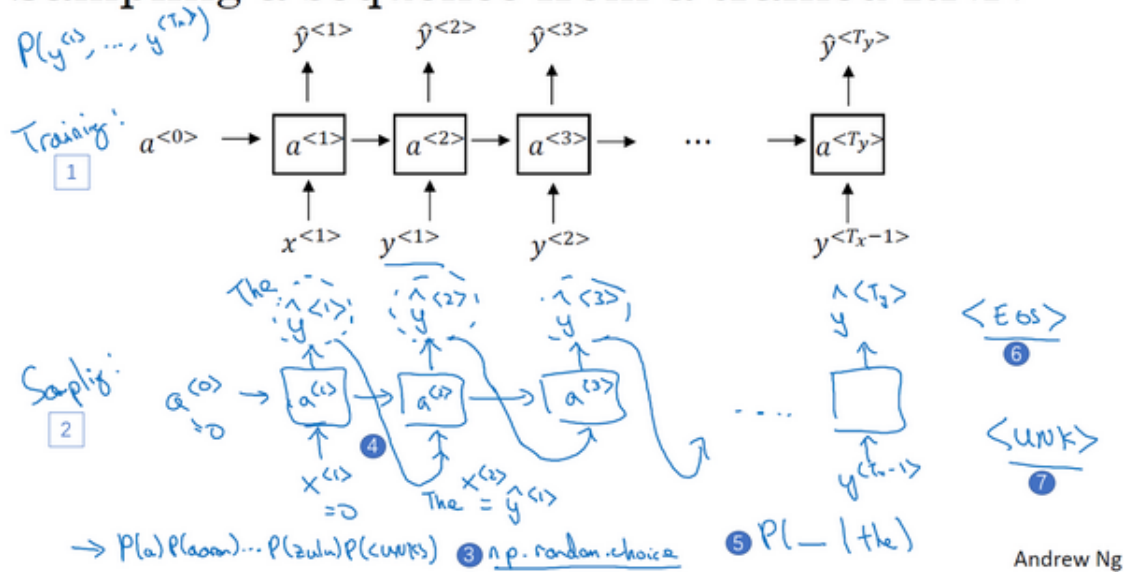


1.7 对新序列采样 (Sampling novel sequences)

在你训练一个序列模型之后，要想了解到这个模型学到了什么，一种非正式的方法就是进行一次新序列采样，来看看到底应该怎么做。

记住一个序列模型模拟了任意特定单词序列的概率，我们要做的就是对这些概率分布进行采样来生成一个新的单词序列。下图编号 1 所示的网络已经被上方所展示的结构训练训练过了，而为了进行采样（下图编号 2 所示的网络），你要做一些截然不同的事情。

Sampling a sequence from a trained RNN



第一步要做的就是对你想要模型生成的第一个词进行采样，于是你输入 $x^{<1>} = 0$ ， $a^{<0>} = 0$ ，现在你的第一个时间步得到的是所有可能的输出是经过 **softmax** 层后得到的概率，然后根据这个 **softmax** 的分布进行随机采样。**Softmax** 分布给你的信息就是第一个词 **a** 的概率是多少，第一个词是 **aaron** 的概率是多少，第一个词是 **zulu** 的概率是多少，还有第一个词是 **UNK**（未知标识）的概率是多少，这个标识可能代表句子的结尾，然后对这个向量使用例如 **numpy** 命令，`np.random.choice`（上图编号 3 所示），来根据向量中这些概率的分布进行采样，这样就能对第一个词进行采样了。

然后继续下一个时间步，记住第二个时间步需要 $\hat{y}^{<1>}$ 作为输入，而现在要做的是把刚刚采样得到的 $\hat{y}^{<1>}$ 放到 $a^{<2>}$ （上图编号 4 所示），作为下一个时间步的输入，所以不管你在第一个时间步得到的是什么词，都要把它传递到下一个位置作为输入，然后 **softmax** 层就会预测 $\hat{y}^{<2>}$ 是什么。举个例子，假如说对第一个词进行抽样后，得到的是 **The**，**The** 作为第一个

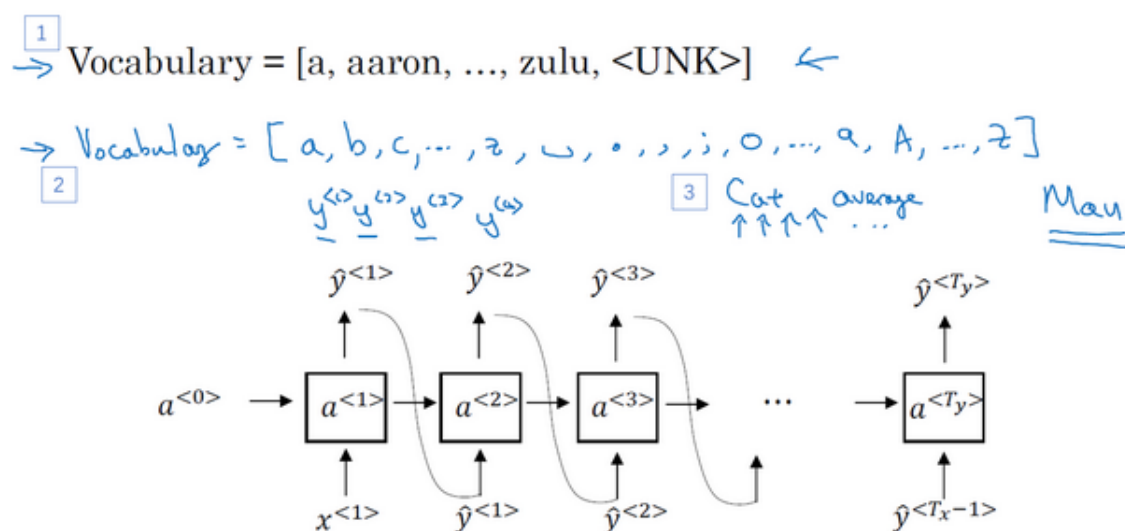
词的情况很常见，然后把 **The** 当成 $x^{<2>}$ ，现在 $x^{<2>}$ 就是 $\hat{y}^{<1>}$ ，现在你要计算出在第一词是 **The** 的情况下，第二个词应该是什么（上图编号 5 所示），然后得到的结果就是 $\hat{y}^{<2>}$ ，然后再次用这个采样函数来对 $\hat{y}^{<2>}$ 进行采样。

然后再到下一个时间步，无论你得到什么样的用 **one-hot** 码表示的选择结果，都把它传递到下一个时间步，然后对第三个词进行采样。不管得到什么都把它传递下去，一直这样直到最后一个时间步。

那么你要怎样知道一个句子结束了呢？方法之一就是，如果代表句子结尾的标识在你的字典中，你可以一直进行采样直到得到 **EOS** 标识（上图编号 6 所示），这代表着已经抵达结尾，可以停止采样了。另一种情况是，如果你的字典中没有这个词，你可以决定从 20 个或 100 个或其他个单词进行采样，然后一直将采样进行下去直到达到所设定的时间步。不过这种过程有时候会产生一些未知标识（上图编号 7 所示），如果你要确保你的算法不会输出这种标识，你能做的一件事就是拒绝采样过程中产生任何未知的标识，一旦出现就继续在剩下的词中进行重采样，直到得到一个不是未知标识的词。如果你不介意有未知标识产生的话，你也可以完全不管它们。

这就是你如何从你的 **RNN** 语言模型中生成一个随机选择的句子。直到现在我们所建立的是基于词汇的 **RNN** 模型，意思就是字典中的词都是英语单词（下图编号 1 所示）。

Character-level language model



根据你实际的应用，你还可以构建一个基于字符的 **RNN 结构**，在这种情况下，你的字典仅包含从 **a** 到 **z** 的字母，可能还会有空格符，如果你需要的话，还可以有数字 0 到 9，如

如果你想区分字母大小写，你可以再加上大写的字母，你还可以实际地看一看训练集中可能会出现字符，然后用这些字符组成你的字典（上图编号 2 所示）。

如果你建立一个基于字符的语言模型，比起基于词汇的语言模型，你的序列 $\hat{y}^{<1>}$, $\hat{y}^{<2>}$, $\hat{y}^{<3>}$ 在你的训练数据中将会是单独的字符，而不是单独的词汇。所以对于前面的例子来说，那个句子（上图编号 3 所示），“Cats average 15 hours of sleep a day.”，在该例中 **C** 就是 $\hat{y}^{<1>}$ ，**a** 就是 $\hat{y}^{<2>}$ ，**t** 就是 $\hat{y}^{<3>}$ ，空格符就是 $\hat{y}^{<4>}$ 等等。

使用基于字符的语言模型有有点也有缺点，优点就是你不必担心会出现未知的标识，例如基于字符的语言模型会将 **Mau** 这样的序列也视为可能性非零的序列。而对于基于词汇的语言模型，如果 **Mau** 不在字典中，你只能把它当作未知标识 **UNK**。不过基于字符的语言模型一个主要缺点就是你最后会得到太多太长的序列，大多数英语句子只有 10 到 20 个的单词，但却可能包含很多很多字符。所以基于字符的语言模型在捕捉句子中的依赖关系也就是句子较前部分如何影响较后部分不如基于词汇的语言模型那样可以捕捉长范围的关系，并且基于字符的语言模型训练起来计算成本比较高昂。所以我见到的自然语言处理的趋势就是，绝大多数都是使用基于词汇的语言模型，但随着计算机性能越来越高，会有更多的应用。在一些特殊情况下，会开始使用基于字符的模型。但是这确实需要更昂贵的计算力来训练，所以现在并没有得到广泛地使用，除了一些比较专门需要处理大量未知的文本或者未知词汇的应用，还有一些要面对很多专有词汇的应用。

在现有的方法下，现在你可以构建一个 **RNN** 结构，看一看英文文本的语料库，然后建立一个基于词汇的或者基于字符的语言模型，然后从训练的语言模型中进行采样。

Sequence generation

News

President enrique peña nieto, announced
sench's sulk former coming football langston
paring.

"I was not at all surprised," said hich langston.

"Concussion epidemic", to be examined. ←

The gray football the told some and this has on
the uefa icon, should money as.

Shakespeare

The mortal moon hath her eclipse in love.

And subject of this thou art another this fold.

When besser be my love to me see sabl's.

For whose are ruse of mine eyes heaves.

这里有一些样本，它们是从一个语言模型中采样得到的，准确来说是基于字符的语言模型，你可以在编程练习中自己实现这样的模型。如果模型是用新闻文章训练的，它就会生成左边这样的文本，这有点像一篇不太合乎语法的新闻文本，不过听起来，这句“**Concussion epidemic**”，**to be examined**，确实有点像新闻报道。用莎士比亚的文章训练后生成了右边这篇东西，听起来很像是莎士比亚写的东西：

“The mortal moon hath her eclipse in love.

And subject of this thou art another this fold.

When besser be my love to me see sabl's.

For whose are ruse of mine eyes heavens.”

这些就是基础的 **RNN** 结构和如何去建立一个语言模型并使用它，对于训练出的语言模型进行采样。在之后的视频中，我想探讨在训练 **RNN** 时一些更加深入的挑战以及如何适应这些挑战，特别是梯度消失问题来建立更加强大的 **RNN** 模型。下节课，我们将谈到梯度消失并且会开始谈到 **GRU**，也就是门控循环单元和 **LSTM** 长期记忆网络模型。