Exploring Adversarial Examples with Various Input Size

Alex Jones
963906
963906@swansea.ac.uk

ABSTRACT

This paper presents an experiment carried out on the topic of Adversarial Examples. These are specialised inputs formed to confuse neural networks and cause them to incorrectly classify examples given. From the literature studied, we aim to:

- Comprehensively explain what Adversarial Examples cause
- Understand how such inputs could pose threats to data privacy and model security
- The current state-of-the-art (henceforth SOTA)
- If such examples are purely a problem or can be beneficial in any sense

The paper in question that was explored [2] has supplementary code (here).

INTRODUCTION

Deep Learning (DL) has shown large success when compared to human performance in tasks like: speech processing, image recognition, and classification tasks. However, it has come to light that even with non-distinguishable features, Adversarial Examples can successfully fool DL models with a high-success rate. This issue is at the forefront of minds across the globe as images that appear the same can completely change the output of any given model. What this means is that these models do not predict things how we thought they did and in fact still retain high-levels of uncertainty after training. It has been shown that different models *"misclassify the same example"*[2] and that this problem has a *"universal scaling...and holds for datasets...models...and attacks"*[1]. In computer vision approaches, most neural networks map the features of images as a space so that the perceived distance of images is almost the same as the Euclidean distance in that space; meaning how different the images are as seen by you should be approximately how far away they are in that space. As most networks have limited precision on individual input features, digital images mostly use 8 bits per pixel of the image meaning that any information below 1/255 of the range is discarded, most adversarial examples use perturbations that add less than this range to each pixel. The disappointment, therefore, stems from the fact that with Adversarial Examples and indistinguishable image differences a vastly different resultant classification is hard to prevent, and when prevented its hard to retain a SOTA model that is accurate for clean inputs. From solving and understand these attacks on DL models we stand to gain a greater knowledge of the inner workings of neural networks and how they classify, whilst also being able to make more robust and trustworthy systems knowing exactly how they arrive at respective outcomes.

METHOD AND EXPERIMENT

Potential Alternative Algorithms

In the field of Gradient-Based Optimization of models there are many potential whitebox attacks, 5 of which are from the FGSM family [5]. Other than the fact that FGSM was one of the originally proposed optimization attacks for models, it has been shown that even in SOTA it still is very effective. The other 2 Gradient-Based Optimization Attacks were not used as [5, 6]:

- JSMA was not used as it has a high computational complexity.
- DeepFool was not chosen as despite it having low complexity, its small perturbations and that it is designed to attack completely linear models did not appeal. Although it finds the minimal perturbation, the single-step to FGSM seemed superior for our experiement.

Jupyter

This experiment was carried out using Jupyter Notebook and Python using several common subsidiary libraries to aid the training process of the network. The model is a ResNet designed for speed called MobileNetV2 [4] and had its weights pre-trained on the commonly used image database ImageNet. As the model used is a ResNet architecture, the images input had to be *(224,224,3)* therefore the first function made was for image preprocessing, we then retrieved all labels from imagenet (of which there are 1000, also here). The function then defined was using the method described by *Goodfellow, I, et al.*[2] which was the Fast Gradient Sign Method (FGSM) which calculates the loss of the prediction based on the true label, from here it calculates the loss gradient based on the image and then the sign of the gradient; which overall leads to the generation of the adversarial pattern to attack the model. With a time constraint on the project this was chosen as it is successful across many models and its analytic gradient computation allows quick solutions [5]. Despite choosing to use FGSM we understood that the disadvantages were potential low success rate or label leakage [6]; but the advantages of low computational complexity and high transferability outweighed these.

These images were then displayed with varying strengths of adversarial perturbations (also called the *epsilon* value, for this we conducted using $\varepsilon = 0.01, 0.1, 0.15$).

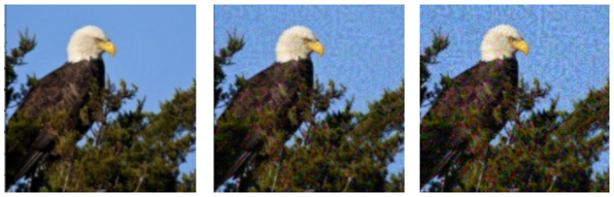


Figure 1: From left to right, epsilon values of 0.01, 0.1, 0.15 visual changes to the image

Experiment

We knew from our research that Adversarial Examples are valid and useful for attacking neural networks, we chose to use FGSM as it is a whitebox method of attack. As described, we chose our εvalues as they caused minimal visual differences to the human eye and would not cause as large errors as [2]. FGSM is also efficient, fast and causes many models to misclassify, knowing this we wanted to instead test how various input image sizes (then being upsampled) would affect the predictions. We have already seen that it is possible to have attacks caused by images with no perturbations [3], and the choice of use of FGSM instead of black-box methods or other whitebox methods is due to its speed.

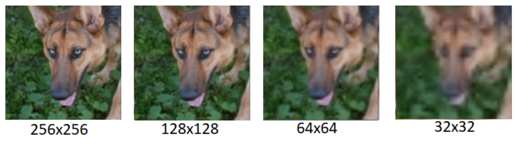
We hypothesised for the experiment that Adversarial Examples would change the classification much more drastically as the input images got smaller, theoretically this made sense as the upsampling of images causes blurring (to the human eye) and changes to pixel values as they are spread over a larger area. Using the tensorflow function to resize images this saved time when experimenting, from there our focus was to curate a dataset that was small and efficient. We also had the requirement that these images had to be the same across varying sizes and so we selected the Linnaeus 5 dataset; it contains 5 classes, 4 image sizes and the same images across these resolutions. We then selected 13 images to test our model on with Adversarial Examples. We aimed at: varying distances of target from camera, colour schemes, and orientation of target. These were to enhance understanding of if there were any examples more easily picked up by the pretrained model.

RESULT AND DISCUSSION

To collate the results of the experiment carried out after the review and research of relevant literature, we have found that with SOTA model and varying upsampled images misclassifications can occur much more drastically when the images are smaller. This shows there could be a high risk of security breaches through the generation of such Adversarial Examples. After testing across 13 differing images, we picked 4 results that stood out to us the most, for the remainder of the results, these can be found in the Github.

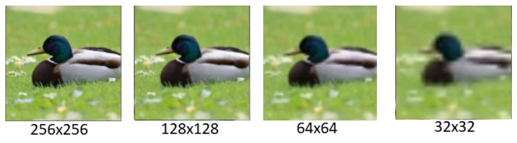
weimaraner.jpg	(Classification| Confidence)							
Image size	input		ε = 0.01		ε = 0.1		ε = 0.15	
32x32	labrador_retri	60.28%	golden_retrie	33.10%	chow	43.51%	jigsaw_puzzle	38.39%
64x64	labrador_retri	54.44%	labrador_retri	75.23%	chow	35.09%	border_terrier	17.29%
128x128	labrador_retri	42.01%	labrador_retri	29.45%	chow	42.60%	chow	17.69%
256x256	labrador_retri	38.95%	leonberg	41.02%	chow	56.79%	brown_bear	44.44%

Figure 2: Showing the model failing to classify correctly from the input onwards.



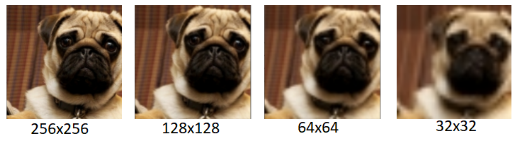
german_shepherd.jpg	(Classification| Confidence)							
Image size	input		ε = 0.01		ε = 0.1		ε = 0.15	
32x32	german_shepl	44.64%	leopard	23.48%	tabby	34.12%	prayer_rug	24.59%
64x64	german_shepl	19.69%	rhodesian_rid	7.75%	dingo	38.68%	tiger_cat	21.11%
128x128	kelpie	29.40%	toy_terrier	33.95%	dingo	35.80%	dingo	43.40%
256x256	german_shepl	74.36%	kelpie	11.77%	basenji	29.67%	dingo	35.10%

Figure 3: Showing the model struggling to classify after the input is perturbed once.



duck.jpg	(Classification| Confidence)							
Image size	input		ε = 0.01		ε = 0.1		ε = 0.15	
32x32	red-breasted_	26.32%	croquet_ball	37.11%	standard_poo	15.42%	porcupine	11.41%
64x64	drake	61.42%	drake	96.02%	standard_poo	11.40%	standard_poo	9.24%
128x128	drake	81.04%	drake	95.43%	croquet_ball	37.58%	croquet_ball	12.62%
256x256	drake	92.42%	drake	99.32%	drake	89.61%	peacock	41.47%

Figure 4: Showing the model failing after the first epsilon value generally.



pug.jpg	(Classification| Confidence)							
Image size	input		ε = 0.01		ε = 0.1		ε = 0.15	
32x32	pug	51.51%	Pekinese	65.52%	Pug	44.28%	Wombat	4.68%
64x64	pug	69.25%	pug	6.14%	pug	90.92%	pug	77.95%
128x128	pug	83.78%	pug	13.87%	pug	91.22%	pug	97.34%
256x256	pug	89.27%	pug	13.46%	pug	72.47%	pug	91.32%

Figure 5: Despite vast image resolution changes, shows the model still correctly classifying despite perturbations.

MobileNetV2 showed the fact that it was a less complex network when examining a few of our test images (see Figure 2) as it misclassified the image wrong without any need for Adversarial Examples. However, this could be down to many reasons that include but are not limited to: detail of image, closeness, hidden features in the latent space that we cannot perceive. When collecting our results there were a couple of anomalous cases in which the model would either get much more accurate by the last iteration or would misclassify altogether. These facts can partly be seen in both of the figures above. The general rule of thumb was that the classification would remain correct on input on original images sizes of 256, 128, and 64 pixels respectively but 32 pixels proved too tricky. Then, it can be seen in the results spreadsheet that for an εvalue of 0.01 about half of the predictions remained correct. From εof 0.1 and above, it is seen that unless there are very distinct image features the model struggled to classify each given input.

When viewing the results retrieved from our experiment we must consider:

- With a SOTA model was image choice poor?
- Why did this experiment cause the results it did?

- What could this imply for adversarial training?

Our image choices were not poor for this, as stated previously *"adversarial examples occur in contiguous regions of the 1-D subspace"*[2] which is defined by FGSM. Meaning any misclassifications we found through experimentation would likely be misclassified in similar ways by different models trained with different architectures. This is also backed by the fact that there is universality among Adversarial Examples [1].

This experiment caused the vast and diverse changes in predictions as the original images got smaller mainly because the perturbations had a much greater effect on the pixel-values across the input. As the original size shrunk, it meant it had to upscale by that much more, the perturbations were across each pixel, but being upscaled this would have meant that each pixel perturbed would have had that value increased through the resizing. Despite this, a few results such as Figure 5 retained the correct classification across 14 of its 16 total tests. It also saw an increase in the correctness of its classification after $\varepsilon = 0.1$, this anomaly held much different properties from any of the other examples tested and therefore we must assume that the model itself had a strong recognition of the facts that:

- The image was head on
- The animal in question had the majority of its most distinct features in the shot
- The colours were very clearly defined even on the lower quality images

In terms of adversarial training, this is already a common defense used for the prevention of Adversarial Examples and has been heavily researched (and continues to be) [1, 5]. It has been seen that this can decrease training speed [3] however when considering that adversarial examples are not novel to specific models and hold the same power across many Convolutional Networks, it becomes a worthy investment of time and resources because although not used extensively yet it is obvious that these could fast become exploits of systems.

To conclude, this problem is still a noteworthy part of research even in the SOTA and we have seen firsthand how it can affect a network drastically. From our results we saw that some Adversarial Examples could be used to actually increase the accuracy of a classification yet this does bring forward some more confusion. We have discussed that even in the SOTA it can be seen that non-edited or perturbed images can be used as Adversarial Examples and this means that models do not classify in the way that we previously thought. This could potentially lead to malicious aversion of systems and exploitation or gaming of particular models for the selfish gain of the attacker. From this research and discussion we understand that this problem is key and through the solution of this we will gain a deeper knowledge of the inner workings of the topic of DL as a whole and be able to generate robust, explainable, and trustworthy systems; this is also a major pitfall of ML currently.

REFERENCES

[1] Ekin D. Cubuk, Barret Zoph, Samuel S. Schoenholz, and Quoc V. Le. 2017. Intriguing Properties of Adversarial Examples. (2017). DOI: http://dx.doi.org/10.48550/ARXIV.1711.02846

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. (2014). DOI: http://dx.doi.org/10.48550/ARXIV.1412.6572

[3] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2019. Natural Adversarial Examples. (2019). DOI: http://dx.doi.org/10.48550/ARXIV.1907.07174

[4] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *CoRR* abs/1801.04381 (2018). http://arxiv.org/abs/1801.04381

[5] Rey Reza Wiyatno, Anqi Xu, Ousmane Dia, and Archy de Berker. 2019. Adversarial Examples in Modern Machine Learning: A Review. (2019). DOI: http://dx.doi.org/10.48550/ARXIV.1911.05268

[6] Jiliang Zhang and Chen Li. 2020. Adversarial Examples: Opportunities and Challenges. *IEEE Transactions on Neural Networks and Learning Systems* 31, 7 (2020), 2578–2593. DOI: http://dx.doi.org/10.1109/TNNLS.2019.2933524