A Fast Proximal Method for Convolutional Sparse Coding

Rakesh Chalasani Jose C. Principe Naveen Ramakrishnan

***Abstract*—Sparse coding, an unsupervised feature learning technique, is often used as a basic building block to construct deep networks. Convolutional sparse coding is proposed in the literature to overcome the scalability issues of sparse coding techniques to large images. In this paper we propose an efficient algorithm, based on the fast iterative shrinkage thresholding algorithm (FISTA), for learning sparse convolutional features. Through numerical experiments, we show that the proposed convolutional extension of FISTA can not only lead to faster convergence compared to existing methods but can also easily generalize to other cost functions.**

***Index Terms*—Convolution, Sparse Coding, Feature Extraction, Unsupervised Learning.**

I. Introduction

Choosing the appropriate data representation (i.e. feature space) that has desirable properties for a given task (e.g. classification, clustering) is a central issue in statistical machine learning. This issue becomes all the more important for computer vision tasks (e.g. object detection) due to high dimensionality of input features and high variability in instances. Hand crafted features like SIFT [1], HoG [2], etc., have been successful to over come these issues. Recent developments in deep architectures, where multiple layers of feature extractors (e.g. RBM, sparse coding blocks) are stacked, attempt to learn the features automatically in an unsupervised manner using large-scale unlabeled data [3], [4]. Some of the popular deep networks use sparse encoders as building blocks [5], [6], [7] and hence, in this work we focus on these sparse coding blocks.

Generally, sparse coding is based on the idea that an observation, $\mathbf{y} \in \mathbb{R}^p$, can be encoded using an over-complete dictionary of filters, $\mathbf{C} \in \mathbb{R}^{p \times k}; (k > p)$ and a sparse vector $\mathbf{x} \in \mathbb{R}^k$. More formally this can be written as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{C}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \qquad (1)$$

The ℓ_1-norm on $\mathbf{x}$ ensures that the latent vector is sparse. Several efficient solvers like coordinate descent (CoD) [8], fast iterative shrinkage thresholding algorithm (FISTA) [9], feature-sign algorithm [10], etc, can be readily applied to solve the above optimization problem. The dictionary $\mathbf{C}$ can also be learned from the data [10], [11].

However, in most applications of sparse coding many overlapping patches across the image are processed separately. This is often too slow in practice, making it difficult to scale to large images. Moreover, sparse coding alone is not capable of encoding translations in the observations. Learning the dictionary in this context produces several shifted versions of the same filter, such that each patch can be reconstructed individually [12]. During inference, when performed on all the overlapping patches, this can lead to a very redundant representation. To over come these limitations, convolutional sparse coding is proposed [5], [12]. Here sparse coding is applied over the entire image and the dictionary is a convolutional filter bank with 'M' kernels such that,

$$x = \arg\min_{x} \frac{1}{2}\|\mathcal{I} - \sum_{m=1}^{M} C_m * x_m\|_2^2 + \lambda \sum_{m=1}^{M} \|x_m\|_1 \qquad (2)$$

where $\mathcal{I}$ is an image of size $(w \times h)$, C_m is a filter kernel of size $(s \times s)$ in a dictionary, x_m is a sparse matrix of size $(w + s - 1) \times (h + s - 1)$, λ is the sparsity parameter and '$*$' represents a 2D convolution operator [1].

In this work, we propose an algorithm to solve the optimization problem in (2) efficiently and which scales to large images. This is an extension to fast iterative shrinkage thresholding algorithm (FISTA) [9] and is solved using proximal gradient. In addition, we also extend our method to include a feed-forward predictor (predictive sparse decomposition [6]) in the cost for joint optimization during inference and learning.

Convolutional sparse coding is also previously studied in [5], [12]. Zeiler et. al. [5] proposed a method to solve the optimization problem in (2) by introducing additional auxiliary variable which demands solving a large linear system (the size of the linear system is proportional to size of the image) at every iteration; although the complexity could be reduced by using conjugate gradient, their approach does not scale to large images. On the other hand, the authors in [12] proposed a convolutional extension to CoD where the computation per iteration is small compared to the method in [5] but the number of iterations required for convergence becomes large for large images. In the following sections, we compare and contrast convolutional CoD with our method to show the performance improvements achieved.

The rest of the paper is organized as follows: section II describes convolutional FISTA and procedures to learn the parameters of the model. Also, the extension of the method for predictive sparse decomposition (PSD) is discussed. Several experiments are described in section III to show the performance of the proposed method and compare with the

Rakesh Chalasani and Jose C. Principe are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA . Naveen Ramakrishnan is with Robert Bosch LLC, Research and Technology Center North America, Pittsburgh, PA, USA. (email: rakeshch@ufl.edu, principe@cnel.ufl.edu,Naveen.Ramakrishnan@us.bosch.com)

This work is partially supported by ONR grant $\#N000141010375$. The first author performed part of this work while at Bosch RTC, Pittsburgh during summer 2012.

[1]All the variable henceforth represent a matrix, unless otherwise stated. Also the convolution operators is applied in 'full' or 'valid' modes, depending on the context.

existing methods. The paper concludes in section IV.

II. Convolutional FISTA and Dictionary Learning

Since the cost function in (2) is not jointly convex in both C and x, learning the dictionary of filters involves a block coordinate descent kind of optimization [10], [11]; where in the objective in (2) is alteratively minimized w.r.t x and C while keeping one of them fixed. This section describes the convolutional generalization of the popular sparse coding algorithm FISTA for inferring the latent variable x while C is held fixed. Then, a procedure for updating the parameters in C with fixed x is described. In addition to this, the effectiveness of the proposed method to generalize to other cost functions is shown using predictive sparse decomposition (PSD) [12].

A. Inference

ISTA is a popular *first order* proximal method to infer sparse codes from a linear inverse problem in (1). That is, it has the advantage of being a simple gradient based algorithm involving simple computations like matrix multiplications with $\mathbf{C}$ and $\mathbf{C}^T$ followed by a soft thresholding function. However, it tends to have a very slow convergence rate. To overcome this, Beck and Teboulle [9] have proposed FISTA, which has significantly better global convergence rate while preserving the computational simplicity of ISTA. In fact, FISTA can be generalized to any non-smooth convex optimization problem with cost function of the form:

$$\mathcal{F}(x) = f(x) + g(x) \tag{3}$$

where f, g are convex and g is possibly non-smooth [9]. Most notable advantage of FISTA is that it keeps ISTA's number of gradient evaluations (just one per iteration) but involves finding a point that is smartly chosen for updating the latent variable, x. This additional point is a function of the difference between the previous two updates of x. Such "momentum" term introduced into the update helps to obtain faster convergence and is shown to have a convergence rate of order $\mathcal{O}(1/k^2)$, where k is the number of iterations.

A straight forward way to use FISTA (or any sparse coding method) to solve (2) is by replacing the convolution operator by a matrix-vector product. The matrix, equivalent to the dictionary in (1), is made by first constructing a "Toeplitz" matrix for each filter containing all its shifted versions and then concatenating all such matrices. The resulting problem is similar to (1) and FISTA is guaranteed to converge for such a problem. However, in practice working on such large matrices can be computationally expensive and might be unnecessary if one can take advantage of the convolutional nature of the formulation.

As discussed before, FISTA involves computing the gradient of the convex part of the cost function in (2). So, the key to computational simplicity in this work comes from computing this gradient efficiently and can be obtained as follows: the derivative of f with respect to z_n (nth map corresponding to the nth filter) is given by

$$\nabla f(z_n) = C_n' * \big(\mathcal{I} - \sum_{m=1}^{M} C_m * z_m\big) \tag{4}$$

where C_n' is equivalent to 180^o rotation of matrix C. This leads to very efficient computation of the gradient, while avoiding constructing any large matrices.

Equipped with this efficient gradient computation, we can easily extend FISTA for convolutional sparse coding. Algorithm 1 describes the convolutional generalization of FISTA; where $p_{L^k}(.)$, for ℓ_1 regularization in the cost function, is given by

$$p_{L^{(k)}}(z^{(k)}) = \mathcal{T}_{\lambda/L^{(k)}}\big(z^{(k)} - \frac{1}{L^{(k)}}\nabla f(z^{(k)})\big) \tag{5}$$

where $\mathcal{T}_\alpha(.)$ is an elementwise soft thresholding function

$$\mathcal{T}_\alpha(x_i) = (|x_i| - \alpha)_+ sgn(x_i) \tag{6}$$

where x_i indicates the ith element of a point x in the latent space. $\nabla f(z)$ is the gradient of the quadratic term in (2), denoted by f, at some point z in the latent space and is computed from (4). Also, the line search to find the appropriate step size, L_k, requires computing $\mathcal{F}(x)$, the cost function in (2), and $\mathcal{Q}(p_L(x), x)$ given by

$$\mathcal{Q}(p_L(x), x) = f(x) + \langle p_L(x) - x, \nabla f(x)\rangle + \frac{L}{2}\|p_L(x) - x\|^2 + |p_L(x)|_1$$

where $f(x)$ is the quadratic loss in (2) and $\langle.\rangle$ is the inner product between two vectorized quantities.

Note that, compared to convolutional CoD [12], the computations required per iteration here are more. However, because of the "momentum" term introduced in step 7 of Algorithm.1, the number of iterations required for convergence are much less.

Algorithm 1 Convolutional extension for fast iterative shrinkage thresholding algorithm (FISTA).

Require: Input image: $\mathcal{I}$, $L^{(0)} > 0$, $\eta > 1$ and $x^{(0)} \in \mathbb{R}^{(w+s-1)\times(h+s-1)\times M}$

1: Initialize $z^{(1)} = x^{(0)}$, $t^{(1)} = 1$ and $k = 0$.
2: **while** (convergence) **do**
3: $k = k + 1$
4: *Line search*: Find the smallest non-negative integer $i^{(k)}$ such that with $\bar{L} = \eta^{i^{(k)}} L^{(k-1)}$

$$\mathcal{F}(p_{\bar{L}}(z^{(k)})) \leq \mathcal{Q}_{\bar{L}}(p_{\bar{L}}(z^{(k)}), z^{(k)})$$

5: Set $L^{(k)} = \eta^{i^{(k)}} L^{(k-1)}$
6: $x^{(k)} = p_{L^{(k)}}(z^{(k)})$
7: $t^{(k+1)} = \frac{1+\sqrt{1+4t^{(k)2}}}{2}$
8: $z^{(k+1)} = x^{(k)} + \left(\frac{t^{(k)}-1}{t^{(k+1)}}\right)(x^{(k)} - x^{(k-1)})$
9: **end while**
10: **return** $x^{(k)}$

B. Dictionary Learning

With the inferred latent variable x fixed, the dictionary of filters are updated using gradient descent method. For faster convergence limited-memory BFGS [13] is used for updating the filters over a batch of images. The gradient w.r.t to a single filter, C_n is

$$\nabla f(C_n) = \sum_{b=1}^{B} x'_{b,n} * (\mathcal{I}_b - \sum_{m=1}^{M} C_m * x_{b,m}) \tag{7}$$

where B is the number of images in a batch and $x'_{b,m}$ is computed in a similar manner to C'_m described in (4). Each filter is normalized to have a unit norm post update to avoid any redundant solution.

The above described inference and learning procedure can be readily applied for image denoising [14] and constructing deep networks like deconvolutional network [5]. However, unlike convolutional CoD, FISTA can be generalized to other cost functions easily, like predictive sparse decomposition (PSD) described below.

C. Extension to PSD

The cost function used in PSD [6] contains two parts: a decoder, containing a quadratic cost function along with a sparsity constraint similar to (2) and feed-forward, non-linear encoder to approximate the sparse code. The cost function with this additional prediction term looks like:

$$\begin{aligned} x = \arg\min_x \frac{1}{2}\|\mathcal{I} - \sum_{m=1}^{M} C_m * x_m\|_2^2 \\ + \sum_{m=1}^{M} \|x_m - \varphi(W_m * \mathcal{I})\|_2^2 + \lambda \sum_{m=1}^{M} \|x_m\|_1 \end{aligned} \tag{8}$$

where $W_m \in \mathbb{R}^{s \times s}$ is a feed-forward filter and $\varphi(.)$ is a non-linear function to approximate sparse codes. For further details refer to [12].

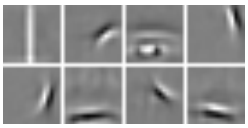
Now, the first two quadratic terms in (8), referred to as f below, are convex and continuously differentiable with respect to x and this can be readily solved using convolutional FISTA described above with gradient w.r.t x_n computed as

$$\nabla f(x_n) = C'_n * \Big(\mathcal{I} - \sum_{m=1}^{M} C_m * x_m\Big) + \Big(x_n - \varphi(W_n * \mathcal{I})\Big)$$

Note that convolutional CoD cannot deal with this cost function, without losing its efficiency.

III. Experiments

In this section, convolutional FISTA (ConvFISTA) is compared with the other existing algorithms. Gray scale images of different sizes are used for testing purposes. All the images are first pre-processed such that their mean is subtracted and then contrast normalized using a 5×5 average filter. This step ensures that the low-frequency components are removed and high frequency edges are made prominent. This helps to speed up the learning and inference during sparse coding [15].



(a) Learned convolution filters

(b) Preprocessed input image.

(c) Reconstructed image

Fig. 1: Shows (a) the dictionary of filters learned from the face images, (b) original contrast normalized image and (c) reconstruction of the images from the inferred variables.

A. Dictionary Learning

Face images from AT&T database [16] are used for learning the dictionary. Each gray scale image is first resized to be 64×64 pixel and pre-processed as stated above. From here $8,\ 16 \times 16$ convolutional filters are learned using a batch of 20 images per epoch. Fig.1 shows the results obtained. It is important to note that taking 8 convolutional filters makes the system that may times over-complete at every point. This typically leads to a more sparse representation that what would be achieved from patch level sparse coding, because each element in the latent variables competes with the other filters not just to explain one point but a local neighborhood in the image.

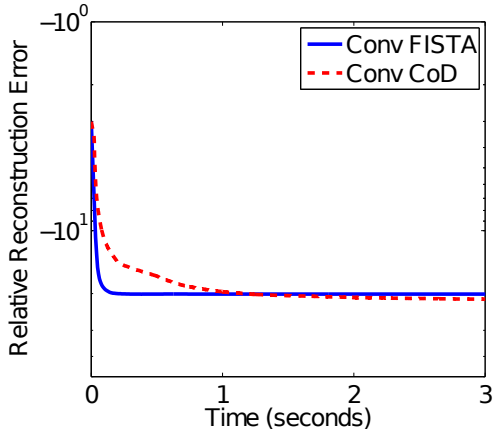
B. Convergence Rate

As discussed before, one of the important advantages of ConvFISTA is its faster convergence rate during inference. To compare the performance of the proposed model with convolutional CoD (ConvCoD) [12], the ability of both the methods to reconstruct the image, with a pre-learned dictionary, is considered as the metric. In other words, the relative error between the true (contrast normalized) image and reconstructed image obtained from the inferred latent representation

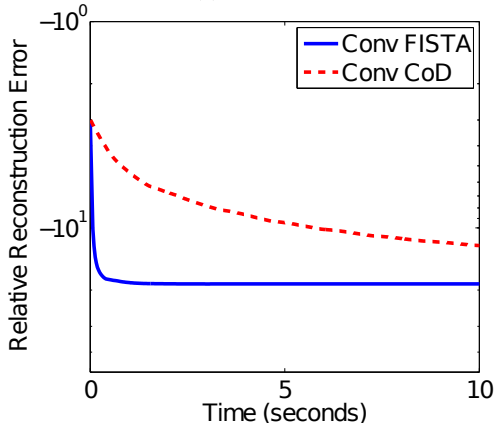
$$\mathcal{M} = \frac{\|\mathcal{I} - \hat{\mathcal{I}}\|_2^2}{\|\mathcal{I}\|_2^2} \tag{9}$$

where $\hat{\mathcal{I}}$ is the reconstructed image, is used as the metric.

To make a fair comparison, the same model is used for both the methods, i.e., the sparsity parameter is held fixed ($\lambda = 0.1$) and a fixed dictionary of $32,\ 9 \times 9$ filters learned from Berkeley segmentation dataset [17] is used. Images from the same dataset of varying size are used to compare the performance with scale. Fig.2 shows the comparison. It is observed that ConvFISTA is able to out perform ConvCoD in terms of convergence rate. The difference in performance becomes more pronounced with increase in the size of the images because ConvCoD, which updates only one carefully

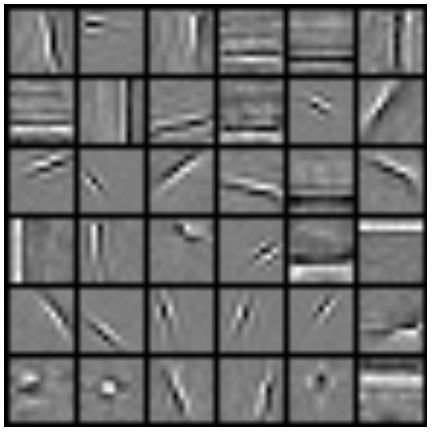


(a) 50×50

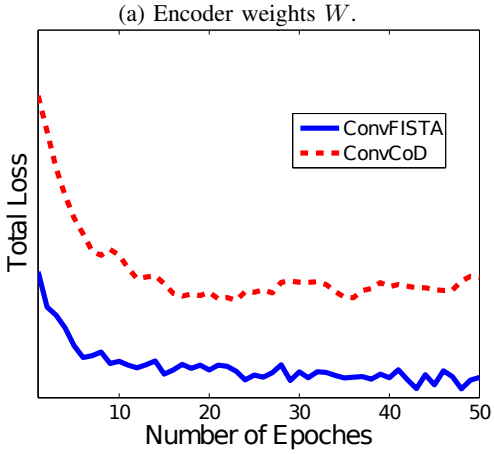


(b) 150×150

Fig. 2: Comparison of the convergence rate between convolutional CoD (ConvCoD) and convolutional FISTA (ConvFISTA). Two different images sizes,(a) 50×50 pixel and (b) 150×150 pixel, are used for comparison.



(a) Encoder weights W.



(b) Instantaneous total loss.

Fig. 3: (a) Shows $36, 11 \times 11$ encoder filter weights learned using ConvFISTA. They resemble in the learned dictionary filters (not shown). (b) Shows the instantaneous total loss at every epoch for both ConvFISTA and ConvCoD.

chosen element at every iteration, requires more iterations for convergence.

C. Predictive Sparse Decomposition

As described in section. II-C, predictive sparse coding (PSD), used as a basic block for building deep networks, has a decoder (involving C) and an feed-forward encoder (involving W) in the cost function as in (8). However, once the model is trained, the decoder is discarded while the encoder alone is used for approximate inference. Therefore, it is important to minimize the joint cost function, so that the encoder would be able to approximate the decoder function more accurately.

Here ConvFISTA is used for inferring the latent representation from the joint cost function involving both the decoder and the encoder. The non-linear function used in the encoder is considered as in [18] with: $\varphi_m(\mathcal{I}) = \beta_m(tanh(W_m * \mathcal{I} - \theta) + tanh(W_m * \mathcal{I} + \theta))$. The parameters of the encoder, β and W, are also learned along with the convolution filters, C (each one is updated while all the others are held fixed). On the other hand, for comparison we use the procedure described in [12], where the encoder is learned separately as feed-forward model, similar to convolutional networks [19], after obtaining the sparse representation using ConvCoD. Results obtained a shown in Fig.3. Similar to the previous experiment, Berkeley segementation dataset [17] is used, with each image re-sized to be 150×150 pixel and preprocessed. Fig.3a shows the learned encoder weights corresponding to $36, 11 \times 11$ convolutional filters. We observe (Fig.3b) that the total loss, the quadratic term plus the prediction term in (8), reaches a lower value when using ConvFISTA for jointly optimizating the cost than using ConvCoD for separately training the model.

IV. Conclusion

In this paper, we proposed a convolutional extension to the FISTA algorithm for solving the sparse coding problem. The convolutional FISTA proposed here has two major advantages: (1) Unlike ConvCoD, where only one carefully chosen element in updated at every iteration, ConvFISTA updates all the elements in parallel and achieves faster convergence rate and (2) because of simple first order updates, ConvFISTA

can be easily generalized to other convex loss functions as well.

With the help of recent advances in using proximal methods for structured sparsity models [20], it might be possible to extend the method proposed here to learn more complex structures within the convolutional filters; for example, introducing tree structured relationship between the filters might better represent the variations between different features within an image. Such extensions still remains to be studied.

REFERENCES

[1] David G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, Washington, DC, USA, 1999, ICCV '99, pp. 1150–, IEEE Computer Society.

[2] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, Washington, DC, USA, 2005, CVPR '05, pp. 886–893, IEEE Computer Society.

[3] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comp.*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[4] Yoshua Bengio, "Learning Deep Architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[5] M.D. Zeiler, D. Krishnan, G.W. Taylor, and R. Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[6] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun, "Fast inference in sparse coding algorithms with applications to object recognition," *CoRR*, vol. abs/1010.3467, 2010.

[7] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, 2009, ICML '09, pp. 609–616, ACM.

[8] Y. Li and S. Osher, "Coordinate descent optimization for 1 minimization with application to compressed sensing; a greedy algorithm," *Inverse Problems and Imaging*, vol. 3, no. 3, pp. 487–503, 2009.

[9] Amir Beck and Marc Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[10] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, "Efficient sparse coding algorithms," in *In NIPS*, 2007, pp. 801–808.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 689–696.

[12] K. Kavukcuoglu, P. Sermanet, Y.L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition," *Advances in Neural Information Processing Systems*, pp. 1090–1098, 2010.

[13] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[14] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[15] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images.," *Nature*, vol. 381, no. 6583, pp. 607–609, June 1996.

[16] Ferdinando S Samaria and Andy C Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. IEEE, 1994, pp. 138–142.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, July 2001, vol. 2, pp. 416–423.

[18] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Johannes Fürnkranz and Thorsten Joachims, Eds., Haifa, Israel, June 2010, pp. 399–406, Omnipress.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[20] X. Chen, Q. Lin, S. Kim, J.G. Carbonell, and E.P. Xing, "Smoothing proximal gradient method for general structured sparse regression," *The Annals of Applied Statistics*, vol. 6, no. 2, pp. 719–752, 2012.