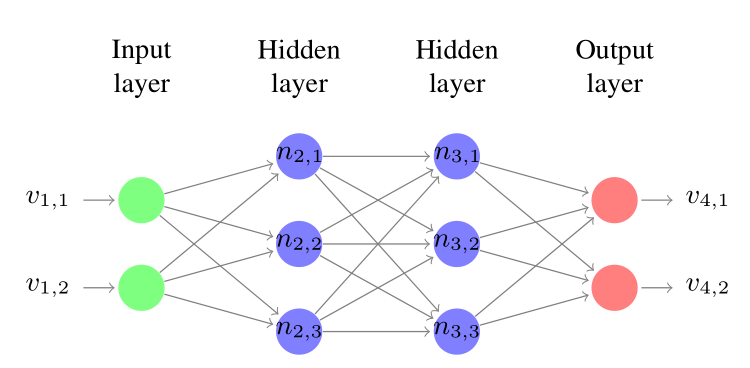
本次实验需要同学们设计指定覆盖准则的计算覆盖率的程序。我们会提供一个小规模的数据集供大家实现，需要大家设计出在给大规模数据集的时候只需要修改程序中的参数也可以计算出覆盖率的程序。

1. 背景介绍

本次课程设计的数据集是深度神经网络中各个神经元的权值。

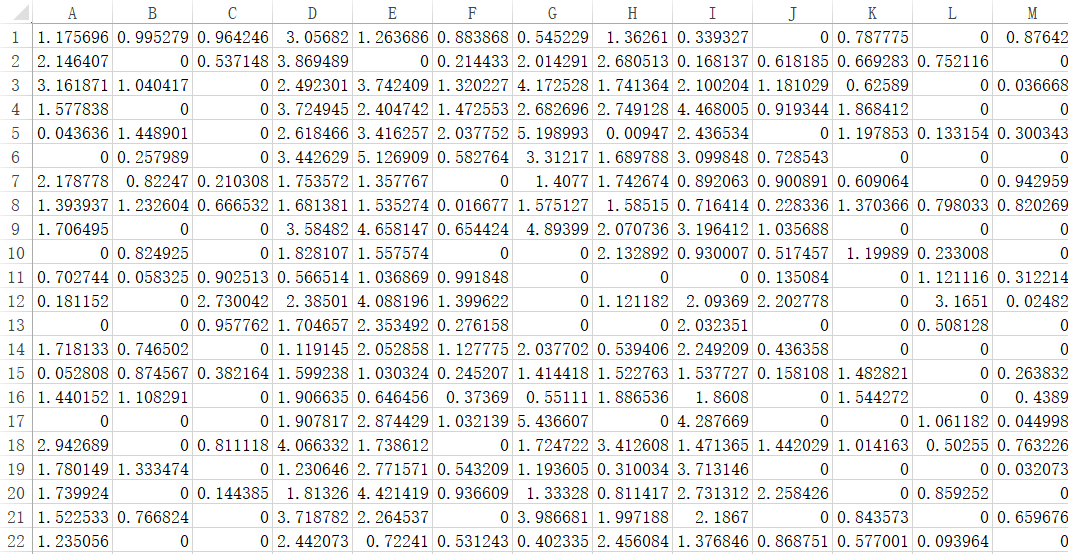
一个简单的神经网络如图：



其中最左边的是输入层，最右边的是输出层，中间为隐藏层。比如n2,1 n2,2 n2,3就是第一层隐藏层的三个神经元。**（重点理解层的概念，因为下文覆盖准则中有的考虑某一层中神经元的相互作用，有的考虑相邻层之间神经元的作用。）**本次课程设计**只考虑隐藏层**中神经元的覆盖率，探索隐藏层中神经元的作用。

1. 预备知识

**2.1 数据集详解**

****

上图是我们提供的数据集。其形式是50行 X 120列的数据。

其中每一行代表一个测试用例，一共有50个测试用例。 120列代表有**120个神经元**。其中前50列是第一层的神经元，中间40列是第二层的神经元，最后30列是第三层的神经元。举例，表中A1的值1.175696代表在第一个测试用例下，第一层第一个神经元的权值。再比如第一行第51列的数，就是在第一个测试用例下，第二层第一个神经元的权值。其他同理。**（每个测试用例即为一个输入，下文中用X1,X2…表示）**

简单的说，每一行就是**一个测试用例**下，各个神经元的权值。

每一列，就是**一个神经元**在不同的测试用例下的权值。

**2.2 计算覆盖率**

神经网络上的覆盖率与普通程序的覆盖率计算方法大同小异。总的来说就是计算在所有测试用例的输入下，被覆盖到的对象占对象总个数的占比（本次课程设计的对象为神经元个数、神经元对的个数、神经元的状态、神经元的组合）。其一般公式可以表达为：

覆盖率=

1. 覆盖准则

本次课程设计共有11个测试准则。每个准则介绍分别如下：

1. **神经元覆盖率**

我们认为神经元权值大于0时，这个神经元是激活的。如果权值小于等于0，那么这个神经元是未激活的。神经元覆盖准则是计算有多少神经元在某一个测试用例的情况下曾经激活过。即对某个神经元而言，只要在某一测试用例情况下，神经元的权值大于0（不可等于0），那么这个神经元就是活跃的神经元。

神经元覆盖率计算公式：

神经元覆盖率=

1. **K段神经元覆盖率**

除了2.1的数据集，我们还提供了“边界值”的数据集，该数据集只有两行，有120列。对应2.1数据集的每个神经元的上边界值和下边界值。现在对每一个神经元而言，可以将神经元的权值[对应的下边界，对应的上边界]分成N等份。接着计算每个神经元的N等份被测试用例覆盖到了多少份**（只要某一测试用例下该神经元的权值在某一份中，就算该份被覆盖到了，被多个测试用例覆盖不叠加。）**最后把所有神经元覆盖的份数相加，除以神经元个数乘以N，就是该覆盖率。**（本次实验，取N=50）**

K段神经元覆盖率计算公式：

K段神经元覆盖率=

1. **top-k神经元覆盖率**

**本覆盖率需要铭记前文提到的“层”的概念。**

Top-k神经元覆盖率是计算有多少神经元曾经表现为它的层中最活跃的前k个神经元。 所谓活跃，就是权值大。所以本覆盖率就是计算有多少神经元曾经在某一测试用例中表现为它的层中权值前k大的神经元。设计程序时,k分别取1,2,3。

注意事项：（1）、找权值前k大的神经元时，需要对三层分别找。比如当k取1时，需要在前50列找最大的，51到90列找最大的，91到120列找最大的。记住层的概念。

（2）、如果某一个神经元多次表现为前k活跃，仅算一个神经元不可重复计算。

（3）、K作为一个参数，可取1,2,3，验收时需要k的三个值的不同的覆盖率。

Top-k神经元覆盖率计算公式：

Top-k神经元覆盖率=

1. **TKNP**

理论基础同上文top-k神经元覆盖率。不过需要计算的对象不同。(请先阅读第三个测试准则相关介绍)

TKNP需要计算不同的测试用例下，前K活跃的神经元组合数量。

举例：令k值为2，那么需要统计每一层曾经变现为前2活跃的神经元组合。假如现在的神经网络为第一层神经元为n1,n2,n3，第二层神经元为n4,n5,n6， 第三层神经元为n7,n8,n9.

在输入为X1时，每一层最活跃的前2个神经元为(n1,n2) (n4,n5)(n7,n8)

在输入为X2时，每一层最活跃的前2个神经元为(n1,n3) (n4,n6)(n7,n9)

在输入为X3时，每一层最活跃的前2个神经元为(n1,n2) (n4,n5)(n7,n8)

那么在该测试用例集下，TKNP=2（因为第一个和第三个组合重复了）

其公式为：

TKNP= 每一层前k活跃的神经元组合的个数 **（TKNP不是覆盖率）**

设计程序时，需取k=1,2,3

1. **边界覆盖率**

除了2.1的数据集，我们还提供了“边界值”的数据集，该数据集只有两行，有120列。对应2.1数据集的每个神经元的上边界值和下边界值（本测试准则只用到第一行的上边界值）。边界覆盖率是计算有多少神经元在某一测试用例下，表现的权值大于了上边界值。只要在某一测试用例下权值比该神经元的边界值大，那么该神经元就是上边界神经元。

边界覆盖率的公式为：

边界覆盖率=

1. **组合稀疏覆盖率**

介绍该覆盖率之前需要先搞懂组合的概念：

T维组合，即维度为t的组合。比如某一层神经元有n1,n2,n3,n4.

那么该层神经元的2维组合有6个：

(n1,n2)(n1,n3)(n1,n4)(n2,n3)(n2,n4)(n3,n4)

该层神经元的3维组合有4个：

(n1,n2,n3)(n1,n2,n4)(n1,n3,n4)(n2,n3,n4)

该层神经元的4维组合有1个：

(n1,n2,n3,n4)

在实际计算中，我们只取同一层内神经元的t维组合。比如，我们令t=2，

数据集中前50列为第一层，中间40列为第二层，最后30列为第三层。那么一共有组合。当t=3时，只需要把公式中2换为3即可。

在本覆盖率中，我们认为神经元权值大于0时，神经元为激活状态，记为1，神经元权值小于等于0时，神经元为未激活状态，记为0。

那么对每一个神经元而言，都应该有（0，1）两种状态。

对组合而言，比如（n1,n2）的组合一共应该有（0,1）(0,0)(1,0)(1,1)四种状态。

组合稀疏覆盖率，就是计算有多少组合的全部状态都被测试用例覆盖到了。

组合稀疏覆盖率公式为：

组合稀疏覆盖率=

注意事项：1、上述举例只举了t=2的例子，实验需要实现t=2,3,4三种情况下的覆盖率计算。

2、当t=3时，组合会有8种状态，t=4时，组合会有16种状态。只有当8种或者16种状态都被覆盖到，该组合才算被覆盖到了。

**3、选取组合时，只能在单层之内取，不可以跨层。.**

1. **组合密集覆盖率**

介绍该覆盖率之前需要先搞懂组合的概念：

T维组合，即维度为t的组合。比如某一层神经元有n1,n2,n3,n4.

那么该层神经元的2维组合有6个：

(n1,n2)(n1,n3)(n1,n4)(n2,n3)(n2,n4)(n3,n4)

该层神经元的3维组合有4个：

(n1,n2,n3)(n1,n2,n4)(n1,n3,n4)(n2,n3,n4)

该层神经元的4维组合有1个：

(n1,n2,n3,n4)

在实际计算中，我们只取同一层内神经元的t维组合。比如，我们令t=2，

数据集中前50列为第一层，中间40列为第二层，最后30列为第三层。那么一共有组合。当t=3时，只需要把公式中2换为3即可。

在本覆盖率中，我们认为神经元权值大于0时，神经元为激活状态，记为1，神经元权值小于等于0时，神经元为未激活状态，记为0。

那么对每一个神经元而言，都应该有（0，1）两种状态。

对组合而言，比如（n1,n2）的组合一共应该有（0,1）(0,0)(1,0)(1,1)四种状态。

组合密集覆盖率就是计算所有的组合的所有的状态中，有多少状态被测试用例覆盖到了。

组合密集覆盖率的公式为：

组合密集覆盖率=

注意事项：1、上述举例只举了t=2的例子，实验需要实现t=2,3,4三种情况下的覆盖率计算。

**2、选取组合时，只能在单层之内取，不可以跨层。.**

3、公式分母为组合个数乘以2^t，因为在t维时，每个组合会有2^t个状态。

**8-11覆盖准则的同学请详读下面这段介绍内容：**

8-11覆盖准则均是研究前一层神经元对后一层某一个神经元的单独影响。每个前一层的神经元与**相邻的**后一层的神经元组成一个**神经元对。**前层的神经元称为**条件神经元**，后层的神经元称为**决策神经元**。举例，如下图1，该神经网络的神经元对一共有12个，分别是(n1,n4) (n1,n5) (n2,n4) (n2,n5) (n3,n4) (n3,n5)(n4,n6)(n4,n7)(n4,n8)(n5,n6)(n5,n7)(n5,n8)。其中，对神经元对(n1,n4)来说，n1是条件神经元，n4是决策神经元。

**强调：神经元对一定是相邻层的神经元对，不可跨层。比如（n1,n6）就不可以作为一个神经元对。**

8-11四个覆盖准则的公式可以统一定义为：

**满足条件的神经元对/神经元对的总个数**

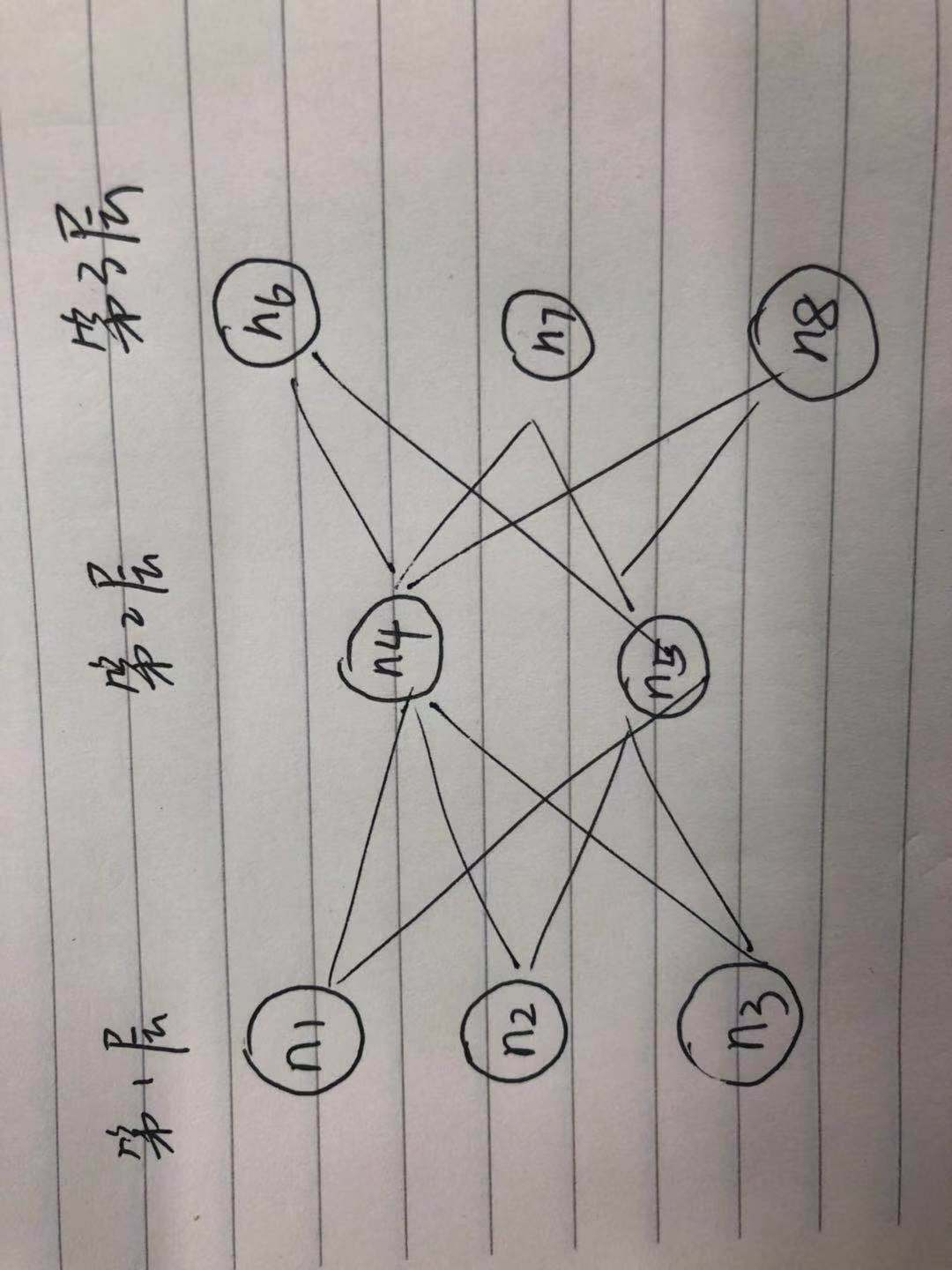


图1：简单的神经网络模型

1. **Sign-sign cover**

请先阅读第八个覆盖准则前的介绍内容。

该覆盖准则的神经元对需要满足以下三个条件：

（1）、存在两个输入X1,X2，当输入分别为X1和X2时，条件神经元的**符号改变**。

（2）、当输入分别为X1,X2时，条件神经元的那一层神经元中，除了条件神经元其他所有神经元的**符号都没有改变**。

（3）、当输入分别为X1,X2时，决策神经元的**符号改变**。

该覆盖准则的公式为：

Sign-sign coverage=

注意事项：

我们认为，当神经元权值大于0时，符号为正，小于等于0时，符号为负，所谓符号改变，就是正负性产生了改变。

1. **Distance-sign cover**

请先阅读第八个覆盖准则前的介绍内容。

该覆盖准则的神经元对需要满足以下三个条件：

（1）、存在两个输入X1,X2，当输入分别为X1和X2时，前一层的神经元**距离足够小**。

距离足够小的判断标准如下：

假设当输入为X1时，该层神经元的权值集合为U(X1)

当输入为X2时，该层神经元的权值的集合为U（X2）

当||U(X1)-U(X2)||≤5时，称一层神经元的距离足够小。

（2）、当输入为X1，X2时，条件神经元的这一层所有神经元的符号**均未改变。**

（3）、当输入为X1,X2时，决策神经元的符号**产生了改变**。

.该覆盖准则的公式为：

Distance-sign coverage=

注意事项：

1. 我们认为，当神经元权值大于0时，符号为正，小于等于0时，符号为负，所谓符号改变，就是正负性产生了改变。
2. 判断距离足够小的公式为L1范数距离公式。
3. **Sign-value cover**

请先阅读第八个覆盖准则前的介绍内容。

该覆盖准则的神经元对需要满足以下三个条件：

（1）、存在两个输入X1,X2，当输入分别为X1和X2时，条件神经元的**符号改变**。

（2）、当输入分别为X1,X2时，条件神经元的那一层神经元中，除了条件神经元其他所有神经元的**符号都没有改变**。

（3）、当输入分别为X1,X2时，决策神经元的**值改变的足够大**且**符号未变。**

判断值改变的足够大的标准为：

假设当输入为X1时，决策神经元的值为V1。

当输入为X2时，决策神经元的值为V2。

如果V1/V2＞2或者V1/V2＜0.5

那么称决策神经元的值改变的足够大。

该覆盖准则的公式为：

Sign-value coverage=

注意事项：

1. 我们认为，当神经元权值大于0时，符号为正，小于等于0时，符号为负，所谓符号改变，就是正负性产生了改变。
2. 由于V1,V2不知道谁大谁小，所以比值大于2或者小于0.5都算距离足够大。
3. **Distance-value cover**

请先阅读第八个覆盖准则前的介绍内容。

该覆盖准则的神经元对需要满足以下三个条件：

（1）、存在两个输入X1,X2，当输入分别为X1和X2时，前一层的神经元**距离足够小**。

距离足够小的判断标准如下：

假设当输入为X1时，该层神经元的权值集合为U(X1)

当输入为X2时，该层神经元的权值的集合为U（X2）

当||U(X1)-U(X2)||≤5时，称一层神经元的距离足够小。

（2）、当输入为X1，X2时，条件神经元的这一层所有神经元的符号**均未改变。**

（3）、当输入为X1,X2时，决策神经元的**值改变足够大**且**符号未改变**。

判断值改变的足够大的标准为：

假设当输入为X1时，决策神经元的值为V1。

当输入为X2时，决策神经元的值为V2。

如果V1/V2＞5或者V1/V2＜0.2

那么称决策神经元的值改变的足够大。

该覆盖准则的公式为：

Distance-value coverage=

注意事项：

1. 我们认为，当神经元权值大于0时，符号为正，小于等于0时，符号为负，所谓符号改变，就是正负性产生了改变。
2. 由于V1,V2不知道谁大谁小，所以比值大于5或者小于0.2都算距离足够大。
3. 判断距离足够小的公式为**L1范数距离公式**。

四、验收标准

1、本次课程设计，输入集只有50个，如果扩大输入集，需要程序同样可以运行。

2、需要确保获得正确的覆盖率结果。

3、算法优化得较好、运行时间较短的程序可以加分。