

个人资料



访问： 728584次
积分： 8894
等级： 5
排名： 第1124名

原创： 137篇 转载： 98篇
译文： 119篇 评论： 259条

文章搜索

文章分类

- IOI2006竞赛题 (9)
- LSI与PCB设计 (6)
- 人工智能和机器思维 (1)
- 图形图像算法 (7)
- 射影几何入门(连载) (13)
- 微软101道面试题 (1)
- 数理逻辑和数学基础 (92)
- 游戏编程 (6)
- 神经网络入门(连载) (6)
- 遗传算法入门(连载) (10)
- JAVA图形编程入门(连载) (9)
- 吴润昭私院 (1)
- 嘉兴新塍 (0)
- 党史陈列馆 (0)
- 我们的老家 (0)
- 程序设计自动化 (2)
- 算术算子的编译 (1)
- 4地址机器 (1)
- 53CM (1)
- 人工智能 (6)
- 二进制 (1)
- 十进制 (1)
- 轮盘赌 (1)
- 进化 (2)
- 变异 (1)

神经网络入门(连载之五)

标签： 网络 vector 算法 generation output

2006-01-11 18:24 21083人阅读 评论(7) 举报

分类： 神经网络入门(连载) (5)

游戏编程中的人工智能技术

神经网络入门
(连载之五)

4.5 神经网络的编码 (Encoding the Network)

在本书的开始几章中，你已经看到过怎样用各种各样的方法为遗传算法编码。但当时我并没有向你介绍过一个用实数编码的具体例子，因为我知道我要留在这里向你介绍。我曾经讲到，为了设计一个前馈型神经网络，编码是很容易的。我们从左到右读每一层神经细胞的权重，读完第一个隐藏层，再向上读它的下一层，把所读到的数据依次保存到一个向量中，这样就实现了网络的编码。因此，如果我们有图14所示的网络，则它的权重编码向量将为：

0.3, -0.8, -0.2, 0.6, 0.1, -0.1, 0.4, 0.5

在这一网络中，为了简单，我没有把偏移值的权重包括进去。但在实际实现编码时，你必须包含偏移值这个权重，否则你肯定无法获得你所需要的结果。

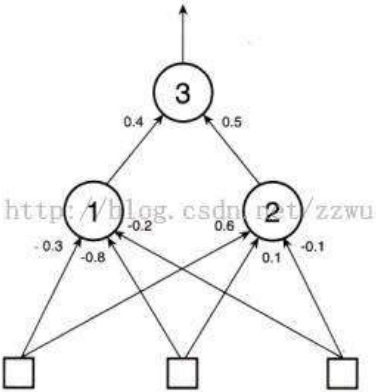


图14 为权重编码。

在此之前讲的事情你都懂了吗？好极了，那下面就让我们转来考虑，怎样用遗传算法来操纵已编码的基因吧。

- 旋转轮 (1)
- Roulette Wheel (0)
- 走迷宫 (1)
- 实无穷 (1)
- 潜在可实现性 (1)
- 马尔科夫<算法论> (1)
- 数理逻辑 (4)
- 数学基础 (1)
- 算法 (1)
- 游戏中的的人工智能技术 (1)
- 勘误 (1)
- 清华大学出版社 (1)
- c语言 (1)
- pascal语言 (1)
- 集成开发环境 (1)
- 中文操作系统 (1)
- IDE (1)
- 分子图形 (7)
- MOL4D (4)
- 图形算法 (1)
- z缓冲法 (2)
- 画家算法 (2)
- 稳压器 (17)
- 稳压电源 (17)
- 负反馈 (1)
- 调整 (1)
- dosbox (1)
- dos程序 (1)
- dos编程 (1)
- Pascal线 (1)
- Steiner point (1)
- Plucker line (1)
- 计算机应用能力 (1)
- 上机考试 (1)
- 模拟软件 (1)
- 微波法检测 (1)
- 微波无损检测 (1)
- NDT (1)
- 微波应用 (1)
- 微波成像 (0)
- 何燮侯 (1)
- 何燮侯后代 (1)
- 免费电子图书 (1)
- 网上图书 (1)
- 外文书 (1)
- 科技书 (0)
- modern geometry (1)
- 现代几何 (1)
- 隐藏层 (1)
- 扫雷机 (1)
- 输入输出 (1)
- 神童·疯子 (1)
- 北医 (1)
- 赵炳文 (1)
- 同班同桌 (1)
- ATX开关电源电路图 (1)
- Godel完全定理 (1)
- Godel完备定理 (0)
- 哥德尔完全定理 (0)
- 哥德尔完备定理的证明 (0)
- Kirkman point (1)
- 北大 (1)
- 生物分类 (1)
- hp1000笔记本 (1)
- 分区 (1)

4.6 遗传算法（The Genetic Algorithm）

到此，所有的权重已经象二进制编码的基因组那样，形成了一个串，我们就可以象本书早先讨论过的那样来应用遗传算法了。遗传算法（GA）是在扫雷机已被允许按照用户指定的帧数（为了某种缘故，我下面更喜欢将帧数称作滴答数，英文是ticks）运转后执行的。你可以在ini文件中找到这个滴答数（iNumTicks）的设置。下面是基因组结构体的代码。这些对于你应该是十分面熟的东西了。

```
Struct SGenome
{
    vector <double>    vecWeights;

    double            dFitness;

    SGenome():dFitness(0) {}

    SGenome(vector <double> w, double f):vecWeights(w),dFitness(f){}

    //重载 '<' 的排序方法
    friend bool operator<(const SGenome& lhs, const SGenome& rhs)
    {
        return (lhs.dFitness < rhs.dFitness);
    }
};
```

从上面的代码你可看出，这一SGenome结构和我们在本书所有其他地方见到的SGenome结构几乎完全一致，唯一的差别就是这里的染色体是一个双精度向量std::vector。因此，可以和通常一样来应用杂交操作和选择操作。但突变操作则稍微有些不同，这里的权重值是用一个最大值为dMaxPerturbation的随机数来骚扰的。这一参数dMaxPerturbation在ini文件中已作了声明。另外，作为浮点数遗传算法，突变率也被设定得更有些。在本工程中，它被设定成为0.1。下面就是扫雷机工程遗传算法类中所见到的突变函数的形式：

```
void CGenAlg::Mutate(vector<double> &chromo)
{
    // 遍历权重向量，按突变率将每一个权重进行突变
    for (int i=0; i<chromo.size(); ++i)
    {
        // 我们要骚扰这个权重吗？
        if (RandFloat() < m_dMutationRate)
        {
            // 为权重增加或减小一个小的数量
            chromo[i] += (RandomClamped() * CParams::dMaxPerturbation);
        }
    }
}
```

如同以前的工程那样，我已为v1.0版本的Smart Minesweepers工程保留了一个非常简单的遗传算法。这样就能给你留下许多余地，可让你利用以前学到的技术来改进它。就象大多数别的工程一样，v1.0版只用轮盘赌方式选精英，并采用单点式杂交。

注意：当程序运行时，权重可以被演化成为任意的大小，它们不受任何形式的限制。

4.7 扫雷机类（The CMinesweeper Class）

这一个类用来定义一个扫雷机。就象上一章描述的登月艇类一样，扫雷机类中有一个包含了扫雷机位置、速度、以及如何转换方向等数据的纪录。类中还包含扫雷机的视线向量（look-at vector）；它的2个分量被用来作为神经网络的2个输入。这是一个规范化的向量，它是在每一帧中根据扫雷机本身的转动角度计算出来的，它指示了扫雷机当前是朝着哪一个方向，如图11所示。下面就是CMinesweeper扫雷机类的声

重装系统 (1)
win7 (1)
CAI (1)
CAI发展历史 (1)
马云 (2)
首富 (1)
传奇 (1)
FDISK模拟 (1)
Lehmer (1)
FOCR (1)
Deutche 困难问题 (1)
QiQu Channel Router (1)
崎岖通道最优布线 (1)
达沃斯 (1)
Pascal定理证明 (1)
Pascal定理的Pacal证明原稿 (1)
键盘vk值 (1)
LSI设计 (2)
PCB设计 (2)
FOCR的应用 (1)
快速最优通道布线 (1)
动画 (0)
gif动画不能动 (0)
RIBBONS (1)
蛋白质造型 (2)
顺势营养滴液 (1)
JAVA小程序编程 (1)
苏联哲学百科 (2)
编译自动化 (1)
VLSI布线 (1)
PCB布线 (1)
计算机应用能力上机考试 (1)
博客分类目录 (1)
MOL4DP (1)
MOL4DE (1)
zzwu博客 (1)
设计 (1)
JAVA:First Contact (1)
磁芯测试仪 (1)
Pascal Hexagrammum Mysticum(六角迷魂图) (1)
马尔采夫定理 (1)
无穷长公式 (1)
CSDN QA (1)
证明论 (1)
寻找计算机原理讲义 (1)
世界7大数学难题 (1)
Circular points at infinity (1)
新浪博客怎样写 (1)
Fields Medallists (1)
数学分支代码 (1)
AI分支 (1)
喀秋莎 (1)
多值逻辑 (1)
eknigu (1)
美国院士 (1)
国内外broks (1)
Delphi应用程序的调试 (1)
Delsphi的应用程序调试 (1)
分子图形软件介绍 (1)
我唱过的歌 (1)
U盘存储原理 (1)
日文中的汉字 (1)
哲学系长寿多 (1)
100强企业 (1)

明：

```
class CMinesweeper
{
private:
    // 扫雷机的神经网络
    CNeuralNet      m_ItsBrain;

    // 它在世界坐标里的位置
    SVector2D      m_vPosition;

    // 扫雷机面对的方向
    SVector2D      m_vLookAt;

    // 它的旋转(surprise surprise)
    double      m_dRotation;

    double      m_dSpeed;

    // 根据ANN保存输出
    double      m_lTrack,
                m_rTrack;

    m_lTrack和m_rTrack根据网络保存当前帧的输出。
    这些就是用来决定扫雷机的移动速率和转动角度的数值。

    // 用于度量扫雷机适应性的分数
    double      m_dFitness;

    每当扫雷机找到一个地雷，它的适应性分数就要增加。

    // 扫雷机画出来时的大小比例
    double      m_dScale;

    // 扫雷机最邻近地雷的下标位置
    int      m_iClosestMine;

    在控制器类CController中，有一个属于所有地雷的成员向量std::vector。而m_iClosestMine就是
    代表最靠近扫雷机的那个地雷在该向量中的位置的下标。

public:
    CMinesweeper();

    // 利用从扫雷机环境得到的信息来更新人工神经网络
    bool Update(vector<SVector2D> &mines);

    // 用来对扫雷机各个顶点进行变换，以便接着可以画它出来
    void WorldTransform(vector<SPoint> &sweeper);

    // 返回一个向量到最邻近的地雷
    SVector2D GetClosestMine(vector<SVector2D> &objects);

    // 检查扫雷机看它是否已经发现地雷
    int CheckForMine(vector<SVector2D> &mines, double size);

    void Reset();

    // ----- 定义各种供访问用的函数
    SVector2D Position()const { return m_vPosition; }
    void IncrementFitness(double val) { m_dFitness += val; }
    double Fitness()const { return m_dFitness; }
```

- 中国上网费问题 (1)
- MSG200 (1)
- 中国上市公司价值排名100强名单 (1)
- 什么是椭圆几何与双曲线？ (1)
- 数理逻辑在计算机科学中的地位 (0)
- ACM Computing Classification System (1)
- 应对当前股市危机的八点建议 (1)
- Free Fortran Compilers (1)
- markdown (1)
- a (0)
- 最伟大的俄罗斯人 (1)
- 外公与大外公 (1)
- cmos工艺 (0)
- 著译目录 (1)
- 来函照登 (1)
- cmos工艺流程 (1)
- 一阶谓词逻辑的几个系统 (1)
- 潘多拉病毒 (1)
- 我的设计 (0)
- 一阶谓词逻辑系统的一个扩充 (1)
- 透视：一点透视 (0)
- 我的著译目录 (0)
- 参军65年：一些老照片 (0)
- 一些老照片 (0)
- 回忆类杂文 (0)
- 老照片 (0)
- 快速最优通道布线算法 (1)
- 中国学术界腐败 (0)
- 学术界腐败 (1)
- 扯下中国院士的神秘面纱(二) (1)
- 软件收费种类 (1)
- 抗战胜利日 (1)
- 50项世界顶级科技 (1)
- 国际设计自动化会议 (1)
- 计算机辅助设计年会 (1)
- 绍兴一中建校百年史 (1)
- 9.18 (1)
- 流亡三部曲 (1)
- 饮食寡淡是老年痴呆的祸根！ (1)
- desargues定理 (1)
- EKNIGU真棒！ (0)
- 简单因子和简单向 (0)
- 人工智能游戏编程 (0)
- 抗美援朝64周年纪念：老照片 (1)
- 基本形 (1)
- "一大"代表包惠僧de离奇经历 (1)
- 托派 (0)
- 刘仁静 (1)
- 中共一大代表李汉俊生平简介 (0)
- 能量与物质 (0)
- 抗战时期为何汉奸多 (0)
- 默克尔赠送中国桑叶地图 (0)
- 一大代表15人 (0)
- 能量的正负 (0)
- 负能量 (1)
- Puphotoshop用于照片整理 (0)
- 一大代表15人的图片处理 (1)
- 博客排名 (1)
- 李达在中共“一大”前后 (1)

```
void PutWeights(vector<double> &w) { m_ItsBrain.PutWeights(w); }

int GetNumberOfWeights() const
{ return m_ItsBrain.GetNumberOfWeights(); }

};

4.7.1 The CMinesweeper::Update Function（扫雷机更新函数）

需要更详细地向你说明的CMinesweeper类的方法只有一个，这就是Update更新函数。该函数在每一帧中都要被调用，以更新扫雷机神经网络。让我们考察这函数的肚子里有些什么货色：

bool CMinesweeper::Update(vector<SVector2D> &mines)
{
    //这一向量用来存放神经网络所有的输入
    vector<double> inputs;

    //计算从扫雷机到与其最接近的地雷（2个点）之间的向量
    SVector2D vClosestMine = GetClosestMine(mines);

    //将该向量规范化
    Vec2DNormalize(vClosestMine);

    首先，该函数计算了扫雷机到与其最靠近的地雷之间的向量，然后使它规范化。（记住，向量规范化后它的长度等于1。）但扫雷机的视线向量（look-at vector）这时不需要再作规范化，因为它的长度已经等于1了。由于两个向量都有效地化成了同样的大小范围，我们就可以认为输入已经是标准化了，这我前面已讲过了。

    //加入扫雷机->最近地雷之间的向量
    Inputs.push_back(vClosestMine.x);
    Inputs.push_back(vClosestMine.y);

    //加入扫雷机的视线向量
    Inputs.push_back(m_vLookAt.x);
    Inputs.push_back(m_vLookAt.y);

    //更新大脑，并从网络得到输出
    vector<double> output = m_ItsBrain.Update(inputs);

    然后再把视线向量、以及扫雷机与它最接近的地雷之间的向量，都输入到神经网络。函数CNeuralNet::Update利用这些信息来更新扫雷机网络，并返回一个std::vector向量作为输出。

    //保证在输出的计算中没有发生错误
    if (output.size() < CParams::iNumOutputs)
    {
        return false;
    }

    // 把输出赋值到扫雷机的左、右轮轨
    m_lTrack = output[0];
    m_rTrack = output[1];

    在更新神经网络时，当检测到确实没有错误时，程序把输出赋给m_lTrack和m_rTrack。 这些值代表施加到扫雷机左、右履带轮轨上的力。

    // 计算驾驶的力
    double RotForce = m_lTrack - m_rTrack;

    // 进行左转或右转
    Clamp(RotForce, -CParams::dMaxTurnRate, CParams::dMaxTurnRate);
```

- 托派分子 (0)
- 中国托派陈独秀 (1)
- 一大代表15人简单介绍 (1)
- 尼可尔斯基 (1)
- 一大代表15人的结局 (1)
- 歌德下载 (1)
- Manifest der Kommunistischen Partei (0)
- 一大代表竟有三个在黄埔军校当过官 (1)
- 初到俄罗斯 (0)

文章存档

- 2016年02月 (2)
- 2016年01月 (6)
- 2015年12月 (6)
- 2015年11月 (6)
- 2015年10月 (3)

展开

阅读排行

- 神经网络入门(连载之一) (97449)
- 神经网络入门(连载之二) (45246)
- 神经网络入门(连载之三) (34875)
- 神经网络入门(连载之四) (26882)
- 遗传算法入门(连载之一) (21753)
- 神经网络入门(连载之五) (21077)
- 遗传算法入门(连载之四) (18373)
- 神经网络入门(连载之六) (17881)
- 遗传算法入门(连载之二) (15456)
- 遗传算法入门(连载之十) (14698)

评论排行

- 神经网络入门(连载之一) (27)
- 遗传算法入门(连载之十) (26)
- 神经网络入门(连载之二) (24)
- 遗传算法入门(连载之一) (20)
- 神经网络入门(连载之四) (16)
- 神经网络入门(连载之三) (12)
- 遗传算法入门(连载之四) (11)
- 遗传算法入门(连载之五) (10)
- 遗传算法入门(连载之九) (10)
- 神经网络入门(连载之五) (7)

最新评论

- 我在CSDN论坛的提问与回复
zzwu: 1楼 密斯大白: 您好! 谢谢您的好意, 我因身体原因, 已不可能参加"CSDN老友记"答谢...
初到俄罗斯
zzwu: 物价那么贵, 不知工资有多少?
神经网络入门(连载之四)
x249271850: // 计算权重*输入的乘积的总和。 netinput += m_vecLayers.m_vecNe...
神经网络入门(连载之四)
a_big_pig: 与9楼同问

m_dSpeed = (m_lTrack + m_rTrack);

扫雷机车的转动是利用施加到它左、右轮轨上的力之差来计算的。并规定，施加到左轨道上的力减去施加到右轨道上的力，就得到扫雷机车辆的转动力。然后就把此力施加给扫雷机车，使它实行不超过ini文件所规定的最大转动率的转动。而扫雷机车的行进速度不过就是它的左侧轮轨速度与它的右侧轮轨速度的和。既然我们知道了扫雷机的转动力和速度，它的位置和偏转角度也就都能更新了。

//更新扫雷机左右转向的角度
m_dRotation += RotForce;

// 更新视线角度
m_vLookAt.x = -sin(m_dRotation);
m_vLookAt.y = cos(m_dRotation);

// 更新它的位置
m_vPosition += (m_vLookAt* m_dSpeed);

// 如果扫雷机到达窗体四周，则让它实行环绕，使它不至于离开窗体而消失
If (m_vPosition.x > CParams::WindowWidth) m_vPosition.x = 0;
If (m_vPosition.x < 0) m_vPosition.x = CParams::WindowWidth;
If (m_vPosition.y > CParams::WindowHeight) m_vPosition.y = 0;
If (m_vPosition.y < 0) m_vPosition.y = CParams::WindowHeight;

为了使事情尽可能简单，我已让扫雷机在碰到窗体边框时就环绕折回(wrap)。采用这种方法程序就不再需要做任何碰撞-响应方面的工作。环绕一块空地打转对我们人来说是一桩非常不可思议的动作，但对扫雷机，这就像池塘中的鸭子。

Return true;
}

4.8 CController Class（控制器类）

CController类是和一切都有联系的类。图15指出了其他的各个类和CController类的关系。下面就是这个类的定义：

class CController
{

private:

// 基因组群体的动态存储器（一个向量）
vector<SGenome> m_vecThePopulation;

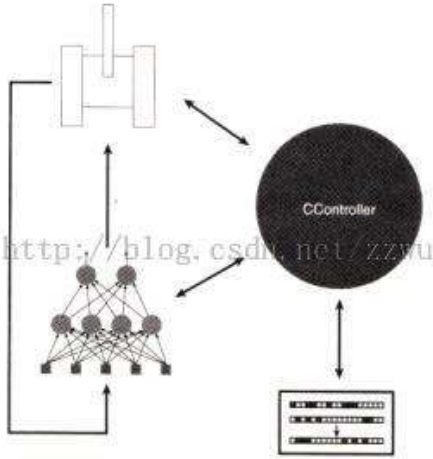


图15 minesweeper工程的程序流程图

// 保存扫雷机的向量

遗传算法入门(连载之十) 神经网络入门
a_big_pig: 3ks

一大代表15人的图片修理

zzwu: 欢迎大家指正和补充，特别是美术爱好者。

射影几何入门（连载一） - 1-1对1
baidu_33307963: 我的意思是如果两个无穷小 如果不是等比例缩小的话 是否还能建立一一映射关系？

射影几何入门（连载一） - 1-1对1
baidu_33307963: 我有一个地方始终想不清楚，希望能得到回复，无穷直线对应一条线段，我觉得如果从微分角度来考虑，是无法建...

神经网络入门(连载之四)

evolone: 楼主大大，下载网址已经不能用了，能不能把工程的代码发给我一份，小弟毕业论文就是神经网络的题目，所以想...

我的博客（总目录）

ershjiu: 向默默奉献的吴老师致敬！

```
vector<CMinesweeper> m_vecSweepers;

// 保存地雷的向量
vector<SVector2D> m_vecMines;

// 指向遗传算法对象的指针
CGenAIg* m_pGA;

int m_NumSweepers;

int m_NumMines;

// 神经网络中使用的权重值的总数
int m_NumWeightsInNN;

// 存放扫雷机形状各顶点的缓冲区
vector<SPoint> m_SweeperVB;

// 存放地雷形状各顶点的缓冲区
vector<SPoint> m_MineVB;

// 存放每一代的平均适应性分数，供绘图用
vector<double> m_vecAvFitness;

// 存放每一代的最高适应性分
vector<double> m_vecBestFitness;

// 我们使用的各种不同类型的画笔
HPEN m_RedPen;
HPEN m_BluePen;
HPEN m_GreenPen;
HPEN m_OldPen;

// 应用程序窗口的句柄
HWND m_hwndMain;

// 切换扫雷程序运行的速度
bool m_bFastRender;

// 每一代的帧数（滴答数）
int m_iTicks;

// 代的计数
int m_iGenerations;

// 窗体客户区的大小
int cxClient, cyClient;

// 本函数在运行过程中画出具有平均-, 和最优适应性值的图
void PlotStats(HDC surface);

public:

    CController(HWND hwndMain);

    ~CController();

    void Render(HDC surface);

    void WorldTransform(vector<SPoint> &VBuffer,
                        SVector2D vPos);
```



```

bool    Update();

// 几个公用的访问方法

bool    FastRender() { return m_bFastRender; }

void    FastRender(bool arg) { m_bFastRender = arg; }

void    FastRenderToggle() { m_bFastRender = !m_bFastRender; }

};

```

当创建CController类的某个实例时，会有一系列的事情发生：

- *创建CMinesweeper对象。
- *统计神经网络中所使用的权重的总数，然后此数字即被用来初始化遗传算法类的一个实例。
- *从遗传算法对象中随机提取染色体(权重)并(利用细心的脑外科手术)插入到扫雷机的神经网络中。
- *创建了大量的地雷并被随机地散播到各地。
- *为绘图函数创建了所有需要用到的GDI画笔。
- *为扫雷机和地雷的形状创建了顶点缓冲区。

所有的一切现都已完成初始化，由此Update方法就能在每一帧中被调用来对扫雷机进行演化。

4.8.1 CController::Update Method（控制器的更新方法）

控制器更新方法CController::Update方法（或函数）在每一帧中都要被调用。当调用update函数时，函数的前一半通过对所有扫雷机进行循环，如发现某一扫雷机找到了地雷，就update该扫雷机的适应性分数。由于m_vecThePopulation包含了所有基因组的拷贝，相关的适应性分数也要在这时进行调整。如果为完成一个代（generation）所需要的帧数均已通过，本方法就执行一个遗传算法时代(epoch)来产生新一代的权重。这些权重被用来代替扫雷机神经网络中原有的旧的权重，使扫雷机的每一个参数被重新设置，从而为进入新一generation做好准备。

```

bool CController::Update()
{
    // 扫雷机运行总数为CParams::iNumTicks次的循环。在此循环周期中，扫雷机的神经网络
    // 不断利用它周围特有的环境信息进行更新。而从神经网络得到的输出，使扫雷机实现所需的
    // 动作。如果扫雷机遇见了一个地雷，则它的适应性将相应地被更新，且同样地更新了它对应
    // 基因组的适应性。

    if (m_iTicks++ < CParams::iNumTicks)
    {
        for (int i=0; i<m_NumSweepers; ++i)
        {
            //更新神经网络和位置

            if (!m_vecSweepers[i].Update(m_vecMines))
            {
                //处理神经网络时出现了错误，显示错误后退出

                MessageBox(m_hwndMain, "Wrong amount of NN inputs!",
                    "Error", MB_OK);

                return false;
            }

            // 检查这一扫雷机是否已经发现地雷

            int GrabHit = m_vecSweepers[i].CheckForMine(m_vecMines,
                CParams::dMineScale);

            if (GrabHit >= 0)
            {
                // 扫雷机已找到了地雷，所以要增加它的适应性分数
                m_vecSweepers[i].IncrementFitness();
            }
        }
    }
}

```

```
// 去掉被扫雷机找到的地雷，用在随机位置放置的一个新地雷来代替
m_vecMines[GrabHit] = SVector2D(RandFloat() * cxClient,
                                  RandFloat() * cyClient);

}

// 更新基因组的适应性值
m_vecThePopulation[i].dFitness = m_vecSweepers[i].Fitness();
}

}

// 一个代已被完成了。
// 进入运行遗传算法并用新的神经网络更新扫雷机的时期
else
{
    // 更新用在我们状态窗口中状态
    m_vecAvFitness.push_back(m_pGA->AverageFitness());
    m_vecBestFitness.push_back(m_pGA->BestFitness());

    // 增加代计数器的值
    ++m_iGenerations;

    // 将帧计数器复位
    m_iTicks = 0;

    // 运行GA创建一个新的群体
    m_vecThePopulation = m_pGA->Epoch(m_vecThePopulation);

    // 在各扫雷机中从新插入新的（有希望）被改进的大脑
    // 并将它们的位置进行复位，等
    for(int i=0; i<m_NumSweepers; ++i)
        {m_vecSweepers[i].m_ItsBrain.PutWeights(m_vecThePopulation[i].vecWeights);

        m_vecSweepers[i].Reset();
        }
}

return true;
}
```

概括起来，程序为每一世代做的工作是：

- 1. 为所有扫雷机和为iNumTicks个帧组织循环，调用Update函数并根据情况增加扫雷机适应值的得分。
- 2. 从扫雷机神经网络提取权重向量。
- 3. 用遗传算法去演化出一个新的网络权重群体。
- 4. 把新的权重插入到扫雷机神经网络。
- 5. 转到第1步进行重复，直到获得理想性能时为止。

最后，表3列出了Smart Sweepers工程 v1.0版所有缺省参数的设置值。

表3 Smart Sweepers v1.0工程的缺省设置

神经网络	
参 数	设 置 值
输入数目	4
输出数目	2
隐藏层数目	1

隐藏层神经元数目	1
隐藏层数目	10
激励响应	1
遗 传 算 法	
参 数	设 置 值
群体大小	30
选择类型	旋转轮
杂交类型	单点
杂交率	0.7
突变率	0.1
精英设置（on/off）	on
精英数目	1/4
总 体 特 性	
参 数	设 置 值
每时代的帧数目	2000

4.9 运行此程序 (Running the Program)

当你运行程序时，“F”键用来切换2种不同的显示状态，一种是显示扫雷机怎样学习寻找地雷，一种是示在运行期中产生的最优的与平均的适当性分数的统计图表。 当显示图表时，程序将会加速运行。

顶

2

踩

1

上一篇

神经网络入门(连载之四)

下一篇

神经网络入门(连载之六)

我的同类文章

神经网络入门(连载) (5)

主题推荐

color

神经网络

游戏编程

人工智能

微软

编码

猜你在找

查看评论

7楼 zzwu 2015-09-23 19:33发表



谢谢各位！但也要再次说明：这要归功于原书作者MAT先生的细腻的笔法，我只是照翻一下。
(译者)

6楼 KittyMonkey 2015-09-06 12:02发表



还没看完，不过要情不自禁的点赞了！

5楼 大信 2014-09-08 14:59发表



能这样一步一步做出详细解释的书，比那些包含天花乱坠的语言的书要来的是在和实用，赞

4楼 ccstrike 2014-03-04 16:07发表



一直潜水，从未发过言的我看了这篇，不得不对作者喊一声赞。译者翻译的也非常好。

3楼 -jack- 2012-10-31 08:58发表



佩服佩服！越看越觉得作者厉害！

2楼 [游客] 2012-07-02 10:34发表



谢谢！这要归功于原书作者MAT先生的细腻的笔法，我只是照翻一下。
(译者)

1楼 wenfengbism 2012-02-07 18:32发表

写的非常好

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop
- AWS
- 移动游戏
- Java
- Android
- iOS
- Swift
- 智能硬件
- Docker
- OpenStack
- VPN
- Spark
- ERP
- IE10
- Eclipse
- CRM
- JavaScript
- 数据库
- Ubuntu
- NFC
- WAP
- jQuery
- BI
- HTML5
- Spring
- Apache
- .NET
- API
- HTML
- SDK
- IIS
- Fedora
- XML
- LBS
- Unity
- Splashtop
- UML
- components
- Windows Mobile
- Rails
- QEMU
- KDE
- Cassandra
- CloudStack
- FTC
- coremail
- OPhone
- CouchBase
- 云计算
- iOS6
- Rackspace
- Web App
- SpringSide
- Maemo
- Compuware
- 大数据
- aptech
- Perl
- Tornado
- Ruby
- Hibernate
- ThinkPHP
- HBase
- Pure
- Solr
- Angular
- Cloud Foundry
- Redis
- Scala
- Django
- Bootstrap