# 第一章 循环序列模型

# 为什么选择序列模型

序列模型，循环神经网络（RNN）之类的模型，在语音识别，自言语言处理方面作用很大。在语音识别中，给定了一个输入音频片段X，以及要求输出判断对应的文字记录Y，这个例子里面的输入和输出数据都是序列数据，因为X是按时序播放的音频，Y是一系列单词。以及音乐生成，只有输出数据Y是序列，而输入数据可以是空集，也可以是一个单一的整数，这个数可能代指你想要生成的曲子的风格，也可能是想要生成的曲子的前几个音符。无论如何，输入的X可以是空的，或者是数字，然后输出序列Y。在处理情感分类时，输入的是序列句子，输出可能是这句话对应几星。序列模型在DNA分析中也十分有用，根据序列可以知道那一段对应什么蛋白质。以及机器翻译中，视频行为识别，命名实体识别，在一个句子中找出人名地名等。所有这些问题都可以被称作使用标签数据（X,Y）作为训练集的监督学习，但是从这些例子中，可以看出，序列问题有很多不同类型，有些问题中，输入X和输出Y都是序列，就算这种情况下，X和Y可能不一样长，或者也可能有一样的长度，或者只有X或只有Y是序列。

![序列模型用在哪些情景](<https://upload-images.jianshu.io/upload_images/1779926-6c8384ecc2864d73.png?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240>)

# 数字符号

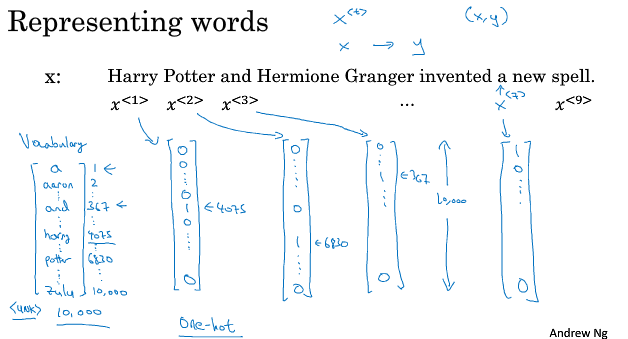
从定义符号开始一步步建立序列模型：假设输入一段文字，想要建立一个识别出句中人名位置的序列模型，这是一个命名实体识别问题，常用于搜索引擎，也可以用于查找不同文本中的人名，公司名，时间，地点，国家名等。

现在想要完成的是有一个输出Y，使得输入的每个单词都对应一个输出值，同时这个Y能够表明输入的单是否是人名的一部分，从技术上来说，这不是输出的最好形式，还有更复杂的形式，不仅能表明是否是人名，还能告诉你人名在这个句子中从哪里开始到哪里结束。视频中学习比较简单的输出形式。

输入数据：Harry Potter and Hermione Granger invented a new spell.

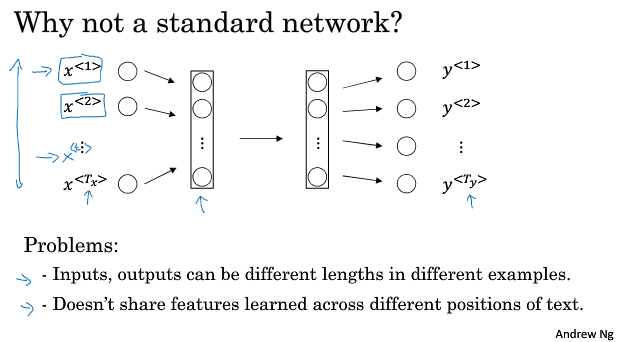
这个输入是有9个单词的序列，所以最终会有9个特征集合来表示这9个单词，并按序列中的位置进行索引。用来索引位置，他表示是时序序列，但不论是否是时序序列，都会使用t来索引序列中的位置，输出数据也一样，同索引表示，使用表示输入序列的长度，表示输出的长度。表示第i个样本的第t个元素，表示第i个样本的长度，同理也有以及。

这个问题是一个NLP问题，自然语言处理，要解决的一个问题就是如何表示一个序列里单独的单词，或者到底如何表示。首先做一个词表，列一列表示方法中要用到的单词，比如构造一个使用10000个单词大小的词典，对现代NPL来说太小，但对一般规模的商业应用来说，30000到50000大小的词典比较常见。构建这个词典的一个方法是遍历你的训练集，并且找到前10000个常用词，也可以浏览一些网络词典，接下来就可以用one-hot表示法，来表示词典里的每一个单词。也就是像softmax分类一样，对应哪个单词，则是对应位置为1，其余为0。如果词典是10000维的，那么每一个向量都是10000维的。所以指代句子里的任意词，就是一个one-bot向量，会有9个one-hot向量来表示这个句中的9个单词，也就是这一个句子。目的是用这样的表示方式表示X，用序列模型在X和目标输出Y之间学习建立一个映射。把它当成监督学习的问题，给定（x，y）标签的数据。

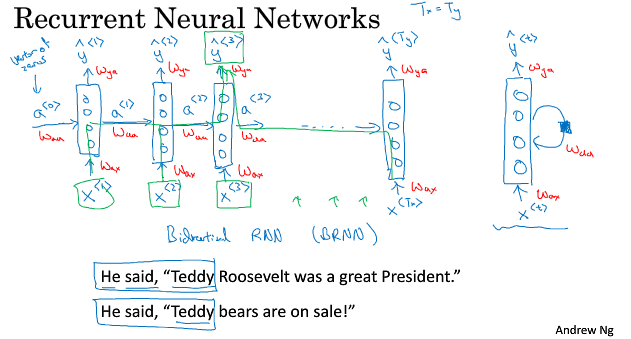


# 循环神经网络模型

建立一个模型学习X到Y的映射，尝试的方法之一使用标准的神经网络，如图，输出9个单元，分别表示是否0或1是否是人名的一部分，结果表明这个方法并不好。主要有两个问题，一个是输入和输出数据在不同例子中可以有不同的长度，即使每个例子都有最大长度，可以使用pad填充，但也不是一种好的解决方案。第二个问题，像这样单纯的神经网络并不会共享从文本的不同位置上学到的特征，具体的说，就是网络学习到了第一个Harry这个词可能是人名的一部分，出现在其他位置也应该自动识别出是人名一部分就很好。类似于卷积神经网络中看到的，希望部分图片中学到的内容快速推广到图片的其他部分，对序列数据也希望有类似的效果。并且使用一个好的模型的表达方式可以减少模型中的参数数量。



什么是循环卷积神经网络，recurrent neural network，就是在下一步时间片上输入上一步的激活值，最开始输入的可以是0向量，也可以随机其他方法初始化，不过使用0向量作为零时刻的伪激活值是最常见的选择。RNN倾向于从左向右扫描数据，同时每个时间步的参数是共享的。用来表示管理着从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，而水平参数是由参数来决定的，同时每一个时间步也使用相同的参数，输出的结果由决定。也就是说循环神经网络中，例如要得到的值，不仅需要的输入，还需要前面和的信息，这个RNN的缺点就是只使用了这个序列中之前的信息来做出预测，问题就是判断前面的单词时，仅使用前面的一点点的信息是完全不够的，后面的信息也是十分有用的。所以这个特定的结构限制就是在某一时刻的预测仅使用了从序列中之前的输入信息。BRNN（双向循环神经网络）可以解决这个问题。



接下来了解一下这个网络到底计算了什么，关于某一曾会计算两个公式：

- tanh/relu

- 输出选择哪个激活函数由你具体的分类问题来决定

这两个公式里的激活函数是不一样的。RNN经常选用的激活函数是tanh，有时候也会用relu，但是tanh更多。有其他方法来避免梯度消失。输出激活函数选择，sigmoid二分分类，softmax多分分类。不过这里的激活函数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只能是0或1。

一般公式：

如图，从左到右完成前向传播，为了帮助我们能够建立更加复杂的神经网络，简化一下符号。

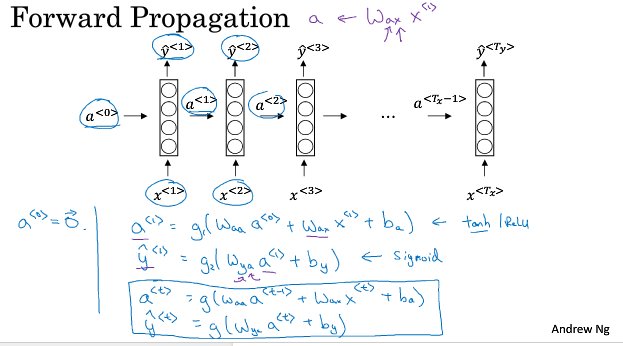
修改成：

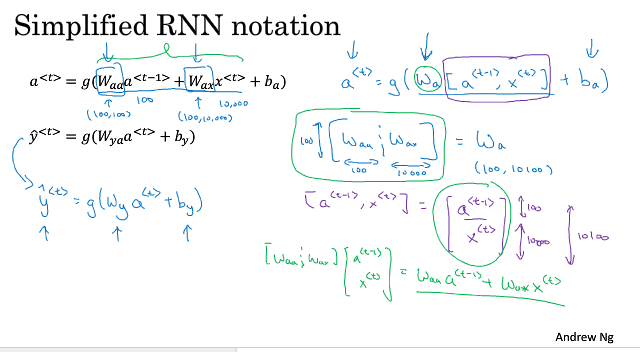
意思就是将和横向堆叠起来，然后纵向堆叠起来，如下表：

理解：假设是100维的，是10000维的，那么是一个(100,10100)的，是(100,100),是（100，10000）。那么就直接是一个10100维的。

对于下面那个式子也可以进行简化，

那么总结下来就4个参数了，，，，，下标则是用来表示计算什么量的矩阵。

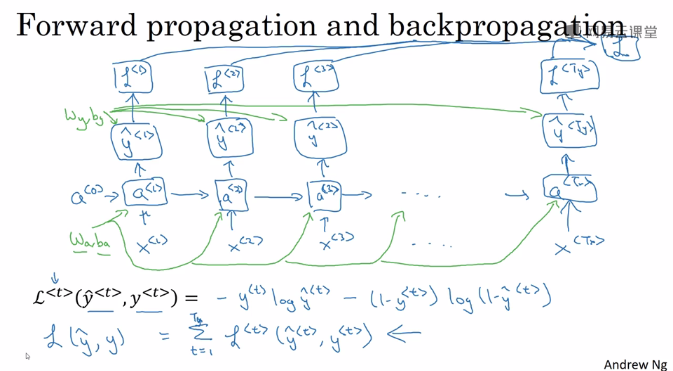




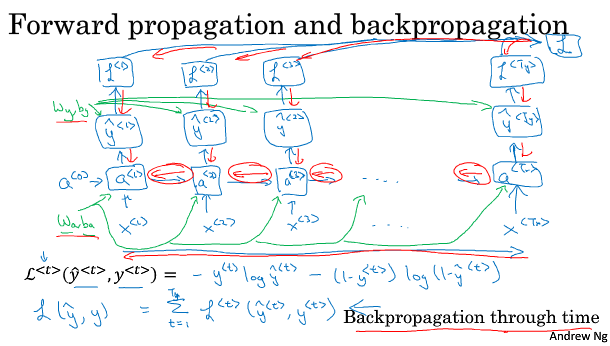
# 通过时间的反向传播

了解RNN中的反向传播，反向传播方向和前向传播是相反的，首先我们需要一个代价函数，先定义一个元素损失函数，对应着序列中的一个具体的词，如果是某个人的名字，那么，是这个词是名字的概率值，将损失函数定义为标准logistic回归损失函数，也叫交叉熵损失函数，

这就是某个时间步t上某个单词的预测值的损失函数，整个序列的损失函数就应该是：把每个单独时间步的损失都加在一起就是总的代价。



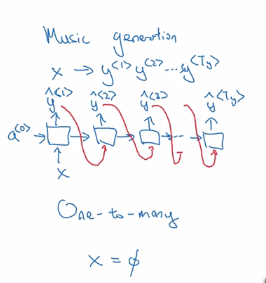
可以通过相反的传递信息的方向，计算出相关的导数等，根据梯度下降法更新参数，在这个反向传播过程中，最重要的信息传递，或者说最重要的递归运算就是从右到左的运算，这也是这个网络有一个特别的名字backpropagetion through time.通过时间的反向传播，反向传播就像时间穿越回过去。



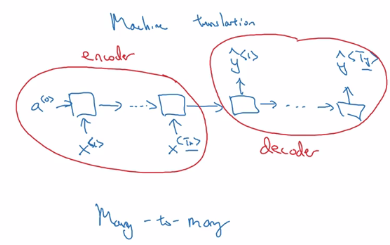
# 不同类型的循环神经网络

目前这种rnn模型，输入数量等于输出数量，就比如生成音乐中，输入长度可能是1或者0，再比如电影情感分类，输出y是关于星级评定，是1到5的整数，而输入是一段评价的序列；在机器翻译中，输入输出都是序列，但是长度不等，所以需要修改基本RNN的结构，来适用于不同的情况。

《RNN不合理的有效性》，上一节我们说到的基本RNN结构，也叫做多对多的结构，因为输入和输出都有多个例子，而在电影评论情感分析里面，X是一段输入文本，为y可能是0或1表示正面还是负面情绪，数字1到5表示评定的星级，对于这种应用的修改就是，不再让每一个都产生对应的，而是让RNN网络读入整个句子，在最后一个时间步上做出预测，这种网络就叫做多对一的结构。一对一的结构就是一个小型的神经网络。除了多对一还有一对多的结构，例子就是音乐生成，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可以是一个数字表示想要的音乐类型，或者你想要的音乐的第一个音符，或者什么都不输入，那么网络结构就可以是只有一个输入，就生成，，等，一直到合成这个音乐作品的最后一个音符，有一个技术细节，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，所以实际的网络结构如图所示：

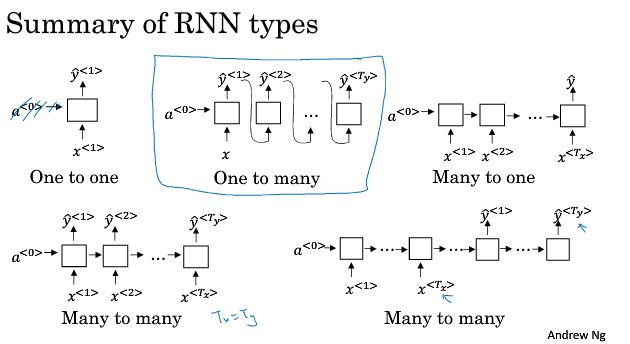


其中多对多的情况还有输入和输出长度不相等的情况没有说，机器翻译，也有一种新的网络结构，首先读入一个句子，最后先输出第一个结果，再依次输出后面的结果，直到，这种结构的和的长度就可以是不一样的了。这种网络有两个部分，左边是一个编码器，获取输入，右边是一个解码器，会读取这个句子，然后输出翻译成其他语言的结果：



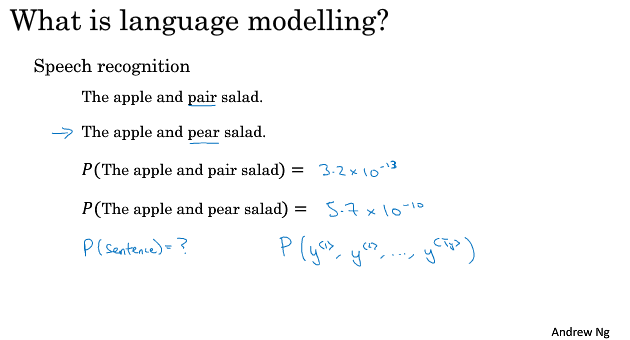
接下来就能对各种各样结构的基本构件有一个很好的理解，严格来说还有一种结构，attention based architecture注意力结构，但是根据我们现在上面说到的结构，不太好理解这个模型。

希望可以学到用这些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，序列模型还有一些不一样的地方。

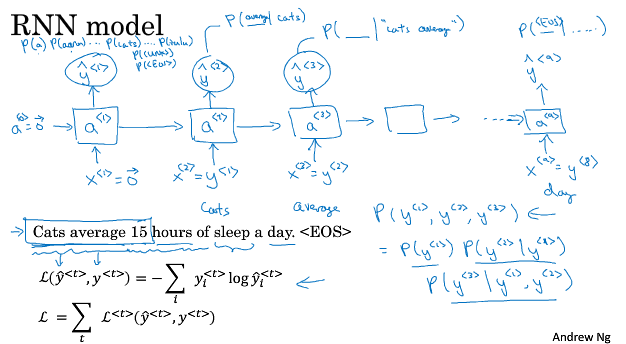


# 语言模型和序列生成

自然语言处理中构建语言模型是最基础的，能用RNN很好的实现，这一节的变成作业可以完成生成一个莎士比亚文风的文本，或者其他类型的文本，所以什么是语言模型？语音识别系统，听到一个句子，帮助输出的东西即使输入的两句话很相似，让语音识别系统去选择更准确的那个句子，The apple and pear salad,而不是The apple and pair salad，的办法就是使用语言模型，能计算出这两句话各自的可能性，语言模型的作用就是告诉你某个特定的句子它出现的概率是多少，根据所说的概率，假设你随机拿起一张报纸，打开一个任意邮件或者任何网页或者听某个人说下一句话，会是某个特定句子的概率是多少，它是两种系统的基本组成部分，一个是语音识别系统，还有机器翻译系统，它要能正确输出最接近的句子。而语言模型做的基本工作就是输入一个句子，准确的说是一个文本序列，语言模型会估计某个句子序列中各个单词出现的可能性。



为了使RNN完成这样的功能，首先需要一个训练集，包含一个很大的英文文本语料库，或者其他的你想用于构建模型的语言的语料库，语料库是自然语言处理的一个专有名词，就是很长的或者说数量众多的英文句子组成的文本。假如说你咋训练集中得到这么一句话，“Cats average 15 hours of sleep a day”，首先要做的就是将这个句子标记化tokenize，像之前视频中一样建立一个字典，然后将每个单词都转换成对应的one-hot向量，也就是字典中的索引，可能还需要定义句子的结尾，增加一个额外的标记，EOS来表示句子的结尾，能够帮助搞清楚一个句子什么时候结束，如果想要模型能够准确识别句子结尾的话，就需要在训练集中每一个句子后面都加上EOS标记。标记化的过程可以自行决定。如果要结束标记的话，那么这句话就有9个输入，分别是，一直到，在这个例子中忽略了标点符号，如果要标点符号也进行标记，那么也要加入字典中。还有个问题，如果训练集中有一些词并不在字典中，比如你字典有10000个最常用的词，句子中出现了并不是预先的常见词，这种情况下可以把Mau替换成一个叫做UNK（unkonwn）代表未知词的标志，只针对UNK建立概率模型，而不针对这个具体的少见词。完成标志化就意味着输入的所有词都映射到了各个标志上，或者说字典中的各个词上，下一步就要构建一个RNN来构建这些序列的概率模型，将设置为。现在来建立RNN模型， 会被设为全是0的集合，按照惯例也是0向量，要做的就是通过softmax进行一些预测来计算出第一个词可能会是什么，结果就是，这一步其实就是通过一个softmax层来预测字典中的任何一个词会是出现在第一个词的概率是多少，只是计算概率，而不管结果是什么，这个例子中我们的最终结果是cats，所以softmax层可能会输出10000种结果，加上UNK和EOS可能是10002种，然后RNN进入下一个时间步，的输入是cats，告诉模型第一个正确的单词是cats，这就是为什么的原因，第二层输出结果同样经过softmax层进行预测，RNN的职责就是预测这些词的概率是多少，只会考虑之前得到的词，所以在这种情况下，模型猜正确答案是average，然后再进行RNN的下一个时间步，就要计算，预测第三个词是什么，现在就要告诉它前面的两个词，cats和avergae，所以该时间步的输入，输入average之后，就要计算序列中的下一个词是什么，或者说计算出下一个词可能的概率通过之前得到的cats和average的基础上，在这种情况下正确结果会是15，一直到最后，第9个时间步，输入传给模型，然后就算，最后得到的是EOS的标志，在这一步中，通过前面的单词，不管是什么，希望能预测出EOS句子结尾标志会有很高的概率，所以RNN中的每一步，都会考虑前面得到的单词，比如给它前3个单词让它给出下个词的分布，这就是RNN如何学习从左到右的每次预测一个词，接下里，为了训练这个网络，就需要定义代价函数，



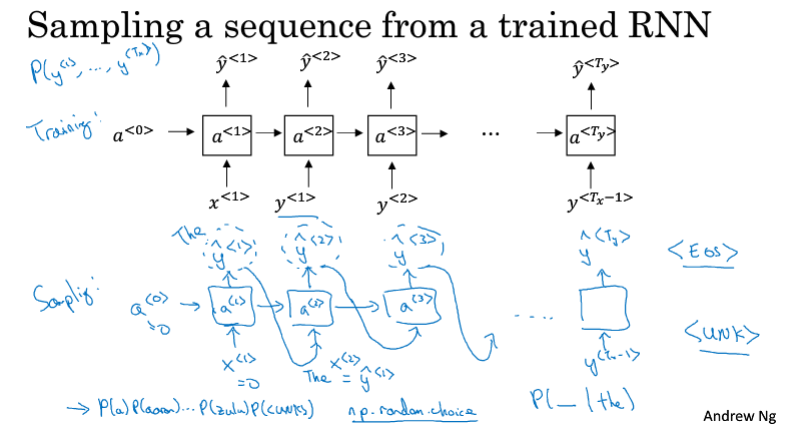
如果真正的词是，而神经网络的softmax层预测出来的结果值，这就是softmax损失函数，总体损失函数就是把所有单个预测的损失函数都相加起来。

如果你用很大的数据集来训练神经网络，就可以通过开头一系列单词来预测之后的单词的概率。现在有一个新的句子，它是，，，假如只有3个词，现在要计算出句子中各个单词的概率，方法就是第一个softmax层告诉你是的概率，第二个softmax层告诉你在考虑的情况下，的概率，同理，最后得到：

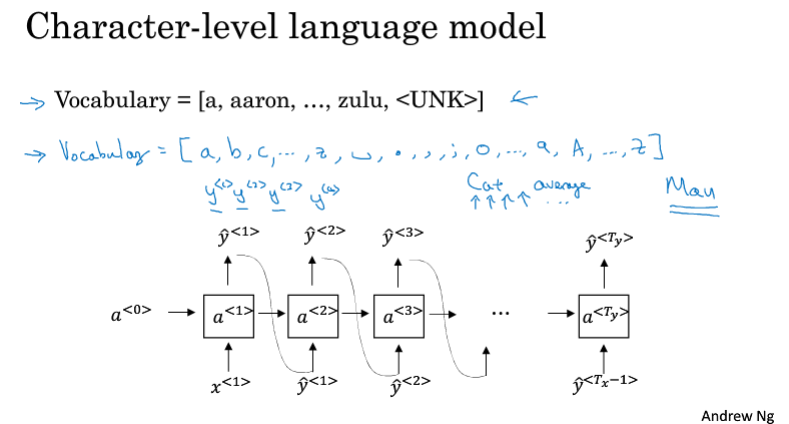
前两节中提到的是关于命名实体识别，是多对多并且输入和输出长度相同的情况，所以每一个输入都会有一个对应的，而语言模型需要的是计算这个句子可能的概率，那么就是对应每个位置的单词的概率的乘积，其中每个单词的乘积又是根据前面的所有的词来判断当前这个位置的词是某个值的概率，所以输入的时候。

# 对新序列采样

在训练了一个模型之后，想要知道这个模型学习到了什么，一种非正式的方法就是进行一次新序列采样。首先记住一个序列模型模拟了任意特定单词序列的概率，要做的就是对这个概率分布进行采样来生成一个新的单词序列，循环神经网络的每一个时间步都能通过采样得到一个词，经过每一个时间步后，就能得到预测出来的一个句子。例如，首先在第一个时间步进行采样，输入和都为0，计算后根据softmax层得出第一步后所有可能出现在首词的单词的概率，再使用numpy进行随机采样，对得到的每一个单词的概率的向量进行随机采样，使用代码：np.random.choice。然后继续进行后面的时间步，同理，需要将上一个时间步的结果词作为下一个时间步的输入，即, 继续计算softmax得到在第一个词是某个词的情况下，第二个词的采样结果。即用one-hot表示的选择结果否会进入下一个时间步，就这样一直到最后一个时间步。结果确定方法，一是根据采样结果的单词为结尾标志如EOS时，结束采样，二是可以确定时间步数或者说单词数，直到达到所设定的时间步。在这个过程中，如果出现未知符UNK，那么可以对剩下的词进行重采样，保证不输出位置词。这是RNN语言模型生成一个随机句子，字典中的词都是英语单词。



根据实际应用，还可以建立基于字符的RNN结构，字典中只含有从a到z的字母，以及空格符，数字0到9，以及如果区分大小写，还有大写A到Z，可以参考样本中可能会出现的字符，用这些字符组成你的字典，基于字符的RNN结构每个时间步的输出将会是单个字符。基于字符有优点也有缺点，优点是不再担心出现未知标识，缺点是最后得出的序列太长，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，并且基于字符的结构训练起来计算成本比较高，所以自然语言模型大多数都是使用基于词汇的语言模型，但随着计算机计算能力越来越高，基于字符也变得多了，但也没有广泛使用，除非在专门处理大量未知文本或者未知词汇的应用，或者有很多专有词汇的情况。语料库的选择可以决定最后生成的内容的风格。



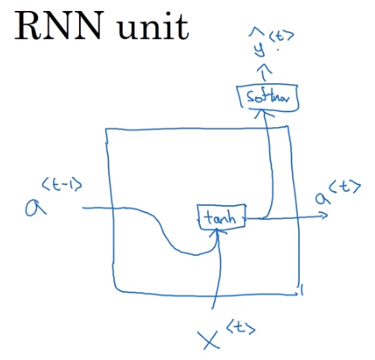
# 循环神经网络的梯度爆炸

RNN还有一个问题，梯度消失。对于循环神经网络，对于一个英语句子，有时候语法需要考虑主谓语态对应的关系，而目前最基本的的RNN不擅长捕获这种长期依赖关系。原因就在于梯度消失，深度神经网络从左到右从和输出Y的梯度很难再传播回去，很难影响靠前层的权重，同样RNN反向传播很困难，后面层的输出误差很难影响前面层的计算，这就意味着很难让神经网络意识到它需要记住看到的是单数名词还是复数名词，然后在序列后面生成依赖单复数形式的was还是were。因为这个原因，所以基本的RNN模型会有很多局部影响，比如要受很多它附近的值的影响，而中间隔得太远就很难得到信息，后面区域就很难到达序列的前面部分，因此网络很难调整序列前面的计算。这是基本RNN的一个缺点，不擅长处理长期依赖，事实上梯度下降是训练RNN的首要问题，梯度爆炸很容易发现，会看到很多NaN，或者不是数字的情况，意味着网络计算出现了数值溢出，解决办法：梯度修剪，观察梯度向量，如果大于某个阈值，缩放梯度向量，保证不会太大，通过一些最大值来修剪；然而梯度消失更难解决。

# GRU单元

Gated Recurrent unit门控循环单元，改变了Rnn的隐藏层，使其更好的捕捉深层连接，改善梯度消失的问题。在时间t处计算a公式：

其可视化如图：

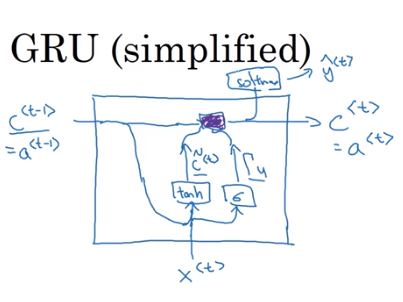


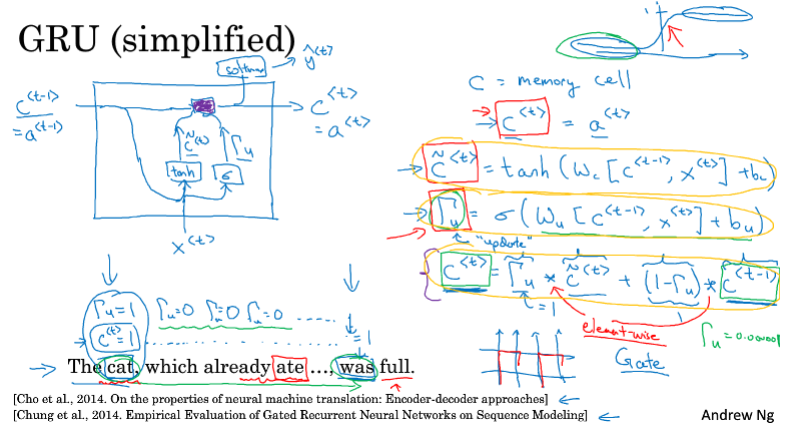
许多GRU的想法都来自于两篇论文，对于例句： The cat, which already ate… was full. 从左到右读句子，有个新的变量为c，代表细胞，即记忆细胞，记忆功能，比如一只猫是单数还是复数，看到句子的时候仍能够判断句子的主语，在时间步t处，，这个公式表示了GRU的计算，将用一个候选值重写记忆细胞，即，。

GRU中重要思想是有一个门，这是一个0到1的值，实际上这个值是这样计算出来的，，对于大多数输入，这个计算结果都是很接近0或者1的数，前面那个计算公式用来更新记忆细胞c的值，而这个则是用来决定是否要更新计算的值。比如，在cat单词时，设置c的值为0或者1来表示这是单数还是复数，然后GRU单元将一直记住这个值然后在后面用来确认使用单数还是复数，所以的作用是决定什么时候会更新这个值，特别是在看到主语，是说一个新的概念的时候就是一个好的时间去更新，所以GRU计算公式：

当计算出来的为1时，则表明要更新，为0时则为不要更新，，即使过程没有更新，但是一路过去都记住了前面的值。使用图解来理解GRU：

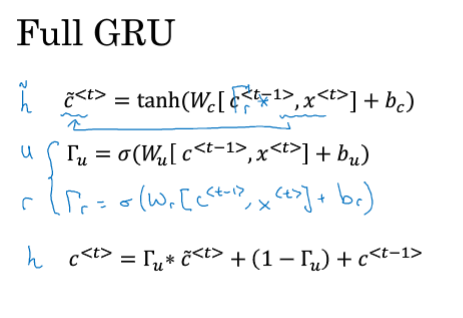
先假设上一步的，然后把它和两个用合适权重结合起来计算出，再用一个不同的参数计算出，即更新门，最后所有的值通过另一个运算符结合计算出记忆细胞的新值。





优点是通过门决定，当你从左到右扫描一个句子的时候，这个步骤是更新记忆细胞还是不更新，直到真的需要使用记忆细胞的时候，并且计算出来是个可能0.00001或者更小，就不会有梯度消失的问题了，这就是解决梯度消失问题的关键。理解，最后计算出来的激活函数也许一直到保持了前面的记忆细胞的值，就不再像普通神经网络一样梯度呈指数级变化。因此允许神经网络运行在非常庞大的依赖词上，比如cat和was中间，即便有很多很多的单词分割开。

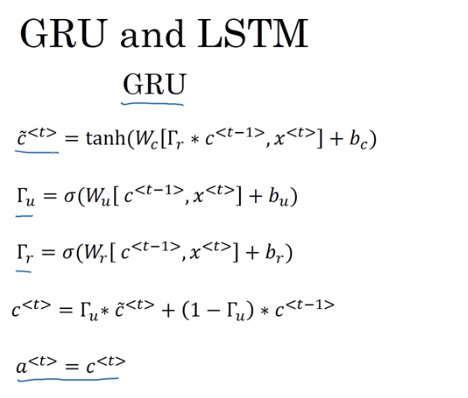
一些实现细节，其中，，以及都是向量，所以计算时是向量元素对应相乘，加入这是100维向量，其中就是100个接近0或者1的数，但其实不会真的等于0或者1，有时候是中间值，只是在理解时候只管思考比较方便，当成0或者1。记忆细胞是个向量，说明可以只用其中的一些比特去记忆猫是单数还是复数，其他比特用来理解是在讨论食物，可以每个时间点只改变一些比特。以上是简单的GRU单元，对于完整的GRU单元，还要加入一些东西。在第一个计算替代值时增加一个门，r可以认为代表相关性，这个门告诉你计算出的下一个候选值和的相关性有多大。



有很多方法可以用来设计这些类型的神经网络，研究者尝试了很多方法来设计这些单元去尝试让神经网络有更深层的连接，尝试产生更大范围的影响还有解决梯度消失的问题，GRU就是其中一个标准版本。

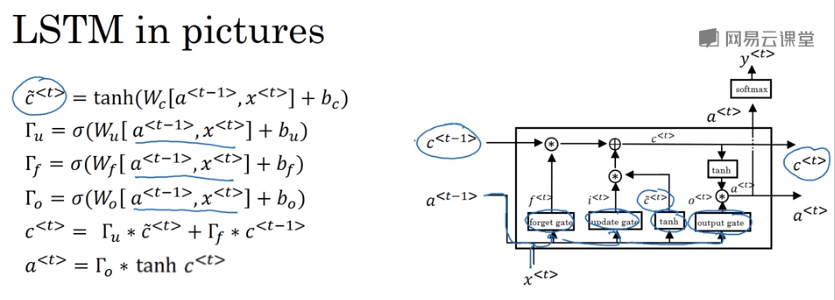
# 长短期记忆（LSTM）

GRU可以在序列中学习非常深的连接，其他类型也可以，比如LSTM，long short term memory units，长短时记忆单元，比GRU更有效。



和分别是视觉门（optic gate）和相关门（relevance gate），是代替记忆细胞的候选值，然后使用来决定是否需要更新。

LSTM是更强大和通用的版本，它的一个特性是不只有一个更新门控制，除了以外，还增加了一个遗忘门，以及一个新的sigmoid输出门。最后的计算出来的记忆细胞的公式改变，给了记忆细胞去选择权去维持旧的c值。



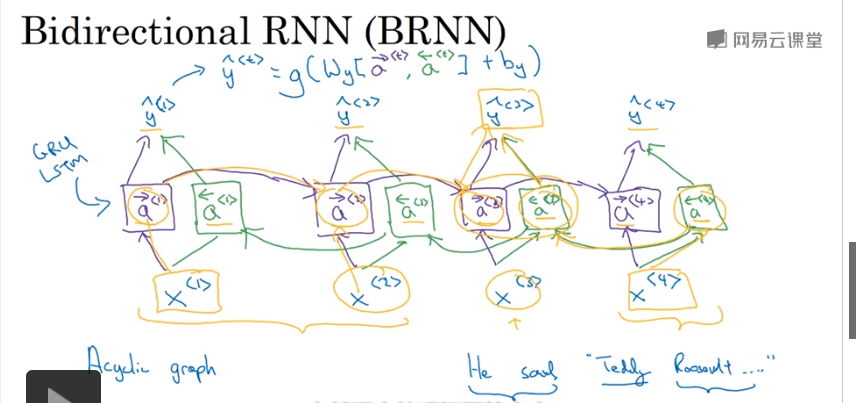
最常用一种版本可能是，窥视孔连接，a peephole connection。就是在计算门值不仅使用了与，还使用了，其实意思就是门值不仅使用了a和x值，还使用了上一个记忆细胞的值，然后就可以把三个门放在一起计算了。另外一个细节就是，记忆细胞是个维度的向量，其中第几个元素只会影响三个门的该位置的对应元素，而不会影响其他位置的元素。所以，如果说到窥视孔连接，就是指门值的计算还用到了上一个记忆细胞的值。

何时使用GRU，LSTM没有统一的标准，LSTM是最先研究出来的，GRU在使用的时候是个更简单的模型，所以更容易创建一个更大的网络，只有两个门，在计算上，运行的更快，可以扩大模型的规模，更适合规模大的问题。但是LSTM更加强大和灵活，经常需要选择的时候，LSTM也是一个默认选择。两者都可以用来建立捕获更加深层连接的神经网络。

# 双向神经网络

目前了解了很多RNN模型的关键构件，两个方法帮助建立RNN模型，一是双向神经网络，可以在序列的某点处不仅获取之前的信息，还可以获取未来的信息。二是深层RNN网络。

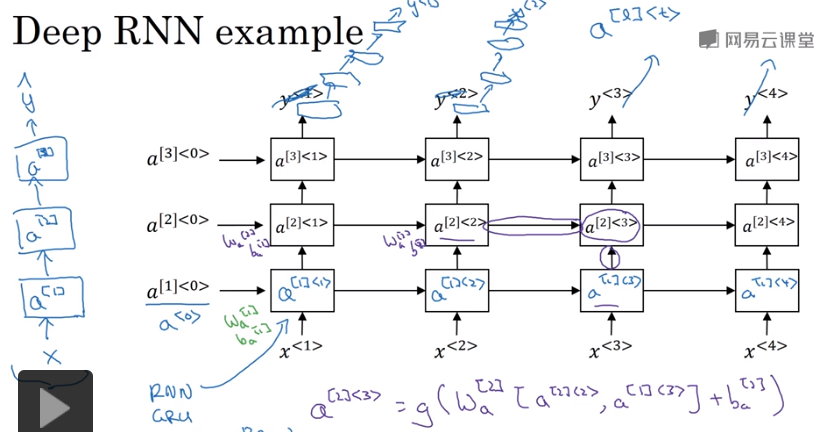
双向RNN的动机，比如在命名实体识别中，teddy只看前面出现过的单词是无法判断出到底是不是人名的。单向RNN网络，不管这些单元是标准的RNN块还是GRU或LSTM单元，只要这些构件都是只有前向的。双向RNN的工作原理，增加一个反向循环层，构成无环图，acyclic graph，给定一个输入序列，先计算前向的到，而反向序列从开始反向计算，计算的是网络激活值，所以不是反向传播，而都是正向传播，一部分计算从左到右，一部分从右到左。把正向和反向的激活值都计算完，就可以计算预测值了。



事实上，很多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模型使用的最多。所以如果有NLP问题，并且文本句子是完整的，首先需要标定这些句子，一个带LSTM单元的双向RNN模型，有正向和反向两个过程是个不错的首选。双向RNN能够预测任意位置，即使在句子中间，因为模型能考虑整个句子的信息。缺点就是需要完整的数据序列，才能预测任意位置。所以比如构建一个语音识别系统，如果你总是获取到一句完整的表达，那么使用双向RNN模型就是很有效的，但是实际上，如果要用这个模型，需要等待一个人说完，获取整个语音表达，才能进一步 做语音识别，所以对于实际的语音识别的应用，通常有更加复杂的模块。

# 深层循环神经网络

要学习复杂的函数，通常会把RNN的多个层堆叠在一起，构建更深的模型。这个深层的意思是按时间展开，比如一个有三个隐层的新的RNN网络如图，



由于时间维度上会很大，所以通常不会存在很深的RNN模型，三层也已经不少了，但是也有一种很常见的，就是在时间上增加新的隐藏深度，在计算y值时经过更过的循环层，但没有水平方向上的连接。并且上面使用的单元都不必是标准的RNN单元。也可以构件双向的深度RNN网络，但是计算成本会很高，尽管没有多少隐藏层。