第十一届“认证杯”数学中国

数学建模国际赛

承　诺　书

我们仔细阅读了第十一届“认证杯”数学中国数学建模国际赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为:

我们选择的题目是:

参赛队员 (签名) ：

队员 1: 黄子安

队员 2: 马佳瑶

队员 3: 周若妍

参赛队教练员 (签名): 无

第十一届“认证杯”数学中国

数学建模国际赛

编 号 专 用 页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：

1695

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

A Model of Human Activities Classification Based on BP Neural Network

Abstract:

At present, the rapid development of machine learning and other technologies has made human activity recognition and classification based on inertial sensors widely used in various fields, so the research on human activity classification has certain significance. In this paper, we establish a BP neural network model to complete feature extraction and classification of human activities for a large number of data sets collected by sensors, and evaluate the generalization ability of the model while overcoming the overfitting problem so that the human activity classification algorithm can be applied to a wider range of scenarios.

For problem 1, a set of features and efficient algorithms are designed to classify 19 human behaviors from the data of these wearable sensors. Data processing is first performed to consolidate the obtained data into a single file and to standardize the data. After processing, the data obtained from 45 sensors were subjected to feature extraction using Factor Analysis, thus reducing the dimensions of variables in order to improve the learning speed of the neural network. After KMO and Barlett tests, the overall variance contribution of the extracted components reached more than 85%, achieving effective extraction of the data features.

Based on the extracted features we use a three-layer BP neural network algorithm to classify the obtained data and use the Levenberg-Marquardt (LM) algorithm to train the neural network. The training data based on the best performance up to 999 as well as the mean square error value of 9.1821 shows that the network has high accuracy and efficiency.

For problem 2, a feasible method needs to be designed to evaluate the generalization ability of the model. To solve the problem, we used the hold-out method and metrics such as accuracy, precision, recall, F1 score, and confusion matrix to perform the evaluation of the generalization ability of our established BP neural network. The viewable and regression fit lines obtained based on the evaluation of these metrics by using the software MATLAB show that the network has good performance with high classification accuracy and precision.

For problem 3, the overfitting problem needs to be investigated and overcome. We use regularization and Dropout discarding to solve the overfitting problem and train the neural network model again to obtain a model that can be more widely used.

At the end of this paper, we analyzed the strengths and weaknesses of the established model, and explain the prospect of the model.

Key words: Data Standardization, Factor Analysis, BP Neural Network, Levenberg-Marquardt Algorithm, the Generalization Ability.

Contents

I. Introduction

In order to indicate the origin of the human activities classification, the following background is worth mentioning.

1.1 Problem Background

The detection and recognition of human activity enables the understanding of human behavior and its application in various fields to bring wider convenience to people's lives. With the rapid development of data sampling, Micro Electro-Mechanical System (MEMS), Machine Learning, Neural Network and other technologies, as well as people's concern for their own health and the gradual improvement of their pursuit of quality of life, it makes human activity recognition and classification based on wearable sensors a popular research topic nowadays. Inertial sensor-based human activity recognition technology plays an essential role in remote monitoring and observation of the elderly and children through personal alarm systems, assisted living for the elderly, healthcare and rehabilitation, sports, games and entertainment, film and television production, various scientific research studies and virtual reality, etc.

Micro inertial sensors capable of data acquisition of human activity signals are mainly composed of accelerometers, gyroscopes and magnetometers, which are used to collect acceleration signals, angular velocity signals and magnetic fields generated by human activity. The development of miniature inertial sensor technology has led to its gradual development with small size, low production cost, low energy consumption and high sensitivity, which is expected to be integrated and used in various wearable devices, which will considerably increase the application in the field of human activity classification. Consequently, it is of great relevance to process the data obtained from the miniature inertial sensors and to classify human activities according to the processing results.

1.2 Restatement of the Problem

In order to classify the human activities, data were obtained using miniature inertial sensors and magnetometers positioned in different parts of the body. Each of the 19 activities was performed by 8 subjects (4 female, 4 male, aged 20-30 years) for 5 minutes each. The sensor acquired data at a sampling frequency of 25 Hz. The 5-minute signal was divided into 5-second signal segments, resulting in 480 (= 60 x 8) signal segments per activity.

Considering the background information, data obtained and restricted conditions identified in the problem statement, we need to build mathematical models to solve the following problems:

1. Perform multi-file processing on the data acquired by the wearable inertial sensors and design a set of features and efficient algorithms to classify 19 human behaviors based on the processed data.
2. Simplify the data set to reduce the data volume problem under the premise of ensuring the performance, and design a feasible method to evaluate the effect of the generalization ability of the model.
3. Process the neural network model and data, study and overcome the overfitting problem of the model, so that the designed human behavior classification algorithm can be widely used.

1.3 Our Work

First, we extract features from the data by Factor Analysis. Then we use the LM algorithm to build a BP neural network to classify the data, and evaluate the obtained results by using several evaluation indicators such as accuracy, precision, recall,F1 score and so on. Finally, we use regularization, drop out and other methods to reduce the occurrence of over fitting, and reevaluate the obtained indicators.

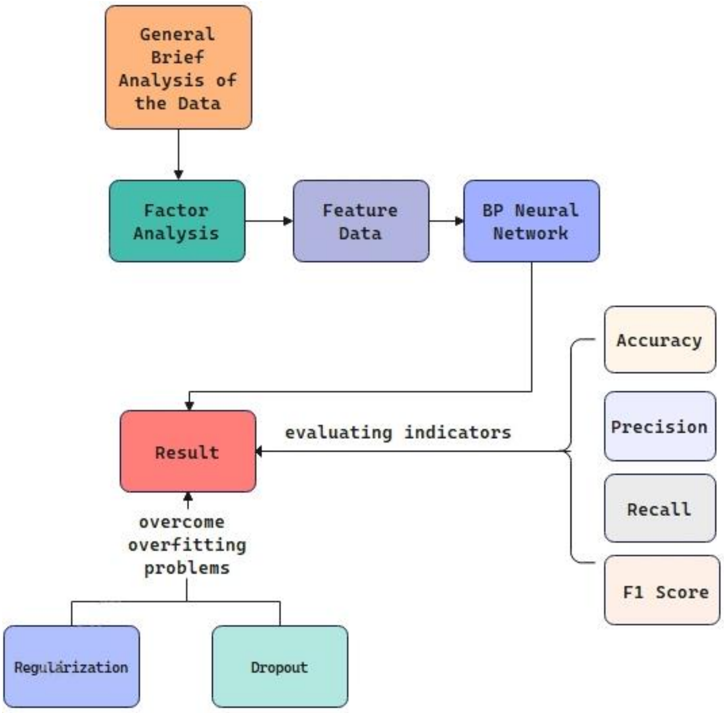

Figure 1: Our Work

II. Assumptions

By fully analyzing the problem, and in order to simplify our model, we have made the following sufficiently reasonable assumptions.

1.It is assumed that all the data acquisition objects are healthy and have no disabilities.

2. Assume that the subjects perform activities as required, have their own activity styles and are not limited by the activity methods.

3. Assume that the test is not interfered by external factors such as step height, elevator running speed, propeller length and quality.

III. Notations

Table 1: Name and label of human activities

ACTIVITIES	TAG
Sitting	1
Standing	2
Lying on back	3
Lying on right side	4
Ascending stairs	5
Descending stairs	6
Standing in an elevator still	7
Moving around in an elevator	8
Walking in a parking lot	9
Walking on a treadmill with a speed of 4 km/h in flat position and 15 deg inclined positions	10
Walking on a treadmill with a speed of 4 km/h in 15 deg inclined positions	11
Running on a treadmill with a speed of 8 km/h	12
Exercising on a stepper	13
Exercising on a cross trainer	14
Cycling on an exercise bike in horizontal position	15
Cycling on an exercise bike in vertical position	16
Rowing	17
Jumping	18
Playing basketball	19

Table 2: Symbol description

Symbol	Description
N	The number of samples collected by the sensor
X_i	Characteristic factor of human behavior
f	Neural network hidden layer
w_{ij}	The weight from node i to node j
O_i	Outputted human behavior classification serial number
prec	Precision
rec	Recall
f1-score	Harmonic average of model accuracy and recall rate

IV. Models

4.1 Model I: Factor Analysis

4.1.1 Analysis of the Problem 1 Part I

We need to process multi-file on the data obtained from the wearable inertial sensors and design a set of features and an efficient algorithm to classify 19 types of human activities based on the processed data. To solve the problem of multi-file processing, we will use Python to process the multiple files where the dataset obtained from the human wearable sensor is located, so that we can consolidate them to get one file, and get a more standard data format in that file, where all the activity signals of all the segments are in the same file, completing the standardisation of the dataset. To achieve the identification and classification of 19 types of human activities, features should be extracted from the data, and we will use the Factor Analysis model to group the more closely linked features into the same class and achieve dimensionality reduction on the data, so as to realize the purpose of data standardization and achieving the classification of human activities.

4.1.2 The Foundation of Model

Factor analysis method explores the basic structure of observation data by studying the internal dependence relationship among many variables, and classifies the complicated relationship into a few comprehensive factors. In this problem, it can be used to extract features from the data obtained by 45 sensors. Meanwhile, the reduction of the number of variables is conducive to the subsequent improvement of the learning speed and generalization ability of the neural network.

4.1.3 Algorithms used in the model

Based on the analysis above, we design our simulation arithmetic as follows.

1. General Brief Analysis of the Data

First we briefly described the relevant statistics for the obtained data. We have 19 folders in total, each of them has 8 folders, and each of these 8 folders have 60 text files. Also, there are 125 lines in each text files. So the total number of samples is

$$N = 114000$$

We drew visual scatter plots of the mean, maximum, and minimum values. By the scatter plots we found that the distribution ranges of these data have large differences, and to reduce the impact on the factor analysis, we first standardized the data.

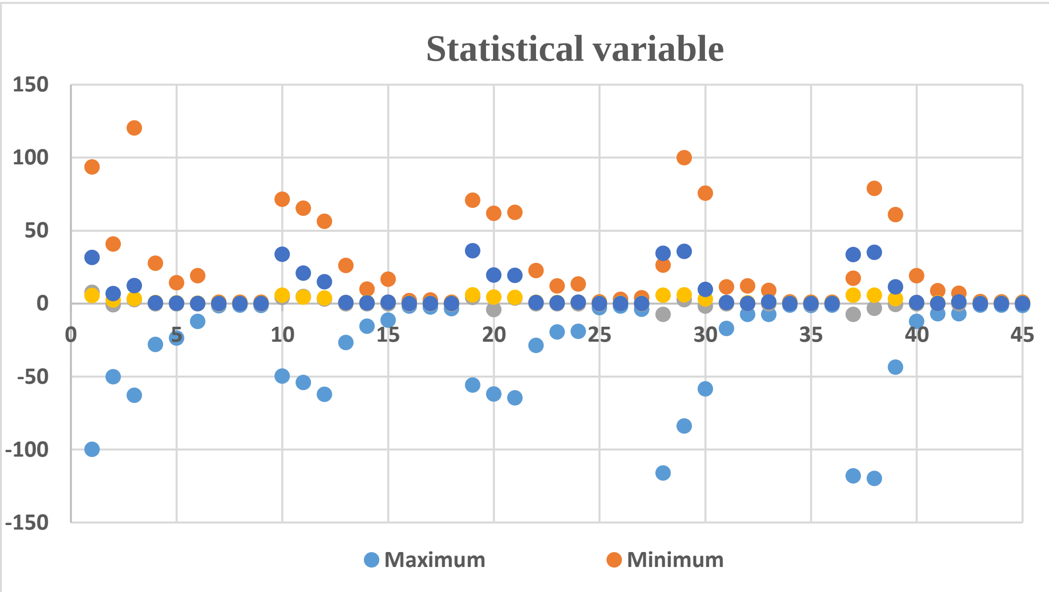

Figure 2: Data Scatter Plot

2. Data Standardization

In this problem, there are 45 indicator variables for factor analysis, $x1, x2 \ldots x45$, with a total of 1140000 data, i.e. there are 1140000 evaluation objects, and the value of the j-th indicator of the i-th evaluation object is x_{ij}. The value of each indicator x_{ij} is transformed into a standardized scalar $\tilde{x}_{ij}$ (1).

$$\tilde{x}_{ij} = \frac{x_{ij} - x_j}{s_j} (i = 1,2,\cdots,N; j = 1,2,\cdots,45) \quad (1)$$

The sample mean value of the j-th indicator (2)

$$\overline{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \quad (2)$$

The standard deviation of the sample of the j-th index (3)

$$s_j = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \overline{x}_j)^2, (j = 1,2,\cdots,45) \quad (3)$$

We standardized the data using Z-Score normalization.

Standard deviation formula (4)

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (4)$$

z-score standardized conversion formula (5)

$$Z = \frac{x-\mu}{\sigma} \quad (5)$$

After this standardization we eliminate the effect of the magnitude between different data.

3. Calculate the Correlation Matrix *R*

Correlation matrix (6)

$$R = (r_{ij})_{45\times 45} \quad (6)$$

$$r_{ij} = \frac{\sum_{k=1}^{N}\tilde{x}_{ij}\cdot\tilde{x}_{kj}}{N-1}, (i,j = 1,2,\cdots 45) \quad (7)$$

In the formula,

$$r_{ii} = 1, \quad r_{ij} = r_{ji}$$

and r_{ij} is the correlation coefficient between the i-th indicator and the j-th indicator

4. KMO and Bartley Tests

By performing KMO and Bartlett tests for the correlation matrix, we obtain the following data.

Table 3: KMO and Bartley tests

KMO sampling test tangibility number		0.747
Bartlett's sphericity test	**Approximate chi square**	26873782.160
	Degree of freedom	1035.000
	Significance	0.000

Noting that the KMO sampling fitness measures lie between 0.7 and 0.8 and the significance of the Bartlett's sphericity test is close to 0, it indicates that for the data obtained from these sensors, we can perform feature extraction by using the factor analysis method.

5. Calculate the Primary Loading Matrix

Calculate the eigenvalues of the correlation coefficient matrix R, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{45} \geq 0$, and the corresponding eigenvectors, $u_1, u_2, \cdots u_{45}$, and for u_j (8)

$$u_j = (u_{1j}, u_{2j}, \cdots u_{Nj})^T \quad (8)$$

The primary loading matrix(9)

$$A = [\sqrt{\lambda_1} u_1 \sqrt{\lambda_2} u_2 \cdots \sqrt{\lambda_{45}} u_{45}] \quad (9)$$

We estimated the factor loading matrix using principal component analysis, and calculated the contribution of each common factor based on the primary loading matrix. Combining the plotted gravel plot and the total variance explained plot, we found that the contribution reached 60% at the 9th component and 85% at the 22nd component, and we chose the 22nd component after a trade-off in order to ensure a small amount of information loss from the data.

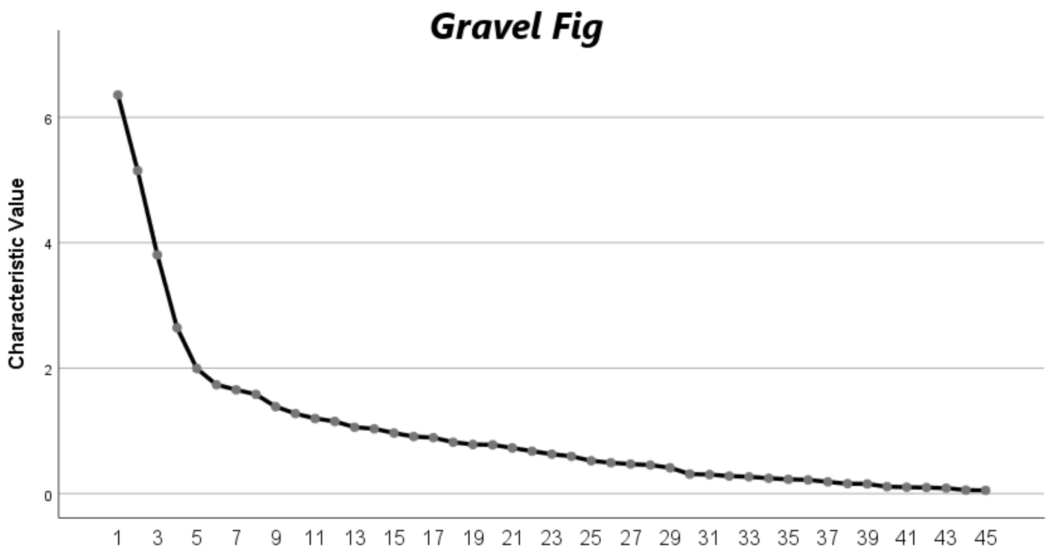

Figure 3: Gravel Fig

Table 4: Total Variance Explained

Component	Total	Initial Eigenvalues of Variance	Cumulative
1	6.356	14.125	14.125
2	5.149	15.125	25.567
3	3.807	16.125	34.027
4	2.644	17.125	39.904
5	1.995	18.125	44.337
6	1.735	19.125	48.193
7	1.654	20.125	51.869
8	1.582	21.125	55.385
9	1.386	22.125	58.465
10	1.275	23.125	61.299
11	1.197	24.125	63.958
12	1.151	25.125	66.516
13	1.058	26.125	68.867
14	1.033	27.125	71.163
15	0.965	28.125	73.309
16	0.909	29.125	75.33
17	0.892	30.125	77.312
18	0.819	31.125	79.132
19	0.783	32.125	80.871
20	0.778	33.125	82.599
21	0.727	34.125	84.216
22	0.676	35.125	85.719

6. Factor Rotation by Maximum Variance Method

On the basis of the primary factor load matrix obtained by the principal axis method, the transformation method of the rotated factor load matrix is obtained according to the simple structure criterion. So that the elements of each column of the transformed factor load matrix have the maximum variance after squaring each element while maintaining independence from each other.

7. Calculate the Factor Score

In this problem the regression method is used to obtain a single factor score function (10)

$$\begin{cases} \hat{F}_1 = b_{1,1}\tilde{x}_1 + \cdots + b_{1,45}\tilde{x}_{45} \\ \hat{F}_2 = b_{2,1}\tilde{x}_1 + \cdots + b_{2,45}\tilde{x}_{45} \\ \cdots \end{cases} \quad (10)$$

Denote the estimate of the j-th factor score F_j at the i-th sample point

$$\hat{F}_{ij} = b_{j1}\tilde{x}_{i1} + b_{j2}\tilde{x}_{i2} + \cdots + b_{jp}\tilde{x}_{ip} (i = 1,2,\cdots,45, j = 1,2,\cdots) \quad (11)$$

$$\begin{bmatrix} b_{11} & b_{21} \cdots & b_{j1} \\ b_{12} & b_{22} \cdots & b_{j2} \\ \vdots & \vdots & \vdots \\ b_{1,45} & b_{2,45} \cdots & b_{j,45} \end{bmatrix} = R^{-1}B \quad (12)$$

$$\hat{F} = (\hat{F}_{ij})_{n\times m} = X_0 R^{-1} B \quad (13)$$

In this formula, X_0 is the original data matrix (1140000×45), R is the correlation matrix, and B is the loading matrix.

4.1.4 Result Presentation

Considering the large number of variables before and after extraction, we use visual graphs to present our results.

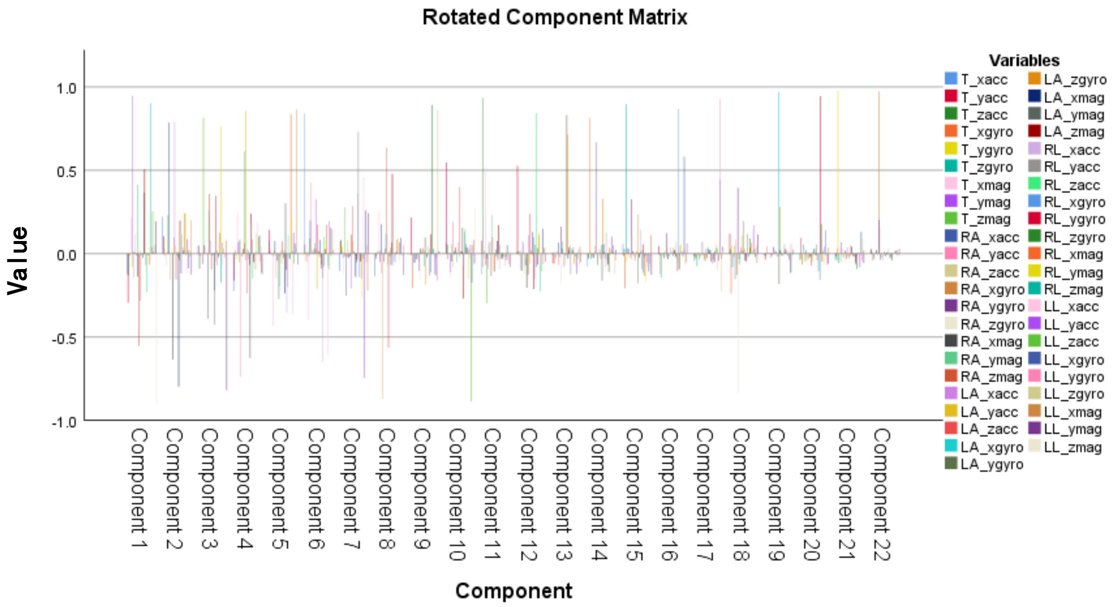

Figure 4: Rotated Component Matrix

4.1.5 Analysis of the Result

The given data were subjected to KMO test and Bartlett's test, and the overall variance contribution of each component after extraction reached more than 85%, and as shown by the visualized rotated component matrix, each component can replace the originally obtained 45 sensor data with better effect, we can believe that the effective extraction of data features can be achieved by using the factor analysis method of the above process.

4.2 Model II: Neural Network

4.2.1 Analysis of the Problem 1 Part II

We have extracted features from the data and now we need to design a classification algorithm based on these features for correlating the 19 activities with them. Considering the sufficient number of samples and the large number of final classifications, we choose to use a BP neural network algorithm to classify the obtained data.

4.2.2 The Foundation of Model

The learning capability of multi-layer networks is much enhanced compared to single-layer perceptrons, but more powerful learning algorithms are required to train multi-layer networks. The BP algorithm can be used not only for multilayer feedforward neural networks, but also for other types of neural networks, which are usually referred to as multilayer feedforward neural networks trained with the BP algorithm. The structure of the three-layer BP neural network is shown in the figure below.

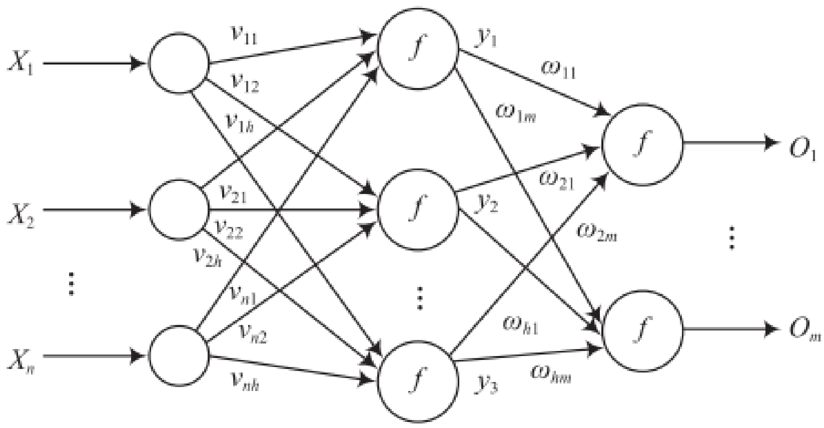

Figure 5: Three-layer Neural Network Structure

4.2.3 Algorithms used in the model

We use the Levenberg-Marquardt (LM) algorithm to train the neural network.

LM algorithm is an iterative algorithm for finding the extreme value of a function. LM algorithm combines the characteristics of both Newton's method for finding the extreme value and the gradient method for finding the extreme value.

The recursive formula of the Gaussian Newton method for finding the extreme value is (14)

$$x_{s+1} = x_s - H^{-1}G \quad (14)$$

Where H is the matrix of multidimensional vectors, G is the first order gradient of multidimensional vectors.

One problem with the Gauss-Newton method is that the matrix may not be inverti ble. This can be overcome by using the following modification to the approximate He ssian matrix (15)

$$G = H + \mu I \quad (15)$$

The recursive formula for the extreme value of the gradient method is(16)

$$x_{s+1} = x_s - \alpha \Delta f(x) \quad (16)$$

where α is the step of gradient descent and $\Delta f(x)$ is the first-order gradient of the multidimensional vector.

Using this gradient direction, the approximate performance index is recomputed. If a smaller value is the yield, then the procedure is continued with the μ_k divided by some factor $\vartheta > 1$. If the value of the performance index is not reduced, then μ_k is multiplied by ϑ for the next iteration step.

The LM algorithm equation is (17)

$$x_{s+1} = x_s - (H + \alpha\, I)^{-1} G \quad (17)$$

It can be seen that this formula adds a moderating factor αI to the Gaussian Newton formula H, where α is the step size and I is the unit matrix (since it is a matrix, the step size is expressed in matrix form here).

The LM algorithm has the following characteristics.

using a smaller α when the descent is too fast, making the whole formula close to the Gaussian Newton method.

A larger α is used when the descent is too slow, so that the whole formula is close to the gradient method.

4.2.4 The Solution of Problem

1. The data are imported and the test samples are stored in the X-vector and the result data are stored in the Y-vector.

2. Build the neural network and train the dataset using LM algorithm.

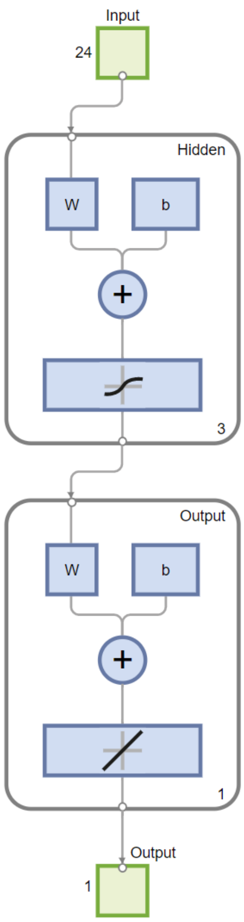

Figure 6: Neural Network Construction

For this problem, the BP neural network model consists of 1 input layer, 3 hidden layers and 1 output layer, the number of input and output nodes are 24 (dimensionality of principal components after dimensionality reduction) and 1 (output values 1~19 represent different human activities), epochs are set to 50, learning rate (lr) is set to 0.001, and LM back propagation algorithm is used for network training. The training and prediction of the BP neural network were performed on the training and testing sample sets, respectively. The imported data were divided into training set, validation set and test set. The data were split into the following parts.

70% is used for training.

15% is used to verify that the network is generalizing and to stop training before overfitting.

15% was used to independently test the network generalization.

Data training is performed using MATLAB software. Training will continue until a pre-defined stopping condition is met (Epoch greater than or equal to 1000 or the validation error of the iteration continues to increase).

4.2.5 Analysis of the Result

The corresponding (target) network predictions (output) about the training set, validation set and test set are displayed by generating linear regression plots. If the fit is perfect, the data should fall along a 45 degree line where the network output is equal to the response. For this problem, it can be seen from the figure below that the dataset fits very well. To get more accurate results, the dataset can be trained again with different network initial weights and biases, and the network can be improved after retraining.

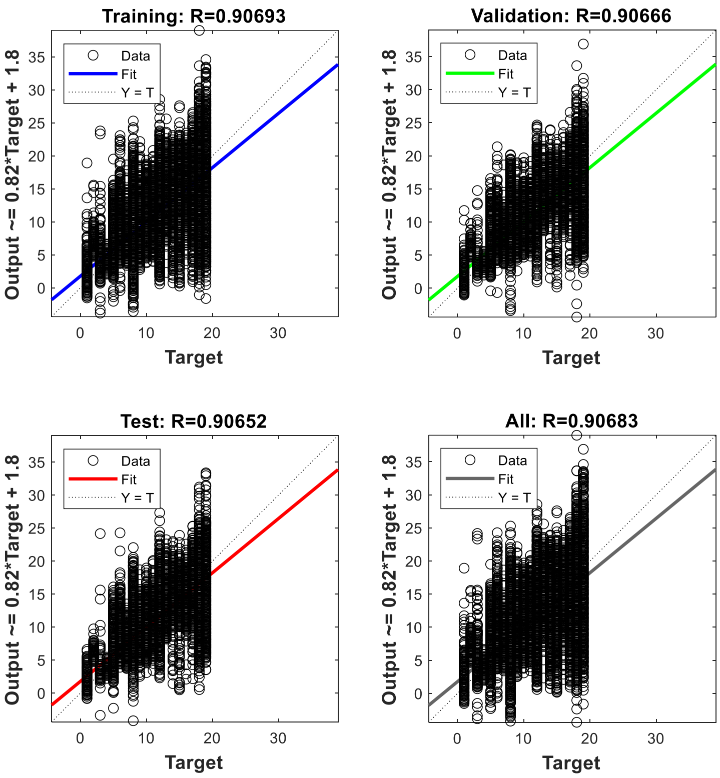

Figure 7: Regression Fit Graph

The results of the mean square error MSE and R-value show that the network has high accuracy in classifying human activities, which reflects the efficiency of the network model.

Parameter Index Chart

	Samples	MSE	R
Training	798000	4.66	0.92
Validation	171000	0.00	0.00
Testing	171000	4.66	0.92

Figure 8: Parameter Index Chart

View the error histogram for additional verification of network performance.

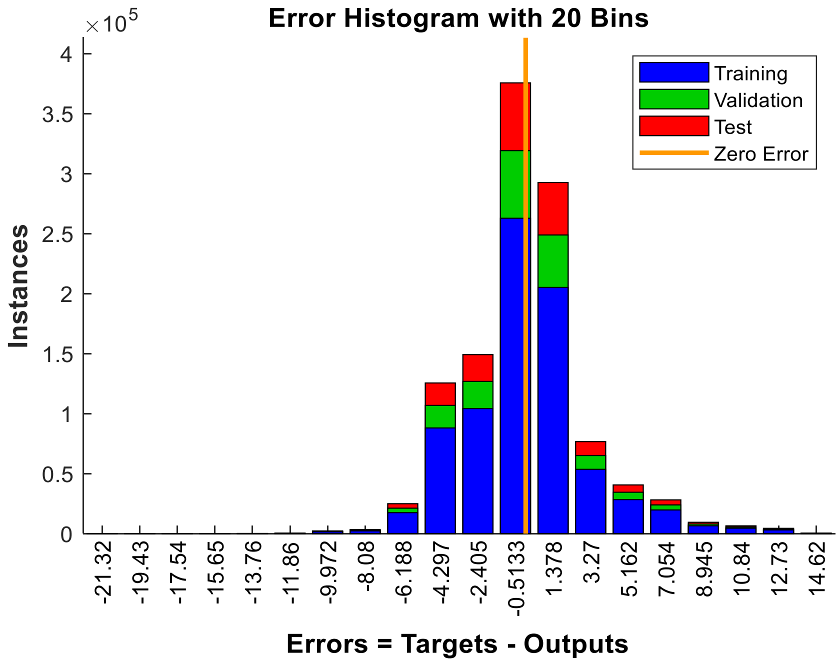

Figure 9: Error Histogram

The blue bars indicate training data, the green bars indicate validation data, and the red bars indicate test data. The histogram can indicate outliers, which are data points that fit significantly worse than the majority of the data. The outliers are examined to determine if the data are poor within the error range or if any of these data points differ from the rest of the data set, and the data points with valid outliers are interpolated using the network.

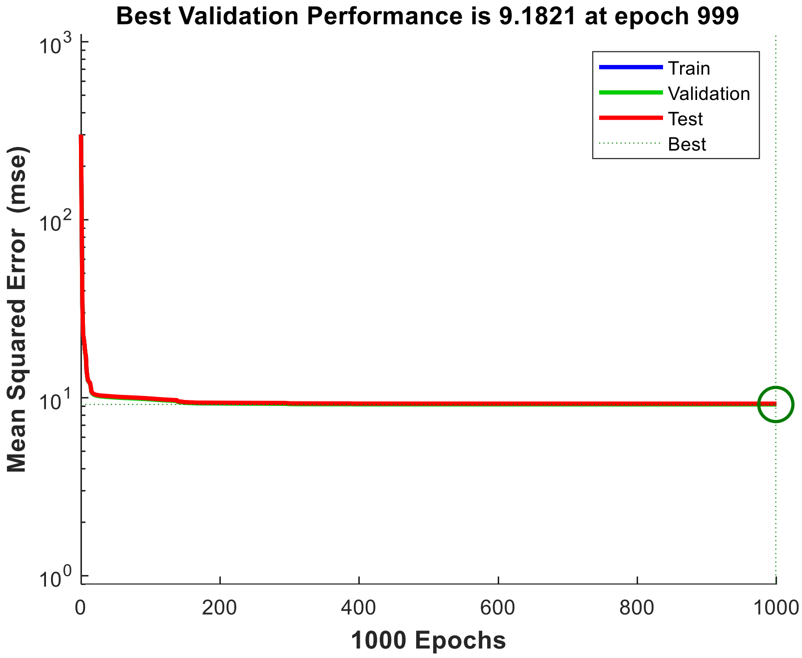

Figure 10: Optimal state chart

The conditional stop training was reached at 999 Epochs and the best performance was reached at 999 with a mean square error value of 9.1821.

4.3 Model III: Generalization Ability Evaluation

4.3.1 Analysis of the Problem 2

For problem 2, we need to maintain good generalization ability of the model on a limited dataset, for this question we used the leave-out method and metrics such as accuracy, precision, recall, F1 score, and confusion matrix to perform the evaluation of the generalization ability of our established BP neural network.

4.3.2 The Foundation of Model

1. Error rate and precision

Error rate is the ratio of the number of misclassified samples to the total number of samples, while precision is the ratio of the number of correctly classified samples to the total number of samples.

2. Accuracy rate, accuracy rate and F1

1) Precision

Precision is the probability that the real positive cases account for the predicted positive cases.

2)Recall

Recall is the probability that the true positive cases account for all positive cases.

In general, accuracy and recall are contradictory measures. For example, if the rate of all is high, i.e., if we want to select all the positive cases as much as possible, it will increase the probability of predicting positive cases and thus decrease the accuracy.

3) PR curve

The samples are ranked according to the prediction effect of the learner, and the first ones are the ones that the learner thinks are "most likely" to be positive, and the last ones are the ones that the learner thinks are "least likely" to be positive. In this order, the threshold is lowered and the samples are predicted as positive examples one by one, and the current detection rate and accuracy rate can be calculated each time. Using the accuracy rate as the vertical axis and the completeness rate as the horizontal axis, we can get the accuracy-completeness curve.

If the PR curve of a learner is completely "wrapped" by the curve of another learner, the latter has better performance than the former. This is because the encapsulated P-R curve represents a lower rate of full search and accuracy, i.e., the learner's performance is not good.

3. ROC and AUC

The full name of ROC is "subject operating characteristic". Unlike the P-R curve, which uses the accuracy and completeness as the vertical and horizontal axes, the vertical axis of the ROC curve is the True Positive Rate (TPR) and the horizontal axis is the False Positive Rate (FPR).

TPR indicates the probability that a positive case is predicted to be a positive case

FPR indicates the probability that a negative case is predicted to be a positive case.

4. Cost-sensitive error rate and cost curve

The consequences of different types of errors are different, and in order to weigh the different losses caused by different errors, we can assign a "non-equal cost" to the errors.

Each point on the ROC curve corresponds to a line segment on the cost plane, and the area under the line segment represents the expected overall cost under that condition. Thus, each point on the ROC curve is transformed into a line segment on the cost plane, and the area enclosed by the lower bound of all the line segments is the expected overall cost of the learner under all conditions

4.3.3 Algorithms used in the model

1. Hold-out Method

When there is sufficient data, the data set D is directly divided into two mutually exclusive sets, one for training and the other for testing. After training the model on the training set S, the model is tested on the test set T. The test error is used as an estimate of the generalization error of the model.

$$D = S \cup T, S \cap T = \emptyset (18)$$

2.Performance Metrics

1) Accuracy

Accuracy is the ratio of the number of correctly predicted samples (TP and TN) to the number of all samples. Generally speaking, the higher the accuracy, the better the classifier is, and the formula is as follows. (19)

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

2) Precision

Precision is the ratio of the number of samples correctly predicted to be positive (TP) to the number of all predicted to be positive (TP and FP) and is calculated as follows. (20)

$$Precison = \frac{TP}{TP + FP} \quad (20)$$

3) Recall

Recall is the ratio of the number of samples correctly predicted to be positive (TP) to the total number actually positive (TP and TN) and is calculated as follows. (21)

$$Recall = \frac{TP}{TP + FN} \quad (21)$$

4) F1 Score

F1 Score is the arithmetic mean divided by the geometric mean, and the larger the better, calculated by the following formula. (22)

$$\frac{2}{F1}=\frac{1}{Precision}+\frac{1}{Recall} \quad (22)$$

The conversion gives (where m is the total number of samples) (23)

$$F1=\frac{2PR}{P+R}=\frac{2TP}{2TP+FP+FN}=\frac{2TP}{m+TP-TN} \quad (23)$$

In summary: Precision and recall are contradictory metrics. In general, when precision is high, recall tends to be low, and when recall is high, precision tends to be low.

4.3.4 The Solution and Result

The generalization ability of the model was evaluated by MATLAB programming. Functions such as ErrorAndAccuracy, F1, PrecisionAndRecall, MSE were written to evaluate the generalization ability of the model on each parameter by taking appropriate samples (100,000, 500,000, 10,000,000) from the dataset and comparing the predicted type data with the real type of the test data through BP neural network simulation.

Table 5: Model evaluation parameters(I)

	precision	**recall**	**f1-score**	**support**
0	0.83	0.65	0.73	100
1	0.73	0.77	0.75	100
2	0.68	0.65	0.66	100
accuracy				0.801

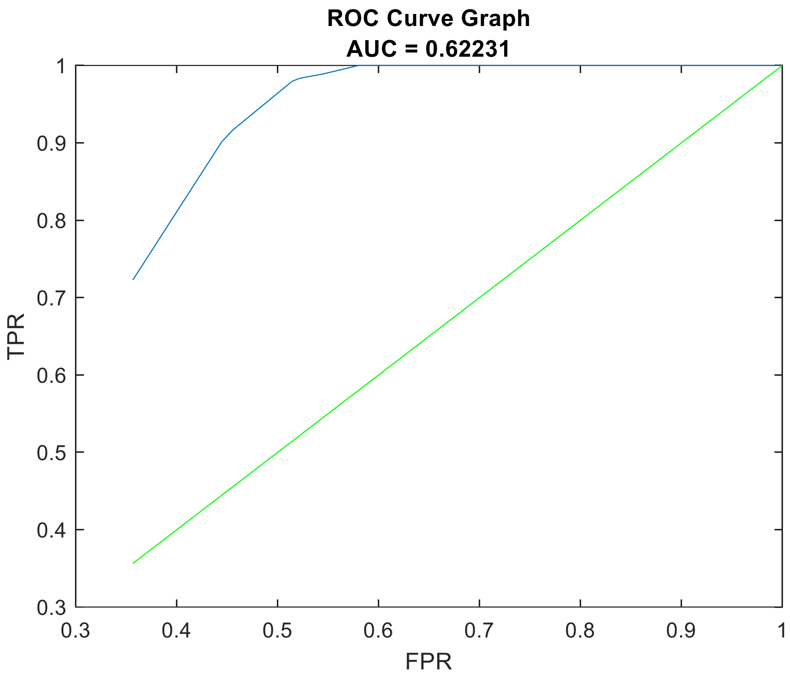

Figure 11: ROC Curve Graph

The neural network classification results are compared with the test samples and the following visualization results are obtained by plotting.

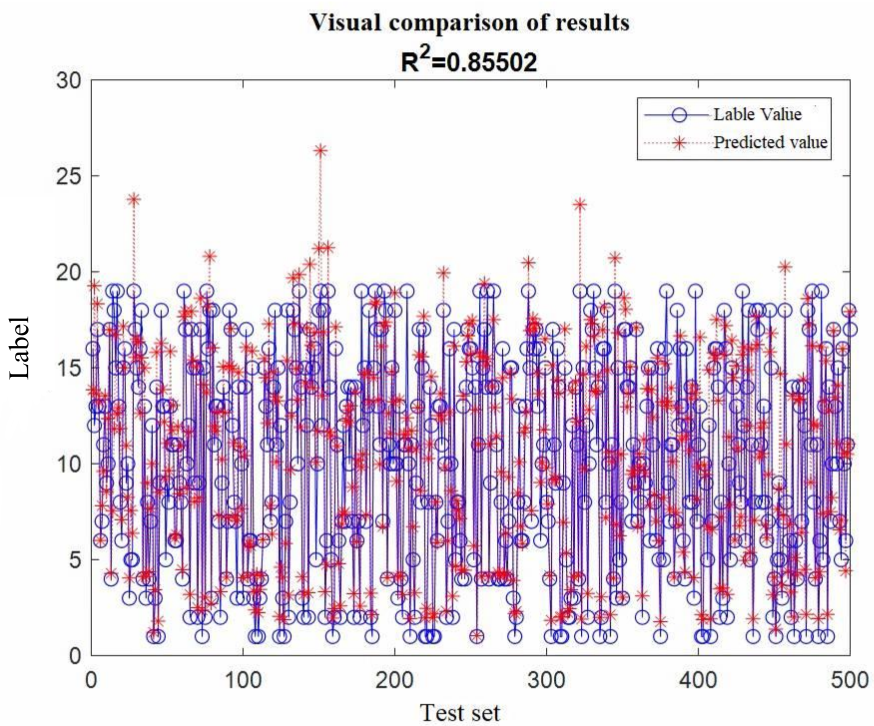

Figure 12: Visualization Comparison(I)

4.3.5 Analysis of the Result

From the analysis of the data in the above table, it can be seen that for different data sets, the accuracy of the BP network for human action recognition results fluctuates more, while recall and f1-score are more stable. It can be seen from the comparative results of viewable views and the regression fit lines that the network has good performance with high classification accuracy and precision. Therefore, the model has certain generalization ability and there is also room for improvement.

4.4 Model IV: Overfitting Problem Optimization

4.4.1 Analysis of Problem 3

The regular methods to solve overfitting include reducing eigenvalue, implementing regularization and so on. In the first problem, we have adopted Factor Analysis to reduce the dimensions of the given data in order to extract the characteristics of the them, which has already played a role in solving overfitting problem. In this question, we will try to overcome the overfitting problem by using regularization and dropout discarding .

4.4.2 The Foundation of Model

1.Regularization

We choose L2 Regularization to process our parameters. We add a special term to the original cost function, and we will get a new cost function, which is

$$C = C_0 + \frac{\lambda}{2n}\sum_w w^2.$$

Then we calculate the derivative of the parameters, and we get

$$\frac{\partial C}{\partial w} = \frac{\partial C_0}{\partial w} + \frac{\lambda}{n} w$$

Using this derivative, we have the new update formula for parameters

$$w := w - \alpha \frac{\partial C_0}{\partial w} - \alpha \frac{\lambda}{n} w = (1 - \frac{\alpha\lambda}{n}) w - \alpha \frac{\partial C_0}{\partial w}$$

in which α is the learning rate, and via observing the new update formula we can find that the values of w will gradually decrease in the training, so we finally can simplify our neural network and reduce the probability of overfitting phenomenon.

2.Dropout

When our neural network is in Forward Propagation, we will let some of the neurons stop working with a certain probability p, which can allow the model get more stronger generalization ability, because it does not rely too much on some local characteristics. The neurons lost in each iteration are different, resulting in obtaining different models after their training.

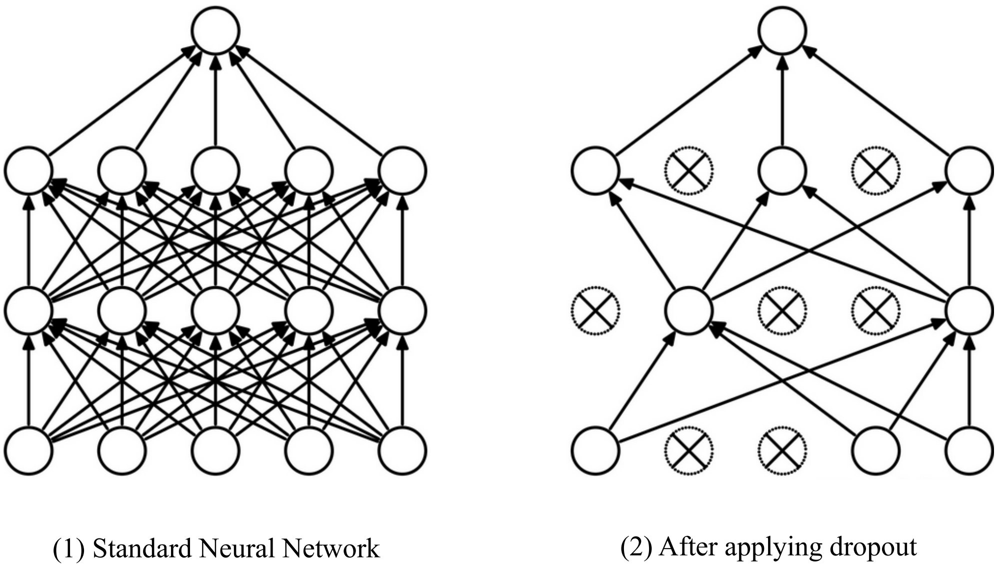

Figure 13: The demo of dropout

4.4.3 Result Presentation

After using the methods above and training the neural network again, we finally get a new result .Analyzing from the new result, we can find that the all indicators have been optimized and according to the Model III we can conclude that the generalization ability has been improved, which is contributing to overcome the overfitting problems.

Model evaluation parameters(II)

	precision	recall	f1-score	support
0	0.85	0.74	0.79	100
1	0.78	0.79	0.78	100
2	0.75	0.73	0.74	100
accuracy				0.847

Figure 14: Visualization Comparison(II)

V. Model Evaluation and Further Discussion

5.1 Strength

Our model has proved that it has a strong generalization ability. Moreover, we have drawn some useful conclusions about our model.

1. Based on LM algorithm, our network has a very fast and effective learning speed, specifically it can complete the learning of millions of data in a short time.
2. With the help of factor analysis to extract data features, we eliminate the correlation between data as much as possible, and greatly improve the performance of our model.
3. The factor features extracted by factor analysis can have practical significance. Under certain conditions, they can be given specific names and meanings to facilitate our understanding of data features

5.2 Weakness

This model just applies to classify the few given human activities. As we have stated, our model has only 19 output cases and these 19 behaviors have obvious differences in characteristics. However, in real life, human behavior is diverse and countless, if we need to classify more human activities, the generalization of our model may decrease, which requires us to improve the model, so that our model can maintain good generalization ability. That’s just what we should do in the improved model.

5.3 Further Discussion

In addition to the what mentioned above, it is also one of our important tasks to think how to transplant the model to some specific situations. This model can be used in ambulatory monitoring, home-based rehabilitation, and so on. This requires us to acquire the necessary data sets in these specific situations and redesign the output classified content to meet the need.

VI. References

[1] Zhang Liping, Kuang Zhenwu, Li Kunjian, et al . Acceleration sensor and neural network based on human activity behavior recognition based on acceleration sensors and neural networks [J]. Modern Electronic Technology, 2019, 42(16):71-74.

[2] LEUTHEUSERH, SCHULDHAUS D, ESKOFIER B M.Hierarchical, multi-sensor based classification of daily life activities:Comparison with state-of-the-art algorithms using a benchmark dataset [J]. Plos one, 2013, 8(10):75196.

[3]Dong Bo Chen Aierui Zhang Ming The role of machine learning in solving the phenomenon of overfitting Psychological Science Journal of Psychological Science 2021, 44(2):274-281

[4] MARKATOUM, TIAN Hong, BISWAS S,et al. Analysis of variance of cross-validation estimators of the generalization error[J].Journal of Machine Learning Research,2005,6: 1127-1168

[5] CHVALOVSKÝ K, JAKUBŮV J, SUDA M, et al. ENIGMA-NG:Efficient Neural and Gradient-boosted Inference Guidance for E[C]//Springer. International Conference on Automated Deduction, August 27-30, 2019, Natal, Brazil. Cham: Springer, 2019:197-215.

VII. Appendix

7.1 Codes of files process

```python
import csv

def get_num(i):
    if i < 10:
        return '0' + str(i)
    else:
        return str(i)

out = open('data.csv', 'w', newline='', encoding='gbk')
csv_writer = csv.writer(out, dialect='excel')

ll = ['T_xacc', 'T_yacc', 'T_zacc', 'T_xgyro', 'T_ygyro', 'T_zgyro', 'T_xmag', 'T_ymag', 'T_zmag',
      'RA_xacc', 'RA_yacc', 'RA_zacc', 'RA_xgyro', 'RA_ygyro', 'RA_zgyro', 'RA_xmag', 'RA_ymag', 'RA_zmag',
      'LA_xacc', 'LA_yacc', 'LA_zacc', 'LA_xgyro', 'LA_ygyro', 'LA_zgyro', 'LA_xmag', 'LA_ymag', 'LA_zmag',
      'RL_xacc', 'RL_yacc', 'RL_zacc', 'RL_xgyro', 'RL_ygyro', 'RL_zgyro', 'RL_xmag', 'RL_ymag', 'RL_zmag',
      'LL_xacc', 'LL_yacc', 'LL_zacc', 'LL_xgyro', 'LL_ygyro', 'LL_zgyro', 'LL_xmag', 'LL_ymag', 'LL_zmag','subject'
      ]

csv_writer.writerow(ll)

for i in range(1, 20):  # 20 9 61
    for j in range(1, 9):
        for k in range(1,61):
            path = "./data/a" + get_num(i) + '/p' + str(j) + '/s' + get_num(k) + '.txt'
            file = open(path, 'r', encoding='utf-8')

            for line in file.readlines():
                l = line.strip().split(',')
                l.append(str(i))
                csv_writer.writerow(l)
            file.close()
```

7.2 Codes of neural network

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Auto-generated by MATLAB, 04-Dec-2022 14:52:18.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = Qx24 matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = Qx1 matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = data.xs;
x1_step1.gain = data.gi;
x1_step1.ymin = -1;

% Layer 1
b1 = data.b;
IW1_1 = data.iw;
% Layer 2
b2 = 0.74436015070712513442;
LW2_1 = data.lw;
% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 0.111111111111111;
y1_step1.xoffset = 1;

% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
```

```
    X = {X};
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},1); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    X{1,ts} = X{1,ts}';
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
    Y{1,ts} = Y{1,ts}';
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS ========
```

```
% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```