DEEP LEARNING FOR BRAIN DECODING

Orhan Firat[⋆] *Ilke Oztekin*[†] *Fatos T. Yarman Vural*[⋆]

[⋆] Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
{*orhan.firat,vural*} @*ceng.metu.edu.tr*
[†]Department of Psychology, Koc University, Istanbul, Turkey
ioztekin@ku.edu.tr

ABSTRACT

Learning low dimensional embedding spaces (manifolds) for efficient feature representation is crucial for complex and high dimensional input spaces. Functional magnetic resonance imaging (fMRI) produces high dimensional input data and with a less then ideal number of labeled samples for a classification task. In this study, we explore deep learning methods for fMRI classification tasks in order to reduce dimensions of feature space, along with improving classification performance for brain decoding. We employ sparse autoencoders for unsupervised feature learning, leveraging unlabeled fMRI data to learn efficient, non-linear representations as the building blocks of a deep learning architecture by stacking them. Proposed method is tested on a memory encoding/retrieval experiment with ten classes. The results support the efficiency compared to the baseline multi-voxel pattern analysis techniques.

Index Terms— Deep Learning, Stacked Autoencoders, fMRI, MVPA, brain state decoding

1. INTRODUCTION

Multi-voxel Pattern Analysis (MVPA) methods have become increasingly popular in the analysis of functional Magnetic Resonance Imaging (fMRI) data [1–4]. These methods are employed in many problems, such as hypothesis validation [5], diagnosing disorders [4, 6] and recently brain state decoding, also known as mind reading [1, 7]. The major difficulty of using the fMRI data for pattern analysis is the deficiency of the labeled samples in comparison to the high dimensional feature representations. The classical pattern recognition techniques are prone to several problems, such as over-fitting, insignificance and curse-of-dimensionality. In order to overcome the limitations of the classical MVPA methods, the fMRI data is restricted *a priori* to a subset of voxels which obviously results in losing several relational information-channels (relations between voxels) [5–8]. The trade-off between the restriction on number of voxels (region of interest selection) and the need to employ voxels from whole brain (even redundant and un-informative ones) may result in misleading analyses about the nature of the brain states. The shortcomings of the available techniques necessitates us to study approaches that are robust to the sparsity and information integrity for feature representations.

It is well-established that deep learning architectures resemble some similarities with the human brain in terms of scalability. Recent studies on unsupervised feature learning and deep learning propose an alternative to the classical MVPA techniques by making use of the unlabeled data and learning a feature hierarchy automatically [9–11]. Unsupervised feature learning leverages unlabeled data to learn basic patterns (regularities) which can be used in forthcoming supervised tasks [12, 13]. On the other hand, deep learning constructs several layers of features where higher levels capture more abstract forms of variations in data [14]. By combining the deep learning architecture with the unsupervised feature learning backbone, it is possible to discover a low-dimensional feature space. This manifold space can then be used to map high dimensional input data [15, 16]. Deep learning is a rapidly growing area of AI and recently emerging for computational medical imaging [17–20]. A stacked autoencoder (SAE) is used in [19], for organ detection using 4D data by making use of spatial and temporal filters which are learned automatically. A similar SAE is used in [18] to learn a joint feature representation using for MRI, PET and CSF for Alzheimer's disease detection. In [20], a low-dimensional manifold is learned using a stacked convolutional restricted Boltzmann machine for Alzheiher's detection using structural MRI. Image segmentation for MRI is subjected in [17] by a stacked convolutional independent subspace analysis network.

In this study, we propose to model the 4-dimensional spatio-temporal fMRI data by the features extracted from the unsupervised feature learning method. Then, we explore the deep learning openings for better discrimination of cognitive processes. We conduct a memory encoding and retrieval experiment with ten semantic categories. Then, we use the unlabeled fMRI data for unsupervised learning representations along with a stacked sparse-autoencoder to reduce

Thanks to TUBITAK agency for funding with grant no:112E315.

dimensionality for brain state classification.

2. AUTOENCODERS FOR UNSUPERVISED FEATURE LEARNING

An autoencoder is a type of neural network which reconstructs its input by setting the target values to be equal to the inputs, $x \approx \bar{x}$. An autoencoder consists of two consecutive functions. The first one is an encoder function $f_{\theta_1}(x)$ applied on input data x with parameters $\theta_1 = \{W^{(1)}, b^{(1)}\}$ (transition and bias respectively). This function maps the input data to a hidden representation h. The second function is a decoder function $g_{\theta_2}(f_{\theta_1}(x))$ which maps the hidden representations to a reconstruction $\tilde{x}$ of input, parametrized by $\theta_2 = \{W^{(2)}, b^{(2)}\}$. Note that, encoding and decoding functions are mappings, such that, $f : \mathbb{R}^N \rightarrow \mathbb{R}^K$ and $g : \mathbb{R}^K \rightarrow \mathbb{R}^N$, where N is the input dimensionality and K is the number of neurons in the hidden layer. By enforcing some constraints, (a sparsity [21] or contracting term [22] regarding to hidden layer activations or distorting input respectively [23]) the autoencoder learns a compact and non-linear representation of the input. Therefore, it keeps away from learning an identity function.

In this study, we focus on sparse autoencoders having single hidden layers in the encoding/decoding functions by enforcing a sparsity (activation around zero) to hidden unit neurons via the cost function. After learning model parameters $\Theta = \{\theta_1, \theta_2\}$, the sparse autoencoder learns the non-linear feature mapping function f which can be used in further classification tasks or feeding the input layer of another autoencoder.

A sparse autoencoder having K hidden neurons, is trained in order to minimize squared reconstruction error using back-propagation by minimizing the following cost function,

$$J_{sparse}(\Theta) = J_{NN}(\Theta) + \beta J_{\hat{\rho}} \qquad (1)$$

where $J_{NN}(\Theta)$ being regularized neural network cost and $J_{\hat{\rho}}$ regarding to sparsity term. Hyper-parameter β controls the importance of sparsity in the model. In the general, $L2$ regularized neural network (having 1 hidden layer) cost with m examples is as follows,

$$J_{NN}(\Theta) = \frac{1}{2m}\sum_{i}^{m}\left\|\tilde{x}^{(i)} - x^{(i)}\right\|^2 + \frac{\lambda}{2}\sum_{l,u,v}\left(W_{uv}^{(l)}\right)^2, \qquad (2)$$

where first term corresponds to the squared reconstruction error and the second term corresponds to the weight decay term that penalizes large values of transition parameters for all entries u, v. The activation of the each layer l is computed as $a^{(l)} = \sigma(W^{(l-1)}a^{(l-1)} + b^{(l-1)})$ where λ is the regularization parameter, σ is the sigmoid function and $\|.\|$ indicates $L2$ norm and $a^{(0)} = x$. Note that, $f_{\theta_1}(x) = a^{(1)}$ in our model. Further, let

$$\hat{\rho}_j = \frac{1}{m}\sum_{i} a_j^{(2)}(x^{(i)}), \qquad (3)$$

be the average activation of hidden unit j over the dataset and enforce the constraint $\hat{\rho}_j = \rho$ where ρ is the sparsity parameter (chosen to be close to zero). In order to measure the sparsity cost, Kullback-Leibler divergence between average activation of a unit $\hat{\rho}$ and sparsity parameters ρ is employed. Finally, sparsity term of the overall cost function in (1) is given as follows:

$$J_{\hat{\rho}} = \sum_{j}^{K} KL(\rho\|\hat{\rho}_j) = \sum_{j}^{K} \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \qquad (4)$$

Optimization of the model parameters can be achieved by gradient descent. Since the objective gradient can be computed exactly, it is plausible to use advanced optimization methods. For this reason, we employ a quasi-newton L-BFGS method. Note that, for the inputs that are not in $[0 - 1]$ range, output layer activations can be set directly as the weighted sum of its pre-activations without passing through a point-wise sigmoid function, meaning $a^{(2)} = g_{\theta_2}(f_{\theta_1}(x)) = W^{(1)}a^{(1)} + b^{(1)}$.

3. STACKED AUTOENCODERS FOR DEEP LEARNING REPRESENTATIONS

Given the unlabeled fMRI data, the autoencoder learns non-linear intermediate representations (codes) to reconstruct the data. In order to achieve a better reconstruction, representations must specifically capture the necessary variations to distinguish examples in the input data. This constitutes the directions of variations on the manifold where the probability mass (the data generating distribution) concentrates [14]. It has also been shown that highly complex functions can be represented with fewer parameters through the composition of many non-linearities, which is a deep architecture as opposed to the shallow ones [9, 15]. Hence by stacking autoencoders (or any non-linear feature mappings), more abstract and complex representations are achieved which also makes the data generating distribution more uniform manifolds. The rationale of unfolding the manifold in representation space is that, it is the most efficient way to represent the information where linear perturbations in higher levels will still move near manifold [16,24]. This process also helps to reduce dimensionality of the input because we will need less and less manifold coordinates to span an unfolded manifold as deeper as our model.

In this study we employed stacked sparse autoencoders to learn a low-dimensional non-linear feature representations for fMRI pattern classification. The sparsity term and number of neurons in each layer of stacked autoencoder, are selected to lower the representation dimension without hurting the reconstruction error drastically. In order to obtain a stacked autoencoder, we employed greedy layer-wise pre-training [12, 25] where we trained one layer at a time. That is, we first train the first level autoencoder by using training data as input, and then train the second level autoencoder with the outputs from the first layer autoencoder encoding function as input and so on. Note that, pre-training is performed in an unsupervised fashion. We can formalize a stacked autoencoder as follows,

$$h^L = f_{\theta_1}^{(L)}(\cdots f_{\theta_1}^{(2)}(f_{\theta_1}^{(1)}(x))), \tag{5}$$

where we discard decoder functions $g(\cdot)$ as we finish training the autoencoder, starting from the very first layer using input data. Final representation (manifold coordinates) can be obtained by the encoder function of the last layer L with the hidden representations h^L.

4. CLASSIFICATION OF BRAIN STATES USING DEEP REPRESENTATIONS

To improve classification performance, deep learning architectures can be fine-tuned for discrimination which is called supervised *fine-tuning* [25]. Final fine-tuning phase makes all the parameters $\{\theta_1^{(i)}\}_{i=1}^{L}$ tuned for the supervised task at hand, by modifying representations slightly to get the category boundaries right. This can be achieved easily by unfolding the stacked autoencoder into a multi-layer neural network (initializing the weights) and adding a fully connected softmax layer on top where the number of neurons in the softmax layer is equal to the number of classes to be discriminated. It is then straight forward to optimize the network by backpropagation with gradient descent using the error derivatives of the top-most softmax layer. As expected, fine-tuning phase necessitates labelled data, but due to the pre-training phase, it does not need to discover new features, which reduces the need for a high number of labelled samples for the discriminative phase. It has also been shown that this type of backpropagation works well even if most of the data is unlabelled [13], which might often be the case for fMRI pattern classification tasks.

5. EXPERIMENTS ON FMRI DATA REPRESENTATION AND CLASSIFICATION

fMRI data is composed of 3-dimensional brain volumes across time $\{t_i\}_{i=1}^{n}$, where each 3D volume is formed by stacking several 2D scans (slices). Each pixel in these 2D images are actually represents the intensity of a small volume of brain tissue (voxel) at a time instant t_i. A typical fMRI experiment consists of several runs, where in each run the subject is exposed to some task specific stimuli $\{c_j\}_{j=1}^{S}$ at the predefined time instants, where S is the total number of semantic classes in the experiment and n is the total length of the experiment across runs (see Figure 1). The problem arises for classification tasks here because within a large amount of samples across time, only few have assigned class labels. The rest of the samples that are not having a class label t_{i/c_j} are simply discarded. The unsupervised feature learning motivation in this study is rooted with this fact; namely, by making use of the discarded data, which might be as high as ten times more, compared to labelled samples in a typical experiment.

5.1. Data and Preprocessing

In the current study, fMRI recording was conducted during a recognition memory task. Each participant is shown a list of words belonging to a specified category in the encoding phase (e.g. fruits or tools). Following a delay period where the participant solves mathematical problems, a test probe is presented and the participant executes a yes/no response indicating whether the word belongs to the current study list (e.g., see [5]). For the classification task we focused on lateral temporal cortex region having 8142 voxels. fMRI data consists of 2400 time points with 240 class labels for the encoding phase and 240 class labels for the retrieval phase (for each of ten classes, 24 samples are obtained for both encoding and retrieval).

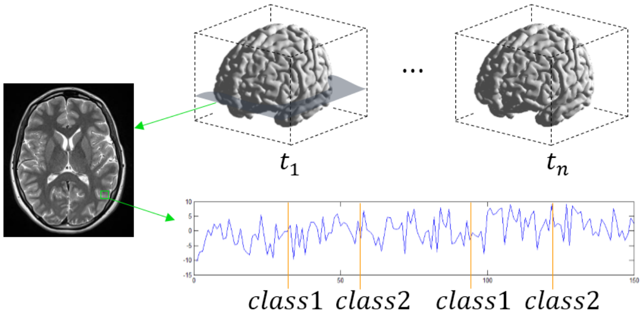

Fig. 1. Example of a typical fMRI experiment for brain state decoding. 4D fMRI data consists of several volumes across time, some of which have assigned to a class labels(indicated by vertical lines in time axis)

5.2. Classification of Brain States

The classification task we try to accomplish is to predict class labels of the samples in the retrieval phase by using samples in the encoding phase. Measurements recorded in the encoding phase are used as labelled training samples and measurements in the retrieval phase are used as test samples. The critical component in this supervised classification task is the way we incorporated the unlabelled samples. All the unlabelled samples along with labelled training samples of encoding phase are used in unsupervised feature learning with a stacked autoencoder. After learning a high level representation, labelled data is employed to either fine tune the model or directly to classify using the last layer activation of stacked autoencoder as described above.

5.3. Testing Procedures

In order to assess the applicability of the deep learning methods on the fMRI data, we initially used a small subset of measurements (100 voxels) and trained a stacked autoencoder up to two levels. Next, we employed a larger set of voxels (1024) stacked up to three levels. In both settings, we compared the results with the state of the art MVPA methods. The first method used for comparison, is the classical approach where voxel intensity values are directly fed into classifiers (indicated as MVPA). The second method is a local-linear model, where each voxel is represented in the feature space with the linear regression weights estimated using its surrounding voxels (indicated as MAD) [26]. For the unsupervised feature

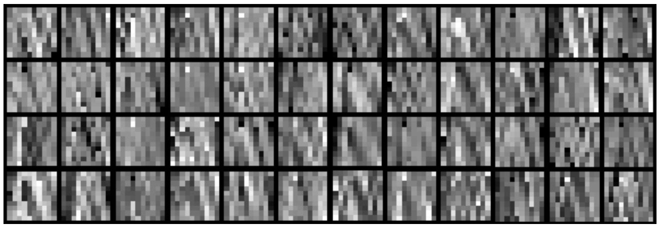

Fig. 2. A subset of first layer filters learned using sparse autoencoder for 100 voxels in a 2D slice. Each square box above corresponds to a filter that is sensitive for a specific activation pattern of voxels. A general wedge shape activation pattern with varying intensity levels is observed along diagonals.)

learning phase, we employed all measurements (2160), excluding the retrieval samples to be used for the test phase. For sparse auto-encoders, untied weights ($\theta_1 \neq \theta_2^T$) were used in all layers as it gives better performance than tied weights in our experiments. In order to reduce dimensions in the top-most layer representations, we gradually decreased the number of neurons in each layer and reduce the β parameter that controls the importance of sparsity. For the fine-tuning phase, we attached a softmax layer with 10 units on top. For comparison with baseline MVPA methods, penultimate layer activations were fed to a k-Nearest Neighbor (knn) classifier. Model parameters and hyper-parameters were empirically selected on a held-out cross validation set. For implementations of the methods mentioned above, we used libORF (http://www.ceng.metu.edu.tr/∼e1697481/libORF.html)

6. RESULTS

In order to qualitatively assess the validity of the unsupervised feature learning phase, we visualized the first layer filters learned for stacked autoencoders with 100 voxel experiment. The learned filters are illustrated in Figure 2. It is expected to observe reasonable activation patterns for these 100 voxels since they reside in a spatially connected space (nearest voxels in a 2D-slice). Plausible patterns should resemble an fMRI alike intensity map as illustrated in Figure 2. It can be deduced that, the selected 100 voxels have various activation patterns that exhibit a wedge like pattern with various intensity levels along diagonal of learned filters. Learned filters are also very similar to actual fMRI 2D-slices when we zoom in and compare with the slice in Figure 1. As the aim of unsupervised feature learning is to capture a codeword for fMRI data in a lower dimensional space (manifold), we can conclude that it is promising to use the activation patterns learned in this phase to construct a feature space for the forthcoming classification task.

Method	100 voxels		1024 voxels		Classifier
Employed	Acc %	Dim	Acc %	Dim	
MVPA	44.83	100	58.62	1024	knn
MAD [26]	48.28	400	65.52	19456	knn
NLFM1	44.83	50	-	-	softmax
NLFM1*	48.28	50	58.62	500	softmax
NLFM1*	**51.72**	50	65.52	500	knn
NLFM2	**58.62**	49	**68.97**	450	knn
NLFM3	-	-	**72.41**	150	knn

Table 1. Classification performances of Non-Linear Feature Mapping (NLFM) with classical methods for fMRI ML tasks (MVPA) and a local-linear feature extraction method (MAD). Feature dimensions are indicated in Dim columns.

Pattern analysis results were evaluated by considering final feature space dimensions in classifiers by comparing classification accuracy, shown in Table 1. Proposed non-linear feature mapping (NLFM) method using stacked sparse autoencoders is tested with varying depth and fine-tuning options (indicated with a * in Table 1). A shallow feature learning with a single layer autoencoder (NLFM1) is tested initially which gives almost similar performance with the baseline methods (MVPA,MAD), by reducing the dimension by half, which is expected as the non-linear mapping compared to the lieear ones. We further reduced the dimension by stacking more layers up to 150 from 1024 for the second experiment where we consider 1024 voxels. Classification performances gradually increase as we stack up additional layers which can be seen in Table 1 up to 59% and 72% respectively for 100 and 1024 voxels experiments. We also observed a notable performance increase with fine-tuning in our experiments. Compared to the baseline methods we observed a drastic reduction in the feature dimensions (150 compared to 1024 of MVPA and 19456 for MAD), with improving performance. Increased dimensions in the feature space is prone to over-fitting in fMRI classification tasks when we consider the scantiness of labelled data. As we increase the feature dimensions the gain in the generalization performance gets stuck up to a level due to over-fitting where curse-of-dimensionality arises with linear models. The results show that multi-layer non-linear feature mapping methods are promising for high dimensional input spaces with a low number of labelled samples by leveraging unlabelled data.

7. CONCLUSION

In this investigation, we modeled the neural activity using non-linear feature mapping strategies. We explored the the multiple layers of autoencoders in a deep learning framework. Stacked sparse autoencoders are used in order to learn a non-linear feature hierarchy with using the unlabeled fMRI data. The manifold assumption is further exploited by reducing dimensions as we stack up layers, resulting in a low dimensional feature space that is further used as the feature space for two fMRI classification tasks. In addition to being able to efficiently reducing dimensions, results indicated an improved performance compared to the baseline methods. Future work can target application of the proposed method to whole brain data for brain decoding using 3D convolutional models. We further suggest to employing manifold tangent based classification regimes for k- Nearest Neighbor methods to further improve classification performance.

8. REFERENCES

[1] K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby, "Beyond mind-reading: multi-voxel pattern analysis of fMRI data." *Trends in cognitive sciences*, vol. 10, 2006.

[2] B. Chai, D. B. Walther, D. M. Beck, and F.-F. L., "Exploring Functional Connectivity of the Human Brain using Multivariate Information Analysis," *Advances in neural information processing systems*, 2009.

[3] T. Schmah, G. Yourganov, R. S. Zemel, G. E. Hinton, S. L. Small, and S. C. Strother, "Comparing classification methods for longitudinal fMRI studies." *Neural computation*, vol. 22, 2010.

[4] M. N. Coutanche, S. L. Thompson-Schill, and R. T. Schultz, "Multi-voxel pattern analysis of fMRI data predicts clinical symptom severity." *NeuroImage*, vol. 57, 2011.

[5] I. Oztekin and D. Badre, "Distributed Patterns of Brain Activity that Lead to Forgetting." *Frontiers in human neuroscience*, vol. 5, 2011.

[6] T. Schmah, G. E. Hinton, R. S. Zemel, S. L. Small, and S. C. Strother, "Generative versus discriminative training of rbms for classification of fmri images." in *NIPS*, 2008.

[7] J.-D. Haynes and G. Rees, "Decoding mental states from brain activity in humans." *Nature reviews. Neuroscience*, vol. 7, 2006.

[8] L. I. Kuncheva and J. J. Rodríguez, "Classifier ensembles for fMRI data analysis: an experiment." *Magnetic resonance imaging*, vol. 28, 2010.

[9] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, 2009.

[10] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, 2010.

[11] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives." *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, 2013.

[12] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, 2007.

[13] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?" *The Journal of Machine Learning Research*, vol. 11, 2010.

[14] Y. Bengio, "Deep learning of representations: Looking forward," *CoRR*, vol. abs/1305.0445, 2013.

[15] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards ai," *Large-Scale Kernel Machines*, vol. 34, 2007.

[16] S. Rifai, Y. Dauphin, P. Vincent, Y. Bengio, and X. Muller, "The manifold tangent classifier," in *NIPS*, 2011, pp. 2294–2302.

[17] M. Kim, G. Wu, and D. Shen, "Unsupervised deep learning for hippocampus segmentation in 7.0 tesla mr images," in *Machine Learning in Medical Imaging*, 2013.

[18] H.-I. Suk and D. Shen, "Deep learning-based feature representation for ad/mci classification," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, 2013.

[19] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, 2013.

[20] T. Brosch and R. Tam, "Manifold learning of brain mris by deep learning," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*, 2013.

[21] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Computer Vision and Pattern Recognition,CVPR. IEEE Conference on*, 2009, pp. 1605–1612.

[22] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008.

[24] Y. Bengio, E. Thibodeau-Laufer, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," *CoRR*, vol. abs/1306.1091, 2013.

[25] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, 2006.

[26] M. Ozay, I. Öztekin, U. Öztekin, and F. T. Y. Vural, "Mesh Learning for Classifying Cognitive Processes," *Arxiv:1205.2382*, 2012.