基于分布式神经网络的语言模型训练方法及其系统

专利申请文件确认

尊敬的客户：您好！

关于贵方委托我司代理的“基于分布式神经网络的语言模型训练方法及其系统”发明专利申请，我方代理人已完成申请文件二稿，请您修改或确认。

该文本我方已存为修改模式，如有修改意见，请您直接在文中改正，我方将可明确的看到修改之处，**请务必保留修改标记**。

如贵方经修改、确认认为可以提交，请在回传邮件中添加提交文本附件，并在正文标注“确认以附件文本提交专利申请”字样。

感谢您对我们工作的配合与协助，如有任何疑问请随时和我们联系。

顺祝

业祺！

北京方韬知识产权代理有限公司

北京方韬法业专利代理事务所

客户代表：许玉

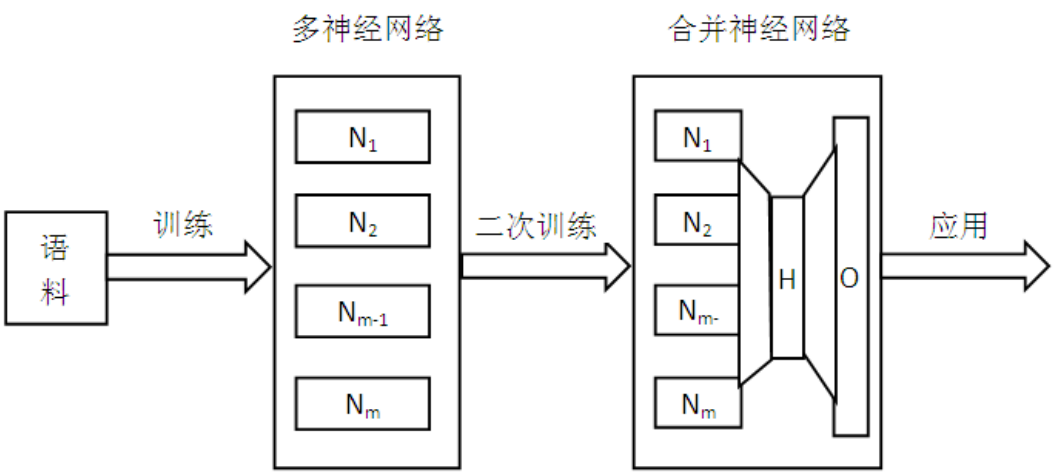
专利部　刘晶婷

Tel：010-68335705/06/07/08

说明书摘要

本发明是有关于一种基于分布式神经网络的语言模型（NNLM）训练方法及其系统，该方法包括：将大词表拆分为多个小词表；将每个小词表对应一个神经网络语言模型，每个神经网络语言模型的输入维数相同且独立进行第一次训练；将各神经网络语言模型的输出向量合并并进行第二次训练；得到归一化的神经网络语言模型。该系统包括：输入模块、第一次训练模块、第二次训练模块和输出模块。本发明通过多个神经网络训练学习不同词表，充分利用神经网络的学习能力，大大降低对大词表学习训练的时间，同时将大词表的输出进行归一化，实现多个神经网络的归一和共享，使得 NNLM 尽可能学习更多的信息，从而提高大规模语音识别和机器翻译等相关应用任务中的准确率。

摘　要　附　图



权利要求书

1、一种基于分布式神经网络的语言模型训练方法，其特征在于包括以下步骤：

将大词表拆分为多个小词表；

将每个小词表对应一个小神经网络语言模型，每个小神经网络语言模型的输入维数相同且独立进行第一次训练；

将各小神经网络语言模型的输出向量合并并进行第二次训练；

得到归一化的神经网络语言模型。

2、根据权利要求 1 所述的基于分布式神经网络的语言模型训练方法，其特征在于所述的第二次训练神经网络的输出概率计算公式为：

$p_i=\mathrm{softmax}_i(\varphi((P_1(o)^T, P_2(o)^T, \cdots, P_m(o)^T)^T))$

其中，$\varphi(x)=(\tanh(x\times M+b))\times V+d$，

$\mathrm{softmax}(x)=\exp(x_i)/(\Sigma_r(\exp(x_r)))$，

$P_1(o)^T$，$P_2(o)^T$，…，$P_m(o)^T$为每个小神经网络语言模型的输出，

x 为第二次训练神经网络的输入，M 是映射层到隐藏层的系数矩阵，b 为隐藏层节点的偏移量，V 为隐藏层到输出层的系数矩阵，d 为输出层节点的偏移量。

3、根据权利要求 1 所述的基于分布式神经网络的语言模型训练方法，其特征在于所述的第一次训练神经网络的输出概率计算公式为：

$P(w_j|h_j)=\varphi_1(w_j)p_1(w_j|h_j)+\varphi_2(w_j)p_2(w_j|h_j)+\cdots+\varphi_m(w_j)p_m(w_j|h_j)$

其中，w_j即为该小神经网络语言模型对应词表中的第 j 个词，$\varphi(w)$是归一化函数 $\Sigma\varphi(w)=1$，当 w 在第 m 小词表时，相应的φ_m较大。

4、一种应用权利要求 1-3 中任一项所述方法的基于分布式神经网络的语言模型训练系统，其特征在于包括：

输入模块，用于将大词表拆分为多个小词表；

第一次训练模块，包括多个独立的小神经网络语言模型模块，每个小神经网络语言模型模块对应输入模块的一个小词表进行独立训练；

第二次训练模块，用于将第一次训练模块中各小神经网络语言模型模块的输出向量合并，并进行第二次训练；

输出模块，用于输出第二次训练模块得到归一化的神经网络语言模型。

5、根据权利要求 4 所述的基于分布式神经网络的语言模型训练系统，其特征在于所述的第一次训练模块中的各小神经网络语言模型模块包括接收输入量的映射层、进行概率计算的隐藏层和输出层。

说　明　书

基于分布式神经网络的语言模型训练方法及其系统

技术领域

本发明涉及一种语言模型，特别是涉及一种基于分布式神经网络的语言模型训练方法及其系统。

背景技术

语言模型在自然语言处理中有着非常重要的作用，特别是在大规模语音识别和机器翻译中。当前主流的语言模型为基于概率的统计语言模型，特别是基于 n-gram 的统计模型。随着神经网络的兴起，越来越多的人利用神经网络生成统计语言模型。

统计语言模型被广泛应用于各种自然语言处理问题，如语言识别、分词、机器翻译、词性标注等。简单的说，统计语言模型就是用来计算一个句子的概率的模型，即

$p(w_1, w_2, \cdots, w_k)$

已知一个句子（词语序列）$s=w_1, w_2, \cdots, w_k$，他们的概率可以表示为：

$$p(s)=p(w_1, w_2, \cdots, w_k)=p(w_1)p(w_2|w_1)\cdots p(w_k|w_1, w_2, \cdots, w_{k-1})$$

举个简单的例子，输入拼音串为 *nixianzaiganshenme*，对应的输出可以有多种形式，如“你现在干什么”、“你西安再赶什么”，等等。那么到底哪个才是正确的转换结果呢？利用语言模型，我们知道前者的概率大于后者，

因此转换成前者在多数情况下比较合理。

目前使用比较多的统计语言模型是 n-gram 语言模型，就是将上式中条件概率部分 $p(w_k|w_1, w_2, \cdots, w_{k-1})$简化为 $p(w_k|w_1, w_2, \cdots, w_{n-1})$。，实际应用中 n 一般取为 n=3 或 n=4，即三元和四元的 n-gram 语言模型。

基于神经网络的语言模型最早由 Bengio 等人在 2001 年发表在 NIPS 上的文章《A Neural Probabilistic Language Model》中提出。请参阅图 1 所示，w_{j-n+1}，w_{j-n+2}，…，w_{j-1}就是当前词 w_j的前 n−1 个词。现在需要根据这已知的 n−1 个词预测词 w_j的概率，即计算：

$$P(w_j=i|h_j) \qquad \forall i \in [1.N]$$

首先，从输入层到映射层（projectionlayer）由一个|V|×m 维的映射矩阵 C(W)完成，其中|V|表示词表的大小（语料中的总词数），m 表示映射空间的维度。

网络的第一层（映射层）是将 $C(w_{j-n+1})$，$C(w_{j-n+2})$，…，C(wj−1)这 n−1 个向量首尾相接拼起来，形成一个(n−1)*m 维的向量，记为 C_l。网络的第二层（隐藏层）由映射层经过线性变换 d+Hx 附加一个激活函数 tanh（）得到，其中 d 是一个偏置量，tanh()定义如下：。

$$d_j=\tanh\left(\Sigma_l m_{jl} c_l + b_j\right)。$$

网络的第三层（输出层）一共有|V|个节点，每个节点 y_i表示下一个词为 i 的概率。该层由隐藏层输出经过线性变换后附加 softmax 激活函数进行归一化得到，计算公式为：

$$o_i=\Sigma_j v_{ij} d_j + k_i$$

$$P_i=\exp(o_i)/(\Sigma_r (\exp(o_r)))$$

相比于当前主流的 n-gram 语言模型，基于神经网络的语言模型（NNLM）对模型参数的共享更直接有效（共享映射矩阵），因而对低频词具有天然的光滑性，因此在建模能力上具有显著优势。另一方面，NNLM 也

具有明显的弱点,,包括:

1）训练效率低，耗时长，特别是当词表增大到超过 10w 时，训练时间难以接受；

2）解码时间较长，不能满足实际要求；

3）单个神经网络结构学习大词表能力差，一旦需要学习的信息增多，单个神经网络结构就无法满足。

为了解决上述问题，可以考虑利用分布式神经网络进行大词表语言模型学习。但是，基于现有技术，在进行多个神经网络学习时，各个神经网络需要独立学习不同词汇，极易导致最后产生的语言模型的概率尺度不统一，即未归一化。

因此，如何能同时解决大词表神经网络语言模型学习和多个神经网络之间的归一化问题，在大规模语音识别和机器翻译中显得尤为重要。

发明内容

本发明要解决的技术问题是提供一种基于分布式神经网络的语言模型训练方法及其系统，使其能够同时解决大词表神经网络语言模型学习和多个神经网络之间归一化的问题，从而克服现有的神经网络语言模型学习方法的不足。

为解决上述技术问题，本发明提出一种基于分布式神经网络的语言模型训练方法，包括以下步骤：将大词表拆分为多个小词表；将每个小词表对应一个小神经网络语言模型，每个小神经网络语言模型的输入维数相同且独立进行第一次训练；将各小神经网络语言模型的输出向量合并并进行第二次训练；得到归一化的神经网络语言模型。

作为本发明的一种改进，第二次训练的概率计算公式为：

$p_i=\mathrm{softmax}_i(\varphi((P_1(o)^T, P_2(o)^T, \cdots, P_m(o)^T)^T))$

其中，$\varphi(x)=(\tanh(x\times M+b))\times V+d$，

$\mathrm{softmax}(x)=\exp(x_i)/(\Sigma_r(\exp(x_r)))$，

$P_1(o)^T$，$P_2(o)^T$，…，$P_m(o)^T$为每个小神经网络语言模型的输出，

x 为输入的词向量，M 是映射层到隐藏层的系数矩阵，b 为隐藏层节点的偏移量，V 为隐藏层到输出层的系数矩阵，d 为输出层节点的偏移量。

所述的第一次训练的概率计算公式为：

$$P(w_j|h_j)=\varphi_1(w_j)p_1(w_j|h_j)+\varphi_2(w_j)p_2(w_j|h_j)+\cdots+\varphi_m(w_j)p_m(w_j|h_j)$$

其中，w_j即为该小神经网络语言模型对应词表中的第 j 个词，$\varphi(w)$是归一化函数 $\Sigma\varphi(w)=1$，当 w 在第 m 小词表时，相应的φ_m较大。

此外，本发明还提供了一种应用上述方法的基于分布式神经网络的语言模型训练系统，包括：输入模块，用于将大词表拆分为多个小词表；第一次训练模块，包括多个独立的小神经网络语言模型模块，每个小神经网络语言模型模块对应输入模块的一个小词表进行独立训练；第二次训练模块，用于将第一次训练模块中各小神经网络语言模型模块的输出向量合并，并进行第二次训练；输出模块，用于输出第二次训练模块得到归一化的神经网络语言模型。

作为进一步改进，所述的第一次训练模块中的各小神经网络语言模型模块包括接收输入量的映射层、进行概率计算的隐藏层和输出层。

采用这样的设计后，本发明至少具有以下优点和有益效果：

1、通过多个神经网络训练学习不同词表，充分利用神经网络的学习能力，大大降低对大词表学习训练的时间，这样可以解决利用 NNLM 训练学习大词表的时间问题和未充分利用神经网络的问题；

2、可以将大词表的输出进行归一化，实现多个神经网络的归一和共享，

使得 NNLM 尽可能学习更多的信息，从而提高大规模语音识别和机器翻译的准确率。

附图说明

上述仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，以下结合附图与具体实施方式对本发明作进一步的详细说明。

图 1 是现有的神经网络语言模型的示意图。

图 2 是本发明基于分布式神经网络的语言模型的示意图。

图 3 是本发明基于分布式神经网络的语言模型的归一模型示意图。

图 4 是本发明基于分布式神经网络的语言模型的训练方法流程示意图。

具体实施方式

请参阅图 2 所示，为了解决大词表的神经网络模型训练和测试时间过长的问题，我们提出了基于分布式神经网络的语言模型。即将大词表拆分成多个小词表，每个小词表对应一个小神经网络，并且每个小神经网络的输入维数是相同的。

例如，配合参阅图 1 所示，目前现有 10w 的词表，即神经网络的输出层是 10w 维，$P(w_j|h)$中 w 是从 1-10w。本发明分布式神经网络的语言模型就是将输出层拆分成 10 个，即利用 10 个小神经网模型来训练不同的词表，$p_1(w_j|h)$中 w 从 1-1w，$p_2(w_j|h)$中 w 从 1w-2w，依次类推，最后进行网络的合并。

进一步具体来说，从图 2 可以看到，对于分布式神经网络语言模型，首先要对应不同的词表进行训练学习。如现有 N 的词表，将 N 平均分成 m 个小词表。利用图 1 所示的结构进行训练（注意输入层是一样的），分别得

到 m 个小神经网络语言模型：P_1，P_2，P_3....P_m，如图 2 所示将 m 个小神经网络语言模型进行合并，形成一个大的神经网络 P。因此，概率计算公式：

$$P(w_j|h_j)=\varphi_1(w_j)p_1(w_j|h_j)+\varphi_2(w_j)p_2(w_j|h_j)+\cdots+\varphi_m(w_j)p_m(w_j|h_j)$$

其中，$\varphi(w)$是归一化函数：$\Sigma\varphi(w)=1$，当 w 在第 m 小词表时，相应的φ_m将变大。这与神经网络的结构是对应的，因为第 m 个词表独立学习 w 的能力是最强的，因此权重比较大。

这个神经网络即包含了不同词表的更多的信息，同时还可以支持大词表的语言模型的使用。通过利用多个小神经网络训练学习不同词表的语言模型，对大词表中所有词进行学习训练，充分利用神经网络的学习能力，大大降低对大词表学习训练的时间，这样可以解决利用 NNLM 训练学习大词表的时间问题和未充分利用神经网络的问题。

上述虽然解决了大词表的训练问题，但是由于不同小神经网络训练相互独立，所以在最后合并神经网络模型时，需要解决各个小神经网络最后输出概率不归一的问题。因此，为了解决此问题，本发明提出了二次学习归一和合并的分布式神经网络语言模型的算法，在神经网络后增加一个隐藏层和输出层，将多个小神经网络模型进行归一合并，训练生成 $\varphi(w)$。

请继续参阅图 3 所示，在得到的训练好的 m 个小神经网络语言模型（Multi Net）后加一层隐藏层和输出层。首先 m 个模型（Multi Net）同时产生 m 个输出层，将 m 个输出向量合并成一个大的向量 F($F=(P_1(o)^T, P_2(o)^T, \cdots, P_m(o)^T)^T$)，作为第二次训练模型的隐藏层的输入向量。由此，我们可以可得到 w_i 的概率公式：

第二次训练的概率计算公式为：

$$p_i=\text{softmax}_i(\varphi((P_1(o)^T, P_2(o)^T, \cdots, P_m(o)^T)^T))$$

其中，$\varphi(x)=(\tanh(x\times M+b))\times V+d$，

$\text{softmax}(x)=\exp(x_i)/(\text{sum}(\exp(x_r)))$，

$P_1(o)^T$，$P_2(o)^T$，…，$P_m(o)^T$为每个小神经网络语言模型的输出，

x 为输入的词向量，M 是映射层到隐藏层的系数矩阵，b 为隐藏层节点的偏移量，V 为隐藏层到输出层的系数矩阵，d 为输出层节点的偏移量。

经过二次训练，神经网络会不断优化函数 φ(x)，从中也可以看出，当φ(x)被训练的充分光滑后，可以使得多个网络最后完整归为一个大的网络，充分利用了多个神经网络的优势，并且不存在归一化问题。

由于二次训练学习的目标是合并和归一化分布式的神经网络语言模型，所以在更新的过程中，不需要对各个小神经网络进行更新，只需不断训练函数 φ(x)，因而大大减少了计算量。

请配合参阅图 4 所示，本发明对于分布式神经网络语言模型的训练流程为：首先需要利用图 2 的多个小神经网络训练并行生成多个小神经网络 N_1，N_2… N_m，分别对应不同的词表；第一步训练完成后，接下来需要进行二次训练归一和合并多个小神经网络。图 3 给出了对应神经网络模型的构造，利用第一次训练的语料进行二次训练，得到合并和归一后的神经网络。

以上所述，仅是本发明的较佳实施例而已，并非对本发明作任何形式上的限制，本领域技术人员利用上述揭示的技术内容做出些许简单修改、等同变化或修饰，均落在本发明的保护范围内。

说 明 书 附 图

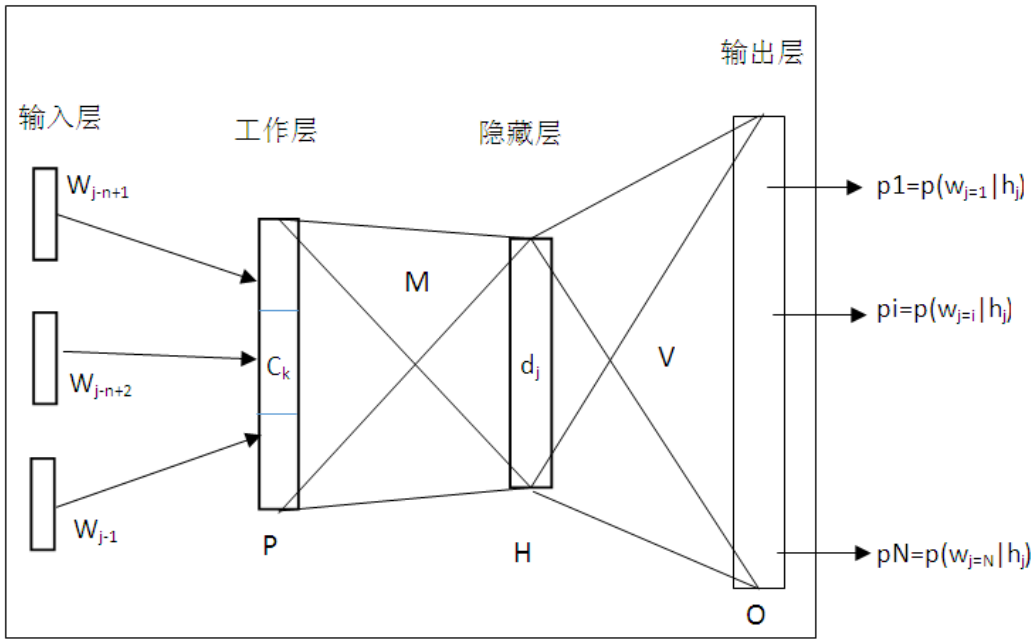

输出层
输入层
工作层
隐藏层
W_{j-n+1}
W_{j-n+2}
W_{j-1}
C_k
M
d_j
V
p1=p($w_{j=1}$|h_j)
pi=p($w_{j=i}$|h_j)
pN=p($w_{j=N}$|h_j)
P
H
O

图 1

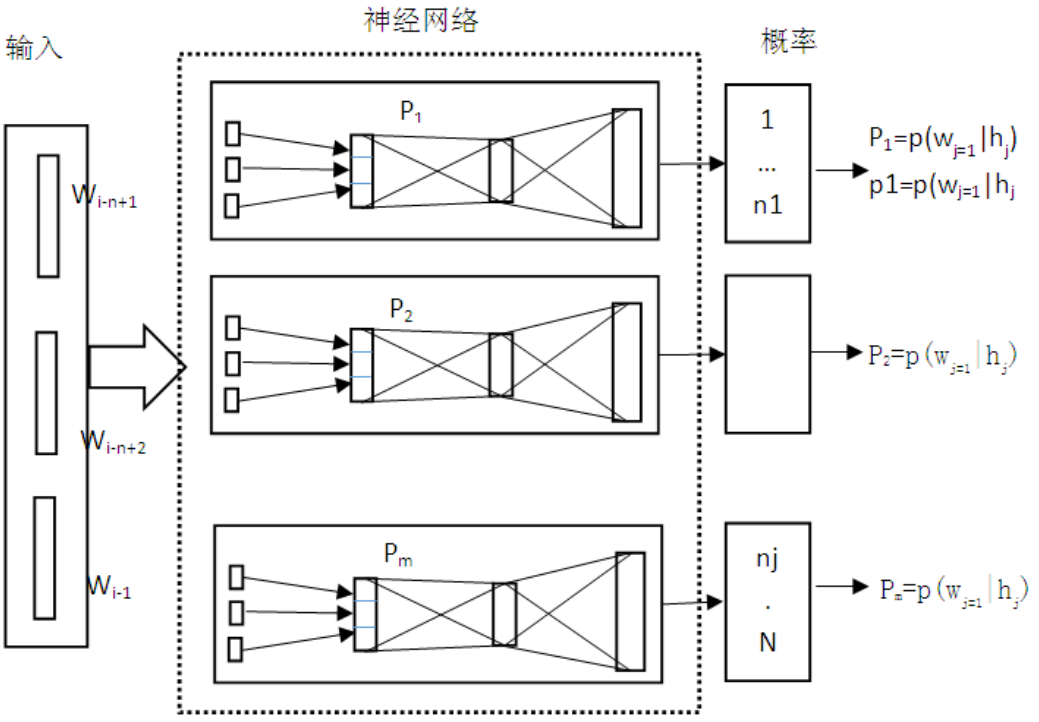

神经网络
输入
概率
W_{i-n+1}
W_{i-n+2}
W_{i-1}
P_1
P_2
P_m
1
...
n1
nj
.
N
P_1=p($w_{j=1}$|h_j)
p1=p($w_{j=1}$|h_j)
P_2=p($w_{j=1}$|h_j)
P_n=p($w_{j=1}$|h_j)

图 2

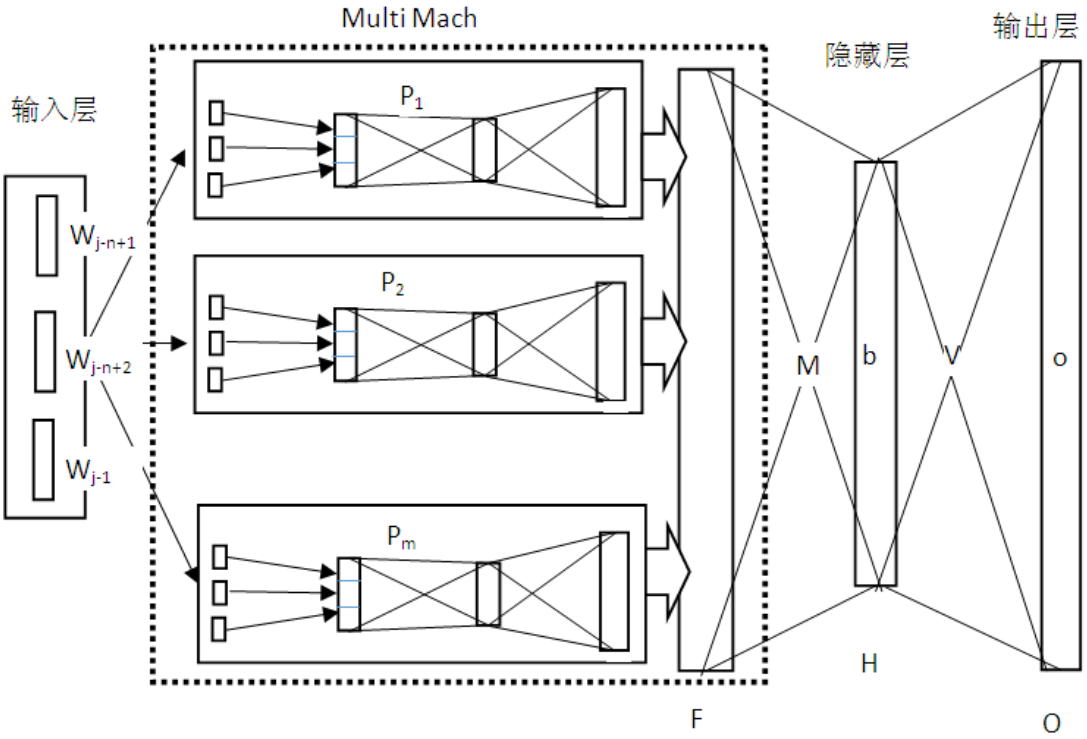

Multi Mach
输入层
隐藏层
输出层
W_{j-n+1}
W_{j-n+2}
W_{j-1}
P_1
P_2
P_m
M
b
V
o
H
F
O

图 3

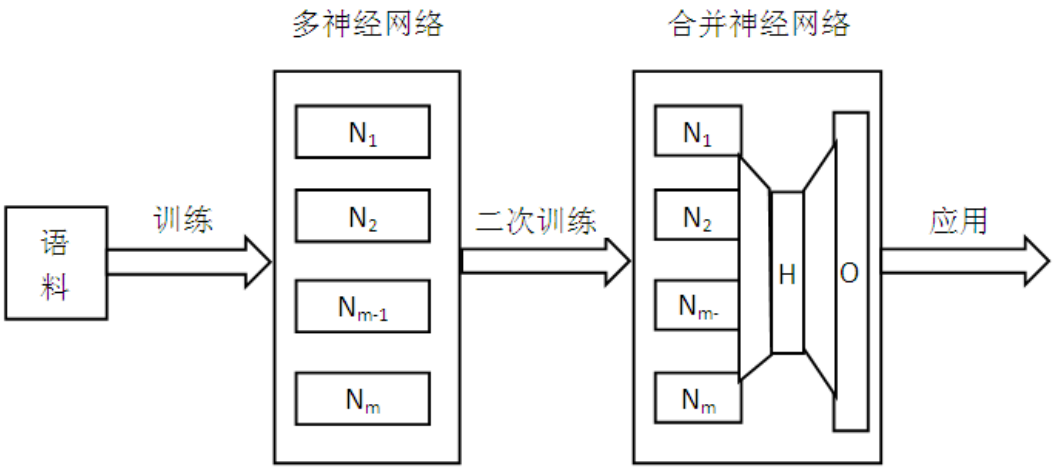

多神经网络
合并神经网络
语料
训练
N_1
N_2
N_{m-1}
N_m
二次训练
N_1
N_2
N_{m-}
N_m
H
O
应用

图 4