UNSW SYDNEY

Distinguishing Images of dogs from cats

COMP9417 Project

Team members:

Yiyan Yang z5183946

Daniel Qian z5183850

Links to the competition:

https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/overview

1.Introduction

The “dogs vs cats” problem has been a classic classification problem since 2013 on Kaggle which requires the machine to Distinguish images of dogs and cats from each other. Although it is a simple task for humans to tell the difference between cats and dogs, it might be difficult for computers to do so without a proper algorithm to Separate dogs from cats. This is quite a typical problem to apply classic image classification algorithms to examine their power.

Convolutional neural network is selected as the model to fit the dataset which is split into train and validation set with ratio of 4 to 1. It is a simple but a powerful model to build a neural network for image classifying as in a CNN, features of specific regions of a picture are extracted for classification by simulating visual nerve cells. By mimicking neurons, the model can finally approach to a high accuracy.

3.Implementation

Data pre-processing

Firstly, we downloaded the datasets and extracted them in a separate file. Then we resize the pictures into the same size to keep the consistency of the dataset and randomly flip them for robusty. Finally, we extracted the filenames in the train set as labels for classification.

Model building

Pytorch is chosen to as the tool for building the model. The network consists of 3 convolutional layers and two linear layers as shown in Figure 3.1. For each convolutional layer, 3 is used as the kernel size with stride of 2, and BatchNorm2d, ReLU and Maxpool2d with size of 2 were applied. A relu function is also inserted between the two linear layers as a stable activation. Finally, the model returns the probability of the input to be a cat or a dog.

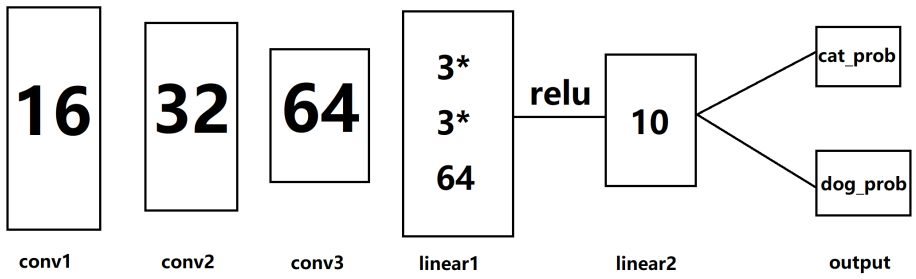

Figure 3.1

4.Experiments

As the model is simple to construct with the help of pytorch, we focus on experiments on the parameters and other tools that may be useful for this problem. Firstly, we determined the final model by comparing the efficiency of several

networks, and then chose appropriate optimizer and loss function. Secondly, lots of adjustments were made on the meta-parameters to achieve an acceptable accuracy with high efficiency. Finally, we tried some transformations on the input data to make the model more robust.

Models

Linear model, Multi-Layer perceptron and Convolutional Neural Network were tested on the same dataset. Finally, the convolution network approaches to the highest accuracy.

SGD and Adam were applied as the optimizer and it suggested that Adam should be used because of the higher efficiency and similar accuracy.

Cross entropy and MSE were compared as candidates and it was found that Cross entropy works better in classification tasks so it is determined as the loss function for the submit version.

Meta-Parameters

The parameters that were tested are listed below.

lr: Learning rate. Because Adam is chosen as optimizer, typically a smaller learning rate is preferred. After several tests on different learning rate that are picked according to divide and conquer strategy, 0.001 is found to be the optimal learning rate.

epoch: In each epoch, the model is trained through the whole dataset. Hence the larger the epoch, the more accurate the model would be. However, in the sake of avoiding overfitting, value of epoch should not be too high. we finally set it to 10 with which the model can generate acceptable answers.

Batch size: It does not change the overall accuracy in a significant scale, but with a larger batch size, the train procedure could be faster.

Data processing

Picture transform: We rescaled the picture in the same size and randomly rotate the picture. Initially the accuracy was low but after 10 epochs, the overall accuracy was lifted.

5.Conclusion

This study has selected CNN as its learning method, the result shows that the log loss is 0.48, which is just acceptable according to the standard of Kaggle. However, this algorithm is sensitive with the changes of the training set, and alone with the difficulties to transfer to other subjects such as identifying other species. Learning rate changes could be made to adjust the algorithm to fit for more complex test cases, otherwise this model could not show accurate result in predicting other animals. The study further proves CNN could have done greater than this in classification on sets of images in machine learning.

6.Further Study

This study has tested a CNN learning algorithm on classification which can be expanded in many areas. After further work in the future, this model should be able to identify different things inside one picture. That is, listing all the components of a single picture, such as google reverse picture search.