

经典网络解析

鲁鹏

北京邮电大学 计算机学院 智能科学与技术中心

经典网络解析

- AlexNet
- ZFNet
- VGG
- GoogLeNet
- ResNet

经典网络解析——AlexNet

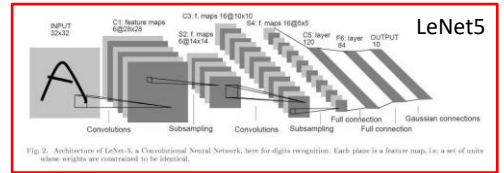
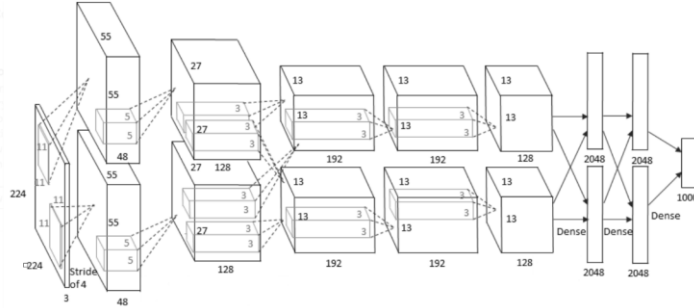
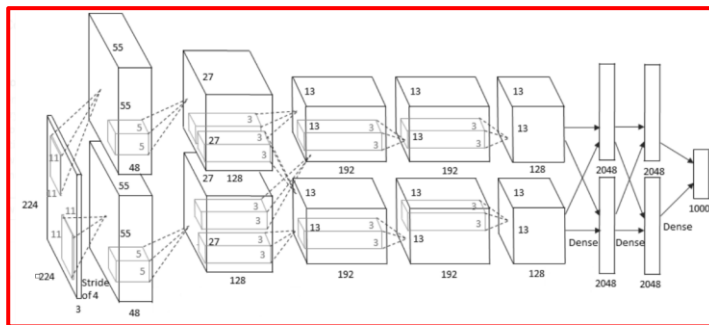


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digit recognition. Each plane is a feature map, i.e. a set of nodes whose weights are constrained to be identical.

经典网络解析——AlexNet



AlexNet——2012年ImageNet大规模视觉识别挑战赛冠军, 精度提升超过10个百分点!

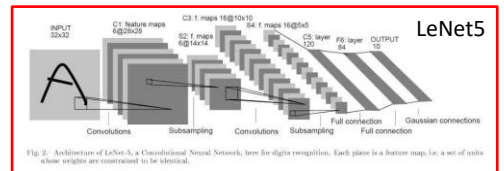


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digit recognition. Each plane is a feature map, i.e. a set of nodes whose weights are constrained to be identical.

ImageNet大规模视觉识别挑战赛

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

2020/4/28

北京邮电大学计算机学院 鲁鹏

4

ImageNet大规模视觉识别挑战赛

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

➤ 计算机视觉领域最具权威的学术竞赛之一

2020/4/28

北京邮电大学计算机学院 鲁鹏

5

ImageNet大规模视觉识别挑战赛

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- 计算机视觉领域最具权威的学术竞赛之一
- ImageNet数据集——由斯坦福大学李飞飞教授主导制作，其包含了超过1400万张全尺寸的有标记图片

2020/4/28

北京邮电大学计算机学院 鲁鹏

6

ImageNet大规模视觉识别挑战赛

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- 计算机视觉领域最具权威的学术竞赛之一
- ImageNet数据集——由斯坦福大学李飞飞教授主导制作，其包含了超过1400万张全尺寸的有标记图片
- ILSVRC从ImageNet数据集中抽出子集作为竞赛数据

2020/4/28

北京邮电大学计算机学院 鲁鹏

7

ImageNet大规模视觉识别挑战赛

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- 计算机视觉领域最具权威的学术竞赛之一
- ImageNet数据集——由斯坦福大学李飞飞教授主导制作，其包含了超过1400万张全尺寸的有标记图片
- ILSVRC从ImageNet数据集中抽出子集作为竞赛数据

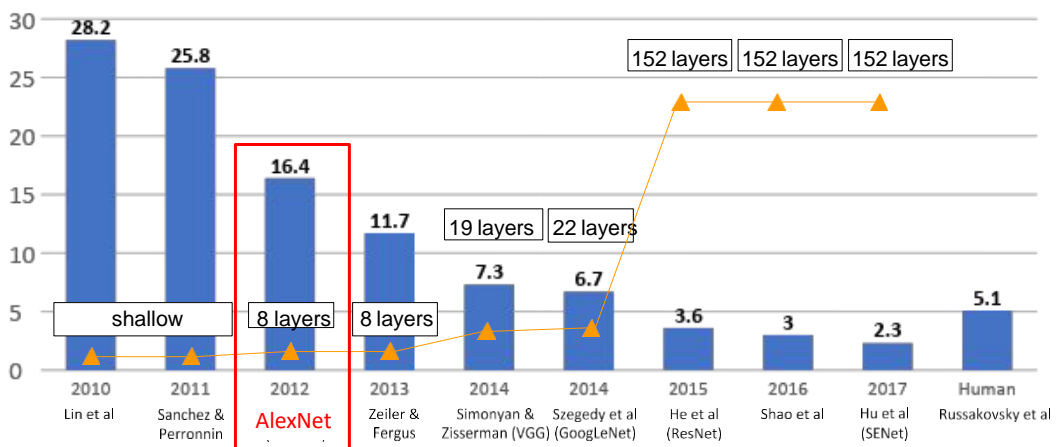
2012年ILSVRC——1281167张训练集，50000张验证集，100000张测试集。

2020/4/28

北京邮电大学计算机学院 鲁鹏

8

ImageNet大赛历年冠军

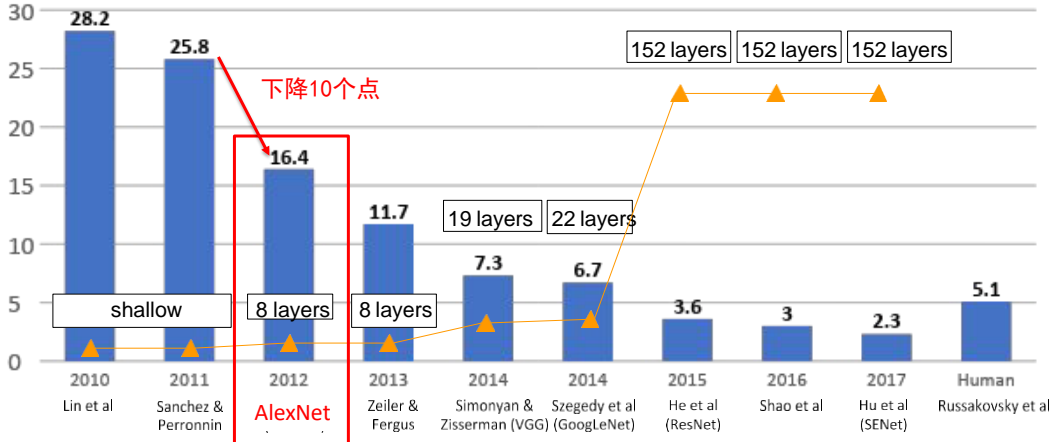


2020/4/28

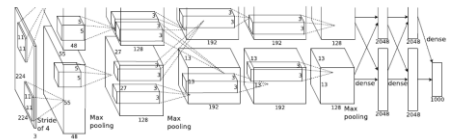
北京邮电大学计算机学院 鲁鹏

9

ImageNet大赛历年冠军



经典网络解析——AlexNet



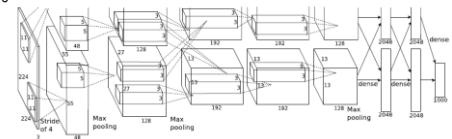
经典网络解析——AlexNet



AlexNet——验证了深度卷积神经网络的高效性

主体贡献

1. 提出了一种卷积层加全连接层的卷积神经网络结构
2. 首次使用ReLU函数做为神经网络的激活函数
3. 首次提出Dropout正则化来控制过拟合
4. 使用加入动量的小批量梯度下降算法加速了训练过程的收敛；
5. 使用数据增强策略极大地抑制了训练过程的过拟合；
6. 利用了GPU的并行计算能力，加速了网络的训练与推断。



2020/4/28

北京邮电大学计算机学院 鲁鹏

12

经典网络解析——AlexNet

结构:

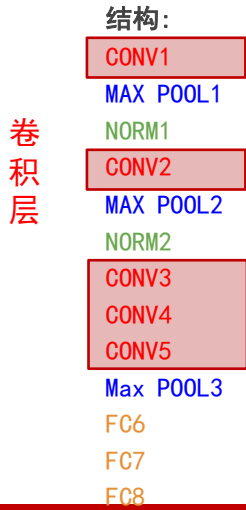
CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

13

经典网络解析——AlexNet

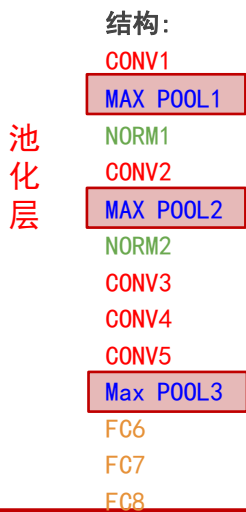


2020/4/28

北京邮电大学计算机学院 鲁鹏

14

经典网络解析——AlexNet



2020/4/28

北京邮电大学计算机学院 鲁鹏

15

经典网络解析——AlexNet

局部响应归一化层

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

16

经典网络解析——AlexNet

全连接层

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

17

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

层数统计说明:

- 计算网络层数时仅统计卷积层与全连接层;

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

层数统计说明:

- 计算网络层数时仅统计卷积层与全连接层;
- 池化层与各种归一化层都是对它们前面卷积层输出的特征图进行后处理，不单独算作一层。

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

层数统计说明:

- 计算网络层数时仅统计卷积层与全连接层;
- 池化层与各种归一化层都是对它们前面卷积层输出的特征图进行后处理, 不单独算作一层。

AlexNet共8层:

- 5个卷积层 (CONV1——CONV5)
- 3个全连接层 (FC6-FC8)

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第一层 (CONV1): 96 个11x11 卷积核, 步长为 4, 没有零填充

2020/4/28

北京邮电大学计算机学院 鲁鹏

22

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第一层 (CONV1): 96 个11x11 卷积核, 步长为 4, 没有零填充

问题: 输入: 227x227x3 大小的图像, 输出特征图个数及尺寸为多少?

尺寸: $(227-11)/4+1 = 55$

个数: 96

2020/4/28

北京邮电大学计算机学院 鲁鹏

23

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第一层 (CONV1): 96 个 11×11 卷积核, 步长为 4, 没有零填充

问题: 输入: $227 \times 227 \times 3$ 大小的图像, 输出特征图个数及尺寸为多少?

尺寸: $(227-11)/4+1 = 55$

个数: 96

问题: 这层有多少个参数?

参数: $(11 \times 11 \times 3 + 1) \times 96 = 35K$

2020/4/28

北京邮电大学计算机学院 鲁鹏

24

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第一层 (CONV1): 96 个 11×11 卷积核, 步长为 4, 没有零填充

问题: 输入: $227 \times 227 \times 3$ 大小的图像, 输出特征图个数及尺寸为多少?

尺寸: $(227-11)/4+1 = 55$

个数: 96

问题: 这层有多少个参数?

参数: $(11 \times 11 \times 3 + 1) \times 96 = 35K$

➤ 第一个卷积层提取了96种结构的响应信息, 得到了96个特征相应图;

2020/4/28

北京邮电大学计算机学院 鲁鹏

25

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第一层 (CONV1): 96 个11x11 卷积核, 步长为 4, 没有零填充

问题: 输入: 227x227x3 大小的图像, 输出特征图个数及尺寸为多少?

尺寸: $(227-11)/4+1 = 55$

个数: 96

问题: 这层有多少个参数?

参数: $(11*11*3+1)*96 = 35K$

- 第一个卷积层提取了96种结构的响应信息, 得到了96个特征相应图;
- 特征图每个元素经过ReLU函数操作后输出。

2020/4/28

北京邮电大学计算机学院 鲁鹏

26

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

27

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

Max POOL1: 窗口大小3x3, 步长为 2

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

28

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

Max POOL1: 窗口大小3x3, 步长为 2

NORM1

作用: 降低特征图尺寸, 对抗轻微的目标偏移带来的影响

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

29

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

重叠有助于
对抗过拟合

Max POOL1: 窗口大小3x3, 步长为 2

作用: 降低特征图尺寸, 对抗轻微的目标偏移带来的影响

2020/4/28

北京邮电大学计算机学院 鲁鹏

30

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

重叠有助于
对抗过拟合

Max POOL1: 窗口大小3x3, 步长为 2

作用: 降低特征图尺寸, 对抗轻微的目标偏移带来的影响

输出尺寸: $(55-3)/2+1 = 27$

特征图个数: 96

2020/4/28

北京邮电大学计算机学院 鲁鹏

31

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

重叠有助于
对抗过拟合

Max POOL1: 窗口大小3x3, 步长为 2

作用: 降低特征图尺寸, 对抗轻微的目标偏移带来的影响

输出尺寸: $(55-3)/2+1 = 27$

特征图个数: 96

参数个数: 0

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

局部相应归一化层 (NORM1)

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

局部相应归一化层 (NORM1) 作用:

- 对局部神经元的活动创建竞争机制;
- 响应比较大的值变得相对更大;
- 抑制其他反馈较小的神经元;
- 增强模型的泛化能力

2020/4/28

北京邮电大学计算机学院 鲁鹏

34

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

局部相应归一化层 (NORM1) 作用:

- 对局部神经元的活动创建竞争机制;
- 响应比较大的值变得相对更大;
- 抑制其他反馈较小的神经元;
- 增强模型的泛化能力

后来的研究表明:
更深的网络中该层
对分类性能的提升
效果并不明显, 且
会增加计算量与存
储空间。

2020/4/28

北京邮电大学计算机学院 鲁鹏

35

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

2020/4/28

北京邮电大学计算机学院 鲁鹏

36

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第二层 (CONV2): 256 个5x5 卷积核, 步长为 1, 使用零填充 $p=2$

2020/4/28

北京邮电大学计算机学院 鲁鹏

37

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第二层 (CONV2): 256 个5x5 卷积核, 步长为 1, 使用零填充p=2

问题: 输入: 27x27x96 大小的特征图组, 输出特征图个数及尺寸为多少?

尺寸: $(27 - 5 + 2*2)/1+1 = 27$

个数: 256

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第二层 (CONV2): 256 个5x5 卷积核, 步长为 1, 使用零填充p=2

问题: 输入: 27x27x256 大小的特征图组, 输出特征图个数及尺寸为多少?

尺寸: $(27 - 5 + 2*2)/1+1 = 27$

个数: 256

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第三、四层 (CONV3、CONV4): 384 个3x3 卷积核, 步长为 1, 使用零填充
p=1

问题: CONV3输入: 13x13x256 大小的特征图组, 输出特征图个数及尺寸为多少?

尺寸: $(13 - 3 + 2*1)/1+1 = 13$

个数: 384

2020/4/28

北京邮电大学计算机学院 鲁鹏

40

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第三、四层 (CONV3、CONV4): 384 个3x3 卷积核, 步长为 1, 使用零填充
p=1

问题: CONV3输入: 13x13x256 大小的特征图组, 输出特征图个数及尺寸为多少?

尺寸: $(13 - 3 + 2*1)/1+1 = 13$

个数: 384

没有进行最大池化与局部归一化

2020/4/28

北京邮电大学计算机学院 鲁鹏

41

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第五层 (CONV5): 256 个3x3 卷积核, 步长为 1, 使用零填充 $p=1$

2020/4/28

北京邮电大学计算机学院 鲁鹏

42

经典网络解析——AlexNet

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

第五层 (CONV5): 256 个3x3 卷积核, 步长为 1, 使用零填充 $p=1$

最大池化层来进一步缩小特征图尺寸

2020/4/28

北京邮电大学计算机学院 鲁鹏

43

经典网络解析——AlexNet

结构:

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- MAX POOL3

全连接层

- FC6
- FC7
- FC8

第六——八层 (FC6、FC7 、FC8):全连接神经网络分类器

经典网络解析——AlexNet

结构:

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- MAX POOL3

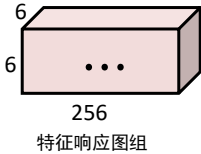
全连接层

- FC6
- FC7
- FC8

第六——八层 (FC6、FC7 、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

输出: 图像类别概率



经典网络解析——AlexNet

结构:

全
连
接
层

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3

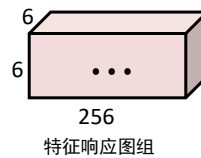
FC6
FC7
FC8

第六——八层 (FC6、FC7、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

输出: 图像类别概率

FC6期望输入: 向量



2020/4/28

北京邮电大学计算机学院 鲁鹏

46

经典网络解析——AlexNet

结构:

全
连
接
层

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3

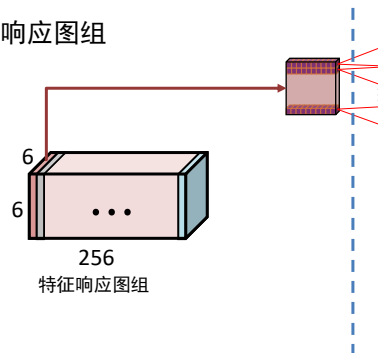
FC6
FC7
FC8

第六——八层 (FC6、FC7、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

FC6期望输入: 向量

输出: 图像类别概率



2020/4/28

北京邮电大学计算机学院 鲁鹏

47

经典网络解析——AlexNet

全
连
接
层

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

MAX POOL3

FC6

FC7

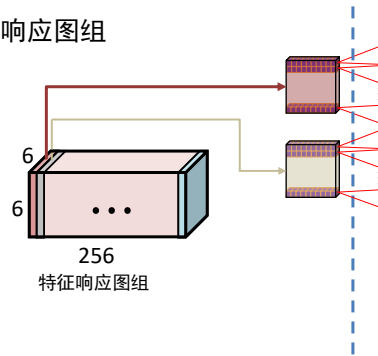
FC8

第六——八层 (FC6、FC7、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

FC6期望输入: 向量

输出: 图像类别概率



2020/4/28

北京邮电大学计算机学院 鲁鹏

48

经典网络解析——AlexNet

全
连
接
层

结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

MAX POOL3

FC6

FC7

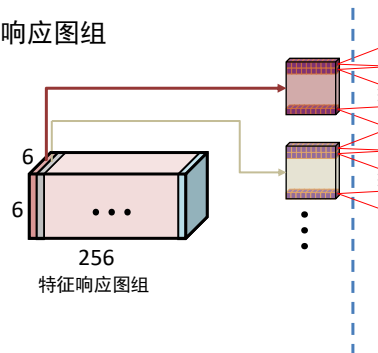
FC8

第六——八层 (FC6、FC7、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

FC6期望输入: 向量

输出: 图像类别概率



2020/4/28

北京邮电大学计算机学院 鲁鹏

49

经典网络解析——AlexNet

全
连
接
层

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3

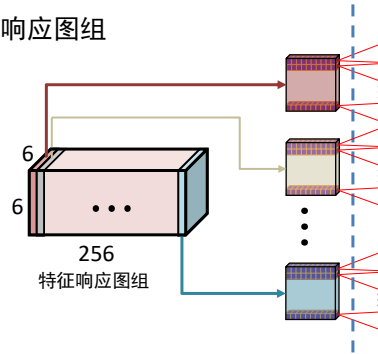
FC6
FC7
FC8

第六——八层 (FC6、FC7、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

FC6期望输入: 向量

输出: 图像类别概率



2020/4/28

北京邮电大学计算机学院 鲁鹏

50

经典网络解析——AlexNet

全
连
接
层

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3

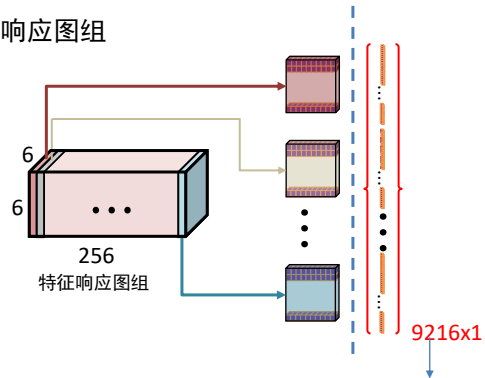
FC6
FC7
FC8

第六——八层 (FC6、FC7、FC8):全连接神经网络分类器

MAX POOL3的输出: 特征响应图组

FC6输入: 9216维向量

输出: 图像类别概率



2020/4/28

北京邮电大学计算机学院 鲁鹏

9216=6x6x256 51

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要说明:

2020/4/28

北京邮电大学计算机学院 鲁鹏

52

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要说明:

- 用于提取图像特征的卷积层以及用于分类的全连接层是同时学习的;

2020/4/28

北京邮电大学计算机学院 鲁鹏

53

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要说明:

- 用于提取图像特征的卷积层以及用于分类的全连接层是同时学习的;
- 卷积层与全连接层在学习过程中会相互影响、相互促进

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要技巧:

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要技巧:

- Dropout策略防止过拟合;

2020/4/28

北京邮电大学计算机学院 鲁鹏

56

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要技巧:

- Dropout策略防止过拟合;
- 使用加入动量的随机梯度下降算法, 加速收敛;

2020/4/28

北京邮电大学计算机学院 鲁鹏

57

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要技巧:

- Dropout策略防止过拟合;
- 使用加入动量的随机梯度下降算法, 加速收敛;
- 验证集损失不下降时, 手动降低10倍的学习率;

2020/4/28

北京邮电大学计算机学院 鲁鹏

58

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要技巧:

- Dropout策略防止过拟合;
- 使用加入动量的随机梯度下降算法, 加速收敛;
- 验证集损失不下降时, 手动降低10倍的学习率;
- 采用样本增强策略增加训练样本数量, 防止过拟合;

2020/4/28

北京邮电大学计算机学院 鲁鹏

59

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8

重要技巧:

- Dropout策略防止过拟合;
- 使用加入动量的随机梯度下降算法, 加速收敛;
- 验证集损失不下降时, 手动降低10倍的学习率;
- 采用样本增强策略增加训练样本数量, 防止过拟合;
- 集成多个模型, 进一步提高精度。

2020/4/28

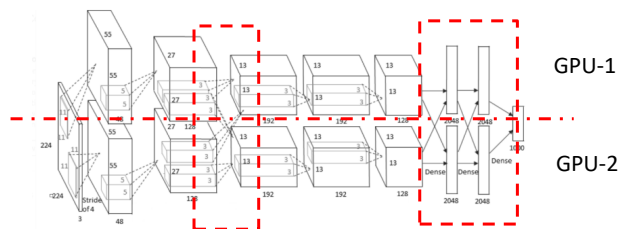
北京邮电大学计算机学院 鲁鹏

60

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8



- AlexNet分布于两个GPU, 每个GPU各有一半神经元;
- CONV1、CONV2、CONV4、CONV5仅使用在同一GPU的前层网络输出的特征图作为输入;
- CONV3、FC6、FC7、FC8拼接前一层所有的特征图, 实现了神经网络的跨GPU计算。

2020/4/28

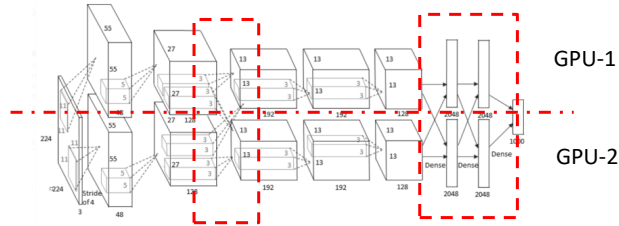
北京邮电大学计算机学院 鲁鹏

61

经典网络解析——AlexNet

结构:

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3
FC6
FC7
FC8



- AlexNet分布于两个GPU，每个GPU各有一半神经元；
- CONV1、CONV2、CONV4、CONV5仅使用在同一GPU的前层网络输出的特征图作为输入；
- CONV3、FC6、FC7、FC8拼接前一层所有的特征图，实现了神经网络的跨GPU计算。

GTX 580 GPU仅有3GB显存!!!

2020/4/28

北京邮电大学计算机学院 鲁鹏

62

问题：AlexNet卷积层在做什么？

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
Max POOL3

FC6

FC7

FC8

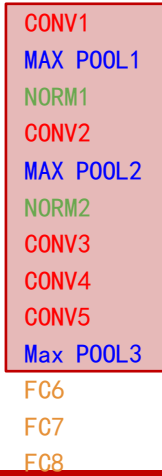
2020/4/28

北京邮电大学计算机学院 鲁鹏

63

问题：AlexNet卷积层在做什么？

输入：
227x227x3



输出：
6x6x256

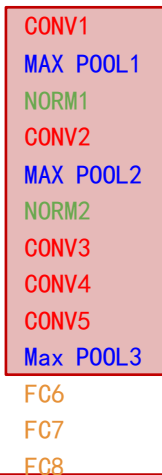
2020/4/28

北京邮电大学计算机学院 鲁鹏

64

问题：AlexNet卷积层在做什么？

输入：
227x227x3



输出：
6x6x256

2020/4/28

北京邮电大学计算机学院 鲁鹏

65

问题：AlexNet卷积层在做什么？

输入：
227x227x3



- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



超级卷
积核组

输出：
6x6x256

问题：AlexNet卷积层在做什么？



- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



超级卷
积核组

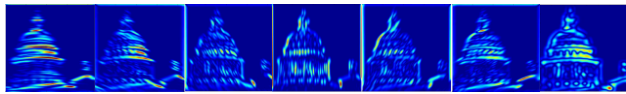
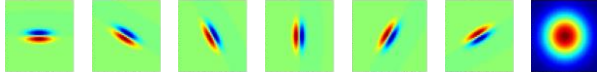
输出：
6x6x256

问题：AlexNet卷积层在做什么？

输入：
227x227x3



- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



输出：
6x6x256

问题：AlexNet卷积层在做什么？

具有“语义结构”

输入：
227x227x3



- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



卷积核

超级卷
积核组

输出：
6x6x256

问题：AlexNet卷积层在做什么？

- 具有“语义结构”
- 从数据中习得

输入：
227x227x3

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



卷积核



超级卷积核组

输出：
6x6x256

问题：AlexNet卷积层在做什么？

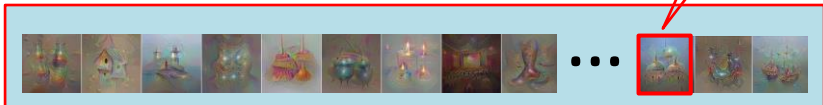
- 具有“语义结构”
- 从数据中习得

输入：
227x227x3

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



卷积核



超级卷积核组



特征响应图

输出：
6x6x256

问题：AlexNet卷积层在做什么？

- 具有“语义结构”
- 从数据中习得

输入：
227x227x3

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



卷积核

超级卷积核组



特征响应图

响应值高

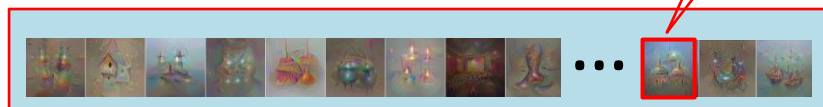
输出：
6x6x256

问题：AlexNet卷积层在做什么？

- 具有“语义结构”
- 从数据中习得

输入：
227x227x3

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



卷积核

超级卷积核组



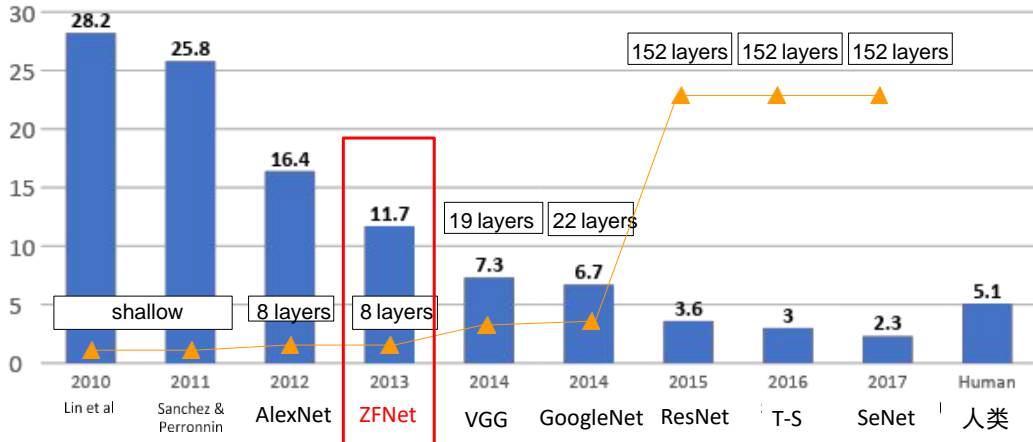
特征响应图

响应值高

输出：
6x6x256

- 从数据中学习对于分类有意义的结构特征，
- 描述输入图像中的结构信息
- 描述结果存储在256个6x6的特征响应图里。

ImageNet大赛历年冠军



2020/4/28

北京邮电大学计算机学院 鲁鹏

74

经典网络解析—— ZFNet

ZFNet结构:

CONV1
 MAX POOL1
 NORM1
 CONV2
 MAX POOL2
 NORM2
 CONV3
 CONV4
 CONV5
 Max POOL3
 FC6
 FC7
 FC8

与AlexNet网络结构基本一致!

2020/4/28

北京邮电大学计算机学院 鲁鹏

75

经典网络解析—— ZFNet

ZFNet结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

与AlexNet网络结构基本一致!

主要改进:

- 将第一个卷积层的卷积核大小改为了 7×7 ;

2020/4/28

北京邮电大学计算机学院 鲁鹏

76

经典网络解析—— ZFNet

ZFNet结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

与AlexNet网络结构基本一致!

主要改进:

- 将第一个卷积层的卷积核大小改为了 7×7 ;
- 将第二、第三个卷积层的卷积步长都设置为2;

2020/4/28

北京邮电大学计算机学院 鲁鹏

77

经典网络解析—— ZFNet

ZFNet结构:

CONV1

MAX POOL1

NORM1

CONV2

MAX POOL2

NORM2

CONV3

CONV4

CONV5

Max POOL3

FC6

FC7

FC8

与AlexNet网络结构基本一致!

主要改进:

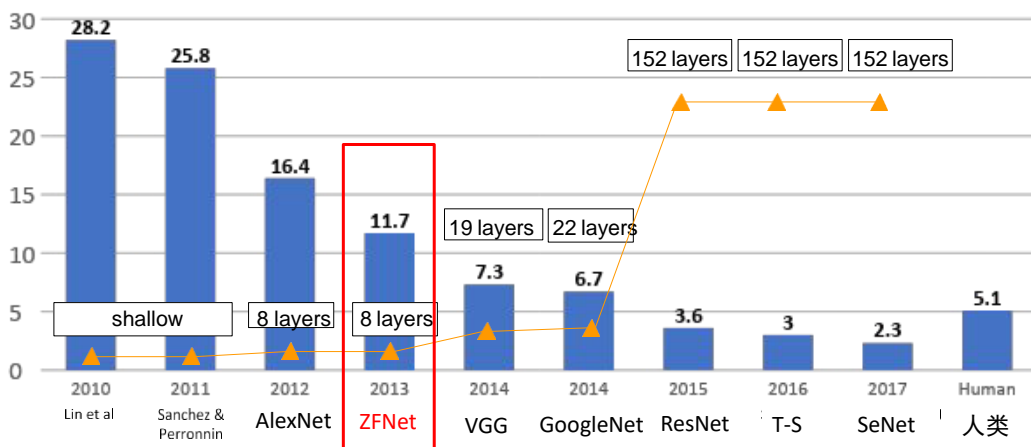
- 将第一个卷积层的卷积核大小改为了 7×7 ;
- 将第二、第三个卷积层的卷积步长都设置为2;
- 增加了第三、第四个卷积层的卷积核个数。

2020/4/28

北京邮电大学计算机学院 鲁鹏

78

ImageNet大赛历年冠军

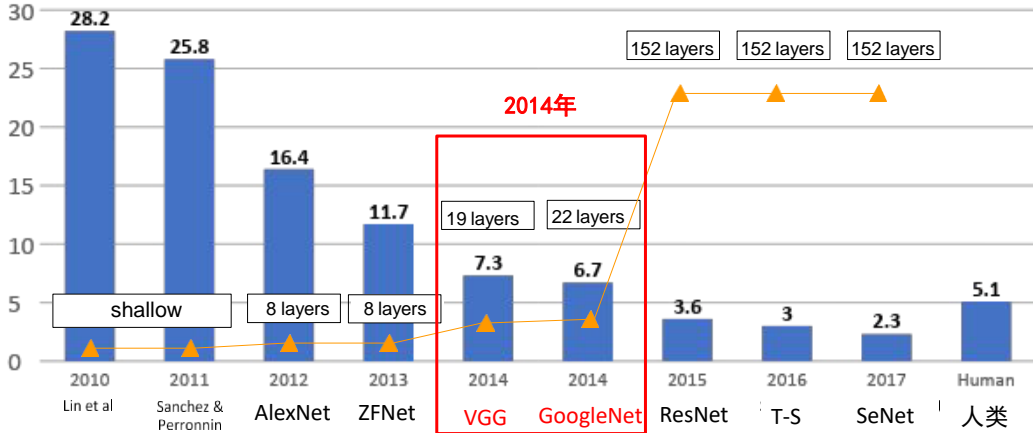


2020/4/28

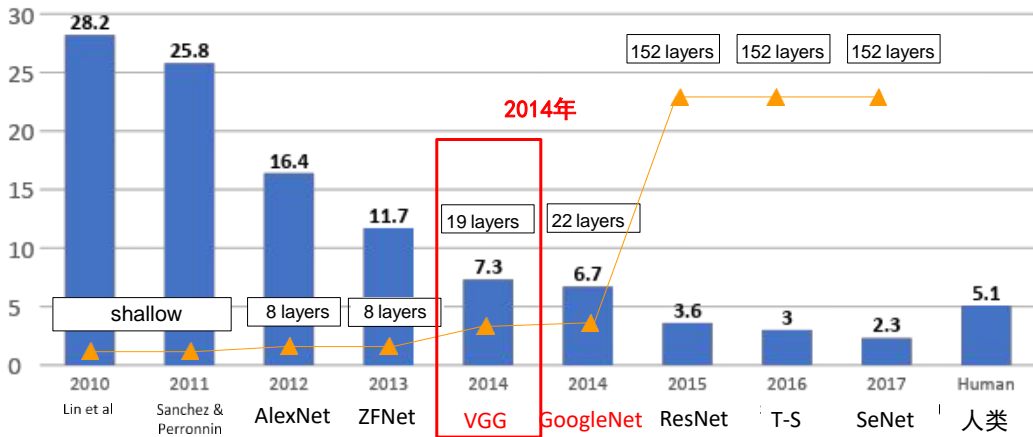
北京邮电大学计算机学院 鲁鹏

79

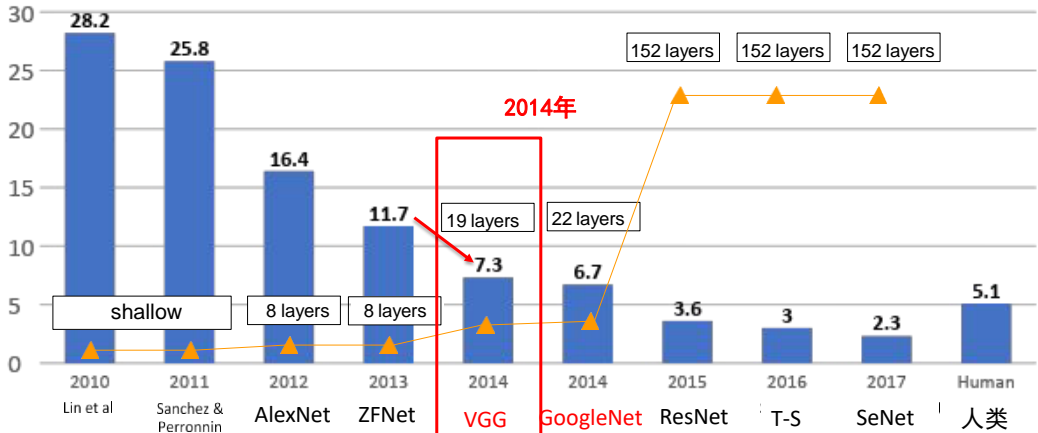
ImageNet大赛历年冠军



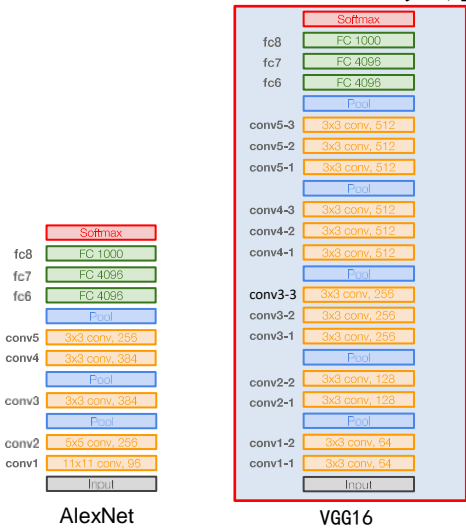
ImageNet大赛历年冠军



ImageNet大赛历年冠军



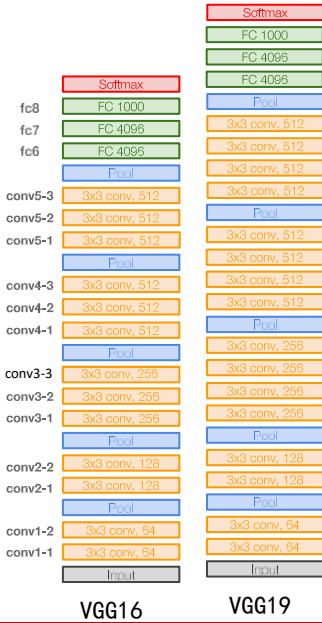
经典网络解析——VGG



VGG网络贡献:

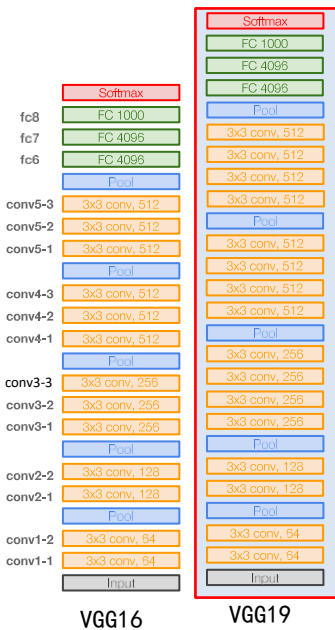
- 使用尺寸更小的3x3卷积核串联来获得更大的感受野;
- 放弃使用11x11和5x5这样的大尺寸卷积核;
- 深度更深、非线性更强, 网络的参数也更少;
- 去掉了AlexNet中的局部响应归一化层 (LRN) 层。

经典网络解析——VGG



VGG16 vs VGG19

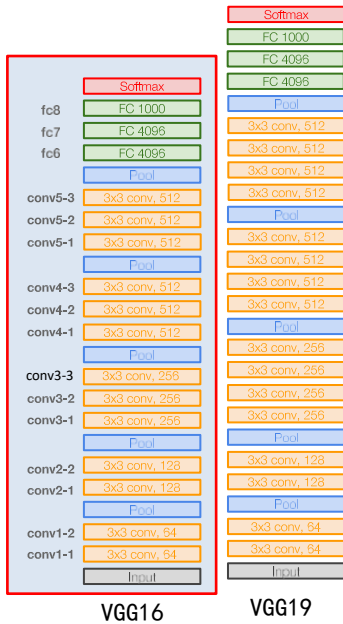
经典网络解析——VGG



VGG16 vs VGG19

19层VGG更深、精度略微，但所需内存更多

经典网络解析——VGG



VGG16 vs VGG19

19层VGG更深、精度略微，但所需内存更多

VGG16更常用

2020/4/28

北京邮电大学计算机学院 鲁鹏

86

经典网络解析——VGG



VGG16

➤ 13个卷积层与3个全连接

2020/4/28

北京邮电大学计算机学院 鲁鹏

87

经典网络解析——VGG



VGG16

➤ 13个卷积层与3个全连接

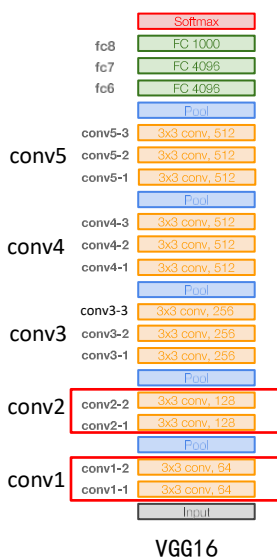
➤ 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同

2020/4/28

北京邮电大学计算机学院 鲁鹏

88

经典网络解析——VGG



VGG16

➤ 13个卷积层与3个全连接

➤ 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同

2020/4/28

北京邮电大学计算机学院 鲁鹏

89

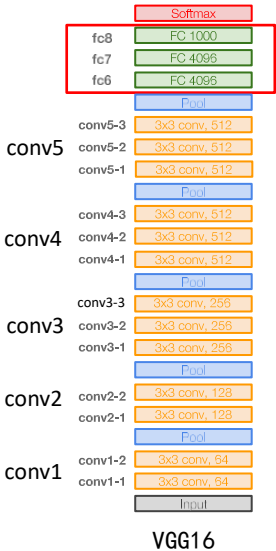
经典网络解析——VGG



VGG16

- 13个卷积层与3个全连接
- 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同

经典网络解析——VGG



VGG16

- 13个卷积层与3个全连接
- 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同

经典网络解析——VGG



VGG16

- 13个卷积层与3个全连接
- 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同
- 卷积层均采用3x3的卷积核及ReLU激活函数;

2020/4/28

北京邮电大学计算机学院 鲁鹏

92

经典网络解析——VGG



VGG16

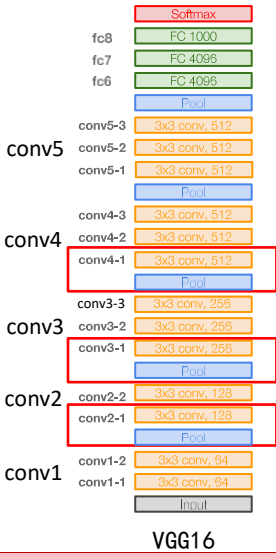
- 13个卷积层与3个全连接
- 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同
- 所有卷积层均采用3x3的卷积核及ReLU激活函数;
- 池化层均采用最大池化, 其窗口大小为2x2、步长为2;

2020/4/28

北京邮电大学计算机学院 鲁鹏

93

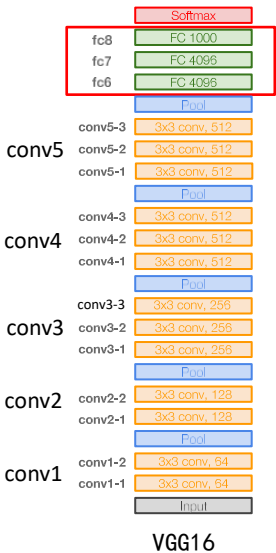
经典网络解析——VGG



VGG16

- 13个卷积层与3个全连接
- 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同
- 所有卷积层均采用3x3的卷积核及ReLU激活函数;
- 所有的池化层都采用最大池化, 其窗口大小为2x2、步长为2;
- 经过一次池化操作, 其后卷积层的卷积核个数就增加一倍, 直至到达512;

经典网络解析——VGG



VGG16

- 13个卷积层与3个全连接
- 分为5段conv1, ..., conv5, 每一段中卷积层的卷积核个数均相同
- 所有卷积层均采用3x3的卷积核及ReLU激活函数;
- 所有的池化层都采用最大池化, 其窗口大小为2x2、步长为2;
- 经过一次池化操作, 其后卷积层的卷积核个数就增加一倍, 直至到达512;
- 全连接层中也使用了Dropout策略

思考

问题1：小卷积核有哪些优势？

2020/4/28

北京邮电大学计算机学院 鲁鹏

96

思考

问题1：小卷积核有哪些优势？

回答：多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野；

2020/4/28

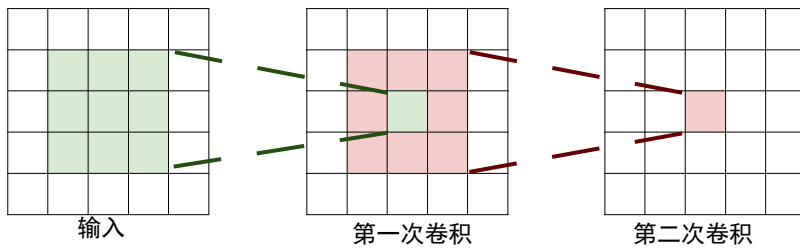
北京邮电大学计算机学院 鲁鹏

97

思考

问题1: 小卷积核有哪些优势?

回答: 多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野;



2个3x3卷积核串联, 感受野为5x5

2020/4/28

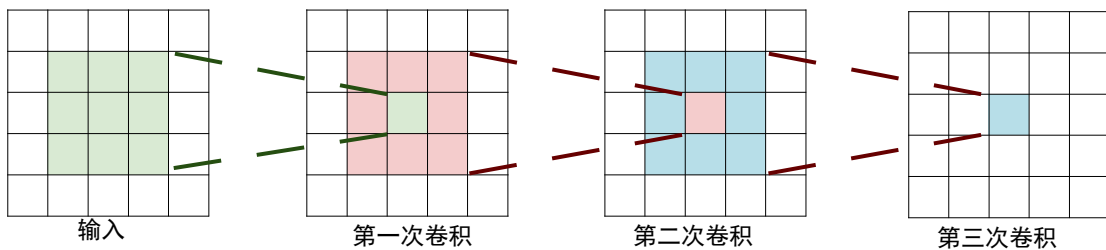
北京邮电大学计算机学院 鲁鹏

98

思考

问题1: 小卷积核有哪些优势?

回答: 多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野;



3个3x3卷积核串联, 感受野为7x7

2020/4/28

北京邮电大学计算机学院 鲁鹏

99

思考

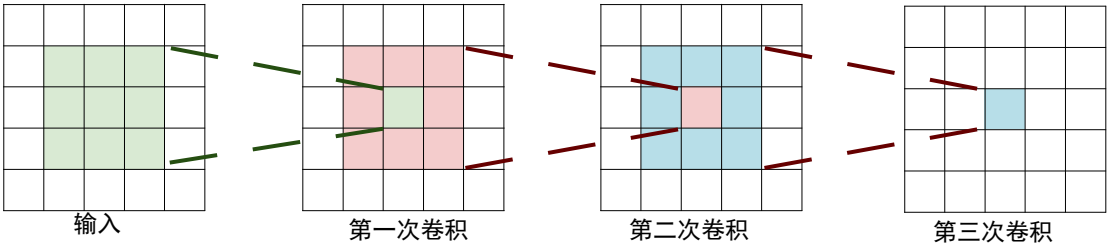
$$(3 \times 3 \times C) \times C \times 3 = 27C^2$$

$$(7 \times 7 \times C) \times C = 49C^2$$

问题1: 小卷积核有哪些优势?

回答: 多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野;

使用小卷积核串联构建的网络深度更深、非线性更强、参数也更少。



3个3x3卷积核串联, 感受野为7x7

思考

问题1: 小卷积核有哪些优势?

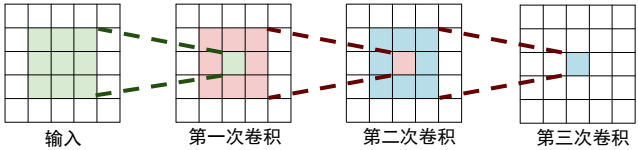
回答: 多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野;

使用小卷积核串联构建的网络深度更深、非线性更强、参数也更少。

假设卷积层输入和输出的特征图个数均为C:

三个3 x 3的卷积串联参数个数

一个7 x 7的卷积层卷积参数个数



思考

问题1: 小卷积核有哪些优势?

回答: 多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野;
使用小卷积核串联构建的网络深度更深、非线性更强、参数也更少。

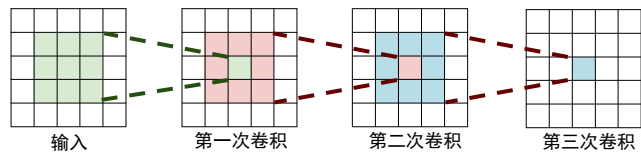
假设卷积层输入和输出的特征图个数均为C:

三个 3×3 的卷积串联参数个数

$$(3 \times 3 \times C) \times C \times 3 = 27C^2$$

一个 7×7 的卷积层卷积参数个数

$$(7 \times 7 \times C) \times C = 49C^2$$



2020/4/28

北京邮电大学计算机学院 鲁鹏

102

思考

问题1: 小卷积核有哪些优势?

回答: 多个小尺寸卷积核串联可以得到与大尺寸卷积核相同的感受野;
使用小卷积核串联构建的网络深度更深、非线性更强、参数也更少。

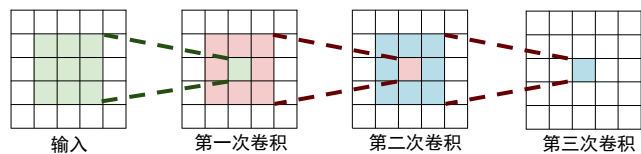
假设卷积层输入和输出的特征图个数均为C:

三个 3×3 的卷积串联参数个数

$$(3 \times 3 \times C) \times C \times 3 = 27C^2$$

一个 7×7 的卷积层卷积参数个数

$$(7 \times 7 \times C) \times C = 49C^2$$



2020/4/28

北京邮电大学计算机学院 鲁鹏

103

思考

问题2: 为什么VGG网络前四段里, 每经过一次池化操作, 卷积核个数就增加一倍?

2020/4/28

北京邮电大学计算机学院 鲁鹏

104

思考

问题2: 为什么VGG网络前四段里, 每经过一次池化操作, 卷积核个数就增加一倍?

回答:

1. 池化操作可以减小特征图尺寸, 降低显存占用

2020/4/28

北京邮电大学计算机学院 鲁鹏

105

思考

问题2：为什么VGG网络前四段里，每经过一次池化操作，卷积核个数就增加一倍？

回答：

1. 池化操作可以减小特征图尺寸，降低显存占用
2. 增加卷积核个数有助于学习更多的结构特征，但会增加网络参数数量以及内存消耗

思考

问题2：为什么VGG网络前四段里，每经过一次池化操作，卷积核个数就增加一倍？

回答：

1. 池化操作可以减小特征图尺寸，降低显存占用
2. 增加卷积核个数有助于学习更多的结构特征，但会增加网络参数数量以及内存消耗
3. 一减一增的设计平衡了识别精度与存储、计算开销

思考

问题2：为什么VGG网络前四段里，每经过一次池化操作，卷积核个数就增加一倍？

回答：

1. 池化操作可以减小特征图尺寸，降低显存占用
2. 增加卷积核个数有助于学习更多的结构特征，但会增加网络参数数量以及内存消耗
3. 一减一增的设计平衡了识别精度与存储、计算开销

最终提升了网络性能！

思考

问题3：为什么卷积核个数增加到512后就不再增加了？

思考

问题3: 为什么卷积核个数增加到512后就不再增加了?

回答:

1. 第一个全连接层含102M参数, 占总参数个数的74%;

2020/4/28

北京邮电大学计算机学院 鲁鹏

110

思考

问题3: 为什么卷积核个数增加到512后就不再增加了?

回答:

1. 第一个全连接层含102M参数, 占总参数个数的74%;
2. 这一层的参数个数是特征图的尺寸与个数的乘积;

2020/4/28

北京邮电大学计算机学院 鲁鹏

111

思考

问题3: 为什么卷积核个数增加到512后就不再增加了?

回答:

1. 第一个全连接层含102M参数, 占总参数个数的74%;
2. 这一层的参数个数是特征图的尺寸与个数的乘积;

如将最后一层卷积核个数增加至1024, 这一层参数个数为:

$$7*7*1024*4096 = 205,520,896 \\ \approx 200M$$

思考

问题3: 为什么卷积核个数增加到512后就不再增加了?

回答:

1. 第一个全连接层含102M参数, 占总参数个数的74%;
2. 这一层的参数个数是特征图的尺寸与个数的乘积;
3. 参数过多容易过拟合, 且不易被训练

如将最后一层卷积核个数增加至1024, 这一层参数个数为:

$$7*7*1024*4096 = 205,520,896 \\ \approx 200M$$

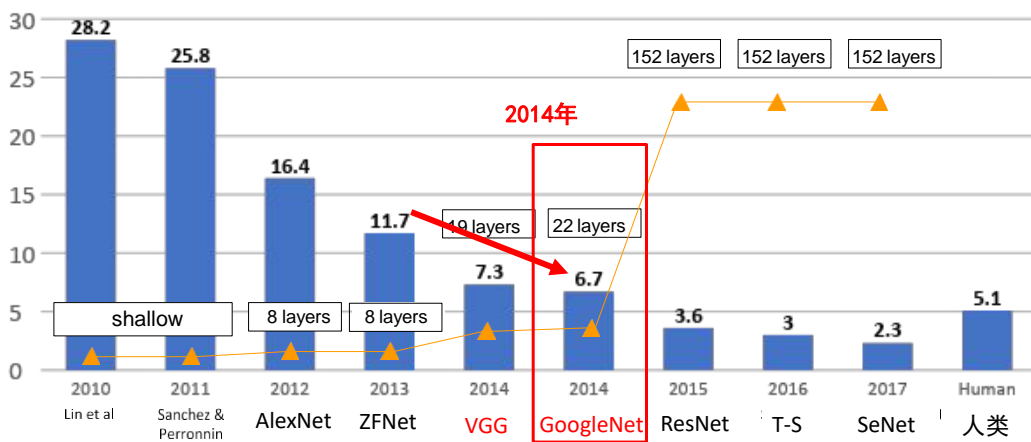
GoogLeNet

2020/4/28

北京邮电大学计算机学院 鲁鹏

114

ImageNet大赛历年冠军

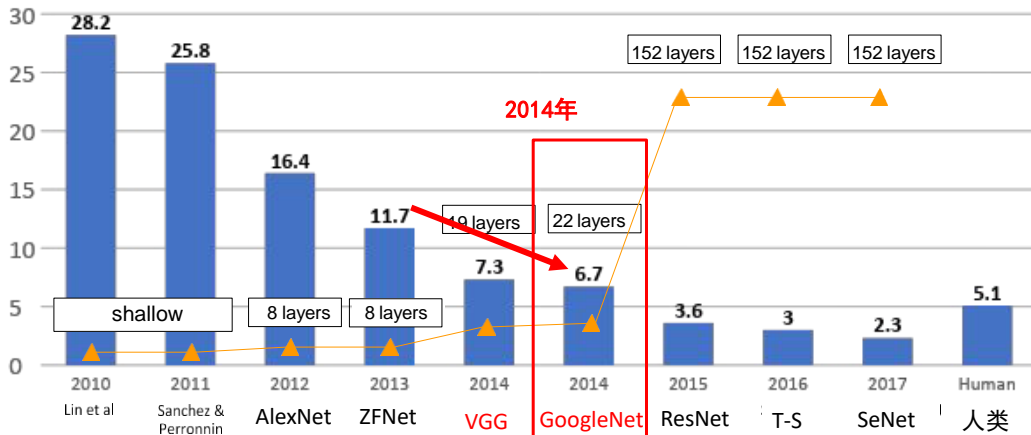


2020/4/28

北京邮电大学计算机学院 鲁鹏

115

ImageNet大赛历年冠军

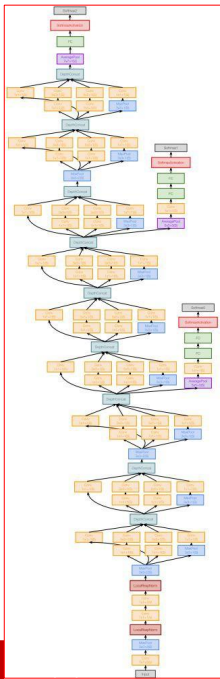


2020/4/28

北京邮电大学计算机学院 鲁鹏

116

经典网络解析——GoogLeNet

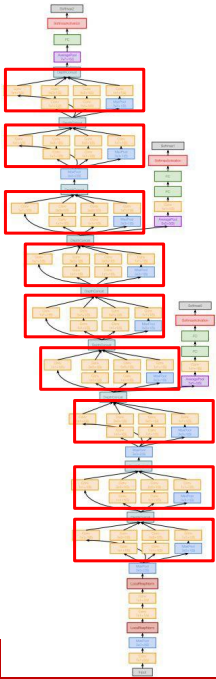


GoogLeNet的创新点：

北京邮电大学计算机学院 鲁鹏

117

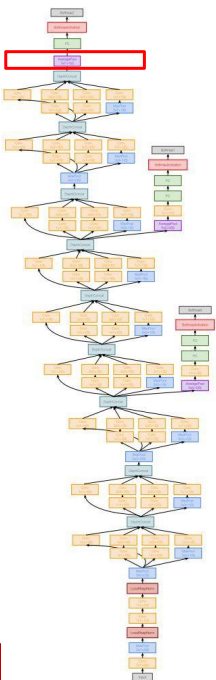
经典网络解析——GoogLeNet



GoogLeNet的创新点：

- 提出了一种Inception结构，它能保留输入信号中的更多特征信息；

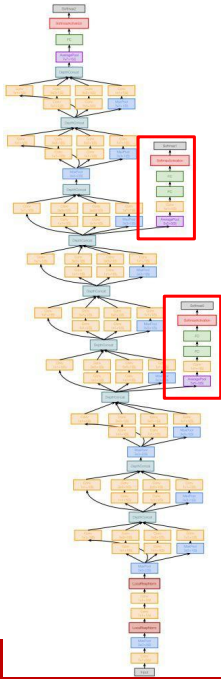
经典网络解析——GoogLeNet



GoogLeNet的创新点：

- 提出了一种Inception结构，它能保留输入信号中的更多特征信息；
- 去掉了AlexNet的前两个全连接层，并采用了平均池化，这一设计使得GoogLeNet只有500万参数，比AlexNet少了12倍；

经典网络解析——GoogLeNet



GoogLeNet的创新点：

- 提出了一种Inception结构，它能保留输入信号中的更多特征信息；
- 去掉了AlexNet的前两个全连接层，并采用了平均池化，这一设计使得GoogLeNet只有500万参数，比AlexNet少了12倍；
- 在网络的中部引入了辅助分类器，克服了训练过程中的梯度消失问题。

经典网络解析——GoogLeNet



串联结构（如VGG）存在的问题

经典网络解析——GoogLeNet



串联结构（如VGG）存在的问题——后面的卷积层只能处理前层输出的特征图；前层因某些原因（比如感受野限制）丢失重要信息，后层无法找回。

2020/4/28

北京邮电大学计算机学院 鲁鹏

122

经典网络解析——GoogLeNet



串联结构（如VGG）存在的问题——后面的卷积层只能处理前层输出的特征图；前层丢失重要信息，后层无法找回。

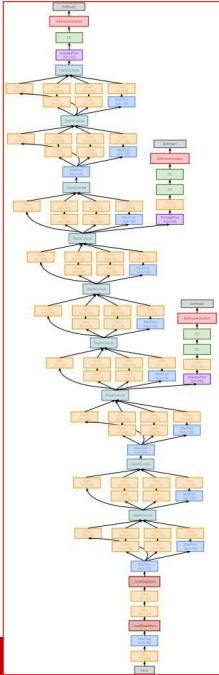
解决方案——每一层尽量多的保留输入信号中的信息。

2020/4/28

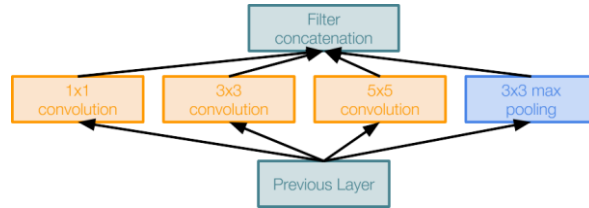
北京邮电大学计算机学院 鲁鹏

123

经典网络解析——GoogLeNet



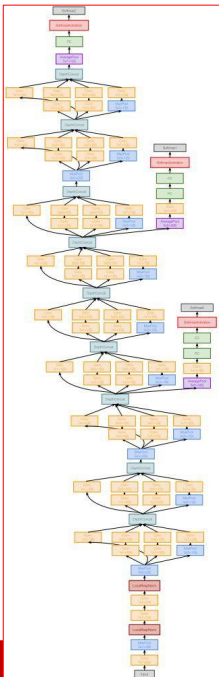
Inception
模块



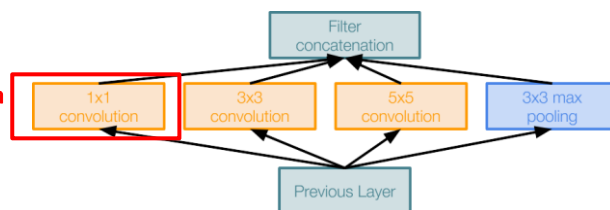
北京邮电大学计算机学院 鲁鹏

124

经典网络解析——GoogLeNet



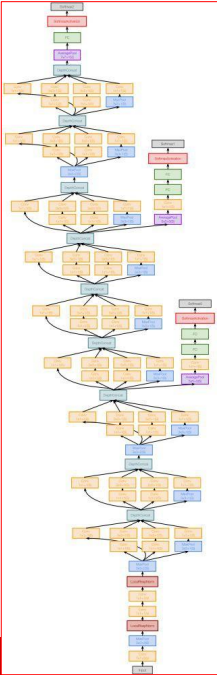
Inception
模块



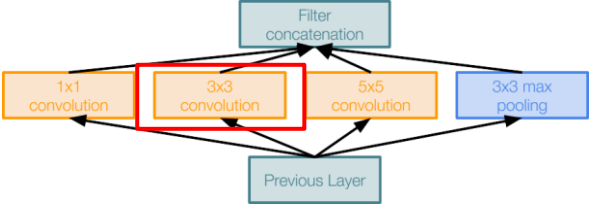
北京邮电大学计算机学院 鲁鹏

125

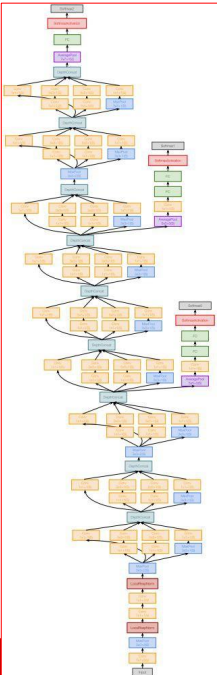
经典网络解析——GoogLeNet



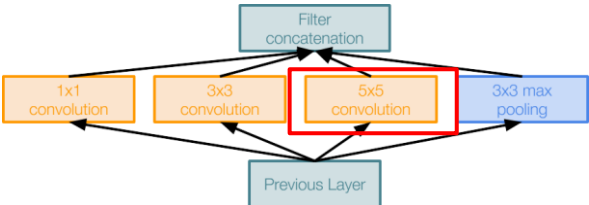
Inception
模块



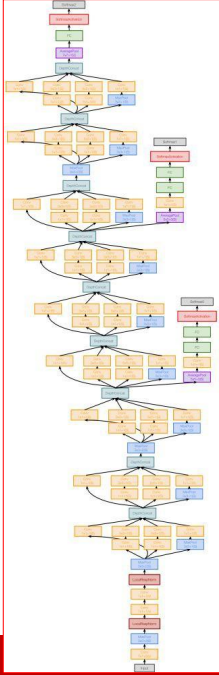
经典网络解析——GoogLeNet



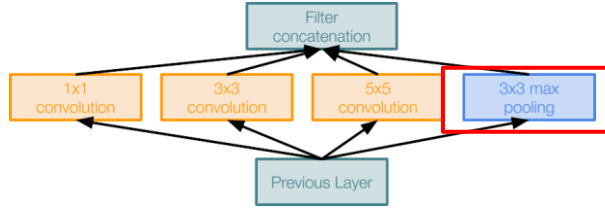
Inception
模块



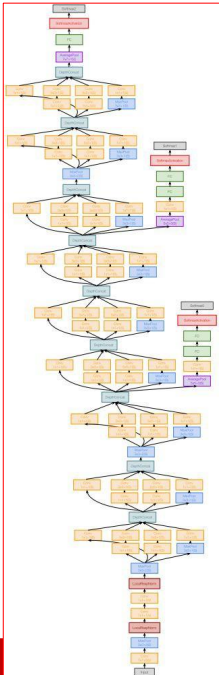
经典网络解析——GoogLeNet



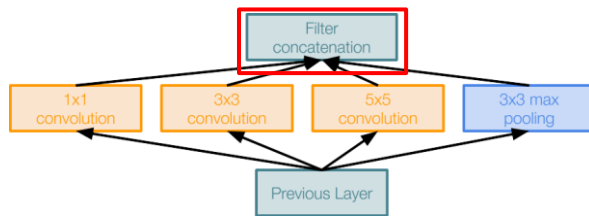
Inception
模块



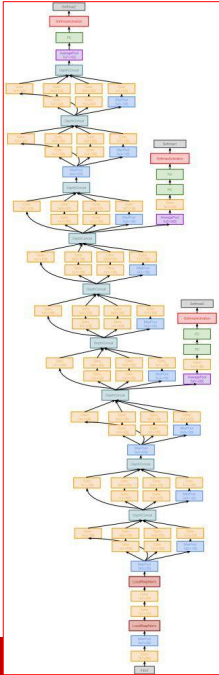
经典网络解析——GoogLeNet



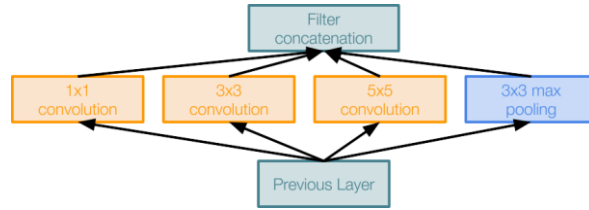
Inception
模块



经典网络解析——GoogLeNet



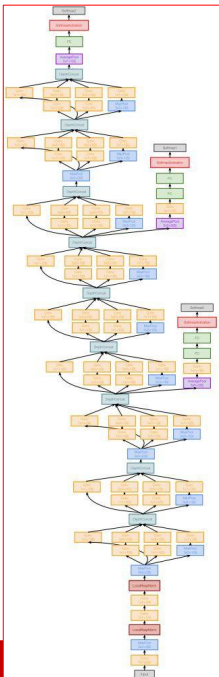
Inception
模块



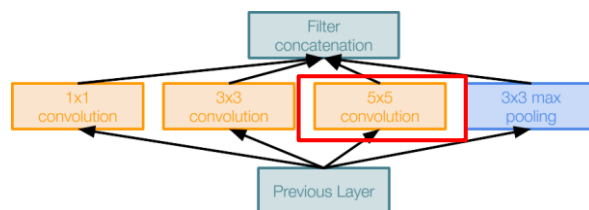
北京邮电大学计算机学院 鲁鹏

130

经典网络解析——GoogLeNet

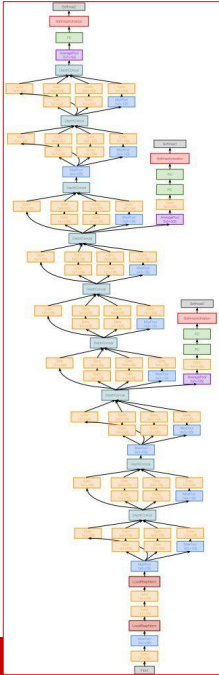


Inception
模块

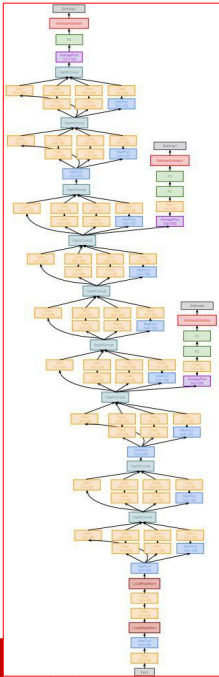
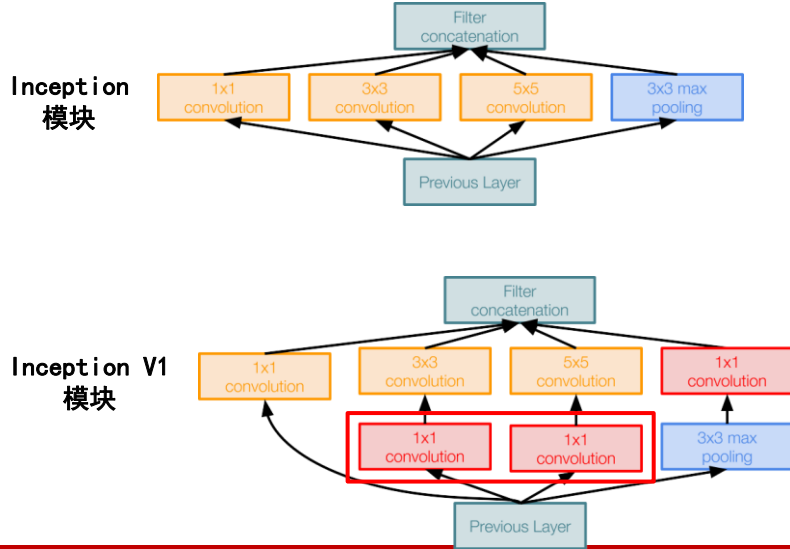


北京邮电大学计算机学院 鲁鹏

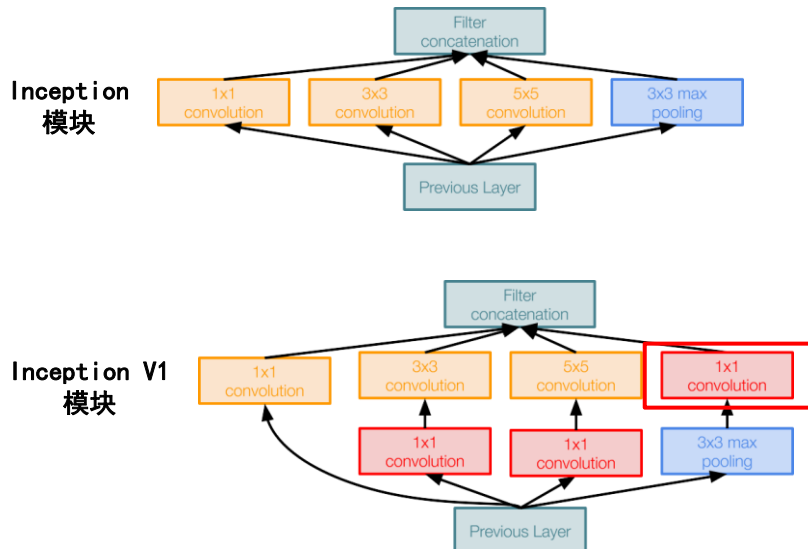
131

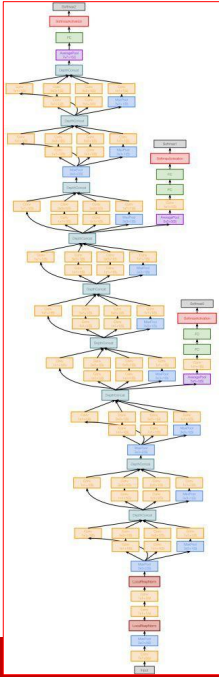


经典网络解析——GoogLeNet

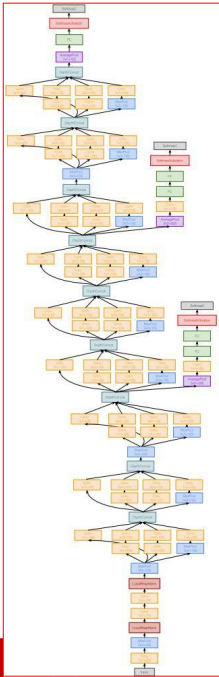
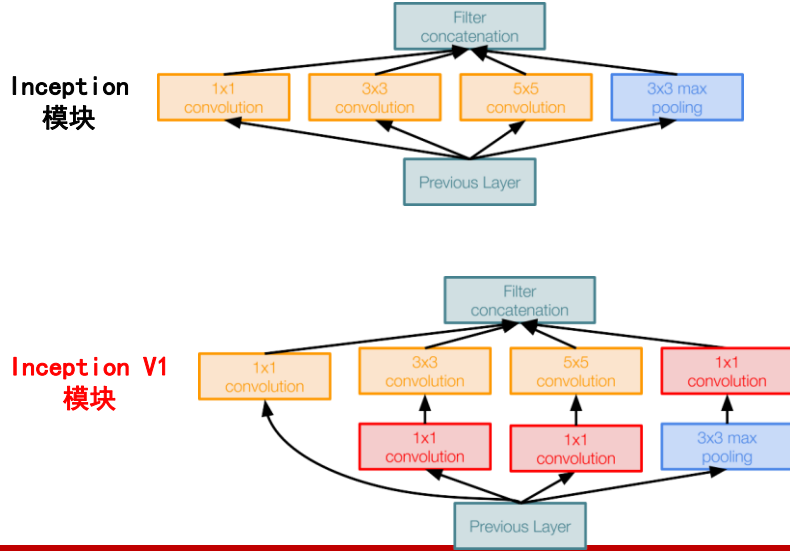


经典网络解析——GoogLeNet

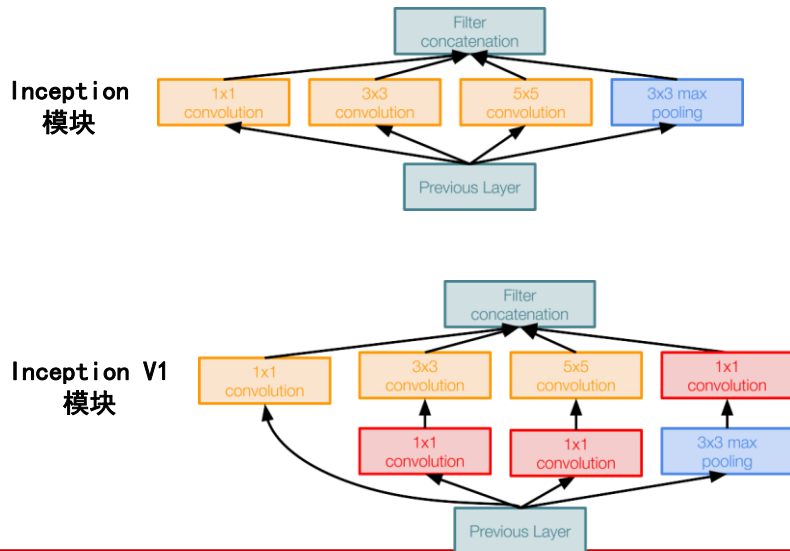


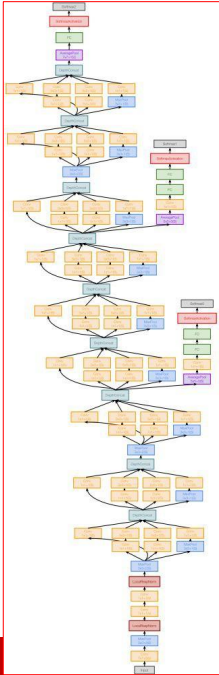


经典网络解析——GoogLeNet

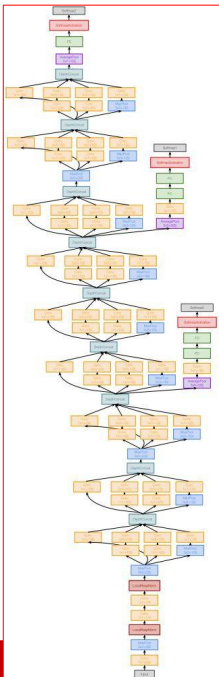
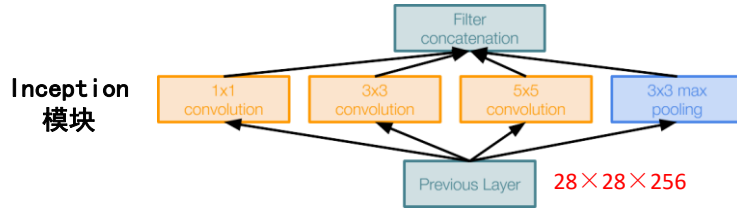


经典网络解析——GoogLeNet

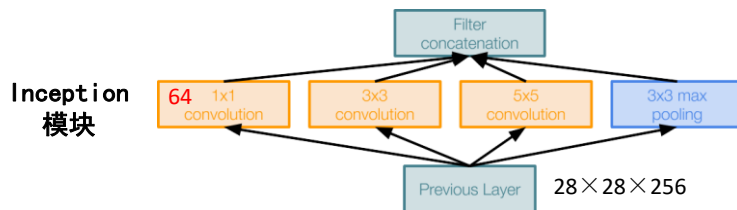




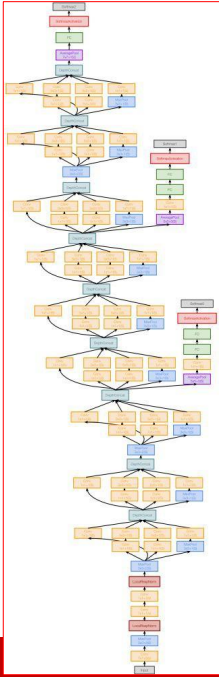
经典网络解析——GoogLeNet



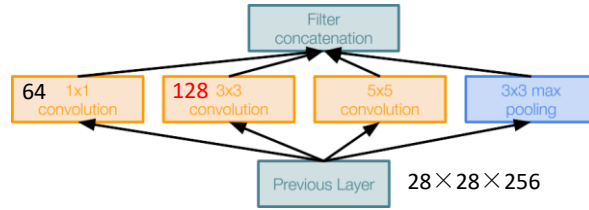
经典网络解析——GoogLeNet



经典网络解析——GoogLeNet



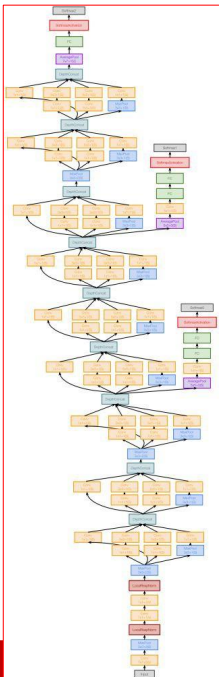
Inception
模块



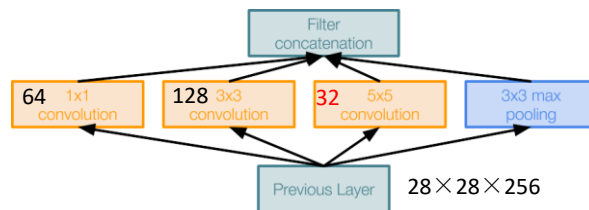
北京邮电大学计算机学院 鲁鹏

138

经典网络解析——GoogLeNet



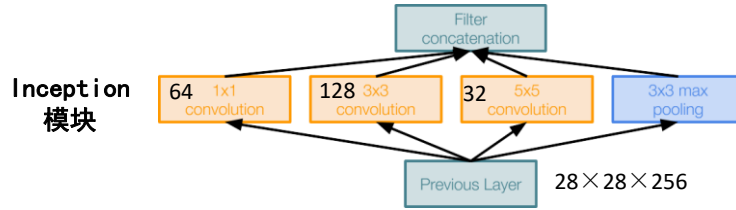
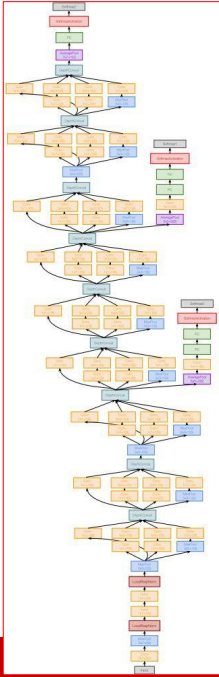
Inception
模块



北京邮电大学计算机学院 鲁鹏

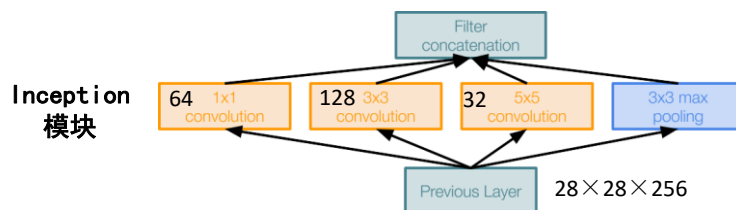
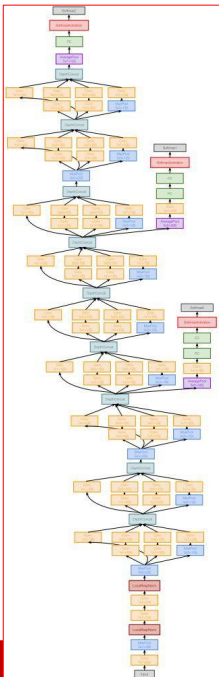
139

经典网络解析——GoogLeNet



$$\text{参数个数} = 1 \times 1 \times 256 \times 64 + 3 \times 3 \times 256 \times 128 + 5 \times 5 \times 256 \times 32 = 516,096$$

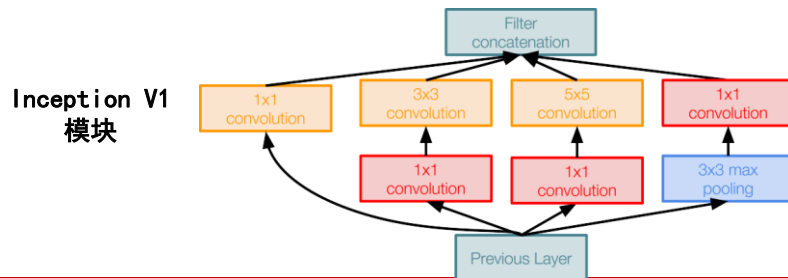
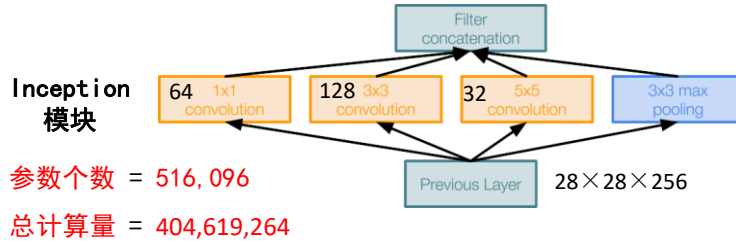
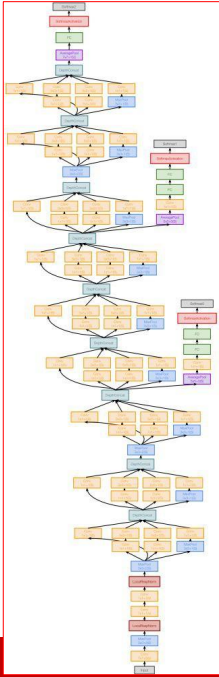
经典网络解析——GoogLeNet



$$\text{参数个数} = 1 \times 1 \times 256 \times 64 + 3 \times 3 \times 256 \times 128 + 5 \times 5 \times 256 \times 32 = 516,096$$

$$\text{总计算量} = 28 \times 28 \times 516,096 = 404,619,264$$

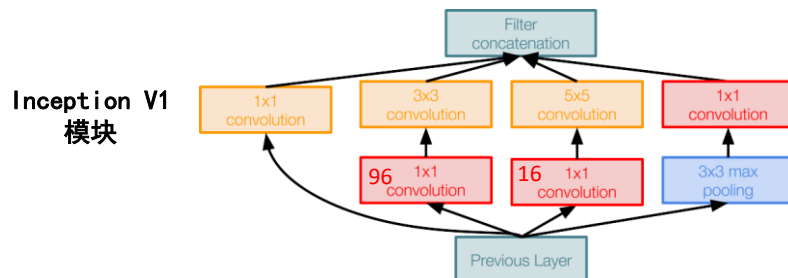
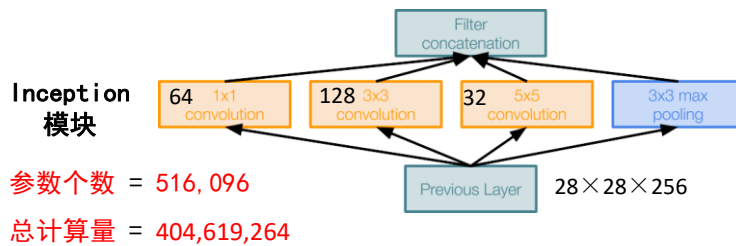
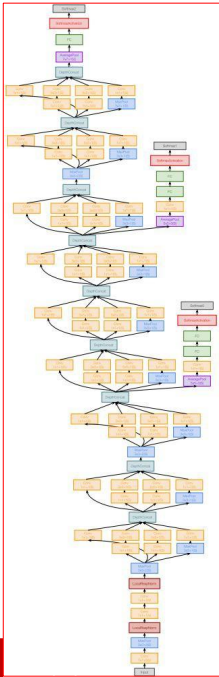
经典网络解析——GoogLeNet



北京邮电大学计算机学院 鲁鹏

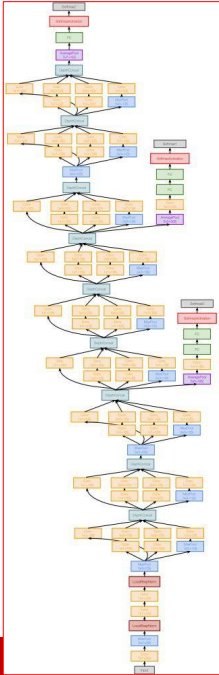
142

经典网络解析——GoogLeNet

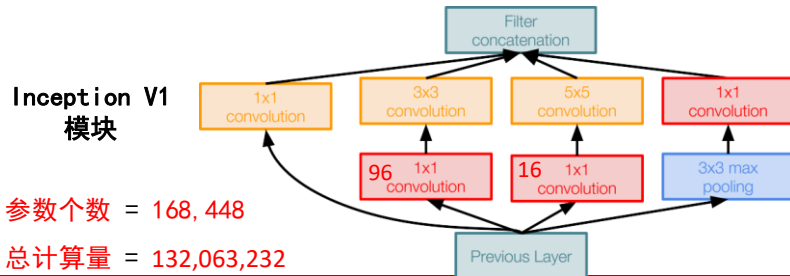
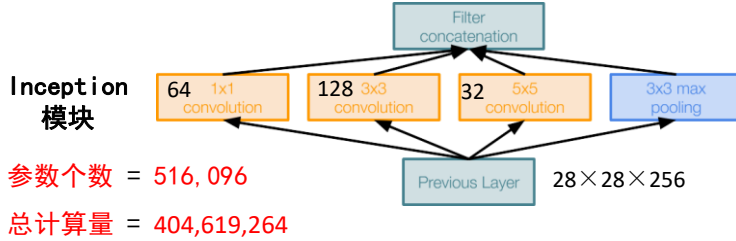


北京邮电大学计算机学院 鲁鹏

143

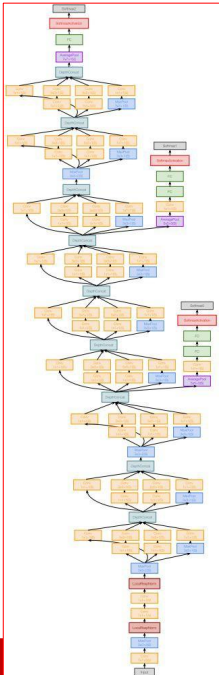


经典网络解析——GoogLeNet

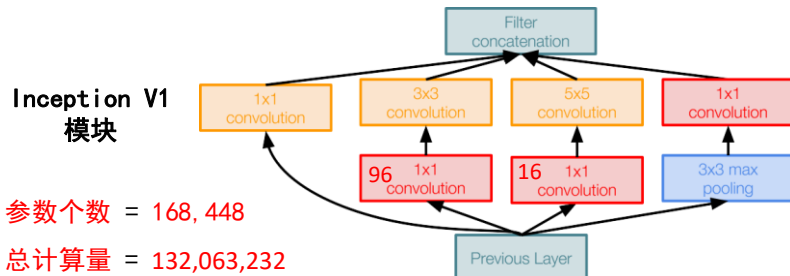
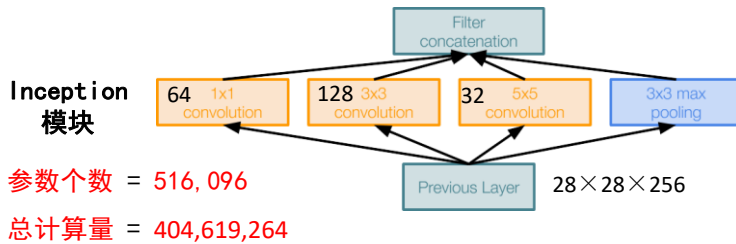


北京邮电大学计算机学院 鲁鹏

144



经典网络解析——GoogLeNet

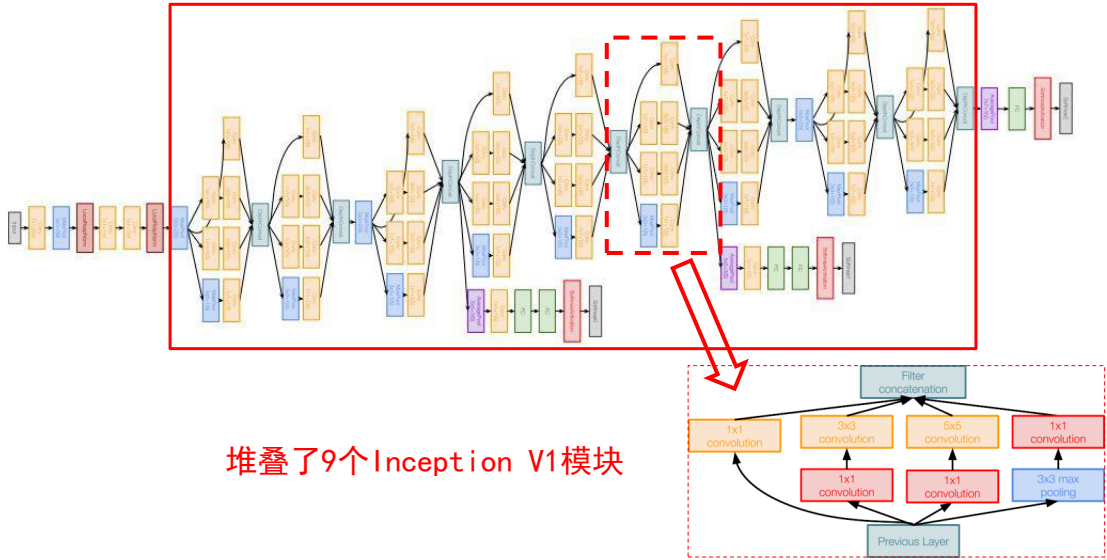


层数更深、
参数更少、
计算效率更
高、非线性
表达能力也
更强

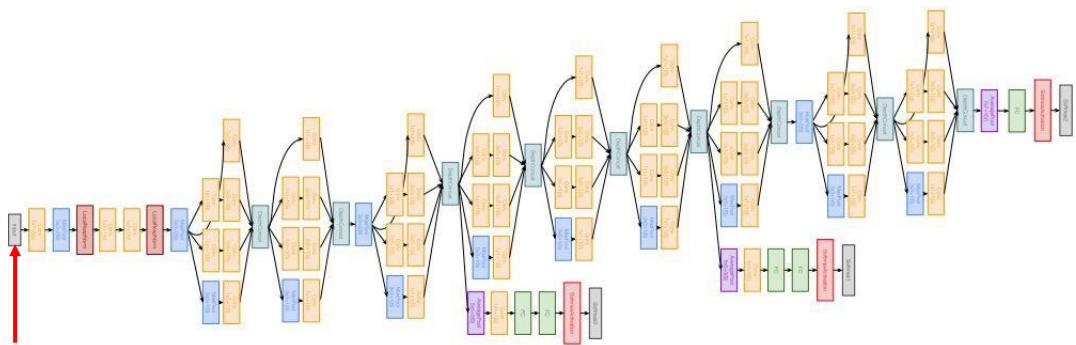
北京邮电大学计算机学院 鲁鹏

145

经典网络解析——GoogLeNet

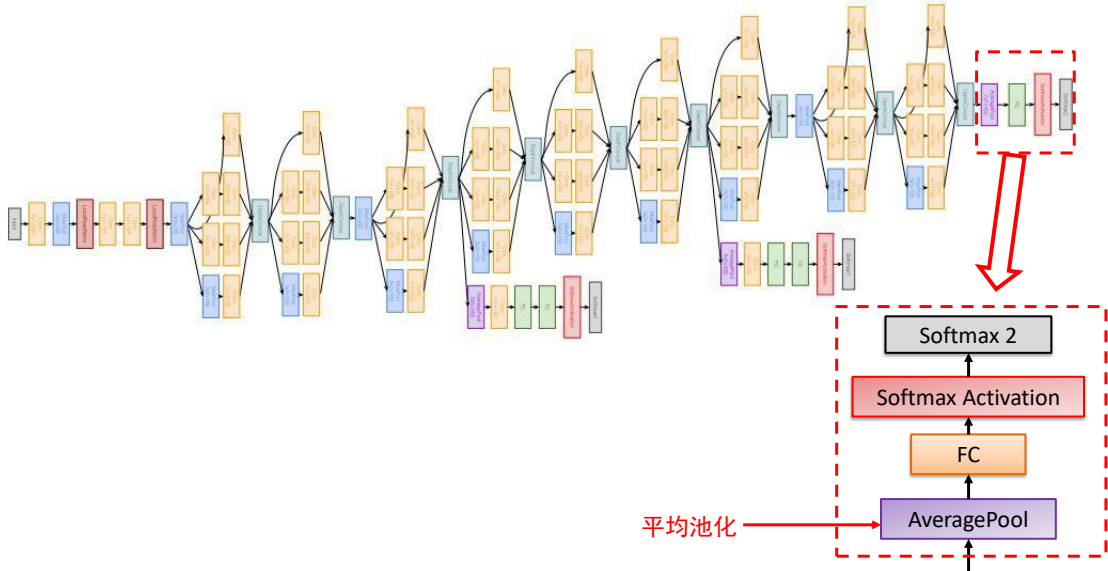


经典网络解析——GoogLeNet

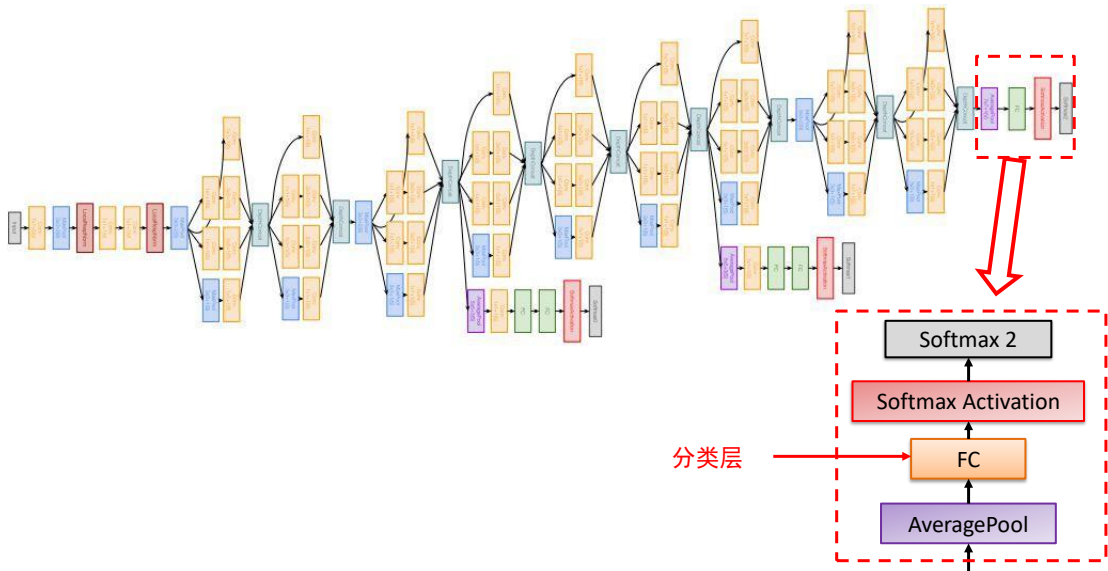


输入：减去均值后224x224彩色图像

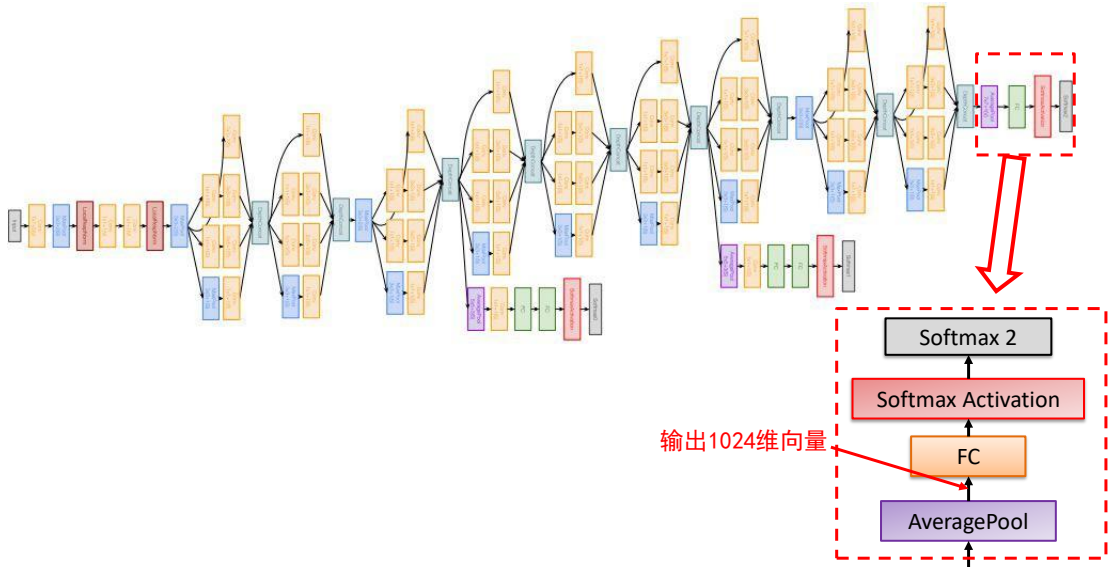
经典网络解析——GoogLeNet



经典网络解析——GoogLeNet



经典网络解析——GoogLeNet

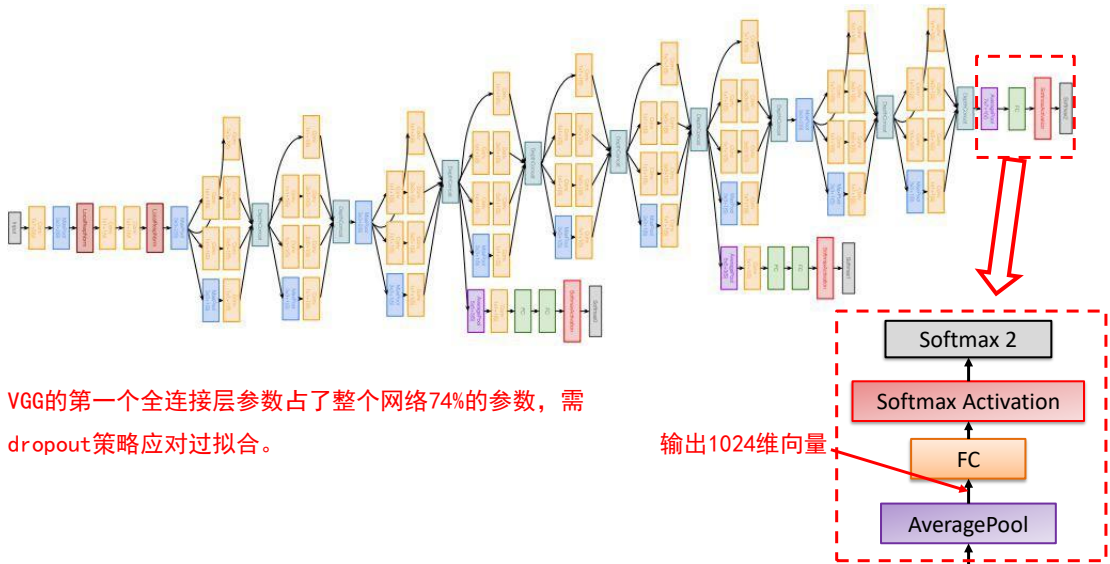


2020/4/28

北京邮电大学计算机学院 鲁鹏

输入1024个特征图 150

经典网络解析——GoogLeNet



- VGG的第一个全连接层参数占了整个网络74%的参数，需 dropout策略应对过拟合。

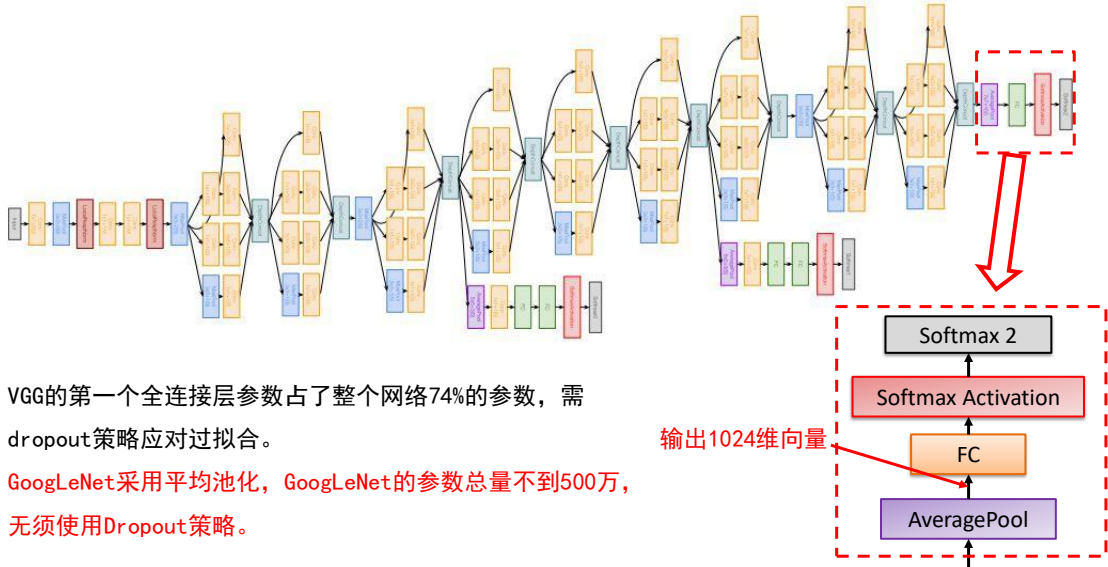
输出1024维向量

2020/4/28

北京邮电大学计算机学院 鲁鹏

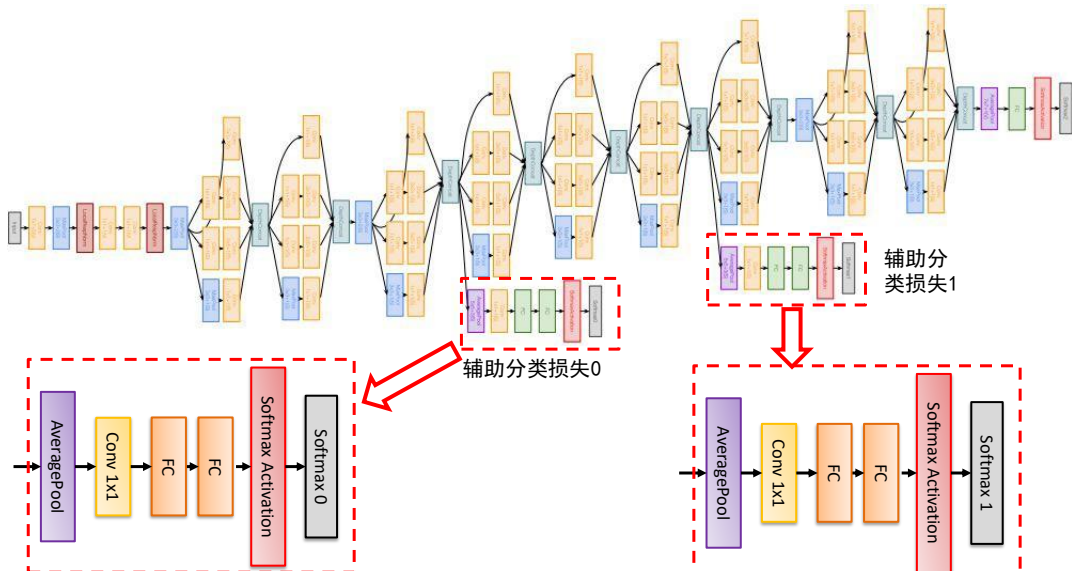
输入1024个特征图 151

经典网络解析——GoogLeNet

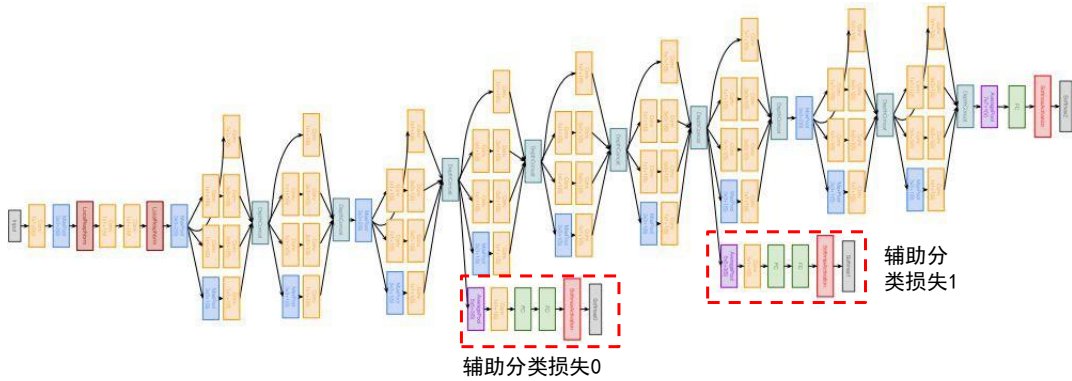


- VGG的第一个全连接层参数占了整个网络74%的参数，需 dropout策略应对过拟合。
- GoogLeNet采用平均池化，GoogLeNet的参数总量不到500万，无须使用Dropout策略。

经典网络解析——GoogLeNet



经典网络解析——GoogLeNet



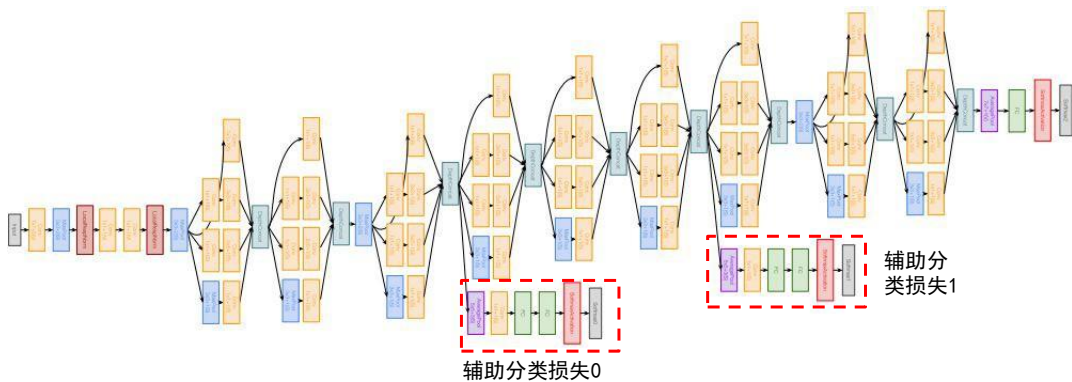
- 原因——虽然ReLU单元能够一定程度解决梯度消失问题，但是并不能完全解决深层网络难以训练的问题。离输出远的层就不如靠近输出的层训练得好。

2020/4/28

北京邮电大学计算机学院 鲁鹏

154

经典网络解析——GoogLeNet



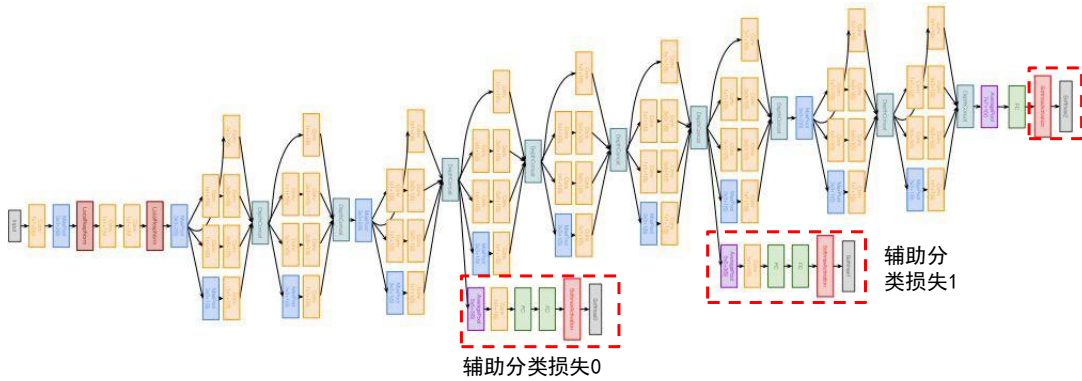
- 原因——虽然ReLU单元能够一定程度解决梯度消失问题，但是并不能完全解决深层网络难以训练的问题。离输出远的层就不如靠近输出的层训练得好。
- 结果——让低层的卷积层学习到的特征也有很好的区分能力，从而让网络更好地被训练，而且低层的卷积层学到了好的特征也能加速整个网络的收敛。

2020/4/28

北京邮电大学计算机学院 鲁鹏

155

经典网络解析——GoogLeNet



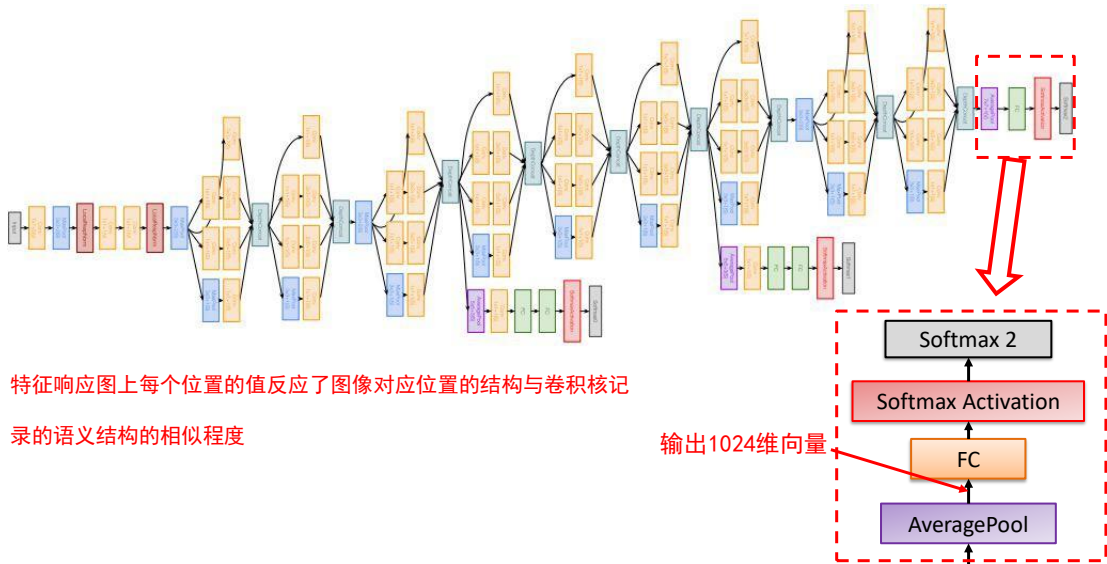
- 原因——虽然ReLU单元能够一定程度解决梯度消失问题，但是并不能完全解决深层网络难以训练的问题。离输出远的层就不如靠近输出的层训练得好。
- 结果——让低层的卷积层学习到的特征也有很好的区分能力，从而让网络更好地被训练，而且低层的卷积层学到了好的特征也能加速整个网络的收敛。
- 网络推断——仅利用网络最后的输出作为预测结果，忽略辅助分类器的输出。

2020/4/28

北京邮电大学计算机学院 鲁鹏

156

问题1: 平均池化向量化与直接展开向量化有什么区别?



- 特征响应图上每个位置的值反应了图像对应位置的结构与卷积核记录的语义结构的相似程度

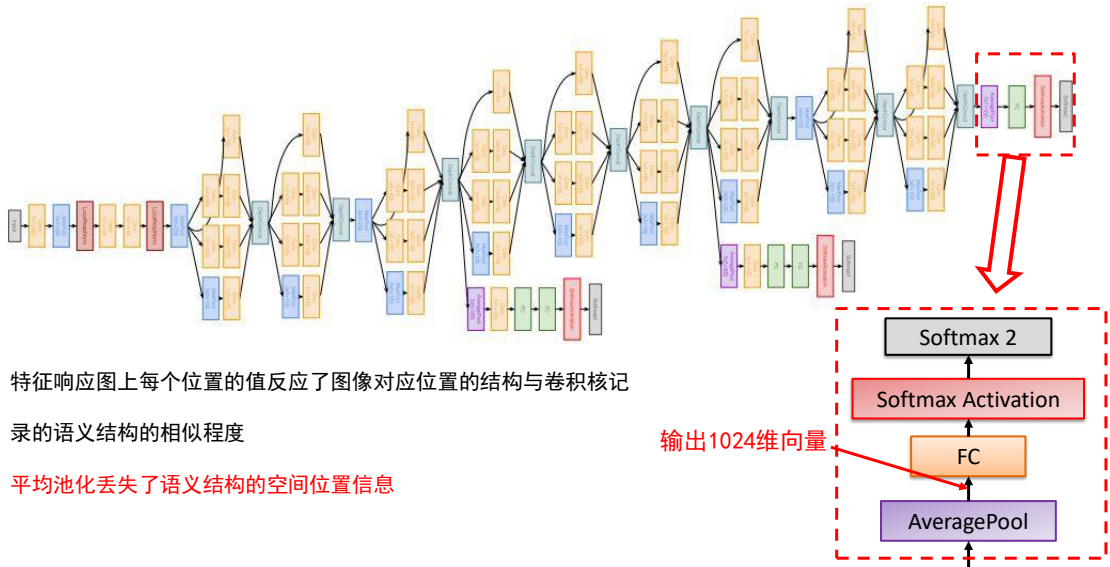
输出1024维向量

输入1024个特征图 157

2020/4/28

北京邮电大学计算机学院 鲁鹏

问题1: 平均池化向量化与直接展开向量化有什么区别?

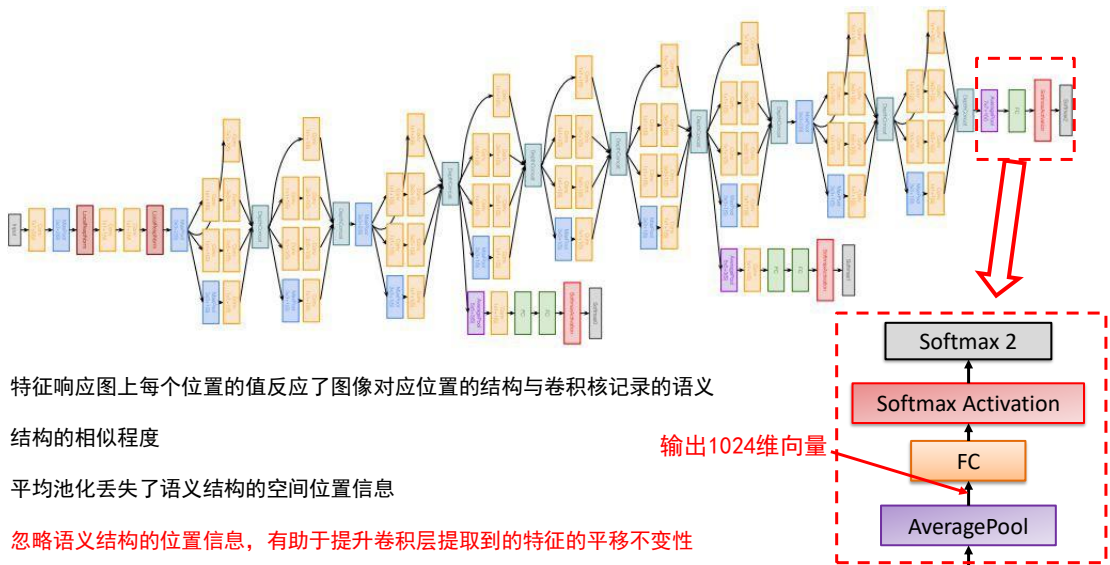


2020/4/28

北京邮电大学计算机学院 鲁鹏

输入1024个特征图 158

问题1: 平均池化向量化与直接展开向量化有什么区别?

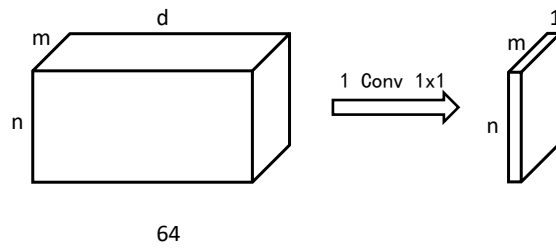


2020/4/28

北京邮电大学计算机学院 鲁鹏

输入1024个特征图 159

问题2: 利用1x1卷积进行压缩会损失信息吗?

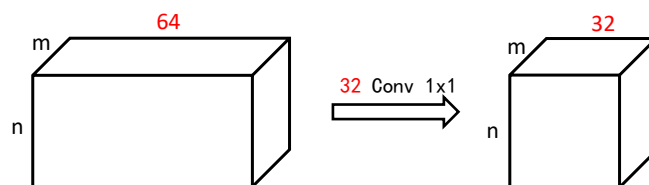


2020/4/28

北京邮电大学计算机学院 鲁鹏

160

问题2: 利用1x1卷积进行压缩会损失信息吗?

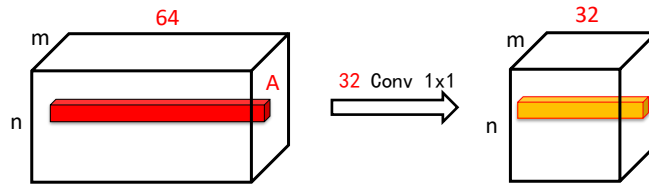


2020/4/28

北京邮电大学计算机学院 鲁鹏

161

问题2: 利用1x1卷积进行压缩会损失信息吗?



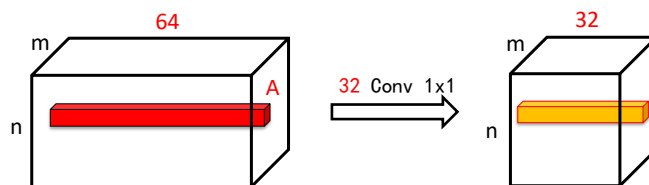
这种压缩是否会损失信息呢?

2020/4/28

北京邮电大学计算机学院 鲁鹏

162

问题2: 利用1x1卷积进行压缩会损失信息吗?



这种压缩是否会损失信息呢?

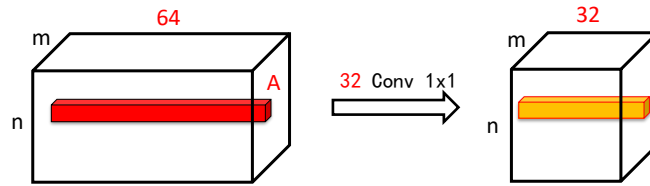
位置A的这个64维向量是一个非常稀疏向量

2020/4/28

北京邮电大学计算机学院 鲁鹏

163

问题2: 利用1x1卷积进行压缩会损失信息吗?



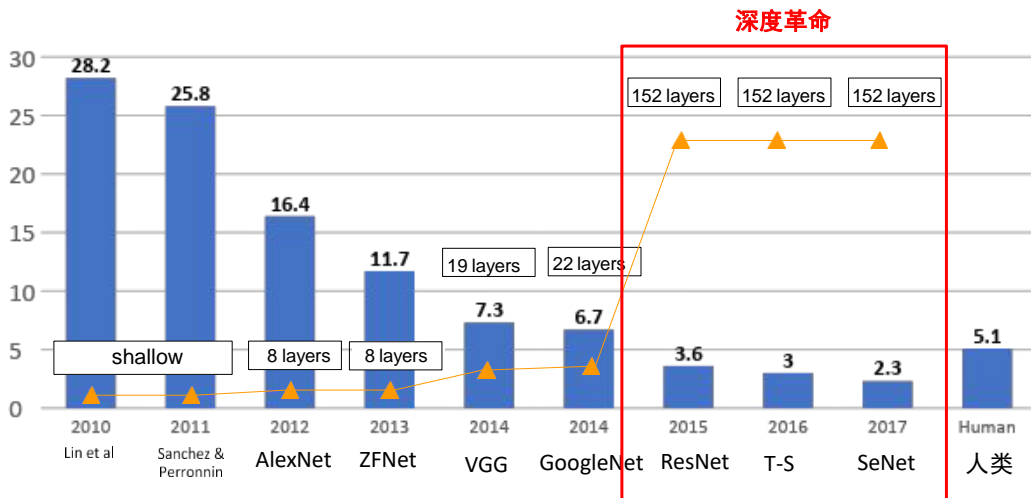
这种压缩是否会损失信息呢?

位置A的这个64维向量是一个非常稀疏向量

利用1x1卷积进行非线性压缩通常不会损失信息。

ResNet

ImageNet大赛历年冠军



2020/4/28

北京邮电大学计算机学院 鲁鹏

经典网络解析——ResNet

实验：持续向一个“基础”的卷积神经网络上面叠加更深的层数会发生什么？

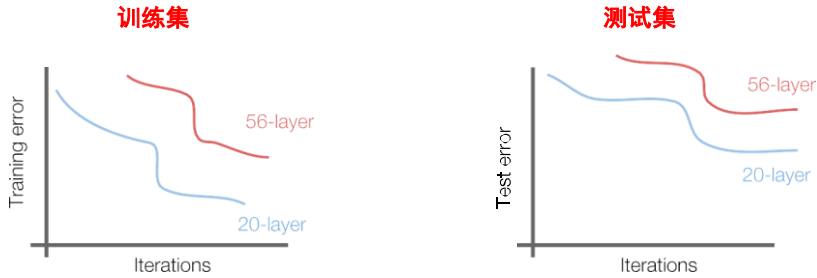
2020/4/28

北京邮电大学计算机学院 鲁鹏

167

经典网络解析——ResNet

实验：持续向一个“基础”的卷积神经网络上面叠加更深的层数会发生什么？



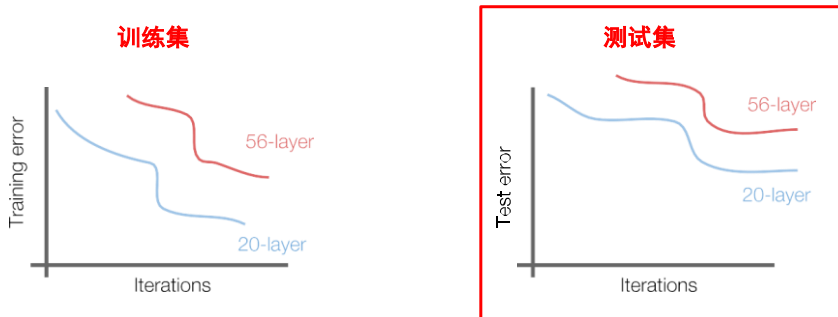
2020/4/28

北京邮电大学计算机学院 鲁鹏

168

经典网络解析——ResNet

实验：持续向一个“基础”的卷积神经网络上面叠加更深的层数会发生什么？



猜测：加深网络层数引起过拟合，
导致错误率上升

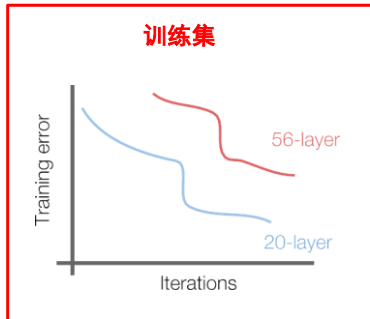
2020/4/28

北京邮电大学计算机学院 鲁鹏

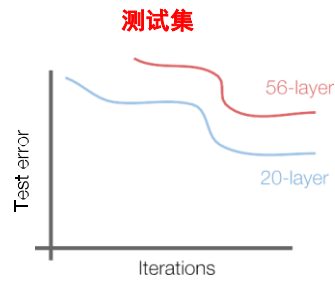
169

经典网络解析——ResNet

实验：持续向一个“基础”的卷积神经网络上面叠加更深的层数会发生什么？



这是为什么呢？



猜测：加深网络层数引起过拟合，
导致错误率上升

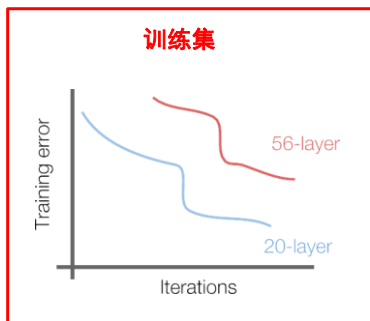
2020/4/28

北京邮电大学计算机学院 鲁鹏

170

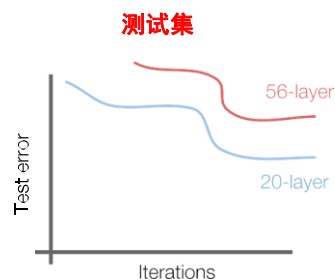
经典网络解析——ResNet

实验：持续向一个“基础”的卷积神经网络上面叠加更深的层数会发生什么？



这是为什么呢？

原因：训练过程中网络的正、反向信息流动不顺畅，
网络没有被充分训练。



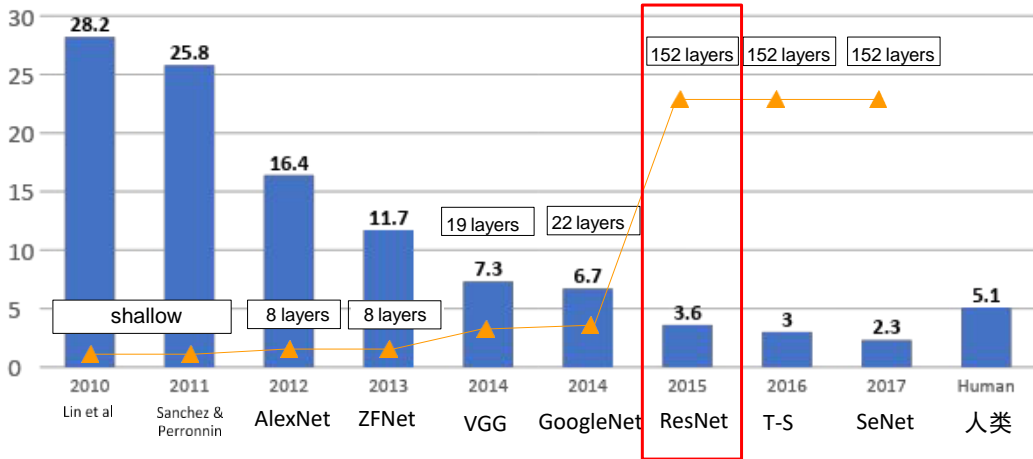
猜测：加深网络层数引起过拟合，
导致错误率上升

2020/4/28

北京邮电大学计算机学院 鲁鹏

171

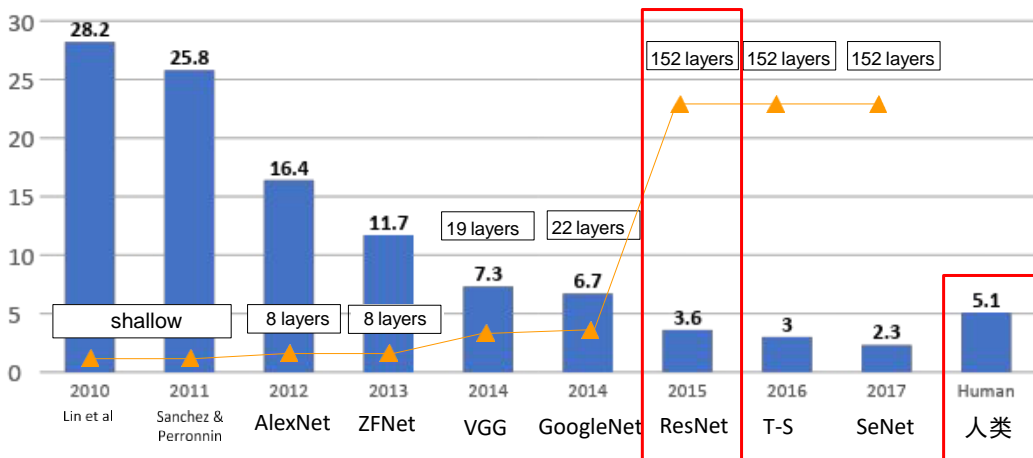
ImageNet大赛历年冠军



2020/4/28

北京邮电大学计算机学院 鲁鹏

ImageNet大赛历年冠军



2020/4/28

北京邮电大学计算机学院 鲁鹏

经典网络解析——ResNet

ResNet具有以下贡献：

2020/4/28

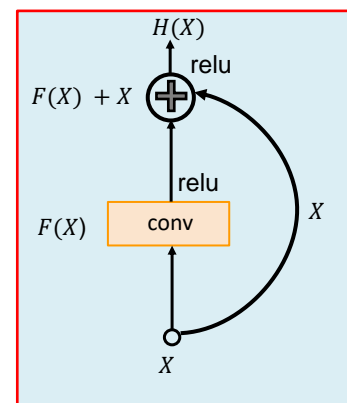
北京邮电大学计算机学院 鲁鹏

174

经典网络解析——ResNet

ResNet具有以下贡献：

- 提出了一种残差模块，通过堆叠残差模块可以构建任意深度的神经网络，而不会出现“退化”现象。



残差模块

2020/4/28

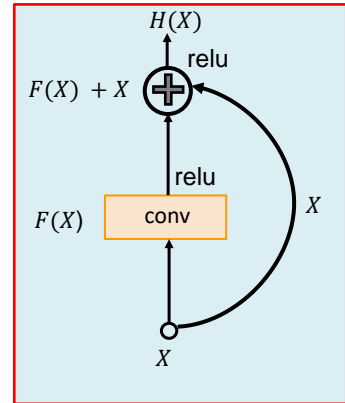
北京邮电大学计算机学院 鲁鹏

175

经典网络解析——ResNet

ResNet具有以下贡献：

- 提出了一种残差模块，通过堆叠残差模块可以构建任意深度的神经网络，而不会出现“退化”现象。
- 提出了批归一化方法来对抗梯度消失，该方法降低了网络训练过程对于权重初始化的依赖；



残差模块

2020/4/28

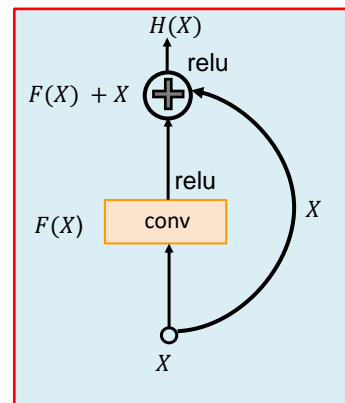
北京邮电大学计算机学院 鲁鹏

176

经典网络解析——ResNet

ResNet具有以下贡献：

- 提出了一种残差模块，通过堆叠残差模块可以构建任意深度的神经网络，而不会出现“退化”现象。
- 提出了批归一化方法来对抗梯度消失，该方法降低了网络训练过程对于权重初始化的依赖；
- 提出了一种针对ReLU激活函数的初始化方法；



残差模块

2020/4/28

北京邮电大学计算机学院 鲁鹏

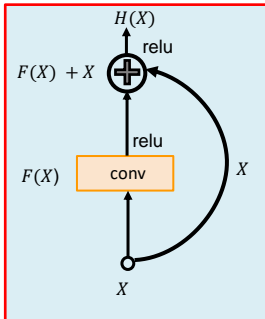
177

经典网络解析——ResNet

研究者考虑了这样一个问题：

浅层网络学习到了有效的分类模式后，如何向上堆积新层来建立更深的网络，使其满足即使不能提升浅层网络的性能，深层网络也不应降低性能。

恒等映射结构



残差模块

解决方案：残差模块

假设卷积层学习的变换为 $F(X)$ ，残差结构的输出是 $H(X)$ ，则有：

$$H(X) = F(X) + X$$

$$F(X) = H(X) - X$$

$$\begin{array}{c} \uparrow \quad \uparrow \quad \uparrow \\ \text{残差} = \text{输出} - \text{输入} \end{array}$$

2020/4/28

北京邮电大学计算机学院 鲁鹏

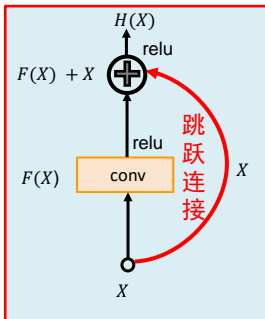
180

经典网络解析——ResNet

研究者考虑了这样一个问题：

浅层网络学习到了有效的分类模式后，如何向上堆积新层来建立更深的网络，使其满足即使不能提升浅层网络的性能，深层网络也不应降低性能。

恒等映射结构



残差模块

解决方案：残差模块

假设卷积层学习的变换为 $F(X)$ ，残差结构的输出是 $H(X)$ ，则有：

$$H(X) = F(X) + X$$

$$F(X) = H(X) - X$$

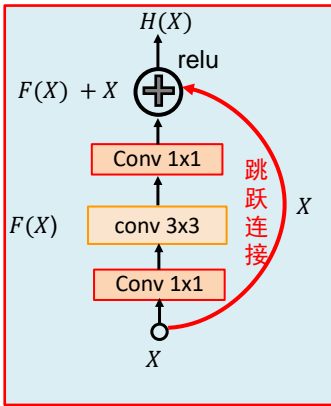
$$\begin{array}{c} \uparrow \quad \uparrow \quad \uparrow \\ \text{残差} = \text{输出} - \text{输入} \end{array}$$

2020/4/28

北京邮电大学计算机学院 鲁鹏

181

经典网络解析——ResNet



“瓶颈”结构残差模块

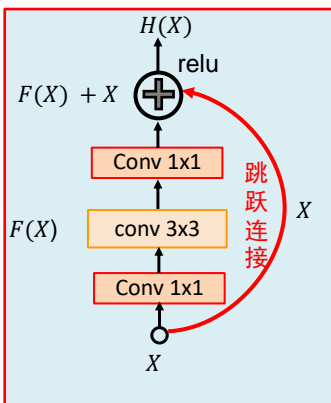
关于残差结构：

2020/4/28

北京邮电大学计算机学院 鲁鹏

182

经典网络解析——ResNet



“瓶颈”结构残差模块

关于残差结构：

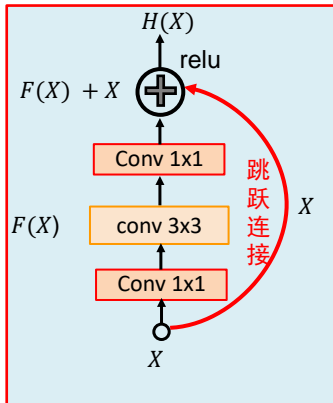
1. 残差结构能够避免普通的卷积层堆叠存在信息丢失问题，保证前向信息流的顺畅。

2020/4/28

北京邮电大学计算机学院 鲁鹏

183

经典网络解析——ResNet



“瓶颈”结构残差模块

关于残差结构：

1. 残差结构能够避免普通的卷积层堆叠存在信息丢失问题，保证前向信息流的顺畅。
2. 残差结构能够应对梯度反传过程中的梯度消失问题，保证反向梯度流的通顺。

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

残差模块组成

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

残差模块重复堆叠次数

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

2020/4/28

北京邮电大学计算机学院 鲁鹏

188

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

2020/4/28

北京邮电大学计算机学院 鲁鹏

189

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

2020/4/28

北京邮电大学计算机学院 鲁鹏

190

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

2020/4/28

北京邮电大学计算机学院 鲁鹏

191

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

2020/4/28

北京邮电大学计算机学院 鲁鹏

192

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

其他关键点：

2020/4/28

北京邮电大学计算机学院 鲁鹏

193

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

其他关键点:

- 提出了批归一化策略，降低了网络训练过程对于权重初始化的依赖

经典网络解析——ResNet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

其他关键点:

- 提出了批归一化策略，降低了网络训练过程对于权重初始化的依赖
- 提出了针对ReLU非线性单元的初始化方法

问题：为什么残差网络性能这么好？

2020/4/28

北京邮电大学计算机学院 鲁鹏

196

问题：为什么残差网络性能这么好？

一种典型的解释：残差网络可以看作是一种集成模型！

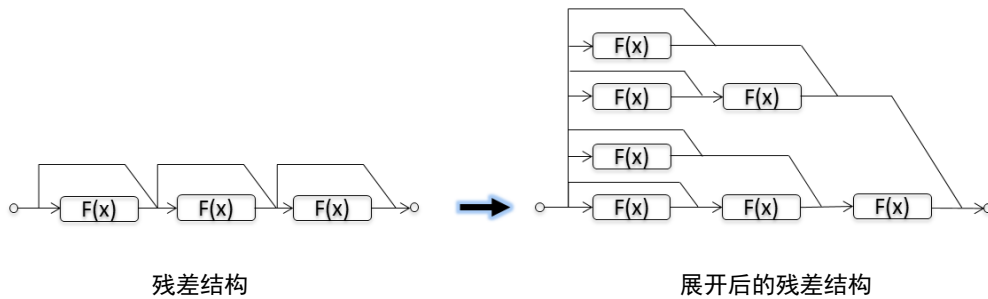
2020/4/28

北京邮电大学计算机学院 鲁鹏

197

问题：为什么残差网络性能这么好？

一种典型的解释：残差网络可以看作是一种集成模型！



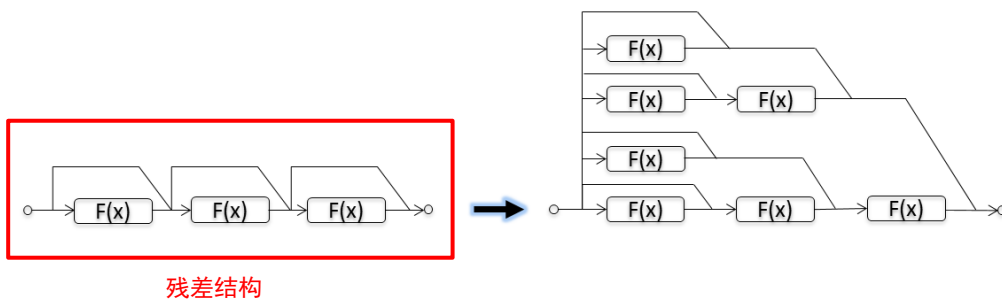
2020/4/28

北京邮电大学计算机学院 鲁鹏

198

问题：为什么残差网络性能这么好？

一种典型的解释：残差网络可以看作是一种集成模型！



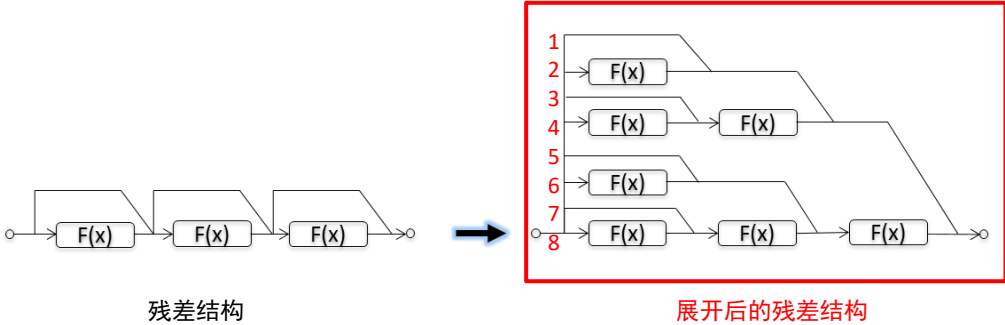
2020/4/28

北京邮电大学计算机学院 鲁鹏

199

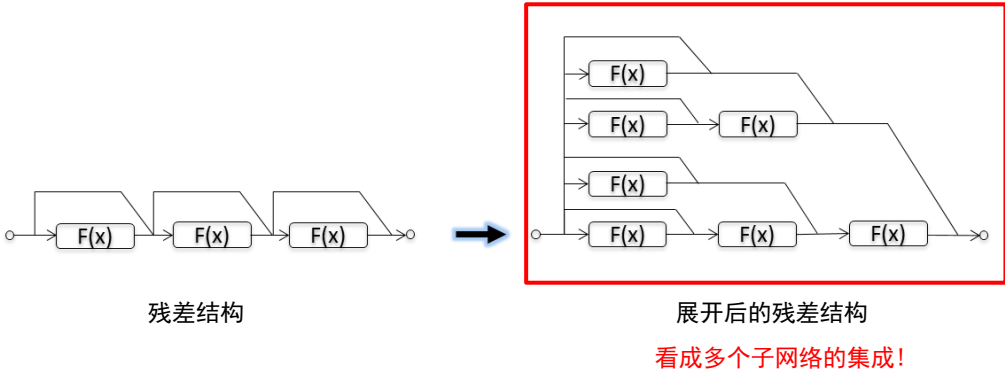
问题：为什么残差网络性能这么好？

一种典型的解释：残差网络可以看作是一种集成模型！



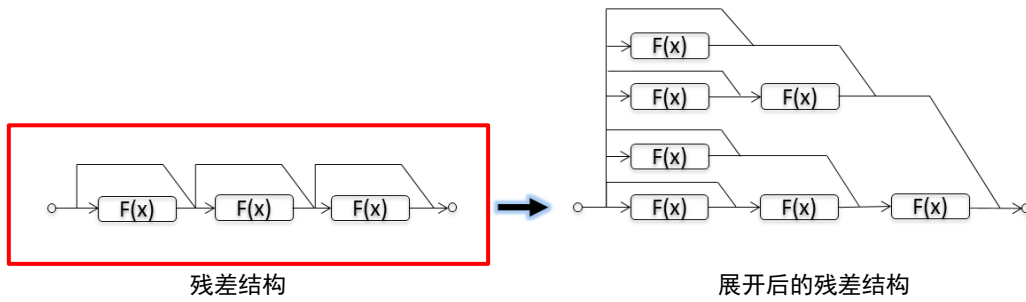
问题：为什么残差网络性能这么好？

一种典型的解释：残差网络可以看作是一种集成模型！



问题：为什么残差网络性能这么好？

一种典型的解释：残差网络可以看作是一种集成模型！



结论：残差网络是一种集成模型, 这是重要特点也是它高效的一个原因！

本章小结

本章小结

- 介绍了5种经典的卷积神经网络AlexNet、ZFNet、VGG、GoogLeNet和ResNet

2020/4/28

北京邮电大学计算机学院 鲁鹏

204

本章小结

- 介绍了5种经典的卷积神经网络AlexNet、ZFNet、VGG、GoogLeNet和ResNet
- 残差网络和Inception V4是公认的推广性能最好的两个分类模型

2020/4/28

北京邮电大学计算机学院 鲁鹏

205

本章小结

- 介绍了5种经典的卷积神经网络AlexNet、ZFNet、VGG、GoogLeNet和ResNet
- 残差网络和Inception V4是公认的推广性能最好的两个分类模型
- 特殊应用环境下的模型：面向有限存储资源的SqueezeNet以及面向有限计算资源的

MobileNet和ShuffleNet