CSCI 4160/6963, ECSE 4965/6965 Reinforcement Learning

Homework 8

Overview

This homework will be a warm-up deep learning assignment. You will experiment with hyper-parameters and their effect on training a fully-connected architecture on the MNIST and CIFAR10 datasets. CIFAR10 is a dataset of color images, where each image is 32×32 (so total number of dimensions is 3*32*32). Similar to MNIST, CIFAR10 has 10 labels, such as plane, car, horse, etc. Once you experiment with MNIST and CIFAR10, you will create a synthetic dataset in order to understand the generalization capabilities of neural networks.

Logistics

All coding assignments in this course will be in Python. If you need help with Python, please talk to me ahead of time, so we can discuss the best way to get familiar with it.

For this assignment, you will be using your own computer, which should be sufficient. The code should not require more than 15 minutes to run on any standard laptop.

You are provided with skeleton starting code. The MNIST dataset is the same as in previous assignments. The CIFAR10 dataset will be downloaded by the provided skeleton code. To run the code, you will need a version of Python 3, as well as at least the following modules (in addition to modules from previous homeworks): torch, torchvision. To install a package, you can use pip as follows:

```
pip install [package-name]
```

This assignment has more skeleton code than usual because setting up deep learning from scratch can be a bit tricky. Make sure you look at all provided code. The `model.py` file contains the neural network definition. The classification files have similar starting code. Make sure you look at all the hyper-parameters and figure out what they do. Also, take a close look at the rest of the provided code and understand what it does.

Grading

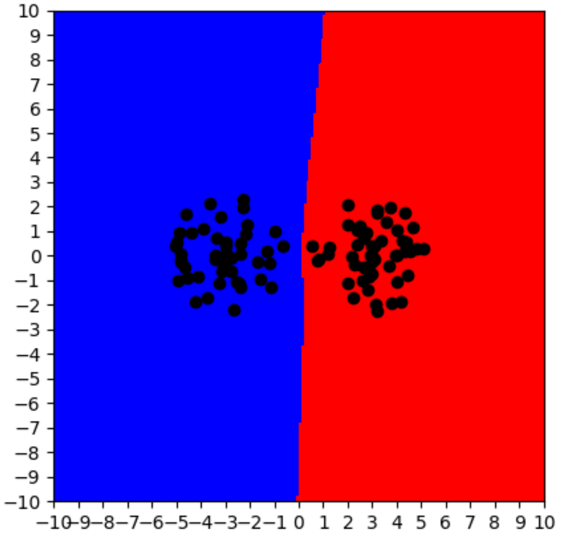
This assignment has three coding tasks. The **MNIST task** is to fill up the training loop code (together with a test function call at the end). Then, you are required to achieve 97% test accuracy (and will lose 5 points for each percentage point below 97, rounded up) by changing

only one hyper-parameter from its default value. Feel free to play around with the various hyper-parameters to get a feel for their effect.

The **CIFAR10 task** will be a bit unusual. First, same as in the MNIST case, you need to fill up the training loop code (together with a test function call at the end). Then, you need to achieve 60% training set accuracy (note that this is accuracy on the training set, **not** the test set). You are allowed to change **only two** hyper-parameters: the one from the MNIST case and an additional one that you find useful. You will lose 5 points for each percentage point of training accuracy below 60, rounded up.

The **last task** will be a synthetic data exercise. You will build a training set, where class 0 contains 50 examples within radius 2.5 of the point (3,0). Similarly, class one contains 50 examples within radius 2.5 of the point (-3,0). Those two are clearly separable by a vertical line on the y-axis. However, the test set will be sampled as follows: class 0 has 50 points sampled randomly in the square $[0.5,5] \times [-10,10]$; class 1 has 50 points sampled randomly in the square $[-5,-0.5] \times [-10,10]$. So the y-axis should still separate the test data.

Your job is to train a network on the training set and then show the test accuracy. Repeat this 10 times for different random data sets. What is your test accuracy? Finally, you need to plot the decision space of the network (one graph is enough here, no need for all 10) over the entire sampling space, along with the training data. There are no styling requirements, something like this should be enough:



Finally, please provide answers to each of the questions below:

1) Why is it so hard to achieve even high training accuracy on CIFAR10? Also, why is the test accuracy much lower than the train accuracy?

2) Why does the neural network make mistakes on the synthetic test set? Which model class would be better suited for this task?

3) [**Graduate students only**] How many parameters does your CIFAR10 neural network have? What is another neural network architecture that could achieve even higher accuracy with fewer parameters?

Hints and Tips

- Make sure you plot some CIFAR10 images before you start, to get a feel for the dataset.
- When experimenting with hyper-parameters, think about the general effect each hyper-parameter should have on the data.
- When you create your synthetic dataset, make sure you use torch (or if you use numpy first, then cast your arrays into torch tensors). You won't be able to train the network if the data are not tensors.
- To plot the decision boundary, you can grid up the space and output the network's prediction for the middle of each cell. The grid will not be 100% accurate since you're not trying all points, but the network's output doesn't vary that drastically, so the grid will be a good approximation.

Submission

Please use `LMS` to submit a zip file containing 1) your **.py** code (in three files, as provided), along with instructions on how to run it, 2) a **.pdf** file containing your answers to the above questions, 3) a **.png** file with a decision boundary plot. The deadline is **11:59 pm, Thursday, Nov. 14**.