Emotion detection from facial features

COL 774: Assignment 4

Sem I, 2020-21

Due Date(Both Parts): January 17, 11:50 pm
Total Points: 40 + 60

Notes:

- You should submit all your code (including any pre-processing scripts written by you) and any graphs that you might plot.
- Include **a write-up (pdf) file**, which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- **You should use Python as your programming language and PyTorch as the deep learning framework.**
- **Your code should have appropriate documentation for readability.**
- You will be graded based on what you have submitted as well as your ability to explain your code.
- This assignment is supposed to be done in **teams of 2**. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. We will also **include submissions from the internet** to maintain integrity. Any cheating will result in a zero on the assignment and possibly much stricter penalties (including a **fail grade** and/or a **DISCO**).
- Submission Guidelines will be posted within a week on Piazza or MS Teams. Kindly follow them strictly and sincerely to avoid additional penalties.

The goal of this assignment is to experiment with various learning algorithms on a real world dataset. There are two parts of this assignment:

1. Non-Competitive: Where you have to implement a set of fixed learning algorithms
2. Competitive: Which is a competition where you have to try out algorithms of your choice - independent of whether they have been covered in the class or not. This assignment should provide an opportunity for you to learn new techniques/tools on your own as well as try the ones that you have already seen in the class.

In this assignment, we will deal with the problem of identifying a person's emotional state through an image of his/her facial features. The dataset comes from a set of 7 different labels (emotional states). Each image is of size 48×48 pixels and containing only 1 channel(greyscale). **Note that the dataset is imbalanced hence we will use Macro-F1 as the evaluation metric instead of accuracy.** You can read up on Macro-F1 from this reference.

1. **Competition Link**: Link
2. **Dataset Links**
   (a) **Train Dataset:** Link
   (b) **Public Test Dataset:** Link

Each row of both the files contains values corresponding to a 48×48 greyscale image. The first entry is the label of the image $\in \{0, 1, 2, 3, 4, 5, 6\}$ and the next 2304 entries contain the pixel values of the image. The image is stored in row-major order, so that the first 48 entries of the array are the channel values of the first row of the image and so on and so forth.

Train Dataset and Public Test Dataset have been provided for the non-competitive part of the assignment. Kaggle contest will contain a private test dataset without labels on which prediction will be used to evaluate the challenge.

3. **(40 points) Non Competitive Part**

   (a) **(10 points) Vanilla Neural Network**: In this part, try out a plain deep neural network taking in the image as the input. You are allowed to create Neural Network using Pytorch or use your implementation from first principle in Assignment 3. Your neural network should have 1 hidden layer with 100 perceptrons. Have softmax as the output layer. Train using cross-entropy loss and experiment with various activation functions. Comment on your observations.

   (b) **(10 points) Feature Engineering**: In the previous part pixel values are being used as features of the image. Can some features other than the absolute pixel values be used? Experiment with two feature extraction techniques commonly used in computer vision:

      i. Gabor Filters → Reference
      ii. Histogram of Oriented Gradients → Reference

   You can import skimage library for implementing above. Use the neural network architecture for part (a) while adjusting the input layer to incorporate new features. How does these changes affect your performance compared to the previous part? Comment on your observations and report your best performing design.

   (c) **(20 points) Convolutional Neural Network**: Write a program to implement a convolutional neural network with the following structure:

      i. **CONV1**: Kernel size → 3×3 . Stride → 3, Padding → 0, Input size → 1, Output size → 64.
      ii. **POOL1**: Kernel size → 2×2 . Stride → 2, Padding → 0.
      iii. **CONV2**: Kernel size → 2×2 . Stride → 2, Padding → 0, Input size → 64, Output size → 128.
      iv. **POOL2**: Kernel size → 2×2 . Stride → 2, Padding → 0.
      v. **FC1**: Fully Connected Layer with 256 outputs.
      vi. **FC2**: Fully Connected Layer with 7 outputs.
      vii. **SM**: Softmax Layer.

   All layers except for the pooling layers and the softmax layer should use ReLU as non-linearity. Use batch-normalization after every layer activation except the pooling and softmax layers.

Comment on the trade-off concerning inference time and model size besides the macro-F1 score between Neural Network and Convolutional Neural Networks. Include this in your report alongside your observations.

4. **(60 points) Competitive Part**
   Try out algorithms of your choice to maximize the performance on the test data. Improve your model as much as you can. You are strongly encouraged to try out variations of the algorithms described in the class as well as try new algorithms/models by looking up online resources.

   You will submit the labels for the test dataset on Kaggle and your performance (on Macro-F1 score) will be ranked on the leader-board. There are 2 leader-boards; a Public leader-board on which you can monitor your accuracy on 30% of the test dataset; a Private leader-board (which you cannot see) which will be used to determine the final rankings on the other 70% of the test data.

   Note: The score obtained in this part will solely depend on the quality of your predictions on the test labels. You must submit the code for your best performing model (as well as include all the details in your report) so that we can replicate the results if required.