Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 7410701, 10 pages
https://doi.org/10.1155/2019/7410701

Research Article

Low-Rank Deep Convolutional Neural Network for Multitask Learning

Fang Su,[1] Hai-Yang Shang,[2] and Jing-Yan Wang[3]

[1]*Shaanxi University of Science & Technology, Xi'an, Shaanxi Province 710021, China*
[2]*Northwest University of Political Science and Law, Xi'an, Shaanxi Province 710063, China*
[3]*New York University Abu Dhabi, Abu Dhabi, UAE*

Correspondence should be addressed to Hai-Yang Shang; haiyan.shang@outlook.com

Received 16 March 2019; Accepted 28 March 2019; Published 19 May 2019

Academic Editor: Ezequiel López-Rubio

In this paper, we propose a novel multitask learning method based on the deep convolutional network. The proposed deep network has four convolutional layers, three max-pooling layers, and two parallel fully connected layers. To adjust the deep network to multitask learning problem, we propose to learn a low-rank deep network so that the relation among different tasks can be explored. We proposed to minimize the number of independent parameter rows of one fully connected layer to explore the relations among different tasks, which is measured by the nuclear norm of the parameter of one fully connected layer, and seek a low-rank parameter matrix. Meanwhile, we also propose to regularize another fully connected layer by sparsity penalty so that the useful features learned by the lower layers can be selected. The learning problem is solved by an iterative algorithm based on gradient descent and back-propagation algorithms. The proposed algorithm is evaluated over benchmark datasets of multiple face attribute prediction, multitask natural language processing, and joint economics index predictions. The evaluation results show the advantage of the low-rank deep CNN model over multitask problems.

1. Introduction

1.1. Backgrounds. In machine learning applications, multitask learning has been a popular topic [1–9]. It tries to solve multiple related machine learning problems simultaneously. The motive is that, for many situations, multiple tasks are closely related, and the prediction results of different tasks should be consistent. Accordingly, borrowing the prediction of other tasks to help the prediction of a given task is natural. For example, in the face attribute prediction problem, given an image, the prediction of female gender and wearing long hair is usually related [10–14]. Moreover, in the problem of natural language processing, it is also natural to leverage the problems of part-of-speech (POS) tagging and noun chuck prediction, since a word with a POS of a noun usually appears in a noun chunk [15–19]. Multitask learning aims to build a joint model for multiple tasks from the same input data.

In recent years, deep learning has been proven to be the most powerful data representation method [20–32]. Deep learning methods learn a neural network of multiple layers to extract the hierarchical patterns from the original data and provide high-level and abstractive features for the learning problems. For example, for the face-recognition problems, a deep learning model learns simple patterns by the low-level layers, such as lines, edges, circles, and squares. In the median-level layers, parts of faces are learned, such as eyes, noses, mouths, etc. In the high-level layers, patterns of entire faces of different users are obtained. With the deep learning model, we can explore the hidden but effective patterns from the original data directly with multiple layers, even without domain knowledge and hand-coded features. This is a critical advantage compared to traditional shallow learning paradigms models.

Remark 1. If shallow learning paradigm is applied in this case, the model structure will not be sufficient to extract complex hierarchical features. The users of these shallow learning models have to code all these complex hierarchical

features manually in the feature extraction process, which is difficult and some times impossible.

Remark 2. If other nonneural networks learning models is used, such as spectral clustering, the hidden pattern of input data features cannot be directly explored. For example, spectral clustering treats each data point as a node in a graph and separates them by cutting the graph. However, it still needs a powerful data representation method to build the graph and cannot work itself well with a high-quality graph. Meanwhile, neural network models, especially deep neural network models, have the ability to represent the hidden patterns of input data points and build the high-quality graph accordingly. Thus, the nonneural network models and neural network models are complementary. Most recently, deep learning has been found very effective for multitask learning problems. For example, the following studies have discussed the usage of deep learning for multitask prediction.

(i) Zhang et al. [33] formulated a deep learning model constrained by multiple tasks, so that the early stopping can be applied to different tasks to obtain good learning convergence. Furthermore, different tasks regarding face images, including facial landmark detection, head pose estimation, and facial attribute detection are considered together by using a common deep convolutional neural network.

(ii) Liu et al. [34] proposed a deep neural network learning method for multitask learning problems, especially for learning representations across multiple tasks. The proposed method can combine cross-task data, and also regularize the neural network to make it generalized to new tasks. It can be used for both multiple domain classification problems and information retrieval problems.

(iii) Collobert and Weston [15] proposed a convolutional neural network for multitask learning problem in natural language processing applications. The targeted multiple tasks include POS tagging, noun chunk prediction, named entity recognition, etc. The proposed network is not only applied to multitask learning, but also applied to semisupervised learning, where only a part of the training set is labeled.

(iv) Seltzer and Droppo [35] proposed to learn a deep neural network for multiple tasks which shares the same data representations. This model is used to the applications of acoustic models with a primary task and one or more additional tasks. The tasks include phone labeling, phone context prediction, and state context prediction.

However, the relation among different tasks is not explored explicitly. Although the deep neural model can learn effective high-level abstractive features, without explicitly exploring the relation of different tasks, different groups of level features may be used to different tasks. Thus, the deep features are separated for different tasks, and the relationships among different tasks are ignored during the learning process of the deep network. To solve this problem, we propose a novel deep learning method by regularizing the parameters of the neural network regarding multiple tasks by low-rank.

1.2. Our Contributions. The proposed deep neural network is composed of four convolutional layers, three max-pooling layers, and two parallel fully connected layers. The convolutional layers are used to extract useful patterns from the local region of the input data, and the max-pooling layers are used to reduce the size of the intermediate outputs of convolutional layers while keeping the significant responses. The last two fully connected layers are used to map the outputs of convolutional and max-pooling layers to the labels of multiple tasks.

The rows of the transformation matrices of the full connection layers are corresponding to the mapping of different tasks. We assume that the tasks under consideration are closely related; thus, the rows of the transformation matrices are not completely independent to each other; thus, we seek such a transformation matrix with a minimum number of independent rows. We use the rank of the transformation matrix to measure the number of the independent rows and measure it by the nuclear norm. During the learning process, we propose to minimize the nuclear norm of one fully connected layer's transformation matrix. Meanwhile, we also assume that, for a group of related tasks, only all the high-level features generated by the convolutional layers and max-pooling layers are useful. Thus, it is necessary to select useful features. To this end, we propose to seek sparse rows for the second fully connected layer. The sparsity of the second transformation matrix is measured by its ℓ_1 norm, and we also minimize it in the learning process. Of course, we hope the predictions of the two fully connected layers could be low-rank and sparse simultaneously and also consistent with each other. Thus, we propose to minimize the squared ℓ_2 norm distance between the prediction vectors of the two fully connected layers. Meanwhile, we also reduce the prediction error and the complexity of the filters of the convolutional layers measured by the squared ℓ_2 norms. The objective function is the linear combination of these terms.

We developed an iterative algorithm to minimize the objective function. In each iteration, the transformation matrices and the filters are updated alternately. The transformation matrices are optimized by the gradient descent algorithm, and the filters are optimized by the back-propagation algorithms.

1.3. Paper Origination. The rest parts of this paper are organized as follows. In Section 2, we introduce the proposed method by modeling the problem as a minimization problem and develop an iterative algorithm to solve it. In Section 4, we conclude the paper.

2. Proposed Method

2.1. Problem Modeling. Suppose we have a set of n data points for the training process, denoted as $\{x_1, \ldots, x_n\}$,

where x_i is the i-th data point. x_i could be an image (presented as a matrix of pixels) or text (a sequence of embedding vectors of words). The problem of multitask learning is to predict the label vectors of m tasks. For x_i, the label vector is denoted as $\mathbf{y}_i = [\mathbf{y}_{i1}, \ldots, \mathbf{y}_{im}]^{\mathrm{T}} \in \{1, -1\}^m$, where $\mathbf{y}_{ij} = 1$ if x_i is a positive sample for the j-th task, and $\mathbf{y}_{ij} = -1$, otherwise.

To this end, we build a deep convolutional network to map the input data point to an output label vector. The network is composed of 4 convolutional layers, 3 max-pooling layers, and 2 parallel fully connected layers. The structure of the deep network is given in Figure 1. Please note that, for different types of input data, the convolutional and max-pooling layers are adjustable. For matrix inputs such as images, the layers perform 2D convolution and 2D max-pooling, while for sequences such as text, the layers conduct 1-D convolution and 1-D max-pooling.

We denote the intermediate output vector of the first 7 layers as $\phi(x) \in R^p$, where x is the input, and p is the number of pools of the last max-pooling layer. The set of filters in the convolutional layers of $\phi(x)$ are denoted as Φ. The outputs of the two parallel fully connected layers are denoted as

$$\begin{aligned} \mathbf{f}_1(x) &= W_1\phi(x) \in R^m, \\ \mathbf{f}_2(x) &= W_2\phi(x) \in R^m, \end{aligned} \tag{1}$$

where $W_1 \in R^{m\times p}$ and $W_2 \in R^{m\times}$ are the transformation matrix of the two layers. In the two fully connected layers map, the p-dimensional vector $\phi(x)$ of two vectors of m scores for m tasks. Each score measures the degree of the given data point belonging to the positive class. The two fully connected layers are corresponding to the low-rank and sparse prediction results of the network. By fusing their results, we can explore both the low-rank structure of the prediction scores of multitasks and also the sparse structure of the deep features learned from the network. In our model, the first fully connected layer $\mathbf{f}_1(x)$ is responsible for the low-rank structure, while the second fully connected layer $\mathbf{f}_1(x)$ is responsible for the sparse structure.

The final outputs of the network are the summation of the outputs of the two fully connected layers:

$$\mathbf{g}_1(x) = W_1\phi(x) + W_2\phi(x) \in R^m. \tag{2}$$

To learn the parameters of the deep network of $\mathbf{g}_1(x)$, we consider the following four problems:

(i) *Low-Rank Regularization*. As we discussed earlier, the tasks are not completely independent from each other, but they are closely related to each other. To explore the relationships between different tasks, we learn a deep and shared representation $\phi(x)$ for the input data x. Based on this shared representation, we also request the transformation matrix W_1 of one of the last fully connected layer to be of low-rank. The motive is that the m columns of W_1 actually map the representation $\phi(x)$ to the m scores of m tasks. The rank of W_1 measures the maximum number of linearly independent columns of W_1. Thus, by minimizing the rank of W_1, we can impose the mapping functions of different tasks to be dependent on each other and minimize the number of independent tasks. To measure the rank the matrix W_1, $\mathrm{rank}(W_1)$, we use the nuclear norm of W_1, denoted as $\|W_1\|_*$. $\|W_1\|_*$ is calculated as the summation of its singular values:

$$\|W_1\|_* = \sum_l \varrho_l, \tag{3}$$

where ϱ_l is its l-th singular value. We propose to learn W_1 by regularizing its rank as follows:

$$\min_{W_1} \|W_1\|_*. \tag{4}$$

(ii) *Sparse Regularization*. We further regularize the mapping transformation matrix of the second fully connected layer by sparsity. The motive of the sparsity is that the effective deep features for different tasks might be different, and for each task, not all the features are needed. Although we learn a group of deep features in $\phi(x)$ and share it with all the tasks, for a specific task and its relevant tasks, only a small number of deep features are necessary, and feature selection is a critical step. For the purpose of features selection, we impose the sparsity penalty to the transformation matrix of the second fully connected layer, W_2, since it maps the deep features to the prediction scores of m tasks. To measure the sparsity of W_2, we use the ℓ_1 norm of W_2, which is the summation of the absolute values of all the elements of the matrix:

$$\|W_2\|_1 = \sum_{jk} \left|[W_2]_{jk}\right|. \tag{5}$$

We minimize the ℓ_1 norm of W_2 to learn a sparse W_2

$$\min_{W_2} \frac{1}{2}\|W_2\|_1. \tag{6}$$

(iii) *Prediction Consistency*. The outputs of the two fully connected layers of low-rank and sparsity may give different results. However, they can be consistent with each other so that the prediction results can be low-rank and sparse simultaneously. To this end, we impose to minimize the squared ℓ_2 norm distance between the prediction results of the two layers over all the training data points:

$$\min_{\Phi, W_1, W_2} \frac{1}{2}\sum_{i=1}^{n} \|W_1\phi(x_i) - W_2\phi(x_i)\|_2^2. \tag{7}$$

(iv) *Prediction Error Minimization*. We also propose to learn an effective multitask predictor by minimizing the prediction error. To measure the prediction error of a data point, x, we calculate the squared ℓ_2 norm distance between its prediction result $g(x)$ and its true label vector $\mathbf{y}$:

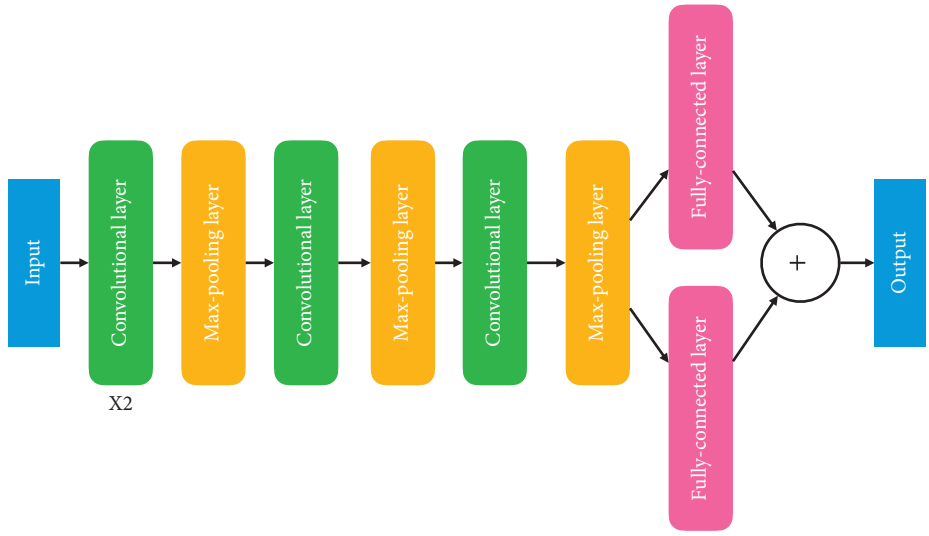

Figure 1: Sturcture of the proposed deep convolutional network.

$$\|\mathbf{y} - g(x)\|_2^2 = \|\mathbf{y} - (W_1\phi(x) + W_2\phi(x))\|_2^2. \tag{8}$$

We learn the parameters of the deep network by minimizing the errors over all the training data points:

$$\min_{\Phi,W_1,W_2} \frac{1}{2}\sum_{i=1}^{n} \|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\|_2^2. \tag{9}$$

(v) *Complexity Reduction*. Finally, we regularize the filters of the convolutional layers, Φ, by the squared ℓ_2 norms to prevent the network from being over complex:

$$\min_{\Phi} \frac{1}{2}\|\Phi\|_2^2. \tag{10}$$

The overall optimization problem is the weighted combination of the problems above:

$$\min_{\Phi,W_1,W_2} \left\{ g = \frac{1}{2}\|\Phi\|_2^2 + \frac{C_1}{2}\sum_{i=1}^{n} \|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\|_2^2 + C_2\|W_1\|_* + \frac{C_3}{2}\|W_2\|_1 + \frac{C_4}{2}\sum_{i=1}^{n} \|W_1\phi(x_i) - W_2\phi(x_i)\|_2^2 \right\}, \tag{11}$$

where C_1, C_2, C_3, and C_4 are the weights of different objective terms, and g is the overall objective function. By optimizing this problem, we can obtain a deep convolutional network with a low-rank and sparse deep features for the problem of multitask learning.

2.2. Optimization. To solve the problem in (12), we use the alternate optimization method. The parameters are updated iteratively in an iterative algorithm. When one parameter is updated, others are fixed. In the following sections, we will discuss how to solve them separately.

2.2.1. Updating W_1. When we update W_1, we fix W_2 and Φ, remove the terms irrelevant to W_1 from (12), and obtain the following optimization problem:

$$\min_{W_1} \left\{ g_1(W_1) = \frac{C_1}{2}\sum_{i=1}^{n} \|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\|_2^2 + C_2\|W_1\|_* + \frac{C_4}{2}\sum_{i=1}^{n} \|W_1\phi(x_i) - W_2\phi(x_i)\|_2^2 \right\}, \tag{12}$$

where g_1 is the objective function of this problem. To solve this problem, we use the gradient descent algorithm. W_1 is descended to the direction of the gradient of $g_1(W_1)$:

$$W_1 \longleftarrow W_1 - \varsigma\nabla g_1(W_1), \tag{13}$$

where $\nabla g_1(W_1)$ is the gradient function of $g_1(W_1)$, and ς is the descent step size. To calculate the gradient function $\nabla g_1(W_1)$, we first split the objective into two terms:

$$g_1(W_1) = g_{11}(W_1) + g_{12}(W_1), \tag{14}$$

where

$$g_{11}(W_1) = \frac{C_1}{2}\sum_{i=1}^{n} \|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\|_2^2 + \frac{C_4}{2}\sum_{i=1}^{n} \|W_1\phi(x_i) - W_2\phi(x_i)\|_2^2, \tag{15}$$

$$g_{12}(W_1) = C_2\|W_1\|_*.$$

The first term $g_{11}(W_1)$ is a quadratic term while $g_{12}(W_1)$ is a unclear term. Thus, the gradient function of $g_1(W_1)$ is the sum of the gradient functions of the two terms:

$$\nabla g_1(W_1) = \nabla g_{11}(W_1) + \nabla g_{12}(W_1), \tag{16}$$

where $\nabla g_{11}(W_1)$ can be easily obtained as

$$g_{11}(W_1) = -C_1 \sum_{i=1}^{n} (\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i)))\phi(x_i)^{\mathrm{T}} + C_4 \sum_{i=1}^{n} (W_1\phi(x_i) - W_2\phi(x_i))\phi(x_i)^{\mathrm{T}}. \tag{17}$$

To obtain the gradient function of $g_{12}(W_1) = C_2\|W_1\|_*$, we first decompose W_1 by singular value decomposition (SVD):

$$W_1 = U \sum V, \tag{18}$$

where U and V are the two orthogonal matrices, $\sum$ is a diagonal matrix containing all the singular values. According to the Proposition 1 of [36], the gradient of $\|W_1\|_* = U\sum^{-1}|\sum|V$; thus,

$$g_{12}(W_1) = C_2 U \sum\nolimits^{-1} \Big|\sum\Big| V. \tag{19}$$

2.2.2. Updating W_2. To update W_2, we also fix other parameters and remove the irrelevant terms:

$$\begin{aligned} g_2(W_2) &= \frac{C_1}{2} \sum_{i=1}^{n} \left\|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\right\|_2^2 \\ &\quad + \frac{C_3}{2}\|W_2\|_1 + \frac{C_4}{2} \sum_{i=1}^{n} \left\|W_1\phi(x_i) - W_2\phi(x_i)\right\|_2^2 \\ &= g_{21}(W_2) + g_{22}(W_2), \end{aligned} \tag{20}$$

where

$$\begin{aligned} g_{21}(W_2) &= \frac{C_1}{2} \sum_{i=1}^{n} \left\|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\right\|_2^2 \\ &\quad + \frac{C_4}{2} \sum_{i=1}^{n} \left\|W_1\phi(x_i) - W_2\phi(x_i)\right\|_2^2, \end{aligned} \tag{21}$$

is a quadratic term, and

$$g_{22}(W_2) = \frac{C_3}{2}\|W_2\|_1, \tag{22}$$

is a ℓ_1 norm term. We also use the gradient descent algorithm to update W_2:

$$W_2 \longleftarrow W_2 - \varsigma \nabla g_2(W_2), \tag{23}$$

where

$$\begin{aligned} \nabla g_2(W_2) &= \nabla g_{21}(W_2) + \nabla g_{21}(W_2), \\ \nabla g_{21}(W_2) &= -C_1 \sum_{i=1}^{n} (\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i)))\phi(x_i)^{\mathrm{T}} \\ &\quad - C_4 \sum_{i=1}^{n} (W_1\phi(x_i) - W_2\phi(x_i))\phi(x_i)^{\mathrm{T}}. \end{aligned} \tag{24}$$

To obtain the gradient function of $g_{21}(W_2)$, we rewrite W_2 and $\|W_2\|_1$ as follows:

$$W_2 = \begin{bmatrix} \mathbf{w}_{21} \\ \vdots \\ \mathbf{w}_{2m} \end{bmatrix}, \tag{25}$$

where

$$\begin{aligned} \left\|W_2\right\|_1 &= \sum_{i=1}^{m} \left\|\mathbf{w}_{2i}\right\|_1 = \sum_{i=1}^{m} \left\|\mathbf{w}_{2i}\right\|_1, \\ \left\|\mathbf{w}_{2i}\right\|_1 &= \sum_{j=1}^{d} \left|\mathbf{w}_{2ij}\right| = \sum_{j=1}^{d} \frac{\mathbf{w}_{2ij}^2}{\left|\mathbf{w}_{2ij}\right|} \\ &= \mathbf{w}_{2i}\ \mathrm{diag}\left(\left|\mathbf{w}_{2i1}\right|, \ldots, \left|\mathbf{w}_{2id}\right|\right)^{-1} \mathbf{w}_{2i}^{\mathrm{T}}, \end{aligned} \tag{26}$$

and $\mathbf{w}_{2i} = [\mathbf{w}_{2i1}, \ldots, \mathbf{w}_{2id}]$ is the i-th row of W_2. For the gradient function of $g_{22}(W_2)$ regarding W_2, we decompose the problem to the gradients of g_{22} with regard to different rows of W_2, since in the problem, the rows are independent to each other:

$$\nabla g_{22}(W_2) = \begin{bmatrix} \nabla g_{22}(\mathbf{w}_{21}) \\ \vdots \\ \nabla g_{22}(\mathbf{w}_{2m}) \end{bmatrix}, \tag{27}$$

where $\nabla g_{22}(\mathbf{w}_{2i})$ is the gradient of g_{22} regarding $\mathbf{w}_{2i}$, and according to (25) and (26), we have the subgradient of g_{22} as follows:

$$\nabla g_{22}(\mathbf{w}_{2i}) = C_3 \mathbf{w}_{2i}\ \mathrm{diag}\left(\left|\mathbf{w}_{2i1}\right|, \ldots, \left|\mathbf{w}_{2id}\right|\right)^{-1}. \tag{28}$$

2.2.3. Updating Φ. To optimize the filters of the deep network, we fix both W_1 and W_2 and use the back-propagation algorithm based on the chain rule. The corresponding problem is given as follows:

$$\begin{aligned} \min_{\Phi} \Bigg\{ g_3(\Phi) &= \frac{1}{2}\|\Phi\|_2^2 + \sum_{i=1}^{n} \Bigg(\frac{C_1}{2} \left\|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\right\|_2^2 \\ &\quad + \frac{C_4}{2} \left\|W_1\phi(x_i) - W_2\phi(x_i)\right\|_2^2 \Bigg) \Bigg\} = \frac{1}{2}\|\Phi\|_2^2 + \sum_{i=1}^{n} g_{3i}(\Phi), \end{aligned} \tag{29}$$

where

$$\begin{aligned} g_{3i}(\Phi) &= \frac{C_1}{2} \left\|\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i))\right\|_2^2 \\ &\quad + \frac{C_4}{2} \left\|W_1\phi(x_i) - W_2\phi(x_i)\right\|_2^2. \end{aligned} \tag{30}$$

is a data pointwise term. Back propagation is based on gradient descent algorithm:

$$\Phi \longleftarrow \Phi - \varsigma \nabla g_3(\Phi). \tag{31}$$

and according to the chain rule,

$$\nabla g_3(\Phi) = \Phi + \sum_{i=1}^{n} \nabla g_{3i}(\Phi), \tag{32}$$

where

$$\begin{aligned}\nabla g_{3i}(\Phi) &= \nabla g_{3i}(\phi(x_i))\nabla_\Phi \phi(x_i),\\ \nabla g_{3i}(\phi(x_i)) &= -C_1(W_1 + W_2)^{\mathrm{T}}\\ &\quad \cdot (\mathbf{y}_i - (W_1\phi(x_i) + W_2\phi(x_i)))\\ &\quad + C_4(W_1 - W_2)^{\mathrm{T}}(W_1\phi(x_i) - W_2\phi(x_i)).\end{aligned} \tag{33}$$

3. Experiments

In this section, we test the proposed method over several multitask learning problems and compare it to the state-of-the-art deep learning methods for the multitask learning problem.

3.1. Experiment Setting. We test the proposed method over the following benchmark datasets:

(i) *Large-scale CelebFaces Attributes (CelebA) Dataset*. The first dataset we used is a face image dataset, named CelebA Dataset [37]. This dataset has 2,02,599 images, and each image has 40 binary attributes, such as wearing eyeglasses, wearing hats, having a pointy nose, smiling, etc. The prediction of each attribute is treated as a task; thus, this is 40-task multitask learning problem. The input data is image pixels. The downloading URL for this dataset is at http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

(ii) *Annotated Corpus for Named Entity Recognition*. The second dataset we used is a dataset for named entity recognition. It contains 47,959 sentences, which contain 10,48,576 words. Each word is tagged by a named entity type, such as Geographical Entity, Organization, Person, etc., or a nonnamed entity. Moreover, each work is also tagged by a part-of-speech (POS) type, such as noun, pronoun, adjective, determiner, verb, adverb, etc. Meanwhile, we also have the labels of noun chunk. We have three tasks for each work, named entity recognition (NER), POS labeling, and noun clunking. For each word, we use a window of size 7 to extract the context, and the embedding vectors of the words in the window are used as the input. This dataset can be downloaded from https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus.

(iii) *Economics*. The third dataset we used is a dataset for tasks of property price trend and stock price trend prediction. The input data is the wave of historical data of property prices and stock prices and each data point is the data of three months of both prices of properties and stocks, and the label of each data point is the trend of stock price and property price. We collect the data of last 20 years of USA and China and generate a total number of 480 data points.

In the experiments, we split an entire dataset to a training set and a test set of the equal sizes. The training set is used to learn the parameters of the deep network, and then we use the test set to evaluate the performance of the proposed learning method. To measure the performance, we use the average accuracy for different tasks.

3.2. Experiment Results

3.2.1. Comparison of Prediction Accuracy of Different Methods. We compare the proposed method against several deep learning-based multitask methods, including the methods proposed by Zhang et al. [33], Liu et al. [34], Collobert and Weston [15], and Seltzer and Droppo [35]. The results are reported in Figure 2. According to the results, the proposed methods always achieve the best prediction performances, over three multitask learning tasks, especially in the NER and Economics. For the Economics benchmark dataset, our method is the only method which obtains an average prediction accuracy higher than 0.80, while the other methods only obtain accuracies lower than 0.75. This is not surprising since our method has the ability to explore the inner relation between different tasks by the low-rank regularization of the weights of the CNN model for different tasks. In the Economics benchmark dataset, the number of training examples is small; thus, it is even more necessary to borrow the data representation of different tasks. For the CelebFaces dataset, the improvement of the proposed method over the other methods is slight. Moreover, we also observe that the methods of Zhang et al. [33] and Liu et al. [34] outperforms the methods of Collobert and Weston [15] and Seltzer and Droppo [35] in most cases.

3.2.2. Comparison of Running Time of Different Methods. We also report the running time of the training processes of the compared methods in Figure 3. According to the results reported in the figure, the training process of Seltzer and Droppo's [35] method is the longest, and the most efficient method is Collobert and Weston's [15] algorithm. Our method's running time of the training process is longer than Zhang et al.'s [33] and Collobert and Weston's [15] methods, but still acceptable for the datasets of CelebFaces and NER. While for the training process over the Economics benchmark dataset, the running time is very short compared to the other two datasets, since its size is relatively small.

3.2.3. Influence of Tradeoff Parameters. In our method, there are four important tradeoff parameters, which control the weights of the terms of classification errors, the rank of the weight matrix, and the ℓ_1 norm sparsity of the weight matrix, and the consistency of predictions of the sparse model and low-rank model. The four tradeoff parameters are C_1, C_2, C_3, and C_4. We study the influences of the changes of their values to the prediction accuracy and report the results of

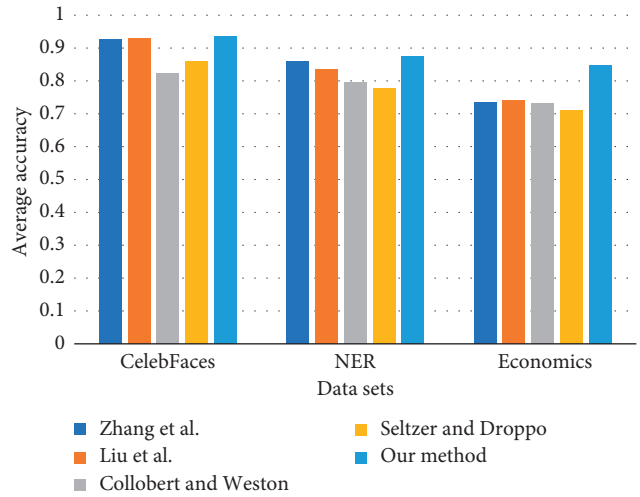

Figure 2: Prediction performance of compared methods over benchmark datasets.

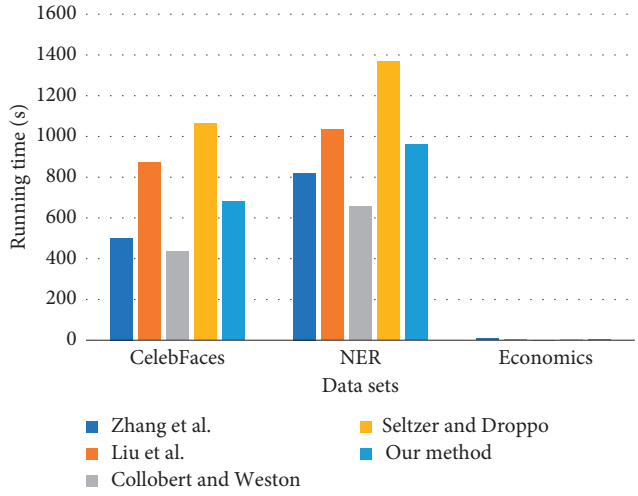

Figure 3: Running time of compared methods over benchmark datasets.

our method with varying values of these parameters in Figure 4. We have the following observations as follows:

(i) According to the results in Figure 4, when the values of C_1 increase from 0.01 to 100, the prediction accuracy keeps growing. This is due to the fact that this parameter is the weight of the classification error term, and when its value is increasing, the classification error over the training set plays a more and more important role in the learning process; thus, it boosts the classification performance accordingly. But when its value is larger than 100, the performance improvement is not significant anymore.

(ii) When the values of C_2 increases, the performance of the proposed keeps improving. This is due to the importance of the low-rank regularization of the proposed method. C_2 controls the weight of the low-rank regularization term, and it is the key to explore the relationships among different tasks of multitask problem. This is even more obvious for the Economics dataset, where the data size is small, and cross-task information plays a more important role.

(iii) The proposed algorithm seems stable to the changes in the values of C_3, which is the weight of the sparsity term of the objective. This term plays the role of feature selection over the convolutional representation of the input data. The stability over the changes of C_3 implies that the convolutional features extracted by our model already give good performances; thus, the feature selection does not significantly improve the performances.

(iv) For the parameter C_4, the average accuracy improves slightly when its value increases until it reaches 100; then the performances seem to decrease slightly. This suggests that the consistency between sparsity and low-rank somehow improves the performance, but it does not always help. For forcing the consistency with a large weight for the consistency term, the performance will not be improved.

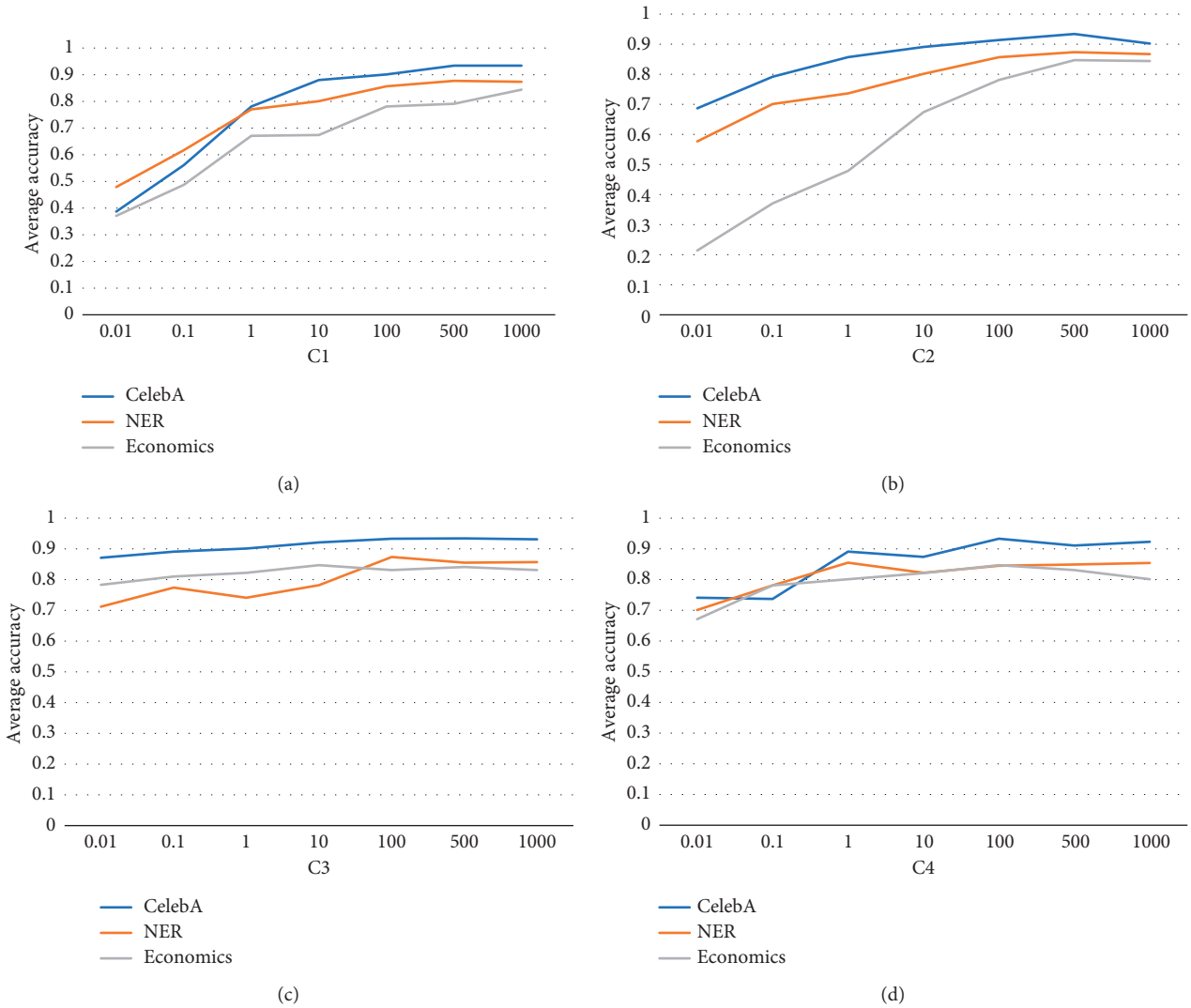

Figure 4: Influences of tradeoff parameters over benchmark dataset.

4. Conclusion

In this paper, we proposed a novel deep learning method for the multitask learning problem. The proposed deep network has convolutional, max-pooling, and fully connected layers. The parameters of the network are regularized by low-rank to explore the relationships among different tasks. Meanwhile, it also has the function of deep feature selection by imposing sparsity regularization. The learning of the parameters are modeled as a joint minimization problem and solved by an iterative algorithm. The experiments over the benchmark datasets show its advantage over the state-of-the-art deep learning-based multitask models.

Data Availability

The large-scale CelebFaces Attributes (CelebA) dataset used to support the findings of this study have been deposited in the CelebA repository at http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html. The Annotated Corpus for Named Entity Recognition data used to support the findings of this study have been deposited in the entity-annotated-corpus repository at https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the MOE Project of Humanities and Social Sciences of China under Grant No. 19YJAZH076, and the "Thousand People Plan" Specially Invited Expert for Young Professionals in Shaanxi Province, Shaanxi Soft Science Research Program under Grant No. 2018KRM065.

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems*, pp. 41–48, MIT Press, Cambridge, MA, USA, 2007.

[2] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[3] N. Doulamis and A. Voulodimos, "Fast-mdl: fast adaptive supervised training of multi-layered deep learning models for consistent object tracking and classification," in *Proceedings of the 2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 318–323, IEEE, Chania, Crete Island, Greece, October 2016.

[4] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117, ACM, Seattle, WA, USA, August 2004.

[5] L. Jacob, J. p. Vert, and F. R. Bach, "Clustered multi-task learning: a convex formulation," in *Advances in Neural Information Processing Systems*, pp. 745–752, MIT Press, Cambridge, MA, USA, 2009.

[6] W. Mao, X. Mu, Y. Zheng, and G. Yan, "Leave-one-out cross-validation-based model selection for multi-input multi-output support vector machine," *Neural Computing and Applications*, vol. 24, no. 2, pp. 441–451, 2014.

[7] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, https://arxiv.org/abs/1706.05098.

[8] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[9] D. Zhou, J. Wang, B. Jiang, and Y. Li, "Multiple-relations-constrained image classification with limited training samples via pareto optimization," *Neural Computing and Applications*, pp. 1–22, 2018.

[10] S.-Y. Cho and J.-J. Wong, "Human face recognition by adaptive processing of tree structures representation," *Neural Computing and Applications*, vol. 17, no. 3, pp. 201–215, 2008.

[11] E. Owusu, Y.-Z. Zhan, and Q.-R. Mao, "An svm–adaboost-based face detection system," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 26, no. 4, pp. 477–491, 2014.

[12] J.-J. Wong and S.-Y. Cho, "A face emotion tree structure representation with probabilistic recursive neural network modeling," *Neural Computing and Applications*, vol. 19, no. 1, pp. 33–54, 2010.

[13] S. Yao, Z. Chen, Y. Jia, and C. Liu, "Cascade heterogeneous face sketch-photo synthesis via dual-scale markov network," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 30, no. 2, pp. 217–233, 2018.

[14] Y. Zhong, J. Sullivan, and H. Li, "Face attribute prediction using off-the-shelf cnn features," in *Proceedings of the 2016 International Conference on Biometrics (ICB)*, pp. 1–7, IEEE, Seoul, Korea, August 2016.

[15] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, ACM, Helsinki, Finland, July 2008.

[16] S. Herath, T. Ikeda, S. Ishizaki, Y. Anzai, and H. Aiso, "Analysis system for sinhalese unit structure," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 4, no. 1, pp. 29–48, 1992.

[17] A. Jabbar, S. Iqbal, A. Akhunzada, and Q. Abbas, "An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 30, no. 5, pp. 1–21, 2018.

[18] C. Lyon and R. Frank, "Using single layer networks for discrete, sequential data: an example from natural language processing," *Neural Computing and Applications*, vol. 5, no. 4, pp. 196–214, 1997.

[19] R. Shams and R. E. Mercer, "Supervised classification of spam emails with natural language stylometry," *Neural Computing and Applications*, vol. 27, no. 8, pp. 2315–2331, 2016.

[20] Y. Geng, G. Zhang, W. Li et al., "A novel image tag completion method based on convolutional neural transformation," in *Proceedings of the International Conference on Artificial Neural Networks and Machine Learning–ICANN 2017*, pp. 539–546, Springer, Alghero, Italy, September 2017.

[21] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: a deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 513–520, Bellevue, WA, USA, June-July 2011.

[22] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[23] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 689–696, Bellevue, WA, USA, June-July 2011.

[26] L. Sadouk, T. Gadi, and E. Essoufi, "A novel deep learning approach for recognizing stereotypical motor movements within and across subjects on the autism spectrum disorder," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7186762, 16 pages, 2018.

[27] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[28] A. Voulodimos, N. Doulamis, G. Bebis, and T. Stathaki, "Recent developments in deep learning for engineering applications," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 8141259, 2 pages, 2018.

[29] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: a brief review," *Computational intelligence and neuroscience*, vol. 2018, Article ID 7068349, 13 pages, 2018.

[30] Y. Wu, H. Zhai, M. Li, F. Cui, L. Wang, and N. Patil, "Learning image convolutional representations and complete tags jointly," *Neural Computing and Applications*, pp. 1–12, 2017.

[31] G. Zhang, G. Liang, W. Li et al., "Learning convolutional ranking-score function by query preference regularization," in *Lecture Notes in Computer Science*, pp. 1–8, Springer, Berlin, Germany, 2017.

[32] G. Zhang, G. Liang, F. Su, F. Qu, and J.-Y. Wang, "Cross-domain attribute representation based on convolutional neural network," in *Intelligent Computing Methodologies*, pp. 134–142, Springer, Berlin, Germany, 2018.

[33] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proceedings of the European Conference on Computer Vision–ECCV 2014*, pp. 94–108, Springer, Amsterdam, The Netherlands, October 2014.

[34] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y. Y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," in *Proceedings of the Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies–HLT-NAACL, pp. 912–921, Denver, CO, USA, May-June 2015.

[35] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proceedings of the 2013 IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP)*, pp. 6965–6969, IEEE, Vancouver, Canada, May 2013.

[36] X. Zhen, M. Yu, X. He, and S. Li, "Multi-target regression via robust low-rank learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 497–504, 2017.

[37] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, Tampa, FL, USA, December 2015.