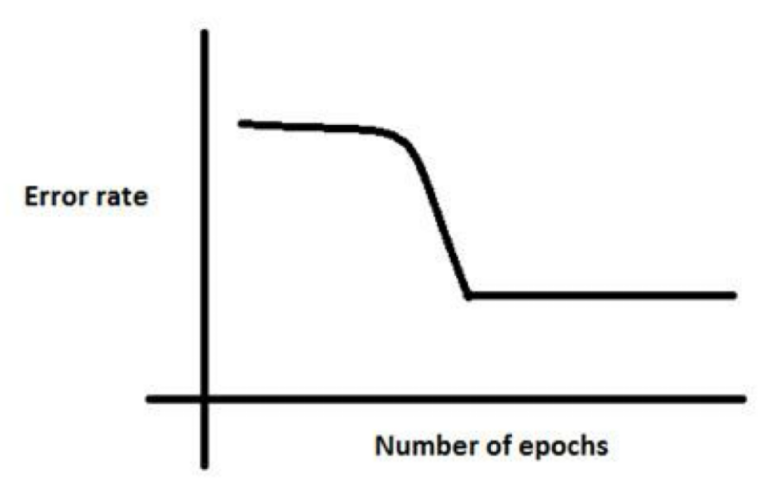
1. **在训练神经网络时，损失函数（Loss）在最初的几个 epochs时没有下降，可能的原因是？**



学习率太低；正则参数太高；陷入局部最小值

1. **批规范化（Batch Normalization）的好处有？**
2. 让每一层的输入的范围都大致固定
3. 它将权重的归一化平均值和标准差
4. 它是一种非常有效的反向传播（BP）方法
5. 这些均不是

a

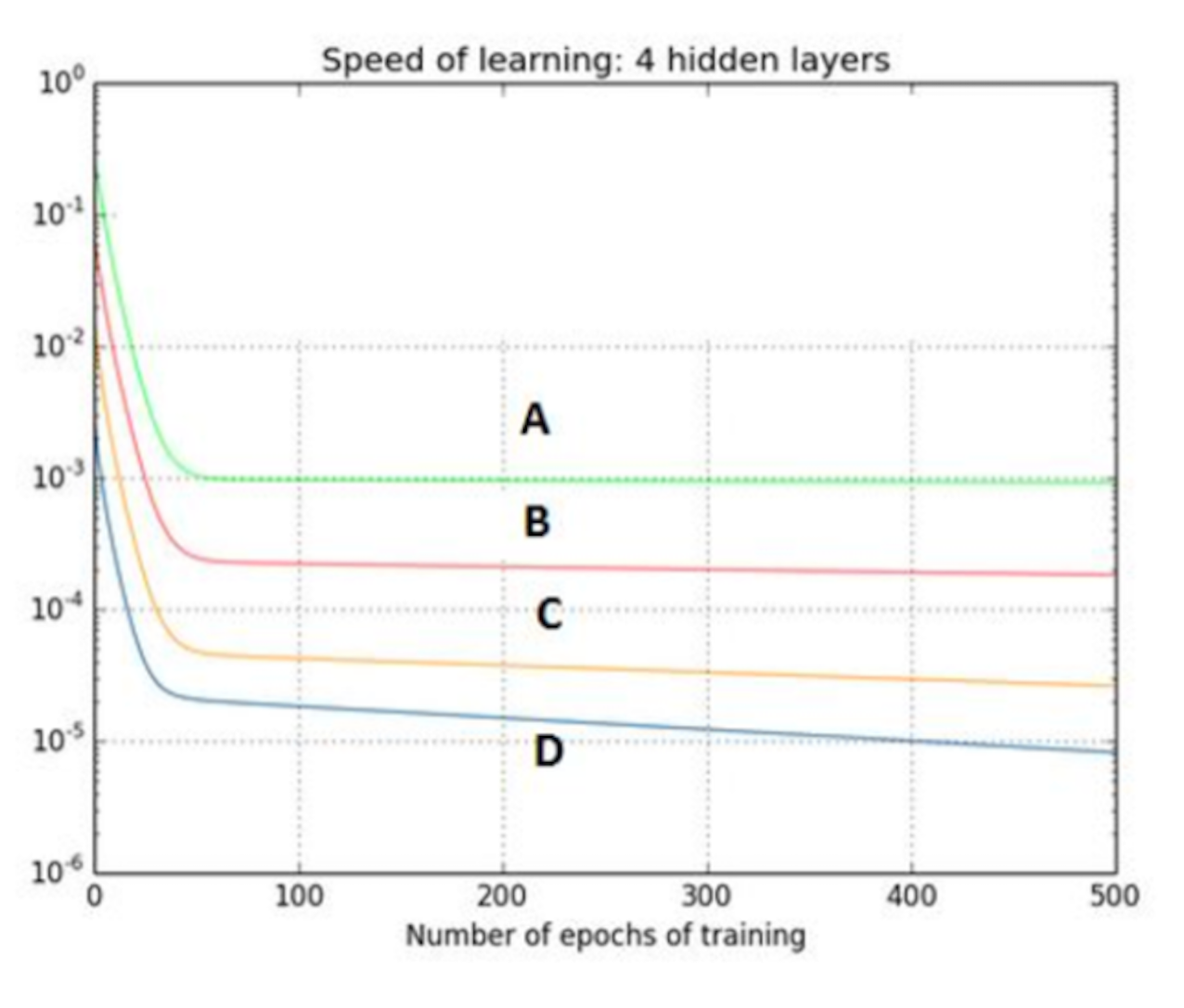
1. **在一个神经网络中，下面哪种方法可以用来处理过拟合？**
2. Dropout
3. Batch Normalization
4. 正则化
5. 以上都可以

d

1. **考虑某个具体问题时，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？**
2. 把除了最后一层外所有的层都冻结，重新训练最后一层
3. 对新数据重新训练整个模型
4. 只对最后几层进行调参（fine tune）
5. 对每一层模型进行评估，选择其中的少数来使用

c

1. **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（纵坐标表示梯度大小）？**



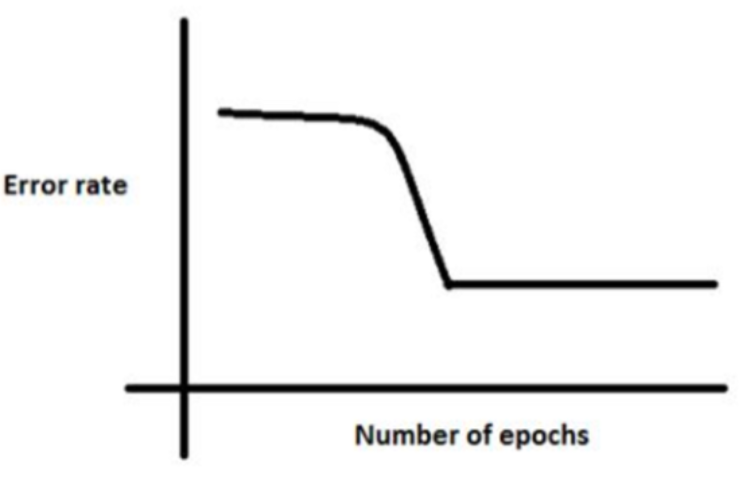
1. 第一隐藏层对应D，第二隐藏层对应C，第三隐藏层对应B，第四隐藏层对应A
2. 第一隐藏层对应A，第二隐藏层对应C，第三隐藏层对应B，第四隐藏层对应D
3. 第一隐藏层对应A，第二隐藏层对应B，第三隐藏层对应C，第四隐藏层对应D
4. 第一隐藏层对应B，第二隐藏层对应D，第三隐藏层对应C，第四隐藏层对应A

a

1. **对于一个分类任务，如果开始时神经网络的权重不是随机赋值的，而是都设成0，下面哪个叙述是正确的？**
2. 其他选项都不对
3. 没啥问题，神经网络会正常开始训练
4. 神经网络可以训练，但是所有的神经元最后都会变成识别同样的东西
5. 神经网络不会开始训练，因为没有梯度改变

c

1. **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？**



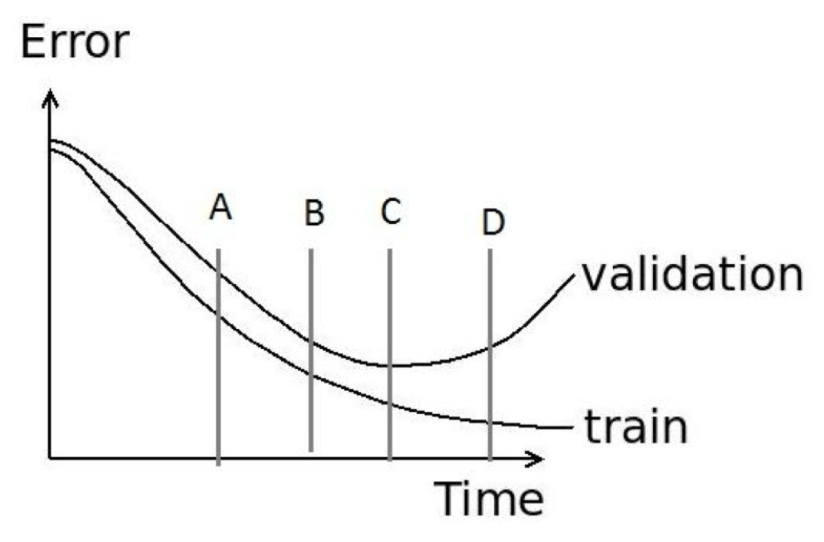
1. 改变学习速率，比如一开始的几个训练周期不断更改学习速率
2. 一开始将学习速率减小10倍，然后用动量项 （momentum）
3. 增加参数数目，这样神经网络就不会卡在局部最优处
4. 其他都不对

a

1. **下列的哪种方法可以用来降低深度学习模型的过拟合问题？ 1.增加更多的数据 2.使用数据扩增技术（data augmentation） 3.使用归纳性更好的架构 4.正则化数据 5.降低架构的复杂度**
2. 145
3. 123
4. 1345
5. 都可以

d

1. **当训练一个神经网络来作图像识别任务时，通常会绘制一张训练集误差和交叉训练集误差图来进行调试。在图中，最好在哪个时间停止训练？**



c

1. **声明1：可以通过将所有权重初始化为0来训练网络**

**声明2：可以通过将偏差初始化为0来很好地训练网络。**

**以上哪些陈述是真实的？**

1. 1
2. 2
3. 12
4. 都是错的

b

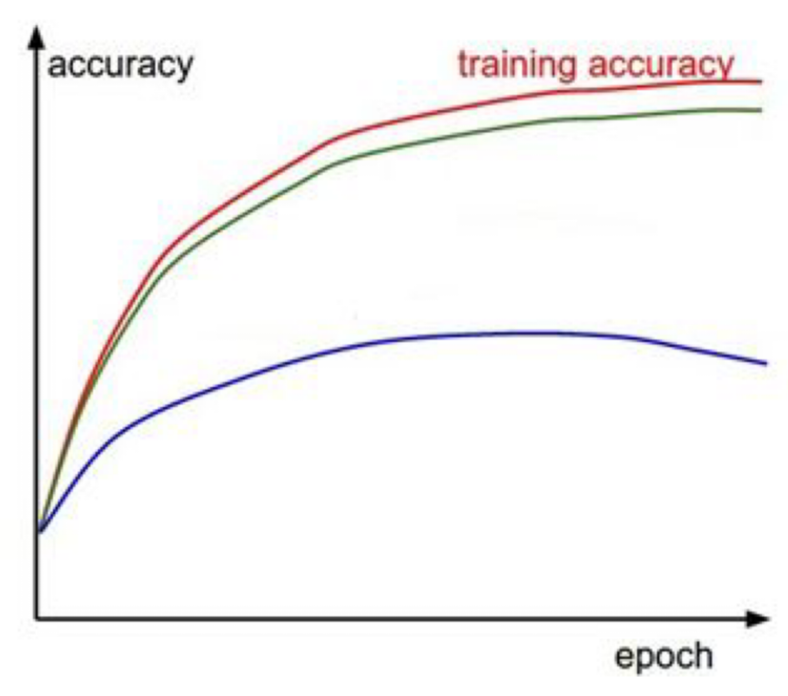
1. **如果我们希望预测n个类（p1,p2,…,pk）的概率使得所有n个p的和等于1，那么下列哪个函数可以用作输出层中的激活函数？**
2. Softmax
3. Relu
4. Sigmoid
5. Tanh

a

1. **使用批量归一化可以解决神经网络训练中的哪些问题？**
2. 过拟合
3. 限制输出过大或过小
4. 训练过慢
5. 以上所有

d

1. **上面的红色曲线表示关于深度学习算法中每个时期的训练精度。绿色和蓝色曲线都表示验证的准确性。 哪条曲线表示过拟合overfitting？**



1. 绿色曲线
2. 蓝色曲线

b

1. **我们可以采取哪些措施来防止神经网络中的过拟合？**
2. 数据增强
3. 权重共享
4. 提前停止
5. Dropout
6. 以上全部

e

1. **以下模型中，在数据预处理时，不需要考虑归一化处理的是**
2. logistic回归
3. SVM
4. 树形模型
5. 神经网络

c

1. **假设用表示一个来自小批量的输入，批量归一化层的参数为：拉伸参数（scale）和偏移参数(shift)，它们的形状与相同。请写出经过批量归一化转换后的表达式。**
2. **批归一化和Dropout一般不会同时使用，请简述原因。**

目标冲突：批归一化的目的是调整激活层的输出，使其保持一定的均值和方差，从而加速训练过程并提高模型稳定性。Dropout则是通过随机地关闭神经网络中的一部分神经元，来防止模型对特定的训练数据过拟合。这两种技术的目标不同。批归一化希望保持激活值的分布稳定，而Dropout则通过引入噪声来增加网络的泛化能力，这在一定程度上是相互矛盾的。

内部协变量偏移：批归一化通过规范化激活层的输出来减少内部协变量偏移。但是，当使用Dropout时，由于随机关闭神经元，激活层的输出分布在每次迭代中都会发生变化，这可能会破坏批归一化试图解决的问题。

训练动态：当同时使用这两种技术时，网络的训练动态可能会受到干扰。Dropout会改变网络中信息的流动方式，而批归一化依赖于这些信息的稳定分布来进行有效的规范化。这种干扰可能会导致训练过程不稳定，甚至降低模型性能。

1. **简述Dropout能够防止过拟合的原因。**

模拟集成学习：Dropout 通过在训练过程中随机关闭（即设为零）网络中的一部分神经元，可以被看作是在训练每个小批量数据时都使用了一个略有不同的网络结构。这类似于训练多个不同的模型并进行集成学习。在测试时，使用所有神经元（通常会调整其输出以反映训练时的Dropout比例）相当于对这些“模型”进行平均，这有助于减少模型对特定训练样本的过度依赖，提高其泛化能力。

减少复杂共适应性：在没有Dropout的情况下，网络的神经元可能会学会共同适应并对特定的训练数据过度拟合，即它们共同调整它们的行为来记住训练数据的特定特征。使用Dropout意味着网络的神经元不能依赖于特定其他神经元的存在，因为这些神经元可能在任何给定的训练步骤中被“丢弃”掉。因此，每个神经元必须学习更加鲁棒的特征，这些特征在各种不同的网络子集中都有用。

减少过度依赖：在传统的神经网络中，某些神经元可能对特定的特征或模式过度敏感，导致过拟合。Dropout 强迫网络在缺少部分信息的情况下进行学习，这减少了模型对任何单个神经元的依赖，从而减少了过拟合的风险。

正则化效果：从正则化的角度来看，Dropout 可以被视为一种增加模型的随机性、降低模型复杂度的方式。通过在训练期间引入噪声，Dropout 防止了网络对训练数据的“记忆效应”，从而起到了类似于传统正则化技术（如L1或L2正则化）的作用。

1. **假设Dropout概率为，为了保证期望值不变，则在测试时，该层模型权重应该变为？**
2. **设模型初始参数为，学习率为0.1，动量更新权重为1，若模型在两次训练过程中的梯度依次为，求两次训练后模型的参数.**