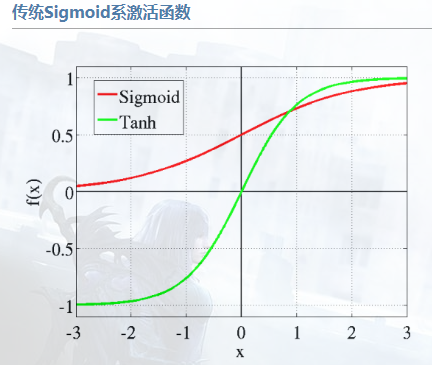
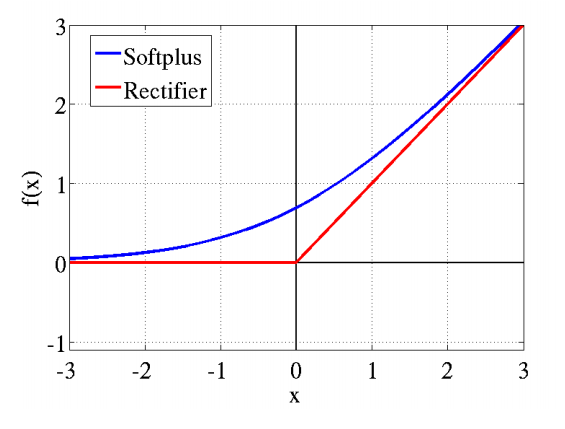
1.Sigmoid函数（1/(1+e-x)）和Tanh((1- e-x）/(1+e-x)）函数

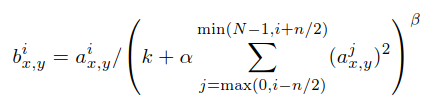


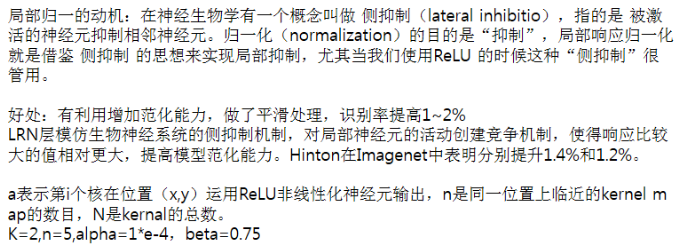
2.ReLu函数((max(0，x))比sigmoid函数训练时间更短，不需要正则化防止过拟和



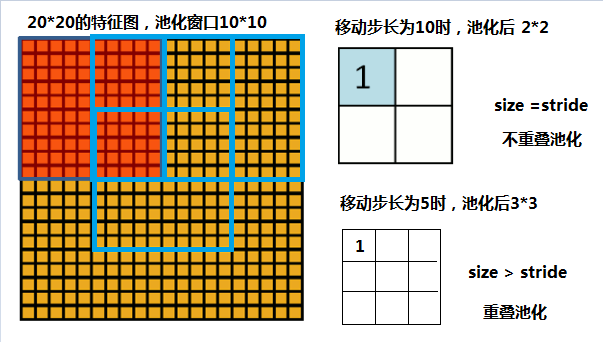
从图形可以看出，后图中的ReLU与softplus函数与前图中的传统sigmoid系激活函数相比，主要变化有三点：①单侧抑制 ②相对宽阔的兴奋边界 ③稀疏激活性（重点，可以看到红框里前端状态完全没有激活）。

3.局部响应归一化（LRN，Local Response Normalization）





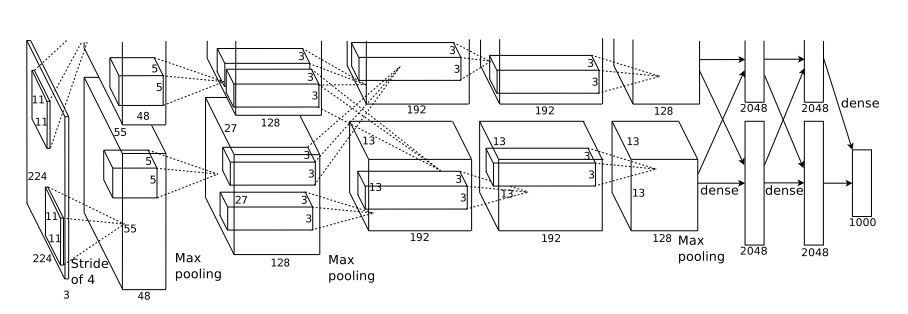
4.重叠池：有助于减少过拟和



5. Dropout

结合多个模型的预测值是减少错误的有效方法，但是对于训练时间用好几天的大型神经网络太耗费时间。Dropout是有效的模型集成学习方法，具有0.5的概率将隐藏神经元设置输出为0。运用了这种机制的神经元不会干扰前向传递也不影响后续操作。因此当有输入的时候，神经网络采样不用的结构，但是这些结构都共享一个权重。这就减少了神经元适应的复杂性。测试时，用0.5的概率随机失活神经元。dropout减少了过拟合，也使收敛迭代次数增加一倍。

6.AlexNet模型的各层结构解析



前五层是卷积层，后三层是全连接层

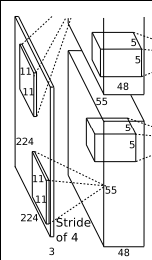
第一个和第二个卷积层后面，各有一个LRN（局部响应归一化）层

在2个LRN后还有各有一个池化层，第5层后面有一个池化层

使用Relu函数作为激活函数，对各卷积层和全连接层的输出，进行激活

**总结(双CPU运行)**

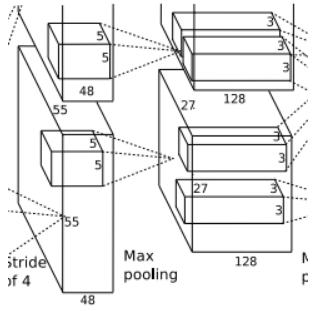
原图像 🡪 （卷积1 🡪LRN1 🡪 池化1） 🡪 （卷积2 🡪 LRN2🡪 池化2）🡪 卷积3 🡪卷积4 🡪 （卷积5 🡪 池化3）🡪 全连接1 🡪 全连接2 🡪 全连接3



第一层输入数据为原始图像的227\*227\*3的图像（最开始是224\*224\*3，为后续处理方便必须进行调整，步长4,卷积核11\*11，所以最后一个卷积核少了3个像素，4\*54=216，还剩8，比11少了3个像素），由于第一层有96个卷积核，所以就会形成55\*55\*96个像素层。系统是采用双GPU处理，因此分为2组数据：55\*55\*48的像素层数据。

局部响应归一化层(LRN)：归一化运算的尺度为5\*5，维度尺寸不变

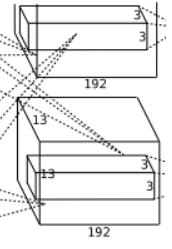
重叠池化层： 3\*3的池，运算的步长为2，则池化后的图像的尺寸为：（55-3）/2+1=27。即经过池化处理过的规模为27\*27\*48（两组）.



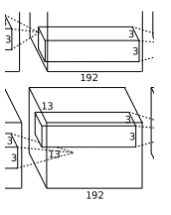
第二层输入数据为第一层输出的27\*27\*48\*2的像素层（为方便后续处理，这对每幅像素层进行像素填充），分为2组像素数据，两组像素数据分别在两个不同的GPU中进行运算。每组像素数据（27\*27\*48）被5\*5\*48的卷积核（128个每组）进行卷积运算，同理按照第一层的方式进行（步长为1）：27个像素，一共有256（2组）个卷积核（5\*5\*48），这样也就有了27\*27\*128\*2（两组）像素层。

局部响应归一化层(LRN)：归一化运算的尺度为5\*5，维度尺寸不变

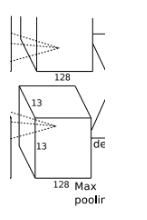
重叠pool池化层：3\*3的池，运算的步长为2，池化后的图像尺寸为（27-3）/2+1=13，即池化后像素的规模为2组13\*13\*128的像素层。



第三层输入数据为第二层输出的2组13\*13\*128的像素层（为方便后续处理，这对每幅像素层进行像素填充），每组像素数据被3\*3\*256（文章写得，个人觉得应该是128）的卷积核进行卷积运算，一共有384（每组192）个卷积核，步长为1，同理按照第一层的方式进行：（13-3+1\*2）/1+1=13个像素，这样也就有了13\*13\*192\*2（两组）像素层。

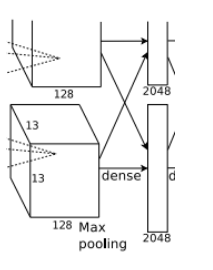


第四层输入数据为第三层输出的2组13\*13\*192的像素层（为方便后续处理，这对每幅像素层进行像素填充），每组像素数据被3\*3\*192的卷积核进行卷积运算，一共有384（每组192）个卷积核，步长为1，同理按照第一层的方式进行：（13-3+1\*2）/1+1=13个像素，这样也就有了13\*13\*192\*2（两组）像素层。



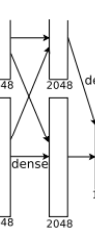
第五层输入数据为第四层输出的2组13\*13\*192的像素层（为方便后续处理，这对每幅像素层进行像素填充）。每组像素数据被3\*3\*192的卷积核进行卷积运算，一共有256（每组128）个卷积核，步长为1，同理按照第一层的方式进行：（13-3+1\*2）/1+1=13个像素，一共有256个卷积核，这样也就有了13\*13\*128\*2（两组）像素层。

重叠pool池化层：3\*3的池，运算的步长为2，池化后像素的尺寸为（13-3）/2+1=6，即池化后像素的规模变成了2组6\*6\*128的像素层，共6\*6\*256规模的像素层。

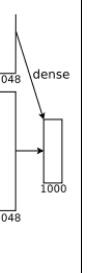


第6层输入数据的尺寸是6\*6\*128（2组），采用6\*6\*128尺寸的滤波器（卷积核）对第六层的输入数据进行卷积运算；一共4096个卷积核（2组，每组2048个），此处只生成1\*1的值，即1\*1\*2048\*2（两组）；然后通过ReLU激活函数以及dropout运算输出4096个本层的输出结果值。

很明显在第6层中，采用的滤波器的尺寸（6\*6\*256）和待处理的feature map的尺寸（6\*6\*256）相同，即滤波器中的每个系数只与feature map中的一个像素值相乘；而采用的滤波器的尺寸和待处理的feature map的尺寸不相同，每个滤波器的系数都会与多个feature map中像素相乘。因此第6层被称为全连接层。



第6层输出的4096个数据与第7层的4096个神经元进行全连接，然后经由ReLU和Dropout进行处理后生成4096个数据。



第7层输入的4096个数据与第8层的1000个神经元进行全连接，经过训练后输出被训练的数值。