

University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2020
Assignment 3
Feature Extraction, Classification, Deep Learning and Regularization
Due: 11:59pm April 22, 2020

Overview

Collaboration:

- You can do your work alone or in pairs.
- You can collaborate on the right tools to use and setting up your programming environment, but each team must produce their own code, report and kaggle submission.
- If you are working in a pair, report your team on this spreadsheet: <https://bit.ly/2QBixDQ>

Hand in:

- One report per team, via the CROWDMARK site in PDF format. You will need to divide the PDF up into multiple 3 files and drag and drop each onto the relevant question. You should receive an invite to crowdmark by email. You can write the report in any tool you like, it does not need to be a printout of a notebook. It can be created with LaTeX, Word, google docs or anything you like. As long as it is clear and readable with your analysis and plots saved as a pdf.
- Also submit the code/scripts needed to reproduce your work as a python jupyter notebook to the LEARN dropbox.

Broad objectives:

- Experiment with feature extraction methods to process image data and use classic data analysis and machine learning algorithms
- Set up and train a small Convolutional Neural Network in python tensorflow for classification.
- Compare the results of the methods used.

Tools: You can use any libraries available in python, tensorflow, keras, scikitlearn for this project. You need to mention explicitly which libraries you are using, any blogs or papers you used to figure out how to carry out your calculations.

Data Set - Fashion MNIST ... with a Twist

An image dataset based on "Fashion MNIST". The input features will be the same but the *label* you will be using will be new, something we have computed and created based on the data. The same dataset is hosted on two separate Kaggle competitions for the class only. The only difference between the two is that Part 1 is for non Deep Learning based solutions, and Part 2 is for Deep Learning (ie. CNN) solutions.

- Part 1:
 - Public link: <https://www.kaggle.com/c/ece-657a-w20-asg3-part-1/overview>
 - To join: <https://www.kaggle.com/t/e89aaf2776124cfd91f9b60729e9d2e3>
- Part 2:
 - Public link: <https://www.kaggle.com/c/ece657a-w20-asg3-part2/overview>
 - To join: <https://www.kaggle.com/t/80ab6ed9101f486aa7fa738a5504982c>

Other links:

- Link to another public Kaggle that works on the original Fashion MNIST and shows lots of things that can be done. ¹
- Original FashionMNIST Kaggle²
- Python setup - for this assignment you can use any python following packages that are useful, you should focus on `scikitlearn`, `keras`, `tensorflow`.

Question 1: Classification: Feature Extraction + Classical Methods (40 points)

1.1: Explanation of Design and Implementation Choices of your Model (15 points)

Describe the underlying algorithms you are using, how they work and why you chose them.

1.2: Implementation of your Design Choices (5 points)

Show some of the important code blocks to implement your model. We will also consult your full code on LEARN, so this is your chance to guide us to understand your code and how you achieved your result.

¹<https://www.kaggle.com/fuzzywizard/fashion-mnist-cnn-keras-accuracy-93/>

²<https://www.kaggle.com/zalando-research/fashionmnist>

1.3: Kaggle Competition Score (5 points)

Report the highest score your submissions received on Kaggle. If you have any explanations for varying performance by different teams explain it here.

1.4: Results Analysis (15 points)

- Runtime performance for training and testing.
- Comparison of the different algorithms and parameters you tried.
- Explanation of your model (algorithms, design choices, numbers of parameters)
- Use a **ROC curve** used for some method in your initial or results analysis such as exploring the impact on accuracy of some parameter.
- Evaluate your code with other metrics on the training data (by using some of it as test data) and argue for the benefit of your approach.

Question 2: Classification : Convolutional Neural Networks (60 points)

2.1: Design and Implementation Choices of your Model (25 points)

- Use some CNN-based approach to solve the classification problem. You can explore any architecture of the network you like.
- You should also consider exploring other ML methods and Deep Neural Network variants to solve the problem, cite sources you used : library, as well as papers or blogs. (eg. can the use of Resnet learn a better classifier than a simple CNN?)

2.3: Implementation of your Design Choices (10 points)

Show some of the important code blocks to implement your model. We will also consult your full code on LEARN, so this is your chance to guide us to understand your code and how you achieved your result.

2.4: Kaggle Competition Score (5 points)

Report the highest score your submissions received on Kaggle. If you have any explanations for varying performance by different teams explain it here.

2.5: Results Analysis (20 points)

- Runtime performance for training and testing.
- Comparison of the different algorithms and parameters you tried.
- Explanation of your model (algorithms, network architecture, optimizers, regularization, design choices, numbers of parameters)
- You can use any plots to explain the performance of your approach. But at the very least **produce two plots**, one of **training epoch vs. loss** and one of **classification accuracy vs. loss** on both your training and test set.
- Evaluate your code with other metrics on the training data (by using some of it as test data) and argue for the benefit of your approach.