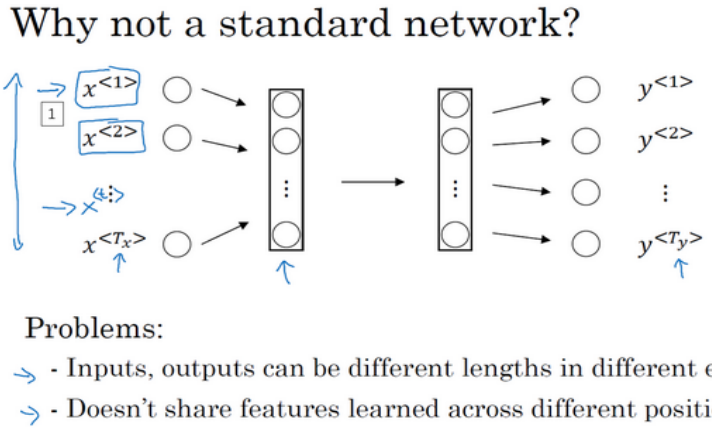
1.3 循环神经网络模型（Recurrent Neural Network Model）

上节视频中，你了解了我们用来定义序列学习问题的符号。现在我们讨论一下怎样才能建立一个模型，建立一个神经网络来学习X到Y的映射。

可以尝试的方法之一是使用标准神经网络，在我们之前的例子中，我们有 9 个输入单词。想象一下，把这 9 个输入单词，可能是 9 个 **one-hot** 向量，然后将它们输入到一个标准神经网络中，经过一些隐藏层，最终会输出 9 个值为 0 或 1 的项，它表明每个输入单词是否是人名的一部分。



但结果表明这个方法并不好，主要有两个问题：

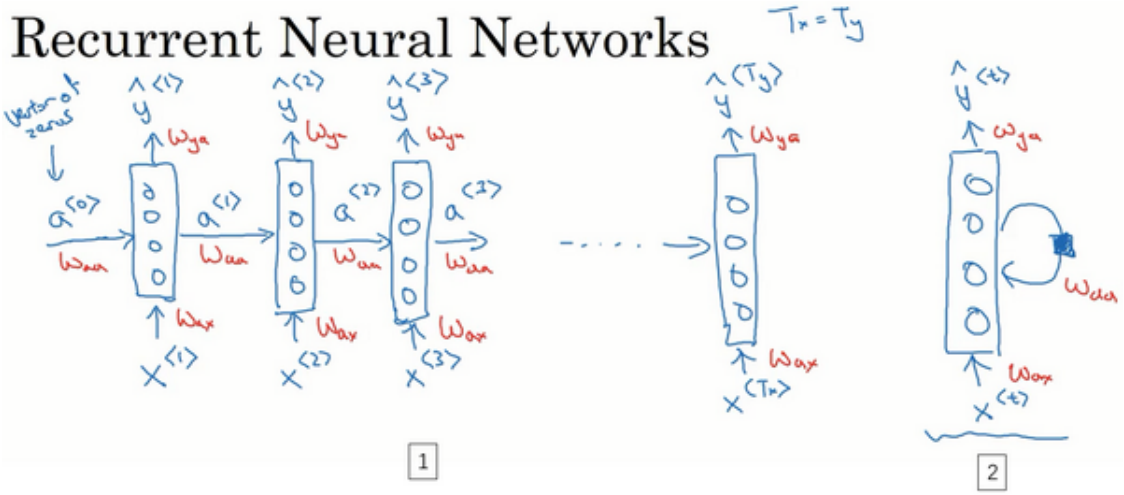
一、是输入和输出数据在不同例子中可以有不同的长度，不是所有的例子都有着同样输入长度T_x或是同样输出长度的T_y。即使每个句子都有最大长度，也许你能够填充（**pad**）或零填充（**zero pad**）使每个输入语句都达到最大长度，但仍然看起来不是一个好的表达方式。

二、一个像这样单纯的神经网络结构，它并不共享从文本的不同位置上学到的特征。具体来说，如果神经网络已经学习到了在位置 1 出现的 **Harry** 可能是人名的一部分，那么如果 **Harry** 出现在其他位置，比如$x^{<t>}$时，它也能够自动识别其为人名的一部分的话，这就很棒了。这可能类似于你在卷积神经网络中看到的，你希望将部分图片里学到的内容快速推广到图片的其他部分，而我们希望对序列数据也有相似的效果。和你在卷积网络中学到的类似，用一个更好的表达方式也能够让你减少模型中参数的数量。

之前我们提到过这些（上图编号 1 所示的$x^{<1>}$.....$x^{<t>}$.....$x^{<T_x>}$）都是 10,000 维的 **one-hot** 向量，因此这会是十分庞大的输入层。如果总的输入大小是最大单词数乘以 10,000，那么第一层的权重矩阵就会有着巨量的参数。但循环神经网络就没有上述的两个问题。

那么什么是循环神经网络呢？我们先建立一个（下图编号 1 所示）。如果你以从左到右

的顺序读这个句子，第一个单词就是，假如说是$x^{<1>}$，我们要做的就是将第一个词输入一个神经网络层，我打算这样画，第一个神经网络的隐藏层，我们可以让神经网络尝试预测输出，判断这是否是人名的一部分。循环神经网络做的是，当它读到句中的第二个单词时，假设是$x^{<2>}$，它不是仅用$x^{<2>}$就预测出$\hat{y}^{<2>}$，他也会输入一些来自时间步 1 的信息。具体而言，**时间步 1 的激活值就会传递到时间步 2。**然后，在下一个时间步，循环神经网络输入了单词$x^{<3>}$，然后它尝试预测输出了预测结果$\hat{y}^{<3>}$，等等，一直到最后一个时间步，输入了$x^{<T_x>}$，然后输出了$\hat{y}^{<T_y>}$。至少在这个例子中$T_x = T_y$，同时如果T_x和T_y不相同，这个结构会需要作出一些改变。**所以在每一个时间步中，循环神经网络传递一个激活值到下一个时间步中用于计算。**

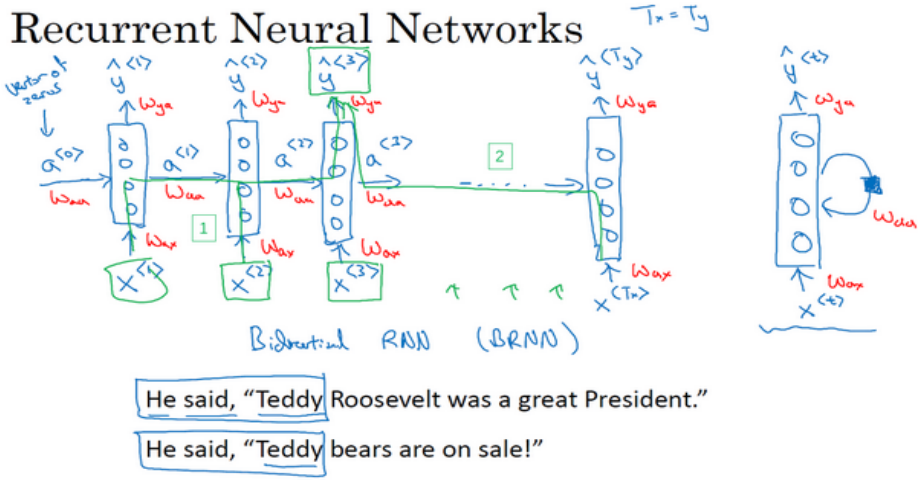


要开始整个流程，**在零时刻需要构造一个激活值$a^{<0>}$**，这通常是零向量。有些研究人员会随机用其他方法初始化$a^{<0>}$，不过使用零向量作为零时刻的伪激活值是最常见的选择，因此我们把它输入神经网络。

在一些研究论文中或是一些书中你会看到这类神经网络，用这样的图形来表示（上图编号 2 所示），在每一个时间步中，你输入$x^{<t>}$然后输出$y^{<t>}$。然后为了表示循环连接有时人们会像这样画个圈，表示输回网络层，有时他们会画一个黑色方块，来表示在这个黑色方块处会延迟一个时间步。我个人认为这些循环图很难理解，所以在本次课程中，我画图更倾向于使用左边这种分布画法（上图编号 1 所示）。不过如果你在教材中或是研究论文中看到了右边这种图表的画法（上图编号 2 所示），它可以在心中将这图展开成左图那样。

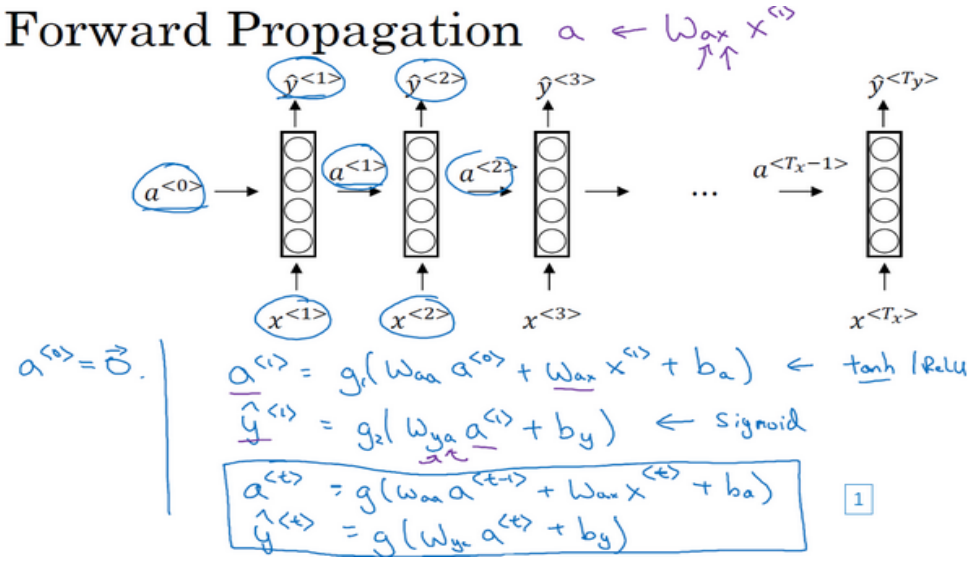
循环神经网络是从左向右扫描数据，同时每个时间步的参数也是共享的，所以下页幻灯片中我们会详细讲述它的一套参数，我们用**W_{ax}来表示管理着从$x^{<1>}$到隐藏层的连接的一系**

列参数，每个时间步使用的都是相同的参数W_{ax}。而激活值也就是水平联系是由参数W_{aa}决定的，同时每一个时间步都使用相同的参数W_{aa}，同样的输出结果由W_{ya}决定。下图详细讲述这些参数是如何起作用。



在这个循环神经网络中，它的意思是在预测$\hat{y}^{<3>}$时，不仅要使用$x^{<3>}$的信息，还要使用来自$x^{<1>}$和$x^{<2>}$的信息，因为来自$x^{<1>}$的信息可以通过这样的路径（上图编号 1 所示的路径）来帮助预测$\hat{y}^{<3>}$。这个循环神经网络的一个缺点就是它只使用了这个序列中之前的信息来做出预测，尤其当预测$\hat{y}^{<3>}$时，它没有用到$x^{<4>}$，$x^{<5>}$，$x^{<6>}$等等的信息。所以这就有一个问题，因为如果给定了这个句子，**"Teddy Roosevelt was a great President."**，为了判断 **Teddy** 是否是人名的一部分，仅仅知道句中前两个词是完全不够的，还需要知道句中后部分的信息，这也是十分有用的，因为句子也可能是这样的，**"Teddy bears are on sale!"**。因此如果只给定前三个单词，是不可能确切地知道 **Teddy** 是否是人名的一部分，第一个例子是人名，第二个例子就不是，所以你不可能只看前三个单词就能分辨出其中的区别。

所以这样特定的神经网络结构的一个限制是它在某一时刻的预测仅使用了从序列之前的输入信息并没有使用序列中后部分的信息，我们会在之后的双向循环神经网络（**BRNN**）的视频中处理这个问题。但对于现在，这个更简单的单向神经网络结构就够我们来解释关键概念了，之后只要在此基础上作出修改就能同时使用序列中前面和后面的信息来预测$\hat{y}^{<3>}$，不过我们会在之后的视频讲述这些内容，接下来我们具体地写出这个神经网络计算了些什么。



这里是一张清理后的神经网络示意图，和我之前提及的一样，一般开始先输入$a^{<0>}$，它是一个零向量。接着就是前向传播过程，先计算激活值$a^{<1>}$，然后再计算$y^{<1>}$。

$$a^{<1>} = g_1(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a)$$

$$\hat{y}^{<1>} = g_2(W_{ya}a^{<1>} + b_y)$$

我将用这样的符号约定来表示这些矩阵下标，举个例子W_{ax}，第二个下标意味着W_{ax}要乘以某个x类型的量，然后第一个下标a表示它是用来计算某个a类型的变量。同样的，可以看出这里的W_{ya}乘上了某个a类型的量，用来计算出某个$\hat{y}$类型的量。

循环神经网络用的激活函数经常是 tanh，不过有时候也会用 ReLU，但是 tanh 是更通常的选择，我们有其他方法来避免梯度消失问题，我们将在之后进行讲述。**选用哪个激活函数是取决于你的输出y，如果它是一个二分问题，那么我猜你会用 sigmoid 函数作为激活函数，如果是k类别分类问题的话，那么可以选用 softmax 作为激活函数。**不过这里激活函数的类型取决于你有什么样类型的输出y，对于命名实体识别来说y只可能是 0 或者 1，那我猜这里第二个激活函数g可以是 **sigmoid** 激活函数。

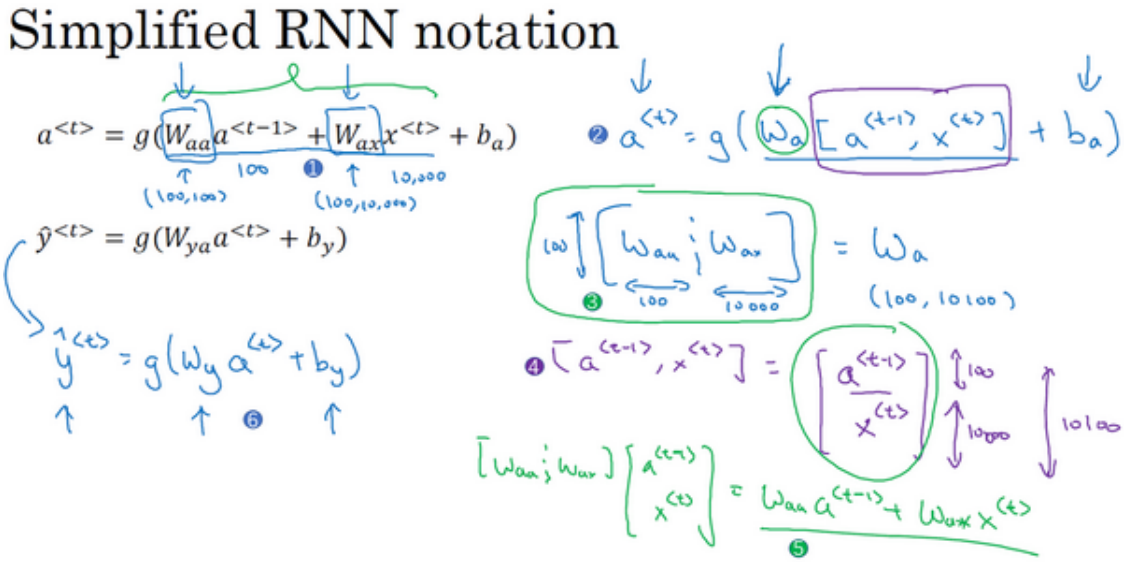
更一般的情况下，在t时刻，

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

所以这些等式定义了神经网络的前向传播，你可以从零向量$a^{<0>}$开始，然后用$a^{<0>}$和$x^{<1>}$来计算出$a^{<1>}$和$\hat{y}^{<1>}$，然后用$x^{<2>}$和$a^{<1>}$一起算出$a^{<2>}$和$\hat{y}^{<2>}$等等，像图中这样，从左到右完成前向传播。

现在为了帮我们建立更复杂的神经网络，我实际要将这个符号简化一下，我在下一张幻灯片里复制了这两个等式（上图编号 1 所示的两个等式）。

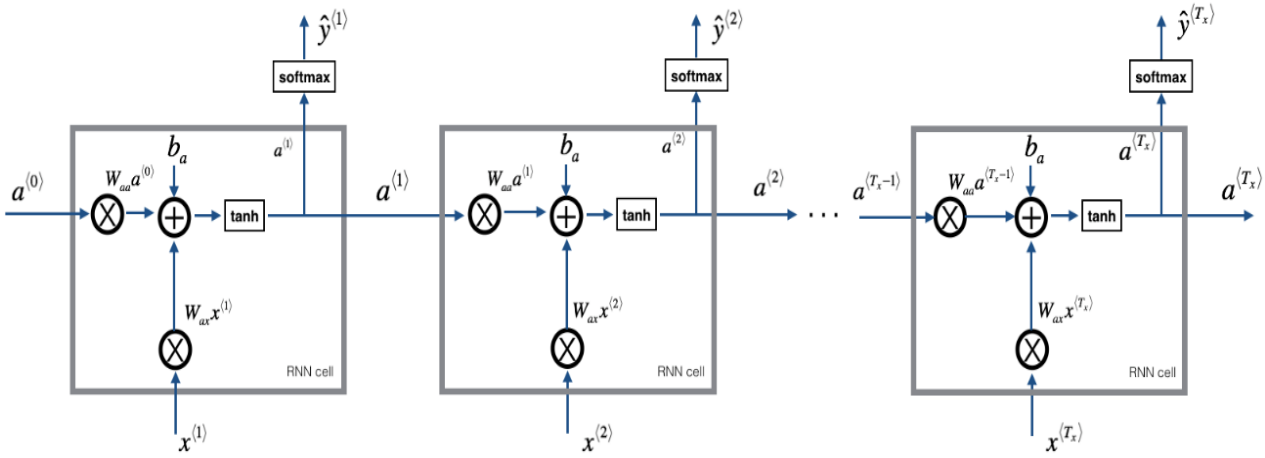


接下来为了简化这些符号，我要将这部分（$W_{aa}a^{<t-1>} + W_{ax}x^{<t>}$）（上图编号 1 所示）以更简单的形式写出来，我把它写做$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a)$（上图编号 2 所示），那么左右两边划线部分应该是等价的。所以我们定义W_a的方式是将矩阵W_{aa}和矩阵W_{ax}水平并列放置，$[W_{aa} : W_{ax}] = W_a$（上图编号 3 所示）。举个例子，如果a是 100 维的，然后延续之前的例子，x是 10,000 维的，那么W_{aa}就是个（100，100）维的矩阵，W_{ax}就是个（100，10,000）维的矩阵，因此如果将这两个矩阵堆起来，W_a就会是个（100，10,100）维的矩阵。

用这个符号（$[a^{<t-1>}, x^{<t>}]$）的意思是将这两个向量堆在一起，我会用这个符号表示，即$\begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix}$（上图编号 4 所示），最终这就是个 10,100 维的向量。你可以自己检查一下，用这个矩阵乘以这个向量，刚好能够得到原来的量，因为此时，矩阵$[W_{aa} : W_{ax}]$乘以$\begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix}$，刚好等于$W_{aa}a^{<t-1>} + W_{ax}x^{<t>}$，刚好等于之前的这个结论（上图编号 5 所示）。这种记法的好处是我们可以不使用两个参数矩阵W_{aa}和W_{ax}，而是将其压缩成一个参数矩阵W_a，所以当我们建立更复杂模型时这就能够简化我们要用到的符号。

同样对于这个例子（$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$），我会用更简单的方式重写，$\hat{y}^{<t>} = g(W_y a^{<t>} + b_y)$（上图编号 6 所示）。现在$W_y$和$b_y$符号仅有一个下标，它表示在计算时会输出什么类型的量，所以W_y就表明它是计算y类型的量的权重矩阵，而上面的W_a和b_a则表示这些参数是用来计算a类型或者说是激活值的。

RNN 前向传播示意图：



你现在知道了基本的循环神经网络，下节课我们会一起来讨论反向传播，以及你如何能够用 **RNN** 进行学习。