LIGHT-SERNET: A LIGHTWEIGHT FULLY CONVOLUTIONAL NEURAL NETWORK FOR SPEECH EMOTION RECOGNITION

Antonio Scardino, Luca Distefano, Andrea Cadoli

Introduction

Detecting emotions directly from a speech signal plays an important role in effective human-computer interactions, for example in intelligent dialogue systems and voice assistants, such as Apple Siri, Amazon Alexa, etc.

In this report, we discuss the experiments that we have performed and the modifications that we have applied to the network presented in the reference paper [1].

In that paper [1], the authors propose a novel model for SER that can learn spectro-temporal information from Mel frequency cepstral coefficients (MFCC), by only making use of a CNN, which is noticeably lightweight, therefore suitable for online applications and on small embedded systems and IoT devices with limited resources. The use of CNNs not only reduces model complexity, but provides better generalization, as stated by the authors themselves.

We have developed the experiments from scratch on Google Collab in PyTorch, instead of TensorFlow as the authors of the paper did, since we had previous experience with the library and also because PyTorch offers a lot of functionalities, like the standardization of the training function.

Datasets:

The experiments have been conducted on three different datasets, with increasing size in the number of examples and different number of classes (emotions):

EMO-DB: This dataset is in German language, recorded by ten professional actors and actresses (five men and five women). The dataset includes 535 emotional utterances in 7 classes: anger, natural, sadness, fear, disgust, happiness and boredom.

We cut the audio files of this dataset into segments of 2 seconds length. If the audios were longer, we continued to cut them into 2 seconds pieces until there were not enough frames left (the remaining ones were discarded). In this dataset size optimization, we increased the samples in the dataset, without losing too much generalization, since we inspected the audio files and noticed changes in tone and frequency of the longer audio files. Otherwise, if the audio waveform were smaller than 2 seconds, we applied zero-padding until the desired dimension.

IEMOCAP: This multimodal dataset, recorded at the University of Southern California, includes 12 hours of audio-visual data divided into five sessions, recorded by male and female professional actors and actresses with scripted and improvised scenarios. The scripted part is performed for predetermined emotions, while the improvised part is closer to natural speech. The dataset includes 5531 samples with a class distribution of happiness, excitement, sadness, frustration, fear, neutral, disgust, xxx (case that the annotators were not able to have agreement on the label).

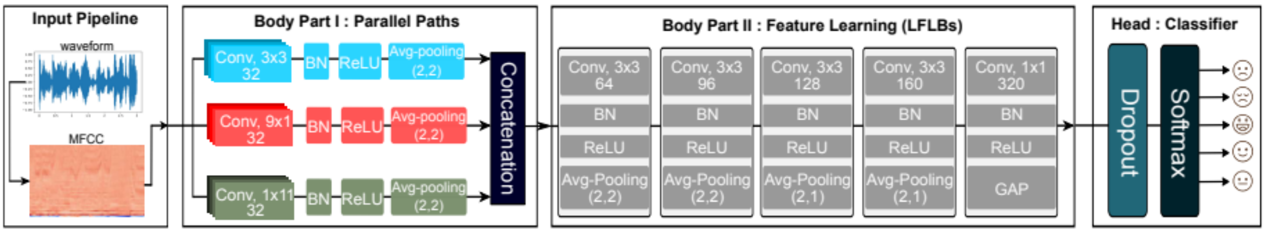
In our experiments, we only used one of the five sessions, with both scripted and unscripted registrations. We choose only one of those, otherwise the dataset would have been too large, and the computation capabilities of Google Colab, the tool we used to train our model, are limited: the training time session could persist for several grueling hours. Each audio registration contains multiple emotions, so it needed to be trimmed in order to be ready for training: one emotion per sample. Therefore, we followed the indications present in the "EmoEvaluation" file to correctly trim the registrations. In the end, the audio waveforms were 1800. Furthermore, the waveforms were cut to 3 seconds and if they were shorter zero-padding was applied.

ASVP-ESD:

The ASVP_ESD database contains 12625 audio files in the Audio directory (with additional 1204 files for babies' voices in 3 folders range in bonus directory); It is an emotional-based database, containing speech and non-speech emotional sound; The audio was recorded and collected from movies, tv shows, YouTube channels, and others website. Compared to other publicly available emotional databases, ASVP-ESD is more realistic and non-scripted with no language restriction. Emotional sounds include boredom (sigh, yawn), neutral, happiness (laugh, gaggle), sadness(cry), anger, fear (scream, panic), surprise (amazed, gasp), disgust(contempt), excite (Triumph, elation), pleasure(desire), pain(groan), disappointment. The average length of the file is between 0.5 to 20 seconds, for a total of more than 11 hours.

Like in the IEMOCAP dataset, the audios were cut to 3 seconds and if an audio was shorter than the cut length, zero-padding was applied.

Architecture Design



Input pre-processing

The pre-processing pipeline is the same of the paper: after normalizing audio signals between −1 and 1, we apply a 1024-point Fast Fourier transform (FFT) to every frame, using a Hamming window to split the audio signal into 64-ms frames with 16ms overlaps, as they can be considered as quasi-stationary segments. The spectrogram goes through a Mel-scale filter bank analysis, in a range of 40 Hz - 7600 Hz. In the Mel scale more importance to the higher frequencies is given, which are more present in signals transmitted by voice. The MFCCs of each frame are then calculated using an inverse discrete cosine transform, where the first 40 coefficients are selected to train the model.

Feature Extraction

The input image goes through three parallel paths and the result is concatenated. The usage of big receptive fields brings better accuracy results. Bigger receptive fields can be obtained by:

- increasing the numbers of layers (deeper network) and with deeper convolutions;
- through pooling functions and increased stride.

The increasing number of layers can, however, cause overfitting as the model parameters grow. The reason why deeper networks have better receptive fields is because for every added layer the receptive field increases by the dimension of the kernel. In the network a specific kernel is set for every path:

1. 1x11 to extract temporal features;
2. 9x1 to extract frequency/spectral features;
3. 3x3 to extract spectral-temporal features.

In this way the computational complexity is lower than the one there would be in the case in which a single path with bigger number of parameters was used. We performed a lot of experiments modifying the parameters of the network and convolutions e.g., kernel size, stride, etc. Furthermore, we introduced three residual connections inside the parallel paths, one in each one of them. The input is added at the end of each parallel path, which permits better gradient backpropagation without influencing the forward pass. Skip connections brought us better result as they avoid the vanishing gradient problem.

Feature Learning

In this block there are five LFLBs (Local Feature Learning Blocks) with different configurations applied to the low-level features, taken from the last CNN block, whose objective is to capture high-level features.
The LFLB consists of:

- A layer for convolutions,
- A layer for batch normalization (BN),
- A layer for ReLU,
- A layer for avg pooling.

The last LFLB uses a global average pooling (GAP) in order to allow training on different dataset sizes without changing the architecture.

Inside the whole network, we chose PReLU rather than ReLU in order to keep negative values, not sending them to zero which also helps the gradient flow. In the ASVP dataset we increased the number of convolution channels in Body 1 and Body 2, as the audio files could be classified in more emotion classes than the other two datasets.

Classifier Head

The classification layers include only:

- A fully-connected layer;
- And a dropout layer, necessary for reducing overfitting

In our implementation there is no Softmax activation function inside the network because PyTorch cross-entropy loss function already includes a Softmax.

Experiments and results

Experimental setup

We use the PyTorch v1.12.1 Python Library, to implement our experiments. The models are trained on:

- Nvidia Tesla P100 for 200 epochs for IEMOCAP dataset;
- Nvidia Tesla P100 for 300 epochs for ASVP-ESD;
- Nvidia Tesla T4 for 300 epochs for EMOD-DB.

The batch size for all datasets is 32. We have used the Adam optimizer with an initial learning rate of 10^{-4} for IEMOCAP and ASVP-ESD, 10^{-5} for EMO-DB. The learning rate is exponentially decreased each epoch with a rate of $e^{-0.9}$. In order to avoid overfitting, as in the paper we used batch normalization, dropout and a weight decay of 10^{-6}.

Furthermore, we also applied a 10-fold cross-validation, however this only increased the computational overhead for the training time without any increase in accuracy.

Metrics

Three metrics were used to evaluate the proposed model: unweighted accuracy, weighted accuracy, and F1-score. In particular, the weighted accuracy is the one on which we based the choice of the best epoch. In fact, considering the results obtained by this metric in the validation step, we saved the weights of the network to be then used to validate the performances on the whole dataset using a custom accuracy function created by us. We considered the weighted accuracy since there is data imbalance among classes of datasets, which means that the network will then be biased into learning only certain features.

Impact of input length

We have evaluated the proposed model for input lengths of 3 seconds for IEMOCAP and ASVP-ESD datasets and 2 s for EMO-DB. We tried also with higher inputs, and we obtained same results but with higher computational cost, due to the dimension of the image, and peak memory usage.

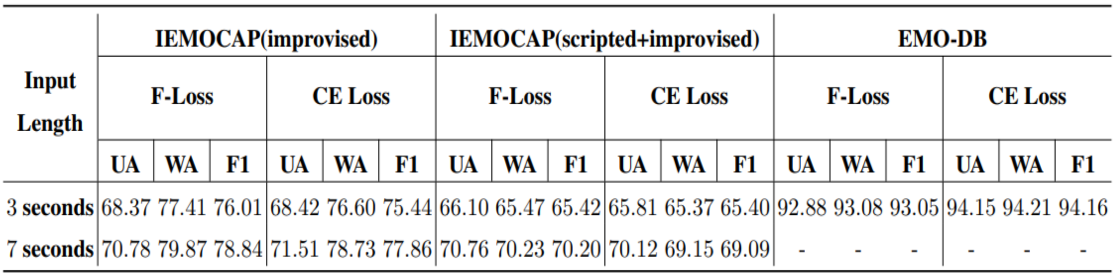
Comparison

Our results

Dataset	EMO-DB	IEMOCAP	ASVP-ESD
Accuracy	78.63	32.78	49.37
F1	78.63		
Weighted Accuracy	82.05	49.75	65.68
Focal Loss	0.04	0.17	0.07
CE Loss	0.68	1.67	1.11

Accuracy on:	EMO-DB	IEMOCAP	ASVP-ESD
Whole Dataset	94.40	43.37	81.90
Training Dataset	96.39	44.81	85.55
Validation Dataset	86.44	37.63	67.28

Original paper results



Input Length	IEMOCAP(improvised)						IEMOCAP(scripted+improvised)						EMO-DB					
	F-Loss			CE Loss			F-Loss			CE Loss			F-Loss			CE Loss		
	UA	WA	F1	UA	WA	F1	UA	WA	F1	UA	WA	F1	UA	WA	F1	UA	WA	F1
3 seconds	68.37	77.41	76.01	68.42	76.60	75.44	66.10	65.47	65.42	65.81	65.37	65.40	92.88	93.08	93.05	94.15	94.21	94.16
7 seconds	70.78	79.87	78.84	71.51	78.73	77.86	70.76	70.23	70.20	70.12	69.15	69.09	-	-	-	-	-	-

While with EMO-DB and ASVP-ESD datasets the results follow the ones of the original code, IEMOCAP has significantly worse results (10% less). This is because the audio files were stereo, therefore we followed a standard procedure in literature and averaged the two. In this way, the quality of the audio files was lost. Finally, as already said before the use of a CNN reduces complexity and provides better generalization, with respect to other networks found in literature, e.g. LSTMs, Transformers etc.

Bibliography

[1] LIGHT-SERNET: A LIGHT WEIGHTFULLY CONVOLUTIONAL NEURAL NETWORK FOR SPEECH EMOTION RECOGNITION by Arya Aftab, Alireza Morsali, Shahrokh Ghaemmaghami, Benoit Champagne