Proceedings of the 38th Chinese Control Conference
July 27-30, 2019, Guangzhou, China

Semi-NMF network for image classification

Haonan Huang[1,2], Zuyuan Yang[1,3], Naiyao Liang[1,4], Zhenni Li[1,4]

1. School of Automation, Guangdong University of Technology, Guangzhou 510006, P. R. China
E-mail: haonanhuang@mail2.gdut.edu.cn

2. Guangdong Key Laboratory of IoT Information Processing, Guangzhou 510006, P. R. China

3. Key Laboratory of Ministry of Education, Guangzhou 510006, P. R. China

4. State Key Laboratory of Precision Electronic Manufacturing Technology and Equipment, Guangzhou 510006, P. R. China

Abstract: Semi non-negative matrix factorization (Semi-NMF) is an algorithm that gets a low-dimensional representation of a database. In this paper, we propose an efficient convolutional neural network (CNN) based on Semi-NMF to address the image classification problem. Unlike the traditional learning approach of CNN, convolutional filters are achieved by the Semi-NMF on input images. In the output layer, binary hashing and blockwise histograms are used to obtain the feature maps. To get better classification results, we introduce the weakly supervised Semi-NMF network which incorporates known attribute information for computing the convolutional filters in first layer. Experiments on MNIST dataset show our methods perform better than the state-of-the-art methods.

Key Words: Semi non-negative matrix factorization, Convolutional neural network, Image classification, Weakly supervised.

1 Introduction

The challenge of image classification has always been a research hotspot in computer vision. The classical algorithms for database feature extraction are principal component analysis (PCA) and non-negative matrix factorization (NMF) which can learn the holistic and parts-based representation respectively [1]. Local binary patterns (LBP) and histogram of oriented gradient (HOG) have also shown reasonable performance in object recognition with manually low-level features. In the most cases, support vector machine (SVM) classifier is used. However, hand-crafted features have poor robustness in a new condition.

In recent years, deep learning models have become the leading architecture for most image classification tasks. Deep convolutional neural network has demonstrated outstanding performance for Large-Scale-Visual-Recognition-Challenge (ILSVRC) 2012 [2]. As well-known, traditional deep learning architecture cost lots of computing memory and training time, because the supervised back propagation algorithm is used to train CNN [3, 4]. Therefore, combining traditional machine learning algorithm to optimize deep learning model has become a new research hotspot [5–7]. Scattering convolutional networks (ScatNet) [8] is a novel work which has clear mathematical justification and the convolutional filters in ScatNet are prefixed through wavelet operators. PCANet [9] is a deep learning network which has a tow-stage structure; the convolutional filters is obtained by PCA computation; the output layer combined with binary hashing and block-wise histograms. CUNet [10] adopted K-means algorithm to get the convolutional filters instead of PCA and combined non-linear activation and pooling. However, ScatNet and PCANet can not extract the nonlinear feature of convolutional kernels, CUNet will cost lots of computational memory.

This work is supported by National Natural Science Foundation (NNSF) of China under Grants 61722304 and 61803096, the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2014A030306037 and Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation under Grant NO.PDJHB0616.

Non-negative matrix factorization (NMF) [11] is a successful dimensionality reduction technique in lots of areas including, but not limit to image classification and blind source separation. For example, non-negative data matrix V is decomposed into W and H that are also non-negative, such that $V \approx WH$. Sparse-NMF [12, 13] and Ans-NMF [14] are both useful algorithms to solve the problems with sparse distributions. In order to improve the ability of NMF in case where the data matrix V is not strictly non-negative, Ding et al. [15] proposed Semi-NMF, an NMF variant the imposes non-negative constrains only on the second factor H. This approach allows Semi-NMF to learn lower-dimensional features from the data and have a convenient clustering interpretation. Yang et al. [16] proposed NMF with dual constraints (NMF-DC) which combined label constraint and sparse constraint in NMF.

To learn the non-linear characteristics of convolutional filters with fast speed. In this paper, we propose Semi-NMF network (SNnet) which combines convolutional network structure and Semi-NMF algorithm. The convolutional filter bank in SNnet is constructed by applying the Semi-NMF on extracted image patches. Fig. 1 has shown the general Semi-NMF network structure and we can extend the model to multi-layer on the basis of this structure. The main contribution of our works can be concluded in three aspects:

- The filter bank is constructed by applying the Semi-NMF on a set of extracted image patches. The number of filters is also the dimension of our model being reduced. Semi-NMF can extract data features effectively and has a clear mathematical justification of its effectiveness.
- The most important difference from CNNs is that we abandoned back propagation trick in SNnet, this approach can speed up the training time of SNnet and save lots of computational memory.
- In order to make full use of the image labels, we also introduce weakly supervised Semi-NMF network (S-SNnet) which combined graph regularization with Semi-NMF. The feature extraction ability of the S-SNnet is stronger than of the SNnet.

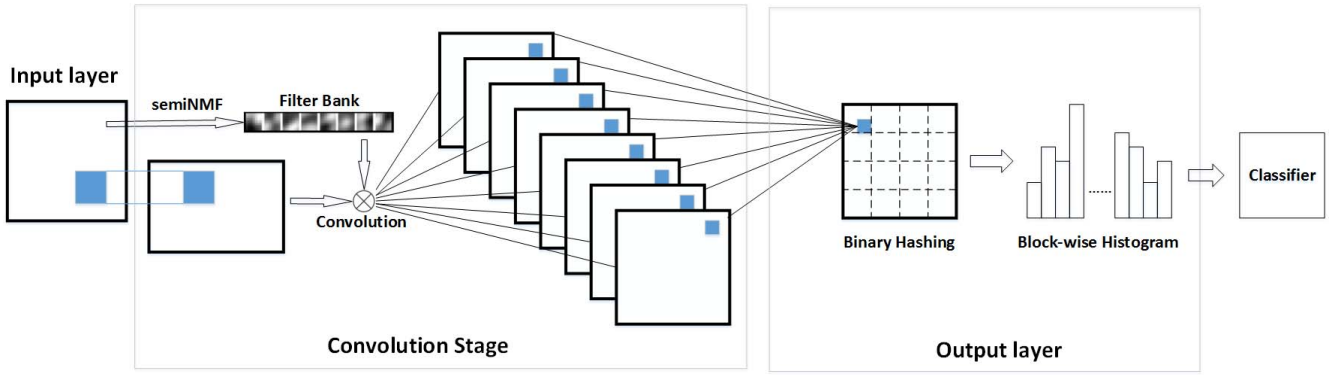

Fig. 1: The general Semi-NMF network structure.

2 Methodology

In this work, suppose we are given n input training images of size $p_1 \times p_2$. We initialize $a \times b$ patch to sample each input images and construct a new matrix $\tilde{X}_i \in \mathbb{R}^{ab \times \tilde{p_1}\tilde{p_2}}$, where $\tilde{p_1} = p_1 - a + 1$ and $\tilde{p_2} = p_2 - b + 1$. Each $\tilde{X}_i$ is normalized by subtracting the mean, then we obtained the dataset $\bar{X} = [\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_n]$. In the convolutional stage, we define the matrix $V = \sum_{i=1}^{n} \bar{X}_i$ for factorization.

2.1 Semi-NMF

Non-negative matrix factorization (NMF) [11, 17],which factorizes a non-negative matrix into two non-negative matrices, has well performance in feature extraction. Semi non-negative matrix factorization (Semi-NMF) [15] is a new variation in the theme of NMF, which can factorize a non-restricted $V^{\pm}$ into a non-restricted matrix $W^{\pm}$ and a non-negative matrix H^{+}, thus approximating the following factorization:

$$V^{\pm} \approx W^{\pm} H^{+} \tag{1}$$

We use Semi-NMF instead of NMF because Semi-NMF relaxes the non-negativity constrains of NMF and keep the positive and negative symmetry of the convolution kernel. The cost function we optimize for approximating the Semi-NMF factors is indeed:

$$C_{Semi-NMF} = \|V - WH\|_F^2 \tag{2}$$

where data matrix V, W are non-restricted but H is non-negative. Then we update W through fixing the H

$$W = VH(H^{\top}H)^{-1} \tag{3}$$

The $H^{\top}H$ is a positive semidefinite matrix. Generally, $H^{\top}H$ is nonsingular and when $H^{\top}H$ is singular ,we take the pseudo-inverse. Then we update H while W is fixing

$$H = H\sqrt{\frac{(W^{\top}V)^{+} + [H(W^{\top}W)]^{-}}{(W^{\top}V)^{-} + [H(W^{\top}W)]^{+}}}) \tag{4}$$

2.2 Weakly Supervised Semi-NMF

Motivated by NMF with manifold regularization [18] and deep Semi-NMF model [19], consider a graph with N vertices, where each vertex corresponds to a data point. A node i is connected to another node j if we have a priori knowledge that those samples share the same label, and this edge has a weight w_{ij}. In this work, we use a binary weight matrix $w_{ij} = 1$, if nodes i and j are connected by an edge. Then we formulate L which denotes the Graph Laplacian that stores the prior knowledge about the relationship of the samples. With the above defined weight matrix w, we use the R term to measure the smoothness of the low-dimensional representation:

$$\begin{aligned} R &= \sum_{i,j=1}^{N} \|h_i - h_j\|^2 w_{ij} \\ &= \sum_{i,j=1}^{N} h_i^{\top} h_j D_{ii} - \sum_{i,j=1}^{N} h_i^{\top} h_j w_{ij} \\ &= Tr(H^{\top}DH) - Tr(H^{\top}wH) \\ &= Tr(H^{\top}LH) \end{aligned} \tag{5}$$

where $Tr(.)$ denotes the trace of a matrix, h_i is the low-dimensional representation of sample i and D is a diagonal matrix whose entries are column (or row, since w is symmetric) sums of w, $D_{ii} = \sum_{iw} w_{ij}$.

By combining the term R as shown in formula 5, we obtain the cost function for weakly supervised Semi-NMF:

$$C_{S-Semi-NMF} = \|V - WH\|_F^2 + \lambda Tr(H^T LH) \tag{6}$$

where the hyper-parameter $\lambda \geq 0$ controls the smoothness of the new representation. The update rule about W as formula 3 and we get a new update rule about H through fixing the W

$$H = H\sqrt{\frac{(W^{\top}V)^{+} + [H(W^{\top}W)]^{-} + \lambda Hw}{(W^{\top}V)^{-} + [H(W^{\top}W)]^{+} + \lambda HD}} \tag{7}$$

2.3 Convolution Stage

Assuming that the number of filters in the first layer is K_1 , we run Semi-NMF on V to acquire the feature matrix $W \in \mathbb{R}^{ab \times K_1}$. The first layer filter banks J_1 is reshaped by W and denoted as $J_1 = \{F_1, F_2, \ldots, F_{K_1}\}$. We convolve each input images X_i with the filters, let the K_1-th filter output of the first layer be

$$O_i^{K^1} = X_i * F_{K_1}, i = 1, 2, ..., n, \tag{8}$$

where $*$ denotes $2D$ convolution, and there are $n \times K_1$ output images of this stage. We have f convolutional stages, the number of output images at each stage is $Q_i, i = 1, 2, ..., f$. The first layer output $Q_1 = n \times K_1$. It's remarkable that we only add Supervised Semi-NMF in the first convolutional layer because the convolution method will lead to exponential increase in the amount of data.

2.4 Output layer

In this work, we take the inspiration from [9] and construct two convolutional layers. The number of output images at the final stage is $Q_2 = Q_1 \times K_2 = n \times K_1 \times K_2$. Each set of the K_2 feature are binary-mapped by performing $B(O_i^{K_2})$. Then we map the Q_2 outputs into a single integer-valued image:

$$I_i = \sum_{l=1}^{Q_2} 2^{l-1} B(O_i^{K_2}), i = 1, 2, ..., n, \tag{9}$$

Finally, we compute the histograms gained from each group of $a \times a$ blocks as the final feature of each image X_i, followed by a classification of images based on a linear SVM. Generally, the hyper-parameters of SNnet include the patch size (or filter size) $a \times b$, the rank K_1, K_2 of the factorization (or the number of filters), the number of stages and the block size for histograms in the output layer.

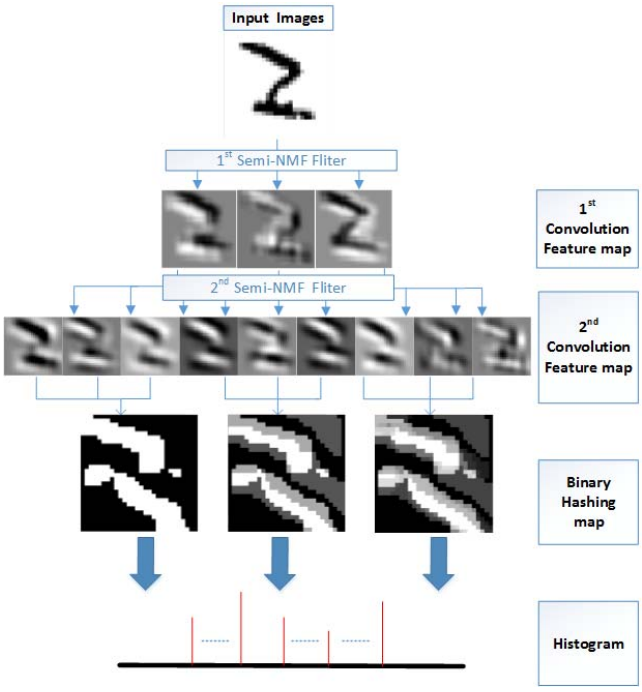

Fig. 2: Visual processing with the image number two.

3 Experiment on MNIST

The basic MNIST dataset [22] consists of $620,000$ grayscale images of handwritten digits from 0 to 9 and the size is 28×28. We divide dataset into 10,000 training samples, 2000 validation, and 50,000 testing samples. In this work, we randomly set up 5000 test samples and change the training size from 25 to 10000 to test the experimental errors under different training samples. It can be seen from Table 1 that the SNnet can obtain 28.98% error rates even with 25 training datas. At the same time, with the increase of the training sizes, the error decreases rapidly. By comparing the two models, we can find that SNnet with weakly supervised can achieve better results in the case of large-scale training samples. Fig. 2 clearly illustrates the processing of SNnet with the image number two in MNIST basic dataset.

In addition, we present the learned SNnet and S-SNnet filters in Fig. 3 and Fig. 4. In this work, we built 2 layers structure and set the number of filters to be $K_1 = 8$, $K_2 = 8$. The filter size of this network is $a = b = 7$, the block size is 8×8, and the overlapping region between blocks is half of the block size. A linear SVM classifier [23] is employed in this experiment. To compare the performance with the state-of-the-art methods, the parameters of our models are the same as PCANet. Table 2 shows the classification error rates on basic MNIST of different methods. SNnet achieves the test error rate of 1.00% when combining graph regularization with Semi-NMF. The results demonstrate our models are able to obtain the better result in comparison to the state-of-the-art methods.

Table 1: Comparison of Error Rates(%) Obtained by SNnet and S-SNnet on MNIST basic With Different Training Size

Training size	SNnet Error rate(%)	S-SNnet Error rate(%)
25	28.98	31.90
50	19.72	20.46
100	18.06	18.48
500	5.34	5.48
1000	3.72	2.76
2500	2.32	2.30
5000	1.70	1.38
10000	1.32	1.02

Table 2: Comparison of Error Rates(%) Obtained by Different Methods on MNIST basic Without Data Augmentation

Methods	Error rate(%)
DDL [20]	2.72
CAE [21]	2.48
CUNet + Average pooling[10]	1.90
CUNet + Weighted pooling[10]	1.80
CNN [20]	1.44
ScatNet [8]	1.27
PCANet [9]	1.07
S-SNnet (ours)	1.00

4 Conclusion

We propose a novel convolutional neural network structure based on semi non-negative matrix factorization algorithm, the SNnet, that can handle image classification tasks. The main innovation is that the convolutional filter in SNnet is learned by decomposing the input images. As a result, SNnet does not require numerical optimization solver and parameters tuning.

Moreover, an improved version of SNnet which combined graph regularization with Semi-NMF i.e., S-SNnet. Experiment on handwritten digits database (MNIST) verify our methods perform better than the state-of-the-art works. In future work, a main target is to extend the multi-layers SNnet, such that it can get the high-dimensional features and

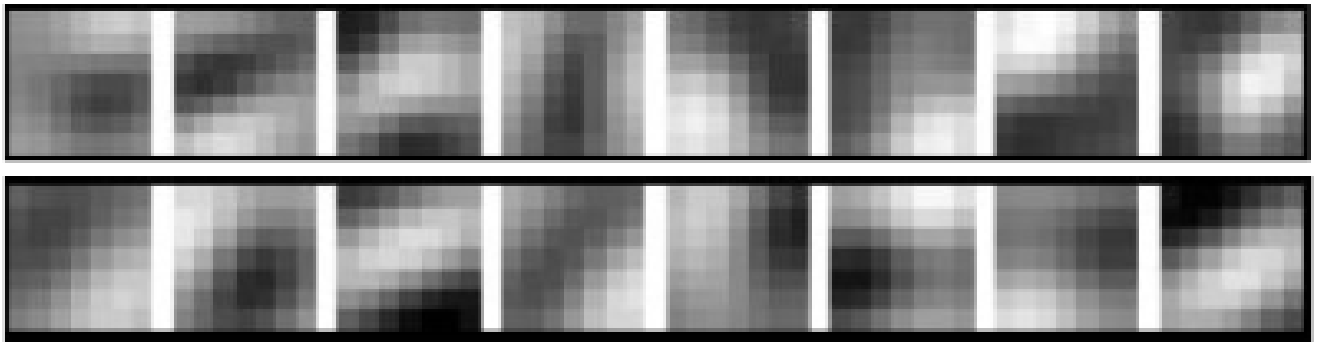

Fig. 3: The SNnet filters learned on MNIST basic dataset. The top row shows the filters of the first stage; the bottom row shows the filters of the second stage.

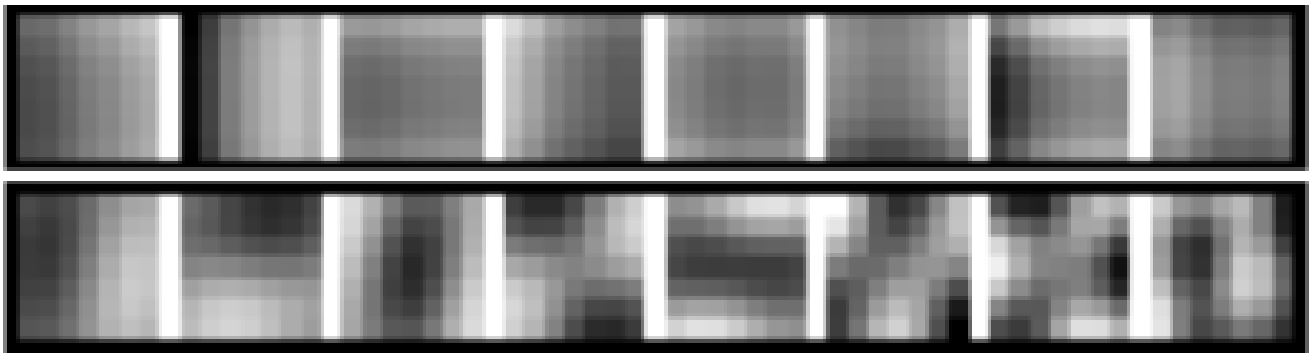

Fig. 4: The S-SNnet filters learned on MNIST basic dataset. The top row shows the filters of the first stage; the bottom row shows the filters of the second stage.

achieve higher accuracy. SNnet also provides a new way to solve the problem of traditional convolutional neural network which can not be proved by mathematics.

References

[1] K. Allab, L. Lazhar, and N. Mohamed, A Semi-NMF-PCA unified framework for data clustering, *IEEE Trans. on Knowledge and Data Engineering*, 29(1): 2–16, 2017.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 2012: 1097–1105.

[3] M. D. Zeiler, F. Rob, Visualizing and understanding convolutional networks, *European conference on computer vision*, 2014: 818–833.

[4] J. Long, S. Evan, and D. Trevor, Fully convolutional networks for semantic segmentation, *IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 3431–40.

[5] Z. Li, Z. Yang, and S. Xie, Computing Resource Trading for Edge-Cloud-assisted Internet of Things, *IEEE Trans. on Industrial Informatics*, DOI: 10.1109/TII.2019.2897364, 2019.

[6] C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan, and S. Xie, Reinforcement Learning-based Adaptive Optimal Exponential Tracking Control of Linear Systems with Unknown Dynamics, *IEEE Trans. on Automatic Control*, DOI: 10.1109/TAC.2019.2905215, 2019.

[7] M. Yin, J. Gao, S. Xie, and Y. Guo, Multiview Subspace Clustering via Tensorial t-Product Representation, *IEEE Trans. on Neural Networks and Learning Systems*, 30(3): 851–864, 2019.

[8] J. Bruna, S. Mallat, Invariant scattering convolution networks, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(8): 1872–1886, 2013.

[9] T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, PCANet: A simple deep learning baseline for image classification?, *IEEE Trans. on Image Processing*, 24(12): 5017–5032, 2015.

[10] L. Dong, L. He, and M. Mao, CUNet: A Compact unsupervised network for image classification, *IEEE Trans. on Multimedia*, 20(8): 2012–2021, 2018.

[11] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, 401: 788-791, 1999.

[12] Z. Yang, G. Zhou,S. Xie, S. Ding, J. Yang, and J. Zhang, Blind spectral unmixing based on sparse nonnegative matrix factorization, *IEEE Trans. on Image Processing*, 20(4): 1112–1125, 2010.

[13] J. Yang, Y. Guo, Z. Yang, and S. Xie, Under determined convolutive blind source separation combining density-based clustering and sparse reconstruction in time-frequency domain, *IEEE Trans. on Circuits and Systems I: Regular Papers*, DOI: 10.1109/TCSI.2019.2908394, 2019.

[14] Z. Yang, Y. Xiang, K. Xie, and Y. Lai, Adaptive method for nonsmooth nonnegative matrix factorization, *IEEE Trans. on Neural Networks and Learning Systems*, 28(4): 948-960, 2017.

[15] C. Ding, T. Li, and M.I. Jordan, Convex and semi-nonnegative matrix factorizations, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(1): 45–55, 2010.

[16] Z. Yang, Y. Zhang,Y. Xiang, W. Yan, and S. Xie, Non-negative matrix factorization with dual constraints for image clustering, *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, DOI: 10.1109/TSMC.2018.2820084, 2019.

[17] D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, *Advances in neural information processing systems*, 2001: 556–562.

[18] D. Cai, X. He, J. Han, and T. S. Huang, Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):

1548–1560, 2011.

[19] G. Trigeorgis, K. Bousmails, S. Zafeiriou, and B. W. Schuller, A deep matrix factorization method for learning attribute representations, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(3): 417–429, 2017.

[20] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa, Deep dictionary learning, *IEEE Access*, 4: 10096– 10109, 2016.

[21] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction, *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011: 833–840.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11): 2278-2324, 1998.

[23] 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*, 9(Aug): 1871-1874, 2008.