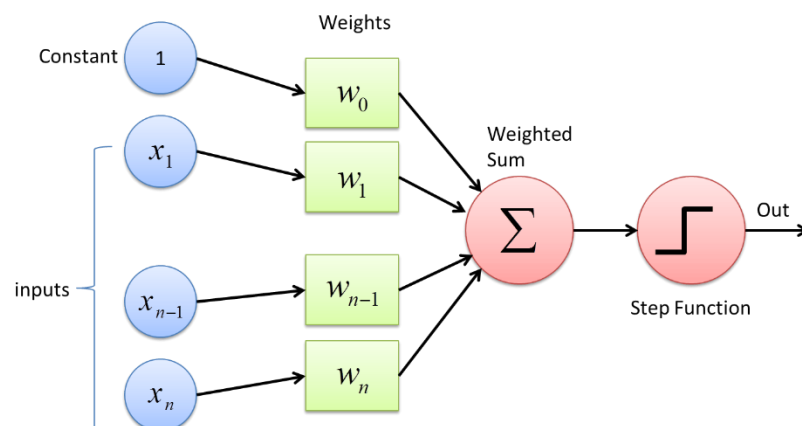


TERMS: Neural network, not neutral network.
Derivation equals to “origin”
Inhibiting – prevent/ slow something down

PERCEPTRON

Perceptron: 感知器，就是用来处理数据的一个节点。它由 4 部分组成：输入信息或输入信息层、权系数和阈值、算术求和、激励函数。

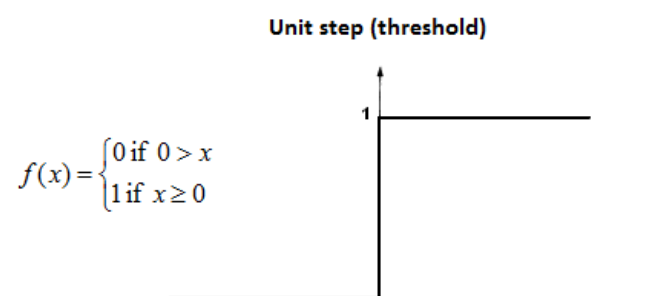
图片来源于网络：



注解：这幅图中没有表示出阈值，阈值应在权系数之后。

输入信息经过“权重求和”后，流向激励函数节点。激励函数将会根据激励函数的处理准则，处理流入的信息。通常，激励函数的准则是将输入信息一一映射到(0, 1)或者其他固定区间范围内。比如，如果激励函数是下图中的形式，那么激励函数将根据其输入信息的正负，输出 1 或者 0。在这个例子中，可以看出激励函数的二分功能。

图片来源于网络：



感知器是单个节点，从感知器接受信息开始，到信息流向激励函数之前，信息其实在接受一个一次多项式的数学处理，也即线性变换输入信息。当完成线性变换后，再由激励函数完成广义上的二分功能，所以感知器也被称为线性二分器。

WHY PERCEPTRON

输入信息中每个元素代表着现实世界中的一个维度,考虑这样由这样一组元素组成的输入信息:(时刻, 经度, 纬度, 地球在公转平面上的位置)。物体在未来的某个时刻, 出现在地球在公转平面上的某个位置时的某个经度和某个纬度上。输入信息中的 4 个元素互不影响, 也就是说, 输入信息中的 4 个元素描述了高维空间中的一个点, 由空间原点指向该点的一条向量确定了该点。感知器中对输入信息的线性变换, 其实是输入信息向量和权系数向量的点积。

$$\sum_{i=1}^n x_i \omega_i = (x_1, x_2, \dots, x_n) \cdot (\omega_1, \omega_2, \dots, \omega_n)$$

假设输入信息向量和权系数向量的均是单位长度的向量, 则二者的点积等于两条向量空间夹角的余弦值。由余弦函数的性质可知, 两条向量的空间夹角越小, 两者的点积越大。因此, 感知器的线性变换计算, 实际上是在高维空间中测量权系数向量和输入信息向量靠近程度。之后, 两个向量间的“靠近程度”信息将作为激励函数的输入。

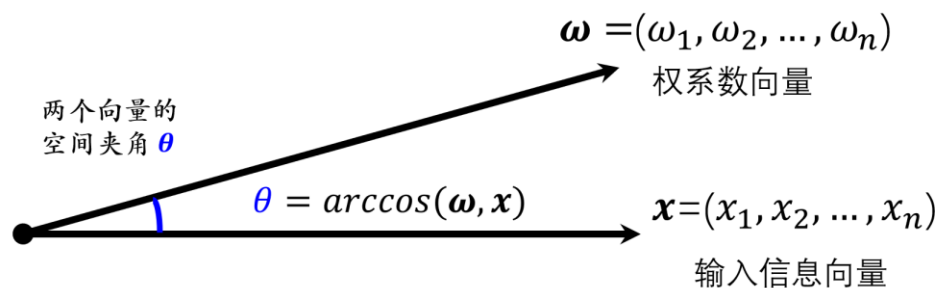


FIGURE. 感知器通过线性变换计算权系数向量和输入信息向量之间的高维空间夹角

THE HEMMING NET

同一层次中, 有很多个感知器同时和输入信息相连。信息流连接着多个感知器和单独的一组输入信息, 这样的信息网络结构称为 hemming net。(像缝衣服一样, 把信息缝起来, 衣服缝好了会有条线痕, 缝制信息的线痕就是网络中的连接线, 通过连接就把信息关系像两块布一样固定住了。可以先这么理解 hemming net)

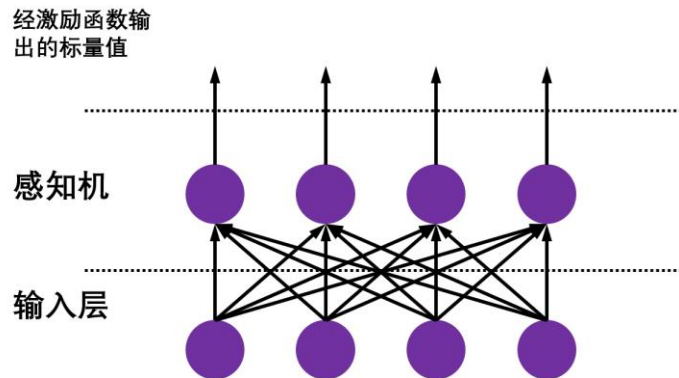


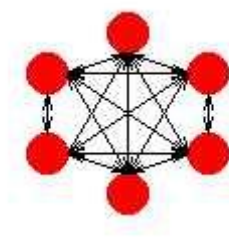
FIGURE. HEMMING NET 示意图

Hemming net 的输出是各个感知器的输出，即将两个向量的“靠近程度”作为激励函数的输入，从激励函数处获得的输出。不同的感知器有不同的权系数设置，多个感知器就有多个权系数设置。也就是说，不同的感知器的互异的权系数设置，决定了有的感知器和输入信息向量夹角更小，有的感知器和输入信息向量的夹角更大（注解：感知器就是一组权系数设置状态，它并不能生产出权系数，权系数是需要被指定给感知器的）。所以，hemming net 得到了所有感知器与输入信息向量的“靠近程度”分布情况。如何挑选出最“靠近”的那个呢？也就是如何挑选出最大的余弦值。当然可以人工比较数值大小完成挑选，也可以使用排序算法。但是需要注意，我们要在网络的框架下完成对最大值的搜寻，否则将失去机器学习的意义。对于 neural network 而言，可以使用对抗网络找到对抗网络输入信息中的最大值（在这里，对抗网络的输入信息是标量）。

THE MAXNET

Maxnet 是一种节点间互相连接的网络。节点通过向其他节点发送抑制信号的方式，和其他节点竞争。原理如图所示。

图片来自网络：



节点和其他节点间的连接权系数是负值，通过下列算法，除了初值最大的节点之外，其他所有节点收敛至 0。对抗网络最终将找到值最大的节点。

图片来自网络：

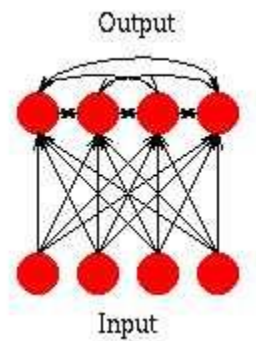
```
Let the value of node i be Xi
until (Xi doesn't change) {
    foreach i {
        
$$X_i = X_i + \sum_{j \neq i} \omega_{ij} X_j$$

        where  $\omega$  is negative
    }
}
```

PUT HEMMING NET AND MAXNET TOGETHER

将 hemming net 和 maxnet 融合，即在 hemming net 的感知器间互相建立连接。Hemming net 负责完成对权系数向量和输入信息向量的“距离计算”，maxnet 负责筛选出感知器的最大输出，也即对应着最近空间距离的最大余弦值。

图片来自网络：



先训练，后用于分类

可以划定一个群集范围，当权系数向量均落在群集范围内时，训练可以结束。对于新出现的输入信息，将使用训练出的权系数配置在网络中计算。Hemming net 给出新输入信息和每个权系数向量的夹角信息，Maxnet 将突出和新输入信息空间距离最近的权系数配置，从而将新输入信息归类到这个权系数配置对应的输入类别中去。

训练的目的是为了得到和输入信息特征向量同样指向的权系数向量，而输入信息的特征向量指向应是固定的，故权系数向量也应是确定的指向，训练过程中权系数向量产生的偏差小于精度要求时，可以认为训练收敛。

训练目的：控制权系数配置，使其和同一特征下的输入信息的空间夹角足够近。从而对于同一特征下的其他输入信息能够通过内积运算尽可能给出 $\vec{a} \cdot \vec{b}$ 的最大值，从而便于使用 maxnet 进行分类。

参考资料：

<https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>

<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/competitive.html>