

Lettuce2

本节内容为词向量、Word Senses、Neural Network Classifiers

Word2Vec

Word2Vec 算法族

- **Word2Vec** 包含了两种模型：
 - **Continuous Bag-of-Words (CBOW)** : **CBOW** 是根据中心词周围的上下文单词来预测该词的词向量
 - **Skip-Gram** : **Skip-Gram** 是根据中心词预测周围上下文的词的概率分布
- 同时, **Word2Vec** 也包含了两种算法：
 - **Negative Sampling** 通过抽取负样本来定义目标
 - **Hierarchical Softmax** 通过使用一个有效的树结构来计算所有词的概率来定义目标

共现矩阵

概念

为构建共现矩阵 X , 要有 2 个条件: 窗口与完整文档。

- 窗口: 类似于 **word2vec**, 在每个单词周围使用窗口来捕获一些句法和语义信息
- 文档: 若单词 w_i 出现在文档 d_j 中, 则共现矩阵元素 X_{ij} 加 1

一个共现矩阵的例子如下图:

如果假设窗口长度为 1（通常是 5 - 10），共现矩阵是对称的（与上下文的左或右无关），语料库为如下三句话：I like deep learning. I like NLP. I enjoy flying.

那么共现矩阵如下(见图 2.7):

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

图 2.7: 简单的共现矩阵

共现向量

- 事实上，上述矩阵的一列，即一个向量即可表示一个词。然而，这种表示方法下词向量的维度过高，需要降维
- 一个可行的对共现矩阵进行降维的方法是对该矩阵进行 **SVD** 分解

Glove模型

推导

- Glove** 模型的创新之处在于，其发现了共现概率的比率可以用于对词向量的编码，示例如下图：

	定性推理				定量计算			
	x=solid	x=gas	x=water	x=random	x=solid	x=gas	x=water	x=fashion
P(x ice)	large	small	large	small	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
P(x steam)	small	large	large	small	2.2×10^{-4}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{P(x ice)}{P(x steam)}$	large	small	~ 1	~ 1	8.9	8.5×10^{-2}	1.36	0.96

图 2.12: 目标词 ice 和 steam 与从 60 亿个符记语料库中选择的上下文词的共现概率

图注：只有在比率中才会有来自 water 和 fashion 等非歧视性词的噪音抵消，因此大值（远大于 1）与特定于 ice 的属性相关性很好，而小值（远小于 1）与特定于 steam 的属性相关性很好。

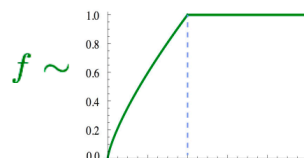
- 基于此，**Glove** 模型的目标函数如下图：



$$w_i \cdot w_j = \log P(i|j)$$

$$J = \sum_{i,j=1}^V f(X_{ij}) (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

- Fast training
- Scalable to huge corpora
- Good performance even with small corpus and small vectors



评估

- NLP中的评估方式大体分为两类：
 - 内在评估：对中间子任务的评估，速度快，但不能确定是否在实际应用中有效
 - 外在评估：对实际任务的评估，耗时较长
- 对 **Glove** 模型的内在评估可以通过词向量类比、可视化等方式进行
- 对 **Glove** 模型的外在评估使用了 GloVe 模型使用基于 **CRF** 的模型在 NER 任务上的结果和其他模型对比

神经网络分类器

Softmax函数的局限性

逻辑回归主要工具是 **Softmax** 分类器，但是 **Softmax** 分类器只给出线性决策边界，其本身并不是很强大，而且其功能还可能非常有限，特别是在问题复杂时无能为力。例如，如图2.24(a)所示，**Softmax** 分类器把箭头所指的两个点简单地、线性地、错误地分为负类。其实，如果用非线性分类器(例如下面即将介绍的 **Sigmoid** 激活函数)，则可正确地把它们分为正类，见图 2.24(b)。

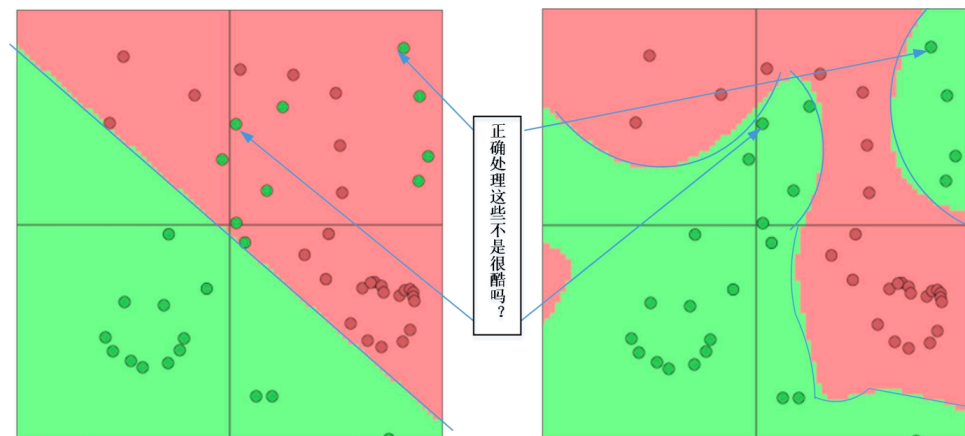


图 2.24: 相同的数据在线性分类和非线性分类下的不同结果

Sigmoid函数

我们为什么需要非线性 **Sigmoid** 激活函数？因为它可以逼近任意复杂函数。例如：如果没有非线性，深度神经网络只能做线性变换，额外的层也只可以编译成一个单一的线性变换，如果有了非线性和更多的层，则可以逼近更复杂的任意函数，见图 2.28：

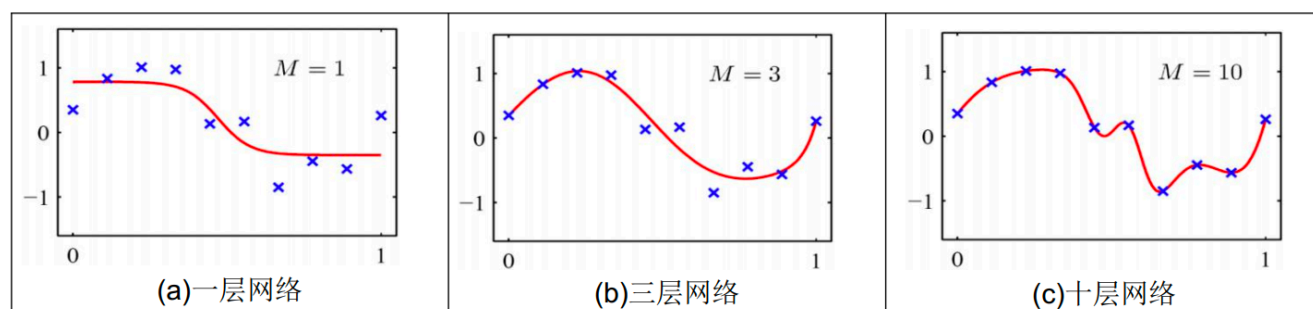


图 2.28: 非线性的多层神经网络，可以逼近更复杂函数