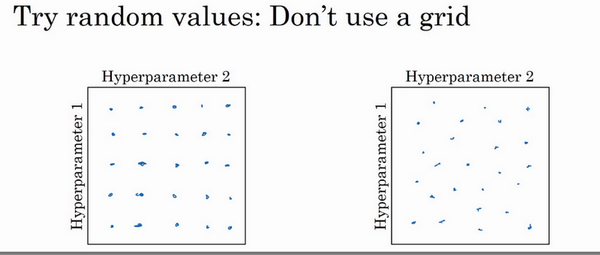
# 超参数调试 Batch正则化和程序框架

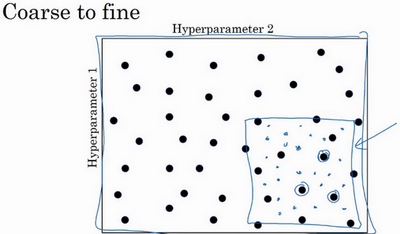
## 调试处理

神经网络的开发过程中会涉及到许许多多的超参数，我们需要学习一些超参数调试的指导性和系统组织超参数调试的技巧。

事实证明，有一些超参数比其他一些更加重要，比如说学习率α，α是需要调试的最重要的超参数。除了α，还有一些重要的超参数，momentum的超参数β，mini-batch的大小，隐藏单元的数量。重要性再往下的就是神经网络的层次，学习率衰减率，以及Adam优化器的参数。

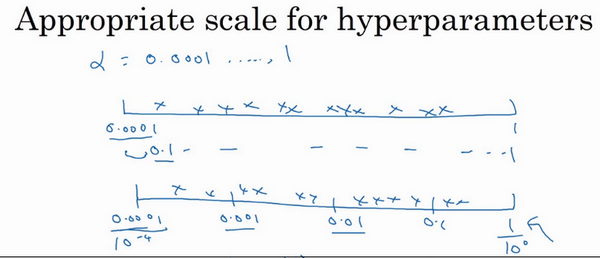
那么，该如何选择调试值呢，在早期的机器学习方法中，常常采用网格取样法，如图所示使用网格取样法，我们可以取样5x5=25个点，然后选择效果最好的那组参数。在参数数量较小的情况下，这种方法通常比较有用。

在深度学习领域，我们推荐使用随机取样法，随机取25个点来进行实验，因为有的时候一个参数的值比其他重要很多，假如参数1是α，参数2是ε，那么我们使用网格取样法时，实际上相当于只取了五个点。而使用随机取样法， 我们对α就会有更多的尝试。

当给超参数取值时，另一种惯例就是从粗糙到精细。如图所示，我们在二维空间中取值时，有时会发现部分区域的表现会比其他区域好，那么我们要做的就是放大这小块区域，继续进行搜索。

## 为超参数选择合适的范围

超参数的随机取值不是在一定范围内均匀取值，而是通过选用合适的标尺进行取值。

我们看下面这个例子，假如我们要对超参数α进行搜索，我们怀疑其值最小是0.0001，最大值是1，如果我们进行均匀取值，我们会发现，百分之90的值都会落在0.1-1之间， 而在0.0001-0.1之间，只占了百分之10搜索资源，这并不合理。

反之，我们使用对数标尺进行超参数搜索会更加合理。我们分别依次取0.0001，0.001，0.01，0.1，1，这样我们的参数分布会更加合理。即对于r∈[-4,0]进行均匀搜索。

另一个棘手的例子是给β取值，用于计算指数平均值。我们由公式可得，取0.9相当于在10天取平均值，0.99就是100天。所以对β搜索的时候，我们不应该使用线性的均匀搜索。我们要探究的是1-β，1-β的值在0.1-0.001之间，令1-β = 10^r，我们要做的就是搜索r，r处于[-3,-1]之间。

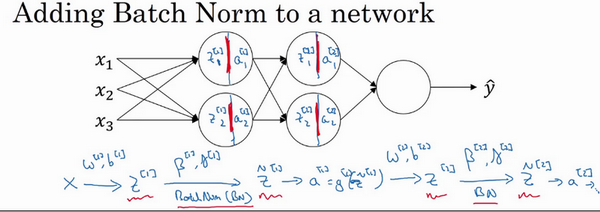
## 归一化网络的激活函数

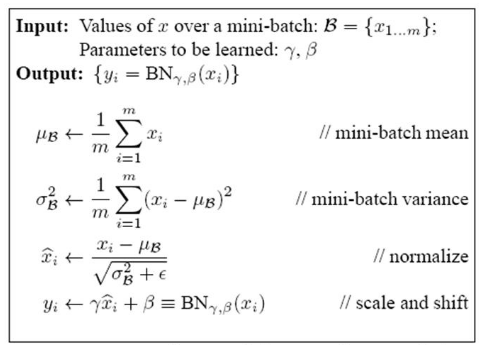
深度学习中有一种很重要的算法叫做Batch归一化，batch归一化会使得参数搜索变得容易，会让神经网络更加稳定。

我们在之前逻辑回归的学习中，以及对输入的特征做了归一化处理，这样会使得训练更加容易，现在我们要做的，就是使隐藏层也进行均值归一化。

我们首先输入训练集，并对得到的各个Z值，进行均值归一化，即减去均值再除以方差。但我们不是总想让z平均值为0，方差为1，我们要做的就是，对z进行变化，，我们可以初始选择γ和β作为参数，在反向传播时，γ和β也会被更新。

## 将Batch Norm拟合进神经网络

我们的计算过程很简单，如下图所示，按往常计算出Z，然后再根据γ和β计算出新的Z，并传入激活函数中。



## 为什么Batch-Norm会奏效

一个原因是，他在做和均值归一化一样的工作，即加快梯度下降的速度。另一个原因是，它能使神经网络的深层参数更能抵御变化，即能够使数据分布保持稳定。即减少了前层参数和后层参数之间的联系。同时，batch-norm也带有一小些正则化效果。Batch-size越大，正则化效果越弱。

## 测试时的Batch-Norm

在测试时，我们只有一个样本，无法计算均值和方差，为此，我们需要使用全局的均值和方差进行替代。我们使用指数平均值来计算全局的均值和方差，并以此应用到测试中。

## Softmax回归

在目前为止，我们实现的一直都是二分类，那么，我们有什么办法进行多分类呢？一种可能的方法就是使用softmax进行多分类学习。假如我们需要进行n分类，那我们的输出层就将有n个节点，计算出这一层的Z后，我们使用softmax函数，首先，我们计算一个向量t，，接着，我们计算a，，a就是n个分类可能出现的概率。

## 训练一个softmax分类器

将损失函数设置为,n个分类

进行反向传播时，求导方法与二分类时相同。