CS 529: Assignment #4

Z. Berkay Celik
zcelik@purdue.edu
(Due by **December 2, 2022 11:59 PM**)

Computer Science, Purdue University

Instructions

> **Info:** This HW includes both theory and coding problems. Please read the course policy before starting your HW.
>
> - Your code must work with Python 3.5+ (you may install the Anaconda distribution of Python).
> - You need to submit a report including solutions to theory and coding problems (in pdf format), and Jupyter notebooks that include your source code.
> - You will compress your Jupyter notebooks in a zip file and submit them on **Brightspace**. The reports, however, will be submitted to the assignment on **Gradescope**; please check **Gradescope** to understand the format for the submission of the answers for each question.
> - Please mark your solutions correctly while submitting the report on Gradescope.
> - Please **include** your ipynb snippets (e.g., code and results) in your reports. These snippets can be in the form of a screenshot or a PDF. Do not forget to answer everything in the report as well.
> - You can always use the course **Campuswire** page to ask your questions. I encourage you to answer questions to help each other.
> - Failure to follow the instructions will lead to a deduction in points.

1 Problem 1: Security and Privacy of ML systems [20 pts]

> **Info:** Please use at most “five” sentences per question. Each question is 5pts.

1. What are the attacker’s goals in membership inference and model inversion attacks?
2. What are the challenges of generating adversarial samples in constrained domains (e.g., for malware classification)?
3. What is the difference between data poisoning and backdoor (trojan) attacks?
4. What is gradient masking? Why do the gradient masking-based defenses fail against adversarial examples?

2 Problem 2: Black-box Attacks and Transferability [30 pts]

Info: In this question, you will implement simplified black-box attacks (i.e., you do not have any access to the target model and its parameters) and evaluate transferability. You need to deliver `Problem2.pynb` file that includes your code.

The `target model` that you are attacking is trained on the MNIST dataset of 28×28 pixel images of handwritten digits, and you don't know any other details. You decide to collect ten images from each dataset class (numbers 0-9) to train your substitute model (also called auxiliary or surrogate model). You can use any dataset available online to have your own training dataset.

1. You start your attack by building your own CNN-based substitute model using the data you have collected. You then generate adversarial samples using your substitute model. You can use different networks to train your substitute model. To craft targeted adversarial examples, you will use three techniques (similar to HW #3, problem 5). You need to craft 100 adversarial images for three different "targeted" attacks of your choice (be sure that you have ten adversarial images per class). Overall, you have 300 adversarial examples crafted with three different targeted techniques (100 adversarial images for each attack). Plot the `Misclassification Ratio` (MR) for each attack. `MR` is defined as the percentage of adversarial examples misclassified into the target classes as specified before. (10 pts)

2. For each of the adversarial samples that you have generated, you desire to find the amount of perturbation between the original image and the adversarial image. You will use $\mathtt{L}_0$, $\mathtt{L}_2$ and $\mathtt{L}_\infty$ norms to see this difference. That is, if you have generated an adversarial example for class 2, then you will find the norms of an original class image and the adversarial image that you have crafted for that class. Report your average norms in a table for each class over three different techniques. The table will have 3 rows for each attack and columns for the norms (**Note:** you take the ten adversarial images per class and then find the average norms). **Intuition:** This question is about imperceptibility, which implies that the adversarial example would still be correctly classified by humans, which ensures that the adversarial and benign examples convey the same semantic meaning. (10 pts)

3. You believe that the target model you attack might be trained on using the model given in your HW#3 Problem 4. Recall that this model is trained on the MNIST dataset of 28×28 pixel images of handwritten digits. You want to evaluate whether adversarial examples generated with your substitute model will transfer to the target model. To do so, you will find the percentage of adversarial samples produced using the substitute misclassified by the target model for three different attacks. Be sure that you report your results for each different attack. Here crucial takeaway is that you have used a different model and limited dataset to craft adversarial examples, and you want to see whether your adversarial examples are successful on the target model. (10 pts)

4. **Extra Credit [10pts]**: You now would like to evaluate the transferability of your adversarial examples to other classifiers. You assume that the target model might be trained on the MNIST dataset through six different classifiers: (1) ANN, (2) SVM, (3) Logistic Regression, (4) kNN, (5) Naive Bayes, and (6) Voting classifiers. We assume these classifiers are trained on the MNIST dataset available in keras.datasets, and grid search is used to improve their accuracy. Therefore, you need to use the grid search and try different parameters to obtain high accuracy.

   How transferable are your adversarial images across all the trained models? To answer this question, you will report the percentage of adversarial samples produced using the substitute model misclassified by target models. The table includes rows of the different classifiers and two columns for their test accuracy and the percentage of adversarial transferred examples for three different attacks. What do you infer from your observation(s)?

   Hint: Read more about grid search. Voting classifiers use estimators; you will use ANN, SVM, Logistic Regression, kNN, and Naive Bayes for your estimators. You can use the optimal parameters that you acquired in gridsearch of single classifiers in the voting classifier as well. As a side note, you may need to give a range of parameters that are given to grid search to limit the runtime.

3 Problem 3: Defense against Adversarial Samples [20 pts]

> **Info:** You are provided with the Jupyter notebook, `Problem3.ipynb`, for this question. Please complete the following questions in that notebook. You will use the adversarial attack method you have implemented in `Homework#3`. To better understand the questions below, you need to first check `Problem3.ipynb`.

1. You will generate 10K adversarial images and mix that up with your training dataset with the correct labels. You will create a new neural network (with your own network parameters) and train it on 10000 adversarial examples in addition to 50000 original training set examples (augmented model). Report the accuracy of your augmented model on the original test set using your model. [10 pts]

2. You will evaluate whether adversarial training is successful against adversarial examples. First, you will create a function that compares the original network to the new network on adversarial examples. Here, show an adversarial example image's output on original network prediction and new network prediction. You will then report the accuracy of your augmented model on adversarial examples on the new adversarial test set. Is adversarial training effective against adversarial images? [10 pts]

3. **Extra Credit [15pts]**: You can use existing libraries such as cleverhans, IBM's ART, or publicly available codes to evaluate three more defense techniques against adversarial examples using your own network. You need to show their robustness using three utility metrics of your choice defined for defense evaluation, such as Classification Accuracy Variance (CAV), Classification Rectify/Sacrifice Ratio (CRR/CSR), Classification Confidence Variance (CCV), and Classification Output Stability (COS). These metrics are explained in detail in the assigned paper-review. Please Include your solutions in the provided notebook.

4 Problem 4: Generative Adversarial Networks [30 pts]

> **Info:** In this assignment, you will be implementing GANs using `torchgan` library. This library provides implementations of a few well-known GAN architectures and provides a framework for you to implement your own GAN architecture. To understand how this library works, please refer to `https://torchgan.readthedocs.io/en/latest/index.html`. `torchgan` provides implementations of well-known GAN architectures, and for this homework, you will implement DCGAN (Deep Convolutional GAN) and one additional architecture to analyze how it works using MINST dataset. In the problem notebook – `Problem4.ipynb`, we have a base for the implementation of GANs, and you will be filling in the content as and where asked.

1. First, you should load the MNIST dataset from the dataset library from the pytorch datasets. You will construct and train a DCGAN model. To see the details about the DCGAN architecture, please refer to the paper: `https://arxiv.org/pdf/1511.06434v2.pdf`. Once the model is trained, you will compute the accuracy of the generator and the discriminator. (Hint: To compute the accuracy of the generator, we have different score types such as `Inspector Score`, which is a score of how accurately human eyes can identify the images as real. See `https://medium.com/octavian-ai/a-simple-explanation-of-theinception-score-372dff6a8c7a` for more details. Discriminator accuracy is basically how well the discriminator can differentiate between real and fake images.) (15pt).

2. Use different epoch values for training the model to see how the accuracy of the discriminator varies as the number of epochs changes. Submit the notebook with the one which you find to have the higher accuracy but submit both accuracy values and please provide 10 randomly sampled generated images from each class (10x10 image matrix with each column being a single digit from MNIST) (10pt).

3. What is your inference from this? What is one drawback of DCGAN neural network? (5pt)

4. **Extra Credit [10pts]**: Implement another GAN architecture supported by torchgan that helps overcome this drawback. Follow the previous steps and submit the same experimentation results as you did in DCGAN model.

5. **Extra Credit [5pts]**: Did overcoming the drawback help to increase the accuracy of the model? Could there be another factor for the accuracy increase/decrease? Note that there is no single correct answer to this question.

6. **Extra Credit [10pts]**: Plot the training loss of the discriminator and generator and explain the plot mathematically based on the common GAN loss equation (minmax loss function).