Final Project Description
EEE 4773, Fall 2020

Title: Speech Emotion Classification

Project Due: Wednesday, December 11, 2020, 11:59 PM
Group Size: up to 4 individuals
Material Due: final report and code implementation

1. Description

In the final project, you will develop a Machine Learning algorithm to recognize (a fixed set of) emotions from speech signals. The data set is to be collected by the students enrolled in this course. You can implement this yourselves or using a package/library. You can use any packages that come as a default option with Anaconda or PyTorch. *I need to be able to run your implementation on my machine. So, be sure to get approval from me for any special packages! If I cannot run the code, you will lose a significant number of points.*

1.1. Data Set

Each group will collect part of the training set that everyone will use to train their models. In order to collect the data, you will read a sentence out loud in a particular *emotional tone*. You will record this reading using the Voice Recorder App using the following settings:

- **Microphone Adjustment:** Device auto control
- **Record file type:** .wav
- **Recording Quality:** Mono - 44kHz
- **Default File Name:** see the instructions below

- **Duration:** Trim all your voice recordings to have 2 seconds duration exactly (this is important!). If speech is cut-off, re-record your trial (also crucially important!).

The **emotion labels** are: (1) neutral, (2) calm, (3) happy, (4) sad, (5) angry, (6) fearful, (7) disgust, (8) surprise. The speech statements you should read are: (1) "Kids are talking by the door", and (2) "Dogs are sitting by the door".

Each student should record **5 trials** of each statement and each emotional label, giving a total of 80 recordings.

I recommend you save your files using a coding system, e.g. **ID-trial-statement-label**. First give a number from 1 to 4 to each team member, this is the ID. Then, for example, when team member with ID 4 is recording hers/his 5th recording trial of the statement (2) "Dogs sitting by the door" in a happy tone (emotional label 3), the file name should read "4-5-2-3.wav".

After collecting all teams' recordings, I will be partitioning this data into a **training set** (about 70%) and an ***easy* test set** (about 30%). You will be given the **training set** to fit your model. I will hold the ***easy* test set** until after you submit your code implementation. This test set will be used for grading.

I will also create a separate ***hard* test set** that will contain recordings from the classroom but also include other voices outside the classroom and other sounds/emotional tones outside the provided labels. This test set will be used for **extra credit contest** - see details below.

1.2. Project Report

You should write a report that includes the sections listed below. Your report should follow the IEEE transactions format (single spaced, double column).

- You can find a template for the IEEE transactions format here: `https://www.ieee.org/conferences/publishing/templates.html`

Focus your report on your training and testing strategies for the contest and any unique implementations. The **maximum number** of pages for the report is 4. If there are any pages beyond page 4, they will be discarded and not read or graded. It should be written with correct English grammar and spelling. Be precise - use pseudo-code or equations to be precise.

For full credit consideration, your report should include the following sections:

- *Abstract.* A summary description of the contents of the report and your findings.
- *Introduction.* Overview of your experiment/s and a literature review. For the literature review, include any references to any relevant papers for your experiment/s. So, whatever you decide to do, search the ACM and IEEE (or other) literature for relevant papers to read and refer to.
- *Implementation.* Describe and outline any specific implementation details for your project. A reader should be able to recreate your implementation and experiments from your project report. If you participate in the extra credit contest, be sure to describe the methodology on how you will identify emotional tones or other sounds that were not in the training data?
- *Experiments.* Carefully describe your experiments with the training data set and any data augmentation set you constructed or existing data sets. Include a description for the goal of each experiment and experimental findings. This is the bulk of what you will be graded on - if your experimental design is not sound or your experiments do not make sense, you will lose points.
- *Conclusions.* Describe any conclusions or things you learned from the project. Your conclusions must follow from what you did. Do not copy something out of a paper or say something that has no experimental support in the Experiments section.
- *References.* Listing of all references in IEEE bibliography format.

When writing the report as a group, I recommend you to use **Google Docs** using your UFL account. This way you can all make synchronous and simultaneous edits in your project report.

1.3. Project Code Implementation

You can use any packages that come as a default option with Anaconda or PyTorch. *I need to be able to run your implementation on my machine. So, be sure to get approval from me for any special packages! If I cannot run the code, you will lose a significant number of points.*

You can implement your algorithm using Jupyter Notebook or your favorite development environment (e.g. mine is Spyder). Your final code submission should contain 3 files:

- README file - directly editable in your team repository
- train.py or a Notebook with a function "train"
- test.py or a Notebook with a function "test". This function should receive data and labels in the same format as the training data and output an accuracy value and the predicted labels.
- if you compete in the contest, you can create a separate file for testing on the hard test set. This function should receive data and labels in the same format as the training data and output an accuracy value and the predicted labels.

1.4. Submission Details

Turn in your project report and your code on your group GitHub repository on **Friday, December 11 at 11:59 PM**. In Canvas, you should submit your GitHub URL **AND** the project report.

Be sure your repository contains the following files: **train.py** (includes a *function* that will run your training code on an input data set X and desired output vector Y. Any parameter settings must be easy to find and modify.), **test.py** (includes a *function* that will run your testing code on an input data set X. Note: Your test.py code should already be trained and have parameters set! Any parameter settings must be easy to find and modify. It should return a vector with the class label associated with each input data point X) and a concise **README.txt** file that clearly illustrates how to run your code. Your classification accuracy on a small test data set will factor into your project grade.

2. Grading Details

Your grade will be determined using the following breakdown:

- 25% Implementation
  - Turn in code that runs correctly and easily on my machine. This requires a very clear README and easy to modify parameter settings. This also requires clearly listing what packages/libraries are needed to run your code - and checking with me before the due date to ensure I have those libraries.
  - Turn in code that follows the submission requirements described above

- 25% Accuracy on "easy" blind test data set
  - The "easy" test set is composed of audio recordings for all 8 emotion labels ((1) neutral, (2) calm, (3) happy, (4) sad, (5) angry, (6) fearful, (7) disgust, and (8) surprise.
  - Your code should produce the labels '1' for 'neutral', '2' for 'calm', '3' for 'happy', '4' for 'sad', '5' for 'angry', '6' for 'fearful', '7' for 'disgust' and '8' for 'surprise'.
  - Full points on this component will be obtained if you correctly classify 90% of the blind test data or have a classification accuracy rate greater than the average classification accuracy rate of the class (whichever is lower).
- 40% Project report
  - This component will be graded based on the requirements outlined above.
- 10% Data Collection
  - You will be graded on data collection. Each person should collect 80 recordings with the settings mentioned above.

2.1. Extra Credit Contest

Your goal is to train your system to distinguish between emotion tone in speech signals. The teams with the best classification accuracy on the "hard" data set will get extra credit. The "hard" data set consists of the following labels: '1' (neutral), '2' (calm), '6' (fearful), '8' (surprise) and '-1' (unknown). There will be test data points from classes that do not appear in the training data. So, you will want to come up with a way to identify when a test point class is "unknown" or was not in the training data. The label you should return for this case is -1.

Please have your code output a class label that matches the class value in the provided training data. These should be: 1, 2, 6, 8 and -1 respectively.