

到它，而其他时候根本不关联。这称为条件相关，或者选择相关。

## 6.19 选择性相关的秘密

当值低于 0 时“关闭”节点。

这看起来太简单了，好像不会产生什么作用，但是不妨考虑一下：如果一个节点的值低于 0，通常情况下这个节点仍然与输入具有一定的相关性。只是这个相关性的值恰好是负的而已。但是如果节点为负的时候“关闭”它(将其设置为 0)，那么它与任何输入的相关性都会为 0。

这意味着什么？现在，节点可以在它想要的时候选择性地关联相应的对象。这使得它可以完成下面这种操作：“**让**我和左边的输入完全相关，但是只有当右边的输入被关闭的时候。”它是如何做到这样的呢？好吧，如果左边输入的权重值为 1.0，右边输入的权重值为一个大负数，这种情况下，同时接受左边的和右边的输入将导致节点的值一直为 0。但是，如果只“打开”左边的输入，节点将仅接受左边输入的值。

这在以前是不可能的。早些时候，中间节点要么始终与某个输入相关，要么始终不相关。现在它可以是条件相关的。终于，它可以表达自己的意见了。

**解决方案：**通过在任何中间节点为负时关闭它，你能够允许神经网络选择性接受来自不同输入的相关性。这对于两层神经网络来说是不可能的，因此，三层网络具有更强的能力。

这种“如果节点为负，则将其设置为 0”的逻辑有一个漂亮的术语：“非线性”。没有这项调整的话，神经网络将是线性的。也就是说，如果没有这项技术，输出层能够选择的相关性和它在两层网络中可选的相关性没有什么区别。它直接与输入层相关，就意味着它不能解决新的路灯数据集的问题。

非线性变换有很多种。但这里讨论的是许多情况下最好的方法。这也是最简单的方法(叫作 `relu`)。

值得注意的是，大多数其他书籍和课程都告诉我们矩阵乘法是一个线性变换。我觉得这并不直观。这也使我们更难理解非线性所带来的贡献——为什么会选择非线性而不是线性的模型(后文会详述)。它讲的是，“如果没有非线性变换，那么两个矩阵相乘可能等价于一个矩阵”。我的解释虽然不是最简洁的答案，却是对为什么需要非线性的一个直观解释。