Project 2 – Emotion Recognition using Physiological Signals

Final Deliverables due 04/05/2022 by 11:59pm

1. Overview, Task, and Data Description

Emotion recognition has applications in the medical field, education, defense, and commerce. Your task for this project is to use physiological signals to classify the following 10 emotions: (1) Happy; (2) Sad; (3) Surprise; (4) Pain; (5) Disgust; (6) Afraid; (7) Startled; (8) Skeptical; (9) Embarrassment; (10) Fear.

The data comes from the BP4D+ dataset, which was collected at Binghamton University in New York. The physiological signals are temporal in nature. Considering this, you will need to implement a neural network architecture, that can handle this type of data, for classification. Details regarding the dataset can be found (on Canvas) in the paper Zhang et al. Multimodal Spontaneous Emotion Corpus for Human Behavior Analysis, CVPR 2016.

Data directory is setup as follows.

- Physiological
  - Training
  - Validation
  - Testing

2. Programming Requirements

For this project, you will need to implement a neural network architecture that can handle temporal data. There are no specific requirements for the type of architecture that you need to implement, just that you can handle the temporal nature of the data. You can use the architectures that you learned about in class, as well as any other that was not discussed. The following points are requirements for the project.

1. Must be written in python using tensorflow/keras. The main script must be called Project2.py. You can have other scripts, but the main script must have this name.
2. The following metrics, to evaluate the results of your test data, must be print to the command line.
   a. Accuracy
   b. Recall (macro and micro)
   c. Precision (macro and micro)
   d. F1 score (macro and micro)
   e. Confusion Matrix
3. Your python script must take multiple command line parameters.
   a. Whether you want to run training/validation or test data.
   b. Path to dataset
   c. Model name to save (training) or load (testing)

d. Type of data you want to run

An example run would be: **python Project2.py train ./Physiological eda EDA**. This would run Project2.py in training mode using EDA data from the directory ./Physiological. The model would be saved to eda.h5.

The general format for the command line script is **python Project2.py <testOrTrain> <data_directory_path> <model_name> <data_type>.** See Table 1 below for each of the possible input values for each parameter.

Table 1. Command line parameters values.

Command Line Parameter	Possible Values
<testOrTrain>	train; test
<data_directory_path>	Any valid path
<model_name>	Any valid model name
<data_type>	EDA; DIA; volt; sys; resp; pulse; mean; mmhg; all

4. Custom File I/O must be written by you. You will need to read in the signals and class labels to train the network and run validation/testing.
   a. The data text files contain the subject ID, class, and data type in the file name. For example, the file F056_1_BP DIA_mmHg.txt is subject F056, class 1 (task T1 from paper), and it contains BP DIA data.
5. When training the network, you need to save your best model (as discussed in class). You can do this based on whatever metric you want such as lowest training loss, lowest validation loss, highest accuracy, etc. Save the model in the same directory as your python scripts. You must use the .h5 model extension (as discussed in class).
6. For testing, you must load a model to do the predictions on the test data.
7. The baseline accuracy for this project is around 40% accuracy for the ***test*** data. Your model needs to at least achieve this accuracy, if not beat it.
8. You will need to create a model for each data type. For example, if you run the script with EDA as the data type, it will only load and train the model on EDA data. If all is selected, then all data is loaded, and the model is trained on all data types at once. See Table 2 for the data types and associated command line parameters.

Table 2. Command line parameters and associated signal type.

Command Line Parameter (<data_type>)	Physiological Signal
EDA	EDA_microsiemens
mmhg	BP_mmHg
mean	LA Mean BP_mmHg
sys	LA Systolic BP_mmHg
pulse	Pulse Rate_BPM
DIA	BP Dia_mmHg

volt	Resp_Volts
resp	Respiration Rate_BPM
all	All Signals

9. The data are variable length. Each file has < 1000 rows, where each row is a data point. You will need to make sure the input data, to your network, has the same length. NOTE: we covered a way to do this in the sample code from class.

3. Paper Requirements

For the paper, you will need to turn in a 4-page plus reference paper in the IEEE double column format. The 4-pages plus references is a hard limit. You can NOT have any content, other than references, on the 5th page. You can find Latex and Word templates on Canvas. Do not change the templates (e.g., font size, etc.) This paper needs to be written like a scientific article that would be submitted to a conference so they all should have the same format. The same sections/ideas will apply in this paper, and you need to at least have the following sections (you can add more if needed).

1. Abstract (~150 words)
2. Introduction
3. Related works (related works can also be combined with intro)
   a. Give a brief overview of at least 10 papers that are related to physiological signals and emotion recognition. Note, they do NOT have to use BP4D+.
4. Method
   a. What kind of architecture did you create and why?
5. Experiments and results
   a. Show figures/tables for your experiments. What model gave the best performance and why? Tables can include each signal type and the accuracy, precision, recall, etc.
6. Conclusion
7. References
   a. At least 10 references needed. Note all references must be cited in the paper. Do not include a reference that does not have a citation in the paper.

4. Grading

Paper – 40% of grade

1. Abstract and introduction – 5% of grade
2. Related works – 5% of grade
3. Method – 10% of grade
4. Experiments and Results – 15% of grade
5. Conclusion – 5% of grade.

Code – 60% of grade

1. File I/O (reading in physiological signals for training, validation, and testing). This also includes correct formatting of data so that the network accepts it as input data – 25%.
2. Implementation of architecture that achieves/beats baseline – 30%.
3. Saving and loading best model, command line parameters, and printing metrics – 5%.

5. Deliverables

1. Python script(s).
2. Best model (.h5 format).
3. PDF of your paper detailing the project.
4. Create directory called <lastName_Project2> and place scripts, model, and PDF in this directory.
5. Zip directory up and upload to Canvas.