Targeted Attacks of Variational Autoencoders using the Fast Gradient Sign Method and a Quadratic Program Approximation

CS 544 - Optimization in Computer Vision

Christian Howard
howard28@illinois.edu

Abstract

In this work, we introduce adversarial examples and motivate the study of attacking mechanisms for generating adversarial examples. We discuss basic fundamentals of variational autoencoders (VAEs), the interior point method for constrained optimization, and use these ideas to discuss targeted adversarial examples for VAEs. In the discussion of targeted adversarial examples, we introduce a constrained optimization formalization of an adversarial attack for images. We step through the design of the attack objective and derive an approximate solution referred to as the Fast Gradient Sign Method (FGSM) for targeted attacks. An approximate targeted attack formulation referred to as the Quadratic Program (QP) attack is derived and the fundamental properties of this formulation are discussed. We then discuss the high level implementation details and results comparing the FGSM and QP attack methods. The results confirm that the FGSM method has comparably damaging affects to VAE reconstructions as the QP method but is much faster on average.

1 Introduction

In the context of developing intelligent systems using Machine Learning and other Artificial Intelligence techniques, there is a necessity to ensure such systems are robust to a wide range of inputs. The first thought is to ensure such system is robust against variations in the same type of input, which is certainly reasonable. However, more recent progress [1] has shown that an intelligent agent can produce strategic *adversarial* variations in inputs that can be used to make intelligent systems less robust.

To be able to properly defend against such adversarial inputs, we must understand processes to construct *adversarial examples*, examples that are designed to break an intelligent system in some manner. In this report, we will create an optimization attack variation applied to a generative model, specifically a Variational Autoencoder (VAE), and compare its performance relative to a common attack technique referred to as the Fast Gradient Sign Method (FGSM).

2 Fundamental Concepts

2.1 Variational Autoencoder

A variational autoencoder (VAE) is a generative model that uses concepts from variational inference to construct autoencoder models that are approximated by Gaussian factorizations. The VAE is comprised of two key parts, namely an *encoder* that maps data to likely latent variable instances and a *decoder* that maps latent variables to their likely data counterpart. For a given dataset $D \coloneqq \{x_i \in R^d\}_{i=1}^m$, we construct the desired encoder and decoder using, in our case, neural network architectures and applying variational inference to train these models such that they optimize a KL divergence cost. Once the model has been constructed, we can then use the decoder part to generate new data by sampling using a simple distribution. In our case, we use a normal distribution as the simple distribution.

2.2 Interior Point Methods

Within the class of optimization methods, *interior point methods* are a class of algorithms that effectively tackle constrained optimization problems by encoding their constraints into terms that are added to the objective. Using a log-barrier function, for example, we can transform the constrained optimization problem in (1) to the unconstrained optimization problem in (2).

$$\min_x f(x) \text{ such that } c_i(x) \geq 0 \, \forall i \in \{1, \cdots, K\} \tag{1}$$

$$\min_x f(x) - \tau \sum_{i=1}^{K} \log\left(c_i(x)\right) \tag{2}$$

where τ is a parameter you shrink algorithmically to refine your estimate over some number of iterations in hopes that you obtain a near optimal result. For a convex $f(x)$, the log-barrier in this example keeps the objective convex within the feasible set, making this technique quite powerful for convex problems. Of course, this technique can work for nonconvex problems as well, though you will not have the same global optimum guarantees. One other added benefit of this technique is that given you start within the feasible set, your final answer will end up in the feasible set as long as you never cross the barriers imposed by your constraints. With careful implementations, it can readily be seen that your approximate local minima is indeed a feasible result by construction.

2.3 Adversarial Examples for VAEs

In our context, let us specifically turn our attention to modeling images. Let us define the space of single channel images to be $\mathcal{I} \coloneqq \{I \in [0,1]^{r \times c}\}$, where r is the number of rows in the image and c is the number of columns. It is trivial to construct a bijection, and it need not be unique, between $\mathcal{I}$ and the set $\mathcal{X} \coloneqq \{X \in [0,1]^{rc}\}$, which we can view as the vectorization of the images. Assuming we call this bijection $\mathcal{H}(\cdot)$, we know there exists an inverse denoted $\mathcal{H}^{-1}(\cdot)$ and we will use these two operators for going from raw images into vectors and vice versa, such as when we need to produce figures or need to treat a raw image as a vector for optimization purposes.

Let us now define our latent variable space as $\mathcal{Z} \coloneqq \{z \in R^k\}$. Suppose we have some fixed VAE denoted $v(x) = d(e(x))$, where $e : \mathcal{X} \to \mathcal{Z}$ is the encoder and $d : \mathcal{Z} \to \mathcal{X}$ is the decoder. For some given base vectorized image x_b and target vectorized image x_t, we can pose a basic attack as the optimization seen in (3) for a given norm $\|\cdot\|$, producing the vectorized *adversarial example* $x' = x_b + \delta$.

$$\min_{\delta \in R^{rc}} \|v(x_b + \delta) - x_t\| \text{ such that } (x_b + \delta) \in \mathcal{X} \tag{3}$$

Of course, this attack may not necessarily be a good one because $\|\delta\|_\infty$ might be large enough that a human could inspect the adversarial example and decide it has been tampered with. This implies that it could be beneficial to add the additional constraint that $\|\delta\|_\infty \leq \epsilon$ for some small $\epsilon > 0$ to help ensure we find an adversarial example that is hard to identify as adversarial. This added constraint give us the problem posed in (4) for some fixed $\epsilon > 0$.

$$\min_{\delta \in R^{rc}} \|v(x_b + \delta) - x_t\| \text{ such that } ((x_b + \delta) \in \mathcal{X}) \wedge (\|\delta\|_\infty \leq \epsilon) \tag{4}$$

The infinity norm constraint implies a set of linear constraints of the form $-\epsilon \leq \delta_i \leq \epsilon$ for all $i \in \{1, \cdots, rc\}$. Due to the nature of our vectorized image space, the constraint that the adversarial example is a member of $\mathcal{X}$ also implies a set of linear constraints of the form $-(x_b)_i \leq \delta_i \leq 1 - (x_b)_i$ for all i. For our

purposes and to represent the software implementation, we can merge these two sets of constraints into the single set of constraints $\max\{-\epsilon, -(x_b)_i\} \leq \delta_i \leq \min\{\epsilon, 1-(x_b)_i\}$ for all i.

For our purposes, we will set our objective function norm to be the l_2 norm and square it for convenience. Now suppose we define $J(\delta) \coloneqq \|v(x_b+\delta) - x_t\|^2$ and make the approximation $J(\delta) \approx \hat{J}(\delta) \coloneqq J(0) + \nabla_\delta J(0)^T \delta$. The original attack formulation optimizes $J(\delta)$ but imagine we instead optimize $\hat{J}(\delta)$ subject to the constraint $\|\delta\|_\infty \leq \epsilon$ and ignore the constraint that the adversarial image is in the set $\mathcal{X}$ since we will just project the result into it. This gives us the alternative optimization formulation found in (6).

$$\min_{\delta \in R^{rc}} \sum_{i=1}^{rc} \left(\nabla_\delta J(0)\right)_i \delta_i \text{ such that } -\epsilon \leq \delta_i \leq \epsilon \, \forall i \in \{1, \cdots, rc\} \tag{5}$$

The optimization in (6) can be solved component-wise, allowing us to solve rc problems of the form

$$\min_{\delta_i \in R} \left(\nabla_\delta J(0)\right)_i \delta_i \text{ such that } -\epsilon \leq \delta_i \leq \epsilon \tag{6}$$

The objective in this case is affine and one-dimensional, so we can consider the cases where $\left(\nabla_\delta J(0)\right)_i$ is positive and negative. If $\left(\nabla_\delta J(0)\right)_i > 0$, we need to decrease δ_i to be smaller, thus $\delta_i^* = -\epsilon$. If $\left(\nabla_\delta J(0)\right)_i < 0$, we need to increase δ_i to be smaller, thus $\delta_i^* = \epsilon$. Using these two cases, we see that the optimal solution for any value $\left(\nabla_\delta J(0)\right)_i$ can be written as $\delta_i^* \coloneqq -\epsilon \text{sign}\left(\left(\nabla_\delta J(0)\right)_i\right)$. Taking this element-wise solution and generalizing, we find that the optimal deviation under this approximate model is $\delta^* \coloneqq -\epsilon \text{sign}\left(\nabla_\delta J(0)\right)$ and the resulting adversarial image is $x' = \Pi\left(x_b + \delta^*\right)$, where Π projects an object in R^{rc} into its closest element in $\mathcal{X}$. This optimal perturbation δ^* under the linearized model is similar to the result in [1] referred to as the Fast Gradient Sign Method (FGSM) but in this context differs by a sign since we are using a targeted output.

For this project, we extend this idea of approximating the original optimization formulation but consider approximating $J(\delta) \approx \bar{J}(\delta) = J(0) + \nabla_\delta J(0)^T \delta + \frac{1}{2}\delta^T H_J(0) \delta$, where $H_J(0)$ is the hessian of $J(\delta)$ evaluated at $\delta = 0$. This formulation clearly makes an assumption of continuity of the underlying objective and VAE used, but we assume such details are satisfied. Under this approximation, we have the new optimization to solve represented in (7).

$$\min_{\delta \in R^{rc}} \nabla_\delta J(0)^T \delta + \frac{1}{2}\delta^T H_J(0) \delta \text{ such that } -\epsilon \leq \delta_i \leq \epsilon \, \forall i \tag{7}$$

This new problem, which we will call a Quadratic Program (QP) attack, is clearly a quadratic program with linear constraints. If $H_J(0)$ was positive semi-definite, then this optimization problem would be convex and could take advantage of many powerful packages. Unfortunately, the nonconvexity common

in neural networks does not allow us to make this assumption. This means more general solvers need to be used that can handle quadratic programs such that $H_J(0)$ can be an indefinite symmetric matrix.

3 Implementation and Results

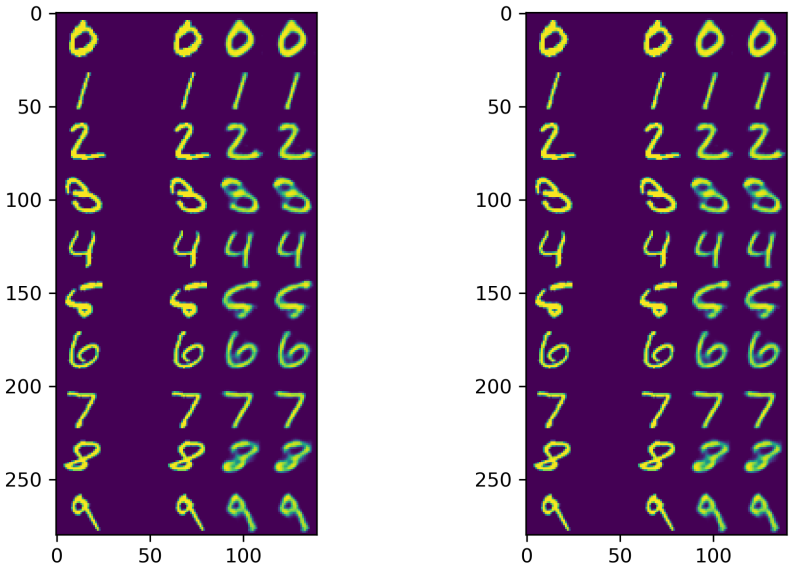
For a practical comparison of the FGSM attack and the QP attack, we use the MNIST[1] dataset which is comprised of 28×28 dimension images of handwritten digits. We separate the dataset into a training and test portion and use these partitions to construct our VAE model such that it has the best test error. The VAE model is a neural network with one input layer of dimension 784, three hidden layers using Rectified Linear Units (ReLU) activation functions with 512-16-512 as the number of nodes in their respective layers, and then an output layer if dimension 784 that uses sigmoid activation functions. The VAE model was trained with the ADAM optimization method for 50 epochs and a batch size of 128 data samples.

After training the VAE model, a random collection of digits were selected, one from each digit class going from 0 to 9. For digit d, it is assigned digit $d+1$ as its target unless $d = 9$, where for $d = 9$ its target digit is assigned to be 0. If we define $x^{(d)}$ as the vectorized image for digit d, this means in our optimization that for a given digit d, we have that $x_b = x^{(d)}$ and $x_t = x^{(d+1)}$ except in the case of $d = 9$ where we have that $x_b = x^{(9)}$ and $x_t = x^{(0)}$. We use this pairing to test and compare the FGSM attack to the QP attack on a range of different digits. For our purposes, we make use of a well known interior point optimization solver called IPOPT to solve the optimization associated with the QP attack. The IPOPT package uses a primal-dual interior point optimization method to tackle optimization problems of the form

$$\begin{aligned} \min_{x \in R^d} \quad & f(x) \\ \text{subject to} \quad & g^L \leq g(x) \leq g^R \\ & u^L \leq x \leq u^R \end{aligned} \tag{8}$$

We can see in Figures 1 through 5 the results between the FGSM attack and the QP attack for different values of ϵ. Both attack procedures do not produce any noticeable change to the VAE output for $\epsilon = 10^{-3}$ and $\epsilon = 10^{-2}$, but for $\epsilon = 10^{-1}$ and above, we see differences. A noticeable difference between the perturbations found using the FGSM attack and the QP attack is that the FGSM appears to remove more values from the base image while the QP attack appears to add more to the base image. These differences result in the reconstructions from the VAE for FGSM attack adversarial images to be dimmer and less sharp while the reconstructions for the QP attack adversarial images tend to have a stronger appearance and be sharper.

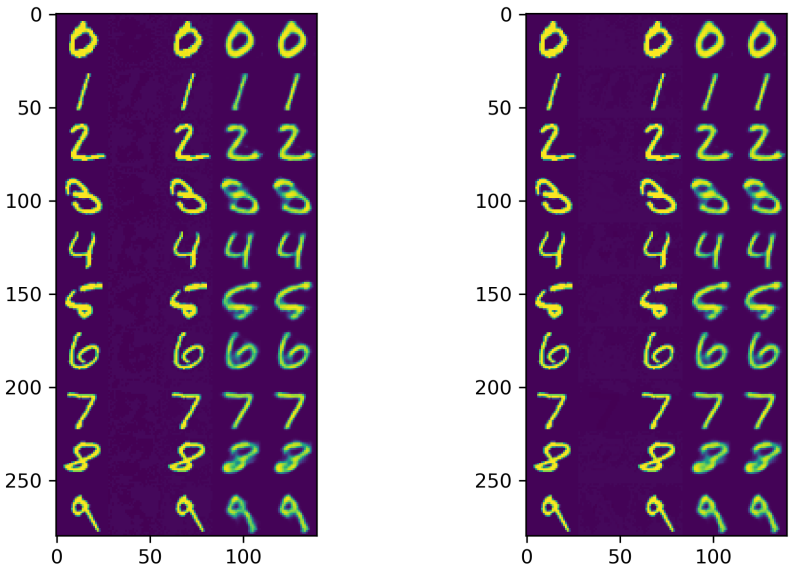
[1]MNIST site: http://yann.lecun.com/exdb/mnist/



(a) FGSM attack results with $\epsilon = 10^{-3}$ (b) QP attack results with $\epsilon = 10^{-3}$

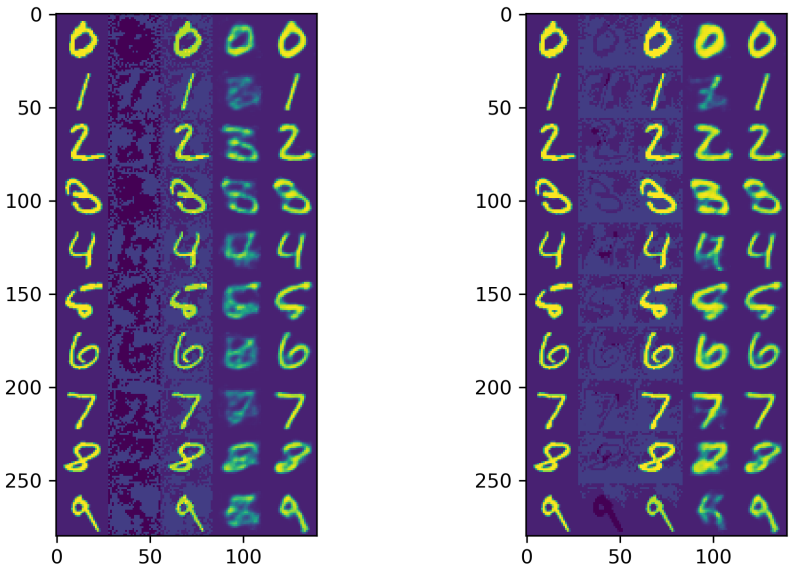
Figure 1: FGSM and QP attack results; Column 1 is clean image, column 2 is perturbation, column 3 is adversarial example, column 4 is output of VAE using adversarial example, column 5 is output of VAE using clean example

Both techniques make clear changes to the VAE reconstruction relative to the expected reconstruction based on the base image, but is it clear the FGSM attack wears away the structure of the base image in the reconstruction while the QP attack adds information that obfuscates what the base image was. Both methods have questionable success at achieving a target output that looks similar to the paired digit. The FGSM attack is computationally very efficient, taking only milliseconds to compute for a given image, while the QP attack using the IPOPT package in Python could take anywhere from a couple seconds to a couple minutes for a single image. Seeing as both attacks appear to generate adversarial images that sufficiently hurt the reconstruction of the VAE, the FGSM attack technique appears superior due to the efficient runtime.



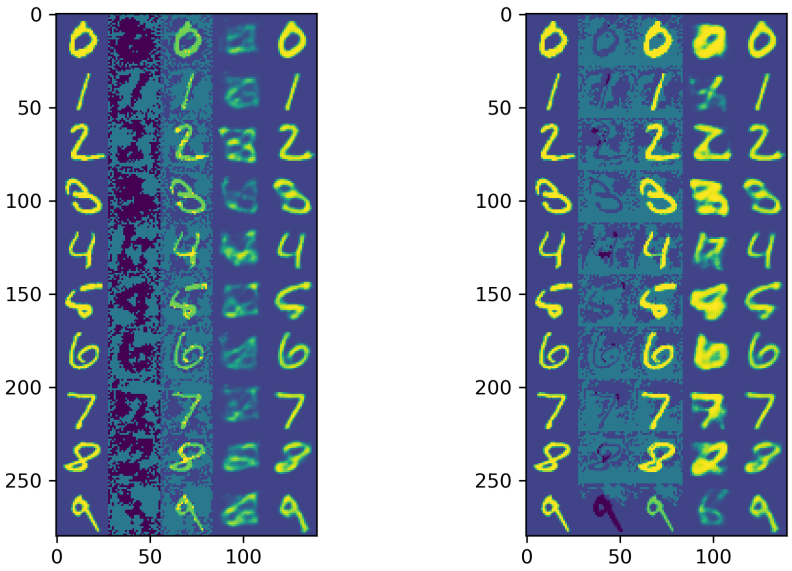
(a) FGSM attack results with $\epsilon = 10^{-2}$ (b) QP attack results with $\epsilon = 10^{-2}$

Figure 2: FGSM and QP attack results; Column 1 is clean image, column 2 is perturbation, column 3 is adversarial example, column 4 is output of VAE using adversarial example, column 5 is output of VAE using clean example



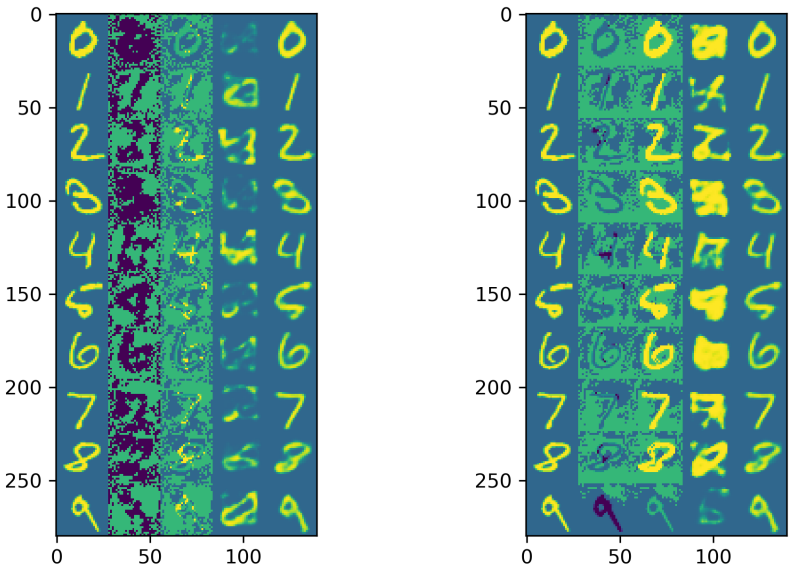
(a) FGSM attack results with $\epsilon = 10^{-1}$ (b) QP attack results with $\epsilon = 10^{-1}$

Figure 3: FGSM and QP attack results; Column 1 is clean image, column 2 is perturbation, column 3 is adversarial example, column 4 is output of VAE using adversarial example, column 5 is output of VAE using clean example



(a) FGSM attack results with $\epsilon = 0.25$ (b) QP attack results with $\epsilon = 0.25$

Figure 4: FGSM and QP attack results; Column 1 is clean image, column 2 is perturbation, column 3 is adversarial example, column 4 is output of VAE using adversarial example, column 5 is output of VAE using clean example



(a) FGSM attack results with $\epsilon = 0.5$ (b) QP attack results with $\epsilon = 0.5$

Figure 5: FGSM and QP attack results; Column 1 is clean image, column 2 is perturbation, column 3 is adversarial example, column 4 is output of VAE using adversarial example, column 5 is output of VAE using clean example

References

[1] I. J. Goodfellow, J. Schlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015, published in conference paper at ICLR 2015. [Online]. Available: https://arxiv.org/pdf/1412.6572.pdf

[2] D. A. Forsyth, "Interior point methods," 2019. [Online]. Available: http://luthuli.cs.uiuc.edu/~daf/courses/Opt-2019/Notes/interiorpoint2012.pdf

[3] A. Wchter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 2, pp. 25–57, 2006. [Online]. Available: https://link.springer.com/article/10.1007%2Fs10107-004-0559-y