International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 5, May 2016

Learning a Phoneme Manifold Using Multitask Learning for DNN based Synthesis of Children's Stories

[1]Naina Teertha, [2]Sai Sirisha Rallabandi, [3] Sai Krishna Rallabandi, [4] Dr. Kumaraswamy, [5] Suryakanth V Gangashetty
[1], [4] Siddaganga Institute of Technology, Tumkur, Karnataka, [2],[3],[5] International Institute of Information Technology, Hyderabad
[1]naina.i91i98@gmail.com, [2] sirisha.rallabandi@research.iiit.ac.in, [3]saikrishna.r@research.iiit.ac.in, [4]hyrkswamy@gmail.com, [5]svg@iiit.ac.in

***Abstract*— Deep Neural Networks(DNNs) use a cascade of hidden representations to enable the learning of complex mappings from input tom output features and are shown to produce more natural synthetic speech than conventional HMM-based statistical parametric systems. However even though it offers greater flexibility and controllability than unit selection, the naturalness of speech generated by DNN SPSS is still below that of human speech, and cannot compete with good unit selection systems. DNNs are very powerful models and it might be the case that we haven't yet found the best possible way to use them. In this paper, we investigate the learning of a phoneme manifold as a secondary task in a Multitask Learning setting for acoustic modeling and show that the hidden representation used within a DNN can be improved using such a method. The rationale behind the techniques is independent of the architecture and can also be extended to the recurrent/recursive variants of the neural networks.**

***Index Terms*—Manifold, Multitask Learning, Synthesis.**

I. INTRODUCTION

Statistical Parametric Speech Synthesis (SPSS) has made significant advances in naturalness [1] and is highly intelligible [2]. Zen et al.[1] suggest various factors which limit naturalness or quality.

Neural Networks have re-emerged as a potential powerful acoustic model for SPSS. In [4],[5],[6],[7],[8], feed-forward neural networks are employed to map a linguistic representation derived from input text directly to acoustic features. In [5], a Deep Belief Network (DBN) was used to model the linguistic and acoustic representations jointly. In [9] and [10] Mixed Density Networks (MDNs) and real valued neural autoregressive density estimators (RNADEs) were proposed respectively, to predict acoustic feature distributions given input linguistic features. In [11], authors point out that replacing decision trees with DNNs and moving from state-level to frame-level predictions both significantly improve listener's naturalness ratings of synthetic speech produced by the systems.

Despite this, it would be difficult to argue as of now that deep neural networks have had the same success in synthesis that they have had in ASR. DNN –influenced improvements in synthesis have mostly been fairly moderate. This becomes fairly evident when looking at the submissions to the Blizzard Challenge [12] in the recent past. Few of the submitted systems use DNNs in any part of the pipeline, and those that do use DNNs, do not seem to have any advantage over traditional well-trained systems. Even in the cases where improvements look promising, the techniques have had to rely on the use of much larger datasets than is typically used. The end result is that the DNN based SPSS ends up having to loose the advantage it has over traditional unit selection systems [3] in terms of amount of data needed to build a reasonable system.

DNNs are extremely powerful models, and like many algorithms at the forefront of machine learning research, it might be the case we have not yet found the best possible way to use them. With this in mind, in this paper we investigate the usage of multitask learning with different levels of secondary tasks for acoustic modeling in DNN based SPSS.

1.1. Motivation for the Proposed Method of Spectral Mapping

Although a lot of issues pertaining to the regression mapping have been addressed, there is one problem that has not been stressed upon in literature so far that of sub optimality while training the mapping function,

which arises due to design of optimization function. The training criterion typically aims to maximize the likelihood of spectral features of speech which might not be the best representation. However, as the choice of speech features is constrained by the requirements of vocoder, many interesting and powerful representations which might lead to better output quality are avoided from being used in the mapping function. In this submission, we investigate the use of multitask learning (MTL) [13] in a DNN framework to alleviate this problem. Eventhough, similar approach has been explored in [14], all the secondary tasks are at frame level (formants,LSF,etc).

1.2. Mutlitask Learning

Mutlitask Learing(MTL) is a mechanism to train a global model for various different yet related tasks using a shared representation[13]. Typically, there is one main task and one or more secondary tasks. It is generally believed that model learned in multitask learning can generalize better and make more accurate predictions than a model for a single task, provided that the secondary task(s) are related to the main task and at the same time complementary. MTL has produced good results in Speech Recognition [15], Synthesis [16] and Natural Language Processing [17]. When using MTL with a DNN, the main task and the secondary tasks share the same hidden representations. The extra target outputs associated with additional tasks are added to the original output for training the network and are discarded during the speech generation. In [9], acoustic features and various secondary features were trained together to improve voice quality of SPSS, demonstrating that the statistical model can be improved if the second task is chosen well. Specifically, the second task should be related with the primary task, with parameter sharing serving to improve the structure of the model. The DNN learns to predict a representation of the target speech as a secondary task, in parallel to learning to predict the usual invertible vocoder parameters as the main task. The predictions of the representation are discarded at reconstruction time as their purpose is to guide the hidden layers of the network during training towards obtaining a qualitatively more robust representation by providing additional supervision.

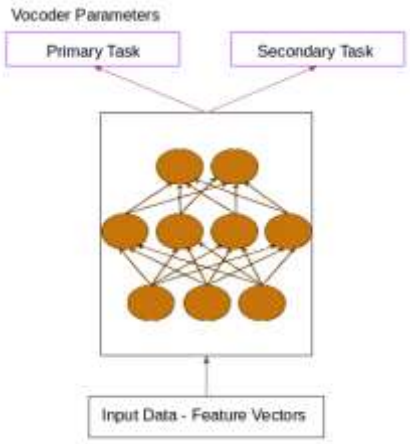

Figure 1: Implementation of Multitask Learning

In the current task there are three possible levels at which secondary task can be chosen:

- At the utterance level, with the task being minimizing a metric such as the MCD
- At the speech segment level, with the task being maximizing the probability of intermediate phoneme representations such as bottleneck or phoneme manifold or continuous representation obtained from text in an unsupervised method such as LSA or positive point wise mutual information matrix factorization technique and
- At the frame level, where the task is to predict perceptual representation of speech such as formants or LSF, so as to improve the quality of final output.

In our current work, we investigate the system performance at all the sub segmental levels. In particular, at the utterance level, we try to minimize the MCD of the test utterance, try to maximize the phoneme manifold at the segment level, and predict formant frequencies [16] at the frame level. Although there is interesting possibility to see the performance of the weighted combination of the different secondary tasks, we have not done it in the current work.

The paper is organized as: In Section 2, we briefly describe Secondary tasks, followed by the implementation of the system along with the chosen secondary task in section 3 We evaluate the designed systems in section 4 followed by the conclusion.

II. SECONDARY TASKS

2.1. Phoneme Manifold

A Manifold is a non-Euclidean space that can be approximated by Euclidean patches in small neighborhoods.

We assume that speech resides on a manifold μ of dimension d within with d < D. Manifold serves as a useful low dimensional representation of a segmental entity such as phoneme and is already employed for applications such as speech recognition and denoising[19]. It has been suggested that the acoustic feature space is confined to lie on one or more low dimensional manifolds. Therefore, a feature space transformation technique that explicitly models and preserves the local relationships of data along the underlying manifold should be more effective for speech processing. We investigate if learning to predict the phoneme manifold serves as a secondary task, aiding the vocoder parameter prediciton. For this, we have obtained phone boundaries using NNET recipe of Kaldi Speech Recognition toolkit [20].

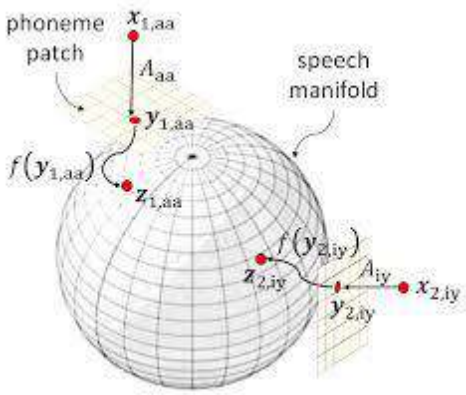

Figure 2: Illustration of Phoneme Manifold

The phoneme manifold was obtained the following way:

- Conversion of the time-domain representation of y[n] to the spectral domain using the short-time Fourier Transform (STFT) with a 20ms Hamming window shifted by 10ms. We used a 512-point FFT for the STFT, so the resulting spectral representation is in D = 257-dimensional space.

- For each phoneme (boundaries obtained from Kaldi), we embedded its speech frames into a low dimensional space using Neighborhood Preserving Embedding (NPE)[21]. NPE preserves the local Euclidean structure by representing *XIP*, the *i*th speech frame labeled as phoneme *p*, as a linear combination of its *K* nearest neighbors, and preserving the weights in the linear combination when each speech frame is mapped to a low dimensional space. This ensures that the local geometry in high-dimensional space will be retained in low dimensional space.

- We use the Manifold Charting method to join these patches together [22]. Manifold Charting determines the best global Cartesian coordinate system to represent the speech frames by re-aligning the coordinate system of each of the phoneme patches.

Thus, with the phoneme maps *Ap* from NPE and the global coordinate map *f* from Manifold Charting, we have a non-linear mapping between the high-dimensional spectral space and the low-dimensional speech manifold: *zip* = *f*(*yip*) = *f*(*Apxip*).

2.2. Continuous Representation of Text Using PPMI Matrix:

Distributed dense representations of phonemes are shown to improve the performance of the connectionist repressors [23] for prediction of the acoustic parameters. We wish to investigate if the representation derived in such an unsupervised fashion using the co-occurrence statistics of the phonemes can serve as an assisting secondary task during multitask learning. We use the continuous representations obtained using PPMI factorization method mentioned in [23]. Precisely, we pose the task of obtaining a continuous representation of text as a matrix factorization problem where the matrix is populated by the co occurrence statistics and solve it using Symmetric Singular Value Decomposition. PMI(p, c) measures the association between a phone p and a context c by calculating the log of the ratio between their joint probability (the frequency in which they occur together) and their marginal probabilities (the frequency in which they occur independently). PMI can be estimated empirically by considering the actual number of observations in the corpus.

Where,

$$PMI(p,c) = log \frac{(n_p c) * (n_d)}{n_p * n_c}$$

- *NPC* is the frequency of occurrence of the phone in the corpus.
- *ND* is the size of the corpus.
- *NC* is the frequency of the occurrence of the context in the corpus.
- *NPC* is the frequency of the occurrence of the phone IN the context and appearing in the corpus.

2.3. Speaker Representation Using I Vector

An i-vector is a low-dimensional vector representing speaker identity. I-vectors have dramatically improved the performance of text-independent speaker verification and now define the state-of-the-art. Given a

speaker-dependent GMM, the corresponding mean super vector s can be represented as,

$$s = m + Ti$$

where m is the super-vector defined by the mean super-vector of a speaker-independent universal background model (UBM) that benefits from multiple speakers training corpora, s is the speaker super-vector which is the mean super-vector of the speaker dependent GMM model (adapted from the UBM), T is the total variability matrix estimated on the background data, and i is the speaker identity vector, also called the i-vector. In the current context, when training a speaker independent DNN model. As suggested in the literatures [24], length normlisation is performed on all the i-vectors. In practice, we used the ALIZE toolkit [25] to extract i-vectors.

III. IMPLEMENTATION

3.1. Database

Speech and text data for six Indian languages i) Bengali, ii) Hindi, iii) Malayalam, iv) Marathi, v) Tamil and vi) Telugu that were released as a part of Blizzard Challenge 2015 have been used. The speech data for each language was around 4 hours (sampled at 16 KHz), recorded by professional speakers in a high quality studio environment. Along with the speech data the corresponding text was provided in UTF-8 format.

3.2. Baseline systems

For comparison, we built an HMM system on the same data, employing five-state, left-to-right Hidden Semi-Markov Models (HSMM). The MCCs and BAPs with deltas and delta-deltas appended were modelled by single-component Gaussians, and log F0 with delta and delta-delta was modeled by a 3-dimension multi space probability distribution (MSD). Decision tree state clustering used a minimum description length (MDL) factor of 1.0. During parameter generation, global variance (GV) enhancement was applied. We have also built a Random Forest based system using the publicly available Festival toolkit.

3.3. Top line Systems

All the topline systems designed had the basic implementation of the DNN systems in addition to a specific secondary task(s) the systems with two secondary tasks always had speaker I vector prediction as one of the tasks. In the systems MTL+MCD and MTL+MCD+ivector, the aim was to minimize the Mel Cepstral distortion of the generated wavefile. This system was designed to see if such a metric minimization aimed training result in an acceptable output. The Systems MTL+Manifold and MTL+Manifold+iVector were designed using the minimization of the speech phoneme manifold as the secondary task. The idea behind the design of this system is to obtain the shared representation in the hidden layers such that the best approximation to a low dimensional representation of the phoneme being synthesized so that the representation can now better model the vocoder parameters at the output. The Systems MTL+Dense and MTL+Dense+iVector are similar to the Manifold systems except in the fact that the dense representation in this case is obtained from the text using the co-occurence statistics where as the manifold is obtained directly from the speech segment. The Systems MTL+Formant and MTL+Formant+iVector are designed so as to predict the second formant as the secondary task. Second formant is known to be contributing to the coarticulation [26] and therefore, the idea is to use such ability in the prediction of the speech generation parameters.

3.4. DNN System Overview

STRAIGHT [28] was used to extract 60-dimensional Mel Cepstral Coefficients (MCCs), 25 band aperiodicities (BAPs) and logarithmic fundamental frequency (log F0) at 5 MSEC frame intervals. In the DNN-based systems, the input features consisted of 592 binary features and 9 numerical features. The binary features were derived from a subset of the questions used by the decision tree clustering in the HMM system, and included linguistic contexts such as quinphone identity, and parts-of-speech positional information within the syllable, word and phrase, and so on. 9 numerical features were appended: the frame position within the HMM state and phoneme, the state position within the phoneme, and state and phoneme durations. Frame-aligned training data for the DNN was created by forced alignment using the HMM system described above.

The main task DNN outputs comprised MCCs, BAPs and continuous log F0 (all with deltas and delta-deltas) plus a voiced/unvoiced binary value. We have used a 0.2 dropout at each of the hidden layers. Input features were normalized to the range of [0.01, 0.99] and output features were normalized to zero mean and unit variance. MLPG using pre-computed variances from the training data was applied to the main task output features, and spectral enhancement post-filtering was applied to the MCCs . In both DNN and MTLDNN, the ReLu function was used as the hidden activation function, and a linear activation function was employed at the output layer.
During training L2 regularization was applied on the weights with penalty factor of 0.00001, the mini-batch size was set to 256 and momentum was used. For the first 10 epochs, momentum was 0.3 with a fixed learning rate of 0.002. After 10 epochs, the momentum was increased to 0.9 and from that point on the learning rate was halved at

each epoch. The learning rate of the top two layers was half that of other layers.

The maximum epochs was set to 25 (early stopping). For software implementation, we used Keras, a wrapper around Theano and training was conducted on a GPU.

Table1 : Objective Evaluation of the various systems Designed

Model	Bengali	Hindi	Malayalam	Marathi	Tamil	Telugu
HMM	5.83	5.61	5.73	5.23	5.91	5.63
Decison Tree	5.91	5.93	5.87	5.51	5.97	5.80
Random Forests	5.64	5.87	5.66	5.32	5.67	5.33
DNN	5.02	5.11	5.01	5.33	5.61	5.26
MTL+MCD	4.67	4.70	4.52	4.81	4.61	5.1
MTL+Manifold	4.58	4.61	4.43	4.77	4.52	4.86
MTL+Dense	4.72	4.72	4.62	4.83	4.72	4.91
MTL+Formant	4.69	4.70	4.60	4.82	4.70	4.90
MTL+MCD+iVector	4.49	4.70	4.32	4.60	4.41	4.76
MTL+Manifold+iVector	4.51	4.60	4.39	4.66	4.48	4.88
MTL+Dense+iVector	4.67	4.69	4.60	4.81	4.58	4.68
MTL+Formant+iVector	4.68	4.70	4.59	4.81	4.69	4.88

IV. EVALUATION

We conducted objective evaluation to analyze the performance of each individual system. The results are presented in Table 2. From initial observations, two conclusions can be drawn (1) Topline systems perform better than the baseline systems in terms of the objective measure and (2) There doesn't seem to be language based dependency on the technique. Across all the languages, the four top line with single secondary task, the phoneme manifold approach achieves the lowest Mel Cepstral distortions. When combining the secondary tasks with the i-vector however, the system using MCD optimization obtains the best result, indicating the positive impact of the addition of I vector as the task. There is no specific improvement in the MCD scores of the formant based system with the addition of I vector across any language. It might be interesting to combine the phoneme manifold and the dense continuous representation of the phoneme obtained form the text and use the combined vector as the secondary task in this setting. We can also use other speaker representative features such as the bottleneck features obtained from an auto associative neural network, which is known to capture the distribution of the given data in the higher dimensional feature space[27]. Eventhough its ideal to perform subjective evaluation using listeners from the native languages, we have, in the current study, performed objective evaluation of the systems designed.

V. CONCLUSION

Though DNNs are powerful in acoustic modelling, the naturalness obtained from the DNN based system is poorer than Unit Selection System. Hence, our work is in the direction of investigating if the speaker and phoneme level information as the secondary task would help in increasing the performance of the system and bring about intelligibility in the synthesized voice. Even though objective evaluation obtained was showing improvement in the system performance, we need subjective evaluation for better understanding which we could not do as there were multiple systems in multiple languages.

REFERENCES

[1] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," Speech Communication, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] S. King, "Measuring a decade of progress in text-to speech," Loquens, vol. 1, 1 2014.

[3] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on, vol. 1. IEEE, 1996, pp. 373–376.

[4] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 7962– 7966.

[5] S. Kang, X. Qian, and H.-Y. Meng, "Multi-distribution deep belief network for speech synthesis," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 8012–8016.

[6] Z. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 10, pp. 2129–2139, October 2013.

[7] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," in 8th ISCA Workshop on Speech Synthesis, Barcelona, Spain, Aug. 2013, pp. 281–285.

[8] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (dnn) for parametric TTS synthesis," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 3829–3833.

[9] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in Acoustics, Speech and Signal Processing **5** (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 3844–3848.

[10] B. Uria, I. Murray, S. Renals, C. Valentini, and J. Bridle, "Modelling acoustic feature dependencies with artificial neural networks: Trajectory-rnade," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2015, 2015.

[11] O. Watts, "From hmms to dnns: Where do the improvements come from?"

[12] K. Prahallad, A. Vadapalli, N. Elluru, G. Mantena, B. Pulugundla, P. Bhaskararao, H. Murthy, S. King, V. Karaiskos, and A. Black, "The blizzard challenge 2013–indian language task," in Blizzard Challenge Workshop 2013, 2013.

[13] R. Caruana, Learning to Learn. Boston, MA: Springer US, 1998, ch. Multitask Learning, pp. 95–133.

[14] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 4460–4464.

[15] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 6965–6969.

[16] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, April 2015, pp. 4460–4464.

[17] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings of the 25th international conference on Machine learning. ACM, 2008, pp. 160–167.

[18] V. S. Tomar and R. C. Rose, "Efficient manifold learning for speech recognition using locality sensitive hashing," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 6995–6999.

[19] C. Vaz and S. Narayanan, "Learning a speech manifold for signal subspace speech denoising," in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The kaldi speech recognition toolkit," in IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[21] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2. IEEE, 2005, pp. 1208–1213.

[22] M. Brand and M. Brand, "Charting a manifold," in Advances in Neural Information Processing Systems 15. MIT Press, 2003, pp. 961–968.

[23] S. K. Rallabandi, S. S. Rallabandi, P. Bandi, and S. V. Gangashetty, "Learning continuous representation of text for phone duration modeling in statistical parametric speech synthesis," in 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Dec 2015, pp. 111–115.

[24] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in Interspeech, 2011, pp. 249–252.

[25] A. Larcher, J. Bonastre, B. G. B. Fauve, K. Lee, C. Levy, ´ H. Li, J. S. D. Mason, and J. Parfait, "ALIZE 3.0 – open source toolkit for state-of-the-art speaker recognition," in INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013, 2013, pp. 2768–2772.

[26] D. Krull, "Second formant locus patterns and consonant vowel coarticulation in spontaneous speech," Phonetic Experimental Research at the Institute of Linguistics, University of Stockholm, vol. 10, pp. 87–108, 1989.

[27] B. Yegnanarayana and S. P. Kishore, "Aann: an alternative to GMM for pattern recognition," Neural Networks, vol. 15, no. 3, pp. 459–469, 2002.

[28] Kawahara, H,: Staright, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. Acoustical science and technology 27(6),349-353(2006)