Fine-Grained Classification Task Using Sparse Autoencoder And Casper algorithm

Zeji Hui

Australian National University

Abstract: This paper applies and analysis a neural network algorithm Casper to evaluate a large dataset VehicleX. Cascade algorithm is an advanced version of Cascade Correlation algorithm, has proved to be a powerful model for training neural networks. The Cascade algorithm has a constructive architecture to build the network during the learning process along with the PROP optimizer. The dataset VehicleX is a fine-grained synthetic dataset with a large number of classes: it contains 1362 cars and each data has 2048 features. We propose a Autoencoder method to reduce the dimension of the data for a more efficient training, specifically we test the encoder to find a best dimension reduction scale. We implement Casper and Sparse Autoencoder to train the model and report the result compared with a baseline result and the dataset paper result. Casper is shown to get a relative poor result compared to other methods. At last we talk about the limitation of the Casper and the future work.

Keywords: Fine-Grained Classification; Autoencoder; Casper Algorithm; Neural Network;

Introduction

The Fine-Grained Classification Task (FGCT) focuses on differentiating between object classes of subtle difference, such as species of animals, plants, and identifying the plate number or models of vehicles. In this paper, we explore the Vehicle Re-ID task of identifying the same vehicle across multiple cameras. The data has subtle difference between in each other, like camera angle, lightness, and camera position, we need to identify a car with these multiple images. As a machine learning method, after the model was trained, one of its application scenarios is the traffic control, since it can identify one car with images captured by different cameras in different streets to track one special vehicle. It can also be applied for Artificial City, hunt down a criminal and automotive marketing.

To solve this problem, Neural Network techniques has emerged in recent years as powerful tools for learning feature from data properly and classing the object in a high accuracy [1]. There are many structures for the Network, like Bilinear Neural Network, ResNet [] and TransFG. In these structures the last layer of the net like full connected lay, is always served as classifier, deal with the features extract from the net and class the object. Since the dataset is a feature dataset, we just choose to implement a classifier without feature extracting.

The dataset apply in this paper named VehicleX[2] which is a large-scale synthetic dataset generator engine. Compared to the real word dataset like VehicleID [7], it could generate 1362 classed of cars with infinite images. Specially it also contains the attribute labels (*e.g* lightness). Show in the Table 1. In this paper we select more than 50000 images as a dataset.

Dataset	IDs	Images
VehicleX	26328	222,629
VehicleID	1362	infinite

Table 1

In our experiment, we propose to design a network named Casper to solve this fine-grained classification problem. Casper is a Neuron Network structure, in the experiment, we implement the net, test and evaluate the result, compare to a benchmark method and the result in the original paper [3]. Since the dataset is a large dataset, we try to reduce the dimension of the data with the Autoencoder method.

Method

Dataset preparation

Before implement the algorithm and train the model, we need to prepare the dataset.. In this paper the dataset offered by Vehicle-X, which is a publicly available 3D engine [2]. A total of 1,362 vehicles are annotated with detailed labels in this dataset. However, in this task the dataset is just objects with features extracted from ResNet[8] which is pretrained on ImageNet. It contains 45,438 images for training, 14936 features for validation and 15142 features for testing. The dataset has already been normalized ie. each element of the data is a float number in [0,1] and flattened in shape [1,2048]

However, the label of the dataset for sets train, validation, test are in one file and we need to extract the label information and store in three separate txt files , each file includes the path of data and the label of the data, ie, the vehicle ID of the vehicle. We rewrite the Dataset class in Pytorch and load our data with batches and tune the batch size hyperparameter to match our GPU memory since it is a relative huge dataset with more than 50000 data elements.

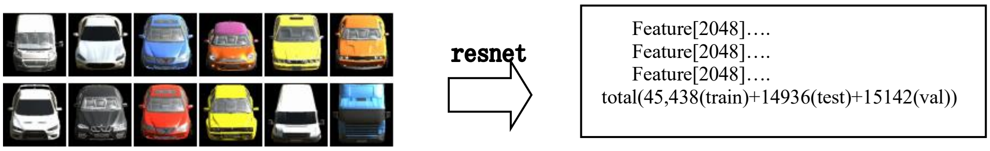


Fig.1 the dataset structure

Casper Algorithm

Casper is an ancient and dynamic neural network created by Treadgold and Gedeon in 1997. Casper is an improved version of Cascade Correlation which is a Neural Network structure [5]. But Casper using different learning rates for the 3 different sections of the network and RPROP optimizer ((Riedmiller and Braun, 1993; Riedmiller, 1994). The structure of the Casper is dynamic, starts with a single hidden neuron, and increase hidden neurons successively.

RPROP is used to update the wight value each time a hidden neuron is added. When a new neuron addition, the initial RPROP learning rates set according to three different region in network. however, such that when a new neuron is added the initial learning rates for the weights in the network are reset to different values, depending on the position of the weight in the network.:

– Region L1: Weights connecting to new neuron.
– Region L2: Weights connected from new neuron to output neurons.
– Region L3: Remaining weights

One of the important feature of the Casper is that no neurons are frozen, Therefore, we can get benefits in the following ways. The network modifies the old weights, which will happen, although the initial speed is slower Than the weight of the new unit connected to it. So Casper retained weight freeze and Cascor's related technology while removing saturation problems caused by related measures, and permanent installation underperforming neurons caused by neuron weight freezing.

In order to make use of the weight decay to improve the generalization properties of the network structure, the Casper follow formular as error gradient. HEpoch in the equation means the number of epochs elimated since the addition of the last hidden neuron, sign returns the sign (positive/negative) of its operand, and k is a user defined parameter which effects the magnitude of weight decay used.

$$\delta E/\delta w_{ij} = \delta E/\delta w_{ij} - k * \text{sign}(w_{ij}) * w_{ij}^2 * 2^{-0.01*HEpoch} \quad (1)$$

To control the size of the net, Casper introduce a new parameter time period to check the RMS error. The RMS error must fall by at least 1% of its previous value in a given time period

$$\text{time period} = 15 + P*N \quad (2)$$

where N is the number of currently installed neurons, and P is a hyperparameter.

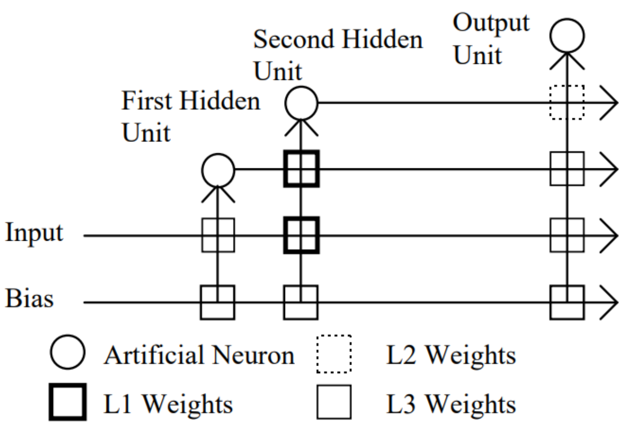


Fig2. The Casper architecture: when the second neutron been added Compared to Cascor this structure has no neurons been frozen.

Sparse AutoEncoder

When train the model with Casper, with more neurons added to the network, the network becomes very complex so it will cost a lot. In order to solve this problem we propose the Sparse Autoencoder [9]. An autoencoder is a neural network that could use data compression like PCA. It has an internal (hidden) layer that describes a code used to represent the input, and it is constituted by two main parts: an encoder that maps the input into the code, and a decoder that maps the code to a reconstruction of the input. In this paper we just need the encoder part.

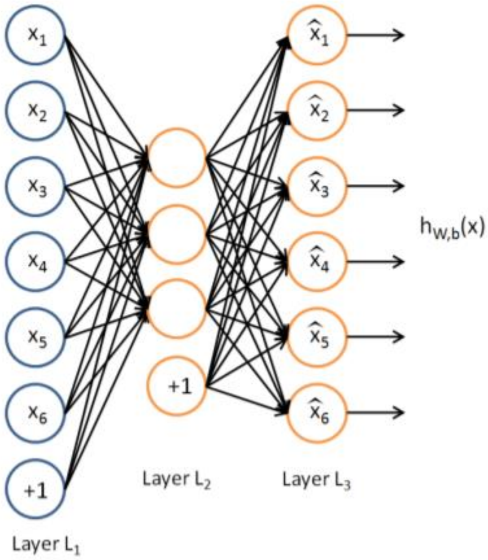


Fig3 structure of autoencoder

In this task ,the hidden neurons is larger and so we impose a sparsity penalty on the hidden units loss, then the autoencoder will still discover interesting structure in the data. **a** denotes the activation of hidden unit **j** in the autoencoder and we get be the average activation of hidden unit j:

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right] \qquad \hat{\rho}_j = \rho$$

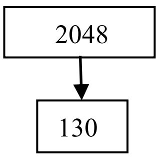
So we get ρ is a sparsity penalty item, we set it a small value close to zero (say ρ = 0.05). To satisfy this constraint, the hidden unit's activations must mostly be near 0. To achieve this, we will add an extra penalty term to our optimization objective that penalizes ˆρ deviating significantly from ρ:

$$\sum_{j=1}^{s_2} \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j}$$

Results and Discussion

Sparse Autoencoder

With experiment many times to the baseline methods and the Casper methods, under the premise that the dimension reduction will have subtle influence to the accuracy, we find that the minimum hidden neuron of the autoencoder is 130, so our initial structure is:



also, we set the parameters: BETA = 3 and ρ = 0.05. with other parameters for the network are identical, the Casper network without sparse autoencoder could get maximum 4.7% and 4.4% without autoencoder.

Benchmarking result

To test the performance of the Casper algorithm, we analysis and compare the effectiveness and accuracy against the baseline method Multi-layer Perceptron net structure. The baseline structure has following hyperparameters: three full connection layer, 1500 neurons, SGD optimizer with learning rate 0.026 and the result is Figure 3.

And for the Casper algorithm.In addition, a constant parameter 0.0001 was create to the derivative of the sigmoid in order to solve the 'flat spot' problem. And we use hyperbolic arctan error function, and the standard sum of squares error function was used for regression problems. All weights were initialised to random values. Training of the initial network used the initial update value Δ0 = 0.2. The values of L1, L2, and L3 were set toΔL1, ΔL2,ΔL3 respectively. The remaining parameter values, k (the weight decay value) and N (the training length) the hyperparameter table is as followed (table 1):

P	FCneurons	k	N	ΔL1	ΔL2	ΔL3
1	100	1e-6	1	0.001	0.2	0.005

Table.2 hyper parameter of Casper

To get fully converged, after 200 epochs, the baseline model reap a roughly 80% train accuracy and 10% test accuracy. As show in figure 3. And beside that, the validation dataset is apply to validate the model, it get 8.9% accuracy after the 200 epochs and maximum 9.7% accuracy at epoch 86.

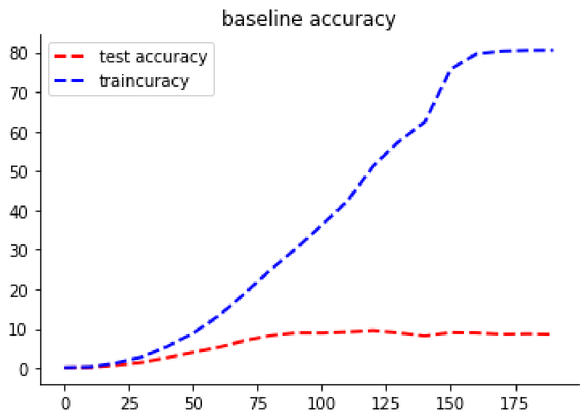


Fig.4 Baseline accuracy

Same as baseline methods the Casper converge process is Figure 4. And beside that, the validation dataset is apply to validate the model, it get 1.3% accuracy after the 200 epochs and maximum 4.2% accuracy at epoch 48 with 1 hidden Neurons.

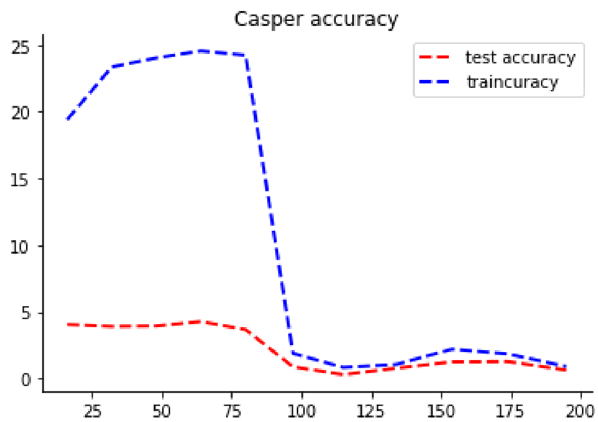


Fig.5 Casper accuracy

Compare with VehicleX paper:

In order to evaluate the performance of our network, we would like to compare our result to the VehicleX paper[2]. However in [2] the model is very different for the validation and test dataset, it combines the synthetic dataset with real dataset. And also it use rank-1 and mAP as the evaluation parameter, however we use accuracy as the evaluation parameter. But we can find that when train the model only in VehicleX dataset, it can get mAP up to 18.6, which can be a reference.

Discussion

The Autoencoder could enhance the efficiency by reduce the dimension of the data under the prime that it will not reduce the accuracy significantly. And the sparse penalty perform well to the network.

As we can see, the accuracy of Casper compared to the baseline methods is poor, it gets less accuracy. Although there has been further attempt with other datasets, as mentioned in the start of the paper the vehicle Re-ID task is relative difficult compare to the traditional classification problems, since it has huge number of classes. So it cannot conclude that the poor result is only depends on the Casper algorithm.

When analysis the influence of neuron number We can see that the accuracy of the train dataset reap the max at with the 1 hidden neuron and the max train accuracy reduce when a new neuron is added. And when observe the Figure 4, we can find that when a new neuron added, the model struggle to converge, it get a worse result. In the original paper the author just list the updated learning rate and the checking point epoch number, so it would be good if experiment other hyperparameters or algorithm to determine when the new neuron should been added.

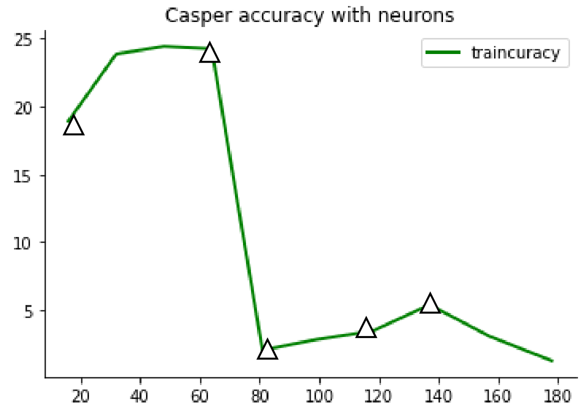


Fig.4 Casper train accuracy with neuron added

Setting aside the results from the original paper, both the baseline model and the Casper algorithm get a not good class accuracy result. It is less than 10%, so for the fine-grained classification task there need more work to reap a better accuracy.

The limitation to the Casper algorithm can be in different aspect. Firstly, the network may not train enough due to the limitation of the large dataset and it may perform well in some dataset with less data or less class numbers. Secondly with neurons added to the network, the network becomes very complex and will cost a lot when the data set is large.

Conclusion and future work

In our experiment we tend to apply the Casper algorithm to the FGCT problems to class the vehicle into more than 1000 classes of vehicles, and the Casper model show the basic ability to this task, however the Casper model performs relative poor compared to the baseline network. In this datasets, adding new neuron to the net cannot improve the performance of the accuracy. And our Autoencoder reduced the dimension of the dataset without reduce the accuracy.

The future work in can be in two part. First is to improve the accuracy of the dataset, The dataset offered is just features in this task, so we can apply other new methods to extract the features of the raw vehicle images find the relationship between the multiple similar images, and then apply some methods to enlarge the difference between them, so as to perform well when apply the classification methods. Beside that, the dataset offers multiple label of the images, like the vehicle type or the lightness ,we can apply this label to multiple label training, to get a better result.

And the second is to improve the performance of the Casper algorithm, as more neuron added to the net the structure of the net becomes very complex and the cost of the computing resource increased suddenly, so maybe we can find a method to drop out some connection to improve the efficiency and avoid overfitting.

Also as we mentioned above some of the dropout method can be used to enhance the performance of the network.

References

1. Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195--197 (1981)
2. Simulating Content Consistent Vehicle Datasets with Attribute Descent. Yao, Yue, Zheng, Liang, Yang, Xiaodong. Naphade, Milind. Gedeon, Tom. ECCV (2020)
3. A Cascade Network Algorithm Employing Progressive RPROP. Treadgold, N.K., Gedeon, T.D. (2005)
4. The cascade-correlatoin learning architecture. Fahlman, S.E., Lebiere, C. Advances in Neural Information Processing II, pp.524-532, Morgan Kauffman, San Mateo (1988)
5. The cascade-correlation learning architecture.Fahlman, C Lebiere (1990)
6. Zheng, Zhedong, et al. "VehicleNet: learning robust visual representation for vehicle re-identification." IEEE Transactions on Multimedia (2020).
7. Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, Tiejun Huang; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2167-2175
8. S Targ, D Almeida, K Lyman - arXiv preprint arXiv:1603.08029, 2016
9. A Ng - CS294A Lecture notes, 2011 - ailab.chonbuk.ac.kr