Data Analysis and Planning Analysis Management System Based on Tensorflow Machine Learning Algorithms

Shuang Hu\textsuperscript{1,2}, Li Huang\textsuperscript{1,2*}, Feng Bai\textsuperscript{1,2} and Jun Zhang\textsuperscript{1,2}

\textsuperscript{1}School of Computer Science and Technology, Wuhan University of Science and Technology
\textsuperscript{2}Intelligent Information Processing and Real-time Industrial System Hubei Key Laboratory Institute

*Corresponding author’s e-mail: Lihuango82@wust.edu.cn

Abstract. At present, machine learning is quite popular, and tensorflow framework is also very popular. This system takes the plan management system as the practice platform, realizes the data analysis function through the machine learning understanding and the key point. Most of the current market are planning management system, but there is no self-analysis ability, need to base on data, there are limitations. Firstly, the system collects training data such as weight, body fat rate and so on, then builds a deep learning neural network, and finally runs the model to realize plan analysis, so as to help users analyze the feasibility of the plan and provide suggestions for users, and the analysis results will be displayed through processing.

1. Introduction

In view of the social background in which the rhythm of life is speeding up day by day, we are busy and diligent with our work and life. Often because of onerous tasks and tremendous pressure and exhaustion. It is very important to plan and manage your own tasks, which is conducive to managing and controlling your time. Through this project, we wish to enable you to deal with your tasks or requirements in a more organized and planned way. Tasks can be quantified so that there are strict criteria for judging them\cite{1}.

The current situation and development trend are that in recent years, with the rapid development of network information technology and cloud computing technology in China, network data is also growing rapidly, and a huge amount of data is generated every day. This phenomenon marks that China has entered the era of big data\cite{2}. In today’s internet era, people cannot live, clothe, food and transportation without the network. Accompanied by this, a variety of network data information has been generated. In the era of vast data, the amount of data is larger and the structure is more complex. Therefore, it is particularly important to extract valuable information from these information. Under the background of big data era, it is necessary to fully excavate the hidden value of data and strengthen the analysis of data. Because the traditional database cannot meet the need of data processing effectively in the era of big data, data analysis should adopt more advanced methods to process data\cite{3}.

The project is innovative and timeliness. In the era of big data, machine learning is used to acquire data, and then data analysis is carried out to draw conclusions. Machine learning is applied to data processing, and then the results are analyzed to get a more reasonable result, which is related to real life. With these data and plans, people can have a very reasonable plan for their daily life and work in a busy modern society. Make life easier\cite{4}.
2. Case analysis
Take the fitness plan as an example: by collecting the information of the body weight, body fat rate, running time and swimming time of the fitness person, and the specific value of the final weight loss, we use the deep neural network to study, and analyze the relationship between each variable and the change of weight loss. Then we can predict the relationship between a person's daily running and weight loss according to the plan time. Swimming, can you let your weight to achieve the desired results in a period of time[5].

3. Training sample collection
Firstly, 100 people's information is collected through the way of network questionnaire survey, which is stored in JSON file for subsequent reading, as shown in figure 1. Specifically include weight, body fat rate, running time, swimming time, specific values of weight loss.

```
{
  "info": {
    "description": "train_data",
    "data_created": "2019-03-04 00:52:50.852455"
  },
  "data": [
    {"weight": 50, "body_fat_rate": 0.5, "running_time": 30, "swimming_time": 30, "lost_weight": 3},
    {"weight": 52, "body_fat_rate": 0.2, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6},
    {"weight": 53, "body_fat_rate": 0.22, "running_time": 15, "swimming_time": 60, "lost_weight": 6}
  ]
}
```

Figure 1. Training sample

4. Building Neural Networks
Neural network is a mathematical model, which can be used to model the complex relationship between input and output. Its training principle is to provide training data for the network, which is predicted by the network. Compared with the correct results, a loss value is obtained. Then the parameters of the neural network are modified by gradient descent and back propagation, so as to reduce the loss value. Finally, a neural network model which can solve the specified problem is obtained. For this problem, mainly through the training of input information, let the neural network to get the relevant parameters, and then predict the unknown things. Specifically, input information such as weight, body fat rate, running time and swimming time every day. Neural network predicts the specific value of weight loss[6].

A basic neural network structure with input layer, hidden layer and output layer, as shown in figure 2.

```
Figure 2. Neural Network
```

Next, we begin to build a neural network:
(1) Definition node receives training data, that is, input layer defines four nodes, which represent the weight, body fat rate, running time and swimming time.

(2) Define the hidden layer, which contains five nodes, to abstract the features of input data to another dimension, to show its more abstract features, so as to construct the non-linear relationship between input and output. Let's take a neuron as an example. If all variables in the neuron graph are represented by symbols, and the output formula is written out, as shown in figure 3 below.

![Figure 3. Neuronal computation](image)

It can be seen that \( z \) is the linear weighted sum of the input and the weight, and the value of a function \( g \) is superimposed. This function \( g \) activation function, when the activation function is a non-linear function, through the deepening of the neural network, can construct a variety of interesting functions. In deep learning, the commonly used activation functions are sigmoid function, tanh function and ReLU function. We use the ReLU function, as shown in formula (1), and the figure is shown in figure 4.

![Figure 4. ReLU function](image)

\[
a^{(3)}_1 = g\left(\theta^{(2)}_0 x^{(2)}_0 + \theta^{(2)}_1 x^{(2)}_1 + \theta^{(2)}_2 x^{(2)}_2\right) g(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{if } z < 0 \end{cases}
\]

ReLU function only has a linear relationship, so whether it is forward or backward propagation, the calculation speed is very fast, which is suitable for the network built in this time.

(3) Define loss function and use gradient descent method to reduce loss. Following is an example of \( y(i) = x(i) \times w \):

The parameter \( W \) in the optimization model \( y = w \times x \) is optimized to minimize the loss function of \( N \) samples in the training set. As shown in formula (2).

\[
\xi = \sum_{i=1}^{N} \|t_i - y_i\|^2
\]

The purpose of optimization is shown in formula (3):

\[
\arg\min_w \sum_{i=1}^{N} \|t_i - y_i\|^2
\]

From the function, it can be found that batch training accumulates the errors of all samples. In training, one sample is trained at a time. This method is often used in online training. For simple loss functions, it may be possible to see at a glance what the optimal weight is, but for complex or high-dimensional loss functions, it is necessary to use optimization methods to find the optimal weight. This time, the gradient descent method is used. The principle of gradient descent algorithm is to derive the loss function for each parameter, and update the parameters by using the negative gradient. Weights are updated through loops. Specific deductions are as follows:
W (k) denotes the value when the weight W is updated to step K. for w (k + 1), as shown in formula (4):
\[ w(k + 1) = w(k) - \Delta w(k) \]  
\[ \Delta w = \mu \frac{\partial \xi}{\partial w} \]  
\[ \mu \] denotes the learning rate, which means the span size of each step when the parameters are updated, \[ \frac{\partial \xi}{\partial w} \] denotes the gradient of loss function \( \xi \) to \( w \), as shown in formula (5).

For each training sample \( i \), \( \xi_i \) is the loss function of the first sample \( i \). The corresponding gradient can be derived by using the chain rule, as shown in formula (6):
\[ \frac{\partial \xi_i}{\partial w} = \frac{\partial y_i}{\partial w} \frac{\partial \xi_i}{\partial y_i} \]  
\[ \frac{\partial \xi_i}{\partial y_i} = \frac{\partial (t_i - y_i)}{\partial y_i} = -2(t_i - y_i) = 2(y_i - t_i) \]  
Because \( y(i) = x(i) \times w \), the derivation of \( \frac{\partial y_i}{\partial w} \) can be made in this way, as shown in formula (8):
\[ \frac{\partial y_i}{\partial w} = \frac{\partial (x_i \times w)}{\partial w} = x_i \]  
Therefore, \( \frac{\partial \xi_i}{\partial y_i} \) can be deduced in this way, as shown in formula (7):
\[ \Delta w = \mu \frac{\partial \xi_i}{\partial w} = \mu \times 2x_i(y_i - t_i) \]  
In batch processing, all gradients are accumulated, as shown in formula (10):
\[ \Delta w = \mu \times 2 \sum_{i=1}^{N} x_i(y_i - t_i) \]  
The next step is to visualize the gradient descent process, as shown in figure 5, where the abscissa represents \( W \) and the ordinate represents loss. It can be seen that when \( w = 2 \), the loss is minimized.

(4) Define the output layer to represent the specific value of weight loss.

5. Neural Network Training and Prediction

After the successful construction of the neural network model, the following variables are initialized: first, the parameter matrix is randomly initialized, then the activation values of the neurons in the next layer are calculated from the input layer, and then the activation values of the output neurons are calculated. This process is forward propagation. Now a simple three-layer neural network is used to introduce forward propagation. As shown in figure 6.
(1) The random initialization parameter matrices Θ(1) and Θ(2), as shown in formula (11):

\[
Θ^{(1)} = \begin{bmatrix}
\theta_{00}^{(1)} & \theta_{01}^{(1)} & \theta_{12}^{(1)} & \theta_{13}^{(1)} \\
\theta_{20}^{(1)} & \theta_{21}^{(1)} & \theta_{22}^{(1)} & \theta_{23}^{(1)}
\end{bmatrix}

Θ^{(2)} = \begin{bmatrix}
\theta_{10}^{(2)} & \theta_{11}^{(2)} & \theta_{12}^{(2)} \\
\theta_{20}^{(2)} & \theta_{21}^{(2)} & \theta_{22}^{(2)}
\end{bmatrix}
\] (11)

(2) Calculate the activation value of each neuron in the hidden layer, as shown in formula (12):

\[
a^{(2)}_1 = g \left( \theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3 \right)

a^{(2)}_2 = g \left( \theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3 \right)
\] (12)

(3) Calculate the activation value of neurons in the output layer, as shown in formula (13):

\[
a^{(3)}_1 = g \left( \theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} \right)
\] (13)

This is the process of calculating the activation value of forward propagation. Then the gradient of each layer's parameters is calculated by backpropagation, and the parameters are updated back to front. This process is called backpropagation. After much training, the final parameters are shown in figure 7.

Users input their own characteristics, and the data are fed into the neural network model. Then they can output specific values to predict weight loss, and then they can judge whether their plans are reasonable, as shown in figure 8.

6. Conclusion
In this system, firstly, the training samples are collected. Only enough training samples can get more accurate results. Secondly, for the construction of the model, the parameters such as learning rate are modified to find the minimum loss in the process of running the model. At the same time, the over-
fitting can be avoided.

Acknowledgements
This research is financially supported by the training program of innovation and entrepreneurship for college students in Hubei province in 2018 (No: 201810488089).

Reference
[1] Mingliang Liu, Dario Grana; Accelerating geostatistical seismic inversion using TensorFlow: A heterogeneous distributed deep learning framework; Elsevier journal; 2019-01-15
[2] Shouqiang Liu, Mengjing Yu, Miao Li, Qingzhen Xu; The research of virtual face based on Deep Convolutional Generative Adversarial Networks using TensorFlow; Elsevier journal; 2019-03-20
[3] Raul Orus Perez; Using TensorFlow-based Neural Network to estimate GNSS single frequency ionospheric delay (IONONet); Elsevier journal; 2018
[4] Evangelos Pournaras; Cross-disciplinary higher education of data science – beyond the computer science student; IOS Press; 2017
[5] E. Sande, E. Segers, L. Verhoeven; Supporting executive functions during children’s preliteracy learning with the computer; Wiley journal; 2016
[6] Evangelos Pournaras; Cross-disciplinary higher education of data science – beyond the computer science student; IOS Press; 2017