SIL765: Networks and System Security

Semester II, 2023-2024

Assignment-5

April 18, 2024

Problem: ML Classification and Adversarial Attacks (100 Marks)

Introduction

This assignment delves into two fundamental concepts in machine learning: classification and adversarial attacks. **Classification:** Classification tasks involve categorizing data points into predefined classes. This assignment will focus on image classification, specifically recognizing handwritten digits. **Adversarial Attacks:** Adversarial attacks exploit vulnerabilities in machine learning models by introducing imperceptible modifications to input data. These modifications, called adversarial examples, can cause the model to make incorrect predictions despite appearing similar to the original data.

We'll utilise the well-known MNIST dataset, a collection of handwritten digits from 0-9. This dataset serves as a standard benchmark for evaluating image classification models.

We'll employ the Fast Gradient Sign Method (FGSM) to generate adversarial examples. FGSM is a simple yet powerful technique that perturbs input data in the direction of the gradient of the loss function. This manipulation maximises the model's prediction error, essentially tricking the model into misclassifying the adversarial example.

Throughout this assignment, you'll gain hands-on experience with these concepts by implementing a classification model, generating adversarial examples using FGSM, and evaluating the model's robustness against such attacks.

Part-I : Building and Evaluating an ANN Model (30 Marks)

To fulfill the requirements of this part, you need to follow these steps:
Data Preparation: Utilize the MNIST dataset and split it into a 60:40 ratio for training and testing respectively. Save the training data in a folder named ”train data” and the test data in a folder named ”test data”.

Model Implementation: Construct an Artificial Neural Network (ANN) model for classification. Define the structure of the model and explain the hyperparameters used. Save the implementation in a file named ”classify.py”.

Model Training: Train the ANN model using the training data. Monitor and record the training and validation accuracy.

Model Evaluation: Evaluate the performance of the trained model based on its accuracy in classifying the test data. Record the test accuracy.

Analysis and Improvement: Emphasize on improving the accuracy of the ANN classifier and making it a robust classifier. Discuss any techniques or strategies employed for improving robustness.

Performance Metrics: Mention the training accuracy, validation accuracy, and the test accuracy of the model. Plot the validation loss and training loss to visualize the model's performance during training.

Part-II: Generating Adversarial Examples (70 Marks)

Imagine you train a super-intelligent AI to tell cats from dogs in pictures perfectly. Adversarial examples are like putting on a tiny, invisible costume on a dog that makes the AI think it's a cat! These changes are so tiny humans wouldn't notice, but they trick the AI. Adversarial examples are specially crafted inputs that trick machine learning models into making incorrect predictions. These examples are often imperceptible to humans, appearing similar to legitimate data.

The Fast Gradient Sign Method (FGSM) is one method for generating adversarial examples. FGSM works by adding an imperceptible, calculated noise to the input data in the direction that maximizes the model's error. This ”pushes” the input towards a region where the model is less confident, causing misclassification.

Your task is randomly choosing 1000 images from the MNIST test set and crafting corresponding adversarial images using FGSM. These adversarial examples aim to lower the accuracy of the trained model in part 1 as low as possible. The adversarial examples should be crafted for the trained model in part 1.

Along with generating the adversarial examples using FGSM (40 Marks), answer these questions:

1. **Evasion Rate** is a metric used to measure the evasiveness of adversarial examples against a trained ML model. You'll report the evasion rate of your generated adversarial examples against your trained model. The evasion rate is calculated by taking the ratio of the total number of adversarial examples misclassified and the total number of examples. (10 Marks)
2. You have to pick random images of all ten digits and do a plot to show the corresponding adversarial examples and the corresponding adversarial noise. The plot should be $3 * 10$ in dimensions. The first row shows original images of ten different digits, the second row shows their corresponding adversarial examples, and the final row shows the L_2 norm of their difference (L_2 norm of the adversarial noise) (10 Marks)
3. You have to randomly pick one digit and report to which class it got misclassified the most. Why do you think this is the case? Do you see any structural similarity between those two digits (your chosen digit and the most misclassified digit)? (10 Marks)

The generated adversarial examples would be tested against the stored model (a robust model). Hence, you must generate adversarial examples in a specific format to pass the hidden test cases. Refer to the code template to find the checks and ensure your returned results are in the appropriate format.

Note: You don't have to submit any adversarial examples. We will generate adversarial examples from your algorithm and test them against the robust model. All you need to make sure is the attack algorithm (FGSM) works.

Submission

You should submit a single folder containing all files related to this assignment.
The folder should be named ⟨your_entry_number⟩-Assignment-⟨assignment_number⟩.
It should contain:

- All the files and folders mentioned in Part-I
- model.py: Code template provided
- attack.py: Code template provided
- model.pth: Generated trained model
- readme.MD: Report describing your methods and answers to the questions mentioned above.