Distinctiveness Pruning Convolutional Neural Networks for Vehicle Classification

Ruotian Zhang

Australian National University
U7076906@anu.edu.com

Abstract. During the past decade, deep learning has seen groundbreaking developments in the field of AI. Convolution Neural network (CNN) is an important type of deep learning model and plays a critical role in many AI fields, such as computer vision and natural language processing. Generally, a CNN model will be designed with a large amount of layers and units, so there will be a huge number of parameters. Although such much parameters may result better performance, it is unclear whether all of the units have unique functionality. If some of units do not have positive effect for the performance of a model, removing them will be beneficial for reducing the amount of calculation of the model. To investigate this problem, we realize a pruning method, called distinctiveness pruning. This method uses pattern vector to represent the functionality of the hidden unit in input space (*i.e.*, on the training data) and removes some of the hidden units whose pattern vectors have high similarity. Therefore, using this method, we can evaluate the model before and after pruning to observe the effect of the pruned units on testing results. Meanwhile, we can also analyses how the functionality of unit changes during the training based on their pattern vector similarities. Using the ResNet50 as studying target, extensive evaluations are conducted on the VehicleX dataset. We study the influence of distinctiveness pruning on the recognition accuracy, the relationship between pruning ratio and model hyperparameters, such as training epochs and number of hidden units. We discuss experimental results in detail and draw several meaningful conclusions.

Keywords: Convolutional Neural Networks · Distinctiveness pruning · Vehicle classification

1 Introduction

The development of deep learning relies heavily on computing resources and data sets. With huge dataset, *e.g.*, ImageNet [6], and GPUs, deep learning models with a large number of parameters can be trained successfully. However, a large number of parameters means a larger model size and a large number of calculations, which may leads to bigger store space and slower running speed, *etc.,* problems. For example, In some scenes requiring real-time inference, such as person and vehicle re-identification, the inference speed of such a large model is particularly important. Therefore, lots of research focus on simplifying deep learning models. The purpose of pruning neural network is to remove unnecessary parameters in the existing network without affecting the performance, which helps to speed up the inference stage and improve the generalization of the model. Moreover, understanding the unique function of each neuron in a neural network is necessary because we can remove the units that have the same function.

To have a better understanding of the effect of hidden units, we implement a units pruning method, called distinctiveness pruning. The key of this method is to measure the functionality similarity between units. Based on the similarity value, a series of processing can be conducted on units to analyse their effect. For example, we can remove some units having similar functionality to observe their effect on the performance of the deep learning model. Meanwhile, we can observe the changes of the similarity between the units during training to analyse how units are trained with the procedure of training. Specifically, referring to existing work [1], we fist construct a pattern vector to represent functionality of the each hidden unit. Then, the we calculate similarity of the pairs of vectors by the calculation of the angle between them on training data. Finally, pruning units are conducted based on the similarity value. More details of the pruning method are introduced in our Section 2.4.

Experiments are conducted to find the relationship between pruning ratio and some hyperparameters, such as training epochs and number of hidden units. Besides, the effect of distinctiveness pruning on the testing accuracy is analysed. Based on the experimental results, we suggest that distinctiveness pruning can be beneficial if the number of hidden units is relatively big. The details will be introduced in Section 3.1.

2 Method

2.1 Data Processing and Data Augmentation

The image used in vehicle re-identification task is cropped from the whole image based on the bounding box of the vehicle. The crop region (bounding box) is from object detection algorithm or human annotation. Usually the size of the vehicle is different, so the bounding box images have different size. We first resize the images to 256x128 to avoid any errors caused by size difference. Then we use the mean and variance of ImageNet to normalize the data by the following formula:

$$z = \frac{x - mean}{std} \tag{1}$$

Data augmentation techniques can help to improve the model's generalization ability especially in deep learning. Therefore, we use random flipping and color jitters in the training stage to train a better model.

2.2 Model Selection

Researchers in deep learning field have designed many powerful network architectures such as VGGNet [7], ResNet [3] and GoogleNet [8], which are widely used as backbone networks in subsequent research. These models may have some variation of performance on different dataset, and usually we should conduct experiments to select the best model on our own task. In our vehicle classification task, the VehicleX[9] dataset is at a large scale, containing 45438 images of 1362 identities ,and is ideal for training large scale deep learning models. We investigate the performance of several CNN models on VehicleX dataset, such as VGGNet, Resnet18, Resnet50, GoogleNet and DenseNet121 [4], with the same hyper-parameter settings. The models are pre-trained on the ImageNet dataset and we further fine-tune all the the layers on the VehicleX dataset for 30 epochs with 0.001 learning rate. The experimental results are shown in Fig. 1.

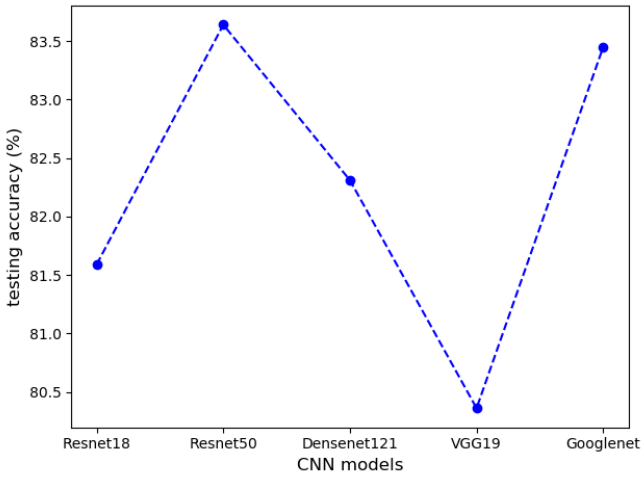

Fig. 1. The testing accuracy of different CNN models on VehicleX

It is clear that ResNet50 can achieve the the highest accuracy rate and we will use it as the backbone network to further study the effect of distinctiveness pruning. According to Gedeon and Harris[1], distinctiveness pruning is applied on fully connected layer, but the only fully connected layer is the classifier in the Resnet50 model. It is not wise to remove some of the distinctive visual feature because this will seriously affect the classification accuracy of the model. Therefore, We modified the Resnet50 model to better meet the needs. We reshape the backbone network to our new model with slight modifications. The structure before the original global average pooling (GAP) layer is maintained exactly the same as the backbone model. After GAP layer, we insert a fully connected layer on which we will apply pruning and we call it hidden layer for convenience in this paper, then next layer is the classifier. The architecture of our model are shown in Fig. 2.

Fig. 2. The network architecture

2.3 Hyperparameters

Hyperparameters have a significant impact on the training and inference of the model. We conduct experiments to study the influence of hyperparameters, such as learning rate, batch size, optimizers and so on. We try different combination of learning rate and batch size ,and we find that under this setting (learing rate=0.001, batch size=64, optimizer=Adam[5]), our model can achieve highest accuracy rate according to the experiments results. However, for our task, the most important parameter is the number of neurons of the layer we will apply distinctiveness pruning on since it is closely related to our research objectives. We train 6 models for 50 epochs with different number of neurons of the hidden layer, and the experimental results are shown in Table 1.

# unit	accuracy(%)
32	81.45
64	82.23
128	83.61
256	84.26
512	84.91
1,024	84.56

Table 1. Testing accuracy of models with different number of hidden units (# unit)

From the table we can find that our model gets the best performance with the number of hidden units equal to 512. The highest accuracy rate is 84.91%.

2.4 Distinctiveness Pruning

There are many network reduction methods based on relevance, contributions, sensitivity, *etc* and we mainly study the distinctiveness pruning in this paper.The distinctiveness of hidden units is determined from the unit output activation vector over the pattern presentation set[1]. According to Geoden and Harris, a neuron is redundant and should be removed if its pattern vector is similar to other neurons because the are of same functionality. In this paper, we perform distinctiveness pruning on the hidden layer for the following steps[1].

- After training the model, we fix the weights. Input all the patterns to get the output of the hidden layer, thus we get the hidden pattern vector for every unit.
- Normalize every element of the pattern vector between 0 to 1.
- Subtract 0.5 to scale the elements of pattern vectors to -0.5 to 0.5. The angular range now becomes 0-180° rather than 0-90°.
- Calculate the angles between every pairwise combination of the pattern vectors.
- If the angle of their pattern vectors is less than 15°, we consider they are sufficiently similar. Remove either of the two units and the remaining unit will retain the weights of the removed one.
- If the angle of their pattern vectors is greater than 15°, we consider they are complementary. Remove both of them.

3 Results and Discussion

In this section, we study the effect of distinctiveness pruning on classification accuracy and the relationship between pruning ratio and several factors. We report the experimental results on the VehicleX dataset. Meanwhile, several interesting findings are got about the distinctiveness pruning algorithm.

3.1 Effects of pruning on classification accuracy

In this section, we study the effects of distinctiveness pruning by comparing the testing accuracy before and after applying pruning on the hidden layers. We fix the training epochs to 50, and trained 6 models with different number of units of the hidden layer. We expect that distinctiveness pruning technique will be beneficial when the hidden units number is getting bigger since there will be more redundant units. Intuitively, the performance of the model should become better after pruning these unit. The results are shown in Fig. 3.

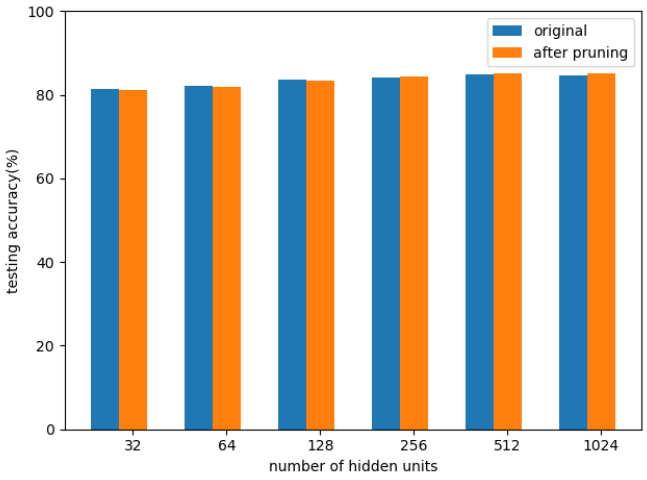

Fig. 3. Testing accuracy before and after applying pruning

It is obvious that the experimental results prove that our hypothesis is correct. When the number of hidden units increase, there will be more units of identical functionality. We observe that the number of pruned units is 520 when we set the hidden number to 1,024, which means more than half of the units are judged to be undesired by the distinctiveness pruning method. This result is very impressive because the computations are reduced by half, but the performance improve slightly. However, this approach may not work well for a small number of hidden units since every unit will be trained to have its own unique functionality to fit the big amount of data. From Fig. 3, we can find that the accuracy rate drops after pruning on the model with less than 128 hidden units.

3.2 Pruning ratio and training epoch

Pruning ratio can, to some extent, reflect whether the training process is effective. Ideally, the pruning ratio should become smaller as training progresses since every unit will gradually have its own unique functionality. We conduct experiments to investigate the relationship between training epoch and pruning ratio on our model. We set the number of hidden units to 512 and train the model with different epochs. The results are shown in Fig. 4.

From the figure, it is evident that training epoch and pruning ratio are negatively correlated in general.At the beginning, the weights are randomly initialized and the units do not have independent functionality. Therefore, it is more possible that the two units are similar. When the model converges, the number of units with the same function will be greatly reduced.

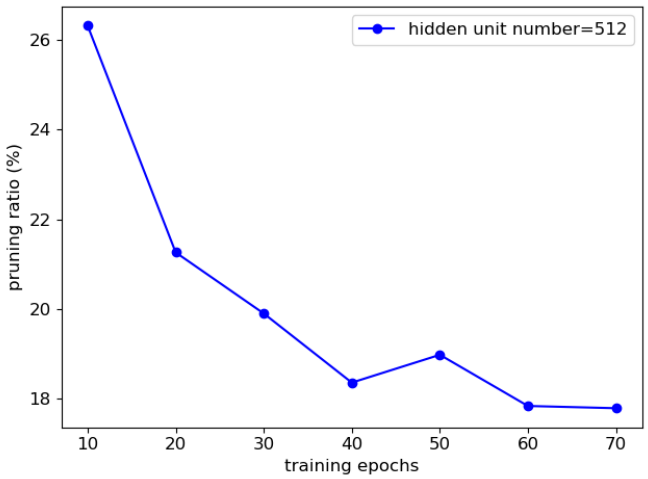

Fig. 4. The relationship between training epochs and pruning ratio

3.3 Pruning ratio and number of hidden units

Model selection is a very important part of machine learning. If we can not choose the right model, it usually causes some serious problems, *i.e*, under-fitting and over-fitting[10]. Under-fitting refers to a model that can neither model the training data nor generalize to new data, and over-fitting refers to a model that models the training data too well but cannot generalize well on other data. Both of them will result in the model not being able to have good generalization ability. We think distinctiveness pruning theory can give us some enlightenment on model selection. If the pruning ratio of a hidden layer is relatively low, we believe that every unit performs its duty and no units are redundant. We may choose a larger model to learn from the data. On the contrary, if a large number of units are judged to be similar, the model is at risk of over-fitting. We conduct experiments to study the relationship between number of hidden units and pruning ratio, and the results are shown in Fig. 5.

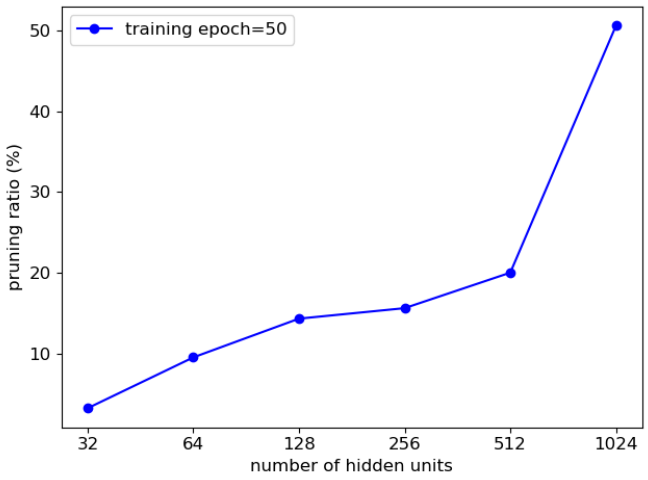

Fig. 5. The relationship between number of hidden units and pruning ratio

We can find from the figure that the pruning ratio generally shows an upward trend as the number increases, which is in line with our expectations. From the figure, we can find that when the number of hidden units is 32, the pruning ratio is about 3.15%. But it becomes nearly 50% when the number is 1,024. We believe this is reasonable. When the number of hidden units is small, every unit must have an unique functionality to fit the data adequately and we can not remove any of them. When the number of hidden units is relatively big, we can prune the units with

similar functionality without affecting the inference ability seriously. For example, the testing accuracy of model become actually higher after pruning nearly half of the units when the number of hidden units is 1,024.

4 Conclusion and Future Work

In this paper, we implement a network reduction method, called distinctiveness pruning, to investigate the functionality of hidden units. Experiments are conducted on the VehicleX dataset to study the relationship between pruning ratio and model hyperparameters,such as training epochs and number of hidden units. We also discuss in detail about the influence of distinctiveness pruning on the recognition accuracy. We argue that the distinctiveness pruning method can only be beneficial when the model has relatively big number of hidden units. Otherwise, although the inference process is accelerated, the classification accuracy will become worse.

In this paper, we experimentally test the standard threshold proposed by Gedeon, but it is still unclear whether the standard threshold is the best. In the future, we can study the theory of this problem, or we can do experiments to find the optimal threshold. Moreover, the results show us that the pruning ratio is positively correlated with the number of hidden units. Can we design an algorithm to change the threshold automatically according to the the number of hidden units so that high accuracy can always be guaranteed after pruning? In addition, the current approach to form the pattern vector may not be suitable for large dataset. For example, the training set contains 45,438 images and the pattern vectors are of 45,438 dimension in VehicleX. It is not only computationally expensive to calculate the angle between to high dimensional vectors, but also may be damaged by the curse of dimensionality. According to the curse of dimensionality theory, two unit random vectors in high dimension are almost orthogonal with high probability[2], which means the current pruning method may be invalid if the training set is huge. We think a feasible method to reduce the dimension of the pattern vector is that randomly sample from training patterns to form a lower-dimensional pattern vector.

References

1. Gedeon, T., Harris, D.: Network reduction techniques. Proceedings International Conference on Neural Networks Methodologies and Applications **1**, 119–126 (1991)
2. Gorban, A.N., Tyukin, I.Y.: Blessing of dimensionality: mathematical foundations of the statistical physics of data. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **376**(2118), 20170237 (2018)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
4. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
9. Yao, Y., Zheng, L., Yang, X., Naphade, M., Gedeon, T.: Simulating content consistent vehicle datasets with attribute descent. In: ECCV (2020)
10. Zucchini, W.: An introduction to model selection. Journal of mathematical psychology **44**(1), 41–61 (2000)