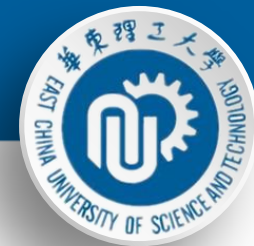


课时内容

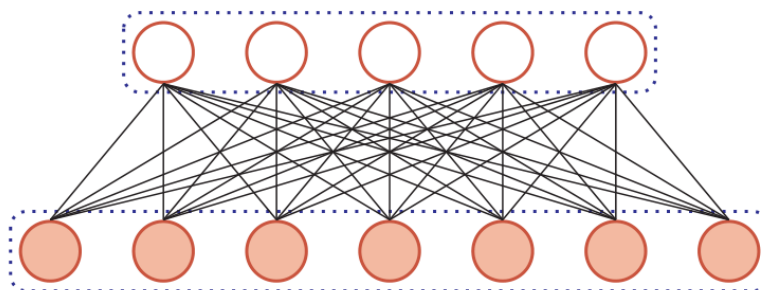
第9章 卷积神经网络



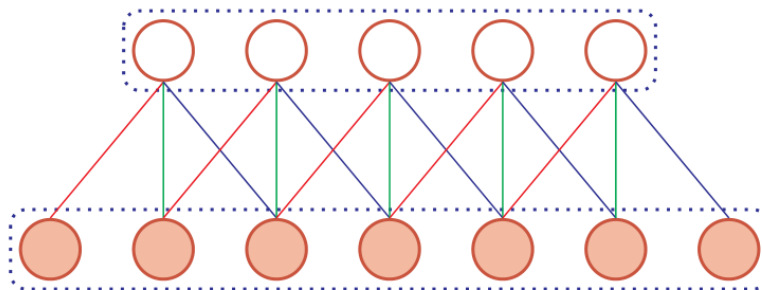


◆ 卷积神经网络

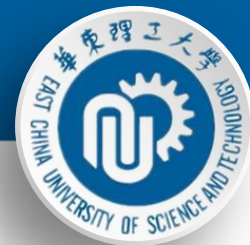
□ 用卷积层代替全连接层



(a) 全连接层



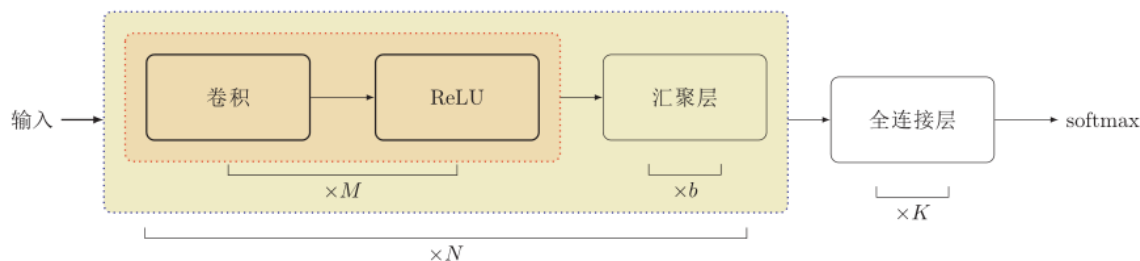
(b) 卷积层



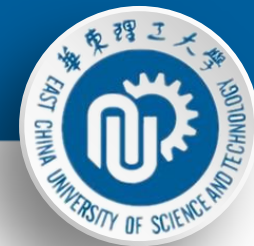
◆ 卷积网络结构

□ 卷积网络是由卷积层、子采样层、全连接层交叉堆叠而成。

□ 典型结构

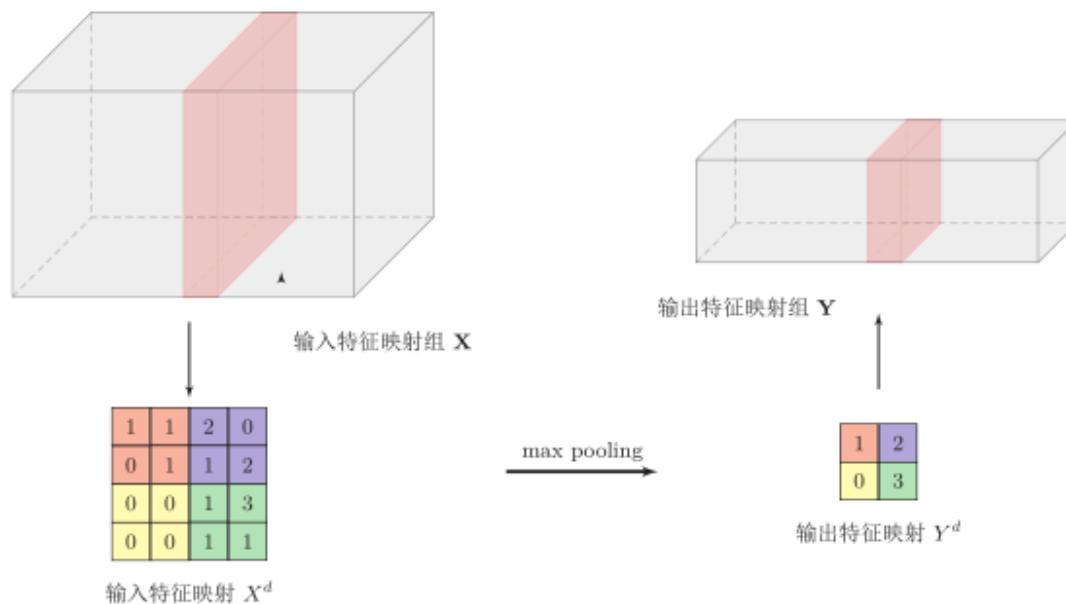


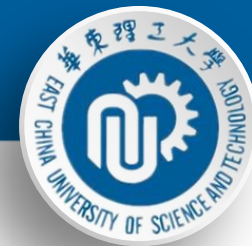
一个卷积块为连续 M 个卷积层和 b 个汇聚/池化层 (M 通常设置为 $2 \sim 5$, b 为0或1)。一个卷积网络中可以堆叠 N 个连续的卷积块, 然后在接着 K 个全连接层 (N 的取值区间比较大, 比如 $1 \sim 100$ 或者更大; K 一般为 $0 \sim 2$)



◆ 汇聚/池化层

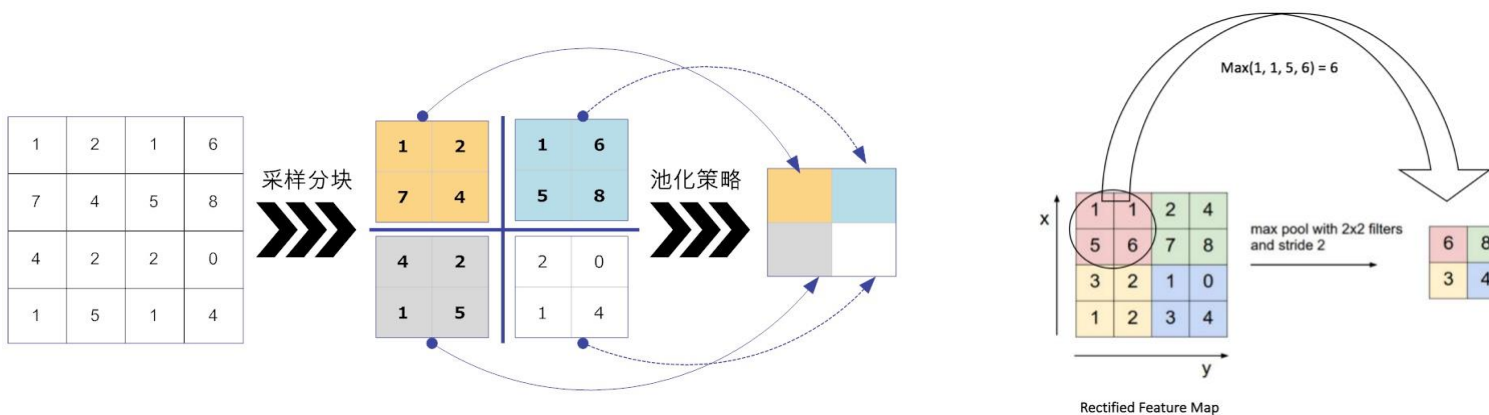
- 卷积层虽然可以显著减少连接的个数，但是每一个特征映射的神经元个数并没有显著减少。

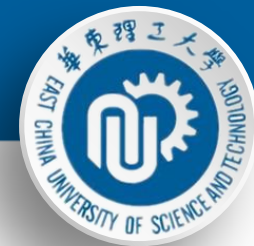




◆ 汇聚/池化层

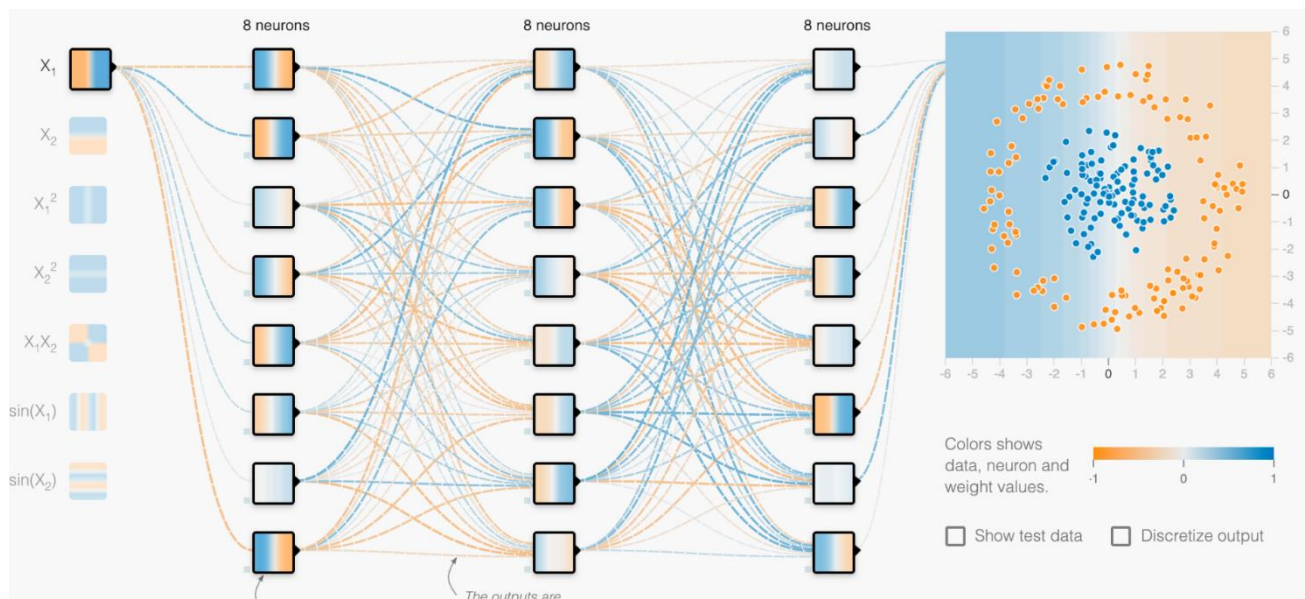
□ 采样的本质就是力图以合理的方式 “以偏概全”





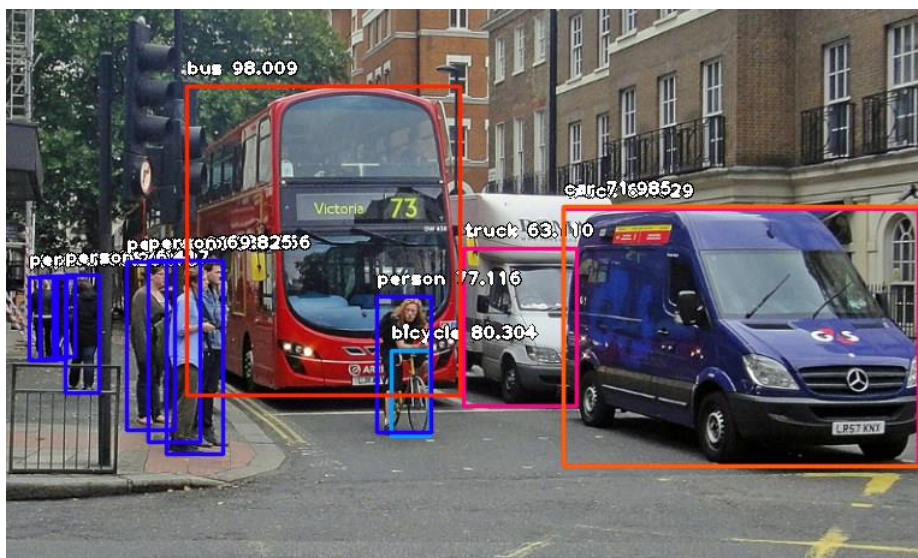
◆ 全连接层

- 全连接层(Dense Layer)目的:在于分类/回归等
- 前面若干层(卷积、激活、池化等)目的: 数据变换-->提取特征, 作为全连接层的输入为全连接层服务。



□ 目标检测

- 对输入图像进行分类的同时，检测图像中是否包含某些目标，并对他们准确定位并标识。

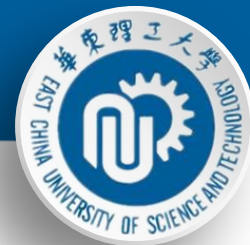


◆卷积神经网络的应用

□ 光学文字识别OCR

- 对文本资料的图像文件进行分析识别处理，获取文字及版面信息。

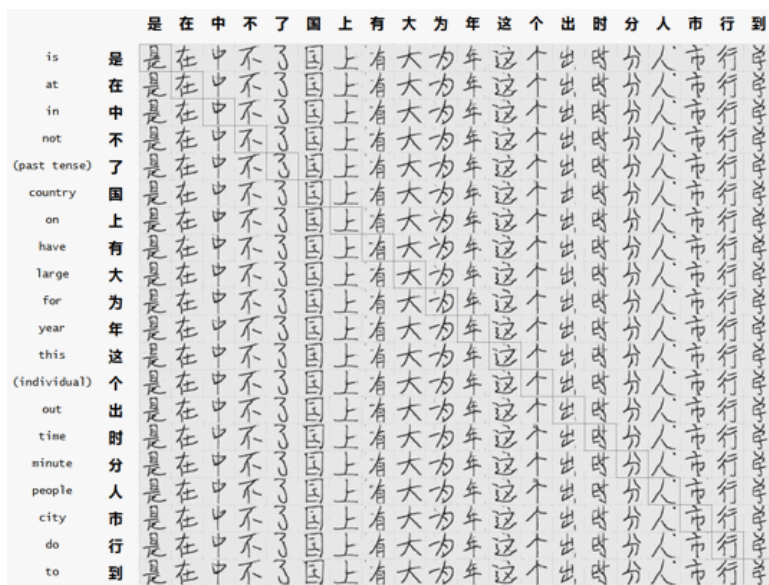


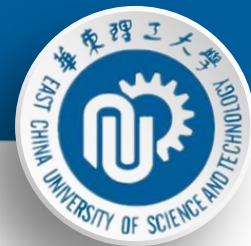


◆卷积神经网络的应用

□ 图像生成

- 生成准确和流畅的图像描述，而且能够通过可视化展示出它在生成每个词时关注的图片区域。



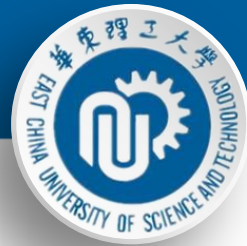


◆卷积神经网络的应用

□ 画风迁移

- 使用CNN将输入的内容图像与风格图像的目标特征提取，在此基础上输入噪声图像来组合之前提取的目标特征，并结合特征集来生成损失函数，之后在反复的参数迭代中完成对目标迁移图像的调整，最终把内容图像与风格图像二者的特征图结合起来。





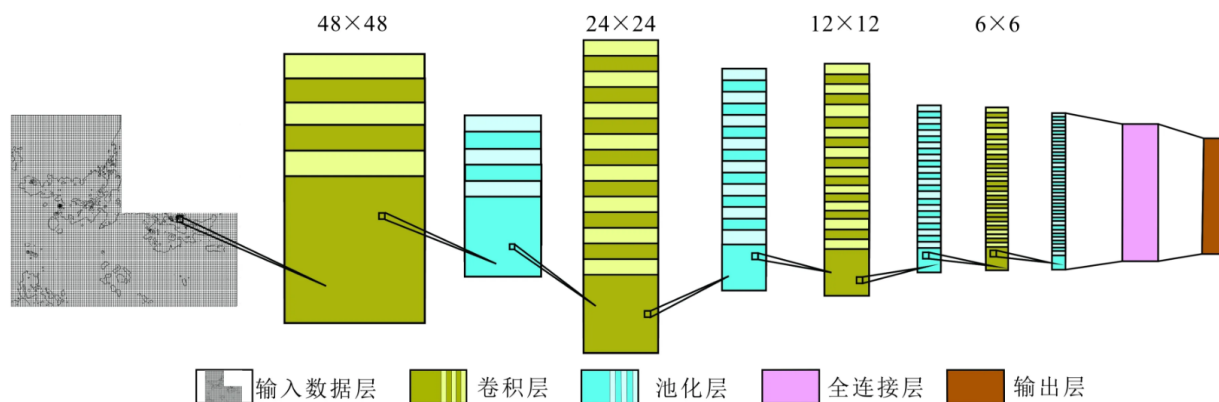
◆卷积神经网络的应用案例——找矿预测

□ CNN的结构

- 卷积神经网络的找矿预测模型主要由数据输入层、卷积层、池化层、全连接层和输出层 5 种网络层。
- 数据输入层是将网格化的化探和航磁数据作为输入数据写入神经单元，卷积层与池化层分别选择合适的激活函数完成对数据特征的提取和下采样，全连接层则是在网络末端实现特征的映射和分类，而输出层可用于结果输出或特征可视化。
- 找矿预测采用的卷积神经网络模型由 4 个卷积层、4 个池化层和 1 个全连接层组成。

◆卷积神经网络的应用案例—找矿预测

□ 找矿预测CNN的模型架构

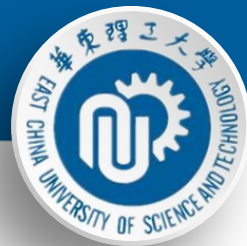


找矿预测卷积神经网络模型架构

课时内容

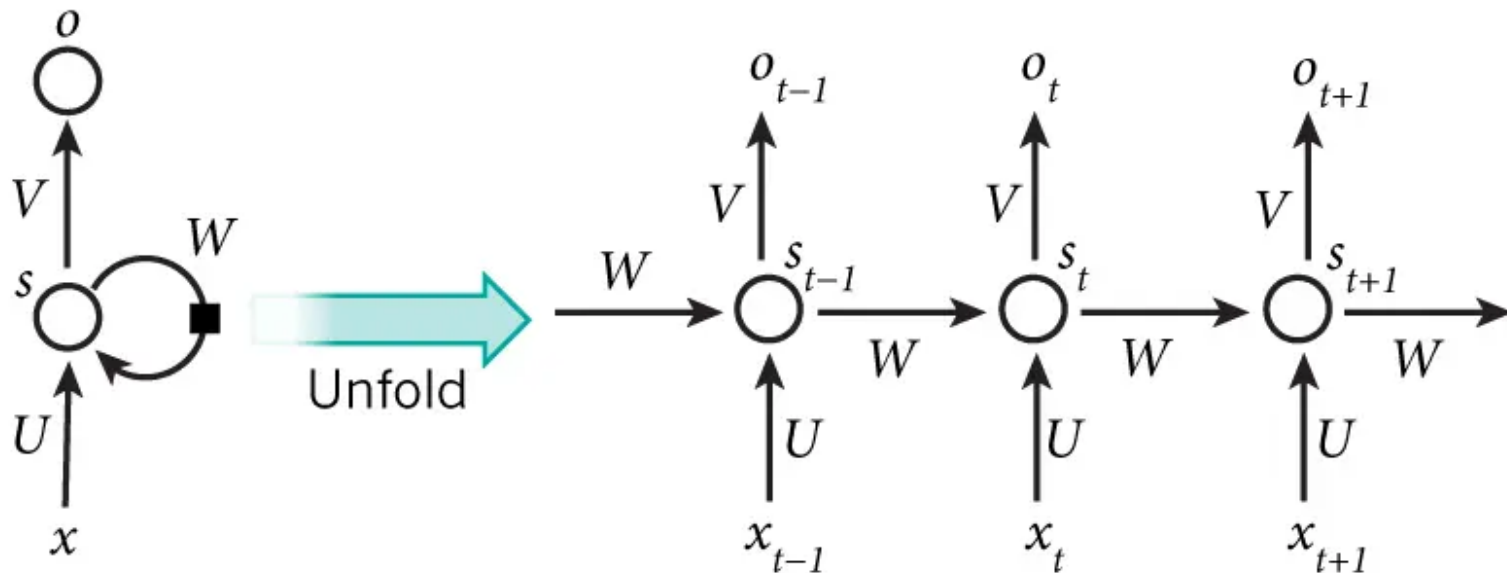
第10章 循环神经网络

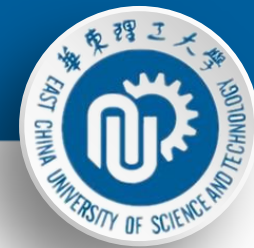




◆ 循环神经网络的结构图

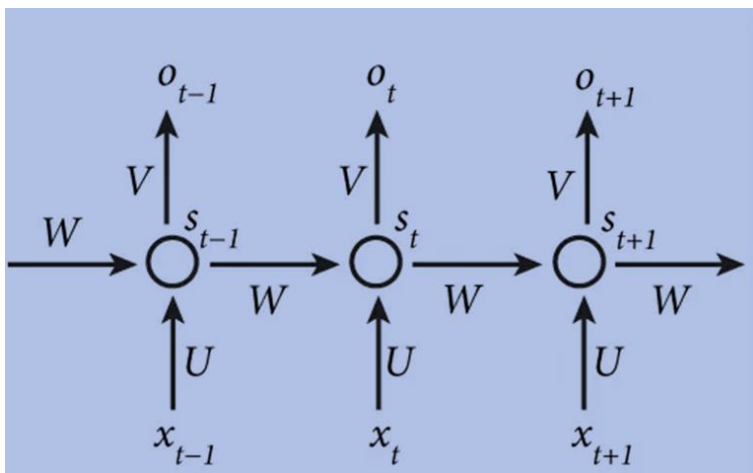
循环神经网络的当前 t 时间步状态受上一 $t - 1$ 时间步状态的影响，是能处理时序数据的神经网络。





◆ 循环神经网络的基本原理

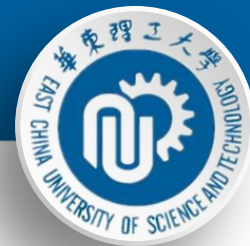
- 循环神经网络的神经元接受其他神经元和自身两部分信息，进而形成具有环路的网络结构，循环神经网络的公式如下：



$$O_t = g(V \cdot S_t)$$

$$S_t = f(U \cdot X_t + W \cdot S_{t-1})$$

S_t 的值不仅仅取决于 X_t ，还取决于 S_{t-1}



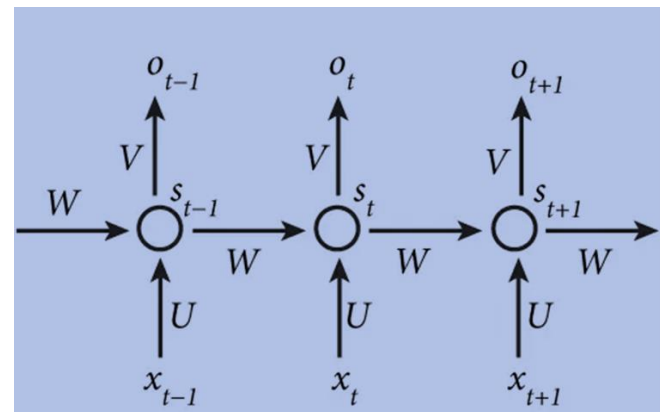
◆ 循环神经网络的计算方法

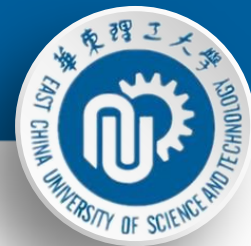
□ 循环神经网络的神经元计算公式如下：

$$\begin{aligned}o_t &= g(Vs_t) \\s_t &= f(Ux_t + Ws_{t-1})\end{aligned}$$

□ 反复把式 2 带入到式 1，将得到：

$$\begin{aligned}o_t &= g(Vs_t) \\&= Vf(Ux_t + Ws_{t-1}) \\&= Vf(Ux_t + Wf(Ux_{t-1} + Ws_{t-2})) \\&= Vf(Ux_t + Wf(Ux_{t-1} + Wf(Ux_{t-2} + Ws_{t-3}))) \\&= Vf(Ux_t + Wf(Ux_{t-1} + Wf(Ux_{t-2} + Wf(Ux_{t-3} + \dots))))\end{aligned}$$



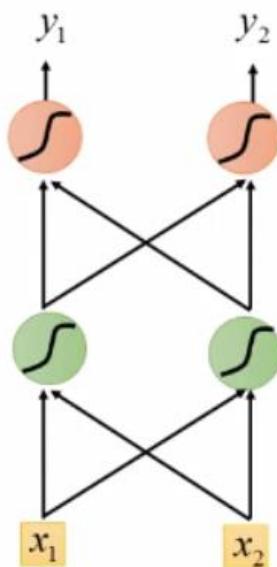


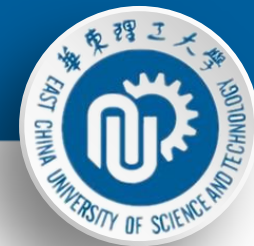
◆ 循环神经网络的计算示例

- 假设每个输入有两个，即输入层的维度是二维的；隐藏层也假设为二维，输出也假设是二维，所有权重的值都为1、没有偏差、所有激活函数都是线性函数，现在输入一个序列，到该模型中，一步步求解出输出序列：

Input sequence: $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} \dots$

- 循环神经网络模型如图：

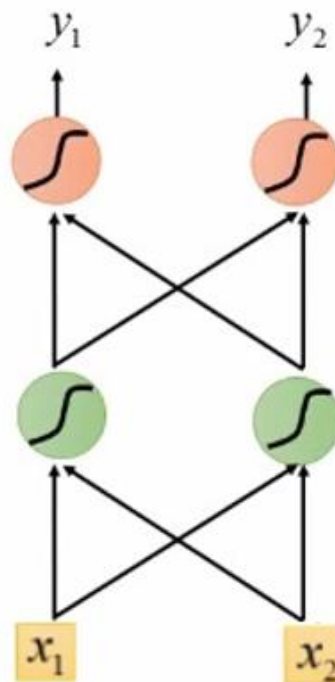


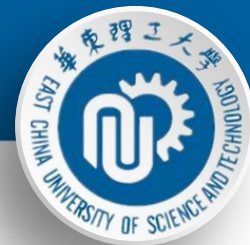


◆ 循环神经网络的计算示例

- 如图a1, a2可以看做每一时刻存下来的值, 初始时a1, a2是没有存值的, 因此都初始化为0:

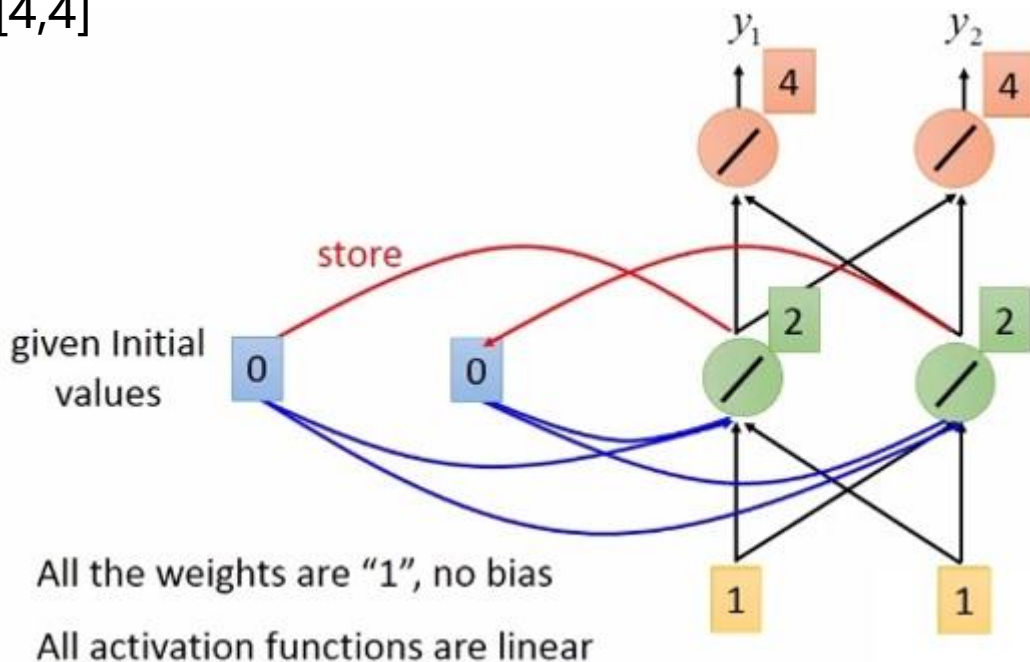
The output of hidden layer are stored in the memory.

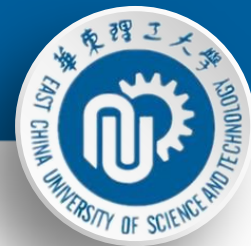




◆ 循环神经网络的计算示例

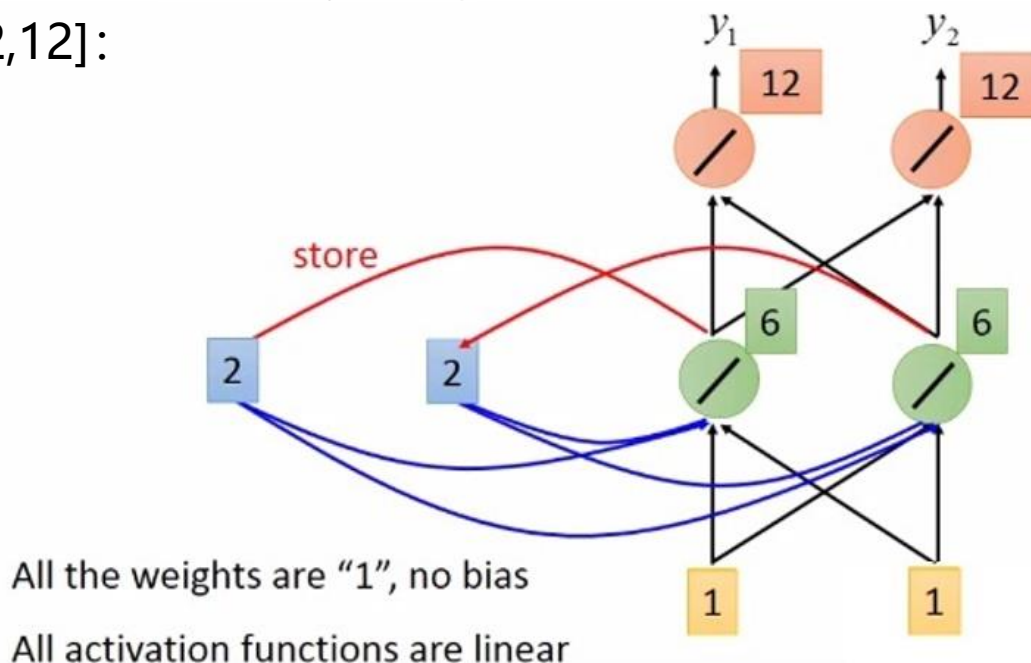
- 当我们输入第一个序列, $[1,1]$, 如下图, 其中隐藏层的值, 也就是绿色神经元, 是通过公式 $S_t = f(U \cdot X_t + W \cdot S_{t-1})$ 计算得到的, 因为所有权重都是1, 所以也就是 $1 * 1 + 1 * 1 + 1 * 0 + 1 * 0 = 2$ 输出层的值4是通过公式 $O_t = g(V \cdot S_t)$ 计算得到的, 也就是 $2 * 1 + 2 * 1 = 4$ 得到输出向量 $[4,4]$

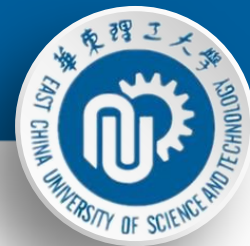




◆ 循环神经网络的计算示例

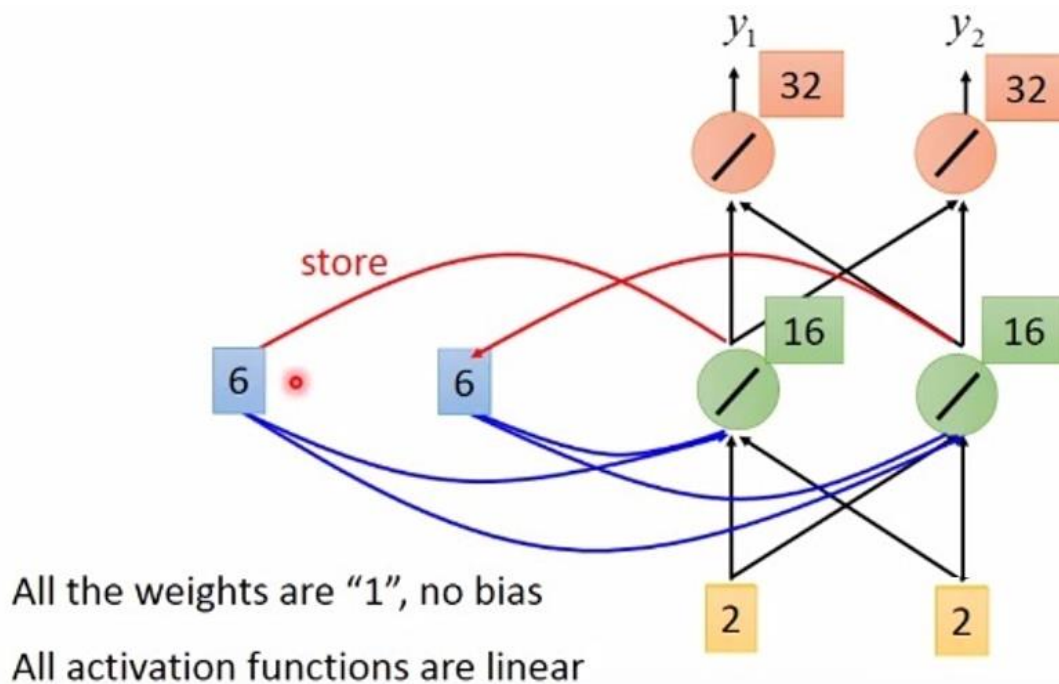
- 当[1,1]输入过后， a_1 ， a_2 的值不再是0，而是把这一时刻的隐藏状态放在里面，即变成了2，如图，输入下一个向量[1,1]，隐藏层的值通过公式
- $S_t = f(U \cdot X_t + W \cdot S_{t-1})$ 得到， $1 * 1 + 1 * 1 + 1 * 2 + 1 * 2 = 6$ ，输出层的值通过公式 $O_t = g(V \cdot S_t)$ ，得到 $6 * 1 + 6 * 1 = 12$ ，最终得到输出向量[12,12]:

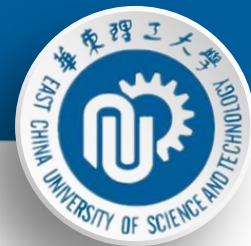




◆ 循环神经网络的计算示例

- 同理，该时刻过后 a_1 ， a_2 的值变成了6，也就是输入第二个 $[1,1]$ 过后所存下来的值，同理，输入第三个向量 $[2,2]$ ，如图，细节过程不再描述，得到输出向量 $[32,32]$ ：





◆ 循环神经网络的计算示例

□ 最终的输出序列为：

Output sequence: $\begin{bmatrix} 4 \\ 4 \end{bmatrix}, \begin{bmatrix} 12 \\ 12 \end{bmatrix}, \begin{bmatrix} 32 \\ 32 \end{bmatrix}$

□ 至此，一个完整的RNN结构遍历结束，可以看到，每一时刻的输出结果都与上一时刻的输入有关系，如果将输入序列换个顺序，那么得到的结果也将是是不相同的，这就是循环神经网络的特性，可以处理序列数据，同时对序列也很敏感。