Noisy Deep Dictionary Learning: Application to Alzheimer's Disease Classification

Vanika Singhal
IIIT Delhi
New Delhi, India
vanikas@iiitd.ac.in

Angshul Majumdar
IIIT Delhi
New Delhi, India
angshul@iiitd.ac.in

***Abstract*— A recent work introduced the concept of deep dictionary learning. In deep dictionary learning, the first level proceeds like standard dictionary learning; in sub-sequent layers the (scaled) output coefficients from the previous layer are used as inputs for dictionary learning. This is an unsupervised deep learning approach. The features from the final / deepest layer are employed for subsequent analysis and classification. The seminal paper of stacked denoising autoencoders have shown that robust deep models can be learnt when noisy data is used for training stacked autoencoders instead of clean data. We adopt this idea into the deep dictionary learning framework; instead of using only clean data we augment the training dataset by adding noise; this improves robustness. Experimental evaluation on benchmark deep learning datasets and real world problem of AD classification show that our proposal yields considerable improvement.**

***Index Terms*— dictionary learning, deep learning, AD classification**

I. Introduction

In deep learning, there are three well known tools - Restricted Boltzmann Machine (RBM) [1], Autoencoder [2] and Convolutional Neural Network. RBM has a probabilistic cost function which learns the weights and the representation by minimizing the Boltzmann cost. Autoencoders learn a decoder and an encoder by minimizing the Euclidean norm of the reconstruction error.

Deep architectures can be built using RBM or autoencoders as basic blocks. Deep Belief Networks (DBN) is built by stacking one RBM after the other and stacked autoencoders are built by nesting one autoencoder inside the other. The features from the last / innermost layer are usually used for classification.

The interpretation of dictionary learning is different (Figure 1). It learns a basis (D) for representing (Z) the data (X) [3]. The columns of D are called 'atoms'. In this work, we look at dictionary learning in a different manner. Instead of interpreting the columns as atoms, we can think of them as connections between the input and the representation layer. To showcase the similarity, we have kept the color scheme intact in Fig. 1.

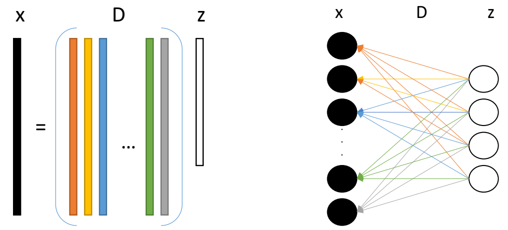

Fig. 1. Dictionary Learning. Left – Conventional Interpretation and Right – Our Interepretation

Unlike a neural network, which is directed from the input to the representation, the dictionary learning can be viewed as a network that points in the reverse direction – from representation to the input. This is what is called 'synthesis dictionary learning' in signal processing. The dictionary is learnt so that the features (along with the dictionary) can synthesize / generate the data. It employs a Euclidean cost function (1), given by

$$\min_{D,Z} \|X - DZ\|_F^2 \quad (1)$$

This formulation was introduced in [3]. Today, most studies impose an additional sparsity constraint on the representation (Z) [4], but it is not mandatory.

Building on the neural network type interpretation, we propose deeper architecture with dictionary learning. An example of two-layer architecture is shown in Figure 2.

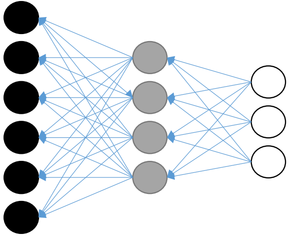

Fig. 2. Deep Dictionary learning

For the first layer, a dictionary is learnt to represent the data. In the second layer, the representation from the first layer acts as input and it learns a second dictionary to represent the

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

features from the first level. This concept can be further extended to deeper layers [5]. This constitutes deep dictionary learning. We learn about it in details later.

Our current work is motivated by the success of stacked denoising autoencoder [6]. In this work instead of learning the mapping between the input and itself, stacked autoencoders were made to learn the encoder and decoder for a noisy version of the input and a clean output, i.e. the input training data was corrupted by noise while its clean version corresponded to the output. This is a stochastic regularization technique which allows a training of more robust models since the stacked autoencoder learns to handle noise.

The same idea is adopted in this work. Instead of only training the deep dictionaries with clean data, we augment the training data with noisy samples. Thus ensuring the learnt dictionaries to be more robust. This kind of training helps in two ways – 1. Augmenting the training data helps combat over-fitting, and 2. Learning from noisy data makes the dictionaries robust. We carry out experiments on the benchmark deep learning datasets as well as on the practical problem of Alzheimer's Disease classification.

II. Noisy Deep Dictionary Learning

In this section, we describe the concept of deep dictionary learning. A single / shallow level of dictionary learning yields a latent representation of data and the dictionary atoms. Deep dictionary learning proposes to learn latent representation of data by learning multi-level dictionaries. The idea of learning deeper levels of dictionaries stems from the success of deep learning. In this section, for ease of understanding, we first explain the concept with two-layer deep dictionary learning and then extend it to multi-level dictionary.

Dictionary learning follows a synthesis framework (1), i.e. the dictionary is learnt such that the features synthesize the data along with the dictionary.

$$X = D_1 Z \tag{2}$$

Deep dictionary learning proposes to extend the shallow dictionary learning into multiple layers – leading to deep dictionary learning. Mathematically, the representation at the second layer can be written as:

$$X = D_1 \varphi(D_2 Z) \tag{3}$$

Learning the dictionaries along with the deepest level features is a hard problem for two reasons:

1) Shallow dictionary learning is a bi-linear. However recent studies such as [7] have proven the convergence of shallow dictionary learning. Learning all the layers in one go is a difficult if not an intractable problem. Extending the convergence proofs over multiple layers, that too with non-linear activations is almost impossible.
2) Moreover, the number of parameters required to be solved increases when multiple layers of dictionaries are learnt simultaneously. With limited training data, this could lead to over-fitting.

Deep dictionary learning proposes to learn the dictionaries in a greedy manner which is in sync with other deep learning techniques [8]. Moreover, layer-wise learning will guarantee the convergence at each layer.

Extending this idea, a multi-level dictionary learning problem with non-linear activation can be expressed as,

$$\min_{z_{test}} \left\| \varphi^{-1}(z_{N-1,test}) - D_N z_{test} \right\|_2^2 + \lambda \left\| z_{test} \right\|_1 \tag{4}$$

A. Training

Ideally, one would have to solve the following problem.

$$\min_{D_1,\ldots D_N,Z} \left\| X - D_1 \varphi\left(D_2 \varphi(\ldots\varphi(D_N Z))\right) \right\|_F^2 + \mu \left\| Z \right\|_1 \tag{5}$$

However, such a problem is highly non-convex and requires solving huge number of parameters. To address these issues, as mentioned before, a greedy approach is proposed where one layer is learnt at a time. With the substitution $Z_1 = \varphi\left(D_2 \varphi(\ldots\varphi(D_N Z))\right)$, Equation (5) can be written as as $X = D_1 Z_1$ such that it can be solved as single layer dictionary learning.

$$\min_{D_1,Z_1} \left\| X - D_1 Z_1 \right\|_F^2 \tag{6}$$

For the second layer, one substitutes $Z_2 = \varphi(D_3 \ldots \varphi(D_N Z))$, which leads to $Z_1 = \varphi(D_2 Z_2)$, or alternately, $\varphi^{-1}(Z_1) = D_2 Z_2$; this too is a single layer dictionary learning.

$$\min_{D_2,Z_2} \left\| \varphi^{-1}(Z_1) - D_2 Z_2 \right\|_F^2 \tag{7}$$

Continuing in this fashion till the penultimate layer, in the final layer one has $Z_{N-1} = \varphi(D_N Z)$ or $\varphi^{-1}(Z_{N-1}) = D_N Z$. In the last level the coefficient Z can be sparse. For learning sparse features, one needs to regularize by applying l_1-norm on the features. This is given by:

$$\min_{D_N,Z} \left\| \varphi^{-1}(Z_{N-1}) - D_N Z \right\|_F^2 + \lambda \left\| Z \right\|_1 \tag{8}$$

Here we have shown how the deep dictionary learning problem can be greedily segregated into shallow dictionary learning problems. Solving the shallow dictionary learning problem is a well researched area so we do not get into those details.

B. Testing.

For testing the learnt dictionaries are used to generate the code for the test sample.

$$\min_{z_{test}} \left\| x_{test} - D_1 \varphi\left(D_2 \varphi(\ldots\varphi(D_N z_{test}))\right) \right\|_2^2 + \lambda \left\| z_{test} \right\|_1 \tag{9}$$

One does not solve (9) in one go; rather in a staggered fashion one layer at a time similar to training, i.e. in the first stage one solves for the first level of coefficients using the substitution in (6).

$$\min_{z_{1,test}} \left\| x_{test} - D_1 z_{1,test} \right\|_2^2 \quad (10)$$

For the second level one would require solving,

$$\min_{z_{2,test}} \left\| \varphi^{-1}(z_{1,test}) - D_2 z_{2,test} \right\|_2^2 \quad (11)$$

This continues till the penultimate level; in the final level we need to impose sparsity constraint. This requires solving (12) for the final test feature.

$$\min_{z_{test}} \left\| \varphi^{-1}(z_{N-1,test}) - D_N z_{test} \right\|_2^2 + \lambda \left\| z_{test} \right\|_1 \quad (12)$$

C. Noisy Learning

Our work is motivated from stacked denoising autoencoders; which in turn are based on denoising autoencoders. Regular autoencoders learn a mapping (reconstruction from the input to itself; it learns to preserve information in the Euclidean sense. Denoising autoencoders follow a stochastic regularization technique. They corrupt the input with noise, so that the autoencoder learns to reconstruct a clean signal from a noisy version of the input; this makes the autoencoder more robust. Stacked denoising autoencoders are deeper versions of such denoising autoencoders.

There are several other well known stochastic regularization techniques – DropOut [9] and DropConnect [10]. In DropOut, some of the nodes are randomly dropped (forced to zero); this prevents the nodes from co-adapting. On one hand it prevents over-fitting; on the other it enforces the nodes to act independently. In DropConnect, the nodes are not dropped, the connections are. This enforces the connections to learn independently preventing co-adaptation.

In this we follow the former regularization technique. We simply augment the training data by adding noisy samples and input it for training deep dictionary learning. We postulate that training from such corrupted samples would increase the robustness of the learnt dictionaries which would result in better classification.

III. Experimental Evaluation

A. Benchmark Experiments

We have carried the experiments on benchmark datasets. The first one is the MNIST. The MNIST digit classification task is composed of 28x28 images of the 10 handwritten digits. There are 60,000 training images with 10,000 test images in this benchmark. No preprocessing has been done on this dataset.

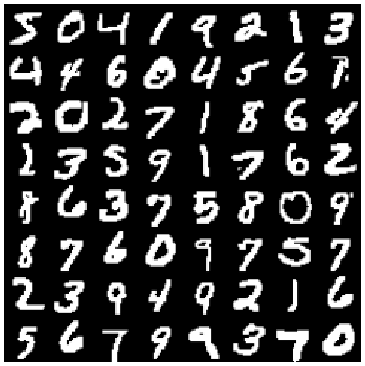

Fig. 3. MNIST Samples

The CIFAR-10 dataset is composed of 10 classes of natural images with 50,000 training examples in total, 5,000 per class. Each image is an RGB image of size 32x32 taken from the tiny images dataset and labeled by hand. These images need to be preprocessed. We follow the standard preprocessing technique – the RGB is converted to YUV and the Y channel is used. Before putting it for training, mean subtraction and global contrast normalization is done.

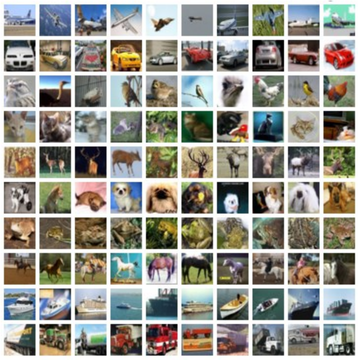

Fig. 4. CIFAR-10 Samples

The Street View House Numbers (SVHN) dataset is composed of 604,388 images (using both the difficult training set and simpler extra set) and 26,032 test images. The goal of this task is to classify the digit in the center of each cropped 32x32 color image. This is a difficult real world problem since multiple digits may be visible within each image. We preprocessed these samples in the same way as CIFAR.

Fig. 5. SVHN Samples

The benchmarking was carried out with state-of-the-art deep learning tools – Deep Belief Network (DBN), Stacked Denoising Autoencoder (SDAE), Deep Dictionary Learning (DDL) and Convolutional Neural Network (CNN). For each dataset 5 fold cross validation on the training set was done to tune the parameters for noisy deep dictionary learning. For each dataset we had a noise combination – MNIST (Gaussian 0.25 standard deviation + 15% Impulse); CIFAR (Gaussian 0.25 standard deviation + 20% Impulse); SVHN (Gaussian 0.1 standard deviation + 15% Impulse). We did not try other types of noise such as speckle noise or uniform noise.

For this study, compared against, pre-trained models for the benchmark problems are available. For DBN, SDAE, DDL and our proposed method, a neural network was used for classification; for all these a three layer architecture is used where the number of nodes are halved in every subsequent level. Our method uses a simple Nearest Neighbour classifier.

Results on benchmark datasets indicate that our method is superior to DBN, SDAE, DDL and CNN overall; it is only slightly worse than CNN for the SVHN. This is because SVHN has a large volume of training data and CNN excels in such situations; for realistic problems with limited training data our method outperforms.

TABLE I. COMPARATIVE CLASSIFICATION ACCURACY

Dataset	Proposed	DBN	SDAE	DDL	CNN
MNIST	**99.21**	98.78	98.72	98.05	99.06
CIFAR-10	**84.22**	78.90	74.30	82.32	83.40
SVHN	94.12	92.60	89.70	92.64	**94.97**

TABLE II. CLASSIFICATION ACCURACY FOR AD

Tasks	Robust Deep Learning [11]	Deep Dictionary Learning [5]	Proposed
AD vs HC	91.4, ±1.8	92.1, ±2.8	**95.4, ±2.0**
MCI vs HC	77.3, ±1.8	78.2, ±2.1	**85.7, ±2.0**
MCI.C vs MCI.N	57.5, ±3.4	59.0, ±3.4	**64.1, ±2.1**
Average	75.4	76.4	**81.7**

B. Alzheimer's Disease (AD) Classification

In the last sub-section we have shown comprehensively that our proposed techniques work at par or even better than existing deep learning tools. In this sub-section we apply our proposed technique on a real world problem of AD classification.

One of the most recent studies on using deep learning for AD classification is [11]; we follow the protocol defined therein. The well-known ADNI data set [12] is used for the experiments. The data set consists of MRI, PET, and CSF data from 51 AD patients, 99 MCI (Mild Cognitive Impairment) patients (43 MCI patients who converted to AD (MCI.C), and 56 MCI patients who did not progress to AD in 18 months (MCI.NC)) alongwith 52 healthy normal controls (HC). In addition to the crisp diagnostic result (AD or MCI), this data set contains two additional clinical scores, MMSE and ADAS-Cog for each patient. Following [11] we extracted image processing based features from the 3D MRI and PET volumes – the feature extraction scheme follows from [13]-[15] Finally, we extracted 93 features from MRI and PET volume, respectively, and three CSF biomarkers, $A\beta_{42}$,t−tau, and p-tau were computed, resulting in 189 features for each subject.

We consider three classification tasks – 1) AD patients vs Healthy Control subjects (AD vs HC), 2) MCI patients vs HC (MCI vs HC) and 3) MCI-converted vs MCI-non converted (MCI.C vs MCI.N). For each task we followed the protocol outlined in [11] – 10 fold cross validation was performed for evaluation. The data partitioning was done randomly; in order to improve robustness and reliability division into 10 folds were done 10 times and the average overall accuracies were reported.

We benchmark against the technique proposed in [11]. This is one of the most recent and comprehensive studies on this topic and have shown to outperform other methods. CNN and SDAE have been also used for this problem, but the said technique outperforms them; therefore we only compare with [11]. The interested reader can peruse the aforesaid reference to get the details; we skip it for the sake of brevity.

In this work we used a three level architecture 100-50-25. The input features were converted to unit norm – this is not a requirement for dictionary learning but helps in generating the noise. To the input features a combination of Gaussian (0 mean and standard deviation 0.2), Impulse (10% corrupted features) and Uniform noise (between 0 and 0.05) is added. The training dataset was augmented 5 times by adding noise; the clean data was also used alongwith the noisy data for training. The value of $\lambda = 0.2$ is used throughout for deep dictionary learning. Since these are binary classification problems, an SVM classifier is suitable. We use one with an rbf kernel.

We see that deep dictionary learning only improves the results slightly, compared to the robust deep learning [11]. But with our proposed noisy deep dictionary learning method the results improve vastly; one does not require any statistical test to judge the improvement – it is apparent.

IV. Conclusion

Recently a new tool for deep learning has been proposed – deep dictionary learning. Here, instead of expressing the data in terms of a single basis / dictionary, it is represented in terms of multiple levels of dictionaries. In [5] it was shown that deep dictionary learning performs at par or better than other deep learning tools such as deep belief network (DBN), stacked denoising autoencoder (SDAE) or convolutional neural network (CNN).

In this work we propose a stochastic regularization technique for deep dictionary learning. Instead of learning the dictionaries from the clean samples, we learn dictionaries from noisy samples; this idea stems from the success of SDAE. Here we have shown that such a scheme indeed improves the results even further. We have carried out experiments on becnhmark datasets as well as on a real problem of AD classification.

In the recent past, the basic version of deep dictionary learning has been used for a variety of problems [16-20]. We expect that, in future the proposed stochastic regularization technique will benefit these areas and more.

Acknowledgement

We are thankful to the Infosys Center for Artificial Intelligence @ IIITD for partial support.

References

[1] Salakhutdinov, R., Mnih, A. and Hinton, G. 2007. Restricted Boltzmann machines for collaborative filtering. ICML.

[2] Baldi, P. 2012. Autoencoders, Unsupervised Learning, and Deep Architectures. Workshop on Unsupervised and Transfer Learning.

[3] Engan, K., Aase, S., O. and Hakon Husoy, J. 1999. Method of optimal directions for frame design. IEEE ICASSP,

[4] Rubinstein, R., Bruckstein, A. M. and Elad, M. 2010. Dictionaries for Sparse Representation Modeling. Proceedings of the IEEE, 98, 6, pp. 1045-1057.

[5] Tariyal, S., Majumdar, A., Singh, R. and Vatsa, M. 2016. Deep Dictionary Learning. IEEE Access, 2016.

[6] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P. A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11, 3371-3408.

[7] Jain, P., Netrapalli, P. and Sanghavi, S. 2013. Low-rank Matrix Completion using Alternating Minimization. STOC.

[8] Bengio, Y. 2009. Learning deep architectures for AI. Foundations and Trends in Machine Learning, 1, 2, 1-127, 2009.

[9] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 1929-1958.

[10] Wan, L., Zeiler, M., Zhang, S., LeCun, Y. and Fergus, R. 2013. Regularization of Neural Network using DropConnect. ICML.

[11] Li. F., Tran. L., Thung. K. H., Ji. S., Shen. D. and Li. J. 2015. A Robust Deep Model for Improved Classification of AD/MCI Patients. IEEE Journal of Biomedical Health Informatics. 19, 5,1610-6.

[12] http://www.loni.ucla.edu/ADNI

[13] Davatzikos, C., Bhatt, P., Shaw, L.M., Batmanghelich, K.N., Trojanowski, J.Q. 2011. Prediction of MCI to AD conversion, via MRI, CSF biomarkers, and pattern classification. Neurobiology of Aging. 27, 2322, 19-27.

[14] Kabani, N., MacDonald, D., Holmes, C., Evans, A. 1998. A 3D atlas of the human brain. NeuroImage 7(4), S717.

[15] Hinrichs, C., Singh, V., Xu, G. and Johnson, S.C. 2011. Predictive markers for AD in a multi-modality framework: An analysis of MCI progression in the ADNI population. NeuroImage. 55, (2), 574-589.

[16] V. Singal and A. Majumdar, "Majorization Minimization Technique for Optimally Solving Deep Dictionary Learning", Neural Processing Letters, (accepted).

[17] 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 (accepted).

[18] V. Singhal and A. Majumdar, "Noisy Deep Dictionary Learning", CODS 2017.

[19] V. Singhal, S. Singh and A. Majumdar, "How to Train Your Deep Neural Network with Dictionary Learning", Data Compression Conference, 2017.

[20] V. Singhal, A. Gogna and A. Majumdar, “Deep Dictionary Learning vs Deep Belief Network vs Stacked Autoencoder: An Empirical Analysis”, ICONIP, pp. 337-344 2016