Robustness of Bayesian Neural Networks to Gradient-Based Attacks

Ginevra Carbone*
Department of Mathematics and Geosciences
University of Trieste, Trieste, Italy
ginevra.carbone@phd.units.it

Matthew Wicker*
Departement of Computer Science
University of Oxford,
Oxford, United Kingdom
matthew.wicker@wolfson.ox.ac.uk

Luca Laurenti
Departement of Computer Science,
University of Oxford,
Oxford, United Kingdom
luca.laurenti@cs.ox.ac.uk

Andrea Patane
Departement of Computer Science,
University of Oxford,
Oxford, United Kingdom
patane.andre@gmail.com

Luca Bortolussi
Department of Mathematics and Geosciences
University of Trieste, Trieste, Italy;
Modeling and Simulation Group,
Saarland University, Saarland, Germany
luca.bortolussi@gmail.com

Guido Sanguinetti
School of Informatics, University
of Edinburgh, Edinburgh, United Kingdom;
SISSA, Trieste, Italy
gsanguin@inf.ed.ac.uk

Abstract

Vulnerability to adversarial attacks is one of the principal hurdles to the adoption of deep learning in safety-critical applications. Despite significant efforts, both practical and theoretical, the problem remains open. In this paper, we analyse the geometry of adversarial attacks in the large-data, overparametrized limit for Bayesian Neural Networks (BNNs). We show that, in the limit, vulnerability to gradient-based attacks arises as a result of degeneracy in the data distribution, i.e., when the data lies on a lower-dimensional submanifold of the ambient space. As a direct consequence, we demonstrate that in the limit BNN posteriors are robust to gradient-based adversarial attacks. Experimental results on the MNIST and Fashion MNIST datasets, representing the finite data regime, with BNNs trained with Hamiltonian Monte Carlo and Variational Inference support this line of argument, showing that BNNs can display both high accuracy and robustness to gradient based adversarial attacks.

1 Introduction

Adversarial attacks are small, potentially imperceptible pertubations of test inputs that can lead to catastrophic misclassifications in high-dimensional classifiers such as deep Neural Networks (NN). Since the seminal work of Szegedy et al. [2013], adversarial attacks have been intensively studied, and even state-of-the-art deep learning models, trained on very large data sets, have been shown to be susceptible to such attacks [Goodfellow et al., 2014]. In the absence of effective defenses, the widespread existence of adversarial examples has raised serious concerns about the security and robustness of models learned from data [Biggio and Roli, 2018]. As a consequence, the development of machine learnig models that are robust to adversarial perturbations is an essential pre-condition for

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

their application in safety-critical scenarios, where model failures have already led to fatal accidents [Yadron and Tynan, 2016].

Many attack strategies are based on identifying directions of high variability in the loss function by evaluating gradients w.r.t. input points (see, e.g., Goodfellow et al. [2014], Madry et al. [2017]). Since such variability can be intuitively linked to uncertainty in the prediction, Bayesian Neural Networks (BNNs) [Neal, 2012] have been recently suggested as a more robust deep learning paradigm, a claim that has found some empirical support [Feinman et al., 2017, Gal and Smith, 2018, Bekasov and Murray, 2018, Liu et al., 2018]. However, neither the source of this robustness, nor its general applicability are well understood mathematically.

In this paper we show a remarkable property of BNNs: in a suitably defined large data limit, we prove that the gradients of the expected loss function of a BNN w.r.t. the input points vanish. Our analysis shows that adversarial attacks for deterministic NNs in the large data limit arise necessarily from the low dimensional support of the data generating distribution. By averaging over nuisance dimensions, BNNs achieve zero expected gradient of the loss and are thus provably immune to gradient-based adversarial attacks.

We experimentally support our theoretical findings on various BNN architectures trained with Hamiltonian Monte Carlo (HMC) and with Variational Inference (VI) on both MNIST and Fashion MNIST data sets, empirically showing that the magnitude of the gradients decreases as more samples are taken from the BNN posterior. We also test this decreasing effect when approaching towards the overparametrized case on the Half Moons dataset. We experimentally show that two popular gradient-based attack strategies for attacking NNs are unsuccessful on BNNs. Finally, we conduct a large-scale experiment on thousands of different networks, showing that for BNNs high accuracy correlates with high robustness to gradient-based adversarial attacks, contrary to what observed for deterministic NNs trained via standard Stochastic Gradient Descent (SGD) [Su et al., 2018].

In summary, this paper makes the following contributions:

- A theoretical framework to analyse adversarial robustness of BNNs in the large data limit.
- A proof that, in this limit, the posterior average of the gradients of the loss function vanishes, providing robustness against gradient-based attacks.
- Large-scale experiments, showing empirically that BNNs are robust to gradient-based attacks and can resist the well known accuracy-robustness trade-off.[1]

Related Work The robustess of BNNs to adversarial examples has been already observed by Gal and Smith [2018], Bekasov and Murray [2018]. In particular, in [Bekasov and Murray, 2018] the authors define Bayesian adversarial spheres and empirically show that, for BNNs trained with HMC, adversarial examples tend to have high uncertanity, while in [Gal and Smith, 2018] sufficient conditions for idealised BNNs to avoid adversarial examples are derived. However, it is unclear how such conditions could be checked in practice, as it would require one to check that the BNN architecture is invariant under all the symmetries of the data.

Empirical methods to detect adversarial examples for BNNs that utilise pointwise uncertainty have been introduced in [Li and Gal, 2017, Feinman et al., 2017, Rawat et al., 2017]. However, most of these approaches have largely relied on Monte Carlo dropout for posterior inference [Carlini and Wagner, 2017]. Statistical techniques for the quantification of adversarial robustness of BNNs have been introduced by [Cardelli et al., 2019a] and employed in [Michelmore et al., 2019] to detect erroneous behaviours in the context of autonomous driving. Furthermore, in [Ye and Zhu, 2018] a Bayesian approach has been considered in the context of adversarial training, where the authors showed improved performances with respect to other, non-Bayesian, adversarial training approaches.

2 Bayesian Neural Networks and Adversarial Attacks

Bayesian modelling aims to capture the intrinsic epistemic uncertainty of data driven models by defining ensembles of predictors [Barber, 2012]; it does so by turning algorithm parameters (and consequently predictions) into random variables. In the NN scenario, for a NN $f(\mathbf{x}, \mathbf{w})$ with input $\mathbf{x}$ and network parameters (weights) $\mathbf{w}$, one starts with a prior measure over the network weights $p(\mathbf{w})$

[1] The code for the experiments can be found at: `https://github.com/ginevracoal/robustBNNs`.

[Neal, 2012]. The fit of the network with weights $\mathbf{w}$ to the data D is assessed through the likelihood $p(D|\mathbf{w})$ [Bishop, 2006]. Bayesian inference then combines likelihood and prior via Bayes theorem to obtain a *posterior* measure on the space of weights $p(\mathbf{w}|D) \propto p(D|\mathbf{w})\, p(\mathbf{w})$.

Maximising the likelihood function w.r.t. the weights $\mathbf{w}$ is in general equivalent to minimising the loss function in standard NNs; indeed, standard training of NNs can be viewed as an approximation to Bayesian inference which replaces the posterior distribution with a delta function at its mode. Obtaining the posterior distribution exactly is impossible for non-linear/non-conjugate models such as NNs. Asymptotically exact samples from the posterior distribution can be obtained via procedures such as Hamiltonian Monte Carlo (HMC) [Neal et al., 2011], while approximate samples can be obtained more cheaply via Variational Inference (VI) [Blundell et al., 2015]. Irrespective of the posterior inference method of choice, Bayesian predictions at a new input $\mathbf{x}^*$ are obtained from an ensemble of n NNs, each with its individual weights drawn from the posterior distribution $p(\mathbf{w}|D)$:

$$f(\mathbf{x}^*|D) = \langle f(\mathbf{x}^*, \mathbf{w}) \rangle_{p(\mathbf{w}|D)} \simeq \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}^*, \mathbf{w}_i) \qquad \mathbf{w}_i \sim p(\mathbf{w}|D) \tag{1}$$

where $\langle \cdot \rangle_p$ denotes expectation w.r.t. the distribution p. The ensemble of NNs yields the *predictive distribution* of the BNN.

Given an input point $\mathbf{x}^*$ and a strength (i.e. maximum perturbation magnitude) $\epsilon > 0$, the worst-case adversarial perturbation can be defined as the point around $\mathbf{x}^*$ that maximises the loss function $L(\tilde{\mathbf{x}}, \mathbf{w})$[2]:

$$\tilde{\mathbf{x}} := \underset{\tilde{\mathbf{x}}: ||\tilde{\mathbf{x}} - \mathbf{x}^*|| \leq \epsilon}{\arg\max} \langle L(\tilde{\mathbf{x}}, \mathbf{w}) \rangle_{p(\mathbf{w}|D)}.$$

If the network prediction on $\tilde{\mathbf{x}}$ differs from the original prediction on $\mathbf{x}^*$, then we call $\tilde{\mathbf{x}}$ an *adversarial example*. As $f(\mathbf{x}, \mathbf{w})$ is non-linear, computing $\tilde{\mathbf{x}}$ is a non-linear optimisation problem for which several approximate solution methods have been proposed and among them, gradient-based attacks are arguably the most prominent [Biggio and Roli, 2018]. In particular, the Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2014] is among the most commonly employed attacks and works by approximating $\tilde{\mathbf{x}}$ by taking an ϵ-step in the direction of the sign of the gradient at $\mathbf{x}$. In the context of BNNs, where attacks are against the predictive distribution of Eqn (1), FGSM becomes

$$\tilde{\mathbf{x}} \simeq \mathbf{x} + \epsilon \operatorname{sgn}\left(\langle \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}) \rangle_{p(\mathbf{w}|D)} \right) \simeq \mathbf{x} + \epsilon \operatorname{sgn}\left(\sum_{i=1}^{n} \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}_i) \right) \tag{2}$$

where the final expression is a Monte Carlo approximation with samples $\mathbf{w}_i$ drawn from the posterior $p(\mathbf{w}|D)$. The expressions for the Projected Gradient Descent method (PGD) [Madry et al., 2017] or other gradient-based attacks are analogous. While the results discussed in Section 3 hold for any gradient-based method, in the experiments reported in Section 5 we focus on the Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2014] and the Projected Gradient Descent method (PGD) [Madry et al., 2017].

3 Adversarial robustness of Bayesian predictive distributions

Equation (2) suggests a possible explanation for the observed robustness of BNNs to adversarial attacks: the averaging under the posterior might lead to cancellations in the final expectation of the gradient. It turns out that this averaging property is intimately related to the geometry of the so called *data manifold* $\mathcal{M}_D \subset \mathbb{R}^d$, i.e. the support of the data generating distribution $p(D)$. The key result that we leverage is a recent breakthrough [Du et al., 2018, Rotskoff and Vanden-Eijnden, 2018, Mei et al., 2018] which proved global convergence of (stochastic) gradient descent (at the distributional level) in the overparametrised, large data limit. Precise definitions can be found in the original publications and in the supplementary material. In our setting, a *fully trained, overparametrized BNN* is an ensemble of NNs satisfying the conditions in [Rotskoff and Vanden-Eijnden, 2018] and at full convergence of the training algorithm, hence they all coincide on the data manifold, but can differ outside of it. We now state our main result whose full proof is in the supplementary material:

[2]For simplicity we omit the dependence of the loss from the ground truth.

Theorem 1. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized BNN on a prediction problem with data manifold $\mathcal{M}_D \subset \mathbb{R}^d$ and posterior weight distribution $p(\mathbf{w}|D)$. Assuming $\mathcal{M}_D \in \mathcal{C}^\infty$ almost everywhere, in the large data limit we have a.e. on $\mathcal{M}_D$*

$$\left(\langle \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}) \rangle_{p(\mathbf{w}|D)}\right) = \mathbf{0}. \tag{3}$$

By the definition of the FGSM attack (Equation (2)) and other gradient-based attacks, Theorem 1 directly implies that any gradient-based attack will be ineffective against a BNN in the limit. The theorem is proved by first showing that in a fully trained BNN in the large data limit, the gradient of the loss is orthogonal to the data manifold (Lemma 1 and Corollary 1), then proving a symmetry property of a fully trained BNN with an uninformative prior, guaranteeing that the orthogonal component of the gradient cancels out in expectation with respect to the BNN posterior (Lemma 2).

Dimensionality of the data manifold To investigate the effect of dimensionality of the data manifold on adversarial examples, we start from the following

Lemma 1. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized NN on a prediction problem with a.e. smooth data manifold $\mathcal{M}_D \subset \mathbb{R}^d$. Let $\mathbf{x}^* \in \mathcal{M}_D$ s.t. $B_d(\mathbf{x}^*, \epsilon) \subset \mathcal{M}_D$, with $B_d(\mathbf{x}^*, \epsilon)$ being the d-dimensional ball centred at $\mathbf{x}^*$ of radius ϵ for some $\epsilon > 0$. Then $f(\mathbf{x}, \mathbf{w})$ is robust to gradient-based attacks at $\mathbf{x}^*$ of strength $\leq \epsilon$ (i.e. restricted in $B_d(\mathbf{x}^*, \epsilon)$).*

This is a trivial consequence of an important result proved in [Du et al., 2018, Rotskoff and Vanden-Eijnden, 2018, Mei et al., 2018]: at convergence, overparametrised NNs provably achieve zero loss on the whole data manifold $\mathcal{M}_D$ in the infinite data limit, which implies that the function f would be locally constant at $\mathbf{x}^*$. A corollary of Lemma 1 is

Corollary 1. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized NN on a prediction problem with data manifold $\mathcal{M}_D \subset \mathbb{R}^d$ smooth a.e. (where the measure is given by the data distribution $p(D)$). If f is vulnerable to gradient-based attacks at $x^* \in \mathcal{M}_D$ in the infinite data limit, then a.s. $\dim(\mathcal{M}_D) < d$ in a neighbourhood of x^*.*

This corollary confirms the widely held conjecture that adversarial attacks may originate from degeneracies of the data manifold [Goodfellow et al., 2014, Fawzi et al., 2018]. In fact, it had been already empirically noticed [Khoury and Hadfield-Menell, 2018] that adversarial perturbations often arise in directions which are normal to the data manifold. The higher the codimension of the data manifold into the embedding space, the more it is likely to select random directions which are normal to it. The suggestion that lower-dimensional data structures might be ubiquitous in NN problems is also corroborated by recent results [Goldt et al., 2019] showing that the characteristic training dynamics of NNs are intimately linked to data lying on a lower-dimensional manifold. Notice that the implication is only one way; it is perfectly possible for the data manifold to be low dimensional and still not vulnerable at many points.

Notice that the assumption of smoothness a.e. for the data manifold is needed to avoid pathologies in the data distribution (e.g. its support being a closed but dense subset of $\mathbb{R}^d$). Additionally, this assumption guarantees that the dimensionality of $\mathcal{M}_D$ is locally constant. A consequence of Corollary 1 is that $\forall \mathbf{x} \in \mathcal{M}_D$ the gradient of the loss function is orthogonal to the data manifold as it is zero along the data manifold, i.e., $\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}) = \nabla_{\perp \mathbf{x}} L(\mathbf{x}, \mathbf{w})$, where $\nabla_{\perp \mathbf{x}}$ denotes the gradient projected into the normal subspace of $\mathcal{M}_D$ at $\mathbf{x}$.

Bayesian averaging of normal gradients In order to complete the proof of Theorem 1, we therefore need to show that the normal gradient has expectation zero under the posterior distribution

$$\nabla_{\perp \mathbf{x}} \langle L(\mathbf{x}, \mathbf{w}) \rangle_{p(\mathbf{w}|D)} = 0.$$

The key to this result is the fact that, assuming an uninformative prior[3] on the weights $\mathbf{w}$, all NNs that agree on the data manifold will by definition receive the same posterior weight in the ensemble, since they achieve exactly the same likelihood. Therefore, it remains to be proved the following symmetry of the normal gradient at almost any point $\hat{\mathbf{x}} \in \mathcal{M}_D$:

[3]Both a uniform distribution and a wide Gaussian distribution act as uninformative priors.

Lemma 2. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized NN on a prediction problem on data manifold $\mathcal{M}_D \subset \mathbb{R}^d$ a.e. smooth. Let $\hat{\mathbf{x}} \in \mathcal{M}_D$ to be the perturbed input and let the normal gradient at $\hat{\mathbf{x}}$ be $\mathbf{v}_{\mathbf{w}}(\hat{\mathbf{x}}) = \nabla_{\perp \hat{\mathbf{x}}} L(\mathbf{x}, \mathbf{w})$ be different from zero. Then, in the infinite data limit and for almost all $\hat{\mathbf{x}}$, there exists a set of weights $\mathbf{w}'$ such that*

$$f(\mathbf{x}, \mathbf{w}') = f(\mathbf{x}, \mathbf{w}) \text{ a.e. in } \mathcal{M}_D, \tag{4}$$

$$\nabla_{\perp \hat{\mathbf{x}}} L(\hat{\mathbf{x}}, \mathbf{w}') = -\mathbf{v}_{\mathbf{w}}(\hat{\mathbf{x}}). \tag{5}$$

The proof of this lemma rests on constructing a function satisfying (4) and (5) by suitably perturbing locally the fully trained NN $f(\mathbf{x}, \mathbf{w})$, i.e. by adding a function ϕ which is zero on the data manifold and enforces condition (5) on $\hat{\mathbf{x}}$. Since we are in the overparametrized, large data limit, any such function will be realisable as a NN with suitable weights choice $\mathbf{w}'$.

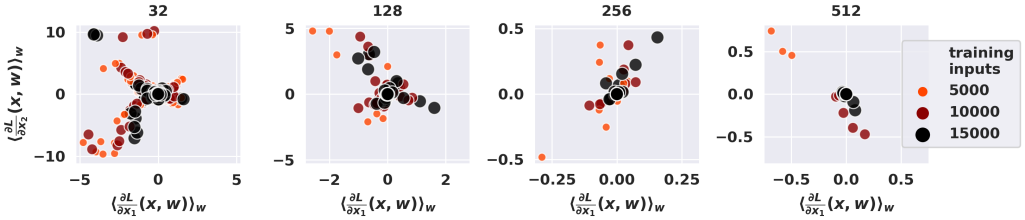

Figure 1: Expected loss gradients components on 100 two-dimensional test points from the Half Moons dataset [Rozza et al., 2014] (both partial derivatives of the loss function are shown). Each dot represents a different NN architecture. We used a collection of HMC BNNs, by varying the number of hidden neurons and training points. Only models with test accuracy greater than 80% were taken into account. We refer the reader to the supplementary material for training hyperparameters.

4 Consequences and limitations

Theorem 1 has the natural consequence of protecting BNNs against all gradient-based attacks, due to the vanishing average of the expectation of the gradients in the limit. Its proof also sheds light on a number of observations made in recent years. Before moving on to empirically validating the theorem, it is worth reflecting on some of its implications and limitations:

- Theorem 1 holds in a specific thermodynamic limit, however we expect the averaging effect of BNN gradients to still provide considerable protection in conditions where the network architecture and the data amount lead to high accuracy and strong expressivity. In practice, high accuracy might be a good indicator of robustness for BNNs. In Figure 1, we examine the impact of the assumptions made in Theorem 1 by exploiting a setting in which we have access to the data-generating distribution, the half-moons dataset [Rozza et al., 2014]. We show that the magnitude of the expectation of the gradient shrinks as we increase the network's parameters and the number of training inputs.
- Theorem 1 holds when the ensemble is drawn from the true posterior; nevertheless it is not obvious (and likely not true) that the posterior distribution is the sole ensemble with the zero averaging property of the gradients. Cheaper approximate Bayesian inference methods which retain ensemble predictions such as VI may in practice provide good protection.
- Theorem 1 is proven under the assumption of uniform priors; in practice, (vague) Gaussian priors are more frequently used for computational reasons. Once again, unless the priors are too informative, we do not expect a major deviation from the idealised case.
- Gaussian Processes [Williams and Rasmussen, 2006] are equivalent to infinitely wide BNNs and therefore constitute overparametrized BNNs by definition (although scaling their training to the large data limit might be problematic). Theorem 1 provides theoretical backing to recent empirical observations of their adversarial robustness [Blaas et al., 2019, Cardelli et al., 2019b].
- While the Bayesian posterior ensemble may not be the only randomization to provide protection, it is clear that some simpler randomizations such as bootstrap will be ineffective, as noted

empirically in [Bekasov and Murray, 2018]. This is because bootstrap resampling introduces variability along the data manifold, rather than in orthogonal directions. In this sense, the often repeated mantra that bootstrap is an approximation to Bayesian inference is strikingly inaccurate when the data distribution has zero measure support. Similarly, we do not expect gradient smoothing approaches to be successful [Athalye et al., 2018], since the type of smoothing performed by Bayesian inference is specifically informed by the geometry of the data manifold.

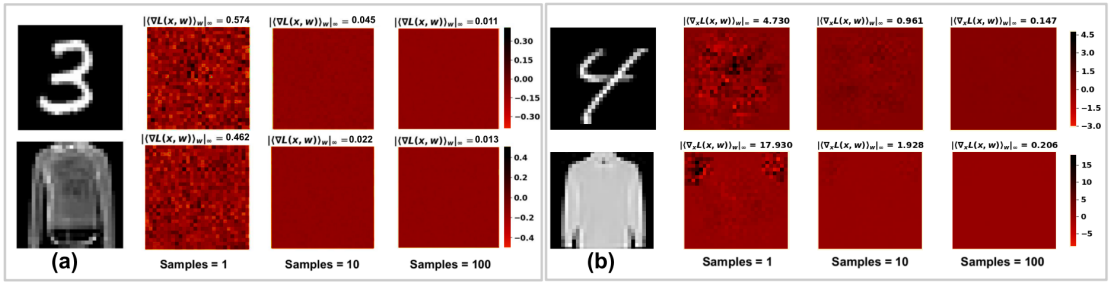

Figure 2: The expected loss gradients of BNNs exhibit a vanishing behaviour when increasing the number of samples from the posterior predictive distribution. We show example images from MNIST (top row) and Fashion MNIST (bottom row) and their expected loss gradients wrt networks trained with HMC (left) and VI (right). To the right of the images we plot a heat map of gradient values.

5 Empirical Results

In this section we empirically investigate our theoretical findings on different BNNs. We train a variety of BNNs on the MNIST and Fashion MNIST [Xiao et al., 2017] datasets, and evaluate their posterior distributions using HMC and VI approximate inference methods. In Section 5.1, we experimentally verify the validity of the zero-averaging property of gradients implied by Theorem 1, and discuss its implications on the behaviours of FGSM and PGD attacks on BNNs in Section 5.2. In Section 5.3 we analyse the relationship between robustness and accuracy on thousands of different NN architectures, comparing the results obtained by Bayesian and by deterministic training. Details on the experimental settings and BNN training parameters can be found in the Supplementary Material.

5.1 Evaluation of the Gradient of the Loss for BNNs

We investigate the vanishing behavior of input gradients - established by Theorem 1 for the thermodynamic limit regime - in the finite, practical settings, that is with a finite number of training data and with finite-width BNNs. Specifically, we train a two hidden layers BNN (with 1024 neurons per layer for a total of about 1.8 million parameters) with HMC and a three hidden layers BNN (512 neurons per layer) with VI. These achieve approximately 95% test accuracy on MNIST and 89% on Fashion MNIST when trained with HMC; as well as 95% and 92%, respectively, when trained with VI. More details about the hyperparameters used for training can be found in the Supplementary Material.

Figure 2 depicts anecdotal evidence on the behaviour of the component-wise expectation of the loss gradient as more samples from the posterior distribution are incorporated into the BNN predictive distribution. Similarly to how in Figure 1 for the half-moons dataset we observe that the gradient of the loss goes to zero when increasing number of training points and number of parameters, here we have that, as the number of samples taken from the posterior distribution of $\mathbf{w}$ increases, all the components of the gradient approach zero. Notice that the gradient of the individual NNs (that is those with just one sampled weight), is far away from being zero. As shown in Theorem 1, it is only through the Bayesian averaging of ensemble predictor that the gradients cancel out.

This is confirmed in Figure 3, where we provide a systematic analysis of the aggregated gradient convergence properties on 1k test images for MNIST and Fashion-MNIST. Each dot shown in the plots represents a component of the expected loss gradient from each one of the images, for a total of 784k points used to visually approximate the empirical distribution of the component-wise expected loss gradient. For both HMC and VI the magnitude of the gradient components drops as the number

of samples increases, and tends to stabilize around zero already with 100 samples drawn from the posterior distribution, suggesting that the conditions laid down in Theorem 1 are approximately met by the BNN analysed here. Notice that the gradients computed on HMC trained networks drops more quickly toward zero. This is in accordance to what is discussed in Section 4, as VI introduces additional approximations in the Bayesian posterior computation.

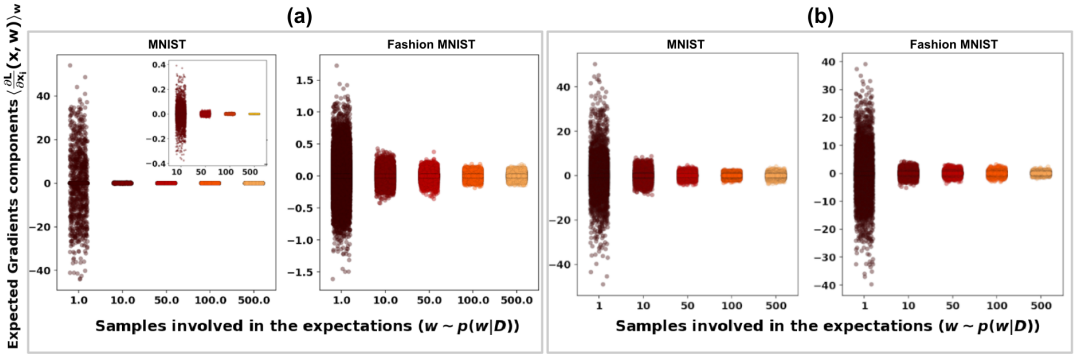

Figure 3: The components of the expected loss gradients approach zero as the number of samples from the posterior distribution increases. For each number of samples, the figure shows 784 gradient components for 1k different test images, from both the MNIST and Fashion MNIST datasets. The gradients are computed on HMC (a) and VI (b) trained BNNs.

5.2 Gradient-Based Attacks for BNNs

The fact that gradient cancellation occurs in the limit does not directly imply that BNN predictions are robust to gradient-based attacks in the finite case. For example, FGSM attacks are crafted such that the direction of the manipulation is given only by the sign of the expectation of the loss gradient and not by its magnitude. Thus, even if the entries of the expectation drop to an infinitesimal magnitude but maintains a meaningful sign, then FGSM could potentially produce effective attacks. In order to test the implications of vanishing gradients on the robustness of the posterior predictive distribution against gradient-based attacks, we compare the behaviour of FGSM and PGD[4] attacks to a randomly devised attack. Namely, the random attack mimics a randomised version of FGSM in which the sign of the attack is sampled at random. In practice, we perturb each component of a test image by a random value in $\{-\epsilon, \epsilon\}$. In Table 1 we compare the effectiveness of FGSM, PGD and of the random attack. We report the adversarial accuracy (i.e. probability that the network returns the ground truth label for the perturbed input) for 500 images. For each image, we compute the expected gradient using 250 posterior samples. The attacks were performed with $\epsilon = 0.1$. In almost all cases, we see that the random attack outperforms the gradient-based attacks, showing how the vanishing behaviour of the gradient makes FGSM and PGD attacks uneffective. For all attacks we used the categorical cross-entropy loss function which is related to the likelihood used during training. Furthermore, in Table 2 in the Supplementary we also run the same evaluation for when the same network employed in Table 1 is trained with SGD and deep ensembles. In both cases both FGSM and PGD are effective, suggesting how simply model averaging and mini-batches are not enough to achieve a robust model.

Dataset/Method	Rand	FGSM	PGD
MNIST/HMC	**0.850**	0.960	0.970
MNIST/VI	0.956	**0.936**	0.938
Fashion/HMC	**0.812**	0.848	0.826
Fashion/VI	**0.744**	0.834	0.916

Table 1: Adversarial robustness of BNNs trained with HMC and VI with respect to the *random attack* (Rand), FGSM and PGD.

[4] with 15 iterations and 1 restart.

5.3 Robustness Accuracy Analysis in Deterministic and Bayesian Neural Networks

In Section 4, we noticed that as a consequence of Theorem 1, high accuracy might be related to high robustness to gradient-based attacks for BNNs. Notice, that this would run counter to what has been observed for deterministic NNs trained with SGD [Su et al., 2018]. In this section, we look at an array of more than 1000 BNNs with different hyperparameters trained with HMC and VI on MNIST and Fashion-MNIST.[5] We experimentally evaluate their accuracy/robustness trade-off on FGSM attacks as compared to that obtained with deterministic NNs trained via SGD based methods. For the robustness evaluation we consider the average difference in the softmax prediction between the original test points and the crafted adversarial input, as this provides a quantitative and smooth measure of adversarial robustness that is closely related with mis-classification ratios [Cardelli et al., 2019a]. That is, for a collection of N test point, we compute $\frac{1}{N}\sum_{j=1}^{N} |\langle f(\mathbf{x}_j, \mathbf{w})\rangle_{p(\mathbf{w}|D)} - \langle f(\tilde{\mathbf{x}}_j, \mathbf{w})\rangle_{p(\mathbf{w}|D)}|_\infty$.

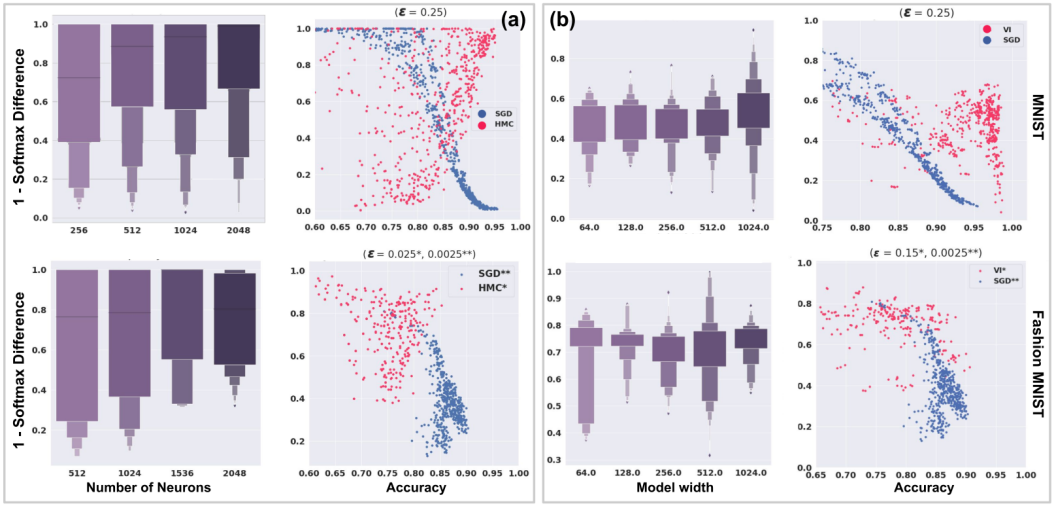

Figure 4: Robustness-Accuracy trade-off on MNIST (first row) and Fashion MNIST (second row) for BNNs trained with HMC (a), VI (b) and SGD (blue dots). While a trade-off between accuracy and robustness occur for deterministic NNs, experiments on HMC show a positive correlation between accuracy and robustness. The boxplots show the correlation between model capacity and robustness. Different attack strength (ϵ) are used for the three methods accordingly to their average robustness.

The results of the analysis are plotted in Figure 4 for MNIST and Fashion MNIST. Each dot in the scatter plots represents the results obtained for each specific network architecture trained with SGD (blue dots), HMC (pink dots in plots (a)) and VI (pink dots in plots (b)). As already reported in the literature [Su et al., 2018] we observe a marked trade-off between accuracy and robustness (i.e., 1 - softmax difference) for high-performing deterministic networks. Interestingly, this trend is fully reversed for BNNs trained with HMC (plots (a) in Figure 4) where we find that as networks become more accurate, they additionally become more robust to FGSM attacks as well. We further examine this trend in the boxplots that represent the effect that the network width has on the robustness of the resulting posterior. We find the existence of an increasing trend in robustness as the number of neurons in the network is increased. This is in line with our theoretical findings, i.e., as the BNN approaches the over-parametrised limit, the conditions for Theorem 1 are approximately met and the network is protected against gradient-based attacks. On the other hand, the trade-off behaviours are less obvious for the BNNs trained with VI and on Fashion-MNIST. In particular, in plot (b) of Figure 4 we find that, similarly to the deterministic case, also for BNNs trained with VI, robustness seems to have a negative correlation with accuracy. Furthermore, for VI we observe that there is some trend dealing with the size of the model, but we only observe this in the case of VI trained on MNIST where it can be seen that model robustness may increase as the width of the layers increases, but this can also lead to poor robustness (which may be indicative of mode collapse).

[5]Details on the NN architectures used can be found in the Supplementary Material.

6 Conclusions

The quest for robust, data-driven models is an essential component towards the construction of AI-based technologies. In this respect, we believe that the fact that Bayesian ensembles of NNs can evade a broad class of adversarial attacks will be of great relevance. While promising, this result comes with some significant limitations. First and foremost, performing Bayesian inference in large non-linear models is extremely challenging. While in our experiments cheaper approximations such as VI also enjoyed a degree of adversarial robustness, albeit reduced, there are no guarantees that this will hold in general. To this end, we hope that this result will spark renewed interest in the pursuit of efficient Bayesian inference algorithms. Secondly, our theoretical results hold in a thermodynamic limit which is never realised in practice. More worryingly, we currently have no rigorous diagnostics to understand how near we are to the limit case, and can only reason about this empirically. We notice here that several other studies [Bekasov and Murray, 2018, Li and Gal, 2017, Feinman et al., 2017, Rawat et al., 2017] have focused on pointwise uncertainty to detect adversarial behaviour; while this does not appear relevant in the limit scenario, it might be a promising indicator of robustness in finite data conditions. Thirdly, we have focused on two attack strategies which directly utilise gradients in our empirical evaluation. More complex gradient-based attacks, such as [Carlini and Wagner, 2016, Papernot et al., 2017, Moosavi-Dezfooli et al., 2016], as well as non-gradient based/ query-based attacks, also exist [Ilyas et al., 2018, Wicker et al., 2018]. Evaluating the robustness of BNNs against these attacks would also be interesting.

Finally, the proof of our main result highlighted a profound connection between adversarial vulnerability and the geometry of data manifolds; it was this connection that enabled us to show that principled randomisation might be an effective way to provide robustness in the high dimensional context. We hope that this connection will inspire novel algorithmic strategies which can offer adversarial protection at a cheaper computational cost.

7 Broader Impact

This work is a theoretical investigation in the large data limit of vulnerability of Bayesian Neural Networks to gradient-based attacks. The main result is that, in this limit, BNNs are not vulnerable to such attacks, as the input gradient vanishes in expectation. This advancement provides a theoretically-provable rational for selecting BNNs in applications where there is concern about attackers performing fast, gradient-based attacks. However, it does not provide any guarantee on the actual safety of BNNs trained on a finite amount of data. Our work may positively benefit the study of adversarial robustness for BNNs and the investigation of properties that make these networks less vulnerable than deterministic ones. These features could then potentially be transferred to other network paradigms and lead to greater robustness of machine learning algorithms in general. However, there may still exist different attacks leading BNNs to misclassifications and our contribution does not provide any defence technique against them.

In the last few years adversarial examples have presented a major hurdle to the adoption of AI systems in any security related field, whose applications go from self-driving vehicles to medical diagnoses. Machine learning algorithms show remarkable performance and generalization capabilities, but they also manifest weaknesses that are not consistent with human understanding of the world. Ultimately, the lack of knowledge about the difference between human and machine interpretation of reality leads to an issue of public trust. The development of procedures that are robust to changes in the output and that represent calibrated uncertainty, would inherently be more trust-worthy and allow for wide-spread adoption of deep learning in safety and security critical tasks.

8 Funding Disclosure

This project was partially funded by the EU's Horizon 2020 program under the Marie Sklodowska-Curie grant agreement No 722022 "AffecTec" and by the Italian PRIN project "SEDUCE" No 2017TWRCNB.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Artur Bekasov and Iain Murray. Bayesian adversarial spheres: Bayesian inference and adversarial examples in a noiseless setting. *arXiv preprint arXiv:1811.12335*, 2018.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Arno Blaas, Luca Laurenti, Andrea Patane, Luca Cardelli, Marta Kwiatkowska, and Stephen Roberts. Robustness quantification for classification with gaussian processes. *arXiv preprint arXiv:1905.11876*, 2019.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Andrea Patane, and Matthew Wicker. Statistical guarantees for the robustness of bayesian neural networks. *arXiv preprint arXiv:1903.01980*, 2019a.

Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, and Andrea Patane. Robustness guarantees for bayesian inference with gaussian processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7759–7768, 2019b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2016.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. In *Advances in Neural Information Processing Systems*, pages 1178–1187, 2018.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Yarin Gal and Lewis Smith. Sufficient conditions for idealised models to have no adversarial examples: a theoretical and empirical study with bayesian neural networks. *arXiv preprint arXiv:1806.00667*, 2018.

Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modelling the influence of data structure on learning in neural networks. *arXiv preprint arXiv:1909.11500*, 2019.

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

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *CoRR*, abs/1811.00525, 2018. URL `http://arxiv.org/abs/1811.00525`.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2052–2061. JMLR. org, 2017.

Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. *arXiv preprint arXiv:1810.01279*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Rhiannon Michelmore, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. *arXiv preprint arXiv:1909.09884*, 2019.

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

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

Ambrish Rawat, Martin Wistuba, and Maria-Irina Nicolae. Adversarial phenomenon in the eyes of bayesian deep learning. *arXiv preprint arXiv:1711.08244*, 2017.

Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

Alessandro Rozza, Mario Manzo, and Alfredo Petrosino. A novel graph-based fisher kernel method for semi-supervised learning. In *Proceedings of the 2014 22nd International Conference on Pattern Recognition*, ICPR '14, page 3786–3791, USA, 2014. IEEE Computer Society. ISBN 9781479952090. doi: 10.1109/ICPR.2014.650. URL `https://doi.org/10.1109/ICPR.2014.650`.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

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

Matthew Wicker, Xiaowei Huang, and Marta Kwiatkowska. Feature-guided black-box safety testing of deep neural networks. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 408–426. Springer, 2018.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

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

Danny Yadron and Dan Tynan. Tesla driver dies in first fatal crash while using autopilot mode. *the Guardian*, 1, 2016.

Nanyang Ye and Zhanxing Zhu. Bayesian adversarial learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6892–6901. Curran Associates Inc., 2018.

Additional theory background and proofs

9 Global convergence of over-parameterised DNNs

We briefly recapitulate here the main results on global convergence of over-parameterised neural networks [Du et al., 2018, Mei et al., 2018, Rotskoff and Vanden-Eijnden, 2018]. We follow more closely the notation of Rotskoff and Vanden-Eijnden [2018] and reference to that paper for more formal proofs and definitions.

The setup of the problem is as follows: we are using a NN $f(\mathbf{x}, \mathbf{w})$ to approximate a function $\tilde{f}(\mathbf{x})$. The target function is observed at points drawn from a data distribution $p(D)$ while the weights of the NN are drawn from a measure $\mu(\mathbf{w})$. The support of the data distribution $p(D)$ is the *data manifold* $\mathcal{M}_D \subset \mathbb{R}^d$. The discrepancy between the observed target function and the approximating function is measured through a suitable loss function $L(\mathbf{x}, \mathbf{w})$ which needs to be a convex function of the difference between observed and predicted values (e.g. squared loss for regression or cross-entropy loss for classification).

These results require a set of technical but rather standard assumptions on the target function and the NN units (assumptions 2.1-2.4 and 3.1 in Rotskoff and Vanden-Eijnden [2018]), which we recall here for convenience:

- The input and feature space are closed Riemannian manifolds, and the NN units are differentiable.
- The unit is *discriminating*, i.e. if it integrates to zero when multiplied by a function g for all values of the weights, then $g = 0$ a.e. .
- The network is sufficiently expressive to be able to represent the target function.
- The distribution of the data input is not degenerate (Assumption 3.1).

One can then prove the following results:

- The loss function is a convex functional of the measure on the space of weights.
- Training a NN (with a finite number of units/ weights) by gradient descent approximates a gradient flow in the space of measures. Therefore, by the Law of Large Numbers, gradient descent on the exact loss (infinite data limit) converges to the global minimum (constant zero loss) when the number of hidden units grows to infinity (overparameterised limit).
- Stochastic gradient descent also converges to the global minimum under the assumption that every minibatch consists of novel examples.

In other words, Rotskoff and Vanden-Eijnden [2018] show that the celebrated Universal Approximation Theorem of Cybenko [1989] is realised dynamically by stochastic gradient descent in the infinite data/ overparameterised limit.

10 Proofs of technical results

We provide here additional details for the theoretical results in the main text. We will assume that the assumptions of Rotskoff and Vanden-Eijnden [2018] hold, as recalled in Section 9.

Lemma 3. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized NN on a prediction problem with a.e. smooth data manifold $\mathcal{M}_D \subset \mathbb{R}^d$. Let $\mathbf{x}^* \in \mathcal{M}_D$ s.t. $B_d(\mathbf{x}^*, \epsilon) \subset \mathcal{M}_D$, with $B_d(\mathbf{x}^*, \epsilon)$ the d-dimensional ball centred at $\mathbf{x}^*$ of radius ϵ for some $\epsilon > 0$. Then $f(\mathbf{x}, \mathbf{w})$ is robust to gradient-based attacks at $\mathbf{x}^*$ of strength $\leq \epsilon$ (i.e. restricted in $B_d(\mathbf{x}^*, \epsilon)$).*

Proof. By the results of Rotskoff and Vanden-Eijnden [2018], Mei et al. [2018], Du et al. [2018] we know that, in the large data limit, an overparametrized NN will achieve zero loss on the data manifold once fully trained. By assumption, the data manifold contains a whole open ball centred at $\mathbf{x}^*$, so the loss will be constant (and zero) in an open neighbourhood of $\mathbf{x}^*$. Consequently, the loss gradient at $\mathbf{x}^*$ will be zero in a whole open neighbourhood of $\mathbf{x}^*$; therefore, any attack based on moving the input point in the direction of the gradient at $\mathbf{x}^*$ or a nearby point (such as PGD) will fail to change the input and consequently fail to change the output value, thus guaranteeing robustness. □

Corollary 2. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized NN on a prediction problem with data manifold $\mathcal{M}_D \subset \mathbb{R}^d$ smooth a.e. (where the measure is given by the data distribution $p(D)$). If f is vulnerable to gradient-based attacks at $\mathbf{x}^* \in \mathcal{M}_D$ in the infinite data limit, then a.s. $\dim(\mathcal{M}_D) < d$ in a neighbourhood of $\mathbf{x}^*$.*

Proof. If f is vulnerable at $\mathbf{x}^*$ to gradient based attacks, then the gradient of the loss at $\mathbf{x}^*$ must be non-zero. By Lemma 1 we know that, if the data manifold $\mathcal{M}_D$ has locally dimension d, then the gradient has to be zero. Hence $\dim(\mathcal{M}_D) < d$ in a neighbourhood of $\mathbf{x}^*$. □

Lemma 4. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized NN on a prediction problem on data manifold $\mathcal{M}_D \subset \mathbb{R}^d$ a.e. smooth. Let $\hat{\mathbf{x}} \in \mathcal{M}_D$ to be attacked and let the normal gradient at $\hat{\mathbf{x}}$ be $\mathbf{v}_{\mathbf{w}}(\hat{\mathbf{x}}) := \nabla_{\perp \mathbf{x}} L(\hat{\mathbf{x}}, \mathbf{w})$ be different from zero. Then, in the infinite data limit and for almost all $\hat{\mathbf{x}}$, there exists a set of weights $\mathbf{w}'$ such that*

$$f(\mathbf{x}, \mathbf{w}') = f(\mathbf{x}, \mathbf{w}) \text{ a.e. in } \mathcal{M}_D,$$

$$\nabla_{\perp \mathbf{x}} L(\hat{\mathbf{x}}, \mathbf{w}') = -\mathbf{v}_{\mathbf{w}}(\hat{\mathbf{x}}).$$

Proof. By assumption, the function $f(\mathbf{x}, \mathbf{w})$ is realisable by the NN and therefore differentiable. To show that there exists (at least) one set of weights that lead to a function satisfying the constraints in (4) and (5), we proceed by steps. First, we observe that the loss is a functional over functions $g \colon \mathcal{M}_D \to [0, 1]$, given explicitly by

$$L[g] = \int_{\mathcal{M}_D} dq \sum_y p(y|\theta) \log g(\theta)$$

where θ is a parametrisation on $\mathcal{M}_D$, and we have written the data generating distribution $p(D) = p(y|\theta)q(\theta)$ as the product of the distribution of input values times the class conditional distribution. However, evaluating the loss over a function $\phi : \mathbb{R}^d \to [0, 1]$ defined over the whole ambient space only makes sense if one also defines a projection from the ambient space into the data manifold. It is however still possible, given a function defined over the whole ambient space, to define the loss computed on its restriction over $\mathcal{M}_D$ and the normal gradient to the manifold by using the ambient space metric and the decomposition it induces of the tangent space into directions along $\mathcal{M}_D$ and directions orthogonal. Normal derivatives of $L[\phi(\mathbf{x})]$ can then be defined as standard. For any function $\phi(\mathbf{x})$ on $\mathcal{M}_D$ the normal gradient of the loss function[6] is

$$\nabla_{\perp \mathbf{x}} L(\phi(\mathbf{x})) = \frac{\delta L(\phi)}{\delta \phi} \nabla_{\perp \mathbf{x}} \phi(\mathbf{x})$$

Assuming the functional derivative of the loss is a differentiable function, as is the case e.g. with cross-entropy, then condition 5 can be rewritten as

$$h(\phi(\hat{\mathbf{x}}), \nabla_{\perp \mathbf{x}} \phi(\hat{\mathbf{x}})) = 0 \tag{6}$$

for a suitably smooth function h.

To construct a function ϕ that satisfies both conditions (4) and (5), we assume that the data manifold admits smooth local coordinates in an open ball $\mathcal{M}_D \cap B_d(\hat{\mathbf{x}}, \epsilon)$ of radius ϵ centred at $\hat{\mathbf{x}}$ (which is true for almost all points by assumption). We then define $\phi(\mathbf{x}) = f(\mathbf{x}, \mathbf{w}) + g(\mathbf{x})$, where $g(\mathbf{x})$ is smooth, supported in $B_d(\hat{\mathbf{x}}, \epsilon)$ and zero on the boundary of the ball $\partial B_d(\hat{\mathbf{x}}, \epsilon)$, and $g(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \mathcal{M}_D \cap B_d(\hat{\mathbf{x}}, \epsilon)$. Therefore, ϕ satisfies condition (4) by construction. In particular we can impose condition (4) on g in the local coordinates around $\hat{\mathbf{x}}$, by using a slice chart on $\mathcal{M}_D \cap B_d(\hat{\mathbf{x}}, \epsilon)$.

In the overparametrized limit, it will always be possible to approximate the resulting function ϕ by choosing suitable weights $\mathbf{w}'$ for the NN, thus proving the Lemma. Notice that condition 6 holds on a fixed point $\hat{\mathbf{x}}$ under attack, hence at different attack points we may in principle have different w' satisfying the lemma.

□

[6] Notice that this is only defined on the data manifold $\mathcal{M}_D$, while $\mathbf{x}$ is a coordinate system in the ambient space $\mathbb{R}^d$.

11 Comparison with Deep Ensembles

Deep ensembles, as proposed by Lakshminarayanan et al. [2017], are an ensemble of neural networks trained from different randomly selected initial conditions, which are then averaged in order to make a prediction. In Table 2 we consider the same network used to perform the experiments in Section 5.2 (hyper-parameters are reported in Table 4) and run a comparison with both deterministic NNs and deep ensembles. As expected, Bayesian NNs are more robust than deterministic ones. Moreover, we find that deep ensembles and deterministic NNs are comparable in terms of robustness, suggesting that simply averaging predictions for different weight initialization and mini-batching is not enough to achieve a robust model.

Model	**Test accuracy**	**FGSM accuracy**	**PGD accuracy**
Deterministic NN	97.69	21.19	1.45
Ensemble NN	99.4	20.6	0.3
Bayesian NN	96.1	90.0	89.8

Table 2: FGSM and PGD attacks on the network employed in Section 5.2. We compare a deterministic NN, a deep ensemble NN (of size 100), and a BNN (trained with VI). Attacks are performed on 1k test points from the MNIST dataset. We observe that VI trained network achieve better robustness against PGD and FGSM.

12 Training hyperparameters for BNNs

Half moons grid search	
Posterior samples	{250}
HMC warmup samples	{100, 200, 500}
Training inputs	{5000, 10000, 15000}
Hidden size	{32, 128, 256, 512}
Nonlinear activation	Leaky ReLU
Architecture	2 fully connected layers

Figure 5: Hyperparameters for training BNNs in Figure 1

Training hyperparameters for HMC		
Dataset	MNIST	Fashion MNIST
Training inputs	60k	60k
Hidden size	1024	1024
Nonlinear activation	ReLU	ReLU
Architecture	Fully Connected	Fully Connected
Posterior Samples	500	500
Numerical Integrator Stepsize	0.002	0.001
Number of steps for Numerical Integrator	10	10

Table 3: Hyperparameters for training BNNs using HMC in Figures 2 and 3.

Training hyperparameters for VI

Dataset	MNIST	Fashion MNIST
Training inputs	60k	60k
Hidden size	512	1024
Nonlinear activation	Leaky ReLU	Leaky ReLU
Architecture	Convolutional	Convolutional
Training epochs	5	10
Learning rate	0.01	0.001

Table 4: Hyperparameters for training BNNs using VI in Figures 2 and 3.

HMC MNIST/Fashion MNIST grid search

Posterior samples	{250, 500, 750*}
Numerical Integrator Stepsize	{0.01, 0.005*, 0.001, 0.0001}
Numerical Integrator Steps	{10*, 15, 20}
Hidden size	{128, 256, 512*}
Nonlinear activation	{relu*, tanh, sigmoid}
Architecture	{1*,2,3} fully connected layers

Table 5: Hyperparameters for training BNNs with HMC in Figure 4. * indicates the parameters used in Table 1 of the main text.

SGD MNIST/Fashion MNIST grid search

Learning Rate	{0.001*}
Minibatch Size	{128, 256*, 512, 1024}
Hidden size	{64, 128, 256, 512, 1024*}
Nonlinear activation	{relu*, tanh, sigmoid}
Architecture	{1*,2,3} fully connected layers
Training epochs	{3,5*,7,9,12,15} epochs

Table 6: Hyperparameters for training BNNs with SGD in Figure 4. * indicates the parameters used in Table 1 of the main text.

SGD MNIST/Fashion MNIST grid search

Learning Rate	{0.001, 0.005, 0.01, 0.05}
Hidden size	{64, 128, 256, 512}
Nonlinear activation	{relu, tanh, sigmoid}
Architecture	{2, 3, 4, 5} fully connected layers
Training epochs	{5, 10, 15, 20, 25} epochs

Table 7: Hyperparameters for training BNNs with SGD in Figure 4.