

智能之门

神经网络和深度学习入门

(基于Python的实现)

STEP 8 卷积神经网络

第 18 章

卷积神经网络应用

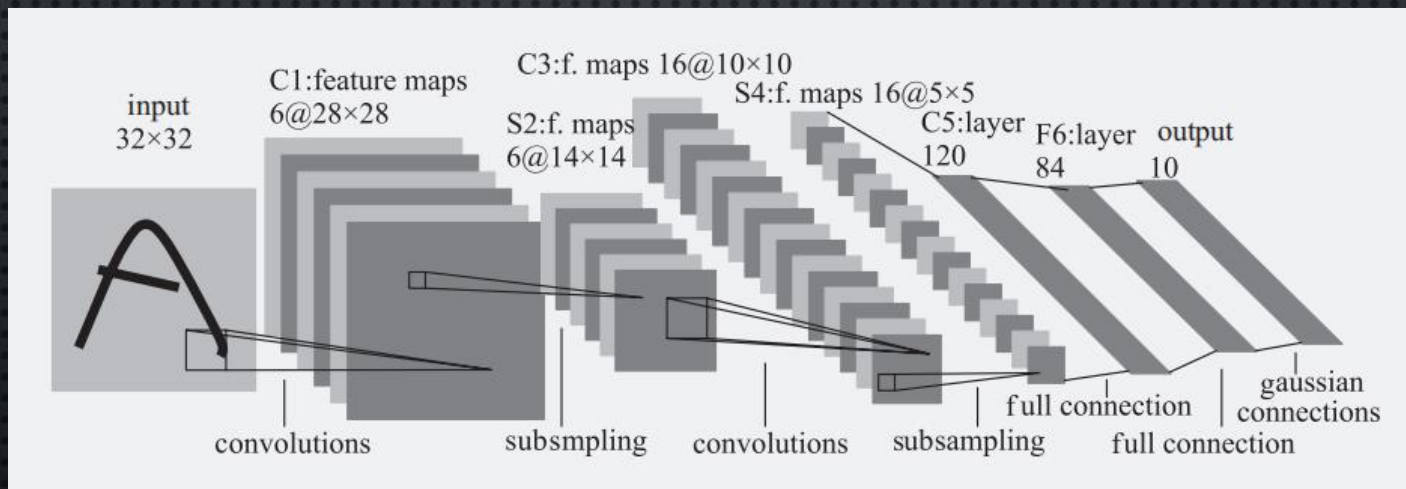
- 18.1 经典模型
- 18.2 颜色与几何图形分类
- 18.3 MNIST分类
- 18.4 Fashion-MNIST分类
- 18.5 Cifar-10分类

本部分我们会介绍一些经典的卷积模型，并展示一些分类实例，向大师们学习一些方法论问题。

18.1 经典模型

➤ LeNet (1998)

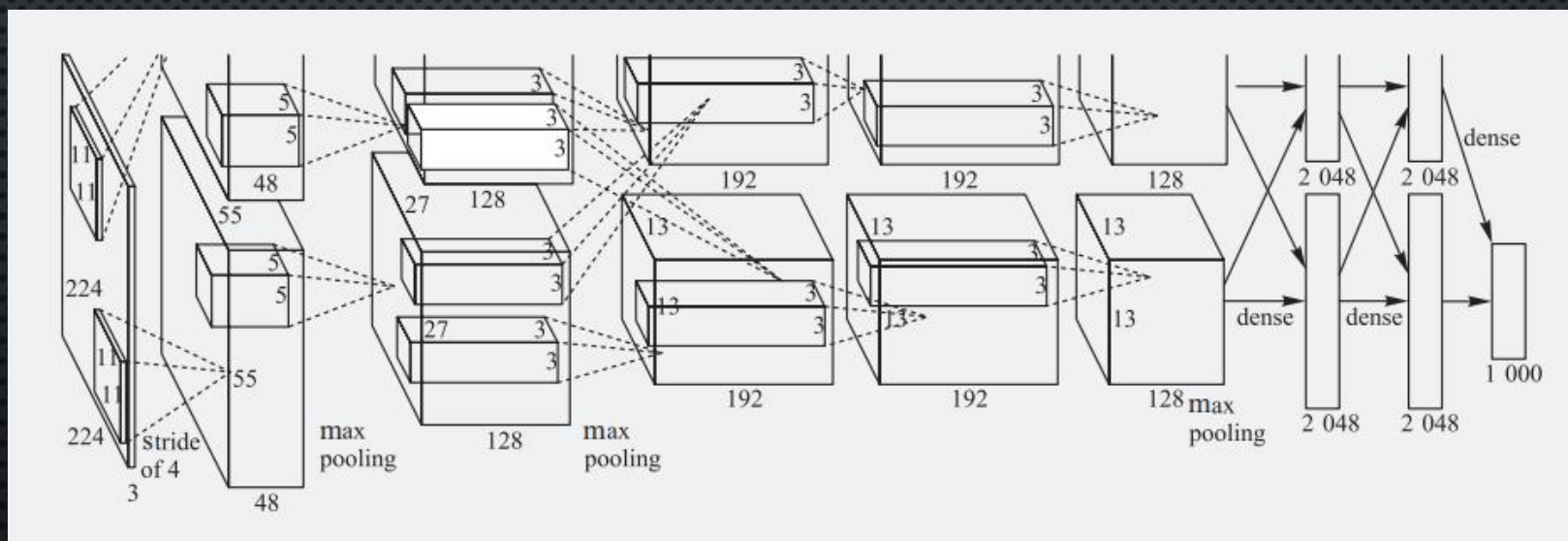
- 由卷积神经网络的开创者LeCun在1998年提出，用于解决手写数字识别的视觉任务。自那时起，卷积神经网络的最基本的架构就定下来了：卷积层、池化层、全连接层。
- 如今各大深度学习框架中所使用的LeNet都是简化改进过的LeNet-5（5表示具有5个层），和原始的LeNet有些许不同，比如把激活函数改为了现在很常用的ReLU。



18.1 经典模型

➤ AlexNet (2012)

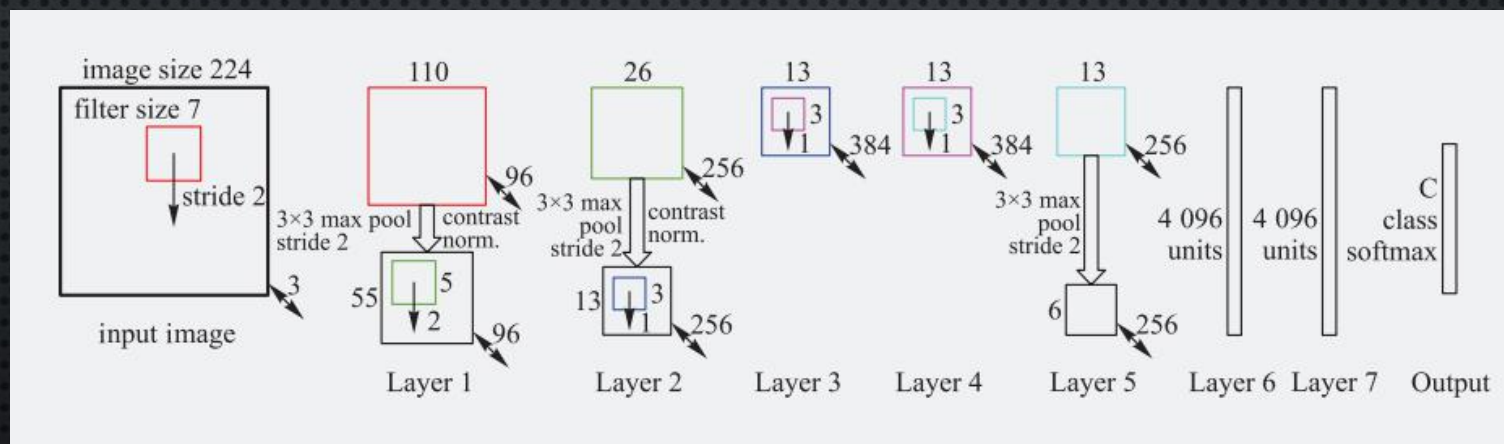
- AlexNet网络结构在整体上类似于LeNet，都是先卷积然后在全连接。但在细节上有很大不同。AlexNet有60million个参数和65000个神经元，五层卷积，三层全连接网络，最终的输出层是1000通道的Softmax。AlexNet用两块GPU并行计算，大大提高了训练效率。



18.1 经典模型

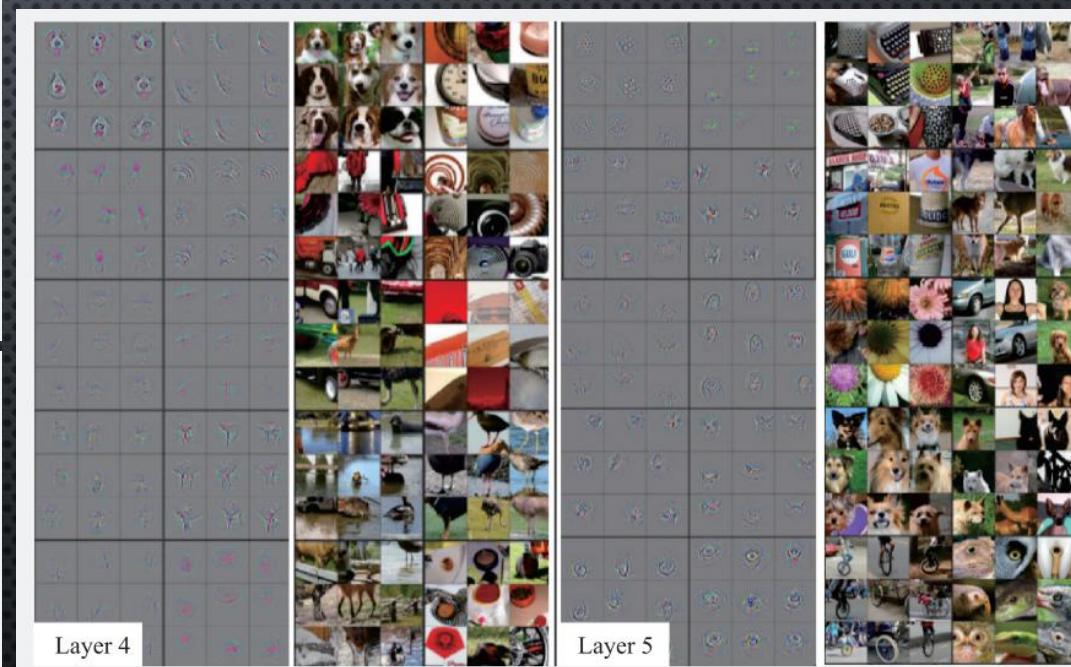
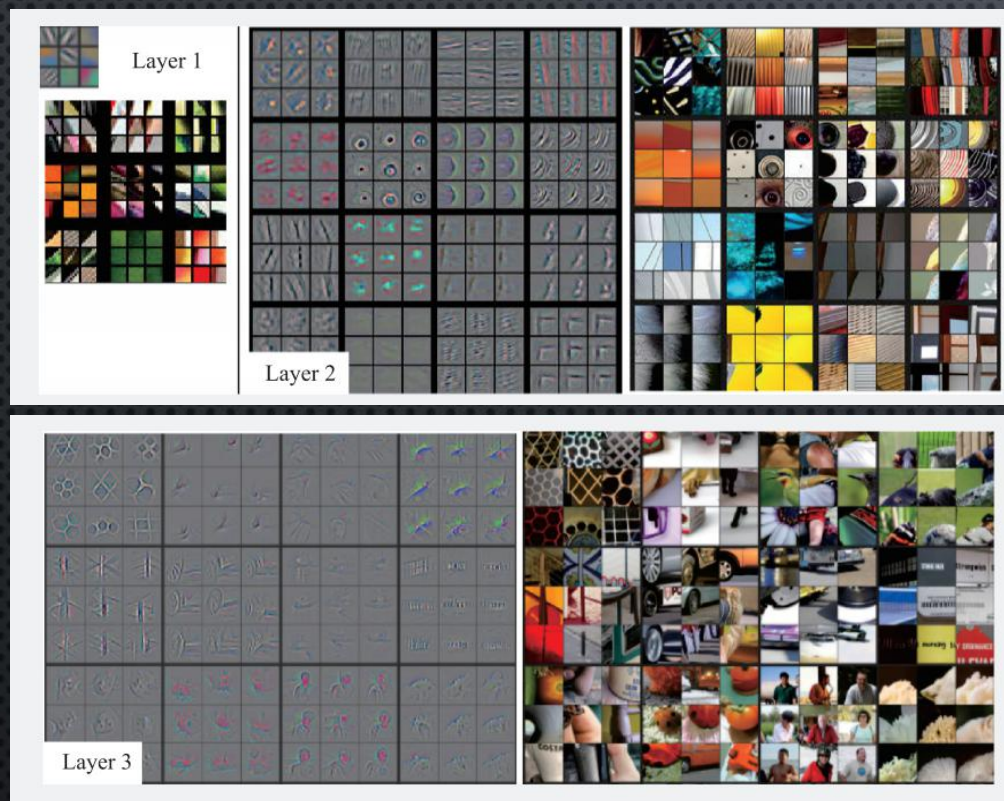
➤ ZFNet (2013)

- ZFNet首次系统化地对卷积神经网络做了可视化的研究，从而找到了AlexNet的缺点并加以改正，提高了网络的能力。总的来说，通过卷积神经网络学习后，我们学习到的特征，是具有辨别性的特征，比如要我们区分人脸和狗头，那么通过卷积神经网络学习后，背景部位的激活度基本很少，我们通过可视化就可以看到我们提取到的特征忽视了背景，而是把关键的信息给提取出来了。



18.1 经典模型

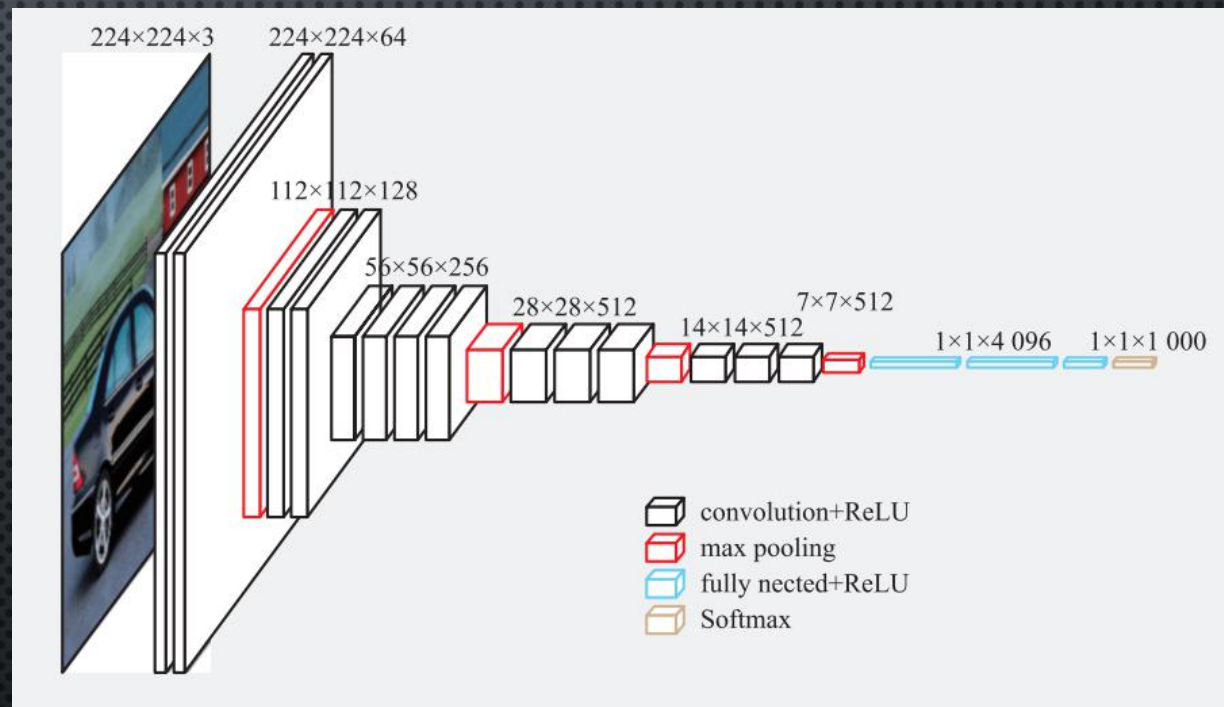
- ZFNet提取的特征



18.1 经典模型

➤ VGGNet (2015)

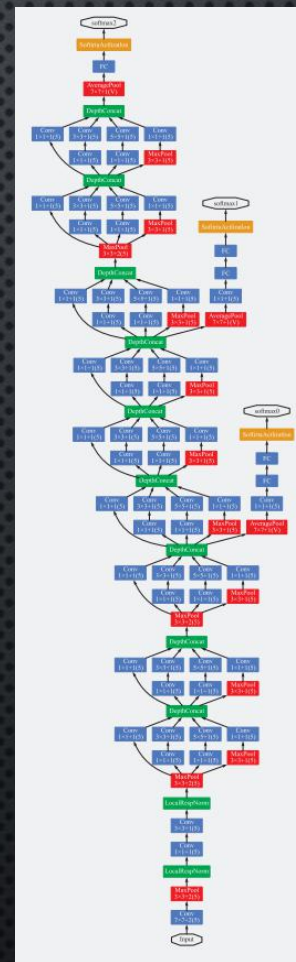
- VGGNet是由牛津大学的视觉几何组和Google DeepMind公司的研究员一起研发的深度卷积神经网络。
- VGGNet的卷积层有一个特点：特征图的空间分辨率单调递减，特征图的通道数单调递增，使得输入图像在维度上流畅地转换到分类向量。



18.1 经典模型

➤ GoogleNet (2014)

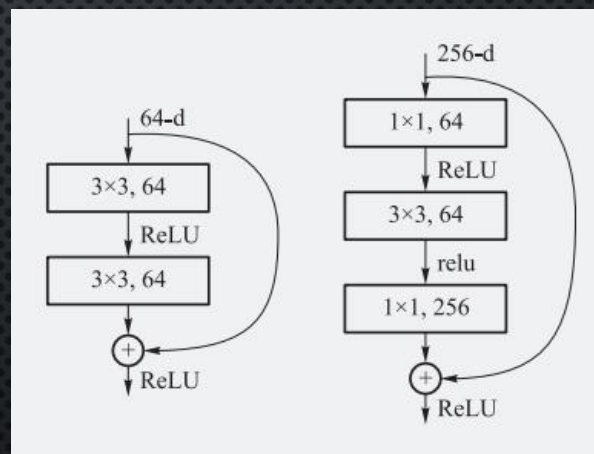
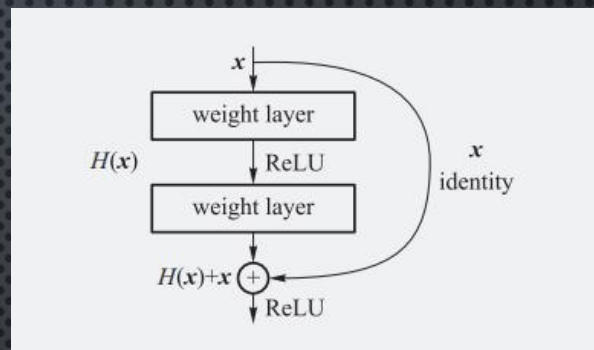
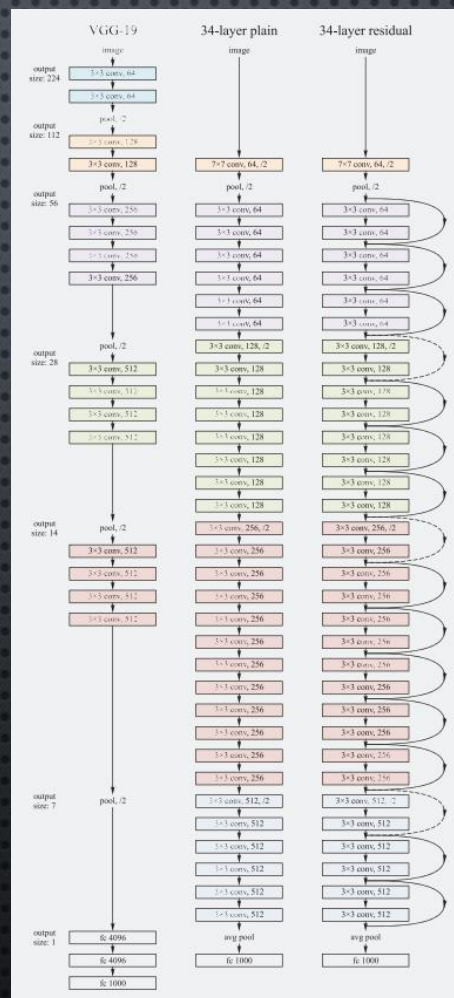
- GoogLeNet跟AlexNet,VGG-Nets这种单纯依靠加深网络结构进而改进网络性能的思路不一样，它另辟幽径，在加深网络的同时（22层），也在网络结构上做了创新，引入Inception结构代替了单纯的卷积+激活的传统操作。GoogLeNet进一步把对卷积神经网络的研究推上新的高度。
- GoogLeNet网络结构中有3个LOSS单元，这样的网络设计是为了帮助网络的收敛。在中间层加入辅助计算的LOSS单元，目的是计算损失时让低层的特征也有很好的区分能力，从而让网络更好地被训练。
- GoogLeNet还有一个闪光点值得一提，那就是将后面的全连接层全部替换为简单的全局平均pooling，在最后参数会变的更少。使用大网络在宽度和深度允许GoogleNet移除全连接层，但并不会影响到结果的精度，在ImageNet中实现93.3%的精度，而且要比VGG还要快。



18.1 经典模型

➤ ResNet (2015)

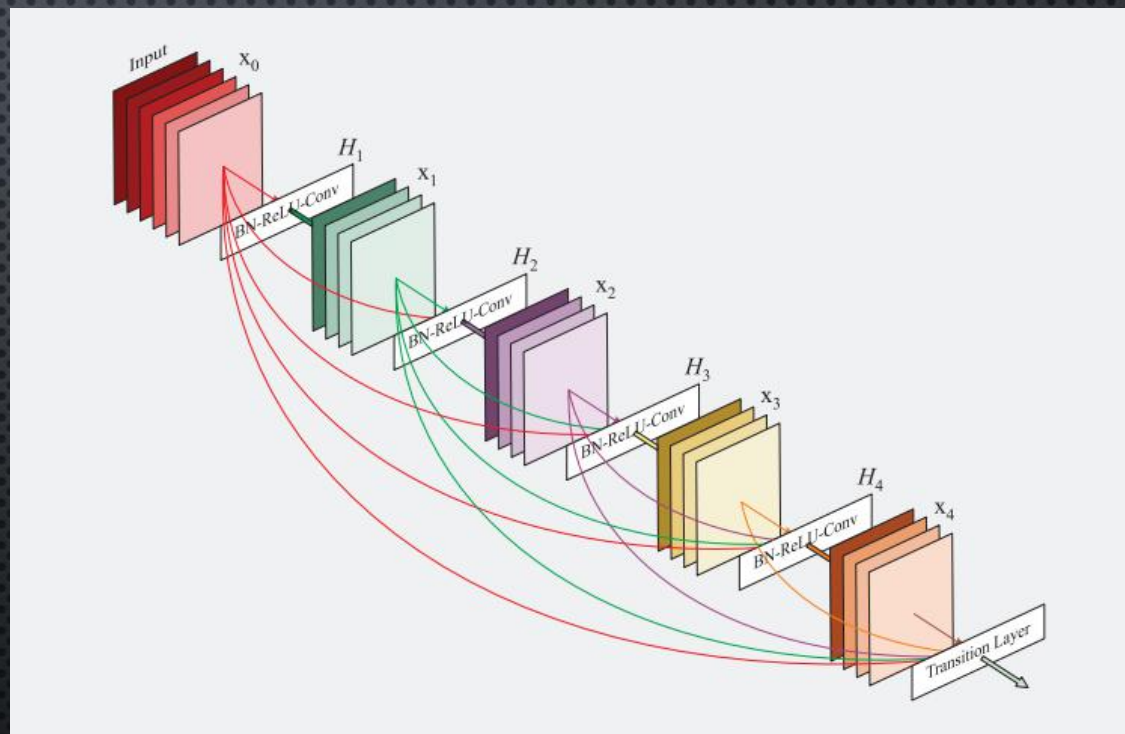
- ResNet称为残差网络。若将输入设为 X ，将某一有参网络层设为 H ，那么以 X 为输入的此层的输出将为 $H(X)$ 。一般的卷积神经网络网络如Alexnet/VGG等会直接通过训练学习出参数函数 H 的表达，从而直接学习 $X \rightarrow H(X)$ 。
- 而残差学习则是致力于使用多个有参网络层来学习输入、输出之间的参差即 $H(X) - X$ 即学习 $X \rightarrow (H(X) - X) + X$ 。其中 X 这一部分为直接的identity mapping，而 $H(X) - X$ 则为有参网络层要学习的输入输出间残差。



18.1 经典模型

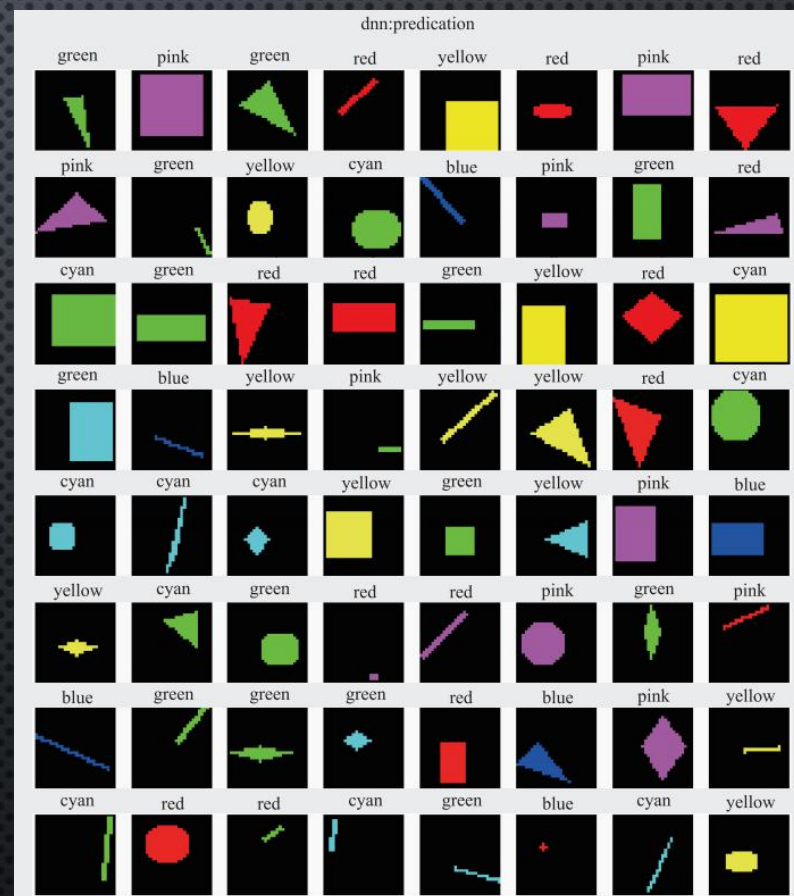
➤ DenseNet (2017)

- DenseNet是一种具有密集连接的卷积神经网络。在该网络中，任何两层之间都有直接的连接，也就是说，网络每一层的输入都是前面所有层输出的并集，而该层所学习的特征图也会被直接传给其后面所有层作为输入。
 - ✓ 相比ResNet拥有更少的参数数量
 - ✓ 旁路加强了特征的重用
 - ✓ 网络更易于训练,并具有一定的正则效果
 - ✓ 缓解了gradient vanishing和model degradation的问题



18.2 颜色与几何图形分类

- 前馈神经网络运行结果：对两类形状上的颜色判断不准：很细的线和很大的色块。
 - ✓ 针对细直线，由于带颜色的像素点的数量非常少，被拆成向量后，这些像素点就会在 1×784 的矢量中彼此相距很远，特征不明显，很容易被判别成噪音。
 - ✓ 针对大色块，由于带颜色的像素点的数量非常多，即使被拆成向量，也会占据很大的部分，这样特征点与背景点的比例失衡，导致无法判断出到底哪个是特征点。



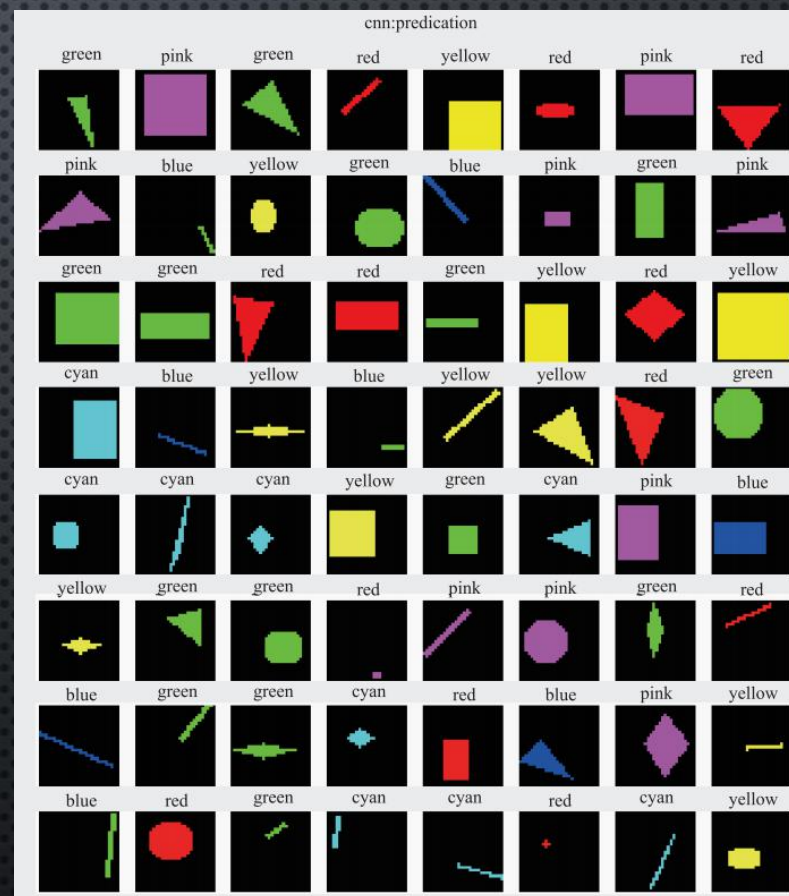
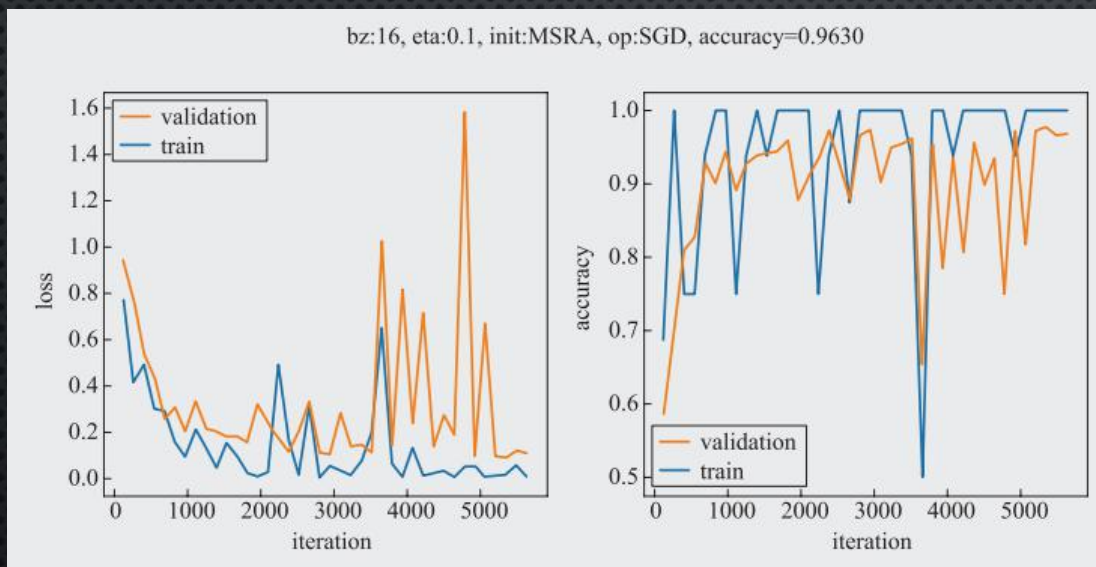
18.2 颜色与几何图形分类

- 卷积神经网络模型

ID	类型	参数	输入尺寸	输出尺寸
1	卷积	2x1x1,S=1	3x28x28	2x28x28
2	激活	Relu	2x28x28	2x28x28
3	池化	2x2,S=2,Max	2x14x14	2x14x14
4	卷积	3x3x3,S=1	2x14x14	3x12x12
5	激活	Relu	3x12x12	3x12x12
6	池化	2x2,S=2,Max	3x12x12	3x6x6
7	全连接	32	108	32
8	归一化		32	32
9	激活	Relu	32	32
10	全连接	6	32	6
11	分类	Softmax	6	6

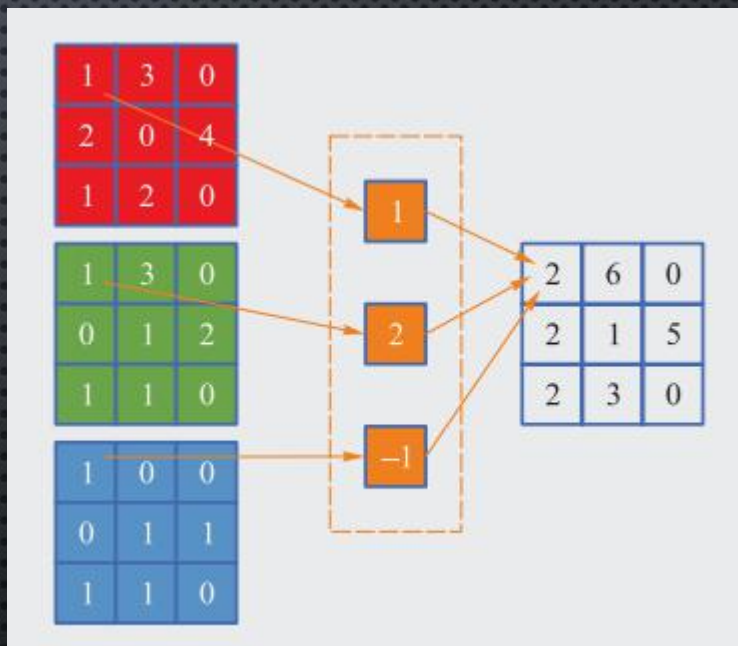
18.2 颜色与几何图形分类

- 卷积神经网络训练曲线和运行结果
 - ✓ 在测试集上得到了 96.3% 的准确度，比前馈神经网络模型要高出很多，这也证明了卷积神经网络在图像识别上的能力。



18.2 颜色与几何图形分类

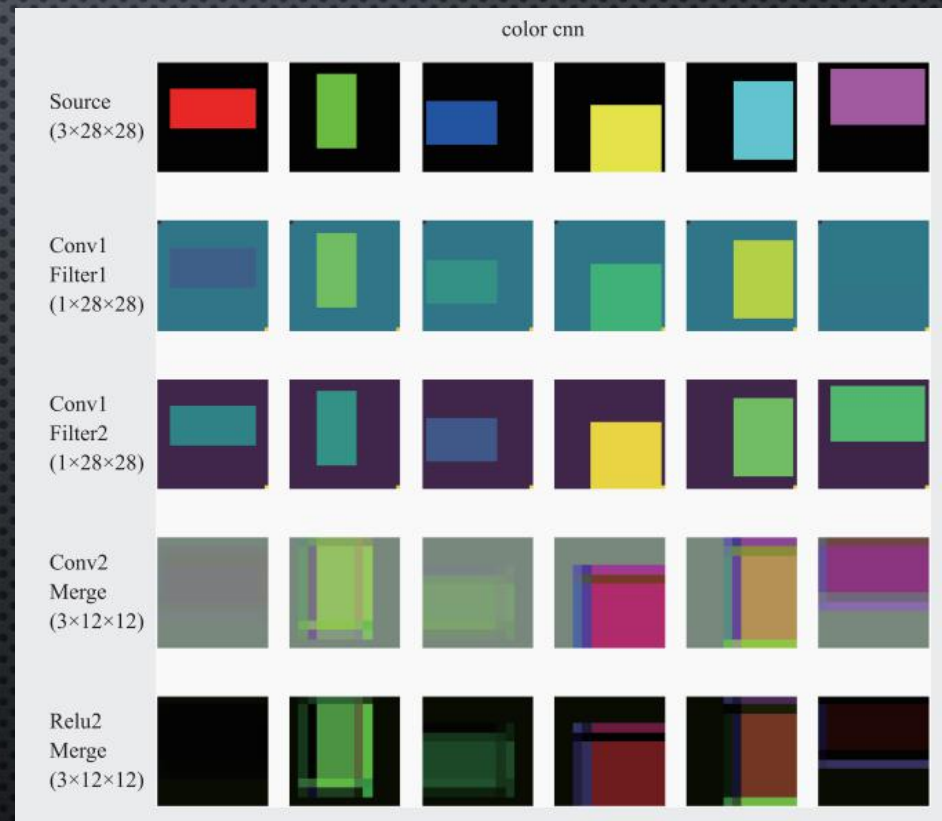
- 1×1 卷积核
 - ✓ 跨通道信息整合。
 - ✓ 降维以减少学习参数。
 - ✓ 关注的是不同通道的相同位置的像素之间的相关性，而不是同一通道内的像素的相关性。



18.2 颜色与几何图形分类

- 可视化

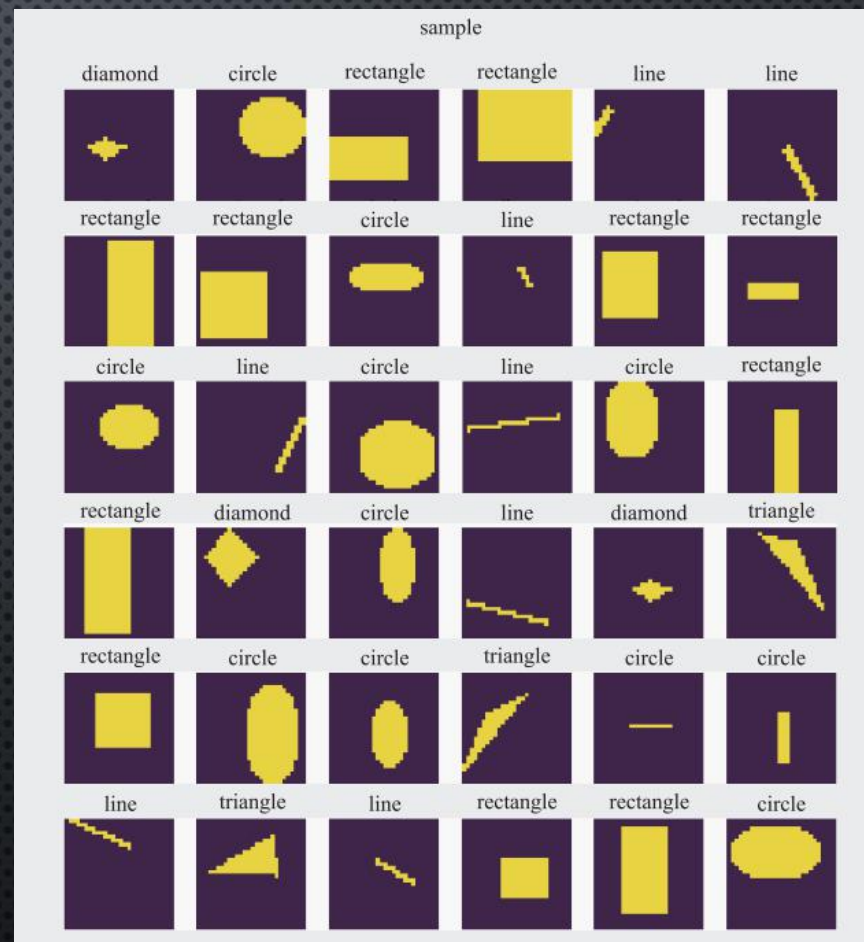
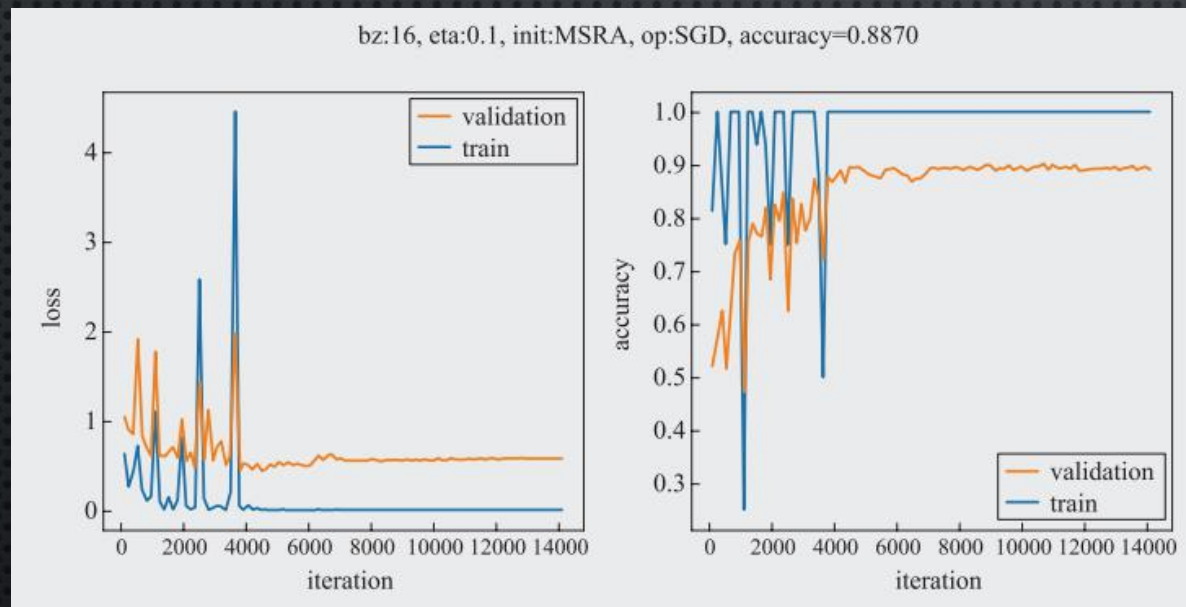
- ✓ 第一行是原始彩色图片，三通道 28×28 ，特意挑出来都是矩形的6种颜色。
- ✓ 第二行是第一卷积组合梯队的第1个 1×1 的卷积核在原始图片上的卷积结果。
- ✓ 第三行是第一卷积组合梯队的第2个 1×1 的卷积核在原始图片上的卷积结果。
- ✓ 第四行是第二卷积组合梯队的三个卷积核的卷积结果图，把三个特征图当作RGB通道后所生成的彩色图。
- ✓ 第五行是第二卷积组合梯队的激活函数结果，和原始图片相差很大。



18.2 颜色与几何图形分类

➤ 几何形状分类

- 前馈神经网络的训练曲线
 - ✓ 在测试集上得到的的准确度是 89.8%。



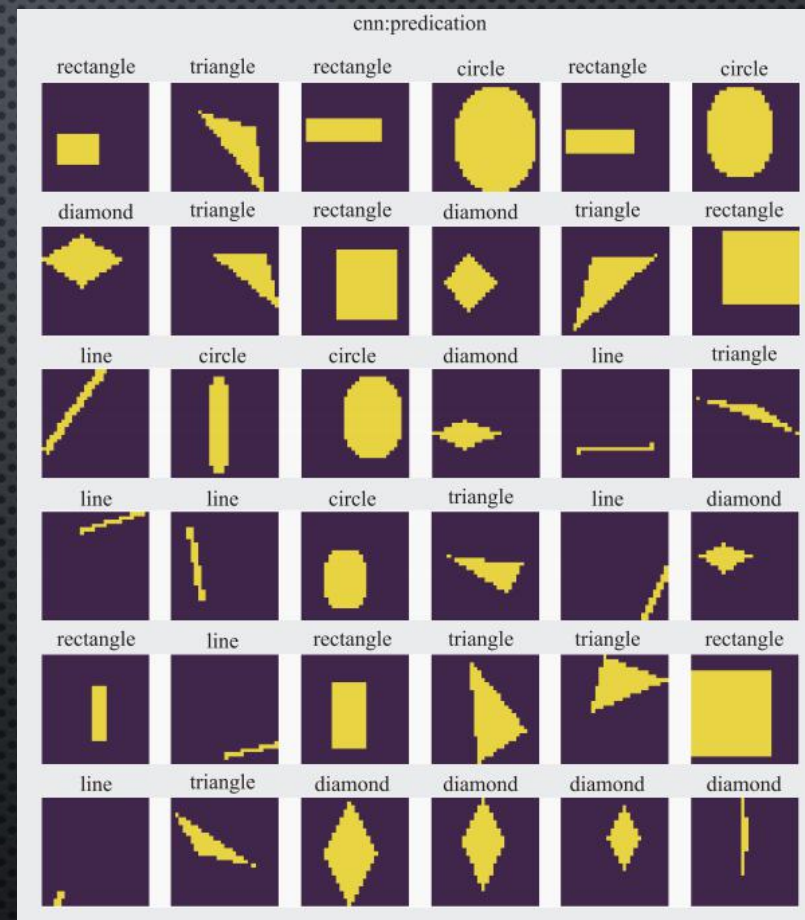
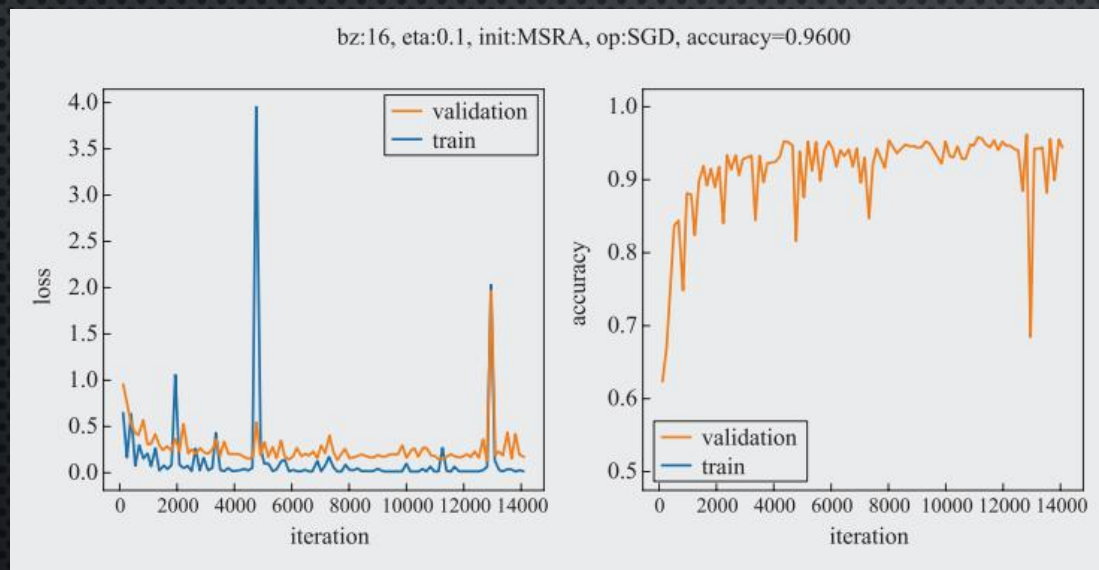
18.2 颜色与几何图形分类

- 卷积神经网络模型

ID	类型	参数	输入尺寸	输出尺寸
1	卷积	$8 \times 3 \times 3, S=1, P=1$	$1 \times 28 \times 28$	$8 \times 28 \times 28$
2	激活	Relu	$8 \times 28 \times 28$	$8 \times 28 \times 28$
3	池化	$2 \times 2, S=2, \text{Max}$	$8 \times 28 \times 28$	$8 \times 14 \times 14$
4	卷积	$16 \times 3 \times 3, S=1$	$8 \times 14 \times 14$	$16 \times 12 \times 12$
5	激活	Relu	$16 \times 12 \times 12$	$16 \times 12 \times 12$
6	池化	$2 \times 2, S=2, \text{Max}$	$16 \times 6 \times 6$	$16 \times 6 \times 6$
7	全连接	32	576	32
8	归一化		32	32
9	激活	ReLU	32	32
10	全连接	5	32	5
11	分类	Softmax	5	5

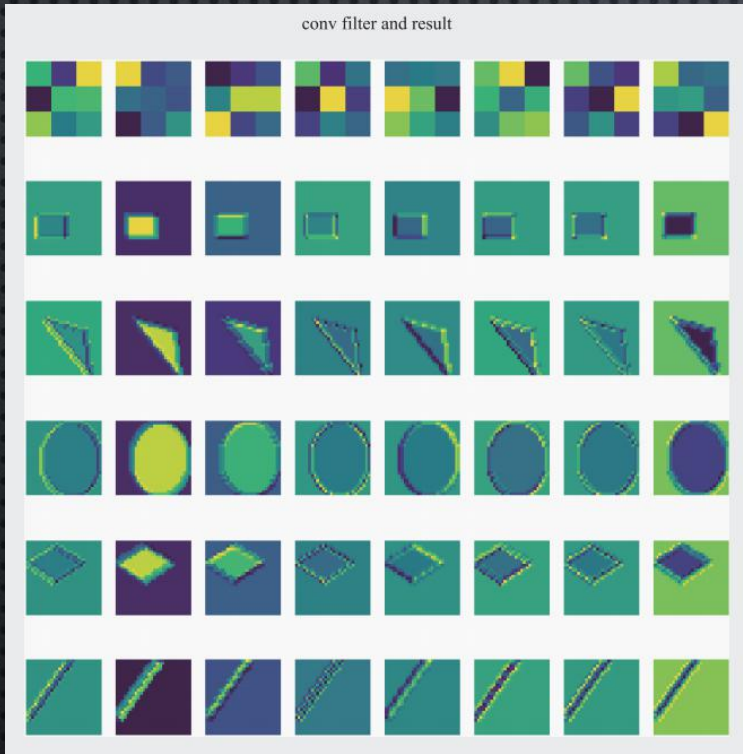
18.2 颜色与几何图形分类

- 卷积神经网络训练曲线和运行结果
 - 在测试集上得到了 96% 的准确度，比前馈神经网络模型要高出很多，这也证明了卷积神经网络在图像识别上的能力。



18.2 颜色与几何图形分类

- 可视化



- 卷积核的作用

卷积核序号	作用	直线	三角形	菱形	矩形	圆形
1	左侧边缘	0	1	0	1	1
2	大色块区域	0	1	1	1	1
3	左上侧边缘	0	1	1	0	1
4	45度短边	1	1	1	0	1
5	右侧边缘、上横边	0	0	0	1	1
6	左上、右上、右下	0	1	1	0	1
7	左边框和右下角	0	0	0	1	1
8	左上和右下，及背景	0	0	1	0	1

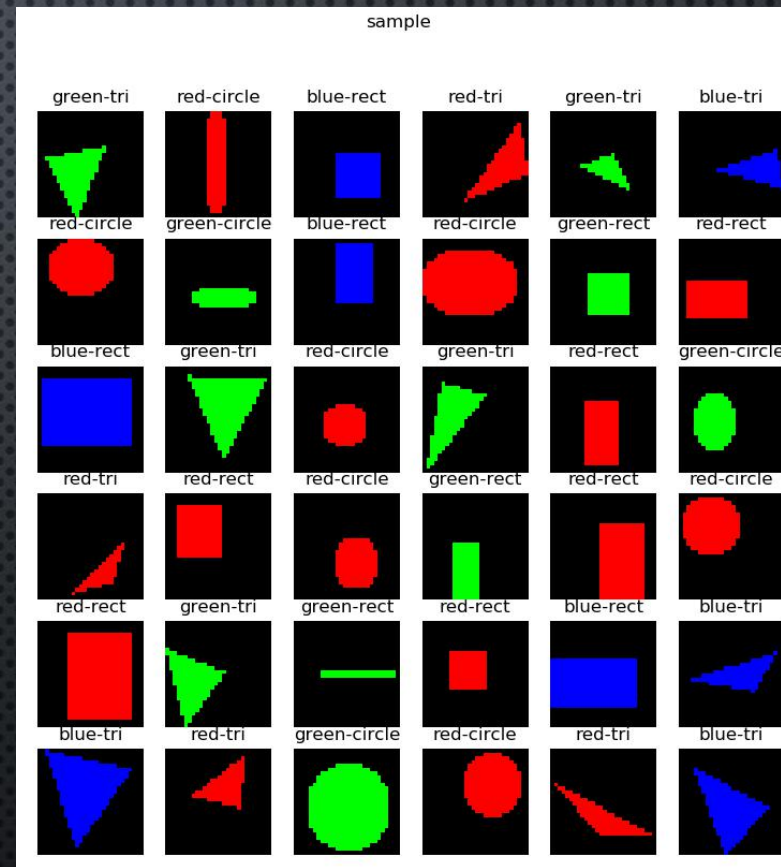
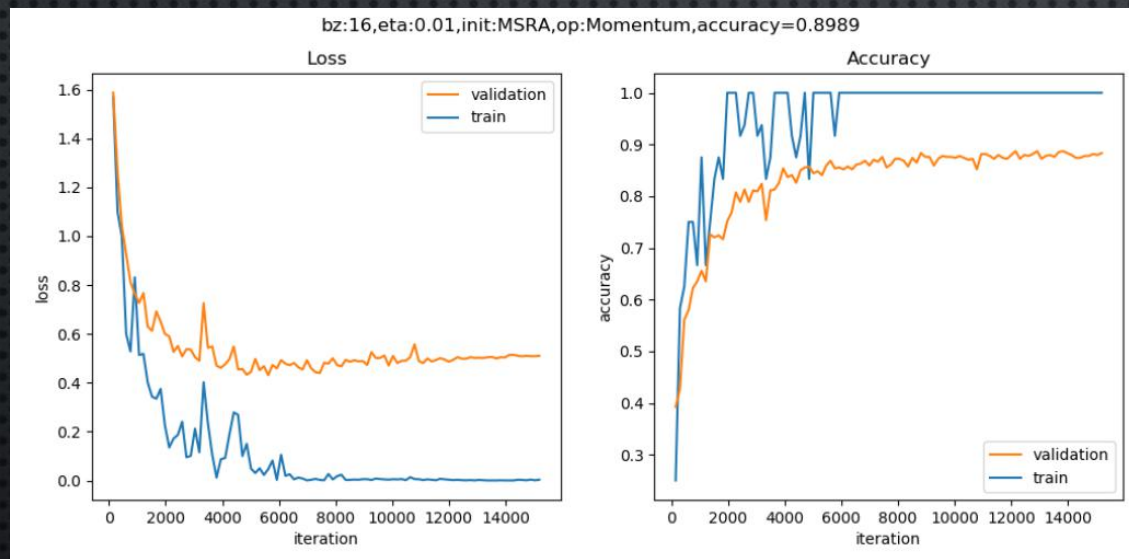
当然，神经网络可能不是按照我们分析的顺序来判定形状的，可以有很多其他种路径的组合，但最终总能够把5种形状分离开。

18.2 颜色与几何图形分类

➤ 同时分类

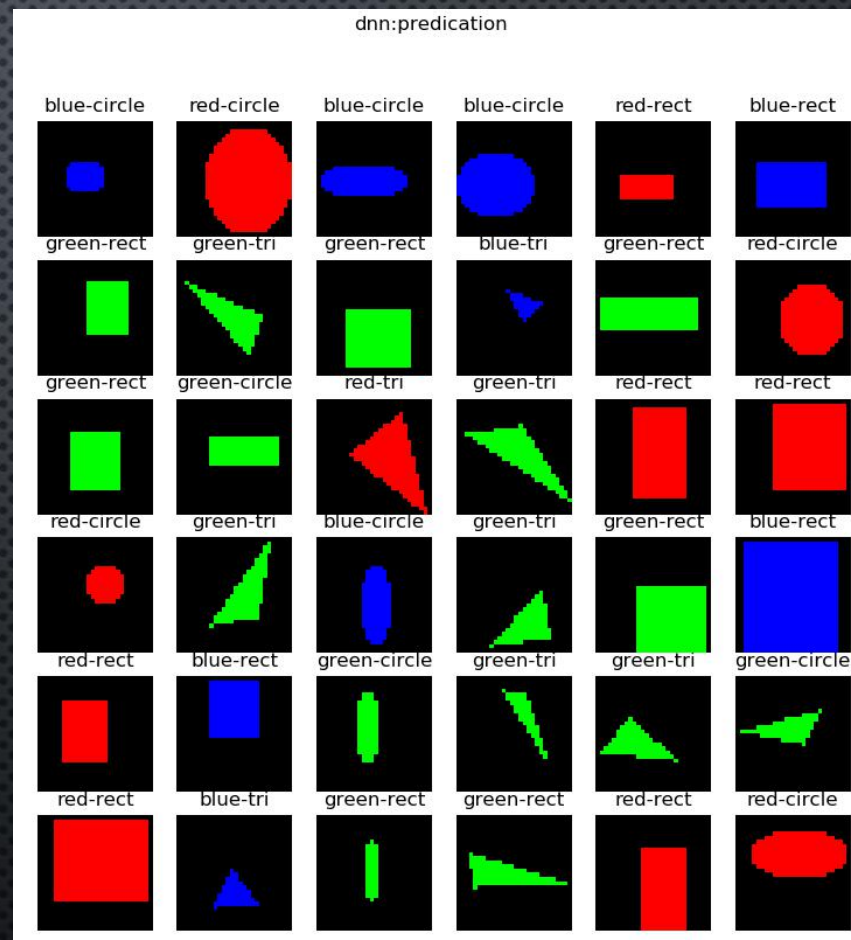
- 前馈神经网络的训练曲线
 - 在测试集上得到的的准确度是 89%。

	红色	蓝色	绿色
圆形	600:100	600:100	600:100
矩形	600:100	600:100	600:100
三角形	600:100	600:100	600:100



18.2 颜色与几何图形分类

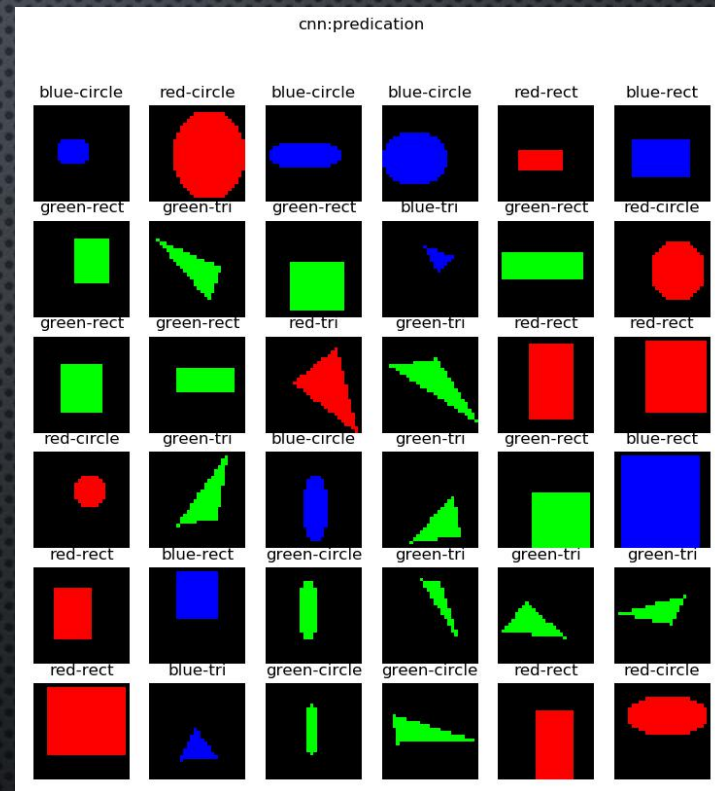
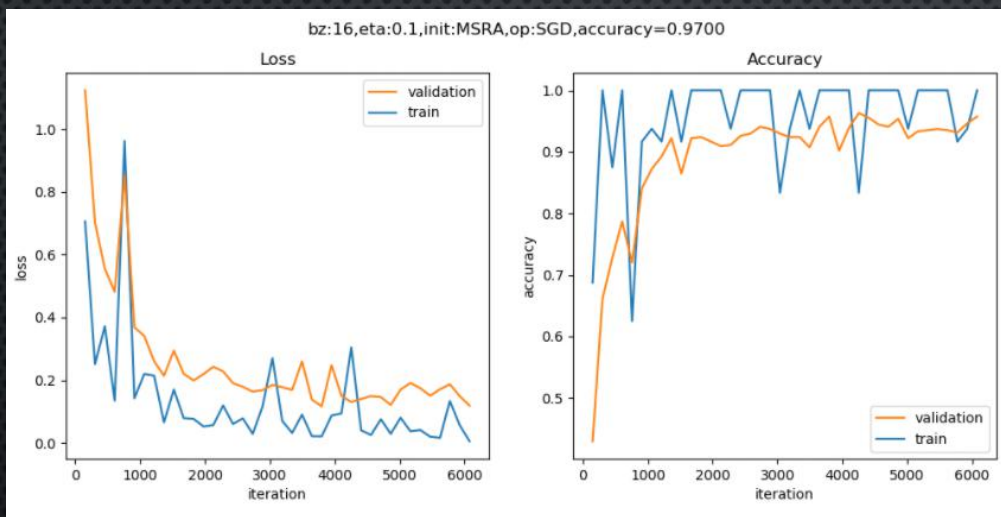
- 前馈神经网络运行结果
 - ✓ 绝大部分样本预测是正确的，但是第3行第2列的样本，应该是green-rect，被预测成green-circle；最后两行的两个green-tri也被预测错了形状，颜色并没有错。



18.2 颜色与几何图形分类

- 卷积神经网络训练曲线和运行结果

- ✓ 在测试集上得到了97%的准确度，比DNN模型要高出很多，这也证明了卷积神经网络在图像识别上的能力。
- ✓ 绝大部分样本预测是正确的，只有最后一行第4个样本，本来是green-triangle，被预测成green-circle。



18.3 MNIST分类

➤ MNIST分类

- 搭建模型
- 模型各层参数

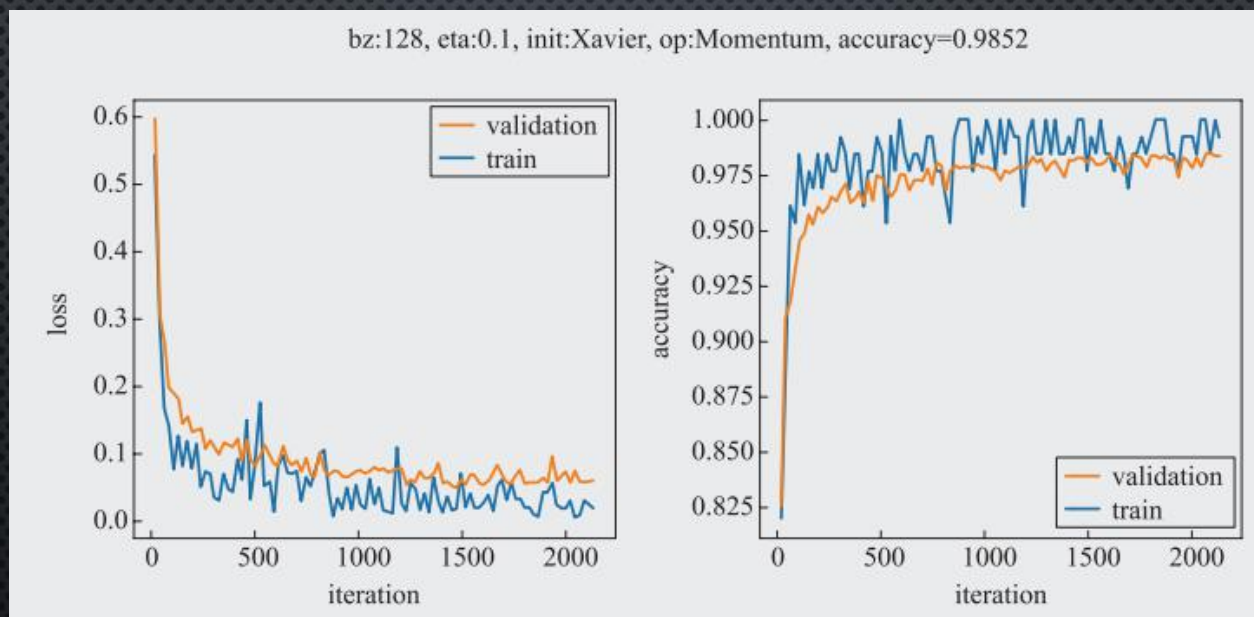


Layer	参数	输入	输出	参数个数
卷积层	$8 \times 5 \times 5, s=1$	$1 \times 28 \times 28$	$8 \times 24 \times 24$	$200+8$
激活层	$2 \times 2, s=2, \max$	$8 \times 24 \times 24$	$8 \times 24 \times 24$	
池化层	Relu	$8 \times 24 \times 24$	$8 \times 12 \times 12$	
卷积层	$16 \times 5 \times 5, s=1$	$8 \times 12 \times 12$	$16 \times 8 \times 8$	$400+16$
激活层	Relu	$16 \times 8 \times 8$	$16 \times 8 \times 8$	
池化层	$2 \times 2, s=2, \max$	$16 \times 8 \times 8$	$16 \times 4 \times 4$	
全连接层	256×32	256	32	$8192+32$
批归一化层		32	32	
激活层	Relu	32	32	
全连接层	32×10	32	10	$320+10$
分类层	softmax, 10	10	10	

18.3 MNIST分类

- 运行结果

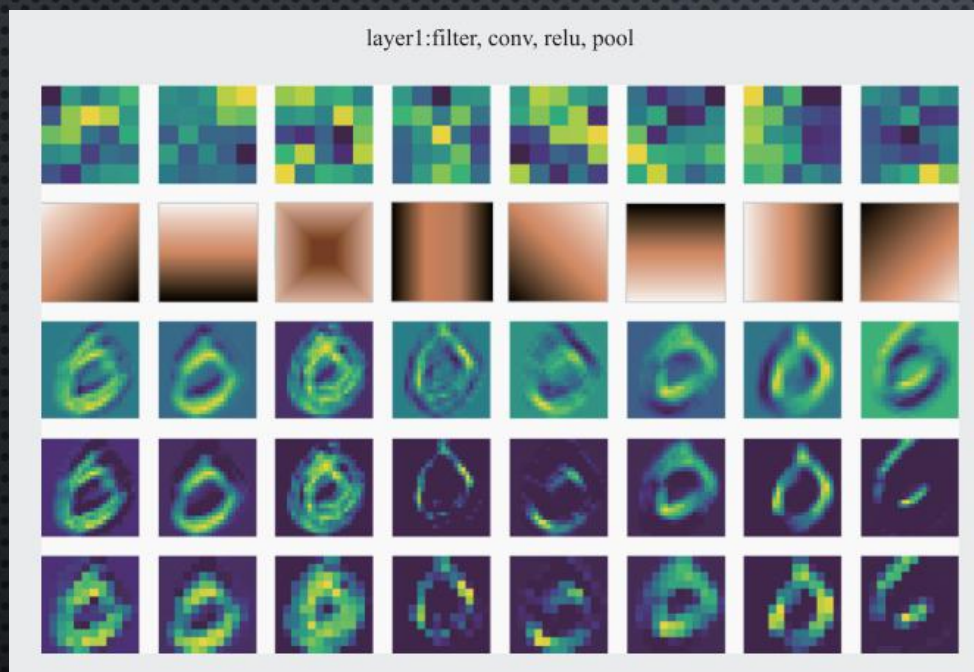
- ✓ 最后可以得到 98.44% 的准确率，比全连接网络要高 1 个百分点。如果想进一步提高准确率，可以尝试增加卷积层的能力，比如使用更多的卷积核来提取更多的特征。



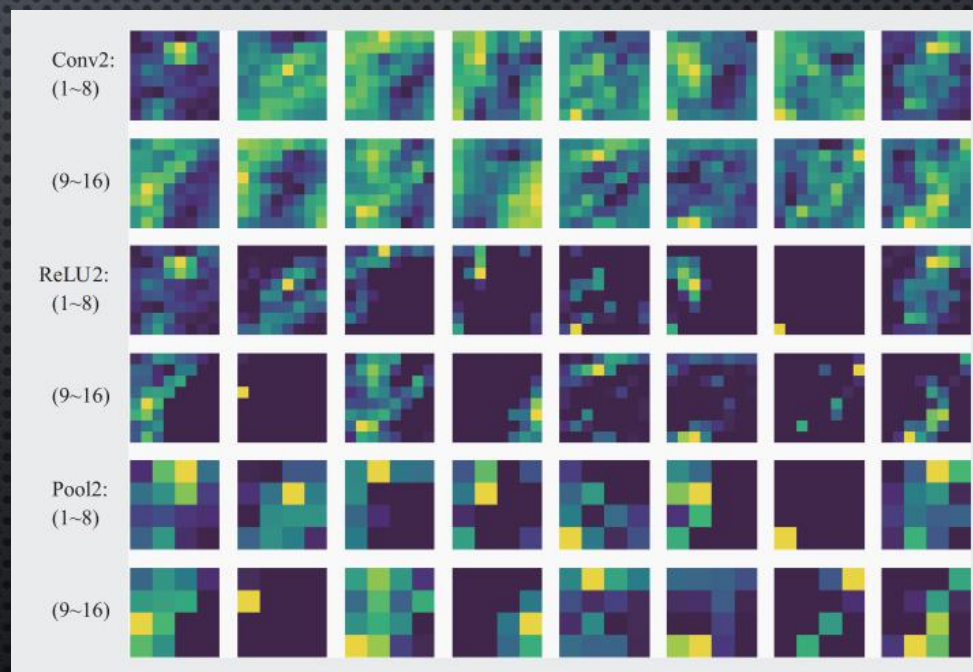
18.3 MNIST分类

- 卷积核可视化

- ✓ 第一组卷积核



- ✓ 第二组卷积核

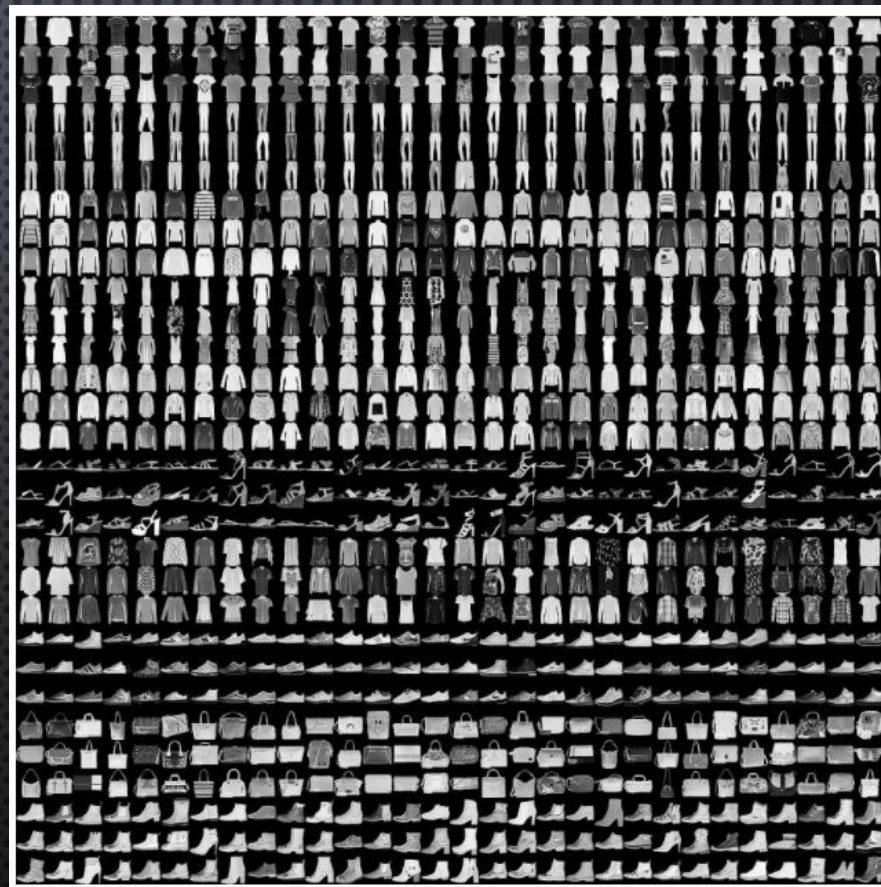


18.4 Fashion-MNIST分类

➤ Fashion-MNIST分类

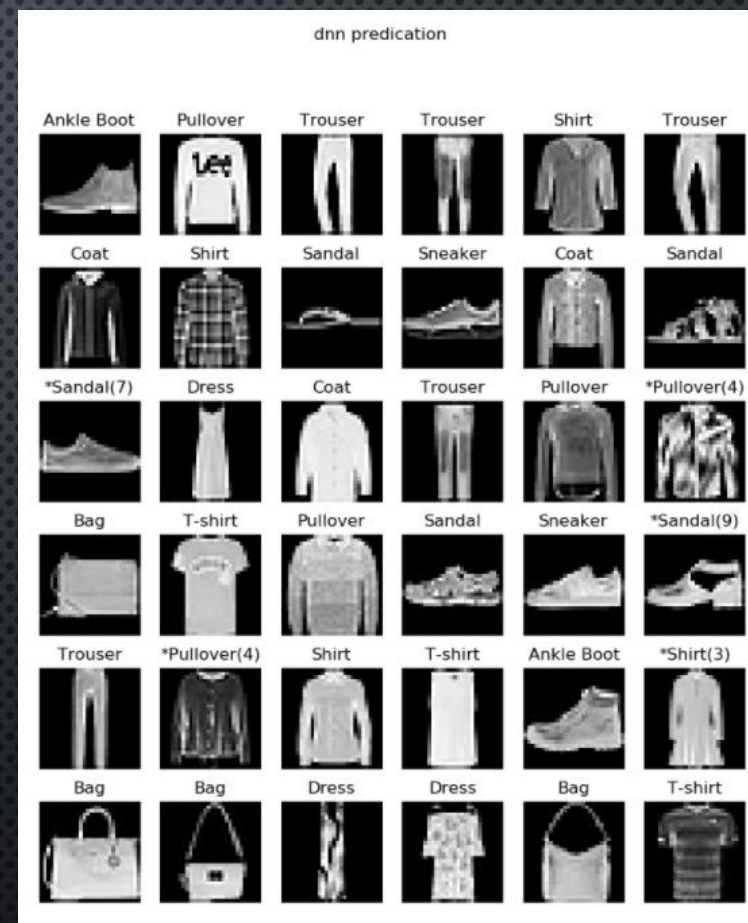
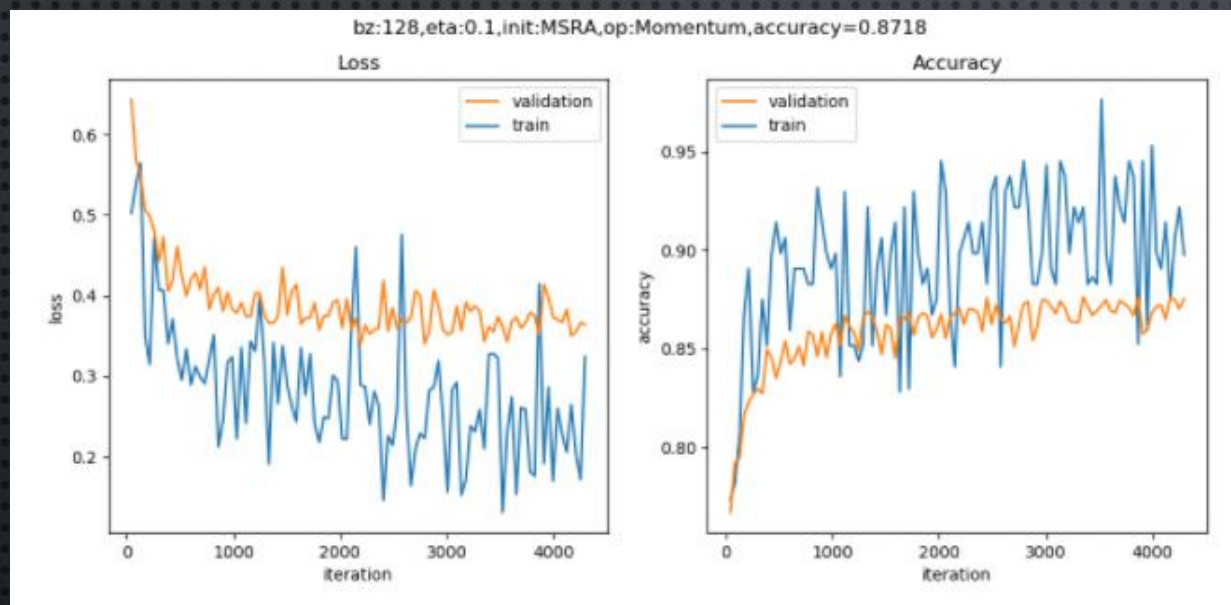
- 每3行是一类样本，按样本类别分行显示：

- ✓ T-Shirt, T恤衫 (1-3行)
- ✓ Trouser, 裤子 (4-6行)
- ✓ Pullover, 套头衫 (7-9行)
- ✓ Dress, 连衣裙 (10-12行)
- ✓ Coat, 外套 (13-15行)
- ✓ Sandal, 凉鞋 (16-18行)
- ✓ Shirt, 衬衫 (19-21行)
- ✓ Sneaker, 运动鞋 (22-24行)
- ✓ Bag, 包 (25-27行)
- ✓ Ankle Boot, 短靴 (28-30行)



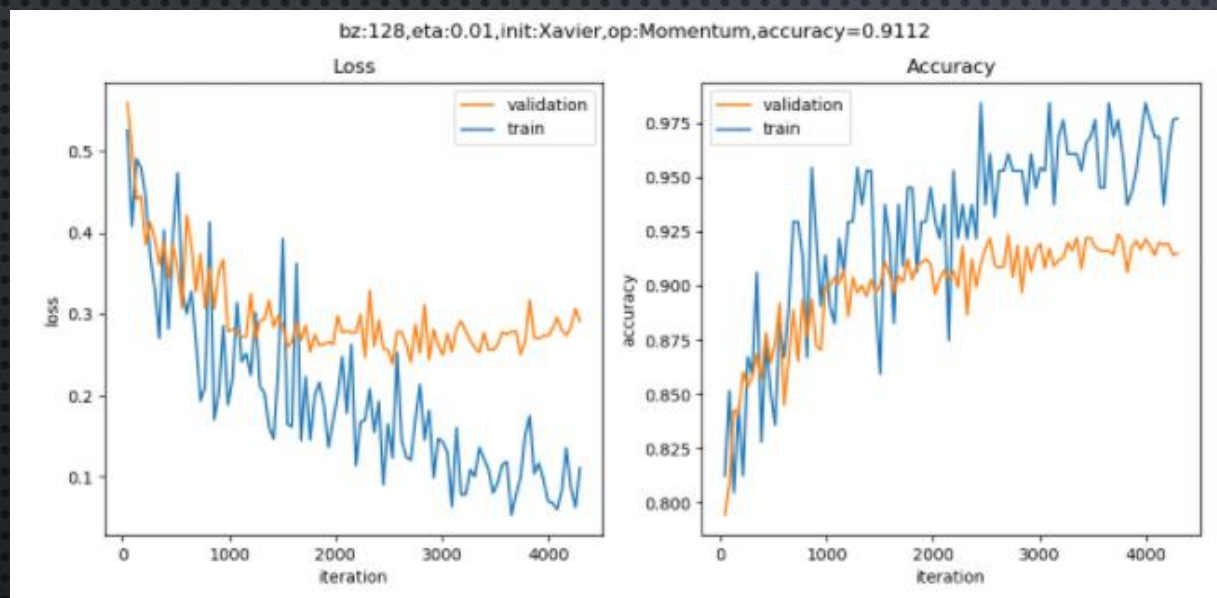
18.4 Fashion-MNIST分类

- 前馈神经网络运行结果



18.4 Fashion-MNIST分类

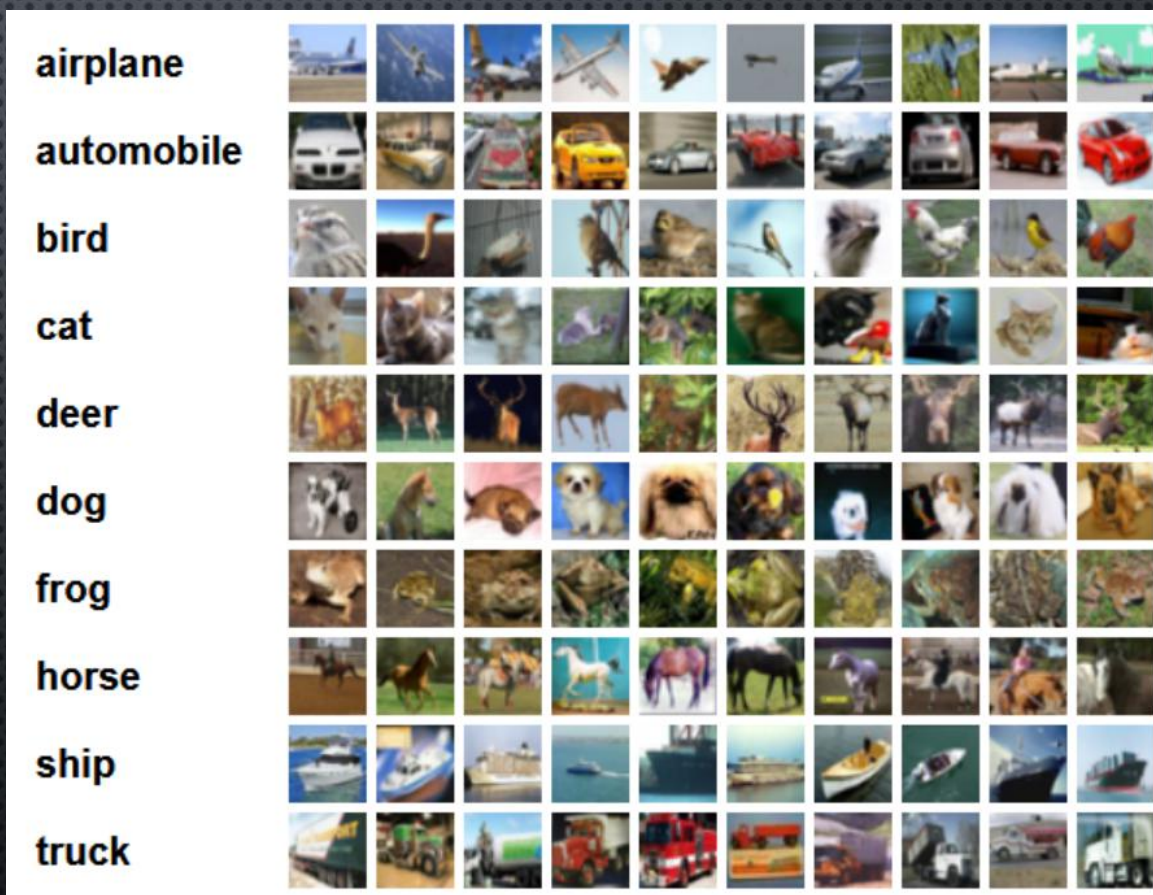
- 卷积神经网络运行结果



18.5 Cifar-10分类

➤ Cifar-10分类

- Cifar-10由60000张32*32的RGB 彩色图片构成，共10个分类。50000张训练，10000张测试。分为6个文件，5个训练数据文件，每个文件中包含10000张图片，随机打乱顺序，1个测试数据文件，也是10000张图片。
- 面对彩色数据集，用CPU做训练所花费的时间实在是太长了，所以本节将使用GPU来训练神经网络。



18.5 Cifar-10分类

- 环境搭建
 - ✓ 安装Python 3.6 (本书中所有案例在Python 3.6上开发测试)
 - ✓ 安装CUDA (没有GPU的读者请跳过)
 - ✓ 安装cuDNN (没有GPU的读者请跳过)
 - ✓ 安装TensorFlow, 有GPU硬件的一定要按照GPU版, 没有的只能安装CPU版
 - ✓ 安装Keras

Package	Version
-----	-----
Keras	2.2.5
Keras-Applications	1.0.8
Keras-Preprocessing	1.1.0
matplotlib	3.1.1
numpy	1.17.0
tensorboard	1.13.1
tensorflow-estimator	1.13.0
tensorflow-gpu	1.13.1

18.5 Cifar-10分类

- 在这个模型中：
 - ✓ 先用卷积->激活->卷积->激活->池化->丢弃层，做为第一梯队，卷积核32个；
 - ✓ 然后再用卷积->激活->卷积->激活->池化->丢弃层做为第二梯队，卷积核64个；
 - ✓ Flatten和Dense相当于把池化的结果转成 $N \times 512$ 的全连接层，N是池化输出的尺寸，被Flatten扁平化了；
 - ✓ 再接丢弃层，避免过拟合；
 - ✓ 最后接10个神经元的全连接层加Softmax输出。

18.5 Cifar-10分类

- 训练结果

- ✓ GPU

```
Epoch 1/25
1563/1563 [=====] - 33s 21ms/step - loss: 1.8770 - acc: 0.3103 - val_loss: 1.6447 - val_acc: 0.4098
.....
Epoch 25/25
1563/1563 [=====] - 87s 55ms/step - loss: 0.8809 - acc: 0.6960 - val_loss: 0.7724 - val_acc: 0.7372

Test loss: 0.772429921245575
Test accuracy: 0.7372
```

- ✓ CPU

```
Epoch 1/10
1563/1563 [=====] - 133s 85ms/step - loss: 1.8563 - acc: 0.3198 - val_loss: 1.5658 - val_acc: 0.4343
.....

Epoch 10/10
1563/1563 [=====] - 131s 84ms/step - loss: 1.0972 - acc: 0.6117 - val_loss: 1.0426 - val_acc: 0.6361

10000/10000 [=====] - 7s 684us/step
Test loss: 1.042622245979309
Test accuracy: 0.6361
```


THE END

谢谢！