第四讲 人工神经元的连接

本讲主要内容

- 神经元的联结综述
- 生物神经元的联接
- 神经元的扩展（宽度和深度）

  宽度：以多输出的单层感知器为例

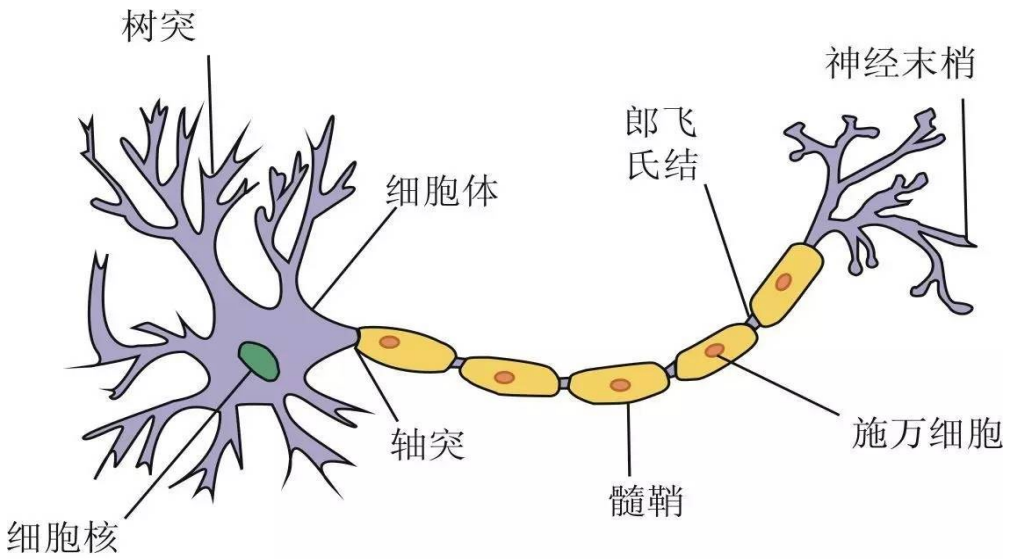
  深度：多层感知器
- 其他连接方式

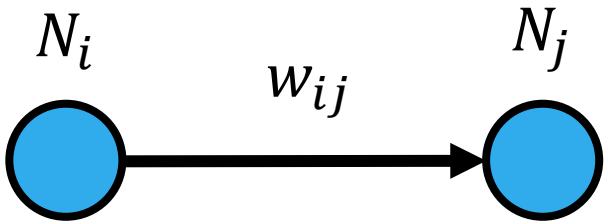
一、神经元的联结综述

连接：神经元之间传递信息的方式

- 单个神经元
  - 计算能力有限
  - 虽然能够解决一部分问题，但是在实际生活中是远远不够的
- “人多力量大”——使用多个神经元共同解决问题
  - 大脑皮质中成千上万的神经元组成了神经系统
- 神经元通过连接交流信息，共同完成计算任务

连接的拓扑表示

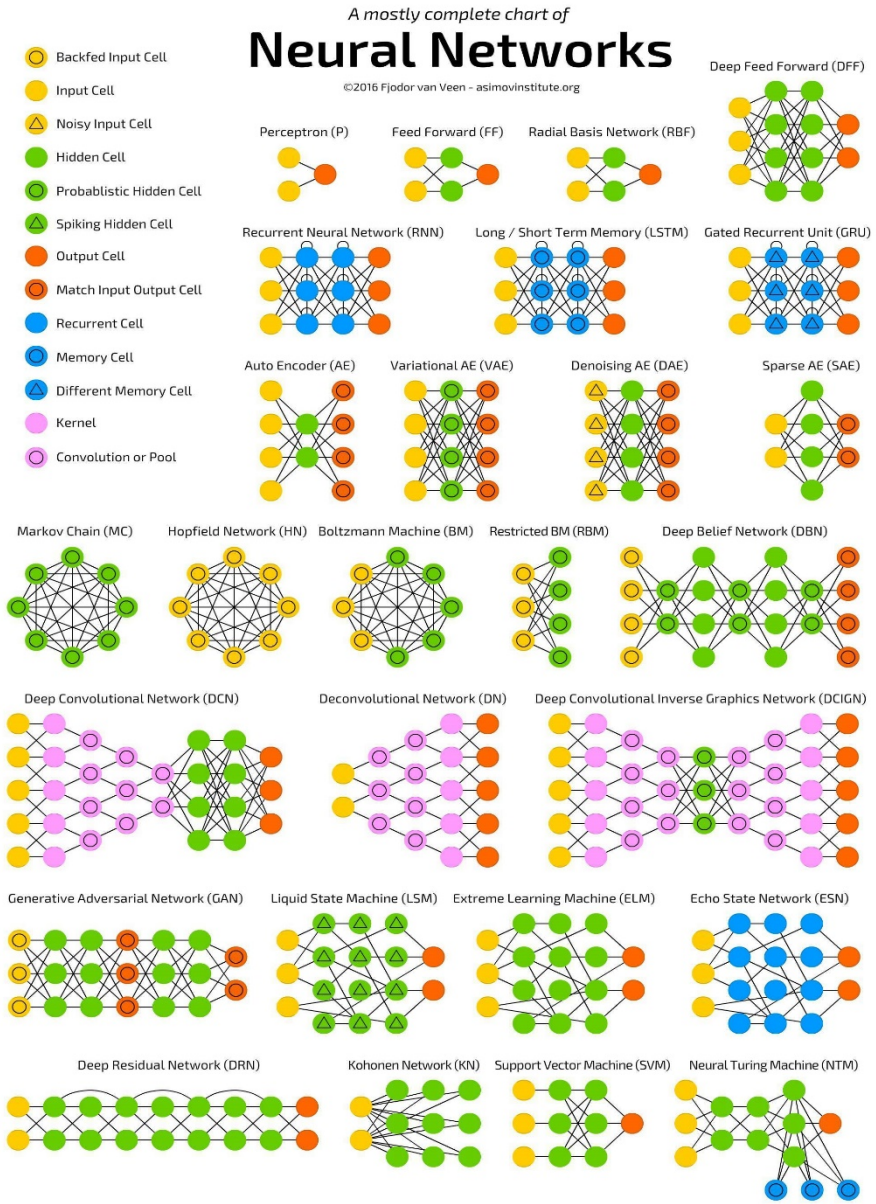




- 用正号（“+”，可省略）表示传送来的信号起刺激作用，它用于增加神经元的活跃度；
- 用负号（“-”）表示传送来的信号起抑制作用，它用于降低神经元的活跃度。

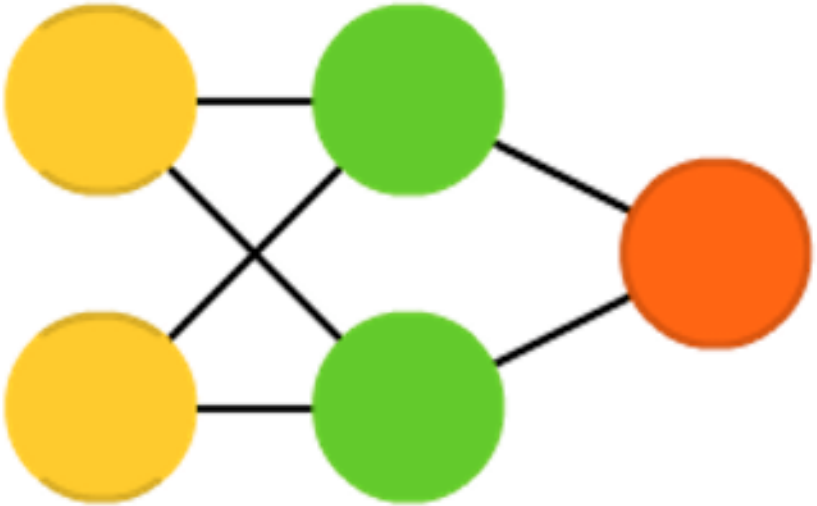
神经元的连接方式

- 神经元根据不同的连接方式，组成不同功能的神经网络。

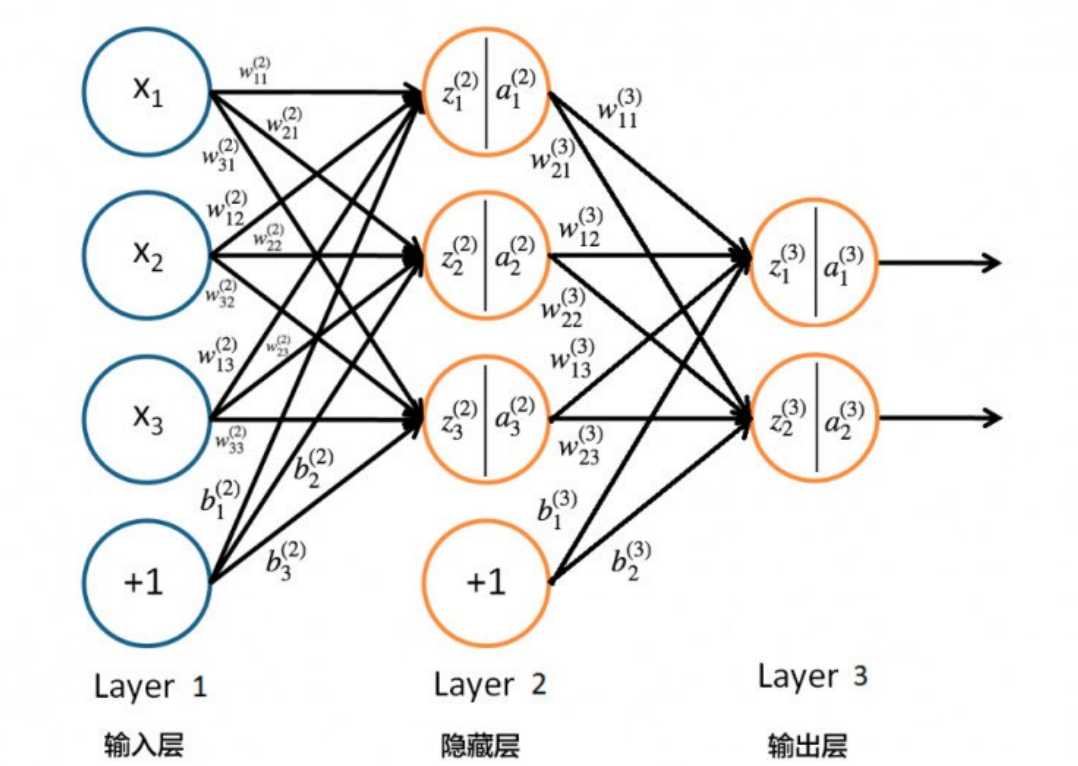


层级结构和互联网型结构

- 感知器和前馈神经网络是非常典型的层级连接结构。其中每层由并行的单元组成。通常同一层不具有连接、两个相邻层完全连接(每一层的每一个神经元到另一层的每个神经元)。



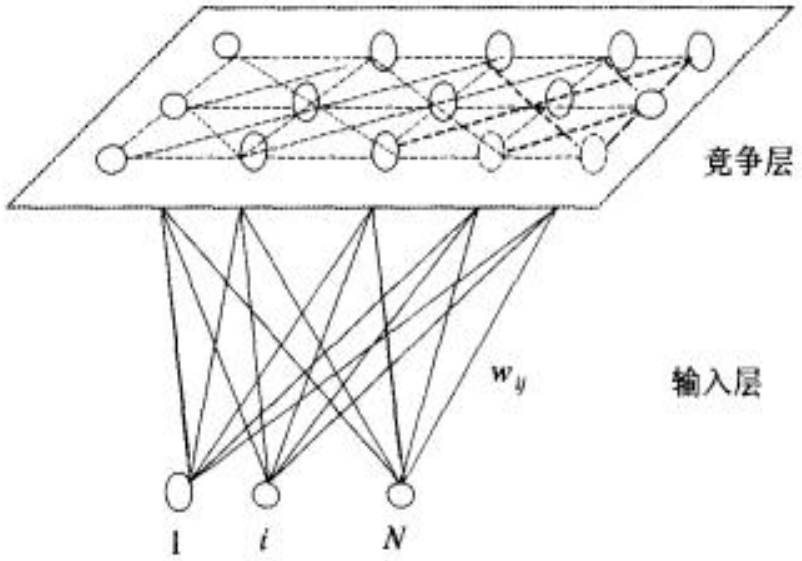
层级划分

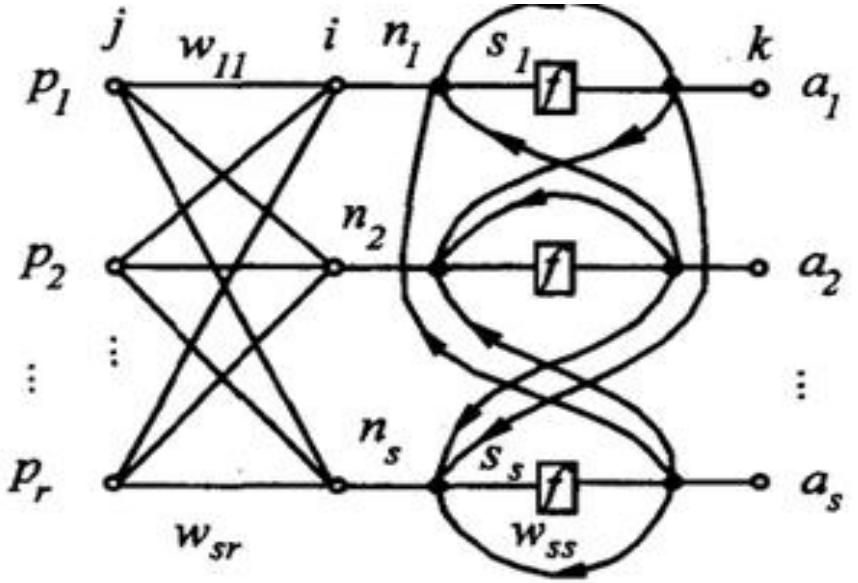


- 层级结构网络具有足够的隐藏神经元，理论上可以总是对输入和输出之间的关系建模。

- 实际应用中层级结构的应用是很有限的，通常将它们与其他网络结合形成新的网络。

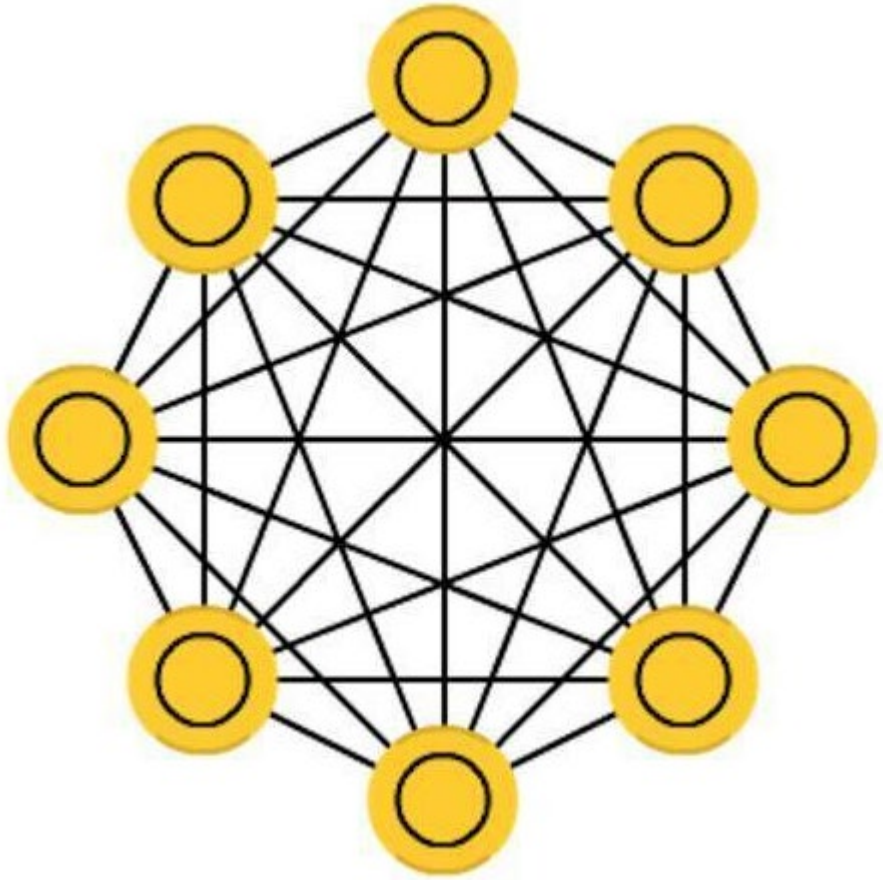
层（级）内联结模式





- 层（级）内联接:区域内（Intra-field）联接
  - 用来加强和完成层内神经元之间的竞争

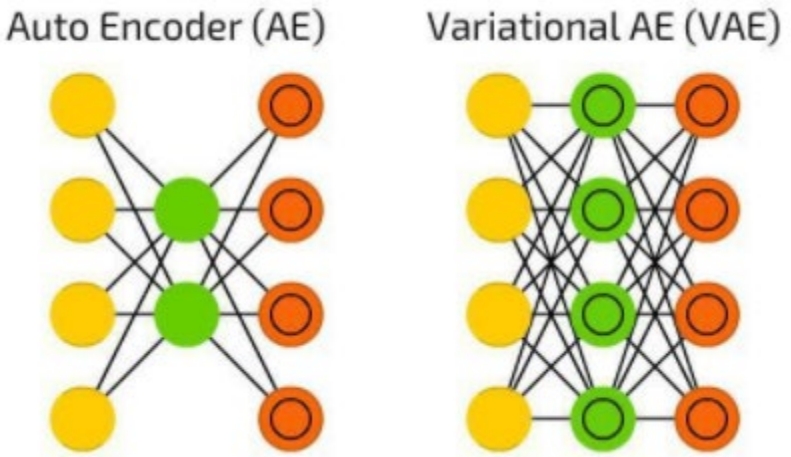
互联网型连接模式



- 最典型的就是马可夫链和Hopfield网络(HN)。
- Hopfield网络(HN)的每个神经元被连接到其他神经元;。每个节点在训练前输入，然后在训练期间隐藏并输出。通过将神经元的值设置为期望的模式来训练网络，此后权重不变。
- 马可夫链(MC或离散时间马尔可夫链，DTMC)是BM和HN的前身。它虽然不是真正的神经网络，但类似于神经网络，并且构成了BM和HNs的理论基础。

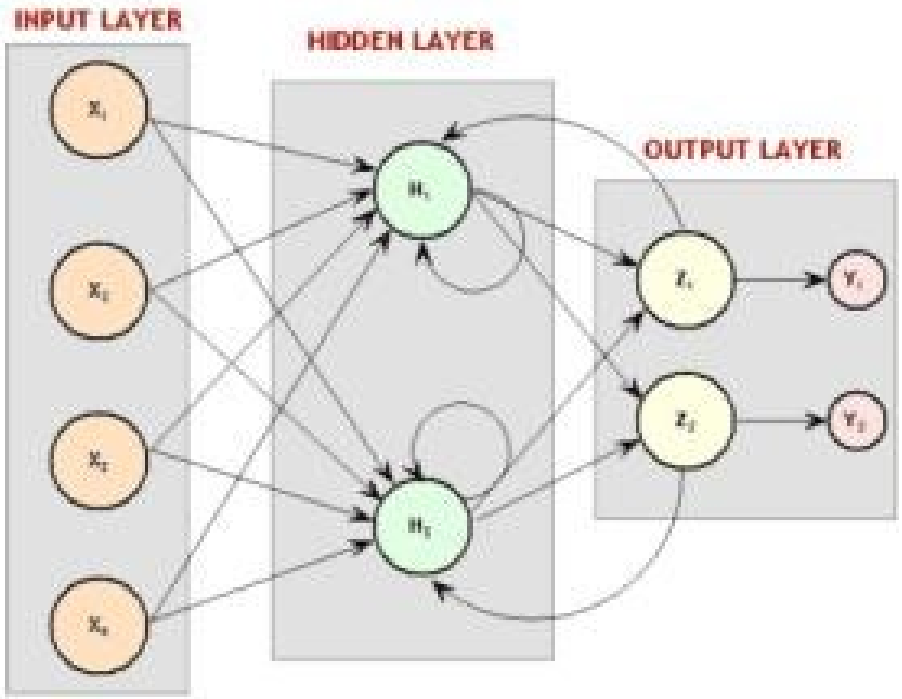
神经元计算方式

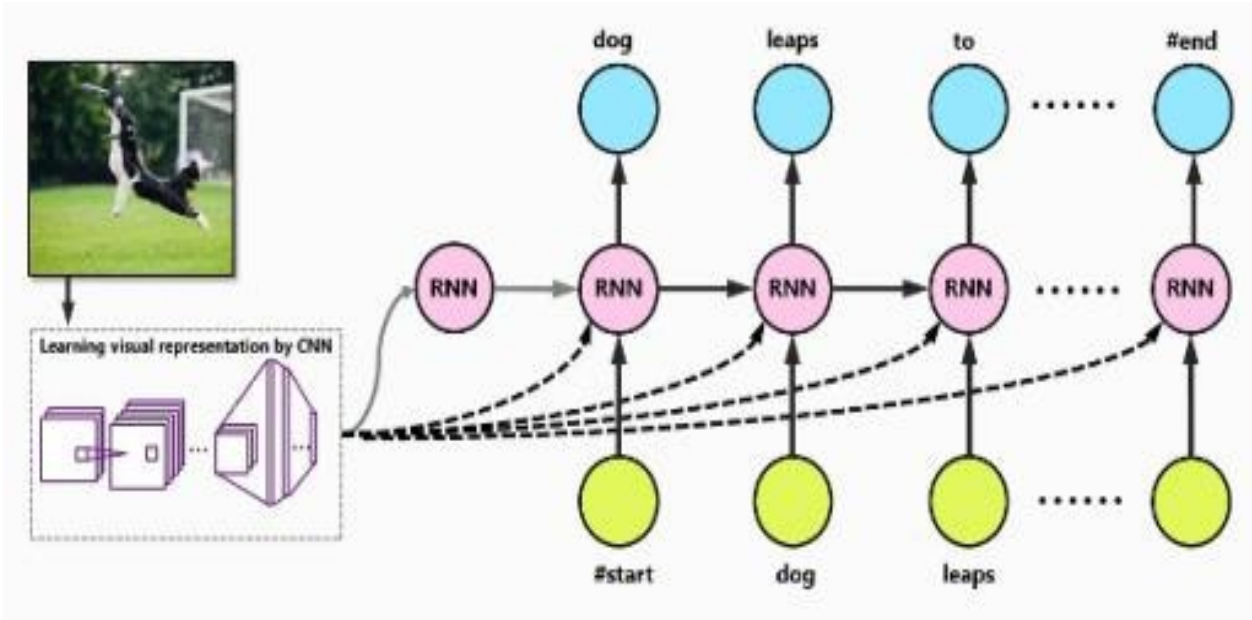
- 网络的连接并不能完全表示神经元的计算方式，还要根据每个单元的计算方式而定：



- 目前最主流的两种神经网络为RNN和CNN，很多复杂模型基于两者建立。

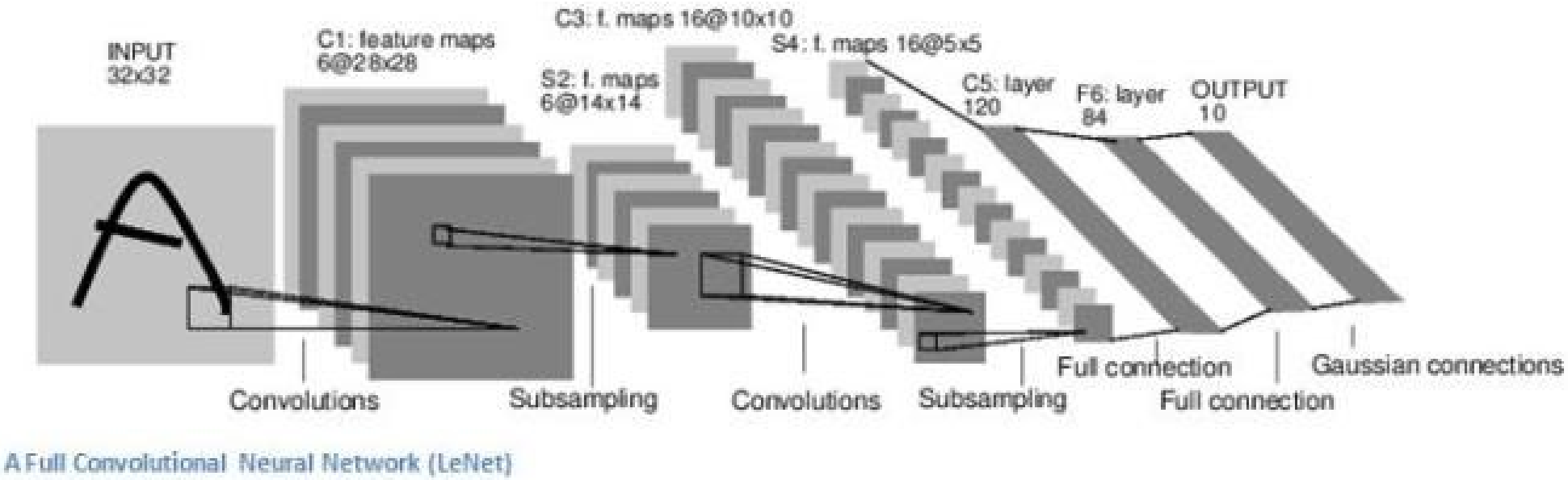
循环联结模式





- 循环联接：反馈信号

卷积计算模式



A Full Convolutional Neural Network (LeNet)

- 卷积结构：滑动窗口，共享权重

二、生物神经元的连接

神经元的功能

- 神经元的功能是分析和传递信息：神经系统的基本目标
  - 神经元如何为神经元发送信号提供能量?
  - 能量如何用于神经元内信号的产生?
  - 神经元间如何实现信号的相互传递?
- 神经元加工信息的目的
  - 接收信息、评估信息、将信号传递给其它神经元
  - 由此构成了局部或长程环路

神经冲动

- 神经信号
  - 化学形式的信号：神经递质、环境中能产生感觉的化学成分
  - 物理形式的信号：触摸、光线、电信号等
  - 信号引起突触后神经元细胞膜的变化，导致电流流入或流出神经元
- 轴突工作方式
  - 在每一个点上重新产生一个新的冲动
  - 距离大脑较远部位的轴突传递冲动的速度快于距离大脑较近部位的轴突

神经信号中的电流

- 电流在神经元内发挥信号作用，潜在影响远离传入突触位点的神经元膜
- 电流是由离子流传导的，离子流载有带电荷的离子
  - 不同于金属电线内的电流（由电子传导而非溶解的离子传导）
- 动作电位：长程传递的信号
  - 由锋电位启动区产生
    - 整合来自许多突触传入的或者来自被刺激的感受器的电流
  - 动作电位的结果是一个沿轴突下行传播到轴突末梢的信号
    - 最终引起神经递质的释放

神经元导电性

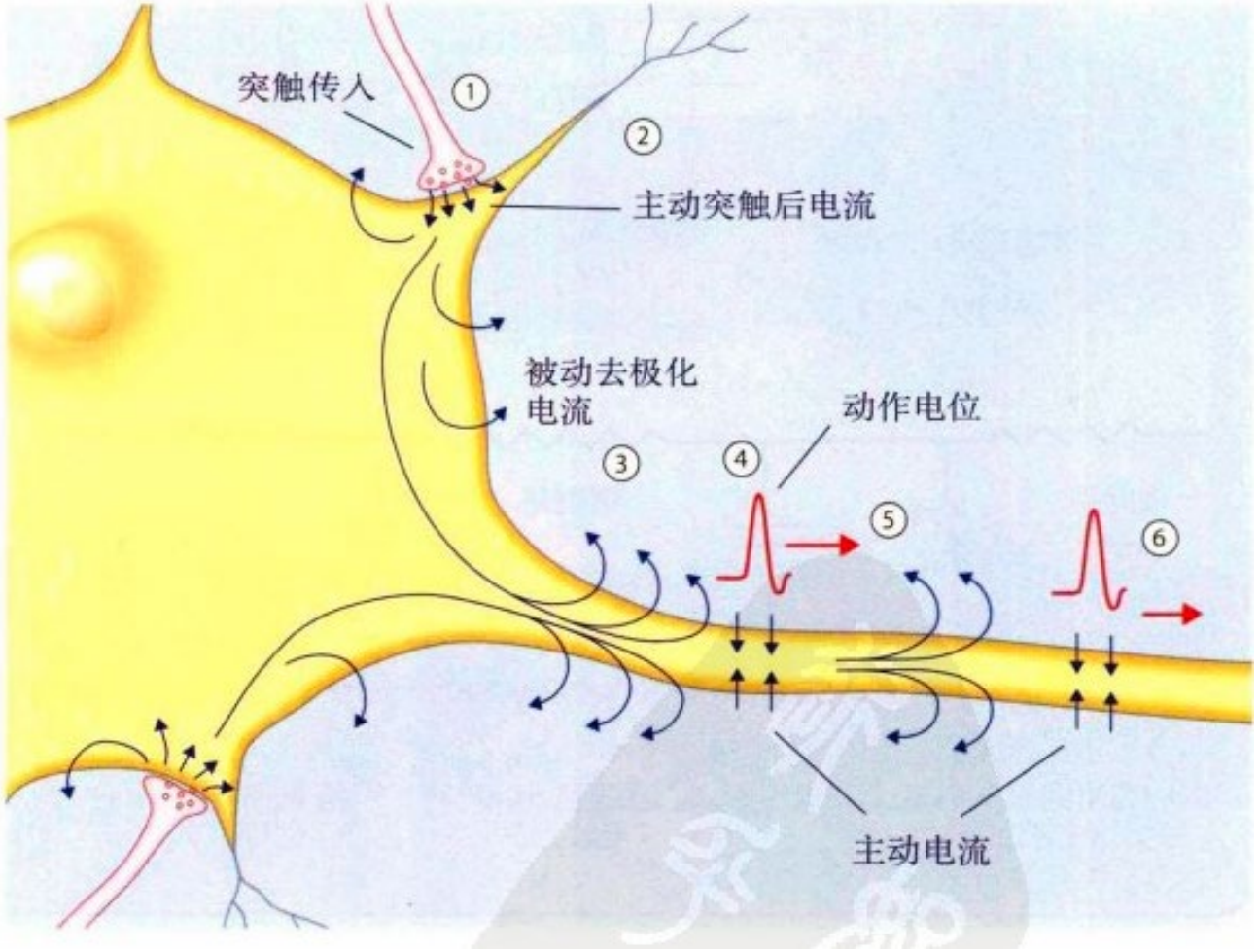


图 2.14 神经元间的信号传导概述。突触传入①影响突触后膜，导致突触后电流②。这些电流是主动的，但通过被动的电流传导被传导至细胞内③。如果电流足够强，将引起细胞膜去极化，从而启动轴突产生动作电位④。每个动作电位是一个主动的加工过程，只有在膜内门控的 Na^+ 通道开放的状态下才能发生。在此动作电位阶段所产生的内向电流遵循电传导原理沿轴突传导⑤。结果导致邻近区域细胞膜的去极化，然后产生另一个动作电位，这一过程沿轴突持续进行⑥。

- 主动和被动电流
  - 突触被激活，其附近产生主动的穿膜电流，从而产生突触电位
  - 该电流穿过突触后膜的局部区域，引起整个神经元的被动电流传导
- 被动电流
  - 极化：细胞膜内外存在电位差（静息膜电位）
  - 去极化的：兴奋性突触后电位，使细胞内的电环境更趋正性——易产生动作电位
  - 超极化的：抑制性突触后电位，使细胞内的电环境更趋负性——不易产生动作电位
- 动作电位是主动的膜活动
  - 其涉及的膜导电性改变是通过离子通道的开放和关闭实现的
  - 轴突内产生被动电流——引起邻近的轴突区域去极化，产生新的动作电位
- 沿轴突不断进行，直至轴突末梢
  - 末梢释放神经递质，在下一个神经元“上演”上述过程

电衰减传导：被动电特性

- 在没有主动过程（即动作电位）补充能量的情况下，被动离子电流沿树突、轴突、细胞体传导的距离有多远?
  - 起始电流的幅度、神经元细胞膜的电阻（和电容）、细胞内液和细胞外液的导电性
- 电流幅度随距离增加而衰减
  - 被动电流传导只在短距离发挥作用，不适合用于长距离信号交流（1mm）
  - 电流越强——传导距离越远
  - 膜电阻提高时——电流以轴突内流动为主，很少流出
  - 细胞内的电导越强——距离越远

跳跃式传导和髓鞘的作用

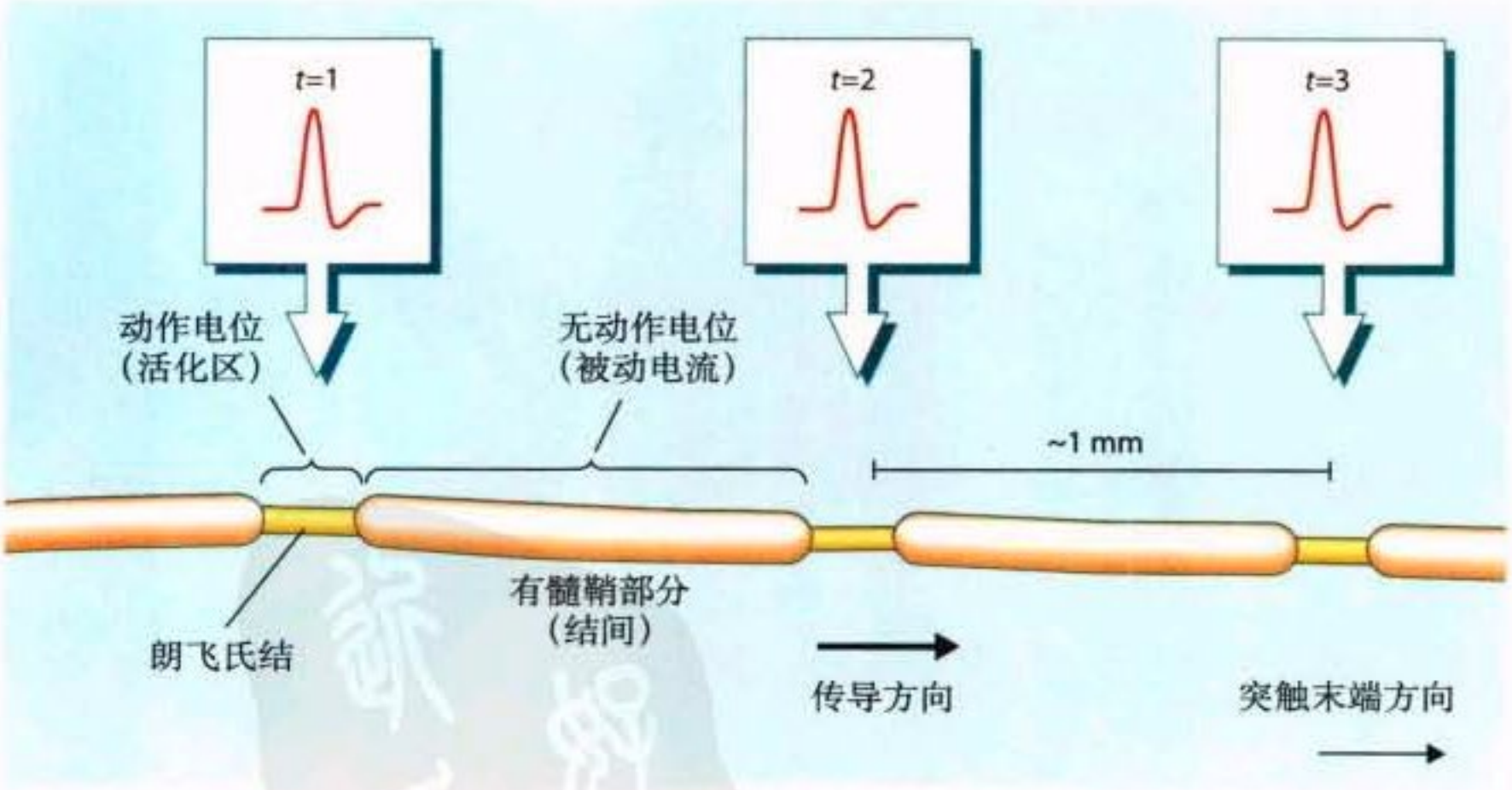
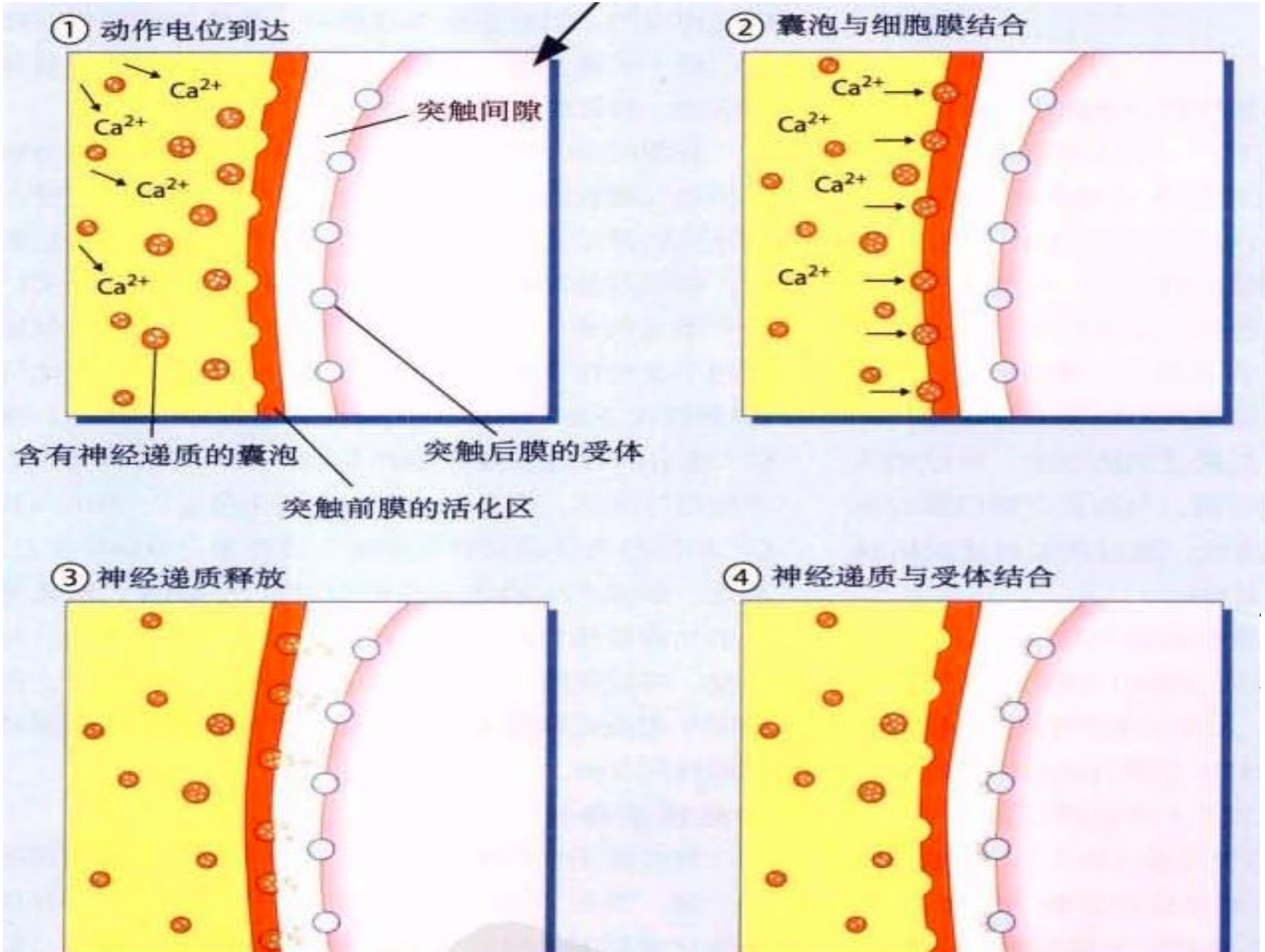


图 2.24 髓鞘纤维的跳跃式传导。有髓纤维的动作电位只在髓鞘中断的朗飞氏结处产生。朗飞氏结处产生的电流遵循电传导的基本原则在朗飞氏结间流动。结间的距离是由轴突的长度常数所决定的，受轴突髓鞘化程度和轴突直径的影响。因此朗飞氏结是呈最佳的分布，来自最后一个朗飞氏结的动作电位电流也能使此处的细胞膜去极化程度超过阈值，从而通过朗飞氏结的电流再生实现整个轴突的信号传递。

突触传递


① 动作电位到达
Ca2+
Ca2+
Ca2+
Ca2+
突触间隙
含有神经递质的囊泡
突触后膜的受体
突触前膜的活化区
② 囊泡与细胞膜结合
Ca2+
Ca2+
Ca2+
Ca2+
③ 神经递质释放
④ 神经递质与受体结合

流

体分

神经递质

- 神经元信息传递的一个重要元素是神经递质，有100多种神经递质
  - 都是突触前神经元合成
  - 运送至轴突末梢
  - 储存在囊泡内并可胞吐至突触间歇
  - 在与突触后细胞键合后，被酶降解或清除
- 每个神经元能合成一种或多种神经递质
  - 突触前神经元可调节神经递质的合成和释放
  - 根据刺激条件而单独释放或全部释放
- 神经递质对突触后神经元的作用取决于突触后神经元而不是神经递质本身
  - 相同神经递质对不同突触后神经元触发不同作用

电突触

- 电突触没有分隔两个神经元的突触间隙
  - 细胞膜相互接触、细胞浆相连续
  - 适用于需快速传导信息的情况（无脊椎动物的逃避反射）
  - 同样适用于需要多个神经元同步运作的情况
- 缺点
  - 不能传递抑制性信息
  - 缺乏可塑性，不能放大信号
- 电突触不是少见的突触类型
  - 哺乳动物中枢神经系统的神经信息传递包括化学传递和电传递

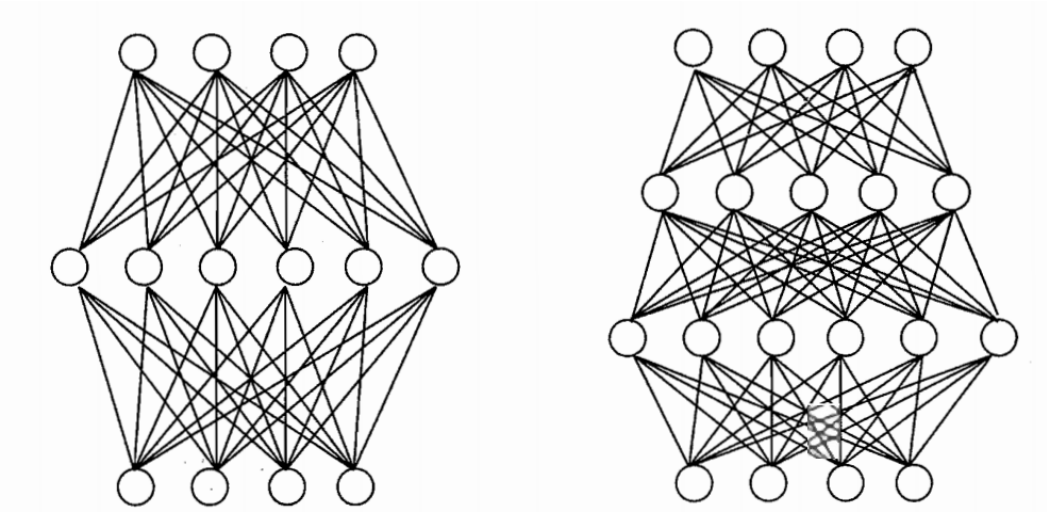
神经网络的实现方法

- 如何选取网络结构
  - 网络的输入个数=应用问题的输入数
  - 输出层神经元的数目=应用问题的输出数目
  - 输出层的激活函数选择至少部分依赖于应用问题的输出描述
- 实现方法
  - 硬件：通过电子元件，设计神经网络电路来模拟大脑（神经网络计算机）
  - 软件：通过计算机来模拟大脑
- 软件实现途径
  - 建立一个神经网络模型（结构、激活函数）
  - 对该神经网络模型进行训练
  - 利用训练好的神经网络完成一些任务

神经元的扩展

- 宽度:

- 深度:



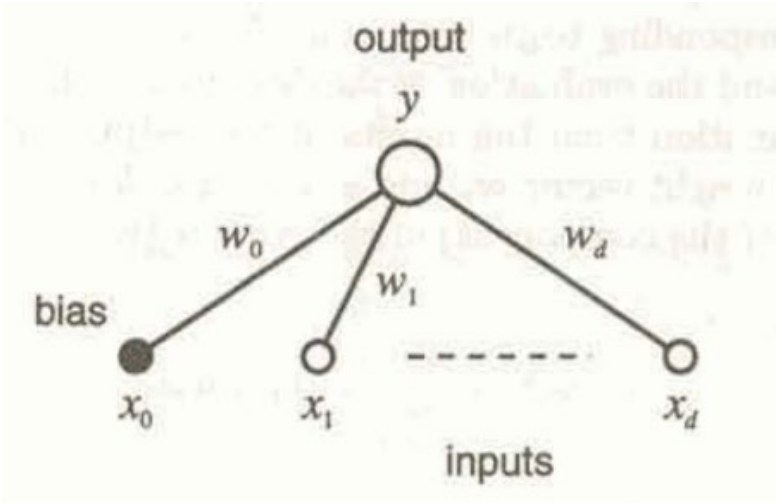
三、宽度扩展：
多输出的单层感知器为例

多输出的单层感知器

- 回顾第二讲中苹果和柠檬的二分类实例

水果	宽度	高度	水果	宽度	高度	水果	宽度	高度	水果	宽度	高度
苹果	8.4	7.3	苹果	7.6	7.3	苹果	7.6	7.9	柠檬	6.0	7.5
苹果	8.0	6.8	苹果	7.7	7.1	柠檬	7.2	10.3	柠檬	5.9	8.0
苹果	7.4	7.2	苹果	7.6	7.5	柠檬	7.3	10.5	柠檬	6.0	8.4
苹果	7.1	7.8	苹果	7.5	7.6	柠檬	7.2	9.2	柠檬	6.1	8.5
苹果	7.4	7.0	苹果	7.5	7.1	柠檬	7.3	10.2	柠檬	6.3	7.7
苹果	6.9	7.3	苹果	7.4	7.2	柠檬	7.3	9.7	柠檬	5.9	8.1
苹果	7.1	7.6	苹果	7.5	7.5	柠檬	7.3	10.1	柠檬	6.5	8.5
苹果	7.0	7.1	苹果	7.4	7.4	柠檬	5.8	8.7	柠檬	6.1	8.1
苹果	7.3	7.7	苹果	7.3	7.1	柠檬	6.0	8.2			

- 二分类问题：



思考：如果是多分类问题如何设计神经网络?

多输出的单层感知器

- 在输出层上对每一个输出类有一个输出神经元
- 训练过程：

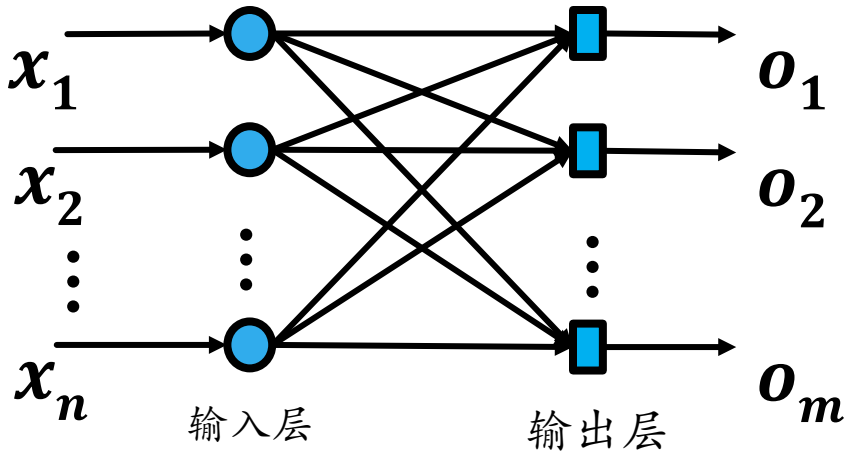
反复调整权值在相关输出神经元产生输出1，在其他剩余的神经元上产生输出0，将每一个输入模式映射到正确的输出类。

- 问题

会不会存在许多输入模式映射到几个类？这意味着什么？

离散多输出感知器训练算法

- 样本集：$\{(\boldsymbol{x},\boldsymbol{y})|\boldsymbol{y}$为输入向量$\boldsymbol{x}$对应的输出$\}$
- 输入向量：$\boldsymbol{x}=(x_1,x_2,\ldots,x_n)$
- 理想输出向量：$\boldsymbol{y}=(y_1,y_2,\ldots,y_m)$
- 激活函数：F
- 权矩阵：$\boldsymbol{W}=(w_{ij})$
- 实际输出向量：$\boldsymbol{o}=(o_1,o_2,\ldots,om)$



多输出感知器

离散多输出感知器训练算法

1.初始化权矩阵$\boldsymbol{W}$;

2.重复下列过程，直到训练完成:

2.1 对每个样本$(\boldsymbol{x}, \boldsymbol{y})$，重复如下过程:

2.1.1 输入$\boldsymbol{x}$;

2.1.2 计算$\boldsymbol{o} = F(\boldsymbol{x}\,\boldsymbol{W})$;

2.1.3 for j=1 to m do 执行如下操作:

if $o_j \neq y_j$ then

if $o_i = 0$ then for $i = 1$ to n

$w_{ij} = w_{ij} + x_i$

else for $i = 1$ to n do

$w_{ij} = w_{ij} - xi$

离散多输出感知器训练算法

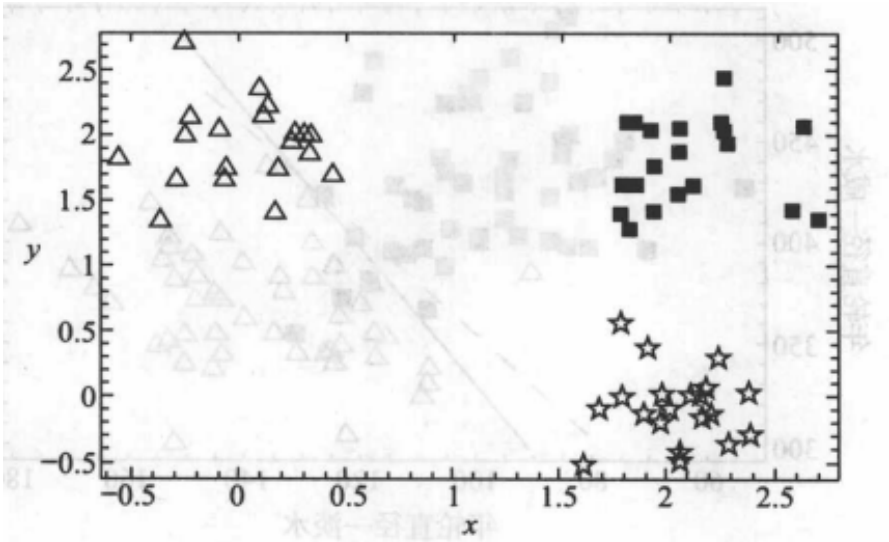
- 算法思想：将单输出感知器的处理逐个地用于多输出感知器输出层的每一个神经元的处理。
- 权矩阵的初始化：一系列小伪随机数。
- 循环控制
  - 方法1：循环次数控制法：对样本集执行规定次数的迭代
    - 改进——分阶段迭代控制：设定一个基本的迭代次数N，每当训练完成N次迭代后，就给出一个中间结果
  - 方法2：精度控制法：给定一个精度控制参数
  - 方法3：综合控制法：将这两种方法结合起来使用

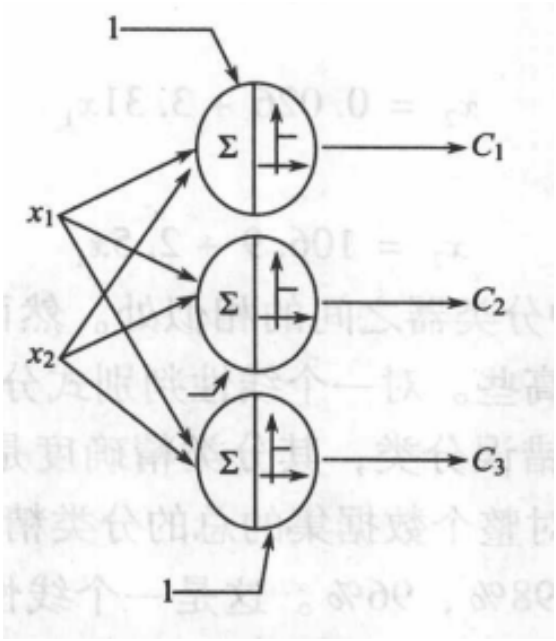
离散多输出感知器训练算法

- 精度度量：实际输出向量与理想输出向量的对应分量的差的绝对值之和；
- 实际输出向量与理想输出向量的欧氏距离的和
- 注意：精度参数的设置。根据实际问题选定；初始测试阶段，精度要求低，测试完成后，再给出实际的精度要求。
- “死循环”：网络无法表示样本所代表的问题

示例：两维多类感知器网络

- 数据： 三类数据
- 网络结构：三个输出感知器神经元，每个对应一个类





示例：两维多类感知器网络

- 训练：感知器学习算法
- 不同类别的目标输出

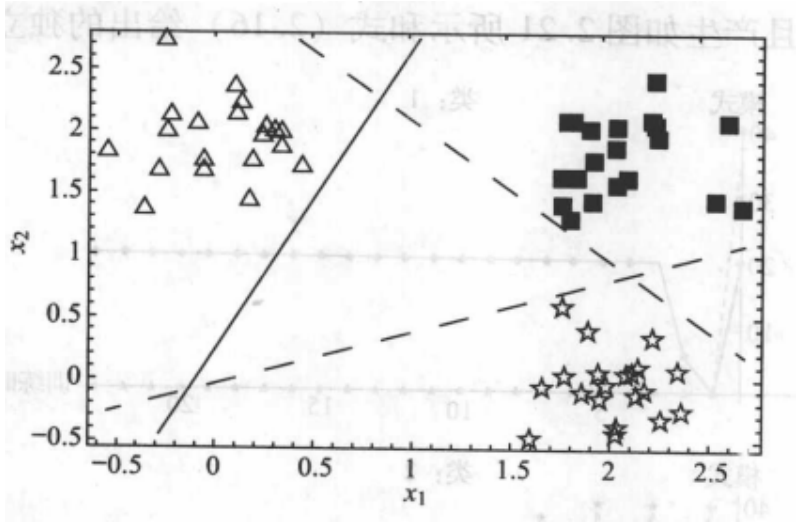
类别	C1	C2	C3
类1	1	0	0
类2	0	1	0
类3	0	0	1

- 通过将三个感知器的实际输出与目标输出进行比较，使用学习规则调整不正确分类的感知器权值，直到所有感知器都能对数据进行正确分类。
- 每个感知器将属于它们的数据进行描述并且不会误分其他数据。

示例：两维多类感知器网络

- 训练后的分类界限

每个感知器定义一个线性分类界限，所以有三条线性分界线，每一个分类界限将一类与其他两类分割开。



$$-3.8-42.8x_1+17.5x_2=0$$
$$-78.2+26.9x_1+24.3x_2=0$$
$$-1.35+16.6x_1-38.0x_2=0$$

每条分界线的公式，系数代表神经元的权值，截距代表与偏差输入+1相关的权值

- 问题：为什么会存在重叠区域?

连续多输出感知器训练算法

- 用公式$w_{ij} = w_{ij} + \alpha（y_j - oj）xi$取代了前面算法第2.1.3步中的多个判断
- y_j与o_j之间的差别对w_{ij}的影响由$\alpha（y_j - oj）xi$表现出来
- 好处：不仅使得算法的控制在结构上更容易理解，而且还使得它的适应面更宽

连续多输出感知器训练算法

1. 用适当的小伪随机数初始化权矩阵$\boldsymbol{W}$;
2. 初置精度控制参数ε，学习率α，精度控制变量$d = \varepsilon + 1$;
3. While $d \geq \varepsilon$ do

3.1 $d = 0$;

3.2 for 每个样本$(\boldsymbol{x}, \boldsymbol{y})$do

3.2.1 输入$\boldsymbol{x}(= (x_1, x_2, \ldots, x_n))$;

3.2.2 求$\boldsymbol{o} = F(\boldsymbol{x}\,\boldsymbol{W})$;

3.2.3 修改权矩阵$\boldsymbol{W}$:

for $i = 1$ to n, $j = 1$ to m do

$w_{ij} = w_{ij} + \alpha(y_j - oj)xi$;

3.2.4 累积误差

for $j = 1$ to m do

$d = d + (y_j - oj)^2$

连续多输出感知器训练算法

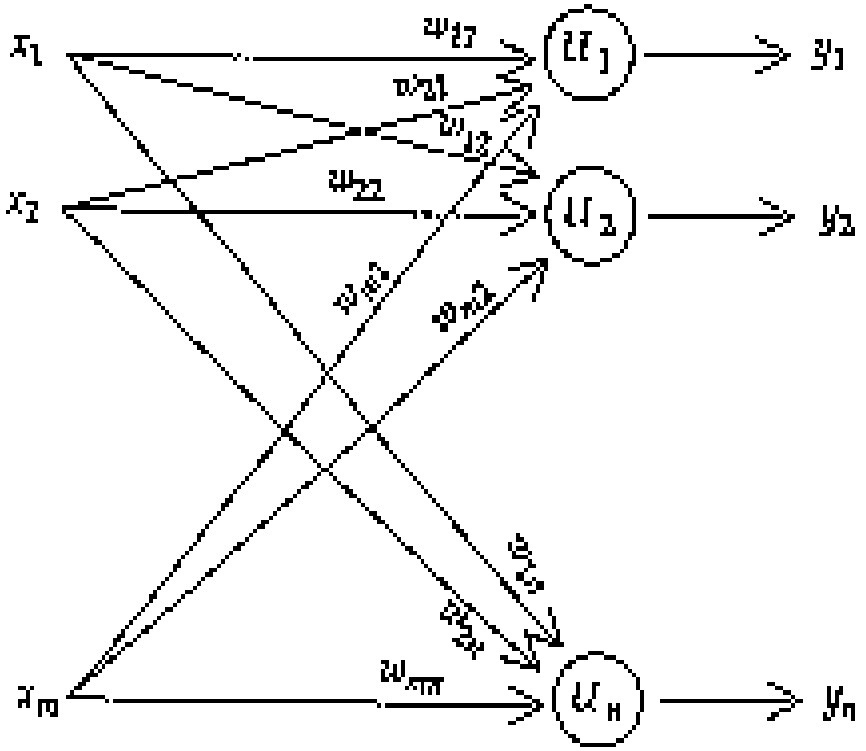
- 程序实现:ε、α、d、i、j、n、m为简单变量来表示，

$\boldsymbol{W}$为n行m列的二维数组，样本集为二维数组

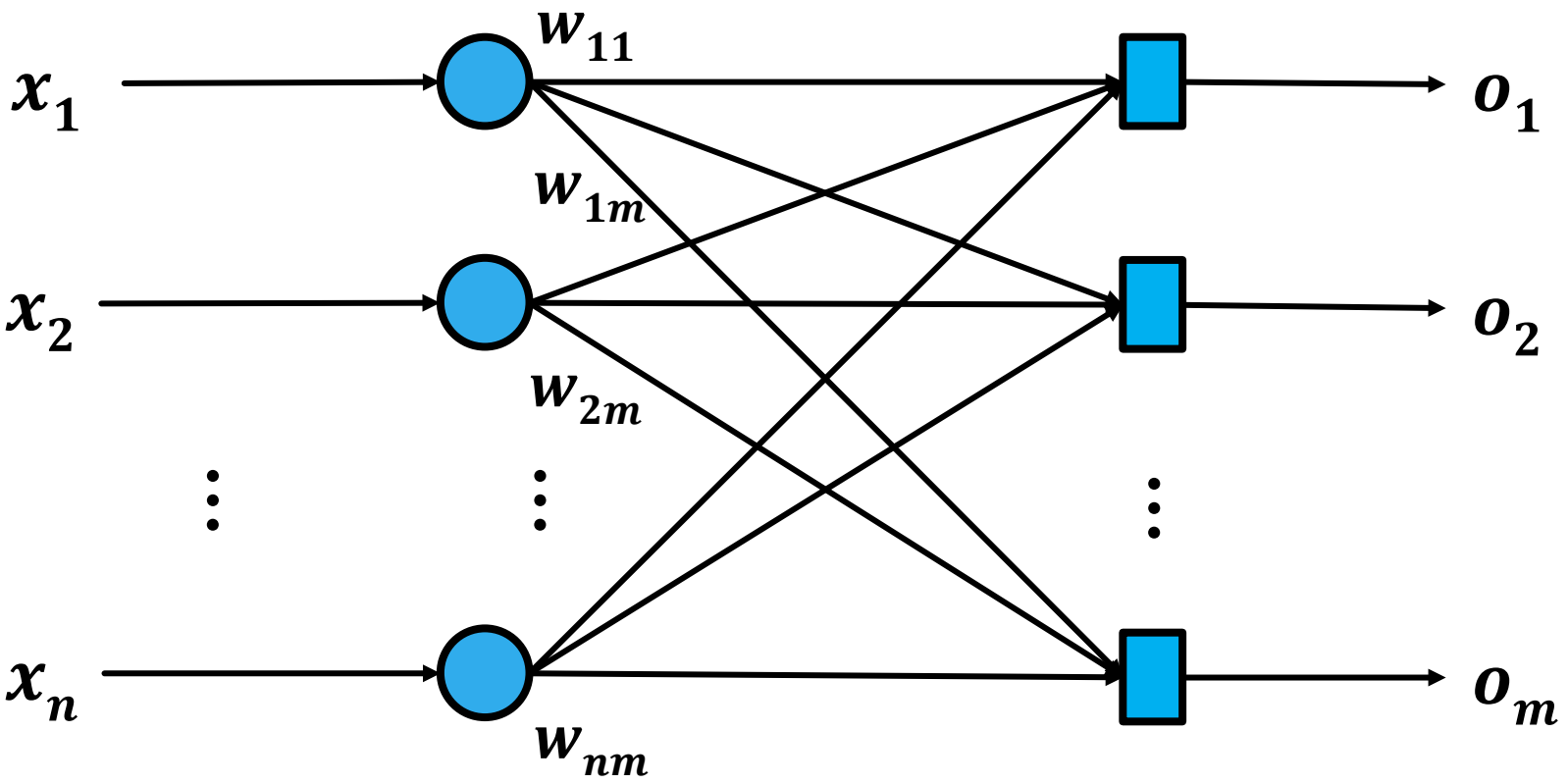
- Minsky在1969年证明，有许多基本问题是感知器无法解决：线性不可分问题
- 很难从样本数据集直接看出问题是否线性可分
- 未能证明，一个感知器究竟需要经过多少步才能完成训练

单级前馈网
（Single-Layer Feedforward Net）

- 只有一层连接权重
- 输入层：从外部接受数据
- 输出层：输出神经网络的反应
- 输入层单元和输出层单元全连接
- 输入层单元自身不相连
- 输出层自身不相连
- 代表：单层感知器



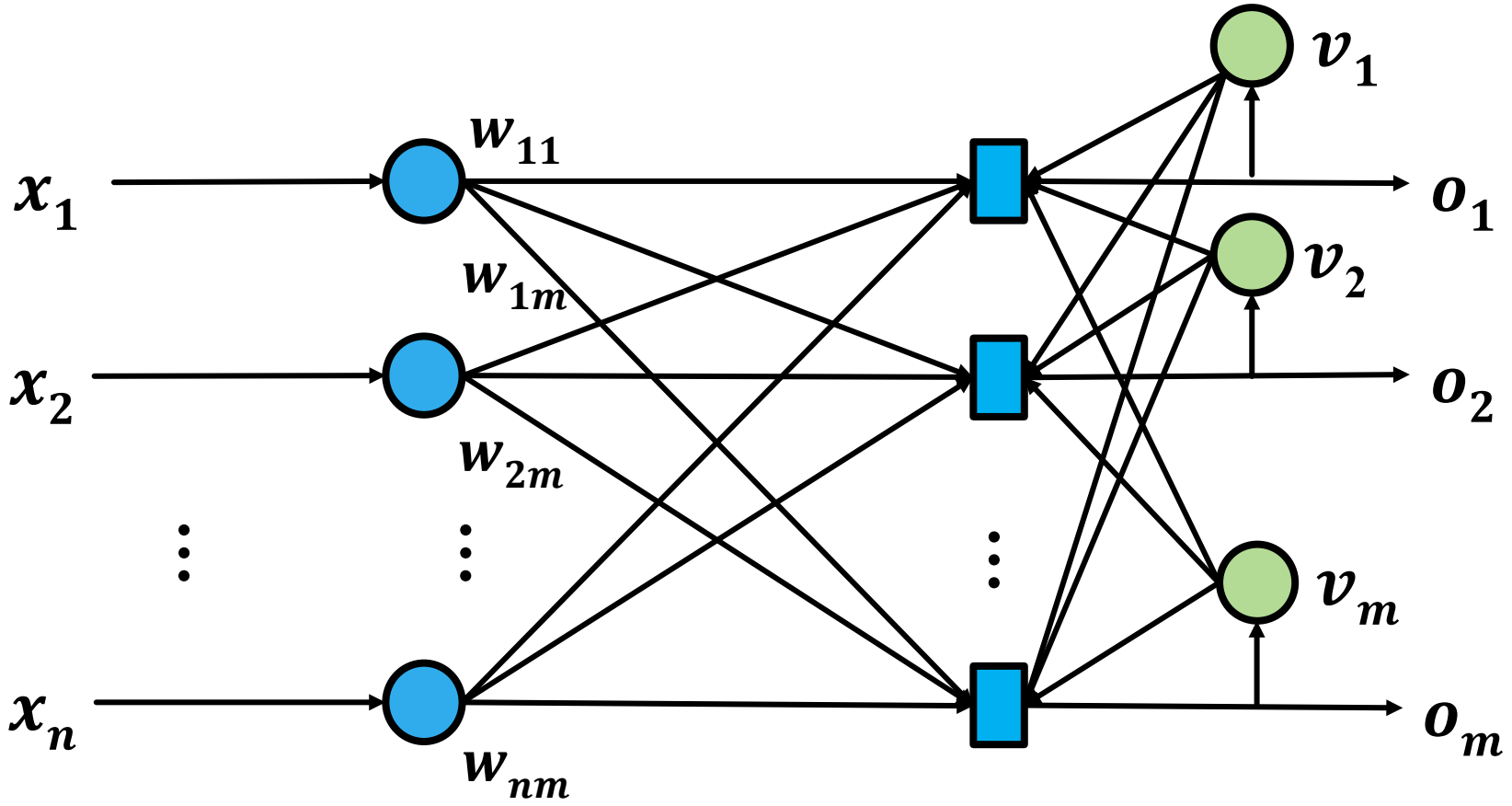
单级前馈网



单级前馈网

- 信号传播
  - $\boldsymbol{W} = (w_{ij})$
  - 输出层的第j个神经元的网络输入记为net_j
    - $net_j = x_1 w_{1j} + x_2 w_{2j} + \cdots + x_n w_{nj}, \quad 1 \le j \le m$
  - 取$\boldsymbol{NET} = (net_1, net_2, \ldots, net_m)$
    - $\boldsymbol{NET} = \boldsymbol{XW}$
    - $\boldsymbol{O} = F(\boldsymbol{NET})$

单级横向反馈网



输入层　　输出层

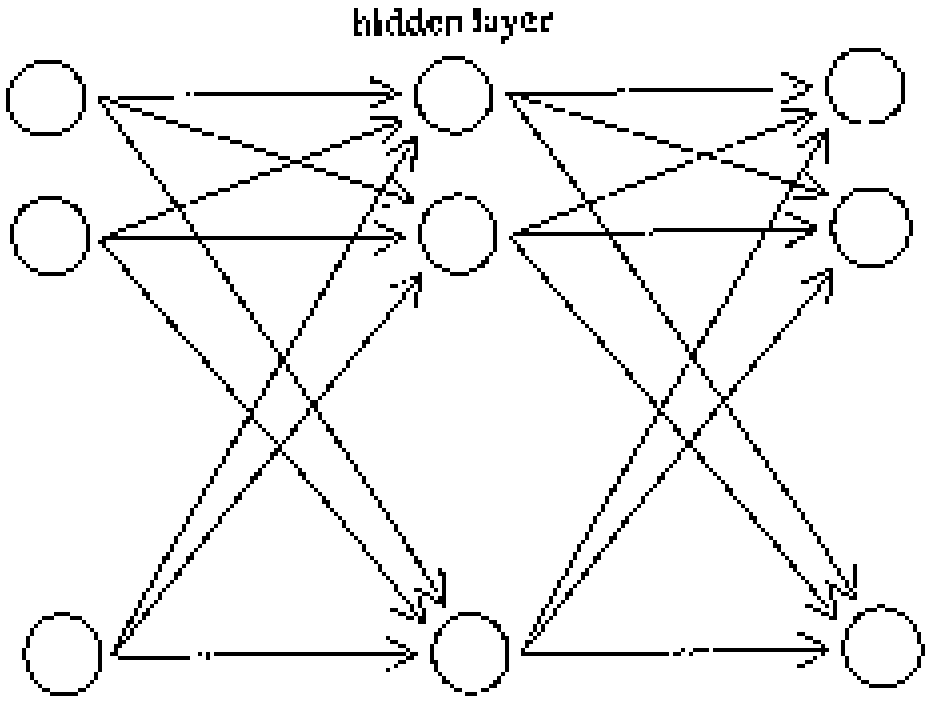
单级横向反馈网

- 信号传播
  - $\boldsymbol{V} = (\boldsymbol{v_{ij}})$
  - $\boldsymbol{NET} = \boldsymbol{XW} + \boldsymbol{OV}$
  - $\boldsymbol{O} = F(\boldsymbol{NET})$
  - 时间参数——神经元的状态在主时钟的控制下同步变化
  - 考虑$\boldsymbol{X}$总加在网上的情况
    - $\boldsymbol{NET}(t+1) = \boldsymbol{X}(t)\boldsymbol{W} + \boldsymbol{O}(t)\boldsymbol{V}$
    - $\boldsymbol{O}(t+1) = F\big(\boldsymbol{NET}(t+1)\big)$
    - $\boldsymbol{O}(0) = \boldsymbol{0}$

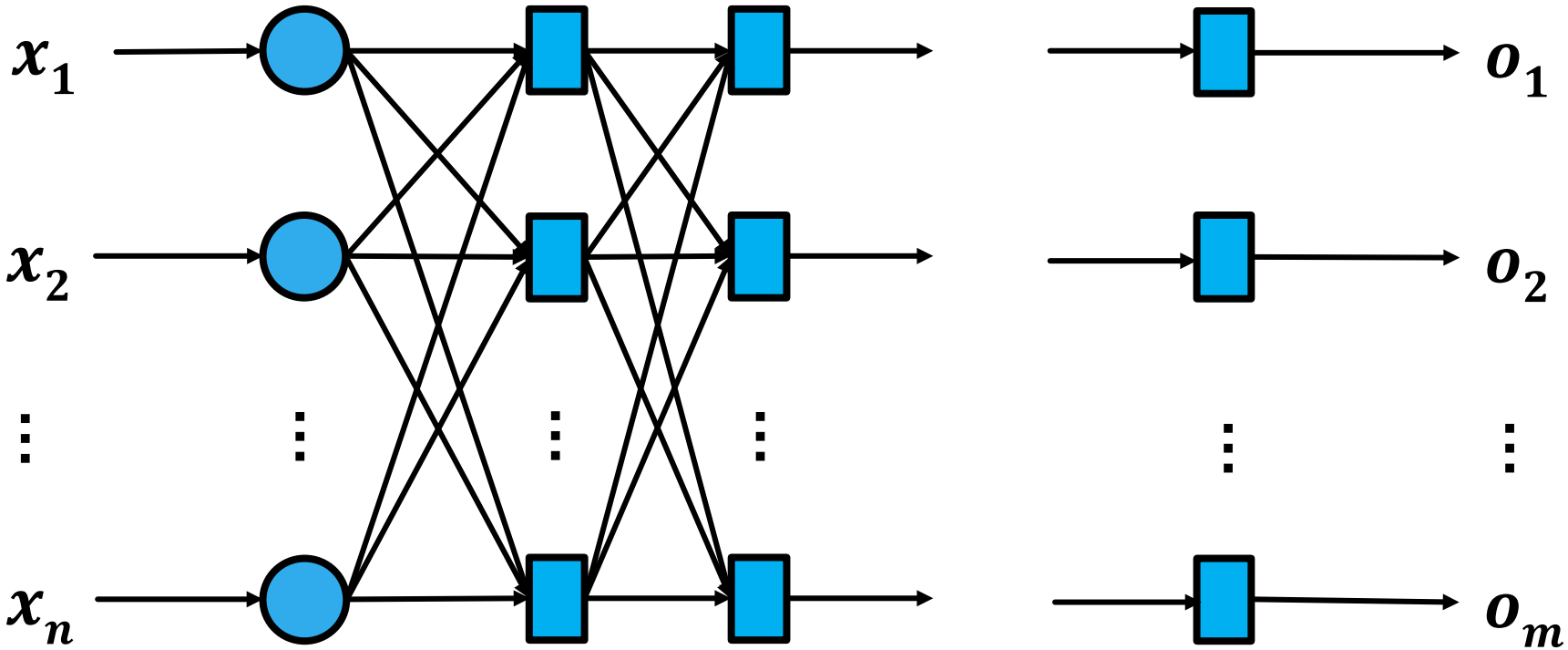
四、深度扩展：多层感知器

多级前馈网
（Multilayer Feedforward Net）

- 在输入层和输出层之间至少有一层（称之为隐藏层，hidden layer）
- 和单层神经网络相比
  - 能解决更加复杂的问题
  - 训练更加困难
  - 有些情况下能解决单层不能解决的问题
- 问题
  - 隐藏层数目
  - 隐藏层单元数目
- 代表：多层感知器、CNN



多级网



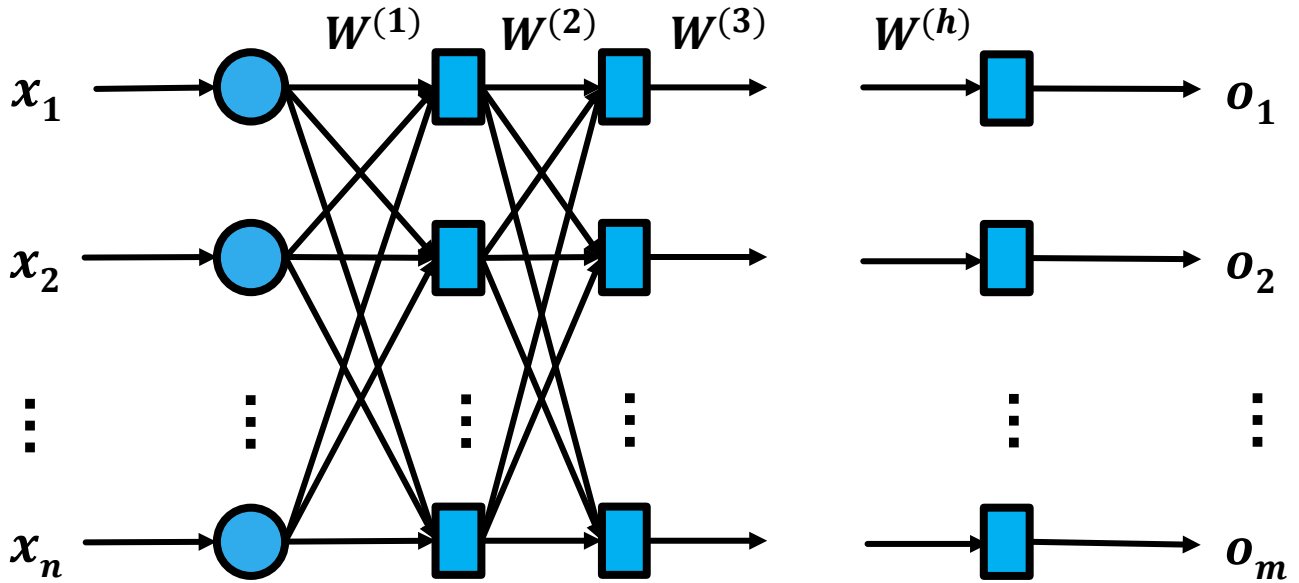
输入层　　隐藏层　　输出层

多级网层次划分

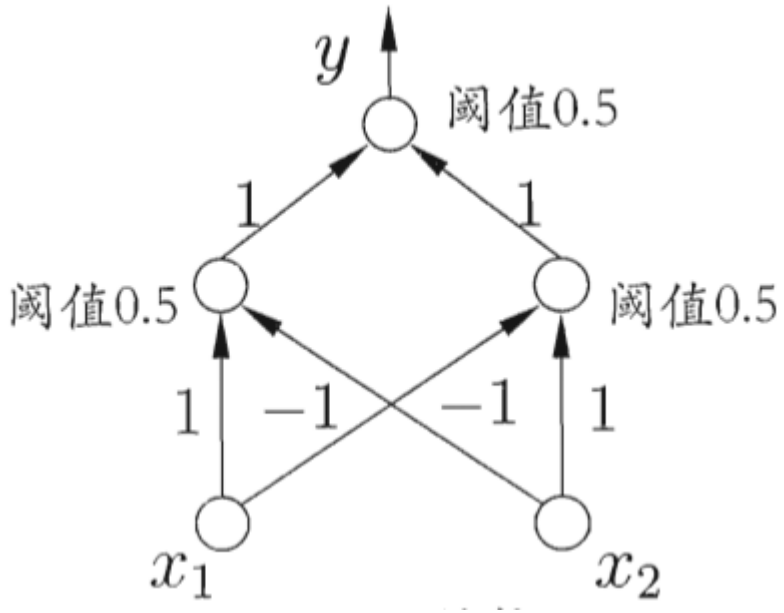
- 信号只被允许从较低层流向较高层。
- 层号确定层的高低：层号较小者，层次较低，层号较大者，层次较高。
- 输入层：被记作第0层。该层负责接收来自网络外部的信息
- 第j层：第$j-1$层的直接后继层$(j>0)$，它直接接受第$j-1$层的输出。
- 输出层：它是网络的最后一层，具有该网络的最大层号，负责输出网络的计算结果。
- 隐藏层：除输入层和输出层以外的其它各层叫隐藏层。隐藏层不直接接受外界的信号，也不直接向外界发送信号

术语说明

- 输出层的层号为该网络的层数：n层网络，或n级网络。
- 第$j-1$层到第j层的联接矩阵为第j层联接矩阵，输出层对应的矩阵叫输出层联接矩阵。今后，在需要的时候，一般我们用$W^{(j)}$表示第j层矩阵。



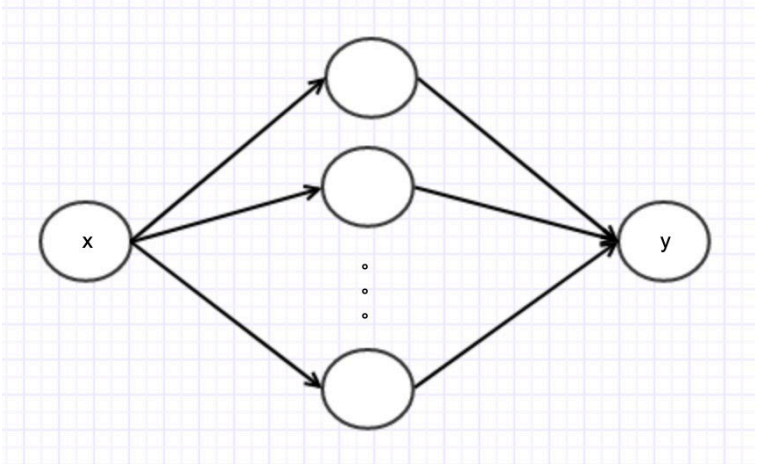
- 回顾作业题：设计实现XOR的感知器



- 单层感知器无法应对非线性的问题

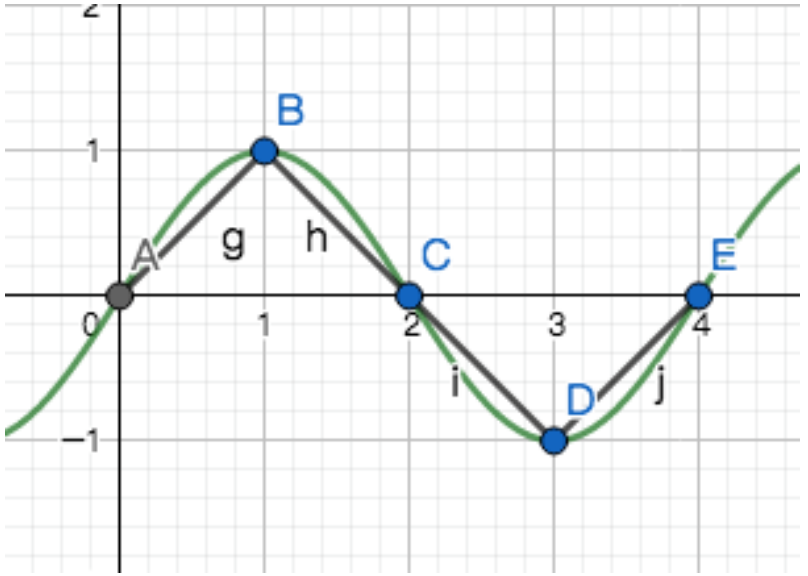
含有一个隐含层的多层感知器

- 隐含层：介于输入层和输出层之间，和输入层、输出层相比，与外界环境隔离。
- 具有单隐层的神经网络可以拟合任意函数。
- 隐含层的数量可以任意增加。



Sin函数的拟合问题

- 我们先定义一个在二维平面内的Sin函数上采样A-E五个坐标点的问题。



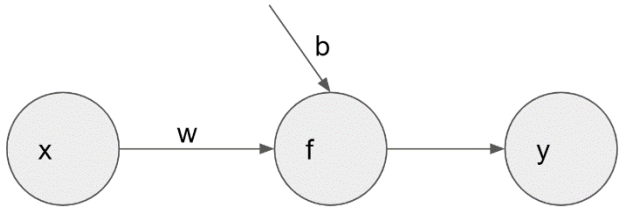
$Sin(\pi * x/2)$的函数图像

手工搭建并设置单隐藏层网络模型

- 首先, 拟合在区间$[0, 2\pi]$上的$\mathrm{Sin}(\pi * x/2)$函数图像上采样的五个A-E点。

- 采取的激活神经元的为ReLU激活神经元。

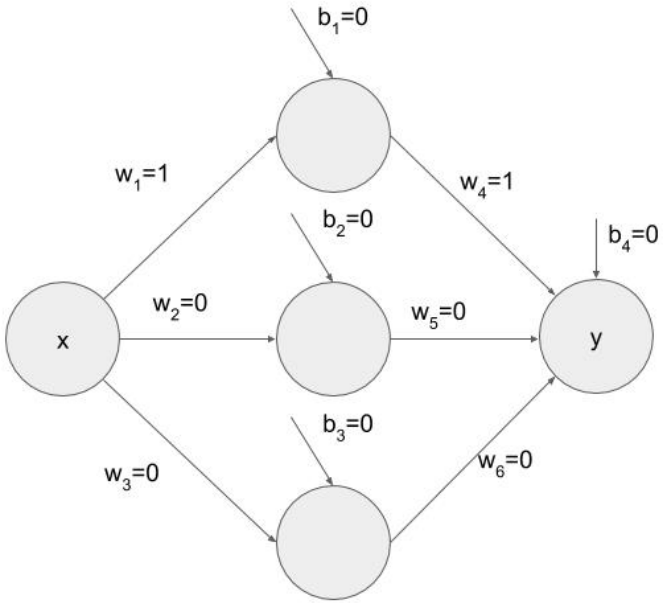
ReLU激活神经元



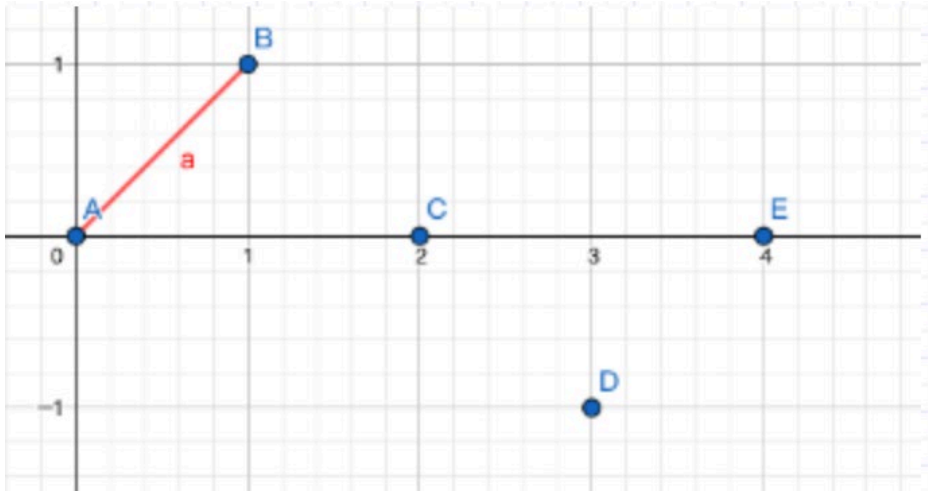
y = f(wx + b)

中间神经元用ReLU激活函数进行激活

手工搭建并设置单隐藏层网络模型

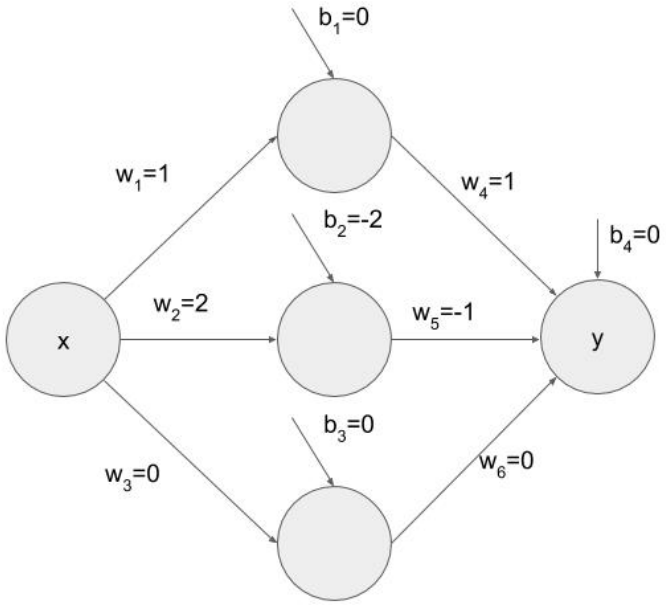


隐藏层使用relu激活函数，输出层使用线性激活函数

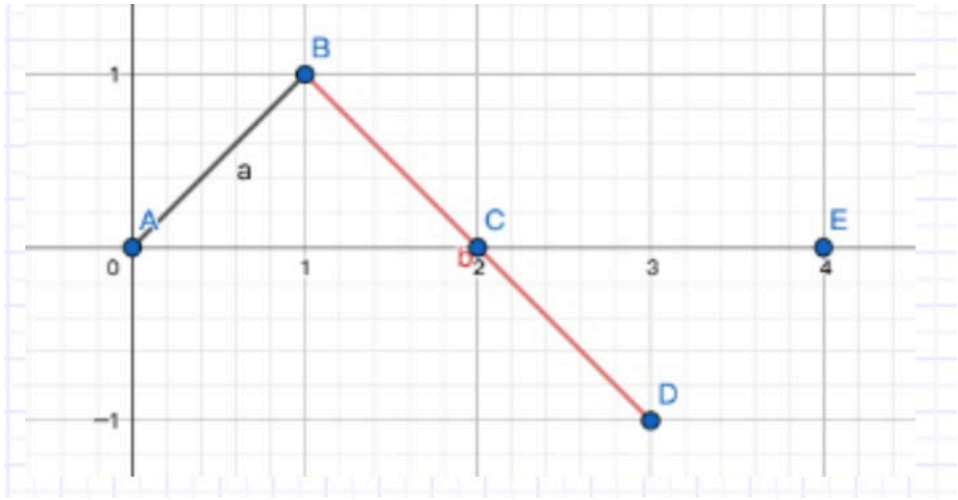


手工设置单个隐藏层参数，拟合A, B两点的这线段

手工搭建并设置单隐藏层网络模型

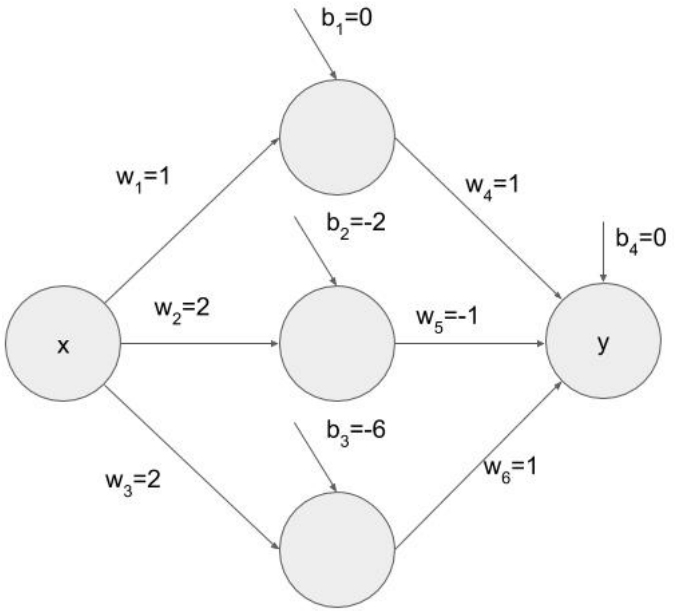


隐藏层使用relu激活函数，输出层使用线性激活函数

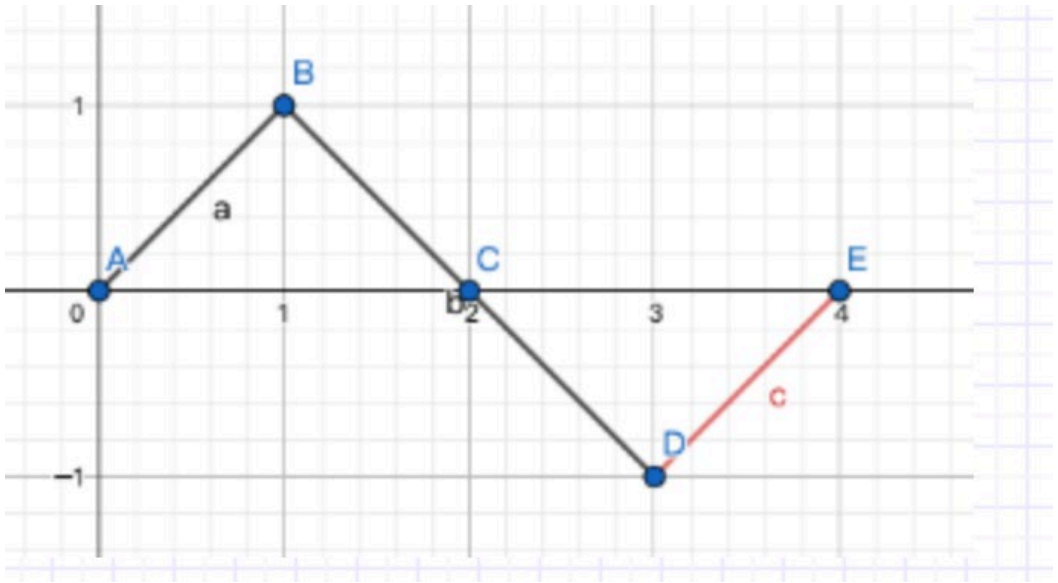


拟合A, B,C,D四点的网络模型和对应的函数图像

手工搭建并设置单隐藏层网络模型



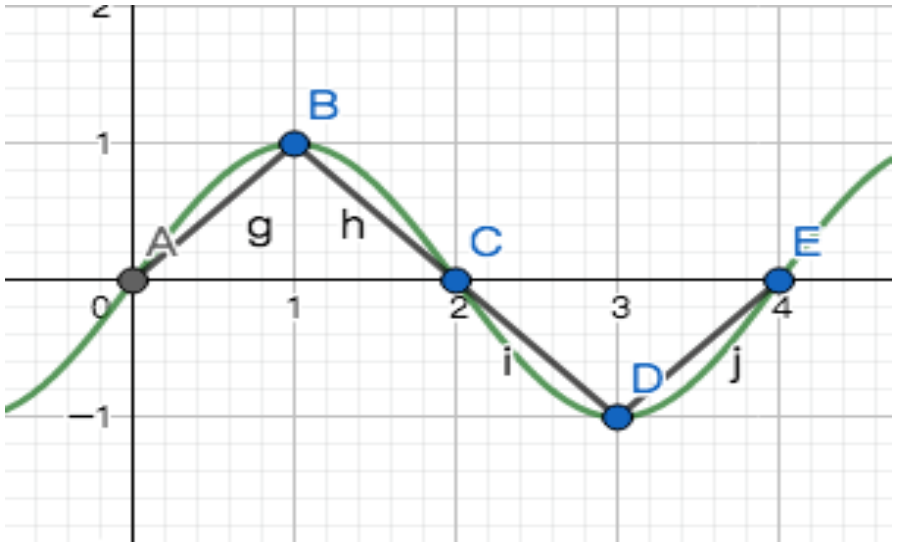
隐藏层使用relu激活函数，输出层使用线性激活函数

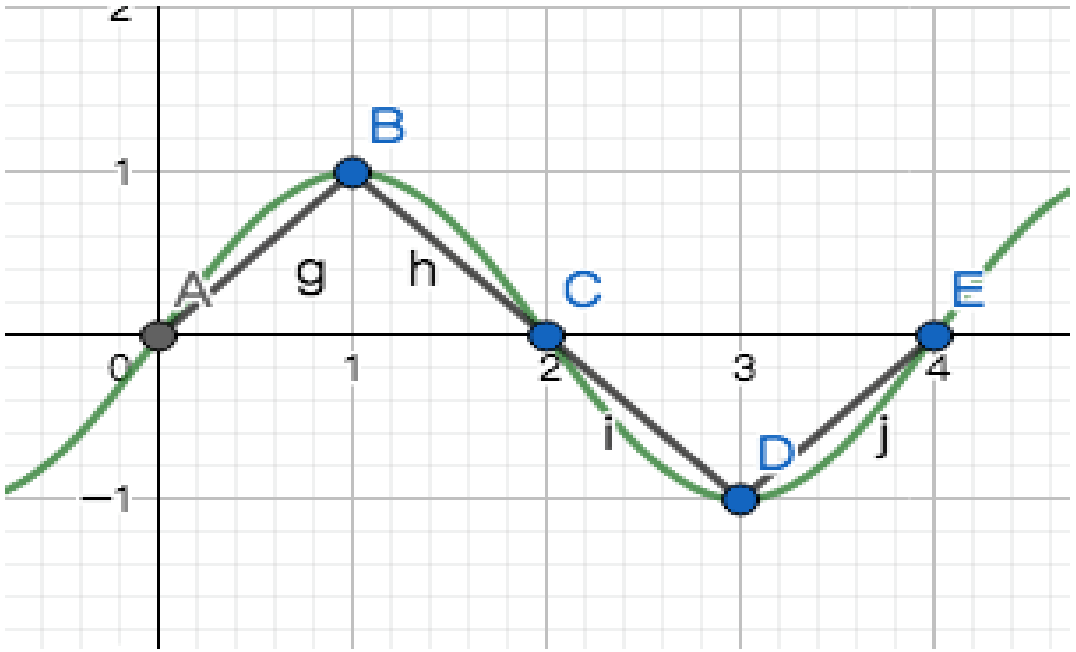


拟合A, B,C,D,E点的网络模型和对应的函数图像

隐藏层ReLU激活神经元与折线拟合

- 从上述手工设置网络参数拟合数据点的过程中发现的网络拟合特性。
  - 隐藏层的ReLU激活神经元，拟合类似于折线段的拟合过程，可以逐点的拟合数据点集。
  - 所搭建的网络模型能有效地拟合A-E五个点
  - 其拟合的函数图像仍与实际的Sin函数图像差距较大，存在着较大的误差。

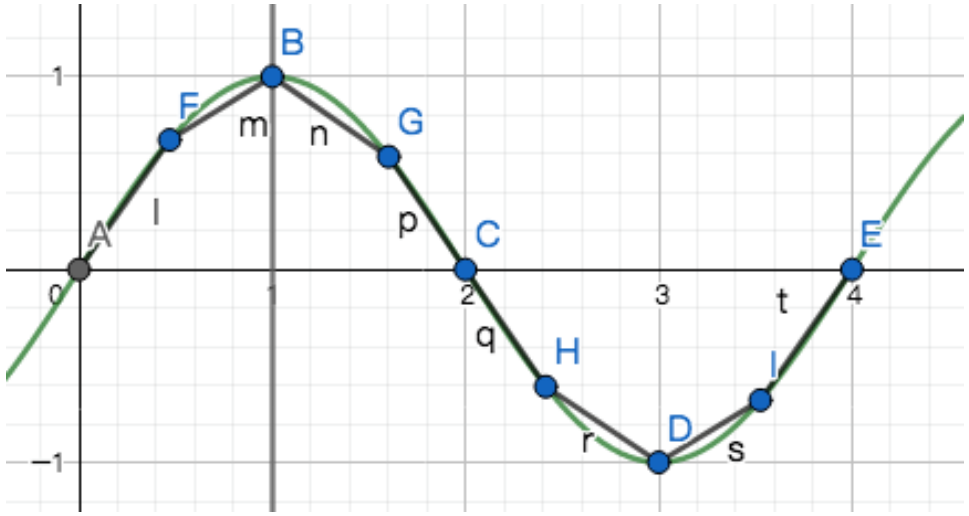




思考：基于以上思想，如何进一步更“贴合”地拟合sin函数

解决方法：在Sin函数上密集采样进行拟合

- 上述手工设置网络参数已经能有效拟合Sin函数上采样的A-E五个点.
- 扩增在Sin函数上进行采样的数据点集.
- 相应的扩大隐藏神经元的数量.



拟合更大的数据点集的函数图像

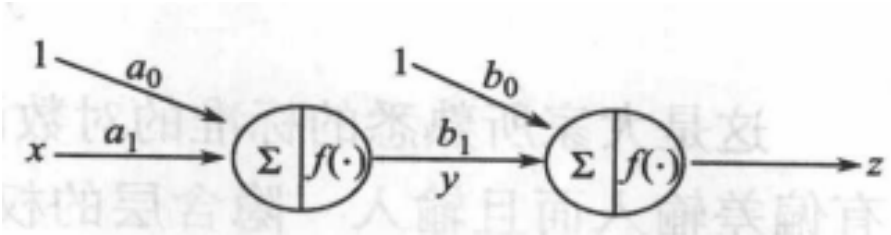
单隐层的神经网络

- 基于以上思想：

只要神经元个数足够多，理论上单隐层的神经网络具有拟合任何函数的能力。（采样足够多的数据点）

单输入多层非线性网络

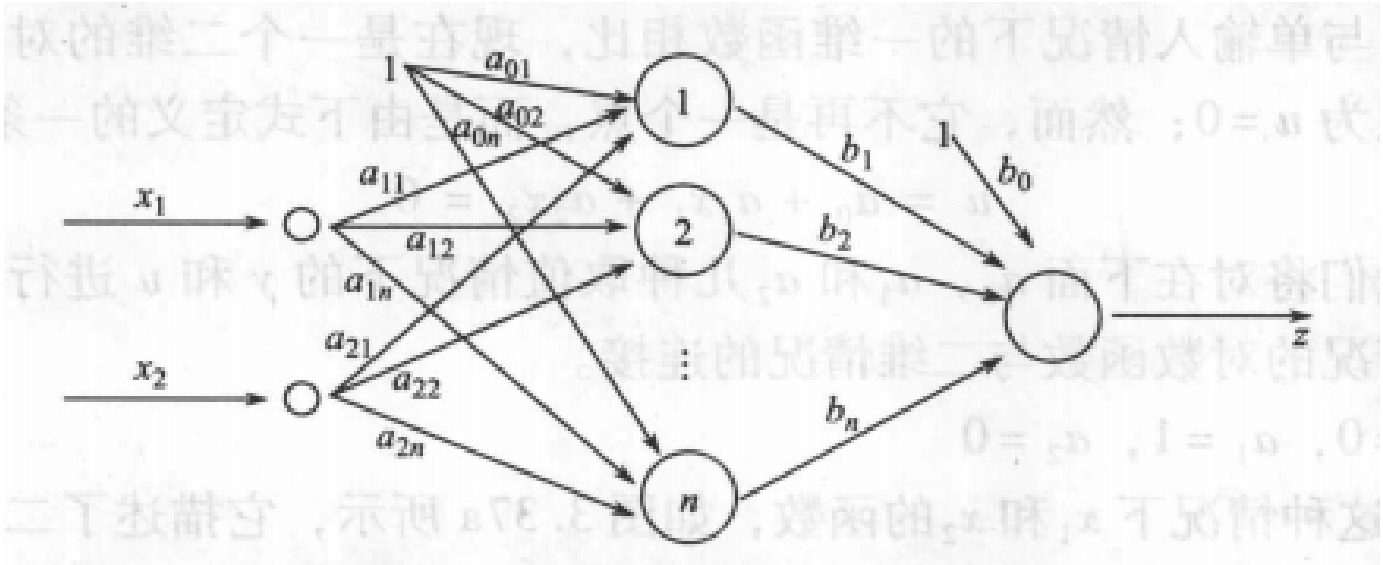
- 一个输入，一个隐含神经元，一个输出的神经网络



- 输入传到网络里，隐含神经元计算输入的加权和（包括偏差），并且将它传递到对数函数产生隐含层输出y。
- 隐含层输出y作为输入通过相关的连接，传入输出神经元。
- 输出神经元，计算加权的输入，再通过神经元的激励函数传递，得到整个网络的输出。

两输入多层非线性网络

- 网络结构：两个输入，一个或更多的隐含神经元和一个输出
- 当有两个输入时，网络需要更多的附加权值，因此权值的数目优化更复杂



双输入的多层网络结构

两输入多层非线性网络

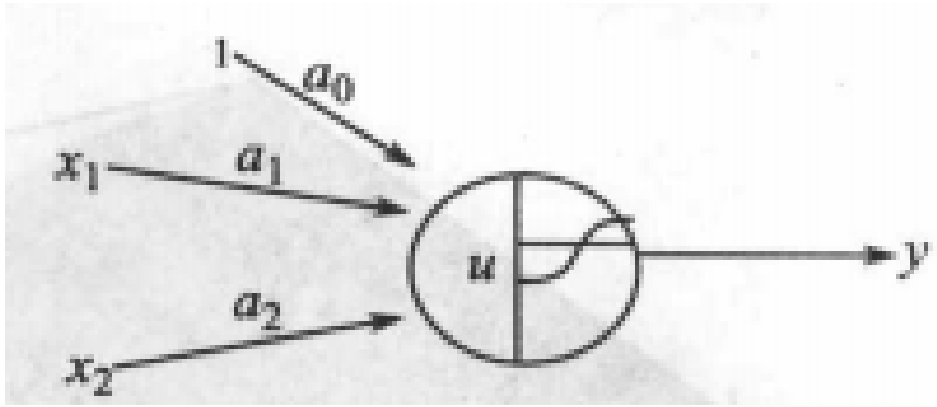
- 隐含神经元的处理:x_1和x_2为输入，a_0、a_1、a_2是偏差和输入隐含神经元的权值。
- u表示输入的加权和

$$u = a_0 + a_1 x_1 + a_2 x_2$$

- y是隐含神经元的输出

$$y = \frac{1}{1+e^{-u}}$$

- 假设所有的激励函数都是对数函数



两输入多层非线性网络

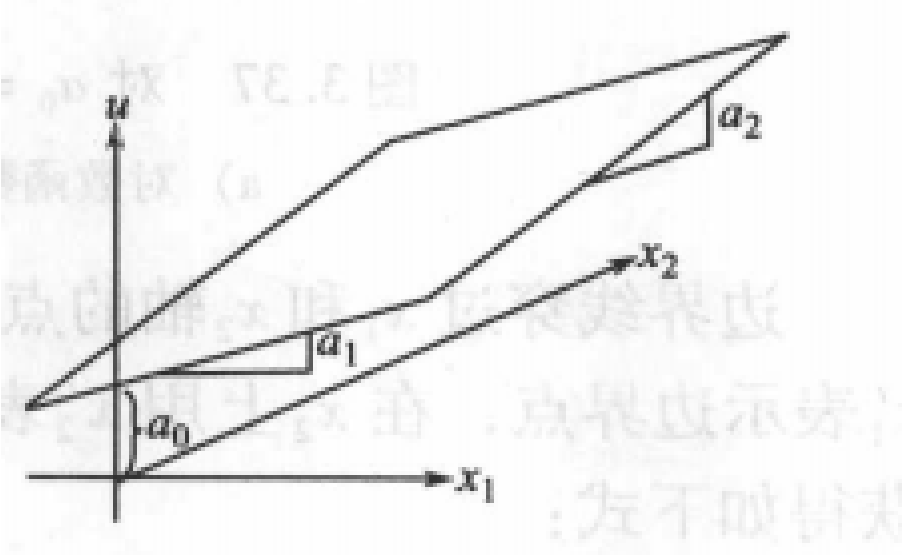
- 隐含神经元的处理:x_1和x_2为输入，a_0、a_1、a_2是偏差和输入隐含神经元的权值。
- u表示输入的加权和

$$u = a_0 + a_1 x_1 + a_2 x_2$$

这个关系在x_1和x_2的二维空间是一个平面

权重a_1控制平面相对于x_1轴的斜率

权重a_2控制平面相对于x_2轴的斜率



两输入多层非线性网络

- 隐含神经元的处理:x_1和x_2为输入，a_0、a_1、a_2是偏差和输入隐含神经元的权值。
- y是隐含神经元的输出

$$y = \frac{1}{1+e^{-u}}$$

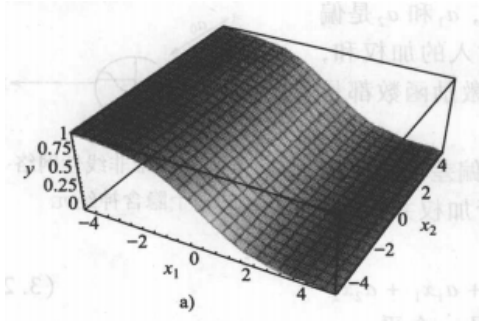
$$y = \frac{1}{1+e^{-(a_0+a_1x_1+a_2x_2)}}$$

边界定义为：$u = 0$,表示的为一条直线，而不是一个点。

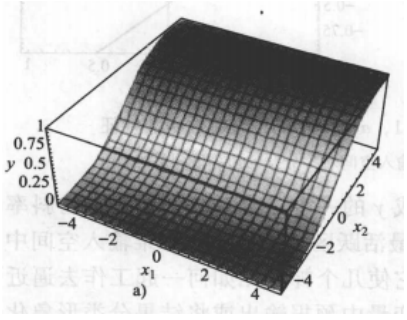
$$u = a_0 + a_1x_1 + a_2x_2 = 0$$

两输入多层非线性网络

- 隐含神经元的处理:
- 不同的a_0、a_1、a_2描述了二维空间中不同的对数函数。
- 斜率由权值a_1、a_2控制
- 最活跃区域转移由权值a_0控制

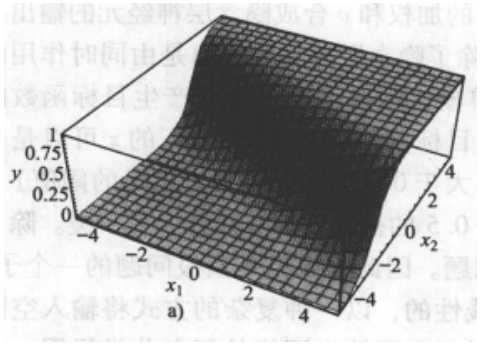


$a_0 = 0$、$a_1 = 1$、$a_2 = 0$

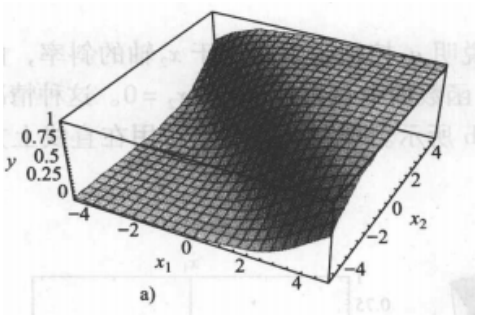


$a_0 = 0$、$a_1 = 0$、$a_2 = 1$

与一维输入情况相比较，二维输入的情况使得几个神经元如何一起工作去逼近一个二维函数或一个模型，如何从两个自变量中预报输出或将结果分类形象化成为可能。



$a_0 = 0$、$a_1 = 1$、$a_2 = 2$



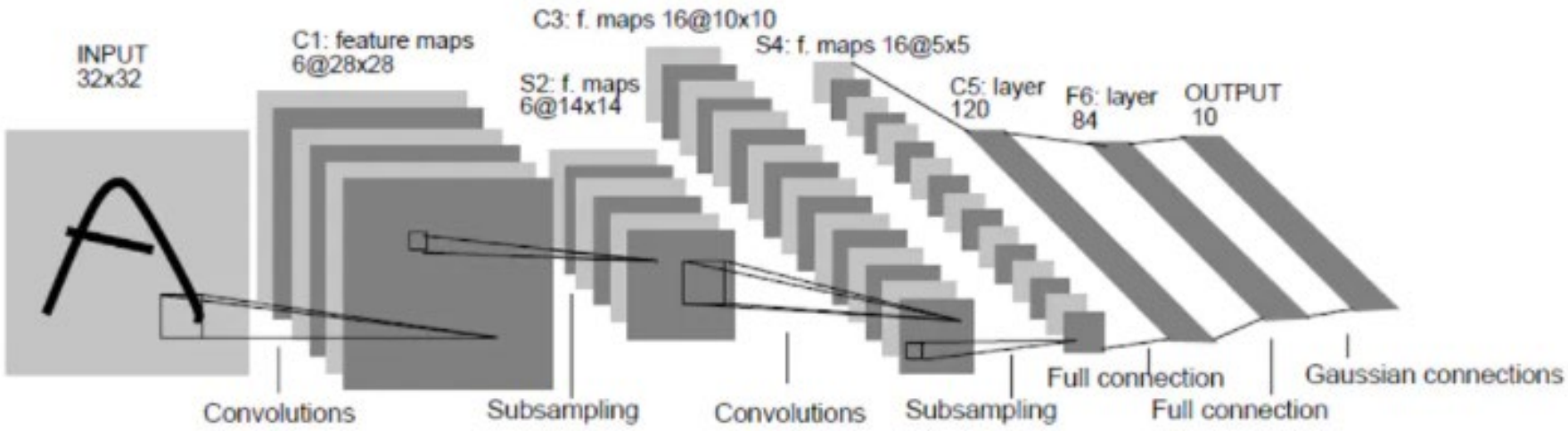
$a_0 = -0.5\ a_1 = 1\ a_2 = -$

五、其他连接方式

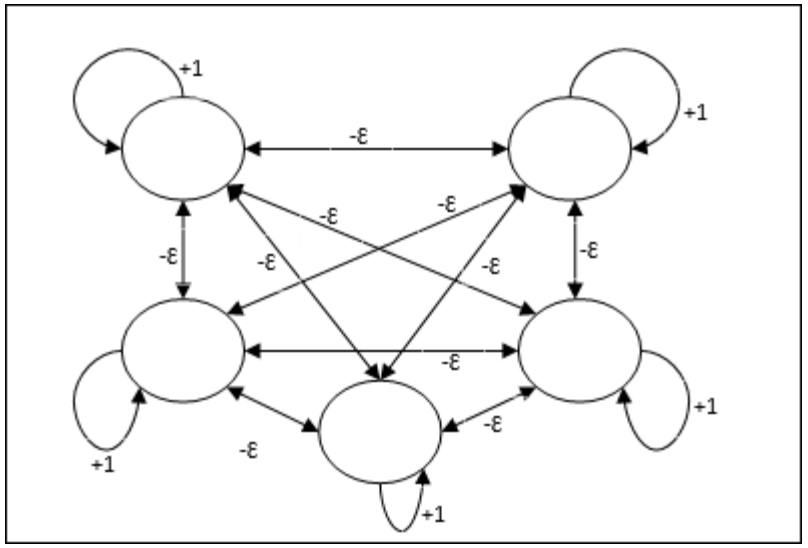
- 卷积神经网络（Convolutional Net）
- 竞争神经网络（Competitive Net）
- 循环神经网络（Recurrent Net）
- 其他类型网络

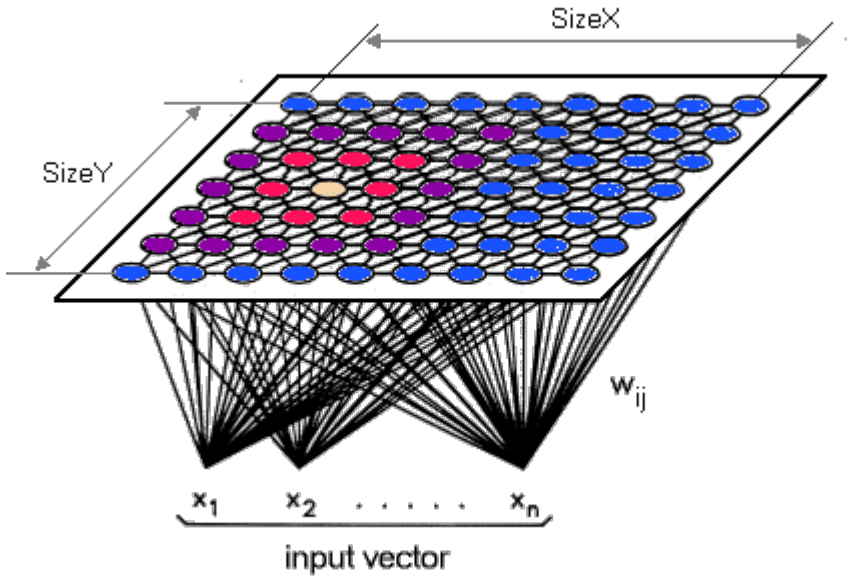
卷积神经网络
（Convolutional Net）

- 包含卷积计算且具有深度结构的前馈神经网络
- 主要结构：卷积层、池化层、全连接层



竞争神经网络
（Competitive Net）

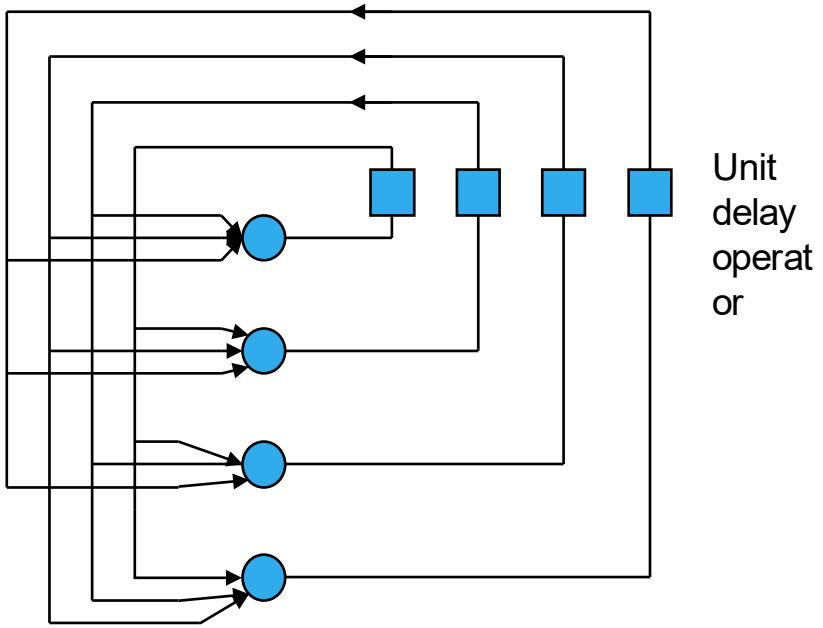




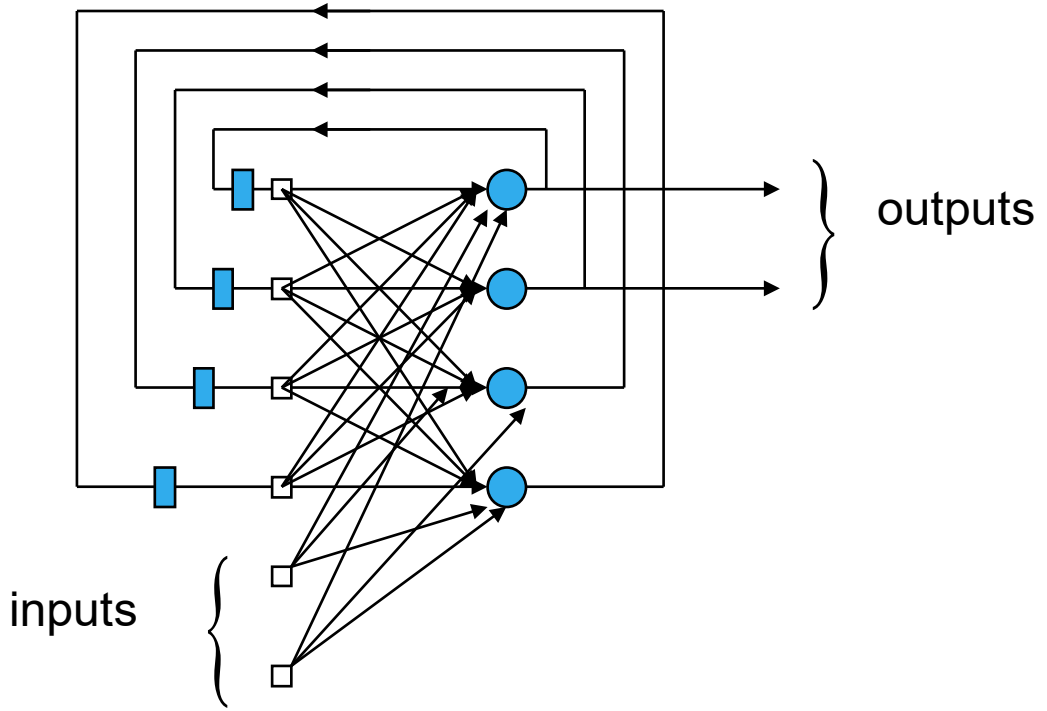
- 神经节点之间互相连接
- 很多神经网络采用竞争层作为其中一部分
- Winner-Take-All
- 例子：MAXNET、SOM

循环神经网络（Recurrent Net）

- 以序列数据作为输入，在序列的演进方向进行递归且所有节点按链式连接的递归神经网络



无隐藏层

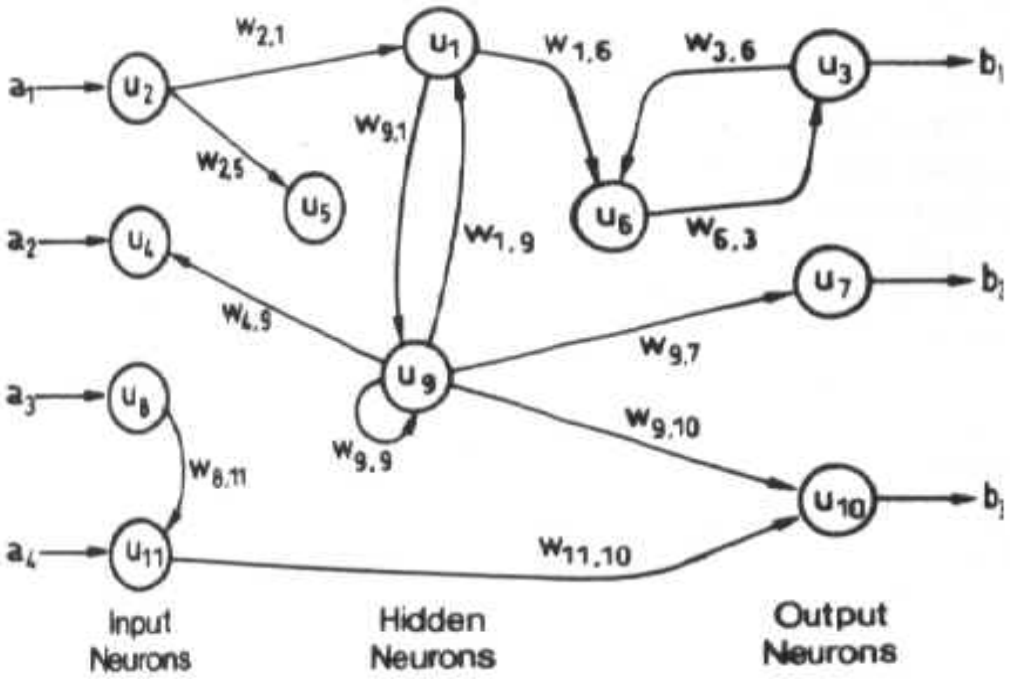


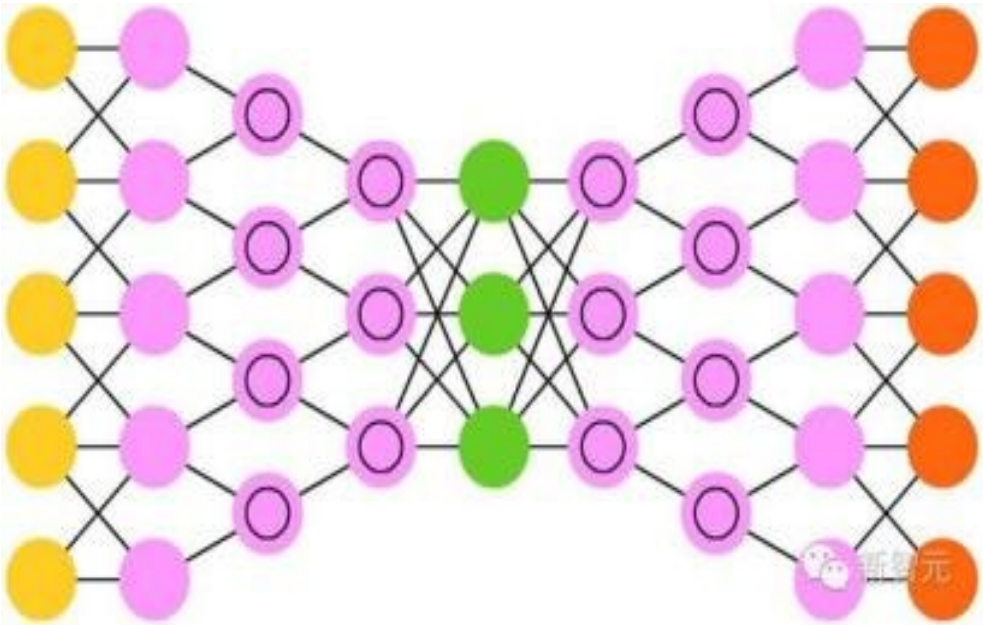
有隐藏层

循环网

- 如果将输出信号反馈到输入端,就可构成一个多层的循环网络。
- 输入的原始信号被逐步地“加强”、被“修复”。
- 大脑的短期记忆特征——看到的东西不是一下子就从脑海里消失的。
- 稳定：反馈信号会引起网络输出的不断变化。我们希望这种变化逐渐减小，并且最后能消失。当变化最后消失时，网络达到了平衡状态。如果这种变化不能消失，则称该网络是不稳定的。

其他神经网络





六、思考题

- 如下函数适合作为神经网络的激活函数吗？请说明理由。

$$f(x)=\begin{cases}0, x<0\\ 1, x\geq 0\end{cases}$$

- 有如下神经网络，隐藏层使用relu激活函数，输出层使用sigmoid激活函数且没有偏置。对于输入$X_i=\{i_1,i_2\}$，其对应输出记为Y_i'，该数据的真实标签记为Y_i，这里$i\in\{1,2,\dots,n\}$，损失函数定义为

$$E=\sum_{i=1,..,n}(Y_i'-Y_i)^2$$

请推导该损失函数对于w_1,b_2,w_5的偏导。当w_1,w_2,w_3,w_4,w_5,w_6分别取值$0.4,0.5,0.2,0.4,3.5,0.6$，$b1=0.3,b2=0.8$，且只有一条输入数据$X_1=\{0.3,2.8\},Y_1=5.6$的时候，计算该损失函数对于w_3的具体值，结果保留两位小数。