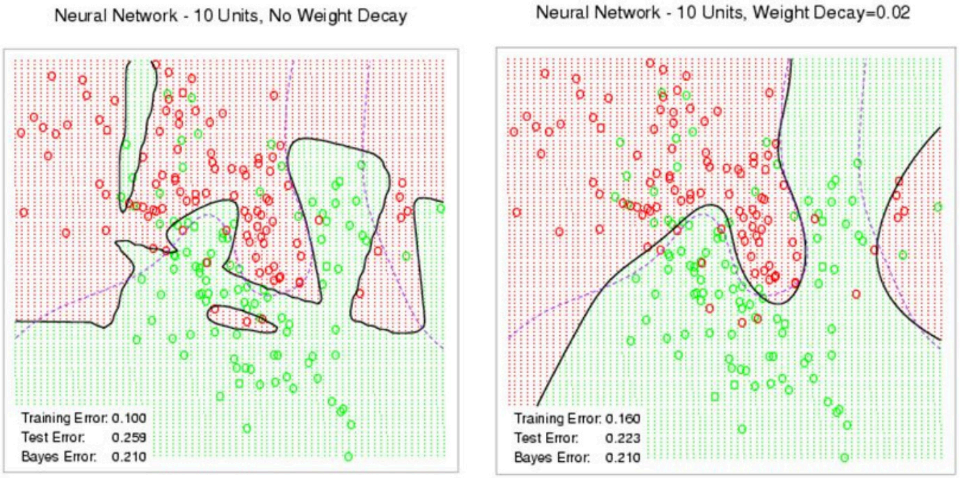
权重衰退

Weight Decay，用于处理过拟合的工具

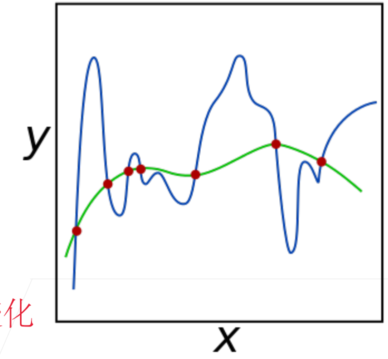


使用均方范数作为硬性限制

通常不会直接用

- 通过限制参数值的选择范围来控制模型容量

$$\min \ \ell(\mathbf{w}, b) \quad \text{subject to} \quad \|\mathbf{w}\|^2 \leq \theta$$



通过sita来限制模型的参数，即限定参数在某个范围内变化

- 通常不限制偏移 b （限不限制都差不多）
- 小的 θ 意味着更强的正则项

即对模型的限制更强。极端情况下w全为0，只剩下一个偏置b

使用均方范数作为柔性限制

- 对每个 θ，都可以找到 λ 使得之前的目标函数等价于下面

$$\min \ \ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

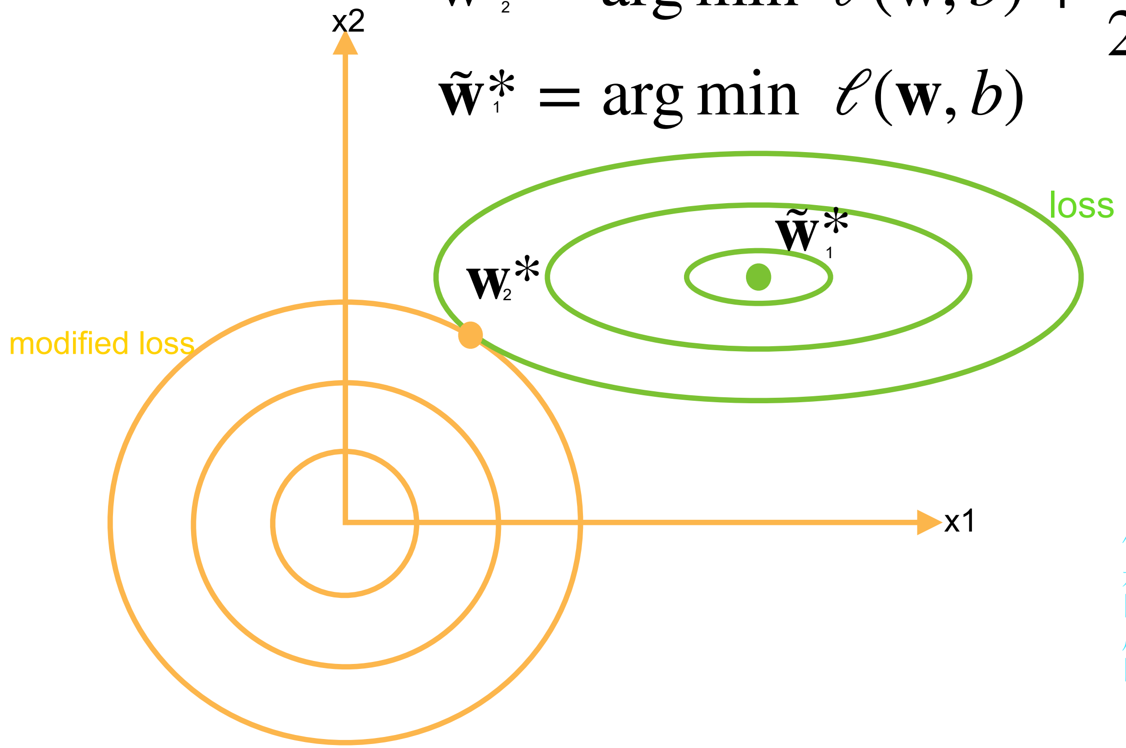
这个更常用，但之前的硬性限制更好理解

  - 可以通过拉格朗日乘子来证明
- 超参数 λ 控制了正则项的重要程度
  - $\lambda = 0$：无作用
  - $\lambda \to \infty, \mathbf{w}^* \to \mathbf{0}$ 相当于sita趋于0

演示对最优解的影响

$$\mathbf{w}^*_2 = \arg\min \ \ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\tilde{\mathbf{w}}^*_1 = \arg\min \ \ell(\mathbf{w}, b)$$



由图，w1本来是原loss的最优点，但是loss
经过修改后，w1不是修改过的loss的最优点。

这时，我们回忆一下L2loss的梯度分布，离原点越近越小。
所以，我们可以变相把梯度理解为拉扯力。此时modified loss
对w的拉扯强于loss，所以最后会在w2处形成平衡。

总体来看，增加了lamda对图中的影响就是把w向原点拉扯，
即w的大小变小了，模型复杂度随之降低，减少了过拟合的
风险。

但是，如果本来函数的最优点就是w1呢？这实际上
是不可能的，因为如果最优点就是w1，那么有噪音
的数据，训练出来的最优解一定在w1附近而不是w1，
所以我们还是需要λ，把训练出来的结果往理论最优
的地方拉扯。

参数更新法则

- 计算梯度

$$\frac{\partial}{\partial \mathbf{w}}\left(\ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2\right) = \frac{\partial \ell(\mathbf{w}, b)}{\partial \mathbf{w}} + \lambda \mathbf{w}$$

- 时间 t 更新参数

$$\mathbf{w}_{t+1} = (1 - \eta\lambda)\mathbf{w}_t - \eta \frac{\partial \ell(\mathbf{w}_t, b_t)}{\partial \mathbf{w}_t}$$

  - 通常 $\eta\lambda < 1$，在深度学习中通常叫做权重衰退

权重衰退——L2正则化：
简单理解为，在原来梯度下降的基础上，
每次在w向loss最低处走一步之前，先把它
往上拉一小段，再让w走下一步。

总结

L2loss: 平方，L1loss：绝对值

- 权重衰退通过 L2 正则项使得模型参数不会过大，从而控制模型复杂度
- 正则项权重是控制模型复杂度的超参数

在神经网络里， y=w1x1 + w2x2 + w3x3 + b 有两种减少复杂度的形式，一是减少x的数量，二是减小w的大小

一般我们采用后者，因为输入特征是定死的，而且特征的弃留抉择比较困难。