimal and the RBF SVM classifiers. This suggests that, at least in some situations, the presence of adversarial examples represents a failure of current, suboptimal learning methods, rather than being an unavoidable property of learning in high dimensions.

We are by no means advocating a return to using RBF SVMs. Rather, we believe that explicit regularization methods for CNNs may enable learning robust classifiers while maintaining the power of deep architectures. Recent theoretical work on gradient descent methods suggests that they implicitly reward large margin classifiers in both shallow and deep architectures (Soudry et al., 2018; Poggio et al., 2017; Lyu and Li, 2020) although convergence to a large margin classifier may require exponential time.

In general, when trying to understand a complex effect, it is often useful to disentangle the different causes. The Bayes-Optimal perspective on adversarial examples identifies two possible causes: asymmetries in the datasets and suboptimal learning. Furthermore, it allows us to create tractable and realistic datasets in which one of the two causes can be clearly implicated. We are optimistic that this approach will be of great use in developing new learning algorithms that are practical and robust.

Acknowledgments

This work was supported by the Israeli Science Foundation, the Ministry of Science and Technology, the Gatsby Foundation and the Center for Interdisciplinary Data Science Research (CIDR).

Appendix A. Additional Proofs

A.1 Proof of Theorem 1 – *Symmetric* Isotropic Gaussian

Proof: For spherical and equal covariances, the Bayes-Optimal classifier simply projects the data onto the direction $\mu_1 - \mu_2$ and classifies a point based on whether that projection is closer to the projection of μ_1 or the projection of μ_2. This means that the problem reduces to a scalar problem, with two distributions whose means are at distance $\|d\|$ and have scalar variance σ^2. With high probability, under the assumption, all the points are close to one of the means, so they are at distance $\|d\|/2$ from the decision boundary. □

A.2 Proof of Theorem 2 – *Symmetric* MFA

We will first handle the simpler case where each class is a single Factor Analyzer:

Lemma A.1: *Symmetric* low-rank + diagonal. Suppose both classes have a covariance that is a sum of a low-rank matrix plus a diagonal matrix $\Sigma_i = A_i A_i^T + \sigma^2 I$ and A_1, A_2 have the same singular values and σ is the same for both classes. Let $\boldsymbol{d}$ be the minimal distance between the two linear subspaces:

$$\boldsymbol{d} = \min_{z_1, z_2} \|\mu_1 + A_1 z_1 - (\mu_2 + A_2 z_2)\|$$

As $\sigma \to 0$ then almost all points are at distance $\boldsymbol{d}/2$ from the Bayes-Optimal decision boundary.

Proof: If $x \sim N(\mu, AA^T + \sigma^2 I)$ then the distribution of x can be described by the following generative model:

$$\begin{aligned} z &\sim N(0, I) && (1) \\ \eta &\sim N(0, \sigma^2 I) && (2) \\ x &= Az + \mu + \eta && (3) \end{aligned}$$

We can also write the likelihood of x as:

$$P(x) = \int_z P(x, z) dz \quad (4)$$

Since x, z are jointly Gaussian, $P(x, z)$ is an unnormalized Gaussian function of x and the integral is given by the height of the unnormalized Gaussian divided by the square root of the determinant of the second derivative of $\log P(x, z)$ with respect to z (see for example MacKay and Mac Kay 2003 p. 341). So we can write:

$$\log P(x) = -\min_z \left(\frac{1}{2\sigma^2} \|\mu + Az - x\|^2 - \frac{1}{2} z^T z \right) - \frac{1}{2} \log \det(I + \frac{1}{\sigma^2} A^T A) + \frac{N}{2} \log(2\pi)$$

Since A_1, A_2 are assumed to have the same singular values, the log determinant term in the log likelihood is the same for both classes so the Bayes-Optimal classifier will classify a point as belonging to class 1 if:

$$-\min_z \left(\frac{1}{2\sigma^2} \|\mu_1 + A_1 z - x\|^2 - \frac{1}{2} z^T z \right) > -\min_z \left(\frac{1}{2\sigma^2} \|\mu_2 + A_2 z - x\|^2 - \frac{1}{2} z^T z \right)$$

As $\sigma \to 0$ the Bayes-Optimal decision rule will simply be:

$$\min_z \|\mu_1 + A_1 z - x\|^2 < \min_z \|\mu_2 + A_2 z - x\|^2$$

so that a point is classified based on which subspace it is closer to.

Now, as $\sigma \to 0$ almost any point x from class 1 will have distance close to 0 to the first linear subspace and distance of at least $\boldsymbol{d}$ to the second subspace. If we perturb x by a vector δ then the distance to each subspace can change by no more than $\|\delta\|$. This means that if $\|\delta\| < \boldsymbol{d}/2$, the distance to the first subspace will remain smaller than the distance to the second subspace, and hence the Bayes-Optimal classifier will not be fooled by any perturbation whose norm is less than $\boldsymbol{d}/2$. □

Mixture Model. We now assume that the distribution in each class can be represented as a Gaussian Mixture Model and denote by p_{ik} the kth Gaussian in class i and by π_{ik} its prior probability. We also assume that within each class, the components are well separated, i.e. that for each datapoint the assignment probabilities put all their mass on one of the components. More formally, denote by $g_{ik}(x) = \frac{\pi_{ik} p_{ik}(x)}{\sum_j \pi_{ij} p_{ij}(x)}$ the assignment probability of a datapoint x to component k, then we assume that for each x, $g_{ik}(x) = \delta(k - k_i^*(x))$ where $k_i^*(x)$ is the index of the component that is most likely to have generated x under probability $p_i(x)$. Under this assumption, the probability of generating a point x under p_1 is simply $p_1(x) = \pi_{k_1^*(x)} p_{1k_1^*(x)}(x)$. We will also assume that within each distribution, the assigned component does not change when we perturb x by a perturbation δ smaller than $d/2$: $k_i^*(x) = k_i^*(x+\delta)$.

Proof: By the well separateness assumption, the optimal classifier simply compares the likelihood of a point x under the most likely Gaussian component in each class. This means we can directly apply Lemma A.1, where p_1, p_2 are the most likely Gaussian component in each class. □

A.3 Proof of Theorem 3 – *Asymmetric* GMM

We will first prove Lemma 1 – *Asymmetric* Gaussian:

Proof: We denote by $\sigma_2 = d^T \Sigma_2 d$ the variance of the data in direction d under the distribution of the second class p_2. Note that by the assumption that Σ_2 is full rank, this variance must be nonzero. We write the vector x as (t, s) where t is the projection in direction d and s is a vector of projections in directions orthogonal to d. We denote by t_i the projection of μ_i in direction d. The decision surface as a function of t is a solution to:

$$\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right) t^2 + 2\left(\frac{t_2}{\sigma_2^2} - \frac{t_1}{\sigma_1^2}\right) t + \left(\frac{t_1^2}{\sigma_1^2} - \frac{t_2^2}{\sigma_2^2} + \log \frac{\sigma_1^2}{\sigma_2^2}\right) = \log p_2(s|t) - \log p_1(s|t)$$

Now since d is a direction of minimal variance, it must be an eigenvector of Σ_1 so that s and t are independent under p_1 and we can write $\log p_1(s|t) = \log p_1(s)$. Using the standard equation for conditional Gaussians, $p_2(s|t)$ will also be a Gaussian with the following mean and covariance:

$$\mu_{s|t} = \mu_s + \Sigma_2^{st}(t - t_2) \tag{5}$$

$$\Sigma_{s|t} = \Sigma_2^{ss} - \Sigma_2^{st} \frac{1}{\sigma_2^2} \Sigma_2^{ts} \tag{6}$$

where $\Sigma_2^{ss}, \Sigma_2^{st}$ are the appropriate submatrices of the covariance matrix Σ_2. As $\sigma_1 \rightarrow 0$ then $t \rightarrow t_1$ and $\mu_{s|t}$, $\Sigma_{s|t}$ will not depend on t. This means that the right-hand side of equation 5 depends only on s and not on t or σ_1.

As $\frac{\sigma_1}{\sigma_2}$ approaches zero, the solutions of this equation approach t_1. And because $\sigma_1 \rightarrow 0$, almost all samples from the first Gaussian will be close to t_1, so moving x by a tiny amount in direction d will change the optimal decision. □

The proof of the GMM case is now similar to that of the mixture model in Theorem 2, but using Lemma 1 instead of Lemma A.1.

A.4 Robustness of the Optimal Classifier – Non Gaussian Distributions

A natural question following Theorems 1-3 is to what extent the result depends on the Gaussian distribution. To address this, we now consider *discrete* distributions. We assume that every instance x is described by quantized features that can take on a discrete number of values. For example, the features can be wavelet coefficients of an image that are discretized into 256 possible values. This means that $p_1(f), p_2(f)$ are simply very large tables that give the probability of observing a particular discrete set of image features given each of the classes. Of course learning such a large table is infeasible without additional assumptions, but recall that we are analyzing the Bayes-Optimal case, where we assume $p_1(f), p_2(f)$ are known.

Lemma A.2: Assume there exists a feature i and a quantization level k so that $p_1(f_i = k) \rightarrow 1$. Assume also that for any feature vector f, $p_2(f) > 0$. Then almost any point in class 1 is one quantization level away from the optimal decision boundary.

Proof: We again write $f = (s, t)$ where s is the *ith* feature and t are all other features.

$$p_1(s,t) = p_1(s)p_1(t|s); \tag{7}$$

Now since $p_1(s)$ approaches 1 for $s = k$ and 0 otherwise, for almost any point in class 1 the value of that feature is equal to k. We now change the feature by one quantization level and obtain a new feature vector $(\tilde{s}, t)$ and $p_1(\tilde{s}, t) = p_1(\tilde{s})p_1(t|\tilde{s}) \rightarrow 0$. On the other hand, by the assumption $p_2(\tilde{s}, t) > 0$, so that this point would now be classified as belonging to class 2. □

As in the Gaussian case, we do not need $p_1(s)$ to be exactly equal to a delta function for the decision surface to be close to most points. It is enough that $p_1(\tilde{s})$ be much smaller than the minimal values of p_2 for the decision to be flipped when we replace s with $\tilde{s}$. Figures 3g,c illustrate this dependence. In both cases, the data is sampled from a discrete distribution where the features are simply discretization of the two spatial coordinates into 100 levels each. In other words, p_1 and p_2 are tables of size $10,000$ and each entry in the table represents the probability of generating a point at one of the $10,000$ possible locations. In the top example, the minimal value of the probability table is approximately the same in both classes, while in the bottom example, there is a strong asymmetry and the decision boundary becomes close to all points in one of the classes.

It is easy to see that Theorems 2,3 that discusses mixture distributions are applicable to the discrete case as well.

Appendix B. Models

In this section we provide additional information about the different classification models – architecture, hyper-parameters and training procedure.

B.1 MFA

A Mixture of Factor Analyzers (MFA) (Ghahramani et al., 1996) is a Gaussian Mixture Model where each component is a Factor Analyzer parameterized by a *low rank plus diagonal* covariance matrix. MFA provides a good trade-off between the non-expressive diagonal-covariance model and a full-covariance model, which is too computationally expensive for high-dimensional data such as full images.

The model for a single Factor Analyzer component is:

$$x = Az + \mu + \eta \ , \ z \sim \mathcal{N}(0,\, I) \ , \ \eta \sim \mathcal{N}(0,\, D)\,, \tag{8}$$

where A is the rectangular *factor loading* matrix, z is a low-dimensional latent factors vector, μ is the mean and η is the added noise with a diagonal covariance D (which may be isotropic: $D = \sigma^2 I$) This results in the Gaussian distribution $x \sim \mathcal{N}(\mu,\, AA^T + D)$. The MFA is a mixture of such Gaussians.

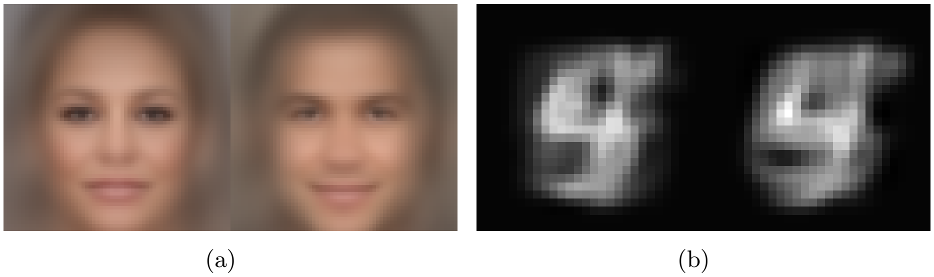
The MFA model was trained using the code provided by Richardson and Weiss (2018). The models are trained using Stochastic Gradient Descent. The training data (CelebA, MNIST) is first split by the desired binary attribute (e.g. Smiling / Not Smiling) and then a separate MFA model was trained independently for each subset of training samples. Because of imbalance in the number of samples per class in CelebA, we set the number of components as the number of samples divided by 1000. For MNIST we used a fixed value of 25 components per class. We chose an MFA latent dimension of 10 for CelebA and 6 for MNIST.

To allow attacking the MFA model with standard adversarial attacks such as CW-L2, we implemented it in TensorFlow as a standard CleverHans (Papernot et al., 2016) model.

B.2 Bayes-Optimal

The MFA model is the Bayes-Optimal classifier when the data is sampled from that model. We modified the MFA models that were trained for the different classes to define pairs of Bayes-optimal models – *symmetric* and *asymmetric*.

For the *symmetric* models, we simply fixed all noise variance values in D to a small value ($\sigma = 0.01$). To construct the *asymmetric* models we added two outlier components (one for each class) that are equal to the dataset global mean plus changes along a direction of low-variance: We performed PCA over the entire dataset and took the eigenvector for the 50th largest eigenvalue as this direction, where the mean of one outlier is in the positive direction and the mean of the other in the negative one. We set $A = 0$ and $\sigma = 0.5$ for both outliers, making them spherical Gaussians around the two means with relatively large noise compared to the other components. See Figure 13 for the outlier component means for CelebA and for MNIST.



(a) (b)

Figure 13: Outlier components means for CelebA (a) and for MNIST (b). Samples from these components are the shown means plus strong isotropic noise with $\sigma = 0.5$.

B.3 CNN

We used the reference CNN implementation from the CleverHans library (Papernot et al., 2016), which is a benchmark library for evaluating adversarial attacks and defences. The network consists of 2D convolution layers with a kernel size of 3 and *Leaky ReLU* activations. We used a stride of 2 in several equally-spaced layers along the depth of the network to reduce the spatial dimension and at each such layer we doubled the width (number of channels). The network ends with a single fully-connected layer. All other hyper-parameters were left at their default values and the optimization method was *Adam*. Our baseline small CNN achieves 100% train and test accuracy on the symmetric datasets and we also experimented with increasing both the depth and the width of the CNN by a large factor (see Figure 9).

B.4 Linear SVM

We used the standard 2-class linear SVC implementation provided by sklearn/libsvm (Pedregosa et al., 2011). Linear SVM is trained directly on the vectorized image samples. The learned model consists of a weight vector W and a scalar bias b. The decision for a sample x is simply $\text{sign}(W^T x + b)$.

B.5 RBF SVM

We used sklearn for the Radial Basis Function (RBF) kernel SVM as well. Selection of two hyper-parameters is required, the radial kernel coefficient $\gamma = \frac{1}{2\sigma^2}$ and C, a regularization term. We used the default $C = 1.0$ and the highest γ value that still provided a high classification accuracy.

Appendix C. Attacks

C.1 CW-L2

The Carlini & Wagner L2 attack (Carlini and Wagner, 2017) is a recommended strong attack that minimizes the perturbation L2 norm. The attack minimizes a weighted combination of a classification loss with the perturbation L2 size. The relative weight is a parameter that is found using a binary-search. We used the CleverHans implementation with the following hyper-parameters: 500 iterations, 3 binary-searches and a learning rate of 0.01.

C.2 Gradient Descent Attack

Since the MFA and SVM models provides a simple closed-form expression for the likelihood and its gradient, we implemented a simple and fast version of a gradient-attack for these models. Our attack performs multiple fixed-size steps in the direction of the gradient of the difference in log-likelihood between the source and target Gaussian components, until the decision boundary is crossed. We repeated some of the experiments with the (much slower) CW-L2 attack and verified that the results are similar (i.e. models that are shown to be robust to our gradient descent attack are also robust to the CW-L2 attack with similar perturbation magnitudes).

Appendix D. Additional Results

In this section we provide additional experimental results for the different datasets, attributes and models.

D.1 Distance to the Decision Surface – Symmetric Datasets

According to Theorem 2, if the data can be represented as a mixture of low-rank plus diagonal Gaussians and the off-manifold (diagonal) variances are all small and similar (no strong asymmetries), then the distance from a sample to the optimal decision surface will be half the distance to the nearest component subspace in the other (target) class.

In toy distributions (e.g. Figure 3) we can control these distances arbitrarily, but what will the distances between subspaces be in our *symmetric* datasets, which approximate the manifold of real image datasets? As can be seen in Figure 14 (for Male/Female dataset), the distances to the nearest subspace in the other class are large – mean of 6.2). These values are consistent with the mean adversarial perturbation sizes that were actually required to fool the Bayes-Optimal classifier (mean ℓ_2 of 3 – half of the mean distance to the nearest component subspace). The same is true for the other symmetric datasets.

D.2 Symmetric vs. Asymmetric

Figure 15 presents additional examples comparing symmetric and asymmetric datasets and the relative robustness of their Bayes-Optimal classifiers to adversarial examples.

D.3 Symmetric Datasets

Table 1 lists the clean and adversarial classification accuracy of all models for all *symmetric* datasets.

Figures 16-19 show original and adversarial samples and perturbations as well as histograms of the perturbations in pixel values for all models in different *symmetric* datasets (the number at the top of each histogram is the mean perturbation L2 over all test samples).

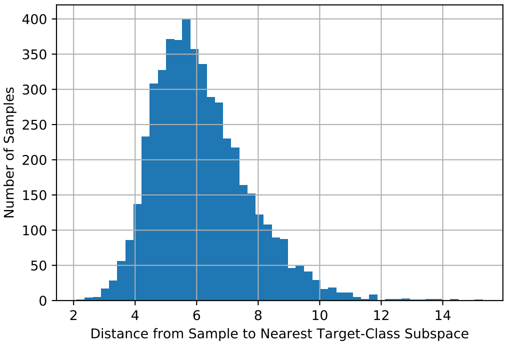
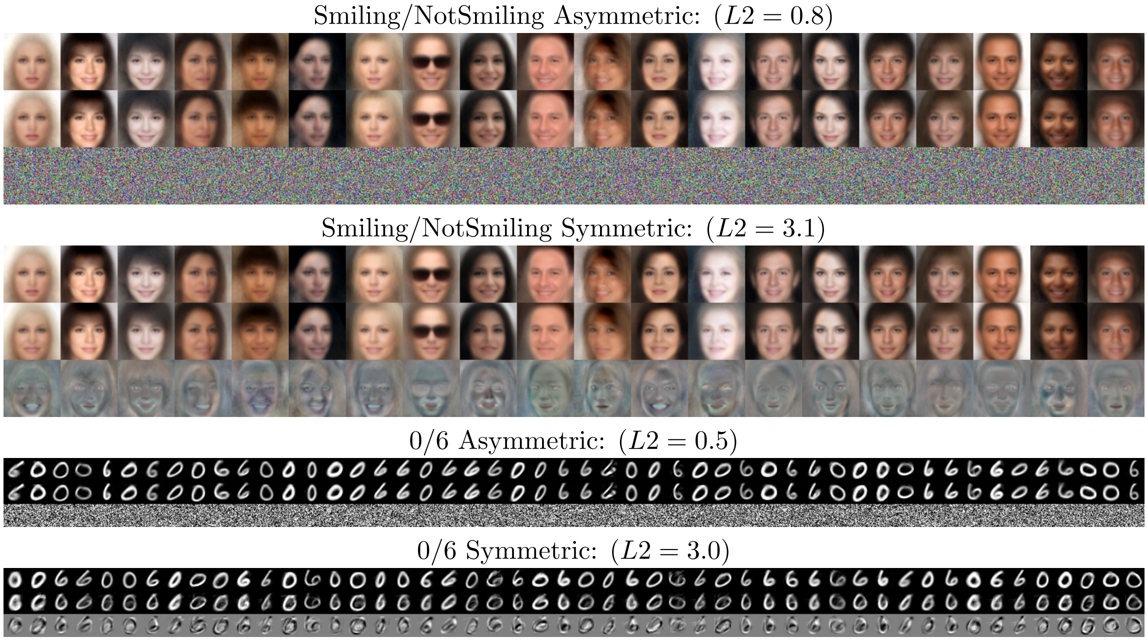


Figure 14: Histogram of distances (ℓ_2) from samples from the Male/Female *symmetric* dataset to the nearest Gaussian subspace in the other class. Results are consistent with the required adversarial perturbation magnitude.


Smiling/NotSmiling Asymmetric: ($L2 = 0.8$)

Smiling/NotSmiling Symmetric: ($L2 = 3.1$)

0/6 Asymmetric: ($L2 = 0.5$)

0/6 Symmetric: ($L2 = 3.0$)

Figure 15: The vulnerability of the optimal classifier depends on the presence of asymmetries. Each result shows the original images, adversarial images and perturbations (magnified for visibility as necessary). In the symmetric datasets, attacking the optimal classifier requires large and perceptually meaningful perturbations.

Table 1: Clean and adversarial (in brackets) classification accuracy values for different models for the *symmetric* datasets.

Dataset / Attribute	Bayes-Optimal	CNN	Linear SVM	RBF SVM
CelebA / Bangs	100% (0%)	100% (0%)	100% (0%)	96% (18%)
CelebA / Black Hair	100% (0%)	98% (2%)	100% (0%)	96% (26%)
CelebA / Brown Hair	100% (0%)	99% (1%)	100% (0%)	86% (16%)
CelebA / Heavy Makeup	100% (0%)	100% (0%)	100% (0%)	96% (14%)
CelebA / High Cheekbones	100% (0%)	100% (0%)	100% (0%)	96% (10%)
CelebA / Male	100% (0%)	100% (0%)	100% (0%)	100% (8%)
CelebA / Mouth Slightly Open	100% (0%)	100% (0%)	100% (0%)	94% (8%)
CelebA / No Beard	100% (0%)	100% (0%)	100% (0%)	94% (10%)
CelebA / Smiling	100% (0%)	100% (0%)	100% (0%)	96% (10%)
CelebA / Wearing Earrings	100% (0%)	99% (1%)	100% (0%)	94% (16%)
CelebA / Wearing Lipstick	100% (0%)	100% (0%)	100% (0%)	92% (18%)
CelebA / Eyeglasses	100% (0%)	100% (0%)	100% (0%)	100% (22%)
MNIST / 06	100% (0%)	100% (0%)	100% (0%)	100% (4%)
MNIST / 27	100% (0%)	100% (0%)	100% (0%)	100% (7%)
MNIST / 45	100% (0%)	100% (0%)	100% (0%)	100% (8%)

Table 2: Clean and adversarial (in brackets) classification accuracy values for different models for *real* CelebA data.

Dataset / Attribute	CNN	Linear SVM	RBF SVM
CelebA / Male	94.7% (5.30%)	89.7% (10.3%)	89.7% (11.3%)
CelebA / Smiling	88.0% (12.0%)	88.3% (11.7%)	84.3% (16.3%)
CelebA / Bangs	94.0% (6.00%)	87.0% (13.0%)	93.7% (24.7%)
CelebA / Heavy Makeup	86.0% (14.0%)	83.7% (16.3%)	85.7% (20.7%)
CelebA / Eyeglasses	100.% (0.00%)	93.7% (6.33%)	96.3% (32.3%)

D.4 Results on Real Data

D.4.1 Training and testing on real data

As shown in the main paper (Figure 11), results on real data are consistent with our *symmetric* datasets results – CNN and Linear SVM learn a vulnerable classifier while RBF SVM is robust. Table 2 lists the clean and adversarial accuracy values for the different models trained on different *real* image datasets.

D.4.2 Training on symmetric data and testing on real data

To estimate how close our symmetric datasets are to the real datasets, we tested the CNNs that were trained on the symmetric dataset on real test samples and compared the test accuracy to that of CNNs that were trained on the real training data. As can be seen in Table 3, there is an average accuracy reduction of just 5%, indicating that the symmetric datasets are not that far from the original data.

Table 3: Real test images classification accuracy for CNNs trained on *real* training data vs CNNs trained on samples from the *symmetric* dataset.

Dataset / Attribute	Real Train Data	Symmetric Train Data
CelebA / Male	92.3%	88.0%
CelebA / Smiling	89.5%	85.6%
CelebA / Bangs	94.3%	90.4%
CelebA / Heavy Makeup	86.3%	79.9%
CelebA / Eyeglasses	98.4%	91.7%

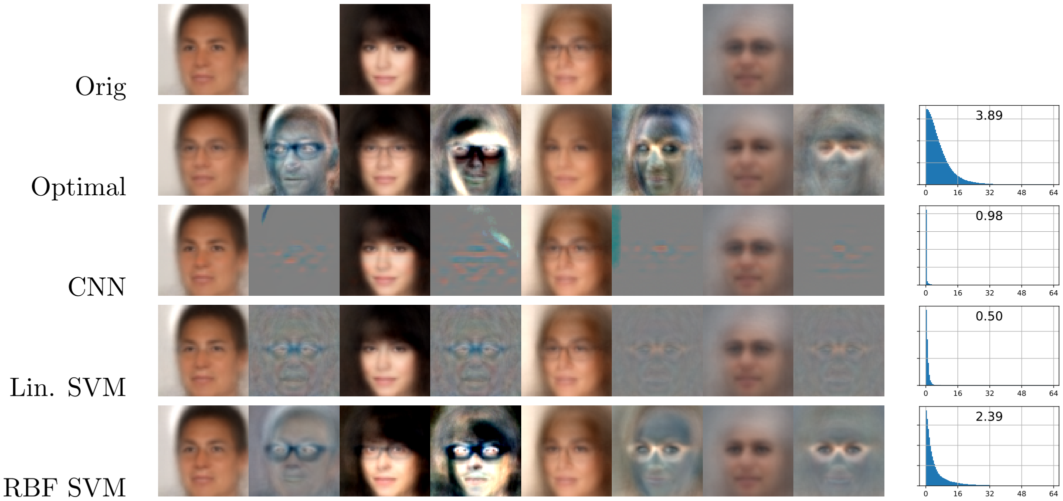


Figure 16: Samples, perturbations and histograms for CelebA attribute 'Eyeglasses'

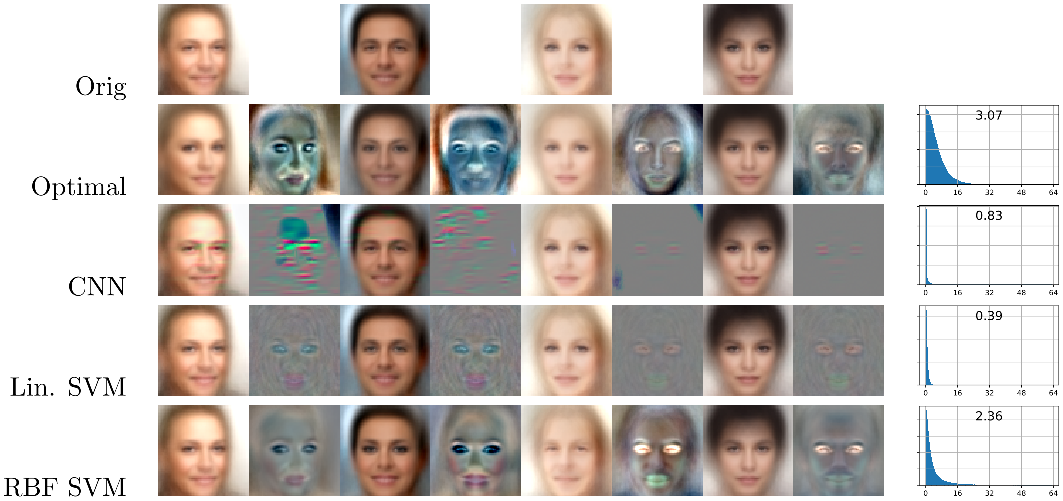


Figure 17: Samples, perturbations and histograms for CelebA attribute 'Heavy Makeup'

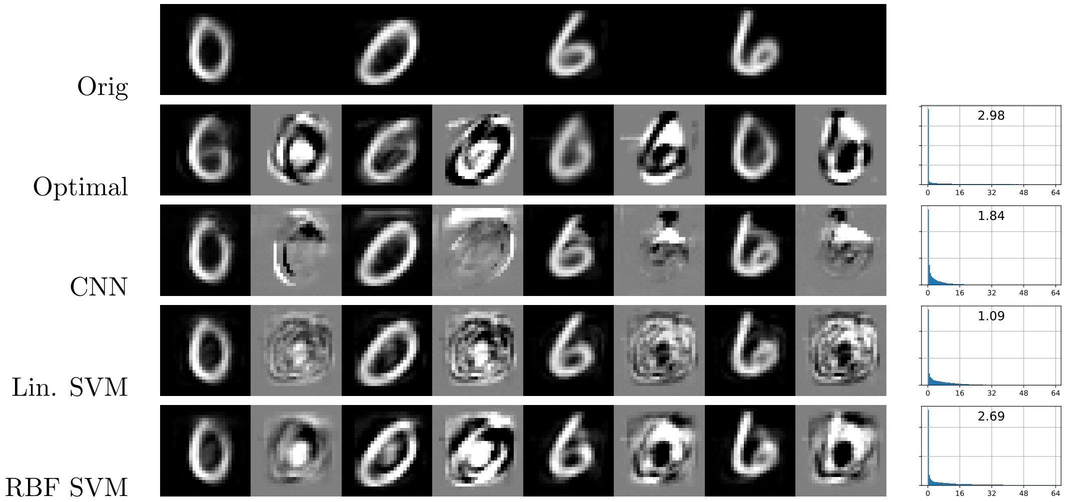


Figure 18: Samples, perturbations and histograms for MNIST digits 0 vs. 6

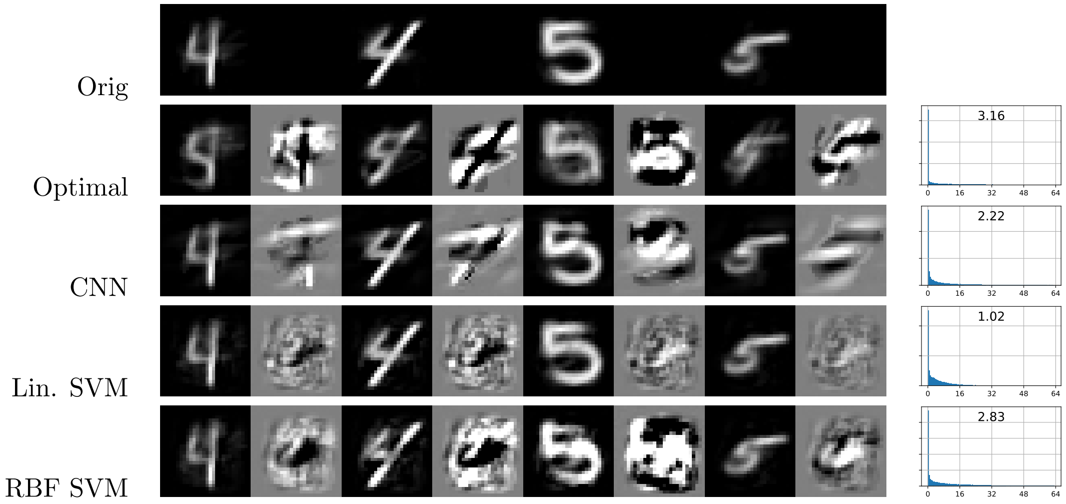


Figure 19: Samples, perturbations and histograms for MNIST digits 4 vs. 5

References

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57, 2017. doi: 10.1109/SP.2017.49. URL `https://doi.org/10.1109/SP.2017.49`.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian J. Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *CoRR*, abs/1902.06705, 2019. URL `http://arxiv.org/abs/1902.06705`.

Dimitrios Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarial risk and robustness: General definitions and implications for the uniform distribution. In *Advances in Neural Information Processing Systems*, pages 10359–10368, 2018.

Elvis Dohmatob. Limitations of adversarial robustness: strong no free lunch theorem. *arXiv preprint arXiv:1810.04065*, 2018.

Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification.* Wiley, New York, 1973.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1178–1187, 2018a.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018b. doi: 10.1007/s10994-017-5663-3. URL `https://doi.org/10.1007/s10994-017-5663-3`.

Nicolas Ford, Justin Gilmer, and Ekin D. Cubuk. Adversarial examples are a natural consequence of test error in noise, 2019. URL `https://openreview.net/forum?id=S1xoy3CcYX`.

Zoubin Ghahramani, Geoffrey E Hinton, et al. The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. In *International Conference on Learning Representations*, 2018.

Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66, June 2018. ISSN 0001-0782. doi: 10.1145/3134599. URL `http://doi.acm.org/10.1145/3134599`.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct): 1391–1415, 2004.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

Susmit Jha, Uyeong Jang, Somesh Jha, and Brian Jalaian. Detecting adversarial examples using data manifolds. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 547–552. IEEE, 2018.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SJeLIgBKPS`.

David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4536–4543, 2019.

Preetum Nakkiran. A discussion of 'adversarial examples are not bugs, they are features': Adversarial examples are just bugs, too. *Distill*, 2019. doi: 10.23915/distill.00019.5. https://distill.pub/2019/advex-bugs-discussion/response-5.

Nicolas Papernot, Ian Goodfellow, Ryan Sheatsley, Reuben Feinman, and Patrick McDaniel. cleverhans v1. 0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 10, 2016.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct): 2825–2830, 2011.

Tomaso Poggio, Kenji Kawaguchi, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Xavier Boix, Jack Hidary, and Hrushikesh Mhaskar. Theory of deep learning iii: explaining the non-overfitting puzzle, 2017.

Eitan Richardson and Yair Weiss. On gans and gmms. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 5852–5863, 2018. URL http://papers.nips.cc/paper/7826-on-gans-and-gmms.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems 31*, volume abs/1804.11285, pages 5014–5026, 2018.

L. Schott, J. Rauber, W. Brendel, and M. Bethge. Towards the first adversarially robust neural network model on mnist. In *International Conference on Learning Representations (ICLR)*, May 2019. URL https://arxiv.org/pdf/1805.09190.pdf.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2019.

Adi Shamir, Itay Safran, Eyal Ronen, and Orr Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance. *CoRR*, abs/1901.10861, 2019. URL http://arxiv.org/abs/1901.10861.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19:70:1–70:57, 2018. URL http://jmlr.org/papers/v19/18-188.html.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Thomas Tanay and Lewis D. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *CoRR*, abs/1608.07690, 2016. URL http://arxiv.org/abs/1608.07690.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings* *of the 36th International Conference on Machine Learning*, volume 97, pages 7472–7482, 2019.

Yonggang Zhang, Xinmei Tian, Ya Li, Xinchao Wang, and Dacheng Tao. Principal component adversarial example. *IEEE Transactions on Image Processing*, 29:4804–4815, 2020.