Task - Adversarial attack and an adversarial training procedure for the MNIST dataset.

Name: Dheeraj Chittari, Mat No: 420501, contact: dheerajtukl@gmail.com

Introduction: Adversarial attack is adding a slight perturbation (noise) to the input that changes the classifier's prediction on that input to a different class. The added perturbations are very small that are visually imperceptible to humans yet they completely change the classifier's prediction on the input. They can be targeted and untargeted attacks. Most of the machine learning models are susceptible to the adversarial attacks. Therefore, adversarial training procedure is required to make the classifier more robust. General procedure for adversarial training is to incorporate adversarial samples into classifier's training procedure. By doing so, the classifier becomes more robust to that particular kind of attack. But it may or may not perform well on other type of attack that the classifier is not trained on. Hence a general procedure is required to make the classifier's more robust against any kind of adversarial attacks. This is a highly ongoing research topic in this domain.

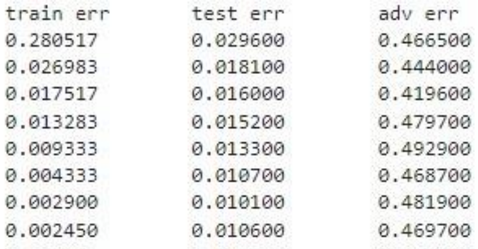
Method: The adversarial method followed in the current task is Projected Gradient Descent (PGD). In this attack, for a given sample x, we compute the weighted average of *L_2 and L_infinity norm* perturbation in the direction of its input gradient and add the perturbation to the sample. The perturbation must be less than some predefined *epsilon* as the perturbations added to the input must be visually imperceptible to humans. Fast Gradient Sign Method (FGSM) is a single step approach as it considers the maximum values of epsilon and updates the input only once, whereas in PGD we consider a step size (*alpha*) and performs the input update several times.

Procedure: Perform standard training procedure on MNIST dataset using the Pytorch framework. I have employed neural network model for the training with cross entropy loss. SGD (stochastic Gradient Descent) optimizer is used to update the parameters. Evaluate the model accuracy with and without adversarial samples generated by PGD attack.

Then perform adversarial training on a new CNN model (say robust_cnn_model). Then evaluate the robust model accuracy with adversarial sample created by PGD attack and compare its result with the above standard CNN model.

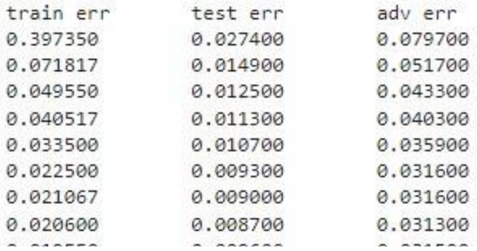
PGD hyperparameters: Here we have taken the step size (alpha) to be on the same scale as the total perturbation epsilon. Then, it is reasonable to choose alpha to be a small fraction of epsilon and get total number of iterations to be a multiple of epsilon/alpha. Therefore, I have considered alpha as 0.1, epsilon as 2.0 and number of iterations =20 (alpha*no. of iterations = epsilon).

Results: In this section, we will go through the standard and adversarial training errors. Later, the corresponding visualization on MNIST test data set are provided.



```
train err       test err        adv err
0.280517        0.029600        0.466500
0.026983        0.018100        0.444000
0.017517        0.016000        0.419600
0.013283        0.015200        0.479700
0.009333        0.013300        0.492900
0.004333        0.010700        0.468700
0.002900        0.010100        0.481900
0.002450        0.010600        0.469700
```

Figure 1 monitor training, test and adversarial loss during ***standard training***



```
train err       test err        adv err
0.397350        0.027400        0.079700
0.071817        0.014900        0.051700
0.049550        0.012500        0.043300
0.040517        0.011300        0.040300
0.033500        0.010700        0.035900
0.022500        0.009300        0.031600
0.021067        0.009000        0.031600
0.020600        0.008700        0.031300
```

Figure 2 monitor training, test and adversarial loss during ***adversarial training***

From figure 1, during standard training the train_err, test_err is decreasing. However, the adversarial error is very high and increasing. From figure 2, during adversarial training the train, test error as well adversarial error on MNIST dataset is decreasing. Thus, it indicates that the model is robust against the PGD adversarial attacks.

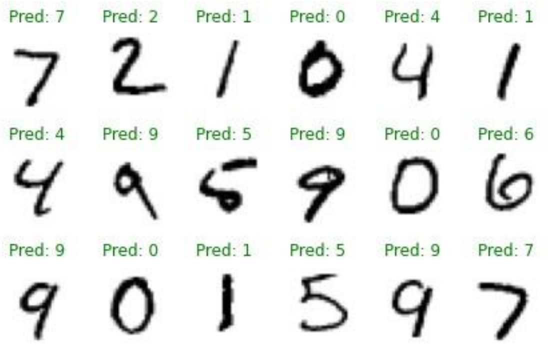


Figure 3 Visualize MNIST test dataset samples accuracy classified by trained CNN model.

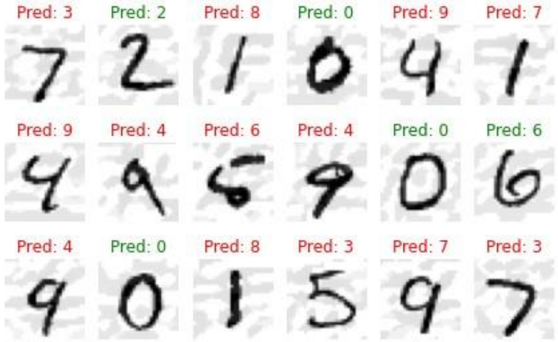


*Figure 4 Visualize test dataset samples accuracy that are **perturbed by PGD attack** and classified by trained CNN model.*

In figure 3, the background of each sample is completely white without any noise. Whereas, in figure 4, the background of each sample has some noise introduced by PGD attack. Therefore, it has several misclassifications by the standard CNN model.

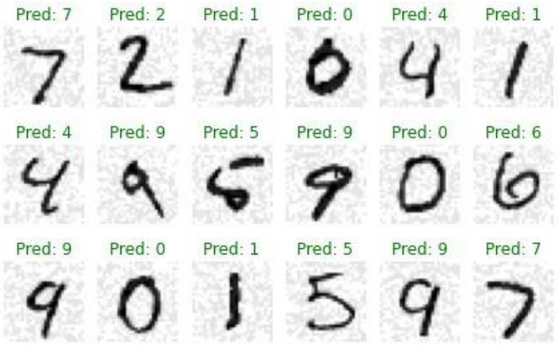


Figure 5 Visualize test dataset samples accuracy that are perturbed by PGD attack and classified by robust CNN model.

In Figure 5, even though the background of each image has some noise introduced by PGD attack, the images are still classified correctly by the robust_cnn_model as it has been trained on these adversarial attacks.

References:

[1] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv*. https://doi.org/10.48550/arXiv.1706.06083
[2] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. *arXiv*. https://doi.org/10.48550/arXiv.1412.6572
[3] https://adversarial-ml-tutorial.org/adversarial_examples/ (last accessed on 12.09.2022)

Complete code is available at:
https://github.com/dheerajvarma24/Adversarial_attack_and_training_task/blob/main/Adversarial_attack_and_training_HiWi_task.ipynb