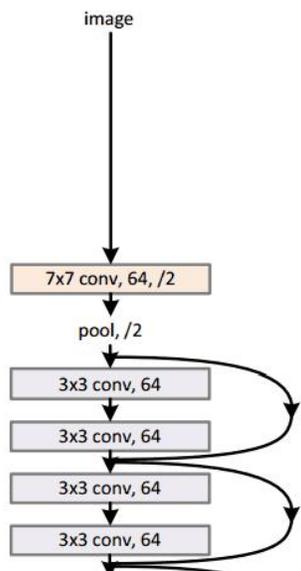
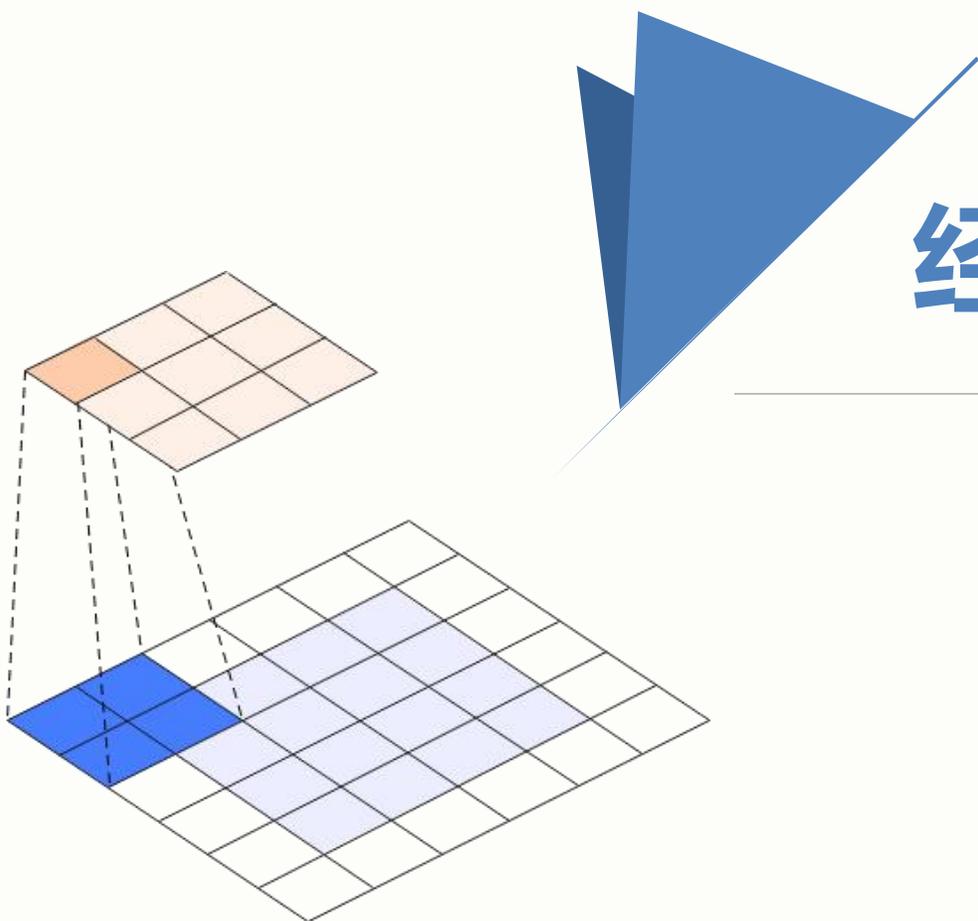


第06章 卷积神经网络

34-layer residual



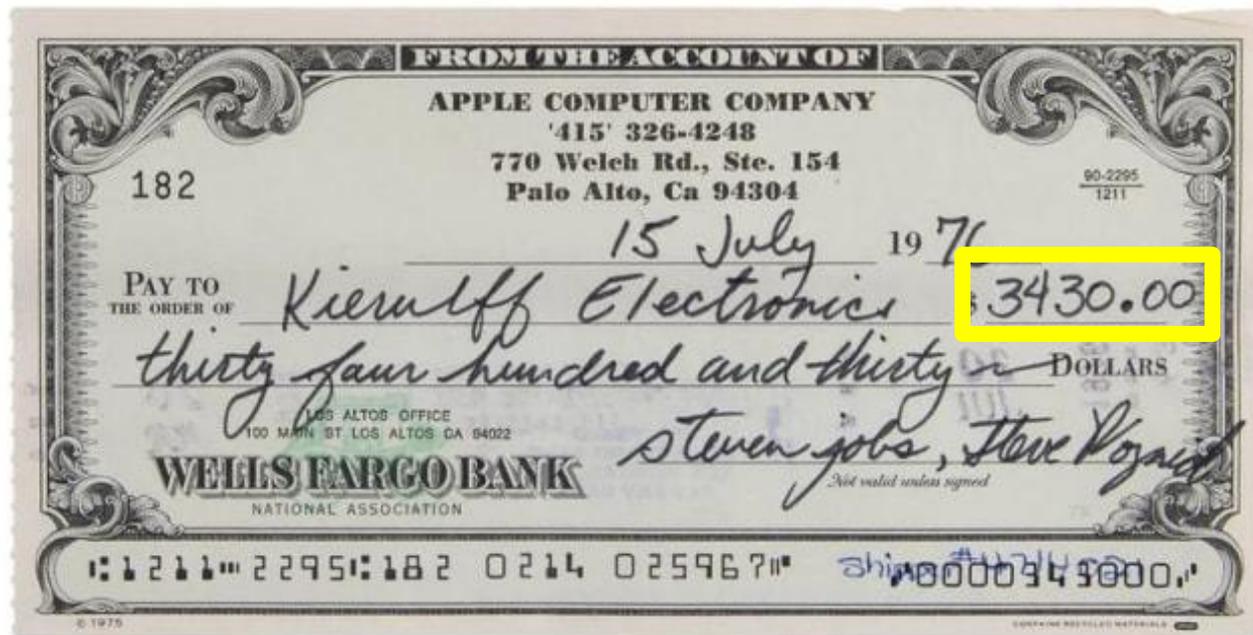
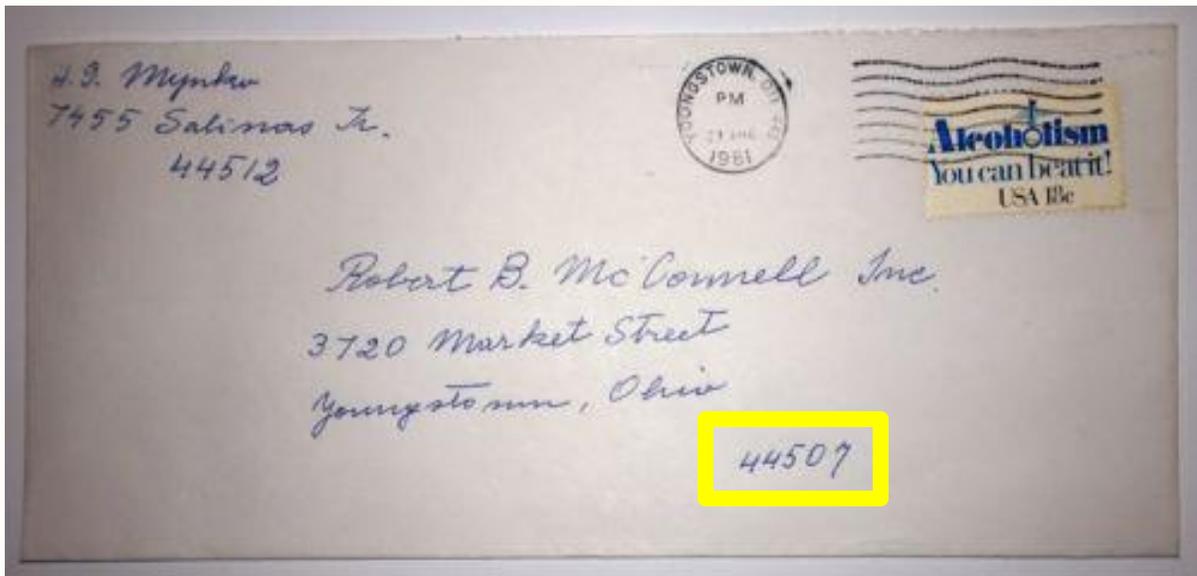
欧新宇



经典卷积神经网络LeNet

经典卷积神经网络LeNet

LeNet的起源

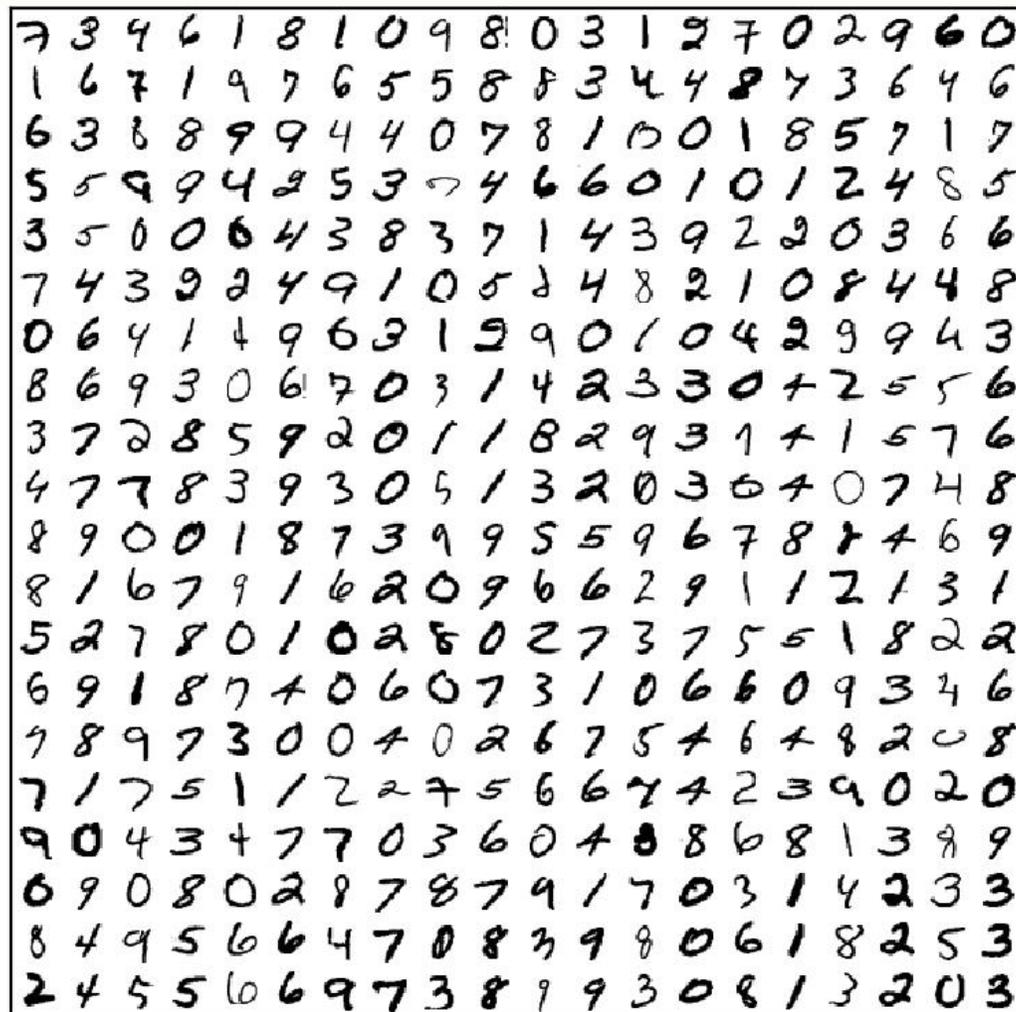


经典卷积神经网络LeNet

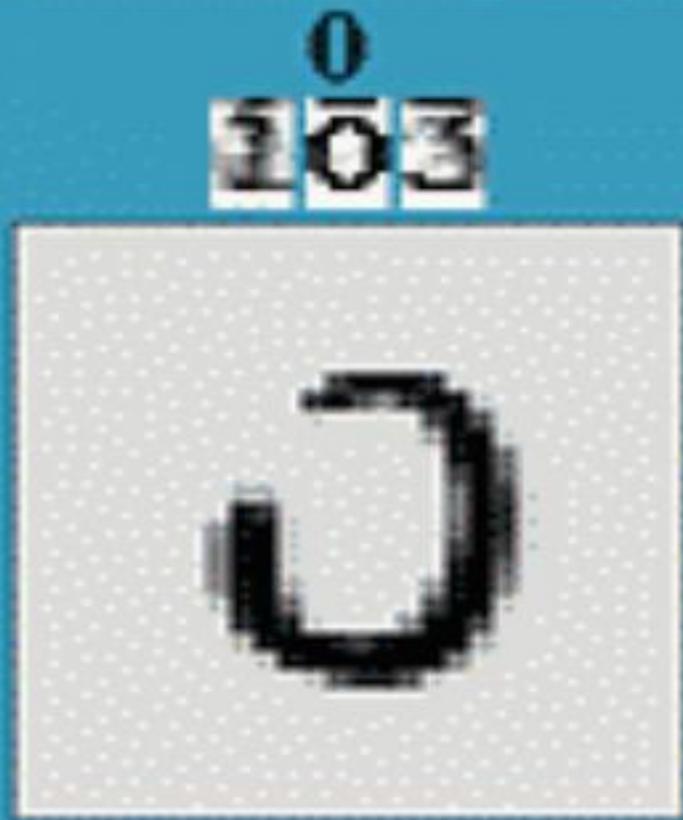
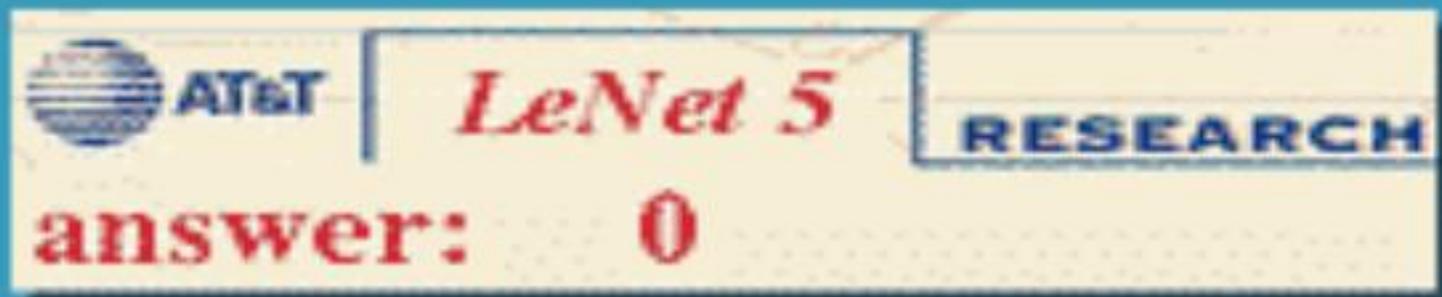
MNIST

MNIST 数据集来自美国国家标准与技术研究所(NIST, National Institute of Standards and Technology)。训练集由来自250 个不同人手写的数字构成, 其中 50% 是高中学生, 50% 来自人口普查局的工作人员. 测试集也是同样比例的手写数字数。

- 训练数据: 60000个
- 测试数据: 10000个
- 图像尺寸: 28×28
- 类别数: 10



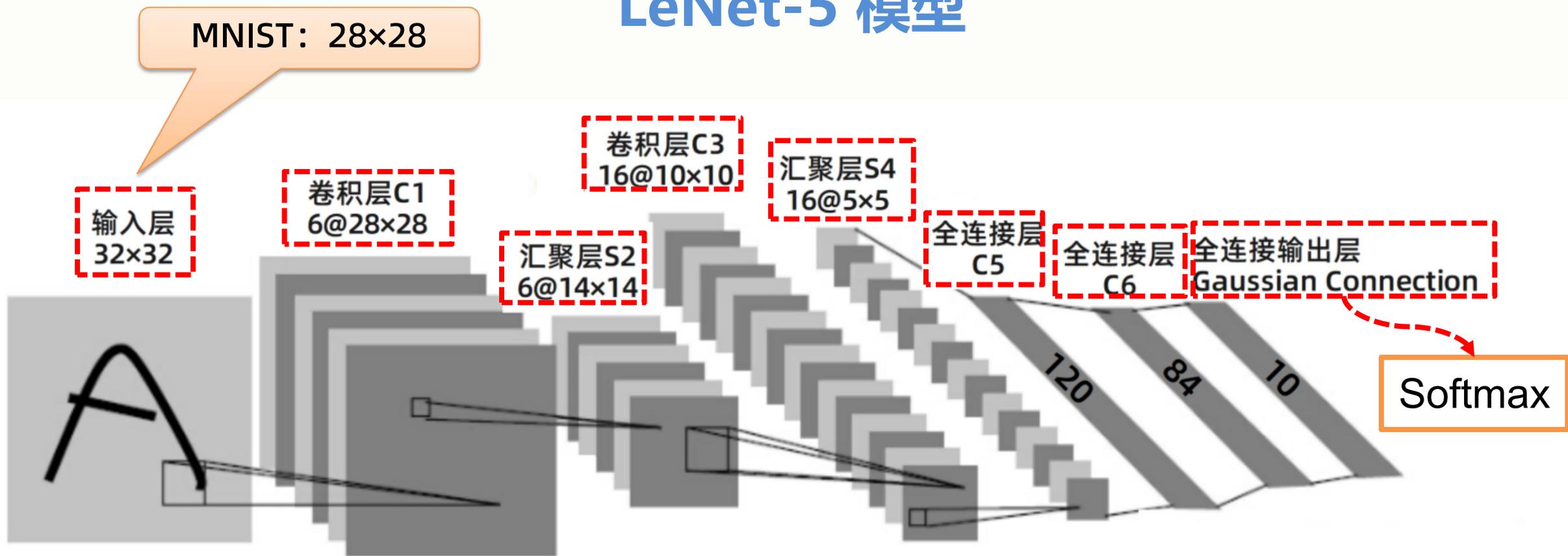
经典卷积神经网络LeNet



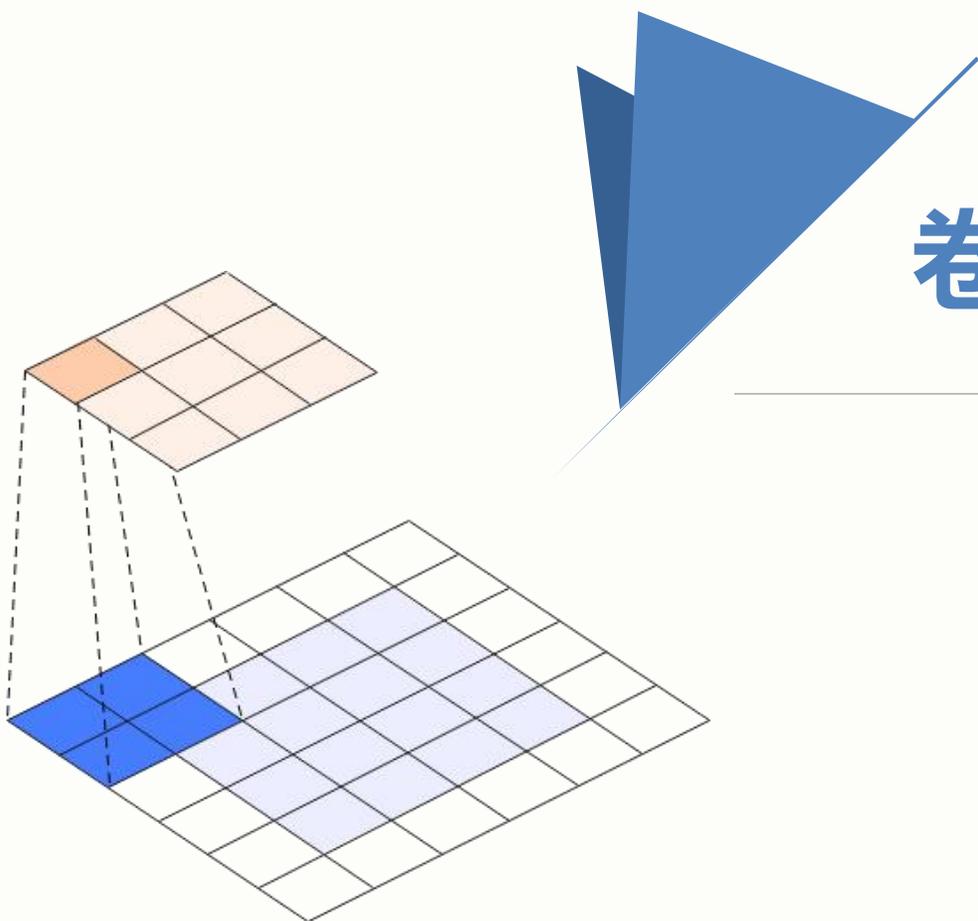
Yann LeCun, Leon Bottou,
Toshua Bengio, Patrick Haffne.
Gradient-Based Learning Applied
to Document Recognition. 1998.

经典卷积神经网络LeNet

LeNet-5 模型



卷积层: $\text{kernels_size}=[5,5], \text{stride}=[1,1], \text{padding}=[2,0]$ 汇聚层: $\text{kernel_size}=[2,2], \text{stride}=[2,2]$



卷积神经网络的基本训练流程

卷积神经网络的基本训练流程

“七步法” CNN训练代码流程

Step1: 导入全局依赖及全局参数配置

Step2: 数据准备: 定义数据预处理、数据载入、数据显示和测试

Step3: 定义和实例化网络类

Step4: 定义过程可视化函数

Step5: 配置训练基本参数: 优化器、损失函数和评价指标

Step6: 模型训练和在线测试

Step7: 离线评估

卷积神经网络的基本训练流程

“七步法” CNN训练代码流程

Step1: 导入全局依赖及全局参数配置

```
import numpy as np
import matplotlib.pyplot as plt
import paddle
```

Step2: 数据准备: 定义数据预处理、数据载入、数据显示和测试

```
1 # 1. 定义数据预处理
2 transform = transforms.Compose(transforms.ToTensor())
3
4 # 2. 从paddle.vision载入内置Mnist库, 并使用transform类进行归一化预处理
5 dataset_train = paddle.vision.datasets.MNIST(mode='train', transform=transform)
6 dataset_test = paddle.vision.datasets.MNIST(mode='test', transform=transform)
7
8 # 3. 创建数据迭代读取器
9 train_reader = paddle.io.DataLoader(dataset_train, batch_size=256, shuffle=True, drop_last=True)
10 test_reader = paddle.io.DataLoader(dataset_test, batch_size=256, shuffle=False, drop_last=False)
```

Python

卷积神经网络的基本训练流程

“七步法” CNN训练代码流程

Step3: 定义和实例化网络类

```

1 # 定义网络模型
2 # 方法一: 从paddle.vision中载入模型
3 LeNet5 = paddle.vision.models.LeNet(num_classes=10)
4
5 # 方法二: 自定义模型类
6 class LeNet5(paddle.nn.Layer):
7     def __init__(self):
8         super(LeNet5, self).__init__()
9
10 # 实例化模型的结构和参数, 并进行可视化
11 model = LeNet5()
12 paddle.summary(model, (1,1,28,28))

```



Layer (type)	Input Shape	Output Shape	Param #
Conv2D-1	[[1, 1, 28, 28]]	[1, 6, 28, 28]	156
Sigmoid-1	[[1, 6, 28, 28]]	[1, 6, 28, 28]	0
AvgPool2D-1	[[1, 6, 28, 28]]	[1, 6, 14, 14]	0
Conv2D-2	[[1, 6, 14, 14]]	[1, 16, 10, 10]	2,416
Sigmoid-2	[[1, 16, 10, 10]]	[1, 16, 10, 10]	0
AvgPool2D-2	[[1, 16, 10, 10]]	[1, 16, 5, 5]	0
Linear-1	[[1, 400]]	[1, 120]	48,120
Sigmoid-3	[[1, 120]]	[1, 120]	0
Linear-2	[[1, 120]]	[1, 84]	10,164
Sigmoid-4	[[1, 84]]	[1, 84]	0
Linear-3	[[1, 84]]	[1, 10]	850

```

Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0

```

```

Input size (MB): 0.00
Forward/backward pass size (MB): 0.11
Params size (MB): 0.24
Estimated Total Size (MB): 0.35

```

Step4: 定义过程可视化函数(略)

```
1 DrawAnimator(logs)
```

卷积神经网络的基本训练流程

“七步法” CNN训练代码流程

Step5: 配置训练基本参数: 优化器、损失函数和评价指标

```
1 # 1. 设置输入样本的维度、数据类型及名称
2 input_spec = paddle.static.InputSpec(shape=[None, 1, 28, 28], dtype='float32', name='image')
3 label_spec = paddle.static.InputSpec(shape=[None, 1], dtype='int64', name='label')
4
5 # 2. 载入预先定义好的网络, 并实例化model变量
6 network = LeNet5()
7 model = paddle.Model(network, input_spec, label_spec)
8
9 # 3. 设置学习率、优化器、损失函数和评价指标
10 optimizer = paddle.optimizer.SGD(learning_rate=0.9, parameters=model.parameters())
11 model.prepare(optimizer, # SGD优化函数
12               paddle.nn.CrossEntropyLoss(), # 交叉熵损失函数
13               paddle.metric.Accuracy(topk=(1,5))) # 精度评价指标
```

Step6: 模型训练和在线测试

```
1 total_epoch = 10
2 dl.train1(model, train_reader, test_reader, total_epoch)
```

卷积神经网络的基本训练流程

“七步法” CNN训练代码流程

Step7: 离线评估

1. 载入模型

```
model.load('D:\\Workspace\\DeepLearning\\Website\\Data\\Projects\\Project004LeNetMNIST\\Project004LeNetMNIST')
```

2. 运行评估函数在测试集上进行性能评估

```
model.evaluate(test_dataset, batch_size=64, verbose=1)
```

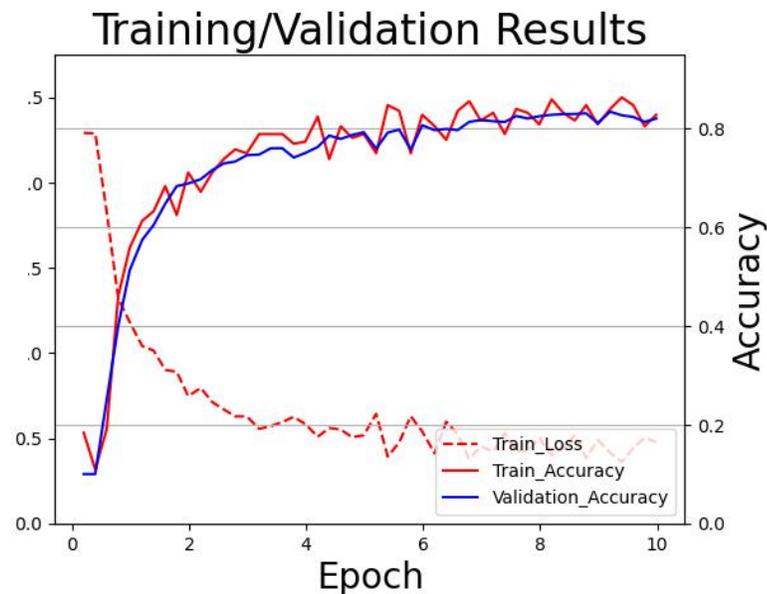
train_loss: 0.369,

train_acc_top1: 0.852,

train_acc_top5: 0.996,

val_acc_top1: 0.860,

val_acc_top5: 0.998



经典卷积神经网络LeNet

小 结

- LeNet是最早发布的卷积神经网络，它奠定了卷积神经网络在图像分析中的地位
- 卷积层用来学习图片的空间信息
- 汇聚层用来降低图像的维度，并获取尺度不变性
- 全连接层用来转换类别空间
- MNIST手写数字数据集和MNIST-Fashion数据集可以用来快速测试模型的可用性
- CNN模型的训练可以使用七步法

读万卷书 行万里路 只为最好的修炼



QQ: 14777591 (宇宙骑士)

Email: ouxinyu@alumni.hust.edu.cn

Website: <http://ouxinyu.cn>

Tel: 18687840023