A Comparison Study Base on Deep Learning and Multi-task Learning

Danyang Li

Research School of Computer Science,
Australian National University
U7169663@anu.edu.au

Abstract. Artificial neural networks[1], as useful and powerful tools, have the ability to learn from examples. In this task, a three-layer neural network and a multi-task model been used to preform fine-grained classification on VehicleX[2] dataset to predict different vehicle. The gradient of input is also be calculated later in order to compute the decision boundary of each attribute and produce a set of rules for classification. Furthermore, the deep learning methods including a simple convolutional neural network and ResNet18 been applied for process the original VehicleX image dataset. The results show that the explain grade rule can have a great performance in classification while deep learning may need to face the overfitting problem.

Keywords: Three-layer neural network, Fine-grained classification, Decision boundary, Explain grade, Convolution neural network, ResNet18, Multi-task Learning

1 Introduction

Fine-grained classification aims to distinguish between common superior and subordinate class, for example, distinguishing different Iris or vehicle, etc. usually, the subordinate classes are decided by domain experts using complex rules, which focus on subtle differences in specific areas. Deep learning method including convolution neural network, has been used in multi different task and has achieved dramatic improvements. However, the application of deep learning for fine-grained classification is still unsatisfactory even it has been wildly used in many different tasks and promoted the computer vision research. This is mainly because the difficulty in finding the information area and extracting distinguishing features[3].

1.1 Dataset

The data been used in the simple neural network is a 3D vehicle dataset generated by VehicleX[2]. It contains features extracted from ResNet which is pretrained on ImageNet. There are 45438 features with size 2048 in training dataset, 14936 features for validation and 15142 features in testing dataset. A total of 1,362 vehicles are annotated with detailed labels. Other detailed labelling includes vehicle orientation, light intensity, light direction, camera distance, camera height, vehicle type and vehicle colour are also provided for multi-task learning.

For the deep learning part, the original image datasets of VehicleX are provided, which means there are 45438 images in training dataset, 14936 images for validation and 15142 images for testing.

1.2 Problem description

This project aims to process features with different neural network and preform fine-grained classification and process the image dataset with deep learning methods. Since the feature data are all from 3D Vehicle data, the task for this project on machine learning part is to allocate these features to corresponding individual vehicle ID and type ID. According to previous work, developing an auto method identify the information area in the image is the key to do fine-grained classification [3]. Since the dataset we have are vectors of output activations from the final hidden layer, they do not have a semantic meaning like grade or color, etc. We will mainly use a three-layer neural network and a multi-task neural network to do the classification and try to use characteristic input to find which sets of features are indicative of a certain class. For the deep learning part, we will allocate the image data into corresponding type ID by ResNet18 and a self-written convolution neural network. At the end, the results from different method will be compared and discussed.

2 Method

The hardware basis of this project is: Memory: 16GB; CPU: Intel(R) Core(TM) i7-8559U CPU @ 2.70GHz; Operating system: macOS Big Sur 11.2.3; Software: python3.7

2.1 Pre-process

Since the dataset be provided includes features which already pretrained by Convolutional Neural Networks, so there is no need to consider original image color format or image transformation. However, the problem needed to solve is how to read data from files in dataset. All features in dataset are in .npy files and all labels are in finegrained_label.xml, if every time when using these features and labels, we need to read the .npy and match the corresponding label with .npy file name, that will take lots of times. In that case, all features and labels had been matched and restore in one .txt file. All features and labels in dataset are read from the new .txt file.

To normalize the original data, min-max scale is used after loading the data from the .txt file. The function of min-max normalization is as following, which means use the maximum value of x in each row and the minimum value of x in each row to normalize all values in this row into range [0,1].

$$x^i = \frac{x - min}{\max - min}$$

For the second task, the original image datasets of VehicleX are too large for personal computer. In this experiment, 5500 images are separated from the original image datasets and been allocate to training, valid and testing datasets. Since the deep learning task aim to apply type classification on these vehicle images, there are 300 images from each vehicle type in the new training dataset, 100 from each type in valid and testing dataset. These images are also be resized to 224 and normalized by 0.5 in RGB values.

2.2 Neural Network and Multi-task Neural Network

Neural Network is a group of algorithms which designed to identify potential relationships in target dataset by imitate the human brain process operates. One main advantage for neural network is, it can adapt to constantly changing inputs, in that case, instead of redesign the output standard, it can produce the best fitted result by itself[4].

Since all features in this dataset are 1D array, there is no way to apply convolutional layer in this Neural Network. Therefore, a simple three-layer Neural Network is first implemented in this task.

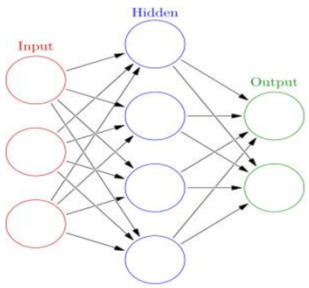

Fig. 1. Three-layer neural network

Different from the perceptron, three-layer neural network contain lots of input signs which multiplied by the weight, outputs signs, and there is activation function between input and output signs. Today, plenty of different activation function are available to use, including sigmoid, tanh, ReLU, softmax etc. These activation function can be part of three-layer neural network for processing non-linear problems[5].

The Neural Network has one input layer, one hidden layer and one output layer. The size of neurons in input layer and output layer is same as input size and output size. The hidden layer has 64 neurons in total. To get a batter performant, several different activation functions include Sigmoid function, tanh function and ReLU function have been tried in this Neural Network.

Logistic sigmoid function (or sigmoid function) has monotonous and continuous domain, and it is differentiable everywhere. It introduces the concept of probability to neural networks. But since the output value is not centered at zero,

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

it will cause the model to converge slowly. During the test, the sigmoid activation function was given up since the converge of it was extremely slow.

The advantage of ReLU activation function make it more suitable for this task. The ReLU function has more simple and efficient calculation, no exponential calculation compared to sigmoid. In the positive interval does not saturate, solve the problem of gradient disappearance. Also the convergence speed is faster for ReLU, about 6 times than sigmoid[6].

$$ReLU(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$$

The domain of tanh activation function is from -1 to 1, it is continued and differentiable. It is normally used after the hidden layer [7]. Compare to the sigmoid function, the convergence of tanh function is faster because of the slope value in the linear region near 0.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

The final network contains one ReLU activation function between the input layer and hidden layer and one tanh activation function after the hidden layer. SGD optimizer and cross entropy loss are also included in the network train, the learning rate of optimizer is 0.5.

To improve the simple three-layer neural network, a more complicated network is also used to solve multi-task classification including vehicle ID and type. The structure of this multi-task network is as figure 2, each layer in this network contains a BatchNorm1d, a full connection layer and an activation function. There are 3 layers for the vehicle ID classification, for the input size is 2048, hidden neurons are 64 and output size is 1363. After allocating the vehicle ID, the output of previous layer is used as input of type ID classification, for the input size is 1363, size of hidden neurons is 32 and output size is 11.

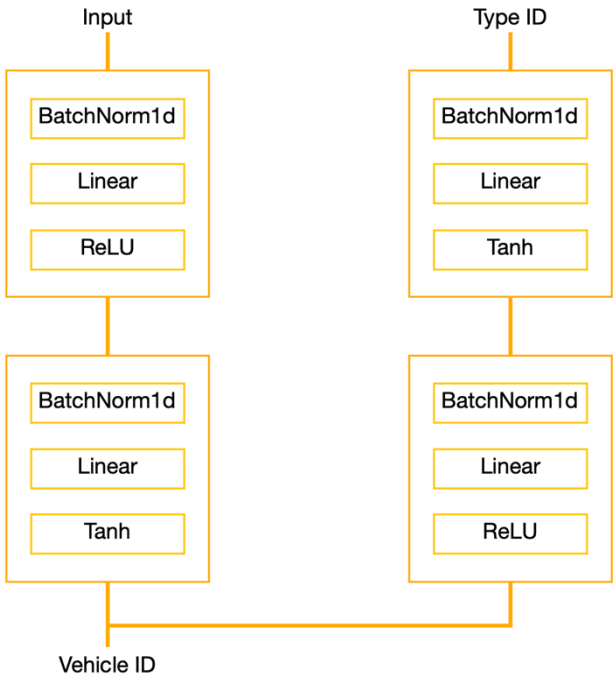

Fig. 2. Multi-task network structure

2.3 Characteristic input

Input pattern can be classified according to their impact on each specific output. The set of patterns can turn on the output are for producing characteristic ON pattern. Finding the characteristic pattern can be done by several statistical methods including arithmetic mean and median. In this project, the mean of vector components been used to describe characteristic input [8].

2.4 Decision Boundary

According to Yoda's work, the rate of change of an output neuron y_k with respect to an input neuron x_i is found by calculating the derivative dy_k/dx_i using the chain rule of differentiation [9]. Therefore, the rate of change is equal to the multiplication of derivative of activation function and weight. As we are using ReLU activate function instead of sigmoid, the derivative of activation functions is equal to 1 and the weights be used are uniform distribution by kaiming distribution, calculating dy_k/dx_i following the chain rule is uncertain. In this task, the rate of change is directed calculated from input dataset.

For one specific class (for example, label equal to 1), all training features with label 1 will be collect. Then compute the derivative of each column, that is, get the partial derivative of one output neuron and one single input neuron. To get the boundary, we need to get the maxima or minima value of each input neuron for label 1, in other word, we need to find the input neuron value with partial derivative equal to 0. Note that this way can only find the local maxima or minima value. For convenience, I simply use linear regression to fit the partial derivative function and get the boundary. At last, compare these boundary value to characteristic patterns to decide is it an upper boundary or a lower boundary. Therefore, I have one set of boundary value and one set of Boolean value to describe the boundaries. Both of these sets will become the rule of allocating class[8, 10].

2.5 Deep Learning Method

There are two convolutional neural networks been applied for performance contrast. The first one is a self-written simple convolutional neural network with four convolutional layers and two fully connection layers. In each convolutional layer, there is one conv2d method with fixed input and output size, one BatchNorm2d for data normalization, one Dropout method with parameter equal to 0.3 in order to avoid overfitting, and an activation function ReLU. The two fully connection layer are used at the end of the CNN for classifier the features into different labels.

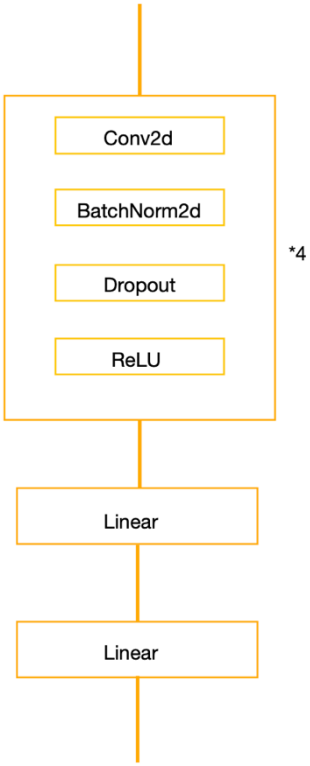

Fig. 3. Simple Convolutional neural network structure

The second model is the ResNet18 neural network, a very classical model in deep learning area. By using the residual learning block, the ResNet can get a faster convergence than directly learn the mapping between input and output [11].

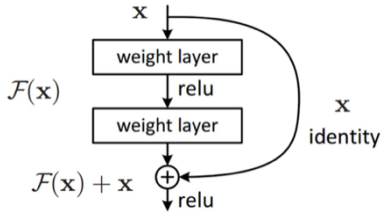

Fig. 4. Residual learning: a building block [11]

While training, since the dataset been used is quite small, the validation dataset was also added into the training function for early stopping. When the loss of validation stops to decrease and starts to increase, the training section will stop.

3 Result and Discussion

For a simple three layers with sigmoid function, it is very hard to get a satisfied result, when running 1000 epoch, the accuracy only reaches to more than 1%, and the loss reduces to 6.9 . After change the activation function to ReLU function and tanh function, when training 600 epoch, the training accuracy can reach to 2.87% and testing accuracy is equal to 2.66%. The running time of training and testing is 724.1s. However, the rule I got from decision boundary preform quite while. The average accuracy of each class is more than 96%. But the running time of explain grade is every long, and because of that, only 100 different labels had been tested, the running time of complete 100 labels is 1471s, so the proximately running time of whole dataset is around 1472*13.6s. The training accuracy of multi-task neural network on vehicle ID classification is 2.85% and the testing accuracy is 1.12%.

Table 1. Test accuracy and run time of different methods in vehicle ID classification

	Test Accuracy	Run Time
Three-layer Neural Network	2.66%	724.1s
Grade Rule	96.49%	1472.1s*13.6
Multi-task Neural Network	1.12%	7950.9s

The training loss and accuracy curve of different method are shown as following figures.

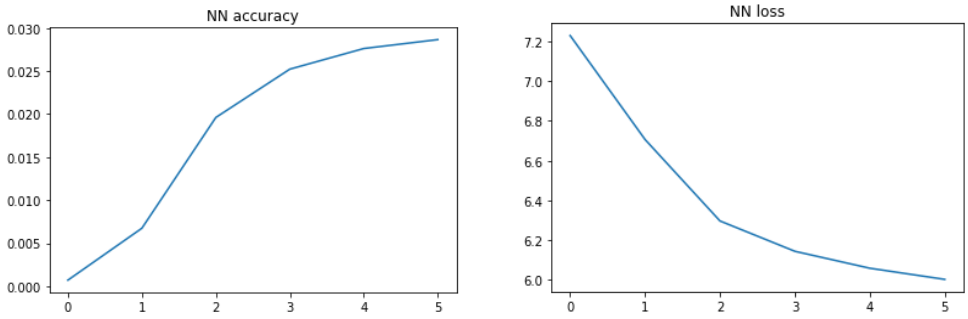

Fig. 5. Loss and accuracy of three-layer neural network, where the x-axis is hundred epochs

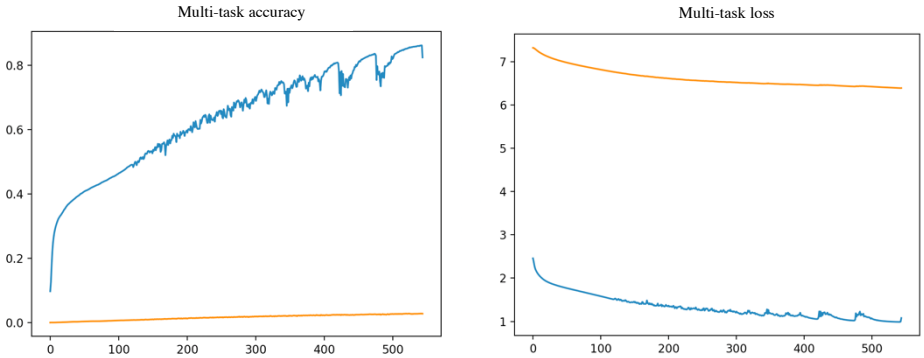

Fig. 6. Loss and accuracy of multi-task neural network: the orange lines are for vehicle ID and the blue lines are for type ID, the x-axis is epoch.

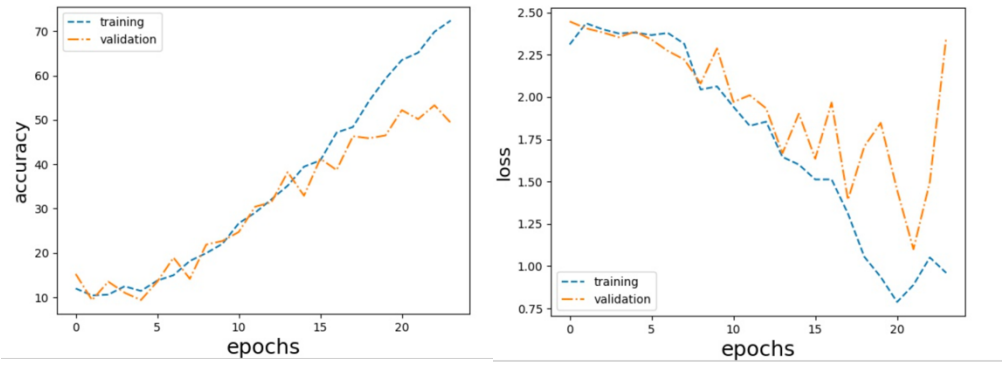

Fig. 7. Loss and accuracy of ResNet18

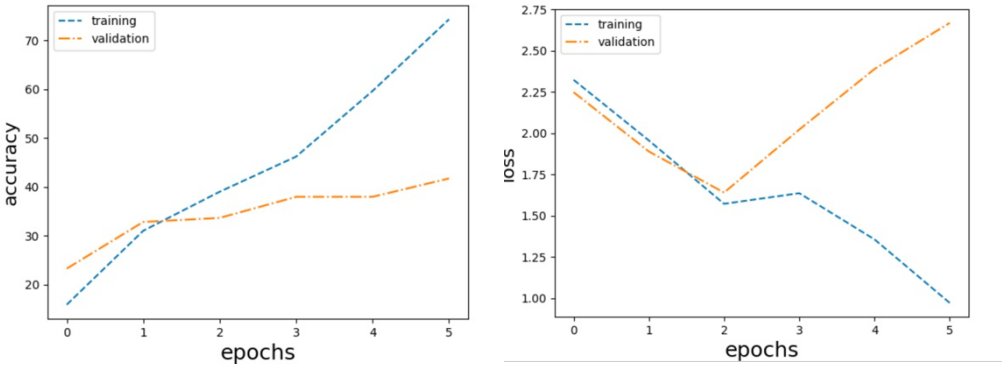

Fig. 8. Loss and accuracy of CNN

In the type ID classification, the training accuracy of convolution neural network is 74.33% and the training loss is 0.971. The training accuracy and loss of ResNet18 is 72.43% and 0.960. For the type ID part of the multi-task learning, the training accuracy and training loss is 86.12% and 0.989.

Table 2. Test accuracy and loss different methods in type ID classification

	Test Accuracy	Test Loss
CNN	34.74	2.337
ResNet18	40.81	2.370
Multi-task Neural Network	36.24	2.037

From the result of classify type ID, no matter using the multi-task learning or deep learning, the overfitting happened. Especially in the convolutional neural network, after only 2 or 3 epochs, the validation loss starts to increase, even with BatchNorm and Dropout methods. One possible reason for this phenomenon is the size of dataset are too small, however, since the hardware equipment (listed in the beginning of the method section) do not contain a GPU, it is impossible for the author to run more than 40000 images.

4 Conclusion

Before starting this project, several different models had been considered including CNN, LSTM(be give up because the dataset is not relay on time) and Decision Tree(to many index), at first, the three-layer neural network is a simple choice for this task. From the result, it is clear that a simple three-layer neural network cannot complete this task well since the number of classes is quite large. To get more results, a multi-task network been used to show the classification result of both vehicle ID and type ID from neural network, the results are also unsatisfied. The task should be done with a more complex neural network. The method of using gradient to find boundary and using explain grade to get the result is quite successful if not consider the running time, for some specific label, the accuracy could even reach to 99%. Feasible method for reducing the run time problem which can be down in the future is to reduce the feature dimension by removing the less influence feature or grouping the feature with same influence, etc. For the deep learning part, this report gives a demo result on training the original image dataset by convolutional neural network and ResNet18, since the data been used for deep learning is just a small part of the original datasets. Future work could be down by better hardware or resize the image to an even smaller size, the second way could lead to information loss.

References

[1] Turner H. and Gedeon T.D. “Extracting Meaning from Neural Networks,” Proceedings 13th Int. Con. on AI, Avignon, 1993

[2] Yao Y, et al. “Simulating content consistent vehicle datasets with attribute descent”. arXiv preprint arXiv:1912.08855, 2019.

[3] Yang Z., et al. "Learning to navigate for fine-grained classification." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[4] Wang S.C. "Artificial neural network." Springer, Boston, MA, pp. 81-100, 2003.

[5] Hecht-Nielsen R. "Theory of the backpropagation neural network." Neural networks for perception. Academic Press, pp. 65-93, 1992.

[6] Agarap A.F. Deep Learning using Rectified Linear Units (ReLU). p.2, 2018.

[7] Kalman B.L. and Kwasny S.C. Why tanh: choosing a sigmoidal function. IEEE. p.1, 1992.

[8] Gedeon T. D. and Turner H. S., “Explaining student grades predicted by a neural network.” In Proceedings of 1993 International Joint Conference on Neural Networks, 1993.

[9] Yoda M., Baba K. and Enbutu I. “Explicit representation of knowledge aquired from plant historical data using neural networks,” International Joint Conference on Neural Networks, San Diego, vol. 3, pp. 155-160, 1991.

[10] Engelbrecht A.P., Viktor H.L. "Rule improvement through decision boundary detection using sensitivity analysis." International Work-Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 1999.

[11] He K. Zhang X. Ren S. et al. “Deep residual learning for image recognition” P*roceedings of the IEEE conference on computer vision and pattern recognition*. pp: 770-778, 2016.