Transfer Learning for Speech Recognition on a Budget

Julius Kunze[1], Louis Kirsch[1], Ilia Kurenkov[2], Andreas Krug[2],
Jens Johannsmeier[2], and Sebastian Stober[2]

[1]Hasso Plattner Institute, Potsdam, Germany
juliuskunze@gmail.com, mail@louiskirsch.com
[2]University of Potsdam, Potsdam, Germany
{kurenkov,ankrug,johannsmeier,sstober}@uni-potsdam.de

Abstract

End-to-end training of automated speech recognition (ASR) systems requires massive data and compute resources. We explore transfer learning based on model adaptation as an approach for training ASR models under constrained GPU memory, throughput and training data. We conduct several systematic experiments adapting a Wav2Letter convolutional neural network originally trained for English ASR to the German language. We show that this technique allows faster training on consumer-grade resources while requiring less training data in order to achieve the same accuracy, thereby lowering the cost of training ASR models in other languages. Model introspection revealed that small adaptations to the network's weights were sufficient for good performance, especially for inner layers.

1 Introduction

Automated speech recognition (ASR) is the task of translating spoken language to text in real-time. Recently, end-to-end deep learning approaches have surpassed previously predominant solutions based on Hidden Markov Models. In an influential paper, Amodei et al. (2015) used convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to redefine the state of the art. However, Amodei et al. (2015) also highlighted the shortcomings of the deep learning approach. Performing forward and backward propagation on complex deep networks in a reasonable amount of time requires expensive specialized hardware. Additionally, in order to set the large number of parameters of a deep network properly, one needs to train on large amounts of audio recordings. Most of the time, the recordings need to be transcribed by hand. Such data in adequate quantities is currently available for few languages other than English.

We propose an approach combining two methodologies to address these shortcomings. Firstly, we use a simpler model with a lower resource footprint. Secondly, we apply a technique called *transfer learning* to significantly reduce the amount of non-English training data needed to achieve competitive accuracy in an ASR task. We investigate the efficacy of this approach on the specific example of adapting a CNN-based end-to-end model originally trained on English to recognize German speech. In particular, we freeze the parameters of its lower layers while retraining the upper layers on a German corpus which is smaller than its English counterpart.

We expect this approach to yield the following three improvements. Taking advantage of the representation learned by the English model will lead to shorter training times compared to training from scratch. Relatedly, the model trained using transfer learning requires less data for an equivalent score than a German-only model. Finally, the more layers we freeze the fewer layers we need to back-propagate through during training. Thus we expect to see a decrease in GPU memory usage since we do not have to maintain gradients for all layers.

This paper is structured as follows. Section 2 gives an overview of other transfer learning approaches to ASR tasks. Details about our implementation of the Wav2Letter model and how we trained it can be found in Section 3. The data we used and how we preprocessed it is described in Section 4. After a short introduction of the performed experiments in Section 5 we present and discuss the results in Section 6 followed by a conclusion in Section 7.

2 Related Work

Annotated speech data of sufficient quantity and quality to train end-to-end speech recognizers is scarce for most languages other than English. Nevertheless, there is demand for high-quality ASR

Proceedings of the 2nd Workshop on Representation Learning for NLP, pages 168–177,
Vancouver, Canada, August 3, 2017. ©2017 Association for Computational Linguistics

systems for these languages. Dealing with this issue requires specialized methods.

One such method, known as *transfer learning*, is a machine learning technique for enhancing a model's performance in a data-scarce domain by cross-training on data from other domains or tasks. There are several kinds of transfer learning. The predominant one being applied to ASR is *heterogeneous transfer learning* (Wang and Zheng, 2015) which involves training a base model on multiple languages (and tasks) simultaneously. While this achieves some competitive results (Chen and Mak, 2015; Knill et al., 2014), it still requires large amounts of data to yield robust improvements (Heigold et al., 2013).

In terms of how much data is needed for effective retraining, a much more promising type of transfer learning is called *model adaptation* (Wang and Zheng, 2015). With this technique, we first train a model on one (or more) languages, then retrain all or parts of it on another language which was unseen during the first training round. The parameters learned from the first language serve as a starting point, similar in effect to pre-training. Vu and Schultz (2013) applied this technique by first learning a multilayer perceptron (MLP) from multiple languages with relatively abundant data, such as English, and then getting competitive results on languages like Czech and Vietnamese, for which there is not as much data available.

The method presented in this paper differs from Vu and Schultz (2013) in that it does not force the representation to be compressed into *bottleneck features* (Grezl and Fousek, 2008) and use the result as the output of the pre-trained network. The idea of freezing only certain layers is another way in which our approach differs.

3 Model Architecture

One of the reasons Amodei et al. (2015) had to train their network using many GPUs was its complexity. It uses both convolutional and recurrent units stacked in many layers. Recently, a much simpler architecture called Wav2Letter has been proposed by Collobert et al. (2016). This model does not sacrifice accuracy for faster training. It relies entirely on its loss function to handle aligning the audio and the transcription sequences while the network itself consists only of convolutional units.

The resulting shorter training time and lower hardware requirements make Wav2Letter a solid basis for all of our transfer learning experiments. Since the general structure of the network is described in the original paper, we only mention what is notable in our adaptation of it in the following. An overview of their architecture is shown in Figure 1.

Collobert et al. (2016) do not specify the optimizer they used. We tried several conventional gradient descent optimizers and achieved best convergence with Adam (Kingma and Ba, 2014). Hyperparameters were slightly adapted from the defaults given by Kingma and Ba (2014), that is, we used the learning rate $\alpha = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Collobert et al. (2016) note that the choice of activation function for the inner convolution layers does not seem to matter. We chose rectified linear units as our activation function because they have been shown to work well for acoustic models (Maas et al., 2013). Weights are initialized Xavier uniformly as introduced by Glorot and Bengio (2010).

At test time, decoding is performed using a beam search algorithm based on KenLM (Heafield et al., 2013). The decoding procedure follows the TensorFlow implementation based on (Graves, 2012). A beam is scored using two hyperparameters that were derived using a local search optimized to yield the best combined word error rate (WER) and letter error rate (LER) on the LibriSpeech (Panayotov et al., 2015) validation set. For the weight of the language model we chose $w_{lm} = 0.8$ and a weight multiplied with the number of vocabulary words in the transcription $w_{valid\_word} = 2.3$.

The CNN was implemented in Keras (Chollet, 2015). The language model and beam search were done in TensorFlow (Abadi et al., 2015) and the introspection in NumPy (van der Walt et al., 2011). The source code can be found at: https://github.com/transfer-learning-asr/transfer-learning-asr.

One of the innovations in Collobert et al. (2016) was the introduction of the AutoSegCriterion (ASG) loss function. The authors reported it mainly improving the model's throughput with negligible effect on WER and LER compared to the Connectionist Temporal Classification (CTC) loss introduced by Graves et al. (2006). Since there is currently no publicly available implementation of this loss function, we decided to stay with an existing TensorFlow implementation of the CTC loss instead.

The English model achieved a LER of 13.66% and WER of 43.58% on the LibriSpeech (Panayotov et al., 2015) test-clean corpus. This is worse than the results of Collobert et al. (2016). Since the authors of that paper did not publish their source code, we

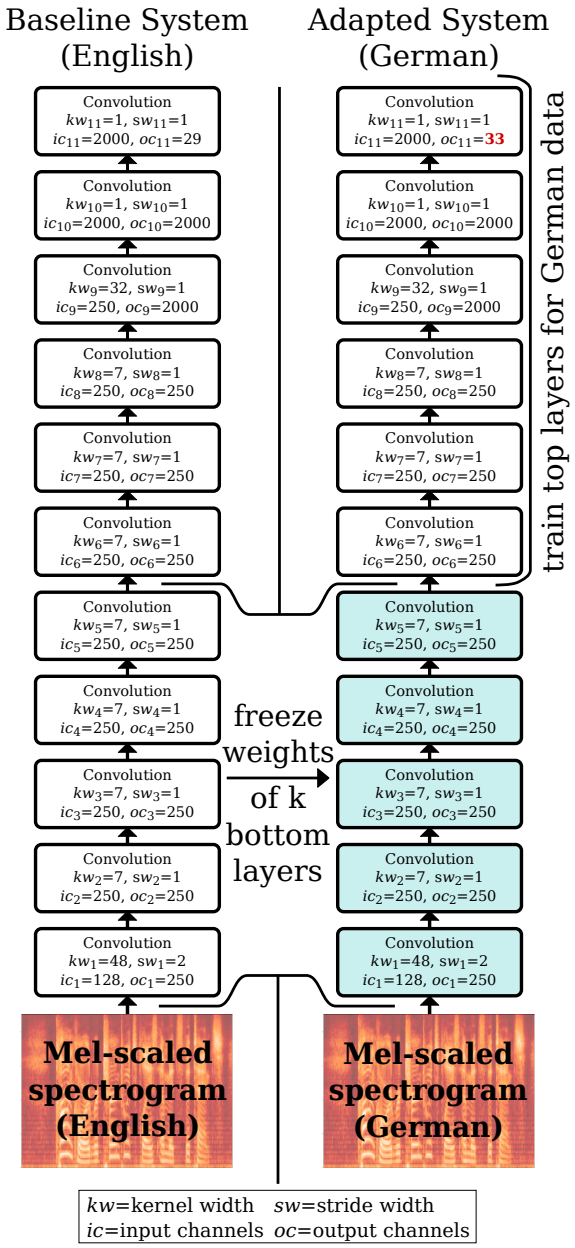

Figure 1: Network architecture adapted from Collobert et al. (2016).

were not able to reproduce their results reliably. All of our transfer learning experiments are based on this model and for our experiments it is assumed that such a model is already given for the transfer learning task that is to be performed.

4 Datasets

For training the English model, we used the LibriSpeech corpus (Panayotov et al., 2015). This dataset consists of about 1000 hours of read speech, sampled at 16 kHz, from the domain of audio books. This is the same dataset that was used to train the original Wav2Letter model.

The German models were trained on several corpora taken from the Bavarian Archive for Speech Signals (BAS) (Schiel, 1998; Reichel et al., 2016) as well as the dataset described in Radeck-Arneth et al. (2015), which will be referred to as "RADECK" from now on. Overall, we had a total of 383 hours of training data, which is only slightly more than one third of the English corpus. Additional quantitative information regarding each corpus, as well as any available references, is given in Table 1. Information about the kind of recording contained in each corpus is given in Table 2. It is also important to point out that some of the corpora pose additional challenges for speech recognition like partially intoxicated people, recordings over telephone, and different dialects.

Each German corpus was split into training and test sets. We grouped the audio by speakers and used 10% of the groups for testing. Therefore, no speaker appears in both training and test set ensuring that results are not due to overfitting to certain speakers. Exceptions to this procedure are: The VM corpora, which were used exclusively for training because obtaining a split based on speakers was not trivial here; SC10, which was used only for testing because it consists of recordings of speakers with German as a second language and strong foreign accents with only 5.8 hours in size; and RADECK, where we used the original splits.

We also rely on text corpora for the KenLM decoding step. For the English corpus (Panayotov et al., 2015), the provided 4-gram model based on all training transcriptions was used like in the original Wav2Letter implementation. For the German corpus, our n-gram model came from a preprocessed version of the German Wikipedia, the European Parliament Proceedings Parallel Corpus[1], and all the training transcriptions. Validation and test sets were carefully excluded.

4.1 Preprocessing

Since the English model was trained on data with a sampling rate of 16 kHz, the German speech data needed to be brought into the same format so that the convolutional filters could operate on the same timescale. To this end, all data was resampled to 16 kHz. Preprocessing was done using librosa (McFee et al., 2015) and consisted of applying a Short-time Fourier transform (STFT) to obtain power level spectrum features from the raw audio as described

[1] https://github.com/tudarmstadt-lt/kaldi-tuda-de/

Name	Size	Number of speakers	S LER	S WER	TL LER	TL WER
ALC (Schiel et al., 2012)	54.54h	162	13.48%	32.83%	8.23%	21.14%
HEMPEL (Draxler and Schiel, 2002)	14.21h	3909	34.05%	71.74%	19.13%	46.78%
PD1	19.36h	201	21.02%	34.37%	8.32%	11.85%
PD2	4.33h	16	7.60%	19.64%	1.97%	5.96%
RVG-J (Draxler and Schiel, 2002)	46.28h	182	17.43%	39.87%	10.85%	24.92%
SC10	5.80h	70	25.62%	78.82%	17.59%	57.84%
VM1 (Wahlster, 1993)	32.40h	654	-	-	-	-
VM2 (Wahlster, 1993)	43.90h	214	-	-	-	-
ZIPTEL (Draxler and Schiel, 2002)	12.96h	1957	22.87%	62.27%	15.07%	46.25%
RADECK (Radeck-Arneth et al., 2015)	181.96h	180	27.83%	65.13%	20.83%	56.17%
All corpora	415.7h	7545	22.78%	58.36%	15.05%	42.49%

Table 1: Quantitative information on the corpora used to train the German model. References to individual corpora are given where available. Size and number of speakers refer only to the subsets we used (including training and test sets). Test set LER and WER are reported for the best transfer learning (TL) model and the model from scratch (S) after 103h of training.

Name	Speech Type	Topic	Idiosyncrasies
ALC	read, spontaneous	car control commands, tongue twisters, answering questions	partially recorded in running car; speakers partially intoxicated
HEMPEL	spontaneous	answer: What did you do in the last hour?	recorded over telephone
PD1	read	phonetically balanced sentences, two stories: "Buttergeschichte" and "Nordwind und Sonne"	recordings were repeated until error-free
PD2	read	sentences from a train query task	recordings were repeated until error-free
RVG-J	read, spontaneous	numbers, phonetically balanced sentences, free-form responses to questions	speakers are adolescents mostly between the ages 13–15
SC10	read, spontaneous	phonetically balanced sentences, numbers, "Nordwind und Sonne", free dialogue, free retelling of "Der Enkel und der Grossvater"	multi-language corpus; only German data was used
VM1	spontaneous	dialogues for appointment scheduling	multi-language corpus; only German data was used
VM2	spontaneous	dialogues for appointment scheduling, travel planning and leisure time planning	multi-language corpus; only German data was used
ZIPTEL	read	street names, ZIP codes, telephone numbers, city names	recorded over telephone
RADECK	read, semi-spontaneous	Wikipedia, European Parliament transcriptions, commands for command-and-control settings	contains six microphone recordings of each speech signal

Table 2: Information on the kind of speech data contained in each corpus.

in Collobert et al. (2016). After that, spectrum features were mel-scaled and then directly fed into the CNN. Originally, the parameters were set to window length $w = 25\text{ms}$, stride $s = 10\text{ms}$ and number of components $n = 257$. We adapted the window length to $w_{new} = 32\text{ms}$ which equals a Fourier transform window of 512 samples, in order to follow the convention of using power-of-two window sizes. The stride was set to $s_{new} = 8\text{ms}$ in order to achieve 75% overlap of successive frames. We observed that $n = 257$ results in many of the components being 0 due to the limited window length. We therefore decreased the parameter to $n_{new} = 128$. After the generation of the spectrograms, we normalized them to mean 0 and standard deviation 1 per input sequence.

Any individual recordings in the German corpora longer than 35 seconds were removed due to GPU memory limitations. This could have been solved instead by splitting audio files using their word alignments where provided (and their corresponding transcriptions), but we chose not to do so since the loss of data incurred by simply ignoring overly long files was negligible. Corpora sizes given in Table 1 are after removal of said sequences. We excluded 1046 invalid samples in the RADECK corpus due to truncated audio as well as 569 samples with empty transcriptions.

5 Experiments

Given the English model, we froze k of the lower layers and trained all $11 - k$ layers above with the

German corpora. This means the gradient was only calculated for the weights of those $11-k$ layers and gradient descent was then applied to update those as usual. The process of freezing k layers is visualized in Figure 1. The transfer training was performed based on both the original weights as well as a new random initialization for comparison. Except for changing the training data, the German corpora introduce four new class labels *äöüß* in addition to the original 28 labels. We set the initial weights and biases of the final softmax layer for these labels to zero. Additionally, as a baseline for the performance of a Wav2Letter based German ASR, we trained one model from scratch on all German training corpora. For all experiments we used a batch size of 64, both during training as well as evaluation.

6 Results and Discussion

As initially hypothesized, transfer learning could give us three benefits: Reduced computing time, lower GPU memory requirements and a smaller required amount of German speech data. In addition to that, we may find structural similarities between languages for the ASR task. In the subsequent sections, we will first report general observations, evaluate each hypothesis based on the performed experiments and then analyze the learned weights using introspection techniques. We report overall test scores and scores on each test set in the form of WERs and LERs. Finally, we discuss the language specific assumptions that were required for the experiments and how transfer learning may perform on other languages.

6.1 Retaining or reinitializing weights?

When the transfer learning training is performed, one could either continue training on the existing weights or reinitialize them. Reusing existing weights might lead to stochastic gradient descent (SGD) being stuck in a local minimum, reinitializing may take longer to converge. For $k=8$ we compared the speed of training for both methods. As it can be seen in Figure 2, using existing weights is much faster without a decrease in quality.

6.2 Reduced computing time

Given that languages share common features in their pronunciation, lower layers should contain common features that can be reused when transferring the model to a different language. Therefore, we subsequently froze k layers of the original English model, choosing a different k in each experiment. Our

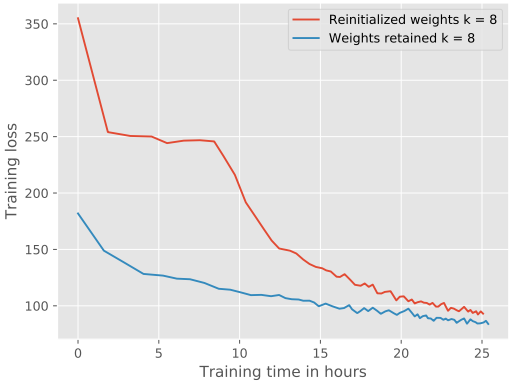

Figure 2: Comparison of learning curves for 25 hours of training with either reinitialized or retained weights. In both cases $k=8$ layers were frozen.

experiments showed that this assumption is indeed true, it is sufficient to adjust only at least two layers for achieving training losses below 100 after 25 hours. The loss curve for different k can be seen in Figure 3.

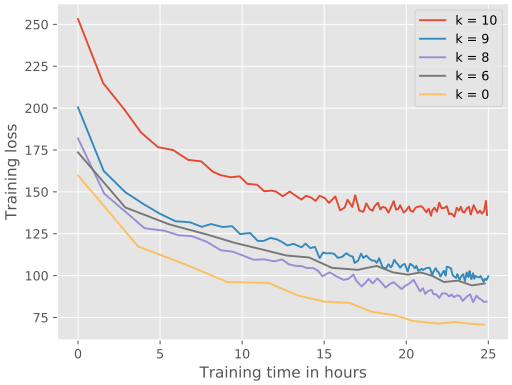

Figure 3: Learning curves for 25 hours of training with different numbers k of frozen layers. Note that due to the decreased time to process a batch (cf. Figure 4), training models with higher k (more frozen layers) allows to iterate over the training data more often in the same amount of time. But eventually, this does not help to beat the model with $k=0$ which is trained with the fewest dataset iterations but still at any time achieves the lowest loss.

For bigger k we need to backpropagate through fewer layers, therefore training time per step (training one batch) decreases almost monotonically with k in Figure 4. Despite that boost in training time, experiments show that loss is almost always smaller at any given point in time for smaller k. In Figure 3

this manifests in $k = 0$ always having the smallest training loss. We conclude that in terms of achieving small loss, there is no reason to favor big values for k, freezing layers is not necessary.

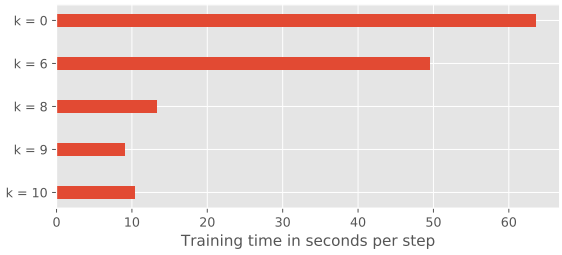

Figure 4: The more layers we freeze, the faster one batch of 64 is trained. Measured over 25h of training each.

When we compare the best transfer learning model with $k = 0$ with a German model trained from scratch in Figure 5, we are able to see huge improvements in terms of computing time required for achieving the same loss. We conclude that a good weight starting configuration from another language's ASR is beneficial.

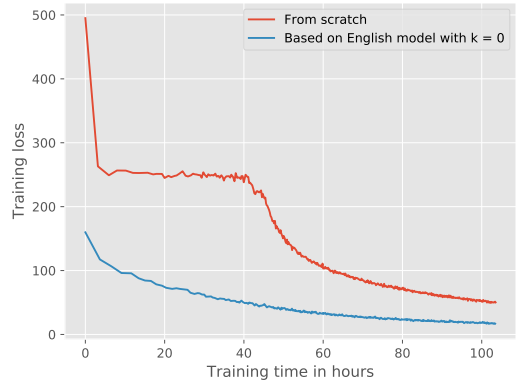

Figure 5: Applying transfer learning by using the weights from the English model leads to small losses more quickly than training from scratch.

6.3 Lower GPU memory requirements

Not only does it matter how long training takes with given resources, many researchers may also have only limited GPU memory at disposal. All of our experiments were performed on a single GeForce GTX Titan X graphics card, but the more layers k we freeze, the fewer layers we need to backpropagate through. Therefore, memory requirements for the GPU are lower. For a batch size of 64, forward propagation takes less than 3 GB of memory, while training the whole network requires more than 10.4 GB. In contrast to that, freezing 8 layers already enables training with less than 5.5 GB of GPU memory.

6.4 Little German speech data required

We hypothesized that little training data may be required for the transfer learning task. Additionally to using the whole 383 hours of data we had available, we also tried an even more scarce variant. In order to prevent overfitting, we used a transfer learning model with $k = 8$ for our experiments. As it can be seen in Figure 6, for a model with $k = 8$ that was trained for 25 hours, the LER using 100 hours of audio is almost equal to using the complete training data. Longer training causes overfitting. When using just 20 hours of training data this problem occurs even earlier. We can conclude that even though training for just 25 hours works well with only 100 hours of audio, beyond that overfitting appears nevertheless.

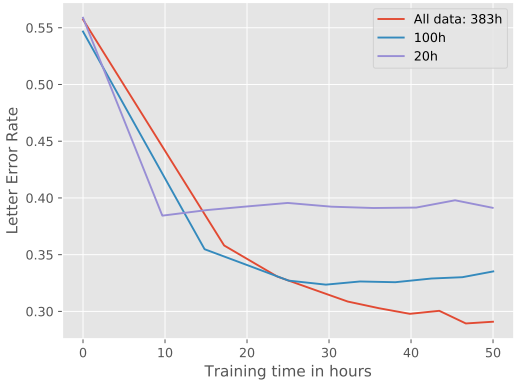

Figure 6: LER as a mean over all test samples for different training set sizes with k = 8 for all experiments

6.5 Model Introspection

When applying transfer learning, it is of interest how much the model needs to be adapted and which portions of the model are shared between different languages. To get insights into those differences, we compared the learned parameters both between the English model and adapted German model (for $k = 0$) as well as between different points in time during training. Since the output layers of both models do not use the same number of output features, we excluded this layer from the comparison. First, we investigated the distribution of weights and corresponding changes between the English and adapted model, visualized on the left side of Figure 7. The plot shows the fraction

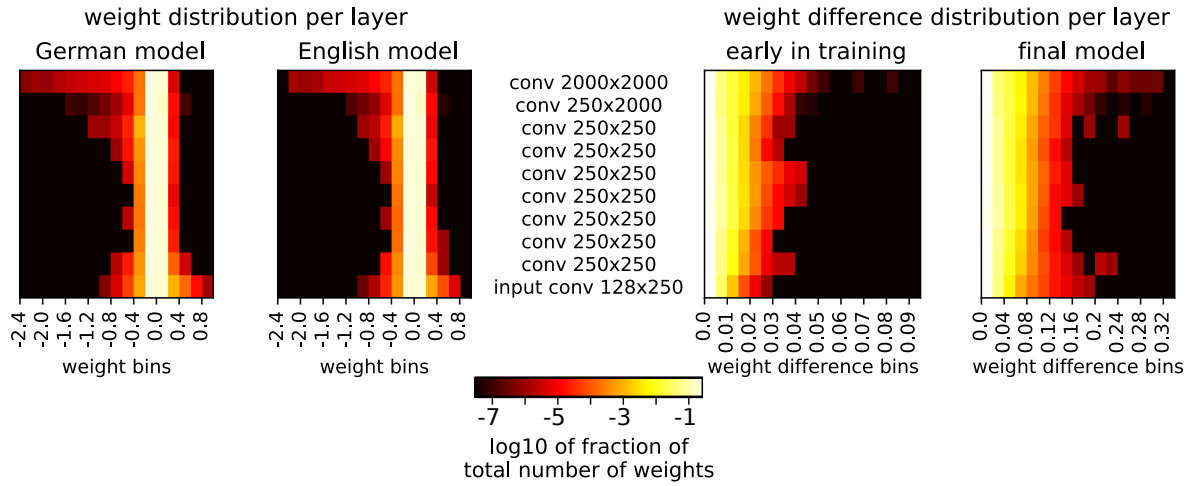

Figure 7: Weight distributions of the German and English model (left) and weight difference distributions both in an early stage and for the final model (right).

of weights in a layer lying in the respective range of values. Because most of the weights are between -0.2 and 0.2 (in just 2 bins), we used a $\log_{10}$-scale for the fraction of weights in each bin. We observed that the weights of highest absolute values are in the input and topmost layer. This indicates that the transformations in the middle layers are smaller than in the outer ones. Moreover, the weights of each layer are distributed with a mean value close to zero and very small variance. Due to the similar distributions, it is reasonable to compare the weights and their differences in the following. Between both models, there are only minor changes in the weight distributions, which supports the assumption that transfer learning is performing well because the English model is a suitable model for being adapted to German.

Since the adaptation to German is not explainable based on the distributions, we further investigated the differences between the individual weights. Therefore, we determined the absolute distance between weights as shown in Figure 7 on the right side. In the plot, we visualize the distribution of weight changes. We observed only small changes, therefore a $\log_{10}$-scale is used again. Figure 7 on the right side shows this analysis for the transfer learning model early in training as well as the final model after four days. In the early phase, weights had only been adapted little with a maximum difference of 0.1, while the final model weights changed up to 0.36. Additionally, we observed that the weights changed more in the middle and top layers earlier, but with progressing training the input layer experiences more changes. This higher variability in the outer layers can both be observed in the weights of each individual model as well as in their differences. That is an indication that the model needs to alter the outer layers more than the inner ones in order to adapt to a particular language.

Finally, we looked into the changes of individual filters. Due to the large number of neurons, we provide the complete set of filters from all layers only in the supplement.[2] We present our findings for a selected set of neurons of the input layer that showed well-interpretable patterns. The weights of those filters and their differences between the English and German model are shown in Figure 8. The top row shows neurons that can be interpreted as detectors for short percussive sounds (e.g. t or k) and the end of high pitched noise (e.g. s). The bottom neurons might detect rising or falling pitch in vowels. Of these four filters, the upper left differs most between English and German with a maximum difference of 0.15. This supports that it is detecting percussive sounds as German language has considerably stronger pronunciation of corresponding letters than English. On the other hand, the bottom row filters experienced less change (both < 0.1 maximum difference). This supports them being related to vocal detection since there are few differences in pronunciation between English and German speakers.

6.6 Overall test set accuracy

All test set LERs and WERs scores are consistent with the differences of loss in the performed experiments. After 103 hours of training, the best transfer learning model is therefore $k = 0$ with a LER of 15.05% and WER of 42.49% as the mean over all test samples.

[2] supplements: https://doi.org/10.6084/m9.figshare.5048965

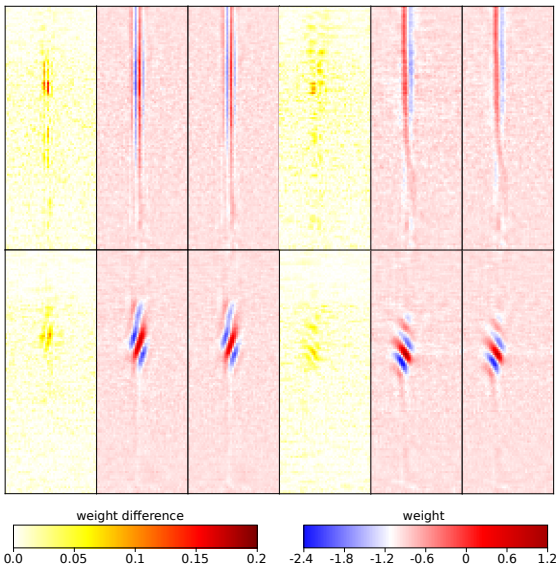

Figure 8: Differences in specific filters of the input layer. Neurons were chosen based on particular patterns. Each triplet of images shows the weight differences and the corresponding weights in the German and English model (from left to right).

The model that has been trained from scratch for the same amount of time achieves a LER of 22.78% and WER of 58.36%. Table 1 gives details about the accuracy on each test set.

Some very high WERs are due to heavy German dialect that is particularly problematic with numbers, e.g.

> Expected: "sechsundneunzig"
> Predicted: "sechs un nmeunsche"
> LER 47%, WER 300%, loss: 43.15

This shows, that there is both room for improvement in terms of word compounds as well as ASR of different dialects where data is even more scarce.

6.7 Accuracy boost through language model decoding

The original Wav2Letter network did not report on improvements in LER and WER due to the KenLM integration. In Table 3 We compared decoding performed through KenLM scored beam search with a greedy decoding on the German corpora.

6.8 Transfer learning for other languages

In our speech recognizer, the lower layers of the network learn phonological features whereas the higher (deeper) ones map these features onto graphemes. Thus for ASR these two types of features clearly matter the most. German and English have many phonemes and graphemes in common. The apparent success of our transfer learning approach was greatly facilitated by these similarities. Not all languages share as much in terms of these features. We anticipate that our approach will be less effective for such pairs. This means we expect the adaptation to a less similar language to require more data and training time. We further suspect that differences in grapheme inventories cause other effects than differences in phonemes. This is because only the mapping of phonological features to graphemes has to be adapted for a different orthography. In contrast, differences in phoneme inventories require more changes in features learned at lower layers of the network. Moreover, there could be differences in the importance of specific features. For instance, having vowels in common is potentially more important for transfer learning than sharing many consonants, because vowels experience higher variability in pronunciation. At the same time very drastic differences in orthography could probably trigger a stronger change of weights in lower network layers. We expect our transfer learning approach to encounter strong difficulties sharing knowledge between English and a logographic language like Mandarin Chinese. Despite those difficulties, using weights from a pre-trained ASR-network is a more reasonable initialization than random weights. This is because very basic audio features are shared between all languages. Therefore even for more different language pairs, we expect transfer learning to decrease the necessary amount of training data and time.

	LER	WER
with LM	15.05%	42.49%
without LM	16.77%	56.14%

Table 3: Comparing LER and WER with and without KenLM based on model with $k = 0$

7 Conclusions

We were able to show that transfer learning using model adaptation can improve the speed of learning when only 383 hours of training data are available. Given an English model, we trained a German model that outperforms the German baseline model trained from scratch in the same amount of training time. Thus, with little time, our approach allows training

better models. We showed that the English model's weights are a good starting configuration and allow the transfer learning model to reach smaller training losses in comparison to a weight reinitialization. When less GPU memory is available, freezing the lower 8 layers allows to train batches of 64 with less than 5.5 GB instead of more than 10.4 GB while still performing similar after 25 hours of training. Model introspection determined that lower and upper layers, in contrast to the layers in the center, need to change more thoroughly in order to adapt to the new language.

We identified several interesting directions for future work. Test accuracy showed that word compounds can be challenging and dialects pose difficulties when little training data is available. GPU memory consumption could be further reduced by caching the representation that needs only forward propagation. An open source version of the ASG loss would enable faster training. Finally, future research should investigate how well this transfer learning approach generalizes by applying it to more distinct languages.

Acknowledgments

This research was supported by the donation of a GeForce GTX Titan X graphics card from the NVIDIA Corporation.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. http://tensorflow.org/.

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. 2015. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR* abs/1512.02595. http://arxiv.org/abs/1512.02595.

Dongpeng Chen and Brian Kan-Wing Mak. 2015. Multitask learning of deep neural networks for low-resource speech recognition. *IEEE/ACM Trans. Audio, Speech & Language Processing* 23(7):1172–1183. http://dx.doi.org/10.1109/TASLP.2015.2422573.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system. *CoRR* abs/1609.03193. http://arxiv.org/abs/1609.03193.

Christoph Draxler and Florian Schiel. 2002. Three New Corpora at the Bavarian Archive for Speech Signals – and a First Step Towards Distributed Web-Based Recording. In *Third International Conference on Language Resources and Evaluation (LREC)*. Gonzles Rodriguez, Manual, pages 21–24.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. volume 9, pages 249–256.

Alex Graves. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer. http://dx.doi.org/10.1007/978-3-642-24797-2.

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*. ACM, pages 369–376.

Frantisek Grezl and Petr Fousek. 2008. Optimizing bottleneck features for lvcsr. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pages 4729–4732.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pages 690–696.

Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, M. Ranzato, Matthieu Devin, and Jeffrey Dean. 2013. Multilingual acoustic models using distributed deep neural networks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pages 8619–8623.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Kate Knill, Mark J. F. Gales, Anton Ragni, and Shakti P. Rath. 2014. Language independent and unsupervised acoustic models for speech recognition and keyword spotting. In *15th Annual Conference of the International Speech (INTERSPEECH) Communication Association, Singapore, September 14-18, 2014*. pages 16–20. http://www.isca-speech.org/archive/interspeech_2014/i14_0016.html.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*. volume 30.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th python in science conference*. pages 18–25.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. LibriSpeech: an ASR corpus based on public domain audio books. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pages 5206–5210.

Stephan Radeck-Arneth, Benjamin Milde, Arvid Lange, Evandro Gouvêa, Stefan Radomski, Max Mühlhäuser, and Chris Biemann. 2015. Open source german distant speech recognition: Corpus and acoustic model. In *International Conference on Text, Speech, and Dialogue*. Springer International Publishing, pages 480–488.

Uwe D. Reichel, Florian Schiel, Thomas Kisler, Christoph Draxler, and Nina Pörner. 2016. The BAS Speech Data Repository .

Florian Schiel. 1998. Speech and speech-related resources at BAS. In *Proceedings of the First International Conference on Language Resources and Evaluation*. pages 343–349.

Florian Schiel, Christian Heinrich, and Sabine Barfüsser. 2012. Alcohol language corpus: the first public corpus of alcoholized German speech. *Language resources and evaluation* 46(3):503–521.

Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. 2011. The numpy array: a structure for efficient numerical computation. *CoRR* abs/1102.1523. http://arxiv.org/abs/1102.1523.

Ngoc Thang Vu and Tanja Schultz. 2013. Multilingual multilayer perceptron for rapid language adaptation between and across language families. In Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier, editors, *INTERSPEECH*. ISCA, pages 515–519.

Wolfgang Wahlster. 1993. Verbmobil. In *Grundlagen und Anwendungen der Künstlichen Intelligenz*. Springer Berlin Heidelberg, pages 393–402.

Dong Wang and Thomas Fang Zheng. 2015. Transfer Learning for Speech and Language Processing. *arXiv:1511.06066 [cs]* http://arxiv.org/abs/1511.06066.