Nat. Volatiles & Essent. Oils, 2021; 8(6): 6252-6259

NVEO
Natural Volatiles &
Essential Oils

Tensorflow Based Cnn Model

MEGANATHAN.E[1] , DR. P. Malathi[2]

[1]Assistant Professor, Department of Information Technology, Dhanalakshmi srinivasan College of Engineering and Technology

[2]Associate Professor, Dhanalakshmi Srinivasan College of Engineering and Technology.

Abstract

Google's open source deep learning and machine learning framework is TensorFlow, which is flexible and convenient to make the modern mainstream deep learning model. The advantage of a convolutional neural network over other deep learning models is the powerful feature extraction capabilities of convolutional blocks. A convolutional neural network model with two convolution layers was developed using the TensorFlow platform. The MNIST data collection was used to train and test the model. the test accuracy rate could reach 99.25 percent, compared to 98.69 percent with only one-convolution. Layer model, demonstrating that the two-convolution-layers convolutional natural network model has a stronger ability of feature extraction and classification decision-making. The advantage of a convolutional neural network over other deep learning models is the powerful feature extraction capabilities of convolutional blocks

The,advantage of a convolutional neural network over other deep learning models is thepowerful feature extraction capabilities of convolutional blocks.

A convolutional neural network model with two convolution layers was developed using the TensorFlow platform. The MNIST data collection was used to train and test the model. The test accuracy rate was 99.15 percent, compared to 98.69 percent with a single-convolution-layer model, demonstrating that the two-convolution-layers convolutional neural network model has a stronger ability of feature extraction and classification decision-making.

Keywords: CNN, Deep learning, machine learning, Tensorflow.

INTRODUCTION

The technical revolution brought on by deep learning has extended rapidly in the artificial intelligence wave. Deep learning, as a form of machine learning, can not only learn the relationship between features and tasks, but also extract more complicated characteristics from simple samples automatically. Machine learning paves the way for artificial intelligence, whereas deep learning turns machine learning into reality. [1]. Deep learning has been extended to different disciplines of machine learning in recent years, bringing significant convenience to people's lives. Google's TensorFlow, Microsoft's CNTK, Baidu's PaddlePaddle, California University's Caffe, Montreal University's Theano, Facebook's Torch, and others are currently the most popular open source deep learning frameworks. In November 2015, TensorFlow

became open source. It is a deep learning system of the second generation. The system's stability is also improved, so consumers are more concerned and use it. TensorFlow is a high-level machine learning package with C++ and Python as supporting languages. Building a neural network structure does not require users to write sophisticated code. The optimization techniques and functions used in modelling can simply call the TensorFlow library's comparable functions, drastically lowering the deep learning threshold. Ordinary computers can now generate deep learning models on the TensorFlow platform, thanks to rapid advancements in computer technology and tremendous increases in processing power, lowering the cost of deep learning and making algorithm verification easier. [3]

2. Deep learning model

A deep neural network is a convolutional neural network (CNN). CNN's core concept can be summed up in two words: sparse connection and shared weights. An input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer make up a standard convolutional neural network. Forward and reverse propagation are the two basic processes involved. Finally, the former outputs the prediction result via the network structure, while the latter adjusts the parameters based on the difference between the predicted result and the actual value. The input layer is a matrix of Nrr that contains image data that has to be input. A fixed-size filter (convolution kernel) is twisted with the image of the preceding layer in the convolutional layer.

$$xj\ l = f\ (\sum xi\ l{-}1\ i{\in}Sj * \omega ij\ l + b\ ) \quad (1)$$

Sj represents a set of input images, xi l1 represents a feature map of the (l-1)th layer's i-th input, ij l represents the convolution kernel from the (l-1)th layer's i-th input feature map to the jth output feature map of the l-th layer, bj l represents the bias value corresponding to the j-th output feature map of the l-th layer, f(•) The down-pooling layer, also known as the pooling layer, is a unique convolutional layer. There are two types of pooling: maximum pooling and average pooling.

The approach is the same as in equation 1, only the activation function is replaced with a poolin g function. The transformation's goal is to minimise the dimension of the feature map, not the number of feature maps, but the size of the feature map, reducing the parameters in the fully linked layer and speeding up the calculation. The convolutional layer and the ordinary layer are both connected to the fully connected layer. From the beginning of the input to the full connected layer with the "convolution block"; from the full connected layer to the output, which are all ordinary layers of the final "NN block." The fully connected layer will take the high-dimensional data from the parent layer (convolution layer) as input in a flat form similar to that shown in Figure 2, apply a nonlinear transformation (activation function), and then feed the result into the following system of ordinary layers. In these levels, more decisions (prediction and classification) are made. As a result, the "convolution block" is renamed "feature extractor," and the "NN block" is renamed "decision-making classifier." The degree of model fitting is reflected in the loss function. The gradient vanishing problem is perfectly solved by the cross entropy loss function. Equations (2) [4] demonstrate the principle.

$$L(y, G(x)) = -[ylnG(x) + (1 - y)ln(1 - G(x))]\ ,\ G(x) = (v1,\ldots, vk)T\ ,\ \sum vk = 1\ k\ 1 \quad (2)$$

The adaptive learning rate Adam algorithm, which is an extension of the stochastic gradient descent algorithm and is the most extensively used and generally best algorithm in deep learning applications, is utilised in the back propagation optimization method of convolutional neural networks. The Adam algorithm back propagates the error, updating the parameter values of the convolutional neural network layers layer by layer. Equations (4) [5] demonstrate the reduced version's premise.

$\Delta \Leftarrow \beta 1\Delta + (1 - \beta 1)\Delta wt$, $\nabla 2 \Leftarrow \beta 2\nabla 2 + (1 - \beta 2)\Delta 2wt$, $\Delta * wt \triangleq \Delta \nabla + \eta$ (3)

2.1 Classification Model of CNN

A convolutional neural network classification model is shown in Figure 1 as a schematic structure diagram. The input layer, two convolutional layers, two pooling layers, a fully connected layer, a Dropout layer, a Softmax layer, and an output layer are all shown in the diagram. The convolution layer convolves the data of the upper layer through the convolution kernel, and then uses the activation function to obtain the convolution layer's feature map; the pooling layer finds the maximum value through each of the upper layer's 22 neighbourhoods in the feature map, so that the feature map's dimension after pooling is half that of the upper layer; and the fully connected layer tiles the 64 feature maps of the upper layer.The dropout layer is used to prevent overfitting and improve the model's generalisation capacity. F inally, the softmax regression model classifies the data to produce the category.

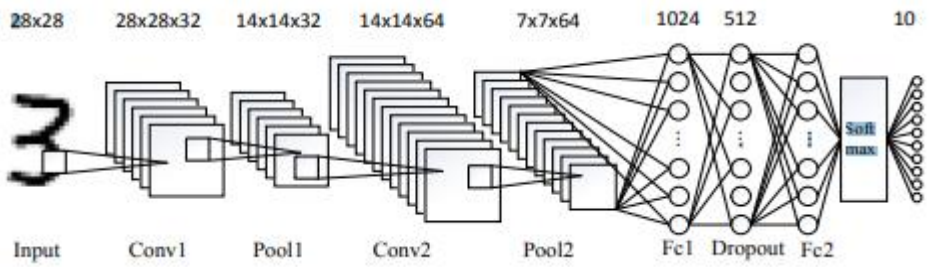

Fig 1: CNN Structure's in Semantic Diagram

3. **TensorFlow using MNIST data set**

3.1. **TensorFlow and TensorBoard**

The calculation model: calculation graph, the data model: tensor, and the runtime model: session are the three models of TensorFlow. A Tensor data structure represents the data in TensorFlow, while Flow represents the data flow and calculation. It can assign (or retrieve) data in the tensor using the feed (or fetch).

TensorFlow is a deep learning programming framework that uses a computational graph to express calculations. Figure 2 [6] depicts the TensorFlow programme development flow chart, which is divided into two stages: building and execution graph.

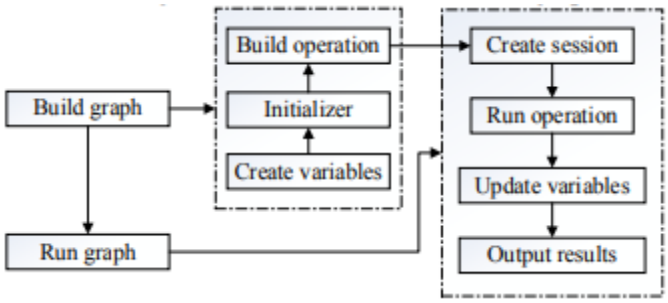

Fig 2. program development flow chart for TensorFlow

The TensorBoard tool provides a visual representation of the TensorFlow calculation graph. It can see the output log files while the TensorFlow programme is running, as well as effectively display the calculation graph and the trend of various parameter indicators over time while the programme is running, making it easier to comprehend and troubleshoot.

3.2. **MNIST data set**

The MNIST handwritten digit recognition data collection contains two types of images: 60,000 training sample sets, which include 55,000 training samples and labels, 5000 training verification samples and labels, and 10,000 test samples and labels. Each sample is a picture, and each picture is represented by a 28 x 28 matrix tiled into a 724-dimensional data row. The label value is a 10-dimensional vector that represents the 0-9 category number in one go. Handwritten digit recognition focuses on feature extraction and classification decision-making [7].

3.3. **Program implementation steps**

Anaconda, the most popular open source Python data science platform, was used to write the application, while Spyder was used as the Python development environment (Scientific Python Development Environment). Figure 3 shows a computational graph of the entire model structure displayed by the TensorBoard, which includes two convolutional layers Conv1 and Conv2, two pooling layers Pool1 and Pool2, one fully connected layer Fc1, one output layer Fc2, and Dropout between the Fc1 and Fc2 layers to prevent data overfitting; accuracy indicates the model's accuracy when using the test sample to predict the result and compare with the label value; cross entropy is the cross entropy function model.

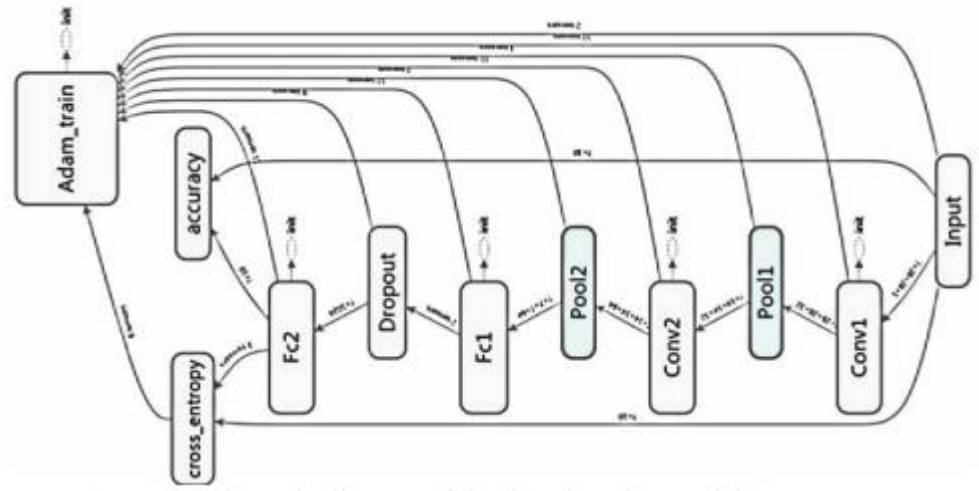

Figure 3: Deep learning model construction schematic picture

3.3.1. Input layer and Initialize the weight and bias value.

The training and test sets' inputs are defined as two placeholders x and y, and the 784-dimension data is transformed into a 2828 matrix picture. The bias value is initialised to 0.1 and the weight is initialised with a truncated normal distribution.

3.3.2. Convolution and pooling layer.

The convolution kernel's step size in both the x and y directions is one when defining the convolution process. padding='SAME' is used. The pooling window size is 22; the first convolution layer's weight values are initialised, the sampling window size is 5 5, 32 convolution kernels extract features from one plane, and each convolution kernel has a bias value; convolve the input image x image with the weight vector, plus the bias value, then apply to the relu activation function; Finally, optimise output pooling; similarly, initialise the second convolutional layer's weight and bias value, except that 64 convolution kernels take information from 32 planes. Layer 3.3.3 is fully connected. Because of the SAME PADDING, the initial 2828 picture is still 2828 after the first convolution, as seen in the prior method. Because of the pooled 22 window and the step size of 2, the feature map becomes 1414 after the first pooling; similarly, it becomes 1414 after the second convolution and 77% after the second pooling. Finally, the aforementioned operation yields 6477 planes, which are used as the input to the next fully connected layer. Set 1024 neurons, 1024 bias values, then flatten the 64 feature maps into 6477-dimensional data before multiplying them by the weight vector plus the offset value. Then it was used to find the relu activation function.

3.3.4. Dropout layer.

To avoid data from overfitting, use the keep prob function to signal the neuron's output probability and the tf.nn.dropout() function to achieve dropout.

3.3.5. Softmax output layer.

Set 10 output neurons in the second fully connected layer, and use the tf.nn.softmax() regression model function to output the final classification result.

3.3.6. loss function and Optimization training.

Select the tf.nn.softmax cross entropy with logits() cross entropy loss function, then optimise and minimise the loss after constructing it with the tf.train. The AdamOptimizer() function is an optimization training algorithm.

3.3.7. Accuracy

Put the classification results in a boolean list and use the argmax() function to get the largest value in the tensor's location. Finally, use the euqal () comparison function to see if the result is valid, and get the classification accuracy..

3.3.8. Session execution.

Due to the large amount of training data, batch training is employed to create 100 training charts every batch. All sample data can be trained after 550 trainings. Sessions are created, variables are initialised, training data is given, and the model is trained for 10,000 steps, with the parameters and test set accuracy recorded every 100 steps.

3.4. Result analysis

The final test output after 10,000 trainings. The two-layer network on the left has an accuracy of 99.15 percent, which is 0.46 percentage points greater than the one-layer network's accuracy of 98.69 percent. Figure 7 shows the relationship curve between CNN model training times and accuracy, as obtained from the visualisation tool TensorBoard, where the abscissa represents the number of trainings and the ordinate represents the recognition accuracy.

The number of training sessions rises, the accuracy improves steadily until stabilizing. When trained to 200-300 times, the accuracy rate of the two-layer convolutional layer neural network reaches 90%; when trained to 1100 times, the accuracy rate is 97 percent; and when trained to 6000 times, the accuracy rate is essentially stable at approximately 99 percent. The accuracy curve of the two-convolution-layers neural network is generally higher than that of the one-convolution-layer neural network, and the error of the two models decreases with the number of trainings, but the error curve of the two-convolution-layers neural network is significantly lower than that of the one-convolution-layer neural network. The data demonstrates that the two-convolution-layer neural network model has higher recognition accuracy, faster convergence, and better feature extraction and classification.

4. Conclusion

This study creates a deep learning model called CNN using the open source TensorFlow library and the Spyder Python language development environment. To train and evaluate the model, the MNIST data set is utilised as an example. The study uses the visual tool TensorBoard to show the structure of the deep

learning model, as well as test results and trend curves, to validate the model's validity. A layer of convolution and pooling operations is added to the traditional convolutional neural network, and the second convolution pooling operation is useful for deeper levels to extract a larger feature quantity, resulting in a recognition accuracy of 99.25 percent.

5. References

[1] Zeyu Zheng., TensorFlow actual Google deep learning framework [M]. Electronic Industry Press., 2018:1-4,287.

[2] Tao Jing, Yongai Zhang., Digital recognition based on deep learning under the TensorFlow platform [J]. Information technology and network security, 2018,37 (04): 74-78.

[3] Jiahui Xiao., The application prospect of clustering algorithm in TensorFlow platform is discussed [J]. Digital technology and applications, 2018,36(05):150+187.

[4] Ian Goodfellow, Youshua Bengio, Aaron Courville., DEEP LEARNING[M]. Posts and Telecom Press, 2018:201-203.

[5] Yujian He., Python works with machine learning [M]. Electronic Industry Press, 2017:200,221- 222. [6] Shanjie Han, Shizhe Tan., Design and implementation of a deep learning model for stock forecasting based on TensorFlow [J]. Computer applications and software, 2018,35(06):267- 271+291.

[7] Yuan Xing., Application of deep learning in handwritten numeral recognition [D]. Suzhou University, 2017.

[8] Yun Xu, Haowei Yuan, Zhi Li. Handwritten numeral recognition in convolutional neural network and TensorFlow [J]. Shanghai electric technology,2018,11(01):31-34+61.

[9] Hao Chen, Hai Guo, Daquan Liu, Jiaqi Zhang., Handwritten numeral recognition system based on TensorFlow [J]. Information communication,2018(03):108-110.

[10] Panhai Zheng, Ling Guo, Libing Ding., Research and implementation of convolutional neural network based on TensorFlow [J/OL]. Electronic technology and software engineering, 2018(18):20-22[2018-10-11].

[11] Zhao Jin., Comparison and analysis of different deep convolutional neural networks based on TensorFlow [J]. Electronics World,2018(06):25-26.

[12] Yanlu Hou, Shifei Ding, Tongfeng Sun., The mixed depth learning model C-RF and its application in handwritten numeral recognition [J]. data acquisition and processing, 2018,33(02):343-350.

[13] F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow,2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 755-758.

14] T. Makkar, Y. Kumar, A. K. Dubey, Á. Rocha and A. Goyal, "Analogizing time complexity of KNN and CNN in recognizing handwritten digits," 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, 2017, pp. 1-6.

[15] M. A. R. Alif, S. Ahmed and M. A. Hasan, "Isolated Bangla handwritten character recognition with convolutional neural network," 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, 2017, pp. 1-6.

[16] S. Iamsa-at and P. Horata, "Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Network," 2013 International Conference on IT Convergence and Security (ICITCS), Macao, 2013, pp. 1-5.