1 Review and analysis of AdvGAN

As an important component of adversarial generative models, AdvGAN [1] first uses the generator G to generate perturbations $G(x)$ for the original input x. The synthesized adversarial sample $x + G(x)$ is then fed into the Discriminator D, which is used to distinguish between adversarial samples and real samples. The goal of G is to generate samples that can deceive the discriminator into classifying them as real samples, while D aims to correctly differentiate between real and adversarial samples. Through iterative and competitive training of G and D, AdvGAN ultimately obtains convincing adversarial samples to ensure the success of the attack on the target neural network f. It is worth noting that once G is trained, it does not require additional access to the original target network f to generate perturbations for any input data and conduct semi-whitebox attacks.

1.1 Loss function of Generator Assuming a target attack scenario (where the label of the benign sample is b), in order to maximize the misclassification of the manipulated sample's label by the target model f as the target label t, AdvGAN utilizes the loss function L_{adv}^{f} to estimate the likelihood of misleading f. The mathematical expression of L_{adv}^{f} is as follows:

$$L_{adv}^{f} = E_x l_f(x + G(x), t) \tag{1.1}$$

where E_x denotes the expectation value of the input data x, according to the unknown distribution P_{data}. l_f represents the loss function (e.g., cross-entropy loss) used for training the target model f. By continuously optimizing and minimizing L_{adv}^{f}, we can obtain the adversarial sample whose label is closest to the benign label b. At this point, it can be considered that G has been completely trained. Moreover, AdvGAN can also perform untargeted attacks by maximizing the distance between the predicted label and the benign label.

1.2 Loss function of Discriminator For the discriminator part, AdvGAN utilizes the loss function L_{GAN} to measure the similarity between the manipulated data and the real data in D. The mathematical expression of L_{GAN} is as follows:

$$L_{GAN} = E_x log D(x) + E_x log(1 - D(x + G(x))) \tag{1.2}$$

where $E_x log D(x)$ measures the discriminator's ability to accurately predict original samples, expecting the prediction to be close to 1. Similarly, $Exlog(1 - D(G(x)))$ assesses the discriminator's inability to accurately predict generated samples $x + G(x)$, hoping that the prediction is close to 0. Therefore, by maximizing the value of L_{GAN}, the trained D can make the original samples indistinguishable from the adversarial samples.

In order to maximize the discrimination between generated and real samples by D and to bound the magnitude of the perturbation, AdvGAN incorporates the soft hinge loss, building upon some previous research [2, 3, 4].

$$L_{hinge} = E_x max(0, \|G(x)\|_2 - c) \tag{1.3}$$

where $\|\cdot\|_2$ denotes the L_2 norm. c represents a user-specified bound.

1.3 Objective loss function of AdvGAN Considering Eq. 1.1-1.3 comprehensively, the objective loss function of AdvGAN can be expressed as L.

$$L = L_{adv}^{f} + \alpha L_{GAN} + \beta L_{hinge} \tag{1.4}$$

where α and β are hyperparameters that control the relative importance of L_{GAN} and L_{hinge}. As we analyzed in Section. 1.1 and 1.2, we can generate adversarial samples with different labels and similar appearance to the original samples by taking the extreme value of L_{adv}^{f} and L_{GAN}. Therefore, the final trained G and D can be obtained by the following optimization formula:

$$arg \min_{G} \max_{D} L \tag{1.5}$$

References

[1] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.

[2] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, pp. 39–57, Ieee, 2017.

[3] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *arXiv preprint arXiv:1611.02770*, 2016.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.