Chapter 16
Deep Learning

Zhongke Gao and Xinmin Wang

Abstract Brain-computer interface (BCI) technologies enable direct communications between humans and computers by analyzing EEG signals. One of the challenges with traditional methods in classification tasks is receiving unsatisfactory recognition effects from EEG signals. In recent years, deep learning has drawn a great deal of attentions in diverse research fields, and could provide a novel solution for learning robust representations from EEG signals. In this chapter, we firstly introduce the basic concepts of deep learning techniques and two commonly used structures in time series analysis, namely, convolutional neural network and recurrent neural network. Then, we provide the applications of these two DL models to focus on the eye state detection task, which both achieve excellent recognition effects and are expected to be useful for broader applications in BCI systems.

Keywords Fatigue detection · EEG analysis · Brain-computer interface · Deep learning

Machine learning techniques allow to extract effective information from EEG signals, which play a vital role in different EEG-based classification research tasks. And machine learning methods have been applied to many control applications. For example, ErrP signals are decoded from a human operator in real time to control robots to perform a binary object selection task (Salazar-Gomez et al. 2017). Such systems may be further upgraded for more examples. The core point of EEG-based recognition systems is to develop practical computational algorithms, which are increasingly recognized as novel tools for rehabilitation therapy. In spite of much progress has been conducted, there is still considerable improvement room for the accuracy of information extraction from EEG signals. Hence, a direction from the area of machine learning attracts great interests of researchers, which is deep learning.

Z. Gao (✉) · X. Wang
School of Electrical and Information Engineering, Tianjin University, Tianjin, China
e-mail: zhongkegao@tju.edu.cn

L. Hu, Z. Zhang (eds.), *EEG Signal Processing and Feature Extraction*,
https://doi.org/10.1007/978-981-13-9113-2_16

16.1 What Is Deep Learning?

Machine learning is a subbranch of AI that focuses on teaching computers how to learn without the need to be programmed for specific tasks. The core idea of ML methods is to develop algorithms that learn from and make prediction on data. Deep learning is a particular subset of ML methods using artificial neural networks (ANNs), which are slightly inspired by the structure of neurons located on the human brain. Informally, the word *deep* refers to the presence of many layers in the artificial neural network, but this definition of deep has changed over time (Gulli and Pal 2017).

Recently, deep learning techniques have shown prominent abilities in so many fields, such as image classification, speech recognition, and time series prediction. It significantly improves previous state-of-the-art results achieved over dozens of years, which is due to the availability of more training data (such as ImageNet for images) and the relatively low-cost availability of GPUs for very efficient numerical computation. Based on these existing conditions, the research on network design is a very important part of deep learning techniques. Moreover, various studies have started to investigate the potential of CNNs and their variations for EEG signals decoding, including motor imagery classification, fatigue driving evaluation, and emotion recognition. Note that EEG signals are different with 2D images and speech signals, which is truly challenging to design a proper network for EEG-based classification tasks.

A deep learning framework is a function that takes the values of various features of EEG signals (or raw signals) as the network input, and predicts the class of the samples. The experiments are carried out to collect the EEG signals of subjects with cognitive or perceptual responses. Let vector $\mathrm{X} \in \mathrm{R}^{E \times T}$ and its class label as $y \in R^N$, where E denotes the cap electrode plates, T denotes the signal sampling points, and N denotes the classes of response states. A DL framework can build up the relationships between input samples and labels using the training data, and then can predict the label of a given sample.

16.2 Typical Deep Learning Methods for Time Series

Learning effective representations from EEG signals is a challenging problem to be investigated. Various novel methods have been proposed to build up a proper framework for classification tasks. These methods have shown great potentials for learning effective features from EEG signals. The commonly used methods include convolutional neural networks and recurrent neural networks.

16.2.1 Convolutional Neural Networks

In the following sections, we first explain the basic idea of CNNs, and then introduce architecture choices for EEG analysis. Finally, we describe the implementation details of CNN training.

Generally, CNNs have shown great superiority in many learning tasks, such as images and audio signals. These signals often have an inherent hierarchical structure. For example, images typically consist of edges that together form simple shapes which form larger and more complex shapes. CNNs can learn local non-linear features through convolutions and nonlinearities and represent higher-level features as compositions of low-level features through multiple layers of processing. In addition, many CNNs employ pooling layers to create a coarser inter-mediate feature representation to increase translation-invariant.

To decode EEG signals, CNNs are designed to extract a wide range of features from signal sequences with few prior knowledge, which can reach competitive performances. It has been proved that standard CNNs can be used as a general-purpose tool for brain-signal decoding tasks. Besides, keeping the architecture generic also increases the further applications. To deal with raw EEG signals, the CNNs should typically have two convolutional layers to better handle the large number of input channels, where one input channel per electrode compared to three input channels in RGB images. The first convolution is used across time to precede temporal information and the second convolution is used across electrodes to handle electrode relations. Therefore, the DL framework can be regarded as an integration of spatial block and temporal block. Many tricks are proposed to improve the performance of DL framework on spatial and temporal dimensions (Schirrmeister et al. 2017).

Meanwhile, many attempts are conducted to combine DL methods with the existing analysis methods. The EEG signals are converted into new characteristics by feature extraction methods, which can be fed into DL frameworks with more concrete information. Recently, by modifying the filter-bank common spatial patterns methods, EEG signals are turned into new temporal representations and a convolutional neural network architecture is introduced for motor imagery EEG data classification. The framework outperforms the existing results on the MI dataset (S. Sakhavi et al. 2018). Concretely, the first two layers of these frameworks should perform a temporal and a spatial convolution. Besides, they embed all the computational steps in a single network, and thus all steps can be optimized jointly. Also, due to having several pooling regions with one trial, these frameworks can learn a temporal structure of EEG signals, which is proved to help classification.

As for design choices, we give details for some of these aspects. (1) Batch normalization standardizes intermediate outputs of the network to zero mean and unit variance for a batch of training samples. This is meant to facilitate the optimization by keeping the inputs of layers closer to a normal distribution during training (S. Ioffe and Szegedy 2015). (2) Dropout randomly sets some inputs for a layer to zero in each training update. It is meant to prevent co-adaption of different units and can be seen as analogous to training an ensemble of networks. There are many other ways to improve accuracies and we do not list all here.

16.2.2 Recurrent Neural Networks

A recurrent neural network (RNN) is an extension of a conventional feedforward neural network which can handle a variable-length input. The RNNs can handle the variable-length sequence by having a recurrent hidden state whose activation at each time is dependent on that of the previous time. Standard RNNs are hard to train due to the well-known vanishing or exploding gradient problems. To address these problems, long short-term memory (LSTM) is proposed as the gated recurrent network architectures (Golmohammadi et al. 2017). The most commonly used architecture is described in (Graves and Schmidhuber 2005) as follows:

$$i_t = \sigma\left(U^i x_t + W^i s_{t-1} + p^i \cdot c_{t-1} + b^i\right) \tag{16.1}$$

$$f_t = \sigma\left(U^f x_t + W^f s_{t-1} + p^f \cdot c_{t-1} + b^f\right) \tag{16.2}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g(U^c x_t + W^c s_{t-1} + b^c) \tag{16.3}$$

$$o_t = \sigma(U^o x_t + W^o s_{t-1} + p^o \cdot c_t + b^o) \tag{16.4}$$

$$s_t = o_t \cdot g(c_t) \tag{16.5}$$

where i_t, f_t, c_t, o_t, and s_t are the input gate, forget gate, cell state, output gate and block output at time instance t, respectively; x_t is the input at time t; U^* and W^* are the weight matrices applied on input and recurrent hidden units, respectively; $\sigma(.)$ and g (.) are the sigmoid and tangent activation functions, respectively; p^* and b^* are the peep-hole connections and biases, respectively; and $\cdot$ means element-wise product. More specifically, a memory block of LSTMs is shown in Fig. 16.1.

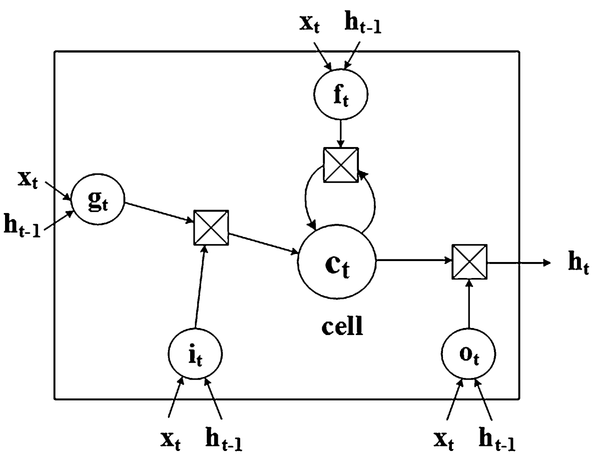

Fig. 16.1 An LSTM memory block

16.3 Deep Learning Framework with EEG Signals: Two Examples

In this chapter, we introduce two examples of DL-based studies to verify the superiority of deep learning methods for multi-channel EEG signals.

Opening and closing the eyes are fundamental behaviors for directing attention to the external versus internal world. However, it remains indistinct whether the resting state of eyes-open relative to eyes-closed are associated with different topological organizations of functional brain networks. Studies on resting-state functional networks from an electrophysiological perspective can take advantage of high temporal resolution. On consideration, EEG signal is the chief source for input, which contains a great deal of physiological information of a working brain. Here, we give two baseline methods to focus on eye state detection, convolutional neural networks, and recurrent neural networks, respectively. The section EEG acquisition and preprocessing is same with the section complex network analysis.

16.3.1 Convolutional Neural Networks

In the experiment, we reach a network with depth of 4 layers. Table 16.1 shows the details of the baseline method CNN. In the framework, a convolutional layer with a size of $[61 \times 1]$ is initially performed as input layer. Another $[1 \times 10]$ convolutional layer is employed with it, and then a dense layer with 100 nodes is appended ended with a softmax classifier.

A unit in the CNN is denoted by $x(l, m, j)$, where l is the layer, m is the feature map, and j is the position of the unit in this feature map. Likewise, $\sigma(l, m, j)$ is denoted as the scalar product between a group of input neurons and the weight connection between these neurons with l, m and j sharing equal meanings:

$$x(l, m, j) = f(\sigma(l, m, j)) \tag{16.6}$$

where f is the rectified linear units function (Nair et al. Nair and Hinton 2010) used for whole network layers.

Notably, each neuron of one feature map in each convolutional layer shares the same set of weights, which aims to decrease the amount of weight parameters. And they are attached to a subset of the neurons of former layer, which depend on the exact position of this neuron. Or rather, the neuron weights are trained independently

Table 16.1 Details of the CNN framework

Layers	Output size	CNN
Input	61×100	–
Convolution	1×100	61×1, map 8
Convolution	1×10	1×10, map 16, stride 10
Dense layer	100	fully connected
Softmax	2	Softmax

to their corresponding receptive fields. Let layer n as Ln, then the information transmission process could be described as:

1. For layer L1

$$\sigma(1, \mathrm{m}, \mathrm{j}) = \omega(1, \mathrm{m}, 0) + \sum_{i=0}^{i<Ne} I_{i,j} w_{(1,m,i)} \tag{16.7}$$

where $w(1, m, 0)$ is a threshold and $w(1, m, i)$ denotes a set of weights with $N_e = 61$. Here, m corresponds to the convolutional kernel used in this framework.

2. For layer L2

$$\sigma_{(2,m,j)} = w(2, m, 0) + \sum_{\mathrm{i}=0}^{i<10} x(1, m, j * 10 + i) \cdot w(2, m, i) \tag{16.8}$$

where $w(2, m, 0)$ is a threshold. This layer is employed to extract valid temporal features.

3. For layer L3

$$\sigma_{(3,j)} = w(3, 0, j) + \sum_{\mathrm{i}=0}^{i<100} x(3, i) \cdot w(3, i) \tag{16.9}$$

where $w(3, 0, j)$ is a threshold, and each neuron of L3 is fully connected to each neuron of L2.

We use the classical back propagation as learning algorithm to tune up the thresholds and weights of the network (D. E. Rumelhart et al. 1986), which is reflected by the promotion of model accuracy on validation set. As a loss function, cross-entropy objective function is employed for model performance estimation. Figure 16.2 shows the architecture of the proposed convolution neural network.

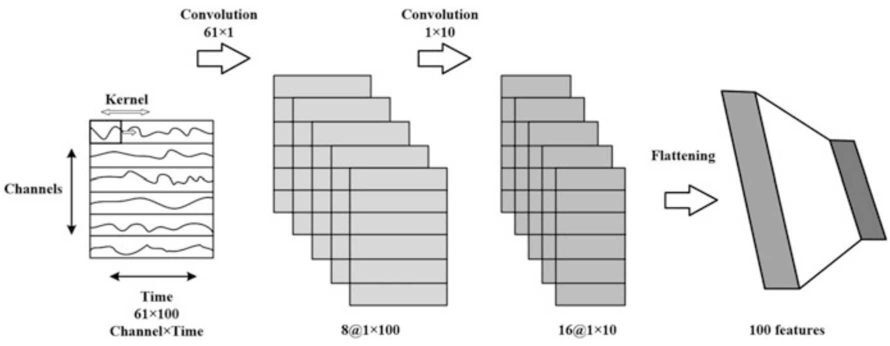

Fig. 16.2 The architecture of the proposed convolution neural network

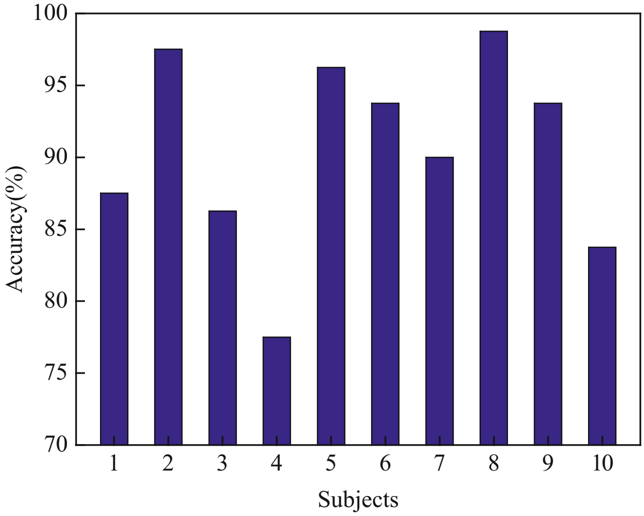

Fig. 16.3 Performances of the CNN framework on each subject

Table 16.2 Details of the LSTM framework

Layers	Output size	LSTM
Input	61×100	–
LSTM	61×100	100 units
LSTM	61×10	50 units
Dense layer	100	Fully connected
Softmax	2	Softmax

The developed CNN is trained to recognize different eye states for each subject. Individual performances are obtained on 10 subjects, and they are shown in Fig. 16.3. We find that the CNN framework is effective on the EEG dataset with an average accuracy of 90.5%. The performances of five subjects are over the average accuracy while the other five are below it. These results reflect that the CNN model provides an effective relationship between eye states and EEG signals.

16.3.2 Recurrent Neural Networks

The deep LSTM architecture is illustrated in Table 16.2. It consists of an input layer, the first sequence-to-sequence LSTM layer, a many-to-one LSTM layer, a 20% dropout layer, and a final sigmoid activation function for binary classification. The first hidden layer contained 50 LSTM units while the second hidden layer used 10 units. Dropout on the input gates to each LSTM layer and between the final LSTM and fully connected sigmoid layer served as a method of regularization and is

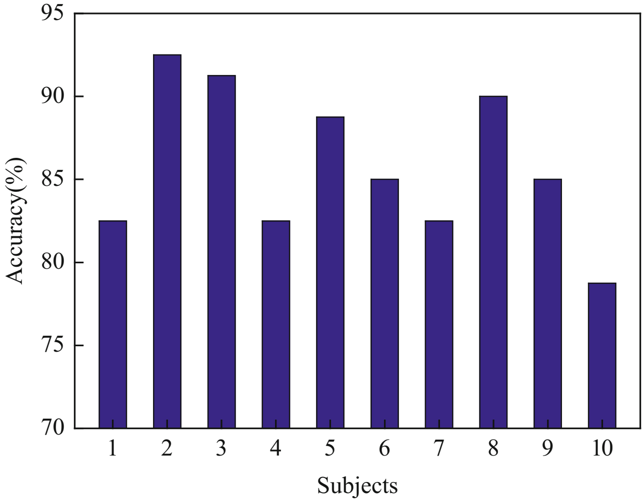

Fig. 16.4 Performances of the LSTM framework on each subject

set to 20%. The purpose of the recurrent connection in a LSTM is to store important long-term dependencies.

The developed LSTM is trained to recognize different eye states for each subject. Individual performances are obtained on 10 subjects, and they are shown in Fig. 16.4. As can be also seen, the LSTM framework receives an average accuracy of 85.88% when evaluated on the EEG dataset. The performances of four subjects are over the average accuracy while the other six are below it. The LSTM framework clearly enables learning effective representations from EEG signals to classify eyes-closed and eyes-open states.

References

Golmohammadi M, et al. Gated recurrent networks for seizure detection. In: Signal Processing in Medicine and Biology Symposium (SPMB). 2017. p. 1–5.

Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM networks. In: International Joint Conference on Neural Networks. 2005. 2047–2052.

Gulli A, Pal S. Deep Learning with Keras. Birmingham: Packt Publishing Ltd; 2017.

Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. 2015. p. 448–456.

Nair V., Hinton GE. Rectified linear units improve restricted boltzmann machines. In: International conference on machine learning. 2010. 807–814.

Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323(6088):533.

Sakhavi S, Guan C, Yan S. Learning temporal information for brain-computer interface using convolutional neural networks. IEEE Trans Neural Netw Learn Syst. 2018;29(11):5619–29.

Salazar-Gomez AF, et al. Correcting robot mistakes in real time using EEG signals. In: IEEE International Conference on Robotics and Automation (ICRA). 2017. p. 6570–6577.

Schirrmeister RT, et al. Deep learning with convolutional neural networks for EEG decoding and visualization. Hum Brain Mapp. 2017;38(11):5391–420.