Middle East Technical University Department of Computer Engineering

CENG 499

Introduction to Machine Learning

Spring 2018-2019

Homework 1 - Artificial Neural Network

version 1

Due date: 26 03 2019, Tuesday, 23:59

1 Introduction

For this assignment you are going to implement an ANN (Artificial Neural Network) to classify the color of the t-shirts of pedestrians for the dataset we provided. You are going to implement various architectures and optimize their hyperparameters. The implementation will be done in python, using Keras library using TensorFlow backend. 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 originally taken from: `http://mmlab.ie.cuhk.edu.hk/projects/PCN.html` which uses the pictures from Market-1501(`http://www.liangzheng.org/Project/project_reid.html`) dataset.
However, this assignment asks you to work on slightly different task. Therefore, you should download the dataset using this link: `http://user.ceng.metu.edu.tr/~artun/ceng499/dataset.zip`.
The dataset has 2 folders: "train_set" and "test_set" in which the input images reside. The color labels are stored in "train_set_labels.txt" and "test_set_labels.txt". Every line in color label files are in the format "image_name color_code". The color code is the index of a color. There are 11 different color classes labeled from 0 to 10. The name of the colors can be found in "color_codes.txt".

3 Networks and Reports

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

- **Do not use the test set in training or hyperparameter optimization.** You should only use training data for these; otherwise, you won't get any points. You can divide your training data into validation and training sets or use cross validation or similar methods.

- To get reproducible and consistent results, before anything that may use randomized algorithms(Keras initializes weights randomly and training set is shuffled randomly), you write the code below:

```
from numpy.random import seed
seed(1234)
from tensorflow import set_random_seed
set_random_seed(4321)
```

- You are expected to use fully-connected(dense) layers in your network.
- 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 data, the number of units in the final layer is always 11. Therefore, you may need to prepare the one-hot encodings of the output labels.
- You can choose a suitable optimizer. You don't have to spend too much time on it though. However, to select different learning rates, you may need to create an object of an optimizer and give it instead the name of the optimizer as a string while compiling the model. Adam optimizer can be a good initial choice.
- Try **normalizing** the data. It helps a lot. You can scale the inputs so that they reside in -1 to 1 range.
- 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):
  - 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 can also try different number of neurons in each 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 different combinations of **sigmoid**, **ReLU**, and **tanh**. You can implement them yourself if you want but they are already implemented in Keras. You are free to use them.
  - Try different learning rates. Especially ReLU may behave poorly at high learning rates and you may want to make it lower than the default value of the optimizer. For example, you can divide learning rate by $\sqrt{10}$ 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.
- Discuss how you decided the best hyperparameters. Which method did you use?
- For each k-layer network, select the best performing hyperparameters (which is done by the hyperparameter optimization) and draw its training and validation losses on the same graph and comment on it. What countermeasure did you take against overfitting? How may one understand when a network starts to overfit during training?
- An epoch is passed when all the input/output pairs are used once. You should discuss in your report which method you used to decide where to end the training procedure.
- To calculate the test accuracy you can use `np.argmax` function with its specific parameters.
- Comment on your test accuracy results. Try this: After the random initialization of the weights do not train the model and calculate the accuracy on test set. Comment on the accuracy you got. Is it similar to what you expected?

- In your report you should give the scores of different combinations of hyperparameters for specific architecture of each k-layer network. Templates are given in "report.tex" file.
- You should be able to achieve at least 0.5 accuracy in test set.
- Discuss whether accuracy is a suitable metric to measure the performance of a network for this specific dataset. Explain your reasons. (You don't have to use another metric in your experiments if you can achieve the minimum accuracy required for this homework. Just discuss how using another metric might have changed hyperparameter optimization results.)

4 Specifications

- The codes must be in python and must use Keras. Any other programming language or library will not be accepted. Python 3 is preferable but you are allowed to use Python 2 as well.
- 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.
- You have total of 3 late days for **all** your homeworks. For each day you have submitted late, you will lose 10 points. The homeworks you submit late after your total late days have exceeded 3 will not be graded.
- 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 Keras official website. The violators will be punished according to the department regulations.
- Follow the course page on ODTUCLASS or COW for any updates and clarifications. Please ask your questions on discussion section of ODTUCLASS or COW instead of e-mailing if the question does not contain code or solution.

5 Submission

Submission will be done via ODTUCLASS. If you do not have access to ODTUCLASS send your homework to this address "artun@ceng.metu.edu.tr" before the deadline. You will submit a zip file called "hw1.zip" that contains all your source code, a README file explaining which file contains what, and your report in a pdf format compiled from the given latex file.