

基于部分失真搜索的自组织映射学习算法

陈作平¹, 叶正麟¹, 赵红星^{1,2}, 郑红婵¹

(1. 西北工业大学 理学院, 西安 710072; 2. 榆林学院 数学系, 陕西 榆林 719000)

(ykhmiou@sina.com)

摘 要:针对传统的自组织映射网络在大数据量或高维情形下训练过程较慢的问题,提出了分别使用部分失真搜索和扩展的部分失真搜索来完成传统算法中最耗时的最近邻搜索过程,减少了完成训练所需乘法次数。实验表明,相对于传统的自组织映射学习算法,所提两种方法分别可以节约近 1/3 和 1/2 以上的计算量。

关键词:自组织映射;部分失真搜索;最近邻搜索

中图分类号:TP301 **文献标识码:**A

Learning algorithms for self organizing mapping based on partial distortion search

CHEN Zuo-ping¹, YE Zheng-lin¹, ZHAO Hong-xing^{1,2}, ZHENG Hong-chan¹

(1. School of Science, Northwestern Polytechnical University, Xi'an Shannxi 710072, China;

2. Department of Mathematics, School of Yulin, Yulin Shannxi 719000, China)

Abstract: To accelerate the learning process of Self-Organizing Mapping in the situation of large mount of data or high dimension, two learning algorithms were proposed in this paper, by using Partial Distortion Search and Extended Partial Distortion Search respectively to solve the problem of Nearest Neighbor Search during learning process, which could reduce the multiplications greatly. Experiment results indicate that the proposed algorithms can save up to 1/3 and 1/2 multiplications, compared with traditional Self-Organizing Mapping learning algorithm.

Key words: Self Organizing Map; partial distortion search; nearest neighbor search

0 引言

目前,前向网络、反馈网络等模型虽已得到了广泛的应用,但它们未能充分借鉴人脑的特点,因而其功能有许多不足之处。例如,多层前向网络存在着只能适用于平稳的环境、网络稳定性较差、学习速度慢、可能陷入局部最小值等问题。为此,人们研究了一种更接近于人脑工作特性的人工神经网络模型,即竞争神经网络或自组织神经网络,其代表是芬兰学者 Kohonen 提出的自组织映射网络 (Self-Organizing Mapping, SOM) 或称为自组织特征映射网络 (Self-Organizing Feature Mapping)^[1]。该网络经充分训练并达到收敛后,具有对输入空间的良好近似、拓朴有序、独特的提取非线性数据的内在特征等功能,从而在数据分类、模式识别等领域发挥了重要作用。然而,作为无监督学习网络,不能利用导师信号是其固有缺点;另外,该网络还存在训练过程与结果对初始权值选取及训练样本的输入方式敏感、训练速度慢等缺点。对敏感性问题,许多学者对它在诸如网络结构^[2-6]等方面进行了改进;对速度问题,文献[6]利用主成分分析来构建初始映射,提出了一种快速的批学习算法,文献[7]通过对训练集进行抽样来加速训练过程,文献[8]则通过并行计算来解决训练速度问题,前两者由于是对近似或局部的训练样本进行训练,因此其

训练结果准确性肯定受到影响,后者则是以硬件开销为代价。本文在对传统的 SOM 学习算法进行深入分析的基础上,基于部分失真搜索,提出了两种快速的 SOM 学习算法。

1 传统 SOM 学习算法及其计算复杂度

SOM 的学习算法属于无监督学习 (Unsupervised Learning) 或自组织学习 (Self-Organized Learning) 算法,包含竞争、合作和更新三个过程,具体步骤如下:

1) 设置变量和参量:

$X(n) = [x_1(n), x_2(n), \dots, x_K(n)]^T$ 为输入向量, 或称训练样本;

$W_i(n) = [w_{i1}(n), w_{i2}(n), \dots, w_{iK}(n)]^T$ 为权值向量, $i = 1, 2, \dots, M$;

设置迭代总次数为 N 。

2) 初始化。设置初始学习速率 $\eta(0)$; 用小的随机值初始化权值向量 $W_i(0)$, 并对它以及所有的输入向量 X 进行归一化:

$$X' = \frac{X}{\|X\|}, W'_i(0) = \frac{W_i(0)}{\|W_i(0)\|}$$

其中 $\|W_i(0)\| = \sum_{j=1}^K [w_{ij}(0)]^2$, $\|X\| = \sum_{j=1}^K (x_j)^2$ 分别是权值向量和输入向量的欧氏范数。

收稿日期:2005-08-18; 修订日期:2005-10-30

基金项目:国家自然科学基金资助项目(10070006); 西北工业大学研究生创业种子基金(Z200570)

作者简介:陈作平(1979-),男,湖南桂阳人,博士研究生,主要研究方向:计算机图形学、图像处理; 叶正麟(1943-),男,浙江鄞县人,教授,博士生导师,主要研究方向:计算几何、计算机图形学、图像处理; 赵红星(1961-),男,陕西榆林人,教授,主要研究方向:计算几何; 郑红婵(1971-),女,陕西户县人,讲师,主要研究方向:计算几何。

3) 采样:从输入空间中选取训练样本 X' 。

4) 近似匹配:通过欧氏距离最小的标准:

$$\|X' - W'_c\| = \min_j \|X' - W'_j\|, j = 1, 2, \dots, M \quad (1)$$

来选取获胜神经元 c , 从而实现神经元的竞争过程。

5) 更新:对获胜神经元拓扑邻域 $N_c(n)$ 内的兴奋神经元, 以 Hebb 学习规则:

$$W'_i(n+1) = W'_i(n) + \eta(n)(X' - W'_i(n))$$

更新神经元的权值向量, 从而实现了神经元的合作和更新过程, 其中 $N_c(n)$ 为拓扑邻域半径。

6) 更新学习速率及拓扑邻域并对学习后的权值重新归一化:

$$\eta(n) = \eta(0)(1 - \frac{n}{N}), N_c(n) = \text{INT}[N_c(0)(1 - \frac{n}{N})]$$

$$W'_i(n+1) = \frac{W'_i(n+1)}{\|W'_i(n+1)\|}$$

其中 $\text{INT}(\cdot)$ 是取整函数。

7) 判断迭代次数 n 是否超过 N , 如果 $n \leq N$ 就转到 3), 否则结束迭代过程。

算法中的归一化处理是为了确保通过欧氏距离最小条件选取的获胜神经元, 具有最大的输出; 对于学习速率和拓扑邻域的改变规则, 也可选取别的函数, 但一定要保证函数值随着迭代次数的增加而减小。

下面对算法的计算复杂度进行分析。在 4) 中, (1) 式即是所谓的“最近邻搜索”(Nearest Neighbor Search, NNS) 问题。由上述过程知, 算法的绝大部分计算量都在于此; 特别地, 在欧氏范数下, 算法的计算量主要花费在乘法上。为求解 (1), 传统做法是依次求出 M 个距离, 然后在它们中寻找一个最小者, 我们称之为全失真搜索 (Full Distortion Search, FDS), 其时间复杂度为 $O(M)$ 。假设整个算法经过 N 次迭代后达到收敛, 则其所需乘法次数为 $O(N \times M \times K)$, 其中 N 是与数据集本身性质有关的量, $M \times K$ 是采用 FDS 求解 (1) 所需乘法次数。因此, 当数据集规模 (包括向量维数 K 和个数 M) 较大时, SOM 的学习过程是很慢的。关于最近邻搜索, 已有一些基于树结构的方法, 如 KD-Tree^[9]、r-Tree^[10] 等, 它们通过将数据空间作某种划分, 可把 (1) 的时间复杂度降为 $O(\log_2 M)$, 但是它们存在如下缺点: 一是为存储树结构需要较大空间开销, 如果向量的每个分量用 b 个字节来存储, 则所需额外存储空间至少为 $M \times K \times b$ 字节; 二是建树需要一定的时间, 因此这些方法适合“一次建树, 多次搜索”, 这显然不适合 SOM, 因为后者在每次迭代都需要建立树结构; 三是存在所谓“维数灾难”(dimension disaster), 即其时间复杂度将随向量维数呈指数增长。

我们提出使用下述的部分失真搜索来求解 (1), 一方面它简单易行, 其次它可以尽可能地减少算法的乘法次数, 从而降低 SOM 学习算法的时间复杂度。

2 基于部分失真搜索的 SOM 学习算法

部分失真搜索^[11] (Partial Distortion Search, PDS) 是一种简单有效的最近邻搜索算法, 其基本思想是在计算待搜索向量 W'_i 和目标向量 X 之间失真的过程中始终判断累加的部分

失真是否已超过目前的最小失真; 若是, 则终止它们之间的计算: 假定目前最小失真为 $d_{\min} = d(X, W'_p)$, $1 \leq p \leq M$, 若

$$\sum_{i=1}^s (x_i - w'_{ji})^2 \geq d_{\min}, 1 \leq s \leq K, \text{ 则 } d(X, W'_i) \geq d_{\min}。$$

在 PDS 的基础上, 文献[12]提出了扩展的部分失真搜索方法 (Extended Partial Distortion Search, EPDS), 被称为是一种最优的 PDS 算法, 它能最大限度地减少最近邻搜索的乘法次数。其基本过程可描述如下:

1) 令 $l_j = 1$, 并计算 X 的第 1 维分量 x_1 和权值向量 W'_j 的第 1 维分量 w'_{j1} 间的失真 $D_j = (x_1 - w'_{j1})^2$, $j = 1, 2, \dots, M$, 其中 l_j 表示向量 W'_j 的第 l_j 维;

2) 找出 $D_s = \min_j D_j$, $s = \arg \min D_j$;

3) 若 $l_s = K$, 则 W'_s 为距离 X 最近的权值向量, 算法终止; 否则 $l_s = l_s + 1$, 计算 X 与 W'_s 间的第 l_s 维失真, 将结果同 D_s 相加, 转 2)。

以上两种算法的效率皆在于以比较运算的增加来换取乘法运算的减少 (例如, PDS 是以 s 次比较换得 $K - s$ 次乘法和 $2(K - s)$ 次加法), 因此它们都适用于计算机体系, 因为在后者中比较运算的复杂度同乘法运算相比可以忽略。三种算法的计算代价如表 1 所示。

表 1 FDS, PDS, EPDS 的计算代价比较

	空间开销	乘法次数		比较次数	
		最好情形	最坏情形	最好情形	最坏情形
FDS	无	$M \times K$	$M \times K$	$M - 1$	$M - 1$
PDS	无	$K + M - 1$	$M \times K$	$M - 1$	$(M - 1)K$
EPDS	$M \times b$	K	$M \times K$	$(M - 1)K$	$(M - 1)K^2$

表中各参数含义与第 2 节中所述相同。实践中, 运用 PDS 或 EPDS 进行求解 (1) 所需乘法和比较次数取决于向量各维分量的分布: 分布越不均匀, 所需乘法次数越少。目前, 两者都已被用于矢量量化编码的加速并取得了较好的效果。

根据前面的分析, 我们可用 PDS 和 EPDS 来完成 (1) 的求解, 从而减少 SOM 学习算法的乘法次数。算法步骤如下:

1) 设置变量和参量;

2) 初始化权值向量、学习速率及拓扑邻域半径, 并将权值向量和输入向量归一化;

3) 采样:从输入空间中选取训练样本 X' ;

4) 使用 PDS 或 EPDS, 从权值向量 W'_j , $j = 1, \dots, M$ 中找出与 X' 距离最近的向量;

5) 以 Hebb 学习规则更新获胜神经元拓扑邻域 $N_c(n)$ 内的兴奋神经元;

6) 更新学习速率及拓扑邻域并对学习后的权值重新归一化;

7) 判断迭代次数 n 是否超过 N , 如果 $n \leq N$ 就转到 3), 否则结束迭代过程。

假设算法在 N 次迭代后达到收敛, 则其乘法运算次数为 $\sum_{i=1}^N L_i$, 其中 L_i 表示第 i 次迭代时使用 PDS 或 EPDS 寻找与当前训练样本距离最近的权值向量所需乘法次数。由于 PDS 和 EPDS 对数据集本身的依赖性, 此处的乘法运算次数不再像 FDS 那样有显式的表达式。

3 数值实验

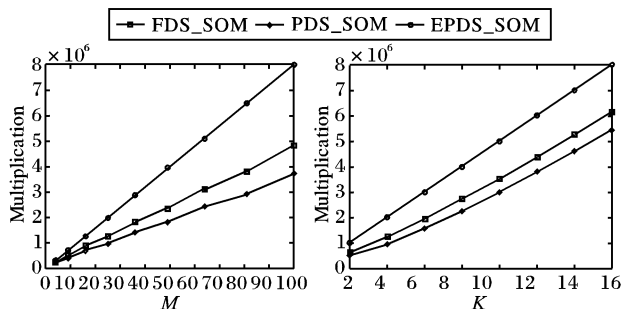
为验证算法的有效性,我们使用 2000 个均匀随机向量(各分量皆是 $[0,1]$ 内的随机数)作为训练样本对基本 SOM 学习算法(记为 FDS_SOM)和改进的 SOM 学习算法(记为 PDS_SOM 和 EPDS_SOM)作了比较实验。根据我们前面的分析,这是 PDS 和 EPDS 表现比较坏的情形。实验参数为:输出层神经元数目 M ,代表着将样本分(聚)为 M 个类,向量维数 K ;比较指标为达到收敛时算法所进行的乘法次数 Multiplications。

首先,我们考察固定 K 时, Multiplication 随 M 的变化情况。固定 $K = 4$ 时,各算法所需的乘法次数如表 2 所示。

表 2 三种算法所在不同聚类数目下所需乘法次数

M	FDS_SOM	PDS_SOM	EPDS_SOM
4	320 000	264 881	226 001
9	720 000	490 801	379 110
16	1 280 000	881 921	684 513
25	2 000 000	1 258 249	959 057
36	2 880 000	1 819 294	1 410 145
49	3 920 000	2 353 175	1 793 499
64	5 120 000	3 120 028	2 423 851
81	6 480 000	3 812 988	2 913 873
100	8 000 000	4 832 464	3 710 724

根据这些数据所作的对比图如图 1(a)所示。在此情形下,相对于 FDS_SOM, PDS_SOM 和 EPDS_SOM 分别可以平均节省 34.9% 和 49.1% 的乘法运算量。



(a) 乘法次数随聚类数目的变化 (b) 乘法次数随向量维数的变化
图 1 乘法次数的变化情况

其次,我们固定 M ,考察 Multiplication 随 K 的变化情况。令 $M = 25$,各算法所需乘法次数如表 3 所示。

表 3 三种算法在不同向量维数下所需乘法次数

K	FDS_SOM	PDS_SOM	EPDS_SOM
2	1 000 000	658 126	546 360
4	2 000 000	1 258 249	959 057
6	3 000 000	1 938 665	1 553 072
8	4 000 000	2 740 748	2 251 843
10	5 000 000	3 538 324	2 993 226
12	6 000 000	4 397 585	3 790 890
14	7 000 000	5 268 950	4 607 936
16	8 000 000	6 128 077	5 422 979

此情形下分别可平均节省乘法运算量 30.3% 和 41.6%, 其对比图见图 1(b)。从图 1 可见三种算法所需乘法次数随聚类数和维数增长几乎是线性增长关系,因此随聚类数和维

数的增加, PDS_SOM 和 EPDS_SOM 所节省的计算量越明显, 从而在大数据量和高维情形, 本文方法所体现的潜在优势越大。此外, 还可看到 EPDS_SOM 相对于 PDS_SOM 方法在节省乘法计算量方面的优势, 这一点与第 3 节中的分析是一致的。

综上, 两种情形中, PDS_SOM 和 EPDS_SOM 分别可平均节省乘法运算量为 32.6% 和 45.4%。此外, 若训练样本来自于非均匀分布, 则使用 PDS_SOM 和 EPDS_SOM 可更大幅度地节省乘法次数。

4 结语

本文通过使用部分失真搜索来实现 SOM 学习算法中最耗时的最近邻搜索, 提出了两种简单、快速的 SOM 学习算法: PDS_SOM 和 EPDS_SOM。实验表明, 即使在较坏情形下, 相对于传统的 SOM 学习算法, 本文算法分别亦可节省 1/3 和 1/2 的乘法运算量, 从而较大幅度地提高了 SOM 的学习速度。鉴于 SOM 网络的广泛应用性, 本文的快速 SOM 学习算法可用于任何需要数据分类或聚类的场合, 例如可以将它用于分形编码的加速方面, 这是我们下一步的工作。

参考文献:

- [1] KOHONEN T. Self-Organizing Maps [M]. Springer Verlag, New York, 1997.
- [2] OJA M, KASKI S, KOHONEN T. Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum [J]. NEURAL COMPUTING SURVEYS 3, 2002, 1-156.
- [3] 傅彦, 周俊临. 基于自增长型多级自组织映射网络的模式识别 [J]. 计算机科学, 2004, 31(5): 159-162.
- [4] 王升明, 李森. 一种基于改进的自组织特征映射网络的文档聚类方法 [J]. 计算机工程与应用, 2005, 41(3): 167-169.
- [5] SEIFFERT U, MICHAELIS B. Growing Multi-Dimensional Self-Organizing Maps [J]. International Journal of Knowledge-Based Intelligent Engineering Systems, 1998, 2(1): 42-48.
- [6] KINOUCHI M, et al. Quick Learning for Batch-Learning Self-Organizing Map [J]. Genome Informatics, 2002, 13: 266-267.
- [7] GOLLI AE. Speeding up the self organizing map for dissimilarity data [Z].
- [8] BANDEIRA N, LOBO VJ, MOURA-PIRES F. Training a self-organizing map distributed on a PVM network [J]. in: Proceedings of IEEE Joint Conference on Neural Networks, 1998: 457-461.
- [9] FRIEDMAN JH, BENTLEY JL, FINKEL RA. An algorithm for finding best matches in logarithmic expected time [J]. ACM Trans. Math. Software, 1997, 3: 209-226.
- [10] ARYA S, MOUNT D, NETANYAHU N, et al. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions [A]. Proceedings of the Fifth Annual ACM-SIAM Symp on Discrete Algorithms [C]. 1994. 573-582.
- [11] BEI CD, GREY RM. An Improvement of the minimum distortion encoding algorithm for vector quantization [J]. IEEE Trans. On Communications, 1985, 33(10): 1132-1133.
- [12] CHEN SH, PAN JS. Fast search algorithm for VQ-based recognition of isolated word [J]. IEE Proceedings-I, 1989, 136(6): 391-396.