Network In Network architecture: The beginning of Inception

2017-10-20 • deep learning • Comments
By Anand Saha

Introduction

In this post, I explain the `Network In Network` paper by Min Lin, Qiang Chen, Shuicheng Yan (2013). This paper was quite influential in that it had a new take on convolutional filter design, which inspired the Inception line of deep architectures from Google.

Motivation

Anyone getting introduced to convolutional networks first come across this familiar arrangement of neurons designed by Yann LeCun decades ago:

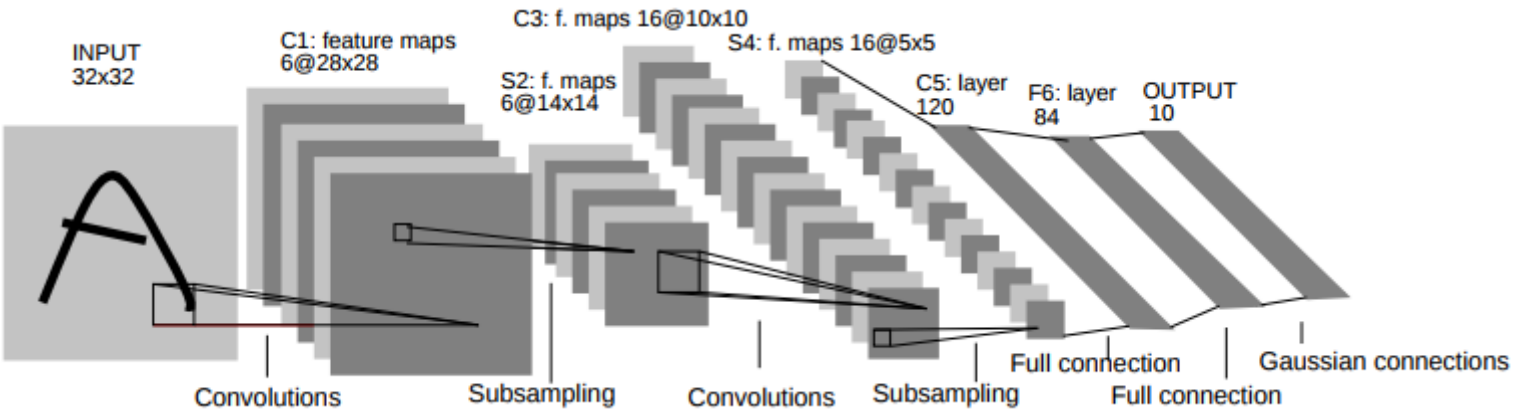

Fig. LeNet-5

Yann LeCun's work (1989, 1998) triggered the convolutional approach, which takes into account the inherent structure of the incoming data (mostly image data) while propagating them through the network and learning about them.

If you need a CNN refresher, this is an excellent read.

The idea of convolution is very simple and a genius one. Images have spatial information (height, width, and channels) and this arrangement of pixels is important to understanding their contents. Conventional neural networks would flatten their input before applying the weights, thereby losing the spatial information.

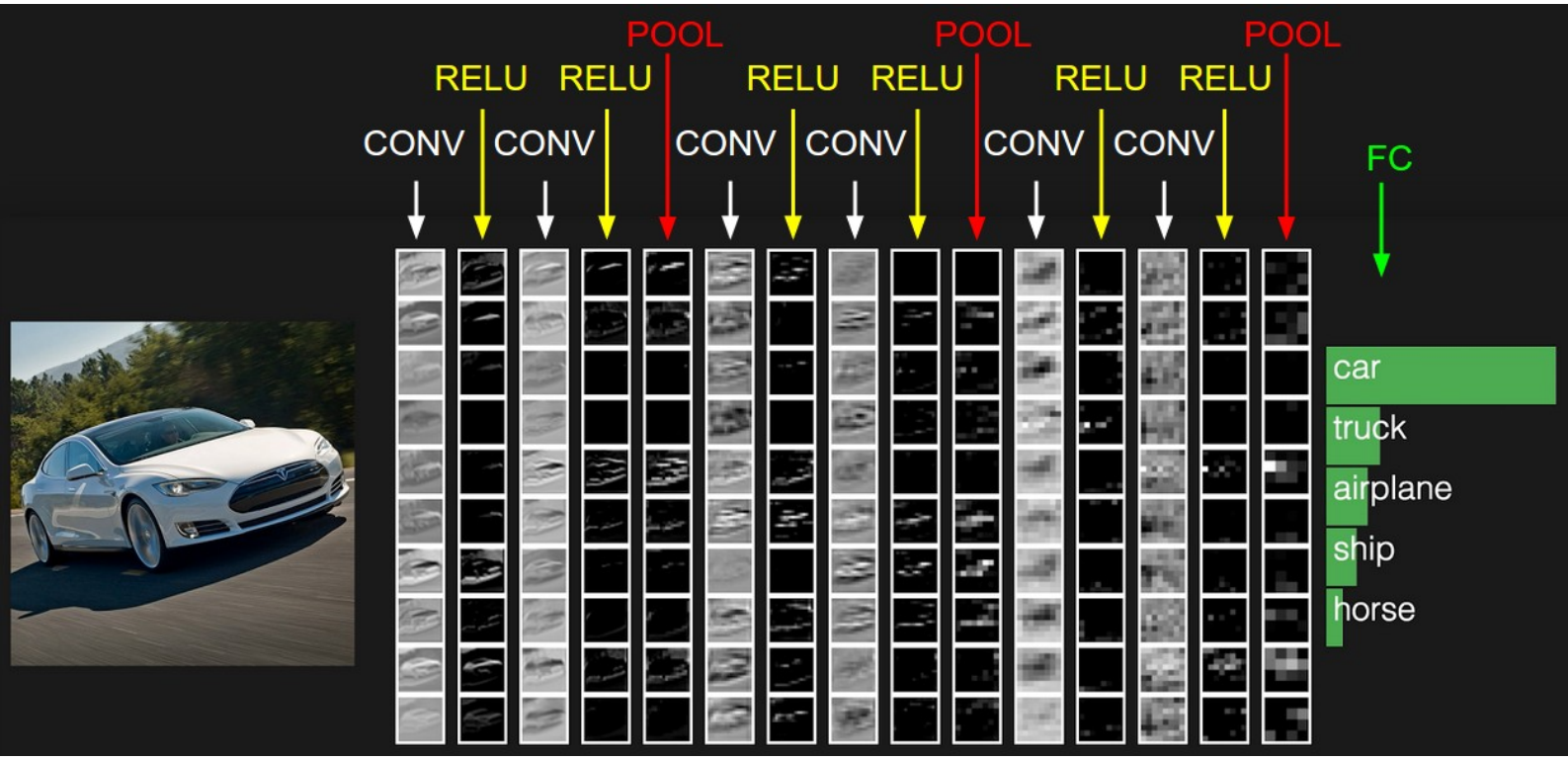

Fig. The typical arrangement of various layers in a CovNet (Img Credit: cs231n)

Convolutional networks, on the other hand, operate directly on the images as is. The filters (also called kernels) are moved across the image left to right, top to bottom as if scanning the image and weighted sum of products are calculated between the filter and subset of the image the filter is superimposing on. This is the convolution operation. What is to be noted here is that the operation is *linear*. Of course these are then passed through various other operations like non-linear activations and pooling.

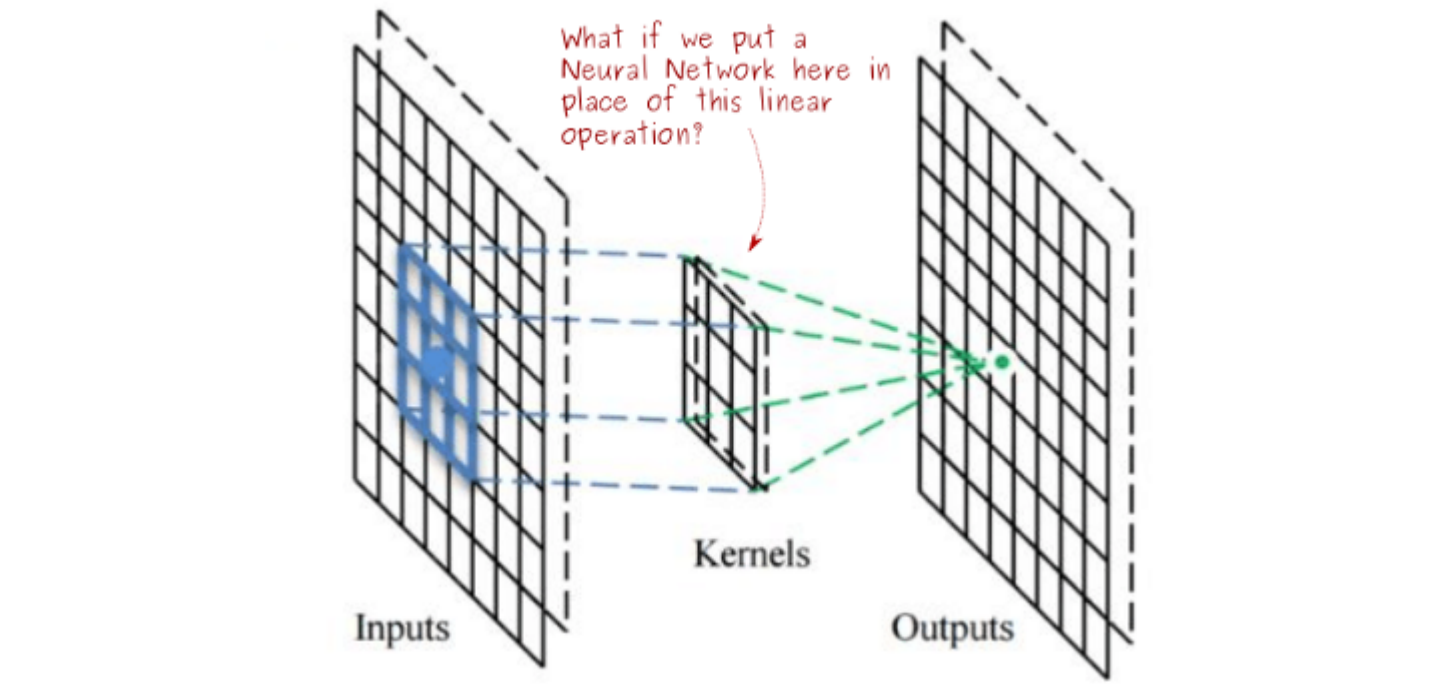

Fig: Conventional Convolution operation, and an idea (Img Credit: ResearchGate with my annotations)

The aspect to note in the convolution operation is the *linearity* of the operation. Does it need to be linear? Can it be something else that can extract richer features?

The idea of Network In Network

This paper had a new take on how the convolution filters are designed and how we map extracted features to class scores. This formed the basis of the Inception architecture. Two new concepts were introduced in this CNN architecture design:

- **MLPconv**: Replaced linear filters with nonlinear `Multi Linear Perceptrons` to extract better features within the receipt field (see the figure above). This helped in better abstraction and accuracy.
- **Global Average Pooling**: Got rid of the fully connected layers at the end thereby reducing parameters and complexity. This was replaced by the creation of as many activation maps in the last layer as there are classes. This was followed by averaging these maps to arrive at final scores, which is passed to softmax. This is performant and more intuitive.

MLPConv

Traditional CNN architectures use linear filters to do the convolution and extract features out of images. The early layers try to extract primitive features like lines, edges, and corners, while the later layers build on early layers and extract higher-level features like eyes, ears, nose etc. These are called latent features.

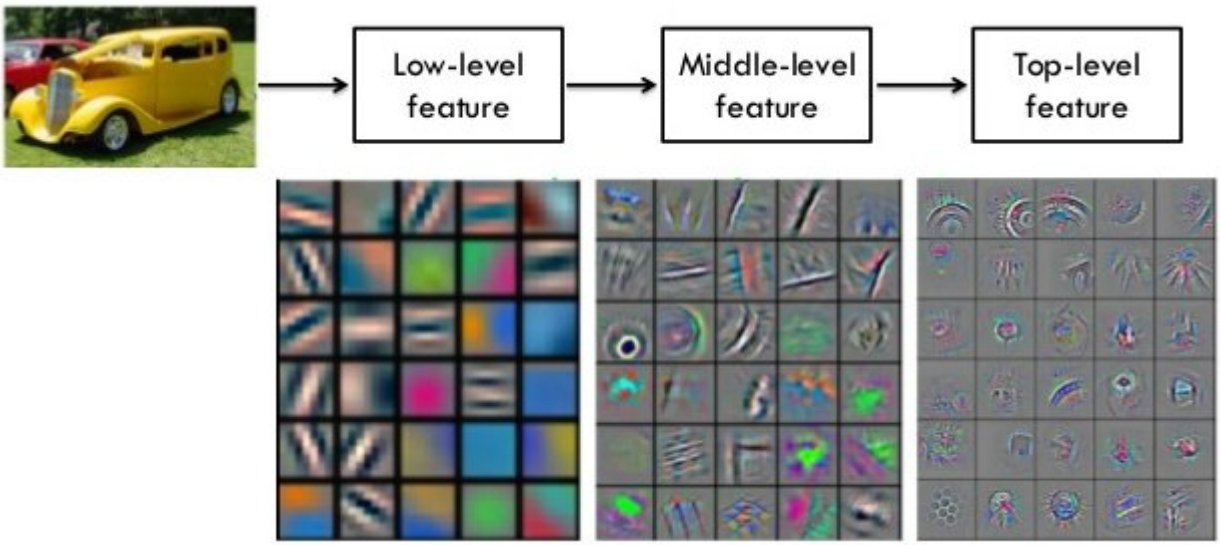

Fig.Hierarchy of features extracted from various layers (Img Credit: https://arxiv.org/abs/1311.2901)

Now, there can be variations in each of those features - there can be many different variations in eyes alone for e.g.. A linear filter (for e.g. to detect eyes) tries to draw straight lines to extract these features. Thus conventional CNN implicitly makes the assumption that the latent concepts are linearly separable. But a straight line may not always fit. The separation of the various types of eye features and non-eye features may not be a straight line. Using a richer nonlinear function approximator can serve as a better feature extractor.

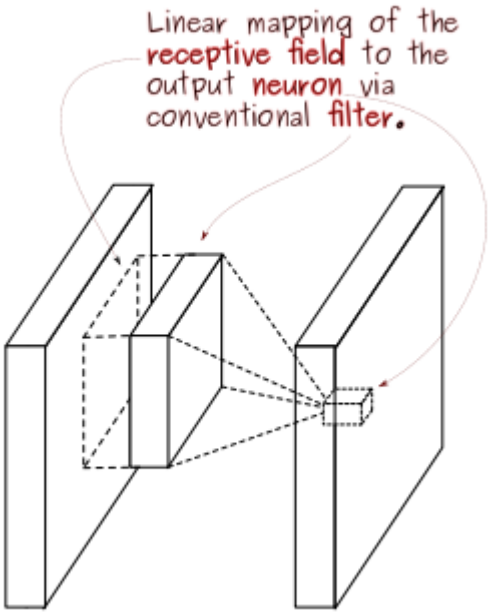

Fig. Conventional linear convolution layer (Img Source: https://arxiv.org/abs/1312.4400)

This paper introduced the concept of having a neural network itself in place of a convolution filter. The input to this mini network would be the convolution, and the output would be the value of a neuron in the activation. Hence it does not alter the input/output characteristics of traditional filters. This mini network, called MLPconv, can then convolved over the input. The benefit of having such an arrangement is two-fold:

- It is compatible with the backpropagation logic of neural nets, thus this fits well into existing architectures of CNN's
- It can itself be a deep model leading to rich separation between latent features

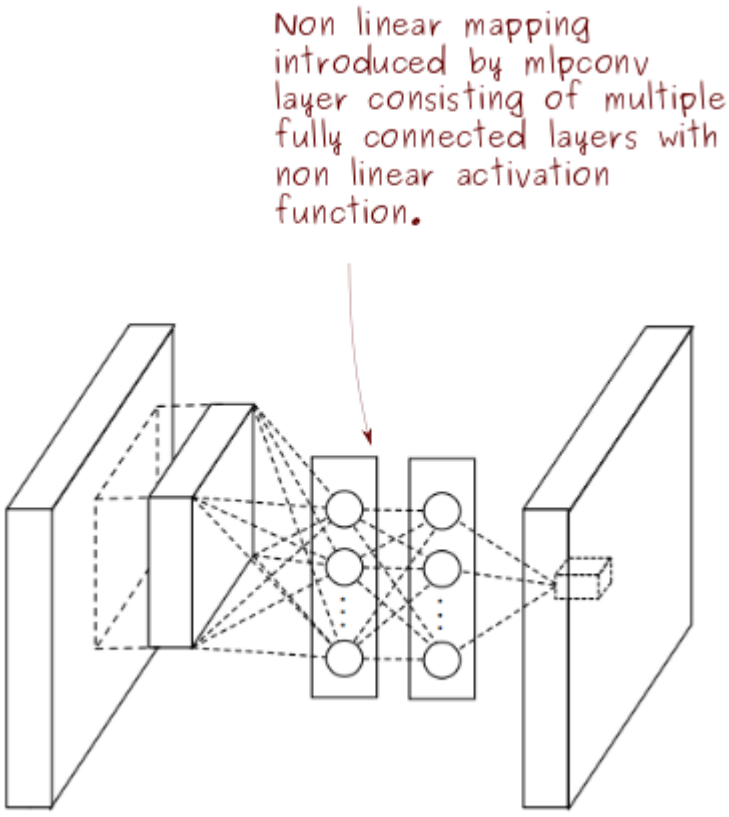

Fig. MLPconv layer (Img Source: https://arxiv.org/abs/1312.4400)

Global Average Pooling

In traditional CNN architectures, the feature maps of the last convolution layer are flattened and passed on to one or more fully connected layers, which are then passed on to softmax logistics layer for spitting out class probabilities. The issue with this approach is that it is hard to decode how the usual fully connected layers seen at the end of CNN architectures map to class probabilities. They are black boxes between the convolution layers and the classifier. They are also prone to overfitting and come with lots of parameters to train. An estimate says that the last FC layers contain 90% of the parameters of the network.

Can we do better?

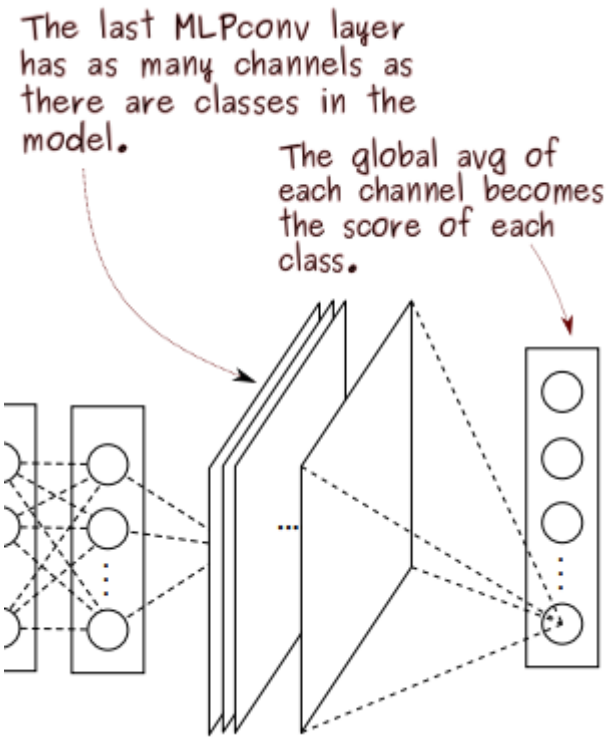

Fig. Global Average Pooling (Img Source: https://arxiv.org/abs/1312.4400)

In the approach proposed by the paper, the last MLPconv layer produces as many activation maps as the number of classes being predicted. Then, each map is averaged giving rise to the raw scores of the classes. These are then fed to a SoftMax layer to produce the probabilities, totally making FC layers redundant.

The advantages of this approach are:

- The mapping between the extracted features and the class scores is more intuitive and direct. The feature can be treated as category confidence.
- An implicit advantage is that there are no new parameters to train (unlike the FC layers), leading to less overfitting.
- Global average pooling sums out the spatial information, thus it is more robust to spatial translations of the input.

Conclusion

The paper `demonstrated the state-of-the-art classification performances with NIN on CIFAR-10 and CIFAR-100, and reasonable performances on SVHN and MNIST datasets` but more importantly gave a new direction to the design of convolution filters.

References:

- https://arxiv.org/abs/1312.4400
- https://openreview.net/forum?id=ylE6yojDR5yqX

Please leave a comment below if anything was unclear or can be improved in the post.

#deep learning #architecture #inception

< Decoding the ResNet architecture

Class Notes: Computer Vision (Georgia Tech) >