Developing Robust and Lightweight Adversarial Defenders by Enforcing Orthogonality on Attack-Agnostic Denoising Autoencoders

Aristeidis Bifis
University of Patras,
Patras, GR
bifis@ceid.upatras.gr

Emmanouil Z. Psarakis
University of Patras,
Patras, GR
psarakis@ceid.upatras.gr

Dimitrios Kosmopoulos
University of Patras,
Patras, GR
dkosmo@upatras.gr

Abstract

Adversarial attacks have become a critical threat to the security and reliability of machine learning models. We propose a solution to the problem of defending against adversarial attacks using a deep Denoising Auto Encoder (DAE). The proposed DAE is trained to enforce orthogonality between the noise and the range space of its output in each layer of the encoder's chain. Furthermore, the pseudoinverse decoder of the DAE is designed to ensure that the reconstructed image and the null space of its intermediate representations in each layer of the chain maintain orthogonality as it progresses from the target space to the latent space. The denoising problem is formulated as an equality constrained optimization problem, which is solved by finding the stationary points of the Lagrangian function. The noisy data are generated by adding realizations of multiple random noise distributions to pristine data during DAE training, resulting in excellent denoising performance. We compare the performance of our full weights and tied-weights DAEs, showing that the latter not only has half the complexity of the former, but also outperforms the former in denoising and in strong adversarial attacks. To demonstrate the effectiveness of the proposed solution we evaluate our networks against recent works in the literature, specifically those focusing on defending against adversarial attacks.

1. Introduction

The problem of correctly classifying unlabeled data, especially in the domain of image processing, is an interesting one. Alongside the flourishing of classification techniques, mainly with the utilization and progress of deep learning, a new set of attacks [8], [25], [4], [14], [11], [6], [16] have been proposed, that aim at exposing the fragility of classification schemes. These attacks focus on adding a perturbation on the test unlabeled data, during inference, to force the trained classifiers at misinterpreting them and predicting a false label. The perturbation that is added to the test data usually is limited to have small amount of energy, measured by its norm, while maximizing the effect of the attack confirmed by the drop in classification accuracy.

As a means to defend against attacks on the classification accuracy, we can view the problem at hand as a noise attack, where the tampered image consists of the original data with added noise of unknown distribution. Denoising systems, particularly DAEs, have demonstrated significant success in extracting valuable information from noisy data and reconstructing clean data [7]. This has been observed in various domains where the noise in the data is caused by adverse data acquisition conditions or faulty acquisition systems. Additionally DAEs can in fact boost the performance of subsequent classifiers, in a purely unsupervised fashion [28]. In [27] the authors propose a training principle for unsupervised learning of DAEs that is based on the idea of making the learned representations robust to partial corruption of the input pattern. These DAEs can be stacked to initialize deep architectures.

In this paper we try to tackle such attacks with a simple defence mechanism, which assumes no prior knowledge of the tested attack scheme, thus adopting the term attack-agnostic and without utilizing any form of adversarial training (e.g., [8] [21]) or ensemble adversarial training (e.g., [25]) for the defender or the classifier. The defence mechanism aims at modeling the problem of classifying tempered data, by hypothesizing that a tempered image can be seen as a noisy perturbation of the pristine one and thus, we can utilize a denoising mechanism that is proven to be robust at such modeling.

2. Related work

The most effective tampering methods against trainable classifiers, are the adversarial attacks [23]. Adversarial attacks are not only contained within the context of affecting the classification accuracy. Malicious individuals can target machine learning systems during inference by tampering with the input data [2]. These systems can affect neural networks in various conditions, where depending on the level of prior knowledge of the classifier's weights and architecture, or even the training data, adversarial attack setups can be categorized as white-box, semi-white-box (or gray-box) and black-box.

To defend against adversarial attacks on classification systems many defence mechanisms have been proposed so far [25], [8], [14], [24], [29], [32], [15], [19], [22].

Most notably in [18] authors empirically show that in a new training technique for the classifiers, called distillation [9], the model is less susceptible to adversarial inputs. The classification architecture consists of two identical networks, where each one of them produces "soft" and "hard" labels. The "soft" labels produced from the first network are fed as input to the second network, whose final layer is a modified version of softmax, to achieve the same level of accuracy. However such defence mechanisms, with the recent advancement in adversarial attacks, can be easily avoided [17], [4].

One of the most famous adversarial attacks of recent years is produced by a generative neural network known as AdvGAN [30]. AdvGAN is trained in an adversarial fashion to produce noise perturbations with a small energy, that when added to pristine data, degrades drastically the accuracy of well performing classifiers. The architecture of AdvGAN consists of a generator which takes as input a pristine image data and produces the noise perturbation. The perturbation is added to the original image and the result is passed to the system's discriminator. The two networks are trained in an adversarial fashion, such that the resulting adversaries are not typically perceived as contaminated by the human eye, while simultaneously the targeted classifier performs poorly.

Some more recent works on defenses against adversarial attacks are presented in [10, 13, 12]. Specifically, in [10], the authors produce per-pixel deep features and use the features in the neighborhood of query pixel for predicting the clean RGB value. In [13] the authors comprise two components, an optimized friendly noise that is generated to maximally perturb examples without degrading the performance, and a randomly varying noise component. Finally in [12], the authors learn defense transformations to counterattack the adversarial examples by parameterizing the affine transformations and exploiting the boundary information of DNNs.

DAEs have already been used as a defence mechanism against adversarial attacks. Authors in [1], propose to add a deep denoising sparse autoencoder (DDSA) as a preprocessing block before any classification model. In their proposed scheme a sparsity constraint is added to the Fully Connected (FC) layers of the DDSA block, to force neurons that produce the latent output to be inactive most of the time, in order to extract only meaningful and relevant features. Specifically, the sparsity constraint allows the activations of hidden units to be equal to some target activation such as the one of the pristine data.

In [24] a use of Probabilistic Adversarial Robustness (PAR) as a fundamental approach to neutralize adversarial attacks is proposed. The concept is to utilize the application loss function to guide the probabilistic model in projecting adversarial examples to the adversarial-free zones. The PAR framework and its implementation via ShieldNets is designed to provide proactive protection to an existing fixed model.

Adopting the pipeline proposed in [1] (lower branch shown in Figure 1) for fighting the adversarial attacks, in this work we propose the use of a DAE trained such that, on average, in each layer of the encoder's chain the noise is enforced to be orthogonal to the output's **range**. Moreover, the pseudoinverse decoder of the DAE is defined and in each layer of its chain the reconstructed image and its images as the chain proceeds from the target space to the latent one, is enforced, on average, to be orthogonal to the output's **null-space**. The solution is based on an equality constrained optimization problem thus admitting the use of Lagrange multipliers based techniques. For the training of the DAE, the noisy data is generated by adding to the pristine data realizations of multiple random noise distributions (upper branch in the first block shown in Figure 1). Imposing the proposed constraints, we have trained full weights and tied-weights DAEs and the later not only has half the complexity of the former but shows significant improvements in its performance in both denoising and the adversarial attack.

3. Problem formulation

It is well known that the aim of a DAE is to learn lower-dimensional ”noise free” representations of higher-dimensional noisy data, and to remap them onto a space of the same dimensionality of the original ones. This is achieved by distilling or capturing the most important parts of the attacked (noisy) input images through the encoder and then remapping them, via the decoder, into a domain of the same dimensionality that in our case is a manifold that contains the pristine images. Let us formally define, with the help of Figure 2, the problem we are interested in, that can be summarized as follows: We are given two multivariate Random Variables (RVs). The first one belongs to the **source** domain $\mathbb{S}$ and follows a multivariate probability density $g_{\mathcal{X}_w}(\mathbf{x})$ and the second belongs to the **target** domain $\mathbb{T}$ with the multivariate density $g_{\mathcal{X}}(\mathbf{x})$, which is of the same dimensionality with $g_{\mathcal{X}_w}(\mathbf{x})$ and is the pdf of the pristine images. The most interesting and difficult denoising problems arise when the density function $g_{\mathcal{X}_w}(\mathbf{x})$ is unknown. On the other hand the density $g_{\mathcal{X}}(\mathbf{x})$ in most cases is considered known. Our goal is to find two (2) deterministic transformations $E(.)$ and $D(.)$ so that by applying the following composition of transformations:

$$\mathcal{X} = D\big(E(\mathcal{X}_w)\big) \qquad (1)$$

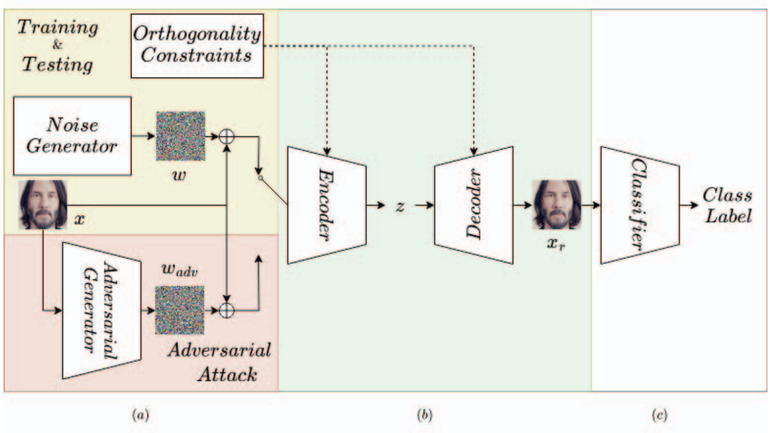

Figure 1. The pipeline of the training-testing & adversarial attack complete process, consisting of the training & adversarial attack blocks (a), the defence system (b) and the classifier under attack (c)

the RV $\mathcal{X}_w = \mathcal{X} + \mathcal{W}$, that describes the noisy data, to be transformed onto the RV $\mathcal{X}$, whose distribution should be the target density $g_{\mathcal{X}}(\mathbf{x})$ of the pristine data. Of course one may wonder whether the proposed problem enjoys any solution, namely, whether indeed there exists a composition of transformations $D\big(E(\mathbf{x})\big)$ capable of transforming the RV $\mathcal{X}_w$ into $\mathcal{X}$ with the former following the source density $g_{\mathcal{X}_w}(\mathbf{x})$ and the latter the target density $g_{\mathcal{X}}(\mathbf{x})$. The problem of transforming random vectors has been analyzed in [3], where existence is shown under general conditions.

We are going to limit the class of permissible transformations $E(.), D(.)$ into the class of the deep Autoencoders. Therefore the transformation $E(.)$ will be replaced by $E(\mathcal{X};\ \mathcal{P}_E)$ while $D(.)$ by $D(\mathcal{Z};\ \mathcal{P}_D)$ where $\mathcal{P}_E$, $\mathcal{P}_D$ contain the parameters of the two DAE components. More formally, let:

$$\begin{aligned} \mathcal{Z} &= E(\mathcal{X}_w;\ \mathcal{P}_E) : \mathbb{S} \subseteq [0,\ 1]^N \rightarrow \mathbb{L} \subseteq \mathbb{R}^M \text{ and} \\ \mathcal{X} &= D(\mathcal{Z};\ \mathcal{P}_D) : \mathbb{L} \subseteq \mathbb{R}^M \rightarrow \mathbb{T} \subseteq [0,\ 1]^N \qquad (2) \end{aligned}$$

be the **encoder** and **decoder** of the DAE respectively with $\mathcal{X}_w,\ \mathcal{Z},\ \mathcal{X}$ random variables (RVs) of the **source** $\mathbb{S}$, **latent** $\mathbb{L}$ and **target** domain $\mathbb{T}$ respectively, and $\mathcal{P}_E,\ \mathcal{P}_D$ the sets of the encoder's and decoder's parameters respectively. Each one of the above mentioned sets contains the weights $\mathcal{W}$, as well as the parameters α of the activation functions of each layer of the network, that is:

$$\begin{aligned} \mathcal{P}_I &= \Big\{\mathcal{W}_I,\ \mathcal{F}_I\Big\}, \qquad \text{with } \mathcal{W}_I = \Big\{W_{I_l}\Big\}_{l=1}^{L} \qquad \text{and} \\ \mathcal{F}_I &= \Big\{f_{I_l}(\ .\ ; \alpha_{I_l})\Big\}_{l=1}^{L-1} \qquad (3) \end{aligned}$$

with the subscript I taking values from the set $\mathcal{S}_I = \{E, D\}$. For a detailed description of the AE's architecture, please see Section 4. Note that the encoder of the DAE can be considered as a mapping of the source domain $\mathbb{S}$, which, as we said, in our case is the domain of the contaminated images, onto the latent space $\mathbb{L}$; the decoder is a mapping of the $\mathbb{L}$ one onto the target domain $\mathbb{T}$, which has to coincide with that of the pristine images.

Having defined the DAE we can concentrate ourselves to a critical issue that is related to the invertibility of the decoder. More specifically, under some mild assumptions, as we are going to see, we can define an inverse mapping that can be used for solving, in an efficient way, the image denoising problem via the use of DAE. This issue is investigated in the following proposition.

Proposition 1: Let $\mathcal{W}_D,\ \mathcal{F}_D$ be the sets contained in the decoder's parameters set $\mathcal{P}_D$ defined in Eq. (3). If the activation functions of set $\mathcal{F}_D$ are **invertible** and the tall matrices of set $\mathcal{W}_D$ are of full column rank we can define the following **pseudoinverse decoder's** mapping defined from **target** domain $\mathbb{T}$ into the **latent** space $\mathbb{L}$:

$$\tilde{\mathcal{Z}} = D^{\dagger}(\mathcal{Y}; \mathcal{P}_{D^{\dagger}}) \qquad (4)$$

with:

$$\begin{aligned} \mathcal{W}_{D^{\dagger}} &= \Big\{W^{\dagger}_{D_l} = (W^T_{D_l} W_{D_l})^{-1} W^T_{D_l}\Big\}_{l=1}^{L} \text{ and} \\ \mathcal{F}_{D^{-1}} &= \Big\{f_{D^{-1}{}_l}(\ .\ ; \alpha_{D_l}) = f^{-1}_{D_l}(\ .\ ; \alpha_{D_l})\Big\}_{l=1}^{L-1} . \qquad (5) \end{aligned}$$

Proof: The proof is easy and is omitted. □

Note that the above defined mapping is a many-to-one mapping. Indeed, if we define the **null-manifold** of the above defined mapping by extending the definition of the **null-space**[1] of a matrix, that is:

$$\mathcal{N}(D^{\dagger}) = \{\mathbf{x} \in\ \mathbb{T} \,\big|\, D^{\dagger}(\mathbf{x})\ =\ \mathbf{0}_M \in \mathbb{L}\} \qquad (6)$$

[1] The **range** and the **null-space** or **kernel** of a size $N \times M$ matrix $A : \mathbb{R}^N \rightarrow \mathbb{R}^M$, denoted by $\mathcal{R}(A)$ and $\mathcal{N}(A)$ respectively, are defined

we can easily prove our claim. This manifold as well as the **range** of the mapping:

$$\mathcal{R}(D^{\dagger}) = \left\{\mathbf{x} \in \ \mathbb{T} \,\middle|\, D^{\dagger}(\mathbf{x}) \ \neq \ \mathbf{0}_M \in \mathbb{L}\right\} \tag{7}$$

will play a vital role in the proposed denoising technique.

4. Deep Autoencoder Architecture

Let us define the encoder and decoder of a deep architecture Autoencoder. An **encoder** of deep architecture composed by L-layers, can be defined, in a recursive way, as follows (for the basic building block of such a system, please see Figure 2.(a)):

$$\begin{aligned} \xi_0 = \mathcal{X}, \quad \xi_l &= f_{E_l}\big(W_{E_l}\xi_{l-1}\big),\ l = 1, 2, \cdots,\ L-1 \\ \text{and } \xi_L &= W_{E_L}\xi_{L-1}. \end{aligned} \tag{8}$$

Note that its first layer coincides with the DAE's input while its output feeds the decoder's input, that is $\xi_L \equiv \mathcal{Z}$. As it was mentioned the encoder maps the elements of the **source** domain $\mathbb{S}$ onto the **latent** or **code** space $\mathbb{L}$.

A **decoder** of similar architecture, can be defined as follows (for the basic building block of such a system, please see Figure 2.(b)):

$$\begin{aligned} \zeta_0 = \mathcal{Z}, \quad \zeta_l &= f_{D_l}\big(W_{D_l}\zeta_{l-1}\big),\ l = 1, 2, 3, \cdots,\ L-1 \\ \text{and } \quad \zeta_L &= W_{D_L}\zeta_{L-1} \end{aligned} \tag{9}$$

with $\zeta_L \equiv \mathcal{Y}$, that is the DAE's output. Note also that both the encoder's and decoder's outputs ξ_l, ζ_l of each layer are multivariate random variables whose dimensions are specified by the dimension of the row space of the corresponding matrix (let say M_l). We also assume that the dimensions of the encoder's cascaded layers as well as the decoder's ones, are of the appropriate size so that the connectivity between the layers to be ensured. The **pseudoinverse-decoder** of the above defined decoder, whose parameters are already defined in proposition 1, will be a useful tool for solving the problem at hand, and can be defined as follows:

$$\begin{aligned} \tilde{\zeta}_0 &= \mathcal{Y}, \\ \tilde{\zeta}_l &= f^{-1}_{D_{L-l}}\big(W^{\dagger}_{D_{L+1-l}}\tilde{\zeta}_{l-1}\big),\ l = 1, 2, \cdots,\ L-1, \\ \text{and } \tilde{\zeta}_L &= W^{\dagger}_{D_1}\tilde{\zeta}_{L-1} \equiv \tilde{\mathcal{Z}}. \end{aligned} \tag{10}$$

We must stress at this point that this mapping, as we mentioned, is many-to-one, and this is not desirable.

5. Proposed Training Methodology

As it is explained in details in the previous section, in a $2L$-layer autoencoder, the encoder and the decoder can be considered as composition machines. Thus, via the encoder is created a chain of L multivariate RVs whose dimensions, are reduced as we proceed from the **source** domain to the **latent** one due to the **bottleneck** effect. Due to this dimensionality reduction and using the notion of the **null-space** defined in Eq. (6) of Section 3, the resulting mapping is many-to-one. On the decoder side there is a "mirror" chain where, the dimensionality of the RVs increases as we proceed from the **latent** space to the **target**, so the achieved mapping is one-to-many.

Ideally, we would like each sample of RV $\mathcal{X}_r$ that constitutes the representation of the denoised (reconstructed) images to be orthogonal to the **null-space** of each link of the chain formed by the pseudoinverse decoder's mapping. On the other hand, each sample of the RV $\mathcal{X}_w - \mathcal{X}_r$, that constitutes the total error, that is the noise plus the reconstruction error, ideally should be orthogonal to the **range** of each link (layer's output) of the chain formed by the encoder.

Note that if both the above mentioned constraints are imposed during the training phase of the net, the noise will be mapped into the **null-space** of the matrices W_{E_l} while the reconstructed images into the **range** of matrices W_{D_l} respectively. However, since it is not possible to be achieved for all the samples, we are going to achieve it in the mean sense.

The above mentioned requirements are general and do not impose any kind of symmetry restriction. Indeed, in the next subsection we formulate a fully parameterized DAE by imposing the aforementioned orthogonality requirements on the encoder and the pseudoinverse decoder.

6. Deep DAE

In this section we are going to impose the orthogonality requirements on the encoder and the pseudoinverse decoder and formulate the training problem of a full and a tied-weights parameters DAE.

6.1. Full Parameters Deep DAE

To this end, let us express the necessary orthogonality constraints as functions of its parameters. Let $V_{I_l} = [V_{I_{l_{\mathcal{R}}}},\ V_{I_{l_{\mathcal{N}}}}],\ l = 1, 2, \cdots, L$ be the orthonormal bases of $\mathbb{R}^{N_l},\ l = 1, 2, \cdots, L$ resulting from the SVD of matrices $W_{I_l},\ l = 1, 2, \cdots, L$ with the subscript I taking values from the set $\{E,\ D^{\dagger}\}$, that is the matrices of the encoder and the pseudoinverse decoder and matrices $V_{I_{l_{\mathcal{R}}}}$, $V_{I_{l_{\mathcal{N}}}}$ corresponding to the bases of the **range** and the **null-space** of W_{I_l} respectively. In addition, let $\bar{\xi}_l, l = 0, 1, \cdots\ , L$ the sequence of RVs that are produced by the **encoder** using the Eq. (8) in the paper when the RV $\mathcal{X}_w - \mathcal{X}_r$ is fed in its input; RV $\mathcal{X}_w$ constitutes the representation of the attacked

by the following relations:

$$\begin{aligned} \mathcal{R}(A) &= \{\mathbf{x} \in \mathbb{R}^N \,|\, A\mathbf{x} \neq \mathbf{0} \in \mathbb{R}^M\}, \\ \mathcal{N}(A) &= \{\mathbf{x} \in \mathbb{R}^N \,|\, A\mathbf{x} \ = \ \mathbf{0} \in \mathbb{R}^M\} \end{aligned}$$

.

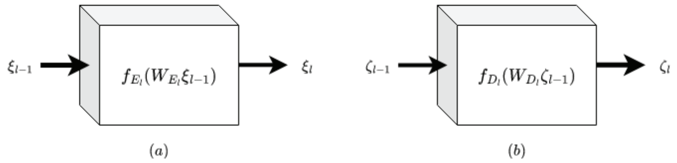

Figure 2. The building blocks of the Encoder (a) and the Decoder (b) of an Autoencoder

images, whose pdf is unknown, and $\mathcal{X}_r$ the reconstruction of the attacked image; the $\mathcal{X}_r$ pdf, ideally, should coincide with that of the pristine, which either is known or can be estimated from the given training set of the pristine images. Finally, let $\tilde{\zeta}_l,$, $l = 0, 1, 2, \cdots, L$ the sequence of RVs that are produced by the **pseudoinverse decoder**, defined in Eq. (10) when the RV $\mathcal{X}_r$ is fed in its input.

Having defined all the necessary quantities, in the next lemma we present the constraints that must be satisfied by the weights of the DAE to enforce the desired orthogonality.

Lemma 1: Let $V_{E_{l_\mathcal{R}}}$ be a base of the **range** of the encoder matrix W_{E_l} and $V_{D^\dagger_{l_\mathcal{N}}}$ a base of the **null-space** of the pseudoinverse decoder matrix $W_{D^\dagger_l}$. Then, the following constraints:

$$\begin{aligned} V^T_{E_{l_\mathcal{R}}} \mathbb{E}_{\bar{\xi}_{l-1} \sim g_{\bar{\xi}_{l-1}}} \left[\bar{\xi}_{l-1} \right] &= \mathbf{0}_{M_l} \\ V^T_{D^\dagger_{l_\mathcal{N}}} \mathbb{E}_{\tilde{\zeta}_{l-1} \sim g_{\tilde{\zeta}_{l-1}}} \left[\tilde{\zeta}_{l-1} \right] &= \mathbf{0}_{N_l - M_l},\ l = 1, 2, \cdots, L \end{aligned} \tag{11}$$

with $\mathbb{E}_{\bar{\xi}_{l-1} \sim g_{\bar{\xi}_{l-1}}}[.]$ denoting the expectation operator over $\bar{\xi}_{l-1}$ and $\mathbb{E}_{\tilde{\zeta}_{l-1} \sim g_{\tilde{\zeta}_{l-1}}}[.]$ denoting the expectation operator over $\tilde{\zeta}_{l-1}$, which ensure, on average, that the noise is mapped into the **noise subspace** and the reconstructed images into the **signal** one.

□

Based on **Lemma 1**, we propose the following optimization problem:

$$\begin{aligned} \min_{\mathcal{P}_E,\, \mathcal{P}_D} \quad & \mathbb{E}_{\mathcal{X} \sim g_\mathcal{X}} \left[||\mathcal{X} - \mathcal{X}_r||^2_2 \right] \\ s.t. \quad & \sum_{l=1}^{L} ||V^T_{E_{l_\mathcal{R}}} \mathbb{E}_{\bar{\xi}_{l-1} \sim g_{\bar{\xi}_{l-1}}} \left[\bar{\xi}_{l-1} \right] ||^2_2 = 0, \\ & \sum_{l=1}^{L} ||V^T_{D^\dagger_{l_\mathcal{N}}} \mathbb{E}_{\tilde{\zeta}_{l-1} \sim g_{\tilde{\zeta}_{l-1}}} \left[\tilde{\zeta}_{l-1} \right] ||^2_2 = 0, \\ & \qquad l = 1, 2, \cdots, L \end{aligned} \tag{12}$$

for solving the denoising and adversarial attack problem.

We must stress at this point that in the constrained optimization (Eq. (12)), the first equality constraint refers to the encoder's weights while the other to the pseudoinverse decoder ones, and consequently the computational cost of its solution is heavier since the computation of the SVD of both of the above mentioned matrices is needed. For the solution of the above defined problem we follow a similar procedure with that presented in the main paper for the solution of the tied-weights Deep DAE counterpart. This concludes our presentation for the full DAE. In the following sub-section we concentrate ourselves on the tied-weights DAE, which offers almost the same performance and much smaller parameter space.

6.2. Tied-Weights Deep DAE

Let us constrain the decoder's architecture to be the transpose of the encoder one, that is:

$$\mathcal{W}_D = \left\{ W^T_{E_l} \right\}^{1}_{l=L} \tag{13}$$

with T denoting the transpose operator. Then, it is clear that such a constraint does not only reduce the DAE complexity, but also has a major impact on the form of the orthogonality constraints of the minimization problem we are going to define. More specifically, because of the imposed symmetry expressed by Eq. (13), the definition of the Moore–Penrose inverse of a matrix and the definition of the pseudoinverse decoder in Eq. (10), the needed bases for the **null-space** and **range** in each link of the chains, refer to the same weight matrix and this drastically reduces the computational cost for the training of the net. All those issues are presented by the following lemma.

Lemma 2: Let $V_{E_l} = [V_{E_{l_\mathcal{R}}},\ V_{E_{l_\mathcal{N}}}],\ l = 1, 2, \cdots, L$ be the orthonormal bases of $\mathbb{R}^{N_l},\ l = 1, 2, \cdots, L$ resulting from the SVD of matrices W_{E_l},of the encoder, where matrices $V_{E_{l_\mathcal{R}}},\ V_{E_{l_\mathcal{N}}}$ are bases of the **range** and **null-space** of weights matrices W_{E_l} of the encoder and $\bar{\xi}_l = \xi_l - \tilde{\zeta}_l$ with $\xi_l,\ \tilde{\zeta}_l,\ l = 0, 1, \cdots, L$ the sequences of RVs, produced by the encoder and inverse decoder. Then, the following constraints hold:

$$\begin{aligned} V^T_{E_{l_\mathcal{R}}} \mathbb{E}_{\bar{\xi}_{l-1} \sim g_{\bar{\xi}_{l-1}}} \left[\bar{\xi}_{l-1} \right] &= \mathbf{0}_{M_l} \\ V^T_{E_{l_\mathcal{N}}} \mathbb{E}_{\tilde{\zeta}_{l-1} \sim g_{\tilde{\zeta}_{l-1}}} \left[\tilde{\zeta}_{l-1} \right] &= \mathbf{0}_{N_l - M_l},\ l = 1, 2, \cdots, L \end{aligned} \tag{14}$$

with $\mathbb{E}_{\bar{\xi}_{l-1} \sim g_{\bar{\xi}_{l-1}}}[.]$ and $\mathbb{E}_{\tilde{\zeta}_{l-1} \sim g_{\tilde{\zeta}_{l-1}}}[.]$ denoting the expectation operators over the RVs $\bar{\xi}_{l-1}$ and $\tilde{\zeta}_{l-1}$ in the $l-$th layer of the encoder. Note that in each layer of the AE this ensures, on average, the mapping of the transformed noise into the **noise subspace** and of the transformed reconstructed images into the **signal subspace**. In particular, note that for $l = 1$, on average, the mapping of the input noise into the **noise subspace** and the reconstructed image into the **signal subspace** is ensured.

□

By employing the MSE of the reconstruction as our cost function and using **Lemma 2**, we define the following constrained optimization problem:

$$\begin{aligned} \min_{\mathcal{P}_E} \quad & \mathbb{E}_{\mathcal{X}\sim g_{\mathcal{X}}}\left[||\mathcal{X}-\mathcal{X}_r||_2^2\right] \\ s.t. \quad & \sum_{l=1}^{L} ||V^T_{E_{l_{\mathcal{R}}}} \mathbb{E}_{\bar{\xi}_{l-1}\sim g_{\bar{\xi}_{l-1}}}\left[\bar{\xi}_{l-1}\right]||_2^2 = 0 \quad \text{and} \\ & \sum_{l=1}^{L} ||V^T_{E_{l_{\mathcal{N}}}} \mathbb{E}_{\tilde{\zeta}_{l-1}\sim g_{\tilde{\zeta}_{l-1}}}\left[\tilde{\zeta}_{l-1}\right]||_2^2 = 0 \end{aligned} \tag{15}$$

for solving the problem at hand, with the equality constraints imposing the desired orthogonality.
Note that in the optimization problem, only the encoder's parameter set $\mathcal{P}_E$ is needed, thus reducing drastically the computational burden for solving the problem.

It is important to stress at this point that in order to make the solution of the problem easier the $\sum_{l=1}^{L} N_l$ constraints defined in **Lemma 2** have been replaced by just two. Since the optimization problem (Eq. (15)) is a constrained optimization problem with equality constraints, we can define the following Lagrangian function:

$$\begin{aligned} \mathcal{L}(\mathcal{P}_E,\, \lambda_E,\, \lambda_D) \quad = \quad & \mathbb{E}_{\mathcal{X}\sim g_{\mathcal{X}}}\left[||\mathcal{X}-\mathcal{X}_r||_2^2\right] \\ + \quad & \lambda_E \sum_{l=1}^{L} ||V^T_{E_{l_{\mathcal{R}}}} \mathbb{E}_{\bar{\xi}_{l-1}\sim g_{\bar{\xi}_{l-1}}}\left[\bar{\xi}_{l-1}\right]||_2^2 \\ + \quad & \lambda_D \sum_{l=1}^{L} ||V^T_{E_{l_{\mathcal{N}}}} \mathbb{E}_{\tilde{\zeta}_{l-1}\sim g_{\tilde{\zeta}_{l-1}}}\left[\tilde{\zeta}_{l-1}\right]||_2^2 \end{aligned} \tag{16}$$

with $\lambda_E,\ \lambda_D$ denoting the Lagrange multipliers, and solve the above defined unconstrained optimization problem by finding the stationary points of the Lagrangian function defined in Eq. (16) [3], [5], [20].

Data-driven approach

It is clear that in order to solve the above defined optimization problem the pdfs $g_{\bar{\xi}_{l-1}}(.)$ and $g_{\tilde{\zeta}_{l-1}}(.)$, $l = 1, 2, \cdots, L$ must be known. However, in a data-driven version of the problem two collections of training data $\{\mathbf{x}_{w_i}\}_{i=1}^{K}$ and $\{\mathbf{x}_i\}_{i=1}^{K}$ are given instead, each one containing K samples of RV $\mathcal{X}_w$ and $\mathcal{X}$ respectively, and the expected values are estimated using the large numbers' law from those sets.

7. Experiments

In this section we present our experimental results when dealing with the problem of defending against the adversaries produced from the AdvGAN [30], FGSM [8], R-FGSM [25], C&W [4], PGD [14], BIM [11]. MI-FGSM [6] and DeepFool [16] frameworks and we compare our results with the results obtained by the state-of-the-art in [1] & [24] and the techniques contained therein on the MNIST and Fashion-MNIST datasets. Additional results for the problem of denoising, as well as the experimental setup, architectures of the classifier models and the DAEs are described in the supplementary material for the sake of space.

7.1. Evaluating the DAEs against adversarial attacks on the MNIST dataset

In this setup we used our attack-agnostic trained DAEs to evaluate the capabilities of the proposed training, in the task of defending against various adversarial attacks, namely FGSM [8], R-FGSM [25] and PGD [14] frameworks. To validate our proposed defences we have firstly compared their performances in terms of the classification accuracy against the DDSA [1] framework. This framework follows similar concepts in terms of architecture and defence strategy to ours, thus justifying the comparison with what could be considered an older work in the literature, by highlighting that our network can achieve better results with less computational cost and network size in terms of parameters, with the performance of the proposed tied-weights DAE being the most prominent. As we can see from the contents of Table 1, the proposed DAEs outperform DDSA for most of the attacks while simultaneously utilizing less than half the parameters. In the case of the FGSM we can see that our network falls behind DDSA, but still performs adequate in this attack, without losing the ability to generalize well over different attacks and in different scenarios, which is the main purpose of a defence module.

Additionally we must stress at this point that the performance of the two DAE architectures, namely full and tied-weights, differs significantly. In the case of the tied-weights DAE we achieve from 1.5% to 3.5% increase in the classification accuracy, even with half the model parameters (and subsequently degrees of freedom). The architecture of the DAEs consists of three fully connected layers in the encoder and the decoder additionally, summing up to six total layers. We used leaky ReLUs as nonlinearities in the encoder with learnable parameters for the slope in the negative part and their inverse counterparts as the nonlinearities in the decoder. In the case of the tied-weights DAE the decoder consists of the transposed parameters of the encoder.

We also tested our conditioned defenders against the famous AdvGAN [30] attack and present our results. In Figure 3, we can see a set of 64 non cherry-picked images from the MNIST dataset, their AdvGAN contaminated counterparts, as well as the results of the denoising when we feed

Table 1. Classification accuracies under various black-box and gray-box attacks of the proposed DAEs on MNIST dataset, compared to [1]. *:Ours

Class of Attack	**black-box**			**gray-box**		
Attack	FGSM	R-FGSM	PGD	FGSM	R-FGSM	PGD
*Full DAE	84.58	88.66	94.08	83.73	88.12	93.43
*Full DAE with Constraints	85.18	88.95	93.97	84.41	88.22	93.97
*Tied-weights DAE	82.93	88.04	93.6	82.12	87.81	93.09
*Tied-weights DAE with Constraints	85.1	**89.59**	**94.49**	83.57	**89.09**	**94.12**
DDSA [1]	**90.2**	88.9	91.1	**89.9**	84.9	88.9

the adversaries in our trained networks. The DAEs work well on reverting the effects of the adversarial attack on the images and this is evident when we see the classification results on the denoised images.

The AdvGAN attack reduced the classifier accuracy to 1.3%. By first denoising the adversaries, we raised the accuracy back to 90.5% in the case of the full DAE and 93.5% in the case of the tied-weights DAE. In addition, when training the DAEs without the proposed constraints, the classification accuracies were 89% for the full weighs and 91.5% for the tied-weights DAE respectively, indicating that the added constraints improve the performance of DAEs against the adversarial attacks. In [30], authors evaluate the AdvGAN framework against various state-of-the-art defences on MNIST, namely standard FGSM adversarial training [8], ensemble adversarial training [25] and iterative training [14], in the context of a black-box attack scenario. It is clear that, in this context, our proposed defence outperforms the proposed techniques (see Table 2).

Table 2. Adversarial Success Rate (ASR) of the proposed DAEs vs state-of-the-art defences. *:Ours

Defences	AdvGAN
Adversarial training [8]	11.5%
Ensemble adversarial training [25]	10.3%
Iterative training [14]	12.2%
*Full DAE with Constraints	9.5 %
*Tied-weights DAE with Constraints	**6.5%**

7.2. Evaluating the DAEs against adversarial attacks on the Fashion-MNIST dataset

Next up, to show how our proposed defence can generalize in more complex scenarios, we tested our defence on the fashion-MNIST dataset [31]. The dataset includes images that are comparable in size to the MNIST dataset, along with a similar number of training and testing images, as well as classes. The main difference is that, this dataset consists of images with richer texture, thus defending against adversarial attacks performed on this dataset poses a greater challenge. In this experiment, we also added two convolutional layers before the encoder and two layers after the decoder. The purpose of this decision is to first extend the reconstruction capabilities of our DAE in order to follow up with the more complex dataset and second to highlight that our technique works not only on pixel values, but also on features produced by the convolutional layers.

Table 3. Classification accuracies and complexities of the two compared classifiers

Classifier	**Accuracy**	**Number of Parameters**
Resnet	93.51%	34k
Ours	90.08%	13.5k

In Figure 4 we can see the effects of our tied-weights defender on the tampered images from the FGSM attack. Additional results comparing the use of full and tied-weights DAEs trained with and without the proposed constraints in different attack scenarios can also be found in the supplementary material, justifying the use of our tied-weights constrained DAE in the following sections.

To compare our work with state-of-the-art defences on this dataset, we selected a few recent publications and ran our experiments on similar setups for fairness of comparison. To our knowledge, ShieldNets [24] shows the best performance up to this day on the fashion-MNIST dataset. The difference between the state-of-the-art results and ours (Table 4) can be attributed to the complexity of the Resnet based classifier showcased by the achieved classification accuracy on the test set, as we can see from Table 3. Additionally, besides the difference in classifier complexity and pristine accuracy, ShieldNets utilize the PixelCNN [26] architecture, which consists of 15 layers ($\sim$ 1M parameters), compared to our tied-weights DAE of 6 layers ($\sim$ 320k parameters).

While our results are lower than the ones presented in

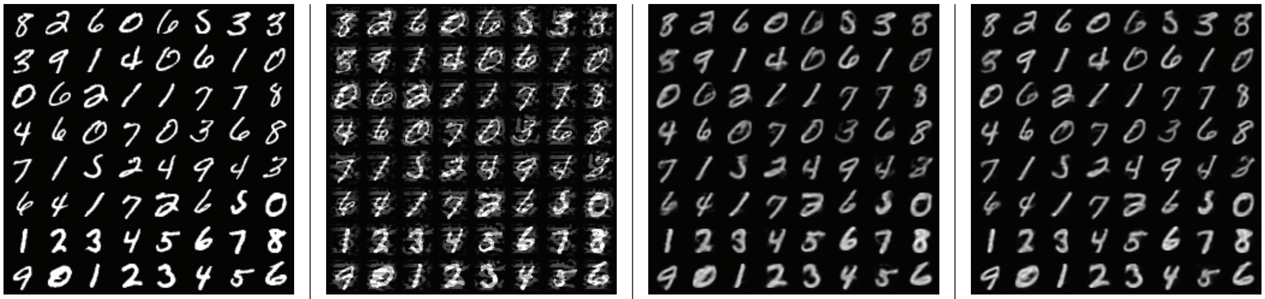
(a) (b) (c) (d)

Figure 3. The performance of the under comparison techniques on an adversarial attacked set (b) of the set of the pristine MNIST images shown in (a). Full parameters (c) and tied-weights DAE (d).

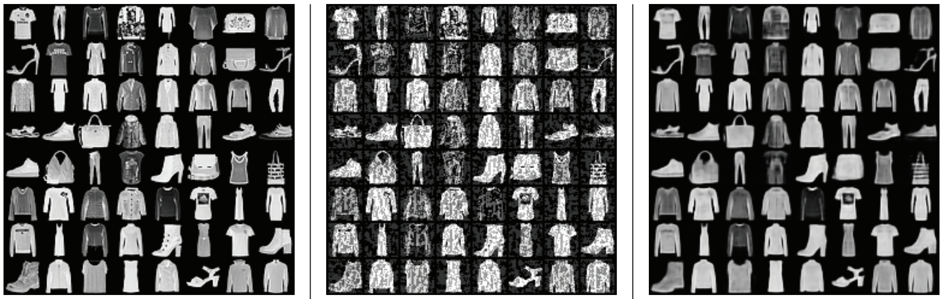
(a) (b) (c)

Figure 4. 64 Non-cherry picked images from fashion-MNIST (a). Original (b) with adversarial noise from FGSM (c) Defended.

Table 4. Comparison of the defences against various attacks on Fashion-MNIST, e=8/25. *:Ours

Network	Training Technique	FGSM	BIM	DeepFool	CW	MI-FGSM
Resnet	Label Smoothing [18]	64.23/36.81	9.76/0.00	22.42/3.37	20.77/4.61	4.25/0.00
	Adversarial FGSM [14]	82.49/78.43	44.34/6.46	57.28/11.92	51.03/15.70	39.72/0.00
	Pixel Defend [22]	85.00/74.00	83.00/76.00	87.00/87.00	87.00/87.00	NA
	ShieldNets [24]	**91.59/89.04**	**91.17/89.74**	**92.62/90.28**	**92.66/90.78**	**90.63/90.47**
*CNN	*Tied-weights DAE with Constraints	87.54/78.85	87.18/82.69	89.75/89.05	89.74/89.52	85.90/85.21

[24] in terms of classification accuracy on the defended results, the main takeaway is that, our DAE can generalise its ability to defend against various attacks and we can also achieve comparable results to more complex classifier and defence architectures with simpler solutions.

8. Conclusion

In this study, we proposed a solution to the image denoising problem using a deep DAE. Our DAE architecture enforces orthogonality between noise and intermediate representations in the encoder's chain and between the reconstructed image and its intermediate representations in the pseudoinverse decoder. Our proposed solution is based on an equality constrained optimization problem that uses Lagrange multipliers. We trained both full-weights and tied-weights DAEs and found that the latter architecture not only gave significant improvements in denoising and defending against strong adversarial attacks on two datasets, but was also much simpler. Both architectures performed well as defenders when trained in an attack-agnostic setup. To further improve the performance of the proposed solution, its effects on more complex classification architectures are investigated. Additionally, we plan to evaluate the effectiveness of the proposed defenders on more challenging datasets to ensure their robustness in real-world scenarios. Finally, we are in the process of finding optimal and memory efficient ways to expand our technique on the more commonly used convolutional networks. Our study highlights the potential of the proposed DAE architecture as a lightweight and promising solution for image denoising and defending against adversarial attacks.

References

[1] Yassine Bakhti, Sid Ahmed Fezza, Wassim Hamidouche, and Olivier Déforges. Ddsa: A defense against adversarial attacks using deep denoising sparse autoencoder. *IEEE Access*, 7:160397–160407, 2019. 2, 6, 7

[2] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012. 1

[3] G. E. Box and D. R. Cox. An analysis of transformations. *journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211—243, 1964. 3, 6

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 1, 2, 6

[5] Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *Algorithmic Learning Theory*, pages 300–332. PMLR, 2019. 6

[6] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1, 6

[7] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246, 2016. 1

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2, 6, 7

[9] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 2

[10] Chih-Hui Ho and Nuno Vasconcelos. Disco: Adversarial defense with local implicit functions. *Advances in Neural Information Processing Systems*, 35:23818–23837, 2022. 2

[11] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. 1, 6

[12] Jincheng Li, Shuhai Zhang, Jiezhang Cao, and Mingkui Tan. Learning defense transformations for counterattacking adversarial examples. *Neural Networks*, 164:177–185, 2023. 2

[13] Tian Yu Liu, Yu Yang, and Baharan Mirzasoleiman. Friendly noise against adversarial noise: a powerful defense against data poisoning attack. *Advances in Neural Information Processing Systems*, 35:11947–11959, 2022. 2

[14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2, 6, 7, 8

[15] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017. 2

[16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1, 6

[17] 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*, ASIA CCS '17, page 506–519, New York, NY, USA, 2017. Association for Computing Machinery. 2

[18] 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, 2016. 2, 8

[19] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 2

[20] Sara Sangalli, Ertunc Erdil, Andeas Hötker, Olivio Donati, and Ender Konukoglu. Constrained optimization to train neural networks on critical and under-represented classes. *Advances in Neural Information Processing Systems*, 34:25400–25411, 2021. 6

[21] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[22] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. 2, 8

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[24] Rajkumar Theagarajan, Ming Chen, Bir Bhanu, and Jing Zhang. Shieldnets: Defending against adversarial attacks using probabilistic adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6988–6996, 2019. 2, 6, 7, 8

[25] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 1, 2, 6, 7

[26] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 7

[27] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. 1

[28] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 1

[29] David Warde-Farley and Ian Goodfellow. 11 adversarial perturbations of deep neural networks. *Perturbations, Optimization, and Statistics*, 311:5, 2016. 2

[30] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 2, 6, 7

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 7

[32] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 2