Margin Maximization for Robust Classification using Deep Learning

Alexander Matyasko, Lap-Pui Chau
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore
{aliaksan001, elpchau}@ntu.edu.sg

***Abstract*—Deep neural networks have achieved significant success for image recognition problems. Despite the wide success, recent experiments demonstrated that neural networks are sensitive to small input perturbations, or *adversarial noise*. The lack of robustness is intuitively undesirable and limits neural networks applications in adversarial settings, and for image search and retrieval problems. Current approaches consider augmenting training dataset using *adversarial examples* to improve robustness. However, when using data augmentation, the model fails to anticipate changes in an adversary. In this paper, we consider maximizing the geometric margin of the classifier. Intuitively, a large margin relates to classifier robustness. We introduce novel margin maximization objective for deep neural networks. We theoretically show that the proposed objective is equivalent to the robust optimization problem for a neural network. Our work seamlessly generalizes SVM margin objective to deep neural networks. In the experiments, we extensively verify the effectiveness of the proposed margin maximization objective to improve neural network robustness and to reduce overfitting on *MNIST* and *CIFAR-10* dataset.**

I. Introduction

Deep neural networks are powerful learning models that have achieved state-of-the-art performance for computer vision [1]. Despite the wide success for many computer vision problems [2], [3], recent experiments demonstrated that deep neural networks are not robust to small input perturbations known as an *adversarial noise* [4], [5]. Changing input image state-of-the-art classifiers by adding adversarial noise can produce incorrect predictions with high confidence [4]. The intriguing result is that the original and altered images are visually similar. This result does not only question generalization ability of deep neural networks, but additionally complicates near-duplicate detection and image search problems [6], and limits deep learning applications in real-world, potentially adversarial settings [7].

Adversarial noise is a minimal input perturbation which changes classifier prediction. *Adversarial examples* are the corrupted inputs which are visually similar to the original input images but for which classifier produces incorrect label prediction (see fig. 1). Designing classifiers robust to worst-case perturbation has several potential advantages. Firstly, an input to classifier can contain measurements errors or can be maliciously corrupted in adversarial settings by the attacker [7]. Secondly, worst-case analysis using robustness provides generalization bounds alternative to the traditional uniform convergence bounds used in statistical learning theory [8]. Lastly, local stability of the prediction is important for many applications, e.g. image search, near-duplicate detection, video-frames classification [6].

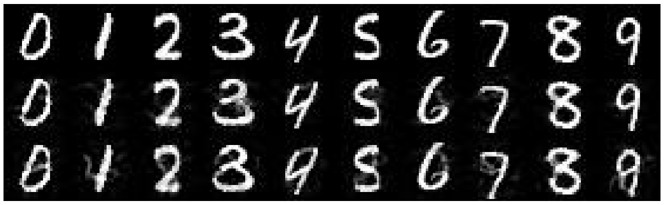


Fig. 1. Adversarial examples generated using [9] for convolutional neural network Lenet-5 [10] on MNIST dataset. First and second row depicts original images and adversarial images. The third row shows adversarial images after our regularization method was applied. As the result of our regularization, we observe that adversarial images are distinguishable in a visually meaningful way from the original images.

The topic of adversarial examples has spurred significant interest in deep learning research. Current approaches for improving network robustness against *adversarial noise* can be viewed as a form of data regularization using adversarial examples [5], [9], [11]. Data augmentation is conceptually simple, but it has several limitations. Firstly, an adversarial distortion should be label non-changing. If the perturbation is too large, it can alter the true class label. Additionally, generating an adversarial noise is time-consuming, and approximate methods are used in [5], [11]. Secondly and most importantly, data regularization methods [5] discard information about the dependency between the model parameters and an adversarial noise. The model fails to anticipate changes in the adversary. As a result, neural network tends to overfit the fooling method used for data augmentation [4], [9]. Regularized neural network classifies seen adversarial examples correctly but remains vulnerable to the newly generated adversarial examples.

In this paper, we propose a novel margin maximization objective for deep neural networks. Our margin formulation for binary classification minimizes the norm of the Jacobian which is analogous to [12] but offers a novel interpretation of the contractive penalty. For multiclass problems, we minimize the maximum norm of the difference between gradient w.r.t. inputs of the correct and incorrect prediction. We establish a connection between a margin of a classifier and its robustness. We prove that the introduced margin maximization objective and the robust optimization are equivalent for deep neural net-

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

works. Thus, the proposed objective theoretically guarantees robustness of the neural network. Our formulation generalizes support vector machine (SVM) margin maximization [13]. In a special case of deep linear networks, our objective reduces to a soft-margin SVM. We note that using SVM objective with deep neural networks was considered before [14]. However, the formulation used in [14] considered margin maximization in the feature space of the final layer which does not guarantee robustness in the input space [15]. To our best knowledge, this work is the first one to consider margin maximization in the input space for deep neural networks.

To summarize, our main contributions are as follows:

- We introduce a novel margin maximization objective for deep neural networks. Our formulation generalizes linear SVM margin objective to nonlinear classifiers. We theoretically show that the proposed margin maximization objective is equivalent to the robust optimization problem for a neural network.
- We experimentally verify the effectiveness of the proposed approach on MNIST and CIFAR-10 datasets and show that adversary-aware model optimization significantly improves robustness of regularized networks and decreases overfitting. Additionally, we qualitatively observe that as a result of the proposed regularization adversarial images are distinguishable in a visually meaningful way from the original images.

II. Related work

Fooling deep neural networks Szegedy [4] considered maximizing the likelihood of incorrect prediction with norm constraint on the input perturbation. Box-constrained L-BFGS-B optimization with line search was used to find a minimal adversarial perturbation. LFGS-B optimization is time-consuming, thus impractical as a method for generating adversarial noise during training. Goodfellow [5] proposed to use a first-order approximation of the loss function to find a minimal adversarial noise. The authors showed that the scaled sign of the loss function gradient w.r.t. inputs is an adversarial direction for l_∞-norm perturbation constraint. This method known as fast gradient sign is imprecise, but it has dramatically improved the speed of generating adversarial noise. DeepFool [9] iteratively estimates adversarial noise by taking a step in the direction of the closest decision boundary till the prediction is changed. Bastani [16] observed that for a network with linear rectified units adversarial perturbation could be found by solving linearly constrained program. Sabour [17] proposed to generate an adversarial noise such that the source image has a similar hidden layer representation to the guided image. Their experiments demonstrated that the hidden layer representation is also not robust to the adversarial noise. Nguyen [18] proposed an evolutionary algorithm to generate synthetic images which are classified by the neural network with high confidence.

Improving deep neural networks robustness Training on a mixture of clean and adversarial examples was considered as a way to regularize model in [4]. Computationally intensive L-BFGS-S optimization in [4] limited an application of adversarial noise for data augmentation during training. Goodfellow [5] introduced adversarial training (AT) which uses a weighted combination of the loss on clean examples and the loss on adversarial examples during training. Adversarial noise was generated using less accurate fast gradient sign method [5] which allowed efficient data regularization during model training. Layer-wise contractive penalty [12] was considered to improve robustness of deep neural networks in [19]. Miyato [11] proposed virtual adversarial training (VAT) using local distributional smoothness (LDS) regularization. LDS objective is defined as a KL-divergence between prediction distribution on clean images and prediction distribution on adversarially corrupted images. Notably, VAT training can be applied to semi-supervised learning. Improving robustness to image preprocessing (e.g. compression, cropping) was considered in [6]. Network distillation [20] as a defense against adversarial noise was proposed in [21]. Double Backpropagation [22] is also related to our work. Double backpropagation minimizes the sum of squared input gradients and is similar to the contractive penalty [12].

Robustness and generalization Robust classification and optimization under input uncertainty have been extensively studied in machine learning [23]. For several classifiers, their regularized formulation was shown to be equivalent to the robust optimization problem. The most notable example is a regularized support vector machine [13]. Intuitively, margin maximization objective of SVM guarantees robustness of the prediction to any adversarial perturbation with the norm less than optimal margin. This connection between SVM margin maximization and robust optimization was formally established in [15]. Xu [15] proved that soft-margin SVM is equivalent to the robust optimization problem. In addition, the authors showed that a similar result holds for kernelized SVM with locally smooth kernels. Generalized algorithmic robustness was introduced in [8] and was used to derive and improve generalization bounds. Our work generalizes SVM margin maximization objective to nonlinear classifiers and provides similar theoretical guarantees. We note that SVM margin objective was applied to deep neural network in [14]. But, standard SVM formulation when applied to deep neural networks does not guarantee robustness in the input space.

Several works examined why neural networks are unstable to adversarial noise. Szegedy [4] argued that adversarial examples occupy low-probability pockets in an input space of zero measure. Contrary to that, Goodfellow [5] demonstrated that adversarial noise could be generated by taking a single step in the direction of gradient w.r.t. inputs, thus adversarial examples form linear subspaces in the input space. Further experiments in [24] verified that adversarial examples occupy large regions in the pixel space. Fawzi [25] studied the relationship between robustness to random noise and adversarial noise. Their theoretical analysis showed that in high dimensions, given a small curvature of the classifier decision boundary, robustness to the random noise can be guaranteed even when the network is vulnerable to the adversarial noise.

III. Margin maximization and robustness

Before introducing margin maximization objective for deep neural networks, we first recall formulation of linear SVM [13] and review some results for the robustness of regularized SVM [15].

Let $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ be a binary classifier where $\mathbf{w} \in \mathbb{R}^N$ and $b \in \mathbb{R}$ are the classifier parameters. Given the input $\mathbf{x}$, the prediction $\hat{y}$ is $\operatorname{sign}(f(\mathbf{x}))$. Following standard SVM terminology, we define *functional margin* $\tilde{\gamma}$ and *geometric margin* γ for a given training example $(\mathbf{x}, y)$ as:

$$\tilde{\gamma} = y(\mathbf{w}^T\mathbf{x} + b) \tag{1}$$

$$\gamma = \frac{\tilde{\gamma}}{\|\mathbf{w}\|_2} \tag{2}$$

In addition, we introduce *margin sensitivity* ϵ which we define as a scaling factor in the relationship between functional and geometric margin. *Margin sensitivity* measures how much functional margin changes as we move towards the decision boundary. It is constant for affine classifier and equal to $1/\|\mathbf{w}\|_2$. However, this scaling factor for geometric margin is, in general, a function of training input $\mathbf{x}$ and training label y which motivates our introduction of a new definition.

Simple arithmetic manipulations can be used to show that maximizing minimum geometric margin is equivalent to the following optimization problem:

$$\begin{aligned} &\min \|\mathbf{w}\|_2^2 \\ &\quad \text{s.t. } y_i(\mathbf{w}^T x_i + b) \geq 1 \quad \forall\, i = 1, \ldots, m \end{aligned} \tag{3}$$

where m is the number of training examples. The above formulation is known as hard-margin SVM. Hard-margin SVM works only for separable data and is sensitive to the outliers. Slack variables are usually introduced to maximize soft margin. The objective for soft-margin SVM is defined as follows:

$$\min C\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m} \big(1 - y_i(\mathbf{w}^T x_i + b)\big)_+ \tag{4}$$

where $z_+ = \max(z, 0)$ is a hinge loss.

The theory of SVM is well developed. Large margin is usually associated with reduced classifier complexity and improved generalization ability of classifier [13]. By contrast, in this work we explore an another facet of large margin, namely an improved robustness of classifier. Classifier margin and its robustness are closely related. Indeed, classifier prediction $\hat{y} = f(\mathbf{x})$ for a given data point $(\mathbf{x}, y)$ is stable to any perturbation $\mathbf{r}$ with norm less than geometric margin γ for that point:

$$\operatorname{sign} f(\mathbf{x} + \mathbf{r}) = \operatorname{sign} f(\mathbf{x}) \quad \forall\, \|\mathbf{r}\|_2 \leq \gamma(\mathbf{x}) \tag{5}$$

In particular, for any given datapoint $(\mathbf{x}, y)$ hard-margin SVM is robust to any perturbation $\mathbf{r}$ with norm less than the optimal margin $1/\|\mathbf{w}\|_2$ (see fig. 2).

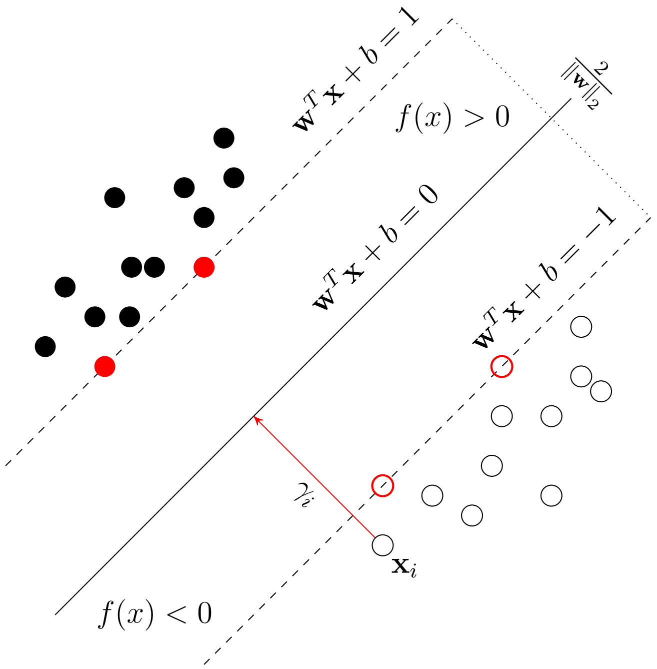

$\mathbf{w}^T\mathbf{x} + b = 1$
$f(x) > 0$
$\frac{2}{\|\mathbf{w}\|_2}$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$
γ_i
$\mathbf{x}_i$
$f(x) < 0$

Fig. 2. Decision surface of linear SVM classifier. For a given datapoint $\mathbf{x}_i$ any input perturbation with norm less than γ_i does not change classifier prediction $\hat{y}_i$. In particular, hard-margin SVM which maximizes minimum separating margin is robust to any perturbation with the norm less than $1/\|w\|_2$. The diagram is best viewed on screen.

We measure classifier robustness as average geometric margin for each datapoint:

$$\bar{\gamma} = \frac{1}{m}\sum_{i=1}^{m} \gamma_i = \frac{1}{m}\sum_{i=1}^{m} \frac{y_i(\mathbf{w}^T\mathbf{x}_i + b)}{\|\mathbf{w}\|_2} \tag{6}$$

Robust classifier should have large average margin. We can observe similarities between definition of average margin in eq. (6) and soft-margin SVM objective in eq. (4). Soft-margin SVM formulation maximizes average functional margin and minimizes margin sensitivity which is numerator and denominator in eq. (6). This observation is informal but provides useful intuition which we will use later to generalize soft-margin maximization objective to deep neural networks.

A formal connection between soft-margin SVM and robust optimization was established in [15]. Xu [15, Theorem 3] showed that for non-separable dataset soft-margin SVM is equivalent to the following optimization problem:

$$\min : \max_{(\mathbf{r}_1,\ldots,\mathbf{r}_m)\in\mathcal{T}} \sum_{i=1}^{m} \Big(1 - y_i\left(\mathbf{w}^T(\mathbf{x}_i - \mathbf{r}_i) + b\right)\Big)_+ \tag{7}$$

where $\mathcal{T} = \left\{(\mathbf{r}_i, \ldots, \mathbf{r}_m) \,\middle|\, \sum_{i=1}^{m} \|\mathbf{r}_i\|^* \leq C\right\}$ is an uncertainty set. The above theorem holds for any l_p weight norm in soft-margin formulation and its dual norm in uncertainty set $\mathcal{T}$. To simplify our discussion, we consider l_2-norm geometric margin which is used in standard soft-margin SVM. Generalization to l_p-norm should be straightforward.

Next, we generalize binary soft-margin SVM to multiclass classification. Let $f(\mathbf{x}) = \mathbf{W}^T x + \mathbf{b}$ be a k-class classifier where $\mathbf{W} \in \mathbb{R}^{N \times k}$ and $\mathbf{b} \in \mathbb{R}^k$ are its parameters. Given the input $\mathbf{x}$, the prediction $\hat{y}$ is $\arg\max f(\mathbf{x})$. A multiclass classifier can be viewed as a collection of $\binom{k}{2}$ binary classifiers. A binary classifier for a pair of classes i and j is defined as $f_{i,j}(\mathbf{x}) = \mathbf{w}_{i,j}^T \mathbf{x} + b_{i,j}$ where $\mathbf{w}_{i,j}$ is the difference between columns i and j of weight matrix $\mathbf{W}$ and $b_{i,j}$ is the difference between class biases.

For a given input pair $(\mathbf{x}, y)$, its geometric margin is a minimum margin selected among all binary classifiers pairs:

$$\gamma = \min_{i \neq y} \frac{\mathbf{w}_{y,i}^T \mathbf{x} + b_{y,i}}{\|\mathbf{w}_{y,i}\|_2} \tag{8}$$

Geometrical margin γ for a multiclass classifier is the minimum distance to the side of the inscribed polyhedron.

We now formulate robust multiclass SVM using the definition of geometric margin in eq. (8):

$$\begin{aligned} \min \; & \sum_{i=1}^{k} C \max_{j \neq i, j > i} \|\mathbf{w}_{i,j}\|_2^2 + \sum_{i=1}^{m} \epsilon_i \\ \text{s.t. } & \mathbf{w}_{y_i,j}^T \mathbf{x}_i + \delta_{y_i,j} - b_{y_i,j} \geq 1 - \epsilon_i \quad \forall\, i, j \end{aligned} \tag{9}$$

where $\delta_{i,j}$ is a Kronecker delta, $\mathbf{w}_{i,j}$ is the difference between columns i and j in $\mathbf{W}$ and $b_{i,j}$ is the difference between class biases. The equivalence to the robust optimization problem can be shown using an argument similar to [15, Theorem 3] where the equivalence was proved for binary soft-margin SVM. Due to lack of space, we skip the proof here.

The closest to the proposed optimization problem is Crammer-Singer's multiclass SVM [26]. Crammer [26] considered minimizing average column difference of weight matrix $\mathbf{W}$ which is equivalent to minimizing matrix norm. By contrast, we propose to minimize maximum column difference. The distinction is important as our formulation is equivalent to the robust optimization problem.

IV. Deep Margin Maximization

Next, we develop margin maximization objective for deep neural networks. Similar to the discussion in the previous section, we first consider binary classification.

Let $f(\mathbf{x}; \mathbf{W})$, shorthand $f(\mathbf{x})$, be a neural network where $\mathbf{W}$ represents the network parameters. The class prediction $\hat{y}$ is $\operatorname{sign} f(\mathbf{x})$. Importantly, we consider that the output of the neural network is unnormalized. For a given datapoint $(\mathbf{x}, y)$, its l_2-margin, which is a minimum distance to the decision hyperplane, can be defined as follows:

$$\gamma = \min\{\|\mathbf{r}\|_2 \,|\, f(\mathbf{x} + \mathbf{r}) = 0\} \tag{10}$$

where $\mathbf{r}$ is the input perturbation.

For a linear classifier, such as SVM, geometric margin can be found using closed-form solution. Non-linear decision boundary of a neural network requires approximation. We consider a first-order approximation of the function level set:

$$\begin{aligned} & \min \|\mathbf{r}\|_2 \\ & \text{s.t. } f(\mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x})^T \mathbf{r} = 0 \end{aligned} \tag{11}$$

where $\nabla_{\mathbf{x}} f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ is the gradient of the output w.r.t. input. Similar function approximation is used in Deepfool [9] during computation of a minimal adversarial distortion.

The solution of eq. (11) can be found using a method of Lagrange multipliers. Then, l_2-distance to the decision boundary is:

$$\gamma = \frac{|f(\mathbf{x})|}{\|\nabla_{\mathbf{x}} f(\mathbf{x})\|_2} \tag{12}$$

And l_p geometric margin is:

$$\gamma = \frac{|f(\mathbf{x})|}{\|\nabla_{\mathbf{x}} f(\mathbf{x})\|_q} \tag{13}$$

where p and q such that $\frac{1}{p} + \frac{1}{q} = 1$. We can see from the above definition of the margin that *margin sensitivity* ϵ for non-linear classifier depends on the input $\mathbf{x}$ and is equal to the norm of the input gradient $\|\nabla_{\mathbf{x}} f(\mathbf{x})\|$.

Similar to the binary linear SVM, we propose soft-margin maximization objective for the binary neural network:

$$\min \sum_{i=1}^{m} C\|\nabla_{\mathbf{x}} f(\mathbf{x}_i)\|_2 + (1 - y_i f(\mathbf{x}_i))_+ \tag{14}$$

We can observe that the formulation in eq. (14) for deep linear networks reduces to the standard soft-margin SVM. Additionally, our formulation provides a novel margin interpretation of the methods [12], [22] which minimize the sum of squared input gradients.

Next theorem establishes a connection between the above objective and robust optimization for binary deep neural networks.

Theorem IV.1. *Let $\mathcal{T}_i = \{\mathbf{r}_i \,|\, \|\mathbf{r}_i\|^* \leq C\}$ be an uncertainty set where $\mathbf{r}_i$ is the perturbation for $\mathbf{x}_i$. Then, the optimization problem in eq.* (14) *approximately minimizes the following robust optimization problem:*

$$\min : \sum_{i=1}^{m} \max_{\mathbf{r}_i \in \mathcal{T}_i} (1 - y_i f(\mathbf{x}_i - \mathbf{r}_i))_+$$

Proof. Applying first-order approximation, we get:

$$\begin{aligned} & \sum_{i=1}^{m} \sup_{\mathbf{r}_i \in \mathcal{T}_i} (1 - y_i f(\mathbf{x}_i - \mathbf{r}_i))_+ \\ \approx & \sum_{i=1}^{m} \sup_{\mathbf{r}_i \in \mathcal{T}_i} \left(1 - y_i (f(x_i) - \nabla_{\mathbf{x}} f(x_i)^T \mathbf{r}_i)\right)_+ \\ \leq & \sum_{i=1}^{m} (1 - y_i f(x_i))_+ + \sup_{\mathbf{r}_i \in \mathcal{T}_i} (y_i \nabla_{\mathbf{x}} f(x_i)^T \mathbf{r}_i) \\ = & \sum_{i=1}^{m} (1 - y_i f(x_i))_+ + C\|\nabla_{\mathbf{x}} f(x_i)\| \end{aligned}$$

which completes the proof. □

Our proof mirrors the proof of [15, Theorem 3] where similar equivalence was established for binary soft-margin SVM. Unlike SVM, our result holds only in approximation. We can also observe that the above formulation generalizes

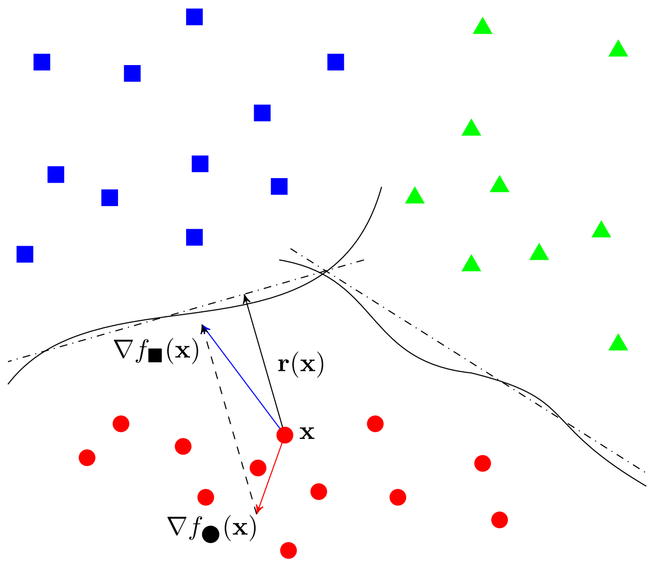

$\nabla f_{\blacksquare}(\mathbf{x})$
$\mathbf{r}(\mathbf{x})$
$\mathbf{x}$
$\nabla f_{\bullet}(\mathbf{x})$

Fig. 3. Margin maximization for a multiclass neural network. Prediction region $f(\mathbf{x}+\mathbf{r}) = f(\mathbf{x})$ is approximated using polyhedron. Adversarial perturbation $\mathbf{r}(\mathbf{x})$ is the minimum distance to its side. Our method minimizes *margin sensitivity* which is the norm of the difference between input gradient for the correct prediction $\nabla_{\mathbf{x}} f_{\bullet}(\mathbf{x})$ and the incorrect one $\nabla_{\mathbf{x}} f_{\blacksquare}(\mathbf{x})$ (showed in dashed line). The diagram is best viewed on screen.

soft-margin SVM. Indeed, soft-margin SVM for deep linear networks is equivalent to the proposed objective. Next, we generalize formulation of the robust multiclass SVM eq. (9) to deep neural networks.

Neural network $f(\mathbf{x})$ for k-class classification can be viewed as a collection of $\binom{k}{2}$ binary neural networks with shared parameters for each pair of classes. For a pair of classes i and j, the binary classifier can be defined as follows:

$$f_{i,j}(\mathbf{x}) = f_i(\mathbf{x}) - f_j(\mathbf{x}) \tag{15}$$

where $f_i(\mathbf{x})$ is the output of i-logits unit in the neural network. The prediction is class i if $f_{i,j}(\mathbf{x}) \geq 0$ and class j otherwise.

Combining with eq. (12), geometric margin between class i and j is given by:

$$\gamma_{i,j} = \frac{|f_i(\mathbf{x}) - f_j(\mathbf{x})|}{\|\nabla_{\mathbf{x}} f_i(\mathbf{x}) - \nabla_{\mathbf{x}} f_j(\mathbf{x})\|} \tag{16}$$

Then, geometric margin for an input $\mathbf{x}$ is a minimum margin among all class pairs given the target label y:

$$\gamma = \min_{y \neq j} \gamma_{y,j} = \min_{j \neq y} \frac{|f_y(\mathbf{x}) - f_j(\mathbf{x})|}{\|\nabla_{\mathbf{x}} f_y(\mathbf{x}) - \nabla_{\mathbf{x}} f_j(\mathbf{x})\|} \tag{17}$$

Geometric margin for multiclass deep neural network has an interpretation of the shortest distance to the side of the decision region which is approximated using polyhedron (see fig. 3). *Margin sensitivity* for multiclass deep neural network is the maximum norm of the difference between the input gradient of the correct prediction and the incorrect prediction. If datapoint $(\mathbf{x}, y)$ is classified correctly with the functional margin at least some constant C, we can maximize geometric margin by minimizing maximum margin sensitivity.

Using the above definition of geometric margin, we propose novel margin maximization objective for robust multiclass classification using deep learning as follows:

$$\min \sum_{i=1}^{m} \max_{j \neq y_i} \left(1 + f_{y_i}(\mathbf{x}_i) - f_j(\mathbf{x}_i)\right)_+ + C\epsilon(\mathbf{x}_i, y_i) \tag{18}$$

where $\epsilon(\mathbf{x}, y) = \max_{j \neq i} \|\nabla_{\mathbf{x}} f_i(\mathbf{x}) - \nabla_{\mathbf{x}} f_j(\mathbf{x})\|$ is the *margin sensitivity* for a given datapoint. In experiments, we considered minimization of l_2 and l_1 margin sensitivity. We note that minimizing l_1-norm corresponds to the maximization of l_∞-distance to the decision boundary.

Next theorem establishes an equivalence of the above objective and robust multiclass optimization.

Theorem IV.2. *Let $\mathcal{T}_i = \{\mathbf{r}_i \mid \|\mathbf{r}_i\|_* \leq C\}$ be an uncertainty set where $\mathbf{r}_i$ is the perturbation for $\mathbf{x}_i$. Then, the optimization problem in eq.* (18) *approximately minimizes the following robust optimization problem:*

$$\min : \sum_{i=1}^{m} \max_{\mathbf{r}_i \in \mathcal{T}_i; j \neq y_i} \left(1 + f_j(\hat{\mathbf{x}}_i) - f_{y_i}(\hat{\mathbf{x}}_i)\right)_+ \tag{19}$$

where $\hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{r}_i$ is the corrupted input.

Proof. Let $f_{i,j}(\mathbf{x}) = f_i(\mathbf{x}) - f_j(\mathbf{x})$ be the difference between the prediction for class i and j. Now, for any i, the following holds:

$$\begin{aligned}
&\sup_{\mathbf{r}_i \in \mathcal{T}_i} \max_{j \neq y_i} \left(1 - f_{j,y_i}(\mathbf{x}_i - \mathbf{r}_i)\right)_+ \\
&\approx \sup_{\mathbf{r}_i \in \mathcal{T}_i} \max_{j \neq y_i} \left(1 - f_{j,y_i}(\mathbf{x}_i) + \nabla_{\mathbf{x}} f_{j,y_i}(\mathbf{x}_i)^T \mathbf{r}_i\right) \\
&\leq \max_{j \neq y_i} \left(1 - f_{j,y_i}(\mathbf{x}_i)\right) + \sup_{\mathbf{r}_i \in \mathcal{T}_i} \left(\nabla_{\mathbf{x}} f_{j,y_i}(\mathbf{x}_i)^T \mathbf{r}_i\right) \\
&= \max_{j \neq y_i} \left(1 - f_{j,y_i}(\mathbf{x}_i)\right) + C\|\nabla_{\mathbf{x}} f_{j,y_i}(\mathbf{x}_i)\|
\end{aligned}$$

Since each perturbation r_i is independent, summing up for all i completes the proof. □

The proposed margin maximization objective has an interpretation of minimizing input sensitivity of the classifier prediction in the direction of the closest decision boundary. Besides, the connection between contractive penalty [12] and our regularization method can be shown. Recall the contractive penalty is defined as a Frobenius norm of the network Jacobian $\mathcal{J}(\mathbf{x})$. Now, consider l-layer neural network in which the last layer is fully-connected. Then, our margin maximization can be viewed as minimizing the projection of the $(l-1)$-layer Jacobian $(\mathbf{w}_j^{(l)} - \mathbf{w}_y^{(l)})\mathbf{J}_{l-1}(\mathbf{x})$ where $\mathbf{w}_j^{(l)}$ is the j-column in the last layer weight matrix and y is the correct label.

V. Experiments

All computations were conducted using Theano [27] which supports symbolic differentiation of complex expressions and computation on GPU.

We performed several experiments to study regularization effects and robustness of the proposed margin objective on

MNIST and CIFAR-10 datasets. Minimal adversarial perturbation $\mathbf{r}(\mathbf{r})$ was estimated using Deepfool algorithm [9]. And we measured average robustness of the neural network as follows:

$$\rho_{\text{adv}}(f) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \frac{\|\mathbf{r}(\mathbf{x})\|}{\|\mathbf{x}\|} \tag{20}$$

where $\mathcal{D}$ is the subset of test examples which were classified correctly. We note that Deepfool tends to overestimate a minimal adversarial perturbation for the regularized networks. To avoid this behavior, we clipped the norm of the perturbation at 0.5 for each iteration. The number of iterations was limited to 50.

In addition, we calculated test error on images corrupted using fast gradient sign method [5] for which adversarial direction is given as follows:

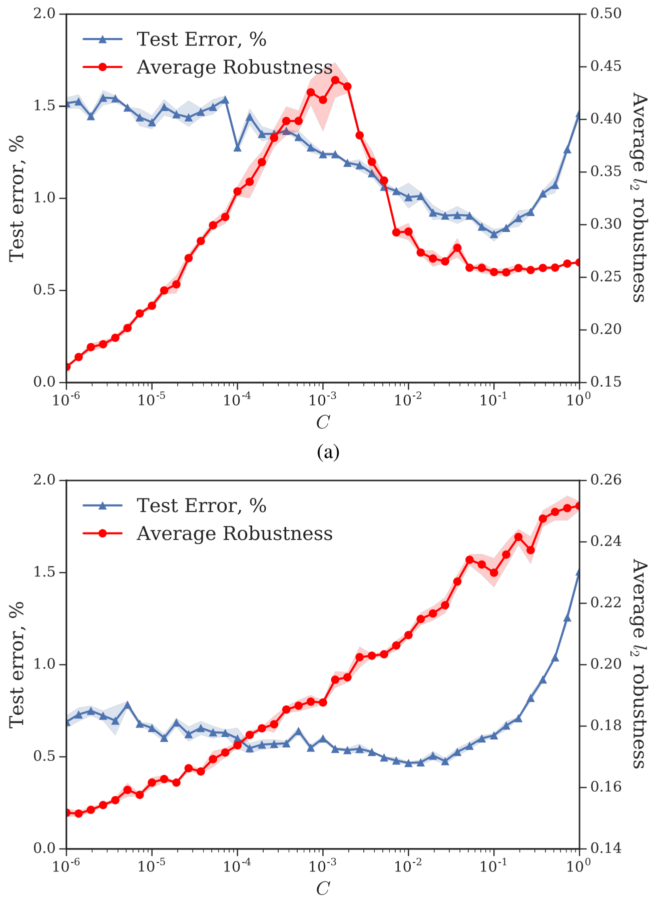
$$\mathbf{r}(\mathbf{x}) = \epsilon \operatorname{sign}(\nabla_{\mathbf{x}} \mathcal{L}(f, \mathbf{x}, y)) \tag{21}$$

where $\mathcal{L}$ is the categorical cross-entropy function.

A. MNIST experiments

MNIST is a dataset of handwritten digits images which consists of 60000 training and 10000 testing examples. Each image has 28×28 size and its corresponding label is one of the numerals from 0 to 9. We randomly split the original 60000 training examples into 50000 and 10000 samples used for training and validation respectively. We used validation dataset for tuning hyperparameters. The images were preprocessed to be in $[0, 1]$ range. No data augmentation was used in all experiments. We studied effects of the proposed regularization for two types of neural network: fully-connected network with three hidden layers of size $(1000, 1000, 1000)$, and convolutional network Lenet-5 [10]. We used Adam optimizer [28] with default learning rate and exponential learning decay with 0.95 decay rate. All networks were trained for 100 epochs with minibatch of 100 examples.

First, we studied the dependency between test error and network robustness for different values of regularization coefficient C. We trained Lenet-5 and fully-connected neural network with l_2 margin regularization for $C \in [1.0, 10^{-6}]$. Average robustness and test error were computed for three separate runs with different random seeds. In this experiment, we considered only l_2 margin maximization as we observed that training with l_1 margin sensitivity is unstable for $C \geq 0.1$ which suggest that it has stronger regularization effect. The results are shown in Figure 4. We can observe in fig. 4b that the robustness of convolutional neural network Lenet-5 smoothly varies as we decrease the strength of our regularization. For the fully-connected network, we visually examined adversarial examples for $C \in [10^{-4}, 10^{-3}]$ and observed that, while the robustness is high, adversarial examples are not visually meaningful. This suggests that the extreme values of the robustness is the result of a failure of Deepfool algorithm [9] to find small adversarial perturbation. Using validation dataset, we selected optimal value of $C = 0.1$ for l_2-margin regularization and used $C = 0.01$ for l_1-margin regularization.



(a)

(b)

Fig. 4. Test error and network robustness for values of C in interval $[1, 10^{-6}]$. Figure 4a shows the dependency between test error and robustness for fully-connected neural network and Figure 4b for convolutional neural network Lenet-5. See full text for details.

Next, we performed a comparison of our l_1- and l_2 margin maximization method against baseline (no regularization), dropout [29], adversarial training (AT) [5], virtual adversarial training (VAT) [11]. For AT [5], we used fixed $\epsilon = 0.2$ which we selected using validation dataset. For VAT [11], we used implementation provided by the authors which at the moment of this publication supports training only of fully-connected neural networks. For our method, we considered l_2 and l_1 margin regularization with $C = 0.1$ and $C = 0.01$ respectively. We trained neural networks for ten separated runs with different random seeds. We reported results for test error, test error on images distorted using FastGrad [5] with $\epsilon = 0.2$ and average robustness. The results are shown in Table I. We can observe that three hidden layer MLP noticeably overfits without any regularization. The proposed margin objective and VAT significantly reduces overfitting and achieves high robustness scores. AT [5] performs the best on images corrupted using fast gradient sign method but is noticeably less robust. Arguably, this is the result of overfitting to the particular noise type when using data regularization. Baseline convolutional

Network	Error %	Adv. Error %	$\rho_{\text{adv}} \times 10^{-1}$
MLP baseline	1.42 ± 0.08	95.8 ± 1.89	1.14 ± 0.01
Dropout [29]	1.34 ± 0.05	93.4 ± 2.08	1.20 ± 0.01
AT [5]	1.19 ± 0.06	$\mathbf{10.17 \pm 0.69}$	1.60 ± 0.05
VAT [11]	$\mathbf{0.87 \pm 0.04}$	24.33 ± 1.38	$\mathbf{2.69 \pm 0.02}$
Our l_1	$\mathbf{0.84 \pm 0.03}$	32.43 ± 1.25	$\mathbf{2.73 \pm 0.08}$
Our l_2	$\mathbf{0.86 \pm 0.04}$	42.56 ± 1.37	2.59 ± 0.05
Lenet-5 baseline	0.72 ± 0.06	72.14 ± 2.20	1.54 ± 0.04
Dropout [29]	$\mathbf{0.58 \pm 0.03}$	61.66 ± 3.45	1.70 ± 0.05
AT [5]	0.73 ± 0.05	$\mathbf{4.95 \pm 1.26}$	2.00 ± 0.03
Our l_1	0.64 ± 0.02	20.47 ± 2.52	$\mathbf{2.22 \pm 0.05}$
Our l_2	0.62 ± 0.04	28.26 ± 3.49	$\mathbf{2.17 \pm 0.06}$

TABLE I
RESULTS ON MNIST DATASET FOR FULLY-CONNECTED NETWORK (MLP) AND CONVOLUTIONAL NETWORK (LENET-5).

network Lenet-5 is significantly more robust compared to the fully-connected network. We think this is due to using prior information about image structure in convolutional layers, e.g. translation invariance. We also note that minimization of l_1 margin sensitivity performs consistently better than l_2 margin loss on images corrupted using fast gradient sign. This result is in line with our theoretical analysis because l_1 margin objective maximizes the robustness to l_∞ perturbation.

In addition, we compared histograms of adversarial noise for the networks trained with different regularization strategies. Distribution histograms are shown in Figure 5. Our margin regularization and VAT method significantly improve the robustness of the models. However, even after the regularization was applied, the networks remain relatively vulnerable to small perturbations $\rho_{\text{adv}} \leq 0.1$. Nonzero density for extreme values of the robustness suggests that Deepfool algorithm [9] often fails to estimate small adversarial distortion for the strongly regularized networks. Lastly, we visually examined the quality of the generated adversarial images. Ideally, adversarial examples for the robust neural network should match human perception of the visually confusing images. Adversarial examples are shown in Figure 6. We can observe that adversarial noise convey a certain visual meaning. For example, digit "4" was changed to digit "9" by adding an upper stroke. Similarly, digit "1" was changed to digit "7" or digit "4". Thus, as the result of our regularization, our algorithm improves improve the robustness of neural networks both quantitatively and qualitatively.

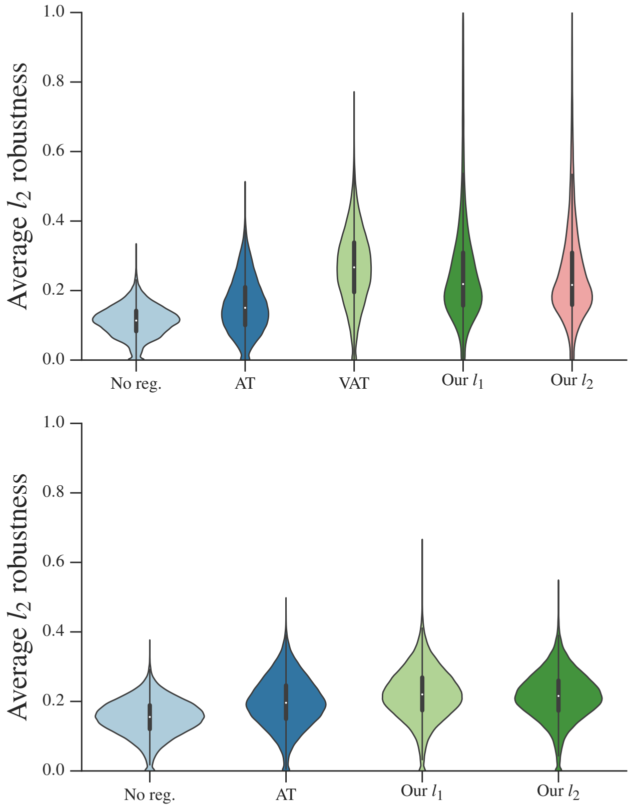


Fig. 5. Robustness ρ_{adv} for different regularization strategies. The distributions were estimated using kernel density estimator and the values for the robustness were computed for 10 different runs. The top plot shows robustness for multilayer perceptron and the bottom plot for Lenet-5 network.

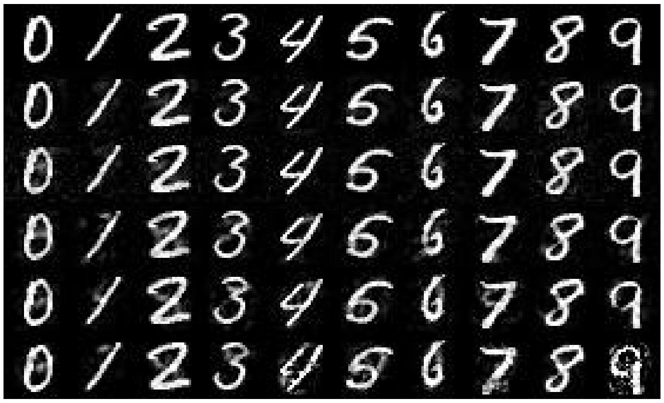


Fig. 6. Adversarial examples for fully-connected neural network. First row shows the original images. Rows from second to the bottom show adversarial examples for the network trained with no regularization, AT [5], VAT [11], l_2-margin and l_1-margin.

B. CIFAR-10 experiments

CIFAR-10 is a 10-class dataset of 32×32 which consists of 50000 training and 10000 testing images. We used last 5000 examples from training dataset for validation. Data was preprocessed using global contrast normalization and ZCA whitening which is commonly applied to this dataset. We trained Network in Network (NIN) [30] for 200 epochs with batch size 128 using stochastic gradient descent with momentum 0.9. Learning rate was divided by 5 at epochs $[80, 120, 160]$. Image flipping and random translation were used for data augmentation. Our reference implementation achieved 9.10% with data augmentation and 11.30% without. We found that NIN with multiclass hinge loss and our margin objective is unstable and often diverges. Due to its instability, we considered training NIN with categorical cross-entropy. We disabled weight decay when training with our margin maximization objective as it was increasing validation error.

Network	Error %	Adv. Error %	$\rho_{\text{adv}} \times 10^{-2}$
NIN w/o data augm.	11.30	68.85	5.12
NIN l_2, $C = 0.01$	11.13	**55.88**	**6.83**
NIN l_2, $C = 0.001$	**10.42**	61.53	6.45
NIN with data augm.	**9.10**	65.98	5.41
NIN l_2, $C = 0.005$	10.02	**52.80**	**7.77**

TABLE II
RESULTS ON CIFAR-10 FOR NETWORK IN NETWORK.

We report test error on clean and corrupted images using fast gradient sign [5] with $\epsilon = 0.05$ and average robustness computed using eq. (20). The results for CIFAR-10 dataset are reported in Table II. We observe a small improvement in test and adversarial accuracy and robustness with our regularization method. In future work, we plan to address scalability of the proposed method to more complex datasets.

VI. CONCLUSION

We proposed a novel regularization objective for neural network inspired by margin maximization principle. We theoretically proved that the proposed margin objective approximately minimizes robust optimization problem. In experiments, we showed that the introduced regularization method substantially improves the robustness of neural networks both quantitatively and qualitatively and reduces overfitting. Adversarial examples for robust neural network should be confusing for a human observer. Ideally, future work should consider how human perceive visually confusing images as a reference to compare regularization methods with. Another issue which we plan to address is the scalability of the proposed method to more complex datasets and larger network models.

ACKNOWLEDGMENTS

This research is supported by Singapore International Graduate Award. We also thank NVIDIA Corporation for donating hardware used in this research.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1106–1114.

[2] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[3] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1653–1660.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[6] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the Robustness of Deep Neural Networks via Stability Training," *ArXiv e-prints*, Apr. 2016.

[7] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, 2016, pp. 372–387.

[8] H. Xu and S. Mannor, "Robustness and generalization," *Machine Learning*, vol. 86, no. 3, pp. 391–423, 2011.

[9] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2574–2582.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[11] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional Smoothing with Virtual Adversarial Training," in *International Conference on Learning Representations (ICLR)*, 2015.

[12] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 2011, pp. 833–840.

[13] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[14] Y. Tang, "Deep learning using linear support vector machines," in *Workshop on Representational Learning, ICML*, 2013.

[15] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1485–1510, 2009.

[16] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, "Measuring neural net robustness with constraints," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2613–2621.

[17] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial Manipulation of Deep Representations," in *International Conference on Learning Representations (ICLR)*, 2016.

[18] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015, pp. 427–436.

[19] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," in *International Conference on Learning Representations (ICLR), Workshop Contribution*, 2014.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *ArXiv e-prints*, Mar. 2015.

[21] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, 2016, pp. 582–597.

[22] H. Drucker and Y. L. Cun, "Double backpropagation increasing generalization performance," in *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, vol. ii, Jul 1991, pp. 145–150 vol.2.

[23] C. Caramanis, S. Mannor, and H. Xu, "Robust optimization in machine learning," in *Optimization for machine learning*, S. Sra, S. Nowozin, and S. J. Wright, Eds. Cambridge, MA, USA: Mit Press, 2012, pp. 369–402.

[24] P. Tabacof and E. Valle, "Exploring the space of adversarial images," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 426–433.

[25] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 1632–1640.

[26] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," *Machine Learning*, vol. 47, no. 2, pp. 201–233, 2002.

[27] The Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *ArXiv e-prints*, May 2016.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv e-prints*, Dec. 2014.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[30] M. Lin, Q. Chen, and S. Yan, "Network In Network," in *International Conference on Learning Representations (ICLR)*, 2014.