

雨石

关注程序语言、算法、数据结构，兴趣方向：机器学习，NLP

目录视图

摘要视图

RSS 订阅

个人资料



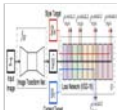
张雨石



访问： 737980次  
积分： 6986  
等级： **BLOG > 6**  
排名： 第2960名

原创： 81篇  
转载： 4篇  
译文： 4篇  
评论： 494条

博客专栏



深度学习  
文章： 15篇  
阅读： 240826



机器学习笔记  
文章： 15篇  
阅读： 129029

博客公告

Email : stdcoutzyx@163.com  
weibo: zyx\_1991  
非技术博客: 以梦为马

文章分类

机器学习笔记 (15)  
机器学习算法 (21)  
图像处理 (2)  
自然语言处理 (4)  
编程&&工具 (9)  
论文笔记 (27)

【活动】2017 CSDN博客专栏评选    【评论送书】5月书讯：流畅的Python    CSDN日报20170531 ——《这个架构能实现吗？》    CSDN 日报 | 4.19-5.19 上榜作者排行出炉

## 卷积神经网络

标签： 卷积神经网络 CNN

2014-11-29 16:20    98614人阅读    评论(49)    收藏    举报

本文章已收录于： 人工智能机器学习知识库

分类： 机器学习算法 (20)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)    [+]

## 卷积神经网络

转载请注明：http://blog.csdn.net/stdcoutzyx/article/details/41596663

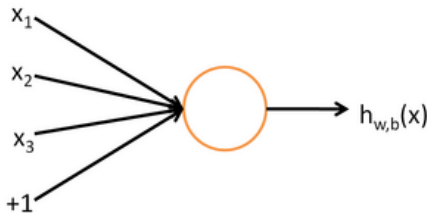
自今年七月份以来，一直在实验室负责卷积神经网络（Convolutional Neural Network，CNN），期间配置和使用过theano和cuda-convnet、cuda-convnet2。为了增进CNN的理解和使用，特写此博文，以与其人交流，互有增益。正文之前，先说几点自己对于CNN的感触。先明确一点就是，Deep Learning是全部深度学习算法的总称，CNN是深度学习算法在图像处理领域的一个应用。

- 第一点，在学习Deep learning和CNN之前，总以为它们是很了不得的知识，总以为它们能解决很多问题，学习了之后，才知道它们不过与其他机器学习算法如svm等相似，仍然可以把它当做一个分类器，仍然可以像使用一个黑盒子那样使用它。
- 第二点，Deep Learning强大的地方就是可以利用网络中间某一层的输出当做是数据的另一种表达，从而可以将其认为是经过网络学习到的特征。基于该特征，可以进行进一步的相似度比较等。
- 第三点，Deep Learning算法能够有效的关键其实是大规模的数据，这一点原因在于每个DL都有众多的参数，少量数据无法将参数训练充分。

接下来话不多说，直奔主题开始CNN之旅。

## 1. 神经网络

首先介绍神经网络，这一步的详细可以参考资源1。简要介绍下。神经网络



关闭

百度云

云计算新用户

注册送520元代金券

立即领取

想法&&读书&&随记 (16)

技术博客链接  
我爱机器学习  
计算机视觉  
我爱自然语言处理  
结构之法 算法之道  
LeftNotEasy  
Jerrylead

- 阅读排行
- 卷积神经网络 (98340)
  - DeepID人脸识别算法之 (41966)
  - 隐马尔科夫模型 (HMM) (26065)
  - tensorflow架构 (23601)
  - 理解dropout (23527)
  - FaceNet--Google的人脸 (18100)
  - 线性规划、梯度下降、正 (16338)
  - DeepID算法实践 (15339)
  - DeepID2——强大的人脸 (15215)
  - 生成学习、高斯判别、朴 (14995)

- 文章存档
- 2017年04月 (3)
  - 2017年03月 (1)
  - 2017年02月 (1)
  - 2017年01月 (2)
  - 2016年12月 (3)
- 展开

百度云

云计算新用户

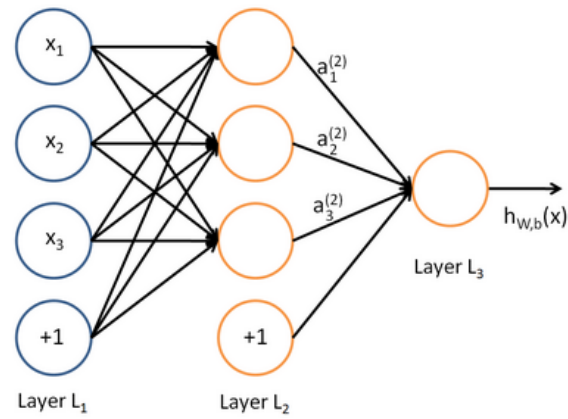
注册送520元代金券

立即领取

其对应的公式如下：

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^3 W_i x_i + b)$$

其中，该单元也可以被称作是Logistic回归模型。当将多个单元组合起来并具有分层结构时，就形成了神经网络模型。下图展示了一个具有一个隐含层的神经网络。



其对应的公式如下：

$$\begin{aligned} a_1^{(2)} &= f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)}) \\ a_2^{(2)} &= f(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)}) \\ a_3^{(2)} &= f(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)}) \\ h_{W,b}(x) &= a_1^{(3)} = f(W_{11}^{(2)} a_1^{(2)} + W_{12}^{(2)} a_2^{(2)} + W_{13}^{(2)} a_3^{(2)} + b_1^{(2)}) \end{aligned}$$

比较类似的，可以拓展到有2,3,4,5, ...个隐含层。

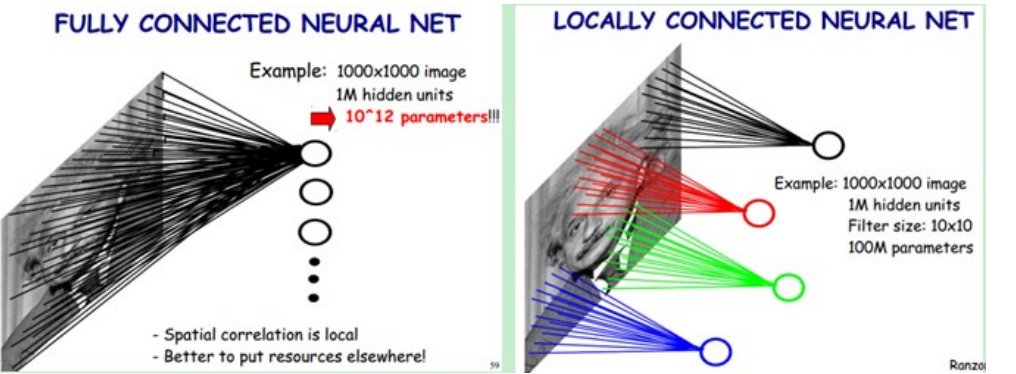
神经网络的训练方法也同Logistic类似，不过由于其多层性，还需要利用链式求导法则对隐含层的节点进行求导，即梯度下降+链式求导法则，专业名称为反向传播。关于训练算法，本文暂不涉及。

2 卷积神经网络

在图像处理中，往往把图像表示为像素的向量，比如一个1000×1000的图像，可以表示为一个1000000的向量。在上一节中提到的神经网络中，如果隐含层数目与输入层一样，即也是1000000时，那么输入层到隐含层的参数数据为1000000×1000000=10<sup>12</sup>，这样就太多了，基本没法训练。所以图像处理要想练成神经网络大法，必先减少参数加快速度。就跟辟邪剑谱似的，普通人练得很挫，一旦自宫后内力变强剑法变快，就变的很牛了。

2.1 局部感知

卷积神经网络有两种神器可以降低参数数目，第一种神器叫做局部感知野。一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息。网络部分连通的思想，也是受启发于生物学里面的视觉系统结构。视觉皮层的神经元就是局部接受信息的（即这些神经元只响应某些特定区域的刺激）。如下图所示：左图为全连接，右图为局部连接。



在上右图中，假如每个神经元只和10×10个像素值相连，那么权值数据为1000000×100个参数，减少为原来的万分之一。而那10×10个像素值对应的10×10个参数，其实就相当于卷积操作。

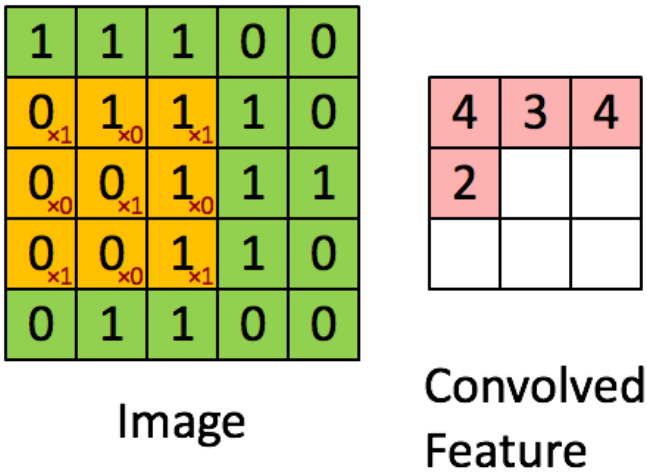
2.2 参数共享

但其实这样的话参数仍然过多，那么就启动第二级神器，即权值共享。在上面的局部连接中，每个神经元都对应100个参数，一共1000000个神经元，如果这1000000个神经元的100个参数都是相等的，那么参数数目就变为100了。

怎么理解权值共享呢？我们可以这100个参数（也就是卷积操作）看成是提取特征的方式，该方式与位置无关。这其中隐含的原理则是：图像的一部分的统计特性与其他部分是一样的。这也意味着我们在这一部分学习的特征也能用在另一部分上，所以对于这个图像上的所有位置，我们都能使用同样的学习特征。

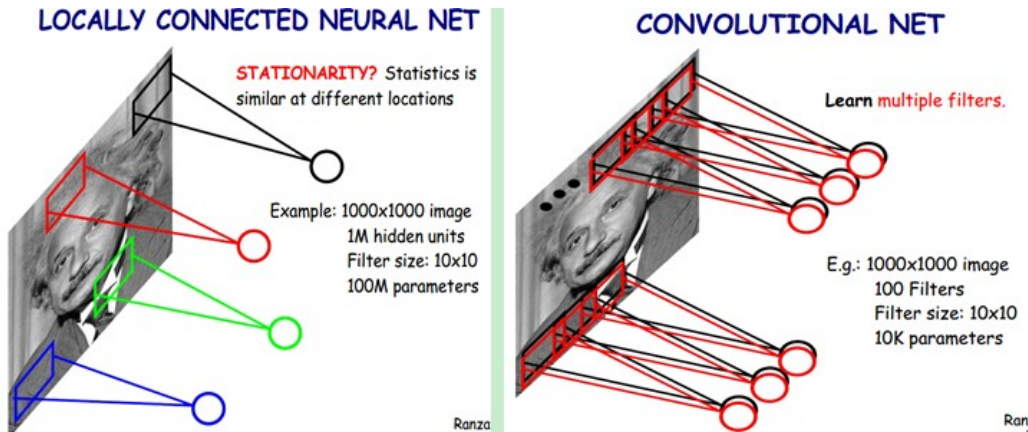
更直观一些，当从一个大尺寸图像中随机选取一小块，比如说 8x8 作为样本，并且从这个小块样本中学习到了的一些特征，这时我们可以把从这个 8x8 样本中学习到的特征作为探测器，应用到这个图像的任意地方中去。特别是，我们可以用从 8x8 样本中所学习到的特征跟原本的大尺寸图像作卷积，从而对这个大尺寸图像上的任一位置获得一个不同特征的激活值。

如下图所示，展示了一个3x3的卷积核在5x5的图像上做卷积的过程。每个卷积都是一种特征提取方式，就像一个筛子，将图像中符合条件（激活值越大越符合条件）的部分筛选出来。



2.3 多卷积核

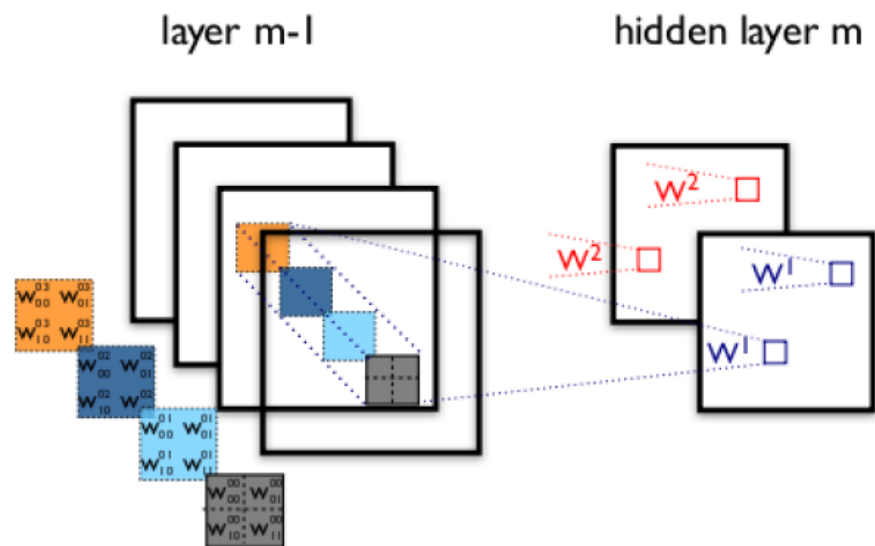
上面所述只有100个参数时，表明只有1个10x10的卷积核，显然，特征提取是不充分的，我们可以添加多个卷积核，比如32个卷积核，可以学习32种特征。在有多多个卷积核时，如下图所示：



上图右，不同颜色表明不同的卷积核。每个卷积核都会将图像成为另一幅图像。比如两个卷积核就可以将生成两幅图像，这两幅图像可以看做是一张图像的不同通道。如下图所示，下图有个小错误，即将w1改为w0，w2改为w1即可。下文中仍以w1和w2称呼它们。

下图展示了在四个通道上的卷积操作，有两个卷积核，生成两个通道。其中需要注意的是，四个通道上每个通道对应一个卷积核，先将w2忽略，只看w1，那么在w1的某位置 (i,j) 处的值，是由四个通道上 (i,j) 处的卷积结果相加然后再取激活函数值得到的。

$$h_{ij}^k = \tanh((W^k * x)_{ij} + b_k)$$

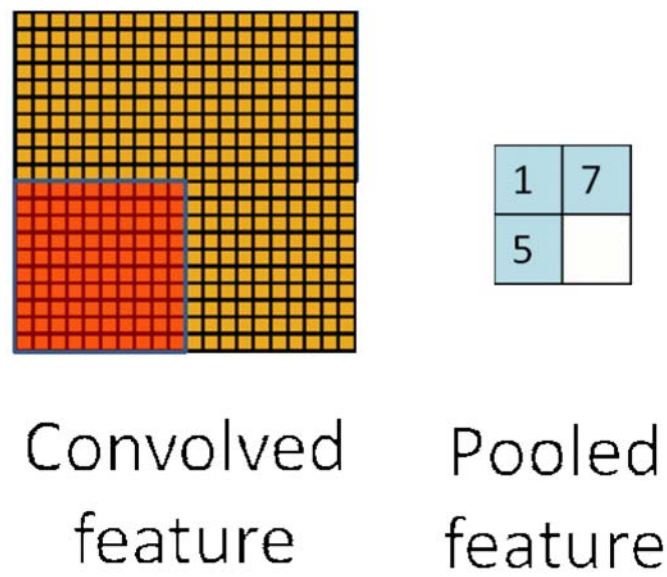


所以，在上图由4个通道卷积得到2个通道的过程中，参数的数目为 $4 \times 2 \times 2 \times 2$ 个，其中4表示4个通道，第一个2表示生成2个通道，最后的 $2 \times 2$ 表示卷积核大小。

### 2.4 Down-pooling

在通过卷积获得了特征 (features) 之后，下一步我们希望利用这些特征去做分类。理论上讲，人们可以用所有提取得到的特征去训练分类器，例如 softmax 分类器，但这样做面临计算量的挑战。例如：对于一个  $96 \times 96$  像素的图像，假设我们已经学习得到了400个定义在 $8 \times 8$ 输入上的特征，每一个特征和图像卷积都会得到一个  $(96 - 8 + 1) \times (96 - 8 + 1) = 7921$  维的卷积特征，由于有 400 个特征，所以每个样例 (example) 都会得到一个  $7921 \times 400 = 3,168,400$  维的卷积特征向量。学习一个拥有超过 3 百万特征输入的分类器十分不便，并且容易出现过拟合 (over-fitting)。

为了解决这个问题，首先回忆一下，我们之所以决定使用卷积后的特征是因为图像具有一种“静态性”的属性，这也就意味着在一个图像区域有用的特征极有可能在另一个区域同样适用。因此，为了描述大的图像，一个很自然的想法就是对不同位置的特征进行聚合统计，例如，人们可以计算图像一个区域上的某个特定特征的平均值 (或最大值)。这些概要统计特征不仅具有低得多的维度 (相比使用所有提取得到的特征)，同时还会改善结果 (不容易过拟合)。这种聚合的操作就叫做池化 (pooling)，有时也称为平均池化或者最大池化 (取决于计算池化的方法)。



至此，卷积神经网络的基本结构和原理已经阐述完毕。

### 2.5 多层卷积

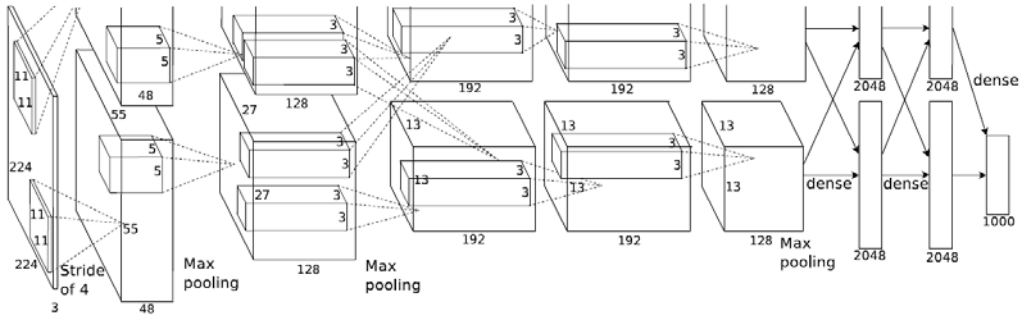
在实际应用中，往往使用多层卷积，然后再使用全连接层进行训练，多层卷积的目的是一层卷积学到的特征往往是局部的，层数越高，学到的特征就越全局化。

## 3 ImageNet-2010网络结构



ImageNet LSVRC是一个图片分类的比赛，其训练集包括127W+张图片，验证集有5W张图片，测试集有15W张图片。本文截取2010年Alex Krizhevsky的CNN结构进行说明，该结构在2010年取得冠军，top-5错误率为15.3%。值得一提的是，在今天的ImageNet LSVRC比赛中，取得冠军的GoogNet已经达到了top-5错误率6.67%。可见，深度学习的提升空间还很巨大。

下图即为Alex的CNN结构图。需要注意的是，该模型采用了2-GPU并行结构，即第1、2、4、5卷积层都是将模型参数分为2部分进行训练的。在这里，更进一步，并行结构分为数据并行与模型并行。数据并行是指在不同的GPU上，模型结构相同，但将训练数据进行切分，分别训练得到不同的模型，然后再将模型进行融合。而模型并行则是，将若干层的模型参数进行切分，不同的GPU上使用相同的数据进行训练，得到的结果直接连接作为下一层的输入。



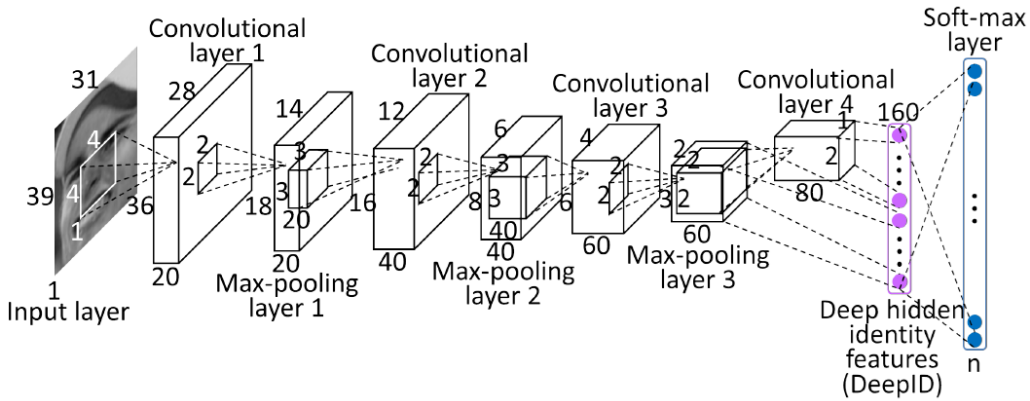
上图模型的基本参数为：

- 输入：224×224大小的图片，3通道
- 第一层卷积：11×11大小的卷积核96个，每个GPU上48个。
- 第一层max-pooling：2×2的核。
- 第二层卷积：5×5卷积核256个，每个GPU上128个。
- 第二层max-pooling：2×2的核。
- 第三层卷积：与上一层是全连接，3×3的卷积核384个。分到两个GPU上个192个。
- 第四层卷积：3×3的卷积核384个，两个GPU各192个。该层与上一层连接没有经过pooling层。
- 第五层卷积：3×3的卷积核256个，两个GPU上个128个。
- 第五层max-pooling：2×2的核。
- 第一层全连接：4096维，将第五层max-pooling的输出连接成为一个一维向量，作为该层的输入。
- 第二层全连接：4096维
- Softmax层：输出为1000，输出的每一维都是图片属于该类别的概率。

## 4 DeepID网络结构

DeepID网络结构是香港中文大学的Sun Yi开发出来用来学习人脸特征的卷积神经网络。每张输入的人脸被表示为160维的向量，学习到的向量经过其他模型进行分类，在人脸验证试验上得到了97.45%的正确率，更进一步的，原作者改进了CNN，又得到了99.15%的正确率。

如下图所示，该结构与ImageNet的具体参数类似，所以只解释一下不同的部分吧。



上图中的结构，在最后只有一层全连接层，然后就是softmax层了。论文中就是以该全连接层作为图像的代表。在全连接层，以第四层卷积和第三层max-pooling的输出作为全连接层的输入，这样可以学习到局部的和全局的特征。

5 参考资源

- [1] <http://deeplearning.stanford.edu/wiki/index.php/UFLDL%E6%95%99%E7%A8%8B> 梶子花对Stanford深度学习研究团队的深度学习教程的翻译
- [2] <http://blog.csdn.net/zouxy09/article/details/14222605> csdn博主zouxy09深度学习教程系列
- [3] <http://deeplearning.net/tutorial/theano实现deep learning>
- [4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
- [5] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014: 1891-1898.

顶 82 踩 4

上一篇 DeepID2——强大的人脸分类算法  
下一篇 北美公司面试经验笔记

相关文章推荐

- deeplearning4j之卷积神经网络实现
- 卷积神经网络(CNN)及其theano实现
- VS2013配置Caffe卷积神经网络工具（64位Windows...
- Spark MLlib Deep Learning Convolution Neural Net...
- 卷积神经网络cnn的个人白话总结
- 卷积神经网络CNN初解
- 卷积神经网络（CNN）的细节问题（滤波器的大小选...
- 初学卷积神经网络，基础知识与概念
- 卷积神经网络学习笔记
- Top100论文导读：深入理解卷积神经网络CNN（Par...



德国阳光蓄电



学习平面设计



excel进销存



亚瑟士跑鞋



三国策略类游戏



限时特卖



学习平面设计

参考知识库

猜你在找

- 《C语言/C++学习指南》加密解密篇（安全相关算法）

数据结构与算法在实战项目中的应用

使用决策树算法对测试数据进行分类实战

C语言系列之 字符串压缩算法与结构体初探

C语言系列之 递归算法示例与 Windows 趣味小项目
- 高性能缓存Memcached服务深度原理及实战

使用决策树算法对测试数据进行分类实战

模板匹配的字符识别(OCR) 算法原理

C语言系列之 字符串相关算法

C语言系列之 数组与算法实战



美工学习



智能水表



安全公开课




英语学习英语



酷睿i8处理器

查看评论

33楼 EricaWu 2017-05-11 11:12发表



写得很好，但是新手刚刚开始学习，希望能和您交流一些问题

32楼 wanzi\_antang 2017-04-18 14:58发表



受益匪浅

31楼 天易飞鹰 2017-04-11 18:29发表



受益匪浅

30楼 baidu\_37612361 2017-02-22 11:36发表



由于有 400 个特征，所以每个样例 (example) 都会得到一个  $892 \times 400 = 3,168,400$  维的卷积特征向量。学习一个拥有超过 3 百万特征输入的分类器十分不便，并且容易出现过拟合 (over-fitting)。

892

29楼 baidu\_37612361 2017-02-22 11:35发表



由于有 400 个特征，所以每个样例 (example) 都会得到一个  $892 \times 400 = 3,168,400$  维的卷积特征向量。学习一个拥有超过 3 百万特征输入的分类器十分不便，并且容易出现过拟合 (over-fitting)。

892肯定不对啊

28楼 Painting时光 2017-01-14 21:01发表



224\*224经过11\*11的卷积和2\*2的滤波能得到55\*55吗？感觉不太对啊

27楼 米克米不克 2017-01-13 10:20发表



老哥是北航的啊！！

26楼 R2D2pp 2016-11-24 22:55发表



3 ImageNet-2010网络结构的分析有误，参考论文 ImageNet Classification with Deep Convolutional Neural Networks。第一步，stride of 4是步长，224\*224的图片经过 (11,11) 的Kernel卷积（步长为4的情况），出来的结果应该是55\*55。理论上 (11,11) 的Kernel步长为4, 55\*55变换前图像理论上应该是  $55 \times 4 + 7 = 227$ ，而54\*54为223，原图是224的。这里不知道为何这样设置。这样会导致最后一步步长为1（如果前边都为4的话）

Re: R2D2pp 2016-11-24 23:11发表



回复sinat\_36229016：找到了，这里应该是原文英文paper的问题，第一步是227\*227的输入，(11,11) 的Kernel，步长为3，结果出来就是55\*55，不需要池化。可以参考源代码。《深入理解AlexNet》里给出了源代码，可以参考下。

Re: qq\_30593987 2017-03-22 16:37发表



回复sinat\_36229016：第一层卷积：11\*11大小的卷积核96个，请问一下这个96是怎么得到的啊

25楼 少年张翠山 2016-09-23 23:21发表



最近机器学习很火，看到你的技术文章，也来了，虽不明，但觉厉。

24楼 RachelRn 2016-09-11 14:48发表



good

23楼 RachelRn 2016-09-11 14:41发表



大牛，我在Github下载的DeepLearnToolbox工具箱，里面的CNN只能输入灰度图，我该怎么输入N\*N\*3的彩色图呢，看您博客中说输入224\*224的3个通道的，是输入的彩色图吗？期待大牛的回复

22楼 weixin\_36032111 2016-09-02 14:43发表



博主，请问是在G里做tensorflow吗？有问题想请教，坐标硅谷。多谢了

21楼 zuglerfu 2016-07-29 16:48发表



有一点问题，3 ImageNet-2010网络结构中第五层max-pooling：2\*2的核的话，算出来结果和2048不符合，是否是3\*3的呢？

20楼 qq\_19479035 2016-06-30 19:48发表



2.3 和3里的通道含义不一样

19楼 maolijuan 2016-05-30 09:55发表



大牛，想跟你沟通下，能加你QQ吗？

18楼 重邮故地 2016-05-23 17:24发表

楼主写的很好，但是2.3倒数第二行“参数的数目为 $4 \times 2 \times 2 \times 2$ 个”有错误，参数应该是“ $2 \times 2 \times 2$ ”，参数的个数与上一层的feature



map无关。

Re: yuirel媛 2016-09-08 09:31发表



回复u012505395: 博主说的是对的。2.3的第m-1层上有4个通道，他们对应两个卷积核，每个卷积核大小为4维的2\*2，（这里的卷积核是立体的，不是平面的小窗！）卷积核的每个维度分别与第m-1层的每个通道做内积，所得的4个值相加再加上bias就得到了第m层的一个单元的值。所以参数数目是2个卷积核\*(4维\*2\*2)=32

Re: 张雨石 2016-06-30 00:34发表



回复u012505395: 不，是有关系的，在上一层的feature\_map上，每个feature\_map上都会对应一个卷积核，各个卷积核与对应的feature\_map会生成卷积后的新矩阵，然后这些矩阵再加权聚合。形成下一层的一个feature\_map。所以下一层的一个feature\_map，会用到 上层feature\_map数目 \* 卷积核面积 个参数。

Re: 我师父微微 2017-03-19 22:31发表



回复xinzhangyanxiang: 这样岂不是说每个通道对应的2个卷积核各不相同，其实是4通道8种卷积核

Re: 立冬winter 2016-08-17 11:02发表



回复xinzhangyanxiang: 博主说得对，但是这里已经假定为两个卷积核了，因此的确跟上一层的参数无关了吧。

Re: zjsyhsI 2016-05-24 17:57发表



回复u012505395: 赞同，觉得你说的对

Re: zjsyhsI 2016-05-24 17:56发表



回复u012505395: 赞同

17楼 ttanggqichao 2016-05-02 17:31发表



讲得不错，话说深度学习核心算法大概是哪几个？博主总结过么？

16楼 ofooo 2016-03-22 16:44发表



感谢，写得好！

15楼 qq\_34147555 2016-03-02 19:42发表



博主，第一层卷积：11\*11大小的卷积核96个，每个GPU上48个，96是如何得到的？

14楼 星夜落尘 2016-01-19 14:38发表



一直很奇怪为啥CNN的卷积、下采样、卷积、下采样...这样一系列操作之后就产生了神气的预测结果呢，背后有啥数学物理上的原理意义吗？

Re: ProLianGee 2016-03-15 12:55发表



回复ADF1230: 和BP是一样的原理吧，利用反向传播，计算参数对损失函数的影响，迭代更新参数，使得损失函数最小。

13楼 fangfei0110 2015-12-16 21:07发表



imagenet里的max pooling应该是3\*3，stride=2  
但是第5层后的max pooling如果也是这样的话得不到2048这个特征数，文章中没有交代，但是反推的话应该是4\*4，stride=3的pooling

12楼 changlingling 2015-12-15 20:34发表



能否提供下图片出处，灰常谢谢！

11楼 止的不是战 2015-12-08 19:32发表



不错！！因为是新手，到网络结构那里就有点看不懂了。

10楼 自在逍遥 2015-11-21 23:42发表



写的非常棒,特别清晰。

9楼 jia\_zhen 2015-11-08 17:15发表



GoogLeNet

8楼 小灰兔呼噜噜 2015-06-05 17:17发表



请问楼主，Alex的CNN结构图第一层的"stride of 4"是什么意思？第一层的max-pooling如果是2\*2的话，怎么从第一层的224\*224变成第二层的55\*55呢？第一层的max-pooling应该是4\*4的吧？



Re: CJEQ 2015-12-13 11:04发表



回复I281865263：步长是4，可以看成是一个滑块每次移动多少个格子

7楼 小灰兔呼噜噜 2015-06-05 16:13发表



一直对CNN的通道、卷积核大小、filter size、共享卷积等概念没有搞清楚，读了博主这篇文章以后豁然开朗，真的是讲得非常细致清楚，感谢博主，向博主学习！

6楼 stephenxy1 2015-03-19 05:18发表



大牛能不能推荐一下这方面比较强大的convolution kernel 和classification model. 还有讲解一下线性变换在图像识别中的具体应用

Re: 张雨石 2015-03-19 13:38发表



回复stephenxy1：微软还出了一个比GoogNet更强大的图像识别算法。至于线性变换，能否说一下你具体在什么应用上需要用到么？

5楼 huyi100 2015-03-11 10:22发表



有一个地方写错了

“所以每个样例 (example) 都会得到一个  $892 \times 400 = 3,168,400$  维的卷积特征向量。”  
应该是  $7921 \times 400$  才对

Re: 张雨石 2015-03-11 11:50发表



回复huyi100：你说的对，已经改过来了，多谢指出。

4楼 NFD\_Year 2015-03-08 19:42发表



博主叙述的很精彩。但描述AlexNet时的第一层卷积是96个11\*11\*3的卷积核

Re: 张雨石 2015-03-11 11:46发表



回复u010668083：你说的是对的，已改过来了。多谢！

3楼 GK\_2014 2015-01-13 10:23发表



顶一个，学习一下

2楼 精灵小弟 2014-12-24 19:52发表



xuedi,jiayou

1楼 青色河畔 2014-12-02 22:23发表



大牛，已经关注你很久了，虽然不知道你姓甚名谁但是觉得你很厉害，默默支持，继续加油！！~~~

Re: 张雨石 2014-12-03 00:31发表



回复yintianxu514：侧边栏有我的微博和邮箱，可以多交流哈。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack
VPN	Spark	ERP	IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP
jQuery	BI	HTML5	Spring	Apache	.NET	API	HTML	SDK	IIS	Fedora
XML	LBS	Unity	Splashtop	UML	components	Windows Mobile	Rails	QEMU	KDE	Cassandra
CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	SpringSide	Maemo
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr
Angular	Cloud Foundry	Redis	Scala	Django	Bootstrap					

