END-TO-END TRAINABLE WEAKLY NON-NEGATIVE FACTORIZATION

Takumi Kobayashi and Kenji Watanabe

National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Ibaraki, Japan

ABSTRACT

Non-negative matrix factorization (NMF) is widely applied to analyze pattern data in an unsupervised manner. It imposes *hard* non-negativity constraints on factors to extract intrinsic characteristics from an input matrix, though demanding complicated optimization techniques which hinder the general applicability. Toward flexible formulation, we propose *weakly* non-negative factorization. In contrast to the strict non-negative approach, our method permits factors to contain small amount of negative values. The relaxation theoretically leads to an efficient factorization formulation which can be implemented by means of off-the-shelf techniques used in a deep learning literature. Thus, the method is flexibly applicable to versatile factorization tasks, such as deep NMF and structured NMF. In the experiments on the NMF-related tasks, we demonstrate that the weak non-negativity produces effective factors similarly to NMF and the method exhibits favorable performance in comparison to the other approaches.

***Index Terms*—** factorization, weak non-negativity, regularization, leaky ReLU

1. INTRODUCTION

Real-world pattern data, such as images and their feature vectors, are described in a redundant form and thus mining their intrinsic representation is useful for pattern analysis. It is plausible to make a low-rank assumption that data are constructed on the basis of only a small number of factors, which derives diverse factorization methods. While a naive factorization is given by singular value decomposition (SVD) or PCA, a non-negative matrix factorization (NMF) [1] has been successfully applied in various fields such as signal processing and computer vision [2]. Considering that real-world data is frequently represented by *non-negative* physical quantities (e.g., pixel intensities), NMF reveals from pattern matrices inherent characteristics such as sparsity [1].

NMF is a general framework to optimize two non-negative factors so that an input matrix is well approximated by their product. It is formulated as an optimization problem with non-negativity constraints as shown in the mathematical form (1). Thus, the problem is addressed by various optimization approaches [2, 3] to cope with the constraints, and one of the popular techniques is a multiplicative updating [4, 5] which naturally involves non-negativity into parameter updates. It is dependent on a cost function measuring reconstruction errors; MSE [4] and β-divergence [5] are favorable for the multiplicative updating. However, such complicated optimization techniques and/or dependency on loss functions would degrade general applicability of NMF such as to a (deep) end-to-end framework [6].

In this paper, we propose a flexible non-negative factorization such that it can be embedded in versatile frameworks via end-to-end learning. The non-negativity constraint, which is fundamental for NMF, requires careful treatment by the optimization technique, making simple gradient descent useless. We formulate *weakly* non-negative factorization by relaxing the hard constraint of non-negativity into *regularization* through reformulation of NMF with re-parameterized factors. It permits factors to contain *small* amount of negative elements while NMF imposes strictly non-negative constraints on the factors. The weakly non-negative factorization is formulated as an unconstrained problem to which simple optimization process, such as back-prop gradient descent, is effectively applicable. It is thus noteworthy that the proposed method can be implemented by using well-established deep learning techniques including optimizers which are intensively studied in recent years [7]. Our flexible formulation is directly applicable to versatile factorization tasks such as deep NMF [8] and structured NMF.

2. METHOD

We start with a constrained optimization problem for NMF and then formulate an *unconstrained* problem to address *weakly* non-negative factorization.

2.1. Optimization problem for NMF

Suppose an input matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is decomposed into two factors $\boldsymbol{W} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{H} \in \mathbb{R}^{r\times n}$ as $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$ with rank r. By introducing non-negativity constraints into factors [1], we formulate the following optimization problem for NMF;

$$\min_{\boldsymbol{W},\boldsymbol{H}} \ell(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}), \quad s.t.\, \boldsymbol{W} \geq 0,\, \boldsymbol{H} \geq 0, \tag{1}$$

where an inequality $\boldsymbol{A} \geq 0$ indicates element-wise non-negativity for all components in a matrix $\boldsymbol{A}$. The loss function ℓ measures discrepancy between $\boldsymbol{X}$ and $\boldsymbol{W}\boldsymbol{H}$; for NMF, one can usually employ Euclidean distance or β-divergence to induce an efficient multiplicative updating [4, 5]. For robust factorization, L_2-norm regularization with a parameter λ is imposed on the factors as

$$\min_{\boldsymbol{W},\boldsymbol{H}} \ell(\boldsymbol{X},\boldsymbol{W}\boldsymbol{H})+\lambda(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{H}\|_F^2),\ s.t.\boldsymbol{W}\geq 0,\ \boldsymbol{H}\geq 0. \tag{2}$$

2.2. Weakly non-negative factorization

In order to be further aware of non-negativity, we disentangle the factors into

$$\boldsymbol{W} = \boldsymbol{W}_+ - \boldsymbol{W}_-,\ \ \boldsymbol{H} = \boldsymbol{H}_+ - \boldsymbol{H}_-, \tag{3}$$

where $\boldsymbol{A}_+$ and $\boldsymbol{A}_-$ indicate non-negative matrices responsible for positive and negative parts of a matrix $\boldsymbol{A}$, respectively. By using this re-parameterization, similarly to (2), we formulate the following optimization problem;

$$\begin{aligned}\min_{\boldsymbol{W}_+,\boldsymbol{W}_-,\boldsymbol{H}_+,\boldsymbol{H}_-} & \ell[\boldsymbol{X},(\boldsymbol{W}_+ - \boldsymbol{W}_-)(\boldsymbol{H}_+ - \boldsymbol{H}_-)] & (4)\\ & + \lambda(\|\boldsymbol{W}_+\|_F^2 + \|\boldsymbol{H}_+\|_F^2) + \lambda_-(\|\boldsymbol{W}_-\|_F^2 + \|\boldsymbol{H}_-\|_F^2), & \\ & s.t.\ \boldsymbol{W}_+ \geq 0,\ \boldsymbol{W}_- \geq 0,\ \boldsymbol{H}_+ \geq 0,\ \boldsymbol{H}_- \geq 0, & (5)\end{aligned}$$

where we additionally introduce a regularization parameter λ_- for the negative part of factors. It should be noted that setting $\lambda_- \to \infty$ enforces $\boldsymbol{W}_- \to 0$ and $\boldsymbol{H}_- \to 0$ to derive the original NMF (2). In other words, the parameter λ_- controls *non-negativity*, which thereby motivates us to formulate *weakly non-negative* matrix factorization by applying moderately large λ_- in comparison to λ. It should be noted that unlike NMF, in a weakly non-negative matrix, each element is not strictly non-negative but can take *small* negative value if it falls into negative part. The amount of negativity on the factors is suppressed by the large λ_-, especially $\lambda/\lambda_- < 1$.

We further rewrite the negative part matrices $\boldsymbol{W}_-$ and $\boldsymbol{H}_-$ by using $\eta = \sqrt{\lambda/\lambda_-}$ as

$$\boldsymbol{W}_- = \sqrt{\lambda/\lambda_-}\hat{\boldsymbol{W}}_- = \eta\hat{\boldsymbol{W}}_-,\ \ \boldsymbol{H}_- = \eta\hat{\boldsymbol{H}}_-, \tag{6}$$

to simplify (4) into

$$\begin{aligned}\min_{\boldsymbol{W}_+,\hat{\boldsymbol{W}}_-,\boldsymbol{H}_+,\hat{\boldsymbol{H}}_-} & \ell[\boldsymbol{X},(\boldsymbol{W}_+ - \eta\hat{\boldsymbol{W}}_-)(\boldsymbol{H}_+ - \eta\hat{\boldsymbol{H}}_-)] & (7)\\ & + \lambda(\|\boldsymbol{W}_+\|_F^2 + \|\boldsymbol{H}_+\|_F^2 + \|\hat{\boldsymbol{W}}_-\|_F^2 + \|\hat{\boldsymbol{H}}_-\|_F^2), & \\ & s.t.\ \boldsymbol{W}_+ \geq 0,\ \hat{\boldsymbol{W}}_- \geq 0,\ \boldsymbol{H}_+ \geq 0,\ \hat{\boldsymbol{H}}_- \geq 0. & (8)\end{aligned}$$

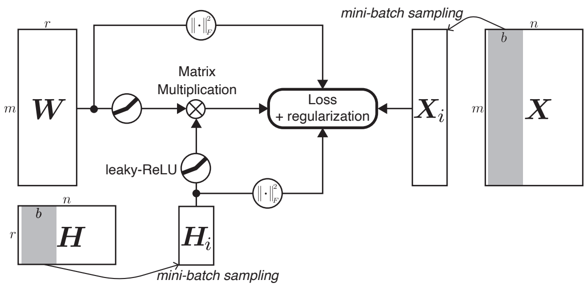


Fig. 1. Weakly non-negative matrix factorization. It is implemented in an end-to-end mini-batch training with size b.

In this optimization problem, Karush Kuhn–Tucker (KKT) conditions [9] imply that $\boldsymbol{W}_+ \odot \hat{\boldsymbol{W}}_- = \boldsymbol{0}$ and $\boldsymbol{H}_+ \odot \hat{\boldsymbol{H}}_- = \boldsymbol{0}$, meaning that each element either of $\boldsymbol{W}_+$ or $\hat{\boldsymbol{W}}_-$ is zero. We can plug this relationship into (7) to provide

$$\min_{\boldsymbol{W},\boldsymbol{H}} \ell[\boldsymbol{X},\phi_\eta(\boldsymbol{W})\phi_\eta(\boldsymbol{H})] + \lambda(\|\boldsymbol{W}\|_F^2 + \|\boldsymbol{H}\|_F^2), \tag{9}$$

where $\boldsymbol{W}$ and $\boldsymbol{H}$ are real-valued matrices and a non-linear function ϕ_η works on a matrix in an element-wise manner as

$$\phi_\eta(A_{ij}) = \begin{cases} A_{ij} & \text{if } A_{ij} \geq 0 \\ \eta A_{ij} & \text{if } A_{ij} < 0 \end{cases} = \max[A_{ij}, \eta A_{ij}], \tag{10}$$

where $\eta < 1$ ($\Leftrightarrow \lambda/\lambda_- < 1$). The proposed formulation (9, 10) is advantageous in the following two points.

First, the function ϕ_η is equivalent to *leaky ReLU* [10] which is well established and frequently used in a deep learning literature. Due to $\eta = \sqrt{\lambda/\lambda_-}$, the parameter η controls non-negativity of factors. The original NMF (2) is obtained simply by setting $\eta = 0$ which corresponds to $\lambda_- \to \infty$ in (4). In case of $\eta = 0$, however, the non-linear function $\phi_{\eta=0}$ is reduced into *ReLU* [11], making the problem (9) harder to optimize. Due to the ReLU function being *flat* on a negative part, once some elements fall into the negative part, they hardly enjoy favorable loss gradients. It would accordingly demand some optimization techniques, such as projected gradients [3], which degrades the flexibility of the formulation (9). Therefore, we simply derive small $\eta \ll 1$ from large $\lambda_- \gg 1$ for rendering *weakly* non-negative factorization. In the another approach, it is conceivable to tune the parameter η during training as in curriculum learning [12]; η is gradually decreased toward 0 so that ϕ_η eventually results in ReLU. These approaches are compared in the experiments of Sec. 3.

Second, the formulation (9) addresses NMF (2) in a way of *unconstrained* optimization. Together with the leaky ReLU ϕ_η, it is implemented in a end-to-end framework as shown in Fig. 1. In contrast to the standard end-to-end method [13], the proposed method does not feed a pattern data, e.g., images, to the model but just samples a subset $\boldsymbol{H}^{(i)} \in \mathbb{R}^{r\times b}$ from the whole factor $\boldsymbol{H} \in \mathbb{R}^{r\times n}$ in an efficient mini-batch learning with the mini-batch size b. Besides, off-the-shelf deep learning techniques are directly applied to optimize it. For example, we can flexibly employ various types of loss function ℓ, not limited to MSE and β-divergence, as well as leverage sophisticated optimizers to minimize the loss; we apply AdamW [7] with a learning rate of 0.001 to an L_1 loss of $\ell(x,y) = \|x-y\|_1$. The proposed method is so general that the weakly non-negative factorization can be embedded in various tasks as presented in Sec. 3.

Table 1. Datasets.

	# sample	# dimension	# class
ORL face	400	92×112 pixels	40
AR face	2,600	$120 \times 165 \times 3$ pixels	100
multi-feature	2,000	585 features	10
ImageNet-CNN	50,000	2,048 features	1,000

Table 2. Ablation study. Performance score is $100\times$ NMI.

k				$\phi_\eta(x) = \max[x, \eta x]$		
20	30	40	50	η	λ	ℓ
67.57	75.19	80.20	81.57	1 linear	0	$\|\cdot\|_1$
52.17	58.76	57.71	59.82	0 ReLU	0	$\|\cdot\|_1$
72.05	79.78	85.73	85.95	0.1 leaky-ReLU	0	$\|\cdot\|_1$
76.64	83.35	88.71	87.90	1 linear	0.1	$\|\cdot\|_1$
75.71	83.36	89.08	89.25	0 ReLU	0.1	$\|\cdot\|_1$
78.12	**86.67**	**89.80**	**91.26**	0.1 leaky-ReLU	0.1	$\|\cdot\|_1$
75.24	82.04	86.92	88.29	$1 \rightsquigarrow 0$ curriculum	0.1	$\|\cdot\|_1$
75.36	79.28	84.27	87.27	0.1 leaky-ReLU	0.1	$\|\cdot\|_2^2$

3. EXPERIMENTAL RESULTS

We apply the proposed method to three types of non-negative factorization; simple NMF (Sec. 3.1), deep NMF (Sec. 3.2) and deconvolutional factorization (Sec. 3.3).

Dataset. Factorization methods are tested on four datasets (Table 1). ORL face dataset [14] contains 400 images of 92×112 gray-scaled pixels captured from 40 subjects. AR face dataset [15] is constructed by taking 2,600 images of 120×165 RGB-color pixels from 100 subjects under various conditions. Multiple-feature dataset [16] contains 2,000 samples of 585-dimensional non-negative features[1] in 10 classes. We built an ImageNet-CNN dataset by applying pre-trained ResNet-50 [17] to ImageNet [18] validation set to provide 50,000 samples of 2,048-dimensional feature vectors; note that ResNet-50 excluding the last FC layer intrinsically produces non-negative features due to ReLU activation.

Evaluation. For quantitatively evaluating the factorization results, we apply k-means clustering to n latent samples of $\boldsymbol{H} = \{\boldsymbol{h}_i \in \mathbb{R}^r\}_{i=1}^n$ and then measure normalized mutual information (NMI) [19] between the clusters and class labels.

3.1. Non-negative matrix factorization (NMF)

The method (9) is straightforwardly applied to the task of NMF as $\boldsymbol{X} \approx \phi_\eta(\boldsymbol{W})\phi_\eta(\boldsymbol{H})$.

Ablation study. We first analyze the proposed method on ORL face dataset [14]. As shown in (9), the method is distinctive in terms of the non-linear function $\phi_\eta(x) = \max[x, \eta x]$ and the regularization with λ. We evaluate performance of the method in an ablation manner regarding those two aspects in Table 2 where the function ϕ_η provides three types of activation, linear ($\eta = 1$), ReLU ($\eta = 0$) and leaky-ReLU ($\eta = 0.1$), and we set $\lambda = 0.1$ to inject L_2 regularization.

Without regularization ($\lambda = 0$), ReLU activation ($\eta = 0$) significantly degrades performance. In this factorization, the factors $\boldsymbol{W}$ and $\boldsymbol{H}$ are simply initialized by uniform values including negatives according to the successful training recipe [20]. The ReLU activation provides no updating with the initially negative parameters, thereby failing to optimize the factorization. Without regularization, leaky ReLU ($\eta = 0.1$) poorly works in terms of both performance and non-negativity; Fig. 2 indicates that it fails to produce non-negative factors. Combining leaky ReLU ($\eta = 0.1$) with the regularization ($\lambda = 0.1$) significantly improves performance.

Even though the method is based on *weak* non-negativity, the learnt factors are almost non-negative. We show distribution of weakly non-negative factor components in Fig. 2. Both factors, $\phi_\eta(\boldsymbol{W})$ and $\phi_\eta(\boldsymbol{H})$, are favorably *non-negative*; actually, $\min_{W \in \boldsymbol{W}} \phi_\eta(W) = -0.008$ and $\min_{H \in \boldsymbol{H}} \phi_\eta(H) = -0.03$, meaning that factors are *almost* non-negative. The basis factor $\phi_\eta(\boldsymbol{W})$ are visualized in Fig. 3a. Similarly to NMF factors (Fig. 3b), it extracts sparse parts of face appearance due to the (weak) non-negativity.

The parameter η is set to small $\eta = 0.1$ for weakly non-negative factorization. We also evaluate the curriculum approach (Sec. 2.2) by gradually decreasing η as $1 \rightsquigarrow 0$ to ensure strict non-negativity of factors in the end. As shown in Table 2, it is inferior to $\eta = 0.1$ and even to $\eta = 0$, demonstrating that the constant-η works well with the regularization without tuning curriculum schedule of η.

The proposed formulation (9) can accept various types of loss function ℓ and it is so far set to an L_1 loss, $\ell(x, y) = \|x - y\|_1$. We compare the L_1 loss with L_2 one which is widely applied in an NMF framework [1, 4] and Table 2 shows superiority of the L_1 loss. Our framework is effectively flexible in terms of designing loss functions for improving performance.

Performance comparison. The proposed method is compared with the other NMF methods including the original NMF [4] that applies multiplicative updating to non-negative factors, and unfolding NMF [21] which unfolds such multiplicative updating in a deep framework. The performance results in Table 3a demonstrate the efficacy of our method.

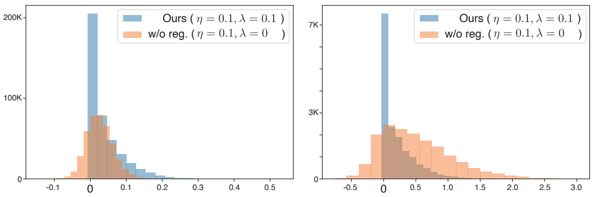
3.2. Deep non-negative matrix factorization

We apply the method in a framework of deep NMF [22] by stacking non-negative factorization in multiple layers. The proposed method is naturally applicable to stack L-factorization as, given $\boldsymbol{X} \in \mathbb{R}^{m \times n}$,

$$\boldsymbol{X} \approx \phi_\eta(\boldsymbol{W}_L) \cdots \phi_\eta(\boldsymbol{W}_1)\phi_\eta(\boldsymbol{H}_1), \tag{11}$$

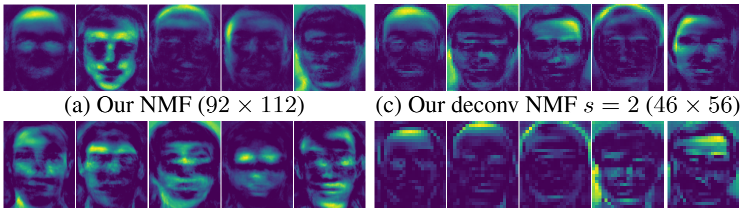
where $\boldsymbol{W}_l \in \mathbb{R}^{r_{l+1} \times r_l}$ and $\boldsymbol{H}_1 \in \mathbb{R}^{r_1 \times n}$ with $r_{L+1} = m$ and $l \in \{1, \cdots, L\}$. The latent representation at the l-th

[1] We exclude 64 real-value components from whole 649 features.



(a) $\phi_\eta(\boldsymbol{W})$ (b) $\phi_\eta(\boldsymbol{H})$

Fig. 2. Histograms of factor elements.



(a) Our NMF (92 × 112) (c) Our deconv NMF $s = 2$ (46 × 56)

(b) NMF (92 × 112) (d) Our deconv NMF $s = 4$ (23 × 28)

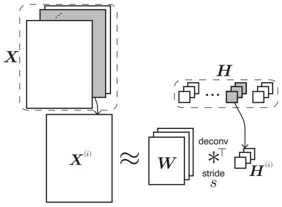
Fig. 3. Learnt factors $\boldsymbol{H}$ whose resolutions are shown in parentheses.

Table 3. Performance comparison on tasks of NMF (a) and deep NMF (b). For deep NMF, two-layer ranks (r_1, r_2) are set to $(20, 40)$ on ORL, $(50, 100)$ on AR, $(10, 20)$ on multi-feature and $(500, 1000)$ on ImageNet-CNN.

		ORL				AR				multi-feature				ImageNet-CNN			
	k	20	30	40	50	50	75	100	150	2	5	10	20	500	750	1000	1500
(a) NMF	NMF	67.12	75.36	82.51	84.79	33.08	39.37	43.19	48.59	29.21	43.73	65.78	63.72	75.63	78.89	80.12	80.57
	Unfold [21]	66.03	70.46	73.84	74.54	33.17	37.34	40.81	44.74	26.64	51.23	61.95	63.91	57.63	59.38	60.76	62.82
	Ours (9)	78.12	86.67	89.80	91.26	32.88	39.43	44.31	52.36	37.61	68.75	85.51	76.15	77.11	80.05	81.01	81.16
(b) deep NMF	NMF $l = 1$	74.19	77.99	82.06	84.82	32.10	38.83	43.14	49.22	33.95	55.98	62.73	64.59	35.51	39.60	42.23	45.68
	NMF $l = 2$	71.09	78.89	79.63	84.00	30.32	37.42	41.63	48.13	30.90	53.86	64.94	66.41	35.83	39.71	42.37	45.75
	Ours $l = 1$	74.45	82.59	84.20	86.11	30.64	39.18	42.84	49.72	37.39	66.85	84.11	76.99	76.84	79.88	80.94	81.03
	Ours $l = 2$	74.79	80.01	83.74	85.67	31.98	37.33	41.58	47.97	36.62	67.23	78.60	76.37	76.93	79.85	81.16	81.25

Table 4. Deconvolutional factorization on ORL dataset.

(a) Framework



(b) Performance results

	k	20	30	40	50
raw	$s = 2$	74.60	81.53	85.39	86.93
	$s = 4$	71.75	79.73	83.31	83.87
sum	$s = 2$	76.40	84.34	88.98	89.31
	$s = 4$	79.08	82.72	87.12	89.50

layer is given by $\phi_\eta(\boldsymbol{W}_{l-1}) \cdots \phi_\eta(\boldsymbol{W}_1)\phi_\eta(\boldsymbol{H}_1)$; the $l = 1$-st layer provides simply $\phi_\eta(\boldsymbol{H}_1)$ as latent. While there are some variants of deep NMF incorporating additional techniques [8, 23, 24], for fair comparison, the proposed method is compared with a deep NMF which updates factors in a multiplicative form with L_2 regularization of factors. In this experiment, we apply $L = 2$-layered NMF and report performance in Table 3b; two ranks (r_1, r_2) on respective datasets are shown in the caption. Our method favorably works on decomposition of an input matrix into multiple factors.

3.3. Deconvolutional factorization

Finally, we leverage our flexible method to factorize images in a deconvolutional manner. A standard NMF (Sec. 3.1) explores factor images $\boldsymbol{W}$ of the *same resolution* as input images. For further efficient representation, we extract *smaller-sized* factor images which can reconstruct an input image by means of deconvolution [25] equipped with step (upscaling) size s and kernel filters of $2s \times 2s$ size to be optimized as well. The deconvolutional factorization is described as

$$\boldsymbol{X}^{(i)} \approx \phi_\eta(\boldsymbol{W}) *_s^T \phi_\eta(\boldsymbol{H}^{(i)}), \tag{12}$$

where $*_s^T$ indicates a deconvolutional operator with step size s which means upscaling by a factor of s, the i-th image is denoted by $\boldsymbol{X}^{(i)} \in \mathbb{R}^{h \times w}$, and two factors of rank r are given as $\boldsymbol{W} \in \mathbb{R}^{\frac{h}{s} \times \frac{w}{s} \times r}$ and $\boldsymbol{H}^{(i)} \in \mathbb{R}^{2s \times 2s \times r}$. This factorization contrasts with [26, 27] which address convolutional decomposition of a single temporal sequence without upscaling. The proposed deconvolutional factorization is evaluated on ORL face dataset to report performance in Table 4 as well as show factors in Fig. 3cd. The *raw* latent space is straightforwardly given by flattening $\boldsymbol{H}^{(i)}$ into $4s^2r$-dimensional feature vector. The factor $\boldsymbol{H}^{(i)}$, however, contains detailed upscaling filter weights which are less relevant to latent representation. Thus, for more favorable latent space, we measure significance of factors at respective ranks by marginalizing out spatial filter elements $\{\sum_{x,y} H^{(i)}_{xyr'}\}_{r'=1}^{r}$ to produce r-dimensional latent vectors akin to the standard NMF latent ones (Sec. 3.1); the approach is denoted by *sum* in Table 4. The factors (Fig. 3d) are ×16 smaller than those of NMF (Fig. 3a) while producing favorable performance in comparison to Table 3a.

4. CONCLUSION

We have proposed a weakly non-negative factorization method. The method is theoretically derived from an NMF optimization formulation by relaxing non-negativity constraints into regularization via re-parameterization of factors. It efficiently provides almost non-negative factors by means of off-the-shelf deep learning techniques in an end-to-end fashion. The experimental results demonstrate that the proposed method is flexibly applicable to versatile NMF-related frameworks and produces favorable performance on those tasks.

5. REFERENCES

[1] Daniel D. Lee and Hyunjune S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[2] Yu-Xiong Wang and Yu-Jin Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[3] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan, "Nenmf: An optimal gradient method for non-negative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2882–2898, 2012.

[4] Daniel D. Lee and Hyunjune S. Seung, "Algorithms for non-negative matrix factorization," in *NeurIPS*, 2000, pp. 556–562.

[5] Cédric Févotte and Jérôme Idier, "Algorithms for non-negative matrix factorization with the β-divergence," *Neural Comput.*, vol. 23, no. 9, pp. 2421–2456, 2011.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[7] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[8] Pierre De Handschutter, Nicolas Gillis, and Xavier Siebert, "A survey on deep matrix factorizations," *Computer Science Review*, vol. 42, pp. 100423, 2021.

[9] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[10] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013.

[11] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[12] Yoshua Bengio dn Jerome Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *ICML*, 2009.

[13] Paris Smaragdis and Shrikant Venkataramani, "A neural network alternative to non-negative audio models," in *ICASSP*, 2017, pp. 86–90.

[14] F.S. Samaria and A.C. Harter, "Parameterisation of a stochastic model for human face identification," in *IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.

[15] Aleix M. Martínez and Avinash C. Kak, "Pca versus lda," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, 2001.

[16] Dheeru Dua and Casey Graff, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[19] Haifeng Liu, Zhaohui Wu, Xuelong Li, Deng Cai, and Thomas S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, 2012.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015.

[21] Rami Nasser, Yonina C. Eldar, and Roded Sharan, "Deep unfolding for non-negative matrix factorization with application to mutational signature analysis," *J. Comput. Biol.*, vol. 29, no. 1, pp. 45–55, 2022.

[22] Andrzej Cichocki and Rafal Zdunek, "Multilayer non-negative matrix factorisation," *Electronics Letters*, vol. 42, no. 16, pp. 947–948, 2006.

[23] Jianyong Sun, Qingming Kong, and Zongben Xu, "Deep alternating non-negative matrix factorisation," *Knowledge-Based Systems*, vol. 251, pp. 109210, 2022.

[24] Pierre De Handschutter, Nicolas Gillis, and Xavier Siebert, "Hierarchical feature extraction by multi-layer non-negative matrix factorization network for classification task," *Neurocomputing*, vol. 165, pp. 63–74, 2015.

[25] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus, "Deconvolutional networks," in *CVPR*, 2010.

[26] Paris Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *ICA*, 2004, pp. 494–499.

[27] Mikkel N. Schmidt and Morten Mørup, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *ICA*, 2006, pp. 700–707.