TU Delft

Delft University of Technology

Enhancing Classifier Conservativeness and Robustness by Polynomiality

Wang, Ziqi; Loog, Marco

DOI
10.1109/CVPR52688.2022.01297

Publication date
2022

Document Version
Final published version

Published in
Proceedings - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022

Citation (APA)
Wang, Z., & Loog, M. (2022). Enhancing Classifier Conservativeness and Robustness by Polynomiality. In *Proceedings - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022* (pp. 13317-13326). IEEE . https://doi.org/10.1109/CVPR52688.2022.01297

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Enhancing Classifier Conservativeness and Robustness by Polynomiality

Ziqi Wang•
•Delft University of Technology
The Netherlands
z.wang-8@tudelft.nl

Marco Loog°•
°University of Copenhagen
Denmark
m.loog@tudelft.nl

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-6946-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CVPR52688.2022.01297

Abstract

We illustrate the detrimental effect, such as overconfident decisions, that exponential behavior can have in methods like classical LDA and logistic regression. We then show how polynomiality can remedy the situation. This, among others, leads purposefully to random-level performance in the tails, away from the bulk of the training data. A directly related, simple, yet important technical novelty we subsequently present is softRmax*: a reasoned alternative to the standard softmax function employed in contemporary (deep) neural networks. It is derived through linking the standard softmax to Gaussian class-conditional models, as employed in LDA, and replacing those by a polynomial alternative. We show that two aspects of softRmax, conservativeness and inherent gradient regularization, lead to robustness against adversarial attacks without gradient obfuscation.*

1. Introduction

Models that show some form of exponential behavior are ubiquitous in machine learning: from the Gaussian class conditional distribution in linear discriminant analysis (LDA) [11, 15] to sigmoid activation for logistic regression [19, 27], and the softmax activation function in deep neural networks [23, 37]. Models with such use of exponentiality can, however, have unwanted behavior. We describe and illustrate such behavior, examine its reason, and propose a partial remedy by switching to models that behave polynomially. Like [6, 17], we consider the distribution tail and show that samples in the tails receive overconfident posterior predictions [21]. This renders the model sensitive to outliers and causes overfitting, especially in the case of distribution shift. Moreover, we link overconfident predictions to the lack of robustness against gradient based adversarial attacks.

A model should not be certain about a sample that deviates too much from the training data. Overconfident predictions on samples in the distribution tails should often be avoided, e.g. an atypical patient may otherwise be classified to be healthy or diseased with strong confidence. We want what we call *conservativeness*, which expresses the fact that we are uncertain. Specifically, we define it to be random guess-level prediction for samples in the tail of the distribution and show that this can be achieved by moving from exponential to polynomial behavior both in LDA and logistic regression. In addition, for logistic regression and deep learning, studies into the standard softmax activation have shown that it is not necessarily the best choice in many settings [8, 18, 40]. We propose a polynomial form of softmax posterior estimation that we coin *softRmax*. For this, we exploit the connection between the standard softmax function and LDA [4] and adopt a modified Cauchy distribution as the substitute for the (super)exponential Gaussian term.

Besides overconfident predictions, the use of exponentiality is also linked to vulnerability to adversarial attacks. Such attacks aim to cause malicious prediction changes by adding an unnoticeable perturbation to the original input. *Robustness* is the ability to maintain performance under adversarial attacks [5]. We demonstrate that a higher robustness of neural networks can be obtained by simply substituting the standard softmax with our softRmax. We show that the robustness can be linked back to the conservativeness of softRmax and inherent gradient regularization. The first factor, conservativeness, mainly brings robustness against gradient based attacks. The second leads to an enlarged margin between samples and the decision boundary, thereby boosting robustness against attacks as well. The effectiveness of various strategies countering adversarial attacks can be attributed to gradient obfuscation [2, 10]. We show that our inherent gradient regularization does not rely on such obfuscation.

We sketch the benefits of conservativeness under covariate shift [14, 38, 39] and show it when a model is under attack. We verify the robustness of our polynomial substitutes empirically on toy and public datasets. We further propose a semi-black-box attack, which we call an average-sample attack, to confirm that the robustness of our softRmax indeed comes from the above two factors. We also introduce a scale-invariant metric, the magnitude-margin ratio, for comparing the robustness of different models under the same level of attack.

978-1-6654-6946-3/22/$31.00 ©2022 IEEE
DOI 10.1109/CVPR52688.2022.01297

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.

2. Background Material and Related Methods

Adversarial attacks are used for robustness evaluation in our work. They are categorized into white-box and black-box attacks, depending on whether the network is available or not [33]. Black-box attacks do not need the network architecture and usually involve the training of a substitute network that mimics the decision boundary of the target network [31]. A gradient-based adversarial attack is a typical white box attack [12, 25]. It aims to find the perturbation direction that can lead to the fastest change in the prediction.

FGSM [12] is a simple yet effective approach where a small perturbation η is added to the input $\mathbf{x}$ to increase the overall loss. The perturbation η is ϵ multiplied by the sign of the loss gradient $\nabla_x J(\mathbf{w}, \mathbf{x}, y)$. The perturbed input becomes:

$$\mathbf{x}' = \mathbf{x} + \epsilon \,\text{sign}(\nabla_x J(\mathbf{w}, \mathbf{x}, y)). \tag{1}$$

Similarly, a gradient-based target attack [32] aims to perturb the sample to a target class y_t by decreasing the loss that corresponds to the target class:

$$\mathbf{x}' = \mathbf{x} - \epsilon \,\text{sign}(\nabla_x J(\mathbf{w}, \mathbf{x}, y_t)). \tag{2}$$

BIM [25] performs the attack iteratively in T steps. With the same attacking scale ϵ, BIM applies the attack at the scale of $\alpha = \epsilon/T$ in each step to form an attacked input $\mathbf{x}'_t$ at step t:

$$\mathbf{x}'_{t+1} = \mathbf{x}'_t + \alpha \,\text{sign}(\nabla_x J(\mathbf{w}, \mathbf{x}, y)). \tag{3}$$

We need the notion of a prediction margin M_z [41] to measure the robustness to adversaries, which has an indirect link to the classical (geometrical) margin in the input space [36]. Our work uses it to evaluate the margin and the robustness of our method. For this, we consider a mapping from the input $\mathbf{x}$ to the latent or representation space: $\mathbf{z} = f(\mathbf{x}, \mathbf{w})$, with $\mathbf{z} \in \mathbb{R}^k$ and $\mathbf{z}_i$ the output of the final layer corresponding to class $i \in \{0, 1, \ldots, k\}$. Assuming a sample $\mathbf{x}$ is correctly classified to its class y, $\mathbf{z}_y$ takes on the maximum value in $\mathbf{z}$. The prediction margin is defined as the distance between $\mathbf{z}_y$ and the second largest value in $\mathbf{z}$:

$$M_z := z_y - \max_{i \neq y} \{z_i\}. \tag{4}$$

Adversarial defenses for deep learning have been achieved by adversarial training [24], distillation [3,34], constructing a maximum margin in the latent space [16, 29, 30, 42] and gradient regularization [9, 13, 16, 28, 35, 41]. Explanations for gradient regularization approaches are heuristic and their successes often hinge on gradient obfuscation [10]. The latter refers to an unnecessarily rough loss landscape that hinders gradient-based adversarial attacks, which get readily stuck in the local minima of the roughened loss. It should be noted, however, that this approach does not solve the problem of adversarial attacks inherently [2]. Increasing the iteration number in BIM attacks [25] and using black-box attacks are standard to detect gradient obfuscation. We use both in our work to show that our approach does not rely on gradient obfuscation.

Covariate shift is a specific problem within domain adaptation. Domain adaptation refers to the scenario where the training data and the test data are not i.i.d. [7, 20]. The training and test data are referred to as the source domain and the target domain, respectively. One standard solution of this problem is to approximate the target domain by assigning the source samples weights determined by the source and target distribution. In the original work [38], these are estimated using Gaussian distributions. With this approach, adding very few samples in the tail of the source distribution can lead to considerable overfitting to the added outliers. We illustrate that, if a polynomial t-distribution instead of Gaussian distribution is adopted in the procedure of density estimation, the influence of outliers is limited.

3. Exponentiality vs Polynomiality

We first demonstrate the presence of overconfident prediction in the distribution tail and sensitivity to adversarial attacks with classical LDA, logistic regression, and deep learning. We then replace the exponential terms in each scenario by polynomial ones and show that this substitution is a simple yet effective approach to deliver conservativeness and improved robustness. Notably, for the latter, no adversarial training or extra regularization is required.

3.1. Conservativeness

Conservativeness is defined as estimating the posterior class probabilities $p(y_i|\mathbf{x})$ at random-guess level for $\mathbf{x}$ in the tail, away from the bulk of the data. To study such tail behavior, we basically study $\mathbf{x}$ for which the norm grows indefinitely, i.e., $\|\mathbf{x}\| \to \infty$. Assuming k classes and ignoring class priors, conservativeness comes down to the requirement that we can informally state as:

$$\lim_{\|\mathbf{x}\| \to \infty} p(y_i|\mathbf{x}) \approx \frac{1}{k}. \tag{5}$$

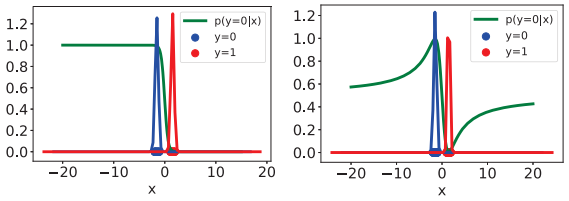
3.1.1 LDA

We consider k-class classification using LDA. We elaborate upon the link between the overconfident prediction and exponentiality. Following Bayes' rule, the posterior of class y_i, under equal priors, is

$$p(y_i|\mathbf{x}) = \frac{p(y_i)p(\mathbf{x}|y_i)}{\sum_k p(y_k)p(\mathbf{x}|y_k)} = \frac{p(\mathbf{x}|y_i)}{\sum_k p(\mathbf{x}|y_k)}. \tag{6}$$

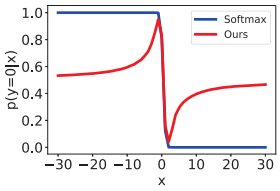
Consider $\mathbf{x}$ to be 1D for simplicity. The class conditional distribution $p(x|y)$ is estimated by fitting a Gaussian

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.



(a) Gaussian distribution.

(b) Student's t-distribution.



(c) Logistic regression.

Figure 1. LDA and logistic regression with exponential and polynomial assumptions. Posteriors $p(y_0|x)$ are compared. Subfigure 1a and 1b show the predicted posterior by LDA with class conditional Gaussian or t-distributions. Subfigure 1c compares the posterior of softmax and softRmax. LDA with Gaussian assumption and softmax with exponential functions show overconfident predictions in the distribution tails. Conservative prediction is achieved by substituting polynomial for exponential behavior.

$N(x|\mu_k, \sigma^2)$ with μ_k and σ^2 being the mean and variance of class k. When x goes to $\pm$infinity, the posterior saturates to one-hot encoding due to the (faster than) exponential rate of decrease of the Gaussian distribution. Specifically, we have

$$p(y_i|x) = \left(1 + \sum_{k\neq i} \exp\left(-\frac{1}{2\sigma^2}(2x(\mu_i - \mu_k) - \mu_i^2 + \mu_k^2)\right)\right)^{-1} \tag{7}$$

from which we see that $\lim_{x\to\pm\infty} p(y_i|x) = 0$, unless y_i is the mean closest to $x = \pm\infty$, in which case the posterior will be 1. This is also illustrated in Figure 1a.

Polynomial substitute. We propose to substitute the Gaussian distribution with the (noncentral) Student's t-distribution in the density estimation. Other distributions that fall of polynomially can be considered as long as the power of the leading terms are the same for all k class conditional distributions. In this way, conservative posteriors with a behavior as in Equation (5) are obtained.

The reason for this is that the limit of x going to $\pm$infinity for Equation (6) behaves rather different when the numerator and denominator contain polynomial instead of exponential terms. For the former, convergence is controlled by the polynomial decay rate of the posteriors $p(x|y_k)$. When equal, the limit posterior, assuming all priors equal, is $\frac{1}{k}$.

Example. We consider a binary classification task in 1D data. We assume a uniform distribution in the range $[-2, -1]$ for class y_0 and $[1, 2]$ for class y_1. With Gaussian distributions for the class conditional distributions $p(x|y_0)$ and $p(x|y_1)$—fitted using maximum likelihood, we get the change of posterior $p(x|y_0)$ w.r.t. input $\mathbf{x}$ as in Figure 1a. When $x \to -\infty$, $p(y_0|x) = 1$ and when $x \to \infty$ $p(y_1|x) = 1$. Substituting the t-distribution for the Gaussian, as shown in Figure 1b, for samples that are in the bulk of the class conditional distribution, we still obtain a posterior $p(y_0|x)$ close to 1. But for samples in the tail, we find more conservative prediction where $p(y_0|x)$ and $p(y_1|x)$ are approximately $\frac{1}{2}$.

3.1.2 Logistic Regression and Softmax

The softmax in neural network, as employed in the last layer to come to posterior estimates, works in the same way as multi-class logistic regression for classification tasks. Here, we consider a basic linear transformation $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}+\mathbf{b} = \mathbf{z}$, though our analysis can be readily generalized to nonlinear neural networks.

With the standard softmax activation ς^S, the embedding $\mathbf{z}$ is mapped to a vector of posteriors with $p(y_i|\mathbf{x}) = \varsigma_i^S(\mathbf{z}) = e^{\mathbf{z}_i} / \sum_k e^{\mathbf{z}_k}$. Equivalent to Equation (7), we have

$$\varsigma_i^S(\mathbf{z}) = \frac{\exp(\mathbf{w}_i^T\mathbf{x} + \mathbf{b}_i)}{\sum_k \exp(\mathbf{w}_k^T\mathbf{x} + \mathbf{b}_k)} = \left(1 + \sum_{k\neq i} \exp\left((\mathbf{w}_k - \mathbf{w}_i)^T\mathbf{x} + (\mathbf{b}_k - \mathbf{b}_i)\right)\right)^{-1}. \tag{8}$$

When $\|\mathbf{x}\| \to \infty$, if $(\mathbf{w}_k - \mathbf{w}_i)^T\mathbf{x}$ is negative for all $k, k \neq i$, then the posterior will be 1, otherwise it is 0.

We make the connection of softmax with LDA here. Let us position k normal distributions with identity covariance, $N(\cdot|\mathbf{m}, \mathbf{I})$, in Z. Their means are the k standard basis vector $\mathbf{e}_k$. Based on these distributions—every single one of them representing one of the k classes, we can map every $\mathbf{z} \in Z$ to a vector of posteriors $\varsigma^G(\mathbf{z})$, simply by setting

$$\varsigma_i^G(\mathbf{z}) := \frac{N(\mathbf{z}|\mathbf{e}_i, \mathbf{I})}{\sum_k N(\mathbf{z}|\mathbf{e}_k, \mathbf{I})}. \tag{9}$$

This, in turn, can be directly related to the softmax ς^S. First, we realize that, for $\mathbf{z}$ fixed,

$$\begin{aligned} N(\mathbf{z}|\mathbf{e}_i, \mathbf{I}) &\propto \exp(-\tfrac{1}{2}\|\mathbf{z} - \mathbf{e}_i\|^2) \\ &\propto \exp\left(-\frac{1}{2}\sum_k \mathbf{z}_k^2\right)\exp(\mathbf{z}_i)\exp(-\tfrac{1}{2}) \propto e^{\mathbf{z}_i}. \end{aligned} \tag{10}$$

From this, we immediately see that

$$\varsigma_i^G(\mathbf{z}) = \frac{N(\mathbf{z}|\mathbf{e}_i, \mathbf{I})}{\sum_k N(\mathbf{z}|\mathbf{e}_k, \mathbf{I})} = \frac{e^{\mathbf{z}_i}}{\sum_k e^{\mathbf{z}_k}} = \varsigma_i^S(\mathbf{z}). \tag{11}$$

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.

Polynomial substitute. Inspired by the standard Cauchy distribution $p_C(x) = \frac{1}{\pi(1+x^2)}$—a specific t-distribution, we use a polynomial term with the power of -2 to substitute the Gaussian class conditional distribution $N(\mathbf{z}|\mathbf{e}_i, \mathbf{I})$ in Equation (11), which gives our *softRmax* activation function ς^C:

$$\varsigma_i^C(\mathbf{z}) := \frac{\frac{1}{\|\mathbf{z}-\mathbf{e}_i\|^2}}{\sum_k \frac{1}{\|\mathbf{z}-\mathbf{e}_k\|^2}}. \tag{12}$$

By adopting the polynomial function, the posterior becomes conservative, because

$$\begin{aligned} p(y_i|\mathbf{x}) = \varsigma_i^C(\mathbf{z}) &= \frac{\|\mathbf{w}^T\mathbf{x}+\mathbf{b}-\mathbf{e}_i\|^{-2}}{\sum_k \|\mathbf{w}^T\mathbf{x}+\mathbf{b}-\mathbf{e}_k\|^{-2}} \\ &= \frac{1}{1+\sum_{k\neq i}\left\|\frac{\mathbf{w}^T\mathbf{x}+\mathbf{b}-\mathbf{e}_i}{\mathbf{w}^T\mathbf{x}+\mathbf{b}-\mathbf{e}_k}\right\|^2} \end{aligned} \tag{13}$$

and the terms $\left\|\frac{\mathbf{w}^T\mathbf{x}+\mathbf{b}-\mathbf{e}_i}{\mathbf{w}^T\mathbf{x}+\mathbf{b}-\mathbf{e}_k}\right\|^2$ converge to 1 when $\|\mathbf{x}\| \to \infty$.

Example. We consider logistic regression for binary classification in 1D. Similar to the previous example, we assume a uniform distributions in the ranges $[-1, 0]$ and $[1, 2]$ for the two classes y_0 and y_1. The sigmoid/softmax function is substituted with the softRmax activation function from Equation (12) to construct a conservative regressor. In Figure 1c, we see that the posterior $p(y_0|x)$ goes to $\frac{1}{2}$ on both ends.

3.2. Robustness

Next to softRmax being conservative, simply substituting the standard softmax with it in any probabilistic deep net also brings more adversarial robustness. We show that this comes from conservativeness in the tail and an inherent weight regularization that leads to an enlarged margin between samples and the decision boundary.

3.2.1 Robustness from Conservativeness

Most gradient-based adversarial attacks try to maximize the overall loss [12] or minimize the loss of a target class [32]. For a properly converged network that employs the standard softmax, attacking a correctly classified sample pushes it away from the tail, as the overall loss would not increase moving towards it (and the target class loss would not decrease). This is because the posterior of the correct class does not decrease towards the direction of the tail (see Figure 1c). The loss landscape using softRmax is different due to the conservativeness in the tail, as also illustrated in Figure 1c. For samples that are already positioned in the direction of the tail, an attack would actually push them even further into the tail. This increases the overall loss or decrease the target class loss. A perturbation towards the tail does, however, not change the accuracy so the attack fails. This leads a neural network using our softRmax to be more robust to gradient-based attacks. Note that this defense is different from gradient obfuscation because our loss landscape is not unnecessarily rough but simply has a different structure. This will be further elaborated in the corresponding experiment in Subsection 4.2.2.

3.2.2 Robustness from Enlarged Margin

The other factor that contributes to the robustness of softRmax, is the enlarged margin. To illustrate, we again consider a simple linear mapping of the input: $\mathbf{z} = \mathbf{w}^T\mathbf{x} + \mathbf{b}$. The theory can be generalized to nonlinear mappings by substituting the weight $\mathbf{w}$ with the gradient $\nabla_{\mathbf{x}}\mathbf{z}$ in the following derivation. With the output of the activation function being the general ς, the posterior gradient equals:

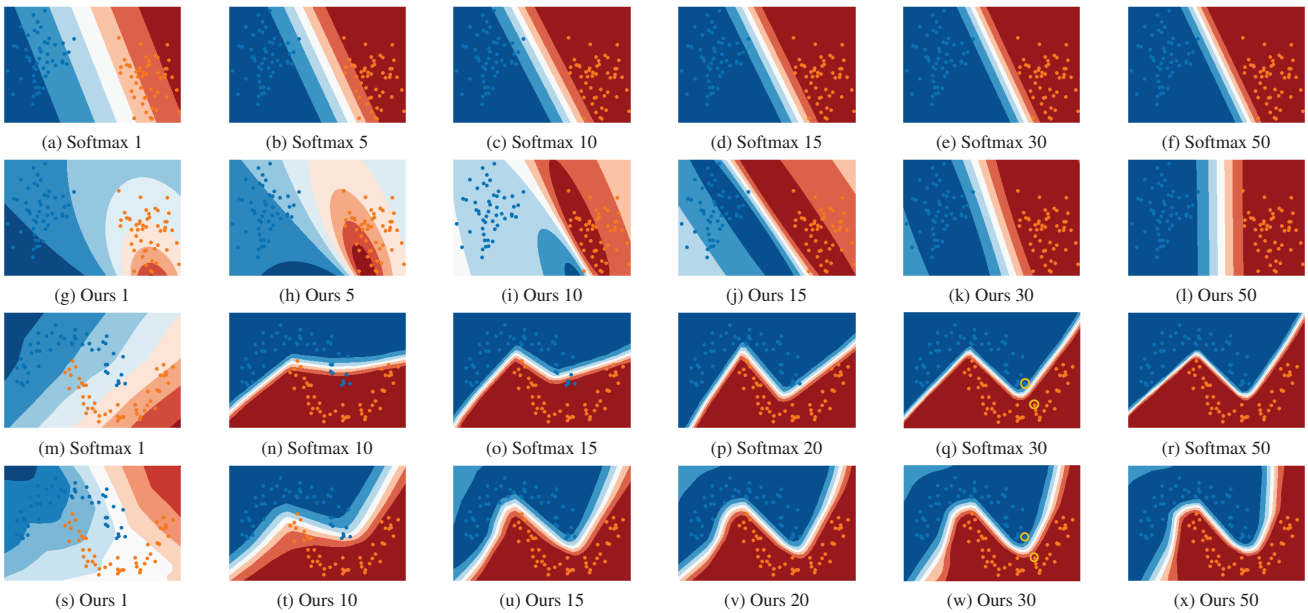
$$\frac{\partial\varsigma(\mathbf{z})}{\partial\mathbf{x}} = \frac{\partial\varsigma(\mathbf{z})}{\partial\mathbf{z}}\frac{\partial\mathbf{z}}{\partial\mathbf{x}} = \nabla_{\mathbf{z}}\varsigma\mathbf{w}. \tag{14}$$

The network weights are optimized by minimizing a posterior based loss function, which means the posterior of the labeled class should be maximized. For a separable dataset, there are many possible decision boundaries that can be learned by the network. When a decision boundary is biased by some samples close to the decision boundary (like in Figures 2c and 2p), the network generally has two options to further decrease the loss. It can either move the decision boundary to enlarge the classifier margin, or make the transient of posterior steeper at the decision boundary so posteriors of correctly classified samples saturate to 1 quicker. Both of the two approaches decrease the loss.

We observe that with softmax being the activation, the network tends to increase the posterior by making the posterior transient steep (as shown in Figures 2f and 2q). We believe that this is because the magnitude m of $\mathbf{w}$ is not regularized, so the network can simply increase m during the optimization process to increase the posterior gradient in Equation (14). This leads to the fast transient of the posterior around the decision boundary. A problem in the optimization is that the posteriors of samples will quickly saturate to 1 and do not contribute to gradient updates anymore. If a decision boundary is biased like in Figure 2c, it hardly changes in subsequent epochs. Such decision boundary correctly separates all data, but is more vulnerable to adversarial attacks because the classifier margin is not maximized.

Different from softmax, softRmax optimizes the loss in the other way: by enlarging the margin. It maps $\mathbf{x}$ to $\mathbf{z}$ around the kth row of the identity matrix $\mathbf{e}_k$ for class k, so $\|\mathbf{z}\| \approx 1$. Correspondingly, the magnitude of weight $\mathbf{w}$ is inherently regularized by $\|\mathbf{w}^T\mathbf{x} + \mathbf{b}\| \approx 1$. This avoids increasing the weights to values that can lead to a sharp decision boundary and leaves just one of the two above-mentioned optimization options to decrease the loss, i.e., enlarging the margin (see Figs. 2l and 2w), resulting in increased robustness.

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.



(a) Softmax 1 (b) Softmax 5 (c) Softmax 10 (d) Softmax 15 (e) Softmax 30 (f) Softmax 50

(g) Ours 1 (h) Ours 5 (i) Ours 10 (j) Ours 15 (k) Ours 30 (l) Ours 50

(m) Softmax 1 (n) Softmax 10 (o) Softmax 15 (p) Softmax 20 (q) Softmax 30 (r) Softmax 50

(s) Ours 1 (t) Ours 10 (u) Ours 15 (v) Ours 20 (w) Ours 30 (x) Ours 50

Figure 2. Margin change for linearly separable dataset and the moon dataset with softmax and our softRmax. Different colored points represent the two classes. The color bands show the posterior develops in the input space. The number in the title of each subfigure is the training epoch. With the standard softmax, the model makes the posterior change around the decision boundary sharp to minimize the loss. Due to the regularization of weights w with softRmax, it is harder to minimize the loss by increasing the posterior fast at the decision boundary, which enables the model to find a larger margin.

4. Experiments

We present experimental results on conservativeness and robustness when using standard exponential terms and polynomial substitutes respectively. First, we use covariate shift adaptation by importance weighting with outliers as an example to demonstrate that the conservativeness brought by polynomiality is necessary in an LDA-like setting. A next experiment shows that, even under attack, softRmax gives conservative posteriors. We also perform standard adversarial attacks on public datasets to compare the robustness of softmax and softRmax. To better understand the behavior of softRmax, we introduce a new, so-called, average-sample attack and the magnitude-margin ratio.

4.1. Conservativeness

4.1.1 Covariate Shift

Under covariate shift between a source domain D_s and a target domain D_t, a fixed labelling function is assumed. We consider a standard weighting approach [38] to make the source domain distribution p_{D_s} approximate the target domain distribution p_{D_t}:

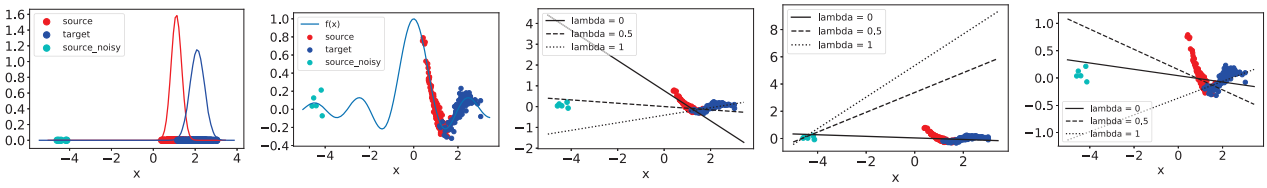
$$w_{cov} = \left(\frac{p_{D_t}(x)}{p_{D_s}(x)} \right)^{\lambda}. \tag{15}$$

Here, λ controls the strength of the weighting scheme. Similar to Section 3.1.1, Gaussian distributions $N(x|\mu;\sigma^2)$ with mean μ and variance σ^2 are estimated for p_{D_s} and p_{D_t}. When outliers occur in the tail of the source distribution, extreme weights w_{cov} are assigned to those outliers if the target distribution p_{D_t} has a larger variance $\sigma^2_{D_t}$ than $\sigma^2_{D_s}$ of the source domain. This will lead to overfitting to these outliers only. With the use of a t-distribution, a weight of 1 is obtained, resulting in improved estimator behavior.

A domain adaptation regression setting is considered similar to the original work [38]. The target function is $f(x) = \text{sinc}(x)$, shown in Figure 3b. The source and the target densities are $p_{D_s}(x) = N(x|1.1, (1/2)^2)$ and $p_{D_t}(x) = N(x|2.1, (10/17)^2)$, respectively. We add noise ϵ_{s_i} to the target function to create the output values for the source domain $y_{s_i} = f(x_{s_i}) + \epsilon_{s_i}$ with $p(\epsilon_s) = N(\epsilon_s|0, (1/4)^2)$. We set the source sample size to $n_s = 150$ and the target sample size to $n_t = 100$. To approximate the target domain, each source sample is assigned a weight w_{cov} according to Eq. (15). We randomly sample 5 outliers in the range $[-5, -4]$ and add them to the source domain after density estimation. All parameters are estimated by maximum likelihood.

The sensitivities to outliers with Gaussian density estimation and using the t-distribution are compared in Figure 3. With Gaussian density estimation, noisy samples receive large weights due to the larger variance of the target domain, therefore the regressor overfits to the noisy samples when λ is not 0. Student's t-distribution leads to small weights for the noisy samples because of its heavy tail.

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.



(a) Marginal distribution (b) Target function (c) Wihtout outliers (d) Gaussian (e) Student-t

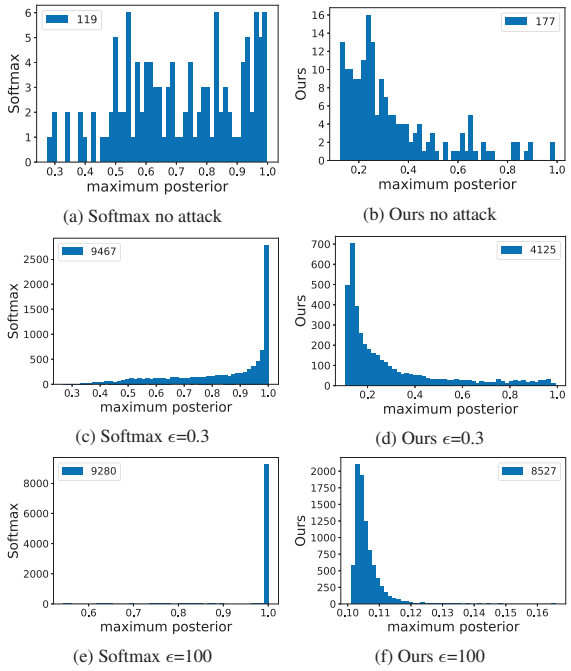
Figure 3. (a) visualizes the marginal distribution of the source and target domain. The target function and the output value for the regression task are shown in (b). Figure (c) visualizes the fitted lines in the original scenario with Gaussian density estimation without outliers added. Figures (d) and (e) present the adaptation results of Gaussian density estimation and using Student's t-distribution with outliers separately. Lambda in the legend refers to the power λ in Equation (15). Gaussian density estimation overfits to the outliers.

4.1.2 Conservative Prediction

We show that conservativeness leads to further desirable behavior under adversarial attacks. For networks trained with softmax, adversarial attacks make the network misclassify samples with high confidence. But when a model with softRmax is attacked, the sample is misclassified but with low confidence due to the conservativeness.

We show the confidence of misclassified sample is low with softRmax by examining the posteriors of misclassified samples from the public dataset MNIST [26] under different levels of adversarial FGSM attacks. The network has four convolutional layers and one fully connected layer. We set a batchsize of 32 and optimize the network by Adam with a learning rate of $1e-3$. We use the same architecture for the softmax and softRmax setting, with the only difference being the activation function after the final layer.

As shown in Figure 4, for the network trained with the standard softmax, posteriors on the predicted class of misclassified samples are high on average. Specifically, using softmax, under large scale attacks with $\epsilon = 100$, all samples are misclassified with a posterior of 1. Due to the conservativeness in the tail and the soft posterior change, our softRmax leads to posteriors around random-guess level.



(a) Softmax no attack (b) Ours no attack

(c) Softmax ϵ=0.3 (d) Ours ϵ=0.3

(e) Softmax ϵ=100 (f) Ours ϵ=100

Figure 4. Posteriors of predicted class for misclassified MNIST test samples under different levels of FGSM attacks. The legends give the number of misclassified samples. Misclassified samples in the setting of softRmax receive less confident prediction. Even under extreme attacks with $\epsilon = 100$, our softRmax gives non-saturated posteriors at random-guess level.

4.2. Robustness

4.2.1 Adversarial Defense

We perform experiments on public datasets MNIST [26], CIFAR10 [22], and CIFAR100 with standard softmax and our softRmax. The setting of MNIST is the same as in Section 4.1.2. A randomly initialized VGG16 network is used for CIFAR10 classification. We optimize VGG16 by SGD with a learning rate of $5e-3$, batchsize 256 and weight decay $5e-6$. For CIFAR100, we adopt ResNet50 pretrained on ImageNet and finetune it with Adam. We set the learning rate to be $1e-4$, batchsize 512 and weight decay $5e-6$. Note that no extra data augmentation is used in any experiment. The only difference between the baseline with softmax and our approach is the activation function after the final fully connected layer. By simply substituting the softmax activation function with the polynomial softRmax activation function, the network develops strong adversarial defense ability (see Table 1). Without being combined with other approaches, the naive softRmax model can outperform state of the art adversarial defense approaches based on attention mechanism on CIFAR datasets [1].

4.2.2 Gradient Obfuscation

As we noted in Section 3.2.1, different from gradient obfuscation, our loss landscape is not rough but simply has a

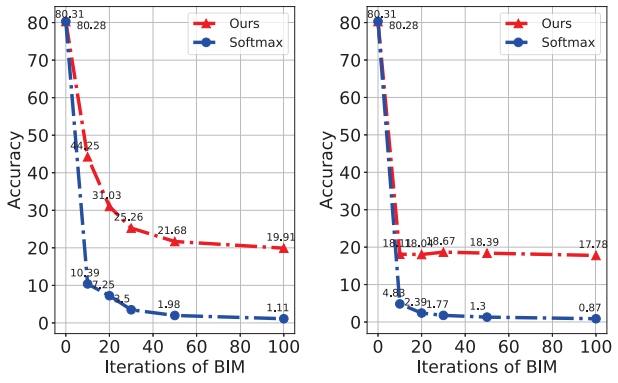
Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.

Table 1. Adversarial defense results. We compare networks with softmax and our softRmax activation under FGSM and BIM attacks (T=10). 'Clean' refers to the classification accuracy on the testset without any attack. We consider binary classification for class 3 and 7 from MNIST, MNIST, CIFAR10, and CIFAR100 under different attack levels ϵ. The results show a clear improvement of the robustness to adversarial attacks with softRmax.

Dataset	Method	Clean	FGSM		BIM	
			ϵ=0.1	ϵ=0.3	ϵ=0.1	ϵ=0.3
MNIST 3&7	softmax	99.75	77.18	18.69	71.64	0.24
	ours	**99.95**	**95.88**	**88.71**	**94.90**	**66.24**
MNIST	softmax	96.8	48.3	0.41	53.49	0.02
	ours	**97.78**	**75.55**	**49.30**	**69.73**	**33.94**
CIFAR10	softmax	**80.31**	17.62	13.95	10.39	4.83
	ours	80.28	**49.93**	**41.03**	**44.25**	**18.11**
CIFAR100	softmax	**61.43**	11.23	6.78	1.94	0.05
	ours	61.04	**19.31**	**11.04**	**9.06**	**2.16**

different structure in the tail of the distribution. Leading a sample to the tail is different from blocking the attack by local minimum due to a rough loss landscape. The tail is the right direction to perturb the sample from the point of view of the gradient based attacks because the overall loss monotonically increases towards the tail.

Nevertheless, we also rule out the possibility of gradient obfuscation by performing the iterative BIM attack at very large iteration numbers. In fact, softRmax shows stronger robustness after the accuracy stabilises with increased iterations, as shown in Figure 5. The experiment in the next section shows that the black-box attack is a weaker attack than the white-box attack, which further diminishes the possibility that the softRmax robustness can be explained by gradient obfuscation.



(a) ϵ=0.1 (b) ϵ=0.3

Figure 5. BIM attack on CIFAR10 with different iteration numbers. For both attack levels $\epsilon = 0.1$ and $\epsilon = 0.3$, the stabilized accuracy of softRmax is significantly higher than that of softmax.

4.2.3 Robustness from Conservativeness

Existing gradient-based attacks can only examine the robustness of a model as a whole but cannot show whether the robustness comes from the enlarged margin or from the conservativeness or both. To check the effect of the enlarged margin and the conservative tail on robustness, we propose a semi-black-box attack, coined the average-sample attack. It does not rely on the gradient but simply perturbs a sample to the direction of a selected target class based on a precomputed average, so the sample is guaranteed to not be pushed towards the tail. We precompute the average sample $\mathbf{Avg}_y = \frac{1}{n}\sum^n \mathbf{x}_{ny}$ for each class y. With t the target class, the adversarial input $\mathbf{x}'$ then becomes

$$\mathbf{x}'_y = \mathbf{x}_y + \epsilon \text{sign}(\mathbf{Avg}_t - \mathbf{Avg}_y). \tag{16}$$

In general, our average-sample attack should not be stronger than the gradient-based ones because the attacking direction found by the former attack is not optimized. It prevents the sample from going to the tail, so if it becomes a stronger attack for the softRmax, it indicates that the conservativeness in the tail indeed gives added robustness. Otherwise the gradient based white-box attack should be the worst attack for our softRmax model as well.

Table 2. Results of targeted attack on MNIST dataset. White refers to the targeted attack with the network available for generating adversarial samples. Black is the black-box version of the targeted attack, where a substitute network is first learned to approximate the decision boundary of the original model. Avg is our average-sample attack. The column Clean is the original per class accuracy of different models without any adversarial attack. The rest results are the accuracy of all the 10 classes under adversarial attacks. We highlight the **worst** performance of each model among all attacks.

Classes	Clean		White		Black		Avg	
	softmax	Ours	softmax	Ours	softmax	Ours	softmax	Ours
0	98.88	99.18	**11.14**	**53.2**	33.55	74.3	30.69	58.13
1	98.50	99.21	**15.05**	**46.73**	53.41	60.73	48.77	63.44
2	98.26	98.16	**10.50**	54.00	25.93	50.42	16.53	**42.26**
3	96.34	98.12	**10.31**	51.73	27.16	58.99	15.54	**37.77**
4	96.84	97.35	**13.28**	51.84	37.21	63.03	26.97	**47.94**
5	97.87	98.09	**9.33**	52.24	32.84	64.52	16.68	**46.76**
6	97.91	98.64	**12.00**	51.52	35.46	52.14	24.43	**44.56**
7	96.60	96.69	**12.11**	52.88	32.46	60.10	22.03	**45.50**
8	92.61	96.61	**9.36**	54.38	24.96	73.04	18.90	**41.93**
9	94.05	95.64	**6.33**	51.72	30.65	68.58	29.31	**49.67**

To check whether the average-sample attack is a stronger attack on the softRmax model, we also perform a gradient based targeted attack in both the white-box setting and the black-box setting. The latter one checks whether gradient obfuscation happens. In the black-box attack [31], a substitute model that is used to generate adversarial samples is first learned to mimic the decision boundary of the original

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.

model. We show that the white-box attack is the strongest attack for softmax while our customized average-sample attack is more effective on the softRmax (see Table 2). This indicates that the conservative tail of softRmax indeed leads to robustness. The fact that the black-box attack is weaker than the white-box attack eliminates the possibility that the weaker performance of average-sample attack is brought by gradient obfuscation. Also, even when the average-sample attack gives the lowest accuracy on softRmax, its performance is still significantly better than that of softmax.

4.2.4 Robustness from Enlarged Margin

We further demonstrate that the robustness of softRmax also comes from the enlarged margin. If all samples are pushed to the decision boundary instead of the tail, then the model with a larger margin is more robust. It is hard to measure the margin in the input space so we use the prediction margin M_z, as specified in Equation (4), as alternative. If the perturbation in $\mathbf{x}$ can push the sample across the margin, then it means the corresponding change in M_z is also larger than the original prediction margin M_z. However, M_z cannot be used to measure the margin directly due to different mappings from $\mathbf{x}$ to $\mathbf{z}$ of different models. A larger M_z does not imply a larger margin in the input space. So we introduce a new metric, the magnitude-margin ratio, to measure the change in M_z caused by an attack with respect to the original prediction margin. If the change is larger than the original prediction margin for a sample, it indicates that this sample can be successfully attacked.

To derive the ratio for an $\mathbf{x}$, we assume the index for $\max_{i \neq y}\{\mathbf{z}_i\}$ is j. We denote the gradient of $\mathbf{z}_y$ and $\mathbf{z}_j$ of the input $\mathbf{x}$ by $\mathbf{w}_y$ and $\mathbf{w}_j$, respectively. After adding a perturbation η in the input $\mathbf{x}$, $\mathbf{z}_y$ and $\mathbf{z}_j$ change to $\tilde{\mathbf{z}_y}$ and $\tilde{\mathbf{z}_j}$, where $\tilde{\mathbf{z}_y} = \mathbf{w}_y^T\mathbf{x} + \mathbf{w}_y^T\eta$. The new prediction margin $\tilde{M_z} = \tilde{\mathbf{z}_y} - \tilde{\mathbf{z}_j}$. According to [12], $\mathbf{w}^T\eta$ can be approximated by the magnitude m of gradients, the attacking level ϵ of η, and the dimension n of input as ϵmn and so

$$r = \frac{|\tilde{M_z} - M_z|}{M_z} = \frac{|(\mathbf{w}_y^T - \mathbf{w}_j^T)\eta|}{M_z} \approx \frac{\epsilon mn}{M_z}. \qquad (17)$$

Given the same input dimension n and attacking level ϵ, the simplified ratio $R = \frac{m}{M_z}$ can serve as the metric. A model with a distribution of lower ratio R means that, with the same level of attack, it is harder to change the prediction margin M_z, which indicates a larger margin in the input space and a higher robustness of this model. Figure 6 shows that the model with softRmax has ratios R lower than softmax has on MNIST, CIFAR10, and CIFAR100.

5. Discussion and Conclusion

We suggest an easy substitution of polynomiality for exponentiality in several scenarios, showing it leads to conservative behavior regarding samples in the tail of the distribution. For our polynomial softRmax, this behavior also leads to increased robustness against adversarial attacks. We show that the robustness of softRmax also comes from an enlarged margin and link this to the inherent gradient regularization of softRmax, which demonstrates that its success does not stem from gradient obfuscation. Our softRmax can be combined readily with many other adversarial defense strategies and it would be of interest to study their combined strength.

Given the type of conservative behavior polynomiality induces, it seems worthwhile to study its usage in OOD detection and other problems related to non-i.i.d. sampling, domain adaptation, etc.

As for softmax, good DNN weight initialization is important to avoid gradient vanishing for softRmax. Apart from that, considering the current level of understanding and the experimental evidence provided, we see no restrictions to its usage. In conclusion: why not give softRmax a try?

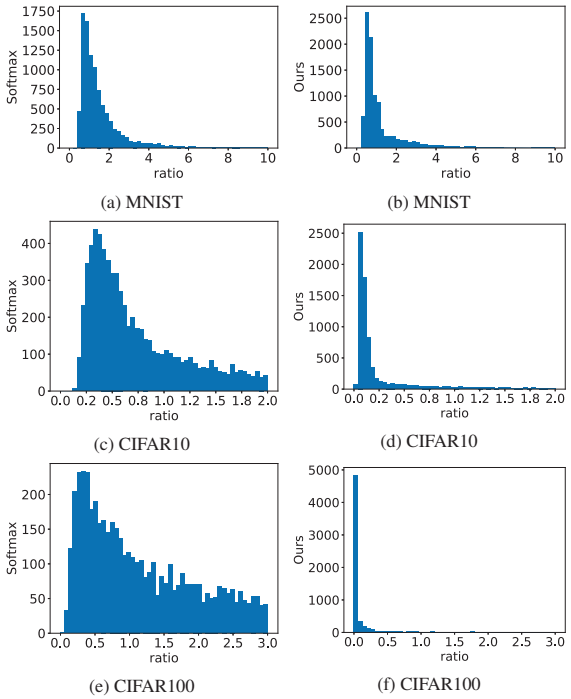


Figure 6. Histogram of the magnitude-margin ratio R of softmax and softRmax on all test data from MNIST, CIFAR10, and CIFAR100. The ratio of softRmax is significantly smaller, which indicates a higher robustness against adversarial attacks.

Acknowledgements. Many thanks to Jan van Gemert for his feedback and invaluable help with the rebuttal. Funded in part by the Netherlands Organization for Scientific Research (NWO), research program C2D–Horizontal Data Science for Evolving Content (628.011.002).

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.

References

[1] Prachi Agrawal, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Impact of attention on adversarial robustness of image classification models. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3013–3019. IEEE Computer Society, 2021.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[3] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 2654–2662, 2014.

[4] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[6] Robert Chen. A remark on the tail probability of a distribution. *Journal of Multivariate Analysis*, 8(2):328–333, 1978.

[7] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[8] Alexandre de Brébisson and Pascal Vincent. An exploration of softmax alternatives belonging to the spherical loss family. In *ICLR (Poster)*, 2016.

[9] Harris Drucker and Yann Le Cun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

[10] Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness. *Machine Learning with Applications*, 3:100017, 2021.

[11] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050:20, 2015.

[13] Rostislav Goroshin and Yann LeCun. Saturating auto-encoders. In *1st International Conference on Learning Representations, ICLR 2013*, 2013.

[14] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[15] Trevor Hastie and Robert Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):155–176, 1996.

[16] Sibylle Hess, Wouter Duivesteijn, and Decebal Mocanu. Softmax-based classification is k-means clustering: Formal proof, consequences for adversarial attacks, and improvement through centroid based tailoring. *arXiv preprint arXiv:2001.01987*, 2020.

[17] Bruce M Hill. A simple general approach to inference about the tail of a distribution. *The annals of statistics*, pages 1163–1174, 1975.

[18] Sekitoshi Kanai, Yuki Yamanaka, Yasuhiro Fujiwara, and Shuichi Adachi. Sigsoftmax: Reanalysis of the softmax bottleneck. *Advances in Neural Information Processing Systems*, 2018:286–296, 2018.

[19] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[20] Wouter Marco Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[21] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, pages 5436–5446. PMLR, 2020.

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[24] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[25] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[26] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[27] Scott Menard. *Applied logistic regression analysis*, volume 106. Sage, 2002.

[28] Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

[29] Tianyu Pang, Chao Du, and Jun Zhu. Max-mahalanobis linear discriminant analysis networks. In *International Conference on Machine Learning*, pages 4016–4025. PMLR, 2018.

[30] Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019.

[31] 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.

[32] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[33] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[34] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.

[35] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[36] Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

[39] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[40] Michalis K Titsias. One-vs-each approximation to softmax for scalable estimation of probabilities. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4168–4176, 2016.

[41] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: scalable certification of perturbation invariance for deep neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6542–6551, 2018.

[42] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9117–9126, 2018.

Authorized licensed use limited to: TU Delft Library. Downloaded on December 12,2022 at 21:08:52 UTC from IEEE Xplore. Restrictions apply.