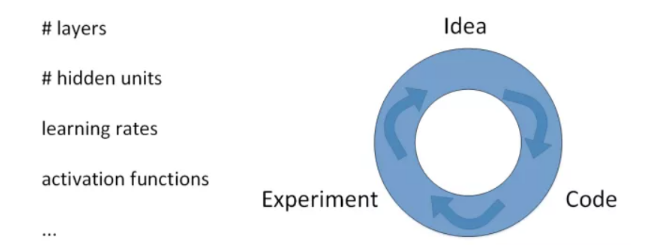
### 数据集划分

对于一个神经网络，选择最佳的训练集（Training sets）、验证集（Development sets）、测试集（Test sets）对神经网络的性能影响非常重要。除此之外，在构建一个神经网络的时候，我们需要设置许多参数，例如神经网络的层数、每个隐藏层包含的神经元个数、学习因子（学习速率）、激活函数的选择等等。实际上很难在第一次设置的时候就选择到这些最佳的参数，而是需要通过不断地迭代更新来获得。这个循环迭代的过程是这样的：我们先有个想法Idea，先选择初始的参数值，构建神经网络模型结构；然后通过代码Code的形式，实现这个神经网络；最后，通过实验Experiment验证这些参数对应的神经网络的表现性能。根据验证结果，我们对参数进行适当的调整优化，再进行下一次的Idea->Code->Experiment循环。通过很多次的循环，不断调整参数，选定最佳的参数值，从而让神经网络性能最优化。



一般情况下，所有数据样本将被划分为训练集、验证集和测试集三个部分。训练集（Train sets）用来训练你的算法模型；验证集（Dev sets）用来验证不同算法的表现情况，从中选择最好的算法模型；测试集（Test sets）用来测试最好算法的实际表现，作为该算法的无偏估计。三部分的数据量一般为3：1：1。

### 偏差和方差

偏差（Bias）和方差（Variance）是机器学习领域非常重要的两个概念和需要解决的问题。在传统的机器学习算法中，Bias和Variance是对立的，分别对应着欠拟合和过拟合，我们常常需要在Bias和Variance之间进行权衡。而在深度学习中，我们可以同时减小Bias和Variance，构建最佳神经网络模型。

如下图所示：

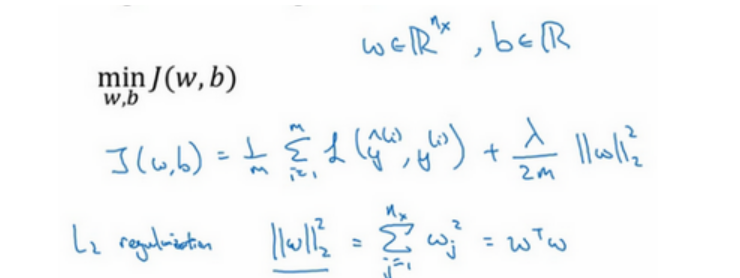


### 正则化

正则化是解决深度学习存在的过拟合问题的方法，其基本做法是对于逻辑回归的w和b两个参数中加入正则化系数λ。

#### 3.1 L2正则化

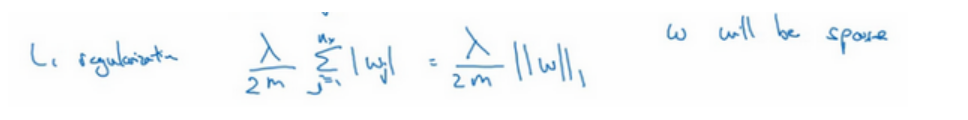
L2正则化：(λ/2m)\*|w|^2



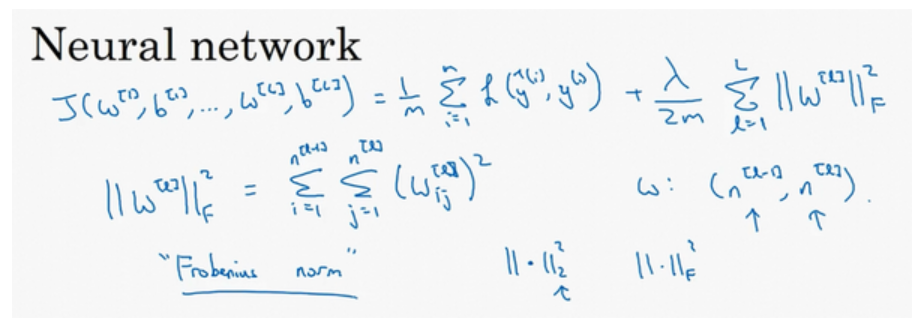
L2正则化中省略参数b的原因：w中已经包含了几乎所有的参数，b仅仅是一个单一的参数，是否将b加入到正则化中无伤大雅。

上图中的J(w,b)为L2正则化后的损失函数

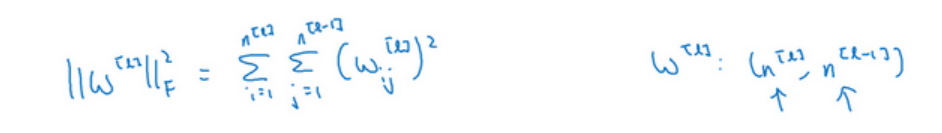
L1正则化：



在神经网络中成本函数可以表示为：



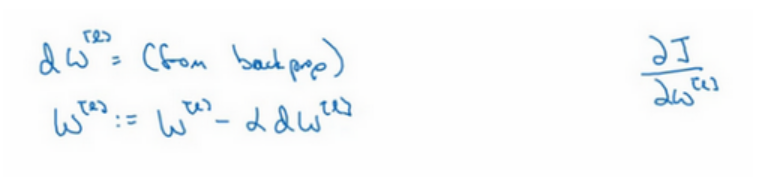
其中



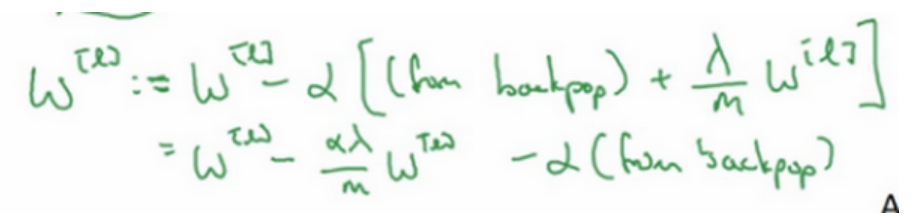
该矩阵范数被称为“弗罗贝尼乌斯范数”。

之后实现梯度下降：

采用反向传递计算出dW，并将W替换为W-αdW，如下图所示：



再增加了λ作为正则化系数之后上式将变为：



#### 3.2 正则化预防过拟合的原因

直观上理解就是如果正则化λ设置得足够大，权重矩阵W被设置为接近于0的值，直观理解就是把多隐藏单元的权重设为0，于是基本上消除了这些隐藏单元的许多影响。如果是这种情况，这个被大大简化了的神经网络会变成一个很小的网络，小到如同一个逻辑回归单元，可是深度却很大，它会使这个网络从过度拟合的状态更接近左图的高偏差状态。

但是会存在一个中间值，于是λ会有一个接近“Just Right”的中间状态。

大致过程如下图所示：

