# Deep Learning(M)
## *Coursework*

Molin Liu
*University of Glasgow*
2473686l@student.gla.ac.uk

March 11, 2020

**Abstract**

This is abstract **Keyword**: ResNet, Deep Learning, Image Classification, Reinforcement Learning

# 1 Introduction

Converlutional Neural Network(CNN) has been widly used on image classification. In this report, I will use transfer learning skill, try to train a CNN model based on my cat breed dataset.

The images of cats in all dataset are not that good as humans', especially face datasets like *LFW*. The pose, occlusions, lights, are more worse, which bring many challenges to classify these images, see Fig.1.


Figure 1: Example of cats' dataset

I used to think to do the *Cat-and-dog* problem, which is a classic problem in image classification. However, when I finished my training of ResNet18 on the *Cat-and-dog*, the result was too good, as shown in Fig.2, that I can't do more things with this model. While the trained model can be used in the process of creating my dataset.

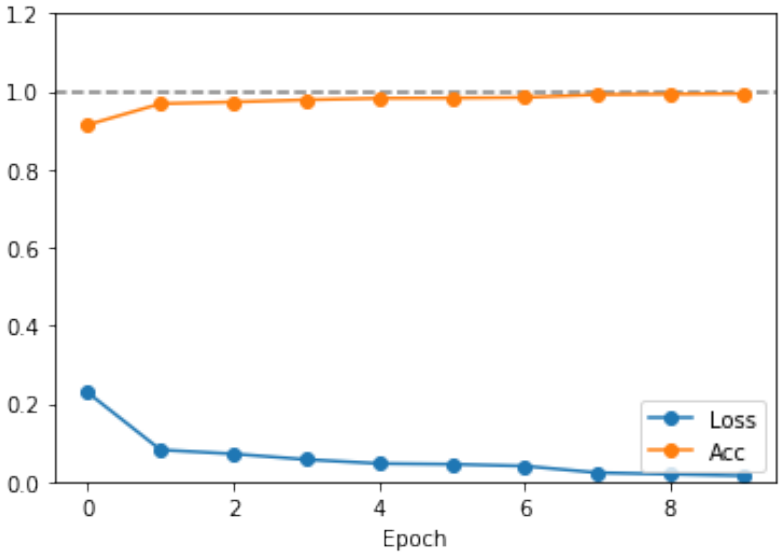


Figure 2: Example of cats' dataset

## 2 Dataset

The data set used in this report, based on *Cats and Dogs Breeds Classification Oxford Dataset*. However, the data set contains some irrelevent images, or formate error. Hence, before we load data into data loader, I need to clean the dataset. First I join different dataset togather, use *Cats and Dogs Breeds Classification Oxford Dataset* as the base dataset, extended from *Cat Breeds Dataset*, *The Oxford-IIIT Pet Dataset*. In the instruction files of those datasets, they all specify the label of the animals, which can be used to seperate cats' images from others.

Meanwhile, I set a threshold for each class in the cats dataset to meet the requirement of the coursework–simply drop those classes of cat which has less than 100 images.

Then I got a dataset with 12 classes of cats. As for those irrelevent images, first I judge according to the type of file. These images are all with .jpg extention, however, PIL provide a Image class, which can be used to check the format of the image. Then I use a ResNet18 trained on Cat-Dog dataset, to seperate the cat images from others.

## 3 Transfer Learning

In this work, we are supposed to use a pre-trained model as the features extraction method. The reason that I choose ResNet18 as my pre-trained model is that ResNet is one of the most famous classification model[1][2], andf the ResNet18 has faster trainning speed than ResNet152, and they don't have significant difference of accuracy on my dataset.

As shown in the instruction, all of the layers would be frozon. The code is shown in the

*Colab Notebook*. Then I create a new class called *CatBreedModel*. A *Sequence* is used to create a block of the pre-trained model. The batch of data will be first fed into pre-trained model, and then go throught the rest part of the student-define model. I only use a Linear block to convert the dimension to our dataset. As for the parameters setting, it will be illustrated in the following part.

# 4 Simple CNN

We also need to create a baseline for comparison. In this work, I create a simple CNN classifier as my baseline. This simple CNN classifier contains 2 convolution operations, togather with ReLu activation function. I try to let these two Conv have large output depth, and small filter size, so that they will extract the detail of my dataset image, given that cat breed classification is kind of sophisticated classification, the detail always matters. However, too large output may lead the training time unaccepatable. Here I use 2 Max-pooling operations. Finally, I set 3 fully connected layer to linearly project the flow-in data into class space.

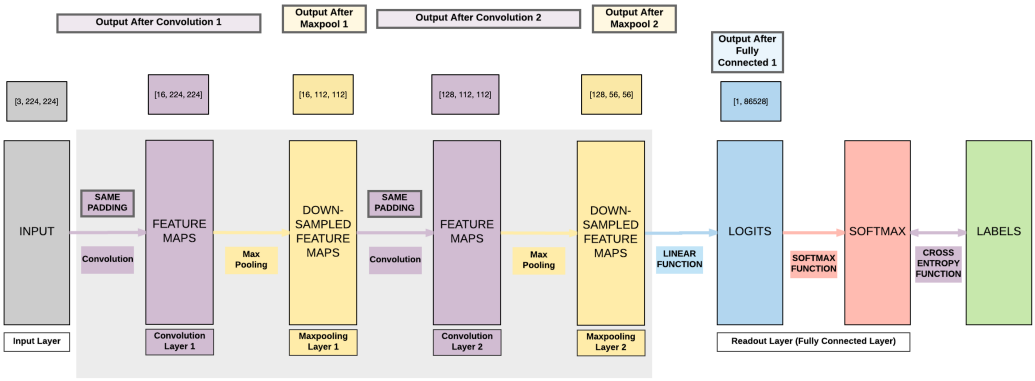


Figure 3: The Structure of Simple CNN

As for the parameter setting, they are calculated as follow. The output formula for Convolution is shown below:

$$O = \frac{W - K + 2P}{S} + 1 \tag{1}$$

where $O$ is the output size, $W$ is the input size, $K$ is filter size (kernel size), $P$ is the padding size, and $S$ is the stride size.

$$O = \frac{W - K}{S} + 1 \tag{2}$$

In Formula.2, it's used to calculate the output of pool. Combine with these equations, then we can decide the parameters' setting.

# 5 Analysis

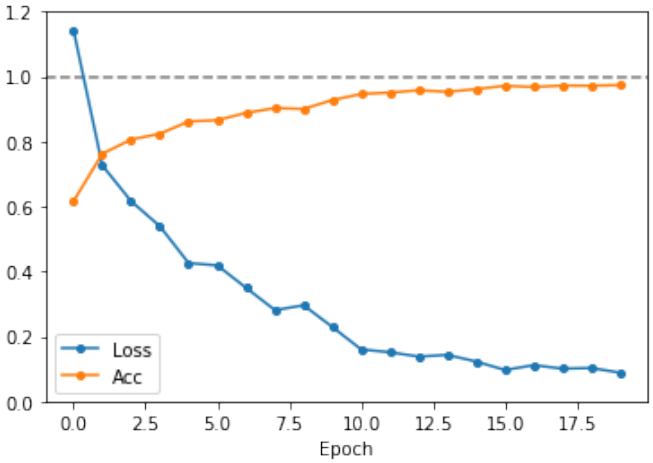


Figure 4: The Result of CatBreed Model

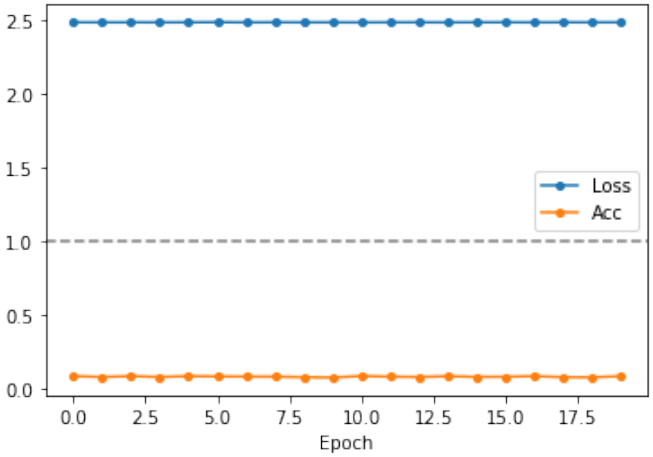


Figure 5: The Result of Simple CNN

From Fig.4 and Fig.5, we can easily find that CatBreedModel has a good performance in this dataset. The curves converge to their limits after the model iterates for 15 epoches. The accuracy of this model after 20-epoch-training is 0.9982.

### Wrong Case Study

The effective way to know how to improve our model is to see the wrong case in the model prediction. In Fig.6 nad Fig.7, the problem is caused by incomplete cat images.

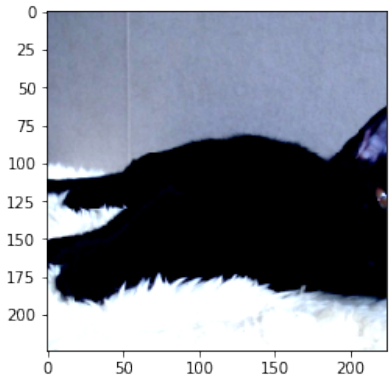


Figure 6: The Result of ResNet18

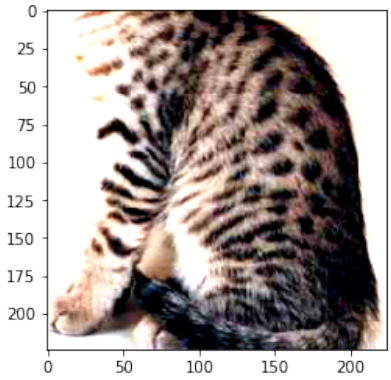


Figure 7: The Result of ResNet18

As for the Simple CNN, it's obviously that the features extractions does not work. The simple classifiers like this, are desigened to handle very some dataset. And the size of input is small as well, in the example, which is $10 \times 10$ not $224 \times 224$ in my dataset.

Then I compare Fig8 and Fig4, CatBreedModel converges more easily, although the change is small.

# 6 Conclusion

In this work, I use transfer learning method provided py pytorch, to build a classifier on my cat categorie dataset. It shows its power on feature extraction, which can save a lot of time to train a model from scratch.

**Acknowledgements** This work was supported by Miss. Tiana. Thanks for her accompany.

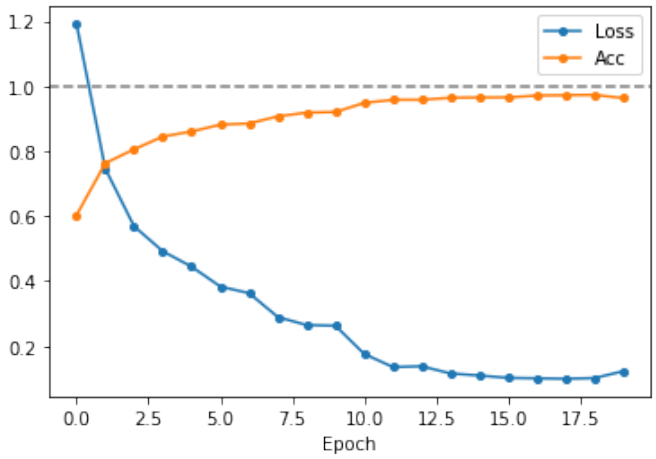


Figure 8: The Result of ResNet18

# References

[1] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[2] T. Akiba, S. Suzuki, and K. Fukuda, "Extremely large minibatch sgd: Training resnet-50 on imagenet in 15 minutes," *arXiv preprint arXiv:1711.04325*, 2017.