Deep Learning
Assignment 2

Ting Chun Yeh

NM6124012

National Cheng Kung University

Abstract—**This assignment is separated into two parts. Both parts are about image classification. First, it is to design a model that contents a dynamic convolution module, which is a special convolution module that is spatial size invariant and can handle an arbitrary number of input channels. Secondly, it is to design a model with two to four layers of CNN, Transformer or RNN network, and try to achieve at least 90% performance of the model ResNet34 with the same base. In the assignment, we are going to make a thorough inquiry of the use of convolution.**

Keywords— Deep Learning, Classification, Dynamic Convolution, RRDB, Receptive Field, Feature Extraction.

I. Introduction

Image classification has always been a popular topic in the field of computer vision. Different combination of modules can have different effect and create different level of performance. As we all know, most of the time, the input channel have to be fixed due to the architecture of the model. However, changing the design of the model can easily solve the problem. When the input channels are variablized, the model needs to be flexible in order to accept the floating number. And here comes the dynamic convolution module. The dynamic convolution module is spatial invariant and can handle an arbitrary number of input channels. It is fine if the input images are with various channel combinations. As long as we change the convolution into dynamic convolution, the problem is solved. Deep learning is one of the reason why image classification can achieve high performance. Most of the time, we require the network to be as deep as possible, and it has already been proved by a lot of case that it is true that deeper models do perform better than models with fewer layers. Nonetheless, depth is not the only way to make the model perform better. Packaging each layer is another possible way. To increase receive field, making good use of different kind of blocks, attention mechanism, or Graph Convolution Networks (GCN) are all good ways, and these will thus enhance feature extraction capabilities.

In the assignment, all the models are trained, validated, and tested on the same dataset with the same base and the same setting. We are going to see whether changing the design of the convolution module can still maintain the performance. We will go into detail about the effect of the convolution modules.

II. Method

A. *Task: Designing a Convolution Module for Variable Input Channels*

Most of the time, we try to keep the channels of the input same for the convenience of the network. However, there are times that the input channels are various, unifying them may cause information loss. Keeping the inputs the original format is the best case. Therefore, if the datasets of the images are with various channel combination, we will need a convolution module that can accept different input channels and modify the convolution kernel of the network according to the input. And this is the basic idea of dynamic convolution module, modify the setting of the input kernel size of the convolution module base on the number of channel of each input data.

A dynamic convolution involves generating the filters dynamically based on the input data. To design a dynamic convolution module, first, we will need a module in order to generate the attention weight according to the input data. The attention weights represent the optimal aggregation of linear models for a given input. We apply squeeze-and-excitation to compute the attentions. Dynamic convolution has K convolution kernels that share the same kernel size, input dimension, and output dimension. We use the attention weights which just computed to aggregate the kernels. This process compose a dynamic convolution. Figure 1 is the flowchart of a dynamic convolution. It briefly interprets the working process and the components of a dynamic convolution.

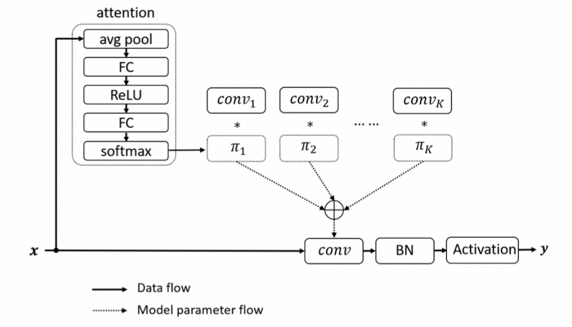

Figure 1. The design of a dynamic convolution layer

B. *Designing a Two-Layer Network for Image Classification*

From multi-layer perceptron to deep learning neural network like ResNet108, how many layers in a model seems to be an important factor which will influence the performance of the model. However, only keep building the layers is not the only thing that is important and is influential to the model. Feature extraction is another key when building a model. If we want to reach high performance, having a reach base of feature and a good knowledge of the dataset is critical. In order to enhance feature extraction, one of the way is to increase receptive field.

We apply technique: residual-in-residual dense block, RRDB to the model in order to increase the receptive field and enhance feature extraction. The classification model is made of three blocks of residual-in-residual dense block, and each residual-in-residual dense block contents four layers of convolution layer, the last layer of convolution layer is the output layer.The RRDB will keep the original input data feature and add on the detailed feature information extracted from the block. Through the way of keep extracting the feature but still keeping the original feature, it clearly enhance the feature extraction and reach the goal of increasing receptive field.

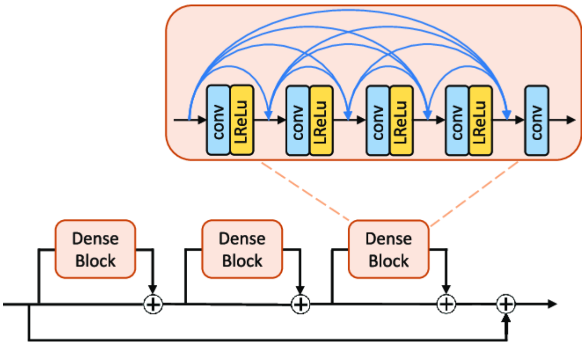

Figure 2. The residual-in-residual dense block RRDB

III. Result

For comparison, we use Mini-ImageNet as the dataset. The dataset is divided into train set, validation set, and test set with 63325 images, 450 images, and 450 images respectively. The model designed by ourselves and the comparison network are trained and inference on the same base with the same setting.

A. *Task: Designing a Convolution Module for Variable Input Channels*

For the task designing a convolution module for variable input channels, the backbone used for the model is ResNet18; therefore, by the sentence: Compare the performance of the model using the dynamic convolution module with naive models across different input channel combinations, here we compare with ResNet18.

	ResNet18	Model with Convolution Module

Epoch = 10
Lr = 0.1
		43.56%

Epoch = 30
Lr = 0.001
	39.33%	25.56%

Table 1. The best accuracy of validation set of task A

	ResNet18	Model with Convolution Module

Epoch = 10
Lr = 0.1		

Epoch = 30
Lr = 0.001
		17.11%

Table 2. The accuracy of testing set of task A

B. *Designing a Two-Layer Network for Image Classification*

For the part designing a two-layer network for image classification, we compare the performance with ResNet34 according to the instruction and hope to achieve at least 90 % of ResNet34's performance.

Identify applicable funding agency here. If none, delete this text box.

	ResNet34	Two-Layer Network
Epoch = 20	41.78%	10.22%
Epoch = 50		

Table 3. The best accuracy of validation set of task B

	ResNet34	Two-Layer Network
Epoch = 20		
Epoch = 50		

Table 4. The accuracy of testing set of task B

IV. Conclusion

I try hard to get the best performance; however the outcomes aren't satisfying. For task A, even though the result seems quite good when doing the experiment setting epoch equals to 10, it kind of overfit due to the setting of learning rate. I try to do it again setting learning rate equals to 0.001, epoch equals to 30, but the performance isn't as good as the previous setting. As for task B, I tried to train with 20 epoch only, but the accuracy is pretty low. I try to change the structure and train with more epochs. However, the kernel crashed again and again during the process of training.

Due to the limited computation environment, the data, and the model structure, it costs a lot of time to train each of the model. Averagely, one epoch for one hour. Also the kernel is quite easy to crash. Therefore, there are only few of finished experiments. And that's why some of the results are empty.

Dynamic convolution module do bring the convenience to the user. At the same time, it seems that the performance can still maintain at a certain level. Trying to prove whether the two-layer network can achieve such accuracy, but it seems failed at the end of the experiment. Hope to have more time and resource to finish the experiment.

GitHub: https://github.com/FierceTiffany/DeepLearningAssignment2

References

1. https://github.com/kaijieshi7/Dynamic-convolution-Pytorch
2. https://machinelearningmastery.com/training-a-pytorch-model-with-dataloader-and-dataset/
3. https://blog.csdn.net/tang330023555/article/details/118733184

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.