ON THE USEFULNESS OF STATISTICAL NORMALISATION OF BOTTLENECK FEATURES FOR SPEECH RECOGNITION

Erfan Loweimi, Peter Bell and Steve Renals

Centre for Speech Technology Research (CSTR), School of Informatics, The University of Edinburgh
{e.loweimi, peter.bell, s.renals}@ed.ac.uk

ABSTRACT

DNNs play a major role in the state-of-the-art ASR systems. They can be used for extracting features and building probabilistic models for acoustic and language modelling. Despite their huge practical success, the level of theoretical understanding has remained shallow. This paper investigates DNNs from a statistical standpoint. In particular, the effect of activation functions on the distribution of the pre-activations and activations is investigated and discussed from both analytic and empirical viewpoints. This study, among others, shows that the pre-activation density in the bottleneck layer can be well fitted with a diagonal GMM with a few Gaussians and how and why the ReLU activation function promotes sparsity. Motivated by the statistical properties of the pre-activations, the usefulness of statistical normalisation of bottleneck features was also investigated. To this end, methods such as mean(-variance) normalisation, Gaussianisation, and histogram equalisation (HEQ) were employed and up to 2% (absolute) WER reduction achieved in the Aurora-4 task.

Index Terms— Deep Neural Networks, bottleneck features, probability density function, statistical normalisation

1. INTRODUCTION

Deep Neural Networks (DNN) play a central role in building the state-of-the-art pattern recognition systems. Their remarkable learning capability has triggered considerable research and led to many novel architectures that have notably elevated the performance in different tasks. Automatic speech recognition (ASR) is one of the fields which has witnessed a breakthrough due to DNNs (e.g. [1, 2]).

Despite great practical advances in terms of performance and building large-scale systems, theoretical understanding about deep learning has remained shallow. A deeper understanding of DNNs would be of considerable interest and could help to shed further light on these mysterious black-boxes. Furthermore, improved theoretical understanding can lead to improved practice, potentially contributing to training algorithms with faster convergence or less required training data.

In this work, we investigate the behaviour of neural networks from a statistical standpoint. The distribution of nodes in the network after using different activation functions is analytically derived and compared with empirical studies. The influential parameters and factors are discussed and, it is shown how and why Rectified Linear Units (ReLUs) give rise to sparsity. In addition, based on the statistical properties of the pre-activations, the usefulness of post-processing bottleneck features using various statistical normalisation techniques such as mean (and variance) normalisation, histogram equalisation [3] and Gaussianisation [4] has been studied and up to 2% (absolute) WER reduction has been achieved on the Aurora-4 [22] task.

Following an analytical study of the statistical distributions of the pre-activation and activations in a neural network (Section 2), we compare the developed theoretical models with empirical studies (Section 3). In Section 4, we report on a series of speech recognition experiments to investigate the effect of statistical normalisation of the bottleneck features.

2. STATISTICAL DISTRIBUTION OF BOTTLENECK FEATURES

2.1. Effect of Activation Function on Density Function

Consider a node in a network with input $\mathbf{x}$, weight $\mathbf{w}$, output y and activation function f where $y = f(\mathbf{w^T x})$. The scalar $z = \mathbf{w^T x}$ (T indicates transpose) is termed the *pre-activation*. The distribution of the pre-activation z, namely $P_Z(z)$[1] is unknown, however, the density of y can be derived analytically as a function of the distribution of z as follows [5]

$$P_Y(y) = P_Z(f^{-1}(y)) \left| \frac{d}{dy} f^{-1}(y) \right| \tag{1}$$

where f^{-1} is the inverse of the activation function f. To be invertible, $f(z)$ should be a one-to-one function. Although this is true for most of the activation functions such as Sigmoid (σ) or hyperbolic tangent ($\tanh$), it is not the case for the rectified linear unit (ReLU) unless a modified variant such as exponential linear unit (ELU) [6] is used. Note that the absolute value in (1) can be discarded when f^{-1} is a non-decreasing function. For the activation functions we consider, this is the case; hence we drop the absolute value symbols.

Supported by EPSRC Project EP/R012180/1 (SpeechWave).

[1] In $P_Z(z)$, Z denotes the random variable name and z is its value.

The process of estimating the distribution for different activation functions is similar. Here, to save space, we derive the density function of y when f is the $\tanh$ function:

$$f^{-1}(y) = \frac{1}{2}\log(\frac{1+y}{1-y}) \quad , \quad \frac{d}{dy}f^{-1}(y) = \frac{1}{(1-y^2)}$$
$$\Rightarrow P_Y^{\tanh}(y) = \frac{1}{1-y^2}P_Z(\frac{1}{2}\log\frac{1+y}{1-y}). \quad (2)$$

To understand and interpret (2), we should estimate $P_Z(z)$.

2.2. Approximating $P_Z(z)$

$P_Z(z)$ could be approximated as a Gaussian distribution, which can be justified by the central limit theorem (CLT) [5] since the pre-activation z is the weighted sum of the activations of the previous layer. The CLT makes some assumptions which may not be met well; we address this in Section 3. Assuming $z \dot{\sim} \mathcal{N}(z; \mu_z, \sigma_z^2)$, (1) can be rewritten as follows

$$P_Y(y) = \mathcal{N}(f^{-1}(y); \mu_z, \sigma_z^2) \left|\frac{d}{dy}f^{-1}(y)\right|. \quad (3)$$

Next the mean (μ_z) and the variance (σ_z^2) should be estimated. We assume the mean of the random variable z is zero. This may be more plausible for a function like $\tanh$ which enforces sign anti-symmetry and (at least a priori) there is no particular preference over positive/negative activation values.

As shown in Section 3, the zero-mean approximation also holds relatively well for Sigmoid and ReLU activations. Viewed from the perspective of the model's weights, although in these two cases the activations are always positive, there is no reason to (a priori) prefer positive weight values over negative ones or vice versa. This, approximately and in expected sense, pushes the mean of the random variable z toward zero.

2.3. Density Estimating for Nodes with tanh Activation

Assuming $z \dot{\sim} \mathcal{N}(z; 0, \sigma_z^2)$, the density function for $P_Y^{\tanh}(y)$, with some algebraic manipulation, can be derived as follows

$$P_Y^{\tanh}(y) = \frac{1}{1-y^2}\mathcal{N}(\frac{1}{2}\log\frac{1+y}{1-y}; 0, \sigma_z^2))$$
$$= \underbrace{\frac{1}{1-y^2}}_{F_Y^{<1>}(y)} \underbrace{\frac{1}{\sqrt{2\pi}\sigma_z}(\frac{1+y}{1-y})^{-\frac{1}{8\sigma_z^2}\log\frac{1+y}{1-y}}}_{F_Y^{<2>}(y,\sigma_z)} \quad (4)$$

where $F_Y^{<1>}(y)$ and $F_Y^{<2>}(y, \sigma_z)$ are two main factors of $P_Y^{\tanh}(y)$. The structure of the former is related to the activation function type of the current layer (y) and the form of the latter relates to the pre-activations and the previous layer. Fig. 1. shows these two parts along with $P_Y^{\tanh}(y)$ for different values of σ_z which is the main parameter of $P_Y^{\tanh}(y)$.

As Figs. 1(b) and (c) illustrate, σ_z acts as a shape parameter and depending upon whether it is less than, equal to or more than 1, the overall shape of the second part, and consequently the total density function change. Fig. 1(c), depicts

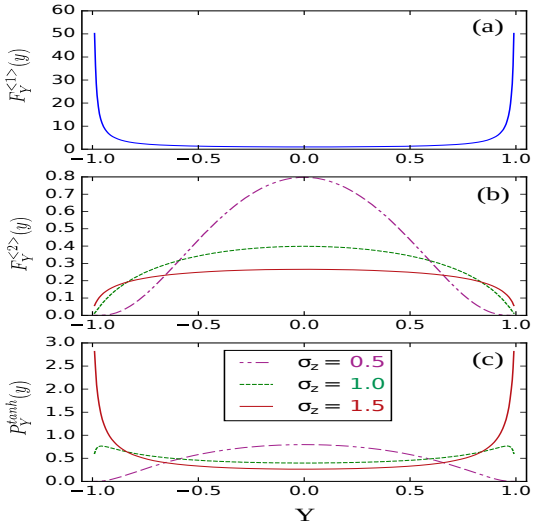

Fig. 1. Components of the $P_Y^{\tanh}(y)$. (a) $F_Y^{<1>}(y)$, (b) $F_Y^{<2>}(y)$. When σ_z increases $F_Y^{<2>}(y)$ becomes flatter and $F_Y^{<1>}(y)$ dominates the product and consequently $P_Y^{\tanh}(y)$.

$P_Y^{\tanh}(y)$ for $\sigma_z = 1$, $\sigma_z = 0.5$ (exemplifies the shape of $P_Y^{\tanh}(y)$ when $\sigma_z < 1$) and for $\sigma_z = 1.5$ (represents the $\sigma_z > 1$ case). When $\sigma_z < 1$, the second part dominates the density and it further resembles a bell-shaped function. On the other hand, by increasing this parameter, specifically when $\sigma_z > 1$, $F_Y^{<2>}(y, \sigma_z)$ becomes flatter and the first factor dominates the product and the distribution $P_Y^{\tanh}(y)$.

The question at this point would be the range of the σ_z and particularly whether it is less or larger than unity.

2.4. Non-linearity of NNs and Density Shape Parameter

Although it appears hopeless to analytically derive some approximation for σ_z value, looking at the Fig.1(c) from linear/non-linear systems standpoint is insightful. After all one of the factors which largely contributes to the capabilities of the DNNs is that they are non-linear models. The depth of the network which plays a crucial role in the success of such models, becomes meaningless if the activation functions were linear. Also recall that for $y = \tanh(z)$, by moving away from 0 toward 1 or -1, the behaviour of this function changes from almost linear in the vicinity of 0 to a non-linear function when $|y|$ gets close to one.

As Fig. 1(c) illustrates, when $\sigma_z < 1$, the bulk of the probability of y is around zero. This is the place where the nodes and collectively the network behave like a linear system. It is not desirable unless the process/function to be modelled is approximately linear in which case, DNNs alternatives are remarkably cheaper options. When $\sigma_z > 1$, the probability mass moves toward edges (-1 and 1 for $\tanh$) and the non-linearity of the overall system increases which is desirable in modelling complicated decision borders. Based on this argument, we expect σ_z to be noticeably larger than one to make the model operates in the non-linear mode.

3. EMPIRICAL STUDIES

To examine the validity of the derived formula and the associated assumptions, we compare with empirical studies. In this regard, a time-delay neural network (TDNN) [7] consisting of 7 layers with a bottleneck layer just before the output layer is employed. The network was trained by Kaldi [8] (nnet3) using WSJ-5k (SI-84 set). The input is 40-dimension mean-variance normalised log-filterbank energies, with hidden layers 1024 nodes, and a bottleneck layer with 40 nodes. The output layer consists of about 2000 nodes (number of state-clustered triphones). The complete SI-84 set was used which provides more than 5.4 M frames (for 10 ms frame shift).

3.1. Discussion

Fig. 2 depicts the error bar (mean $\pm$ standard deviation), covariance matrix and histograms of the pre-activation (z) as well as the activation (y) for all the 40 bottleneck features. As illustrated in Fig. 2(a), the mean of the pre-activation z tends to zero and the standard deviation is larger than one which matches with the assumptions made in Section 2.2 and argument propounded in Section 2.4. Further, the covariance matrix diagonal-dominant structure shows that the network decorrelates the features in the bottleneck layer. This lends theoretical support to why the bottleneck features can be used directly in an HMM-GMM system without the need to be decorrelated, e.g. through discrete cosine transform (DCT).

Comparing Fig. 2(b) and (d) also explains why as bottleneck features, the pre-activation (z) should be used, not the activations (y). The pre-activation distribution can be easily fitted by a mixture of a few Gaussians whereas fitting y with a GMM is more problematic. Finally, Fig. 2(d) shows the nodes are mostly operating in the saturation regions (close to 1 or -1) and this allows the network to do non-linear modelling as argued in Section 2.4.

Fig. 3 shows similar statistics when Sigmoid is used as activation function and the points mentioned for the $\tanh$, extends to the logistic function, too.

3.2. Sparsity of ReLU

For the ReLU activation function there is a concentration of activation values around positive zero (0^+) (Fig. 4)[2]. In this study, we observed that the histogram support extends to about 35. However, after 0.15 the histogram values get almost zero and the overwhelming bulk of the density occurs around positive zero. An important advantage of this point is boosting the sparsity which makes the network more biologically plausible [9, 10] and also brings about some mathematical advantages from modelling and learning viewpoints [11].

Glorot et al [12] observed the sparsity of ReLU, explaining it based on rectifying nodes behaviour and assuming the number of negative pre-activations equals the positive ones.

[2]Fig. 4 illustrates the truncated support of Y histogram which was done for a better visualisation.

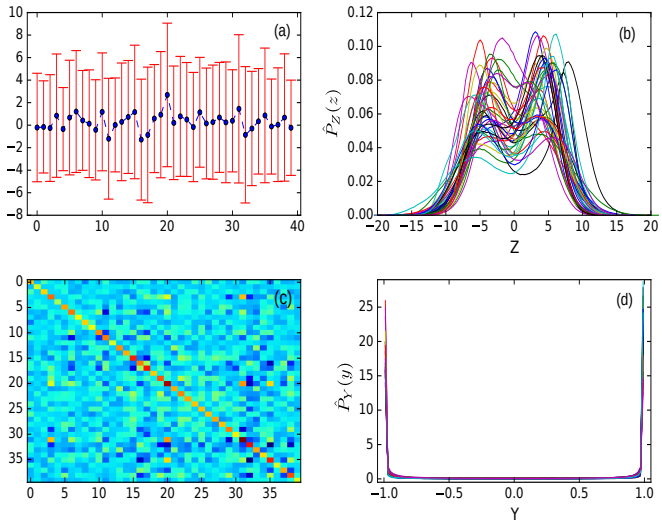

Fig. 2. Statistical analysis of preactivation (Z) and activation (Y) for all nodes in the bottleneck layer when $\tanh$ is used. (a) Error bar of preactivations ($\mu_z \pm \sigma_z$), (b) distribution of Z, (c) covariance matrix of Z, (d) distribution of Y.

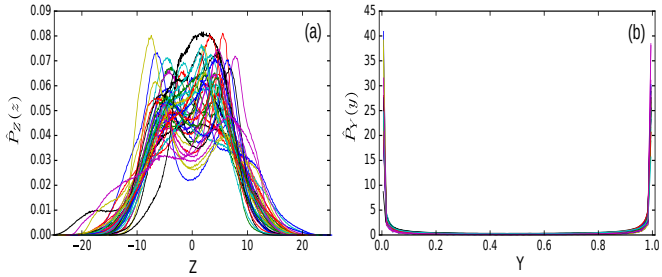

Fig. 3. Distributions of the pre-activation (Z) and activation (Y) when Sigmoid is used as activation function.

Fig. 3(b) supports this argument for Sigmoid, too. That is, about half of the probability mass is around (positive) zero but Sigmoid does not afford sparsity. From a statistical viewpoint, sparsity intuitively means the density function have a dominant mode in zero. Although for Sigmoid there is a mode at zero (Fig. 3(b)), it is not dominant as there is a big mode around one. Therefore, this does not translate into sparsity.

We believe the sparsity provided by ReLU is better justified based on the argument presented in Section 2.4: to get the network operate in non-linear mode, the operating point of the units should be around positive zero because before zero ReLU blocks information and after zero it acts like a linear system. Therefore, the sparsity of ReLU is due to the coincidence of zero activations with the *only* region where ReLU shows the desirable non-linear behaviour.

3.3. Gaussian Approximation for Pre-activation

As seen in Figs. 2(b), 3(a) and 4(a), distribution of Z is not Gaussian, although it can be well-fitted with a few Gaussians.

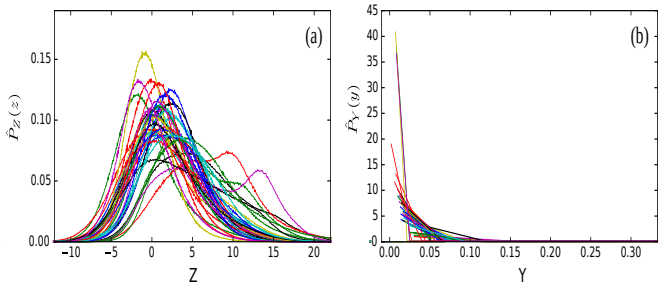

Fig. 4. Distributions of the per-activation (Z) and activation (Y) when ReLU is used as activation function. The support (x-axis) of Y histogram is truncated for a better visualisation.

Nevertheless, a Gaussian approximately fits the mean of all the pre-activation densities. This, however, does not undermine (4) because as argued in Section 2, due to $\sigma_z > 1$, $F_Y^{<2>}(y, \sigma_z)$ plays a marginal role and $F_Y^{<1>}(y, \sigma_z)$ is the dominant factor. As such this approximation and also the error associated with the zero-mean assumption is insignificant.

4. STATISTICAL NORMALISATION OF THE BOTTLENECK FEATURES

Statistical properties of the pre-activations in the bottleneck layer, make the bottleneck features (BN) a good representation for a GMM-HMM ASR system [13, 14, 21]. Also, they are amenable to be post-processed through statistical normalisation techniques. In [15–17] the effect of *pre-processing* the features before feeding the DNN through vector Taylor series (VTS) [18] and generalised VTS [19, 20] methods was studied. It was observed that in mismatched conditions between the test and train conditions, e.g. training with additive noise, testing in presence of channel mismatch, significant performance gains can be achieved.

Here, we aim at investigating the usefulness of *post-processing* the BN features through DCT, mean normalisation (MN), mean-variance normalisation (MVN), Gaussianisation (G) and histogram equalisation (HEQ). All the normalisations has been done on the utterance level. Note that techniques like VTS, although are more powerful for normalising the features, are not applicable here as they require the environment model which is not available after DNN processing.

The ASR system was build based on Kaldi recipe for Aurora-4, as described in Section 3.1. For training the network, the noisy training set was used in which the data is distorted by only additive noise. Using such training data allows for a better investigation of the statistical normalisation effect in the matched (A and B) and mismatched (C and D) conditions (Aurora-4 has four test sets: A (clean data), B (additive noise), C (channel mismatch) and D (both additive and channel distortion). Table 1 shows the results for a GMM-HMM state-clustered triphone system trained by Kaldi and $Ave4$ indicated the average WER of the four test sets.

Since DNNs learn a sequence of optimal linear/non-linear transforms, do not leave that much room for shallow signal processing normalisation techniques to improve the performance unless there is some mismatch between the test/train conditions which makes the learned transforms subopitmal. However, Table 1 shows, post-DNN statistical normalisation can lead to a significant performance improvement.

Table 1. *WER for Aurora-4 (LDA-MLLT [8]).*

Feature	A	B	C	D	$Ave4$
BN	3.87	7.96	21.80	32.72	16.58
BN+MN	3.64	7.66	21.02	32.20	16.13
BN+MVN	4.07	8.31	20.34	33.04	16.44
BN+G	4.15	8.12	20.18	32.67	16.28
BN+HEQ	3.96	7.43	19.76	30.87	15.50
BN+DCT	3.77	7.77	21.76	32.49	16.44
BN+DCT+MN	3.96	7.82	20.19	32.08	16.01
BN+DCT+MVN	3.98	8.15	20.77	32.79	16.42
BN+DCT+G	4.18	8.12	21.07	33.01	16.59
BN+DCT+HEQ	3.98	7.35	20.49	30.94	15.69

Techniques like MN, MVN and Gaussianisation work best when the feature density function is quasi-Gaussian. As well as Figs. 2-4, the difference between the MVN and Gaussianisation, although is small, indicates the feature distribution is not Normal, as for this distribution they are identical. Among these normalisation techniques, MN returns a higher and consistent gain over the baseline system (unprocessed BN) in both matched and mismatched conditions.

DCT does not lead to any noticeable gain which prove that the DNN, among others, decorrelate the data in the bottleneck layer, as shown in Fig. 2(c). Such decorrelation not only facilitates training GMMs with diagonal covariance matrices, but also helps in making the Gaussianisation and HEQ techniques more optimal. That is, for mathematical convenience both are carried out for each dimension independently.

HEQ appears to be the best option for normalising/post-processing the BN features: theoretically, it is more flexible as neither requires Gaussianity nor the environment model and performance-wise, it achieves the highest gain compared with other normalisation techniques. As seen in Table 1, the maximum WER reduction is 2% (absolute), achieved for test set C when BN features were post-processed by HEQ.

5. CONCLUSION

This paper studied the DNNs from a statistical standpoint. The density function for $\tanh$ activation functions was analytically derived, the results compared with empirical studies, and the influential factors were discussed. Furthermore, the usefulness of statistical normalisation techniques for post-processing bottleneck features was evaluated and histogram equalisation returned the highest gain. Investigating the effect of statistical normalisation of the bottleneck features in the low and middle layers is recommended for future work.

6. REFERENCES

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving Human Parity in Conversational Speech Recognition," *ArXiv e-prints*, Oct. 2016.

[2] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English Conversational Telephone Speech Recognition by Humans and Machines," *ArXiv e-prints*, Mar. 2017.

[3] A. Torre, A.M. Peinado, J.C. Segura, J.L. Perez-Cordoba, M.C. Benitez, and A.J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, May 2005.

[4] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *2001: A Speaker Odyssey - The Speaker Recognition Workshop*, Crete, Greece, 2001, pp. 213–218, International Speech Communication Association (ISCA).

[5] A. Papoulis and S.U. Pillai, *Probability, Random Variables, and Stochastic Processes*, McGraw Hill, fourth edition, 2002.

[6] A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015.

[7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Readings in speech recognition," chapter Phoneme Recognition Using Time-delay Neural Networks, pp. 393–404. 1990.

[8] 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 *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[9] D. Attwell and S.B. Laughlin, "An energy budget for signaling in the grey matter of the brain," *Journal of Cerebral Blood Flow & Metabolism*, vol. 21, no. 10, pp. 1133–1145, 2001.

[10] P. Dayan and L.F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, The MIT Press, 2005.

[11] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, Chapman & Hall/CRC, 2015.

[12] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Geoffrey Gordon, David Dunson, and Miroslav Dudk, Eds., 2011, vol. 15 of *Proceedings of Machine Learning Research*, pp. 315–323.

[13] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for lvcsr of meetings," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, April 2007, vol. 4, pp. IV–757–IV–760.

[14] F. Grezl and P. Fousek, "Optimizing bottle-neck features for lvcsr," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 4729–4732.

[15] M. Seltzer, . Yu, Y. Wang, and M. Seltzer, "An investigation of deep neural networks for noise robust speech recognition," in *ICASSP 2013*, January 2013.

[16] E. Loweimi, J. Barker, and T. Hain, "Exploring the use of group delay for generalised vts based noise compensation," in *ICASSP*, 2018.

[17] Erfan Loweimi, *Robust Phase-based Speech Signal Processing; From Source-Filter Separation to Model-Based Robust ASR*, Ph.D. thesis, Sheffield, UK, Feb 2018.

[18] P.J. Moreno, B. Raj, and R.M. Stern, "A vector taylor series approach for environment-independent speech recognition," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, May 1996, vol. 2, pp. 733–736 vol. 2.

[19] E. Loweimi, J. Barker, and T. Hain, "Use of generalised nonlinearity in vector taylor series noise compensation for robust speech recognition," in *INTERSPEECH*, San Francisco, USA, 2016, pp. 3798–3802.

[20] E. Loweimi, J. Barker, and T. Hain, "Channel compensation in the generalised vector taylor series approach to robust asr," in *INTERSPEECH*, Sweden, 2017, pp. 2466–2470.

[21] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International conference on*, Florence, Italy, May 2014.

[22] N. Parihar and J. Picone, "Aurora working group: Dsr front end lvcsr evaluation au/384/02," *Inst. for Signal and Information Process, Mississippi State University, Tech. Rep*, vol. 40, pp. 94, 2002.