Alzheimer's Disease Classification Based on Deep BLSTM-LSTM Network Model

Authors: Li Hongbei

Date: 2024-01-10T00:00:00+00:00

Abstract

As science and technology advance, EEG signals are being applied in increasingly more domains due to their rich physiological, psychological, and pathological information content. Relevant experiments conducted on EEG signal datasets demonstrate that the LSTM-based model implemented in this project has achieved promising results for Alzheimer's disease classification, with an average accuracy of approximately 93%.

Full Text

Alzheimer's Disease Classification Based on a Deep BLSTM-LSTM Network Model

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 10010

With the advancement of science and technology, electroencephalogram (EEG) signals have been increasingly utilized across various domains due to their rich physiological, psychological, and pathological information. Experimental results on EEG datasets demonstrate that our proposed approach for Alzheimer's disease classification using BLSTM and LSTM models achieves promising performance, with an average accuracy of approximately 93%.

Keywords: EEG signals, Alzheimer's disease, Long Short-Term Memory network

1 Research Background

Alzheimer's disease is a neurodegenerative disorder commonly affecting the elderly population and represents the fourth leading cause of death worldwide. Characterized by progressive memory loss and cognitive decline, patients gradually transition from normal aging to mild cognitive impairment. As the disease progresses, neurons and their connections are progressively destroyed, ultimately leading to complete cognitive failure and dementia. According to a 2019 report, approximately 6% of China's middle-aged and elderly population suffers from dementia, with over 10 million individuals enduring the suffering caused by this disease [1].

The diagnosis of Alzheimer's disease is complicated by multiple factors. The condition presents with diverse and complex symptoms requiring extensive examinations, yet few hospitals are equipped with complete early-stage diagnostic capabilities. Additionally, the high cost of pathological examinations and other factors create significant challenges for diagnostic techniques and methods. With China's elderly population proportion increasing daily and the persistent issue of "difficulty in accessing medical care" remaining unresolved, the country's current healthcare resource allocation is imbalanced, with remote and impoverished regions having far fewer medical resources compared to first-tier cities. Consequently, over half of Alzheimer's patients in China are already at moderate to severe stages at the time of diagnosis. Since earlier treatment substantially improves the likelihood of recovery, developing methods for early diagnosis of Alzheimer's disease holds significant research importance [2].

Electroencephalogram (EEG) signals record the spontaneous, rhythmic electrical activity of brain cell populations through electrodes placed on the scalp or via electrode caps. Widely employed in clinical research for detecting and identifying various brain lesions and diseases, EEG serves as a primary tool for neurologists and clinical specialists in diagnosing epilepsy, sleep disorders, schizophrenia, seizure prediction, depression, and other conditions [3].

In light of this background, this paper proposes an approach for Alzheimer's disease diagnosis using EEG signal data combined with deep learning models.

2 Related Technology

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network specifically designed to address the long-term dependency problems inherent in standard RNNs. All RNNs possess a chain-like structure of repeating neural network modules. In standard RNNs, this repeating structural module has a very simple architecture, typically consisting of a single tanh layer.

Due to the phenomena of gradient explosion and vanishing, simple recurrent neural network models struggle to process long-distance sequential data. To address the inability of errors to propagate backward over long distances, Sepp Hochreiter and Jürgen Schmidhuber proposed an improved recurrent neural network called Long Short-Term Memory in 1997 [4]. The most significant characteristic of this neural network is its ability to avoid the gradient vanishing problem plaguing simple RNNs, leading to its widespread adoption in industrial applications.

In unidirectional recurrent neural networks, the model utilizes only "previous context” information without considering “future context.” However, real-world predictions often require information from the entire input sequence. Consequently, bidirectional recurrent neural networks have become the mainstream approach in the field. As the name suggests, bidirectional RNNs combine one RNN that moves from the sequence start to end with another that moves from the sequence end to start. As an extension of RNNs, LSTM can naturally be combined with a reverse sequence to form a Bidirectional Long Short-Term Memory (Bi-LSTM) network, composed of forward and backward LSTMs. This approach separates the component responsible for the positive time direction (forward state) from the component responsible for the negative time direction (backward state), which has a significant impact on the recurrent neural network architecture of LSTM.

[Figure 1: see original paper] illustrates the structure of a bidirectional LSTM network. The LSTM network model structure is described as follows:

[Figure 2: see original paper] shows the network model structure of LSTM. The core mechanism involves controlling the input and output of unit state information ct. LSTM employs three control switches, or “gates,” to determine changes to the unit state. The hidden layer at time t can be simplified into the schematic diagram shown below, which reveals three switches, or “gates,” controlling the intermediate unit state ct. The left gate is the forget gate, which controls how much of the previous unit state ct-1 continues to be retained at time t. The bottom gate is the input gate, controlling the information from the current input to the long-term state. The right gate is the output gate, controlling the output information from the unit state at the current time t.

[Figure 3: see original paper] shows the simplified schematic diagram of the LSTM model. These “gates” are actually implemented as fully connected layers in the algorithm. As shown in the diagram below, x represents the input vector, W denotes the gate weights, and σ is the activation function.

[Figure 4: see original paper] illustrates the gate structure schematic diagram. The gate can be expressed as: $\mathrm{g(x)} = \sigma(\mathrm{Wx} + \mathrm{b})$. The internal operations of an LSTM unit are divided into three stages: forgetting, memory, and output. The forgetting stage uses the forget gate to weight the internal state, enabling the forgetting of long-term memory. The memory stage extracts features from the input while using the input gate to preserve important features in the internal state, completing the update of long-term memory. The output stage uses the output gate to select output information from the internal state.

3 LSTM-Based Model Architecture

[Figure 5: see original paper] presents the Deep BLSTM-LSTM model architecture. As LSTM networks possess the ability to discover potential temporal dependencies in sequential data, this paper adopts the Deep BLSTM-LSTM architecture proposed in [5], which consists of approximately nine layers including one BLSTM layer, two LSTM layers, as well as Dropout, batch normalization, and fully connected layers.

The input signal first enters a Bi-LSTM layer composed of 256 neurons, followed by a Dropout layer. Next, batch normalization is performed, and the results are fed into an LSTM layer containing 128 hidden neurons. After another batch normalization layer, the signal passes through a second LSTM layer with 64 hidden neurons and another normalization layer. Finally, the signal proceeds through two fully connected layers.

This model processes raw EEG signals in an end-to-end manner through the following stages: (1) Morphological feature extraction, which learns discriminative features between normal and Alzheimer's patient EEG signals. This paper employs a bidirectional LSTM neural network to extract and mine morphological characteristics from EEG signals. Additionally, a Dropout layer is added after the Bi-LSTM layer to prevent overfitting. (2) Feature enhancement, which utilizes LSTM to learn temporal dependencies between signal samples to strengthen features. After passing through the LSTM layers, the signals are transformed into combinations containing both signal characteristics and temporal dependencies, which are then further deepened and dimensionally adjusted in the Dense layers.

4.1 Data Preprocessing

In the utilized dataset, each file has a format of 1024 $\times$ 19, and the overall data quantity is relatively small. To expand the dataset and facilitate better data processing by the model, the data is segmented during the preprocessing stage into smaller pieces of 32 $\times$ 19 and stored as new txt files in a sequential numbering order. Additionally, this stage designs labels for different file types, marking healthy subjects as '0' and patients as '1', with these labels saved to txt files.

4.2 Data Encapsulation

The preprocessed data is encapsulated as a class inheriting from Dataset, with its functions overridden to facilitate subsequent data reading and loading operations. A load_{mydata} function is implemented to enable convenient dataset loading.

4.3 Data Splitting

Regarding data splitting and loading: All data is read in and shuffled during loading. The data is divided into training and test sets according to a certain ratio. The txt file storing the correspondence between filenames and labels is similarly split into two txt files for training and test sets according to the same ratio.

4.4 Training and Testing

Training follows a general four-step strategy: (1) Calculate the loss between the forward output results and actual results under current network weights; (2) Zero the gradients to prevent accumulation; (3) Perform backpropagation; (4) Execute gradient descent for one step of parameter update. The model is trained using an alternating approach of one training epoch followed by one testing epoch.

5.1 Dataset

The dataset selected is Quantitative EEG for Alzheimer's disease diagnosis, designed by Dr. Dennis Duke and other researchers from Florida State University. The data was recorded from 19 scalp sites based on the International 10-20 system using a Biologic Systems Brain Atlas III Plus workstation. Groups A and B represent the control group, consisting of 24 healthy elderly individuals who tested negative for all neurological or psychiatric diseases. Groups C and D comprise 24 probable Alzheimer's patients diagnosed according to the criteria of the National Institute of Neurological and Communicative Disorders and Stroke and the Alzheimer's Disease and Related Disorders Association (NINCDS-ADRDA) and the Diagnostic and Statistical Manual of Mental Disorders (DSM)-III-R.

5.2 Evaluation Metrics

For a given test dataset, accuracy is calculated as the number of correctly predicted samples divided by the total number of observed samples. In the code implementation, perfectly matched prediction samples are counted and divided by the total number of samples.

5.3 Hyperparameter Optimization

Hyperparameter settings are essential for better learning of data structures and ensuring the model does not overfit. Hyperparameters are critical controls for model structure, efficiency, and performance, allowing adjustment of learning rate, iteration count, number of hidden layers, activation functions, optimizer, etc.

1. Learning Rate Optimization: We first compare different initial learning rates by conducting experiments with learning rates of 0.1, 0.01, and 0.001. As shown in the figure, when the learning rate is 0.01, the curve changes more smoothly and yields better results.

[Figure 6: see original paper] Comparison of accuracy under different learning rates

2. Optimizer Selection: - Stochastic Gradient Descent (SGD) selects a mini-batch rather than all samples for each iteration to update model parameters using gradient descent. While it addresses the issue of small batch samples, it still suffers from problems such as non-adaptive learning rates and susceptibility to getting stuck in regions with small gradients. - **Adaptive Gradient (AdaGrad)** is an extension of gradient descent optimization algorithms. AdaGrad is a gradient descent optimization method with adaptive learning rates, making parameter learning rates adaptive and particularly suitable for processing sparse data, thereby significantly improving SGD's robustness. - **Adam Optimizer** combines the advantages of both AdaGrad and RMSProp optimization algorithms. Parameter updates are unaffected by gradient scaling transformations, hyperparameters are highly interpretable, and typically require no adjustment or only minimal fine-tuning.

We conducted experiments with these three optimizers on the dataset for the same number of iteration cycles, with results shown in . According to the data in the table, the Adam optimizer achieves faster loss reduction and higher accuracy within the same iteration cycle compared to SGD and AdaGrad optimizers.

Comparison under different optimizers

5.4 Experimental Results

Following the hyperparameter optimization described in Section 5.3, the model was adjusted with a learning rate of 0.01 for training and testing. [Figure 7: see original paper] illustrates the relationship between loss values and epochs for both training and test sets, showing a decreasing trend as training progresses.

[Figure 7: see original paper] Loss values during training and testing phases across epochs

[Figure 8: see original paper] displays the loss values and accuracy changes over 20 training epochs. The plotted results show accuracy approaching 1 and loss values gradually converging.

[Figure 8: see original paper] Accuracy and loss values over 20 training epochs

With the advancement of science and technology, EEG signals have been increasingly utilized across various domains due to their rich physiological, psychological, and pathological information. Experimental results on EEG datasets demonstrate that our implementation of a deep BLSTM-LSTM model for Alzheimer's disease classification achieves promising results, with an average accuracy of approximately 93%, suggesting that EEG signals may serve as a valuable aid in Alzheimer's disease diagnosis. The project data and code are available at https://gitcode.net/lhb_{taylor}/bilstm-lstm.

As observed from the plotted curves, slight overfitting occurs in the test set. Attempts to mitigate this by adding Dropout layers did not yield significant improvement, and we hope to explore solutions in future work. Compared with traditional machine learning methods, deep learning exhibits a strong dependence on massive training data, as it requires large volumes of data to understand underlying patterns. The dataset used in this project is relatively small, and although sample augmentation was performed through segmentation, the data volume remains limited. We consider employing few-shot learning and transfer learning to optimize the model. Further research is needed on improving training efficiency and refining the network architecture.

[1] Sun Xiaochi. Application of machine learning methods in Alzheimer's disease diagnosis[D]. Zhongnan University of Economics and Law, 2023. DOI:10.27660/d.cnki.gzczu.2021.002053.
[2] Zhang Fan. Research on Alzheimer's disease classification based on multi-feature fusion of hypernetwork functional magnetic resonance imaging[D]. Taiyuan University of Technology, 2018.
[3] Sanei S, Chambers J A. EEG signal processing[M]. John Wiley & Sons, 2013:22-29.
[4] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780. DOI:10.1162/neco.1997.9.8.1735.
[5] P. Kaushik, A. Gupta, P. P. Roy and D. P. Dogra. EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model[J]. IEEE Sensors Journal, 2019, (17)7: 2634-2641. DOI: 10.1109/JSEN.2018.2885582.

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