Fooling a Deep Neural Network Forever

Anubhav Ashok Sree Harsha Kalli Prachee Sharma

Abstract

With deep learning systems becoming increasingly ubiquitous in the real-world, an adversarial attack could cause tremendous damage. Work presented in [1, 2] show that neural networks can be fooled easily. In light of this, we propose a system which automatically generates examples, which seem benign to humans, but consistently fool neural networks. While previous methods either only work on a single example or require explicit knowledge of the neural network, our proposed method assumes neither.

1. Introduction

Neural networks have become prevalent in commercial applications and affect humans on a daily basis. It is thus vital to ensure that neural networks are resistant to adversarial attacks. The AI community has begun to realize the direness of this issue, evidenced by the rise of organizations such as OpenAI and the Future of Life foundation. Papers such as [1] have explored the use of imperceptible adversarial perturbations to trick neural networks into misclassifying a particular image. Examples of attacks range from identify theft to taking control of an autonomous vehicles [8].

In response to these recent works, we propose to train a generative model that produces images of a particular class, that fools the neural network with high probability. Our approach is to train a GAN on images of dogs and concurrently teach it to learn the adversarial perturbations that fool the network as shown in 1

In the training, we combine an adversarial perturbation loss [2], with the discriminator loss [3] During testing time, the trained generator is able to generate dog-like images that fool the network into misclassifying it as human.

2. Related Work

Fooling Neural Networks In the paper [1] the authors discuss how deep neural networks learn input-output mappings that are discontinuous to a significant extent. This can cause a network to misclassify an image by an imperceptible perturbation, which is found by maximizing the networks prediction error. They show how the same perturbation can cause a different network trained on a different subset of the dataset, to misclassify the same input.

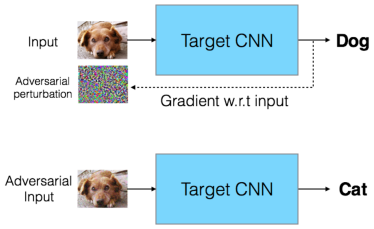

Figure 1. Fast signed gradient method

Generating adversarial examples In [2] the authors show that the primary cause of neural networks vulnerability to adversarial perturbation is their linear nature. By virtue of this, adding the gradient of images (computed using fast gradient sign method explained in the paper) to the images can produce adversarial examples which look imperceptibly similar to the original images, but result in the model outputting an incorrect answer with high confidence. The paper [3] formalizes the space of adversaries against deep neural networks and introduces a new class of algorithms to generate adversarial samples based on an understanding of the mapping between inputs and outputs of DNNs.

In our project, we generated a ground-truth dataset of adversarial perturbations for a target network using the fast signed gradient method which is as follows:

Algorithm 1 Fast Signed Gradient Method (FSGM)

procedure FSGM(x, y, f_θ)
  $\hat{y} \leftarrow f_\theta(x)$
  $\eta \leftarrow \nabla_x J(\hat{y}, y)$ ▷ Take gradient w.r.t input
  $\eta^* \leftarrow sign(\eta)$ ▷ Compute sign of gradient at each pixel
  $x^* \leftarrow x + \epsilon\eta^*$ ▷ Generate adversarial input
  return x^*
end procedure

We first perform a forward-backward pass on the target network to produce the gradient with respect to the input. We then take the signs of this gradient and add them to the input with a small scaling factor ϵ, we set this to value of 0.007 for imagenet images in the range of $[0, 1]$.

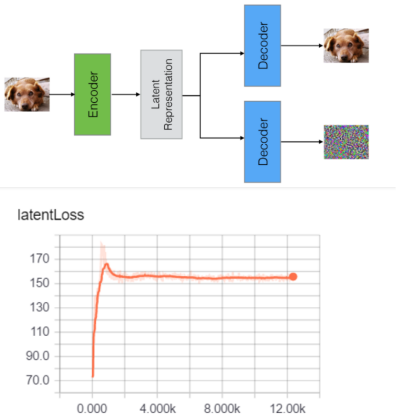

Figure 2. Variational AutoEncoder approach

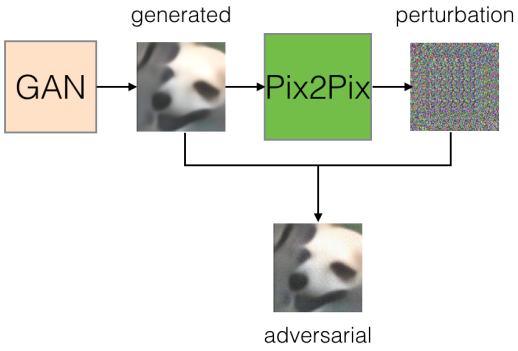

Figure 3. Overview of our method during test time

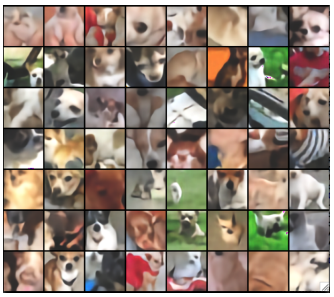

Figure 4. Dogs generated using BEGAN

Black Box Attacks In the paper [7] the authors introduce the first practical demonstration of an attacker controlling a remotely hosted DNN with no knowledge of either a model internals or its training data. They train a local model to substitute for the target DNN, using inputs synthetically generated by an adversary and labeled by the target DNN. They use the local substitute to craft adversarial examples.

3. Proposed Methods

3.1. Variational Autoencoder Architecture

Variational Autoencoders (VAEs) have emerged as one of the most popular approaches to unsupervised learning of complicated distributions. VAEs are appealing because they are built on top of standard function approximators (neural networks), and can be trained with stochastic gradient descent.

We initially tried to train the VAE with a joint loss by encoding the original image to a latent code and then jointly reconstructing the original image as well as its corresponding adversarial perturbation (obtained from the target network). Our architecture 2 used one encoder, one decoder to generate images and another decoder to generate gradients. We observed that the reconstruction did not converge correctly and in fact inherited some of the patchy patterns of the gradients. While this approach did not work, it was an interesting negative result worth mentioning. The loss plots and architecture are shown in 2.

3.2. Hybrid GAN Architecture

In this section we describe our hybrid GAN architecture 3 that actually worked well in this task. We first use a Boundary Equilibrium GAN to generate images of dogs, and then use a Conditional GAN, Pix2Pix, to translate the generated image into its adversarial perturbation. Finally we combine the generated image and its adversarial perturbation to form an adversarial example that is able to fool the black-box target neural network during test-time.

3.2.1 Boundary Equilibrium GAN

The Boundary Equilibrium GAN (BEGAN) [10] network uses an equilibrium enforcing method coupled with a loss computed using the Wasserstein distance for training autoencoder based GANs. This balances the generator and discriminator during training and provides fast training which is robust to parameter changes, has a convergence measure and high visual quality.

We used this in order to be able to generate high quality images of dogs. The image in 4 shows a sample of the results we obtained by training the model for 200K iterations. While there are some very realistic looking dog images, some of them do not resemble dogs. Regardless, 80% of the generated images do indeed get classified as dogs.

The original BEGAN paper uses a 360K large dataset of celebrity images, however the dataset we used **The Stanford Dogs Dataset** which only had 20.5K images of dogs. We think the results can be improved by using a larger dataset or using data augmentation to generate more images.

For our experiments we used a discriminator which was architected as an autoencoder as in the original paper. We used 3x3 convolutions with exponential linear units at the outputs. Each layer was repeated two times and the convolution filters were increased linearly with each downsam-

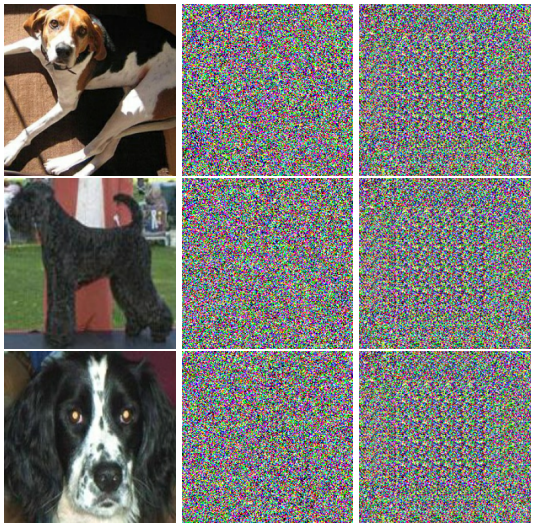

Figure 5. original image—adversarial gradient—learned adversarial perturbation

pling. For down sampling we subsampled with stride 2. Upsampling was done using nearest neighbor. Between the encoder and the decoder, the data was mapped using fully connected layers (without any non linearities), to and from an embedding state. For the generator the same architecture as the discriminator decoder was used but with different weights. The input state was initialized uniformly from [-1, 1]. We also used the Adam optimizer with an initial learning rate in [5 105 , 104].

3.2.2 Pix2Pix

The second component of the project was the conditional GAN architecture also known as Pix2Pix[11]. Pix2pix is a conditional adversarial network for image to image translation problems which learns the mapping from input image to output image along with a loss function to train this mapping. It is effective at generating photos from label maps, reconstructing objects from edge maps, and colorizing images. We used this network to form a mapping between an input image and its adversarial perturbation. It was trained with pairs of training images of dogs and corresponding gradients previously generated using the fast-signed gradient method. We also attempted using images generated by the GAN and their corresponding gradients to train this network but using the original images worked better. While the outputs produced by the Pix2Pix network were not the exact gradients, they worked about 70% of the time. This high success rate, while surprising, agrees with the discussion in recent paper[12] which states that adversarial examples span a contiguous subspace of large dimensionality and that a significant fraction of this space is shared between different models.

5 shows images of the outputs of the Pix2Pix network.

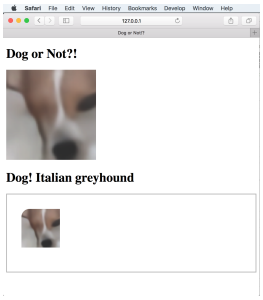

Figure 6. Dog or Not!? : Web demo interface

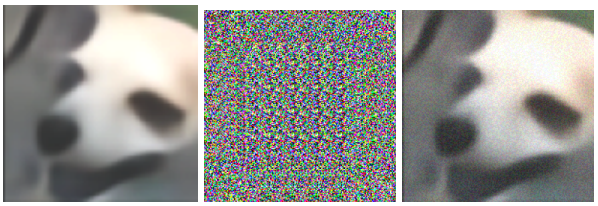

Figure 7. BEGAN Image classified as Dog—Pix2Pix Perturbation—Perturbed image classified as Golf Ball

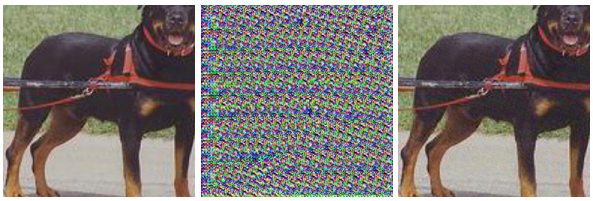

Figure 8. Real Image classified as Dog—Pix2Pix Perturbation—Perturbed image classified as Horse cart

4. Experiments

To demonstrate our results, we built a web interface 6, which had a tensorflow backend running a VGG-16 model which was pre-trained on ImageNet. We first uploaded the unperturbed image (either generated or real) and confirmed that it classified as a dog, then using the pix2pix network to generate a perturbation. We then combined this perturbation with the image to generate an adversarial example. We then tested whether the adversarial example fooled the neural network. We found that different examples had slightly different optimal ϵ values and that a higher ϵ doesn't necessarily increase the probability of fooling the network.

The figures in 7, 8 show our final results in which we are able to fool a neural network. We are able to generate perturbations for an image generated by our generative model, the BEGAN which misclassify the image. We also show in that we can use Pix2Pix to generate perturbations for real images. This is important as our experiments showed that the images generated by BEGAN sometimes do not resemble dogs. In the future, the BEGAN architecture can be replaced by any generative model that is able to generate realistic dog images without changing the Pix2Pix network.

References

[1] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).

[2] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[3] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

[4] Hinton, Geoffrey E. "To recognize shapes, first learn to generate images." Progress in brain research 165 (2007): 535-547.

[5] Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[6] Moosavi-Dezfooli, Seyed-Mohsen, et al. "Universal adversarial perturbations." arXiv preprint arXiv:1610.08401 (2016).

[7] Papernot, Nicolas, et al. "The limitations of deep learning in adversarial settings." Security and Privacy (EuroS&P), 2016 IEEE European Symposium on. IEEE, 2016.

[8] Papernot, Nicolas, et al. "Practical black-box attacks against deep learning systems using adversarial examples." arXiv preprint arXiv:1602.02697 (2016).

[9] Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." arXiv preprint arXiv:1605.07277 (2016).

[10] Berthelot, David, Tom Schumm, and Luke Metz. "BEGAN: Boundary Equilibrium Generative Adversarial Networks." arXiv preprint arXiv:1703.10717 (2017).

[11] Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." arXiv preprint arXiv:1611.07004 (2016).

[12] Tramr, Florian, et al. "The Space of Transferable Adversarial Examples." arXiv preprint arXiv:1704.03453 (2017).