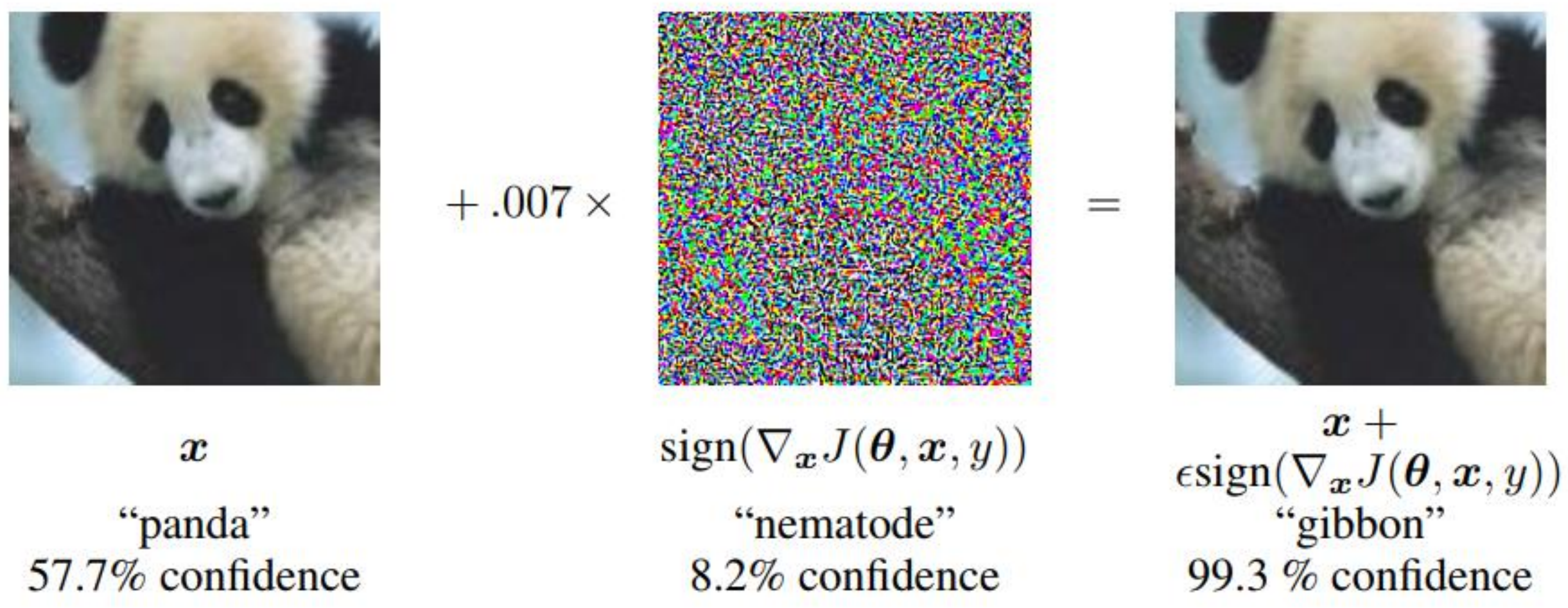
Generating Adversarial Examples with Generative Adversarial Networks

組別：1072A2　　指導教授：吳晉賢 教授
組員：B10502103 游子慶、B10502136 洪昌陞

Introduction

While researchers achieved great success in computer vision tasks using CNNs in the past few years, it's a known fact that many machine learning algorithms, e.g. CNNs, fully-connected networks, SVMs, etc., are easily fooled [1]. By adding carefully constructed perturbations to input images, one can mislead known models or even unknown models into classifying input into desired classes. Those carefully designed examples are also known as the adversarial examples. Adversarial perturbations are pretty much imperceptible to human eyes, while those perturbations lead models into misclassification. In this project, we utilize conditional GANs to generate adversarial examples, which is inspired by [2].


+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Generation process of adversarial example with FGSM [3]

Background

Generative Adversarial Networks (GANs)

GANs [4] are a branch of generative models, and GANs have known for producing high-quality and realistic examples compared to other kinds of generative models. The concept of GAN is a two-player game, in which there are discriminator and generator. The discriminator tries to distinguish between real data and fake data generated by the generator, and the generator tries to generate data that fools the discriminator. The implementations of generator and discriminator are usually deep neural networks especially deep convolutional neural networks. The discriminator and generator are trained with following loss functions:

$$L_D = -E_{(x,y)\sim p_{data}}[\min(0, -1+D(x,y))] - E_{z\sim p_z, y\sim p_{data}}[\min(0, -1-D(G(z),y))]$$

$$L_G = -E_{z\sim p_z, y\sim p_{data}} D(G(z), y)$$

Adversarial Examples

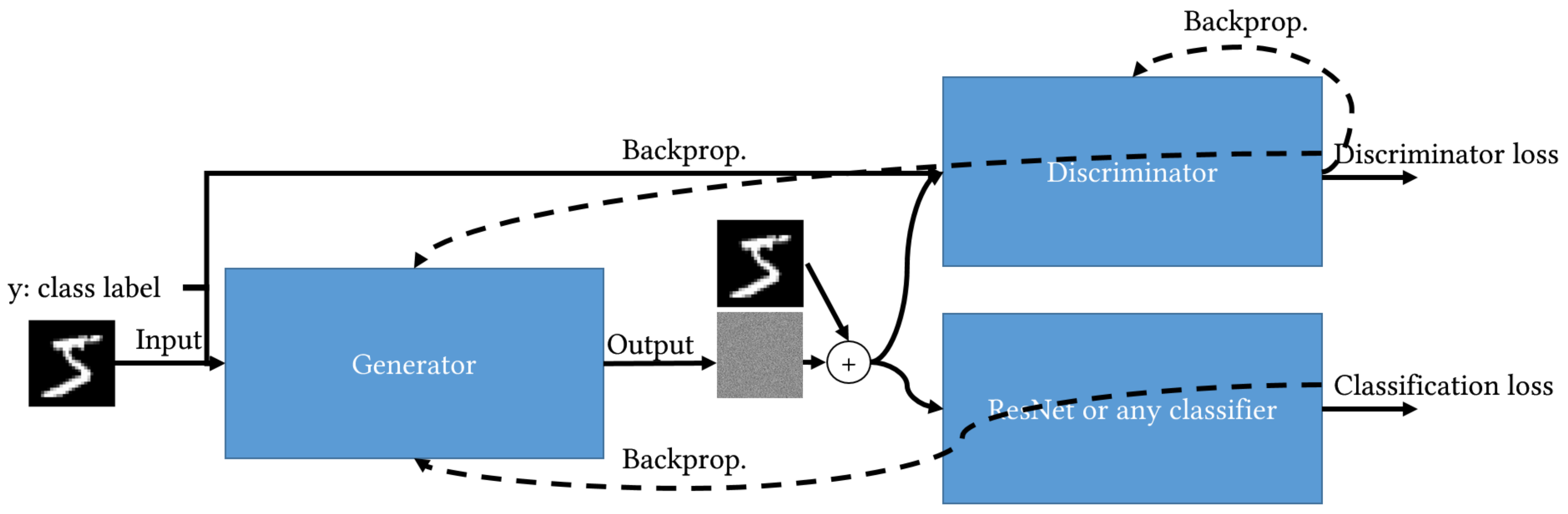
Adversarial examples are described as input images which are applied imperceptible non-random perturbations which change a classifier's predictions. In this project, we optimize the following function [5]:

$$\mathcal{F} = \max(\max_{i\neq t}(Z(x')_i) - Z(x')_t, -k)$$

where x' is an adversarial example, and $Z(x) = z$ are the logits which are the output of the classifier before softmax layer. k controls the confidence level of the adversarial example. Example is more likely to transfer to other models with larger k, because the loss is minimized only when the difference of the logit of the target class and the maximum logit among logits of classes other than the target class is greater than k.

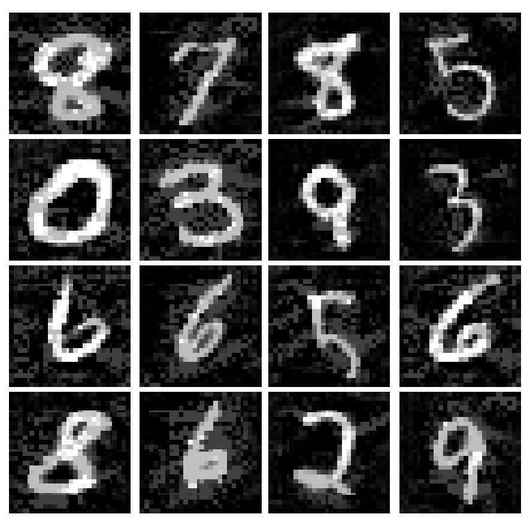
Architecture

We adopt the overall architecture of AdvGAN [1], and the model diagram is shown below. The generator is based on U-Net architecture which is an encoder-decoder architecture, and the discriminator is based on patchGAN architecture. The generator is jointly trained with GAN loss, adversarial loss, and perturbation loss. GAN loss make sure that the generated examples look like samples from real data distribution. Adversarial loss corresponds to the classification loss of the target model. Perturbation loss controls the magnitude of the perturbation.



Results & Conclusion

We first implemented LeNet5 and trained it using MNIST dataset, and then we used it as a target model for our GAN. The generated adversarial examples are shown below.



The predictions are:　3, 3, 8, 9, 4, 5, 8, 0, 1, 0, 1, 3, 3, 8, 6, 3.
The target labels are:　3, 3, 8, 9, 4, 5, 8, 0, 1, 0, 1, 3, 3, 8, 6, 3.

We also experiment with the CIFAR-10 dataset, and the perturbation is much more imperceptible.

It's easy to construct adversarial examples if the model's weight and gradient are exposed to the public. Therefore, it's important to be aware of potential adversarial attack while those applications are being designed.

References

[1] Intriguing properties of neural networks, C. Szegedy
[2] Generating Adversarial Examples with Adversarial Networks, Chaowei Xiao
[3] Explaining and harnessing adversarial examples, Ian Goodfellow
[4] Generative Adversarial Networks, Ian Goodfellow
[5] Towards Evaluating the Robustness of Neural Networks, Nicholas Carlini