FINE-TUNING WAV2VEC2 FOR SPEAKER RECOGNITION

Nik Vaessen, David A. van Leeuwen

Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
{nvaessen,dvanleeuwen}@science.ru.nl

ABSTRACT

This paper explores applying the wav2vec2 framework to speaker recognition instead of speech recognition. We study the effectiveness of the pre-trained weights on the speaker recognition task, and how to pool the wav2vec2 output sequence into a fixed-length speaker embedding. To adapt the framework to speaker recognition, we propose a single-utterance classification variant with cross-entropy or additive angular softmax loss, and an utterance-pair classification variant with BCE loss. Our best performing variant achieves a 1.88% EER on the extended voxceleb1 test set compared to 1.69% EER with an ECAPA-TDNN baseline. Code is available at `github.com/nikvaessen/w2v2-speaker`.

Index Terms— speaker recognition, wav2vec2, transfer learning

1. INTRODUCTION

In the field of natural language processing (NLP) it has become standard to fine-tune self-supervised pre-trained models, such as BERT [1], XLNet [2], and T5 [3], on a wide variety of NLP tasks. Recently, this framework of pre-training and fine-tuning has also been successfully used in automatic speech recognition with wav2vec2 [4]. The aim of this study is to explore the feasibility of fine-tuning the wav2vec2 pre-trained network on a different task than speech recognition, namely *speaker* recognition.

The BERT and wav2vec2 network have commonalities in their design. Both have stacks of transformer layers and they use self-supervised, contrastive pre-training with masked input. However, they differ in three major aspects: 1) the input tokens to the encoder in wav2vec2 are raw audio processed by a CNN instead of WordPiece embeddings, 2) wav2vec2 uses relative positional embeddings computed by a CNN instead of sinusoidal positional embeddings, 3) there is no class token and equivalent next-sentence prediction task in the pre-training procedure of wav2vec2.

The class token in BERT is used when fine-tuning on sentence-pair classification tasks, e.g., entailment, and for single-sentence classification tasks, e.g., sentiment analysis. It is not used for single sentence tagging tasks, e.g., named entity recognition. In this respect, speech recognition is analogous to sentence tagging where each output token can represent a phone or letter. Hence, speech recognition does not require a class token. In contrast, speaker recognition corresponds to either sentence-pair classification, or single-sentence classification, and might therefore benefit from a class token which summarizes the whole input sequence.

Our work focuses on the following questions. First and foremost, we want to find out whether the pre-trained wav2vec2 weights are an effective initialization for speaker recognition. We further want to explore if wav2vec2 can be adapted to speaker recognition without a class token and next-sentence prediction task. One solution is to pool the variable-length sequence of wav2vec2 embeddings into a fixed-size speaker embedding. This raises the questions if pooling is an effective replacement for a class token, and if so, which pooling method is most suitable.

2. RELATED WORK

Wav2vec2 has already been applied to a variety of speech-related tasks. In [5] the network is used for speaker recognition and language identification in both a single and multi-task learning setting. The authors of [6] show good performance for language identification with 25 languages but modify wav2vec2 to use log-mel spectogram input instead of raw waveforms. In [7] the (frozen) wav2vec2 embeddings are input to a learnable downstream model for carrying out emotion recognition. Meanwhile [8] manages to fine-tune the wav2vec2 model itself on emotion recognition with CTC loss by using emotion-labeled phonetic units. The LeBenchmark [9] uses the wav2vec2 models as a baseline and encapsulates speech recognition, spoken language understanding, emotion recognition and speech translation in a single benchmark.

3. METHODOLOGY

3.1. The wav2vec2 architecture

The wav2vec2 framework [4] applies the concept of self-supervised pre-training with transformers to automatic speech recognition. In Fig. 1 we show a general overview of the network architecture during fine-tuning. The next subsections summarize each component.

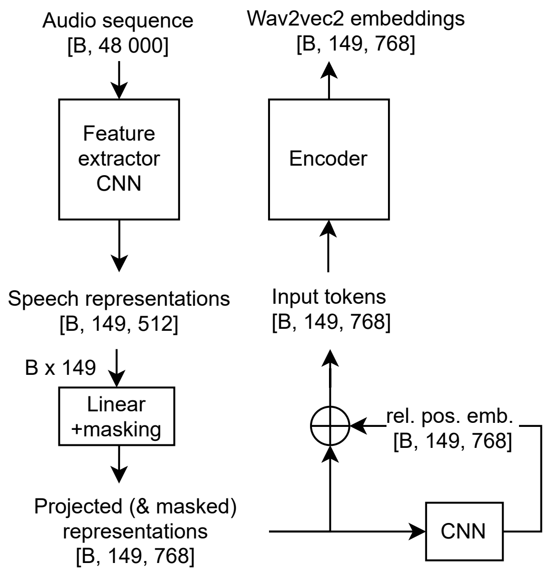

Fig. 1. Overview of the Wav2vec2 architecture. Shapes are specified for a batch of B audio samples with a length of 3 seconds.

3.1.1. Feature extraction

The first step is to encode a raw audio waveform (normalized to zero mean and unit variance) into learned representations with a discrete time unit. The *feature extractor* consists of 7 consecutive 1-dimensional convolutions with 512 channels and respective kernel sizes of (10, 3, 3, 3, 3, 2, 2) and stride (5, 2, 2, 2, 2, 2, 2). The output of the first convolutional layer is group normalized [10] such that each of the 512 channel sequences has zero mean and unit variance before GELU activation [11] is applied. The other convolutional layers do not have any normalization layers and their output is directly activated with GELU. The output of the feature extractor is an encoded vector sequence with dimensionality 512. Each vector has a receptive field of 20 ms which is similar to the window sizes in spectral-based representations.

3.1.2. Projection, SpecAugment & positional embedding

After the feature extraction each encoded vector representation in the sequence is independently normalized to zero mean and unit variance and projected into 768 dimensions by a single, shared fully-connected layer called the *feature projector*. On all projections dropout is applied (but no activation). Then, masking is applied over the whole sequence analogous to SpecAugment [12]; 0 or more random sets of consecutive vectors (masking in time domain) as well as 0 or more random sets of consecutive channels (masking in "frequency" domain) have their values blanked to 0. This masked projected sequence is then convolved by a single layer with a kernel size of 128, a stride of 1, padding of 64 and 16 groups followed by GELU activation in order to create a *relative positional embedding* for each projected representation. This relative positional embedding is summed with the original input of the convolution, which changes the receptive field from 20 ms to 2.5 s. As a final step each vector is independently normalized with LayerNorm and dropout is applied again.

3.1.3. Transformer

The masked and projected sequence with both local and positional information is fed through an *encoder* with 12 consecutive transformer layers. Each transformer layer consists of a residual 12-headed self-attention module and a residual 2-layer feed forward network with respectively 3072 and 768 units. LayerDrop [13, 14] is applied such that each transformer layer is potentially skipped. The final output sequence, with each representation potentially having both local and global information due to self-attention, is used in a downstream task.

3.2. Three wav2vec2 variants for speaker recognition

The original wav2vec2 framework fine-tunes on speech recognition by independently labeling each wav2vec2 output embedding with a shared fully-connected layer, and optimizes with CTC loss [15]. We propose two adaptions to this design for the speaker recognition task, inspired by BERTs [1] single-sentence and sentence-pair classification setup.

3.2.1. Speaker recognition as single-utterance classification

The current paradigm in speaker recognition with deep neural networks is to train models with a classification-based approach [16, 17]. To mimic this architectural paradigm two modifications to wav2vec2 are made. First, the sequence of wav2vec2 embeddings is reduced to a single embedding during training (see subsection 3.3). Secondly, we add a fully connected layer which uses the pooled embedding to classify each speaker in the training data with cross-entropy (CE) or angular additive softmax (AAM) [18, 19] loss. A trial is evaluated with the cosine similarity between two pooled embeddings. We refer to these variants as *w2v2-ce* and *w2v2-aam*.

3.2.2. Speaker recognition as utterance-pair classification

The second approach directly computes a similarity score. The two audio segments of a speaker recognition trial are first processed independently up to the encoder part of the network. Then, the two sequences of input tokens are concatenated, accompanied by special tokens at the embedding level: A start (all $+1$), separator (all -1), and end (all -1) token. The first wav2vec2 embedding in the output sequence (corresponding to the start token) is used as input to a logistic regression with one dense layer. The singular output is the logit for the BCE loss and the score for evaluating a trial.

During training a batch consists of 8 speakers, 4 utterances per speaker, and 16 same and different speaker-pairs. We refer to this third variant as *w2v2-bce*.

3.3. Pooling methods

We propose several pooling methods to reduce the variable-length sequence of wav2vec2 embeddings to a fixed-size speaker embedding. We first consider the standard statistical pooling methods *mean*, *max*, *mean&std* and *quantile*. They aggregate each dimension over the time axis. The *mean&std* variant doubles the embedding dimensionality while *quantile* pooling expands each dimension five-fold with quantiles (0, 0.25, 0.5, 0.75, 1). We also assess taking the *first*, *middle* or *last* embedding of the sequence as a "pooling" strategy as well as *random*ly selecting the index. Lastly we consider inserting a "start" token (all values $+1$) before the input sequence of the encoder and then selecting the first output token as the speaker embedding. This is termed *first&cls*. Unlike BERT our "start" token token does not have a meaningful prior.

4. EXPERIMENTS

4.1. Data

All experiments are conducted on the VoxCeleb datasets [20, 21], which consist of interviews of celebrities extracted from YouTube. The VoxCeleb2 dev set, which contains ~1.1 M audio files over 6 k speakers, is used for training. A validation set is created based on ~2 % of the dev set data which includes all speakers but does not overlap in recordings. For this validation set a random trial set of 5 k same and 5 k different pairs is generated, from which we compute the validation EER. This is used to select the best checkpoint during a training run as well as to tune hyperparameters. For evaluation we use the cleaned *original* (vox1-o, 40 speakers, ~37 k trials), *extended* (vox1-e, 1251 speakers, ~580 k trials) and *hard* (vox1-h, 1190 speakers, ~550 k trials, equal nationality and sex) test sets from voxceleb 1[21]. There is no speaker overlap between the voxceleb1 test sets and the voxceleb2 dev set. The experiments with the wav2vec2 network use the pre-trained weights[1] on Librispeech [22] released on HuggingFace [23] by Fairseq [24].

4.2. Computational budget and fair comparison

We compare the performances of models under similar computational budgets. Each network is trained with a batch size of 3.2M audio samples [4] by randomly sampling 3 seconds from 66 different audio files. No data augmentation techniques are used. We train for 100k iterations, approximately 6 epochs, with Adam [25] and a OneCycle learning rate schedule [26]. The maximum learning rate (LR) of the cycle is found by 1) performing an LR range test [27] with 5k iterations and 2) tuning on a 7-sized grid centered around the LR with the steepest slope and bounded by the LRs where the loss respectively started and stopped decreasing. Models are evaluated with a cosine score between speaker embeddings lacking any further post-processing, except for the wav2vec2-bce variant which computes scores directly.

[1]See `https://huggingface.co/facebook/wav2vec2-base`

Table 1. EER performance of standard filterbank-based speaker recognition networks as well as the fine-tuned wav2vec2 variations. We ran $N = 4$ runs to compute the standard deviations. Bold indicates best performance.

	EER performance (mean, std in %)					
NETWORK	vox1-o		vox1-e		vox1-h	
x-vector [16]	5.22	0.12	5.60	0.05	8.75	0.05
ECAPA-TDNN [17]	**1.61**	0.03	**1.69**	0.03	**3.10**	0.05
w2v2-ce	2.25	0.20	2.58	0.10	4.91	0.13
w2v2-aam	1.91	0.12	2.22	0.04	4.33	0.08
w2v2-bce	7.28	0.22	7.19	0.22	11.34	0.83

4.3. Baseline systems

We train two popular baselines models for speaker recognition, x-vector [16, 28] and ECAPA-TDNN [17, 28], and compare them to the three wav2vec2 adaptions w2v2-ce, w2v2-aam and w2v2-bce. All five models have the computation budget described in subsection 4.2. The X-vector and ECAPA-TDNN networks use 40-dimensional filterbanks as input. The w2v2-ce and w2v2-aam variants use mean&std pooling which was chosen as it is also used in the x-vector network architecture. The w2v2-aam variant and ECAPA-TDNN use the AAM softmax loss with a scale of 30 and a margin of 0.2 in this and further experiments. The feature encoder part of the wav2vec2 architecture is frozen for the whole training procedure which corresponds to [4].

4.4. Variation in pooling

Next we explore the different poolings methods proposed in Section 3.3. This was carried out for the single-utterance classification variants: w2v2-ce and w2v2-aam. We use the same training settings as in the baseline comparison and only vary the pooling method. Note therefore that the learning rate was tuned to the "mean&std" setup.

4.5. Ablation study

We perform several ablations on the best-performing wav2vec2 variant for speaker recognition. These ablations are trained with 100k iterations except in the experiments for the batch size. The first set of ablations study the effect of not freezing the feature extractor in the fine-tuning procedure as well as randomly initialising the whole network and thus not using any pre-trained weights. The second set of ablations explore the relative importance of the regularisation techniques in the network architecture. We first disable only LayerDrop,

Table 2. The performance of different pooling strategies for the single utterance classification architectures on the extended voxceleb1 test set. Each method was trained with $N = 3$ random seeds. The evaluation of random pooling was repeated four times for each run.

	EER on vox1-e (mean, std in %)			
pooling	w2v2-ce		w2v2-aam	
max	4.79	0.55	2.27	0.04
quantile	2.75	0.17	2.21	0.03
mean	2.69	0.06	2.11	0.05
mean&std	2.60	0.08	2.18	0.03
first	2.61	0.10	2.15	0.05
first&cls	**2.52**	0.11	**2.06**	0.03
middle	2.55	0.07	2.18	0.03
last	2.58	0.07	2.18	0.03
random ($N = 1$)	2.56	0.002	2.40	0.002
random ($N = 3$)	2.70	0.13	2.37	0.03

then sequentially disable dropout and the masking of certain frames as well. The third set simply studies the effect of increasing (with 50k iterations) or decreasing (with 200k iterations) the chosen batch size by a factor of 2. The last ablations involve the learning rate schedule. We first test two constant learning rates: 3×10^{-6} is the LR where the loss started increasing in the LR range test and 10^{-5} is the LR with the steepest decrease. The second schedule exponentially decays the LR from 10^{-5} to 3×10^{-6}. The final schedule is the tri-stage learning rate schedule similar to [4] which includes a warm-up phase linearly increasing the LR from 10^{-7} to 10^{-5} in 10k iterations, a constant phase of 40k iterations, and an exponentially decreasing stage from 10^{-5} to 10^{-7} for the remaining iterations.

5. RESULTS

5.1. Baseline comparison

The LR range test and grid-tuning approach found the following learning rates: 10^{-4} for x-vector, 10^{-3} for ECAPA-TDNN, 9×10^{-5} for w2v2-ce, 5×10^{-5} for w2v2-aam, and 3×10^{-5} for w2v2-bce. Table 1 shows four runs with these learning rates. We see that ECAPA-TDNN performs best on all test sets. The w2v2-aam network is the best performing wav2vec2 variant. Both w2v2-ce and w2v2-aam manage to improve on the x-vector architecture. Modeling speaker recognition as utterance-pair classification (w2v2-bce) performed worst.

5.2. Pooling methods

Table 2 compares the 9 different pooling strategies with the w2v2-ce and w2v2-aam networks. We observe that first&cls pooling performs best for both networks. The difference between first&cls pooling and the other pooling methods is more pronounced for w2v2-aam than for w2v2-ce. The low inter-model variance of random pooling shows that each wav2vec2 embedding is a stand-alone speaker embedding. Not shown, using random pooling in the evaluation for networks which were trained with first&cls pooling degrades the EER with 0.2% points. Moreover, creating ensembles out of different embeddings in the output sequence did not improve performance. This suggests that the transformer layer processes each embedding in the sequence similarly.

Table 3. EER performance under varying ablated configurations for the w2v2-aam network variant. Each configuration was run with $N = 3$ random seeds. One run with exponential decay diverged and we therefore show $N = 2$ results for that row.

	EER on vox-e	
ablation	mean(%)	std (%)
default w2v2-aam first&cls setup	2.06	0.03
unfrozen feature extractor	1.88	0.08
& no pre-trained weights	5.08	0.08
no layerdrop ($p = 0$)	2.45	0.05
& no dropout ($p = 0$)	2.39	0.04
&& no time masking ($p = 0$)	2.39	0.09
batch size 32	2.12	0.10
batch size 128	2.05	0.01
constant (1e-5)	50.0	0.00
constant (3e-6)	3.71	0.08
exponential decay ($N = 2$)	2.42	0.07
tri-stage	1.97	0.04

5.3. Ablation study

Table 3 shows the results of the ablation study. We see that freezing the feature extractor leads to worse performance but is more stable across runs with different seeds. We also note a large degradation in performance when initializing with random weights instead of the pre-trained weights. The regularisation settings for fine-tuning on speech recognition are also beneficial for fine-tuning on speaker recognition. Increasing the batch size beyond 3.2M audio samples does not increase performance, although it does decrease the variance slightly. Using a learning rate schedule with a warm-up phase, such as the tri-stage or OneCycle schedule, is critical for stable and good performance.

6. CONCLUSION AND FUTURE WORK

We have shown that the wav2vec2 framework can be successfully adapted to the speaker recognition task and that the pre-trained weights used for fine-tuning on speech recognition are also useful for fine-tuning on speaker recognition. Good results with first&cls pooling indicate that including a class token in the pre-training procedure is promising future work. Modeling speaker recognition as a paired-utterance classification problem did not perform well. Future work in that direction might involve limiting the attention mechanism to the opposite utterance to simplify the learning task.

7. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. June 2019, pp. 4171–4186, Association for Computational Linguistics.

[2] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, "XLNet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[4] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 12449–12460.

[5] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu, "Exploring wav2vec 2.0 on Speaker Verification and Language Identification," in *Proc. Interspeech 2021*, 2021, pp. 1509–1513.

[6] Andros Tjandra, Diptanu Gon Choudhury, Frank Zhang, Kritika Singh, Alexei Baevski, Assaf Sela, Yatharth Saraf, and Michael Auli, "Improved language identification through cross-lingual self-supervised learning," *arXiv preprint arXiv:2107.04082*, 2021.

[7] Leonardo Pepino, Pablo Riera, and Luciana Ferrer, "Emotion Recognition from Speech Using wav2vec 2.0 Embeddings," in *Proc. Interspeech 2021*, 2021, pp. 3400–3404.

[8] Jiahong Yuan, Xingyu Cai, Renjie Zheng, Liang Huang, and Kenneth Church, "The role of phonetic units in speech emotion recognition," *arXiv preprint arXiv:2108.01132*, 2021.

[9] Solène Evain et al., " LeBenchmark: A Reproducible Framework for Assessing Self-Supervised Representation Learning from Speech," in *Proc. Interspeech 2021*, 2021, pp. 1439–1443.

[10] Yuxin Wu and Kaiming He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[11] Dan Hendrycks and Kevin Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[12] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[13] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*. Springer, 2016, pp. 646–661.

[14] Angela Fan, Edouard Grave, and Armand Joulin, "Reducing transformer depth on demand with structured dropout," *arXiv preprint arXiv:1909.11556*, 2019.

[15] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[16] 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.

[17] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[18] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Yi Liu, Liang He, and Jia Liu, "Large Margin Softmax Loss for Speaker Verification," in *Proc. Interspeech 2019*, 2019, pp. 2873–2877.

[20] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[21] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[22] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[23] Thomas Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, Oct. 2020, pp. 38–45, Association for Computational Linguistics.

[24] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[25] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] Leslie N Smith and Nicholay Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. International Society for Optics and Photonics, 2019, vol. 11006, p. 1100612.

[27] Leslie N Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.

[28] Mirco Ravanelli et al., "Speechbrain: A general-purpose speech toolkit," 2021.