Real-time Reconstruction of EEG Signals from Compressive Measurements via Deep Learning

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

Rabab Ward
University of British Columbia
Vancouver, Canada
rababw@ece.ubc.ca

***Abstract*— To elongate the battery life of sensors worn in wireless body area networks, recent studies have advocated compressing the acquired biological signals before transmitting them. The signals are compressed using compressive sensing (CS), by projecting them onto a lower dimension. The original signals are then recovered using CS recovery techniques at the base station, where the computational power is assumed to be abundant. This assumption however is not entirely true when a mobile phone acts as the base station. The computational capacity of a mobile phone is limited; therefore solving the CS recovery problem in the phone would be time consuming. In many cases (e,g. heart stroke detection or monitoring applications) this latency cannot be tolerated. In this work we propose a new technique to solve the inverse problem using stacked autoencoders. We show that the reconstruction of the proposed method can be done in real-time, and there is only a slight degradation in accuracy compared to CS based inversion methods.**

***Index Terms*— autoencoder, WBAN, EEG**

I. INTRODUCTION

In recent years there has been a growing interest in tele-monitoring people's health conditions. In developed countries the target subject is the growing elderly population. In developing countries like India, where more than 70% of the population is rural, the reach of medical facilities is limited. There is a need to improve tele-monitoring facilities for such rural populace. In both cases, the subjects may not require immediate medical attention, but such a tele-monitoring facility would definitely improve the health condition of the overall population.

For tele-monitoring the patient wears some sensor nodes that acquire his/her biomedical signals and these signals are transmitted instantaneously to a remote facility where the patient is monitored by analyzing the acquired signals. Current systems for acquiring some common signals like EEG and ECG are however wired, cumbersome and need trained personnel to use them. We cannot expect the elderly population to undergo such training; neither can the semi-skilled paramedics in developing countries operate such complex devises. Therefore, the acquisition devices must be simple, portable and wireless.

The main constraint in such a wireless body area network (WBAN) is the energy consumption at the sensor nodes. Since the desired device is a portable wireless system, the battery would be small. Therefore the sensor should be designed such that the battery lasts as long as possible. There are three power sinks in a WBAN – sensing, processing and transmission. The energy required for transmission is the highest; therefore every effort needs to be made to reduce the transmission cost. One can save the transmission energy by compressing the signal before transmitting it However, modern compression techniques require considerable processing power; the portable sensor nodes are not endowed with such capabilities and hence such compression techniques cannot be implemented directly at the sensor nodes.

We therefore envisage the wireless sensor nodes to be used for collecting the data, compressing them using a simple technique (e.g a compressive sensing (CS)) and transmitting them wirelessly to a local smartphone. The smartphone would reconstruct the received signals and compress them using a more efficient compression method (transform coding) before sending them to the remote location for analysis and patient monitoring. In such a scenario a simple CS technique is employed at the sensor node; whereby a section of the signal is projected onto a lower dimension by a simple matrix vector product. The smartphone reconstructs the signal using compressive sensing (CS) reconstruction techniques. It then compresses the signal once again using a transform coding technique and sends the compressed signal wirelessly to a remote station.

Such a system strikes a balance between computational complexity and energy efficiency. The sensor node cannot employ complex transform coding technique owing to its limited computational resources; it also needs to save its battery power. Therefore it uses a cheaper strategy to compress – matrix vector product. Since this is not a very efficient compression technique, the resultant compressed signal is sent to a local smartphone which in turn reconstructs the original signal (employing CS) and optimally compresses it using transform coding before transmission.

Solving CS problems is computationally demanding; they need to be solved iteratively and hence can never be real-time. Therefore solving the CS reconstruction at the smartphone is time consuming, not to mention the associated heating of the device which may cause associated problems to the device. In many situations, the delay in reconstruction cannot be tolerated. For example consider a monitoring activity like stroke detection, if the aforementioned protocol is in place the clinician at the remote medical facility has to wait about a minute (time needed for CS reconstruction) for every 2-3 seconds of signal!

CS reconstruction is iterative and therefore is inherently time consuming and can never be real-time. In this work we propose an alternative reconstruction technique that uses deep learning – one based on stacked autoencoder (SAE). We will learn an SAE to solve the inverse problem; the learning process is offline and time consuming. However the operational stage is real-time since reconstruction would only require a few matrix vector products.

This is the first work that proposes to employ stacked autoencoders to solve an inverse problem. CS reconstruction is a non-linear inverse problem. Proponents of deep learning argue that autoencoders can learn arbitrary functional relationships given enough training data. Here we will train a stacked autoencoder to emulate a CS type inversion operation.

To understand the problem, the readers must understand compressive sensing. We will therefore review the basics for the sake of completeness. Literature review on the topic of biomedical signal compression and reconstruction will also be discussed in the following section. The proposed method is described in section 3. The experimental results are shown in section 4. The conclusions of this work and future direction of research is discussed in section 5.

II. Compressed Sensing

Compressed Sensing studies the problem of solving an under-determined system of linear equations when the solution is known to be sparse. In general a problem of the following form has infinitely many solutions;

$$y_{m\times 1} = A_{m\times n} x_{n\times 1},\ m < n \tag{1}$$

However, when the solution is sparse, it is necessarily unique [1]; i.e. if there is a k-sparse solution (only k non-zeroes out of n), then there cannot be another solution which is k-sparse as well or there cannot be a solution which is $k+1$ sparse. Thus, if one seeks a sparse solution to (1), one may as well seek the sparsest solution. Mathematically, this is formulated as follows:

$$\min_x \|x\|_0 \text{ subject to } y = Ax \tag{2}$$

The l_0-norm counts the number of non-zeroes in the vector.

Unfortunately minimizing the l_0-norm is an NP hard problem [1] and thus is not practical. There are two ways to address this problem; the first approach is to solve (2) using greedy approximate (sub-optimal) algorithms such as variants of Orthogonal Matching Pursuit [2]. The other approach is to relax the NP hard l_0-norm by its nearest convex surrogate - the l_1-norm (3) [3, 4]. The second method offers stronger theoretical recovery guarantees compared to the greedy approach.

$$\min_x \|x\|_1 \text{ subject to } y = Ax \tag{3}$$

The l_1-norm is the sum of absolute values in the vector. The l_1-norm minimization (3) is a convex problem and can be solved efficiently via linear programming.

In practice, the system is corrupted by white Gaussian noise; The noisy version of (1) is expressed as,

$$y = Ax + \eta,\ \eta \sim \qquad) \tag{4}$$

In such a situation, the equality constraint of (3) is relaxed by a quadratic constraint,

$$\min_x \|x\|_1 \text{ subject to } \|y - Ax\|_2^2 \le \varepsilon,\ \varepsilon = m\sigma^2 \tag{5}$$

This is a perturbed linear programming problem and can be solved by quadratic programming.

To make the scenario more realistic one should consider approximately sparse / compressible solutions; the solution (x) is assumed to have a fast decay but is not exactly compressible. By compressible we mean that the top-k values (irrespective of the sign) in x will contain maximum possible energy, i.e. the l_2-norm difference between the original solution x^0 and the top-k term approximation x^k, $\|x^0 - x^k\|_2^2$ will be very small. In such a situation, (5) is employed to recover the solution. The low values in x (those not in x^k) are considered as modeling noise and cannot be recovered.

It has been proven in [3] that for noisy compressible cases, the recovery error by (5) is bounded as follows,

$$\|x^* - x^0\|_2^2 \le C_1\varepsilon + C_2 \frac{\|x^0 - x^k\|_1}{k} \tag{6}$$

Here x^* is the recovered solution.

This expression (6) indicates that the error between the actual and reconstructed solution is bound by the noise variance (ε) and the best k-term approximation of the compressible signal x^0. The first term, $C_1\varepsilon$ arises owing to the noise in the system; the second term $C_2 \frac{\|x^0 - x^k\|_1}{k}$ is due to the fact that the signal is not exactly sparse, but compressible.

So far, we have only discussed how the solution to the inverse problem can be recovered. It is a good news that there are theoretical recovery guarantees which proves that the convex problem (3 / 5) can yield the same solution as the NP hard problem (2). However, one must remember that there is no free lunch. The minimum number of equations required by (2) in order to solve the system of equations is:

$$m = 2k + 1 \tag{7}$$

However, when the convex optimization (3 / 5) is employed ore sampes are required [4]:

$$m = Ck \log n \tag{8}$$

where C is a constant.

Combining the concept of minimum number of equations (8) with the idea of approximate sparsity (6), one can conclude that - the sparser the signal the better will be the recovery; i.e. lower the reconstruction error. This is an essential concept that will be important later on.

Natural signals are almost never sparse in their physical domain, e.g. biomedical signals EEG, ECG, MEG and speech are not sparse in time domain and images are never sparse in pixel domain. However most natural signals have an approximately sparse representation in a transform domain, viz. speech is sparse in Short Time Fourier Transform, images are sparse in wavelets etc.

In CS, we are mostly interested in orthogonal and tight-frame transforms[1]; both of these follow the analysis-synthesis equations:

$$\begin{aligned} &analysis: \ \alpha = Wx \\ &synthesis: \ x = W^T \alpha \end{aligned} \qquad (9)$$

Here x is the signal of interest (dense), W is the sparsifying transform and α the transform coefficients is assumed to be sparse.

Using the synthesis form, the system of equations (4) can be expressed as,

$$y = AW^T \alpha + \eta \qquad (10)$$

Thus, we are back to the sparse regime and the solution can be recovered by,

$$\hat{\alpha} = \min_{\alpha} \|\alpha\|_1 \text{ subject to } \|y - AW^T \alpha\|_2^2 \leq \varepsilon \qquad (11)$$

The signal is then estimated as: $\hat{x} = W^T \hat{\alpha}$.

In CS, the matrix A is called the measurement operator; and W is the sparsifying transform. The coherence (μ) between A and W plays an important role in the number of measurements required [9]:

$$m = C\mu^2 k \log n \qquad (12)$$

The more the coherence, more the number of measurements required. In other words, more the coherence worse is the recovery (for a fixed number of measurements).

III. Literature Review

One of the earliest works that applied CS for EEG signal compression and transmission is [6]. It projected the EEG signal onto an i.i.d Gaussian basis for compression and used CS to recover the EEG signal by exploiting the signal's sparsity in the Gabor domain. The compression can be expressed as –

$$b = \Phi z \qquad (13)$$

where z is the EEG signal, Φ is the projection / compression matrix and b is the compressed data.

The work [6] employed a synthesis prior formulation for sparse signal recovery using Gabor as a sparsifying basis. Synthesis prior only works for orthogonal and tight-frame sparsifying transforms such as DCT, wavelets, curvelets etc. Unfortunately Gabor is not an orthogonal or tight frame transform. Thus their formulation, can perfectly recover the Gabor coefficients, but reconstructing the EEG signal will not be accurate. This paper uses a Gaussian matrix for compression; for practical reasons, a Gaussian compression basis is not very suitable; since the matrix is dense and therefore is neither memory efficient nor easy to be operated with.

In [7], different sparsifying transforms were compared – wavelets, Gabor, splines; it was reported that Gabor yielded the best reconstruction results. However, this paper is fraught with the same issue as [6]; they pose EEG reconstruction as a synthesis prior problem using the Gabor basis - which is not an orthogonal transform. This work too compressed the signal using an i.i.d Gaussian matrix.

The problem arising in [6, 7] owing to the non-orthogonality of the Gabor transform was rectified in [8]; the authors simply used an analysis prior formulation (14) instead of the more often used synthesis prior formulation and the results improved drastically.

$$\min_z \|\Psi z\|_1 \text{ subject to } b = \Phi z \qquad (14)$$

Here Ψ denotes the Gabor transform. The analysis prior formulation is more generic and is applicable to any linear transform; it need not be Orthogonal or Tight-frame. In [8] the issue regarding the inoperability of Gaussian measurement matrices was also rectified; they proposed using sparse binary matrices [9]. Such matrices do not satisfy CS properties in theory but yields excellent results in practice and are easy to implement in hardware.

The possibility of exploiting inter-channel correlation in order to improve EEG signal reconstruction was mentioned in [6], but there was no concrete idea regarding how to model it. This problem is partially addressed in [10]. They do not explicitly model the inter-channel correlation, but frame a joint reconstruction problem where the signals from all the channels are reconstructed simultaneously. This work uses wavelets as the sparsifying basis; and could therefore formulate the problem as a synthesis prior.

Let 'i' denote the channel number, then the compression for this channel can be represented as –

$$b_i = \Phi z_i \qquad (15)$$

This can be organized as follows,

$$\begin{bmatrix} b_1 \\ \ldots \\ b_C \end{bmatrix} = \begin{bmatrix} \Phi & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \Phi \end{bmatrix} \begin{bmatrix} z_1 \\ \ldots \\ z_C \end{bmatrix} \qquad (16)$$

The concatenated solution will be sparse in wavelet domain; this sparsity of the signals from all channels is exploited in [10]. At a first glance this looks like a joint reconstruction problem, but a closer look reveals that this is actually as good as channel by channel reconstruction; this formulation (16) does not exploit any structure across the channels.

A more recent work model assumes a block structure of the EEG signals [11] in a transform domain (DCT or wavelet). There is no theoretical or physical intuition behind this assumption; however it was shown in [11] that a Block Sparse Bayesian Learning (BSBL) algorithm yields good recovery results. This work too uses a random Binary matrix for projection.

A recent work proposed CS techniques for EEG signal compression and transmission, but instead of sending the raw

---

[1] $Orthogonal: \ W^T W = I = WW^T$

$Tight-frame: \ W^T W = I \neq W^T W$

EEG signals it subtracted the mean from all the signals thereby reducing the number of bits to be transmitted [12]. However, such a scheme will not be energy efficient, since in order to compute the mean EEG signal, the nodes need to communicate with each other – and such communication consumes considerable energy.

In another work [13], it was shown that for certain specific tasks like seizure detection, instead of sending the full signal, it is possible to send some distinct features which can be further analyzed at the base station for possible risks. Such a technique requires more computation than standard CS techniques, but the number of features to be transmitted are very few. Unfortunately such a technique cannot be generalized for other applications.

EEG signals are inherently correlated. Prior studies hinted at using the inter-channel correlation but did not propose any formulation to exploit this information. In [14] the common inter-channel sparsity pattern was exploited to capture the correlation. Instead of (16), the organization in [14] is –

$$\left[b_1 \mid \ldots \mid b_C\right] = \Phi\left[z_1 \mid \ldots \mid z_C\right] \quad (17)$$

The signals from different channels being correlated will share a common sparsity pattern in the transform domain. Thus the matrix $\Psi\left[z_1 \mid \ldots \mid z_C\right]$ will be row-sparse. Hence can be recovered by $l_{2,1}$-minimization.

$$\min_Z \|Z\|_{2,1} \text{ subject to } B = \Phi Z \quad (18)$$

where $B = \left[b_1 \mid \ldots \mid b_C\right]$ and $Z = \left[z_1 \mid \ldots \mid z_C\right]$

Here the $l_{2,1}$-norm is defined as the sum of the l_2-norm of the rows. The outer l_1-norm (summation) promotes sparsity in the selection of rows. The inner l_2-norm promotes a dense solution in the selected row [15].

In [15], a separate take on correlation is proposed. The authors argued that if the signals are correlated they will be linearly dependent; therefore when stacked as columns will form a low-rank matrix, i.e. Z (18) will be low-rank. This property was exploited in [16]; a matrix completion based formulation was utilized to recover the signal ensemble.

$$\min_Z \|Z\|_{NN} \text{ subject to } B = \Phi Z \quad (19)$$

The nuclear norm (defined as the sum of nuclear values) is the closest convex surrogate of rank. In [16] the problem was solved using the singular value shrinkage algorithm [17].

In [18, 19] the inter-channel correlation was exploited by jointly formulating the reconstruction as a combined row-sparse and low-rank recovery problem. Some recent studies [20, 21] exploited the Blind Compressive Sensing (BCS) formulation; here instead of using a fixed sparsifying basis like wavelet / Gabor, it is learnt from the data.

IV. Proposed Methodology

Let us reiterate the protocol for EEG signal compression at the sensor nodes. To compress the signal, a batch of sample are projected onto a suitable matrix. Say z is the signal Φ and the compression / projection matrix, the lower dimensional samples are obtained as:

$$b = \Phi z$$

CS assumes that the signal is compressible in some domain (W) like wavelet or Gabor. Therefore it can be reconstructed using:

$$\ldots - \Phi\Psi\alpha\|_2^2 + \lambda\|\alpha\|_1 \quad (20)$$

Here α is the transform coefficient.

One can also solve the sparse inverse problem using greedy algorithms like Orthogonal Matching Pursuit (OMP) – this would solve the ideal l_0-norm minimization problem albeit sub-optimally.

All the studies from [6-13] are basically variants of the basic approach outlined above. The rest [14-19] are only applicable on EEG signal ensembles when the number of channels are larger (64 and beyond). This is a very restrictive scenario as most commercial (non-medical grade) EEG acquisition devises like Emotiv and Neurosky have 2-14 channels. In such situations the later techniques [14-19] are inapplicable. Therefore we will concentrate our discussion on the basic CS technique discussed here.

A. Non-linearity of CS Inversion

We will first dissect two popular CS reconstruction techniques – l_1-minimization and OMP, to show that they are indeed non-linear inversion techniques. To do so, we will use the simple notation for the problem. The task is to find a sparse solution to the inverse problem: $y=Ax$.

The l_1-minimization problem can be simply solved using Iterative Soft Thresholding algorithm [22]. ISTA consists of two steps –

$$c = x_{k-1} + \sigma A^T\left(y - x_{k-1}\right) \quad (21)$$

$$x_k = signum(\mathrm{c}) \cdot \max\left(0, |c| - \lambda\sigma\right) \quad (22)$$

The first step is a simple gradient descent step with step-size σ; this is a linear operation. The second step is soft thresholding, it is a non-linear operation owing to the max(.) operator. This makes l_1-minimization a non-linear technique.

Similarly we can dissect the OMP algorithm.

Initialize: x=0; Ω=[]
In iteration k

$$r = y - A_\Omega x_\Omega$$

$$c = abs\left(A^T r\right)$$

$$\gamma = \arg\max c$$

$$\Omega = \Omega \bigcup$$

$$x_\Omega = \min_x \|y - A_\Omega x\|_2^2$$

The OMP algorithm starts with an all zero estimate of the solution. In every iteration, it selects one non-zero index by computing the correlation between the residual (r) and the columns of A. The index having the maximum correlation is assumed to have a non-zero value in x. This index is added to the present support set (Ω). The coefficients of *x* belonging to the current support is estimated via least squares. OMP too is a non-linear technique since it requires computing the max(.) in every iteration.

B. Proposed Technique for Real-time Reconstruction

In short, whatever be the recovery technique to find a sparse solution one needs to resort to a non-linear method. CS has been very successful in the past, but suffers from one severe shortcoming – it is an iterative approach and hence cannot be used for real-time problems. This may be a problem for many EEG applications. For example in any monitoring application, the doctor needs to see the signals in real-time. If the task is to predict a stroke, the signs only arrives a few seconds prior to the actual stroke. In such case, a literally life and death situation, retrospectively reconstructing the samples will not be acceptable.

The other challenge is computational. In the introduction we mentioned that the reconstruction is likely to be done in a smartphone. Usually a smartphone does not have enough computational power to solve such a demanding inversion problem. Hence the inversion is very slow. Besides, with such computationally challenging tasks, the phone heats up significantly; this in turn reduces the life of the device.

To the best of our knowledge this is a first work that proposes a technique for real-time reconstruction of EEG signals. Although our discussion pertains to such signals in particular, the technique developed here can be applied to any type of biomedical signal in general.

It is well known that Neural Networks act as universal function approximators. Given enough training data the non-linear activation functions learn to represent arbitrary functions; this was proven by Cybenko [23] and Hornik [24]. A more fundamental work on this topic dates back to Kolmogorov [25] where he showed that a continuous function of many variables can be approximated by a superposition of continuous functions of one variable. We make use of this universal functional approximation property of neural networks to address our problem.

The protocol remains the same as before. A part of the signal (*z*) is projected onto a compression matrix (Φ). The compressed data (*b*) is transmitted to a smartphone where the original signal is reconstructed. This signal is compressed using transform coding and transmitted to a medical facility for analysis. The only change happens in the reconstruction of the signal at the smartphone. Instead of solving an inverse problem we will 'learn' the inversion.

To learn the inversion we need training data. Let this be *Z*; the columns of *Z* are training samples (a fixed number of samples from EEG signals). From this, we will simulate the compression by:

$$B = AZ \tag{23}$$

First step in the inversion process is to compute a poor man's inverse by applying the transpose of the projection matrix on the compressed data, i.e.

$$Z' = \Phi^T B \tag{24}$$

Had the projection matrix been orthogonal, the inversion (21) would have been ideal, but it is not – hence the name 'poor man's inverse'.

The *Z'* is a 'noisy' version of *Z*. In recent years, denoising autoencoders have been trained to 'clean' samples [26-28]. We make use of the same approach here. We train an autoencoder with noisy Z' at the input and Z at the output; the autoencoder will learn to map Z' to Z.

C. Autoencoder Design

To design an autoencoder, there are two design aspects that needs to be specified - the number of hidden layers and the number of nodes in these hidden layers. In principle it is possible to learn arbitrary non-linear functional relationships with a single layer having a very large number of nodes - usually much larger than the dimensionality of the input samples. But this is not preferred. This is because, to learn a large number of weights the volume of training data needs to be very large. This is usually not available. Over the years, proponents of stacked autoencoders and other deep learning tools have proposed a stacked approach instead.

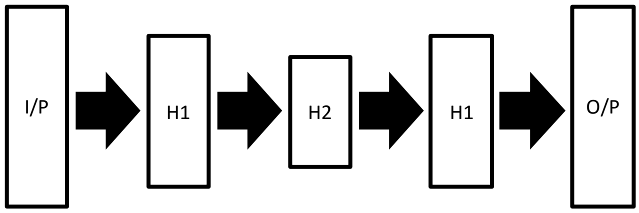

Fig. 1. Two Layer SDAE

In practice the number of nodes in the hidden layers reduce as one goes deeper into the SDAE (although this is more prevalent, this is not the only approach).

To understand the learning process, we take a simple SDAE in Fig. 1. First the weights corresponding to the outermost hidden layer (H1) is learnt by solving the following problem.

$$\min_{W_D^1, W_E^1} \left\| Z - W_D^1 \phi(W_E^1 Z') \right\|_F^2 \tag{22}$$

Here W_E^1 is the encoding weights between input at H1 and W_D^1 is the decoding weights between H1 and output. The activation function is ϕ.

This learns the weights and gives us the first level features $Z = \phi(W_E^1 X)$. The first level features are now used as training data for the second input layer. The weights for the second layer is solved by,

$$\min_{W_E^2, W_D^2} \left\| Z - W_D^2 \phi(W_E^2 Z) \right\|_F^2 \tag{23}$$

This process can be continued if there are more than 2 hidden layers.

There is no analytical rule that helps design an autoencoder. It is largely based on the experience of the researcher. Bengio and Hinton have independently suggested that care must be taken while reducing the number of nodes, the reduction should not be too fast or too soon.

Learning the non-linear inversion operation is time consuming. This can be learnt in a powerful computer and the learnt model ported to a smartphone during actual operation. The advantage of our proposed method is that it is considerably fast compared to CS based techniques. The computational complexity for every iteration of a general purpose l_1-minimization algorithm is $O(n^3)$ and the iterations need to be run approximately $O(\sqrt{n})$ [29]. Stacked autoencoders only require matrix vector productions. The computational complexity of our proposed approach is therefore $O(n^2)$ - matrix vector multiplications; this is much smaller than $O(n^{3\frac{1}{2}})$.

V. Experimental Evaluation

The experiments are carried out on the BCI III dataset 1 [30]. During the BCI experiment, the subject had to perform imagined movements of either the left small finger or the tongue. The time series of the electrical brain activity was picked up during these trials using an 8x8 ECoG platinum electrode grid which was placed on the contralateral (right) motor cortex. The grid was assumed to cover the right motor cortex completely. All recordings were performed with a sampling rate of 1000Hz. After amplification the recorded potentials were stored as microvolt values. Every trial consisted of either an imagined tongue or an imagined finger movement and was recorded for 3 seconds duration. To avoid visually evoked potentials being reflected by the data, the recording intervals started 0.5 seconds after the visual cue had ended.

In this work our goal is to show that our proposed technique does not degrade the quality of reconstruction too much, compared to CS, but can accelerate reconstruction drastically. We compare our work with [8] – this is the technically correct way to frame reconstruction as an analysis prior CS problem.

We use a two layer autoencoder, the number of nodes in the hidden layers are halved from the previous layer. We reconstruct chunks of EEG signals of length 256; therefore in the first (and third) hidden layers the number of nodes are 128 and in the innermost layer the number is 64.

We test the results on two compression / projection techniques. In the first, we use a sparse binary matrix. Such a matrix is easy to realize in hardware and hence is practical [8, 10, 11]. The second approach is to randomly sub-sample the signal; this too is practical and is more energy efficient as shown in [15-19].

In Table 1 we show the reconstruction results (mean and standard deviations) from these two projection matrices for different sampling ratios using CS and our proposed method. The accuracy is measured in terms of Normalized Mean Squared Error $NMSE = \frac{\|groundtruth - reconstructed\|_2}{\|reconstructed\|}$.

TABLE I. Reconstruction Accuracy

Technique	Sparse Binary		Sub-sampling	
	2:1	4:1	2:1	4:1
CS	0.121, ±0.056	0.262, ±0.114	0.160, ±0.084	0.328, ±0.186
Proposed	0.142, ±0.034	0.288, ±0.060	0.178, ±0.056	0.362, ±0.090

The results show that our proposed method is slightly worse than CS based technique in terms of average reconstruction error. But, the reconstruction is more robust (lesser deviation). This is because CS reconstruction is dependent on the signal structure – the sparser it can be represented the better is the recovery. Owing to variability in the signals the reconstruction therefore changes considerably. Our method is less sensitive to signal structure; hence is more robust.

For visual evaluation, we show the results in Fig. 2. The original and the reconstructed signals are overlayed for comparison. The CS reconstruction yields slightly better result.

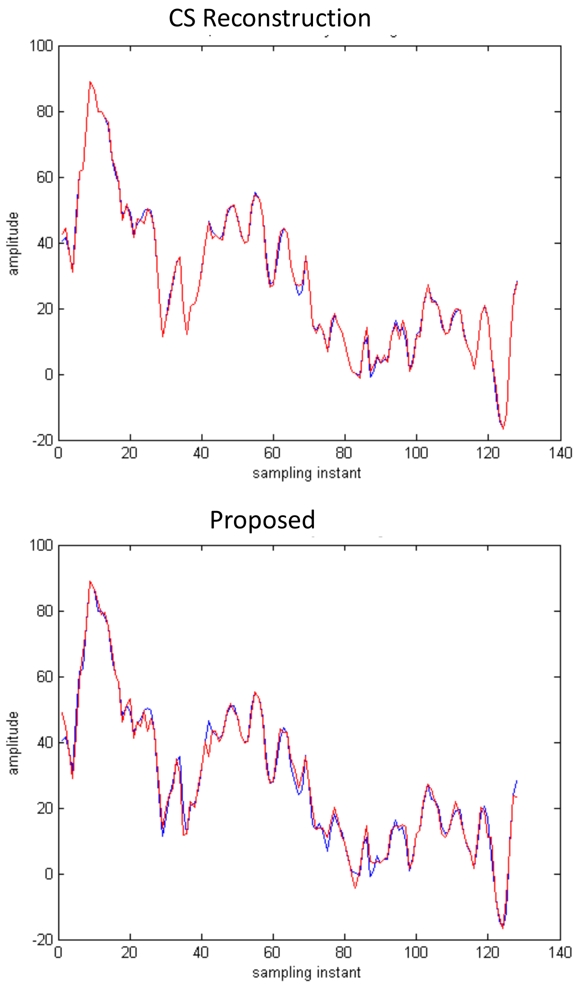

Fig. 2. Visual Comparison of Reconstruction

Next we discuss the average reconstruction speeds from the two techniques – CS vis-à-vis our proposed. The experiments were run on a Core i7 CPU (3.1 GHz) having a 16 GB RAM running 64 Windows 7. The platform is Matlab 2012a.

TABLE II. RECONSTRUCTION TIME (SECONDS)

Technique	**Sparse Binary**		**Sub-sampling**	
	2:1	4:1	2:1	4:1
CS	1.20	0.9	1.41	1.03
Proposed	0.011	0.009	0.013	0.009

Our method is two orders of magnitude faster and can reconstruct in real-time. The data is collected at 1kHz, to collect 256 time-points approximately 0.25 seconds are needed; our method only takes about 0.01 seconds to reconstruct 256 time-points. On the other hand, the time for recovering signals of duration 0.25 seconds, CS takes around 1 second to recover. This delay would not permit real-time recovery.

VI. CONCLUSION

This is the first work that proposes a technique to solve the EEG reconstruction problem using stacked autoencoder approach. The advantage of our method is that it is real-time and can be run with cheaper computational resources. All prior techniques on signal reconstruction were based on the CS framework. This requires significant computational power and hence has a larger run-time. The reconstruction is not real-time. This precludes the use of such reconstruction techniques in monitoring applications.

Our method is slightly worse than CS in terms of reconstruction accuracy. We would like to improve the accuracy at a future stage. It has been shown that using sparsity in autoencoders improves performance [28]. We would like to test if that same can be leveraged to solve our problem. We would also work on accelerating the training time for the autoencoder. Current autoencoders use back-propagation for training, this is slow. We would like to employ modern optimization techniques to improve convergence.

ACKNOWLEDGEMENT

This work was made possible by NPRP grant 7 - 684 - 1 - 127 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] B. K. Natarajan, "Sparse approximate solutions to linear systems", SIAM Journal on Computing, 24(1995), 227-234

[2] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit", IEEE Trans. on Information Theory, 53(12) pp. 4655-4666, 2007

[3] E. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics, 59(8), pp. 1207-1223, 2006.

[4] E. Candès and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies? (IEEE Trans. on Information Theory, 52(12), pp. 5406 - 5425, December 2006

[5] Y. Chen, "Incoherence-Optimal Matrix Completion", arXiv:1310.0154.

[6] S. Aviyente, "Compressed Sensing Framework for EEG Compression", IEEE Workshop on Statistical Signal Processing, pp.181,184, 2007.

[7] A. M. Abdulghani, A. J. Casson and E. Rodriguez-Villegas, "Quantifying the performance of compressive sensing on scalp EEG signals", International Symposium on Applied Sciences in Biomedical and Communication Technologies,pp.1,5, pp. 7-10, 2010

[8] M. Mohsina and A. Majumdar, "Gabor Based Analysis Prior Formulation For EEG Signal Reconstruction", Biomedical Signal Processing and Control.

[9] A. Gilbert and P. Indyk "Sparse recovery using sparse matrices" Proceedings of the IEEE, vol. 98, no. 6, pp. 937–947, 2010.

[10] M. Hosseini Kamal, M. Shoaran, Y. Leblebici, A. Schmid and P. Vandergheynst. Compressive Multichannel Cortical Signal Recording. 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, Canada, 2013.

[11] Z. Zhang, T. P. Jung, S. Makeig and B. D. Rao, "Compressed Sensing of EEG for Wireless Telemonitoring with Low Energy Consumption and Inexpensive Hardware", IEEE Transactions on Biomedical Engineering, (accepted).

[12] S. Fauvel and R. K. Ward, "An Energy Efficient Compressed Sensing Framework for the Compression of Electroencephalogram Signals", Sensors 2014, 14(1), 1474-1496

[13] J. Chiang and R. K. Ward, "Energy-Efficient Data Reduction Techniques for Wireless Seizure Detection Systems", Sensors 2014, 14(2), 2036-2051

[14] A. Majumdar and R. K. Ward, "Non-Convex Row-sparse MMV Analysis Prior Formulation For EEG Signal Reconstruction", Biomedical Signal Processing and Control, Vol. 13, pp. 142–147, 2014.

[15] A. Majumdar and R. K. Ward, "Synthesis and Analysis Prior Algorithms for Joint-Sparse Recovery", IEEE ICASSP, pp. 3421-3424, 2012.

[16] A. Majumdar, A. Gogna and R. Ward, "Low-rank Matrix Recovery Approach For Energy Efficient EEG Acquisition for Wireless Body Area Network", Sensors, Special Issue on State-of-the-art Sensor Technologies in Canada, Vol. 14(9), pp. 15729-15748, 2014

[17] A. Majumdar and R. K. Ward, "Some Empirical Advances in Matrix Completion", Signal Processing, Vol. 91 (5), pp. 1334-1338, 2011.

[18] A. Shukla and A. Majumdar, "Exploiting Inter-channel Correlation in EEG Signal Reconstruction", Biomedical Signal Processing and Control, Vol. 18 (4), pp. 49–55, 2015

[19] A. Majumdar, A. Shukla and R. K. Ward, "Combining Sparsity with Rank-Deficiency for Energy Efficient EEG Sensing and Transmission over Wireless Body Area Network", ICASSP 2015.

[20] A. Majumdar and R. K. Ward, “Energy Efficient EEG Sensing and Transmission for Wireless Body Area Networks: A Blind Compressed Sensing Approach”, Biomedical Signal Processing and Control, in press.

[21] A. Shukla and A. Majumdar, "Row-sparse Blind Compressed Sensing for Reconstructing Multi-channel EEG signals", Biomedical Signal Processing and Control, Vol. 18 (4), pp. 174–178, 2015

[22] Sparse Signal Restoration: cnx.org/content/m32168/latest/

[23] G. Cybenko, “Approximations by superpositions of sigmoidal functions”, Mathematics of Control, Signals, and Systems, 2 (4), 303-314, 1989.

[24] K. Hornik, “Approximation Capabilities of Multilayer Feedforward Networks”, Neural Networks, Vol. 4(2), pp. 251–257, 1991.

[25] Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. Doklady Akademii Nauk SSSR, 144, 679-681; American Mathematical Society Translation, 28, 55-59 [1963]

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

[27] M. Chen, K. Weinberger, F. Sha, Y. Bengio, "Marginalized Denoising Autoencoders for Nonlinear Representation", ICML 2014.

[28] K. H. Cho, “Simple Sparsification Improves Sparse Denoising Autoencoders in Denoising Highly Noisy Images”, ICML 2013.

[29] A. Yang, A. Ganesh, S. Sastry and Y. Ma, “Fast L1-Minimization Algorithms and An Application in Robust Face Recognition: A Review”, Technical Report No. UCB/EECS-2010-13

[30] http://www.bbci.de/competition/iii/