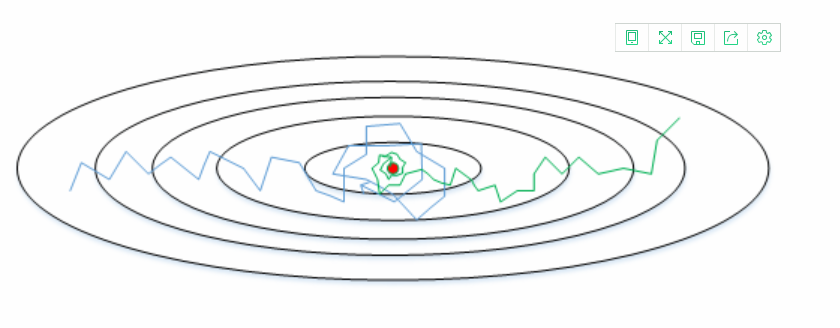
1. 学习率衰减-减小α也能有效提高神经网络训练速度。

简而言之就是随着迭代次数增加，学习因子α逐渐减小。

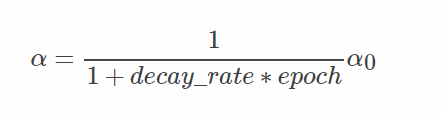
例如下图所示：



图中，蓝色折线表示使用恒定的学习因子α，由于每次训练α相同，步进长度不变，在接近最优值处的振荡也大，在最优值附近较大范围内振荡，与最优值距离就比较远。

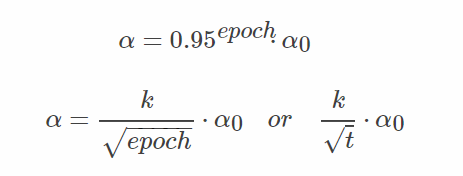
绿色折线表示使用不断减小的α，随着训练次数增加，α逐渐减小，步进长度减小，使得能够在最优值处较小范围内微弱振荡，不断逼近最优值。相比较恒定的α来说，学习率衰减更接近最优值。

学习率衰减公式如下：



其中，deacy\_rate是参数（可调），epoch是训练完所有样本的次数。随着epoch增加，α会不断变小。

其它学习率衰减公式：



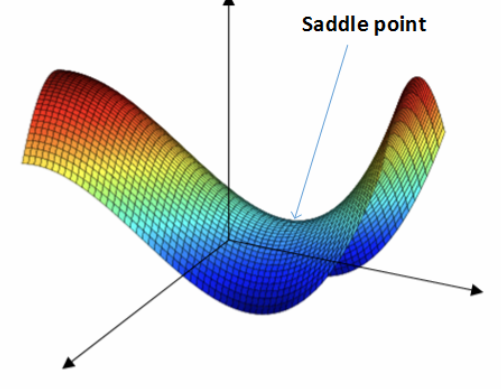
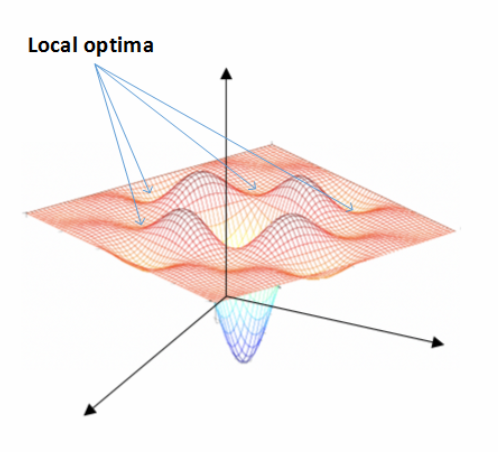
其中，k为可调参数，t为mini-bach数。

注意：除此之外，还可以设置α为关于t的离散值，随着t增加，α呈阶梯式减小。当然，也可以根据训练情况灵活调整当前的α值，但会比较耗时间。

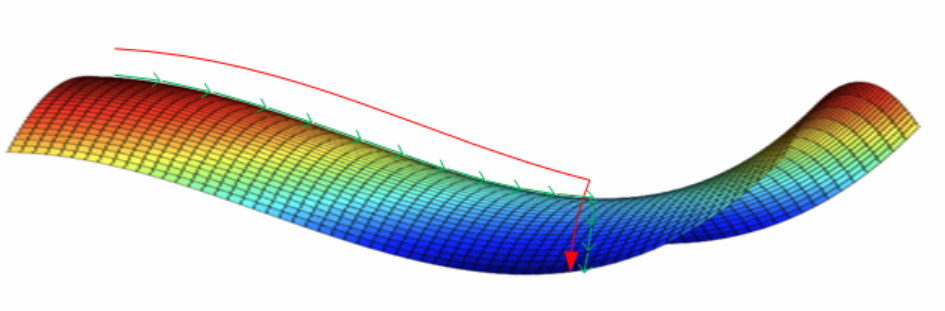
1. 局部最优问题-在使用梯度下降算法不断减小代价函数时，可能会得到局部最优解而不是全局最优解。

之前我们对局部最优解的理解是形如碗状的凹槽，如下图左边所示。

但是在神经网络中，通常大部分梯度为零的“最优点”并不是这些凹槽处，而是形如下图右边所示的马鞍状。也就是说，梯度为零并不能保证都是极小值，也有可能是极大值。



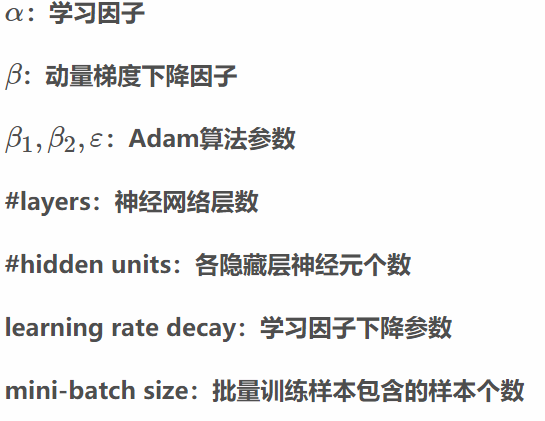
马鞍状会出现的问题：其会降低神经网络学习速度。在梯度接近于零的平缓区域，如下图所示。在平缓区域上梯度很小，前进缓慢，到达最优点需要很长时间。到达最优点后，由于随机扰动，梯度一般能够沿着图中绿色箭头，离开最优点，继续前进，只是在平缓区域上花费了太多时间。



* 注意：****只要选择合理的强大的神经网络，一般不太可能陷入局部最优。平缓区域可能会使梯度下降变慢，降低学习速度。上文介绍的动量梯度下降，RMSprop，Adam算法都能有效解决平缓区域下降过慢的问题，大大提高神经网络的学习速度。****

1. 超参数调试

**1.深度神经网络需要调试的超参数如下**：

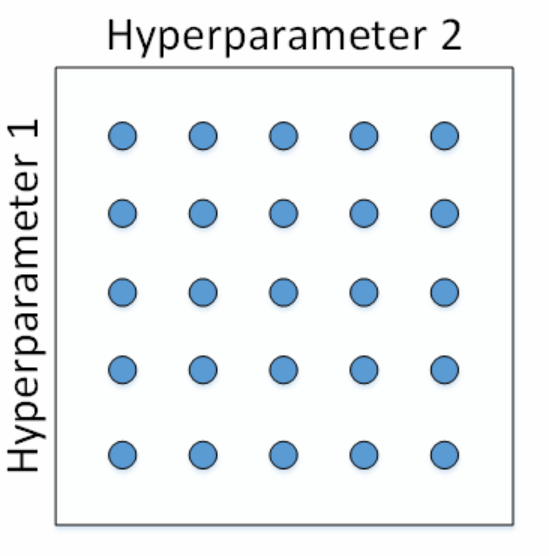


通常来说，学习因子α是最重要的超参数，也是需要重点调试的超参数。动量梯度下降因子β、各隐藏层神经元个数#hidden units和mini-batch size的重要性仅次于α。然后就是神经网络层数#layers和学习因子下降参数learning rate decay。最后，Adam算法的三个参数β1,β2,ε一般常设置为0.9，0.999和10−8，不需要反复调试。当然，这里超参数重要性的排名并不是绝对的，具体情况，具体分析。

**2.选择和调试超参数**

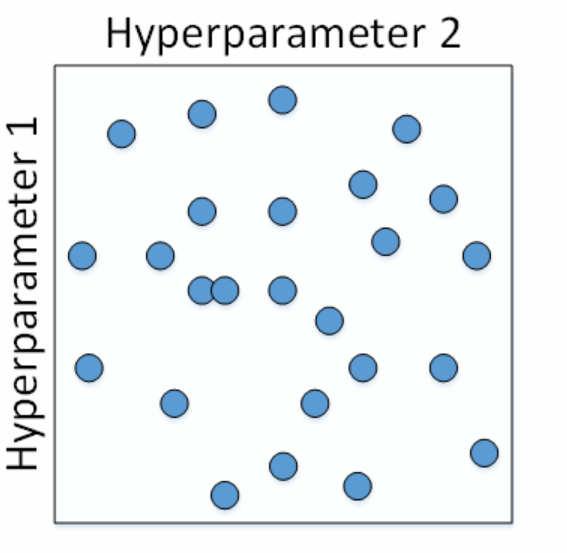
1. **传统的机器学习**中，我们对每个参数等距离选取任意个数的点，然后，分别使用不同点对应的参数组合进行训练，最后根据验证集上的表现好坏，来选定最佳的参数。

例如有两个待调试的参数，分别在每个参数上选取5个点，这样构成了5x5=25中参数组合，这种做法在参数比较少的时候效果较好。如下图所示：



1. **深度神经网络模型**中，我们一般不采用这种均匀间隔取点的方法，比较好的做法是使用**随机选择-第一步。**

也就是说，对于上面这个例子，我们随机选择25个点，作为待调试的超参数，随机化选择参数的目的是为了尽可能地得到更多种参数组合。如下图所示：

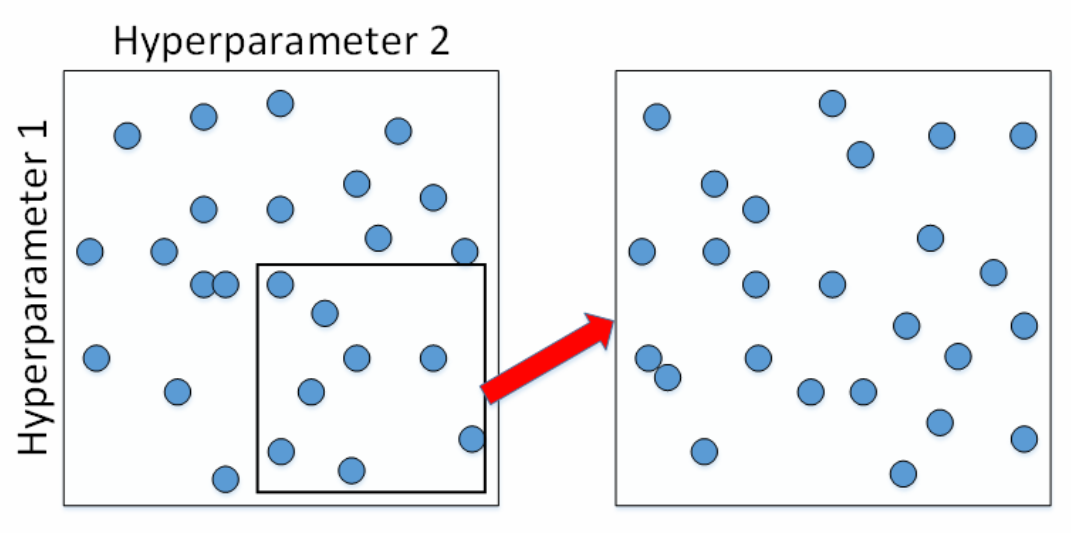


**随机采样的好处：**如果使用均匀采样的话，每个参数只有5种情况；而使用随机采样的话，每个参数有25种可能的情况，因此更有可能得到最佳的参数组合。

这种做法带来的另外一个好处就是对重要性不同的参数之间的选择效果更好。假设hyperparameter1为α，hyperparameter2为ε，显然二者的重要性是不一样的。如果使用第一种均匀采样的方法，ε的影响很小，相当于只选择了5个α值。而如果使用第二种随机采样的方法，ε和α都有可能选择25种不同值。这大大增加了α调试的个数，更有可能选择到最优值。其实，在实际应用中完全不知道哪个参数更加重要的情况下，随机采样的方式能有效解决这一问题，但是均匀采样做不到这点。

**第二步-由粗到细的采样，**经过随机采样之后，我们可能得到某些区域模型的表现较好。然而，为了得到更精确的最佳参数，我们应该继续对表现较好的区域进行放大，再对此区域做更密集的随机采样。

例如，对下图中右下角的方形区域再做25点的随机采样，以获得最佳参数。



1. **引入尺度进行调参**

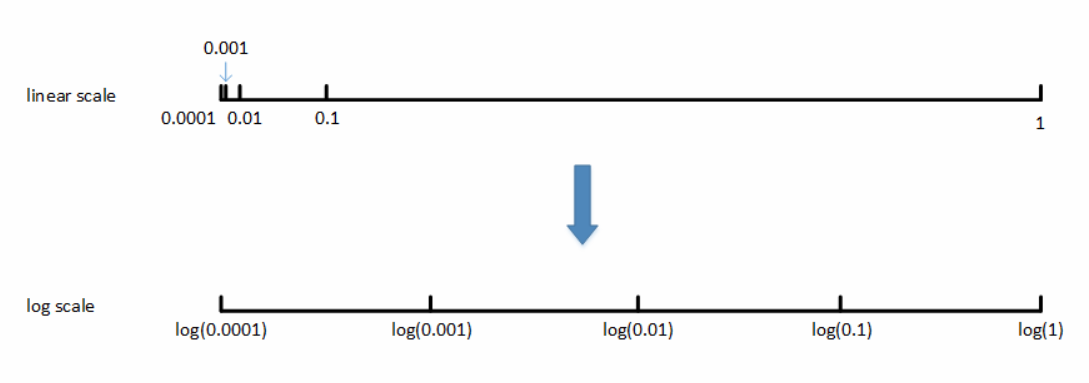
上一部分讲的调试参数使用随机采样，对于某些超参数是可以进行尺度均匀采样的，但是某些超参数需要选择不同的合适尺度进行随机采样。

**不同尺度问题为：**

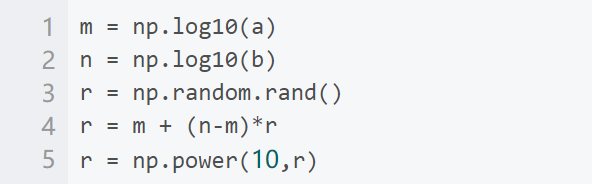
例如对于超参数#layers和#hidden units，都是正整数，是可以进行均匀随机采样的，即超参数每次变化的尺度都是一致的。

但是，对于某些超参数，可能需要非均匀随机采样。例如超参数α，待调范围是[0.0001, 1]。如果使用均匀随机采样，那么有90%的采样点分布在[0.1, 1]之间，只有10%分布在[0.0001, 0.1]之间。这在实际应用中是不太好的，因为最佳的α值可能主要分布在[0.0001, 0.1]之间，而[0.1, 1]范围内α值效果并不好。因此我们更关注的是区间[0.0001, 0.1]，应该在这个区间内细分更多刻度。

**解决的做法：**将linear scale转换为log scale，将均匀尺度转化为非均匀尺度，然后再在log scale下进行均匀采样。这样，[0.0001, 0.001]，[0.001, 0.01]，[0.01, 0.1]，[0.1, 1]各个区间内随机采样的超参数个数基本一致，也就扩大了之前[0.0001, 0.1]区间内采样值个数。



一般解法是，如果线性区间为[a, b]，令m=log(a)，n=log(b)，则对应的log区间为[m,n]。对log区间的[m,n]进行随机均匀采样，然后得到的采样值r，最后反推到线性区间，即10r。10r就是最终采样的超参数。相应的Python语句为：



**注意：**除了α之外，动量梯度因子β也是一样，在超参数调试的时候也需要进行非均匀采样。一般β的取值范围在[0.9, 0.999]之间，那么1−β的取值范围就在[0.001, 0.1]之间。那么直接对1−β在[0.001, 0.1]区间内进行log变换即可。

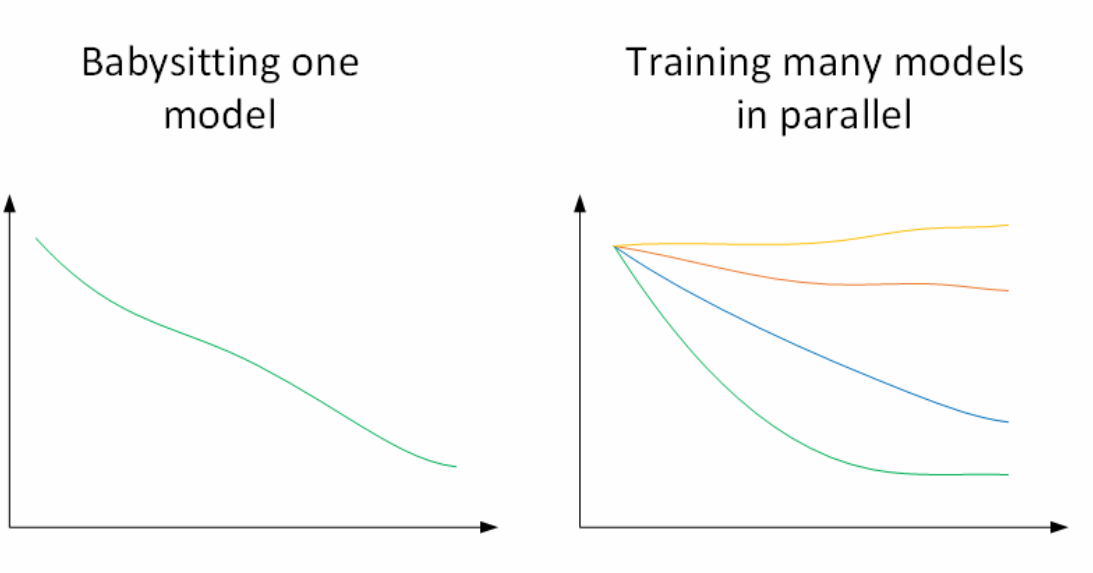
这里解释下为什么β也需要向α那样做非均匀采样。假设β从0.9000变化为0.9005，那么1/1−β基本没有变化。但假设β从0.9990变化为0.9995，那么1/1−β前后差别1000。β越接近1，指数加权平均的个数越多，变化越大。所以对β接近1的区间，应该采集得更密集一些。

1. **实践中的超参数调谐：Pandas vs. Caviar**

经过调试选择完最佳的超参数并不是一成不变的，一段时间之后，需要根据新的数据和实际情况，再次调试超参数，以获得实时的最佳模型。

在训练深度神经网络时，一种情况是受计算能力所限，我们只能对一个模型进行训练，调试不同的超参数，使得这个模型有最佳的表现。我们称之为Babysitting one model。

另外一种情况是可以对多个模型同时进行训练，每个模型上调试不同的超参数，根据表现情况，选择最佳的模型。我们称之为Training many models in parallel。



因为第一种情况只使用一个模型，所以类比做Panda approach；第二种情况同时训练多个模型，类比做Caviar approach。使用哪种模型是由计算资源、计算能力所决定的。一般来说，对于非常复杂或者数据量很大的模型，使用Panda approach更多一些。