How transferable are features in convolutional neural network acoustic models across languages?

Jessica A.F. Thompson
BRAMS, Mila
University of Montreal, Canada
j.thompson@umontreal.ca

Marc Schönwiesner
Institut for Biology
University of Leipzig, Germany

Yoshua Bengio
Mila
University of Montreal, Canada

Daniel Willett
Nuance Communications
Aachen, Germany

Abstract

Characterization of the representations learned in intermediate layers of deep networks can provide valuable insight into the nature of a task and can guide the development of well-tailored learning strategies. Here we study convolutional neural network-based acoustic models in the context of automatic speech recognition. Adapting a method proposed by Yosinski et al. [2014], we measure the transferability of each layer between German and English to assess their language-specifity. We observe three distinct regions of transferability: (1) the first two layers are entirely transferable between languages, (2) layers 2–8 are also highly transferable but we find evidence of some language specificity, (3) the subsequent fully connected layers are more language specific but can be successfully finetuned to the target language. To further probe the effect of weight freezing, we performed follow-up experiments using freeze-training [Raghu et al., 2017]. Our results are consistent with the observation that CNNs converge 'bottom up' during training and demonstrate the benefit of freeze training, especially for transfer learning.

1 Introduction

The acoustic properties of speech vary across languages. This is evidenced by the fact that monolingual acoustic models (AMs) are the de facto standard in automatic speech recognition (ASR), while multi-lingual AMs are an active area of development [Heigold et al., 2013, Tuske et al., 2013, Sercu et al., 2016, Watanabe et al., 2017]. Requiring large amounts of training data to build separate AMs for every language is a barrier to successful ASR systems for low-resource languages. Ideally, AMs would be designed to strategically leverage off-task data as much as possible. AMs often take the form of a deep network which learns to map from acoustic features to context-dependent phones in a language-specific phone set. It is not clear how exactly this transformation is performed or what is represented in the intermediate layers of such networks. Better characterization of the intermediate representations of AMs may help to guide data-efficient training procedures. Similar characterizations of networks trained on visual tasks have inspired new transfer learning procedures. For example, Yosinski et al. [2014] characterized the task specificity at each layer of a network trained on ImageNet using transferability as a proxy for task-specificity. This characterization motivated Adaptive Transfer Networks [Long et al., 2015] where parts of a network are trained on the source domain while other parts of the network are finetuned, or adapted, to the target domain, preserving the limited target data for learning highly task-specific parameters. Similar adaptive transfer learning procedures may also prove to be useful for building AMs for data-poor languages. However, the exact shape of the transition from task-general to task-specific representations in deep network-based AMs is unknown.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Much of the previous work on characterizing intermediate layers of deep networks has focused on relatively solvable tasks in the visual domain (e.g. hand written digit recognition, visual object recognition). Few studies have characterized the intermediate representations of networks trained on acoustic tasks [Lee et al., 2009, Golik et al., 2015, Nagamine et al., 2015], which, in practice, are not always trained long enough to converge completely (test error still slowly decreasing at the end of training) due to the long training time required. It is not clear to what extent existing methods developed to probe networks trained on visual tasks will be applicable and useful to study networks that may be underfitting on difficult acoustic tasks.

Here we studied convolutional neural networks (CNNs) used for acoustic modeling in ASR systems. We characterized the language-specificity of each layer across languages using an approach inspired by Yosinski et al. [2014]. Subsets of a network trained on one language were “implanted” into another network which was trained on a second language. The effect of the implant on performance indicated the language-specificity of the features in the implant. Our main contribution is the characterization of the language-specificity of intermediate layers of CNN-based acoustic models. Additionally, we demonstrate the adaptation of an analysis method originally designed to probe visual networks to study networks in an underfitting regime on a phone classification task.

2 Experiments

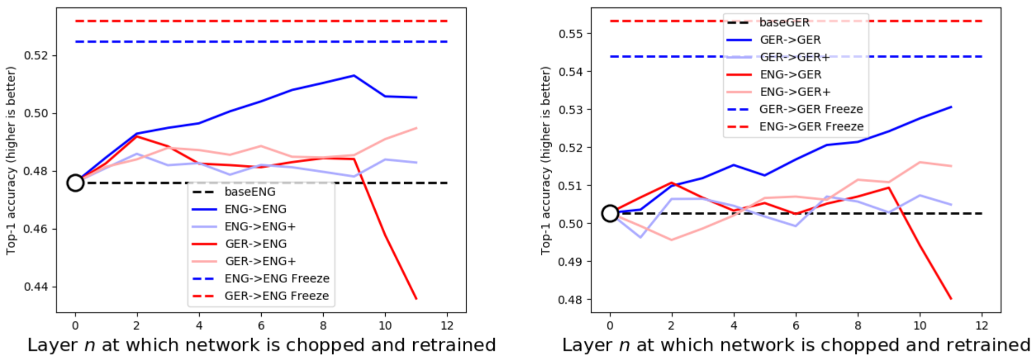
The datasets for this experiment consisted of 68 hours and 83 hours of German (GER) and American English (ENU) speech respectively, recorded in comparable environments, with corresponding text transcriptions. We chose these languages because we expected a large degree of transferability based on their phonetic similarity. Logarithmic Mel filter bank features were calculated, creating a 45-dimensional feature vector for every 10ms of audio (spectrograms). Each observation was associated with one of 9000 target context-dependent phone classes. Phonesets consisted of 54 unique phones for English and 47 unique phones for German.

A CNN consisting of nine convolutional layers followed by three fully connected layers was trained to recognize context-dependent phones from each language. The architecture was as follows, where the triplets specify the filter size and number of feature maps in each conv layer and the singletons specify how many units in each of the fully connected layers: (7, 7, 1024), (3, 3, 256), (3, 3, 256), (3, 3, 128), (3, 3, 128), (3, 3, 128), (3, 3, 64), (3, 3, 64), (3, 3, 64), (600), (190), (9000). This resulted in a total of approximately 7.2 million parameters. Both networks were trained using the ADAM optimizer [Kingma and Ba, 2015] as implemented in Tensorflow [Abadi et al., 2016] with a minibatch size of 256, a starting learning rate of $10e^{-}5$ and rectified linear units. Approximately 98% of the data was used for training and the remaining 2% for testing. All model parameters were replicated on four GPUs. Different minibatches were given to each GPU, their gradients averaged to calculate updates. As a balance between training time and accuracy, each network was trained for a fixed period of 100 epochs (which took approximately two weeks).

The subsequent experimental setup was similar to that described in Yosinski et al. [2014]. Several ‘network surgeries’ were performed. The first n layers of a network trained on Language A were implanted into a new network of identical architecture where the layers after layer n were randomly initialized. This ‘chimera’ network was further trained in four different ways. It was either trained on language A (self-transfer or ‘selfer’ network) or language B (transfer network) and the implanted parameters were either fixed or allowed to be finetuned during training. This process was repeated $\forall$ $1 \leq n \leq 11$ and for both English and German resulting in 88 networks total (see Figure 1 in Yosinski et al. [2014] for a graphical depiction of a similar experimental setup). The selfer networks served as a control to capture any changes in performance associated with the surgery but unrelated to the transfer. All networks were trained for 100 epochs. Training parameters were identical to those of the baseline models.

3 Results

We found representations throughout the networks to be highly transferable between English and German. Top-1 test phone classification accuracy for each network is plotted as a function of the layer at which the surgery was performed in Figure 1. Phone classification accuracy is measured with respect to per frame phone-labels established in a forced alignment. The only models that



(a) German to English

(b) English to German

Figure 1: Test accuracy as a function of depth after 100 epochs. The plus sign indicates that the implanted pretrained layers were finetuned. Dashed black line indicates the performance of the monolingual baseline model. Up to the ninth layer, layers trained on German could be used as they were (without finetuning) in a network whose subsequent layers were trained on English with no loss in performance compared to baseline. Selfer networks without finetuning show an improvement compared to baseline. Freeze trained transfer networks yielded the best overall performance. The pattern is the same for the reverse transfer (from English to German).

performed considerably worse than the monolingual baseline models were the transfer networks without finetuning whose surgery occurred at one of the fully connected layers (the penultimate two layers). Transfer networks cut at any of the convolutional layers performed as well as the monolingual baseline model, regardless of whether the implanted layers were finetuned or not. We observed a slight improvement over the monolingual baseline (~1.5 percentage points (pp)) for transfer networks with finetuning chopped at one of the fully connected layers. All selfer networks with finetuning performed at the same level as the mono-lingual baseline. Somewhat unexpectedly, the selfer networks without finetuning performed best overall among the chimera networks. Selfer networks chopped at late layers whose implants were not finetuned showed an improvement of 3+ pp. Previous work has shown that random, untrained weights can often perform remarkably well in certain scenarios [Jarrett et al., 2009, Rahimi and Recht, 2007]. Figure **??** shows accuracy as a function of the layer at which training began, meaning that layers below layer n were randomly initialized and never updated. Subsequent layers (layer n and above) were trained for 100 epochs with the same training parameters as the baseline models. We observed a gradual drop in performance as a function of depth. Random weights in early layers did not have a large impact on performance. Using random weights for all but the last layer resulted in near-chance performance. This verifies the non-triviality of the success of our transfer networks without finetuning. The training of our selfer

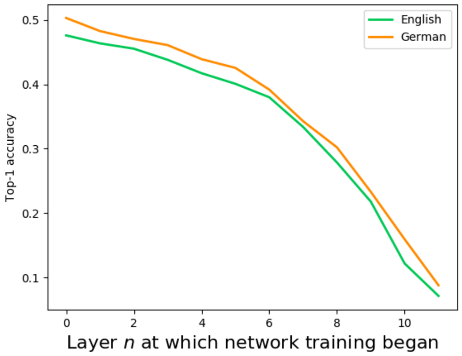


Figure 2: Using random weights up to layer n. The leftmost points represent the baseline models. Performance decays gradually as more layers are left untrained, only reaching near-chance performance when nearly all layers are random.

networks without finetuning somewhat resembles the *freeze training* procedure proposed by Raghu et al. [2017]. According to this procedure, layers are successively frozen over the course of training, gradually reducing the number of parameters to be updated until, by the end of training, only the last layer is being updated. We hypothesized that weight freezing partly explained the success of our selfer networks without finetuning, so we created freeze trained versions of both our selfer and transfer networks. Starting with a pre-trained network, layers 1–11 (excluding layer 0) were trained for 5 epochs. Then, for the next 5 epochs, only layers 2–11 were trained. From then on, another layer was removed from the trainable parameters every 10 epochs for a total of 100 training epochs. The freeze trained models are represented by the coloured dashed lines in Figure 1. The freeze training led to better overall performance, especially among the transfer networks.

4 Discussion

Our results suggest that, despite a large degree of transferability of intermediate acoustic features between languages, naive approaches to transfer (e.g. initializing with parameters from another language) are not the most efficient. In particular, early layers need not be finetuned on the target language at all. Subsequent layers benefit greatly from freeze training on the target language. These freeze trained transfer networks outperform networks trained solely on the target language, which demonstrates the improved generalization that can be achieved when incorporating data from multiple sources.

The performance of the networks with finetuning is largely consistent with Yosinski et al. [2014]. However, the performance of networks without finetuning deviates considerably. The transfer networks without finetuning in Yosinski et al. [2014] show a gradual drop in performance, starting at the 4th convolutional layer and eventually dropping nearly 8 pp by the penultimate layer (see Figure 2 from Yosinski et al. [2014]). Our transfer networks without finetuning, on the other hand, show a sharp drop in performance that starts only at the first fully connected layer (layer 9). For the selfer networks without finetuning, we did not observe a performance drop when networks were chopped at middle layers, as was reported in Yosinski et al. [2014]. Instead, our selfer networks without finetuning outperformed all other models, with accuracy increasing nearly monotonically with the depth at which the network was chopped. Yosinski et al. [2014]'s experiments with random weights quickly drop to near-chance performance by layer 3, whereas our networks with random weights decline gradually with depth, only approaching near-chance performance when all but the last layer are random.

The success of our selfer networks without finetuning is at least partly explained by the fact that we are in an underfitting regime. Unlike in Yosinski et al. [2014], our baseline model has not converged completely and we would expect continued training to improve performance. However, if that were the only factor at play, then we would expect our selfer networks with finetuning to also improve but they do not. Something about freezing all but the last layer(s) facilitates a ~3 pp improvement over baseline in the selfer but not the transfer networks. This suggests that there is some important language-specific information in the layers that show a difference between the selfer and transfer networks without finetuning (layer 3+). Layers 10 and 11 show worse than baseline performance for the transfer network without finetuning, indicating a larger degree of language-specificity in these representations.

Our freeze training results corroborate the interpretation that weight freezing is responsible for the success of our selfer networks without finetuning. Furthermore, our freeze-trained transfer networks performed best overall, demonstrating that freeze training can actually recover the language-specific information lacking in our transfer networks without finetuning, yielding improved generalization. This likely reflects the observation from Raghu et al. [2017] that CNNs converge 'bottom-up' during training, with early layers stabilizing earlier in training. Relatedly, Alain and Bengio [2016] state the proposition that no intermediate layer of a multi-layer neural network will contain more target-related information than the raw input, which requires a 'bottom-up' flow of information; intermediate layers cannot pass on target-related information that they do not receive. Thus we conclude that freezing the weights of a given layer can improve performance iff that layer already passes on the target-related information in a representation that can be disentangled by subsequent layers. This was not generally the case in our transfer chimera networks because important language-specific information was not being conveyed. The progressive freeze training regime, proposed by Raghu et al. [2017], allowed this important language-specific information to be learned, whereas generic fine-tuning did not. In this way, making fewer parameter updates actually led to significant performance gains. This may be partly explained by the fact that smaller networks train faster [Saxe et al., 2015]. Perhaps generic fine-tuning would eventually achieve the same accuracy, but after many more iterations.

References

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27*, 2014.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Understanding and Improvement. *Advances in Neural Information Processing Systems*, 2017.

Georg Heigold, Vincent Vanhoucke, Andrew Senior, Patrick Nguyen, M Ranzato, M Devin, and J Dean. Multilingual Acoustic Models using Distributed Deep Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8619–8623, 2013.

Zoltan Tuske, Joel Pinto, Daniel Willett, and Ralf Schluter. Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7349–7353, 2013.

Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun. Very Deep Multilingual Convolutional Neural Networks for LVCSR. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

Shinji Watanabe, Takaaki Hori, and John R . Hershey. LANGUAGE INDEPENDENT END-TO-END ARCHITECTURE FOR JOINT LANGUAGE IDENTIFICATION AND SPEECH RECOGNITION. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 265–271, 2017.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning Transferable Features with Deep Adaptation Networks. In *International Conference on Machine Learning*, volume 37, 2015.

Honglak Lee, Peter Pham, Y Largman, and AY Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*, 2009.

Pavel Golik, Zoltan Tuske, Ralf Schluter, and Hermann Ney. Convolutional Neural Networks for Acoustic Modeling of Raw Time Signal in LVCSR. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 26–30, 2015.

Tasha Nagamine, Michael L Seltzer, and Nima Mesgarani. Exploring How Deep Neural Networks Form Phonemic Categories. *Interspeech*, pages 1912–1916, 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations (ICLR)*, 2015.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, and Google Brain. TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 2016.

K Jarrett, K Kavukcuoglu, M Ranzato, and Y LeCun. What is the best multi-stage architecture for object recognition? In *IEEE 12th International Conference on Computer Vision (ICCV)*, pages 2146–2153, 2009.

Ali Rahimi and Ben Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 2007.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv*, page 1610.01644v3, 2016.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference for Learning Representations (ICLR)*, 2015.