Speech and Language Recognition with Low-rank Adaptation of Pretrained Models

Amrutha Prasad[1,2], *Srikanth Madikeri*[1], *Driss Khalil*[1], *Petr Motlicek*[1,2], *Christof Schuepbach*[3]

[1]Idiap Research Institute, Martigny, Switzerland [2]Brno University of Technology, Brno, Czech Republic
[3]Armasuisse Science and Technology, Thun, Switzerland

{amrutha.prasad, mrsrikanth, driss.khalil, petr.motlicek}@idiap.ch,
christof.schuepbach@armasuisse.ch

Abstract

Finetuning large pretrained models demands considerable computational resources, posing practical constraints. Majority of the total number of parameters in these models are used by fully connected layers. In this work, we consider applying a semi-orthogonal constraint, followed by full finetuning to the fully connected layers reduces model parameters significantly without sacrificing efficacy in downstream tasks. Specifically, we consider wav2vec2.0 XLS-R and Whisper models for Automatic Speech Recognition and Language Recognition. Our results show that we can reduce the model size by approximately 24% during both training and inference time with 0.7% absolute drop in performance for XLS-R and no drop in performance for Whisper for ASR. In combination with performance-efficient training with low-rank adapters, the resource requirements for training can be further reduced by up to 90%.

Index Terms: parameter reduction, language identification, speech recognition, wav2vec2.0

1. Introduction

Finetuning large pretrained models to achieve state-of-the-art performance across various downstream tasks has become a standard practice in machine learning. However, this approach presents challenges, particularly in resource-constrained environments where the computational demand of finetuning these expansive models poses a significant obstacle. Recognizing this limitation, researchers have directed their efforts towards mitigating the parameter budget of these models. One solution during training time is to keep the weight matrices frozen and learn the updates through adapters, an additional set of parameters for transfer learning. This does not, however, reduce the computational requirements during inference. In this paper, we study a simple parameter reduction approach when using pretrained models for developing Automatic Speech Recognition (ASR) and Language Identification (LID) systems.

Evidence from recent literature suggests that constraining finetuning of large pretrained models to a low-rank provides competitive, and sometimes better, performance on downstream tasks [1, 2, 3]. This suggests that the models may be overparameterized and parameter reduction techniques may benefit from exploiting such redundancy. In [4], the authors demonstrate that adapting only the Fully Connected (FC) layers in the MLP component of the Transformer [5] alone may be sufficient for many downstream tasks. Prior to Deep Neural Network (DNN) based systems, reduction of parameters during acoustic modeling has been studied in [6, 7, 8] for Gaussaian Mixture Models (GMM). In this paper, we study the extent of this overparameterization by applying Singular Value Decomposition (SVD) to the linear layer weights of the MLP and choosing parameters corresponding to only the highest eigenvalues, thereby reducing the model size by at least 28%. This technique effectively reduces the parameter budget during both finetuning and evaluation stages. Application of SVD on the weights in deep neural network architectures for speech processing tasks have been explored extensively in the past [9, 10, 11, 12]. In [13], the structure of SVD is applied on the output layer to reduce the parameter by up to 30%. In [14], a semi-orthogonal structure is imposed within the FC layers right from the beginning of model training to avoid training-time stability issues. The effect of semi-orthogonal constraint is studied in [15] for various back-end classifiers during finetuning for LID.

Further, we also study combining low-rank factorization with parameter-efficient finetuning. Specifically, we apply Low-Rank Adapters (LoRA) [2] during finetuning after the FC layers of the MLP have been reduced with SVD. Thus, we not only exploit the parameter reduction from low-rank factorization, but will also be able to utilize the benefits of Parameter Efficient Training (PEFT). Two popular pretrained models are considered: specifically the XLS-R [16] and Whisper [17] medium models; one trained in self-supervised fashion while the other trained in a weakly-supervised way. We consider two tasks supported by both models: ASR and LID.

The rest of the paper is organized as follows: Section 2 explains the low-rank adaptation approach employed in this work. The datasets used in training and evaluation of ASR and LID tasks are described in Section 3. The experimental setup and the results are presented in Section 4 and the findings of our work are discussed in Section 5.

2. Low-rank Factorization

The Transformer [5] architecture provided a significant advancement in the field of natural language processing (NLP) and is the foundation for numerous state-of-the-art models now in speech processing. Each transformer layer consist of multi-head self-attention (MHA) modules and feed-forward neural networks (we refer to this as MLP from hereon), each followed by layer normalization and residual connections to provide stability during training. In a commonly used configurations, both the MHA and MLP components contribute almost equally to the parameter count of models.

We consider the following two popular pretrained transformer-based models in this work:

XLS-R [16]: Based on the wav2vec 2.0 architecture [18], XLS-R is a large-scale multilingual model trained in self-supervised fashion. It uses unlabeled speech from 128 languages including data from VoxPopuli, MLS, CommonVoice,

This work was supported by Armasuisse Science and Technology.

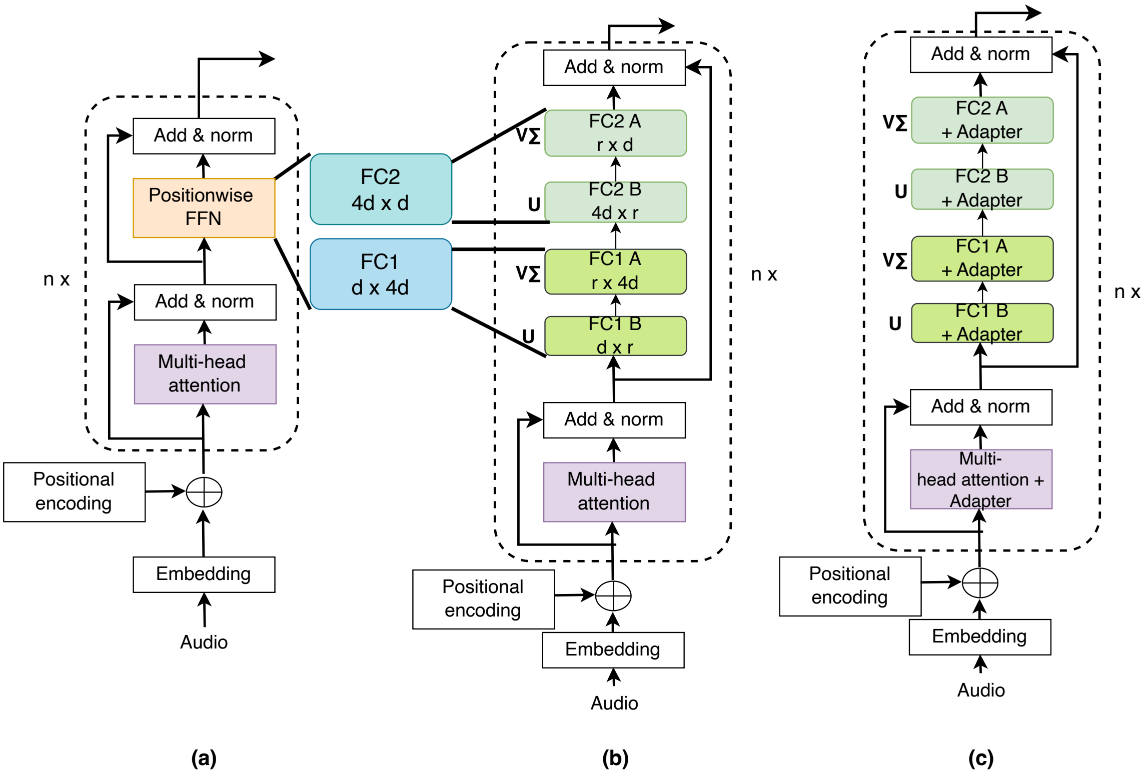

Figure 1: *Overview of a single transformer layer for various settings: (a) shows the overview of the standard transformer layer used during full finetuning towards downstream tasks. (b) shows the transformer layer after applying matrix factorization. This is applied for inference only of the finetuned model and during matrix factorized finetuning. (c) shows the transformer layer during low-rank adaptation. n is number of transformer layers which is 24 for XLS-R model and 48 for Whisper model. A rank r is the dimension used to apply SVD. In this work $r = 512, 256, 128$.*

BABEL, and VoxLingua107 [19] amounting to a total of 436k hours. There are 24 transformer layers resulting in 300M parameter model, of which 67% of the total parameter budget is used by the FC layers in the MLP components.

Whisper [17]: A transformer-based encoder-decoder general-purpose model for various speech processing tasks, including multilingual speech recognition, speech translation, spoken language identification, and voice activity detection. Each task is represented by a special token that is jointly predicted by the decoder. It is trained on 680k hours of weakly-labeled speech data. We use the 770M parameter ‘medium’ version of this model, which consists of 48 transformer layers and is multilingual. The FC parameters take up 52% of the total parameter count.

We aim to reduce the parameters of the FC layers which results in fewer parameters during training and inference followed by low-rank adaptation for faster training. The choice of FC over the weights in the MHA is given as follows: the weights for query, key and value are already low-rank in a multihead configuration. Moreover, as mentioned earlier, results in [4, 2] clearly demonstrate that only updating the FC layers with low-rank weight matrices provides majority of the boost in performance from adapter-based methods. Thus, we propose to study parameter pruning via low-rank matrix factorization using SVD.

2.1. Low-rank matrix factorization with SVD

We apply SVD [20] to each linear layer in the MLP component of the transformer module. Weight $\mathbf{W}$ of a linear layer is factorized as $\mathbf{U\Sigma V}$, deriving two linear layers $\mathbf{B} = \mathbf{U}$ and $\mathbf{A} = \mathbf{\Sigma V}$. Effectively, we approximate $\mathbf{W} \approx \mathbf{BA}$. In both XLS-R and Whisper models, there are two linear layers with a non-linearity inbetween. The first layer projects a 1024-dimensional embedding to 4096 dimensions, and the second one reverses this projection. Each of these linear layers is now factorized into two linear components of rank r: one $m \times r$ and another $r \times n$, where the initial weight matrix is of shape $m \times n$. For $r = 512, 256, 128$, the size of the model reduces up to $\approx$ 24%, 44% and 54% respectively (the exact reduction is shared later in Section 4), thus reducing the total number of model parameters during both finetuning and inference.

SVD is applied prior to model finetuning. The directions corresponding to the lowest eigenvalues are removed. The models are then finetuned for 10 epochs for ASR and 2 epochs for LID, respectively.

In order to preserve the orthogonality of the matrices, we apply an orthonormal constraint after each update. We follow the update procedure from [14]. Unlike in AdaLoRA [21] (Adaptive Low Rank Adapters), the constraint is applied only on $\mathbf{U}$ in the above formulation.

Model pruning with SVD has been explored in techniques such as RankDyna [22], which uses an information criterion to select the directions to prune out. One disadvantage with such methods is the additional requirement of tracking the first order and second order momentum of parameter groups. AdaLoRA uses a similar approach during parameter-efficient finetuning to impose a parameter budget on adapter layers. We consider rank pruning strategies in the aforementioned works as a part of our future investigation.

2.2. Low-rank Adaptation

Low-rank adaptation (LoRA) based methods are now a common approach for PEFT of large pretrained models [2, 21, 23]. In LoRA, model finetuning is expressed as an update $\Delta\mathbf{W}$ to the original model parameters $\mathbf{W}_0$, where a low-rank structure is imposed on $\Delta\mathbf{W}$. Thus, the parameter size of $\Delta\mathbf{W}$ is significantly lesser than $\mathbf{W}_0$. The resource requirements for finetuning of large models are significantly reduced.

A key difference between finetuning after SVD and finetuning with LoRA based methods is that the former constrains the directions of the low-rank structure implicitly by choosing the eigenvectors as initial weights. It also limits the choices of the rank as reducing the value significantly can affect performance severely [22]. PEFT with LoRA therefore can be seen as a complementary technique to SVD. Thus, after reducing the parameters with SVD, we freeze the model and update the weights with LoRA instead of full-finetuning proposed in the previous subsection.

3. Datasets

3.1. ASR

We use the AMI Meeting Corpus [24] for ASR evaluation. Only the IHM part of the training and evaluation are used. The meetings were recorded in English using three different rooms with different acoustic properties, and include mostly non-native speakers. In our experiments, the training, development and test set consist of 77 h, 9 h and 8.7 h respectively. Three-fold speed perturbation is applied on the training data [25].

3.2. LID

We train our LID system on Voxlingua107 train, and evaluate it on Voxlingua107 dev and LRE17 eval sets.

Voxlingua107 [19]: The dataset consists of short speech segments extracted from YouTube videos, and labeled according to the language of the video title and description, with some post-processing steps to filter out false positives [19]. It contains 6628 h of speech from 107 languages which is used for training. The manually annotated dev set provided for 33 languages is used for evaluations. The duration of this set is 4.5 h

LRE17: We consider another evaluation set i.e., the NIST Language Recognition Evaluation dataset that consists of 14 languages and 3 parts: train, dev, and eval sets [26]. The splits contain 2061 h, 21 h, and 236 h of data, respectively. The eval set is used for our evaluations. In the files with duration greater than 30 s, we consider only the first 30 s of the file. To evaluate for language identification, the accent information is dropped in order to evaluate the data with the Whisper model.

4. Experiments

Experiments are conducted to evaluate the following: (1) full finetuning of models, (2) applying low-rank factorization during full finetuning, (3) parameter efficient finetuning (PEFT) with LoRA, and finally (4) PEFT with LoRA after low-rank factorization. Model finetuning and inference are done one a single GPU node with Nvidia RTX 3090. The next subsections provide details about model finetuning and evaluation setups. Model training after matrix factorization with SVD follow the same setup as full-finetuning.

4.1. ASR finetuning

The espresso [27] toolkit which uses fairseq [28] is used for finetuning XLS-R model. We use the end-to-end LF-MMI criterion [29] with a learning rate of 1e-5. In addition to the pretrained parameters, three layers of factorized TDNN (TDNN-F) [14] are added with a learning rate factor of 20 using the implementation from Pkwrap [30]. The standard AMI setup from Kaldi is modified to use only IHM recordings for training and evaluation [25]. The model is finetuned for 20000 updates with the learning rate scheduler tuned as follows: 20% warm-up, 60% constant learning rate, and 20% decay. The pretrained model parameters are frozen for the first 500 updates.

The k2/Icefall framework [1] is used for finetuning Whisper medium. Learning rate of 1e-5 and label smoothing loss with smoothing factor 0.1 is used. The features are masked with SpecAug during training [31]. AdamW [32] optimizer is used for all setups in this paper. A lower-cased version of the transcripts are used for training. finetuning is carried out for 10 epochs and the best model is chosen based on the WER on development set.

4.2. LID finetuning

Whisper model is finetuned with VoxLingua107 dataset with 33 languages subset included in the Voxlingua107 dev set to allow for rapid finetuning.

In Whisper, we used the start of the sentence token to extract only the language labels. During finetuning and evaluation, we do not constrain the languages to be one among the 33. That is, the classifier is allowed to predict any of the 99 languages. The classification is obtained by taking the language token with the maximum value over the logit vector obtained at the end of processing an audio.

4.3. Low rank adaptation

For experiments with XLS-R we implemented the low-rank adapters in the fairseq toolkit. For experiments with k2/ICefall we used the peft package's LoRA implementation. The parameter α, which adjusts the weight used (α/r where r is the rank of the matrix), is adjusted such that the weight is always 0.5.

4.4. Results

4.4.1. Rank of updates after full-finetuning

The ASR and LID systems are trained and evaluated independently. Before discussing the results on the two downstream tasks, we first analyze the rank of the updates on weight matrices obtained after regular finetuning. This is done in order to contrast the claim that task-dependent finetuning is low-rank in nature. Let W_0 is the weight matrix of a linear layer in the pretrained model and W_f is the corresponding weight matrix obtained after finetuning. According to the adapter terminology, $\Delta W = W_f - W_0$. We then evaluate the rank of ΔW for each weight matrix in each linear layer of the pretrained model. In every case, the matrix is full-rank (the rank is 1024 in the case of FC weights in the MLP layers) suggesting that the full finetuning may be inefficient.

4.4.2. ASR results

The performances of the ASR systems are presented in Table 1. We report Word Error Rate (WER%) for both dev and eval splits of the AMI dataset. All the ASR systems are evaluated after text normalization. The baseline WER% with Whisper is significantly better than that of the XLS-R model (10.8% vs 12.4% WER). For the XLS-R system, 12.4% WER is obtained after full finetuning. Matrix factorization results in a degradation of 0.7% WER absolute with a reduction of 23.8% of the parameters. Further reduction of the rank of the matrix to 256 results in total degradation of 1.5% WER (which is not as severe as reducing to 128 dimension with a total degradation of $\approx 7\%$ in WER). Similar trends are observed with the Whisper model. However, this could be partially attributed to the size of the model and the amount of training data used. The degradation when reducing

[1] https://github.com/k2-fsa/icefall

Table 1: *WER(%) of AMI dev and test sets evaluated with various rank for SVD on ASR task. No FT: Zero-shot evaluation, FT: full finetuning and LoRA indicates parameter efficient finetuning with LoRA.*

Rank	Config	XLS-R				Whisper medium			
		No. of params(M)		WER%		No. of params(M)		WER%	
		Training	Inference	dev	eval	Training	Inference	dev	eval
Full	No FT	-	-	-	-	776	776	23.3	22.9
	FT	315	315	14.3	12.4	776	776	13.4	10.8
	LoRA	28	315	15.0	13.6	55	776	11.1	9.6
512	FT	240	240	15.0	13.1	600	600	13.7	11.1
	LoRA	32	240	15.6	14.0	55	600	12.2	10.6
256	FT	177	177	15.7	13.9	485	485	14.8	11.9
	LoRA	32	177	17.3	15.6	55	485	14.9	13.8
128	FT	146	146	20.5	19.3	422	422	16.8	14.0
	LoRA	32	146	20.4	18.6	55	422	21.9	21.5

Table 2: *LID Accuracy on Voxlingua dev and LRE17 test sets for various rank in Whisper models in LID task. †: the numbers in the parentheses indicate the percentage of reduction in the number of parameters.*

Rank	Config	No. of params (M) †		Accuracy(%)	
		Training	Inference	dev	test
Full	No FT	776	776	92.6	86.1
	FT	776		97.1	91.2
	LoRA	55 (92.9)		96.4	90.8
512	FT	600 (22.7)	600 (22.7)	95.4	84.7
	LoRA	55 (92.9)		96.0	90.5
256	FT	485 (37.5)	485 (37.5)	92.8	80.5
	LoRA	55 (92.9)		90.3	80.1
128	FT	422 (45.0)	422 (45.0)	47.6	40.6
	LoRA	55 (92.9)		93.4	76.9

the rank to 512 and 256 is limited (0.3% and 1% absolute).

In case of the Whisper model, applying LoRA alone provided the best ASR performance (9.6% WER), which to the best our knowledge is the best ASR performance achieved on AMI-IHM data yet. The combination of factorization and LoRA performs as well as full-finetuning even after 22% reduction in model size. In addition, we are also able to take advantage of PEFT with only 55M parameters required during training.

When reducing the rank to 256 or 128 prior to training, a degradation in ASR performance was observed. However, the degradation with rank 256, where the model size reduces by up to 48.5%, the worst possible WER was only 11.9% with Whisper and 13.9% with XLS-R. WERs were beyond 20% when further reducing the rank from 256 to 128, suggesting the importance of information lost in the particular directions dropped.

4.4.3. LID results

The LID system is trained with only the Whisper model and evaluated for Voxlingua dev and LRE17 test sets using the LID accuracy(%). We observe trends similar to that reported in the previous section for ASR. The performance gap is negligible when combining low-rank factorization and finetuning with LoRA. The drop in performance is more noticeable on out-of-domain test set (LRE17) when using ranks of 256 and 128 for factorization. In particular, training stability was a concern when finetuning without adapters when using a rank of 128 for factorization.

Table 3: *Results of dev and eval sets of AMI when using orthonormal constraint during finetuning of the matrix factorized layers for XLS-R and Whisper models.*

Rank	WER(%)			
	dev	eval	dev	eval
	XLS-R		Whisper	
512	15.6	14.6	13.7	11.1
+ Orthonormal constraint	15.0	13.1	13.7	11.2
256	16.0	14.9	14.8	11.9
+ Orthonormal constraint	15.7	13.9	14.9	12.0
128	17.3	15.7	16.8	14.0
+ Orthonormal constraint	20.5	19.3	17.0	14.3

4.4.4. Effect of Orthonormal constraint

The effect of applying the orthonormal constraint after low-rank factorization is demonstrated in the results in Table 3. We only present the results on ASR systems for brevity. For ranks 512 and 256, the orthonormal constraint provides gains in WER% of up to 0.4% absolute. The advantage is observed only with the XLS-R system. With the Whisper models, no performance benefits are observed. When the rank is reduced to 128, a degradation of in WER up to 3.6% absolute is observed. Thus, for our experiments we did not employ the orthonormal constraint when using rank=128.

5. Conclusion

In this paper, we studied the application of low-rank factorization with SVD on the fully connected layers of the feedforward components of the Transformer model for speech and language recognition tasks. We studied the technique independently and in combination with parameter-efficient finetuning with Low Rank adapters (LoRA). We observe that by reducing the rank of the fully connected layers from 1024 to 512, thereby reducing model parameter effectively by 22.7% for the Whisper medium model and 28% for the XLS-R model, no performance degradation was observed. Further reduction of the rank to 256 introduced performance trade-offs. Our future work aims at addressing this behaviour.

6. References

[1] A. Aghajanyan, L. Zettlemoyer, and S. Gupta, "Intrinsic dimensionality explains the effectiveness of language model finetuning," *arXiv preprint arXiv:2012.13255*, 2020.

[2] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[3] G. Vanderreydt, A. Prasad, D. Khalil, S. Madikeri, K. Demuynck, and P. Motlicek, "Parameter-efficient tuning with adaptive bottlenecks for automatic speech recognition," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–7.

[4] Z. Lin, A. Madotto, and P. Fung, "Exploring versatile generative language model via parameter-efficient transfer learning," *arXiv preprint arXiv:2004.03829*, 2020.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] P. Motlicek, P. N. Garner, N. Kim, and J. Cho, "Accent adaptation using subspace gaussian mixture models," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7170–7174.

[7] P. Motlicek, D. Povey, and M. Karafiat, "Feature and score level combination of subspace gaussinas in lvcsr task," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7604–7608.

[8] P. Motlicek, S. Dey, S. Madikeri, and L. Burget, "Employment of subspace gaussian mixture models in speaker recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4445–4449.

[9] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Interspeech*, 2013, pp. 2365–2369.

[10] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model compression applied to small-footprint keyword spotting," 2016.

[11] R. Prabhavalkar, O. Alsharif, A. Bruguier, and L. McGraw, "On the compression of recurrent neural networks with an application to lvcsr acoustic modeling for embedded speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5970–5974.

[12] H. Du, X. Tian, L. Xie, and H. Li, "Wavenet factorization with singular value decomposition for voice conversion," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 152–159.

[13] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6655–6659.

[14] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Interspeech*, 2018, pp. 3743–3747.

[15] A. Prasad, A. Carofilis, G. Vanderreydt, D. Khalil, S. Madikeri, P. Motlicek, and C. Schuepbach, "Fine-tuning self-supervised models for language identification using orthonormal constraint," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 11 921–11 925.

[16] A. Babu *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. of Interspeech*, 2022, pp. 2278–2282.

[17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[18] A. Baevski *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[19] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *Proc. IEEE SLT Workshop*, 2021.

[20] G. Strang, *Linear algebra and its applications*, 2012.

[21] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adaptive budget allocation for parameter-efficient finetuning," *arXiv preprint arXiv:2303.10512*, 2023.

[22] T. Hua, X. Li, S. Gao, Y.-C. Hsu, Y. Shen, and H. Jin, "Dynamic low-rank estimation for transformer-based language models," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 9275–9287.

[23] D. J. Kopiczko, T. Blankevoort, and Y. M. Asano, "Vera: Vector-based random matrix adaptation," *arXiv preprint arXiv:2310.11454*, 2023.

[24] W. Kraaij, T. Hain, M. Lincoln, and W. Post, "The ami meeting corpus," in *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.

[25] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. of IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[26] S. O. Sadjadi *et al.*, "The 2017 NIST language recognition evaluation." in *Odyssey*, 2018, pp. 82–89.

[27] Y. Wang *et al.*, "Espresso: A fast end-to-end neural speech recognition toolkit," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 136–143.

[28] M. Ott *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[29] A. Vyas, S. Madikeri, and H. Bourlard, "Lattice-free mmi adaptation of self-supervised pretrained acoustic models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6219–6223.

[30] S. Madikeri, S. Tong, J. Zuluaga-Gomez, A. Vyas, P. Motlicek, and H. Bourlard, "Pkwrap: a pytorch package for lf-mmi training of acoustic models," *arXiv preprint arXiv:2010.03466*, 2020.

[31] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.