META-LEARNING FOR LOW-RESOURCE SPEECH EMOTION RECOGNITION

Yash Chandra (211187)

1 Implementation details

1.1 Approaches and code summary

The paper compares three approaches for training a LSTM model for Speech Emotion Recognition on low resource language.

- **Transfer-Learning:** This approach trains the base model(train.py) on source datasets and then perform transfer learning(frac_train.py) by freezing all layer weights except last two layers and training on target dataset. LSTM.py defines the LSTM model. dataloader.py is used to load source datasets while myDataloader.py is used to load target dataset.
- **Multitask-Classification:** model.py defines the LSTM model. utils.py loads train and test data in batches. Agent defines different tasks that can be performed we use only MultiTaskSeparateAgent which performs learning on source datasets in round-robin fashion. main.py call these function to perform the training and evaluation.
- **MAML:** This approach involves performing k shot learning task and then updating the model using the loss after k shot training. src folder consists of data_loader.py used to load data and model.py to define LSTM model. utils.py contains functions for saving and loading models and parameters. params.json lists the parameter values.

1.2 Files not present in original github repository

- Dataset used for the experiment were downloaded from kaggle. Processed to use only [happy, angry, sad, neutral] classes and extracting MFCCs to get final data uploaded on drive.
- Files to process (and split) test datasets were not present and were implemented. Random seed for spliting and any additional step not mentioned in paper lost which can deviate results from the paper.
- train.py and utils.py file in MAML-in-pytorch folder were implemented. This can lead to slight deviations as information like number of epochs, support and query size were lost. In this the support and query size depends on dataset.

  Support and query size for Shemo dataset were available but these will not work for other datasets and no set rule to determine or tune these. Results for different number of epochs in K shot task were calculated.

2 Dataset description

2.1 Datasets

The datasets used in this project are:

- High Resource (Source): English datasets (TESS, RAVDESS, SAVEE, IEMOCAP) and one German dataset (EMODB)
- Low Resource (Target): Italian (EMOVO), Persian (SHEMO), and Urdu (URDU)

2.2 Processing

2.2.1 Feature Extraction

These datasets consists of a .wav file with name of file indication the emotion. Emotion label was extracted from the file name and a .csv file consisting of two columns one with file path and other with emotion label was created. MFCCs features were extracted using 120 frames of the .wav file and stored in a .pkl file with file path. Only points correspoinding to labels in [happy, angry, sad, neutral] were used for training and testing purposes.

2.2.2 Splitting

- source datasets were split in train(70%), val(10%), test(20%) sets.
- target datasets were split in train(80%) and test(20%) sets.

3 Results

Table 1: MultiSER

Dataset	**F1 Score**	**Paper Result**
SHEMO	0.5467	0.61
EMOVO	0.3711	0.44
URDU	0.6204	0.60

Table 2: TransferSER

Dataset	**F1 Score**	**Paper Result**
SHEMO	0.5195	0.58
EMOVO	0.3674	0.38
URDU	0.6842	0.40

Table 3: MetaSER

eval epochs	**F1 Score**
5	0.5277
8	0.5296
20	0.5379

- All the above results are for source dataset TESS, EMODB, RAVDESS Table 1 shows the F1 scores obtained for the multiSER which are very close to the ones in the paper.
- Table 2 shows the results for TransferSER and values for SHEMO and EMOVO are very close to the results in the paper but for URDU value differs significantly. This can be due to the fact that this dataset is relatively smaller and more susceptible to bad splitting as test set is small. Even the base model performed at 0.40 f1 for URDU.
- Table 3 stores the result for SHEMO dataset on metaSER for different values of eval epochs for K shot learning. Eval epochs is the number of SGD steps taken on the K datapoints. F1 increases with more steps. These evaluations were done every 100 epochs of meta model to get the best value.
- If the evaluations are done every 1000 epochs we get f1 for 5 eval epochs case to be 0.51. this shows that the f1 can increase if the evaluation frequency is increased but these values were not specified in the paper.

Due to high computation requirement and lack of parameters results for other datasets in case of metaSER could not be computed.

4 Reasons behind abnormality or deviations observations

- Abnormal results for URDU dataset can be due to its small size leading to small test set.
- Lack of random seed and details for data preprocessing step for test datasets.
- Lack of implementation details for Train and evaluate function of metaSER(MAML)