Car Model and Price Classification Model

Authors
Jaewoo Kang
Jieying Liu

Abstract

The project comprises two parts; the model in the first part classifies the car images then passes the result to the second model which predicts the car price. When the image classification model is presented with a new image, it classifies the car's model, year, type, and manufacturer. However, car model classification requires a large amount of data and sophisticated shape detection since the cars of the same kind (coupe, convertible, etc...) are very alike. Our group created a model that provides sophisticated shape and object detection. The price prediction model trained a large dataset with a size of 400,000 with 4 dense layers. It took the outputs from part one and predict price as a dependent feature. The model performs with a relatively low mean squared log error and high accuracy.

1 Objective

Interested in computer vision, our group wanted to challenge ourselves to make a model to classify objects that look similar but have unique characteristics. In addition, we implemented a model to estimate the car price with given information from the first model. We will take car images as input, recognize the manufacturer, model name, and other features of the cars and eventually predict the prices of those cars through various deep learning models.

2 Part 1

2.1 Data Preparation

The data used in the project were collected from Kaggle. Part1 of the project uses a dataset of about 16,700 photos of cars. The dataset contains 196 sub-directories. Each sub-directories(each car model) have about 80 images of a specific car model. Part one of the project uses a validation split developing this model, using 80% of the images for training and 20% of the pictures for validation. The data set normalizes the RGB channel values to 255 to [0,1] range by using the rescaling layer.

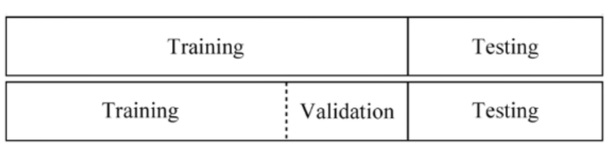


Figure 1: data was split into 8:2 for training and testing

2.2 Model building

First, we tried to implement the model that had three convolution blocks with a max pool layer in each of them. It was fully connected with 128 units on top of it, and it was activated by a ReLU activation.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

It used Adam optimizer and Sparse Categorical Crossentropy for its loss function. However, The best accuracy that our model can get was around 10%. The model was not able to accurately predict any car images other than the ones that it was taught. After re-creating various models, the accuracy did not increase significantly. So, we decided to switch to a pre-trained model.

For part one, we used a pre-trained convolutional neural network model that has 50 layers, which is called ResNET50. ResNet50 is a variant of the ResNet model, which has 48 CNN layers and 1 MaxPool, and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. The model is evaluated as a groundbreaking model using residuals. In addition, ResNet uniquely uses a technique call skip connection to add the output from a previous layer to a later layer, which helps mitigate the vanishing gradient problem.

After applying the model, we were able to increase model accuracy up to 38% which is significant considering that most cars have similar shapes and characteristics. The accuracy is shown in figure 2 shows that the car prediction model starts to lose accuracy starting 12th epochs due to overfitting. We did add an early stopping method to prevent overfitting in the later model.

Figure 2: Accuracy drops starting 12th epoches

3 Part 2: Car price prediction

In this part, we trained models to predict the car selling price. We used a dataset from Kaggle.com [1].

3.1 Pre-processing on dataset

We found a car feature data set from Kaggle.com. The data have 7,557,728 rows and 16 columns. Since there are severe typing errors in the dataset, we decided to use the first 400,000 rows of the dataset. The “Selling price” column is going to be the target column. We removed irrelevant columns like “trim” (additional string in car model name), “vin number”, “state”, “interior”, “number of doors”, “seller”, “sale date” since those columns include data that varies for each car and have no significant correlation with the selling price. We used the other 7 columns as the independent features for price prediction. During further observation of the data set, we found that there were 4 categorical columns – “manufacturer”, “model name”, “body type”, “transmission”. We replaced each category string with a unique integer for data training. Then we took 80% of the data for training and 20% for testing.

[1] https://www.kaggle.com/tunguz/used-car-auction-prices

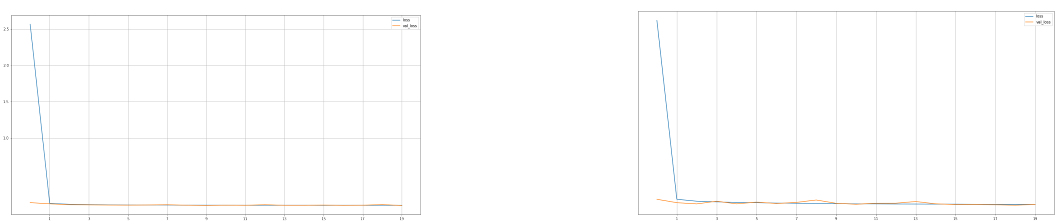
	year	Manufacture	Model	trim	Body	transmission	vin	state	condition	odometer	color	interior	seller	mmr	sellingprice
0	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg566472	ca	5	16639.0	white	black	kia motors america, inc	20500	21500
1	2015	Kia	Sorento	LX	SUV	automatic	5xyktca69fg561319	ca	5	9393.0	white	beige	kia motors america, inc	20800	21500
2	2014	BMW	3 Series	328i SULEV	Sedan	automatic	wba3c1c51ek116351	ca	4.5	1331.0	gray	black	financial services remarketing (lease)	31900	30000
3	2015	Volvo	S60	T5	Sedan	automatic	yv1612tb4f1310987	ca	4.1	14282.0	white	black	volvo na rep/world omni	27500	27750
4	2014	BMW	6 Series Gran Coupe	650i	Sedan	automatic	wba6b2c57ed129731	ca	4.3	2641.0	gray	black	financial services remarketing (lease)	66000	67000

Figure 3: example of input data set for price prediction

3.2 Model building

We built two deep learning models with 4 dense layers, one with a small network and another one with a large network. The first three layers have the “relu” activation function and the last layer works with the "linear" activation. The validation loss remains low in both models, as shown in figure 4, which implies well performance of the models

For comparison, we also trained the data with the linear regression model and Random Forest model. We evaluated the models using mean squared error, mean absolute error, and mean squared log error. Our deep learning model with a large network has a comparatively lower mean squared log error, 0.079, than the linear regression model, but not as low as the random forest model.



(a) Loss and validation loss of the model with a large network

(b) Loss and validation loss of the model with a large network

Figure 4: Evaluation on deep learning models

3.3 Result

The result of price prediction with our deep learning model is shown in figure 6. The figure shows that our model is performing well on the price prediction. The last step is to take the output from part 1 image recognition and predict the price for the recognized cars. We compared the predicted prices with the MMR (Manheim Market Report) price, which is a popular indicator of wholesale prices in used car marketing [2]. Figure 5 (b) shows the result of our prediction and the MMR values. The graph shows that our prediction has an overall matching trending with the MMR values, which indicates that our model performs well on the price prediction.

4 Further Work

The features we read from car images are limited. It was hard to find similarly matching data for image recognition and price prediction. For instance, we did not identify the odometer, condition, and color, which are relatively essential variables to evaluate used car price. Image recognition models may misclassify car models that were not taught. For further development on this project, we can add color classification as an additional feature. Given that car color is a relative feature to influence

[2] "What are MMR values?", https://joinyaa.com/guides/what-are-mmr-values/

(a) Data set from Kaggle.com

(b) Data set from car image recognition

Figure 5: Price prediction

car price, identifying the color from images may help improve the accuracy of the price prediction model.

References

[1]“CNN Image Classification | Image Classification Using CNN.” Analytics Vidhya, 18 Feb. 2020, www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/.

[2]“ML Practicum: Image Classification | Machine Learning Practica.” Google Developers, developers.google.com/machine-learning/practica/image-classification.

[3]Jason Brownlee. “A Gentle Introduction to Object Recognition with Deep Learning.” Machine Learning Mastery, 5 July 2019, machinelearningmastery.com/object-recognition-with-deep-learning/.

[4]Dwivedi, Priya. “Understanding and Coding a ResNet in Keras - Towards Data Science.” Medium, 27 Mar. 2019, towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

[5]Kharwal, Aman. "Predict Car Prices with Machine Learning." Clever Programmer, 21, September. 2020, thecleverprogrammer.com/2020/09/21/predict-car-prices-with-machine-learning.

[6]Sherma, Palase. "Keras Dense Layer Explained for Beginners." Machine Learning Knowledge, 20, October, 2020, machinelearningknowledge.ai/keras-dense-layer-explained-for-beginners.

[7]Kang, Jaewoo; Liu, Jieying. “timg5599/CS523-FinalProject” GitHub, github.com/timg5599/CS523-FinalProject/blob/main/README.md.