文章编号：2096-1472(2016)-04-08-02

C语言中浮点数的表示范围浅析

田 祎，樊景博
（商洛学院经济与管理学院，陕西 商洛 726000）

摘 要：浮点数是C语言中的一种数据类型，但在标准C中并没有给出其具体的描述，即数的存储格式及表示范围。部分经典的C语言程序设计教程中给出了浮点数的表示范围，但存在不严谨和值得商榷的地方。结合IEEE754标准，就C语言中浮点数内在存储格式进行分析并给出结论。

关键词：C语言；浮点数；表示范围
中图分类号：TP313 **文献标识码**：A

C Language Floating-point Number of Represent the Range and Analysis

TIAN Yi,FAN Jingbo
(*School of Economics & Management,Shangluo University,Shangluo* 726000,*China*)

Abstract: The float is a data type in C language,but its in standard C and did not give a specific of description:that is the number of storage format and the scope of representation.Part of classically C language programming tutorial gives a range of floating-point represent,but there is not rigorous and need to discussion.Combined IEEE754 standards,provided analyzation and conclusions in C language in internal storage floating-point format.

Keywords:C language;floating-point;scope of representation

1 引言(Introduction)

浮点数运算是科学计算必须面对的问题，由于计算机内部本身不能精确地处理某些整数或小数，因此在运算时可能存在较大的误差，运算结果将直接影响到系统的可靠性和安全性等。C语言因功能强大、程序设计灵活且支持底层应用，在科学计算、数据处理等领域中得到了广泛应用，但C语言在浮点数运算方面也存在数据表示的不精确性等问题。经典C语言并没有对浮点数专门说明，国内很多教材虽述及浮点数，但也只是给出表示范围，对于浮点数的解释尚不够充分，描述尚不够严谨，因此学生在对浮点数的学习过程中经常存在这样或那样理解上的困惑。这里就浮点数的表示范围，结合IEEE754做进一步的分析，为以后浮点数教学和学习给出参考[1]。

2 浮点数的表示及范围(The range of floating-point and representation)

总体而言，浮点数的表示形式一般格式指满足一般的二进制数机器码(包括定点整数和定点小数)的规定规则；而IEEE754[2]格式则在一般格式上进一步做了一些约定，以便表示数时比较方便和高效。

(1)一般表示法

其主要有两种格式，分别如图1和图2所示。

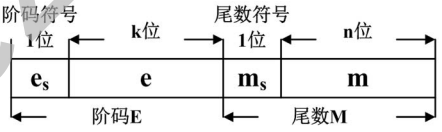

图1 浮点数表示形式1

Fig.1 Floating point representation 1

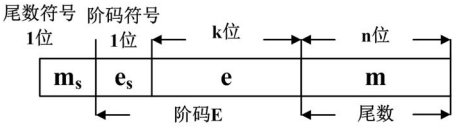

图2 浮点数表示形式2

Fig.2 Floating point representation 2

一般浮点数尾数采用纯小数(隐含位为0)来表示，即尾数M与定点小数表示方法相同，由于尾数的符号位决定整个浮点数的符号，故有时采用图2的形式；当尾数真值为0(不论阶码何值)，或阶码的值比能在机器中表示的最小值还小，计算机把该浮点数看成零值，称为机器零，即浮点数表示不了真值绝对值很接近0的数，只能看成0处理；尾数通常用原码或补码表示，阶码一般用移码或补码表示。因此其表示范围如图3所示。

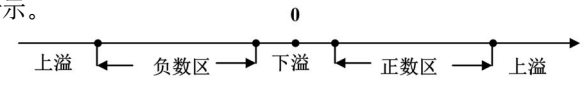

图3 浮点数的表示范围

Fig.3 Range of floating point numbers

基金项目：商洛学院教改项目(15jyjx135).

最大正数=最大正尾数×$2^{最大阶码}$

最小正数=最小正尾数×$2^{最小阶码}$

最大负数=最大负尾数×$2^{最小阶码}$

最小负数=最小负尾数×$2^{最大阶码}$

3 C语言中浮点数的表示(C language representation of floating point numbers)

C语言所使用的浮点数符合IEEE754标准，该标准在1985年审核通过，目的是让遵守IEEE标准的机器之间运行的程序可以相互直接移植，另外也让程序员可以轻松写出有用的、鲁棒的浮点数应用程序。

3.1 IEEE754

IEEE标准从逻辑上用三元组{S,E,M}表示一个数N[3]，如图4所示。

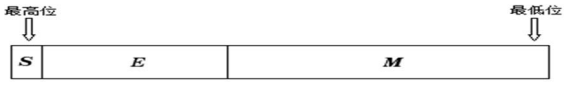

图4 IEEE标准的表示形式

Fig.4 Representation form of IEEE standard

N的实际值n由下式表示：

$$n=(-1)^s\times m\times 2^e$$

IEEE标准754规定了三种浮点数格式：单精度、双精度、扩展精度，分别对应C语言里头的float、double和long double。不同的编译系统对long double型的处理方法不同，Turbo c分配16个字节，而Visual C++6.0则分配8个字节。

单精度:N共32位，其中S占1位、E占8位、M占23位，如图5所示。

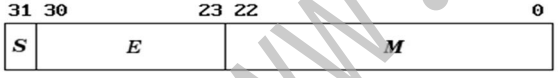

图5 单精度的表示形式

Fig.5 Single precision representation

双精度:N共64位，其中S占1位，E占11位，M占52位，如图6所示。

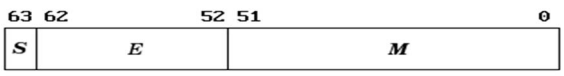

图6 双精度表示形式

Fig.6 Double precision representation

M虽然是23位或者52位，但它们只是表示小数点之后的二进制位数，也就是说，假定 M为“010110011...”，在二进制数值上其实是“.010110011...”。而事实上，标准规定小数点左边还有一个隐含位[4]，绝大多数情况下是1，当N对应的n非常小的时候，比如小于2^(−126)(32位单精度浮点数)，于是M对应的m最后结果可能是“m=1.010110011...”或者“m=0.010110011...”。

3.2 杨路明先生教材中对实数类型的描述

杨路明先生在其主编的《C语言程序设计教程》是这样描述实数类型的：实数类型的数据即实型数据，在C语言中实型数据又被称为浮点型数据[1]。实型数据的值域在计算机中表示只是数学中实数的一个子集。Turbo C的实型数据又分为单精度型和双精度型两种，它们所占内存字节数及取值范围如表1所示。

表1 Turbo C支持的实型数据

Tab.1 Turbo C to support the real data

关键字	字节数	取值范围	精度(位)
Float	4	约$\pm 3.4\times 10^{\pm 38}$	7
double	8	约$\pm 1.4\times 10^{\pm 308}$	15

3.3 存在问题

根据以上描述，单精度数的取值范围大约在−1038—1038；双精度数的取值范围大约−10308—10308，这个表述本身是没有问题的，但为了有利于基础教学，为学生建立一个正确的概念[5]，认为以上表述不够精确。因为尾数隐藏位值可能为1或者0，因此存在最小可以规格化的数。如果按标准规定隐藏位值为1，以上表述的取值范围就过于笼统，而且从最小可规格化的数到0的表示之间，也没有任何形式的过渡。比如最小规格化的数再小一点的数，便只能是0了，所以这种笼统的表示一方面不利于学生理解；另一方面，不能使学生真正明白为什么不能进行浮点数判等。

4 结论(Conclusion)

浮点数的表示范围与阶码和尾数的位数以及采用的浮点数表示格式有关。IEEE754标准中，单精度数所能表示的最大正规格化数，其阶码和尾数的值分别为$(11111110)_b$，$(111\ 1111\ 1111\ 1111\ 1111\ 1111)_b$，该数二进制数值为$1.(23个1)\times 2^{127}$，而能表示的最小正规格化数，其阶码和尾数部分的二进制值分别为1和0，该数二进制数值为$1.(23个0)\times 2^{-126}$。同理可得，双精度所能表示的最大正规格化数和最小正规格化数，其二进制数值分别为$1.(51个1)\times 2^{1023}$和$1.(51个0)\times 2^{1022}$。因此，在基础课程教学中，应将其表示如表2所示。

表2 Turbo C支持的实型数据

Tab.2 Turbo C to support the real data

关键字	有效数字(二进制)	最小正规格化数	最大正规格化数	有效数字(十进制)
Float	24	$1.175\cdots\times 10^{-38}$	$3.402\cdots\times 10^{38}$	6~9
double	53	$2.225\cdots\times 10^{-308}$	$1.797\cdots\times 10^{308}$	15~17

由表2学习者可清楚看到浮点数的表示范围，并可得到两个问题，一是0如何表示，是否有正负0之分；二是在最小规格化数到0之间的数如何表示。

0的偏移指数为00…00b，有效数字段亦为00…00b。0的偏移指数是保留的，也就是说0的偏移数不能用来表示正常的实数。并且，只有0做除数时，0才有正负之分，否则正0和负0没有区别。

当N对应的n非常小的时候，在最小规格化数到0之间，称其为微小数，比如小于2^(-126)(32位单精度浮点数)，于是M对应的m最后结果是m=0.010110011...，它的有效数字的最高位为0，这种表示为非规格化表示，引起精度丢失，但有效扩展了能表示的非常小数的范围。

《C程序设计》是理工类专业学生步入计算机程序设计世界的第一门课程[5]，建立科学严谨的计算机认识观尤为重要，而杨路明先生主编的《C语言程序设计教程》是C语言教学的经典教材，在诸如以上讨论的方面有值得商榷的地方，应该给予必要的论述，促使学生深入理解计算机的内部世界[6]，为学生步入计算机世界打下一个良好的基础。

参考文献(References)

[1] 杨路明.C语言程序设计教程[M].北京:北京邮电大学出版社,2008.

[2] 唐朔飞.计算机组成原理(第2版)[M].北京:高等教育出版社,2011.

[3] 朱亚超.基于IEEE754的浮点数存储格式分析研究[J].计算与信息技术,2010,8(10):1207-1280.

[4] IEEE Standard for Binary Floating-Point Arithmetic.ANSI/IEEE Standard 754-1985.Institute of Electrical and Electronics Engineers,August 1985.

[5] 田祎,樊景博.计算机程序设计语言类课程整合教学探讨[J].商洛学院学报,2012,4(26):28-30.

[6] 田祎.项目教学法在计算机语言实验教学中的应用[J].商洛学院学报,2010,5(24):91-93.

作者简介：

田　祎(1983-)，男，硕士，讲师.研究领域：计算机应用技术.

樊景博(1966-)，男，本科，教授.研究领域：数据库.

(上接第12页)

4 仿真与验证(Simulation and verification)

为了衡量该设计的模型准确性，本文设计了对比试验，针对同样的数据，使用本文设计模型和经典的K-means算法分别进行对比试验。试验所用到的数据环境详见表1。

表1 试验数据

Tab.1 Experimental data

试验次数	标记正常序列	未标记	
		正常序列	异常序列
1	200	20	50
2	200	20	100
3	200	20	150
4	200	20	200
5	200	20	250

对上表中的数据分别进行本文设计的聚类算法模型分析以及传统的K-means算法分析，其检测成功率如图2所示。

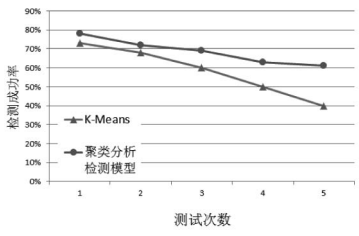

图2 检测成功率

Fig.2 Detection accuracy

5 结论(Conclusion)

随着互联网的发展和数据量的快速增长，传统的入侵检测技术已经不能满足如今高速发展的网络安全的需求，本文设计了一直基于聚类分析的入侵检测模型，给出了模型的工作流程，设计思想，和实现过程，最后，设计了对比仿真实验，结果表明本文设计的检测模型能够有效抵抗异常攻击，具备一定的实用价值。

参考文献(References)

[1] 张鹏,赵辉.关于入侵检测模型的研究与分析[J].网络安全技术与应用,2009(03):6-8.

[2] 喻莉,罗宁.基于机器学习的入侵检测模型[J].信息安全与通信保密,2005(03):112-114.

[3] Richard Lippmann,et al.Robert Cunningham.Evaluating and Strengthening Enterprise Network Security Using Attack Graphs[R].MIT Lincoln Laboratory Report,2005.

[4] 高宜楠.基于机器学习与人工免疫的入侵检测系统研究[D].西安电子科技大学,2010.

[5] 陈海,丁邦旭,王炜立.基于神经网络LMBP算法的入侵检测方法[J].计算机应用与软件,2007(08):183-185;188.

[6] Wenke Lee,et al.A data mining framework for building intrusion detection models[C].Proceedings of the 2007IEEE.

作者简介：

付明柏(1967-)，男，硕士，副教授.研究领域：软件理论，软件工程.