

你了解了如何使用感知器来构建 AND 和 XOR 运算，但它们的权重都是人为设定的。如果你要进行一个运算，例如预测大学录取结果，但你不知道正确的权重是什么，该怎么办？你要从样本中学习权重，然后用这些权重来做预测。

要了解我们将如何找到这些权重，可以从我们的目标开始考虑。我们想让网络做出的预测与真实值尽可能接近。为了能够衡量，我们需要有一个指标来了解预测有多差，也就是**误差 (error)**。一个普遍的指标是误差平方和 sum of the squared errors (SSE)：

$$E = \frac{1}{2} \sum_{\mu} \sum_j [y_j^{\mu} - \hat{y}_j^{\mu}]^2$$

这里  $\hat{y}$  是预测值  $y$  是真实值。一个是所有输出单元  $j$  的和，另一个是所有数据点  $\mu$  的和。这里看上去很复杂，但你一旦理解了这些符号之后，你就能明白这是怎么回事了。

首先是内部这个对  $j$  的求和。变量  $j$  代表网络输出单元。所以这个内部的求和是指对于每一个输出单元，计算预测值  $\hat{y}$  与真实值  $y$  之间的差的平方，再求和。

另一个对  $\mu$  的求和是针对所有的数据点。也就是说，对每一个数据点，计算其对应输出单元的方差和，然后把每个数据点的方差和加在一起。这就是你整个输出的总误差。

SSE 是一个很好的选择有几个原因：误差的平方总是正的，对大误差的惩罚大于小误差。同时，它对数学运算也更友好。

回想神经网络的输出，也就是预测值，取决于权重

$$\hat{y}_j^{\mu} = f(\sum_i w_{ij} x_i^{\mu})$$

相应的，误差也取决于权重

$$E = \frac{1}{2} \sum_{\mu} \sum_j [y_j^{\mu} - f(\sum_i w_{ij} x_i^{\mu})]^2$$

我们想让网络预测的误差尽可能小，权重是让我们能够实现这个目标的调节旋钮。我们的目的是寻找权重  $w_{ij}$  使得误差平方  $E$  最小。通常来说神经网络通过**梯度下降**来实现这一点。



