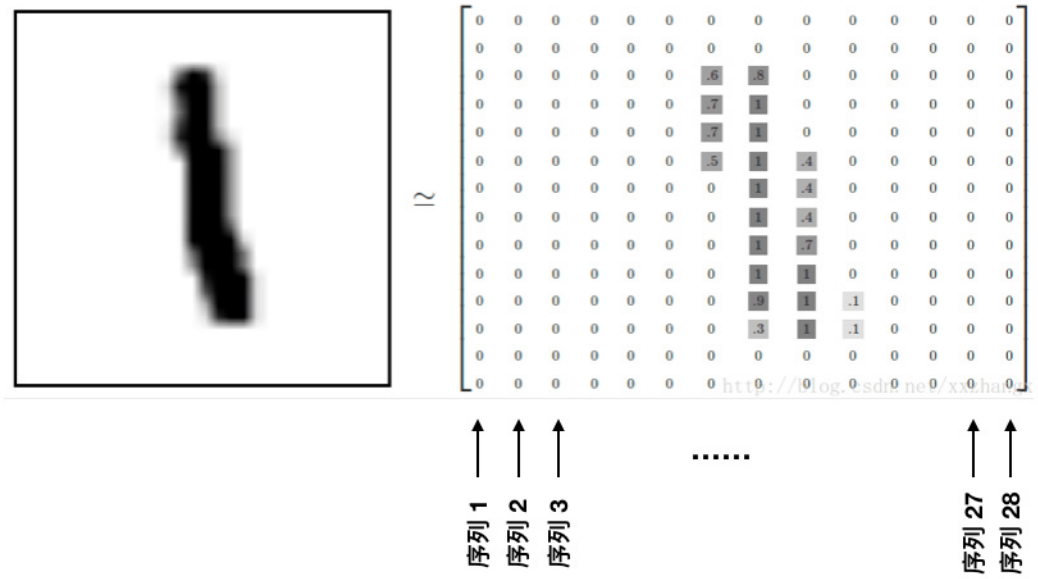
RNN的应用

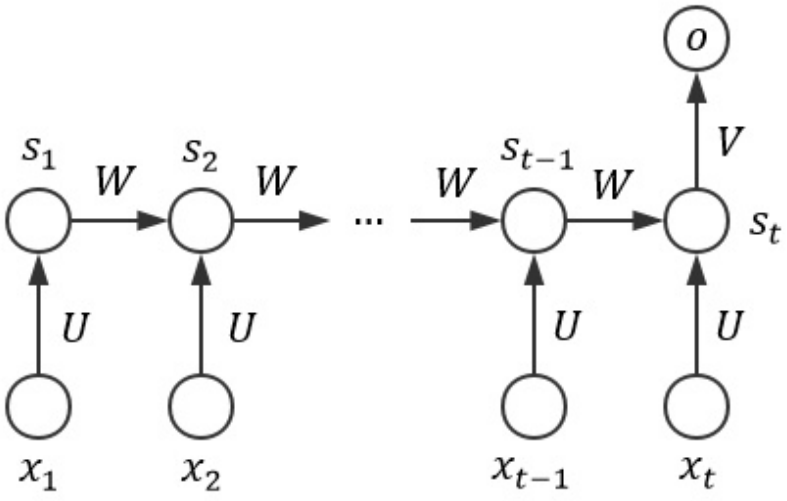
RNN 做图像分类

前面我们讲了 RNN 特别适合做序列类型的数据，那么 RNN 能不能想 CNN 一样用来做图像分类呢？下面我们用 mnist 手写字体的例子来展示一下如何用 RNN 做图像分类，但是这种方法并不是主流，这里我们只是作为举例。

对于一张手写字体的图片，其大小是 28 * 28，我们可以将其看做是一个长为 28 的序列，每个序列的特征都是 28，也就是



这样我们解决了输入序列的问题，对于输出序列怎么办呢？其实非常简单，虽然我们的输出是一个序列，但是我们只需要保留其中一个作为输出结果就可以了，这样的话肯定保留最后一个结果是最好的，因为最后一个结果有前面所有序列的信息，就像下面这样



自然语言处理的应用

词嵌入

前面讲了循环神经网络做简单的图像分类问题和飞机流量时序预测，但是现在循环神经网络最火热的应用是自然语言处理，下面我们介绍一下自然语言处理中如果运用循环神经网络，首先我们介绍一下第一个概念，词嵌入。

对于图像分类问题，我们可以使用 one-hot 的类型去编码，比如一共有 5 类，那么属于第二类就可以用 (0, 1, 0, 0, 0) 去表示，对于分类问题，这样当然忒别简单，但是在自然语言处理中，因为单词的数目过多，这样做就行不通了，比如有 10000 个不同的词，那么使用 one-hot 不仅效率低，同时还没有办法表达出单词的特点，这个时候就引入了词嵌入去表达每一个单词。

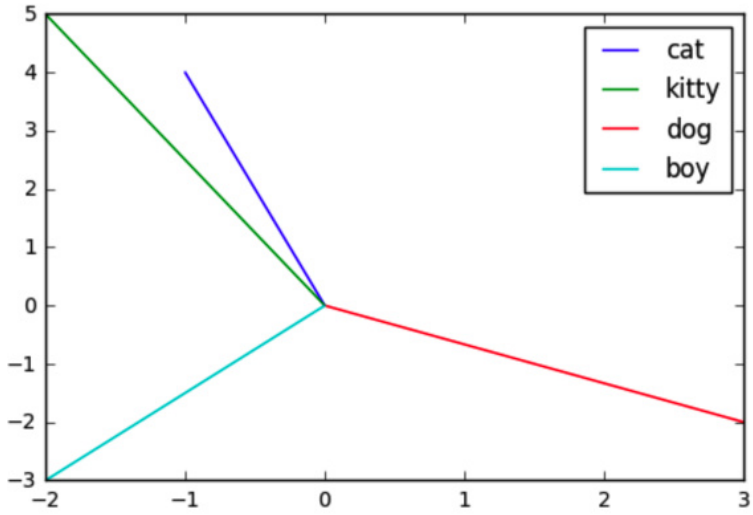
词向量简单来说就是用一个向量去表示一个词语，但是这个向量并不是随机的，因为这样并没有任何意义，所以我们需要对每个词有一个特定的向量去表示他们，而有一些词的词性是相近的，比如”(love)喜欢”和”(like)爱”，对于这种词性相近的词，我们需要他们的向量表示也能够相近，如何去度量和定义向量之间的相近呢？非常简单，就是使用两个向量的夹角，夹角越小，越相近，这样就有了一个完备的定义。

我们举一个例子，下面有 4 段话

1. The cat likes playing wool.
2. The kitty likes playing wool.
3. The dog likes playing ball.
4. The boy likes playing ball.

这里面有 4 个词，分别是 cat, kitty, dog 和 boy。下面我们使用一个二维的词向量 (a, b) 来表示每一个词，其中 a，b 分别代表着这个词的一种属性，比如 a 代表是否喜欢玩飞盘，b 代表是否喜欢玩毛线，数值越大表示越喜欢，那么我们就能够用数值来定义每一个单词。

对于 cat，我们可以定义它的词嵌入为 (-1, 4)，因为他不喜欢玩飞盘，喜欢玩毛线，同时可以定义 kitty 为 (-2, 5)，dog 为 (3, 2) 以及 boy 为 (-2, -3)，那么把这四个向量在坐标系中表示出来，就是



可以看到，上面这张图就显示了不同词嵌入之间的夹角，kitty 和 cat 之间的夹角比较小，所以他们更相似，dog 和 boy 之间的夹角很大，所以他们是不相似的。

Skip-Gram 模型

虽然我们知道了如何定义词向量的相似性，但是我们仍然不知道如何得到词嵌入，因为如果一个词嵌入式 100 维，这显然不可能人为去赋值，所以为了得到词向量，需要介绍 skip-gram 模型。

Skip-Gram 模型

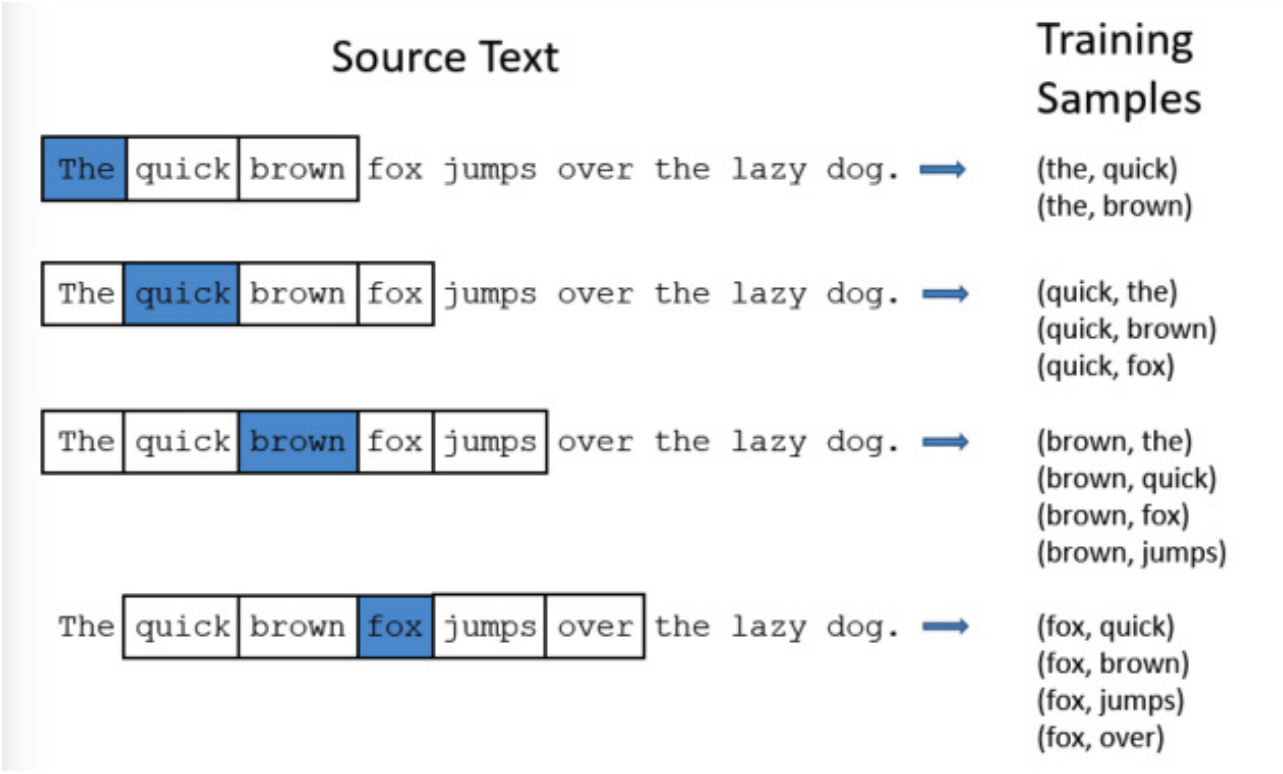
Skip Gram 模型是 Word2Vec 这篇论文的网络架构，下面我们来讲一讲这个模型。

模型结构

skip-gram 模型非常简单，我们在一段文本中训练一个简单的网络，这个网络的任务是通过一个词周围的词来预测这个词，然而我们实际上要做的就是训练我们的词嵌入。

比如我们给定一句话中的一个词，看看它周围的词，然后随机挑选一个，我们希望网络能够输出一个概率值，这个概率值能够告诉我们到底这个词离我们选择的词的远近程度，比如这么一句话 'A dog is playing with a ball'，如果我们选的词是 'ball'，那么 'playing' 就要比 'dog' 离我们选择的词更近。

对于一段话，我们可以按照顺序选择不同的词，然后构建训练样本和 label，比如



对于这个例子，我们依次取一个词以及其周围的词构成一个训练样本，比如第一次选择的词是 'the'，那么我们取其前后两个词作为训练样本，这个也可以被称为一个滑动窗口，对于第一个词，其左边没有单词，所以训练集就是三个词，然后我们在这三个词中选择 'the' 作为输入，另外两个词都是他的输出，就构成了两个训练样本，又比如选择 'fox' 这个词，那么加上其左边两个词，右边两个词，一共是 5 个词，然后选择 'fox' 作为输入，那么输出就是其周围的四个词，一共可以构成 4 个训练样本，通过这个办法，我们就能够训练出需要的词嵌入。

N-Gram 模型

上一节课，我们讲了词嵌入以及词嵌入是如何得到的，现在我们来讲讲词嵌入如何来训练语言模型，首先我们介绍一下 N-Gram 模型的原理和其要解决的问题。

对于一句话，单词的排列顺序是非常重要的，所以我们能否由前面的几个词来预测后面的几个单词呢，比如 'I lived in France for 10 years, I can speak _' 这句话中，我们能够预测出最后一个词是 French。

对于一句话 T，其由 $w_1, w_2, \cdots, w_n$ 这 n 个词构成，

$$P(T) = P(w_1)P(w_2|w_1)P(w_3|w_2w_1)\cdots P(w_n|w_{n-1}w_{n-2}\cdots w_2w_1) \tag{1}$$

我们可以再简化一下这个模型，比如对于一个词，它并不需要前面所有的词作为条件概率，也就是说一个词可以只与其前面的几个词有关，这就是马尔科夫假设。

对于这里的条件概率，传统的方法是统计语料中每个词出现的频率，根据贝叶斯定理来估计这个条件概率，这里我们就可以用词嵌入对其进行代替，然后使用 RNN 进行条件概率的计算，然后最大化这个条件概率不仅修改词嵌入，同时能够使得模型可以依据计算的条件概率对其中的一个单词进行预测。

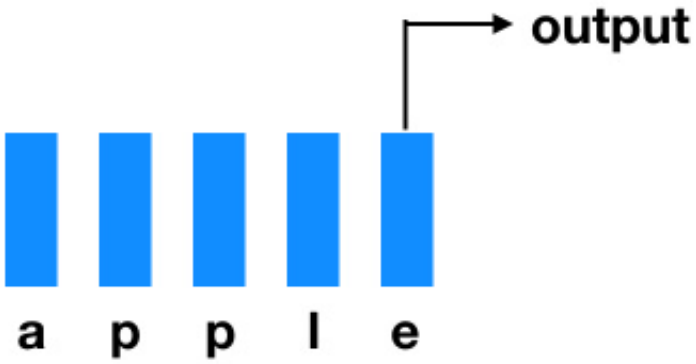
LSTM 做词性预测

前面我们讲了词嵌入以及 n-gram 模型做单词预测，但是目前还没有用到 RNN，在最后这一次课中，我们会结合前面讲的所有预备知识，教大家如何使用 LSTM 来做词性预测。

模型介绍

对于一个单词，会有这不同的词性，首先能够根据一个单词的后缀来初步判断，比如 -ly 这种后缀，很大概率是一个副词，除此之外，一个相同的单词可以表示两种不同的词性，比如 book 既可以表示名词，也可以表示动词，所以到底这个词是什么词性需要结合前后文来具体判断。

根据这个问题，我们可以使用 lstm 模型来进行预测，首先对于一个单词，可以将其看作一个序列，比如 apple 是由 a p p l e 这 5 个单词构成，这就形成了长度为 5 的序列，我们可以对这些字符构建词嵌入，然后输入 lstm，就像 lstm 做图像分类一样，只取最后一个输出作为预测结果，整个单词的字符串能够形成一种记忆的特性，帮助我们更好的预测词性。



接着我们把这个单词和其前面几个单词构成序列，可以对这些单词构建新的词嵌入，最后输出结果是单词的词性，也就是根据前面几个词的信息对这个词的词性进行分类。