Neuron Activation Profiles for Interpreting Convolutional Speech Recognition Models

Andreas Krug[1], **René Knaebel**[1] and **Sebastian Stober**[2]
[1]Research Focus Cognitive Sciences, University of Potsdam
[2]Artificial Intelligence Lab, Otto-von-Guericke-University Magdeburg
{ankrug, rknaebel}@uni-potsdam.de, stober@ovgu.de

Abstract

The increasing complexity of deep Artificial Neural Networks (ANNs) allows to solve complex tasks in various applications. This comes with less understanding of decision processes in ANNs. Therefore, introspection techniques have been proposed to interpret how the network accomplishes its task. Those methods mostly visualize their results in the input domain and often only process single samples. For images, highlighting important features or creating inputs which activate certain neurons is intuitively interpretable. The same introspection for speech is much harder to interpret. In this paper, we propose an alternative method which analyzes neuron activations for whole data sets. Its generality allows application to complex data like speech. We introduce time-independent Neuron Activation Profiles (NAPs) as characteristic network responses to certain groups of inputs. By clustering those time-independent NAPs, we reveal that layers are specific to certain groups. We demonstrate our method for a fully-convolutional speech recognizer. There, we investigate whether phonemes are implicitly learned as an intermediate representation for predicting graphemes. We show that our method reveals, which layers encode phonemes and graphemes and that similarities between phonetic categories are reflected in the clustering of time-independent NAPs.

1 Introduction

Artificial Neural Networks (ANNs) have proven to be highly successful in various fields of application. The success of Deep Learning (DL) is related to the implementation of ANNs with deeper or wider architectures [30]. This allows the model to learn complex patterns for solving a particular task. Such pattern detection performs particularly well in computer vision, for example in image classification [12] or segmentation [5]. Besides computer vision, DL is successful for audio processing tasks, including machine translation [33] or speech recognition [4].

The growing complexity of DL models complicates the understanding of how the network accomplishes its task [35]. To interpret those decision processes, several introspection methods have been proposed [28, 36, 27]. Most of those techniques have been developed for the computer vision domain. This is due to the intuitive interpretation of introspection results by a human observer. In speech related tasks like Automatic Speech Recognition (ASR), evaluating the results is not as intuitive. Therefore, visualization of input patterns which are typical for certain predicted graphemes has been proposed [13]. However, visualizing those patterns gives only little insight to the actual decision processes within the network.

Assessing whether a speech-related ANN learns reasonable representations in its layers remains a hard challenge. The intermediate representations of speech have been researched, but only for predicting acoustic features using simple architectures [20, 21, 19, 16]. Those methods are not applicable for analyzing more complex networks for ASR. For representing speech as accurately and robust as possible, words can be split into individual acoustic components, their phonemes [6]. This leads to the question whether an ANN for end-to-end speech-to-text prediction is learning its task in a similar way.

In this work, we propose a method to assess intermediate representations and demonstrate it for an ANN which predicts graphemes from speech. To this end, we introduce Neuron Activation Profiles (NAPs) to characterize neuron responses to groups of input examples in convolutional networks. Moreover, we adapt a method for visualizing representations by Nagamine et al. [20] using our NAPs so it can be applied to Convolutional Neural Networks (CNNs). With this adapted method, we investigate to which extent an ANN for ASR learns phonemes as an intermediate representation to predict graphemes. We show that our method reveals that phonemes are encoded in earlier layers than graphemes. Additionally, we demonstrate that similarity between phonetic categories is reflected in clustering of NAPs for phonemes and graphemes.

2 Background

2.1 Automatic Speech Recognition

Application of CNNs for speech is not uncommon, but often as part of hybrid models. Such models can for example involve Hidden Markov Models [1] or Recurrrent Neural Networks (RNNs) [32]. Due to their complexity, understanding their decision processes is much harder than those of fully-convolutional networks. Besides ASR, CNNs are also used for other speech-related tasks, like learning spectrum feature representations [8].

For ASR, we implement an architecture based on Wav2Letter [7]. This model is a 11-layer fully-1D-convolutional neural network, which predicts graphemes from spectrograms. We follow training and network design by Kunze et al., described in detail in [14]. The network is trained on z-normalized spectrograms, scaled to 128 mel-frequency bins. The training data are whole-sequence audio recordings from the LibriSpeech corpus [23]. Each grapheme prediction can use 206 time steps (a time frame of 2.1 s) due to the receptive field of the convolutions. While the acoustic model outputs one grapheme per frame, the sequence of predicted letters is decoded using a Connectionist Temporal Classification (CTC) beam search decoder [11]. Different to Kunze et al. we slightly changed the number of neurons per layer to powers of two (250 to 256 neurons and 2000 to 2048 neurons). Moreover, we used a vocabulary with repetition characters like Collobert et al. [7] used with their Auto Segmentation Criterion (ASG) loss.

2.2 Introspection

Currently there are two major categories of introspection techniques. Firstly, local introspection methods are tracing back, which parts of the input caused an activation of a particular neuron [28, 36, 27, 10]. This is usually performed for the output neuron, to reveal prediction-relevant regions. Local introspection only analyzes single examples, making it hard to draw conclusions about the decision processes in general. The second category is global introspection, which performs introspection without using a specific input. This mostly is done by learning an input which maximally activates certain parts of the network [9, 18, 35]. It is common to optimize for neurons of intermediate layers or even whole layers, as in Google DeepDream [18].

The aforementioned introspection techniques are visualizing network responses by back-projecting them onto the input layer. This makes sense for applications in computer vision, as a human observer can easily interpret images. In speech recognition, evaluating artificial optimal audio or relevant frequencies in the input requires expert knowledge or can even be impossible. Therefore, to analyze networks for speech recognition, it is reasonable to analyze neuron activations themselves instead of projecting information onto the input.

A common method to analyze the representational power of layers in an ANN is to decode classes directly from intermediate representations using linear classifiers [2]. However, this method can not identify similarities between representations of different classes. For example, in speech recognition, it is not possible to identify whether similar phonetic categories are reflected in similar network responses. Another option is to perform statistical analyses on neuron activations. For example, recent research applied Canonical Correlation Analysis (CCA) to analyze and compare representations in ANNs [17]. In speech, previous research investigated representations in acoustic modelling using Multi-Layer Perceptrons (MLPs) [20, 21, 19] or Deep Belief Nets [16]. Nagamine et al. trained an MLP for predicting phonemes from speech and analyzed whether similarity of neuron activations corresponds to similar phonetic categories in the input [20]. They performed clustering on average neuron responses over phonemes and showed that phonetic categories are reflected in the neuron activations. In later publications, they extended this analysis to sigmoid and Rectified Linear Unit (ReLU) units and more analysis methods like multidimensional scaling of softmax layer outputs or sample-dependent linear transform [21, 19]. However, those studies in speech only use MLPs and predict phonemes. Recent models for ASR use more complex architectures [37] and some predict graphemes in an end-to-end manner [7].

2.3 Grapheme-to-Phoneme translation

Grapheme-to-Phoneme (G2P) models are important for Natural Language Processing (NLP), as they are often used in ASR and text-to-speech (TTS) systems. Phonemes are an abstract representation of speech sounds, which describe pronunciation of words. Therefore, instead of mapping speech directly to text, it is simpler to use phonemes as an intermediate step. Data of speech and the corresponding text is often available, yet the phonetic representation is not. We use a network trained on the LibriSpeech corpus, which is not providing phonetic representations. Therefore, we need to obtain phonemes for the data set. The problem of obtaining the phonemes can be modelled as a translation problem, where the graphemes represent the input language and phonemes are considered the output language.

Words have a specific pronunciation. However, there are exceptions like heteronymes, where words are spelled identical and pronounced differently. The idea of the International Pronunciation Alphabet (IPA) is to provide a table of symbols that allows to generally describe words of any kind of language. Smaller versions of the IPA exist for specific languages, removing unused sounds and stresses, such as the CMU Pronunciation Dictionary (CMUDict) for the English language. The CMUDict [15] is an open source pronunciation dictionary developed at Carnegie Mellon University [1]. Currently, version 0.7b contains over 134.000 entries.

Recent research uses neural models for G2P, especially Long Short-Term Memory (LSTM) architectures [34, 25, 29], which outperform former statistical models. For our task, we adopt the approach of Toshniwal & Livescu [31]. We train this model on the CMUDict and generate a phonetic transcription of our output texts. The model is designed as an attention-based encoder-decoder model to generate phoneme representations for the whole data set. Two layers of bidirectional LSTMs form the encoder. The decoder uses a global attention mechanism [3] and further consists of two unidirectional LSTMs.

3 Experiments

There is a large variety of spectrograms which are predicted as the same grapheme. This is due to different pronunciations of the same grapheme, for example dependent on the speaker and the word it is included in. Moreover, the pronunciation of a word given its written form is highly idiosyncratic. For example, compare how the grapheme "W" is articulated in the word "where" in contrast to the word "who". Phonemes also show variability across speakers but are not dependent on other phonemes in their context. Therefore, the representation of a phoneme learned by an ANN should be less complex than that of a grapheme. Moreover, we hypothesize that for predicting grapheme for a given speech sample, it is necessary to implicitly learn a representation of phonemes as well.

For an exemplary speech recognizer, which predicts graphemes from spectrograms, we investigate whether it learned phonemes as an intermediate representation. To this end, we performed a comprehensive analysis of neuron activations upon processing data related to certain graphemes or phonemes, respectively.

Our analysis is inspired by Nagamine et al. [20]. They introduced hierarchical clustering of class-averaged neuron activations in an MLP for acoustic modeling. However, we needed to adapt their approach significantly, due to major differences in the model and scientific question. Firstly, our model uses a 1D-convolutional architecture to apply it to larger inputs, like sentences. The obtained neuron responses are not single values, but vectors of activations. Therefore, we introduce a novel method to obtain and compare neuron responses. Secondly, instead of phonemes, our model predicts

[1] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

text from speech in an end-to-end manner. This allows us to investigate whether the speech recognizer learns phonemes as intermediate representations.

Our analysis consists of three main steps. Firstly, we compute NAPs, which characterize the network response across many different instances of the same group. Secondly, we compare these characteristic profiles between classes in different layers using hierarchical clustering. This allows to determine, how specific certain layer's neurons are in response to a particular group. Finally, we contrast the findings for graphemes with those for phonemes to determine which layers are specific to each of them, respectively.

3.1 Time-independent Neuron Activation Profiles

We compute time-independent Neuron Activation Profiles (NAPs) to obtain characteristic responses for particular groups. A group is not restricted to prediction classes, but can be any grouping of inputs. The process is visualized exemplary for one group and one layer in Figure 1.

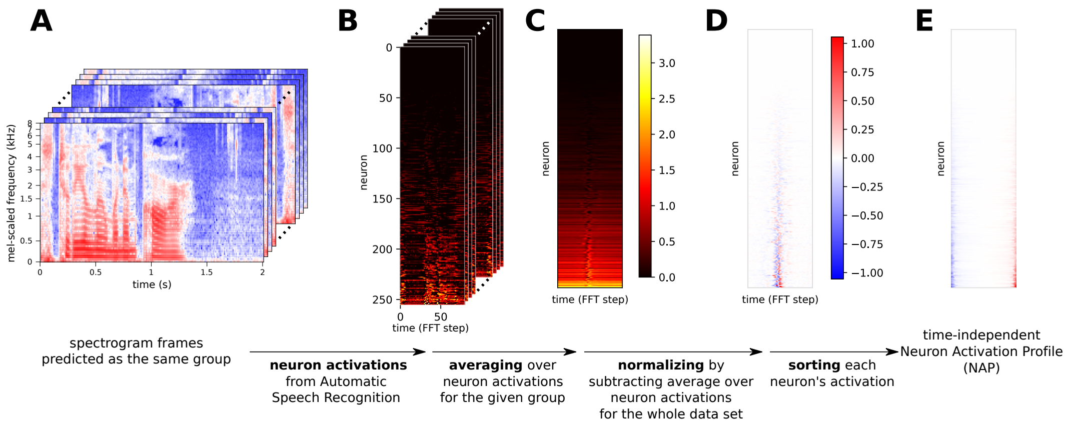

A
B
C
D
E
mel-scaled frequency (kHz)
time (s)
neuron
time (FFT step)
spectrogram frames predicted as the same group
neuron activations from Automatic Speech Recognition
averaging over neuron activations for the given group
normalizing by subtracting average over neuron activations for the whole data set
sorting each neuron's activation
time-independent Neuron Activation Profile (NAP)

Figure 1: Computation of time-independent Neuron Activation Profiles (NAPs). The procedure is exemplary shown for one group of inputs (predicted as grapheme 'A') and one layer (the first one). *A*: All spectrogram frames belonging to the same group (in this case grapheme 'A') are used as input. *B*: Data is processed with an ASR network and neuron activations are stored. *C*: Neuron activations are averaged over all instances of the same group. *D*: Normalization is performed by subtracting the average neuron activation over the whole data set. *E*: By sorting each neuron's activation, the profiles become independent of the time of activation.

As inputs, we use spectrogram frames of 206 time steps (the size of the receptive field), as exactly one grapheme is predicted without adding any padding (1A). Each of these frames is fed to the ANN and neuron activations are stored for each layer (1B). Due to the convolutional architecture, we obtain two-dimensional activation patterns in all layers. To obtain characteristic neuron responses for the given group, we compute the point-wise average over all activations of this group (1C). This averaging process reduces the activation to those which are common across most of the group examples. Even after averaging, some neurons have a high baseline activation, which is not specific to any group. Probably, those are activated for natural speech in general. Those dominant baseline activations can hide group-specific information. In order to avoid this, we normalize the activations (1D). We perform this normalization by subtracting the average activation over the inputs for all groups. The normalized activation then indicates increased or decreased response for a particular group compared to the overall mean activation. By using convolutions, features can be detected independent of their location. However, the exact position of detected features is not of interest for evaluation of group-specificity. Moreover, comparing activation strengths of different profiles would be hard, as the active positions are not aligned. Therefore, we sort each neuron's normalized activations by their value (1E). With this sorting, we obtain a time-independent NAP. We use these to characterize the network response to spectrogram frames which correspond to certain graphemes or phonemes.

3.2 Mapping phonemes to acoustic model output

To compare time-independent NAPs between graphemes and phonemes, we need the phonetic representation of the words predicted by our speech recognizer. This process includes several steps, which are visualized in Figure 2.

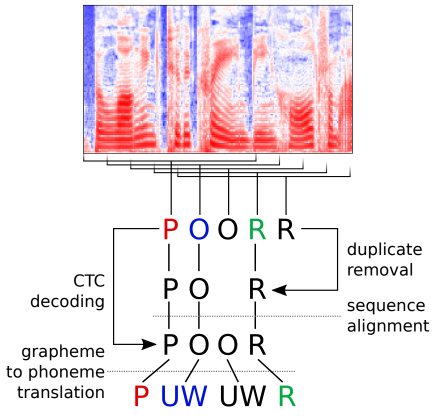


Figure 2: Assigning phoneme labels to spectrogram frames. In the acoustic model, spectrogram frames are associated to a single grapheme prediction. The phoneme representation was derived for the CTC decoded output. To map phonemes to spectrograms, we apply the visualized cascade of mappings.

The acoustic model outputs a grapheme for every time step. This sequence of graphemes is translated to words using a CTC decoder. The decoding does not retain a mapping to the original sequence. Therefore, we remove duplicates from the acoustic model output and globally align it to the decoder output using the Needleman-Wunsch algorithm [22] (match $= 1$, mismatch $= -1$, gap $= -1$).

We transform the CTC decoder output to phonemes with the G2P model described in Section 2.3. Training of the model was performed on the CMUDict data, from which we select 125,515 entries of words and abbreviations with multiple pronunciations. To obtain a single pronunciation per word, we split entries with multiple pronunciations. For example, the word `THE` can be pronounced as `/DH AH/` or `/DH IY/`. We split this into two word-to-phoneme training examples `(THE, /DH AH/)` and `(THE, /DH IY/)`. Thus, we obtain 134,093 samples and leave 20 % of the data for testing. For evaluation of the G2P translation model, we measure word error rate (WER) and BLEU score [24] to select the best performing model. Our model for phoneme prediction achieves 0.29 WER and 0.78 BLEU score. This is around 0.05 WER worse than the results of Toshnival & Levescu [31]. However, many errors are averaged out when creating NAPs for phonemes. Therefore, our analysis is robust to small changes in G2P prediction quality.

Because grapheme and phoneme sequences might have different lengths, they are hard to compare. In addition to the prediction, we want to assign every grapheme a phoneme. We use the model's attention for this assignment. Each grapheme is replaced by the phoneme with the highest attentional activation at this time step.

3.3 Comparison of NAPs across phonemes and graphemes

To compare the network responses to phonemes and graphemes, we use clustering of their time-independent NAPs (described in Section 3.1). We use hierarchical clustering with Euclidean distance and complete linkage. The distance threshold for forming clusters was set to the 80th percentile of distances in each layer, respectively. We apply this clustering to the set of graphemes and phonemes separately. Figures 3 and 4 show the result of the clustering for different layers. Each plot shows the distance matrix, which the clustering is computed from. Rows and columns are ordered by a dendrogram resulting from the hierarchical clustering. The dendrogram on each plot's left side shows the similarities between different groups. Subtrees of equal color indicate the clusters that were formed using the 80th percentile of distances.

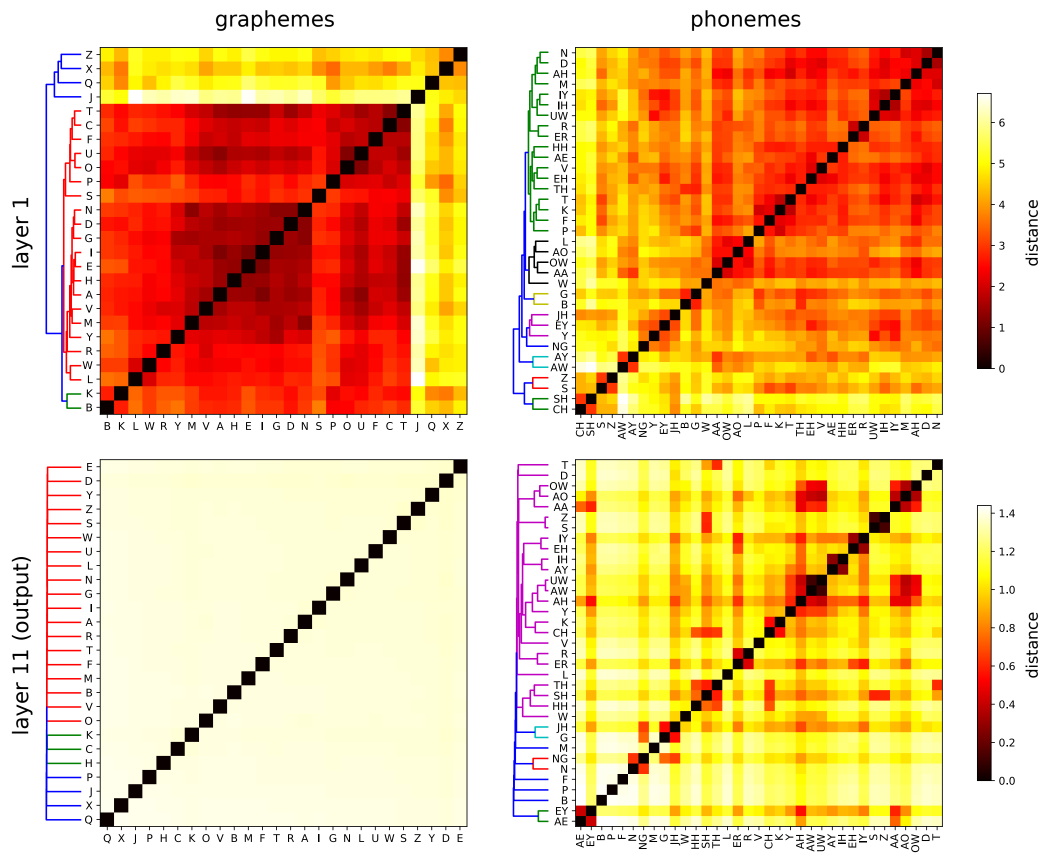


Figure 3: Activation profile clustering in the first and output layer. Clustering was performed between time-independent NAPs of graphemes (*left*) and phonemes (*right*). The first layer (*top*) shows no interpretable clustering, as those neurons only detect very basic features. Graphemes are predicted in the output layer (*bottom*). Therefore, each NAP for graphemes can be perfectly distinguished from each other. Clustering of phonemes shows similarities, if they are associated to the same grapheme.

First and output layer The first and output layer are trivial cases where we first demonstrate our method of comparison (Figure 3). A single convolutional layer can only detect simple features in the input data. Therefore, the first layer is not expected to reflect grapheme or phoneme similarities in its activations. Our results confirm this expectation (Figure 3 top), as there is no meaningful clustering for graphemes or phonemes. However, we can already observe similar activations for some very similar sets of graphemes and phonemes, respectively. For example, the graphemes 'U' and 'O' show small distance between activation profiles, as well as the phonemes 'AA', 'AO' and 'OW'. The output layer (Figure 3 bottom) for graphemes shows the trivial result that each grapheme prediction is independent with respect to their activations. This is obvious, as the grapheme groups were obtained from this layer's output. The corresponding clustering for phonemes shows a similar result, but phonemes which often are associated with the same grapheme also show high similarity. For example, phonemes 'N' and 'NG' are very similar in the output layer, as they are both associated with the prediction of 'N'.

Phoneme-encoding layers We hypothesized better encoding of phonemes in lower layers than encoding of graphemes. With our method, we could observe this in layers 9 and 10, as shown in Figure 4. We picked those layers manually, because in earlier layer than the 9th one, emerged clusters did not reflect group similarities, neither for graphemes nor for phonemes. In the appendix, NAP clustering results are shown for all layers. We highlighted clusters obtained from applying the aforementioned 80th percentile distance threshold.

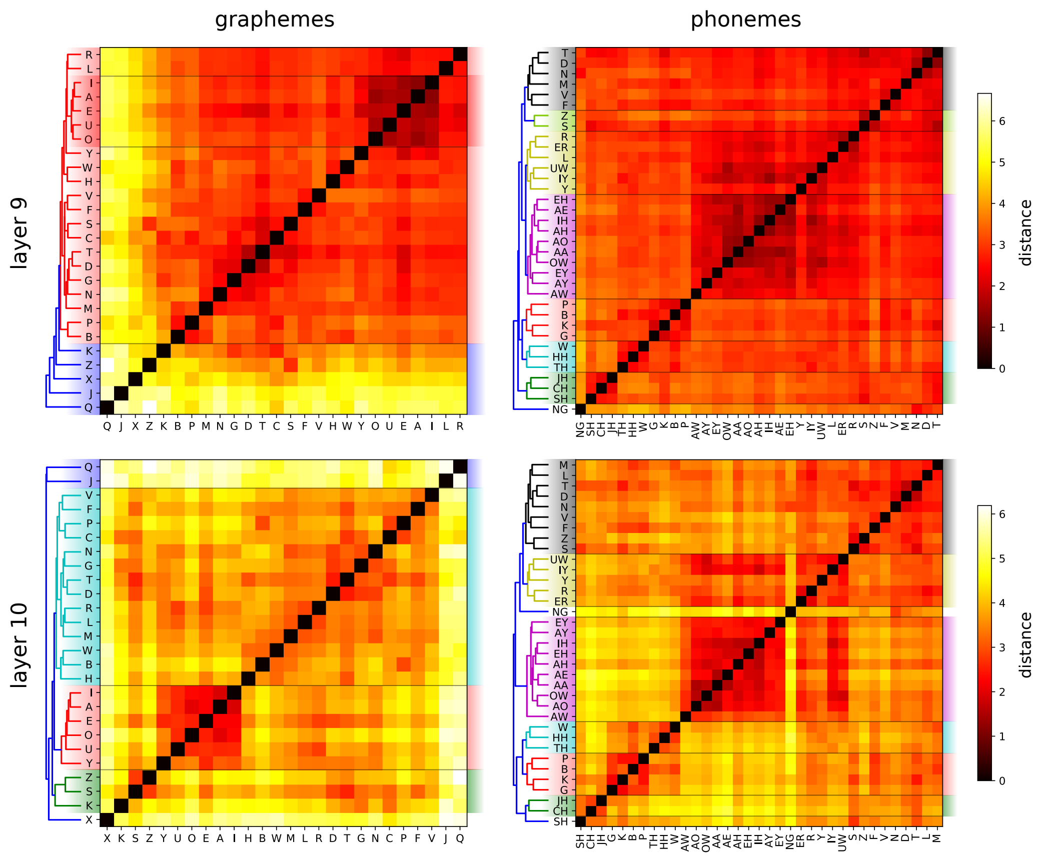


Figure 4: Activation profile clustering in the 9th and 10th layer. The neurons of the 9th and 10th layer start to show specificity to phonemes (*right*) and graphemes *left*. Hierarchical clustering was applied to time-independent NAPs and resulting clusters were highlighted by the same color in the dendrogram. In the 9th layer (*top*), there is no distinct clustering of groups of graphemes, as distances are not large enough. In contrast, phonemes already cluster into mostly meaningful phonetic groups. The 10th layer also shows more interpretable clustering for graphemes, as vowel characters are separated from the others. The phoneme clustering only changes slightly from layer 9 to 10, but distances between clusters increase.

In the 9th layer the clusters are very indistinct for graphemes (top left). A stronger similarity within vowel characters can be observed, but there are little differences in the distance to other grapheme activation profiles. This results in only 2 clusters, where one (blue) mainly includes rarely occurring graphemes ('Q', 'J', 'X, 'Z'). NAPs of rare graphemes show large distances to those of abundant graphemes, because for classes with few examples, example-specific information are not averaged out. In contrast, clustering neuron activation profiles for phonemes in the same layer (top right) reveals phonetically meaningful groups of phonemes. For example, the purple cluster includes only phonemes which correspond to vowels. The yellow cluster contains the remaining vowel phonemes, but is still more similar to the purple cluster than to the remaining phonemes. Another examplary meaningful cluster is the red one, which includes four plosives. Still, this cluster does not contain all plosives, as phonemes 'T' and 'D' (black cluster) are not included. Moreover, the activation profiles of those two phonemes are not even close to the ones in the red cluster. This indicates that either the phonetic concept of plosives is not reflected in the audio signal or the network has not learned this relation.

In the 10th layer (Figure 4 bottom) the clusters for both graphemes and phonemes are more distinct of each other than in layer 9. For graphemes the distance of the vowel characters (including 'Y') to the other characters increase. Therefore, they now form a cluster (highlighted in red). However, the activation patterns do not form phonetically meaningful clusters for other graphemes. We assume, this is due to the idiosyncrasy of the orthography resulting in smaller differences between neuron activation patterns of graphemes. In contrast, we observe more distinction between groups of phonemes in the network. There was only little change in which phonemes form clusters comparing the 9th to the 10th layer. However, the distance between phonemes in different clusters increases, which is shown by lighter colors in the visualization.

The fact that neuron activation profiles of graphemes are not well distinguishable in lower layers than the output layer is reasonable. As the network is trained to predict graphemes in the output layer, it has no reason to encode those graphemes in earlier layers. Moreover, there is no guarantee, that the intermediate layers have a phonetically meaningful interpretation. Yet, our results for the phonemes show that the network learned such a meaningful intermediate representation. This also explains the success of the transfer learning approach from English to German in Kunze et al. 2017 [14]. Because the intermediate phoneme representation can be used for both languages, the pre-trained English model was easily adaptable for German. Despite observing groups which are interpretable as similar phonetic features, not all observed clusters are explainable. For example, we could not observe a meaningful grouping of sibilants like phonemes 'S', 'Z', 'CH' or 'SH'. Also, the similarity of phonemes 'M', 'T' and 'F' (in the black cluster in both layer 9 and 10) is not explainable with phonetic concepts.

4 Conclusion

We investigated the question, whether an ANN for ASR learns phonemes as intermediate representations for predicting graphemes. To this end, we propose time-independent NAPs as group-characteristic network responses. We clustered those NAPs to reveal neuron specificity to phonemes and graphemes in different layers. In an exemplary speech recognizer, our method confirmed the hypothesis that phonemes are learned as intermediate representation for predicting graphemes. This indicates that the ANN learned to process speech by first perceiving acoustic features and inferring the words as graphemes from it. Learning these levels of abstraction seems to be straightforward, but is not guaranteed in ANNs. The network uses spectrograms as input, which is quite different to what a human would perceive. Moreover, orthography is highly idiosyncratic regarding acoustic features, so detecting phonemes is not necessarily an optimal processing step in the prediction of graphemes. Therefore, we conclude that there still is enough regularity in written (English) language, that an intermediate phonetic representation facilitates the prediction of graphemes.

Although demonstrated for a 1D-convolutional speech recognizer, our method is generally applicable to various neural network architectures. This for example includes CNNs with kernels of higher dimension and RNNs. The only requirement is the possibility to obtain groups of input samples, which are expected to be of different abstraction level. It can even be applied to unsupervised learning tasks, for example with Auto-Encoders or Deep Generative Models, as long as the input data can be grouped. Moreover, our method is open to any field of application, as it is independent of the kind of input data.

For our speech recognizer, we observed that phonemes are learned in very deep layers. Regarding pronunciation of words, phonemes are less ambiguous than graphemes. Therefore, we expected the encoding in earlier layers. This is an indication that earlier layers are not fully utilized as the architecture is too deep for the given task. In future work, we will further investigate, how our method can guide to achieve better interpretability of ANNs. To this end, we will optimize the network architecture based on the neuron specificity for phonemes and graphemes. Besides increasing specificity of neurons for those groups, this would significantly reduce the training time of the model.

At this point, our method needs manual assessment of the quality of clusters and their similarity to phonetic categories. First attempts on automatizing this assessment using the Silhouette score [26] have not been successful. Future work will explore more sophisticated methods for automatizing the process. This will for example include using linear classifier probes as proposed by Alain & Bengio (2016) [2].

Because our method is generally applicable and easy to implement, it has potential for interpreting decision processes in different kinds of ANNs in various fields of research. Therefore, we believe that this will motivate researchers to assess and increase the interpretability of their models.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research of Germany (BMBF) and supported by the donation of a GeForce GTX Titan X graphics card from the NVIDIA Corporation.

References

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[2] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[4] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE, 2016.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[6] N. Chomsky and M. Halle. *The Sound Pattern of English*. Studies in language. MIT Press, 1991.

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

[8] Nicholas Cummins, Shahin Amiriparian, Gerhard Hagerer, Anton Batliner, Stefan Steidl, and Björn W Schuller. An image-based deep spectrum feature representation for the recognition of emotional speech. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 478–484. ACM, 2017.

[9] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[10] Muriel Gevrey, Ioannis Dimopoulos, and Sovan Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160(3):249–264, 2003.

[11] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772, 2014.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[13] Andreas Krug and Sebastian Stober. Introspection for convolutional automatic speech recognition. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 187–199, 2018.

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

[15] Kevin Lenzo. The cmu pronouncing dictionary. *Carnegie Melon University*, 2007.

[16] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn. Understanding how deep belief networks perform acoustic modelling. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4273–4276. IEEE, 2012.

[17] Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *arXiv preprint arXiv:1806.05759*, 2018.

[18] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*, 20(14):5, 2015.

[19] Tasha Nagamine and Nima Mesgarani. Understanding the representation and computation of multilayer perceptrons: A case study in speech recognition. In *International Conference on Machine Learning*, pages 2564–2573, 2017.

[20] Tasha Nagamine, Michael L Seltzer, and Nima Mesgarani. Exploring how deep neural networks form phonemic categories. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[21] Tasha Nagamine, Michael L Seltzer, and Nima Mesgarani. On the role of nonlinear transformations in deep neural network acoustic models. In *Interspeech*, pages 803–807, 2016.

[22] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

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

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[25] K. Rao, F. Peng, H. Sak, and F. Beaufays. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229, 2015.

[26] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[27] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016.

[28] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[31] Shubham Toshniwal and Karen Livescu. Jointly learning to align and convert graphemes to phonemes with neural attention models. *CoRR*, abs/1610.06540, 2016.

[32] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5200–5204. IEEE, 2016.

[33] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[34] Kaisheng Yao and Geoffrey Zweig. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *CoRR*, abs/1506.00196, 2015.

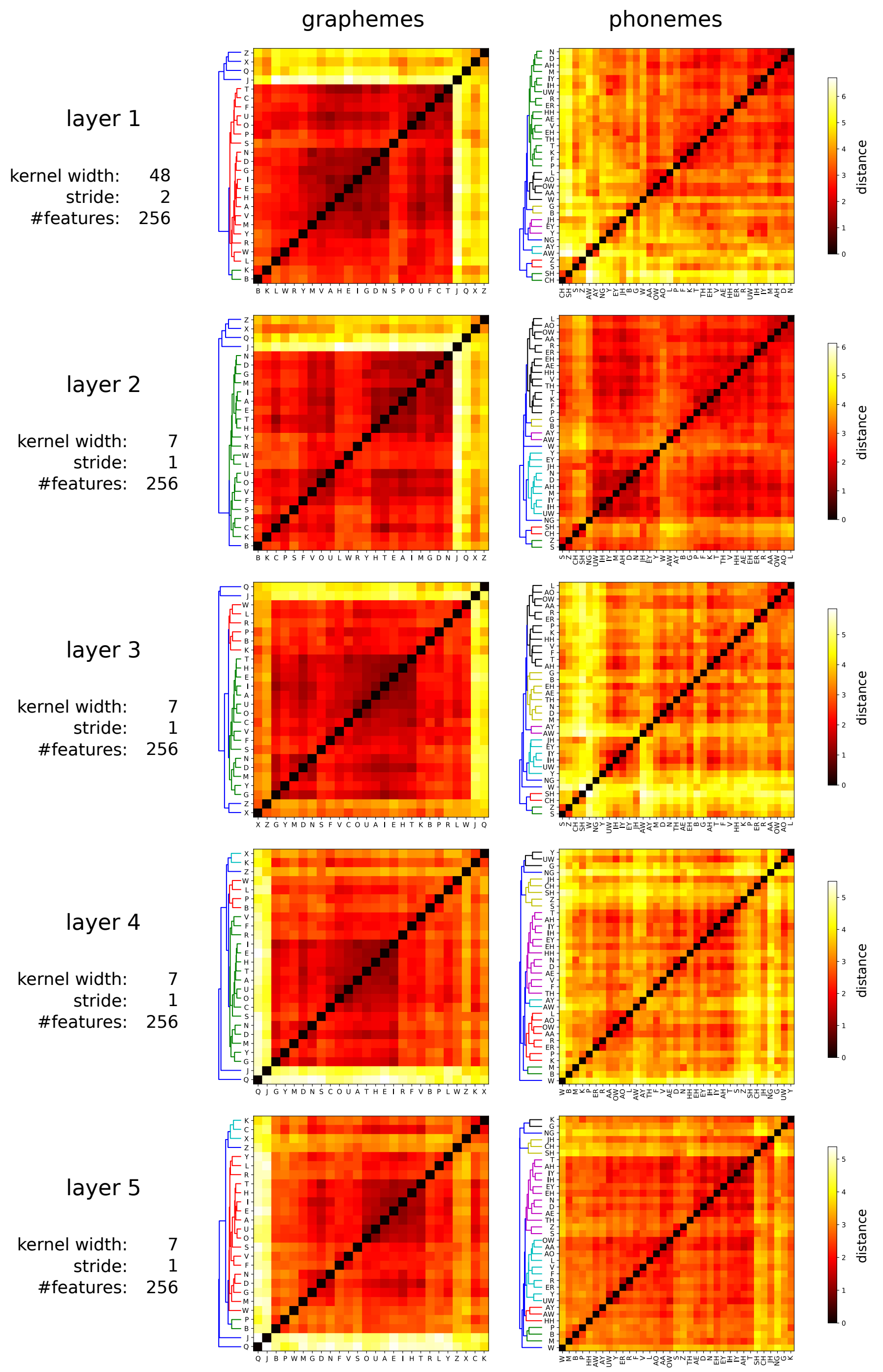
[35] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[36] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

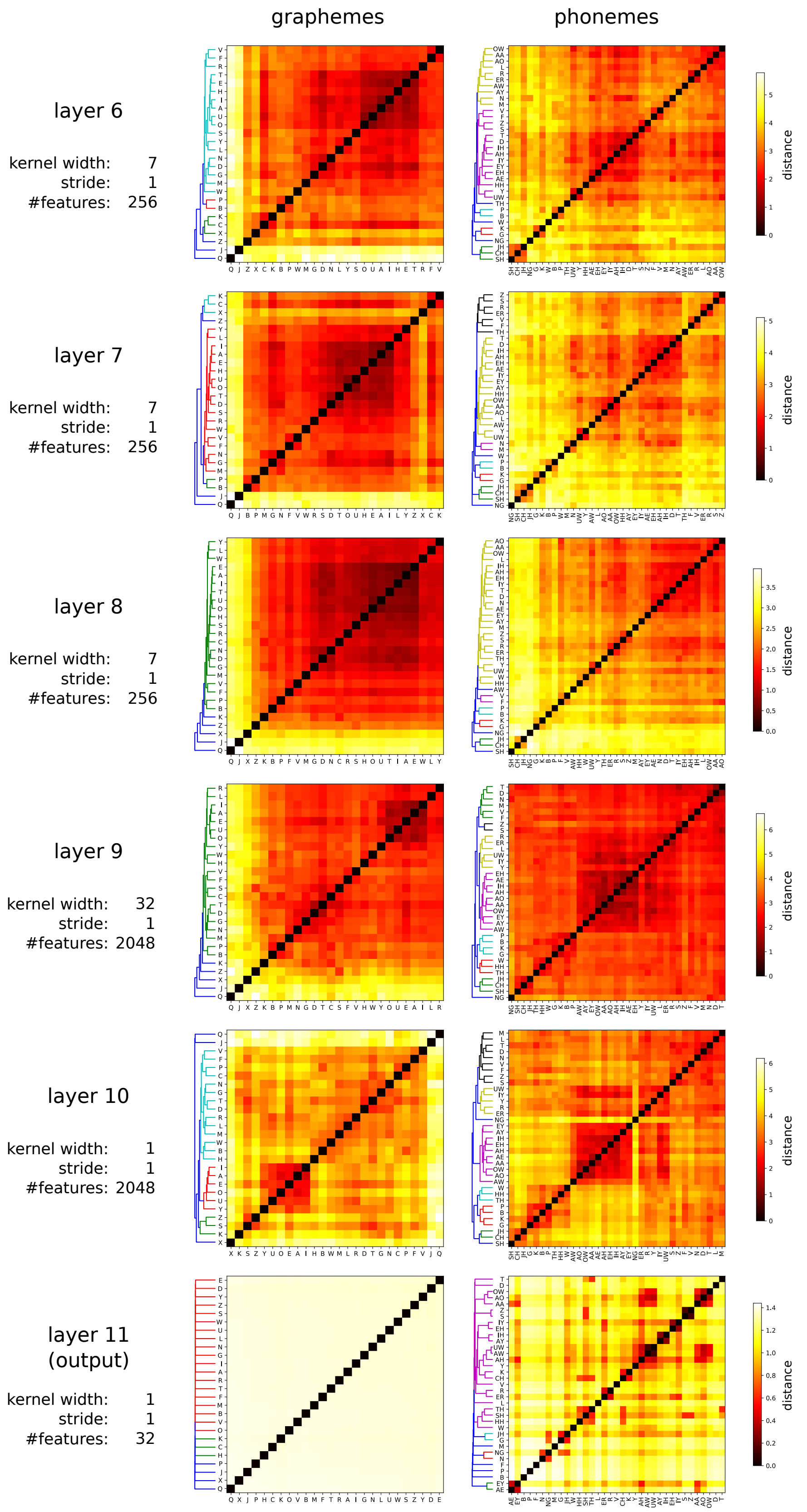
[37] Ying Zhang, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, Cesar Laurent Yoshua Bengio, and Aaron Courville. Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720*, 2017.

Appendix

A1 Clustering of NAPs in layers 1 to 5.



A2 Clustering of NAPs in layers 6 to 11.


graphemes
phonemes
layer 6
kernel width: 7
stride: 1
#features: 256
layer 7
kernel width: 7
stride: 1
#features: 256
layer 8
kernel width: 7
stride: 1
#features: 256
layer 9
kernel width: 32
stride: 1
#features: 2048
layer 10
kernel width: 1
stride: 1
#features: 2048
layer 11
(output)
kernel width: 1
stride: 1
#features: 32
distance