0. 写在前面

1 本课程总体结构

章节		教学内容
第一章 引言 (刘均, 2)		概念与研究背景；主要任务；挑战与研究方向；相关资源
第二章：自然语言的统计特性 (刘均, 1)		Zipf定律、Heaps定律、Benford 定律。
第三章：语言模型	词袋模型 (刘均, 3)	语言模型；词袋模型（BoW）；TF-IDF。 NLU任务：情感分析、文本聚类。
	概率语言模型 (李辰, 6)	概率语言模型；n-gram 模型；最大似然估计；平滑技术。 NLU任务：分词、语义关系抽取。
	主题模型 (刘均, 6)	生成模型；主题模型的图表示；LSA、PLSA、LDA；NMF等。 NLU任务：话题检测、推荐。
	神经网络语言模型 (李辰, 6)	分布式表示；C&W、CBOW、Skip-Gram、Glove等。 NLU任务：对话、实体消歧。
第四章：机器翻译	概述 (李辰, 1)	面临的挑战；发展历程；方法类别及特点；MT评估。
	统计机器翻译 (李辰, 3)	统计MT；Noisy Channel模型；IBM模型。
	神经网络机器翻译 与大语言模型 (刘均, 4)	RNN与LSTM简介；Encoder-Decoder框架；Attention模型；大语言模型。

- 这门课由于由两门老师授课，个人感觉结构比较混乱
- 由于时间紧任务重经费无，所以笔记还是按PPT内容和以上结构展开，即使有很多不合理的地方

2 考试有关事项

1. 词表示&NN语言模型

1.1. 离散式表示

1️⃣ One-Hot编码

1. 构建方法：
   - 构建词汇表：包含所有要处理的独立词，表大小(词数量)决定了向量维度

     ```
     Vocabulary = [apple, banana, pineapple]
     ```

   - 向量分配：每个词分配一个唯一的向量，即每词对应一个维度并设为1(其余设为0)

     ```
     [1,0,0] = apple
     [0,1,0] = banana
     [0,0,1] = pineapple
     ```

2. 缺点：高维且稀疏，无法体现语义关系(例如$\cos\langle \text{apple}, \text{banana}\rangle = 0$即使这二者关系很大)

2️⃣ WordNet

1. 语义关系

关系	含义	实例
上位词↔下位词	更一般的概念↔更具体的概念	animal ↔ dog
部分类↔整体类	某物组成部分↔某物整体组合	wheel ↔ car
反义词	意义相反的词	small ↔ big
多义词	一个词具有多重含义	bank表示河岸和银行

2. WordNet概述：
   - 概念：一大型英语词汇库，将名词/动词/形容词/副词组织为一系列同义词集(如下)

     ```
     car, automobile, machine, motorcar
     ```

   - 层次：语义网络，即结点(同义词集)+边(同义词集间的语义关系)
3. 缺点：更新困难/设计时具有主观性/多义词的存在.....

1.2. 分布式表示

1️⃣ 基本概念

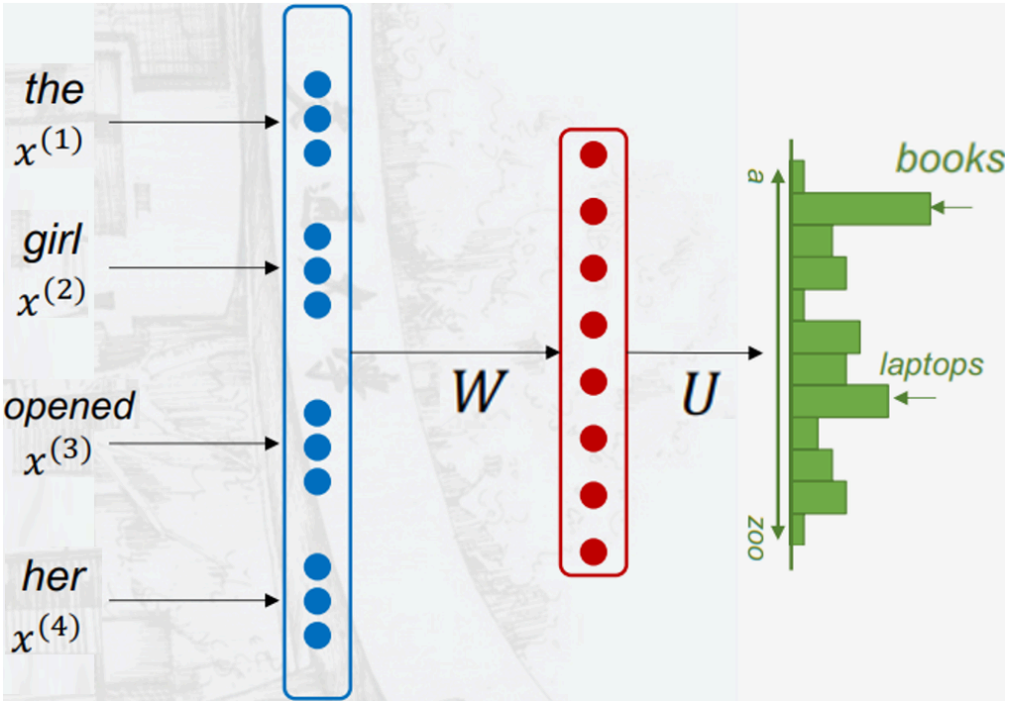
1. 目的：将词/句子$\xrightarrow{\text{编码}}$稠密低维向量
2. 核心：通过词的上下文(**词固定窗口范围内的内容**)提取词的含义，并将其含义编码在自身向量中

2️⃣ 相似性度量：余弦相似度$\text{cosine}(A,B)=\dfrac{A\times B}{\|A\|\|B\|}$

3️⃣分布式模型：基于神经网络的语言模型

原理	实例
基于预测	Word2Vec(CBOW, Skip-gram)/GloVe
基于上下文表示	BiLSTM/BERT

1.3. 神经网络语言模型的结构

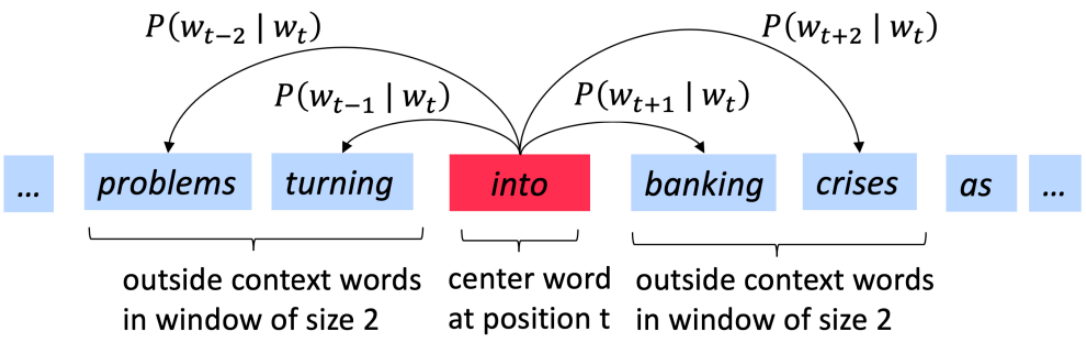


步骤	描述
词向量	通过分布式表示等，得到$\{x^{(1)}, x^{(2)}, ..., x^{(t)}\}$
词嵌入	$\{x^{(1)}, x^{(2)}, ..., x^{(t)}\} \xrightarrow{\text{embedding}} \{e^{(1)}, e^{(2)}, e^{(3)}, ...\}$
隐藏层	获得$h=f(We+b_1)$
输出层	获得$\hat{y}=\text{Softmax}\,(Uh+b_2) \in \mathbb{R}^{\lvert V\rvert}$

2. Word2vec模型

2.1. 模型概述

1️⃣基本思想：通过词的上下文来学习其语义，而每个单一词向量无具体含义



1. 构建词汇表：大小固定，其中每个词用词向量表示
2. 文本的表示：每个词的位置t被视为中心词c，词t所在窗口内其它词视为上下文o
3. 优化的途径：不断计算c/o之间的相似度$P(o|c)$或$P(c|o)$，调整c/o词向量使概率最大化

2️⃣ 目标函数：给定待优化参数集θ和上下文窗口$[-m, m]$

1. 最大似然：$L(\theta)=\prod_{t=1}^{T}\prod_{-m\leq j\leq m}P\left(w_{t+j}|w_t;\theta\right)$

公式	含义
$\prod_{-m\leq j\leq m}P\left(w_{t+j}\mid w_t;\theta\right)$	以t为中心位置，生成$[-m, m]$范围内所有上下文的概率
$\prod_{t=1}^{T}\prod_{-m\leq j\leq m}P\left(w_{t+j}\mid w_t;\theta\right)$	在句子中滑动t，将每个t位置的上下文生成概率累乘

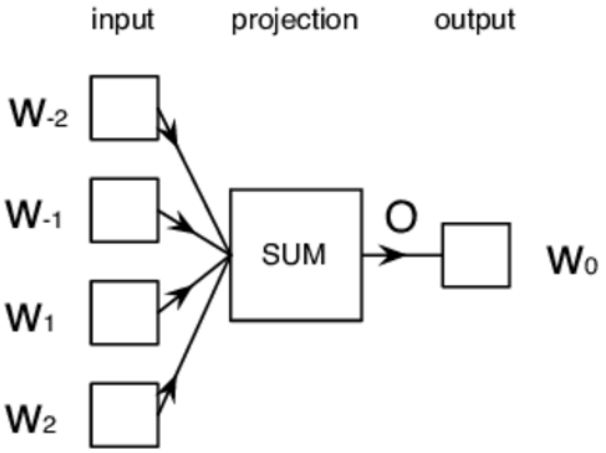
2. 目标函数：$J(\theta)=-\frac{1}{T}\log L(\theta)=-\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m}\log P\left(w_{t+j}|w_t;\theta\right)$
   - 含义：对模型预测错误的惩罚，需要使其最小之
3. 预测函数：即Softmax函数$P\left(w_{t+j}|w_t\right)=\frac{\exp\left(u_{w_{t+j}}^{T}v_{w_t}\right)}{\sum_{w\in V}\exp\left(u_{w}^{T}v_{w_t}\right)}$
   - 含义：计算词w_{t+j}/w_t间的相似度，并将除以w_t**在整个词汇表上计算概率分布**以归一化

3️⃣ 模型训练：

1. 目标：最小化目标函数$J(\theta)$，其中参数向量θ包括所有词的中心向量+上下词向量
2. 优化：依$\theta_j^{\text{new}}=\theta_j^{\text{old}}-\alpha\frac{\partial J(\theta)}{\partial\theta_j}$梯度下降，或者随机梯度下降

2.2. 两种训练方法

1️⃣ CBOW：根据上下文预测中心词



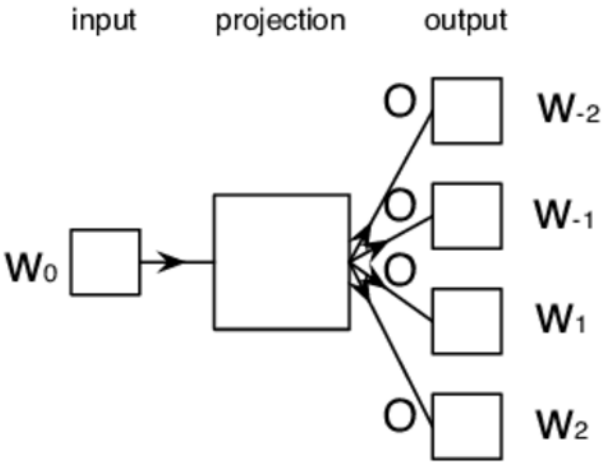
1. 模型流程：得到中心词的概率分布$\hat{y}$→优化损失函数最大化$p(c|o)$

流程	描述	形式化
输入	上下文共$2m$个词的One-Hot向量	$\{x^{c-m},...,x^{c-1},x^{c+1},...,x^{c+m}\}\in\mathbb{R}^{\|V\|}$
词嵌入	使用嵌入矩阵W将高维编码转为低维嵌入	$v_i=(x_i\times W)\in\mathbb{R}^N$
池化	计算所有上下文嵌入向量平均(综合向量)	$\hat{v}=\frac{v_{c-m}+\cdots+v_{c+m}}{2m}\in\mathbb{R}^N$
打分	将综合向量$\hat{v}$映射为得分向量z	$z=(\hat{v}\times W^T)\in\mathbb{R}^{\|V\|}$
概率化	用Softmax将打分z映射为概率分布$\hat{y}$	$\hat{y}=\text{Softmax}(z)$

2. 损失函数
    - 目标/似然函数：最小化$P\left(x_c|x_{c-m},...,x_{c+m}\right)$
    - 损失函数：

$$J=-\log P\left(x_c|x_{c-m},...,x_{c+m}\right)\xrightarrow[\text{Softmax}]{\text{展开/池化}}-x_c^T\times\hat{v}+\log\sum_{j=1}^{|V|}\exp\left(x_j^T\times\hat{v}\right)$$

2️⃣ Skip-gram(SG)：根据中心词预测上下文



1. 模型流程：得到上下文的概率分布$\hat{y}$→优化损失函数最大化$p(o|c)$

流程	描述	形式化
输入	中心词的One-Hot向量	$x^c\in\mathbb{R}^{\|V\|}$
词嵌入	使用嵌入矩阵W将高维编码转为低维嵌入	$v_c=(x_c\times W)\in\mathbb{R}^N$
打分	将综合向量$\hat{v}$映射为得分向量z	$z=(v_c\times W^T)\in\mathbb{R}^{\|V\|}$
概率化	用Softmax将打分z映射为概率分布$\hat{y}$	$\hat{y}=\text{Softmax}(z)$

2. 损失函数
    - 目标/似然函数：最小化$P\left(x_{c-m},...,x_{c+m}|x_c\right)$

- 损失函数：

$$J=-\log P \xrightarrow[\text{Softmax}]{\text{展开}} -\left(\sum_{j=0,j\neq m}^{2m} v_{c-m+j}^{T}\times v_c+2m\log\sum_{k=1}^{|V|}\exp\left(v_k^T\times v_c\right)\right)$$

2.3. 两种训练优化

1 层次Softmax

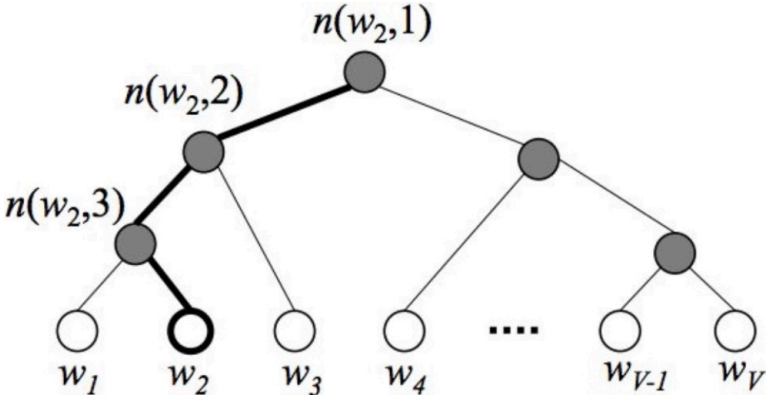
1. 二叉树构造：根节点表示整个词汇表V，不断二分(例如用Huffman树)到叶节点表示单个词汇w_i
2. 概率计算：$P\left(w=w_o\right)=\prod_{j=1}^{L(w)-1}\sigma\left(\left[\left[n(w,j+1)=\text{ch}(n(w,j))\right]\right]\times v_{n(w,j)}^{T}h\right)$
   - 参数含义：

参数	含义
$L(w)$	从根节点到目标单词w，所经过的节点数
$n(w,j)$	到w的路径中的第j个结点，其子节点为$n(w,j+1)$
σ	Sigmoid激活函数

   - 随机游走：

路径中第$j+1$个结点为第j个结点的	$\left[\left[n(w,j+1)=\text{ch}(n(w,j))\right]\right]\times v_{n(w,j)}^{T}h$含义
左子节点	乘以 $\sigma\left(v_{n(w,j)}^{T}h\right)$
右子节点	乘以 $\sigma\left(-v_{n(w,j)}^{T}h\right)$

3. 示例：



   - 游走路径：$P\left(n\left(w_2,1\right),\text{left}\right)\times P\left(n\left(w_2,2\right),\text{left}\right)\times P\left(n\left(w_2,3\right),\text{right}\right)$
   - 概率展开：$P\left(w=w_2\right)=\sigma\left(v_{n(w_2,1)}^{T}h\right)\times\sigma\left(v_{n(w_2,2)}^{T}h\right)\times\sigma\left(-v_{n(w_2,3)}^{T}h\right)$

2 负采样

1. 核心思想：
   - 不遍历整个词汇表V
   - 只遍历一个与词汇频率顺序匹配的噪声概率分布$p_n(w)$中采样的几个Negative例子

2. 模型定义：

实例	含义	优化目标
正例	实际出现词对(w_c, w_o)	最大化(w_c, w_o)出现概率 $J_{\text{pos}} = \log \mathrm{P}(\mathrm{D}=1\|w, c)$
负例	随机采样词对(w_c, w_j)	最小化(w_c, w_o)不出现概率 $J_{\text{pos}} = \log \mathrm{P}(\mathrm{D}=0\|w, c)$

3. 损失函数：

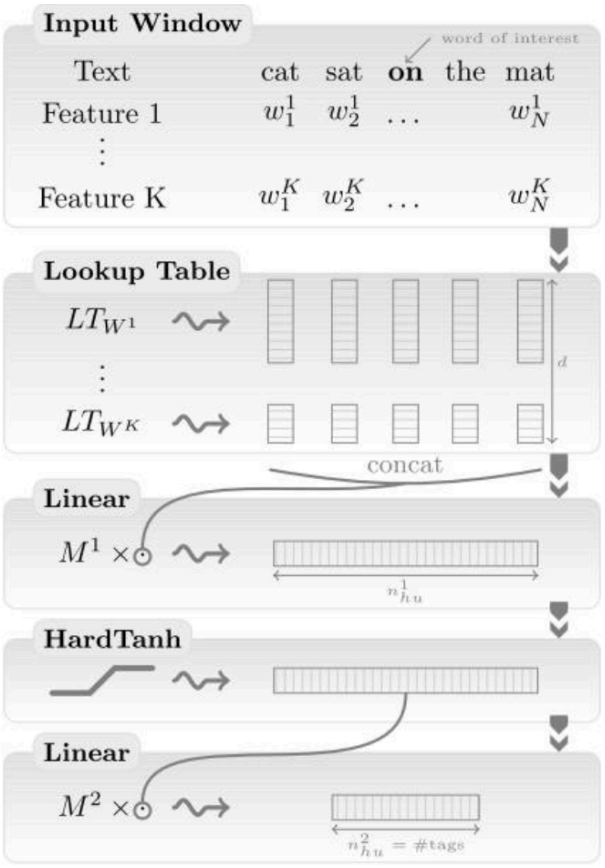
训练方式	损失函数
CBOW	$J = -\log \sigma\left(v_c^T \times \hat{v}\right) - \sum_{k=1}^{K} \log \sigma\left(-\tilde{v}_k^T \times \hat{v}\right)$
Skip-Gram	$J = -\log \sigma\left(v_{c-m+j}^T \times v_c\right) - \sum_{k=1}^{K} \log \sigma\left(-\tilde{v}_k^T \times v_c\right)$

3. 其它模型

3.1. C&W模型

1 模型目标：抛弃传统语言模型对条件概率的计算，转而通过打分函数衡量一段词序的合理性

2 层次结构：



结构	描述
输入层	是一个大小为n的连续窗口，输入序列包含n个词向量
查找表层	对每个词进行查找→获得每个词的稠密词向量
卷积层	对输入序列的每个窗口应用卷积操作，提取局部特征
最大池化层	从卷积层提取最显著的特征，输出固定维度向量
分类层	对生成的向量进行打分，高分者视为语言上自然的

3️⃣模型训练：

1. 输入：对于一个窗口内的连续词$[w_{i-n},...,w_i,...,w_{i+n}]$

序列类型	操作	性质
正序列	保留$[w_{i-n},...,w_i,...,w_{i+n}]$	在语言上自然
负序列	将窗口中心处词w_i换成w_j，即$[w_{i-n},...,w_j,...,w_{i+n}]$	在语言上不自然

2. 目标：最大化正序列$[w_{i-n},...,w_i,...,w_{i+n}]$得分+最小化负序列$[w_{i-n},...,w_j,...,w_{i+n}]$得分
3. 损失：$\mathcal{L}=\max\left(0,1-s\left(w_{\text{true}}\right)+s\left(w_{\text{false}}\right)\right)$，正列得分≫负序列$\overset{\text{等价于}}{\Longleftrightarrow}\mathcal{L}\to 0$

3.2. GloVe模型

1️⃣核心思想

1. 目标：学习词的一种分布式表示，使词向量可以捕捉词语的语义关系
2. 假设：语义可从其共现信息中推断，例如ice/cold共现比ice/steam频率高$\overset{\text{认为}}{\longrightarrow}$反映了语义关系

2️⃣模型要点

1. 共现矩阵$\mathbf{X}$：
   - 无权值情况：$\mathbf{X}_{ij}$表示词i与词j在某滑动窗口共现的和，每共现一次增加1

```
the cat sat on the mat (窗口大小=3)
[the cat sat] -> X_the_sat=1
[cat sat on]  -> X_the_sat=1+0
[sat on the]  -> X_the_sat=1+0+1
[on the mat]  -> X_the_sat=1+0+1+0
```

   - 有权值情况：在原有基础上，每共现一次增加一权值$\text{decay}=\frac{1}{d}$(其中d为词ij在窗口中的距离)

```
the cat sat on the mat (窗口大小=3)
[the cat sat] -> X_the_sat=1/2
[cat sat on]  -> X_the_sat=1/2+0
[sat on the]  -> X_the_sat=1/2+0+1/2
[on the mat]  -> X_the_sat=1/2+0+1/2+0
```

2. 近似关系：$\mathbf{X}_{ij} \xrightarrow{\text{变换+分解}} \log\left(X_{ij}\right)=w_i^T\tilde{w}_j+b_i+\tilde{b}_j \rightarrow$用$w_i^T\tilde{w}_j$表示词$ij$的近似关系+偏置修正
3. 损失函数：$J=\sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T\tilde{w}_j+b_i+\tilde{b}_j-\log\left(X_{ij}\right)\right)^2$

项	含义
$w_i^T\tilde{w}_j+b_i+\tilde{b}_j$	所预测的词ij共现次数
$\log(X_{ij})$	实际的词ij共现次数
$f(X_{ij})$	加权函数(较少共现词对模型的影像)

   - 对于共现次数阈值$x_{\max}$有$f\left(X_{ij}\right)=\begin{cases}\left(\frac{X_{ij}}{x_{\max}}\right)^{\alpha}, X_{ij}<x_{\max}\\ 1, \text{otherwise}\end{cases}$