IERG4160 Image and Video Processing

Project 2 Report

GUO Yang 1155029204
JIANG Zhehao 1155047133
LI Haochen 1155047102
WANG Tianming 1155029084

Abstraction

In this project, we have to train a deep neutral network to achieve attribute recognition in images. We use SUN Attribute Dataset for our high-level scene recognition, and have built two kinds of artificial neural networks, Residual Network (ResNet) and GoogLeNet. In addition, to improve the performance of networks, we test five more ResNet modules and two GoogLeNet modules. In this report, we will describe how we construct our neutral networks and how does it work.

Overview

The most common and straightforward way to improve the performance of neural network is by increasing its size and depth. Size refers to the complication level for each layer and depth refers to the number of layers. It's acknowledged that the deeper the convolutional neural network is established, the more fine-grained visual categories it may observed. However, the drawback arises that large and deep network not only consumes quadratic computing resources but also cause the problem of overfitting. The previous work dealing with these problems would be ultimately moving from the fully connected to sparse connected structure. Our experiment mainly explores two efforts which were made to achieve this goal including GoogLeNet and residual network. The first one makes use of the inception architecture, an extra sparsity structure which exploits the hardware by utilizing the computation on dense matrixes. Plus, the inception architecture always achieves the locally optimal. The latter one reformulates the layers as learning residual functions regarding the layer inputs and it's comprehended that this network can approach to the optimal by considerable increasing the depth.

Methodology

1. GoogLeNet v4

   The main idea of GoogLeNet is to find the optimal sparse structure by readily available dense components. Within the inception architecture the clusters are generated based on the correlation statistics of the previous layer and they form the units of the next layer. Each layer is formed from multiple units which concentrate on different region of the input image. High correlated clusters concentrate in a single local region and they can be covered by a one-by-one convolution layer in the next layer to perform dimension reduction. Besides, the filter size is restricted as 1x1, 3x3 and 5x5 to avoid the patch-alignment issue. The inception module is presented as:

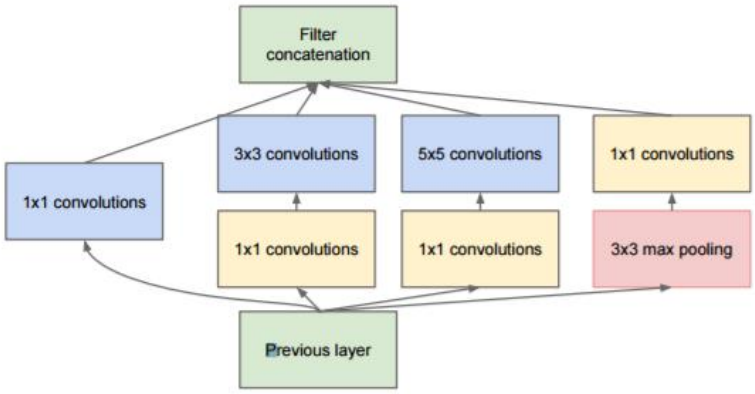
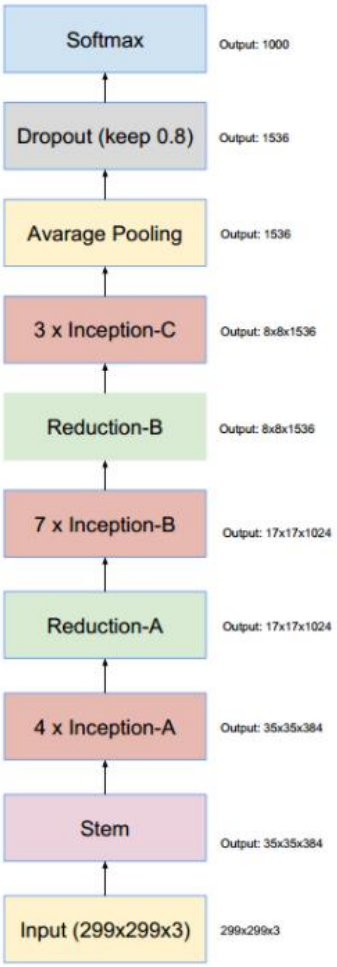


Fig. Inception module with dimension reduction

In the GoogLeNet, all the convolutions use a rectified linear activation function and the size of the receptive field is 299x299 taking the RGB color channels. The reduce block includes some 1x1 filters before the 3x3 or 5x5 convolution. This reduce block is used to increase the computational efficiency.

GoogLeNet v4 is the fourth version of GoogLeNet which is much more complicated than the previous versions. Author in [2] compares the performance of Inception-ResNet-v2 and Inception v4 and find out they have similar complexity as well as accuracy. Another structure called the residual connection is also introduced to increase the converge speed. What we observe is that the high complexity of GoogLeNet v4 network is the ultimate resource inducing high accuracy and performance. The following is the raw architecture of the GoogLeNet v4.

In our experiment, we adopt the design methodology of Inception v4 and develop our own convolutional neural network. Due to the limited computing power, we judiciously scale down the size of the network by decreasing the number of each inception cells. Our trials include (2, 2, 2) and (5, 10, 5) corresponding to the number of Inception-A, Inception-B and Inception-C. The result will be demonstrated in the result section.



Inception v4

2. Residual network

Deep residual network consists of many stacked "Residual Units" and each residual unit is expressed in a form as [3]:

$$y_l = h(x_l) + \mathcal{F}(x_l + W_l)$$
$$x_{l+1} = f(y_l)$$

Where x_l and x_{l+1} refer to the input and output of the l[th] unit and $\mathcal{F}$ is the residual function. In our trail, we choose $h(x_l)$ as an identity mapping and f as a ReLU function. The deep residual network framework was proposed to solve a degradation problem that with more layers stacked onto the network, the accuracy of the network may be affected and get saturated and then degrade rapidly. This problem is not caused by overfitting but the previous work observed that with more layers added up, the training error will also increase. Hence, the basic idea to solve this problem is by adding an identity mapping layer to other layers from learned shallower model, which theoretically will not introduce more training error than its shallower counterpart. A residual building

block is established as h as an identity mapping.

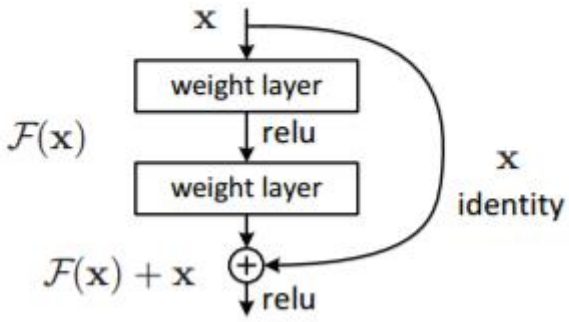


Fig. Building block

The whole network is an evolution of one traditional plain network which resembles the idea of VGG nets. The plain network includes convolutional layers mostly have 3x3 filters. The network follows two rules, one is that for the same output feature map size, the layers have the same number of filters and another is that if the feature map is halved, the number of filters will be doubled to preserve the same computing complexity for each layer. The evolved residual network embeds the identity shortcut connections which implements the addition part in the first expression with identity mapping. The architecture of the network is demonstrated as below:

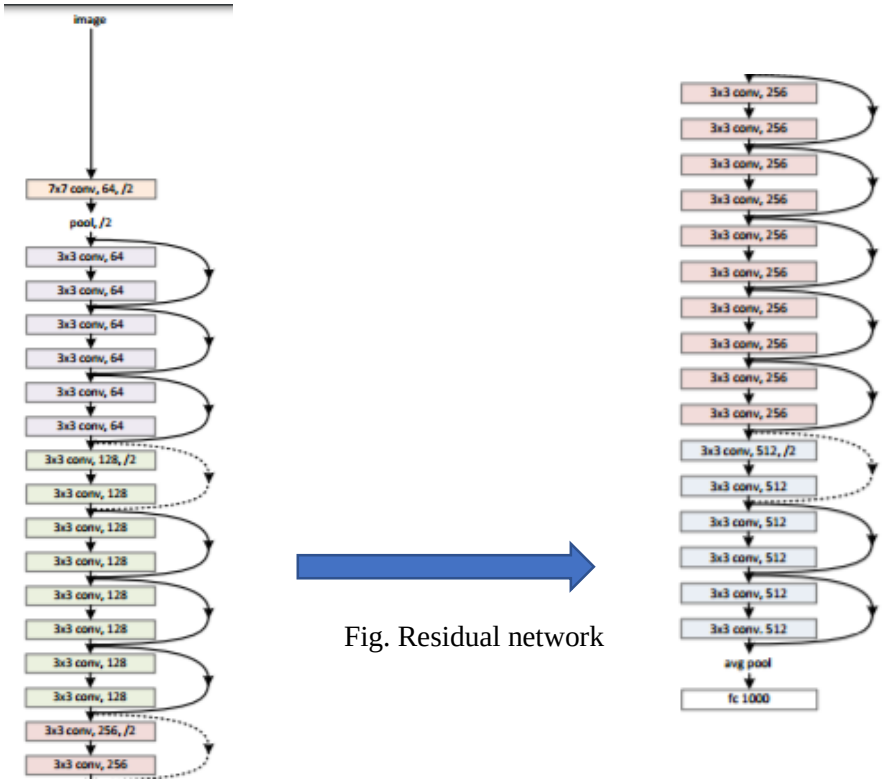


Fig. Residual network

Our experiment follows the structure proposed in [4] and due to the limited computing power, we judiciously adjusted the number of residual blocks to gain the best performance. With different number of layers the networks behave quite different and results are shown in the result section.

Results

In this section, we experiment with 5 ResNet systems: ResNet-18, ResNet-26 *Bottleneck*, ResNet-34, ResNet-50 and a 77 layer *Bottleneck* architecture (denoted as ResNET-77). We also design and implement 2 GoogLeNet systems: GoogLeNet-40 (40 layers) and GoogLeNet-101 (101 layers).

Classification accuracy (%) on the image dataset using different models are presented in Table below.

Methods	Classification accuracy (%)
ResNet-18	88.38
ResNet-26 *Bottleneck*	88.62
ResNet-34	88.03
ResNet-50 (baseline)	87.87
ResNet-77 *Bottleneck*	87.55
GoogLeNet-40	84.834
GoogLeNet-101	84.837

Experiment result shows that the ResNet models are consistently better than the GoogLeNet models in terms of classification accuracy. And we also find out that deeper models does not necessarily improve performance partially due to over-fitting issue. Finally, for all 7 models, ResNet-26 *Bottleneck* achieves the best classification result and is the model that we choose.

Reference

1. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).
2. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).
4. He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In *European Conference on Computer Vision* (pp. 630-645). Springer International Publishing.