Effects of Layer Freezing on Transferring a Speech Recognition System to Under-resourced Languages

Onno Eberhard and **Torsten Zesch**
Language Technology Lab
University of Duisburg-Essen, Germany
onno.eberhard@stud.uni-due.de
torsten.zesch@uni-due.de

Abstract

In this paper, we investigate the effect of layer freezing on the effectiveness of model transfer in the area of automatic speech recognition. We experiment with Mozilla's DeepSpeech architecture on German and Swiss German speech datasets and compare the results of either training from scratch vs. transferring a pre-trained model. We compare different layer freezing schemes and find that even freezing only one layer already significantly improves results.

1 Introduction

The field of automatic speech recognition (ASR) is dominated by research specific to the English language. There exist plenty available text-to-speech models pre-trained on (and optimized for) English data. When it comes to a low-resource language like Swiss German, or even standard German, only a very limited number of small-scale models is available. In this paper, we train Mozilla's implementation[1] of Baidu's DeepSpeech ASR architecture (Hannun et al., 2014) on these two languages. We use transfer learning to leverage the availability of a pre-trained English version of DeepSpeech and observe the difference made by freezing different numbers of layers during training.

2 Transfer Learning and Layer Freezing

Deep neural networks can excel at many different tasks, but they often require very large amounts of training data and computational resources. To remedy this, it is often advantageous to employ transfer learning: Instead of initializing the parameters of the network randomly, the optimized parameters of a network trained on a similar task are reused. Those parameters can then be fine-tuned to the specific task at hand, using less data and fewer computational resources. In the fine-tuning process many parameters of the original model may be "frozen", i.e. held constant during training. This can speed up training and improve results when less training data is available (Kunze et al., 2017). The idea of taking deep neural networks trained on large datasets and fine-tuning them on tasks with less available training data has been popular in computer vision for years (Huh et al., 2016). More recently, with the emergence of end-to-end deep neural networks for automatic speech recognition (like DeepSpeech), it has also been used in this area (Kunze et al., 2017; Li et al., 2019).

Deep neural networks learn representations of the input data in a hierarchical manner. The input is transformed into simplistic features in the first layers of a neural network and into more complex features in the layers closer to the output. If we assume the simplistic feature representations are applicable in similar, but different, contexts, layer-wise freezing of parameters seems like a good choice. This is further reinforced by findings from image classification (Yosinski et al., 2014), where the learned features can additionally be nicely visualized (Zeiler and Fergus, 2014).

As for automatic speech recognition, the representations learned by the layers is not as clear-cut as within image processing. Nonetheless, some findings, for example that affricates are better represented at later layers in the network (Belinkov and Glass, 2017), seem to affirm the hypothesis that the later layers learn more abstract features and earlier layers learn more primitive features. This is important for fine-tuning, because it only makes sense to freeze parameters if they don't need to be adjusted for the new task. If it is known that the first layers of a network learn to identify "lower-level"-features, i.e. simple shapes in the context of

[1] https://github.com/mozilla/DeepSpeech

	Dataset	Hours	Speakers
Pre-training	English	>6,500	?
Transfer	German	315	4,823
	Swiss German	70	191

Table 1: Overview of datasets

image processing or simple sounds in the context of ASR, these layers can be frozen completely during fine-tuning.

3 Experimental Setup

In our experiments, we transfer an English pre-trained version of DeepSpeech to German and to Swiss German data and observe the impact of freezing fewer or more layers during training.

3.1 Datasets

We trained the models for (standard) German on the German part of the Mozilla Common Voice speech dataset (Ardila et al., 2020). The utterances are typically between 3 and 5 seconds long and are collected from and reviewed by volunteers. This collection method entails a rather high number of speakers and quite some noise. The Swiss German models were trained on the data provided by Plüss et al. (2020). This speech data was collected from speeches at the Bernese parliament. The English pre-trained model was trained by Mozilla on a combination of English speech datasets, including LibriSpeech and Common Voice English.[2] The datasets for all three languages are described in Table 1. For inference and testing we used the language model KenLM (Heafield, 2011), trained on the corpus described by Radeck-Arneth et al. (2015, Section 3.2). This corpus consists of a mixture of texts from the sources Wikipedia and Europarl as well as crawled sentences. The whole corpus was preprocessed with MaryTTS (Schröder and Trouvain, 2003).

3.2 ASR Architecture

We use Mozilla's DeepSpeech version 0.7 for our experiments. The implementation differs in many ways from the original model presented by Hannun et al. (2014). The architecture is described in detail in the official documentation[3] and is depicted in Figure 1. From the raw speech data, Mel-Frequency Cepstral Coefficients (Imai, 1983) are

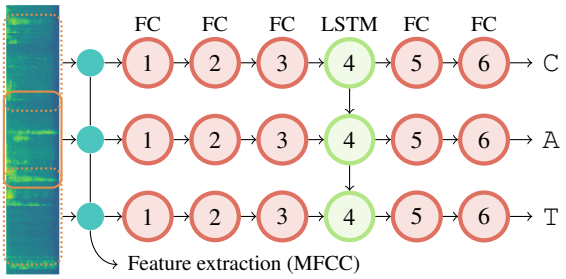

Figure 1: DeepSpeech architecture. The fully connected (FC) layers 1 – 3 and 5 are ReLU activated, the last layer uses a softmax function to compute character probabilities.

extracted and passed to a 6-layer deep recurrent neural network. The first three layers are fully connected with a ReLU activation function. The fourth layer is a Long Short-Term Memory (LSTM) unit (Hochreiter and Schmidhuber, 1997); the fifth layer is again fully connected and ReLU activated. The last layer outputs probabilities for each character in the language's alphabet. It is fully connected and uses a softmax activation for normalization. The character-probabilities are used to calculate a Connectionist Temporal Classification (CTC) loss function (Graves et al., 2006). The weights of the model are optimized using the Adam method (Kingma and Ba, 2014) with respect to the CTC loss.

3.3 Training Details

As a baseline, we directly train the German and Swiss German model on the available data from scratch, without any transfer (hereafter called "Baseline"). To assess the effects of layer freezing, we then re-train the model based on weight initialization from the English pre-trained model.[4] In this step, we freeze the first N layers during training, where $N = 0, \ldots, 5$. For $N = 4$ we additionally experiment with freezing the 5$^{\text{th}}$ layer instead of the LSTM layer, which we denote as "Layers 1-3,5 Frozen". We do this because we see the LSTM as the most essential and flexible part of the architecture; the 5$^{\text{th}}$ and 6$^{\text{th}}$ layer have a simpler interpretation as transforming the LSTM hidden state into character-level information. This stage should be equivalent across languages, as long as the LSTM hidden state is learned accordingly, which is ensured by not freezing the LSTM. For all models, we reinitialize the last layer, because of the different alphabet sizes of German / Swiss German and

[2] https://github.com/mozilla/DeepSpeech/releases/tag/v0.7.0

[3] https://deepspeech.readthedocs.io/en/latest/DeepSpeech.html

[4] https://github.com/mozilla/DeepSpeech/releases

English (ä, ö, ü), but don't reinitialize any other layers (as done e.g. by Hjortnaes et al. (2020)). The complete training script, as well as the modified versions of DeepSpeech that utilize layer freezing are available online[5]. The weights were frozen by adding `trainable=False` at the appropriate places in the TensorFlow code, though some other custom modifications were necessary and are described online[5]. For Swiss German, we do not train the network on the German dataset first and transfer from German to Swiss German, as this has been shown to lead to worse results (Agarwal and Zesch, 2020).

3.4 Hyperparameters & Server

In training each model, we used a batch size of 24, a learning rate of 0.0005 and a dropout rate of 0.4. We did not perform any hyperparameter optimization. The training was done on a Linux machine with 96 Intel Xeon Platinum 8160 CPUs @ 2.10GHz, 256GB of memory and an NVIDIA GeForce GTX 1080 Ti GPU with 11GB of memory. Training the German language models for 30 epochs took approximately one hour per model. Training the Swiss German models took about 4 hours for 30 epochs on each model. We did not observe a correlation between training time and the number of frozen layers. For testing, the epoch with the best validation loss during training was taken for each model.

4 Results & Discussion

Results of our baselines are very close to the values reported for German by Agarwal and Zesch (2019) and Swiss German by Agarwal and Zesch (2020) using the same architecture.

The test results for both languages from the different models described in Section 3.3 are compiled in Table 2. Figures 2 and 3 show the learning curves for all training procedures for German and Swiss German, respectively. The epochs used for testing (cf. Table 2) are also marked in the figures.

For both languages, the best results were achieved by the models with the first two to three layers frozen during training. It is notable however, that the other models that utilize layer freezing are not far off, the learning curves look remarkably similar (in both plots, these are the lower six curves). For both languages, these models achieve much better results than the two models without layer freezing ("Baseline" and "0 Frozen Layers"). The results seem to indicate that freezing the first layer brings the largest advantage in training, with diminishing returns on freezing the second and third layers. For German, additionally freezing the fourth or fifth layer slightly worsens the result, though interestingly, freezing both results in better error rates. This might however only be due to statistic fluctuations, as it can be seen in Figure 2 that on the validation set, the model with 5 frozen layers performs worse than those with 3 or 4 frozen layers. For Swiss German, the result slightly worsens when the third layer is frozen and performance further drops when freezing subsequent layers. Similar results were achieved by Ardila et al. (2020), where freezing two or three layers also achieved the best transfer results for German, with a word error rate of 44%. They also used DeepSpeech and a different version of the German Common Voice dataset.

	German		Swiss	
Method	WER	CER	WER	CER
Baseline	.70	.42	.74	.52
0 Frozen Layers	.63	.37	.76	.54
Layer 1 Frozen	.48	.26	.69	.48
Layers 1-2 Frozen	**.44**	**.22**	**.67**	**.45**
Layers 1-3 Frozen	**.44**	**.22**	.68	.47
Layers 1-4 Frozen	.45	.24	.68	.47
Layers 1-3,5 Frozen	.46	.25	.68	.46
Layers 1-5 Frozen	**.44**	.23	.70	.48

Table 2: Results on test sets (cf. Section 3.3)

The results don't show a significant difference between freezing the fourth or the fifth layer of the network ("Layers 1-4 Frozen" vs. "Layers 1-3,5 Frozen"). This indicates that the features learned by the LSTM are not as language-specific as we hypothesized. It might even be that, in general, it does not matter much which specific layers are frozen, if the number of frozen parameters is the same. It might be interesting to see what happens if the last instead of the first layers are frozen (not necessarily with this architecture), thereby breaking the motivation of hierarchically learned features, with later layers being more task-specific.

It is interesting that the models with four or five frozen layers, i.e. only 2 or 1 learnable layers, still achieve good results. This indicates that the features extracted by DeepSpeech when trained on

[5] https://github.com/onnoeberhard/deepspeech

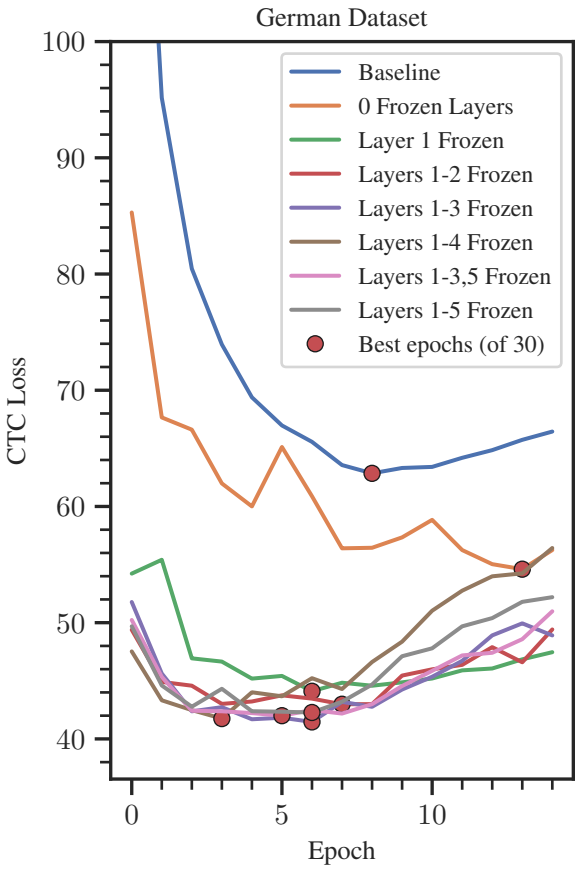

Figure 2: Learning curves (validation loss) on the German dataset. Layer freezing has a noticeable impact, but how many layers are frozen does not seem to make much of a difference. See Section 3.3 for details.

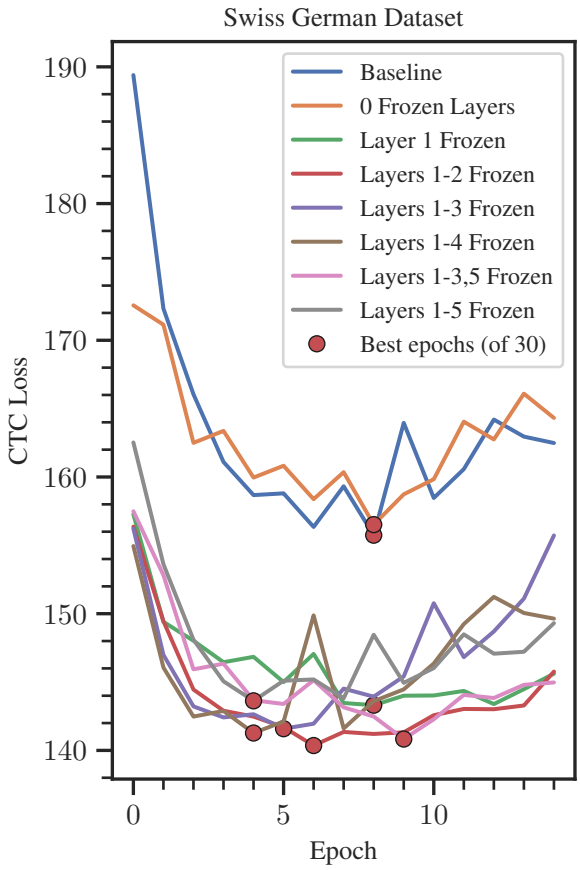

Figure 3: Learning curves (validation loss) on the Swiss German dataset. Compare with Figure 2.

English are general enough to really be applicable for other languages as well. It is probable that with a larger dataset the benefits of freezing weights decrease and better results are achieved with freezing fewer or no layers. For both languages it is evident that the transfer learning approach is promising.

Limitations Our experiment is limited to a transfer between closely related languages. For example, when just transcribing speech there is no need for such a model to learn intonation features. This might be a problem when trying to transfer such a pre-trained model to a tonal language like Mandarin or Thai. There might also be phonemes that don't exist or are very rare in English but abundant in other languages.

5 Summary

We investigate the effect of layer freezing on the effectiveness of transferring a speech recognition model to a new language with limited training data. We find that transfer is not very effective without layer freezing, but that already one frozen layer yields quite good results. The differences between freezing schemes are surprisingly small, even when freezing all layers but the last.

Acknowledgements

We want to thank Aashish Agarwal for valuable help in setting up DeepSpeech and for providing preprocessing scripts as well as the hyperparameters we used for training.

References

Aashish Agarwal and Torsten Zesch. 2019. German end-to-end speech recognition based on deepspeech. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 111–119, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Aashish Agarwal and Torsten Zesch. 2020. Ltl-ude at low-resource speech-to-text shared task: Investigating mozilla deepspeech in a low-resource setting.

Rosana Ardila, Megan Branson, Kelly Davis, Michael

Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4218–4222. European Language Resources Association.

Yonatan Belinkov and James Glass. 2017. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Advances in Neural Information Processing Systems*, volume 30, pages 2441–2451.

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

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep speech: Scaling up end-to-end speech recognition.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Nils Hjortnaes, Niko Partanen, Michael Rießler, and Francis M. Tyers. 2020. Towards a speech recognizer for Komi, an endangered and low-resource uralic language. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 31–37, Wien, Austria. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Minyoung Huh, Pulkit Agrawal, and Alexei A. Efros. 2016. What makes imagenet good for transfer learning?

Satoshi Imai. 1983. Cepstral analysis synthesis on the mel frequency scale. In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 93–96. IEEE.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Julius Kunze, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. 2017. Transfer learning for speech recognition on a budget. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 168–177. Association for Computational Linguistics.

Bryan Li, Xinyue Wang, and Homayoon S. M. Beigi. 2019. Cantonese automatic speech recognition using transfer learning from mandarin. *CoRR*.

Michel Plüss, Lukas Neukom, and Manfred Vogel. 2020. Germeval 2020 task 4: Low-resource speech-to-text.

Stephan Radeck-Arneth, Benjamin Milde, Arvid Lange, Evandro Gouvea, Stefan Radomski, Max Mühlhäuser, and Chris Biemann. 2015. Open Source German Distant Speech Recognition: Corpus and Acoustic Model. In *Proceedings Text, Speech and Dialogue (TSD)*, pages 480–488, Pilsen, Czech Republic.

Marc Schröder and Jürgen Trouvain. 2003. The german text-to-speech synthesis system mary: A tool for research, development and teaching. *International Journal of Speech Technology*, 6(4):365–377.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3320–3328.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833. Springer International Publishing.