

哈尔滨工业大学(深圳)

# 《数据结构》实验报告

---

## 实验四

### 图性结构及其应用

学 院: 计算机科学与技术

姓 名: 朱海峰

学 号: 190110716

专 业: 计算机科学与技术

日 期: 2020-05-08

## 一、问题分析

### 1.1 问题

问题一：

网络化的时代，谣言的传播顺延着关系网的飞速传播着。假设现在有  $N$  个节点，标记为 1 到  $N$ 。给定一个列表  $times$ ，表示一个谣言经过有向边的传递时间， $times[i] = (u, v, w)$ ，其中  $u$  是源节点， $v$  是目标节点， $w$  是一个谣言从源节点传递到目标节点的时间。假设当前节点  $K$  产生了一个谣言。请你来分析最少需要多少时间，整个网络将会充斥这个谣言。

问题二：

神经网络是一门重要的机器学习技术。简单起见，这里将神经网络简化成一个有向无环图  $G(V, E)$  模型，图的节点称为神经元。神经网络分为三层：输入层、隐藏层和输出层。输入层神经元无有向边连入，输出层神经元无有向边连出，其余神经元属于隐藏层。神经元与神经元之间由有向边连接，连接神经元  $v_i$  和  $v_j$  的有向边上带有系数  $W_{ij}$ 。每个输入层神经元含一个初始权值  $C_i = X_i$ ，则其他神经元权值通过公式计算：

$$C_i = \sum_{\langle i, j \rangle \in E} W_{ij} C_j$$

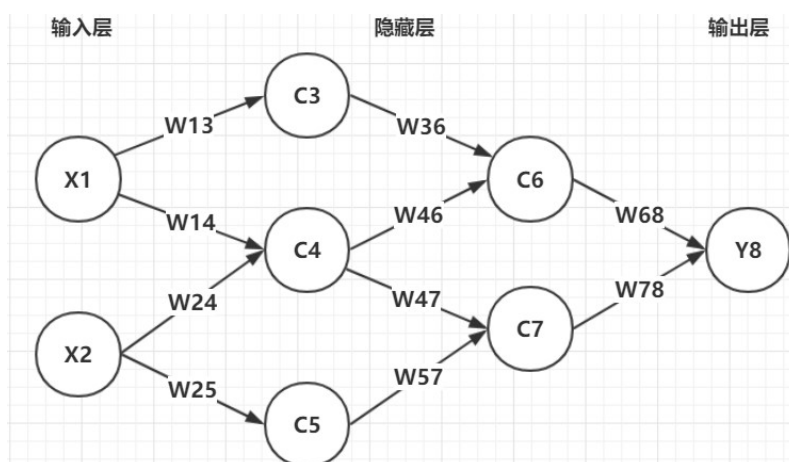


图 1

即每个神经元的权值等于所有有边指向它的神经元权值乘指向它的边系数的加和。最终，输出层神经元的输出等于输出层神经元的权值 $Y = C_n$ 。

我们把上述过程叫做正向传播。

现在，以邻接矩阵的形式给出神经网络的结构，并给出每个输入层神经元的输入，请完成正向传播算法，求出每个输出层神经元输出的权值 $Y_i$ 。

## 1.2 分析

对于问题一：

可通过输入的节点数，带权有向边构建一个带权有向图；从指定起点，用 Dijkstra 算法得到到达各个顶点的最短路径，并用数组存储；找到数组中最大值即为谣言完全充斥的最终时间。

对于问题二：

同问题一方法建立带权有向图；从输入层开始，逐个将求出的权的节点通过带权边，在边指向的目的神经元中加入贡献的权值，直到得到每个神经元的权值，存储在一个数组中；最后输出在输出层的神经元权值。

## 二、详细设计

### 2.1 设计思想

有向带权图的建立：

按照输入的顶点和边的个数，读入每个带权边，边以链式向前星的形式存储；用到了静态链表的思想，存储和操作上，兼具间距邻接矩阵和邻接表的优点，存储不浪费，操作不麻烦。

问题一：

问题一转化为找到出发节点到每个节点的最短路径问题，这类问题有 Dijkstra 和 floyd 两种算法，其中 Floyd 常用于找任意两点最短距离，故这里用 Dijkstra 算法，即：

1. 集合 S 的初值为  $S=\{1\}$ ；
2. D 为各顶点当前最小路径；

3. 从  $V-S$  中选择顶点  $w$ ，使  $D[w]$  的值最小；
4. 并将  $w$  加入集合  $S$ ，则  $w$  的最短路径已求出；
5. 调整其他各结点的当前最小路径， $D[k]=\min\{D[k], D[w]+C[w][k]\}$ ；
6. 直到  $S$  中包含所有顶点。

问题二：

问题二中的有向边均是小编号顶点指向大编号顶点，顺序性极强，用正向传播算法。只需按照编号顺序往每个顶点接出的有向边，在接入顶点中加上该边的贡献值，最终即可得到所有节点的权值。

## 2.2 存储结构及操作

### (1) 存储结构

1. 带权有向图，其中有向边以使用链式前向星方式存储：

```
struct Edge
{
    int start, end, weight; //该边的起点终点和权
    int next; //同一起点的下一条边以 next 作为链式寻找同一起点的所有边
};
struct Graph
{
    struct Edge edges[maxm]; //所有的边
    int head[maxn]; //head[i] = j, 则 i 为起点的第一条边为 edges[j]
    int tot; //总边数
};
```

2. 数组，在实验内容一和二中存储相关数据：

```
minTime[] //临时变量，存储各节点的当前最短时间
visited[] //标记个点是否访问过
ans []    //存储各点的权值
```

### (2) 涉及的操作

//初始化一个图

```
struct Graph *init_graph()
```

//加边

```

void add_edge(int start, int end, int weight, struct Graph *graph)
//访问图，打印每一条边
void visit_graph(int n, struct Graph *graph)
//计算到达所有节点的所需最少时间
int get_min_time(int n, int startpoint, struct Graph *graph)
//计算所有节点的权值，存储在数组中
void forward_propagation(int n_total, int n_input, int n_output, int *input_array,
struct Graph *graph)

```

## 2.3 程序整体流程

1. 整体流程，如下图 2 所示；
2. 建立图，如下图 3 所示；

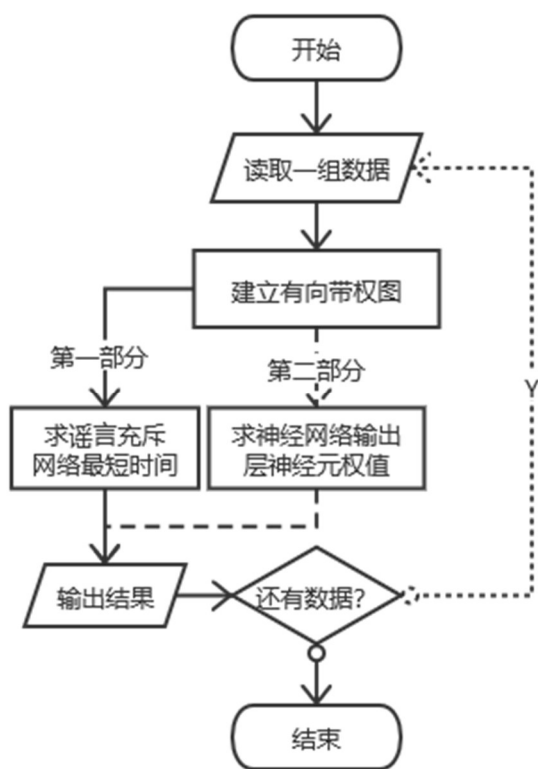


图 2

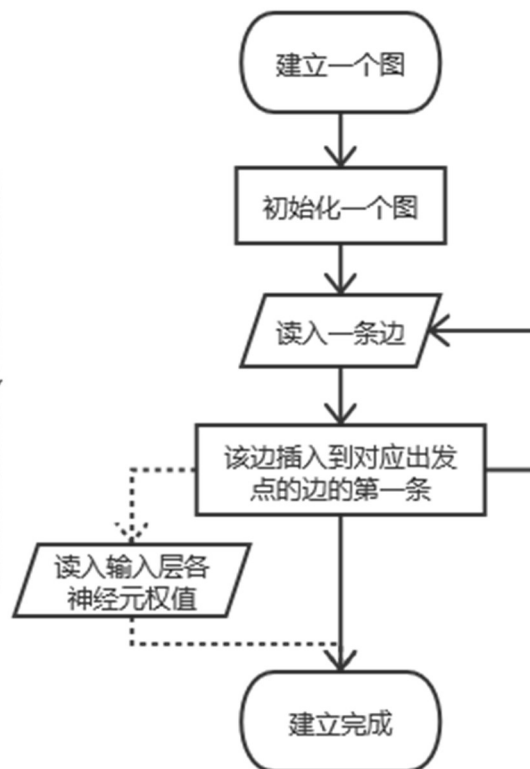


图 3

3. Dijkstra 算法，求单源最短路径，如下图 4 所示；
4. 正向传播算法，求输出层神经元权值，如下图 5 所示；

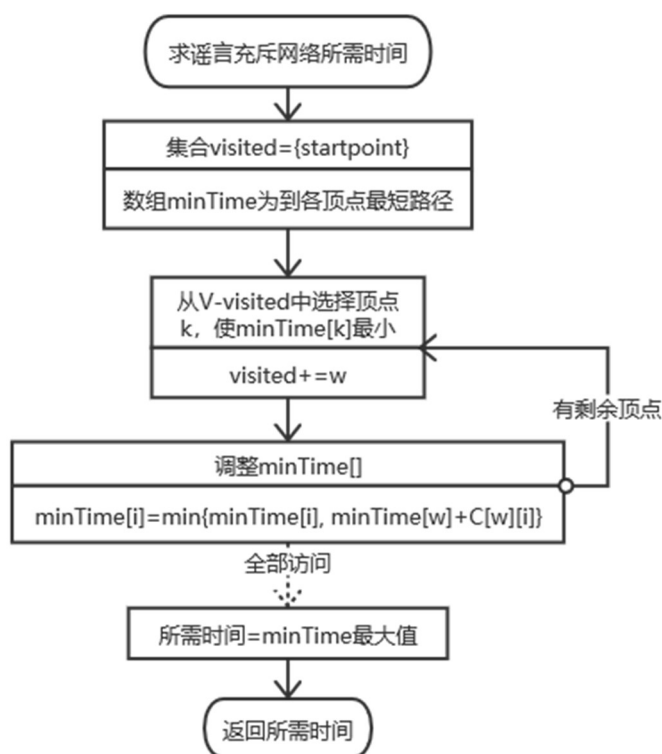


图 4

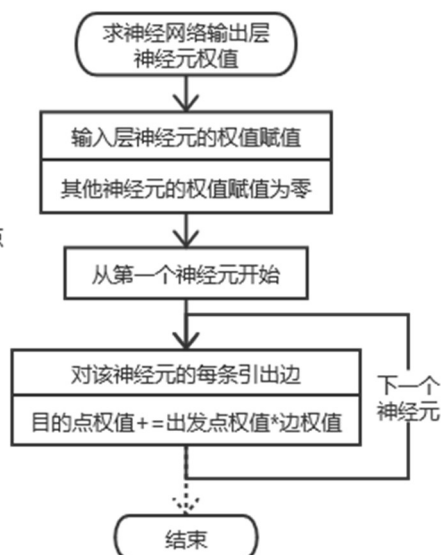


图 5

### 三、用户手册

欢迎使用本程序！

本程序分为求谣言充斥网络所需时间、求神经网络输出层神经元输出的权值两个部分。

#### 3.1 求谣言充斥网络所需时间

在求谣言充斥网络所需时间中，先在程序同目录文件“4\_1\_input.in”（文件中已有输入样例）中，输入图的数据。

其中，第一行包含三个整数 $N M S$ ，分别表示点的个数、有向边的个数、出发点的编号；

接下来 $M$ 行每行包含三个整数 $F_i G_i W_i$ ，分别表示第 $i$ 条有向边的出发点、目标点和传播时间。

如：

---

```

4 6 1
1 2 2
2 3 2
2 4 1
1 3 5
3 4 3
1 4 4

```

---

第一行的 4 代表此图有四个顶点，6 代表六条边，1 则是出发结点号；

第二行的 1 代表该边出发节点，第二个数 2 代表到达结点，最后一个数 2 为该边传播所需时间；

同理，第三行至第六行含义与第二行类似，均为边。

然后系统自动计算出从出发结点到达所有结点的最短时间，并在控制台输出，按照此次样例输入，输出结果为 4，即 4 个时间单位后谣言将充斥整个网络。

完成每组数据计算后，程序正常结束。

### 3.2 求神经网络输出层神经元权值

在求神经网络输出层神经元输出的权值程序中，先在程序同目录文件“4\_2\_input.in”（文件中已有输入样例）中，输入图的相关数据。

其中，第一行包含四个正整数  $N M P Q$ ，表示总神经元个数，总的边个数，输入层神经元个数以及输出层神经元个数<sup>1</sup>；

接下来  $M$  行，每三个正整数  $i j w_{ij}$ ，表示有一条从  $i$  号神经元指向  $j$  号神经元的有向边，边权为  $w_{ij}$ ；

接下来一行  $P$  个正整数，表示每个输入层神经元的输入值。

如：

---

```

8 10 2 1
1 3 2
1 4 1
2 4 3
2 5 3
3 6 4

```

---

<sup>1</sup> 1~ $P$  号节点为输入层神经元， $P+1\sim N-Q$  号节点为隐藏层神经元， $N-Q+1\sim N$  号节点为输出层神经元。

```
4 6 2
4 7 2
5 7 7
6 8 2
7 8 1
5 10
```

第一行的 8 代表此图有八个顶点，10 代表十条边，2 为输入层神经元个数，4 为输出层神经元个数；

第二行的 1 代表该边出发节点，3 代表到达结点，2 为该边权；

同理，第三行至第十行含义与第二行类似，均为边；

最后一行是输入层神经元的输入值。

然后系统从出发结点自动计算出所有神经元的权值，并在控制台输出输出层各个神经元权值，按照此次样例输入，输出结果为 500。

完成每组数据计算后，程序正常结束。

感谢您的使用！

欢迎提出宝贵意见！

## 四、结果

实验内容一、二结果如图 6、7 所示。

图 6





```
====case1=====
min time=4
====case2=====
min time=2147483630
====case3=====
min time=5
====case4=====
min time=2147483630
====case5=====
min time=6
====case6=====
min time=14
====case7=====
min time=0
====case8=====
min time=3
====case9=====
min time=3
====case10=====
min time=6
====case11=====
min time=2
====case12=====
min time=7
====case13=====
min time=3
====case14=====
min time=2
====case15=====
min time=2147483630
====case16=====
min time=1
====case17=====
min time=100

Process returned 0 (0x0)   execution time : 6.576 s
Press any key to continue.
_
```

图 7

## 五、总结

实验用到有向带权图的数据结构，其中图的边以链式向前星的形式储存。

实验涉及到 Dijkstra 算法，用以求单源最短路径；正向传播算法，用以求输

出层神经元权值。

此次实验中有的内容是之前没用到过的，实验中也在学理论。比如链式向前星是没有学习过的，但是我觉得它实用意义很大，体现了静态链表的思想，不仅像邻接表操作方便，又不会像邻接矩阵浪费空间；正向传播算法也是没有学过的，但是实现起来极其简单。

每次实验都是调试起来最费时间，这次可能是运气不好加上精神状态也不好，先是选择了实现较难的第一部分，然后写起来也是，每个部分都有 bug，在函数里边，单行调试直接就崩了，非要我一部分一部分来打印相关数据整了好久。在调试方法和经验上，也算是收获不小。