ON CONSTRUCTING AND ANALYSING AN INTERPRETABLE BRAIN MODEL FOR THE DNN BASED ON HIDDEN ACTIVITY PATTERNS

Khe Chai Sim

Computer Science Department, School of Computing
National University of Singapore, Singapore

ABSTRACT

Deep Neural Network (DNN) has been well received as a powerful machine learning model in a wide range of pattern classification tasks. Despite its superior performance in handling complex real-world problems, DNNs have been used pretty much as a *black box*, without offering much insights in terms of how and why high quality classification performance has been achieved. To address this problem, this paper studies the DNN hidden unit activities and presents a novel interpretable DNN visualisation technique that projects the hidden units of the DNN onto a meaningful 2-dimensional subspace. The projected points are displayed with colours to reflect the activation values for the purpose of visualisation. In this paper, the proposed technique is used to visualise two DNN acoustic models trained on the multi-condition data from the Aurora 4 corpus. The technique is able to produce a two dimensional representation of the DNN "brain" with interpretable regions. It also accentuates the effect of how the behaviour of the hidden units changes across different layers.

Index Terms— deep neural network, visualisation, interpretability

1. INTRODUCTION

This work is primarily motivated by the lack of interpretability of the Deep Neural Network (DNN) models despite the impressive performance it has achieved in solving many complex real-world classification problems. In speech recognition, DNN has been used to improve acoustic modelling in several ways. In a hybrid DNN/HMM system [1], a DNN is used to predict the posterior probability of the senones, replacing the Gaussian Mixture Model (GMM) in the conventional GMM/HMM systems. DNN has also been used as a feature extractor in the tandem systems [2]. There have been a number of techniques proposed to improve the robustness of DNN-based acoustic models in terms of speaker adaptation [3, 4, 5, 6, 7, 8] and noise compensation [9, 10, 11]. However, many of these techniques focused only on improving the classification performance without offering much insights.

This research was partially supported by the Google Faculty Research Award.

Many of the work that attempts to understand the DNN models come from the vision community [12, 13, 14, 15], which is not surprising given the nature of the task. There has not been much work from the speech community in analysing and understanding the DNN acoustic models. The work in [16] attempts to visualise how the speech features have been transformed by each hidden layer in a 2-dimensional space using t-distribution Stochastic Neighbour Embedding (t-SNE) [17]. t-SNE has also been used to visualise the outputs of multilingual bottleneck layers [18], By contrast, this paper proposes a novel DNN visualisation technique that emphasises on understanding the hidden units. Hidden units are projected onto a 2-dimensional activity space using t-SNE such that units that exhibit similar activity patterns are placed close to one another. This allows interpretable regions to be constructed for the purpose of visualising and analysing the hidden unit activations. The proposed technique can also be used to understand DNN models for other classification tasks.

Essentially, this paper will walk you through the process of building a "brain model" for a DNN. Section 2 first describes the experimental setup. The first step, which is to define a proper activity measure for each hidden unit, is described in Section 3. Section 4 describes the next step, which is constructing a 2-dimensional hidden activity space such that the placement of the hidden units in this space is based on the similarity of their activities. Section 5 describes how to construct meaningful and interpretable regions in this 2-dimensional activity space, which effectively groups the hidden units based on functionality. Section 6 explains how to utilise the activity space to display the hidden unit activities for visualisation, where activities for different phones, speakers and noise attributes will be compared.

2. EXPERIMENTAL SETUP

Two DNN acoustic models trained on the Aurora 4 dataset [19] are chosen for this study. The first acoustic model is trained on a window of 11 frames of the Linear Discriminant Analysis (LDA) [20] features. Each LDA feature is a 40-dimensional vector projected from 39×7 Mel Frequency Cepstral Coefficients (MFCCs) [21] features (including energy, delta and delta-delta over a 7-frame window). The second model is

trained on 11 frames of the 40-dimensional Constrained Maximum Likelihood Linear Regression (CMLLR) [22] transformed features. A global CMLLR transform is estimated for each speaker on top of the LDA features to maximise the likelihood of a GMM/HMM system. Both of the DNN models have a 440-dimensional input layer and 7 hidden layers of 2048 dimension each. Sigmoid nonlinear activation functions are used for all the hidden units. The LDA DNN model has a 2031-dimensional output layer while the CMLLR model's output layer size is 2013. These models achieved 12.5% and 10.1% word error rates on the Aurora 4 test set. All the acoustic models are trained using Kaldi [23].

In this work, three attributes are considered: *phone*, *speaker* and *noise*. For the phone attribute, 40 position-independent phones are used as the attribute instances. For the speaker attributes, there are a total of 83 training speakers and 10 development speakers. For the noise attributes there are 7 types of noise conditions (including the clean condition) and 2 channel conditions to give a total 14 noise attribute instances. There are 15.11 hours of speech (7137 utterances) in the multi-condition training set and 8.94 hours of speech (4620 utterances) in the development set.

3. HIDDEN UNIT ACTIVITIES

The first step towards building a brain model for a DNN is to define an activity measure for each hidden unit. Suppose that the hidden activation of the ith node in the lth layer at time t is given by $h_i^{(l)}(t)$, then the activity of the hidden unit with respect to a certain attribute (*e.g.* phone, speaker or noise), can be defined as an S-dimensional *activity vector*:

$$\boldsymbol{a}_i^{(l)} = \left[\begin{array}{cccc} a_i^{(l)}(1) & a_i^{(l)}(2) & \dots & a_i^{(l)}(S) \end{array} \right] \qquad (1)$$

where S is the total number of attribute instances and the sth element of the vector is given by

$$a_i^{(l)}(s) = \frac{\sum_t \gamma_s(t) h_i^{(l)}(t)}{\sum_{s=1}^{S} \sum_t \gamma_s(t) h_i^{(l)}(t)}$$

$\gamma_s(t)$ is the probability of associating attribute s with the acoustic feature at time t. For the phone attribute, $\gamma_s(t)$ can be obtained using a forward-backward algorithm (soft alignment) or a Viterbi algorithm (hard alignment). For speaker and noise attributes, $\gamma_s(t)$ is set to 1 for all the frames that belong to attribute s and 0 otherwise. Note that $\sum_s a_i^{(l)}(s) = 1$ and $a_i^{(l)}(s) \geq 0$ for all s[1]. Therefore, $a_i^{(l)}(s)$ effectively measures the weightage of the hidden unit's activations with respective to attribute s. A higher value of $a_i^{(l)}(s)$ indicates that the hidden unit will have a higher activation when the acoustic frame belongs to attribute s, and *vice versa*.

[1]This is true for sigmoid units where $h_i^{(l)}(t) \geq 0$. For other activation functions, it may be necessary to rescale the activation values.

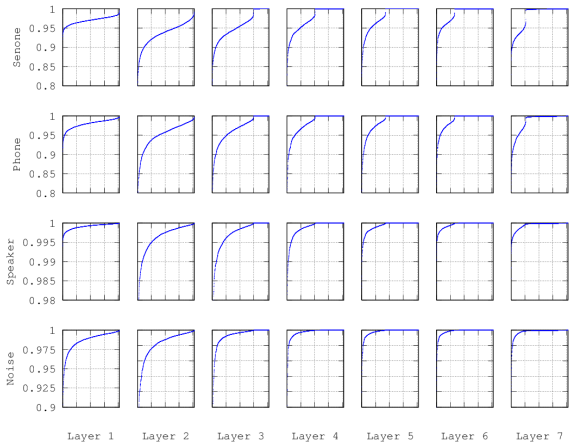

Fig. 1. Plots of normalised entropies for hidden units in different hidden layer. Hidden units are ordered according to ascending entropy values. Each row corresponds to an attribute (top to bottom): senone, phone, speaker and noise.

Since $\boldsymbol{a}_i^{(l)}$ is a probability (sum-to-one) vector, we can also measure the sensitivity of that hidden node with respective to an attribute using the *normalised entropy* measure:

$$E_i^{(l)} = \frac{-\sum_s a_i^{(l)}(s) \log a_i^{(l)}(s)}{-\sum_s \frac{1}{S} \log \frac{1}{S}}$$

The values of $E_i^{(l)}$ lie in the range $[0, 1]$. A lower entropy value means higher information content, which indicates a higher sensitivity to a given attribute. A hidden unit is insensitive to an attribute when $E_i^{(l)} = 1$, which happens when $\boldsymbol{a}_i^{(l)}$ is a uniform vector, *i.e.* $a_i^{(l)}(s) = 1/S$.

Fig. 1 shows the entropy profile for different hidden layers of the LDA DNN acoustic model with respect to the senones, phones, speakers and noise types. The hidden units are sorted in ascending order according to the entropy values. In general, there seems to be a consistent trend where the hidden units become less sensitive to all the attributes as the depth of the layer increases. In fact, there is a notable increase in the number of *insensitive* hidden units 'plateaued' at the top of the graph, where the normalised entropy is 1. This is clearly observed starting from layer 3 for all the attributes. This suggests that there are *inactive* hidden units in the DNN that do not contribute much to the final classification performance. To verify this, we prune the DNN model by removing hidden units whose entropy values based on the phone attribute are above a certain threshold. Setting the threshold to be the nth percentile of the entropy values of all the hidden units will retain n% of the hidden units after pruning. Fig. 2 shows the frame accuracies of the pruned DNN models on both the training and development data. There is only a very small drop in frame accuracies when up to 40% of the hidden units

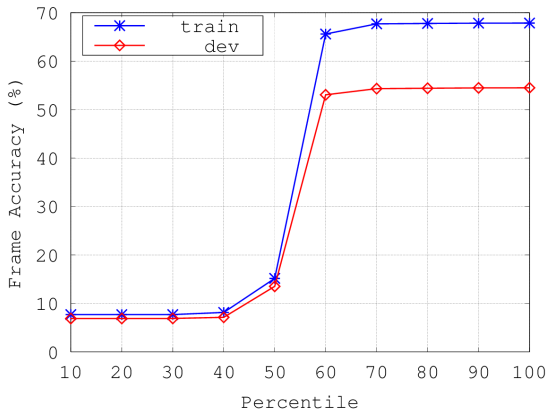

Fig. 2. Frame accuracies of the pruned DNNs on the training and development data for different pruning thresholds.

%	Hidden Layer Size						
	1	2	3	4	5	6	7
10	21	488	344	176	156	84	165
20	115	965	638	368	290	172	319
30	359	1331	911	539	430	291	440
40	734	1644	1158	702	569	422	505
50	1263	1874	1340	870	726	570	525
60	2008	2043	1488	1013	860	655	535
70	2048	2048	1489	1019	860	658	1913
80	2048	2048	1494	1259	860	1712	2048
90	2048	2048	1801	2048	861	2048	2048

Table 1. Comparison of hidden layer sizes for different levels of entropy-based pruning. The '%' column indicates the percentage number of hidden units retained after pruning.

are removed. Beyond that, the classification accuracy quickly drops to below 20% when 50% or more of the hidden units are removed. A more aggressive pruning beyond 50% leads to a large performance degradation. Table 1 shows the corresponding hidden layer sizes of the pruned DNN models for different pruning thresholds. At 60% and above, the hidden units being removed corresponds to those at the *plateau* region of Fig. 1. At 50% and below, sensitive units are being removed, which results in a large performance degradation.

Besides, it is interesting to note that the hidden units in the second hidden layer generally have a lower entropy with respect to all the attributes. This suggests that the network is potentially learning some attribute-specific information in the first two layers. For the speaker and noise attributes, which are not relevant to senone classification, the network may be trying to identify useful information that could be used to suppress the speaker and noise effects in the subsequent layers. It is also worth pointing out that the hidden units are generally less sensitive to the speaker attribute.

4. HIDDEN ACTIVITY SPACE

A *hidden activity space* is a 2-dimensional space where the position of the hidden units are determined such that hidden units that exhibit similar activity measures are placed closer together. Hence, the hidden activity space is specific to an attribute. Given the *activity vector* of each hidden unit with respect to an attribute (*c.f.* Eq. 1), the location of the hidden units in the *hidden activity space* can be obtained by applying the T-distribution Stochastic Neighbour Embedding (t-SNE) projection method [17]. Fig. 3 shows the projection of the hidden units for the different layers of the LDA and CMLLR DNN models in the *hidden activity space*. The top two, middle two and bottom two rows of the graphs show the *hidden activity space* with respect to the phone, speaker and noise attributes, respectively. These spaces cannot be compared directly as they are obtained by learning separate t-SNE projections. By contrast, it is possible to learn a single *hidden activity space* for all the hidden units from different hidden layers and across different DNN models, so long as a consistent set of *activity vectors* are used to describe the hidden units. This is useful as it allows easy comparison of models trained with different acoustic features, or model architecture. There are several interesting observations that can be made from Fig. 3, which will be discussed in the following subsections.

4.1. The shape of the hidden activity spaces

The shape of the hidden activity space is given by the convex hull of the projection of the hidden units in this space. This is the shape of our 'brain model' for each hidden layer! In general, the shape of the hidden activity spaces are rather similar across different hidden layers and between the LDA and CMLLR models. However, there is a noticeable difference in shape between the LDA and CMLLR models with respect to the speaker attribute. This is expected given the nature of the features these models are trained with. This suggests that there exists similar 'extreme' hidden units across different layers and even different DNN models that form the convex hull of the shape. It will be shown later that these 'extreme' points are indeed the sensitive hidden units with low entropy values.

4.2. The distribution of the hidden units

The distribution of the hidden units varies tremendously across different hidden layers. However, it looks relatively more similar when comparing the same hidden layer across the two different DNN models. This suggests that the functionality of the hidden units changes dramatically from one hidden layer to another.

For the phone attribute, the hidden units of the first layer are more or less uniformly distributed over the entire space except the 'hollow' region at the top left and the middle of the space (see the next point for the discussion of the 'hollow'

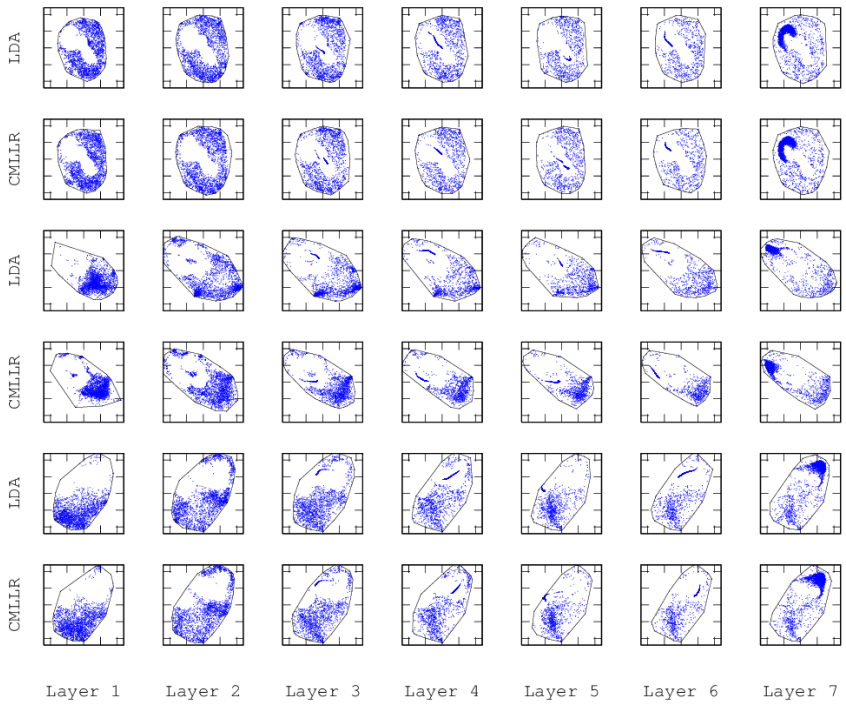

Fig. 3. Projection of hidden units onto the *hidden activity space* with respect to phones (top 2 rows), speakers (middle two rows) and noise (bottom 2 rows). The *hidden activity spaces* across different attributes cannot be compared directly.

regions). As the depth of the hidden layer increases to 3, two regions with higher density are formed at the top and bottom. At the same time, strips of hidden units begin to appear within the hollow region. Finally, in the last layer, most of the hidden units concentrate at the top left of the 'hollow' region.

Similarly, for the speaker and noise attributes, there are 'hollow' regions in their respective hidden activity spaces with very few hidden units across all the hidden layers. The hidden units in the first layer concentrate mostly at the bottom of the space; while the final layer hidden units are mostly at the top, forming a high density patch. As before, strips of hidden units begin to surface within the 'hollow' region starting from layer 3.

4.3. The 'hollow' region within the activity space

Fig. 4 depicts the condensed plot of the hidden units from all the layers of the two DNN models for each attribute. In addition, each hidden unit is now denoted by a larger circle filled with colours that reflect the entropy value of that hidden unit. A darker colour corresponds to a lower entropy, which indicates a more sensitive hidden unit. In general, more sensitive hidden units are mostly projected onto the outer region of the space; while less sensitive units concentrate in the middle.

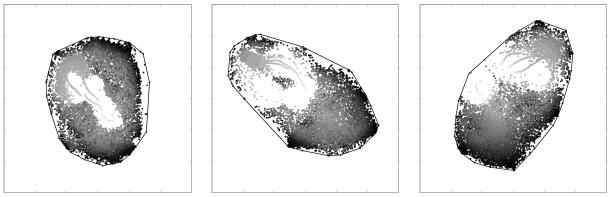

Fig. 4. Hidden units shown as larger circles filled with colours corresponding to their entropy values with respect to the phone (left), speaker (middle) and noise (right) attributes.

What happen is that the t-SNE projection attempts to place the highly sensitive hidden units (whose activity vectors are essentially close to a *'one-hot'* vector) as far apart as possible from each other as well as from the insensitive hidden units. Consequently, these highly sensitive hidden units tend to be pushed to the edge of the space, forming the 'extreme' points of the convex hull. Less sensitive units are found in the inner region of the space. The formation of the 'hollow' regions is the result of the gap created between the sensitive and insensitive hidden units. The strips of hidden units within the 'hollow region' are in fact the insensitive hidden units that form the *plateau* of the entropy profile graphs in Fig. 1.

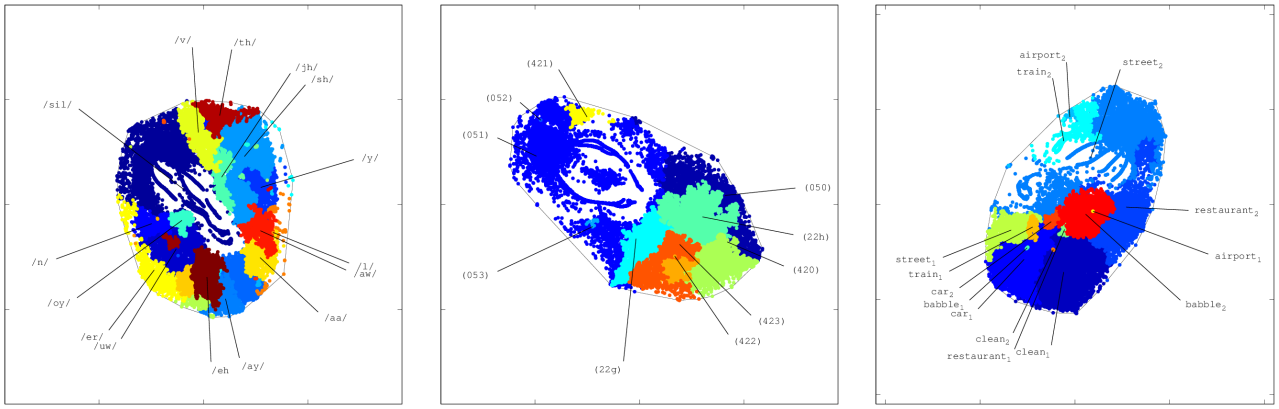

Fig. 5. Interpretable regions of the hidden activity space with respect to the phone (left), speaker (middle) and noise (right) attributes. Some labels are omitted for clarity.

5. INTERPRETABLE ACTIVITY REGIONS

Now that the hidden activity space is constructed for the hidden units, we are now ready to partition the hidden activity space into meaningful and interpretable regions according to the attribute of interest. Fig. 5 illustrates the interpretable regions of the hidden activity space for the phone, speaker and noise attributes. These regions are created using the following procedure:

1. Find a *seed* hidden unit for each attribute s;
2. Use the seed to initialise the regions, $\mathcal{P}_s$
3. Set the ranking threshold to $r^* = 1$;
4. For each s, recursively add all connected units where $r_i^{(l)}(s) \leq r^*$ to $\mathcal{P}_s$;
5. Increment the threshold: $r^* \leftarrow r^* + 1$;
6. If $r^* \leq S$, go to step 4;

The above procedure requires a set of neighbouring hidden units to be defined for each hidden unit. This is achieved by perform Delaunay triangulation [24] to construct a triangle mesh where the vertices of the triangles are given by the hidden unit locations. The edges of the triangles define the two connected vertices as neighbours. The procedure also requires a rank vector to be computed for each hidden unit:

$$\boldsymbol{r}_i^{(l)} = \left[\begin{array}{cccc} r_i^{(l)}(1) & r_i^{(l)}(2) & \ldots & r_i^{(l)}(S) \end{array} \right] \qquad (2)$$

where $r_i^{(l)}(s) \in \{1, 2, \ldots, S\}$ denotes the rank of attribute s according to $a_i^{(l)}(s)$ such that

$$r_i^{(l)}(s) < r_i^{(l)}(s') \iff a_i^{(l)}(s) > a_i^{(l)}(s') \qquad \forall s, s'$$

The first step of the procedure finds a *seed* hidden unit for each s, which has the largest average activity value computed from itself and its immediate neighbours. The seeds are then used to initialise the regions. The regions are expanded by adding all the connected hidden units whose rank values, $r_i^{(l)}(s)$, are smaller or equal to a threshold, r^*. The threshold is initialised as $r^* = 1$ and progressively incremented to gradually grow the regions until the entire hidden activity space is covered.

Fig. 5 shows that the hidden units (of all the layers) form clear attribute-dependent regions in the hidden activity spaces. For example, the top left region of the phone activity space corresponds to silence. Besides, the regions for two similar phones, /eh/ and /ay/, are found next to each other, at the bottom of the space. For the speaker attribute, speaker 051 occupies a large portion of the top left region of the space while the regions for speakers 052 and 053 are very small. For noise attribute, there are seven noise conditions, each has two channel types, denoted by the subscript 1 and 2 respectively. For certain noise conditions, such as `clean` and `car`, the two channel regions are next to each other. For `airport` and `train`, however, the channel regions are quite far apart. Some noise attributes have large regions, such as $\texttt{street}_2$, $\texttt{clean}_1$ and $\texttt{car}_1$. Other noise attributes have relatively smaller regions, such as $\texttt{babble}_1$, $\texttt{clean}_2$ and $\texttt{airport}_1$. Larger activity regions may correspond to more important and distinct attributes since more hidden units are used to identify them.

6. ACTIVITY VISUALISATION

Finally, we can use the hidden activity space to visualise and interpret the activation of the hidden units. Each hidden unit is visualised as a 2-dimensional triangle mesh constructed by applying the Delaunay triangulation algorithm to the hidden

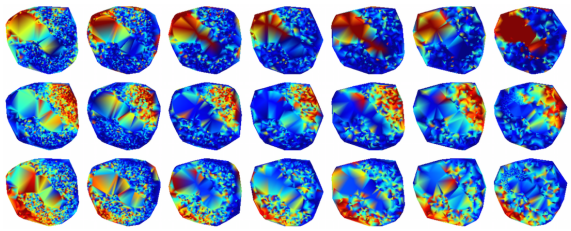

Fig. 6. Visualisation of the hidden unit activations for three different phones: /sil/ (top), /sh/ (middle) and /er/ (bottom).

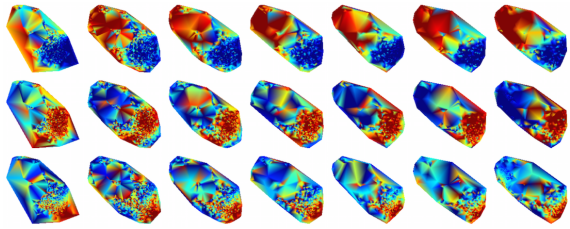

Fig. 7. Visualisation of the hidden unit activations for three different speakers: `051` (top), `22h` (middle) and `420` (bottom).

unit locations on the hidden activity space. Since the vertices of the triangles correspond to the hidden units, we can assign a colour to each vertex according to their activation. Each triangle is then filled with a gradient colour based on the linear interpolation of its vertex colours. Fig. 6, 7 and 8 show the visualisation of the hidden unit activations of the CMLLR model with respect to the phone, speaker and noise attributes, respectively. Each column corresponds to one hidden layer. Three attribute instances are chosen for each attribute. The jet colour map is used to convert the hidden unit activations into vertex colours. Blue colour indicates a low activation while red colour correspond to a high activation. As expected, the regions with high activation (red regions) correspond roughly to the respective interpretable regions of the attributes as depicted in Fig. 5. For example, the /sil/ phone and the `051` speaker show high activity in the top left region of their respective hidden activity spaces. On the other hand, $\texttt{clean}_1$ shows a prominent high-activity region at the bottom of the space.

In Fig. 6, the activity vector of the three phones, /sil/, /sh/ and /er/ are compared. For /sil/, the size of the high-activity region at the top left corner increases with increasing hidden layer depth. This shows that more hidden units are used to classify silence in the deeper layers, many of which actually correspond to the *insensitive* units as described in Section 3. Similar patterns are also observed for the `051` speaker and the $\texttt{train}_2$ noise in Fig. 7 and Fig. 8 respectively. On the other hand, for the /sh/ and /er/ phones, the high-activity regions are relatively smaller compared to that of /sil/ and they become more concentrated in deeper layers. This suggests that more

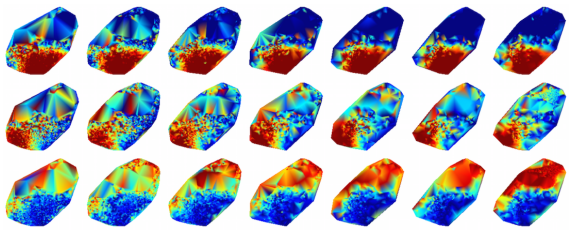

Fig. 8. Visualisation of the hidden unit activations for three different noise types: $\texttt{clean}_1$ (top), $\texttt{car}_1$ (middle) and $\texttt{train}_2$ (bottom).

of the hidden units in the first few layers are used to classify these two phones, perhaps in dealing with context, speaker and noise variability.

So, far, we have developed a method of constructing a visualisable and interpretable hidden activity space to help understand and analyse the hidden activity patterns of a DNN. We believe that having a better insight into how different hidden units response to different attributes will enable us to manipulate DNNs more effectively. Although not shown in this paper, this visualisation technique can also be used to observe changes in the hidden activity patterns over time. Hopefully, this work will open up possibilities and inspire future work to better understand, discover and manipulate DNNs. For the future work, we will extend this work to develop new speaker and noise adaptation methods for DNN.

7. CONCLUSIONS

This paper has presented a novel technique for analysing and understanding DNN in terms of the activity patterns of its hidden units. By examining the entropy of these activity patterns, it is possible to identify *insensitive* hidden units that have negligible contribution towards the final classification performance, which can be removed from the network without affecting much of its classification performance. Moreover, a 2-dimensional hidden activity space can be constructed such that hidden units that exhibit similar activity patterns are closer together in this space. The proposed technique is used to analyse two DNN models trained on the multi condition data from the Aurora 4 dataset. The analysis reveals that up to 40% of the hidden units can be removed without affecting much of the classification performance. Moreover, by measuring the sensitivity of the hidden units with respect to different attributes, interpretable regions within the hidden activity space can be constructed to yield a meaningful visualisation of the hidden unit activations. The size of the activity regions may also reflect the importance of different attributes. Besides, a consistent hidden activity space can be constructed across different hidden layers and DNN models, which make it possible to analyse how the hidden activity patterns change from one layer to another.

8. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[2] M. Ferras and H. Bourlard, "MLP-based factor analysis for tandem speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[3] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 24–29.

[4] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[5] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[6] A. Senior and I. L. Moreno, "Improving dnn speaker independence with i-vector inputs," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

[7] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Proc. Interspeech*, 2014.

[8] Pawel Swietojanski and Steve Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proceedings of the IEEE Workshop on Automatic Spoken Language Technology (SLT)*, 2014.

[9] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[10] B. Li and K. C. Sim, "Noise adaptive front-end normalisation based on vector Taylor series for deep neural networks in robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[11] B. Li and K. C. Sim, "A spectral masking approach to noise-robust speech recognition using deep neural networks," *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 2014.

[12] G. David Garson, "Interpreting neural-network connection weights," *AI Expert*, vol. 6, no. 4, pp. 46–51, Apr. 1991.

[13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Computing Research Repository (CoRR)*, vol. abs/1312.6034, 2013.

[14] Aravindh Mahendran and Andrea Vedaldi, "Understanding deep image representations by inverting them," *Computing Research Repository (CoRR)*, vol. abs/1412.0035, 2014.

[15] Matthew D. Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," *Computing Research Repository (CoRR)*, vol. abs/1311.2901, 2013.

[16] Abdel Rahman Mohamed, Geoffrey Hinton, and Gerald Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.

[17] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 219, no. 1, pp. 1–48, 2008.

[18] Ngoc Thang Vu, Jochen Weiner, and Tanja Schultz, "Investigating the learning effect of multilingual bottleneck features for ASR," in *Proc. Interspeech*, 2014.

[19] G. Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task, version 2.0," ETSI STQ-Aurora DSR Working Group, 2002.

[20] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *Proc. ICASSP*, 2000.

[21] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[22] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech and Languages*, vol. 10, pp. 249–264, 1996.

[23] D. Povey, a. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 1–4.

[24] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Computer and Information Sciences*, vol. 9, no. 3, pp. 219–242, June 1980.