人工智能概论

慕彩红
人工智能学院
mucaihongxd@foxmail.com

第八章 神经网络及机器学习

神经网络的生物学基础

大脑含有大量（约10^{11}个）高度连接的神经元。神经元是神经系统的基本单元，它们按不同的结合方式构成了复杂的神经网络。通过神经元及其联接的可塑性，使得大脑具有学习、记忆和认知等各种智能。

生物神经元模型

胞体：是神经细胞的本体；
树突：用以接受来自其它细胞元的信号；
轴突：用以输出信号，与多个神经元连接；
突触：是一个神经元与另一个神经元相联系的特殊部位，通过神经元轴突的端部靠化学接触和电接触将信号传递给下一个神经元的树突。

工作机制：每个神经元根据上一个神经元传入的信号会处于抑制或激发状态，并产生相应的信号传给下一个神经元；众多神经元相互连接，最终产生对输入信号的认知。

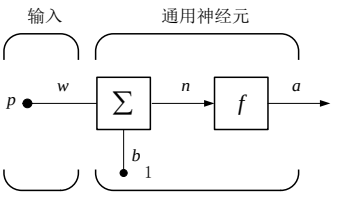
生物神经元的工作状态

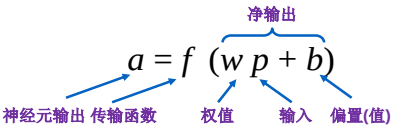
- 生物神经元有两种工作状态——兴奋和抑制。
- 平时处于抑制状态的神经元，其树突和细胞体接受其它神经元经由突触传来的兴奋电位，多个输入在神经元中以代数和的方式叠加；
- 如输入兴奋总量超过阈值，神经元被激发进入兴奋状态，发出输出脉冲，由轴突的突触传递给其它神经元。

人工神经网络

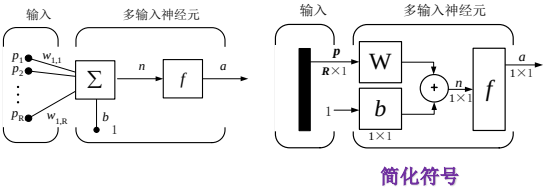
- 人工神经网络（Artificial Neural Nets，ANN）是由大量处理单元经广泛互连而组成的人工网络，用来模拟脑神经系统的结构和功能。而这些处理单元称作人工神经元。

单输入人工神经元





多输入神经元



$$a = f\ (\mathbf{W}\boldsymbol{p} + b)$$

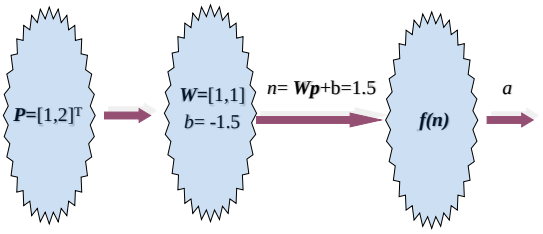
传输函数总结

名称	输入/输出关系	图标	Matlab函数
硬极限函数	$\begin{cases} a=0 & n<0 \\ a=1 & n\geq 0 \end{cases}$		Hardlim
对称极限函数	$\begin{cases} a=-1 & n<0 \\ a=1 & n\geq 0 \end{cases}$		Hardlims
线性函数	$a=n$		Pureline
饱和线性函数	$\begin{cases} a=0 & n<0 \\ a=n & 0\leq n\leq 1 \\ a=1 & n\geq 1 \end{cases}$		Satlin

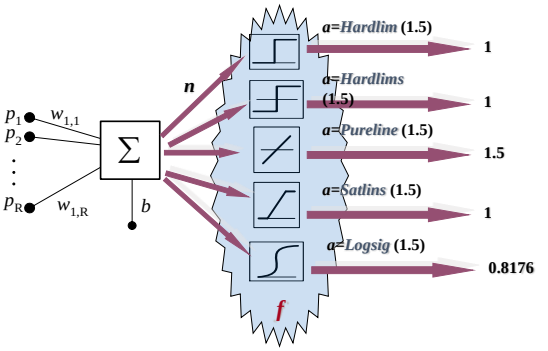
传输函数总结

对称饱和线性函数	$\begin{cases} a=-1 & n<0 \\ a=n & 0\leq n\leq 1 \\ a=1 & n\geq 1 \end{cases}$		Satlins
对数S型函数	$a=\frac{1}{1+e^{-n}}$		Logsig
双曲正切S型函数	$a=\frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$		Tansig
正线性函数	$\begin{cases} a=0 & n<0 \\ a=n & n\geq 0 \end{cases}$		Poslin
竞争函数	$\begin{cases} a=1 & \max(n) \\ a=0 & \text{其它} \end{cases}$		Compet

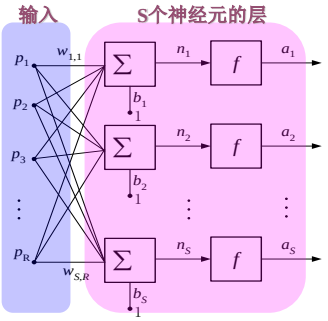
传输函数作用的实例



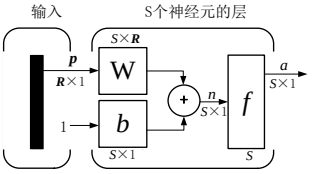
传输函数作用的实例



网络结构—神经元的层



神经元的层简化模型

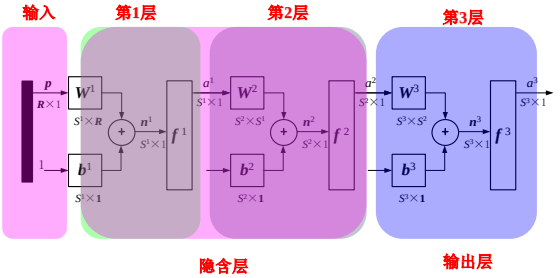


$$a = f\ (Wp + b)$$

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_S \end{bmatrix}$$

多层神经网络（3层）简化表示



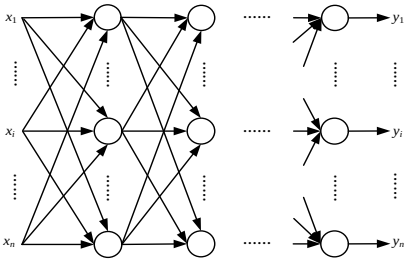
$$a^3=f^3(W^3f^2(W^2f^1(W^1p+b^1)+b^2)+b^3)$$

神经网络中的常见模型

- 神经网络中的常见模型：
- 前馈型神经网络（也叫前向神经网络）
- 反馈型神经网络（也叫递归神经网络）

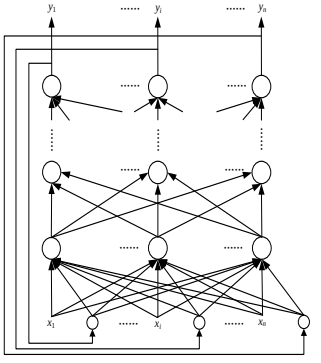
前馈型网络

- 前馈型网络的信号由输入层到输出层单向传输
- 每层的神经元仅与其前一层的神经元相连，仅接受前一层传输来的信息
- 是一种最为广泛使用的神经网络模型，因为它本身的结构也不太复杂，学习和调整方案也比较容易操作，而且由于采用了多层的网络结构，其求解问题的能力也得到明显的加强，基本上可以满足使用要求



反馈型网络

- 这种网络结构在输入输出之间还建立了另外一种关系，就是网络的输出层存在一个反馈回路到输入层作为输入层的一个输入，而网络本身还是前馈型的
- 这种神经网络的输入层不仅接受外界的输入信号，同时接受网络自身的输出信号。输出反馈信号可以是原始输出信号，也可以是经过转化的输出信号；可以是本时刻的输出信号，也可以是经过一定延迟的输出信号
- 此种网络经常用于系统控制、实时信号处理等需要根据系统当前状态进行调节的场合



感知机

- **1943年，McCulloch和Pitts提出首个神经元模型（M-P模型），该模型是一个简单的二元分类器。**
- **1957年，Rosenblatt定义了一个称为感知机（perceptron）的神经网络结构，该模型与神经元模型类似，主要有以下几点改进：**
- **输入为一个实数向量；**
- **有多种激活函数可以选择；**
- **属于一个可学习模型；**
- **感知机中第一次引入了学习的概念，使人脑所具备的学习功能在基于符号处理的数学模型中得到了一定程度的模拟，所以引起了广泛的关注。**
- **简单感知机：简单感知机模型实际上仍然是M-P模型的结构。它是一种单层感知机模型，一层为输入层（只负责接收输入信号，无信息处理能力），另一层具有计算单元，可以通过采用监督学习来逐步增强模式划分的能力，达到学习的目的。**

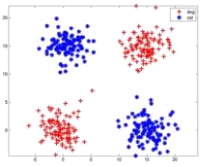
感知机算法

感知机算法实质上是一种赏罚过程：

- 对正确分类的模式则“赏”，实际上是“不罚”，即权向量不变。
- 对错误分类的模式则“罚”，需要对权向量进行相应的更新。
- 当用全部模式样本训练过一轮以后，只要有一个模式是判别错误的，则需要进行下一轮迭代，即用全部模式样本再训练一次。
- 如此不断反复直到全部模式样本进行训练都能得到正确的分类结果为止。

感知机模型的缺陷

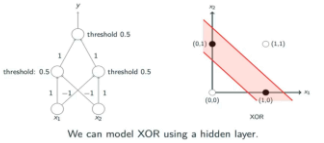
1969年：Minsky和Papert指出感知机的缺陷：仅能解决一阶谓词逻辑，即只能完成线性划分，对于非线性或者其他分类会遇到很多困难，就连简单的 XOR（异或）问题都解决不了。



对于左图中的分类问题，单层感知机模型无法找到一个超平面将两类样本区分开。

多层感知机的提出

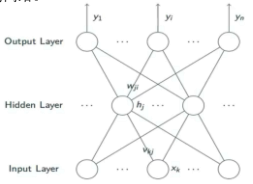
异或问题无法用一个超平面将两类样本分隔开，于是人们考虑对多个感知机模型进行组合，即采用多个超平面去分割样本——多层感知机



We can model XOR using a hidden layer.

多层感知机的难题

多层感知机具有多个神经元，之前神经元模型的学习方法不适用于多个神经元组成的网络。



多层感知机带来大量的权重，需要一种高效的学习方法来训练这样的网络。

梯度下降的思想

人们通常使用均方误差来衡量预测值与真实值之间的差距，此处考虑单输出的情况，即：

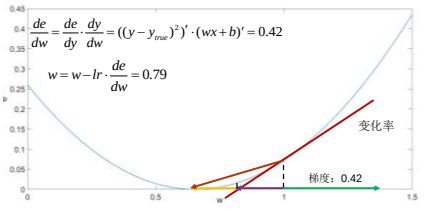
$$e = |\mathbf{y} - \mathbf{y}_{true}|^2$$

对于神经网络模型：

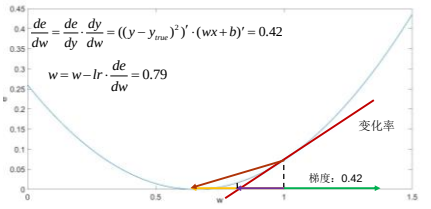
$$\mathbf{y} = \mathbf{wx} + \mathbf{b}$$

我们训练神经网络的目的实际上是通过确定合适的w，b使得误差e尽可能地变小；要想知道w，b是如何影响误差e的，我们可以用误差e对w，b进行求导；导数的方向为误差e上升最快的方向，我们只需将w，b向着导数的相反方向进行调整，即可使误差e减小——梯度下降。

例：
若预测值：$y = wx + b = 1.78$，真实值$y_{true} = 1.51$，其中w=1，b=1，x=0.78，试用梯度下降法对w进行调整，假设学习率lr=0.5。



注意：
为了使w能较快地收敛于一个使函数e极小的解，lr值的选择是很重要的。
- 若lr值太小，则收敛太慢；
- 若lr值太大，则搜索可能过头，引起发散。



反向传播算法（BP）的提出

1986年：Rumelhart等人提出的反向传播算法，为多层网络的训练提供了有力的工具。

为什么需要反向传播算法：梯度下降可以应对带有明确求导函数的情况，或者说可以应对那些可以求出误差的情况，我们可以把它看做没有隐层的网络；但对于多隐层的神经网络，输出层可以直接求出误差来更新参数，但其中隐层的误差是不存在的，因此不能对它直接应用梯度下降，而是先将误差反向传播至隐层，然后再应用梯度下降，其中将误差从末层往前传递的过程需要链式法则（Chain Rule）的帮助，因此反向传播算法可以说是梯度下降在链式法则中的应用。

反向传播算法（BP）

- 多层网络的学习能力比单层感知机强得多。要训练多层网络，简单感知机学习规则显然不够，需要更强大的学习算法。
- 误差反向传播(error Back Propagation, BP)算法就是其中最杰出的代表，它是迄今最成功的神经网络学习算法。
- 现实任务中使用神经网络时，大多是在使用BP算法进行训练。BP算法不仅可用于多层前馈神经网络，还可用于其他类型的神经网络，如训练递归神经网络。

反向传播算法（BP）

- 在感知器算法中我们实际上是在利用理想输出与实际输出之间的误差作为增量来修正权值，然而在多层感知器中，我们只能计算出输出层的误差，中间隐层由于不直接与外界连接，其误差无法估计。
- 反向传播算法（BP算法）的思想：从后向前反向逐层传播输出层的误差，以间接计算隐层的误差。算法可以分为两个阶段：
  - 正向过程：从输入层经隐层逐层正向计算各单元的输出
  - 反向过程：由输出误差逐层反向计算隐层各单元的误差，并用此误差修正前层的权值。

反向传播算法学习过程

（1）选择一组训练样本，每一个样本由输入信息和期望的输出结果两部分组成。

（2）从训练样本集中取一样本，把输入信息输入到网络中。

（3）分别计算经神经元处理后的各层结点的输出。

（4）计算网络的实际输出和期望输出的误差。

（5）从输出层反向计算到第一个隐层，并按照某种能使误差向减小方向发展的原则，调整网络中各神经元的连接权值。

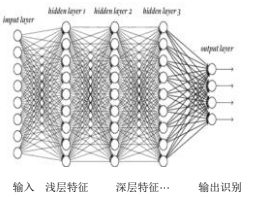
（6）对训练样本集中的每一个样本重复（3）-（5）的步骤，直到对整个训练样本集的误差达到要求时为止。

反向传播算法优缺点

- 优点：
- 理论基础牢固，推导过程严谨，物理概念清晰，通用性好等。所以，它是目前用来训练前馈多层网络较好的算法。
- 缺点：
- BP算法的收敛速度一般来说比较慢；
- BP算法只能收敛于局部最优解，不能保证收敛于全局最优解；
- 当隐层神经元的数量足够多时，网络对训练样本的识别率很高，但对测试样本的识别率有可能很差，即网络的泛化能力有可能较差。

神经网络的隐层到底在学什么？

有了反向传播算法后，神经网络变得越来越深，并且随着神经网络隐藏层的增加，神经网络的表示能力也越来越强。一种通俗的理解是：隐藏层的神经元会提取输入中的特征，并且随着神经网络层数的增加，深层网络会将浅层网络中获得的特征进一步抽象，得到更高级的特征。



神经网络的隐层可视化

2012年多伦多大学的Krizhevsky等人构建了一个超大型卷积神经网络，该网络有9层，65万个神经元，6千万个参数。网络的输入是图片，输出是1000个类，对应图片中物体的类别，如小虫、美洲豹、救生船等等。随后纽约大学的Zeiler和Fergusi对这个训练好的网络进行了可视化，他们将网络中的某些神经元挑选出来，将其与输入图像进行对比，发现中间层的神经元响应了某些十分抽象的特征。

神经网络的隐层可视化

第一层神经元主要负责识别颜色和简单纹理。

神经网络的隐层可视化

第二层的一些神经元可以识别更加细化的纹理，比如布纹、刻度、叶纹。

神经网络的隐层可视化

第三层的一些神经元负责感受黑夜里的黄色烛光、鸡蛋黄、高光。

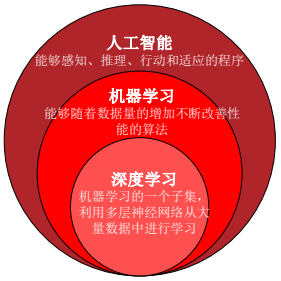
神经网络的隐层可视化

第四层的一些神经元负责识别萌狗的脸，七星瓢虫和一堆圆形物体的存在。

神经网络的隐层可视化

可以发现，输入在网络中的传递，实际上是一个特征提取的过程。浅层的网络提取出颜色、纹理等初级特征，深层的网络对这些简单特征进一步抽象，获得更高级的特征，最后输出层将这些高级特征映射到我们想要的输出，如所属类别、预测结果、生成样本等。

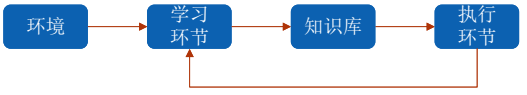
人工智能、机器学习与深度学习



- 人工智能、机器学习和深度学习是非常相关的几个领域。
- 人工智能是一类非常广泛的问题，机器学习是解决这类问题的一个重要的手段。而深度学习是机器学习的一个分支。
- 在很多人工智能问题上，深度学习的方法突破了传统机器学习方法的瓶颈，推动了人工智能领域的发展。

- 机器学习系统的基本结构

根据西蒙（Simon）关于学习的定义，我们可以构建如下机器学习系统的基本模型。



环境和知识库分别代表外界信息来源和系统具有的知识。

“学习环节”则利用“环境”中的信息对系统的“知识库”进行改进，以增进系统“执行环节”完成任务的效能。

“执行环节”根据知识库中的知识来完成某种任务，同时把获得的信息反馈给“学习环节”。

在具体的应用中，环境，知识库和执行环节决定了具体的工作内容，学习环节所需要解决的问题完全由上述三个部分决定。

- 机器学习方法的分类

按照不同的分类标准，机器学习可以有多种分类。例如：

按学习方式分类
- 监督学习
- 无监督学习
- 半监督学习
- 强化学习

此外，还可以按照学习方法、数据形式、学习目标、学习策略等标准分类。

- 机器学习方法的一些术语

一个样本　一种特征

编号	色泽	根蒂	敲声
1	青绿	蜷缩	浊响
2	乌黑	硬挺	清脆
3	浅白	稍蜷	沉闷
…	…	…	…

生活中，我们可以根据西瓜的色泽、根蒂、敲声这几种特征来判断一个西瓜的好坏。

假如我们要使用机器学习的方法去判断西瓜的好坏，并且收集到了如表中所示的数据。

我们把所有数据的集合称为一个**数据集**。其中的每条记录是对某个对象的描述，称为一个**样本**。

反映对象某方面的性质的事项，如表中的“色泽”、“根蒂”、“敲声”称为属性或**特征**。

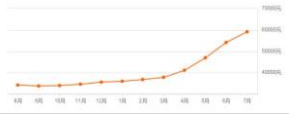
从数据中学习模型的过程称为“学习”或“**训练**”，训练过程中使用的样本称为**训练样本**。有时候，我们为了训练模型还需要样本的“结果信息”，例如1号西瓜的结果为“好瓜”，我们称这种结果信息为“**标签**”。而将一条拥有标签的样本称为一个“**样例**”。

- 监督学习

在监督学习中，训练样本的标签信息是已知的，学习目标是通过对样本以及对应标签的学习，得到样本特征与标签之间的内在联系，从而实现对未知样本的标签进行预测。

根据样本标签是连续还是离散，监督学习又可以分为**回归**和**分类**。

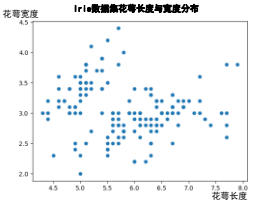
例如我们可以通过西瓜的瓜蒂、色泽、敲声来训练模型，把西瓜分为好瓜和坏瓜，这就是一种分类。而我们通过对过去三年的各地段各房型房价变化推测今年的房价，就是一种回归。



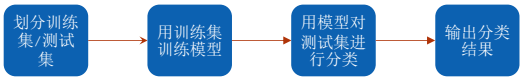
- 监督学习的流程

Iris数据集是常用的分类实验数据集，它包含150条鸢尾花的样本。每条样本包含花萼长度，花萼宽度，花瓣长度，花瓣宽度4个特征。这些样本根据特征的不同分为三类：Iris Setosa（山鸢尾）、Iris Versicolour（杂色鸢尾），以及Iris Virginica（维吉尼亚鸢尾），每类各有50条样本。

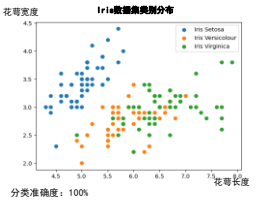
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
…
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
…
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica
…



- 监督学习的流程



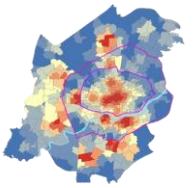
1 我们从每个类别随机选取10个样本组成训练样本集，剩余样本作为测试集。
2 使用训练样本集训练一个分类模型；
3 使用训练好的模型对测试集样本进行分类；
4 将分类结果输出。



- 无监督学习

在无监督学习中，训练样本的标签信息是未知的，目标是通过对无标签训练样本的学习来揭示数据内在的性质及规律，为进一步的数据分析提供基础。此类学习任务中研究最多、应用最广的是“聚类”。

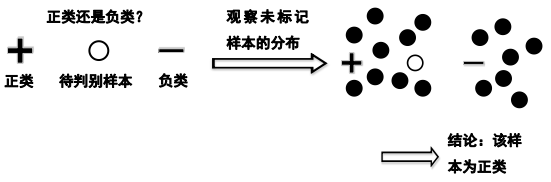
例如我们可以将一张城市地图按照建筑物密集程度分成不同的区块。这就是一种聚类。



- 半监督学习

无标签样本虽然没有直接包含标签信息，但是如果它们与有标记样本是从同样的数据源中独立同分布采样而来，则它们包含的关于数据分布的信息是十分有用的。如图所示，如果仅基于有标签样本很难判断待判别样本的类别，而结合无标签样本的分布情况，则可以判断它属于正类。

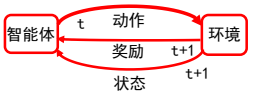
既利用标记样本进行学习，又利用未标记样本来提升学习性能，就是半监督学习。



- 强化学习

强化学习是20世纪80年代兴起的，受行为心理学启发的机器学习方法。它关注身处某个环境的决策器通过采取行动获得最大化的累积收益。AlphaGo击败围棋世界冠军，其核心技术就是深度学习与强化学习的结合。

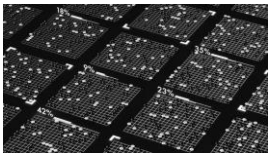
强化学习主要由环境、智能体、状态、动作、奖励等基本概念构成。智能体在环境中会做各种动作，环境可能因此改变自身状态，同时环境会给智能体以奖励。智能体的目标就是使用一些策略，做合适的动作，取得最大化的累积收益。



- 强化学习

在棋类问题中，棋手面对棋局有多种走棋策略可以选，而对于不同的走棋策略，对手也会有不同的应对，好的走棋可以增大赢面，不好的走棋可能会导致失败。棋手只有每一步都采取合适策略，不断扩大自己的赢面，才能最后取得胜利。这是一个典型的强化学习场景。

AlphaGo主要的网络结构中，就包含了强化学习的策略网络和强化学习的估值网络。前者通过强化学习过程自我对弈，优化最终受益，改进策略网络，后者是预测网络采取不同策略后的胜率。



小结

人工神经元模型
感知机
多层感知机
神经网络中的常见模型
BP算法
机器学习系统的基本结构
机器学习分类（按学习方法分类）