一、摘要:

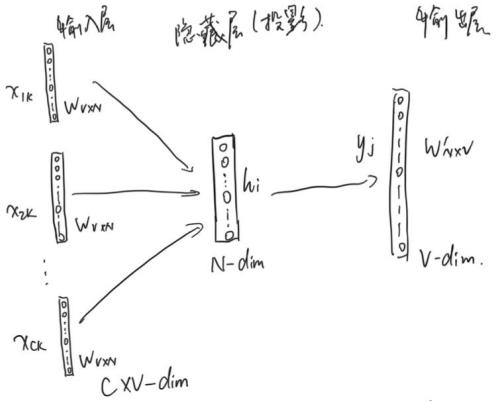
当前自然语言处理领域面临一些问题，许多 NLP 系统把单词当作原子单位，以索引表示，没有单词相似性概念，虽然这种简单方式有其优势，但在自动语音识别、机器翻译等任务中，由于数据量的限制，简单技术已达瓶颈，单纯扩大基础技术规模难以取得显著进展。随着机器学习技术发展，复杂模型可在更大数据集上训练且表现更优，分布式词表示成为趋势，如神经网络语言模型优于 N-gram 模型。然而，之前的词向量学习架构存在局限，无法在大规模数据上训练，词向量维度有限，计算成本较高。在此背景下，论文提出了两种模型架构用于计算词向量，在词向量学习技术上取得重要进展，连续词袋模型（CBOW）和连续跳字模型（Skip-gram），去除复杂的非线性隐藏层，降低计算复杂度，能在大规模数据集上高效训练高质量词向量。CBOW 基于上下文预测当前词，Skip-gram 则依据当前词预测周围词，二者从不同角度利用文本信息学习词向量。同时，论文设计了包含多种语义和句法问题的综合测试集，全面评估词向量质量。实验表明，利用 DistBelief 分布式框架，结合随机梯度下降和 Adagrad 自适应学习率方法，实现模型大规模并行训练。训练效率大幅提升。该模型可在一天内从 16 亿词数据集学习高质量词向量，比以往模型训练效率更高，且能处理更大规模数据和词汇表，同时测试集上达到了最先进的性能。

二、背景与动机:

在自然语言处理领域，让计算机理解和处理人类语言一直是核心目标。但自然语言具有高度复杂性和歧义性，计算机难以直接处理。为解决这一问题，需将自然语言转化为计算机能理解的数学形式，词向量表示应运而生，其在自然语言处理中至关重要。 传统上，自然语言处理采用基于规则的方法，语言学家总结语法和语义规则编写程序，让计算机依据规则处理语言。但语言丰富多变，规则难以涵盖所有情况，面对复杂句子和新词汇，基于规则的系统表现不佳，且人工制定规则耗时费力，难以适应大规模文本处理需求。 随着数据量增长和机器学习技术发展，基于统计的自然语言处理方法兴起，统计语言模型通过对大规模语料库分析，计算词序列出现概率来处理自然语言。N-gram 模型是经典的统计语言模型，它基于马尔可夫假设，认为一个词出现的概率只与前面(n-1)个词相关，通过统计语料库中 N-gram 序列频率来估计句子概率。但 N-gram 模型存在局限性，参数数量随 n 增大呈指数增长，计算复杂度高，且对罕见词串统计不准确，会导致数据稀疏问题，影响模型性能。 神经概率语言模型的出现为解决这些问题带来曙光，它使用神经网络学习词向量和语言模型，能捕捉词之间复杂语义和句法关系，克服 N-gram 模型的数据稀疏问题。在神经概率语言模型中，词向量作为重要组成部分，将词映射到低维连续向量空间，相似语义的词在空间中距离相近。如在语料库中，“狗”和“猫”常出现在相似上下文，其词向量也相近，这使模型能更好处理语义相似的词，提升自然语言处理任务效果。 尽管神经概率语言模型取得进展，但训练效率和大规模数据处理能力仍有待提高。许多神经语言模型训练复杂，计算成本高，难以在大规模语料库上快速训练高质量词向量。在实际应用中，如搜索引擎、机器翻译、智能客服等，需要处理海量文本数据，对模型训练速度和效果有更高要求。在此背景下，2013 年 Mikolov 等人提出 Word2Vec，旨在高效生成高质量词向量，满足大规模自然语言处理需求。Word2Vec 通过简单而有效的神经网络架构，利用大量文本数据无监督学习词向量，能快速处理大规模语料库，学习到的词向量能捕捉丰富语义和句法信息，为后续自然语言处理任务奠定良好基础，推动自然语言处理技术在实际应用中的发展。

三、方法:

CBOW Model:



$$h = \frac{1}{C} W^T (x_1 + x_2 + \cdots x_C)$$
$$= \frac{1}{C} (V_{w_1} + V_{w_2} + \cdots + V_{w_C})^T.$$

$$L = -\max \log p(w \mid \text{Context}(w))$$
$$= -\max \log (y_j^*)$$
$$= -\log p(w_o \mid w_{1,1}, w_{1,2}, \cdots, w_{1,C})$$
$$= -u_{j^*} + \log \sum_{j'=1}^{V} \exp(u_{j'})$$
$$= -V'^T_{w_o} \cdot h + \log \sum_{j=1}^{V} \exp(V'^T_{w_j} \cdot h).$$

↓ 为 input 1-C下 output的概率.

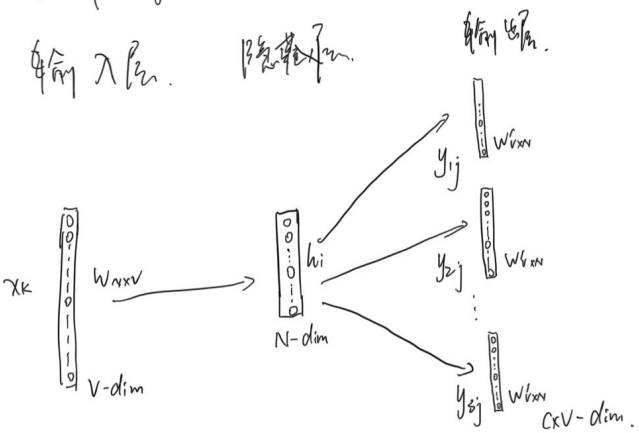
输出层经过 Softmax 归一化.

设 u_{j^*} 代表归一化前结果.

则此 $\max \log (y_j^*) = \max \log \left(\frac{\exp(u_{j^*})}{\sum \exp(u_k)} \right)$.

$$= \max u_{j^*} - \log \sum_{k=1}^{V} \exp(u_k).$$

Skip-gram Model:



Skip-gram 其实就是 CBOW 的逆向过程.

给定一个词去预测为个词当成的概率

从输入层到隐藏层的过程是由单个词的 CBOW 相同.

$h = W^T \cdot X = V^T_{W_I}$.

现在介绍从 $h \to o$ 的过程.

$p(w_{cj} = w_{O,c} | w_I) = y_{cj}$

① 共享权重矩阵.

$u_{cj} = u_j = v'^T_{w_j} \cdot h \quad , \quad c = 1, 2, 3 \cdots C$

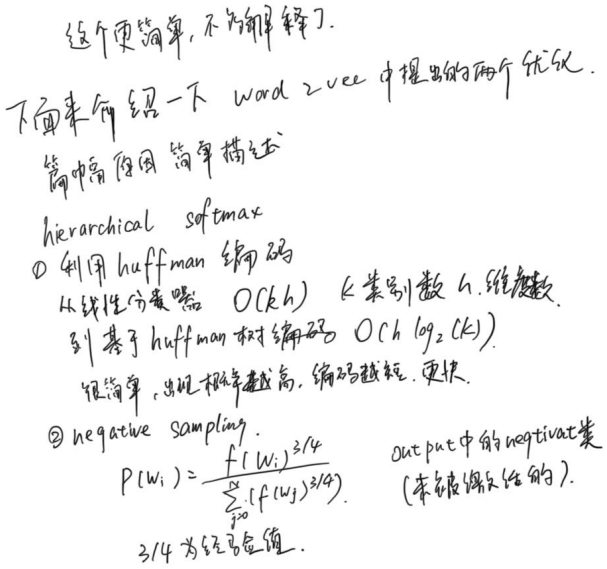
通过全重复计算了, 比较直接的想法.

② Loss 函数

$L = -\log p(w_{O,1}, w_{O,2}, \cdots, w_{O,C} | w_I)$ ← 在 Input 下 Output 1-C 的概率.

$$= -\log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$

$$= -\sum_{c=1}^{C} u_{j_c^*} + C \cdot \log \sum_{j'=1}^{V} \exp(u_{j'})$$



CBOW（连续词袋模型）和 Skip-gram 是 Word2Vec 的两种核心架构。CBOW 的目标是根据上下文词预测中心词。具体来说，它将上下文词的 One-Hot 编码映射到共享的嵌入矩阵，再对这些向量求平均，得到一个综合表示。这个表示随后输入到分类器，预测中心词的概率分布。CBOW 计算简单，适用于高频词较多的场景，训练速度较快。

Skip-gram 则采用相反的方式。它以中心词为输入，预测周围的上下文词。模型先将中心词的 One-Hot 编码转换为低维向量，然后分别计算每个上下文词的概率分布。由于需要针对多个位置进行预测，计算量比 CBOW 更大。但 Skip-gram 在低频词的学习上更有效，能捕捉更精细的语义关系。例如，在句子“猫追逐老鼠”中，它可以通过“猫”预测“追逐”和“老鼠”，强化词语之间的关联。

Word2Vec 采用了两种优化方法，以提升训练效率。层次 Softmax 通过哈夫曼树构造词汇的层级结构，每个词对应一条唯一路径。计算时，只需沿路径更新节点权重，使时间复杂度从 O(V) 降到 O(log V)。这种方法对高频词尤为高效，因为它们的路径更短。负采样则通过随机选择少量负样本（非上下文词）进行训练，而非计算整个词汇表。每次训练时，模型只需更新正样本和少数负样本的权重，极大降低了计算成本，同时保持词向量质量。例如，在预测“猫”的上下文时，负采样可能选择“太阳”或“汽车”作为干扰项，迫使模型学习更具区分性的特征。

这些优化方法使 Word2Vec 能够在海量语料上高效学习高质量的词向量。CBOW 适合处理高频词和小窗口数据，而 Skip-gram 在低频词建模上更具优势。优化技术的引入不仅加快了训练速度，还提升了模型在大规模场景中的适用性，为自然语言处理的后续发展奠定了基础。

四、实验与结果:

在实验设置方面，论文使用 Google News 语料库来训练词向量，并将词汇表大小限制为 100 万个最频繁的单词。为了估计最佳的模型架构，先在训练数据的子集上对模型进行

评估，此时词汇表限制为最频繁的 30k 单词。训练过程采用随机梯度下降和反向传播的方法，训练轮数通常在 3－50 之间，常见选择为 3 轮，起始学习率设为 0.025，并让其线性递减，在最后一轮训练结束时接近零。此外，为了在大规模数据集上训练模型，还在 DistBelief 分布式框架上实现了多种模型，使用小批量异步梯度下降和 Adagrad 自适应学习率过程，训练时通常使用 50－100 个模型副本。 论文所使用的数据集主要是 Google News 语料库，该语料库包含约 60 亿个标记。在比较不同模型架构时，还使用了几个 LDC 语料库，其包含 3.2 亿字，词汇表大小为 8.2 万 。在与公开可用词向量进行比较时，涉及到多种不同规模和来源的训练数据，例如 Collobert-Weston NNLM 使用 660M 词进行训练，Turian NNLM 使用 37M 词等。 评估方法上，论文定义了一个综合测试集，该测试集包含 5 种语义问题和 9 种句法问题，一共有 8869 个语义问题和 10675 个句法问题。评估时，通过对词向量执行简单的代数运算来回答问题，比如计算“vector("biggest") - vector("big") + vector("small")” ，然后在向量空间中寻找与计算结果最接近的词作为答案。只有当找到的这个词与问题中的正确词完全一致时，才认为该问题被正确回答，同义词被视为错误答案。最终以所有问题类型的整体准确率，以及语义、句法问题各自的准确率作为评估指标。 主要实验结果如下：在比较不同模型架构时，以相同训练数据和 640 维词向量为条件，RNNLM 的词向量在句法问题上表现较好；NNLM 向量比 RNNLM 表现更优；CBOW 架构在句法任务上比 NNLM 表现更好，在语义任务上二者表现相近；Skip－gram 架构在句法任务上略逊于 CBOW，但在语义任务上比其他模型表现好得多。将论文中的模型与公开可用词向量比较，CBOW 和 Skip-gram 模型在不同维度和训练数据规模下，整体准确率优于部分公开模型。例如，Skip－gram 模型在训练词数为 783M、向量维度为 300 时，总准确率达到 53.3% 。关于训练轮数和数据量的影响，实验发现增加训练数据量和向量维度可提高准确率，但存在边际效应。同时，使用双倍数据训练一轮比在相同数据上迭代三轮效果相当或更好，还能有额外的小幅度加速。例如，1 epoch 的 Skip－gram 模型在训练词数为 16 亿、向量维度为 300 时，总准确率达到 53.8%，超过了 3 epoch 的 CBOW 模型在训练词数为 783M、向量维度为 300 时的总准确率。在利用 DistBelief 分布式框架进行大规模并行训练的实验中，不同模型在大规模训练下性能表现有所差异，且由于分布式框架的开销，CBOW 模型和 Skip－gram 模型的 CPU 使用率比单机实现时更接近。在微软句子完成挑战任务中，Skip－gram 模型单独使用时表现不如 LSA 相似性，但与 RNNLMs 结合后，取得了新的最优结果，准确率达到 58.9% 。

五、讨论：

Word2Vec 以其高效性和泛化能力，在自然语言处理领域占据重要地位。它采用 CBOW 和 Skip-gram 这两种轻量级架构，能够在大规模语料库（如数十亿单词）上快速训练词向量。由于没有复杂的非线性隐藏层，计算成本显著降低。这种方法不仅高效，还能学习词语的语义和句法关系。例如，“国王 - 男人 + 女人 = 女王” 这一类比关系，就体现了它的语义捕捉能力。因此，在文本分类、信息检索和机器翻译等任务中，它提供了强大的词向量基础。

然而，Word2Vec 也有明显局限。首先，它的词向量是静态的，无法根据不同上下文动态调整。例如，“苹果” 可能指水果，也可能指公司，但 Word2Vec 只能给它一个固定的向量。其次，它主要依赖局部上下文信息，难以建模长距离依赖关系。在需要理解复杂文本结构的任务中，它的表现会受限。此外，它对低频词的表示能力较弱，容易忽略罕见但重要的词汇。由于缺乏多义词的显式区分，它在语义歧义问题上也显得无力。

尽管如此，Word2Vec 依然在许多场景中表现出色。对于小规模文本分类、简单问答系统或轻量级推荐引擎，它提供了一种快速有效的词向量解决方案。特别是在计算资源受限的环境下，它的高效性使其成为工业界的首选工具。然而，在更复杂的任务中，比如阅读理解和语义角色标注，它的静态特性难以满足需求。因此，近年来，BERT、GPT 等基于 Transformer 的模型逐渐取代 Word2Vec，成为更强大的选择。

六、个人见解：

Word2Vec 的成功证明了分布式词向量的强大潜力，但它仍然存在改进空间。未来，动态上下文建模可能是重要方向。例如，ELMo 和 BERT 通过 Transformer 结构建模词的上下文依赖，使语义表示更加灵活。这种方法能够有效解决词义歧义问题，增强模型的理解能力。此外，外部知识的引入也值得关注。比如，结合 ConceptNet 这样的知识图谱，模型可以获得常识推理能力，更精准地区分 “bank” 这种多义词的语境含义。

在模型优化上，轻量化架构提供了新的思路。FastText 通过子词信息改进低频词的表示，使模型在小样本情况下仍能保持较好的效果。这对于低资源语言处理尤其重要。同时，优化分布式训练框架也是未来发展方向。更高效的并行计算方法，如 DistBelief 的升级版本，可以支持在万亿级别语料上训练超大规模词向量，进一步提升语义表示的精度。

从实际应用来看，Word2Vec 在情感分析、命名实体识别等传统 NLP 任务中依然具有价值。但由于其静态特性，它在对话系统、机器翻译等动态任务中的表现受到限制。未来的词向量技术可能需要融合多模态数据，例如结合图像、语音信息，以增强模型的泛化能力。同时，强化学习或许可以帮助词向量更好地适应特定任务需求，提升模型在复杂场景下的表现。总的来说，词向量技术仍需在效率、灵活性和知识融合方面持续演进，以应对自然语言处理领域不断增长的挑战。