在知乎看到一篇挺好的外网翻译来的卷积神经网络的介绍，整理了一下

By:ChenYongLin

原贴地址：知乎 CNN（卷积神经网络）是什么？有何入门简介或文章吗？作者：机器之心https://www.zhihu.com/question/52668301/answer/131573702

### 一、卷积神经网络简介

**概况**

卷积神经网络在计算机视觉领域非常具有影响力。在很多地方都有用途。脸书用于标注算法，谷歌用于图片搜索，亚马逊用于商品推荐，Ins用于搜索架构等

最流行用于图像处理中的图像分类

**大概流程**

图像分类就是对输入图像进行操作，最终输出一组分类的图像或者分类的概率。

具体来说，提供三通道的图片，计算机输出某一特定分类的概率数字

**如何执行**

需要计算机提取出图片的特征，如何提取，一些明显的特点（比如猫狗的爪子和四条腿）或者寻找边缘或者曲线等低级特点。再通过一系列“卷积”层级构造更抽象的玩意儿。

**卷积神经网络（以下简称CNN）的来源**

从生物学和神经学来，人体视觉皮层中的部分细胞对一些特定方向的边缘做出反应，并且通过协同工作，使得人体的视觉产生感知。

**CNN的结构**

输入、卷积层、非线性层、池化层、完全连接层、输出

### 二、卷积层

**浅显的看卷积层**

从图片中，选取一定的大小，在原图中滑动，映射到新的图像中。

可以比喻成一个手电筒照亮着输入所有区域。手电筒照亮的区域叫感受区，手电筒是过滤器（Filter）

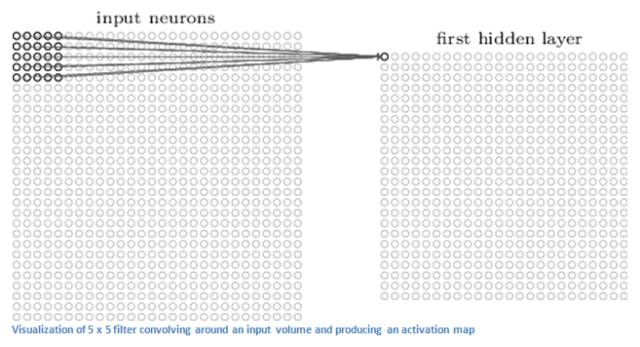
过滤器（Filter）也可以叫神经元（Neuron）或 核 （Kernel）

感受区（Receptive field）：被照过的区域

重要的地方：过滤器的深度（维度）必须和输入内容的深度相同

过滤器的行为：过滤器的值与图像中的原始像素值相乘，结果总和得到的数字放在新的结果层(卷积层)上。

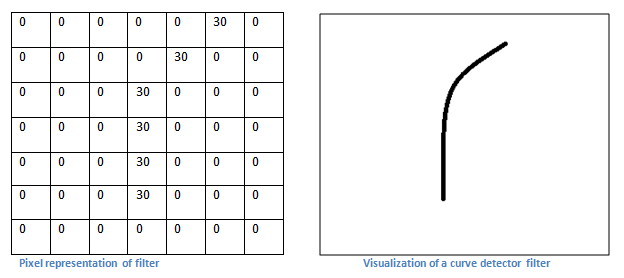
当过滤器滑动到最后的位置，将得到一个新的数组。这个过程叫做激活映射(activation map)或特征映射(feature map)



**再深一层次地看卷积层**

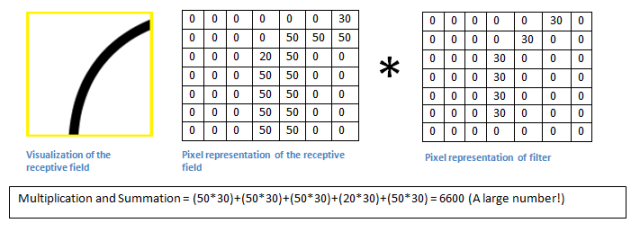
过滤器可以看成是一个特征标识符

这里的特征比如是图像中的边缘、色彩、曲线之类的东西

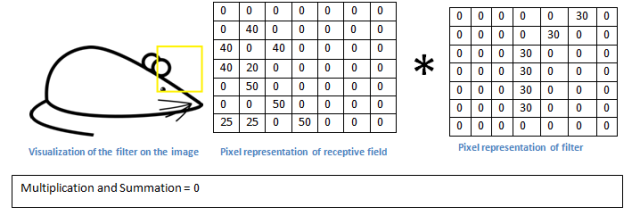


例如右图是一个曲线，左图则是此曲线的特征标识符

然后过滤器（即是每一个特征标识符）与原图像素值之间进行点积



如果原图所选取的区域也是一个右转曲线，则会得到一个很大的值。



如果原图选取的区域不相符，则会得到一个很小的值

也就是说得出的点积，就是激活映射结果中那个像素的值。

如何更好地理解激活映射？

激活映射的值如果很高，那就是说原图中输入内容激活了过滤器。也就是说与特征标识符的内容匹配了。

上图给的例子仅仅是一个右转的曲线的过滤器，当然还有很多其他特征的过滤器，过滤器越多，激活映射的深度越大，对输入内容的了解度也就更多

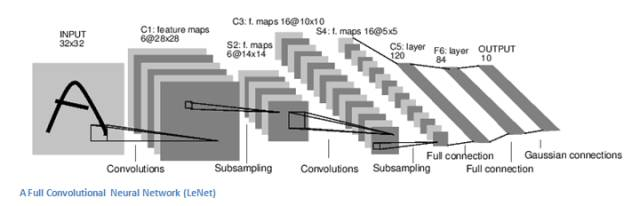
**卷积层的更多**

在传统CNN中，卷积层之间还有其他的层，来提供非线性的维度保留，有助于提高网络的鲁棒性，并控制过拟合。

一个典型的CNN结构看起来是这样的：



输入→卷积→ReLU→卷积→ReLU→池化→ReLU→卷积→ReLU→池化→全连接



第一层一般过滤一些低级的特征。

第二层卷积的输入一般是第一层卷积输出的激活映射。

第二层再经过一些过滤器，输出更高级的特征的激活映射

在例子中，第二层可以是半圆（曲线和直线的组合），或者是四边形（几条直线的组合）。

越深入网络，过滤器的感受区越大。也就是可以更大范围的处理输入内容。

随着网络的递进以及更多的卷积层，会得到更加复杂的激活映射。

### 三、完全连接层

这一层的输入将会是卷积层或者ReLU层或者池化层的输出。

完全连接层的输出将会是一个N维向量。每一维度代表着一个特定类别的概率。

至于完全连接层，要观察上一层的输出的一些高级特征，然后根据一定的计算，得到不同的正确概率

### 四、如何训练CNN

第一层卷积层如何寻找边缘与曲线？连接层如何观察激活图？每一层的滤波器要知道哪些值？

这些数值通过反向传播训练来调整过滤值或权重

**反向传播四个部分**

前向传导、损失函数、后向传导、以及权重更新

**反向传播如何进行**

比喻刚出生的人类，啥也不懂。

权重和过滤器的值都是随机的。但是随着年龄的增长，老师和父母跟我们介绍各种各样的图片（做出标记） CNN经历的就是一个介绍图片与分类标记的训练过程。

也就是说在训练之前，要先设定一个训练集，训练集之中有很多的猫狗的照片，每一张都被标记。

**1.前向传导**

前向传导中，选择一张图片，并让他通过整个网络。因为初始的时候权重都是随机的，那样的网络做不出合理的分类。

**2.损失函数**

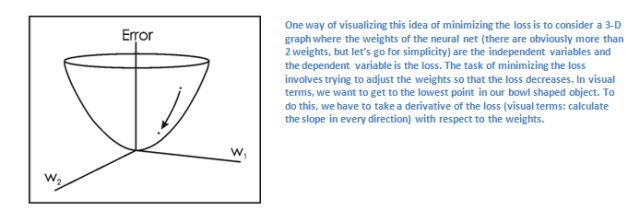
下面拿一种例子，均方误差（MSE）的损失函数



假设变量L是该值，前两张图片训练的损失会非常高。而为了达成目的，我们要做到损失函数最小化。

将其视为微积分问题，找出哪一部分的输入（权值）导致了网络的损失

下图是一个dL/dW 的数学等式，W是特定层的权重。



**3.向后传导**

接下来要做的就是网络要进行向后传导，测试哪一部分导致了最大的损失，并且找到调整损失的方法。计算出该导数。

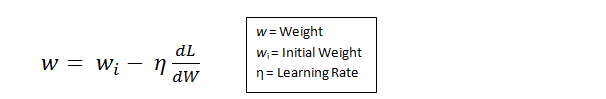
**4.权重更新**

根据上面计算的导数，进行权重的更新。

这样的话所有的过滤器就会顺着梯度方向改变

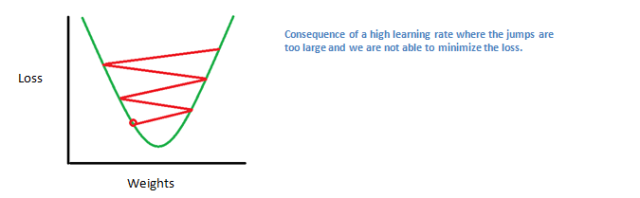
**学习速率**

权值=旧权值-学习速率（dl/dW）



学习速率是程序员决定的，高速率会导致更新的动作很大

这样的话花费的时间比较少，但是有可能达不到最优点



**总结**

总的来说，前向传导、损失函数、后向传导、参数更新称为一个学习的周期

每一张训练的图片，程序会重复固定数目的周期过程，一旦完成了最后样本的参数更新，则该网络差不多就成型了。

### 五、CNN运用

数据越多，那么网络训练的迭代次数就越多，权值更新越多，调整出的网络就会越好。