Supervised Matrix Factorization with Sparseness Constraints and Fast Inference

Markus Thom, Roland Schweiger, and Günther Palm

***Abstract*—Non-negative Matrix Factorization is a technique for decomposing large data sets into bases and code words, where all entries of the occurring matrices are non-negative. A recently proposed technique also incorporates sparseness constraints, in such a way that the amount of nonzero entries in both bases and code words becomes controllable. This paper extends the Non-negative Matrix Factorization with Sparseness Constraints. First, a modification of the optimization criteria ensures fast inference of the code words. Thus, the approach is real-time capable for use in time critical applications. Second, in case a teacher signal is associated with the samples, it is considered in order to ensure that inferred code words of different classes can be well distinguished. Thus, the derived bases generate discriminative code words, which is a crucial prerequisite for training powerful classifiers. Experiments on natural image patches show, similar to recent results in the field of sparse coding algorithms, that Gabor-like filters are minimizing the reconstruction error while retaining inference capabilities. However, applying the approach with incorporation of the teacher signal to handwritten digits yields morphologically completely different bases, while achieving superior classification results.**

I. Introduction

The computation of decompositions of large data sets is a fundamental part in modern artificial information processing systems. In many cases, biological sensory systems serve as prototypes for mathematical models and often lead to interesting insights. One of those concepts, that recently has found its way into the pattern classification community, is sparseness [1].

There is a two-fold meaning of the term sparseness when referring to such biological systems. Firstly, neurons being small scale information processing units are not connected to every other neuron in the network, so there is a sparse connection among them [2], [3]. Secondly, when processing information, only a small fraction of neurons is active, thus in the context of the whole population a sparse neuronal activity is encountered [4], [5].

One of the most important mathematical models that is known to produce sparse representations of certain data sets is the Non-negative Matrix Factorization (NMF) [6], [7]. The NMF aims to factorize a data matrix with non-negative entries into a product of a matrix of bases and a matrix of code words, both with non-negative entries. However, there are data sets where the NMF fails to produce sparse representations without further modifications to the algorithm itself [8]. The recently proposed Non-negative Matrix Factorization with Sparseness Constraints (NMFSC) [8] overcomes this problem by explicitly enforcing a sparse representation. Though the NMFSC provides an elegant way of controlling the desired sparseness of the generated representation, computing a sparse representation given a sample that is not part of the optimization process involves a cost-intensive optimization. Therefore the NMFSC is inapplicable in real-time scenarios.

Both the NMF and the NMFSC are unsupervised algorithms, that is they ignore any class labels in classification tasks. In order to improve classification capabilities, the bases should be optimized by incorporating the teacher signal such that they distinguish between the presented classes.

The following section of this paper summarizes approaches to the problem of computing sparse representations of large data sets. Section III presents an extension to the NMFSC, namely a supervised matrix factorization with sparseness constraints that also ensures fast inference of the code words, suitable for real-time applications. In Sect. IV, the results of the proposed technique applied to natural image patches and handwritten digits are demonstrated. The final section gives a summary and concludes this paper.

II. Previous Work

Several approaches for computing sparse representations have emerged over the past years. Two methods presented here, the NMF and the NMFSC, are decoding-only architectures, that is they do not provide an explicit way of efficiently encoding an arbitrary sample. Two other methods, namely Sparse Encoding Symmetric Machine (SESM) and Predictive Sparse Decomposition (PSD) are auto-encoders and thus provide a natural way of encoding any sample. Supervised extensions of sparse decoding-only architectures, e.g. [9], [10], are not discussed here.

A. Non-negative Matrix Factorization

The NMF [6], [7] is known to compute sparse representations of certain data sets. Assume that there is a data set of M samples from $\mathbb{R}^d$ with non-negative entries, written into one matrix $X \in \mathbb{R}_{\geq 0}^{d \times M}$. The NMF aims to find a factorization of X into a matrix of bases $W \in \mathbb{R}_{\geq 0}^{d \times n}$ and a matrix of code words $H \in \mathbb{R}_{\geq 0}^{n \times M}$ such that the reproduction error in Frobenius norm is minimized:

$$E_{\mathrm{NMF}}(W,H) := \|X - WH\|_F^2 \xrightarrow{!} \min_{W,H}. \tag{1}$$

Markus Thom and Roland Schweiger are with Daimler AG, Department Environment Perception (GR/PAP), Ulm, Germany (email: {markus.thom, roland.schweiger}@daimler.com).

Günther Palm is with University of Ulm, Institute of Neural Information Processing, Ulm, Germany (email: guenther.palm@uni-ulm.de).

978-1-4244-9637-2/11/$26.00 ©2011 IEEE

B. Non-negative Matrix Factorization with Sparseness Constraints

The idea of controlling the degree of sparseness leads to the definition of the NMFSC [8]. In doing so, a formal sparseness measure has to be defined. The original author's proposal of the sparseness σ is based on a normalized quotient of the L^1 norm and the L^2 norm of a vector:

$$\sigma\colon \mathbb{R}^d \setminus \{0\} \to [0,1], \quad x \mapsto \frac{\sqrt{d} - \frac{\|x\|_1}{\|x\|_2}}{\sqrt{d}-1}. \tag{2}$$

This sparseness measure has turned out to be very effective in the context of minimizing error functions that have the structure of a quadratic form. Using σ, the NMFSC is the problem of minimizing E_{NMF} under the constraints

$$\sigma(We_i) \stackrel{!}{=} \sigma_W \text{ for all } i \in \{1,\dots,n\} \tag{3}$$

$$\text{and } \sigma(H^t e_i) \stackrel{!}{=} \sigma_H \text{ for all } i \in \{1,\dots,n\} \tag{4}$$

for fixed degrees of sparseness $\sigma_W, \sigma_H \in (0,1)$ and with e_i being the i-th canonical basis vector. The biconvex objective function is minimized via alternating gradient descent on W and H, with each step followed by a projection to meet the sparseness constraints. This sparseness projection is achieved by an effective iterative algorithm [8], [11], [12] for computing the closest non-negative vector in the L^2 norm, meeting a predefined sparseness, for any given vector $x \in \mathbb{R}^d_{\geq 0}$. Let $\lambda_1, \lambda_2 > 0$ be constants and consider

$$S^{(\lambda_1,\lambda_2)}_{\geq 0} := \left\{ s \in \mathbb{R}^d_{\geq 0} \;\middle|\; \|s\|_1 = \lambda_1 \text{ and } \|s\|_2 = \lambda_2 \right\}. \tag{5}$$

Clearly, σ is constant on $S^{(\lambda_1,\lambda_2)}_{\geq 0}$ for fixed values of λ_1 and λ_2. The sparseness projection is then the problem of computing

$$\pi^{(\lambda_1,\lambda_2)}_{\geq 0}\colon \mathbb{R}^d_{\geq 0} \to S^{(\lambda_1,\lambda_2)}_{\geq 0}, \quad x \mapsto \operatorname*{arg\,min}_{s \in S^{(\lambda_1,\lambda_2)}_{\geq 0}} \|x-s\|_2. \tag{6}$$

Note that in case of dropping the non-negativity constraints, $\pi^{(\lambda_1,\lambda_2)}$ is the counterpart of this sparseness projection, and is computed using a slightly modified algorithm [8].

In the NMFSC, the sparseness of the individual columns of the weight matrix W is controlled by σ_W. As very sparse vectors yield high values of σ, a high value of σ_W indicates that only a small number of entries in every column of W is nonzero. σ_H controls the fraction of samples each column of W contributes to. A high value of σ_H indicates that each column influences the reconstruction of only a small number of samples. This has the effect of information distribution among W's columns in contrast to global methods like principal component analysis [13], where each column captures the residual maximum of information.

With the goal of using sparse representations in classification tasks, a sparse code word h of an arbitrary sample x has to be computed. When using the NMF, h is usually computed as being the projection of x onto W. However, when this approach is applied to the NMFSC, h forfeits its sparseness. A workaround for this issue is carrying out the cost-intensive optimization of finding $\arg\min_{h \in S} \|x - Wh\|_2^2$ where $S := S^{(\lambda_1,\lambda_2)}_{\geq 0}$, with $\lambda_1 := \mathbb{E}(\|h\|_1)$ and $\lambda_2 := \mathbb{E}(\|h\|_2)$ being the expected values of the norms of the sparse representations of the training samples in the L^1 sense and the L^2 sense respectively. Unfortunately, no real-time capable algorithm is known for solving this problem.

C. Sparse Encoding Symmetric Machine and Predictive Sparse Decomposition

The SESM [14] is an architecture which involves an encoder and a decoder part. Characteristically, both parts employ the same filter bank, that is they are using a filter matrix $W \in \mathbb{R}^{d \times n}$ and thresholds $\theta_{\text{enc}} \in \mathbb{R}^n$ for encoding and $\theta_{\text{dec}} \in \mathbb{R}^d$ for decoding. The behavior of the machine is determined by an encoder function and a decoder function:

$$f_{\text{enc}}\colon \mathbb{R}^d \to \mathbb{R}^n, \quad x \mapsto W^t x + \theta_{\text{enc}}, \tag{7}$$

$$f_{\text{dec}}\colon \mathbb{R}^n \to \mathbb{R}^d, \quad h \mapsto W \operatorname{Fermi}_\beta(h) + \theta_{\text{dec}} \text{ with } \beta > 0. \tag{8}$$

The compatibility between a sample $x \in \mathbb{R}^d$ and a potential decomposition $h \in \mathbb{R}^n$ is measured via the energy function $E(x,h) := \alpha_e \|h - f_{\text{enc}}(x)\|_2^2 + \|x - f_{\text{dec}}(h)\|_2^2$. Using the sparseness measure $\ell\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, $x \mapsto \sum_{i=1}^d \log\left(1 + x_i^2\right)$, the loss function $L_{\text{SESM}} := E(x,h) + \alpha_s \ell(h) + \alpha_r \|W\|_1$ is minimized in a sample after sample fashion via alternating gradient descent on h for finding the optimal decomposition and on the remaining parameters to update the machine. In this context, α_e, α_s and α_r are constants for weighting the individual penalty terms.

In contrast to the SESM, the PSD [15] uses two different filter matrices for encoding and decoding, $W_{\text{enc}} \in \mathbb{R}^{n \times d}$ and $W_{\text{dec}} \in \mathbb{R}^{d \times n}$ respectively. Further, the inference involves a nonlinearity using the mapping $x \mapsto \operatorname{diag}(g) \cdot \tanh(W_{\text{enc}} x + \theta_{\text{enc}})$, where $g \in \mathbb{R}^n$ compensates for the scaling of inferred decompositions. Optimization of the loss function

$$L_{\text{PSD}} := \|x - W_{\text{dec}} h\|_2^2 + \lambda \|h\|_1 + \alpha \|h - \operatorname{diag}(g) \cdot \tanh(W_{\text{enc}} x + \theta_{\text{enc}})\|_2^2 \tag{9}$$

again takes place by alternating stochastic gradient descent. Note, that while non-sparseness of decompositions h is penalized by the additive term of $\|h\|_1$ scaled by a constant $\lambda > 0$, non-sparseness of the filter matrices is neither penalized nor preferred.

III. Supervised Matrix Factorization with Sparseness Constraints and Fast Inference

Combining the ideas of the previous section, a new algorithm, called Sparse Coding for Fast Classification (SCFC), is proposed in this section. The algorithm aims at the computation of discriminative, sparse representations that may be employed in real-time classification tasks. SCFC unites all the advantages of the previous algorithms, such as rapid computation during learning and inference as well as high scalability, both in the number of available training samples and in their dimensionality. A classifier based on the sparse information processing paradigm can be realized by directly

incorporating the teacher signal, if one is associated with the investigated samples.

Sparseness is forced by using the projection operator $\pi^{(\lambda_1,\lambda_2)}$. Therefore, there is no need to enforce sparseness using the non-negativity constraints of the NMF where it led to a parts-based representation on some datasets [6]. Additionally, normalizing the training samples to zero mean and unit variance is advantageous in classification scenarios [16]. Thus in the remainder of this paper, the non-negativity constraint is dropped. If it is needed for specific applications, it can easily be adopted by replacing the unconstrained projection operator with the non-negative one.

A. Architecture

Let $X \in \mathbb{R}^{d\times M}$ be the fixed matrix of M samples, $W \in \mathbb{R}^{d\times n}$ be a filter matrix, which is a matrix of bases when employed for reconstruction, and $H \in \mathbb{R}^{n\times M}$ be a code word matrix. Furthermore, let $\theta_{\text{enc}} \in \mathbb{R}^n$ be a vector of thresholds used for code inference and $f\colon \mathbb{R}^n \to \mathbb{R}^n$ be a nonlinear transfer function. In case of supervised learning, let $T \in \mathbb{R}^{1\times M}$ be the vector of scalar teacher signals associated with the samples, $w_{\text{out}} \in \mathbb{R}^n$ be a weight vector, $\theta_{\text{out}} \in \mathbb{R}$ be a threshold used for predicting the class labels of the samples and let $g\colon \mathbb{R} \to \mathbb{R}$ be a nonlinear transfer function which is chosen to be the hyperbolic tangent. Using this nomenclature, Fig. 1 illustrates the dynamics of the proposed architecture.

It follows a symmetric design consisting of a reconstruction path and an inference path, which is extended by a classification path in case a teacher signal is available. The reconstruction path is adopted from the NMF approach such that a latent code word h is decoded by linear combination with the filter matrix, $h \mapsto \tilde{x} := Wh$. Similar to the NMF, the reconstruction error on the whole data set is measured using

$$E_R(W,H) := E_{\text{NMF}}(W,H) = \|X - WH\|_F^2 . \tag{10}$$

The inference path uses the same filter matrix W for computing sparse decompositions of input samples. In addition, the threshold vector θ_{enc} and the transfer function f are used for computing the feedforward code word $x \mapsto \tilde{h} := f(W^t x + \theta_{\text{enc}})$. By computing the squared difference between the code word matrix and the matrix of feedforward code words, the inference error is given by

$$E_I(W,\theta_{\text{enc}},H) := \left\|H - f\left(W^t X + \theta_{\text{enc}} \cdot J_{1\times M}\right)\right\|_F^2 . \tag{11}$$

Here, $J_{1\times M} \in \mathbb{R}^{1\times M}$ denotes a matrix containing only numerical ones and is employed for repeating the threshold over all M samples.

The classification path processes the feedforward code words through an output layer by a linear combination with the output weight vector w_{out}, followed by adding a threshold θ_{out} and applying the second transfer function, $\tilde{h} \mapsto y := g(w_{\text{out}}^t \tilde{h} + \theta_{\text{out}})$. Thus, inference and classification path together form a two layer neural network. The classification error is measured using the squared difference between the predicted class label and the actual teacher signal,

$$E_C(W,\theta_{\text{enc}},w_{\text{out}},\theta_{\text{out}}) := \left\|T - g\left(w_{\text{out}}^t \tilde{H} + \theta_{\text{out}} \cdot J_{1\times M}\right)\right\|_F^2 , \tag{12}$$

where $\tilde{H} := f(W^t X + \theta_{\text{enc}} \cdot J_{1\times M}) \in \mathbb{R}^{n\times M}$ denotes the matrix of all inferred code words.

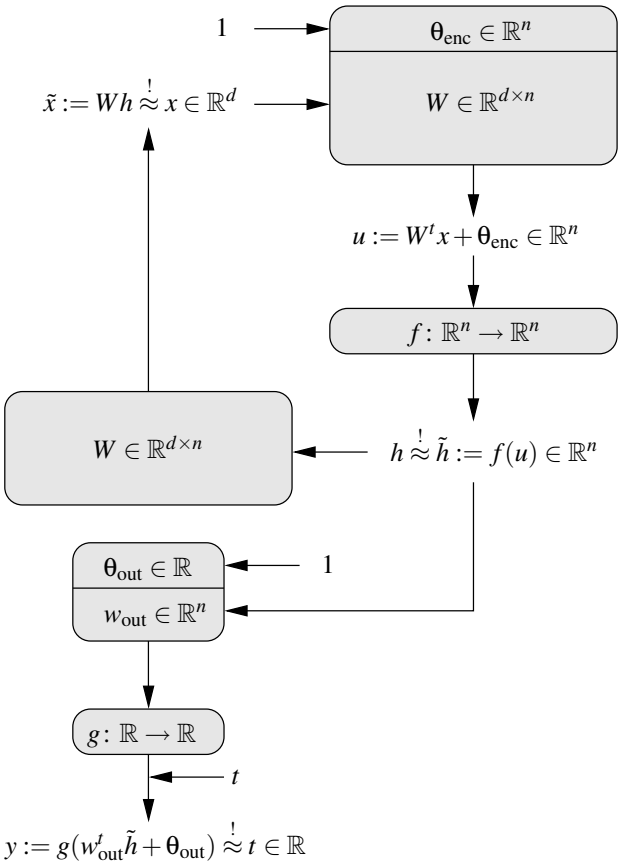


Fig. 1. Proposed architecture consisting of a reconstruction ($h \mapsto \tilde{x}$) and an inference path ($x \mapsto \tilde{h}$). The architecture is extended by a classification path ($x \mapsto \tilde{h} \mapsto y$) if a teacher signal is available.

With $\alpha_I, \alpha_C \in [0,1]$ such that $0 \le \alpha_I + \alpha_C \le 1$ controlling the trade-off between reconstruction, inference and classification capabilities the objective function of the proposed architecture is defined to be a convex combination of the three individual error functions defined above:

$$\begin{aligned} E(W,\theta_{\text{enc}},H,w_{\text{out}},\theta_{\text{out}}) &:= (1-\alpha_I-\alpha_C)\cdot E_R(W,H) \\ &\quad + \alpha_I \cdot E_I(W,\theta_{\text{enc}},H) + \alpha_C \cdot E_C(W,\theta_{\text{enc}},w_{\text{out}},\theta_{\text{out}}). \end{aligned} \tag{13}$$

In case no teacher signal is available, $E_C \equiv 0$ is assumed and α_C is kept fixed to zero throughout the whole optimization process. E is minimized subject to the same sparseness constraints that applied when discussing the NMFSC, (3) and (4).

B. Optimization

Similar to the NMFSC, the optimization of E during SCFC training makes use of a sparseness projection by employing the projection operator $\pi^{(\lambda_1,\lambda_2)}$ as defined in Sect. II-B after each step. Let $\pi_W\colon \mathbb{R}^{d\times n} \to \mathbb{R}^{d\times n}$ be the projection

operator that performs a column-wise sparseness projection on W, preserving the L^2 norms of the columns and choosing the L^1 norms such that a sparseness of σ_W is achieved, that is

$$\pi_W(W)e_i := \pi^{(\ell_W \cdot \|We_i\|_2, \|We_i\|_2)}(We_i) \tag{14}$$

for all $i \in \{1,\dots,n\}$ with $\ell_W := \sqrt{d} - \sigma_W\left(\sqrt{d}-1\right)$. In the same way $\pi_H \colon \mathbb{R}^{n\times M} \to \mathbb{R}^{n\times M}$ is defined row-wise, ensuring unit L^2 norm and an appropriate L^1 norm is achieved for a sparseness of σ_H:

$$\pi_H(H)^t e_i := \pi^{(\ell_H, 1)}(H^t e_i) \tag{15}$$

for all $i \in \{1,\dots,n\}$ with $\ell_H := \sqrt{M} - \sigma_H\left(\sqrt{M}-1\right)$.

For optimization, the variables W, θ_{enc}, H, w_{out} and θ_{out} are first initialized using small random values, followed by a projection using π_W and π_H to fulfill the sparseness constraints on W and H, respectively. Then in every iteration, small steps of gradient descent are performed on the variables in an alternating fashion. In that context, each step of gradient descent on W and H is immediately followed by a sparseness projection step using π_W or π_H, respectively. To guarantee that the objective function E always decreases with every step, the step sizes for the gradient descent are kept separately for all variables and are adjusted accordingly.

In contrast to similar optimization techniques, the trade-off constants α_I and α_C are not fixed throughout the whole process. Both are adjusted starting from small values and reaching an intermediate value asymptotically. Starting at zero, α_I is gradually increased to reach its maximum of $1/2$. Thus in the first few iterations the optimization is similar to the one used by the NMFSC, that is $E \approx E_R$, while the impact of E_I slowly grows stronger each iteration. This approach stems from the fact, that for small values of α_I good reconstruction capabilities can be obtained quickly, while inference capabilities are only slowly converging to an optimum. If the unsupervised variant is employed, α_C remains fixed at zero. Otherwise, α_C is kept at zero while α_I is ascending. As soon as $\alpha_I \equiv 1/2$ is reached, α_C is gradually increased until $\alpha_I + \alpha_C = 1$. This is motivated by the fact, that a positive value of α_C causes the model to diverge, unless rudimentary inference and reconstruction capabilities have been guaranteed.

C. Transfer Function

The transfer function f used for code word inference is chosen to be a hyperbolic tangent raised to an odd exponent greater or equal to three, that is

$$f_{(\beta,q)} \colon \mathbb{R} \to \mathbb{R}, \quad x \mapsto (\tanh(\beta x))^q \tag{16}$$

with $\beta > 0$ and $q \in 2\mathbb{N}_0 + 3 = \{3,5,7,\dots\}$. Thus a small plateau with vanishing slope emerges in a neighborhood of zero. This has a similar effect as applying a shrinkage function after a hyperbolic tangent transfer function, but has the advantage of being differentiable everywhere. Note, that by dropping the exponentiation (that is choosing $q = 1$) matching the inferred codes with the latent code words is virtually impossible, that is the inference error E_I is bounded from below by a large constant value. With $q \in 2\mathbb{N}_0 + 3$, sparse vectors are achieved easily. f is continued on $\mathbb{R}^n$ by element-wise evaluation.

D. Comparison with Stochastic Gradient Methods

SCFC belongs to the family of batch gradient algorithms as it works on the whole data set simultaneously. While stochastic gradient algorithms usually are more suited to large data sets [17], this does not hold when sparseness constraints on the overall activation of small scale information processing units are incorporated in contrast to sparseness constraints of a single code word. These two values, however, are connected.

During optimization, the rows of H are scaled to unit L^2 norm and to an L^1 norm of ℓ_H. The expected values of the norms of the columns of H, that is the code words h, are computed by interchanging the order of summation to yield $\mathbb{E}(\|h\|_1) = \frac{n}{M} \cdot \ell_H$ and $\mathbb{E}\left(\|h\|_2^2\right) = \frac{n}{M}$. By computing a Taylor expansion and using $0 \ll n \ll M$, an approximation of the expected sparseness of a code word is given by

$$\begin{aligned}\mathbb{E}(\sigma(h)) &= \frac{\sqrt{n} - \mathbb{E}\left(\frac{\|h\|_1}{\|h\|_2}\right)}{\sqrt{n}-1} \approx \frac{\sqrt{n} - \frac{\mathbb{E}(\|h\|_1)}{\sqrt{\mathbb{E}(\|h\|_2^2)}}}{\sqrt{n}-1} \\ &= \sigma_H \cdot \frac{\sqrt{n} - \sqrt{\frac{n}{M}}}{\sqrt{n}-1} \approx \sigma_H \cdot \left(1 + \frac{1}{\sqrt{n}}\right). \end{aligned} \tag{17}$$

This value is slightly greater than the selected row sparseness of H. Although many assumptions had to be made to deduce this approximation, experiments have shown that the relative error between the approximation and the precise mean sparseness is not greater than 3%, in most cases the relative error is smaller than 0.5%. In addition, the variance of the code word sparseness is found to be almost vanishing, meaning that 99% of the code words possess a sparseness that is not farther than 0.1 in absolute value from the mean value.

This approximation shows that controlling the sparseness of the latent code words over both space and time is achieved easily. In stochastic gradient methods, the sparseness of a code word may be easily controllable as well, but there is no a priori guarantee that this affects the activation considered over the whole data set as well.

An analysis of the space complexity of SCFC, neglecting variables that consume only very little memory, shows it needs approximately $S := 3n + 3dn + 4nM + 2dM$ scalar values. Since usually $M \gg d$ and $M \gg n$ holds, S is dominated by its last two terms, which scale linearly with the number of samples. When using double precision IEEE 754 numbers on the MNIST data set [18] (where $M = 60000$ and $d = 784$), the algorithm employs 1191 MB of memory for $n = 256$, which is roughly one third of a modern workstation's system memory of 4096 MB. That shows that the often mentioned disadvantage of high memory consumption when using batch gradient methods is no real limitation to a great variety of applications.

IV. Experiments

The performance of SCFC is evaluated on two data sets. The first data set consists of natural image patches without teacher signal and is used to evaluate the unsupervised variant. The second data set is the popular MNIST database of handwritten digits, which provides ten unique labels. Both the unsupervised and the supervised variant of the proposed algorithm are applied to it.

A. Natural Image Patches

To verify that the proposed architecture is able to produce results similar to previous experiments, a sparse representation based on natural image patches was computed. Being a benchmark in the sparse coding community [4], [19], this problem is known to yield Gabor-like filters. For this experiment, the original images [20] have been used to generate 30000 patches of size 14×14 pixels. The filters learned from this data set for parameters $n := 64$ and $\sigma_H := 0.85$ are shown in Fig. 2 for varying sparseness constraints σ_W. The filters shown are exact instances of Gabor filters modulo pixel-wise affine linear transformations. The Gabor hyperparameter that best matches a sparse filter is found by maximizing the pixel-wise correlation coefficient ρ between the values of a sparse filter and the resulting Gabor filter. The best results were achieved for high values of the filter sparseness σ_W, namely $\rho \geq 0.98$ for $\sigma_W = 0.75$. If $\sigma_W = 0.5$ is chosen, ρ can be bounded from below by 0.95.

B. Handwritten Digits

For evaluating SCFC on another benchmark data set, the MNIST database of handwritten digits [18] has been chosen. The data set consists of 70000 samples, divided into a learning set of 60000 samples and a test set of 10000 samples, where each sample represents a digit of size 28×28 pixels. A teacher signal in $\{0,\ldots,9\}$ is associated with each digit. The only preprocessing applied to the samples is a normalization to zero mean and unit variance.

For evaluating the unsupervised variant of SCFC, it is run with several degrees of sparseness σ_W and σ_H varied between 0.1 and 0.9. In each run, $n := 192$ filters were computed using the samples from the learning set. After convergence, a linear SVM classifier [24], [25] was trained using the feedforward codes in a one vs. one fashion, resulting in 45 subclassifiers for each combination of (σ_W, σ_H). The final classifiers were then used to compute the classification error on the test set.

Figure 3a shows a selection of filters with $\sigma_H = 0.8$ and $\sigma_W \in \{0.5, 0.75, 0.85\}$. For $\sigma_W = 0.5$, the individual filters resemble entire digits, while for $\sigma_W = 0.85$, only strokes of the digits are left. The optimum classification performance is achieved by rather high degrees of sparseness, as can be seen in Fig. 3b. A minimum error of 2.48% on the test set is achieved for $(\sigma_W, \sigma_H) = (0.8, 0.7)$, which compares quite well to the 4.7% error of a two layer neural network with 300 hidden units reported by [17]. Figure 3c reveals a connection between reconstruction and classification error. A very low reconstruction error implies a low classification error. The converse does not hold, which means that the impact of the reconstruction abilities on the classification abilities is not completely negligible.

For comparing the unsupervised with the supervised variant of SCFC, 1000 filters have been computed using both variants for parameters $\sigma_W = 0.75$ and $\sigma_H = 0.8$. As the supervised variant only supports binary labels, 100 filters have been computed in each run of a one vs. all fashion and eventually been combined to yield a filter bank with 1000 filters in total. A comparison of the filters computed from both variants is given in Fig. 3d. While the sparse filters that stem from the unsupervised variant only possess non-negative entries, the filters of the supervised variant possess about as many positive entries as negative entries in adjacent regions. This means that contrast detectors are optimizers of sparse filters if classification performance is involved in the optimization, while local blobs of equal sign are optimizers of sparse filters if reconstruction capabilities are involved instead.

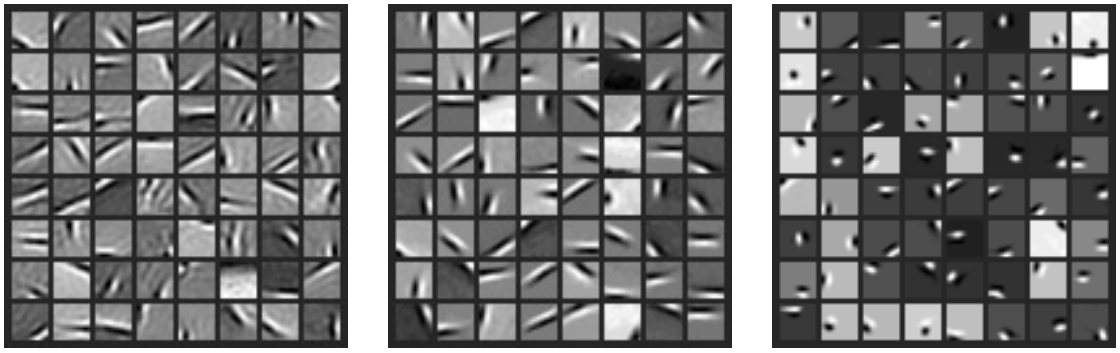
Finally, a linear SVM classifier was trained on the feedforward codes to compute two classifiers. The classifier using filters from the unsupervised optimization run achieved a classification error of 1.7% on the test set, while the classifier using the combined filter bank from the supervised optimization runs achieved a classification error of 1.4%. Thus, incorporating the teacher signal into the computation of the feature detector additionally boosts the classification performance significantly compared to when the teacher signal is ignored.

Table I gives an overview of various approaches to classifying the MNIST dataset. As the approach proposed in this paper is non-convolutional, the presented selection is restricted to similar methods only. The overview can be divided into permutation-invariant methods and methods that take the arrangement of individual pixels into account. The latter are those that employ virtual training samples generated from the original training dataset using elastic distortions [21].

Clearly, two layer neural network architectures [17], [21] trained using the well-established backpropagation learning algorithm [26], [27] are inferior to the comparable architecture of a combination of a supervisedly trainable feature detector with a linear SVM classifier. Thus the approach presented in this paper is quite competitive compared to simple architectures. Only by using very deep classification architectures [22], [23] or permutation-variant generation of artificial training data [21], further improvements can be achieved. The current record of 0.35% error rate is being held by the most complex architecture in conjunction with a huge artificial enhancement of the training data [23]. Concluding, the restriction to simple architectures permits an easier understanding of the intrinsic properties of classification systems. Further, optimal structures become apparent due to sparseness constraints, see Fig. 3d.

V. Conclusions

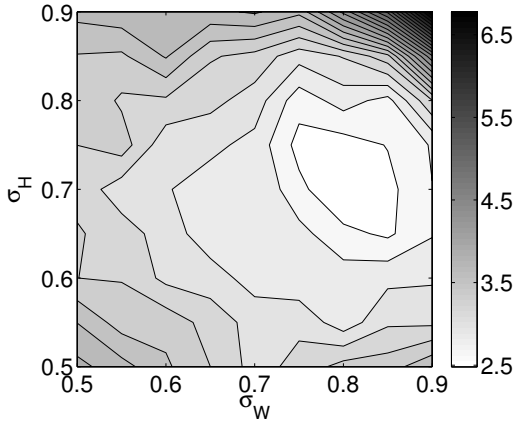
Non-negative Matrix Factorization with Sparseness Constraints has proven to be a useful tool in the analysis of a



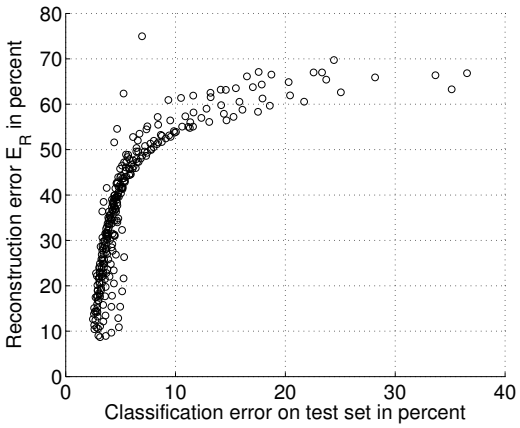
(a) no constraints on W (b) $\sigma_W = 0.5$ (c) $\sigma_W = 0.75$

Fig. 2. Filters learned from the original sparse coding data set [19] for (a) no sparseness constraint on W, (b) a medium and (c) a rather strong sparseness constraint. These filters are exact instance of Gabor filters modulo pixel-wise affine linear transformations. The Gabor filter inversion works better with increasing degree of filter sparseness σ_W.

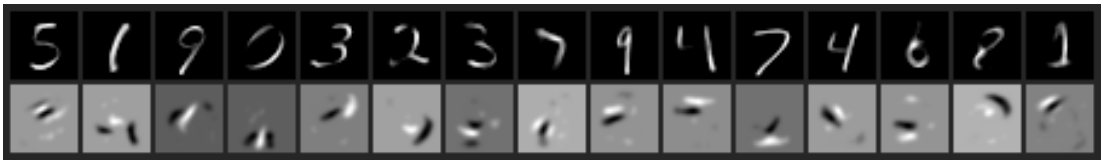
(a) Selection of filters with $\sigma_H = 0.8$ computed for $\sigma_W \in \{0.5, 0.75, 0.85\}$ (top, middle and bottom row respectively) using the unsupervised variant of SCFC.



(b) Classification error in percent using a linear SVM classifier on test set for $(\sigma_W, \sigma_H) \in [0.5, 0.9]^2$.



(c) Scatter plot of classification error and reconstruction error, both in percent, for all pairs (σ_W, σ_H).



(d) Selection of filters with $\sigma_W = 0.75$ and $\sigma_H = 0.8$ using the unsupervised variant (top row) and the supervised variant (bottom row) of SCFC. The pixel values of the filters have been normalized for displaying purposes. The filters from the unsupervised variant possess only non-negative entries (positive entries are marked white, vanishing entries are marked black). The filters of the supervised variant possess about as many positive entries (white regions) as negative entries (black regions) in adjacent regions, the remainder consists of vanishing entries (gray regions).

Fig. 3. Various results of the application of the proposed algorithm, SCFC, on the MNIST database of handwritten digits. Subfigures (a), (b) and (c) show results of the unsupervised variant, and Subfig. (d) shows a comparison of filters of both the unsupervised and the supervised variant.

TABLE I
OVERVIEW OF VARIOUS METHODS APPLIED TO THE MNIST DATABASE OF HANDWRITTEN DIGITS.

Method	Architecture (in-hidden-out)	Pretraining	Virtual samples	Test set error	Reference
Two layer NN	784 – 1000 – 10	none	no	4.5%	[17]
Two layer NN	784 – 800 – 10	none	no	1.6%	[21]
Two layer sparse NN	784 – 1000 – 10	unsupervised	no	1.4%	this paper
Four layer RBM	784 – 500 – 500 – 2000 – 10	unsupervised	no	1.2%	[22]
Two layer NN	784 – 800 – 10	none	elastic distortions	0.7%	[21]
Six layer NN	784 – 2500 – 2000 – 1500 – 1000 – 500 – 10	none	elastic distortions	0.35%	[23]

diverse range of data sets. By controlling the degree of sparseness while minimizing the reproduction error, the NMFSC has proven to be powerful in the field of data analysis. The main disadvantage is that it is only a decoding architecture, that is the NMFSC does not provide an explicit way of efficiently encoding an arbitrary sample. In this paper, SCFC, which is an efficient algorithm for the decomposition of large data sets into sparse bases and code words has been proposed. By including reconstruction as well as inference capabilities into the optimization criteria, the presented algorithm has the advantage of rapid computation not only during learning, but also during inference. This property of fast code word inference renders SCFC suitable for real-time classification scenarios. The algorithm can be implemented very efficiently and scales well with large data sets. Being a batch gradient algorithm, sparseness over both space and time is achieved easily. When applied to natural image patches, Gabor-like filters are resulting, showing that they are minimizers of both reconstruction and inference error. Furthermore, the SCFC allows the incorporation of teacher signals, thereby creating more discriminative features. Experiments on handwritten digits show that this yields morphologically completely different bases, which, however, induce superior results in subsequent classification. Thus the supervised variant provides a useful and important extension of existing procedures.

REFERENCES

[1] D. J. Field, "What is the Goal of Sensory Coding?" *Neural Computation*, vol. 6, pp. 559–601, 1994.

[2] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii, "Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits," *Public Library of Sciences Biology*, vol. 3, no. 3, p. e68, 2005.

[3] Y. Yoshimura, J. L. M. Dantzker, and E. M. Callaway, "Excitatory cortical neurons form fine-scale functional networks," *Nature*, vol. 433, no. 7028, pp. 868–873, 2005.

[4] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[5] ——, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, pp. 481–487, 2004.

[6] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[7] ——, "Algorithms for Non-negative Matrix Factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[8] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[9] M. Heiler and C. Schnörr, "Learning Sparse Representations by Non-Negative Matrix Factorization and Sequential Cone Programming," *Journal of Machine Learning Research*, vol. 7, pp. 1385–1407, 2006.

[10] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised Dictionary Learning," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1033–1040, 2009.

[11] F. J. Theis, K. Stadlthanner, and T. Tanaka, "First results on uniqueness of sparse non-negative matrix factorization," *Proceedings of European Signal Processing Conference*, 2005.

[12] F. J. Theis and T. Tanaka, "Sparseness by iterative projections onto spheres," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[13] H. Hotelling, "Analysis of a Complex of Statistical Variables Into Principal Components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[14] M. Ranzato, Y.-L. Boureau, and Y. LeCun, "Sparse Feature Learning for Deep Belief Networks," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1185–1192, 2008.

[15] K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "Fast Inference in Sparse Coding Algorithms with Applications to Object Recognition," Computational and Biological Learning Lab, Courant Institute, NYU, Tech. Rep. CBLL-TR-2008-12-01, 2008.

[16] Y. LeCun, I. Kanter, and S. A. Solla, "Eigenvalues of Covariance Matrices: Application to Neural-Network Learning," *Physical Review Letters*, vol. 66, no. 18, pp. 2396–2399, 1991.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] Y. LeCun and C. Cortes, "The MNIST Database of Handwritten Digits," http://yann.lecun.com/exdb/mnist.

[19] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, pp. 3311–3325, 1996.

[20] B. A. Olshausen, "Sparse coding simulation software," http://redwood.berkeley.edu/bruno/sparsenet.

[21] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," *Proceedings of International Conference on Document Analysis and Recogntion*, pp. 958–962, 2003.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 1527–1554, 2006.

[23] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[24] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[25] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[26] J. Arthur E. Bryson and Y.-C. Ho, *Applied Optimal Control*. Taylor & Francis Group, LLC, 1975.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.