Autonomous Driving Recognition and Decision

Project: Traffic Sign Classification

Automotive Engineering, A2021019, Taesan Kim

1. Problem Statement

The main task of this project is to classify traffic signs from image dataset which contain a various types of traffic signs. The image dataset in this project is called GTSRB(German Traffic Sign Recognition Benchmark) dataset. GTSRB dataset includes total 39209 images for image classification and it is divided into 43 classes in accordance with traffic signs.

To classify traffic sign images into this 43 classes, it is needed to design a deep learning model properly that can extract features from images and learn from these features to predict labels. All of factors which decide the model architecture such as kinds of layers or the number of layers are hyper-parameters. Therefore, developer should decide them by various experiments to result in the best performance of classification.

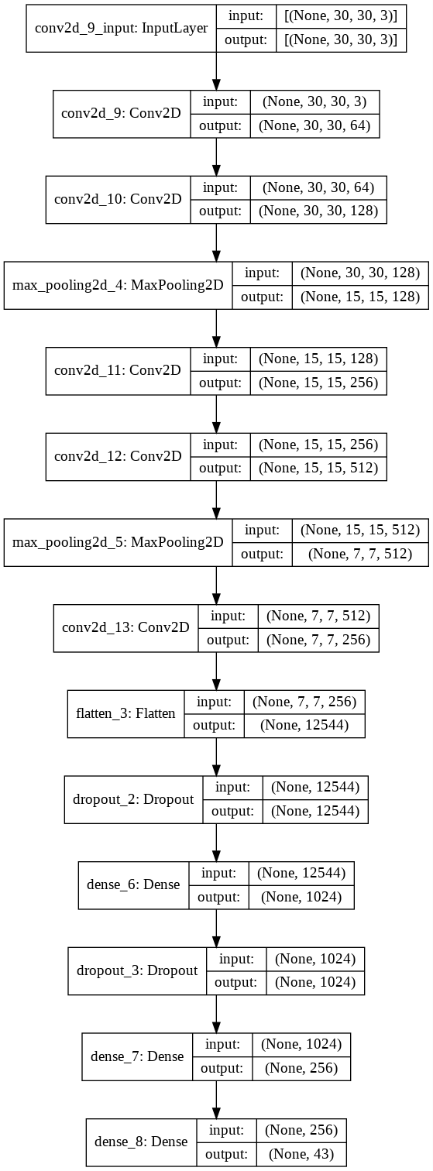
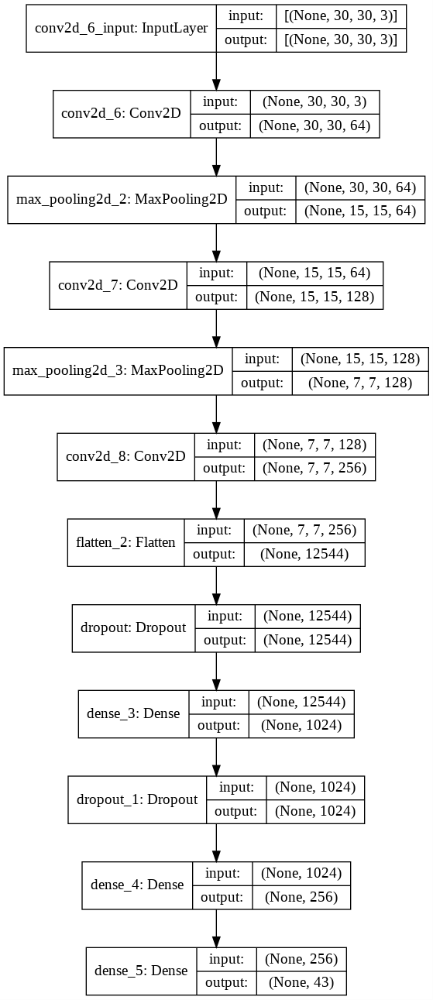
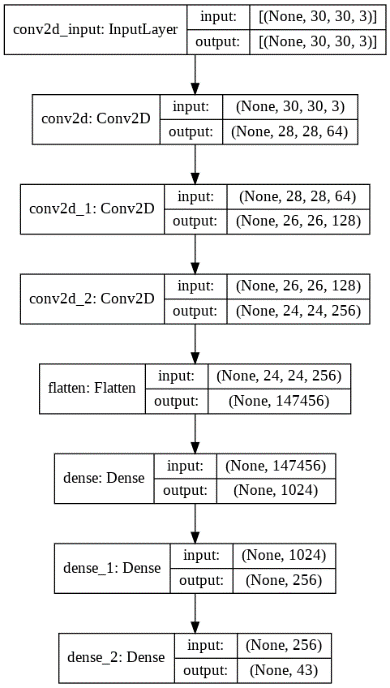
1. Proposed model

In this project, I have tried to build several models which are combined with various types of layers such as CNN, Dense, Dropout and MaxPooling layer. To summarize, I propose a decent model that reaches 98.36% of accuracy in validation mode after few experiments.

At first, I built a model, which is named “Model1”, that only consists of 3 layers of CNN for feature extraction and 3 layers of Dense for classification. The model architecture is shown in below. A problem of Model1 was that it has no MaxPooling or Dropout layer so there is no dimensionality reduction effect and regularization effect. As a result, Model1 has too many weights for learning and it overfits to the dataset. I will discuss it more in next sections.

After Model1, I built another model, named “Model2”. Model2 basically follows the architecture of Model1, but I additionally put 2 MaxPooling layers between CNN layers and 2 Dropout layers between FC layers. By adding MaxPooling layers, it could have dimensionality reduction effect and compress the feature map from CNN to highlight important features of images. Dropout layers disconnect the connections between FC layers in order to put some regularization effects and prevent the model from overfitting. In conclusion, it could be seen that there was no more overfitting and Model2 reached 97.26% of accuracy in validation mode.

Finally, to make more increase of validation accuracy, Model3 is built by adding 2 more CNN layers between MaxPooling and CNN layer. The architecture of Model3 is shown in below as well. As more CNN layers were added, the model complexity increased more than before and it could reach 98.36% of validation accuracy. Accordingly, I chose Model3 as a final model for this project that can perform the classification task in the highest accuracy among 3 models.



<Model1 architecture> <Model2 architecture> <Model3 architecture>

1. Experiments

In this section, I will compare three models that I built and explain why I chose Model3 as a final model for this project in more details.

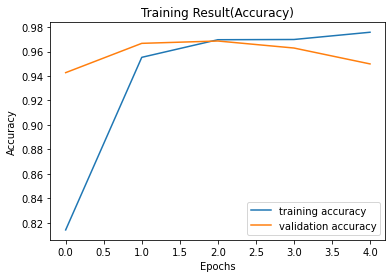
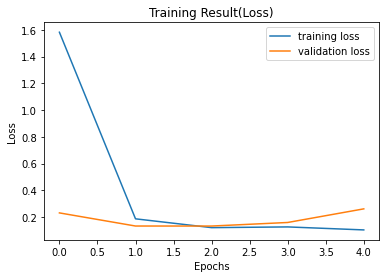
Total number of trainable weights of three models are shown in Table 1. As it can be seen in Table 1, Model1 has the largest number of weights due to no MaxPooling layer and it results in overfitting. By Figure 1, we can notice the overfitting of Model 1. In training loss graph of figure 1, as the epoch increases, the training loss keeps decreasing while the validation loss starts increase after few epochs. It means that Model1 is too complex to represent the dataset and Model1 overfits to training set so that its ability to predict new images becomes bad.

To improve Model1, I put MaxPooling and Dropout layers in Model2 and it can be seen in Table 1. The number of weights of Dense layer decreases a lot compared to Model1 because of dimensionality reduction effect of MaxPooling layers. As you can see in Figure 2, both the accuracy of training and validation keep increasing which means there is no more overfitting.

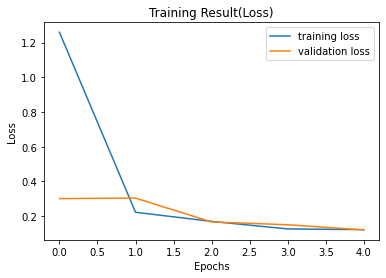
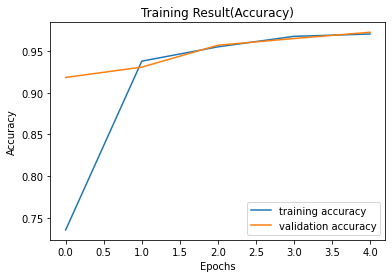
After preventing Model2 from overfitting, I added more CNN layers to gain higher validation accuracy and the number of Model3’s weights of CNN increases about 7 times compared to Model2. As a result, the validation accuracy could increment from 97.26% to 98.36%. It is fairly decent result so I chose Model3 as a final model of this project.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dense | CNN | Total |
| Model1 | 151,269,419 | 370,816 | 151,640,235 |
| Model2 | 13,119,531 | 370,816 | 13,490,347 |
| Model3 | 13,119,531 | 2,730,880 | 15,850,411 |

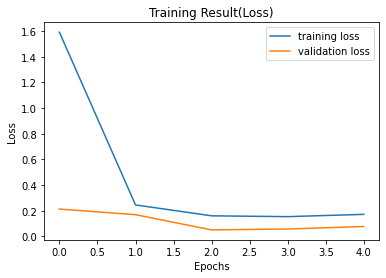
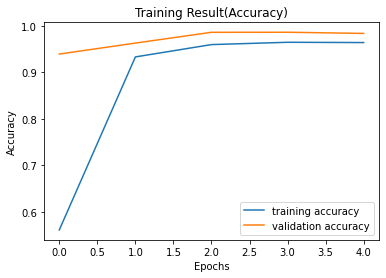
<Table 1. The number of trainable weights of three models>



<Figure 1. Accuracy and Loss of Model1>



<Figure 2. Accuracy and Loss of Model2>



<Figure 3. Accuracy and Loss of Model3>

1. Model performance evaluation

In this section, I compare the three models and discuss the trade-off between the model complexity and training time. As you can see in Table 2, Model1 has the largest number of weights so that it takes too much time to train and the accuracy is the worst among three models. Thus, Model1 can be excluded when deciding the final model.

When comparing Model2 and Model3, Model2 has fewer number of weights than that of Model3 and it takes less training time compared to Model3. On the other hand, Model3 is slightly more complex than Model2 so Model3 is superior to Model2 when it comes to validation accuracy.

In this point, the developer need to decide the final model among these models in the trade-off relationship. Consequently, I chose Model3 as the final model because of its highest accuracy and I thought that the training time of Model3 is in acceptable ranges as well.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Total Weights | Validation Accuracy(%) | Training time(s) | |
| Epoch | Total |
| Model1 | 151,640,235 | 94.99 | 95.6 | 478 |
| Model2 | 13,490,347 | 97.26 | 16.2 | 81 |
| Model3 | 15,850,411 | 98.36 | 40.8 | 204 |

<Table 2. Validation accuracy and training time of three models>

1. Discussion

In this project, I built several models for traffic sign classification task and compare them to decide the final one. By the first model, Model1, I could recognize the overfitting and tried to avoid it by constructing another model, Model2. Then, as a consequence, I could finally built a model, Model3, with the highest validation accuracy.

As I built deep learning models myself, I had done plenty of tries to discover the proper models for this task by adding and removing several layers. I could experience that deciding the model architecture is not easy and not straightforward as itself.

As a future work, I will try to build other models which can get higher accuracy than now. In addition, I am going to search for other classification tasks and solve them by building deep learning models and training them as well.