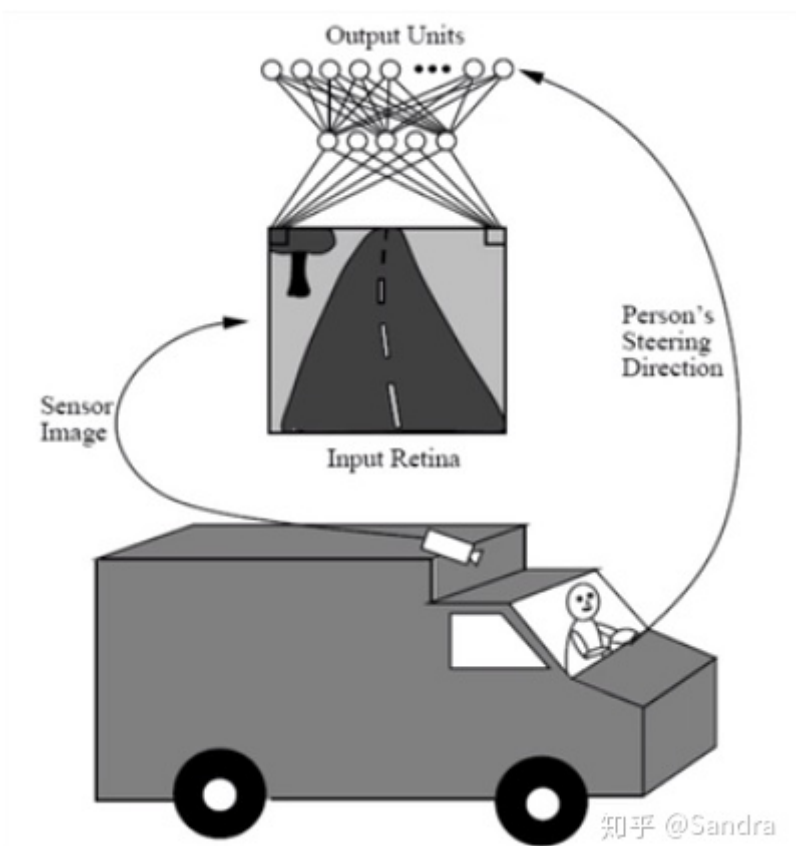
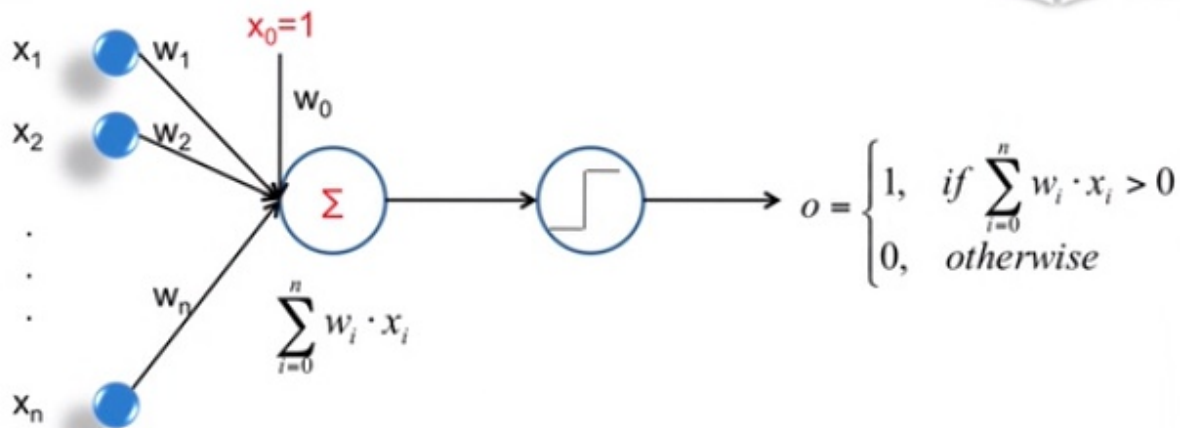


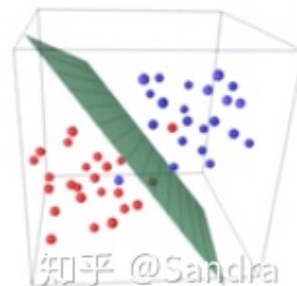
神经网络是有输入，有输出，各个神经元之间有联系的一个模型。
无人车的机器学习：



输入是图片（并记录司机的操作模式，作为训练数据），输出是一排神经元
简单的神经元：感知机（一个神经元，线性分类器！） --> 空间的超平面

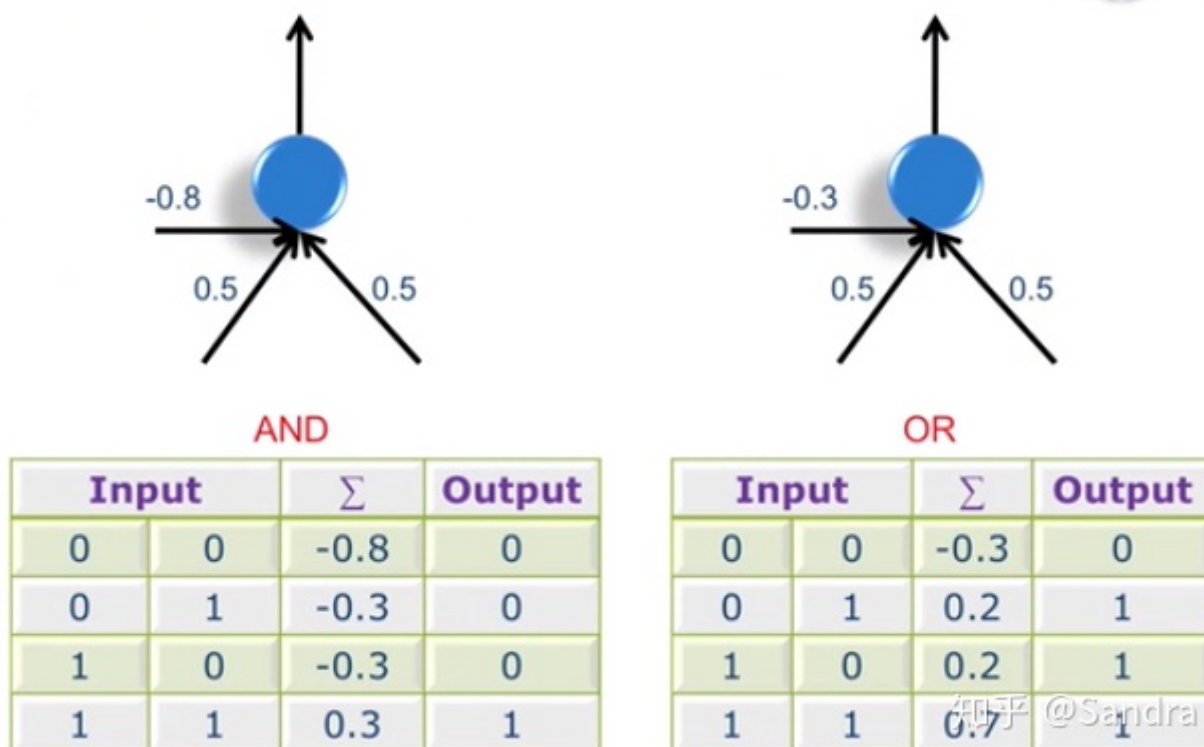


$$o(x_1, \dots, x_n) = \begin{cases} 1, & \text{if } w_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n > 0 \\ 0, & \text{otherwise} \end{cases}$$



w表示权重 (n+1个权重) , 要与输入x做一个内积 (x0始终等于一, w0可以看做是一个偏值, 没有w0平面始终经过原点)

神经元的逻辑层面实现:



与门、或门 (不要忘记w0, 如果忘记最后结果可能一直都不收敛)

感知机的权重设置:

期望将误差收敛至最小 --> 梯度下降法

$$E(\vec{w}) \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \quad \leftarrow \text{Batch Learning}$$

知乎 @Sandra

将每个神经元的误差 (期望输出-实际输出) 累计起来, 调整权重, 1/2为了求导
已知误差求偏导, 可以知道该增加或减小权重

$$\nabla E(\vec{w}) \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right]$$

$$w_i \leftarrow w_i + \Delta w_i \quad \text{where} \quad \Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Learning Rate

负号的含义: 当误差对w求偏导为正数 (随着w增加, 误差会增加) 的时候, 目的是减小这个误差 (减小权重) 学习率: 是为了控制每次调整的幅度, 防止调整过

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id}$$

$$= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2$$

$$= \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \quad o(x) = w \cdot x$$

$$= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - w \cdot x_d)$$

$$= \sum_{d \in D} (t_d - o_d) (-x_{id})$$

知乎 @Sandra

为了好求导，假设节点是线性的（输入=输出），用 o 代替。门线在0的地方倒数不存在，其余的地方导数是0，是不好计算的。

Δw_i

（调整的分量）的理解：假设现在期望输出是1，实际输出小， $t-o>0$ ，将如何修改权重呢？这取决于当前输入 x ，若 $x>0$ ，还想让其更大，所以要增加权重

GRADIENT_DESCENT (*training_examples*, η)

- ❖ Initialize each w_i to some small random value.
- ❖ Until the termination condition is met, Do
 - Initialize each Δw_i to zero.
 - For each $\langle x, t \rangle$ in *training_examples*, Do
 - Input the instance x to the unit and compute the output o
 - For each linear unit weight w_i , Do
 - $\Delta w_i \leftarrow \Delta w_i + \eta(t-o)x_i$
 - For each linear unit weight w_i , Do
 - $w_i \leftarrow w_i + \Delta w_i$

知乎 @Sandra

是一种批处理的学习！他只是调整了调整的分量（修改分量，不修改 w ），累积起来，最后整体做一次更新

$$w_i \leftarrow w_i + \Delta w_i \quad \text{where} \quad \Delta w_i = \eta(t - o)x_i$$

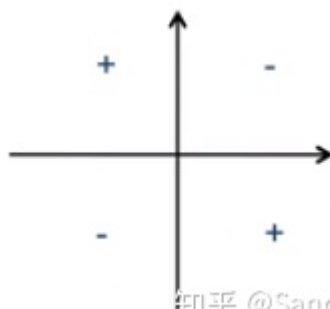
For example, if $x_i=0.8$, $\eta=0.1$, $t=1$ and $o=0$

$$\Delta w_i = \eta(t-o)x_i = 0.1 \times (1-0) \times 0.8 = 0.08$$

知乎 @Sandra

及时更新

但图二的方式是无论如何都训练不出一个分界面的：线性不可分



知乎 @Sandra

例子：（一行一行的看，最后收敛至最终一个solution）

Input			Tar get	Initial Weights			Output					Error	Correction	Final Weights		
							Individual			Sum	Final Output					
x_0	x_1	x_2	t	w_0	w_1	w_2	C_0	C_1	C_2	S	o	E	R	w_0	w_1	w_2
							x_0 · w_0	x_1 · w_1	x_2 · w_2	$C_0 +$ $C_1 +$ C_2			$t-o$			
1	0	0	1	0	0	0	0	0	0	0	0	1	+0.1	0.1	0	0
1	0	1	1	0.1	0	0	0.1	0	0	0.1	0	1	+0.1	0.2	0	0.1
1	1	0	1	0.2	0	0.1	0.2	0	0	0.2	0	1	+0.1	0.3	0.1	0.1
1	1	1	0	0.3	0.1	0.1	0.3	0.1	0.1	0.5	0	0	0	0.3	0.1	0.1
1	0	0	1	0.3	0.1	0.1	0.3	0	0	0.3	0	1	+0.1	0.4	0.1	0.1
1	0	1	1	0.4	0.1	0.1	0.4	0	0.1	0.5	0	1	+0.1	0.5	0.1	0.2
1	1	0	1	0.5	0.1	0.2	0.5	0.1	0	0.6	1	0	0	0.5	0.1	0.2
1	1	1	0	0.5	0.1	0.2	0.5	0.1	0.2	0.8	1	-1	-0.1	0.4	0	0.1
1	0	0	1	0.4	0	0.1	0.4	0	0	0.4	0	1	+0.1	0.5	0	0.1
.
1	1	0	1	0.8	-0.2	-0.1	0.8	-0.2	0	0.6	1	0	0	0.8	-0.2	-0.1

threshold=0.5 learning rate=0.1

知乎 @Sandra