Training Universal Adversarial Perturbations with Alternating Loss Functions

Abstract

Despite being very successful, deep learning models were shown to be vulnerable to crafted perturbations. Furthermore, changing the prediction of a network over any image by learning a single universal adversarial perturbation (UAP) was shown to be possible. In this work, we propose 3 different ways of training UAPs that can attain a predefined fooling rate, while, in association, optimizing L_2 or L_∞ norms. To stabilize around a predefined fooling rate, we have integrated an alternating loss function scheme that changes the current loss function based on a given condition. In particular, the loss functions we propose are: Batch Alternating Loss, Epoch-Batch Alternating Loss and Progressive Alternating Loss. In addition, we empirically observed that UAPs that were learned by minimization attacks contain strong image-like features around the edges, hence we propose integrating a circular masking operation to the training to further alleviate visible perturbations. The proposed L_2 Progressive Alternating Loss method outperforms the popular attacks by providing a higher fooling rate at equal L_2 norms. Furthermore Filtered Progressive Alternating Loss can further reduce the L_2 norm by 33.3% at the same fooling rate. When optimized with regards to L_∞, Progressive Alternating Loss manages to stabilize on the desired fooling rate of 95% with only 1 percentage point of deviation, despite L_∞ norm being particularly sensitive to small updates.

Introduction

Deep learning models have been adopted as standard methods in many visual tasks due to their success. On the other hand, deep neural networks have also been shown to be vulnerable against purposefully generated data samples called adversarial examples. The most popular way of generating adversarial examples is applying an adversarial attack to a benign sample, and obtain a particular perturbation that leads to misclassification when added to this benign sample. With this method, generating a whole dataset of adversarial examples involve putting each image through the same algorithm to calculate an image dependent perturbation, which results in a significant time overhead. Recently, it has been shown that a single perturbation, can be used to make any sample an adversarial example; these perturbations are called universal adversarial perturbations (UAP).

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Table 1: Overview of the proposed UAP training methods

Attack	**Abbreviation**	**Loss Alteration Condition**
Batch Alternating Loss	B-AL	Fooling rate of each batch
Epoch-Batch Alternating Loss	EB-AL	Fooling rate of each batch, if the previous epoch reached the fooling rate
Progressive Alternating Loss	P-AL	Fooling rate up to the point of processing the current batch
Filtered Progressive Alternating Loss	FP-AL	Same as P-AL but after each batch the filter in Equation 3 is applied on the UAP

These types of perturbations have distinct properties compared to image dependent adversarial perturbations, such as having image-like features by themselves (Zhang et al. 2020), whereas traditional perturbations are perceived as noise by humans.

Related Work

Adversarial perturbations are traditionally generated specifically for a single sample. Fast gradient sign method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) is an adversarial attack which can be used with L_1, L_2 and L_∞ norms, and despite its simplicity, it is still being widely used. Basic iterative method (Kurakin, Goodfellow, and Bengio 2017) and projected gradient descent (Madry et al. 2018) algorithms; as opposed to FGSM, iteratively optimize the perturbation with fixed size steps. Different to these L_p bounded attacks, there are also minimization attacks. DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) aims to geometrically shift the benign image to the closest decision boundary to force misclassification. Carlini&Wagner attack (Carlini and Wagner 2017) reformulates a constrained optimization problem to generate the smallest successful adversarial perturbation. Perceptual Color distance Alternating Loss (Zhao, Liu,

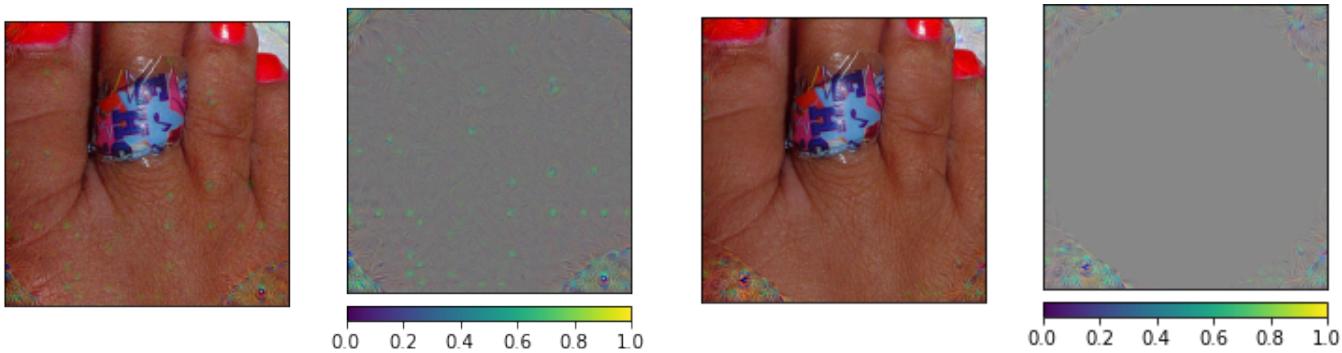

Figure 1: Sample UAPs trained with L_2 P-AL (left) and Filtered P-AL (right). The prediction for benign image is the correct class, *band aid*, with 98.70% confidence. The adversarial examples yield *peacock* predictions with 99.90% and 99.97% confidence respectively. The predictions are from ResNet50.

and Larson 2020) is a modified version of Carlini&Wagner that decouples the norm and adversarial optimization using alternating loss method, which is also adopted in our proposed algorithms.

Universal adversarial perturbations were formally introduced in (Moosavi-Dezfooli et al. 2017), which applies DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) algorithm to each sample iteratively, updates the overall universal perturbation and projects the perturbation to a L_p ball. Generative models were also trained to obtain UAPs. Network for adversary generation (NAG) (Mopuri et al. 2018), is a generative adversarial network framework that trains a generator, using a freezed target classification network, to generate a UAP, from an input noise vector. On the other hand, Fast Feature Fool (Mopuri, Garg, and Babu 2017) is a data-free algorithm, that trains a UAP that maximizes the activation values of convolutional layers. This algorithm generally performs worse than data dependent attacks, but is a good proof that UAPs can be generated by only using the properties of the target convolutional network. Feature-UAP (Zhang et al. 2020) is a L_p constrained attack that trains a UAP using mini-batch training to achieve state of the art fooling rates and the authors provide a detailed comparison between image-dependent attacks and universal attacks. High-Pass-UAP (Zhang et al. 2021) is a similar algorithm that also trains UAPs using mini-batches, but also applies a Fourier domain high-pass filter to the current UAP, after revealing that UAPs tend to perform better when they contain more high frequency features, while being imperceptible to human eye. In the same work, Universal Secret Adversarial Perturbation (Zhang et al. 2021) was introduced, where a UAP not only fools models, but also contains extractable information. Training UAPs to make a network to perceive a predefined class as another target class, was introduced and named 'Double Targeted UAPs' (Benz et al. 2020).

In this paper, we propose 3 alternative approaches using alternating loss for training UAPs: Batch Alternating Loss (B-AL), Epoch-Batch Alternating Loss (EB-AL) and Progressive Alternating Loss Training (P-AL). All the universal attacks in the literature are norm bounded; thus the norm optimization is not a stochastic operation, rather it is a projection. Our method is different from these works in this regard. In addition, we propose integrating a filtering to the training to further reduce the perturbations at the same fooling levels.

Algorithm 1: Batch Alternating Loss Training(B-AL)

Input: Dataset μ, target class t, fooling rate δ, epoch k, model f, norm p
Variables: Counter i, fooling rate fr, prediction out, adversarial loss function adv, loss L
Output: Universal adversarial perturbation v

```
v ← 0
i ← 0
while i < k do
    for x ~ μ do
        out ← f(x + v)
        fr ← # of correct predictions / batch size
        if fr < δ then
            L ← adv(out, t)
        else
            L ← ||v||_p
        end if
        backpropagate L
        update v
        i ← i + 1
    end for
end while
return v
```

Methodology

The universal adversarial attack problem can be formally defined as in Equation 1, where v is the UAP, x is a benign image sampled from a dataset μ, f is the target model, δ is the minimum fooling rate and ϵ is the maximum L_p norm of v.

$$P_{x\sim\mu}(f(x+v) \neq f(x)) \geq \delta \text{ s.t. } \quad \|\nu\|_p \leq \epsilon \qquad (1)$$

The norm bounded attack concept is widely used in adver-

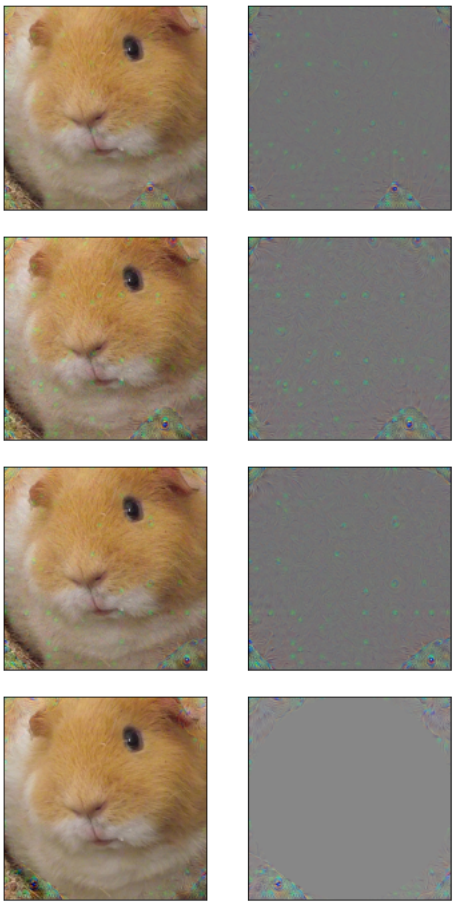

Figure 2: UAP calculated for *Peacock* target class applied over a hamster image (left column) and the UAP images (right column), corresponding to B-AL, EB-AL, P-AL, Filtered P-AL, from top to bottom.

sarial machine learning, however it is also possible to formulate it as a minimization problem as in Equation 2, by slight modifications over Equation 1.

$$min_{||v||_p} P_{x \sim \mu}(f(x+v) = t) \approx \delta \tag{2}$$

The variable t is the target class in this equation. Now, the problem becomes finding the smallest $||v||_p$, that attains the desired fooling rate. This problem can be turned into a min-max problem as well by setting δ to 1.

In this work, we propose a solution to this problem by introducing 3 UAP training methods (shown in Table 1) leading to different attacks that take advantage of the alternating loss strategy. Alternating loss scheme switches between 2 loss functions depending on the current state of the training; this strategy is used in image dependent adversarial attacks (optimize the norm of the perturbation if the current image is adversarial, if not, optimize the adversarial loss), however it is not directly applicable to the UAP domain. The first proposed method is Batch Alternating Loss (B-AL), which aims to reach the desired fooling rate by achieving the same fooling rate for each batch. The second method is Epoch-Batch Alternating Loss (EB-AL), which also takes into account the fooling rate achieved over the epoch, alongside with each individual batch. The final training method is Progressive Alternating Loss (P-AL), which uses the fooling rate achieved until the current batch to alter the loss function. We also empirically find that stronger features are generated around the edges, along with smaller artifacts in the middle; therefore we propose applying filtering during training to alleviate these artifacts. The proposed filtering scheme can be integrated into any minimization based UAP training.

Alternating loss scheme requires a decoupled decision mechanism that will set the current loss function to either adversarial loss which changes the prediction of the network, or the norm of the UAP which is either L_2 or L_∞. In image-dependent attacks, alternating loss function can be iteratively selected based on the current state of the perturbation; when the current perturbation is successful in making the image an adversarial example, minimize the norm of the perturbation, else, optimize the adversarial loss to obtain an adversarial example (Zhao, Liu, and Larson 2020). However, training a UAP for several iterations for a single image while changing the loss function at each iteration would be incompatible with mini-batch training. We can instead use batches to train the UAP, while changing the current loss function based on the fooling rate achieved on that batch. Note that the main parameter in this optimization is the desired fooling rate over the whole training dataset.

Batch Alternating Loss (B-AL)

Algorithm 1 shows the pseudo-code of B-AL. In this approach, the loss function is switched according to the fooling rate performance of the current state of the UAP, over the current batch: if UAP can achieve the desired fooling rate over the current batch, norm loss is selected, otherwise, adversarial loss is selected, which is chosen to be the cross entropy function. This training method can bring the fooling rate around the desired level in several epochs. However, when some of the batches yield the desired fooling rate, the loss function lowers the adversarial energy to decrease the norm of the UAP, thus the overall fooling rate stays below the target fooling rate, since the majority of the images remain benign. To address this problem, the following 2 methods of training are proposed.

Epoch-Batch Alternating Loss (EB-AL)

Algorithm 2 shows the pseudo-code of EB-AL. This method aims to ensure that the UAP does not start diminishing the adversarial energy until the desired fooling rate is achieved. Before reaching the target, the loss function strictly becomes adversarial, regardless of the individual performance of each batch. At the end of each epoch, we check whether the target fooling rate was achieved, if it was, then in the next epoch, the same loss function alteration scheme presented in B-AL training is applied. EB-AL almost always achieves the desired fooling rate, if it is possible at all. When an epoch is completed with a successful fooling rate, many of the batches may yield fooling rates above the target. This causes the extensive usage of the norm loss function, which

Algorithm 2: Epoch-Batch Alternating Loss Training (EB-AL)

Input: Dataset μ, target class t, fooling rate δ, epoch k, model f, norm p
Variables: Counter i, fooling rate fr, prediction out, adversarial loss function adv, loss L, optimization mode m, number of correct predictions $correct$, image number counter $imcount$, fooling rate over the epoch $epochfr$
Output: Universal adversarial perturbation v

$v \leftarrow 0$
$i \leftarrow 0$
$m \leftarrow$ 'epoch'
while $i < k$ **do**
  $correct \leftarrow 0$
  $imcount \leftarrow 0$
  for $x \sim \mu$ **do**
    $out \leftarrow f(x + v)$
    $correct \leftarrow correct+$ # of correct predictions
    $imcount \leftarrow imcount+$ batch size
    $fr \leftarrow$ # of correct predictions / batch size
    if (m == 'epoch') **or** (m == 'batch' and $fr < \delta$) **then**
      $L \leftarrow adv(out, t)$
    else
      $L \leftarrow ||v||_p$
    end if
    backpropagate L
    update v
    $i \leftarrow i + 1$
  end for
  $epochfr \leftarrow 1 - correct/imcount$
  if $epochfr < \delta$ **then**
    $m \leftarrow$ 'epoch'
  else
    $m \leftarrow$ 'batch'
  end if
end while
return v

brings the overall fooling rate down, which will then make the loss function strictly adversarial in the next epoch. This phenomenon makes the fooling rate oscillate around the target, which may cause some imprecision in attaining the target fool rate.

Progressive Alternating Loss (P-AL)

Algorithm 3 shows the pseudo-code of P-AL. Because of the nature of the training procedure, it is not trivial to have fooling rate completely stabilized over the target, however, it is possible to minimize the oscillation caused by the phenomena explained in the previous section. In P-AL training, similar to EB-AL, the adversarial loss is maintained until the target fooling rate is achieved. Furthermore, after reaching the target, the loss function is altered based on the fooling rate achieved from the beginning of the epoch until the currently optimized batch. By this way, it is possible to maintain the overall fooling rate, while optimizing the norm when possible. Although, this method cannot completely prevent the oscillation, it minimizes it to a certain degree.

Algorithm 3: Filtered Progressive Alternating Loss Training (FP-AL)

Input: Dataset μ, target class t, fooling rate δ, epoch k, model f, norm p, mask radius D
Variables: Counter i, fooling rate fr, prediction out, adversarial loss function adv, loss L, number of correct predictions $correct$, image number counter $imcount$, circlar filter $filter$
Output: Universal adversarial perturbation v

$v \leftarrow 0$
$i \leftarrow 0$
$filter \leftarrow$ filter in Equation 3 with D
while $i < k$ **do**
  $correct \leftarrow 0$
  $imcount \leftarrow 0$
  for $x \sim \mu$ **do**
    $out \leftarrow f(x + v)$
    $correct \leftarrow correct$ + # of correct predictions
    $imcount \leftarrow imcount$ + batch size
    $fr \leftarrow$ length of correct / $imcount$
    if $fr < \delta$ **then**
      $L \leftarrow adv(out, t)$
    else
      $L \leftarrow ||v||_p$
    end if
    backpropagate L
    update v
    $v \leftarrow filter(v)$
    $i \leftarrow i + 1$
  end for
end while
return v

Masked Training

We empirically found out that when the UAP is not normalized by a norm constraint, perturbations with high intensities tend to accumulate around the edges and corners. These perturbations also contain more image-like features, thus they influence the prediction towards the target. Applying masks to smooth out perturbations was investigated for image dependent attacks (Aksoy and Temizel 2020), and it is a simple yet efficient way to control the geometry of the perturbations, hence we propose integrating a masking operation in the UAP training procedure. To reduce the perturbations around the center of the image (which are also mostly positioned on top of the target object), we apply a filter (Equation 3) to the UAP after each batch, where (x, y) is the pixel position, h and w are the height-width values of UAP respectively, and D is the radius of the circle. Algorithm 3 shows FP-AL training, which is the P-AL training with filtering. The visual effect of filtered training can be seen in Figure 1.

Table 2: L_2 attack results, provided in terms of L_2 and L_∞ metrics and FR refers to the Fooling Rate. Note that UAP (Moosavi-Dezfooli et al. 2017) and F-UAP attacks (Zhang et al. 2020) are set to reach the same L_2 values as P-AL to allow comparison of FR at the same level of perturbation.

Method	DenseNet121			ResNet50			GoogleNet			VGG16		
	L2	L_∞	FR	L2	L_∞	FR	L2	L_∞	FR	L2	L_∞	FR
B-AL	9.14	0.45	0.93	9.08	0.52	0.93	9.56	0.45	0.91	7.10	0.47	0.95
EB-AL	14.11	0.47	0.98	14.36	0.60	0.98	15.73	0.58	0.98	7.39	0.44	0.95
P-AL	**11.16**	0.43	**0.95**	**11.66**	0.53	**0.95**	**13.01**	0.60	**0.95**	**5.52**	0.44	**0.95**
UAP	11.16	0.29	0.33	11.66	**0.21**	0.34	13.01	**0.27**	0.43	5.52	**0.15**	0.30
F-UAP	11.16	**0.23**	0.90	11.66	0.32	0.93	13.01	0.34	0.91	5.52	0.16	0.70

Table 3: L_∞ attack results, provided in terms of L_2 and L_∞ metrics and FR refers to the Fooling Rate. Note that UAP (Moosavi-Dezfooli et al. 2017) and F-UAP attacks (Zhang et al. 2020) are set to reach the same L_∞ values as P-AL to allow comparison of FR at the same level of perturbation.

Method	DenseNet121			ResNet50			GoogleNet			VGG16		
	L_∞	L_2	FR	L_∞	L_2	FR	L_∞	L_2	FR	L_∞	L_2	FR
B-AL	0.17	24.96	1.00	0.17	26.64	1.00	0.20	26.63	1.00	0.16	19.04	1.00
EB-AL	0.16	24.42	1.00	0.18	29.39	1.00	0.22	33.55	1.00	0.17	25.70	1.00
P-AL	**0.11**	16.76	**0.96**	**0.13**	16.61	**0.96**	**0.16**	20.60	**0.96**	**0.11**	12.45	**0.95**
UAP	0.11	25.71	0.52	0.13	**29.64**	0.60	0.16	**35.32**	0.79	0.11	**25.84**	0.75
F-UAP	0.11	**25.55**	0.99	0.13	26.74	0.99	0.16	31.00	0.99	0.11	24.57	0.99

$$f(x) = \begin{cases} 1, & \text{if } \sqrt{(\frac{w}{2} - x)^2 + (\frac{h}{2} - y)^2} \geq D \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

This method can be applied to any minimization based UAP training scenario, as for the norm constrained attacks, the perturbations do not always mitigate towards the edges. Also, we empirically found that smooth circular filters such as 2D Gaussian or Butterworth (Butterworth et al. 1930) filters tend to limit the adversarial capacity of the UAPs, by slightly smoothing the features around the edges.

Experimental Design

We have trained the UAPs using a sampled ImageNet dataset containing a total of 10000 images, formed by 10 images from each class. We have compared our attack with 2 other attacks which can be trained with small dataset sizes; vanilla UAP (Moosavi-Dezfooli et al. 2017) and Feature-UAP attacks (Zhang et al. 2020). We have chosen the target class as *peacock* for our attacks and Feature-UAP (vanilla UAP is strictly an untargeted attack as it is based on DeepFool). Similar to our attack, vanilla UAP allows specification of a target fooling rate over the training set. Therefore to allow comparisons on the same ground, we set this parameter to be the same, specifically, to a target fooling rate of 95%. However, as both of these attacks are norm constrained attacks, a direct comparison is not possible; therefore we first trained UAPs with our attacks, then we set the constraints, i.e. epsilons to match our obtained L_2 or L_∞ values depending on the type of the norm which was selected to be optimized. To measure the performance of the attacks, we have used the standard ImageNet validation set having 50000 images. For fast convergence, Adam was selected as the optimizer, and the UAPs have been trained for 20 epochs. For the experiments where filtering is applied, a radius of 112 is used, as the dimensions of the input images are 224×224.

Results

Two different experiments have been conducted by applying the attacks with L_2 and L_∞ norms, which are both supported by all attack types in question. The results are then compared with regards to both L_2 and L_∞ values.

L_2 attacks

Table 2 shows the L_2 attack results for B-AL, EB-AL, P-AL, vanilla UAP and Feature-UAP (F-UAP). For 95% fooling rate constrained attacks, the attack is regarded successful if it is above and close to the target. Despite achieving the smallest L_2 value compared to the other base models, B-AL cannot attain the target FR for DenseNet121 (Huang et al. 2018), ResNet50 (He et al. 2015) and GoogleNet (Szegedy et al. 2014). It can only reach the target for VGG16 (Simonyan and Zisserman 2015) with batch normalization, however in that case its L_2 results are comparatively higher. On the other hand, EB-AL is above the target FR by 3 percentage points (except for VGG16 where is achieves the target FR), which renders the attack sub-optimal; furthermore the L_2 values are consistently higher than both B-AL and P-AL. P-AL, which was introduced to address the inefficiencies of B-AL and EB-AL, consistently achieves the desired fooling rate while also having the smallest L_2 values. Overall, P-AL stays inside the desired range; furthermore by only integrating a filter during training, (Table 4) FP-AL achieves even lower perturbation levels, both in terms of L_2 and distance from the desired fooling rate. This algorithm consistently achieves 95% fooling rate, while yielding the smallest

Table 4: Comparison between the results of P-AL and FP-AL, in both L_2 and L_∞. FR signifies the fooling rate over the whole dataset.

Method	DenseNet121			ResNet50			GoogleNet			VGG16		
	L_2	L_∞	FR	L_2	L_∞	FR	L_2	L_∞	FR	L_2	L_∞	FR
L_2 P-AL	11.16	0.43	**0.95**	11.66	**0.53**	**0.95**	13.01	0.60	**0.95**	5.52	0.44	**0.95**
L_2 FP-AL	**8.38**	**0.41**	**0.95**	**10.75**	0.60	**0.95**	**10.62**	**0.52**	**0.95**	**8.93**	**0.43**	**0.95**
L_∞ P-AL	16.76	**0.11**	**0.96**	16.61	**0.13**	**0.96**	20.60	**0.16**	**0.96**	**12.45**	**0.11**	**0.95**
L_∞ FP-AL	**15.57**	0.15	0.97	**16.42**	0.15	**0.96**	**18.80**	0.18	0.97	13.28	0.13	0.96

-successful- L_2 distance over any of the given attacks.

It should be noted that both UAP and F-UAP are meant to be mainly run under L_∞ constraints, however the algorithms are suitable for L_2 normalization during training. As mentioned earlier, both of these attacks are norm constrained attacks as opposed to our minimization attacks which makes it difficult to compare them. However, when the L_2 constraints are equalized at the level of our attacks, we see that both UAP and F-UAP fall below the desired fooling rate; nonetheless both reach significantly smaller L_∞ values compared to our attacks.

L_∞ attacks

Table 3 shows the results of the L_∞ attacks. It should be noted that while our attacks are mainly designed to minimize L_2 norms of the UAPs, they can minimize L_∞ as well. This time, B-AL and EB-AL overshoots the desired fooling rate, which is not optimal in our constraints; besides, their L_∞ values are comparatively higher. P-AL achieves both better L_2 and L_∞ values, while staying closer to the desired fooling rate. According to the results on Table 4, FP-AL slightly increases the L_∞ values while also getting further from the target, in exchange for an overall decrease in L_2 norm.

As UAP and F-UAP are mainly L_∞ bounded attacks, it is fair to expect better results from them. UAP shows much better results compared to L_2 normalized attack, however still cannot reach the target fooling rate against either of the networks. F-UAP achieves 99% fooling rate for each network type, albeit yielding comparatively higher L_2 values.

Discussion

Altering the loss function based on the progressive fooling rate gives the best results in both L_2 and L_∞ attacks. Although our other attacks take a similar approach, because each batch affects the optimization too much, alternating loss scheme causes unstable behaviour, thus makes it harder to converge to the desired fooling rate. Another possible drawback from these attacks comes from the fact that the batch size takes a crucial role on how the optimization proceeds. For instance, the final UAPs that were trained with batch sizes of 32 and 128 may have severe performance difference, since by increasing the sample size, we get a better understanding about the performance over the whole dataset. P-AL is independent of the batch size and it is more stable around the desired fooling rate.

We should also point out that P-AL addresses the problem of obtaining a fooling rate around the target level, not exceeding it to obtain even a better fooling rate. In a case where the target is to maximize fooling rate as much as possible, our attacks are likely to be less optimal. For instance, if we set the target fooling rate as 100%, the UAP will be trained only with the adversarial loss function, therefore never minimizing the L_p norm. To maximize the fooling rate, EB-AL may be the better choice, since it will first bring the fooling rate to 100%, if possible, then it will try to minimize the norm. Table 2 and 3 shows that higher fooling rates can be achieved by EB-AL, although the L_p norms are slightly higher. In that note, using L_∞ norm to optimize the fooling rate can be another way of maximizing the fooling rate, along with B-AL and EB-AL. Using L_∞ usually makes the optimization converge much faster at high fooling rate targets; since L_∞ norm is in s scale [0-1], the adversarial loss function (cross-entropy) takes much higher values thus updates the perturbation values more drastically. In those cases, while optimizing the UAP in a strong adversarial manner, a small norm optimization is also done, which can yield a UAP with a very high fooling rate.

Perturbation Features

Figure 2 shows sample UAPs trained with different methods. The UAPs on the first 3 rows (obtained without a filter) exhibit perturbations with visible image features accumulated around the edges. However, this phenomenon is not caused by the alternating loss scheme adopted, rather it is a consequence of performing targeted universal attacks that minimize the targets standard loss. The gradients on the perturbation are high over the edges after only few iterations, which causes the image features to concentrate on these regions. Figure 3 shows the scatter plots of the mean gradient values when a UAP is applied on the whole dataset, versus the distance of each pixel containing the gradient. The scatter plots are generated using a state of a UAP that is trained; from left to right, top to bottom, the UAP is taken from iterations 1, 30, 150 and 300 of the training. The gradients flow to the pixels that are far from the center of the images. We speculate that, since usually the main objects of the images are located around the center, the magnitudes of gradients with respect to a loss function whose objective class is different from the original class become relatively higher where features from the original object are absent, hence the edges and corners. On the other hand, it is also possible to see small feature-full perturbations that are generated around the center of the image, such as the green dots that can be seen in Figure 2. It is known that universal perturbations take advantage of image features that outweigh the original image features, hence why it can be possible to understand the target

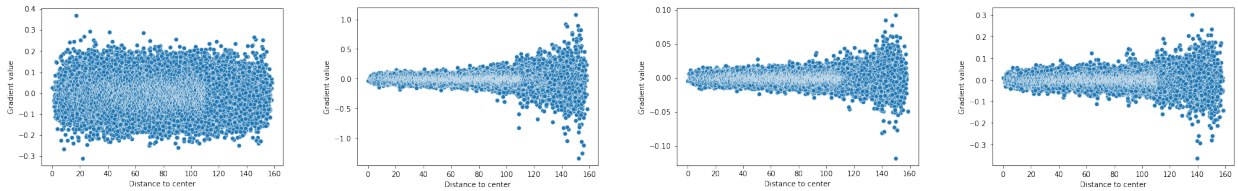

Figure 3: Vertical axes show the mean gradient value, horizontal axes show the distance between the pixel containing the corresponding mean gradient, and the center of the image. The scatter plots are extracted from UAP states after iteration number 1, 30, 150 and 300.

class by only looking at the universal perturbations; yet, the small accumulations around the center defy these assumptions. For that reason, our filtered training scheme not only makes the center of attention clean of perturbations, but also quantitatively yield better L_p norms and stable fooling rates.

Conclusion

In this work, we propose and evaluate alternative approaches for training a UAP that can achieve target fooling rates for a dataset, while being a minimization optimization, rather than being L_p bounded. For that, we have integrated 'alternating loss', an image-dependent attack strategy into universal adversarial domain. As it was not directly possible to integrate this strategy into a training procedure, we came up with 3 different approaches for its utilization. B-AL training altered the loss function based on solely the currently processed batch. EB-AL training also took the performance of the UAP over the whole dataset, before altering the loss function. Finally, P-AL training took into account the fooling rate up to the point where a batch is processed. Using P-AL, we achieved remarkable L_2 distances, while maintaining the desired fooling rates. On top of P-AL, we have also applied circular filtering to mitigate the small perturbations that appear in the center of the UAP, to the edges. By this way, we obtain perceptually better UAPs, having less perturbations in the center of the image, while achieving even smaller L_2 distances. On the other hand, this work can further be improved by regularizing the altered loss functions to achieve better L_p norms. Also, investigating the mitigation of the image-features on the UAPs can also be helpful for understanding not only the existence of these perturbations, but also the behaviour of the deep neural networks.

References

Aksoy, B.; and Temizel, A. 2020. Attack Type Agnostic Perceptual Enhancement Of Adversarial Images. International Workshop on Adversarial Machine Learning And Security (AMLAS), IEEE World Congress on Computational Intelligence (IEEE WCCI),19 July 2020.

Benz, P.; Zhang, C.; Imtiaz, T.; and Kweon, I. S. 2020. Double targeted universal adversarial perturbations. In *Proceedings of the Asian Conference on Computer Vision*.

Butterworth, S.; et al. 1930. On the theory of filter amplifiers. *Wireless Engineer*, 7(6): 536–541.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy)*, 39–57.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2018. Densely Connected Convolutional Networks. arXiv:1608.06993.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial examples in the physical world. arXiv:1607.02533.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2574–2582.

Mopuri, K. R.; Garg, U.; and Babu, R. V. 2017. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*.

Mopuri, K. R.; Ojha, U.; Garg, U.; and Babu, R. V. 2018. NAG: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 742–751.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2014. Going Deeper with Convolutions. arXiv:1409.4842.

Zhang, C.; Benz, P.; Imtiaz, T.; and Kweon, I. S. 2020. Understanding adversarial examples from the mutual influence of images and perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14521–14530.

Zhang, C.; Benz, P.; Karjauv, A.; and Kweon, I. S. 2021. Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. *arXiv preprint arXiv:2102.06479*.

Zhao, Z.; Liu, Z.; and Larson, M. 2020. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1039–1048.