Deep Learning in Gait Recognition

Keywords: CNN, Gait recognition, Inertial data

By Xiangwu Li.

April 30th, 2018

I. Overview

1. Introduction

As a new biometric identification technology, gait recognition has broad market prospects in the fields of security, medicine, and public transportation due to its non-contact and difficultly-disguised features. Current gait recognition methods mainly focus on gait images, pressure data, and inertial data. Given that inertial data recognition is free from site and vision, gait recognition based on inertial data is considered as the research object in this project.

In the research, I collected tri-axial acceleration and angular velocity and processed tri-axial data through filtering, smoothing, normalization and feature extraction; then constructed multi-channel convolutional neural networks (CNNs) for classification.

Experiments tested the influence of processing methods, different splicing ways, various channel models on the classification ability, and the accuracy of the optimal model on out-of-sample test reached **93.4%**, indicating that CNN model can well identify gait characteristics based on inertial signals. (code and data of this project can be found at https://github.com/Luffy-wu/Gait-recognition .)

2. Model Description

A convolutional neural network (CNN) is a class of feed-forward artificial neural networks, consisting of an input layer, convolutional layers, pooling layers, fully connected layers, normalization layers and an output layer. It is like a batch filter and continuously rolls input information, from layers to layers, from low-level features to advanced features, from the local to the whole, and finally realizes high-dimensional information recognition.

Structure	Description
Convolutional Layer	The convolutional layer is calculated by using a convolution kernel to scan the previous input layer. The sliding window is the receptive field whose role is to determine the scanned size of the former layer. The purpose of convolution is to extract features of the input, and as the neural network deepens, the features extracted by the convolutional layer will develop from the simple to the complex, from the lower to the higher.
Activation Unit	The activation unit is often applied to introduce nonlinear features into the

	information domains. Commonly used functions include relu, sigmoid, tanh, softplus, etc.
Pooling Layer	The pooling layer divides the information field into a number of rectangular sub-areas, and compresses the dimensions by extracting features of the sub-areas. The sampling methods include maximum pooling, average pooling, and second paradigm pooling, etc.
Fully connected layer	The fully connected layer connects every neuron in one layer to every neuron in another layer, synthesizing the features extracted from all the front layers.

II. Data Exploration

1. Data Collection

I drew on the accelerometer and gyroscope of Gait.apk to collect tri-axial acceleration and angular velocity signals.

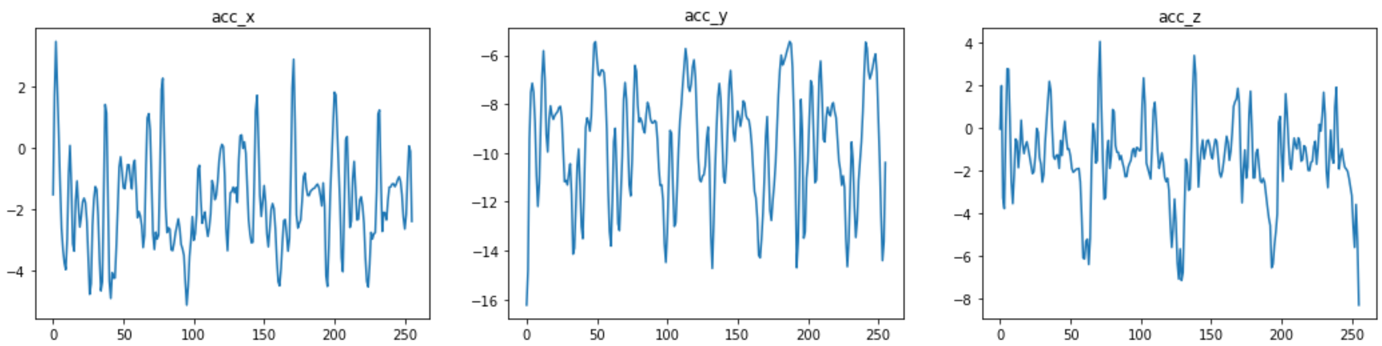
a. Data Objects

I collected walking data about 80 students from Wuhan University, including 40 boys and 40 girls.

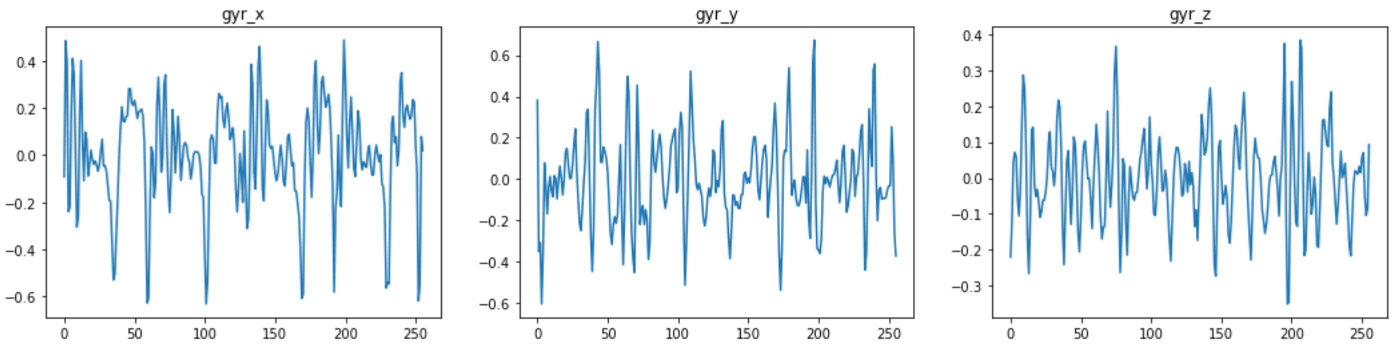
b. Data Dimension

The X axis of acceleration indicates the right side of the body, the Y axis indicates the direction of gravity, and the Z axis indicates the direction in which the experimenter walks. The X-axis, Y-axis, and Z-axis of the angular velocity represent pitch angular velocity, azimuthal angular velocity, and oblique angular velocity, respectively. The tri-axial acceleration can capture the speed change of the experimenter, and the tri-axial angular velocity can well record the posture change of the experimenter while walking.

Unit Signal for tri-axial Acceleration



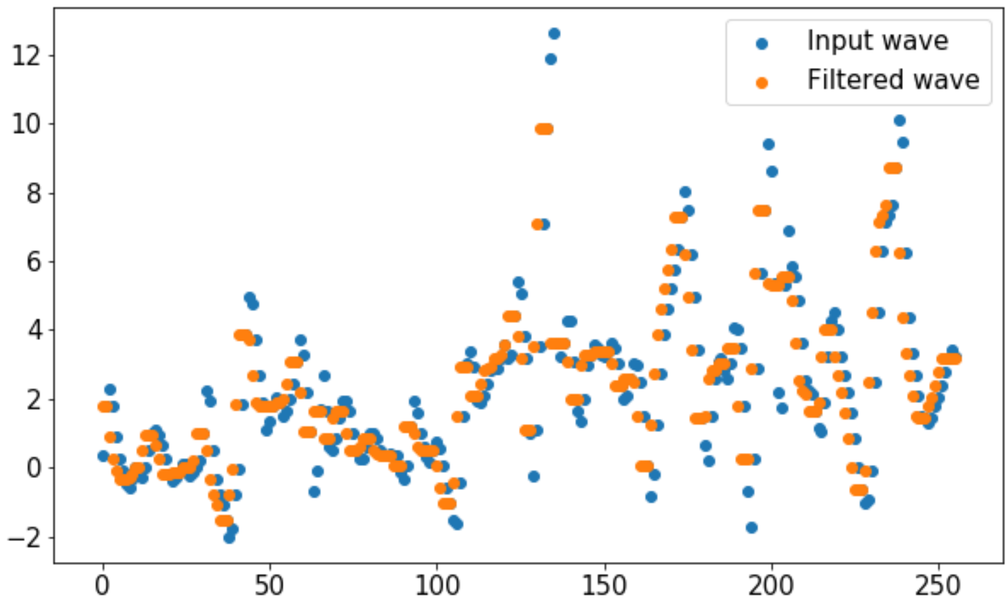
Unit Signal for tri-axial Angular Velocity



2. Data Processing

- Removed unstable signals at the beginning and end of walking
- Segmented and normalized data
- Filtered data noise using median filters
- Generated one-hot labels, randomly shuffled it and divided into training & test set

Input Wave and Filtered Wave



III. Experiment and Result Analysis

1. Architecture

a. Data preparation

I used 1-80 to represent 80 experimenters and transformed numbers into one-hot vectors (shown below). One-Hot encoding mapped discrete features to Euclidean spaces, allowing the output channels directly correspond to experimenter categories.

One-Hot Encoding

0---[1, 0, 0, 0, …, 0]

1---[0, 1, 0, 0, …, 0]

2---[0, 0, 1, 0, …, 0]

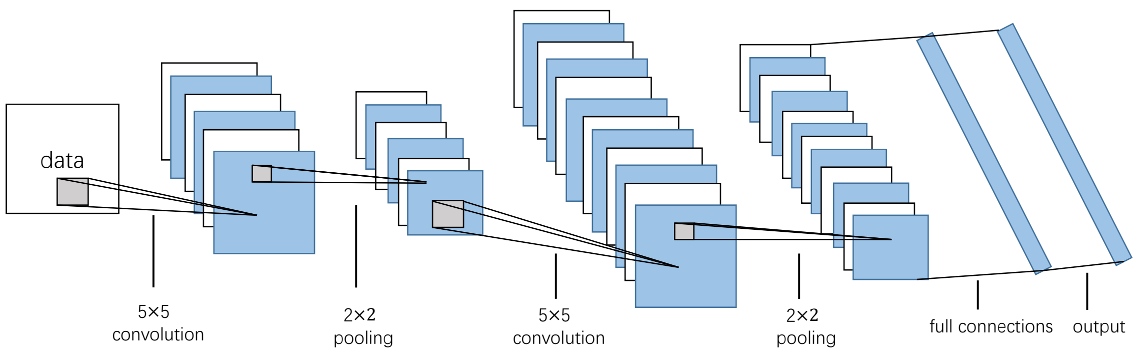
3---[0, 0, 0, 1, …, 0]

…

80---[0, 0, 0, 0, …, 1]

b. Network layer Settings

The constructed network structure was as follows:



The convolutional layer set the receptive field to 5×5, the first convolution kernel was set to 32, and the second layer was 64; the activation unit used the relu function to introduce nonlinear features into the model; the pooling layer adopted 2×2 maximum pooling, and the output dimension of the full output layer was set to 80, corresponding to the experimenters.

$$ReLU(x) = \begin{cases} x & if\ x > 0 \\ 0 & if\ x \leq 0 \end{cases}$$

c. Training Settings

In order to speed up the training process, I applied truncated normal distribution function to assign initial values; to avoid overfitting, a dropout function was introduced to reduce the interaction between feature detecting layers and avoid overtraining of the training set.

d. Channel Settings

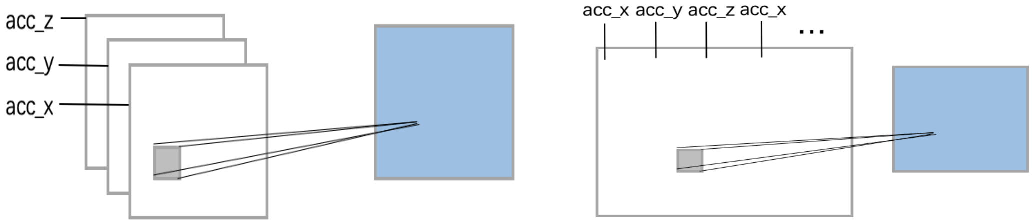
For each experimenter, tri-axial acceleration and angular velocity were obtained; for acceleration, I denoted acc_x, acc_y, acc_z to represent the three axles of acceleration, with gyr_x, gyr_y, gyr_z for angular velocity; then I made the following three settings:

One-channel model. Only the experimenter's acc_x data was taken as input data.

Three-channel model. The acc_x, acc_y, and acc_z data were taken as input data; then, the rearrangement was performed in two ways: first, the acc_x, acc_y, and acc_z were directly integrated horizontally to check whether the time-series organization of acc_x, acc_y, and acc_z could provide more effective information for the model; the second was to crosswise arrange acc_x , acc_y, acc_z to investigate whether the parallel organization of acc_x, acc_y, acc_z could improve the classification accuracy.

Six-channel model. The acc_x, acc_y, acc_z, gyr_x, gyr_y, and gyr_z were all taken and transformed as input for model.

Horizontal Splicing & Crosswise Splicing



2. Experiment Result Analysis

a. Processing Analysis

The experiment found that when the raw data and the processed data were trained as input sets respectively, the accuracy on the test set showed a significant difference. For raw data input, the accuracy was only 5.54%, a total failure. On the contrary, the data after standardization and filtering can well eliminate outliers and noise, and significantly improve data quality.

Processing Analysis	raw data	processed data
accuracy	0.0554273	0.84642

b. Splicing Methods Analysis

Through comparing two splicing methods, it found that crosswise splicing acc_x, acc_y, and acc_z yielded a better accuracy of 93.07%, which suggested that the parallel organization method, analogous to the RGB properties of colored images, had an advantage than the time-series one.

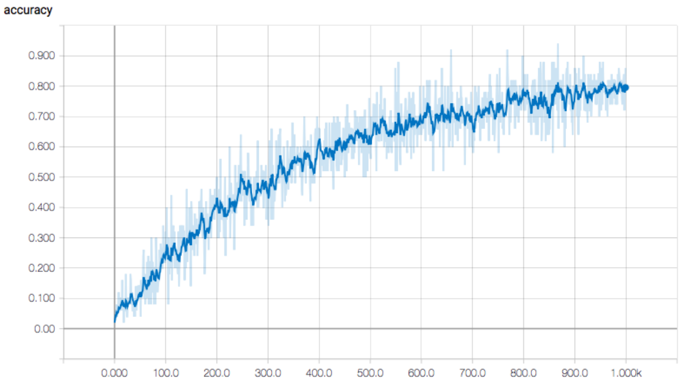
Splicing Methods Analysis	horizontal splicing	crosswise splicing
accuracy	0.897229	0.930716

c. Channel Number Analysis

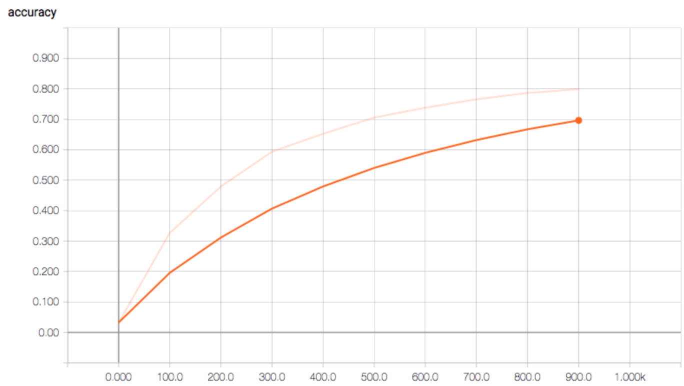
I set the starting channels of input layers to 1, 3, and 6, respectively. The experiment found that three channels could significantly improve the recognition effect compared to the one channel, indicating that tri-axial acceleration data could better reflect the gait characteristics. But the accuracy of six channels was only slightly higher than three channels.

Channel Number Analysis	one-channel	three-channel	six-channel
accuracy	0.84642	0.930716	0.93418

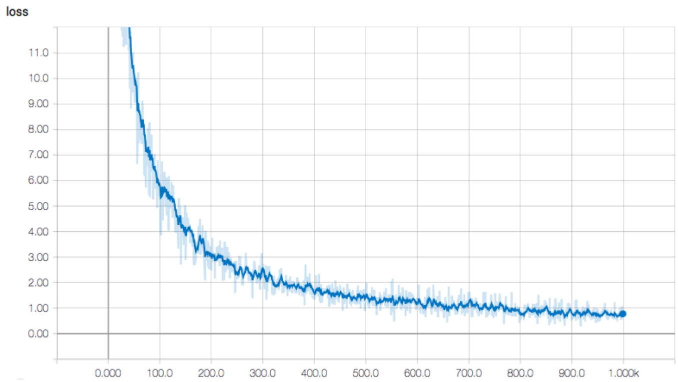
Accuracy Convergence Curve in Training Set



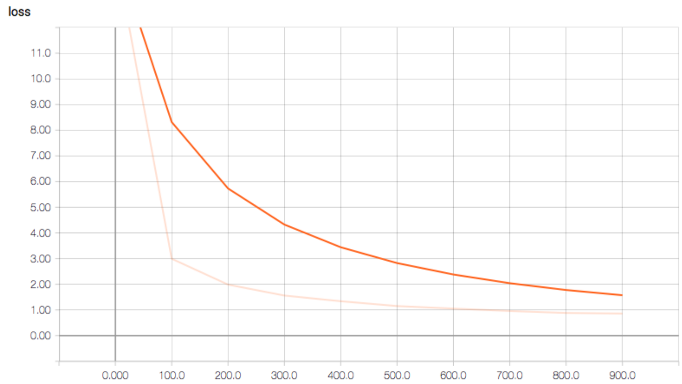
Accuracy Convergence Curve in Test Set



Loss Convergence Curve in Training Set



Loss Convergence Curve in Test Set



IV. Conclusion

In this project, I collected tri-axial acceleration and angular velocity signals about 80 students, then clarified the way of data processing, constructed the corresponding convolutional neural network (CNN) models.

Research experiments found that the accuracy rate of the optimal model on the test set reached 93%, indicating that the CNN model could well identify the gait characteristics based on inertial data. Meanwhile, it found that compared with unstandardized raw data, standardized data could significantly improve the recognition accuracy by excluding noise and outliers. In addition, the crosswise splicing of X, Y, and Z axes of acceleration or angular velocity could achieve effective fusion of features in three directions, and further enhance classification. Through comparison among one-channel, three-channel, and six-channel models, it was found that the three channels could significantly improve the accuracy, but the further promotion effect of the six-channel was minimal.