Cpts570 Machine Learning Final Project

Lieh-Hsuan Chen
Washington State University

***Abstract*—In this research paper, my objective is to conduct an experiment to solve the problem of low accuracy in large amount of classes during convolutional neural network (CNN) classification. To do experiment, the data was imported from Google which provided to Kaggle that contained 3,0000 landmark images and 4742 classes, and divided them into training set and testing set. Firstly, the data will be trained with different number of convolutional layers and shown the accuracies. Secondly, the data will be normalized among the training process and shown the accuracies. Thirdly, the data will be trained with different optimizer and shown the accuracies. Finally, the program will draw the curve diagrams to compare the accuracy of each method.**

Keywords—CNN, classification, classes, experiment, accuracy

I. INTRODUCTION

Nowadays, images recognition is very common technology in the world, more and more company implement this technology to help them to pursue huge benefits, like parking system can recognize each car by take photo with its license plate, then charge parking fee automatically without human resource. However, with prediction classes increased, the accuracy of image classification will decrease. People need to solve this problem to make machine be able to predict more complex classification. Therefore, in this research, I will execute experiments and discuss how to solve this problem with convolutional neural network (CNN).

Before step into problem section, the introduction of CNN will be given below:

Convolutional Neural Networks are very similar to ordinary Neural Networks, which is composed of Convolutional Layer, Pooling Layer, and Fully-Connected Layer. For convolutional layer, the input images will go through several times of filtering to extract significant information, and the size will become gradually small. As for pooling layer, it progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network, and hence to also control overfitting. Lastly, the data will be flatten and drop into full connection layers (as same as artificial neural networks)

II. PROBLEM SETUP

In CNN images classification, the machine will learn and make prediction of each image. In particular, for binary or classes which less than hundred, CNN is able to learn efficiently and produce high accuracy model. Nevertheless, in large amount of classes, the accuracy will be low. To be concise, with the classes increased, CNN will need to deal with more parameters and output units. Consequently, the accuracy will be very low if there are thousands of classes. Thus, in this paper, the different methods will be employed to test the accuracy of predictions which made by CNN on thousands of classes classification, and compare which method is most powerful to increase accuracy in this situation.

III. SOLUTION APPROACH

Firstly, the most simple solution is adding more convolutional layers in CNN. In convolutional layers, each image will be filtered into several new images in order to extract significant information from original image. To be specific, if one convolutional layer accept the image of size (W1, H1, D1), the filter K, strides S, spatial extent F and non-zero padding P will produce the image of size (W2, H2, D3), where $W2=(W1-F+2P)/S+1$, $H2=(H1-F+2P)/S+1$, $D2=K$ [1]. There is an example called ResNet (Residual Network), it not only contains up to 34 CNN layers, but fetching additional information extraction by connecting some cnn layers with their heads and tails. Therefore, if the more image information can be place into training process, the accuracy will increase.

Secondly, batch normalization is also a solution to increase accuracy. Since the activation input value of the deep neural network before the nonlinear transformation is deepened on the depth of neural network or deepened during the training process, its distribution gradually shifts or changes. The reason why the training convergence is slow is generally that the overall distribution gradually goes to the nonlinear function. The upper and lower limits of the value interval are close to each other, so this leads to the disappearance of the gradient of the low-level neural network in the back propagation, which is the essential reason for the slower convergence of the training deep neural network. Thus, batch normalization is to force the distribution of the input value of any neuron in each layer of neural network back to the standard normal distribution with a mean of 0 variance by a certain standardization method, and increase NN's accuracy.

Lastly, different optimizer can produce higher accuracy in the end. There are many machine learning optimizer in the world, such as SGD, Adagrad, Adadelta, Adam and Nadam. For instance, SGD (Stochastic Gradient Descent) is one of the well-known optimizer in machine learning, it can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient by an estimation of data which is calculated from a randomly selected subset. Another example is Adam (Adaptive Moment Estimation), it is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation, and the reason it's called that is because Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network. Thus, the different optimizer employed in training will lead different accuracies, the suitable optimizer can solve this problem [2].

IV. EXPERIMENTS AND RESULTS

In the experiments, the project imported 3,0000 landmark images from Kaggle which contained 4742 classes to be classified. On the other hand, in order to do experiments, the dataset was divided into 2,0000 training data and 1,0000 testing data.

Pre-setup of convolutional neural network:

- Epochs: 25
- Image size: 96 x 96
- Rescale: 1/255
- Basic Structure:

(Conv→Conv→Batchnormalize→Pooling→Dropout)*3→Flatten→Full connect

- Dropout: 0.25
- Pooling: MaxPooling
- CNN activation functions: Relu
- Last activation function: Softmax

Experiment – Increasement of convolutional layers

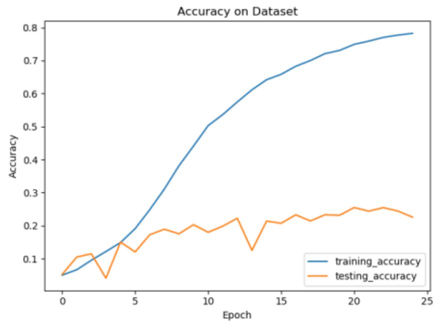

Fig1. 2 convolutional layers with adam

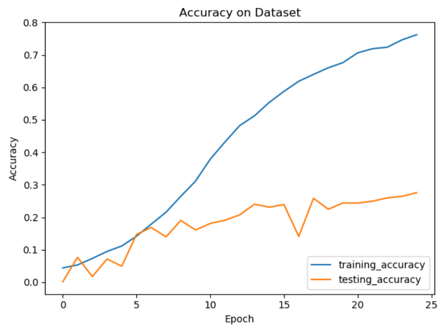

Fig2. 3 convolutional layers with adam

In this experiment, we compare the accuracy of 2 convolutional layers and 3 convolutional layers. Both of them optimized by adam algorithm. The result shown in Fig1 and Fig2, for high quantity of classes, 3 convolutional layers is less overfitting than 2 convolutional layers, where existed in epoch 5. Moreover, 3 convolutional layers got higher accuracy on testing data which is 0.27 and 2 convolutional layers got 0.22. However, both accuracy on training data were approximately same.

Experiment – Implementation of Batch Normalization

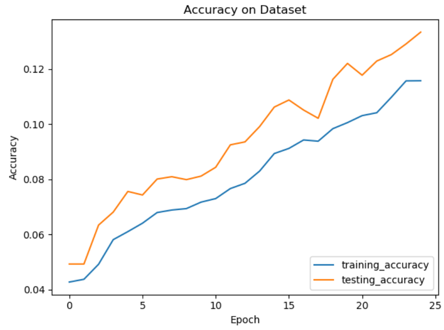

Fig3. 3 convolutional layers no batch normalize

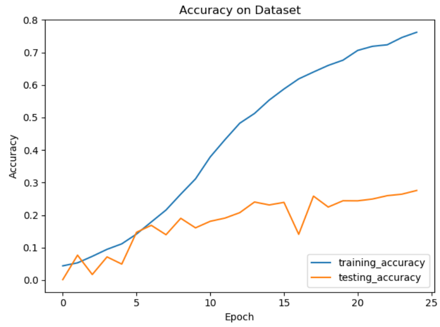

Fig4. 3 convolutional layers batch normalize

In this experiment, we take out the bath normalizations in the training process. The result shown in Fig3 and Fig4, after 25 epochs, Fig3 only got accuracy of 0.13 on testing data which is much different with Fig4. Besides, the training convergence speed was dramatically slow without batch normalization in Fig3.

Experiment – Difference of Optimizer

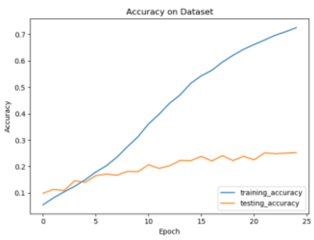

Fig5. Optimizer of Adagrad

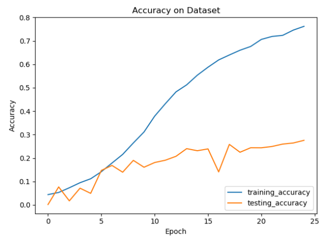

Fig6. Optimizer of Adam

Fig7. Optimizer of Nadam

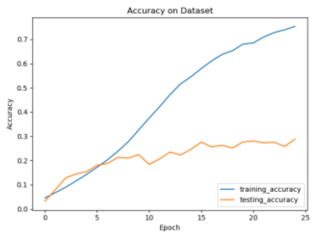

Fig8. Optimizer of SGD

In this experiment, the CNN were optimized by 4 different algorithms: adagradm adamm nadam, SGD, and we can compare those methods to know which optimizer could perform better in high quantity of classes. In the result, the accuracy on training data and testing data are:

- Adagrad: 0.72/0.25
- Adam: 0.76/0.27
- Nadam: 0.81/0.31
- SGD: 0.75/0.28

Consequently, the nadam optimizer got highest accuracy and adagrad optimizer got lowest accuracy in the experiment. Since Nadam has a stronger constraint on the learning rate, and has a more direct impact on gradient updates, the place where you want to use momentum RMSprop, or Adam, it can mostly employ Nadam alogrithm to get better results. On the other hand, SGD (Stochastic Gradient Descent) is much harder to choose correspond learning rate, and sometimes we may want to update faster for features that occur infrequently and slower for features that appear frequently. At this time, SGD is not able to meet the requirements.

V. Conclusions and Future Work

In summary, high quantity of classes prediction is always a problem in machine learning, since there are massive of features or classes to fit into machine learning algorithms. In this paper, we did three experiments to test whether those methods can improve learning performance. Firstly, more convolutional layers can reduce overfitting and increase accuracy. Secondly, batch normalization is a practical way to speed up convergence of training models and improve accuracy. Thirdly, various optimization algorithms will lead to different result, for high quantity of classes, nadam performed better than other algorithms in the experiment. However, the overfitting in the experiment is obvious. Since there are massive of classes in dataset, although there are some drop out layers, it's easily cause overfitting when training data. In order to solve this problem, we can employ ResNet to solve this problem in the furture.

Acknowledgments

[1] CNN:http://cs231n.github.io/convolutional-networks/

[2] Adam:https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[3] Kaggle: https://www.kaggle.com/c/landmark-retrieval-2019/overview.