MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Deep NMF for Speech Separation

Le Roux, J.; Hershey, J.R.; Weninger, F.J.

TR2015-029 April 2015

Abstract

Non-negative matrix factorization (NMF) has been widely used for challenging single-channel audio source separation tasks. However, inference in NMF-based models relies on iterative inference methods, typically formulated as multiplicative updates. We propose "deep NMF", a novel non-negative deep network architecture which results from unfolding the NMF iterations and untying its parameters. This architecture can be discriminatively trained for optimal separation performance. To optimize its non-negative parameters, we show how a new form of back-propagation, based on multiplicative updates, can be used to preserve non-negativity, without the need for constrained optimization. We show on a challenging speech separation task that deep NMF improves in terms of accuracy upon NMF and is competitive with conventional sigmoid deep neural networks, while requiring a tenth of the number of parameters.

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

DEEP NMF FOR SPEECH SEPARATION

Jonathan Le Roux[1]*, John R. Hershey*[1]*, Felix Weninger*[2]

[1]Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA
[2]Technische Universität München, 80290 Munich, Germany

ABSTRACT

Non-negative matrix factorization (NMF) has been widely used for challenging single-channel audio source separation tasks. However, inference in NMF-based models relies on iterative inference methods, typically formulated as multiplicative updates. We propose "deep NMF", a novel non-negative deep network architecture which results from unfolding the NMF iterations and untying its parameters. This architecture can be discriminatively trained for optimal separation performance. To optimize its non-negative parameters, we show how a new form of back-propagation, based on multiplicative updates, can be used to preserve non-negativity, without the need for constrained optimization. We show on a challenging speech separation task that deep NMF improves in terms of accuracy upon NMF and is competitive with conventional sigmoid deep neural networks, while requiring a tenth of the number of parameters.

Index Terms— Deep unfolding, Non-negative Matrix Factorization, Deep Neural Network, Non-negative Back-propagation

1. INTRODUCTION

Non-negative matrix factorization (NMF) [1] is a popular algorithm commonly used for challenging single-channel audio source separation tasks, such as speech separation (i.e., speech enhancement in the presence of difficult non-stationary noises such as music and other speech) [2, 3]. In this context, the basic idea is to represent the features of the sources via sets of basis functions and their activation coefficients, one set per source. Mixtures of signals are then analyzed using the concatenated sets of basis functions, and each source is reconstructed using its corresponding activations and basis set.

A fundamental issue in most NMF-based methods is that their training-time and test-time objectives differ: their parameters are optimized to best represent single sources, but at test time they are used to analyze mixtures. In particular, the training objective does not consider separation performance in the context of a mixture signal. Such optimization, termed *discriminative NMF*, is generally difficult, but recently two different approaches have been proposed [4, 5]. While [5] optimizes the original NMF bases by cleverly solving for some derivatives of the objective function, [4] proposes to circumvent the difficulty of the optimization by generalizing the original NMF model, having the added advantage of leading to a more expressive model. The key idea is to consider two separate sets of bases, one used to analyze the mixture and obtain a set of activation coefficients, and another to reconstruct a target signal using these activation coefficients. In other words, the basis parameters used at the final reconstruction step are *untied* from those used in the analysis steps.

As the analysis in NMF is generally performed using an iterative algorithm, we propose to take the parameter untying idea of [4] further, and untie the parameters not only in the reconstruction step but also in the iterations of the analysis step. Interestingly, the resulting inference algorithm can be interpreted as a novel deep network architecture with non-linear activation functions that are determined by the update equations of the NMF iterations. We call it *deep NMF*.

The concept of unfolding an iterative inference algorithm from a model-based method and untying its parameters into a deep network architecture is a very general one, called *deep unfolding*, which we recently proposed [6]. Whereas, for example, conventional sigmoid neural networks can be obtained by unfolding mean-field inference in Markov random fields, deep NMF is not only novel within the NMF literature, but it is also the first example of a novel deep network architecture obtained by deep unfolding of a model-based approach.

Main contributions: a novel non-negative deep network with non-negative parameters, derived from NMF-based source separation; a non-negative back-propagation algorithm from which one can obtain multiplicative update equations for the deep NMF parameters; finally, experiments showing the benefit of this approach in the domain of speech separation.

Relationship to the literature: there is to our knowledge no prior work on untying NMF basis parameters within the iterative inference procedure and discriminatively training them, except our discriminative NMF approach [4], which only took care of the final reconstruction layer. Various authors in the machine learning literature have considered unfolding iterative inference procedures into deep networks and discriminatively training their parameters [7], including some with applications to NMF [8, 5], but without untying the parameters, so they were in essence still within the realm of the original model.

2. DEEP NON-NEGATIVE MATRIX FACTORIZATION

NMF operates on a matrix of F-dimensional non-negative spectral features, usually the power or magnitude spectrogram of the mixture, $\mathbf{M} = [\mathbf{m}_1 \cdots \mathbf{m}_T]$, where T is the number of frames and $\mathbf{m}_t \in \mathbb{R}_+^F$, $t = 1, \ldots, T$ are obtained by short-time Fourier transformation of the time-domain signal. With L sources, each source $l \in \{1, \ldots, L\}$ is represented using a matrix containing R_l non-negative basis column vectors, $\mathbf{W}^l = \{\mathbf{w}_r^l\}_{r=1}^{R_l}$, multiplied by a matrix of activation column vectors $\mathbf{H}^l = \{\mathbf{h}_t^l\}_{t=1}^{T}$, for each time t. The rth row of $\mathbf{H}^l$ contains the activations for the corresponding basis $\mathbf{w}_r^l$ at each time t. A column-wise normalized $\widetilde{\mathbf{W}}^l$ can be used to avoid scaling indeterminacy. The basic assumptions can then be written as

$$\mathbf{M} \approx \sum_l \mathbf{S}^l \approx \sum_l \widetilde{\mathbf{W}}^l \mathbf{H}^l = \widetilde{\mathbf{W}}\mathbf{H}. \tag{1}$$

The β-divergence, D_β, is an appropriate cost function for this approximation [9], which casts inference as an optimization of $\hat{\mathbf{H}}$,

$$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} D_\beta(\mathbf{M} \mid \widetilde{\mathbf{W}}\mathbf{H}) + \mu|\mathbf{H}|_1. \tag{2}$$

For $\beta = 1$, D_β is the generalized KL divergence, and $\beta = 2$ yields the squared error. An L1 sparsity constraint with weight μ is added to favor solutions where only few basis vectors are active at a time.

The following multiplicative updates for iteration $k \in \{1, \ldots, K\}$ minimize (2) subject to non-negativity constraints [9]:

$$\mathbf{H}^k = \mathbf{H}^{k-1} \circ \frac{\widetilde{\mathbf{W}}^T(\mathbf{M} \circ (\widetilde{\mathbf{W}}\mathbf{H}^{k-1})^{\beta-2})}{\widetilde{\mathbf{W}}^T(\widetilde{\mathbf{W}}\mathbf{H}^{k-1})^{\beta-1} + \mu}, \tag{3}$$

where $\circ$ denotes element-wise multiplication, the matrix quotient is element-wise, and $\mathbf{H}^0$ is initialized randomly.

After K iterations, to reconstruct each source, typically a Wiener filtering-like approach is used, which enforces the constraint that all the source estimates $\widetilde{\mathbf{S}}^{l,K}$ sum up to the mixture:

$$\widetilde{\mathbf{S}}^{l,K} = \frac{\widetilde{\mathbf{W}}^l \mathbf{H}^{l,K}}{\sum_{l'} \widetilde{\mathbf{W}}^{l'} \mathbf{H}^{l',K}} \circ \mathbf{M}. \tag{4}$$

A commonly used approach has been to train NMF bases independently on each source, before combining them. However the combination was generally not trained for good separation performance from a mixture. Recently, discriminative methods have been applied to sparse dictionary based methods to achieve better performance in particular tasks [10]. In a similar way, we can discriminatively train NMF bases for source separation. The following optimization problem for training bases, termed *discriminative NMF* (DNMF) was proposed in [4, 5]:

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_l \gamma_l D_{\beta_2}\left(\mathbf{S}^l \mid \widetilde{\mathbf{W}}^l \hat{\mathbf{H}}^l(\mathbf{M}, \mathbf{W})\right), \tag{5}$$

$$\hat{\mathbf{H}}(\mathbf{M}, \mathbf{W}) = \arg\min_{\mathbf{H}} D_{\beta_1}(\mathbf{M} \mid \widetilde{\mathbf{W}}\mathbf{H}) + \mu|\mathbf{H}|_1, \tag{6}$$

and where β_1 controls the divergence used in the bottom-level analysis objective, and β_2 controls the divergence used in the top-level reconstruction objective. The weights γ_l account for the application-dependent importance of source l; for example, in speech de-noising, we focus on reconstructing the speech signal. The first part (5) minimizes the reconstruction error given $\hat{\mathbf{H}}$. The second part ensures that $\hat{\mathbf{H}}$ are the activations that arise from the test-time inference objective. Given the bases $\mathbf{W}$, the activations $\hat{\mathbf{H}}(\mathbf{M}, \mathbf{W})$ are uniquely determined, due to the convexity of (6). Nonetheless, the above remains a difficult bi-level optimization problem, since the bases $\mathbf{W}$ occur in both levels.

In [5] the bi-level problem was approached by directly solving for the derivatives of the lower level problem after convergence. In [4], the problem was approached by untying the bases used for reconstruction in (5) from the analysis bases used in (6), and discriminatively training only the reconstruction bases, while the analysis bases are classically trained separately on each source type. In addition, (4) was incorporated into the discriminative criteria as

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \sum_l \gamma_l D_{\beta_2}\left(\mathbf{S}^l \mid \widetilde{\mathbf{S}}^{l,K}(\mathbf{M}, \mathbf{W})\right). \tag{7}$$

Here, we propose to take this further by unfolding the entire model as a deep non-negative neural network, and untying the parameters across layers as $\mathbf{W}^k$ for $k = 0, \ldots, K$. This leads to the following architecture with $K + 1$ layers, illustrated in Fig. 1:

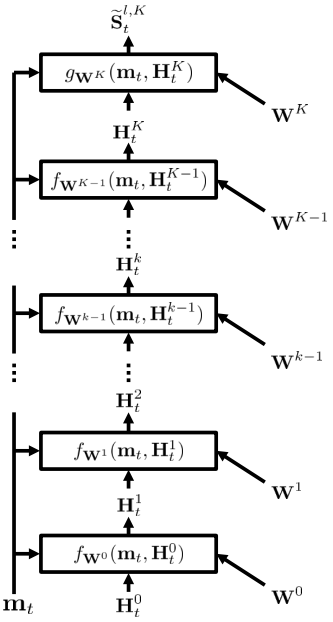

Fig. 1. Illustration of the proposed deep NMF neural network

$$\begin{aligned}\mathbf{H}_t^k &= f_{\mathbf{W^{k-1}}}(\mathbf{m}_t, \mathbf{H}_t^{k-1}), \\ &= \mathbf{H}_t^{k-1} \circ \frac{(\widetilde{\mathbf{W}}^{k-1})^T(\mathbf{m}_t \circ (\widetilde{\mathbf{W}}^{k-1}\mathbf{H}_t^{k-1})^{\beta-2})}{(\widetilde{\mathbf{W}}^{k-1})^T(\widetilde{\mathbf{W}}^{k-1}\mathbf{H}_t^{k-1})^{\beta-1} + \mu},\end{aligned} \tag{8}$$

$$\widetilde{\mathbf{S}}_t^{l,K} = g_{\mathbf{W}^K}(\mathbf{m}_t, \mathbf{H}_t^K) = \frac{\widetilde{\mathbf{W}}^{l,K}\mathbf{H}_t^{l,K}}{\sum_{l'} \widetilde{\mathbf{W}}^{l',K}\mathbf{H}_t^{l',K}} \circ \mathbf{m}_t. \tag{9}$$

We call this new model *deep NMF*.

In order to train this network while enforcing the non-negativity constraints, we derive recursively-defined multiplicative update equations by back-propagating a split between positive and negative parts of the gradient. Multiplicative updates are often derived using a heuristic approach which uses the ratio of the negative part to the positive part as a multiplication factor to update the value of that variable of interest. Here we do the same for each $\mathbf{W}^k$ matrix in the unfolded network:

$$\mathbf{W}^k \Leftarrow \mathbf{W}^k \circ \frac{[\nabla_{\mathbf{W}^k}\mathcal{E}]_-}{[\nabla_{\mathbf{W}^k}\mathcal{E}]_+}. \tag{10}$$

To propagate the positive and negative parts, we use:

$$\left[\frac{\partial \mathcal{E}}{\partial h_{r,t}^k}\right]_+ = \sum_{r'} \left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+ \left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_+ + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_- \left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_- \right)$$

$$\left[\frac{\partial \mathcal{E}}{\partial h_{r,t}^k}\right]_- = \sum_{r'} \left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+ \left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_- + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_- \left[\frac{\partial h_{r',t}^{k+1}}{\partial h_{r,t}^k}\right]_+ \right)$$

$$\left[\frac{\partial \mathcal{E}}{\partial w_{f,r}^k}\right]_+ = \sum_{t,r'} \left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+ \left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_+ + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_- \left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_- \right)$$

$$\left[\frac{\partial \mathcal{E}}{\partial w_{f,r}^k}\right]_- = \sum_{t,r'} \left(\left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_+ \left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_- + \left[\frac{\partial \mathcal{E}}{\partial h_{r',t}^{k+1}}\right]_- \left[\frac{\partial h_{r',t}^{k+1}}{\partial w_{f,r}^k}\right]_+ \right)$$

where $h_{r,t}^k$ are the activation coefficients at time t for the rth basis set in the kth layer, and $w_{f,r}^k$ are the values of the rth basis vector in the fth feature dimension in the kth layer.

3. EXPERIMENTS

The deep NMF method was evaluated along with competitive models on the 2nd CHiME Speech Separation and Recognition Challenge corpus[1] [11]. The task is speech separation in reverberated noisy mixtures ($S = 2$, $l = 1$: speech, $l = 2$: noise). The background noise, recorded in a home environment, consists of naturally occurring interference such as children, household appliances, television, radio, and so on, most of which is non-stationary. Training, development, and test sets of noisy mixtures along with noise-free reference signals are created from disjoint parts of the Wall Street Journal (WSJ-0) corpus of read speech and the noise recordings. The dry speech recordings are convolved with time-varying room impulse responses estimated from the same environment as the noise. The training set consists of 7 138 utterances at six SNRs from -6 to 9 dB, in steps of 3 dB. The development and test sets consist of 410 and 330 utterances at each of these SNRs, for a total of 2 460 / 1 980 utterances. By construction of the WSJ-0 corpus, our evaluation is speaker-independent. The background noise recordings in the development and test set are different from the training noise recordings, and different room impulse responses are used to convolve the dry utterances. In this paper, we present results on the development set. To reduce complexity we use only 10 % of the training utterances for all methods. Our evaluation measure for speech separation is source-to-distortion ratio (SDR) [12].

3.1. Feature extraction

Each feature vector concatenates $T = 9$ consecutive frames of left context, ending with the target frame, obtained as short-time Fourier spectral magnitudes, using 25 ms window size, 10 ms window shift, and the square root of the Hann window. This leads to feature vectors of size TF where $F = 200$ is the number of frequencies. Similarly to the features in $\mathbf{M}$, each column of $\hat{\mathbf{S}}^l$ corresponds to a sliding window of consecutive reconstructed frames. Only the last frame in each sliding window is reconstructed, which leads to an on-line algorithm. For the NMF-based approaches, we use the same number of basis vectors for speech and noise ($R^1 = R^2$), and consider $R^l = 100$ and $R^l = 1000$. We denote the total as $R = \sum_l R^l$. We investigate two regimes in the total number of iterations, $K = 4$ for which NMF-based approaches still have significant room for improvement in performance, and $K = 25$ for which, based on preliminary experiments, they are close to asymptotic performance.

3.2. Baseline 1: Deep Neural Network

To compare our deep NMF architecture with standard K-layer deep neural networks, we used the following setting. The feed-forward DNNs have $K - 1$ hidden layers with hyperbolic tangent activation functions and an output layer with logistic activation functions. Denoting the output layer activations for time index t by $\mathbf{y}_t = (y_{1,t}, \ldots, y_{F,t})^T \in [0, 1]^F$, the DNN computes the deterministic function

$$\mathbf{y}_t = \sigma(\mathbf{W}^K \tanh(\mathbf{W}^{K-1} \cdots \tanh(\mathbf{W}^1 \mathbf{x}_t)) \cdots),$$

where $\mathbf{x}_t$ are the input feature vectors and σ and $\tanh$ denote element-wise logistic and hyperbolic tangent functions. As in the deep NMF experiments, $T = 9$ consecutive frames of context are concatenated together, but here the vectors $\mathbf{x}_t$ are logarithmic magnitude spectra. Thus, the only difference in the input feature representation with respect to deep NMF is the compression of the spectral amplitudes, which is generally considered useful in speech processing, but breaks the linearity assumption of NMF.

Previous attempts with DNNs have focused on direct estimation of the clean speech without taking into account the mixture in the output layer, or on direct estimation of a masking function without considering its effect upon the speech estimate. Here, based on our experience with model-based approaches, we train the masking function such that, when applied to the mixture, it best reconstructs the clean speech, which was also proposed in [13]. This amounts to optimizing the following objective function for the DNN training:

$$E = \sum_{f,t} \left(y_{f,t} m_{f,t} - s^l_{f,t} \right)^2 = \sum_{f,t} \left(\tilde{s}^l_{f,t} - s^l_{f,t} \right)^2, \qquad (11)$$

where m are the mixture magnitudes and s^l are the speech magnitudes. Thus, the sequence of output layer activations $\mathbf{y}_t$ can be interpreted as a time-frequency mask in the magnitude spectral domain, similar to the 'Wiener filter' in the output layer of deep NMF (7). This approach, which we refer to as signal approximation, leads in our experiments to 1.5 dB improvements relative to mask approximation, in which the masking function is trained to match a target mask (mask approximation results are not reported here). Although this comes from the model-based approach, we include it here so that the DNN results are comparable solely on the context of the deep architecture and not the output layer.

Our implementation is based on the open-source software CURRENNT[2]. During training, the above objective function is minimized on the CHiME training set, using back-propagation, stochastic gradient descent with momentum, and discriminative layer-wise pre-training. Early stopping based on cross-validation with the CHiME development set, and Gaussian input noise (standard deviation 0.1) are used to prevent aggressive over-optimization on the training set. Unfortunately, our current experiments for deep NMF do not use cross-validation, but despite the advantage this gives to the DNN, as shown below deep NMF nevertheless performs better.

We investigate different DNN topologies (number of layers and number of hidden units per layer) in terms of SDR performance on the CHiME development set. Results are shown in Table 1.

Table 1. DNN source separation performance on the CHiME development set for various topologies.

SDR [dB]	Input SNR [dB]							
Topology	-6	-3	0	3	6	9	Avg.	# params
3x256	3.71	5.78	7.71	9.08	10.80	12.75	8.31	644 K
1x1024	5.10	7.12	8.84	10.13	11.80	13.58	9.43	2.0 M
2x1024	5.14	7.18	8.87	10.20	11.85	13.66	9.48	3.1 M
3x1024	4.75	6.74	8.47	9.81	11.53	13.38	9.11	4.1 M
2x1536	5.42	7.26	8.95	10.21	11.88	13.67	9.57	5.5 M

3.3. Baseline 2: sparse NMF

Sparse NMF (SNMF) [14] is used as a baseline, by optimizing the training objective,

$$\overline{\mathbf{W}}^l, \overline{\mathbf{H}}^l = \arg\min_{\mathbf{W}^l, \mathbf{H}^l} D_\beta(\mathbf{S}^l \mid \widetilde{\mathbf{W}}^l \mathbf{H}^l) + \mu |\mathbf{H}^l|_1, \qquad (12)$$

for each source, l. A multiplicative update algorithm to optimize (12) for arbitrary $\beta \geq 0$ is given by [15]. During training, we set $\mathbf{S}^1$ and $\mathbf{S}^2$ in (12) to the spectrograms of the concatenated noise-free CHiME training set and the corresponding background noise in the multi-condition training set. This yields SNMF bases $\overline{\mathbf{W}}^l$, $l = 1, 2$.

[1] http://spandh.dcs.shef.ac.uk/chime_challenge/ – as of June 2014

[2] http://currennt.sf.net/

Table 2. Deep NMF source separation performance on CHiME Challenge (WSJ-0) development set.

SDR [dB]	Input SNR [dB]								
$R^l = 100$	-6	-3	0	3	6	9	Avg.	P_D	P
$K = 4,\ C = 0$ (SNMF)	2.03	4.66	7.08	8.76	10.67	12.74	7.66	-	360 K
$K = 4,\ C = 1$ (DNMF)	2.91	5.43	7.57	9.12	10.97	13.02	8.17	40 K	400 K
$K = 4,\ C = 2$	3.19	5.68	7.78	9.28	11.09	13.07	8.35	80 K	440 K
$K = 4,\ C = 3$	3.22	5.69	7.79	9.28	11.09	13.05	8.35	120 K	480 K
$K = 4,\ C = 4$	3.32	5.76	7.84	9.31	11.11	13.05	8.40	160 K	520 K
$K = 25, C = 0$ (SNMF)	4.16	6.46	8.51	9.90	11.61	13.40	9.01	-	360 K
$K = 25, C = 1$ (DNMF)	4.92	7.09	8.90	10.24	12.02	13.83	9.50	40 K	400 K
$K = 25, C = 2$	5.16	7.28	9.05	10.36	12.12	13.89	9.64	80 K	440 K
$K = 25, C = 3$	5.30	7.38	9.14	10.43	12.18	13.93	9.73	120 K	480 K
$K = 25, C = 4$	5.39	7.44	9.19	10.48	12.22	13.95	9.78	160 K	520 K
$R^l = 1000$	-6	-3	0	3	6	9	Avg.	P_D	P
$K = 4,\ C = 0$ (SNMF)	1.79	4.45	6.94	8.66	10.61	12.76	7.54	-	3.6 M
$K = 4,\ C = 1$ (DNMF)	2.94	5.45	7.60	9.15	11.00	13.06	8.20	400 K	4 M
$K = 4,\ C = 2$	3.14	5.62	7.74	9.26	11.10	13.12	8.33	800 K	4.4 M
$K = 4,\ C = 3$	3.36	5.80	7.89	9.37	11.19	13.18	8.47	1.2 M	4.8 M
$K = 4,\ C = 4$	3.55	5.95	8.01	9.48	11.28	13.23	8.58	1.6 M	5.2 M
$K = 25, C = 0$ (SNMF)	4.39	6.60	8.67	10.06	11.82	13.67	9.20	-	3.6 M
$K = 25, C = 1$ (DNMF)	5.74	7.75	9.55	10.82	12.55	14.35	10.13	400 K	4 M
$K = 25, C = 2$	5.80	7.80	9.59	10.86	12.59	14.39	10.17	800 K	4.4 M
$K = 25, C = 3$	5.84	7.82	9.62	10.89	12.61	14.40	10.20	1.2 M	4.8 M

As initial solution for $\overline{\mathbf{W}}$, we use exemplar bases sampled at random from the training data for each source. For the sparsity weight we use $\mu = 5$, which performed well for SNMF and DNMF algorithms for both $R^l = 100$ and $R^l = 1000$ in the experiments of [4]. In the SNMF experiments, the same basis matrix $\overline{\mathbf{W}}$ is used both for determining $\hat{\mathbf{H}}$ according to (2) and for reconstruction using (4).

3.4. Deep NMF

In the deep NMF experiments, the KL divergence ($\beta_1 = 1$) is used for the update equations (i.e., in layers $k = 1, \ldots, K-1$), but we use the squared error ($\beta_2 = 2$) in the discriminative objective (7) (i.e., in the top layer $k = K$) since this corresponds closely to the SDR evaluation metric, and this combination performed well in [4]. In all the deep NMF models we initialize the basis sets for all layers using the SNMF bases, $\overline{\mathbf{W}}$, trained as described in Section 3.3. We then consider the C last layers to be *discriminatively trained*, for various values of C. This means that we untie the bases for the final C layers (counting the reconstruction layer and analysis layers), and train the bases $\mathbf{W}^k$ for k such that $K-C+1 \leq k \leq K$ using the multiplicative back-propagation updates described in Section 2. Thus $C = 0$ corresponds to SNMF, $C \geq 1$ corresponds to deep NMF, with the special case $C = 1$ previously described as DNMF [4]. While all layers could be discriminatively trained, as in the general framework of Section 2, we here focus on a few last layers to investigate the influence of discriminatively training more and more layers.

In the experiments, the $K - C$ non-discriminatively trained layers use the full bases $\overline{\mathbf{W}}$, which contain multiple context frames. In contrast the C discriminatively trained layers are restricted to a single frame of context. This has the advantage of dramatically reducing the number of parameters, and is motivated by the fact that the network is being trained to reconstruct a single target frame, whereas using the full context in $\mathbf{W}^k$ and $\mathbf{M}$ would enforce the additivity constraints across reconstructions of the full context in each layer. Here, $\mathbf{W}^{k>K-C}$ is thus of size $(F \times R)$, and is initialized to the last F rows of $\overline{\mathbf{W}}$ (those corresponding to the features of the current frame), and the matrix $\mathbf{M}'$, consisting of the last F rows of $\overline{\mathbf{W}}$, is used in place of $\mathbf{M}$. For deep NMF, the fixed basis functions $\overline{\mathbf{W}}$ contain $D_{\mathrm{F}} = TFR$ parameters that are not discriminatively trained, whereas the final C layers together have $P_{\mathrm{D}} = CFR$ discriminatively trained parameters, for a total of $P = (T + C)FR$.

4. DISCUSSION

Results in terms of SDR are shown for the experiments using DNNs in Table 1, and for the deep NMF family in Table 2, for a range of topologies. The first thing to note is that the deep NMF framework yields strong improvements relative to SNMF. Comparing the DNN and deep NMF approaches, we can first see that the best deep NMF topology achieves an SDR of 10.20 dB, outperforming the best DNN result of 9.57 dB, for a comparable number of parameters (4.8M for deep NMF versus 5.5M for the DNN). The smallest deep NMF topology that outperforms the best DNN topology, is obtained for $R^l = 100, K = 25, C = 2$, and achieves an SDR of 9.64 dB using at least an order of magnitude fewer parameters (only 440K parameters overall, only 80K of which are discriminatively trained).

For deep NMF, discriminatively training the first layer gives the most improvement, but training more and more layers consistently improves performance, especially in low SNR conditions, while only modestly increasing the parameter size. Increasing the parameter size from $R^l = 100$ to $R^l = 1000$ does not lead to as much gain as one might expect. This may be because we are currently only training on 10 % of the data, and used a conservative convergence criterion. For the same model size, using $K = 25$ layers leads to large gains in performance without increasing training time and complexity. However, it comes at the price of increased computational cost at inference time. Intermediate topology regimes need to be further explored to better understand the speed/accuracy trade-off.

In subsequent experiments on DNNs, improved features and training procedures brought the best DNN performance to 10.46 dB with 4.1M parameters [16]. Application of these improvements to deep NMF is indicated so that the two methods can be compared on an equal footing. In [16] recurrent networks further improved performance on the same task to 12.23 dB. Future work on deep NMF should therefore also focus on developing recurrent extensions.

5. REFERENCES

[1] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001.

[2] C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, "Real-time speech separation by semi-supervised nonnegative matrix factorization," in *Proc. of LVA/ICA*, Mar. 2012.

[3] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Proc. of Interspeech*, 2010.

[4] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in *Proc. of ISCA Interspeech*, Sep. 2014.

[5] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Supervised non-euclidean sparse NMF via bilevel optimization with applications to speech enhancement," in *HSCMA*, May 2014.

[6] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," MERL - Mitsubishi Electric Research Laboratories, Tech. Rep. TR2014-117, Aug. 2014.

[7] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *ICML*, 2010.

[8] T. B. Yakar, R. Litman, P. Sprechmann, A. Bronstein, and G. Sapiro, "Bilevel sparse models for polyphonic music transcription," in *ISMIR*, Nov. 2013.

[9] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis," *Neural Computation*, vol. 21, no. 3, March 2009.

[10] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, 2012.

[11] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. of ICASSP*, 2013.

[12] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, Jul. 2006.

[13] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *ICASSP*, May 2014.

[14] J. Eggert and E. Körner, "Sparse coding and NMF," in *Proc. Neural Networks*, vol. 4, 2004.

[15] P. D. O'Grady and B. A. Pearlmutter, "Discovering convolutive speech phones using sparseness and non-negativity," in *Proc. ICA*, 2007.

[16] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE GlobalSIP Symposium on Machine Learning Applications in Speech Processing*, 2014.