Neural Networks 103 (2018) 63–71

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A nonnegative matrix factorization algorithm based on a discrete-time projection neural network ☆

Hangjun Che, Jun Wang *

Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Hong Kong
Shenzhen Research Institute, City University of Hong Kong, Shenzhen, China

ARTICLE INFO

Article history:
Received 13 June 2017
Received in revised form 18 December 2017
Accepted 6 March 2018
Available online 20 March 2018

Keywords:
Nonnegative matrix factorization
Discrete-time projection neural network
Biconvex optimization

ABSTRACT

This paper presents an algorithm for nonnegative matrix factorization based on a biconvex optimization formulation. First, a discrete-time projection neural network is introduced. An upper bound of its step size is derived to guarantee the stability of the neural network. Then, an algorithm is proposed based on the discrete-time projection neural network and a backtracking step-size adaptation. The proposed algorithm is proven to be able to reduce the objective function value iteratively until attaining a partial optimum of the formulated biconvex optimization problem. Experimental results based on various data sets are presented to substantiate the efficacy of the algorithm.

1. Introduction

Non-negative matrix factorization (NMF) aims to decompose a high-dimensional matrix $V \in R_+^{m\times n}$ into two low-rank matrices such that $V \approx WH$, where $W \in R_+^{m\times r}$, $H \in R_+^{r\times n}$ and $1 \le r < \min(m, n)$ (Lee & Seung, 1999). NMF is regarded as an effective technique to reduce data dimensions and discover part-based representations (Arabnejad, Moghaddam, & Cheriet, 2017; Cai, He, Han, & Huang, 2011; Fan & Wang, 2017; Zhu & Honeine, 2016).

NMF is usually formulated as a constrained optimization problem as follows:

$$\begin{aligned} &\min \quad f(W,H) \\ s.t. \quad & W \ge 0, \quad H \ge 0, \end{aligned} \tag{1}$$

where $f(W, H)$ is an objective function. Two most popular objective functions are used in problem (1): one is squared Frobenius norm of the factorization error matrix (Lee & Seung, 1999):

$$f_1(W,H) = \frac{1}{2}\|V - WH\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm, and the other is the Kullback–Leibler (K–L) divergence (Lee & Seung, 2001):

$$\begin{aligned} f_2(W,H) &= D(V \,\|\, WH) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\Big(V_{ij}\log\frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij}\Big). \end{aligned} \tag{3}$$

In Lee and Seung (2001), an NMF algorithm called multiplicative update rule (MUR) is proposed. MUR is simple to implement, but it may fail to converge to a stationary point or converge slowly (Lin, 2007). As an alternative, alternating least squares (ALS) is presented based on alternating least squares (Berry, Browne, Langville, Pauca, & Plemmons, 2007). However, ALS requires a high computational cost and the convergence is not guaranteed (Kim & Park, 2008a). To expedite convergence, projected gradient (PG) method is proposed based on reformulating NMF as two nonnegative least-squares subproblems (Lin, 2007). Active set (AS) method and block principal pivoting (BPP) method are presented for NMF in Kim and Park (2008a, b), respectively. AS and BPP algorithms expedite convergence, but they suffer from numerical instability if W and H^T are not full column rank (Guan, Tao, Luo, & Yuan, 2012). In Guan et al. (2012), an NMF algorithm called NeNMF based on Nesterov's optimal gradient method with a convergence rate $O(1/k^2)$ is proposed. An important factor of NeNMF is the Lipschitz constant, but it is difficult to compute it if the objective function is complex. In Cichocki, Zdunek, and Amari (2006), Fvotte, Bertin, and Durrieu (2009) and Nakano et al. (2010) three general divergence functions are proposed to measure the quality of NMF and the K–L function is the special case of them. Essentially, the NMF algorithms based on

☆ This work was supported in part by the Research Grants Council, Hong Kong, under Grants 14207614 and 11208517, and in part by the National Natural Science Foundation of China under Grant 61673330.

* Corresponding author at: Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Hong Kong.

E-mail addresses: hjche2-c@my.city.edu.hk.com (H. Che), jwang.cs@cityu.edu.hk (J. Wang).

https://doi.org/10.1016/j.neunet.2018.03.003
0893-6080/

the general divergence functions are the variants of MUR (Cichocki et al., 2006; Fvotte et al., 2009; Nakano et al., 2010).

As a parallel optimization approach, neurodynamic optimization shows superior search ability in various optimization problems such as linear programming (He, Li, Huang, Li, & Huang, 2014; Liu, Cao, & Chen, 2010; Liu & Wang, 2008), convex programming (Che, Li, He, & Huang, 2016; Liu & Wang, 2011, 2016; Wang, 1994; Xia, Feng, & Wang, 2004; Xia & Wang, 2004), nonsmooth convex optimization (Li, Yan, & Wang, 2014; Liu & Wang, 2013; Qin, Bian, & Xue, 2013) and constrained nonconvex optimization (Li, Yan, & Wang, 2015). In recent years, collaborative neurodynamic optimization method based on multiple recurrent neural networks searching for the global optimal solutions cooperatively with the help of particle swarm optimization is proposed. The collaborative approach shows superior performance in searching for the global optima (Che, Li, He, & Huang, 2015; Liu, Yang, & Wang, 2017; Yan, Fan, & Wang, 2017; Yan, Wang, & Li, 2014).

In view of the powerful computing ability of neural networks, this paper presents an NMF algorithm based on a discrete-time projection neural network. Theoretical and experimental results show the efficacy of the algorithm.

The remainder of this paper is organized as follows. In Section 2, continuous-time projection neural network and basic concepts of biconvex optimization are introduced. A discrete-time projection neural network (DTPNN) is developed and analyzed in Section 3. An algorithm for NMF based on DTPNN is proposed in Section 4. Experimental results are discussed in Section 5. Conclusions are given in Section 6.

2. Preliminaries

2.1. Continuous-time projection neural network

Consider the following optimization problem:

$$\min \quad f(x) \quad s.t. \quad l \le x \le u. \tag{4}$$

A one-layer projection neural network can be used to solve (4) (Xia & Wang, 2000):

$$\epsilon \frac{dx}{dt} = -x + g(x - \nabla f(x)) \tag{5}$$

where $\epsilon > 0$ is a time constant, $\nabla f(x)$ denotes the gradient of f and $g(\cdot)$ is a piecewise activation function defined as follows:

$$g(\zeta_i) = \begin{cases} l_i, & \zeta_i < l_i \\ \zeta_i, & l_i \le \zeta_i \le u_i \\ u_i, & \zeta_i > u_i. \end{cases}$$

In particular for NMF, u_i is ∞ and l_i is 0. Therefore, $g(\cdot)$ is the activation function called rectified linear unit defined as follows:

$$g(\zeta_i) = \begin{cases} 0, & \zeta_i < 0 \\ \zeta_i, & \zeta_i \ge 0. \end{cases} \tag{6}$$

The following lemma shows the property of the continuous-time projection neural network (5).

Lemma 1 (*Hu & Wang, 2006*)**.** *If $f(x)$ is pseudoconvex and twice continuously differentiable on the closed convex set Ω, then the projection neural network (5) is stable in the sense of Lyapunov and globally convergent to the optimal solution of (4).*

2.2. Biconvex optimization

Definition 1. The set $\mathcal{B} \subset \mathcal{X} \times \mathcal{Y}$ is called biconvex set on $\mathcal{X} \times \mathcal{Y}$, if $\mathcal{B}_x$ is convex for every $x \in \mathcal{X}$ and $\mathcal{B}_y$ is convex for every $y \in \mathcal{Y}$, where $\mathcal{X} \subseteq R^n$ and $\mathcal{Y} \subseteq R^m$ are two non-empty convex sets. $\mathcal{B}_x$ and $\mathcal{B}_y$ are two sections of $\mathcal{B}$ defined as follows:

$$\mathcal{B}_x = \{(x, y) \in \mathcal{B} | y \in \mathcal{Y}\}, \mathcal{B}_y = \{(x, y) \in \mathcal{B} | x \in \mathcal{X}\}.$$

Definition 2. A function $f(x, y)$: $\mathcal{B} \to R$ is called a biconvex function on $\mathcal{B} \subseteq \mathcal{X} \times \mathcal{Y}$, if $f(x, \cdot) : \mathcal{B}_x \to R$ is a convex function on $\mathcal{B}_x$ for every fixed $x \in \mathcal{X}$ and $f(\cdot, y) : \mathcal{B}_y \to R$ is a convex function on $\mathcal{B}_y$ for every fixed $y \in \mathcal{Y}$.

Definition 3. A biconvex optimization problem is defined as follows:

$$\min f(x, y) \qquad s.t. \quad (x, y) \in \mathcal{B}, \tag{7}$$

where the feasible set $\mathcal{B}$ is biconvex on $\mathcal{X} \times \mathcal{Y}$, and $f(x, y)$ is a biconvex function on $\mathcal{B}$. In this paper, we assume that $f(x, y)$ is twice differentiable on $\mathcal{B}$.

Definition 4. Let f: $\mathcal{B} \to R$ be a given function and $(x^*, y^*) \in R^n \times R^m$, (x^*, y^*) is called a partial optimum of f on $\mathcal{B}$, if
$f(x^*, y^*) \le f(x, y^*)\ \forall x \in \mathcal{B}_{y^*}$ and $f(x^*, y^*) \le f(x^*, y)\ \forall y \in \mathcal{B}_{x^*}$.

Definition 5. Let f: $R^n \to R$ be a given function and $\xi \in R^n$. If the partial derivatives of f at ξ exist and $\nabla f(\xi) = 0$, ξ is called a stationary point of f.

Lemma 2 (*Gorski, Pfeuffer, & Klamroth, 2007*)**.** *Let f: $\mathcal{B} \to R$ be partial differentiable at $z^* \in int(\mathcal{B})$ and let z^* be a partial optimum. z^* is a stationary point of f in $\mathcal{B}$.*

Lemma 3 (*Gorski et al., 2007*)**.** *Let $\mathcal{B}$ be a biconvex set and $f : \mathcal{B} \to R$ is a differentiable biconvex function. Each stationary point of f is a partial optimum.*

In problem (1), feasible region is a biconvex set, to show this is a biconvex optimization problem, we need to prove the biconvexity of $f_1(W, H)$ and $f_2(W, H)$. In Guan et al. (2012), $f_1(W, H)$ is proven to be biconvex, here we prove the biconvexity of $f_2(W, H)$.

Lemma 4. *$f_2(W, H)$ is biconvex.*

Proof. The K–L divergence objective function for nonnegative matrix factorization is as follows:

$$\begin{aligned} f = \sum_i \sum_j (v_{ij} \log_a \frac{v_{ij}}{(WH)_{ij}} - v_{ij} + (WH)_{ij}) \\ V \in R_+^{m \times n}, \quad W \in R_+^{m \times r}, \quad H \in R_+^{r \times n}. \end{aligned} \tag{8}$$

where v_{ij} is in the ith row and jth column of V, $(WH)_{ij}$ is in the ith row and jth column of matrix WH, $r \ll \min(m, n)$, $a > 0$ and $a \neq 1$. W and H are defined as follows:

$$\begin{aligned} W = [w_1^T, \ldots, w_i^T, \ldots, w_m^T]^T, w_i \in R^r, i = 1, \ldots, m \\ H = [h_1, \ldots, h_j, \ldots, h_n], h_j \in R^r, j = 1, \ldots, n. \end{aligned} \tag{9}$$

Consider the corresponding objective function of v_{ij}, let w_i^T be a constant vector, h_j is a variable vector, so

$$\begin{aligned} f_{ij} &= v_{ij} \log_a \frac{v_{ij}}{w_i^T h_j} - v_{ij} + w_i^T h_j \\ &= v_{ij}(\log_a(v_{ij}) - \log_a(w_i^T h_j) - v_{ij} + w_i^T h_j). \end{aligned} \tag{10}$$

The partial derivative of h_j is

$$\frac{df_{ij}}{dh_j} = -\frac{v_{ij}}{\ln a} \frac{w_i}{w_i^T h_j} + w_i \tag{11}$$