卷积神经网络框架

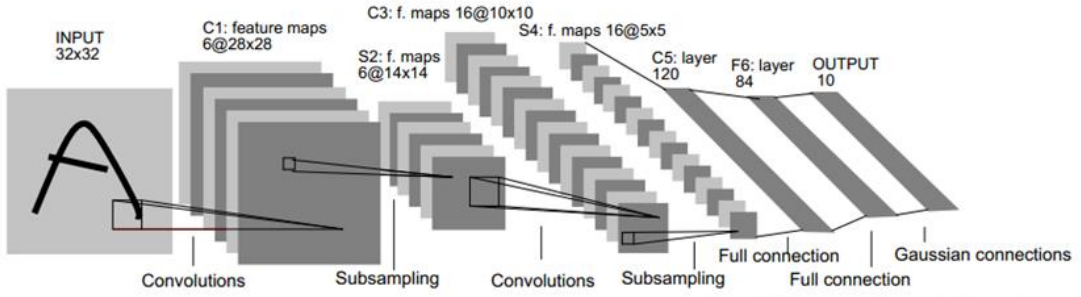
饶晓龙

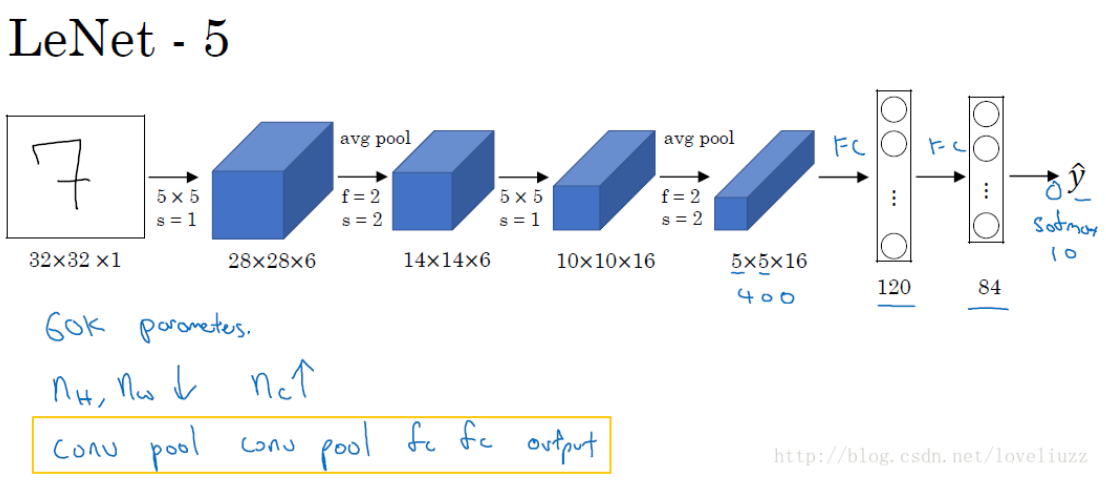
LeNet

背景

LeNet5 诞生于 1994 年，是最早的卷积神经网络之一，并且推动了深度学习领域的发展。自从 1988 年开始，在多年的研究和许多次成功的迭代后，这项由 Yann LeCun 完成的开拓性成果被命名为 LeNet5。

构成





输入尺寸：32*32

卷积层：2 个

降采样层(池化层)：2 个

全连接层：2 个

输出层：1 个。10 个类别（数字 0-9 的概率）

LeNet-5 网络是针对灰度图进行训练的，输入图像大小为 32*32*1，不包含输入层的情况下共有 7 层，每层都包含可训练参数（连接权重）。注：每个层有多个 Feature Map，每个 Feature Map 通过一种卷积滤波器提取输入的一种特征，然后每个 Feature Map 有多个神经元。

1、C1 层是一个卷积层（通过卷积运算，可以使原信号特征增强，并且降低噪音）

第一层使用 5*5 大小的过滤器 6 个，步长 s = 1，padding = 0。即：由 6 个特征图 Feature Map 构成，特征图中每个神经元与输入中 5*5 的邻域相连，输出得到的特征图大小为 28*28*6。C1 有 156 个可训练参数（每个滤波器 5*5=25 个 unit 参数和一个 bias 参数，一共 6 个滤波器，共(5*5+1)*6=156 个参数），共 156*(28*28)=122,304 个连接。

2、S2 层是一个下采样层（平均池化层）(利用图像局部相关性的原理，对图像进行子抽样，可以 1.减少数据处理量同时保留有用信息，2.降低网络训练参数及模型的过拟合程度)。

第二层使用 2*2 大小的过滤器，步长 s = 2，padding = 0。即：特征图中的每个单元与 C1 中相对应特征图的 2*2 邻域相连接，有 6 个 14*14 的特征图，输出得到的特征图大小为 14*14*6。池化层只有一组超参数 f 和 s，没有需要学习的参数。

3、C3 层是一个卷积层

第三层使用 5*5 大小的过滤器 16 个，步长 s = 1，padding = 0。即：由 16 个特征图 Feature Map 构成，特征图中每个神经元与输入中 5*5 的邻域相连，输出得到的特征图大小为 10*10*16。C3 有 416 个可训练参数（每个滤波器 5*5=25 个 unit 参数和一个 bias 参数，一共 16 个滤波器，共(5*5+1)*16=416 个参数)。

4、S4 层是一个下采样层（平均池化层）

第四层使用 2*2 大小的过滤器，步长 s = 2，padding = 0。即：特征图中的每个单元与 C3 中相对应特征图的 2*2 邻域相连接，有 16 个 5*5 的特征图，输出得到的特征图大小为 5*5*16。没有需要学习的参数。

5、F5 层是一个全连接层

有 120 个单元。每个单元与 S4 层的全部 400 个单元之间进行全连接。F5 层有 120*（400+1）=48120 个可训练参数。如同经典神经网络，F5 层计算输入向量和权重向量之间的点积，再加上一个偏置。

6、F6 层是一个全连接层

有 84 个单元。每个单元与 F5 层的全部 120 个单元之间进行全连接。F6 层有 84*（120+1）=10164 个可训练参数。如同经典神经网络，F6 层计算输入向量和权重向量之间的点积，再加上一个偏置。

7、Output 输出层

输出层由欧式径向基函数（Euclidean Radial Basis Function）单元组成，每类一个单元，每个有 84 个输入。 换句话说，每个输出 RBF 单元计算输入向量和参数向量之间的欧

式距离。输入离参数向量越远，RBF 输出的越大。用概率术语来说，RBF 输出可以被理解为 F6 层配置空间的高斯分布的负 log-likelihood。给定一个输式，损失函数应能使得 F6 的配置与 RBF 参数向量（即模式的期望分类）足够接近。

应用

计算机视觉，图像数字识别。

实验

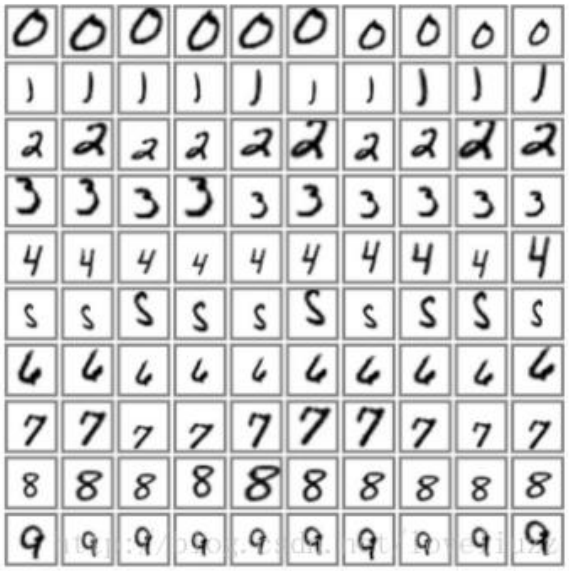
尝试过，目前还在改代码。

层 (layer)	激活后的维度 (Activation Shape)	激活后的大小 (Activation Size)	参数w、b (parameters)
Input	(32,32,1)	1024	0
CONV1 (f=5,s=1)	(28,28,6)	4704	(5*5+1)*6=156
POOL1	(14,14,6)	1176	0
CONV2 (f=5,s=1)	(10,10,16)	1600	(5*5*6+1)*16=2416
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	120*(400+1)=48120
FC4	(84,1)	84	84*(120+1)=10164
Softmax	(10,1)	10	10*(84+1)=850

LeNet-5 on MNIST

60,000 original datasets

Test error: 0.95%



540,000 artificial distortions

+ 60,000 original

Test error: 0.8%

优缺点

LeNet-5的特点:

> (1)每个卷积层包含三个部分：卷积、池化和非线性激活函数
> (2)使用卷积提取空间特征
> (3)降采样（Subsample）的平均池化层（Average Pooling）
> (4)双曲正切（Tanh）或S型（Sigmoid）的激活函数
> MLP作为最后的分类器
> (5)层与层之间的稀疏连接减少计算复杂度

AlexNet

背景

AlexNet 由 Alex Krizhevsky 于 2012 年提出，夺得 2012 年 ILSVRC 比赛的冠军，top5 预测的错误率为 16.4%，远超第一名。AlexNet 采用 8 层的神经网络，5 个卷积层和 3 个全连接层(3 个卷积层后面加了最大池化层)，包含 6 亿 3000 万个链接，6000 万个 参数和 65 万个神经元。

构成

网络结构

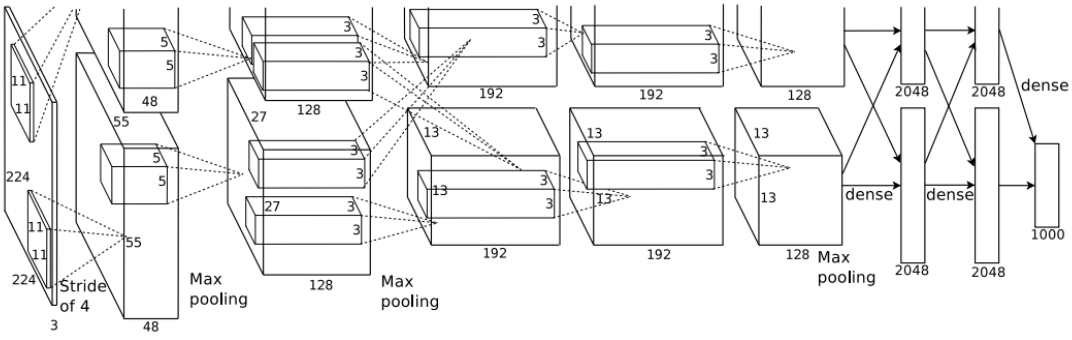


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

网络包含 8 个带权重的层；前 5 层是卷积层，剩下的 3 层是全连接层。最后一层全连接层的输出是 1000 维 softmax 的输入，softmax 会产生 1000 类标签的分布网络包含 8 个带权重的层；前 5 层是卷积层，剩下的 3 层是全连接层。最后一层全连接层的输出是 1000 维 softmax 的输入，softmax 会产生 1000 类标签的分布。

- 卷积层C1
  该层的处理流程是： 卷积-->ReLU-->池化-->归一化。
  - 卷积，输入是227×227，使用96个$11 \times 11 \times 3$的卷积核，得到的FeatureMap为$55 \times 55 \times 96$。
  - ReLU，将卷积层输出的FeatureMap输入到ReLU函数中。
  - 池化，使用3×3步长为2的池化单元（重叠池化，步长小于池化单元的宽度），输出为$27 \times 27 \times 96$ $((55-3)/2+1=27)$
  - 局部响应归一化，使用$k=2, n=5, \alpha=10^{-4}, \beta=0.75$进行局部归一化，输出的仍然为$27 \times 27 \times 96$，输出分为两组，每组的大小为$27 \times 27 \times 48$

- 卷积层C2

  该层的处理流程是：卷积-->ReLU-->池化-->归一化

  - 卷积，输入是2组$27 \times 27 \times 48$。使用2组，每组128个尺寸为$5 \times 5 \times 48$的卷积核，并作了边缘填充padding=2，卷积的步长为1. 则输出的FeatureMap为2组，每组的大小为$27 \times 27 \ times 128$. $((27 + 2 * 2 - 5)/1 + 1 = 27)$

  - ReLU，将卷积层输出的FeatureMap输入到ReLU函数中

  - 池化运算的尺寸为3×3，步长为2，池化后图像的尺寸为$(27 - 3)/2 + 1 = 13$，输出为$13 \times 13 \times 256$

  - 局部响应归一化，使用$k = 2, n = 5, \alpha = 10^{-4}, \beta = 0.75$进行局部归一化，输出的仍然为$13 \times 13 \times 256$，输出分为2组，每组的大小为$13 \times 13 \times 128$

- 卷积层C3

  该层的处理流程是： 卷积-->ReLU

  - 卷积，输入是$13 \times 13 \times 256$，使用2组共384尺寸为$3 \times 3 \times 256$的卷积核，做了边缘填充padding=1，卷积的步长为1.则输出的FeatureMap为$13 \times 13 \ times 384$

  - ReLU，将卷积层输出的FeatureMap输入到ReLU函数中

- 卷积层C4

  该层的处理流程是： 卷积-->ReLU

  该层和C3类似。

  - 卷积，输入是$13 \times 13 \times 384$，分为两组，每组为$13 \times 13 \times 192$.使用2组，每组192个尺寸为$3 \times 3 \times 192$的卷积核，做了边缘填充padding=1，卷积的步长为1.则输出的FeatureMap为$13 \times 13 \ times 384$，分为两组，每组为$13 \times 13 \times 192$

  - ReLU，将卷积层输出的FeatureMap输入到ReLU函数中

- 卷积层C5

  该层处理流程为：卷积-->ReLU-->池化

  - 卷积，输入为$13 \times 13 \times 384$，分为两组，每组为$13 \times 13 \times 192$。使用2组，每组为128尺寸为$3 \times 3 \times 192$的卷积核，做了边缘填充padding=1，卷积的步长为1.则输出的FeatureMap为$13 \times 13 \times 256$

  - ReLU，将卷积层输出的FeatureMap输入到ReLU函数中

  - 池化，池化运算的尺寸为3×3，步长为2，池化后图像的尺寸为 $(13 - 3)/2 + 1 = 6$,即池化后的输出为$6 \times 6 \times 256$

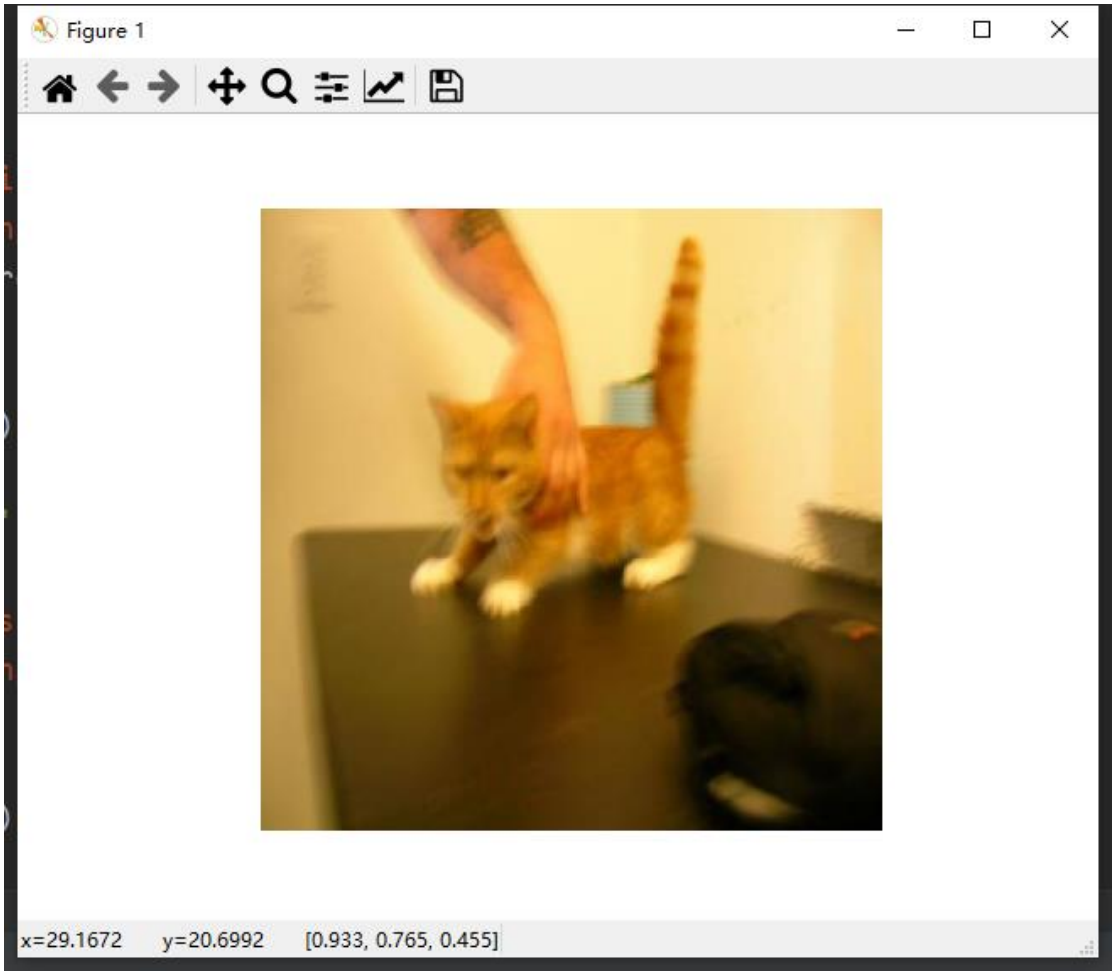
- 全连接层FC6
  该层的流程为：（卷积）全连接 -->ReLU -->Dropout
  - 卷积->全连接： 输入为$6 \times 6 \times 256$,该层有4096个卷积核，每个卷积核的大小为$6 \times 6 \times 256$。由于卷积核的尺寸刚好与待处理特征图（输入）的尺寸相同，即卷积核中的每个系数只与特征图（输入）尺寸的一个像素值相乘，一一对应，因此，该层被称为全连接层。由于卷积核与特征图的尺寸相同，卷积运算后只有一个值，因此，卷积后的像素层尺寸为$4096 \times 1 \times 1$，即有4096个神经元。
  - ReLU,这4096个运算结果通过ReLU激活函数生成4096个值
  - Dropout,抑制过拟合，随机的断开某些神经元的连接或者是不激活某些神经元
- 全连接层FC7
  流程为：全连接-->ReLU-->Dropout
  - 全连接，输入为4096的向量
  - ReLU,这4096个运算结果通过ReLU激活函数生成4096个值
  - Dropout,抑制过拟合，随机的断开某些神经元的连接或者是不激活某些神经元
- 输出层
  第七层输出的4096个数据与第八层的1000个神经元进行全连接，经过训练后输出1000个float型的值，这就是预测结果。

应用

人工智能，神经网络，计算机视觉。

实验

猫狗识别。
使用了网上的代码进行实验。




```
3776/4000 [===========================>..] - ETA: 23s - loss: 0.6984 - accuracy: 0.5326
3840/4000 [===========================>..] - ETA: 16s - loss: 0.6977 - accuracy: 0.5346
3904/4000 [===========================>.] - ETA: 10s - loss: 0.6984 - accuracy: 0.5336
3968/4000 [===========================>.] - ETA: 3s - loss: 0.6985 - accuracy: 0.5338
4000/4000 [==============================] - 451s 113ms/step - loss: 0.6982 - accuracy: 0.5340 - val_loss: 0.6930 - val_accuracy: 0.5000
```

因为是使用 cpu 进行训练的，导致训练时间很长，所以先使用原 github 上的结果。

In [22]:
```
scores = model.evaluate(train_data, train_label, verbose=1)
print(scores)
```

```
5000/5000 [==============================] - 3s 686us/step
[0.20330142648881303, 0.9666]
```

In [23]:
```
scores = model.evaluate(test_data, test_label, verbose=1)
print(scores)
```

```
5000/5000 [==============================] - 3s 686us/step
[1.0541164167642594, 0.8208]
```

优缺点

AlexNet是在LeNet的基础上加深了网络的结构，学习更丰富更高维的图像特征。AlexNet的特点：

- 更深的网络结构
- 使用层叠的卷积层，即卷积层+卷积层+池化层来提取图像的特征
- 使用Dropout抑制过拟合
- 使用数据增强Data Augmentation抑制过拟合
- 使用Relu替换之前的sigmoid的作为激活函数
- 多GPU训练

VGG

背景

VGG 的全称是 Oxford Visual Geometry Group 的简称。该小组隶属于 1985 年成立的 Robotics Research Group，该 Group 研究范围包括了机器学习到移动机器人。该团队斩获 2014 年 ImageNet 挑战赛分类第二（第一是 GoogLeNet），定位任务第一。VGG 可以看成是加深版的 AlexNet，整个网络由卷积层和全连接层叠加而成，和 AlexNet 不同的是，VGG 中使用的都是小尺寸的卷积核(3×33×3)。

构成

VGG网络相比AlexNet层数多了不少，但是其结构却简单不少。

- VGG的输入为$224 \times 224 \times 3$的图像
- 对图像做均值预处理，每个像素中减去在训练集上计算的RGB均值。
- 网络使用连续的小卷积核(3×3)做连续卷积，卷积的固定步长为1，并在图像的边缘填充1个像素，这样卷积后保持图像的分辨率不变。
- 连续的卷积层会接着一个池化层，降低图像的分辨率。空间池化由五个最大池化层进行，这些层在一些卷积层之后（不是所有的卷积层之后都是最大池化）。在2×2像素窗口上进行最大池化，步长为2。
- 卷积层后，接着的是3个全连接层，前两个每个都有4096个通道，第三是输出层输出1000个分类。
- 所有的隐藏层的激活函数都使用的是ReLU
- 使用1×1的卷积核，为了添加非线性激活函数的个数，而且不影响卷积层的感受野。
- 没有使用局部归一化，作者发现局部归一化并不能提高网络的性能。

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
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					

应用

人脸识别等图像分类领域。

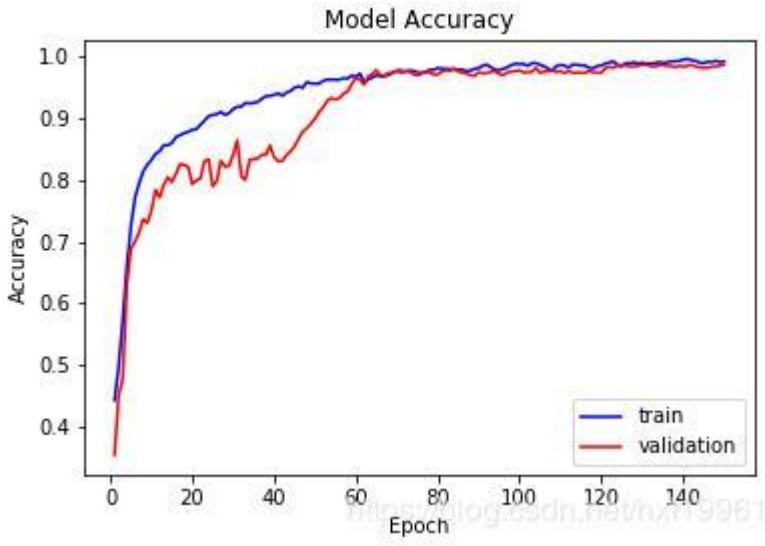
实验

VGG-16 实验。

实验使用的数据是 5 种花的图片，真实图片如下所示：

5 种花简单用 0－4 标签，训练一个不错的网络模型需要大量的数据，本次实验样本数量如下表：

类别	train	test
0	533	100
1	798	100
2	541	100
3	599	100
4	699	100



优缺点

优点：

（1）层数深使得特征图更宽，更加适合于大的数据集，该网络可以解决 1000 类图像分类和定位问题。

（2）卷积核的大小影响到了参数量，感受野，前者关系到训练的难易以及是否方便部署到移动端等，后者关系到参数的更新、特征图的大小、特征是否提取的足够多、模型的复杂程度。

（VGG 用较深的网络结构和较小的卷积核既可以保证感受视野，又能够减少卷积层的参数，比如两个 33 的卷积层叠加等价于一个 55 卷积核的效果，3 个 33 卷积核叠加相加相当于一个 77 的卷积核，而且参数更少。大约是 77 卷积层的（333）/（77）=0.55，三个卷积层的叠加，对特征学习能力更强）

（3）池化层：从 AlexNet 的 kernel size 为 33，stride 为 2 的 max-pooling 改变为 kernel size 均为 22，stride 为 2 的 max-pooling，小的池化核能够带来更细节的信息捕获（当时也有 average pooling，但是在图像任务上 max-pooling 的效果更好，max 更加容易捕捉图像上的变化，带来更大的局部信息差异性，更好的描述边缘纹理等，用 average-

pooling 可能会使得图像模糊了，类似与数字图像处理的高斯模糊）

GoogLeNet

背景

GoogLeNet 是一个基于 Hebbian 法则和多尺度处理构建的网络结构，共有 22 层，相较于当时两年前的 AlexNet 参数少了 12 倍，但精确率提高了很多；不仅能用于分类任务，还可以用于检测任务。

在目标检测领域中，最大的提升并不是来自于使用更大更深的网络，而是将传统 CV 方法和更深的网络结构结合在一起，比如 R-CNN：先利用低层语义信息找到 bbox，再用 CNN 进行分类。

由于移动设备和嵌入式计算的发展，算法的效率即精确率和内存占用变得很重要，GoogLeNet 考虑到了这一点，并权衡了这两者的关系。一个好的模型应当不仅有学术性，而且还能应用于真实世界中。

Inception 结构的想法来源：

之前的研究中有人利用多个不同尺寸的 Gabor 滤波器处理多尺度问题（其实这个想法是来源于灵长类动物的视觉皮质的神经科学模型），但模型层数太少，而 GoogLeNet 将

Inception 重复很多次以此实现 22 层的网络结构 1 x 1 卷积的想法来源于 NiN

构成

Inception 该网络 的特点是提升了计算资源的利用率，可以在保持网络计算资源不变的前提下，通过工艺上的设计来增加网络的宽度和深度，基于 Hebbian 法则和多尺度处理来优化性能。在 ILSVRC2014 中提交的版本叫 GoogLeNet，共有 22 层。

GoogLeNet 用的参数比 ILSVRC2012 的冠军 AlexNet 少 12 倍，但准确率更高。现在的目标检测例如 R-CNN，结合了深度架构和传统计算机视觉方法进行目标检测。

由于移动设备和嵌入式计算的发展，算法的效率很重要，一个好的算法，不仅要具有学术性，也要能用于实际中。

直接提升深度卷积网络性能的方法是从深度与宽度两方面增加尺寸，但大尺寸的网络需要更多参数，容易导致过拟合，尤其是数据集不够大的时候，直接增加 尺寸的另一个弊端是需要大量计算资源。根本的解决办法是将全连接层变为稀疏链接层，而非均匀稀疏网络的弊端是计算效率不高，可以采用将多个稀疏矩阵合并成 相关的稠密子矩阵的方法来解决。

Inception 架构的主要思想是找出如何让已有的稠密组件接近与覆盖卷积视觉网络中的最佳局部稀疏结构。现在需要找出最优的局部构造，并且重复 几次。之前的一篇文献提出一个层与层的结构，在最后一层进行相关性统计，将高相关性的聚集到一起。这些聚类构成下一层的单元，且与上一层单元连接。假设前 面层的每个单元对应于输入图像的某些区域，这些单元被分为滤波器组。在接近输入层的低层中，相关单元集中在某些局部区域，最终得到在单个区域中的大量聚 类，在下一层通过 1x1 的卷积覆盖。

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	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

应用

图像分类，计算机视觉领域

实验

分别训练了 7 个模型，每个模型初始值相同，但模型的采样方法和输入图片的随机序列不同。在测试时，将图片短边 resize 到 4 个固定的 S 值（256，288，320，352），再根据最短边的数值裁剪 resize 后图像的上中下（或左中右）三个正方形区域，然后将正方形图像的 5 个区域（左上、右上、左下、右下、中央）和该正方形图像整体（共 6 张图片）resize 到 224*224 同时再水平翻转一次作为网络输入。这样一张图片能够扩增为 4*3*6*2=144 张测试图片。将这些图片在 softmax 层做平均，获得最后的预测结果。

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Table 3: GoogLeNet classification performance break down

优缺点

1. 保证算力情况下增大宽度和深度。
2. 宽度：利用 Inception 结构同时执行多个网络结构。
3. 深度：利用辅助分类器防止梯度消失多尺度训练和预测。

4. 适用于多种计算机视觉任务。