

使用 PyTorch 搭建深度卷积神经网络, 并进行图像分类识别

2018 年 11 月 28 日

摘 要

卷积神经网络 (Convolutional Neural Network, CNN) 是一种前馈神经网络, 它的人工神经元可以响应一部分覆盖范围内的周围单元, 对于大型图像处理有出色表现。

卷积神经网络由一个或多个卷积层和顶端的全连通层 (对应经典的神经网络) 组成, 同时也包括关联权重和池化层 (pooling layer)。与其他深度学习结构相比, 卷积神经网络在图像和语音识别方面能够给出更好的结果。

卷积层

卷积层 (Convolutional layer), 卷积神经网络中每层卷积层由若干卷积单元组成, 每个卷积单元的参数都是通过反向传播算法最优化得到的。卷积运算的目的是提取输入的不同特征, 第一层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级, 更多层的网路能从低级特征中迭代提取更复杂的特征。

池化层

池化 (Pooling) 是卷积神经网络中另一个重要的概念, 它实际上是一种形式的降采样。有多种不同形式的非线性池化函数, 而其中 “最大池化 (Max pooling)” 是最为常见的。直觉上, 这种机制能够有效地原因在于, 在发现一个特征之后, 它的精确位置远不及它和其他特征的相对位置的关系重要。池化层会不断地减小数据的空间大小, 因此参数的数量和计算量也会下降, 这在一定程度上也控制了过拟合。通常来说, CNN 的卷积层之间都会周期性地插入池化层。[1]

1 问题描述

使用 PyTorch[2][3]机器学习库搭建卷积神经网络 (CNN), 并能够高效的对 cifar10 数据集进行分类。利用数据集中给出的训练模型进行训练, 并在测试集上进行分类以验证模型的效能。

2 数据集描述

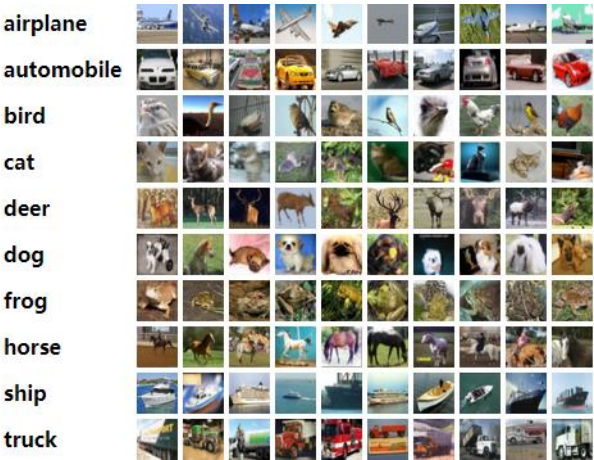
数据集名称: CIFAR-10[4]

CIFAR-10 数据集包含了 60,000 32x32 彩色图像, 共有 10 个分类。其中有 50,000 张图片为训练集, 10,000 张图片为测试集。

数据集被分成了 5 批训练数据和 1 批测试数据，每一批为 10,000 张图片。测试集包含了从每个类中随机选取的共 10,000 张图片，训练集也是随机选取，但是训练批可能在某一个类中包含更多的数据。

10 个类分别为：
airplane, automobile, bird, cat,
deer, dog, frog, horse, ship,
truck。

同时 automobiles 和 trucks 之间没有重叠部分，完全不同。两类图片之间均没有皮卡。

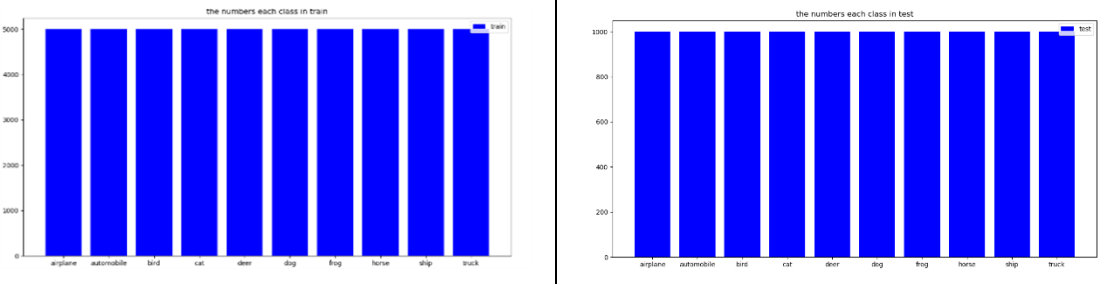


图片来源：[4]

数据集下载地址：<https://www.cs.toronto.edu/~kriz/cifar.html>

官网上同时提供了从每一批次数据提取文件的方法。

下图分别为在训练集和测试集上的各个类别的个数：



3 实验结果图

3.1 CNN 神经网络的参数描述：

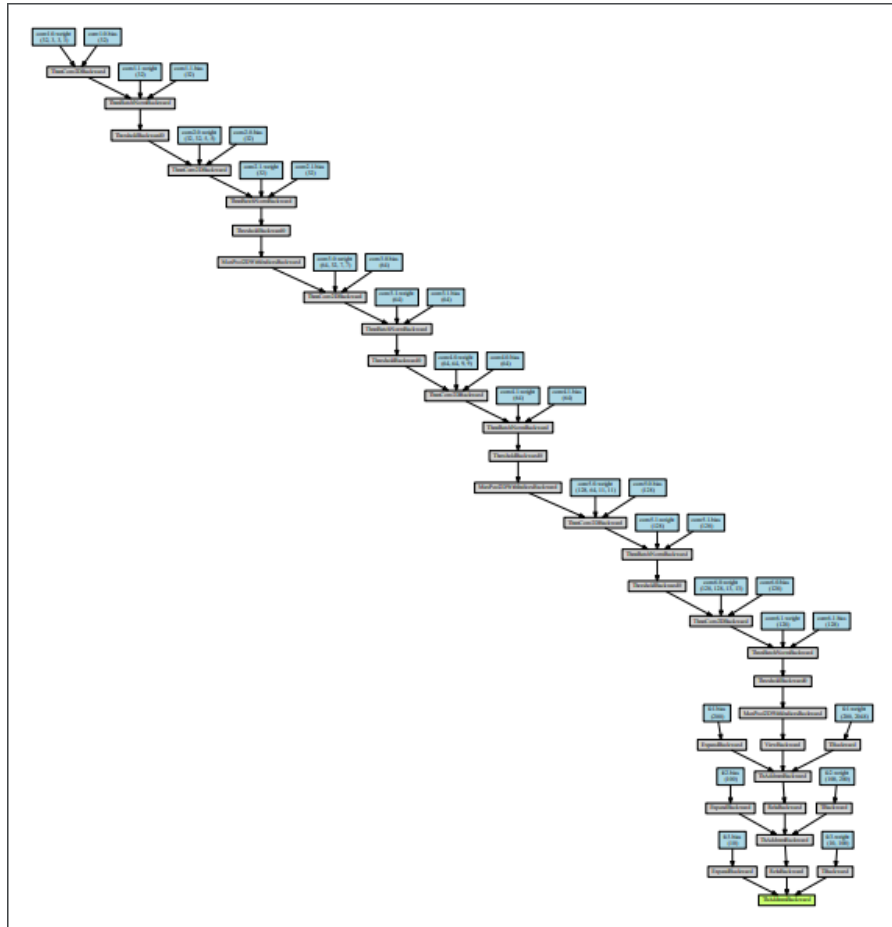
- 文字描述：

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	896
BatchNorm2d-2	[-1, 32, 32, 32]	64
ReLU-3	[-1, 32, 32, 32]	0
Conv2d-4	[-1, 32, 32, 32]	25,632
BatchNorm2d-5	[-1, 32, 32, 32]	64
ReLU-6	[-1, 32, 32, 32]	0
MaxPool2d-7	[-1, 32, 16, 16]	0

Conv2d-8	[-1, 64, 16, 16]	100,416
BatchNorm2d-9	[-1, 64, 16, 16]	128
ReLU-10	[-1, 64, 16, 16]	0
=====		
Conv2d-11	[-1, 64, 16, 16]	331,840
BatchNorm2d-12	[-1, 64, 16, 16]	128
ReLU-13	[-1, 64, 16, 16]	0
MaxPool2d-14	[-1, 64, 8, 8]	0
=====		
Conv2d-15	[-1, 128, 8, 8]	991,360
BatchNorm2d-16	[-1, 128, 8, 8]	256
ReLU-17	[-1, 128, 8, 8]	0
=====		
Conv2d-18	[-1, 128, 8, 8]	2,769,024
BatchNorm2d-19	[-1, 128, 8, 8]	256
ReLU-20	[-1, 128, 8, 8]	0
MaxPool2d-21	[-1, 128, 4, 4]	0
=====		
Linear-22	[-1, 200]	409,800
Linear-23	[-1, 100]	20,100
Linear-24	[-1, 10]	1,010
=====		
Total params: 4,650,974		
Trainable params: 4,650,974		
Non-trainable params: 0		

Input size (MB): 0.01		
Forward/backward pass size (MB): 2.74		
Params size (MB): 17.74		
Estimated Total Size (MB): 20.49		

- 图形化的网络结构展示:

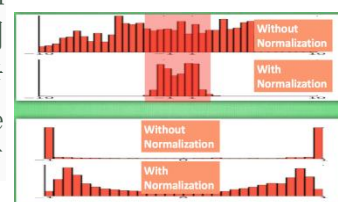
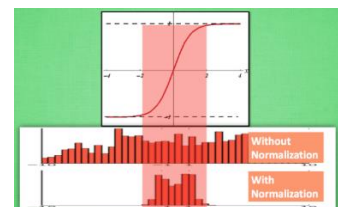


高清大图请点击[链接](#)

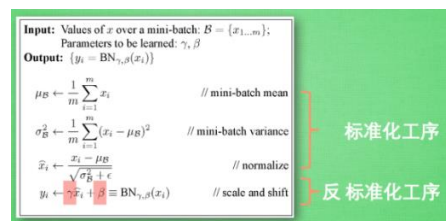
- Learning rate 设置为 0.001
- 六层卷积核的个数和大小分别为 <大小, 个数> :
<3, 32>, <5, 32>, <7, 64>, <9, 64>, <11, 128>, <13, 128>
- 池化层的设置为每两层卷积设置一个池化层, 均为: kernel size = 2, stride = 2
- 每层卷积后紧跟一层 Batch Normalization, 有效提高准确率。

Batch Normalization (BN) 被添加在激励函数之前, 这里以 tanh 激活函数为例, 只有当数据值大多数处在这个区间分布的时候, 才能进行更有效地传递。

没有 normalize 的数据 使用 tanh 激活以后, 激活值大部分都分布到了饱和阶段, 也就是大部分的激活值不是 -1, 就是 1, 而 normalize 以后, 大部分的激活值在每个分布区间都还有存在。



右侧为批标准化的公式



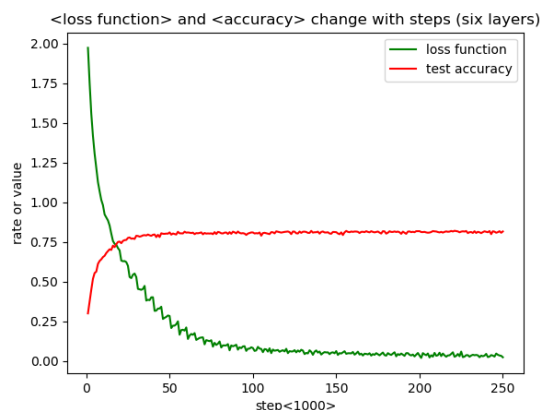
图片来源：莫烦网站

- 激活函数均为 ReLu
- Dense 层设置为三层全连接<in_features, out_features>:
<2048, 200>, <200, 100>, <100, 10>
- 损失函数为 crossEntropyLoss: 该函数计算一共分为两步
 - 1、先对输入计算 log softmax
 - 2、对上一步的结果计算交叉熵 最后输出

3.2 损失函数的变化曲线等

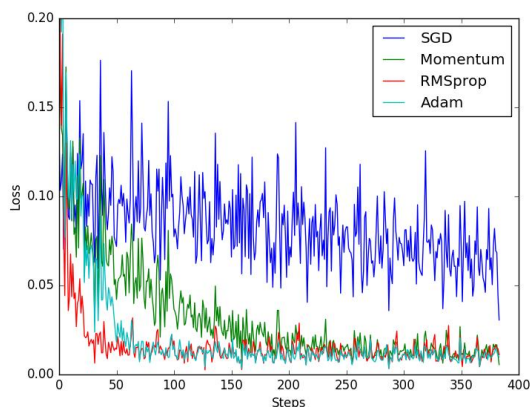
神经网络在测试集上的最高准确率为 82%。相对来说该模型的泛化能力比较不错。batch = 10; epoch = 25; 每经过 1000 次训练输出一次在测试集上的准确率。

大约在第五个 epoch 的时候准确率已经保持不变，但损失函数还在下降，说明该模型已经达到了饱和状态，继续训练的话也许会过拟合。



优化器选择：Adam

网络上查找资料，经对比该优化器的优化效果相对来说要好一点。



图片来源：莫烦网站

为了验证卷积层和池化层在 CNN 网络结构中的作用，我另搭建了八层只有

全连接的网络结构。

实验前猜想因为全连接没有抛弃图片的特征，准确率可能会比 CNN 要高。

也许因为我的网络层数不够多，或者其他原因也有可能。

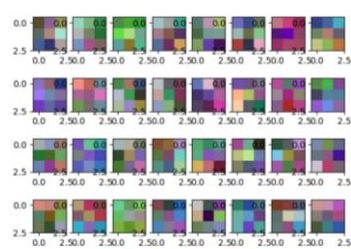
但据实验的结构表明，全连接网络不仅训练过程会比较困难，损失函数的值下降的比较慢，同时在测试集上的准确率也只能达到 50%。

所以 CNN 网络在图片识别提取图片的特征方面更有优势

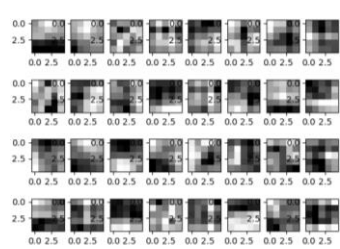


3.3 各层卷积核的可视化

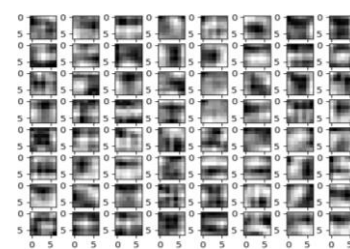
● 现在的网络模型



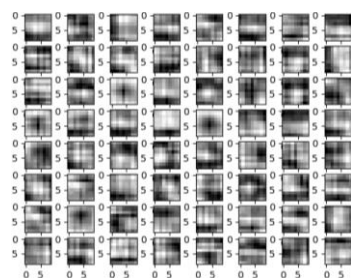
第一层卷积核 (32, 3, 3)



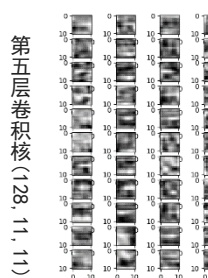
第二层卷积核 (32, 5, 5)



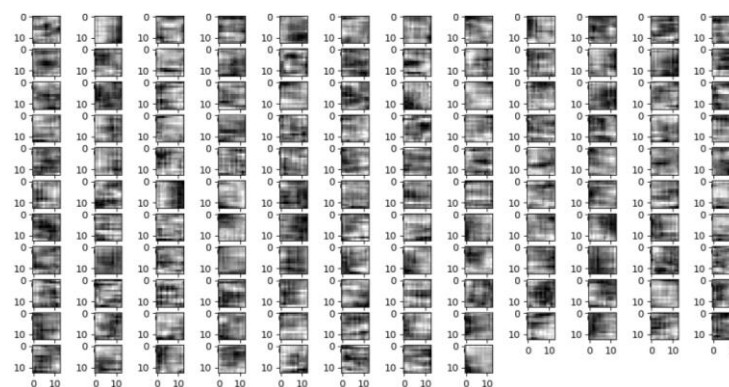
第三层卷积核 (64, 7, 7)



第四层卷积核 (64, 9, 9)

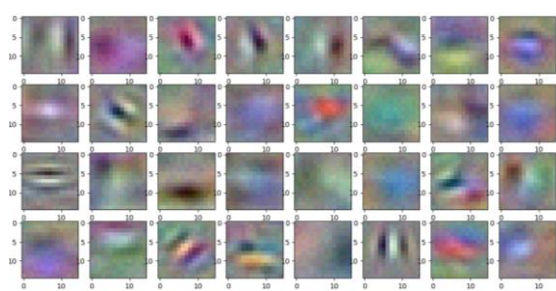


第五层卷积核 (128, 11, 11)

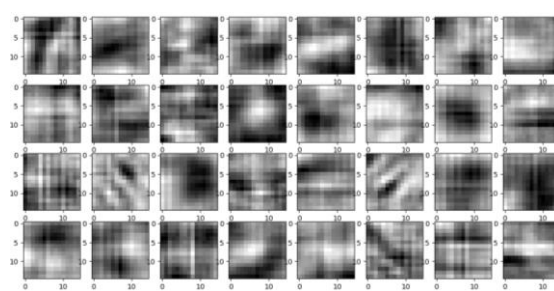


第六层卷积核 (128, 13, 13)

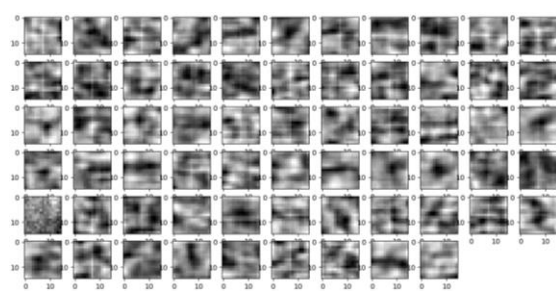
● 为了可视化卷积核而搭建的网络模型



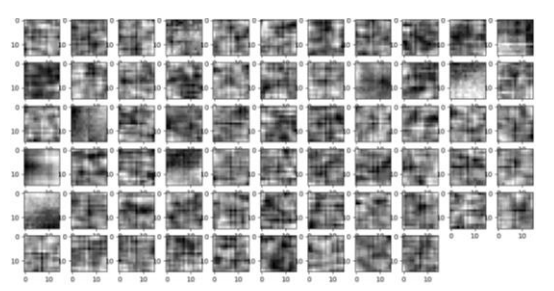
第一层卷积核 (32, 15, 15)



第二层卷积核 (32, 15, 15)



第三层卷积核 (64, 15, 15)

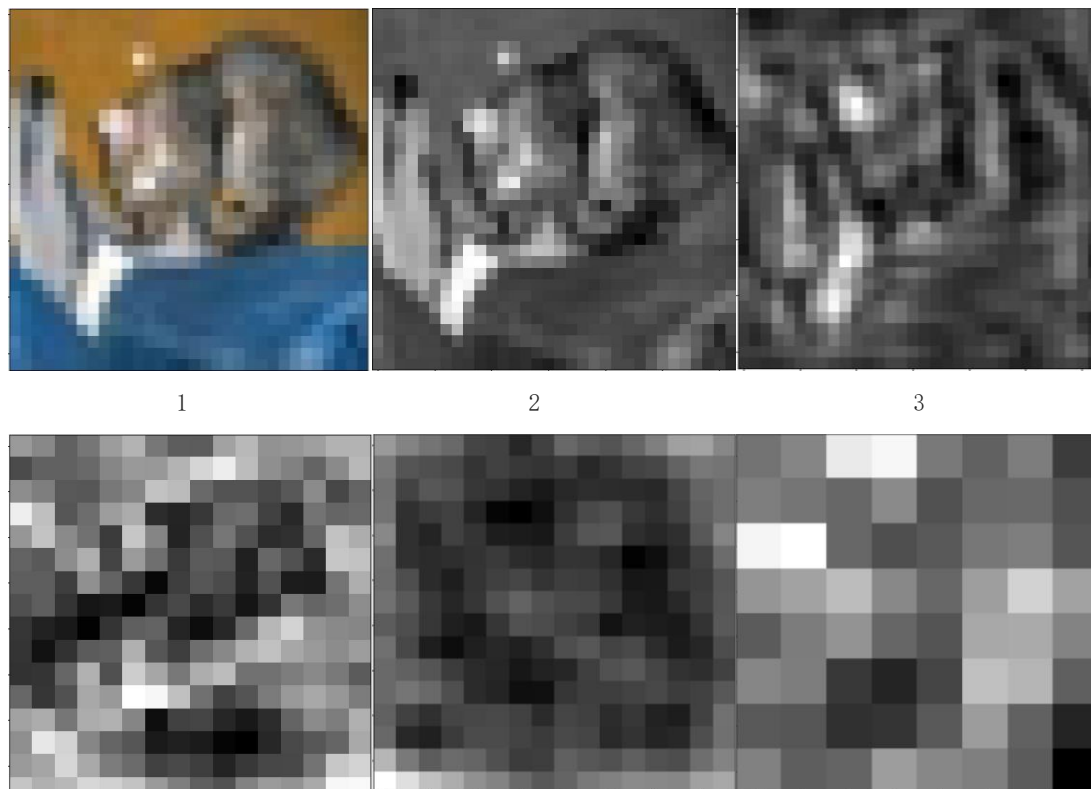


第四层卷积核 (64, 15, 15)

第二个网络在测试集上的准确率能够达到 75%

第一层的卷积核学到了图片一些局部线条以及一些颜色特征。由于之后的卷积核的通道数大于三，所以无法用 RGB 模式显示，只能压缩为灰度图像。可以看出第二层卷积核学到的特征要比第一层卷积核更多且有更大的轮廓。直到最后一层卷积核，开始显示图接近原本图像的一些轮廓。

3.4 随机选取某一图像在各层卷积后的结构



选择了一张猫的图片，可爱的猫咪在经历了残忍的卷积层和池化层的洗礼之后变得面目全非。

第一张图片为原始图片，经过第一层（卷积，batchnormal，relu）之后保留大部分特征（第二张图片），经过下一层（卷积，batchnormal，relu，池化），保有形状和高亮一些特征（第三张图片）。以此向前传到，可以观察到原本图片的特征越来越模糊，直到最后变成了只有灰暗亮度没有形状的色块。

4 实验结果分析

卷积神经网络中最主要的两个部分就是卷积层和池化层。

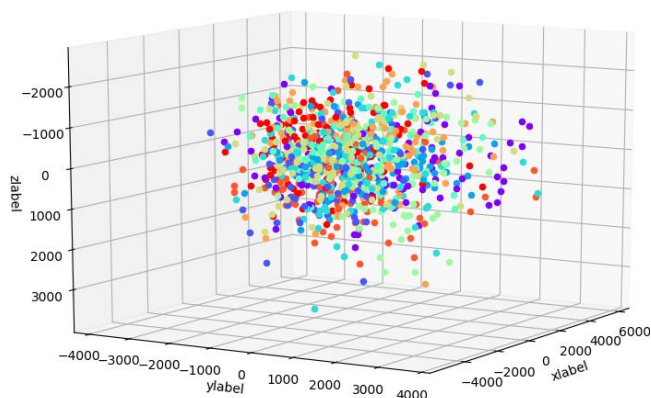
卷积层主要的作用就是提取图片的线条特征、色彩特征以及一些其他的形状特征，且越深层的卷积提取出来的特征就越多。卷积核的作用和动物视觉细胞的分工类似。

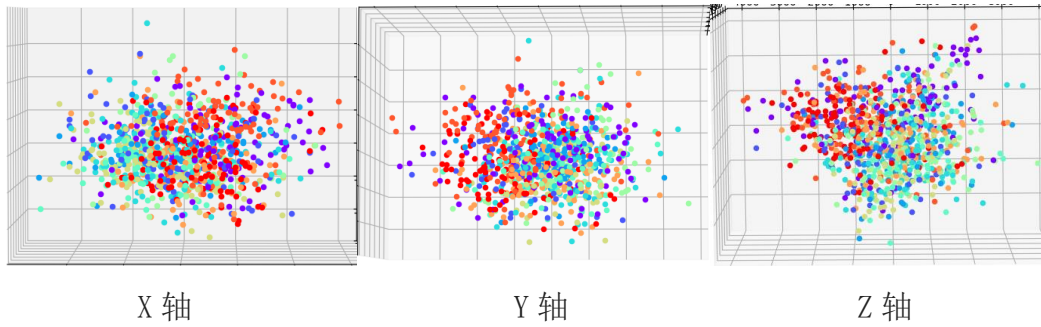
除此之外，卷积层还能够减少冗余的数据，抛弃一些不重要的数据，减少训练的负担。

池化层的主要作用就是降采样，减少数据的数量，提高网络的计算效率。并且池化能够帮助输入保持一种平移不变性，既不用关心某个特征出现的位置，而只关心是否具有该特征。

5 对图片降维

选取测试集中 1000 张图片（32x32）降到三维，观察不同类别的图片在三维空间中是否有某种聚类关系。





总体来说，从各个不同的方向观察三维空间的三点图，属于同一个类别的图片的点的位置呈聚集的态势。

Note: 以上大部分图片均可点击，以有更良好的视觉体验

参考资料

- [1] 维基百科 [卷积神经网络](#)
- [2] 维基百科 [PyTorch](#)
- [3] PyTorch [官网网站](#)
- [4] CIFAR [dataset](#)