J Sci Comput (2018) 76:1396–1406
https://doi.org/10.1007/s10915-018-0666-1

Linear Feature Transform and Enhancement of Classification on Deep Neural Network

Penghang Yin[1] · **Jack Xin**[2] · **Yingyong Qi**[2]

Received: 6 February 2017 / Revised: 19 September 2017 / Accepted: 6 February 2018 /
Published online: 23 February 2018

Abstract A weighted and convex regularized nuclear norm model is introduced to construct a rank constrained linear transform on feature vectors of deep neural networks. The feature vectors of each class are modeled by a subspace, and the linear transform aims to enlarge the pairwise angles of the subspaces. The weight and convex regularization resolve the rank degeneracy of the linear transform. The model is computed by a difference of convex function algorithm whose descent and convergence properties are analyzed. Numerical experiments are carried out in convolutional neural networks on CAFFE platform for 10 class handwritten digit images (MNIST) and small object color images (CIFAR-10) in the public domain. The transformed feature vectors improve the accuracy of the network in the regime of low dimensional features subsequent to dimensional reduction via principal component analysis. The feature transform is independent of the network structure, and can be applied to reduce complexity of the final fully-connected layer without retraining the feature extraction layers of the network.

Keywords Linear feature transform · Difference of weighted nuclear norms · Low dimensional features · Enhanced classification

Mathematics Subject Classification 68W40 · 15A04 · 90C26 · 15A18

This work was partially supported by NSF Grants DMS-1222507, DMS-1522383, IIS-1632935.

✉ Penghang Yin
yph@ucla.edu

Jack Xin
jxin@math.uci.edu

Yingyong Qi
yqi@uci.edu

1 Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA

2 Department of Mathematics, University of California, Irvine, Irvine, CA 92697, USA

1 Introduction

Deep neural networks (DNN, [5,9,13]) are the state of the art methods in object classification tasks in computer vision [8] among other fields [20]. The basic form of DNN is convolutional neural networks (CNN) [2,3]. An open source platform to study CNN on handwritten digits (MNIST [10]) and image classification (CIFAR [7]) is CAFFE [6]. Typically, a large number of multi-scale features arise from DNN [5,7,9,13]. On the other hand, learning a rank-constrained transformation to group the features into clusters on the order of the number of classes has been shown recently to increase the performance of classifiers [11]. Each cluster or class is modeled as a subspace. The learned linear transformation aims to restore a low-rank structure for data from the same subspace, while enforcing a maximally separated structure for data from different subspaces.

In this paper, we study such a geometric linear feature transform (LFT) acting at the entrance of the last fully connected layer of DNN, see Fig. 1 for an illustration. A high dimensional feature vector appears (after the curved arrow in the middle of Fig. 1) and a large size weight matrix is necessary to map it to a small number of classes (10 in the data sets we study here). To overcome the complexity of the fully-connected layer and maintain the classification accuracy, we construct a linear transform to project the DNN feature vectors at the entrance of the fully connected layer to a low dimension, then train a small network with a single hidden layer to serve as a proxy for generating class probabilities. We shall work with the existing LeNet and cuda-convnet [2] on CAFFE for the MNIST and CIFAR-10 data sets respectively. It is well-known that there is a lot of redundancy in DNN features, hence performing standard dimensional reduction such as the principal component analysis (PCA) on the DNN features to certain threshold low dimension will nearly maintain the accuracy. Below the threshold, DNN performance will downgrade significantly. Our main finding is that performing rank constrained LFT helps to bring up the accuracy in the low dimensional feature regime. This can be done without retraining or modifying the network where the original features come from. Moreover, the LFT model and algorithm can be applied to most DNNs and be used to produce a low cost solution.

The major assumption of LFT is that the feature vectors (denoted by column vectors of matrix Y) approximately lie in a subspace and thus have low dimensional structure. Therefore, we aim to find a linear transform $T \in \mathbb{R}^{m\times n}$, such that dimension of the transformed features TY is greatly reduced (i.e., $m \ll n$), meanwhile the classification performance is maintained. The advantages of having low dimensional features include speed up of computation during

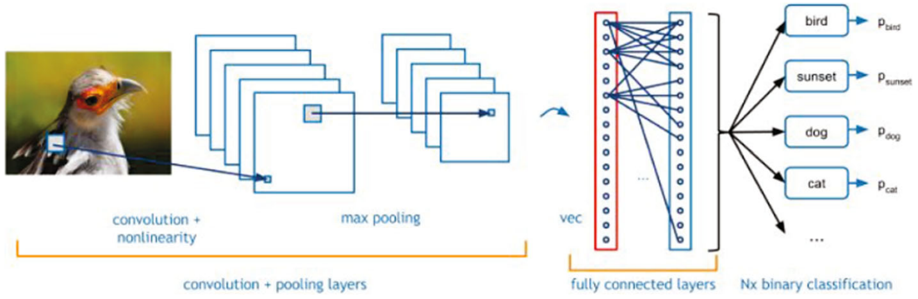

Fig. 1 An illustration of DNN for image classification. From left to right: multi-layers of feature extractions involve convolution and nonlinearities, the last layer is fully connected and leads to the final class probabilities

inference stage of network, as well as low memory and low energy consumption demand on mobile devices. Various simplifications exploiting linear concepts and structures have been studied to reduce costs in filtering and convolution layers of the network, such as rank-1 (separable) filter learning [14], low rank compression [4], sparse network and mimic learning (chapter 7, [20]). Mimic learning [1] refers to teaching a small or shallow NN (SNN) with a large and high performance DNN. The SNN is like a student, and is trained on synthetic labels to mimic the functionality of the teacher (large DNN). The synthetic lables are obtained by passing unlabeled data through the large and accurate DNN and collecting the scores. Experiments on CIFAR-10 image dataset [1] showed that a mimic SNN with a single convolution layer for feature extraction recovers the performance of the teacher DNN. With the speedup at inference by a combination of these strategies (especially mimic learning), our low dimensional feature transform can further save memory and computation at the fully connected layer.

The paper is organized as follows. In Sect. 2, we revisit the LFT model of [11] and observe the rank deficiency phenomenon. The norm constraint of the model [11] prevents the iterations from approaching zero but may not exclude rank degeneracy of the transform. We also note that the LFT algorithm of [11] is not descending in general. Our main contribution here is to fix the rank deficiency and non-descending issues in [11] by proposing a weighted difference of convex function (WDC) model augmented with a convex regularization. In Sect. 3, we present the associated DC algorithm (DCA) and show that it is descending and subsequentially convergent. In Sect. 4, numerical experiments show that our proposed algorithm indeed computes LFT to enhance the accuracy on CIFAR-10 and MNIST data when feature dimension is reduced by a factor of 32 while the accuracy is nearly maintained. The more the dimension is reduced, the higher the LFT enhancement. Concluding remarks are in Sect. 5.

Notations Throughout the paper, for any matrix $X \in \mathbb{R}^{m\times n}$ of rank r, we refer to the singular value decomposition (SVD) of X by the form $U\Sigma V^{\mathrm{T}}$, where $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal. $\|X\|_F := \sqrt{\sum_{i,j} X_{ij}^2}$ denotes the Frobenius norm of X. Let $\sigma_i(X)$ be the i-th largest singular value of X. $\|X\| := \sigma_1(X)$ denotes the spectral norm of X, whereas $\|X\|_* := \sum_{i=1}^r \sigma_i(X)$ denotes the nuclear norm of X. The subdifferential of $\|X\|_*$ is given by [17]

$$\partial\|X\|_* = \left\{UV^{\mathrm{T}} + W :\ U^{\mathrm{T}}W = 0,\ WV = 0,\ \|W\| \le 1\right\}.$$

2 LFT and Nuclear Norm Models

In this section, we review the LFT nuclear norm model of [11], point out the rank defects and propose our weighted-regularized model for DNN experiments in Sect. 4.

In [11], the authors propose to learn a global linear transformation on subspaces that preserves the low-rank structure for data within the same subspace, and, meanwhile introduces a maximally separated structure for data from different subspaces. More precisely, for the task of classification, they propose to solve the following minimization problem for the transformation matrix $\hat{T}$:

$$\hat{T} = \arg\min_T \sum_{i=1}^{c} \|TY_i\|_* - \|TY\|_* \quad \text{s.t.} \quad \|T\| = 1, \tag{2.1}$$

where c is the total number of classes, Y_i is the matrix of training data for the i-th class, Y is the concatenation of all Y_i's containing the whole training data. The norm constraint $\|T\| = 1$ simply prevents the trivial solution $\hat{T} = 0$. The nuclear norm serves as a convex relaxation of rank functional. Beyond that, it is shown in [11] that the objective function in (2.1) satisfies

$$\sum_{i=1}^{c} \|TY_i\|_* - \|TY\|_* \geq 0, \tag{2.2}$$

with equality when all transformed data from different classes are orthogonal to each other, i.e., $(TY_i)^{\mathrm{T}} TY_j = 0, \ \forall \ i \neq j$. When feature vectors of each class belong to a proper subspace of $\mathbb{R}^n$, the transform $\hat{T}$ tends to align feature vectors in each subspace while enlarge angles between subspaces, thus intuitively promoting accuracy of classification.

On the computational side, since the objective is a difference of two convex functions, the non-convex minimization problem (2.1) can be solved by the so-called difference of convex function algorithm (DCA) [16,18,19] via the iteration:

$$T^{k+1} = \arg\min_{T} \sum_{i=1}^{c} \|TY_i\|_* - \left\langle S^k, TY \right\rangle \quad \text{s.t.} \quad \|T\| = 1. \tag{2.3}$$

where $S^k \in \partial \|T^k Y\|_*$ is a subgradient of $\|\cdot\|_*$ at $T^k Y$. Note that although the objective function is convex, (2.3) is still a non-convex program because of the constraint.

It is easy to see that a necessary condition for the transformed feature subspaces to be pairwise orthogonal is

$$\sum_{i=1}^{c} d_i \leq n, \tag{2.4}$$

where d_i is the dimension of the subspace of the i-th class. However, this condition is somewhat too restrictive and often violated in real-world examples such as CIFAR-10 in our experiments. When subspace dimensions are relatively large, the pairwise orthogonality between transformed subspaces is clearly unachievable. In this case, we observed numerically that $\hat{T}$ tends to be rank deficient, in particular rank-1 which aligns all the feature vectors along a line. The norm constraint $\|T\| = 1$ in (2.1) does not prevent such a rank-1 defect solution from occurring. Moreover, since the subproblem (2.3) of DCA is non-convex due to the norm constraint which is implemented by normalization in [11], the iteration sequences from (2.3) can be non-descending.

To fix the issues aforementioned, we introduce a weight $w > 1$ to the second term $\|TY\|_*$ to enforce enlargement of angles between subspaces. We also replace the constraint $\|T\| = 1$ with a convex penalty term. Our new model is the following unconstrained minimization problem:

$$\min_{T} \Psi(T) := \sum_{i=1}^{c} \|TY_i\|_* - w\|TY\|_* + \frac{\lambda}{2}\|T - P\|_F^2. \tag{2.5}$$

In this new model, we search for $\hat{T}$ in the neighborhood of a candidate P whose size is controlled by the parameter $\lambda > 0$. For $m = n$, we may simply take P as the identity matrix $I_n \in \mathbb{R}^{n\times n}$. If $m < n$, we choose P via principal component analysis (PCA). Let the SVD of Y be $Y = U\Sigma V^{\mathrm{T}}$, then $P = U_{\cdot,1:m}^{\mathrm{T}} \in \mathbb{R}^{m\times n}$ with $U_{\cdot,1:m}$ consisting of left singular vectors of Y associated with the m largest singular values. We choose PCA as a simple unsupervised dimension reduction method for initiating P because there is no class information at the testing (inference) time and PCA strikes a good balance between efficiency and computational costs. In our numerical experiments, the PCA initialization did much better than random subspace projection or selecting a fixed subspace such as projecting onto the first few coordinates.

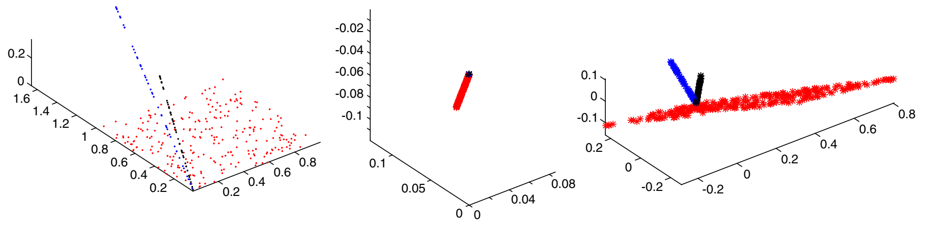

Fig. 2 Left: data points of the three classes. The angles between classes are 4.70°, 10.52°, 10.14°, respectively. Middle: degenerate solutions by solving (2.3). The transformed data points end up lying on a line. Right: separation results by solving (2.5). The corresponding angles are enlarged to 38.48°, 22.01°, 18.73°, respectively

The (strongly) convex penalty is much better than using unit-norm constraint in preserving the rank. To demonstrate improvement of the proposed (2.5) over (2.3), we provide a synthetic example in $\mathbb{R}^3$ of three classes (subspaces) of dimensions 2, 1, 1, respectively, as shown in Fig. 2 (left plot). In this case, mutual orthogonality is impossible. Solving (2.3) gives a rank-1 deficient $\hat{T}$ (middle plot), whereas solving (2.5) with $P = I_3$, outputs a full rank $\hat{T}$ helping enforce the separation between classes (right plot).

3 Algorithms

In this section, we present difference of convex functions algorithm and its convergence property for our model. Let us consider a general objective function $\Phi(X) = \Phi_1(X) - \Phi_2(X)$, where Φ_1 and Φ_2 are convex functions. DCA deals with the minimization of $\Phi(X)$ and takes the following form

$$\begin{cases} W^k \in \partial\Phi_2\left(X^k\right) \\ X^{k+1} = \arg\min_X \Phi_1(X) - \left(\Phi_2(X^k) + \left\langle W^k, X - X^k\right\rangle\right) \end{cases}$$

By the definition of subgradient, we have

$$\Phi_2\left(X^{k+1}\right) \geq \Phi_2\left(X^k\right) + \left\langle W^k, X^{k+1} - X^k\right\rangle.$$

As a result,

$$\begin{aligned} \Phi\left(X^k\right) &= \Phi_1\left(X^k\right) - \Phi_2\left(X^k\right) \geq \Phi_1\left(X^{k+1}\right) - \left(\Phi_2(X^k) + \left\langle W^k, X^{k+1} - X^k\right\rangle\right) \\ &\geq \Phi_1\left(X^{k+1}\right) - \Phi_2\left(X^{k+1}\right) = \Phi\left(X^{k+1}\right), \end{aligned}$$

We used the fact that X^{k+1} minimizes $\Phi_1(X) - (\Phi_2(X^k) + \langle W^k, X - X^k\rangle)$ in the first inequality above. Therefore, DCA permits a decreasing sequence $\{\Phi(X^k)\}$, leading to its convergence provided $\Phi(X)$ is bounded from below.

The DCA for solving (2.5) is:

$$T^{k+1} = \arg\min_T \sum_{i=1}^{c} \|TY_i\|_* - w\left\langle S^k, TY\right\rangle + \frac{\lambda}{2}\|T - P\|_F^2 \tag{3.1}$$

with $S^k \in \partial\|T^kY\|_*$. Suppose the SVD of T^kY is $U^k\Sigma^k {V^k}^{\mathrm{T}}$, then we can choose $S^k = U^k {V^k}^{\mathrm{T}}$.

Now that the subproblem (3.1) is a convex program, the DCA for (2.5) is always descending provided that (3.1) is solved properly, which is a nice mathematical property to have.

3.1 Convergence

Due to weighting in the nuclear norm model, the convex penalty term is needed to guarantee the lower bound of objective function. Next we show that the objective in (2.5) has a lower bound under mild conditions, that $\{\Psi(T^k)\}$ converges and T^k is uniformly bounded in k.

Proposition 3.1 *For any positive parameters* $(\lambda, w) > 0$, $\Psi(T) = \sum_{i=1}^c \|TY_i\|_* - w\|TY\|_* + \frac{\lambda}{2}\|T - P\|_F^2$ *is bounded from below by zero if* $w \le 1$, *and by* $-\frac{\lambda}{2}\|P\|_F^2 - \frac{(w-1)^2}{\lambda}\|Y\|_F^2$ *if* $w > 1$.

Proof Since by (2.2), $\sum_{i=1}^c \|TY_i\|_* - \|TY\|_* \ge 0$, it suffices to show that $\frac{\lambda}{2}\|T - P\|_F^2 - (w-1)\|TY\|_*$ has lower bound. The low bound is clearly zero if $w \in (0, 1]$. Now consider $w > 1$. By an alternative definition of nuclear norm [12], for any constant $\rho > 0$,

$$\|X\|_* := \inf_{Q,R}\left\{\frac{1}{2}\left(\rho\|Q\|_F^2 + \rho^{-1}\|R\|_F^2\right) : X = QR^{\mathrm{T}}\right\}, \tag{3.2}$$

therefore,

$$\|TY\|_* \le \frac{1}{2}\left(\rho\|T\|_F^2 + \rho^{-1}\|Y^{\mathrm{T}}\|_F^2\right). \tag{3.3}$$

Choosing $\rho = \frac{\lambda}{2(w-1)}$, we have

$$\frac{\lambda}{2}\|T - P\|_F^2 - (w-1)\|TY\|_* \ge \frac{\lambda}{4}\|T\|_F^2 - \lambda\langle T, P\rangle + \frac{\lambda}{2}\|P\|_F^2 - \frac{(w-1)^2}{\lambda}\|Y\|_F^2 \tag{3.4}$$

$$= \frac{\lambda}{4}\left(\|T - 2P\|_F^2 - 4\|P\|_F^2\right) + \frac{\lambda}{2}\|P\|_F^2 - \frac{(w-1)^2}{\lambda}\|Y\|_F^2 \tag{3.5}$$

$$\ge -\frac{\lambda}{2}\|P\|_F^2 - \frac{(w-1)^2}{\lambda}\|Y\|_F^2 \tag{3.6}$$

□

In the following, we show the descending property of $\Psi(T^k)$ and that $\|T^k - T^{k+1}\|_F \to 0$ as $k \to \infty$.

Proposition 3.2 *Let* $\{T^k\}$ *be the sequence of iterates generated by* (3.1) *with* $w \ge 1$. *Then* $\{\Psi(T^k)\}$ *is descending and convergent, and* $\|T^k - T^{k+1}\|_F \to 0$ *as* $k \to \infty$.

Proof

$$\Psi\left(T^k\right) - \Psi\left(T^{k+1}\right) = \frac{\lambda}{2}\|T^k - T^{k+1}\|_F^2 + \lambda\left\langle T^k - T^{k+1}, T^{k+1} - P\right\rangle + w\left(\|T^{k+1}Y\|_* - \|T^kY\|_*\right) + \sum_{i=1}^c\left(\|T^kY_i\|_* - \|T^{k+1}Y_i\|_*\right) \tag{3.7}$$

By the the first-order optimality condition for (3.1), we have that there exist $L_i^{k+1} \in \partial \|T^{k+1}Y_i\|_*$ for $1 \le i \le c$, such that

$$\sum_{i=1}^{c} L_i^{k+1} Y_i^{\mathrm{T}} - w S^k Y^{\mathrm{T}} + \lambda \left(T^{k+1} - P\right) = 0,$$

and therefore,

$$\begin{aligned}\lambda \left\langle T^k - T^{k+1}, T^{k+1} - P \right\rangle &= -\sum_{i=1}^{c} \langle L_i^{k+1}, \left(T^k - T^{k+1}\right) Y_i \rangle + w \left\langle S^k, \left(T^k - T^{k+1}\right) Y \right\rangle \\ &= \sum_{i=1}^{c} \left(\|T^{k+1}Y_i\|_* - \left\langle L_i^{k+1}, T^k Y_i \right\rangle \right) \\ &\quad + w \left(\|T^k Y\|_* - \left\langle S^k, T^{k+1} Y \right\rangle \right).\end{aligned}$$

Upon substitution into (3.7), we have

$$\begin{aligned}\Psi\left(T^k\right) - \Psi\left(T^{k+1}\right) &= \frac{\lambda}{2} \|T^k - T^{k+1}\|_F^2 + w \left(\|T^{k+1}Y\|_* - \left\langle S^k, T^{k+1} Y \right\rangle \right) \\ &\quad + \sum_{i=1}^{c} \left(\|T^k Y_i\|_* - \left\langle L_i^{k+1}, T^k Y_i \right\rangle \right) \\ &\ge \frac{\lambda}{2} \|T^k - T^{k+1}\|_F^2,\end{aligned}$$

implying the descent and convergence properties of $\Psi(T^k)$. In the above arguments, we have used the facts that

$$\left\langle L_i^{k+1}, T Y_i \right\rangle \le \|T Y_i\|_*, \text{ for all } T \in \mathbb{R}^{m \times n} \text{ and } 1 \le i \le c$$

with equality at $T = T^{k+1}$, and that

$$\left\langle S^k, T Y \right\rangle \le \|T Y\|_*, \text{ for all } T \in \mathbb{R}^{m \times n}$$

with equality at $T = T^k$.

Finally, since $\{\Psi(T^k)\}$ converges, we must have $\|T^k - T^{k+1}\|_F \to 0$ as $k \to \infty$. □

Remark 3.1 Our proof above is self-contained and more direct than the general theory [15,16] which treated the vector case.

Corollary 3.1 *There exists a positive constant $C = C(\lambda, w)$ such that $\|T^k\|_F \le C$, for all $k \ge 1$, hence the sequence $\{T^k\}$ is sub-sequentially convergent.*

Proof The descent property of $\Psi(T^k)$ and (3.4)–(3.5) give:

$$\begin{aligned}\Psi\left(P = T^0\right) &\ge \Psi\left(T^k\right) \\ &\ge \frac{\lambda}{2}\|T^k - P\|_F^2 - (w-1)\|T^k Y\|_* \\ &\ge \frac{\lambda}{4}\|T^k - 2\,P\|_F^2 - \frac{\lambda}{2}\|P\|_F^2 - \frac{(w-1)^2}{2}\|Y\|_F^2\end{aligned}$$

from which the corollary follows. □

3.2 Solving the Subproblem

Each DCA step for T^{k+1} can be updated via the alternating direction method of multipliers (ADMM). By introducing the auxiliary variable Z and multiplier Λ, we first recast (3.1) as

$$\min_T \sum_{i=1}^{c} \|Z_i\|_* - w\left\langle S^k, TY\right\rangle + \frac{\lambda}{2}\|T-P\|_F^2 \quad \text{s.t.} \quad Z - TY = 0,$$

and then form the augmented Lagrangian:

$$\begin{aligned}&\sum_{i=1}^{c} \|Z_i\|_* - w\left\langle S^k, TY\right\rangle + \frac{\lambda}{2}\|T-P\|_F^2 + \langle \Lambda, Z - TY\rangle + \frac{\delta}{2}\|Z-TY\|_F^2 \\ &= \sum_{i=1}^{c} \|Z_i\|_* - w\left\langle S^k, TY\right\rangle + \frac{\lambda}{2}\|T-P\|_F^2 + \sum_{i=1}^{c}\langle \Lambda_i, Z_i - TY_i\rangle + \sum_{i=1}^{c}\frac{\delta}{2}\|Z_i - TY_i\|_F^2,\end{aligned}$$

where $Z = [Z_1, \ldots, Z_c]$ and $\Lambda = [\Lambda_1, \ldots, \Lambda_c]$ are partitioned in the same way as Y is. By ignoring constants, ADMM takes the iteration:

$$\begin{aligned} T^{l+1} &= \arg\min_T -w\left\langle S^k, TY\right\rangle + \frac{\lambda}{2}\|T-P\|_F^2 + \left\langle \Lambda^l, Z^l - TY\right\rangle + \frac{\delta}{2}\|Z^l - TY\|_F^2 \\ Z_i^{l+1} &= \arg\min_{Z_i} \|Z_i\|_* + \left\langle \Lambda_i^l, Z_i - T^{l+1}Y_i\right\rangle + \frac{\delta}{2}\|Z_i - T^{l+1}Y_i\|_F^2, \ i = 1, \ldots, c \\ \Lambda_i^{l+1} &= \Lambda_i^l + \delta\left(Z_i^{l+1} - T^{l+1}Y_i\right), \ i = 1, \ldots, c \end{aligned}$$

The ADMM steps for updating T^{l+1} and Z_i^{l+1} have closed form solutions. Hereby we summarize the algorithm for solving (3.1) in Algorithm 1. In Algorithm 1,

$$\mathcal{S}_r(X) := \sum_{i=1}^{n} \mathbf{1}_{\{\sigma_i(X)>r\}}\left(\sigma_i(X) - r\right) u_i v_i^{\mathrm{T}}$$

denotes the soft-thresholding operator on singular values of X, where $\mathbf{1}_{\{\sigma_i(X)>r\}}$ is the indicator function given by

$$\mathbf{1}_{\{\sigma_i(X)>r\}} := \begin{cases} 1, & \sigma_i(X) > r \\ 0, & \text{otherwise} \end{cases}$$

Algorithm 1 ADMM for updating T^{k+1} in (3.1)

```
Input: $T^k,\ Y = [Y_1, \ldots, Y_c],\ S^k,\ P,\ \delta > 0$.
Initialize: $\{Z_i^0\}_{i=1}^c,\ \{\Lambda_i^0\}_{i=1}^c$.
  for $l = 0, 1, \ldots$ do
    $Z^l = [Z_1^l, \ldots, Z_c^l]$
    $\Lambda^l = [\Lambda_1^l, \ldots, \Lambda_c^l]$
    $T^{l+1} = (\Lambda^l Y^{\mathrm{T}} + \lambda P + wS^k Y^{\mathrm{T}} + \delta Z^l Y^{\mathrm{T}})(\delta Y Y^{\mathrm{T}} + \lambda I_n)^{-1}$
    $Z_i^{l+1} = \mathcal{S}_{1/\delta}(T^{l+1}Y_i - \Lambda_i^l/\delta),\ i = 1, \ldots, c$
    $\Lambda_i^{l+1} = \Lambda_i^l + \delta(Z_i^{l+1} - T^{l+1}Y_i),\ i = 1, \ldots, c$
  end for
Output: $T^{k+1}$.
```

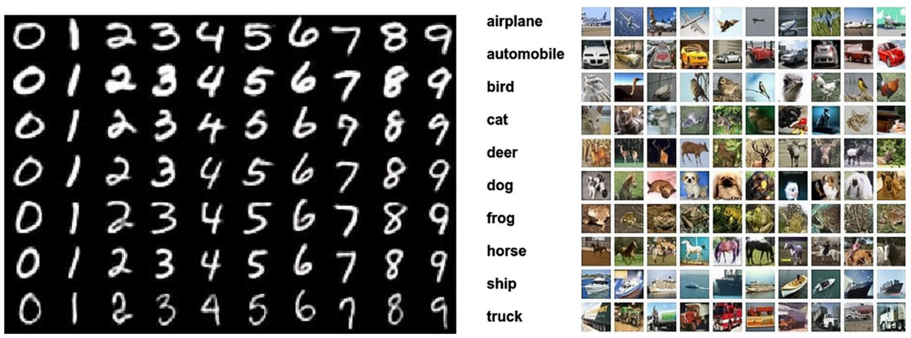

Fig. 3 Left: sample images of handwritten digits in MNIST. Right: 10 random example images from each class in CIFAR-10

4 Numerical Experiments

We present numerical experiments on the benchmark image datasets MNIST [10] and CIFAR-10 [7], using neural network classifiers. The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST database contains 70,000 28×28 images, including 60,000 training images and 10,000 testing images. The CIFAR-10 dataset consists of 60,000 color images of size 32×32. Each image is labeled with one of 10 classes (for example, airplane, automobile, bird, etc). These 60,000 images are partitioned into a training set of 50,000 images and a test set of 10,000 images; see Fig. 3 for sample images from the datasets.

We extract both training and testing features through trained convolutional neural nets (CNN) on Caffe [6]. Caffe is a deep learning framework developed by the Berkeley Vision and Learning Center and by community contributors. LeNet [10] and cuda-convnet [2] are two baseline CNNs (BCNN) on Caffe, working with MNIST and CIFAR-10 datasets respectively. The extracted features of CIFAR-10 images through cuda-convnet are 3-D arrays of dimensions $64\times4\times4$, while that of MNIST through LeNet are vectors in $\mathbb{R}^{500}$. We convert CIFAR-10 features into vectors in $\mathbb{R}^{1024}$. After $\hat{T} \in \mathbb{R}^{m\times n}$ ($n = 1024$ for CIFAR-10 and $n = 500$ for MNIST) is computed from the training feature vectors only, we then apply it to both the training and testing data, and feed the transformed data to a single layer neural net classifier (i.e., multi-class perceptron) from Scikit-learn package implemented in Python. Comparison of BCNN + PCA and BCNN + PCA + LFT is shown in Tables 1 and 2. The PCA provides $m \ll n$ orthogonal left singular vectors corresponding to the top m singular values of the feature matrix Y. These $m \ll n$ singular vectors span the reduced feature vector space.

For CIFAR-10, when feature dimensions are reduced to $m = 64$ and $m = 32$ from $n = 1024$, the accuracy dropped noticeably. The LFT can further improve the accuracy on top of PCA (or another dimensional reduction method). The P in model (2.5) is provided by PCA, with parameters $w = 3$ and $\lambda = 200$. The λ value is chosen so that the iterations deviate enough from the initial value while being non-degenerate. The w value is selected to enlarge angles among subspaces. Though there is room to vary these values, we have not observed much difference in the resulting classification accuracies reported here. These two parameter values are the same in all our numerical experiments. The stopping condition for the LFT iterations is that $\|T^{k+1} - T^k\|_F/\|T^k\|_F \leq 10^{-5}$. The additional gain from LFT is 2% at feature dimension $m = 32$, and 0.7% at dimension $m = 64$. For MNIST, when reduced

Table 1 Accuracy for CIFAR-10 with reduced feature dimensions ($m = 256, 128, 64, 32$) from $n = 1024$ in BCNN

Reduced dim	BCNN + PCA (%)	BCNN + PCA + LFT (%)
256	81.75	81.68
128	81.12	81.44
64	80.21	80.90
32	77.91	79.95

Table 2 Accuracy for MNIST with reduced feature dimensions ($m = 64, 32, 16, 8$) from $n = 500$ in BCNN

Reduced dim	BCNN + PCA (%)	BCNN + PCA + LFT (%)
64	98.98	99.06
16	98.92	98.95
32	98.14	98.33
8	95.31	97.10

Table 3 Accuracy for CIFAR-10 and MNIST using CAFFE when $m = n$

Dataset	Baseline-CNN (BCNN) (%)	BCNN + LFT (%)
CIFAR10	81.77	81.97
MNIST	99.05	99.12

feature dimensions are $m = 8$ and $m = 16$ from $n = 500$, LFT improves the accuracy by 1.8% and 0.2% respectively. We see in both cases that the lower the reduced dimension, the greater the improvement from LFT. If feature dimension remains at the original level (or $m = n$, $P = I_n$), the improvement by LFT is less than when $m \ll n$ as seen in Table 3. We remark that at $w = 1$, or using the method of [11], we have consistently encountered degeneracies in LFT iterations on either CIFAR-10 or MNIST data. This may indicate that the dimensionality condition (2.4) fails for features from deep neural networks, in other words $\sum_{i=1}^{c} d_i > n$. Nonetheless, the weighted model (2.5) offers a non-degenerate solution, especially when $m \ll n$.

5 Concluding Remarks

From the classification experiments on MNIST and CIFAR-10, we found that LFT can improve CNN classifiers based on the new linear feature transform model (2.5) especially in the regime of low feature dimensions. This is attactive for reducing the complexity of the last fully-connected layer where a large weight matrix multiplication is involved. With the help of LFT, a much smaller classifier can be used as a proxy to output class probabilities. Since LFT is independent of the CNN architecture, it is applicable to a state-of-the-art classifier without retraining.

Acknowledgements The authors wish to thank Dr. Xiangxin Zhu of Google Inc., Dr. Yang Yang, and Dr. Xin Zhong for helpful conversations on DNN and CAFFE.

References

1. Ba, L.J., Caruana, R.: Do Deep Nets Really Need to be Deep? arxiv:1312.6184 (2013)
2. cuda-convnet. https://code.google.com/p/cuda-convnet
3. Deng, L., Yu, D.: Deep Learning: Methods and Applications. NOW Publishers, Breda (2014)
4. Denton, E., Zaremba, W., Bruna, J., LeCun, Y., Fergus, R.: Exploiting linear structure within convolutional networks for efficient evaluation. In: Advances in Neural Information Processing Systems (NIPS), pp. 1269–1277 (2014)
5. Hinton, G.: Learning multiple layers of representation. Trends Cognit. Sci **11**(10), 428–434 (2007)
6. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093, (2014)
7. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. www.cs.toronto.edu/~kriz/index.htm (2009)
8. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
9. LeCun, Y., Bottou, L., Orr, G., Müller, K.: Neural Networks: Tricks of the Trade. Springer, Berlin (1998)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
11. Qiu, Q., Sapiro, G.: Learning transformations for clustering and classification. J. Mach. Learn. Res. **16**, 187–225 (2015)
12. Recht, B., Ré, C.: Parallel stochastic gradient algorithms for large-scale matrix completion. Math. Program. Comput. **5**(2), 201–226 (2013)
13. Schmidhuber, J.: Deep Learning in Neural Networks: An Overview. arXiv:1404.7828v4 (2014)
14. Sironi, A., Tekin, B., Rigamonti, R., Lepetit, V., Fua, P.: Learning separable filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(1), 94–106 (2015)
15. Tao, P.D., An, L.T.H.: Convex analysis approach to d.c. programming: theory, algorithms and applications. Acta Math. Vietnam. **22**, 289–355 (1997)
16. Tao, P.D., An, L.T.H.: A DC optimization algorithm for solving the trust-region subproblem. SIAM J. Optim. **8**(2), 476–505 (1998)
17. Watson, G.A.: Characterization of the subdifferential of some matrix norms. Linear Algebra Appl. **170**, 33–45 (1992)
18. Yin, P., Xin, J.: PhaseLiftOff: an accurate and stable phase retrieval method based on difference of trace and Frobenius norms. Commun. Mathe. Sci. **13**(2), 1033–1049 (2015)
19. Yin, P., Xin, J.: Iterative ℓ_1 minimization for non-convex compressed sensing. J. Comput. Math **35**(4), 437–449 (2017)
20. Yu, D., Deng, L.: Automatic Speech Recognition: A Deep Learning Approach. Signals and Communications Technology. Springer, Berlin (2015)