

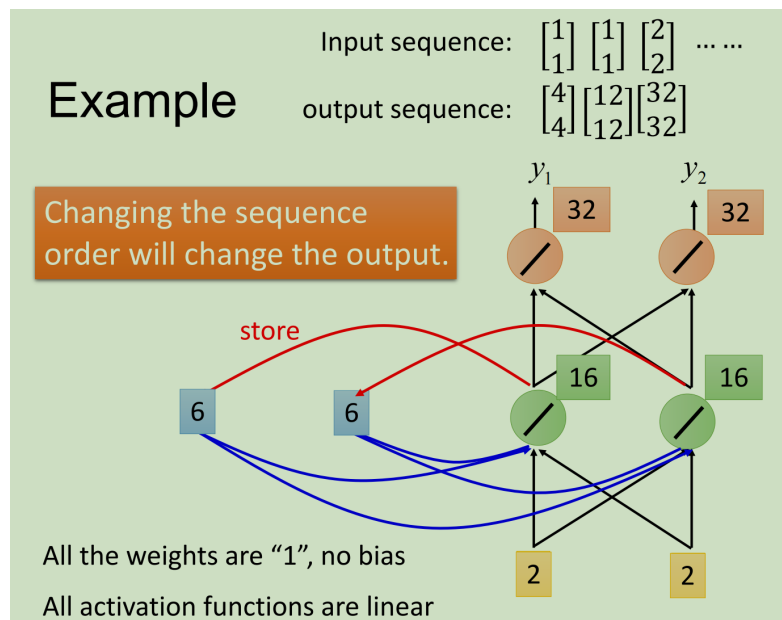
现在不用了，都用attention和transformer

运算过程

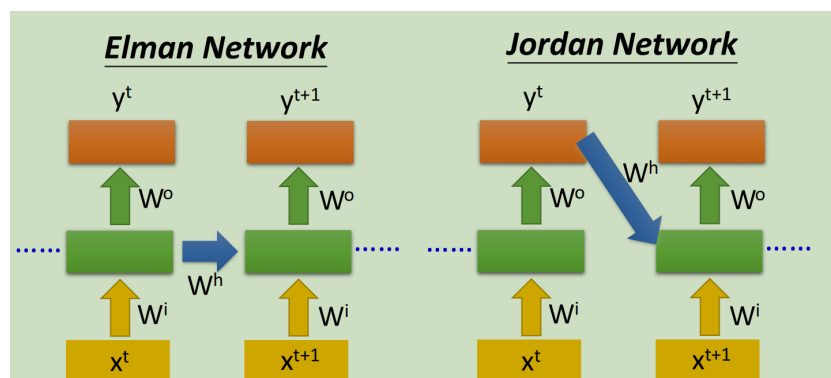
hidden值既output, 又存在memory中等待下一次一起输入

- memory初始化为0, 两个输入加两个memory乘上各自权重得到输出
- 把输出当作下一层的输入, 并存在memory中等待下一次输入, 又输入进当前层
- memory历程: 0, 2, 6, 16....

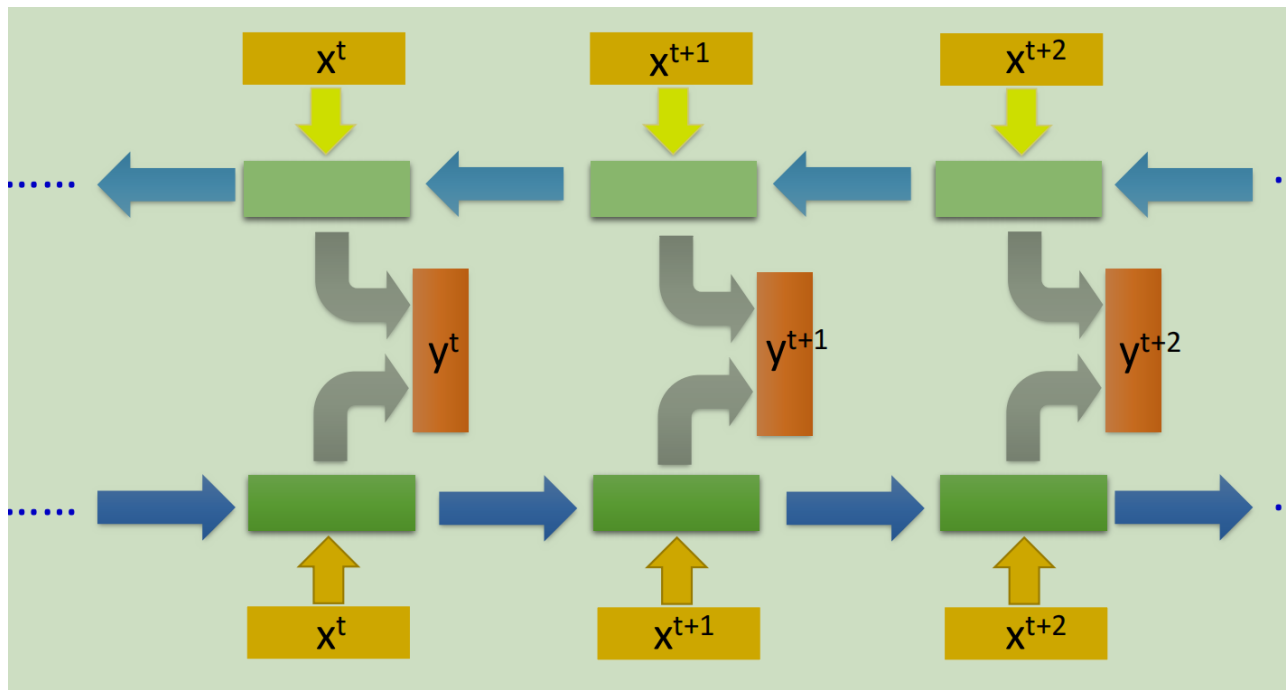
缺点：每次输入都把memory替换，迅速遗忘



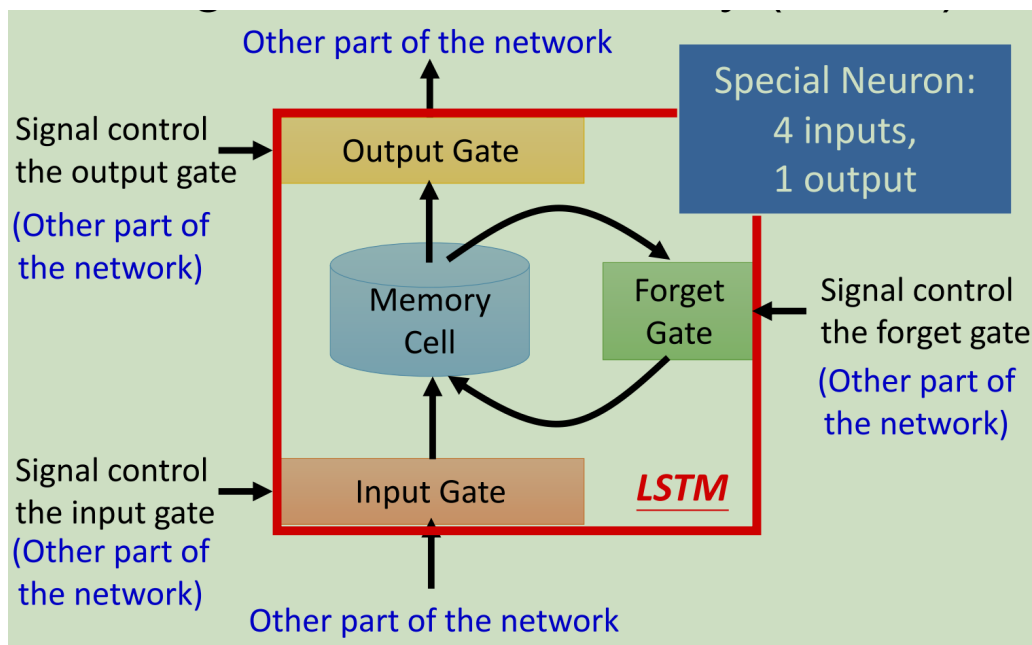
memory传入位置更换



Bidirectional RNN



LSTM



一个input, 三个signal控制信号可调节。共4个input, 一个output

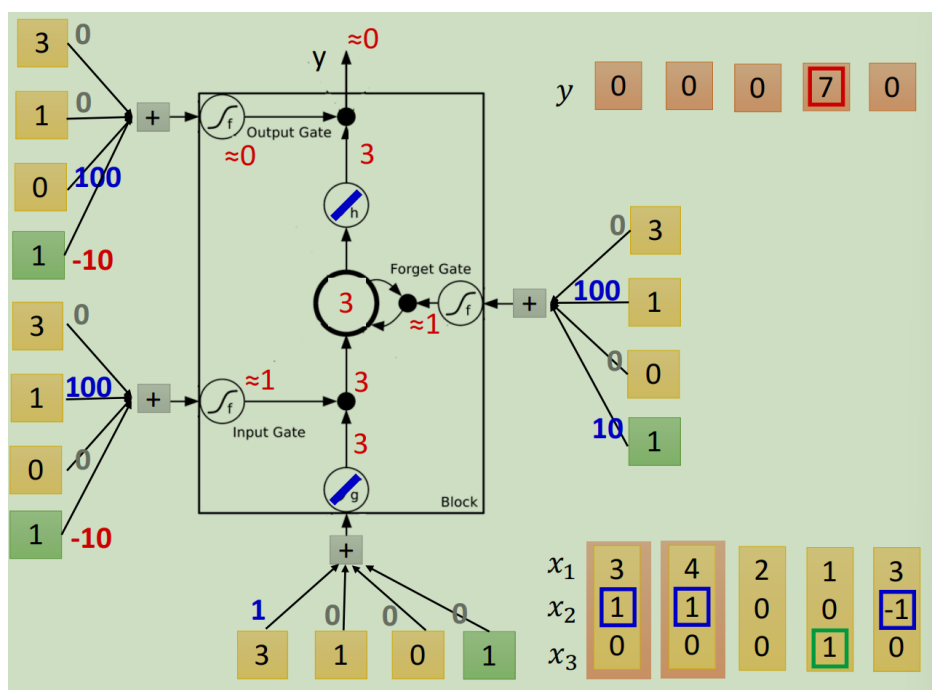
- 遗忘门控制哪些memory影响下一次输出
- 一些-1和1的特殊含义

	0	0	3	3	7	7	7	0	6
x_1	1	3	2	4	2	1	3	6	1
x_2	0	1	0	1	0	0	-1	1	0
x_3	0	0	0	0	0	1	0	0	1
y	0	0	0	0	0	7	0	0	6

When $x_2 = 1$, add the numbers of x_1 into the memory

When $x_2 = -1$, reset the memory

When $x_3 = 1$, output the number in the memory.



g 和 h 简化为 $y=x$, 即摆设

- 输入通过权 (1000) 计算出3
- 输入门输出90, 经sigmoid约等于1
- 所以最终输入为 $3*1=3$
- 遗忘门sigmoid出约等于1, 因此memory存入 $3*1=3$
- 输出们sigmoid出0, 因此输出0

计算慢, GPU加不起来, 序列越长越慢, 比attention慢, 因为必须序列按顺序一点点算

GRU

少一组参数, 把input和forget连起来

鲁棒性提高, 不容易过拟合