Traditional 2-layer Neural Network and Convolutional Neural Network for Image Classification and the Implementation of Explanation Method

Zizheng Huang

Research School of Computer Science,
The Australian National University,
Canberra ACT 2601,
U6248330@anu.edu.au

Abstract. In this paper, I trained a traditional 2-layer neural network and a convolutional neural network (CNN) for classification task on a large-scale synthetic database VehicleX. This database contains many images of different vehicles. Each image has corresponding labels, such as vehicle orientation, camera height, light intensity and so on. For traditional neural network processing, each image is represented as an array of 2048 in length. I chose the top 250 most important numbers in the array as the input. For CNN, the inputs are images which are resized to 3×32×32. The neural network is used to predict the type of vehicle in the image. The results show that the two kinds of neural network model both can make correct prediction to a certain extent, and CNN has much better performance. At the same time, I used the average of the inputs with the same predictive output as the characteristic pattern to build a simple set of rules to explain my models. Finally, the performance and applicability of this interpretation method are compared and analyzed.

Keywords: Convolutional Neural Network, Neural network, Image classification, Characteristic input method, Neural network explanation

1 Introduction

1.1 Dataset

As the visual basis for humans to perceive the world, images are an important means for humans to obtain information, express information and transmit information. Automatically classifying vehicle images has broad application prospects in surveillance recognition, autonomous driving and other fields. Therefore, how to use neural networks to process images has become an important topic nowadays.

The purpose of this experiment is to use neural networks to classify vehicle pictures. This database used in this paper is called VehicleX. I used the train file and the test file from it. The train file contains 45438 pictures, and the test file contains 15,142 pictures. Also, an xml file provides detailed labels for each image, including vehicle orientation, light intensity, light direction, camera distance, Camera height, camera ID, vehicle type and vehicle color. Those images are all synthetic data, so they won't be affected by the privacy issues and the complexity of manual labelling that may arise from real data [1]. So, this data set can be used for neural network learning. In this paper, the classification target I selected is TypeID, each picture has a TypeID, and each ID corresponds to a type of vehicle. VehicleX contains a total of eleven types: sedan, SUV, van, hatchback, MPV, pickup, bus, truck, estate, sportscar, and RV [1]. For traditional neural network processing, each picture is converted in the form of an array. Each array contains 2048 features. For CNN, we directly use the original image as input, and the size of each image is 3×256×256.

1.2 Problem and neural network models

The problem to be solved in this paper is to predict the type of vehicle in the picture by building and training a simple neural network model and a convolutional neural network model. The position, orientation, light intensity and angle of the vehicle in each picture are different. This is quite challenging for traditional NN forecasting. For the database, although there are eleven types of vehicles in the database. But their distribution is not uniform. In order to avoid the lack of sufficient data for certain types of vehicles for training the traditional neural network. I selected three vehicle types with the highest proportion in the dataset for training and prediction. In additional, to get as accurate results as possible, I tried many hyperparameter settings, such as the number of hidden layers, the number of hidden neurons, the type of activation function, and so on. The adjusted neural network has one hidden layer. The input layer accepts the 250 most influential features in each picture. The activation function is the ReLu function, next is the SoftMax layer. Finally, the model uses backpropagation to reduce Cross Entropy Error. For CNN model, since convolutional neural networks are more suitable for processing multi-dimensional data, so I choose original pictures as input, My CNN model is an improvement based on the LeNet model, it uses convolution, parameter sharing and pooling to extract features, avoiding huge computational costs, and finally using a fully connected layers for classification and recognition [7].

1.3 Characteristic Input Method

Although the neural network can predict the type of vehicle. But we still hope that there is a way to intuitively explain how this network makes predictions. The second part of this paper uses the Characteristic Input Method to explain the model. The core idea of the Characteristic Input Method is to classify the input based on the predicted output. And find the corresponding rules in each type of input to explain this neural network [2]. There are many ways to generate rules. In this paper, I used the arithmetic mean value of each input in the same category as the characteristic pattern of this category. Then predict the output of the neural network by calculating the Euclidean distance between the input and the pattern. Finally, I test and compare the applicability of this expression on different models and datasets.

2 Method

2.1 Data preprocess for traditional neural network

The original database contains a total of 75,516 images, which are divided into train set, test set, and verification set. I selected train set and test set for neural network training. Each picture is stored in the form of an array. There is also an xml file, which contains 9 different kinds of detailed labels for each picture, from which I chose TypeID as the prediction target. Different TypeID corresponds to different vehicle types. Considering that each picture contains 2048 features, inputting all of them into the neural network will greatly increase the computational complexity. So, I compute pairwise correlation of columns to select the top 250 features that are most relevant to TypeID. Taking the train set as an example, as we can see from Fig. 1 that the data distribution in the database is not uniform, the vehicles with type IDs of 0, 3, and 9 account for the majority. In order to prevent the inaccurate predictions caused by the inability of the neural network model to be adequately trained on the vehicle types with small amount, I only retained the top three vehicles with the largest proportion for neural network prediction. They are sedan, hatchback and sportscar. In order to make the data set more coherent, I reassigned the TypeID to 0: sedan, 1: hatchback and 2: sportscar.

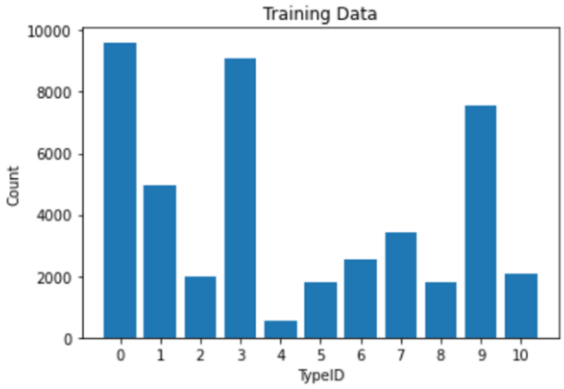

Fig. 1. Distribution of TypeID in training data.

2.2 Data preprocess for convolutional neural network

In the convolutional neural network model, I will use convolution, parameter sharing and pooling to extract features. This can effectively reduce the complexity of the model, so I directly use pictures as input. But due to the large number of pictures and the larger size of the pictures (3×256×256), I resize the pictures to 3×32×32, which can further reduce the computational cost. In order to make the CNN model being fully trained and have better applicability, the training target I adopted this time is all eleven kinds of TypeID. Next, I use the defined mean and standard deviation to standardize the images, then I transform the images into tensors for further processing.

2.3 Traditional 2-layer Neural Network

The first part of this paper is to build a simple neural network based on the python torch library. Fig.2 shows the structure of this neural network. It contains an input layer, a hidden layer and an output layer. The input layer has 250 neurons, corresponding to the 250 features of each image. The hidden layer has 64 neurons, and the output layer has 3 neurons, corresponding to the three types of vehicles. I choose ReLu as the activation function. Compared with the sigmoid function, which requires exponential calculations, ReLu has a smaller amount of calculation, so it will converge faster when using backpropagation calculations. And because the derivative on the positive half axis of x is

always 1, It alleviates the phenomenon that the use of the sigmoid activation function in the deep network will cause the vanishing gradient problem. It can also alleviate the problem of overfitting. Because the ReLu function turns values less than zero into zero, which causes the sparsity of the network and reduces the interdependence of parameters to avoid the problem of overfitting [3].

The model also passes through the Softmax layer before outputting the result. The Softmax function can map the input to a real number between 0-1, and the normalization guarantees that the sum is 1. Then it uses Cross Entropy Error to measure the quality of the prediction results. Then I use backpropagation to optimize the parameters. Among it, I use stochastic gradient descent as the optimization algorithm. It can speed up the convergence process and solve the problem of excessive data volume [4]. After a series of parameter tuning. The result shows that when the learning rate is 0.08, the best prediction result can be obtained after 1500 epochs.

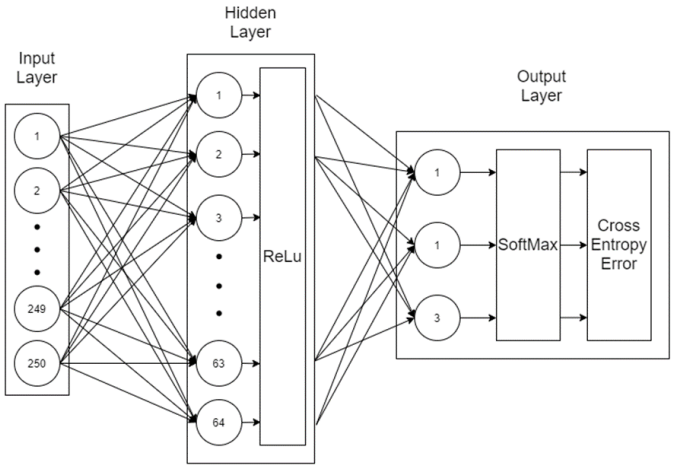

Fig. 2. The structure of the 2-layer Neural Network.

2.4 Convolutional Neural Network

The CNN model used in this paper is adjusted and improved from the LeNet model [7]. Fig.3 shows the structure of this model. After the image is normalized and resized, it enters the convolutional layer first, it uses 10 convolution kernels with a size of 3*3 to perform operations to obtain 10 feature maps. Then the maximum pooling operation is performed, in this layer, the kernel size is 2*2 and the stride is 2. Then repeat the convolution and pooling operations, and finally perform image classification through the fully connected layer. After a series of parameter tuning. The result shows that when the learning rate is 0.002, the best prediction result can be obtained after 20 epochs.

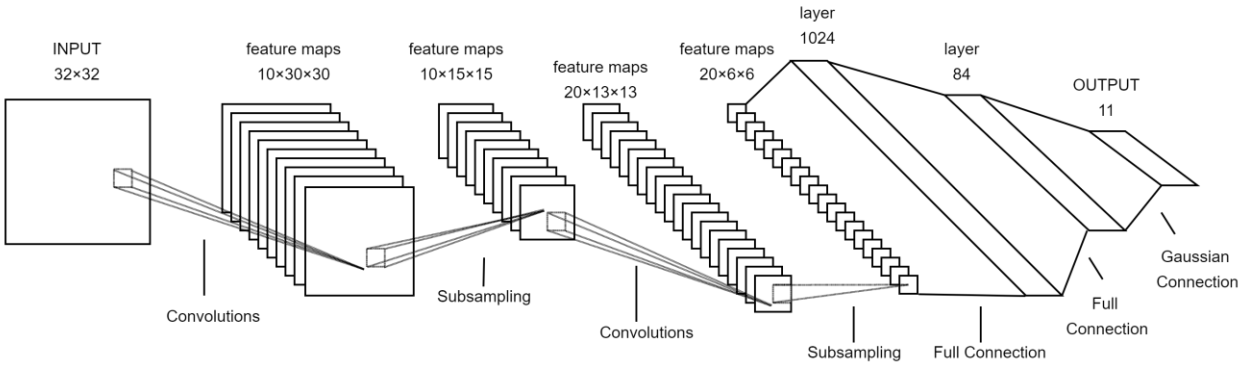

Fig. 3. The structure of the Convolutional Neural Network.

2.5 Neural network explaining

The Characteristic Input Method can transform the neural network into a series of rules and use these rules to intuitively express how the neural network makes predictions. In this paper [2], they used characteristic ON pattern and characteristic OFF pattern. This is for binary classification problem, where the input is grouped according to the predicted output. And in each group, extract the rules as Characteristic Input pattern. However, there is a multi-classification problem to be solved in my thesis. So, I divided the train dataset into 11 different characteristic patterns according to the TypeID predicted by the neural network. After dividing into groups, there are many ways to extract rules. One of the methods is to generate a series of IF-ELSE rules [2]. However, this method cannot be applied to the database in this paper. First, there will be great computational complexity to generate rules for thousands of input. Secondly, these features represent points in the picture and do not have semantic meaning. Decision tree is more suitable for interpretation rules for features with semantic meaning such as sepal length in iris data. So, I chose to calculate the arithmetic mean of each kind of inputs in the same group. Then calculate the Euclidean distance between each picture in the test dataset and those three characteristic patterns, choose the closest one as the corresponding output. Finally, the prediction result of this characteristic Input method is compared with the prediction result of the neural network to verify whether the method can explain the neural network well.

3 Results and Discussion

Fig. 4 shows the loss of the traditional neural network converges at 1500 epochs, the loss value currently is about 1, and the value of accuracy is about 51%. This is not a satisfactory accuracy rate. There are two main factors leading to this result, one is the complexity of the database itself, and the other is the limitations of simple neural networks. First, in these pictures, the vehicle orientation, light intensity, light direction, camera distance, camera height and vehicle color are all different. This creates a lot of noise for classification tasks that only focus on vehicle types. For example, the same SUV vehicle may produce very different pictures because the orientation of the vehicle and the angle of the camera are different. Secondly, considering the structure of an ordinary neural network, I only selected a small part of the key points in each picture as data training, which will inevitably lead to a lack of information.

At the same time, as a neural network with only the most basic structure, the model cannot well capture the relationship between various input features, which means that it cannot extract all the information contained in the input image [6]. Especially in this dataset, which is arrays converted from pictures. Therefore, the numbers in each array should not be calculated in isolation.

However, as we can see from Fig.5 and Table 1, the loss of the CNN converges at 20 epochs. Compared with traditional neural network, the performance of CNN has been greatly improved. The prediction accuracy of traditional neural networks for the 3 TypeID is only 50%, while the prediction accuracy of CNN for all 11 TypeID has increased to 72%. This proves that CNN can effectively improve the shortcomings proposed in the previous paragraph. The CNN model uses the original image as input, and uses convolutional layer, pooling layer, activation function, as well as local receptive fields, weight sharing, and downsampling, those strategies reduce the complexity of the network model and can effectively learn the corresponding features from a large number of samples, avoiding the complex feature extraction process [5]. In CNN, it only perceives the local pixels of the image, and then merge this local information at a higher layer to obtain all the characterization information of the image. This kind of network structure is highly invariant to translation, zoom, tilt or other forms of deformation. Moreover, the convolutional neural network uses the original image as the input, which can effectively learn the corresponding features from a large number of samples, avoiding the complex feature extraction process, thereby improving the prediction accuracy.

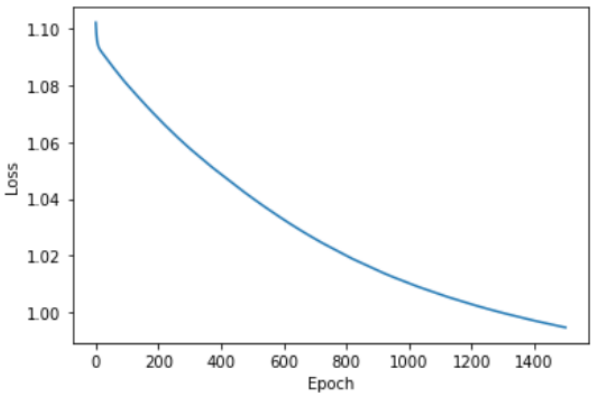

Fig. 4. training loss over the number of epochs of traditional NN.

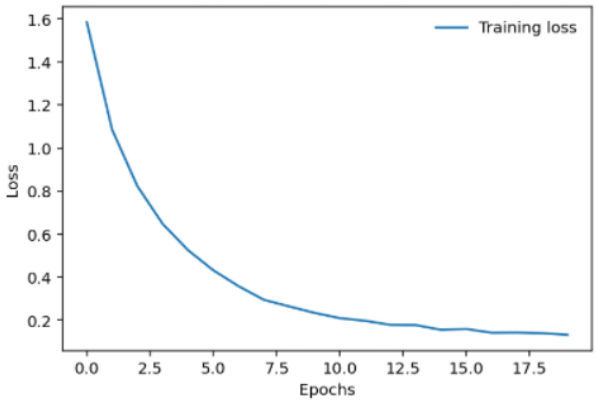

Fig. 5. training loss over the number of epochs of CNN.

Model	Accuracy	Cross Entropy Loss
Traditional NN	50.59% (for 3 TypeID)	0.9924
CNN	71.71% (for 11 TypeID)	0.1325

Table 1. The Loss and Accuracy of traditional NN and CNN

As the table 2 shows when I apply the characteristic pattern trained on the train set to the test set and compare it with the prediction result of the two kinds of neural network, it shows that the interpretation accuracy of this method on traditional NN is 65%, however it is only 16% for CNN. This result indicates that the neural network can be explained to a certain extent by comparing the distance between the input and the characteristic input pattern on the traditional NN dataset, but it can hardly make an effective explanation on the CNN data set. Since for the CNN dataset, in order to calculate the distance between characteristic pattern, I flatten the 3×32×32 image into a one-dimensional array, which has 3072 features for each image array. Obviously, such an unprocessed array contains a lot of noise, so it requires more preprocessing. Since each picture contains three layers of RGB, and the color is not helpful for the recognition of the vehicle type. The same type of car can be any color, so it is more appropriate to convert it into a grayscale image first. Characteristic Input Method has better results on traditional NN databases, because these arrays have been preprocessed, but there are still two main reasons that affect the prediction effect. One is that simply taking the input average value as a characteristic pattern is not accurate enough. Not all input features have the same importance. The rules formed by the decision tree are more robust, but they are not suitable for long sequences which are converted from image pixels. Secondly, simply taking the Euclidean distance may cause the prediction result to still be biased. If machine learning is used for classification training, the accuracy may be improved, but it will also increase the computational burden accordingly.

Model	Accuracy of characteristic pattern
Traditional NN	64.99%
CNN	16.46%

Table 2. The Accuracy of characteristic pattern explanation of traditional NN and CNN

4 Conclusion and Future Work

This paper uses a two-layer neural network and a Convolutional Neural Network to solve a picture classification problem and uses the Characteristic Input Method to explain the neural networks. The experimental results show that this two-layer neural network also has good potential in dealing with image classification problems, but it still needs further optimization. CNN has solved the shortcomings of traditional neural networks and achieved better results. It also proves that the Characteristic Input Method can effectively explain the neural network. However, there are higher requirements for the processed data, and the data needs to be preprocessed for different problems. The generated rules can help people more intuitively understand how the neural network completes the classification task.

Future work has three aspects, namely database processing, neural network structure and Characteristic input method improvement. First, there are many features of vehicles in this database. If we want to classify only one of the features, it is best to control the variables. Keep other features, such as vehicle orientation the same will improve accuracy. A complete data set is more suitable for fine-grained classification, such as identifying each car. In this case, there will be 1362 different classes. Second, the previous part has demonstrated the limitations of simple two-layer neural network processing images. Convolutional Neural Networks is a better choice, it can be further improved by Increase the depth of the network and use methods such as drop-out to improve generalization capabilities. Third, for the Characteristic Input Method, we can use the weighted average to calculate a characteristic pattern to reflect the difference in importance of input features and improve the accuracy of interpretation through machine learning.

References

1. Yao, Y., Zheng, L., Yang, X., Naphade, M., & Gedeon, T. (2019). Simulating content consistent vehicle datasets with attribute descent. *arXiv preprint arXiv:1912.08855*.

2. Gedeon, T. D., & Turner, H. S. (1993, October). Explaining student grades predicted by a neural network. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)* (Vol. 1, pp. 609-612). IEEE.

3. Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

4. Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177-186). Physica-Verlag HD.

5. Bouvrie, J. (2006). Notes on convolutional neural networks.

6. Khotanzad, A., & Lu, J. H. (1990). Classification of invariant image representations using a neural network. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(6), 1028-1038.

7. El-Sawy, A., Hazem, E. B., & Loey, M. (2016, October). CNN for handwritten arabic digits recognition based on LeNet-5. In *International conference on advanced intelligent systems and informatics* (pp. 566-575). Springer, Cham.