Sparse and Transformation-Invariant Hierarchical NMF

Sven Rebhan[1], Julian Eggert[1], Horst-Michael Groß[2], and Edgar Körner[1]

[1] HONDA Research Institute Europe GmbH
Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany
[2] Ilmenau Technical University, Dept. of Neuroinformatics and Cognitive Robotics
P.O.B. 100565, 98684 Ilmenau, Germany

Abstract. The hierarchical non-negative matrix factorization (HNMF) is a multilayer generative network for decomposing strictly positive data into strictly positive activations and base vectors in a hierarchical manner. However, the standard hierarchical NMF is not suited for overcomplete representations and does not code efficiently for transformations in the input data. Therefore we extend the standard HNMF by sparsity conditions and transformation-invariance in a natural, straightforward way. The idea is to factorize the input data into several hierarchical layers of activations, base vectors and transformations under sparsity constraints, leading to a less redundant and sparse encoding of the input data.

1 Introduction

The NMF has been introduced by Lee and Seung [1,2] as an efficient factorization method for decomposing multivariant data under the constraint of non-negativity. This results in a parts-based representation, because it allows only additive combination of components. While the standard NMF makes no further assumption on the input data, even so it is often used on inputs containing particular transformation properties, e.g. input images presented at different positions, scales and rotations. The resulting base vectors of the standard NMF then encode each transformation implicitly, which leads to a large amount of redundancy in the base vectors.

Ahn et al. developed a hierarchical multilayered variant of the NMF [3] by stacking multiple layers of NMF networks. This way a hierarchical representation of the input can be learned, where higher layers of the hierarchy code for more complex features composed of less complex features from lower layers. One can interprete this as a more and more abstract representation of the input with increasing hierarchy levels. Despite the interesting property of increasing abstraction in the layers of the network, the main problem of the NMF, the implicit coding of transformations in the base vectors, remains.

By assuming transformation properties in the input data we introduce, based on the work of Eggert et al. [4,5], a hierarchical, sparse and transformation-invariant version of the NMF. It has been shown that the proposed separation of the input data into activations, base vectors and transformations leads to

J. Marques de Sá et al. (Eds.): ICANN 2007, Part I, LNCS 4668, pp. 894–903, 2007.

a sparser and less redundant representation of the input than in the standard NMF. Extending the approach of Eggert et al. in a hierarchical way, we combine the advantage of sparsity and reduced redundancy in the representation with the advantage of growing abstraction in the hierarchical network of Ahn et al.

In Sect. 2 we extend the energy term formulations of [4,5] and derive the update rules for the activations and base vectors. Afterwards we discuss two possible update schemes in Sect. 2.3 and finally present simulation results using the new algorithm in Sect. 3. A short discussion finalizes the paper.

2 Hierarchical Extension to the Sparse, Transformation-Invariant NMF Framework

2.1 Sparse and Transformation-Invariant NMF

First we look at the sparse and transformation-invariant NMF, which serves as the base for our hierarchical extension. The energy term of the sparse and transformation-invariant NMF is defined as the Euclidian distance between the i-th input $\mathbf{V}_i$ and its reconstruction $\mathbf{R}_i$ plus the sparsity term $\lambda_H \cdot g_H(H)$

$$F(H, \hat{W}) = \frac{1}{2} \sum_i ||\mathbf{V}_i - \mathbf{R}_i||^2 + \lambda_H \cdot g_H(H) \tag{1}$$

where $g_H(H)$ is a sparsity function and λ_H is used to control the sparsity in the activations H_i. The reconstruction $\mathbf{R}_i$ itself is gained by linearly overlapping the normalized base vectors $\hat{\mathbf{W}}_j$ transformed by the operators T^m, weighted by the activation $H_i^{j,m}$

$$\mathbf{R}_i = \sum_j \sum_m H_i^{j,m} T^m \hat{\mathbf{W}}_j \ . \tag{2}$$

To avoid the scaling problem described in [4] the base vectors have to be normalized. Now one can calculate the derivation of the energy function with respect to $H_i^{j,m}$ and $\mathbf{W}_j$ and update them according to the standard NMF update rules:

1. Calculate the reconstruction $\mathbf{R}_i$ according to (2).
2. Update the activations according to [1]

$$H_i^{j,m} \leftarrow H_i^{j,m} \odot \frac{\left(T^m \hat{\mathbf{W}}_j\right)^T \mathbf{V}_i}{\left(T^m \hat{\mathbf{W}}_j\right)^T \mathbf{R}_i + \lambda_H \cdot g'_H\left(H_i^{j,m}\right)} \ . \tag{3}$$

3. Calculate the reconstruction $\mathbf{R}_i$ using the new activations according to (2).
4. Update the non-transformed base vectors according to

$$\mathbf{W}_j \leftarrow \mathbf{W}_j \odot \frac{\sum_m \left[(T^m)^T V \left(\mathbf{H}^{j,m}\right)^T + \left[\left(\hat{\mathbf{W}}_j\right)^T (T^m)^T R \left(\mathbf{H}^{j,m}\right)^T \right] \nabla_{\mathbf{W}_j} \left(\hat{\mathbf{W}}_j\right) \right]}{\sum_m \left[(T^m)^T R \left(\mathbf{H}^{j,m}\right)^T + \left[\left(\hat{\mathbf{W}}_j\right)^T (T^m)^T V \left(\mathbf{H}^{j,m}\right)^T \right] \nabla_{\mathbf{W}_j} \left(\hat{\mathbf{W}}_j\right) \right]} \ . \tag{4}$$

[1] Where $\odot$ denotes componentwise multiplication as $\mathbf{C} = \mathbf{A} \odot \mathbf{B} := C_i = A_i \cdot B_i, \ \forall i$.

5. Return to 1 until convergence.

The terms for updating the activations and base vectors can be found in [4], in addition the transformation matrix T^m from [5] is already included to make the sparse NMF transformation-invariant. Based on the update rules above we formulate the hierarchical energy equation and the corresponding update rules.

2.2 Sparse and Transformation-Invariant Hierarchical NMF

The sparse and transformation-invariant HNMF can be seen, similar as suggested in [3], as a network composed of multiple sparse and transformation-invariant NMF layers. This leads to the following Euclidian energy formulation[2]:

$$F(H^{(L)}, W^{(1)}, \ldots, W^{(L)}) = \frac{1}{2} \sum_i \left|\left| \mathbf{V}_i - R_i^{(1)} \right|\right|^2 , \tag{5}$$

where L denotes the topmost layer of the network. The reconstruction $R^{(l)}$ of the layer l in the hierarchy serves as the activation of the layer (l-1) and is calculated according to the transformation-invariant NMF

$$\mathbf{R}_i^{(l),m_{l-1}} := \mathbf{H}_i^{(l-1),m_{l-1}} = \sum_{m_l} \mathbf{H}_i^{(l),m_l} T^{(l),m_l} W^{(l),m_{l-1}} . \tag{6}$$

This recursive definition reveals that the reconstruction of the input data depends only on the highest layer activations $H^{(L)}$ and all base vectors. Having a closer look at (6) we see that the transformation information m_{l-1} is propagated down the hierarchy.

As [5] shows, sparsity is absolutely necessary in a transformation-invariant NMF network to avoid trivial solutions for the base vectors. Therefore we have to include at least sparsity in the activations. In contrast to the activations of all other layers, which are defined through down-propagation, the activations $H^{(L)}$ are independent. The extended energy formulation reads as

$$F = \frac{1}{2} \sum_i \left|\left| \mathbf{V}_i - R_i^{(1)} \right|\right|^2 + \lambda_H \cdot g_H \left(H^{(L)} \right) . \tag{7}$$

This step additionally requires the normalization of all base vectors $\hat{W}^{(1)}$, $\ldots, \hat{W}^{(L)}$. We choose the normalization function for the base vectors as

$$\hat{\mathbf{W}}_{j_l}^{(l),m_{l-1}} = \frac{\mathbf{W}_{j_l}^{(l),m_{l-1}}}{\sum_a \sum_b W_{j_l}^{(l),a,b}} , \tag{8}$$

which normalizes the length of each base vector to one. This leads to

$$\mathbf{R}_i^{(l),m_{l-1}} := \mathbf{H}_i^{(l-1),m_{l-1}} = \sum_{m_l} \mathbf{H}_i^{(l),m_l} T^{(l),m_l} \hat{W}^{(l),m_{l-1}} . \tag{9}$$

[2] Denoted as F from now on for convenience.

To be able to control the arrangement in the base vectors it is useful to include sparsity in the base vectors as well. Therefore we add another sparsity term to the energy function, which is now composed of three elements

$$F = \frac{1}{2} \underbrace{\sum_i \left|\left| \mathbf{V}_i - R_i^{(1)} \right|\right|^2}_{\text{Reconstruction error}} + \underbrace{\lambda_H \cdot g_H\left(H^{(L)}\right)}_{\text{Activation sparsity}} + \underbrace{\sum_l \lambda_W^{(l)} \cdot g_W\left(\hat{W}^{(l)}\right)}_{\text{Base vector sparsity}} . \quad (10)$$

The sparsity terms for the activations and base vectors are chosen as

$$g_H\left(H^{(L)}\right) = \sum_i \sum_{j_L} \sum_{m_L} H_i^{(L),j_L,m_L} \quad (11)$$

$$g_W\left(\hat{W}^{(l)}\right) = \sum_{j_l} \sum_{j_{l-1}} \sum_{m_{l-1}} \hat{W}_{j_l}^{(l),j_{l-1},m_{l-1}} . \quad (12)$$

Starting from the functions above we calculate the gradients with respect to the activations $H^{(L)}$ and base vectors $W^{(l)}$.

In order to formulate the multiplicative update rule we split the remaining two gradient terms into the positive part ∇^+ and the negative part ∇^-.[3] For the highest layer activations we get the following update rule

$$\mathbf{H}_i^{(L),m_L} \leftarrow \mathbf{H}_i^{(L),m_L} \odot \frac{\sum_{m_{L-1}} \left(T^{(L),m_L} \hat{W}^{(L),m_{L-1}}\right)^T \mathbf{V}_i^{(L),m_{L-1}}}{\sum_{m_{L-1}} \left(T^{(L),m_L} \hat{W}^{(L),m_{L-1}}\right)^T \mathbf{R}_i^{(L),m_{L-1}} + \lambda_H} \quad (13)$$

with the substitutions

$$\mathbf{V}_i^{(l+1),m_l} = \sum_{m_{l-1}} \left(T^{(l),m_l} \hat{W}^{(l),m_{l-1}}\right)^T \mathbf{V}_i^{(l),m_{l-1}} \quad (14)$$

$$\mathbf{R}_i^{(l+1),m_l} = \sum_{m_{l-1}} \left(T^{(l),m_l} \hat{W}^{(l),m_{l-1}}\right)^T \mathbf{R}_i^{(l),m_{l-1}} . \quad (15)$$

The reconstruction error is propagated from the bottom to the top layer of the hierarchy and is then used to adjust the activations of the highest layer. The sparsity term itself is an additional constraint which is independent of the reconstruction quality of the network. As a consequence the sparse and transformation-invariant HNMF network has to find a tradeoff between reconstruction quality and sparsity, controlled by the sparsity parameter λ_H.

Performing the same steps for the gradient with respect to $W^{(l)}$, we get

$$\mathbf{W}_{j_l}^{(l),m_{l-1}} \leftarrow \mathbf{W}_{j_l}^{(l),m_{l-1}} \odot \frac{\nabla^-_{W^{(l)}} F}{\nabla^+_{W^{(l)}} F} . \quad (16)$$

[3] This is possible due to the non-negative character of all elements in the equation.

The gradient of the sparsity term $g_H\left(H^{(L)}\right)$ for the highest layer activations is zero, because $H^{(L)}$ is independent of the base vectors. For the two parts of the gradient of (16) we get

$$\nabla^{-}_{W^{(l)}} F = \mathbf{W_V}{}^{(l),m_{l-1}}_{j_l} + \sum_k \left[\left(\hat{\mathbf{W}}^{(l),k}_{j_l} \right)^T \left[\mathbf{W_R}{}^{(l),k}_{j_l} + \lambda^{(l)}_W \right] \right] \hat{\mathbf{W}}^{(l),m_{l-1}}_{j_l} \quad (17)$$

$$\nabla^{+}_{W^{(l)}} F = \mathbf{W_R}{}^{(l),m_{l-1}}_{j_l} + \sum_k \left[\left(\hat{\mathbf{W}}^{(l),k}_{j_l} \right)^T \mathbf{W_V}{}^{(l),k}_{j_l} \right] \hat{\mathbf{W}}^{(l),m_{l-1}}_{j_l} + \lambda^{(l)}_W \quad (18)$$

with the substitutions

$$\mathbf{W_R}{}^{(l),m_{l-1}}_{j_l} = \sum_{m_l} \left[\left(T^{(l),m_l} \right)^T R^{(l),m_{l-1}} \left(\mathbf{H}^{(l),j_l,m_l} \right)^T \right] \quad (19)$$

$$\mathbf{W_V}{}^{(l),m_{l-1}}_{j_l} = \sum_{m_l} \left[\left(T^{(l),m_l} \right)^T V^{(l),m_{l-1}} \left(\mathbf{H}^{(l),j_l,m_l} \right)^T \right] . \quad (20)$$

Similar to the substitutions for the activations we see an upwards propagation of the reconstruction error. The sparsity constraint on the base vectors, controlled by the parameter $\lambda^{(l)}_W$, can also be seen in the update function (17) and (18). The normalization of the base vectors which is required by the sparsity in the activations leads to an additional term in the update rule for $W^{(l)}$.

In the next section we discuss two possible update schemes, starting from the update rules (13) to (20).

2.3 Possible Update Schemes

With the extension of the NMF to a hierarchical network new possibilities to update the whole network come along.

1. Update the network by iterating layer by layer
2. Update the whole network by propagating through all layers

In the following we discuss the pro and contra of the two methods.

Update Network Layer by Layer

In this scheme each layer is learned separately, beginning from the lowest layer, as shown in Fig. 1a. The goal is to reconstruct the activations of the layer below as a linear combination of base vectors. Afterwards the layer above is adapted and so on. This means that the activations of each layer are adapted sequentially and are therefore mutually independent. In this sense you get a stack of separate transformation-invariant, sparse NMF networks. A big advantage of this independent relaxation is the fact that all parameters of one layer are independent from the parameters of the other layers, which makes the parameter setting much easier. Another advantage is the extensibility of the framework. After the convergence of the network you can add another layer on top of the

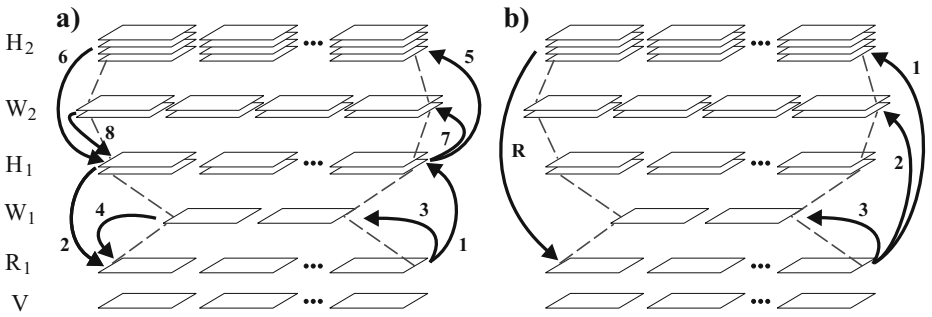

Fig. 1. The graphics above show two possible update sequences for the HNMF.
a) In this scheme each layer is iterated independently. First 1-4 is iterated until convergence, then 5-8 is iterated, trying to reconstruct H_1.
b) In this scheme the whole network is iterated in a combined manner by first calculating 1 followed by the reconstruction R, then 2, the reconstruction R, 3 and finally the reconstruction R until convergence.

existing layers and learn the new one. All other layers can be left untouched. The big disadvantage is that each layer only minimizes the local energy function on the activations of the lower layer. This leads to a smaller number of minima in the energy function, but has the disadvantage that in most cases the global minimum of the whole network is not found (see [3]).

Update Network as a Whole
Contrary to the independent relaxation, in this scheme we learn the base vectors and the highest layer activations simultaneously as shown in Fig. 1b. The sequence of updating is the following:

1. Calculate the reconstruction of the lowest layer by propagating down the highest layer activations through the base vectors by applying (9) iteratively.
2. Propagate the reconstruction error between $R^{(1)}$ and the input $V^{(1)}$ to the highest layer using (14) and (15).
3. Adapt the activations of the highest layer as described in (13).
4. Execute step 1 using the updated activations $H^{(L)}$.
5. Adapt the base vectors of the lowest layer $W^{(1)}$ using (16).
6. Execute step 1 using the updated base vectors.
7. Propagate the reconstruction error between $R^{(1)}$ and the input $V^{(1)}$ to the next higher layer using (14) and (15).
8. Adapt the base vectors of the next higher layer $W^{(l+1)}$ using (16).
9. Repeat step 6 to 8 until the base vectors of all layers are updated.
10. Repeat beginning with step 1 until convergence.

The advantage of this combined relaxation is the minimization of the overall energy function, which leads to a better reconstruction and a sparser representation. One drawback is the introduction of relations between the sparsity

parameters by combined relaxation, which makes the selection of the parameters more difficult. Because of the better reconstruction results, we choose the combined update scheme for the experiments we present in the next section.

3 Results

For the following experiments we set up a two layer, sparse and **translation-invariant** hierarchical network. We use only translation for $T^{(l)}$ because this transformation can be coded very efficiently using correlations. The learning of the base vectors is performed with the combined relaxation scheme discussed in Sect. 2.3. The used dataset (see examples in Fig. 2) consists of 162 bar images of 4x4 pixel size. Each of the images is a superposition of up to four horizontal and vertical bars. The horizontal bar can be applied at four different horizontal positions; the vertical bar at four different vertical positions. A complete overlap of two bars is not allowed.

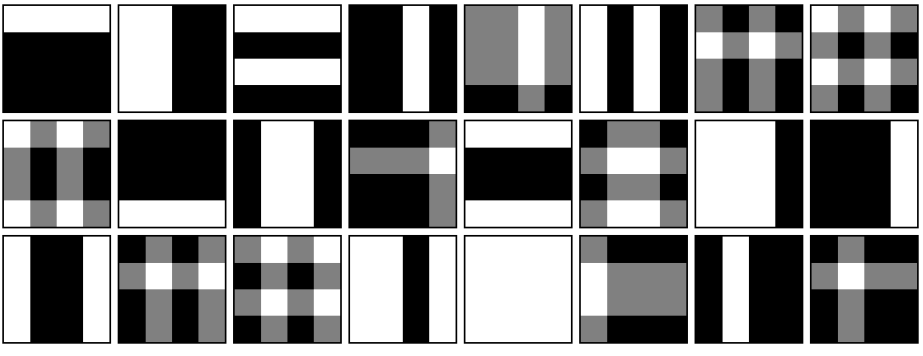

Fig. 2. These are 24 examples for the input dataset, which consists of 162 images. Each image has a size of 4x4 pixel and is a superposition of horizontal and vertical bars.

The task for the network is to find a set of base vectors that encodes the input under a given sparsity constraint. In Fig. 3 the two lower layer base vectors of the translation-invariant, sparse HNMF network are shown. One can see that the network finds the two original bars (one horizontal, one vertical). Based on these vectors, the 64 upper layer base vectors compose more complex structures in order to satisfy the sparsity constraint.

Figure 4 shows the base vectors of the upper layer projected to the input space. The vectors themselves consist of very sparse, sporadic peaks. By increasing the sparsity constraint in the activations, the base vectors get more and more complex, whereas the sparsity in the base vectors leads to a reallocation of the information between the different layers. As a result, the sparsity settings for the network are essential to force a meaningful distribution of the information within the hierarchy.

Fig. 3. Here the two base vectors of the lower layer, which are nearly perfect reconstructions of the original vectors, are shown

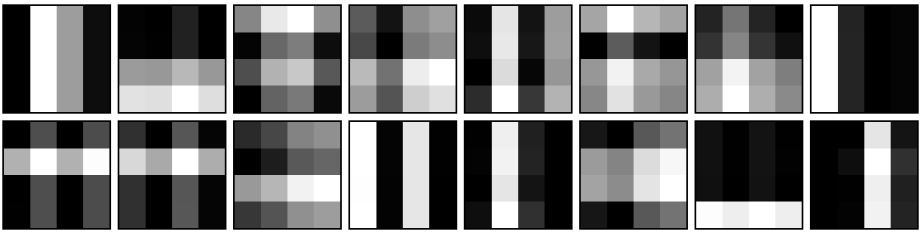

Fig. 4. This shows the 16 resulting base vectors of the upper layer projected into the input space. As you can see the set also contains the two vectors of the layer below.

If we have a look at the energy function depicted in Fig. 5 we see three phases in the relaxation process. In the first phase, which comprises the steps 0 to 380, the overall energy decreases very slowly. This is mainly a result of the reconstruction error optimization done by the network. As one can see clearly the minimization of the reconstruction error is taking place in the lower base vectors, because in this phase the higher layer vectors are not changed significantly. In the second phase, including the steps 380 to 600, a minimization of the sparsity penalization is taking place, where large changes in the higher layer base vectors can be seen. We can observe a reorganization in the HNMF network, where the information stored in the upper layer base vectors is transferred down to lower layer vectors. Along with the transfer of information, the base vectors in the upper layer get sparser. This has a major effect on the energy function. The following third phase, starting at step 600, is characterized by using the reorganized structure of the network to optimize both the reconstruction error and the sparsity. This is achieved by modifying the lower layer base vectors and the activations, leaving the upper layer base vectors mostly unchanged.

Through the whole relaxation process the energy function is steadily decreasing with a jump at the beginning of the reorganization phase. This leads to the conclusion that during the process the focus of what should be minimized is shifted between reconstruction and sparsity according to the chosen parameter set. When choosing extreme settings for the sparsity parameters the focus switches to sparsity maximization whereas the minimization of the reconstruction error does not play a role anymore and vice versa.

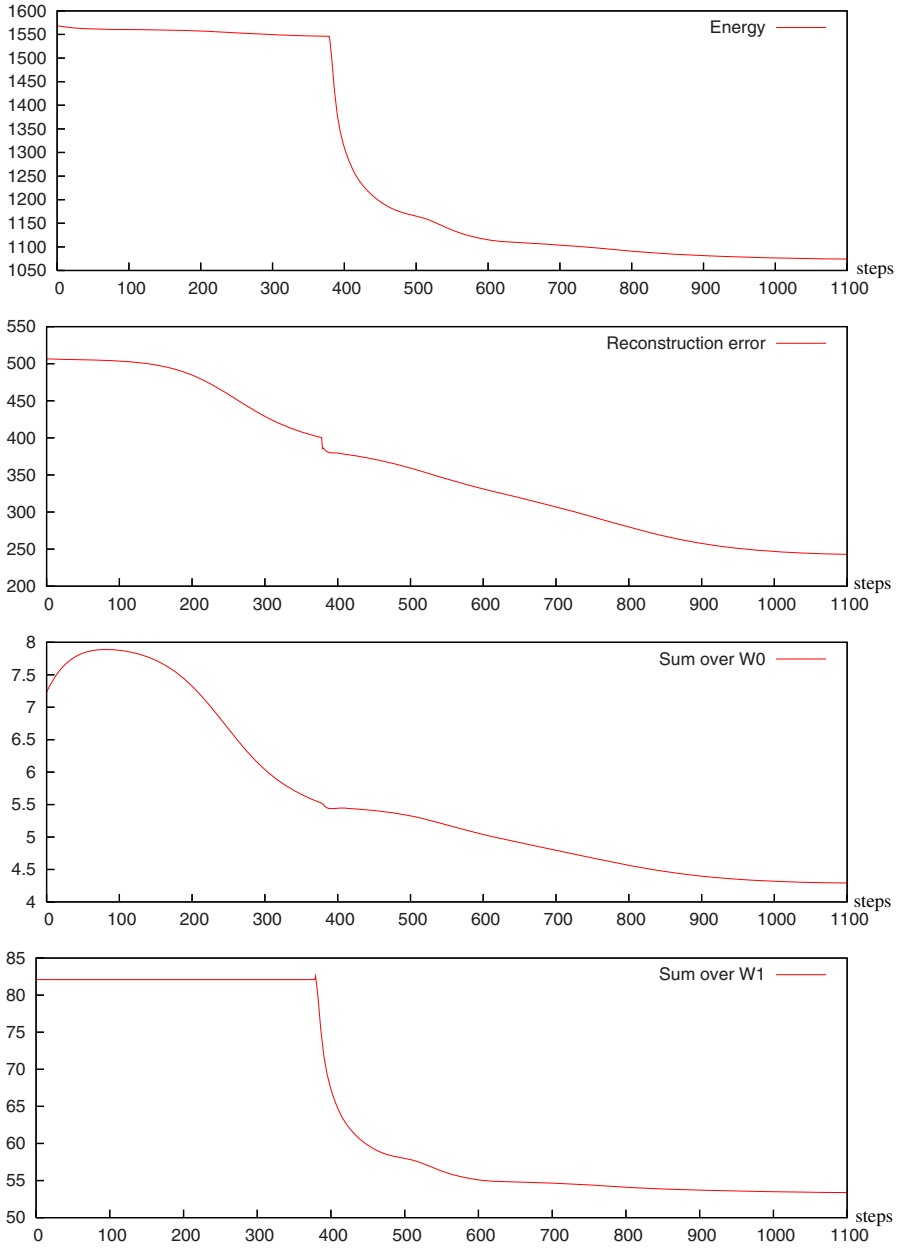

Fig. 5. These plots show the three phases of a two-layer HNMF network relaxation process for (from top to bottom) the energy function, the reconstruction error and the sparsity measure in the base vectors. The phases are: minimization of the reconstruction error, reorganization within the network to meet the sparsity constraint and the combined minimization of the reconstruction error and the sparsity penalization.

4 Conclusion

In this paper we propose the extension of the sparse Overlapping NMF introduced in [4] and [5] to a hierarchical, sparse and transformation-invariant network. This extension was done in a straightforward manner by defining the activations of each layer as a reconstruction of the layer above (see (6)). While other hierarchical approaches as in [3] store input transformations implicitly in the base vectors, our approach encodes the transformations explicitly. This explicit encoding leads to a reduction of redundancy in the base vectors, making the representation sparser and more efficient.

In Sect. 2.3 we discussed two different update schemes and concluded that only a combined relaxation of the whole network leads to a minimization of the overall energy function and is therefore preferable. Using the combined relaxation scheme on bar stimuli, we achieved the results depicted in Sect. 3, which show that the transformation-invariant and sparse HNMF is able to decompose the stimuli into the original parts. In this process the basic parts of the data set are stored in the lowest layer base vectors, whereas the higher layer base vectors compose a more complex and more abstract representation of the input by combining lower layer vectors. The resulting decomposition is a sparse representation of the input, having also very good reconstruction properties. By adapting the sparsity parameters, the network solution can be steered towards a perfect reconstruction or towards a sparse representation, where extreme settings will lead to insensible or trivial solutions.

As a final interesting point we want to mention that the proposed algorithm includes all previous approaches as special cases. To emulate [3] we just set all transformation matrices to unity matrices (no transformation), for [4,5] we take a single layer network and choose the sparsity parameters accordingly. So the transformation-invariant and sparse hierarchical NMF can be seen as a unification of the three mentioned NMF approaches.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects with nonnegative matrix factorization. Nature 401, 788–791 (1999)
2. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing Systems 13, 556–562 (2001)
3. Ahn, J.-H., Choi, S., Oh, J.-H.: A multiplicative up-propagation algorithm. In: Proceedings of the 21th International Conference, pp. 17–24 (2004)
4. Eggert, J., Körner, E.: Sparse coding and NMF. In: IJCNN 2004. Proceedings of the International Joint Conference on Neural Networks, pp. 2529–2533 (2004)
5. Eggert, J., Wersing, H., Körner, E.: Transformation-invariant representation and NMF. In: IJCNN 2004. Proceedings of the International Joint Conference on Neural Networks, pp. 2535–2539 (2004)