

第九讲 卷积神经网络架构

2019年5月12日 10:04

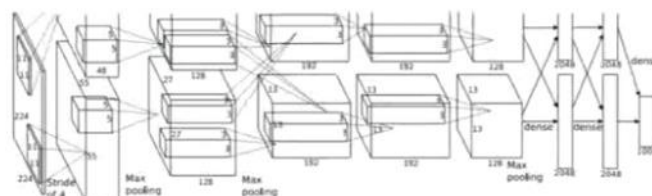
1. AlexNet

Case Study: AlexNet

[Krizhevsky et al. 2012]

Full (simplified) AlexNet architecture:

[227x227x3] INPUT
[55x55x96] **CONV1**: 96 11x11 filters at stride 4, pad 0
[27x27x96] **MAX POOL1**: 3x3 filters at stride 2
[27x27x96] **NORM1**: Normalization layer
[27x27x256] **CONV2**: 256 5x5 filters at stride 1, pad 2
[13x13x256] **MAX POOL2**: 3x3 filters at stride 2
[13x13x256] **NORM2**: Normalization layer
[13x13x384] **CONV3**: 384 3x3 filters at stride 1, pad 1
[13x13x384] **CONV4**: 384 3x3 filters at stride 1, pad 1
[13x13x256] **CONV5**: 256 3x3 filters at stride 1, pad 1
[6x6x256] **MAX POOL3**: 3x3 filters at stride 2
[4096] **FC6**: 4096 neurons
[4096] **FC7**: 4096 neurons

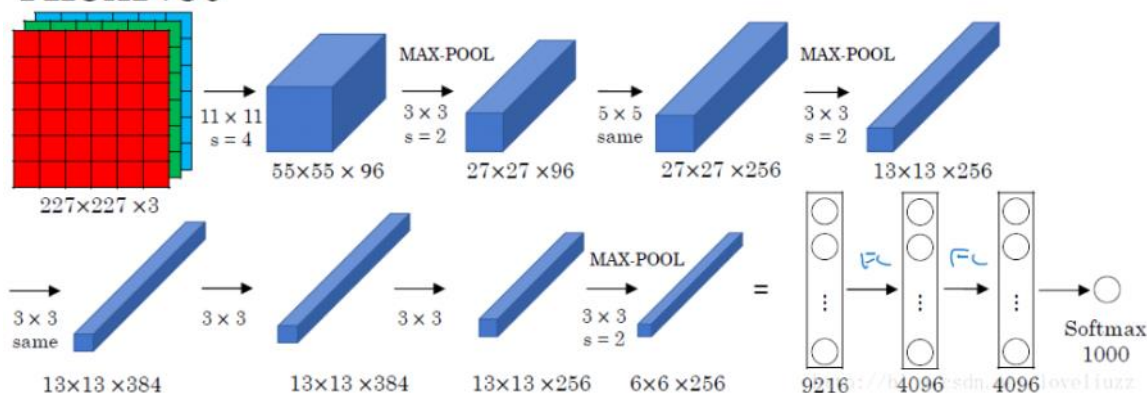


Details/Retrospectives:

- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

图9.1.1 Alexnet结构

AlexNet



alexnet数据前馈过程:

input: 227*227*3

Input-->Conv1: 卷积核11*11 Conv1 output: 55*55*96

: 卷积核数目96

: 步长4

Conv1-->Pool1: 池化大小3*3 Pool output: 27*27*96

: 池化步长2

...

...

...

FC层 --> Softmax

卷积操作和池化操作的输出计算公式见第五章。

2.VGGNet

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Small filters, Deeper networks

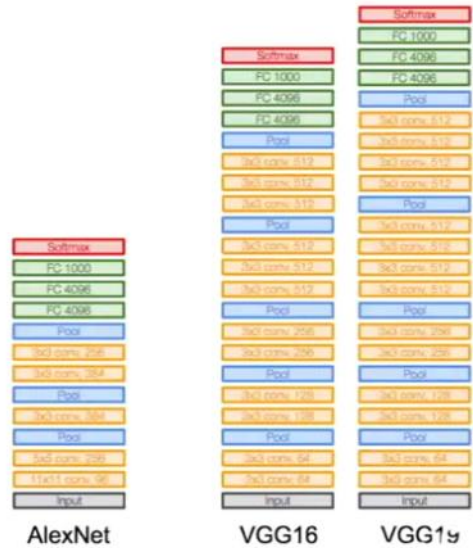
8 layers (AlexNet)

-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

11.7% top 5 error in ILSVRC'13
(ZNet)

-> 7.3% top 5 error in ILSVRC'14



INPUT: [224x224x3] memory: 224*224*3=150K params: 0 (not counting biases)
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K params: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K params: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K params: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K params: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K params: 0
FC: [1x1x4096] memory: 4096 params: 7*7*512*4096 = 102,760,448
FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000



图 9.1.2 VGG16 架构

从alex

3.GoogleNet

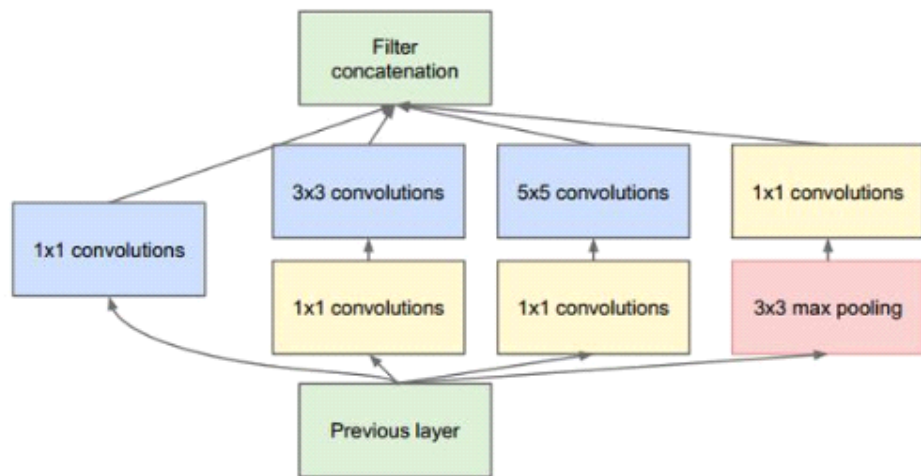


图9.1.3 GoogleNet瓶颈层

4.ResNet

	Alexnet	VGG	GoogleNet	ResNet
时间	2012	2014	2014	2015
层数	8	19	22	152
Top-5 ERR	16.4%	7.3%	6.7%	3.75%
Data Augmentation	+	+	+	+
Inception	-	-	+	-
卷积层数	5	16	21	151
卷积核大小	11,5,3	3	7,1,3,5	7,1,3,5
全连接层数	3 4096,4096,1000	3 4096,4096,1000	1 1000	1 1000
Dropout	+	+	+	+
LRN	+	-	+	-
Batch Normalization	-	-	-	+

- 1) Alexnet与Lecun1989年提出的初级深度网络在模型架构、训练策略上有何异同？为什么能获得识别精度如此大的飞跃？
- 2) VGGNet、GoogleNet、ResNet分别相比前者在模型架构、训练策略上有何改进？为什么能保证精度持续提升、同时参数个数和预测时间还下降了？