

第 3 章 前馈神经网络盲均衡算法的研究

本章提要

本章在分析前馈神经网络基本原理的基础上,首先研究了基于三层、四层、五层前馈神经网络盲均衡算法,推导出算法的迭代公式,进行了计算机仿真,理论分析及实验验证了随着层数增加,算法收敛速度变慢、运算复杂度增加,但收敛后稳态剩余误差减小,即逼近能力增强。其次将改进 BP 算法应用于盲均衡算法中,研究了基于动量项盲均衡算法、时变动量项盲均衡算法和变步长盲均衡算法,同传统 BP 算法盲均衡算法相比,性能得以改善。

3.1 前馈神经网络的基本原理

3.1.1 前馈神经网络的概念

前馈神经网络是一种最简单的神经网络,各神经元分层排列,每个神经元只与前一层的神经元相连,接收前一层的输出,并输出给下一层,各层间没有反馈。是目前应用最广泛、发展最迅速的人工神经网络之一。研究从 20 世纪 60 年代开始,目前理论研究和实际应用达到了很高的水平。

3.1.2 前馈神经网络的结构

前馈神经网络采用一种单向多层结构,如图 3.1 所示。其中每一层包含若干个神经元,同一层的神经元之间没有互相连接,层间信息的传送只沿一个方向进行。其中第一层称为输入层,最后一层为输出层,中间为隐含层,简称隐层。隐层可以是一层,也可以是多层。

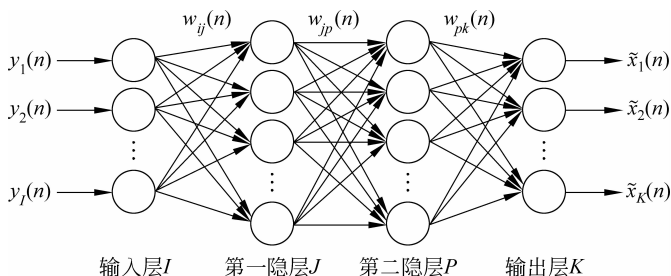


图 3.1 前馈神经网络结构

在前馈神经网络中, $\mathbf{Y}(n) = [y_1(n) \ y_2(n) \ \cdots \ y_I(n)]^T$ 为输入矢量, $w_{ij}(n)$ 、 $w_{jp}(n)$ 和 $w_{pk}(n)$ 为各层间的连接权值, $\tilde{\mathbf{X}}(n) = [\tilde{x}_1(n) \ \tilde{x}_2(n) \ \cdots \ \tilde{x}_K(n)]^T$ 为输出矢量。

3.1.3 前馈神经网络的特点

前馈神经网络结构简单,应用广泛,能够以任意精度逼近任意连续函数及平方可积函

数,而且可以精确实现任意有限训练样本集。从系统的观点看,前馈网络是一种静态非线性映射,通过简单非线性处理单元的复合映射,可获得复杂的非线性处理能力。从计算的观点看,缺乏丰富的动力学行为。大部分前馈网络都是学习网络,其分类能力和模式识别能力一般都强于反馈网络。

3.2 三层前馈神经网络盲均衡算法

3.2.1 三层前馈神经网络的模型

图 3.2 给出了三层前馈神经网络的模型,由输入层、输出层和一个隐层组成。设输入层与隐层间的连接权值为 $w_{ij}(n)$,其中 i 表示输入层的神经元($i=0,1,\cdots,m$), j 表示隐层的神经元($j=0,1,\cdots,k$);隐层与输出层的连接权值为 $w_j(n)$ 。假定输入层的输入为 $\mathbf{Y}(n)=[y(n) \ y(n-1) \ \cdots \ y(n-m)]^T$,隐层的输入为 $u_j(n)$,输出为 $I_j(n)$,输出层的输出为 $v(n)$,网络总输出为 $\tilde{x}(n)$ 。则前馈神经网络的状态方程可表示为

$$u_j(n) = \sum_{i=0}^m w_{ij}(n) y(n-i) \quad (3.1)$$

$$I_j(n) = f[u_j(n)] \quad (3.2)$$

$$v(n) = \sum_{j=0}^k w_j(n) I_j(n) \quad (3.3)$$

$$\tilde{x}(n) = f[v(n)] \quad (3.4)$$

式中, $f(\cdot)$ 表示隐层、输出层的输入和输出之间的传递函数,也称为非线性函数。

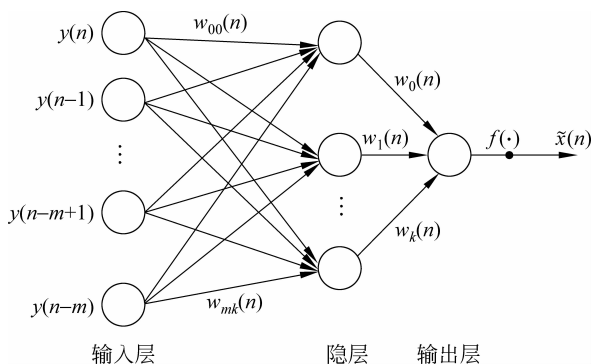


图 3.2 三层前馈神经网络结构

应用前馈神经网络进行盲均衡的关键在于确定神经元之间的连接权值和选择传递函数。连接权值通常是利用某些学习算法和系统的输入输出来进行训练得到。

3.2.2 实数条件下三层前馈神经网络盲均衡算法

1. 算法形式的推导

在传统的前馈神经网络研究中,传递函数一般选用阈值函数、Sigmoid 函数以及双曲正切函数等。本章选取双曲正切函数作为传递函数的基础来设计新的网络传递函数。因为该

函数具有平滑、渐进和单调的特性,有利于对输入序列进行判别。新的传递函数定义为

$$f(x) = x + \alpha \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.5)$$

式中, α 为比例因子,对于幅度间隔较大的信号, α 应取较大的值,幅度间隔较小的信号应取较小的值,这样有利于对输出信号进行更好地分类。

根据传统恒模盲均衡算法^[140]和前馈神经网络的训练方法,定义一种新的代价函数为

$$J(n) = \frac{1}{2} [\|\tilde{x}(n)\|^2 - R_2]^2 \quad (3.6)$$

式中, $\tilde{x}(n)$ 为前馈神经网络盲均衡器的输出序列,为实数信号; R_2 与传统恒模算法中的定义相同,即 $R_2 = \frac{E[\|x(n)\|^4]}{E[\|x(n)\|^2]}$ 。

根据最陡下降法,可以得到新的基于前馈神经网络盲均衡算法的权值迭代公式,为

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu \nabla \quad (3.7)$$

$$\nabla = \frac{\partial J(n)}{\partial \mathbf{W}(n)} = 2[\tilde{x}^2(n) - R_2]\tilde{x}(n) \frac{\partial \tilde{x}(n)}{\partial \mathbf{W}(n)} \quad (3.8)$$

由于三层前馈神经网络中含有隐层和输出层,故其权值迭代公式不同。

(1) 输出层权值迭代公式

对于网络的输出层,因它与隐层的连接权值为 $w_j(n)$,则有

$$\frac{\partial \tilde{x}(n)}{\partial w_j(n)} = f'[v(n)]I_j(n) \quad (3.9)$$

将式(3.9)和式(3.8)代入式(3.7),得到输出层权值迭代公式

$$w_j(n+1) = w_j(n) - \mu_1 k(n)I_j(n) \quad (3.10)$$

式中, $k(n) = 2[\tilde{x}^2(n) - R_2]\tilde{x}(n)f'[v(n)]$; μ_1 为输出层迭代步长因子。

(2) 隐层的权值迭代公式

对于网络的隐层,因它与输入层的连接权为 $w_{ij}(n)$,则

$$\frac{\partial \tilde{x}(n)}{\partial w_{ij}(n)} = f'[v(n)] \frac{\partial v(n)}{\partial w_{ij}(n)} = w_j(n)f'[v(n)]f'[u_j(n)]y(n-i) \quad (3.11)$$

将式(3.11)和式(3.8)代入式(3.7),得到隐层的权值迭代公式为

$$w_{ij}(n+1) = w_{ij}(n) - \mu_2 k_j(n)y(n-i) \quad (3.12)$$

式中, $k_j(n) = f'[u_j(n)]w_j(n)k(n)$; μ_2 为隐层迭代步长因子。

根据上述权值迭代公式,则可以用神经网络进行盲均衡。当网络的层数增加时(即含有两个或两个以上的隐层),隐层的迭代公式依然可用式(3.12)^[141]。

2. 计算机仿真

输入序列采用 8PAM 信号,信噪比为 20dB,滤波器阶数为 11,信道分别采用典型电话信道和普通信道。典型电话信道的传输函数见式(2.84),普通信道的传输函数^[142]为

$$H_2(z) = 1 + 0.5z^{-1} + 0.25z^{-2} + 0.125z^{-3} \quad (3.13)$$

图 3.3 和图 3.4 分别给出了两种信道中的收敛曲线,图中 NFNN 为本章研究的盲均衡算法, FNN 为 Cheolwoo You^[141]研究的盲均衡算法。仿真时,步长因子 μ_1 和 μ_2 取 0.001。

从收敛曲线可以看出, NFNN 算法在收敛速度上比 FNN 算法有了明显改进,但收敛后的稳态剩余误差要略大于 FNN 算法。

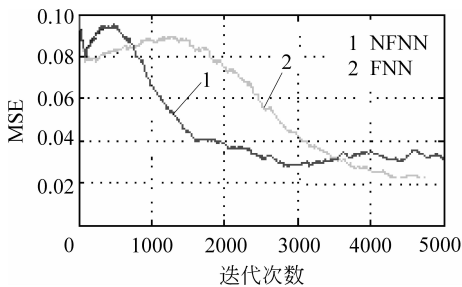


图 3.3 典型电话信道中的收敛曲线

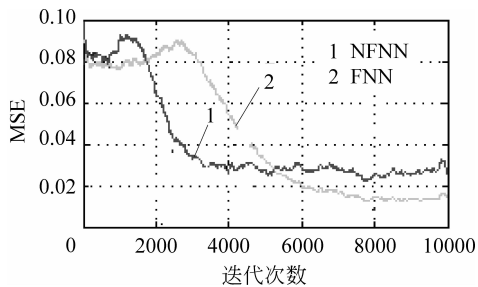


图 3.4 普通信道中的收敛曲线

3.2.3 复数条件下三层前馈神经网络盲均衡算法

1. 算法形式的推导

当将算法扩展到复值范围时,需要解决两方面的问题。一是必须设计一个合适的复值函数作为网络的传递函数。二是复数信号通过复数信道时,将不可避免地产生相位偏转。所以,算法不但要纠正信号的幅度失真,而且还要恢复信号的相位信息。

由于复数是由实部和虚部组成,所以考虑将网络结构进行适当变化。信号在进入非线性传递函数前分为两路,一路用于传输信号的实部,另一路传输虚部,通过非线性传递函数后再合成一路复数信号,其网络结构如图 3.5 所示^[141]。

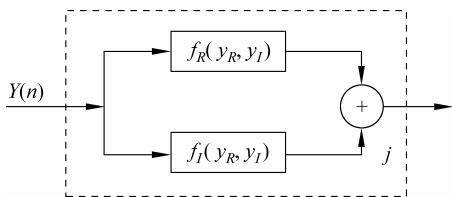


图 3.5 神经网络中复值处理单元模型

同实数情况一样,复数条件下的代价函数仍用式(3.6)的定义。

由于神经网络中非线性传递函数的存在,所以网络的权值形式应分为实部和虚部,即

$$\mathbf{W}(n) = \mathbf{W}_R(n) + j\mathbf{W}_I(n) \quad (3.14)$$

$$\nabla = \frac{\partial J(n)}{\partial \mathbf{W}(n)} = 2[|\tilde{x}(n)|^2 - R_2] |\tilde{x}(n)| \left[\frac{\partial |\tilde{x}(n)|}{\partial \mathbf{W}_R(n)} + j \frac{\partial |\tilde{x}(n)|}{\partial \mathbf{W}_I(n)} \right] \quad (3.15)$$

同时,为了便于公式推导,将网络信号也写为复数形式

$$y(n-i) = y_R(n-i) + jy_I(n-i) \quad (3.16)$$

$$w_{ij}(n) = w_{ij,R}(n) + jw_{ij,I}(n) \quad (3.17)$$

$$\begin{aligned} u_j(n) &= \sum_i w_{ij}(n) y(n-i) \\ &= \sum_i [w_{ij,R}(n) y_R(n-i) - w_{ij,I}(n) y_I(n-i)] \\ &\quad + j \sum_i [w_{ij,R}(n) y_I(n-i) + w_{ij,I}(n) y_R(n-i)] \end{aligned} \quad (3.18)$$

$$I_j(n) = f[u_{j,R}(n)] + jf[u_{j,I}(n)] \quad (3.19)$$

$$w_j(n) = w_{j,R}(n) + jw_{j,I}(n) \quad (3.20)$$

$$\begin{aligned} v(n) &= \sum_j w_j(n) I_j(n) = \sum_j [w_{j,R}(n) I_{j,R}(n) - w_{j,I}(n) I_{j,I}(n)] \\ &\quad + j \sum_j [w_{j,R}(n) I_{j,I}(n) + w_{j,I}(n) I_{j,R}(n)] \end{aligned} \quad (3.21)$$

$$\tilde{x}(n) = f[v_R(n)] + jf[v_I(n)] \quad (3.22)$$

与实数情况相类似,由于三层复值前馈神经网络有隐层和输出层,故其权值迭代公式不同。

(1) 输出层权值迭代公式

对于复值前馈神经网络的输出层,它与隐层的连接权值为 $w_j(n)$,则有

$$\begin{aligned} \frac{\partial |\tilde{x}(n)|}{\partial w_{j,R}(n)} &= \frac{1}{2 |\tilde{x}(n)|} \frac{\partial [\tilde{x}(n)\tilde{x}^*(n)]}{\partial w_{j,R}(n)} \\ &= \frac{1}{2 |\tilde{x}(n)|} \frac{\partial \{f^2[v_R(n)] + f^2[v_I(n)]\}}{\partial w_{j,R}(n)} \\ &= \frac{1}{|\tilde{x}(n)|} \{f[v_R(n)]f'[v_R(n)]I_{j,R}(n) + f[v_I(n)]f'[v_I(n)]I_{j,I}(n)\} \end{aligned} \quad (3.23)$$

$$\begin{aligned} \frac{\partial |\tilde{x}(n)|}{\partial w_{j,I}(n)} &= \frac{1}{2 |\tilde{x}(n)|} \frac{\partial [\tilde{x}(n)\tilde{x}^*(n)]}{\partial w_{j,I}(n)} \\ &= \frac{1}{2 |\tilde{x}(n)|} \frac{\partial \{f^2[v_R(n)] + f^2[v_I(n)]\}}{\partial w_{j,I}(n)} \\ &= \frac{1}{|\tilde{x}(n)|} \{-f[v_R(n)]f'[v_R(n)]I_{j,I}(n) + f[v_I(n)]f'[v_I(n)]I_{j,R}(n)\} \end{aligned} \quad (3.24)$$

由式(3.23)和式(3.24)可得

$$\begin{aligned} \frac{\partial |\tilde{x}(n)|}{\partial w_{j,R}(n)} + j \frac{\partial |\tilde{x}(n)|}{\partial w_{j,I}(n)} &= \frac{1}{|\tilde{x}(n)|} \{f[v_R(n)]f'[v_R(n)]I_{j,R}(n) + f[v_I(n)]f'[v_I(n)]I_{j,I}(n)\} \\ &\quad + j \frac{1}{|\tilde{x}(n)|} \{f[v_I(n)]f'[v_I(n)]I_{j,R}(n) - f[v_R(n)]f'[v_R(n)]I_{j,I}(n)\} \\ &= \frac{1}{|\tilde{x}(n)|} \{f[v_R(n)]f'[v_R(n)] + jf[v_I(n)]f'[v_I(n)]\} I_j^*(n) \end{aligned} \quad (3.25)$$

将式(3.25)代入式(3.14)和式(3.15),得到输出层权值迭代公式

$$w_j(n+1) = w_j(n) - 2\mu_1 k(n) I_j^*(n) \quad (3.26)$$

式中, $k(n) = [|\tilde{x}(n)|^2 - R_2] \{f[v_R(n)]f'[v_R(n)] + jf[v_I(n)]f'[v_I(n)]\}$; μ_1 为输出层迭代步长因子。

(2) 隐层权值迭代公式

对于复值前馈神经网络的隐层,因它与输入层的连接权值为 $w_{ij}(n)$,则可推导出其权值迭代公式(具体推导过程见附录 A)为

$$w_{ij}(n+1) = w_{ij}(n) - 2\mu_2 k_j(n) y^*(n-i) \quad (3.27)$$

式中, μ_2 为隐层迭代步长因子。

$$\begin{aligned} k_j(n) &= 2 [|\tilde{x}(n)|^2 - R_2] \{f[u_{j,R}(n)]\text{Re}\{[f[v_R(n)]f'[v_R(n)] + jf[v_I(n)]f'[v_I(n)]\}w_j^*(n)\} \\ &\quad + j2 [|\tilde{x}(n)|^2 - R_2] f'[u_{j,I}(n)]\text{Im}\{[f[v_R(n)]f'[v_R(n)] + jf[v_I(n)]f'[v_I(n)]\}w_j^*(n)\} \end{aligned} \quad (3.28)$$

根据上述公式,将前馈神经网络盲均衡算法扩展到了复数域内。与实数情况相类似,如

果神经网络的隐层层数增加,则隐层迭代公式依然可用。

2. 传递函数选择

将算法扩展到复数域时,网络的传递函数必须具有一系列特殊的特征。G. M. Georgiou 等^[143]分析了复值传递函数应具备的一般形式及条件。

复值传递函数应具备的通式为

$$F(x) = \xi(x_R, x_I) + j\zeta(x_R, x_I) \quad (3.29)$$

式中, $\xi(\cdot)$, $\zeta(\cdot)$ 为实值函数。

复值传递函数必须具备以下条件。

(1) 传递函数 $F(\cdot)$ 在复数域内对于信号的实部和虚部都是非线性函数。

(2) 为了保证系统的稳定性,在给定范围内, $F(\cdot)$ 应该是有界函数,并且保证在此范围内没有奇点(间断点)。另外,如果 $F(\cdot)$ 是非解析的,则 $F(\cdot)$ 在给定的范围内是连续的。

(3) $F(\cdot)$ 应该具有连续的一阶偏导数,即

$$\frac{\partial \xi(x_R, x_I)}{\partial x_R}, \quad \frac{\partial \xi(x_R, x_I)}{\partial x_I}, \quad \frac{\partial \zeta(x_R, x_I)}{\partial x_R}, \quad \frac{\partial \zeta(x_R, x_I)}{\partial x_I}$$

均存在。

(4) 对于复数信号,传递函数的实部与虚部应具有以下关系:

$$\frac{\partial u(x_R, x_I)}{\partial x_R} \frac{\partial v(x_R, x_I)}{\partial x_I} \neq \frac{\partial v(x_R, x_I)}{\partial x_R} \frac{\partial u(x_R, x_I)}{\partial x_I} \quad (3.30)$$

(5) 由于 QAM 调制方式决定了信号在复数平面对称的(即星座图的对称分布),所以输出信号 $\xi(x_R, x_I)$ 和 $\zeta(x_R, x_I)$ 应该有相同的动态范围。在一般通信系统中,没有畸变的复值信号关于实轴和虚轴是对称的,并且信号间的间隔是等距离的。

根据以上限定,在传递函数的选取上进行了两种选择。

(1) 在实数信道中,采用以双曲正切函数为基础的传递函数。当将该函数作为复值传递函数时,以上 5 个限定条件均满足。所以该函数可以用于复值前馈神经网络盲均衡算法中。但在计算机仿真过程中发现,如果传输信道为实数信道,应用该传递函数的算法性能要优于 Cheolwoo You^[141]提出的算法性能。这是因为在实数信道中传输复值信号时,信号的相位信息不会发生变化,所以其性能与实数系统相类似。但将该传递函数用于复值信道时,其纠正相位信息的能力要劣于 Cheolwoo You 提出的传递函数。因此,在复值信道中该传递函数不适用。

(2) 根据复值传递函数的五个限定条件,并针对复数系统中 QAM 信号在幅度和相位上的二维性,Cheolwoo You 曾提出一种传递函数,并成功应用于复值多层前馈神经网络盲均衡算法(FNN 算法)中,函数形式为

$$f(x) = f(\tilde{x}_R) + jf(\tilde{x}_I) \quad (3.31)$$

式中, $f(x) = x + \alpha \sin \pi x$, 为实值函数; $\alpha > 0$, 为比例系数。

通过具体仿真实验发现,FNN 算法中比例因子 α 的取值对算法的收敛性能有较大的影响。 α 取值较大,收敛速度较快,但收敛后的稳态剩余误差较大; α 取值较小,收敛后的稳态剩余误差较小,但收敛速度变慢。

基于 α 对算法收敛性能的影响,本章对传递函数的非线性部分进行了修正。在算法开

始阶段,应加快收敛速度,可以选择较大的 α 值,随着迭代过程的进行,应减小 α 值,使得收敛后的稳态剩余误差减小,以达到既加快收敛速度,又提高收敛精度的目的。

为了使 α 的值随收敛过程进行变化,采用均方误差来代替 α ,则修正后的传递函数为

$$f(x)=x+E[e^2(n)]\sin \pi x \tag{3.32}$$

3. 计算机仿真

(1) 传递函数采用 $f(x)=x+\alpha \frac{e^x-e^{-x}}{e^x+e^{-x}}$ 的计算机仿真

输入序列采用 16QAM 信号,信噪比为 20dB,信道分别采用典型电话信道和普通信道,其传输函数见式(2.84)和式(3.13)。

图 3.6 给出了 16QAM 信号在典型电话信道($\mu_1=\mu_2=0.02$)和普通信道($\mu_1=\mu_2=0.015$)中的收敛曲线。从图中可以看出,NFNN 算法在收敛速度上有了明显的改进,而且收敛后的稳态剩余误差与 FNN 算法基本相近。

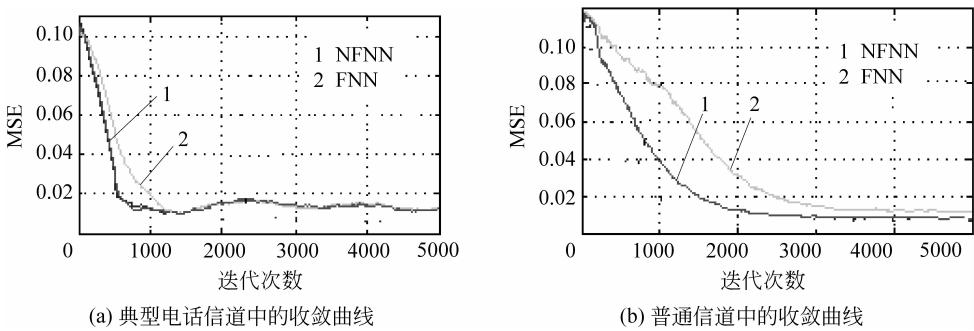


图 3.6 典型电话信道和普通信道中的收敛曲线

图 3.7 和图 3.8 给出了两种算法在上述信道中的误比特率比较曲线。从图中可以看出,在同等信噪比情况下,NFNN 算法的误比特率低于 FNN 算法,表明 NFNN 算法的误码特性优于 FNN 算法。

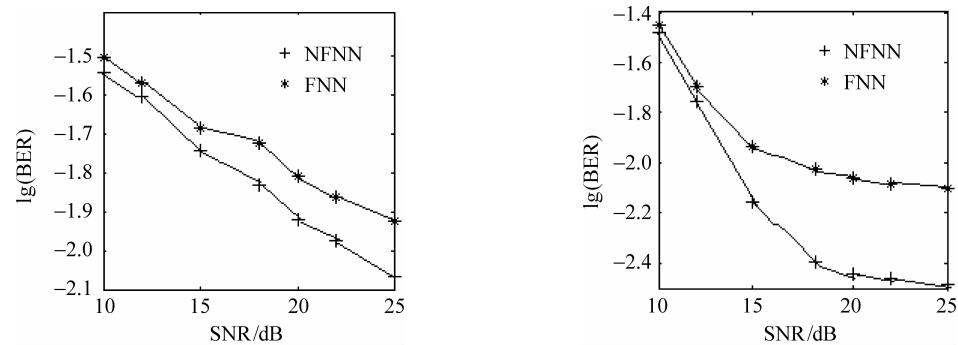


图 3.9 为两种算法在经过 5000 次迭代后的星座图。FNN 算法收敛后星座图在中心点附近有一定偏移,而 NFNN 算法收敛后未发生偏移现象,克服了 FNN 算法星座图不均匀的现象。

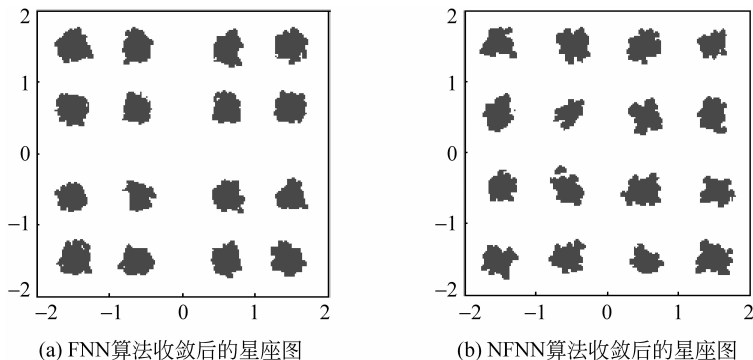


图 3.9 16QAM 信号在 FNN 和 NFNN 收敛后的星座图

(2) 传递函数采用 $f(v) = x + E[e^2(n)]\sin\pi x$ 的计算机仿真

输入序列采用 16QAM 信号, 信噪比为 20dB, 信道采用无线数字通信信道, 其传输函数^[141]为

$$\begin{aligned}
 H_3(z) = & (0.0410 + j0.0109)z^0 + (0.0495 + j0.0123)z^{-1} + (0.0672 + j0.0170)z^{-2} \\
 & + (0.0919 + j0.0235)z^{-3} + (0.7920 + j0.1281)z^{-4} + (0.3960 + j0.0871)z^{-5} \\
 & + (0.2715 + j0.0498)z^{-6} + (0.2291 + j0.0414)z^{-7} + (0.1287 + j0.0154)z^{-8} \\
 & + (0.1032 + j0.0119)z^{-9}
 \end{aligned} \quad (3.33)$$

图 3.10 给出了 16QAM 信号在无线数字通信信道中的收敛曲线, 迭代步长为 $\mu_1 = \mu_2 = 0.004$ 。由图可以看出, NFNN 算法和 FNN 算法相比, 收敛性能提高, 即收敛速度加快, 稳态剩余误差减小。这表明采用 NFNN 算法提高了收敛性能, 同时也验证了传递函数中参数选取的合理性。

图 3.11 给出了经过 5000 次迭代后的误比特率曲线图。在相同信噪比情况下, NFNN 的误比特率明显降低。

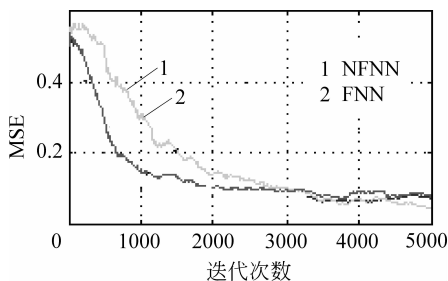


图 3.10 无线数字通信信道中的收敛曲线

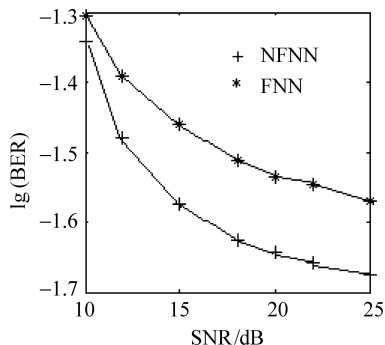
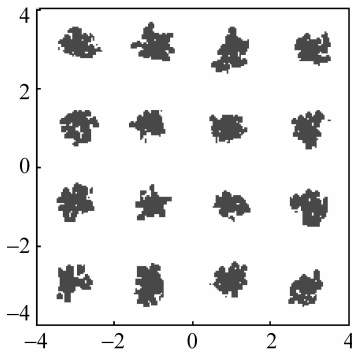
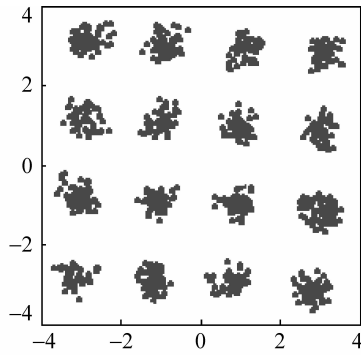


图 3.11 无线数字通信信道中的误比特率曲线

图 3.12 给出了 16QAM 信号经过 5000 个采样后两种算法的星座图。从图中可以看出, NFNN 算法的星座图在集中程度上优于 FNN 算法。



(a) NFNN算法收敛后的星座图



(b) FNN算法收敛后的星座图

图 3.12 NFNN 算法和 FNN 算法收敛后的星座图

3.3 多层前馈神经网络盲均衡算法

3.3.1 多层前馈神经网络的概念

多层前馈神经网络是指隐层层数多于一层的前馈神经网络。

1987 年,Robert Hecht Nielsen^[144]证明了对于任何在闭区间的一个连续函数都可以用一个隐层的 BP 神经网络来逼近。这说明用一个三层神经网络就可以完成任意 n 维到 m 维的映射。但是只含一个隐层的前馈神经网络是一个通用的函数逼近器,仅能说明有一个隐层已足够逼近,并不能说明一个隐层是最好的,有时采用两个隐层能得到更好的结果。文献 [106]也证明,BP 神经网络映射一个连续函数时,一个隐层是足够的,当要学习不连续函数时,则需要两个隐层。

增加隐层可以增加人工神经网络的处理能力,进一步减少误差,提高精度,但必然使得训练复杂化、训练样本数目增加和训练时间增长。

3.3.2 四层前馈神经网络盲均衡算法

1. 四层前馈神经网络的模型

四层前馈神经网络是指具有两个隐层的神经网络,如图 3.13 所示。图中,神经网络间只有前向输出,各层神经元之间用权值连接。输入层与第一隐层间的连接权值为 $w_{ij}(n)$ ($i=1,2,\dots,I; j=1,2,\dots,J$),第一隐层与第二隐层的连接权值为 $w_{jk}(n)$ ($k=1,2,\dots,K$),第二隐层与输出层之间的连接权值为 $w_k(n)$ 。神经元的输入用 u 表示,输出用 v 表示, u, v 的上标表示层,下标表示该层中的某个神经元,整个神经网络的输入为 $y(n-i)$,输出为 $\hat{x}(n)$ 。则四层前馈神经网络的状态方程为

$$u_i^I(n) = y(n-i) \quad (3.34)$$

$$v_i^I(n) = u_i^I(n) = y(n-i) \quad (3.35)$$

$$u_j^I(n) = \sum_{i=1}^I w_{ij}(n) v_i^I(n) = \sum_{i=1}^I w_{ij}(n) y(n-i) \quad (3.36)$$

$$v_j^I(n) = f_1(u_j^I(n)) = f_1\left(\sum_{i=1}^I w_{ij}(n)y(n-i)\right) \quad (3.37)$$

$$u_k^K(n) = \sum_{j=1}^J w_{jk}(n)v_j^I(n) \quad (3.38)$$

$$v_k^K(n) = f_2(u_k^K(n)) = f_2\left(\sum_{j=1}^J w_{jk}(n)v_j^I(n)\right) \quad (3.39)$$

$$u(n) = \sum_{k=1}^K w_k(n)v_k^K(n) \quad (3.40)$$

$$v(n) = \tilde{x}(n) = f_3(u(n)) = f_3\left(\sum_{k=1}^K w_k(n)v_k^K(n)\right) \quad (3.41)$$

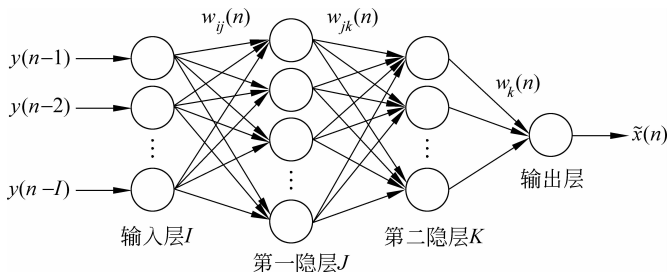


图 3.13 四层前馈神经网络结构

2. 算法形式的推导

选取传递函数为^[141]

$$f_1(x) = f_2(x) = f_3(x) = f(x) = x + \alpha \sin \pi x \quad \alpha > 0 \quad (3.42)$$

为了保证传递函数的单调性,要求它的导数 $f'(x)$ 恒大于 0, 所以应满足

$$f'(x) = 1 + \alpha \pi \cos \pi x > 0 \quad (3.43)$$

该传递函数的作用是对信号值在一定范围内进行修正,使其更接近于原信号,非线性函数 $\alpha \sin \pi x$ 为函数非线性修正项,它使得在原信号中心点附近左右摆动信号向原信号靠拢。 $f(x)$ 的输入输出特性曲线及其导数的特性如图 3.14 和图 3.15 所示。在原文中 α 的取值范围为 $0 < \alpha < 1/\pi \approx 0.318$ 。但在进行具体仿真时, α 的取值是在取值范围内取一固定的值,通过试验证明, α 的取值大小对算法效果有较大的影响。 α 的取值较大时,在一定程度上加快了 $f(x)$ 对输出信号的调整速度,算法的收敛速度较快,但收敛后的稳态剩余误差较大。反之,当 α 的取值较小时, $f(x)$ 对输出信号的调整速度减慢,算法收敛后的稳态剩余误差较小,但收敛速度变慢。

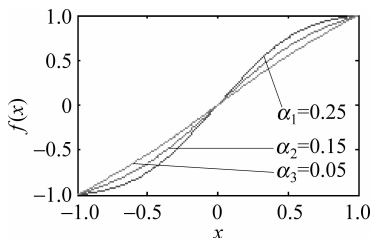


图 3.14 $f(x)$ 的输入输出特性曲线

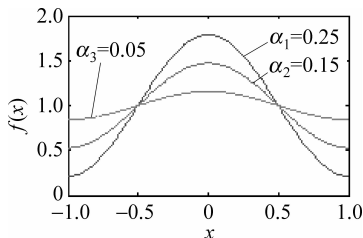


图 3.15 $f'(x)$ 的输入输出特性曲线

代价函数选择见式(3.6),网络权值的迭代公式一般采用最速梯度下降法形成,即

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu \frac{\partial J(n)}{\partial \mathbf{W}(n)} \quad (3.44)$$

$$\frac{\partial J(n)}{\partial \mathbf{W}(n)} = 2\tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial \mathbf{W}(n)} \quad (3.45)$$

式中, μ 是迭代步长。

由于四层前馈神经网络含有两个隐层和输出层,故其权值迭代公式不同。

(1) 输出层权值迭代公式

由于输出层与第二隐层的连接权值为 $w_k(n)$,其权值迭代公式为

$$w_k(n+1) = w_k(n) - 2\mu_1 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_k(n)} \quad (3.46)$$

$$\frac{\partial \tilde{x}(n)}{\partial w_k(n)} = f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) v_k^K(n) \quad (3.47)$$

$$w_k(n+1) = w_k(n) - 2\mu_1 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) v_k^K(n) \quad (3.48)$$

式中, μ_1 为输出层的迭代步长。

(2) 第一隐层与第二隐层的权值迭代公式

由于第一隐层与第二隐层的连接权值为 $w_{jk}(n)$,其权值迭代公式为

$$w_{jk}(n+1) = w_{jk}(n) - 2\mu_2 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_{jk}(n)} \quad (3.49)$$

$$\begin{aligned} \frac{\partial \tilde{x}(n)}{\partial w_{jk}(n)} &= f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \frac{\partial u(n)}{\partial w_{jk}(n)} = f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) \frac{\partial v_k^K(n)}{\partial w_{jk}(n)} \\ &= f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) f'_2 \left(\sum_{j=1}^J w_{jk}(n) v_j^I(n) \right) v_j^I(n) \end{aligned} \quad (3.50)$$

$$\begin{aligned} w_{jk}(n+1) &= w_{jk}(n) - 2\mu_2 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \\ &\quad \times f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) f'_2 \left(\sum_{j=1}^J w_{jk}(n) v_j^I(n) \right) w_k(n) v_j^I(n) \end{aligned} \quad (3.51)$$

式中, μ_2 为第一隐层与第二隐层间的迭代步长。

(3) 输入层与第一隐层的权值迭代公式

由于输入层与第一隐层的连接权值为 $w_{ij}(n)$,其权值迭代公式为

$$w_{ij}(n+1) = w_{ij}(n) - 2\mu_3 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_{ij}(n)} \quad (3.52)$$

$$\begin{aligned} \frac{\partial \tilde{x}(n)}{\partial w_{ij}(n)} &= f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) f'_2 \left(\sum_{j=1}^J w_{jk}(n) v_j^I(n) \right) \\ &\quad \times f'_1 \left(\sum_{i=1}^I w_{ij}(n) y(n-i) \right) w_k(n) w_{jk}(n) y(n-i) \end{aligned} \quad (3.53)$$

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) - 2\mu_3 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] f'_3 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \\ &\quad \times f'_2 \left(\sum_{j=1}^J w_{jk}(n) v_j^I(n) \right) f'_1 \left(\sum_{i=1}^I w_{ij}(n) y(n-i) \right) w_k(n) w_{jk}(n) y(n-i) \end{aligned} \quad (3.54)$$

式中, μ_3 为输入层与第一隐层间的迭代步长。

3. 计算机仿真

输入信号采用 4PAM 和 8PAM 两种信号, 信噪比为 20dB, 信道分别采用典型电话信道和普通信道, 其传输函数分别见式(2.84)和式(3.13)。迭加噪声采用均值为 0、方差为 1 的高斯白噪声。

典型电话信道中 4PAM 和 8PAM 的迭代步长分别为 $\mu_1 = \mu_2 = \mu_3 = 0.000\ 028$ 和 $\mu_1 = \mu_2 = \mu_3 = 0.000\ 001\ 5$ 。

普通信道中的 4PAM 和 8PAM 的迭代步长分别为 $\mu_1 = \mu_2 = \mu_3 = 0.000\ 018$ 和 $\mu_1 = \mu_2 = \mu_3 = 0.000\ 001$ 。

在 4PAM 信号仿真中采用 11-3-2-1 型前馈神经网络, 初始化权值设计为:

$$(1) \text{ 输入层到第一隐层为 } \mathbf{w}_{ij}(n) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T;$$

$$(2) \text{ 第一隐层到第二隐层为 } \mathbf{w}_{jk} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T;$$

$$(3) \text{ 第二隐层到输出层为 } \mathbf{w}_k = [1 \quad 0.1]^T。$$

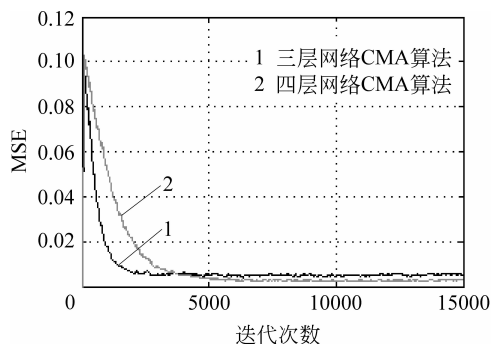
在 8PAM 信号仿真中采用 5-4-4-1 型前馈神经网络, 初始化权值设计为:

$$(1) \text{ 输入层到第一隐层为 } \mathbf{w}_{ij}(n) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T;$$

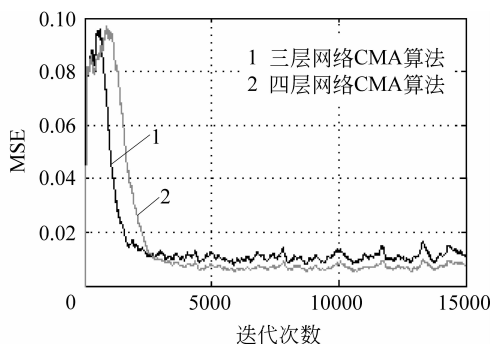
$$(2) \text{ 第一隐层到第二隐层为 } \mathbf{w}_{jk}(n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}^T;$$

$$(3) \text{ 第二隐层到输出层为 } \mathbf{w}_k(n) = [1.1 \quad -0.1 \quad -0.1 \quad -0.1]^T。$$

图 3.16~图 3.19 分别给出了 4PAM 和 8PAM 信号在典型电话信道和普通信道中的收敛曲线以及误码率曲线。其中误码率曲线是进行了 40 000 次迭代后, 不同信噪比对应的误码率。

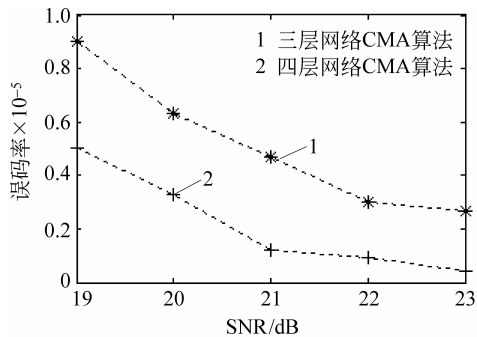


(a) 4PAM信号的收敛曲线

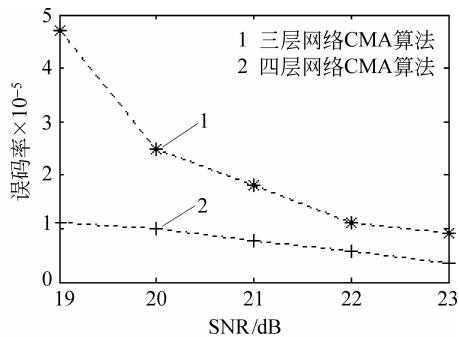


(b) 8PAM信号的收敛曲线

图 3.16 通过典型电话信道时 4PAM 与 8PAM 信号的收敛曲线

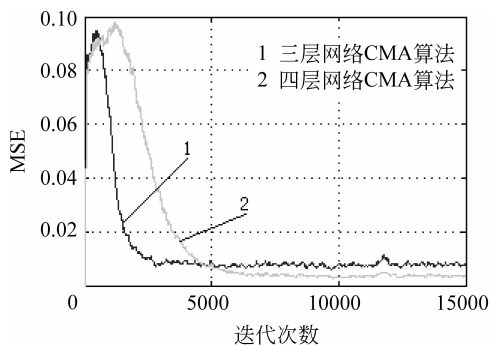


(a) 4PAM信号的误码率曲线

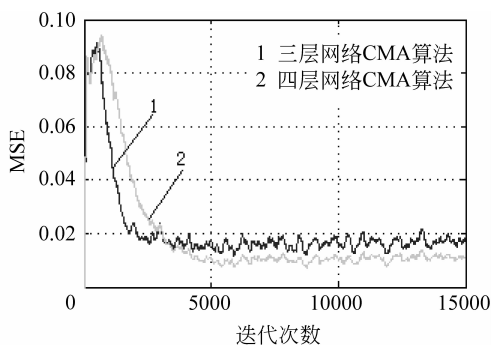


(b) 8PAM信号的误码率曲线

图 3.17 通过典型电话信道时 4PAM 与 8PAM 信号的误码率曲线

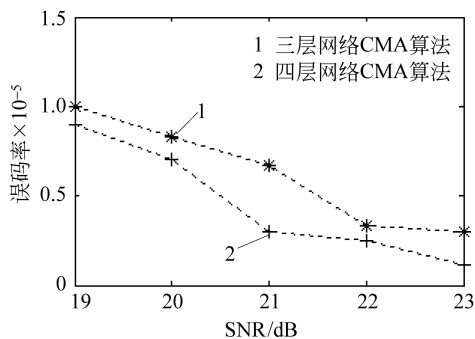


(a) 4PAM信号的收敛曲线

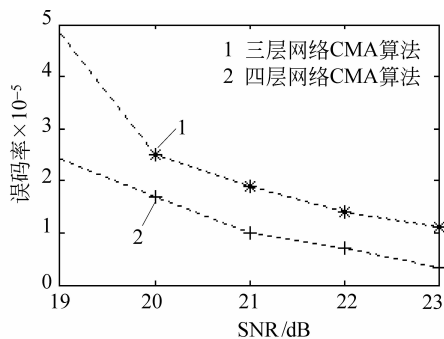


(b) 8PAM信号的收敛曲线

图 3.18 通过普通信道时 4PAM 与 8PAM 信号的收敛曲线



(a) 4PAM信号的误码率曲线



(b) 8PAM信号的误码率曲线

图 3.19 通过普通信道时 4PAM 与 8PAM 信号的误码率曲线

由图可以看出,四层前馈神经网络盲均衡算法与三层前馈神经网络盲均衡算法相比,稳态剩余误差和误码率有所减小,但收敛速度变慢。

3.3.3 五层前馈神经网络盲均衡算法

1. 五层前馈神经网络的模型

五层前馈神经网络是指含有 3 个隐层的神经网络,其结构如图 3.20 所示。

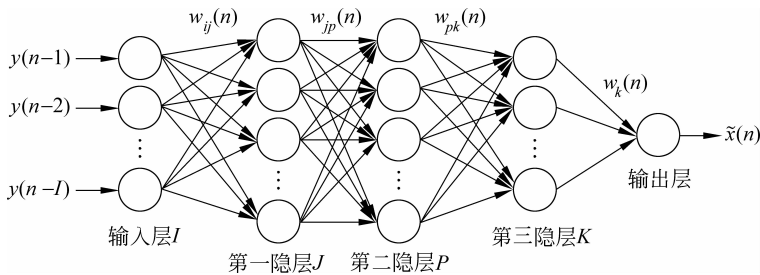


图 3.20 五层 BP 神经网络结构

图中,神经网络间只有前向输出,各层神经元之间用权值连接。输入层与第一隐层间的连接权值为 $w_{ij}(n)$ ($i=1,2,\dots,I; j=1,2,\dots,J$),第一隐层与第二隐层的连接权值为 $w_{jp}(n)$ ($p=1,2,\dots,P$),第二隐层与第三隐层的连接权值为 $w_{pk}(n)$ ($k=1,2,\dots,K$),第三隐层与输出层之间的连接权值为 $w_k(n)$ 。神经元的输入用 u 表示,输出用 v 表示, u, v 的上标表示层,下标表示该层中的某个神经元,整个神经网络的输入为 $y(n-i)$,输出为 $\tilde{x}(n)$ 。则前馈神经网络的状态方程为

$$u_i^I(n) = y(n-i) \quad (3.55)$$

$$v_i^I(n) = u_i^I(n) = y(n-i) \quad (3.56)$$

$$u_j^J(n) = \sum_{i=1}^I w_{ij}(n) v_i^I(n) = \sum_{i=1}^I w_{ij}(n) y(n-i) \quad (3.57)$$

$$v_j^J(n) = f_1(u_j^J(n)) = f_1\left(\sum_{i=1}^I w_{ij}(n) y(n-i)\right) \quad (3.58)$$

$$u_p^P(n) = \sum_{j=1}^J w_{jp}(n) v_j^J(n) \quad (3.59)$$

$$v_p^P(n) = f_2(u_p^P(n)) = f_2\left(\sum_{j=1}^J w_{jp}(n) v_j^J(n)\right) \quad (3.60)$$

$$u_k^K(n) = \sum_{p=1}^P w_{pk}(n) v_p^P(n) \quad (3.61)$$

$$v_k^K(n) = f_3(u_k^K(n)) = f_3\left(\sum_{p=1}^P w_{pk}(n) v_p^P(n)\right) \quad (3.62)$$

$$u(n) = \sum_{k=1}^K w_k(n) v_k^K(n) \quad (3.63)$$

$$v(n) = \tilde{x}(n) = f_4(u(n)) = f_4\left(\sum_{k=1}^K w_k(n) v_k^K(n)\right) \quad (3.64)$$

2. 算法形式的推导

选取的代价函数见式(3.6),传递函数为^[141]

$$f_1(x) = f_2(x) = f_3(x) = f_4(x) = f(x) = x + \alpha \sin \pi x \quad \alpha > 0 \quad (3.65)$$

在盲均衡算法中,网络权值的迭代公式一般采用最速梯度下降法形成,即

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu \frac{\partial J(n)}{\partial \mathbf{W}(n)} \quad (3.66)$$

$$\frac{\partial J(n)}{\partial \mathbf{W}(n)} = 2\tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial \mathbf{W}(n)} \quad (3.67)$$

式中, μ 是迭代步长。

由于五层前馈神经网络含有 3 个隐层和输出层, 故其权值迭代公式不同。

(1) 输出层权值迭代公式

由于输出层与第三隐层的连接权值为 $w_k(n)$, 其权值迭代公式为

$$w_k(n+1) = w_k(n) - 2\mu_1 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_k(n)} \quad (3.68)$$

$$\frac{\partial \tilde{x}(n)}{\partial w_k(n)} = f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) v_k^K(n) \quad (3.69)$$

$$w_k(n+1) = w_k(n) - 2\mu_1 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) v_k^K(n) \quad (3.70)$$

式中, μ_1 为输出层的迭代步长。

(2) 第二隐层与第三隐层的权值迭代公式

由于第二隐层与第三隐层的连接权值为 $w_{pk}(n)$, 其权值迭代公式为

$$w_{pk}(n+1) = w_{pk}(n) - 2\mu_2 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_{pk}(n)} \quad (3.71)$$

$$\begin{aligned} \frac{\partial \tilde{x}(n)}{\partial w_{pk}(n)} &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \frac{\partial u(n)}{\partial w_{pk}(n)} = f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) \frac{\partial v_k^K(n)}{\partial w_{pk}(n)} \\ &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) v_p^P(n) \end{aligned} \quad (3.72)$$

$$\begin{aligned} w_{pk}(n+1) &= w_{pk}(n) - 2\mu_2 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \\ &\quad \times f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) w_k(n) v_p^P(n) \end{aligned} \quad (3.73)$$

式中, μ_2 为第二隐层与第三隐层间的迭代步长。

(3) 第一隐层与第二隐层的权值迭代公式

由于第一隐层与第二隐层的连接权值为 $w_{jp}(n)$, 其权值迭代公式为

$$w_{jp}(n+1) = w_{jp}(n) - 2\mu_3 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_{jp}(n)} \quad (3.74)$$

$$\begin{aligned} \frac{\partial \tilde{x}(n)}{\partial w_{jp}(n)} &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \frac{\partial u(n)}{\partial w_{jp}(n)} = f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) \frac{\partial v_k^K(n)}{\partial w_{jp}(n)} \\ &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) w_{pk}(n) f'_2 \left(\sum_{j=1}^J w_{jp}(n) v_j^J(n) \right) v_j^J(n) \end{aligned} \quad (3.75)$$

$$\begin{aligned} w_{jp}(n+1) &= w_{jp}(n) - 2\mu_3 \tilde{x}(n)[|\tilde{x}(n)|^2 - R_2] f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \\ &\quad \times f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) f'_2 \left(\sum_{j=1}^J w_{jp}(n) v_j^J(n) \right) w_k(n) w_{pk}(n) v_j^J(n) \end{aligned} \quad (3.76)$$

式中, μ_3 为第一隐层与第二隐层间的迭代步长。

(4) 输入层与第一隐层的权值迭代公式

由于输入层与第一隐层的连接权值为 $w_{ij}(n)$, 其权值迭代公式为

$$w_{ij}(n+1) = w_{ij}(n) - 2\mu_4 \tilde{x}(n) [|\tilde{x}(n)|^2 - R_2] \frac{\partial \tilde{x}(n)}{\partial w_{ij}(n)} \quad (3.77)$$

$$\begin{aligned} \frac{\partial \tilde{x}(n)}{\partial w_{ij}(n)} &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \frac{\partial u(n)}{\partial w_{ij}(n)} = f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) \frac{\partial v_k^K(n)}{\partial w_{ij}(n)} \\ &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) w_{pk}(n) f'_2 \left(\sum_{j=1}^J w_{jp}(n) v_j^J(n) \right) \frac{\partial v_j^J(n)}{\partial w_{ij}(n)} \\ &= f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) w_k(n) f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) w_{pk}(n) f'_2 \left(\sum_{j=1}^J w_{jp}(n) v_j^J(n) \right) \\ &\quad \times w_{jp}(n) f'_1 \left(\sum_{i=1}^I w_{ij}(n) y(n-i) \right) y(n-i) \end{aligned} \quad (3.78)$$

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) - 2\mu_4 \tilde{x}(n) [|\tilde{x}(n)|^2 - R_2] f'_4 \left(\sum_{k=1}^K w_k(n) v_k^K(n) \right) \\ &\quad \times f'_3 \left(\sum_{p=1}^P w_{pk}(n) v_p^P(n) \right) f'_2 \left(\sum_{j=1}^J w_{jp}(n) v_j^J(n) \right) \\ &\quad \times f'_1 \left(\sum_{i=1}^I w_{ij}(n) y(n-i) \right) w_k(n) w_{pk}(n) w_{jp}(n) y(n-i) \end{aligned} \quad (3.79)$$

式中, μ_4 为输入层与第一隐层间的迭代步长。

3. 计算机仿真

输入信号采用 4PAM 和 8PAM 两种信号, 信噪比为 20dB, 信道分别采用典型电话信道和普通信道, 其传输函数分别见式(2.84)和式(3.13)。迭加噪声采用均值为零、方差为 1 的高斯白噪声。

典型电话信道中 4PAM 和 8PAM 的迭代步长分别为 $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.000\ 034$ 和 $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.000\ 023$ 。

普通信道中迭代步长分别为 $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.000\ 028$ 和 $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0.000\ 001\ 8$ 。

在 4PAM 信号仿真中采用 7-5-3-2-1 型前馈神经网络, 初始化权值设计为:

$$(1) \text{ 输入层到第一隐层为 } w_{ij}(n) = \begin{bmatrix} 0.1 & 0 & 10 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}^T;$$

$$(2) \text{ 第一隐层到第二隐层为 } w_{jp}(n) = \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 \\ 0.1 & 0.5 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T;$$

$$(3) \text{ 第二隐层到第三隐层为 } w_{pk}(n) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.1 \end{bmatrix}^T;$$

$$(4) \text{ 第三隐层到输出层为 } w_k(n) = [1 \ 0.1]^T.$$

在 8PAM 信号仿真中采用 5-4-4-2-1 型前馈神经网络,初始化权值设计为:

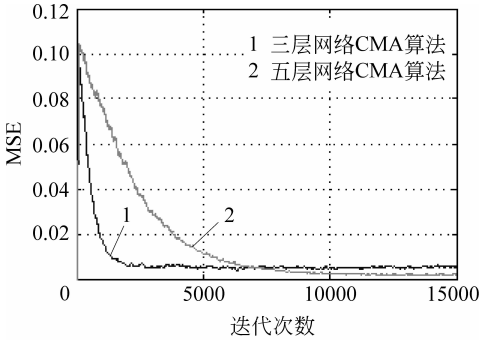
$$(1) \text{ 输入层到第一隐层为 } \mathbf{w}_{ij}(n) = \begin{bmatrix} 0.1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0.1 & 0 & 0 & 0.1 & 0 \\ 0 & 0.1 & 0 & 0.1 & 0 \end{bmatrix}^T;$$

$$(2) \text{ 第一隐层到第二隐层为 } \mathbf{w}_{jp}(n) = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}^T;$$

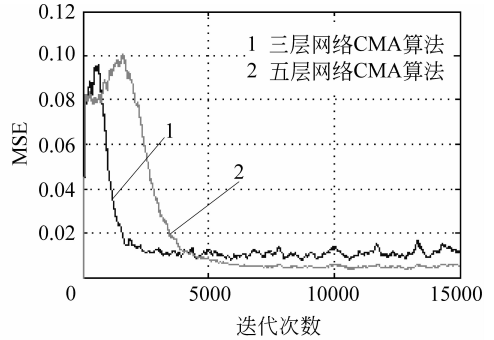
$$(3) \text{ 第二隐层到第三隐层为 } \mathbf{w}_{pk}(n) = \begin{bmatrix} 1 & 0.5 & 0 & 0.1 \\ 0.1 & 0 & 0.1 & 1 \end{bmatrix}^T;$$

$$(4) \text{ 第三隐层到输出层为 } \mathbf{w}_k(n) = [0.1 \quad 0.1]^T.$$

图 3.21~图 3.24 分别给出了 4PAM 和 8PAM 信号在两种信道中五层和三层前馈神经网络盲均衡算法的收敛曲线以及误码率曲线比较。其中,误码率曲线是进行了 40 000 次迭代后,不同信噪比对应的误码率。

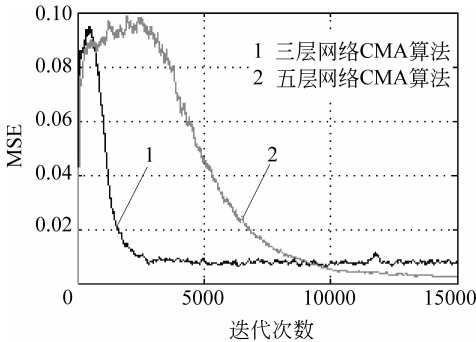


(a) 4PAM信号的收敛曲线

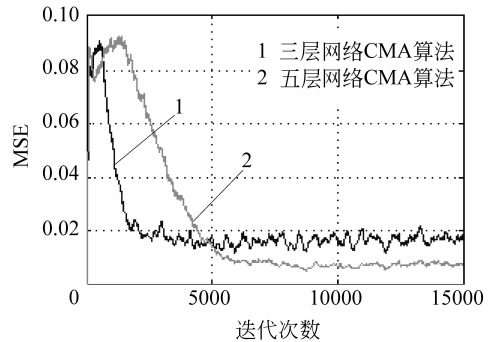


(b) 8PAM信号的收敛曲线

图 3.21 通过典型电话信道时 4PAM 与 8PAM 信号的收敛曲线



(a) 4PAM信号的收敛曲线



(b) 8PAM信号的收敛曲线

图 3.22 通过普通信道时 4PAM 与 8PAM 信号的收敛曲线

从曲线可以看出,在两种信道中五层前馈神经网络盲均衡算法的稳态剩余误差和误码

率均优于三层前馈神经网络盲均衡算法,但收敛速度减慢。

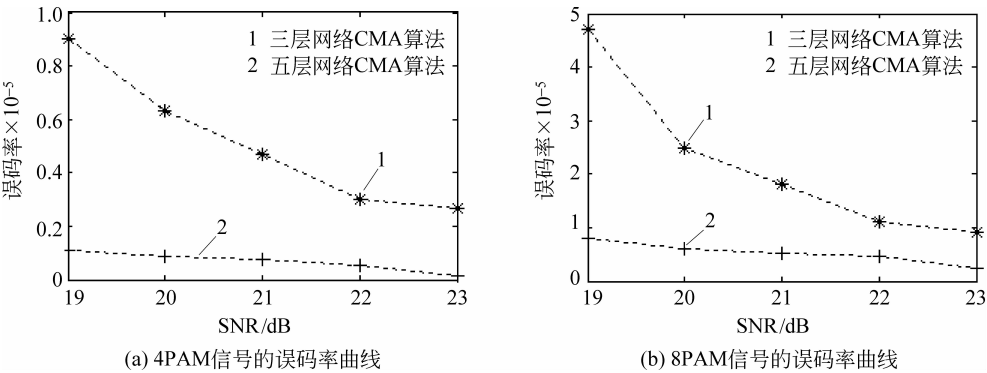


图 3.23 通过典型电话信道时 4PAM 与 8PAM 信号的误码率曲线

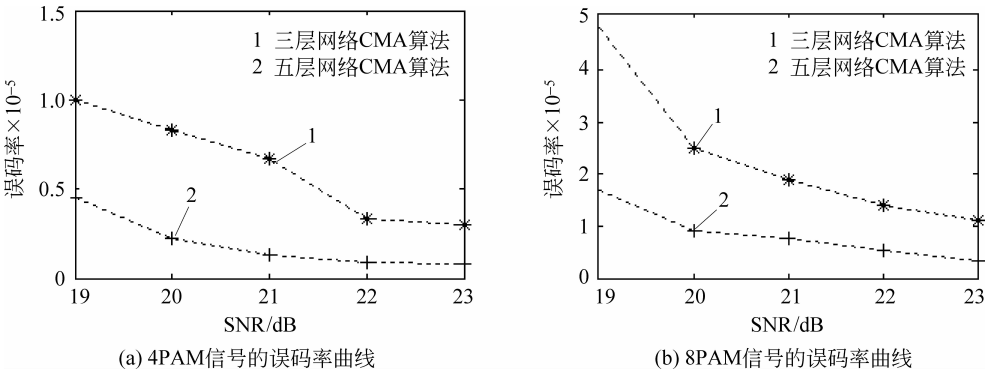


图 3.24 通过普通信道时 4PAM 与 8PAM 信号的误码率曲线

图 3.25~图 3.28 分别给出了 4PAM、8PAM 两种信号在两种信道中五层与四层前馈神经网络盲均衡算法的收敛曲线和误码率曲线。典型电话信道中 4PAM 和 8PAM 的迭代步长分别为 $\mu_1=\mu_2=\mu_3=\mu_4=0.000\ 025$ 和 $\mu_1=\mu_2=\mu_3=\mu_4=0.000\ 013$ 。普通信道中的迭代步长分别为 $\mu_1=\mu_2=\mu_3=\mu_4=0.000\ 018$ 和 $\mu_1=\mu_2=\mu_3=\mu_4=0.000\ 001$ 。误码率曲线是进行了 40 000 次迭代后,不同信噪比对应的误码率。

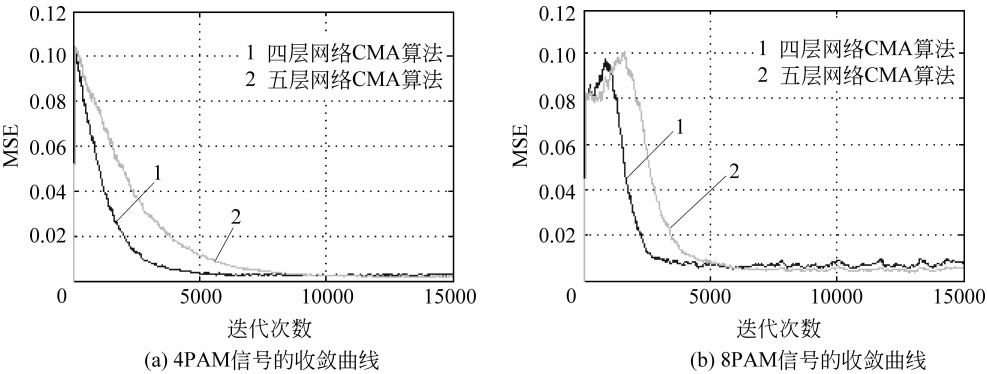
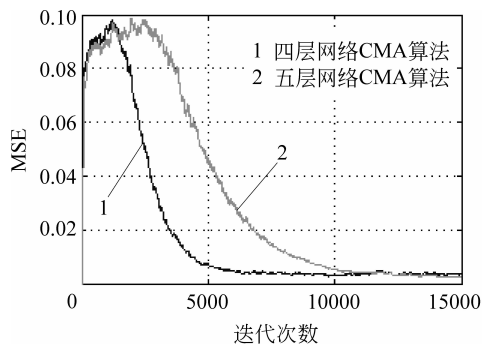
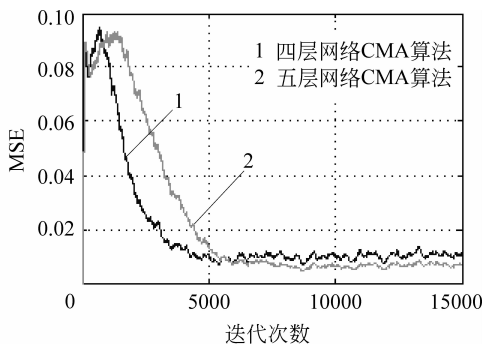


图 3.25 通过典型电话信道时 4PAM 与 8PAM 信号的收敛曲线

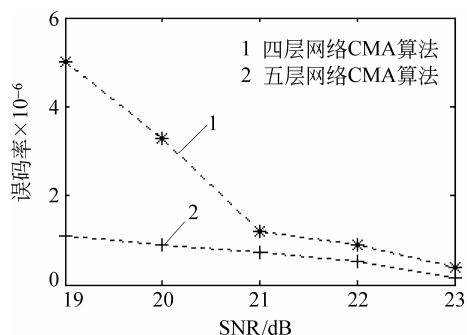


(a) 4PAM信号的收敛曲线

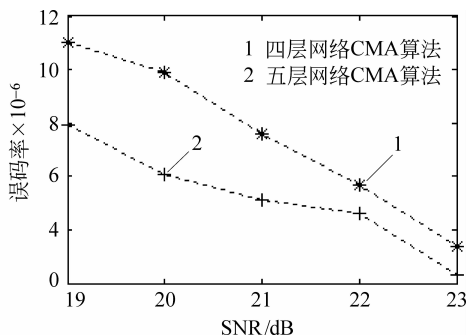


(b) 8PAM信号的收敛曲线

图 3.26 通过普通信道时 4PAM 与 8PAM 信号的收敛曲线

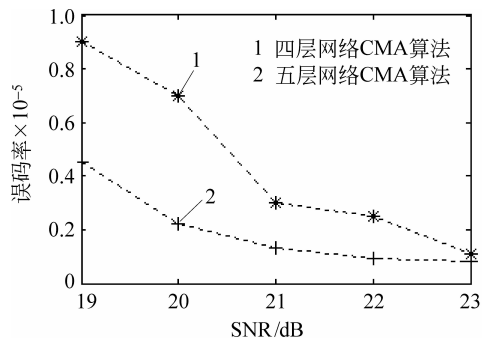


(a) 4PAM信号的误码率曲线

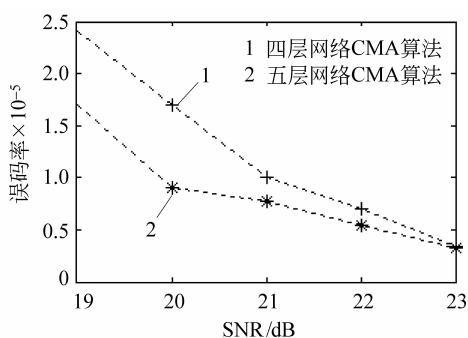


(b) 8PAM信号的误码率曲线

图 3.27 通过典型电话信道时 4PAM 与 8PAM 信号的误码率曲线



(a) 4PAM信号的误码率曲线



(b) 8PAM信号的误码率曲线

图 3.28 通过普通信道时 4PAM 与 8PAM 信号的误码率曲线

从图可以看出,五层前馈神经网络盲均衡算法的稳态剩余误差和误码率均优于四层前馈神经网络盲均衡算法,但收敛速度减慢。

通过本节的分析与仿真可以看出,随着神经网络隐层数的增加,收敛速度和计算复杂度加大,但收敛后的稳态剩余误差和误码率减小。

3.4 动量项前馈神经网络盲均衡算法

3.4.1 算法的基本原理

根据第 2.4.2 节的分析,为了加快收敛速度,可以引入动量项,其权值变化量为

$$\Delta \mathbf{W}(n) = \gamma \Delta \mathbf{W}(n-1) - \mu \frac{\partial J(n)}{\partial \mathbf{W}(n)} \quad (3.80)$$

式中, $0 \leq \gamma < 1$ 为加入的动量项, $\gamma \Delta \mathbf{W}(n-1)$ 记忆了前一时刻权值的修改方向,在时刻 $(n+1)$ 权值的修改方向为时刻 $(n-1)$ 与时刻 n 时的组合。当本次的 $-\mu \frac{\partial J(n)}{\partial \mathbf{W}(n)}$ 与前一次同号时,两次和值增大,使 $\Delta \mathbf{W}(n)$ 增大,从而在稳定调节时增大了调节速度,即加快了收敛;当 $-\mu \frac{\partial J(n)}{\partial \mathbf{W}(n)}$ 与前一次符号相反时,说明有一定的振荡,此时 $\Delta \mathbf{W}(n)$ 减小,起到了稳定作用。当 $\gamma=0$ 时算法上式变为传统的 BP 算法。

图 3.29~图 3.31 给出了传统 BP 算法、动量项 BP 算法以及变学习速率动量项 BP 算法在相同性能曲面上的搜索路径。网络结构采用 1-2-1 型前馈神经网络,初始权值取相同值。隐层到输出层的权值为 $\mathbf{w}=[1,1]^T$,输入层到隐层的权值为 $\mathbf{w}=[1,1]^T$ 。

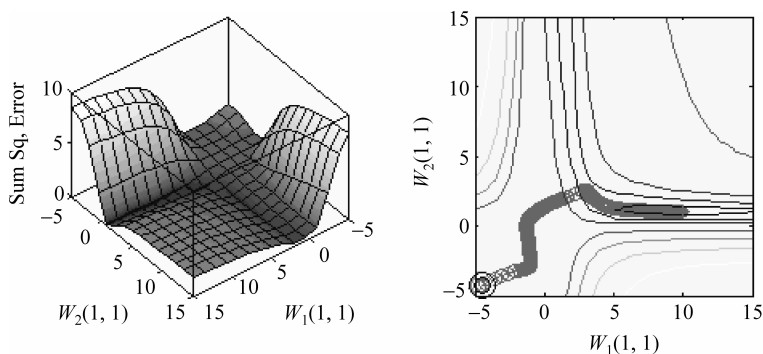


图 3.29 传统 BP 算法在相同性能曲面的收敛轨迹

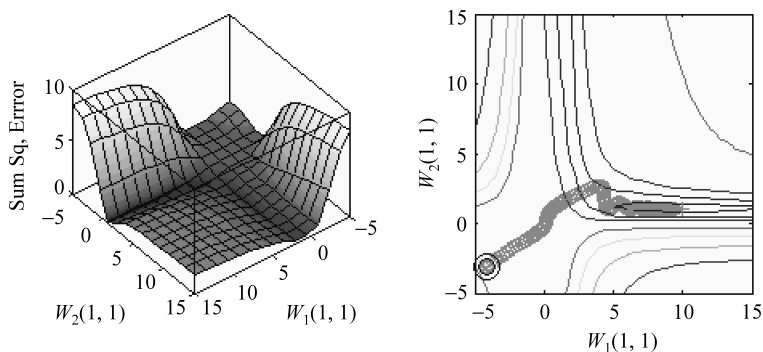


图 3.30 动量项 BP 算法在相同性能曲面的收敛轨迹