Homework 1: Gray-box Attack

Name: Chou Tun Hsiang **Student Number:** B07902050 **Date:** October 23 2020

1 Methodology

The momentum iterative fast gradient sign method is taken to generate adversarial examples. The memorization of previous gradients helps to get through poor local minimum or maximum.

Basically, the gradient based methods solved the following optimization problem

$$\arg\max_{x^*} J(x^*, y), \;\; \text{s.t. } ||x^* - x||_\infty \leq \epsilon \tag{1}$$

where J is the loss function, usually the cross-entropy loss, x^* is the generated adversarial example, and ϵ is the size of adversarial perturbation. A simple yet effective white-box attack fast gradient sign method (FGSM) was proposed, with the assumption of linearity around the decision boundary of the data point. In this method, the adversarial example can be obtained by the equation

$$x^* = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y)) \tag{2}$$

The gradients are taken with respect to the original image in order to maximize the loss. The method works pretty fast since it is a one-step approach. To achieve a stronger white-box attack, one may consider another approach, the iterative fast gradient sign method (I-FGSM).

$$x_0^* = x, \;\; x_{t+1}^* = x_t^* + \alpha \cdot \text{sign}(\nabla_x J(x_t^*, y)) \tag{3}$$

where α equals ϵ divided by number of iterations, in order to meet the perturbation criterion. The equation expresses that FGSM approach is taken many times. However, this method may sacrifice the transferability, that is, the ability to successfully attack deep neural networks of different architectures.

Then it comes to the third method, momentem iterative FGSM (MI-FGSM), which stabilizes update directions and escape from poor local maxima by accumulating a velocity vector in the gradient direction of the loss function. From the discussion above we can see that the FGSM method has a good transferability at the cost of lower attack success rate to the white-box model, while the MI-FGSM can attain a stronger attack, yet suffer from overfitting. The MI-FGSM method alleviates the trade-off between the attack ability and transferability. The generated examples remained adversarial to the white-box model as the number of iterations increases, while keeping the transferability like the FGSM method. The detailed steps are described in Algorithm 1.

The MI-FGSM method can also be applied to ensemble models to help boosting adversarial attack. The main idea is to fuse the output logits, the raw prediction score before softmax process, of the models together.

$$\ell(x) = \sum_{k=1}^{K} w_k \ell_k(x) \;\; \text{subject to } \; w_k \geq 0, \; \sum_{k=1}^{K} w_k = 1 \tag{6}$$

where ℓ_k are the logits of the k^{th} model and w_k is the ensemble weight. Then we define the loss function as

$$J(x, y) = -1_y \cdot \log\Big(\text{softmax}(\ell(x))\Big) \tag{7}$$

Preprint. Under review.

Algorithm 1: MI-FGSM

Data: A classifier f with loss function J; a real example x and ground-truth label y.
Input: The size of perturbation ϵ, iterations T and decay factor μ.
Output: An adversarial example x^* with $||x^* - x||_\infty \leq \epsilon$.

$\alpha = \epsilon/T$;
$g_0 = 0$; $x_0^* = x$;
for $t = 0$ *to* $T - 1$ **do**
  Input x_t^* to f and obtain the gradient $\nabla_x J(x_t^*, y)$;
  Update g_{t+1} by accumulating the velocity vector in the gradient direction as

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{||\nabla_x J(x_t^*, y)||_1} \tag{4}$$

  Update x_{t+1}^* by applying the sign gradient as

$$x_{t+1}^* = x_t^* + \alpha \cdot \text{sign}(g_{t+1}) \tag{5}$$

end

where 1_y is the one-hot encoding of y. Though there are actually several ways to combine the outputs of different models, this was proved to be the most effective one to generate adversarial examples.

2 Experiments

Adversarial examples are generated by MI-FGSM using a single model as well as ensemble method. A range of decay factor μ is also tested with ensemble adversarial training. The result shows that the ensemble adversarial training approach achieves a higher black-box attack success rate on the provided 100 images evaluation dataset.

Five models are included in the first experiment: ResNet, PreResNet, ResNeXt, SE-ResNet, and DenseNet. All models are pre-trained on cifar-10 dataset. DenseNet acts as the black-box model, so as to evaluate the transferability of MI-FGSM training with one or several models. ResNet is used as the single MI-FGSM training model, while ResNet, PreResNet, ResNeXt, and SE-ResNet are chosen in the ensemble method. The max perturbation ϵ is set to 8, the decay factor μ set to 1.0, and the number of iterations T set to 10. In the ensemble MI-FGSM method, weights are identical during logits fusion for all models. Table 1 shows the detailed result.

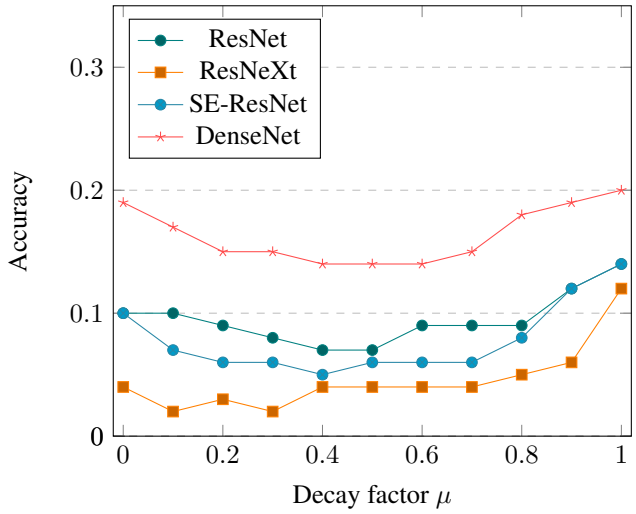
Table 1: Accuracy of models on different datasets

	Images		
	Original images	Single model MI-FGSM	Ensemble MI-FGSM
ResNet	0.97	0.04	0.14
PreResNet	0.97	0.28	0.21
ResNeXt	1.00	0.52	0.12
SE-ResNet	0.98	0.29	0.14
DenseNet	0.98	0.22	0.20

It is not appropriate to compare the predictions of PreResNet, ResNeXt, SE-ResNet with single model and ensemble MI-FGSM directly to say that the latter has a better result, since it would be the comparison between white-box and black-box attack. Apparently, the ensemble method obtains additional information about the models, thus achieve a stronger attack. However, considering the black-box model, DenseNet, ensemble MI-FGSM indeed has a slightly better performance, while single model MI-FGSM significantly reduce the accuracy rate of ResNet. In a real world application, or at least in this assignment, the final evaluation models remained unknown. Hence it is worthwhile taking various neural network structures into account using ensemble MI-FGSM.

The next experiment is used to explore the relationship between the accuracy, the decay factor μ, which controls how much previous gradients contributes to the update direction in the current iteration. The ensemble MI-FGSM is trained by ResNet, PreResNet, ResNeXt, SE-ResNet to generate adversarial examples. The number of iterations T is set to 10 and the weights for all models are identical. Decay factor μ ranging from 0.0 to 1.0 with granularity 0.1 is tested. Tabel 2 shows the result.

Table 2: Accuracy of models versus different decay factor μ



It is shown by the line plot that the accuracy drops lower than 0.2 for all predictions. The black-box model DenseNet, remaining a higher accuracy than the other three white-box models from the adversarial attack, reaches a turning point when μ is around 0.5. Therefore, the submitted adversarial examples are trained with ResNet, PreResNet, ResNeXt, SE-ResNet, and Densenet in a ensemble manner of MI-FGSM where μ is set to 0.5.

References

[1] Y. Dong et al., "Boosting Adversarial Attacks with Momentum," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 9185-9193, doi: 10.1109/CVPR.2018.00957.