

1. 卷积神经网络可以对一个输入进行多种变换（旋转、平移、缩放），这个表述正确吗？

答案：错误

把数据传入神经网络之前需要做一系列数据预处理（也就是旋转、平移、缩放）工作，神经网络本身不能完成这些变换。

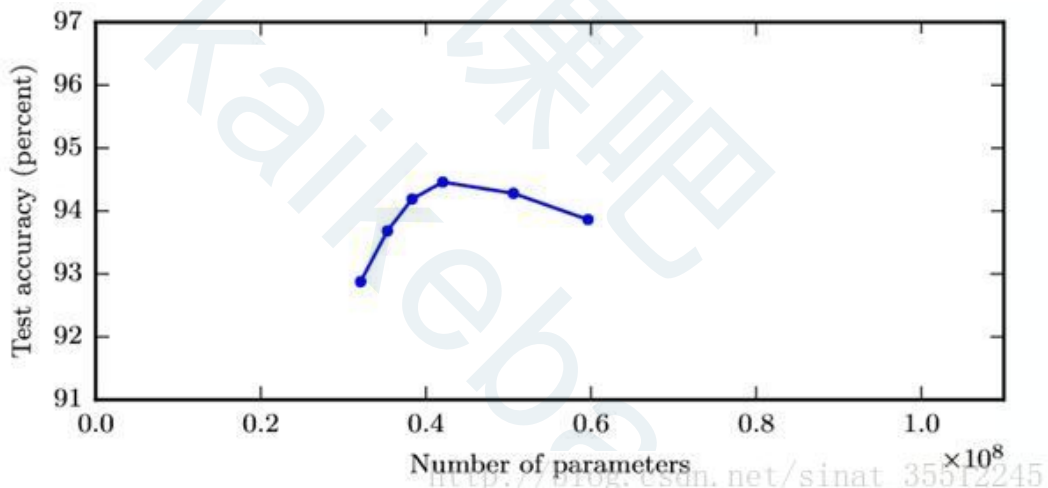
2. 下列哪一项在神经网络中引入了非线性？（ ）

- A. 随机梯度下降
- B. 修正线性单元（ReLU）
- C. 卷积函数
- D. 以上都不正确

答案：（B）

修正线性单元是非线性的激活函数。

3. 下图显示了训练过的 3 层卷积神经网络准确度，与参数数量(特征核的数量)的关系。



从图中趋势可见，如果增加神经网络的宽度，精确度会增加到一个特定阈值后，便开始降低。造成这一现象的可能原因是什么？（ ）

- A. 即使增加卷积核的数量，只有少部分的核会被用作预测
- B. 当卷积核数量增加时，神经网络的预测能力（Power）会降低
- C. 当卷积核数量增加时，它们之间的相关性增加(correlate)，导致过拟合
- D. 以上都不正确

答案：（C）

如 C 选项指出的那样，可能的原因是核之间的相关性。

4. 假设我们已经在 ImageNet 数据集(物体识别)上训练好了一个卷积神经网络。然后给这张卷积神经网络输入一张全白的图片。对于这个输入的输出结果为任何种类的物体的可能性都是一样的，对吗？（ ）

- A. 对的
- B. 不知道
- C. 看情况
- D. 不对

答案：（D）

各个神经元的反应是不一样的

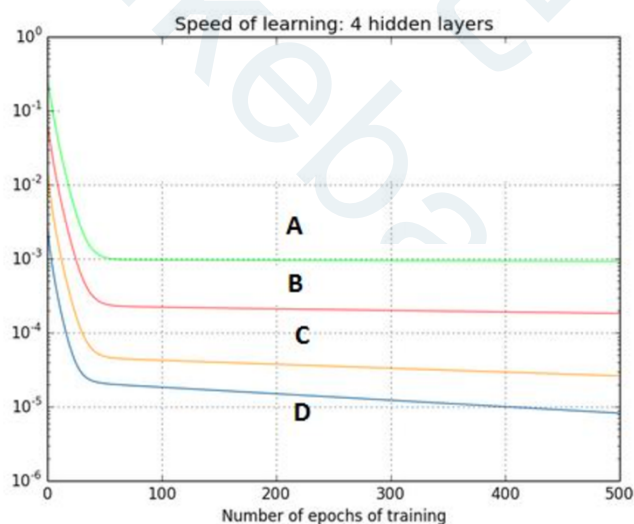
5. 当在卷积神经网络中加入池化层(pooling layer)时, 变换的不变性会被保留, 是吗? ( )

- A. 不知道
- B. 看情况
- C. 是
- D. 否

答案：（C）

使用池化时会导致出现不变性。

6. 下图是一个利用 **sigmoid** 函数作为激活函数的含四个隐藏层的神经网络训练的梯度下降图。这个神经网络遇到了梯度消失的问题。下面哪个叙述是正确的? ( )



[http://blog.csdn.net/sinat\\_35512245](http://blog.csdn.net/sinat_35512245)

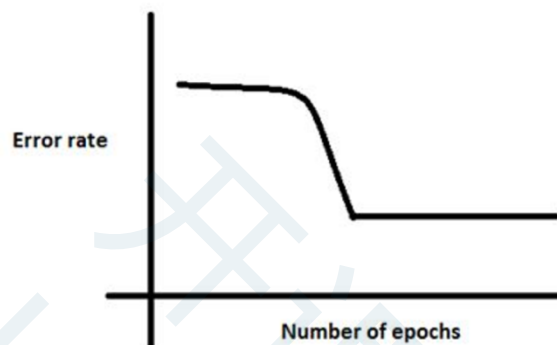
- A. 第一隐藏层对应 D, 第二隐藏层对应 C, 第三隐藏层对应 B, 第四隐藏层对应 A
- B. 第一隐藏层对应 A, 第二隐藏层对应 C, 第三隐藏层对应 B, 第四隐藏层对应 D
- C. 第一隐藏层对应 A, 第二隐藏层对应 B, 第三隐藏层对应 C, 第四隐藏层对应 D

D. 第一隐藏层对应 B，第二隐藏层对应 D，第三隐藏层对应 C，第四隐藏层对应 A

答案：（A）

由于反向传播算法进入起始层，学习能力降低，这就是梯度消失。

7. 下图显示，当开始训练时，误差一直很高，这是因为神经网络在往全局最小值前进之前一直被卡在局部最小值里。为了避免这种情况，我们可以采取下面哪种策略？（ ）



[http://blog.csdn.net/sinat\\_35512245](http://blog.csdn.net/sinat_35512245)

- A. 改变学习速率，比如一开始的几个训练周期不断更改学习速率
- B. 一开始将学习速率减小 10 倍，然后用动量项(momentum)
- C. 增加参数数目，这样神经网络就不会卡在局部最优处
- D. 其他都不对

答案：（A）

选项 A 可以将陷于局部最小值的神经网络提取出来。

8. 考虑某个具体问题，你可能只有少量数据来解决这个问题。不过幸运的是你有一个类似问题已经预先训练好的神经网络。可以用下面哪种方法来利用这个预先训练好的网络？（ ）

- A. 把除了最后一层外所有的层都冻住，重新训练最后一层
- B. 对新数据重新训练整个模型
- C. 只对最后几层进行调参(fine tune)
- D. 对每一层模型进行评估，选择其中的少数来用

答案：（C）

9. 增加卷积核的大小对于改进卷积神经网络的效果是必要的吗？

答案：不是，增加核函数的大小不一定会提高性能。这个问题在很大程度上取决于数据集。

10. 输入图片大小为  $200 \times 200$ ，依次经过一层卷积（kernel size  $5 \times 5$ ，padding 1，stride 2），pooling（kernel size  $3 \times 3$ ，padding 0，stride 1），又一层卷积（kernel size  $3 \times 3$ ，padding 1，stride 1）之后，输出特征图大小为（ ）：

- A. 95
- B. 96
- C. 97
- D. 98

正确答案：（C）

解析：

首先我们应该知道卷积或者池化后大小的计算公式：

$$\text{out\_height} = ((\text{input\_height} - \text{filter\_height} + \text{padding\_top} + \text{padding\_bottom}) / \text{stride\_height}) + 1$$

$$\text{out\_width} = ((\text{input\_width} - \text{filter\_width} + \text{padding\_left} + \text{padding\_right}) / \text{stride\_width}) + 1$$

其中，padding 指的是向外扩展的边缘大小，而 stride 则是步长，即每次移动的长度。

这样一来就容易多了，首先长宽一般大，所以我们只需要计算一个维度即可，这样，经过第一次卷积后的大小为： $(200 - 5 + 2) / 2 + 1$ ，取 99；经过第一次池化后的大小为： $(99 - 3) / 1 + 1$  为 97；经过第二次卷积后的大小为： $(97 - 3 + 2) / 1 + 1$  为 97。

11. 深度学习是当前很热门的机器学习算法，在深度学习中，涉及到大量的矩阵相乘，现在需要计算三个稠密矩阵 A,B,C 的乘积 ABC,假设三个矩阵的尺寸分别为  $m \times n$ ， $n \times p$ ， $p \times q$ ，且  $m < n < p < q$ ，以下计算顺序效率最高的是（ ）

- A. (AB)C
- B. AC(B)
- C. A(BC)
- D. 所以效率都相同

正确答案：（A）

解析：

首先，根据简单的矩阵知识，因为  $A * B$ ，A 的列数必须和 B 的行数相等。因此，可以排除 B 选项。

然后，再看 A、C 选项。在 A 选项中， $m \times n$  的矩阵 A 和  $n \times p$  的矩阵 B 的乘积，得到  $m \times p$  的矩阵  $A * B$ ，而  $A * B$  的每个元素需要  $n$  次乘法和  $n - 1$  次加法，忽略加法，共需要  $m * n * p$  次乘法运算。同样情况分析  $A * B$  之后再乘以 C 时的情况，共需要  $m * p * q$  次乘法运算。因此，A 选项 (AB)C 需要的乘法次数是  $m * n * p + m * p * q$ 。同理分析，C 选项 A(BC) 需要的乘法次数是  $n * p * q + m * n * q$ 。

由于  $m * n * p < m * n * q$ ， $m * p * q < n * p * q$ ，显然 A 运算次数更少，故选 A。

12. 卷积网络有哪些池化操作，其作用又是什么？

解析：

池化层往往在卷积层后面，通过池化来降低卷积层输出的特征向量，同时改善结果（不易出现过拟合）。

\* 为什么可以通过降低维度呢？

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）来代表这个区域的特征。

一般池化（General Pooling）

池化作用于图像中不重合的区域（这与卷积操作不同） 最常见的池化操作为平均池化 mean pooling 和最大池化 max pooling:

\* 平均池化：计算图像区域的平均值作为该区域池化后的值。

\* 最大池化：选图像区域的最大值作为该区域池化后的值。

重叠池化（Overlapping Pooling）

重叠池化正如其名字所说的，相邻池化窗口之间会有重叠区域

空间金字塔池化（Spatial Pyramid Pooling）

空间金字塔池化可以把任何尺度的图像的卷积特征转化成相同维度，这不仅可以让 CNN 处理任意尺度的图像，还能避免 cropping 和 warping 操作，导致一些信息的丢失，具有非常重要的意义。

一般的 CNN 都需要输入图像的大小是固定的，这是因为全连接层的输入需要固定输入维度，但在卷积操作是没有对图像尺度有限制，所有作者提出了空间金字塔池化，先让图像进行卷积操作，然后转化成维度相同的特征输入到全连接层，这个可以把 CNN 扩展到任意大小的图像。

### 13. DropOut 具体做了什么使得其能够降低模型的过拟合？

解析：

正则化方法是通过修改损失函数来提高过拟合能力的，而 dropout 是通过改变网络的结构来提高的，这是和正则化方法最本质的区别

dropout 的提出是为了防止在训练过程中的过拟合现象，那就有人想了，能不能对每一个输入样本训练一个模型，然后在 test 阶段将每个模型取均值，这样通过所有模型共同作用，可以将样本最有用的信息提取出来，而把一些噪声过滤掉。

那如何实现这种想法呢？在每一轮训练过程中，我们对隐含层的每个神经元以一定的概率  $p$  舍弃掉，这样相当于每一个样本都训练出一个模型。假设有  $H$  个神经元，那么就有  $2H$  种可能性，对应  $2H$  模型，训练起来时间复杂度太高。我们通过权重共享（weights sharing）的方法来简化训练过程，每个样本所对应模型是部分权重共享的，只有被舍弃掉那部分权重不同。

使用 dropout 可以使用使一个隐含结点不能与其它隐含结点完全协同合作，因此其它的隐含结点可能被舍弃，这样就不能通过所有的隐含结点共同作用训练出复杂的模型（只针对某一个训练样本），我们不能确定其它隐含结点此时是否被激活，这样就有效的防止了过拟合现象。

### 14. 请解释下 CNN 网络中权值共享与稀疏交互的特性及作用？

解析：

CNN 网络中权值共享

权值共享:具体做法是，假设在局部连接中隐藏层的每一个神经元连接的是一个  $10 \times 10$  的局部图像，因此有  $10 \times 10$  个权值参数，将这  $10 \times 10$  个权值参数共享给剩下的神经元，也就是说隐藏层中  $10^6$  (该值可变) 个神经元的权值参数相同，那么此时不管隐藏层神经元的数目是多少，需要训练的参数就是这  $10 \times 10$  个权值参数（也就是卷积核(也称滤波器)的大小）

### CNN 网络中稀疏交互

稀疏连接 (Sparse Connectivity), 又称稀疏交互、稀疏权重。受神经科学中每个细胞只对一个视觉区域内极小的一部分敏感, 而对其他部分则可以视而不见的现象启发, 稀疏连接成为卷积神经网络的一种重要思想, 以帮助改进机器学习系统。

稀疏交互: 相比于全连接神经网络, CNN 每一层的权重要稀疏很多, 卷积核只与在其感受野范围内的输入单元产生交互, 随着卷积核的滑动完成特征的检测。与原来每个输入单元都与输出单元交互相比, 计算时间复杂度降低。