UNDERSTANDING INCEPTION NETWORK ARCHITECTURE FOR IMAGE CLASSIFICATION

Technical Report · February 2020

DOI: 10.13140/RG.2.2.16212.35204

CITATIONS
8

READS
2,755

4 authors:

Tejas Pandit
University of Waterloo
6 PUBLICATIONS **18** CITATIONS
SEE PROFILE

Akshay Kapoor
University of Waterloo
5 PUBLICATIONS **9** CITATIONS
SEE PROFILE

Rishi Shah
University of Waterloo
2 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Rushi Bhuva
University of Waterloo
2 PUBLICATIONS **8** CITATIONS
SEE PROFILE

UNDERSTANDING INCEPTION NETWORK ARCHITECTURE FOR IMAGE CLASSIFICATION

Akshay Kapoor a43kapoo@uwaterloo.ca
Rishi Shah rr8shah@uwaterloo.ca
Rushi Bhuva rbhuva@uwaterloo.ca
Tejas Pandit tnpandit@uwaterloo.ca

University of Waterloo

ABSTRACT

The network architecture plays an important role in performance and speed of a deep network to classify images. In this paper we study the different architecture schemes and the variants proposed in GoogLeNet and inception networks. These variants are analyzed in terms of their computation efficiency and the network features and performances are juxtaposed on ImageNet 2012 dataset and critical review on inception networks is provided.

KEYWORDS

ImageNet Large Scale Visual Recognition Competition, Convolutional Neural Network, Residual Neural Network, Inception, GoogLeNet, Image Classification, Image Recognition, Batch Normalization, Dropout, Pooling.

1. INTRODUCTION

We are in the headways of development of intelligent systems such as robotics, IoT (Internet of Things), computer vision, etc. in which image classification and detection help us accomplish key roles. As we ameliorate the image classification phenomenon its success will also be reciprocated in object detection, segmentation[14], human pose estimation[15], video classification[17], object tracking[16], super resolution and the list goes on. In this paper we discuss, the concept of inception networks, features of GoogLeNet inception networks, Limitations and challenges faced by some of the architectural schemes used in inception networks. The performance of inception variants is measured on ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 Dataset.

2. INTUTATION BEHIND INCEPTION NETWORK

Going deeper with convolutions[1] introduced the concept of forming dense layers by concatenating the sparse layers. This idea was taken from[6] and resonates well with Hebbian principle, though the calculations for inception networks did not match strict mathematical grounds for its proof. Using multiple windows to form an inception layer also aligns with the intuition that visual information should be processed at various scales and then aggregated so that the next stage can abstract features from different scales simultaneously.

3. ARCHITECTURAL DETAILS

Typical architecture of CNN comprises of convolution layer followed by sub-sampling (i.e. pooling layer) layer. The convolutional layers segregate the features of the image. Sampled features are classified by the fully connected layer followed by the output layer. In order to get a better accuracy network, the general approach is to increase depth of network which causes issues of vanishing gradients and demands more computational power. To deal with these problems [1] introduced the concept of auxiliary units and inception layers in GoogLeNet inception network. Introduction to 1x1 convolution in inception layers helped not only saving the computational power but also in making the network wider[7]. Impact of introducing 1x1 convolution on computational cost can be depicted from Figure 2.

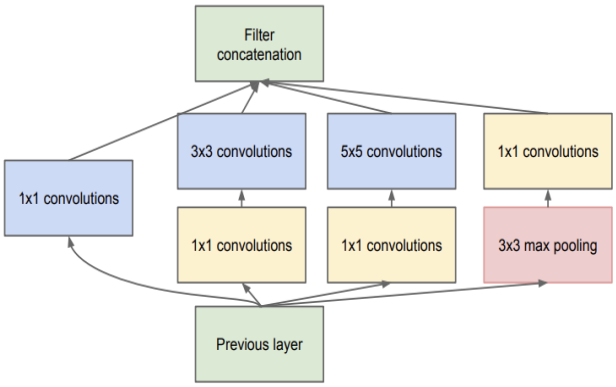

Figure 1. An inception layer of GoogLeNet Network.[1]

[1] introduced the concept of inception layers, forming a concatenated layer using stacks of 1x1 convolutions, 1x1 followed by 3x3 convolutions, 1x1 followed by 5x5 convolutions and 3x3 max pooling layers followed by 1x1 convolutions (see Figure 1). Also, auxiliary layers were

added to prevent zero vanishing of gradients. Further studies on the auxiliary layers are addressed in section 4 of this paper.

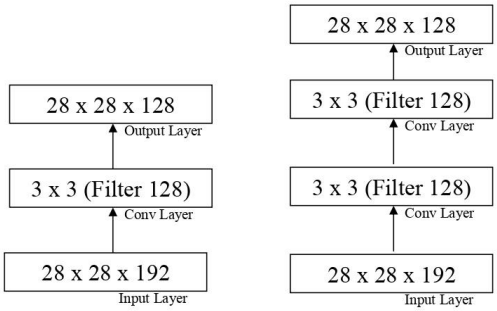

Figure 2. The image on the left, shows conventional 3x3 convolution resulting in 221.18K (3x3x192x128) multiplications, the image on the right uses 1x1 layer before convoluting it with 3x3 window resulting in 86K ((1x1x192x64) + (3x3x64x128)) multiplications.

The contributing factors for performance of GoogLeNet architecture were not clearly described in [1], that lead to the introduction of design principles in [2]. Following principles were used as a base for the different inception architecture (see Figure 3.; Figure 4.; Figure 5.):

- Avoid representational bottlenecking at the earlier layer to prevent information loss from the input images.
- Higher dimensional representation can easily be operated regionally within the network, also increasing receptive field will provide disentangled features.
- Spatial aggregation can be done by lower dimensional filter if strong correlation is present between adjacent activation.
- The optimal improvement with constant computational cost can be reached if width and depth are increased parallelly.

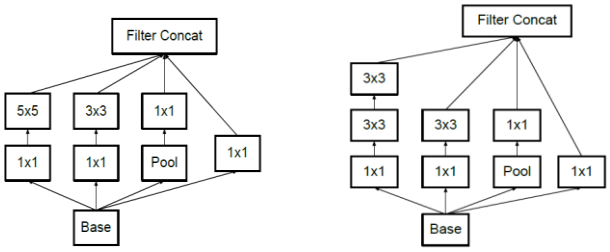

Figure 3. Factorization of a 5x5 filter (i.e. 25 operations in one time) into two 3x3 filters (18 operations in one time). 5X5 window replaced by two 3x3 filter and can save computational expenses by a factor of 0.72, while extracting similar nonlinear parameters from receptive field.[2]

Typically, convolutional layers are followed by the pooling layers to reduce the grid dimension which arise the problem of representational bottleneck (see Figure 6.). To get rid of it [2] discusses efficient grid size reduction techniques (see Figure 7.), provided adjacent activations are highly correlated. This reduces the grid dimensions with expansion in a channel depth.

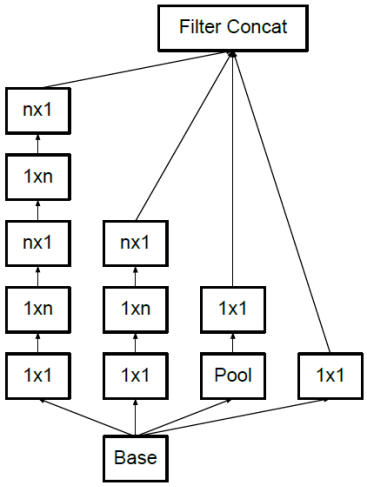

Figure 4. Spatial factorization into asymmetric convolutions as can be seen nxn can be further reduced to 1xn followed by nx1. It also be understood that kernels of size nxn (n>3) might not be used, as they can be further broken into smaller networks.[2]

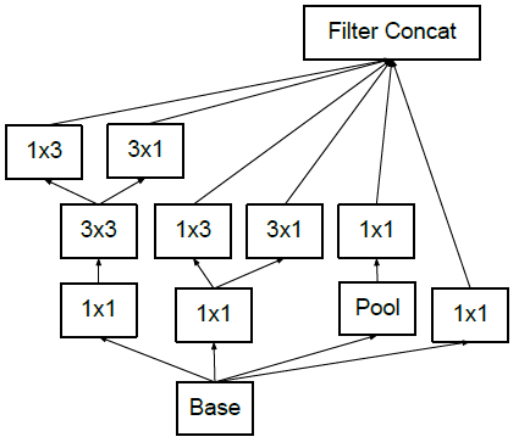

Figure 5. It is mentioned in inception version 3 that above picturized inception layer architecture is to be applied at coarsest grids to encourage the high dimensional representation, as it is suggested in general design principle that singular inception layer can easily process the higher

dimensional feature maps. Increasing receptive field per window gives more disentangled features.[2]

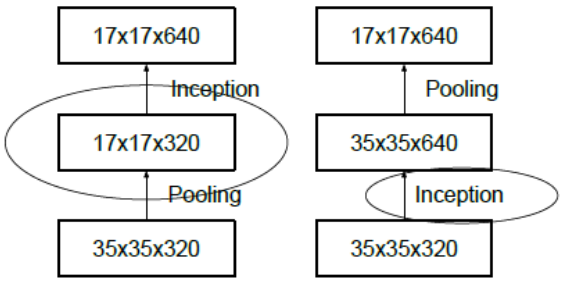

Figure 6. In the left image, application of max pooling followed by inception layer produces representational bottlenecking, while image in the right exhibits application of inception followed by max-pooling, which results in three times more expensive computation.[2]

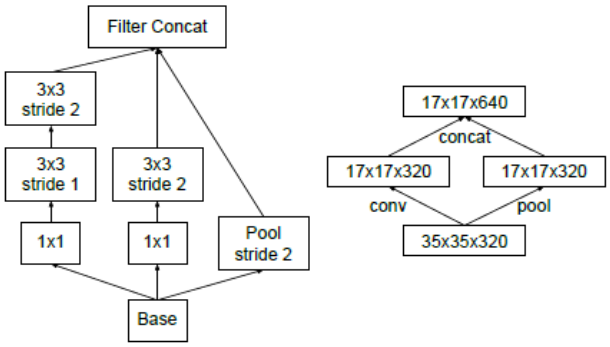

Figure 7. As visualized in the above image reduction layer reduces receptive field with expansion of the channel depth. It is computationally cheap and prevents the representational bottleneck.[2]

Another network design concept; [3], was able to solve the problem of vanishing gradients and was able to train very deep layer networks, winning the ILSVRC 2015 with 3.57 % Top-5 error rate. These models used shortcut connection which helped the deep network to learn the identity mapping, which in turn helped solving the degradation problem of deep networks.

Incorporating the identity shortcut connection in their inception architecture google came up with two new hybrid networks named Inception-ResNet-v1 and Inception-ResNet-v2 with replacement of auxiliary classifier by drop out compared to inception-v2. Inception-ResNet-v2 has shown 4.9% top-5 error on ILSVRC 2012 dataset. The idea behind inception-v4 was to match the performance of inception ResNet v2 without using shortcut connection. The performance of both inception ResNet-v2 and v4 can be compared from Table 1.

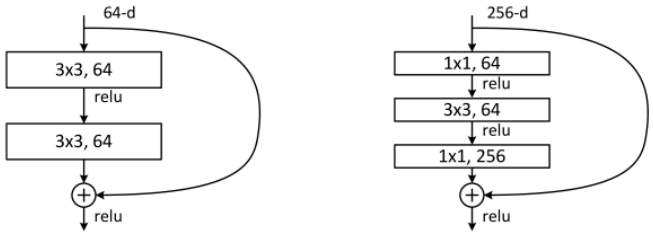

Figure 8. Shortcut connection used for identity mapping.[3]

4. AVERAGE POOLING

Global average pooling was used along with dropout to prevent overfitting. In [1], this resulted in error being reduced by 0.6 % compared to the case when last layer was used as a fully connected layer. The use of average pooling followed by the 1X1 convolution with in the inception layer in inception-v4 depicts similar effect as the one provided by the shortcut connections in Inception-ResNet variants.

Architecture	**Top-5** Error
Inception-ResNet-v2	4.9
Inception-v4	5.0
Inception-ResNet-v1	5.5
Inception-v3	5.6
Inception-v2 Factorized 7x7	5.8
Inception-v2 Label Smoothing	6.1
Inception-v2 RMSProp	6.3
BN-Inception	7.8
GoogleNet	7.89

Table 1. Performance of various inception architecture on ImageNet 2012 dataset.

5. UTILITY OF AUXILIARY CLASSIFIER

Auxiliary classifiers were introduced in the Inception-v1 to improve the convergence of the deep layer and solve the problem of vanishing gradient. interestingly, it was found that the auxiliary classifiers did not result in improved convergence in the early part of the training, but they help before accuracy gets saturated. It was observed that the removal of the lower auxiliary branch did not have any adverse effect on the final quality of network. Finally, it was argued that the auxiliary classifier acts as the regularizer. This was supported by the fact that the main classifier of the network performs well if auxiliary branch is batch normalised or has dropout layer.

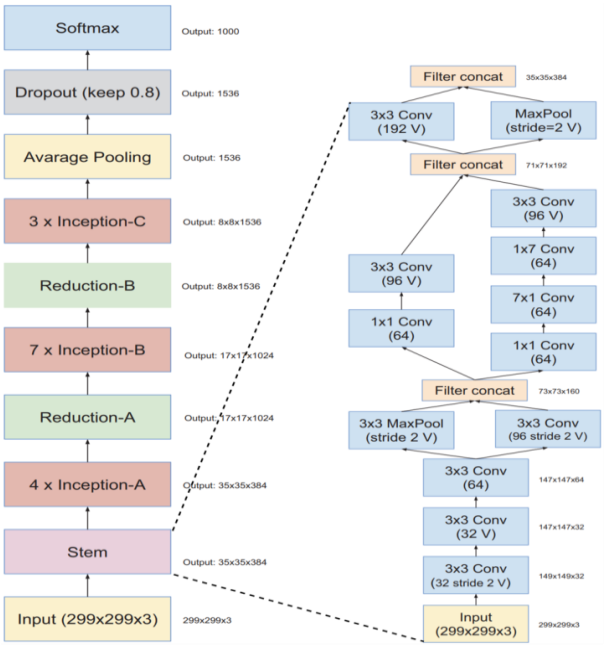

Figure 9. Overall structur and Detailed stem of Inception-v4.[4]

6. PERFORMANCE OF VARIOUS VARIANTS OF INCEPTION NETWORKS

Various models were derived from the GoogLeNet and ResNet architectures, in these models some improvements were made such as application of batch normalization[8], label smoothing[10], addition of average pooling layers in the inception and reduction layers [4], and refined training methodologies were used. The performance of all these models was measured on ILSVRC 2012 [11] classification dataset as top 5% error as can be seen from the table. The table shows performance of GoogLeNet, performance of Inception-v2 with momentum, RMS prop [12], label smoothing[10]. Use of these schemes along with use of factorization improved the performance of Inception-v2 compare to GoogLeNet model by 26.49 %. Further, introduction of batch normalization in auxiliary classifier in Inception-v2 network, i.e. Inception-v3 network provided better performance of 5.6 %. Inception-ResNet-v1 and Inception-ResNet-v2 were constructed to improve performance without having degradation in deep layer architecture, which caused a further reduction in error rate. Inception-v4 was built to imitate the performance of Inception-ResNet-v2 and was just short by 0.1 %.one concluding line.

7. CONCLUSION

We have reviewed GoogLeNet and Inception-v2, Inception-v3, Inception-v4 network performances and compared their architectures. The application of these state-of-the-art inception layers has improved the accuracy of CNN based models significantly. Not only, these networks were able to improve the performance but also were able to reduce the computational cost compared to their conventional predecessors. The implementation of these inception schemes faces issues of generalization and do not provide clarity on some of the experiments performed such as placement and use of various variants within a CNN architecture. Despite irregularities in the above described explanation, these networks exhibit promising research, and various models incorporating these variants have performed spectacularly in image classification challenges.

8. REFERENCES

[1] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015a. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–9.

[2] Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2015b. Rethinking the inception architecture for computer vision. arXiv preprint arXiv:1512.00567.

[3] He, K.; Zhang, X.;Ren, S.;Sun, J.; Deep Residual Learning for Image Recognition. arXiv:1512.03385v1 [cs.CV] 10 Dec 2015.

[4] Szegedy, C.; Ioffe S.; Vanhoucke, V.; Alexander A.; Alemi. Inception-v4, Inceptioin-ResNet and the Impact of Residual Connections on Learning.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia,R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man´e, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, sJ. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vi´egas, O. Vinyals, P. Warden, M.Wattenberg, M.Wicke, Y. Yu, and X. Zheng. Tensor-Flow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[6] Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. CoRR, abs/1310.6343, 2013.

[7] Min Lin, Qiang chen, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013.

[8] Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167v3 [cs.LG] 2 Mar 2015.

[9] Simyon, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[10] Muller, R.; Kornblith, S.; Hinton, G. When Does Label Smoothing Help? arXiv:1906.02629v1 [cs.LG] 6 Jun 2019.

[11] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2014. Imagenet large scale visual recognition challenge.

[12] T. Tieleman and G. Hinton. Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 4, 2012. Accessed: 2015-11-05.

[13] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning (ICML-13), volume 28, pages 1139–1147. JMLR Workshop and Conference Proceedings, May 2013.

[14] J. Long, E. Shelmer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 3431-3440, 2015.

[15] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014.

[16] N. Wang and D.-Y. Yeung. Learning a deep compact image representation for visual tracking. In Advances in neural Information Processing Systems, pages 809-817, 2013.

[17] A. Karpathy, G. Toderici, S. Shetty, T Vision and Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 1725-1732. IEEE, 2014.