

IMU Action Recognition Based on Machine Learning

Yongzhe Zhang^{1*}, Jiachen Zheng²

¹ The University of Mississippi, MS 38677, United States.

² Yanshan University, Qinhuangdao 066000, Hebei Province, China

Publication date: December, 2019

Publication online: 31 December, 2019

***Corresponding author:** Yongzhe Zhang, daimulin@xyzrgroup.com

1 Introduction

With the advent of the global Internet information era, communication and artificial intelligence have been developing rapidly in information fusion. Accurate motion recognition can provide a new way for human-computer interaction, while artificial intelligence combined with motion recognition can conduct intelligent analysis of the collected information.

This paper adopts IMU motion recognition technology based on machine learning. IMU, inertial measurement unit, is a device that uses accelerometer and gyroscope to measure the three-axis attitude Angle (or angular velocity) and acceleration of an object. In a narrow sense, an IMU is equipped with gyroscope and accelerometer on three orthogonal axes, with a total of 6 degrees of freedom, to measure the angular velocity and acceleration of an object in three-dimensional space, which is known as "6-axis IMU". Broadly speaking, the IMU can add magnetometer to accelerometer and gyroscope to form the "9-axis IMU" which is now known to the public.

Nowadays, IMU is not only used in professional navigation, but also widely used in military defense, aerospace, maritime and other fields, as well as daily WeChat positioning, smart phone, car/train, unmanned driving, smart home and other aspects.

2 Principle of Network Design

The RBF neural network is a radial basis function neural network and a forward neural network. The RBF neural network is a three-layer network structure. The principle of the RBF neural network is to map the input data from the input layer to the hidden layer. The output of the hidden layer is weighted by the connection weight of the RBF neural network, and finally the output layer is finally obtained.

The RBF neural network has three important parameters, which are the center of the hidden layer basis function of the RBF neural network, the width, and the connection weight between the hidden layer and the output layer.

RBF neural network is a kind of feedforward neural network with excellent performance. RBF network can approximate arbitrary nonlinear functions with arbitrary precision, and has global approximation ability. The self-topology is compact, structural parameters can be separated and learned, and convergence speed is fast. The RBF network and fuzzy logic can achieve a good complementarity and improve the learning generalization ability of the neural network.

3 Features of the RBF network

(1) Forward network.

(2) The activation function of the hidden unit is usually a function with a locally accepted domain, that is, the hidden unit makes a meaningful non-zero response only when the input falls into a small

designated area in the input space. Therefore, the RBF network is sometimes referred to as a Localized Receptive Field Network.

(3) The local accepting characteristics of the RBF network make the decision-making implied the concept of distance, that is, the network will respond only when the input is close to the accepted domain of the RBF network.

In the RBF network, the weight of the input layer to the output layer is fixed to 1, and the center and radius of the hidden layer RBF unit are usually predetermined, and only the weight between the hidden layer and the output layer is adjustable. The hidden layer of the RBF network performs a fixed nonlinear transformation, mapping the input space R_n to a new hidden layer space R_h , and the output layer realizes a linear combination in the new space. Obviously, due to the linear characteristics of the output unit, its parameter adjustment is extremely simple, and there is no local minimum problem.

(4) In addition, the research also shows that the influence of the nonlinear activation function form used by the RBF network on network performance is not critical, and the key factor is the selection of the basis function center.

4 Advantages of the RBF network

(1) It has the characteristics that there is no local minimum and the only best approximation problem exists.

(2) The RBF neural network has strong input and output mapping functions, and the theory proves that the RBF network in the forward network is the optimal network for completing the mapping function.

(3) The network connection weight is linear with the output.

(4) Good classification ability.

(5) The learning process converges quickly.

5 Procedure

Step1: Configure usage environment

In this project, we get the result by programming the code for the hardware Arduino 101. Arduino is the basic application program that we need use. What's more, it is necessary to install some auxiliary application, such as development board for Arduino 101, library file Madgwick and Curie PME. Madgwick library is a quaternion gradient descent algorithm, which can be

used to calculate the attitude of rigid body in strapdown inertial navigation system. A simple description of the data processing process is to obtain the detection data of acceleration and gyro, obtain the quaternion through calculation, and convert the quaternion into Euler Angle. Curie PME which have 128 neurons supports the algorithm about Radial Function and k-nearest Neighbors. It gives Curie the ability to learn and categorize like a human being, thereby eliminating some of the tedious programming process.

Step2: Program processing

Preprocessing:

The sample rate for accelerometer is 200, the number of bytes that one neuron can hold is 128. Curie PME is a network of neurons consisting of 128 special storage units. Each storage unit can hold up to 128 bytes of data. Each time the learn function is called, the input new data is written to a neuron in the network. That is, Curie PME can perform 128 learning operations in the state of clearing and resetting. The maximum length of the data vector used for learning is 128 bytes. The number of processed samples can fit inside a neuron, such as 1 sample = accel x, y, z. We can set a function to convert ASCII characters A-Z into decimal values 1-26, and back again.

Setup:

Starting the IMU and PME, and set the Accelerometer Rate as mentioned. Choosing the serial and button pin, then start training letters.

Start the loop:

Record IMU data while button is being held, and convert it to a suitable vector. Then use the PME to classify the vector. The parameter vector is the data to be learned, the parameter vector-length is the data length, and the parameter category is the corresponding classification category. Calling the learn function to tell Curie PME that the data vector belongs to the category. For instance, return a category from 1-26, representing a letter from A-Z.

Well, Simple "moving average" filter, removes low noise and other small anomalies, with the effect of smoothing out the data stream.

We need to compress the stream of raw accelerometer data into 128 bytes, so it will fit into a neuron, while preserving as much of the original pattern as possible. Assuming there will typically be 1-2 seconds worth of accelerometer data at 200Hz, we will need to throw away over 90% of it to meet that goal. This is done in 2

ways:

a. Each sample consists of 3 signed 16-bit values (one each for X, Y and Z). Map each 16-bit value to a range of 0-255 and pack it into a byte, cutting sample size in half.

b. Under sample. If we are sampling at 200Hz and the button is held for 1.2 seconds, then we'll have around 240 samples. Since we know now that each sample, once compressed, will occupy 3 of our neuron's 128 bytes, then we know we can only fit

42 of those 240 samples into a single neuron ($128 / 3 = 42.666$). So, if we take (for example) every 5th sample until we have 42, then we should cover most of the sample window and have some semblance of the original pattern.

c. Then we need input the signal. The signal is up to Arduino 101 where we designed a button. If we do not hold the button, digital receives low level. When the button is being held, digital receives high level. After that, the signal convert to array of three axis.