EFFICIENT ADAPTER TRANSFER OF SELF-SUPERVISED SPEECH MODELS FOR AUTOMATIC SPEECH RECOGNITION

Bethan Thomas[†] *Samuel Kessler*[*‡] *Salah Karout*[†]

[†]Huawei R&D UK [‡]University of Oxford

ABSTRACT

Self-supervised learning (SSL) is a powerful tool that allows learning of underlying representations from unlabeled data. Transformer based models such as wav2vec 2.0 and HuBERT are leading the field in the speech domain. Generally these models are fine-tuned on a small amount of labeled data for a downstream task such as Automatic Speech Recognition (ASR). This involves re-training the majority of the model for each task. Adapters are small lightweight modules which are commonly used in Natural Language Processing (NLP) to adapt pre-trained models to new tasks. In this paper we propose applying adapters to wav2vec 2.0 to reduce the number of parameters required for downstream ASR tasks, and increase scalability of the model to multiple tasks or languages. Using adapters we can perform ASR while training fewer than 10% of parameters per task compared to full fine-tuning with little degradation of performance. Ablations show that applying adapters into just the top few layers of the pre-trained network gives similar performance to full transfer, supporting the theory that higher pre-trained layers encode more phonemic information, and further optimizing efficiency.

***Index Terms*—** Automatic Speech Recognition, Self-Supervision, Adapters, Transfer Learning

1. INTRODUCTION

Self-supervision is now standard practice in Natural Language Processing (NLP), with models such as BERT [1] achieving state of the art results. More recently, self-supervised learning (SSL) approaches have been applied to speech tasks with great success. Recent works such as wav2vec 2.0 [2] and HuBERT [3] show that self-supervision in speech can achieve state of the art results.

Most SSL models rely on unsupervised pre-training followed by supervised fine-tuning on a downstream task. The unsupervised pre-training allows a model to benefit from the large amounts of unlabeled data which are readily available. Fine-tuning uses a comparatively small amount of labeled data and retrains the majority of the self-supervised model to the desired downstream task.

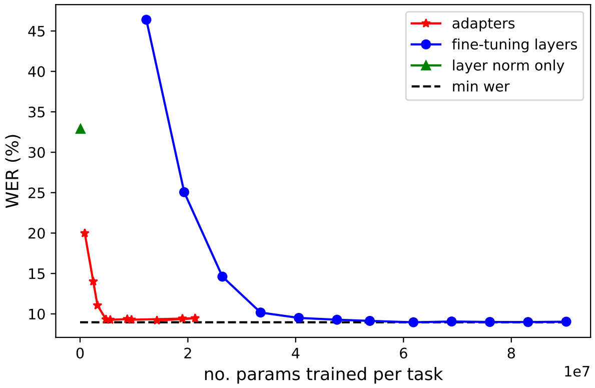

Fig. 1. A comparison of trained model parameters vs WER. Pre-trained self-supervised models are either fine-tuned or adapted for ASR using 10 hours of supervised data and evaluated on LibriSpeech dev-clean. The Y axis shows various iterations of fine-tuning/adapters with different numbers of layers trained, and thus different numbers of trainable parameters. Adapters achieve a similar performance in WER to fine-tuning with only a fraction of parameters. Training just layer normalizations performs poorly.

While these approaches gain excellent results, fine-tuning the model is computationally expensive and does not scale well to multiple tasks, such as in the case of multi-lingual Automatic Speech Recognition (ASR); a complete set of fine-tuned parameters must be learnt and stored per downstream task. This is commonly in the scale of $O(10^8)$ parameters per task. Once the model is fine-tuned for one task, the entire model is fixed for that task, and the base model must be reloaded to transfer to future tasks.

Adapters are small trainable modules that can be applied into the layers of a frozen pre-trained network and tuned to a particular task. Recently these have been applied to pre-trained unsupervised models in NLP [4, 5]. The benefit of adapters is that they allow adaptation to new tasks with a relatively small number of parameters per task. As the existing model parameters remain frozen, the original model can support multiple downstream tasks, only a set of adapter parameters are required per task. This makes adapters more efficient to train, and highly scalable to multiple tasks.

The task of ASR is very well researched for high re-

[*]Work performed during an internship with Huawei R&D UK

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

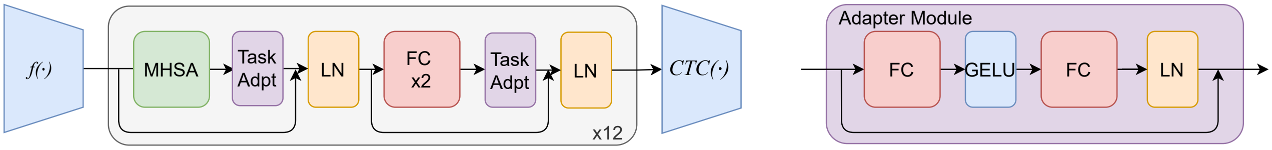

Fig. 2. Left: structure of wav2vec 2.0 model with task specific adapter modules. *f(·)* is a convolutional encoder which is followed by 12 transformer encoder blocks. For downstream ASR a linear classifier, *CTC(·)*, is applied to the output of the transformer blocks. For adapter transfer, adapter modules are inserted into the model, and during adaptation to the downstream ASR task only the adapters, layer normalization layers and linear classifier are trained, the rest of the network is frozen. Right: structure of adapter module consisting of a down-projection, a non-linearity and an up-projection, with a skip connection.

source languages such as English. However, it is more difficult to gain good performance on languages where there is limited paired text and audio training data. wav2vec 2.0 shows good performance with limited labeled data, and multilingual wav2vec 2.0 [6] shows good results across a range of languages including languages unseen during pre-training. Self-supervised speech models are able to leverage generic features from pre-training to adapt well to scenarios with limited labeled data and are thus ideal for multi-lingual ASR. However, it is impractical to train and store fine-tuned parameters for every language. Furthermore, fine-tuning overwrites pre-trained model parameters so may not be the optimal way of utilizing pre-training knowledge.

In this paper we apply adapters to a pre-trained wav2vec 2.0 model in order to transfer the pre-trained representations to the task of downstream ASR. Adapters suggest a way of utilizing these pre-trained representations in a much more efficient manner, with many fewer parameters required per downstream task than usual fine-tuning methods. This has applications for multi-task scenarios such as multi-lingual ASR, and also increases the accessibility of self-supervised speech models for research. To the best of our knowledge this is the first time adapter modules have been applied to a self-supervised speech model for ASR.

2. BACKGROUND

2.1. Self-supervised ASR

Labeled speech data is relatively scarce and expensive to generate. Self-supervised methods are able to take advantage of the huge amount of unlabeled speech data available to learn generic speech features.

There have been various proposed methods such as CPC [7], APC [8], and wav2vec [9]. All aim to learn an encoded representation of a raw speech waveform using various self-supervised tasks which train the underlying network. Many of these models include some form of reconstruction, where masked representations, or elements of the original input waveform, are recreated. The underlying network can then be used to extract representations which are used as input to downstream tasks. This is instead of more conventional speech features such as log Mel filter-bank features. These networks can either be frozen, or further optimized to the downstream task by some form of transfer learning.

wav2vec 2.0 is a leading model in this field [2]. However, the usual method of fine-tuning wav2vec 2.0 for downstream ASR requires re-training most of the model layers. In fact 95.6% of the total model parameters are trained, and must therefore be stored, per task. This is not an issue if monolingual ASR is the only target task. However it is not ideal for multi-lingual speech recognition, or indeed utilizing the wav2vec 2.0 encoder for multiple tasks such as speech translation, as in [10].

Recently advances have been made to the wav2vec 2.0 architecture including adding self-training [11]. While we use wav2vec 2.0 in this work, our approach would scale well to any self-supervised transformer based speech model, including HuBERT [3].

2.2. Adapters

Adapters were introduced in NLP as an efficient method of transfer learning [4]. It was found that adapters approach the performance of full fine-tuning with only a fraction of the parameters in NLP tasks using a pre-trained BERT model. Adapters have also been successfully applied to NMT [12] and vision tasks [13]. Recently dual adapters have been proposed for multi-lingual transfer [5], where one adapter module is used to adapt for language, and another captures task specific adaptations.

Adapters have also been applied to speech in streaming RNNT models [14] and hybrid CTC-attention speech transformers [15] to solve the issue of multi-lingual speech recognition. More recently adapters have been used in speech translation to combine pre-trained modules [10]. Adapters have also been employed for efficient SSL pre-training of new tasks in a continual learning setting [16].

We hypothesize that we will see the same benefits of adapters in a speech model as in an NLP model, namely parameter efficient transfer of the pre-trained network to a downstream task with little performance degradation.

3. METHOD

In wav2vec 2.0 a convolutional encoder takes as input the raw waveform. Then a transformer model is applied to the encoded output. The model is trained by masking input frames and comparing predicted values with positive and negative quantized versions of the masked frames. This is similar to contrastive predictive coding [7].

When applying wav2vec 2.0 to a downstream model the feature encoder is frozen, and a linear classifier is added on top of the transformer context model for fine-tuning. Generally the transformer model is also frozen for the first N updates. When applying wav2vec 2.0 to ASR, the linear classifier and model are fine-tuned with a Connectionist Temporal Classification (CTC) loss using a small amount of labeled speech data.

Adapters are small bottleneck modules consisting of a down projection, a non-linearity, and an up-projection, with a skip connection (see Fig. 2). The initial implementation [4] applies adapters after both the self-attention and feedforward layers. However it is possible to apply adapters in different positions throughout the transformer block [12]. The fully connected layers are initialized as a near identity function. The identity initialization and the skip connection allow the module to be ignored if not deemed necessary during training.

In our experiments with adapters we apply adapters twice in each transformer block. Our experimentation showed that this gave the best results. We apply a linear classifier on top of the transformer network. A set of adapters, layer normalization layers and a linear classifier are trained per task using a CTC loss, the rest of the network remains frozen.

4. EXPERIMENTS AND RESULTS

We use the standard wav2vec 2.0 BASE architecture which contains 12 transformer layers, and use the publicly released pre-trained checkpoint from `fairseq` [17] which has been pre-trained using the LibriSpeech [18] dataset. We use a standard size of 256 for all adapters, and initially apply adapters into every transformer layer.

We also run our own fine-tuning experiments as a comparison and follow the fine-tuning format of wav2vec 2.0. By tuning hyper-parameters we are able to improve on the word error rate (WER) values reported in that paper [2]. We trained for 20k steps, with the transformer layers frozen for just the first 4k updates, and a learning rate of 5e-5.

We investigate performance of fine-tuning and adapters on the 10 hour supervised subset of the LibriLight (LL) dataset [19], and evaluate on standard LibriSpeech dev sets. We also experiment with French ASR to demonstrate the multi-task scenario. We take a 10h subset of the Common Voice (CV) [20] corpus and evaluate on the CV test set. We calculate WER in all cases. We found that the optimal setup for adapter transfer was to run for 10k steps with a learning rate of 5e-4. All experiments are run on 8 V100 GPUs, and without language model fusion to enable distraction free comparison of transfer method.

Table 1. A pre-trained BASE wav2vec2 model is transferred to the downstream ASR task using the 10 hour LibriLight (LL) supervised set, or a random 10 hour subset of the French Common Voice (CV) corpus. Word error rate (WER) is reported on the dev-clean/dev-other sets of LibriSpeech for English, and the CV test set for French. Results are all without a language model. % of trainable parameters compared to the total model parameters are also reported

	wav2vec 2.0 FT[2]	Fine-tune	Adapter
10h LL dev-clean	10.9%	8.98%	9.39%
10h LL dev-other	17.4%	16.9%	17.0%
10h French CV test	N/A	40.2%	39.4%
% trained params	95.6%	95.6%	9.2%

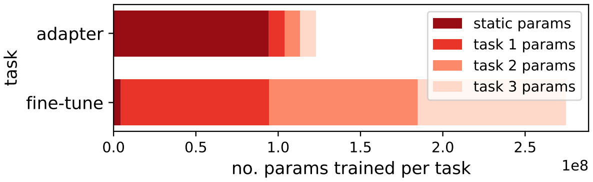

Fig. 3. A summary of number of trained parameters per task with fine-tuning and adapters. Each fine-tuning task nearly doubles the number of trained parameters which must be learnt and stored. Whereas adapters add only a small number of additional parameters per task showing their scalability to multi-task scenarios.

Adapters perform slightly worse than fine-tuning on English ASR (see table 1), however the absolute WER increase is just 0.41% and 0.17% for dev-clean and dev-other respectively. This performance decrease comes with a huge decrease in the number of trained task-specific parameters. Fine-tuning requires training 95.6% of parameters, our adapter approach only trains 9.2% of parameters. For French ASR, there is in fact a slight performance increase when using adapters compared to traditional fine-tuning.

However, the real benefit of this approach comes when scaling to multiple languages or tasks, as can be seen in Fig. 3. Each fine-tuning task nearly doubles the required number of parameters which must be learnt and stored, however adapters add only a small number of additional parameters per task. This makes adapter transfer much more scalable than fine-tuning while attaining a similar performance.

Our results show that even when there is a mismatch between pre-trained and downstream language, in the case of French transfer, both fine-tuning and adapters are able to achieve some success in ASR. This is unsurprising for fine-tuning, since the language specific features of the pre-trained network are overwritten during training. However, with adapter transfer, the original network remains unchanged. Therefore the adapters are able to compensate for the language mismatch, and learn both language and task specific features. This is in contrast to [5] which uses separate

Table 2. A pre-trained bilingual (French and English) model is transferred to English and French downstream ASR using fine-tuning and adapters. Results are reported in WER.

	Fine-tune	Adapter
10h LL dev-clean	12.16%	12.91%
10h French CV test	27.75%	28.25%

adapters for task and language.

It is also worth noting that adapter experiments run more quickly than fine-tuning experiments, both due to the reduction in trainable parameters which increases training speed, and the smaller optimal number of training steps. Adapter experiments do not depend on a freeze steps hyper-parameter, and run successfully with a simple bi-stage learning rate schedule, as in [4], rather than the more complex tri-stage scheduler required for traditional fine-tuning [2]. All these factors make this adapter approach much more experimentally friendly for researchers.

We also pre-trained our own bi-lingual English and French wav2vec 2.0 model using approximately 1000 hours of French CV data, as well as 960h of English LibriSpeech data, and ran English and French ASR experiments (see table 2). This demonstrates that our adapter method is also valid for multi-lingual pre-trained models as adapters again get within close performance of fine-tuning.

Finally, we tested training just the layer normalization parameters as suggested in [4]. However, as found in that work, this approach does not perform well, WER on LibriSpeech dev-clean was 32.9%. This provides evidence that the adapter modules themselves improve performance, rather than the re-trained layer normalization parameters.

4.1. Ablations

Prior work suggests that lower layers of self-supervised models contain more generic speech features, and higher layers contribute more to phone discrimination [8]. Therefore we investigate training just the top N layers of the network using both fine-tuning and adapters. All experiments are run with the BASE English wav2vec 2.0 model, 10 hours LibriLight training data and evaluated on LibriSpeech dev-clean.

Using adapters in just the top 4 layers, out of a total 12 layers, gives nearly as good performance as adding adapters into every layer, and in fact using just 6 adapters gives the best performance with 9.27% WER. This immediately halves the number of required parameters to just 4.85% of total parameters trained. It is also possible to optimize fine-tuning; training just the top 8 transformer layers gives best performance. However fine-tuning requires more layers to be trained than using adapters, and even 8 layers equates to training 65.5% of parameters (see Fig. 4). When just one layer is trained, the method with adapters performs significantly better showing that adapters are better able to utilize the pre-training knowledge of the entire network.

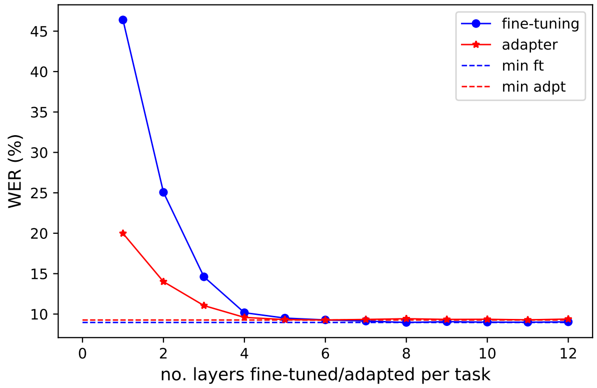

Fig. 4. A comparison of trainable layers vs WER. For fine-tuning experiments, transformer layers of the model are optionally trained or frozen in a top down manner and evaluated for WER on LibriSpeech dev-clean. For adapter experiments, adapters are only inserted into the top N layers. Models with adapters perform better than models which are fine-tuned when the number of trained layers is limited.

The curves presented in Fig. 4 show that the top layers of the network are more important for downstream performance than the bottom layers, supporting the hypothesis that the higher layers of the network encode more phonemic information, and lower layers more generic speech information.

5. DISCUSSION & CONCLUSION

This work provides an insight into how self-supervised speech models can be utilized in a more parameter efficient manner without sacrificing performance. When transferring to a downstream task, fine-tuning the majority of the model is still generally the best performing method. However, using adapters instead of fine-tuning achieves close to that performance with a fraction of parameters and takes less training time. This allows quicker and cheaper experimentation with models such as wav2vec 2.0 and HuBERT, and increases scalability to multiple tasks, for example multi-lingual ASR. More work could be done on utilizing adapters for additional tasks such as speaker recognition and speech translation.

Ablations show that it is unnecessary to transfer all layers of the network to the downstream task, only the top N layers, thereby supporting the theory that the higher layers of these pre-trained networks encode more phonemic information. Adapters are better able to utilize pre-trained information as we achieve better performance with one layer of adapters, than with one layer of fine-tuning, and best performance comes from adapting fewer layers compared to fine-tuning.

While we are the first to utilize adapters in this way for speech, our findings are similar to those in the NLP domain [4, 5]. It would be interesting to perform the same layer ablations in the NLP domain. More broadly, we show that the speech domain can benefit from future NLP work on SSL.

6. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[3] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[4] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[5] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder, "Mad-x: An adapter-based framework for multi-task cross-lingual transfer," *arXiv preprint arXiv:2005.00052*, 2020.

[6] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[7] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[8] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

[9] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[10] Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier, "Lightweight adapter tuning for multilingual speech translation," *arXiv preprint arXiv:2106.01463*, 2021.

[11] Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli, "Self-training and pre-training are complementary for speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3030–3034.

[12] Ankur Bapna, Naveen Arivazhagan, and Orhan Firat, "Simple, scalable adaptation for neural machine translation," *arXiv preprint arXiv:1909.08478*, 2019.

[13] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi, "Learning multiple visual domains with residual adapters," *arXiv preprint arXiv:1705.08045*, 2017.

[14] Anjuli Kannan, Arindrima Datta, Tara N Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," *arXiv preprint arXiv:1909.05330*, 2019.

[15] Genta Indra Winata, Guangsen Wang, Caiming Xiong, and Steven Hoi, "Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition," *arXiv preprint arXiv:2012.01687*, 2020.

[16] Samuel Kessler, Bethan Thomas, and Salah Karout, "Continual-wav2vec2: an application of continual learning for self-supervised automatic speech recognition," *arXiv preprint arXiv:2107.13530*, 2021.

[17] 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.

[18] 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.

[19] Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7669–7673.

[20] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.