



南方科技大学

SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CNN Hands on

04/25/2021

CNN

- 待优化的参数过多导致多拟合（全连接网络）
- 原始图片->特征提取->全连接网络
- 卷积是一种有效提取图像特征的方法：一般采用正方形卷积核，遍历图片上的每个点。图片区域内相对应的每一个像素值，成一卷积核内相对应点的权重，求和再加上偏置。

$5 \times 5 \times 1$

2	5	9	3	2
2	5	9	3	2
2	5	9	3	2
2	5	9	3	2
2	5	9	3	2

input

$3 \times 3 \times 1$

-1	0	1
-1	0	1
-1	0	1

$b=1$

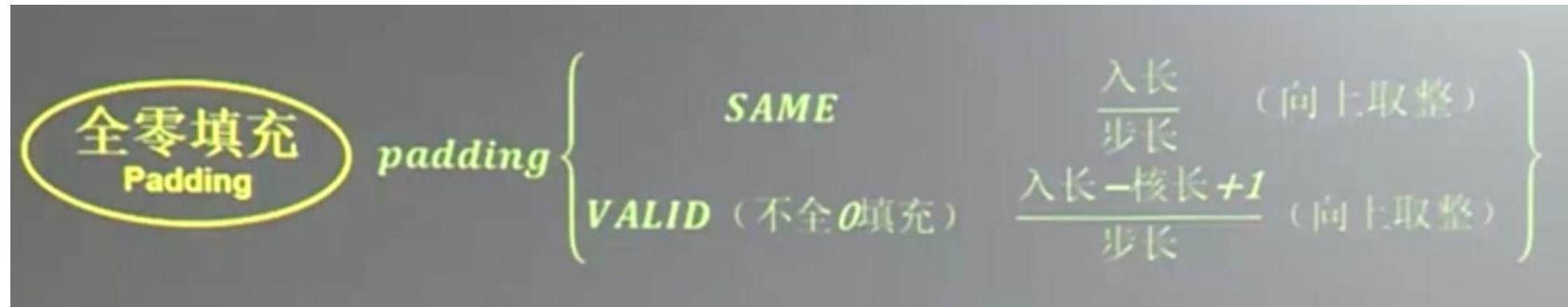
$3 \times 3 \times 1$

4	1	2
0	0	4
9	0	4

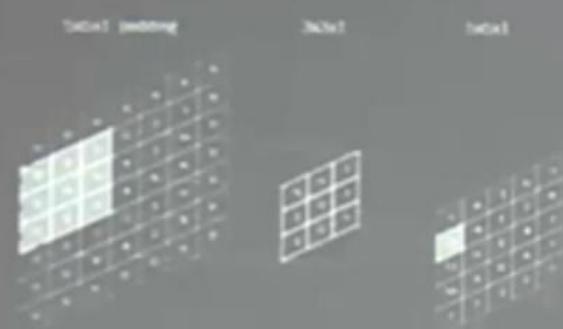
output

$$\begin{aligned} & (-1) * 1 + 0 * 0 + 1 * 2 \\ & + (-1) * 5 + 0 * 4 + 1 * 2 \\ & + (-1) * 3 + 0 * 4 + 1 * 5 \\ & + 1 = 1 \end{aligned}$$

在TensorFlow框架中，可以使用SAME（自动填充，使输出特征图尺寸和输入尺寸一样）和VALID（不做填充，会导致输出尺寸比输入要小）

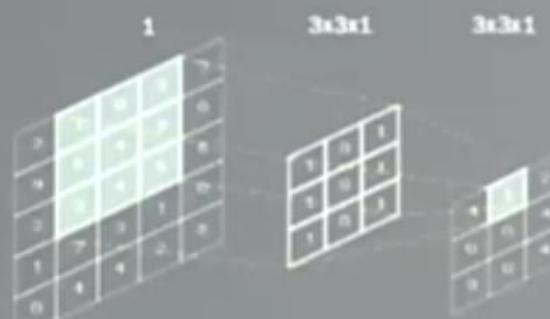


在Tensorflow框架中，
用参数padding = 'SAME' 或 padding = 'VALID'表示



输出长或宽： $\frac{5}{1} = 5$

SAME : $5 \times 5 \times 1 \Rightarrow 5 \times 5 \times 1$

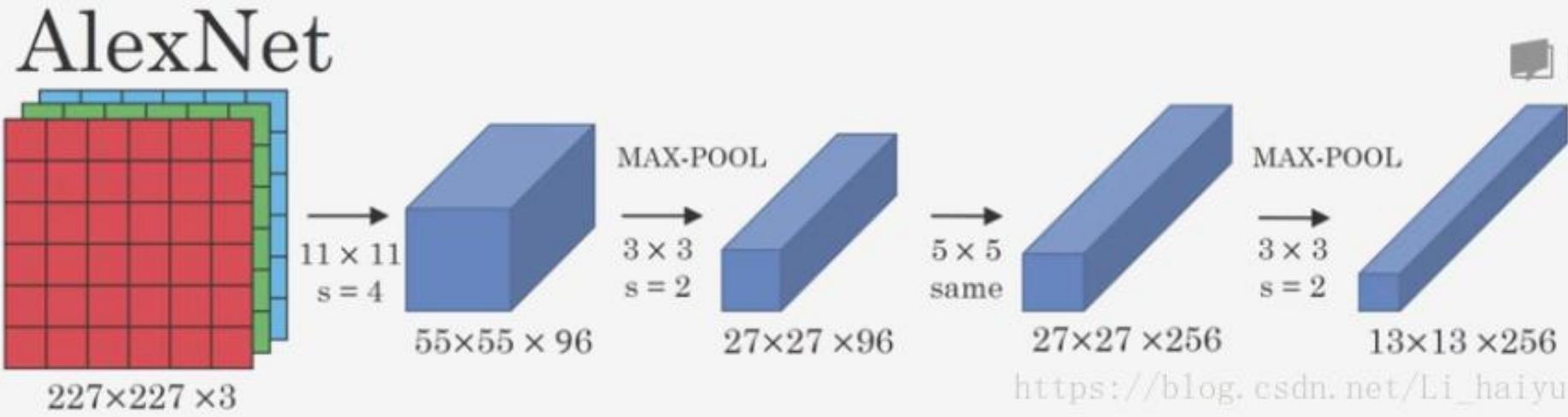


$$\frac{5 \cdot 3 + 1}{1} = 3$$

VALID : $5 \times 5 \times 1 \Rightarrow 3 \times 3 \times 1$

先定义几个参数

- 输入图片大小 $W \times W$
- Filter大小 $F \times F$
- 步长 S
- padding的像素数 P



于是我们可以得出

$$N = (W - F + 2P) / S + 1$$

- ① padding = "VALID", stride = 4, $(227 - 11 + 2*0) / 4 + 1 = 55$
- ② padding = "VALID", stride = 2, $(55 - 3 + 2*0) / 2 + 1 = 27$
- ③ padding = "SAME", stride = 1, $27 / 1 = 27$
- ④ padding = "VALID", stride = 2, $(27 - 3 + 2*0) / 2 + 1 = 13$

s:输出图片大小为 $N \times N$ haiyu

Tensorflow计算卷积：

`tf.nn.conv2d` (输入描述, eg. [batch, 5, 5, 1]

分辨率 通道数

卷积核描述, eg. [3, 3, 1, 16]

行列分辨率 通道数 核个数

核滑动步长, eg. [1, 1, 1, 1]

行步长 列步长

`padding = 'VALID'`)

input

5x5x3 padding

0	0	0	0	0	0	0
0	0	2	2	0	2	0
0	1	1	1	1	1	0
0	0	0	1	2	2	0
0	0	0	1	2	2	0
0	0	2	2	2	0	0
0	0	0	0	0	0	0

3x3x3

1	1	0
1	0	1
1	1	-1

1	0	1
1	1	0
1	0	1

-1	-1	1
0	1	0
0	1	1

0	0	0	0	0	0	0
0	1	0	2	1	1	0
0	1	2	2	1	0	0
0	1	1	0	2	1	0
0	0	0	1	1	1	0
0	2	1	1	2	0	0
0	0	0	0	0	0	0

 $b=1$

0	0	0	0	0	0	0
0	0	1	2	2	0	0
0	0	1	1	0	0	0
0	2	2	0	2	1	0
0	0	1	1	0	2	0
0	1	2	2	1	1	0
0	0	0	0	0	0	0

output

7	9	6	7	4
7	8	6	4	5
7	7	6	7	7
0	5	5	1	3
5	6	5	2	1

$$\begin{aligned}
 & (-1)*1 + 1*1 + 0*1 + (-1)*0 + 0*1 + 1*2 + (-1)*0 + 1*1 + (-1)*2 \\
 & + 1*2 + 0*2 + 1*1 + 1*1 + 1*0 + 0*2 + 1*0 + 0*0 + 1*1 \\
 & + (-1)*1 + (-1)*1 + (-1)*0 + 0*2 + 1*0 + 0*2 + 0*1 + 1*1 + 1*0 \\
 & + 1 = 6
 \end{aligned}$$

Tensorflow计算卷积：

`tf.nn.conv2d` (输入描述, eg. [batch, 5, 5, 3]

↓
行列分辨率 通道数

卷积核描述, eg. [3, 3, 3, 16]

↓
行列分辨率 通道数 核个数

核滑动步长, eg. [1, 1, 1, 1]

↓
行步长 列步长

`padding = 'SAME'`)

池化 Pooling

- 池化用于减少特征数量。
- 最大值池化可提取图片纹理，均值池化可保留背景特征。

Single depth slice

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

x

y

max pool with 2x2 filters
and stride 2



6	8
3	4

mean pool with 2x2 filters
and stride 2



3.25	5.25
2	2

Tensorflow计算池化：

`pool = tf.nn.max_pool` (输入描述, eg. [batch, 28, 28, 6]

`pool = tf.nn.avg_pool`

行列分辨率 通道数

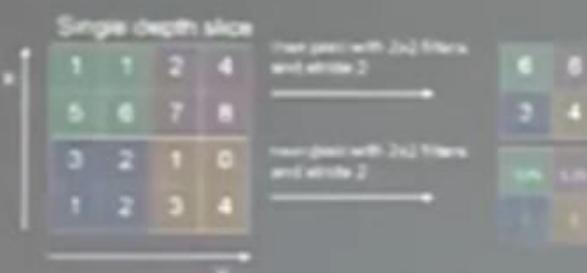
池化核描述 (仅大小), eg. [1, 2, 2, 1]

行列分辨率

池化核滑动步长, eg. [1, 2, 2, 1]

行步长 列步长

`padding = 'SAME'`)



慕课网

人工智能实践：Tensorflow笔记

<https://www.icourse163.org/course/PKU-1002536002>

Keras中文文档

https://keras-cn.readthedocs.io/en/latest/layers/convolutional_layer/

- 六步法

import : 引入相关模块

- train, test: 告知要喂如网络的训练集和测试集是什么, 要指定训练集以及测试集的的输入特征和标签

- model = tf.keras.models.Sequential : 在Sequential中搭建网络结构, 逐层描述每层网络 (=一次前向传播)

- model.compile : 在compile()中配置训练方法, 训练时使用哪种优化器, 哪个损失函数, 哪种评测指标

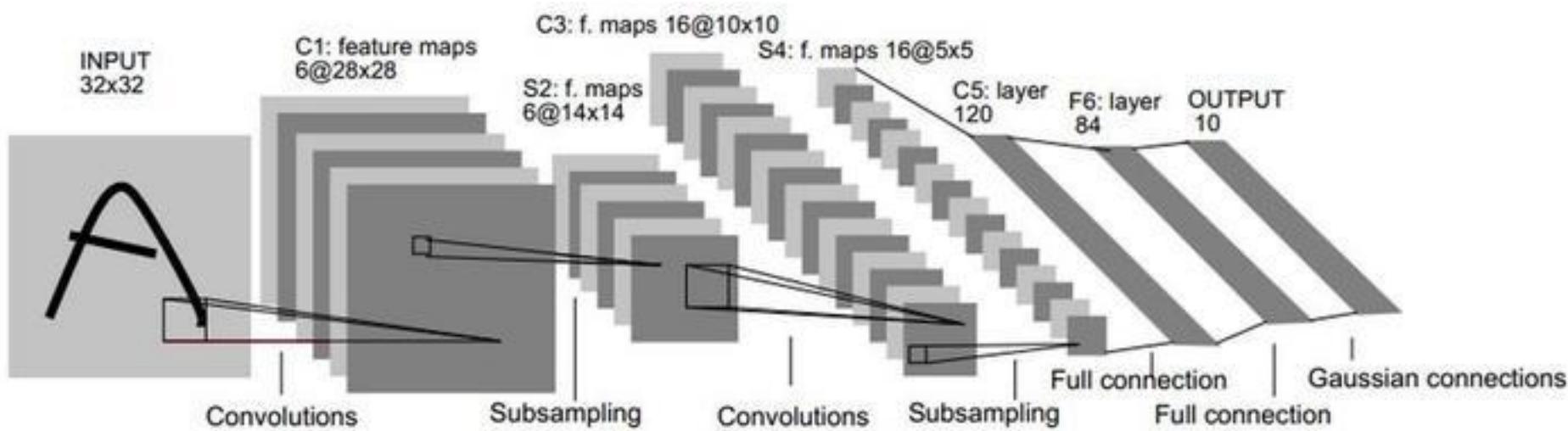
- model.fit : 在fit 中执行训练过程, 告知训练集和测试集的输入特征和标签, 每个batch是多少, 要迭代epoch多少次数据集

- model.summary: 使用summary()打印出网络的结构和参数统计

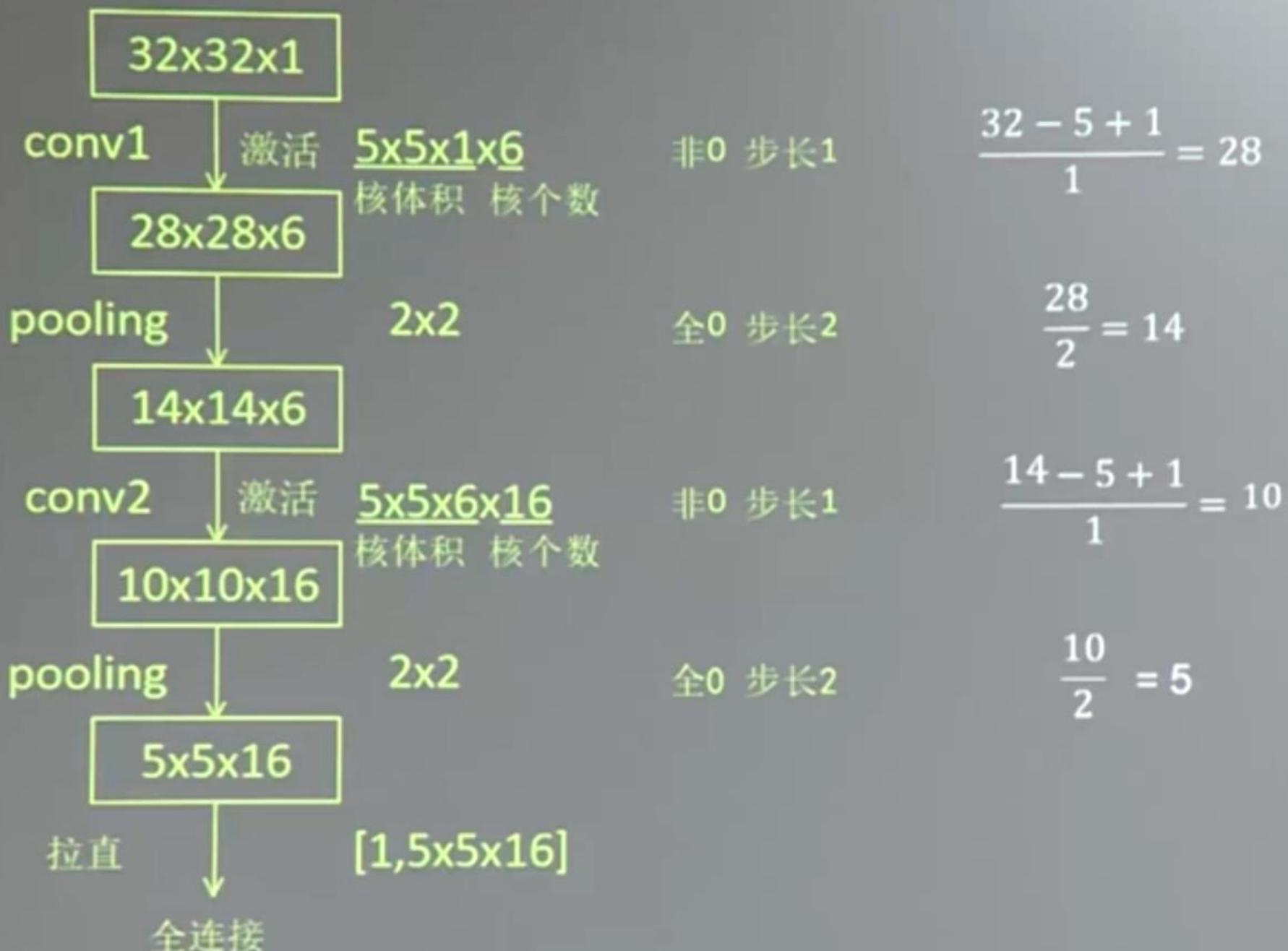
CNN hands on

LeNet模型分析讲解

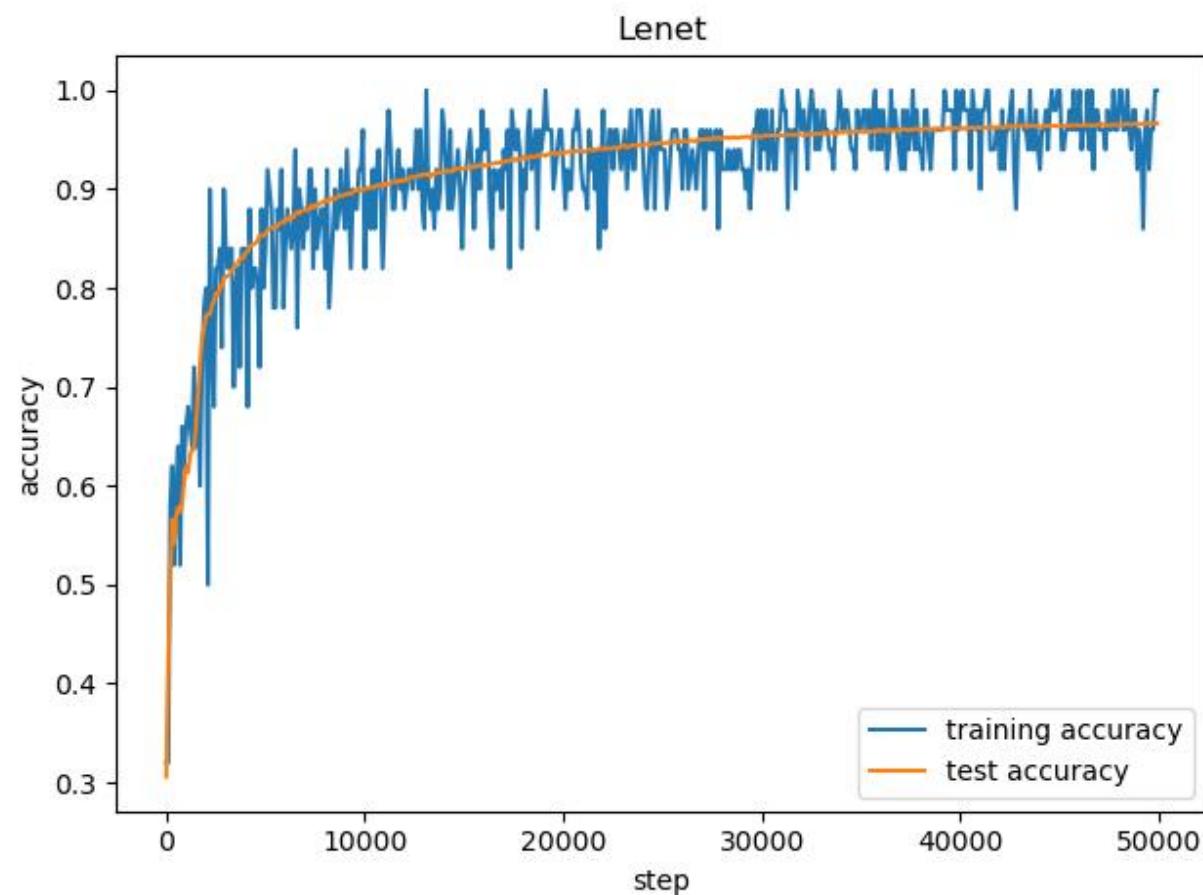
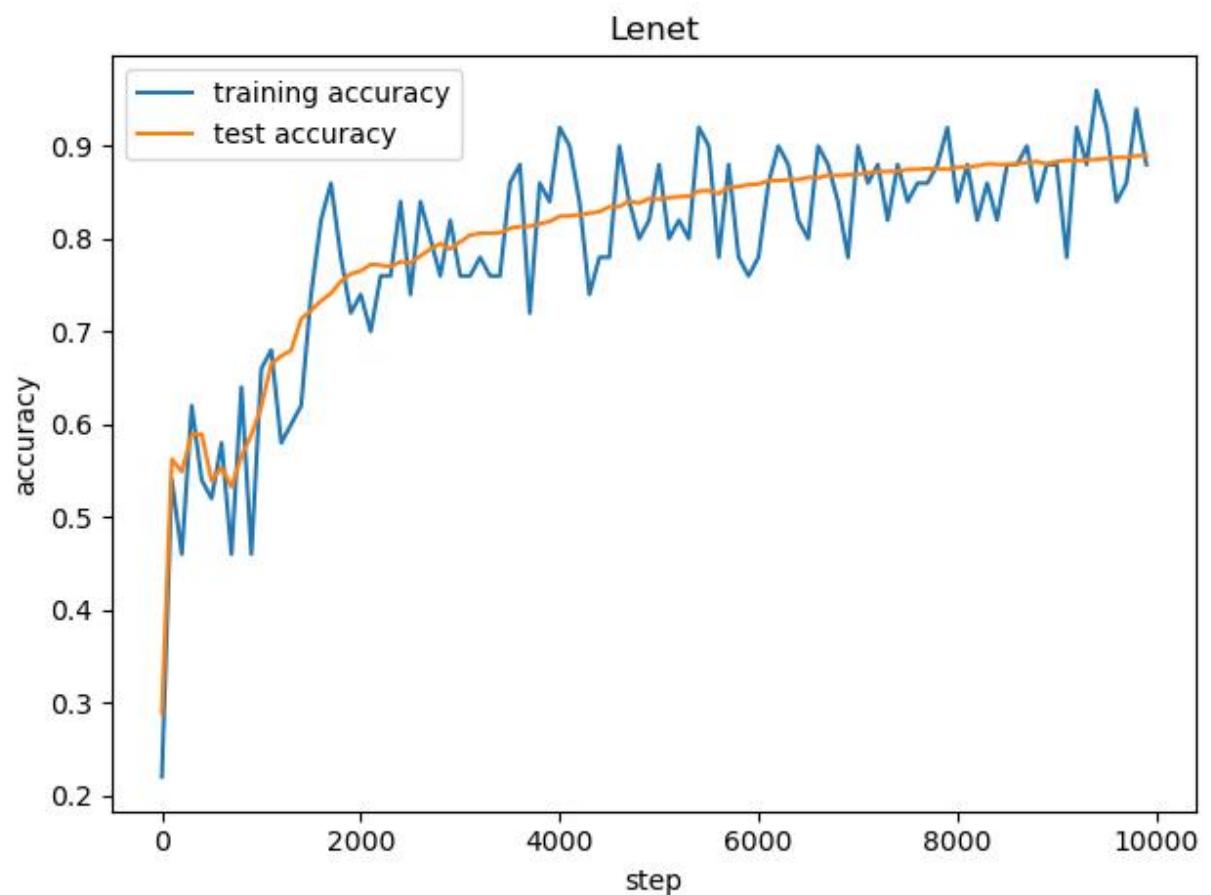
https://blog.csdn.net/qq_37787732/article/details/108502094



✓ Lenet-5



模型结果：



<https://github.com/YouzhiQu/LeNet>

CNN explainer: Learn CNN in your browser

<https://poloclub.github.io/cnn-explainer/>

<https://github.com/poloclub/cnn-explainer>

Convolutional Neural Network Visualizations

<https://github.com/utkuozbulak/pytorch-cnn-visualizations>

Visualizing and understanding convolutional networks

- A PyTorch implementation of the 2014 ECCV paper "Visualizing and understanding convolutional networks"
- <https://github.com/huyber/VisualizingCNN>