Speech emotion recognition with deep learning

Conference Paper · February 2017

DOI: 10.1109/SPIN.2017.8049931

3 authors, including:

Pavol Harár
University of Vienna
20 PUBLICATIONS **229** CITATIONS

SEE PROFILE

Speech Emotion Recognition with Deep Learning

Pavol Harár[1], Radim Burget[1] and Malay Kishore Dutta[2]

Abstract— **This paper describes a method for Speech Emotion Recognition (SER) using Deep Neural Network (DNN) architecture with convolutional, pooling and fully connected layers. We used 3 class subset (angry, neutral, sad) of German Corpus (Berlin Database of Emotional Speech) containing 271 labeled recordings with total length of 783 seconds. Raw audio data were standardized so every audio file has zero mean and unit variance. Every file was split into 20 millisecond segments without overlap. We used Voice Activity Detection (VAD) algorithm to eliminate silent segments and divided all data into TRAIN (80%) VALIDATION (10%) and TESTING (10%) sets. DNN is optimized using Stochastic Gradient Descent. As input we used raw data without and feature selection. Our trained model achieved overall test accuracy of 96.97% on whole-file classification.**

I. Introduction

The primary objective of Speech Emotion Recognition (SER) is to aid human to machine interaction. Despite the fact a lot of progress has been made in the area of Speech Recognition (SR) in past years, still there is a need for a better computer understanding of human emotions which could lead to further improvement in human to machine interaction systems [1].

The ideal way to reach this objective, as the trends are showing, might be to create an end-to-end learning algorithm, which is capable of processing raw input signal directly resulting in desired performance with as little human knowledge and work as possible [2]. Hence, in this paper we investigate the possibilities in unison with this idea.

Nowadays, we are at the dawn of Deep Learning (DL) because in a short time it has dramatically improved the state-of-the-art in many domains including SR. This approach allows us to use complex multi-layer models that learn representations of data with multiple levels of abstraction. SER is not an exception since convolutional nets as well as recurrent nets are applicable for solving this problem. The main advantage of DL is the fact that it requires very little engineering by hand, and it can benefit from today's increases of data amounts and computational power [3].

The reminder of this paper is organized as follows. Section II introduces the related papers in this area of expertise. In section III, the methodology of our experiment will be discussed. The results will be presented in section IV. Conclusions will be drawn in section V.

[1] Dept. of Telecommunications, Brno University of Technology, Brno, Czech Republic harar@phd.feec.vutbr.cz, burgetrm@feec.vutbr.cz

[2] Dept. of Electronics and Communication Engineering, Amity University, Sector 125, Noida, 201313 Uttar Pradesh, India mkdutta@amity.edu

II. Related work

Most of related papers published in past decade use spectral and prosodic features extracted from raw audio signals prior to recognition itself. Usually Mel frequency cepstrum coefficient (MFCC), Mel energy spectrum dynamic coefficient (MEDC), Linear prediction cepstrum coefficient (LPCC), Perceptual linear prediction cepstrum coefficient (PLP), pitch, formants and energy. After extraction, several classifiers have been proposed for this task including Support Vector Machines (SVM), Hidden Markov Models (HMM), Artificial Neural Networks (ANN), Bayesian Networks (BN), K-nearest Neighbors (KNN) or Gaussian mixture models (GMM) [4][5][6].

Yixiong Pan in [7] used SVM for 3 class emotion classification on Berlin Database of Emotional Speech [8] and achieved 95.1% accuracy which is to our knowledge the best result yet published in this particular matter.

S. Lalitha in [9] used pitch and prosody features and SVM classifier reporting 81.1% accuracy on 7 classes of the whole Berlin Database of Emotional Speech. Yu Zhou in [10] combined prosodic and spectral features and used Gaussian mixture model super vector based SVM and reported 88.35% accuracy on 5 classes of Chinese-LDC corpus. Fei Wang used combination of Deep Auto Encoder, various features and SVM in [5] and reported 83.5% accuracy on 6 classes of Chinese emotion corpus CASIA.

In contrast to these traditional approaches a more novel papers have been published recently employing Deep Neural Networks into their experiments with the promising results.

Jianwei Niu in [11] used various features in their recognition system and combined DNN with HMM reporting 92.3% accuracy on 6 classes of 7676 spoken Mandarin Chinese sentences. H.M. Fayek in [12] explored various DNN architectures and reported accuracy around 60% on two different databases eNTERFACE [13] and SAVEE [14] with 6 and 7 classes respectively.

Sadly, we are not aware of any paper that used Deep Learning for Speech Emotion Recognition on Berlin Database of Emotional Speech.

III. Methodology

A. Data

The data set we used was German Corpus (Berlin Database of Emotional Speech) that contains about 800 sentences (7 emotion classes * 5 female and 5 male actors * 10 different sentences + some second versions).

All sentences were recorded in an anechoic chamber using high-quality equipment with sampling frequency of 48kHz and later downsampled do 16kHz (mono) [8].

This preliminary experiment was conducted on a smaller subset of this corpus containing 271 labeled recordings with total length of 783 seconds. Because of nonequality between classes and in order to get comparable results with [7], we used all sentences from all actors but only from 3 emotional states: angry (127 recordings, 334 seconds), neutral (79 recordings, 186 seconds), sad (65 recordings, 263 seconds). To remove the silent parts of the audio signal as depicted in Fig. 1 a Google WebRTC voice activity detector [15] (VAD) was incorporated into our preprocessing. All audio files were standardized to have zero mean and unit variance.

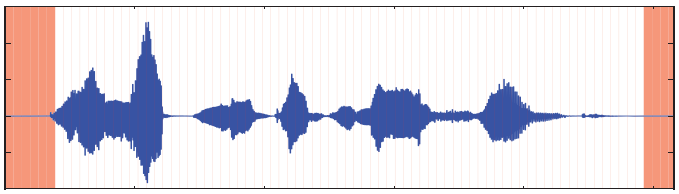

Fig. 1. An example of silent segment detection on raw audio signal of uttrance 08a01Wa.wav

We split every file into 20 millisecond chunks with no overlap - vectors of length 320 (16 kHz * 20ms) obtaining total number of 39052 segments. Then we removed the 3098 silent segments according to VAD and split the prepared data into TRAINING (79.56%), VALIDATION (9.84%) and TESTING (10,60%) sets. To obtain an even distribution of all classes in training and validation sets we adjusted the amounts of segments used from each class according to the class with the smallest number of segments available, hence we used only 73.86% of valid (non-silent) speech segments resulting in 21129 training segments (7043 for each class), 2613 validation segments (871 for each class). The rest was used as testing data with 2814 segments from 33 audio files (11 angry, 12 neutral, 10 sad). All segments used for testing were taken from files that have not been seen by DNN during training or validation.

B. DNN architecture

As the first two layers of our model we used convolutional layer [16] with 32 kernels of size 7 x 1 succeeded with average pooling layer [17]. The third and fourth layers were also convolutional with 32 kernels of size 13 x 1 again succeeded with average pooling. The last two convolutional layers had 16 kernels again of size 13 x 1. After the last convolutional layer we divided the network to two branches. Both branches consisted only of one pooling layer. One with average pooling and the second with max pooling [17] which were afterwards flattened and concatenated back to main branch.

From this point on, the DNN consisted of only fully connected layers. The first of size 480, the second one of size 240 and the last one was an output Softmax layer [18] with 3 output neurons. All pooling layers were used with pool size 2. All convolutional layers border mode was set to 'valid' therefore no zero padding was performed on borders.

For regularization of the DNN, we used a 0.1 dropout [19] for all convolutional and fully connected layers except the last layer with 0.2 dropout. Relu as activation function [20] was used for all layers except the Softmax output layer. All layers were initialized using Glorot uniform initialization [21]. This whole DNN had overall 468003 parameters and its whole architecture is depicted in Fig.2.

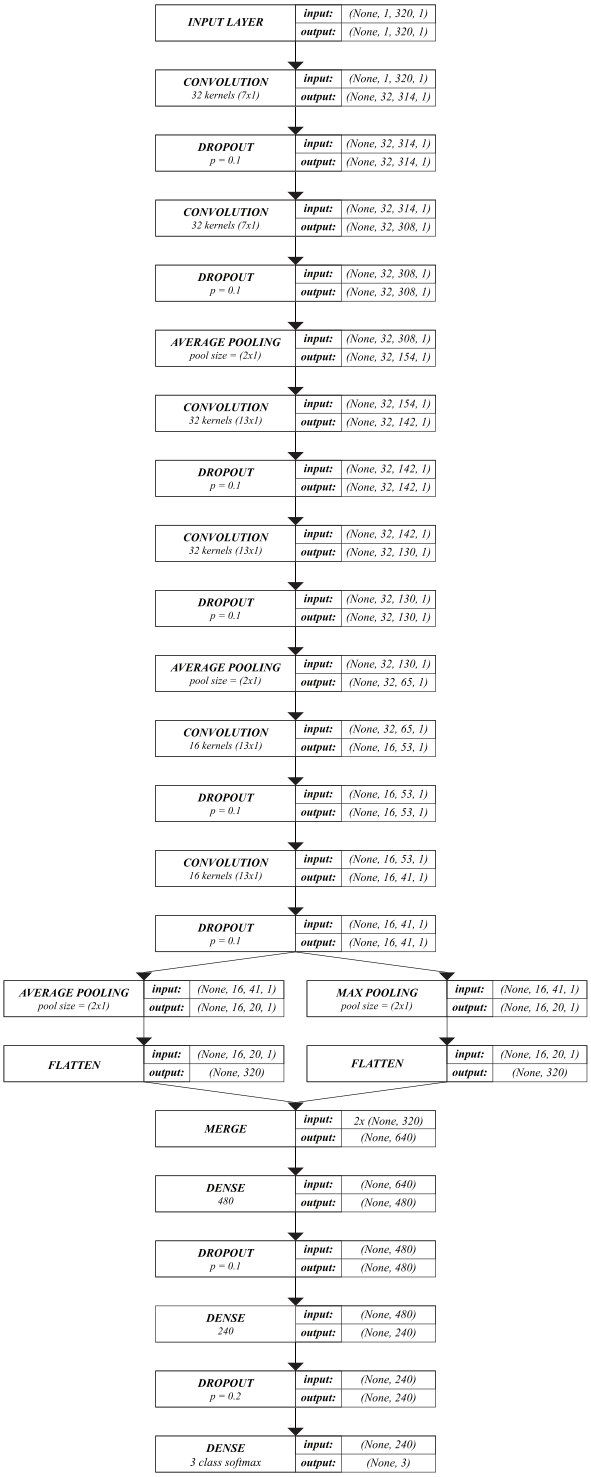

Fig. 2. Detailed architecture of proposed DNN with in/out vector shapes.

C. Experiment

For training our proposed model we utilized Stochastic Gradient Descent algorithm with fixed learning rate of 0.11 to optimize a Binary crossentropy loss function also known as logloss [22].

The input data were presented to the DNN in batches of size 21 in multiple epochs (iterations). Every batch contained exactly the same number of segments from every class and each sample succeeded with a sample of different class (e.g. angry, neutral, sad, angry, neutral, sad, angry ...).

The last batch was populated with the remaining number of segments to complete an epoch and therefore might be smaller if necessary, yet all previous rules of equality remained honored. At each epoch the data were presented to DNN in different order. Since batch size can vary, vector shapes in Fig.2 start with "None" instead of scalar.

To eliminate overfitting we set the patience equal to 15. That means the experiment was terminated if no progress on validation loss had been made for more than 15 epochs of training. The best results were recorded after the 38th epoch of training.

We utilized the capabilities of Keras [23] and Theano [24] frameworks to build the DNN model and accelerate the training on GPU (NVIDIA GeForce GTX 690). The whole 38 epochs long training took only 4.12 minutes to finish. All hyperparameters were tuned based on the validation results.

Using the trained model we were able to obtain the prediction probabilities for each class of each segment from validation and testing sets. We took the maximum value from predicted probabilities to denote the predicted class.

It was of course important to deliver the final results for the whole audio files. For that purpose we computed the average probability of all segments belonging to the particular file and used it to denote the final predicted class. The value of average probability can be viewed as confidence of prediction.

IV. Results

In order to perform 3 class SER predictions on 33 testing audio files of German speech, we built a deep neural network model consisting of convolutional, pooling and fully connected layers. We trained it with raw, standardized input data cleared of silent segments using mini-batches of size 21 over 38 epochs on GPU. The whole training took only 4.12 minutes.

The training and validation sets contained exactly the same number of segments per class as can be seen in Tab. I. The DNN predicted 545 validation segments to belong to a wrong class as opposed to 2068 correct predictions resulting in 79.14% validation accuracy. The precision, recall, f1-score and overall accuracy of validation segments is shown in Tab. II.

Tab. III shows the DNN predicted 633 testing segments to belong to a wrong class as opposed to 2181 correct predictions resulting in 77.51% testing accuracy. The precision, recall, f1-score and overall accuracy of testing segments is shown in Tab. IV.

TABLE I. THE CONFUSION MATRIX OF VALIDATION SEGMENTS

VALIDATION confusion matrix (segments)				
	true: angry	***true: neutral***	***true: sad***	***no. of segments***
pred.: angry	779	74	18	871
pred.: neutral	76	610	185	871
pred.: sad	14	178	679	871

TABLE II. THE CLASSIFICATION REPORT OF VALIDATION SEGMENTS

VALIDATION classification report (segments)			
class	***precision***	***f1-score***	***recall***
angry	0.90	0.90	0.89
neutral	0.71	0.70	0.70
sad	0.77	0.77	0.78
		overall accuracy:	79.14%

TABLE III. THE CONFUSION MATRIX OF TESTING SEGMENTS

TESTING confusion matrix (segments)				
	true: angry	***true: neutral***	***true: sad***	***no. of segments***
pred.: angry	837	25	6	868
pred.: neutral	92	551	235	878
pred.: sad	39	236	793	1068

TABLE IV. THE CLASSIFICATION REPORT OF TESTING SEGMENTS

TESTING classification report (segments)			
class	***precision***	***f1-score***	***recall***
angry	0.86	0.91	0.96
neutral	0.68	0.65	0.63
sad	0.77	0.75	0.74
		overall accuracy:	77.51%

TABLE V. THE CONFUSION MATRIX OF TESTING FILES

TESTING confusion matrix (whole files)				
	true: angry	***true: neutral***	***true: sad***	***no. of wav files***
pred.: angry	11	0	0	11
pred.: neutral	0	11	1	12
pred.: sad	0	0	10	10

TABLE VI. THE CLASSIFICATION REPORT OF TESTING FILES

TESTING classification report (whole files)			
class	***precision***	***f1-score***	***recall***
angry	1.00	1.00	1.00
neutral	1.00	0.96	0.92
sad	0.91	0.95	1.00
		overall accuracy:	96.97%

After combining the predictions of segments, only one file's emotion class out of 33 testing files has been predicted incorrectly as confusion matrix shows in Tab. V. As shown in Tab. VI, our DNN achieved 96.97% accuracy on speech emotion recognition task on testing files. The average confidence of file prediction was 69.55%.

V. Conclusion

The objective of this paper was to predict the emotional state of a person from a short voice recording split into 20 millisecond segments. Our method achieved 96.97% accuracy on testing data with the average confidence of 69.55% on file prediction.

Our approach is context independent which means that all audio segments were classified independently. The DNN thus had no knowledge of the actual context of what the actor is saying nor did it have any knowledge of the rhythm etc. On one hand, this can be viewed as an advantage, but we think that context dependent approach as in [2] using recurrent nets might significantly improve the results in this.

Even though the resulting accuracy is high, our future work will try to further improve the approach, by incorporating recurrent neural networks, using over-sampling or bigger data sets, so the model is capable of bringing the satisfying results on more than 3 classes and across multiple data sets with higher accuracy on validation sets, higher confidence of predictions and higher reliability on real-world data.

Acknowledgment

Research described in this paper was financed by the National Sustainability Program under grant LO1401.

References

[1] El Ayadi, M., Kamel, M.S. and Karray, F., 2011. Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition, 44(3), pp.572-587.

[2] Trigeorgis, G., Ringeval, F., Brueckner, R., Marchi, E., Nicolaou, M.A. and Zafeiriou, S., 2016, March. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5200-5204). IEEE.

[3] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. Nature, 521(7553), pp.436-444.

[4] Chakraborty, R. and Kopparapu, S.K., 2016, July. Improved speech emotion recognition using error correcting codes. In Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on (pp. 1-6). IEEE.

[5] Fei, W., Ye, X., Sun, Z., Huang, Y., Zhang, X. and Shang, S., 2016, June. Research on speech emotion recognition based on deep auto-encoder. In Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2016 IEEE International Conference on (pp. 308-312). IEEE.

[6] Vyas, G., Dutta, M.K., Riha, K. and Prinosil, J., 2015, October. An automatic emotion recognizer using MFCCs and Hidden Markov Models. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2015 7th International Congress on* (pp. 320-324). IEEE.

[7] Pan, Y., Shen, P. and Shen, L., 2012. Speech emotion recognition using support vector machine. International Journal of Smart Home, 6(2), pp.101-108.

[8] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W.F. and Weiss, B., 2005, September. A database of German emotional speech. In Interspeech (Vol. 5, pp. 1517-1520).

[9] Lalitha, S., Madhavan, A., Bhushan, B. and Saketh, S., 2014, October. Speech emotion recognition. In Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on (pp. 1-4). IEEE.

[10] Zhou, Y., Sun, Y., Zhang, J. and Yan, Y., 2009, December. Speech emotion recognition using both spectral and prosodic features. In 2009 International Conference on Information Engineering and Computer Science (pp. 1-4). IEEE.

[11] Niu, J., Qian, Y. and Yu, K., 2014, September. Acoustic emotion recognition using deep neural network. In Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on (pp. 128-132). IEEE.

[12] Fayek, H. M., M. Lech, and L. Cavedon. "Towards real-time speech emotion recognition using deep neural networks." Signal Processing and Communication Systems (ICSPCS), 2015 9th International Conference on. IEEE, 2015.

[13] Martin, O., Kotsia, I., Macq, B. and Pitas, I., 2006, April. The eNTERFACE'05 audio-visual emotion database. In 22nd International Conference on Data Engineering Workshops (ICDEW'06) (pp. 8-8). IEEE.

[14] Jackson, P. and Haq, S., 2014. Surrey Audio-Visual Expressed Emotion(SAVEE) Database.

[15] Google WebRTC. https://webrtc.org/. Accessed 10 Oct 2016

[16] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

[17] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

[18] Bridle, J.S., 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Neurocomputing (pp. 227-236). Springer Berlin Heidelberg.

[19] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), pp.1929-1958.

[20] Nair, V. and Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 807-814).

[21] Glorot, X. and Bengio, Y., 2010, May. Understanding the difficulty of training deep feedforward neural networks. In Aistats (Vol. 9, pp. 249-256).

[22] Zhang, T., 2004, July. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In Proceedings of the twenty-first international conference on Machine learning (p. 116). ACM.

[23] Chollet, F., 2015. Keras: Deep learning library for theano and tensorflow.

[24] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D. and Bengio, Y., 2010, June. Theano: A CPU and GPU math compiler in Python. In Proc. 9th Python in Science Conf (pp. 1-7).