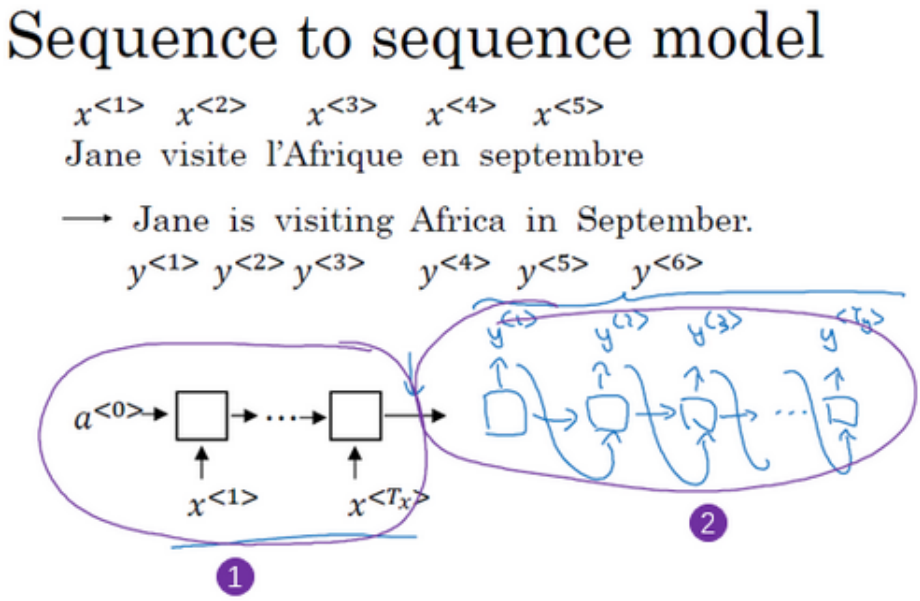
3.1 序列结构的各种序列（Various sequence to sequence architectures）

在这一周，你将会学习 seq2seq（sequence to sequence）模型，从机器翻译到语音识别，它们都能起到很大的作用，从最基本的模型开始。之后你还会学习集束搜索（Beam search）和注意力模型（**Attention Model**），一直到最后的音频模型，比如语音。

现在就开始吧，比如你想通过输入一个法语句子，比如这句 **“Jane visite l'Afrique en septembre.”**，将它翻译成一个英语句子，**“Jane is visiting Africa in September.”**。和之前一样，我们用$x^{<1>}$ 一直到$x^{<5>}$来表示输入的句子的单词，然后我们用$y^{<1>}$到$y^{<6>}$来表示输出的句子的单词，那么，如何训练出一个新的网络来输入序列x和输出序列y呢？



[Sutskever et al., 2014. Sequence to sequence learning with neural networks]

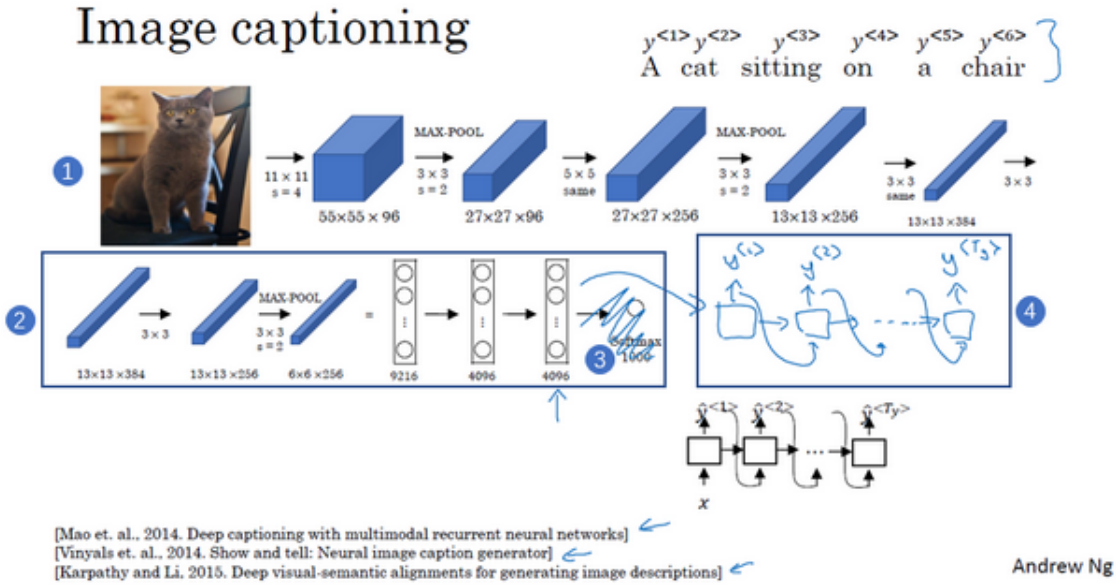
[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation]

这里有一些方法，这些方法主要都来自于两篇论文，作者是 **Sutskever**，**Oriol Vinyals** 和 **Quoc Le**，另一篇的作者是 **Kyunghyun Cho**，**Bart van Merrienboer**，**Caglar Gulcehre**，**Dzmitry Bahdanau**，**Fethi Bougares**，**Holger Schwen** 和 **Yoshua Bengio**。

首先，我们先建立一个网络，这个网络叫做编码网络（encoder network）（上图编号 1 所示），它是一个 **RNN** 的结构， **RNN** 的单元可以是 **GRU** 也可以是 **LSTM**。每次只向该网络中输入一个法语单词，将输入序列接收完毕后，这个 **RNN** 网络会输出一个向量来代表这个

输入序列。之后你可以建立一个解码网络，我把它画出来（上图编号 2 所示），它以编码网络的输出作为输入，编码网络是左边的黑色部分（上图编号 1 所示），之后**它可以被训练为每次输出一个翻译后的单词，一直到它输出序列的结尾或者句子结尾标记，**这个解码网络的工作就结束了。和往常一样我们把每次生成的标记都传递到下一个单元中来进行预测，就像之前用语言模型合成文本时一样。

深度学习在近期最卓越的成果之一就是这个模型确实有效，在给出足够的法语和英语文本的情况下，如果你训练这个模型，通过输入一个法语句子来输出对应的英语翻译，这个模型将会非常有效。这个模型简单地用一个编码网络来对输入的法语句子进行编码，然后用一个解码网络来生成对应的英语翻译。



还有一个与此类似的结构被用来做图像描述，给出一张图片，比如这张猫的图片（上图编号 1 所示），它能自动地输出该图片的描述，一只猫坐在椅子上，那么你如何训练出这样的网络？通过输入图像来输出描述，像这个句子一样。

方法如下，在之前的卷积网络课程中，你已经知道了如何将图片输入到卷积神经网络中，比如一个预训练的 **AlexNet** 结构（上图编号 2 方框所示），然后让其学习图片的编码，或者学习图片的一系列特征。现在幻灯片所展示的就是 **AlexNet** 结构，我们去掉最后的 **softmax** 单元（上图编号 3 所示），这个预训练的 **AlexNet** 结构会给你一个 4096 维的特征向量，向量表示的就是这只猫的图片，所以这个预训练网络可以是图像的编码网络。现在你得到了一个 4096 维的向量来表示这张图片，接着你可以把这个向量输入到 **RNN** 中（上图编号 4 方框所示），RNN 要做的就是生成图像的描述，每次生成一个单词，这和我们在之前将法语译为英语的机器翻译中看到的结构很像，现在你输入一个描述输入的特征向量，然后让网络生成一

个输出序列，或者说一个一个地输出单词序列。

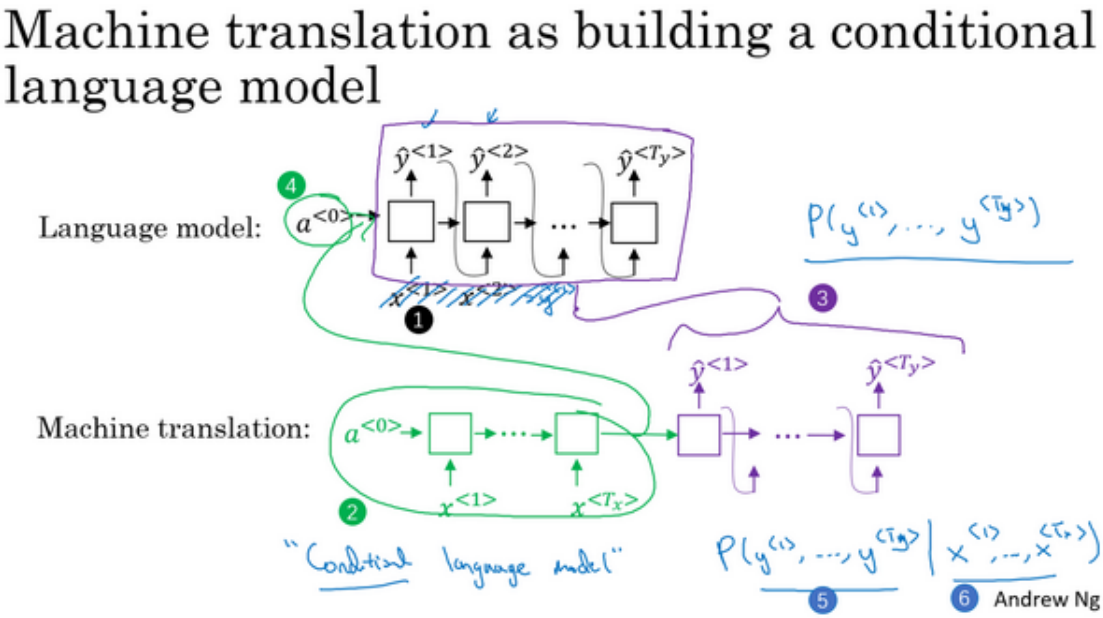
事实证明在图像描述领域，这种方法相当有效，特别是当你想生成的描述不是特别长时。

现在你知道了基本的 **seq2seq** 模型是怎样运作的，以及 **image to sequence** 模型或者说图像描述模型是怎样运作的。不过这两个模型运作方式有一些不同，主要体现在如何用语言模型合成新的文本，并生成对应序列的方面。一个主要的区别就是你大概不会想得到一个随机选取的翻译，你想要的是最准确的翻译，或者说你可能不想要一个随机选取的描述，你想要的是最好的最贴切的描述，我们将在下节视频中介绍如何生成这些序列。

3.2 选择最可能的句子（Picking the most likely sentence）

在 **seq2seq** 机器翻译模型和我们在第一周课程所用的语言模型之间有很多相似的地方，但是它们之间也有许多重要的区别，让我们来一探究竟。

你可以把机器翻译想成是建立一个条件语言模型，在语言模型中上方是一个我们在第一周所建立的模型，这个模型可以让你能够估计句子的可能性，这就是语言模型所做的事情。你也可以将它用于生成一个新的句子，如果你在图上的该处（下图编号 1 所示），有$x^{<1>}$和$x^{<2>}$，那么在该例中$x^{<2>} = y^{<1>}$，但是$x^{<1>}$、$x^{<2>}$等在这里并不重要。为了让图片看起来更简洁，我把它们先抹去，可以理解为$x^{<1>}$是一个全为 0 的向量，然后$x^{<2>}$、$x^{<3>}$等都等于之前所生成的输出，这就是所说的语言模型。

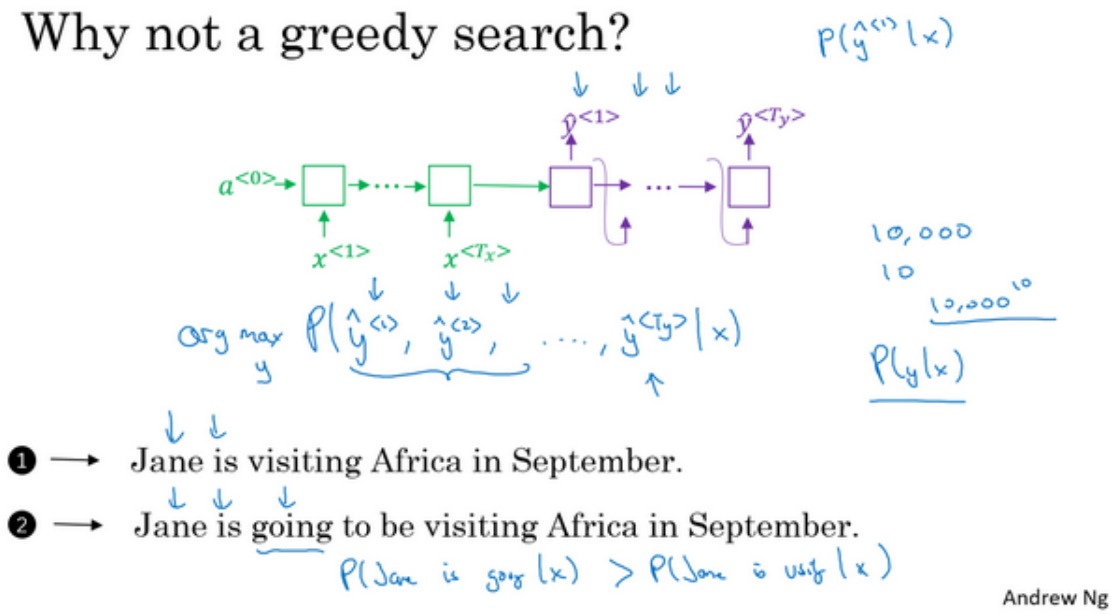


而机器翻译模型是下面这样的，我这里用两种不同的颜色来表示，即绿色和紫色，用绿色（上图编号 2 所示）表示 **encoder** 网络，用紫色（上图编号 3 所示）表示 **decoder** 网络。你会发现 **decoder** 网络看起来和刚才所画的语言模型几乎一模一样，机器翻译模型其实和语言模型非常相似，不同在于语言模型总是以零向量（上图编号 4 所示）开始，而 **encoder** 网络会计算出一系列向量（上图编号 2 所示）来表示输入的句子。有了这个输入句子，**decoder** 网络就可以以这个句子开始，而不是以零向量开始，所以我把它叫做**条件语言模型（conditional language model）**。相比语言模型，输出任意句子的概率，翻译模型会输出句子的英文翻译（上图编号 5 所示），这取决于输入的法语句子（上图编号 6 所示）。换句话说，你将估计一个英文翻译的概率，比如估计这句英语翻译的概率，**"Jane is visiting Africa in September."**，这句翻译是取决于法语句子，**"Jane visite l'Afrique en septembre."**，这就是英

语句子相对于输入的法语句子的可能性，所以它是一个条件语言模型。

现在，假如你想真正地通过模型将法语翻译成英文，通过输入的法语句子模型将会告诉你各种英文翻译所对应的可能性。x在这里是法语句子**"Jane visite l'Afrique en septembre."**，而它将告诉你不同的英语翻译所对应的概率。显然你不想让它随机地进行输出，如果你从这个分布中进行取样得到$P(y|x)$，可能取样一次就能得到很好的翻译，**"Jane is visiting Africa in September."**。但是你可能也会得到一个截然不同的翻译，**"Jane is going to be visiting Africa in September."**，这句话听起来有些笨拙，但它不是一个糟糕的翻译，只是不是最好的而已。有时你也会偶然地得到这样的翻译，**"In September, Jane will visit Africa."**，或者有时候你还会得到一个很糟糕的翻译，**"Her African friend welcomed Jane in September."**。所以当你使用这个模型来进行机器翻译时，你并不是从得到的分布中进行随机取样，而是你要找到一个英语句子y（上图编号 1 所示），使得条件概率最大化。所以在开发机器翻译系统时，你需要做的一件事就是想出一个算法，用来找出合适的y值，使得该项最大化，而解决这种问题最通用的算法就是**束搜索(Beam Search)**，你将会在下节课见到它。

不过在了解束搜索之前，你可能会问一个问题，为什么不用**贪心搜索(Greedy Search)**呢？贪心搜索是一种来自计算机科学的算法，生成第一个词的分布以后，它将会根据你的条件语言模型挑选出最有可能的第一个词进入你的机器翻译模型中，在挑选出第一个词之后它将会继续挑选出最有可能的第二个词，然后继续挑选第三个最有可能的词，这种算法就叫做贪心搜索，但是你真正需要的是一次性挑选出整个单词序列，从$y^{<1>}$、$y^{<2>}$到$y^{<T_y>}$来使得整体的概率最大化。所以这种贪心算法先挑出最好的第一个词，在这之后再挑最好的第二词，然后再挑第三个，这种方法其实并不管用，为了证明这个观点，我们来考虑下面两种翻译。



第一串（上图编号 1 所示）翻译明显比第二个（上图编号 2 所示）好，所以我们希望机器翻译模型会说第一个句子的$P(y|x)$比第二个句子要高，第一个句子对于法语原文来说更好更简洁，虽然第二个也不错，但是有些啰嗦，里面有很多不重要的词。但如果贪心算法挑选出了"**Jane is**"作为前两个词，因为在英语中 **going** 更加常见，于是对于法语句子来说"**Jane is going**"相比"**Jane is visiting**"会有更高的概率作为法语的翻译，所以很有可能如果你仅仅根据前两个词来估计第三个词的可能性，得到的就是 **going**，最终你会得到一个欠佳的句子，在$P(y|x)$模型中这不是一个最好的选择。

我知道这种说法可能比较粗略，但是它确实是一种广泛的现象，当你想得到单词序列$y^{<1>}$、$y^{<2>}$一直到最后一个词总体的概率时，一次仅仅挑选一个词并不是最佳的选择。当然，在英语中各种词汇的组合数量还有很多很多，如果你的字典中有 10,000 个单词，并且你的翻译可能有 10 个词那么长，那么可能的组合就有 10,000 的 10 次方这么多，这仅仅是 10 个单词的句子，从这样大一个字典中来挑选单词，所以可能的句子数量非常巨大，不可能去计算每一种组合的可能性。所以这时最常用的办法就是用一个近似的搜索算法，这个近似的搜索算法做的就是它会尽力地，尽管不一定总会成功，但它将挑选出句子y使得条件概率最大化，尽管它不能保证找到的y值一定可以使概率最大化，但这已经足够了。

最后总结一下，在本视频中，你看到了机器翻译是如何用来解决条件语言模型问题的，这个模型和之前的语言模型一个主要的区别就是，相比之前的模型随机地生成句子，在该模型中你要找到最有可能的英语句子，最可能的英语翻译，但是可能的句子组合数量过于巨大，无法一一列举，所以我们需要一种合适的搜索算法，让我们在下节课中学习集束搜索。