#### ImageNet Classification with Deep Convolutional Neural Networks

**Abstract**

训练的是ImageNet的子集LSVRC-2010，1000个类别，1.2百万的数据量。

在测试集上top1-top5的错误率为37.5%，17.0%

网络有60百万的参数和650000个神经单元。网络包含5个卷机层，某些卷机后面有池化层，三个全联接层，最后是一个1000维的全联接层。

我们使用了GPU，为了解决在全联接层的过拟合的情况，使用了dropout的正则化方法。

在ILSVRC-2012上赢得冠军，top-5的错误率15.3%，相比第二名最好的错误率26.2%。

**Introduciton**

小数据集包括（NORB, Caltech-101/256, cigar-10/100, MNIST）

大数据集包括（LabelMe, ImageNet）

网络的特点在第三部分进行描述。尽管数据量很大，但是为了克服模型的过拟合，设计了dropout在第四部分描述。

模型的卷机层的参数不超过全部参数的1%，但是移除一个结果就会很差。

在GTX 580两卡3GB GPU上运行，需要6天。

**DataSet**

ImageNet的一个子集

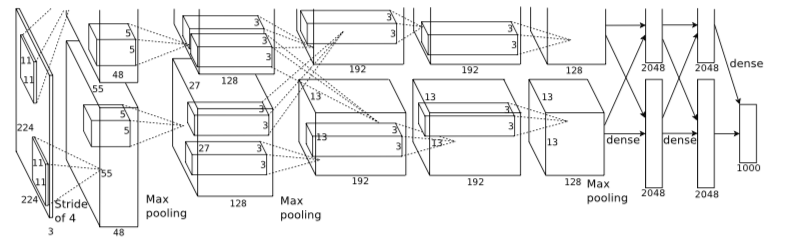
**The Architecture**

Relu：max(0, x)激活函数，比 速度快。

GPU：多卡GPU运算，之前一张显卡内存不够，需要将网络拆成两个部分一起训练。

局部响应归一化：

池化层



**详细细节**

1. 第1，2，5层卷积层后面接池化层。
2. 第2，4，5层与同一GPU前面的一层连接
3. relu 接在卷积层，全联接层的后面。

**过拟合**

采用数据增强

dropout方法，全联接层的前面两层使用。

Mxnet版本的网络构建：

net = nn.Sequential()

net.add(nn.Conv2D(96, kernel\_size=11, strides=4, activation='relu'),

nn.MaxPool2D(pool\_size=3, strides=2),

nn.Conv2D(256, kernel\_size=5, padding=2, activation='relu'),

nn.MaxPool2D(pool\_size=3, strides=2),

nn.Conv2D(384, kernel\_size=3, padding=1, activation='relu'),

nn.Conv2D(384, kernel\_size=3, padding=1, activation='relu'),

nn.Conv2D(256, kernel\_size=3, padding=1, activation='relu'),

nn.MaxPool2D(pool\_size=3, strides=2),

nn.Dense(4096, activation='relu'), nn.Dropout(0.5),

nn.Dense(4096, activation='relu'), nn.Dropout(0.5),

nn.Dense(1000)

)

