ADVERSARIAL EXAMPLES FOR GENERATIVE MODELS

Jernej Kos
National University of Singapore

Ian Fischer
Google Research

Dawn Song
University of California, Berkeley

ABSTRACT

We explore methods of producing adversarial examples on deep generative models such as the variational autoencoder (VAE) and the VAE-GAN. Deep learning architectures are known to be vulnerable to adversarial examples, but previous work has focused on the application of adversarial examples to classification tasks. Deep generative models have recently become popular due to their ability to model input data distributions and generate realistic examples from those distributions. We present two classes of attacks on the VAE-GAN architecture and demonstrate them against networks trained on MNIST, SVHN, and CelebA. Our first attack directly uses the VAE loss function to generate a target reconstruction image from the adversarial example. Our second attack moves beyond relying on the standard loss for computing the gradient and directly optimizes against differences in source and target latent representations. We additionally present a new visualization, which gives insight into how adversarial examples appear in generative models.

1 INTRODUCTION

Adversarial examples have been shown to exist for a variety of deep learning architectures. They are small perturbations of the original inputs, often barely visible to a human observer, but carefully crafted to misguide the network into producing incorrect outputs. Seminal work by Szegedy et al. (2013) and Goodfellow et al. (2014), as well as much recent work, has shown that adversarial examples are abundant and finding them is easy.

Most previous work focuses on the application of adversarial examples to the task of classification, where the deep neural network assigns classes to input images. Deep generative models, such as Kingma & Welling (2013), learn to generate a variety of outputs, ranging from handwritten digits to faces (Kulkarni et al., 2015), realistic scenes (Oord et al., 2016), videos (Kalchbrenner et al., 2016), 3D objects (Dosovitskiy et al., 2016), and audio (van den Oord et al., 2016). These models learn an approximation of the input data distribution in different ways, and then sample from this distribution to generate previously unseen but plausible outputs.

One of the most basic applications of generative models is input reconstruction. Given an input image, the model first encodes it into a lower-dimensional latent representation, and then uses that representation to generate a reconstruction of the original input image. Since the latent representation usually has much fewer dimensions than the original input, it can be used as a form of compression.

These properties of input reconstruction generative networks suggest a variety of different attacks that would be enabled by effective adversaries against generative networks. Specifically, we consider an attack where the latent representation is used as a form of compression when transmitting an image between two parties. The attackers goal is to convince the sender to transmit an image of the attackers choosing to the receiver, but the attacker has no direct control over the bytes sent between the two parties. The sender believes that the receiver will reconstruct the same image that he sees, but if the attack is successful, the receiver will in fact reconstruct an image chosen by the attacker. We explore this idea in more detail as it applies to the application of compressing images using a VAE or VAE-GAN architecture.

We propose two attack methods, the *latent* attack and the $\mathcal{L}_{\text{VAE}}$ attack. Our results show that these attack methods are effective and VAE and VAE-GAN can be easily attacked. Additionally, we provide a new visualization, which gives insight into how adversarial examples appear in generative models.

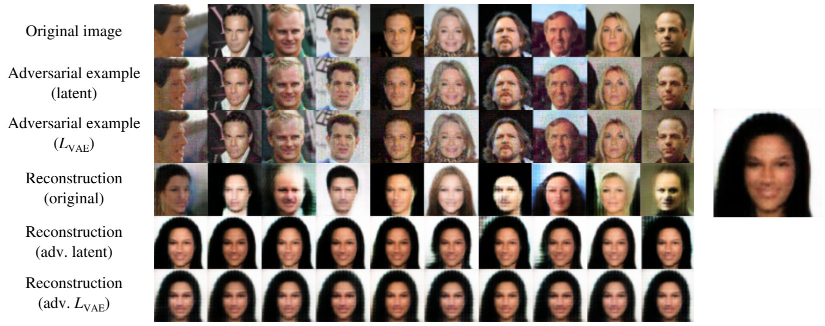
Original image
Adversarial example (latent)
Adversarial example (L_{VAE})
Reconstruction (original)
Reconstruction (adv. latent)
Reconstruction (adv. L_{VAE})

Figure 1: Summary of different attacks on CelebA dataset: original images, adversarial examples for both methods (latent and $\mathcal{L}_{\text{VAE}}$) and reconstructions of original images and adversarial examples. Target reconstruction is shown on the right.

Independent and concurrent to our work, Tabacof et al. (2016)[1] presents an adversarial attack on the VAE model. Their attack is similar to our latent attack, but they instead use the KL divergence between the latent representations of the source and target images as a metric (where we use the L2 distance). They briefly mention a "direct attack" which seems similar to our $\mathcal{L}_{\text{VAE}}$ attack, and say that the attack is not successful as it only makes the reconstructions more blurry. However in our experiments, we show the opposite, that the $\mathcal{L}_{\text{VAE}}$ attack is actually very successful, e.g., on the CelebA faces dataset. We also provide more in-depth and larger-scale study: we evaluate the attack on a more advanced model, the VAE-GAN model, with a more complex dataset, the CelebA faces dataset. Additionally we provide a new visualization, which gives insight into how adversarial examples appear in generative models.

2 METHODS

In this work, we consider generative models such as VAE and VAE-GAN (see Appendix for some background), where f_{dec} and f_{enc} denotes the decoder and encoder respectively. We propose two attack methods and use optimization-based attacks to generate the adversarial examples in both cases.

$\mathcal{L}_{\text{VAE}}$ attack Our first approach generates adversarial perturbations using the VAE loss function, $\mathcal{L}_{\text{VAE}}$. The attacker chooses two inputs, $\mathbf{x}_s$ (the source) and $\mathbf{x}_t$ (the target), and uses one of the standard adversarial methods to perturb $\mathbf{x}_s$ into $\mathbf{x}^*$ such that its reconstruction $\hat{\mathbf{x}}^*$ matches the reconstruction of $\mathbf{x}_t$.

The adversary precomputes the reconstruction $\hat{\mathbf{x}}_t$ by evaluating $f_{\text{dec}}(f_{\text{enc}}(\mathbf{x}_t))$ once before performing optimization. In order to use $\mathcal{L}_{\text{VAE}}$ in an attack, the second term (the reconstruction loss) of $\mathcal{L}_{\text{VAE}}$ is changed so that instead of computing the reconstruction loss between $\mathbf{x}$ and $\hat{\mathbf{x}}$, the loss is computed between $\hat{\mathbf{x}}^*$ and $\hat{\mathbf{x}}_t$. This means that during each optimization iteration, the adversary needs to compute $\hat{\mathbf{x}}^*$, which requires the full $f_{\text{dec}}(f_{\text{enc}}(\mathbf{x}^*))$ to be evaluated.

Latent attack This attack works by targeting the latent representation of the generative model. The adversary chooses a source image $\mathbf{x}_s$ and a target image $\mathbf{x}_t$, generating an adversarial example $\mathbf{x}^*$. The goal is to make the encoder produce a latent representation similar to the latent representation of $\mathbf{x}_t$, while keeping $\mathbf{x}^*$ similar to $\mathbf{x}_s$.

For this attack to work on latent generative models, it is sufficient to compute $\mathbf{z}_t = f_{\text{enc}}(\mathbf{x}_t)$ and then use the following loss function to generate adversarial examples from different source images $\mathbf{x}_s$:

$$\mathcal{L}_{\text{latent}} = L(\mathbf{z}_t, f_{\text{enc}}(\mathbf{x}^*)). \tag{1}$$

$L(\cdot)$ is a distance measure between two vectors. We use the L_2 norm, under the assumption that the latent space is approximately euclidean.

[1] Their work was made public shortly after we published our earlier drafts online.

3 Results

We evaluate the proposed methods on VAE-GAN under MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011) and CelebA (Liu et al., 2015) datasets. Due to space constraints, we only present the most interesting results on SVHN (Figure 2) and CelebA (Figure 1) (additional results are shown in Appendix). Our results show that generative models such as VAE-GAN can be easily fooled. Additionally, in Appendix, we show that the stochasticity of the VAE-GAN model only seems to have a minor effect on the success of adversarial attacks (see Figure 4).

We further generate a visualization of the reconstructions in input image space, showing that the direction of the generated adversarial example is much more effective than a random direction when generating adversarial examples. Similar in meaning to decision boundary plots (Goodfellow et al., 2014) for classification models, Figure 3 shows VAE-GAN reconstructions from different points in input image space spanned by the two directions. We generate the plot by defining two normalized vectors, $\mathbf{d}_1$ and $\mathbf{d}_2$, spanning the input image space. The one shown on the x-axis points in the direction of the generated adversarial perturbation ($\mathbf{d}_1$), while the other shown on the y-axis points in a randomly chosen orthogonal direction ($\mathbf{d}_2$). The images in the plane represent reconstructions computed by $f_{\text{dec}}(f_{\text{enc}}(\mathbf{x} + u\mathbf{d}_1 + v\mathbf{d}_2))$, where $\mathbf{x}$ is the original image. Values on the axes are the values of constants u and v. The target image used for the attack is the same as in Figure 1.

This visualization shows that if you move in the direction of the generated adversarial example, you quickly bump into adversarial examples, while moving in random directions in image space has no major effect on changing the reconstruction.

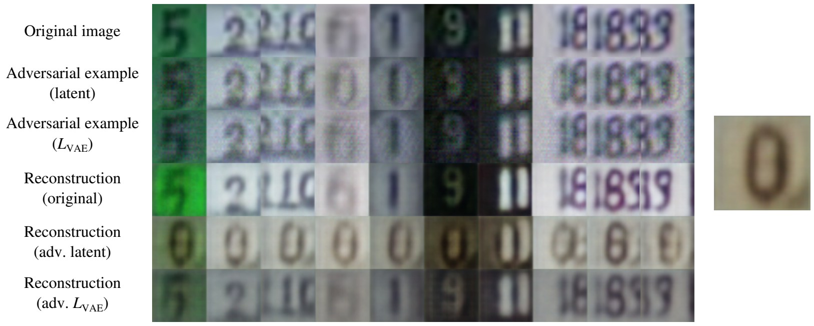
Original image
Adversarial example (latent)
Adversarial example (L_{VAE})
Reconstruction (original)
Reconstruction (adv. latent)
Reconstruction (adv. L_{VAE})

Figure 2: Summary of different attacks on SVHN dataset: original images, adversarial examples for both methods (latent and $\mathcal{L}_{\text{VAE}}$) and reconstructions of original images and adversarial examples. The $\mathcal{L}_{\text{VAE}}$ attack seems ineffective against SVHN in our experiments. Target reconstruction is shown on the right.

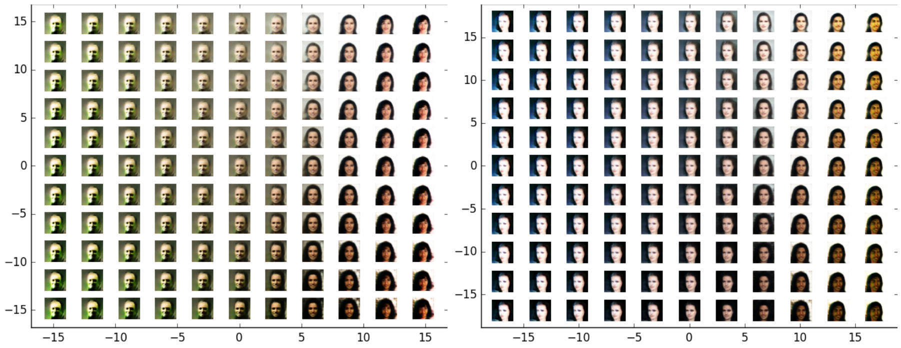

Figure 3: Visualization of VAE-GAN reconstructions in input image space. The x-axis is the attack direction, while the y-axis is a random orthogonal direction. The reconstruction of the original image is at the center $(0, 0)$.

References

Alexey Dosovitskiy, Jost Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2567384.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pp. 2539–2547, 2015.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

P. Tabacof, J. Tavares, and E. Valle. Adversarial Images for Variational Autoencoders. *ArXiv e-prints*, December 2016.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL `http://arxiv.org/abs/1609.03499`.

A Appendix

A.1 Background on VAEs and VAE-GANs

The general architecture of a variational autoencoder consists of three components. The **encoder** $f_{\mathrm{enc}}(\mathbf{x})$ is a neural network mapping a high-dimensional input representation $\mathbf{x}$ into a lower-dimensional (compressed) **latent representation** $\mathbf{z}$. All possible values of $\mathbf{z}$ form a latent space. Similar values in the latent space should produce similar outputs from the decoder in a well-trained VAE. And finally, the **decoder/generator** $f_{\mathrm{dec}}(\mathbf{z})$, which is a neural network mapping the compressed latent representation back to a high-dimensional output $\hat{\mathbf{x}}$. Composing these networks allows basic input reconstruction $\hat{\mathbf{x}} = f_{\mathrm{dec}}(f_{\mathrm{enc}}(\mathbf{x}))$. This composed architecture is used during training to backpropagate errors from the loss function.

The variational autoencoder's loss function $\mathcal{L}_{\mathrm{VAE}}$ enables the network to learn a latent representation that approximates the intractable posterior distribution $p(\mathbf{z}|\mathbf{x})$:

$$\mathcal{L}_{\mathrm{VAE}} = -D_{\mathrm{KL}}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] + E_q[\log p(\mathbf{x}|\mathbf{z})]. \tag{2}$$

$q(\mathbf{z}|\mathbf{x})$ is the learned approximation of the posterior distribution $p(\mathbf{z}|\mathbf{x})$. $p(\mathbf{z})$ is the prior distribution of the latent representation $\mathbf{z}$. D_{KL} denotes the Kullback–Leibler divergence. $E_q[\log p(\mathbf{x}|\mathbf{z})]$ is the variational lower bound, which in the case of input reconstruction is the cross-entropy $H[\mathbf{x}, \hat{\mathbf{x}}]$ between the inputs $\mathbf{x}$ and their reconstructions $\hat{\mathbf{x}}$. In order to generate $\hat{\mathbf{x}}$ the VAE needs to sample $q(\mathbf{z}|\mathbf{x})$ and then compute $f_{\mathrm{dec}}(\mathbf{z})$.

For the VAE to be fully differentiable while sampling from $q(\mathbf{z}|\mathbf{x})$, the reparametrization trick (Kingma & Welling, 2013) extracts the random sampling step from the network and turns it into an input, $\boldsymbol{\varepsilon}$. VAEs are often parameterized with Gaussian distributions. In this case, $f_{\mathrm{enc}}(\mathbf{x})$ outputs the distribution parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$. That distribution is then sampled by computing $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\varepsilon}\sqrt{\boldsymbol{\sigma}^2}$ where $\boldsymbol{\varepsilon} \sim N(0, 1)$ is the input random sample, which does not depend on any parameters of f_{enc}, and thus does not impact differentiation of the network.

The VAE-GAN architecture of Larsen et al. (2015) has the same f_{enc} and f_{dec} pair as in the VAE. It also adds a discriminator f_{disc} that is used during training, as in standard generative adversarial networks (Goodfellow et al., 2014). The loss function of f_{dec} uses the discriminator loss instead of cross-entropy for estimating the reconstruction error.

A.2 Effect of sampling

Additionally, we show that the stochasticity of the VAE-GAN model only seems to have a minor effect on the success of adversarial attacks (see Figure 4).

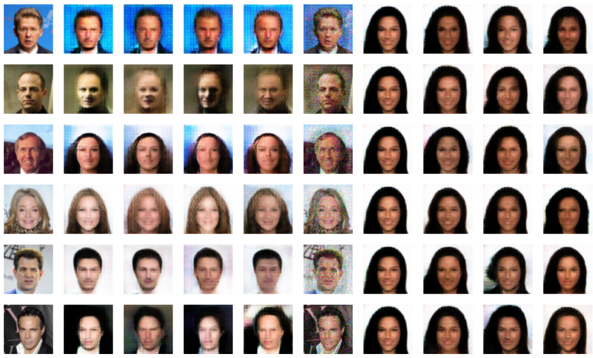

Figure 4: Effect of sampling on adversarial reconstructions. Columns in order: original image, reconstruction of the original image (no sampling), reconstruction of the original image (1 sample), reconstruction of the original image (12 samples), reconstruction of the original image (50 samples), adversarial example (latent attack), reconstruction of the adversarial example (no sampling), reconstruction of the adversarial example (1 sample), reconstruction of the adversarial example (12 samples), reconstruction of the adversarial example (50 samples).

A.3 CelebA

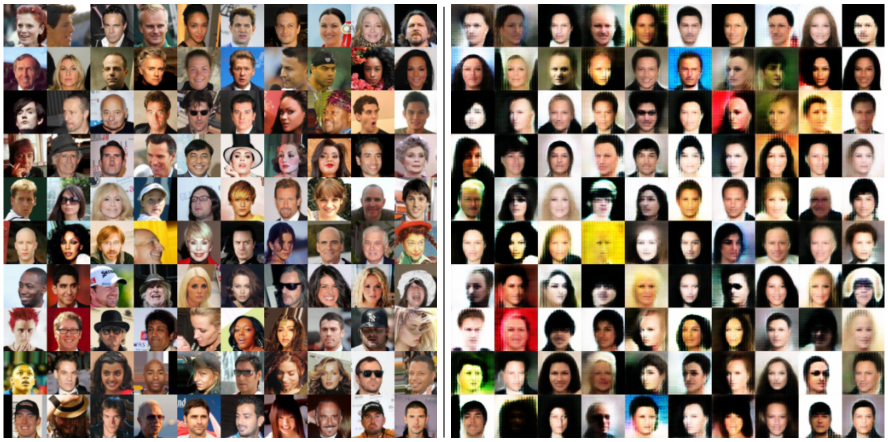

Figure 5: Original images in the CelebA dataset (left) and their VAE-GAN reconstructions (right).

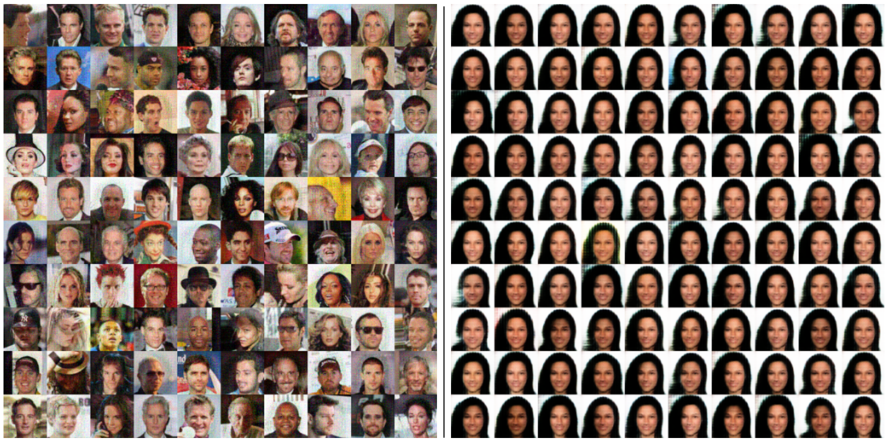

Figure 6: L_2 **Optimization Latent Attack on CelebA Dataset:** Adversarial examples generated for 100 images from the CelebA dataset (left) and their VAE-GAN reconstructions (right).

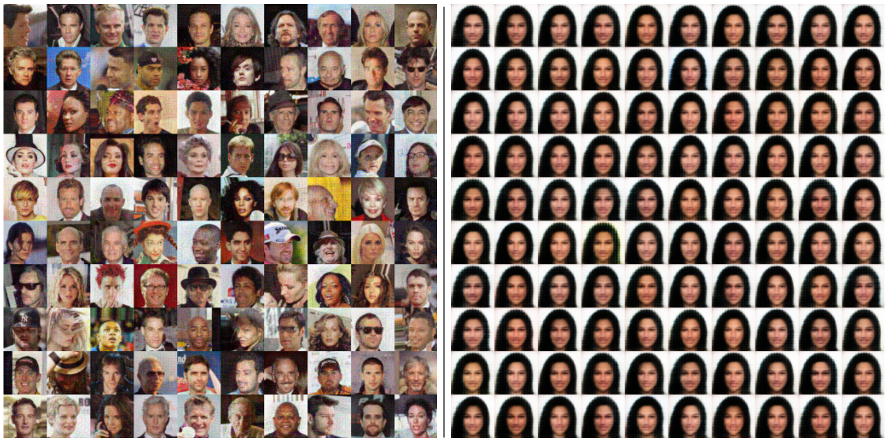

Figure 7: L_2 **Optimization $\mathcal{L}_{\text{VAE}}$ Attack on CelebA Dataset:** Adversarial examples generated for 100 images from the CelebA dataset (left) and their VAE-GAN reconstructions (right).