Deep Feature Learning Using Target Priors with Applications in ECoG Signal Decoding for BCI

Zuoguan Wang[1], **Siwei Lyu**[2], **Gerwin Schalk**[3], **Qiang Ji**[4]
[1,4]Dept. of ECSE, Rensselaer Polytechnic Institute, Troy, NY, 12180
[2]Computer Science, Univ. at Albany, SUNY, Albany, NY, 12222
[3]Wadsworth Center, NYS Dept. of Health, Albany, NY, 12201
Zuoguanwang@gmail.com, lsw@cs.albany.edu, schalk@wadsworth.org, jiq@rpi.edu

Abstract

Recent years have seen a great interest in using deep architectures for feature learning from data. One drawback of the commonly used unsupervised deep feature learning methods is that for supervised or semi-supervised learning tasks, the information in the target variables are not used until the final stage when the classifier or regressor is trained on the learned features. This could lead to over-generalized features that are not competitive on the specific supervised or semi-supervised learning tasks. In this work, we describe a new learning method that combines deep feature learning on mixed labeled and unlabeled data sets. Specifically, we describe a weakly supervised learning method of a prior supervised convolutional stacked auto-encoders (PCSA), of which information in the target variables is represented probabilistically using a Gaussian Bernoulli restricted Boltzmann machine (RBM). We apply this method to the decoding problem of an ECoG based Brain Computer Interface (BCI) system. Our experimental results show that PCSA achieves significant improvement in decoding performance on benchmark data sets compared to the unsupervised feature learning as well as to the current state-of-the-art algorithms that are based on manually crafted features.

1 Introduction

Recently, feature learning using deep hierarchal structures has emerged as an effective methodology in machine learning to automatically extract features from data. Successful applications of deep feature learning algorithms can be found in recent literature across a variety of domains such as object recognition [Nair and Hinton, 2009], acoustic and speech processing [Mohamed *et al.*, 2011], and video analysis [Le *et al.*, 2011]. However, many current deep feature learning algorithms are unsupervised in nature, which use the fidelity in reconstruction of the unlabeled training data set as the optimization criterion. When applying these algorithms to supervised or semi-supervised learning tasks, the effect of the target variables (e.g., class labels or regression targets) usually appear after the low-level features have been learned. While it has been argued that the postponed introduction of target variables is to better regularize the feature learning, general statistical characteristics in the target variables can still provide information to extract features that are more effective for the specific task. This is particularly the case when the target variables are of high dimensionality and complex structures.

The main objective of this work is to incorporate information in the target variables in a supervised learning task into the deep feature learning algorithms without causing overfitting, which keeps a good balance between feature generality and task specificity. To this end, we develop a new deep feature learning method that we term as prior supervised convolutional stacked auto-encoder (PCSA). PCSA is based on the stacked auto-encoder [Bengio *et al.*, 2007], the parameters of which are organized in a convolutional manner to reduce parameter dimensionality [Lecun *et al.*, 1998]. What distinguishes PCSA from the conventional unsupervised deep feature learning methods is that it incorporates a probabilistic model of the target variable in the learning task (known as target priors). The main function of the target prior is to weakly supervise the feature learning by enforcing the predictions of targets from the learned features to be consistent with the general statistical characteristics of the target variables. Thus it serves as a bottleneck that funnels relevant aspects of the target variables into the feature learning process.

The target prior and the unsupervised learning objective balance the generality of the features learned from the unlabeled data and the specificity of the supervised learning task. The use of uncorresponded target variables has a further advantage that it relieves the burden of collecting a large set of labeled data in typical supervised learning tasks. We demonstrate the effectiveness of PCSA in the application of decoding task of an ECoG based BCI system.

BCI systems interpret brain signals into control commands to output devices, and provide a promising means to restoring mobility for patients with physical disabilities [McFarland and Wolpaw, 2008]. Recent studies in neurobiology have suggested that Electrocorticography (ECoG) signals have strong correlations with limb motions [Liang and Bougrain, 2009; Kubánek *et al.*, 2009; Wang *et al.*, 2012a; 2010], and provide a good signal source to build effective BCI systems. The decoding problem in ECoG BCI system concerns transforming the time series corresponding to the lively recorded ECoG signals into the time series of kinematic parameters of the tar-

get motor component (e.g., limb or finger position and joint angles).

The raw ECoG signals are not suitable for direct decoding because of their high dimensionality and the presence of background noise. More usually, low dimensional features robust to noise and relevant to the decoding task are extracted from the ECoG signals and used in the subsequent decoding tasks. So far the most effective ECoG features are based on the power spectrum density [Kubánek *et al.*, 2009] or physical power [Bougrain and Liang, 2009] of a few fixed frequency subbands. These features are justified on empirical observations that certain frequency subbands seem to have strong correlation with the motion intents of the subject that map to motor controls of various brain areas.

We apply the PCSA framework to extract more effective features automatically from a large set of ECoG recordings. In particular, we take advantage of the rich structural regularities in the target variables (in this case, the time series of the kinematic parameters). Such regular patterns come from the commonality of the kinematic parameter time series in performing the same task, notwithstanding the difference across the cortical structure of the subjects. In this work, such common statistical patterns in the target time series are captured with a Gaussian Bernoulli restricted Boltzmann machine (RBM) [Hinton and Salakhutdinov, 2006], which is used as the target prior model. When applied to the decoding task of the ECoG signals, our method demonstrates promising performance improvement compared to the state-of-the-art manually selected features.

The rest of this paper is organized as follows. Section 2 briefly discusses related work of the proposed model. Section 3 introduces the PCSA model. Its learning is discussed in section 3.3 in which the target prior constraint will be introduced. The experiment is presented in section 4, where we demonstrate that the proposed model can learn more effective features by comparing the decoding performance with that of using the existing features. Section 5 summaries the paper and presents our plan for future work.

2 Related Work

Our work is related to several recent works that incorporate prior knowledge of target variables. To reduce the reliance on labeled training data set, learning with uncertain labels [Lefort *et al.*, 2010] uses distribution over class labels for each data example to replace the exact label of data. A similar idea appears in semi-supervised learning where the proportion of different classes [Schapire *et al.*, 2002; Zhu, 2006] was used to predict the class labels on the uncorresponded training data examples. In generalized expectation(GE), the knowledge about class proportions conditioned on some features are used as additional information to learn the classifier [Mann and McCallum, 2007]. Domain knowledge about the target variables has also been used as constraints. For instance, constraint driven learning (CODL) [Chang *et al.*, 2007] learns word labeling model by taking advantage of language structures. CODL can be seen as a special case of posterior regularization [Ganchev *et al.*, 2010] with MAP approximation, which directly imposes regularization on the posterior of the latent target variables. A further generalization of these works that incorporates prior information as measurements in the Bayesian framework is proposed in [Liang *et al.*, 2009]. The recent work of [Wang *et al.*, 2012b] describes a weakly supervised learning framework that incorporates target priors into a regression framework, by which this work is particularly influenced.

Unsupervised pretraining of deep belief network was first described in [Hinton, 2000]. This is further extended to a convolutional deep belief network (CDBN) [Lee *et al.*, 2009], of which the number of parameters is independent from the data dimension and the training is more efficient. Stacked auto-encoder [Bengio *et al.*, 2007] is an alternative to the probabilistic deep models such as DBN or CDBN, and its major advantage is simple training algorithm. Deep learning has been adopted as a viable approach to unsupervised feature learning in several applications [Le *et al.*, 2011; Poon and Domingos, 2011]. However, most emphasis in deep feature learning is on efficient implementations from large unlabeled data sets [Le *et al.*, 2012], with a few exceptions on incorporating domain knowledge in the learning, e.g., [Rifai *et al.*, 2011]. While deep structures are most commonly pretrained with unsupervised learning, the partially supervised pre-training [Bengio *et al.*, 2007] has shown to be particularly important for regression problem. However, it requires labeled samples, which are difficult to obtain in many cases. More importantly, training with labeled data make the features less generalizable. In contrast, our work incorporates generic prior knowledge about target variables instead of labels into deep feature learning to achieve optimal tradeoff between the fidelity of data reconstruction and task prediction.

Deep learning has also been applied to BCI in several previous works. In [Freudenburg *et al.*, 2011], patterns learned from DBN are compared with those learned with PCA, and shown to have more correlations to the neuron patterns. The work of [Wulsin *et al.*, 2011] shows that DBN applied to classify the clinical EEG waveforms achieves comparable performance with the state-of-the-art based on SVM or kNN. Further, features learned by DBNs with raw brain signals as input are shown to have similar performance with hand crafted features in classifying sleeping state [Langkvist *et al.*, 2012].

3 Prior Supervised Convolutional Stacked Auto-Encoders (PCSA)

We describe the *prior supervised convolutional stacked auto-encoder* (PCSA) model in this section. PCSA is a deep learning model that combines convolutional stacked auto-encoders with the target prior, see Figure 1. The prediction of target variables from learned features are enforced to be consistent with the target prior. In this way, the target prior works as a weak supervisor for the feature learning. We first introduce Convolutional Stacked Auto-encoders (CSA) an its learning, then discuss learning CSA with the target prior.

3.1 Convolutional Stacked Auto-encoders

The convolutional stacked auto-encoder [Hadsell *et al.*, 2009] is a deep architecture with convolutional auto-encoder as the building block. In CSA, model parameters are shared among

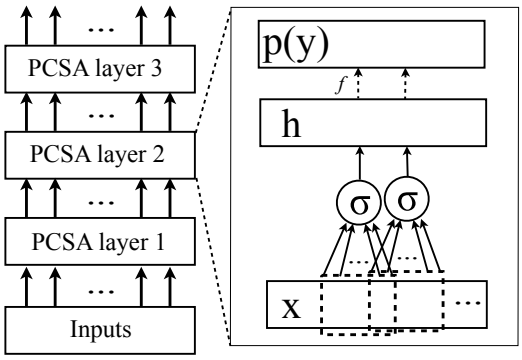

Figure 1: The schematic structure of PCSA

units in a convolution manner, which decouples the number of parameters in a deep model from data dimension and significantly improves the running efficiency [Lee *et al.*, 2009]. The general structure of CSA is shown schematically in Figure 1. Let us denote $\mathbf{X}$ to be the M dimensional time series, and the latent vector $\mathbf{h}^k$ represents the kth feature map. Given the input $\mathbf{X}$, $\mathbf{h}^k$ is computed as:

$$\mathbf{h}^k(t) = \sigma(\sum_{m=1}^{M} \mathbf{X}^m * \mathbf{W}^{mk}(t) + b^k), \tag{1}$$

where $\sigma(t) = 1/(1+e^{-t})$ is the sigmoid function, $*$ denotes the convolution operation and k is the index of feature map. The parameters to be estimated in one layer convolutional stacked auto-encoder (CSA) are $\mathbf{W}^{mk} = (w_1, \cdots, w_L)$, the convolution filter with length L, and $b^k \in \mathcal{R}$, the bias for the kth feature map. CSA has a multi-layer structure, which takes the single layer described above as building blocks and stacks these layers together with the output of a layer serving as the input of the layer above, figure 1. In testing, given raw input $\mathbf{X}$, the latent variable $\mathbf{h}$ is calculated recursively for each layer and those from the top layer are used to predict outputs by applying a linear regressor.

3.2 Learning CSA

An auto-encoder is trained so that it encodes the input $\mathbf{X}$ into another representation $\mathbf{h}$ and at the same time, the input $\mathbf{X}$ could be reconstructed from the representation $\mathbf{h}$. When the dimension of $\mathbf{h}$ is lower than that of $\mathbf{X}$, the latent layer $\mathbf{h}$ works as a code of the input $\mathbf{X}$, which is expected to capture important property of $\mathbf{X}$. The reconstruction step in CSA is given by

$$\hat{\mathbf{X}}^m = \sum_{k \in K} \mathbf{h}^k * \tilde{\mathbf{W}}^{mk} + c_m, \tag{2}$$

where $c_m \in \mathcal{R}$ is the bias. $\tilde{\mathbf{W}}^{mk} = (w_L, w_{L-1}, \cdots, w_1)$ is the flip version of $\mathbf{W}^{mk}$.

We train CSA in a layer-wise manner as in [Hinton, 2000]. For the sake of notational simplicity, we drop the superscripts indicating different layers of the model. The objective function corresponds to minimizing the mean square error between the input $\mathbf{X}$ and the reconstruction $\hat{\mathbf{X}}$ with regards to parameter $\theta = \{\mathbf{W}, b, c\}$, as:

$$E = \frac{1}{2MN} \sum_{m=1}^{M} \sum_{i=0}^{N-1} (\hat{\mathbf{X}}_i^m - \mathbf{X}_i^m)^2, \tag{3}$$

where N is the size of training data.

The reconstruction error alone is usually not sufficient to guarantee unique and non-trivial solution if no further requirement is used for the auto-encoder, as the identity function becomes a trivial solution with zero reconstruction error. Therefore, the reconstruction accuracy is combined with several other terms to form the CSA learning objective.

3.3 Learning CSA with Target Prior

While the conventional CSA learning ensures fidelity in data reconstruction, such learnt features may not perform well on a specific task. To address this issue, we propose to incorporate target prior into feature learning. In many applications, the target (output) variable under different inputs follows a common spatial and/or temporal pattern. For example, while people may walk differently, there are still certain common spatial and temporal pattern of the body pose that underpin all the walks and that differentiates walking from running. We intend to capture such common pattern probabilistically through the target prior and regularize the feature learning by making the prediction of the target variable consistent with the target prior [Wang *et al.*, 2012b]. In previous work [Wang *et al.*, 2011b; 2011a], prior knowledge has effectively improved the performance of BCI decoders. As shown in Figure 1, compared with the standard CSA, PCSA is additionally regularized by the target prior $p(\mathbf{y})$. We explain the details about target prior and its learning in the following.

Target Prior

We assume that the target variable $\mathbf{y}$ is predicted from the input variable $\mathbf{h}$ through a linear regression. Specifically:

$$\mathbf{y}_i = \sum_{k=1}^{K} \mathbf{V}_k f(\mathbf{h}_i^k) + d, \tag{4}$$

where $\mathbf{V} \in \mathcal{R}^{K \times 1}$ and $d \in \mathcal{R}$ are parameters of the regressor. The function $f(\mathbf{h})$ may vary with application. For example, in signal processing $f(h)$ may be power of h.

We assume the target variable $\mathbf{y}$ follows a prior probability distribution $p(\mathbf{y})$, which we propose to capture using the *Gaussian-Bernoulli restricted Boltzmann machine* (GB-RBM) [Hinton and Salakhutdinov, 2006]. We chose to use GB-RBM because of its ability to capture global spatial and temporal patterns. According to GB-RBM, the target prior can be formulated as: $p_\eta(\mathbf{y}) = \frac{1}{Z} \sum_{\bar{\mathbf{h}}} e^{-E_\eta(\mathbf{y}, \bar{\mathbf{h}})}$, where Z is the normalizing constant, and $\bar{\mathbf{h}} \in \{0,1\}^{\bar{\mathcal{H}}}$ are the hidden variables. The energy function over $\mathbf{y}$ and $\bar{\mathbf{h}}$ is defined, as:

$$E_\eta(\mathbf{y}, \bar{\mathbf{h}}) = \sum_{i=1}^{\mathcal{Y}} \frac{(\mathbf{y}_i - \mathbf{c}_i)^2}{2} - \sum_{i=1,j=1}^{\mathcal{Y},\bar{\mathcal{H}}} \mathbf{U}_{ij} \mathbf{y}_i \bar{\mathbf{h}}_j - \sum_{j=1}^{\bar{\mathcal{H}}} \mathbf{b}_j \bar{\mathbf{h}}_j. \tag{5}$$

where $\mathbf{U}_{ij}$ is the interaction strength between the hidden node $\bar{\mathbf{h}}_i$ and visible node $\mathbf{y}_j$. $\mathbf{c}$ and $\mathbf{b}$ are the bias for the visible

layer and hidden layer, respectively. The target variable $\mathbf{y}$ is normalized to have zero mean and unit standard variance. The parameters in this model, $(\mathbf{U}, \mathbf{c}, \mathbf{b})$, are collectively represented with η. Direct maximum likelihood training of GB-RBM is intractable for high dimensional models due to the normalizing factor Z, so we use contrastive divergence [Hinton, 2000] to estimate η from data.

Learning with Target Prior

After the parameters η in the GB-RBM target prior are learned from data, we use the corresponding target prior model to learn the CSA model parameters θ, as well as the parameters in the regressor $\mathbf{V}$ and d. The basis methodology is to enforce the regressor output to follow the target prior, as they reconstruct the inputs. After η is fixed, the normalizing factor Z is a constant, so we drop it off from the objective function. Thus, we set the target prior constraint as

$$C_1 = \log \tilde{p}_\eta(\mathbf{y}), \tag{6}$$

where $\tilde{p}_\eta(\mathbf{y}) = \log \sum_{\bar{\mathbf{h}}} e^{-E_\eta(\mathbf{y}, \bar{\mathbf{h}})}$. $\tilde{p}_\eta(\mathbf{y})$ can be efficiently obtained by noticing that the summation over hidden nodes $\bar{\mathbf{h}}$ can be factorized as the summation over each node $\bar{\mathbf{h}}_j$:

$$\begin{aligned} \log \tilde{p}_\eta(\mathbf{y}) &= -\frac{1}{2}\sum_i (\mathbf{y}_i - \mathbf{c}_i)^2 \\ &+ \log \sum_{\bar{\mathbf{h}}} e^{\sum_{i,j} \mathbf{y}_i \mathbf{U}_{i,j} \bar{\mathbf{h}}_j + \sum_j \mathbf{b}_j \bar{\mathbf{h}}_j} \\ &= -\frac{1}{2}\sum_i (\mathbf{y}_i - \mathbf{c}_i)^2 + \sum_j \log(1 + e^{b_j + \sum_i \mathbf{y}_i \mathbf{U}_{i,j}}). \end{aligned} \tag{7}$$

The derivative of $\log \tilde{p}_\eta(\mathbf{y})$ over $\mathbf{y}$ is given by

$$\frac{\partial \log \tilde{p}_\eta(\mathbf{y})}{\partial \mathbf{y}} = -(\mathbf{y} - \mathbf{c}) + \sum_j \frac{\mathbf{U}_{\cdot,j}}{1 + e^{-b_j - \sum_i \mathbf{U}_{i,j}\mathbf{y}_i}}, \tag{8}$$

where $\mathbf{U}_{\cdot,j}$ represents the jth column of $\mathbf{U}$. The derivative of C_1 over θ is obtained by $\frac{\partial \log \tilde{p}_\eta(\mathbf{y})}{\partial \theta} = \frac{\partial \log \tilde{p}_\eta(\mathbf{y})}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \theta}$.

Additional Constraints

We also require that the extracted features change slowly, which is inspired by the work of slow feature analysis [Wiskott and Sejnowski, 2002]. Specifically, we require that the extracted features vary slowly over time by minimizing the magnitude of the temporal gradient:

$$C_2 = \sqrt{\frac{1}{2(N-1)} \sum_{k=1}^{K} \sum_{i=0}^{N-2} [f(\mathbf{h}_{i+1}^k) - f(\mathbf{h}_i^k)]^2}.$$

To further ensure that the filters $\mathbf{W}^1, ..., \mathbf{W}^K$ are less correlated and correspond to distinct features, we also minimize:

$$C_3 = \sqrt{\frac{1}{4M(K^2 - K)} \sum_{m=1}^{M} \sum_{i \neq j} [(\mathbf{W}^{mi})^T \mathbf{W}^{mj}]^2}. \tag{9}$$

Last, to reduce overfitting we apply ℓ_2 constraint to the filters to normalize their magnitudes:

$$C_4 = \frac{1}{2} \sum_{m=1}^{M} \sum_{i=1}^{K} (\mathbf{W}^{mi})^T \mathbf{W}^{mi}. \tag{10}$$

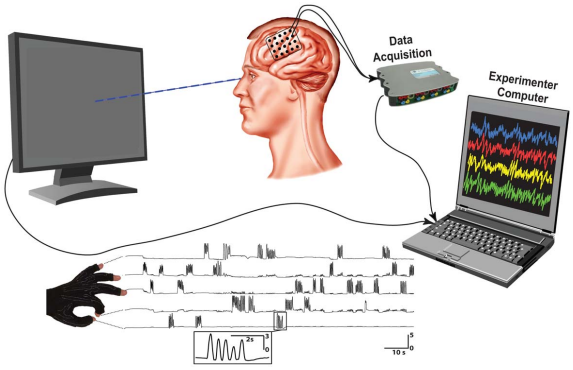

Figure 2: Experiment setup for this study. The subject flexes fingers according to visual cues on the screen. The finger trace is recorded through a data glove and the ECoG is collected through electrode grid placed over the fronto-parietal-temporal region.

Optimal parameters are learned by minimizing the reconstruction error with regards to these constraints (C1, C2, C3 and C4). Mathematically, this is equivalent to the minimization of the Lagrangian formed with the objective function and constraints, as:

$$F = E + \alpha C_1 + \beta C_2 + \gamma C_3 + \delta C_4 \tag{11}$$

where α, β, γ and δ are the regularization hyperparameters. These hyperparameters are estimated using grid search (details are discussed in section 4.2). The objective function is minimized numerically with conjugate gradient descent.

4 Experiments

We apply PCSA to the task of predicting finger flexion traces from ECoG signals in BCI, as illustrated in Figure 2. During the experiment, the subjects were asked to repeatedly flex and extend specific individual fingers according to visual cues that were given on a video screen. The ECoG signals and finger flexion traces were recorded simultaneously. Data used in our study were collected from five subjects, identified as subjects A, B, C, D and E.

We compare the features learned with PCSA with the band power features in terms of decoding performance. The band power features is state of the art features for finger movement decoding in BCI. It is used by the winner of the BCI competition IV [Bougrain and Liang, 2009]. These features are manually chosen for predicting finger movement from ECoG signals. To quantitatively evaluate the performance, we use the data set with corresponded ECoG signals and finger traces. However, during the training of PCSA, we intentionally leave out the labels (the corresponded finger traces). Thus, PCSA also has the potential to be applied to most real BCI applications, in which subjects imagine without actual body movements. Note that the experiment setting for PCSA is a little different from that in [Bougrain and Liang, 2009]. To simply the prior model training, here we consider the moving traces only composed of flexion and extension as in Fig. 3(A), and the rest part is not included. This simplified model is still

practically useful since we can first classify the trace into movement state or rest state and then apply the corresponding regressor for each state [Flamary and Rakotomamonjy, 2009].

4.1 Learning Target Prior Model

The target prior model $p_\eta(\mathbf{y})$ for each subject is learned from the finger traces of other subjects. To effectively capture the trace patterns with limited training data and reduce model parameters, the model is trained on the trace that is down sampled from the original trace by a factor of 25. The down sampled trace still keeps the original finger movement patterns. Accordingly, the target prior constraint will be applied to a subset of the inputs. Under this setting, each subject has around 2400 samples. We model the finger movement trace using the GB-RBM with 64 hidden nodes and 16 visible nodes, which is approximately the length of one round extension and flexion. Then, all segments from 16 successive samples in the data are used to train the prior model.

The GB-RBM is trained with stochastic gradient decent with a mini-batch size of 25 sub-sequences. We run 100 epochs with a fixed learning rate 0.001. We validate the prior model by drawing samples from the learned GB-RBM. Figure 3(B) shows 4 samples, which seem to capture some important properties of the temporal dynamics of the trace.

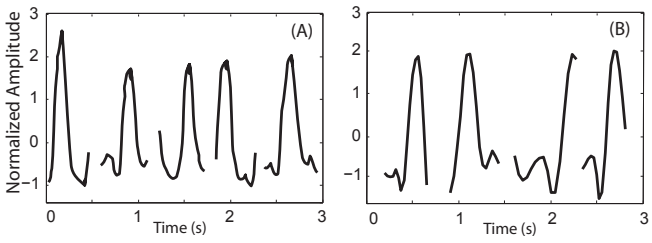

Figure 3: (A) Examples of training trace sub-sequences; (B)4 trace samples from GB-RBM. Each sample is a segment of length 16 data samples.

4.2 Results

The experiment are done on the thumb of five subjects. For each subject, the dataset consists of about 60,000 samples. In the following experiments, we will use five fold cross validation for the training and testing. The parameters $\mathbf{W}$, b, c, $\mathbf{V}$ and d in the model are randomly initialized. The length of each filter is set as $L = 64$ (samples). The input ECoG signals are normalized to zero mean and unit variance for each dimension, and the output of each layer are also normalized in the same manner to serve as the input of the layer above. Here we assume that $\mathbf{y}$ is predicted from the physical power [Bougrain and Liang, 2009] of latent variable $\mathbf{h}$. Thus, we set $f(h_i^k) = (h_i^k - a^k)^2$ in Eq.4, where $a^k = \frac{1}{N}\sum_{j=0}^{N-1} h_j^k$ is the mean or bias of the kth feature map. For the PCSA model, if we take all 48-64 channels as the input, the resulting model will have a large number of parameters that makes its training very inefficient. So we choose five most informative channels decided by linear regression as the input of PCSA. The five channels are selected in a incremental manner so that the combination can achieve the best performance. The band power features are tested based on both all the channels and the same five channels used by PCSA model. Also, to balance the decoding accuracy and computational load, in each layer we use 3 filters for each input dimension. It takes two to three minutes to get the parameter trained.

In eq.(11), the hyperparameters α, β, γ and δ are identified through a grid search. For each set of hyperparameters, we evaluate the model performance by training the model parameters with the hyperparameters fixed. The model is trained with two fold cross validation on the validation data, that is, during the tuning of hyperparameters half the training data are used for training and the remaining are used for validation. These hyperparameters are fixed for testing. To speed up the grid search, the conjugate gradient optimization stops when the number of iterations reach 10. The initial search is in the range $[0, 10]$ with the step 2 for each hyperparameter. Then it is followed by a fine search around the chosen hyperparameters with the step size 0.2 within the range of 1 for each hyperparameter. It takes two to three hours to get the hyperparameters tuned. For the

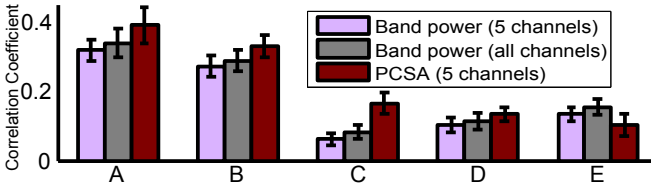

Figure 4: Comparison among PCSA and band power features(averaged across five folds).

PCSA model, both the filters and linear regression parameters (as the variable $\mathbf{V}$ and d in eq.4) are learned simultaneously. For the band power features, we also use linear regression to predict the finger movement. Linear regression has been commonly used for such tasks in BCI though PCSA can also work with nonlinear regression methods. We measure the performance with correlation coefficient, which has been used in the previous work [Kubánek *et al.*, 2009; Bougrain and Liang, 2009]. The features are extracted from the third layer of PCSA model. The comparison results are shown in figure 4. For each subject, we show the averaged performance over five folds. The results show that the learned features by PCSA significantly outperform the band power features ($p < 0.05$, paired t-test on the correlation coefficient for all fingers and subjects and between PCSA and band power features). It comes as an initial surprise that PCSA does not work well on subject E. After examining the data, we found that subject E moves fingers much slower than other subjects, on which the prior model is trained, and thus its properties cannot be captured well with the target prior model. This suggests that the quality of prior model has an important influence on the performance of the final regression.. Figure 5 gives an intuitive comparison for the sample traces predicted by two sets of features on the thumb of subject A. The results show that the ground truth can be better fitted with the trace predicted by learned features than those obtained with the band power features.

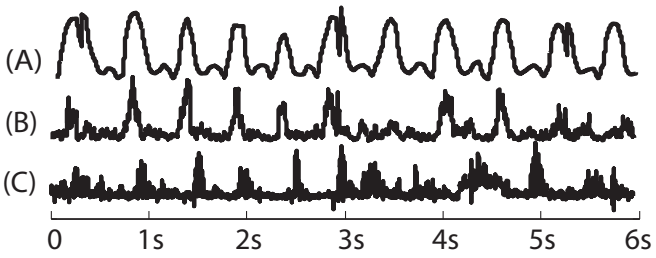

Figure 5: The sample trace predicted by PCSA and band power features. (A) ground truth; (B) prediction by PCSA (correlation coefficient 0.44); (C) prediction by band power features (correlation coefficient 0.30)

4.3 Comparison with Partially Supervised CSA

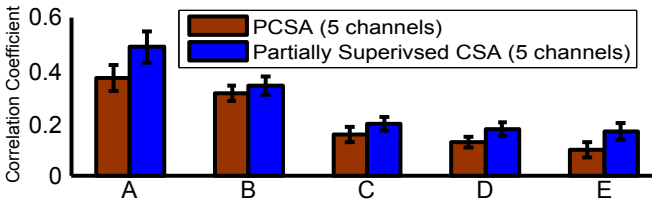

Figure 6: Performance comparison between PCSA and partially supervised CSA

In the setting of partially supervised learning [Bengio *et al.*, 2007], the model is learned in the way that not only minimizes the reconstruction error but also minimizes the prediction error. Actually partially supervised learning replaces the target prior constraint of PCSA in Eq. 6 with a loss function which measures the deviation of the regression outputs from the ground truth. Here we adopt the loss function in the form of mean square error (MSE), i.e., $\frac{1}{2N}\sum_{i=0}^{N-1}[y_i - z_i]^2$, where z is the ground truth. As expected with stronger information the results of partially supervised learning in Figure 6 outperform the PCSA. Although PCSA does not perform as well as

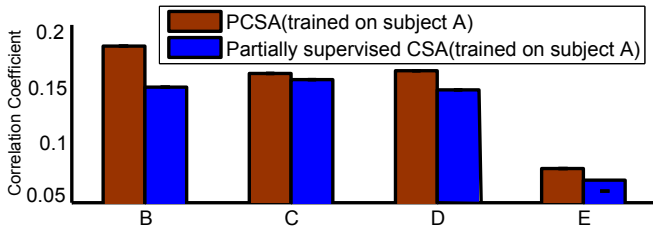

Figure 7: The comparison of generalization across subjects between PCSA and partially supervised CSA

partially partially supervised CSA when trained and tested on the same subject, it tends to have better generalization across subjects. Figure 7 gives the results of PCSA and partially supervised CSA when trained on subject A and tested on subjects B, C, D and E. Clearly PCSA has better performance, and the results are similar when trained on other subject and tested on the remaining subjects. We believe that the better generalization of PCSA results from the learning with more generalizable prior information instead of exact labels.

4.4 Effects of the Target Prior Constraint

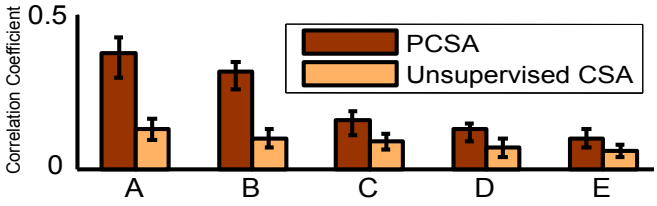

Figure 8: Performance evaluation of PCSA with and without the target prior constraint

We consider the effect of the target prior constraint by dropping it from the objective function and see how the overall performance changes. Without the target prior constraint, the remaining parts form an unsupervised feature learning model. Figure 8 shows the results. We can see that without the target prior constraint, the performance reduces dramatically. Thus, the target prior constraint has a significant contribution to the overall decoding performance.

5 Conclusion

In this paper, we describe a new deep feature learning method, which we term as prior supervised convolutional stacked auto-encoders (PCSA), that combines the learning with target prior with deep feature learning. In PCSA, the statistical properties in the target variables are captured with a probabilistic target prior model. The target prior is then used to regularize the unsupervised learning objective function to produce features that are general to represent input raw data and at the same time effective for the supervised learning task. We apply the method to the BCI problem of decoding finger flexion from ECoG signals. The results show that the features learned by PCSA achieved better performance than state of the art hand features obtained based on heuristics. We further show that PCSA has better generalization that partially supervised CSA, and the target prior has a significant effect in the feature learning of this task.

We would like to extend the current work in several directions. First, in current work we use a simple target prior model in the form of GB-RBM. More flexible probabilistic models, such as Markov random fields and dynamic Bayesian network, can better represent statistical properties in the target variables. Therefore, we would like to incorporate such models into deep learning to further improve performance. Second, We are also interested in extending this framework to feature learning over other high dimensional signals such as images and videos.

Acknowledgments

US Army Research office (W911NF-08-1-0216 (GS)) and NSF CAREER Award (IIS-0953373).

References

[Bengio *et al.*, 2007] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, Universit De Montral, and Montral Qubec. Greedy layer-wise training of deep networks. In *NIPS*. MIT Press, 2007.

[Bougrain and Liang, 2009] Laurent Bougrain and Nanying Liang. Band-specific features improve Finger Flexion Prediction from ECoG. In *JAICC*, Paranà, Argentine, 2009.

[Chang *et al.*, 2007] Mingwei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.

[Flamary and Rakotomamonjy, 2009] Rémi Flamary and Alain Rakotomamonjy. Decoding finger movements from ECoG signals using switching linear models. Technical report, September 2009.

[Freudenburg *et al.*, 2011] Zachary V. Freudenburg, Nicolas F. Ramsey, Mark Wronkeiwicz, William D. Smart, Robert Pless, and Eric C. Leuthardt. Real-time naive learning of neural correlates in ECoG Electrophysiology. *Int. Journal of Machine Learning and Computing*, 2011.

[Ganchev *et al.*, 2010] Kuzman Ganchev, João Graca, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, pages 2001–2049, 2010.

[Hadsell *et al.*, 2009] Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *J. Field Robot.*, 26(2):120–144, February 2009.

[Hinton and Salakhutdinov, 2006] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[Hinton, 2000] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002, 2000.

[Kubánek *et al.*, 2009] J Kubánek, K J Miller, J G Ojemann, J R Wolpaw, and G Schalk. Decoding flexion of individual fingers using electrocorticographic signals in humans. *J Neural Eng*, 6(6):066001–066001, Dec 2009.

[Langkvist *et al.*, 2012] Martin Langkvist, Lars Karlsson, and Amy Loutfi. Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, 2012.

[Le *et al.*, 2011] Quoc V. Le, Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368, 2011.

[Le *et al.*, 2012] Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012.

[Lecun *et al.*, 1998] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[Lee *et al.*, 2009] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, pages 609–616, 2009.

[Lefort *et al.*, 2010] Riwal Lefort, Ronan Fablet, and Jean-Marc Boucher. Weakly supervised classification of objects in images using soft random forests. In *ECCV*, 2010.

[Liang and Bougrain, 2009] Nanying Liang and Laurent Bougrain. Decoding finger flexion using amplitude modulation from band-specific ECoG. *Neural Networks*, (April):22–24, 2009.

[Liang *et al.*, 2009] Percy Liang, Michael I. Jordan, and Dan Klein. Learning from measurements in exponential families. In *ICML '09*, pages 641–648, NY, USA, 2009.

[Mann and McCallum, 2007] Gideon S. Mann and Andrew McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*, 2007.

[McFarland and Wolpaw, 2008] D.J. McFarland and J.R. Wolpaw. Brain-computer interface operation of robotic and prosthetic devices. *Computer*, (10), oct. 2008.

[Mohamed *et al.*, 2011] A. Mohamed, G. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Trans. on*, PP(99):1, 2011.

[Nair and Hinton, 2009] Vinod Nair and Geoffrey Hinton. 3D Object Recognition with Deep Belief Nets. In *NIPS*, pages 1339–1347. 2009.

[Poon and Domingos, 2011] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *UAI*, pages 337–346, 2011.

[Rifai *et al.*, 2011] Salah Rifai, Yann Dauphin, Pascal Vincent, Yoshua Bengio, and Xavier Muller. The manifold tangent classifier. In *NIPS*, pages 2294–2302, 2011.

[Schapire *et al.*, 2002] Robert E. Schapire, Marie Rochery, Mazin G. Rahim, and Narendra Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.

[Wang *et al.*, 2010] Zuoguan Wang, Qiang Ji, Kai J. Miller, and Gerwin Schalk. Decoding finger flexion from electrocorticographic signals using a sparse gaussian process. In *ICPR*, pages 3756–3759, 2010.

[Wang *et al.*, 2011a] Zuoguan Wang, qiang ji, Kai J Miller, and Gerwin Schalk. Prior knowledge improves decoding of finger flexion from electrocorticographic (ECoG) signals. *Frontiers in Neuroscience*, 5(127), 2011.

[Wang *et al.*, 2011b] Zuoguan Wang, Gerwin Schalk, and Qiang Ji. Anatomically constrained decoding of finger flexion from electrocorticographic signals. *NIPS*, 2011.

[Wang *et al.*, 2012a] Z. Wang, Aysegul Gunduz, Peter Brunner, A.L. Ritaccio, Q Ji, and Gerwin Schalk. Decoding onset and direction of movements using electrocorticographic (ECoG) signals in humans. *Frontiers in Neuroengineering*, 5, 2012.

[Wang *et al.*, 2012b] Zuoguan Wang, Siwei Lyu, Gerwin Schalk, and Qiang Ji. Learning with target prior. In *NIPS*, 2012.

[Wiskott and Sejnowski, 2002] Laurenz Wiskott and Terrence J. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Comput.*, April 2002.

[Wulsin *et al.*, 2011] D F Wulsin, J R Gupta, R Mani, J A Blanco, and B Litt. Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement. *Journal of Neural Engineering*, 8(3):036015, 2011.

[Zhu, 2006] Xiaojin Zhu. Semi-supervised learning literature survey, 2006.