Reducing Exploitation of Data Idiosyncrasy Helps Robustify Trained Models

Xindi Wu[1] **Haohan Wang**[1] **Eric Zelikman**[2] **Eric Xing**[1] **Min Xu**[1]
[1]Carnegie Mellon University [2]Stanford University

Abstract

In this paper, we analyze the trade-off between robustness and accuracy in neural networks as a function of a model's ability to *exploit data idiosyncrasy*, that is, superficial representations (which are imperceptible to humans) specific to samples' distribution. Further, our analysis enables simple methods for improving the robustness of a neural network against adversarial examples by perturbing weights to reduce model dependency on data idiosyncrasy. While the improvement of robustness usually comes with minor degeneration of prediction accuracy, as expected by our theoretical study, our method improves the robustness of neural networks' after the models are already trained. As a result, this improvement in robustness comes with marginal computational cost.

Introduction

Deep learning has achieved impressive, often superhuman, empirical predictive *accuracy* on a variety of tasks, such as object detection (He et al. 2015), speech recognition (Xiong et al. 2016), and numerous biological challenges (Yue and Wang 2018). Yet, a closer look into deep learning methods usually reveals that neural networks' *robustness* to imperceptible perturbations is far below human level (Szegedy et al. 2013; Rosenfeld, Zemel, and Tsotsos 2018; Wang, Sun, and Xing 2019), indicating that neural networks' ability to automatically generalize "semantic" information (as humans perceive data) may have been over-estimated.

With respect to accuracy, deep learning models can achieve almost perfect training *accuracy* on datasets even when corresponding labels are shuffled (Zhang et al. 2017), which indicates that neural networks can view data with much higher granularity than humans. This disparity was further highlighted by (Jo and Bengio 2017), demonstrating the neural networks' tendency to capture information through textural information other than "semantic" information. With respect to robustness, dedicated designed subtle changes in data that are imperceptible to humans (*i.e.* adversarial examples) can easily demonstrate the lack of robustness of an otherwise-accurate model (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015). This topic has historically alternated between authors defending models against adversarial examples (Cisse et al. 2017; Madry et al. 2018; Liao et al. 2018; Wong and Kolter 2018) and others proposing new attack manners that expose models' new weakness (Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2017; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Papernot et al. 2016; Carlini and Wagner 2017b). With the attackers winning this back-and-forth by increasingly large margins, some researchers have become concerned that the existence of adversarial examples may be inevitable (Shafahi et al. 2019).

In this paper, we explain the phenomenon of adversarial examples as a direct outcome of deep learning's capacity to view data with higher granularity than humans, which we refer as *exploiting data idiosyncrasy*. Considering this capacity, we study deep learning behaviors as a combination of semantically significant features and data idiosyncrasy, and use this to help explain the trade-off between prediction *accuracy* and *robustness* demonstrated with concrete examples by Tsipras et al. (2019). Further, this regime inspires straightforward methods to improve the *robustness* of a neural network after the model is trained. Specifically, the contribution of this paper can be summarized as:

- We set-up a generalization regime considering a machine learning model's ability to *exploit data idiosyncrasy*.
- On the theoretical side, this regime enables formal discussions of a given model's trade-off between its prediction *accuracy* and *robustness*. The formal discussion also trivially leads to methods that can help improves trained model's *robustness* as a trade-off of its *accuracy*.
- As an application of our regime, we propose three simple methods for improving robustness and demonstrate their efficacy with experiments. Our methods are lightweight and do not require the computational effort of training/fine-tuning a model. Using these, we improve the *robustness* of a trained AlexNet with minimal losses in *accuracy*.

Related Work

The robustness of many machine learning algorithms has been studied, including neural networks (Bishop 1995), regularized regression (El Ghaoui and Lebret 1997; Xu, Caramanis, and Mannor 2009a), and SVMs (Xu, Caramanis, and Mannor 2009b). In recent years, this topic has become particularly popular due to the phenomenon of the existance of adversarial examples (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2017; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Cisse et al. 2017; Carlini and Wagner 2017a; 2017b; Madry et al. 2018; Xu, Evans, and Qi 2017; Liao et al. 2018; Wu et al. 2018; Guo et al. 2018). This naturally leads to the question, *are adversarial examples inevitable?*

Shafahi et al. (2019) argued that adversarial examples are unavoidable. While it is impossible to analyze all real-world data distributions, their paper's empirical results suggest that common distributions in nature lend themselves to adversarial examples. Their argument clashes with a body of work aiming to propose methods that can certify the *robustness* of neural networks (Wong and Kolter 2018; Raghunathan, Steinhardt, and Liang 2018; Sinha, Namkoong, and Duchi 2018; Wong et al. 2018). However, while achieving *robustness*, these methods usually see a slight drop of prediction *accuracy* (Wong and Kolter 2018), leading to another interesting question: are *accuracy* and *robustness* compatible? (Rozsa, Günther, and Boult 2016) demonstrated that more accurate models tend to be more robust on a set of vision models, but later, with a more systematic study, (Hendrycks and Dietterich 2019) showed that the seemingly increased *robustness* was skewed by the increased overall *accuracy*, and more *accurate* vision models (*e.g.* VGG, ResNet) actually have larger drops in performance when presented with adversarial examples. Recently, (Wang et al. 2019b) showed that high-frequency components of images can used in adversarial attacks, further indicating a trade-off between a model's *robustness* and *accuracy*.

There is also a proliferation of works trying to understand the behavior of neural networks with regard to *robustness*. For example, (Sanyal, Kanade, and Torr 2018) showed that representations with a low-rank structure tend to be more *robust*. (Novak et al. 2018) related the *robustness* of a neural network to its "input-output Jacobian", which means the expectation of the magnitudes of the network's output variations over random input perturbations, supporting the arguments in (Sokolić et al. 2017). In a recent brief note, (Nakkiran 2019) argued that the *robustness* of a model may only be achievable via sophisticated designs, which could be understood as arguing that human-level *robustness* needs to be achieved by human-level granularity of perceiving data.

Key difference: This paper aims to extend the discussion of (Tsipras et al. 2019) in the trade-off between a model's *robustness* and *accuracy* to a more general setting that does not rely on specific data distributions. Our argument relies on the key assumption that the cause of the unsatisfying *robustness* of neural networks is the perceptional disparity between humans and models, which is related to (Nakkiran 2019).

Generalization with Data Idiosyncrasy

We first introduce the notations used in this paper: $f(\cdot;\mathbf{\Theta})$ denotes a classifier (*e.g.* a deep learning model) whose parameters are denoted as $\mathbf{\Theta}$, and $\mathbf{\Theta}_{[.]}$ denotes that the model $\mathbf{\Theta}$ operates on data $\cdot$ (*i.e.*, Model $\mathbf{\Theta}$ is trained with data $\cdot$); we use $\mathcal{H}$ to denote a human model, and as a result, $f(\cdot;\mathcal{H})$ denotes how human will classify the data $\cdot$.

$l(\cdot,\cdot)$ denotes a generic loss function (*e.g.* cross entropy loss or MSE loss); $\alpha(\cdot,\cdot)$ denotes a generic evaluation metric (*e.g.* prediction accuracy). Throughout this paper, $\alpha(\cdot,\cdot)$ evaluates prediction accuracy unless specified otherwise.

$\langle\mathbf{X},\mathbf{y}\rangle$ denotes the raw data and corresponding labels, and $\langle\mathbf{x},y\rangle$ denotes a data sample. We use $\langle\mathbf{X}^{\text{train}},\mathbf{y}^{\text{train}}\rangle$, $\langle\mathbf{X}^{\text{val}},\mathbf{y}^{\text{val}}\rangle$, and $\langle\mathbf{X}^{\text{test}},\mathbf{y}^{\text{test}}\rangle$ to denote training, validation, and test data, respectively. We use $\langle\mathbf{X}^{\text{adv}},\mathbf{y}^{\text{test}}\rangle$ to denote the generated adversarial examples as a result of perturbing testing data set, and we use $\mathbf{X}^{\text{adv}}(\mathbf{\Theta})$ to denote that the adversarial examples are generated while the attacking methods are applied to Model $\mathbf{\Theta}$.

We follow (Tsipras et al. 2019), but instead of constructing explicit features, we assume the raw data $\mathbf{X}=\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}+\mathbf{X}_{\mathcal{S}}$, where $\mathbf{X}_{\mathcal{G}}$ denotes semantic information conveyed by the data (*e.g.* the parts of an image that are perceptible to humans), $\mathbf{X}_{\mathcal{D}}$ denotes the information that is statistically associated with the label, but not semantically meaningful to humans (*e.g.* background bias of the images)[1], and $\mathbf{X}_{\mathcal{S}}$ denotes the remaining non-predictive variation (i.e. noise).

These narrative descriptions of $\mathbf{X}_{\mathcal{G}}$, $\mathbf{X}_{\mathcal{D}}$, and $\mathbf{X}_{\mathcal{S}}$ are sufficient for this paper's discussion, but we also offer concrete definitions to make our paper more complete: with the help of Model $\mathcal{H}$, we can define $\mathbf{X}_{\mathcal{G}}$, $\mathbf{X}_{\mathcal{D}}$, and $\mathbf{X}_{\mathcal{S}}$ as follows.

$$
\begin{aligned}
\mathbf{X}_{\mathcal{G}} &:= \{\mathbf{x}_{\mathcal{G}} \,|\, \mathbf{x}_{\mathcal{G}} = \arg\max_{\mathbf{x}'} ||\mathbf{x}-\mathbf{x}'|| \\
&\qquad\qquad \text{s.t.} \quad f(\mathbf{x}';\mathcal{H}) = f(\mathbf{x};\mathcal{H})\} \\
\mathbf{X}_{\mathcal{D}} &:= \{\mathbf{x}_{\mathcal{D}} \,|\, \mathbf{x}_{\mathcal{D}} = \arg\max_{\mathbf{x}'} ||\mathbf{x}-\mathbf{x}_{\mathcal{G}}-\mathbf{x}'|| \\
&\qquad\qquad \text{s.t.} \quad \mathbf{x}' = \arg\min_{\mathbf{x}'} \sum_{\mathbf{x}'} l(f(\mathbf{x}_{\mathcal{G}}+\mathbf{x}';\mathbf{\Theta}),y), \forall\mathbf{\Theta}\} \\
\mathbf{X}_{\mathcal{S}} &:= \{\mathbf{x}_{\mathcal{S}} \,|\, \mathbf{x}_{\mathcal{S}} = \mathbf{x}-\mathbf{x}_{\mathcal{G}}-\mathbf{x}_{\mathcal{D}}\}
\end{aligned} \tag{1}
$$

We do not specify the choice of norms for the purpose of a generic discussion, because adversarial attacks and model *robustness* can be defined over different norms. We refer to $\mathbf{X}_{\mathcal{D}}$ and $\mathbf{X}_{\mathcal{S}}$ as data idiosyncrasy.

Several related assumptions are:

A1: For a Model $\mathbf{\Theta}$, we have:

$$\alpha(f(\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}};\mathbf{\Theta}),\mathbf{y}) > \alpha(f(\mathbf{X}_{\mathcal{G}};\mathbf{\Theta}),\mathbf{y}) \tag{2}$$

which can be intuitively understood as, there exists some association between $\mathbf{X}_{\mathcal{D}}$ and $\mathbf{y}$ that cannot be described by $\mathbf{X}_{\mathcal{G}}$ and $\mathbf{y}$. This assumption can be verified by empirical observations such as (Jo and Bengio 2017; Wang et al. 2019b).

A2: $||\mathbf{x}_{\mathcal{D}}|| \lll ||\mathbf{x}_{\mathcal{G}}||$ and $||\mathbf{x}_{\mathcal{S}}|| \lll ||\mathbf{x}_{\mathcal{G}}||$, which can intuitively understood as the magnitude of $\mathbf{X}_{\mathcal{D}}$ and $\mathbf{X}_{\mathcal{S}}$ are negligible. We believe we can safely assume so because both $\mathbf{X}_{\mathcal{D}}$ and $\mathbf{X}_{\mathcal{S}}$ are imperceptible to humans.

With testing data $\langle\mathbf{X}^{\text{test}},\mathbf{y}^{\text{test}}\rangle$, the *accuracy* of the model $\mathbf{\Theta}$ is denoted as:

$$\alpha(f(\mathbf{X}^{\text{test}};\mathbf{\Theta}),\mathbf{y}^{\text{test}}) \tag{3}$$

[1] one good illustrative example might be the "wearing glasses" signal discussed in Fig.1 in (Wang et al. 2017)

and we consider the following definition of the *accuracy*-independent *robustness*:

$$\mathbb{E}_{\mathbf{x}_\epsilon \leq C}[\alpha(f(\mathbf{X}^{\text{test}} + \mathbf{X}_\epsilon; \mathbf{\Theta}), \mathbf{y}^{\text{test}})] - \alpha(f(\mathbf{X}^{\text{test}}; \mathbf{\Theta}), \mathbf{y}^{\text{test}}) \tag{4}$$

where C is the maximal perturbation considered. The evaluation score is upper bounded by 0 and lower bounded by -1. The higher the score is, the more robust the evaluated model is.

Further, we want to emphasize a seemingly under-appreciated point: some literature appears to describe the training process of a neural network as:

$$\mathbf{\Theta} = \arg\min_{\mathbf{\Theta}} l(f(\mathbf{X}^{\text{train}}; \mathbf{\Theta}), \mathbf{y}^{\text{train}}), \tag{5}$$

however, in practice, deep learning models are usually trained with a regularization operating on empirical performance:

$$\begin{aligned} \mathbf{\Theta} &= \arg\max_{\mathbf{\Theta}'} \alpha(f(\mathbf{X}^{\text{val}}; \mathbf{\Theta}'), \mathbf{y}^{\text{val}}) \\ \text{s.t.} \quad \mathbf{\Theta}' &= \arg\min_{\mathbf{\Theta}''} l(f(\mathbf{X}^{\text{train}}; \mathbf{\Theta}''), \mathbf{y}^{\text{train}}) \end{aligned} \tag{6}$$

We define $\mathbf{X}^{\text{train}}$ and $\mathbf{X}^{\text{val}}$ from the same distribution as:

$$||\alpha(f(\mathbf{X}^{\text{train}}_{\mathcal{D}}; \mathbf{\Theta}), \mathbf{y}^{\text{train}}) - \alpha(f(\mathbf{X}^{\text{val}}_{\mathcal{D}}; \mathbf{\Theta}), \mathbf{y}^{\text{val}})|| < \epsilon$$

where ϵ is a small scalar. Intuitively: $\mathbf{X}^{\text{train}}$ and $\mathbf{X}^{\text{val}}$ are from the same distribution/domain means a model $\mathbf{\Theta}$ can learn similar statistical signals from non-semantic components of the data: $\mathbf{X}^{\text{train}}_{\mathcal{D}}$ and $\mathbf{X}^{\text{val}}_{\mathcal{D}}$.

Therefore, when $\mathbf{X}^{\text{train}}$ and $\mathbf{X}^{\text{val}}$ are from the same distribution/domain, Optimization 6 results in the model $\mathbf{\Theta}_{\mathbf{X}} = \mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}$. In other words, the trained model from Optimization 6 learns to exploit $\mathbf{X}_{\mathcal{D}}$, when $\mathbf{X}^{\text{train}}$ and $\mathbf{X}^{\text{val}}$ are from the same distribution/domain.

By considering the data idiosyncrasy, many interesting empirical deep learning results can be straightforwardly explained: the capacity to reduce training error to zero even when the labels are shuffled (Zhang et al. 2017) can be seen as a result of exploiting $\mathbf{X}_{\mathcal{S}}$; the tendency of CNN's to learn superficial statistics (Jo and Bengio 2017) can been seen as a result of exploiting $\mathbf{X}_{\mathcal{D}}$.

As one may expect, the phenomenon of adversarial samples (Szegedy et al. 2013) results from perturbing $\mathbf{X}_{\mathcal{D}}$ that are exploited by the trained deep learning models. Similarly, the performance drop when a well-behaved model is applied out-of-domain (*e.g.* (Rosenfeld, Zemel, and Tsotsos 2018; Wang, Sun, and Xing 2019)) is because cross-domain data does not share the signals of $\mathbf{X}_{\mathcal{D}}$.

As an aside, one may ask why Optimization 6 will prefer to learn signals from $\mathbf{X}_{\mathcal{G}}$ and $\mathbf{X}_{\mathcal{D}}$, instead of memorizing $\mathbf{X}_{\mathcal{S}}$ as a model will do in the label-shuffled case (Zhang et al. 2017). We believe such a preference is due to a neural network's tendency to learn simpler functions. One may refer to relevant discussions such as (Soudry, Hoffer, and Srebro 2017; Neyshabur et al. 2017; Poggio et al. 2017). These discussions are beyond the scope of this paper.

In addition to the above explanation, this new generalization regime allows us to reiterate the main result in (Tsipras et al. 2019). Instead of a discussion relies on the specific design of data, our main result is applicable generally to any data and any model that exploits data idiosyncrasy.

Remark 1. *With Assumptions A1 and A2, for two models $\mathbf{\Theta}_i$ and $\mathbf{\Theta}_j$ with equivalent capacity to apply semantically meaningful relationships, $\mathbf{X}_{\mathcal{G}}$ (i.e. $f(\mathbf{X}_{\mathcal{G}}, \mathbf{\Theta}_i) = f(\mathbf{X}_{\mathcal{G}}, \mathbf{\Theta}_j)$), there is a trade-off between the model's accuracy (as defined in 3) and the model's robustness (as defined in 4, when $C \leq \max_{\mathbf{x}} ||\mathbf{x}_{\mathcal{D}}||$)*

Improving Robustness by Perturbing Weights

In this section, we are interested in improving a model's *robustness* by forcing the model to ignore $\mathbf{X}_{\mathcal{D}}$, which will result in drop of *accuracy*. Within the scope of this paper, we are interested in the methods that operate on trained models by perturbing weights, instead of re-training the model, given the usefulness of improving the *robustness* of an existing model without extensive computational resources.

To proceed with the theoretical discussion, we work on an intermediate target variable $\mathbf{t}$ as a replacement of $\mathbf{y}$ to free our study from cross-entropy loss to the simpler regression loss. One can consider $\mathbf{t}$ as the golden standard logits generated by the last layer of the network that our model is optimized to learn. Despite the simplicity we introduced by studying $\mathbf{t}$, our empirical results presented later indicate the connection between cross-entropy loss with $\mathbf{y}$ and the regression loss with $\mathbf{t}$. The connection is also discussed previously by Bishop (1995), who also first used regression loss for derivations. As we do not assume a specific definition of regression loss, we study both the absolute loss (*i.e.* $||f(\mathbf{X}; \mathbf{\Theta}) - \mathbf{t}||_1^1$) and the MSE loss (*i.e.* $||f(\mathbf{X}; \mathbf{\Theta}) - \mathbf{t}||_2^2$). For K-class classification problem, we use $\mathbf{t}^k$ to denote the k^{th} class regression target, and we use $\mathbf{\Theta}^k$ to denote corresponding model parameters, where $\mathbf{\Theta} = \cup_{k=1}^K \mathbf{\Theta}^k$ and $\cup$ denotes the union operation.

Lemma 1. *With Assumption A2, if we have*

$$\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]} = \arg\min_{\mathbf{\Theta}} \sum_{k=1}^K ||f(\mathbf{X}_{\mathcal{G}} + \mathbf{X}_{\mathcal{D}}; \mathbf{\Theta}^k) - \mathbf{t}^k||_1^1 \tag{7}$$

and

$$\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}]} = \arg\min_{\mathbf{\Theta}} \sum_{k=1}^K ||f(\mathbf{X}_{\mathcal{G}}; \mathbf{\Theta}^k) - \mathbf{t}^k||_1^1, \tag{8}$$

then $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}]}$ is a shrinkage version of $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}$ in terms of shrinking the element-wise ℓ_1 norm of $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}$.

The proof uses the Taylor series to expand $f(\mathbf{X}_{\mathcal{G}} + \mathbf{X}_{\mathcal{D}}; \mathbf{\Theta}^k)$ as powers of $\mathbf{X}_{\mathcal{D}}$ and then the triangle inequality to separate $\mathbf{X}_{\mathcal{D}}$ from the remaining terms. The complete proof is shown in the Appendix.

Lemma 2. *Regarding $\mathbf{X}_{\mathcal{D}}$ as a random variable, with assumptions A2, $\mathbb{E}[\mathbf{X}_{\mathcal{D}}] = 0$, and $\mathbb{E}[\mathbf{X}_{\mathcal{D}}^2] < \infty$, if we have*

$$\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]} = \arg\min_{\mathbf{\Theta}} \sum_{k=1}^K ||f(\mathbf{X}_{\mathcal{G}} + \mathbf{X}_{\mathcal{D}}; \mathbf{\Theta}^k) - \mathbf{t}^k||_2^2$$

and

$$\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}]} = \arg\min_{\mathbf{\Theta}} \sum_{k=1}^K ||f(\mathbf{X}_{\mathcal{G}}; \mathbf{\Theta}^k) - \mathbf{t}^k||_2^2,$$

then $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}]}$ is a shrinkage version of $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}$ in terms of shrinking the Frobenius norm $||\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}||_F$.

The proof is similar to the previous one with an additional step of integrating out the random variable. It is also shown in the Appendix.

The above two lemmas suggest: given a trained model $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}$ from Optimization 6, a more *robust* version of this model can be found as a result of shrinking $\mathbf{\Theta}_{[\mathbf{X}_{\mathcal{G}}+\mathbf{X}_{\mathcal{D}}]}$ following certain manners, but this resulting new model will have lower prediction *accuracy* as it exploits $\mathbf{X}_{\mathcal{D}}$ less.

As our theoretical study is designed for general classifiers and general data, we do not have further theoretical guidance for universally applicable methods that serve as the best shrinking methods to improve the *robustness* of trained models, just as Shafahi et al. (2019) argues: it is impossible to analyze the distribution of real-world data. However, based on our lemmas, we can propose the following simple heuristic methods that perturbs the neural networks' fully connected layer (denoted as θ):

M1: We remove the columns (or rows) *i.e.*, the output (or input) dimension of θ, that are activated with low frequency when training data is passed through the final trained model, resulting in a model with less ℓ_∞ (or ℓ_1) matrix norm, thus, less element-wise ℓ_1 norm, of a fully connected layer (or if the fully connect layer is the uppermost layer)

M2: We remove the trailing singular values of θ by setting them to zeros, producing a model with a reduced Frobenius norm.

M3: We apply both M1 and M2.

There exist several works that regularize a neural network for smaller norms of weights, such as weight decay (Krogh and Hertz 1992; Zhang et al. 2019), regularizing Jacobian matrix (Sokolić et al. 2017), progressive pruning (Guo et al. 2018), or even dropout (Srivastava et al. 2014). In comparison, a distinct advantage of our methods is that we directly work on trained networks, while other regularization methods usually require training the model.

We intentionally prioritize the simplicity of our proposed methods for two reasons: 1) we believe simpler methods tend to have more practical value because they can be more easily used by practitioners with less related experience; 2) as these methods and following experiments mainly serve as verification of our theoretical study, we limit the complexity of our proposed methods to eliminate potential extra influences introduced by sophisticated heuristics.

Experiments

Our experiments serve two goals: 1) to verify the main theoretical argument of this paper: these exists a trade-off between a given neural network's *accuracy* and *robustness*; 2) to demonstrate the effectiveness of our proposed simple methods in improving the *robustness* of a network.

We do not compare our methods to other existing adversarial defense methods for several reasons: 1) the main theoretical argument can be well justified only with comparisons towards the original model; 2) even if our methods result in less robust models than what other adversarial defense methods can achieve, our methods still have the distinct advantages of simplicity. For example, this simplicity allows us to experiment with full ImageNet data set and giant models such as ResNet. As noted by Cohen, Rosenfeld, and Kolter (2019), no other adversarial defense methods have demonstrated effectiveness on the full ImageNet scale.

Robustness Against Adversarial Attacks

Experimental Setup To evaluate the performance of methods M1-M3, we sequentially reduce $p\%$ ($p = 0, 1, 2, \dots 99$) components of weights θ. Specifically: for M1, we discard the columns that are active (have non-zero values) less than $p\%$ of the time when training samples are passed through the model; for M2, we discard the $p\%$ trailing singular values by setting them to zeros and then reconstruct the layer; for M3, we apply M1 and M2 simultaneously.

We consider three attack methods: FGSM (Goodfellow, Shlens, and Szegedy 2014), DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016), and C&W (Carlini and Wagner 2017b). We use the default parameters in Foolbox (Rauber, Brendel, and Bethge 2017). Our experiments show that these default parameters are effective enough in most cases.

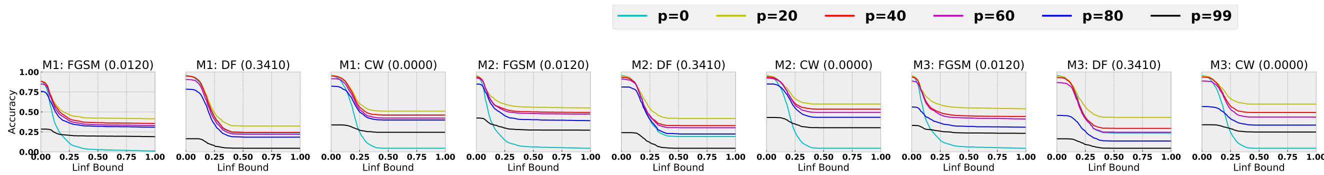
We experiment with three data sets: MNIST (LeCun 1998), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and CIFAR-10 (Krizhevsky and Hinton 2009). We used convolutional neural networks that have been demonstrated with reasonable high testing accuracy in these data sets ($95\%+$ on MNIST, $91\%+$ on Fashion-MNIST, $91\%+$ on CIFAR-10) as baseline model for our experiment.

Robustness Against Adversarial Attacks of the Original Models Our first experiment focuses on the resilience of weights-perturbed network towards the adversarial examples generated according to the original model. We start with a neural network trained according to Optimization 6 with reasonably high validation set accuracy as a Model $\mathbf{\Theta}$ $\langle \mathbf{X}^{\text{test}}, \mathbf{y}^{\text{test}} \rangle$, then we generate the adversarial examples $\langle \mathbf{X}^{\text{adv}}(\mathbf{\Theta}), \mathbf{y}^{\text{test}} \rangle$ according to the trained model, then we apply our method to get a sequence of models $\mathbf{\Theta}_p$ ($p = 1, 2, \dots 99$) and test these models $\mathbf{\Theta}_p$ over the generated adversarial examples.

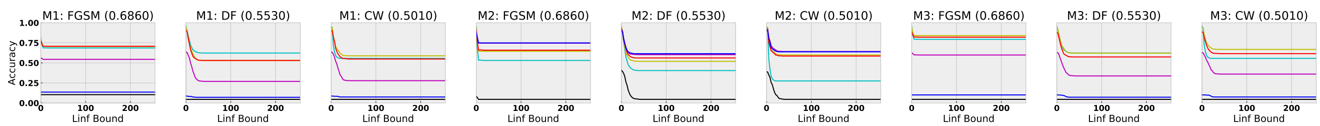
The results are shown in Figure 1. These figures show the curve of prediction accuracy of adversarial examples (Y-axis) over the maximum ℓ_∞-norm perturbation allowed between the adversarial examples and the original image (X-axis). We show the changes of *accuracy* as we increase p for different data/model and different attack methods.

We notice that these figures tend to confirm our main theoretical justification by showing that the drop of a model's *accuracy* resulting from reduced dependence on data idiosyncrasy can result in the improvement of its *robustness*. Remarkably, we notice that a slight sacrifice of the *accuracy* can sometimes lead to huge improvements in the *robustness*.

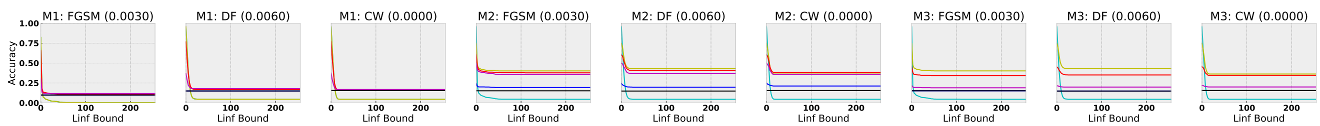
We notice that our methods, in general, behave better against C&W attacks than against other attacks across many of the settings. We believe this is a positive sign as C&W attacks are often regarded as the most powerful attack methods because they search for the perturbation under the constraint that the perturbation will mislead the classifier. According to our generalization regime considering models' abilities in *exploiting data idiosyncrasy*, there are almost no effective



(a) MNIST



(b) Fashion-MNIST



(c) CIFAR-10

Figure 1: Illustration of the accuracy of the models as a function of the bound of various adversarial attacks. Methods M1-M3 reduce the norm of the parameters of the network (by discarding p percentage of weights according to M1-M3). When the methods discard too many elements and become dysfunctional, both *robustness* and *accuracy* drop significantly.

defenses against C&W attacks unless the model discards the information learned through data idiosyncrasy. Therefore, as our methods perturb the weights, we can observe that the evaluated *robustness* increases and the *accuracy* decreases.

Interestingly, we notice that our methods are ineffective against DeepFool attacks in the MNIST and Fashion-MNIST case, but help in the CIFAR-10 case. Although different methods behave differently in these settings, the overall performance supports our main claim made in Remark 1.

Robustness Against New Adversarial Attacks We continue to study whether our simple methods can result in more *robust* models against attacks targeting the new models. For the evaluation, we consider the following metric (similar to NIP2018 adversarial vision challenge):

- Given model $\mathbf{\Theta}_p$, we apply our attack methods to generate adversarial examples $\mathbf{X}^{\text{adv}}(\mathbf{\Theta}_p)$.
- For every sample i, we use our model to predict $\widehat{\mathbf{y}}_i = f(\mathbf{X}_i^{\text{adv}}(\mathbf{\Theta}_p); \mathbf{\Theta}_p)$
- For every sample i, we consider the distance defined as:

$$d_i = \begin{cases} ||\mathbf{X}_i^{\text{adv}}(\mathbf{\Theta_p}) - \mathbf{X}_i^{\text{test}}||_2^2 & \text{if} \quad \widehat{\mathbf{y}}_i \neq y_i^{test} \cap y_i = \mathbf{y}_i^{\text{test}} \\ 0 & \text{otherwise} \end{cases}$$

- We report the mean as all d_i across all the samples as the final testing score. Higher score indicates a better model.

We report our results with this evaluation metric in Table 1 when methods M1-M3 are applied with $p = 1, 2, \ldots, 5$. The scores where our methods improve upon the original scores are shown in bold. Our methods improve the performance in most cases. Interestingly, methods M1-M3 all help the scores on CIFAR-10, and the improvements on M2 and M3 are quite significant in comparison to others. In the MNIST case, both M2 and M3 work well, and M1 shows a trend in improving the performance as p increases. In the Fashion-MNIST case, only M3 improves the performance. Although there are several cases where our methods do not help, we believe the main message of this paper is well justified by these experiments: we can improve the *robustness* of a neural network by lowering the *accuracy* by perturbing the weights, and even straightforward methods such as M1-M3 can achieve the goal.

Robustness Test in Real-world Corruption on ImageNet Data

Now we consider another setting of model *robustness* with the help of ImageNet-C data set introduced by Hendrycks and Dietterich (2019). ImageNet-C is a benchmark data set that is an extension of the popular ImageNet data set (Deng et al. 2009) by introducing a total of 75 sets (15 types $\times$ 5 levels) of corrupted version of ImageNet validation data.

We experiment with two popular network architectures that have been reported with reasonably high accuracy on the original ImageNet data set: AlexNet (Krizhevsky, Sutskever, and Hinton 2012) and ResNet (He et al. 2016). We consider the 18-layer architecture of ResNet, denoted ResNet18. As our methods conveniently allow us to work with pre-trained weights, we begin with the existing weights for AlexNet and ResNet and do not perform further fine-tuning for fair comparison. For AlexNet, our method is applied to the second-to-last layer, and for ResNet, our method is applied to the last layer as this is the only full-connected layer in ResNet.

We first introduce the evaluation metric: with t denoting

Dataset	Method	0%	Score according to percentage of weights perturbed 1%	2%	3%	4%	5%
MNIST	M1	14.2297	**+0.0548**	**+0.0361**	-0.3864	-0.0901	**+0.0254**
	M2		**+0.0889**	**+0.7655**	-0.0112	-0.5397	**+0.0862**
	M3		**+0.5176**	**+0.8347**	-0.2603	-0.3924	-0.0806
Fashion MNIST	M1	297.9973	**+2322.9249**	-64.3923	-71.1064	**+397.8617**	**+59.3002**
	M2		**+376.8122**	**+97.5650**	**+757.4839**	-55.7828	**+591.4694**
	M3		**+326.9057**	**+81.8152**	**+206.2184**	-63.1231	**+430.9850**
Cifar-10	M1	329007.4147	**+3.3899**	0.0000	-0.0034	-93.0458	-191.7609
	M2		**+29531.7244**	**+29531.7244**	**+30348.6489**	**+29551.9851**	**+29572.9443**
	M3		**+29531.7244**	**+29418.0512**	**+29418.0497**	**+29531.7244**	**+29394.3522**

Table 1: The change in the robustness score after perturbing the weights of the original model to various degrees.

the type of corruption and l denoting the level of corruption, we have the corrupted data denoted as $\langle \mathbf{X}^{C}_{t,l}, \mathbf{y}^{\text{test}} \rangle$, as defined by (Hendrycks and Dietterich 2019), the Relative mean Corruption Error $\mathbf{R}m\mathbf{CE}$ of Model $\mathbf{\Theta}$ is:

$$\mathbf{R}m\mathbf{CE}(\mathbf{\Theta}) = \frac{1}{15}\sum_{t=1}^{15}\frac{\delta(\mathbf{\Theta})}{\delta(\mathbf{\Theta}_{\text{AlexNet}})}$$

where

$$\delta(\cdot) = \sum_{l=1}^{5}(\alpha(f(\mathbf{X}^{\text{test}};\cdot), \mathbf{y}^{\text{test}}) - \alpha(f(\mathbf{X}^{C}_{t,l};\cdot), \mathbf{y}^{\text{test}}))$$

With this evaluation metric, the evaluation of model's *robustness* will be independent of the model's *accuracy*.

Further, we report the $\mathbf{R}m\mathbf{CE}(\mathbf{\Theta}_{\text{AlexNet}}) - \mathbf{R}m\mathbf{CE}(\mathbf{\Theta})$ as the measure of *robustness* to center this metric of the baseline model AlexNet to be zero. For the same reason, we report

$$\frac{\alpha(f(\mathbf{X}^{\text{test}};\mathbf{\Theta}), \mathbf{y}^{\text{test}}) - \alpha(f(\mathbf{X}^{\text{test}};\mathbf{\Theta}_{\text{AlexNet}}), \mathbf{y}^{\text{test}})}{\alpha(f(\mathbf{X}^{\text{test}};\mathbf{\Theta}_{\text{AlexNet}}), \mathbf{y}^{\text{test}})}$$

as the measure of *accuracy*.

With our measures of *robustness* and *accuracy*, we can plot the trade-off between *robustness* and *accuracy* of these models in Figure 2, where information of SqueezeNet, VGG11, VGG19, and ResNet50 are from (Hendrycks and Dietterich 2019) for reference. The exact coordinates used to plot the figure are shown in the Appendix.

As we can see, no model is both more *robust* and more *accurate* than AlexNet at the same time. SqueezeNet (Iandola et al. 2016) and VGG (Simonyan and Zisserman 2014) improve upon AlexNet's *accuracy* at a relatively big loss of *robustness*. ResNet50 (He et al. 2016) is likely a preferred model as it increases the *accuracy* by a relatively large margin, but only decreases the *robustness* by a small gap.

Our methods perturb the weights of a model to trade the *accuracy* for *robustness*. Remarkably, we notice that a resulting model AlexNet(M1P30) leads to the improvement of *robustness* with almost no drop of *accuracy*. Thus, AlexNet(M1P30) should be preferred over AlexNet in general, and may also be preferred over ResNet50, depending on the practical needs.

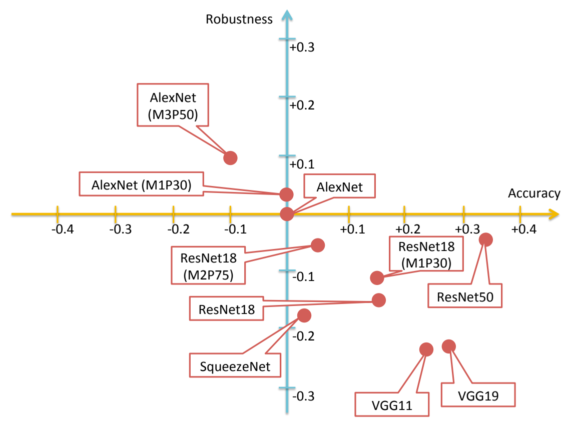


Figure 2: *Robustness-accuracy* trade-off of the vision models and their weight-perturbed versions by our proposed methods: while there is no model that can be simultaneously more (or equally) *robust* and *accurate* than AlexNet, our method generates a version of AlexNet that is more *robust* and almost equally *accurate*.

Discussion

Does this paper suggest we will never have a robust and accurate model? No. While this paper is discusses the existing trade-off between a neural network's *robustness* and *accuracy*, our discussion does not deny the future possibility of a model that is both *robust* and *accurate*, because models can break the prerequisites and assumptions of Remark 1. A crucial assumption to break is that models perceive the data at a different granularity than humans. The trade-off exists as long as the model exploits $\mathbf{X}_{\mathcal{D}}$. Therefore, one future direction is to encourage models to analyze the data at the human level, as argued by (Nakkiran 2019), and another direction is to force the model to discard the information learned while exploiting data idiosyncrasy (Wang et al. 2019a).

The methods we introduced in this paper (M1-M3) are straightforward, as we prioritized simplicity in a theoretical study. We believe more sophisticated methods to reduce the norms of weights after training can lead to more *robust* models with slighter loss of *accuracy*. For example, good empirical performance has been demonstrated on specific applications with methods that remove the weights under the guidance of additional information (Xiao et al. 2016; Wang, Wu, and Xing 2019).

Conclusion

In this paper, we analyzed the implications of data idiosyncrasy as a source of adversarial attack vulnerability. To study this trade-off, we introduced a new generalization regime that considers model's ability to *exploit data idiosyncrasy*, which means the model can learn to utilize the superficial information of data imperceptible to humans, leaving it vulnerable to adversarial attacks. With this regime, we formally demonstrate the *robustness-accuracy* trade-off when a model is trained to *exploit data idiosyncrasy*. Further, our theoretical analysis directly leads to simple methods to improve the model's *robustness* for *accuracy*.

Our experiments support our theoretical argument on the trade-off and also demonstrate the effectiveness of our proposed methods against several adversarial attacks. We apply our methods to improve the *robustness* of AlexNet and ResNet for corrupted ImageNet classification. Remarkably, no models tested (including variations of ResNet, VGG, and SqueezeNet) are simultaneously more *robust* and *accurate* than AlexNet. Our method finds a perturbed version of AlexNet (*i.e.* AlexNet(M1P30)) that is more *robust* and almost as *accurate* as the original AlexNet. We hope that the methods presented in this paper will be used as a low-cost way of increasing robustness of existing models. Ultimately, we believe that this perspective, of data idiosyncrasy as a fundamental challenge to adversarial robustness, can provide an effective framework for developing for robust models in the future.

References

Bishop, C. M. 1995. Training with noise is equivalent to tikhonov regularization. *Neural computation* 7(1):108–116.

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

Carlini, N., and Wagner, D. 2017b. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In Chaudhuri, K., and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 1310–1320. Long Beach, California, USA: PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

El Ghaoui, L., and Lebret, H. 1997. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on matrix analysis and applications* 18(4):1035–1064.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples (2014). In *International Conference on Learning Representations*.

Guo, Y.; Zhang, C.; Zhang, C.; and Chen, Y. 2018. Sparse dnns with improved adversarial robustness. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc. 240–249.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D., and Dietterich, T. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K. 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.

Jo, J., and Bengio, Y. 2017. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Krogh, A., and Hertz, J. A. 1992. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, 950–957.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial examples in the physical world. In *Workshop of International Conference on Learning Representations*.

LeCun, Y. 1998. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Zhu, J.; and Hu, X. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1778–1787.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2574–2582.

Nakkiran, P. 2019. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*.

Neyshabur, B.; Bhojanapalli, S.; McAllester, D.; and Srebro, N. 2017. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, 5947–5956.

Novak, R.; Bahri, Y.; Abolafia, D. A.; Pennington, J.; and Sohl-Dickstein, J. 2018. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, 372–387. IEEE.

Poggio, T.; Kawaguchi, K.; Liao, Q.; Miranda, B.; Rosasco, L.; Boix, X.; Hidary, J.; and Mhaskar, H. 2017. Theory of deep learning iii: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified defenses against adversarial examples.

Rauber, J.; Brendel, W.; and Bethge, M. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

Rosenfeld, A.; Zemel, R.; and Tsotsos, J. K. 2018. The elephant in the room. *arXiv preprint arXiv:1808.03305*.

Rozsa, A.; Günther, M.; and Boult, T. E. 2016. Are accuracy and robustness correlated. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, 227–232. IEEE.

Sanyal, A.; Kanade, V.; and Torr, P. H. 2018. Intriguing properties of learned representations.

Shafahi, A.; Huang, W. R.; Studer, C.; Feizi, S.; and Goldstein, T. 2019. Are adversarial examples inevitable? In *International Conference on Learning Representations*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sinha, A.; Namkoong, H.; and Duchi, J. 2018. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*.

Sokolić, J.; Giryes, R.; Sapiro, G.; and Rodrigues, M. R. 2017. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing* 65(16):4265–4280.

Soudry, D.; Hoffer, E.; and Srebro, N. 2017. The implicit bias of gradient descent on separable data. *arXiv preprint arXiv:1710.10345*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.

Wang, H.; Meghawat, A.; Morency, L.-P.; and Xing, E. P. 2017. Select-additive learning: Improving generalization in multimodal sentiment analysis. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 949–954. IEEE.

Wang, H.; He, Z.; Lipton, Z. C.; and Xing, E. P. 2019a. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*.

Wang, H.; Wu, X.; Yin, P.; and Xing, E. P. 2019b. High frequency component helps explain the generalization of convolutional neural networks. *arXiv preprint arXiv:1905.13545*.

Wang, H.; Sun, D.; and Xing, E. P. 2019. What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language inference tasks. In *Proceedings of AAAI*.

Wang, H.; Wu, Z.; and Xing, E. P. 2019. Removing confounding factors associated weights in deep neural networks improves the prediction accuracy for healthcare applications. Proceedings of 24th Pacific Symposium on Biocomputing (PSB 2019).

Wong, E., and Kolter, Z. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, 5283–5292.

Wong, E.; Schmidt, F.; Metzen, J. H.; and Kolter, J. Z. 2018. Scaling provable adversarial defenses. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc. 8410–8419.

Wu, X.; Jang, U.; Chen, J.; Chen, L.; and Jha, S. 2018. Reinforcing adversarial robustness using model confidence induced by adversarial training. In Dy, J., and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5334–5342. Stockholmsmässan, Stockholm Sweden: PMLR.

Xiao, T.; Li, H.; Ouyang, W.; and Wang, X. 2016. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1249–1258.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xiong, W.; Droppo, J.; Huang, X.; Seide, F.; Seltzer, M.; Stolcke, A.; Yu, D.; and Zweig, G. 2016. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*.

Xu, H.; Caramanis, C.; and Mannor, S. 2009a. Robust regression and lasso. In *Advances in Neural Information Processing Systems*, 1801–1808.

Xu, H.; Caramanis, C.; and Mannor, S. 2009b. Robustness and regularization of support vector machines. *Journal of Machine Learning Research* 10(Jul):1485–1510.

Xu, W.; Evans, D.; and Qi, Y. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

Yue, T., and Wang, H. 2018. Deep learning for genomics: A concise overview. *arXiv preprint arXiv:1802.00810*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Zhang, G.; Wang, C.; Xu, B.; and Grosse, R. 2019. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*.

Appendix

Proof of Lemma 4.1

Inspired by (Bishop 1995), we start with the Taylor series of $f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta})$ in powers of $\mathbf{X}_\mathcal{D}$, which is:

$$\begin{aligned}&f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})\\&=f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})\\&+\mathbf{X}_\mathcal{D}\frac{\partial f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D})}|_{\mathbf{X}_\mathcal{D}=\mathbf{0}}\\&+O(\mathbf{X}_\mathcal{D}^2),\end{aligned}$$

where we can safely discard higher order terms following assumption A2.

Inspired by (Xu, Caramanis, and Mannor 2009), we use triangular inequality to expand the loss

$$\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})-\mathbf{t}^k||_1^1$$

into its upper bound (**Function 1**):

$$\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})-\mathbf{t}^k||_1^1+||\mathbf{X}_\mathcal{D}(\frac{\partial f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G})})||_1^1$$

Thus, comparing the above function to

$$\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}]})-\mathbf{t}^k||_1^1,$$

and notice that $\frac{\partial f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G})}$ denotes $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$ by definition.

Function 1 can be seen as a training process to force the model $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$ to operate *only* on $\mathbf{X}_\mathcal{G}$ (as the model $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}]}$ does) by shrinking the element-wise ℓ_1 norm of $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$.

Further, as $\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}=\cup_{k=1}^K\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$, forcing the model to operate on $\mathbf{X}_\mathcal{G}$ can be achieved by shrinking the element-wise ℓ_1 norm of $\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$

Proof of Lemma 4.2

Inspired by (Bishop 1995), we start with the Taylor series of $f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta})$ in powers of $\mathbf{X}_\mathcal{D}$, which is:

$$\begin{aligned}&f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})\\&=f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})\\&+\mathbf{X}_\mathcal{D}\frac{\partial f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D})}|_{\mathbf{X}_\mathcal{D}=\mathbf{0}}\\&+O(\mathbf{X}_\mathcal{D}^2),\end{aligned}$$

where we can safely discard higher order terms following assumption A2.

Thus, we can expand the loss function

$$\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})-\mathbf{t}^k||_2^2$$

into

$$\begin{aligned}&\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})-\mathbf{t}^k||_2^2\\&+2\mathbf{X}_\mathcal{D}\frac{\partial f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G})}(f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})-\mathbf{t}^k)\\&+\mathbf{X}_\mathcal{D}^2(\frac{\partial f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G})})^2\end{aligned}$$

We then integrate over the random variable $\mathbf{X}_\mathcal{D}$ with the assumption that $\mathbb{E}[\mathbf{X}_\mathcal{D}]=\mathbf{0}$, we have the new form of the loss function: (**Function 2**)

$$\begin{aligned}&\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})-\mathbf{t}^k||_2^2\\&+\mathbb{E}[\mathbf{X}_\mathcal{D}^2](\frac{\partial f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G})})^2\end{aligned}$$

Thus, comparing the above function to

$$\sum_{k=1}^{K}||f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}]})-\mathbf{t}^k||_2^2,$$

and notice that $\frac{\partial f(\mathbf{X}_\mathcal{G},\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]})}{\partial(\mathbf{X}_\mathcal{G})}$ denotes $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$ by definition.

Function 2 can be seen as a training process to force the model $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$ to operate *only* on $\mathbf{X}_\mathcal{G}$ (as the model $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}]}$ does) by shrinking the element-wise ℓ_2 norm of $\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$.

Further, as $\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}=\cup_{k=1}^K\mathbf{\Theta}^k_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$, forcing the model to operate on $\mathbf{X}_\mathcal{G}$ can be achieved by shrinking the element-wise ℓ_2 norm, *i.e.* Frobenius norm, of $\mathbf{\Theta}_{[\mathbf{X}_\mathcal{G}+\mathbf{X}_\mathcal{D}]}$

	Accuracy Coordinate	Robustness Coordinate	Details (Average Accuracy) Noise	Blur	Weather	Digital
AlexNet	0.000	0.000	0.1187	0.3187	0.2015	0.277
AlexNet(M1P30)	-0.002	0.013	0.1182	0.3459	0.2027	0.2858
AlexNet(M3P50)	-0.106	0.002	0.0964	0.3011	0.1675	0.2459
ResNet18	0.151	0.022	0.1727	0.3625	0.2657	0.2955
ResNet(M1P25)	0.148	-0.112	0.1729	0.3798	0.27	0.3068
ResNet(M2P75)	0.060	-0.142	0.154	0.3196	0.2357	0.2605
SqueezeNet	0.030	-0.179	from (Hendrycks and Dietterich 2019)			
VGG-11	0.221	-0.233	from (Hendrycks and Dietterich 2019)			
VGG-19	0.281	-0.229	from (Hendrycks and Dietterich 2019)			
ResNet-50	0.347	-0.039	from (Hendrycks and Dietterich 2019)			

Table 1: The exact coordinates used to plot Figure 2.

References

Bishop, C. M. 1995. Training with noise is equivalent to tikhonov regularization. *Neural computation* 7(1):108–116.

Hendrycks, D., and Dietterich, T. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Xu, H.; Caramanis, C.; and Mannor, S. 2009. Robust regression and lasso. In *Advances in Neural Information Processing Systems*, 1801–1808.