Postprint: Alzheimer' s Disease Classification Algorithm Combining DCGAN and LSTM

Authors: Lin Ying, Xiaofeng He, CHEN Lingna, Chen Junxi

Date: 2019-04-01T00:00:00+00:00

Abstract

To address the issues of excessive parameters in traditional 3D models and the lack of continuity features in 2D models for Alzheimer' s disease (AD) classification, we propose a brain magnetic resonance imaging (MRI) image classification algorithm that combines 2D convolutional neural networks with Long Short-Term Memory (LSTM) networks. By leveraging Deep Convolutional Generative Adversarial Networks (DCGAN), the convolutional layers can automatically extract image features from unlabeled data in an unsupervised manner. The approach first trains the convolutional neural network unsupervised; converts MRI image sequences into feature sequences, which are then fed into the Long Short-Term Memory network for training; and finally performs classification by combining the feature sequences with the LSTM' s hidden states. Experimental results show that, compared to 3D models, the proposed algorithm has significantly fewer parameters, achieving 93.93% accuracy for NC versus AD classification, and 86.27% accuracy for NC versus MCI classification.

Full Text

Preamble

Vol. 37 No. 5
Application Research of Computers
ChinaXiv Partner Journal

Alzheimer' s Disease Classification Algorithm Based on DCGAN and LSTM

Lin Ying[1] , He Xiaofeng[1] ,[2], Chen Lingna[1] †, Chen Junxi[1]
(1. a. School of Computer Science; b. Affiliated Nanhua Hospital, University of South China, Hengyang, Hunan 421001, China;
2. Xiangya School of Public Health, Central South University, Changsha 410008, China)

Abstract: Conventional Alzheimer' s disease (AD) classification methods suffer from excessive parameters in 3D models and lack of continuous features in 2D models. To address these issues, this paper proposes a brain MRI (Magnetic Resonance Imaging) image classification algorithm that combines 2D convolutional neural networks with long short-term memory networks. By leveraging deep convolutional generative adversarial networks (DCGAN), convolutional layers can automatically extract image features in an unsupervised manner. The method first trains the convolutional neural network unsupervised, transforms the MRI image sequence into a feature sequence, and then feeds it into the long short-term memory network for training. Finally, classification is performed by combining the feature sequence with the hidden states of the LSTM. Experimental results demonstrate that, compared with 3D models, this algorithm has fewer parameters and achieves 93.93% accuracy for NC vs. AD classification and 86.27% for NC vs. MCI.

Keywords: Alzheimer' s disease; deep convolutional generative adversarial networks; long short-term memory; unsupervised

0 Introduction

Alzheimer's disease is a degenerative brain disorder and the most common cause of dementia [1]. Currently, AD diagnosis primarily relies on psychological tests, neuroimaging, blood biomarkers, and other indicators [2]. According to World Health Organization estimates, the number of Alzheimer' s patients worldwide will quadruple within decades, reaching 114 million by 2050 [3]. Early and accurate diagnosis enables patients to receive supportive treatment that helps them maintain independence longer, potentially reducing associated costs [4].

Structural MRI is a crucial, widely available, and non-invasive biomarker that effectively reveals AD progression in patients [5]. Structural MRI can be used to train computer-aided diagnostic algorithms [6], which have the potential to identify group differences that human experts might miss during qualitative analysis of brain images, yielding more objective diagnostic results than clinical standards [7].

Feature extraction quality profoundly impacts image classification outcomes. Since Krizhevsky' s victory in the 2012 ImageNet competition, deep convolutional neural networks have regained research prominence. Convolutional neural networks have been applied in computer vision for decades. Unlike traditional "hand-crafted" feature methods such as HOG and SIFT, CNNs are data-driven models capable of automatic feature extraction. CNN applications in medical image analysis date back to 1990, when they were used to assist in detecting microcalcifications and lung nodules. Several relevant studies have emerged in Alzheimer's disease research. Zhu et al. [8] proposed a graph-based feature selection method that filters irrelevant features before SVM classification, achieving better results than non-feature-selection approaches. Tong et al. [9] utilized in-

dependent component analysis for feature extraction followed by support vector machine classification.

Hosseini-Asl et al. [10] employed convolutional auto-encoders (CAE) to extract features from 3D MRI images, then used CAE weights to initialize a 3D CNN network, achieving 89.1% classification accuracy on three image categories: AD, MCI (mild cognitive impairment), and NC (normal control). Cheng et al. [11] designed a framework combining 2D CNN and BGRU (bidirectional gated recurrent unit). This method converted 3D PET images into 2D image sequences, extracted image features using convolutional networks, and then used BGRU to extract inter-image features for final classification. While this approach could consider spatial information using 2D convolutions, it suffered from substantial memory requirements and computational costs during training because it needed to compute gradients for all convolutional layer weights at each time step, requiring the convolutional layer to retain all intermediate values during forward computation. Li et al. [12] proposed a deep learning-based AD classification framework that used principal component analysis to extract features from MRI and PET images, applied stability selection and restricted Boltzmann machines (RBM) for feature selection and extraction, and finally used SVM for classification. Liu et al. [13] fused MR and PET images, used stacked autoencoders to extract high-level image features, and fine-tuned the network for multi-classification tasks. Suk et al. [14] first downsampled samples to fewer voxels, then used deep Boltzmann machines (DBM) to learn latent features in "blocks," and applied ensemble learning with multiple classifiers.

Currently, due to the superiority of convolutional networks in image feature extraction, many researchers focus on designing improved prediction models to better extract MRI image features and enhance classification accuracy. These methods have achieved promising results using neural networks, but their performance heavily depends on hyperparameter selection and optimization.

Training a convolutional neural network from scratch is challenging because it requires large amounts of labeled training data and numerous techniques to ensure convergence—requirements that may not be met in the medical domain [15]. Using unsupervised algorithms such as autoencoders for image feature extraction is becoming a trend. Our model separates feature extraction and classification into two training stages. To leverage MRI image spatial information while reducing network complexity, we utilize deep convolutional generative adversarial networks to extract image features in an unsupervised manner. When unrolling the long short-term memory network, since the convolutional layers are already trained, we fix their weights and perform only forward computation without calculating gradients, thereby avoiding the need to retain convolutional layer intermediate values during LSTM training and significantly reducing computational resource requirements.

Our model offers advantages for long sequence computation. Compared with earlier SVM-based methods, deep learning-based classification approaches achieve superior results. Among these, 3D convolutional networks demonstrate remark-

able effectiveness but involve numerous parameters and computational costs.

1.1 Deep Convolutional Generative Adversarial Networks

Generative adversarial networks, proposed by Goodfellow et al. [16] in 2014, bypass the difficulty of solving likelihood functions by directly generating samples to fit the training data distribution. GANs consist of a generator and a discriminator: the generator maps random noise to the sample space, while the discriminator outputs a probability between 0 and 1 to determine whether an input sample is real or generated. For real samples, the discriminator outputs probabilities close to 1; for generated samples, probabilities near 0. The two components are trained alternately—fixing the generator's parameters when training the discriminator, and fixing the discriminator' s parameters when training the generator—until the distribution of generated data coincides with that of real samples. The network structure is shown in Figure 1 [Figure 1: see original paper].

The objective function of generative adversarial networks is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

where D is the discriminator, x represents real samples, G is the generator, and z denotes input random noise. GANs learn through adversarial training, ultimately reaching Nash equilibrium. Unlike typical supervised training, the generator G' s parameter update gradients do not come from real samples but from the discriminator D.

Since the introduction of GANs, numerous variants have emerged. Radford et al. [17] proposed the DCGAN model, which combines convolutional neural networks with generative adversarial networks, demonstrating that the discriminator' s convolutional kernels can learn dataset features from unlabeled image data. DCGAN employs several constraints, such as replacing pooling layers with convolutional layers and adding batch normalization (BN) to each convolutional layer, making CNN training with GANs more stable. DCGAN trains in an unsupervised manner, and its trained discriminator can be used for image feature extraction.

1.2 Long Short-Term Memory Networks

Traditional recurrent neural networks (RNNs) suffer from gradient vanishing when processing long-term dependencies. In 1997, Hochreiter and Schmidhuber [18] proposed long short-term memory networks, introducing memory units. In

2000, Gers added forget gates that determine which information should be remembered based on input and the previous time step' s state. Our model uses LSTM with forget gates, computed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

where σ denotes element-wise multiplication; W and U are weight matrices for each gate; σ is the activation function; the input gate i_t and output gate o_t control information flow into and out of the memory cell; C_t represents the cell state; and h_t is the hidden layer output state. The improved LSTM network resolves gradient propagation issues when processing long sequences, enabling effective network training.

---

2 Our Model

Convolutional neural networks are widely used in various image classification algorithms due to their excellent feature extraction performance. Long short-term memory networks are suitable for sequence-related problems. Our proposed model combines these advantages by converting consecutive 2D images into feature sequences through convolutional neural networks, then feeding them into long short-term memory networks for classification. Our model is illustrated in Figure 2 [Figure 2: see original paper].

DCGAN eliminates pooling layers, and its discriminator consists of convolutional layers with a final fully connected layer that outputs the probability of an image being a real sample. While the original DCGAN model accepts 64×64 inputs, the MRI images in our dataset are larger than this size. To avoid information loss from image compression, our model expands the input to 160×160 pixel grayscale images, comprising a DCGAN discriminator and an LSTM network with two stacked hidden layers. The discriminator uses 5×5 convolutional kernels with a stride of 2. The final convolutional layer produces 10×10 feature maps, which are pooled before being fed into the LSTM network. The model also reduces the number of feature maps accordingly, with the four convolutional layers having 64, 128, 256, and 512 feature maps respectively. The LSTM hidden layer size is 256. After processing all images, the extracted features are input into the LSTM layer to capture inter-image information. The 2D features are then concatenated with the LSTM' s hidden states and fed into

a fully connected network for classification, with a softmax classifier outputting classification probabilities. Sample class labels are encoded using one-hot encoding.

The model training process consists of two stages. First, the DCGAN network is trained. Since GANs only distinguish between real and fake samples during training, image categories need not be differentiated, and all images can be directly input to the discriminator. To accelerate network convergence, we apply z-score normalization to the input data using the transformation:

$$x^* = \frac{x - \mu}{\sigma}$$

where x is the pre-normalization pixel value, μ is the pixel mean, and σ is the pixel standard deviation.

During DCGAN training, the generator receives normally distributed noise as input and feeds generated samples to the discriminator. Real and generated samples are input to the discriminator separately, with the loss values from both parts summed as the discriminator' s total loss. To avoid neuron "death" (where neurons no longer activate for any input), we use LeakyReLU as the activation function, defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

where α is a small constant.

To make generated images sharper, our model adds a penalty term to the DCGAN loss function:

$$\Omega(G) = \mathbb{E}_{z \sim p_z(z)}[||\nabla_x D(G(z))||]$$

The overall loss function becomes:

$$\min_G \max_D V(D, G) = \mathcal{L}_{DCGAN}(D, G) + \lambda \Omega(G)$$

For the LSTM network, we use cross-entropy as the loss function, defined as:

$$H(p, q) = -\sum_x p(x) \log q(x)$$

where p is the true distribution and q is the model' s output probability. The L1 regularization term yields sparser solutions beneficial for feature selection, while L2 regularization reduces model complexity and overfitting risk. Considering

these factors, our model' s LSTM layer employs both L1 and L2 regularization. Since the model performs binary classification, the loss function becomes:

$$\mathcal{L}_{LSTM}(\omega_1, \omega_2) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \ln \hat{y}_i + (1-y_i)\ln(1-\hat{y}_i)] - \lambda_1||\omega_1||_1 - \lambda_2||\omega_2||_2$$

where y is the label; $\hat{y}$ is the model' s output probability for each class; λ_1 is the L1 regularization weight; λ_2 is the L2 regularization weight; ω_1 is the weight matrix for the fully connected layer; and ω_2 represents the weights for the LSTM and output layers. Both DCGAN and LSTM are optimized using the Adam algorithm to minimize the objective function. Regularization weights exclude bias parameters.

When training the LSTM network, the final feature map undergoes max pooling to reduce parameters before being input to the LSTM layer. At this stage, convolutional layer parameters are fixed, and the optimization algorithm does not compute gradients for these weights. Finally, the LSTM output vector $\{h_0, h_1, ..., h_n\}$ is concatenated with the convolutional layer output $\{x_0, x_1, ..., x_n\}$ as input to the fully connected layer, whose output is mapped to classification probabilities by the softmax function:

$$z_i = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

where s_i is the i-th output of the fully connected layer and s_j represents all outputs of the fully connected layer.

---

3 Experiments and Analysis

Our experiments were conducted on a machine with an NVIDIA GeForce 1080Ti GPU and 16GB RAM. The experimental code was written using the TensorFlow deep learning framework and run on the Ubuntu 16.04 platform. All data were obtained from the ADNI (Alzheimer' s Disease Neuroimaging Initiative) database. ADNI is a research project dedicated to improving Alzheimer's disease prevention and treatment, providing multimodal brain imaging data including MRI and PET (positron emission tomography) on its website. The selected samples in our experiments ranged in age from 55 to 91 (inclusive) and were 3D NifTI (neuroimaging informatics technology initiative) format files from 1.5T magnetic resonance imaging.

3.1 Data Preprocessing

Our experiments utilized brain MRI data from 664 patients, including NC, MCI, and AD samples, with each sample containing 256 images (coronal plane).

Specifically, there were 185 normal control samples, 322 mild cognitive impairment samples, and 157 Alzheimer' s disease samples. The 3D NifTI format images contain three imaging planes: axial, sagittal, and coronal. The dataset was divided into training, validation, and test sets using ten-fold cross-validation. The first and last 40 images were removed as they contained little useful information, resulting in 116,844 total images used. When training DCGAN, these images were input to the discriminator without category distinction. To increase sample quantity and improve model generalization, we applied random cropping, flipping, and rotation to the original image data. The same data augmentation was applied when training the LSTM network, while validation and test sets remained unaugmented.

3.2 Parameter Configuration

Proper weight initialization enables more effective neural network training. Our model initializes convolutional and fully connected layers with a normal distribution $\mathcal{N}(0, 0.02)$, LSTM layer weights with orthogonal matrices, and all bias terms with 0.

The DCGAN learning rate is set to 2e-4, with β_1 at 0.5, consistent with the experimental settings in reference [17]. The generator input is noise from $\mathcal{N}(0, 1)$ distribution, and the L1 regularization weight is 2E-3. The discriminator weights use no regularization, with batch size set to 128.

The Adam optimizer is used with a learning rate of 1e-4. Both L1 and L2 regularization weights are set to 2E-3, with batch size at 32. Other parameters use TensorFlow' s default settings. DCGAN training iterations are set to 50k, and LSTM iterations to 10k.

3.3.1 Generated Samples

After DCGAN training completes, the discriminator is used for image feature extraction in our model, while the generator learns to produce samples approximating the real data distribution. Figure 3 [Figure 3: see original paper] shows generated brain MRI image samples, which appear highly similar to real samples. The generated images exhibit some degree of tilt because the input images underwent rotation operations, which the generator learned. Similarly, the discriminator learned translation and mirror invariance of image features. Experimental results demonstrate that Equation (10) accelerates convolutional layer convergence and produces clearer generated samples.

3.3.2 Feature Map Visualization

Visualizing activation layers helps understand what features convolutional neural networks learn. In reference [19], the authors visualized first-layer convolutional kernels, showing that this layer learned low-level features such as edges and colors. This direct visualization approach typically only works for shallow

convolutional kernels. For deep convolutional kernels, the learned features are more abstract and difficult to interpret even when visualized.

To better understand convolutional neural networks, Zeiler et al. [20] visualized feature maps through deconvolution. This method maps feature maps back to the input image pixel space through unpooling, unactivation, and deconvolution processes, producing an image the same size as the input. Figure 4 [Figure 4: see original paper] shows the deconvolution results of the first 16 feature maps from the second convolutional layer. The figure demonstrates that convolutional kernels extract contour information while ignoring background, confirming that the discriminator' s convolutional layers learn image features.

3.3.3 Classification Results

Convolutional neural networks have been extensively studied for medical image classification. Recent methods mostly use 2D CNN for feature extraction followed by fully connected networks for classification. A single MRI image alone cannot easily determine its category, as some images may not contain lesions and cannot provide sufficient information. While averaging classification results across all 2D images can partially solve this problem, it loses information about relationships between images that could improve classification accuracy. Compared with 2D models, 3D convolution can capture local spatial features of MRI images, but its drawbacks are evident: 3D convolution has more parameters, potentially increasing overfitting risk, especially with limited samples. Additionally, 3D model training requires more computational resources.

Table 1 compares our model' s experimental results with several existing methods. Compared with existing Alzheimer' s disease 3D classification models, our proposed model significantly reduces parameters and decreases overfitting likelihood. Meanwhile, compared with 2D models, our model can utilize spatial feature information due to the added LSTM layer. Compared with the method in reference [13], our model does not require repeatedly computing convolutional layer gradients when training the classifier, reducing computational complexity and enabling longer sequence computation.

Figure 5 [Figure 5: see original paper] shows our model' s ROC (Receiver Operating Characteristic) curves. For binary classification models, ROC curves are commonly used to evaluate classification performance. The vertical and horizontal axes represent True Positive Rate (TPR) and False Positive Rate (FPR), defined as:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

where TP represents true positive samples, FP false positive samples, FN false

negative samples, and TN true negative samples. In our experiments, NC group samples were selected as the negative class, while AD and MCI group samples were positive classes.

Besides classification accuracy, AUC (area under curve) is another evaluation metric ranging from 0 to 1. Figure 5 shows our model' s AUC values, reaching 95% for AD/NC and 90% for MCI/NC.

---

4 Conclusion

This paper proposes an Alzheimer' s disease classification model combining deep convolutional generative adversarial networks with long short-term memory networks. Using generative adversarial networks, convolutional networks can extract image features in an unsupervised manner, which is highly useful for medical images lacking labels. Insufficient training samples also contributes to model overfitting. Future work will incorporate MRI images from multiple angles and use conditional generative adversarial networks to generate samples of various categories, increasing data volume and improving model performance.

References

[1] Alzheimer' s Association. 2017 Alzheimer' s disease facts and figures [J]. Alzheimer' s & Dementia, 2017, 13 (4): 325-373.

[2] Ye Yuru. Alzheimer's disease: the serious challenge of modern mental science and medical science [J]. Chinese Bulletin of Life Sciences, 2014, 26 (1): 1.

[3] Alzheimer' s Association. 2015 Alzheimer' s disease facts and figures [J]. Alzheimer' s & Dementia, 2015, 11 (3): 332.

[4] Alzheimer' s Association. 2014 Alzheimer' s disease facts and figures [J]. Alzheimer' s & Dementia, 2014, 10 (2): e47-e92.

[5] Paquerault S. Battle against Alzheimer' s disease: the scope and potential value of magnetic resonance imaging biomarkers [J]. Academic Radiology, 2012, 19 (5): 509-511.

[6] Jr Jack C R, Knopman D S, Jagust W J, et al. Tracking pathophysiological processes in Alzheimer' s disease: an updated hypothetical model of dynamic biomarkers [J]. The Lancet Neurology, 2013, 12(2): 207-216.

[7] Klöppel S, Abdulkadir A, Jr Jack C R, et al. Diagnostic neuroimaging across diseases [J]. Neuroimage, 2012, 61 (2): 457-463.

[8] Zhu Yonghua, Cheng Debo, He Wei, et al. Graph feature selection for Alzheimer' s disease diagnosis [J]. Application Research of Computers, 2017, 34(4): 1018-1021.

[9] Tong Tong, Wolz R, Gao Qqinquan, et al. Multiple instance learning for classification of dementia in brain MRI [J]. Medical Image Analysis, 2014, 18(5): 808-818.

[10] Hosseini-Asl E, Keynton R, El-Baz A. Alzheimer' s disease diagnostics by adaptation of 3D convolutional network [C]//Proc of the 23rd IEEE International Conference on Image Processing. Piscataway, NJ: IEEE Press, 2016: 126-130.

[11] Cheng D, Liu Manhua. Combining convolutional and recurrent neural networks for Alzheimer' s disease diagnosis using PET images [C]//Proc of IEEE International Conference on Imaging Systems and Techniques. Piscataway, NJ: IEEE Press, 2017: 1-5.

[12] Li Feng, Tran L, Thung Kim Han, et al. A robust deep model for improved classification of AD/MCI patients [J]. IEEE Journal of Biomedical and Health Informatics, 2015, 19(5): 1610-1616.

[13] Liu Siqi, Liu Sidong, Cai Weidong, et al. Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer' s disease [J]. IEEE Trans on Biomedical Engineering, 2015, 62(4): 1132-1140.

[14] Suk H I, Lee S W, Shen Dinggang, et al. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis [J]. NeuroImage, 2014, 101: 569-582.

[15] Tajbakhsh N, Shin J Y, Gurudu S R, et al. Convolutional neural networks for medical image analysis: Full training or fine tuning? [J]. IEEE Trans on Medical Imaging, 2016, 35 (5): 1299-1312.

[16] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C]//Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014: 2672-2680.

[17] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks [EB/OL]. (2016-01-07) [2018-10-25]. https://arxiv.org/abs/1511.06434.

[18] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735–1780.

[19] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks [C]//Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2012: 1097-1105.

[20] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks [C]//Proc of European Conference on Computer Vision. New York: Springer, 2014: 818-833.

Note: Figure translations are in progress. See original paper for figures.