LeNet

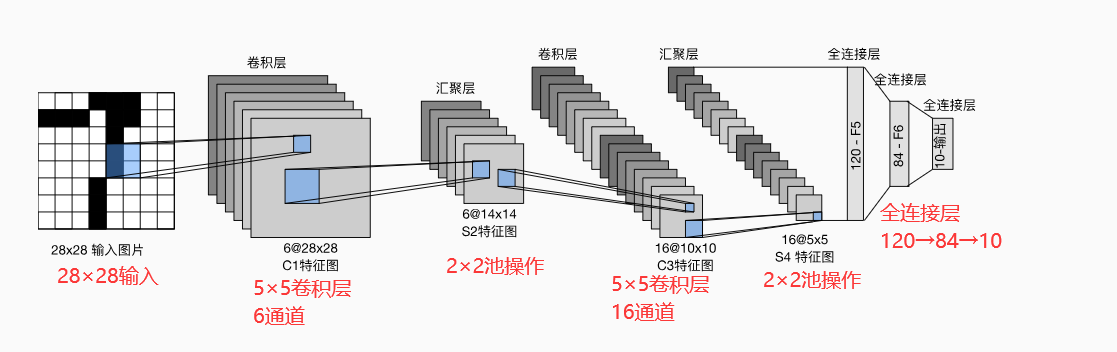
**卷积就是让每一层shape不断压缩变小（高宽减少），通道数增多，把特征信息放到不同的通道里面。每一个通道认为是一个模式。然后再做全连接的输入。**

两层卷积+两层池化+两层全连接

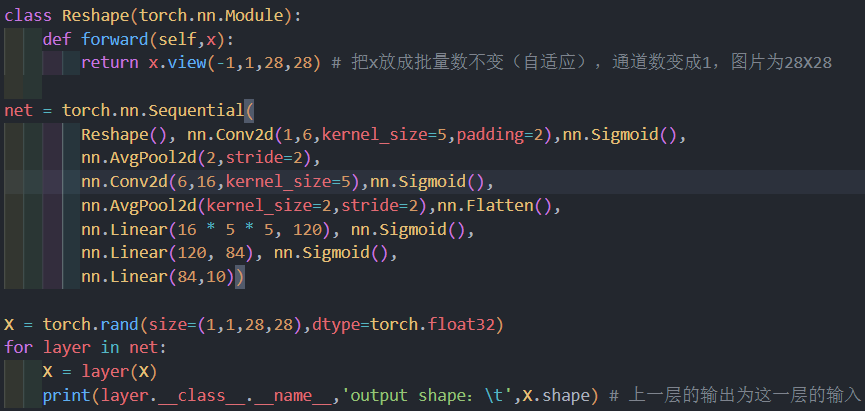
LeNet先用卷积层来学习图片空间信息，再通过池化层降低图片敏感度，最后使用全连接层（多层感知机）来转换到类别空间得到十类。

**LeNet（LeNet-5）**由两个部分组成：

* **卷积编码器**：由两个卷积层组成；
* **全连接层密集块**：由三个全连接层组成。



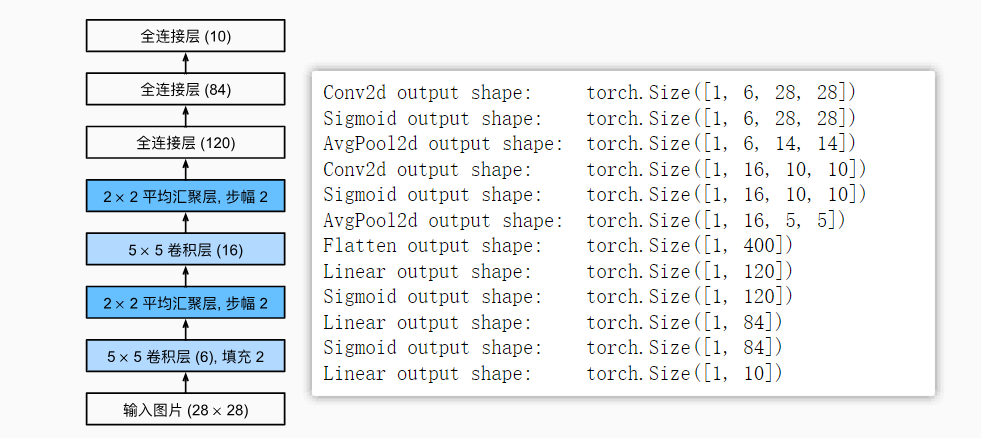
1. 每个卷积块中的基本单元是一个卷积层、一个sigmoid激活函数和平均汇聚层。
2. 每个卷积层使用5×5卷积核和一个sigmoid激活函数。
3. 这些层将输入映射到多个二维特征输出，通常同时增加通道的数量。
4. 每个4×4池操作（步幅2）通过空间下采样将维数减少4倍。



Reshape将输入张量 x 调整形状为 (batch\_size, 1, 28, 28)，即批量大小不变，通道数变为1，图像尺寸为28x28。

**nn.Flatten()**：把第一个批量维度保持，其他维度拉成一个向量。

该模型去掉了最后一层的高斯激活，下面for将一个大小为28×28的单通道（黑白）图像通过LeNet，打印每一层输出的形状。



1.第一个卷积层使用2个像素的填充，来补偿5×5卷积核导致的特征减少。

2.第二个卷积层没有填充，因此高度和宽度都减少了4个像素。

3.随着层叠的上升，通道的数量从输入时的1个，增加到第一个卷积层之后的6个，再到第二个卷积层之后的16个。

4.每个池化层的高度和宽度都减半。

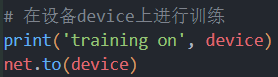
5.每个全连接层减少维数，最终输出一个维数与结果分类数相匹配的输出。



训练函数与之前的基本无差别，这里增加了用GPU做运算。

* device：计算设备，如CPU或GPU。

**Xavier**均匀分布初始化线性层和卷积层的权重。



将网络模型移动到GPU上



把输入输出移动到GPU上