Mixture of techniques to improve classification accuracy and robustness with adversarial training.

Martin Dallaire
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montreal, QC H3C 3J7
martin.dallaire@umontreal.ca

François David
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montreal, QC H3C 3J7
francois.david@umontreal.ca

Abstract

"Adversarial examples" is a very active field of research, with numerous new papers constantly being made on the topic, each offering a variety of different techniques or insights. In this paper, we aim to analyse recent techniques instead of suggesting new ones. Specifically, we focus on the three following techniques: Adversarial Propagation (AdvProp), Adversarial Label Smoothing (ALS) and Smooth Adversarial Training (SAT). Examining through an ablation study, we compare the effects of each techniques individually as well as concurrently. We find that AdvProp is useful if one wants to improve their accuracy on real images ($+1.29\%$ accuracy), while still gaining a respectable increase in robustness (between $+6.79\%$ and $+15.19\%$ depending on the attacker) compared to standard non-adversarial training. Meanwhile, compared to non-adversarial training, ALS offers a slight boost in robustness for strong attackers ($+3.21\%$), but comes at a cost in terms of accuracy (-1.26%). Finally, SAT seems negligibly better than standard adversarial training on strong attackers ($+3.67\%$), but it is not significant, as it is slightly worse on weaker attackers, as well as slightly worse on real images (-0.56%). Although ALS + SAT had the best robustness score amongst models trained on weak attackers when faced with the strongest attacker, we find that the combinations of techniques does not seem to bring added value. Thus, we find that AdvProp is recommended for certain contexts, while the other techniques are not as useful.

Introduction

For many supervised tasks, Deep Neural Networks have proven to be better than humans on real-world data. However, there exists a relatively easy way to fool a model: adversarial examples.

Adversarial examples are small perturbations to the input of a neural network in order to fool the classifier into predicting the wrong class, even if the resulting perturbations are often imperceptible to humans. These adversarial examples can prove to be a serious threat to machine learning models which are used for security-critical tasks Zhang and Li [2019]. It is thus important to train so-called robust models, which are able to reduce the success rate of such attacks. The most common method to defend against adversarial attacks is to train the network with generated adversarial examples simultaneously with the original inputs, significantly improving the robustness of the model as a result. However, when implemented naively, this type of data augmentation can reduce the accuracy of the classifier on clean data (Raghunathan et al. [2019]). This creates a trade-off between accuracy and robustness. Other, simple methods also exist to improve robustness, such as label-smoothing.

In this project, we experiment with 3 techniques that aim at improving the accuracy and/or robustness of classifiers, without incurring a loss of accuracy on clean images. The three techniques we are interested in are Adversarial Label-Smoothing (ALS) (Goibert and Dohmatob [2019]), Smooth Adversarial Training (SAT) (Xie et al. [2020]) and AdvProp (Xie et al. [2019]). The goal is to see how the different methods compare, and if there is a benefit to applying them together. Ablation studies were made and concluded that the Advprop technique significantly improved the accuracy of our classifier on original samples, while also slightly improving robustness when compared to traditional training without adversaries. The SAT technique only marginally improves robustness against strong attackers when compared to standard adversarial training, while decreasing the accuracy on real images. Moreover, the ALS technique resulted in slightly improve robustness compared to the traditional training but significantly decreased the accuracy. After trying out all combination, we concluded that the AdvProp technique alone was the best compromise, as it improved adversarial robustness while also improving the standard accuracy of the model, at the cost of extra computational time. Combinations of techniques did not particularly offer improvements over either using AdvProp alone or standard adversarial training alone.

Related Works

Adversarial examples are small carefully designed perturbations to the input of classifier in order to fool the model into predicting the wrong class Goodfellow et al. [2015]. In order to generate adversarial examples, one can run the backward pass of a network to get the gradients with respect to the input and make small perturbation accordingly. This way, one can optimize the input of the algorithm in order to fool the model into outputting the wrong classification when passed to the network. Notably, theses adversarial examples can often end up looking exactly the same as the original samples to humans, depending on the allowed size of the perturbations, and the strength of the attacker.

Two common techniques of adversarial attacks are the Fast Gradient Sign Method (FGSM) and the Projected Gradient Descent (PGD) Madry et al. [2019]. Both techniques are based on the idea of maximizing the loss with respect to the input. The FGSM method computes the adversary as :

$$x + \alpha \text{sign}(\nabla_x L(\theta, x, y))$$

It can be view as a one-step scheme, since we only take the sign of the gradient and make perturbation for all elements in the sample. On the other hand, the PGD techniques computes the adversary as :

$$x_{t+1} = \Pi_{x+S}(x_t + \alpha \text{sign}(\nabla_x L(\theta, x_t, y)))$$

Where the adversarial example is created based on multiple iterations. We note here that the pixels of the resulting image must all be at most ϵ-close to the original pixels, hence the projection in the formula. Furthermore, this attack has an additional parameter α, which can influence results significantly.

One way to prevent a model from being vulnerable to adversarial attacks is to do some adversarial training. Adversarial training was initially proposed by Goodfellow et al. [2015]. Goodfellow proposed that we train the neural network with adversially-generated images, using the formulas

above to generate such samples. Training a model in such a way reduces the effectiveness of similar attacks. However, doing this standard adversarial training have shown to sometimes affect the performance of the network on the original dataset. Raghunathan et al. [2019].

Theory

AdvProp

In 2019, a new technique, named AdvProp (short for Adversarial Propagation), was proprosed to improve the classification accuracy of DNN classifiers with the use of adversarial examples Xie et al. [2019]. In their paper, the authors theorize that the original and the adversarial samples have different data distributions. Furthermore, they suggest that using the same batch normalization statistics for both type of images might be the cause of the sometimes-observerd drop in prediction accuracy, since the statistics of the two distributions are different. As such, the authors of this technique propose that using different batch normalization statistics should disentangle the mixture of distribution and improve accuracy, Xie et al. [2019]. As shown in the illustration below, this technique uses an auxiliary batch normalization layer for the adversarial examples. This avoids mixing the batch normalization statistics. Figure 1 illustrates this idea.

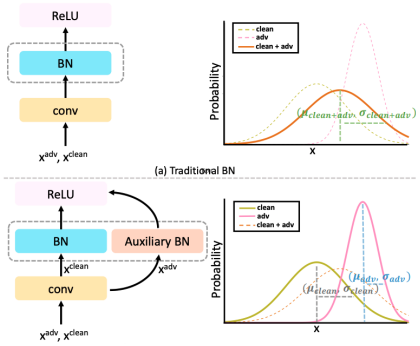

Figure 1: Representation of AdvProp Xie et al. [2019]

During training, this method uses the auxiliary batch normalization layer when processing adversarial examples, while using the usual batch normalization layer when processing real images. At testing time, we use the normal batch normalization layer exclusively, as it is impossible to know in advance which layer to use. This was not a problem for the authors of the paper, as their aim was simply to leverage adversarial examples with the goal of improving accuracy. In general, the original paper of AdvProp argues that the technique improves a classifier's accuracy by essentially augmenting the data set with adversarial examples, while disentangling their distribution. This technique is also said to perform better on larger models, Xie et al. [2019].

However, for our purposes here, we theorize that the AdvProp method will still improve robustness, even when using the wrong batch normalization layer when testing on adversarial examples. Our hypothesis is that, since the models were trained with adversarial examples, it should in fact still be more robust than a traditional training regime which does not see any adversarially-generated images. This is something which the original authors seemingly did not test.

Adversarial Label Smoothing

Adversarial label-smoothing is a technique that was proposed in 2019, it is a form of relatively free adversarial training Goibert and Dohmatob [2019]. It is like a targeted form of label-smoothing that specifically aims at improving the robustness of a DNN classifiers. Traditional label-smoothing is a technique that involves "smoothing" the labels by adjusting them slightly away from the boundaries 0 and 1. Adversarial label-smoothing, on the other hand, finds the category with the lowest prediction score and updates the true label vector at that position with α instead of zero. Then, we replace the true label's correct entry with $(1 - \alpha)$. This defense technique results, in theory, in a less confident model, which could make it harder to find perfect adversarial examples. However, Fu et al. [2020] warns that label-smoothing as a form of adversarial training is volatile and might not defend against

attacks using a naturally trained model. Still, the biggest advantage of this approach is that the cost of implementation is minuscule, since it is only a slight modification of the labels based on the already-calculated probability vector output.

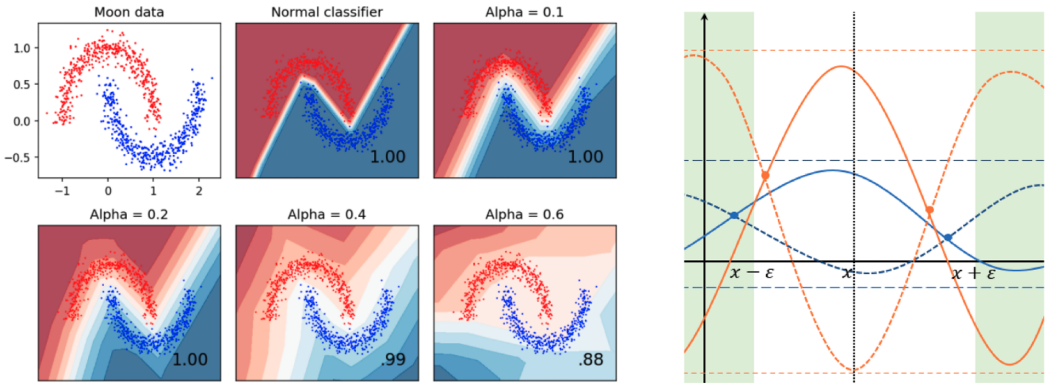

Figure 2: **Left**, representation of the effect of ALS on the decision boundary Goibert and Dohmatob [2019]. **Right**, representation of the effect of Label Smoothing on decision boundaries Fu et al. [2020]. Blue curve is from a model trained with Label Smoothing, yellow curve is from a model trained without.

The figure above is a good illustration of how label smoothing can improve robustness of a model. Since a model with label smoothing is less confident in its choices, its decision boundary points (in blue) are further apart which reduces the number of adversarial points within the permissible range of the attack Fu et al. [2020]. In some situations, it could lead to the inability to find adversarial examples easily, thus reducing the number of possible adversarial attacks.

Smooth Adversarial Training

Another technique, Smooth Adversarial Training (SAT), was proposed in 2020. It consists of using a smooth approximation of the ReLU activation function for the backward passes of the network Xie et al. [2020]. The function they propose is the Parametric Softplus function. The goal of this change is to keep good gradient information during the backward pass, Xie et al. [2020]. They explain that the non-smoothness of the widely used ReLU activation function leads to less useful gradient information near zero, which yields non-optimal adversarial examples. One thing to keep in consideration is that during the forward pass, ReLU is used as usual, and the replacement is only used during the backprogagation. Since the gradient is smoother, the hope is that we can generate better, harder adversarial examples during the adversarial training process, which should further improve the robustness of the model.

$$\textbf{Forward Pass: } f(\alpha, x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$\textbf{Backward Pass: } f(\alpha, x) = \frac{1}{1 + exp(-\alpha x)} \tag{2}$$

Using the standard ReLU function in the forward pass and using of the parametric softplus derivative for the backward pass - A smooth approximation of ReLU.

This technique is said by the authors to induce no drop in accuracy on clean data compared to traditional adversarial training. Moreover, the paper presenting this technique have made ablations studies which denoted that using the Parametric Softplus activation function for the parameter updates of the network as well, generated the best results. The more the parameter α is larger, the more the backward pass looks like the ReLU activation function. The authors of the original paper stated that they used a value of α = 10 Xie et al. [2020].

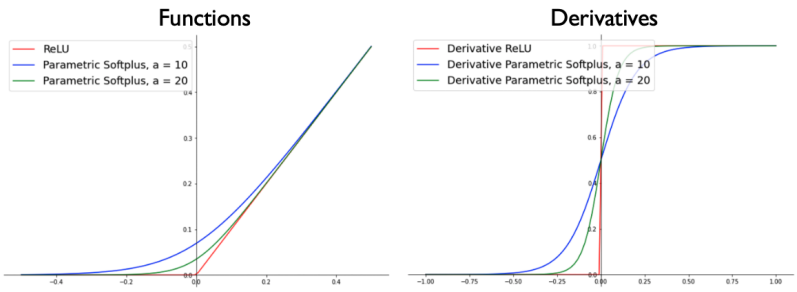

Figure 3: Here are graphical reppresentation of the forward pass and negative passes of the ReLU and Parametric softplus with different α values. The parametric softplus is a smooth approximation of the ReLU activation function.

Combination Hypotheses

In this study, we hoped that combining those methods will lead to architectures and training methods that will combine the upsides of increased robustness and/or increase accuracy on original images even further. In overview, the advprop technique is supposed to increase the accuracy of DNN models, while the other two techniques aims to increase the adversarial robustness. Hopefully, the combination of Smooth Adversarial Training (SAT) and Adversarial Label Smoothing will make a classifier even more robust to adversarial examples. Also, since the SAT technique should generate better adversarial examples, maybe if we use this technique with advprop, the accuracy will increase even further. Thorough ablation studies were made to identify the most promising techniques.

Methodology

Below is a pseudocode of the three techniques combined. If the advprop boolean is passed to the model, it uses different batch normalization layers. We can generate FGSM and PGD adversarial attacks. Multiple parameters can be used in those attacks (learning rate of attacker and allowed perturbance to the original samples). In order to train our model, we used a PGD attack with two iterations so computational reasons

Algorithm 1 Training Loop for the Combination of Techniques

Input: Training images $X_i$and labels y_i.
for each epoch **do**
  for each mini-batch $x = x_{1:m}, y = y_{1:m}$ **do**
    Get normal predictions: $y_{pred} \leftarrow model(x)$
    if Adversarial Training **then**
      if FGSM **then**
        $X_{Attack} \leftarrow$ FGSM_Attack(model, images, labels, α, ϵ)
      else if PGD **then**
        $X_{Attack} \leftarrow$PGD_Attack(model, images, labels, α, ϵ, iters=2)
      Get adversarial predictions: $y_{predAdv} \leftarrow model(X_{Attack}, advProp)$
      Concatenate preds: $y_{pred} \leftarrow (y_{pred}, y_{predAdv})$
      Duplicate true labels: $y \leftarrow (y, y)$
    if ALS **then**
      Compute Loss : SmoothCE(y_{pred}, y)
    else
      Compute Loss : Cross_Entropy(y_{pred}, y)
    Update: model parameters Θ with backprop and loss computed.

In the testing phase, we test the robustness of the model on the test set on FGSM and PGD attack with varying number of allowed perturbance to allow a clear analysis of the results. One thing to take into consideration is that, the original technique of the Smooth Adversarial Training only trains on on adverarial examples :

$$\arg\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim \mathbb{D}} [\max_{\epsilon \in \mathbb{S}} L(\theta + \epsilon, x, y)] \tag{3}$$

However, in this study (in order to combine the methods) we trained this technique with also original samples so that the loss function is compatible with the other techniques.

$$\arg\min_{\theta} \left[\mathop{\mathbb{E}}_{(x,y)\sim \mathbb{D}} [L(\theta, x, y) + \max_{\epsilon \in \mathbb{S}} L(\theta + \epsilon, x, y)] \right] \tag{4}$$

We assume that the same benefits of having better gradients information should translate to same benefits on this loss function since the authors mentioned that the use of SAT for the parameter update of the network also benefited from this smooth non-linearity Xie et al. [2020].

Experiments

Experiment setup

For our experiments, we focused on the CIFAR-10 dataset, as our available computing resources are limited to what Google Colab offers. Furthermore, since the purpose of our project was to analyse the effects of combining multiple techniques together, it required running multiple experiments, accounting for all the different combinations of techniques.

Our architecture was a ResNet18 architecture, slightly modified to work with CIFAR-10. We normalized the data so that the pixel values would be between 0 and 1. Furthermore, we used standard image augmentation technique. This fact will prove to be relevant when we analyse the results below.

We trained each algorithms to convergence, using 100 epochs and a decreasing learning rate. For methods which require adversarial examples to be generated, we first used 2-step PGD attacks with $\epsilon = 0.1$, and we tried two vastly different step sizes for the attacker. We also tried the normal methods with an attacking ϵ of 0.5, for the sake of comparison. We note that while, in theory, an attacker using an ϵ of 0.5 could turn any normalized image into an uniformly grey picture (as mentioned in Madry et al. [2019]), this is not the case in practice, with our 2-step PGD attacker.

Experimental results - stand-alone techniques

As can be seen in Table 1, ALS seems to provide slightly better protection against attackers compared to traditional non-adversarial training, and the added computational cost is minuscule. However, this comes with a drop in accuracy of 1%.

SAT, on the other hand, does not compare favorably to standard adversarial training, neither on weak attackers nor on standard examples. However, there is a slight, additional protection against stronger attackers. Further experiments would be required to confirm this advantage.

As for AdvProp, our small experiments confirm the results of the original paper. They originally tested on ImageNet, with multiple depths of architecture, and they observed an increase in accuracy from the use of adversarial examples along with separate batch norms. Our results show the same phenomenon: methods which use separate batch norms show an increased in accuracy compared to traditional training, at least up to a certain level of difficulty with regards to the adversarial examples. Empirically, we found that methods using separate batch norms benefit greatly from weaker adversarial examples, both in terms of accuracy and in terms of robustness. We can conclude that the adversarial examples do have a disctinct data distribution compared to the original samples.

Furthermore, the original paper describing the AdvProp technique was only interested in improving test accuracy. We wanted to see if the method also provided some robustness, and our results seem to show that it does. While less robust than standard adversarial training, AdvProp is more robust than no adversarial training at all. As a bonus, it seems to significantly improve testing accuracy compared to standard adversarial training.

Description	Test Accuracy	FGSM 0.1-ϵ	PGD 0.1-ϵ $\alpha = 2/255$	FGSM 0.5-ϵ	PGD 0.5-ϵ $\alpha = 0.5$
Traditional	91.87	47.18	40.85	42.51	11.28
Adversarial	92.11	**75.92**	**79.84**	**60.60**	28.40
AdvProp	**93.16**	59.46	56.04	50.91	18.07
ALS	90.61	48.51	41.65	41.44	14.49
SAT	91.55	75.53	78.23	**60.63**	32.07
ALS + SAT	91.73	74.87	78.04	59.23	**33.18**
SAT + AdvProp	**93.15**	58.59	52.74	49.90	19.48
ALS + AdvProp	**93.14**	59.14	56.23	49.16	19.90
ALS + SAT + AdvProp	92.38	56.96	49.79	47.46	18.32

Table 1: Results on CIFAR-10 for our methods. Methods that require adversarial examples are trained using a 2-step, 0.1-PGD attacker trained with a very small step-size. Hence, they do not reach the epsilon at all, in practice. We found that using such a small step size actually provided big benefits to methods such as AdvProps.

Description	Test Accuracy	FGSM 0.1-ϵ	PGD 0.1-ϵ $\alpha = 2/255$	FGSM 0.5-ϵ	PGD 0.5-ϵ $\alpha = 0.5$
SAT	90.79	61.61	41.45	60.27	**85.21**
AdvProp	91.52	54.46	49.20	45.47	10.35
Adversarial	90.97	65.81	45.39	**64.54**	**81.19**

Table 2: The three methods, trained with 0.5-PGD examples, two step-sizes and step-size 0.5. In bold: improvements in robustness compared to the results of figure 1

Experimental results - combination of techniques

Figure 4 shows how each methods fare against each other in our experiments. We can see that, in general, methods that include separate batch norms (AdvProp and its combinations) improve accuracy on real images, while slightly, but noticeably improving robustness. Combining AdvProp with other methods does not seem to change the results much. The use of AdvProp influences the results in a dominant fashion.

Meanwhile, SAT seems to not offer any significant advantage in robustness compared to standard adversarial training on small perturbation attacks, while coming at a slight cost with regards to the testing accuracy on real images. However, there is a slight increase in robustness for large perturbation attacks and Combining with ALS did help robustness for the PGD 0.5 $\epsilon = \alpha = 0.5$.

In terms of accuracy, combining all three techniques gives us results that are strictly worse than simply using AdvProp by itself.

Discussion

We find that AdvProp is useful if one wants to improve their accuracy on real images ($+1.29\%$ accuracy), while still gaining a respectable increase in robustness (between $+6.79\%$ and $+15.19\%$ depending on the attacker) compared to standard non-adversarial training. Meanwhile, compared to non-adversarial training, ALS offers a slight boost in robustness for strong attackers ($+3.21\%$), but comes at a cost in terms of accuracy (-1.26%). Furthermore, SAT seems negligibly better than standard adversarial training on strong attackers ($+3.67\%$), but it is not significant, as it is slightly worse on weaker attackers, as well as significantly worse on real images (-0.56%). Finally, although ALS + SAT had the best robustness score amongst models trained on weak attackers when faced with strong attackers, we find that the combinations of techniques does not seem to bring added value. Thus, we find that AdvProp is recommended for certain contexts, while the other techniques are not as generally useful.

Contrary to our initial hopes, combining the three techniques does not seem to improve results. Multiple factors can explain these results. The first, and, likely, biggest explanation of these results

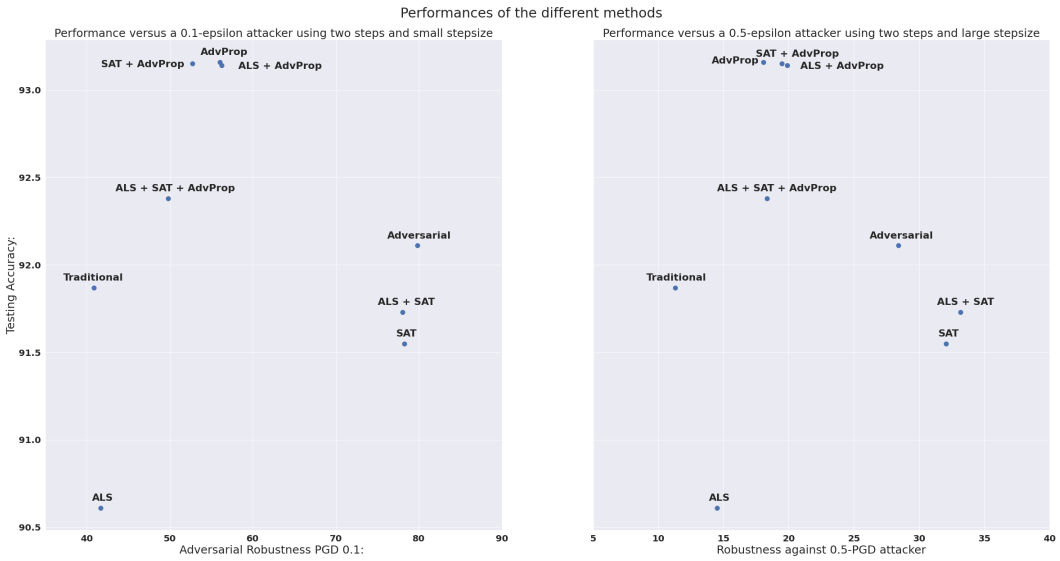

Figure 4: Test accuracy vs robustness of different methods, when compared to two different attackers.

is the fact that we did not observe much advantage in using the ALS or the SAT techniques. While AdvProp proved to be exactly as useful for improving test accuracy as the original paper suggested (with the added bonus of improving robustness, as we theorized), we did not observe the stated advantages of the other methods. Our hypothesis is that the methods are more context-sensitive, and more hyper-parameter tuning would be needed. Another possibility is that the papers only tested the techniques against unreasonably strong opponents, for which we did see a slight increase in robustness. More experiments with with attackers of varying levels of strength should be done in order to confirm the existence of this effect. Due to our limited computing power, we were not able to test this as extensively.

As a summary, we find that AdvProp is very useful and relatively easy to implement. ALS is even easier, but the advantages are less clear, and standard label smoothing is probably good enough. SAT, on the other hand, is more complicated to implement and adds more hyperparameters, which makes it harder to recommend.

Conclusion

We have experimented with a few techniques to improve the accuracy and/or the robustness of a DNN. In our experiments, the Smooth Adversarial Training technique showed some promises for the attacks with large pertubations, but did affect the accuracy (-0.56% compared to standard adversarial training). and the combination with Adversarial Label Smoothing achieved the best results of all models trained with weak adversarial examples against strong PGD attacks using $\epsilon = \alpha = 0.5$. In general, the ALS technique showed increased robustness compared to traditional training, but at the cost of -1.26% of accuracy. The AdvProp technique significantly improved the accuracy of the classifier while also providing considerably better robustness than the traditional training, something that was not tested in its original paper. Unfortunately, no mixture of the techniques worked better than simply AdvProp alone for the accuracy, and we recommend using standard adversarial training for robustness. More experiments on different datasets and with different attackers should be done to confirm the results and explore the effectiveness of the techniques on strong attacks.

References

C. Fu, H. Chen, N. Ruan, and W. Jia. Label smoothing and adversarial robustness, 2020.

M. Goibert and E. Dohmatob. Adversarial robustness via label-smoothing, 2019.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2015.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Adversarial training can hurt generalization. *CoRR*, abs/1906.06032, 2019. URL `http://arxiv.org/abs/1906.06032`.

C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. Adversarial examples improve image recognition. *CoRR*, abs/1911.09665, 2019. URL `http://arxiv.org/abs/1911.09665`.

C. Xie, M. Tan, B. Gong, A. Yuille, and Q. V. Le. Smooth adversarial training, 2020.

J. Zhang and C. Li. Adversarial examples: Opportunities and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–16, 2019. ISSN 2162-2388. doi: 10.1109/tnnls.2019.2933524. URL `http://dx.doi.org/10.1109/TNNLS.2019.2933524`.