ALEXNET

Introduction

AlexNet is a deep convolutional neural network (CNN) architecture that revolutionized computer vision by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet significantly improved image classification performance using deep learning.

AlexNet famously won the 2012 ImageNet LSVRC-2012 competition by a large margin (15.3% vs 26.2%(second place) error rates). Here is the link to original paper.

Major highlights of the paper

1. Used ReLU instead of tanh to add non-linearity.
2. Used dropout instead of regularization to deal with overfitting.
3. Overlap pooling was used to reduce the size of the network.
4. **Input**

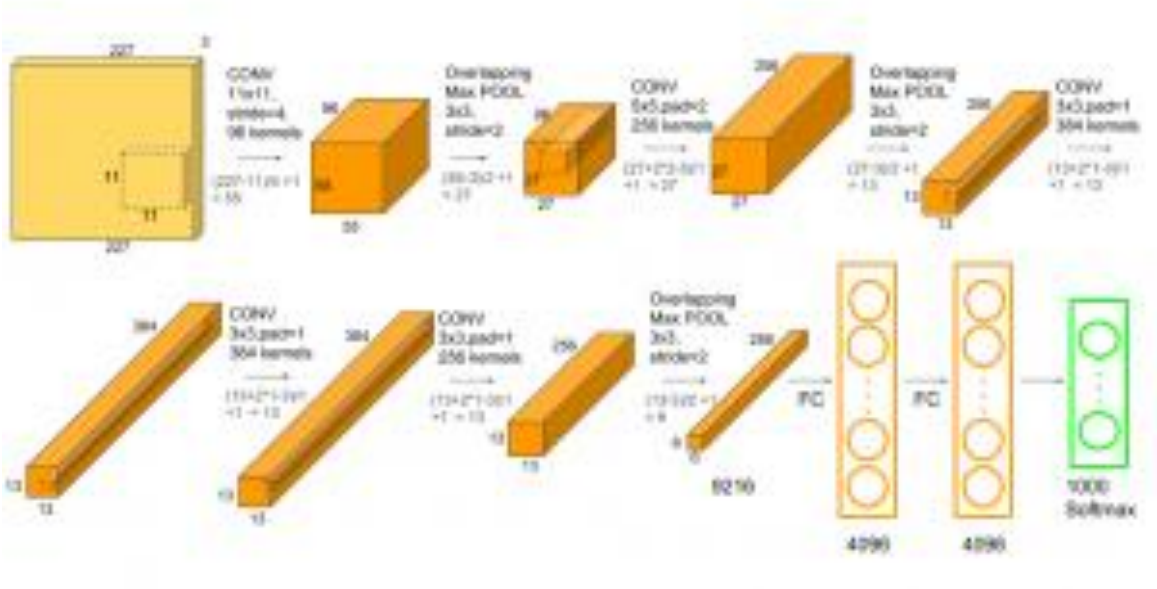
AlexNet solves the problem of image classification with subset of ImageNet dataset with roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. The input is an image of one of 1000 different classes and output is a vector of 1000 numbers.

The input to AlexNet is an RGB image of size 256*256. This mean that all the images in training set and test images are of size 256*256. If the input image is not 256*256, image is rescaled such that shorter size is of length 256, and cropped out the central 256*256 patch from the resulting image.

The image is trained with raw RGB values of pixels. So, if input image is grayscale, it is converted into RGB image . Images of size 257*257 were generated from 256*256 images through random crops and it is feed to the first layer of AlexNet

AlexNet Architecture

AlexNet contains five convolutional layers and three fully connected layers — total of eight layers. AlexNet architecture is shown below:

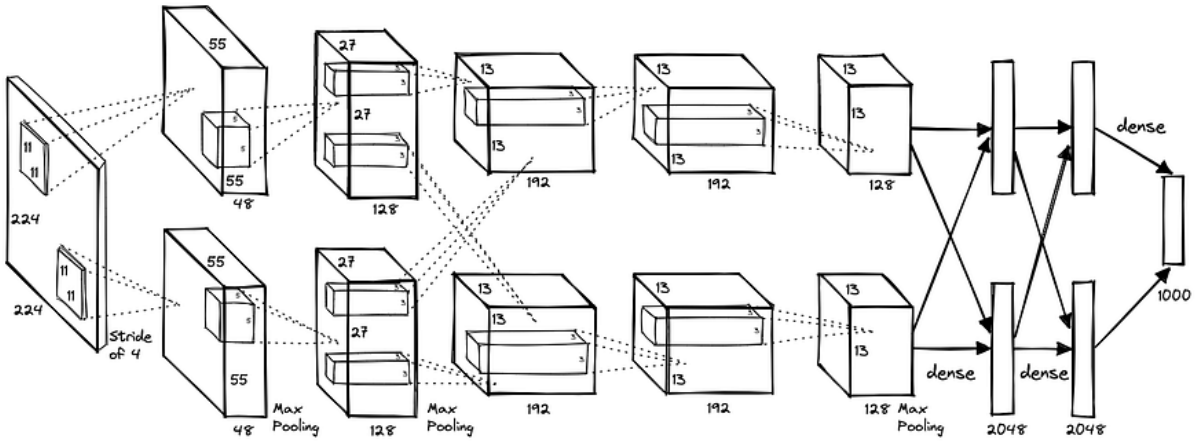


For the first two convolutional layers, each convolutional layers is followed by a Overlapping Max Pooling layer. Third, fourth and fifth convolution layers are directly connected with each other. The fifth convolutional layer is followed by Overlapping Max Pooling Layer, which is then connected to fully connected layers. The fully connected layers have 4096 neurons each and the second fully connected layer is feed into a softmax classifier having 1000 classes.

1. **Five Convolutional Layers**: These layers extract hierarchical features from images, starting from edges and textures in early layers to shapes and complex patterns in later ones.

2. **Three Fully Connected Layers**: After feature extraction, these layers learn to classify based on the detected features.
3. **ReLU Nonlinearity and LRN Layers**: ReLU activation after each convolutional layer accelerates training, while LRN improves generalization.
4. **Max Pooling Layers**: Placed after certain convolutional layers to reduce spatial dimensions and computation.
5. **Dropout in Fully Connected Layers**: Dropout improves model robustness by making neurons independent, which helps prevent overfitting.

The architecture takes 227x227x3 images as input, chosen to match the original model. Let's dive into the setup and implementation.



Applications

- **Image Classification**: Achieves high accuracy on ImageNet and other datasets.
- **Feature Extraction**: Used as a feature extractor in transfer learning.
- **Object Detection**: Forms the backbone of some object detection models.
- **Medical Imaging**: Applied to tasks like tumor classification and disease detection

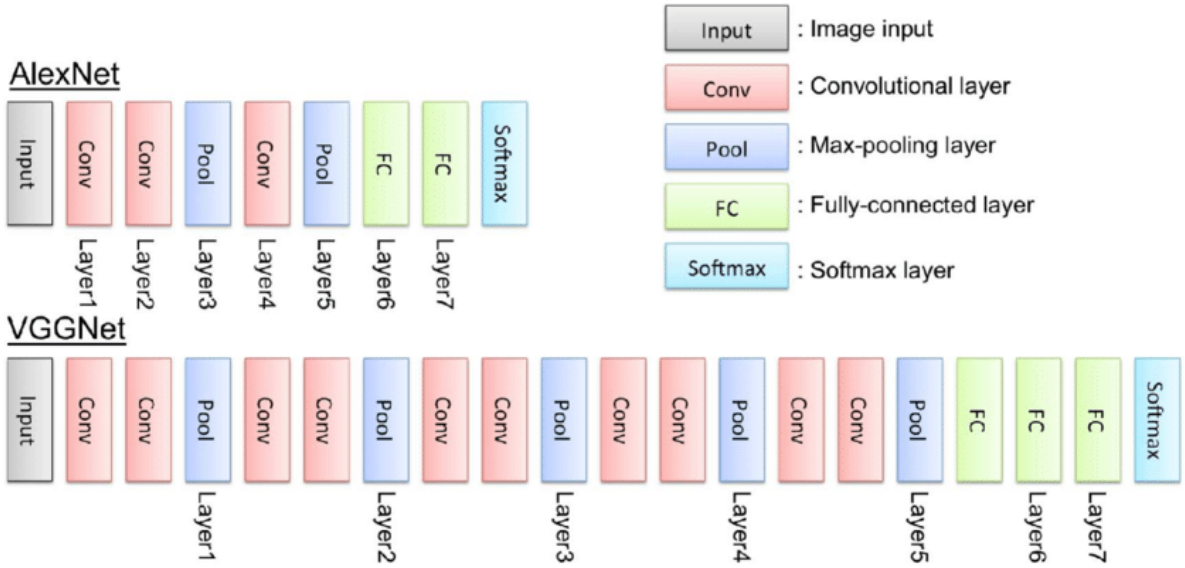
VGG-Net

Introduction

The company Visual Geometry Group created VGGNet (by Oxford University). While GoogLeNet won the classification assignment at ILSVR2014, this architecture came first. Understanding VGGNet is important since many contemporary image classification models are constructed on top of it.

It is a typical deep Convolutional Neural Network (CNN) design with numerous layers, and the abbreviation VGG stands for Visual Geometry Group. The term "deep" describes the number of layers, with VGG-16 or VGG-19 having 16 or 19 convolutional layers, respectively.

Innovative object identification models are built using the VGG architecture. The VGGNet, created as a deep neural network, outperforms benchmarks on a variety of tasks and datasets outside of ImageNet. It also remains one of the most often used image recognition architectures today



Distinguish Between AlexNet (above) and VGGNet (below) architectures

VGG-16

The convolutional neural network model called the VGG model, or VGGNet, that supports 16 layers is also known as VGG16. It was developed by A. Zisserman and K. Simonyan from the University of Oxford. The research paper titled "*Very Deep Convolutional Networks for Large-Scale Image Recognition*" contains the model that these researchers released.

In ImageNet, the VGG16 model achieves top-5 test accuracy of about 92.7 per cent. A dataset called ImageNet has over 14 million photos that fall into almost 1000 types. It was also among the most well-liked models submitted at ILSVRC-2014. It significantly outperforms AlexNet by substituting several 3x3 kernel-sized filters for the huge kernel-sized filters. Nvidia Titan Black GPUs were used to train the VGG16 model over many weeks.

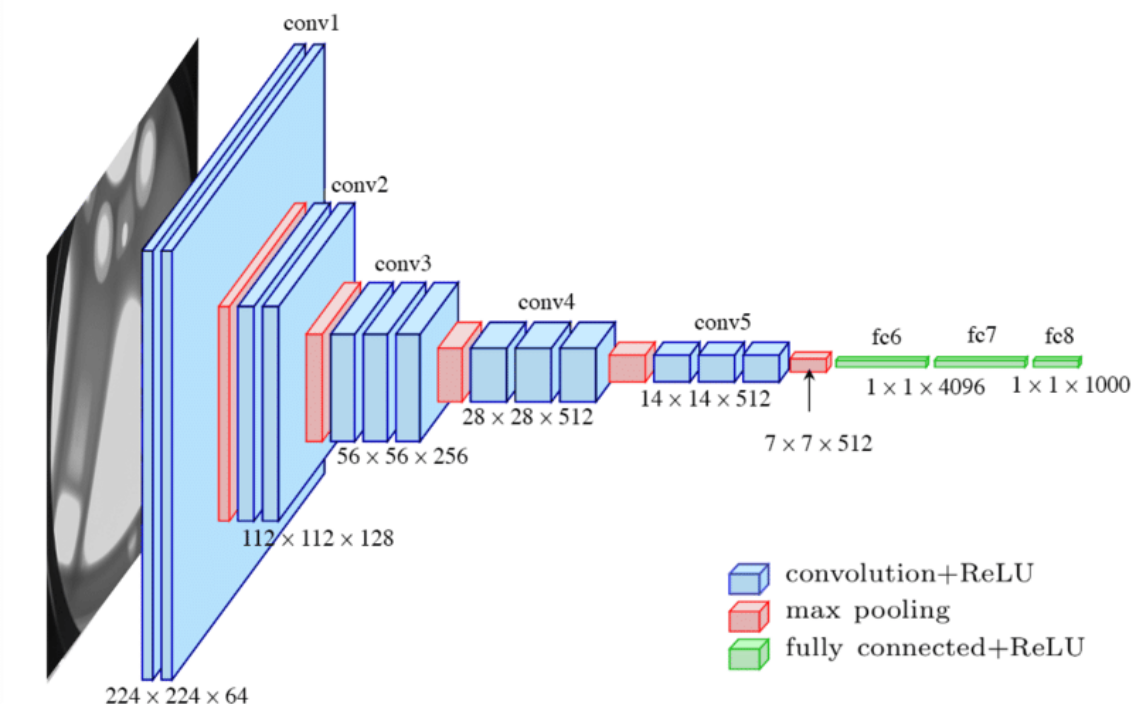
The VGGNet-16 has 16 layers and can classify photos into 1000 different object categories, including keyboard, animals, pencil, mouse, etc., as discussed above. The model also accepts images with a resolution of 224 by 224.7

VGG-19

The VGG19 model (also known as VGGNet-19) has the same basic idea as the VGG16 model, with the exception that it supports 19 layers. The numbers "16" and "19" refer to the model's weight layers (convolutional layers). In comparison to VGG16, VGG19 contains three extra convolutional layers. In the final section of this essay, we'll go into greater detail on the features of the VGG16 and VGG19 networks.

VGG-Net Architecture

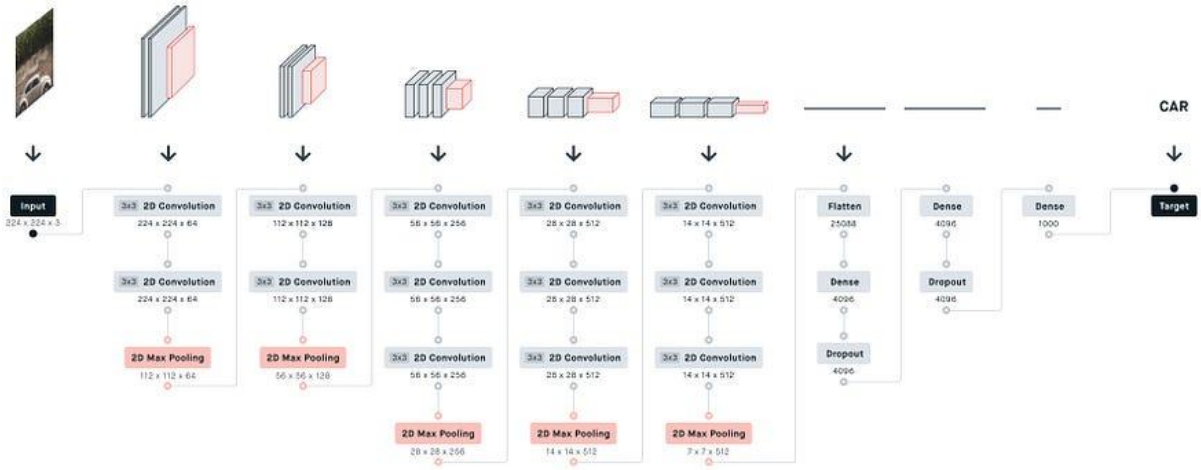
Very tiny convolutional filters are used in the construction of the VGG network. Thirteen convolutional layers and three fully connected layers make up the VGG-16



Understanding VGG-16

The deep neural network's 16 layers are indicated by the number 16 in their name, which is VGG (VGGNet). This indicates that the VGG16 network is quite large, with a total of over 138 million parameters. Even by today's high standards, it is a sizable network. The network is more appealing due to the simplicity of the VGGNet16 architecture, nevertheless. Its architecture alone can be used to describe how uniform it is.

The height and width are decreased by a pooling layer that comes after a few convolution layers. There are around 64 filters available, which we can then multiply by two to get about 128 filters, and so on up to 256 filters. In the last layer, we can use 512 filters.



Applications

- Image Classification: Achieves high accuracy on large datasets.
- Feature Extraction: Used as a backbone for transfer learning in various tasks.
- Object Detection: Forms the base for models like R-CNN.
- Medical Imaging: Applied to X-ray and MRI image classification.

GoogLeNet

Introduction

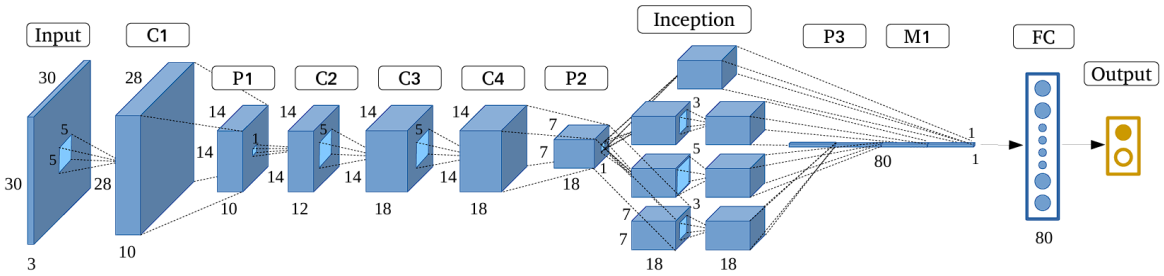
GoogLeNet, released in 2014, set a new benchmark in object classification and detection through its innovative approach (achieving a top-5 error rate of 6.7%, nearly half the error rate of the previous year's winner ZFNet with 11.7%) in ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

GoogLeNet's deep learning model was deeper than all the previous models released, with 22 layers in total. Increasing the depth of the Machine Learning model is intuitive, as deeper models tend to have more learning capacity and as a result, this increases the performance of a model. However, this is only possible if we can solve the vanishing gradient problem

What is special about GoogLeNet?

When designing a deep learning model, one needs to decide what convolution filter size to use (whether it should be 3×3, 5×5, or 1×3) as it affects the model's learning and performance, and when to max pool the layers. However, the inception module, the key innovation introduced by a team of Google researchers solved this problem creatively. Instead of deciding what filter size to use and when to perform a max pooling operation, they combined multiple convolution filters.

Stacking multiple convolution filters together instead of just one increases the parameter count many times. However, GoogLeNet demonstrated by using the inception module that depth and width in a neural network could be increased without exploding computations. We will investigate the inception module in depth



Historical Context of GoogLeNet

The concept of Convolutional Neural Networks (CNNs) isn't new. It dates back to the 1980s with the introduction of the Noncognition by Kunihiko Fukushima. However, CNNs gained popularity in the 1990s after Yann LeCun and his colleagues introduced

LeNet-5 (one of the earliest CNNs), designed for handwritten digit recognition. LeNet-5 laid the groundwork for modern CNNs by using a series of convolutional layers followed by subsampling layers, now commonly referred to as pooling layers.

However, CNNs never saw any widespread adoption for a long time after LeNet-5, due to a lack of computational resources and the unavailability of large datasets, which made the learned models impotent.

The turning point came in 2012 with the introduction of AlexNet by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. AlexNet, for the ImageNet challenge, significantly outperformed other machine learning approaches. This brought deep learning to the forefront of AI research. AlexNet featured several innovations, such as ReLU, dropout for regularization, and overlapping pooling.

After AlexNet, researchers started developing complex and deeper networks. GoogLeNet had 22 layers and VGGNet had 16 layers compared to AlexNet which had only 8 layers in total.

However, in the VGGNet paper, the limitations of simply stacking more layers were highlighted, as it was computationally expensive and led to overfitting. It wasn't possible to keep increasing the layers without any innovation to cater to these problems.

GoogLeNet Variants and Successors

Following the success of the original Google Net (Inception v1), several variants and successors enhanced its architecture. These include Inception v2, v3, v4, and the Inception-ResNet hybrids. Each of these models introduced key improvements and optimizations. These addressed various challenges and pushed the boundaries of what was possible with the CNN architectures.

- Inception v2 (2015): The second version of Inception included improvements such as batch normalization and shortcut connections. It also refined the inception modules by replacing larger convolutions with smaller, more efficient ones. These changes improved accuracy and reduced training time.
- Inception v3 (2015): The v3 model further refined Inception v2 by using atrous convolution (dilated convolutions that expand the network's receptive field without sacrificing resolution and significantly increasing network parameters).

Architecture of GoogLeNet

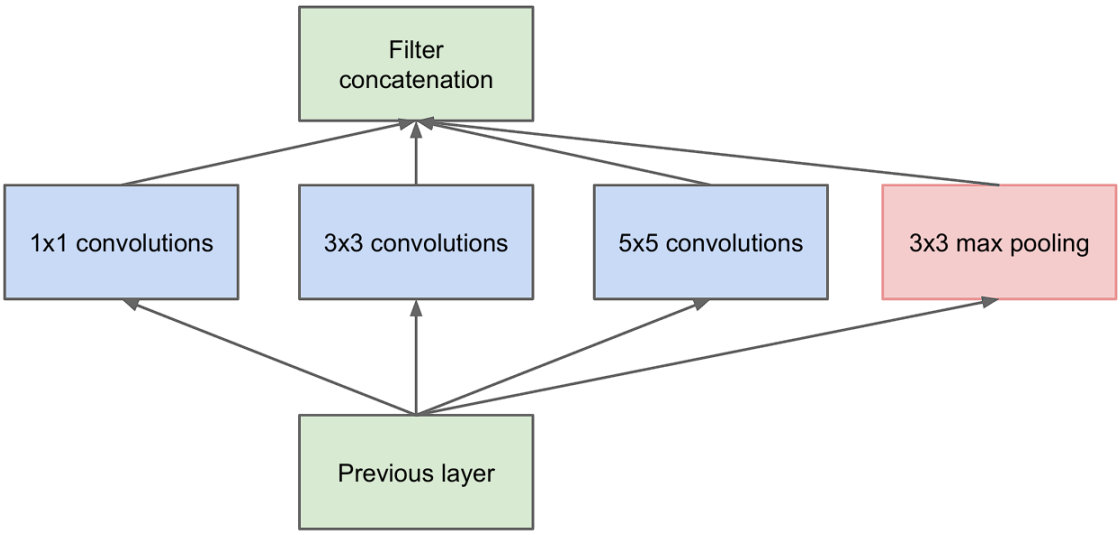
type	patch size/stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1x1x1024	0								
dropout (40%)		1x1x1024	0								
linear		1x1x1000	1							1000K	1M
softmax		1x1x1000	0								

GoogLeNet model is particularly well-known for its use of Inception modules, which serve as its building blocks by using parallel convolutions with various filter sizes (1×1, 3×3, and 5×5) within a single layer. The outputs from these filters are then concatenated. This fusion of outputs from various filters creates a richer representation.

Moreover, the architecture is relatively deep with 22 layers, however, the model maintains computational efficiency despite the increase in the number of layers.

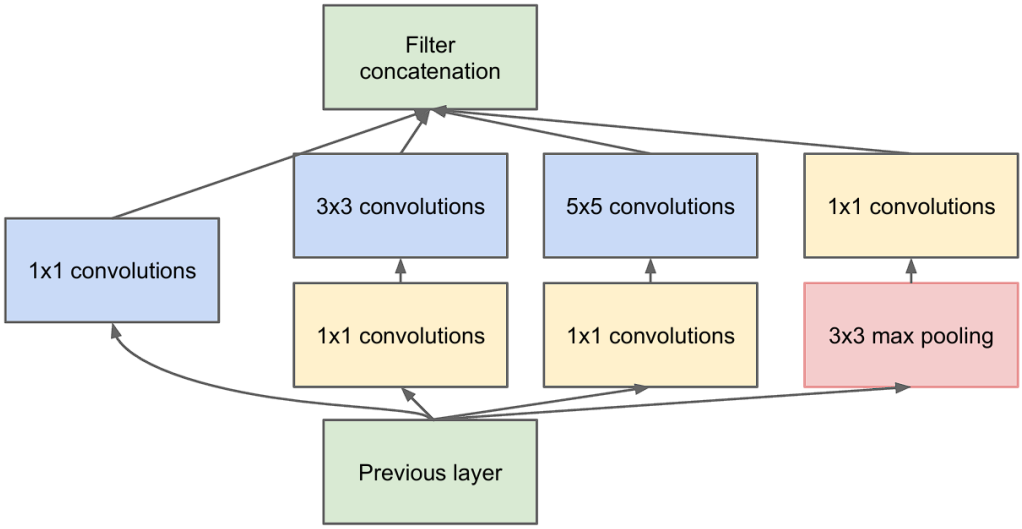
Here are the key features of GoogLeNet:

- Inception Module
- The 1×1 Convolution
- Global Average Pooling
- Auxiliary Classifiers for Training



The Inception Module is the building block of GoogLeNet, as the entire model is made by stacking Inception Modules. Here are the key features of it:

- Multi-Level Feature Extraction: The main idea of the inception module is that it consists of multiple pooling and convolution operations with different sizes (3×3, 5×5) in parallel, instead of using just one filter of a single size.
- Dimension Reduction: However, as we discussed earlier, stacking multiple layers of convolution results in increased computations. To overcome this, the researchers incorporate 1×1 convolution before feeding the data into 3×3 or 5×5 convolutions. We can also refer to this as dimensionality reduction.



To put it into perspective, let's look at the difference.

- Input Feature Map Size: 28×28
- Input Channels (D): 192
- Number of Filters in 3×3 Convolution (F): 96

Without Reduction:

- Total Parameters=3×3×192×96=165,888

With Reduction:

- 1×1 Parameters=1×1×192×64=12,288
- 3×3 Parameters=3×3×64×96=55,296
- Total Parameters with Reduction=12,288+55,296=67,584

Benefits

- Parameter Efficiency: By using 1×1 convolutions, the module reduces dimensionality before applying the more expensive 3×3 and 5×5 convolutions and pooling operations.
- Increased Representation: By incorporating filters of varying sizes and more layers, the network captures a wide range of features in the input data. This results in better representation.

- Enhancing Feature Combination: The 1×1 convolution is also called network in the network. This means that each layer is a micro-neural network that learns to abstract the data before the main convolution filters are applied

Here is a comparison of GoogleNet, VGGNet, and AlexNet

Feature	AlexNet (2012)	VGGNet (2014)	GoogleNet (Inception V1) (2014)
Depth	8 layers (5 Conv + 3 FC)	16 & 19 layers (VGG-16/VGG-19)	22 layers
Architecture	Sequential, large kernel filters (11x11, 5x5)	Deep, uniform 3x3 Conv layers	Inception modules (1x1, 3x3, 5x5 parallel convolutions)
Parameters	60 million	138 million	~5 million
Model Size	Large	Very large	Comparatively small
Accuracy	57% Top-1, 80% Top-5 (ImageNet)	71% Top-1, 89% Top-5 (ImageNet)	74.8% Top-1, 93.3% Top-5 (ImageNet)
Advantages	First deep CNN to win ImageNet, pioneering	Simple, uniform architecture, better than AlexNet	Efficient, fewer parameters, better accuracy
Disadvantages	Large number of parameters, overfitting risk	Heavy computation, large memory requirement	Complex architecture

Summary

- AlexNet introduced deep learning in image classification but had a large number of parameters.
- VGGNet improved upon AlexNet with smaller 3x3 convolutions but required more memory.
- GoogleNet (Inception V1) used Inception modules to reduce computational cost while increasing accuracy.