Assignment 1

Training neural networks

Please submit a single tar file named ”ex1.tar”. This file should contain your code, a weights checkpoint named “<your_id>.ckpt”, a “README.txt” file that will help us to orientate in your code, and an ”Answers.pdf” file in which you should write your answers and provide relevant figures to support your answers. Your pdf should be no longer than 6 pages (and it will probably be shorter). Your feedback and grade will be based mainly on this lab-report pdf.
Please write your ID in the answers.pdf file. Notice that your code will be checked manually, so please write readable, well-documented code.

Exercise requirements

In this exercise, you will:

1. Train the supplied network on the supplied dataset and try to reach the best possible performance. (50 points + 10 points for the competition)
2. Show how the learning rate affects the learning process. (10 points)
3. Generate an adversarial example for your model. (30 points)

A small part (10%) of the grade will be calculated according to the performance of your network in comparison to the class. (this is the competitive part of the exercise).

Setup and computation

You will need pytorch, numpy and matplotlib. We recommend training the model on your laptop. We highly recommend using the Conda package manager.

Collaboration

In order to mimic the collaboration usually occur on campus, the exercise is published in teams. It is recommended to open a WhatsApp group for the team or to create a group in Piazza. The teams meant for mutual help, please avoid unwanted spoilers.
Please tell us at the end of your exercise about the team’s dynamics. We are not interested in who thought about which idea, we just want to know if you had collaborated successfully.

Presenting solution in class

If you want to present your solution briefly in the recitation right after the submission deadline, send me (Omri) a Message after submitting it.

Few additional notes

Our expectations

As you probably already saw, we didn't write a list of required figures/statistics. It is up to you to choose what statistics are needed in order to support your claims.

A note about information sharing

You should not put your hands on someone else's solution. Do not share code. You are welcome to help each other in any other way.

A note about cheating

You will be surprised to hear that it is easy to catch when someone cheats in the competition (for example, pre-train the network using additional data from CIFAR10). Please don't try, It will be your last exercise in the course. As we see it, cheating in competition against your classmates is much worse than cheating us. neither is acceptable.

A note about "participation grade"

Since we don't know how to measure it, we will not measure it for now. You don't need to redirect your communication to Piazza in order to make it possible to grade. We will figure it out toward the next exercise.

Training a neural network

In this chapter, you need to train an NN to classify cars, trucks and cats. Your model will be evaluated against a test-set that contains 100 images per class. Your dataset is somewhat distorted.

1. Inspect the dataset and present your insights.
2. you are provided with a trained baseline model (trained using cross-entropy on the provided dataset). please write an evaluation script and evaluate this model. explain your evaluation metrics and results.
3. Write a training loop and train **the given model** to reach the baseline performance. The code in this is a good place to start (use it).
4. train **the given model** to the best performance you can. explain what you are doing and why.

Together with your code and report, you should submit a checkpoint of your model for evaluation in the competition.
Finally, please add a script for reproducing your final training stage. It doesn't have to be 100% reproducible (randomness, etc) and doesn't need to capture the data preparation. You can add

any additional files needed. Probably, this script will never be executed and it is required in order to encourage organized workflow and discourage cheating.

Playing with learning rate

Show that the learning process does not converge if too high or too low learning-rate is used. Explain this phenomenon shortly.

Adversarial example

generate an adversarial example for the model from the previous section. Please explain what you have done and visualize the output of the model, the image and the noise. You should also submit your code, of course.