南京理工大学
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

Nanjing University of Science and Technology

School of Computer Science and Engineering

《高性能计算引论》第四次作业

姓名：罗文水

学号：*918106840738*

班级：计科一班

课程：高性能计算引论

授课教师：李翔宇

2021 年 7 月 8 日

1 问题一

(1)

1. 在每个时钟周期内执行的操作数量不同，在 AMD Opteron 64-bit processor 中，每个时钟周期最多执行 3 次运算操作，而在全定制的 ASIC 中，每个时钟周期可以执行 700 次运算操作，弥补了 ASIC 时钟周期数少的劣势。

2.ASIC 在每个时钟周期内所做的操作大致相同，根据 ASIC 的设计可以使得运算更加高效，而 AMD Opteron 64-bit processor 是通过指令让其执行操作，无法完全开发处理器的并行度，其中有很多资源被闲置。

(2)

1.GPU 一般的架构是 SIMT，即单指令多线程架构，在计算如下指令的时候，GPU 的方法是使用多个线程进行并行计算，其中每个线程计算的内容是 y[i] += a * x[i]; 和 sum += z[i] * z[i]; 一共需要计算的次数为 $\lceil \frac{n}{thread\ count} \rceil$。在 SIMD 架构中，假设向量寄存器的长度为 m，即每次都能从内存中取出长度为 m 的元素进行运算。则每个向量运算中，每个计算单元执行的是 y[i] += a * x[i]; 和 sum += z[i] * z[i]; 在一个时钟周期内执行完长度为 m 的向量运算。

2. 在共享内存方面两者存在不同，对于 SIMD 架构，在一次向量运算中，内存是寄存器级别的共享，首先由微指令给出取数的起始地址以及向量长度，并行取出数据之后多个运算单元并行计算，然后将计算结果并行写回内存。而对于 GPU 架构，内存是线程级别的共享，所有处理器单元都能够访问全局存储器，单个线程内部，执行向量并行计算时有一个共享存储器，各个并发线程之间无法访问彼此的共享存储器。

```
1    sum = 0.0;
2    for (i = 0; i < n; i++){
3        y[i] += a * x[i];
4        sum += z[i] * z[i];
5    }
```

2 问题二

(1)

对于 n 输入 $\mathbf{x} \in \mathbf{R}^n$，1 输出的感知机，建立如下数学模型进行判别输出，即通过对输入进行加权以及偏置，经过符号函数进行输出，其中参数结构为 $\omega \in$

$\mathbf{R}^{n\times 1}$ $\theta \in \mathbf{R}$。

$$sign(\mathbf{x}^T\omega+\theta)=\begin{cases}+1\ ,\ \mathbf{x}^T\omega+\theta\geq 0\\ 0\ ,\ \mathbf{x}^T\omega+\theta<0\end{cases} \tag{1}$$

(2) 求解该问题等价于求解一个线性规划问题，根据问题（1）建立如下函数进行判别输出。根据感知机的迭代算法，设定初值可以得到该问题的一个解：$\omega_1=-2.0\ ,\omega_2=-0.5\ ,\theta=0$。

$$f(x_1,x_2)=\begin{cases}+1\ ,\ x_1\cdot\omega_1+x_2\cdot\omega_2\geq\theta\\ 0\ ,\ x_1\cdot\omega_1+x_2\cdot\omega_2<\theta\end{cases} \tag{2}$$

(3) 通过 matlab 如下代码进行计算以及查看结果：

```
x = [-0.5,-0.5,+0.3,-0.1;-0.5,+0.5,-0.5,+1.0];
t = [1,1,0,0];
net = perceptron;
net = train(net,x,t);
view(net);
bias = net.b;
display(bias);
weight = net.IW;
display(weight{1,1});
```

得到参数 $\omega_1=-2.0\ ,\omega_2=-0.5\ ,\theta=0$。网络的显示和分类的混淆矩阵如下图所示：

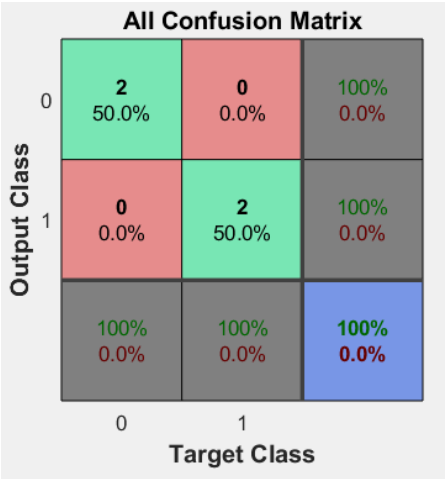

图 1: 分类数据上的混淆矩阵

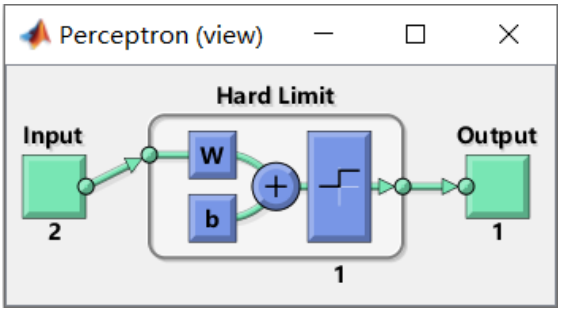

图 2: 神经网络参数结构

通过可视化方法，可以将感知机分类的分割线绘制在图形中，得到如下的结果：

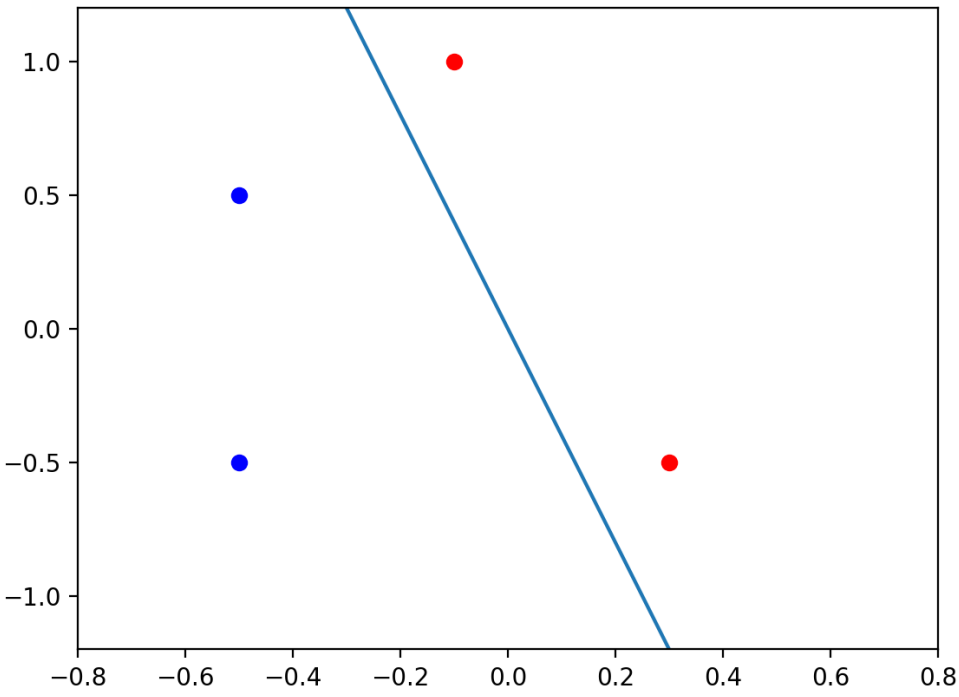

图 3: 分类情况