Deeper Networks for Image Classification

JAEMIN KIM 180806577

1. Introduction

Due to research of Deep Learning and Convolutional Neural Networks, image classification using Deeper Networks has dramatically advanced in the past few years. The primary goals of this report are to understand the deeper networks and to compare the two model: VGG-16 and GoogLeNet. To achieve this, I have implemented the two models using Keras library in Python3 and trained f ashion-MNIST and cifar10 dataset.

This report consists of three main sections. First section is about deeper networks for image classification. And then, in the Method, I described the two models that I compared the performance with MNIST dataset. Lastly, Experiments section shows the train and test results of VGG-16 and GoogLeNet model.

2. Critical Analysis / Related Work

Multilayer perceptrons are known as neural networks which are multiple layers of neurons densely connected to each other. A deep vanilla neural network has such a large number of parameters involved that it is impossible to train such a system without overfitting the model due to the lack of a sufficient number of training examples. However, with Convolutional Neural Networks(ConvNets), the task of training the whole network from the scratch can be carried out using a large dataset like ImageNet. The reason behind this is, sharing of parameters between the neurons and sparse connections in convolutional layers.

AlexNet, proposed by Alex Krizhevsky, was one of the first Deep convolutional network to achieve considerable accuraJHcy on the 2012 ImageNet LSVRC-2012 challenge with an accuracy of 84.7%. It consists of 5 Convolutional (CONV) layers and 3 Fully Connected (FC) layers. The activation used is Rectified Linear Unit (ReLU). it uses ReLu(Rectified Linear Unit) for the non-linear part, instead of a Tanh or Sigmoid function which was the earlier standard for traditional neural network. The advantage of the ReLu is that it trains much faster than the latter

because the derivative of sigmoid becomes very small, and it leads to weights almost vanish. This is called the vanishing gradient problem. And it solved the issue of the over-fitting by using a Dropout layer after every Full-Connected layer(FC layer). After this model, several models are suggested to improve the AlexNet.

3. Model

As mentioned in the introduction, I tested two models for this experiment: VGGNet and GoogLeNet.

3.1 VGG16

First model was born out of the need to reduce the # of parameters in the CONV layers and improve on training time. This architecture is from VGG group, Oxford. It makes the improvement over AlexNet by replacing large kernel-sized filters(11 and 5 in the first and second convolutional layer, respectively) with multiple 3X3 kernel-sized filters one after another. With a given receptive field(the effective area size of input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features, and that too at a lower cost. There are multiple variants of VGGNet (VGG16, VGG19 etc.) which differ only in the total number of layers in the network. For this report, VGG16 is implemented. The structural details of a VGG16 network has been shown in Figure 1.

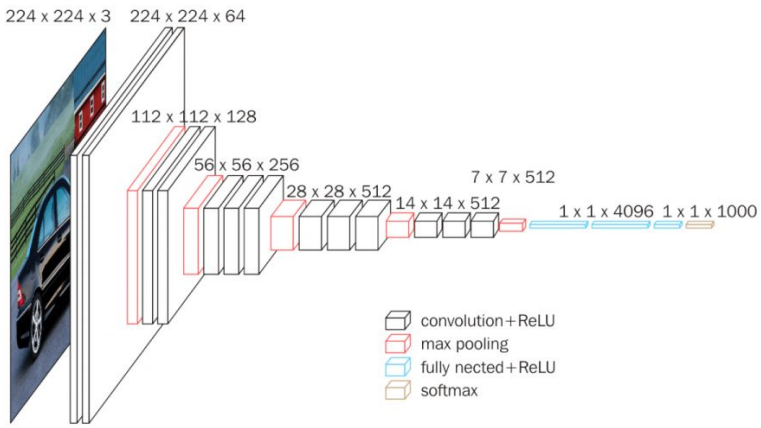


Figure 1. VGG16 Block Diagram (from neurohive.io)

3.2 GoogLeNet

The winner of the ILSVRC 2014 competition was GoogLeNet(a.k.a. Inception V1) from Google. In an Image classification task, the size of salient feature can considerably vary within the image frame. Hence, deciding on a fixed kernel size is rather difficult. Lager kernels are preferred for more global features that are distributed over large area of the image, on the other hand smaller kernels provide good results in detecting area specific features that are distributed across the image frame. For effective recognition of such variable sized feature, we need kernels of different sizes. That is what the inception module does, which is part of GoogLeNet. Instead of simply going deeper in terms of number of layers, it goes wider. Multiple kernels of different sizes are implemented within the same layer. Inception Module can be seen in Figure 2, and Figure 3 shows GoogLeNet structures.

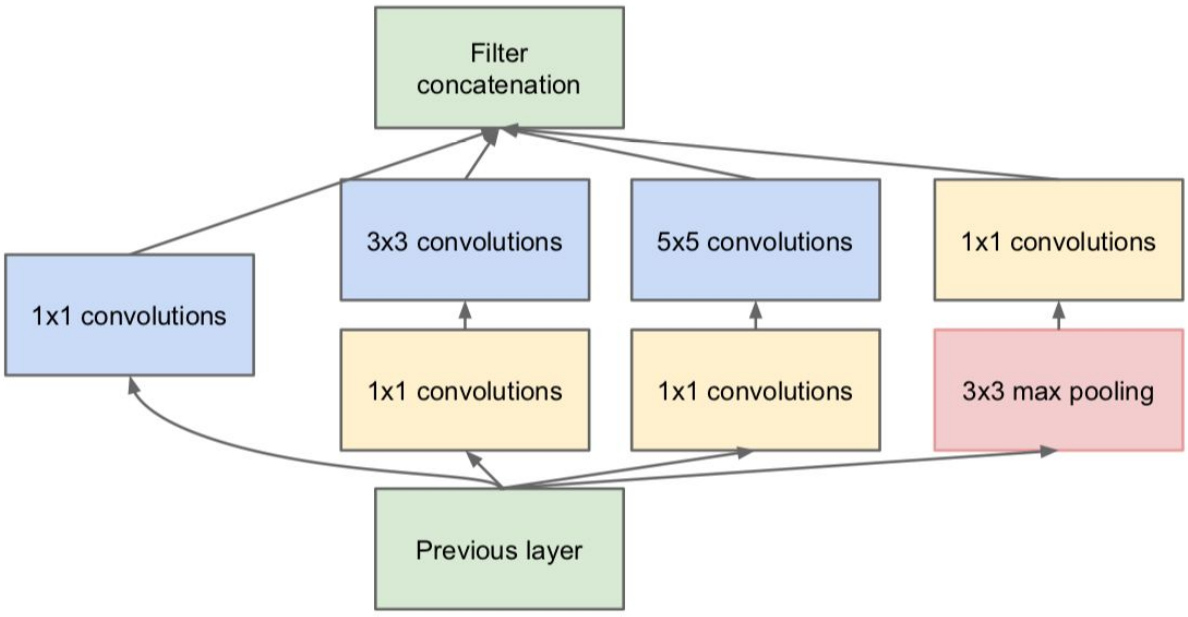


Figure 2. Inception Module (from original paper).

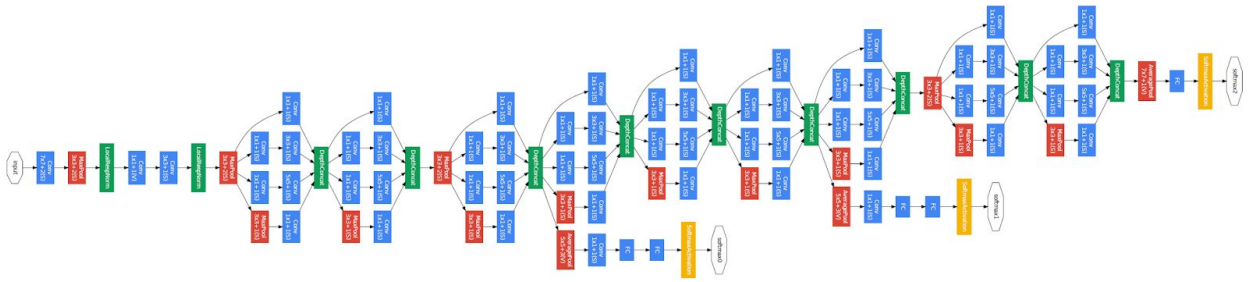


Figure 3. GoogLeNet network (From Left to Right)

There are 22 layers in total, and inception increases the network space from which the best network is to be chosen via training. Each inception module is able to capture salient features at different levels. Global features are captured by the 5x5 conv layer, while the 3x3 conv layer is prone to capturing distributed features. The max-pooling operation is responsible of capturing low level features that standout in a neighborhood. At a given level, all of these features are extracted and concatenated before it is fed to next layer. We leave for the network/training to decide what features hold the most values and weight accordingly. Say if the images in the data-set are rich in global features without too may low level features, then the trained Inception network will have very small weights corresponding to the 3x3 conv kernel as compared to the 5x5 conv kernel.

4. Experiments

All trains and tests are carried out on Google Colab, and all samples are trained with batch size 100 and epochs 20.

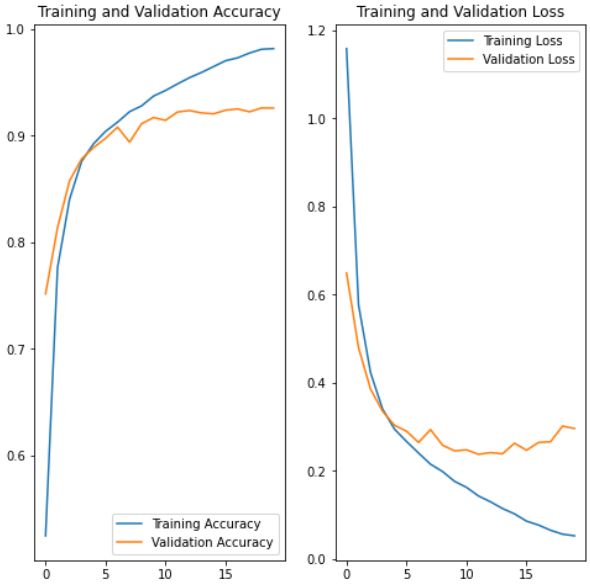
4.1 Datasets

I used Fashion-MNIST and CIFAR-10 datasets. Fashion-MNIST is a dataset of Zalando's article images. Each sample is a 28x28 grayscale image, associated with 10 classes. It consists of training sets of 60,000 and test sets of 10,000. CIFAR-10 datasets consist of 60000 32x32 colour images in 10 classes, with 6000 images per class, and images are color image.

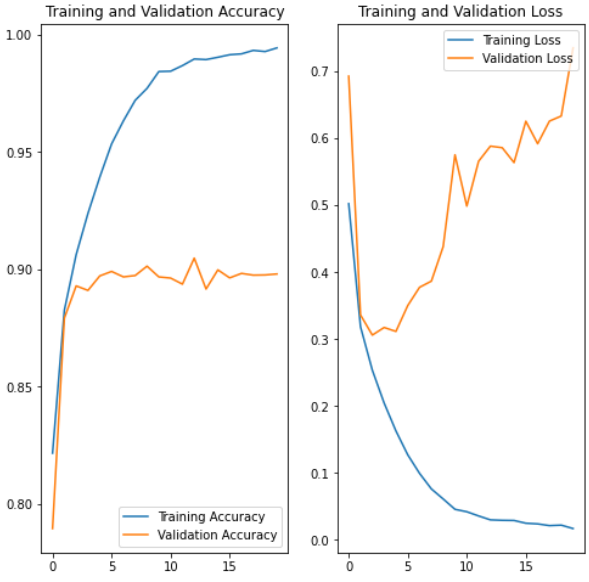
To train and test these small images, original models cannot be used. In the case of VGG16, the minimum size is 32x32. And for GoogLeNet, all samples are resized to 224x224 images. So both

models' pool-size of MaxPooling2D is modified. At first, I did not want this change. Before this modification, I put the preprocess that resizes all images to 32x32 for VGGNet and to 224x224 for GoogLeNet. However, while training the train samples, it continuously produces errors: GPU crashing, out of RAM. I spent 3 days because of this issue.

4.2 Testing results

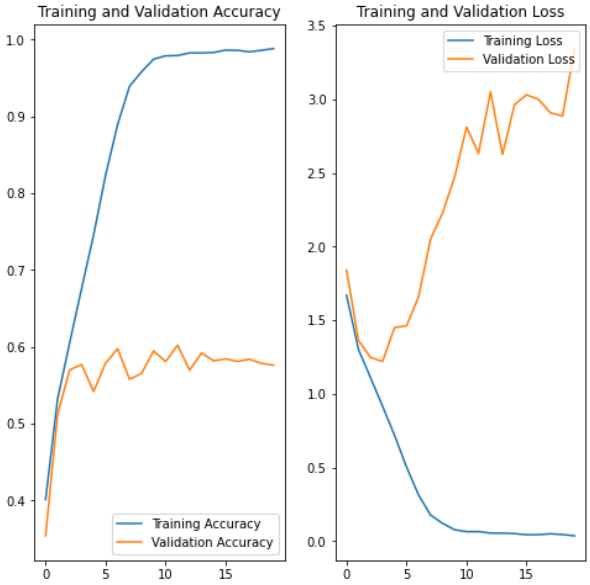


VGG16 with Fashion-MNIST



GoogLeNet with Fashion-MNIST

VGG16 with CIFAR10



GoogLeNet with CIFAR10

Figure 4. Training and Validation accuracy and loss for VGG16 and GoogLeNet

The accuracy and loss graph of the training and validation for VGG16 and GoogLeNet with two datasets can be seen in Figure 4. Graph shows that GoogLeNet reaches better accuracy in the early epochs compared with VGG16. With MNIST dataset, whereas VGG16 got 90% at 7 epochs, GoogLeNet reached at 3 epochs. Likewise, with CIFAR-10, accuracies reached 90% at 8 and 10 epochs respectively. This leads to less train time for GoogLeNet model. However, as epoch increases, GoogLeNet's validation loss for both datasets is dramatically increased.

	VGG-16		GoogLeNet	
	MNIST	CIFAR-10	MNIST	CIFAR-10
Accuracy	0.9244	0.7274	0.8928	0.5791
Loss	0.3034	1.37994	0.776	3.2879

Table 1. Accuracy and loss of test dataset

As you can see in table 1, VGG-16 shows better performance with test samples, and with CIFAR-10, accuracy is relatively lower than MNIST. I assume that the low accuracy of GoogLeNet is affected by low pool-size in MaxPooling2D layer. Compared with train accuracy, test accuracy shows relatively low figures. It could be solved by adding dropout layers between FC Layers.

5. Conclusion

In summary, the result of both model VGG-16 and GoogLeNet deeper convolutional neural networks with MNIST showed high accuracy, although some layers are modified to keep original image size. To solve the potential over-fitting problem and get better performance, I left the tasks of adding dropout layers and training with original input size as future works.

References

[1] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition.

[2] Christian Szegedy, Wei Liu, Andrew Rabinovich. Going deeper with Convolutions.

[3] Koustubh. ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks

[4] Anwar, Aqeel. Difference between AlexNet, VGGNet, ResNet and Inceptio, from https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96