Unsupervised Deep Transform Learning

Jyoti Maggu
IIIT-Delhi
jyotim@iiitd.ac.in

Angshul Majumdar
IIIT-Delhi
angshul@iiitd.ac.in

***Abstract*— We introduce deep transform learning – a new tool for deep learning. Deeper representation is learnt by stacking one transform after another. The model is akin to a feedforward neural network. The first layer learns the transform and features from the input training samples. Subsequent layers use the features (after activation) from the previous layers as training input. However, this explanation is only given for intuitive understanding; the ensuing problem is not solved in a greedy fashion. All the layers are solved jointly. Experiments have been carried out with other state-of-the-art. Results on classification and clustering show that our proposed technique is better than all the said techniques, at least on the benchmark datasets compared on.**

***Keywords*— deep learning, transform learning**

I. INTRODUCTION

Transform learning is an analysis formulation; the transform is learnt such that it analyses the signal to generate coefficients. It has been introduced only recently [1-4]. So far transform learning has been primarily applied for solving inverse problems like denoising and reconstruction; arising in signal processing. There are a few short studies that have used it for unsupervised representation learning [5-7]. Dictionary learning is its synthesis equivalent; a basis (dictionary) is learnt so as to regenerate / synthesize the data from the corresponding representation. Dictionary learning has its roots in matrix factorization [8] / sparse coding [9]. Its popularity has resurfaced in the last decade with the success of KSVD [10]; the unsupervised version is mostly used in signal processing for solving inverse problems [11, 12]. In machine learning dictionary learning has been used immensely for supervised classification (e.g. [13]) and clustering (e.g. [14]). In recent years kernelized versions of dictionary learning also being proposed for machine learning problems [15-17].

The success of deep learning is popular knowledge today. Deeper architectures like stacked autoencoder (SAE) and deep belief network (DBN) are built by stacking shallow representation learning tools like autoencoder and restricted Boltzmann machine (RBM) in a layered fashion.

Dictionary learning, although well known in the machine learning community, has been traditionally overlooked as a candidate for building deep architectures. It is only in recent times, layers of single level dictionaries have been stacked one after the other leading to the framework of deep dictionary learning (DDL) [18, 19]. The framework has been successfully used for addressing real world problems in hyper-spectral imaging [20], biometrics [21] and energy analytics [22] among other areas.

Following the success of deep dictionary learning, this work introduces deep transform learning. The main idea is to learn a representation through multiple layers of transforms connected via non-linear activations. A rudimentary sub-optimal solution to this problem has been proposed by the authors in [23]; the aforesaid solution is greedy – learning proceeds layer-wise without any cross-talk between the layers. This corresponds to the greedy pre-training paradigm in deep learning without any fine-tuning.

In this work we propose an optimal solution where all the layers are learnt jointly. The transition from greedy to the proposed optimal solution is not trivial. The greedy solution is easy, since it does not need any algorithmic development; the standard transform learning technique [1-4] can be used. For the proposed optimal learning paradigm, one needs to derive an algorithm. However the results will show the benefits of the optimal solution; the results improve significantly from the greedy solution.

II. LITERATURE REVIEW

As mentioned before transform learning is the analysis equivalent of dictionary learning. Although the technique is known to the signal processing community we review it for the sake of completeness. The model is expressed as

$$TX = Z \tag{1}$$

Here T is the transform, which operates on the data X to generate the representation Z.

The optimization problem for transform learning is expressed as –

$$\min_{T,Z} \|TX - Z\|_F^2 + \lambda\left(\|T\|_F^2 - \log\det T\right) + \mu\|Z\|_1 \tag{2}$$

The l_1-norm enforces sparsity on the representation. This is important for inverse problems arising in signal processing; but there is no intuitive of theoretical reason for enforcing this for problems arising in machine learning – apart from a regularization term.

The factor $-\log\det T$ imposes a full rank on the learned transform; this prevents the degenerate solution ($T=0$, $Z=0$). The additional penalty $\|T\|_F^2$ is to balance scale; without this $-\log\det T$ can keep on increasing producing degenerate results in the other extreme.

The minimization problem (2) is solved by alternately updating the two variables.

$$Z \leftarrow \min_{Z} \|TX - Z\|_F^2 + \mu \|Z\|_1$$

$$T \leftarrow \min_{T} \|TX - Z\|_F^2 + \lambda \left(\varepsilon \|T\|_F^2 - \log\det T \right)$$

Updating the coefficients is straightforward. It can be updated via one step of soft thresholding. There is a closed form update for the Transform as well. This is given by –

$$XX^T + \lambda \varepsilon I = LL^T$$

$$L^{-1} XZ^T = USV^T$$

$$T = 0.5R\left(S + (S^2 + 2\lambda I)^{1/2}\right) Q^T L^{-1}$$

Our proposed deep transform learning is the multi-layer extension of the shallow one. It can be thought of as application of multiple levels of transforms to generate the coefficients. Mathematically this is expressed as follows –

$$T_N(\varphi...(T_2(\varphi(T_1 X))) = Z \tag{3}$$

Here φ denotes the activation function; without which all the transforms will collapse into a single one.

In [23], we proposed a greedy solution for (3); we solve one layer at a time. With the substitution $\varphi(T_{N-1}...(T_2(\varphi(T_1 X)) = \mathrm{Z}_{N-1}$, (3) can be expressed as,

$$T_N Z_{N-1} = Z \tag{4}$$

where $\varphi(T_{N-1}...(T_2(\varphi(T_1 X)) = \mathrm{Z}_{N-1}$. This can be alternately expressed as,

$$T_{N-1}...(T_2(\varphi(T_1 X)) = \varphi^{-1}(\mathrm{Z}_{N-1}) \tag{5}$$

With the substitution $\varphi(T_{N-2}...(T_2(\varphi(T_1 X)) = \mathrm{Z}_{N-2}$, we have for the next layer,

$$T_{N-1} Z_{N-2} = \varphi^{-1}(\mathrm{Z}_{N-1}) \tag{6}$$

Continuing the substitution in this fashion till the final layer, we have

$$T_1 X = \varphi^{-1}(\mathrm{Z}_1) \tag{7}$$

Note that for all the layers, it is easy to invert the activation φ since they operate element-wise.

We start solving for the different layers of transforms in backward direction; starting from (7); this is easily solved using the standard transform learning formulation (2).

Once Z_1 is solved for, it acts as the input for the second layer for solving T_2 and Z_2 from $T_2 Z_1 = \varphi^{-1}(\mathrm{Z}_2)$ by applying standard transform learning formulation (2). The same process is repeated for other layers. The advantage of such a greedy training paradigm is that for each level, one only needs solving a shallow transform learning problem which has algorithms with convergence guarantees.

During testing, the objective is to generate the test feature (z_{test}) given the input test sample (x_{test}). This is expressed as,

$$T_N(\varphi...(T_2(\varphi(T_1 x_{test}))) = z_{test} \tag{8}$$

The multiple layers of transforms have already been learnt during the training phase. Therefore during testing, one simply needs to apply them one after the other.

III. Proposed Solution

The greedy approach is sub-optimal since there is no flow of information from deeper to shallower layers. In deep learning this issue is addressed by backpropagation during the fine-tuning stage for supervised learning problems. We are proposing an unsupervised representation learning technique; there are no outputs to backpropagate from. Hence, we need to derive an algorithm for solving all the layers of transforms via joint optimization framework.

The deep transform learning formulation is given in (3). We repeat it for the sake of convenience.

$$T_N(\varphi...(T_2(\varphi(T_1 X))) = Z$$

The joint optimization problem we intend to solve is as follows,

$$\begin{aligned} &\min_{T_i's, Z} \|T_N(\varphi...(T_2(\varphi(T_1 X))) - Z\|_F^2 \\ &+ \lambda \sum_{i=1}^{N} \left(\|T_i\|_F^2 - \log\det T_i \right) \end{aligned} \tag{9}$$

This will be solved using the variable splitting approach; a technique that is gaining popularity in signal processing [24, 25]. Using the same substitutions as in the greedy approach, i.e., $\varphi(T_{N-1}...(T_2(\varphi(T_1 X)) = \mathrm{Z}_{N-1}$ we can express the augmented Lagrangian for (14) as,

$$\begin{aligned} &\min_{T_i's, Z, Z_{N-1}} \|T_N Z_{N-1} - Z\|_F^2 + \lambda \sum_{i=1}^{N} \left(\|T_i\|_F^2 - \log\det T_i \right) \\ &+ \mu \left\| T_{N-1}...(T_2(\varphi(T_1 X) - \varphi^{-1}(\mathrm{Z}_{N-1}) \right\|_F^2 \end{aligned} \tag{10}$$

As in the greedy solution, we have used the fact that inverting the activation function is trivial since it operates element-wise.

In the second, step we substitute, $\varphi(T_{N-2}...(T_2(\varphi(T_1 X)) = \mathrm{Z}_{N-2}$; this allows expressing (10) as follows (in terms of augmented Lagrangian),

$$\begin{aligned} &\min_{T_i's, Z, Z_{N-1}, Z_{N-2}} \|T_N Z_{N-1} - Z\|_F^2 + \mu \left\| T_{N-1} Z_{N-2} - \varphi^{-1}(\mathrm{Z}_{N-1}) \right\|_F^2 \\ &+ \lambda \sum_{i=1}^{N} \left(\|T_i\|_F^2 - \log\det T_i \right) \\ &+ \mu \left\| \varphi(T_{N-2}...(T_2(\varphi(T_1 X)) - \mathrm{Z}_{N-2} \right\|_F^2 \end{aligned} \tag{11}$$

Continuing in this fashion, with the final substitution $\varphi(T_1 X) = \mathrm{Z}_1$, we get the complete augmented Lagrangian formulation,

$$\min_{T_i\text{'s},Z,Z_i\text{'s}} \|T_N Z_{N-1} - Z\|_F^2 + \mu \sum_{i=2}^{N-1} \|T_i Z_{i-1} - \varphi^{-1}(Z_i)\|_F^2 + \mu \|T_1 X - \varphi^{-1}(Z_1)\|_F^2 + \lambda \sum_{i=1}^{N} \left(\|T_i\|_F^2 - \log\det T_i \right) \quad (12)$$

The alternating direction method of multipliers (ADMM) [29] allows (12) to be segregated into the following sub-problems.

$$P_1: \min_{T_N} \|T_N Z_{N-1} - Z\|_F^2 + \lambda \left(\|T_N\|_F^2 - \log\det T_N \right)$$

$$P_2: \min_{T_{N-1}} \mu \|T_{N-1} Z_{N-2} - \varphi^{-1}(Z_{N-1})\|_F^2 + \lambda \left(\|T_{N-1}\|_F^2 - \log\det T_{N-1} \right)$$

…

$$P_N: \min_{T_1} \mu \|T_1 X - \varphi^{-1}(Z_1)\|_F^2 + \lambda \left(\|T_1\|_F^2 - \log\det T_1 \right)$$

$$P_{N+1}: \min_{Z} \|T_N Z_{N-1} - Z\|_F^2 \Rightarrow T_N Z_{N-1} = Z$$

$$P_{N+2}: \min_{Z_{N-1}} \|T_N Z_{N-1} - Z\|_F^2 + \mu \|T_{N-1} Z_{N-2} - \varphi^{-1}(Z_{N-1})\|_F^2$$
$$\equiv \min_{Z_{N-1}} \|T_N Z_{N-1} - Z\|_F^2 + \mu \|\varphi(T_{N-1} Z_{N-2}) - Z_{N-1}\|_F^2$$

…

$$P_{2N}: \min_{Z_1} \|T_2 Z_1 - \varphi^{-1}(Z_2)\|_F^2 + \|\varphi(T_1 X) - Z_1\|_F^2$$

We see that the sub-problems P_1 to P_N are all standard transform updates. We already know how to solve them. Solving for the final / deepest representation is simple; follows from P_{N+1}. Solution of the intermediate representations are from P_{N+2} to P_{2N}. All of them are least squares problems in their equivalent form; hence has a closed form solution – pseudo-inverse.

Notice that we have kept the Lagrangian multiplier (μ) constant for all the layer. Moreover, it is imperative that we give equal importance to all the layers and hence we keep μ=1 throughout. This is usually not the case for generic optimization problems. However, in this work we argue that each of the layers have an equal importance, μ equaling unity is a logical choice.

This concludes the derivation of the joint optimization algorithm. Each of the sub-problems have closed form solutions. However unlike the greedy solution, which only required repeated application of transform learning for each layer – and hence was bound to converge (given the convergence guarantees of each layer), proving the convergence guarantees for our proposed joint optimization algorithm is beyond the scope of this work.

The testing stage remains the same as the greedy technique. One only needs to apply one transform after another in the correct order with the activation functions to generate the representation of a test sample.

Initialization

Unlike most deep learning techniques where all the layers are initialized randomly, we only have to initialize T_1, that too for solving $T_1 X = \varphi^{-1}(Z_1)$ from the training data. After solving the transform learning problem, the obtained value of T_1 becomes the initial value for the ensuing iterations.

Once the first layer of coefficient Z_1 is learnt, it is used to initialize the second layer of transform by solving $T_2 Z_1 = \varphi^{-1}(Z_2)$. The second layer representation is used to initialize the transform for the third layer and so on.

IV. Experimental Results

A. Clustering

There are only a handful of studies in deep learning based clustering. We compare with three major ones – deep dictionary learning [19], stacked autoencoder (SAE) [30] and Deep Subspace Clustering (DSC) [31]. We follow the experimental protocol found in [31]; the said paper compares [30] and [19] uses the same protocol as [31].

In the aforesaid paper, experiments were carried out on the COIL20 (object recognition) and Extended YaleB (face recognition) datasets. For both the datasets DSIFT (dense scale invariant feature transform) and HOG (histogram of oriented gradients) features were extracted. They were further reduced by PCA to a dimensionality of 300. Since the groundtruths (class labels) for these datasets are available, clustering accuracy was measured in terms of NMI (normalized mutual information), ARI (adjusted rand index) and F-score. The results are shown in Table I (COIL20) and Table II (YaleB).

The architectures used for comparison have been obtained from the published studies since they were the best performing ones according to the authors. Both SAE and DSC uses a five layer architecture with 300-200-150-200-300 nodes; both of them use tanh activation function. Prior study on DDL used a four tier architecture with 600-300-150-75 nodes; the activation function is tanh.

For our proposed deep transform learning (DTL) a three layer architecture (tanh activation) with 300-150-75 basis in each layer was used. It has already been mentioned that the value of the hyper-parameter μ is fixed at unity. The value of the parameter λ has been fixed to 0.1 throughout; we checked that the results are stable for values of λ between 0.05 and 0.5.

In DDL, SAE and our proposed technique, after the obtaining the coefficients from the final layer a simple K-means clustering is used. In [30] it has been found that using more sophisticated clustering techniques like spectral clustering does not improve the results significantly – they tested with K-means and spectral clustering. Following those results we chose to stick with simple K-means in this work; the same has been done in [19].

TABLE I
CLUSTERING ON COIL 20

Method	DSIFT			HOG		
	NMI	ARI	F-score	NMI	ARI	F-score
SAE	77.09	56.59	59.07	89.26	74.25	75.70
DSC	91.19	84.80	85.58	91.19	81.92	82.86
DDL	91.04	84.60	83.54	90.12	80.20	81.30
Greedy DTL	92.36	85.08	84.96	91.51	81.88	8259
Proposed	**94.38**	**86.92**	**88.56**	**93.37**	**84.24**	**85.82**

TABLE II
CLUSTERING ON YALEB

Method	DSIFT			HOG		
	NMI	ARI	F-score	NMI	ARI	F-score
SAE	87.54	75.82	76.50	93.43	82.57	83.07
DSC	90.85	83.00	83.45	96.91	90.25	89.46
DDL	90.20	81.83	83.42	96.82	88.97	89.13
Greedy DTL	91.27	82.51	84.66	97.12	90.56	90.48
Proposed	**93.26**	**85.62**	**85.86**	**98.93**	**93.43**	**92.06**

The results are consistent with prior studies. Both the greedy technique and the joint optimization technique excel over all previous studies. But the results from joint optimization are the best; it is at least 2% (in most case 2.5% or more) higher than the best previous technique – deep subspace clustering.

B. Classification on Benchmark Datasets

Classification will be carried out on three benchmark deep learning datasets – MNIST (60K training and 10K testing), 20-newsgroup and GTZAN (music genre classification). The reason for choosing these diverse datasets is to show that our proposed technique excels over others in a variety of domains – image (MNIST), NLP (20-newsgroup) and audio (GTZAN).

In the 20-newsgroup dataset, the training and test sets contain documents collected at different times, a setting that is more reflective of a practical application. The training set consists of 11,269 samples and the test set contains 7,505 examples. We have used 5000 most frequent words for the binary input features. We follow the same protocol as outlined by [32].

The GTZAN music genre dataset [33] contains 10000 three-second audio clips, equally distributed among 10 musical genres: blues, classical, country, disco, hip-hop, pop, jazz, metal, reggae and rock. Each example in the set is represented by 592 Mel-Phon Coefficient (MPC) features. ThSince there is no predefined standard split and fewer examples, we have used 10-fold cross validation (procedure mentioned in [34]), where each fold consisted of 9000 training examples and 1000 test examples.

In this work, we compare our proposed deep transform learning (DTL) with three state-of-the-art representation learning techniques; two of them are supervised – they are class sparse stacked autoencoder (CSSAE) [35] and class sparse DBN (CSDBN) [36]. The third one is unsupervised; it is deep dictionary learning (DDL) [19]. For all these techniques we found that by reducing the number of nodes in each layer to half that of the previous layer yields the best results consistently. In the CSSAE and the CSDBN formulations one needs specifying the sparsity parameter; for CSSAE a value of 0.1 yields the best results and for CSDBN the corresponding value is 0.02. There is no parameter required for the DDL technique.

Our proposed DTL uses a simple approach for fixing the number of nodes in each layer. It uses a three layer architecture where the number of basis are halved in subsequent layers, i.e. for MNIST since the input of length 784, the number basis in first layer is 392, in the second it is 196 and in the third it is 98. For other datasets, the values change accordingly.

All the techniques compared against, can only learn a representation; they do not have in-built classifiers. Hence, we employ two off-the-shelf classifiers – nearest neighbor (NN) and support vector machine (SVM) with rbf kernel. The parameters of SVM have been tuned via grid search for each technique. The results are shown in Tables III and IV.

TABLE III
RESULTS FOR NEAREST NEIGHBOUR CLASSIFICATION

Dataset	CSSAE	CSDBN	DDL	Greedy DTL	Proposed
MNIST	97.33	97.05	97.75	97.62	**97.91**
20-newsgroup	70.48	70.09	70.48	70.98	**72.64**
GTZAN	83.31	80.99	83.31	83.31	**83.89**

TABLE IV
RESULTS FOR SVM CLASSIFICATION

Dataset	CSSAE	CSDBN	DDL	Greedy DTL	Proposed
MNIST	98.50	98.53	98.64	98.52	**98.71**
20-newsgroup	71.29	71.18	71.97	72.40	**73.19**
GTZAN	83.42	81.83	84.92	83.68	**85.18**

The results show that our proposed method outperforms others. The previous greedy version does not always beat DDL, but our proposed optimized version does. Our proposed deep transform learning results are significantly better than the supervised stacked autoencoder and deep belief network based results.

The results from CNN for GTZAN is 59.20 (this is in-line with the one reported in [36]); for MNIST CNN yields an accuracy of 98.40 with a modified LeNet architecture; for the 20-newsgroup data CNN is not applicable. Our proposed method beats CNN based results for the MNIST and GTZAN.

V. CONCLUSION

This work introduces a new tool for deep learning – deep transform learning. There are three well known pillars of deep learning today – deep belief network, stacked autoencoder and convolutional neural network. A recently proposed technique, deep dictionary learning is also gaining popularity. Our initial work on deep transform learning, shows promise. On benchmark classification and clustering problems we outperform existing deep learning techniques.

This initial work is unsupervised. In future, we will make it supervised problems. We will modify the proposed formulation with supervised penalties like label consistency, logistic regression and soft-max classification. We also plan to introduce sub-space clustering penalties like Experiments will also be carried out on more practical problems.

REFERENCES

[1] S. Ravishankar and Y. Bresler, "Learning Sparsifying Transforms," IEEE Transactions on Signal Processing, Vol. 61 (5), pp. 1072-1086, 2013.

[2] S. Ravishankar, B. Wen and Y. Bresler, "Online Sparsifying Transform Learning-Part I: Algorithms," IEEE Journal of Selected Topics in Signal Processing, Vol. 9 (4), pp. 625-636, 2015.

[3] S. Ravishankar and Y. Bresler, "Online Sparsifying Transform Learning-Part II: Convergence Analysis," IEEE Journal of Selected Topics in Signal Processing, Vol. 9 (4), pp. 637-646, 2015.

[4] S. Ravishankar and Y. Bresler, "Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging", SIAM Journal on Imaging Sciences, Vol. 8 (4), pp. 2519-2557, 2015.

[5] S. Shekhar, V. M. Patel and R. Chellappa, "Analysis sparse coding models for image-based classification," IEEE ICIP, pp. 5207-5211, 2014.

[6] J. Maggu and A. Majumdar, "Robust Transform Learning", IEEE ICASSP, 2017.

[7] J. Maggu and A. Majumdar, "Alternate Formulation for Transform Learning", ICVGIP, 2016.

[8] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization", Nature, Vol. 401 (6755), pp. 788-791, 1999.

[9] 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 (6583), pp. 607, 1996.

[10] M. Aharon, M. Elad and A. Bruckstein, "$rm K$-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," IEEE Transactions on Signal Processing, Vol. 54 (11), pp. 4311-4322, 2006.

[11] F. Cao, M. Cai, Y. Tan and J. Zhao, "Image Super-Resolution via Adaptive $ell _{p} (0<p<1)$ Regularization and Sparse Representation," IEEE Transactions on Neural Networks and Learning Systems, Vol. 27 (7), pp. 1550-1561, 2016.

[12] J. J. Thiagarajan, K. Natesan Ramamurthy and A. Spanias, "Learning Stable Multilevel Dictionaries for Sparse Representations," IEEE Transactions on Neural Networks and Learning Systems, Vol. 26 (9), pp. 1913-1926, 2015.

[13] Z. Li, Z. Lai, Y. Xu, J. Yang and D. Zhang, "A Locality-Constrained and Label Embedding Dictionary Learning Algorithm for Image Classification," IEEE Transactions on Neural Networks and Learning Systems, Vol. 28 (2), pp. 278-293, 2017.

[14] L. Jing, M. K. Ng and T. Zeng, "Dictionary Learning-Based Subspace Structure Identification in Spectral Clustering," IEEE Transactions on Neural Networks and Learning Systems, Vol. 24 (8), pp. 1188-1199, 2013.

[15] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi and R. Chellappa, "Kernel dictionary learning," IEEE ICASSP, pp. 2021-2024, 2012.

[16] A. Shrivastava, V. M. Patel and R. Chellappa, "Multiple Kernel Learning for Sparse Representation-Based Classification," IEEE Transactions on Image Processing, vol. 23 (7), pp. 3013-3024, 2014.

[17] K. K. Huang, D. Q. Dai, C. X. Ren and Z. R. Lai, "Learning Kernel Extended Dictionary for Face Recognition," IEEE Transactions on Neural Networks and Learning Systems, vol. 28 (5), pp. 1082-1094, 2017.

[18] S. Tariyal, A. Majumdar, R. Singh and M. Vatsa, "Deep Dictionary Learning," IEEE Access, Vol. 4, pp. 10096-10109, 2016.

[19] V. Singal and A. Majumdar, "Majorization Minimization Technique for Optimally Solving Deep Dictionary Learning", Neural Processing Letters, 2017, doi:10.1007/s11063-017-9603-9.

[20] V. Singhal, H. Agrawal, S. Tariyal and A. Majumdar, "Discriminative Robust Deep Dictionary Learning for Hyperspectral Image Classification", IEEE Transactions on Geosciences and Remote Sensing, 2017, DOI: 10.1109/TGRS.2017.2704590.

[21] I. Manjani, S. Tariyal, M. Vatsa, R. Singh, A. Majumdar, "Detecting Silicone Mask based Presentation Attack via Deep Dictionary Learning," IEEE Transactions on Information Forensics and Security, Vol. 12 (7), pp. 1713-1723, 2017.

[22] S. Singh and A. Majumdar, "Deep Sparse Coding for Non-Intrusive Load Monitoring", IEEE Transactions on Smart Grid (accepted).

[23] J. Maggu and A. Majumdar, "Greedy Deep Transform Learning", IEEE ICIP, 2017.

[24] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Multiplicative Noise Removal Using Variable Splitting and Constrained Optimization," IEEE Transactions on Image Processing, Vol. 19 (7), pp. 1720-1730, 2010.

[25] C. Wu and X.-C. Tai. "Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models." SIAM Journal on Imaging Sciences, Vol. 3 (3), pp. 300-339, 2010.

[26] D. Jakoveti?, J. M. F. Moura and J. Xavier, "Linear Convergence Rate of a Class of Distributed Augmented Lagrangian Algorithms," IEEE Transactions on Automatic Control, Vol. 60 (4), pp. 922-936, 2015.

[27] A. R. Conn, N. IM Gould and P. Toint. "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds." SIAM Journal on Numerical Analysis Vol. 28 (2), pp. 545-572, 1991.