Addressing Neural Network Robustness with Mixup and Targeted Labeling Adversarial Training

Alfred Laugros, Alice Caplier, Matthieu Ospici

Addressing Neural Network Robustness with Mixup and Targeted Labeling Adversarial Training

Alfred LAUGROS[1,2], Alice CAPLIER[2], and Matthieu OSPICI[1]

[1] Atos, France {alfred.laugros,matthieu.ospici}@atos.net
[2] Univ. Grenoble Alpes, France alice.caplier@grenoble-inp.fr

Abstract. Despite their performance, Artificial Neural Networks are not reliable enough for most of industrial applications. They are sensitive to noises, rotations, blurs and adversarial examples. There is a need to build defenses that protect against a wide range of perturbations, covering the most traditional common corruptions and adversarial examples. We propose a new data augmentation strategy called M-TLAT and designed to address robustness in a broad sense. Our approach combines the Mixup augmentation and a new adversarial training algorithm called Targeted Labeling Adversarial Training (TLAT). The idea of TLAT is to interpolate the target labels of adversarial examples with the ground-truth labels. We show that M-TLAT can increase the robustness of image classifiers towards nineteen common corruptions and five adversarial attacks, without reducing the accuracy on clean samples.

Keywords: Neural Network, Robustness, Common Corruptions, Adversarial Training, Mixup

1 Introduction

Artificial neural networks have been proven to be very efficient in various image processing tasks [26],[38],[43]. Unfortunately, in real-world computer vision applications, a lot of common corruptions may be encountered such as blurs, colorimetry variations, or noises, etc. Such corruptions can dramatically decrease neural network efficiency [11],[7],[23],[48]. Besides, deep neural networks can be easily attacked with adversarial examples [42]. These attacks can reduce the performance of most of the state-of-the-art neural networks to zero [1],[37].

Some techniques have been proposed to make neural networks more robust. When a specific perturbation is considered, we can build a defense to protect a neural network against it, whether it is a common corruption [25],[58],[48] or an adversarial attack [33],[54],[39]. However, increasing robustness towards a specific perturbation generally does not help with another kind of perturbation. For instance, geometric transformation robustness is orthogonal with worst-case additive noise [10]. Fine tuning on blur does not increase the robustness to Gaussian noise [8]. Even worse, making a model robust to one specific perturbation can

make it more sensitive to another one. For instance, a data augmentation with corruptions located in the high frequency domain tends to decrease the robustness to corruptions located in the low frequency domain [55]. Besides, increasing adversarial robustness often implies a diminution of the *clean accuracy* (the accuracy of a model on not-corrupted samples) [27],[47]. Therefore, a company or a public organism may feel reluctant to use neural networks in their projects since it is hard to make them robust to a large diversity of corruptions, and it is difficult to predict how a neural network will react to unexpected corruptions. There is a need to build new defenses that address robustness in a broad sense, covering the most encountered common corruptions and adversarial examples.

We propose to address this issue with a new data augmentation approach called M-TLAT. M-TLAT is a combination of Mixup [57] and a new kind of adversarial training called Targeted Labeling Adversarial Training (TLAT). The idea of this adversarial training is to label target adversarial examples with soft labels that contain information about the used target. We show that M-TLAT can increase the robustness of image classifiers to nineteen common corruptions and five adversarial attacks, without reducing the accuracy on clean samples. This algorithm is easy to implement and to integrate to an existing training process. It intends to make the neural networks used in real-world applications more reliable.

2 Related Works

2.1 Protecting Neural Networks against Common Corruptions

Neural Networks are known to be sensitive to a lot of perturbations such as noises [25], rotations [11], blurs [48] or colorimetry variations [23], etc. We call these perturbations common corruptions. They are often encountered in industrial applications, but generally absent from academic datasets. For instance the faces in the dataset celeba are always well illuminated with a consistent eyes positioning [32], yet those conditions are not always guaranteed in the industrial applications. Because of common corruptions, the performance of a neural network can be surprisingly low [14].

There are a few methods that succeded in increasing the robustness to several common corruptions simoultaneously. Among them, the robust pre-training algorithm proposed by Liu et al. can make classifiers more robust to noises, occlusions and blurs [31]. In [13], it is proposed to change the style of the training images using style transfer [12]. Neural networks trained this way are obliged to focus more on shapes than textures. The Augmix algorithm proposes to interpolate a clean image with perturbed versions of this image [18]. The obtained images are used to augment the training set of a neural network. These methods are useful to make neural networks more robust to common corruptions but they do not address the case of adversarial examples.

2.2 Protecting Neural Networks against Adversarial Examples

Adversarial Examples are another threat that can make neural networks give unexpected answers [42]. Unlike common corruptions they are artificial distortions. They are crafted by humans so as to especially fool neural networks. Adversarial examples can completely fool even the state-of-the-art models [27]. Those perturbations are even more dangerous because humans can hardly see if an image has been adversarially corrupted or not [1],[37]. In other words, a system can be attacked without anyone noticing it. They are two kinds of adversarial examples called white-box and black-box attacks.

White-box attacks. The attacker has access to the whole target network: its parameters and its architecture. White-box attacks are tailored so as to especially fool a specific network. White-box adversarial examples are very harmful, defending a model against it is a tough task [47],[40],[3].

Black-Box attacks. An adversarial example crafted with a limited access to the targeted network is called a black-box attack. When only the training set of a neural network is known, it is still possible to make a transfer attack. Considering a dataset and two neural networks trained with it, it has been shown that an adversarial example crafted using one of the models, can harm the other one [42],[15]. This phenomenon occurs even when the two models have distinct architectures and parameters.

A lot of methods have been proposed to protect against adversarial examples. Adversarial training uses adversarial examples to augment the training set of a neural network [33],[46]. Defense-Gan [39] and feature squeezing [54] are used to remove adversarial patterns from images. Stochastic Activation Pruning [6] and defensive dropout [51] make the internal operations of neural networks more difficult to access in order to make these networks more difficult to attack. These methods can significantly increase the robustness of neural networks to adversarial examples but they do not provide any protection towards common corruptions.

2.3 Addressing Robustness in a Broad Sense

The methods mentioned above increase either the adversarial robustness or the robustness to common corruptions. Unfortunately, increasing the robustness to common corruptions generally does not imply increasing the robustness to adversarial examples and conversely [29]. The experiments carried out in [55] show that data augmentation with traditional adversarial examples makes models less robust to low frequency corruptions. Robustness to translations and rotations is independent from robustness to the L_p bounded adversarial examples [10].

A natural approach to address robustness in a broad sense is to combine defenses that address common corruptions with defenses that address adversarial examples. Unfortunately, it is possible that two defenses are not compatible and do not combine well [45],[55],[21].

A few standalone methods have been recently proposed to address adversarial robustness and robustness to common corruptions at the same time. In [34],[52], a large set of unlabeled data are leveraged to get a significant increase in common corruption robustness and a limited increase in adversarial robustness. However, using these methods has a prohibitive computational cost. Adversarial Noise Propagation adds adversarial noise into the hidden layers of neural networks during the training phase to address both robustnesses [30]. Adversarial Logit Pairing (ALP) encourages trained models to output similar logits for adversarial examples and their clean counterparts [22]. In addition to the adversarial robustness provided, it has been reported that ALP increases the robustness to some common corruptions [17]. The drawback of these methods is that they reduce the clean accuracy of trained models.

To be useful in real-world applications, we want our method to preserve the clean accuracy of the trained models, to increase the robustness to both adversarial examples and common corruptions, and to be easy to integrate into an existing training framework.

3 Combining Mixup with Targeted Labeling Adversarial Training: M-TLAT

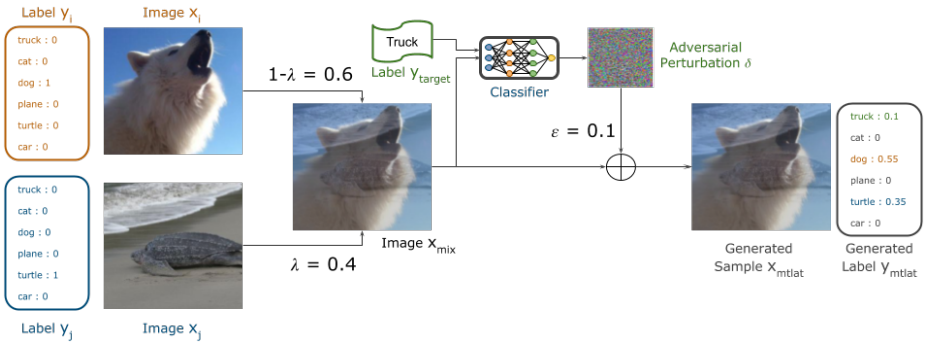

Fig. 1: Visualising of the generation of a new training couple using M-TLAT

Our approach called M-TLAT is a combination of two data augmentation algorithms, which are Mixup [57] and a new adversarial training strategy called Targeted Labeling Adversarial Training (TLAT). Basically, Mixup aims to increase the common corruption robustness while TLAT aims to increase the adversarial robustness. We go more into details in Section 5.3 to understand the contribution of each component. In practice, we observe that those augmentations combine well to address robustness in a broad sense.

3.1 Mixup

Let us consider the couples (x_i, y_i) and (x_j, y_j), where x_i and x_j are images of the training set and y_i and y_j are their associated one-hot encoding labels. Mixup [57] is a data augmentation algorithm that interpolates linearly samples and labels (see Figure 1):

$$\begin{aligned} x_{mix} &= \lambda * x_i + (1 - \lambda) * x_j \\ y_{mix} &= \lambda * y_i + (1 - \lambda) * y_j \end{aligned} \tag{1}$$

where λ is drawn from a Beta distribution defined by an hyperparameter α: $\lambda \sim Beta(\alpha, \alpha)$. This augmentation strategy encourages the trained models to have a linear behavior in-between training samples [57],[49]. In practice, Mixup reduces the generalization error of classifiers, makes neural networks less sensitive to corrupted labels and slightly improves the robustness to adversarial examples. In the ablation study carried out in Section 5.3, we detail the influence of Mixup on neural network robustness.

Augmenting datasets by interpolating samples of training sets have been largely studied [44],[20]. Among the most successful, the Manifold Mixup interpolates hidden representations instead of interpolating the inputs directly [49]. The Mixup Inference proposes to use Mixup in the inference phase to degrade the perturbations that may corrupt the input images [35]. Directional Adversarial Training and Untied Mixup are alternative policies to pick the interpolation ratios of the *mixuped* samples and labels [2]. The proposed M-TLAT algorithm uses the standard Mixup, but it is not incompatible with the other interpolation strategies mentioned in this paragraph.

3.2 Targeted Labeling Adversarial Training: TLAT

M-TLAT relies on a second data augmentation procedure: adversarial training, which consists in adding adversarial examples into the training set [15],[33]. It is one of the most efficient defenses against adversarial examples [3]. We consider an unperturbed sample x_{clean} of size S and its label y_{clean}. We can corrupt x_{clean} to build an adversarial example x_{adv}:

$$\begin{aligned} x_{adv} &= x_{clean} + \underset{\delta \in [-\epsilon, \epsilon]^S}{arg\,max} \{L(x_{clean} + \delta, y_{clean}, \theta)\} \\ y_{adv} &= y_{clean} \end{aligned} \tag{2}$$

Where θ are the parameters of the attacked model. L is a cost function like the cross-entropy function. The value ϵ defines the amount of the introduced adversarial perturbation. As suggested in [1], adversarial training is even more efficient when it uses adversarial examples that target a specific class y_{target}. The augmentation strategy becomes:

$$\begin{aligned} x_{adv} &= x_{clean} - \underset{\delta \in [-\epsilon, \epsilon]^S}{arg\,max}\{L(x_{clean} + \delta, y_{target}, \theta)\} \\ y_{adv} &= y_{clean} \end{aligned} \tag{3}$$

One advantage to use target adversarial examples during training is to prevent label leaking [27]. We propose to improve this augmentation strategy by using y_{target} in the labeling of the adversarial examples. In particular, we propose to mix the one-hot encoding ground-truth labels of the original samples with the one-hot encoding target labels used to craft the adversarial examples:

$$\begin{aligned} x_{adv} &= x_{clean} - \underset{\delta \in [-\epsilon, \epsilon]^S}{arg\,max}\{L(x_{clean} + \delta, y_{target}, \theta)\} \\ y_{adv} &= (1 - \epsilon) * y_{clean} + \epsilon * y_{target} \end{aligned} \tag{4}$$

We call this augmentation strategy Targeted Labeling Adversarial Training (TLAT). The $arg\,max$ part is approximated with an adversarial example algorithm such as target FGSM [1]:

$$x_{adv} = x_{clean} - \epsilon * sign(\nabla_{x_{clean}} L(x_{clean}, y_{target}, \theta)) \tag{5}$$

FGSM is used because it is a computationally efficient way to craft adversarial examples [15],[46]. As for traditional adversarial trainings, models trained with TLAT should be trained on both clean samples and adversarial samples [1]. The advantage of TLAT is to make models have a high clean accuracy compared to the models trained with a standard adversarial training algorithm. More details can be found in Section 5.4.

TLAT uses soft labels instead of one-hot encoding labels. Using soft labels is a recurrent idea in the methods that address robustness. As mentioned above, Mixup interpolates training labels to generate soft labels [57]. Label smoothing replaces the zeros of one-hot encoding labels by a smoothing parameter $s > 0$ and normalizes the high value so that the distribution still sums to one [41]. Distillation learning uses the logits of a trained neural network to train a second neural network [36]. The second network is enforced to make smooth predictions by learning on soft labels. Bilateral training generates soft labels by adversarially perturb training labels [50]. It uses the gradient of the cost function of an attacked model to generate adversarial labels. Models trained on both adversarial examples and adversarial labels are encouraged to have a small gradient magnitude which makes them more robust to adversarial examples.

The originality of TLAT is to use the target labels of adversarial attacks as a component of the soft labels. Intuitions about why this labeling strategy works are provided in Section 5.4.

3.3 M-TLAT

The idea of M-TLAT is to generate new training couples by applying sequentially the TLAT perturbations (4) after the Mixup interpolations (1):

$$\begin{aligned} x_{mtlat} &= x_{mix} - \underset{\delta \in [-\epsilon,\epsilon]^S}{arg\,max}\{L(x_{mix}+\delta, y_{target}, \theta)\} \\ y_{mtlat} &= (1-\epsilon) * y_{mix} + \epsilon * y_{target} \end{aligned} \tag{6}$$

As displayed in Figure 1, x_{mtlat} contains features that come from three distinct sources: two clean images and an adversarial perturbation that targets a specific class. The label y_{mtlat} contains the class and the weight associated with each source of features. These weights are determined by the values λ and ϵ. A model trained with M-TLAT is not only constraint to predict the classes that correspond to the three sources. It also has to predict the weight of each source within the features of x_{mtlat}. We believe that being able to predict the class and the weighting of the three sources requires a subtle understanding of the features present in images. In practice, being trained with augmented couples (x_{mtlat}, y_{mtlat}) makes neural networks more robust.

The expressions (6) are the essence of the algorithm. The whole process of one training step with M-TLAT is provided in the algorithm description 1. As recommended in [1], the training minibatches contain both adversarial and non-adversarial samples. In our algorithm, the non-adversarial samples are obtained using a standard Mixup interpolation, the adversarial samples are crafted by combining Mixup and TLAT.

Another recently proposed approach combines Mixup and adversarial training [28]. Their method called Interpolated Adversarial Training (IAT) is different from M-TLAT for two main reasons. Most importantly, they do not use the labeling strategy of TLAT. Basically, their adversarially corrupted samples are labeled using the standard Mixup interpolation while our labels contain information about the amount and the target of the used adversarial examples. Secondly, we interpolate images before adding the adversarial corruptions. On the contrary they adversarially corrupt images before mixing them up. In practice, we get better adversarial robustness when we proceed in our order. Besides, proceeding in their order doubles the number of adversarial perturbations to compute: it increases the training time. In Section 5.2, we compare the robustness of two models trained with these approaches.

4 Experiment Set-up

4.1 Perturbation Benchmark

We want to evaluate the robustness of neural networks in a broad sense, covering the most encountered common corruptions and adversarial examples. To achieve this, we gather a large set of perturbations that contains nineteen common corruptions and five adversarial attacks.

Algorithm 1 One training step of the M-TLAT algorithm

Require: θ the parameters of the trained neural network
Require: L a cross entropy function
Require: $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \sim$ Dataset
Require: $\lambda_1, \lambda_2 \sim Beta(\alpha, \alpha)$
Require: $\epsilon \sim U[0, \epsilon_{max}]$ where U is a uniform distribution and ϵ_{max} is the maximum perturbation allowed
Require: $y_{target} \sim U[0, N]$ where N is the number of classes
Require: *optim* an optimizer like Adam or SGD

$x_{mix1} = \lambda_1 * x_1 + (1 - \lambda_1) * x_2$
$y_{mix1} = \lambda_1 * y_1 + (1 - \lambda_1) * y_2$

$x_{mix2} = \lambda_2 * x_3 + (1 - \lambda_2) * x_4$
$y_{mix2} = \lambda_2 * y_3 + (1 - \lambda_2) * y_4$

$x_{mtlat} = x_{mix2} - \epsilon * sign(\nabla_{x_{mix2}} L(x_{mix2}, y_{target}, \theta))$
$y_{mtlat} = (1 - \epsilon) * y_{mix2} + \epsilon * y_{target}$

$loss_1 = L(x_{mix1}, y_{mix1}, \theta)$
$loss_2 = L(x_{mtlat}, y_{mtlat}, \theta)$
$loss = loss_1 + loss_2$
$gradients = \nabla_\theta loss$
$optim.update(gradient, \theta)$ The optimizer updates θ according to the gradients

Common Corruptions. The benchmark includes the common corruptions of ImageNet-C [17]. The ImageNet-C set of perturbations contains diverse kinds of common corruptions such as 1) noises: Gaussian noise, shot noise and impulse noise 2) blurs: defocus blur, glass blur, motion blur and zoom blur 3) weather related corruptions: snow, frost, fog and brightness 4) digital distortions: contrast, elastic, pixelate and jpeg compression. Each corruption is associated with five severity levels. We use the corruption functions[3] provided by the authors to generate those perturbations.

The rotation, translation, color distortion and occlusion common corruptions are absent from ImageNet-C, yet they are often encountered in industrial applications. We decided to add those four perturbations to the pool of common corruptions. The occlusion perturbation is modeled by masking a randomly selected region of images with a grey square. For the images corrupted with color distortion, one of the RGB channel is randomly chosen and a constant is added to all the pixels of this channel. For the rotation and translation perturbations, the pixel values of the area outside the transformed images are set to zero.

These four corruptions are associated with a severity range. The lower bound of the severity range has been set to reduce the accuracy of the standard ResNet-50 by five percent on the test set. The upper bound has been set to reduce its accuracy by thirty percent. As a result, the rotations can turn images from 8 to

[3] https://github.com/hendrycks/robustness

40 degrees. In other words, when the images of the test set are rotated by eight degrees, the accuracy of the standard ResNet-50 is five percent lower than for a not-corrupted test set. The side of the square mask used in the occlusions varies from 60 to 127 pixels. The translations can move images from 15 to 62 pixels. For the color distortion corruption, the values of the pixels of the perturbed channel are increased from 8% to 30% of the maximum possible pixel value.

Adversarial Examples. Following the recommendations in [5], we carefully choose the adversarial examples to be added to the benchmark. Firstly, we use adversarial examples in both white box and black box settings. Secondly, we use two different metrics, the L_∞ and the L_2 norms, to compute the bound of adversarial attacks. Thirdly, we employ targeted and untargeted attacks. Fourthly, several amounts of adversarial perturbations are used. Finally, the selected adversarial examples are not used during trainings. We build a set of adversarial examples to cover all the conditions mentioned above.

We note ϵ the amount of the introduced perturbation in adversarial examples. We use PGD with $\epsilon = 0.04$ as a white-box L_∞ bounded attack [33]. We generate targeted adversarial attacks by using PGD_LL, which targets the least likely class according to attacked models [1]. We use MI_FGSM with $\epsilon = 0.04$ and $\epsilon = 0.08$ as black-box attacks [9]. A VGG network trained on the same training set is used to craft these black-box attacks. PGD, PGD_LL and MI_FGSM are computed over ten iterations. We use the Carlini-Wagner attack (CW_2) as a L_2 white-box bounded attack [4]. We perform the optimization process of this attack with 40 iterations and a confidence score of 50.

The gathered common perturbations and adversarial examples constitute the perturbation benchmark. It is used in the experimental section in order to evaluate and compare the robustness of models.

4.2 Training Details

The trained models are either a ResNet-50 [16] or a ResNeXt-50 with the 32x4d template [53]. We use a batch size of 256 and 90 training epochs. The Adam optimizer [24] is used with a learning rate of 0.002 and a weight decay of 10^{-4}. At the end of the epochs 30, 60 and 80, the learning rate is divided by 10. The cost function is the cross entropy. We use the same hyperparameters for all trainings.

All models are trained and tested using ImageNet. Because of a limited computational budget, we used a subset of ImageNet built on 100 randomly chosen classes. For each class, ten percent of the images are preserved for the test set. Then, the training set and the test set contain respectively 10^5 and 10^4 images. The only pre-processing used is the resizing of images to the 224*224 format.

We call the models trained without any data augmentation the standard models. We observed that the highest clean accuracy for the models trained with mixup is reached when $\alpha = 0.4$, so we used this value in all experiments. The adversarial examples used in trainings are crafted using FGSM with $\epsilon \sim U[0, 0.025]$. The range of pixel values of images in our experiments is $[0, 1]$.

5 Performance Evaluation

5.1 Robustness Score

To measure the robustness of a model to a perturbation, we compare the performance of this model on clean samples with its performance on perturbed samples:

$$R_N^\phi = \frac{A_\phi}{A_{clean}} \tag{7}$$

We call R_N^ϕ the robustness score of a neural network N towards a perturbation ϕ. A_{clean} is the accuracy of the model on the clean test set and A_ϕ is its accuracy on the test set corrupted with the ϕ perturbation. ϕ can be either a common corruption or an adversarial attack.

This robustness score metric should be used carefully because it masks the clean accuracy of neural networks. Indeed, an untrained model that always makes random outputs, would have the same accuracy for clean samples than for corrupted samples. Its robustness scores would be equal to 1, so it could be considered as completely robust to any common corruption. Therefore, in this study, before comparing the robustness scores of two neural networks we always make sure that their clean accuracies are also comparable.

5.2 Performances of M-TLAT on the Perturbation Benchmark

For the first experiment we train one Resnet-50 and one ResNeXt-50, using the M-TLAT algorithm. The training of these models took a dozen of hours using a single GPU Nvidia Tesla V100. We also train one Resnet-50 and one ResNeXt-50 with the IAT algorithm [28]. We compute the robustness scores of the trained models towards all the perturbations of the benchmark. The results are reported in Table 1.

In Tables 1 and 2, the *Clean* column contains the accuracy of the models on the not-corrupted test set. Each of the other columns contains the robustness scores to a perturbation of the benchmark. For the corruptions of the ImageNet-C benchmark, the displayed scores correspond to the mean robustness score of the corruptions computed with their five severity levels. To better visualize the effect of the augmentation algorithms, we use either a "-" index or a "+" index, to signify if a model is less or more robust than the standard model to a perturbation.

We observe in Table 1 that the models trained with M-TLAT are slightly more accurate than the standard models on clean images. They are also more robust than the standard models to every single tested common corruption. We see that using M-TLAT makes neural networks much more robust to the CW_2 and PGD_LL attacks. It also makes models less sensitive to black-box adversarial examples. We observe that the robustness gain for the PGD attack is less important.

Table 1: Effect on robustness of M-TLAT and comparison with IAT

(a) Robustness scores towards the common corruptions of ImageNet-C

		Clean	Gauss	Shot	Impul	Defocus	Glass	Motion	Zoom	Snow	Fog	Frost	Bright	Contr	Elastic	Pixelate	Jpeg
ResNet	standard	73.3	0.17	0.17	0.12	0.25	0.35	0.41	0.47	0.31	0.48	0.34	0.78	0.34	0.73	0.65	0.80
	IAT	73.2^{-}	0.47^{+}	0.46^{+}	0.42^{+}	0.51^{+}	0.65^{+}	0.58^{+}	0.68^{+}	0.49^{+}	0.78^{+}	0.66^{+}	0.79^{+}	0.78^{+}	0.84^{+}	0.82^{+}	0.63^{-}
	M-TLAT	73.9^{+}	0.56^{+}	0.54^{+}	0.52^{+}	0.41^{+}	0.61^{+}	0.58^{+}	0.63^{+}	0.49^{+}	0.59^{+}	0.61^{+}	0.82^{+}	0.58^{+}	0.85^{+}	0.94^{+}	0.95^{+}
ResNeXt	standard	76.4	0.25	0.25	0.20	0.28	0.37	0.44	0.48	0.36	0.53	0.37	0.79	0.36	0.75	0.72	0.74
	IAT	74.7^{-}	0.46^{+}	0.44^{+}	0.43^{+}	0.53^{+}	0.67^{+}	0.62^{+}	0.70^{+}	0.51^{+}	0.73^{+}	0.69^{+}	0.81^{+}	0.80^{+}	0.85^{+}	0.83^{+}	0.59^{-}
	M-TLAT	76.5^{+}	0.57^{+}	0.55^{+}	0.54^{+}	0.44^{+}	0.64^{+}	0.60^{+}	0.66^{+}	0.52^{+}	0.68^{+}	0.67^{+}	0.86^{+}	0.70^{+}	0.85^{+}	0.95^{+}	0.95^{+}

(b) Robustness scores towards our additional common corruptions

		Obstru	Color	Trans	Rot
ResNet	standard	0.74	0.76	0.78	0.68
	IAT	0.71^{-}	0.89^{+}	0.75^{-}	0.72^{+}
	M-TLAT	0.75^{+}	0.86^{+}	0.79^{+}	0.74^{+}
ResNeXt	standard	0.75	0.82	0.82	0.72
	IAT	0.72^{-}	0.90^{+}	0.77^{-}	0.71^{-}
	M-TLAT	0.76^{+}	0.89^{+}	0.82	0.74^{+}

(c) Robustness scores towards adversarial examples

		pgd ϵ=0.04	pgd_ll ϵ=0.04	cw_l2	mi_fgsm ϵ=0.04	mi_fgsm ϵ=0.08
ResNet	standard	0.00	0.00	0.00	0.58	0.34
	IAT	0.01^{+}	0.08^{+}	0.84^{+}	0.87^{+}	0.78^{+}
	M-TLAT	0.08^{+}	0.45^{+}	1.00^{+}	0.96^{+}	0.87^{+}
ResNeXt	standard	0.00	0.00	0.00	0.58	0.33
	IAT	0.01^{+}	0.11^{+}	0.95^{+}	0.87^{+}	0.81^{+}
	M-TLAT	0.09^{+}	0.38^{+}	0.99^{+}	0.95^{+}	0.87^{+}

For comparison, the IAT algorithm tends to reduce the clean accuracy. It does not increase the robustness to all the common corruption of the benchmark. In particular, it significantly decreases the robustness towards the Jpeg perturbation. Besides, the IAT models are significantly less robust to adversarial examples than the M-TLAT models.

Using FGSM during training is known to poorly defend models against iterative adversarial attack such as PGD [27]. The robustness of the M-TLAT models towards PGD can likely be increased by replacing FGSM by an iterative adversarial attack [33]. But this would increase significantly the training time. That is the reason why this option has not been tested yet.

To our knowledge, M-TLAT is the first data augmentation approach that is able to increase the robustness to every single common corruption and adversarial example of a large set of diverse perturbations, without reducing the clean accuracy.

5.3 Complementarity between Mixup and TLAT

To better understand the effect of each constituent of the M-TLAT algorithm, we proceed to an ablation study. Two ResNet-50 are respectively trained with the Mixup and TLAT data augmentations. We report their robustness to the perturbation benchmark in Table 2.

First, we notice that Mixup causes an increase of the clean accuracy, which is coherent with observations made in [57]. On the contrary, TLAT makes the trained model less accurate on the clean data. But those two effects seem to cancel each other because the M-TLAT model and the standard model have comparable clean accuracies as observed in Table 1a.

Table 2: Influence on robustness of the Mixup and TLAT data augmentations

(a) Robustness scores towards the corruptions of ImageNet-C

		Clean	Gauss	Shot	Impul	Defocus	Glass	Motion	Zoom	Snow	Fog	Frost	Bright	Contr	Elastic	Pixelate	Jpeg
ResNet	standard	73.3	0.17	0.17	0.12	0.25	0.35	0.41	0.47	0.31	0.48	0.34	0.78	0.34	0.73	0.65	0.80
	Mixup	74.9^{+}	0.28^{+}	0.28^{+}	0.24^{+}	0.25	0.38^{+}	0.39^{-}	0.53^{+}	0.38^{+}	0.79^{+}	0.53^{+}	0.78	0.75^{+}	0.75^{+}	0.58^{-}	0.61^{-}
	TLAT	69.4^{-}	0.57^{+}	0.54^{+}	0.51^{+}	0.43^{+}	0.60^{+}	0.56^{+}	0.60^{+}	0.41^{+}	0.15^{-}	0.41^{+}	0.78	0.13^{-}	0.84^{+}	0.94^{+}	0.97^{+}

(b) Robustness scores towards our additional common corruptions

		Obstru	Color	Trans	Rot
ResNet	standard	0.74	0.76	0.78	0.68
	Mixup	0.75^{+}	0.88^{+}	0.79^{+}	0.71^{+}
	TLAT	0.69^{-}	0.76	0.67^{-}	0.66^{-}

(c) Robustness scores towards adversarial examples

		pgd ϵ=0.04	pgd_ll ϵ=0.04	cw_l2	mi_fgsm ϵ=0.04	mi_fgsm ϵ=0.08
ResNet	standard	0.00	0.00	0.00	0.58	0.34
	Mixup	0.00	0.00	0.202^{+}	0.61^{+}	0.22^{-}
	TLAT	0.10^{+}	0.74^{+}	0.97^{+}	0.98^{+}	0.93^{+}

In Tables 2a and 2b, we observe that Mixup makes the trained model more robust than the standard model to all the common corruptions but the *Motion Blur*, *Pixelate* and *Jpeg* corruptions. We observe in Table 2c that Mixup has a little influence on adversarial robustness, with either a slight increase or decrease of the robustness depending on the considered attack.

Fortunately, TLAT makes models much more robust to any adversarial attacks. Indeed, the TLAT model is much more difficult to attack with the black box adversarial examples or with the CW_2 attack. It is also significantly more robust to PGD_LL and slightly more robust to PGD. For common corruptions, the effect of TLAT is very contrasted. Concerning the noise and blur corruptions (the seven first corruptions of the Table 2a), the TLAT model is much more robust than the standard model. For some other common corruptions like *Fog* or *Contrast*, the TLAT augmentation decreases significantly the robustness scores.

It is clear that the M-TLAT models are much more robust to adversarial examples thanks to the contribution of TLAT. However, for common corruptions, Mixup and TLAT are remarkably complementary. Concerning the few corruptions for which Mixup has a very negative effect on robustness (*Jpeg* and *Pixelate*), TLAT has a strong positive effect. Similarly, for the *Fog* and *Contrast* corruptions, TLAT makes models less robust while Mixup makes them much more robust.

The ablation study indicates that both components are important to increase the robustness to a large diversity of perturbations.

5.4 Labeling in Adversarial Trainings

Comparison of the Labeling Strategies

Adversarial trainings increase the adversarial robustness of the trained models, but they also reduce their accuracy on clean samples [47],[27]. In this section, we want to show that TLAT decreases less the clean accuracy of the trained models than traditional adversarial trainings.

To achieve it, we trained four ResNet-50 with different kinds of adversarial training algorithms. The first model is trained using untarget FGSM and the second is trained using target FGSM with randomly chosen target. Both use adversarial examples labeled with the ground-truth labels. We train another model with target FGSM but regularized via label smoothing (LS) [41], we call it the LS model. For this model, we use a smoothing parameter equal to ϵ, where ϵ is the amount of the FGSM perturbation. In other words, the one values of the one-hot encoding vectors are replaced by $1 - \epsilon$ and the zeros are replaced by ϵ/N where N is the number of classes. The fourth model is trained using the TLAT algorithm. All models are trained with minibatches that contain both clean samples and adversarial examples. We measure the clean accuracy of those models: results are displayed in Table 3.

We see that TLAT is the adversarial training method that reduces the less the clean accuracy. This result shows that TLAT is important to preserve the clean accuracy of the models trained with M-TLAT, all the while making them more robust to adversarial examples.

Using soft labels in trainings is known to help models to generalize [36],[41]. Here we want to make sure that the usage of soft labels is not the main reason of high clean accuracy of TLAT. To achieve it, we compare the performances of the TLAT and LS models. Even if the LS model also uses soft labels during training, it performs worse than the TLAT model. Consequently, the good performances of TLAT are not due to the usage of soft labels. We believe TLAT performs well because it uses labels that contain information about the target of the adversarial examples.

Table 3: Comparison of the performances of the TLAT augmentation with the performances of other kinds of adversarial trainings

	standard	FGSM	target-FGSM	LS	TLAT
clean accuracy	73.3	65.8	68.3	67.1	69.4

Interpretation

The TLAT augmentation is motivated by the works of Ilyas et al [19]. In their study, they reveal the existence of brittle yet highly predictive features in the data. Neural networks largely depend on those features even if they are nearly invisible to human eye. They are called non-robust features. Ilyas et al. show that a model that only uses non-robust features to complete properly a task still generalize well on unseen data.

They use this phenomenon to interpret adversarial vulnerability. Adversarial attacks are small perturbations, so they mainly affect non-robust features. They can especially make those brittle features anti-correlated with the true label. As

neural networks largely rely on non-robust features, their behaviour is completely disturbed by adversarial attacks.

In Adversarial trainings, neural networks encounter adversarial examples that can have non-robust features uncorrelated with the true label. The trained models are then constrained to less rely on non-robust features. This can explain the success of adversarial training: adversarially trained models give less importance to non-robust features so they are much more difficult to attack with small perturbations.

Despite its efficiency, adversarial training generally causes a decrease in clean accuracy [47],[27],[56]. One possible reason could be that adversarial patterns in the training adversarial examples are not coherent with the ground-truth label. Consequently, an adversarially trained model encounters samples for which the features are not completely coherent with labelling.

The proposed method tries to make the labeling of target adversarial examples more correlated with its non-robust features. Targeted adversarial attacks make non-robust features of a sample correlated with a target class. So instead of labelling only with the ground-truth class of the attacked sample, the method introduces a part related to the target class. Therefore, the trained model still learns the true original class of the attacked sample, but the label used for learning is more correlated with the non-robust features of this sample. We believe this could be the reason why our method has better performance than traditional target adversarial training in practice.

6 Conclusion

We propose a new data augmentation strategy that increases the robustness of neural networks to a large set of common corruptions and adversarial examples. The experiments carried out suggest that the effect of M-TLAT is always positive: basically, it increases the robustness to any corruption without reducing the clean accuracy. We believe using M-TLAT can be particularly useful to help industrials to increase the robustness of their neural networks without being afraid of any counterpart.

As part of the M-TLAT algorithm, we use the new adversarial augmentation strategy TLAT. We show that models trained with TLAT have a better accuracy on clean samples than the models trained with a standard adversarial training algorithm. The idea of TLAT is to interpolate the target labels of adversarial examples with the ground-truth labels. This operation is computationally negligible and can be used in any trainings with target adversarial examples in order to improve the clean accuracy.

In future works, we would like to replace FGSM by an iterative adversarial attack in our algorithm and observe how this would influence the clean accuracy and the adversarial robustness of the models trained with M-TLAT. It would be also interesting to replace Mixup by a different interpolation strategy such as Manifold Mixup [49], and test if it further improves the performances of the algorithm.

References

1. Alexey Kurakin, Ian J. Goodfellow, S.B.: Adversarial examples in the physical world. In: International Conference on Learning Representations (2017)
2. Archambault, G.P., Mao, Y., Guo, H., Zhang, R.: Mixup as directional adversarial training. arXiv preprint arXiv:1906.06875 (2019)
3. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning. pp. 274–283. PMLR (2018)
4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57 (May 2017)
5. Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On evaluating adversarial robustness. arXiv preprint arXiv:1902.06705 (2019)
6. Dhillon, G.S., Azizzadenesheli, K., Bernstein, J.D., Kossaifi, J., Khanna, A., Lipton, Z.C., Anandkumar, A.: Stochastic activation pruning for robust adversarial defense. In: International Conference on Learning Representations (2018)
7. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. Eighth International Conference on Quality of Multimedia Experience (QoMEX) (2016)
8. Dodge, S.F., Karam, L.J.: Quality resilient deep neural networks. CoRR **abs/1703.08119** (2017)
9. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
10. Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A.: Exploring the landscape of spatial robustness. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 1802–1811. PMLR (2019)
11. Engstrom, L., Tsipras, D., Schmidt, L., Madry, A.: A rotation and a translation suffice: Fooling cnns with simple transformations. ArXiv **abs/1712.02779** (2017)
12. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2414–2423 (June 2016)
13. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019)
14. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. In: Advances in Neural Information Processing Systems 31, pp. 7538–7550. Curran Associates, Inc. (2018)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. CoRR **abs/1412.6572** (2015)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (June 2016)
17. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019)

18. Hendrycks*, D., Mu*, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple method to improve robustness and uncertainty under data shift. In: International Conference on Learning Representations (2020)
19. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: Advances in Neural Information Processing Systems 32. Curran Associates, Inc. (2019)
20. Inoue, H.: Data augmentation by pairing samples for images classification. arXiv preprint arXiv:1801.02929 (2018)
21. Kang, D., Sun, Y., Hendrycks, D., Brown, T., Steinhardt, J.: Testing robustness against unforeseen adversaries. arXiv preprint arXiv:1908.08016 (2019)
22. Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. arXiv preprint arXiv:1803.06373 (2018)
23. Karahan, S., Kilinc Yildirum, M., Kirtac, K., Rende, F.S., Butun, G., Ekenel, H.K.: How image degradations affect deep cnn-based face recognition? In: 2016 International Conference of the Biometrics Special Interest Group (BIOSIG). pp. 1–5 (Sep 2016)
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR (2015)
25. Koziarski, M., Cyganek, B.: Image recognition with deep neural networks in presence of noise dealing with and taking advantage of distortions. Integrated Computer-Aided Engineering **24**, 1–13 (08 2017)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. pp. 1097–1105. NIPS'12, Curran Associates Inc., USA (2012)
27. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. In: International Conference on Learning Representations (2017)
28. Lamb, A., Verma, V., Kannala, J., Bengio, Y.: Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. p. 95103 (2019)
29. Laugros, A., Caplier, A., Ospici, M.: Are adversarial robustness and common perturbation robustness independant attributes ? In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
30. Liu, A., Liu, X., Zhang, C., Yu, H., Liu, Q., He, J.: Training robust deep neural networks via adversarial noise propagation. arXiv preprintarXiv:1909.09034 (2019)
31. Liu, D., Cheng, B., Wang, Z., Zhang, H., Huang, T.S.: Enhance visual recognition under adverse conditions via deep networks. IEEE Transactions on Image Processing **28**(9), 4401–4412 (2019)
32. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3730–3738 (Dec 2015)
33. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
34. Orhan, A.E.: Robustness properties of facebook's resnext WSL models. CoRR **abs/1907.07640** (2019)
35. Pang*, T., Xu*, K., Zhu, J.: Mixup inference: Better exploiting mixup to defend adversarial attacks. In: International Conference on Learning Representations (2020)

36. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP) (2016)
37. Papernot, N., McDaniel, P.D., Goodfellow, I.J., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: ACM on Asia Conference on Computer and Communications Security (2017)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (06 2015)
39. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In: International Conference on Learning Representations (2018)
40. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially robust generalization requires more data. In: Advances in Neural Information Processing Systems 31, pp. 5014–5026. Curran Associates, Inc. (2018)
41. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
42. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014)
43. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1701–1708 (June 2014)
44. Tokozume, Y., Ushiku, Y., Harada, T.: Between-class learning for image classification. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5486–5494 (June 2018)
45. Tramer, F., Boneh, D.: Adversarial training and robustness for multiple perturbations. In: Advances in Neural Information Processing Systems 32. Curran Associates, Inc. (2019)
46. Tramr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. In: International Conference on Learning Representations (2018)
47. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (2019)
48. Vasiljevic, I., Chakrabarti, A., Shakhnarovich, G.: Examining the impact of blur on recognition by convolutional networks. arXiv preprintarXiv:1611.05760 (2016)
49. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: Proceedings of the 36th International Conference on Machine Learning. pp. 6438–6447. PMLR (2019)
50. Wang, J., Zhang, H.: Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
51. Wang, S., Wang, X., Zhao, P., Wen, W., Kaeli, D., Chin, P., Lin, X.: Defensive dropout for hardening deep neural networks under adversarial attacks. In: 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–8 (2018)
52. Xie, Q., Hovy, E.H., Luong, M.T., Le, Q.V.: Self-training with noisy student improves imagenet classification. ArXiv **abs/1911.04252** (2019)

53. Xie, S., Girshick, R., Dollr, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
54. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: 25th Annual Network and Distributed System Security Symposium (2018)
55. Yin, D., Lopes, R.G., Shlens, J., Cubuk, E.D., Gilmer, J.: A fourier perspective on model robustness in computer vision. ICML Workshop on Uncertainty and Robustness in Deep Learning (2019)
56. Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L.E., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: Proceedings of the 36th International Conference on Machine Learning. pp. 7472–7482 (2019)
57. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
58. Zhou, Y., Song, S., Cheung, N.: On classification of distorted images with deep convolutional neural networks. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1213–1217 (March 2017)