Chapter 1

word2vec 的高速化

1.1 word2vec 的改进 ①

1.1.1 Embedding 层

如 Figure 1.1 所示，如果语料库的词汇量有 100 万个，则单词的 one-hot 表示的维数也会是 100 万，我们需要计算这个巨大向量和权重矩阵的乘积。但是，Figure 1.1 中所做的无非是将矩阵的某个特定的行取出来。因此，直觉上将单词转化为 one-hot 向量的处理和 MatMul 层中的矩阵乘法似乎没有必要。现在，我们创建一个从权重参数中抽取“单词 ID 对应行（向量）”的层，这里我们称之为 Embedding 层。

> 在自然语言处理领域，单词的密集向量表示称为词嵌入（word embedding）或者单词的分布式表示（distributed representation）。过去，将基于计数的方法获得的单词向量称为 distributional representation，将使用神经网络的基于推理的方法获得的单词向量称为 distributed representation。不过，中文里二者都译为“分布式表示”。

1.2 word2vec 的改进 ②

word2vec 的另一个瓶颈在于中间层之后的处理，即矩阵乘积和 Softmax 层的计算。本节的目标就是解决这个瓶颈。这里，我们将采用名为负采样（negative sampling）的方法作为解决方案。

通过引入 Embedding 层，节省了输入层中不必要的计算。剩下的问题就是中间层之后的处理。此时，在以下两个地方需要很多计算时间：

1. 中间层的神经元和权重矩阵（$\boldsymbol{W}_{out}$）的乘积
2. Softmax 层的计算

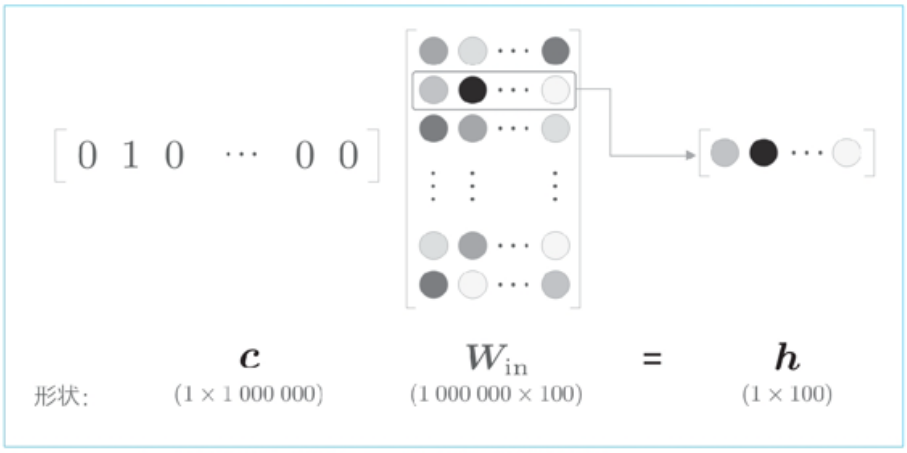

Figure 1.1: one-hot 表示的上下文和 MatMul 层的权重的乘积

1.2.1 从多分类到二分类

下负采样这个方法的关键思想在于二分类（binary classification），更准确地说，是用二分类拟合多分类（multiclass classification），这是理解负采样的重点。

1.2.2 负采样的采样方法

关于这一点，基于语料库的统计数据进行采样的方法比随机抽样要好。具体来说，就是让语料库中经常出现的单词容易被抽到，让语料库中不经常出现的单词难以被抽到。

word2vec 中提出的负采样 word2vec 中提出的负采样：

$$P^{'}(w_i) = \frac{P(w_i)^{0.75}}{\sum_j^n P(w_j)^{0.75}} \tag{1.1}$$

这样处理是防止低频单词被忽略，更准确地说通过取 0.75 次方，低频单词的概率将稍微变高。此外，0.75 这个值并没有什么理论依据，也可以设置成 0.75 以外的值。