

# 主要内容

**5.1 神经网络的基本原理**

**5.2 感知器**

**5.3 BP网络及其算法**

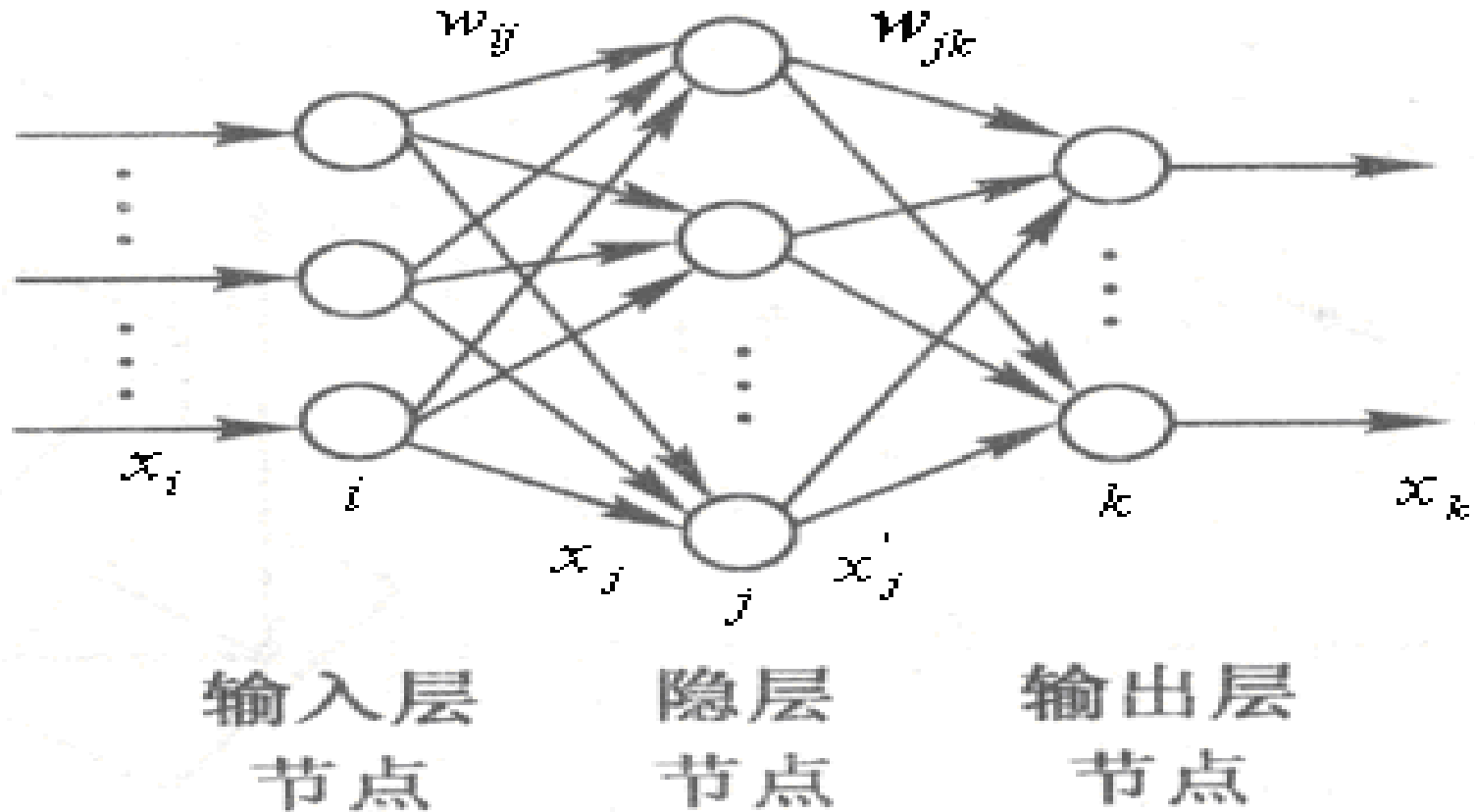
## 5.3 BP网络及其算法

1986年，Rumelhart等提出了误差反向传播神经网络，简称BP网络（Back Propagation），该网络是一种单向传播的多层前向网络。

误差反向传播的BP算法简称BP算法，其基本思想是**梯度下降法**。它采用**梯度搜索技术**，以期使网络的**实际输出值与期望输出值的误差均方值为最小**。

# 1 BP网络结构

含一个隐含层的BP网络结构如下图所示，图中  $i$  为输入层神经元， $j$  为隐层神经元， $k$  为输出层神经元。



BP神经网络结构

## 2 BP网络的特点

(1) 是一种多层网络，包括输入层、隐含层和输出层；

(2) 层与层之间采用全互连方式，同一层神经元之间不连接；

(3) 权值通过  $\delta$  学习算法进行调节；

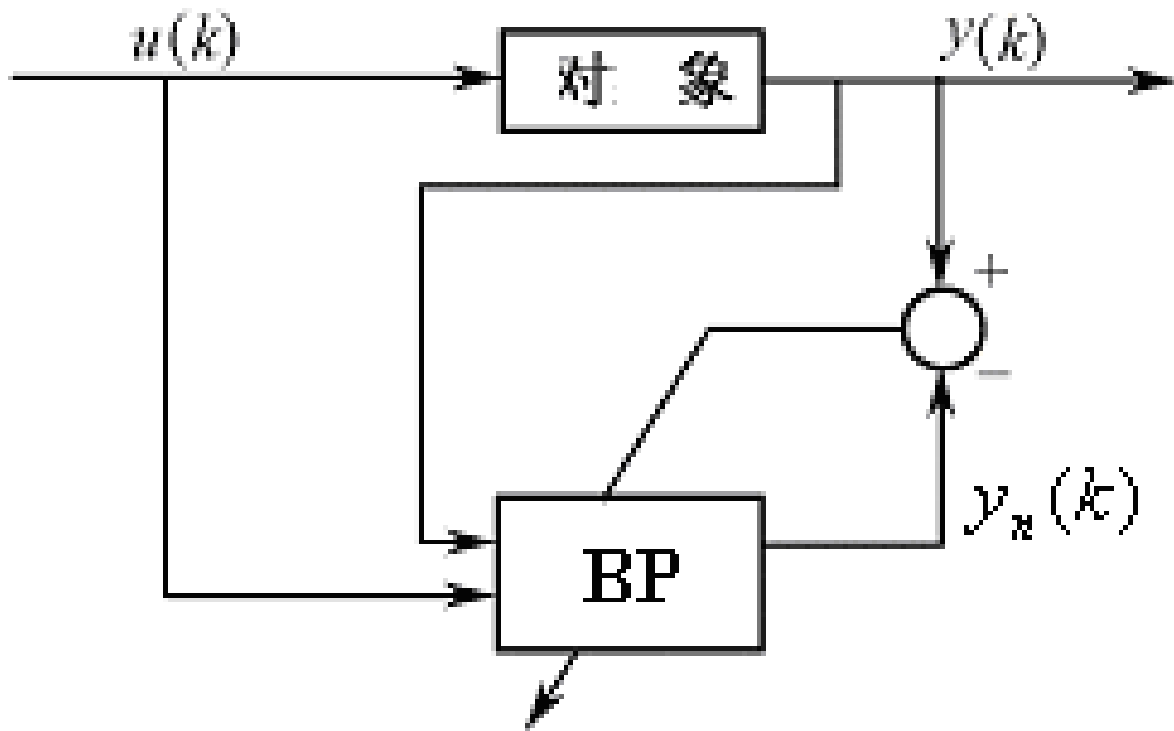
(4) 神经元激发函数为S函数；

(5) 学习算法由正向传播和反向传播组成；

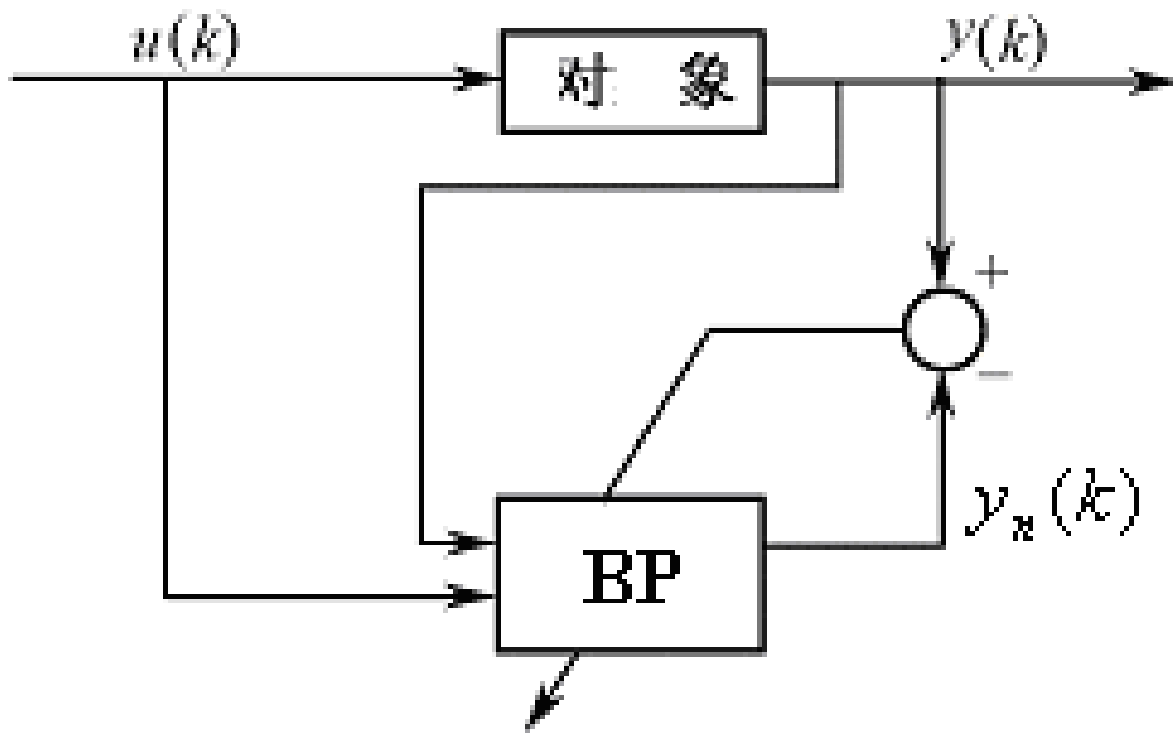
(6) 层与层的连接是单向的，信息的传播是双向的。

### 3 BP网络的逼近

BP网络逼近的结构如下图所示：

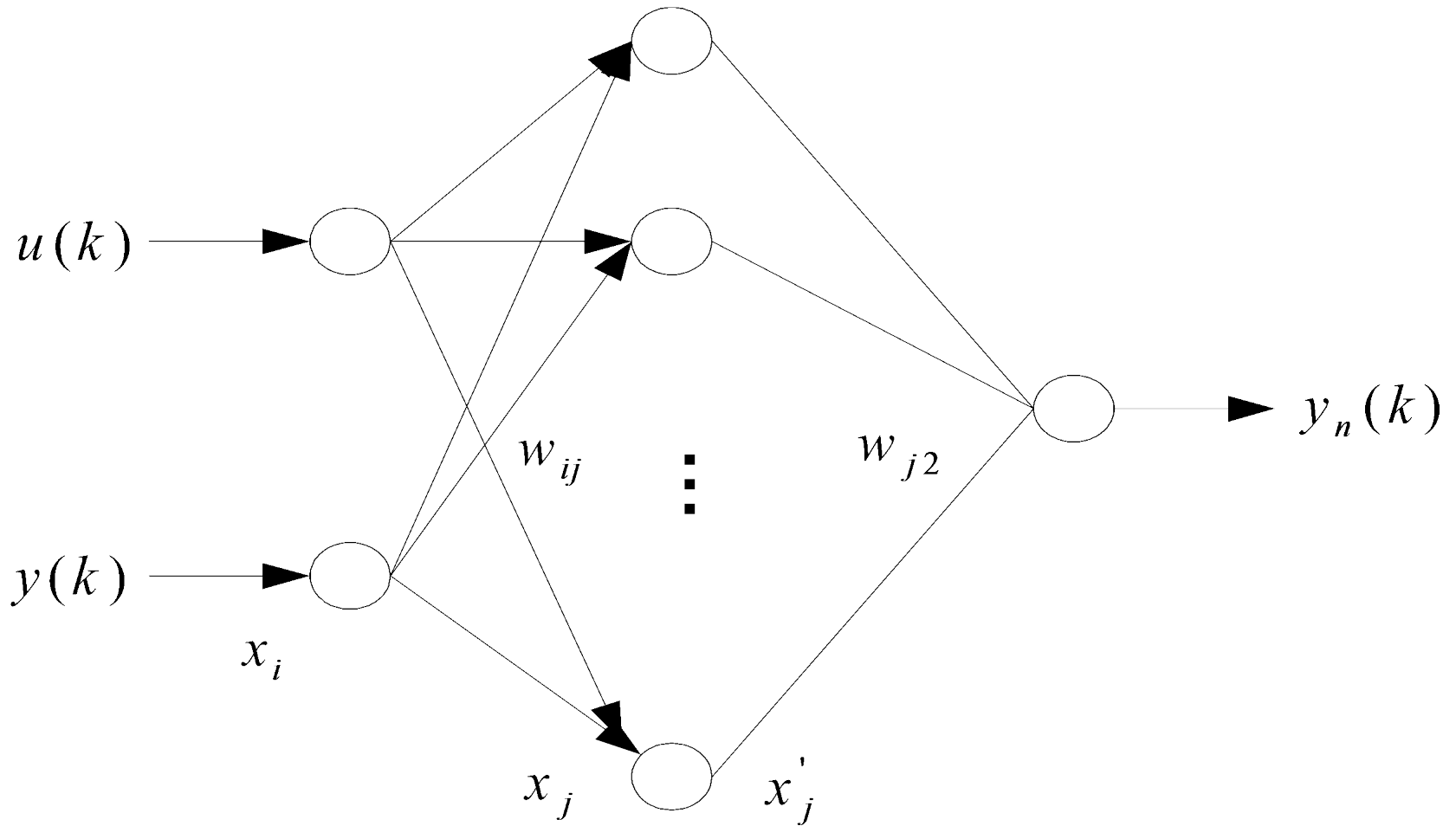


$k$ 为网络的迭代步骤， $u(k)$ 和 $y(k)$ 为逼近器的输入。BP为网络逼近器， $y(k)$ 为被控对象实际输出， $y_n(k)$ 为BP的输出。



将系统输出  $y(k)$  及输入  $u(k)$  的值作为逼近器**BP**的输入，将系统输出与网络输出的误差作为逼近器的调整信号。

用于逼近的BP网络如下图所示。



用于逼近的BP网络

**BP**算法的学习过程由**正向传播**和**反向传播**组成。

在**正向**传播过程中，**输入**信息从**输入层**经**隐层**逐层处理，并传向**输出层**，每层神经元（节点）的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出，则转至反向传播，将**误差信号**（理想输出与实际输出之差）**按联接通路反向计算**，由**梯度下降法**调整各层神经元的权值，使误差信号减小。



## (1) 前向传播：计算网络的输出

隐层神经元的输入为所有输入的加权之和：

$$x_j = \sum_i w_{ij} x_i$$

隐层神经元的输出采用S函数激发：

$$x'_j = f(x_j) = \frac{1}{1 + e^{-x_j}}$$

则

$$\frac{\partial x'_j}{\partial x_j} = x'_j (1 - x'_j)$$

输出层神经元的输出:

$$y_n(k) = \sum_j w_{j2} x_j'$$

网络输出与理想输出误差为:

$$e(k) = y(k) - y_n(k)$$

误差性能指标函数为:

$$E = \frac{1}{2} e(k)^2$$

(2) 反向传播：采用  $\delta$  学习算法，调整各层间的权值。

根据梯度下降法，权值的学习算法如下：  
输出层及隐层的连接权值学习算法为：

$$\Delta w_{j2} = -\eta \frac{\partial E}{\partial w_{j2}} = \eta \cdot e(k) \cdot \frac{\partial y_n}{\partial w_{j2}} = \eta \cdot e(k) \cdot x'_j$$

$k+1$ 时刻网络的权值为：

$$w_{j2}(k+1) = w_{j2}(k) + \Delta w_{j2}$$

隐层及输入层连接权值学习算法为：

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \cdot e(k) \cdot \frac{\partial y_n}{\partial w_{ij}}$$

其中

$$\frac{\partial y_n}{\partial w_{ij}} = \frac{\partial y_n}{\partial x'_j} \cdot \frac{\partial x'_j}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}} = w_{j2} \cdot \frac{\partial x'_j}{\partial x_j} \cdot x_i = w_{j2} \cdot x'_j(1 - x'_j) \cdot x_i$$

**k+1**时刻网络的权值为：

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}$$

如果考虑上次权值对本次权值变化的影响，  
需要加入动量因子  $\alpha$  此时的权值为：

$$w_{j_2}(k+1) = w_{j_2}(k) + \Delta w_{j_2} + \alpha(w_{j_2}(k) - w_{j_2}(k-1))$$

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} + \alpha(w_{ij}(k) - w_{ij}(k-1))$$

其中， $\eta$  为学习速率， $\alpha$  为动量因子。

$$\eta \in [0,1] \quad \alpha \in [0,1]$$

## 4 BP网络的优缺点

BP网络的**优点**为：

(1) 只要有足够多的隐层和隐层节点，**BP**网络可以逼近任意的非线性映射关系；

(2) **BP**网络的学习算法属于全局逼近算法，具有较强的泛化能力。

(3) **BP**网络输入输出之间的关联信息分布地存储在网络的连接权中，个别神经元的损坏只对输入输出关系有较小的影响，因而**BP**网络具有较好的容错性。

## BP网络的主要缺点为：

(1) 待寻优的参数多，收敛速度慢；

(2) 目标函数存在多个极值点，按梯度下降法进行学习，很容易陷入局部极小值；

(3) 难以确定隐层及隐层节点的数目。目前，如何根据特定的问题来确定具体的网络结构尚无很好的方法，仍需根据经验来试凑，但输入层数为影响输出的类别个数。

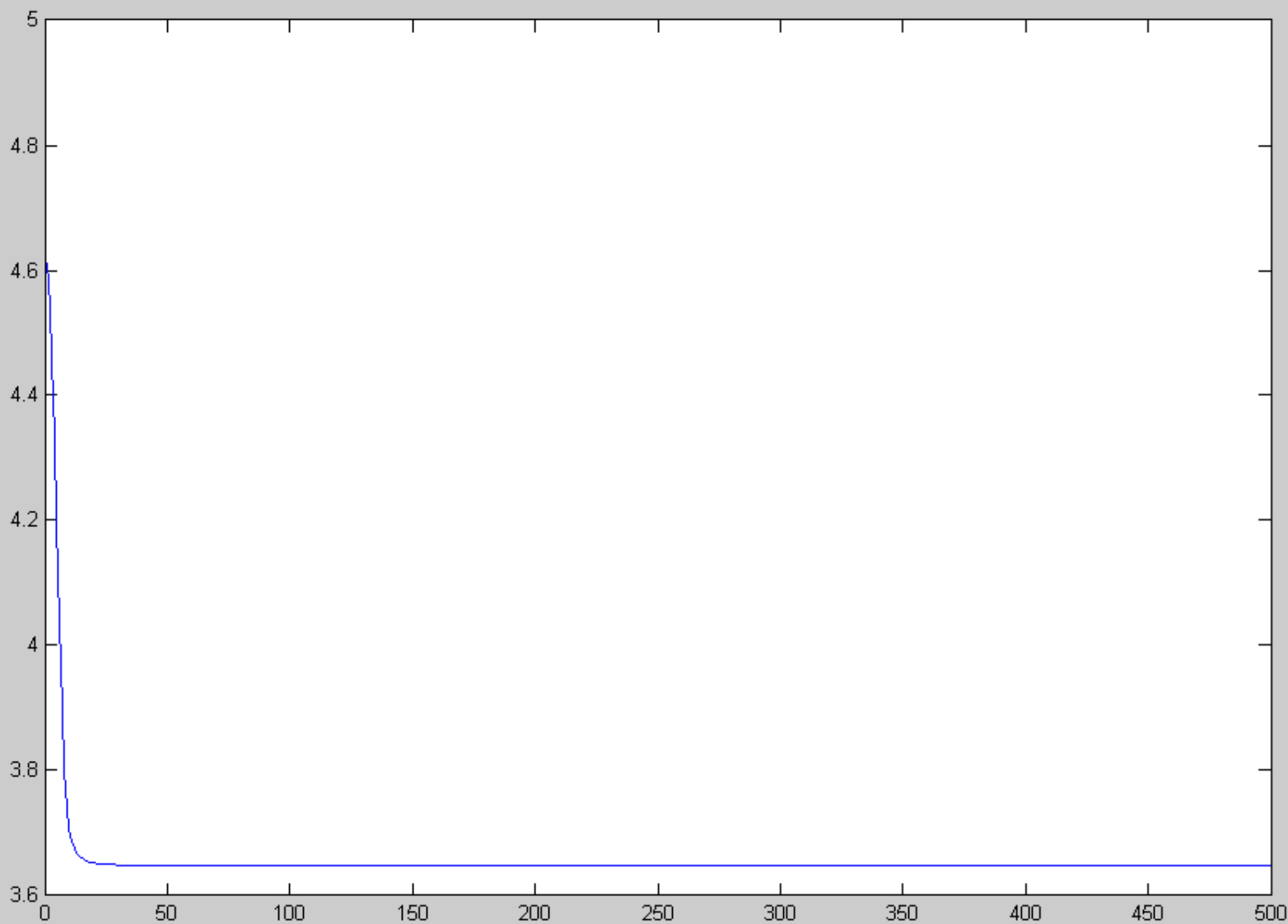
**[例] 用BP网络来实现下面的输入/输出关系:**

<b>输入 矢量P</b>	<b>-6</b>	<b>- 6.1</b>	<b>-4.1</b>	<b>-4</b>	<b>4</b>	<b>4.1</b>	<b>6</b>	<b>6.1</b>
<b>输出 矢量T</b>	<b>0</b>	<b>0</b>	<b>0.97</b>	<b>0.99</b>	<b>0.01</b>	<b>0.03</b>	<b>1</b>	<b>1</b>

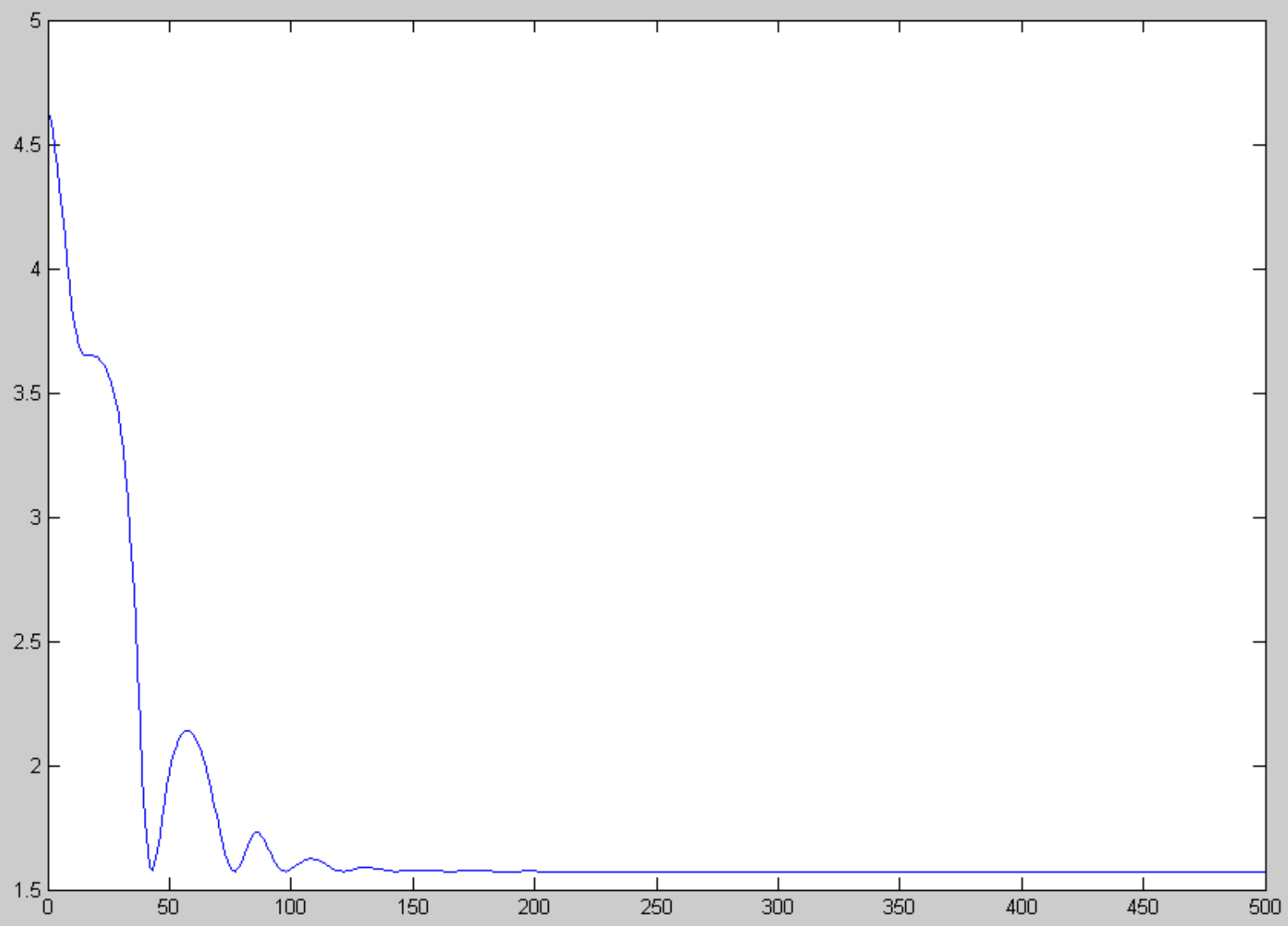




# 训练误差记录图(纯梯度下降法)



# 训练误差记录图(附加动量法)



- 由于**BP**网络具有很好的逼近非线性映射的能力，该网络在模式识别、图像处理、系统辨识、函数拟合、优化计算、最优预测和自适应控制等领域有着较为广泛的应用。
- 由于**BP**网络具有很好的逼近特性和泛化能力，可用于神经网络控制器的设计。但由于**BP**网络收敛速度慢，难以适应实时控制的要求。

## 5 BP网络的训练及其设计过程

为了训练一个BP网络，需要**计算网络加权输入矢量、输出和误差矢量**，然后**求出误差平方和**。

当所训练矢量的误差平方和小于误差目标，训练则停止，否则在输出层计算误差变化，且采用反向传播学习规则来调整权值，并重复此过程。

当网络完成训练后，对网络输入一个不是训练集合中的矢量，网络将以泛化方式给出输出结果。

**!** 在编写网络的设计程序之前，必须首先根据具体问题给出的**输入矢量与目标矢量**，并选定所要设计的**神经网络的结构**，其中包括以下内容：

- ① 网络层数；
- ② 每层的神经元数；
- ③ 每层的激活函数。

## MATLAB中BP网络的训练步骤（2层网络）：

**(1)** 用小的随机数对每一层的权值W和偏差B初始化，并进行以下参数的设定或初始化：

① 期望误差最小值mse；

② 最大循环次数；

③ 修正权值的学习速率lr，一般 $lr=0.01\sim 0.7$

**(2)** 用函数newcf.m建立一个多层前向网络，例如：

```
Net=newcf(minmax(P),[5  
1],{'tansig'purelin'},'traingd');
```

**(3)** 使用`net=train(net,P,T)`来训练网络。

## **[例] 用于函数逼近的BP网络的设计。**

有21组单输入矢量和相对应的目标矢量，试设计神经网络来实现这对数组的函数关系。

$$\mathbf{P} = -1:0.1:1;$$

$$\mathbf{T} = [-0.96 \ -0.577 \ -0.0729 \ 0.377 \ 0.641 \ 0.66 \ 0.461 \\ 0.1336 \ 0.201 \ -0.434 \ -0.5 \ -0.393 \ -0.1647 \ 0.0988 \\ 0.3072 \ 0.396 \ 0.3449 \ 0.1816 \ -0.0312 \ -0.2183 \ -0.3201];$$

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/156022231001010140>