University of Waterloo
ECE 657A: Data and Knowledge Modeling and Analysis
Winter 2021
Assignment 4
Classification using Deep Learning Due: 11:59pm April 20, 2021

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, you need to create the group in a few places to allow us to link both your grades
  1. Create an group on LEARN for this assignment, just pick a group with remaining slots and add your names. Each member of the group needs to do this. Your dropbox for uploading code will be associated with this group.
  2. Pick a *name* for your group or leave it as the default.
  3. In LEARN, each member of the group needs to go to `Submit/Quizzes/KaggleGroupInfoAsgX` for this assignment and enter your exact group name.
  4. In Crowdmark, for this assignment, the group members need to create a group with each other to enable joint submission.

Hand in:

- One report per team, via the Crowdmark site in an image format. You will need to divide the images up into one file for each `[CMX]` question, then 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:

- Set up and train a fully-connected Deep Neural Network for classification.
- Set up and train a Convolutional Neural Network for classification.
- Compare the results of the methods used.

Structure:

There are two datasets: Ontario COVID-10 for the DNN question, and FashionMNIST for the CNN question. Other than this, both questions have the same structure: describe your design, show your code, analyze your results, and submit your best results to Kaggle for automated scoring.

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.

Question 1: Classification Using Fully-connected and/or Recurrent networks (50 points)

Dataset 1 - Covid-19 Outcomes in Ontario

Dataset 1 is about the confirmed COVID-19 cases in Ontario. This is similar to the dataset we used in assignment 2, but here, you will work with it using deep learning and the date labels have also been included. We provide a subset of this data on LEARN under Assignment 4. The original data comes from the following file "Confirmed positive cases of COVID19 in Ontario" at: "`https://data.ontario.ca/dataset/confirmed-positive-cases-of-covid-19-in-ontario`". **NOTE: you should not use that whole dataset!** In that full dataset, the different classes are highly imbalanced so the dataset is quite challenging. For reporting results *you must use the dataset hosted*[1] *on LEARN under Assignment 4.*

The provided dataset has been sampled from the original data to be more balanced amongst the labels. The dataset includes the features age group, gender, case acquisition info, city, outbreak, latitude, and longitude. You should convert categorical features to numerical or one-hot encoded features, as appropriate. Moreover, we include the date features so you can use the recurrent networks, too. Take the feature `outcome1` as the label.

[1] However, if you want more of a challenge, and want to use live data, you can download the whole dataset from the Ontario.ca website, for additional results of your own interest.

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

[CM1] You will build a classification model using Deep Neural Networks on the Ontario COVID-19 dataset. The goal is the predict the `outcome1` label for your test set.

You can use any feed-forward, fully-connected DNN approach to solve the classification problem. You can explore any architecture of the network you like. In your implementations for deep learning, please use Tensorflow or Pytorch.

Describe the design of your network using text and figures. This includes details necessary to reproduce it such as network architecture, optimizers, activation functions, regularization methods, design choices, numbers of parameters.

You can also consider exploring questions about what type of network architecture would work well on this data. (eg. can the use of Resnet learn a better classifier than a simple fully-connected network? or Does a combination of fully-connected and RNN/LSTM perform better?) Be sure to cite any sources you used to research your approach: libraries, as well as papers or blogs.

1.2: Implementation Details (5 points)

[CM2] Show some of the critical code blocks that implement and run 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.

1.3: Results Analysis (20 points)

[CM3] Report the following results, briefly, for your trained model:

- Run-time performance for training and testing.
- Comparison of the different algorithms and parameters you tried.
- 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 validation sets.
- 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.

1.4: Kaggle Submission (5 points)

On LEARN there are two datafiles.

- `dataset`: is for you to train your models, you can divide this dataset into train, test and validate sets as needed.

- `dataset-kaggle`: is the set of input features and id's that you need to produce outputs for and format in a csv file as described on the Kaggle competition website.

You can submit multiple times on Kaggle, restrictions on submission are described on the Kaggle competition website.

Question 2: Classification : Convolutional Neural Networks

Dataset 1 - Fashion MNIST . . . with a Twist

This is an image dataset based on publicly available "Fashion MNIST" dataset. The input features will be the same as that dataset, but the *labels* you will be using will be new. We have computed and created a new label based on the data. So you must download the dataset from LEARN under Assignment 4. There you will also find a link to the Kaggle competition to post your results.

Other links:

- Link to another public Kaggle that works on the original Fashion MNIST and shows lots of things that can be done[2].
- Original FashionMNISTKaggle[3].
- Python setup - for this part you can any python following packages that are useful, you should focus on `scikitlearn, keras, tensorflow`.

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

`[CM4]` You will build a classification model using Convolutional Neural Networks on the Fashion MNIST dataset. The goal is the predict the numerical label, in the range 1-5, associated with each image. Some more details are listed on the Kaggle Competition Data page.

You can use any CNN approach and architecture you like. In your implementations for deep learning, please use Tensorflow or Pytorch.

Describe the design of your network using text and figures. This includes details necessary to reproduce it such as network architecture, optimizers, activation functions, regularization methods, design choices, numbers of parameters.

You should also consider exploring other ML methods and Deep Neural Network variants to solve the problem, (eg. can the addition of the Resnet architecture learn a better classifier than a simple CNN?) Be sure to cite any sources you used to research your approach: libraries, as well as papers or blogs.

[2] `https://www.kaggle.com/fuzzywizard/fashion-mnist-cnn-keras-accuracy-93/`
[3] `https://www.kaggle.com/zalando-research/fashionmnist`

2.2: Implementation of your Design Choices (5 points)

[CM5] Show some of the critical code blocks that implement and run 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.

2.2: Results Analysis (20 points)

[CM6] Report the following results, briefly, for your trained model:

- Run-time performance for training and testing.
- Comparison of the different algorithms and parameters you tried.
- 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 validation sets.
- 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.

2.3: Kaggle Competition Score (5 points)

On LEARN there are two datafiles.

- `dataset`: is for you to train your models, you can divide this dataset into train, test and validate sets as needed.
- `dataset-kaggle`: is the set of input features and id's that you need to produce outputs for and format in a csv file as described on the Kaggle competition website.

You can submit multiple times on Kaggle, restrictions on submission are described on the Kaggle competition website.