Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.

Due Date: Thursday: 24/12/2020
CSED: Data Mining

Assignment 3
Part I
Data Classification

Objectives

1. Exploring different classification models and tune their hyper-parameters
2. Exploring different techniques for evaluating classification models
3. Learning how to analyze observed results and explain observations in a detailed report.

Problem Statement

You are given the Wisconsin Diagnostic Breast Cancer (WDBC) Data Set; the dataset is found in wdbc.data and the description is found in wdbc.names.

It is required to apply preprocessing techniques on the data set and to construct classification model using **Logistic Regression** and tune the hyper-parameters of this model. You are also required to compare the performance of models with each other.

Lab Session

1. Preprocessing

Apply preprocessing techniques (e.g., feature normalization, feature selection/ feature projection, etc.), on both the training and testing sets.

2. Classification

In this step, it is required to apply the following classification models and to perform hyper-parameter tuning using **cross-validation**.

a. Classification model:

You are required to build Logistic Regression model:
Parameters to be tuned: C
Hint: use `LogisticRegression` from `Sklearn`

b. Hyper-parameter tuning:

Use the **cross-validation** approach on the pre-processed training dataset to get the best parameter values for the classifier. Test the models trained with best obtained parameter values on the separate pre-processed testing set.

Hint: you can use `GridSearchCV` from `Sklean` to perform parameter tuning.

c. Evaluation:

Report for each model the following performance metrics: **precision, recall, and F-measure** as well as the resultant confusion matrix using the test data.

Part II
Image Classification

Objectives

1. Understand the basic **Image Classification pipeline** and the data-driven approach (train/predict stages)
2. Understand the train/val/test **splits** and the use of validation data for **hyperparameter tuning**.
3. Develop proficiency in writing efficient **vectorized** code with NumPy.
4. Implement and apply a **Softmax** classifier and a **Two-layer neural network** classifier.
5. Understand the differences and tradeoffs between these classifiers.
6. Get a basic understanding of performance improvements from using **higher-level representations** than raw pixels (e.g. color histograms, Histogram of Gradient (HOG) features)

Problem Statement

In this part, you will be doing the assignment given in Stanford university for the CS231n Convolutional Neural Networks for Visual Recognition Course:
https://cs231n.github.io/assignments2018/assignment1/

You will be required to implement the Softmax classifier and the Two-layer neural network classifier.

Lab session

Implement the requirements set in the softmax.ipynb, two_layer_net.ipynb and features.ipynb and corresponding python files.

Report Requirements

Your report should contain the following:

- Analysis and comparison of the performance results obtained in part 1 evaluation
- Analysis of the effect of hyper-parameter tuning on the performance of different classifiers in part 1
- You should change the tuning parameters used in cross validation and report best accuracy.
- You should explain how data is presented in each classifier.
- You should report models' accuracy and results.
- You should solve all inline questions in the notebooks in part 2.

Notes

- **This lab session needs time. So, try to start working on it early.**
- **Use Google Colab.**
- You should work in groups of two. Each student should answer any question in the lab session.
    - You should deliver a well-documented code as well as a report showing all your work and conclusions.
    - Copied assignments will be penalized; so not delivering the assignment would be much better.
- You should write your code in python.

References

[1] Chapter 8 of the first reference (J. Han, M. Kamber, and J. Pie, “Data Mining: Concepts and Techniques”, 3rd Edition, Morgan Kaufmann, 2012).

[2] Raschka, S. (2015). Python Machine Learning. Packt Publishing.

[3] Dua Dheeru and Efi Karra Taniskidou. 2018. UCI Machine Learning Repository. (2018). http://archive.ics.uci.edu/ml