A Robust Deep Model for Improved Classification of AD/MCI Patients

Journal:	*IEEE Journal of Biomedical and Health Informatics*
Manuscript ID:	Draft
Manuscript Type:	Machine Learning and Data Mining in Medical Imaging
Date Submitted by the Author:	n/a
Complete List of Authors:	Li, Feng; Old Dominion University, Electrical and Computer Engineering Tran, Loc; Old Dominion University, Electrical and Computer Engineering Thung, Kim-Han; Biomedical Research Imaging Center, University of North Carolina, School of Medicine Ji, Shuiwang ; Old Dominion University, Computer Science Shen, Dinggang; Biomedical Research Imaging Center, University of North Carolina, School of Medicine Li, Jiang; Old Dominion University , Electrical and Computer Engineering
TIPS:	Alzheimer's Disease, MRI, PEI , Deep Learning

A Robust Deep Model for Improved Classification of AD/MCI Patients

Feng Li[1], Loc Tran[1], Kim-Han Thung[3], Shuiwang Ji[2], Dinggang Shen[3], and Jiang Li[1]

[1]Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529
[2]Department of Computer Science, Old Dominion University, Norfolk, VA 23529
[3]Department of Radiology, University of North Carolina at Chapel Hill, NC 27599

Accurate classification of Alzheimer's Disease (AD) and its prodromal stage, Mild Cognitive Impairment (MCI), plays a critical role in preventing progression of memory impairment and improving quality of life for AD patients. Among many research tasks, it is of particular interest to identify noninvasive imaging biomarkers for AD diagnosis. In this paper, we present a robust deep learning system to identify different progression stages of AD patients based on MRI and PET scans. We utilized the dropout technique to improve classical deep learning by preventing its weight co-adaptation, which is a typical cause of over-fitting in deep learning. In addition, we incorporated stability selection, an adaptive learning factor and a multi-task learning strategy into the deep learning framework. We applied the proposed method to the ADNI data set and conducted experiments for AD and MCI conversion diagnosis. Experimental results showed that the dropout technique is very effective in AD diagnosis, improving the classification accuracies by 6.2% on average as compared to classical deep learning methods.

***Index Terms*—Alzheimer's Disease, MRI, PET, Deep Learning.**

I. Introduction

ALZHEIMER'S disease is the sixth-leading cause of death in the United States [1]. AD patients usually undergo progressive stages of cognitive and memory function impairment, including prodromal, MCI and AD. For each of these stages, significant amount of research has been conducted aiming to understanding the underlying pathological mechanisms. In addition, imaging biomarkers have been identified using different imaging modalities such as magnetic resonance imaging (MRI) [2], positron emission tomography (PET) [3], and functional MRI (fMRI) [4]. Imaging biomarkers are a set of indicators computed from image modalities and can be used for early detection of AD disease. It has been shown that fusing these different modalities may lead to more effective imaging biomarkers [6].

The first successful deep learning framework, auto-encoder, was developed in 2006 [7]. It was subsequently used in other application fields and achieved state-of-the-art performance in speech recognition, image classification and computer vision [8]. Deep learning itself also evolves after 2006. For instance, the multimodal deep learning framework boosted speech classification by learning a shared representation between video and audio modalities [9]. A dropout technique further improved zip code recognition, document classification and image recognition [10], [11].

In this paper, we developed a robust deep learning framework for AD diagnosis by fusing complementary information from MRI and PET scans. These 3D scans were preprocessed and features were extracted. We first applied the principal component analysis (PCA) to obtain PCs as new features. We then utilized the stability selection technique [13] together with the least absolute shrinkage and selection operator (Lasso) method [14] to select the most effective features, and the selected features were subsequently processed by the deep learning structure. Model weights in the deep structure were first initialized by unsupervised training and then fine-tuned by AD patient labels. During the fine-tune phase, the dropout technique was employed to improve the model's generalization capability. Finally, the learned feature representation was used for AD/MCI classification by support vector machine (SVM).

In addition to discrete patient labels (AD, MCI or Healthy), there are two additional clinical scores, namely Minimum Mental State Examination (MMSE) and Alzheimer's Disease Assessment Scale-Cognitive subscale (ADAS-Cog) associated with each patient. MMSE is an 30-point based, widely used questionnaire to measure cognitive impairment [15]. It is used to estimate the severity and progression of cognitive impairment instead of providing any AD information. ADAS-Cog is the most popular cognitive testing instrument to measure the severity of the most important symptoms of AD including the disturbances of memory, language, praxis, attention and other cognitive abilities referred to the core symptoms of AD [16]. These information is related and identifying the common information among them may be helpful for AD diagnosis. We configured the deep learning structure as a multi-task learning (MTL) framework, and treated the learning of class label, MMSE and ADAS-Cog as related tasks for improved main task (class label) prediction.

We evaluated the proposed method on the ADNI data set [1] and compared it with a baseline method and a similar deep learning system, where the auto-encoder was used as a feature extractor for AD diagnosis [6]. The baseline method contains feature selection and SVM steps but does not use

Manuscript received December 31, 2014; revised xxx. Corresponding author: Jiang Li (email: JLi@odu.edu).

[1]Available at http://www.loni.ucla.edu/ADNI

deep learning. We also evaluated the impact on performance of each of the components in the proposed system. A brief version of this paper was published at MLMI workshop [17].

II. Materials and Methods

The proposed system consists multiple components including PCA, stability selection, unsupervised feature learning, multi-task deep learning and SVM training as shown in Fig. 1. We detail each of these components in the following subsections.

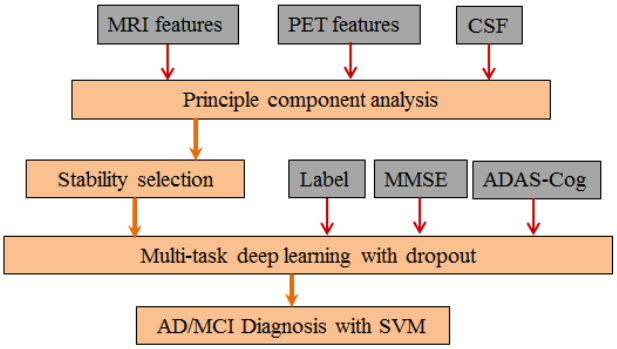

Fig. 1. Diagram of the proposed multi-task deep learning framework.

A. Data preprocessing

We utilized the public ADNI data set to validate our proposed deep learning framework. The data set consists of MRI, PET, and CSF data from 51 AD patients, 99 MCI patients (43 MCI patients who converted to AD (MCI.C), and 56 MCI patients who did not progress to AD in 18 months (MCI.NC)) as long as 52 healthy normal controls. In addition to the crisp diagnostic result (AD or MCI), this data set contains two additional clinical scores, MMSE and ADAS-Cog for each patient. A typical procedure of image processing was applied to the 3D MRI and PET volume [2], [18], [19] including anterior commissure-posterior commissure correction, skull-stripping, cerebellum removal and spatially normalization. Finally, we extracted 93 features from MRI and PET volume, respectively, and three CSF biomarkers, Aβ_{42}, $t-$tau, and p-tau were computed, resulting in 189 features for each subject.

B. Principal component analysis

Principal component analysis (PCA) is a linear orthogonal transformation that converts a set of features into linearly uncorrelated variables in which each of the new variable is a linear combination of all original features [5]. The first principal component (PC) is defined as the one that can explain the largest variance in the original data set, and the second PC has the second largest variance under the constraint that it is orthogonal to the first component. If correlations exist among features, the number of PC can be found is usually less than the number of features in original data. PCA is optimal for preserving energy and it is often used for dimensionality reduction by just keeping the first few PCs.

Let $\mathbf{F}$ denote a feature data set with a size of n by p, where n is the number of data samples and p is the number of features in the data, and each column in $\mathbf{F}$ is centered. PCA can be achieved by performing singular value decomposition (SVD) on $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{1}$$

where $\mathbf{U}$ is an n by n matrix with orthogonal unit columns (left singular vectors of $\mathbf{F}$), Σ is an n by p diagonal matrix consists of singular values of $\mathbf{F}$ from the largest to least and $\mathbf{V}$ is an p by p matrix whose columns are orthogonal unit vectors (right singular vectors of $\mathbf{F}$).

To achieve dimensionality reduction, the first l columns in $\mathbf{V}$ corresponding to the first l largest singular values of $\mathbf{F}$ can be used as a transformation matrix to be applied on $\mathbf{F}$,

$$\mathbf{x} = \mathbf{F}\mathbf{V}_l, \tag{2}$$

where $\mathbf{V}_l$ consists of the first l columns of $\mathbf{V}$.

Geometrically, PCA analysis rotates data to align the maximum variance direction of the data with coordinate system as illustrated in Fig. 2. PCA is an effective tool for dimensionality reduction but the preserved PCs may not be useful for classification. The two dimensional artificial data set in Fig. 2 consists of 'blue' and 'red' classes. After PCA, the whole data set was rotated and its main axis was aligned with the coordinate system. However, even PC 1 has the largest variance, it does not contain any discriminating information for the two classes. For the purpose of classification, PC 2 is preferred and a feature selection step is necessary.

C. Stability selection

In this paper, we first applied PCA to the 189 features and used the resulting PCs as new features. we then applied Lasso [14] to identify the most effective PCs for AD diagnosis. Lasso tries to minimize the following cost function for feature selection:

$$\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{w}\mathbf{x}||_2^2 + \lambda||\mathbf{w}||_1, \tag{3}$$

where $y \subset \{1, -1\}$ is the class label associated with the feature vector/PC $\mathbf{x}$, λ is a regularization parameter and $\mathbf{w}$ is the weight vector in the linear model. Because of the L_1 norm constraint on the weight magnitude, the solution minimizing the above cost function is usually sparse, meaning that if a feature in the feature vector $\mathbf{x}$ is not correlated with the target variable, $\mathbf{y}$, the feature will have a zero weight. Features having none zero weights will be selected and otherwise be excluded.

It is well known that the solution of L_1 norm based optimizations are sensitive to the choice of λ, and it is difficult to determine how many features should be kept in the model. A recent breakthrough sheds a light on selecting the right amount of regularization for stability selection [13]. The idea is to repeat the feature selection procedure multiple times based on bootstrapped datasets and compute the probability of the features to be selected. The final selected features are those having probabilities above a predefined threshold t. It has been shown experientially and theoretically that the feature selection

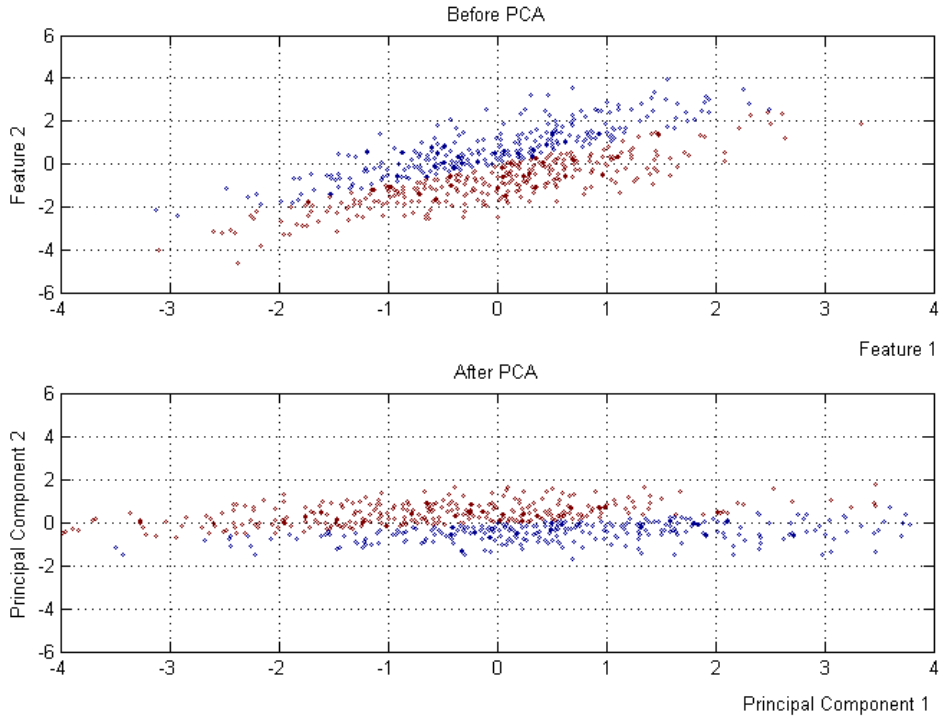

Fig. 2. Principal Component Analysis Example. PC 1 contains the most energy of the data but does not have any discrimination information for the 'red' and 'blue' classes.

results vary little for sensible choices in a range of the cut-off value for t. We incorporate the stability selection concept into the AD patient diagnosis in this paper. In particular, we repeated the Lasso procedure 50 times and each time with a different value for the parameter λ (We used the SLEP toolbox for Lasso [2]). A probability, p_i, for the ith feature was computed by counting frequency of the feature being selected in the 50 experiments. The ith feature was selected if p_i is larger than a pre-defined threshold t.

D. Multi-task deep learning with dropout

In contrast to traditionary three-layer neural network (shallow structure), deep learning is based on a deep architecture consisting of many layers of hidden neurons for modelling. A shallow architecture would involve many duplications of effort to express things and such a fat architecture has been shown to suffer from the problem of over-fitting, which leads to a poor generalization capability. Instead, deep architecture could more gracefully reuse previous computations and discover complicated relations of input [20].

To train a deep architecture, the standard Backpropagation (BP) algorithm did not work well with randomly initialized weights because the error feedback becomes progressively noisier as it goes back to lower levels (close to inputs), making the low level weight updates less effective. Even though experiments have shown that if top layers have enough units, the deep structure can still bring down training errors small enough, it cannot generalize well to new data [21]. This is because the top layers can be effectively trained by gradient based algorithms but low levels cannot. The randomly initialized low level layers behave like random feature detectors so good representations for original data were not achieved leading to degraded generalization capability [21]. In 2006, a breakthrough in deep learning has made deep architecture training possible by utilizing the restricted Boltzmann machine (RBM) to initialize multiple hidden layers one layer at a time in an unsupervised manner [7]. With unsupervised learning, deep learning tries to understand data first, i.e., to obtain a task specific representation from data so that a better classification can be achieved. It has experimentally proven that the unsupervised learning step plays a critical role in the success of deep learning [8]. The proposed deep model shown in Fig. 3 consists of several components that will be described bellow.

1) Pre-training with RBM

Each layer in the proposed deep model is an RBM and the deep model used in this paper consists of a stack of RBMs. RBM is an energy-based model in which a scalar energy is associated with each configuration of the variables in the model, and a probability distribution function (PDF) through the energy function is usually defined. The purpose of learning is to modify the energy function so that a desirable PDF can be achieved, i.e., to have low energy. A basic RBM model having a visible (input) layer and a hidden (output) layer is shown in Fig. 4. The visible layer of the bottom RBM contains real-valued units (receiving data) and all other RBM layers have binary units. Let $v \in R^M$ represent input data (visible units) and $h \in 0, 1^N$ denote binary hidden units

[2]Available at http://www.public.asu.edu/ jye02/Software/SLEP/index.htm

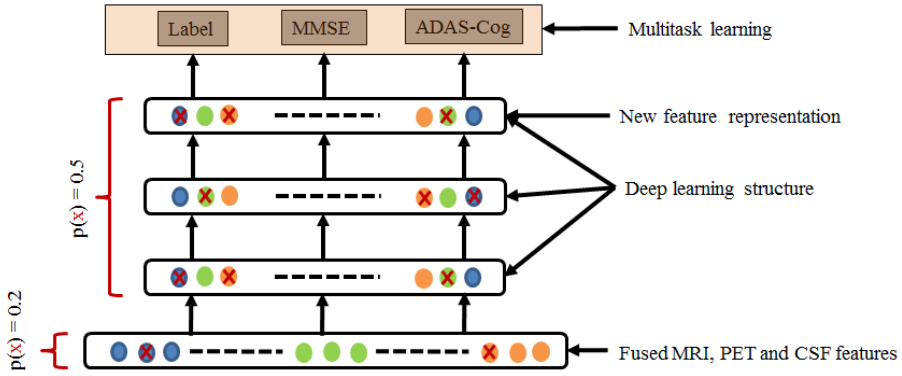

Fig. 3. Multi-task deep learning with dropout. "x" denotes a dropped unit.

for the bottom RBM, we used Gaussian-Bernoulli RBMs to train it [21], [22]. All other RBMs were trained by utilizing Bernoulli-Bernoulli distribution. Variables v and h have a joint probability distribution defined as

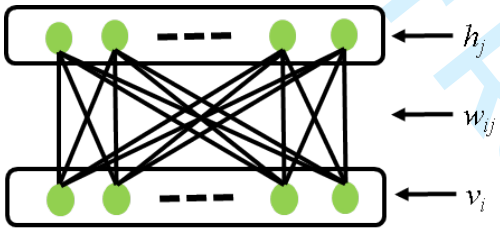

Fig. 4. A basic RBM model.

$$p(v,h) = \frac{1}{Z} exp^{-E(v,h)}, \tag{4}$$

where $E(v,h)$ is an energy function and Z is a normalization constant. For real-valued visible layer RBMs, $E(v,h)$ is defined as

$$E(v,h) = \frac{1}{2\sigma^2}\sum_i v_i^2 - \frac{1}{\sigma^2}(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{i,j} v_i w_{ij} h_j), \tag{5}$$

where c_i and b_j are biases of the ith and jth units in the visible and hidden layers, respectively, w_{ij} is the weight connecting v_i and h_j, and σ^2 is the variance of v. The conditional probability distributions are

$$P(h_j = 1|v) = sigmoid(\frac{1}{\sigma^2}(\sum_i w_{ij} v_i + b_j)), \tag{6}$$

$$P(v_i|h) = N(\sum_j w_{ij} h_j + c_i, \sigma^2). \tag{7}$$

If both visible and hidden layers are binary, the energy function and conditional probability distributions are defined as

$$E(v,h) = -(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{ij} v_i w_{ij} h_j), \tag{8}$$

$$P(h_j = 1|v) = sigmoid(\sum_i w_{ij} v_i + b_j), \tag{9}$$

$$P(v_i = 1|h) = sigmoid(\sum_j w_{ij} h_j + c_i). \tag{10}$$

Model parameters w, b and c are updated using contrastive divergence [23]. For RBM having a real-valued visible layer, the formulas for updating those parameters during each iteration are

$$\Delta W_{ij}^{t+1} = \eta \Delta W_{ij}^t - \epsilon(< \frac{1}{i}\frac{1}{2} v_i h_j >_d - < \frac{1}{i}\frac{1}{2} v_i h_j >_m), \tag{11}$$

$$\Delta b_i^{t+1} = \eta \Delta b_i^t - \epsilon(< \frac{1}{i}\frac{1}{2} v_i >_d - < \frac{1}{i}\frac{1}{2} v_i >_m), \tag{12}$$

$$\Delta c_j^{t+1} = \eta \Delta c_j^t - \epsilon(< h_j >_d - < h_j >_m). \tag{13}$$

where $< \ >_d$ and $< \ >_m$ denote the expectation computed over data and model distributions accordingly, t is iteration index, η is momentum and ϵ is learning rate. For binary RBM, equations (11) and (12) become

$$\Delta W_{ij}^{t+1} = \eta \Delta W_{ij}^t - \epsilon(< v_i h_j >_d - < v_i h_j >_m), \tag{14}$$

$$\Delta b_i^{t+1} = \eta \Delta b_i^t - \epsilon(< v_i >_d - < v_i >_m). \tag{15}$$

Note that pre-training of RBM is unsupervised, i.e., class label (classification task) or desired output (regression) is not needed in the training. After the pre-training, we attached the class label on top of the stacked RBMs and utilized an adaptive backpropagation algorithm to fine-tune the weights in the model. All binary layers are also converted to real-valued units by using their continuous activities. Thus the deep learning model turns to be a traditional multilayer perceptron (MLP) but its weights are initialized by RBM.

2) Multi-task learning

In multi-task learning, related tasks are learnt simultaneously by extracting and utilizing appropriate shared information across tasks to improve performance. It has received attention in broad areas recently such as machine learning, data mining, computer vision, and bioinformatics [24], [25], [26]. This approach is particularly effective when only limited training data for each task is available. It is worthy noting that neural network can simultaneously model multiple outputs making deep learning a natural multi-task learning framework if multiple tasks share inputs [7]. The proposed multi-task deep learning framework is shown in Fig. 3, where we treated class label, MMSE and ADAS-Cog as three different tasks but modeling them simultaneously. MMSE and ADAS-Cog were normalized to the range of [0,1] and we used the deep

structure as a regression model. The class label was coded by the 1-of-k scheme. To classify an input vector, we checked the corresponding k outputs and assign it to the class having the largest output. One drawback of deep model is over-fitting due to large capacity of deep models. This is more prominent if training data is limited. To overcome this limitation, we utilized the dropout technique to improve training.

3) Dropout with adaptive adaptation

Deep learning achieved excellent results in applications where training data size is large. For small sized data sets such as the one in this paper, it is still possible for a deep structure to over-fit the data given the fact that it usually has tens of thousands or even millions of parameters. To improve the generalization capability of the model, the dropout technique tries to prevent weight co-adaptation by randomly dropping out some units in the model during training [10], [11]. We incorporated the dropout technique in the multi-task learning context to improve AD diagnosis as shown in Fig. 3. In the training process, each hidden unit in the model was dropped with a probability of 0.5 when a batch of training cases were present. Previous experiments [10] showed that it is also beneficial if we apply the "dropout" process to the input layer but with a lower probability (0.2 in this paper). In the testing procedure, all hidden units and inputs were used to compute model outputs for a testing case with appropriate compensations, i.e., weights between inputs and the first hidden layer were scaled by 0.8 and all other weights were halved.

During the multi-task fine-tuning step, the stochastic gradient descent method with a fixed learning factor is usually utilized as [7],

$$w_{ij} = w_{ij} + \Delta w_{ij} = w_{ij} - \alpha \frac{\partial L}{\partial w_{ij}}, \tag{16}$$

where $\frac{\partial L}{\partial w_{ij}}$ is the gradient of the cost function L and α is a learning factor. Sometimes, the weights update may contain a momentum term [10]. We proposed to use an adaptive learning factor to speed up the adaptation. The motivation of the adaptive learning is that the learning factor should be large at location where gradient is small and vice verse. Assume the decrease of L due to the change in w_{ij} is approximated by

$$\Delta L^{ij} = L^{ij}_{new} - L^{ij}_{old} \approx \frac{\partial L}{\partial w_{ij}} \times \Delta w_{ij} = -\alpha [\frac{\partial L}{\partial w_{ij}}]^2, \tag{17}$$

then ΔL due to all w_{ij} can be computed as

$$\Delta L = -\alpha \sum_i \sum_j [\frac{\partial L}{\partial w_{ij}}]^2. \tag{18}$$

Suppose we want to decrease L by $\beta\%$, then $L_{new} = (1 - \beta)L_{old}$, and an adaptive learning factor α can be determined as

$$\alpha = \frac{\beta L_{old}}{\sum_i \sum_j [\frac{\partial L}{\partial w_{ij}}]^2}. \tag{19}$$

We set β as 10% in our experiments in this paper. Once the new feature representation is learned, an SVM classifier [12] was trained using the learned representation.

E. SVM Classifier

Given a set of data pairs $\{\mathbf{x}_p, t_p\}_{p=1}^n$, where $\mathbf{x}_p \in \mathcal{R}^M$ is the feature vector extracted from AD patients, $t_p \in \{+1, -1\}$ is a class label (AD vs. non-AD) associated with $\mathbf{x}_p$. An SVM defines a hyperplane

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = 0 \tag{20}$$

separating the data points onto 2 classes. In equation (20), $\mathbf{w}$ and b are the plane parameters, and $\phi(\mathbf{x})$ is a function mapping the vector $\mathbf{x}$ to a higher dimensional space. The hyperplane (20) is determined using the concept of *Structural Risk Minimization* [12] by solving the following optimization problem,

$$\min_{\mathbf{w},b,\xi} \left(\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{p=1}^{n} \xi_p \right) \tag{21}$$

subject to

$$t_p(\mathbf{w}^T \phi(\mathbf{x}_p) + b) \geq 1 - \xi_p, \xi_p \geq 0 \tag{22}$$

here C is a 'soft margin' or a penalty parameter and ξ_p a slack factor. After the hyperplane is determined, an AD case is declared if $f(\mathbf{x}_p) > 0$, otherwise a non-AD is declared.

III. Results and Discussions

A. Experimental setup

1) Ten-fold cross validation

We consider three classification tasks including AD patients vs Healthy Control subjects (AD vs HC), MCI patients vs HC (MCI vs HC) and MCI-converted vs MCI-non converted (MCI.C vs MCI.NC). For each task, we utilized a ten-fold cross-validation (CV) scheme to evaluate the proposed method. In the ten-fold CV, we randomly divided the data set into 10 parts and for one run, we separated one part for testing and applied the proposed framework to the remaining data to train a classification model. This procedure was repeated 10 times so that each part was tested once. Finally, testing accuracies were computed. To obtain a more realizable estimate of the performance, we repeated the ten-fold CV ten times for each task with different random data partitions and computed average accuracy. To compare different classification models, we kept the same data partitions in the ten-fold CV and utilized the paired-t test to evaluate if there is a significant performance difference.

2) Hyperparameter determination

We did preliminary experiments to determine the structure of the deep learning model. For all the three classification tasks, it was found that a three hidden layers with hidden units of 100-50-20 worked the best among the candidate structures considered and was utilized in our experiments. For the SVM classifier, we tried different kernels and a linear kernel was chosen. We also did a grid search for the "soft margin" parameter in the linear kernel SVM model but it did not improve the classification accuracies. Therefore, in all experiments, we utilized a three hidden-layer model with a structure of 100-50-20 for feature learning and a linear SVM with default soft margin as classifier.

TABLE I

PERFORMANCE COMPARISON (IN%) OF THE COMPETING METHODS. THE PROPOSED METHOD CONSISTS OF FOUR COMPONENTS. "-PCA" STANDS FOR "THE PROPOSED METHOD WITHOUT THE PCA COMPONENT" AND "SS" STANDS FOR STABILITY SELECTION, "BASELINE" DENOTES THE FRAMEWORK WITHOUT THE DEEP LEARNING COMPONENT.

Tasks		Proposed	-PCA	-Dropout	-SS	-MultTask	Baseline
AD vs HC		**91.4**(1.8)	89.6(1.3)	84.2(3.0)	89.4(1.6)	90.3(1.7)	86.4(2.0)
MCI vs HC		**77.4**(1.7)	76.4(1.5)	73.1(3.1)	74.3(1.6)	75.6(1.7)	72.1(3.0)
MCI.C vs MCI.NC		57.4(3.6)	**58.1**(1.8)	50.2(3.3)	57.7(1.8)	56.7(3.0)	50.6(4.7)
Average		**75.4**	74.7	69.2	73.8	74.2	69.7

TABLE II

PAIRED-t TEST BETWEEN RESULTS OF THE PROPOSED METHOD VS DEEP LEARNING WITHOUT DROPOUT. THE METHODS OF "SAEF" AND "LLF+SAEF" WERE PROPOSED BY SUK [6]. "SAEF" STANDS FOR STACKED AUTO-ENCODER FEATURES AND "LLF" DENOTES LOW LEVEL FEATURES.

Tasks		Proposed	-Dropout	Improvement	p-value	SAEF	LLF+SAEF
AD vs HC		**91.4**(1.8)	84.2(3.0)	7.2	$< 10^{-3}$	83.2(2.7)	85.3(3.2)
MCI vs HC		**77.4**(1.7)	73.1(3.1)	4.3	0.0034	70.1(2.8)	76.9(2.3)
MCI.C vs MCI.NC		57.4(3.6)	50.2(3.3)	7.2	$< 10^{-3}$	58.4(4.1)	**60.3**(2.3)
Average		**75.4**	69.2	6.2	N/A	70.6	74.2

3) Impact assessment for individual component

There are four components in the proposed framework including PCA, stability selection, dropout and multi-task learning. Inspired by "sensitivity analysis" and "impact assessment" that analyze inputs of or components in a model and identify their impacts on the model objectives by varying the inputs [28]. We incorporated a similar concept to evaluate the impact of each component on model performance by varying the component (presence vs absence). 'Absence' means that the component was not included in the model.

4) Methods for comparison

We compared the proposed method with a baseline method and a similar deep learning system proposed in [6]. The baseline method consists of all components in the proposed system except the deep learning step. The work by Suk in [6] is a auto-encoder based deep learning method in which feature representations for MRI, PET and CSF from the same data set were learned separately and combined by a linear SVM classifier. They also combined the learned representations with original features for AD diagnosis.

B. Results

Table I shows the overall performances of the proposed method and the impact of each component in the framework. The proposed method performed the best in diagnosing AD and MCI patients with accuracies of 91.4% and 77.4%, respectively, and it is significantly better than the baseline method that obtained accuracies of 86.4% and 72.1% for the diagnosis. In the MCI conversion diagnosis (MCI.C vs MCI.NC), the PCA component slightly degraded the proposed method (from 58.1% to 57.4%) but it is still significantly better than the baseline method (57.4% vs 50.6%).

Among those components, it is obvious that "dropout" has the most significant impact on the performances. Without "dropout", deep learning did not improve the baseline method (69.2% vs 69.7% in terms of average acc.). The least important component is "PCA", the average acc. slightly dropped from 75.4% to 74.7% without the PCA component. Without "stability selection" and "multi-task learning", the average accuracy dropped from 75.4% to 73.8% and 74.2%, respectively.

We conducted a paired-t test between results by the proposed method and those from classical deep learning ("-Dropout"). Table II lists the improvements and p-values. The average improvement is 6.2% and the improvements for all the three classification tasks are significant.

The work by Suk [6] on the same data set is also shown in Table II, where "SAEF" corresponds to the method using features learned by a deep auto-encoder and "LLF+SAEF" represents the method that combines original features with the SAEF features for AD diagnosis. The proposed method (75.4%) outperformed the SAEF method (with an average accuracy of 70.6%). By combining SAEF with LLF (LLF+SAEF), the average accuracy was increased to 74.2% [6].

C. Discussions

There are usually two ways to increase the generalization capability of a learned model, adding regularization (L_1 or L_2 norm) on weights or using committee machine. However, solving the regularization problem is usually challenging especially in the deep learning context. In addition, the committee machine technique requires averaging many separately trained models to compute a prediction for a testing case, which is time consuming for deep learning. The dropout procedure does the both (constraint and committee machine) simultaneously in a very efficient way. 1) Each sub-model in training is a sampled model from all possible ones and all sub-models share weights. The weight sharing property is equivalent to the L_1 or L_2 norm constraint on weights, and 2) The testing procedure is an approximation of averaging all trained sub-models for a testing case but it does not separately store them because they share weights. This is an extremely efficient and a smart implementation of a committee machine [10], [11].

The impact evaluation method was inspired by "sensitivity analysis" and "impact assessment" [28]. We were aiming to identify the impact on performance of each component in the model by excluding the component from the pipeline. Note that we did not try to decouple components in the system.

This evaluation method may not be a strict sensitivity analysis or impact assessment by means of their definitions, but we can verify each component if it can improve the AD diagnosis when it is included in the proposed system. Our experiments showed that the dropout component has the largest impact on performance, stability selection ranked the second, multi-task learning the third and PCA has the least impact on the performance.

In terms of stability selection and computational efficiency, there were usually 30 to 40 features left after the stability selection and it took about 1 hour for a personal computer to conduct a ten-fold CV evaluation for one task.

It is worth to note that the results by the proposed method in Table I and Table II only used the new representations learnt by the deep model. We tried to combine the new representations with original features but the combination did not improve the performance. In [6], new representations learnt from auto-encoder did not perform well unless they were combined with original features. Our experiment also showed that deep model without dropout just performed comparably as the baseline method (see Table I). It seems that traditional deep learning cannot extract information effectively from small data set unless it is regularized by techniques such as dropout. In [6], utilizing the multi-kernel SVM (MK-SVM) to combine SAEF features from MRI, PET and CSF boosted the performances to 95.9%, 85.0% and 75.8% for the three tasks, respectively. Since the dropout technique improved upon the basic deep learning significantly in this paper, we are currently investigating if the MK-SVM can further boost the performance of the proposed system.

We did not attempt to perform a comprehensive comparison study of the proposed method with others that have been applied to this data set in literature. Instead, we have evaluated some recently proposed advanced machine learning techniques for AD diagnosis including Lasso, stability selection, multi-task learning, deep learning and dropout. The dropout technique seems to be an effective method of regularization for learning with small data. Without dropout, deep learning has no advantage over the baseline method on ANDI data set (69.2% vs 69.7%). Note that dropout is computationally very efficient as compared to either L_1 norm based regularization or committee machine and it can be extended to many models other than the deep model discussed in this paper.

IV. Conclusion

Our proposed method achieved 91.4%, 77.4% and 57.4% accuracies for AD, MCI and MCI conversion diagnosis, respectively. The framework consists of multiple components including PCA, stability selection, dropout and multi-task deep learning. We showed that dropout is the most effective one. This is not surprising because the size of ADNI data is relatively small compared to that of the deep structure utilized in this paper. Classical deep learning cannot help but with the dropout technique, the average accuracy was improved by 6.2% on average. We are incorporating MK-SVM [6] into our method for improved AD diagnosis.

References

[1] Alzheimer's Association: 2012 Alzheimer's disease facts and figures. Alzheimer's & Dementia 8(2), 131-168 (2012).

[2] Davatzikos, C., Bhatt, P., Shaw, L.M., Batmanghelich, K.N., Trojanowski, J.Q.: Prediction of MCI to AD conversion, via MRI, CSF biomarkers, and pattern classification. *Neurobiology of Aging*, 32(12), 2322.e19-2322.e27 (2011).

[3] Nordberg, A., Rinne, J.O., Kadir, A., Langstrom, B.: The use of PET in Alzheimer disease. *Nature Reviews Neurology*, 6(2), 78-87 (2010).

[4] Greicius, M.D., Srivastava, G., Reiss, A.L., Menon, V.: Default-mode network activity distinguishes Alzheimer's disease from healthy aging: Evidence from functional MRI. *Proceedings of the National Academy of Sciences of the United States of America* 101(13), 4637-4642 (2004).

[5] Jolliffe, I.T., *Principal Component Analysis*, Series: Springer Series in Statistics, 2nd ed. (2002).

[6] Suk H., Shen D., Deep learning-based feature representation for AD/MCI classification, *MICCAI*, 583-590 (2013).

[7] Hinton, G.E., Grivastava, Osindero, S., Teh, Y.W., A fast learning algorithm for deep belief nets, *Neural computation*, 18(7), 1527-1554 (2006).

[8] Bengio, Y., Courville, A., Vincent, P., Representation learning: A review and new perspectives, *PAMI*, 35(8), 1798-1828 (2013).

[9] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng A., Multimodal deep learning, *ICML*, 689-696 (2011).

[10] Hinton, G.E., Srivastave, N., Krizhevsky, A.,Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*:1207.0580 (2012).

[11] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929-1958, (2014).

[12] Cortes, C. and Vapnik, V., Support-vector networks. *Machine Learning*, 20 (3), 273, (1995).

[13] Meinshausen, N., Buhlmann, P.: Stability selection. *J. R. Statist. Soc. B*, 417-473 (2010).

[14] Tibshirani, R.: Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58 (1): 267-288 (1996).

[15] Pangman, VC; Sloan, J; Guse, L., An Examination of Psychometric Properties of the Mini-Mental State Examination and the Standardized Mini-Mental State Examination: Implications for Clinical Practice. *Applied Nursing Research*, 13(4), 209213, (2000).

[16] Kolibas E, Korinkova V, Novotny V, Vajdickova K, Hunakova D., ADAS-cog (Alzheimer's Disease Assessment Scale-cognitive subscale)–validation of the Slovak version, Bratisl Lek Listy, 101(11),598-602, (2000).

[17] Li, F., Tran, L., Thung, KH, Ji, S., Shen, D. and Li, J., Robust Deep Learning for Improved Classification of AD/MCI Patients, *Machine Learning in Medical Imaging*, 240-247, (2014).

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

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

[20] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11, 625-660 (2010).

[21] Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1), 1127 (2009).

[22] Cho, K., Ilin, A. and Raiko, T., Improved learning of Gaussian-Bernoulli restricted Boltzmann machines," in *Artificial Neural Networks and Machine LearningICANN 2011*, ed: Springer, 10-17 (2011).

[23] Hinton, G., Osindero, S. and Teh, Y.-W., A fast learning algorithm for deep belief nets, *Neural computation*, vol. 18, 1527-1554 (2006).

[24] Heisele, B., Serre, T., Pontil, M., Vetter, T. and Poggio. T., Categorization by learning and combining object parts. In *NIPS*, (2001).

[25] Ji, S. and Ye, J., An accelerated gradient method for trace norm minimization, *Proceedings of the 26th Annual International Conference on Machine Learning*, 457464, (2009).

[26] Xue, Y., Liao, X., Carin, L. and Krishnapuram, B., Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8, 35-63 (2007).

[27] Caruana, R. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning*, 28, 41-75 (1997).

[28] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D. Saisana, M., and Tarantola, S., Global Sensitivity Analysis. The Primer, John Wiley & Sons, (2008).