Defenses Against Adversarial Attacks

Haotian Wang

Ph.D. Student

University of Idaho

Computer Science

Outline

- **Introduction**
- Defense against Adversarial Attack Methods
    - Gradient Masking/Obfuscation
    - Robust Optimization
    - Adversarial Examples Detection

Xu et al. "Adversarial attacks and defenses in images, graphs and text: A review." *International Journal of Automation and Computing* 17, no. 2 (2020): 151-178.

ADVERSARY’S KNOWLEDGE

- Defense against Adversarial Attack. The goal of the defense is to build machine learning models which are robust to adversarial examples, i.e., can classify adversarial images correctly.

Cited: NIPS 2017 - Defense Against Adversarial Attack workshop

Non-targeted Adversarial Attack. The goal of the non-targeted attack is to slightly modify a source image in a way that the image will be classified incorrectly by a generally unknown machine learning classifier.

Targeted Adversarial Attack. The goal of the targeted attack is to slightly modify a source image in a way that the image will be classified as specified target class by a generally unknown machine learning classifier.

Proactive defense: The defenses are constantly looking for potential attackers.

Reactivate defense: The defenses react on current attacks.

Black-box attack: The design, or the parameters of the models are not known. The type of attack when the inner working is not available is called black-box attack.

White-box attack: The design, or the parameters of the models are known. The type of attack when the inner working is available is called white-box attack.

Grey-box attack/semi-white box attack: The attacker trains a generative model for producing adversarial examples in a white-box setting.

Outline

- Introduction
- **Defense Against Adversarial Attack Methods**
  - Gradient Masking/Obfuscation
  - Robust Optimization
  - Adversarial Examples Detection

Xu et al. "Adversarial attacks and defenses in images, graphs and text: A review." *International Journal of Automation and Computing* 17, no. 2 (2020): 151-178.

Three main defense strategies:

- Gradient Masking/Obfuscation. Since most attack algorithms are based on the gradient information of the classifier, masking or hiding the gradients will confound the adversaries.
- Robust Optimization. Re-learning a DNN classifier’s parameters can increase its robustness. The trained classifier will correctly classify the subsequently generated adversarial examples.
- Adversarial Examples Detection. It studies the distribution of natural/benign examples, detects adversarial examples, and disallows their input into the classifier.

Gradient Masking/Obfuscation

- **DEFENSIVE DISTILLATION**
- SHATTERED GRADIENTS
- STOCHASTIC/RANDOMIZED GRADIENTS
- EXPLODING & VANISHING GRADIENTS

DEFENSIVE DISTILLATION

Papernot et al. "Distillation as a defense to adversarial perturbations against deep neural networks." In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582-597. IEEE, 2016. 1476 cites Penn State and US Army lab

Problem: Deep learning is vulnerable to adversarial samples.

Method: The authors introduced a defensive mechanism, namely, *defensive distillation* to reduce the effectiveness of adversarial samples on DNNs. The algorithm is based on gradient obfuscation.

Results: The defensive distillation reduced the success rate of adversarial sample crafting [1].

MNIST: 95.89% -> 0.45%

CIFAR10: 87.89% -> 5.11%

[1] Papernot et al. "The limitations of deep learning in adversarial settings."
In *EuroS&P*, pp. 372-387. 2016.

DISTILLATION

The concept of ‘Distillation’ was first introduced by Hinton [1].

[1] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in Deep Learning and Representation Learning Workshop at NIPS 2014.

DISTILLATION

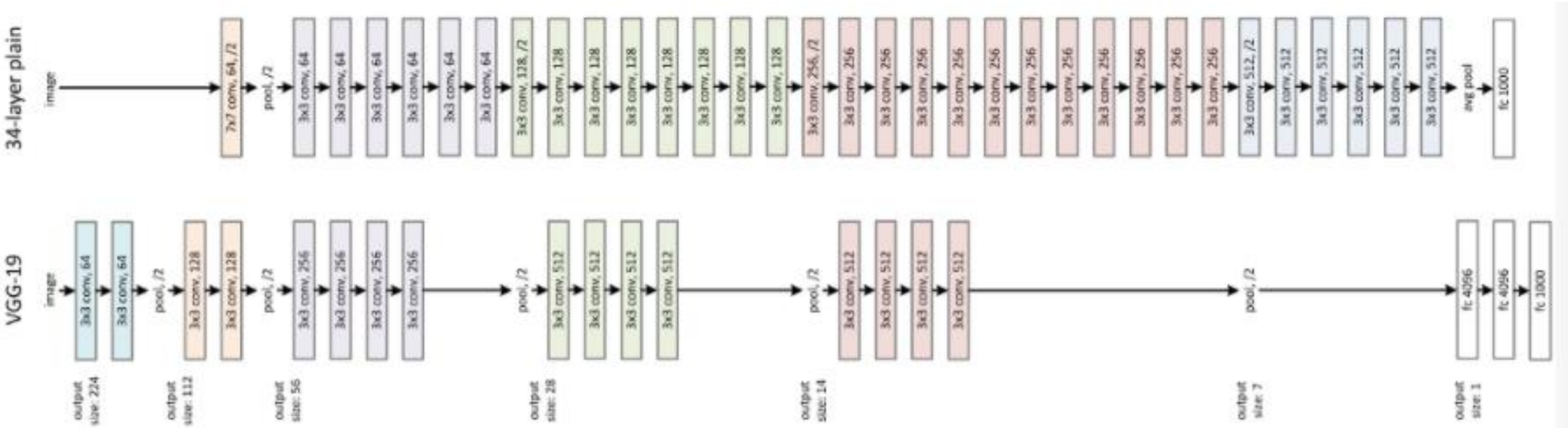
- **Knowledge distillation:** is the process of transferring knowledge from a large model to a smaller one, and achieving similar results. The general intuition behind this technique is to extract class probability vectors produced by a first DNN to train a second DNN of reduced dimensionality, without loss of accuracy. It can be computationally expensive to evaluate a model on a resource constrained device, e.g. phone.
- **Transfer learning (TL):** focuses on storing knowledge gained while solving one problem and applying it to a different, but related problem.

Motivation: Distillation is motivated by the goal of reducing the size of DNN architectures or ensembles of DNN architectures for reducing the computing cost.

DISTILLATION

Network 1

Network 2



Use the prediction results (soft labels) from the first network to train the second network. The second network has a *distillation temperature* [1] mechanism on the softmax function for achieving a comparable results to the first network.

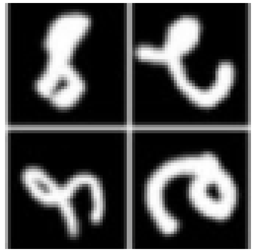
- If it is trained properly, the second network can achieve the same accuracy even though it has a smaller capacity, and improves the generalization.

DISTILLATION

Benefits: when we train a DNN, we use **hard labels**, i.e., we assign 100% probability to the “ground truth” and 0% to the other labels.

In reality, information has its uncertainty. After the training, the output of the DNN model is a probability distribution (say, 0.1, 0.02, …, 0.05).

The distribution actually captures better information on uncertainty and models it better for many real problems.



Defense Distillation

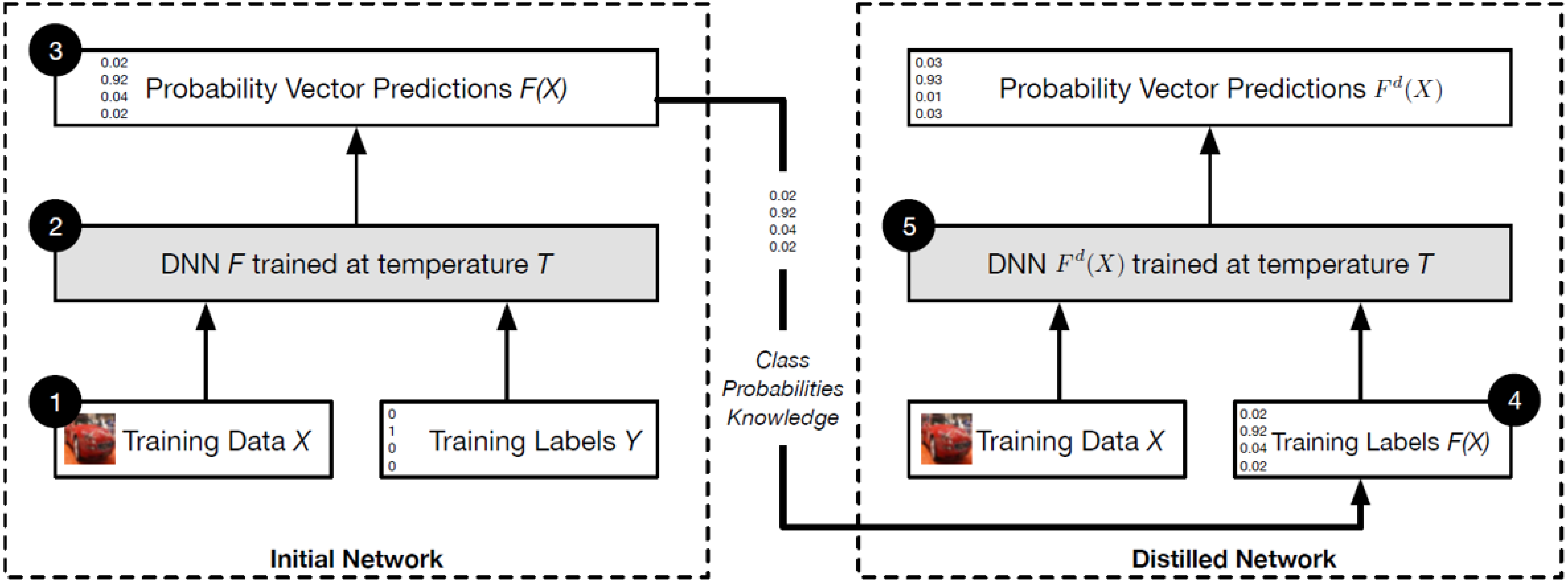

3
0.02
0.92
0.04
0.02
Probability Vector Predictions *F(X)*
2
DNN *F* trained at temperature *T*
1
Training Data *X*
0
1
0
0
Training Labels *Y*
Initial Network
0.02
0.92
0.04
0.02
Class Probabilities Knowledge
0.03
0.93
0.01
0.03
Probability Vector Predictions $F^d(X)$
5
DNN $F^d(X)$ trained at temperature *T*
Training Data *X*
4
0.02
0.92
0.04
0.02
Training Labels *F(X)*
Distilled Network

Fig. 5: **An overview of our defense mechanism based on a transfer of knowledge contained in probability vectors through distillation:** We first train an initial network F on data X with a softmax temperature of T. We then use the probability vector $F(X)$, which includes additional knowledge about classes compared to a class label, predicted by network F to train a distilled network F^d at temperature T on the same data X.

Gradient Masking/Obfuscation

- DEFENSIVE DISTILLATION
- **SHATTERED GRADIENTS**
- STOCHASTIC/RANDOMIZED GRADIENTS
- EXPLODING & VANISHING GRADIENTS

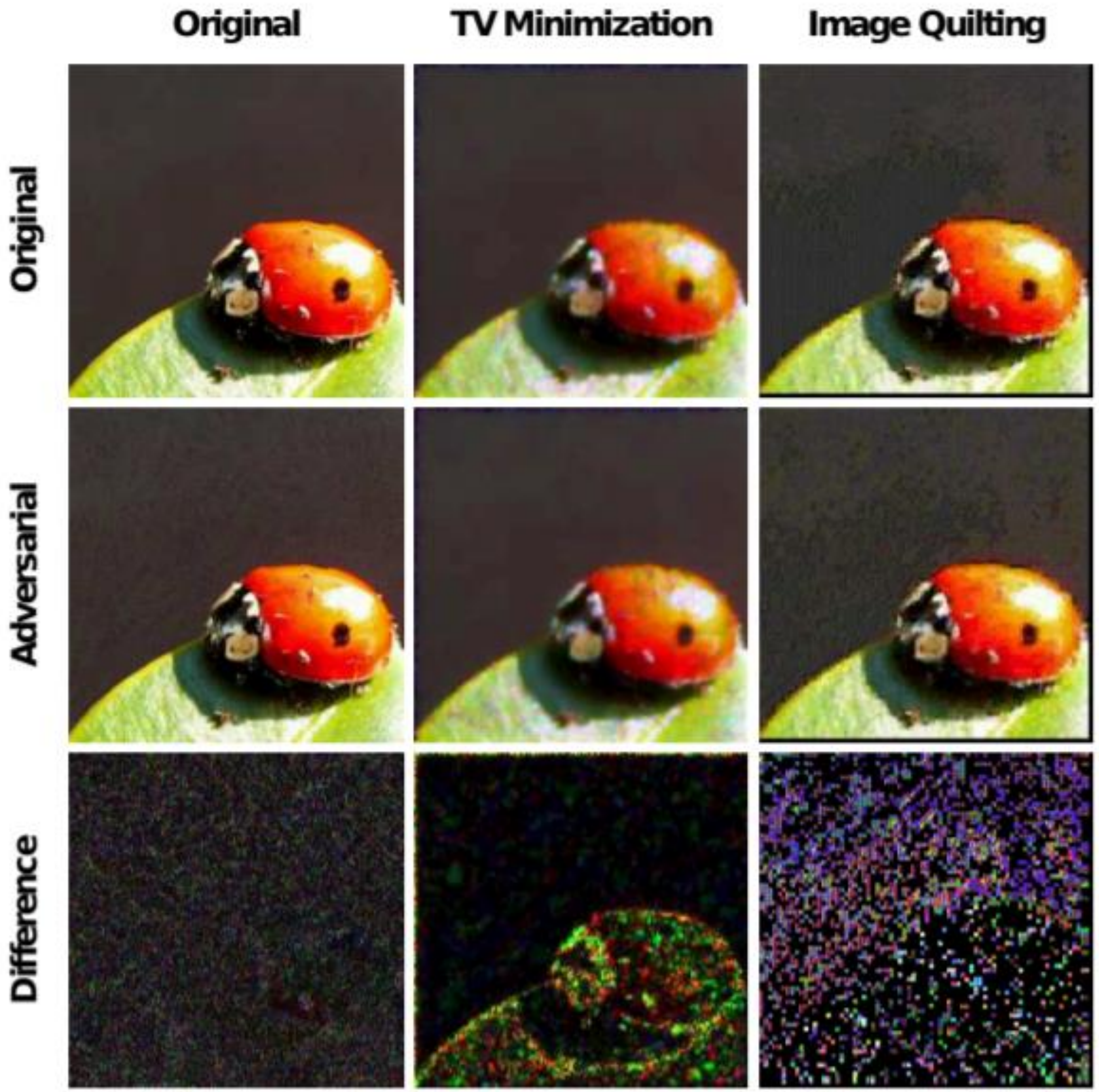
SHATTERED GRADIENTS

Concept: **The algorithm protects the model by preprocessing the input data.**

For example, adding a non-smooth or non-differentiable preprocessor g(.) and then train a DNN model f on g(X). The trained classifier f(g(.)) is not differentiable in term of X, causing failure of adversarial attacks.

Guo [1] studied a number of image processing tools, such as image cropping, compressing, total-variance minimization, bit-depth reduction, image quilting, to determine whether these techniques can help to protect the models against adversarial examples.

[1] Guo et al. "Countering adversarial images using input transformations." *arXiv preprint arXiv:1711.00117* (2017). 497 cites



Results show that the total variance minimization and image quilting are efficient for defending against attacks.

SHATTERED GRADIENTS - Denoiser

One defense strategy is to restore the adversarial examples closer to the originals examples, and remove the added manipulation — a.k.a., denoising.

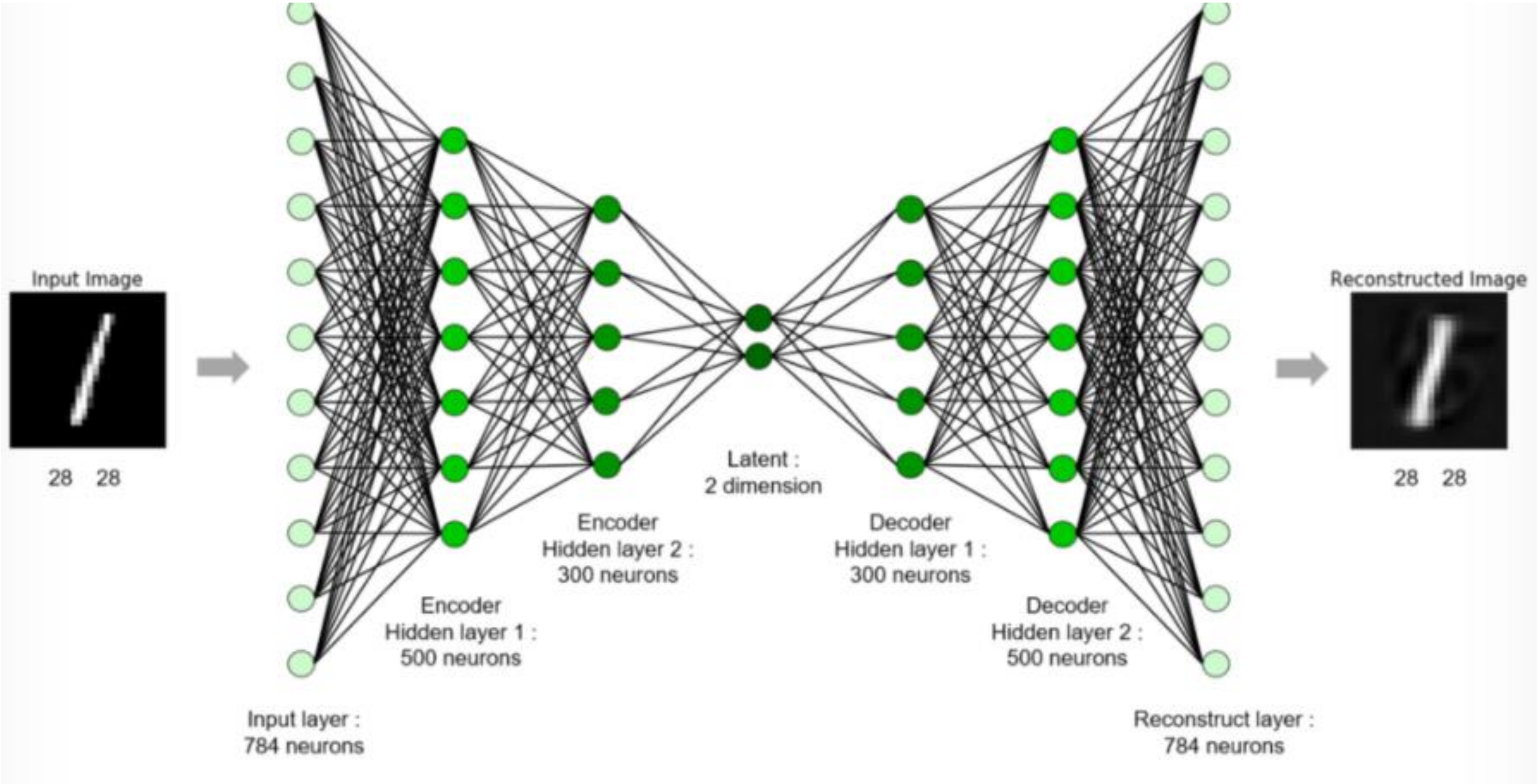
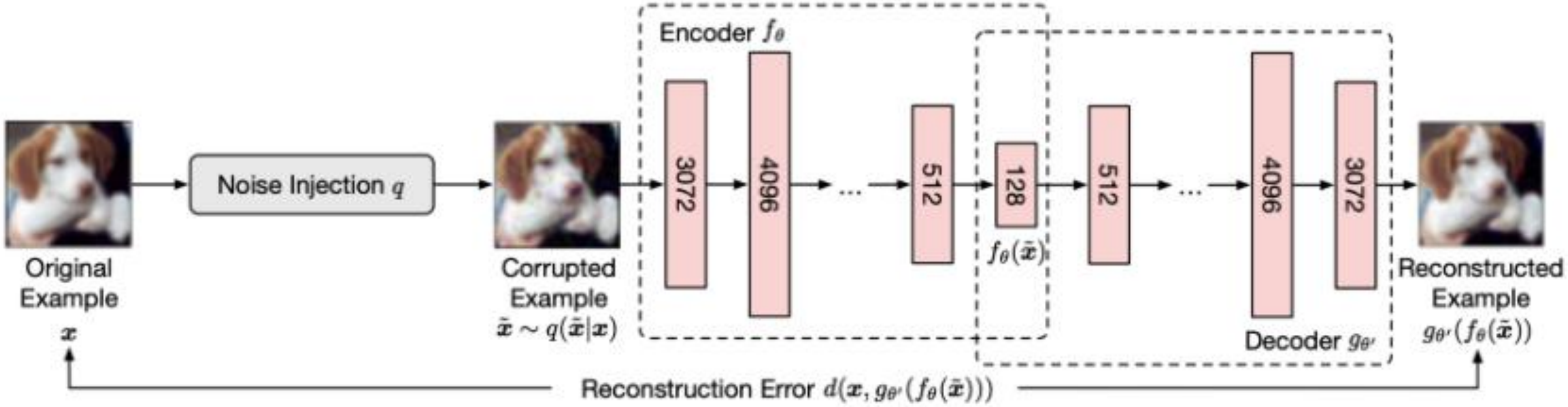


Fig. Autoencoder

Denoiser

Liu presents a method that utilizes an autoencoder architecture to restore the original image. The reconstructed image will have the manipulated data removed (theoretically). Then, feed the denoised data into the model.



Liu et al. "Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness." In MASS, pp. 274-282. IEEE, 2019.

Gradient Masking/Obfuscation

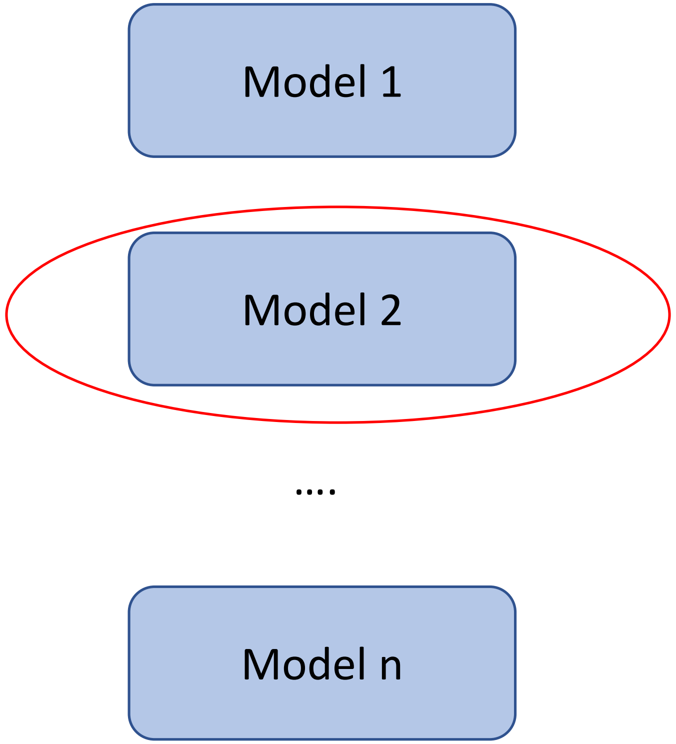
- DEFENSIVE DISTILLATION
- SHATTERED GRADIENTS
- **STOCHASTIC/RANDOMIZED GRADIENTS**
- EXPLODING & VANISHING GRADIENTS

RANDOMIZED GRADIENTS

Try to randomize the DNN model in order to confound the adversary.

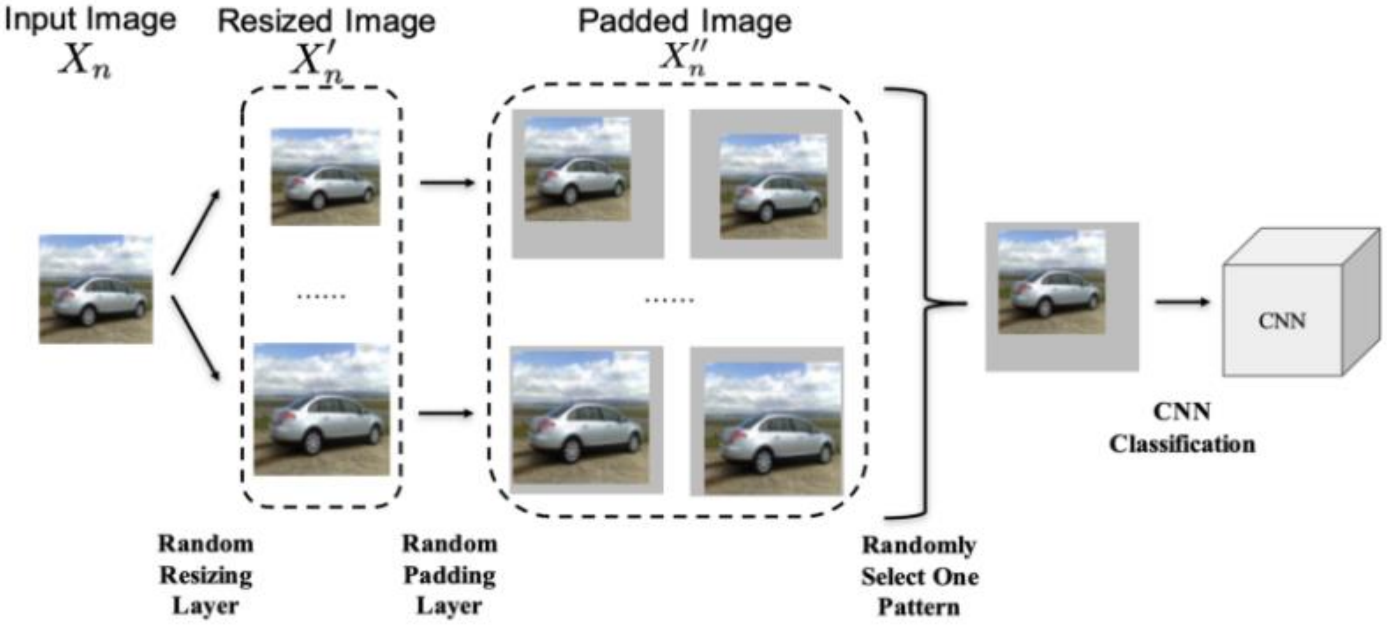
RANDOMIZED GRADIENTS

- Prepare multiple classifiers, and randomly select one to predict.



RANDOMIZED GRADIENTS

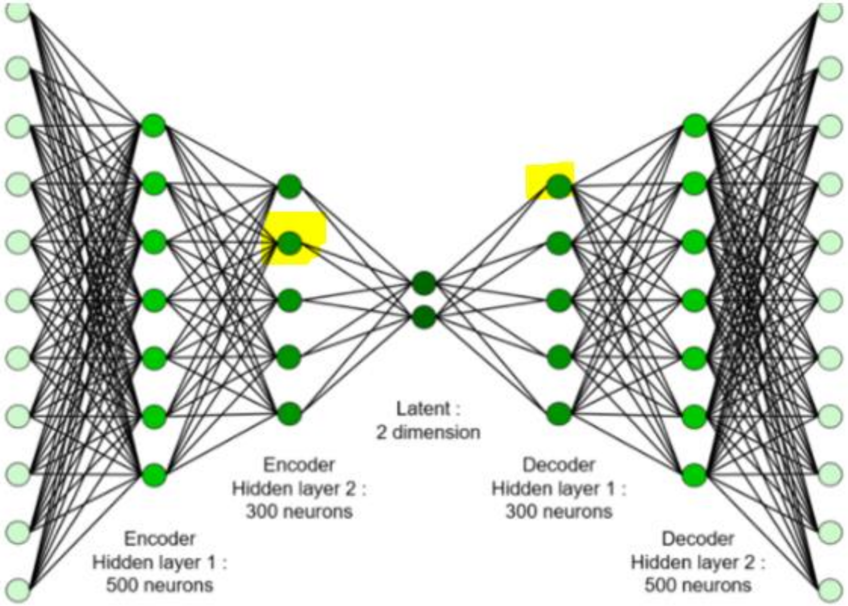
In Xie’s approach, the images were resized to a random size and padded with zeros around the input image. (Ranked 2nd in 2017 NIPS defense AML competition among 200 teams)



Xie, Cihang, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. "Mitigating adversarial effects through randomization." ICLR 2018. 368 cites

STOCHASTIC GRADIENTS

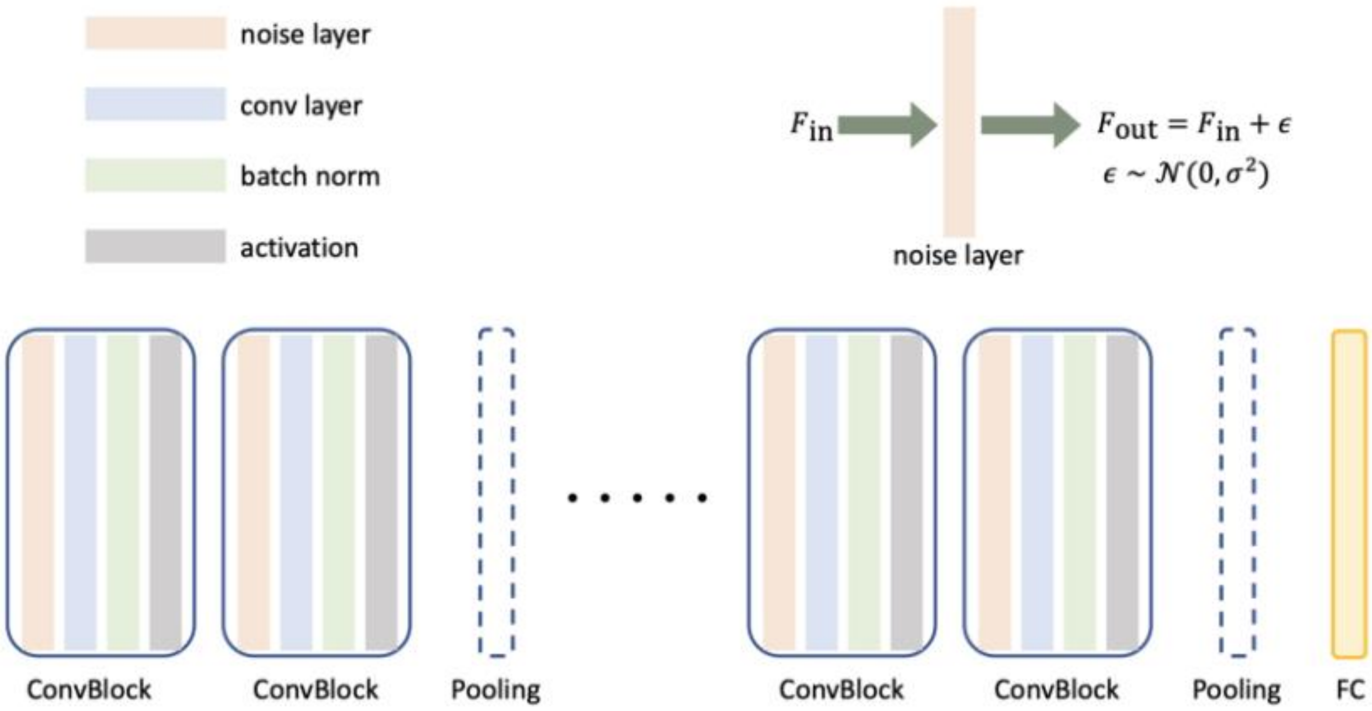
In Dhillon’s approach, during inference, random nodes in the DNN models are dropped.



Dhillon et al. "Stochastic activation pruning for robust adversarial defense." ICLR 2018. 237 cites

STOCHASTIC GRADIENTS

Liu’s approach adds noise layers before the convolution layers to perturb the inputs to the CNN layers.



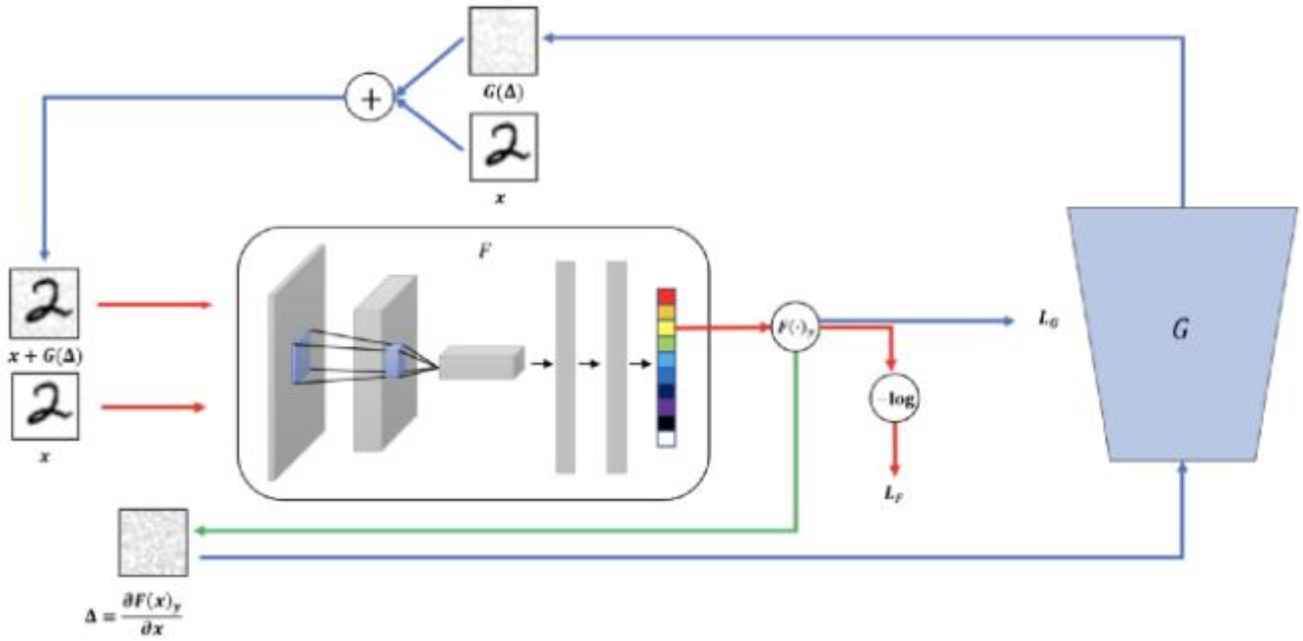
Liu, Xuanqing, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. "Towards robust neural networks via random self-ensemble." In ECCV, pp. 369-385. 2018. 137 cites

Gradient Masking/Obfuscation

- DEFENSIVE DISTILLATION
- SHATTERED GRADIENTS
- STOCHASTIC/RANDOMIZED GRADIENTS
- **EXPLODING & VANISHING GRADIENTS**

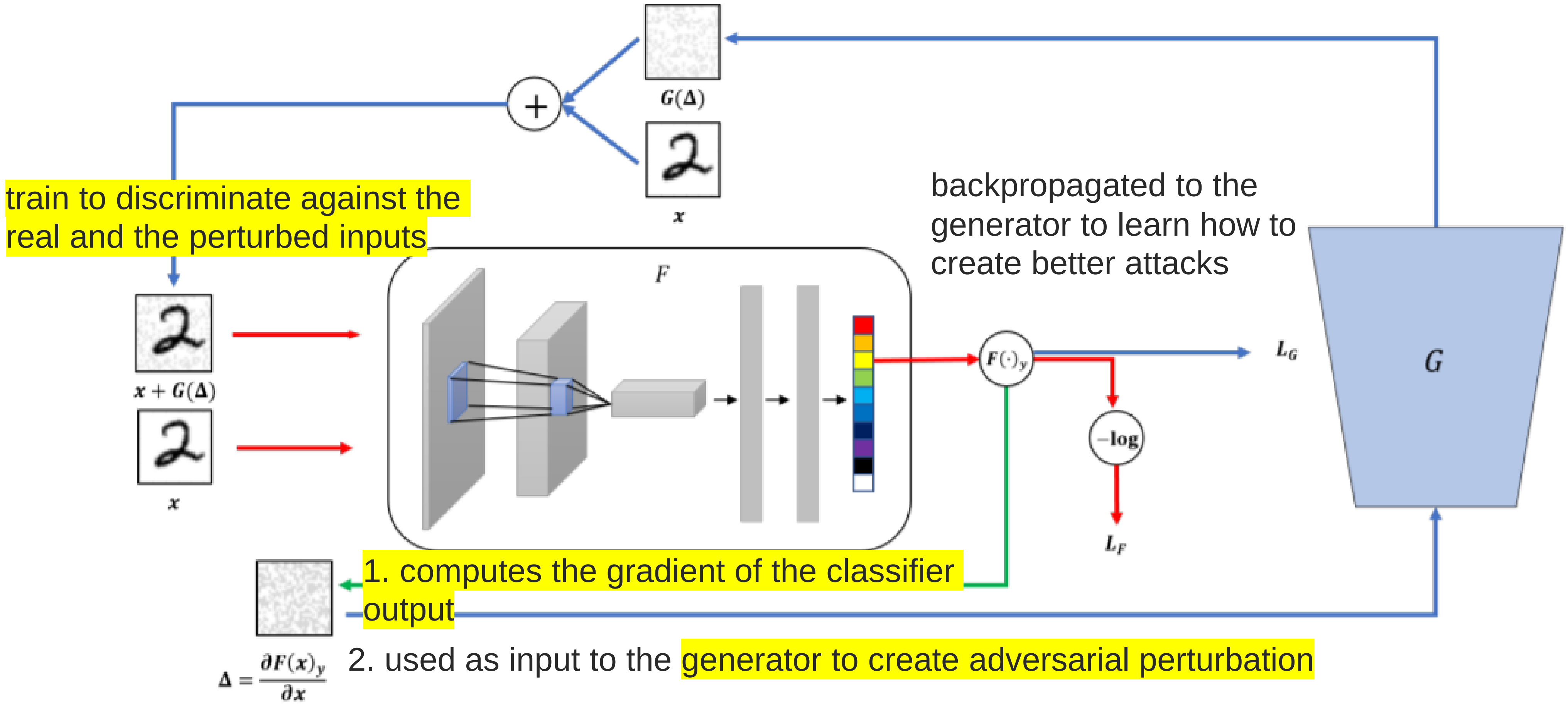
EXPLODING & VANISHING GRADIENTS

Using GAN to generate a better classifier (discriminator) in detecting adversarial samples.



Lee, Hyeungill, Sungyeob Han, and Jungwoo Lee. "Generative adversarial trainer: Defense to adversarial perturbations with GAN." *arXiv preprint arXiv:1705.03387* (2017). 82 cites

Both the classifier and the generator are trained in step to improve each other and eventually the classifier will get better and better in discriminating adversarial samples.



train to discriminate against the real and the perturbed inputs

backpropagated to the generator to learn how to create better attacks

1. computes the gradient of the classifier output

2. used as input to the generator to create adversarial perturbation

Figure 1: **Adversarial training with Generative Adversarial Trainer:** (1) Generative Adversarial Trainer G is trained to generate an adversarial perturbation that can fool the classifier network using the gradient of each image. (2) Classifier Network F is trained to classify correctly both original and adversarial examples generated by G.

Weakness of Gradient Masking/Obfuscation

- The main weakness of the gradient masking strategy is that: it can only “confound” the adversaries, but it cannot eliminate the existence of adversarial examples.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE, 2758 cites

Robust Optimization

- **ADVERSARIAL (RE)TRAINING**
- PROVABLE DEFENSES
- REGULARIZATION METHODS

Adversarial (Re)training

Concept: Feeding generated adversarial examples when training a model.

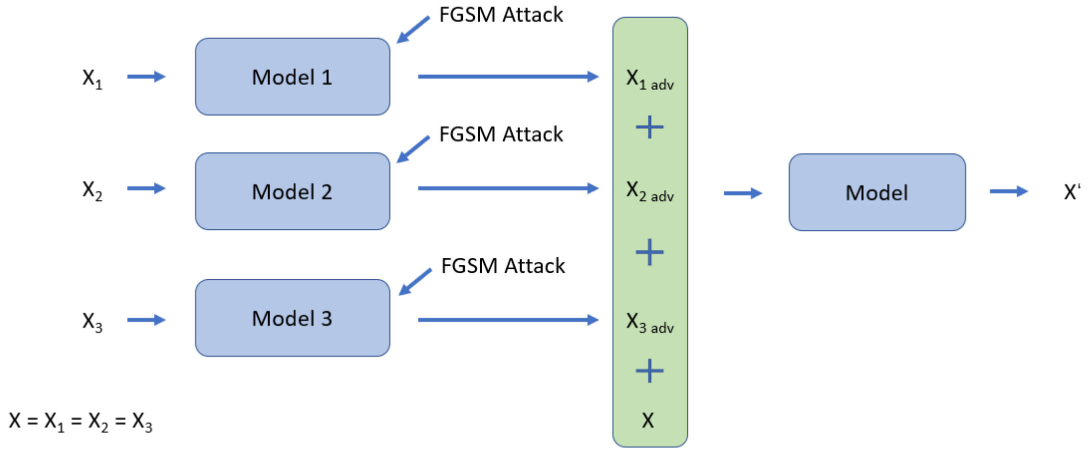
Adversarial (Re)training

Tramèr, Florian, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. "Ensemble adversarial training: Attacks and defenses." *arXiv preprint arXiv:1705.07204* (2017). 1027 cites
Google Brain Group

Ensemble Adversarial Training

Method: The authors introduced an adversarial training method based on **Transferability** in Adversarial Images. It protects CNN models against single-step attacks and can be also applied to large datasets, such as ImageNet.

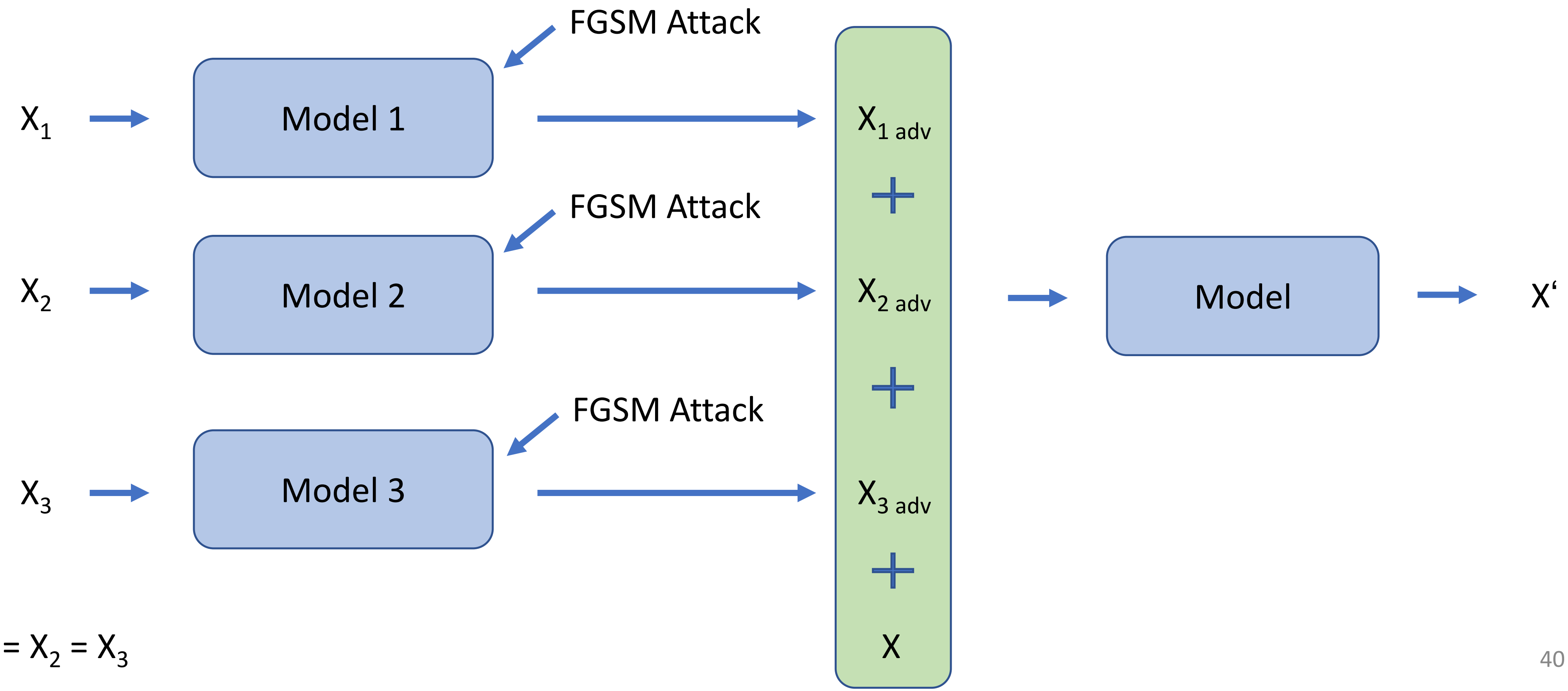
Results: The model won the first round of the NIPS 2017 competition on Defenses against Adversarial Attacks



Transferability in Adversarial Images

Concept: The adversarial images generated to target one victim model also have a high probability of misleading other models.

The ensemble adverbial training is based on the property of Transferability in adversarial images. The main approach is to augment the classifier's training set with adversarial examples crafted from other pre-trained classifiers.



$X = X_1 = X_2 = X_3$

Robust Optimization

- ADVERSARIAL (RE)TRAINING
- **REGULARIZATION METHODS**
- PROVABLE DEFENSES

Regularization Method

Zhang et al. "Theoretically principled trade-off between robustness and accuracy." *arXiv preprint arXiv:1901.08573* (2019). 333 cites CMU, Berkeley

Theoretically principled trade-off between robustness and accuracy

Problem: The trade-off between DNNs robustness and accuracy.

Method: The authors proposed a surrogate-loss that evaluates the trade-off between robustness against accuracy.

Results: Won the 1st place out of ~2,000 submissions on NeurIPS 2018 Adversarial Vision Challenge.

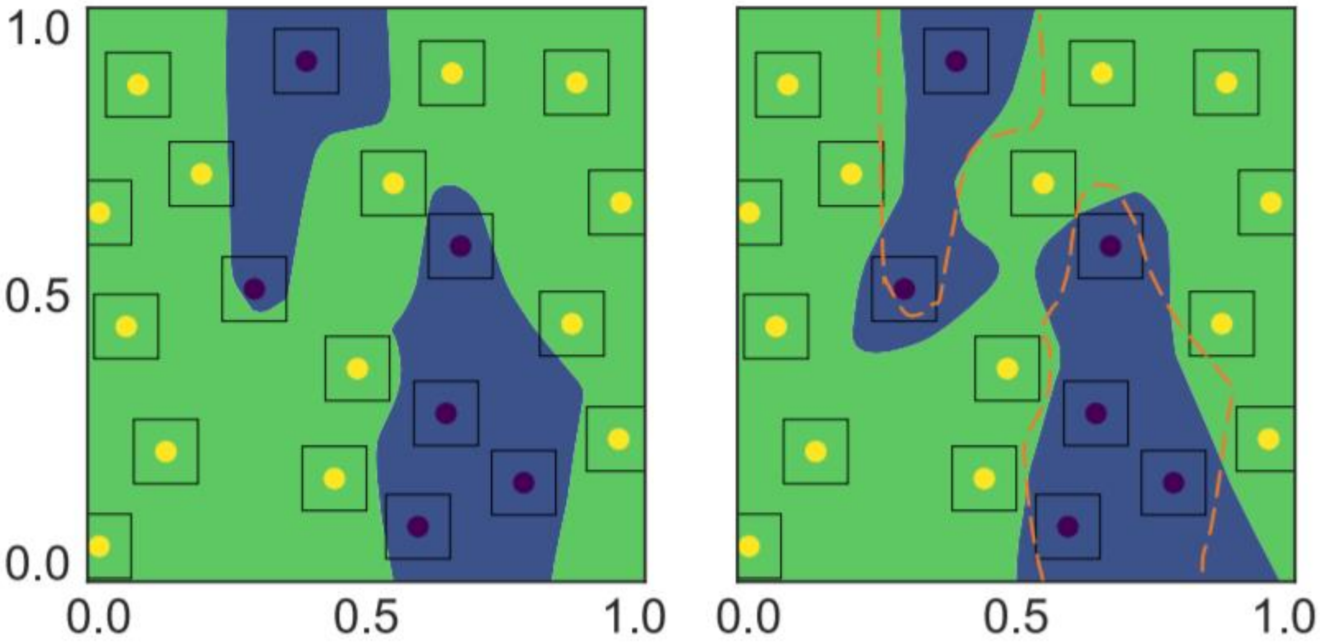


Figure 1: **Left figure:** decision boundary learned by natural training method. **Right figure:** decision boundary learned by our adversarial training method, where the orange dotted line represents the decision boundary in the left figure. It shows that both methods achieve zero natural training error, while our adversarial training method achieves better robust training error than the natural training method.

Robust Optimization

- ADVERSARIAL (RE)TRAINING
- REGULARIZATION METHODS
- **PROVABLE DEFENSES**

PROVABLE DEFENSES

Problem: There is no formal guarantee about the safety of the trained classifiers for defending against attacks. We will never know whether there are more aggressive attacks that can break those defenses. It's not responsible to apply those strategies to safety-critical tasks.

Approach: Develop an algorithm to verify the robustness of DNNs architecture.

PROVABLE DEFENSES

Raghunathan, Aditi, Jacob Steinhardt, and Percy Liang. "Certified defenses against adversarial examples." ICLR 2018. 423 cites Stanford

Certified defenses against adversarial examples

The authors used integration inequalities to derive a certificate and they applied semidefinite programming (SDP) [1] to solve the certificate. Then, they optimized a certain value to encourage robustness against attacks.

Semidefinite programming (SDP) is a convex optimization method.

[1] "Semidefinite programming." In *Interior point methods of mathematical programming*, pp. 369-398. Springer, Boston, MA, 1996.

Adversarial Example Detection

The approaches distinguish whether the input is benign or adversarial. Then, the classifier will refuse to make predictions on adversarial images.

Gong, Zhitao, Wenlu Wang, and Wei-Shinn Ku. "Adversarial and clean data are not twins." *arXiv preprint arXiv:1704.04960* (2017). 139 cites

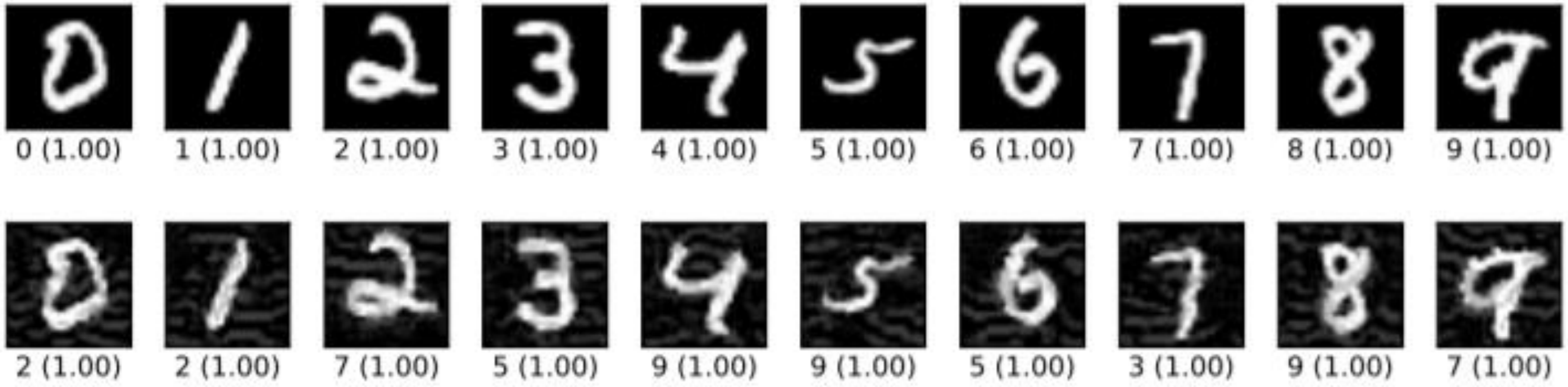


Figure 1. The adversarial images (second row) are generated from the first row via iterative FGSM. The label of each image is shown below with prediction probability in parenthesis. Our model achieves less then 1% error rate on the clean data.

Train a 2-layer CNN to discriminate between adversarial images and clean images.

Q&A