

什么是神经网络？

(中学数学程度可懂)

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

人工智能中最重要的 3 大技术是：

- 逻辑
- 神经网络
- 进化

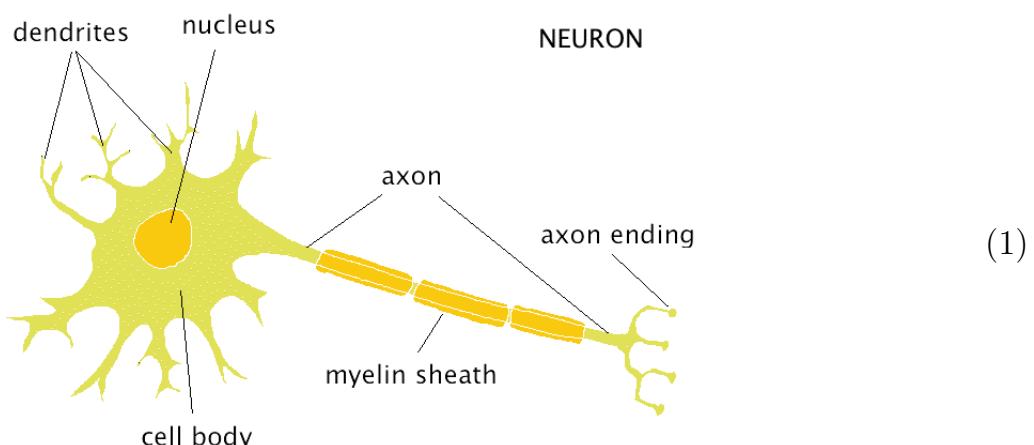
神经网络属于统计学习 (statistical learning) 的一种，这些方法将 vector space 中的某些「点」分类。

Deep learning 的意思，简单来说，是「很多层的神经网络」。深度学习是目前人工智能中最备受瞩目的一种技巧。

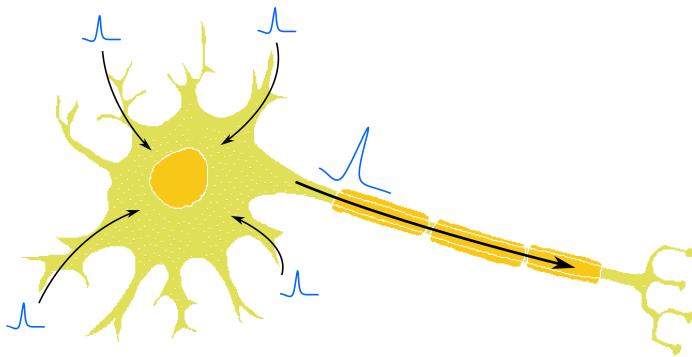
生物的神经细胞

温习一下在中学时学过的生物学☺

这是一粒真的神经元：

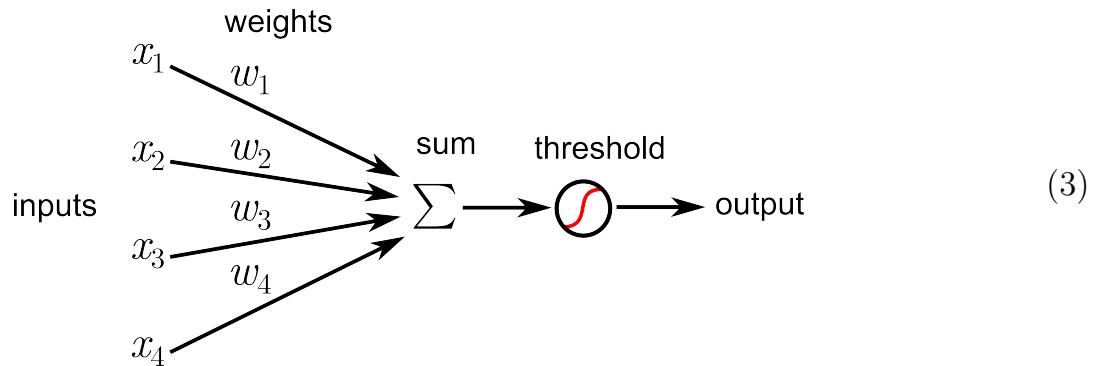


细胞周围的 dendrites 收集电信号，当这些脉冲信号的总和超过某一阀值 (threshold) 时，会发射 (fire) 一个电脉冲信号，从 axon 输出到另一神经元：



(2)

数学上我们将这过程极度简化，变成这样的模型 (model):



意思是：将每个输入加权 (weighted) 加起来，然后经过一个 \textcircled{S} 形状的函数输出。

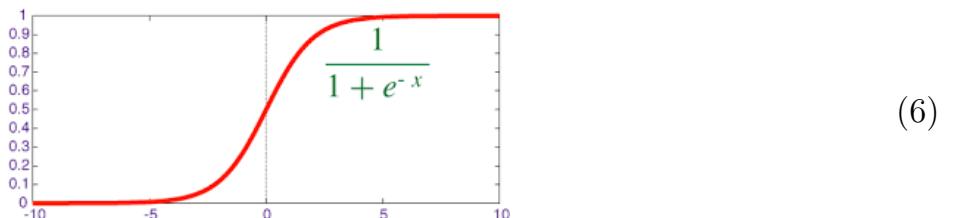
用数式表示：

$$\boxed{\text{output}} \quad y = \textcircled{S} \left[\sum_i (w_i x_i) \right] \quad (4)$$

其中 $\textcircled{S} = \text{sigmoid}$ 函数是：

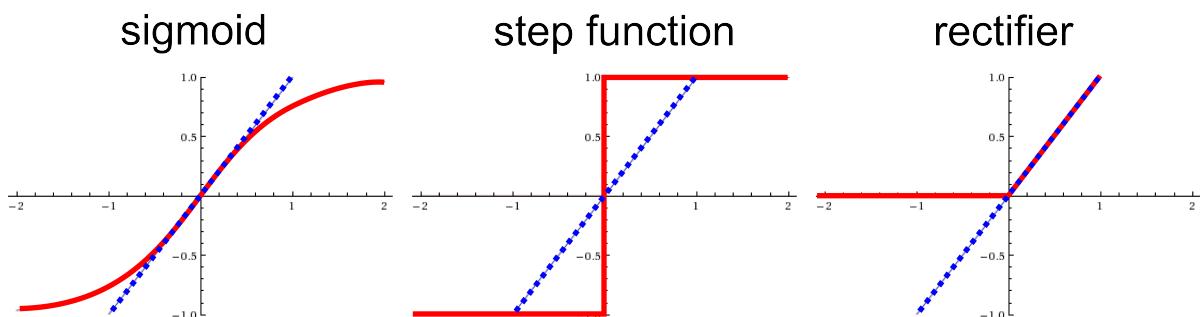
$$\textcircled{S}(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

它的形状是这样的：



它代表在左边没有信号 (0 = nothing)，在右边的信号值是 1 = “yes”。

其实也可以用以下这些函数模拟 threshold 的作用：



Fun fact #1

神经细胞的表面布满了 sodium-potassium (Na^+/K^+) channels，它们用 ATP (adenosine triphosphate, 细胞的能量来源) 以 $3 \text{ Na}^+ : 2 \text{ K}^+$ 的比例将 ions 「泵」到细胞内，造成电压差。这些蓄势待发的电压，当电压超过 threshold 时某些 channels 打开，造成 “action potential”。这个现象可以用微分方程描述，即著名的 Hodgkin-Huxley 方程，及其简化版本 FitzHugh-Nagumo 方程。

这 action potential 最特别的地方是：它是一个 **all-or-nothing effect**，亦即是说，如果输入总和低於阀值，则输出信号是平的 (zero)。为什么要这样呢？因为人的脑袋是由一些水母那样的多细胞生物进化出来的，这些细胞之间慢慢学会用电信号 communicate，但它们在一滩水那样的环境下通讯，噪音很大。直到现在人脑仍然是像一碗汤水那样的环境，而且人要活动，脑部不停有微小震动，造成 **heat noise**。为了要在噪音中运作，必须有机制将噪音抑制下去，这就是 all-or-nothing 的原因。换句话说，人脑的意识，其资讯是有限的，就像电脑一样，并没有什么神秘。

Fun fact #2

神经细胞的细胞膜是由 lipid bi-layer 构成，它的成份是脂肪和胆固醇 (cholesterol)。胆固醇的作用是稳固细胞膜结构，每粒细胞都需要胆固醇。脑里面的神经线全都是用细胞膜组成，所以脑基本上都是脂肪和胆固醇。尤其是中国人吃的猪脑，它的胆固醇含量是所有食物中最高的，高出鸡蛋很多倍！

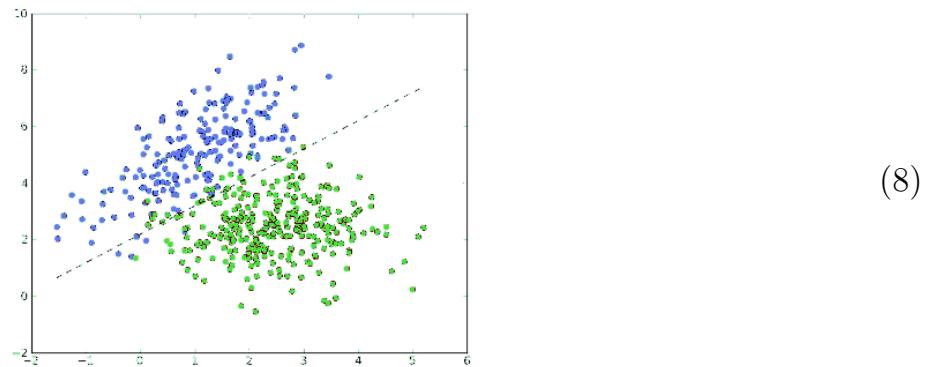
那层 myelin sheath 是脊椎类动物才有的，它好像电线外面的胶套，作用是加快神经传播的速度。八爪鱼是无脊椎动物，所以它的头很大，牠比其他同类聪明，但脊椎类动物的脑袋可以较细小也达到同样的智力水平。

Fun fact #3

神经信号传递到末梢 (synapse)，不再用电传递，而是用化学分子 neuro-transmitter。这些分子种类很多，例如抗抑郁药物常常提到的 serotonin 和 dopamine。但其实最常见的 neurotransmitter 是 glutamate，它是所有动物的神经系统的主要通讯分子。植物没有神经系统，所以植物里面没有 glutamate。人类喜欢吃肉，所以进化出对肉类的味觉，特别喜欢 glutamate 的味道。有个日本科学家发现了海藻内有一种物质，加进食物中可以做到肉的鲜味效果。其实这种物质就是 glutamate，也就是「味精」。所以味精其实对人体没有害，只是经常吃味精而不吃真的肉，会导致营养不均衡。

一粒神经元的几何解释

以前说过，机器学习的目标通常是将空间上的某些「点」分类：



例如，在机器视觉中，一张图像可以有几万个 pixels，每个 pixel 是一个维度，它的颜色就是这个维度上的坐标值。整个空间就是所有图像的空间，每一点是一个图像，这种空间的维数很高（维数就是一张图像的 pixels 个数）。通常我们讲解时用 2 维或 3 维空间，读者们要运用想像力幻想一下很高维的情况。

在中学数学中有学过，一条直线的方程是这形式的：

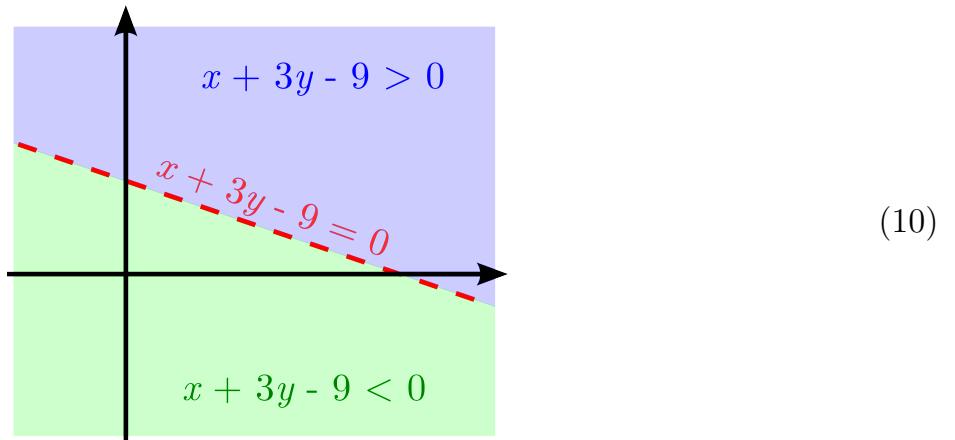
$$ax + by + c = 0$$

constants
variables

(9)

The diagram shows the general equation of a straight line with red annotations. Red lines connect the terms a , b , and c to the words "constants" above the equation, and connect the variables x and y to the word "variables" below it.

它的几何解释是这样的：

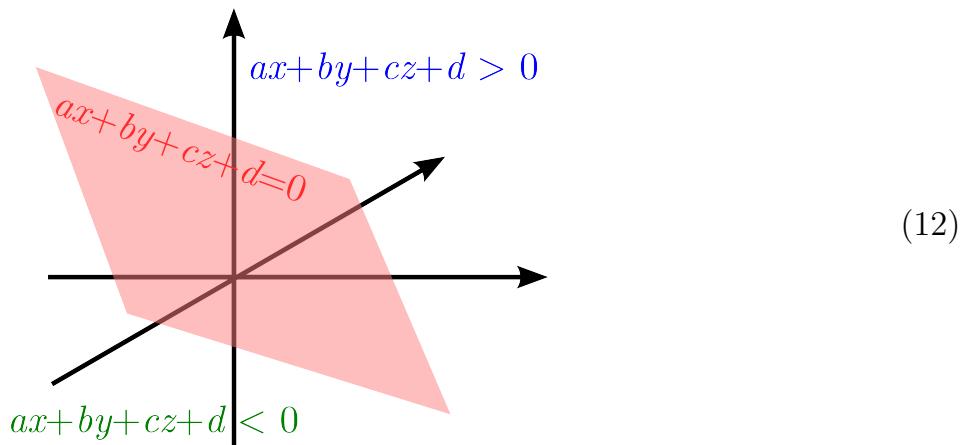


在直线上，那直线的方程 $= 0$ 。直线将它身在的空间分割成两半：一边 > 0 ，另一边 < 0 。

推广到 3 维空间的情况，我们有一个平面的方程：

$$ax + by + cz + d = 0 \quad (11)$$

它同样将空间分割成两半，一半 > 0 ，另一半 < 0 ：



在任意 n -维空间，每一点记作 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ，一个超平面 (hyper-plane) 将该空间分割成两半，它的方程是：

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n + a_0 = 0 \quad (13)$$

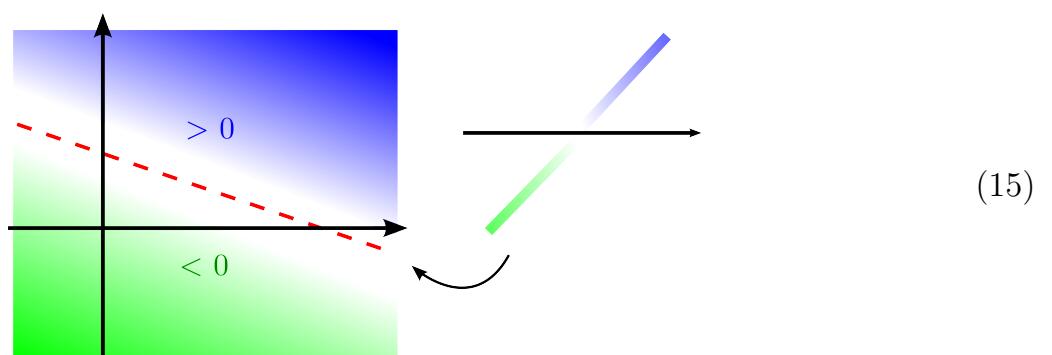
注意：一块超平面的维数是多少？在平面空间里它是一条线（1度空间），在立体空间里它是平面（2度空间），一般来说，在 n -维空间里的超平面是一个 $n - 1$ 维的物体，($n - 1$) 又叫作 co-dimension 1，意思是说 ambient space 的维数是 n ，方程 (13) 减少了一个自由度 (degree of freedom)，所以服从这等式的物体内，只有 $n - 1$ 个自由度。

现在可以看到超平面和神经元之间有些相似，因为神经元在未经过 \textcircled{J} 之前，就是一个 线性组合：

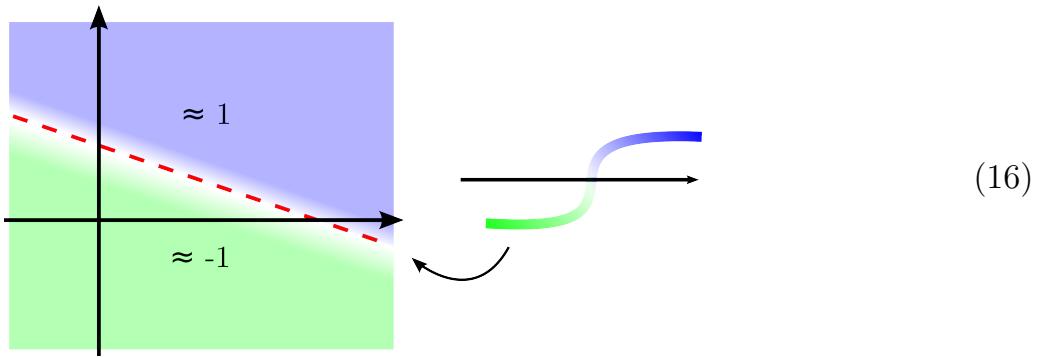
$$\begin{array}{c} \text{linear combination} \\ \boxed{\text{output}} \quad y = \textcircled{J} \quad [\overbrace{\sum_i (w_i x_i)}] \end{array} \quad (14)$$

换句话说：每粒神经元构成一超平面，它将空间切割成两半。

加了 \textcircled{J} 之后怎样？未有 \textcircled{J} 时，分割的两边分别是 > 0 和 < 0 ，如果将颜色看作是「强度」，强度是逐渐变化的：（右边表示从侧面看，立体）

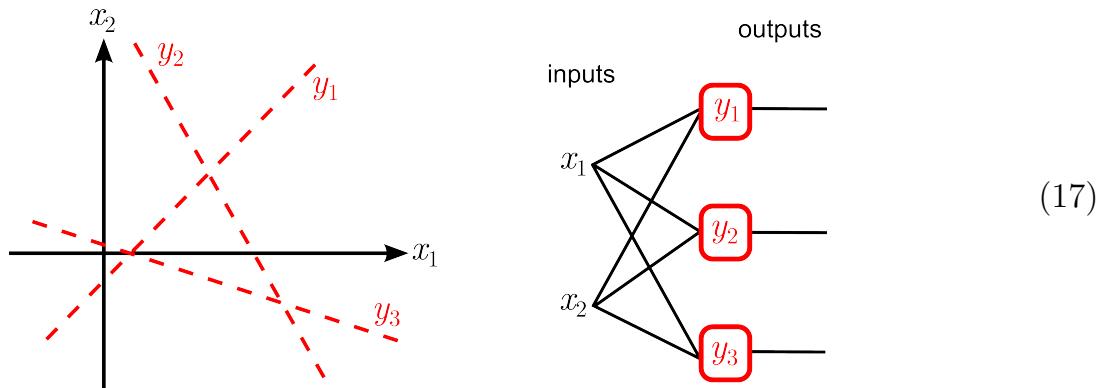


加了 sigmoid 之后，用 1 代表「yes」，0 代表「no」，则两边的对比加强了，亦即更两极化、「非此即彼」：



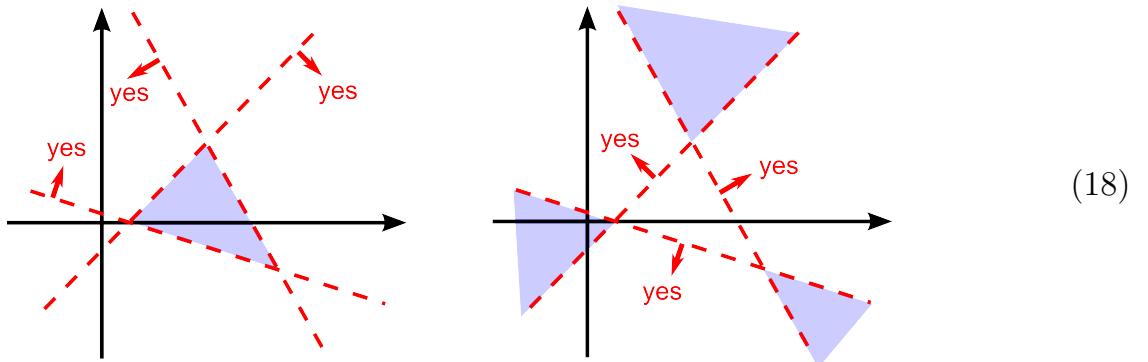
> 1 粒神经元的几何解释

如果有 > 1 粒神经元（在同一层上），例如 3 粒：



注意：座标是 (x_1, x_2) = 输入，输出是 (y_1, y_2, y_3) ，分别用 3 条虚线代表。这一层神经网络的 network topology 如 (17) 右图所示（每粒神经元没有画出 \sum 和 \bigcirc ）。

每粒神经元可以选择某一边为「yes」，这些选择的 conjunction 可以形成不同的形状，例如以下两种：



亦注意在右图中，可以出现几个 disjoint regions（在空间上分离的）。

很明显，可以用神经元将输入空间的点进行切割和分类，达到机器学习的目的。

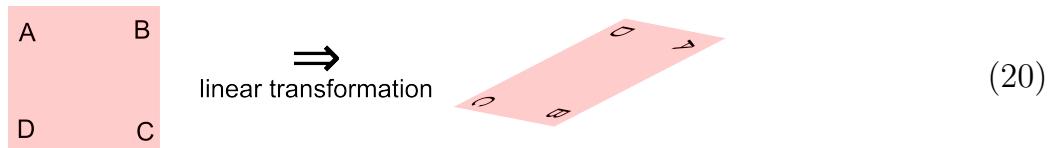
一层神经网络的几何解释

一层神经网络的数学形式是：

$$\mathbf{y} = \textcircled{O}[\mathbf{W}\mathbf{x}] \quad (19)$$

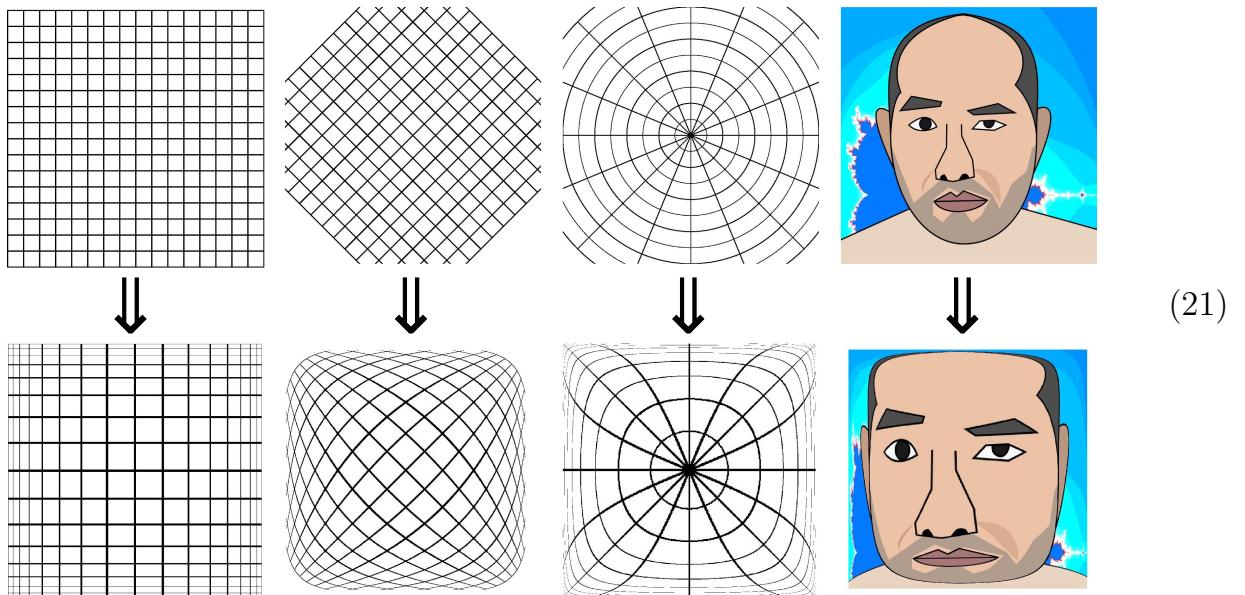
其中 \mathbf{W} 是矩阵，亦即 线性变换； \textcircled{O} 是 非线性变换。

首先要明白什么是 linear transformation：

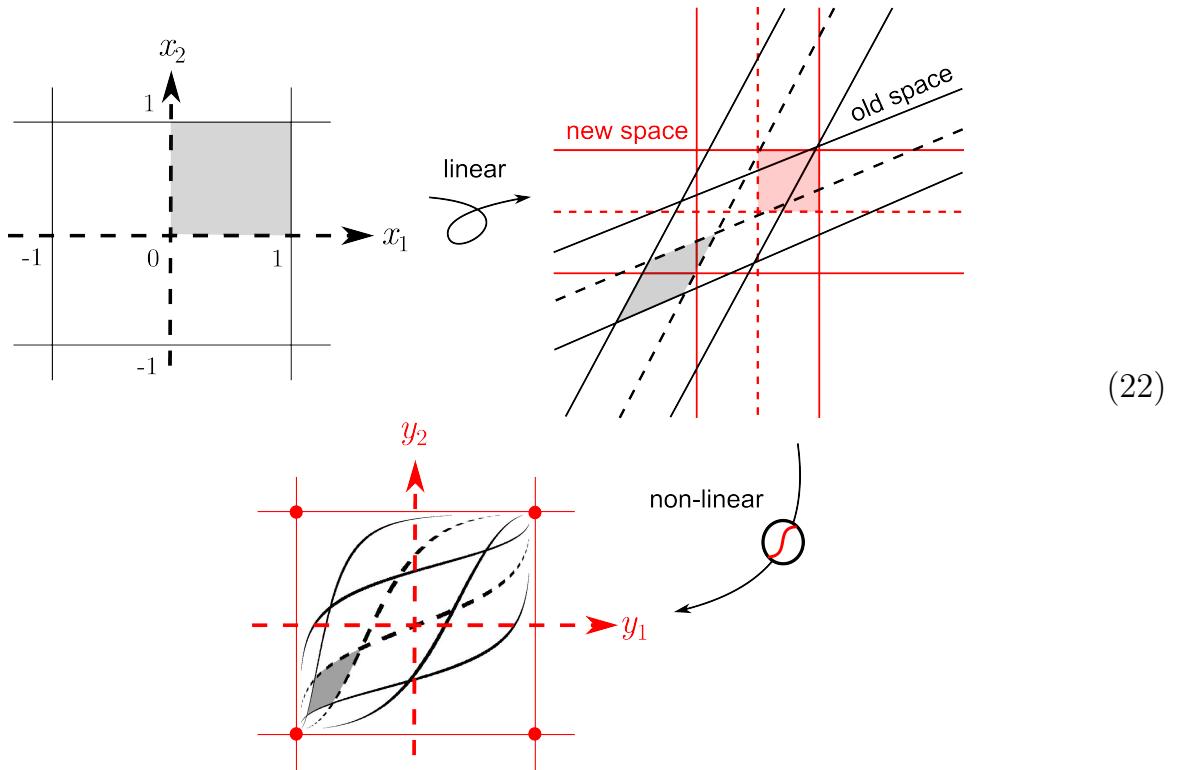


线性变换可以将正方形「扯」成平行四边形，也包括位置上的旋转 (rotation) 和平移 (translation)。但直线仍变为直线，故名。

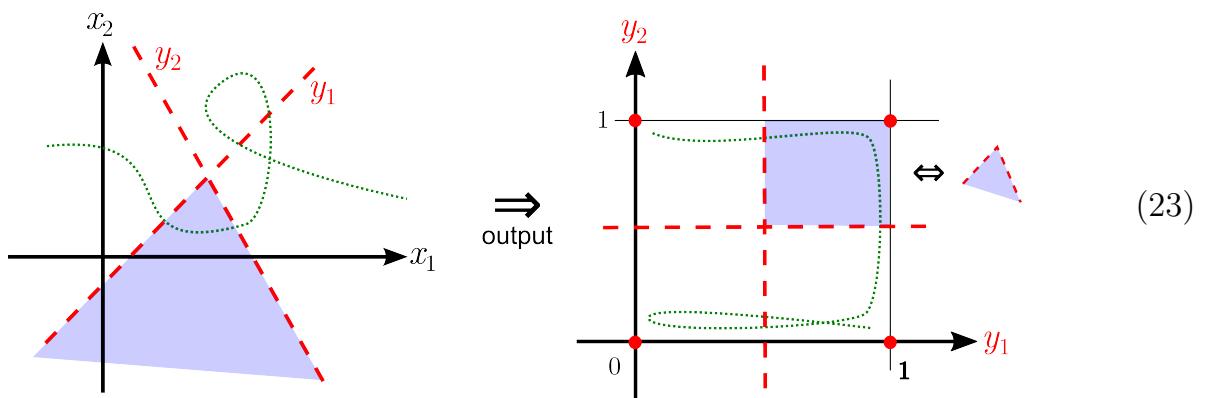
另外要明白 \textcircled{O} 变换的形状：（以下是 x 和 y 座标都被 sigmoid 变换）



可以这样理解 \textcircled{O} ：它将平面上的点「扯向」4个顶点，顶点是吸引子 (attractors)，所以我变了「国字口面」。换句话说，hyper-cube 内部和外部的点，均被吸引到它的顶点上，这些顶点代表各种 yes / no 组合。



为简化讨论，现在只考虑 2 粒神经元的输入和输出空间（都是 2 度空间的平面）：



因为 $\textcircled{1}$ 的缘故，输出的位置会趋近 hyper-cube 的那些 顶点 (•)。这些顶点对应於输入空间中被割开的 regions。例如蓝色那块 region 对应於：

$$(y_1 = \text{yes}, y_2 = \text{yes}) \Rightarrow (1, 1) \quad (24)$$

当输入位置随绿色线游荡时，输出会在 hyper-cube 的顶点之间跳来跳去。你可能觉得这样移动很无聊（因为顶点个数不多），但当神经元的个数 n 增加时，hyper-cube 的顶点个数会以 2^n 的速度增长。

多层神经网络的几何解释

— 完 —