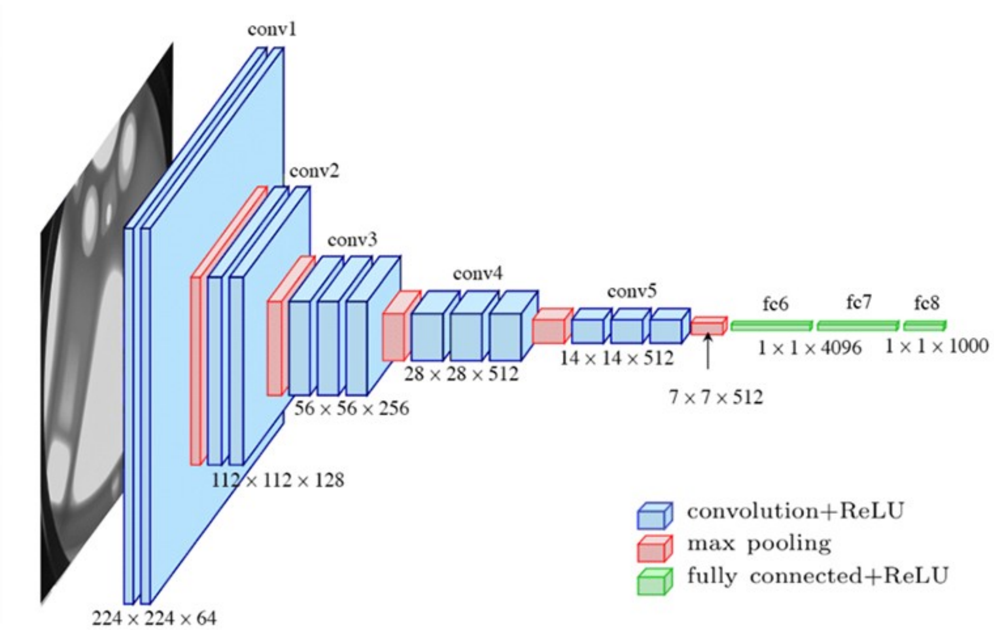
Published in Analytics Vidhya

Draden Liang Han Sheng Follo
Jul 11, 2021 · 4 min read · Listen

VGGNet — Convolutional Network for Classification and Detection



The architecture of VGG16 — uploaded by Max Ferguson in ResearchGate

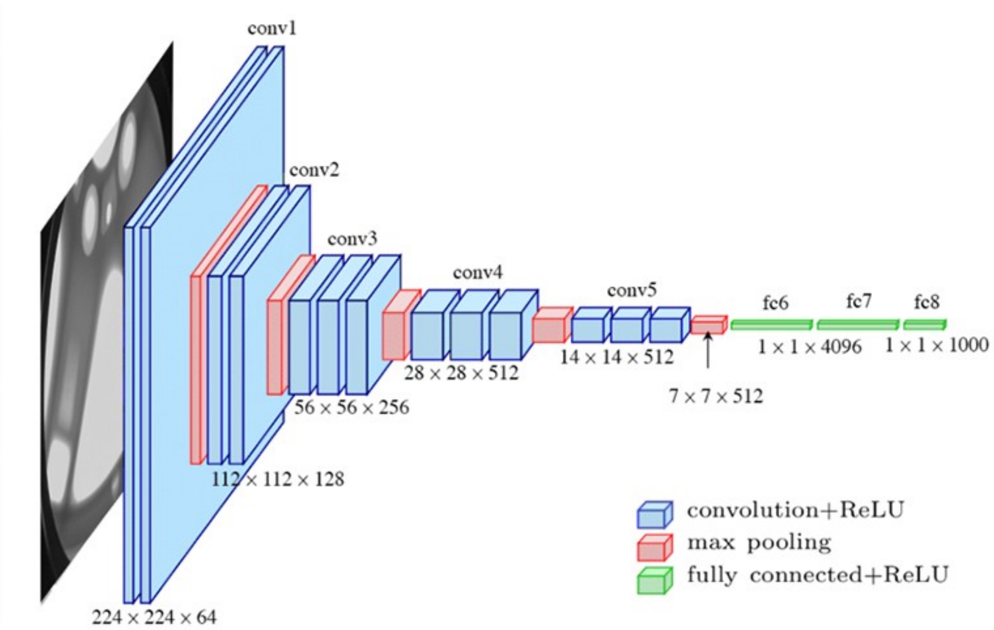
Introduction

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It improves AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional

ImageNet is a dataset of over 15 millic roughly 22,000 categories. The image human labellers using Amazon's Mech 2010, as part of the Pascal Visual Obje ImageNet Large-Scale Visual Recognit uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. There are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images. ImageNet consists of variable-resolution images. Therefore, the images have been down-sampled to a fixed resolution of 256×256. Given a rectangular image, the image is rescaled and cropped out of the central 256×256 patch from the resulting image.

Architecture of VGG16

The architecture depicted below is VGG16.



VGG16 Architecture

The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional layers, where the filters were used with a tiny receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, centre). In one of the configurations, it also utilizes 1×1 convolution filters, max-pooling layers, which follow som layers are followed by max-pooling). I window, with stride 2.

Three Fully-Connected (FC) layers fol different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN). Such normalization does not improve the performance of the ILSVRC dataset but leads to increased memory consumption and computation time.

Configurations

The ConvNet configurations are outlined in figure 2. The nets are referred to as their names (A-E). All configurations follow the generic design present in architecture and differ only in-depth: from 11 weight layers in network A (8 convolution and 3 FC layers) to 19 weight layers in network E (16 convolution and 3 FC layers). The width of convolution layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer until it reaches 512.

Conv					
A	A-LRN	B			
11 weight layers	11 weight layers	13 we… laye…			
input (22…					
conv3-64	conv3-64 **LRN**	conv… **conv…**			
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

ConvNet Configuration for VGGNet

Use-Cases and Implementation

There are two major drawbacks with VGGNet:

1. It is *painfully slow* to train.
2. The network architecture weights themselves are quite large (concerning disk/bandwidth).

Due to its depth and number of fully connected nodes, VGG16 is over 533MB. This makes deploying VGG a tiresome task.VGG16 is used in many deep learning image classification problems; however, smaller network architectures are often more desirable (such as SqueezeNet, GoogLeNet, etc.). But it is a great building block for

Sign in to Medium with Google

Krishnakanta Maity
iamkkmcmd@gmail.com

Continue as Krishnakanta

VGG16 significantly outperforms the
2012 and ILSVRC-2013 competitions.
classification task winner (GoogLeNet
the ILSVRC-2013 winning submission
training data and 11.7% without it. C
architecture achieves the best result (7.0% test error), outperforming a single GoogLeNet by 0.9%.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

Model Performance Comparison

Reference

- https://neurohive.io/en/popular-networks/vgg16/

About Author:

This article is written by Han Sheng, Junior Artificial Intelligence Engineer in CertifAI, Penang, Malaysia. He has a passion for Deep Learning, Computer Vision and also Edge Devices. He made several AI-based Web/Mobile Applications to help clients solving real-world problems. Feel free to read about him via his portfolio or Github profile.

Sign in to Medium with Google

Krishnakanta Maity
iamkkmcmd@gmail.com

Continue as Krishnakanta

Sign up for Analytics Vidhya N

By Analytics Vidhya

Latest news from Analytics Vidhya on our H

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.