# Linear Feature Transform and Enhancement of Classification on Deep Neural Network

Penghang Yin*, Jack Xin*, and Yingyong Qi *

**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 principal component analysis. The feature transform is independent of the network structure, and can be applied without retraining the feature extraction layers of the network.

**Keywords:** Linear feature transform, difference of weighted nuclear norms, deep neural networks, enhanced classification.

**AMS subject classifications:** 68W40, 15A04, 90C26, 15A18.

*Department of Mathematics, UC Irvine, Irvine, CA 92697. This work was partially supported by NSF grants DMS-1222507, DMS-1522383, and a Qualcomm Research Gift to UC Irvine.

# 1 Introduction

Deep neural networks (DNN, [6, 4, 11]) are the state of the art methods in object classification tasks in computer vision [8] among other fields [16]. The basic form of DNN is convolutional neural networks (CNN) [1, 2]. An open source platform to study CNN on handwritten digits (MNIST [7]) and image classification (CIFAR [5]) is CAFFE [3]. Typically, a large number of multi-scale features arise from DNN [4, 5, 6, 11]. 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 [9]. 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.

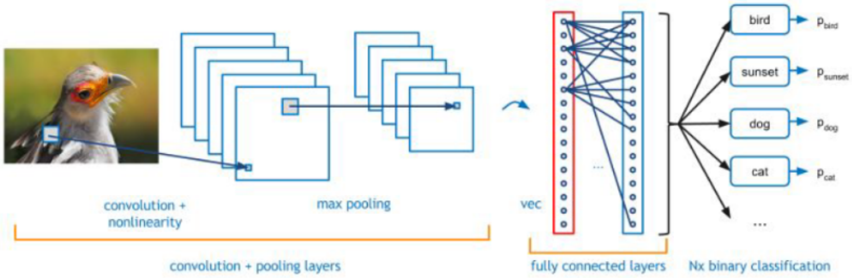

Figure 1: An illustration of DNN for image classification. From left to right: multi-layers of feature extractions involve convolution and nonlinearities, the last layer is fully connected and sends output to a classifier.

In this paper, we study such a geometrically motivated linear feature transform (LFT) at the output of the last fully connected layer of DNN before the classifer, see Fig. 1 for an illustration. We shall work with the existing LeNet and cuda-convnet [1] 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 the network where the original features come from*. Moreover, the *LFT model and algorithm can be applied to most DNNs* and be used as a low dimensional proxy.

The major assumption of LFT is that the feature vectors approximately lie in a subspace and thus have low dimensional structure. Therefore, we can find a linear transform $T \in \mathbb{R}^{m\times n}$, such that dimension of the transformed features $TY$ is greatly reduced (i.e. $m \ll n$), and meanwhile the classification performance is maintained. The advantages of having low dimensional features include speed up of computation during inference stage of network, as well as low memory and low energy consumption demand on mobile devices.

The paper is organized as follows. In section 2, we revisit the LFT model of [9] and observe a possible rank deficiency. The norm constraint of the model [9] prevents the iterations from approaching zero but may not exclude rank degeneracy of the transform. We also note that the LFT algorithm of [9] is not descending in general. To fix these issues, we propose a weighted difference of convex function (DC) model augmented with a convex regularization. In section 3, we present the associated DC algorithm (DCA) and show that it is descending under certain conditions on the weighting and penalty parameters. In section 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 via PCA by a factor of 32 while the accuracy is nearly maintained. The lower the dimension, the higher the enhancement. Concluding remarks are in section 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 [13]

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

# 2 LFT and Nuclear Norm Models

In this section, we review the LFT nuclear norm model of [9], point out the rank defects and propose our weighted-regularized model for DNN experiments in section 4.

In [9], 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 [9] that the objective function in (2.1) satisfies

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

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) [12, 14, 15] via the iteration:

$$T^{k+1} = \arg\min_{T} \sum_{i=1}^{c} \|TY_i\|_* - \langle S^k, TY \rangle \quad \text{s.t.} \quad \|T\| = 1. \tag{2.2}$$

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, (2.2) is still a non-convex program because of the constraint.

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

$$\sum_{i=1}^{c} d_i \leq n, \tag{2.3}$$

where $d_i$ is the dimension of the subspace of the $i$-th class. However, this condition is somewhat 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 (2.2) of DCA is non-convex due to the norm constraint which is implemented by normalization in [9], the iteration sequences from (2.2) 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:

$$\min_{T} \Psi(T) := \sum_{i=1}^{c} \|TY_i\|_* - w\|TY\|_* + \frac{\lambda}{2}\|T - P\|_F^2. \tag{2.4}$$

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.

# 3 Algorithms

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

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

By the definition of subgradient, we have

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

As a result,

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

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 (2.4) is:

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

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 (2.4) is always descending provided that (3.1) is solved properly, which is a nice mathematical property to have.

## 3.1 Convergence

Next we show that the objective in (2.4) has a lower bound under mild conditions, and thus $\{\Psi(T^k)\}$ converges.

**Proposition 3.1.** *For any fixed $w \geq 1$ and $\lambda > w-1$, $\sum_{i=1}^{c} \|TY_i\|_* - w\|TY\|_* + \frac{\lambda}{2}\|T - P\|_F^2$ is bounded from below.*

*Proof.* As $\sum_{i=1}^{c} \|TY_i\|_* - \|TY\|_* \geq 0$, it suffices to show that $\frac{\lambda}{2}\|T-P\|_F^2 - (w-1)\|TY\|_*$ has lower bound. By an alternative definition of nuclear norm [10],

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

therefore,

$$\|TY\|_* \leq \frac{1}{2}(\|T\|_F^2 + \|Y^{\mathrm{T}}\|_F^2).$$

Then we have

$$\begin{aligned}
\frac{\lambda}{2}\|T-P\|_F^2 - (w-1)\|TY\|_* &\geq \frac{\lambda - w + 1}{2}\|T\|_F^2 - \lambda\langle T, P\rangle + \frac{\lambda}{2}\|P\|_F^2 - \frac{w-1}{2}\|Y^{\mathrm{T}}\|_F^2 \\
&= \frac{\lambda - w + 1}{2}\|T - \frac{2\lambda}{\lambda - w + 1}P\|_F^2 \\
&\quad + (\frac{\lambda}{2} - \frac{2\lambda^2}{\lambda - w + 1})\|P\|_F^2 - \frac{w-1}{2}\|Y^{\mathrm{T}}\|_F^2 \\
&\geq (\frac{\lambda}{2} - \frac{2\lambda^2}{\lambda - w + 1})\|P\|_F^2 - \frac{w-1}{2}\|Y^{\mathrm{T}}\|_F^2
\end{aligned}$$

□

We also show 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). Then $\Psi(T^k) - \Psi(T^{k+1}) \geq \frac{\lambda}{2}\|T^k - T^{k+1}\|_F^2$, and $\|T^k - T^{k+1}\|_F \to 0$ as $k \to \infty$.*

*Proof.*

$$\begin{aligned}
\Psi(T^k) - \Psi(T^{k+1}) =& \frac{\lambda}{2}\|T^k - T^{k+1}\|_F^2 + \lambda\langle T^k - T^{k+1}, T^{k+1} - P\rangle \\
&+ w(\|T^{k+1}Y\|_* - \|T^kY\|_*) + \sum_{i=1}^{c}(\|T^kY_i\|_* - \|T^{k+1}Y_i\|_*)
\end{aligned} \tag{3.2}$$

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 $1 \leq i \leq c$, such that

$$\sum_{i=1}^{c} L_i^{k+1}Y_i^{\mathrm{T}} - wS^kY^{\mathrm{T}} + \lambda(T^{k+1} - P) = 0,$$

and therefore,

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

Plug into (3.2), we have

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

In the above arguments, we used the facts that

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

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

$$\langle S^k, TY\rangle \leq \|TY\|_*, \text{ for all } T \in \mathbb{R}^{m\times n}$$

with equality at $T = T^k$.

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

## 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 $\Lambda$, we first recast (3.1) as

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

and then form the augmented Lagrangian:

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

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:

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

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,

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

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

# 4 Numerical experiments

We present numerical experiments on the benchmark image datasets MNIST [7] and CIFAR-10 [5], 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. 2 for sample images from the datasets.

We extract both training and testing features through trained convolutional neural nets (CNN) on Caffe [3]. Caffe is a deep learning framework developed by the Berkeley Vision and Learning Center and by community contributors. LeNet [7] and cuda-convnet [1] are two baseline CNN on Caffe, working with MNIST and CIFAR-10 datasets respectively. The extracted features of CIFAR-10 images through cuda-convnet

Figure 2: Left: sample images of handwritten digits in MNIST. Right: 10 random example images from each class in CIFAR-10.

are 3-D arrays of dimensions 64×4×4, 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 original training and testing data, and feed the transformed data to a single layer neural net classifier from Scikit-learn package implemented in Python. Comparison of PCA and PCA with LFT is shown in Tables 2 and 3.

For CIFAR-10, when feature dimensions are reduced to 64 and 32, the accuracy dropped noticeably. The LFT can further improve the accuracy on top of PCA. The $P$ in model (2.4) is provided by PCA, with parameters $w = 3$ and $\lambda = 200$. The additional gain from LFT is 2% at feature dimension 32, and 0.7% at dimension 64. For MNIST, when reduced feature dimensions are 8 and 16, LFT improves the accuracy by 1.8% and 0.2% respectively. We can see from both cases that the lower the reduced dimension, the greater the improvement from LFT. At the original high dimension, the improvement is minimal or absent as seen in Table 1.

Table 1: Accuracy in % for CIFAR-10.

| Dataset | Original | LFT |
|---|---|---|
| CIFAR10 | 81.77 | 81.97 |
| MNIST | 99.05 | 99 |

Table 2: Accuracy in % for CIFAR-10 with reduced feature dimensions.

| Reduced dim | PCA | PCA + LFT |
|---|---|---|
| 64 | 80.21 | 80.90 |
| 32 | 77.91 | 79.95 |

Table 3: Accuracy in % for MNIST with reduced feature dimensions.

| Reduced dim | PCA | PCA + LFT |
|---|---|---|
| 16 | 98.14 | 98.33 |
| 8 | 95.31 | 97.1 |

## 5 Concluding Remarks

From the experiments on MNIST and CIFAR-10, we found that LFT can improve even a state-of-the-art classifier based on the new model (2.4), although the improvement is not yet significant.

There are two fundamental challenges for the linear transform. One is that the feature vectors of each class may not lie in a subspace with low enough dimension causing limited enlargement of pairwise subspace angles. The other is that the training model (2.4) is independent of the classifier or its decision function which is nonlinear in DNN.

A future line of work is to seek a linear transform to maximize the classification objective directly, for example in conjunction with adjusting the weight of the final fully connected layer in DNN. This way, the linear transform is not prone to the two restrictions above and may potentially improve accuracy more at the cost of retraining the original network via backpropagation.

## 6 Acknowledgements

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

## References

[1] cuda-convnet, `https://code.google.com/p/cuda-convnet`.

[2] L. Deng, D. Yu, ”Deep Learning: Methods and Applications”, NOW Publishers, 2014.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093, 2014.

[4] G. Hinton, *Learning multiple layers of representation*, Trends in Cognitive Sci, 11(10), pp. 428-434, 2007.

[5] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, 2009. `www.cs.toronto.edu/~kriz/index.htm`.

[6] Y. LeCun, L. Bottou, G. Orr, and K. Müller, ”Neural Networks: Tricks of the Trade”, Springer (1998).

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

[8] A. Krizhevsky, I. Sutskever, G. Hinton, *Imagenet classification with deep convolutional neural networks*, In: Advances in neural information processing systems, pp. 1097–1105, 2012.

[9] Q. Qiu, G. Sapiro, *Learning Transformations for Clustering and Classification*, J. Machine Learning Research 16 (2015), pp. 187-225.

[10] B. Recht and C. Ré, *Parallel Stochastic Gradient Algorithms for Large-Scale Matrix Completion*, Mathematical Programming Computation, 5(2):201-226, 2013.

[11] J. Schmidhuber, *Deep Learning in Neural Networks: An Overview*, arXiv:1404.7828v4 (2014).

[12] P. D. Tao and L. T. H. An, *A DC optimization algorithm for solving the trust-region subproblem*, SIAM Journal on Optimization, 8(2), pp. 476-505, 1998.

[13] G.A. Watson, *Characterization of the subdifferential of some matrix norms*, Linear Algebra and its Applications, 170 (1992), pp. 33-45.

[14] P. Yin and J. Xin, *PhaseLiftOff: an Accurate and Stable Phase Retrieval Method Based on Difference of Trace and Frobenius Norms*, Communications in Mathematical Sciences, Vol. 13, No. 2, pp. 1033-1049, 2015.

[15] P. Yin and J. Xin, *Iterative $\ell_1$ Minimization for Non-convex Compressed Sensing*, UCLA CAM Report 16-20.

[16] D. Yu, L. Deng, “Automatic Speech Recognition: A Deep Learning Approach”, Signals and Comm. Technology, Springer, 2015.