LEARNING DECOUPLING FEATURES THROUGH ORTHOGONALITY REGULARIZATION

Li Wang[1], *Rongzhi Gu*[1], *Weiji Zhuang*[2], *Peng Gao*[2], *Yujun Wang*[2], *Yuexian Zou*[1,*]

[1] ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2] Xiaomi Inc., Beijing, China

ABSTRACT

Keyword spotting (KWS) and speaker verification (SV) are two important tasks in speech applications. Research shows that the state-of-art KWS and SV models are trained independently using different datasets since they expect to learn distinctive acoustic features. However, humans can distinguish language content and the speaker identity simultaneously. Motivated by this, we believe it is important to explore a method that can effectively extract common features while decoupling task-specific features. Bearing this in mind, a two-branch deep network (KWS branch and SV branch) with the same network structure is developed and a novel decoupling feature learning method is proposed to push up the performance of KWS and SV simultaneously where speaker-invariant keyword representations and keyword-invariant speaker representations are expected respectively. Experiments are conducted on Google Speech Commands Dataset (GSCD). The results demonstrate that the orthogonality regularization helps the network to achieve SOTA EER of 1.31% and 1.87% on KWS and SV, respectively.

Index Terms— Keyword spotting, orthogonality regularization, speaker verification

1. INTRODUCTION

With the development of speech technology, speech assistants are expected to help people solve affairs more efficiently. People increasingly enjoy the convenience of the hands-free experience. KWS and SV are two necessary key technologies of the human-machine interaction system. The machine can open the human-machine dialogue through keyword spotting (KWS) and get authorized dialogue targets based on speaker verification (SV).

KWS aims at detecting predefined keywords in an audio stream [1]. In recent years, end-to-end deep neural networks (DNN) have been employed in KWS and achieved superior performance [1]. Since then, more elaborately designed neural networks are employed to build better performing KWS systems, including convolutional neural networks [2, 3, 4, 5], recurrent neural networks [6, 7], and neural networks based on attention mechanisms [8, 9], etc. To improve the detection rate of non-target keywords while maintaining the accuracy of target keywords, [10, 11] explored the use of deep metric learning methods for the KWS task, where the DNN model is not used directly for classification but rather as a feature extractor that provides specific embeddings of keywords.

SV aims to verify the claimed identity of a person for a given speech [12]. In this paper, we only focus on text-independent SV that does not need any restriction on lexical content for speaker modeling as well as testing. DNNs are widely used for speaker verification because of their ability to extract speaker features effectively [13, 14]. In particular, for text-independent speaker verification tasks, the speaker features learned by DNNs are independent of the text content.

Traditionally, the state-of-art KWS and SV models are trained independently using different datasets since they expect to learn distinctive acoustic features. KWS requires features requiring linguistic content as much as possible, while SV requires features with rich speaker information. However, this independent treatment is not the way how humans process speech signals: humans always simultaneously decipher speech content and paralinguistic information including languages, speaker characteristics and emotions, etc [15]. Specifically, for KWS and SV tasks, this "multi-task processing" relies on three premises: (1) many common audio features and techniques have been designed and employed for the two tasks, such as Mel-frequency cepstral coefficient (MFCC) features [16, 4], TDNN [13, 5] modeling framework, and metric learning method [10, 17]; (2) all these tasks are basically discriminative tasks, so they can share the same front-end signal processing (e.g. filtering) modules [18, 19] and pipeline; and (3) they are mutual beneficial [15, 20, 21], for example, by paying particular attention to specific voices while understanding information about the content of the language, we can verify the speaker's voice; on the other hand, if we are familiar with a speaker, we tend to recognize his/her voice [20].

The third point above has been experimentally demonstrated by researchers that these KWS and SV tasks are collaborative [21, 20]. Jung *et al.*[21] argued that acoustic and speaker domains are complementary. They proposed a multi-task network and introduced global query attention to use the

*Corresponding author

This paper was partially supported by Shenzhen Science & Technology Fundamental Research Programs (No: JSGG20191129105421211 and GXWD20201231165807007-20200814115301001)

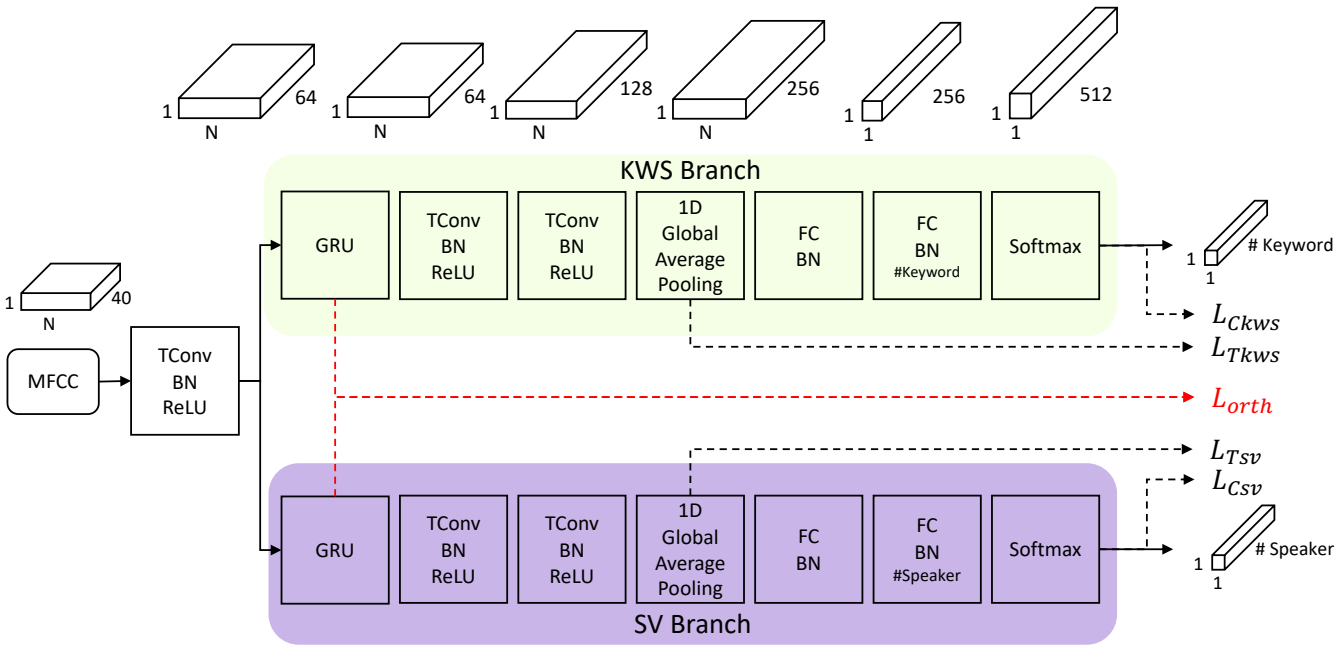

Fig. 1. Architecture of the proposed two-branch neural network. It consists of a shared temporal convolutional layer and two branches, the KWS branch and the SV branch. In order to adopt orthogonal regularization, the GRUs of KWS and SV branches were designed with the same neural network structure. N is the number of frames before and after layers. The details of L_{orth} is described in Section 2.2.

interrelated domain information of KWS and SV. However, the interaction of information between the KWS branch and the SV branch is unidirectional. Speaker features are not used by the KWS branch, which constrains further improvements in its performance. [20] designed a two-branch neural network, the input is a spectrogram, and the two branches output keyword and speaker results respectively. In order to avoid the interaction between keyword information and speaker information, [20] proposed dual attention to remove the information of another branch from the current branch.

We believe that the key to solve the KWS and SV tasks through a network is to efficiently extract the common features of both tasks and decouple the task-related features. In this paper, we propose a two-branch neural network to learn the task-specific features of KWS and SV along with the common feature. Orthogonality regularization [22] is employed to decouple the linguistic content information for KWS and the speaker information for SV. Thus, under the supervised learning paradigm, the KWS and SV branches can extract more distinct features from each other. Experiments are conducted on Google Speech Commands Dataset (GSCD). The results demonstrate that our proposed method achieve SOTA EER of 1.31% and 1.87% on KWS and SV, respectively.

2. PROPOSED METHOD

2.1. Neural Network Architecture

Inspired by SUDA [20], we designed a two-branch neural network for KWS and SV. As shown in Fig.1, the Mel-Frequency Cepstral Coefficient (MFCC) features are fed into the temporal convolution (TConv in Fig.1) to extract the shared features for KWS and SV. Then, the hidden representation from the first shared layer is passed to the next two gate recurrent unit (GRU) networks that focus on extracting valid information for each of the two sub-tasks, KWS and SV.

2.2. Decoupling Feature Learning via Orthogonality Regularization

In this paper, orthogonality regularization is used to decouple latent features between two sub-tasks, KWS and SV. It aims to decouple the hidden representation to learn speaker-invariant keyword representations and keyword-invariant speaker representations utilizing orthogonality regularization.

Specifically, we apply orthogonality regularization to the GRU of the KWS branch and the SV branch (figure 1 red dashed line). For each element x_t in the input sequence of time t, GRU layer computes the following function

$$\begin{aligned} r_t &= \sigma\left(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}\right) \\ z_t &= \sigma\left(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}\right) \\ n_t &= \tanh\left(W_{in}x_t + b_{in} + r_t * \left(W_{hn}h_{(t-1)} + b_{hn}\right)\right) \\ h_t &= (1 - z_t) * n_t + z_t * h_{(t-1)} \end{aligned} \tag{1}$$

where h_t is the hidden state at time t, and r_t, z_t, n_t are the reset, update, and new gates, respectively. σ is the sigmoid function and $*$ is the Hadamard product. W_{ir}, W_{iz}, W_{in}, W_{hr}, W_{hz}, W_{hn} are trainable weight matrices. The following

is an example of the computational procedure for computing orthogonal regularization using W_{ir}

$$L_{orth}^{ir} = \sum W_{ir}^{kws} W_{ir}^{svT} \quad (2)$$

where W_{ir}^{kws} and W_{ir}^{sv} represent the trainable matrices W_{ir} in KWS branch and SV branch, respectively. Summing up the orthogonal regularization for all the weight matrices in equation 1 to get the final orthogonal regularization

$$L_{orth} = L_{orth}^{ir} + L_{orth}^{iz} + L_{orth}^{in} + L_{orth}^{hr} + L_{orth}^{hz} + L_{orth}^{hn} \quad (3)$$

2.3. Loss Function

The loss function consists of three components, KWS task loss, SV task loss and orthogonality regularization

$$L = L_{kws} + L_{sv} + L_{orth} \quad (4)$$

where L_{kws} is the loss for KWS task, L_{sv} is the SV task loss, and L_{orth} is the orthogonal regularization loss, calculated from formula 3. Both L_{kws} and L_{sv} consist of cross-entropy (CE) loss and triplet loss, which provide supervised information for the training process of the model.

$$L_{kws} = L_{Ckws} + L_{Tkws} \quad (5)$$

$$L_{sv} = L_{Csv} + L_{Tsv} \quad (6)$$

where L_{Ckws} and L_{Csv} denote the CE losses, make the KWS branch extract linguistic content and makes the SV branch extract speaker information. L_{Tkws} and L_{Tsv} are triplet losses, it is employed to increase the inter-class distance and reduce intra-class distance. The triplet loss is applied on the 256-dimensional feature vector at the step after 1D global average pooling as shown in Figure 1, it is calculated in both the KWS and SV branches, the selection of the triplet samples are introduced in Section 3.2.

3. EXPERIMENTS

3.1. Dataset

The Google Speech Commands Dataset version 2 (GSCDv2[1]) [23] is used this study, which consists of 105,829 utterances of 35 words. All the utterances are spoken by 2,618 speakers. Every sample has 1 second duration and contains one word. We use only 2,277 speakers, excluding those with less than 11 utterances. Then disjoint sets of 1959, 159 and 159 speakers are randomly selected for training, validation, and test set, as Table 2 shows. Also, to check the robustness of KWS against unseen words, utterances corresponding to the three words 'happy', 'marvin', and 'sheila' are excluded from the training set. In this way, we get 83,636 training samples, 7,644 validation samples and 7,857 test samples from GSCDv2.

[1] http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz

Table 1. Information on the real-world datasets

	Training	Validation	Test	Total
Utterances	60,647	6,048	6,048	72,743
Speakers	1,213	121	121	1,455

In order to verify the validity of the model in the real world, we additionally use the real-world dataset. The keyword is a four-syllable Mandarin Chinese term ("xiao-ai-tong-xue"). We collected 14,543 positive examples and 58,200 negative examples. The splits of the training set, validation set and test set are shown in Table 1.

Table 2. The number of utterances and speakers in training, validation and test set. Extracted from Google Speech Commands Dataset Version 2.

	Training	Validation	Test	Total
Utterances	83,636	7,644	7,857	99,137
Speakers	1,959	159	159	2,277

3.2. Implementation Details

Our implementation was done with Pytorch deep learning toolkit. The raw audio is decomposed into a sequence of frames where the window length is 20 ms and the stride is 10 ms for feature extraction. We use 40-dimensional MFCC.

Training. All the models are trained with a mini-batch of 256 samples using stochastic gradient descent with weight decay of 0.001 and momentum of 0.9. The initial learning rate is set to be 0.01 and decayed by a factor of 2 when the validation KWS or SV equal error rate (EER) does not decrease for 3 epochs. The training is terminated when validation KWS or SV EER does not decrease for 10 epochs. Compared to an enrollment sample, a test sample falls into one of the following four scenarios: (1) same keyword, same speaker; (2) same keyword, different speaker (3) different keyword and same speaker; (4) different keyword, different speaker. To training efficiency, for each training sample, we have randomly choose 4 samples from the training set, corresponding to the 4 scenarios mentioned above.

Evaluation. KWS and SV are basically discrimination tasks that make a decision given a score between embeddings of enrollment and test utterances [21]. EER is used as the metric to evaluate the models in this paper. All utterances in the test set are used once for enrollment. The cosine distance is used to measure the score. We train each model 10 times and report its average performance.

3.3. Baselines

We use the same structure as the KWS and SV branches to train the KWS and SV tasks separately as our baselines for the

KWS and SV tasks to demonstrate the collaborative between the two tasks. To compare the performance impact of orthogonality regularization and dual attention [20], we replicated SUDA [20] and trained it on the GSCDv2 dataset following the training method described in Section 3.2.

3.4. Experimental Results and Analysis

3.4.1. Competing and collaborative information properties of KWS and SV

As shown in the last row of Table 3, the orthogonality regularization leads to a significant performance improvement. In addition, formula 3 can be applied to the LSTM with a simple modification, and we also use orthogonality regularization for the LSTM of the KWS branch and SV branch in SUDA. As show in the "SUDA" row of Table 3, orthogonality regularization brings a consistent performance improvement for proposed model and SUDA. Our experimental results are consistent with the statement of [15] that there is collaborative and competitive information between KWS and SV. Good utilization of the information between the two tasks can lead to improved performance of both tasks. Conversely, without orthogonality regularization, the information competition between them makes both tasks perform worse than if they were trained separately, as shown in Table 3.

Table 3. Performance in EER (%) for the proposed two-branch neural network and baselines on GSCDv2, with 95% confidence intervals..

Model	KWS	SV
KWS single-task	1.86±0.10	-
SV single-task	-	2.27±0.03
SUDA	4.87±0.07	5.95±0.08
w/ L_{orth}	2.48±0.11	3.51±0.06
Proposed	**1.31**±0.03	**1.87**±0.07
w/o L_{orth}	2.08±0.05	2.27 ±0.09

3.4.2. Impact of training data

As mentioned in Section 3.2, one test sample falls into one of four scenarios. In this section, we only keep scenarios (1) same keyword, same speaker and scenarios (4) different keyword, different speaker. The experimental results are shown in Table 4, where there is a performance degradation using the training data from two scenarios compared to the training data from four scenarios. This demonstrates that adding training data from scenarios (2) and (3) allows the model to extract more discriminative features.

3.4.3. Performance in the real world

We test on real-world datasets, as shown in Table 5, where orthogonality regularization leads to performance improvements. A noteworthy point is that the difference between the EER of KWS and the EER of SV is about 10 times, which is an issue worth exploring and we will explore this issue in our future work.

Table 4. Performance in EER (%) for different training samples selection, with 95% confidence intervals.

Training Data	KWS	SV
2 scenarios	1.71±0.05	2.13±0.12
4 scenarios	**1.31**±0.03	**1.87**±0.07

Table 5. Performance in EER (%) for the proposed two-branch neural network and baselines, with 95% confidence intervals.

Model	KWS	SV
Proposed	**0.61**±0.04	**7.01**±0.14
w/o L_{orth}	0.80±0.07	7.43±0.16

4. CONCLUSION

The performance of keyword spotting (KWS) and speaker verification (SV) tasks can be boosted by leveraging from each other. In this paper, we explore a method to extract the common feature while decoupling task-specific features. Specifically, we design a two-branch neural network for KWS and SV, and orthogonality regularization is applied to decouple latent features between KWS and SV. Our proposed method reaches SOTA. In future work, we will explore the usability of orthogonality regularization in other tasks, such as speaker verification and emotion classification; intent detection and text sentimental classification [24].

5. REFERENCES

[1] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[2] Tara N Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[3] Raphael Tang and Jimmy Lin, "Deep residual learning for small-footprint keyword spotting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5484–5488.

[4] Seungwoo Choi, Seokjun Seo, Beomjun Shin, Hyeongmin Byun, Martin Kersner, Beomsu Kim, Dongyoung Kim, and Sungjoo Ha, "Temporal convolution for

real-time keyword spotting on mobile devices," *arXiv preprint arXiv:1904.03814*, 2019.

[5] Taejun Kim and Juhan Nam, "Temporal feedback convolutional recurrent neural networks for keyword spotting," *arXiv preprint arXiv:1911.01803*, 2019.

[6] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *International Conference on Artificial Neural Networks*. Springer, 2007, pp. 220–229.

[7] Martin Woellmer, Bjoern Schuller, and Gerhard Rigoll, "Keyword spotting exploiting long short-term memory," *Speech communication*, vol. 55, no. 2, pp. 252–265, 2013.

[8] Ye Bai, Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Zhengkun Tian, Chenghao Zhao, and Cunhang Fan, "A time delay neural network with shared weight self-attention for small-footprint keyword spotting," *Proc. Interspeech 2019*, pp. 2190–2194, 2019.

[9] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, "Attention-based end-to-end models for small-footprint keyword spotting," *arXiv preprint arXiv:1803.10916*, 2018.

[10] Jaesung Huh, Minjae Lee, Heesoo Heo, Seongkyu Mun, and Joon Son Chung, "Metric learning for keyword spotting," *arXiv preprint arXiv:2005.08776*, 2020.

[11] Niccolo Sacchi, Alexandre Nanchen, Martin Jaggi, and Milos Cernak, "Open-vocabulary keyword spotting with audio and text embeddings," in *INTERSPEECH 2019-IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019, number CONF.

[12] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 4, pp. 1–22, 2004.

[13] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[14] Jee-weon Jung, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," *arXiv preprint arXiv:1904.08104*, 2019.

[15] Zhiyuan Tang, Lantian Li, Dong Wang, and Ravichander Vipperla, "Collaborative joint training with multitask recurrent model for speech and speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 493–504, 2016.

[16] Shi-Huang Chen and Yu-Ren Luo, "Speaker verification using mfcc and support vector machine," in *Proceedings of the International multiconference of engineers and computer scientists*. Citeseer, 2009, vol. 1, pp. 18–20.

[17] Chunlei Zhang, Kazuhito Koishida, and John HL Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1633–1644, 2018.

[18] Amin Saremi, Rainer Beutelmann, Mathias Dietz, Go Ashida, Jutta Kretzberg, and Sarah Verhulst, "A comparative study of seven human cochlear filter models," *The Journal of the Acoustical Society of America*, vol. 140, no. 3, pp. 1618–1634, 2016.

[19] Rajath Kumar, Vaishnavi Yeruva, and Sriram Ganapathy, "On convolutional lstm modeling for joint wake-word detection and text dependent speaker verification.," in *Interspeech*, 2018, pp. 1121–1125.

[20] Tianchi Liu, Rohan Kumar Das, Maulik Madhavi, Shengmei Shen, and Haizhou Li, "Speaker-utterance dual attention for speaker and utterance verification," *arXiv preprint arXiv:2008.08901*, 2020.

[21] Myunghun Jung, Youngmoon Jung, Jahyun Goo, and Hoirin Kim, "Multi-task network for noise-robust keyword spotting and speaker verification using ctc-based soft vad and global query attention," *arXiv preprint arXiv:2005.03867*, 2020.

[22] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang, "Can we gain more from orthogonality regularizations in training deep cnns?," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 4266–4276.

[23] Pete Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[24] Zhiqi Huang, Fenglin Liu, Peilin Zhou, and Yuexian Zou, "Sentiment injected iteratively co-interactive network for spoken language understanding," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7488–7492.