cs231n笔记（一）：

1，pytorch的variable是一个计算图中的节点，若x是一个variable，x.data是一个张量，x.grad是variable的梯度，x.grad.data是梯度的张量。

2，pytorch的variable切换到gpu版本可以使用x=Variable(torch.randn(3,4).cuda()，.....)。

3，TensorFlow里初始化使用tf.contrib.layers.xavier_initializer()可以取得比较好的效果。

4，numpy里面矩阵与向量的操作采用广播机制。

5，scipy和matplotlib的使用。

6，CNN层的排列规律：

最常见的卷积神经网络结构如下:
INPUT -> [[CONV -> RELU]*N -> POOL?]M -> [FC -> RELU]K -> FC
（其中N >=0,通常N<=3,M>=0,K>=0,通常K<3）

一些常见的网络结构规律：
INPUT -> FC,实现一个线性分类器，此处N = M = K = 0。
INPUT -> CONV -> RELU -> FC
INPUT -> [CONV -> RELU -> POOL]*2 -> FC -> RELU -> FC。此处在每个汇聚层之间有一个卷积层。
INPUT -> [CONV -> RELU -> CONV -> RELU ->

POOL]*3 -> [FC -> RELU]*2 -> FC。
此处每个池化层前有两个卷积层，这个思路适用于更大更深的网络，因为在执行具有破坏性的池化操作前，多重的卷积层可以从输入数据中学习到更多的复杂特征。几个小滤波器卷积层的组合比一个大滤波器卷积层好

7，CNN的层的尺寸设置规律

输入层：应该能被2整除
卷积层：应该使用小尺寸滤波器（比如3x3或最多5x5），使用步长为1，对输入数据进行零填充
池化层：最常用的设置是用2x2的max-pooling，步长为2

8，为什么在卷积层使用1的步长?
在实际应用中，更小的步长效果更好。步长为1可以让空间维度的降采样全部由池化层负责，卷积层只负责对输入数据体的深度进行变换。

9，为何使用零填充?
使用零填充除了前面提到的可以让卷积层的输出数据保持和输入数据在空间维度的不变，还可以提高算法性能。如果卷积层值进行卷积而不进行零填充，那么数据体的尺寸就会略微减小，那么图像边缘的信息就会过快地损失掉。