Vehicle Type Classification through Synthetic Images Using Convolutional Pruned Neural Network

Amogh Dhaliwal

Australian National University
{Amogh.Dhaliwal@anu.edu.au}

Abstract. Image Classification refers to the task of assigning a label to an image, its one of the most popular tasks in computer vision/deep learning and has wide application. The aim of this paper is to create a image classification model by fine-tuning ResNet-18 pre-trained on ImageNet. We apply two different pruning techniques Magnitude pruning and Distinctiveness pruning to compress the model and compare the performance of these methods, through experiments we determine that Magnitude pruning performs better by removing as much as 70% of the neurons from the network with minimal penalty to accuracy. From our results we can conclude that we can compress ResNet-18 upto 70% with the similar accuracy as the original mode.

Keywords: Image Classification · Pruning · Transfer Learning · CNN

1 Introduction

Image Classification is the task of categorizing an image into a class. This is widely applicable for many applications including but limited to automated image organization, face recognition and image colorization. This can also be extended to video classification for applications such as autonomous driving. Image Classification can be achieved by assigning vectors of pixels of an image into a class using specific rules. More recently neural networks have been highly successful in achieving this.

We will be building a model to classify each image into its vehicle type. This a complex problem due to many types of varied vehicles and the shapes of similar vehicle looking different due to the angle and orientation of the vehicle in the image. Due to this we require a large and diverse dataset of vehicle images. For such a task obtaining real image data is time consuming and sometimes unfeasible. Due to this we will be using the Synthetic VehicleX Dataset [9]. This dataset uses the graphic engine Unity to simulate a large amount of training data with detailed labels which approximates real images.

In many real time applications, the task of image classification requires both accuracy and speed. Typically for image classification the size of the model is a good indicator of its performance. Our network needs to be large enough to learn from the massive synthetic dataset and at the same time be robust. Naturally, using a Deep Neural Networks offer the accuracy we require but in cost of computational cost and storage. One method to address this is by pruning weights, which enables us to remove redundant nodes making our model lighter and faster with minimal penalty to accuracy [4]. However it needs to be noted that large models retain their accuracy after pruning only if they have been trained properly before pruning [3]. If a network is reduced in size too quickly before being trained properly it will cause significant drop in accuracy. For our network we will be using ResNet-18[7] pre-trained on ImageNet[2]. Residual neural network (ResNet) [7] is a deep learning model based on residual learning. ResNet-101, ResNet-50, and ResNet-18 are all versions of the ResNet architecture.

The paper by T.D. Gedeon & D. Harris [4] uses weight pruning on a neural network used for compression of images, for this paper we will be extending the application of pruning to that of image classification using a DNN. We will be pruning through two methods, by Magnitude and by Distinctiveness and comparing the performance of these two methods. The goal of this paper is to experiment the effectiveness of pruning on a DNN using the Synthetic VehicleX Dataset [9]. In order to achieve the best performance, we have tried a combination of hyper parameter settings and data pre-processing techniques, after achieving our optimal model we applying pruning and discuss the results.

2 DataSet

Synthetic VehicleX Dataset is generated using synthetic dataset generator named VehicleX Engine [9]. The synthetic images are rendered using the Graphics engine called Unity, using a python API which interacts with Unity provides us with detailed labels such as vehicle orientation, light direction, light intensity, camera height, camera distance, car type and color. Realistic backbone models and textures allow the graphics engine to generate accurate and varied images. The dataset contains 45,438 images for training, 14,936 images for validation and 15,142 images for testing along with detailed labels. The dataset contains a total of 11 types of vehicles namely sedan, suv, van, hatchback, mpv, pickup, bus, truck, estate, sports car and RV, with each image taken with varied attributes. For this paper we will be classifying the image into one of these types of vehicle.

2.1 Pre-processing and Feature Extraction

As stated by Carlos Vladimiro Gonzalez Zelaya [5], many of the decisions that affect the model's predictive behaviour are made during data pre-processing, hence it cannot be understated the importance of cleaning and processing the data before feeding to our neural network. Naturally pre-processing varies from the chosen dataset. Ideally a dataset is well-formed and requires little to no processing, but in many cases datasets deal with missing values, normalization, scaling, reduction of dimensionality and unbalanced datasets. For the VehicleX dataset [9] we will be normalize each image so the pixel values are in the range of 0-1 and converting each value to contain decimals (float). The given labels for each input are cleanly organised in a xml file without any inconsistencies or missing values.

3 Method

3.1 Model

The model used is ResNet-18 [7] pre-trained on ImageNet[2]. ResNet-18 [7] is a convolutional neural network which is 18 layers deep. ResNet-18 [7] originally is designed to take an image of input size 224x224 with 3 channels. However our images have dimensions of 256x256 with 3 dimensions. This is resolved due to the implicit cropping which takes place on the average pooling layer before the last FC layer. In some cases this crops out important information and drops the test accuracy of the model. However in our case since our original image dimensions are not significantly larger than the cropped image, we do not experience any drop in accuracy. For the loss function the cross-entropy loss performed best in a multi-classification problem. We trained the model using mini-batch of 100 samples for efficiency and memory management since we have abundance of training/testing data and it also provides more robust convergence. We fine-tuned the model on our training set using Adam optimizer with learning rate of 0.001. The choice of these hyper parameters was determined by the highest validation accuracy of 38.9%.

3.2 Pruning

In recent years, the applicability of neural networks has skyrocketed. With more complex problems and larger datasets the size of the networks is rapidly increasing. And with limited computational resources and storage, the task of reducing network size is becoming an important one. In many cases the size of the network is larger than required, this is because there is no method to find the minimum size of a network. One way we can reduce the size and complexity of the model is through pruning.
There are multiple ways to prune a network such as by neurons[4], weights [6] and filters[8]. However as stated by Davis et al. [1], the main problem is the lack of standardized benchmarks and metrics to compare these methods. In this paper we will be pruning our network based on magnitude and distinctiveness.

Magnitude Pruning is achieved by removing specific weights from our network, this is achieved by removing percentage of the lowest weight magnitudes. For instance if percentage is equal to 10%, then magnitude pruning will remove the lowest 10% of weights from the network.

Distinctiveness Pruning is achieved by removing entire neurons from our network, This is achieved by creating a activation vector v with the same dimensions as the number of patterns in our training set for each of our hidden neuron [4]. In equation,

$$v_n(i) = n(x(i)))$$

n(x(i)) is the output of neuron n given training data x(i) which is the i'th training example. The activation vector v_n contains output for each neuron for each training data. For instance, v_5 is a activation vector containing outputs of neuron n=5. After the activation vector is constructed, the angles between these vectors can be calculated in pattern space and the similarity between them can be determined. The neurons then can be removed depending on some threshold value(s) of the angle. The value(s) for the threshold must be determined experimentally corresponding with the accuracy of the model. Angles closer to 0 ° are similar in functionality and can be removed, the weights attached to the removed neuron are added to any other neuron which is not removed.

3.3 Evaluation

The VehicleX dataset[9] is pre-partitioned into training, validation and test set, with validation set containing 14,936 images and testset containing 15,142 images. The model is trained on the training set for a maximum for 50 epochs. The model is then saved, and then tested on the validation and test set. The model is then pruned using magnitude and distinctiveness pruning, fine tuned and retested on the validation and test set.

4 Results and Discussion

4.1 Results with magnitude pruning

Our base model has test accuracy of 38%. We experimented with different percentage of removed neurons in our network and we display our results in Table 1. As we can see from Table 1 we were successfully able to remove 70% of weights with minimal change in accuracy. Another interesting observation is that in some cases, after pruning our test accuracy even increases above that of the base model. This could be due to the large size of the base model which is more likely to overfit, while our smaller pruned network maybe generalising better.

4.2 Results with distinctiveness pruning

As in the previous case, our base model had test accuraacy of 38%, we pruned the base network with different threshold angles, the results are displayed in Table 2. As we can see from Table 1 we were successfully able to increase the threshold angle to 5° after which the accuracy dropped significantly.

Table 1. Model performance with magnitude pruning

Percentage	Val Accuracy(%)	Test Accuracy(%)
0	38.9	38.0
30	42.9	37.7
50	40.9	38.4
70	39.5	37.7
95	36.9	34.4
99	18.4	20.4

Table 2. Model performance with distinctive pruning

Threshold($^\circ$)	Val Accuracy(%)	Test Accuracy(%)
0	38.9	38.0
5	38.9	38.0
10	28.1	31.7
15	26.1	29.8
20	24.8	24.5
25	22.8	22.5
30	21.4	19.8

4.3 Magnitude vs Distinctiveness pruning

From Table 1 and 2 we can conclude that magnitude pruning is more effective than Distinctiveness pruning, removing upto 70% of nodes with minimal penalty to accuracy, in some cases magnitude pruning also gives us higher test accuracy than even the original base model. The accuracy of the model using distinctiveness pruning starts to drop drastically with threshold angle as low as 10°. In conclusion, Global magnitude pruning is more effective in removing nodes when using ResNet-18[7] on the Vehicle-X dataset [9].

5 Conclusion and Future Work

In this paper we were successfully able to create a deep learning model to classify synthetic vehicle images into their vehicle type. Further more, we were able to show the effectiveness of two different pruning methods(magnitude and distinctiveness) on our network, reducing number of nodes by upto 70% using magnitude pruning with similar accuracy to the original model, and in some cases even increasing the accuracy. We also compared the two pruning methods and concluded that for our network, magnitude pruning is significantly more superior to distinctiveness in pruning our network.

In the future, We will investigate in more detail why distinctiveness pruning drops in accuracy more drastically than compared to magnitude pruning, understanding this we could develop better methods to prune our network. We could also further extend the application of our techniques discussed in this paper by using it for video classification using each frame in a video. More advance pruning techniques will allow us to reduce computational time and resources and will also enable us to run more complex networks on devices with limited computational resources.

References

1. Blalock, D., Ortiz, J.J.G., Frankle, J., Guttag, J.: What is the state of neural network pruning? (2020)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
3. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks (2019)
4. Gedeon, T., Harris, D.: "network reduction techniques," proceedings international conference on neural networks methodologies and applications, amse, vol. 1, pp. 119-126, san diego (1991)
5. Gonzalez Zelaya, C.V.: Towards explaining the effects of data preprocessing on machine learning. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE). pp. 2086–2090 (2019). https://doi.org/10.1109/ICDE.2019.00245
6. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
8. Luo, J.H., Wu, J., Lin, W.: Thinet: A filter level pruning method for deep neural network compression (2017)
9. Yao, Y., Zheng, L., Yang, X., Naphade, M., Gedeon, T.: Simulating content consistent vehicle datasets with attribute descent (2020)