Heliyon 9 (2023) e18086

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Autoencoder and restricted Boltzmann machine for transfer learning in functional magnetic resonance imaging task classification

Jundong Hwang, Niv Lustig, Minyoung Jung, Jong-Hwan Lee [*]

Department of Brain and Cognitive Engineering, Korea University, Seoul, South Korea

ARTICLE INFO

Keywords:
Autoencoder
Deep neural network
Functional magnetic resonance imaging
Human connectome project
Restricted Boltzmann machine
Transfer learning

ABSTRACT

Deep neural networks (DNNs) have been adopted widely as classifiers for functional magnetic resonance imaging (fMRI) data, advancing beyond traditional machine learning models. Consequently, transfer learning of the pre-trained DNN becomes crucial to enhance DNN classification performance, specifically by alleviating an overfitting issue that occurs when a substantial number of DNN parameters are fitted to a relatively small number of fMRI samples. In this study, we first systematically compared the two most popularly used, unsupervised pretraining models for resting-state fMRI (rfMRI) volume data to pre-train the DNNs, namely autoencoder (AE) and restricted Boltzmann machine (RBM). The group in-brain mask used when training AE and RBM displayed a sizable overlap ratio with Yeo's seven functional brain networks (FNs). The parcellated FNs obtained from the RBM were fine-grained compared to those from the AE. The pre-trained AE and RBM served as the weight parameters of the first of the two hidden DNN layers, and the DNN fulfilled the task classifier role for fMRI (tfMRI) data in the Human Connectome Project (HCP). We tested two transfer learning schemes: (1) fixing and (2) fine-tuning the DNN's pre-trained AE or RBM weights. The DNN with transfer learning was compared to a baseline DNN, trained using random initial weights. Overall, DNN classification performance from the transfer learning proved superior when the pre-trained RBM weights were fixed and when the pre-trained AE weights were fine-tuned (average error rates: 14.8% for fixed RBM, 15.1% fine-tuned AE, and 15.5% for the baseline model) compared to the alternative scenarios of DNN transfer learning schemes. Moreover, the optimal transfer learning scheme between the fixed RBM and fine-tuned AE varied according to seven task conditions in the HCP. Nonetheless, the computational load reduced substantially for the fixed-weight-based transfer learning compared to the fine-tuning-based transfer learning (e.g., the number of weight parameters for the fixed-weight-based DNN model reduced to 1.9% compared with a baseline/fine-tuned DNN model). Our findings suggest that weight initialization at the DNN's first layer using RBM-based pre-trained weights provides the most promising approach when the whole-brain fMRI volume supports associated task classification. We believe that our proposed scheme could be applied to a variety of task conditions to improve their classification performance and to utilize computational resources efficiently using our AE/RBM-based pre-trained weights compared to random initial weights for DNN training.

[*] Corresponding author. Department of Brain and Cognitive Engineering, Korea University, Anam-ro 145, Seongbuk-gu, Seoul 02841, South Korea.

E-mail address: jonghwan_lee@korea.ac.kr (J.-H. Lee).

https://doi.org/10.1016/j.heliyon.2023.e18086
Received 7 October 2022; Received in revised form 18 May 2023; Accepted 6 July 2023
Available online 16 July 2023

1. Introduction

Parcellation of brain regions, one of the oldest research interests in neuroscience [1], experienced significant developments with the advancement of neuroimaging techniques [2–4]. Notably, functional magnetic resonance imaging (fMRI) has been used recently to perform functional brain network (FN) parcellations [5–7]. For example, resting-state fMRI (rfMRI) has delivered coarse- and fine-grained FN parcellation [6,7]. Similarly, rfMRI data has collaborated with task fMRI (tfMRI) and structural MRI data to improve parcellation quality [8]. These brain parcellation atlases are essential to several fields of neuroscientific research, including individual identification using rfMRI functional connectivity (FC) within the brain [9] or obtaining individual-level, functional parcellation using group-level prior maps [10].

Neuroimaging data analysis has shifted towards neural network–based computational models to extract meaningful brain function features and characteristics in a data-driven manner [11,12]. Computational models have been used for brain parcellation and segmentation [13–16]. For example, Hjelm and colleagues (2014) utilized a restricted Boltzmann machine (RBM) with simulated and real fMRI data to identify brain networks and their activation patterns more accurately than the alternative independent component analysis (ICA) method.

Unsupervised factorization models such as RBM and autoencoder (AE) establish the building blocks of deep neural networks (DNNs), trained to learn an encoded representation of the input data for reconstruction, often with various regularization methods. For this reason, the learned features obtained from the unsupervised models were valuable for brain parcellation and also assisted as pre-training models in a transfer learning framework [17,18]. The models' trained weights are transferred to a DNN, with the goal of classifying the input whole-brain fMRI volume into different categories, and the model undergoes further training and evaluation. Several studies demonstrated that this type of pre-training offers utility by increasing the classification accuracy of DNN models compared to a DNN model initially trained with random weights [17,19–21]. Initializing the weights in this manner yields lower classification error rates and faster convergence than with random value weights.

However, we perceive a lack of systematic investigation of AE and RBM efficacy in the functional parcellation of rfMRI data. Similarly, we identify an absence in researching the utility of a functionally parcellated brain region atlas for enhancing DNN classification performance in the context of a transfer learning framework. We evaluated the models for various hyperparameters based on the overlap and similarity of weight feature maps with known networks, such as Yeo's seven FNs [22]. We subsequently applied the trained AE/RBM weights as DNN pre-trained weights to classify tfMRI data across multiple task conditions in the Human Connectome Project (HCP) dataset. We hypothesized that extracted rfMRI data features would provide valuable information for tfMRI data classification in the context of transfer learning for whole-brain fMRI activation pattern classification.

2. Materials and methods

2.1. Participants and data acquisition

In HCP dataset, each subject performed seven tasks and two resting-state conditions while fMRI data was acquired. Moreover, fMRI data for each condition was measured twice with both left-to-right (LR) and right-to-left (RL) phase encoding directions. The rfMRI data employed to train the AE/RBM weights for this study was obtained from the HCP S900 release [8], which included 899 healthy subjects (Fig. 1a). The subjects who did not perform any of the four rfMRI runs (i.e., REST1 and REST2 with LR and RL phase encoding) were excluded from the analysis, leaving 839 subjects for training. In the HCP S1200 release, the test-retest data from 45 subjects who were also included in the S900 release were acquired. In the DNN classifier's transfer learning phase, the tfMRI dataset in the S900 release was used as a training set by excluding those 45 test-retest subjects (Fig. 1a). Consequently, the retest tfMRI data from the 45

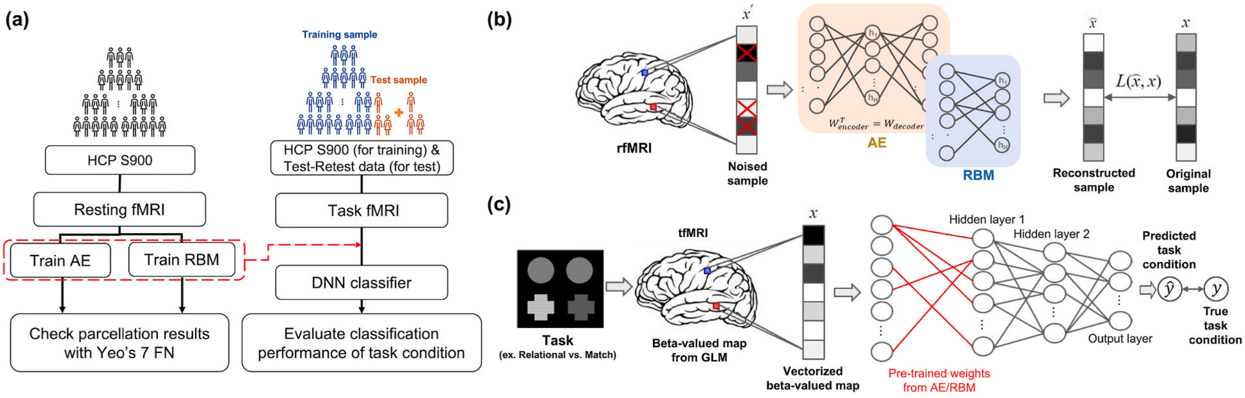

Fig. 1. (a) Flowcharts to describe (i) the AE/RBM training using rfMRI data (left) and (ii) the DNN training for task condition classification via transfer learning (right). (b) Autoencoder (AE) and restricted Boltzmann machine (RBM) training with weight sparsity control scheme and noised input patterns. (c) DNN transfer learning utilizing the weights of the pre-trained AE (i.e., $W_{encoder}$) or RBM models as fixed weights or by fine-tuning weights to predict one of the c classes of task conditions at the output layer using the vectorized beta-valued map from GLM.

subjects became test samples for the trained DNN classifier. Similar to the rfMRI data training AE/RBM, subjects who performed the task conditions for both the LR and RL phase encoding tfMRI runs were included. Subsequently, the total number of subjects that trained and tested the DNN in the transfer learning phase for each task condition was as follows: working memory = 834 for training and 42 for testing; motor = 831 and 42; emotion = 809 and 41; relational = 806 and 41; social = 809 and 42; language = 810 and 41; gambling = 833 and 42. The fMRI acquisition parameters included TR = 720 ms, TE = 33.1 ms, in-plane field of view = 208 × 180 mm, 72 slices, flip angle = 52°, voxel size = 2 × 2 × 2 mm^3, and number of volumes per run = 1200.

2.2. fMRI data preprocessing

The dataset utilized was the minimally preprocessed and FIX-denoised volumetric rfMRI data provided as part of the HCP S900 release. The data was preprocessed using spatial artifact/distortion removal, surface generation, cross-modal registration, and alignment to 2 mm MNI space. The FIX (FMRIB's ICA-based X-noisifier) approach [23,24] was applied as well. We employed further preprocessing steps using the AFNI software including resampling the original EPI data to a 3 mm isotropic voxel size, spatial smoothing using an 8 mm full width at half-maximum (FWHM) Gaussian kernel, and subtraction of each EPI volume's mean value. Finally, an in-brain mask (i.e., group-level in-brain mask) comprised the intersection of all individual in-brain masks provided with the dataset for all available subjects (resulting mask size = 52,470 voxels). This mask transformed the three-dimensional (3D) volumetric data to one-dimensional (1D) vectors, subsequently normalized to zero mean and unit variance to serve as input for the AE/RBM (Fig. 1b). To ensure that our in-brain mask covers a majority of the seven FNs, we calculated the associated overlap ratio. Specifically, spatial overlap between each of Yeo's FNs and those in in-brain mask areas was calculated using Eq. (11). For the transfer learning phase, a general linear model implemented in AFNI (i.e., 3dDeconvolve) was applied to obtain beta-valued maps for each task condition using the task-related regressors defined from onset timing and duration of the task period. Consequently, the resulting beta-valued maps were inputs for the DNN classifier (Fig. 1c).

2.3. Autoencoder and restricted Boltzmann machine

Fig. 2 illustrates AE and RBM training; those models were trained using backpropagation via stochastic gradient descent [25] and contrastive divergence [26] algorithms, respectively. We used a denoising AE-based model [25] with an encoder and decoder consisting of fully connected layers. The AE models in this study possessed a single hidden layer with 5000 nodes. The weights of the encoder and decoder layers were tied in that the decoder's weight matrix was the transpose of the encoder's throughout the model's training. For input noising, masking noise was added to the input data [27], such that a percentage of every input vector's elements was randomly forced to zero in every epoch. Four noise levels (0%, 30%, 50%, and 70%) were used, and the loss between the AE output of noisy input and the original "clean" data was calculated (Fig. 1b). The goal of AE training involved minimizing a loss function based on the average reconstruction error [25] as depicted in the equation below:

$$\theta^*, \theta'^* = \underset{\theta^*, \theta'^*}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L\left(x^{(i)}, z^{(i)};\ \theta^*, \theta'^*\right), \tag{1}$$

where $\theta^* = \{W, b\}$ and $\theta'^* = \{W', c\}$ are the parameters (i.e., weights and biases) at the encoder and decoder (reconstruction) layers, respectively, L is the loss function parameterized by θ^* and θ'^*, and $x^{(i)}$ and $z^{(i)}$ are the respective input and reconstructed data. The learning rate for Eq. (1) settled at 10^{-2} and L2 regularization supplemented the overall loss term with a parameter of 10^{-3}. A stochastic

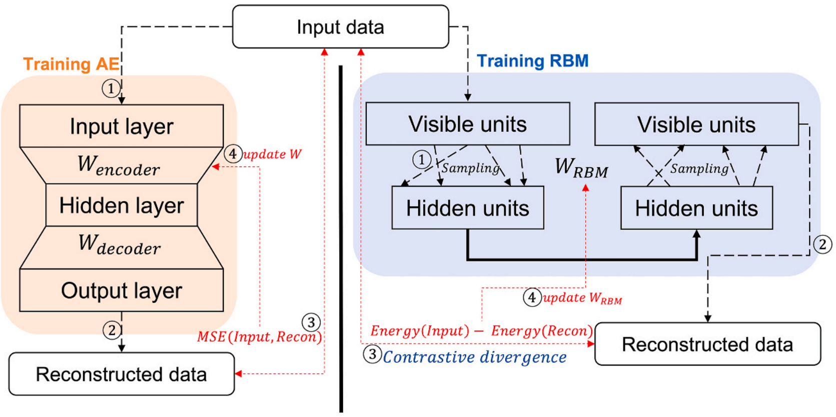

Fig. 2. Flowcharts of the training procedures for AE (left) and RBM (right). Preprocessed fMRI data represents the input. In AE, the weights and biases of the encoder and decoder layers are trained to minimize the mean-squared error (MSE) between the input and output (i.e., reconstructed input) patterns. In RBM, the weights and biases are trained to minimize the negative log-likelihood of the visible nodes based on a stochastic gradient descent scheme with a contrastive divergence and Gibbs sampling chain approximation.

gradient descent optimizer served with a momentum parameter of 0.3. The code incorporated python's PyTorch library (www.pytorch.com).

RBM is an unsupervised model to obtain a building block of DNN for supervised downstream tasks such as classification, in which the visible nodes are projected to hidden units (i.e., latent vector) with a reduced dimensionality [28–30]. The RBM based models effectively handle brain images because of their capacity to simplify high dimensional, functional imaging data while maintaining functional information of the brain [14,31,32]. The RBM used in this study consisted of 52,470 visible and 5000 hidden nodes/units, matching the AE's architecture (i.e., a single fully connected layer with the same numbers of input and hidden nodes). RBM training comprises two steps. First, we define visible node likelihood using the joint probability of the visible and hidden nodes, as an exponential energy function:

$$p(v) = \sum_h p(v,h) = \sum_h \frac{1}{Z} e^{-E(v,h)}, \tag{2}$$

where the partition function $Z = \sum_{v,h} e^{-E(v,h,W)}$, W is the weight matrix of the network, and E is the energy function. For real-valued data with Gaussian noise such as fMRI data, the energy function in Eq. (2) is as follows:

$$E(v,h) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j c_j h_j - \sum_{i,j} \frac{h_j w_{ji} v_i}{\sigma_i}, \tag{3}$$

where v_i and h_j denote the i th visible and j th hidden units; σ_i is the standard deviation of the Gaussian noise for the visible node v_i; b and c are biases of visible and hidden units; and w_{ij} are the weight parameters. Using Eqs. (2) and (3), the weight parameters can be updated by minimizing the negative log-likelihood using gradient descent scheme with contrastive divergence and Gibbs sampling [26,33], as in the equation below:

$$\Delta w_{ji} \propto \frac{1}{N} \sum_{n=1}^{N} \frac{\partial \log p(v^n)}{\partial w_{ji}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}, \tag{4}$$

where N is the number of training samples and $\langle \cdot \rangle$ is an expectation operator under the distribution specified by the subscript. Additionally, update rules for biases of visible and hidden units are as follows:

$$\Delta b_i \propto \frac{1}{\sigma_i^2} v_i^0 - \frac{1}{\sigma_i^2} v_i^1, \tag{5}$$

$$\Delta c_j \propto h_j^0 - h_j^1. \tag{6}$$

The standard deviation of the Gaussian noise for the visible unit, σ_i was fixed at 1. The single-layer RBM was trained for 15 epochs using Eqs. (4)–(6), with batch size 512; more detailed training procedure appear in our previous work [17]. The input noise level candidates for RBM mirrored those of AE (i.e., 0%, 30%, 50%, and 70%). The RBM code was implemented in MATLAB. The adopted AE and RBM codes are available in our GitHub repository (https://github.com/bsplku/dnnwsp).

2.4. Weight sparsity control

We utilized an explicit sparsity term that sets a target sparsity level for weights, adaptively changing the regularization parameter to reach and maintain the target level [17,19,34]. By combining the fidelity loss, $Loss(\hat{x}, x)$, between the reconstructed input $\hat{x}$ and original input x with the adaptive L1 regularization term, the overall loss term emerges as follows:

$$L = Loss(\hat{x}, x) + \beta_j \sum_{j=1}^{m} \|W_j\|_1 \tag{7}$$

where m is the number of hidden layer nodes, $\| \bullet \|_1$ is the L1 norm, W_j is the j th node's weight, and β_j is that node's regularization parameter. β_j values in Eq. (7) are initially set to zero and then updated at each mini-batch, based on the difference between the current and target levels of weight sparsity as described below:

$$\Delta\beta_j = \mu_\beta * sign(\rho_w - \hat{\rho}_j), \tag{8}$$

where μ_β is β_j's learning rate (set to 10^{-5} in this study), $sign(\bullet)$ is the sign function, and ρ_w and $\hat{\rho}_j$ are the target and current weight sparsity levels, respectively. Using Eq. (8), the regularization parameter decreases as the current sparsity level approaches the target and falls to zero when it reaches or exceeds the target. Hoyer's sparsity measure adequately reflects the sparsity of weights [35]:

$$\hat{\rho}_j = \frac{\sqrt{N} - \|W_j\|_1 / \|W_j\|_2}{\sqrt{N} - 1}, \tag{9}$$

where N is the number of weight parameters and $\| \bullet \|_2$ is the L2 norm of the weight features respectively. Hoyer's sparsity measure defined in Eq. (9) ranges from 0 to 1, higher values corresponding to greater sparsity. In this study, we establish the target sparsity levels at 0.8 and 0.9 for reproducibility (Kim et al., 2020), with an additional level of 0.5 for comparison.

2.5. Weight feature extraction and visualization

The model weight matrices were compared for the sparsity and noise level parameters. Each encoder's trained weight matrix was first z-scored across all elements and threshold-set at $|Z| > 1.96$. Significant in-brain voxels were then assigned a node number if the corresponding weight value was the highest among the nodes, as depicted in the equation below:

$$w_v^{(node)} = \underset{h=1\ldots H}{\operatorname{argmax}}\left(\widetilde{W}_{v\times h}\right), \tag{10}$$

where $\widetilde{W}$ is the thresholded weight matrix, $v = 1, \ldots, V$ is the voxel index (the total number of in-brain voxels, i.e., 52,470), and $h = 1, \ldots., H$ is the hidden-node index (the number of nodes, i.e., 5000). Thereafter, the resulting 52,470 × 1 vector $w^{(node)}$ was obtained from Eq. (10), in which entries were zero if the corresponding voxel had neither significant value weights nor a node index (1 to H), was obtained as a node assignment map and projected back to 3D space using the group in-brain mask for visualization.

Additionally, spatial overlap (O) between the threshold weight features and each of the seven FNs parcellations [22] was calculated [36]:

$$O\left(\widetilde{W}_j, \widehat{F}_i\right) = \frac{n\left(\widetilde{W}_j \cap \widehat{F}_i\right)}{n\left(\widehat{F}_i\right)}, \tag{11}$$

where $\widetilde{W}_j$ is the threshold-set weight map of the jth hidden node, F_i is the ith FN, and $n(\bullet)$ is the function that counts the number of voxels. This comparison was performed for each of the seven FNs and all networks combined (denoted as $\widehat{F}_{all}$).

Meanwhile, the significant voxels might not overlap fully with each FN. Thus, we calculated another measure, redundancy (R), to quantify voxel-FN independence. Redundancy was calculated as a voxel ratio, the sum of those in a threshold-set node assignment map of weight features not overlapping with the FNs to the total number of voxels in the weight features map:

$$R\left(\widetilde{W}_j, \widehat{F}_i\right) = \frac{n\left(\widetilde{W}_j - \left|\widetilde{W}_j \cap \widehat{F}_i\right|\right)}{n\left(\widetilde{W}_j\right)}. \tag{12}$$

2.6. rfMRI-based transfer learning scheme for task condition classification

We employed feedforward neural networks (FNN) with two hidden layers as the NN architecture for transfer learning since FNNs with two hidden layers performed as universal function approximators [37] and have been more advantageous than the shallow NN with one hidden layer for function approximation [38]. We denoted the FNN with more than one hidden layer as DNN following the definition in previous studies [39–42]. There were 5000 and 1000 nodes at the first and second hidden layers, respectively, in our DNN (Fig. 1b). The AE/RBM encoder weights were then transferred to the first layer of a DNN classifier. Using the trained weights from both the AE and RBM models with $\rho_w = 0.8$ and 0% input noise level, we trained the DNN classifier in two fashions: (1) fixing the first-layer weights during training (fixed condition) and (2) updating during training (fine-tuned condition). The retest subject tfMRI data was used as the test set to evaluate DNN classification performance. The performance was then compared to a DNN model (with the same architecture) trained with random initial weights and matching the target sparsity level (i.e., 0.8 at the first layer). The target sparsity level for the second layer was set at three different values (0.3, 0.5, and 0.8) to compare model classification performance depending on the second layer's target sparsity level.

We divided training data into five-fold, in which four folds were used for training the DNN model and one remaining fold for validation of the trained DNN. In the training phase, we employed early stopping, confirming no increase in validation error for 200 consecutive epochs calculated using the validation fold to circumvent an overfitting issue. After training, DNN weight features were extracted to visualize and interpret classification performance. We also applied logistic regression in each task as an alternative machine learning model to the proposed transfer learning DNN. We tested all seven HCP task conditions: the working memory task, which has eight classes: face, places, tools, and body with 0-back/2-back conditions (i.e., participants need to respond whether the current stimulus is the same as the pre-defined target stimulus [0-back] or the same as the two stimuli earlier [2-back]) respectively [2]; the motor task conditions, which has five classes: left foot, right foot, left hand, right hand, and tongue movements (i.e., participants were asked to tap their fingers, squeeze their toes, or move their tongue) [43]; and other five binary classes (emotion task: face vs. shape; relational task: relational vs. match; social task: social vs. random; language task: story vs. math; gambling task; reward vs. punishment) [44–48]. The reproducibility of task classification performance for the DNN with each transfer learning scheme was evaluated using the test data from S900 and retest data from S1200 datasets of the 45 test-retest subjects. We employed a paired t-test for statistical comparison of the performance between the models, since each model's prediction performance was obtained from the same group of subjects and we assumed adherence to a normal distribution.

3. Results

3.1. Overlap ratio of group-level in-brain mask and Yeo's seven FNs

The group-level in-brain mask (M) and the group of seven FNs (F_{all}) had an overlap ratio of 0.7, with non-overlapping areas mainly in the white matter (WM) and cerebrospinal fluid (CSF) regions (Fig. 3a). The seven FNs belonging to in-brain voxels were defined ($\widehat{F}_{all}$; Fig. 3b) and the overlapping voxels between parcellated brain regions and each FN as well as their ratios were calculated for each of the seven FNs in $\widehat{F}_{all}$ (Fig. 3c). The number of voxels varied substantially across the FNs; the maximum number was observed in the default-mode (DM) network, followed by the visual (VIS) network, whereas the overlap ratios remained relatively stable across the FNs.

3.2. Parcellation of brain regions using an explicit weight sparsity-controlled AE/RBM

Tables 1 and 2 summarize the number of nodes assigned within each of the seven FNs across all target weight sparsity scenarios and input noise levels in AE and RBM, respectively.

Fig. 4 illustrates the node assignment map for AE with $\rho_W = 0.8$ and 50% input noise level. The overlap with $\widehat{F}_{all}$ was 0.9 or higher except for the VA and LIM FNs (0.78 and 0.14 respectively). Within each FN, only a small fraction of nodes out of the possible 5000

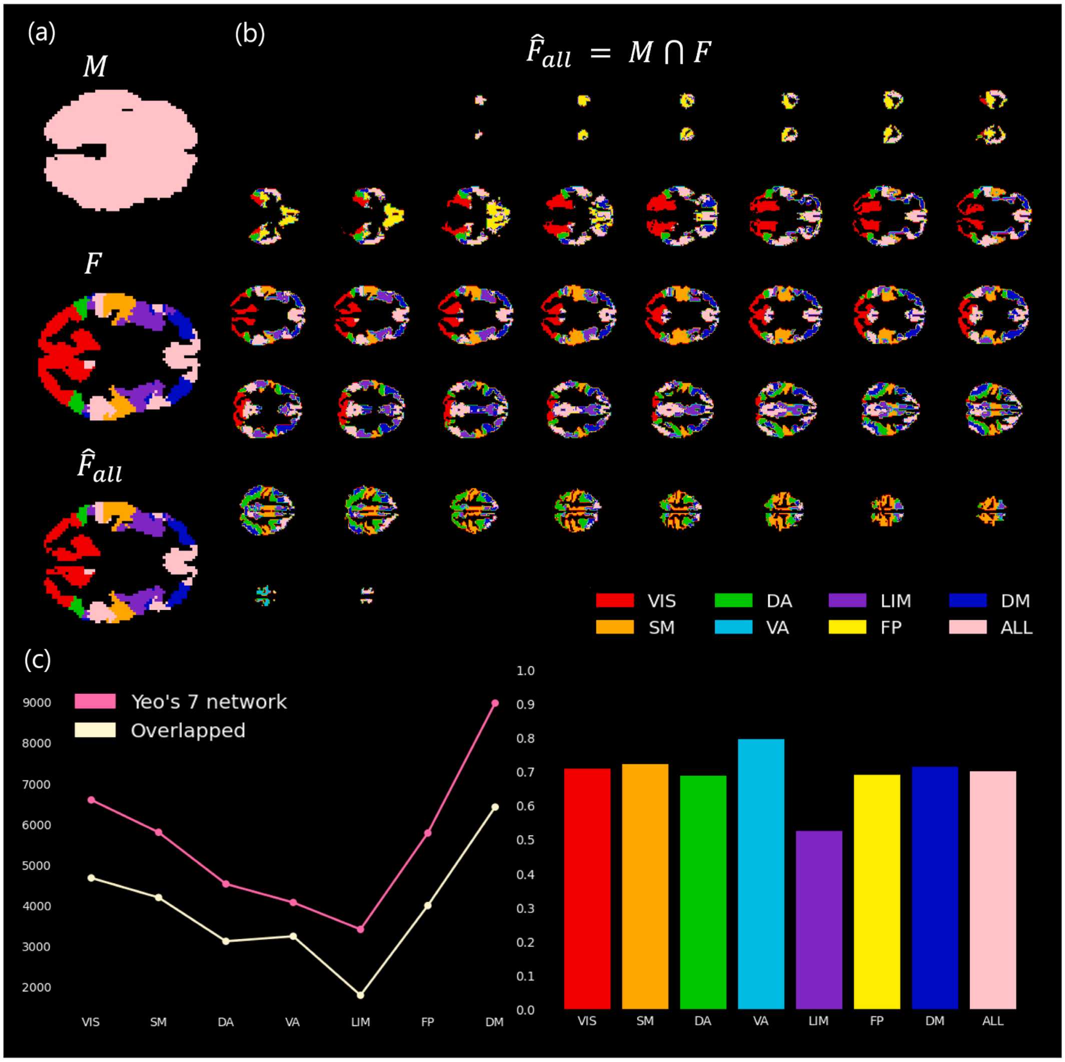

Fig. 3. Group level in-brain masks M for functional networks (FNs) defined based on Yeo's seven FNs (i.e., $F_{ALL} = F_{VIS} \cup F_{SM} \cup F_{DA} \cup F_{VA} \cup F_{LIM} \cup F_{FP} \cup F_{DM}$; [22]) and the seven FN masks in our defined in-brain area (i.e., $\widehat{F}_{ALL} = M \cap F_{ALL}$): (a) Exemplary slice of these masks, (b) entire slices of reference Yeo's seven FNs (i.e., $\widehat{F}_{ALL}$), and (c) the number of overlapping voxels with Yeo's seven FNs (left) and the corresponding ratios of the number of voxels to all voxels in each of the seven FNs (right).

were assigned, particularly for VIS and somatomotor (SM). Fig. 5 demonstrates RBM's node assignment map with $\rho_W = 0.8$ and 50% input noise level. Notably, a significant additional number of hidden nodes in RBM as compared to AE were assigned to each of the seven FNs. Fig. 6 reveals the 5000 parcellated ROIs using the AE and RBM models with $\rho_W = 0.8$ and 0% input noise, clearly illustrating the greater extent of parcellated ROIs for each hidden node in AE and fine-grained parcellation for each hidden node in RBM.

Table 3 summarizes the overlap ratios obtained from Eq. (11), between node-assigned weight feature maps of AE and $\widehat{F}_{all}$ across all weight sparsity (0.5, 0.8, and 0.9) and input noise levels (0%, 30%, 50%, and 70%). Overall, a sparsity target of 0.5 displayed the highest overlap with the FN map, decreasing as sparsity target increased. The input noise level also influenced the overlap ratios, with an inverse ratio to a given sparsity level; this was more pronounced in higher sparsity levels. Overlap ratios of 100% with the seven FNs appeared in the RBM-based weight feature maps, regardless of target sparsity and noise level, possibly because of the fine-grained weight features covering the whole-brain. Table 4 presents the AE redundancy obtained from Eq. (12), with elevated sparsity/noise levels resulting in less redundancy. Using $\rho_w = 0.5$ and no input noise, redundancy was 0.48, implying that nearly half of the voxels in the node-assigned weight maps were not included in the seven functional FNs, belonging instead to non-gray matter (i.e., WM and CSF). The redundancy of RBM equaled 0.48 regardless of target sparsity or input noise level, corresponding to the ratio of WM and CSF areas in the group-level in-brain mask.

3.3. *Task fMRI classification using pre-trained AE and RBM weights*

Logistic regression performances were consistently inferior compared to the proposed DNN methods. Specifically, logistic regression reached 65.2 ± 0.1 in the working memory task (vs. 75.0 ± 0.2 for DNN with fixed RBM weights), 93.6 ± 0.2 in the emotion task (95.5 ± 0.1 for DNN with fine-tuned AE weights), 92.2 ± 0.3 in the motor task (94.1 ± 0.2 for DNN with fine-tuned AE weights), 75.7 ± 0.1 in the relational task (78.1 ± 0.4 for DNN with fixed RBM weights), 87.3 ± 0.1 in the social task (91.3 ± 0.1 for DNN with fixed RBM weights), 94.2 ± 0.1 in the language task (96.1 ± 0.1 for DNN with fixed RBM weights), and 62.7 ± 0.2 in a gambling task (67.4 ± 0.3 for DNN with fixed RBM weights).

Fig. 7 depicts the overall classification accuracies of the DNN models with target sparsity of 0.8 and 0.5 at the first and second layer, respectively. The accuracies obtained from the transfer learning schemes were subtracted from the DNN-trained accuracies with random initial weights as baseline (i.e., a positive or negative value indicates enhanced or degraded performance respectively). Overall, the DNN transfer learning with fixed RBM weights generally yielded higher accuracies than DNNs without the transfer learning scheme, particularly for the working memory, social, and gambling tasks. The paired *t*-test using test subject accuracies on these three tasks resulted in *t*-scores of 11.4 for working memory, 11.5 for social, and 7.3 for gambling (corrected $p < 10^{-3}$). Alternatively, the fixed AE condition demonstrated the worst overall performance. This was most obvious in the motor task: with 87.9% accuracy in the test dataset, 6.1% less than the DNN trained with random initial conditions. The fine-tuned DNN with initial RBM weights presented slightly lower classification accuracy in all tasks, while the fine-tuned AE condition demonstrated comparable or higher classification accuracy compared to the baseline DNN. Table 5 summarizes the *t*-test results using the accuracies from DNN transfer learning compared to the DNN with random initial weights. Overall, transferred DNN classification performance was relatively consistent across the second-layer sparsity levels.

Furthermore, the computational load was reduced substantially for DNN transfer learning by fixing the pre-trained AE/RBM weights in the first layer. More specifically, the number of trainable parameters (i.e., weights and biases) of the baseline/fine-tuned DNN model was 267,358,002 ($[52{,}470 + 1] \times 5000 + [5000 + 1] \times 1000 + [1000 + 1] \times 2$ for DNN with two output nodes), whereas that of the DNN with fixed weights in the first layer was 5,003,002 ($[5000 + 1] \times 1000 + [1000 + 1] \times 2$), approximately 1.9% of the number of parameters in the baseline/fine-tuned DNN.

Fig. 8 visualized the test-retest reproducibility of the task classification performance as bar plots. Overall, it appears that the classification performance between test data and retest data is highly reproducible. Specifically, the FixRBM and FixAE models showed superior and inferior performance, respectively than the baseline model and alternative transfer learning schemes consistently for both the test data and retest data.

Table 1
The number of assigned nodes with each of the seven FNs across all scenarios of target sparsity and input noise levels in AE.

		Functional network							
		%	Visual	Somato-motor	Dorsal attention	Ventral attention	Limbic	Fronto-parietal	Default mode
Hoyer's sparsity target (ρ_w)	0.5	0	124	124	127	162	45	169	174
		30	69	44	113	113	52	118	111
		50	51	37	70	92	50	80	97
		70	38	28	57	56	35	65	96
	0.8	0	83	68	99	100	57	123	125
		30	46	41	64	79	45	87	104
		50	37	40	56	63	23	69	100
		70	20	20	40	39	11	41	60
	0.9	0	49	60	73	65	6	103	114
		30	35	32	55	44	0	61	82
		50	21	27	38	22	0	40	42
		70	5	3	6	9	0	7	7

FN: functional brain network; AE: autoencoder; %: input noise level.

Table 2
The number of assigned nodes with each of the seven FNs across all scenarios of target weight sparsity and input noise levels in RBM.

		Functional network							
		%	Visual	Somato-motor	Dorsal attention	Ventral attention	Limbic	Fronto-parietal	Default mode
Hoyer's sparsity target (ρ_w)	*0.5*	*0*	926	942	776	872	506	1005	1396
		30	896	936	788	866	520	1026	1417
		50	910	920	741	850	493	1026	1361
		70	745	766	617	633	373	874	1106
	0.8	*0*	498	542	476	534	285	647	871
		30	505	573	490	566	287	651	891
		50	517	551	506	568	305	673	926
		70	475	530	491	528	270	625	861
	0.9	*0*	508	528	475	512	251	628	845
		30	516	536	494	534	274	659	868
		50	504	543	505	567	259	686	904
		70	485	567	466	517	252	619	862

FN, functional brain network; RBM, restricted Boltzmann machine; %, input noise level.

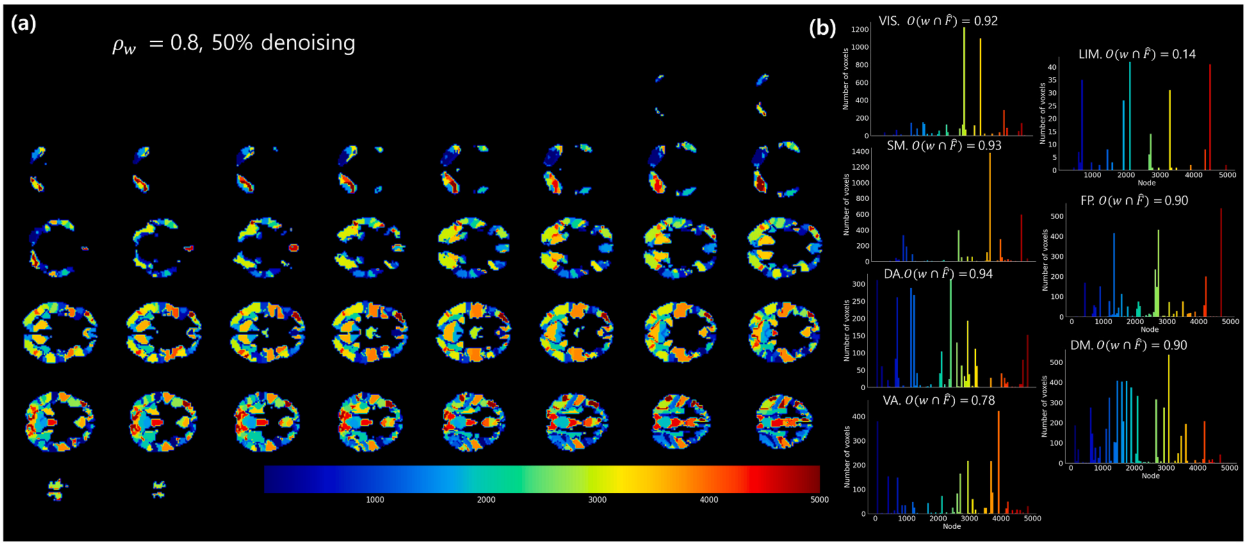
(a)
ρ_w = 0.8, 50% denoising
1000
2000
3000
4000
5000
(b)
VIS. $O(w \cap \hat{F})$ = 0.92
LIM. $O(w \cap \hat{F})$ = 0.14
SM. $O(w \cap \hat{F})$ = 0.93
FP. $O(w \cap \hat{F})$ = 0.90
DA. $O(w \cap \hat{F})$ = 0.94
DM. $O(w \cap \hat{F})$ = 0.90
VA. $O(w \cap \hat{F})$ = 0.78
Number of voxels
Node

Fig. 4. (a) Node assignment results of AE encoding weights for weight sparsity target $\rho_w = 0.8$ with 50% input noise level. (b) Each bar graph shows the number of voxels assigned to each of the seven FNs in each hidden node of trained AE.

3.4. Weight feature visualization of the DNNs

Fig. 9 captures the weight feature map of the fine-tuned DNN with initial AE weights (Finetune AE), the DNN with fixed first-layer weights using the RBM weights (Fixed RBM), and the DNN trained with random initial weights (Random) for a target sparsity levels of 0.8 and 0.5 at the first and second layers respectively, across all seven tasks. Overall, all three models represented similar brain regions that were crucial to each task's classification. Nonetheless, Fixed RBM revealed more clustered brain regions in their weight features than the other two scenarios. Finetune AE showed slightly more clustered brain regions than Random. Notably, Fixed RBM exhibited the highest spatial overlap with the previously reported group activation map using the same HCP dataset [49]. This suggests that Fixed RBM for DNN transfer learning captured the characteristics of each task better than both the Finetune AE scenario and baseline DNN model.

4. Discussion

4.1. Summary

In this study, we described that the AE and RBM models extracted parcellated brain regions from resting-state fMRI data which highly overlapped with Yeo's seven FNs [22]. Additionally, with the node assignment map, we reported that the RBM model learned more fine-grained FNs compared to the AE. When the DNN was fine-tuned to classify the HCP dataset's tfMRI data by initializing its first-layer weights with those from the AE and RBM models, the fine-tuned AE weight DNN performed slightly better than the DNNs fine-tuned with RBM weights or random initial weights. Conversely, when the DNN was trained with fixed first-layer weights from the

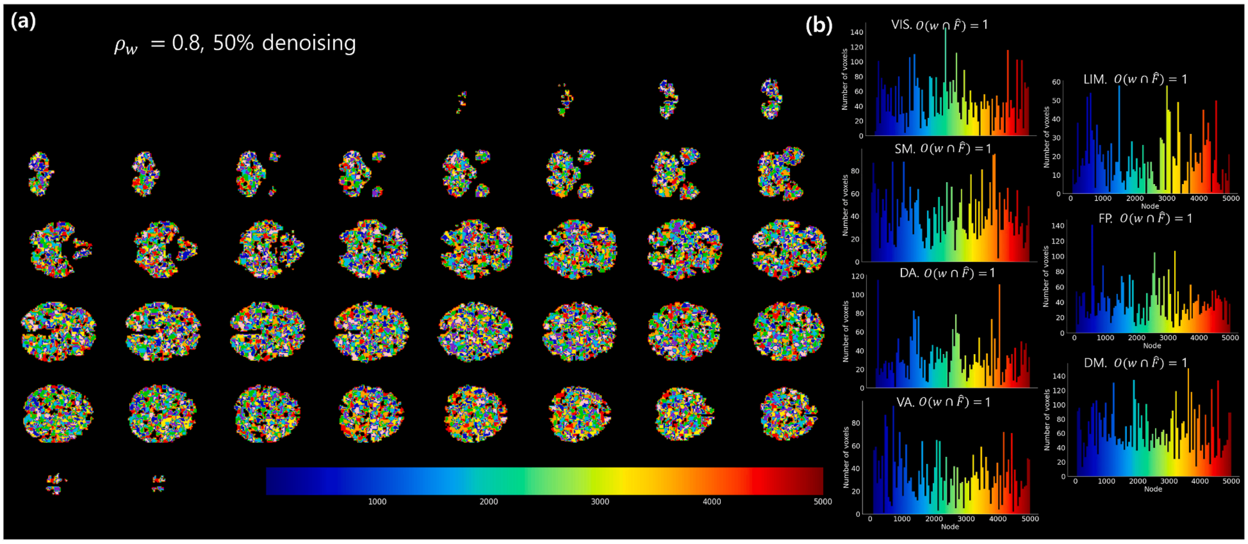

Fig. 5. (a) Node assignment results of RBM encoding weights for target weight sparsity $\rho_w = 0.8$ with 50% input noise level. (b) Each bar graph shows the number of voxels assigned to each of the seven FNs in each hidden node of trained RBM.

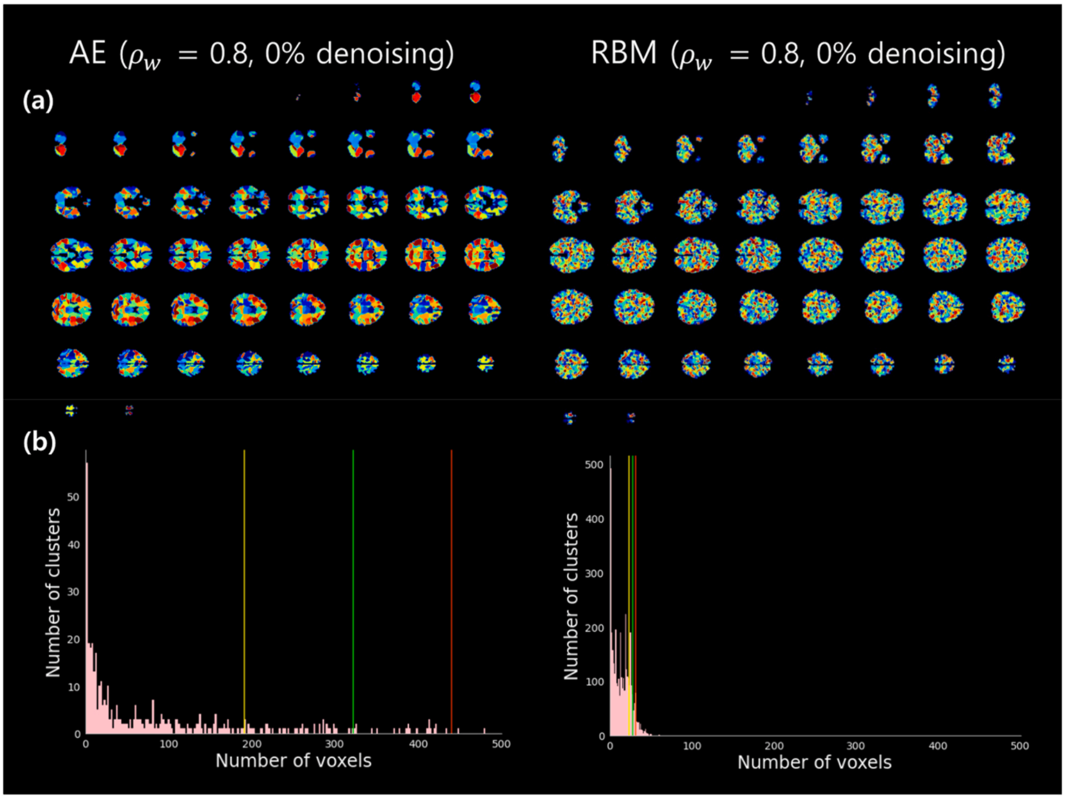

Fig. 6. (a) Comparison of parcellated brain regions corresponding to the hidden nodes from the AE and RBM weights when $\rho_w = 0.8$ with 0% input noise level. (b) Distributions of cluster sizes (i.e., the number of voxels in each brain region parcellated from a hidden node). from AE (left) and RBM (right). The three vertical lines denote 80th (yellow), 90th (green), and 95th (red) percentile thresholds for the cluster sizes. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

AE and RBM models, the DNN fixed with the RBM weights yielded better general performance than the DNNs fixed with AE weights or random initial weights, particularly for the working memory, social, and gambling tasks. Significantly, considering the number of weight parameters to be adjusted during DNN training, the DNN with fixed first-layer RBM weights appears the most suitable approach

Table 3

Overlap between node-assigned weight maps of AE (i.e., w) and the seven FNs (i.e., $\widehat{F}_{all}$), which was defined as: $O(w, \widehat{F}_{all}) = \frac{|w \cap \widehat{F}_{all}|}{|\widehat{F}_{all}|}$.

		Input noise level			
		0%	30%	50%	70%
Hoyer's sparsity target (ρ_w)	*0.5*	1.0	0.99	0.97	0.96
	0.8	0.98	0.92	0.85	0.69
	0.9	0.70	0.49	0.30	0.08

AE: autoencoder; FNs: functional brain networks.

Table 4

Redundancy of node-assigned weight maps of AE (i.e., w) and the seven FNs (i.e., $\widehat{F}_{all}$): $R(w, \widehat{F}_{all}) = \frac{w - |w \cap \widehat{F}_{all}|}{|\widehat{F}_{all}|}$.

		Input noise level			
		0%	30%	50%	70%
Hoyer's sparsity target (ρ_w)	*0.5*	0.48	0.48	0.45	0.39
	0.8	0.44	0.32	0.26	0.16
	0.9	0.16	0.10	0.06	0.03

AE: autoencoder; FNs: functional brain networks.

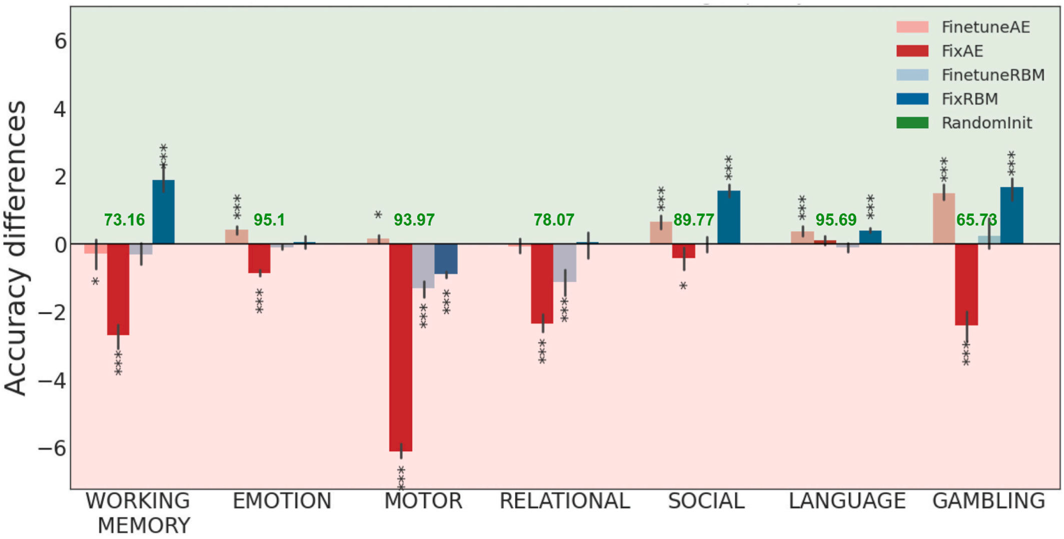

Fig. 7. Classification performance for the DNN models trained via transfer learning either by fixing the first-layer AE/RBM weights (i.e., FixAE or FixRBM), or by fine-tuning the weights after initializing the first-layer AE/RBM weights (i.e., FinetuneAE or FinetuneRBM). The differences denote the classification accuracy for the DNN with transfer learning minus the accuracy for the DNN initialized with random weights (numbers in the figure). The target weight sparsity levels were 0.8 and 0.5 at the first and second layer respectively without input noising. The p-values from the paired t-test were denoted as asterisks (*: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$). The t-statistics and corresponding p-values of the paired t-test results appear in Table 5.

for DNN transfer learning and whole-brain fMRI volume classification, compared to fine-tuned DNN transfer learning and DNN training initialized with random weights.

4.2. *Capacity of the parcellated brain regions using rfMRI data from the AE/RBM models*

The RBM pre-trained weights using rfMRI data and consequent DNN transfer learning performed superbly in classifying task conditions from tfMRI data when the transferred encoder weights of the RBM remained fixed during training. However, accuracy degraded when the encoder weights of the AE remained fixed during training (Fig. 7). We did not update the Gaussian noise for the visible units although the corresponding update of the Gaussian noise may further enhance the performance of the RBM despite the difficulty in setting the corresponding learning rate [50]. Thus, the numbers of trainable parameters for the AE and RBM were the same (i.e., parameters in the weight matrix between input and hidden layers, bias vector of the input layer, and bias vector of the hidden layer). In this context, this striking difference in performance likely resulted from the distinct characteristics of the pre-trained AE and

Table 5

Paired *t*-test results of test dataset in all seven tasks. The *t*-statistics of the accuracies from the DNN with transfer learning compared to the DNN with random initial weights were shown (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$).

Transfer learning scheme of the DNN													
		Finetune AE			Fixed AE			Finetune RBM			Fixed RBM		
Second layer sparsity		0.3	0.5	0.8	0.3	0.5	0.8	0.3	0.5	0.8	0.3	0.5	0.8
Task	WM	−1.1	−2.5*	−2.8**	−9.6***	−11.8***	−7.9***	−6.0***	−1.8	−25.0***	12.0***	11.4***	13.6***
	Emotion	7.1***	5.7***	1.9	−8.6***	−7.2***	−8.7***	−0.9	−1.30	−4.87***	−0.6	0.5	−2.5*
	Motor	3.5***	2.5*	0.1	−25.8***	−25.1***	−23.6***	−13.5***	−12.9***	−11.94***	−9.2***	−8.5***	−9.1***
	Relational	3.0*	−0.5	−0.8	−11.2***	−9.9***	−11.7***	−6.2***	−5.5***	−6.47***	−1.6	0.2	−1.8
	Social	5.9***	6.9***	3.2*	−3.2*	−2.4	−3.2***	−2.7	−0.1	−2.47	12.1***	11.6***	11.3***
	Language	4.9***	6.6***	4.3***	−0.7	1.3	1.4	−4.1***	−1.9	1.38*	4.2***	5.5***	4.8***
	Gambling	3.8***	7.1***	−4.6***	−7.6***	−8.0***	−9.4***	−1.9	1.0	−0.92	6.8***	7.3***	6.3***

DNN: deep neural network; AE: autoencoder; RBM: restricted Boltzmann machine.

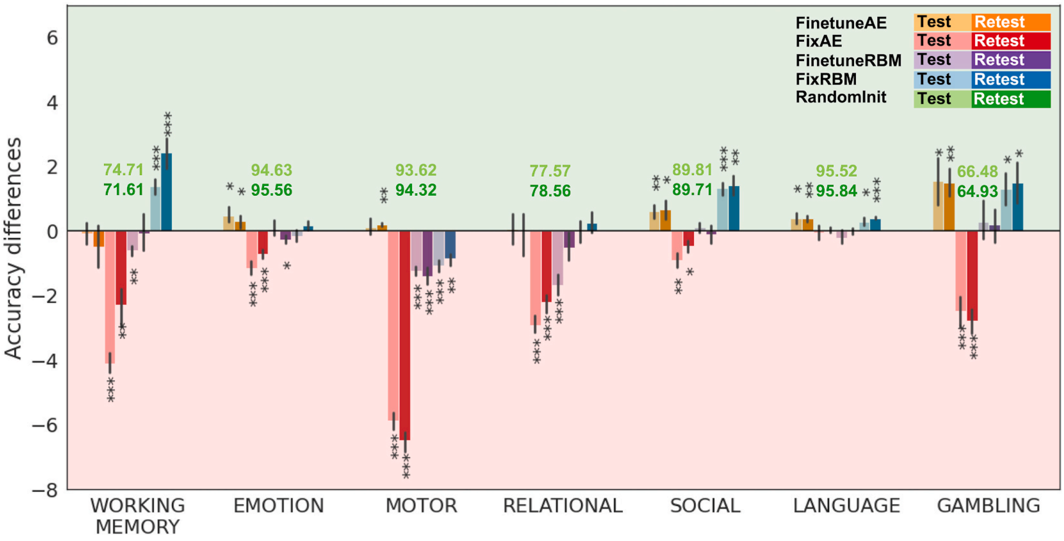

Fig. 8. Classification performance comparison between all task conditions using test samples from S900 (denoted as "Test") and retest samples from S1200 ("Retest") of the 45 test-retest subjects. The accuracy differences denote the classification accuracy for the DNN with transfer learning minus the accuracy for the DNN initialized with random weights (numbers in the figure). The target weight sparsity levels were 0.8 and 0.5 at the first and second layers, respectively without input noising. The *p*-values from the paired *t*-test were denoted as asterisks (*: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$).

RBM weights, which were fixed during the DNN training in the transfer learning (Fig. 1c; weights in red).

More specifically, based on Fig. 1c, the DNN classifier requires a mapping function between the input and output layers (defined as W_{all}). When we fixed the first layer weights ($W_{encoder}$, i.e., W_{AE} or W_{RBM}), the subsequent layer's weights necessitated training to identify the weight representation W: i.e., $W_{RBM} * W \cong W_{all}$ for RBM, and thus $W \cong pinv(W_{RBM}) * W_{all}$, where $pinv(\bullet)$ is a pseudo-inverse function. Therefore, the stability of $pinv(W_{RBM})$ or $pinv(W_{AE})$ seems crucial to finding a classification solution using the transferred DNN. We performed singular value decomposition on the weight matrices to evaluate the pseudo-inverse function's stability as obtained from the AE and RBM. It emerged that the singular values of the AE weights were several orders of magnitude smaller than those of the RBM weights (Fig. 10). This occurred possibly due to the large, substantially overlapping clusters (i.e., parcellated brain regions) from the AE weights, especially when compared to the fine-grained, minimally overlapping smaller clusters from the RBM weights (Figs. 4–6).

In addition to of the aforementioned magnitude difference, the condition number of the weight matrices (i.e., the ratio between the largest and smallest singular values) [51] may also affect the $pinv(W_{RBM})$ or $pinv(W_{AE})$ stability. Thus, we calculated the condition numbers, in which the AE weights presented a markedly higher value (41,041.42) than that of RBM (74.60). This suggests that the AE weight matrix may be ill-conditioned and thus, its inverse is potentially prone to significant numerical errors. Alternatively, the RBM weight matrix is well-conditioned and thus, its inverse can be computed with superior accuracy. The substantially different magnitude of the condition number between the AE and RBM weight matrices would directly impact transfer learning classification performance in a fixed-weight scenario. In the fine-tuning scenario, tracking the changes of singular values and weight matrix condition numbers in the first hidden layer could facilitate the interpretation of the model while training and subsequent effects on classification performance. In addition, training the model parameters with additional constraints to restrict ranges of singular values and condition numbers in the first layer would guide the weight matrices to be well-conditioned throughout the fine-tuning process for the DNN classifier. For example, singular value bounding methods [52] can be applied to maintain their singular values at a certain range and to prevent them from substantial magnitude reductions. This would guarantee that the weight matrix condition numbers remain in a proper range.

4.3. Weight feature maps of the DNN models obtained from transfer learning

We compared the classification performance for tfMRI data using a DNN trained via transfer learning with AE vs. RBM weights, at which point the sparsity of the first layer DNN weights was 0.8 without input noising. Overall, weight feature maps of the transfer learning DNNs displayed highly task-relevant brain regions that were similarly reported from the group activation patterns using the same dataset [49]. Among the DNN models, the network trained using the RBM fixed first-layer weights exhibited the most similar patterns with the group inference maps, moderately overlapping with task-relevant regions, particularly for the motor and social tasks.

Yet the DNN models fined-tuned from the AE weights or trained with random initial weights presented a reduced spatial extent, possibly indicating that the models were vulnerable to variable task-related activation patterns across subjects and runs. This might be caused by overfitting the baseline and fine-tuned AE-based DNN models, which potentially can be alleviated by applying a regularization scheme such as input noising and/or dropout schemes. Thus, future research is warranted to evaluate the transferred DNN performance, as well as to investigate how the resulting weight feature maps systematically depend on the weight sparsity levels and input/hidden layer noising parameters.

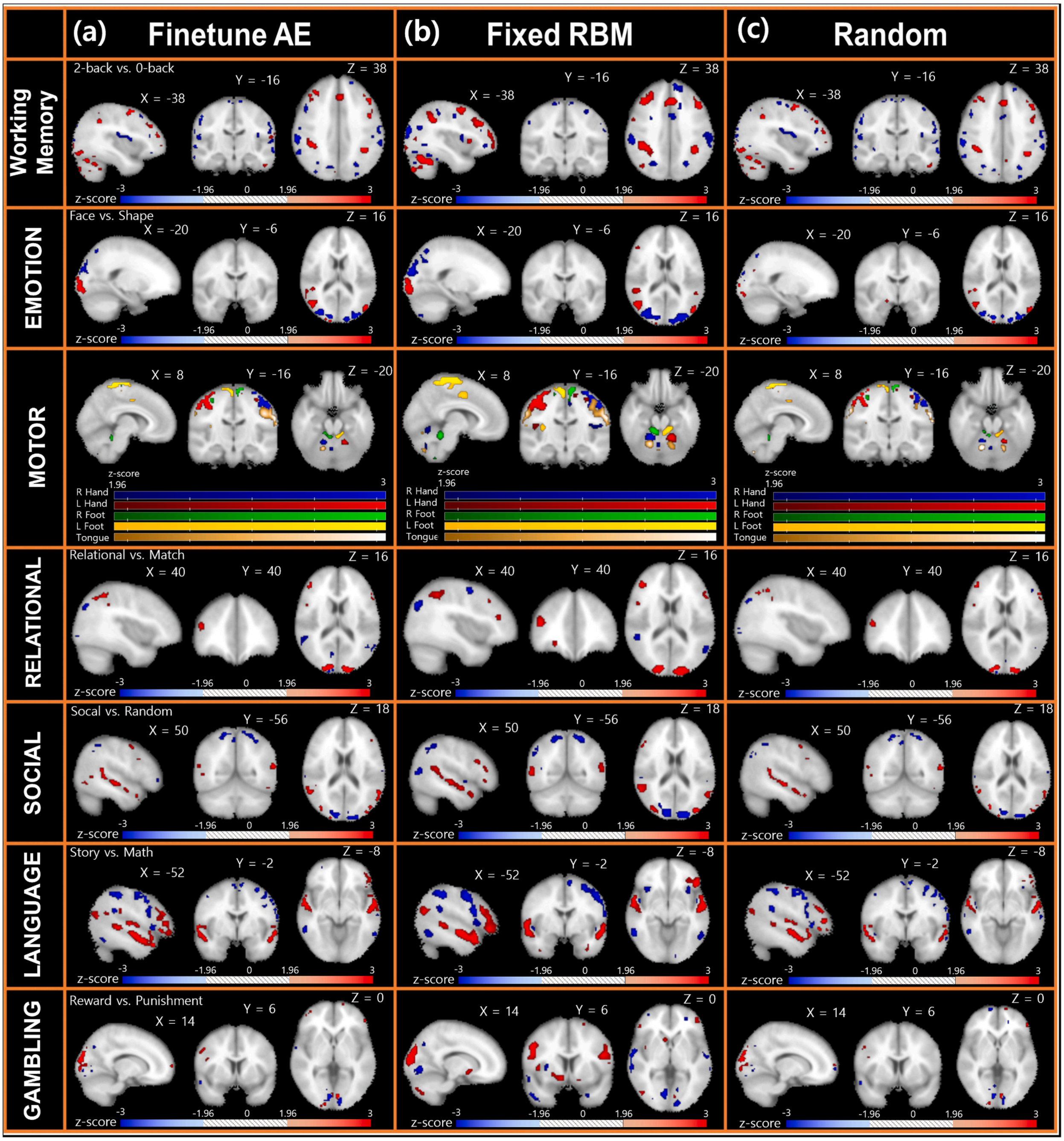

Fig. 9. Weight feature map of the DNN across all seven task conditions for the HCP dataset's tfMRI data. Weight feature maps of (a) fine-tuned AE, (b) fixed RBM, and (c) a baseline DNN trained with random initial weights are shown. Slices used to illustrate each task are matched to the group inference maps obtained using the same tfMRI dataset in the HCP [49], to facilitate comparison across the maps. Weight features were z-scored and threshold-set at the level of 95% to visualize important weights only. The clustering threshold was set to a minimum of ten voxels to remove potential false positives.

4.4. Potential limitations and further work

This study evaluated the classification performance of tfMRI data in the HCP dataset using a DNN trained via transfer learning with AE/RBM weights obtained from rfMRI data in the same dataset. Although our study attempted to utilize as much data as possible, the number of subjects ($n = 900$) may not be sufficient to train a sufficient number of weight parameters in AE/RBM for functional parcellation and DNN for classification. However, please note that there were four rfMRI runs and each rfMRI run consisted of

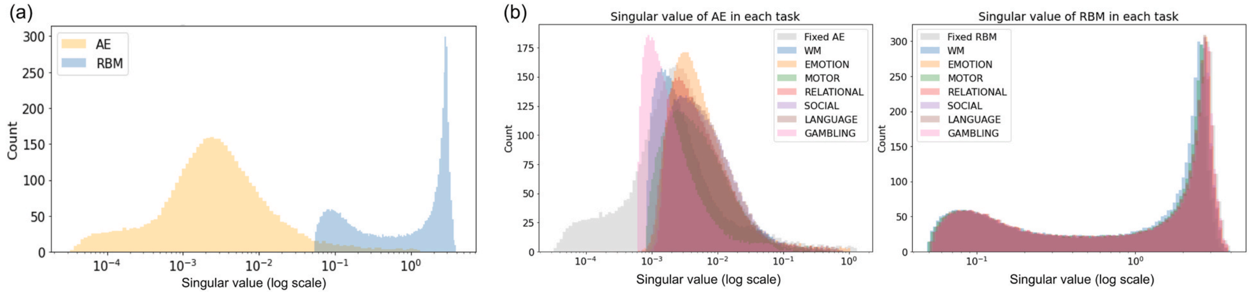

Fig. 10. (a) Distribution of singular values of the AE and RBM encoder weights, trained with the target weight sparsity $\rho_W = 0.8$ and no input noise. (b) Distribution of singular values of the first layer's weights in AE-based DNN (left) and RBM-based DNN (right) in all fine-tuning scenarios.

approximately 1200 fMRI volumes. Therefore, there are approximately 4800 vol per subject that served as input samples to train the AE/RBM models (i.e., 4,320,000 samples across all 900 subjects). For the task condition classification, the beta-valued map was the DNN's input, potentially suffering from the small number of samples per task condition to train all DNN weight parameters. This motivated us to evaluate the performance of the transfer learning by fixing the pre-trained AE/RBM weights, to reduce the number of adjustable parameters. Nonetheless, the DNN training may still suffer from the insufficient number of samples. Thus, it is crucial to evaluate whether our proposed transfer learning scheme approaches for the DNN using the pre-trained AE/RBM weights are also beneficial for alternative downstream tasks involving another publicly available dataset with a substantially increased number of subjects: ABCD (https://abcdstudy.org; $n = 11,875$) and UK Biobank (https://www.ukbiobank.ac.uk; $n = 38,645$ for tfMRI and 44,096 for rfMRI). Additionally, since the FN atlas used in this study was relatively coarse (containing only seven FNs), it would be worthwhile to compare the extracted weight features and node assignment maps with alternative, more fine-grained brain parcellations [5,6].

Additional layers may impact DNN performance [53,54], which poses concerns for explanation by studying their building blocks (i.e., hidden layers) in isolation. It would be equally true that the DNN performance is heavily affected by the weight parameter initialization methods [55,56]. Our study used the DNN as a predictive model for neuronal activations across the whole brain, and that DNN consists of a greater number of parameters in the first layer than in other layers due to the large number of whole brain voxels [17]. This would lead to potential overfitting due to the small number of samples to train the model in general, which may deteriorate prediction performance [34]. Thus, our objective for this study was to investigate systematically the DNN prediction performance for whole-brain neuronal activations depending on the initialization schemes of first layer weights. Consequently, we employed the two most popular unsupervised, pretraining models (i.e., AE and RBM) to initialize the DNN's first layer for whole-brain neuronal activations, systematically evaluating the DNN prediction performance. Altering DNN architectures such as adding the hidden layers or changing the number of nodes per hidden layer may further enhance the transfer learning performance at the expense of computational resources. It is worth noting that the DNNs used in this study are FNNs with two hidden layers, which are relatively shallow compared to the NNs in conventional deep-learning models. For the DNN with increased numbers of hidden layers, stacking multiple layers to AE and RBM in the parcellation step would further reduce the number of weight parameters to be trained and subsequently the training time in the DNN with transfer learning. Thus, a deep belief network with several stacked RBMs is worth investigating in the context of DNN transfer learning for the task fMRI classification from a large cohort dataset.

While both fine-tuned AE and fixed RBM-based transfer learning illustrated superior classification performance in most of the prediction tasks, neither model presented any performance improvement for the relational task. The possible cause appears to be the inability of the models to recognize important brain regions for the classification and discriminate the two potentially subtle conditions (i.e., relational vs. match) in the task. A previous study presented that both relational and match conditions elicited significant activation in the anterior prefrontal cortex [49], with a potentially subtle difference in their spatial layout. From our findings, the transfer learning of the DNN by using fixed RBM-based weights identified a relatively vast blob of the important brain regions (Fig. 9) which would be potentially disadvantageous when optimal classification of target task requires the identification of smaller brain regions. Therefore, several approaches may be applicable to optimize the classification performance by adjusting crucial brain region blob size identified from the DNN via transfer learning based on fixed RBM-based weights. For example, weight sparsity levels (e.g., 0.5 and 0.9) deviating from the adopted sparsity (0.8) when training the RBM weights could increase or reduce the size of the functional parcellations, which will tangibly impact the final blob size of the critical brain regions from the trained DNN. In addition, hyperparameters to alter the model architectures, such as the number of hidden nodes at the first layer (i.e., the same as the number of parcels from the AE/RBM weights) and the subsequent layers would provide varying sizes of parcellated brain regions and possibly alter the final blob size of the critical brain regions for classification.

5. Conclusion

We evaluated classification performance for tfMRI data using a DNN transferred using pre-trained weights obtained from the AE and RBM. To this end, we trained the AE and RBM using the HCP dataset's whole-brain rfMRI data. The resulting weight features were systematically evaluated according to the weight sparsity and input noise levels of the AE/RBM. Similarly, the parcellated brain regions from the obtained weight features were interpreted carefully compared to Yeo's seven FNs [22]. The AE segmented the whole brain into broad parcels with considerable overlapping between the parcels via their weights, whereas the RBM segregated fine-grained regions with minimal overlapping between the parcels. These distinct characteristics of the AE and RBM weight features may have affected the classification performance. Specifically, the DNN classifier trained by fixing the RBM weights in the first layer depicted superior classification over the DNNs trained by fixing the AE weights and fine-tuning them in the first layer. These findings propose that the DNN classifier trained by fixing the first layer RBM weights is the most promising approach, since the model is advantageous in computational load and memory usage while maintaining or improving classification performance. We believe that our proposed DNN classifier based on pre-trained RBM weights will prove useful for the classification of whole-brain fMRI data from both large and small cohorts, as the number of free parameters reduces substantially by applying our proposed transfer learning scheme (fixing the pre-trained RBM weights). More specifically, the computational load and memory size will be substantially reduced from our transfer learning scheme, especially compared to the DNN training with random initial weights.

Funding source

This work was supported by a National Research Foundation (NRF) grant, the Ministry of Science and ICT (MSIT) of Korea (NRF-2017R1E1A1A01077288, NRF-2021M3E5D2A01022515), in part by a National Research Council of Science & Technology (NST) grant provided by the Korean government (MSIT) [No. CAP18015-101], and by an Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [23ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System].

Data availability statement

The code and sample data supporting the findings of this study are available at https://github.com/bsplku/DNN_TL4fMRI. The resting-state fMRI and task fMRI data were obtained from the Human Connectome Project (HCP; http://www.humanconnectomeproject.org).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L.J. Garey (Ed.), Brodmann's' localisation in the cerebral cortex, World Scientific, 1999, pp. 1–9.

[2] P.E. Downing, Y. Jiang, M. Shuman, N. Kanwisher, A cortical area selective for visual processing of the human body, Science 293 (2001) 2470–2473, https://doi.org/10.1126/science.1063414.

[3] A.R. McIntosh, C.L. Grady, L.G. Ungerleider, J.V. Haxby, S.I. Rapoport, B. Horwitz, Network analysis of cortical visual pathways mapped with PET, J. Neurosci. 14 (1994) 655–666.

[4] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, M. Joliot, Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain, Neuroimage 15 (2002) 273–289.

[5] E.M. Gordon, T.O. Laumann, B. Adeyemo, J.F. Huckins, W.M. Kelley, S.E. Petersen, Generation and evaluation of a cortical area parcellation from resting-state correlations, Cerebr. Cortex 26 (2016) 288–303.

[6] X. Shen, F. Tokoglu, X. Papademetris, R.T. Constable, Groupwise whole-brain parcellation from resting-state fMRI data for network node identification, Neuroimage 82 (2013) 403–415.

[7] B.T. Yeo, F.M. Krienen, J. Sepulcre, M.R. Sabuncu, D. Lashkari, M. Hollinshead, J.L. Roffman, J.W. Smoller, L. Zöllei, J.R. Polimeni, et al., The organization of the human cerebral cortex estimated by intrinsic functional connectivity, J. Neurophysiol. 106 (2011) 1125–1165.

[8] M.F. Glasser, T.S. Coalson, E.C. Robinson, C.D. Hacker, J. Harwell, E. Yacoub, K. Ugurbil, J. Andersson, C.F. Beckmann, M. Jenkinson, A multi-modal parcellation of human cerebral cortex, Nature 536 (2016) 171–178.

[9] E.S. Finn, X. Shen, D. Scheinost, M.D. Rosenberg, J. Huang, M.M. Chun, X. Papademetris, R.T. Constable, Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity, Nat. Neurosci. 18 (2015) 1664–1671.

[10] M. Salehi, A. Karbasi, D.S. Barron, D. Scheinost, R.T. Constable, Individualized functional networks reconfigure with cognitive state, Neuroimage 206 (2020), 116233.

[11] U. Güçlü, M.A. van Gerven, Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream, J. Neurosci. 35 (2015) 10005–10014.

[12] S.M. Plis, D.R. Hjelm, R. Salakhutdinov, E.A. Allen, H.J. Bockholt, J.D. Long, H.J. Johnson, J.S. Paulsen, J.A. Turner, V.D. Calhoun, Deep learning for neuroimaging: a validation study, Front. Neurosci. 8 (2014) 229.

[13] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, H. Larochelle, Brain tumor segmentation with deep neural networks, Med. Image Anal. 35 (2017) 18–31.

[14] R.D. Hjelm, V.D. Calhoun, R. Salakhutdinov, E.A. Allen, T. Adali, S.M. Plis, Restricted Boltzmann machines for neuroimaging: an application in identifying intrinsic networks, Neuroimage 96 (2014) 245–260, https://doi.org/10.1016/j.neuroimage.2014.03.048.

[15] H.-C. Kim, H. Jang, J.-H. Lee, Test–retest reliability of spatial patterns from resting-state functional MRI using the restricted Boltzmann machine and hierarchically organized spatial patterns from the deep belief network, J. Neurosci. Methods 330 (2020), 108451, https://doi.org/10.1016/j.jneumeth.2019.108451.

[16] N. Varuna Shree, T.N.R. Kumar, Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network, Brain Informatics 5 (2018) 23–30.

[17] H. Jang, S.M. Plis, V.D. Calhoun, J.-H. Lee, Task-specific feature extraction and classification of fMRI volumes using a deep neural network initialized with a deep belief network: evaluation using sensorimotor tasks, Neuroimage 145 (2017) 314–328, https://doi.org/10.1016/j.neuroimage.2016.04.003.

[18] X. Zheng, J. Shi, Q. Zhang, S. Ying, Y. Li, Improving MRI-based diagnosis of Alzheimer's disease via an ensemble privileged information learning algorithm, in: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), IEEE, 2017, pp. 456–459.

[19] J. Kim, V.D. Calhoun, E. Shim, J.-H. Lee, Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: evidence from whole-brain resting-state functional connectivity patterns of schizophrenia, Neuroimage 124 (2016) 127–146, https://doi.org/10.1016/j.neuroimage.2015.05.018.

[20] T. Wen, R. Keyes, Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning, ArXiv Preprint ArXiv:1905.13628, 2019.

[21] R. Zhang, H. Tao, L. Wu, Y. Guan, Transfer learning with neural networks for bearing fault diagnosis in changing working conditions, IEEE Access 5 (2017) 14347–14357.

[22] T. Yeo, The organization of the human cerebellum estimated, J. Neurosci. 31 (2011) 15065–15071.

[23] L. Griffanti, G. Salimi-Khorshidi, C.F. Beckmann, E.J. Auerbach, G. Douaud, C.E. Sexton, E. Zsoldos, K.P. Ebmeier, N. Filippini, C.E. Mackay, ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging, Neuroimage 95 (2014) 232–247.

[24] G. Salimi-Khorshidi, G. Douaud, C.F. Beckmann, M.F. Glasser, L. Griffanti, S.M. Smith, Automatic denoising of functional MRI data: combining independent component analysis and hierarchical fusion of classifiers, Neuroimage 90 (2014) 449–468.

[25] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 1096–1103.

[26] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (2002) 1771–1800, https://doi.org/10.1162/089976602760128018.

[27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.

[28] C.L.P. Chen, C.-Y. Zhang, L. Chen, M. Gan, Fuzzy restricted Boltzmann machine for the enhancement of deep learning, IEEE Trans. Fuzzy Syst. 23 (2015) 2163–2173, https://doi.org/10.1109/TFUZZ.2015.2406889.

[29] C. Dewi, R.-C. Chen, H.-T. Hung Hendry, Comparative analysis of restricted Boltzmann machine models for image classification, in: N.T. Nguyen, K. Jearanaitanakij, A. Selamat, B. Trawiński, S. Chittayasothorn (Eds.), Intelligent Information and Database Systems, Springer International Publishing, Cham, 2020, pp. 285–296, https://doi.org/10.1007/978-3-030-42058-1_24.

[30] L.-W. Kim, DeepX: deep learning accelerator for restricted Boltzmann machine artificial neural networks, IEEE Transact. Neural Networks Learn. Syst. 29 (2018) 1441–1453, https://doi.org/10.1109/TNNLS.2017.2665555.

[31] M.R. Ahmed, M.S. Ahammed, S. Niu, Y. Zhang, Deep learning approached features for ASD classification using SVM, in: 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), 2020, pp. 287–290, https://doi.org/10.1109/ICAIIS49377.2020.9194791.

[32] X. Hu, H. Huang, B. Peng, J. Han, N. Liu, J. Lv, L. Guo, C. Guo, T. Liu, Latent source mining in FMRI via restricted Boltzmann machine, Hum. Brain Mapp. 39 (2018) 2368–2380, https://doi.org/10.1002/hbm.24005.

[33] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (2012) 82–97, https://doi.org/10.1109/MSP.2012.2205597.

[34] H.-C. Kim, P.A. Bandettini, J.-H. Lee, Deep neural network predicts emotional responses of the human brain from functional magnetic resonance imaging, Neuroimage 186 (2019) 607–627.

[35] P.O. Hoyer, Non-negative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (2004) 1457–1469.

[36] Y. Cui, S. Zhao, Y. Chen, J. Han, L. Guo, L. Xie, T. Liu, Modeling brain diverse and complex hemodynamic response patterns via deep recurrent autoencoder, IEEE Trans. Cogn. Dev. Syst. (2019) 1, https://doi.org/10.1109/TCDS.2019.2949195.

[37] E. Paluzo-Hidalgo, R. Gonzalez-Diaz, M.A. Gutiérrez-Naranjo, Two-hidden-layer feed-forward networks are universal approximators: a constructive approach, Neural Network. 131 (2020) 29–36, https://doi.org/10.1016/j.neunet.2020.07.021.

[38] S. Liang, R. Srikant, Why Deep Neural Networks for Function Approximation?, 2017, https://doi.org/10.48550/arXiv.1610.04161.

[39] A. Koutsoukas, K.J. Monaghan, X. Li, J. Huan, Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data, J. Cheminf. 9 (2017) 42, https://doi.org/10.1186/s13321-017-0226-y.

[40] O. Delalleau, Y. Bengio, Shallow vs. deep sum-product networks, Adv. Neural Inf. Process. Syst. (2011) 24.

[41] P. Saikia, R.D. Baruah, S.K. Singh, P.K. Chaudhuri, Artificial Neural Networks in the domain of reservoir characterization: a review from shallow to deep models, Comput. Geosci. 135 (2020), 104357, https://doi.org/10.1016/j.cageo.2019.104357.

[42] D.E. Kim, M. Gofman, Comparison of shallow and deep neural networks for network intrusion detection, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, pp. 204–208, https://doi.org/10.1109/CCWC.2018.8301755.

[43] R.L. Buckner, F.M. Krienen, A. Castellanos, J.C. Diaz, B.T. Yeo, The organization of the human cerebellum estimated by intrinsic functional connectivity, J. Neurophysiol. 106 (2011) 2322–2345.

[44] J.R. Binder, W.L. Gross, J.B. Allendorfer, L. Bonilha, J. Chapin, J.C. Edwards, T.J. Grabowski, J.T. Langfitt, D.W. Loring, M.J. Lowe, Mapping anterior temporal lobe language areas with fMRI: a multicenter normative study, Neuroimage 54 (2011) 1465–1475.

[45] F. Castelli, F. Happé, U. Frith, C. Frith, Movement and mind: a functional imaging study of perception and interpretation of complex intentional movement patterns, in: Social Neuroscience, Psychology Press, 2013, pp. 155–169.

[46] M.R. Delgado, L.E. Nystrom, C. Fissell, D.C. Noll, J.A. Fiez, Tracking the hemodynamic responses to reward and punishment in the striatum, J. Neurophysiol. 84 (2000) 3072–3077.

[47] A.R. Hariri, A. Tessitore, V.S. Mattay, F. Fera, D.R. Weinberger, The amygdala response to emotional stimuli: a comparison of faces and scenes, Neuroimage 17 (2002) 317–323.

[48] R. Smith, K. Keramatian, K. Christoff, Localizing the rostrolateral prefrontal cortex at the individual level, Neuroimage 36 (2007) 1387–1396.

[49] D.M. Barch, G.C. Burgess, M.P. Harms, S.E. Petersen, B.L. Schlaggar, M. Corbetta, M.F. Glasser, S. Curtiss, S. Dixit, C. Feldt, Function in the human connectome: task-fMRI and individual differences in behavior, Neuroimage 80 (2013) 169–189.

[50] K. Cho, Master's Thesis: Improved Learning Algorithms for Restricted Boltzmann Machines, Aalto University, 2011.

[51] A.K. Cline, C.B. Moler, G.W. Stewart, J.H. Wilkinson, An estimate for the condition number of a matrix, SIAM J. Numer. Anal. 16 (1979) 368–375, https://doi.org/10.1137/0716029.

[52] K. Jia, D. Tao, S. Gao, X. Xu, Improving training of deep neural networks via singular value bounding, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, 2017, pp. 3994–4002, https://doi.org/10.1109/CVPR.2017.425.

[53] F. Siddique, S. Sakib, MdA.B. Siddique, Recognition of handwritten digit using convolutional neural network in Python with tensorflow and comparison of performance for various hidden layers, in: 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), 2019, pp. 541–546, https://doi.org/10.1109/ICAEE48663.2019.8975496.

[54] M. Uzair, N. Jamil, Effects of hidden layers on the efficiency of neural networks, in: 2020 IEEE 23rd International Multitopic Conference (INMIC), 2020, pp. 1–6, https://doi.org/10.1109/INMIC50486.2020.9318195.

[55] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256, in: https://proceedings.mlr.press/v9/glorot10a.html (accessed January 25, 2023).

[56] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. http://arxiv.org/abs/1502.01852, 2015 (accessed January 25, 2023).