A Deep Learning Approach to Fetal-ECG Signal Reconstruction

Priya Ranjan Muduli, Rakesh Reddy Gunukula and Anirban Mukherjee

***Abstract*—Fetal electrocardiogram (FECG) monitoring has become essential due to the current increase in the relative number of cardiac patients worldwide. This paper proposes to use a deep learning approach to compress/recover FECG signals, improving the computation speed in a telemonitoring system. The problem is analogous to the reconstruction of a non-sparse signal in compressive sensing (CS) framework. The architecture incorporates a non-linear mapping using a stacked denoising autoencoder (SDAE). The compression of the raw non-sparse FECG data takes place at the transmitter side using a deep neural network. After pre-training, the whole deep SDAE can be further fine tuned by the mini-batch gradient descent-based back-propagation algorithm. Although the training for SDAE is usually time-consuming, it does not affect the performance due to the one-time off-line training process. The real-time FECG reconstruction is faster due to a few matrix-vector multiplications at the receiver end. The simulations performed by employing standard non-invasive FECG databases shows promising results in terms of the reconstruction quality.**

I. INTRODUCTION

Congenital heart defect is one of the leading causes of birth defect-related death. In this regard, the wireless fetal ECG (FECG) monitoring systems enhance the patient mobility, and the reduction in health-care costs by transmitting the data over a wireless network. Reconstruction of noninvasive FECG is principally influenced by the fetal brain activity, myographic artifacts of both the mother and the fetus. There is an increased interest in Point-of-Care (PoC) based telemonitoring. The PoC devices at the receiver's end are usually battery driven and should be operated at minimal computational complexity. The reconstruction of FECGs is still in its infancy due to the lack of gold-standard databases, partly due to the fusion of the fetal ECG with the maternal ECG. In an ambulatory environment, there is a high demand for robust compression as well as reconstruction techniques. The lossy compression techniques have gained popularity due to its higher compression while retaining an acceptable data quality, which further minimizes the power consumption. Compressed sensing (CS) is applied for developing energy efficient FECG telemonitoring system [1]. Many sparsifying domains, such as Discrete Fourier Transform, Discrete Cosine Transform, Shearlet Transform and Wavelet Transform have been traditionally used for data compression. In [2], the wavelet-based compression is replaced by CS for building energy efficient sensors. The CS-based reconstruction algorithms are usually slow and need to solve a non-smooth optimization problem in an iterative fashion. In each iteration, CS performs a non-linear thresholding operation to reconstruct the source signal in a probabilistic manner. CS techniques might not be a suitable choice in case of real-time signal reconstruction of non-sparse signals. Deep learning technique mitigates the trade-off between the reconstruction speed and reconstruction quality showing potential application in speech feature extraction [3], object recognition [4], natural language processing [5], information retrieval [6], [7], and collaborative filtering [8]. For the efficient learning of statistical features, a deep network (with multiple hidden layers) is preferred [9]. In the literature various state-of-the-art deep neural network (DNN) architectures such as Auto-Encoders (AE) [10], Denoising Auto-Encoders (DAE) [11], Stacked Auto-Encoders (SAE) [12], Convolutional Neural Networks (CNN) [13] are reported. A Restricted Boltzmann Machine (RBM), in which the neurons are binary nodes of a bipartite graph is defined in [14]. Also, for adaptive learning, several dictionary learning algorithms such as the method of optimal directions (MOD) [15], which involves the data matrix-inversion at each step; and the K-SVD [16], which involves computing the SVD to update the dictionary, are being reported. The deep learning is incorporated for dimensionality reduction [17], [18]. Inspired by the success of deep learning techniques, it is proposed to incorporate an SDAE-based deep learning approach for the reconstruction of multichannel FECG signals from a set of undersampled measurements. The architecture employs a Multiple Measurement Vector (MMV) model that is analogous to the CS-based MMV model. An unsupervised pretraining using mini batch gradient descent algorithm is employed in this work. The simulation results using multichannel non-invasive FECG recordings from Physionet database [19] shows promising reconstruction results. The rest of the paper is organized as follows. Section II addresses the deep learning network architecture, used to reconstruct a structured signal. Section III illustrates the application of SDAE technique in FECG signal reconstruction. Finally, the concluding notes are presented in the Section IV.

Priya Ranjan Muduli is with the Department of Electrical Engineering, Indian Institute of Technology Kharagpur, WB, 721302, India priyaranjan@ee.iitkgp.ernet.in

Rakesh Reddy Gunukula is with the Department of Electrical Engineering, Indian Institute of Technology Kharagpur, WB, 721302, India rakeshreddy.gunukula@gmail.com

Anirban Mukherjee is with the Department of Electrical Engineering, Indian Institute of Technology Kharagpur, WB, 721302, India anirban@ee.iitkgp.ernet.in

978-1-5090-2361-5/16/$31.00

II. NOTATION AND PROBLEM FORMULATION

A. Notations

The following notations are used throughout this paper:

- y_i is the i^{th} training example. $f(\cdot)$ represents the activation function. $W^{(l)}$ denotes the weight matrix associated between layer l and layer $l+1$.

- $vec(X) \in R^{NL}$ specifies the row-wise vectorization operation over $X \in R^{N \times L}$.
- $\|X\|_0$ denotes the sparsity level in $X \in R^{N \times L}$.

B. Deep Learning Architectures

In an MMV framework, the generalized inverse problem estimates the source signal, $X \in R^{N \times L}$ from an undersampled measurement, $Y \in R^{m \times L}$ using a suitable sensing matrix, $\Phi \in R^{m \times N}$, $m \ll N$. The problem is formulated as follows:

$$Y = \Phi X + V$$

where $X = [x_1, x_2, \cdots, x_L] \in \mathrm{R}^{N \times L}$ is the unknown source matrix. $Y = [y_1, y_2, \cdots, y_L] \in \mathrm{R}^{m \times L}$ is the measurement matrix, $\Phi \in \mathrm{R}^{m \times N}$, $m \ll N$ is the sensing matrix and $V \in \mathrm{R}^{m \times L}$ is an unknown noise matrix. For $L=1$, the formulation is known as single measurement vector (SMV) problem, while, for $L>1$, it transforms into an MMV model. In the CS framework, the inverse problem for the sparsest signal estimate can be represented as,

$$\hat{X} = \underset{X}{\operatorname{argmin}} \ \|X\|_o \ \text{ such that } Y = \Phi X$$

It is critical to design a suitable source adaptive sensing matrix and to reconstruct the signals which are not sparse in either of the spatial and temporal transform domains. Solving an underdetermined inverse problem, involving a non-sparse signal is one of the most challenging tasks. Learning-based algorithms might be a solution to these problems. Recently, a Bound Optimization-based Block Sparse Bayesian Learning algorithm (BSBL-BO) is proposed in CS framework [1]. Deep learning may be an alternative learning-based framework which learns the features of the original data from higher levels of the hierarchy using the lower level features. Learning the features from multiple levels makes the system adaptive and robust towards reconstructing the source data. Hinton [20] showed that training the layers of deep architecture individually in an unsupervised manner can be more useful than the supervised training. Feedforward Artificial Neural Network (ANN) with many hidden layers can be considered as a standardized deep network. However, there exists various challenges while training the ANN. Some of these are reported here:

- **Nonconvexity**: In general the ANN parameters are updated by solving a non-convex optimization problem, which might not offer the global solution.
- **Hyperparameter selection**: There exists no theoretical justification for selecting suitable parameters, such as batch-size, learning rate, the number of hidden units and layers while training the ANN [14].
- **Vanishing Gradient**: The backward flowing error signal may get vanished as the error propagates through deep layers in the back-propagation technique [21].

1) Autoencoder: An autoencoder (AE) is a feature learning-based three layered ANN that reconstructs its unlabeled input signals. It takes a normalized vector, $x_i = [x_i(j)] \in R^n$ as input with $0 \leq x_i(j) \leq 1$, encodes it to a hidden layer, $y_i \in R^m$; and then attempts to estimate the source signal, $\hat{x}_i \in R^n$. A bias vector, $b \in R^m$ is added to hidden layers and output layer, which allows better learning of the model. Fig. 1 illustrates the architecture of a basic AE. The functional values for the hidden

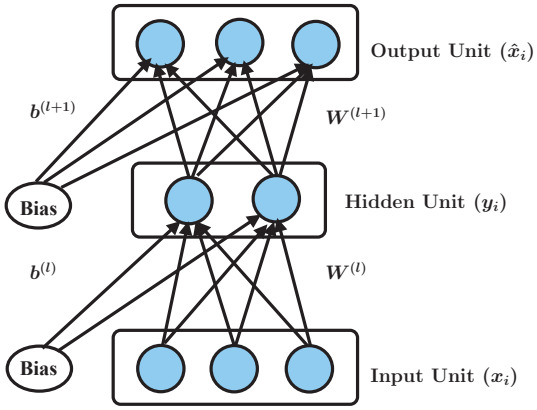


Fig. 1: An AE model with three input, three output and two hidden units.

layer units and output units are given by, $y_i = f(W^{(l)} x_i + b^{(l)})$ and $\hat{x}_i = f(W^{(l+1)} y_i + b^{(l+1)})$, where $W^{(l)} \in R^{m \times n}$ is the weight matrix (analysis operator). The bias vectors are $b^{(l)} \in R^m$ and $b^{(l+1)} \in R^n$. Depending upon the measurement paradigm, the structure of the AE can be linear or non-linear. For normalized inputs, one can choose non-linear activations, such as sigmoid or hyperbolic tangent functions based on the range of the normalized inputs, and modify the loss function accordingly. The sigmoid and hyperbolic tangent activation functions are represented by $f_s(x_i) = \frac{1}{1+\exp(-x_i)}$, and $f_h(x_i) = \frac{\exp(x_i)-\exp(-x_i)}{\exp(x_i)+\exp(-x_i)}$ respectively with $x_i \in R$. An AE using a non-linear hidden layer and a linear (identity activation function) output layer is a linear decoder. The AE utilizes the synthesis operator, $W^{(l+1)} = {W^{(l)}}^T \in R^{n \times m}$ in order to reconstruct the source data, $x_i \in R^n$, where n is the block size. The hyperparameter sets for the hidden layer units y_i and output units $\hat{x}_i$ are represented by $\Omega_1 = [vec(W^{(l)})^T, \ vec(b^{(l)})^T]^T$ and $\Omega_2 = [vec(W^{(l+1)})^T, \ vec(b^{(l+1)})^T]^T$ respectively. The hyperparameters are learned by minimizing the Euclidean distance-based loss function

$$L(x_i, \hat{x}_i) = \frac{1}{K} \sum_{i=1}^{K} \|x_i - \hat{x}_i\|_2^2 \tag{1}$$

where K is the total number of training samples.

The optimization problem in (1) incorporates a non-linear smooth and continuously differentiable activation $f(\cdot)$. Hence it can be solved by using gradient descent methods. For binary-valued vectors as input to AE, the cross-entropy loss function, $L(x_i, \hat{x}_i) = x_i^T \log_2(\hat{x}_i) + (1 - x_i)^T \log_2(1 - \hat{x}_i)$ can be considered. For compression, the constraint, $m \leq n$ is assumed. Although the sparsity is not directly related to the deep learning architecture, using suitable basis some interesting features can be found by exploiting the sparsity of the data. The loss function for a sparse AE can be devised in terms of

the diversity (non-sparsity) measure:

$$L(x_i, y_i) = \frac{1}{K}\sum_{i=1}^{K}\left(\|x_i - \hat{x}_i\|_2^2 + \gamma\|y_i\|_1\right)$$

where the parameter $\gamma \in R$ defines the strength of the penalty imposed on non-sparsity. The regularization of the loss function avoids the over-fitting of learning algorithms in ANN. The l_0-norm-based regularization is nonconvex and NP-hard. The l_1 regularization attempts to develop a sparse model, allowing many hidden activations ($y_i(j) \in R$) to be zero, i.e., the corresponding attributes become irrelevant. A dropout regularization technique can be employed to thin an ANN, by dropping out some of its units. It also provides a way to avoid overfitting of ANN [22].

2) Denoising Autoencoder: The denoising AEs (DAE) have a similar structure as that of the basic AEs except the denoising process [11]. The input is corrupted partially on a random proportion of the input, $x_i \in R^n$ either by zeroing a small fraction of the total data or by adding white Gaussian noise to it. The AE is trained to reproduce x_i from the noisy data, $\tilde{x}_i \in R^n$ by minimizing a suitable loss function, such as $L(x_i, \tilde{x}_i) = \frac{1}{K}\sum_{i=1}^{K}\{\|x_i - \tilde{x}_i\|_2^2\}$.

3) Stacked Denoising Autoencoder: Stacked DAE (SDAE) is the cascaded version of DAEs, forming a deep network structure. First, an AE is trained on the input, $x_i \in R^n$. In this case, the unsupervised pre-training is performed layer by layer. The trained hidden layers are employed as input to train the next AE. Hence, a series of AEs become trained sequentially. The SDAE architecture employed in this work is depicted in the Fig. 2. This SDAE structure has a close

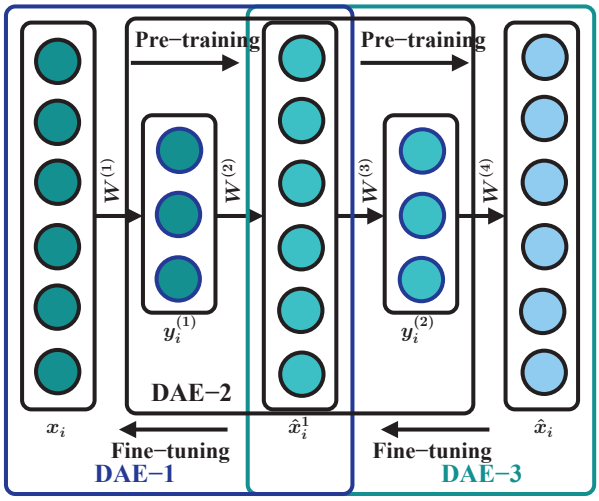


Fig. 2: SDAE architecture for recovering the data.

resemblance with the CS framework. The hidden layer units, $y_i^{(\cdot)}$s, of SDAE may be viewed as a compression action as of the CS paradigm. The first layer of the SDAE takes $x_i \in R^n$ as input, produces the measurement, $y_i^{(1)} = f_s(W^{(1)}x_i + b^{(1)}) \in R^m$. For the compressed measurement case, $m \leq n$ is considered. The compression type in pre-training of SDAE depends upon the weight matrix, $W^{(1)} \in R^{m\times n}$ and the sigmoid activation function $f_s(\cdot)$. To integrate with the CS framework, one can consider the input, $x_i \in R^n$ in a sparse domain, and a linear projection matrix, $W^{(1)} \in R^{m\times n}$ is employed along with the activation function, $f(z) = z$ for $z \in R$. For the second layer of SDAE, $y_i^{(1)}$ becomes the input units, and responses from the subsequent stages after pre-training operation is given by,

$$\begin{aligned}\hat{x}_i^{(1)} &= f_s\left(W^{(2)}y_i^{(1)} + b^{(2)}\right)\\ y_i^{(2)} &= f_s\left(W^{(3)}\hat{x}_i^{(1)} + b^{(3)}\right)\\ \hat{x}_i &= f_s\left(W^{(4)}y_i^{(2)} + b^{(4)}\right)\end{aligned}$$

An unsupervised layer-wise pre-training scheme is incorporated in SDAE followed by a supervised fine-tuning using back-propagation (BP). The BP technique may have slow convergence in case of a deep network, which is typically having a non-convex, non-quadratic, high dimensional cost function. Each individual DAE from the deep network tunes its hyperparameters using (1). The hyperparametric model to be optimized can be represented by $\Omega = \left[vec(W^{(1)})^T \cdots vec(W^{(l)})^T \; b^{(1)^T} \cdots \; b^{(l)^T}\right]^T$, where $W^{(1)}, W^{(3)} \in R^{m\times n}$ and $W^{(2)}, W^{(4)} \in R^{n\times m}$. To achieve optimum hyperparameters with a faster convergence rate, several learning techniques, such as batch learning (BL), stochastic learning (SL) play a vital role during the training process. The theoretical analysis of the weight dynamics of BL technique is well proven. However, the SL technique is often preferred to the BL method, since it is faster, and also results in better solutions. An efficient learning algorithm known as the mini-batch algorithm, which combines both the SL and BL, is devised in [14]. The mini-batch algorithm starts with a small batch size, and the size increases with the progress of training. This updates the hyperparameters for batches containing a fixed size of training sets. For a large size of batches, the performance of the algorithm is close to gradient descent methods. For the given training size, n, and the learning rate, μ, the parameters are updated according to the following equation,

$$\Omega_k \leftarrow \Omega_k - \mu\frac{\partial}{\partial\Omega_k}J(\Omega, (x_i^{(l)}, y_i^{(l)}), \cdots, (x_i^{(l+\zeta)}, y_i^{(l+\zeta)})) \;\; \forall\, k \quad (2)$$

where ζ is the batch size. The update task continues until the convergence.

III. Simulation Results

The detail implementation steps for FECG reconstruction are described here along with the performance analysis.

A. Implementation

For the experiment, multi-channel non-invasive fetal-ECG signals are considered from Physionet database [19]. The raw FECG signal may not be sparse in its natural domain; hence most of the CS algorithms fail to reconstruct the signals. Anyway it can be reconstructed by using a suitable sparsifying transformation. However, the SDAE-based reconstruction

method is independent of the structure of the signal. As the channel length of FECG signals increases, the computational complexity of training the deep network also increases proportionally. To learn all the possible patterns present in the data, an overlapping training scheme is employed taking consecutive overlapping data as training examples. The overlapping training scheme is implemented for pre-training the deep network and the non-overlapping scheme is used for reconstruction.

TABLE I: 100-run average reconstruction time of SDAE and BSBL-BO algorithms on FECG data, from Physionet and DAISY databases at different CRs.

CR(%)	**Physionet data**		**DAISY Data**	
	SDAE (ms)	BSBL-BO (s)	SDAE (ms)	BSBL-BO (s)
25	**0.0859**	1.0162	**0.0821**	1.2198
37.5	**0.0823**	1.0105	**0.0791**	1.2192
50	**0.0221**	1.0096	**0.0771**	1.1859
62.5	**0.0194**	1.0051	**0.0750**	1.0875
75	**0.0171**	1.0002	**0.0542**	1.0853
87.5	**0.0142**	0.9993	**0.0311**	0.9538
93.75	**0.0121**	0.9808	**0.0178**	0.8583

TABLE II: Performance comparison among state-of-the-art algorithms on FECG data, collected from Physionet database (patient id: ecgca154) by varying the compression ratio.

CR(%)	**SNR**		**PCC**		**PRD**	
	SDAE	BSBL-BO	SDAE	BSBL-BO	SDAE	BSBL-BO
25	37.69	**39.71**	0.98	**0.99**	1.66	**0.81**
37.5	36.73	**38.67**	0.98	**0.99**	1.78	**1.05**
50	**35.39**	34.47	0.98	0.98	**1.70**	1.89
62.5	**32.40**	31.14	**0.97**	0.96	**2.33**	2.81
75	**26.78**	26.07	**0.89**	0.87	**4.01**	4.75
87.5	**23.56**	4.09	**0.93**	0.02	**6.65**	30.91
93.75	**22.29**	2.17	**0.61**	0.005	**7.71**	59.22

TABLE III: Performance comparison among state-of-the-art algorithms on FECG data, using DAISY database by varying the compression ratio.

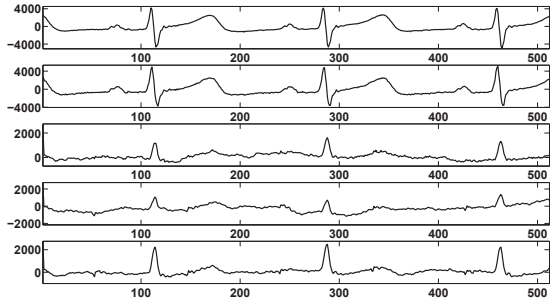
CR(%)	**SNR**		**PCC**		**PRD**	
	SDAE	BSBL-BO	SDAE	BSBL-BO	SDAE	BSBL-BO
25	55.39	**59.49**	0.98	**0.99**	0.16	**0.11**
37.5	54.29	**57.91**	0.97	**0.99**	0.19	**0.13**
50	**52.12**	51.30	0.97	0.97	**0.21**	0.22
62.5	**45.78**	38.16	**0.83**	0.68	**0.59**	1.22
75	**39.36**	2.71	**0.79**	0.03	**2.45**	122
87.5	**31.90**	0.59	**0.71**	0.01	**4.65**	279
93.75	**21.99**	0.43	**0.51**	0.01	**8.59**	301

The SDAE architecture is operated on raw FECG signals, collected from abdominal non-invasive FECG database [19]. The recordings are taken from individual subjects between 21 to 40 weeks of pregnancy. The raw FECG signals are sampled at a rate of 1kHz. For the simulation purpose, two thoracic signals and three abdominal signals are considered. Using a Hamming window-based Finite Impulse Response (FIR) filter the signals are decimated to a sampling rate of 250 Hz, which is common in tele-monitoring applications [23]. A data duration of 128s is considered from total 54 subjects (40 for training and 14 for testing), each having 5 channels at the sampling frequency of 250 Hz. The raw multichannel FECG signals, $X \in R^{N \times L}$ are vectorized to generate $vec(X) \in R^{NL}$ with $N=128 \times 250=32000$ and $L=5$. To reduce the computational complexity during the pre-training of ANN, a fixed block based training scheme is considered with block size, n. The pre-training is performed using overlapped successive data segments, $x_i \in R^n$ as training set. This helps the AE to learn the complex structures present in the dataset. For this experiment, the block size is chosen as $n=64$. After training, the weight matrices, $W^{(1)}$, $W^{(2)}$, and $W^{(3)}$ are stacked in a non-linear fashion to act as the sensing matrix at the transmitter side. During reconstruction, the compressed measurement, $y_i^{(2)} \in R^m$ (at the transmitter) is given by,

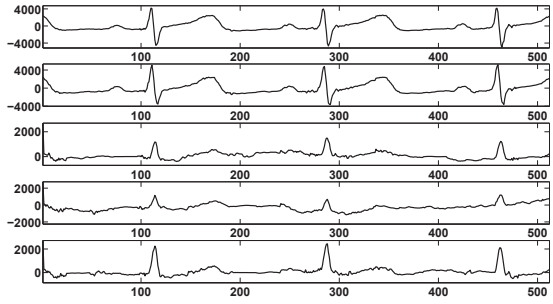
$$y_i^{(2)} = f_s\left(W^{(3)} f_s\left(W^{(2)} f_s\left(W^{(1)} x_i + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right)$$

where $b^{(i)}$ is the bias vector associated with i^{th} layer. Assuming B number of blocks (each of size $n=64$) in the test data, the composite compressed measurement, $y^{(2)} = \left[y_1^{(2)}\ y_2^{(2)} \cdots y_B^{(2)}\right] \in R^{mB}$, is computed using the sensing matrix and is transmitted to the receiver. After training, the weight matrix, $W^{(4)}$, and the bias vector, $b^{(4)}$, are pre-stored at the receiver side for reconstruction. The sigmoid nonlinear operation, $f_s(\cdot)$, is performed on $W^{(4)} y^{(2)} + b^{(4)}$ at the receiver to reconstruct the FECG signal $\hat{x} \in R^{nB}$. The percentage of compression is computed as: $CR(\%) = \frac{nB-mB}{nB} \times 100$. For a fixed block size n, the value of m varies according to the prescribed CR.

To the best of the authors' knowledge, the block sparse Bayesian learning (BSBL-BO) framework [1] can reconstruct the signal irrespective of its sparsity. Therefore, the BSBL-BO is considered for the performance comparison. During the simulation, it is observed that the FECG extraction using BSBL-BO algorithm outperforms the SDAE framework at low CR (less than 50%). However, at higher level of CR (greater than 50%) the SDAE-based reconstruction technique is superior. The signal to noise ratio (SNR), percentage root mean-square difference (PRD) and Pearson's correlation coefficient (PCC) are considered as the the performance indices. The SNR is defined as $20\log_{10}(||x||_2/||x-\hat{x}||_2)$, where $x \in R^{nB}$, $\hat{x} \in R^{nB}$ are the original and the reconstructed signal respectively. The PRD is computed as $PRD = \left(\|x-\hat{x}\|_2 / \|x\|_2\right) \times 100$. The PCC is computed as $\rho_{x,\hat{x}} = \frac{cov(x,\hat{x})}{\sigma_x \sigma_{\hat{x}}}$, where $\text{cov}(\cdot,\cdot)$ is co-variance operator, and σ_x, $\sigma_{\hat{x}}$ are the standard deviations of $x \in R^{nB}$, $\hat{x} \in R^{nB}$ respectively. For the visual evaluation, the original and reconstructed signals (Physiobank database record: $ecgca154edfm$) are depicted in Fig. 3a and Fig. 3b. Furthermore, a post-processing stage is employed over the reconstructed FECG signals. In these experiments, both the original and the reconstructed FECG signals are processed using a bandpass filter of normalized passband frequency, $[0.008, 0.4]$, followed by a whitening process and the independent component analysis (ICA) operation. The

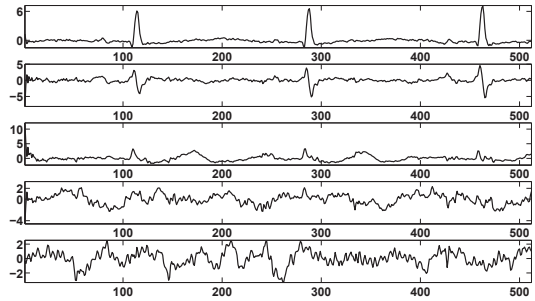


(a) Original FECG recordings.

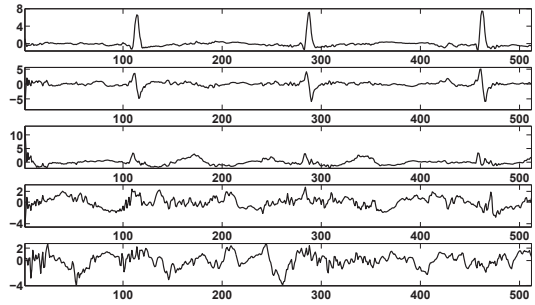


(b) Reconstructed FECG recordings.

Fig. 3: Visual evaluation of the SDAE-based FECG reconstruction technique at 50% compression.



(a) ICA of the original FECG recordings.



(b) ICA of the reconstructed FECG recordings.

Fig. 4: Visual evaluation using the ICA decomposition of original and the reconstructed FECG recordings.

similarity between the independent components (IC) of the reconstructed and the original signals ensures the quality of reconstruction. The ICA decomposition of the original and the reconstructed FECG recordings are depicted in Fig. 4a and Fig. 4b respectively. It is observed that the ICs of the original and the reconstructed signals are visually similar that ensures an acceptable reconstruction quality. Furthermore, in order to illustrate the computational complexity of the SDAE framework and the BSBL-BO algorithm, 100 independent trials are performed on a test data, $x \in R^{nB}$, taken from Physionet and DAISY databases, by varying the CRs. The experiments are run on a 64-bit Microsoft Windows-7 PC with Core-i5 CPU (3.2 GHz) and 8 GB RAM. The average reconstruction time for the SDAE-based method and the BSBL-BO algorithm is shown in Table-I. The recovery time of the SDAE-based reconstruction scheme is substantially superior to that of the BSBL-BO algorithm. To evaluate the performance of the SDAE architecture, the experiments are performed at different CRs. The Table-II illustrates the performance measures (SNR, PRD and PCC) at different CRs for the SDAE-based method and the BSBL-BO algorithm on FECG signals from Physionet database [19]. Table-III provides these same metrics on the DAISY Dataset [24]. These results illustrate that the reconstruction performance by the SDAE method is superior to that of the BSBL-BO method with an understanding of an additional requirement of a separate one-time off-line training stage.

IV. CONCLUSIONS

In this paper, a stacked denoising autoencoder framework is proposed for FECG signal reconstruction. Since an off-line training is performed, it does not affect the real-time signal reconstruction time. The recovery process is fast, as it requires a few matrix-vector multiplications. The performance of the SDAE-based FECG reconstruction architecture is found to be competitive when compared with the BSBL-BO algorithm. The validation of the architecture is performed using the FECG data collected from Physionet database and DAISY database. The performance metrics (SNR, PRD and PCC) and the reconstruction time of SDAE framework are found to be superior to that of the BSBL-BO algorithm.

V. ACKNOWLEDGMENT

This research work is supported by Signals and Systems for Life Science initiative of Ministry of Human Resource Development at Indian Institute of Technology Kharagpur, India.

REFERENCES

[1] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal electrocardiogram via block sparse Bayesian learning," *IEEE Trans. Biomedical Engineering*, vol. 60, no. 2, pp. 300–309, 2013.

[2] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 2011.

[3] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR." in *INTERSPEECH*. Citeseer, 2012.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[5] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. of the 25th Int. Conf. on Machine Learning*. ACM, 2008, pp. 160–167.

[6] M. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *Proc. of the 25th Int. Conf. on Machine Learning*. ACM, 2008, pp. 792–799.

[7] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition, 2008. CVPR 2008*. IEEE, 2008, pp. 1–8.

[8] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann Machines for collaborative filtering," in *Proc. of the 24th Int. Conf. on Machine Learning*. ACM, 2007, pp. 791–798.

[9] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[11] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of the 25th Int. Conf. on Machine Learning*. ACM, 2008, pp. 1096–1103.

[12] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, vol. 19, p. 153, 2007.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] G. Hinton, "A practical guide to training Restricted Boltzmann Machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[15] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 1999, pp. 2443–2446.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Int. Conf. on Artificial Intelligence and Statistics*, 2007, pp. 412–419.

[19] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[20] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[21] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2012.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] R. Paradiso, G. Loriga, and N. Taccini, "A wearable health care system based on knitted integrated sensors," *IEEE Trans. Information Technology in Biomedicine*, vol. 9, no. 3, pp. 337–344, 2005.

[24] B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel, "Daisy: A database for identification of systems," *JOURNAL A*, vol. 38, pp. 4–5, 1997.