Going Deeper with Convolutions

Comment

Abstract

- GoogLeNet, 22-layers deep network
- Hebbian principle and multi-scale processing

1. Introduction

- 12x fewer parameters than *AlexNet*, significantly more accurate
- Power and memory use are efficient – mobile, embedded
- Logical culmination of *NIN*

2. Related Work

- Despite concerns that max-pooling layers result in loss of accurate spatial information, successfully employed for localization, object detection and human pose estimation
- In *NIN*, additional 1x1 convolutional layers are added to network, increasing its depth, which helps dimension reduction, or increasing the depth and width of networks without significant performance penalty
- For object detection, R-CNN decomposes the process into two subproblems
  - Utilize low-level cues such as color and texture to genearte object location proposals in color-agnostic fashion
  - Use CNN classifiers to identify object categories at those locatdons They adopted a similar pipeline but had explored enhancements in both stages, such as multi-box prediction for higher object bounding box recall, and ensemble approaches for better categorization

3. Motivation and High Level Considerations

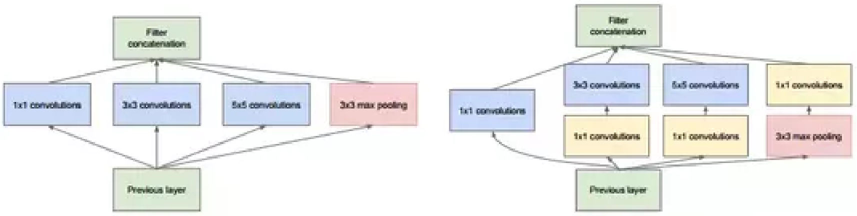
- The most straightforward way of imporving the performance is by increasing networks' size, which needs much more dataset in order to avoid overfitting problem. However the dataset is laborious and expensive to obtain.
- Uniformly increased network size results in quadratic increase of computation due to chained convolutional layers.
- To solve both issues, replace the FC by the ***sparse*** ones

- Thanks to the work of Arora et al. and Hebbian principle – neurons that fire together, wire together – the underlying idea is applicable even under less strict conditions.

- Unfortunately, those (non-uniform) ***sparse*** is very inefficient due to the dominent overhead of lookups and cache misses
- CNN have traditionally used random and sparse connection tables since in order to break the symeetry and improve learning.
- **They suggest that clustering space matrices into relatively dense submatrices tends to give competitive performance for sparse matrix multiplication.**
- Inception is proved to be especially useful in the context of localization and object detection as the base network for *multibox* and *R-CNN*
- Still cautious of the guiding principles that have lead to its construction

4. Architectural Details

- Arora et al. suggests a layer-by layer construction where one should analyzze the correlation statistics of the last layer and cluster them in to groups of units with high correlation.
- Assumed each unit from an earlier layer corresponds to some region of the input image and these units are grouped into filter banks. Thus, end up with lots of clusters concentrated in a single region and they can be covered by 1x1 convolutions (as suggested in *AlexNet*)
- The correlation statistics are bound to vary: as features of higher abstraction are caputred by higher layers, their spatial concentration is expected to decrease, which indicates more usage of 3x3 and 5x5 convolutions in higher layers.



(a) Inception module, naïve version

(b) Inception module with dimension reductions

Figure 2: Inception module

Figure 1:

5. GoogLeNet

- An incarnation of the Inception architecture for ILSVRC 2014
- Slightly tuned and improved marginally
- 22 layers deep (27 if counting pooling)
- Removed FC to average pooling, still with essential dropout
- Auxiliary network was needed to combat the vanishing gradient problem but discarded at the inference time.

6. Training Methodology

7. ILSVRC 2014 Classification Challenge Setup and Results

8. ILSVRC 2014 Detection Challenge Setup and Results

9. Conclusions

- approximating the expected optimal sparse structure by readily available dense building blocks
- For object detecion, desptite not utilizing context nor performing bouding box regression, they suggest yet further evidence of strengths of the Inception architecture
- For both clasification and detection, their approach yields solid evidence that moving to sparser architectures is feasible and useful idea in general