Improving Gradient Regularization using Complex-Valued Neural Networks

Eric Yeats [1] **Yiran Chen** [1] **Hai Li** [1]

Abstract

Gradient regularization is a neural network defense technique that requires no prior knowledge of an adversarial attack and that brings only limited increase in training computational complexity. A form of complex-valued neural network (CVNN) is proposed to improve the performance of gradient regularization on classification tasks of real-valued input in adversarial settings. The activation derivatives of each layer of the CVNN are dependent on the combination of inputs to the layer, and locally stable representations can be learned for inputs the network is trained on. Furthermore, the properties of the CVNN parameter derivatives resist decrease of performance on the standard objective that is caused by competition with the gradient regularization objective. Experimental results show that the performance of gradient regularized CVNN surpasses that of real-valued neural networks with comparable storage and computational complexity. Moreover, gradient regularized complex-valued networks exhibit robust performance approaching that of real-valued networks trained with multi-step adversarial training.

1. Introduction

Recent deep learning (DL) models can outperform humans on image classification tasks (He et al., 2016). While the exceptional performance of DL on tightly controlled classification tasks is a remarkable achievement, we would like for DL's high performance to be robust to all forms of noise that a deployed DL model might encounter. Adversarial examples (Szegedy et al., 2013) provide evidence that this is not the case for standard DL models. Szegedy et al. (2013) showed that neural networks can be fooled reliably by adding adversarially crafted noise to clean DL inputs, and that these inputs are difficult to distinguish from normal inputs. Subsequent investigation by Goodfellow et al. (2014) and Papernot et al. (2016) further demonstrate that adversarial examples generalize across DL models trained on disjoint subsets of the dataset, and transfer well across different machine learning algorithms. Computationally cheap yet effective gradient-based "White-box" attacks (attacks which have knowledge of the network parameters) such as FGSM (Goodfellow et al., 2014), I-FGSM (Kurakin & Bengio, 2017), PGD (Madry et al., 2017), and MIM (Dong et al., 2018) have since been developed. Carlini & Wagner (2017) develop an attack that is extremely difficult to detect by significantly improving the attack optimization process.

For situations in which the DL model parameters are not known to the attacker ("Black-box"), perturbed inputs are often crafted using surrogate networks and later provided to the black-box model. Several works have observed that black-box attacks transfer more effectively if perturbations are designed to alter neuron activations in the feature extraction layers (Inkawhich et al., 2019; Zhou et al., 2018). The existence of adversarial examples and their high transferability questions the fitness of current DL models for security-sensitive applications (Eykholt et al., 2018; Kurakin & Bengio, 2017).

Methods of providing security to neural networks tend to not be as effective as the methods to attack them. One of the most well known and successful security methods is adversarial training (Goodfellow et al., 2014), which incorporates single-step adversarial example generation into the standard training procedure. Subsequent works developed more powerful multi-step attacks to circumvent adversarial training (Kurakin et al., 2016). Madry et al. (2017) reformulated adversarial training as a multi-step dual optimization problem to make it resistant to more powerful white-box attack, at the cost of greatly increasing the computational complexity of the training procedure. Although adversarial training is the best empirical defense to date, it requires prior knowledge of an attack to defend against. This leaves the possibility for an adversary to develop a new attack to circumvent the network's security.

Other security methods rely on promoting some measure

[1]ECE Dept., Duke University, Durham, North Carolina, USA. Correspondence to: Eric Yeats <eric.yeats@duke.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

of information diversity between ensembled networks and taking a group vote during inference (Yang et al., 2020; Pang et al., 2019) to reduce the transferability of adversarial examples generated by a surrogate network. While ensembling methods tend to successfully reduce attack transferability, they typically do not confer significant resistance to white-box attacks. The success of ensemble-based security methods often relies on the number of networks which compose the ensemble, which significantly impacts the computational and storage requirements for both training and inference.

A common theme among these security techniques is that they add much computational complexity to the standard training procedure, making secure networks unobtainable and difficult to deploy without access to significant computing resources (Kurakin et al., 2016). It is therefore necessary to develop new methods of producing robust networks that are less computationally expensive to train and implement.

Gradient regularization is a promising defense method that meets these criteria. Introduced by "Double Backpropagation" by Drucker & Le Cun (1992), gradient regularization is implemented by maintaining the gradient computational graph and minimizing the input gradient with respect to model parameters. It has been shown to improve model generalization, interpretability, and adversarial robustness (Ross & Doshi-Velez, 2018; Lyu et al., 2015; Finlay & Oberman, 2021). Furthermore, the gradient regularization objective can be approximated with just two passes of regular backpropagation, making it scalable to very large networks (Finlay & Oberman, 2021). While this general method improves adversarial resistance without prior knowledge of an attack, the need for an inner optimization loop, or an ensemble, in practice, the method has not performed as well as multi-step adversarial training.

One reason for this is that networks require strong gradient regularization to have *local stability* on inputs the network is trained on, but the strong regularizer leads to decrease in performance on the standard objective (Finlay & Oberman, 2021; Ross & Doshi-Velez, 2018). This work introduces a complex-valued neural network (CVNN) to resist decrease of performance on the standard objective that is due to competition with the gradient regularized objective. We hereafter refer to any neural network with at least one complex-valued layer as a complex-valued neural network, or CVNN. The CVNN exhibits improved gradient regularized performance over real-valued networks, while using comparable parameter and multiply-and-accumulate (MAC) operation counts. Code is available at: `https://github.com/ericyeats/cvnn-security`.

The contributions of the paper are summarized below:

- We introduce a new form of complex-valued convolutional neural networks for real-valued image classification applications that has better robust performance than real-valued networks with similar parameter and and MAC counts.
- We analyze the activation derivatives and Jacobian derivatives of each layer with respect to model parameters for real-valued and complex-valued networks, and demonstrate why complex-valued networks trained with gradient regularization perform better than real-valued networks.
- We provide empirical evidence that complex-valued networks trained with gradient regularization are more capable of satisfying both the standard and gradient regularized objectives.
- We evaluate the robustness of real-valued and complex-valued networks trained with gradient regularization on popular image classification benchmarks and compare the result with that of state-of-the-art adversarial training.

2. Background

Complex-Valued Networks

Complex-valued networks for real-valued classification problems have been explored in a small collection of previous works. Amin & Murase (2009) find that single-layer complex-valued networks can solve linearly inseparable problems and can converge faster than real-valued networks when trained with gradient descent. Amin & Murase (2009) introduce an input transformation from real to complex numbers, and they develop a set of complex-valued activation functions with real-valued outputs. They consider just single-layer complex-valued networks and compare their training and performance with that of real-valued networks on toy classification tasks.

Forms of convolutional complex-valued networks for real-valued classification tasks have been proposed. Worrall et al. (2017) exploit the properties of complex numbers to provide rotation-invariant classification of digits. Trabelsi et al. (2018) introduce a formulation of complex-valued batch normalization and weight initialization for deep complex networks, and they evaluate the performance of different nonlinear activation functions for complex-valued networks on computer vision tasks, reaching comparable performance to standard (real) deep networks with the same number of parameters. The convolutional networks described in Trabelsi et al. (2018) use a fully complex representation whereas our convolutional network described (in Section 2) employs a hybrid real- and complex-valued representation.

Gradient Regularization

To demonstrate how gradient regularization interacts with the standard objective, consider the forward pass of an n-layer neural network f and input vector $\underline{x}$ represented as a composition of functions, where each function output represents a linear combination of inputs followed by a nonlinear activation function,

$$f(\underline{x}) = f_n(f_{n-1}(...f_2(f_1(\underline{x})))). \tag{1}$$

Some scalar-valued loss function $\mathcal{L}(f, \underline{x}, \underline{y})$ is defined on $f(\underline{x})$ and desired output $\underline{y}$. The network is trained using stochastic gradient descent by computing the gradient of $\mathcal{L}(f, \underline{x}, \underline{y})$ with respect to the parameters of f. Backpropagation of the loss (Rumelhart et al., 1985) provides an input gradient

$$\frac{\partial L(f, \underline{x}, \underline{y})}{\partial \underline{x}} = \frac{\partial f_1(\underline{x}_1)}{\partial \underline{x}_1}\Big(\prod_{i=1}^{n-1} \frac{\partial f_{i+1}(\underline{x}_{i+1})}{\partial \underline{x}_{i+1}}\Big)\frac{\partial L(f, \underline{x}, \underline{y})}{\partial \underline{x}_n}, \tag{2}$$

where $\underline{x}_1 = \underline{x}$, $\underline{x}_{i+1} = f_i(\underline{x}_i)$, and the output logit vector of the network is $\underline{x}_n$. In equation (2), the rightmost term represents a vector of loss gradient with respect to the outputs of the final layer, and each function derivative represents a transposed Jacobian matrix. Gradient regularization adds a squared norm penalty to the input gradient

$$\mathcal{R}(f, \underline{x}, \underline{y}) = \beta \left\| \frac{\partial L(f, \underline{x}, \underline{y})}{\partial \underline{x}} \right\|_p^2, \tag{3}$$

where β is a hyperparameter controlling the strength of regularization and $||\cdot||_p$ is some p-norm. Substituting (2) and differentiating (3), a second pass of backpropagation, called double backpropagation (Drucker & Le Cun, 1992), computes the gradient of the norm of the input gradient with respect to the parameters of f. Hence, the derivative of each Jacobian $\frac{\partial f_i(\underline{x}_i)}{\partial \underline{x}_i}$ from (2) with respect to the model parameters f is used to minimize (3). For a standard real-valued network, the Jacobian is the weight matrix W of each layer, and the element-wise derivative of the Jacobian with respect to model parameter W is simply a matrix of ones. Therefore, the gradient for a weight parameter W_i can be expressed as

$$\begin{aligned}\nabla_{W_i}\Big[\mathcal{L}(f, \underline{x}, \underline{y}) + \beta\mathcal{R}(f, \underline{x}, \underline{y})\Big] \\ = \underline{e}_{i\mathcal{L}}\underline{1}^T \cdot \frac{\partial(W_i\underline{x}_i)}{\partial W_i} + \beta\underline{e}_{i\mathcal{L}}\underline{e}_{i\mathcal{R}}^T \cdot \frac{\partial \frac{\partial(W_i\underline{x}_i)}{\partial \underline{x}_i}}{\partial W_i} \\ = \underline{e}_{i\mathcal{L}}(\underline{x}_i + \beta\underline{e}_{i\mathcal{R}})^T,\end{aligned} \tag{4}$$

where $e_{i\mathcal{L}}$ and $e_{i\mathcal{R}}$ are the backpropagated standard loss error and double-backpropagated gradient regularization loss error vectors and x_i is the input vector at layer i, respectively. Note that prior to reaching W_i, backpropagated $e_{i\mathcal{L}}$ is modified element-wise by the derivative of the nonlinear activation function. Since the gradient regularization objective does not necessarily share a relationship with the input to a given layer, gradient regularization will compete with the standard training objective, decreasing the network's performance on the standard training objective.

(a) Complex-Valued Feat Response

(b) Sum of Two Complex-Valued Feature Responses

Figure 1. Two example complex-valued feature responses defined over bounded two-dimensional input. The magnitude of each complex-valued weights in these examples are the same.

Proposed Complex-Valued Network

Contrary to the standard neural network layer, we show that the numerical properties of the CVNN protect the network from competition between $\mathcal{L}(f, \underline{x}, \underline{y})$ and $\mathcal{R}(f, \underline{x}, \underline{y})$. Consider the forward pass of a complex-valued layer g_i for element-wise transformed complex-valued inputs $\underline{x}_R = \cos(\underline{x}_i)$ and $\underline{x}_I = \sin(\underline{x}_i)$:

$$\begin{aligned}g_i(\underline{x}_i) = \Big[(W_R\underline{x}_R - W_I\underline{x}_I + \underline{b}_R)^2 + \\ (W_R\underline{x}_I + W_I\underline{x}_R + \underline{b}_I)^2\Big]^{\frac{1}{2}} = \sqrt{\underline{z}_R^2 + \underline{z}_I^2},\end{aligned} \tag{5}$$

where $\underline{x}_R, \underline{x}_I, W_R, W_I, \underline{b}_R, \underline{b}_I, \underline{z}_R, \underline{z}_I$ are the real and imaginary part inputs, weights, biases, and activations, respectively. The magnitude of the complex activation $\underline{z}_R + j\underline{z}_I$ is taken as the real-valued result $g_i(\underline{x}_i)$.

Figure 1 depicts an example $g_i(\underline{x}_i)$, defined over bounded two-dimensional input $\underline{x}_i$. One complex-valued feature results in a distinct maximum and minimum, where the derivatives for the bounded input are near zero. Summing two or more complex-valued features can result in several maxima and minima, which can promote feature response stability to natural variations in an input distribution. One can show that the derivatives of $g_i(\underline{x}_i)$ with respect to the weight parameters are

$$\frac{\partial g_i(\underline{x}_i)}{\partial W_R} = \frac{\underline{z}_R\underline{x}_R^T + \underline{z}_I\underline{x}_I^T}{g_i(\underline{x}_i)},$$

$$\frac{\partial g_i(\underline{x}_i)}{\partial W_I} = \frac{\underline{z}_I \underline{x}_R^T - \underline{z}_R \underline{x}_I^T}{g_i(\underline{x}_i)},$$

and a constraint between the activation derivatives,

$$(\frac{\partial g_i(\underline{x}_i)}{\partial W_R})^2 + (\frac{\partial g_i(\underline{x}_i)}{\partial W_I})^2 = 1. \tag{6}$$

The constraint (6) implies that when one gradient term is large, the other must be minimal. Further, the Jacobian can be expressed as

$$\frac{\partial g_i(\underline{x}_i)}{\partial \underline{x}_i} = \frac{W_R^T \underline{z}_I - W_I^T \underline{z}_R}{g_i(\underline{x}_i)} \underline{x}_R - \frac{W_R^T \underline{z}_R + W_I^T \underline{z}_I}{g_i(\underline{x}_i)} \underline{x}_I,$$

and that the derivatives of the Jacobian with respect to the model parameters are simply

$$\frac{\partial \frac{\partial g_i(\underline{x}_i)}{\partial \underline{x}_i}}{\partial W_R} = \frac{\partial g_i(\underline{x}_i)}{\partial W_I} \tag{7}$$

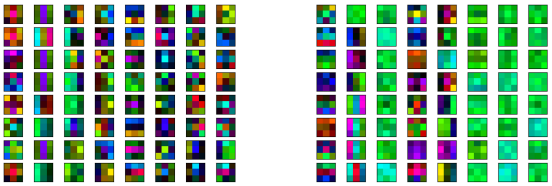
$$\frac{\partial \frac{\partial g_i(\underline{x}_i)}{\partial \underline{x}_i}}{\partial W_I} = -\frac{\partial g_i(\underline{x}_i)}{\partial W_R} \tag{8}$$

Considering (7) and (8), we show that the derivatives of the Jacobian with respect to model parameters also follow the constraint. This relationship provides a degree of mutual exclusivity between the parameter updates for the competing loss objectives. In other words, for a given parameter, when the magnitude of the activation derivative (used to minimize standard loss) is large, the magnitude of the Jacobian derivative (used to minimize gradient regularization loss) must be small, and vice-versa.

$$\nabla_{W_{iR}} \left[\mathcal{L}(f, \underline{x}, \underline{y}) + \beta \mathcal{R}(f, \underline{x}, \underline{y}) \right] = \underline{e}_{i\mathcal{L}} \underline{1}^T \cdot \frac{\partial g_i(\underline{x}_i)}{\partial W_{iR}} + \beta \underline{e}_{i\mathcal{L}} \underline{e}_{i\mathcal{R}}^T \cdot \frac{\partial g_i(\underline{x}_i)}{\partial W_{iI}} \tag{9}$$

$$\nabla_{W_{iI}} \left[\mathcal{L}(f, \underline{x}, \underline{y}) + \beta \mathcal{R}(f, \underline{x}, \underline{y}) \right] = \underline{e}_{i\mathcal{L}} \underline{1}^T \cdot \frac{\partial g_i(\underline{x}_i)}{\partial W_{iI}} - \beta \underline{e}_{i\mathcal{L}} \underline{e}_{i\mathcal{R}}^T \cdot \frac{\partial g_i(\underline{x}_i)}{\partial W_{iR}} \tag{10}$$

Equations (9) and (10) summarize the gradient of the real and imaginary part weights for the combined objective. The combined objective gradients for the complex-valued weights contrast with that of real-valued weights due to the constraint on the activation and Jacobian derivatives. For each complex-valued layer, the constraint (6) provides separation between the standard and gradient regularized objectives. In other words, the constraint implies that when changing a parameter is important for one objective, that same change is unimportant for the other objective. We hypothesize that this gradient constraint is the core reason why complex-valued networks have superior gradient regularized training characteristics than that of real-valued networks. We provide empirical evidence supporting the hypothesis in the following section.



(a) $\mathcal{N}(\mu = 0, \sigma = 0.05)$ (b) $\mathcal{N}(\mu = 0, \sigma = 0)$

Figure 2. Conv1 Filters for gradient regularized complex-valued networks trained with (a) and without (b) a small amount of additive Gaussian noise. Both networks are trained on the Fashion-MNIST training set with $\beta = 64$, and the activations of all filters along a column are summed together, for an output channel count of 8. Hue and value represent the relative angle and magnitude of each weight, respectively.

3. Evaluation

Experiment Setup

All experiments are conducted using the PyTorch library (Paszke et al., 2019). We evaluate the training characteristics, white-box adversarial robustness, and black-box robustness of gradient regularized CVNN. We compare these results with those of real-valued NN trained with gradient regularization or adversarial training. We consider the additional parameter and MAC requirement of complex numbers over real numbers. In general, we consider each complex-valued parameter to be the same size as two real-valued parameters, and we consider each complex-complex MAC to be equivalent to 4 real-real MAC operations. Using the same basic convolutional neural network architecture as the complex-valued networks, we increase output filter count and fully-connected layer width of each real-valued network such that its parameter and MAC count is similar to or surpasses that of the complex-valued network. We tried a variety of convolution and fully-connected configurations for real-valued networks with similar baseline architecture, and in all experiments, real-valued networks with higher parameter and MAC count outperformed the smaller real-valued networks.

White-box attacks are crafted against the networks on four popular image classification benchmark tasks: MNIST (LeCun, 1998), FashionMNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR-10 (Krizhevsky et al., 2009). The MNIST, FashionMNIST, and CIFAR-10 benchmarks consist of 50000 training images and 10000 test images. The SVHN benchmark consists of 73,257 training images and 26,032 test images. The networks are trained on the standard and gradient regularization objectives using the training set, and the networks are evaluated with adversarial examples crafted from the test set.

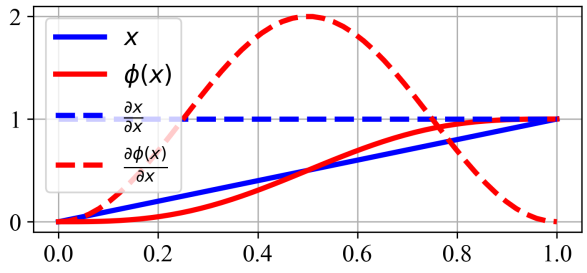


Figure 3. Comparison of identity (blue) mapping with the sinusoidal contrast enhancement (red) for complex-valued networks for CIFAR-10.

Table 1. Model Parameter and Total MAC Count

NETWORK	TOTAL PARAMETERS	TOTAL MACS
REAL NETS		
MNIST	421,642	4,241,152
SVHN	421,642	4,241,152
FASHIONMNIST	421,642	4,241,152
CIFAR_1	620,810	10,848,768
CIFAR_2	1,423,114	40,569,344
COMPLEX NETS		
MNIST	208,718	2,664,448
SVHN	208,718	2,664,448
FASHIONMNIST	107,114	3,612,672
CIFAR_1	1,241,662	41,814,528

The standard objective for all networks is cross-entropy loss, optimized using SGD with Nesterov momentum of $\mu = 0.875$ and weight decay of $l_2 = 10^{-4}$. Networks for MNIST, SVHN, and FashionMNIST are trained for 30 epochs with a minibatch size of 64 and an initial learning rate of $\nu = 0.005$, and networks for CIFAR-10 are trained for 80 epochs with a minibatch size of 128 and an initial learning rate of $\nu = 0.01$. Learning rate is decayed by $\gamma = 0.2$ after 20 epochs for MNIST, SVHN, and FashionMNIST networks, and epochs 40, 60, and 72 for CIFAR-10 networks. We apply no data augmentation to networks trained on MNIST. For SVHN, we take a centered 28x28 crop of the luminance band of each image and apply global contrast normalization onto the range $[0, 1]$. For the FashionMNIST training set, we apply a random horizontal flip and a random crop to 28x28 with a padding of 1. When training complex-valued networks with gradient regularization on the FashionMNIST dataset, we noticed that the conv1 filters would overfit the solid black backgrounds of the images, causing gradient descent to become unstable. Adding a small amount of Gaussian noise $\mathcal{N}(\mu = 0, \sigma = 0.05)$ to each training image increased training stability. Figure 2 depicts the learned conv1 filters with and without the additive noise. When using real-valued networks on CIFAR-10, we normalize each input image using the mean and standard deviation of the training set. For complex-valued networks, we perform no normalization. Instead, we apply an element-wise nonlinear mapping function $\phi(x) = x - \frac{\sin(2\pi x)}{2\pi}$ during the forward pass of each input, depicted in Figure 3. Adversarial examples for both real- and complex-valued networks are crafted using the original test set images.

We employ squared $L1$ norm of the input gradient as the gradient regularization objective, as suggested by Finlay & Oberman (2021). Hyperparameter β controls the strength of regularization. Real-valued image inputs are converted to complex-valued inputs via the method described by Amin & Murase (2009), where each real-valued element $x \in [0, 1]$ is linearly transformed to the phase of a complex number z such that $\angle z \in [0, \pi]$ and $|z| = 1$. Real-valued weights are initialized via He et al. (2015) and complex-valued weights are initialized with magnitude $\frac{1}{\sqrt{fanin}}$ and random angle uniformly distributed on $[-\pi, \pi]$.

Table 1 lists the equivalent storage and MAC requirement of each network, considering the additional storage and computational requirement of implementing complex numbers. The number of convolutions and size of the fully-connected layers were typically less for the complex-valued networks in order to make parameter and MAC count more fair. Additional experiments using identical architecture are presented at the end of the evaluation section.

($\mathbb{R}$) denotes a real-valued layer, and ($\mathbb{C}$) denotes a complex-valued layer. ReLU activation function is used as the nonlinearity for real-valued layers (when appropriate) and no activation function (identity mapping) is used for complex-valued layers. 3x3C16 denotes a convolutional layer with kernel size 3x3 and output channel count 16, 16FC denotes a fully-connected layer with 16 outputs, 16BN denotes a batch norm layer (Ioffe & Szegedy, 2015) for 16 output channels, and 2x2AP denotes an average pooling layer with receptive field 2x2. The same network architecture is used for real-valued networks for the MNIST, SVHN, and FashionMNIST benchmarks: 3x3C32($\mathbb{R}$), 2x2AP($\mathbb{R}$), 3x3C64($\mathbb{R}$), 2x2AP($\mathbb{R}$), 128FC($\mathbb{R}$), 10FC($\mathbb{R}$). Complex-valued networks for MNIST and SVHN have architecture: 3x3C16($\mathbb{C}$), 2x2AP($\mathbb{R}$), 3x3C32($\mathbb{C}$), 2x2AP($\mathbb{R}$), 128FC($\mathbb{C}$), 10FC($\mathbb{C}$). In the convolutions of this network, the response of every two complex-valued output channel is summed together, cutting the dimensionality of each output channel in half and increasing filter response stability. Complex-valued networks for FashionMNIST have architecture 3x3C64($\mathbb{C}$), 2x2AP($\mathbb{R}$), 3x3C32($\mathbb{C}$), 2x2AP($\mathbb{R}$), 64FC($\mathbb{R}$), and 10FC($\mathbb{R}$). Every 8 output channels in the first complex-valued convolution channel are summed together, cutting output channel dimensionality down to 8. The CIFAR_1 architecture con-

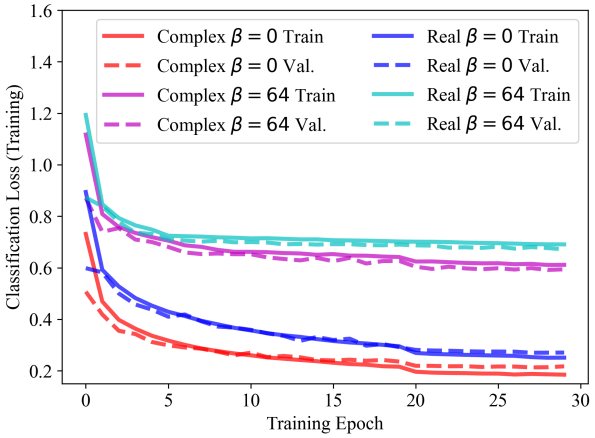


Figure 4. Standard Objective Training and Validation Loss on the FashionMNIST Dataset

sists of 3x3C32, 32BN($\mathbb{R}$), 2x2AP($\mathbb{R}$), 3x3C64, 64BN($\mathbb{R}$), 2x2AP($\mathbb{R}$), 3x3C128, 128BN($\mathbb{R}$), 2x2AP($\mathbb{R}$), 256FC, 10FC. The CIFAR_2 architecture is the CIFAR_1 architecture, but with all output channels of convolutions and batch norms doubled. Real-valued networks have real-valued convolutions and fully-connected layers, whereas complex-valued networks have complex-valued convolutions and fully-connected layers.

Training Behavior with Gradient Regularization

Figure 4 depicts the training and validation losses of complex-valued and real-valued networks with and without strong gradient regularization for the FashionMNIST dataset. Despite having far more parameters and MACs, the real-valued network cannot reach the same clean performance as the complex-valued network. When trained with the same level of strong gradient regularization, the complex-valued network retains higher classification performance on the training and validation sets. We define a metric, parameter update similarity, to measure a neural network's resistance to erosion of standard objective performance. For a network with parameters f, parameter update similarity (ζ) is defined as the cosine similarity between the vectors of parameter gradients for the standard and combined objectives (Equation 11).

$$\zeta = \frac{\nabla_f \mathcal{L}(f, \underline{x}, \underline{y}) \nabla_f \left[\mathcal{L}(f, \underline{x}, \underline{y}) + \beta \mathcal{R}(f, \underline{x}, \underline{y})\right]^T}{\left\|\nabla_f \mathcal{L}(f, \underline{x}, \underline{y})\right\|_2 \left\|\nabla_f \left[\mathcal{L}(f, \underline{x}, \underline{y}) + \beta \mathcal{R}(f, \underline{x}, \underline{y})\right]\right\|_2} \tag{11}$$

Intuitively, a parameter update similarity closer to one indicates that a neural network is learning parameters that are locally optimal for the standard objective. Parameter update similarity is recorded for each minibatch and the average for each epoch is plotted against the average training loss of each epoch in Figure 5. When no gradient regularization is applied, both the complex-valued and real-valued neural networks have parameter update similarities of 1. When the same level of gradient regularization is applied, the complex-valued networks tend to exhibit lower parameter update similarity, indicating that their gradient descent takes a path more biased towards the gradient regularized objective. However, the training loss value of the complex-valued networks is strictly lower than that of the real-valued networks. These results suggest that the complex-valued networks are more capable of satisfying both the standard and gradient regularized objectives. As mentioned previously, we hypothesize that this observation is due to the constraint in equation (6) modulating the combined objective parameter update summarized in equations (9) and (10).

White-Box Attacks

Each network was subjected to a White-Box PGD(8) adversarial examples (Madry et al., 2017) with random initial jump and with varying L_∞ bound. The accuracy of the networks at each ϵ bound is recorded and shown. At $\epsilon = 0$, the images are equal to the clean test set. Undefended networks without gradient regularization or adversarial training tend to have the highest accuracy on these clean examples. However, as ϵ is increased, the performance of undefended networks erodes swiftly.

Figure 6 depicts the White-Box attack results on MNIST. Both real-valued and complex-valued networks were able to obtain 99% accuracy on the clean MNIST test set. As β is increased, all networks tend to perform better against PGD(8) examples with larger L_∞ perturbation bound. However, as β is increased to 512, all complex-valued networks are capable of maintaining 99% clean accuracy, whereas the clean accuracy of the real-valued network is 78% and 62%

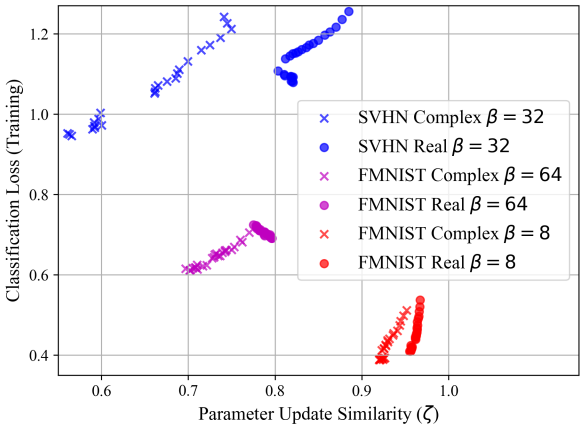


Figure 5. Standard Objective (Classification) Loss vs. Parameter Update Similarity (ζ) of each epoch for the last 25 epochs of training on the FashionMNIST and SVHN training sets. We abbreviate FashionMNIST as FMNIST.

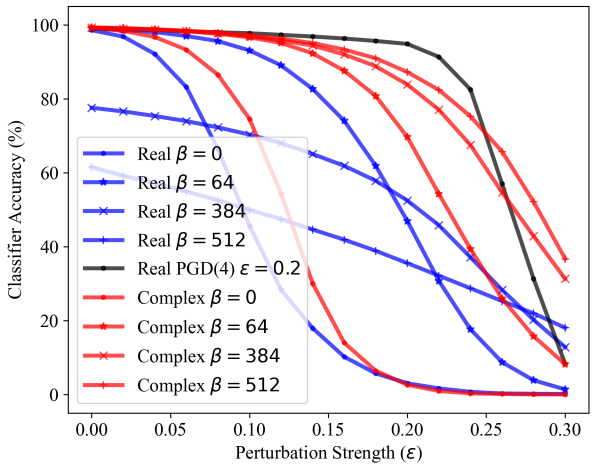


Figure 6. White-Box PGD(8) Attack of varying L_∞ bound (ϵ) on the MNIST test set.

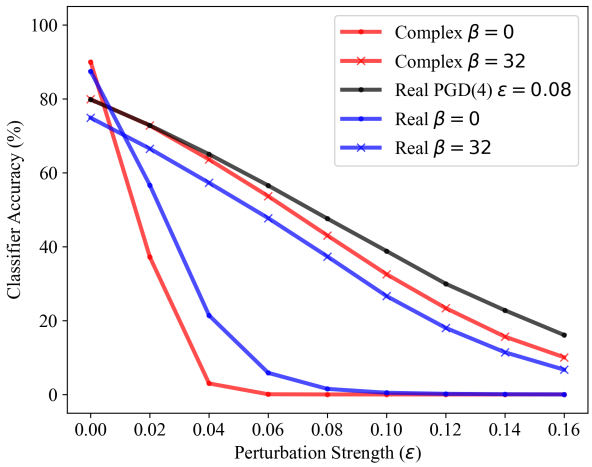


Figure 8. White-Box PGD(8) Attack of varying L_∞ bound (ϵ) on the SVHN test set.

for $\beta = 384$ and $\beta = 512$, respectively. The $\beta = 384$ and $\beta = 512$ real-valued networks have the highest accuracy of all gradient regularized real-valued networks when the L_∞ bound is large, however their clean accuracy is not acceptable. The $\beta = 512$ complex-valued network attains a robust accuracy comparable to that of the PGD(4)-trained real-valued network, without having been trained on any adversarial examples.

Similar trends are observed with the FashionMNIST dataset, depicted in Figure 7. As β is increased, all networks improve in robustness. Increased β also decreases clean accuracy, and we find that both complex- and real-valued networks attain a clean accuracy of 79% when $\beta = 64$. Given the same clean accuracy, the complex-valued network

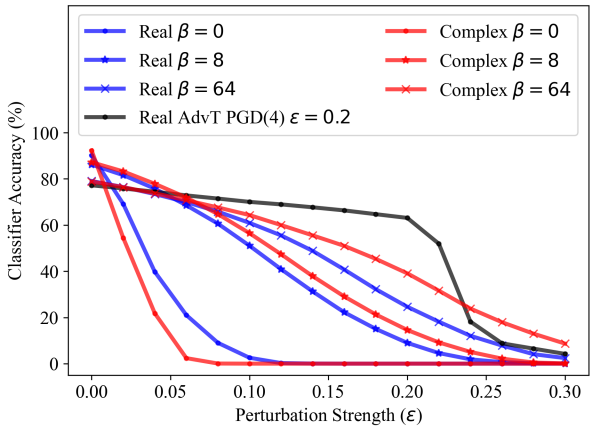


Figure 7. White-Box PGD(8) Attack of varying L_∞ bound (ϵ) on the FashionMNIST test set.

is more robust to adversarial examples than the real-valued network, maintaining an accuracy of 50% when $\epsilon = 0.16$, compared to the real-valued network's 41%. There is a steep drop in performance for the adversarially trained real-valued network when ϵ exceeds the L_∞ bound used in training. We observe that complex-valued networks come closer to the performance of adversarial training.

Likewise, in Figure 8, complex-valued networks close the performance gap between gradient regularization and adversarial training on SVHN. Complex-valued networks trained with gradient regularization achieve the same clean accuracy as the adversarially trained real-valued network, 80%. The complex-valued network retains more adversarial accuracy than the larger real-valued network trained with gradient regularization, approaching the performance of the real-valued network when it has been adversarially trained.

Complex-valued networks trained on the CIFAR-10 datset (Figure 9) did not reach the same clean accuracy as the real-valued networks did. We observed that the complex-valued networks overfit the training set and did not generalize as well. When a white-box attack is conducted, undefended network accuracy approaches zero when the L_∞ bound of the attack approaches $\frac{6}{255}$. Gradient regularized and adversarially trained networks are more robust to attack, and the performance margin between the two security methods is not as significant as the other datasets. While the larger real-valued network (CIFAR_2) attains highest accuracy, its gradient regularized version is less robust to attack. The smaller real-valued network (CIFAR_1) provides a lower bound on the performance of the gradient regularized networks. Although the clean accuracy of the complex-valued network is not highest, it is more robust to white box adversarial examples, approaching the robust performance of the large, adversarially trained real-valued network.

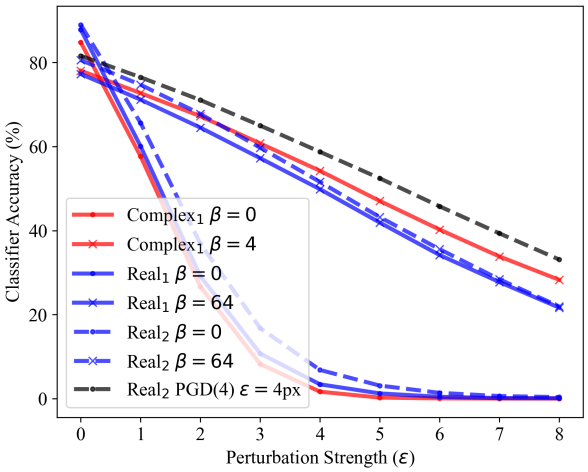


Figure 9. White-Box PGD(8) Attack of varying L_∞ bound (ϵ) on the CIFAR-10 test set. ϵ values are measured in fractions of pixel values, e.g. $\epsilon(4)$ is equivalent to $\frac{4}{255} \in [0, 1]$.

Black-Box Transfer Attacks

In preparation for the black-box transfer attack experiment, we conduct white-box FGSM attacks (Goodfellow et al., 2014) on independently trained real- and complex-valued networks on the MNIST, SVHN, and FashionMNIST benchmarks. For each network type, we train a network on just the standard objective and another with gradient regularization. For fair comparison, we train both the real- and complex-valued gradient regularized networks with $\beta = 64$ on MNIST, $\beta = 32$ on SVHN, and $\beta = 64$ on FashionMNIST. The FGSM attack is used because of its higher transferability (Kurakin et al., 2016); FGSM examples crafted from each network use $\epsilon = 0.16$ for MNIST, $\epsilon = 0.10$ for SVHN, and $\epsilon = 0.16$ for FashionMNIST. The examples are then transferred to other real- and complex-valued networks trained with or without gradient regularization. The transfer results are summarized in Table 2.

In general, FGSM attack was most effective on the white box network and was less effective when transferred to other models. This suggests that the success of the gradient regularized defense of either real- or complex-valued networks is not due to gradient masking (Athalye et al., 2018); this finding agrees with the result of Finlay & Oberman (2021).

We observe that attacks tend to be more transferable between networks of like-type ($\mathbb{R}$ to $\mathbb{R}$, $\mathbb{C}$ to $\mathbb{C}$), and that gradient regularized networks are more susceptible to attacks crafted by other gradient regularized networks, similar to the finding by Ross & Doshi-Velez (2018). The gradient regularized networks are the most resistant to transfer attack from any source. However, the complex-valued gradient regularized network is consistently the most resistant. This trend holds whether the complex-valued network is attacked by a real- or complex-valued network, gradient regularized, or standard-trained.

Table 2. Classification accuracy (%) of various real- ($\mathbb{R}$) and complex-valued ($\mathbb{C}$) networks on FGSM examples transferred from independently trained networks. Entries on the left and right are the classification accuracy of a victim network on adversarial examples transferred from independently trained standard and gradient regularized networks, respectively. We abbreviate FashionMNIST as FMNIST. The "self" row lists the resulting accuracy of the network from which the examples were generated.

Transfer to Network:	MNIST $\epsilon = 0.16$ $\beta = 0/64$	SVHN $\epsilon = 0.10$ $\beta = 0/32$	FMNIST $\epsilon = 0.16$ $\beta = 0/64$
FGSM From Real-Valued Network (Std./G.R.)			
Self	22.5 / 86.6	4.1 / 32.5	2.2 / 53.1
$\mathbb{R}$ (Std.)	36.2 / 74.0	10.3 / 32.0	3.9 / 28.3
$\mathbb{C}$ (Std.)	93.7 / 93.1	22.8 / 40.5	12.6 / 33.7
$\mathbb{R}$ (G.R.)	93.0 / 91.5	52.9 / 34.9	63.9 / 53.8
$\mathbb{C}$ (G.R.)	**95.3 / 95.8**	**55.7 / 41.9**	**68.5 / 60.4**
FGSM From Complex-Valued Network (Std./G.R.)			
Self	58.4 / 93.9	10.4 / 36.7	1.7 / 53.4
$\mathbb{R}$ (Std.)	86.5 / 88.0	50.1 / 31.5	32.4 / 30.7
$\mathbb{C}$ (Std.)	93.1 / 95.7	35.5 / 36.3	15.9 / 31.2
$\mathbb{R}$ (G.R.)	97.1 / 95.8	63.0 / 37.8	70.2 / 57.6
$\mathbb{C}$ (G.R.)	**97.3 / 96.4**	**65.4 / 41.5**	**74.7 / 58.4**

Comparisons using Identical Architecture

We run additional attacks on real- and complex-valued networks using the CNN structure of Ross & Doshi-Velez (2018). The architecture is: 5x5C32, 32BN($\mathbb{R}$), ActFunc($\mathbb{R}$), 2x2AP($\mathbb{R}$), 5x5C64, 64BN($\mathbb{R}$), ActFunc($\mathbb{R}$), 2x2AP($\mathbb{R}$), 1024FC($\mathbb{R}$), 1024BN($\mathbb{R}$), ReLU($\mathbb{R}$), 10FC($\mathbb{R}$). For the real-valued CNN, we employ real-valued convolutions, and the activation function (ActFunc) is ReLU; for the complex-valued CNN, we employ complex-valued convolutions and no activation function (ActFunc is the identity function) following the convolutions. Note that both CNNs have real-valued fully-connected layers with ReLU activation function. Hence, the only difference between the complex- and real-valued networks for this experiment are the type of convolution and use of ReLU non-linearity. All networks have the same input preprocessing (no Gaussian noise), and there is no "summing of features" for the complex-valued convolutions. During training, we employ dropout ($p = 0.5$) on the 1024FC($\mathbb{R}$) layer, cross-entropy loss, and the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-4}$).

Robustness to White-Box Attacks

Like the previous white-box experiments, we generate PGD(8) attacks against each of the networks. Starting with

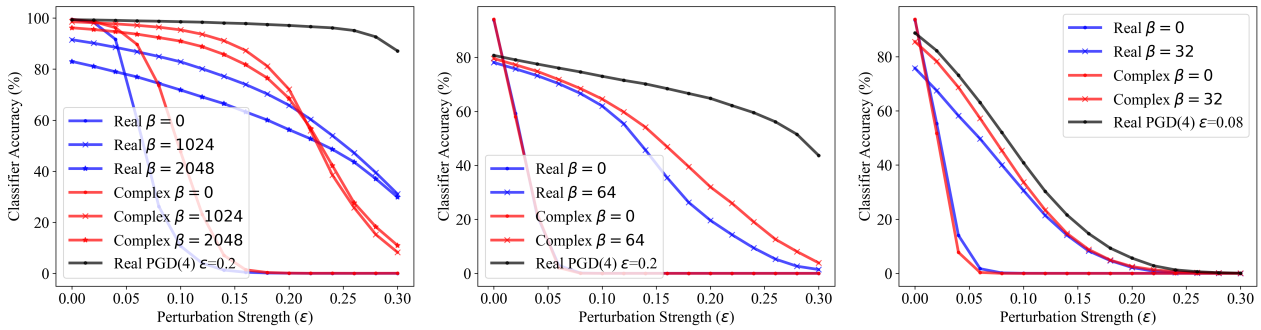


Figure 10. White-box PGD(8) attack results on the test sets of MNIST (left), FashionMNIST (center), and SVHN (right). All networks share the same architecture inspired from Ross & Doshi-Velez (2018).

the unaltered testing dataset, the accuracy of the real- and complex-valued networks is recorded as attack strength ϵ is increased. Figure 10 depicts the white-box attack result.

On MNIST, the CVNNs trained with gradient regularization display increased resistance to decrease in performance on the classification objective: CVNNs trained with $\beta = 2048$ gradient regularization maintain 96% clean accuracy, whereas real-valued networks achieve an accuracy of 83%. On FashionMNIST, the CVNNs and gradient regularization have increased resistance to PGD(8) examples, with CVNNs and real-valued networks maintaining 47% and 35% accuracy, respectively, for $\epsilon = 0.16$ examples. On SVHN, the CVNNs maintain 10% higher accuracy on the clean test set.

Robustness to Black-Box Query-based Attacks

We evaluate the robustness of the networks to the black-box query-based NES attack (Ilyas et al., 2018). This is a gradient-free method to construct adversarial examples: adversarial perturbations are crafted by querying classifier inputs and estimating the direction of change of the decision function. For an 8-step NES attack on 1000 FashionMNIST test images with an L_∞ bound of $\epsilon = 0.16$ and query budget of 4000 queries per image, real-valued networks attain accuracies of 0%, 62.3%, and 76.3% for no defense, $\beta = 64$ gradient regularization, and $\epsilon = 0.2$ adversarial training, respectively. Complex networks attain accuracies of 0% and 68.4% for no defense and $\beta = 64$ gradient regularization, respectively. Gradient regularization, especially when paired with a CVNN, provides resistance to the gradient-free NES attack.

4. Discussion

Initially, we experienced training instability when crafting deeper complex-valued networks with stacked convolutions, which was significantly reduced by using batch normalization (Ioffe & Szegedy, 2015) along the output dimension of each convolution (after summing features). However, simpler, wide architectures for real- and complex-valued tended to have the best gradient regularized training performance, and we used them for our experiments. There is a need for further exploration of deep network architectures and optimization techniques that can take advantage of the complex-valued features.

While the complex-valued network's robust performance approaches that of the state-of-the-art empirical defense, adversarial training, more investigation of attack-independent defense techniques with low storage and computation overhead is needed. We hope that the complex-valued network described in this work will promote ideation of new forms of neural network computation that are better suited to solve the machine learning security problem.

5. Conclusion

We have developed and analyzed a new form of complex-valued multi-layer neural network for secure computer vision applications. The complex-valued network can learn locally stable feature representations for inputs it is trained on, and its gradient update properties make it more able to satisfy both the standard and gradient regularized objectives. With comparable storage and computational requirement, the proposed complex-valued network can outperform real-valued networks trained with gradient regularization on a variety of image classification benchmarks. The robust performance of the gradient regularized complex-valued network approaches that of adversarially trained real-valued networks, without having prior knowledge of an attack.

Acknowledgements

This work is supported by AFRL FA8750-18-2-0121 and DOE DE-SC0021335.

References

Amin, M. F. and Murase, K. Single-layered complex-valued neural network for real-valued classification problems. Neurocomputing, 72(4-6):945–955, 2009.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International Conference on Machine Learning, pp. 274–283. PMLR, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pp. 39–57. IEEE, 2017.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9185–9193, 2018.

Drucker, H. and Le Cun, Y. Improving generalization performance using double backpropagation. IEEE Transactions on Neural Networks, 3(6):991–997, 1992.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1625–1634, 2018.

Finlay, C. and Oberman, A. M. Scaleable input gradient regularization for adversarial robustness. Machine Learning with Applications, 3:100017, 2021.

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

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

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

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In International Conference on Machine Learning, pp. 2137–2146. PMLR, 2018.

Inkawhich, N., Wen, W., Li, H. H., and Chen, Y. Feature space perturbations yield more transferable adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7066–7074, 2019.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pp. 448–456. PMLR, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236, 2016.

Kurakin, A., G. I. J. and Bengio, S. Adversarial examples in the physical world. In Proceedings of the International Conference on Learning Representations, 2017.

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

Lyu, C., Huang, K., and Liang, H.-N. A unified gradient regularization family for adversarial examples. In 2015 IEEE international conference on data mining, pp. 301–309. IEEE, 2015.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Netzer, Y., Wang, T., Coates, A., Biassacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep learning and Unsupervised Feature Learning, 2011.

Pang, T., Xu, K., Du, C., Chen, N., and Zhu, J. Improving adversarial robustness via promoting ensemble diversity. In International Conference on Machine Learning, pp. 4970–4979. PMLR, 2019.

Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc., 2019.

Ross, A. S. and Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In Proceedings of the 18th AAAI Conference on Artificial Intelligence (AAAI 2018), pp. 1660–1669, 2018.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

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

Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J. a. F., Mehri, S., Rostamzadeh, N., Bengio, Y., and Pai, C. J. Deep complex networks. In Proceedings of the International Conference on Learning Representations, 2018.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5028–5037, 2017.

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

Yang, H., Zhang, J., Dong, H., Inkawhich, N., Gardner, A., Touchet, A., Wilkes, W., Berry, H., and Li, H. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. arXiv preprint arXiv:2009.14720, 2020.

Zhou, W., Hou, X., Chen, Y., Tang, M., Huang, X., Gan, X., and Yang, Y. Transferable adversarial perturbations. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 452–467, 2018.