基于 openMP 的鸢尾草分类并行算法实践

田粤蒙 [1)]

[1)]（深圳大学计算机与软件学院　深圳　中国　518000）

摘　要　本文是对多路处理器计算机教学实验系统所配套的《并行算法实践》实验指导书中实验内容的补充。使用龙芯 3A 处理器组成的多路处理器计算机教学实验系统，是一种内可配置外可扩展的实验硬件平台。该实验平台所配套的《并行算法实践》实验指导书包含若干基于 openMP 与 MPI 的典型应用的并行算法编程实现，可以帮助学生了解部分并行算法的设计思想与实现方法。但是，该实验指导书所涉及的算法不包含机器学习、深度学习的相关算法，不利于学生对热门技术的了解与拓展学习。因此本文提出基于 openMP 的鸢尾草分类并行算法，并行化相关基础算法，如：矩阵乘法、矩阵加减等，并为该算法中涉及的深度神经网络（DNN）技术设计了向量并行化优化方法。

关键词　并行算法实践；鸢尾草分类；openMP；DNN

Parallel algorithm of iris classification based on OpenMP

Tian Yue-Meng[1)]

[1)](School of computer and software, Shenzhen University, Shenzhen 518000)

Abstract　This paper is a supplement to the experiment instruction of parallel algorithm practice, which is a complete set of multi processor computer teaching experiment system. The multi-channel processor computer teaching experiment system, which is composed of Godson 3A processor, is a kind of experiment hardware platform which can be configured inside and extended outside. The experimental instruction of parallel algorithm practice, which is supported by the experimental platform, contains several parallel algorithm programming implementations based on typical applications of OpenMP and MPI, which can help students understand the design ideas and implementation methods of some parallel algorithms. However, the algorithm involved in the experimental instruction does not include machine learning and deep learning algorithms, which is not conducive to students' understanding and extended learning of popular technologies. Therefore, this paper proposes a parallel algorithm of iris classification based on OpenMP, which parallelizes the basic algorithms such as matrix multiplication, matrix addition and subtraction, and designs a vector parallel optimization method for the depth neural network (DNN) technology involved in the algorithm.

Keywords Parallel algorithm practice; iris classification; OpenMP; DNN

1 意义和目标

龙芯 3A 处理器组成的多路处理器计算机教学实验系统，是一种内可配置外可扩展的实验硬件平台。该实验平台所配套的《并行算法实践》实验指导书包含若干基于 openMP 与 MPI 的典型应用的并行算法编程实现，可以帮助学生了解部分并行算法的设计思想与实现方法。

但是，该实验指导书所涉及的算法不包含机器学习、深度学习等相关算法，不利于学生对热门技术的了解与拓展学习。本文提出的基于 openMP 的鸢尾草分类并行算法，可以让学生从多角度多层次认识当今较为热门的人工智能领域算法。从其中最基础的算法——矩阵乘法、梯度下降等出发，认识人工智能之本质。

此外，本文为该算法中涉及的深度神经网络技术设计了相关并行化实验指导方案，从而优化人工智能领域经典的“鸢尾草分类”问题，使学生掌握并行程序在深度学习领域中的设计方法。同时对该问

题进行了算法分析，加深学生对影响并行程序性能的关键因素的了解。

2 项目选题概况

2.1 并行实验平台的国内外概况

龙芯 3A 处理器组成的多路处理器计算机教学实验系统，是一种内可配置外可扩展的实验硬件平台。该实验硬件平台核心 CPU 龙芯 3A 处理器是由我国自主研发的高性能处理器，具有自主可控等优点。并配套相关教材如:《计算机体系结构（并行）》、《并行算法实践》、《计算机操作系统（并行）》等，可以满足大学各阶段学生的教学需求。

美国伊利诺伊理工大学为学生建立的 Jarvis 的计算机集群，以及由伯爵大学集群计算小组等共同开发的 LittleFe 可移植计算集群与多路处理器计算机教学实验系统有着相似的结构。但多路处理器计算机教学实验系统部署操作更为方便，存放于实验箱之中，具有小型化等优点。此外，多路处理器计算机教学实验系统拥有多种互联方式，不但可以通过硬件设备直接操作，还可以通过前端 CPU 的操作系统操作，甚至可以通过网络连接的方式，直接在 web 页面或电脑软件上进行操作。

多路处理器计算机教学实验系统的新版实验箱——第三代多路处理器教学实验箱系统，属国内首创，使用 linux 操作系统，可以进行多处理器的程序设计与运算，也可以将多个实验箱进行网络互联，形成实验箱集群，从而获得更多的计算节点，更高的运算性能。

2.2 鸢尾草分类并行算法概况

实际应用中，在机器学习、深度学习领域主要利用 cuda、cudnn、tensorflow、pytorch 等工具在 GPU 硬件上进行加速，可以达到很好的加速效果。

但由于目前的多路处理器计算机教学实验系统不包含 GPU 硬件，因此采用 openMP 技术在 CPU 上进行并行化加速。

相比较使用 GPU 加速，CPU 并行化虽然无法做到最大程度的加速，但是可以达到类似的优化效果。在已有硬件条件下，使用 CPU 并行化也更能发挥多路处理器计算机教学实验系统四块龙芯 3A 处理器的性能优势。

3 项目原理

3.1 鸢尾草分类问题

鸢尾草分类问题的背景故事是一位研究植物的学者，对鸢尾草的品类有浓厚的兴趣。他经过多年的研究，发现鸢尾草主要有以下四种性状，分别是：花瓣的宽度、花瓣的长度、花萼的宽度、花萼的长度，其单位均为厘米。此外他还知道鸢尾草具有三个品类，分别是：山鸢尾、变色鸢尾、维吉尼亚鸢尾。根据已知的性状数据，可以推测鸢尾草的品类。

由于已知鸢尾草的性状数据，因此这是一个监督式学习问题。这个问题的目标是在多个鸢尾草品类中进行预测，因此这也是一个分类问题，其中，输入为鸢尾草的性状数据，输出为鸢尾草的品类。再由于数据集中鸢尾草的品类有三种，因此这还是一个三分类问题。

在本项目研究的鸢尾草分类问题中，将数据集分为训练集与测试集两部分，训练集用于训练模型，测试集用于测试模型。

下表为数据集中的个别样本：

表 1 鸢尾草分类问题部分数据集

瓣宽	瓣长	萼宽	萼长	品类
2.2	5.6	2.8	6.4	2
1.0	3.3	2.3	5.0	1
1.7	4.5	2.5	4.9	2
0.1	1.5	3.1	4.9	2

3.2 深度神经网络原理

3.2.1 深度神经网络分类器拓扑结构

该算法会训练一个具有一个隐藏层，每个隐藏层包含五个节点的深度神经网络分类器。

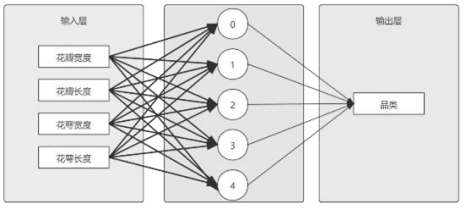

图 1 深度神经网络分类器拓扑结构

3.2.2 梯度下降算法原理

梯度下降算法是一种迭代方法，广泛用于求解机器学习、深度学习等算法的模型参数，这是一种特别经典的最优化算法，可以用来迭代逼近最小偏差模型。

其迭代公式为：

$$W_{k+1} = W_k + \alpha_k \times S^{(k)}$$

$S^{(k)}$指的是梯度负方向。

α_k指的是沿梯度方向上的搜索步长，即学习速率，一般使用时可以是一个常数，这是机器学习领域的一个常见的超参数。

当学习速率过大时，训练模型容易发散，当学习速率太小时，训练模型收敛过慢。本模型设定学习速率为 0.005。

3.3 深度神经网络的前向反馈与反向传播

3.3.1 前向反馈

深度神经网络的前向反馈即为根据输入求解输出的过程。在本项目研究的鸢尾草分类问题中，前向反馈公式如下所示。

输出层到隐藏层：

$$A = \frac{1}{1 + e^{-W_1 \times X}}$$

其中，X是输入矩阵，大小为 4*1，代表鸢尾草四种性状数据；W_1是隐藏层参数，大小为 5*4；A是隐藏层计算结果，大小为 5*1。

隐藏层到输出层：

$$Y' = W_2 \times A$$

其中，A 是隐藏层计算结果，大小为 5*1；W_2是输出层参数，大小为 1*5；Y'是输出结果，大小为 1*1，代表鸢尾草品类。

通过前向反馈，根据输入的性状数据X即可得出当前参数W_1与W_2下预测的输出品类结果Y'。

3.3.2 反向传播

深度神经网络的反向传播即为前向反馈的逆过程，其目的是更新权重参数（即本问题中的W_1与W_2），反向传播主要有如下几个步骤：

1）将前向反馈所得到的输出与期望的输出进行比较，并计算误差

2）进行反向传播，将误差分散到每个权重参数上

3）根据梯度下降原理更新权重参数

4）重复该过程，直至到达停止条件

在本文研究的问题中，使用均方误差函数作为代价函数（cost function）：

$$cost = \frac{1}{n}\sum(Y' - Y)^2$$

设置好最大迭代次数等终止条件，输入训练数据，通过反向传播，即可不断更新权重参数，从而不断优化该深度神经网络分类器模型，最终得以用于预测鸢尾草品类。

3.4 矩阵的并行算法

3.4.1 算法描述

深度神经网络的前向反馈与反向传播计算需要使用到许多矩阵运算，包括矩阵乘法、矩阵数乘、矩阵加减法、矩阵转置、矩阵求和等。从深度神经网络的算法结构便可以看出，深度神经网络的核心是矩阵运算，优化深度神经网络的核心自然也是矩阵运算。这也是为何大多数深度神经网络等人工智能算法都采用 GPU 进行加速的原因，GPU 原本是用于计算机图形计算的，而计算机图形计算的基础也是矩阵，因此，GPU 在结构上相当适合计算矩阵，同时 GPU 还对矩阵运算进行了诸多优化。

在该项目中，由于多路处理器计算机教学实验系统无 GPU 设备，因此采用 CPU 并行处理的方式来对深度神经网络进行优化。采用 openMP 技术，将矩阵乘法、矩阵数乘、矩阵加减法、矩阵转置、矩阵求和并行化。

以下为本项目矩阵运算中最为关键的矩阵乘法的并行化代码（详细代码见附件）：

```
//矩阵乘法
Matrix* Matrix::mul(const Matrix* X) {
    if (w == X->h) {//确保矩阵阶数匹配
        Matrix* re = new Matrix(h, X->w);
        int i, j, k;
#pragma omp parallel private(i, j, k) num_threads(NUM_THREAD)
//设置并行块，设置并行核数为 NUM_THREAD
        {
#pragma omp for schedule(dynamic)
//使用动态调度，以提升并行性能
            for (i = 0; i < h; i++) {
                for (j = 0; j < X->w; j++) {
                    for (k = 0; k < w; k++) {
                        re->m[i][j] = re->m[i][j] + m[i][k] * X->m[k][j];
                    }
                }
            }
        }
        return re;
    }
    else {
        cout << "错误：矩阵阶数不匹配！" << endl;
        return NULL;
    }
}
```

图 2 矩阵乘法并行化代码

3.4.2 算法分析

测试矩阵相乘程序的执行时间和加速比，并分析测试结果：将方阵阶固定为 1000，节点数(线程数)分别取 1、2、4，为了减少可能产生的误差，每项测试进行 5 次，取平均值作为测试结果。

测试结果：

串行平均执行时间为：6.89124 秒

表 2 加速比随节点数的变化情况

节点数	1	2	4
平均值	7.2261	4.8102	3.8514
加速比	0.9537	1.4326	1.7893

以下是由以上数据绘制的加速比随节点数的变化情况的折线图。

根据折线图可以清楚地看出，加速比随节点数的增加呈线性提升。因此该优化方案是行之有效的。

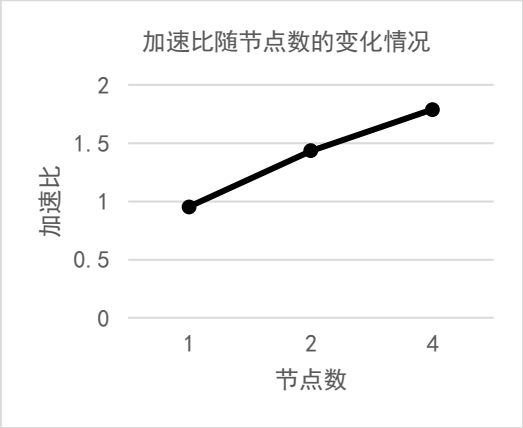

图 3 加速比随节点数变化情况

此外，由于是在个人电脑进行测试，所使用的个人电脑仅有 4 个逻辑内核，与多路处理器计算机教学实验系统的硬件条件相比有一定的差距。

但是这种硬件差距只会影响加速的幅度，最终结果依旧是“加速比随节点数的增加呈线性提升”，因此可以进一步加以使用。

3.5 深度神经网络的向量化并行处理

3.5.1 算法描述

根据 3.3 节可知，该项目的深度神经网络的输入是一个矩阵，代表鸢尾草花瓣的宽度、花瓣的长度、花萼的宽度及花萼的长度。输出是另一个矩阵，代表鸢尾草的品类。

其中，输入矩阵一般是一个 4*1 大小的矩阵，输出矩阵是一个 1*1 大小的矩阵，这对应着一组鸢尾草数据的训练操作。将此操作执行多次，便可以训练出一个可以用于预测鸢尾草品类的深度神经网络模型。

假设有 100 株鸢尾草数据，则要进行 100 次梯度下降操作。我们可以对这个步骤进行向量并行化，把这个问题向量化，最后化为矩阵乘法的并行化问题。从而使之得到优化。

方法是：输入 4*100 大小的矩阵，代表 100 株鸢尾草的 4 种性状数据。同样将该矩阵放入该深度神经网络训练模型，最终得到的是 1*100 大小的输出矩阵，代表这 100 株鸢尾草的品类。如此一来，只需要进行一次梯度下降操作即可训练出具有同样效果的模型。

由于转换为了矩阵乘法的并行化问题，当鸢尾草的数据量巨大时，这个向量化并行方法的加速效果会更为明显。

3.5.2 算法分析

由于该算法最终转换成矩阵乘法问题，因此可以套用上一节矩阵运算的并行化算法分析结论。该向量化并行算法加速比随节点数的增加呈线性提升。

4 实现环境

操作系统：Windows 10 教育版

系统类型：64 位操作系统，基于 x64 的处理器

处理器：Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz

基准速度：2.70 GHz

插槽：1

内核：2

逻辑处理器：4

Hyper-V 支持：是

L1 缓存：128 KB

L2 缓存：512 KB

L3 缓存：3.0 MB

已安装的内存：8.00GB（7.89GB 可用）

开发环境：Visual Studio 2017 社区版

5 实现过程

5.1 编写矩阵类

矩阵类的 UML 图如下（详细代码见附件）：

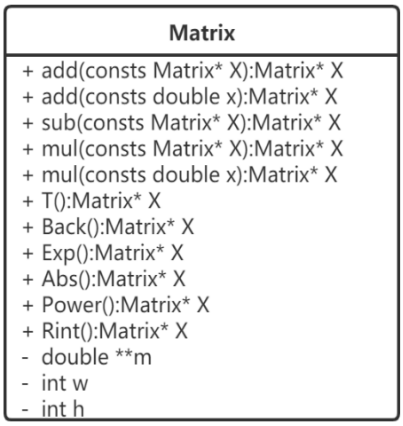

图 4 矩阵类 UML 图

在该项目中，为了提升运行过程时性能，提升内存使用效率，采用的是以指针传参及返回指针的方式实现矩阵类。该方法既可以避免类对象拷贝构造带来的额外开销，又使得程序可以在必要的时候对矩阵所占用的堆内存进行及早释放，从而降低程序的内存占用。

5.2 编写深度神经网络

伪代码如下：

```
for i in range(0, epochs):
    # 前向反馈
    z1 = dot(w1, a0)  # 5*100
    a1 = 1.0 / (1.0 + exp(-z1))  # 5*100
    z2 = dot(w2, a1)  # 1*100
    # 反向传播
    dz2 = z2 - y  # 1*100
    cost.append(sum(power(abs(dz2), 2)) / 100)
    dw2 = dot(dz2, a1.T)  # 1*5
    da1 = dot(w2.T, dz2)  # 5*100
    dz1 = multiply(da1, multiply(a1, 1 - a1))  # 5*100
    dw1 = dot(dz1, a0.T)  # 5*4
    # 更新隐藏层参数
    w1 = w1 - l_rate * dw1
    w2 = w2 - l_rate * dw2
```

图 5 深度神经网络伪代码

dot 代表矩阵乘法；

multiply 代表对两个矩阵按所在位置一一相乘；

exp、abs、power 代表对矩阵每个位置的元素施行相应计算，sum 代表矩阵求和。

6 结果及分析和总结

6.1 运行结果

```
成功读取鸢尾草数据文件："iris.csv"
共包含100组鸢尾草训练数据

开始训练深度神经网络
当前迭代次数为10000：cost = 0.0450706
当前迭代次数为20000：cost = 0.00999052
当前迭代次数为30000：cost = 0.00774986
当前迭代次数为40000：cost = 0.00672006
当前迭代次数为50000：cost = 0.00612204
当前迭代次数为60000：cost = 0.00573141
当前迭代次数为70000：cost = 0.0054545
当前迭代次数为80000：cost = 0.00524624
当前迭代次数为90000：cost = 0.00508233
当前迭代次数为100000：cost = 0.00494841
训练完成

训练总用时：26.6751秒
训练学习速率：0.005
训练迭代次数：100000
训练线程数：4
最终损失函数值：0.00494841

请输入待预测鸢尾草性状组数：
1
请输入鸢尾草性状矩阵(1行，4列)：
如：6.4 2.8 5.6 2.2（正确品类为2）
6.4 2.8 5.6 2.2
鸢尾草的品类为：
2
TIP:
0.山鸢尾
1.变色鸢尾
2.维吉尼亚鸢尾
请按任意键继续. . .
```

图 6 程序运行结果

6.2 结果分析与总结

输入的待预测数据是非训练数据，对比预测结果与真实结果，预测正确。此外，使用绘图程序绘制损失函数值变化情况如下：

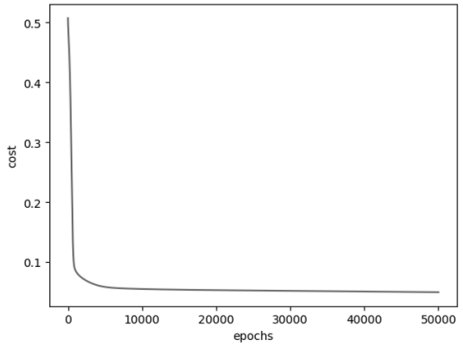

图 7 损失函数值变化情况

根据图形可知，该深度神经网络正确收敛。分析训练时间随训练迭代次数的变化情况，由于测试训练矩阵较小，为 4*100，故将训练线程数设为 1 以达到更快的速度。

表 3 训练时间随迭代次数的变化情况

迭代次数	100	1000	10000	20000	50000
训练时间	0.01	0.09	0.74	1.56	3.69

由以上测试数据绘制折线图如下：

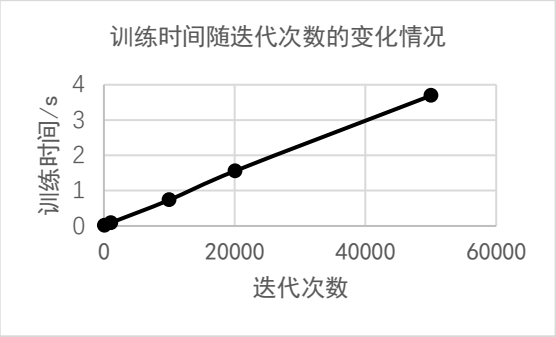

图 8 训练时间随迭代次数的变化情况

由折线图可以清楚地看出，训练时间随迭代次数的增加线性上升。综合图 4 与图 5 可知，若想同时兼顾训练速度与训练准确率，迭代次数为 100000 次左右时最佳。

7 附程序

见附件。

参 考 文 献

[1] 孙志军,薛磊,许阳明,王正.深度学习研究综述[J].计算机应用研究,2012,29(08):2806-2810.

[2] 刘颖超,张纪元.梯度下降法[J].南京理工大学学报(自然科学版),1993(02):12-16+22.

[3] 陈国良. 并行计算[M]. 北京：高等教育出版社，2011.

[4] 陈国良. 并行算法实践[M] . 北京：高等教育出版社，2004.

[5] 吴元斌, 熊江, 陈晓峰. 《大学计算机基础》实验教学的问题及改革[J]. 实验科学与技术, 2014, 12(2):84-86.

[6] 吴婧瑾, 吉家成. 美国伊利诺伊理工大学《并行处理》实验教学探析[J]. 实验科学与技术, 2013, 11(6):312-313.

[7] 汤小丹，梁红兵，哲凤屏，汤子瀛. 计算机操作系统（第三版）[M]. 西安：西安电子科技大学出版社，2007.

[8] 李学干. 计算机系统结构（第五版）[M]. 西安：西安电子科技大学出版社，2011.

[9] 殷顺昌. OpenMP 并行程序性能分析[D].国防科学技术大学,2006.

[10] 游佐勇. OpenMP 并行编程模型与性能优化方法的研究及应用[D].成都理工大学,2011.