**深度神经网络语音识别系统快速稀疏矩阵算法**

**一. 背景**

* 1. **语音识别、深度神经网络与稀疏矩阵运算**

深度神经网络(DNN)已经广泛应用在当代语音识别系统中，并带来识别率的极大提高。一个典型的深度神经网络如图1所示，其中包含一个输入层，多个隐藏层，一个输出层，每层有若干个结点，每个结点的输入由前一层的结点的输出经过线性叠加得到，并通过一个线性或非线性的激励函数，形成该结点的输出。



**图1 DNN结构**

在进行语音识别系统模型训练和识别时，语音数据被分成短时语音帧，这些语音帧经过信号处理之后形成一系列语音特征向量，输入到DNN的输入层，经过神经网络的各个隐藏层，最后进入输出层，形成识别器可用的概率值。

可见，在进行DNN操作时，主要计算为输入向量在整个神经网络的前向传导。这些传导运算可以抽象为矩阵运算。具体而言，将第t层结点的所有结点输出表示成一个向量 OUt ,将第t层到第t+1层之间的网络联接系数表示成At, 则第t+1层结点的输入INt+1可以表示成

INt+1 = At  x OUt 其输出表示为OUt+1 = f (INt), 其中f为激励函数。

 当前语音识别系统中所用的神经网络一般为5-10层，每层结点数为1000到10000，这意味着网络矩阵At 相当庞大，带来巨大的计算压力。如何快速进行矩阵运算，是一个急需解决的问题。

 稀疏矩阵为减小计算量提供了可能。通过将矩阵中绝大部分元素置零，一方面可以节约随储空间，同时可以极大减小计算总量。然则，稀疏矩阵本身的存储和数据索取都需要相应的空间和时间，简单对矩阵依其元素值的大小进行稀疏化并不会提高计算效率。本发明提出一种通过改变稀疏矩阵的拓朴结构对稀疏矩阵进行快速计算的方法。

在下文中的背景知中，我们将简单的介绍一下稀疏矩阵的存储方式和对拓朴结构进行修正过程中需要用到的遗传算法。

* 1. **稀疏矩阵的存储方式**

**CSR(Compressed Sparse Row）**格式是稀疏矩阵中最常用的存储格式，是一种通用存储模式。该存储模式只记录稀疏矩阵中的非零元素，用三个数组分别存储矩阵中的非零元素values，每个非零元对应的列下标columns和每行非零元的起始位置rowIndex。CSR格式的存储实现见图2。



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *values* | = | (1 | -1 | -3 | -2 | 5 | 4 | 6 | 4 | -4 | 2 | 7 | 8 | -5) |
| *columns* | = | (1 | 2 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 3 | 4 | 2 | 5) |
| *rowIndex* | = | (1 | 4 | 6 | 9 | 12 | 14) |   |  |  |  |  |  |  |

其中\*表示矩阵值为零。

**图2 CSR存储格式**

**BSR(Block Compressed Sparse Row)**格式只记录稀疏矩阵中的非零块的位置及其对应的非零块的值。如图3所示，L、M、N、P和Q是矩阵D中的非零元素块，每个非零块对应的列下标rowIndex和行下标columns起始值。

  

 *values* = (1 2 0 1 6 8 7 2 1 5 4 2 7 0 2 0)

*columns* = (1 2 2 3)

 *rowIndex* = (1 3 4 5)

**图3. BSR存储格式**

* 1. **遗传算法**

遗传算法（Genetic Algorithms,GA）是一类借鉴生物界自然选择和自然遗传机制的随机化搜索算法。它模拟自然选择和自然遗传过程中发生的繁殖、交叉和基因突变现象，在每次迭代中都保留一组候选解，并按适度值评估函数从解群中选取较优的个体，利用遗传算子（选择、交叉和变异）对这些个体进行组合，产生新一代的候选解群，重复此过程，直到满足某种收敛指标为止如图4。



**图4. 遗传算法流程**

**二. 问题描述**

1. 基于DNN的语音识别系统需要大量的矩阵运算，利用稀疏矩阵技术可以加快计算速度。然而，基于CSR的存储模式会带来额外的存储开销，同时由于CSR完全打乱了矩阵的空间结构，无法利用CPU的矩阵运算指令，也难以在CPU中进行并行计算，这使得其最终计算效率反而不如原来的稠密矩阵。BSR因为按子矩阵结构存储，因而可以利用CPU的特殊指令和并行计算，比CSR在实际中更有效率。
2. BSR的计算效率取决于实际稀疏矩阵的拓朴结构。稀疏矩阵的非零元素越集中（子矩阵分布越明显），计算效率越高
3. 当前的矩阵稀疏化方法一般采用基于元素值或二阶特性的方法进行矩阵剪裁，使不重要的矩阵元素归零。这一方面并不能保证非零元素分布的规整性和集中性，因而很难通过BSR方式提高效率。

对于上述问题，本专利提出 (1) 一种稀疏矩阵的重整方法，即通过对矩阵的行和列进行调换，修正矩阵的拓扑结构，使得非零值更加集中，从而提高稀疏矩阵的运算效率。 (2) 一种基于遗传算法的拓朴结构选择方法。

**三. 发明要点**

1. **基于行-列置换的稀疏矩阵拓扑结构重整**



**图5. DNN中的t-1/t/t+2层**

为表述清楚，我们将DNN中的任意连续三层抽取出来，如图5所示。 记C为t-1层输出向量，M为t-1层到t层的系数矩阵，B为第t层的偏置向量。V是t层到t+1层的系数矩阵，D为t+1层的偏置向量。依DNN的计算法则，有

H=C$×$M+B (3.1)

O=H$×$V+D (3.2)

注意，在上式3.1和3.2中M，V为稀疏矩阵。为使得稀疏矩阵的非零值更为集中从而提高BSR的计算效率，我们利用神经元网络的特点，在t-1/t层之间加入一个置换层，在t+1层之后再加入一个置换层。通过加入置换层，我们可以任意调换M和V 的行和列顺序，同时保持最后输出结果不变。图6和图7分别以网络拓扑结构和矩阵表示的方示，给出了这一行列交换的例子。



**图6. 神经网络节点调换**



**图7. 神经网络节点调换的矩阵表示**

 如6所示，对神经网络中的节点进行调换，对应了稀疏矩阵的行变换，如图6所示。为了保证在神经网络在交换节点后，神经网络的输出与未变换之前一致，在每层的后面添加了一个置换层，将交换后的结果进行置换，使得输出结果与原神经网络一致。

1. **基于遗传算法的稀疏矩阵拓扑结构选择**

在3.1节中，我们提出通过对矩阵进行列变换可能得到分块更好的稀疏矩阵。然而，如何进行这些行列变换以得到最好的稀疏矩阵依然没有解决。本发明提出用遗传算法对这一问题进行解决。基本思路是，对每经过变换的稀疏矩阵定义一个评价函数(cost function)， 对应的变换方法即定义为遗传算法中的一个个体。通过对个体附加遗传算子，可实现对个体的“遗传与变异”，可以得到更多个体。对这些个体的评价函数时行计算，直到得到的最优个体不再改变，即得到一个最好的变换。

具体方法过程如下：

1. **矩阵的行列变换编码。**如图8所示，对矩阵的行列的下标进行编码，当对矩阵进行行列交换时，对应于下标的变换。



**图8 矩阵变换编码**

1. **选择评价函数。**我们选择矩阵的稀疏性作为评价函数，如3.1式。

 $F=\sum\_{i}^{}c\_{i}×\sum\_{j}^{}b\_{ij}$ (3.1)

其中，c未非零块中非零值的个数，b为非零块中的非零值。3.1式表征了矩阵的稀疏性的大小，当矩阵中非零元素块聚集稠密时，对应的F值变大。

1. **算子选择**

遗传算法的算子包括选择算子、交叉算子和变异算子。但是在矩阵变换的编码中，只是对行列下标进行交换，而不能随便改变行列下标的值。所以在进化过程中，不能使用外变异算子。

 1. 选择算子

为了保证算法的全局搜索能力，采用最优个体保存算子，即父代群体中的最优个体直接进入子代群体中，保证遗传过程中所得到的个体不会被交叉和变异操作所破坏。

2. 交叉算子

交叉算子是产生新个体的主要方法，决定了遗传算法的全局搜索能力，在遗传算法中起关键作用。由于矩阵稀疏变换的编码是包括行和列的下标形式，变换形式比较单一，所以选择简单有效的单点交换算子。

3. 变异算子

变异算子是产生新个体的辅助方法，它决定了遗传算法的局部搜素能力。变异算子和遗传算子相互配合，可以共同完成对搜索空间的全局搜索和局部搜索。为了快速的进行矩阵的稀疏化变换，在这里也引入简单的变异算子。

基于图8所示编码，各种遗传算子的作用形式如图9所示。



**图9. 不同遗传算子的子代产生过程**

**四. 方案优势**

本发明所提出的基于行列交换的拓扑结构重整方法可以有效有高稀疏矩阵的稀疏性，从而有效提高BSR的计算效率; 同时，基于遗传算法的拓扑结构选择可以有效生成并选择优质的拓扑结构，为实现本发明提出的思路提供了算法保证。