Middle East Technical University Department of Computer Engineering

CENG 499

Introduction to Machine Learning

Fall 2020-2021

Homework 1 - Artificial Neural Network

version 1

Due date: 27 November 2020, Friday, 23:59

1 Introduction

For this assignment you are going to implement an ANN (Artificial Neural Network) to classify the 2-digit numbers on the dataset we provided. You are going to implement various architectures and optimize their hyperparameters. The implementation will be done in Python3, using PyTorch library. After getting the results, you are expected to plot the graphs of the best networks and comment on them.

2 Dataset

The dataset you will be working on is created using MNIST Dataset (`http://yann.lecun.com/exdb/mnist/`) and by adding noise over it. However, this assignment requires you to classify numbers from 0 to 99. The dataset can be downloaded through: `http://user.ceng.metu.edu.tr/~artun/ceng499/dataset.zip`.

The dataset has 2 folders: "train" and "test" in which the input images reside. These folders also contain the label files: "labels.txt". For train split, every line in labels.txt contains image name and its label. However, the labels.txt in test folder only contains image names and the labels for the test split are not given. How you are going to calculate accuracy will be discussed in the later sections. We have not specified the validation split. You can create your own validation split from the train split.

3 Test Accuracy Calculation

To emphasize that you cannot use test set during training or hyperparameter optimization, we did not provide the labels for test split. However, you still need to report the test accuracy of your best networks. To calculate your accuracy, you can login the website:

`http://188.166.160.183/`

and submit your prediction in the following format:

```
image_name_1 predicted_label_1
image_name_2 predicted_label_2
...
```

As you can see the format is very similar to labels.txt in train split. For hyperparameter optimization and determining best ones, you can divide your training data into validation and training sets and calculate validation loss/accuracy or use cross validation or similar methods. **The ranking in the leaderboard in the website does not affect the grade you get from this homework.** Your username and passwords will be sent via email.

4 Networks and Reports

You are going to implement different models and train them using different hyperparameters. You should pay attention to the following points:

- To get reproducible and consistent results, before the training procedure, you can write the code below:

```
torch.manual_seed(1234)
```

- You are expected to use fully-connected(linear) layers in your network in this homework. However, you can try other type of layers as well if you want to.
- You are going to make a multiclass classification. Activation function at the output layer should be **softmax** function and you should use **cross entropy** as your loss function.
- Since the output shape of the network should match the number of classes, the number of units in the final layer will be 100 .
- After the random initialization of the weights and before training the model and calculate the accuracy and loss an on some split. Comment on the result you get. Is it similar to what you expected?
- You can choose a suitable optimizer. You don't have to spend too much time on it though. Adam optimizer can be a good initial choice.
- You can stop your optimization process using early stopping. For example, you can stop it if your validation loss hasn't been dropping for some constant number of epochs, etc.
- You should do optimization on the following hyperparameters, draw tables for them and select the best one (you can also experiment on other hyperparameters as well but these are the required ones):
  - Try different number of layers. You should at least try 1(0 hidden layers), 2(1 hidden layer), and 3(2 hidden layers) layer networks. You should also try different number of neurons in each hidden layer; however, you don't have to spend too much time on it.
  - Try different activation functions at the end of each layer. You should try at least **sigmoid**, **ReLU**, and **tanh**. You can implement them yourself if you want but they are already implemented in PyTorch. You are free to use them.
  - Try different learning rates. For example, you can divide learning rate by $\sim \sqrt{10}$ and try these: 0.1, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001, 0.00003, 0.00001 etc. When the learning rate is too small the validation loss may continue decreasing for many epochs. Therefore, you may want to limit the number of epochs since it can take too much time.
- **Do grid search on the hyperparameters above** and fill the tables in the given report.tex. The values in those tables should be validation accuracy/loss. From all the k-layer networks, select the best performing hyperparameters and draw its training and validation losses on the same graph and

comment on it. The hyperparameters you tried should be visible in your codes. (Do not change the code and try values by hand. Write a loop instead.)

- What countermeasure did you take against overfitting? How may one understand when a network starts to overfit during training?
- Comment on your test accuracy results.
- You should be able to achieve accuracy around 0.3 in test set with the best architectures required architectures.

5 Specifications

- The codes must be in Python3 and must use PyTorch.
- Include a README file explaining how to run the training and testing.
- Falsifying results, changing the composition of training and test data are strictly forbidden and you will receive 0 if this is the case. Your programs will be examined to see if you have actually reached the results and if it is working correctly.
- The late policy given in the syllabus applies here. The late submission penalty will be calculated using $5n^2$, that is, 1 day late submission will cost you 5 points, 2 days will cost you 20 points and 3 days will cost you 45 points. No late submission is accepted after reaching a total of 3 late days.
- Using any piece of code that is not your own is strictly forbidden and constitutes as cheating. This includes friends, previous homeworks, or the internet. You can use the examples that are provided in PyTorch's website. The violators will be punished according to the department regulations.
- Follow the Announcements page on ODTUCLASS for the updates and clarifications. You can ask your questions on Homework1 in Discussion section of ODTUCLASS instead of e-mailing if the question does not contain code or solution.

6 Submission

Submission will be done via ODTUCLASS. If you do not have access to ODTUCLASS send send an email to "artun@ceng.metu.edu.tr" as soon as possible. You will submit a zip file called "hw1.zip" that contains all your source code, the README file, and your report in a pdf format compiled from the given latex file.