PAPER • OPEN ACCESS

Research on location detection of railway track spike based on neural network

To cite this article: Chen Lu and Yong Shen 2020 IOP Conf. Ser.: Mater. Sci. Eng. 711 012019

View the article online for updates and enhancements.
Research on location detection of railway track spike based on neural network

Chen Lu, Yong Shen*
School of Electrical Information Engineering, Yunnan Minzu University, Kunming, Yunnan, 650000, China
*Corresponding author’s e-mail: 740568005@qq.com

Abstract. The spike is an important link between the sleeper and the track. It plays a vital role in the quality of the railway and the safe operation of the train. At present, in railway engineering, the position of spike can be obtained by adjusting the parameters of magnetic sensor and repeated measurement tests. This method is simple but requires a large amount of labour cost, so this research attempts to make the "automatic guidance computer system" of the tamping vehicle automatically detect the distance from the iron rail nail according to the driving position of the tamping vehicle by using the BP neural network and the VGG19 neural network, thereby improving the maintenance rate of railway lines and reduced labour cost have laid the foundation for a grand blueprint for unmanned tamping vehicle.

1. Introduction
In recent years, with the advancement of science and technology and the rapid development of social economy, China's railway transportation is moving towards high speed, which puts forward higher requirements on the maintenance speed of railway lines. The spike is an important part of the fixed rail. When using the tamping vehicle to carry out maintenance work on the railway line, it is necessary to judge the position of the pick operation according to the detected position of the spike, so as to maintain and repair railway lines[1]. In railway engineering, the position of spike is determined by magnetic force line data measured by magnetic sensor. Specific rail spike position is obtained by adjusting the parameters of the magnetic sensor and repeated measurement tests. This method requires a large amount of labour cost, so it is envisaged that by using the algorithm of the neural network, the "automatic guiding computer system" of the tamping vehicle can automatically detect the distance between tamping vehicle and spike according to its driving position, thus according to the distance to determine the position of tamping pickaxe[2].

Figure 1. Magnetic signal distribution curve.
Figure 1 shows the distribution curve of magnetic signals measured by the magnetic sensor during a certain operation of the tamping vehicle. The position of each "small protrusion" in the figure is the position of the spike, and the peak of the "M" shape is an abnormal value. This research does not consider the outlier for the time being.

2. BP neural network algorithm

The BP (Back Propagation) network is a multi-layer feedforward network trained by error back propagation algorithm, which is one of the most widely used neural network models[3]. Its learning rule is to use the steepest descent method to continuously adjust the weights and thresholds of the network through back propagation to minimize the sum of squared errors of the network [3-4]. The topological structure of BP neural network model includes input layer, hide layer and output layer.

![Figure 2. Simple BP neural network structure.](image)

Figure 2 is a simple BP neural network structure, consisting of an input layer, two hidden layers, and an output layer, in which the circle is a neuron and the connecting line represents the transmission of values. The input layer and the output layer have two neurons respectively, which means that a two-dimensional vector can be passed in and a two-dimensional vector can be finally output. Taking the second neuron of hidden layer 1 as an example, the calculation principle process of BP neural network is shown in figure 3:

![Figure 3. BP calculation schematic diagram.](image)

Looking from left to right, $x_i$ represents the number of incoming values from the previous neuron. In the weight $w$, $i$ represents the sequence number of the neuron in the previous layer, $j$ represents the sequence number of pointed neuron, and the superscript $l$ is the level, $j$ represents the sequence number of this neuron. The weighted sum gives $v_j$, where $M$ is the total number of neurons in the previous layer. The results are substituted into the activation function $\varphi$, $y_i$ is obtained as the output of this neuron[5].

In order to explain the formula more clearly, the superscript $l$ of the weight $w$ is omitted, and the following formula is used with respect to the neurons of the output layer:

Enter:

$$v_j = \sum_{i=1}^{M} w_{ij} x_i + b_j$$

(1)

Output:
Our goal is to find the right $w$ and $b$, so that the output of the network is as close as possible to the expected output, then when we enter an untrained sample point, we can make an accurate classification[5]. In order to solve the problem, $w$ and $b$ satisfying uniform distribution or normal distribution were generated randomly at the beginning of training, and then the back propagation and gradient descent algorithm were applied to continuously modify $w$ and $b$ to minimize the total error. Now, let’s set the output layer neuron code to $k$, hidden layer 2 code to $j$, and hide layer 1 code to $i$. The gradient formula is deduced as follows:

The formula for deriving the gradient is as follows:

$$y_j = \varphi(v_j) \quad (2)$$

For the output layer neuron, the gradient of the total error with respect to the weight $w_{jk}$ is obtained by the gradient descent algorithm:

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial v_j} \frac{\partial v_j}{\partial x_j} \frac{\partial x_j}{\partial \omega_{jk}} \quad (3)$$

Similarly, the gradient of bias $b_{k}$ can be obtained:

$$\frac{\partial E}{\partial b_{k}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial v_j} \frac{\partial v_j}{\partial \omega_{k}} \quad (4)$$

$$\frac{\partial E}{\partial \omega_{k}} \text{ is reserved in the formula derivation to derive the local gradient:}$$

$$\delta_i = \frac{\partial E}{\partial y_i} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial v_j} = -(d_i - y_i) \varphi'(v_j) \quad (5)$$

With regard to local gradients, the weighted sum of all inputs to a neuron is regarded as a whole, weights $w$ and $b$ only affect other parts of the neural network through this whole[6]. Then the weight $w$ adjustment and the bias $b$ adjustment are:

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} = -\eta \delta_i x_j \quad (6)$$

$$\Delta b_{k} = -\eta \frac{\partial E}{\partial b_{k}} = -\eta \delta_i \quad (7)$$

Among them, $\eta$ is the learning rate, or called the step length. According to the actual situation, generally take a smaller number, such as 0.1, 0.01. Putting a negative sign in front of it means searching in the direction of the negative gradient.

### 3. VGG convolutional neural network

The VGG network uses 3*3 convolution kernels. The changes of the convolution kernel will not only affect the calculation amount of the network, but also affect the size of the network to obtain the receptive field, and ultimately affect the learning mode of the network. Smaller convolution kernels are easier to capture changes in image feature details, and at the same time have a greater impact on the amount of computation of the network[7]. In the network, a smaller convolution kernel (3*3) is used, the stride is 1, and 2-4 convolution layers constitute a "coil unit" of 3*3. Some layers also have a 1*1 convolution layer. The 1*1 convolution kernel can be regarded as a linear transformation and the stride is 1, so the network is deeper. The pooling layers use the maximum pooling layer of 2*2. There are five layers and the stride is 2. In order to obtain more detailed information, ReLU activation function is used for all hidden layers[8]. In the full-image test, the last fully-connected layer is changed to the fully convolutional net, and the parameters during training are reused. Because there is no full connection limit, the network can receive input of any width or height, and it has a remarkable feature: the spatial resolution of the feature map is monotonically decreasing, and the number of channels of the feature graph is monotonically increasing, in order to better convert the image of $H*W*3$ into the output of $1*1*C$. 
4. Analysis of spike detection based on BP and VGG networks

In the data preparation stage, the distribution curve of magnetic signals measured by magnetic sensors is divided into many magnetic signals of the same length according to a step length every 50 meters and a segment length of 784 meters, which were respectively stored in the CSV files. Then, the cut CSV files are randomly divided into training data sets, validation data sets and test data sets. Set a threshold according to the experience, in every section of the magnetic signal, if there is greater than the threshold value is that there is a spike in this signal, and output spike position and the location of tamping car distance value of sample label, if not is greater than the threshold value of the signal, this signal does not exist spike, output sample label value of -1. Finally, the training data and sample labels are input into the established BP neural network model, and the data is trained so that the output result is the distance between the position of the spike and the position of the tamper. And the results are show in figure 4. Then the data is reconstructed to a size of 224*224 and input into the neural network model of VGG19 for train, and the results are show in figure 5.

![Figure 4. BP neural network training results.](image1)

![Figure 5. VGG neural network training results.](image2)

![Figure 6. Comparison of BP and VGG loss values.](image3)

Figure 4 shows the training results using BP neural network. It can be seen that the effects on the training set and test set cannot converge well, and the loss value has no obvious downward trend. Figure 5 shows the training results using the VGG19 neural network. It can be seen that the loss value decreased from about 0.7 at the beginning of the training to about 0.02, and the error rate is about 0.032. Although there is a certain fluctuation, the effect is relatively ideal. Figure 6 is the comparison diagram of loss value between BP neural network and VGG neural network. It is obvious that at the end of training, the loss value of using VGG model is much smaller than that of BP neural network and converges faster. The conclusion can be drawn that the distance between the rail spike and the tamping vehicle can be determined by the neural network method, and the VGG neural network has a good effect.
5. Conclusion
In this research, BP neural network and VGG neural network are adopted to detect railway track nails, and the advantages and disadvantages of the two methods are compared, and the conclusion is drawn that the railway track nails can be identified by using neural network, which provides a certain theoretical basis for the realization of unmanned tamping vehicle and has certain research significance.

References
[1] Xiaoli Kang. (2018) Application and Research of Large-scale Road Maintenance Machinery for Railway Lines. J. China Strategic Emerging Industry., 40: 186-186.
[2] Lin J.M. (2013) BP neural network analysis. https://blog.csdn.net/linj_m/article/details/9897839.
[3] Zuoyong Li, Jiayang Wang, Chun Guo. (2008) New method and simulation experiment of PSO algorithm to optimize BP network. J. Acta electronica sinica., 11: 2224-2228.
[4] Jing Jin, Zonghao Li, Liang Zhu, Xinhao Tong, Changwei Yang. (2019) Application of BP neural network in railway construction risk assessment. J. Journal of railway engineering., 36: 103-109.
[5] Rui Fang, Junyang Yu, Lifeng Dong. (2019) Spam text filtering based on B-Feature feature construction and BP neural network. https://doi.org/10.19678/j.issn.1000-3428.0055414.
[6] Shichun Peng. (2019) Research on image semantic segmentation algorithm in deep learning computer network. J. Journal of guiyang university (natural science edition)., 14: 22-26+61.
[7] Simonyan K, Zisserman A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. J. Computer Science., 34: 64-67.
[8] Mengzhuo Yang, Mengjie Guo, Liang Fang. (2019) Research on image classification algorithm based on keras convolutional neural network. J. Science and technology wind., 23:117-118.