数据降维

2021-09-16

TA：陈姝睿

数据降维方法

- 一般情况下，当数据维度越大时，模型的复杂度会越高，训练模型所需的时间就会比较长，计算量就会越大。
- 文档数量为N，词汇表长度为V，一篇文档的one-hot或者tf-idf向量表示都是V维的。整个文档集的向量表示是 [N*V] 维的。
- **降维是指样本从高维输入空间通过线性或非线性映射投影到一个低维空间，从而找出隐藏在高维观测数据中有意义的低维结构**

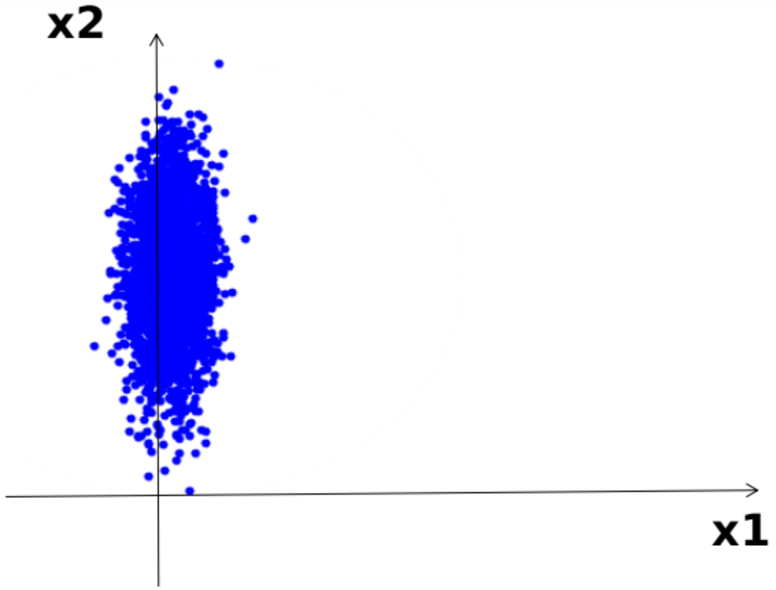
数据降维方法

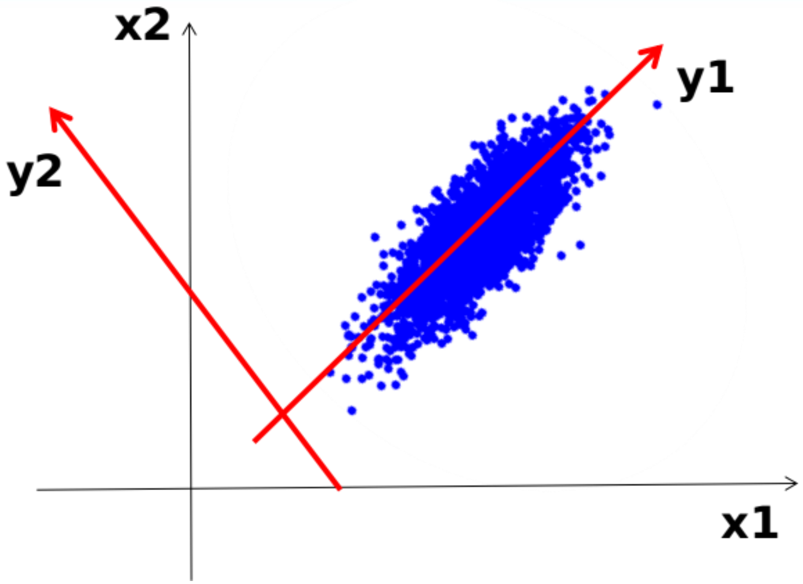
- **盲目减少属性**可能会导致重要信息丢失，导致训练模型很不准确
- 目标：减少需要分析的数量，同时降低原数据信息损失量
  - 也就是利用减少后的属性可以比较大程度地代表原数据
- 常用方法：
  - PCA 主成分分析（简单介绍）
  - LDA 线性判别分析（简单介绍）
  - Word2vec 词向量（重点介绍）

PCA (Principal Component Analysis)

- 主成分分析
- 顾名思义：找到数据中的主成分，尝试用主成分信息代替原有数据，达到数据降维的目的
- 需要使用的知识：
  - 矩阵数值中心化
  - 矩阵的特征值与特征向量
  - 向量投影

- 参考博客：https://blog.csdn.net/luoluonuoyasuolong/article/details/90711318



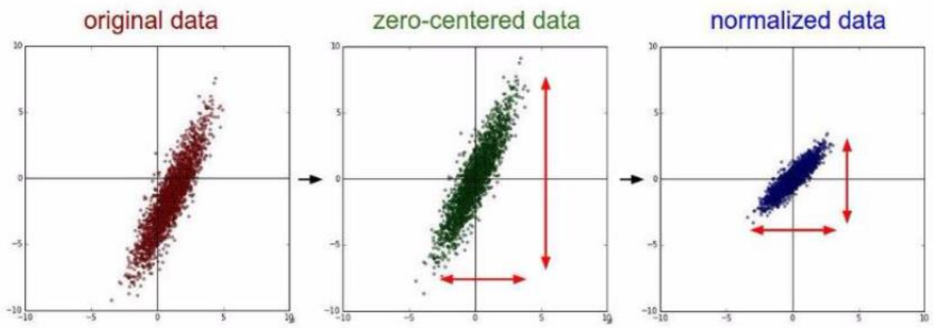


主成分分析可以让数据的投影到那些数据分布比较分散的平面上，比如上图的y1，从而忽视y2的作用，进而达到降维的目的。任何形式的变化在数学上都可以抽象成一个映射，或者函数。X是原数据，Z是降维后的数据。我们的目标是使降维后的数据在那个坐标轴中的分布尽可能分散，数据的分布的离散程度我们用方差来衡量。

$$Z_{m\times k} = X_{m\times n} W_{n\times k}$$

$$\max_{w} \frac{1}{m} \sum_{i}^{m} (z_i - \bar{z})^2 \quad s.t. \quad \|W\|_2 = 1 \quad \xrightarrow{k=1} \quad \max_{w} \frac{1}{m} wCov(x)w^T \quad s.t. \quad \|w\|_2 = 1$$

PCA (Principal Component Analysis)



- 矩阵数值中心化
  - 每一列的各数值减去该列的均值
  - 如：

3	2	3
2	3	4
4	4	2

0	-1	0
-1	0	1
1	1	-1

PCA (Principal Component Analysis)

- 计算数据矩阵的协方差矩阵
  - 假设 D 维数据，协方差矩阵是 D×D 维，假设 D = 3，协方差矩阵如下：
  - $A = \begin{matrix} cov(d_1,d_1) & cov(d_1,d_2) & cov(d_1,d_3) \\ cov(d_2,d_1) & cov(d_2,d_2) & cov(d_2,d_3) \\ cov(d_3,d_1) & cov(d_3,d_2) & cov(d_3,d_3) \end{matrix}$
  - 当数据矩阵 A 是对称矩阵的时候，其奇异值等于其特征值，且存在正交矩阵 Q，使得：
  - $Q^{-1} = Q^T$，$Q^T AQ = E$, E 为由特征值组成的对角矩阵
  - 对 A 进行奇异值分解，即可得到 A 的特征值以及特征向量

PCA (Principal Component Analysis)

- 计算数据矩阵的协方差矩阵
  - $Q^T A Q = E$, E 为由特征值组成的对角矩阵，Q 为特征向量矩阵，E 的每一列中的**非 0 值**为特征值 λ，在 Q 的对应列即该特征值对应的特征向量 ev
  - $E = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$, $Q = [ev_1, ev_2, ev_3]$
  - 此时判断 E 中的数值，哪几个的加和占据了所有特征值之和的 **95%** 以上
  - 这个**阈值**可以自由决定

PCA (Principal Component Analysis)

- 计算数据矩阵的协方差矩阵
  - 假设 $\lambda_1 + \lambda_2$ 占去了特征值总和的 95%，那么这两个特征足以代表原有的数据信息，所以我们将原数据矩阵 C （**M** 个输入向量）映射到这两个特征向量上去，**得到一个只有二维的数据矩阵**
  - $C' = C * Q_{\lambda_1+\lambda_2}, C' = C * [ev_1, ev_2]$
  - C 是 M×D，$[ev_1, ev_2]$ 是 D×2，得到的 C` 是 M×2
  - 以上就完成了 PCA 的流程
  - 对矩阵进行奇异值分解，**求解特征值和特征向量的方法自行查阅**

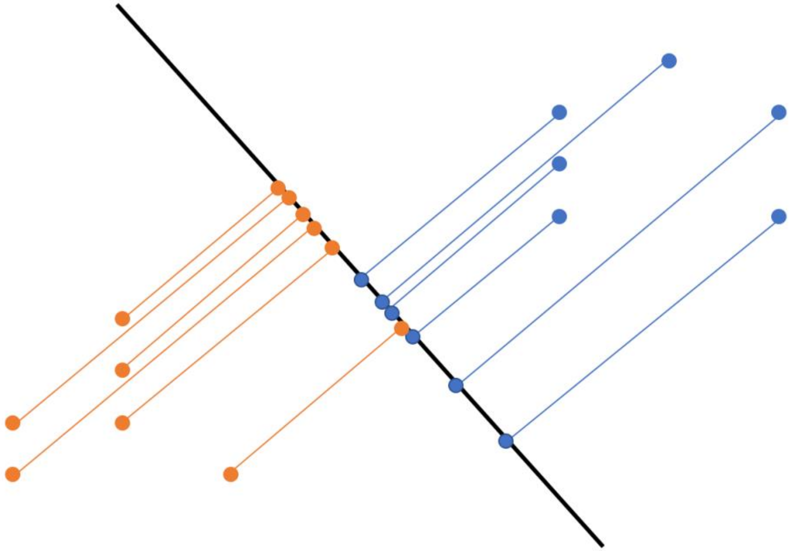
LDA (Linear Discriminant Analysis)

- 是一种有监督算法
  - 要利用到数据集的每个样本的类别标签

- 将多维的样本数据集投影到一个低维空间。投影后，使得样本在新的子空间能够尽量维持类别分组，即有最大的类间距离，也就是说样本在该空间中有最佳的可分离性。

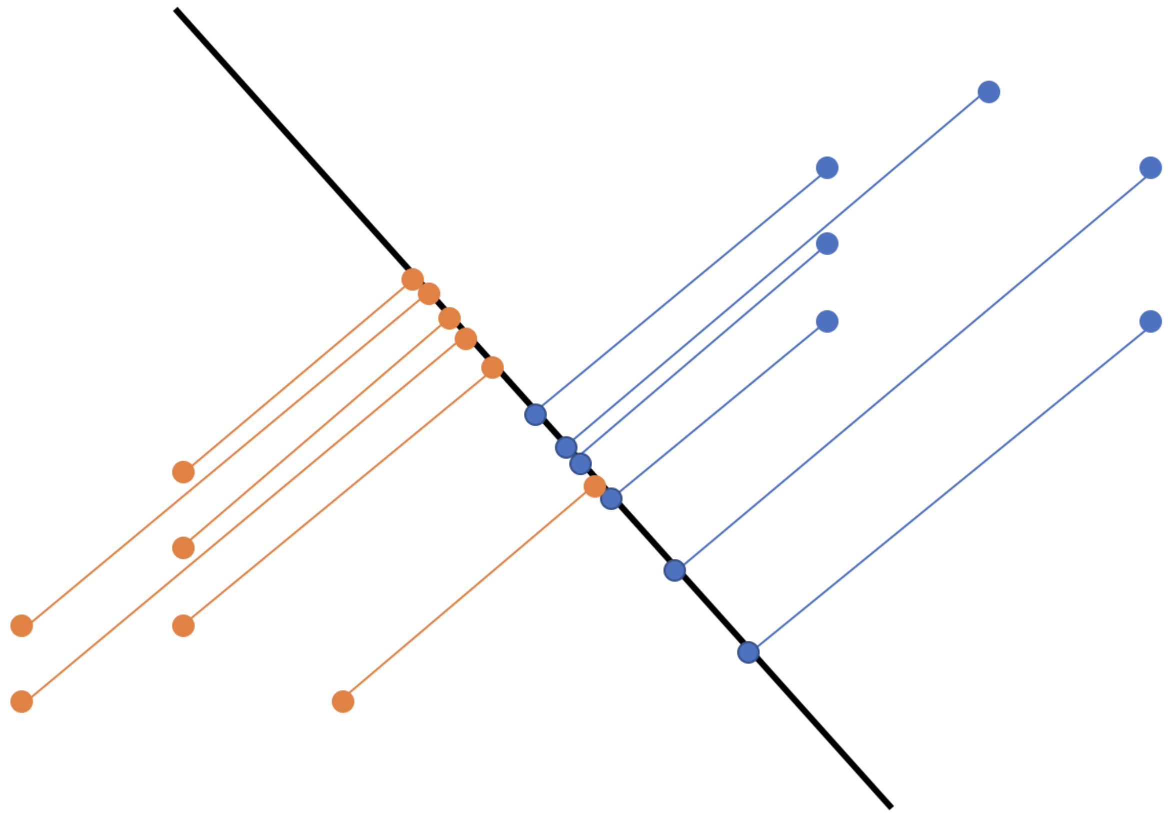
LDA (Linear Discriminant Analysis)

- 考虑二维特征，现在如果我们想要把数据的维度降到一维，但是又不想失去数据的类别信息，也就是我们想在**一维的层面同样能够反映数据所属的类别**
- 从二维到一维，其实就是找到一条直线，把所有的点投影到这条线上，就完成了降维的过程



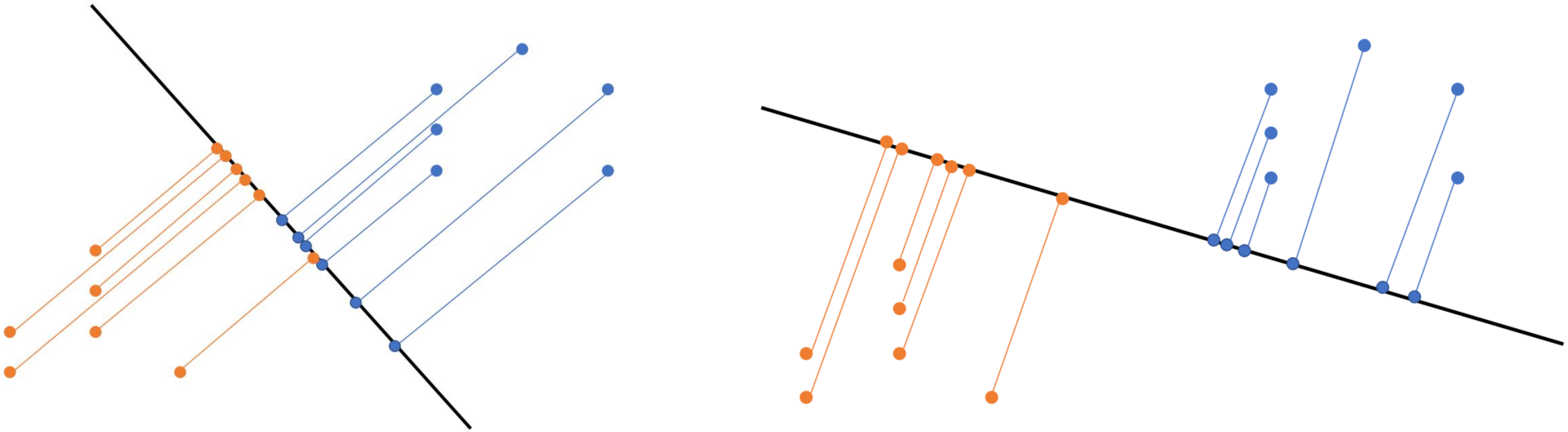
LDA (Linear Discriminant Analysis)

- 但是如果是对随意的直线进行投影，降维之后的类别可能就会很混乱，可能本来可以线性划分的数据集反倒没办法线性划分了，比如刚刚的那幅图



LDA (Linear Discriminant Analysis)

- 降维后想要**类别之间尽量的分开**

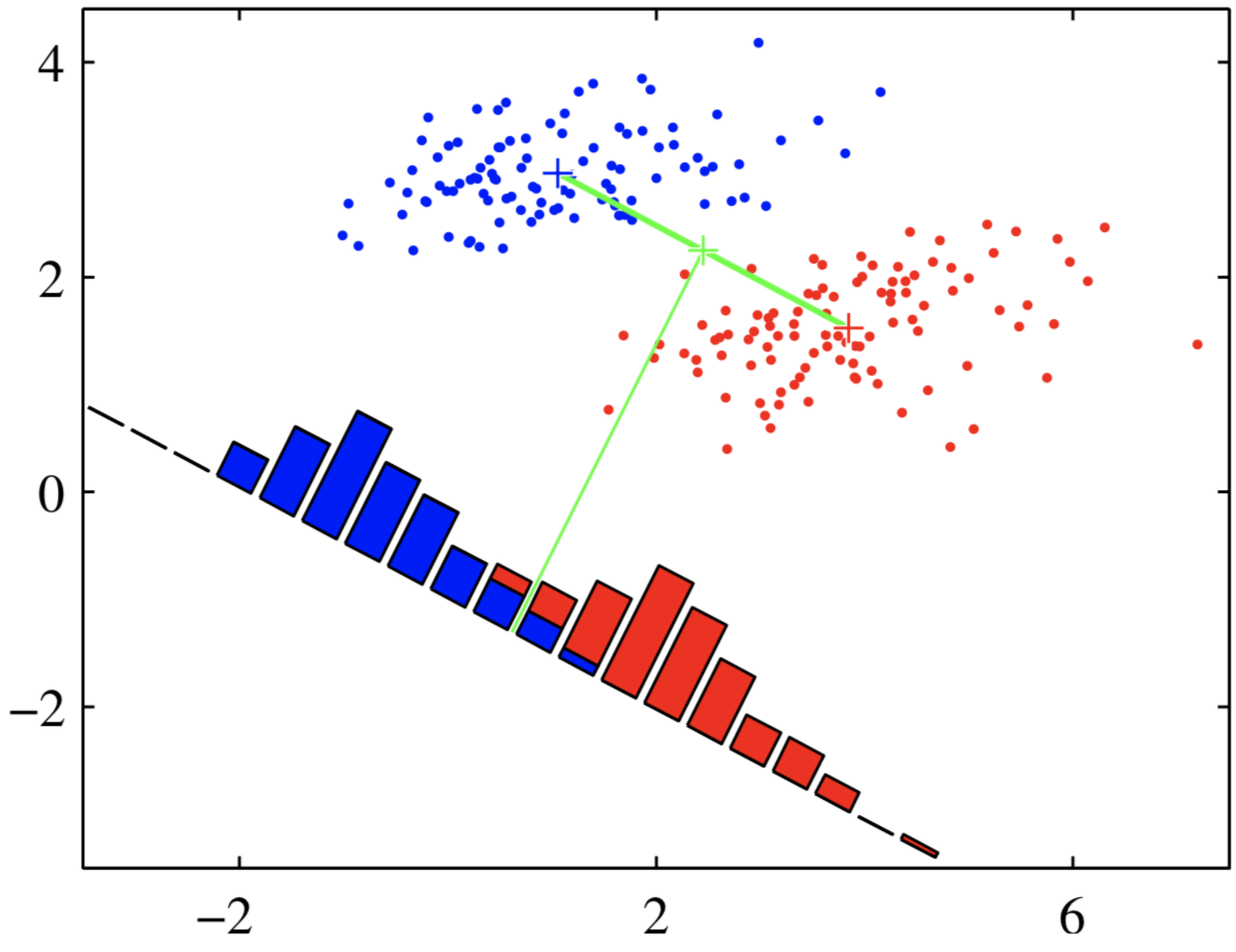


LDA (Linear Discriminant Analysis)

- 降维后想要**类别之间尽量的分开**
- 如何实现这种目的呢？对两个类别 c_1, c_2 先求出中心点
- $m_1 = \frac{1}{N_1}\sum_{n \in c_1} x_n \qquad m_2 = \frac{1}{N_2}\sum_{n \in c_2} x_n$
- **最简单的做法**就是，让映射之后的中心点 m_1', m_2' 离得越远越好
- $maximize\ (m_1' - m_2')^2$
- 那就是说选择用于投影的直线应该与两个中心点连成的直线**平行**

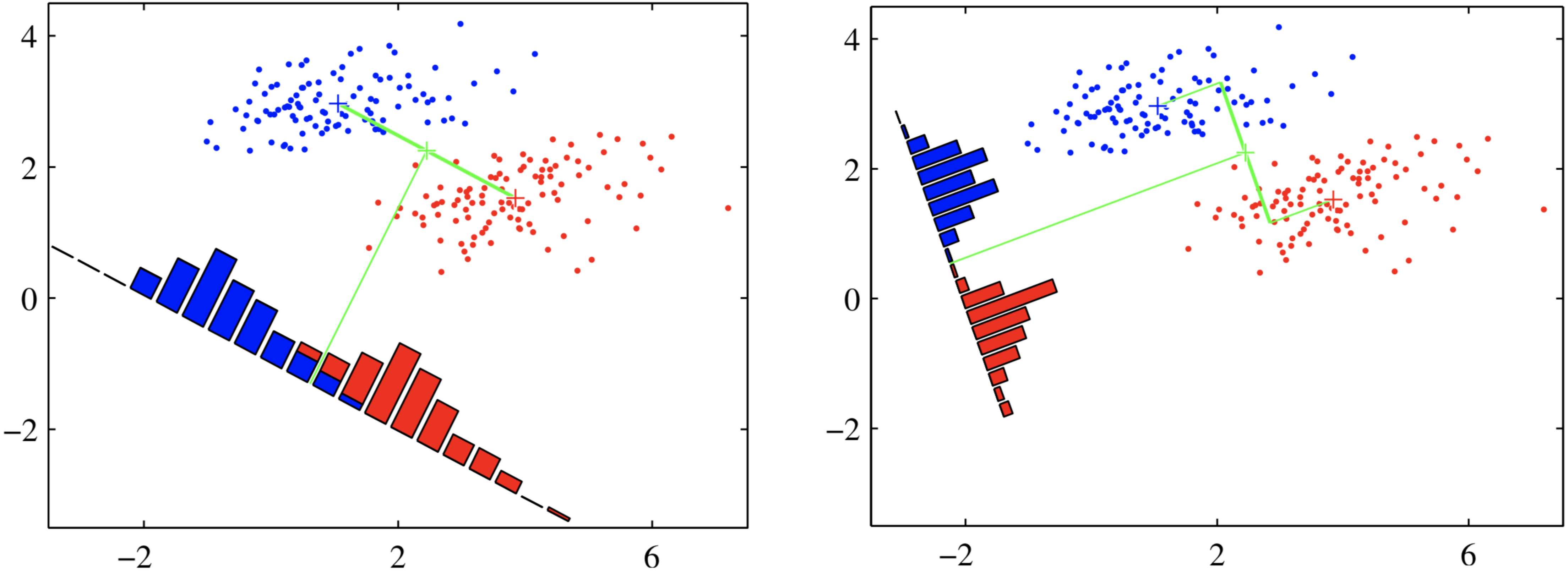
LDA (Linear Discriminant Analysis)

- 用于投影的直线应该与两个中心点连成的直线**平行**



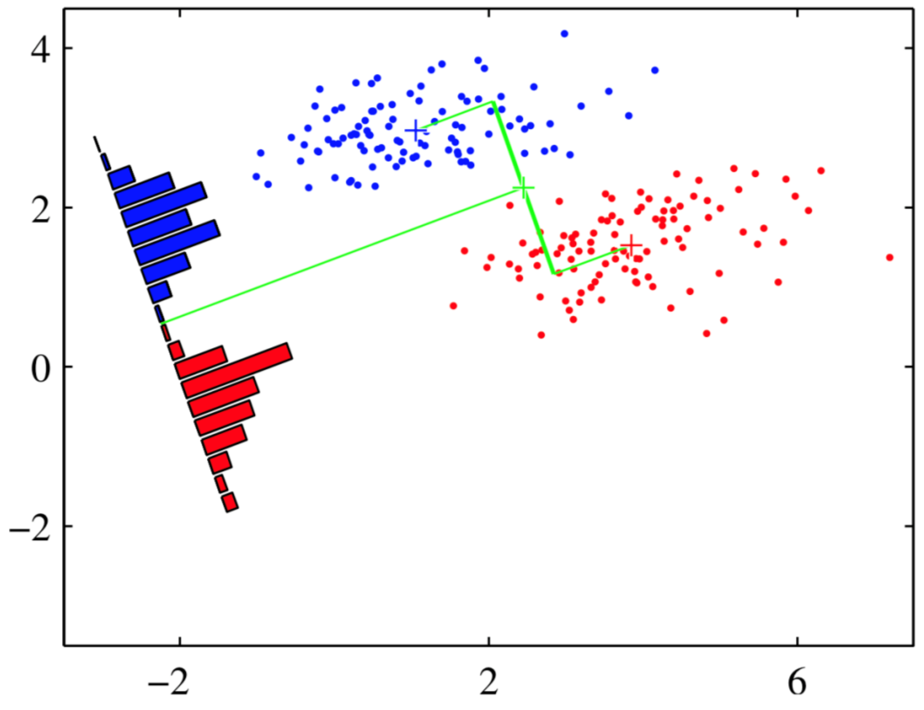
LDA (Linear Discriminant Analysis)

- 问题：然而这样也不代表着投影后有最好的划分效果，比如右图应该是比较好的划分效果，**让中心点映射之后最远的想法不完全正确**



LDA (Linear Discriminant Analysis)

- 要保证映射之后中心点尽可能远，$maximize\ (x_1 - x_2)^2$
- 同时要保证映射之后每个类别的点**尽可能地集中**，也就是说**每个类别的方差要尽可能小**
- $s_1^2 = \sum_{n \in c_1} (y_n - m_1')^2$
- $s_2^2 = \sum_{n \in c_2} (y_n - m_2')^2$
- 要保证 $s_1^2 + s_2^2$ 尽可能的小



LDA (Linear Discriminant Analysis)

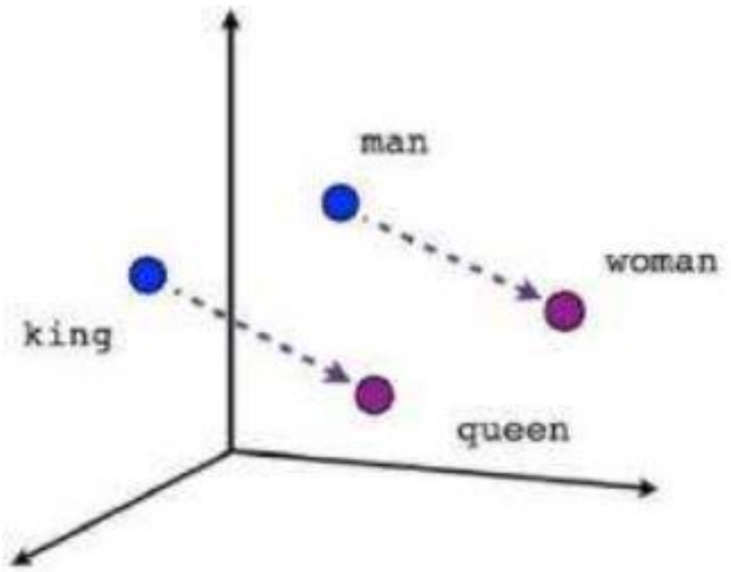
- 目标函数：

$$maximize \quad \frac{(m_1'-m_2')^2}{s_1^2+s_2^2}$$

核心思想：投影后类内方差最小，类间方差最大。

Word2Vec

- Word Embedding：指的是一种单词映射的做法，将单词从原始的数据空间映射到新的低维空间上去
- 可以用在许多文本分析任务的数据集处理上
- Skip-Gram 模型是给定 input 单词预测上下文
- CBOW 模型是给定上下文预测 input 单词



Male-Female

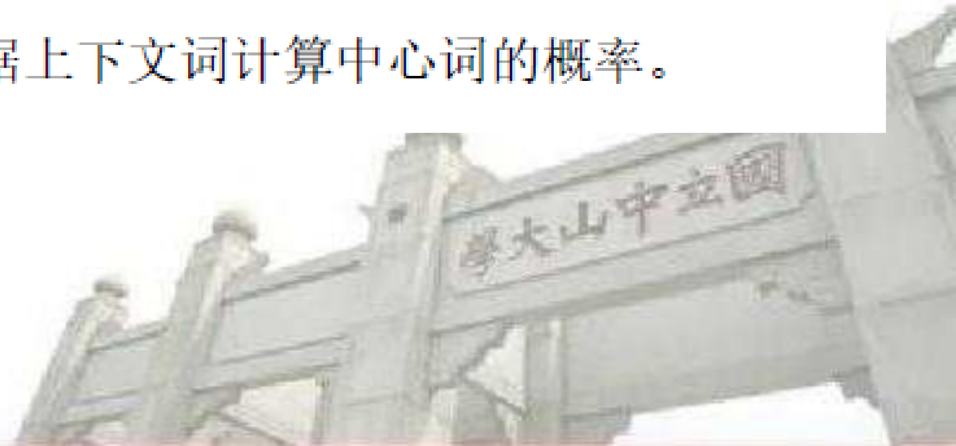
I live in China and I love China. CBOW

I live in China and I love China. Skip-Gram

表达式

$$p(context|w_t) \ or \ p(w_t|context)$$

前者是根据中心词计算上下文词的概率，后者是根据上下文词计算中心词的概率。



Word2Vec – Skip-Gram

- 举例文本：I live in China and I love China. （V = 6）
- 选择一个**输入词**，假设为 live
- 定义一个 skip-window 的参数，假设为 2，这个参数代表着从这个输入词的左侧或右侧选择的单词数量
- 那么选到的词就是 I live in China

Word2Vec – Skip-Gram

- I live in China
- 定义一个 num-skip 的参数，也假设为 2，这个参数代表着从根据skip-window选出的单词中，选择多少个不同的词作为**输出词**
- 假设选到的词是 I 和 in
- 那么我们就得到了两组训练数据，(live, I) (live, in)

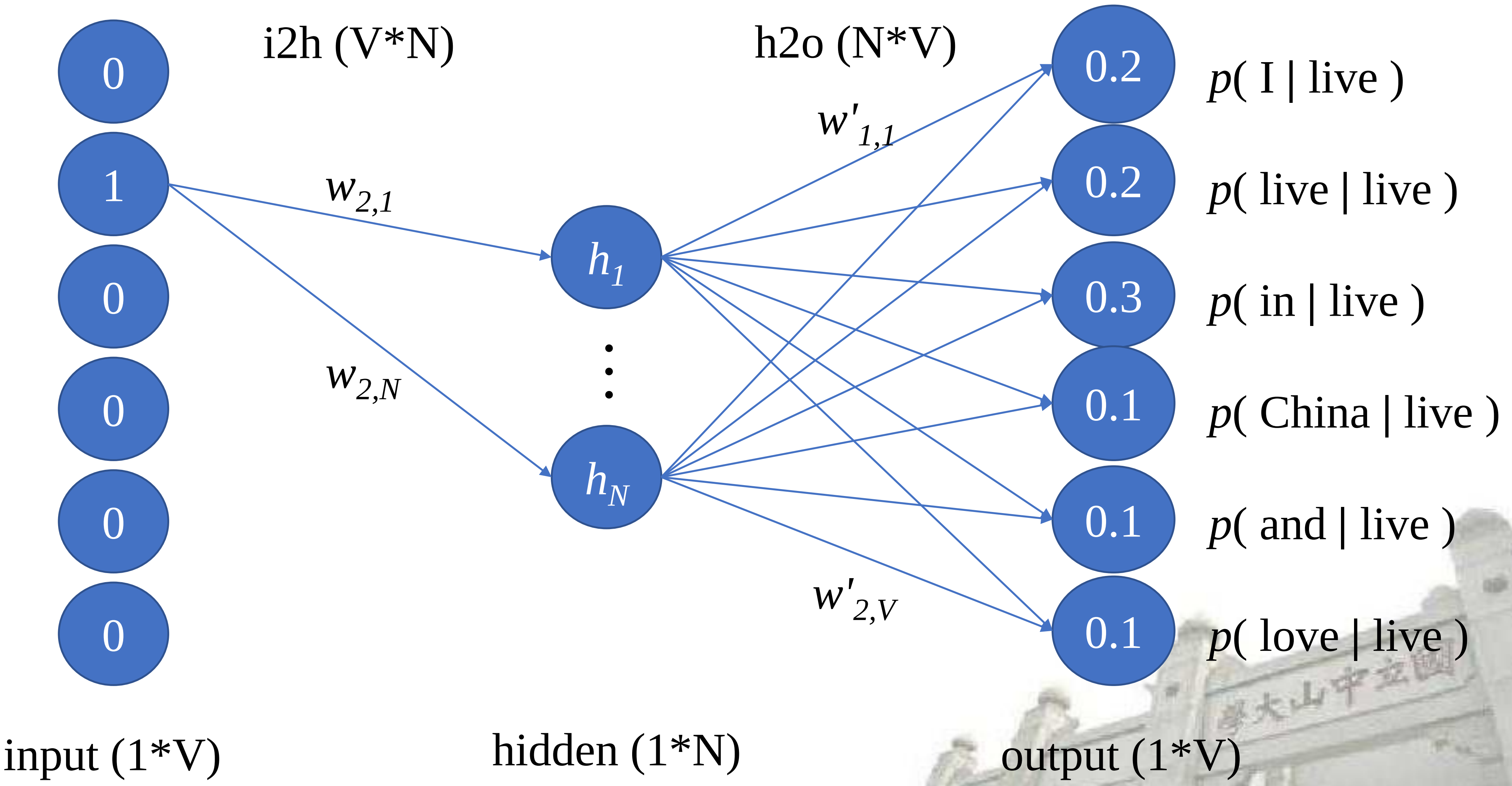
Word2Vec – Skip-Gram

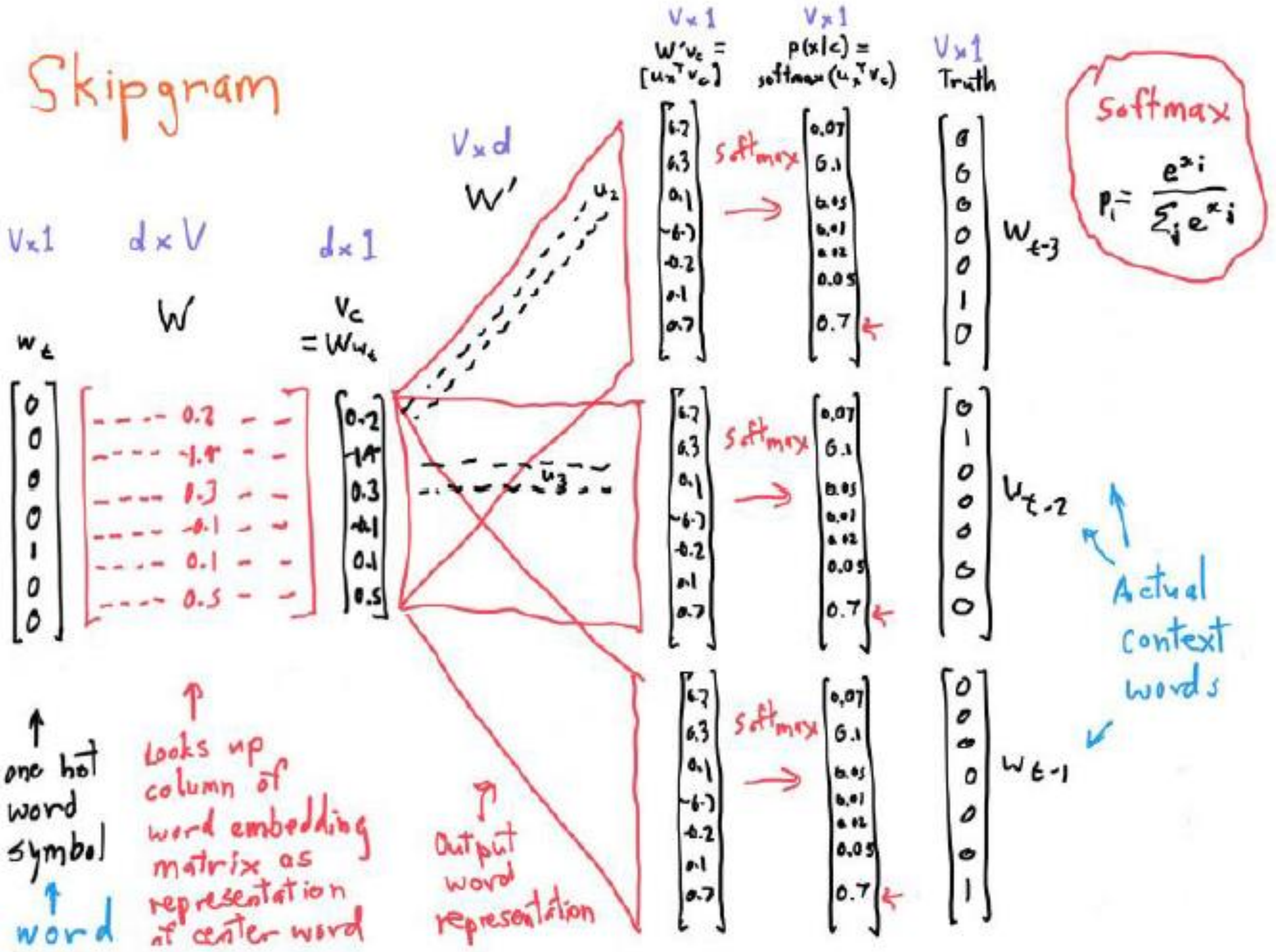
- 两组训练数据，(live, I) (live, in)
- 我们想要通过模型得出的是，由于输出词是在以输入词为中心的窗口中选择出来的，那么这些词应该跟输入词比较相近，也就是说**输入词有更大的几率得到这些输出词**
- 那么不是这些输出词的其他词，**输入词得到他们的概率就应该比较小**

Word2Vec – Skip-Gram

- V = 6，那么 live 用 onehot 向量可以表示成（0,1,0,0,0,0）
- 输入(0,1,0,0,0,0)假设输出是一个概率分布(0.3, 0.2, 0.1, 0.1, 0.2, 0.2)
- 0.3最大，对应词汇表第一个词

Word2Vec – Skip-Gram





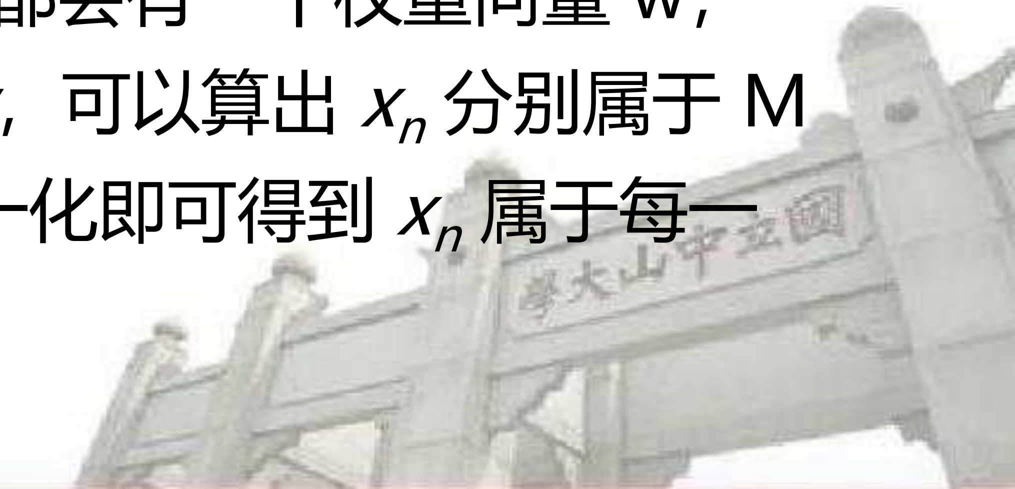
Skipgram
V×1
d×V
d×1
V×d
W
W'
w_t
$v_c = W w_t$
W'v_c = [u_x^T v_c]
p(x|c) = softmax(u_x^T v_c)
Truth
softmax
$p_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$
w_{t-3}
w_{t-2}
w_{t-1}
one hot word symbol
word
Looks up column of word embedding matrix as representation of center word
Output word representation
Actual context words

Word2Vec – Skip-Gram

- softmax 函数

$$p(y_n = i|x_n, W) = \frac{e^{W_i^T x_n}}{\sum_{j=1}^{M} e^{(W_j^T x_n)}}$$

- 假设一共有 M 个类别，那么对应每个类别都会有一个权重向量 w，上述公式的含义就是，利用 M 个类别的 *w*，可以算出 x_n 分别属于 M 个类别的概率，做一个让概率和为 1 的归一化即可得到 x_n 属于每一个类别的概率



Word2Vec – CBOW

- CBOW 和 Skip-Gram 不一样的地方是，对于一个词来说，组成的（输入词，输出词）词对只有一个了，并且**选择的这个词是输出词**
- 我们想要求解的其实就是，**依据这个词附近的这些词，有多大的概率可以得到这个词**

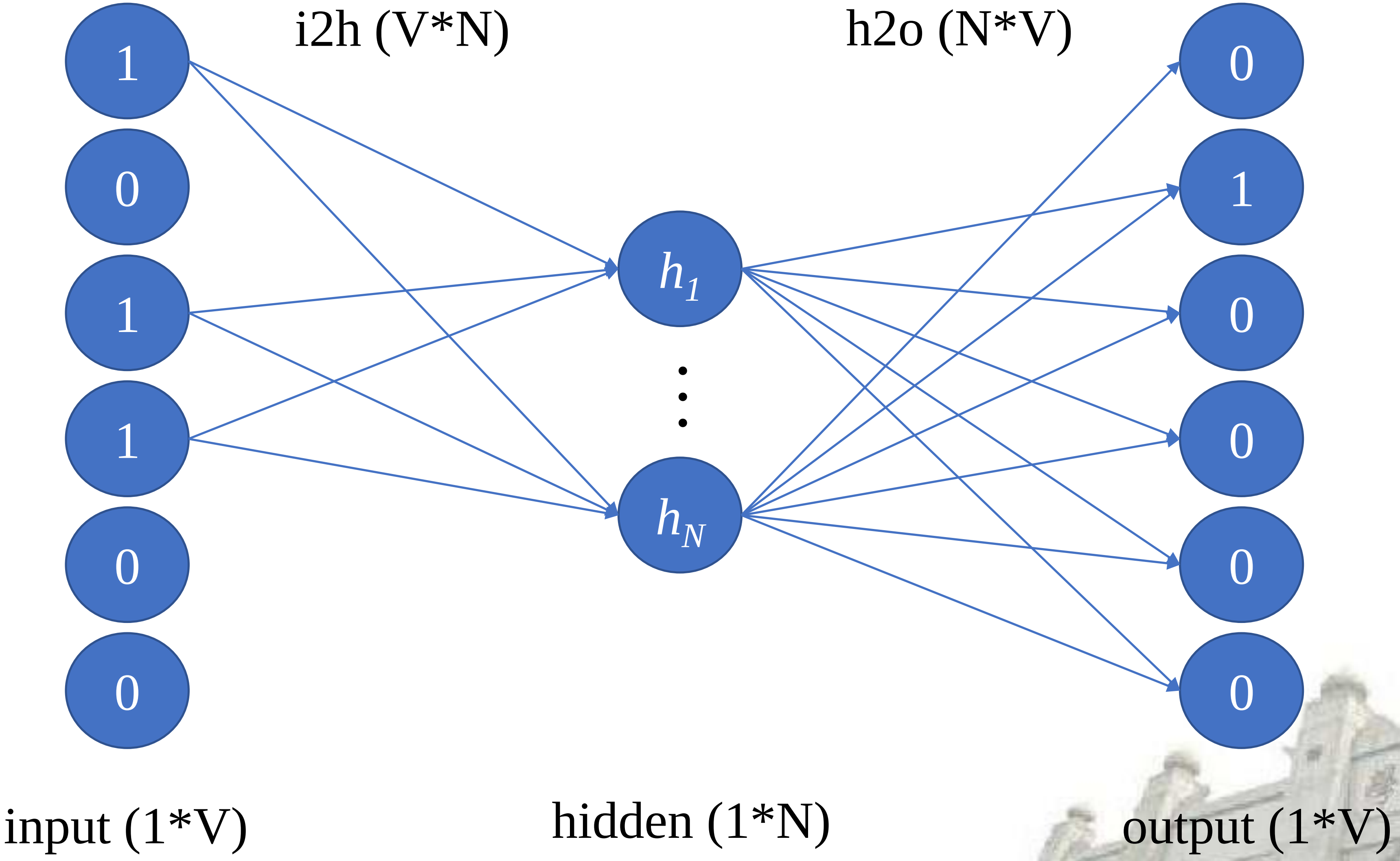
Word2Vec – CBOW

- 举例文本：I live in China and I love China. （V = 6）
- 选择一个**输出词**，假设为 live
- 定义一个 skip-window 的参数，假设为 2，这个参数代表着从这个输入词的左侧或右侧选择的单词数量
- 那么选到的词就是 I live in China

Word2Vec – CBOW

- I live in China
- 那么其输入的词就是 I in China
- 同样我们转为 onehot 向量，就是（1,0,1,1,0,0)
- 那么同样的，我们传入到神经网络里面去训练
- 标准的输出是 live，也就是（0,1,0,0,0,0)

Word2Vec – CBOW



Word2Vec – CBOW

- **把输入词的 onehot 向量当成输入向量，输出向量的第 i 维是这个输入词可以得到词汇表第 i 个单词的概率**
- 同样利用的是 softmax 函数作为输出层的激活，这部分跟 Skip-Gram是一样的

Word2Vec – 加速方法

- 负采样(Negative Sampling)
- 霍夫曼(Hierarchical Softmax)
- 感兴趣的同学可以结合源代码和网上教程自行学习
- **gensim库word2vec的使用**