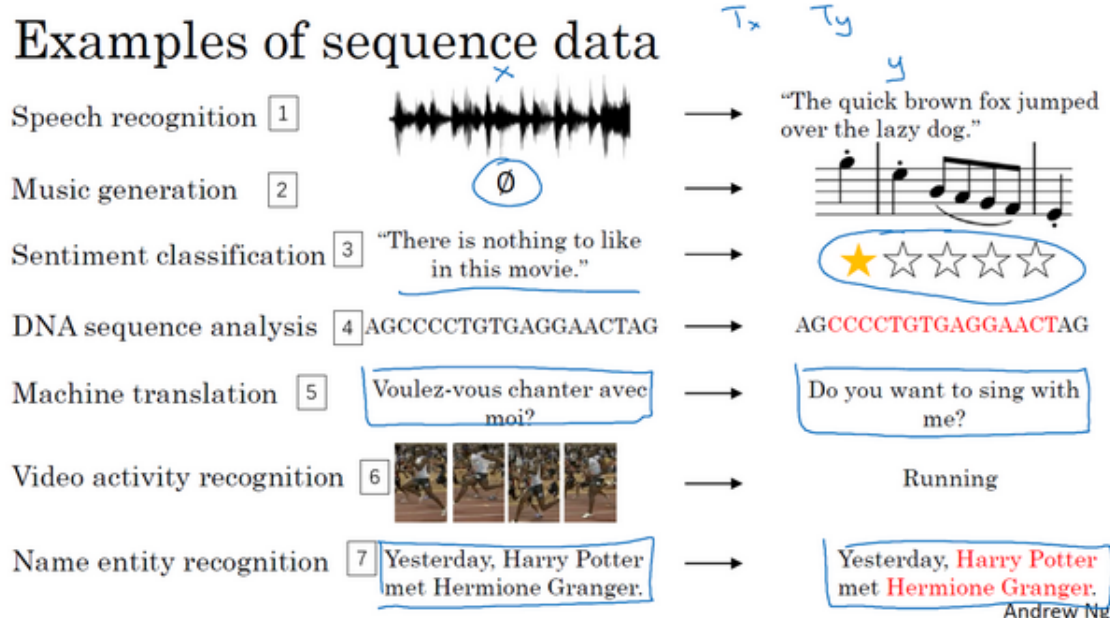


1.5 不同类型的循环神经网络 (Different types of RNNs)

现在你已经了解了一种 **RNN** 结构，它的输入量 T_x 等于输出数量 T_y 。事实上，对于其他一些应用， T_x 和 T_y 并不一定相等。在这个视频里，你会看到更多的 **RNN** 的结构。



你应该还记得这周第一个视频中的那个幻灯片，那里有很多例子输入 x 和输出 y ，有各种类型，并不是所有的情况都满足 $T_x = T_y$ 。

比如音乐生成这个例子， T_x 可以是长度为 1 甚至为空集。再比如电影情感分类，输出 y 可以是 1 到 5 的整数，而输入是一个序列。在命名实体识别中，这个例子中输入长度和输出长度是一样的。

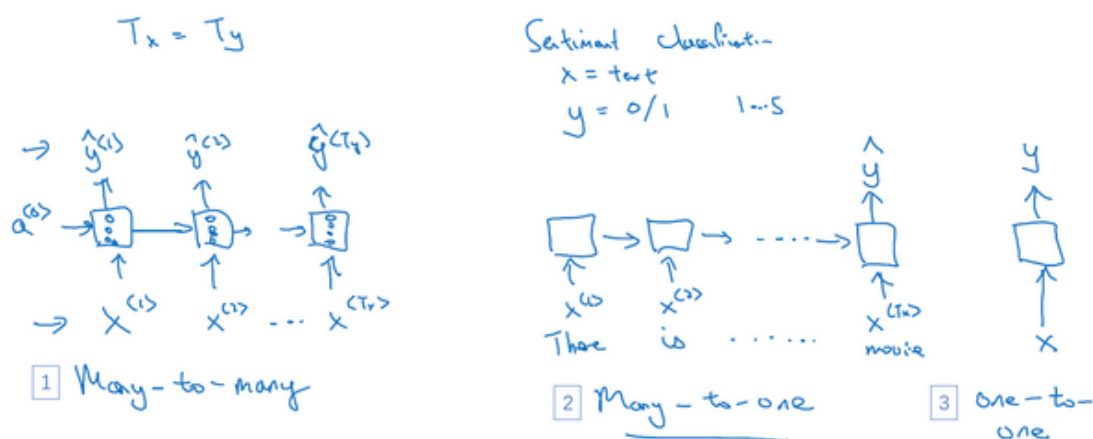
还有一些情况，输入长度和输出长度不同，他们都是序列但长度不同，比如机器翻译，一个法语句子和一个英语句子不同数量的单词却能表达同一个意思。

所以我们应该修改基本的 **RNN** 结构来处理这些问题，这个视频的内容参考了 **Andrej Karpathy** 的博客，一篇叫做《循环神经网络的非理性效果》（“The Unreasonable Effectiveness of Recurrent Neural Networks”）的文章，我们看一些例子。

你已经见过 $T_x = T_y$ 的例子了（下图编号 1 所示），也就是我们输入序列 $x^{<1>}$ ， $x^{<2>}$ ，一直到 $x^{<T_x>}$ ，我们的循环神经网络这样工作，输入 $x^{<1>}$ 来计算 $\hat{y}^{<1>}$ ， $\hat{y}^{<2>}$ 等等一直到 $\hat{y}^{<T_y>}$ 。在原先的图里，我会画一串圆圈表示神经元，大部分时候为了让符号更加简单，此处就以简单的小圈表示。这个就叫做“多对多”（many-to-many）的结构，因为输入序列有很多的输入，而输出序列也有很多输出。

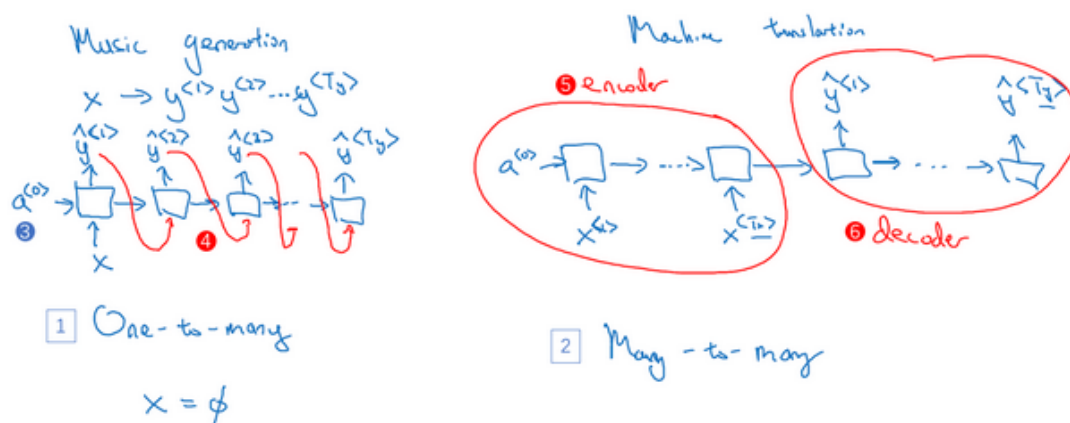
现在我们看另外一个例子，假如说，你想处理情感分类问题（下图编号 2 所示），这里 x 可能是一段文本，比如一个电影的评论，“**These is nothing to like in this movie.**”（“这部电影没什么还看的。”），所以 x 就是一个序列，而 y 可能是从 1 到 5 的一个数字，或者是 0 或 1，这代表正面评价和负面评价，而数字 1 到 5 代表电影是 1 星，2 星，3 星，4 星还是 5 星。所以在这个例子中，我们可以简化神经网络的结构，输入 $x^{<1>}$, $x^{<2>}$ ，一次输入一个单词，如果输入文本是“**These is nothing to like in this movie**”，那么单词的对应如下图编号 2 所示。我们不再在每个时间上都有输出了，而是让这个 RNN 网络读入整个句子，然后在最后一个时间上得到输出，这样输入的就是整个句子，所以这个神经网络叫做“多对一”（**many-to-one**）结构，因为它有很多输入，很多的单词，然后输出一个数字。

Examples of RNN architectures



为了完整性，还要补充一个“一对一”（**one-to-one**）的结构（上图编号 3 所示），这个可能没有那么重要，这就是一个小型的标准的神经网络，输入 x 然后得到输出 y ，我们这个系列课程的前两个课程已经讨论过这种类型的神经网络了。

Examples of RNN architectures

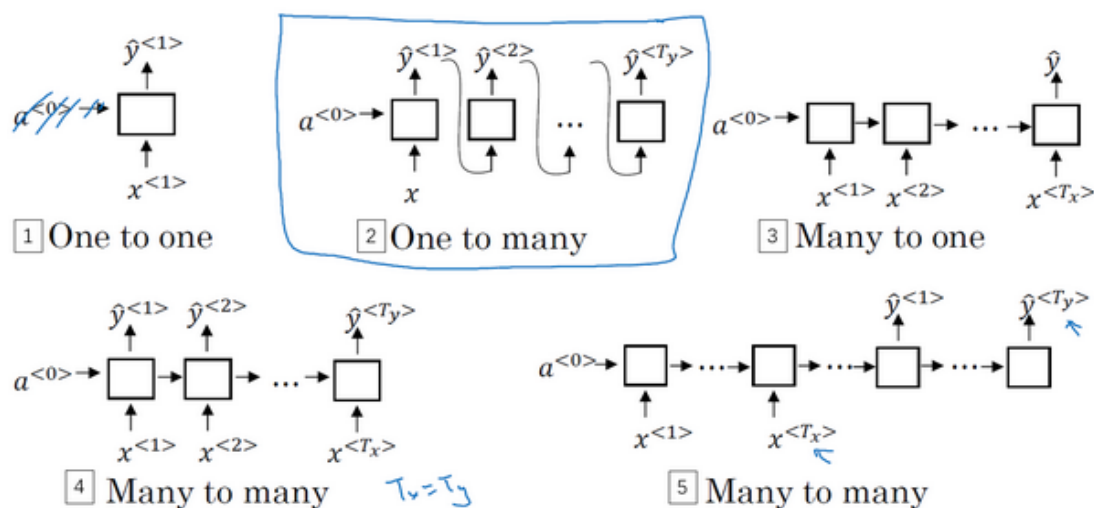


除了“多对一”的结构，也可以有“一对多”（**one-to-many**）的结构。对于一个“一对多”神经网络结构的例子就是音乐生成（上图编号 1 所示），事实上，你会在这个课后编程练习中去实现这样的模型，你的目标是使用一个神经网络输出一些音符。对应于一段音乐，输入 x 可以是一个整数，表示你想要的音乐类型或者是你想要的音乐的第一个音符，并且如果你什么都不想输入， x 可以是空的输入，可设为 0 向量。

这样这个神经网络的结构，首先是你的输入 x ，然后得到 **RNN** 的输出，第一个值，然后就没有输入了，再得到第二个输出，接着输出第三个值等等，一直到合成这个音乐作品的最后一个音符，这里也可以写上输入 $a^{<0>}$ （上图编号 3 所示）。有一个后面才会讲到的技术细节，当你生成序列时通常会把第一个合成的输出也喂给下一层（上图编号 4 所示），所以实际的网络结构最终就像这个样子。

我们已经讨论了“多对多”、“多对一”、“一对一”和“一对多”的结构，对于“多对多”的结构还有一个有趣的例子值得详细说一下，就是输入和输出长度不同的情况。你刚才看过的多对多的例子，它的输入长度和输出长度是完全一样的。而对于像机器翻译这样的应用，输入句子的单词的数量，比如说一个法语的句子，和输出句子的单词数量，比如翻译成英语，这两个句子的长度可能不同，所以还需要一个新的网络结构，一个不同的神经网络（上图编号 2 所示）。首先读入这个句子，读入这个输入，比如你要将法语翻译成英语，读完之后，这个网络就会输出翻译结果。有了这种结构 T_x 和 T_y 就可以是不同的长度了。同样，你也可以画上这个 $a^{<0>}$ 。这个网络的结构有两个不同的部分，这（上图编号 5 所示）是一个编码器，获取输入，比如法语句子，这（上图编号 6 所示）是解码器，它会读取整个句子，然后输出翻译成其他语言的结果。

Summary of RNN types



这就是一个“**多对多**”结构的例子，到这周结束的时候，你就能对这些各种各样结构的基本构件有一个很好的理解。严格来说，还有一种结构，我们会在第四周涉及到，就是“注意力”（**attention based**）结构，但是根据我们现在画的这些图不好理解这个模型。

总结一下这些各种各样的 **RNN** 结构，这（上图编号 1 所示）是“**一对一**”的结构，当去掉 $a^{<0>}$ 时它就是一种标准类型的神经网络。还有一种“**一对多**”的结构（上图编号 2 所示），比如音乐生成或者序列生成。还有“**多对一**”，这（上图编号 3 所示）是情感分类的例子，首先读取输入，一个电影评论的文本，然后判断他们是否喜欢电影还是不喜欢。还有“**多对多**”的结构（上图编号 4 所示），命名实体识别就是“**多对多**”的例子，其中 $T_x = T_y$ 。最后还有一种“**多对多**”结构的其他版本（上图编号 5 所示），对于像机器翻译这样的应用， T_x 和 T_y 就可以不同了。

现在，你已经了解了大部分基本的模块，这些就是差不多所有的神经网络了，除了序列生成，有些细节的问题我们会在下节课讲解。

我希望你从本视频中了解到用这些 **RNN** 的基本模块，把它们组合在一起就可以构建各种各样的模型。但是正如我前面提到的，序列生成还有一些不一样的地方，在这周的练习里，你也会实现它，你需要构建一个语言模型，结果好的话会得到一些有趣的序列或者有意思的文本。下节课深入探讨序列生成。