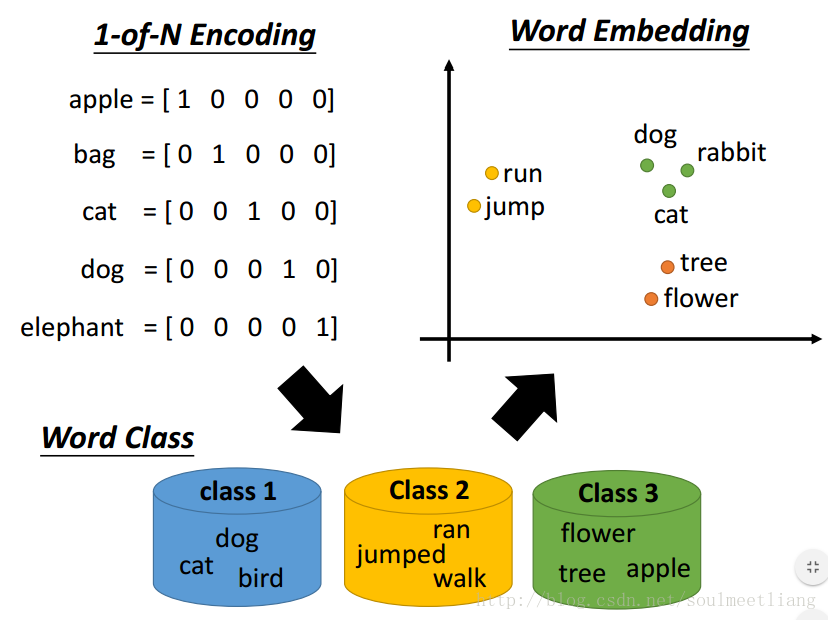
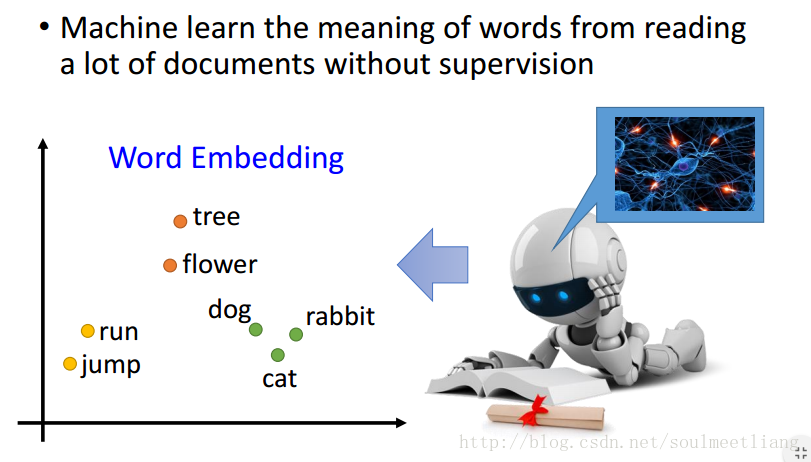
**十五.[**[机器学习](http://lib.csdn.net/base/machinelearning)**入门] 李宏毅机器学习笔记-15 （Unsupervised Learning: Word Embedding；无监督学习：词嵌入）**

# Word Embedding

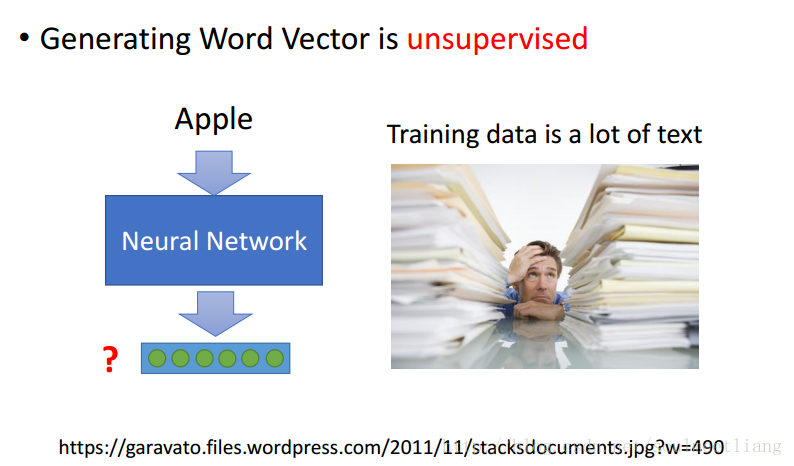
Word Embedding 是前面讲到的dimension reduction一个很好的应用。



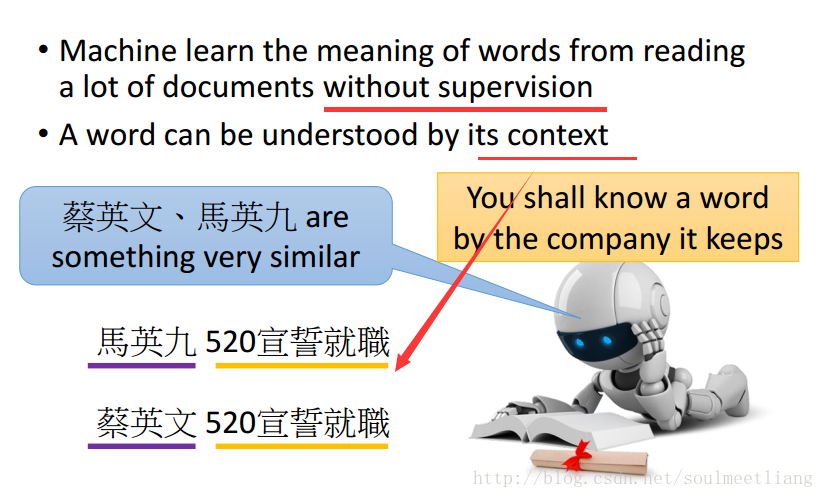
1-of-N Encoding，是把所有的word用一个向量表示，每个向量都不一样，我们没办法从中获得更多的信息，比如猫和狗都属于动物，所以这时我们用Word Class将其分为n个类型，但又有了另一个问题，我们同样无法得知两个类型之间的相关性，所以要采用Word Embedding方法，就是把每一个word都投射到一个高维空间里，两个相似的类型隔得距离较近，这样就能得到相关性。  
那么怎样让机器读懂词汇呢？



只知道输入，不知道输出，这就是一个无监督学习。

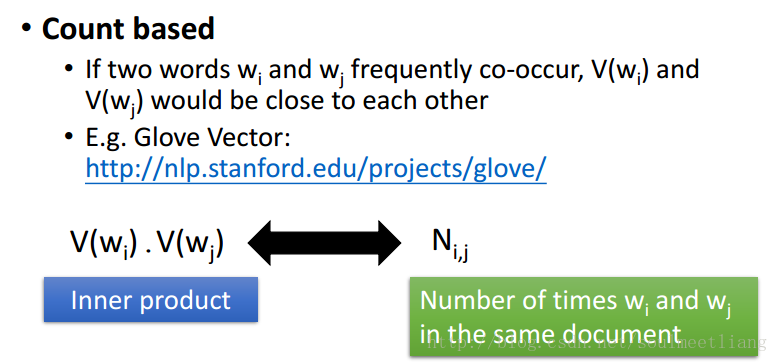


核心思想就是，通过该词语的上下文。

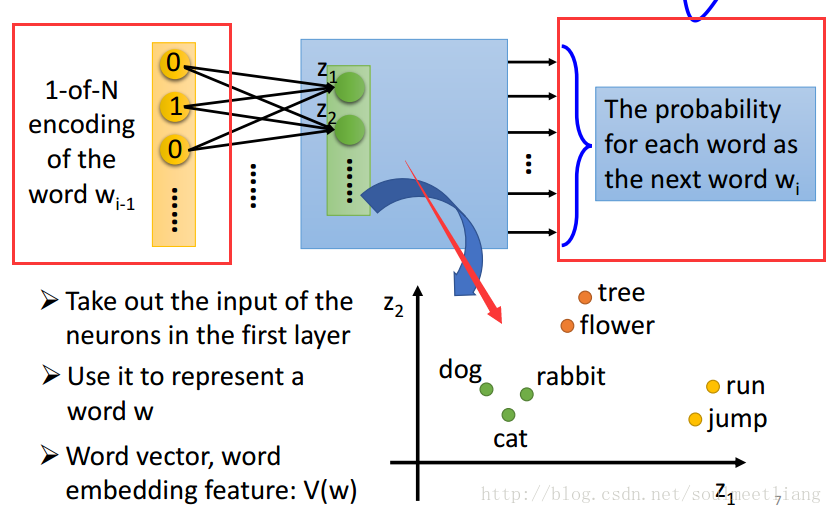


# How to exploit the context?

有两个方法：   
1.Count based

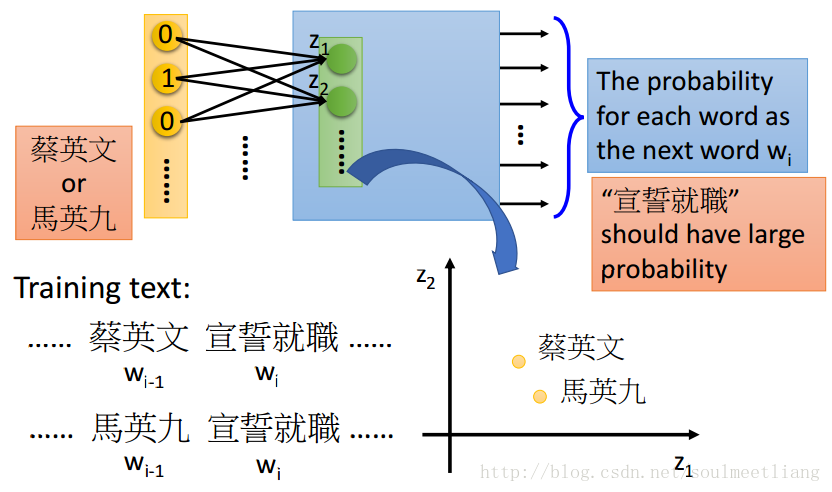


2.Prediction-based   
用 前面words 作为neural network的输入 预测下一个词的几率。



# Prediction-based

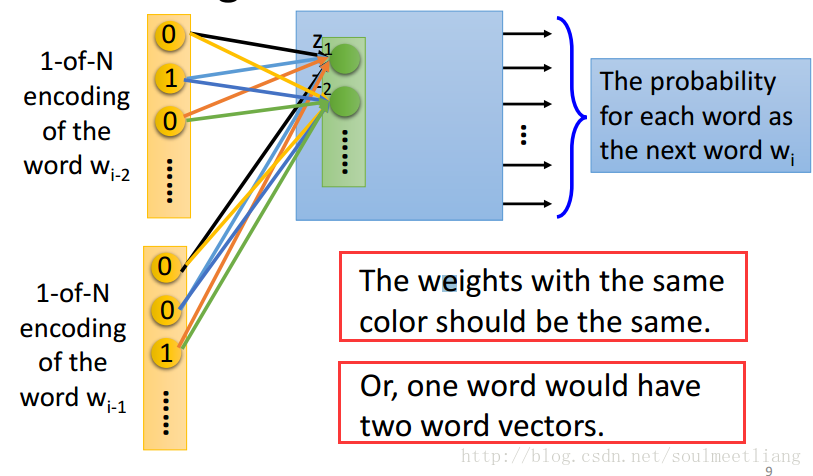
训练出neural network，所以我们将他的第一个hidden layer拿出来，就能将它们对应到相应的空间。



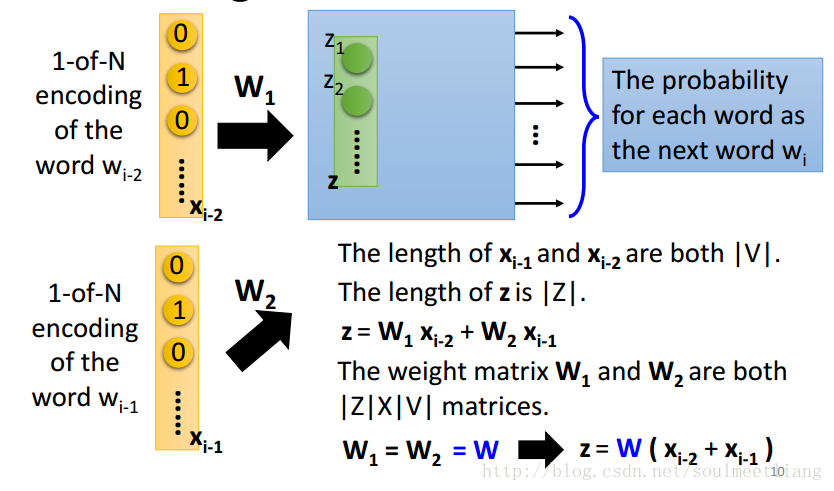
但是，仅通过一个词汇就要预测下一个词汇是很难的，所以通过Sharing Parameters来进行增强。

Sharing Parameters

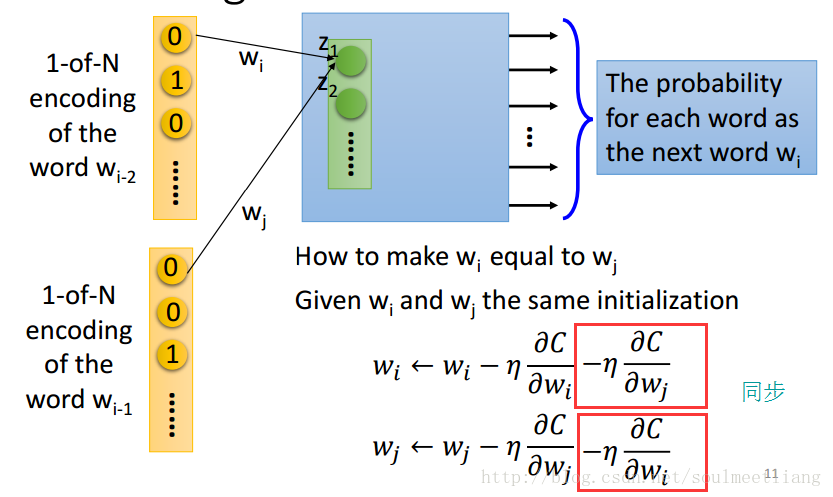
不仅用前一个word，还用前n个词来一起预测。



同样的weight是为了让同一个word放在i-1的位置和i-2的位置都有同样的transform，Or, one word would have two word vectors.另外的好处是可以减少参数量。   
计算过程：



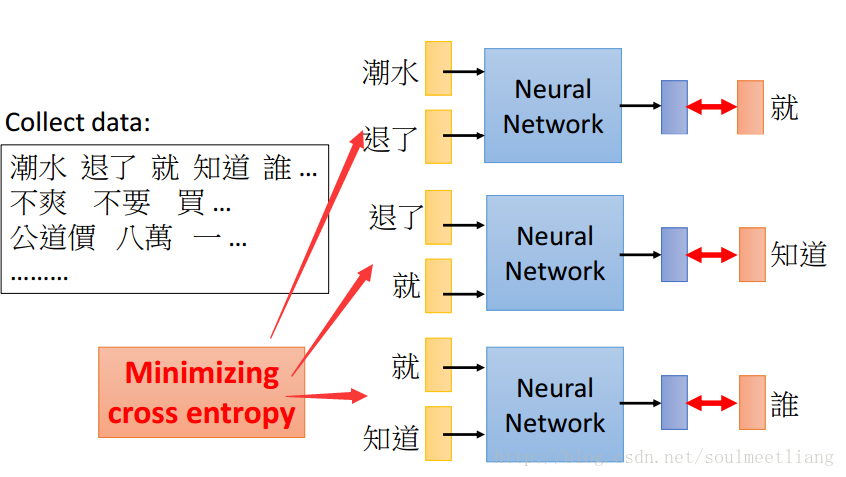
怎样让W1，W2等参数的weight一样呢？



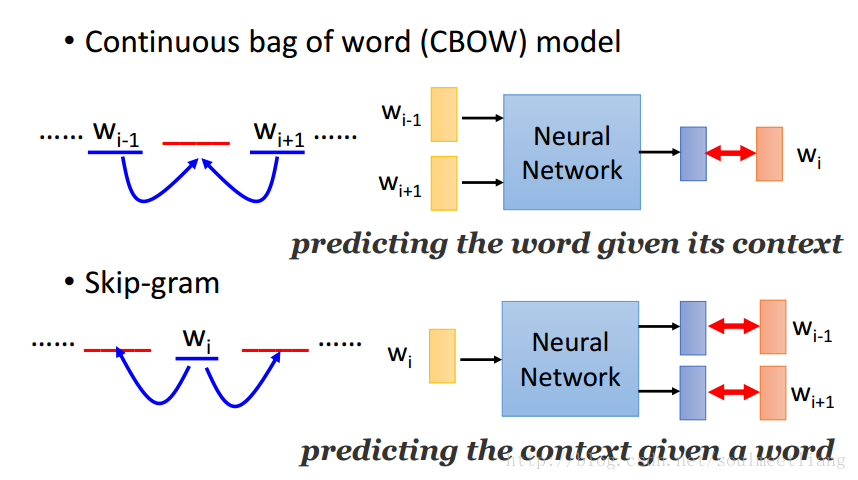
那么怎样来训练这个network呢？

Training

完全是无监督的。



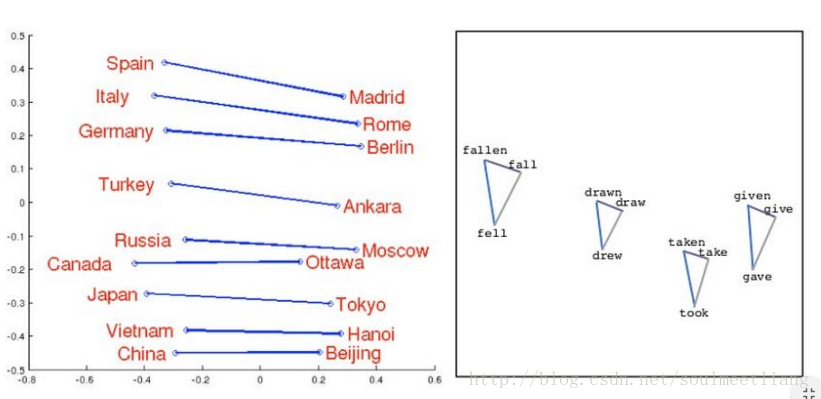
Various Architectures 有几招：



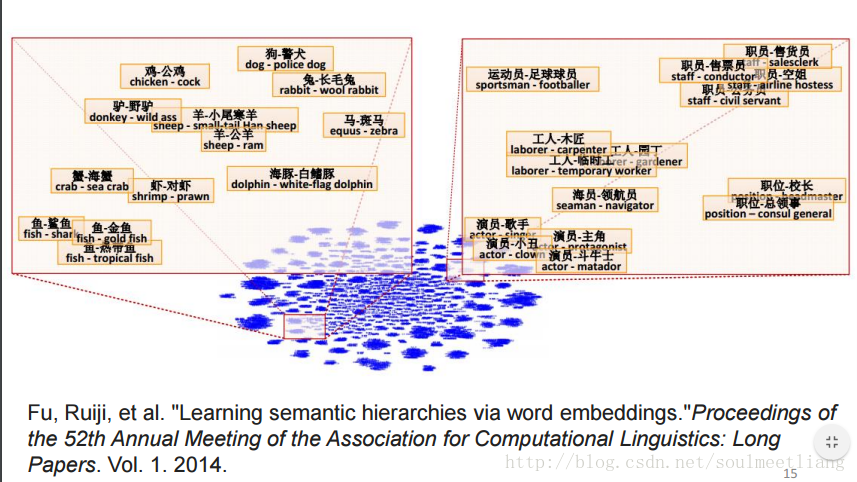
回到Word Embedding

# Word Embedding

可以发现，我们把同样类型的word摆在一起，他们之间是有固定的关系的。



所以就可以从中发现一些关系。   
比如让两个word两两相减，然后project到一个space上，如果落到同一处，则他们之间的关系是很类似的。



有了这个特性，这样我们就可以做一些预测工作。

