Name: Connor Raymond Stewart
ID: 20673233

CS886 Robustness of Machine Learning: Homework One

Methodology and Accuracy Results:

The attacking method was the use of the FGSM attack. The purpose of the FGSM attack is to maximize the loss of with respect to the true label value [1]. The system works by adding a linear amount of noise which is minimally perceivable to cause the learning model to misclassify the image [1]. Noise is calculated by multiplying the sign of the gradient with respect to the image [1]. We weight the noise and multiply by a small epsilon value [1]. The rate that the model is inaccurate is proportional to the epsilon value, however, if the epsilon is too high the perturbations become perceivable [1]. Below is an adversarial function where we define x as the original image, epsilon as a small number, ∇_x is the gradient function with respect to the image, J is the loss function, and y is the true label for the image [1]. See below for the formula [1]:

$$X_{Adversarial} = X + \varepsilon \, . sign\left(\nabla_X J(X, Y)\right),$$

Importantly, we attempt to calculate the gradient such that we maximize the loss for the original image of the true label when we calculate the gradient with respect to the input image x [2]. Fast gradient descent models function since neural networks cannot operate through linear amounts of data perturbation well [2].

The implementation provided in the submission python file operates by importing the MNIST and CIFAR10 datasets and loading them into DataLoader data structures. The data is then iterated over, and the images are fed into the network models. The gradient tracked by the images is obtained and used to calculate the FGSM, which returns the perturbated image. The new image is then converted back to a numpy array and stored in place of the original image in the testing dataset. Afterwards, the dataset is saved to the filesystem and can then be used by the pgd attack python files for the MNIST and CIFAR10 datasets. The reported errors are as follows:

Table 1: FGSM Runtime Results

Dataset	Natural Error (%)	Modified Error (%)
MNIST	0.0052 (99.48%)	0.0204 (97.96%)
CIFAR10	0.1508 (84.92%)	0.1995 (80.05%)

As can be seen above, the FGSM system worked well for the CIFAR10 dataset, but not as well for the MNIST dataset. Although there is noticeable improvement for the MNIST dataset compared to the natural error, the improvement is small. It is likely the case that the fast gradient sign method does not change the overall shape of the numbers present in the MNIST dataset. Linear perturbations only create distortions in the numbers, but a neural network can still make out the shape of the numbers well enough to distinguish them from other numbers properly. Since the CIFAR10 dataset contains shapes with finer edges, linear perturbations have a larger impact on the networks error rate.

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv.org*, 20-Mar-2015. [Online]. Available: https://arxiv.org/abs/1412.6572. [Accessed: 24-May-2022].

[2] K. Tsui, “Perhaps the simplest introduction of adversarial examples ever,” *Medium*, 22-Aug-2018. [Online]. Available: https://towardsdatascience.com/perhaps-the-simplest-introduction-of-adversarial-examples-ever-c0839a759b8d. [Accessed: 24-May-2022].