In the name of God

University of Tehran
Faculty of Electrical and Computer Engineering

Neural Networks and Deep Learning Course
Assignment 5

Question 1	
Designer TA Name	Arian Firouzi
Email	arianfirooziM@gmail.com
Question 2	
Designer TA Name	Mohammad Gorji
Email	mohamadgorjicode@gmail.com
Submission Deadline	2025-06-08

Contents

1 Rules

Before answering the questions, please read the following items carefully:

- Prepare a report of your answers in the format provided on the course page on the Elearn system named `REPORTS_TEMPLATE.docx`.
- It is recommended to do the assignments in groups of two. (More than two people are not allowed, and individual submissions do not receive extra credit). Note that it is mandatory for group members to remain the same until the end of the term (meaning you can do the first assignment with person A and the second assignment with person B).
- The quality of your report is of great importance in the grading process; therefore, please mention all the points and assumptions you considered in your implementations and calculations in the report.
- In your report, according to what is provided in the sample template, consider captions for figures and tables.
- It is not necessary to provide detailed explanations of the code in the report, but the results obtained from it must be reported and analyzed.
- Analysis of the results is mandatory, even if it is not mentioned in the question.
- The teaching assistants are not obliged to run your codes; therefore, any results or analysis requested from you in the question must be clearly and completely stated in the report. In case of non-compliance, it is obvious that points will be deducted from the assignment.
- The codes must be prepared in a notebook with the `.ipynb` extension. At the end of the work, the entire code must be executed, and the output of each cell must be saved in your submitted file. Therefore, for example, if the output of a cell is a plot that you have included in your report, this plot must also be present in the notebook of the codes.
- In case of cheating, the score of all participants will be considered 100-.
- The only authorized programming language is Python.
- Using ready-made codes for the exercises is by no means permissible. If two groups use a common source and submit similar codes, it will be considered cheating.
- The method of calculating the delay is as follows: after the submission deadline, there is a possibility of submission with a delay of up to one week. After this one week, the score for that assignment will be zero for you.
  - First three days: no penalty
  - Fourth day: 5% penalty
  - Fifth day: 10% penalty
  - Sixth day: 15% penalty
  - Seventh day: 20% penalty
- The maximum score that can be obtained for each question is 100, and if the total score of a question is more than 100, if you get a score higher than 100, it will not be applied.
- For example, if the score obtained from question 1 is 105 and the score of question 2 is 95, the final score of the assignment will be 97.5 and not 100.
- Please put the report, codes, and other attachments in a folder with the following name, compress it, and then upload it to the Elearn system:

  `HW[Number]_[Lastname]_[StudentNumber]_[Lastname]_[StudentNumber].zip`

  (Example: `HW1_Ahmadi_810199101_Bagheri_810199102.zip`)
- For groups of two, it is sufficient for one of the members to upload the assignment, but it is recommended that both upload it.

2 Question 1: Image Classification with ViT

2.1 1-1. Introduction (10 points)

In agriculture, the timely diagnosis of diseases and pests in plants plays an important role in the health and fertility of products. In this exercise, we want to use two types of neural networks to classify diseases using plant leaves.

Vision Transformer (ViT): These are models that use the transformer architecture and attention mechanism on image datasets.

Convolutional Neural Network (CNN): As you have become familiar with the workings of this type of neural network before, convolutional neural networks extract image features using convolutional layers.

The purpose of this exercise is to become familiar with the structure of the Vision Transformer and its differences with other neural networks. For this purpose, the dataset used in the article will be used, which can be easily downloaded from this link.

To complete the exercise, you need to study this article.

Regarding the article and Vision Transformers, answer the following questions:

- According to the article, what is the main difference and advantage of using Vision Transformer models compared to traditional models? (5 points)
- When the existing dataset is limited, which model performs better? Justify your answer with respect to the structure and mechanisms used in the models. (5 points)

2.2 1-2. Data Preparation (15 points)

Receive the data and follow the steps below:

- Display a sample from each class: Show one image from each of the 10 classes in the dataset. (2 points)
- Check data balance: Compare the number of images for each class in a table. Is the number of images in the dataset balanced? If your answer is no, argue what kind of data augmentation can be used to balance the data without seriously damaging the quality of the images and apply it. (7 points)
- If you suggest another preprocessing that can improve performance, apply it to the data. According to the input size of the models and the size suggested in the article, the images must be in specific dimensions. For this purpose, the internal structure of the CNN used (which is Inception-V3) must be changed. But since the main topic of the exercise is the Vision Transformer, you can use a different size. (3 points)
- Divide your data into two sets: training and validation. (3 points)

2.3 1-3. CNN Model Training (20 points)

First, load the Inception-V3 model as raw and without pre-training. Set the number of outputs according to the dataset and set the other parameters according to the article. Describe the overall functionality of this model. (7 points)

Explain the loss function used in the article. What other functions are suitable for this task? (3 points)

Train the model for at least 10 epochs and plot the accuracy and loss graphs for both datasets. Plot the model's confusion matrix. (10 points)

2.4 1-4. ViT Model Training (40 points)

Implement the model described in the article. It is necessary to display the output of the model layers, and if a layer acts contrary to what is stated in the article, state the reason. (20 points)

For what purpose is the Patch Embedding layer used in image transformer networks? What effect does decreasing or increasing the size of each patch have on the output in this exercise? (5 points)

Embed a pixel output capability in the Patch Embedding layer and show the output of this layer as an image using it. (5 points)

Train the model for at least 20 epochs and plot the accuracy and loss graphs for both datasets. Plot the model's confusion matrix. (10 points)

2.5 1-5. Analysis and Conclusion (15 points)

Compare the results of the two methods.

- Compare the models in terms of accuracy, precision, and the number of parameters. (8 points)
- Considering which model performed better, explain under what circumstances the weaker model could have performed better. (7 points)

3 Question 2: Robust Zero-Shot Classification

Adversarial attacks in neural networks are one of the fascinating and controversial challenges in deep learning and trustworthy artificial intelligence. In these attacks, samples are designed that are practically indistinguishable from the original samples by humans but can deceive machine learning models. These samples, which are called **adversarial examples**, are created with small and targeted changes in the input data and cause the model to produce an incorrect output. For example, adding small noises to an image can cause an image classification model to misidentify a stop sign as a go sign. (Figure 1)

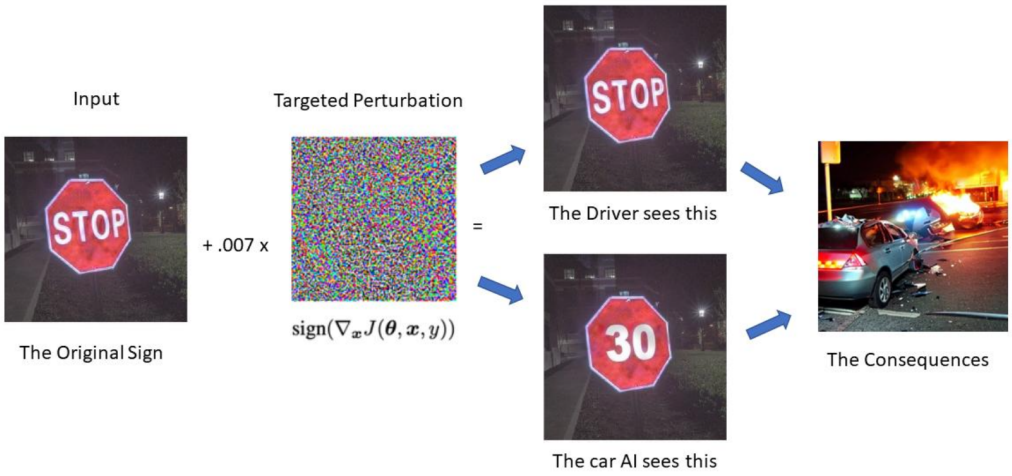

Figure 1: A mistake by an artificial intelligence model in diagnosing the class of an adversarial example.

3.1 2-1. Introduction to the CLIP Model, One-Shot Classification, and Adversarial Attacks

First, study the article and answer the following questions:

1. Explain the two methods of generating adversarial examples, FGSM and PGD, with details of their optimization. (6 points)
2. Describe the architecture of the CLIP model with a diagram and explain the loss function[1] and the type of Contrastive training. (6 points)
3. Explain the key differences between normal classification and one-shot classification[2] and, with a diagram, explain how this is done with the CLIP model. (6 points)
4. There are two types of adversarial attacks, white-box[3] and black-box. Explain each and then compare these two approaches from different desired aspects. (6 points)
5. Explain why transfer attacks[4] are considered a serious threat to real-world problems compared to white-box attacks. (6 points)
6. Explain the fine-tuning[5] method LoRA with a diagram and state three reasons for using this method over other methods. (10 points)
7. Find two research articles that extend or improve the CLIP loss function and summarize each in one or two paragraphs. Also, explain how the impact of these loss functions on increasing the robustness of the CLIP model has been. (10 points)

[1] Loss Function
[2] Zero-shot Classification
[3] White-box & Black-box
[4] Transfer Attacks
[5] Fine-tuning

3.2 2-2. Implementation and Comparison of Adversarial Training Methods

In the second part of this project, you are to fully implement the training and evaluation process of one-shot CLIP models against adversarial attacks using the CIFAR-10 dataset and PyTorch, torchvision, Transformers, and PEFT libraries.

1. For this question, you must use the CIFAR-10 dataset. Download this data using the torchvision module and divide it into three sets: training, validation, and testing. Use a function to display 5 random samples from the dataset. Then, resize the images to 224x224 with suitable transformation functions and normalize them with the specific mean and standard deviation of CLIP. (5 points)

2. After preparing the data, load the CLIP model from the HuggingFace repository and put it in evaluation mode. Also, as a model for generating adversarial examples, download a pre-trained ResNet-20 model on CIFAR-10 from PyTorch Hub using the code snippet below and keep it in eval mode as well. Then, by generating text vectors for each of the ten CIFAR-10 classes (e.g., "a photo of an airplane", "a photo of an automobile", ...), prepare the text space of CLIP to be used for one-shot classification in the next steps. (5 points)

```
import torch
target_model = torch.hub.load("chenyaofo/pytorch-cifar-models",
                              "cifar10_resnet20", pretrained=True)
target_model.eval()
```

3. The next step is to evaluate the CLIP model on clean images. By writing a function that calculates the image feature vector and then normalizes it, and by taking the dot product of these vectors with the text vectors, obtain the classification output and report the model's accuracy. Then, using a PGD attack on the ResNet-20 model (with $\epsilon = 8/255, \alpha = 2/255, \text{steps} = 7$), create adversarial examples (using the torchattacks library) and run the same evaluation function to observe the accuracy of CLIP against these transfer attacks. To make the issue more tangible, take a test image and display it in a three-part figure alongside its adversarial version and the attack noise itself. (5 points)

4. After that, it is time for standard adversarial fine-tuning with the LoRA method. For this, first apply LoraConfig settings with Low-Rank Adaptation to the CLIP vision module layers (for example, r=8 and $\alpha = 32$). Then, put the model in train mode and, in a training epoch, for each batch of images, calculate the CLIP feature vector on the adversarial images and take the dot product with the text vectors. With the CrossEntropyLoss function on the original labels, calculate the gradients and update the LoRA parameters. At the end of this stage, re-measure the clean and adversarial accuracy of the model to see how much standard adversarial training has been able to increase the model's robustness. (15 points)

5. In the next step, implement the TeCoA (Text-guided Contrastive Adversarial Training) algorithm according to equation (3) of the article and, like the previous section, train the model with this cost function and perform the related tests. (15 points)

6. Finally, display the clean and adversarial accuracies of the three main states: the original CLIP model, the fine-tuned CLIP with LoRA and CrossEntropy, and the fine-tuned CLIP with LoRA and the TeCoA algorithm in a comparative table or graph, and briefly discuss the advantages and limitations of each method and the extent of the reduction or increase in clean accuracy and adversarial robustness. (5 points)

7. (Bonus) Using the TeCoA method, fine-tune the model without LoRA and using visual prompting tuning (refer to the main article). Then, compare this method with the other two methods and analyze the results. (5 bonus points)

Note 1: According to the article, for sections 4 and 5, you must implement and compare methods two and five (mentioned in the article). Of course, studying other methods will also help in understanding the whole problem.

1. vanilla cross-entropy loss (CE)

2. standard adversarial training loss (Adv.) with the cross-entropy loss
3. contrastive adversarial training loss (CoAdv.)
4. contrastive adversarial training over images (ImgCoAdv.)
5. our text-guided contrastive adversarial training (TeCoA)

Note 2: For training, use an equal number of samples from the test dataset; there is no need to train on the entire training dataset.