Individual Report

Yu Xi

Introduction

Data was downloaded from Kaggle and consists of 6899 images. The images are all made up of 8 classes: person, motorbike, fruit, flower, car, airplane, cat, dog. We trained CNNs from 2 to 6 layers on the dataset.

Description of your individual work

I mainly did data loading and model training.

Describe the portion of the work that you did on the project in detail.

Loading data, transforming data, training models, creating architecture figures, creating accuracy tables. Experiment setup, results, conclusion in the group report.

Results

The first model we had experiments on is a 2-layer CNN. The architecture of the model is shown in Figure 2. The numbers besides the layers are the dimensions of the corresponding layers. We keep the number of epochs at 5 and vary the learning rate. The accuracy of model is shown in Table 1. From the three models, learning rate with 1e-03 gives the best performance on the validation dataset. We will use this learning rate for the rest of our models.

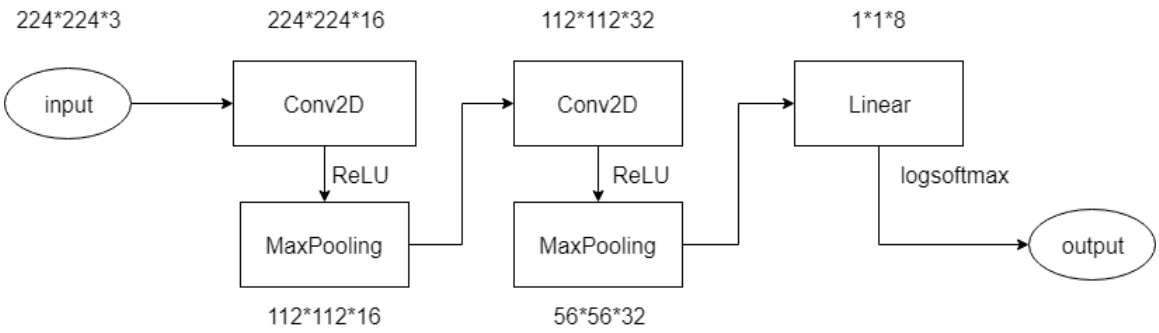

Figure 2. Architecture of a 2-layer CNN

Table 1. Accuracy of a 2-layer CNN on training and validation dataset

Learning Rate	**Training Accuracy (%)**	**Validation Accuracy (%)**
1e-02	91.4	83.8
1e-03	98.6	90.2
1e-04	98.4	66.7

After that, we train a 3-layer CNN using the same parameters with the architecture shown in Figure 3. The training accuracy is 98.8%, which is close to the 2-layer network. However, the validation accuracy is 88.4%, which is worse than the performance of the 2-layer CNN.

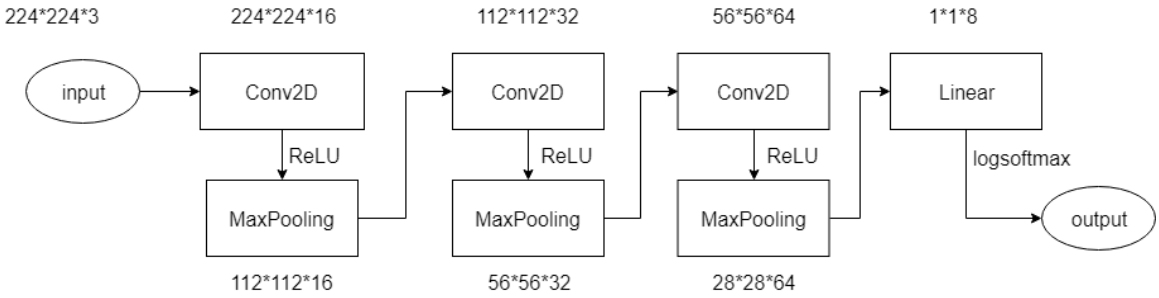

Figure 3. Architecture of a 3-layer CNN

Then, adding another layer gives a 4-layer CNN. We also experiment on a variant of that using only 2 max pooling layers instead of 4. The result of the models is shown in Table 2. For the first trial using 5 epochs, the accuracy on the training set is low, which means the model is underfitting the data. Therefore, we increase the number of epochs to 8. As a result, the training accuracy increased but the validation accuracy decreased. The variant of the 4-layer CNN performs worse on the validation dataset than other models.

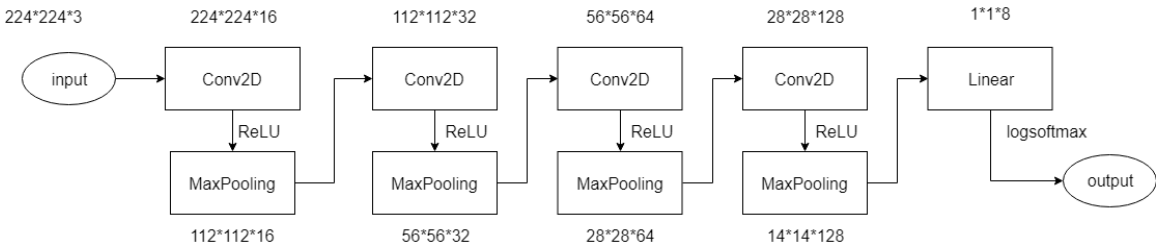

Figure 4-1. Architecture of a 4-layer CNN

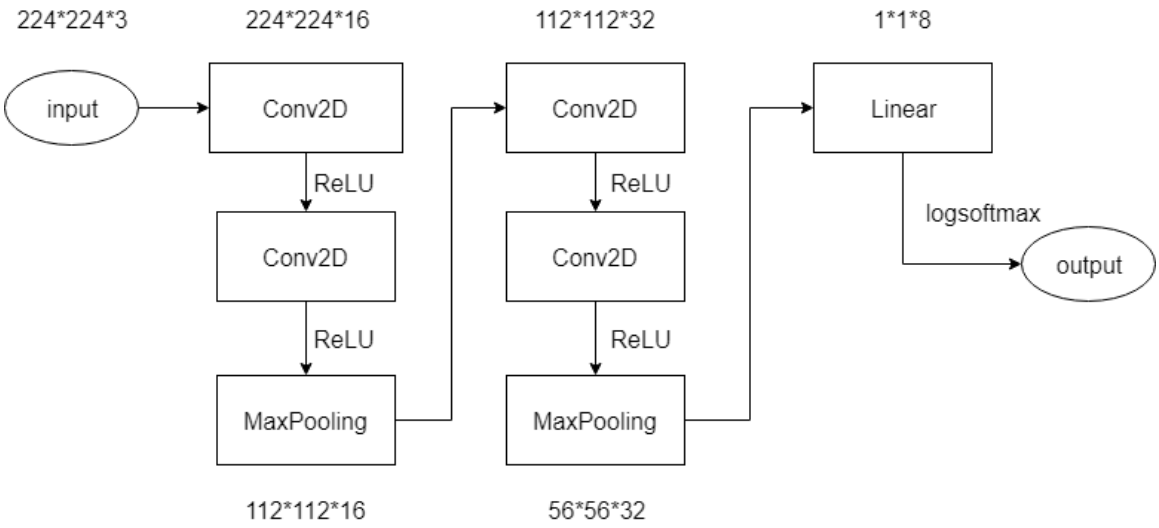

Figure 4-2. Architecture of a 4-layer CNN (Variant)

Table 2. Accuracy of a 4-layer CNN on training and validation dataset

Number of Max Pooling Layers	**Number of epochs**	**Training Accuracy (%)**	**Validation Accuracy (%)**
4	5	92.8	88.7
4	8	96.1	87.7
2	5	98.5	86.2

The next model is a 5-layer CNN shown in Figure 5 and the results in Table 3. Without using dropout layer, the model gives the best performance so far with a validation accuracy of 91.4% using 8 epochs. Since it is the model gives the highest validation accuracy so far. We tested on using an additional dropout layer on the model. However, the two models using a dropout layer does not perform so well as without using it.

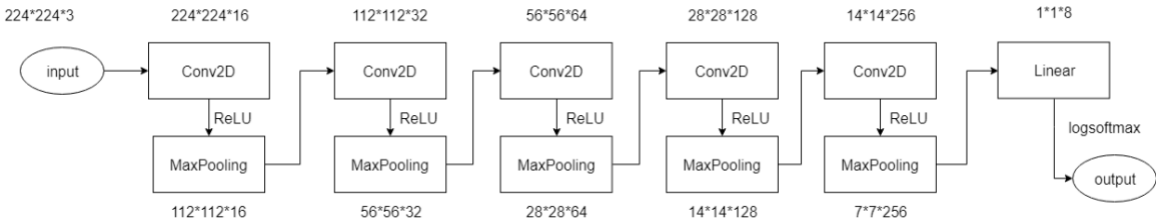

Figure 5. Architecture of a 5-layer CNN

Table 3. Accuracy of a 5-layer CNN on training and validation dataset

Number of epochs	**Dropout Rate**	**Training Accuracy (%)**	**Validation Accuracy (%)**
5	0	92.0	89.1
8	0	97.6	91.4
10	0	95.3	86.7
8	0.2	99.5	89.9
8	0.4	95.6	90.9

Lastly, we tried to experiment on a 6-layer CNN. Adding another convolution and max pooling after 5-layer CNN does not make much sense, because the dimension of the

output would be too small. Using a structure similar to Figure 4-2 with 6 convolution layers and 3 max pooling layers will run out of GPU memory.

After choosing the best model, we use test dataset to provide an unbiased evaluation on the 5-layer model. The test accuracy is 91.2%, which is close to the validation accuracy as expected.

Summary and conclusions.

In conclusion, the best model we have is a 5-layer CNN with learning rate = 1e-03, number of epochs = 8, and dropout rate = 0. The test accuracy of this model is 91.2%. What we have learned is all parameters have some effect on the model, and there are interactions among most of the parameters. With the same parameters, a CNN model has more layers does not necessarily perform better. Also, using more number epochs does not always increase the accuracy on the training dataset. In our experiments, using a dropout layer improves the training accuracy, but not the validation accuracy. Changing other parameters along with the dropout layer may give a better model.

One of the improvements could be make is do to more experiments on the parameters that were not tested. Also, the combinations of the parameters that were not tested may provide more information on the effects. Lastly, if we had more GPU memory, we could test the model with more convolution layers and different architecture of networks.

Calculate the percentage of the code that you found or copied from the internet.

(100-10)/(100+350)=20%

Reference

[1] Roy, Prasun. "Natural Images." *Kaggle*, 11 Aug. 2018, www.kaggle.com/prasunroy/natural-images.

[2] "NN-SVG." *NN SVG*, alexlenail.me/NN-SVG/LeNet.html.

[3] "Pytorch VGG16 Natural Images." *Kaggle*, www.kaggle.com/gabrielloye/pytorch-vgg16-natural-images.

[4] "VGG16 - Convolutional Network for Classification and Detection." *VGG16 - Convolutional Network for Classification and Detection*, 21 Nov. 2018, neurohive.io/en/popular-networks/vgg16/.