Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/107/20241200

Mobile Phone Intelligent Recognition of Human Activity Based on Deep Neural Network

Zihuan Zhao

Nanjing University of Information Science and Technology, CN

202383710037@nuist.edu.cn

Abstract. Because of the rapid development of smartphone sensor technology and the continuous progress of machine learning algorithms, it has become possible to use smartphones for human activity recognition. Sensors such as accelerometers, gyroscopes, and magnetometers built into smartphones are able to collect human motion data in different activities, which provides a rich data source for activity recognition through mobile phones. In this paper, we change the activation function of the hidden layer of the neural network to observe the effect of this variable on the mobile phone to recognize human activities. Taking Rectified Linear Unit (ReLU)function and Sigmoid function as examples, it is found that the neural network using the ReLU function shows higher accuracy (0. 9518 and 0. 9332) and lower loss value in the early stage and middle to late stage of training. In addition, This paper finds that training for 100 epochs takes significantly less time than the model with the Sigmoid activation function (47. 041 seconds vs 179. 022 seconds). The neural network system built by the relu function has faster operation speed and better performance. This study shows that different hidden layer functions have great influence on the quality of the same neural network, and selecting the best hidden layer function will be an important point for the success of the study.

Keywords: Machine learning, Human activity, recognition phone.

1. Introduction

Modern smartphones are commonly equipped with a variety of sensors that can collect detailed information about the user's movement and the environment. Accessibility of data: The portability of smartphones enables them to continuously collect data on users' daily activities, providing a rich data source for research [1,2]. Development of machine learning techniques: With the advancement of machine learning techniques, researchers have begun to explore the application of these techniques to human activity recognition to improve the accuracy and robustness of recognition. Increasing demand for applications: The growing demand for health, safety, and convenience along with the improvement of living standards is driving the need for intelligent monitoring systems that can provide customized services by analyzing user behaviour [3].

The current research focus for smartphone sensors lies in the integration and advancement of deep learning techniques for human behaviour recognition, as well as the combination of various neural network types. Lei Miao et al. explored the use of deep learning architectures, including Convolutional Neural Networks (CNN) to enhance the precision and efficiency of action recognition. Haitao Wu et al. applied CNN for feature extraction and combined it with RNN or LSTM to handle time-series data, thereby improving the performance of human activity recognition. From this research, it is evident that deep learning, alongside different neural network models, offers significant potential and advanced capabilities in the field of human activity recognition, making it a promising area for further investigation [4,5].

This study focuses on examining the differences between neural network hidden layer activation functions and assessing the overall system performance. Specifically, the research aims to compare human activity recognition using two distinct hidden layer activation functions, the ReLU activation function and the sigmoid activation function, to analyze their impact on network performance.

2. Data and methods

2.1. Data

The dataset used in this paper is the human activity recognition data set, which comes from: https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones

The Human Activity Recognition (HAR) database was created using data collected from 30 participants. Each individual performed various daily activities while carrying a smartphone positioned on their waist, which was equipped with inertial sensors to capture movement data. These sensors recorded the subjects' motions, enabling the development of the database for analyzing patterns in human activity. The dataset contains 10299 instances and 561 feature variables (no missing values).

2.2. Method

Deep Neural Networks (DNNs) consist of multiple hidden layers, allowing them to detect complex patterns and extract features from data by mimicking the way the human brain interprets information. DNNs are widely applied in various domains such as image and speech recognition, natural language understanding, medical diagnostics, and more [6].

Neurons are the basic units of DNNS and are similar to neurons in the human brain. Each neuron receives inputs, performs a weighted sum, and passes a nonlinear activation function to generate an output.

DNNS consist of multiple layers, including the input layer, hidden layer and output layer. The input layer receives the raw data, the hidden layer performs feature extraction and transformation, and the output layer generates the final prediction results. The weights refer to the parameters that connect the neurons and determine how much the input signal influences the neuron output. During training, the weights are constantly updated to minimize the prediction error [7,8].

Bias adds a constant offset to the output of a neuron and helps the model fit the data better.

Activation functions are used to introduce nonlinearities into neurons, allowing the network to learn and simulate complex functional mappings. Common activation functions include Rectified Linear Unit (ReLU), Sigmoid, and Tanh [9,10].

A loss function quantifies the disparity between a model's predicted outcome and the actual value. Examples of commonly used loss functions include Mean Squared Error (MSE) and Cross-Entropy. Backpropagation is a training algorithm that calculates the gradient of the loss function for the network’s parameters, which is then used to adjust the network’s weights. An optimizer is responsible for updating these weights during training. Popular optimizers include Stochastic Gradient Descent (SGD), Adam, and RMSprop..

Regularization is a technique used to prevent model overfitting, including L1 and L2 regularization, Dropout, etc. Convolutional layers are used when processing image data to extract local features of the image.

Pooling layers are used to reduce the spatial dimension of features, thus reducing the number of parameters and computational complexity.

The fully connected layer is at the end of the network and is used to map the extracted features to the final output.

The following is a detailed description of the two activation functions used in this experiment

DOI: 10.54254/2755-2721/107/20241200

- The ReLU function, also known as the Rectified Linear Unit, is one of the commonly used activation functions in deep learning. Its mathematical expression is as follows:

$$\mathrm{ReLU(x) = max(0, x)ReLU(x) = max(0, x)} \tag{1}$$

- The feature of this function is that when the input xx is greater than 0, the output is xx itself; when the input xx is less than or equal to 0, the output is 0. The derivative expression of the ReLU function is as follows:
- When x>0, the derivative of ddxReLU(x)=1dxd ReLU(x)=1
- When x≤0, the derivative of ddxReLU(x)=0dxd ReLU(x)=0

The Sigmoid function, also known as the logistic function or hyperbolic tangent function, is a mathematical function widely used in machine learning and artificial intelligence fields. The mathematical expression of the Sigmoid function is as follows:

$$\mathrm{\sigma(x) \quad = \quad 11 + e - x\sigma(x) \quad = \quad 1 + e - x1} \tag{2}$$

Where e is the base of the natural logarithm, approximately equal to 2. 71828, and xx is the input value.

The graph of the Sigmoid function is an S-shaped curve, and its output value range is between 0 and 1, which makes it particularly useful in binary classification problems because it can be interpreted as a probability value. The derivative expression of the Sigmoid function is as follows:

$$\mathrm{ddx\sigma(x)=\sigma(x)(1-\sigma(x))dxd\sigma(x) = \sigma(x)(1-\sigma(x))} \tag{3}$$

2.3. Evaluation Metrics

Model loss refers to the loss of a model during training or validation. The loss value is a measure of the difference between the model prediction and the actual label, the lower the loss, the closer the model's prediction is to the ground truth.

Model accuracy reflects model accuracy represents the proportion of the number of samples correctly predicted by the model to the total number of samples. This is a very important metric as it directly reflects the performance of the model.

The training loss is the loss of the model on the training set. This value usually decreases over time and reflects how well the model fits the training data.

The validation loss is the loss of the model on the validation set. The validation set is the data that the model has not seen during training, and the validation loss is used to evaluate how well the model generalizes to new data.

Training accuracy is the accuracy of the model on the training set. This value usually increases as training progresses, but too high an accuracy may mean that the model overfits the training data.

Validation accuracy is the accuracy of the model on the validation set. This value is a key metric to evaluate the generalization ability of the model, and ideally, we want the model to have high accuracy on both the training and validation sets.

DOI: 10.54254/2755-2721/107/20241200

3. Experimental results and analysis

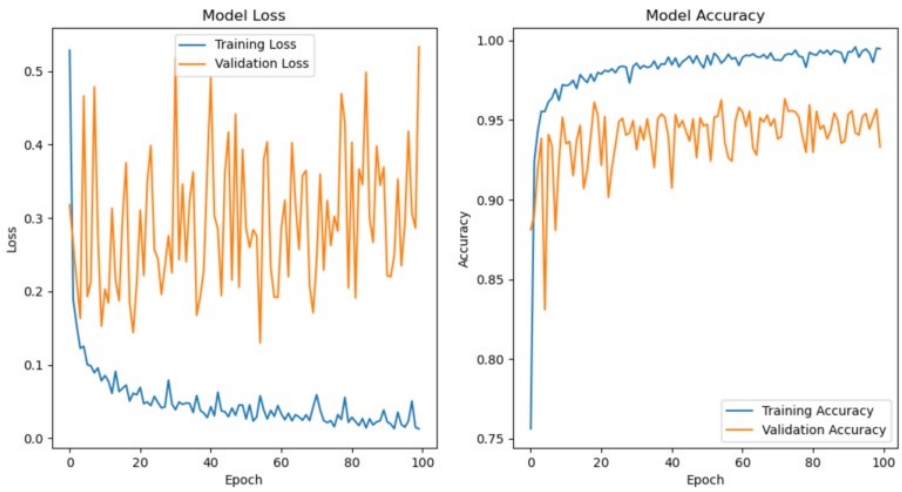

Figure 1. Training plot of relu activation function

Figure 1 shows the experimental results of 100 rounds of training a neural network with relu activation function, showing the change of missing values and the change of accurate values in the training set and the test set, respectively: The loss value of the training set rapidly decreased from 0. 5286 to about 0. 05 in the first 20 training times, and showed a slight downward trend and fluctuates up and down in the next 80 training times. The loss value in the validation set fluctuates greatly. The accuracy value of the training set rapidly increased from 0. 7563 to 0. 9723 in the first 10 training sessions and gradually increased slightly with small fluctuations in the later 90 training sessions. The overall accuracy value of the validation set is increasing but the fluctuation range is large.

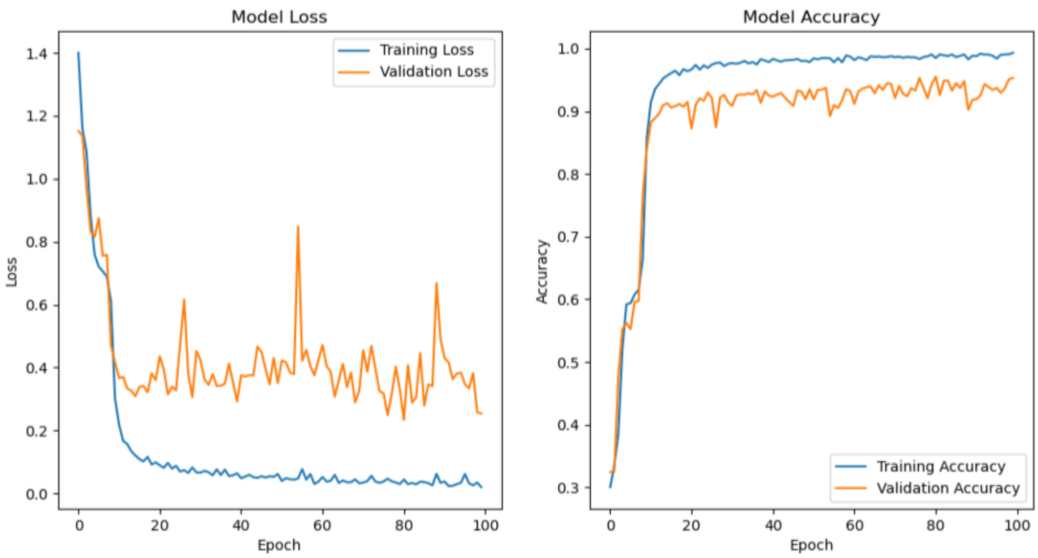

Figure 2. Training plot of sigmoid activation function

Figure 2 shows the results of 100 rounds of training of a neural network with sigmoid activation function, showing the change of missing values and the change of accurate values in the training set and the test set respectively: The loss value of the training set rapidly decreased from 1. 4002 to about 0. 1 in the first 20 training sessions, and showed a slight downward trend and became stable in the next 80 training sessions. In the validation set, the loss value rapidly decreased from 1. 1515 to 0. 4173 after 10 training sessions, but the subsequent training showed large fluctuations. The accuracy value of the training set rapidly increased from 0. 3006 to 0. 8579 in the first 10 training sessions and gradually increased slightly with small fluctuations in the later 90 training sessions. The accuracy values for the validation set are similar to those for the training set but are about 0. 1 lower.

The experimental results in Figure 1 and Figure 2 provide real data support. The following two tables are the comparison of the two models, the comparison of various data from 1 to 10 training times and the comparison of various data from 10 to 100 training times respectively

Table 1. Various metrics of the two functions trained from 1 to 10 times

Hidden layer functions	Epoch	Loss	Accuracy	Val-loss	Val-accuracy
Relu activation function	1~10	0. 5286~0. 0784	0. 7563~0. 9723	0. 3181~0. 1529	0. 8812~0. 9518
Sigmoid activation function	1~10	1. 4002~0. 3019	0. 3006~0. 8579	1. 1515~0. 4173	0. 3247~0. 8395

Table 1 shows the changes in the two functions in various aspects by iterating 10 times. The missing value of relu function in the training set is reduced from 0. 5286 to 0. 0784, and that of sigmoid function is reduced from 1. 4002 to 0. 3019. The accuracy of relu function in the training set increased from 0. 7563 to 0. 9723, and the accuracy of sigmoid function increased from 0. 3006 to 0. 8579. The missing values of the relu function in the validation set were reduced from 0. 3181 to 0. 1529, and the missing values of the sigmoid function were reduced from 1. 1515 to 0. 4173. The accuracy of the relu function on the validation set increased from 0. 8812 to 0. 9518, and the accuracy of the sigmoid function increased from 0. 3247 to 0. 8395.

Table 2. Various metrics for training the two functions for 10 to 100 times

Hidden layer functions	Epoch	Loss	Accuracy	Val-loss	Val-accuracy
Relu activation function	10~100	0. 0784~0. 0129	0. 9723~0. 9948	0. 1529~0. 5331	0. 9518~0. 9332
Sigmoid activation function	10~100	0. 3019~0. 0202	0. 8579~0. 9933	0. 4173~0. 2541	0. 8395~0. 9532

The time required for the hidden layer function to Epoch is 100 times.The Relu activation function 47s41.Sigmoid activation function 179s22.

Table 2 shows the changes in the two functions in various aspects through another 90 iterations. The loss value of ReLU activation function in the training set is further reduced from 0. 0784 to 0. 0129, while the loss value of the Sigmoid activation function is reduced from 0. 3019 to 0. 0202. The accuracy of ReLU function in the training set is improved from 0. 9723 to 0. 9948, and the accuracy of the Sigmoid function is also improved from 0. 8579 to 0. 9933. On the validation set, the loss value of ReLU function increased from 0. 1529 to 0. 5331, while the loss value of the Sigmoid function decreased from 0. 4173 to 0. 2541. The accuracy of the ReLU function in the validation set is reduced from 0. 9518 to 0. 9332, and the accuracy of Sigmoid function is improved from 0. 8395 to 0. 9532.

4. Conclusion

This paper concludes that ReLU activation function usually exhibits faster convergence speed, higher accuracy, and lower loss value during training, which indicates its better performance in this task. The Sigmoid activation function, although improved later in training, still performed worse than ReLU and took longer to train, indicating less efficiency in this task. Considering training efficiency and model performance, ReLU activation function may be a more appropriate choice, especially in scenarios where fast training and high accuracy are required. In the early stage of training (Epoch 1-10), the model with Relu activation function shows a lower loss value (0. 5286~0. 0784) and a higher accuracy (0. 7563~0. 9723). The model with the Sigmoid activation function has higher loss value (1. 4002~0. 3019) and lower accuracy (0. 3006~0. 8579).

In the middle to late training period (Epoch 10 to 100), the model with Relu activation function continued to show lower loss value (0. 0784 to 0. 0129) and higher accuracy (0. 9723 to 0. 9948). Although the loss value of the model with Sigmoid activation function is decreased (0. 3019-0. 0202), the improvement of accuracy is not as significant as that of ReLU (0. 8579-0. 9933).

In terms of the performance of the validation set, the model with ReLU activation function shows higher accuracy (0. 8812-0. 9518 and 0. 9518-0. 9332) and lower loss value in the early stage and the middle to late stage of training. The model with Sigmoid activation function improves the performance on the validation set, but the accuracy and loss are still worse than the ReLU model.

In terms of training time, the model with the Relu activation function took significantly less time than the model with the Sigmoid activation function to train for 100 epochs (47 seconds 41 milliseconds vs. 179 seconds 22 milliseconds).

However, it should be noted that the ReLU activation function may suffer from the gradient saturation problem, that is, when the input is less than 0, the gradient is 0, resulting in some neurons not be updated. Therefore, although ReLU performs better in this study, in practice, the choice of activation function also needs to consider the specific situation and requirements of the model.

References

[1] Geng X Y, Chen W N, Xu Y, et al. 2023 Gait recognition based on intelligent mobile phone research and design. Journal of Modern Information Technology 7(22) 9-14, 19
[2] Qiu T 2021 Human activity recognition based on convolution neural network and the embedded application [Doctoral dissertation, Chongqing University of Posts and Telecommunications]
[3] Zhang Y 2021 Human activity recognition based on triaxial accelerometer research [Doctoral dissertation, Lanzhou Jiaotong University]
[4] Zhang C, Zhao C, Chen C, et al. 2020 Review of human activity recognition based on mobile phone sensors. Computer Science 47(10) 1-8
[5] Ai D, Wang Q, Fan W, et al. 2020 Smartphones sensor human behavior recognition technology. Journal of Xi'an University of Posts and Telecommunications 25(01) 42-48 https://doi.org/10.13682/j.ISSN.2095-6533.2020.01.005
[6] Agrawal K D, Udgata K S, Usaha W 2024 Leveraging smartphone sensor data and machine learning model for human activity recognition and fall classification. Procedia Computer Science 235 1980-1989
[7] Kim Y, Co-Researchers 2018 Activity recognition for a smartphone and web-based human mobility sensing system. Researchers from Nanyang Technological University. Journal of Robotics & Machine Learning 565
[8] Younes A S, Abu A H, Cheol H K 2018 Human activities recognition via smartphones using supervised machine learning classifiers. Primary Health Care Open Access 8(01)
[9] Sarthak G, Ajeet K 2015 Human activity recognition through smartphone's tri-axial accelerometer using time domain wave analysis and machine learning. International Journal of Computer Applications 127(18) 22-26
[10] Sabbir M A, Nova A 2023 A fast and minimal system to identify depression using smartphones: Explainable machine learning-based approach. JMIR Formative Research 7 e28848.