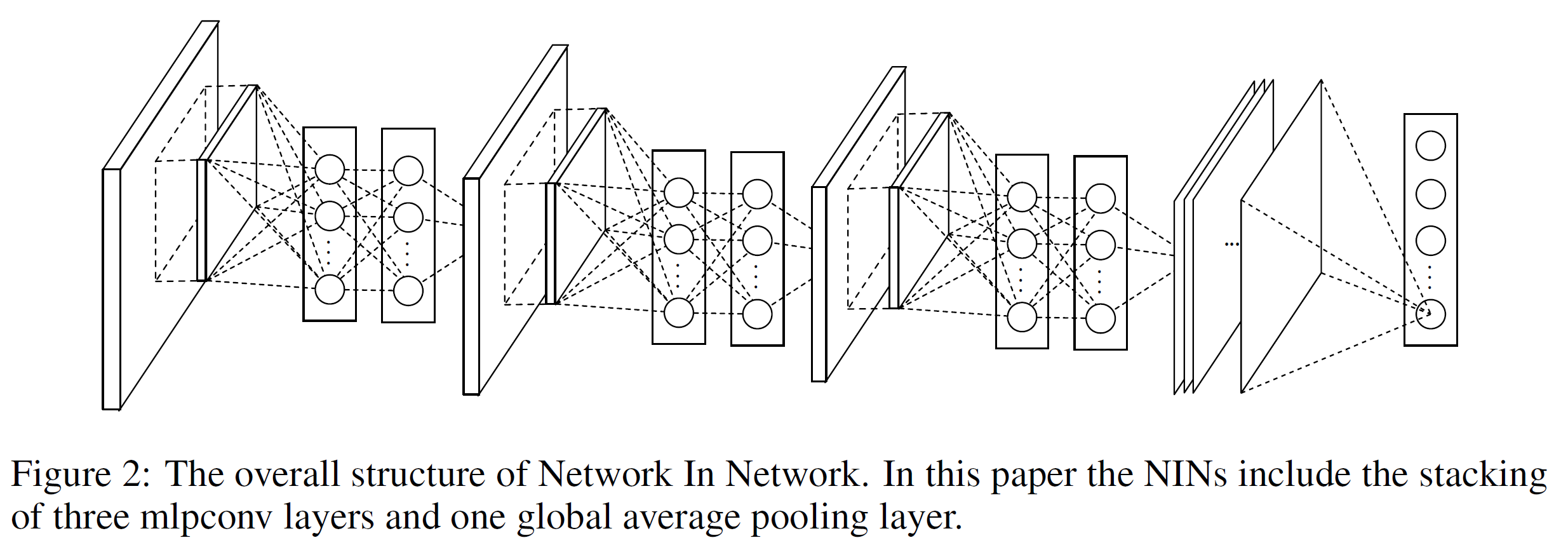
NIN Notes

1. Architecture



* 一般来说，如果要提取的一些潜在的特征是线性可分的话，那么对于线性的卷积运算来说就足够了。然而我们所要提取的特征一般是高度非线性的。在传统的CNN中，也许我们可以用超完备的滤波器，来提取各种潜在的特征。比如我们要提取某个特征，于是我就用了一大堆的滤波器，把所有可能的提取出来，这样就可以把我想要提取的特征也覆盖到，然而这样存在一个缺点，那就是网络太恐怖了，参数太多了。
* 我们知道CNN高层特征其实是低层特征通过某种运算的组合。于是作者就根据这个想法，提出在每个局部感受野中进行更加复杂的运算，提出了对卷积层的改进算法：MLP卷积层。另一方面，传统的CNN最后一层都是全连接层，参数个数非常之多，容易引起过拟合，一个CNN模型，大部分的参数都被全连接层给占用了，故这篇paper采用了全局均值池化，替代全连接层。

1. MLP卷积层

* 本文中用到的网络结构是三个MLP卷积层+一个全局平均池化层，可以从上图中看出来。
* 下面代码是一个MLP卷积层的定义

1. # conv1 layer
2. with tf.name\_scope('conv1'):
3. W\_conv1 = weight\_variable([5, 5, 3, 192], stddev=0.01)
4. b\_conv1 = tf.Variable(tf.random\_normal(
5. [192], stddev=0.01, type=tf.float32))
6. output = tf.nn.relu(conv2d(x\_image, W\_conv1) + b\_conv1)
7. **print**('conv1 output: ', output.shape)   # [None, 32, 32, 3]
8. tf.summary.histogram('conv\_filter', output)
9. tf.summary.scalar('conv\_filter', tf.nn.zero\_fraction(output))
10. # MLP-1-1
11. with tf.name\_scope('mlp\_1\_1'):
12. W\_MLP11 = weight\_variable([1, 1, 192, 160])
13. b\_MLP11 = bias\_variable([160])
14. output = tf.nn.relu(conv2d(output, W\_MLP11) + b\_MLP11)
15. **print**('MLP-1-1 output: ', output.shape)     # [None, 32, 32, 160]
16. tf.summary.histogram('mlp', output)
17. tf.summary.scalar('mlp', tf.nn.zero\_fraction(output))
18. # MLP-1-2
19. with tf.name\_scope('mlp\_1\_2'):
20. W\_MLP12 = weight\_variable([1, 1, 160, 96])
21. b\_MLP12 = bias\_variable([96])
22. output = tf.nn.relu(conv2d(output, W\_MLP12) + b\_MLP12)
23. **print**('MLP-1-2 output: ', output.shape)     # [None, 32, 32, 96]
24. tf.summary.histogram('mlp', output)
25. tf.summary.scalar('mlp', tf.nn.zero\_fraction(output))
27. with tf.variable\_scope('Visualization'):
28. grid = put\_kernels\_on\_grid(W\_conv1)
29. tf.summary.image('conv1/filters', grid, max\_outputs=1)
31. # Max pooling
32. output = max\_pool\_3x3(output)
33. # dropout
34. output = tf.nn.dropout(output, keep\_prob)
35. Global Average Pooling

* 代码如下

1. # global average
2. output = tf.nn.avg\_pool(output, ksize=[1, 8, 8, 1], strides=[1, 1, 1, 1], padding='VALID')