本周学习的内容

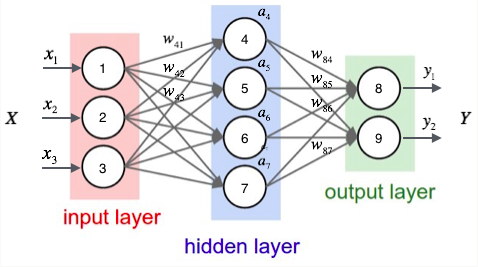
**学习了RNN**

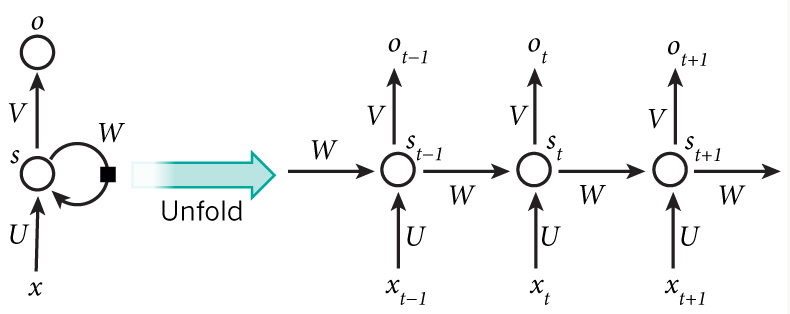
**1. 模型**

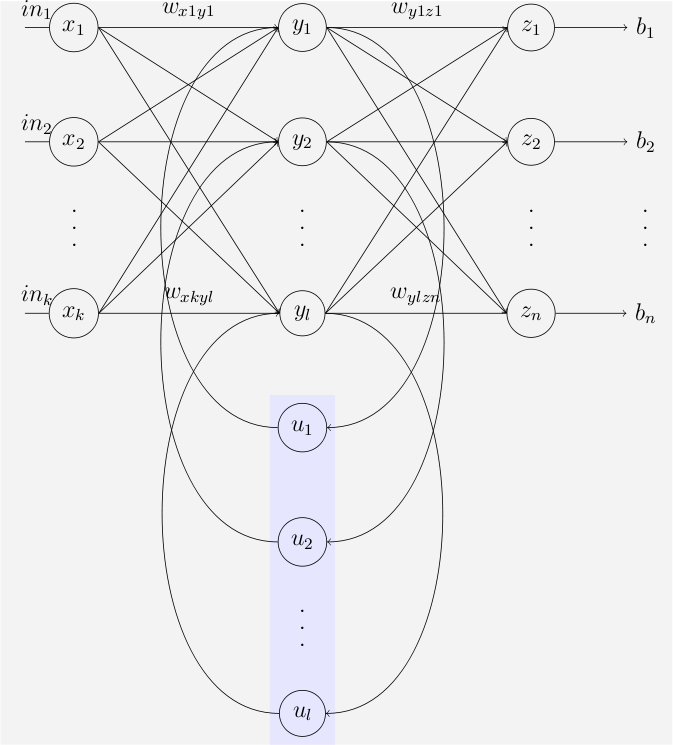
* 和全连接网络的区别
* 更细致到向量级的连接图
* 为什么循环神经网络可以往前看任意多个输入值

**首先看它和全连接网络的区别：**

下图是一个全连接网络：  
它的隐藏层的值只取决于输入的 x

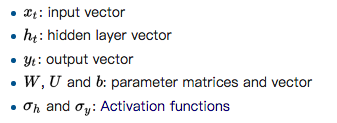


而 RNN 的隐藏层的值 s 不仅仅取决于当前这次的输入 x，还取决于上一次隐藏层的值 s：  
这个过程画成简图是这个样子：

其中，t 是时刻， x 是输入层， s 是隐藏层， o 是输出层，矩阵 W 就是隐藏层上一次的值作为这一次的输入的权重。**上面的简图还不能够说细节，来看一下更细致到向量****级的连接图：**

Elman network

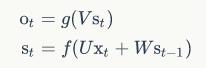
其中各变量含义：



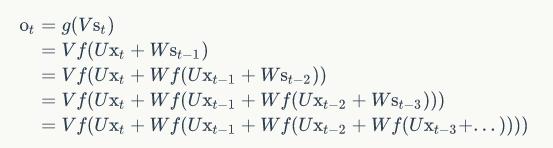
输出层是一个全连接层，它的每个节点都和隐藏层的每个节点相连，  
隐藏层是循环层。

**为什么循环神经网络可以往前看任意多个输入值呢？**

来看下面的公式，即 RNN 的输出层 o 和 隐藏层 s 的计算方法：



如果反复把式 2 带入到式 1，将得到：



这就是原因。

**2. 训练算法**

**RNN 的训练算法为：BPTT**

BPTT 的基本原理和 BP 算法是一样的，同样是三步：

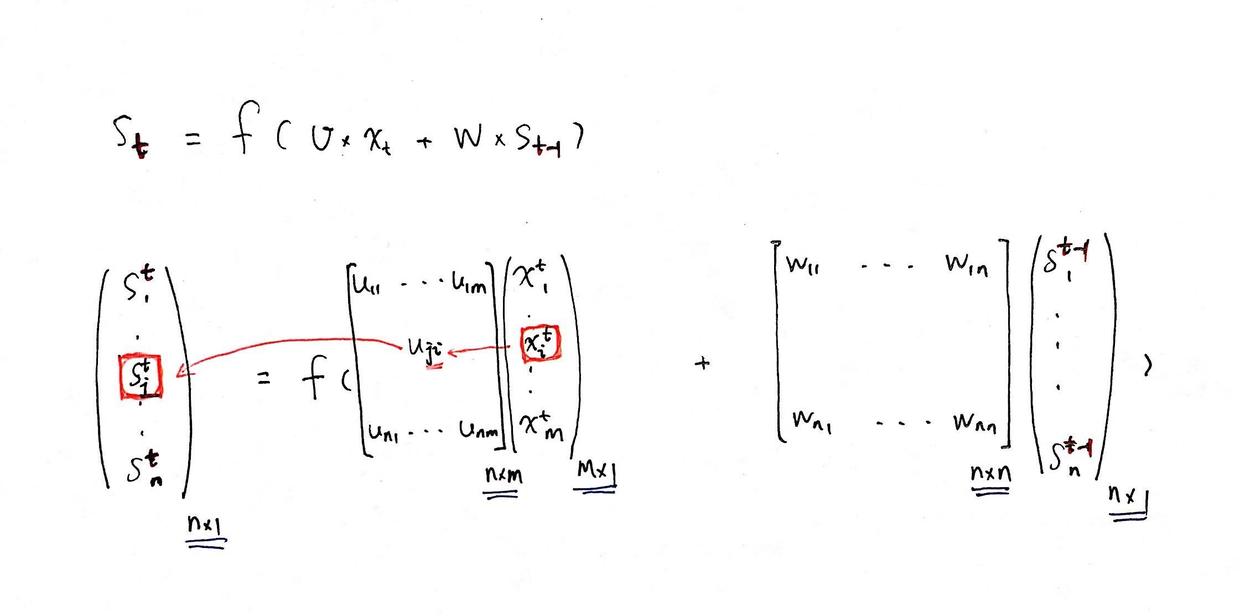
* 1. 前向计算每个神经元的输出值；
  2. 反向计算每个神经元的误差项值，它是误差函数E对神经元j的加权输入的偏导数；
  3. 计算每个权重的梯度。

最后再用随机梯度下降算法更新权重。

**下面详细解析各步骤：**

**1. 前向计算**

计算隐藏层 S 以及它的矩阵形式：  
注意下图中，各变量的维度，标在右下角了，  
s 的上标代表时刻，下标代表这个向量的第几个元素。

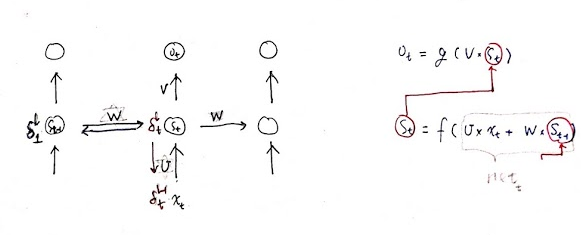


**2. 误差项的计算**

**BTPP 算法就是将第 l 层 t 时刻的误差值沿两个方向传播：**

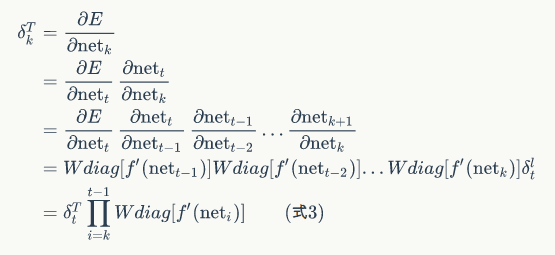
* 一个方向是，传递到上一层网络，这部分只和权重矩阵 U 有关；（就相当于把全连接网络旋转90度来看）
* 另一个是方向是，沿时间线传递到初始时刻，这部分只和权重矩阵 W 有关。

如下图所示：

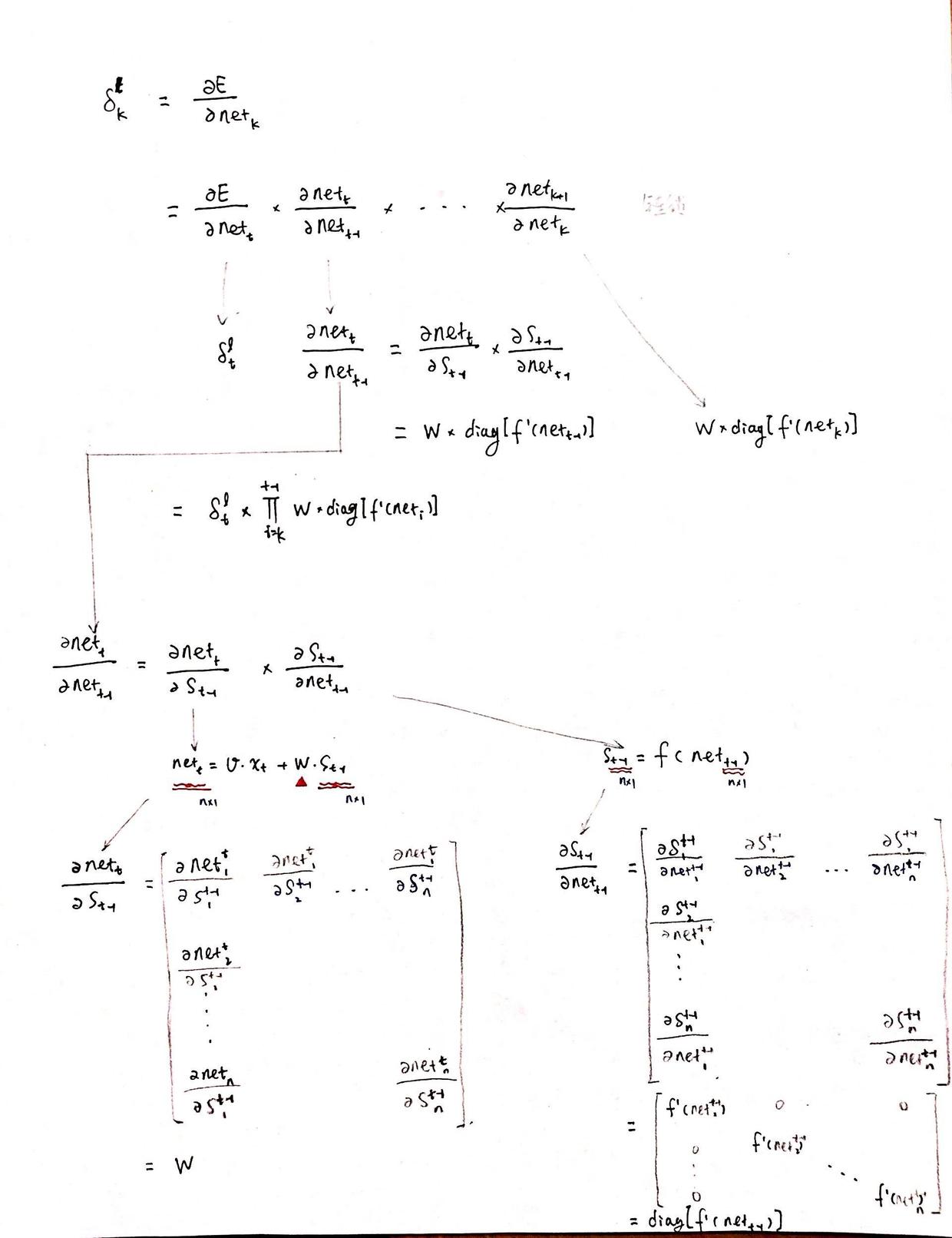


**所以，就是要求这两个方向的误差项的公式：**

**学习资料中式 3 就是将误差项沿时间反向传播的算法，求到了任意时刻k的误差项**

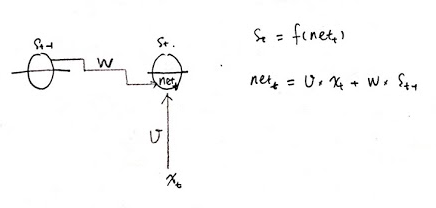


下面是具体的推导过程：  
主要就是用了 链锁反应 和 Jacobian 矩阵

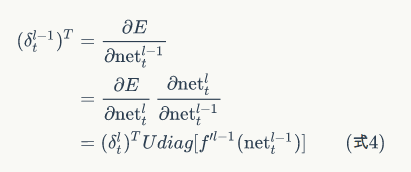


2

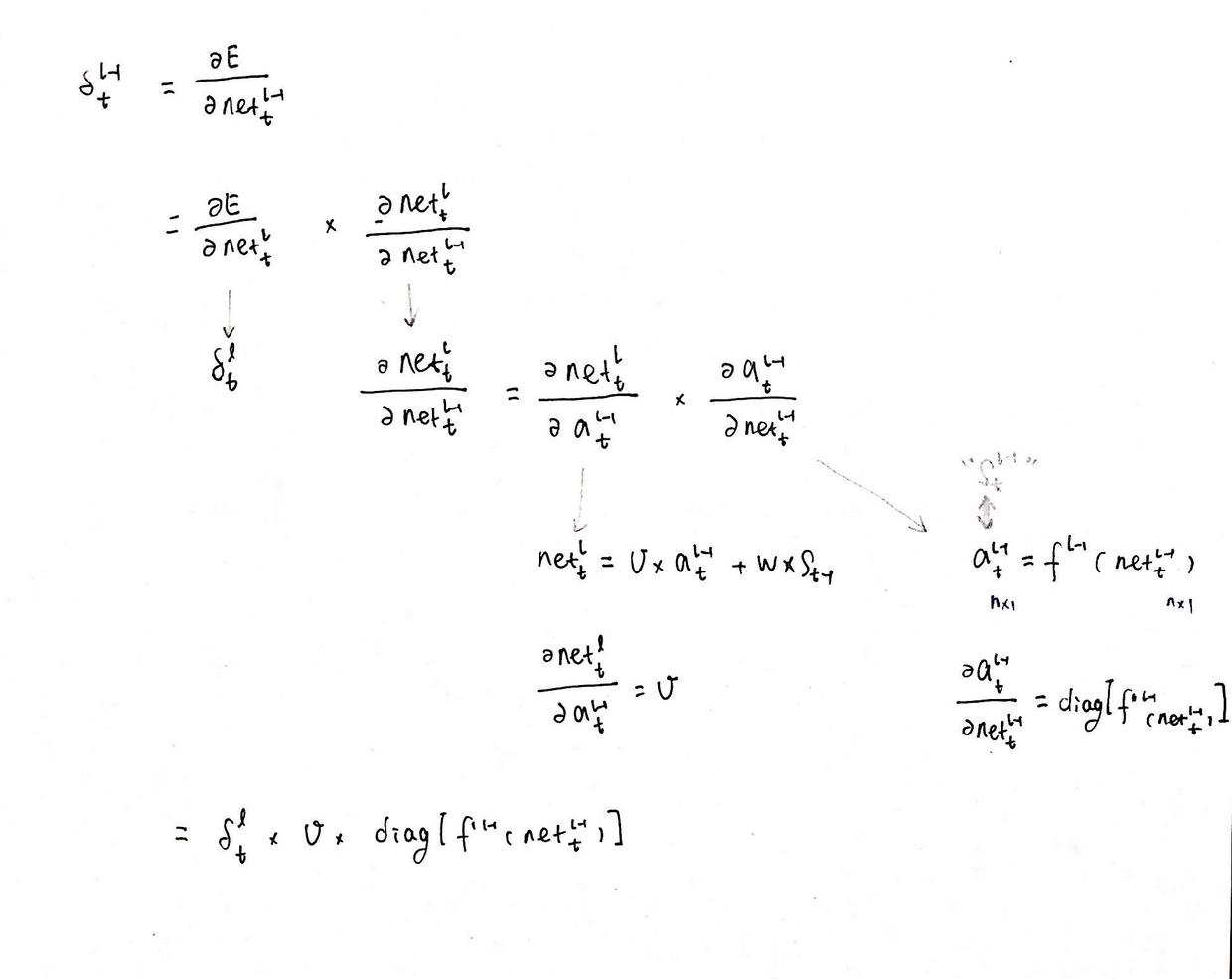
其中 s 和 net 的关系如下，有助于理解求导公式：



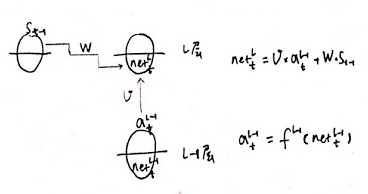
**式 4 就是将误差项传递到上一层算法：**



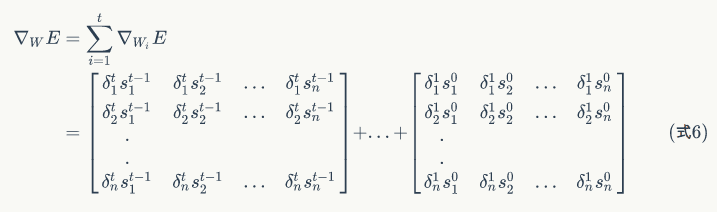
这一步和普通的全连接层的算法是完全一样的，具体的推导过程如下：

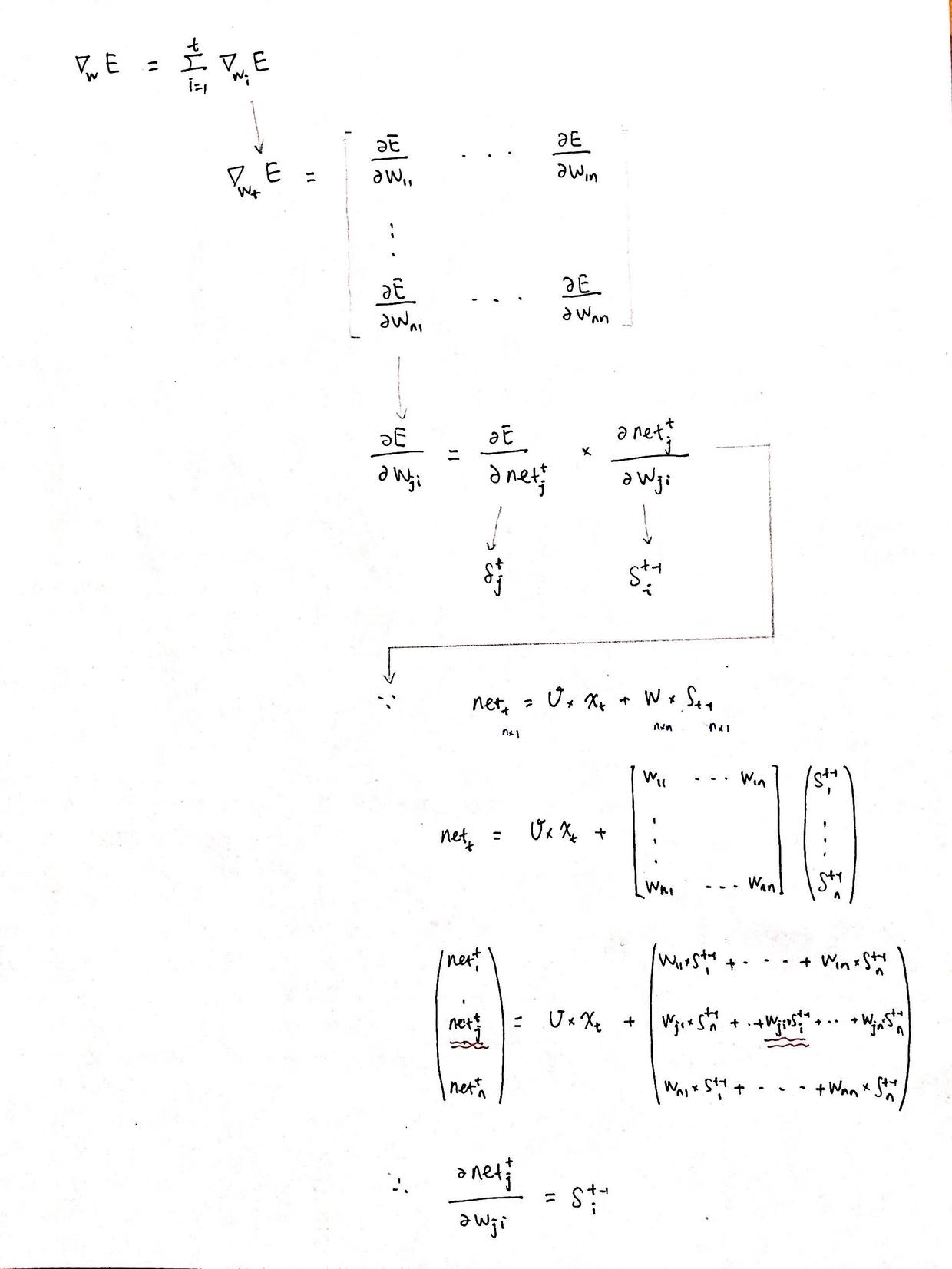


其中 net 的 l 层 和 l－1 层的关系如下：

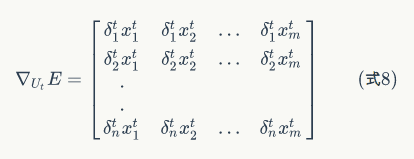


BPTT 算法的最后一步：计算每个权重的梯度  
**学习资料中式 6 就是计算循环层权重矩阵 W 的梯度的公式：**



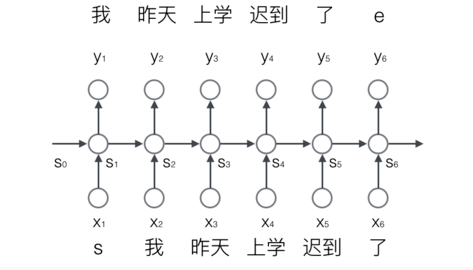
具体的推导过程如下：

**和权重矩阵 W 的梯度计算方式一样，可以得到误差函数在 t 时刻对权重矩阵 U 的梯度：**



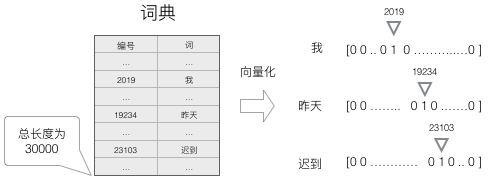
**3. 基于 RNN 的语言模型例子**

我们要用 RNN 做这样一件事情，每输入一个词，循环神经网络就输出截止到目前为止，下一个最可能的词，如下图所示：



**首先，要把词表达为向量的形式：**

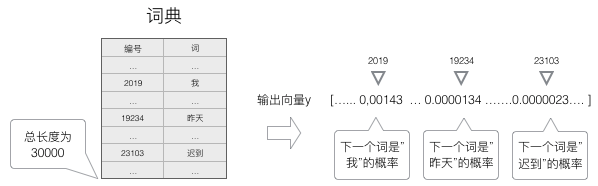
* 建立一个包含所有词的词典，每个词在词典里面有一个唯一的编号。
* 任意一个词都可以用一个N维的one-hot向量来表示。



这种向量化方法，我们就得到了一个高维、稀疏的向量，这之后需要使用一些降维方法，将高维的稀疏向量转变为低维的稠密向量。

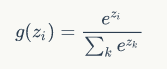
**为了输出 “最可能” 的词，所以需要计算词典中每个词是当前词的下一个词的概率，再选择概率最大的那一个。**

因此，神经网络的输出向量也是一个 N 维向量，向量中的每个元素对应着词典中相应的词是下一个词的概率：

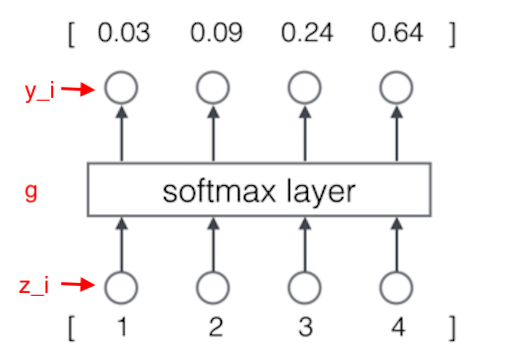


**为了让神经网络输出概率，就要用到 softmax 层作为输出层。**

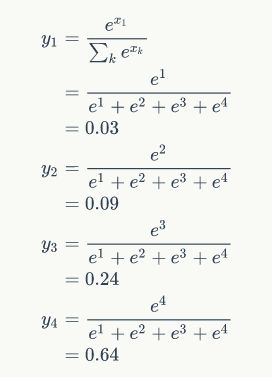
softmax函数的定义：  
因为和概率的特征是一样的，所以可以把它们看做是概率。



例：



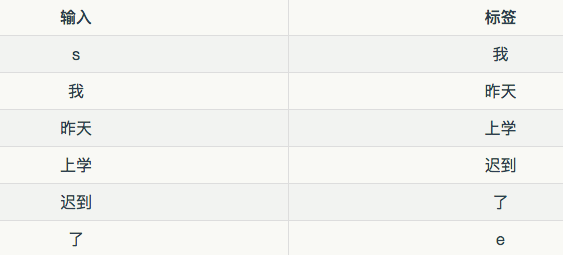
计算过程为：



含义就是：  
模型预测下一个词是词典中第一个词的概率是 0.03，是词典中第二个词的概率是 0.09。

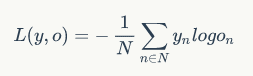
**语言模型如何训练？**

把语料转换成语言模型的训练数据集，即对输入 x 和标签 y 进行向量化，y 也是一个 one-hot 向量

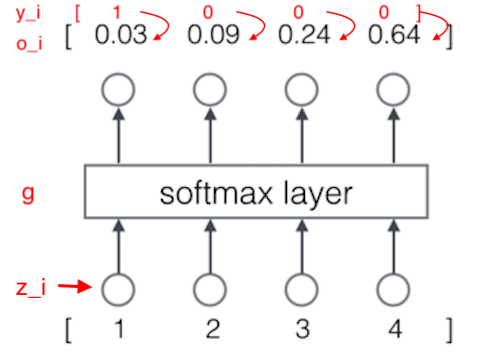


**接下来，对概率进行建模，一般用交叉熵误差函数作为优化目标。**

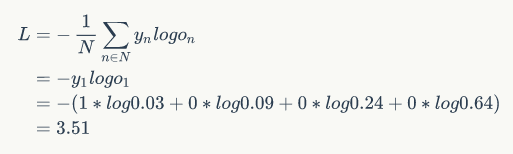
交叉熵误差函数，其定义如下：



用上面例子就是：



计算过程如下：



**有了模型，优化目标，梯度表达式，就可以用梯度下降算法进行训练了。**