EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

Chinese fingerspelling sign language recognition using a nine-layer convolutional neural network

Ya Gao[1], Chengchong Jia[1], Hongli Chen[1], Xianwei Jiang[1,2,*]

[1]School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing 210038, China

[2]Joint Accessibility Key Laboratory, China Disabled Persons' Federation, Nanjing 210038, China

Abstract

INTRODUCTION: Sign language is a form of communication and exchange of ideas by people who are hearing-impaired or unable to speak. Chinese fingerspelling is an important component of Chinese sign language, which is suitable for denoting terminology and using as the basis of gesture sign language learning.

OBJECTIVES: We propose a nine-layer convolutional neural network (CNN) for the classification of Chinese sign language.

METHODS: With self-learning and self-organization abilities, CNN is committed to processing data with similar network structure. CNN has a good application prospect in the aspect of image classification and plays a very important role in the classification of Chinese sign language.

RESULTS : Through experiments on 1320 data samples of 30 categories, the results show that the classification accuracy based on the nine-layer convolutional neural network can reach up to 89.69± 2.10 %, it can be seen that this method can effectively classify Chinese gestures.

CONCLUSION: We proposed a nine-layer convolutional neural network (CNN) that can classify Chinese sign language.

Keywords: deaf-mute, sign language recognition, Chinese fingerspelling recognition, convolutional neural network, stochastic pooling, batch normalization technology, dropout method.

Received on 24 August 2020, accepted on 30 September 2020, published on 12 October 2020

doi:10.4108/eai.12-10-2020.166555

* Corresponding author. Email: jxw@njts.edu.cn

1. Introduction

In daily life, communication is essential. However, In the world, there are thousands of people suffering from hearing impairment [1]. This special group of people is called deaf-mutes, and they cannot communicate through language as normal people can. As a way of expressing information, sign language (SL) has become the main way for deaf people to communicate with the outside world. With the development of artificial intelligence technology, novel, natural and convenient human-computer interaction has become a new trend in various industries. But sign language is not international. It is about more than just hand movements. They take advantage of both manual functions (hand gestures, movements, position and direction), and facial features that are colloquially called "non-manual" (eye gaze, mouth gestures and facial expressions) and upper body posture (nodding/shaking head and shoulder direction) [2]. Chinese Sign Language (CSL) can be generally divided into gesture sign language and fingerspelling language [3]. Gesture sign language emphasizes the form and is relatively complex. However, fingerspelling language focuses on 30 basic finger languages (including 26 basic pinyin letters and 4 upturned tongues), which can be combined to express pinyin or some special meaning.

Sign language recognition (SLR) is a method that uses computer technology to translate sign language information into text, natural language, audio and other information for easy understanding and communication [4]. SLR is divided into two parts, hand feature extraction and hand feature model training. Since 1980, scientists and researchers around the world have been working to grant recognition technology. In 1983, G.J.Grimes in the United States invented data input gloves [5], which can help users to analyze and recognize 72 English letters. Therefore, G.J.Grimes is regarded as the first person to recognize sign language. With the rapid development of computer technology and artificial intelligence technology in recent years, more and more high-tech technologies have been applied to the study of sign language recognition technology. In today's world, the study of sign language recognition technology is divided into two kinds, one is the recognition technology using data gloves as sign language input, the other is the sign language recognition technology based on machine learning. There are many methods for sign language recognition. To solve the problem of serious background interference, incomplete feature extraction and low recognition accuracy commonly existing in traditional sign language recognition. Si Yang et al. proposed a sign language recognition algorithm based on RGB-D images. Compared with the traditional methods, the performance of this way is greatly improved [6]. Peng Ping et al. proposed a sign-sign recognition method based on surf-BOW characteristics. The method collects images through a camera, firstly locates and tracks gestures, then extracts SURF features, and then constructs Surf-BOW as sign language features and USES SVM for recognition[7]. Shen Juan et al. proposed a hidden Markov Model (HMM) sign language recognition method, which is based on the Kinect 3D node[8].In the field of sign language recognition, many traditional machine learning methods are faced with such problems as high accuracy, extensible row and robustness.

In recent years, the booming development of deep learning technology has brought new opportunities for more accurate and real-time sign language recognition [9]. The method of deep learning is based on the idea that using hierarchical concepts, computers can learn complex concepts by constructing simple ones.

Convolutional neural network (CNN) is an important form of deep learning, which is committed to processing data with similar network structures, such as time series and image data. In addition, CNN has a good application prospect in many application fields, especially image classification and auxiliary clinical diagnosis, due to its self-learning and self-organization capabilities [10].

In our paper, we propose a nine-layer convolutional network classification for the classification of Chinese sign language. Our CNN is fully optimized for each layer to achieve excellent performance. The main contributions of this paper are as follows: 1) Convolutional operation and pooling are used for sample data input, which not only reduces the number and calculation amount of parameters but also controls the risk of over-fitting of training data. 2) Batch normalization technology is adopted to normalize the input of the layer into a small batch, so as to solve the problem of continuous training change caused by parameter update of the previous layer, improve the gradient, and

accelerate the speed of learning convergence.3) Dropout method is adopted to solve the problem of time-consuming training.4) The proposed nine-layer CNN optimizes the previous eight-layer CNN.

The rest of this paper is organized as follows: the second part mainly describes the data set, the third part mainly scans the convolutional layer and some methods used in this paper, the fourth part provides the implementation results, the fifth part compares and discusses the methods proposed by us and the latest methods, the sixth part gives the conclusion of this paper.

2. Dataset

2.1 Data Collection

Establishing a data set with sufficient data is a necessary condition for a complete experiment. Therefore, we used the camera to take 1,320 Chinese finger language pictures from 44 different samples (the size of each image is 1080 × 1080), and each sample contains There are 26 basic letters and 4 tongue-rolling pronunciation words, a total of 30 categories, including most commonly used pronunciation words. Adobe Photoshop CS manually divides each Chinese finger picture, removes the hand-shaped area in the picture, and adjusts the image size to 256 × 256. At the same time, normalize the background color and set it to RGB (0, 0, 0), so that our data is richer and more abundant and accurate, reducing the contingency and uncertainty of experimental data. Finally, we save these images in TIF compression format, which can reduce the chance of image information loss. (see *Figure 1*).

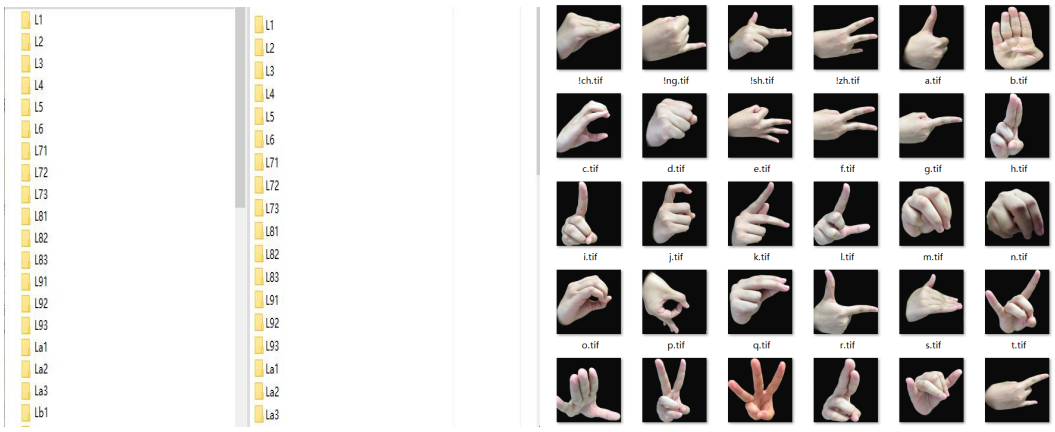

Figure 1. Save images in TIF compression format (Left) and images in L1 folder (Right)

2.2 Data Preprocessing

To ensure the accuracy of the results, we set all the images to the same size and background, and use the tool software Matlab R2018a to traverse all the pictures to get the pronunciation they represent. (see *Figure 2*)

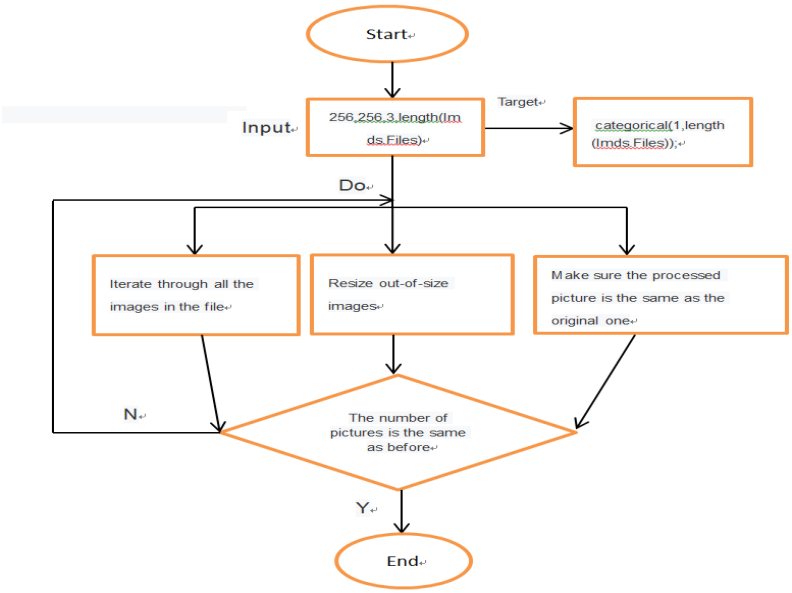

Figure 2. Flow chart for Manipulate pictures

Here, we present two sets of images. Two samples are selected from 44 samples. Two images are drawn from each set of samples. Among them, one image in the first group represents the letter T, and the other image represents the tongue-rolling pronunciation. SH. One image in the second group represents E, and the other image represents tongue-rolling ZH (see *Figure 3* and *Figure 4*):

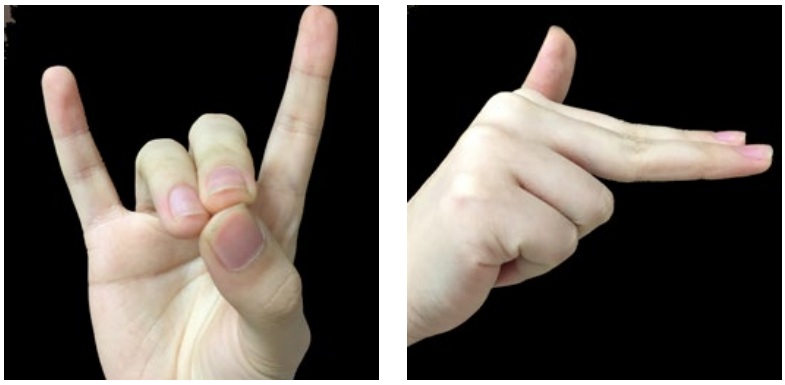

Figure 3. Remove and downsample (256 × 256) hand parts from the background. The first group of images, (left) T; (right) SH.

Figure 4. Remove and downsample (256 × 256) hand parts from the background. The second group of images, (left) E; (right) ZH.

Two groups were randomly selected from 44 samples, and different pronunciation words were randomly selected from the two groups of samples. In this way, the accuracy and completeness of sign language recognition can be better tested.

3. Methodology

3.1 Convolutional layer

The standard convolutional neural network covers all important phases of traditional computer vision methods: feature extraction and reduction, and classification. Its architecture can be thought of as a fusion of a trainable multi-level depth feature extractor and classifier. Among them, the function of the convolutional layer is to perform feature extraction [11], the pooling layer implements dimensionality reduction [12], and the fully connected layer completes classification.

The convolution layer uses multiple filters to perform discrete convolution operations on the input image to extract and combine local features to form a feature map. The convolutional layer is the core of the convolutional neural network, and most calculations are performed in the convolutional layer. On each convolutional layer, there will be a whole set of filters, and the activation maps formed by these filters are stacked in the depth direction to form the output of the convolutional layer [13]. The advantages of the convolutional layer mainly include two: local connection and weight sharing. A convolution operation is shown in *Figure 5*. Because all neurons of the feature map realize parameter sharing, the number of weights is greatly reduced, the learning efficiency is improved, and it is of great help to network training. Assuming the given two-dimensional image I (*x, y*) which is called "input" and the kernel K (*m*, n), the convolution can be expressed as

$$S(x,y) = (I * K)(x,y) = \sum m \sum n\, I(m,n)K(x-m, y-n) \quad (1)$$

Where (*m*, *n*) represents the size of kernel. As the convolution is commutative, the formula is equivalent to

$$S(x,y) = (I * K)(x,y) = \sum m \sum n\, I(x-m, y-n)K(m,n) \quad (2)$$

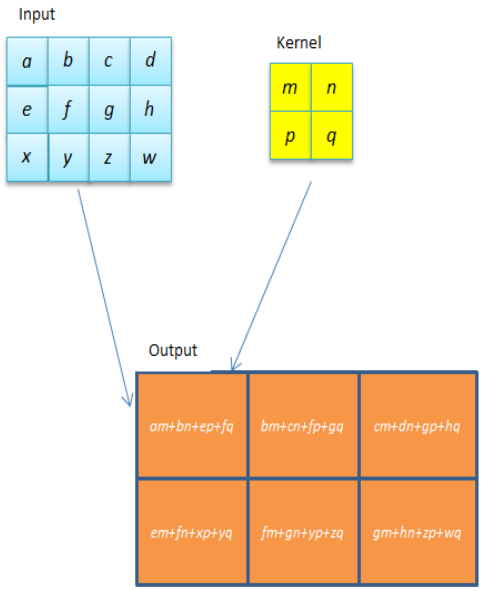

Figure 5. Illustration of the convolution operation

3.2 Pooling layer

A typical layer of a convolutional network includes three stages. The first stage computes multiple convolutions in parallel and generates a set of linear activation responses. In the second stage, each linear activation response passes through a non-linear activation function, such as the modified linear activation function described above. In the third stage, a pool function is introduced to further adjust

the output of this layer. Usually a pooling layer is periodically inserted between the convolutional layers [14]. Its function is to gradually reduce the spatial size of the data volume. This can reduce the number of parameters in the network, decrease the consumption of computing resources, and effectively control the over-simulation together.

Using the pooling function, the output of the network is replaced with the overall statistical characteristics of the adjacent output at a certain position. Max pooling and average pooling are two commonly used pool methods. The former gives the max value of adjacent rectangular areas, and the latter calculates the average value of adjacent rectangular areas. *Figure 6* shows an example showing the max pooling and the average pooling.

Given the pool area R, the activation set U contained in R is

$$U = \{u_i | i \in R\} \quad (3)$$

Then the max pooling P_M can be expressed as

$$P_M = \max(U_R) \quad (4)$$

Another pool strategy, the average pooling PA is defined as:

$$P_A = \frac{\sum U_R}{|U_R|} \quad (5)$$

Where U_R is the number of elements in the set U.

Pooling can help the input representation to be approximately unchanged. Another benefit is that it helps reduce the computational burden. But the max pooling and the average pooling have their own disadvantages. The former is easy to over fit the training data and can only reduce the estimated average offset due to the error of the convolutional layer parameters and cannot be generalized to the test set. The latter may reduce the strong activation when many elements in the pool area are close to zero, and it only reduces the error. The estimated variance is due to the size of the finite field. In order to overcome the shortcomings of the max pooling method and the average pooling method, the stochastic pooling method (SP) has attracted the attention of many researchers. The stochastic pooling method is to output multiple distributed samples formed by activation of each pool area. Essentially, it is sub-sampling based on pixel

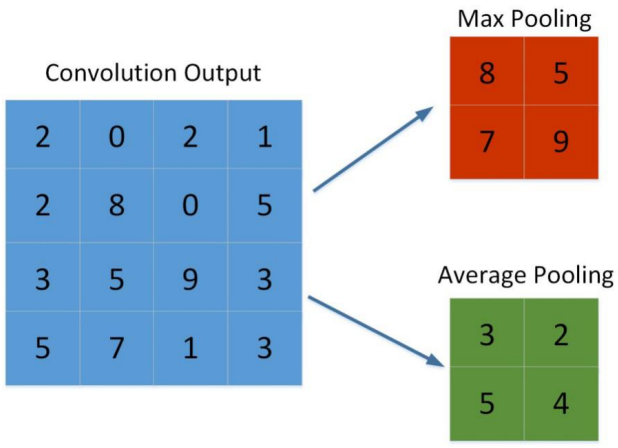

Figure 6 Examples of max pooling and average pooling

probability. Stochastic pooling combines the advantages of average pooling and max pooling. We can describe this process in two main steps:

（1）The probability map p_i is calculated from the original activation map a_i.

$$p_i = \frac{a_i}{\sum_A a_i} \quad (6)$$

（2）Choose a position l in the activation area A according to the probability p.

$$P_S = a_l, where\ l \sim P(p_1, \ldots, p_i, \ldots). \quad (7)$$

Figure 7 shows an example of a stochastic pooling. Firstly, generate a probability map. Secondly, the position l is randomly selected as 3 according to the polynomial

distribution. Finally, the output of P_s at the activation map (l, 3) is 2.5. Stochastic pooling provides excellent performance. Therefore, the pooling layer can continuously reduce the size of the data space. There by reducing the number of parameters and the amount of calculation. At the same time, overfitting is also controlled.

3.3 Fully connected layer

The fully connected layer plays a role of "classifier" in the whole convolutional neural network. If operations such as convolution layer, pooling layer map the original data to the hidden layer feature space, the full connection layer maps the learned "distributed feature representation" to the sample marker space. In practice, each node of the full connection layer is connected to all nodes of the previous layer, which is used to synthesize the previously extracted features. The full connection layer is connected by the convolution kernel between the front layer and the back layer. From *Figure 8*, we can understand the function of convolutional layer:

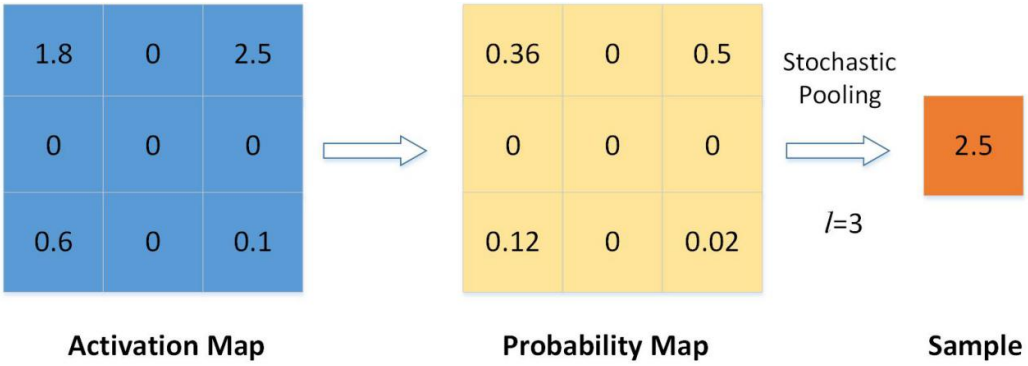

Figure 7 An example of a Stochastic pooling

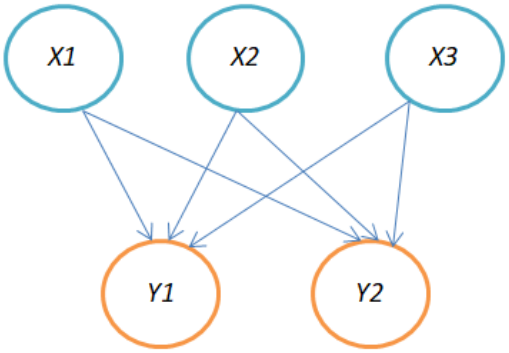

Figure 8 The working principle of the full connection layer

(X_1, X_2, X_3) is the input of the full connection layer, and (Y_1, Y_2) is the output. The connection between the two layers is the convolution kernel:

$$\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{bmatrix} \quad (8)$$

Through the convolution kernel, we can derive:

$$(X_1, X_2, X_3) * \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{bmatrix} = (Y_1, Y_2) \quad (9)$$

This is obvious in the case of a full connection layer with many parameters. In the forward calculation process, which is a linear weighted sum process, each output of the full connection layer can be regarded as each node of the previous layer multiplied by a weight coefficient W, and

finally obtained by adding a bias value b. Assuming a full connection layer with 50*4*4 neuron nodes in the input and 500 nodes in the output, a total of 50*4*4*500= 400,000 weight parameters W and 500 bias parameters b.

Let's use a simple network which shows in *Figure 9* , to specifically introduce the derivation process.

Among them, X_1, X_2, X_3 are the inputs of the fully connected layer, and Y_1, Y_2 and Y_3 are the outputs. Based on what I did earlier in formula (9), there are:

$$Y_1 = W_{11} * X_1 + W_{12} * X_2 + W_{13} * X_3 + b_1 \quad (10)$$

$$Y_2 = W_{21} * X_1 + W_{22} * X_2 + W_{23} * X_3 + b_2 \quad (11)$$

$$Y_3 = W_{31} * X_1 + W_{32} * X_2 + W_{33} * X_3 + b_3 \quad (12)$$

It can be written in the below matrix form:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (13)$$

From the *Figure 10*, we can know backpropagation of the fully connected layer, which we take our first fully connected layer as an example. This layer has 50*4*4=800 input nodes and 500 output nodes.

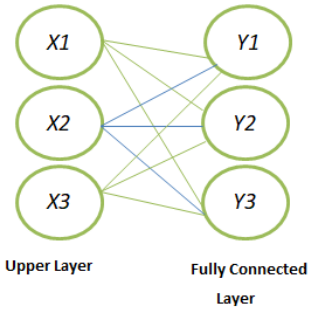

Figure 9 Forward calculation process of full connection layer

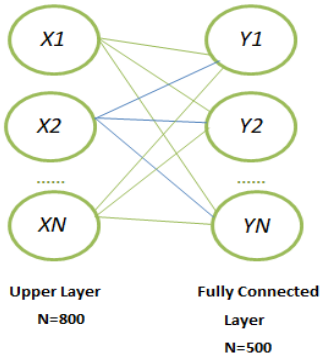

Figure 10 Backpropagation of the fully connected layer

Since we need to update W and b and pass the gradient forward, we need to calculate the following three partial derivatives.

1. Take the derivative of the output of the previous layer (the input of the current layer)

If we know the gradient transferred to this layer $\frac{\partial loss}{\partial Y}$, we can obtain the partial derivative of loss with respect to X by the chain rule. First, we need to obtain the partial derivative of the output Y_i of the layer to the input

$$X_j : \frac{\partial Y_i}{\partial X_j} = \frac{\sum_j^{800} w_{ij} * X_j}{\partial X_j} = w_{ij} \quad (14)$$

.

Then obtain the partial derivative of loss with respect to x by the chain rule:

$$\frac{\partial \text{loss}}{\partial X_k} = \sum_j^{500} \frac{\partial \text{loss}}{\partial Y_j} \frac{\partial Y_j}{\partial x_k} = \sum_j^{500} \frac{\partial \text{loss}}{\partial Y_j} * w_{jk} \quad (15)$$

The result of the derivation above also confirms my previous sentence: In the back propagation process, if the a node of the x-th

layer contributes to the *b* node of the *x*+1 layer through the weight *W*, then in the back propagation process , The gradient propagates from node b back to node a through weight *W*.

If we train 16 images at a time, that is, batch_size=16, then we can transform the calculation into the following matrix form (As shown in *Figure 11*).

2. Differentiate the weight coefficient *W*

Our forward calculation formula is as shown in the formula (10), formula (11), formula (12), the formula shows $\frac{\partial Yi}{\partial w_{ij}} = X_j$, so:

$$\frac{\partial \text{loss}}{\partial w_{kj}} = \frac{\partial \text{loss}}{\partial Y_k}\frac{\partial Y_k}{\partial w_{kj}} = \frac{\partial \text{loss}}{\partial Y_k} * X_j \qquad (16)$$

When batch_size=16, write it in matrix form (As shown in *Figure 12*).

Figure 11 Take the derivative of the output of the previous layer (the input of the current layer)

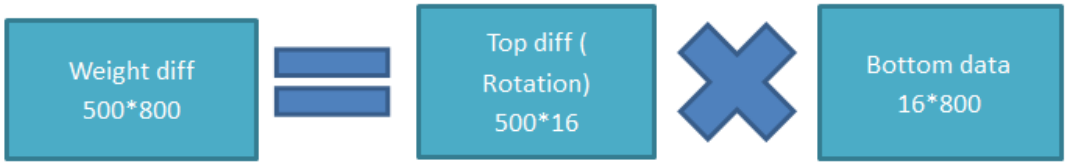

Figure 12 Differentiate the weight coefficient W

3. Find the derivative of the bias coefficient *b*

From the previous derivation formula, we can know: $\frac{\partial Yi}{\partial b_i} = 1$. That is, the partial derivative of loss to the bias coefficient is equal to the partial derivative of the output of the previous layer.

When batch_size=16, just add the partial derivatives of the same *b* corresponding to different batches, and write it in the form of a matrix that is multiplied by a matrix of all ones(As shown in *Figure 13)*.

Each node of the fully connected layer is connected with all the nodes of the previous layer, and is used to integrate the features extracted from the front. Because of its fully connected characteristics, generally the parameters of the fully connected layer are also the most. The core operation of full connection is matrix vector product: Y=W*X, the essence is to linearly transform from one feature space to another feature space. Any dimension of the target space (a cell in the hidden layer) is considered to be affected by every dimension of the source space. Regardless of rigor, it can be said that the target vector is the weighted sum of the source vector.

In CNN, full connection often appears in the last few layers and is used to weight and sum the previously designed features. For example, in mnist data set, the previous convolution and pooling are equivalent to feature engineering, and the latter full connection is equivalent to feature weighting. Convolution is equivalent to the intentional weakening of the full connection. According to

the inspiration of the local field of view, the weak influence outside the local area is directly wiped out to zero influence; a little force is also made, and the parameters used by different parts are actually the same. Weakening reduces the number of parameters, saves the amount of calculation, and specializes in not being greedy for more completeness, forcing further reduction of parameters.

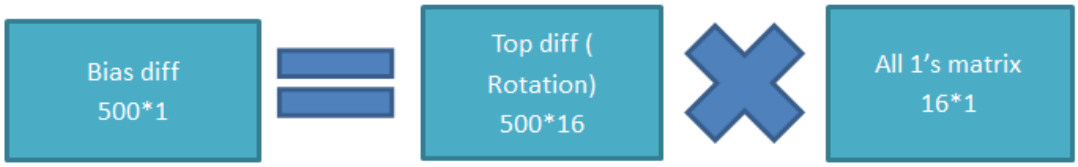

Figure 13 Find the derivative of the bias coefficient b

3.4 ReLU Function

ReLU is a commonly used activation function in artificial neural networks. It is a nonlinear function represented by ramp function and its variables.

It is defined as:

$$f(x)=\max(0,x) \quad (17)$$

In traditional neural networks, the Sigmoid system (Logistic-Sigmoid, Tanh- SIGmoid) is regarded as the core of the neural network. Sigmoid is similar to the neural response of humans and plays a huge role in many superficial models. However, in the recent years of convolutional network learning, we usually choose ReLU function instead of sigmoid or tanh function. First of all, they calculated some of the exponentials in the activation function, which is a relatively large amount of computation. For ReLU, not only is it not affected by these functions, but also the decentralized activity reduces the overall computing cost of the neural network [15]. Secondly, for the deep network, sigmoid also reduces the derivative value transferred to 0.25 times in the best case, and the gradient value obtained from the lower network is significantly smaller. When back propagation gradually becomes a cumulative process, gradient disappearance and gradient explosion will occur, resulting in poor training effect of the model. ReLU solves this problem compared to these methods. At the same time, ReLU will set the output of some neurons to 0, which causes the sparsity of the network, reduces the interdependent relationship between parameters, and alleviates the occurrence of overfitting [16]. Another advantage of ReLU is that it accelerates the convergence rate of stochastic gradient descent.

3.5 Batch normalization

Batch Normalization is a very simple but useful practice that speeds training convergence. In the shallow model, we use "whitening" treatment of input data so that its mean is 0 and variance is 1.Thus, the influence of input distribution changes on the model is not obvious during data training and testing [17]. With the progress of model training, when the parameters of each layer are updated, the output near the output layer changes dramatically. For deep neural networks, even if the output data has been "whiten", the updating of model parameters in training is still likely to cause drastic changes in the output near the output layer. This numerical instability often makes it difficult to train an effective depth model. By introducing batch normalization (BN), the distribution of the input value of any neuron in each layer of the neural network is forcibly pulled back to the standard normal distribution with the mean value of 0 and variance of 1 by optimizing the variance size and mean position. Make the activation input value fall in the area where the nonlinear function is sensitive to the input, so that small changes in the input will lead to large changes in the loss function, make the gradient increase, and avoid the problem of gradient disappearance. Moreover, the increase of the gradient means that the learning convergence speed is fast and accelerate the training speed greatly.

The formula of the batch normalization can be indicated as follow：

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (18)$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \quad (19)$$

$$\widehat{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \gamma}} \quad (20) \qquad y_i = \alpha\widehat{x_i} + \beta \quad (21)$$

Where [x_i] denotes input set, [18] is output in a mini-batch. In our study, we order the mini-batch size as 256. Additionally, α and β are learnable parameters, which realize the reconstruction of transform. A tiny number γ is added to prevent divide by zero. As shown in *Figure 14*. BN is generally utilized before the nonlinearity of the previous layer, and then the output of BN changes into the input of next layer.

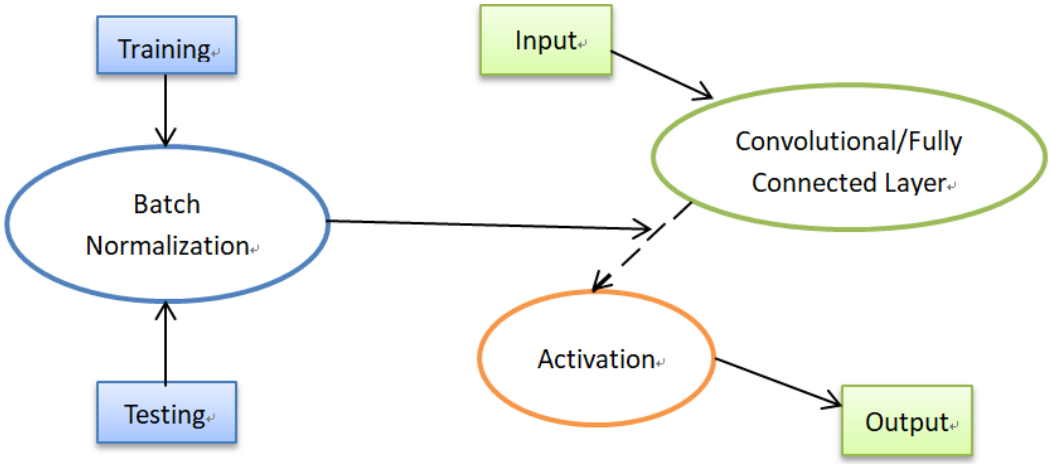

Figure 14. Structure illustration of batch normalization

3.6 Dropout

In neural networks, overfitting and training time are common problems. Although the model has a small loss in training data and high prediction accuracy, it has a large loss function in test data and low prediction accuracy. This is also known as bad generation of new data. To solve this problem, Dropout was introduced [19]. The solution is to go through all layers of the neural network in each training batch, and normalize the distribution of results according to a certain probability *p* of discarding, that is, discarding some neurons randomly, while retaining other neurons with the remaining probability (1-*p*).Here, the value of *p* is mostly considered to be 0.5 or 0.3 of the experience. In this case, Dropout randomly generates the most network structures, which helps reduce overfitting. In this way, the abandoned neurons are set to 0, and the remaining neurons constitute a small-scale new neural network composed of a few nodes, which not only simplifies the original neural network but also overcomes the overfitting phenomenon [20].

As shown in *Figure 15*, the green polygon indicates the normal neuron and the pink polygon denotes the dropout neuron. After application of dropout technology, the entire neural network was shrunk and easier to train, which can effectively mitigate the occurrence of over-fitting.

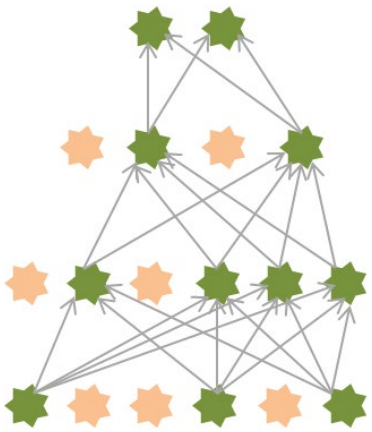

Figure 15. Network with application of dropout

4. Experiment results

4.1 Experiment setting

In order to test the proposed method, some training parameters are set. We set MaxEpochs=30, InitialLearnRate=0.01, and MiniBatchSize=256. At the same time, we set the parameters for different training layers. Specific experimental settings are showed in *Table 1*.

After iterating over the training samples, we get the accuracy and loss that are shown in *Figure 16*.

Table 1. Specific experimental Settings

```
options = trainingOptions('MaxEpochs',30,...
   'InitialLearnRate',0.01,...
'MiniBatchSize',256,...
'LearnRateDropFactor',0.1,...
'L2Regularization',0.005,...

);
```

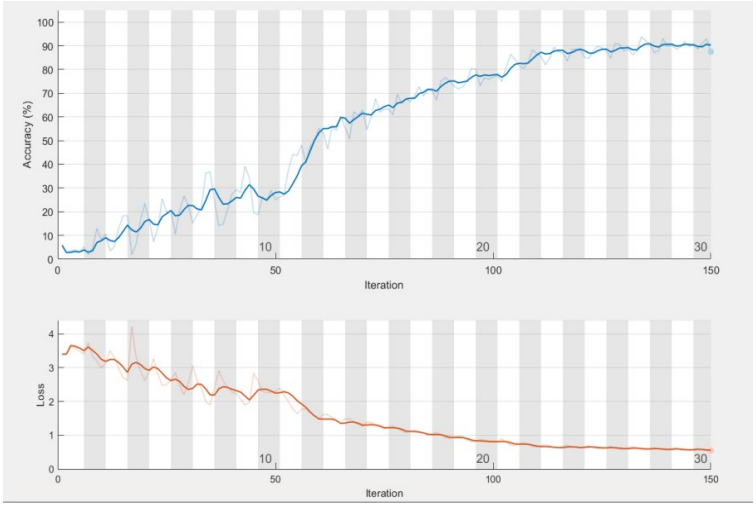

Figure 16. Training Progress

4.2 Structure of proposed CNN

Grid searching approach was used. We create a nine-layer CNN with structure shown in *Table 2*, where the input layer and softmax layer are not considered. The convolutional layer, batch normalization and ReLU layer are combined. Moreover, the first Layer that showed in the table also includes stochastic pooling (SP). Fully connected layer includes function of dropout. Thus, the first layer, six combined layers, the special fully connected layer and the fully connected layer constitute the whole nine-layer CNN. The other hyperparameters are descripted as follow: the size of image input layer is 256×256, stride size is set as 2 and filter size is set to 7*7 and 3*3, respectively. The numbers of filters are correspondingly changed according to the feature map. Dropout rate is selected as 0.4. According to the activations showed in the table, we can see that each output layer's height and width are shrinking regularly with the process going on.

4.3 Statistical analysis

Based on the data set formed by reading the 1320 sign language pictures one by one, we ran the training and test 10 times, the Elapsed time was set randomly each time. According to the results are showed in *Table 3* .we can find that the accuracy of our method are 87.50%, 88.28% , 87.11%, 89.45%, 88.67% , 89.06%, 91.41%, 89.45%, 93.36%, 92.58%. From the table, we can also see that the accuracy rate is more than 90%, appearing in the seventh time, the ninth time and the tenth time. The lower accuracy rate is occurred in the first time and the third time. In general, the accuracy of our method is higher than 87%. And the difference among the highest accuracy is small, which also demonstrates the stability of the experimental results.

Table 2. Structure of proposed CNN

Index	Layer	Parameters	Activations
	Input		256x256x3
1	Conv1, BN, ReLU, SP	Filter size = 7x7, No. filters = 16, stride = 2	128x128x16
2	Conv2, BN, ReLU	Filter size = 3x3, No. filters = 32, stride = 2	64x64x32
3	Conv3, BN, ReLU	Filter size = 3x3, No. filters = 64, stride = 2	32x32x64
4	Conv4, BN, ReLU	Filter size = 3x3, No. filters = 128, stride = 2	16x16x128
5	Conv5, BN, ReLU	Filter size = 3x3, No. filters = 128, stride = 2	8x8x128
6	Conv6, BN, ReLU	Filter size = 3x3, No. filters =256, stride = 2	4x4x256
7	Conv7, BN, ReLU	Filter size = 3x3, No. filters = 256, stride = 2	2x2x256
8	Dropout and FCL_1	Drop rate = 0.4, weights = 100x1024	1x1x100
9	FCL_2	Drop rate = 0.4, weights = 300x100	1x1x30
	Softmax		
	Output		

Table 3. Statistical Results of our method

Run	Accuracy
1	87.50
2	88.28
3	87.11

4	89.45
5	88.67
6	89.06
7	91.41
8	89.45
9	93.36
10	92.58
Mean+SD	89.69± 2.10

5. Discussions

5.1 Comparison of pooling methods

In this paper, we use the pooling function at the pooling layer to replace the output of the network with the overall statistical characteristics of the output adjacent to a certain location. We compare the max pooling with the average pooling [21]. The max pooling is the maximum value given to adjacent rectangular regions, and the average pooling is the average value calculated for adjacent rectangular regions. The max pooling and the average pooling have its own shortcomings, the former is easy to over-fit training data and can only reduce the estimated average offset due to convolution layer parameter errors [22]and cannot be generalized to the test set. The latter may reduce relatively strongly activated when many elements in the pool areas are close to zero, and it only reduces the error estimates of the variance due to the size of the finite field.

In order to overcome the shortcomings of the max pooling method and the average pooling method, the stochastic pooling method (SP) has attracted the attention of many researchers. Stochastic pooling method is to output multiple distributed samples formed by activation of each pool area. Essentially, it is a subsample based on pixel probability. Stochastic pooling combines the advantages of average pooling and max pooling. Stochastic pooling provides excellent performance. Therefore, the pooling layer can continuously reduce the size of the data space [23]. Thus, the number of parameters and the amount of computation are reduced. At the same time, overfitting has been controlled.

5.2 Comparison with the latest methods

In this experiment, four algorithms are studied and compared. The first is the Support Vector Machine (SVM) [24], the gesture recognition stage focus studied the support vector machine classifier. SVM is a pattern recognition mean based on structural risk minimization. It in solving small sample, nonlinear and high dimensional pattern recognition problem has many unique advantages. In the end, we chose the tree based on radial basis kernel function of support vector machine is used for signal classification. And select the optimal parameters by calculation. Based on the gesture recognition algorithm of SVM, SVM classifier is adopted and radial basis kernel function is selected to improve the accuracy and efficiency of gesture recognition. The second is Hidden Markov Model (HMM) [25], which is a pattern recognition method based on statistical theory. In the Chinese sign language recognition method established in HMM, the region of hand shape can be separated from the background by using the feature of histogram through the capture of gesture image, image processing and dimension reduction technology. Then, the region other than hand can be removed from the obtained hand image through dimension reduction processing, so as to obtain the contour of hand. Thus, the static simple sign language recognition can be achieved without other glove tools, and the accuracy rate reaches more than 85%. The traditional training method of HMM is based on the maximum likelihood criterion (MLE) of statistical probability, which can theoretically obtain the optimal result when the number of training samples is large enough. In the study of sign language recognition, it is very difficult to collect enough training samples. The third is dynamic time warping (DTW) [26].The idea of DTW is to conduct a global rough search first, and the gesture words to be recognized will be placed in a group of words with a small range, and then the words

will be recognized through a more accurate local search by HMM. Compared with single-layer recognition using HMM only ,the recognition rate of each word is increased from the original 2.364 seconds to 0.137 seconds, which is increased by 94.2%,and the recognition accuracy is also increased by 4.66%. The last one is Kinect [27]. Based on the Kinect technology, a dynamic sign language recognition method using finite state machine and DTW is proposed .Firstly, Kinect technology is used to obtain the depth image and bone feature information of human body. Then the hand depth image is obtained by hand segmentation algorithm, and then the HOG feature operator with high recognition accuracy is selected to extract hand features. This method can realize the recognition of commonly used sign language words and sentences, and the recognition accuracy can reach 95%. Relatively, the recognition accuracy of the Chinese sign language recognition algorithm based on a nine-layer convolutional neural network in this paper has reached 89.69± 2.10%, which is better than some traditional convolutional neural network and some modern algorithms.(As shown in *Table 4*)

Table 4. Comparison of SVM, HMM, DTW, Kinect

Comparative approach	Recognition accuracy
SVM	84.67%
HMM	85%
nine-layer CNN (ours)	89.69± 2.10%
DTW	94.2%
Kinect	95%

6. Conclusion

In this paper, a nine-layer convolutional network classification is proposed to classify Chinese sign language. Our proposed CNN structure fully optimizes each floor. The structure utilizes the excellent performance of SP, which not only reduces the number and calculation amount of parameters, but also controls the risk of over-fitting of training data. In addition, the batch normalization technology we adopted normalized the input of the layer into a small batch, so as to handle the problem of continuous training change triggered by parameter update of the previous layer, which improve the gradient and accelerates the speed of learning convergence. In this paper, the proposed nine-layer CNN is optimized, and the dropout method is adopted to solve the problems of overfitting and training time, so as to achieve better optimization performance. The post-sequence research will further optimize the network structure and perfect the training speed and recognition rate. Some popular deep neural networks such as ResNet, DenseNet[28], etc. will also be tried.

Acknowledgements

This work was supported from The Natural Science Foundation of Jiangsu Higher Education Institutions of China (19KJA310002), The Philosophy and Social Science Research Foundation Project of Universities of Jiangsu Province (2017SJB0668), The Natural Science Foundation of Jiangsu Province (16KJB520029).

References

[1] Maharani, A., F.R.C.P. Neil Pendleton MB.ChB., and F.R.C.P. Iracema Leroi M.D. Canada, M.R.C.Psych., M.D. UK %J The American Journal of Geriatric Psychiatry, *Hearing Impairment, Loneliness, Social Isolation, and Cognitive Function: Longitudinal Analysis Using English Longitudinal Study on Ageing.* 2019. **27**(12): p. 1348-1356.

[2] Koller, O., et al., *Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers.* 2015. **141**(DEC.): p. 108-125.

[3] Eifring, H.J.J.o.C.L., *Language Contact Across Time: Classical Chinese on Modern Public Signs in Taiwan.* 2019.

[4] Kumar, E.K., et al., *3D sign language recognition with joint distance and angular coded color topographical descriptor on a 2 - stream CNN.* 2020. **372**(Jan.8): p. 40-54.

[5] Han, L., *Study on Sign Language Recognition Method based on Deep learning.*

[6] Yang, S.I., et al., *Sign Language Recognition Algorithm Based on Color and Depth Image.* 2018.

[7] Ping, P. and L.J.M.C. Yuan-Feng, *Research on Sign Language Recognition Base on SURF-Bo W Features.* 2016.

[8] Shen, J., *Continuous HMM Sign Language Recognition based on Kinect 3D node.* Journal of Hefei University of Technology (Natural Science edition), 2017.

[9] He, S. *Research of a Sign Language Translation System Based on Deep Learning*. in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. 2019.

[10] Jiang, X., et al., *Classification of Alzheimer's Disease via Eight-Layer Convolutional Neural Network with Batch Normalization and Dropout Techniques.* 2020.

[11] Guoqiang, S. and Z. Xia, *Local Image Feature Descriptor Algorithm Based on Convolutional Neural Network.* 2020. **037**(001): p. 87-92.

[12] Tu, J., H.J.J.o.t.C.S.f.e. He, and T. Information, *Dimension Decreased Feature Extraction Based on Semantic Analysis.* 2014.

[13] Jianliang, L. and Y. Xianzhen, *"The Application of Fully Parallel Transposed FIR Filter in Accelerating Convolutional Neural Networks", "Modern Computer" Professional Edition, Volume: 2019(000),022, pp.19-21.* 2019(22).

[14] Junshan, L., *A Data Pooling Layer Design Based on MDC to Support Heterogeneous Data", Chinese Science and Technology Papers and Citation Database (CSTPCD), Year (Volume), Issue: .* 2020(000), 002, pp.81-82 , 46.

[15] Ang-Bo, J., W.J.T. Wei-Wei, and M. Technologies, *Research on optimization of ReLU activation function.* 2018.

[16] Printing, W.D., et al., *Optimal Design of ReLU Activation Function in Convolutional Neural Networks.* 2018.

[17] Hu, Q., *Combined with batch normalized multilayer perceptor Diabetes Prediction and Diagnosis Model, Computer System Application.* 2020. **029**(005): p. 182-188.

[18] Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks.* In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger, eds. Advances in Neural Information Processing Systems 25. Curran Associates, Inc., 2012: p. 1097-1105.

[19] Saeed-Ul, H., et al., *Virtual learning environment to predict withdrawal by leveraging deep learning.* 2019. **040**(005): p. 42-46.

[20] Ji, Z., *DROPOUT Optimization of Neural Network Overfitting problem.* 2020. **037**(002): p. 37-39.

[21] Lin, Y.Q., L.I.J.R.S. Jing-Wen, and Technology, *Comparison of RD Algorithm and BP Algorithm Under Severe Range Migration.* 2004.

[22] L, H.X., *Study on fault Diagnosis of Complex Mechanical System based on Hidden Markov Model and EM Algorithm* 2012, Huazhong University of Science and Technology.

[23] Yun, G., *Research on high-resolution Remote sensing Image Retrieval Based on CNN migration Feature Fusion and Pooling*. 2019.

[24] F, D.L., *Static Gesture Recognition and Application based on Hu Moment and Support vector Machine.* 2012, Wuhan University of Technology.

[25] Weining, Z., *Research on Chinese Sign Language recognition based on HMM_SVM.* Journal of changchun university 2011. **21**(010): p. 24-26.

[26] Yao gui-lin, y.h.-x., jiang feng, *A multilayer classifier sign language recognition method based on DTW/ISODATA algorithm* Computer engineering and applications, 2005(08): p. 45-47.

[27] Qian Cheng-hui, S.J.-y., XIA Tao, et al, *Kinect based sign Language recognition method.* 2019.

[28] Jiang, X., et al., *A Survey on Artificial Intelligence in Chinese Sign Language Recognition.* Arabian Journal for Science and Engineering, 2020: p. 1-36.