CHAPTER 11

Overview of adversarial defense

Many techniques have been proposed to improve the robustness of deep neural networks. In this chapter, we focus on methods that try to make neural networks robust against adversarial examples generated by an evasion attack. A defense algorithm can be designed in many different forms. Some defense algorithms modify the training stage of machine learning models to obtain a more robust model, whereas some other algorithms postprocess the model by adding some particular components to improve its robustness. We give an overview of existing techniques and discuss some of the more promising defense approaches in the next few chapters.

11.1 Empirical defense versus certified defense

After the discovering of adversarial examples by Szegedy et al. (2014), many works have been proposed to empirically improve the robustness of neural network models. However, evaluating these defense methods remains tricky. Many earlier works evaluated defense methods only by some simple off-the-shelf attack methods such as FGSM (Goodfellow et al., 2015) or PGD (Kurakin et al., 2017; Madry et al., 2018) attack. However, attack methods provide no formal guarantee on the true adversarial robustness of a defensive model – even if a model is safe evaluated on existing attacks, it is still possible that there still exists a small perturbation to fool the machine learning model, whereas existing attack methods cannot find it. Therefore, without careful empirical evaluations, it is very possible that the defense method is safe against some standard attack methods (e.g., FGSM and PGD) but vulnerable to some other attacks (e.g., attacks particularly developed for attacking the defense model). This may lead to a false sense of security, which indeed happens frequently in the literature. For example, Athalye et al. (2018) showed that most of the defense methods proposed in ICLR 2018, although showing strong defense performance against off-the-shelf attacks, are actually vulnerable under stronger and carefully designed attacks.

To overcome this issue, one approach is to keep improving the attack methods, both in terms of attack performance and their diversity. For example, AutoAttack (Croce and Hein, 2020) gathered a set of diverse and strong

https://doi.org/10.1016/B978-0-12-824020-5.00022-3

attack methods to test the robustness of defensive models. Furthermore, some work also tried to come up with principled way to conduct attack when facing different attack mechanisms (Athalye et al., 2018; Tramer et al., 2020). This is also known as adaptive attack. We will introduce some of the adaptive attack techniques when introducing each particular defense mechanism.

On the other hand, another way to tackle this issue is to study "certified defense" methods. For certified defense, in addition to empirical evaluation of the robustness, we also need to provide a verifiable robustness bound for each example – for each example x, we say the model is certifiably robust on x if x can be predicted correctly and it can be formally verified that any perturbation within the perturbation set cannot change the model's prediction. Certified robust accuracy on a dataset or a data distribution is then defined as the ratio of the examples that can be certifiably robust classified by the model. Certified robust accuracy is a lower bound of robust accuracy, since there could exist examples that can be robustly classified but cannot be easily verified. Certified robust accuracy can usually be computed by some verification methods discussed in the previous sections, and in some particular cases a defense method will have the most suitable verification method for computing certified robust accuracy. In comparison, the "empirical robust accuracy" computed by conducting attacks to the defensive model can only give an upper bound of true robust accuracy, since attack methods fail to find an adversarial example; does not imply that an adversarial example does not exist.

Although certified defense methods are much safer and more reliable, currently they also encounter a serious drawback that existing certified methods lead to very poor certified robust accuracy. For instance, on CIFAR-10 classification with 8/255 ℓ_∞ ball perturbation, a standard deep learning model can achieve over 90% clean accuracy; a good empirical defense model can often achieve over 80% clean accuracy and over 60% empirical robust accuracy; however, the best certified defense methods can only achieve around 50% accuracy and around 40% certified robust accuracy. Therefore there are pros and cons of empirical defense versus certified defense, and we will introduce both of them in the following sections.

11.2 Overview of empirical defenses

In the following, we will categorize some popular defense methods into the following categories, including adversarial training, randomization, detection, filtering/projection, discrete components, and adversarial detection. Among them, adversarial training, randomization and detection has demonstrated more promising results in the literature, so we will introduce those methods in detail in the following chapters.

Adversarial training. To combat adversarial examples, a natural idea is to add adversarial examples in the training set. This leads to the most widely used defense methods known as adversarial training. Even in one of the first papers talking about adversarial examples, Goodfellow et al. (2015) already tried to train the model on adversarial examples generated by Fast Gradient Sign Method (FGSM). Later on, Kurakin et al. (2017) suggested using a multistep FGSM to further improve adversarial robustness. However, these earlier works often generate adversarial examples periodically, and the robustness improvements are not stable. Instead, Madry et al. (2018) showed that adversarial training can be formulated as a min-max optimization problem: given the current neural network model, an attacker generates adversarial examples to maximize the classification loss, and the learner aims to update the model to minimize the classification loss on those adversarial examples. With this clean min-max formulation, the adversarial examples can be generated on-the-fly. In ICLR 2018, many defense methods were proposed including this min-max adversarial training method. When Athalye et al. (2018) tried to develop stronger attacks to evaluate the defense methods published in ICLR 2018, they showed that adversarial training is the only approach that still performs well under stronger attacks, whereas many other methods have almost 0% accuracy under stronger attacks. After that, adversarial training becomes widely used, and many variations of adversarial training have been developed. We will introduce them in Chapter 12.

Randomization. Another effective defense method is to add some randomized components into the model. Intuitively, randomness in the model can make it harder for the attacker to create a fixed attack to fool the model. Therefore randomization has been proposed as heuristics to improve the robustness empirically (Xie et al., 2017; Dhillon et al., 2018; Liu et al., 2018c, 2019d). Although some earlier developed randomization methods are based on heuristics, several theoretical explanations have been developed, showing that randomness can truly boost the robustness of neural networks (Lecuyer et al., 2019; Liu et al., 2020c). We will give more details in Chapter 13.

Detection. The task of making a classifier robust can be more challenging as the input space is very high dimensional, and it is difficult to make classifier correct in all the input region. Therefore another natural way to combat adversarial examples is developing a method to detect "abnormal" examples where the classifiers would not need to make predictions on those examples. To detect adversarial examples, methods have been developed including simple methods that measure the distance between adversarial example and natural images (Lee et al., 2018) and some more complicated methods that detect adversarial examples by feature attribution (Yang et al., 2020d). We will introduce several representative methods in Chapter 15. However, an important problem is how to evaluate the performance of a detection model – it is possible that an attacker can leverage the information about the detecting model to improve their successful rate by trying to bypass the detectors. Therefore it is important to try to evaluate the whole model, jointly including both detector and classifier, when conducting experiments on detecting models.

Filtering and projection. It is commonly believed that adversarial examples fall outside the natural image manifold, for which DNNs are poorly trained. This leads to a natural line of defense methods to project the adversarial examples back to the natural image manifold, or in another words, filter out the "unnatural noises" added into the example. To capture the natural image manifold, most of the approaches in this category adopt some kind of generative models (e.g., some version of autoencoder or generative adversarial networks). Since generative models are trained on natural examples, adversarial examples will be projected to the manifold learned by the generative model. Furthermore, "projecting" the adversarial examples onto the range of the generative model can have the desirable effect of reducing the adversarial perturbation.

Meng and Chen (2017) trained an autoencoder to capture the natural image manifold. An autoencoder is a type of neural network that consists of two major parts, the encoder that maps the input to a low-dimensional space and the decoder that recovers the input from the low-dimensional embedding. The autoencoder is usually trained on the reconstruction loss with respect to the input. Therefore the high-dimensional data are summarized by the low-dimensional embedding through the training process. In MagNet, one autoencoder is chosen at random attesting time to filter the input samples, and thus adversarial perturbation could potentially be removed through this encoding and decoding process. Instead of using autoencoders, Samangouei et al. (2018) proposed to train a generative adversarial network (GAN) to capture the natural image manifold and then use it in the inference stage to filter out unnatural noise in the image. Unfortunately, both MegNet and Defense-GAN are shown to be nonrobust when the attacker can jointly attack the filtering step and the classifier (Athalye and Sutskever, 2018).

More recently, Li et al. (2018b) proposed a different defense framework, termed *ER-Classifier*, which combines the process of filtering and classification in a single joint framework. In fact, any deep classifier can be viewed as a combination of these two parts, an encoder part to extract useful features from the input data and a classifier part to perform classification based on the extracted features. Both the encoder and the classifier are neural networks. ER-Classifier is similar to a regular deep classifier, which first projects the input to a low-dimensional space with an encoder $\boldsymbol{G}$ and then performs classification based on the low-dimensional embedding with a classifier $\boldsymbol{C}$. The main different is that at the training stage, the low-dimensional embedding of ER-Classifier is stabilized with a discriminator $\boldsymbol{D}$ by minimizing the dispersion between the distribution of the embedding and the distribution of a selected prior. The goal of the discriminator is separating the true code sampled from a prior and the "fake" code produced by the encoder, whereas the encoder will try to produce generated code that is similar to the true one. The result of this competition is that the distribution of the embedding space will be pushed toward the prior distribution. Therefore it is expected that this regularization process can help remove the effects of any adversarial distortion and push the adversarial examples back to the natural data manifold. Another difference is that the embedding space dimension is much smaller for the ER-classifier, when compared with a general deep classifier, making it easier for the training process to converge. In this framework the projection is used as a regularization to improve robustness, and the method can be used jointly with other defense (e.g., adversarial training) to further improve the performance.

Discrete components. Since most of the attack algorithms, such as PGD and C&W attacks, rely on the gradient computation, several defense methods try to introduce some discrete and nondifferentiable components into the neural network model to make it harder for gradient-based attacks. For instance, Papernot and McDaniel (2018) and Dubey et al. (2019) adopt a nearest-neighbor finding component in the neural network to project images to the nearest neighbors in the database, which creates difficulty in conducting attacks. Further, some nondifferential activation functions have been proposed by Xiao et al. (2019a). Moreover, it has been shown that voting-based ensembles can improve robustness against adversarial examples (Khatri et al., 2020). However, due to the difficulties of conducting attacks on the defense components, it is harder to reach a conclusion whether a discrete component can truly make models more robust or there exist non-gradient-based attack methods to break these defenses.