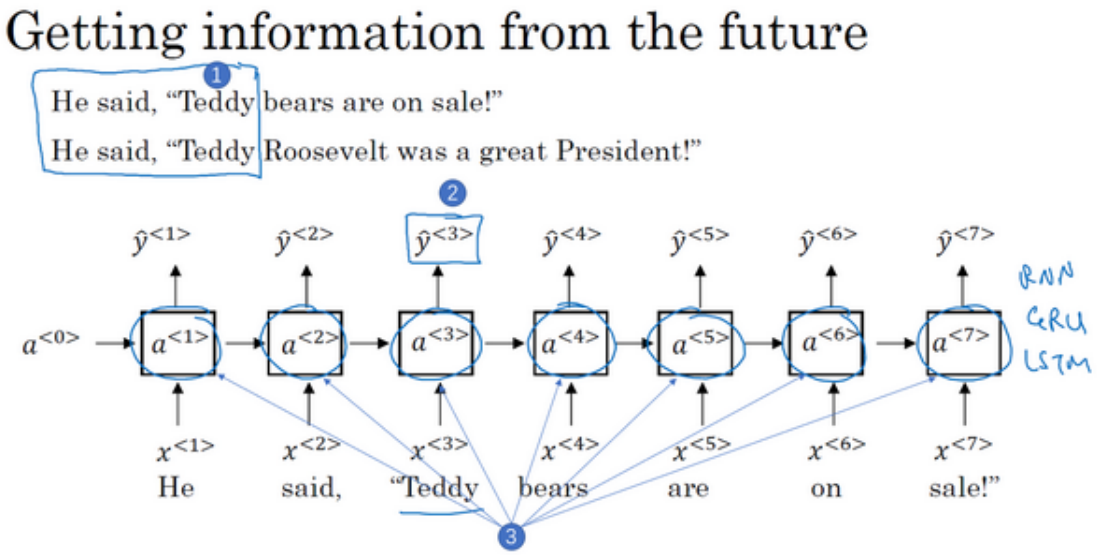
1.11 双向循环神经网络（Bidirectional RNN）

现在，你已经了解了大部分 **RNN** 模型的关键的构件，还有两个方法可以让你构建更好的模型，其中之一就是双向 **RNN** 模型，这个模型可以让你在序列的某点处不仅可以获取之前的信息，还可以获取未来的信息，我们会在这个视频里讲解。第二个就是深层的 **RNN**，我们会在下个视频里见到，现在先从双向 **RNN** 开始吧。

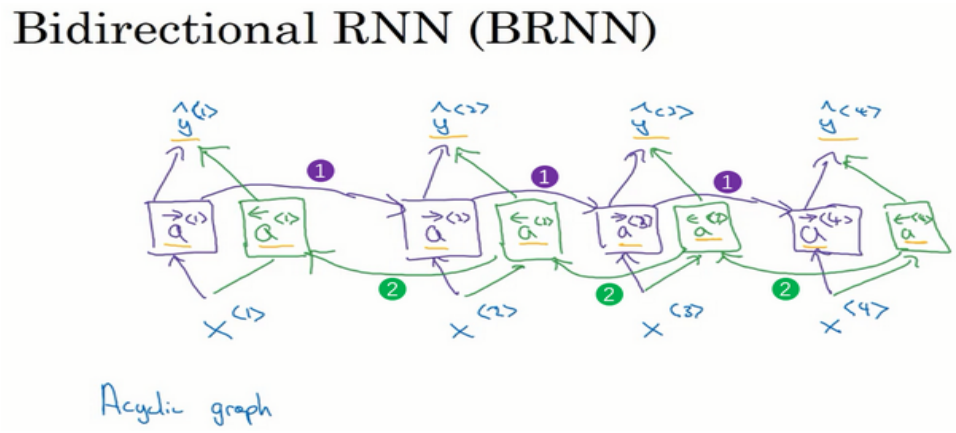


为了了解双向 **RNN** 的动机，我们先看一下之前在命名实体识别中已经见过多次的神经网络。这个网络有一个问题，在判断第三个词 **Teddy**（上图编号 1 所示）是不是人名的一部分时，光看句子前面部分是不够的，为了判断$\hat{y}^{<3>}$（上图编号 2 所示）是 0 还是 1，除了前 3 个单词，你还需要更多的信息，因为根据前 3 个单词无法判断他们说的是 **Teddy 熊**，还是前美国总统 **Teddy Roosevelt**，所以这是一个非双向的或者说只有前向的 **RNN**。我刚才所说的总是成立的，不管这些单元（上图编号 3 所示）是标准的 **RNN** 块，还是 **GRU** 单元或者是 **LSTM** 单元，只要这些构件都是只有前向的。

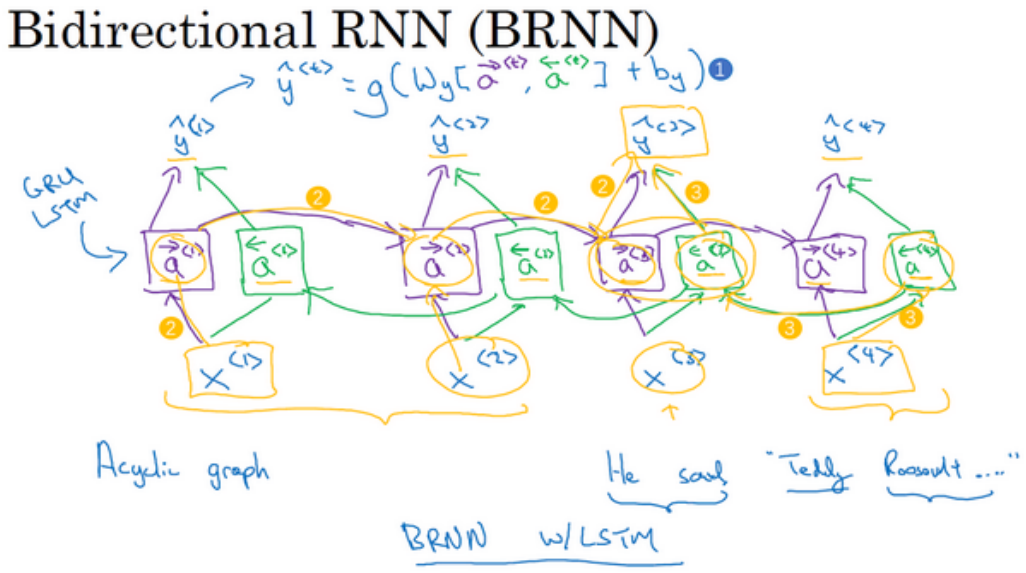
那么一个双向的 **RNN** 是如何解决这个问题的？下面解释双向 **RNN** 的工作原理。为了简单，我们用四个输入或者说一个只有 4 个单词的句子，这样输入只有 4 个，$x^{<1>}$到$x^{<4>}$。从这里开始的这个网络会有一个前向的循环单元叫做$\vec{a}^{<1>}$，$\vec{a}^{<2>}$，$\vec{a}^{<3>}$还有$\vec{a}^{<4>}$，我在这上面加个向右的箭头来表示前向的循环单元，并且他们这样连接（下图编号 1 所示）。这四个循环单元都有一个当前输入x输入进去，得到预测的$\hat{y}^{<1>}$，$\hat{y}^{<2>}$，$\hat{y}^{<3>}$和$\hat{y}^{<4>}$。

到目前为止，我还没做什么，仅仅是把前面幻灯片里的 **RNN** 画在了这里，只是在这些

地方画上了箭头。我之所以在这些地方画上了箭头是因为我们想要增加一个反向循环层，这里有个$\overleftarrow{a}^{<1>}$，左箭头代表反向连接，$\overleftarrow{a}^{<2>}$反向连接，$\overleftarrow{a}^{<3>}$反向连接，$\overleftarrow{a}^{<4>}$反向连接，所以这里的左箭头代表反向连接。



同样，我们把网络这样向上连接，这个a反向连接就依次反向向前连接（上图编号 2 所示）。这样，这个网络就构成了一个无环图。给定一个输入序列$x^{<1>}$到$x^{<4>}$，这个序列首先计算前向的$\vec{a}^{<1>}$，然后计算前向的$\vec{a}^{<2>}$，接着$\vec{a}^{<3>}$，$\vec{a}^{<4>}$。而反向序列从计算$\overleftarrow{a}^{<4>}$开始，反向进行，计算反向的$\overleftarrow{a}^{<3>}$。你计算的是网络激活值，这不是反向而是前向的传播，而图中这个前向传播一部分计算是从左到右，一部分计算是从右到左。计算完了反向的$\overleftarrow{a}^{<3>}$，可以用这些激活值计算反向的$\overleftarrow{a}^{<2>}$，然后是反向的$\overleftarrow{a}^{<1>}$，把所有这些激活值都计算完了就可以计算预测结果了。



举个例子，为了预测结果，你的网络会有如$\hat{y}^{<t>}$，$\hat{y}^{<t>} = g(W_g[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$（上图编号 1 所示）。比如你要观察时间 3 这里的预测结果，信息从$x^{<1>}$过来，流经这里，前向

的$\vec{a}^{<1>}$到前向的$\vec{a}^{<2>}$，这些函数里都有表达，到前向的$\vec{a}^{<3>}$再到$\hat{y}^{<3>}$（上图编号 2 所示的路径），所以从$x^{<1>}$，$x^{<2>}$，$x^{<3>}$来的信息都会考虑在内，而从$x^{<4>}$来的信息会流过反向的$\overleftarrow{a}^{<4>}$，到反向的$\overleftarrow{a}^{<3>}$再到$\hat{y}^{<3>}$（上图编号 3 所示的路径）。这样使得时间 3 的预测结果不仅输入了过去的信息，还有现在的信息，这一步涉及了前向和反向的传播信息以及未来的信息。给定一个句子**"He said Teddy Roosevelt..."**来预测 **Teddy** 是不是人名的一部分，**你需要同时考虑过去和未来的信息。**

这就是双向循环神经网络，并且这些基本单元不仅仅是标准 **RNN** 单元，也可以是 **GRU** 单元或者 **LSTM** 单元。事实上，很多的 **NLP** 问题，对于大量有自然语言处理问题的文本，有 **LSTM** 单元的双向 **RNN** 模型是用的最多的。所以如果有 **NLP** 问题，并且文本句子都是完整的，首先需要标定这些句子，一个有 **LSTM** 单元的双向 **RNN** 模型，有前向和反向过程是一个不错的首选。

以上就是双向 **RNN** 的内容，这个改进的方法不仅能用于基本的 **RNN** 结构，也能用于 **GRU** 和 **LSTM**。通过这些改变，你就可以用一个用 **RNN** 或 **GRU** 或 **LSTM** 构建的模型，并且能够预测任意位置，即使在句子的中间，因为模型能够考虑整个句子的信息。这个双向 **RNN** 网络模型的缺点就是你需要完整的数据的序列，你才能预测任意位置。比如说你要构建一个语音识别系统，那么双向 **RNN** 模型需要你考虑整个语音表达，但是如果直接用这个去实现的话，你需要等待这个人说完，然后获取整个语音表达才能处理这段语音，并进一步做语音识别。对于实际的语音识别的应用通常会有更加复杂的模块，而不是仅仅用我们见过的标准的双向 **RNN** 模型。但是对于很多自然语言处理的应用，如果你总是可以获取整个句子，这个标准的双向 **RNN** 算法实际上很高效。

这就是双向 **RNN**，下一个视频，也是这周的最后一个，我们会讨论如何用这些概念，标准的 **RNN**，**LSTM** 单元，**GRU** 单元，还有双向的版本，构建更深的网络。