Neural networks for compressing and classifying speaker-independent paralinguistic signals

Seokhyun Byun, Seunghyun Yoon and Kyomin Jung
Dept. of *Electrical and Computer Engineering, Seoul National University*, Seoul, Korea
{byuns9334, mysmilesh, kjung}@snu.ac.kr

***Abstract*—Recognizing and classifying paralinguistic signals, with its various applications, is an important problem. In general, this task is considered challenging because the sound information from the signals is difficult to distinguish even by humans. Thus, analyzing signals with machine learning techniques is a reasonable approach to understanding signals. Audio features extracted from paralinguistic signals usually consist of high-dimensional vectors such as prosody, energy, cepstrum, and other speech-related information. Therefore, when the size of a training corpus is not sufficiently large, it is extremely difficult to apply machine learning methods to analyze these signals due to their high feature dimensions. This paper addresses these limitations by using neural networks' feature learning abilities. First, we use a neural network-based autoencoder to compress the signal to eliminate redundancy within the signal feature, and we show that the compressed signal features are competitive in distinguishing the signal compared to the original features. Second, we show by experiment that the neural network-based classification model almost always outperforms nonneural methods such as logistic regression, support vector machines, decision trees, and boosted trees.**

***Index Terms*—computational paralinguistics, neural networks**

I. Introduction

Neural network-based models have achieved state-of-the-art performances in diverse applications, such as computer vision, neural machine translation, recommendation systems, and other task-oriented areas [1], [2]. Along with such impressive advancements, statistical speech processing has also demonstrated many advantages when adopting a neural network-based architecture. For instance, Amodei et al. [3] demonstrated that a convolutional architecture with a recurrent architecture can achieve great performance in speech recognition due to its abilities to learn more salient features in the time domain and temporal dependencies within the utterance. Synthesizing speech with a neural network architecture has also been successful [4] by utilizing the neural network's feature learning ability to incorporate each text-to-speech module to reduce extensive domain expertise and complexity.

In addition, diverse areas exist in the area of paralinguistic signals, such as analyzing the tone/pitch of the voice, nuance, and speech with upper respiratory symptoms [5]. Including these, all paralinguistic studies are receiving growing attention providing considerable assistance in medical science, psychology, and general engineering fields.

However, one of the biggest difficulties in analyzing paralinguistic signals lies in its ambiguity that is indistinguishable even by humans. Thus, researchers have adopted various aspects of a signal, such as prosody, energy, and cepstrum, to analyze the signal. In addition, only a small number of paralinguistic signals in a dataset is usually acquired in real situations for training the model, implying that the model is not able to sufficiently learn the feature representations. To address this issue, Sahu et al., [6] used adversarial autoencoders for dimension reduction and showed that compressed signal representations do not significantly harm overall emotion recognition performance by comparing classification accuracy in original/compressed feature settings.

In this paper, we suggest using various machine learning techniques, such as autoencoders (AE), principal component analysis (PCA), and linear discriminant analysis (LDA), for feature dimension reduction on two different feature sets extending the research in [6]. With the compressed features, we adopt machine learning models such as multilayer perceptron (MLP), support vector machine (SVM), logistic regression (LR), decision tree (DT), and boosted tree (XGB) for classifying the paralinguistic signals.

Experimental results show that most of the models trained with the compressed features provide competitive classification accuracy compared to that of the models trained with original features. In particular, the accuracy with AE-compressed features reached the highest, even overwhelming the original features in some cases. We strongly believe that our approach lessens the insufficient training corpus problem by reducing the redundancy in the high-dimensional features. For the classifier model, the MLP almost always outperforms other models in classifying the signal in the compressed/original feature setup. Hence, we suggest utilizing MLP based on AE-compressed features for efficient signal classification.

II. Related Work

One of the most prominent problems in paralinguistic signal processing studies is speech emotion recognition, as it is a crucial factor in optimal human-computer interaction, including dialog systems. The challenge that speech emotion recognition poses is predicting the emotion behind the speech and classifying it into one of the following categories: happy, sad, neutral, and angry. To achieve this goal, classic machine learning algorithms, such as the hidden Markov model (HMM) and support vector machines (SVM) have been adopted [7], [8]. Later, several studies started utilizing deep learning architectures for speech emotion recognition. A

978-1-5386-7789-6/19/$31.00 ©2019 IEEE

feedforward neural network has been used to extract window-level features, summarize them into a single utterance-level feature with some statistical functions, and generate an output prediction with an extreme learning machine (ELM) [9]. Since the deep neural network with ELM model estimates the probability for each frame of small window length, Lee et al. [10] suggested a deep bidirectional long short-term memory (LSTM) architecture on a low-level acoustic feature set to incorporate long contextual effect and to avoid the vanishing gradient problem. As an effort to consider regionally salient information within a signal, Aldeneh et al. . [11] extracted 40-dimensional log Mel filterbank features (MFBs) from the raw signal and applied convolution layers, max-pooling layers, and dense layers followed by a softmax layer to categorize each utterance into emotion labels.

III. Task Description

This paper concentrates on two paralinguistic tasks that involve classification problems using a small amount of data (i.e., 502, 3342 training instances) with high-dimensional features. Our objective in each task is to build a model that achieves the best classification accuracy. To evaluate the performance of the model, we use weight accuracy recall (WAR), the ratio of correct predictions to the whole test samples, which is widely used in this study.

The heartbeat classification task: This task focuses on distinguishing anomalies of heartbeat sounds. The given types of heartbeat sounds are: *"normal"*, *"mild"*, and *"moderate"*/*"severe"* (heart disease).). For this task, the Heart Sounds Shenzhen (HSS) corpus is gathered from 170 subjects (115 male/55 female, ages ranging from 21 to 88). The data set includes 502, 180 and 163 utterances for training, validation, and testing, respectively.

The atypical affect classification task: An emotion of disabled speakers is recognized. The emotion classes are defined as *"angry"*, *"happy"*, *"sad"* and *"neutral"*. To gather the Emotional Sensitivity Assistance System for People with Disabilities (EmotAsS) dataset, 15 mentally, neurologically, or physically disabled individuals (7 male / 8 female, ages ranging from 20 to 58) were recorded spontaneously in a familiar room in their workplace. Under the supervision of a psychologist, five different tasks were performed to generate emotional utterances: describing images, talking about specific topics, telling a story of a pictured book, introducing their everyday business, and playing together games. The corpus contains 3342 and 4186 utterances for training and testing, respectively [12], [13].

IV. Proposed framework

A. Feature sets

To explore feature reduction effects on various feature settings, we use the Interspeech 2018 ComParE [14] and emobase features extracted using the openSMILE toolkit [15] on all task corpora. These features include signal processed features such as mel-frequency cepstral coefficients (MFCC), F0, and log mel frequency. In addition, they contain statistical functional features within certain time frames. The dimensions of the features within each utterance is fixed to 6,373 and 1,582 in ComParE and emobase, respectively.

B. Compression methods

In this study, various machine learning techniques are suggested for feature dimension reduction to investigate its efficacy on two different feature sets. These techniques include not only classical approaches such as PCA and LDA but also recent neural network-based AEs.

Principal component analysis (PCA): PCA is an unsupervised learning technique that aims to identify the principal components that maximize the variance of transformed data points. Compressed test features are obtained by transforming original test features with pretrained PCA parameters. We use PCA for compressing features into 2- and 200-dimensional spaces (PCA-2 and PCA-200, respectively).

Linear discriminant analysis (LDA): Unlike PCA, the LDA uses the label information of the training set so that it minimizes the distance between the same labels and separates data points belonging to different labels as much as possible. To perform this, we find a linear transformation that maximizes the ratio of "between class scatter" and "within class scatter". Unlike PCA and AE, we set the dimension of the latent code vector as $N-1$, where N is the number of classes in the training corpus.

Autoencoder (AE): This is basically a neural network encoding the original feature vector into a latent vector of small dimensions and decoding the latent vector into the reconstruction vector of the original dimension. We use the mean square error (MSE) between the original feature vector and the reconstruction vector as a loss function, which aims to efficiently contain information on the original features in the latent vector. For the implementation, we use three dense layers with a Selu activation function [16] and batch normalization to stabilize the training procedure. To avoid overfitting, we adopt early stop criteria when the validation MSE loss starts to increase. With the trained model, we extract latent vectors by encoding the original features of the training and testing dataset. In all tasks, ComParE and emobase features of the training set are encoded into 400- and 200-dimensional Euclidean space, respectively.

C. Classification methods

Four classical classifiers and neural network-based models are used for our tasks.

Logistic regression (LR): As a basic classification model, it uses a logistic function with trainable parameters to assign a probability for each label given each feature. The parameters are updated through gradient descent.

Support vector Machine (SVM): The goal of the support vector machine is to find a decision boundary that maximizes the classification margin between the data points in different

TABLE I
Model performance comparisons for the heartbeat task. Top-2 performances are marked as bold.

Feature	Compression	LR	SVM	DT	XGB	MLP
ComParE	-	50.92	42.33	41.10	53.99	55.83
	AE	50.92	54.60	41.72	53.37	55.83
	PCA-2	30.06	37.68	33.74	49.08	54.60
	PCA-200	38.65	49.08	38.65	53.37	53.99
	LDA	49.69	50.31	52.15	52.76	51.53
Emobase	-	50.92	53.37	39.26	**57.06**	55.83
	AE	**57.67**	37.42	38.65	**57.67**	**57.06**
	PCA-2	55.83	55.21	40.49	49.08	55.21
	PCA-200	44.79	57.06	45.40	53.37	**57.67**
	LDA	42.94	44.17	41.72	41.10	44.79

TABLE II
Model performance comparisons for the atypical task. Top-2 performances are marked as bold.

Feature	Compression	LR	SVM	DT	XGB	MLP
ComParE	-	67.13	43.38	51.12	66.67	67.80
	AE	67.82	59.99	50.38	66.15	67.87
	PCA-2	33.85	22.62	49.93	66.20	67.56
	PCA-200	38.13	45.99	49.57	67.30	67.80
	LDA	42.33	45.48	45.39	38.70	39.94
Emobase	-	64.14	66.60	51.82	66.29	66.67
	AE	64.19	64.02	49.73	66.32	64.43
	PCA-2	**68.01**	36.19	49.07	67.49	**67.96**
	PCA-200	65.50	36.62	52.20	67.56	65.86
	LDA	55.02	55.02	48.78	50.00	54.95

classes given all the label information. Based on these pretrained parameters for the decision boundary, new instances are predicted to belong to the label of the highest probability. We implemented linear SVM in our experiments.

Decision tree (DT) : A decision tree comprises three types of nodes: the root node, internal nodes, and terminal nodes. The root node and the internal nodes contain features that determine the path of the training example. The construction of the tree structure starts with the root node, and the iterative dichotomiser 3 (ID3) algorithm selects the feature of each node. The algorithm chooses the attribute with the maximum information gain within each iteration.

Gradient Boosting tree (XGB): The boosting tree is essentially a weighted ensemble of weaker decision trees that optimizes a multiclass objective function [17]. We achieve this by recurrently adding a new decision tree function at every round. To make the model properly learn the structures of trees and the data, the loss function in each iteration is defined as the error between the model's prediction at each round and true value. The regularization term of each additive tree is added to alleviate overfitting on the training set and to promote a better generalization of the whole model.

Multi-layer perceptron (MLP): Multilayer perceptron is part of an artificial neural network, which comprises input nodes, hidden nodes, and output nodes. In our experiments, two hidden layers followed by a softmax layer with the Selu activation function [16] were used for nonlinear transformation. We also applied batch normalization and dropout with probability 0.2. All backpropagated parameters were updated to minimize the loss function at each epoch.

V. Performance evaluation

Heartbeat classification task: Table I demonstrates the results of the heartbeat classification task. Overall, the best performance 57.67% was obtained using MLP with PCA-200 com-pressed emobase features and LR/XGB with AE-compressed emobase features. In general, the MLP classifier outperformed the other four classification models in both the original/compressed feature setups. Additionally, DT is worse than XGB in most cases, as it is a simplified version of XGB. For the experiments with MLP, we stopped training when the validation loss started increasing to avoid overfitting.

Atypical affect classification task: As shown in Table II, using LR and MLP with PCA-2 emobase features achieved 68.01% and 67.96%, respectively. However, when considering all five training models, we observed that the AE-compressed features result in the highest average accuracy. Furthermore, the MLP classifier performed better than other models in most feature/compression settings. For the implementation, we divided the training corpus into an 8:2 ratio for the training/validation set.

VI. Discussion

A. Compression ability of AE

In two tasks, it was clearly observed that the combination of AE compression and the MLP classifier had very competitive performance in all tasks, even better than that of using the original emobase feature set. This shows the efficacy of the approach to training small amounts of data and high dimensions. To interpret these phenomena, we first compressed the ComParE features of the heartbeat training set into the 200-dimensional Euclidean space by PCA and selected 2 dominant principal components of each data for visualization, which were plotted in Fig 1. With these trained parameters of PCA, we compressed the ComParE features in a testing set into 2-dimensional space. For visualization of AE compression, we selected two components of the first and second largest absolute values among each 400-dimension train/test compressed vector because their activations are the most influential for the classification process.

As shown in the Fig 2, the AE-compressed data points belonging to the same classes in the train/test set are comparatively well clustered together, whereas data in different classes are separated. Furthermore, they are aligned linearly with an almost identical gradient, which makes the distribution of the test set features close to the training set's feature distribution. However, the distinction of PCA-compressed data points in different classes is apparently harder than the AE compression

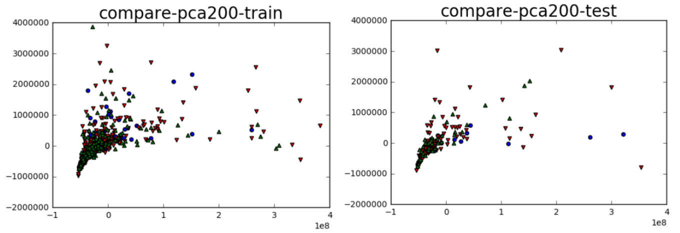

Fig. 1. Visualization of PCA compression.

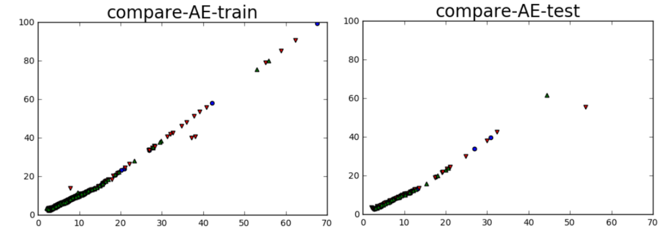

Fig. 2. Visualization of AE compression.

case. We consider all these factors to make AE better than the PCA compression method.

B. Classification with MLP

As described above, our experimental results reveal that MLP almost always outperforms other classifiers both in original and compressed feature settings, overcoming data insufficiency. This demonstrates that neural network architecture can still learn better representations with the compressed feature. In addition, we expect to see continual improvements with neural architecture variants in future works in general paralinguistic signal classification tasks.

VII. Conclusion

In this paper, we propose the implementation of compression frameworks for paralinguistic signal classification tasks. We extract two sets of features (ComParE, emobase2010) for training our models and explore how they vary in the aspect of classification accuracy among the heartbeat and atypical affect classification tasks. We train our models with our original features and the features autonomously compressed by PCA, LDA, and AE.

From the experiments, we observe that AE compression features and the MLP classifier are two key factors for achieving superior classification accuracy. Furthermore, they show even better performances than that of the combination with non-compressed features, which contain more information on the signal. These results demonstrate that the AE-compressed features can practically alternate original features that suffer from high dimensions when the size of the training corpus is limited.

Finally, we show by comparison that the MLP generally achieves a better ability to learn feature representations than classical models in two paralinguistic tasks.

Acknowledgments

K. Jung is with the ASRI, Seoul National University, Seoul, Korea. This work was supported by the Technology Innovation Program (10073144) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

[1] Joseph Redmon and Ali Farhadi, “Yolo9000: Better, faster, stronger,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6517–6525.

[2] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen, “Deep content-based music recommendation,” in *Advances in neural information processing systems*, 2013, pp. 2643–2651.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., “Deep speech 2: End-to-end speech recognition in english and mandarin,” in *International Conference on Machine Learning*, 2016, pp. 173–182.

[4] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., “Tacotron: Towards end-to-end speech synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.

[5] Björn W Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K Burgoon, Alice Baird, Aaron C Elkins, Yue Zhang, Eduardo Coutinho, and Keelan Evanini, “The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language.,” 2016.

[6] Saurabh Sahu, Rahul Gupta, Ganesh Sivaraman, Wael AbdAlmageed, and Carol Espy-Wilson, “Adversarial auto-encoders for speech based emotion recognition,” 2017.

[7] Björn Schuller, Gerhard Rigoll, and Manfred Lang, “Hidden markov model-based speech emotion recognition,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03). 2003 IEEE International Conference on*. IEEE, 2003, vol. 2, pp. II–1.

[8] Yixiong Pan, Peipei Shen, and Liping Shen, “Speech emotion recognition using support vector machine,” *International Journal of Smart Home*, vol. 6, no. 2, pp. 101–108, 2012.

[9] Kun Han, Dong Yu, and Ivan Tashev, “Speech emotion recognition using deep neural network and extreme learning machine,” in *Fifteenth annual conference of the international speech communication association*, 2014.

[10] Jinkyu Lee and Ivan Tashev, “High-level feature representation using recurrent neural network for speech emotion recognition,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] Zakaria Aldeneh and Emily Mower Provost, “Using regional saliency for speech emotion recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2741–2745.

[12] Simone Hantke, Hesam Sagha, Nicholas Cummins, and Björn Schuller, “Emotional speech of mentally and physically disabled individuals: Introducing the emotass database and first findings,” *Proc. Interspeech 2017*, pp. 3137–3141, 2017.

[13] Simone Hantke, Florian Eyben, Tobias Appel, and Björn Schuller, “ihearu-play: Introducing a game for crowdsourced data collection for affective computing,” in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE, 2015, pp. 891–897.

[14] Björn W Schuller, Stefan Steidl, Anton Batliner, Peter B Marschik, Harald Baumeister, Fengquan Dong, Simone Hantke, Florian Pokorny, Eva-Maria Rathner, Katrin D Bartl-Pokorny, et al., “The interspeech 2018 computational paralinguistics challenge: Atypical & self-assessed affect, crying & heart beats,” .

[15] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller, “Recent developments in opensmile, the munich open-source multimedia feature extractor,” in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 835–838.

[16] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter, “Self-normalizing neural networks,” in *Advances in Neural Information Processing Systems*, 2017, pp. 971–980.

[17] Tianqi Chen and Carlos Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.