EVALUATING DEEP SCATTERING SPECTRA WITH DEEP NEURAL NETWORKS ON LARGE SCALE SPONTANEOUS SPEECH TASK

Petr Fousek *Pierre Dognin* *Vaibhava Goel*

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

petr_fousek@cz.ibm.com, {pdognin,vgoel}@us.ibm.com

ABSTRACT

Deep Scattering Network features introduced for image processing have recently proved useful in speech recognition as an alternative to log-mel features for Deep Neural Network (DNN) acoustic models. Scattering features use wavelet decomposition directly producing log-frequency spectrograms which are robust to local time warping and provide additional information within higher order coefficients. This paper extends previous works by showing how scattering features perform on a state-of-the-art spontaneous speech recognition utilizing DNN acoustic model. We revisit feature normalization and compression topics in an extensive study, putting emphasis on comparing models of the same size. We observe that scattering features outperform baseline log-mel in all conditions, with additional gains from multi-resolution processing.

Index Terms— deep scattering networks, deep neural networks, sequence training criterion, spontaneous speech

1. INTRODUCTION

Contemporary state-of-the-art speech recognition systems often replace conventional Gaussian Mixture acoustic Models (GMM) with Deep artificial Neural Networks (DNN) for their superior performance [1]. This work contributes to a search for an optimal speech representation for DNNs by assessing potential benefits of recently proposed Deep Scattering Spectra (DSS) [2]. It extends closely related works on DSS features by Sainath [3] and Peddinti [4] by advancing from standard Broadcast News recognition task to a very large vocabulary spontaneous speech task where the acoustic model is trained under cross-entropy followed by sequence-level objectives. We revisit some feature normalization and compression experiments while putting emphasis on comparing models of the same size which was not done previously.

DNNs have been used in context of acoustic modeling mainly in two ways. Either to extract discriminative features for conventional GMMs [5] or to directly substitute GMM acoustic model, producing observation likelihoods for a decoder [6]. Earlier they had been used to complement standard cepstral GMM systems rather than to replace them [7, 8] until the recent boom of Deep Learning initiated by Hinton's publication of generative pre-training [9, 10]. Our experience following Seide [11] is that we can conveniently avoid generative pre-training by growing the DNN layer by layer under a cross-entropy criterion which we call DNN pre-training. In addition to DNN, we also considered Convolutional Neural Networks (CNN) as baseline acoustic models. However, our evidence on the task presented here suggests that with increasing amounts of data performance of DNN and CNN models converges to very similar word error rates. Since DNN is simpler and faster to train, we will use DNN acoustic model as a baseline.

When trained with multi-style data, DNN acoustic models are capable of ignoring various sources of variability in the speech that is considered irrelevant for transcription such as additive and channel noises, and speaker-related variations. To an extent, this is in contrast to GMM systems which typically benefit from explicit compensation and/or normalization techniques reducing the mentioned variabilities. In an empirical search for a good speech representation for a DNN model we have observed increasing performance when simplifying features from discriminative feature adaptation (feature-level Minimum Phone Error Transform), from speaker-specific adaptation and even from static linear transform (Discrete Cosine Transform), resulting in logarithm of melodic spectra (log-mel) coefficients. Some authors consider stepping even further back in the front-end processing pipeline, for example Sainath lets DNN optimize coefficients of a filter bank [12] but since they are widely used, in this work we will use log-mel spectra as baseline features.

In the following, we will overview scattering features, emphasizing why they have a potential to be better than log-mel features and then we will discuss specifics of modeling scattering features with DNN and show empirical evidence on a mobile search and messaging task.

2. SCATERING FEATURES

Traditional log-mel features and their linearly transformed counterpart, MFCC, are inspired by the human auditory system [13]. They were designed to preserve message-specific content, and to suppress irrelevant variability. They use short-term Fourier transform on windows in which the signal can be assumed stationary to get a representation in the spectral domain, where they drop the phase information since it is considered not useful for one-channel recognition. Human perception properties are emulated by taking the logarithm of the power spectrum, and by converting the frequency axis to a logarithmic scale. However, this processing introduces compromises. Short-term Fourier transform done on speech frames yields uniform resolution on a linear scale although we ultimately want a log-scaled axis. The logarithmic warping is achieved by binning the spectrum by a filter bank which irreversibly discards fine frequency information mainly on high frequencies. Although this information might not be necessary for recognition, the time-frequency uncertainty principle suggests we could trade the lost fine frequency information for a more detailed temporal information, allowing to accurately encode events such as attacks. Standard log-mel pipeline does not offer this but Deep Scattering Network (DSN) does.

DSN uses wavelet transform to decompose PCM signal x into frequency bands. By definition, wavelet filters (denoted ψ_{λ_1}) are uniformly distributed on a logarithmic frequency axis and their impulse reponses shrink for high-frequency filters which allows to sam-

ple higher frequency bands more densely and thus exploit more information. To produce frame-based features for DNNs, the filter bank output modula $|x * \psi_{\lambda_1}|$ are low-passed in time by a filter $\phi(t)$ and sampled uniformly yielding first-order DSS coefficients $S_1(x, \lambda_1)$ but, in contrary to log-mels, the original sub-band modula can be parameterized again by a second wavelet decomposition, $||x * \psi_{\lambda_1}| * \psi_{\lambda_2}|$ encoding all details of the temporal structure. The second-order modula again need to be low-passed $\phi(t)$ and sampled to yield sub-band second-order DSS features $S_2(t, \lambda_1, \lambda_2)$ but a recursive application of wavelet decomposition allows to preserve arbitrary level of detail. First-order coefficients provide information similar to log-mels though presumably more robust to local deformations and second-order coefficients encode sub-band modulations. Note that sub-bands differ in their respective number of non-zero second order coefficients due to different bandwidths. It is good practice to normalize higher order coefficients by the respective lower order coefficients $S_2(t, \lambda_1, \lambda_2)/S_1(t, \lambda_1)$ to make them only depend on the amplitude component of the sub-band signal. For simplicity we shall denote such normalized coefficients S_n and use them as DSS features. For more details and formal definition, see original works by Anden [2] and Bruna [14].

3. SCATTERING FEATURES AND DNN

3.1. Experimental setup

Experiments were conducted on an IBM internal US English corpus called Open Voice Search which collects mobile search queries and dictated messages. It contains 633 h of manually transcribed training data (711k utterances) plus 6.6 h of testing data (6k utterances) recorded as 16 kHz speex-compressed audio. Training data is automatically end-pointed leaving at most 250 ms of silence at the beginning and end of each utterance. Endpointed utterances have a mean duration of 3.2 s. For DNN training purposes, we leave 11.5 h (13K utterances) from the 633 h data aside ("held-out set") and use it to monitor training progress and to drive annealing. The main chunk (622 h) is used for DNN training. For fast-turnaround experiments we alternatively train on a fixed 125 h that were randomly selected from the 622 h set. So there are two possible training sets denoted **622-h** and **125-h**.

A model build starts with a discriminative pre-training where a DNN is grown layer-by-layer with Stochastic Gradient Descent (SGD) under cross-entropy criterion (CE) until the final topology is reached. Each new layer is inserted below a narrow bottle-neck layer (see below) and SGD is run on all training data (1 epoch). The final network is then retrained with SGD and CE in 20 epochs. Learning rate is halved whenever performance on the held-out set ceases to improve more than by a fixed threshold, measured once per epoch. The model is ready to be used for decoding or as a starting point for sequence-level training (ST). ST is based on implementation from [15], with a Hessian-free (HF) procedure similar to [16]. We use a modified procedure called Dynamic Stochastic Average Gradient with HF optimization (DSAG-HF) [17] that displays faster convergence. DSAG-HF splits training data into 6 random subsets and dynamically averages gradients after training on each subset. Training is run to convergence as measured by a Minimum Phone Error loss on the held-out set.

Performance is measured in terms of Word Error Rate (WER) which is influenced by a language model so we allow a weight between acoustic and language-model scores to be tuned on the test data for minimum WER for each experiment. Another measure is Phone Error Rate (PER) evaluated on the held-out set which gives average per-frame DNN classification error.

DNN model is generous in that features can be correlated and no constraints are imposed on their distributions. Therefore we append first and second-order temporal derivatives to log-mel features and then expand context by stacking ctx+1+ctx successive frames at the DNN input. Baseline features are 93-dimensional, with 31 log-mels, 31 deltas and 31 double deltas. With ctx=5 this gives (5+1+5)*93=1023-dim DNN input.

The DNN topology is [I, 5*hidN, B, O] with an Input layer, 5 hidden layers with hidN units in each with sigmoid nonlinearity, a bottle-neck layer with B=100 units and linear activation, and a softmax layer estimating posteriors of O=9000 context-dependent phone states. Targets were obtained from an existing GMM system. Baseline topology and number of model parameters are given in Table 1. DSS features use either the same topology like log-mel except for input size (*same-topo*) or we adjust hidN so that the overall number of DNN parameters is roughly constant (*same-size*). Table 2 illustrates a trend between performance and the model size on 125-h task.

training set	feature dim.	hidN	# parameters
125-h	93*11	1024	6.2M
622-h	93*11	2048	20M

Table 1. Baseline DNN dimensions and model sizes.

feature	dim.	hidN	PER	WER	# parameters
S_1 + LDA$_9$, Q4	90	1024	44.5	13.2	6.2M
S_1 + LDA$_9$, Q4	90	2048	44.3	13.0	20M
S_1 + LDA$_9$, Q4	90	4096	44.2	12.9	72M

Table 2. Illustrating performance of DNNs of various sizes. 125-h, CE training.

DSS features are defined by filter density Q (denoting a target number of filters per octave), filter type and maximum order of coefficients. We commonly use Q=1,4,8,13 and Gabor filters for S_1, and use Q=1 and Mortlett filters for S_2 features. Table 3 gives DSS feature dimensions. S_2 can be compressed by projecting onto Linear Discriminative Analysis (LDA) transform's bases [3]. A naming convention for DSS features is "S_1 + S_2, Q8" for features from Q=8 filter bank, and both S_1, S_2; "S_1 + LDA$_{26}$, Q8,4" for S_1 features from two filter banks Q=8, Q=4 appended to the 26 first LDA coefficients from concatenated S_2 features.

filter density	# S_1 features	# S_2 features
Q1	10	36
Q4	27	86
Q8	45	120
Q13	63	148

Table 3. DSS feature dimensions.

3.2. How to present DSS features to a DNN?

Log-mel features are augmented with deltas and double deltas and presented to a DNN within a 11-frame context covering 165 ms of the signal. Here we examine whether DSS S_1 are better used with

features	dim to DNN	PER	WER	hidN*
S_1 (no delta), Q8	45*11	49.6 (49.5)	13.8 (13.7)	1084
S_1 (no delta), Q8	45*15	45.0 (44.9)	13.3 (13.3)	1063
S_1, Q8	135*11	44.7 (44.8)	13.3 (13.1)	975
$S_1 + S_2$, Q8	255*11	44.9 (45.1)	13.0 (13.3)	848
S_1+LDA$_{13}$, Q8	148*11	44.4 (44.5)	13.2 (13.2)	960

* for *same-size* DNNs

Table 4. Performance of various DSS features with L2-norm. Values in parentheses are for *same-size* DNNs. 125-h task, CE training.

or without deltas, and what benefit we get from S_2 features or their LDA-compressed counterparts.

Table 4 suggests that larger context is important, presented either as deltas or as a wider window of S_1 features (experiments on other data sets favored delta features over larger context so we decided to retain deltas). S_2 features bring an additional gain but as soon as we penalize the large increase of feature dimension by shrinking hidN, the gains tend to vanish and a good compromise may be to use LDA. The number of LDA coefficients is in general chosen to lay at a knee of WER vs. LDA dimension curve, covering roughly 65% of data variance which in this case reduces the number of S_2 features from 120 to 13, significantly reducing the feature footprint. It is also relevant that LDA seems more robust to raw (non L2-normalized) input signal as shown in section 3.3.

Note that adding S_2 features to S_1 harms PER while it improves WER. Our explanation is that S_2 features add new information in a complex form which confuses the low-level DNN classifier but which imprints into DNN posteriors and is exploited by the subsequent strong DNN-HMM model in the decoder. A similar effect was discussed in [18].

3.3. Feature Normalization

DNN training can fail to converge unless data is normalized to zero mean and unit variance [19] so we *always* apply global per-feature normalization. This suffices since training batches are randomized. However, speaker-specific or feature utterance-level normalization may improve things further. Log-mel features use by default utterance-level mean normalization (uttMN), and DSS features use utterance-level L2-norm in audio signal domain (samples x_n are scaled by inverse of $\sqrt{\frac{1}{N}\sum_N x_n^2}$). We evaluate raw, uttMN and L2-norm variants on both log-mel and DSS. Table 5 shows results on 125-h, CE model. DSS with LDA-compressed S_2 features were selected because they perform consistently better than DSS S_1.

features	norm.	PER	WER
S_1 + LDA$_{13}$, Q8	L2 PCM	44.4 (44.5)	13.2 (13.2)
S_1 + LDA$_{13}$, Q8	raw	44.6 (44.6)	13.3 (13.2)
S_1 + LDA$_{13}$, Q8	uttMN	43.8 (43.9)	13.4 (13.6)
S_1 + LDA$_{13}$, Q8	uttMVN	44.2 (43.9)	13.7 (13.6)
log mel (base)	uttMN	46.2	13.7
log mel	raw	47.3	13.6
log mel	L2 PCM	46.9	13.4

Table 5. Effect of normalizations. Values in parentheses are for *same-size* DNNs across rows (DSS has hidN=960). 125-h task, CE training.

We observe that DSS perform best with L2-norm and we get almost no hit from not normalizing. Although best PERs are seen for feature-level normalizations, WER scores are worse for those. To our (recent) surprise, log-mel features display similar trend, suggesting that replacing uttMN with L2-norm could significantly improve the baseline.

Similar experiments with S_1 + S_2, Q8 features reveal that S_2 are more sensitive to L2-norm, these features give 13.0 (13.3) %WER for L2-norm but 13.4 (13.5)%WER for raw[1].

3.4. Filter Bank Resolution

In [3] we showed that for single filter bank and S_1 features, using more than Q8 (45 filters) was detrimental, with Q8 performing best among Q=1,4,8,13. Here we repeat the experiment for S_1+LDA features observing similar trends as shown in Table 6. Note that for raw signal (no L2-norm) and DSS as well as for log-mel the trends are similar (not shown here).

features	dim	PER	WER
S_1+LDA$_5$, Q1	35	51.1	16.5
S_1+LDA$_9$, Q4	90	44.5	13.2
S_1+LDA$_{13}$, Q8	148	44.4 (44.5)	13.2 (13.2)
S_1+LDA$_{21}$, Q13	210	44.8 (44.9)	13.3 (13.3)

Table 6. Effect of filter density of S_1+LDA features, L2-norm. Values in parentheses are for *same-size* DNNs. 125-h task, CE training.

3.5. Multi-Resolution Features

Multi-resolution feature extraction applies simultaneously multiple filter banks differing in spectral resolutions in the aim of extracting different complementary information. Particularly in DSS, denser filter banks offer better frequency detail while sparser banks can encode finer temporal details in higher-order coefficients. The feature vector is composed of concatenated S_1 features from all streams plus LDA coefficients from merged S_2 streams. Log-mel feature extraction has limited potential in multi-resolution processing, however for completeness we also consider multi-resolution log-mel features.

Table 7 evaluates all combinations of Q=8,4,1 resolutions. As feature dimension grows, hidN accordingly shrinks, limiting the modeling power of *same-size* DNNs. Despite the penalty, almost all multi-resolution combinations outperform single-resolution. Interestingly, multi-resolution features seem to suffer more from using raw signal. As expected, log-mel features do not display significant gains from multi-resolution processing.

3.6. Results on Full 622-h Set

Selected DSS features were evaluated on the full 622-h task. The baseline is a state-of-the-art DNN with log-mel features which has been carefully optimized for the best performance. The baseline DNN has 2048 hidden units and 20M parameters. Table 8 shows that on a cross-entropy model, DSS features outperform the baseline by about 4% relative, and that half of the gains vanish if the signal is not normalized. After sequence training the gains are smaller, 3% relative. With no normalization we even get no gains from DSS which suggests that for real-time deployments an online normalization technique is necessary.

[1] Values in parentheses are for *same-size* DNNs, hidN=848.

fea	dim	PER	WER	hidN*
log mel (uttMN)	93	46.2	13.7	1024
L2-norm on audio				
log mel	93	46.9	13.4	1024
log mel, Q8,4,1	246	46.8 (47.1)	13.3 (13.4)	857
S_1+LDA$_5$, Q1	35	51.1	16.5	
S_1+LDA$_9$, Q4	90	44.5	13.2	
S_1+LDA$_{13}$, Q8	148	44.4 (44.5)	13.2 (13.2)	960
S_1+LDA$_{11}$, Q4,1	122	44.5 (44.4)	13.1 (13.0)	990
S_1+LDA$_{15}$, Q8,1	180	44.1 (44.3)	12.8 (13.0)	925
S_1+LDA$_{28}$, Q8,4	244	44.0 (44.2)	12.6 (13.0)	859
S_1+LDA$_{26}$, Q8,4,1	272	43.8 (44.1)	12.7 (12.9)	832
No normalization				
log mel	93	47.3	13.6	1024
log mel, Q8,4,1	246	47.2 (47.5)	13.6 (13.7)	857
S_1+LDA$_5$, Q1	35	51.5	16.8	
S_1+LDA$_9$, Q4	90	44.7	13.3	
S_1+LDA$_{13}$, Q8	148	44.6 (44.6)	13.3 (13.2)	960
S_1+LDA$_{11}$, Q4,1	122	44.6 (44.6)	13.4 (13.3)	990
S_1+LDA$_{15}$, Q8,1	180	44.5 (44.2)	13.1 (13.1)	925
S_1+LDA$_{28}$, Q8,4	244	44.3 (44.2)	13.0 (13.1)	859
S_1+LDA$_{26}$, Q8,4,1	272	44.2 (44.3)	13.1 (13.1)	832

* for *same-size* DNNs

Table 7.

fea	dim	PER	WER	ST WER
log mel (uttMN)	93	39.3	11.5	10.0
L2-norm on audio				
S_1+LDA$_{26}$, Q8,4,1	272	39.5 (38.8)	11.2 (11.1)	9.7 (9.7)
S_1+LDA$_{15}$, Q8,1	180	39.1 (39.3)	11.0 (11.2)	9.8 (9.8)
No normalization				
S_1+LDA$_{26}$, Q8,4,1	272	39.5 (39.2)	11.3 (11.3)	10.0 (9.9)

Table 8. Results with selected DSS features on the full 622-h task. WER is shown for Cross-Entropy and Sequence-Trained models.

4. CONCLUSION

DSS features outperformed log-mel features across the board although by a small margin. Normalization was found to be an important factor, and it remains to be seen how signal-domain L2 normalization influences the state-of-the-art baseline. Almost no hit in performance was seen when penalizing larger feature vectors by thinner DNNs with the same number of parameters. DSS can thus be seen as a safe replacement of log-mel features in DNN systems with no run-time overhead.

5. ACKNOWLEDGMENT

We thank Steven Rennie and Tara Sainath for valuable discussions and insights.

6. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath George Dahl, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, November 2012.

[2] J. Andén and S. Mallat, "Deep scattering spectrum," *Submitted to IEEE Trans. on Signal Processing*, 2013.

[3] Tara N Sainath, Vijayaditya Peddinti, Brian Kingsbury, Petr Fousek, Bhuvana Ramabhadran, and David Nahamoo, "Deep scattering spectra with deep neural networks for lvcsr tasks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[4] Vijayaditya Peddinti, TaraN Sainath, Shay Maymon, Bhuvana Ramabhadran, David Nahamoo, and Vaibhava Goel, "Deep scattering spectrum with deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 210–214.

[5] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Connectionist feature extraction for conventional HMM systems," in *ICASSP'00*, Istanbul, Turkey, 2000.

[6] N. Morgan and H. Bourlard, "An introduction to hybrid HMM/connectionist continuous speech recognition," *IEEE Signal Processing Magazine*, pp. 25–42, May 1995.

[7] Petr Fousek, Lori Lamel, and Jean-Luc Gauvain, "On the use of mlp features forbroadcastnewstranscription," in *Text, Speech and Dialogue*, Petr Sojka, Alea Hork, Ivan Kope ek, and Karel Pala, Eds., vol. 5246 of *Lecture Notes in Computer Science*, pp. 303–310. Springer Berlin Heidelberg, 2008.

[8] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Proc. of INTERSPEECH 2004*, 2004, pp. 921–924.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, pp. 1527–1554, July 2006.

[10] Abdel rahman Mohamed, George E. Dahl, and Geoffrey E. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[11] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU 2011*. December 2011, IEEE.

[12] T.N. Sainath, B. Kingsbury, A-R. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 297–302.

[13] P. Mermelstein, "Distance measures for speech recognition: Psychological and instrumental," in *Pattern Recognition and Artificial Intelligence*, C. H. Chen, Ed., pp. 374–388. Academic Press, New York, 1976.

[14] Joan Bruna and Stéphane Mallat, "Invariant scattering convolution networks," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1872–1886, 2013.

[15] Brian Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *ICASSP*, 2009, pp. 3761–3764.

[16] James Martens, "Deep learning via hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Johannes Fürnkranz and Thorsten Joachims, Eds., Haifa, Israel, June 2010, pp. 735–742, Omnipress.

[17] Pierre L. Dognin and Vaibhava Goel, "Combining stochastic average gradient and hessian-free optimization for sequence

training of deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, 2013, pp. 321–325.

[18] Hynek Hermansky and Petr Fousek, "Multiresolution rasta filtering for tandem-based asr," in *Proc. of Interspeech 2005*, 2005, pp. 361–364.

[19] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.