

项目+深度学习:

1. 后续如何优化? 收益是什么?
2. 单张图片测试的速度?

使用GPU, 一张图片的测试速度是0.032s

3. CNN中对结果影响比较大的参数?

学习率、batch-size、卷积核大小和数量、网络层数。

4. 如何进行fintuning, 层数不相同的情况

层数不同也可以调试, 一般将不想调试的网络层修改名字即可。

5. 反卷积网络是否有所了解?
6. 卷积核大小的选择

更小的卷积核可以减小参数, 节省运算开销, 虽然训练时间会变长, 但是总体上参数和预测时间会减少, 同时小的卷积核会提取到图片中局部的特征, 而大的卷积核则可以提取到更加全局的信息。两个3*3的卷积核可以达到一个5*5卷积核的作用, 并且还能减少参数和预测时间。

7. 1*1 卷积核的作用

1. 实现跨通道的交互和信息整合, 实现多个feature map的线性组合, 实现跨通道的信息整合
 2. 进行卷积核通道数的降维和升维
-
1. 实现平滑操作
 2. 可以在保持feature map 尺寸不变(即不损失分辨率)的前提下大幅增加非线性特性, 把网络做得很deep。

1. CNN对图像分类效果好的原因

图像存在局部相关性，而卷积运算可以获取这种空间相关性。

2. BatchNorm层的作用，为什么使用？用在哪里？什么时候使用？为什么能产生这样的效果？

1. 作用有两个，一个是逐层尺度归一，避免梯度消失和溢出；其次是加快收敛速度，可以防止过拟合；
2. 防止梯度弥散，在BN中，是通过将activation规范为均值和方差一致的手段使得原本会减小的activation的scale变大。

$$x = Wu + b$$

3. BN应作用在非线性映射前，即对做规范化。
4. 在神经网络训练时遇到收敛速度很慢，或梯度爆炸等无法训练的状况时可以尝试BN来解决。另外，在一般使用情况下也可以加入BN来加快训练速度，提高模型精度。

(5) 原因在于神经网络学习过程本质就是为了学习数据分布，一旦训练数据与测试数据的分布不同，那么网络的泛化能力也大大降低；另外一方面，一旦每批训练数据的分布各不相同(batch梯度下降)，那么网络就要在每次迭代都去学习适应不同的分布，这样将会大大降低网络的训练速度，这也正是为什么我们需要对数据都要做一个归一化预处理的原因。

对于深度网络的训练是一个复杂的过程，只要网络的前面几层发生微小的改变，那么后面几层就会被累积放大下去。一旦网络某一层的输入数据的分布发生改变，那么这一层网络就需要去适应学习这个新的数据分布，所以如果训练过程中，训练数据的分布一直在发生变化，那么将会影响网络的训练速度。

我们知道网络一旦train起来，那么参数就要发生更新，除了输入层的数据外(因为输入层数据，我们已经人为的为每个样本归一化)，后面网络每一层的输入数据分布是一直在发生变化的，因为在训练的时候，前面层训练参数的更新将导致后面层输入数据分布的变化。以网络第二层为例：网络的第二层输入，是由第一层的参数和input计算得到的，而第一层的参数在整个训练过程中一直在变化，因此必然会引起后面每一层输入数据分布的改变。我们把网络中间层在训练过程中，数据分布的改变称之为：“Internal Covariate Shift”。Paper所提出的算法，就是要解决在训练过程中，中间层数据分布发生改变的情况，于是就有了Batch Normalization，这个牛逼算法的诞生。

10. 全卷积网络介绍

FCN将传统CNN中的全连接层转化成一个个的卷积层。在传统的CNN结构中，前5层是卷积层，第6层和第7层分别是一个长度为4096的一维向量，第8层是长度为1000的一维向量，分别对应1000个类别

的概率。FCN将这3层表示为卷积层，卷积核的大小(通道数，宽，高)分别为(4096, 1, 1)、(4096, 1, 1)、(1000, 1, 1)。所有的层都是卷积层，故称为全卷积网络。

可以发现，经过多次卷积(还有pooling)以后，得到的图像越来越小，分辨率越来越低(粗略的图像)，那么FCN是如何得到图像中每一个像素的类别的呢？为了从这个分辨率低的粗略图像恢复到原图的分辨率，FCN使用了上采样。例如经过5次卷积(和pooling)以后，图像的分辨率依次缩小了2, 4, 8, 16, 32倍。对于最后一层的输出图像，需要进行32倍的上采样，以得到原图一样的大小。

这个上采样是通过反卷积(deconvolution)实现的。对第5层的输出(32倍放大)反卷积到原图大小，得到的结果还是不够精确，一些细节无法恢复。于是Jonathan将第4层的输出和第3层的输出也依次反卷积，分别需要16倍和8倍上采样，结果就精细一些了。

与经典的CNN在卷积层之后使用全连接层得到固定长度的特征向量进行分类不同，FCN可以接受任意尺寸的输入图像，采用反卷积层对最后一个卷积层的feature map进行上采样，使它恢复到输入图像相同的尺寸，从而可以对每个像素都产生了一个预测，同时也保留了原始输入图像中的空间信息，最后在上采样的特征图上进行逐像素分类。论文中逐像素计算softmax分类的损失，相当于每一个像素对应一个训练样本。

与传统用CNN进行图像分割的方法相比，FCN有两大明显的优点：一是可以接受任意大小的输入图像，而不用要求所有的训练图像和测试图像具有同样的尺寸。二是更加高效，因为避免了由于使用像素块而带来的重复存储和计算卷积的问题。

同时FCN的缺点也比较明显：一是得到的结果还是不够精细。进行8倍上采样虽然比32倍的效果好了很多，但是上采样的结果还是比较模糊和平滑，对图像中的细节不敏感。二是对各个像素进行分类，没有充分考虑像素与像素之间的关系，忽略了在通常的基于像素分类的分割方法中使用的空间规整(spatial regularization)步骤，缺乏空间一致性。

传统的基于CNN的分割方法：为了对一个像素分类，使用该像素周围的一个图像块作为CNN的输入用于训练和预测。这种方法有几个缺点：一是存储开销很大。例如对每个像素使用的图像块的大小为15x15，然后不断滑动窗口，每次滑动的窗口给CNN进行判别分类，因此则所需的存储空间根据滑动窗口的次数和大小急剧上升。二是计算效率低下。相邻的像素块基本上是重复的，针对每个像素块逐个计算卷积，这种计算也有很大程度上的重复。三是像素块大小的限制了感知区域的大小。通常像素块的大小比整幅图像的大小小很多，只能提取一些局部的特征，从而导致分类的性能受到限制。

而全卷积网络(FCN)试图从抽象的特征中恢复出每个像素所属的类别。即从图像级别的分类进一步延伸到像素级别的分类。

1. BP算法？如果对 $w x + b$ 的反向计算是 $w^T * \epsilon$, 为什么是 w 的转置？

(1)

(2) 维度匹配

2. Caffe中im2col函数的作用？

将卷积转为矩阵相乘，加快卷积运算的计算速度。

13. 卷积是如何实现的？

卷积核在卷积计算时没有“翻转”，而是与输入图片做滑动窗口“相关”计算。

1. 动量的作用

计算梯度时考虑历史梯度信息，使随机梯度下降更容易跳出局部最优，加速收敛

15 卷积本质是什么？卷积神经网络本质是什么？

(1)

(2)

1. 项目，与业界最好成绩相比？优化了哪些，或者相比更好的地方在哪里？

2. CNN中，计算速度主要取决于什么？

卷积运算，并且主要是channel这个维度。

3. 提高卷积运算速度，比如提高矩阵运算？

4. 手机实现CNN，主要使用哪个实现提速？

使用ARM的汇编指令。

5. 如何增大卷积网络的感受野?

增大卷积核，加深网络，加pooling层

感受野的定义参考 <http://blog.csdn.net/wonengguwozai/article/details/73133737> 和 <https://zhuanlan.zhihu.com/p/22627224>。

21. 什么样的资料集不适合用深度学习?

- 数据集太小，数据样本不足时，深度学习相对其它机器学习算法，没有明显优势。
- 数据集没有局部相关特性，目前深度学习表现比较好的领域主要是图像 / 语音 / 自然语言处理等领域，这些领域的一个共性是局部相关性。图像中像素组成物体，语音信号中音位组合成单词，文本数据中单词组合成句子，这些特征元素的组合一旦被打乱，表示的含义同时也被改变。对于没有这样的局部相关性的数据集，不适于使用深度学习算法进行处理。举个例子：预测一个人的健康状况，相关的参数会有年龄、职业、收入、家庭状况等各种元素，将这些元素打乱，并不会影响相关的结果。

1. caffe: 为什么要用protobuff?

2. rcnn/fast rcnn/fater rcnn区别?

3. tensorflow: 为什么用图?

机器学习算法

1. 随机森林的原理？
2. 随机森林如何避免过拟合，而决策树会过拟合的原因
3. 决策树特征选择的方法？
4. 决策树，如何特征值是连续的如何选择？

C4.5算法。首先对特征值进行升序排序，然后使用二分法，即寻找相邻两个特征值的中点作为分裂点，如1, 2, 3，可以选择1.5和2.5作为分裂点，选择标准是信息增益，选择信息增益最大的分裂点，然后再计算最佳分裂点的信息增益率作为该特征的信息增益率。

5. 决策树，做回归时使用的准则是？

平方误差最小化

6. KNN的三个注意点，K过大和过小时的偏差和方差？

- (1) K值选择，距离度量和分类决策规则；
- (2) k过小的时候，偏差小而方差大，容易过拟合；k过大时，偏差大，方差小

7. LR使用L1正则化，再使用梯度下降，会出现什么问题？

8. LR的损失函数，为什么使用这个？

这是似然损失函数，使用极大似然估计求解；

参数估计方法除了极大似然估计，还有EM算法、点估计。点估计是计算样本的均值，使用均值作为参数估计。

9 随机森林需要剪枝吗？

不用，一般很多的决策树算法都有一个很重要的步骤-剪枝，这里不需要这样做，因为之前的两个随机采样的过程保证了随机性，即对数据的随机采样和特征的随机采样，就算不减枝，也不会出现过拟合。

1. SVM的kkt条件？用于解决什么问题？

$$\frac{\partial L}{\partial w_v} = w_v - \sum_{i=1}^n \alpha_i y_i x_{iv} = 0, v = 1, \dots, d$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^n \alpha_i y_i = 0$$

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

$$\alpha_i \geq 0$$

$$\alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] = 0$$

(1)，即《统计学习方法》P105。它是指函数的最优值必定满足下面条件：

(1) L 对各个 x 求导为零；

(2) $h(x)=0$,

(3) $\sum \alpha_i g_i(x)=0, \alpha_i \geq 0$

1. 拉格朗日乘子法(Lagrange Multiplier)和KKT(Karush-Kuhn-Tucker)条件是求解约束优化问题的重要方法，在有等式约束时使用拉格朗日乘子法，在有不等约束时使用KKT条件。前提是：只有当目标函数为凸函数时，使用这两种方法才保证求得的是最优解。

2. SVM的核函数？如果使用RBF核函数，可以将原本5维的数据映射到多少维？

(1)

1. 映射到无限维，因为它是使用泰勒展开式，可以展开到无穷维。

泰勒展开式其实是0-n维的多项式核函数的和。我们知道多项式核函数将低维数据映射到高维(维度是有限的)，那么 对于无限个 不同维的多项式核函数之和的高斯核，其中也包括 无穷维度 的 多项式核函数。

1. AUC的含义?

AUC是指 从一堆样本中随机抽一个，抽到正样本的概率 比 抽到负样本的概率 大的可能性。

13. C++实现kmeans, LR, softmax算法

数据结构算法

1. 关于最短路径问题，给出 $n \times n$ 格子，格子上的不同数值，求格子上一A到点B的路径上，数值之和最短的路径？

1. 优先队列可以使用什么实现？

使用堆排序实现

2. 堆排序的建堆时间复杂度是？

$O(n)$ 。在构建堆的过程中，因为是从完全二叉树的最下层最右边的非叶结点开始构建，将它与其孩子进行比较和若有必要的交换，对每个非叶结点，最多进行两次比较和互换操作，这里需要进行这种操作的非叶结点数目是 $n/2$ 个。

3. 图的深度遍历和广度遍历的时间复杂度？

时间复杂度都是 $O(n^2)$ 或者 $O(n+e)$ ，后者是使用邻接表的复杂度， n 是顶点个数， e 个无向图边数，有向图中的弧数。

深度遍历一般使用栈，而广度遍历使用队列。

4. 爬虫如何实现？一天可以爬取数量？

(1)

1. 一天可以爬取大约16-17万张图片，每分钟大约10张左右图片，10分钟大约115张图片，一个小时大约7000张图片。

6. 给出一个元素无序的数组，求出一个数，使得其左边的数都小于它，右边的数都大于等于它。

举例：1, 2, 3, 1, 2, 0, 5, 6，返回下标6（数字为5）。

思路（1）：

朴素算法，对于每一个数，都检测它的左边和右边是否满足题意。

时间复杂度为 $O(n^2)$

思路（2）

使用变量求解：

（1）目前找到符合题意的候选值，nCandid

（2）目前已遍历数组的最大值，nMax：为了选下一次的候选值

（3）目前的候选值是否有效，bIsExist：检测是否需要重新选择候选值

思路：如果候选值有效，可以继续遍历下面的数据

如果候选值小于目前的值，则该候选失效。之后遍历元素时，就要重新选择候选值

选择候选值时，对于某一个元素，如果该元素比之前遍历过元素的最大值还要大nMax，则该元素就为候选。

复杂度：遍历一遍数组即可，时间： $O(n)$ ，空间 $O(1)$

```
1. #include <iostream>
2. #include <assert.h>
3. #include <list>
4. using namespace std;
5.
6. int FindNum(int nArr[],int nLen)
7. {
8.     assert(nArr && nLen > 0);
9.     int nPos = 0;
10.    int nCandid = nArr[0];
11.    int nMax = nArr[0];
12.    bool bIsExist = true;
13.    for (int i = 1;i < nLen;i++)
14.    {
15.        if (bIsExist)//候选有效
```

```

16.     {
17.         if (nCandid > nArr[i])//候选失效
18.         {
19.             bIsExist = false;
20.         }
21.         else
22.         {
23.             nMax = nArr[i];
24.         }
25.     }
26.     else //候选失效
27.     {
28.         if (nArr[i] >= nMax)//重新找到候选
29.         {
30.             bIsExist = true;
31.
32.             nCandid = nArr[i];
33.             nMax = nArr[i];
34.             nPos = i;
35.         }
36.     }
37. }
38. return bIsExist ? nPos : -1;
39. }

```

C++/python 编程语言

1. C++和python的类的区别？

C++默认private，Python默认public；

C++可以实现多态，而python没有。

最后一个问题之你还有什么想问我的？

1. 我进去之后会做什么？
2. 团队是做什么东西的（业务是什么）？
3. 内部项目还是外部项目？

4. 就我之前的表现来看，你觉得我的优缺点在哪里？（这个问题可以侧面打探出他对你的评价，而且可以帮助你给自己查漏补缺）
5. 偏基础还是偏业务（简单粗暴地说，做基础就是写给程序员用的东西，做业务就是写给用户用的东西）？
6. 技术氛围怎么样？主要用到什么技术？有什么开源产出吗？你们做 code review 吗