

第5章 基于多小波和回声状态网络的非线性信道盲均衡算法

导 读

本章首先分析了行波管放大器(TWTA)的非线性对调制信号的影响,在此基础上,利用多小波基函数对均衡器输入信号和判决装置输出信号进行变换,提出了基于多小波双变换的非线性信道盲均衡算法;将多小波基函数作为神经网络神经元激励函数,研究了基于多小波神经网络的非线性盲均衡算法;将支持向量机、分集技术和神经网络相结合,研究了基于空间分集支持向量机多小波神经网络的非线性盲均衡算法;利用混沌搜索算法,优化了基于双神经网络的非线性盲均衡算法;基于回声状态网络动态库内具有反馈回路、训练过程简单、处理能力强等特点,采用 Volterra 滤波器作为回声状态网络输出层并结合主成分分析法(PCA),分析了基于 Volterra 滤波回声状态网络和 PCA 的均衡算法。

第5章

基于多小波和回声状态网络的非线性信道盲均衡算法

在卫星通信系统中,卫星信道的非线性严重影响卫星通信系统性能,会引起带内失真和带外频谱再生,且不能被忽略^[1,2]。这种非线性主要包括卫星通信系统射频端和卫星转发器上行波管放大器(Traveling Wave Tube Amplifier, TWTA)的非线性及信道畸变衰落引起群时延失真的非线性^[3,4],通常可采用发射端预失真或接收端均衡或高阶多模调制方式来减小其影响^[5]。然而,发射端预失真处理虽然有高效的频谱和功率利用率,但会产生畸变,严重影响通信质量^[6];接收端均衡虽然也能有效克服非线性影响,但均衡器权向量长度影响计算复杂度,例如,Volterra 均衡器可以有效地对非线性卫星信道进行补偿,但权系数数量相当大,计算复杂度相当高,收敛速度相当缓慢^[7-8]。文献[9]在没有考虑信道记忆性的情况下,利用判决反馈均衡器对非线性信道进行了补偿,但效果不理想;文献[10]给出的 Hammerstein-Wiener 模型,减少了权向量、降低了计算复杂度并缩短了收敛时间,但当输入信号相关矩阵的最大特征值与最小特征值比值(也称为条件数)较大时,收敛速度下降,影响了跟踪性能。研究表明,均衡器输入信号做多小波变换后,其自相关矩阵具有更加规整的稀疏性,不仅大大降低了计算复杂度,而且有效提高了收敛性能^[11]。

近年来,人工神经网络模型与模糊系统、遗传算法、进化机制等结合形成的智能运算逐渐成为人工智能的一个重要领域。其中,神经网络在非线性信道盲均衡技术中得到了一定应用,取得了一些成果^[12,13]。支持向量机(Support Vector Machine, SVM)通过寻求结构风险的最小化来实现实际风险的最小化,并在有限信息条件下追求最优结果,故支持向量机均衡技术研究也取得了一些成果^[14,15]。混沌是指现实世界中一种貌似无规则的复杂运动形态,是非线性动力学系统在一定条件下表现出的一种运动状态,利用混沌运动进行混沌搜索能有效避免局部极小值。混沌与神经网络

相结合得到的混沌神经网络在非线性系统全局优化和噪声控制中的应用研究已成为新热点^[16-18]。

针对信号处理领域中需解决的实际问题具有动态性、实时性、非线性等特点,所建模型需能执行非线性映射,也需能访问输入和输出信号的时间历程以及内部状态,具有记忆性。递归神经网络(Recurrent Neural Networks,RNN)允许不同层神经元之间有反馈连接且有记忆性,是较为理想的模型。然而,在训练阶段,反馈循环会使代价函数对连接权的导数难以计算,而且网络的不稳定也可能导致各种固有的网络功能失效。在这种情况下,文献[19]最初提出的回声状态网络(Echo State Networks,ESN)保留了RNN的处理潜力,在训练阶段,通过设置递归内层的连接权重(也就是建立所谓的动态库)实现简化处理,无须使用期望信号;通过最小二乘法对输出线性组合器权向量进行调整并将回声状态网络应用于信道均衡问题,在提高处理能力与降低计算复杂度之间实现了较好的折中^[20]。

本章针对卫星信道的非线性对通信质量的影响,主要研究:

- (1) 基于多小波双变换的非线性卫星信道盲均衡算法;
- (2) 基于多小波神经网络的非线性盲均衡算法;
- (3) 基于支持向量机和神经网络的非线性盲均衡算法;
- (4) 基于混沌算法优化的双神经网络盲均衡算法;
- (5) 基于 Volterra 滤波回声状态网络和主成分分析法的均衡算法。

5.1 非线性卫星信道 Volterra 盲均衡系统

非线性卫星通信 Volterra 盲均衡系统,如图 5.1 所示。图中非线性卫星信道模型由发送滤波器、行波管放大器(TWTA)及接收滤波器构成。其中,TWTA 有非线性,均衡器为 Volterra 均衡器。

图中, $a(n)$ 表示输入信号, $b(n)$ 表示发送滤波器的输出信号, $w(n)$ 表示噪声, $x(n)$ 是加入噪声后的信号, $y(n)$ 为接收滤波器的输出信号, $\mathbf{h}_1(n)$ 、 $\mathbf{h}_2(n)$ 与 $\mathbf{h}_3(n)$ 分别是发送滤波器、TWTA、接收滤波器的权向量, $f(n)$ 是 Volterra 均衡器权向量, $z(n)$ 表示均衡器输出信号, $e(n)$ 是由误差生成函数生成的误差信号, $\hat{a}(n)$ 表示判决输出信号。

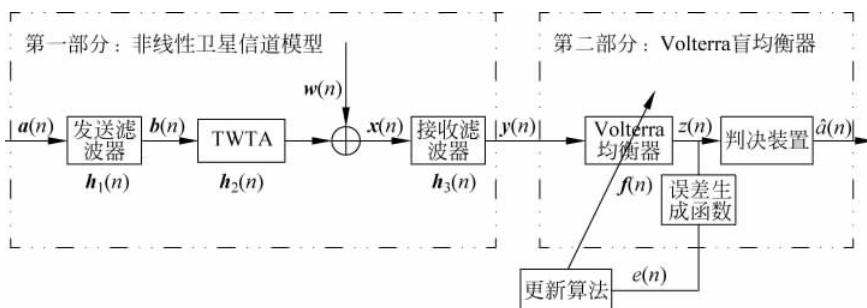


图 5.1 非线性卫星通信 Volterra 盲均衡系统

5.1.1 TWTA 的非线性对调制信号的影响

图 5.1 所示的发送滤波器输出信号 $b(n)$ 通过 TWTA 时, 会产生幅度转换效应(AM/AM)和幅相转换效应(AM/PM), 严重影响通信质量。根据 TWTA 的 Saleh 模型^[21,22], AM/AM、AM/PM 的输入输出函数为

$$A(r) = \alpha_a r / (1 + \beta_a r^2) \quad (5.1.1)$$

$$\Phi(r) = \alpha_p r^2 / (1 + \beta_p r^2) \quad (5.1.2)$$

式中, $\alpha_a = 2$, $\beta_a = 1$, $\alpha_p = \pi/3$, $\beta_p = 1$ 。TWTA 模型函数特性曲线, 如图 5.2 所示。

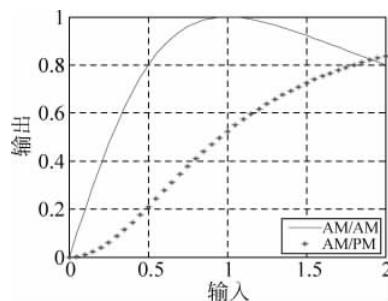


图 5.2 AM/AM、AM/PM 函数特性曲线

图 5.2 表明, 信号通过 TWTA 后, 幅度和相位分别产生了很大的畸变失真。在卫星通信中, 正交相移键控(Quadrature Phase Shift Keying, QPSK)信号在带宽和功率权衡点方面的性能比二进制相移键控(Binary Phase Shift Keying, BPSK)和 16QAM 优越。尽管 BPSK 与 QPSK 具有相同的误比特率, 但后者的能量效率(每比特所消耗的能量)仅为前者的一

半；16QAM 虽然提高了频谱利用率,但误比特率相同时需要 TWTA 有更高的发射功率^[23]。在卫星通信中,频率资源相对丰富,传输带宽较宽,但通信距离远,传输损耗大,功率资源有限,因此卫星通信一般采用功率利用率比较高的调制方式。目前,大多数通信卫星和电视转播卫星都采用 QPSK 调制传输业务数据。而 APSK 信号作为一种正交振幅调制与相移键控(Phase Shift Keying, PSK)调制相结合的方法,具有比较少的信号模值,信号包络的起伏也不是很大,既减小了对 TWTA 非线性的敏感性,又提高了频谱利用率,在卫星通信中是一种比较合适的调制方式。为了分析行波管放大器(TWTA)对 APSK 信号的影响,以 16QAM 和 16PSK 信号为比较对象,以 16APSK 作为 Saleh 行波管放大器(TWTA)的输入信号来分析 TWTA 对 APSK 的影响,如图 5.3 所示。

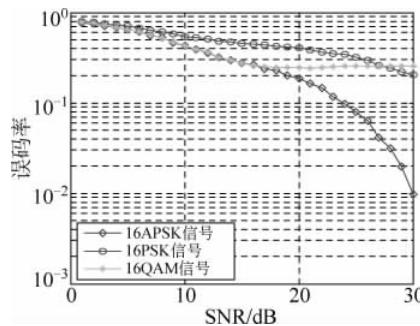


图 5.3 放大器对三种调制信号的影响

图 5.3 表明,16APSK 通过 TWTA 后,误码率最小;也就是说,TWTA 的非线性对 16APSK 失真影响最小。

5.1.2 基于非线性滤波器的盲均衡算法

现介绍两种基本的非线性滤波器:判决反馈滤波器与 Volterra 滤波器。

1. 判决反馈滤波器

具有判决反馈滤波器的均衡系统,如图 5.4 所示。

判决反馈均衡器(Decision Feedback Equalization, DFE)结构中包含一个反馈环,属于非线性结构^[24]。它的前馈滤波器与线性均衡器一样,以信道的输出信号 $y(n)$ 作为输入信号。而反馈滤波器把均衡器的判决信号

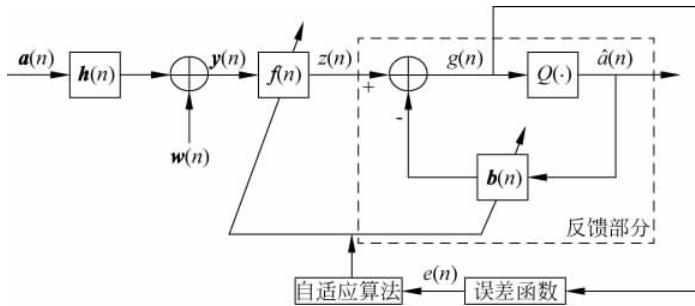


图 5.4 判决反馈均衡系统

$\hat{a}(n)$ 作为输入信号, 其输出信号被用来抵消来自前馈滤波器的部分失真。

假设均衡器的前馈滤波器部分权向量长度为 N_f , 反馈部分为 N_b 个, 前馈部分和反馈部分权向量分别为 $f(k)$ 和 $b(k)$, 则判决反馈均衡器的判决输入为

$$\begin{aligned} g(n) &= \sum_{i=0}^{N_f-1} f^*(n, i) y(n-i) - \sum_{i=0}^{N_b-1} b^*(n, i) \hat{a}(n-i) \\ &= f^H(n) y(n) - b^H(n) \hat{a}(n) \end{aligned} \quad (5.1.3)$$

式中, 在 DFE 的训练模式下, $\hat{a}(n) = a(n)$; 在数据模式下, $\hat{a}(n) = \text{dec}(a(n))$, 其中 dec 表示判决。

2. Volterra 滤波器^[25]

Volterra 级数是一种泛函级数, 作为泰勒级数的有记忆扩展, 被广泛用于非线性系统分析, 有很强的系统表征能力。非线性时不变系统输入/输出函数为

$$\begin{aligned} z(n) &= h_0 + \int_{-\infty}^{\infty} h_1(\tau_1) y(n-\tau_1) d\tau_1 + \\ &\quad \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) y(n-\tau_1) y(n-\tau_2) d\tau_1 d\tau_2 + \\ &\quad \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_k(\tau_1, \tau_2, \dots, \tau_k) y(n-\tau_1) y(n-\tau_2) \cdots \\ &\quad y(n-\tau_k) d\tau_1 d\tau_2 \cdots d\tau_k \end{aligned} \quad (5.1.4)$$

式中, $y(n)$ 表示系统输入, $z(n)$ 表示系统输出。 $(h_0, h_1(\tau_1), h_2(\tau_1, \tau_2), \dots)$ 被称为系统的核, h_0 表示系统的直流响应, h_1 表示系统的线性响应, 二阶及二阶以上的高阶核表示系统的非线性响应。

基于横向滤波器结构的均衡器本质上是有限的脉冲响应滤波器, 而

Volterra 级数具有无数多项,需根据系统实际需求对 Volterra 级数进行截取。根据式(5.1.4), N 阶 Volterra 非线性均衡器输出为

$$\begin{aligned} z(n) = & \sum_{m_1=0}^{N-1} f(m_1) y(n-m_1) + \\ & \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \sum_{m_3=0}^{N-1} f(m_1, m_2, m_3) y(n-m_1) y(n-m_2) y^*(n-m_3) \end{aligned} \quad (5.1.5)$$

式中, $f(m_1)$ 与 $f(m_1, m_2, m_3)$ 为均衡器的权系数, N 也称为该滤波器记忆长度。在卫星通信中,非线性信道是带通滤波器,由于偶次项产生的频率分量远离于载频而被带通滤波器滤除,因此只需考虑奇次项。式(5.1.5)中第一项为线性项,用来实现卫星信道的线性均衡;第三项为非线性项,用来实现卫星信道的非线性均衡。根据式(5.1.5), N 阶 Volterra 非线性均衡器权向量长度为

$$L = \sum_{n=1}^{(N+1)/2} N^{2n-1} \quad (5.1.6)$$

为了方便表示,将均衡器输入信号或接收滤波器的输出信号 $y(n)$ 及 Volterra 均衡器权系数表示为矩阵形式。令 Volterra 均衡器权向量为

$$\mathbf{f}(n) = [f(n), f(n-1), \dots, f(n-L+1)]^T \quad (5.1.7)$$

均衡器输入为

$$\begin{aligned} \mathbf{y}(n) = & [y(n), y(n-1), \dots, y(n-L+1), y^2(n) y^*(n), \dots, \\ & y^2(n-L+1) y^*(n-L+1)]^T \end{aligned} \quad (5.1.8)$$

这时,均衡器输出为

$$z(n) = \mathbf{f}^T(n) \mathbf{y}(n) \quad (5.1.9)$$

5.1.3 Volterra 盲均衡算法

图 5.1 中,利用常数模盲均衡算法(Constant Modulus blind equalization Algorithm, CMA)来更新 Volterra 权向量。Volterra 均衡器权向量、均衡器输入及均衡器输出分别为式(5.1.7)~式(5.1.9)。

CMA 的误差函数为

$$e(n) = R^2 - |z(n)|^2 \quad (5.1.10)$$

式中, R^2 为 CMA 的模值,且

$$R^2 = \frac{E\{|a(n)|^4\}}{E\{|a(n)|^2\}} \quad (5.1.11)$$

CMA 的代价函数为

$$J[e(n)] = E\{e^2(n)\} \quad (5.1.12)$$

则权向量更新公式为

$$\hat{\nabla}_f J[e(n)] = f(n) - \mu \hat{\nabla}_f J[e(n)] \quad (5.1.13)$$

式中, μ 是步长, μ 越大, 收敛速度越快, 但稳态误差越大; μ 越小, 不会使稳态误差增大, 但收敛速度越慢。

$\hat{\nabla}_f J[e(n)]$ 表示 $J[e(n)]$ 对 $f(n)$ 求偏导后取瞬时值。利用代价函数的最小化求梯度向量, 即

$$\hat{\nabla}_f J[e(n)] = \frac{\partial}{\partial f(n)} \{e^2(n)\} = 2e(n) \frac{\partial e(n)}{\partial f(n)} \quad (5.1.14)$$

而

$$\begin{aligned} \frac{\partial e(n)}{\partial f(n)} &= \frac{\partial [z(n)z^*(n)]}{\partial f(n)} = \frac{\partial [f^T(n)y(n)(y^T(n)f(n))^*]}{\partial f(n)} \\ &= 2y^*(n)y^T(n)f(n) = 2y(n)z^*(n) \end{aligned} \quad (5.1.15)$$

因此, Volterra 均衡器权向量的迭代公式为

$$f(n+1) = f(n) - 4\mu e(n)z^*(n)y(n) \quad (5.1.16)$$

Volterra 均衡器兼顾了系统的线性及非线性结构, 适应于各种非线性系统模型, 但它的权系数多(权向量长度大), 计算复杂度随记忆深度增加而迅速增加; 另一方面, 非线性均衡器输入向量的维数远远大于线性均衡器, 会使输入向量自相关矩阵的最大与最小特征值比值增大, 导致收敛速度相当缓慢。针对卫星信道的非线性特点和 Volterra 均衡器的缺陷, 本节拟从以下几个方面改进:

- (1) 用 Wiener 均衡器代替 Volterra 均衡器, 以减小均衡器结构的复杂性;
- (2) 用平衡正交多小波基函数对 Wiener 均衡器的输入信号进行变换, 以降低输入信号的自相关性;
- (3) 在 Wiener 均衡器的输出端增加一级判决反馈滤波器, 同时对判决反馈滤波器的输入信号, 也是 Wiener 均衡器的判决输出信号再作一次平衡正交多小波变换, 以降低判决反馈滤波器输出信号的自相关性。

5.1.4 基于平衡正交多小波双变换的非线性盲均衡算法

针对非线性卫星信道 Volterra 均衡系统的缺陷, 按 5.1.3 节的改进思想将图 5.1 的第二部分修改为图 5.5 所示的结构^[26]。

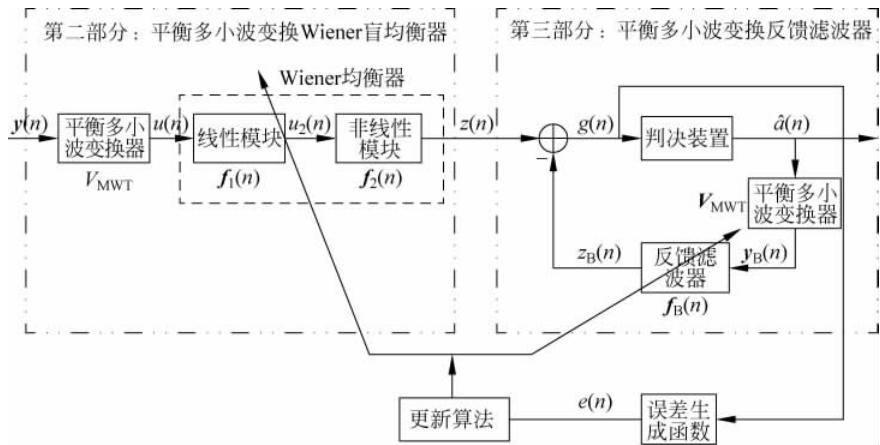


图 5.5 改进后的非线性卫星信道盲均衡器结构

图 5.5 中,平衡多小波变换器与 Wiener 盲均衡算法构成了平衡多小波变换 Wiener 盲均衡算法;平衡多小波变换器与判决反馈滤波器构成了平衡多小波变换反馈滤波器。这种进行了两次平衡多小波变换的盲均衡算法,称为基于平衡正交多小波双变换的盲均衡算法。

1. 均衡器的多小波表示

多小波由多个尺度函数构成,在任意尺度上支集无重叠,不同频率间的相关性较小,去相关性更好,信号经多小波变换后的自相关矩阵的最大与最小特征值之比更小,同时具有正交性、对称性和短支撑性,故用多小波构造的均衡器比用单小波构造的均衡器性能更优^[26,27]。

图 5.5 中,Wiener 均衡器的线性模块 $f_1(n)$ 的冲激响应为有限个值,用多小波表示为

$$f_{1i}(n) = \sum_{l=1}^r \sum_{j=1}^J \sum_{k=0}^{k_j-1} w_{j,k,l}(n) \phi_{j,k,l}(i) + \sum_{l=1}^r \sum_{k=0}^{k_j-1} v_{j,k,l}(n) \varphi_{j,k,l}(i) \quad (5.1.17)$$

当多小波函数和尺度函数确定后,均衡器的特性就由 $w_{j,k,l}(n)$ 和 $v_{j,k,l}(n)$ 确定,此时 $w_{j,k,l}(n)$ 和 $v_{j,k,l}(n)$ 就成了均衡器的权系数;均衡器的输出信号表示为

$$\begin{aligned} u(n) &= \sum_{i=0}^{N-1} f_{1i}(n) y(n-i) \\ &= \sum_{l=1}^r \left(\sum_{j=1}^J \sum_{k=0}^{k_j-1} w_{j,k,l}(n) r_{j,k,l}^w(n) + \sum_{k=0}^{k_j-1} v_{j,k,l}(n) r_{j,k,l}^v(n) \right) \end{aligned} \quad (5.1.18)$$

式中

$$r_{j,k,l}^w(n) = \sum_{i=0}^{N-1} \psi_{j,k,l}(i) y(n-i) \quad (5.1.19a)$$

$$r_{j,k,l}^v(n) = \sum_{i=0}^{N-1} \varphi_{j,k,l}(i) y(n-i) \quad (5.1.19b)$$

式中, $\psi_{j,k,l}(n)$ 表示尺度参数为 j 、平移参数为 k 的第 l 个小波基函数; $\varphi_{j,k,l}(n)$ 表示尺度参数为 J 、平移参数为 k 的第 l 个尺度函数。该式表明, 对输入信号 $y(n)$ 作离散正交多小波变换相当于将输入信号 $y(n)$ 与每一个尺度上的多小波基函数进行卷积运算。

由式(5.1.17)~式(5.1.19)知, 用多小波表示均衡器的实质是对均衡器输入信号进行平衡正交多小波变换, 但在实际应用过程中必须先进行一个预滤波处理, 而预滤波处理则会破坏多小波的对称性与正交性。为了避免这个问题, Lebrun 和 Vetterli 等提出了平衡正交多小波理论, 其基本思想是对已知的不平衡多小波作一定的平衡旋转, 使之变为平衡多小波。一般方法是采用正交酉矩阵对所述正交多小波作一阶或多阶平衡处理, 但采用一阶平衡处理有利于减小计算复杂度。因此, 本节采用一阶平衡处理, 选用的正交酉矩阵为

$$\mathbf{U} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad (5.1.20)$$

则平衡后的多低通滤波器和多高通滤波器分别为

$$\begin{cases} \tilde{\mathbf{H}}(\omega) = \mathbf{U}^T \mathbf{H}(\omega) \mathbf{U} \\ \tilde{\mathbf{G}}(\omega) = \mathbf{G}(\omega) \mathbf{U} \end{cases} \quad (5.1.21)$$

式中, $\mathbf{H}(\omega)$ 、 $\mathbf{G}(\omega)$ 分别为平衡前的多低通滤波器和多高通滤波器。

现根据 Mallat 算法, 对均衡器输入信号向量 $\mathbf{y}(n)$ 作平衡多小波分解。为了对均衡器输入信号向量 $\mathbf{y}(n) = [y(n), y(n-1), \dots, y(n-m+1)]^T$ 进行分解, 定义分块矩阵 \mathbf{P}_j 和 \mathbf{Q}_j , 其中, \mathbf{P}_j 和 \mathbf{Q}_j 分别由平衡后新的多低通滤波器 $\tilde{\mathbf{H}}(\omega)$ 和多高通滤波器 $\tilde{\mathbf{G}}(\omega)$ 的系数构成矩阵, 且

$$\mathbf{P}_j = \begin{bmatrix} \tilde{\mathbf{H}}_0 & \tilde{\mathbf{H}}_1 & \tilde{\mathbf{H}}_2 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \tilde{\mathbf{H}}_0 & \tilde{\mathbf{H}}_1 & \tilde{\mathbf{H}}_2 & 0 & \cdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{H}}_0 & \tilde{\mathbf{H}}_1 & \cdots \end{bmatrix}_{\frac{l}{2} \times l} \quad (5.1.22a)$$

$$\mathbf{Q}_j = \begin{bmatrix} \tilde{\mathbf{G}}_0 & \tilde{\mathbf{G}}_1 & \tilde{\mathbf{G}}_2 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \tilde{\mathbf{G}}_0 & \tilde{\mathbf{G}}_1 & \tilde{\mathbf{G}}_2 & 0 & \cdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & \cdots \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{G}}_0 & \tilde{\mathbf{G}}_1 & \cdots \end{bmatrix}_{\frac{l}{2} \times l} \quad (5.1.22b)$$

式中, $j=1 \sim J$, $l=m/2^{j-1}$, $J, l \in \mathbf{Z}$, 于是 Mallat 分解公式为

$$\mathbf{v}_J = \mathbf{P}_J \mathbf{P}_{J-1} \cdots \mathbf{P}_1 \mathbf{y}, \quad \mathbf{w}_J = \mathbf{Q}_J \mathbf{P}_{J-1} \cdots \mathbf{P}_1 \mathbf{y} \quad (5.1.23)$$

Mallat 分解结构如图 5.6 所示。

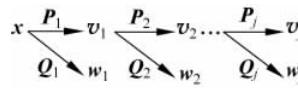


图 5.6 Mallat 分解结构

图中, \mathbf{v}_j 和 \mathbf{w}_j 分别为输入信号 $\mathbf{y}(n)$ 经过 j 层分解后的低通系数和高通系数, 信号经过平衡正交多小波分解后的向量为

$$\begin{aligned} \mathbf{u} &= [\mathbf{d}_1; \mathbf{d}_2; \cdots; \mathbf{d}_J; \mathbf{v}_J] \\ &= [\mathbf{Q}_1; \mathbf{Q}_2 \mathbf{P}_1; \mathbf{Q}_2 \mathbf{P}_1 \mathbf{P}_0; \cdots; \mathbf{Q}_J \mathbf{P}_{J-1} \cdots \mathbf{P}_2 \mathbf{P}_1; \mathbf{P}_J \mathbf{P}_{J-1} \cdots \mathbf{P}_2 \mathbf{P}_1] \mathbf{y} \end{aligned} \quad (5.1.24)$$

所以, 平衡正交多小波变换矩阵为

$$\mathbf{V}_{\text{MWT}} = [\mathbf{Q}_1; \mathbf{Q}_2 \mathbf{P}_1; \mathbf{Q}_2 \mathbf{P}_1 \mathbf{P}_0; \cdots; \mathbf{Q}_J \mathbf{P}_{J-1} \cdots \mathbf{P}_2 \mathbf{P}_1; \mathbf{P}_J \mathbf{P}_{J-1} \cdots \mathbf{P}_2 \mathbf{P}_1] \quad (5.1.25)$$

这时, 式(5.1.24)变为

$$\mathbf{u} = \mathbf{V}_{\text{MWT}} \mathbf{y} \quad (5.1.26)$$

2. 平衡正交多小波 Wiener 均衡算法

在图 5.5 中, Wiener 均衡器的线性模块输入信号向量为式(5.1.26)所示的信号向量 \mathbf{u} , 则 Wiener 均衡器线性模块的输出信号为

$$\begin{aligned} u_2(n) &= f_{10}(n)u(n) + f_{11}(n)u(n-1) + \cdots + f_{1M-1}(n)u(n-M+1) \\ &= \mathbf{f}_1^T(n)\mathbf{u}(n) = \mathbf{f}_1^T(n)\mathbf{V}_{\text{MWT}}\mathbf{y}(n) \end{aligned} \quad (5.1.27)$$

式中, $\mathbf{u} = [u(n), u(n-1), \dots, u(n-M+1)]^T$, $\mathbf{f}_1(n) = [f_{10}(n), f_{11}(n), \dots, f_{1M-1}(n)]^T$ 。

维纳均衡器非线性模块的输出为

$$z(n) = f_{21}(n)u_2(n) + f_{21}(n)u_2^2(n) + \cdots + f_{2M_2}(n)u_2^{M_2}(n) = \mathbf{f}_2^T(n)\mathbf{u}_2(n) \quad (5.1.28)$$

式中, $\mathbf{f}_2(n) = [f_{21}(n), f_{22}(n), \dots, f_{2M_2}(n)]^T$, $\mathbf{u}_2(n) = [u_2(n), u_2^2(n), \dots,$

$u_2^{M_2}(n)]^T$ 。

采用 CMA 分别对权向量 $\mathbf{f}_1(n)$ 和 $\mathbf{f}_2(n)$ 进行调整, 更新公式为

$$\begin{cases} \mathbf{f}_1(n+1) = \mathbf{f}_1(n) + \mu_{f_1} \hat{\mathbf{R}}^{-1}(n) e(n) z^*(n) \frac{\partial z(n)}{\partial \mathbf{f}_1(n)} \\ \mathbf{f}_2(n+1) = \mathbf{f}_2(n) + \mu_{f_2} \hat{\mathbf{R}}^{-1}(n) e(n) z^*(n) \frac{\partial z(n)}{\partial \mathbf{f}_2(n)} \end{cases} \quad (5.1.29)$$

式中, $\hat{\mathbf{R}}^{-1}(n) = \text{diag}[\sigma_{1,0,1}^{2\omega}(n), \dots, \sigma_{J,k,l}^{2\omega}(n), \dots, \sigma_{J,0,1}^{2\nu}(n), \dots, \sigma_{J,k,l}^{2\nu}(n)]$, $\sigma_{J,k,l}^{2\omega}(n)$ 与 $\sigma_{J,k,l}^{2\nu}(n)$ 分别为尺度参数为 J 、平移参数为 k 的第 l 个小波系数 $r_{J,k,l}^\omega(n)$ 与第 l 个尺度系数 $r_{J,k,l}^\nu(n)$ 的平均功率估计, 且

$$\begin{cases} \sigma_{J,k,l}^{2\omega}(n+1) = \beta_\sigma \sigma_{J,k,l}^{2\omega}(n) + (1-\beta_\sigma) |r_{J,k,l}^{2\omega}(n)|^2 \\ \sigma_{J,k,l}^{2\nu}(n+1) = \beta_\sigma \sigma_{J,k,l}^{2\nu}(n) + (1-\beta_\sigma) |r_{J,k,l}^{2\nu}(n)|^2 \end{cases} \quad (5.1.30)$$

式中, β_σ 为遗忘因子, 且 $0 < \beta_\sigma < 1$ 。

$$\begin{aligned} \frac{\partial z(n)}{\partial \mathbf{f}_1(n)} &= \frac{\partial (\mathbf{f}_2^T(n) \mathbf{u}_2(n))}{\partial \mathbf{f}_1(n)} \\ &= \mathbf{f}_2^T(n) \left(\left(\frac{\partial u_2(n)}{\partial \mathbf{f}_1(n)} \right)^T, \left(\frac{\partial u_2^2(n)}{\partial \mathbf{f}_1(n)} \right)^T, \dots, \left(\frac{\partial u_2^{M_2}(n)}{\partial \mathbf{f}_1(n)} \right)^T \right) \\ &= \mathbf{f}_2^T(n) \left(\left(\frac{\partial u_2(n)}{\partial \mathbf{f}_1(n)} \right)^T, 2u_2(n) \left(\frac{\partial u_2(n)}{\partial \mathbf{f}_1(n)} \right)^T, \dots, M_2 u_2^{M_2-1}(n) \left(\frac{\partial u_2(n)}{\partial \mathbf{f}_1(n)} \right)^T \right) \\ &= \begin{cases} \mathbf{u}(n)[\mathbf{f}_2^T(n)\mathbf{U}(n)] \\ \mathbf{u}(n-1)[\mathbf{f}_2^T(n)\mathbf{U}(n)] \\ \vdots \\ \mathbf{u}(n-M_1+1)[\mathbf{f}_2^T(n)\mathbf{U}(n)] \end{cases} \end{aligned} \quad (5.1.31)$$

式中, $\mathbf{U}(n) = [1, 2[\mathbf{f}_1^T(n)\mathbf{u}(n)], \dots, M_2[\mathbf{f}_1^T(n)\mathbf{u}(n)]^{M_2-1}]$

$$\frac{\partial z(n)}{\partial \mathbf{f}_2(n)} = \frac{\partial (\mathbf{f}_2^T(n) \mathbf{u}_2(n))}{\partial \mathbf{f}_2(n)} = \mathbf{u}_2(n) = \begin{cases} [\mathbf{f}_1^T(n)\mathbf{u}(n)] \\ [\mathbf{f}_1^T(n)\mathbf{u}(n)]^2 \\ \vdots \\ [\mathbf{f}_1^T(n)\mathbf{u}(n)]^{M_2} \end{cases} \quad (5.1.32)$$

称式(5.1.24)~式(5.1.32)为基于平衡正交多小波变换的 Wiener 盲均衡算法, 该算法将平衡正交多小波变换放在维纳均衡器之前, 对其输入信号进行正交多小波变换, 减少了信号自相关, 从而达到加快收敛的目的。

3. 平衡正交多小波双变换判决反馈滤波器

借助文献[28]的思想, 在图 5.5 所示的判决反馈滤波器中, 对判决反馈滤波器 $f_B(n)$ 的输入信号, 也就是维纳均衡器的判决输出信号 $\hat{a}(n)$ 作平

衡正交多小波变换,得变换后的判决反馈滤波器 $f_B(n)$ 的输入信号为

$$\mathbf{y}_b(n) = \mathbf{V}_{\text{MWT}} \hat{\mathbf{a}}(n) \quad (5.1.33)$$

此时,判决器输入信号为

$$g(n) = z(n) - z_B(n) = z(n) - f_B(n) \mathbf{y}_B(n) \quad (5.1.34)$$

式中,

$$z_B(n) = f_B(n) \mathbf{y}_B(n) \quad (5.1.35)$$

式中, $z_B(n)$ 为判决反馈滤波器 $f_B(n)$ 的输出信号。由于这个判决反馈滤波器的输入信号作了平衡正交多小波变换,改变了判决反馈滤波器结构,故称之为平衡正交多小波判决反馈滤波器。平衡正交多小波判决反馈滤波器权向量的更新公式为

$$f_B(n+1) = f_B(n) - \mu_B \hat{\mathbf{R}}_B^{-1}(n) e(n) \mathbf{y}_b(n) g^*(n) \quad (5.1.36)$$

式中, $\hat{\mathbf{R}}_B^{-1}(n) = \text{diag}[\sigma_{B(1,0,1)}^{2\omega}(n), \dots, \sigma_{B(J,k,l)}^{2\omega}(n), \dots, \sigma_{B(1,0,1)}^{2v}(n), \dots, \sigma_{B(J,k,l)}^{2v}(n)]$, $\sigma_{B(J,k,l)}^{2\omega}(n)$ 与 $\sigma_{B(J,k,l)}^{2v}(n)$ 分别为尺度参数为 J 、平移参数为 k 的第 l 个小波系数 $r_{B(J,k,l)}^\omega(n)$ 与第 l 个尺度系数 $r_{B(J,k,l)}^v(n)$ 的平均功率估计,且

$$\begin{cases} \sigma_{B(J,k,l)}^{2\omega}(n+1) = \beta_B \sigma_{B(J,k,l)}^{2\omega}(n) + (1-\beta_B) |r_{B(J,k,l)}^{2\omega}(n)|^2 \\ \sigma_{B(J,k,l)}^{2v}(n+1) = \beta_B \sigma_{B(J,k,l)}^{2v}(n) + (1-\beta_B) |r_{B(J,k,l)}^{2v}(n)|^2 \end{cases} \quad (5.1.37)$$

式中, β_B 为遗忘因子,且 $0 < \beta_B < 1$ 。

用平衡正交多小波对 Wiener 均衡器的输入信号和反馈滤波器的输入信号进行变换,既改变了 Wiener 均衡器的结构又改变了反馈滤波器的结构,Wiener 均衡器和反馈滤波器的权向量由 CMA 更新,这样得到的非线性卫星信道盲均衡算法,称为基于多小波双变换的非线性盲均衡算法 (Double Multi-Wavelet Transform based Non-linear Constant Modulus blind equalization Algorithm and Decision Feedback Equalizer, DMWT-NCMA-DFE)。

4. 计算复杂度

现对 Volterra 盲均衡算法和本节 DMWT-NCMA-DFE 的计算复杂度进行分析。由于乘法运算量远大于加法运算量,故以算法运行一次所需的乘法次数作为计算量的衡量标准。由式(5.1.5)知,Volterra 均衡器三阶非线性项内部存在大量乘法运算,信号通过均衡器需进行 $3N^3 + N$ 次乘法运算,由式(5.1.16)知每一次权向量更新需要 $N^3 + N + 1$ 次乘法运算,则 Volterra 盲均衡器进行一次权向量更新所需乘法次数为 $4N^3 + 2N + 1$ 。假设输入信号长度为 H 且信号为实信号,由于 DMWT-NCMA-DFE 在每一

次权向量迭代过程中,都需要运行 H 点平衡正交多小波变换,由式(5.1.26)计算输入信号 $\mathbf{y}(n)$ 的正交多小波变换(注意到 V_{MWT} 是 $H \times H$ 正交矩阵),则所需的最多乘法次数为 H^2 次。考虑到实际信号长度很长,而滤波器长度很短,即 V_{MWT} 为稀疏矩阵,假定 V_{MWT} 中每一行的非零元素个数为 M_1 ($M_1 \ll H$),则计算式(5.1.22)所需的乘法次数为 $M_1 H$;当信号经过 Wiener 均衡器时,由式(5.1.27)和式(5.1.28)所带来的运算量为 $(1+2+\dots+N+M_1)$;由式(5.1.31)和式(5.1.32)所带来的运算量为 $(1+2+\dots+N+M_1)$;经过判决反馈模块后由式(5.1.33)至式(5.1.37)计算所需运算量为 $M_1 H + 7M_1$,则 DMWT-NCMA-DFE 所需的乘法运算量为 $N^2 + N + 2M_1 H + 7M_1$ 。显然,低于 Volterra 盲均衡器的计算量。

5.1.5 仿真验证

为了验证 DMWT-NCMA-DFE 的性能,以 Volterra 盲均衡算法、非线性判决反馈盲均衡算法(Non-linear Constant Modulus blind equalization Algorithm and Decision Feedback Equalizer, NCMA-DFE)、基于平衡正交多小波变换的非线性判决反馈盲均衡算法(Multi-Wavelet Transform based Non-linear Constant Modulus blind equalization Algorithm and Decision Feedback Equalizer, MWT-NCMA-DFE)为比较对象。实验中,Volterra 级数模型记忆长度为 3,维纳均衡器的线性模块和非线性模块的参数 $M_1 = 16$ 和 $M_2 = 3$,初始化权向量 $f_1(0) = [1, 0, \dots, 0]^T$ 和 $f_2(0) = [1, 0, \dots, 0]^T$;反馈滤波器权长 $M_B = 5$,初始化权向量 $f_B(0) = [0, 0, \dots, 0]^T$ 。APSK 星座外、内半径比取为 2.73;发送滤波器及接收滤波器均采用平方根升余弦滤波器,其滚降因子为 0.35,采样率为 8,信噪比为 20, $J = 2$,平均功率初始化为 25。200 次蒙特卡罗仿真结果,如图 5.7 所示。

图 5.7(a)表明,DMWT-NCMA-DFE 的收敛速度比 MWT-NCMA-DFE 约快 4000 步,Volterra 盲均衡算法与 NCMA-DFE 的收敛速度非常缓慢,几乎没有收敛;而在迭代 2000 次时 DMWT-NCMA-DFE(已收敛)的均方误差比 MWT-NCMA-DFE 与 NCMA-DFE(未收敛)的减小 3dB、比 Volterra 均衡算法的减小 5dB。图 5.7(c)~(f)表明,16APSK 信号通过非线性卫星信道时,会产生严重失真(图 5.7(b)所示),信号星座十分模糊,无法分辨;当用 DMWT-NCMA-DFE、MWT-NCMA-DFE、NCMA-DFE 及 Volterra 盲均衡算法恢复信号星座时,DMWT-NCMA-DFE 的输出星座最

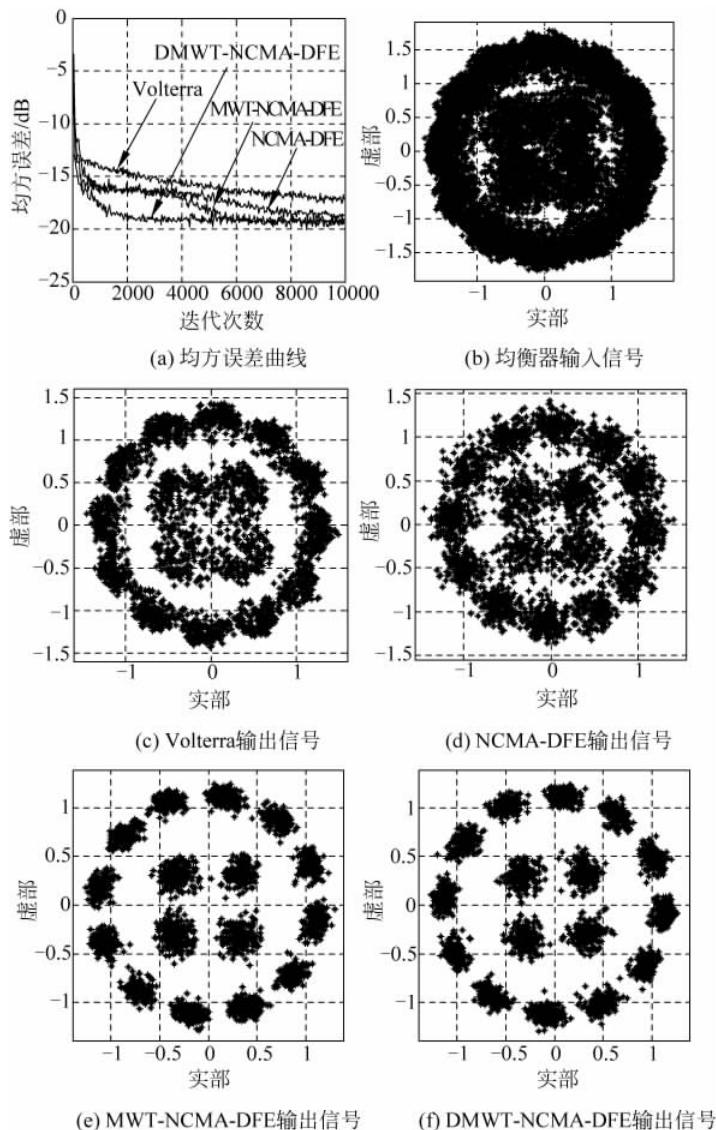


图 5.7 仿真结果

清晰、最集中、最紧凑，效果最好。

为了检验本节的 DMWT-NCMA-DFE 对三种调制信号 16QAM、16PSK、16APSK 通过非线性卫星信道的恢复能力，参数条件相同时，仿真结果如图 5.8 所示。

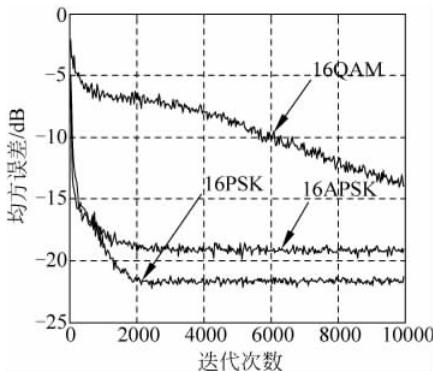


图 5.8 三种调制信号的 DMWT-NCMA-DFE 均方误差曲线

图 5.8 表明, 对 16QAM 信号, DMWT-NCMA-DFE 的收敛速度极其缓慢, 迭代 10000 次时还未收敛; 对于 16PSK 信号, DMWT-NCMA-DFE 收敛后的均方误差比 16APSK 减小了约 3dB, 但收敛速度比 16APSK 慢了 500~1000 步。

5.2 基于多小波神经网络的非线性盲均衡算法

5.2.1 神经网络模型

1. 神经元模型

神经元是神经网络处理信息的基础单元, 它主要包含三种基本元素: 突触或连接链、加法器和激活函数, 如图 5.9 所示。

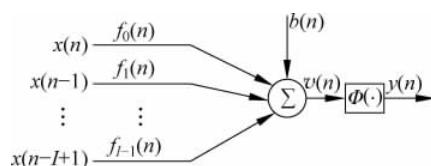


图 5.9 神经元模型

神经元的输出为

$$\begin{aligned}y(n) &= \Phi(v(n)) = \Phi\left(\sum_{i=0}^{l-1} x(n-i)f_i(n) + b(n)\right) \\&= \Phi(x^T(n)f(n) + b(n))\end{aligned}\quad (5.2.1)$$

式中, $x(n) = \{x(n), x(n-1), \dots, x(n-I+1)\}$ 为输入信号; $f(n) = \{f_0(n), f_1(n), \dots, f_{l-1}(n)\}$ 为神经元连接权向量; $b(n)$ 为外部偏置, 其值若为正, 则增加激活函数的网络输入, 反之为负, 则相应地降低; $y(n)$ 为神经元的输出; $\Phi(\cdot)$ 为激励函数, 一般有三种基本结构。

1) 阈值函数

一般称为 Heaviside 函数, 即

$$\Phi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad \text{或 } \Phi(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (5.2.2)$$

2) 高斯型函数

$$\Phi(x) = \exp\left(-\frac{1}{2\sigma_i^2} \sum_j (x_j - w_{ji})^2\right) \quad (5.2.3)$$

3) Sigmoid 函数

此函数是 S 形的, 是人工神经网络中最常用的激活函数, 具有平滑、渐近、单调等性质, 在线性和非线性之间表现出良好的平衡性能, 常用的函数形式为

$$\Phi(x) = (1 + e^{-\alpha x}) - 1 \quad (5.2.4)$$

式中, 参数 α 控制函数的斜率。

2. 神经网络模型

神经网络是一个由互相连接的突触节点和激活连接构成的有向图。其中, 神经元的构造方式和训练网络的学习算法是紧密联系的。从连接方式看, 神经网络主要分为前馈神经网络和反馈神经网络。

1) 前馈神经网络

前馈神经网络如图 5.10 所示。

前馈神经网络中, 前一层神经元的输出作为该层神经元的输入, 该层神经元的输出作为下一层神经元的输入, 输入和输出层均与外界相连。其他中间层作为隐含层, 以某种方式存在于外部输入和网络输出之间。加入一个或多个隐含层, 可以提高网络性能。

2) 反馈神经网络

反馈神经网络中至少有一个反馈环, 如图 5.11 所示, 单层网络的每一个神经元的输出作为输入信号反馈到其他神经元中, 神经元的输入信号受

神经元输出信号的影响。

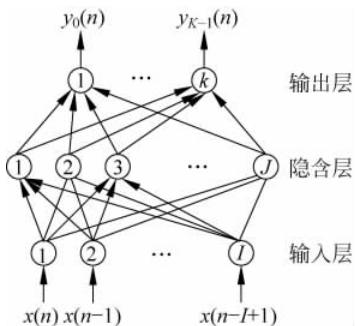


图 5.10 前馈神经网络结构

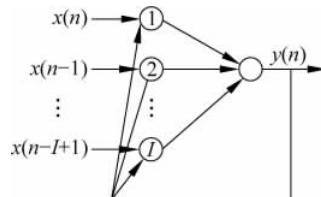


图 5.11 反馈神经网络

5.2.2 基于多小波神经网络的非线性盲均衡算法

前馈神经网络(Feed-Forward Neural Network, FNN),从结构上讲,是分层的,信号从输入层到它的下一层,并且每一层的任意一个单元都有连接,其传统的训练方式为反向传播算法(BP 算法)。从系统论上讲,前馈神经网络是稳态的非线性系统,具有良好的非线性分类功能。

1. 神经网络盲均衡系统模型

神经网络盲均衡算法原理,如图 5.12 所示。 $a(n)$ 表示发射信号, $x(n)$ 表示信道输出信号, $w(n)$ 表示高斯白噪声, $y(n)$ 表示均衡器输入信号, $z(n)$ 表示均衡器输出信号, $\hat{a}(n)$ 表示判决器输出信号。

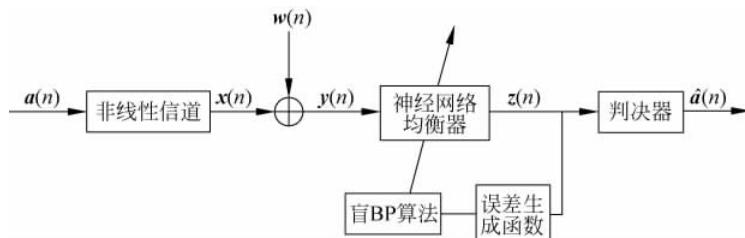


图 5.12 神经网络盲均衡系统

Cybenc 已经证明: 神经网络结构可以任意精度逼近一个连续函数,本节采用三层前馈神经网络结构,如图 5.13 所示。

图中, $f_{ij}(n)$ 为输入层第 i 个神经元至隐含层第 j 个神经元间的连接权

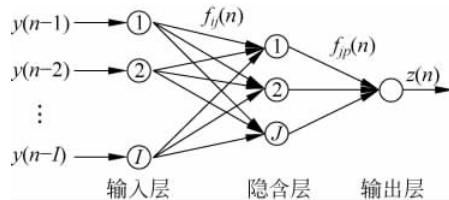


图 5.13 前馈神经网络基本结构

系数, $f_{jp}(n)$ 为隐含层第 j 个神经元至输出层第 p ($p=1, 2, \dots, P$) 个神经元之间的连接权系数, 输入层的输入 $\mathbf{y}(n)=\{y(n-1), y(n-2), \dots, y(n-I)\}^T$; 隐含层第 j 个神经元的输入为 $u_j^I(n)$; 隐含层第 j 个神经元的输出 $v_j^I(n)$; 输出层第 p 个神经元的输出为 $u_p^P(n)$; 神经网络的输出为 $z(n)$ 。神经元的激励函数选为

$$F(x) = x + a \sin(\pi x) \quad (5.2.5)$$

式中, $-\infty < x < \infty$; $0 < a < 1$; x 表示 $u_j^I(n)$ 或 $u_p^P(n)$, 该函数对神经网络输入信号具有良好的辨识能力。神经网络的状态方程组为

$$u_j^I(n) = \sum_{i=1}^I f_{ij}(n) y(n-i) \quad (5.2.6)$$

$$v_j^I(n) = F\left(\sum_{i=1}^I f_{ij}(n) y(n-i)\right) \quad (5.2.7)$$

$$u_p^P(n) = \sum_{j=1}^J f_{jp}(n) v_j^I(n) \quad (5.2.8)$$

$$z(n) = F\left(\sum_{j=1}^J f_{jp}(n) v_j^I(n)\right) \quad (5.2.9)$$

神经网络盲均衡算法中, 神经元之间连接权系数的确定以及激励函数的选择对算法性能起着关键作用。

2. 基于多小波神经网络的非线性盲均衡算法

将平衡正交多小波变换引入到神经网络的非线性盲均衡算法的思路为在同一分解尺度下, 对神经网络均衡器的输入信号进行正交多小波变换, 由经过多小波变换后的信号对神经网络输入层至隐含层之间权向量进行调整。基于多小波神经网络的非线性盲均衡算法 (Multi-Wavelet Transform Feedback Neural Network nonlinear blind equalization algorithm, MWT-FNN) 原理, 如图 5.14 所示。

信道输出信号 $y(n)$ 经过多小波变换后得神经网络的输入信号 $\mathbf{R}(n)$, 即

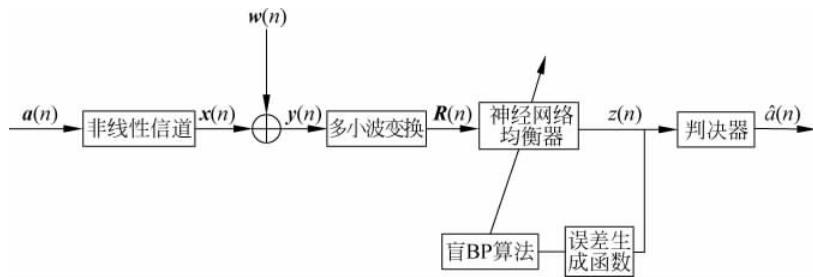


图 5.14 多小波神经网络盲均衡算法

$$\mathbf{R}(n) = \mathbf{V}_{\text{MWT}} \mathbf{y}(n) \quad (5.2.10)$$

式中, \mathbf{V}_{MWT} 表示平衡正交多小波变换。根据式(5.2.10), 此时隐含层的输入、输出修改为

$$u_j^l(n) = \sum_{j=1}^I f_{ij}(n) R_i(n) \quad (5.2.11)$$

$$v_j^l(n) = F\left(\sum_{i=1}^I f_{ij}(n) R_i(n)\right) \quad (5.2.12)$$

$$u_p^p(n) = \sum_{j=1}^J f_{jp}(n) v_j^l(n) \quad (5.2.13)$$

$$z(n) = F\left(\sum_{j=1}^J f_{jp}(n) v_j^l(n)\right) \quad (5.2.14)$$

输入层权向量的迭代公式为

$$\begin{aligned} f_{ij}(n+1) &= f_{ij}(n) - \mu_{ij} \frac{\partial J(n)}{\partial f_{ij}(n)} \\ &= f_{ij}(n) - \mu_{ij} \frac{\partial J(n)}{\partial z(n)} \cdot \frac{\partial z(n)}{\partial u_p^p(n)} \cdot \frac{\partial u_p^p(n)}{\partial v_j^l(n)} \cdot \\ &\quad \frac{\partial v_j^l(n)}{\partial u_j^l(n)} \cdot \frac{\partial u_j^l(n)}{\partial f_{ij}(n)} \end{aligned} \quad (5.2.15)$$

式中, μ_{ij} 为权向量迭代步长。

$$\frac{\partial J(n)}{\partial z(n)} = 2 |z(n)| (|z(n)|^2 - R^2) \quad (5.2.16)$$

$$\frac{\partial z(n)}{\partial u_p^p(n)} = F'(u_p^p(n)) \quad (5.2.17)$$

$$\frac{\partial u_p^p(n)}{\partial v_j^l(n)} = f_{jp}(n) \quad (5.2.18)$$

$$\frac{\partial v_j^J(n)}{\partial u_j^J(n)} = F'(u_j^J(n)) \quad (5.2.19)$$

$$\frac{\partial v_j^J(n)}{\partial f_{ij}(n)} = R_i(n) \quad (5.2.20)$$

同理, 隐含层到输出层权向量的迭代公式为

$$f_{jp}(n+1) = f_{jp}(n) - \mu_{jp} \frac{\partial J(n)}{\partial f_{jp}(n)} \quad (5.2.21)$$

式中, μ_{jp} 为迭代步长。而

$$\frac{\partial J(n)}{\partial f_{jp}(n)} = \frac{\partial J(n)}{\partial z(n)} \cdot \frac{\partial z(n)}{\partial u_p^P(n)} \cdot \frac{\partial u_p^P(n)}{\partial f_{jp}(n)} \quad (5.2.22)$$

从而得到

$$\begin{aligned} f_{jp}(n+1) &= f_{jp}(n) - \mu_{jp} \frac{\partial J(n)}{\partial f_{jp}(n)} \\ &= f_{jp}(n) - \mu_{jp} \frac{\partial J(n)}{\partial z(n)} \cdot \frac{\partial z(n)}{\partial u_p^P(n)} \cdot \frac{\partial u_p^P(n)}{\partial f_{jp}(n)} \end{aligned} \quad (5.2.23)$$

式中, μ_{jp} 为步长。

$$\frac{\partial u_p^P(n)}{\partial f_{jp}(n)} = v_j^J(n) \quad (5.2.24)$$

式(5.2.10)至式(5.2.24)构成了基于多小波神经网络的非线性盲均衡算法。

3. 计算复杂度

MWT-FNN 算法每更新一次权向量都要对信号 $y(n)$ 作正交多小波变换。与常规 FNN 算法相比, 计算量有一定程度的增加。现以乘法次数为复杂度的衡量标准, 设输入信号长度为 N , 神经网络输入层均衡器权长为 L , 则计算式(5.2.10)乘法次数为 LN , 式(5.2.11)~式(5.2.14)的乘法次数为 $(2I+2J)N$, 由式(5.2.15)~式(5.2.20)得输入层权向量更新一次所需的乘法次数为 $8N$ 。同理, 隐含层到输出层权向量迭代一次所需的乘法次数为 $6N$, 则 MWT-FNN 所需的乘法次数为 $(14+2I+2J+L)N$ 。

5.2.3 仿真验证

为了验证基于多小波神经网络的非线性盲均衡算法性能, 输入信号采用 2PAM 调制方式, 神经网络均衡器权向量长度为 16, 第 4 个抽头初始化, 步长 $\mu=1\times10^{-5}$, 信噪比为 20dB; 多小波神经网络输入层到隐含层的权向量长度为 16, 采用 DB2 小波, 步长 $\mu=8\times10^{-4}$, 非线性 Wiener 信道模型的

输入输出关系,如图 5.15 所示。

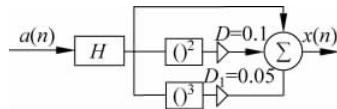


图 5.15 非线性信道 Wiener 模型

图中, H 表示非线性信道中的线性部分, $H = [0.3122 \quad -0.1040 \\ 0.8908 \quad 0.3134]$ 。非线性部分中, $D=0.1$ 为二次项系数, $D_1=0.05$ 为三次项系数。500 次蒙特卡罗仿真结果如图 5.16 所示。

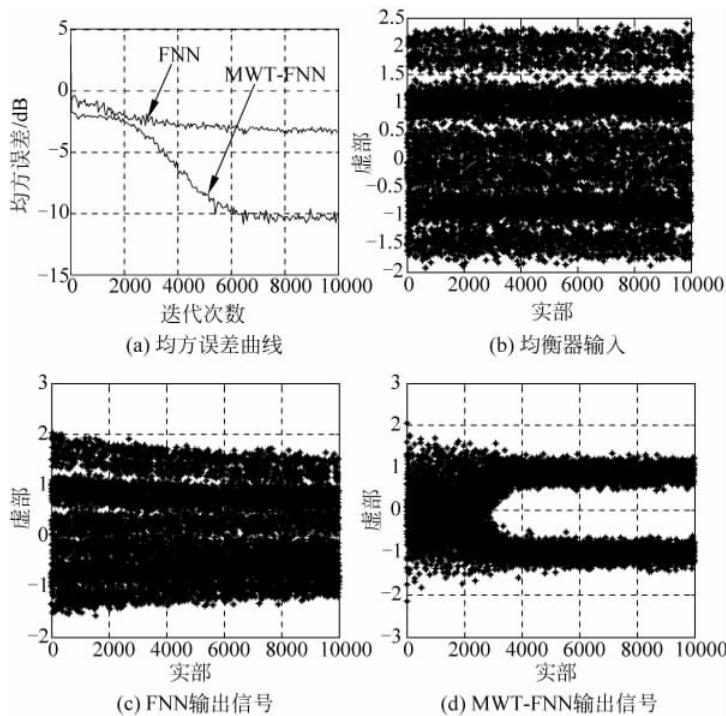


图 5.16 仿真结果

图 5.16(a)表明,基于多小波神经网络的非线性盲均衡算法(MWT-N-FNN)的收敛速度比神经网络的非线性盲均衡算法(N-FNN)快了约 2000 步,稳态误差减少 6dB,星座图更清晰、紧凑。这可能是由于正交多小波具有的良好解相关性、神经网路的非线性及并行分布处理信号的能力强引起的。

5.3 基于支持向量机和神经网络的非线性盲均衡算法

5.3.1 支持向量机基础

支持向量机(SVM)^[30]是从线性可分情况下的最优分类面发展而来的,其思想如图 5.17 所示。

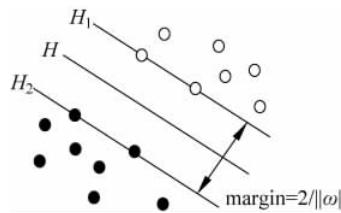


图 5.17 最优分类面示意图

图 5.17 中,黑点和白点分别代表两类不同的数据样本, H 为分类线, H_1 、 H_2 表示各类中离分类线最近且平行于分类线并尽可能多的经过各类数据样本的直线,它们之间的距离 Δ 叫作分类间隔(margin)。最优分类线不但要求分类线能将两类数据集分开,而且还要使其中间的分类距离最大。其中,前者保证了最小经验风险,后者确保了推广性的界置信范围最小,即真实风险最小。如果二维分类推广到多维空间,就能得到最优分类面。

1. 最优分类面

设有线性可分样本集为 $(x_i, y_i), i=1, 2, \dots, n, x_i \in \mathbf{R}^d, y \in \{+1, -1\}$ 是类别符号。一般线性判别函数为 $f(x) = \omega \cdot x + b$, 对应的分类面方程为

$$\omega \cdot x + b = 0 \quad (5.3.1)$$

将判别函数归一化,要求所有样本都能正确分类并满足条件 $|f(x)| \geq 1$ 。此时,离分类面最近的样本 $|f(x)| = 1$, 即满足条件

$$y_i[(\omega \cdot x) + b] - 1 \geq 0, \quad i = 1, 2, \dots, n \quad (5.3.2)$$

此时分类间隔 $2 / \|\omega\|^2$ 最大等价于 $\|\omega\|^2$ 最小。满足式(5.3.2)且使 $\frac{1}{2} \|\omega\|^2$ 最小的分类面就是图 5.17 中的最优分类面 H ; 平行于分类面

的 H 的超平面 H_1, H_2 上的数据样本就是式(5.3.2)中使等号成立的那些数据样本。这些数据样本就是支撑整个最优分类面的支持向量(Support Vector, SV)。

因此, 最优分类面问题可转化为式(5.3.2)的约束优化问题, 即求函数

$$J(\omega) = \frac{1}{2} \|\omega\|^2 \quad (5.3.3)$$

的最小值。这个约束优化问题称为原问题, 它有两个特点:

(1) 代价函数 $J(\omega)$ 是 ω 的凸函数;

(2) 约束条件关于 ω 是线性的。现由 Lagrange 乘子法解决约束最优问题, Lagrange 函数定义为

$$L(\omega, b, \alpha) = \frac{1}{2} \|\omega\|^2 - \sum_{i=1}^n \alpha_i [y_i(\omega \cdot x_i + b) - 1] \quad (5.3.4)$$

式中, $\alpha_i \geq 0$ 为 Lagrange 乘子。为了求式(5.3.4)的最小值, 式(5.3.4)分别对 ω, b, α_i 求偏微分并令其值等于 0, 得

$$\begin{cases} \frac{\partial L}{\partial \omega} = 0 \Rightarrow \omega = \sum_{i=1}^n \alpha_i y_i x_i \\ \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \\ \frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow \alpha_i [y_i(\omega \cdot x_i + b) - 1] = 0 \end{cases} \quad (5.3.5)$$

利用式(5.3.2)和式(5.3.5)的约束条件, 可以将上述最优分类面的求解问题转变为由凸二次规划方法寻找最优解的对偶问题, 即

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{s. t. } \alpha_i \geq 0, \quad i = 1, 2, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (5.3.6)$$

式中, α_i 为对应的 Lagrange 乘子, 这是一个存在唯一解的二次函数寻优问题。若 α^* 为最优解, 则

$$\omega^* = \sum_{i=1}^n \alpha^* y_i x_i \quad (5.3.7)$$

式中, α^* 是非零样本, 即为支持向量。

分类阈值 b^* 可由约束条件 $\alpha_i [y_i(\omega \cdot x_i + b) - 1] = 0$ 求解。求得的最优分类面函数为

$$f(x) = \operatorname{sgn}\{(\boldsymbol{\omega} \cdot \mathbf{x}) + b\} = \operatorname{sgn}\left\{\sum_{i=1}^n \boldsymbol{\alpha}_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*\right\} \quad (5.3.8)$$

上述就是对支持向量机的一般表述。

2. 广义的最优分类面

上述方法只有在确保训练样本全部被正确分类的前提下,才是可行的,也只有这样,才能通过最大化分类间隔获得最好的推广性能。当最优分类面不能把两类数据点正确分开时,若要在经验风险和推广性能间获得平衡,可以考虑引入松弛因子 ξ 并默许存在错分样本。此时,分类面 $\boldsymbol{\omega} \cdot \mathbf{x} + b = 0$ 满足的条件为

$$y_i[(\boldsymbol{\omega} \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i, \quad i = 1, 2, \dots, n \quad (5.3.9)$$

当 $0 < \xi_i < 1$ 时,样本点 \mathbf{x}_i 正确分类;当 $\xi_i \geq 1$ 时,样本点 \mathbf{x}_i 被错分。

在最小化代价函数 $\frac{1}{2} \|\boldsymbol{\omega}\|^2$ 中加入惩罚项 $C \sum_{i=1}^n \xi_i$, 则代价函数为

$$J(\boldsymbol{\omega}, \xi) = \frac{1}{2} \|\boldsymbol{\omega}\|^2 + C \sum_{i=1}^n \xi_i \quad (5.3.10)$$

式中,C 是惩罚因子,是一个正常数。与线性可分的情况相似,式(5.3.10)可通过凸二次规划来实现,如式(5.3.11)所示。

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{s. t. } 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (5.3.11)$$

对于非线性分类问题,若在原始空间中训练样本未能被正确分类,则可以通过某个非线性映射转化为高维特征空间中的线性问题,支持向量机(SVM)就能通过核函数变换来解决这个问题。

3. 核函数

核函数是在支持向量机中被广泛应用的关键技术之一。核函数变换是一种非常便捷的计算途径,这个方法的原理是利用核函数变换 Φ 将 n 维向量空间中的随机向量 \mathbf{x} 映射到高维特征空间,将非线性不可分为线性可分,并在高维空间中设计学习算法,如图 5.18 所示。

常见的向量机核函数有三种: q 次多项式函数、径向基函数和 Sigmoid 函数。

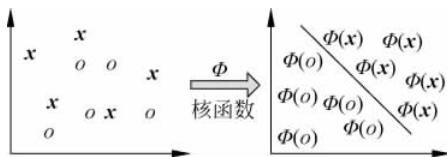


图 5.18 SVM 核函数变换示意图

1) q 次多项式函数

核函数选取为

$$K(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x}, \mathbf{x}_i) + 1]^q \quad (5.3.12)$$

则 SVM 的判别函数为

$$f(\mathbf{x}, \alpha) = \operatorname{sgn} \left\{ \sum_{i=1}^s \alpha_i y_i [(\mathbf{x}_i \cdot \mathbf{x}) + 1]^q - \mathbf{b} \right\} \quad (5.3.13)$$

式中, s 为支持向量的个数。多项式次数 q 的大小可以控制系统的 VC 维数。

2) 基函数

核函数选为

$$K(\mathbf{x}, \mathbf{x}_i) = \exp \left\{ - \frac{|\mathbf{x} - \mathbf{x}_i|^2}{\sigma^2} \right\} \quad (5.3.14)$$

则 SVM 的判别函数为

$$f(\mathbf{x}) = \operatorname{sgn} \left\{ \sum_{i=1}^{N_c} \alpha_i \exp \left\{ - \frac{|\mathbf{x} - \mathbf{x}_i|^2}{\sigma^2} \right\} - \mathbf{b} \right\} \quad (5.3.15)$$

式中, 径向基函数的中心位置由支持向量 \mathbf{x}_i 确定, N_c 是中心数目。假定数据样本在输入空间内是超球面分布, 则尺度参数 σ 的取值即为超球面的半径。

3) Sigmoid 函数

核函数为

$$K(\mathbf{x}, \mathbf{x}_i) = \tanh[\nu(\mathbf{x} \cdot \mathbf{x}_i) + a] \quad (5.3.16)$$

式中, Sigmoid 函数是一个双曲正切函数。当且仅当 ν 和 a 的取值满足 Mercer 条件时, 如 $\nu=2, a=1$ 。此时, 支持向量机的判别函数为

$$f(\mathbf{x}) = \operatorname{sgn} \left\{ \sum_{i=1}^s \alpha_i \tanh[\nu(\mathbf{x} \cdot \mathbf{x}_i) + a] + b \right\} \quad (5.3.17)$$

式(5.3.17)就是三层神经网络模型的判别函数, 其隐含层结点与支持向量相对应。

5.3.2 支持向量机回归原理

回归问题就是设定一个新的输入数据样本 x , 根据给定的样本数据推断对应的输出值 y_i (y_i 可以是任意实数)。设给定的数据样本集合为 $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, $x_i \in \mathbf{R}^n$, $y_i \in \mathbf{R}$, $i=1, 2, \dots, l$, 则回归问题就是寻找 \mathbf{R}^n 上的一个函数 $y=f(x)$, 由任一输入 x 推断所对应的 y 值, 如图 5.19 所示。

图 5.19 中, “ \times ”表示数据样本中集中的样本点, 曲线表示回归目标函数 $f(x)$ 。

将支持向量机应用到回归问题中, 保留了最大间隔算法的主要特征, 非线性函数可以从核特征空间中的线性学习机器得到。

1. 线性支持向量机回归

设有 n 维样本向量, 空间内 l 个数据样本为 $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l) \in \mathbf{R}^n \times \mathbf{R}$ 。

设线性函数为

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \quad (5.3.18)$$

优化问题即最小化问题表示为

$$R(\mathbf{w}, \xi, \xi^*) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (5.3.19)$$

式中, $\frac{1}{2} \mathbf{w} \cdot \mathbf{w}$ 使函数更为平坦, 能提高泛化能力。为了使回归问题更加精确, 这里引入“软化方法”。设 C 为惩罚参数, ξ, ξ^* 为松弛变量, 则约束条件写为

$$f(x_i) - y_i \leq \xi_i^* + \epsilon \quad (5.3.20)$$

$$f(x_i) - y_i \geq \xi_i - \epsilon \quad (5.3.21)$$

式中, $\xi_i^*, \xi_i \geq 0, i=1, 2, \dots, l, \epsilon$ 为一正常数。

引入 Lagrange 函数, 则欲解决的凸二次优化问题为

$$\begin{aligned} L(\mathbf{w}, b, \xi, \xi^*, \alpha, \alpha^*, \gamma, \gamma^*) \\ = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l \alpha_i [\xi_i + \epsilon - y_i + f(x_i)] - \\ \sum_{i=1}^l \alpha_i^* [\xi_i^* + \epsilon + y_i - f(x_i)] - \sum_{i=1}^l (\xi_i \gamma_i - \xi_i^* \gamma_i^*) \end{aligned} \quad (5.4.22)$$

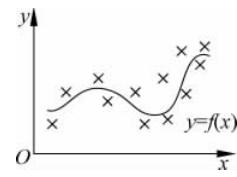


图 5.19 回归算法几何意义

式中, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)$ 、 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_N^*)$ 为 Lagrange 乘子。由函数 L 对 a 、 b 、 ξ_i 、 ξ_i^* 求偏导数, 可得极小值; 由函数 L 对 α_i 、 α_i^* 、 γ_i 、 γ_i^* 求偏导数, 可得极大值。函数 L 的极值应满足的条件为

$$\frac{\partial L}{\partial a} = 0, \quad \frac{\partial L}{\partial b} = 0, \quad \frac{\partial L}{\partial \xi_i} = 0, \quad \frac{\partial L}{\partial \xi_i^*} = 0 \quad (5.3.23)$$

从而得

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \quad (5.3.24)$$

$$a = \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i \quad (5.3.25)$$

$$C - \alpha_i - \gamma_i = 0 \quad (5.3.26)$$

$$C - \alpha_i^* - \gamma_i^* = 0 \quad (5.3.27)$$

式中, $i=1, 2, \dots, l$ 。

由式(5.3.18)~式(5.3.27), 得优化问题的对偶形式为

$$\begin{aligned} W(a, a^*) = & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(x_i \cdot x_j) + \\ & \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \sum_{i=1}^l (\alpha_i + \alpha_i^*) \epsilon \end{aligned} \quad (5.3.28)$$

其约束为

$$a = \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i, \quad 0 \leq \alpha_i, \alpha_i^* \leq C \quad (5.3.29)$$

这也是一个二次优化问题。

2. 非线性支持向量机回归原理

非线性回归和非线性分类问题相似, 数据通过非线性映射, 将低维特征空间内的回归问题转变为高维特性空间内的线性回归问题。

解决回归问题的关键是核函数 $K(x, y)$ 的选用。这里, 优化问题就是满足式(5.3.29)约束条件下优化函数的最大化问题, 即

$$\begin{aligned} W(a, a^*) = & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i \cdot x_j) + \\ & \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \sum_{i=1}^l (\alpha_i + \alpha_i^*) \epsilon \end{aligned} \quad (5.3.30)$$

此时

$$a = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \phi(x_i) \quad (5.3.31)$$

函数 $f(x)$ 可直接表示为

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(\mathbf{x}, \mathbf{x}_i) + b \quad (5.3.32)$$

按照 Kuhn-Tucker 定理, 有

$$\alpha_i [\epsilon + \xi_i - y_i + f(x_i)] = 0 \quad (5.3.33)$$

$$\alpha_i^* [\epsilon + \xi_i^* - y_i + f(x_i)] = 0 \quad (5.3.34)$$

$$\xi_i \gamma_i = 0 \quad (5.3.35)$$

$$\xi_i^* \gamma_i^* = 0 \quad (5.3.36)$$

由式(5.3.35)和式(5.3.36)知, $\alpha_i \alpha_i^* = 0$, 即任何一组 α_i 和 α_i^* 都不会同时为 0。由式(5.3.26)、式(5.3.29)、式(5.3.35)和式(5.3.36), 得

$$(C - \alpha_i) \xi_i = 0 \quad (5.3.37)$$

$$(C - \alpha_i^*) \xi_i^* = 0 \quad (5.3.38)$$

综上可得支持向量机回归的步骤, 下面具体介绍。

步骤 1: 给定训练集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \in (\mathbf{R}^n \times Y^n)$, 其中 $x_i \in \mathbf{R}^n$, $y_i \in \mathbf{R}$, $i=1, 2, \dots, N$;

步骤 2: 选择适当的参数 $\epsilon, \epsilon > 0$;

步骤 3: 构造凸二次规划寻优问题并求解, 解得向量 $\bar{\alpha} = (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_N)$ 和 $\bar{\alpha}^* = (\bar{\alpha}_1^*, \bar{\alpha}_2^*, \dots, \bar{\alpha}_N^*)$;

步骤 4: 计算 $\bar{\alpha} = \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) x_i$ 及 $\bar{b} = y_j - (\bar{\alpha} \cdot x_j) + \epsilon$;

步骤 5: 构造决策函数, 即训练集合的回归函数

$$y = f(\mathbf{x}) = (\bar{\alpha} \cdot \mathbf{x}) + \bar{b} = \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) (x_i \cdot \mathbf{x}) + \bar{b}$$

5.3.3 基于空间分集支持向量机的多小波神经网络盲均衡算法

1. 基于支持向量机多小波神经网络的盲均衡算法

针对多小波神经网络的非线性盲均衡算法缺乏全局搜索能力, 本节利用支持向量机来搜索多小波神经网络输入层与隐含层之间的最优初始连接权向量, 这就得到了基于支持向量机多小波神经网络的非线性盲均衡算法(Support Vector Machine Multi-Wavelet Transform Feedback Neural Network Nonlinear blind equalization algorithm, SVM-MWT-FNN), 原理^[29]如图 5.20 所示。

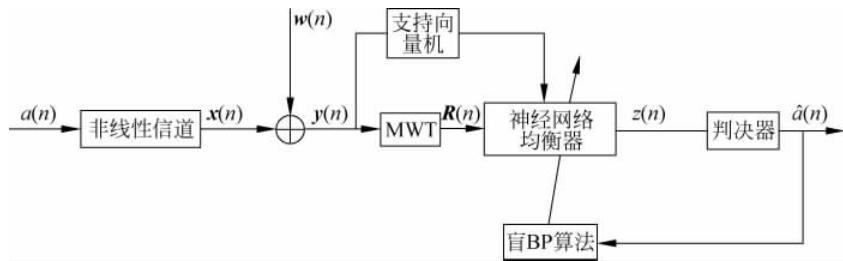


图 5.20 基于支持向量机多小波神经网络的盲均衡算法原理

图中, $a(n)$ 是发射信号; $x(n)$ 是经过非线性信道 Wiener 模型后的信号; $w(n)$ 是加性高斯白噪声; $R(n)$ 是经过多小波变换后的输入信号; $z(n)$ 为均衡器输出信号; $\hat{a}(n)$ 是 $a(n)$ 的判决输出信号。

当发射信号 $a(n)$ 为常模信号时, 对均衡器接收信号 $y(n)$ 的前 N 组向量, 利用支持向量机进行均衡。根据发射信号的统计特性以及结构风险最小化原则, 以精度 ϵ 估计多小波神经网络(MWT-FNN)输入层与隐含层之间初始连接权向量 $f_{ij}(n)$, 则 SVM 回归问题可描述为^[30-34]

$$\min E_{\text{svm}}(f_{ij}(n)) = \frac{1}{2} \| f_{ij}(n) \|^2 \quad (5.3.39)$$

约束条件为

$$\begin{cases} R^2 - ([f_{ij}(n)]^T y(n))^2 \leq \epsilon \\ ([f_{ij}(n)]^T y(n))^2 - R \leq \epsilon \end{cases} \quad (5.3.40)$$

式中, $R^2 = E\{|a(n)|^4\}/E\{|a(n)|^2\}$, 而

$$\epsilon = 3\sigma_n^2 \frac{\ln N}{N} \quad (5.3.41)$$

式中, σ_n^2 为噪声方差, 参数 ϵ 决定了不敏感区域的宽度和支持向量的个数。

为了放松约束, 引入松弛变量 $\xi(n)$ 、 $\tilde{\xi}(n)$ 和惩罚系数 C , 式(5.3.39)和式(5.3.40)的最优化问题就可以转化为求解约束最优化的问题, 即

$$\min E_{\text{svm}}(f_{ij}(n)) = \frac{1}{2} \| f_{ij}(n) \|^2 + C \sum_{n=1}^N (\xi(n) + \tilde{\xi}(n)) \quad (5.3.42)$$

约束条件为

$$\begin{cases} R^2 - ([f_{ij}(n)]^T y(n))^2 \leq \epsilon + \xi(n) \\ ([f_{ij}(n)]^T y(n))^2 - R^2 \leq \epsilon + \tilde{\xi}(n) \\ \xi(n), \tilde{\xi}(n) \geq 0 \end{cases} \quad (5.3.43)$$

式(5.3.42)和式(5.3.43)中, $\xi(n)$ 和 $\tilde{\xi}(n)$ 用来衡量样本离群的距离, 惩罚

变量 C 则体现了对离群样本点的重视程度,且

$$C = \bar{g}_n + 3\sigma_\epsilon \quad (5.3.44)$$

式中, $\bar{g}_n = |\mathbf{y}(n)|^2$, \bar{g}_k 表示均值, σ_ϵ 表示接收信号的方差。

在以上构造支持向量机多小波神经网络非线性盲均衡算法的原始优化问题中,约束条件中均衡器权向量 \mathbf{f}_{ij} 是二次方的,支持向量机所采用的线性规划算法无法解决二次方原始最优化问题,于是,需采用一种迭代权二次规划算法来解决这一问题。设神经网络的隐含层输入为 $\mathbf{u}_{j,\text{svm}}^J(n)$,在求出 $\mathbf{u}_{j,\text{svm}}^J(n)$ 后,将式(5.3.43)中的二次约束改为线性约束,即

$$\begin{cases} R - (\mathbf{f}_{ij,\text{svm}}^J(n))^T \mathbf{x}(n) \mathbf{u}_{j,\text{svm}}^J(n) \leq \varepsilon + \xi(n) \\ (\mathbf{f}_{ij,\text{svm}}^J(n))^T \mathbf{x}(n) \mathbf{u}_{j,\text{svm}}^J(n) - R \leq \varepsilon + \tilde{\xi}(n) \end{cases} \quad (5.3.45)$$

将式(5.3.42)~式(5.3.45)的原始最优化问题转化为凸二次规划问题后,得

$$\begin{aligned} \max E'_{\text{svm}} = & -\frac{1}{2} \sum_{i,n=1}^N (\tilde{\alpha}(n) - \alpha(n))(\tilde{\alpha}(i) - \alpha(i)) \\ & (\mathbf{u}_{j,\text{svm}}^J(n) \mathbf{u}_{j,\text{svm}}^J(i)) < \mathbf{x}(n), \mathbf{x}(i) > - \\ & \varepsilon \sum_{n=1}^N (\tilde{\alpha}(n) + \alpha(n)) + \sum_{n=1}^N (\tilde{\alpha}(n) - \alpha(n)) \end{aligned} \quad (5.3.46)$$

式中, $\tilde{\alpha}(n)$ 和 $\alpha(n)$ 为 Lagrange 乘子。

约束条件为

$$\begin{cases} \sum_{n=1}^N (\tilde{\alpha}(n) - \alpha(n)) = 0 \\ 0 \leq \tilde{\alpha}(n), \alpha(n) \leq C \end{cases} \quad (5.3.47)$$

式中, $< \mathbf{x}(n), \mathbf{x}(i) >$ 表示内积运算, $n=1, 2, \dots, N$ 。

比较原始问题与对偶问题,神经网络均衡器输入层与隐含层连接权值的初始权向量可以表示为

$$\mathbf{f}_{j,\text{svm}}^J(n) = \sum_{n=1}^N (\tilde{\alpha}(n) - \alpha(n)) \mathbf{u}_{j,\text{svm}}^J(n) \mathbf{x}(n) \quad (5.3.48)$$

通过上述步骤,可以计算神经网络输入层与隐含层间初始连接权向量 $\mathbf{f}_{j,\text{svm}}^J(n)$,再循环迭代至满足切换条件为止。 $\mathbf{f}_{j,\text{svm}}^J(n)$ 的更新公式为

$$\mathbf{f}_{j,\text{svm}}^J(n) = \lambda \mathbf{f}_{j,\text{svm}}^J(n-1) + (1-\lambda) \mathbf{f}_{j,\text{svm}}^J(n-2) \quad (5.3.49)$$

式中, λ 表示遗漏因子。

当切换条件

$$\begin{cases} \text{MSE}(n) = \frac{1}{N} \sum_{n=1}^N (|\mathbf{u}_{j,\text{svm}}^J(n)|^2 - R) \\ |\text{MSE}(n) - \text{MSE}(n-1)| \leq \eta \end{cases} \quad (5.3.50)$$

满足时,可得输入层与隐含层间连接权向量 $\mathbf{f}_{j,\text{svm}}^J(n)$ 的初始向量, η 为切换阈值。

2. 基于空间分集支持向量机和多小波神经网络的非线性盲均衡算法

分集技术的基本原理是把接收机获得的多个统计独立、携带统一信息的衰落信道合并以降低衰落的影响。因为该技术对各径信号进行跟踪,及时对更多的信号分量进行处理,能有效地提高通信的可靠性,因此,被广泛用于移动通信中。根据分集资源不同,分集技术可分为:时间分集、空间分集、角度分集等。本节引入空间分集^[35,36],也称为天线分集,它使用多个发射或接收天线。要获得满意的分集效果,接收端各天线间距离 d 需满足 $d \geq 3\lambda$, λ 为工作频率的波长。通常分集天线数越多,消除衰落的效果越明显。但当分集重数增加到一定量时,性能改善效果明显减弱。因此,分集重数以 2~4 为宜。由于分集支路的信号衰落互相独立,而且同时衰落的概率极低,所以将空间分集应用于盲均衡系统中,能有效消除信道衰落和码间干扰。最早由 Monsen 提出将空间分集与均衡器相结合(Spatial Diversity Equalizer, SDE)。空间分集至少需要两副相距一定间隔的天线,分集结构如图 5.21 所示。

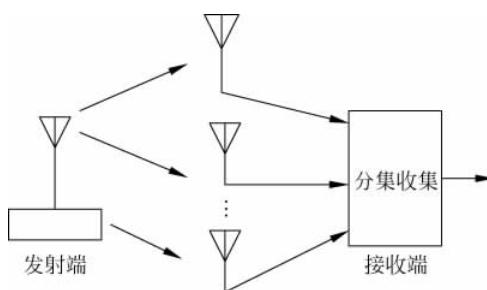


图 5.21 空间分集示意图

将空间分集技术应用于支持向量机多小波神经网络的非线性盲均衡算法时,每一分集支路为支持向量机多小波神经网络盲均衡器^[37~40],对所有支路均衡器的输出进行加权合并,就得到基于空间分集支持向量机多小波神经网路的非线性盲均衡算法(Spatial Diversity Equalizer Support Vector Machine Multi-Wavelet Transform Feedback Neural Network

blind equalization algorithm, SDE-SVM-MWT-FNN)。其原理如图 5.22 所示。

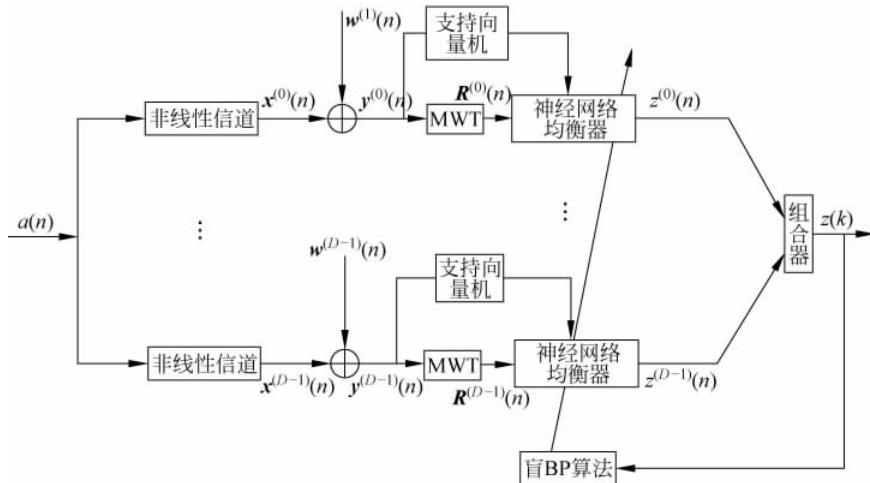


图 5.22 SDE-SVM-MWT-FNN 盲均衡算法原理

根据支持向量机优化小波神经网络均衡器原理, 第 d ($d=0, 1, \dots, D-1$) 路均衡器的输入输出信号为

$$\mathbf{R}^{(d)}(n) = \mathbf{V}_{\text{MWT}} \mathbf{y}^{(d)}(n) \quad (5.3.51)$$

$$\mathbf{u}_j^{J(d)}(n) = \sum_{i=1}^I \mathbf{f}_{ij, \text{svm}}^{(d)}(n) \mathbf{R}^{(d)}(n) \quad (5.3.52)$$

$$\mathbf{v}_j^{J(d)}(n) = F \left(\sum_{i=1}^I \mathbf{f}_{ij, \text{svm}}^{(d)}(n) \mathbf{R}^{(d)}(n) \right) \quad (5.3.53)$$

$$z^{(d)}(n) = F \left(\sum_{j=1}^J \mathbf{f}_{jp}^{(d)}(n) \mathbf{v}_j^{J(d)}(n) \right) \quad (5.3.54)$$

式中, $\mathbf{f}_{ij, \text{svm}}^{(d)}(n)$ 为第 d 个分集支路神经网络均衡器的输入层到隐含层的连接权系数, $\mathbf{f}_{jp}^{(d)}(n)$ 为第 d 个分集支路神经网络均衡器隐含层与输出层的连接权系数。

信号经过组合器加权合并后, 输出信号为

$$z(n) = \sum_{d=0}^D g_d z_d(n) \quad (5.3.55)$$

式中, g_d 为加权系数。若采用最大比值合并, 则 $g_d = A_d / \sigma_d^2$ 。其中, A_d 为第 d 条分集支路的信号幅度, σ_d^2 为第 d 条支路的噪声功率; 若采用等增益合并, 则 $g_d = 1$ 。

3. 计算复杂度

在盲均衡算法结构中增加支持向量机、 $D-1$ 个支路以及一个加权组合器,会在一定程度上增加算法的计算复杂度。根据文献[30],设 n 为样本训练种数,对于一个测试样本,支持向量机的复杂度是 $O(n^3)$,而 MWT-FNN 的计算复杂度为 $O(MN)$,则由式(5.3.51)至式(5.3.55)知:SDE-SVM-MWT-FNN 的计算复杂度为 $O(Dn^3 + DMN + D) = O(Dn^3)$,与支持向量机的复杂度是同一个数量级。

5.3.4 仿真验证

为了检验 SDE-SVM-MWT-FNN 算法的性能,现以 FNN、SVM-FNN 和 SVM-MWT-FNN 作为比较对象,进行仿真实验。实验时,图 5.22 中 $d=2$,即空间分集重数为 2。SDE-SVM-MWT-FNN 采用两种非线性信道混合 Wiener 模型。

信道 1:

$$\begin{aligned} a_2^{(1)}(n) &= (1.1026 + j0.1015)a(n) + \\ &\quad (0.2057 + j0.0330)a(n-1) + (0.1012 + j0.0160)a(n-2) \\ x^{(1)}(n) &= (1.5050 - j0.0118)a_2^{(1)}(n) + \\ &\quad (0.0010 - j0.0004)(a_2^{(1)}(n))^2 + (-0.0011 + j0.0001)(a_2^{(1)}(n))^3 \end{aligned}$$

信道 2:

$$\begin{aligned} a_2^{(2)}(n) &= (1.22 + j0.646)a(n) + (0.063 - j0.001)a(n-1) + \\ &\quad (-0.024 - j0.014)a(n-2) \\ x^{(2)}(n) &= (0.039 - j0.022)a_2^{(2)}(n) + (0.035 - j0.035)(a_2^{(2)}(n))^2 + \\ &\quad (-0.01 - j0.017)(a_2^{(2)}(n))^3 \end{aligned}$$

FNN、SVM-FNN 和 SVM-MWT-FNN 采用信道 1 进行仿真。仿真中,发射信号为 8PSK,信噪比为 20dB,采用三层多小波神经网络,均衡器输入层权长 16,均衡器中心抽头初始化;对 N-FNN,步长 $\mu_1 = 1.2 \times 10^{-5}$;对 SVM-N-FNN,步长 $\mu_2 = 0.0003$;对 SVM-MWT-FNN,步长 $\mu_3 = 0.0003$ 。对信道 1,在 SDE-SVM-MWT-FNN 中,步长 $\mu_4^{(1)} = 0.0012$;对信道 2,步长 $\mu_4^{(2)} = 0.0003$;采用 DB2 多小波;支持向量机盲均衡算法的训练样本数 $N=100$, $\lambda=0.9$,切换阈值 $\eta=10^{-4}$ 。500 次蒙特卡罗仿真结果,如图 5.23 所示。

图 5.23(a)表明,在收敛速度上,SDE-SVM-MWT-FNN 比 SVM-

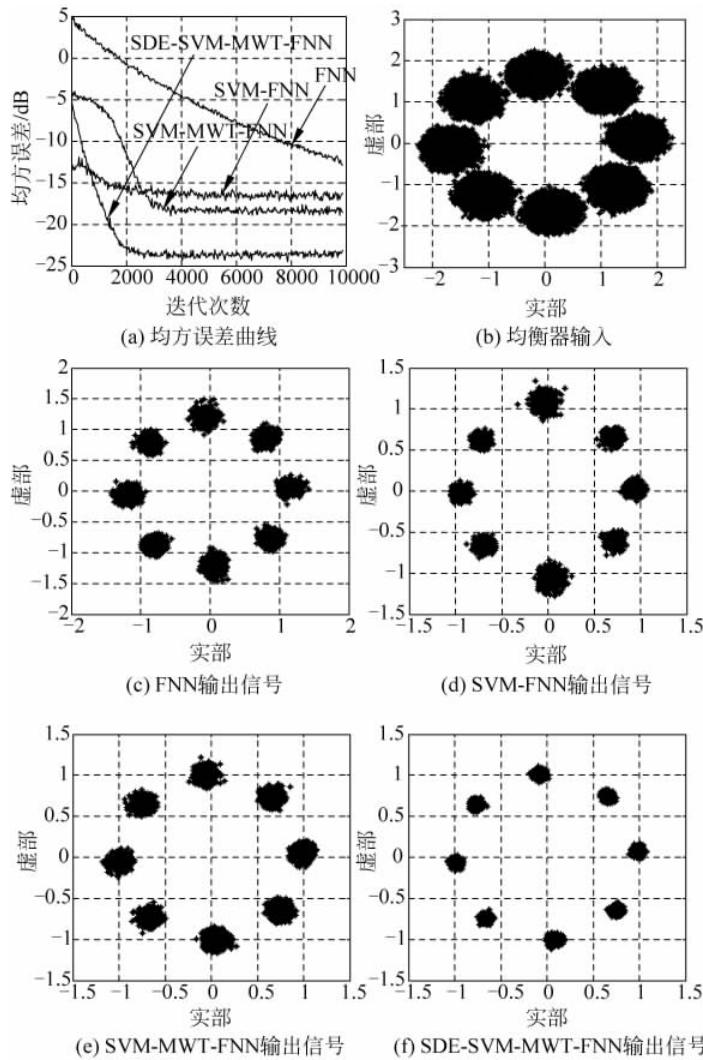


图 5.23 仿真结果

MWT-FNN 快了近 1500 步, 比 SVM-FNN 快了近 2500 步, 比 FNN 快了近 7500 步; 在稳态误差上, SDE-SVM-MWT-FNN 比 SVM-MWT-FNN 减小约 5dB, 比 SVM-FNN 减小约 8dB, 比 FNN 减小了约 13dB。图 5.23(b)~(f)表明, SDE-SVM-MWT-FNN 的星座图比 SVM-MWT-FNN、SVM-FNN 及 FNN 更加清晰、紧凑, 码间干扰明显减少。

5.4 基于混沌算法优化的双神经网络盲均衡算法

混沌是指现实世界中一种貌似无规则的复杂运动形态,是非线性动力学系统在一定条件下表现出的一种运动状态。混沌具有独特的性质:随机性、遍历性和规律性。空间稳定时,混沌运动轨迹可以遍历空间内每一个状态点,不重复、不紊乱。这一特点在混沌搜索过程中可有效避免局部极小值。

5.4.1 混沌算法基础

1. 混沌理论

混沌是一种紊乱的自然现象,是系统处于非平衡过程中所呈现的随机行为。混沌信号有如下特性:

1) 初值敏感依赖性

混沌的初值敏感依赖性又称为蝴蝶效应,如果初始值出现微小的差异,则后续系统会出现很大变化,因而,混沌系统的长期变化轨迹是不可预测的。该特性可以用混沌系统的 Logistic 映射解释。Logistic 映射系统可以表示为

$$x(n+1) = \mu x(n)(1 - x(n)), \quad x(n) \in [0,1], \mu \in [0,4] \quad (5.4.1)$$

式中,当参数 $\mu \in [3.7, 4]$ 时,该映射是混沌的,如图 5.24 所示。

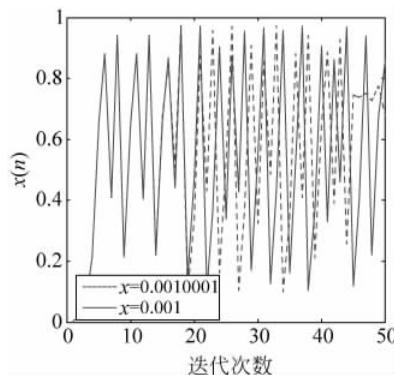


图 5.24 混沌的初值敏感特性

2) 伸长和折叠特性

混沌具有伸长和折叠特性。系统内部不稳定时点与点之间的距离会被扩大,即为伸长特性;但整个系统是稳定的,稳定的系统会限制点与点之间的距离,称之为折叠特性。系统反复的伸长与折叠产生了混沌。

3) 分形性和自相似性

混沌中各种奇怪的吸引子都具有分形结构,由分维数描述其特征。同时,混沌中的分形具有自相似性,这种自相似性可以是严格的,也可以是传统意义上的。分形和混沌是同一种规律的不同表现,在空间上表现为分形,在时间上表现为混沌。

4) 有界性和内随机性

混沌具有内在随机性,源于系统对初始值的敏感依赖性。系统无须附加任何随机因素,完全由确定性方程描述,但系统会表现出类似随机性的行为。图 5.24 所示的运动轨迹一直保持在 $[0,1]$ 内,所以从整体上来说混沌系统又是稳定有界的。

2. 混沌算法

混沌算法^[36,37]主要采用载波变换方式,以混沌变量作为载体,通过线性映射方式将其替换到优化变量区域,搜索优化变量替换为更新混沌序列。混沌变量在搜索时对空间所有状态逐一遍历,且所有经历过的状态不会被重复搜索,大大节省了搜索时间。

Logistic 模型是混沌动力学模型的一个特例,其动力学特征方程简单且研究成熟,因此经常被应用于混沌优化方法中。利用该模型产生混沌变量 $x(n)$ 后,对混沌变量进行转换

$$y(n) = a + x(n) \times (b - a) \quad (5.4.2)$$

式中, b, a 是优化变量 $y(n)$ 取值区间的上下限。在整个优化过程中, $x(n)$ 始终保持在 $(0,1)$ 内搜索更新值, 映射得到对应的 $y(n)$ 值, 与 $x(n)$ 同时遍历整个优化变量空间, 经过 n 次的遍历寻优, 总能搜索到问题的最优解。

5.4.2 混沌优化过程

与其他优化算法相比,在代价函数方面,混沌优化算法不需要很多限制,只需要重复求取目标函数值然后比较更新,简化了中间操作环节,提高了效率;而且根据混沌搜索遍历性的特点,避免了搜索过程陷入局部最优解问题。本节采用文献[41-44]中的混沌优化方式,在混沌变量扰动权向量的当前点,引入时变参数,使扰动的幅度随时变参数的更新逐渐降低,解

决了经典混沌算法的初值敏感性问题,达到最优解附近后再局部细搜索,从而得到所需全局最优权向量。

在混沌优化过程中,类似载波被引入混沌变量进行搜索,在二次载波过程中引入时变参数 $\Psi(t)$,并将 $f(K) = f_{\text{opt}} + \Psi(t)(C(K) - 0.5)$ 引入到优化变量中。其中, $f(K)$ 为优化变量, K, t 为迭代次数, f_{opt} 表示当前最优变量, $C(K) = \{C_1(K), C_2(K), \dots, C_n(K)\}$ 是混沌映射产生的混沌变量, n 为优化变量的维数, $C_i(K)$ 为 $C(K)$ 的第 i 维值。设时变参数的迭代公式为 $\Psi(t+1) = \lambda \Psi(t)$ 。其中, λ 是时变参数的衰减因子, $\lambda \in [0.95, 0.999]$ 。 $\Psi(t)$ 的初始值定义为 $\Psi(0) = (J_{\text{opt}} - J_{\text{max}}) / \ln P$, 其中, $J_{\text{opt}}, J_{\text{max}}$ 为混沌一阶段搜索得到的所有可行解中, 对应于代价函数的最小值和最大值, 参数 P 的选择范围为 $(0, 1)$ 。

本节将混沌优化求解得到的代价函数最小值 J_{opt} 作为神经网络输入层优化权系数。在产生混沌变量时, 使用 Logistic 映射产生混沌信号的映射公式为

$$C(K+1) = \rho C(K)[1 - C(K)] \quad (5.4.3)$$

式中, $\rho \in [3.57, 4]$ 。

设待寻优对象问题的代价函数 $J_{\text{opt}} = J(f_{\text{opt}}) = J_{\min}(f)$, J 代表目标函数, $f \in [d, e]$, f 表示优化向量。设混沌一阶段搜索迭代次数为 N_1 , 二阶段迭代搜索次数为 N_2 , 则混沌优化神经网络均衡器输入层权向量的求解步骤如下:

步骤 1: 取 n 个 $[0, 1]$ 区间中相异值作为初始值 $C(0)$ (其中, 不能取 0、0.25、0.5、0.75、1), 设定权向量最优解 f_{opt} 对应的代价函数值 J_{opt} 及代价函数最大值 J_{max} 的初值, 令 $K=0, K'=0, t=0$ 。

步骤 2: 通过式(5.4.3)迭代得到 n 个混沌变量, 各个变量轨迹不同, 并通过

$$f(K) = s + rC(K) \quad (5.4.4)$$

改变优化变量的取值范围。式中, $s=d, r=e-d$ 。

步骤 3: 混沌变量迭代后, 计算 $J(f(K))$, 并保存 J_{max} 和 J_{opt} 的值。

如果 $J(f(K)) \leq J_{\text{opt}}$, 那么 $J_{\text{opt}} = J(f(K))$, $f_{\text{opt}} = f(K)$; 如果 $J(f(K)) > J_{\text{max}}$, 那么 $J_{\text{max}} = J(f(K))$, $K=K+1$, 如果 $K \leq N_1$, 则转步骤 2。

步骤 4: 经过步骤 3 的 N_1 步搜索得到 J_{opt} , 选择参数 P , 计算时变参数 $\Psi(t)$ 的初始值 $\Psi(0) = (J_{\text{opt}} - J_{\text{max}}) / \ln P$, 并按

$$f(K') = f_{\text{opt}} + \Psi(t)(C(K') - 0.5) \quad (5.4.5)$$

进行二次载波。

步骤 5: 混沌优化变量经过二次载波后进行细搜索, 并计算 $J(f(K'))$ 。

如果 $J(f(K')) \leq J_{\text{opt}}$, 那么 $J_{\text{opt}} = J(f(K')), f_{\text{opt}} = f(K')$; 否则放弃 $f(K')$ 。 $K' = K' + 1, \Psi(t+1) = \lambda \Psi(t), t = t + 1$ 。

步骤 6: 如果 $K' > N_2$, 则停止搜索, 输出 J_{opt} 及其对应的全局最优解 f_{opt} ; 反之, 则返回步骤 5。

综上, 步骤 1~6 即为混沌算法优化过程。本节将混沌优化算法用于优化神经网络输入层与隐含层之间的权向量, 最后输出的 f_{opt} 即为混沌优化多小波神经网络输入层与隐含层间权向量的初始化值, 其流程如图 5.25 所示。

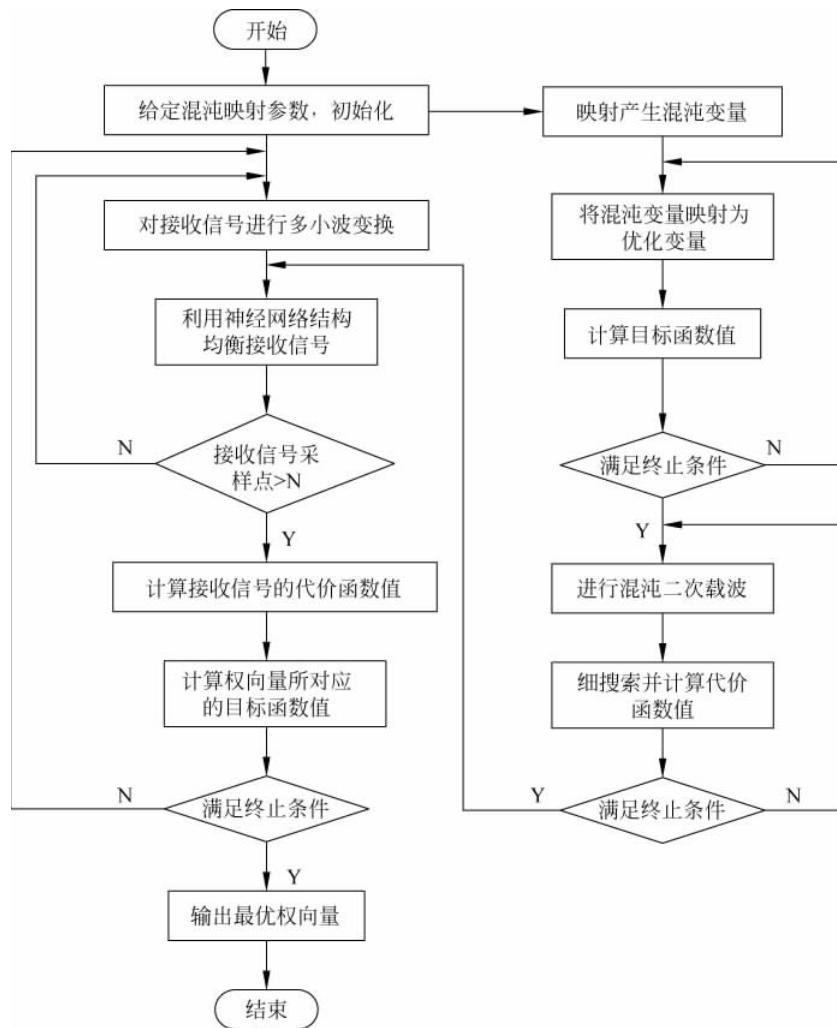


图 5.25 混沌算法优化多小波神经网络盲均衡算法流程

5.4.3 基于混沌算法优化多小波双神经网络的非线性盲均衡算法

神经网络盲均衡算法虽然结构简单,但它的非线性处理能力并不强。本节提出了双神经网络盲均衡算法。

最早的反馈神经网络是由物理学教授 Hopfield 在 1982 年提出来的,他在神经网络中引入“能量函数”的概念,给出了网络稳定的判据,开拓了神经网络在信号处理方面的新领域。本节针对非线性信道均衡问题,将神经网络与判决反馈神经网络相结合,构成了双神经网络盲均衡算法结构,如图 5.26 所示。

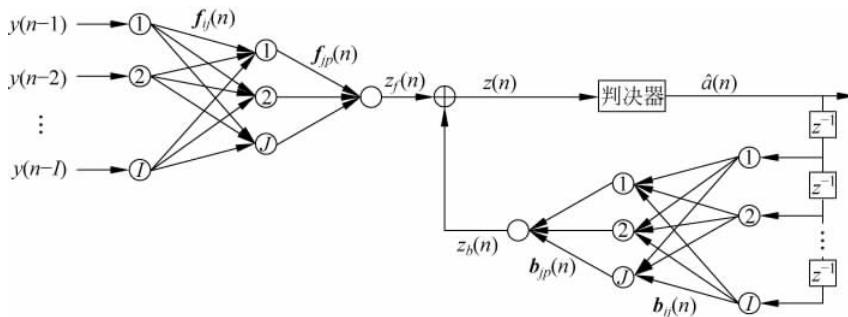


图 5.26 双神经网络盲均衡算法结构

图 5.26 表明,双神经网络盲均衡算法的前置滤波器和反馈滤波器均为一个神经网络滤波器。判决反馈结构的引入以及神经元的增多,增强了盲均衡算法的非线性处理能力。 $f_{ij}(n)$ 为前馈神经网络输入层与隐含层的连接权值, $f_{jp}(n)$ 为前馈神经网络隐含层到输出层连接权值, $z_f(n)$ 为前馈神经网络输出, $b_{ij}(n)$ 为反馈神经网络输入层到隐含层的连接权值, $b_{jp}(n)$ 为前馈神经网络隐含层与输出层的连接权值, $z_b(n)$ 为前馈神经网络输出。

前置滤波器的输出为

$$z_f(n) = F \left(\sum_{j=1}^J f_{jp}(n) F \left(\sum_{i=0}^I f_{ij}(n) \mathbf{y}(n-i) \right) \right) \quad (5.4.6)$$

反馈滤波器的输出为

$$z_b(n) = F \left(\sum_{j=1}^J b_{jp}(n) F \left(\sum_{i=0}^I b_{ij}(n) \hat{a}(n-i) \right) \right) \quad (5.4.7)$$

系统的输出为

$$z(n) = z_f(n) - z_b(n) \quad (5.4.8)$$

前置滤波器输入层到隐含层权向量的迭代公式为

$$\mathbf{f}_{ij}(n+1) = \mathbf{f}_{ij}(n) - 2\mu_{ij}z(n)e(n)F'(u_p^P(n))\mathbf{f}_{jp}(n)F'(u_j^I(n))\mathbf{y}(n) \quad (5.4.9)$$

隐含层到输出层权向量的迭代公式为

$$\mathbf{f}_{jp}(n+1) = \mathbf{f}_{jp}(n) - 2\mu_{jp}z(n)e(n)F'(u_p^P(n))\mathbf{v}_j^I(n) \quad (5.4.10)$$

式中, μ_{ij} 、 μ_{jp} 表示步长。

反馈滤波器输入层到隐含层权向量的迭代公式为

$$\mathbf{b}_{ij}(n+1) = \mathbf{b}_{ij}(n) - 2\mu_{bij}z(n)e(n)F'(m_p^P(n))\mathbf{b}_{jp}(n)F'(O_j^I(n))\hat{a}(n) \quad (5.4.11)$$

隐含层到输出层权向量的迭代公式为

$$\mathbf{b}_{jp}(n+1) = \mathbf{b}_{jp}(n) - 2\mu_{bjp}\hat{a}(n)e(n)F'(m_p^P(n))O_j^I(n) \quad (5.4.12)$$

式中, μ_{bij} 、 μ_{bjp} 表示步长。

$$m_j^I(n) = \sum_{i=0}^I \mathbf{b}_{ij}(n) \hat{a}(n-i) \quad (5.4.13)$$

$$O_j^I(n) = F\left(\sum_{i=1}^I \mathbf{b}_{ij}(n) \hat{a}(n-i)\right) \quad (5.4.14)$$

$$m_p^P(n) = \sum_{i=1}^J \mathbf{b}_{jp}(n) O_j^I(n) \quad (5.4.15)$$

将混沌优化算法和神经网络反馈结构相结合,本节提出了基于混沌算法优化多小波双神经网络的盲均衡算法,原理如图 5.27 所示。

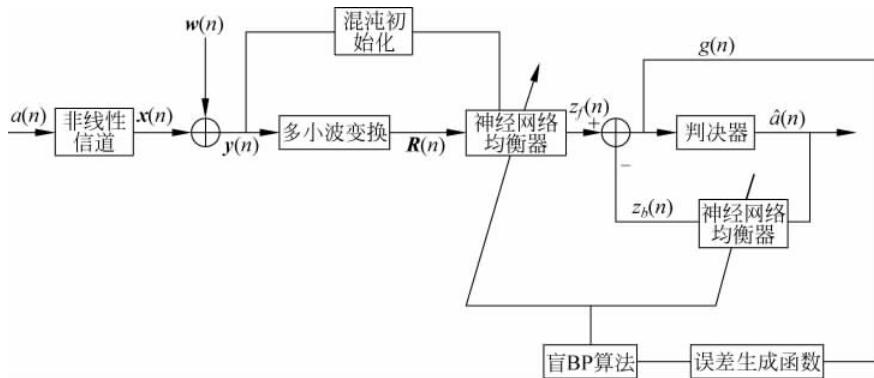


图 5.27 混沌优化多小波双神经网络的盲均衡算法

引入了混沌优化算法以及双神经网络结构后,计算复杂度一定程度上会有所增加。

设输入信号长度为 N ,第一次载波搜索长度为 N_1 ,二次载波搜索长度为 N_2 ,神经网络输入层权向量长度为 L 。由式(5.4.3)产生映射信号的复杂度为 $L^2 + 3L$;根据混沌算法优化神经网络均衡器输入层权向量步骤,计算复杂度为 $L^2 + 3L + 3N_1^2 + 6N_1 + 3N_2^2 + 6N_2$;根据前置滤波器及反馈滤波器均为神经网络滤波器,由式(5.4.6)~式(5.4.15)得计算复杂度为 $2MN$,则 CO-MWT-D-ANN 的计算复杂度为 $L^2 + 3L + 3N_1^2 + 6N_1 + 3N_2^2 + 6N_2 + 2MN$ 。

5.4.4 仿真验证

为了验证 CO-MWT-D-ANN 算法的有效性,以 FNN、MWT-FNN、CO-MWT-FNN 为比较对象。信道模型如图 5.15 所示,输入为 2PAM 信号,信噪比均为 25dB,采样点均为 10000,权向量长度为 16;在 FNN 中, $\mu=0.00001$,第 4 个抽头初始化;在 MWT-FNN 中, $\mu=0.0008$,采用 DB2 多小波,第 4 个抽头初始化;在 CO-MWT-FNN 中, $\mu=0.00002$,采用 DB2 多小波,第 4 个抽头初始化;在 CO-MWT-D-ANN 中, $\mu=\mu_b=0.00001$,采用 DB2 多小波,第 4 个抽头初始化。500 次蒙特卡罗仿真结果如图 5.28 所示。

图 5.28(a)表明,CO-MWT-D-ANN 的收敛速度比 CO-MWT-FNN 快了大约 500 步,比 MWT-FNN 快了约 7500 步,比 FNN 快了约 9000 步;CO-MWT-FNN 的收敛速度比 MWT-FNN 快了约 7000 步,比 FNN 快了约 8500 步。在稳态误差上,CO-MWT-D-ANN 比 FNN 约小 13dB,比 CO-MWT-FNN 约小 6dB; CO-MWT-FNN 比 FNN 约小 7dB。

图 5.28(b~f)表明,CO-MWT-FNN 的输出星座图比 FNN、MWT-FNN 更为清晰、紧凑,CO-MWT-D-ANN 的输出星座图最为清晰、紧凑,信号恢复更加准确。

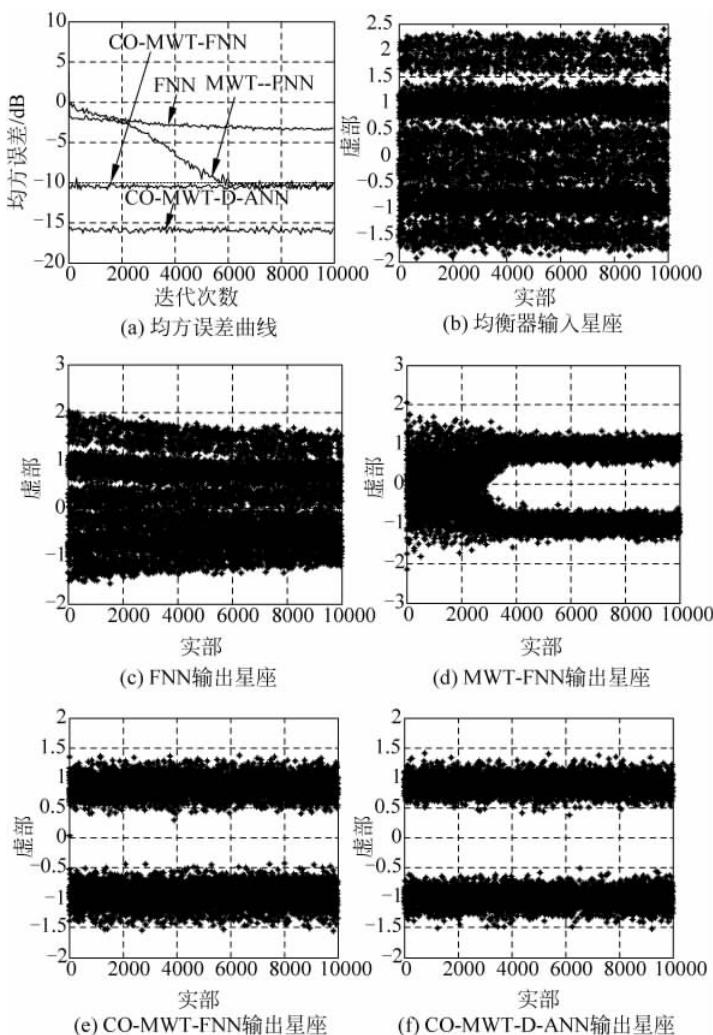


图 5.28 仿真结果

5.5 基于 Volterra 滤波回声状态网络结构和 PCA 的均衡算法

智能信号处理领域的一个关键研究主题是基于动态和实时系统的有效解决方案。由于实际问题的复杂性, 所建模型需能够执行非线性映射,

也需能访问输入和输出信号的时间历程以及内部状态。因此,所建模型中需分有反馈网络结构。而递归神经网络(Recurrent Neural Network, RNN)允许不同层神经元之间的反馈连接,具有记忆性,并通过时间反向传播(BackPropagation-Through-Time, BPTT)^[39]和实时递归学习(Real-Time Recurrent Learning, RTRL)方法实现^[40]。然而,由于反馈递归的存在,代价函数对连接权向量的导数难以计算;网络的不稳定,可能会导致网络结构功能难以有效发挥。为了克服这些不足,Jaeger^[41]最初提出了回声状态网络(Echo State Network, ESN),这是一个强大而有希望的解决方案,这种网络在一定程度上保留了RNN结构的处理潜力,在训练过程中,可通过设置递归内层连接权重(建立所谓的动态库)实现简化,而无须使用期望信号。由于创建动态库会影响网络的表示能力,因此如何确定动态库(或存储层)权系数是ESN设计的一个重要内容。Ozturk^[42]等采用信息论方法,分析了回声状态之间的相关性,提出了一个确定存储层权系数方法;与Jaeger^[43]最初提出的方法相比,Ozturk的方法有利于产生更高程度的多样性;然而,该方法中,输出层仍然为线性组合器结构,既不能完全有效地使用来自存储层输出信号的高阶统计量,又不能保持训练过程尽可能简单,从而在应用上受到限制。为了克服这个限制,文献[44]给出了一种新的ESN结构:线性组合器由Volterra滤波器^[45]替代,这样就将输出层由线性组合器改为非线性组合器。研究表明,该方法中回声状态数量如果增加,则权系数数量会过度增长,从而可能产生“维数灾难”。为了避免维数灾难^[46],在回声状态传递到输出层前,先采用主成分分析法(Principal Component Analysis, PCA)^[52-53]提取回声状态主成分,以减少回声状态数,从而减少需要调整的权向量长度。由于回声状态网络能够实现处理能力和计算复杂度之间的内在折中^[54-55],因此将回声状态网络应用于信道均衡问题对于实时和非实时性来说都是一个有前景且有效的解决方案。

5.5.1 回声状态网络

考虑一个具有 K 个输入单元、 N 个内部单元和 L 个输出单元的离散时间递归神经网络,如图5.29所示。

设 $\mathbf{u}(n)=[u_1(n), u_2(n), \dots, u_K(n)]^T$ 表示输入单元的激活向量,通过线性组合传输到内部神经元;组合器的权重矩阵为 $\mathbf{W}^{\text{in}} \in R^{N \times K}$ 。

内层也称为动态存储层,由完全连接的非线性单元组成,其激活向量

$\mathbf{x}(n) = [x_1(n), x_2(n), \dots, x_K(n)]^T$ 对应于网络状态, 其更新公式为

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{\text{in}} \mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{\text{back}}\mathbf{y}(n)) \quad (5.5.1)$$

式中, $\mathbf{W} \in R^{N \times N}$ 表示存储层内部连接向量, $\mathbf{W}^{\text{back}} \in R^{N \times L}$ 表示输出信号返回到内层单元的相关连接向量, $\mathbf{f}(\cdot) = (f_1(\cdot), f_2(\cdot), \dots, f_N(\cdot))$ 表示内层单元的激活函数。

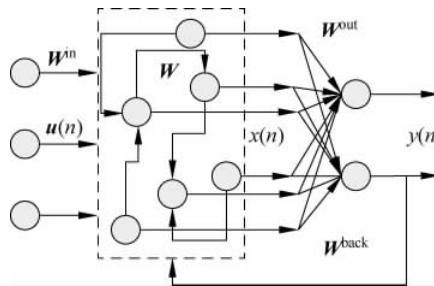


图 5.29 基本的神经网络结构

网络的输出向量 $\mathbf{y}(n) = [y_1(n), y_2(n), \dots, y_L(n)]^T$ 的计算公式为

$$\mathbf{y}(n+1) = \mathbf{f}^{\text{out}}(\mathbf{W}^{\text{out}}\mathbf{x}(n+1)) \quad (5.5.2)$$

式中, $\mathbf{W}^{\text{out}} \in R^{L \times N}$ 表示输出权重矩阵, $\mathbf{f}^{\text{out}}(\cdot) = \{f_1^{\text{out}}(\cdot), f_2^{\text{out}}(\cdot), \dots, f_L^{\text{out}}(\cdot)\}$ 表示内层单元的激活功能。

在 2001 年, Jaeger 研究了图 5.29 所示的 RNN 结构的动态特性。研究表明, 在某些情况下网络状态 $\mathbf{x}(n)$ 渐近地独立于初始条件。换句话说, 如果网络从两个不同的初始状态 $\mathbf{x}(0)$ 和 $\hat{\mathbf{x}}(0)$ 开始, 当输入信号序列相同时, 则得到的网络状态 $\mathbf{x}(n)$ 和 $\hat{\mathbf{x}}(n)$ 序列会收敛到接近于同一个值。当这个属性成立时, 初始状态的影响消失, 而存储层动态性完全依赖于输入信号, 使网络具有回声状态^[47]。

此外, Jaeger 给出了关于回声状态的两个充分条件:

(1) 为了用一个 RNN 表示回声状态属性, 内层权矩阵 \mathbf{W} 的最大奇异值必须小于单位值 ($\sigma_{\max}(\mathbf{W}) < 1$)^[47, 56];

(2) 根据内部权矩阵 \mathbf{W} 的谱半径(对于特征值绝对值的最大值), 给出了建立不存在回声状态网络的条件: 如果 $\|\mathbf{W}\| > 1$, 那么网络没有回声状态。

基于这些观察, Jaeger 给出了简化 RNN 的自适应过程。首先, 构造一个满足 $\|\mathbf{W}\| < 1$ 的权重矩阵 \mathbf{W} 并定义一个任意的输入权重矩阵 \mathbf{W}^{in} , 它不会影响回声状态属性; 其次, 使用线性组合器作为输出层, 这样就将剩余权重的自适应问题转化成求解均方误差极小值问题。因此, 任何线性递归

算法都可以用来执行这样的任务,这就构成了 ESNs^[47]的本质。

ESNs 是传统 RNN 的一个有前景又有趣的替代方案,因为它们能够存取输入和输出信号的时间信息(继承 RNN 特性),由于内部存储层连接权向量是预定义的,因此训练过程简单。与递归神经网络需调整整个权向量相比,这种固定权向量为训练算法提供了灵活性^[57-59]。

需要注意的是,动态库完全由权重矩阵 \mathbf{W} 确定,为了允许动态库的输出足以接近期望信号,权重矩阵 \mathbf{W} 选取应尽可能丰富。但是,按照回声状态方法,动态库设计没有关于应用特性的信息,与具有可调权重矩阵 \mathbf{W} 的类似结构相比,在某种意义上,降低了网络的表示能力。

最近,Ozturk 等将平均状态熵(Average State Entropy, ASE)作为 ESN 动态库丰富程度的量度^[48]。此外,他们观察到回声状态的多样性在一定程度上反映了 \mathbf{W} 的特征值结构。该研究表明:为了使 ASE 最大化,由 ESN 推导的线性系统极点是均匀分布的,而且极点在单位圆内,这就产生了 Kautz 滤波器的概念框架。用 ASE-ESN 表示平均状态熵-回声状态网络,其中,ESN 对应于由 Jaeger 提出的原始网络。

5.5.2 平均状态熵-回声状态网络

ESN 可以通过性能和简单性之间的平衡来解释。一方面,ESN 学习方法不能实现所有潜在的来自底层的处理结构,因为动态库内递归连接不能通过误差信号进行自适应调整。另一方面,ESN 不采用自适应性方法,就避免了经典 RNN 学习算法固有的困难,这是其一个显著的特点。实现性能和简单性的平衡的核心在于回声状态动态库的设计。由于没有可利用的误差信号,就需要产生尽可能高的动态多样性,以便更好地提供输出层的相关信息,再借助参考信号对输出层进行自适应调整。然而,在现有的回声状态网络中,输出层对应一个线性组合器,并不能利用来自动态库神经元非线性动态信息的高阶统计量。

为了避免这个问题,这里采用非线性输出层。然而,为了保持 ESN 固有的简单性,保证输出层输出与网络权系数的线性关系是至关重要的。在这种情况下且在最小二乘(或维纳)意义下,就可能获得闭式解^[54]。为此,输出层采用 Volterra 滤波器结构^[49]。此时,网络的输出由多项式项线性组合为

$$y_i(n) = h_0 + \sum_{i=1}^N h_1(i)x_i(n) + \sum_{i=1}^N \sum_{j=1}^N h_2(i,j)x_i(n)x_j(n) +$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N h_3(i, j, k) x_i(n) x_j(n) x_k(n) + \dots \quad (5.5.3)$$

式中, $x_k(n)$ 表示 n 时刻第 k 个回声状态, $y_i(n)$ 是第 i 个网络的输出。显然, 系数 $h_i(\cdot)$ 与输出是线性相关的。

然而, Volterra 结构带来的问题是: 随着回声状态(N)数量的增加, 内核的数量往往会急剧增加。事实上, 如果观察一些非模糊核数量 N_{ker} 的表达式, 该结论会更加明显。

$$N_{\text{ker}} = 1 + N + \frac{N(N+1)}{2} + \frac{N(N+1)(N+2)}{6} + \dots \quad (5.5.4)$$

该式表明, 随着 N 的增加, N_{ker} 往往会严重增加, 这就影响了 ESN 结构的实际应用。

为了缓解这个问题, 可采用主成分分析法(PCA)来减少传输到输出层的有效状态数量。通过主成分分析法从输出层有效状态中提取主成分, 从而减少需要调整的权向量长度, 而没有造成信息的大量丢失。

设 $\mathbf{R}_{T_s} \in \mathbb{R}^{N \times T_s}$ 是具有 T_s 个输入样本的内部单元的激活矩阵。网络状态的样本协方差矩阵为

$$\hat{\mathbf{C}} = \frac{\mathbf{R}_{T_s} \mathbf{R}_{T_s}^T}{T_s} \quad (5.5.5)$$

式中, $\mathbf{V} \in \mathbb{C}^{N \times N_{\text{pc}}}$ 是由 $\hat{\mathbf{C}}$ 的最大特征值 N_{pc} 对应的特征向量构成的矩阵。因此, 主成分为^[53]

$$\mathbf{s}_i = \mathbf{V}_i^T \mathbf{R}_{T_s} \quad (5.5.6)$$

式中, $i = 1, 2, \dots, N_{\text{pc}}$ 。

由于回声状态之间有不可忽视的冗余存在, 因此可以用 PCA 进行分析。事实上, 文献[48]表明, 回声状态之间存在强烈的相关性, 这意味着保留回声状态 $x(n)$ 中的主成分是可能的。此外, 采用 PCA 能减少后续操作的计算复杂性并降低噪声的影响。

总之, 使用 PCA 和 Volterra 滤波器既能实现压缩和减少自适应权向量长度的数量, 又能有效利用回声状态信息。

5.5.3 信道均衡原理

信道均衡的目的是使信号传输到接收机时失真最小。信道可视为滤波器, 其输入是发射信号 $a(n)$, 输出信号是 $r(n)$, 如图 5.30 所示。

图中, 信道可以是线性的, 也可以是非线性的, 具有不同程度的随机



图 5.30 通信系统模型

性,信道特性决定了在其中传输信号的失真程度。减少失真的一个行之有效的方法是在接收机上采用专门定制的滤波器或均衡器,其作用必须与信道所起的作用相反。在理想情况下,它的输出应该为发射信号的延时信号,即

$$z(n) = ka(n-d) \quad (5.5.7)$$

式中, $z(n)$ 是均衡器输出, k 是一个比例常数, d 表示延时。这种均衡器是迫零的^[54]。

均衡器的设计内容有:

- (1) 一个合适的滤波结构;
- (2) 一个有效的均衡准则;
- (3) 一种有效的优化方法(用来调整滤波器的参数)。

第一个选择是至关重要的。它的设计对均衡效果有直接影响。信道的线性或非线性决定了信道的记忆性影响,信道噪声是一个固有的随机因素。这些因素所造成的影响必须通过滤波结构来进行处理。在某些情况下,线性信道可用线性均衡器来补偿,非线性结构可以解决某些线性信道存在的问题^[55]。无论是线性还是非线性信道,其记忆性对均衡器的记忆性都有很深的影响,在这种情况下需采用递归结构。信道噪声可能需要使用非线性结构描述,信道噪声具有显著的非高斯特征或递归结构^[60]。

第二种选择基本上是对均衡的目标进行转换,即将式(5.5.7)的目标转化为能求解的准则。经典的选择是采用均方误差(Mean Square Error, MSE)标准,也称之为维纳准则。代价函数定义为

$$J_{\text{MSE}} = E\{[a(n-d) - z(n)]^2\} \quad (5.5.8)$$

式中, $E\{\cdot\}$ 表示统计期望。在 MSE 准则下,如果滤波结构相对于权系数是前馈和线性的,那么代价函数具有单个极值点,可以通过直接最小二乘法或迭代 LMS 和 RLS 算法获得全局最优,也就是维纳解^[54]。另一方面,如果滤波结构对于权系数是非线性或递归的,代价函数可能具有多个局部极值点,也就是代价函数具有多样性,这就是一个更复杂的搜索问题。

从前面分析知,从严格的结构观点看,最理想的均衡装置应该是一个非线性的递归结构,因为它能够以最一般的方式满足实际需求;从优化的角度看,找到最优的 MSE 解决方案可能是困难的,这是由于代价函数的多

样性造成的。

5.5.4 仿真验证

下面将研究基于回声状态网络的信道均衡算法的训练和测试方法,以及不同场景下算法的性能,所有实验都是在 MATLAB 环境下进行的。

1. 方法

在仿真中,源信号 $a(n)$ 采用等概率(BPSK 调制)值 $\{+1, -1\}$ 。每个回声状态网络的目的是在同一时刻使用接收信号 $r(n)$ 估计原始信号 $a(n)$,也就是说,没有均衡延迟,输入和输出的数量总是等于 1 ($K=L=1$) ; 从均衡的角度看,这是一个具有挑战性的问题。此外,在有噪声的场景中,分析了信噪比(SNR)对 ESN 性能的影响;如果不考虑噪声的存在,这意味着 ESN 的输入 $u(n)$ 完全等于 $r(n)$ 。

每个网络都使用输入信号的 $T_s=1100$ 个样本点进行训练,丢弃前 100 个样本点,以消除瞬态效应。在测试阶段,也采用同样的做法。将期望信号 $d(n)$ 与网络输出 $z_{\text{ESN}}(n)$ 间的均方差(MSE)定义为

$$\text{MSE} = \frac{1}{T_s} \sum_{n=1}^{T_s} [d(n) - z_{\text{ESN}}(n)]^2 \quad (5.5.9)$$

为了持续评价网络的性能,用 20 次独立运行的平均 MSE(AMSE)作为 MSE 的标准偏差,其反映了由回声状态网络获得的每个信道均衡性能的变化。用内部神经元的不同数量来观察每个回声状态网络的性能。在初步仿真基础上,内部单元的数量在所有实验中保持 $N=40$,而主成分的数量是变化的。

根据文献[47],ESN 的构建方法如下:输入权系数以相同概率设置为 -1 或 $+1$,而存储层权重矩阵 \mathbf{W} 的元素分别设为 -0.4 、 0.4 和 0 ,其对应的概率分别为 0.025 、 0.025 和 0.95 。

对于 ASE-ESN 和本节提出的网络, \mathbf{W}^{in} 的调整类似于 ESN 中的过程。就存储层权重矩阵 \mathbf{W} 而言,ASE-ESN 采用文献[48]的方法,谱半径为 0.8 。由于本节给出的网络可以很容易地适应不同技术设计的存储层,可使用文献[47]和[48]中方法分析其性能。将它们分别称为提出的 ESN 和提出的 ASE-ESN。为了更简洁,将使用产生最佳性能方法所获得的结果进行简化。最后, \mathbf{W}^{back} 在所有的网络中都被设为零,类似于文献[47]和[48]的处理方式。

对于整个网络,给定一个期望信号 $d(n)$,输出权重矩阵 \mathbf{W}^{out} 由维纳解定。需要强调的是输出权系数与信号有关:在 ESN 和 ASE-ESN 中,回声状态 $x(n)$ 是由线性组合生成输出的。在本节给出的网络中,使用 PCA 和 Volterra 滤波器作为输出层的基本框架。在这种情况下,输出权系数对应在内核函数组合中使用的系数。另一方面,在该网络中,采用三阶 Volterra 滤波器计算输出,为了简单起见,不考虑二次项。

2. 第一信道

第一信道的传递函数为

$$H(z) = 0.5 + z^{-1} \quad (5.5.10)$$

这似乎是一个相对简单的信道,但均衡延迟为零时,不能用线性滤波器进行均衡,因为在这种情况下,相应的信道状态不能被线性分离。表 5.1 给出了不同网络结构的性能比较结果,表中括号内的值对应于标准偏差。

表 5.1 第一信道时,由每个回声状态网络得到的 AMSE 值

信道 1		AMSE	
网络	参数	训练	测试
ESN	$N=10$	$2.66(\pm 3.22)e-01$	$2.72(\pm 3.27)e-01$
	$N=40$	$1.87(\pm 2.65)e-05$	$2.07(\pm 3.03)e-05$
	$N=60$	$2.37(\pm 2.19)e-05$	$2.74(\pm 2.58)e-05$
ASE-ESN	$N=10$	$1.24(\pm 0.60)e-01$	$1.28(\pm 0.59)e-01$
	$N=40$	$1.65(\pm 0.97)e-02$	$2.54(\pm 3.27)e-02$
	$N=60$	$9.56(\pm 7.08)e-03$	$1.72(\pm 2.88)e-02$
提出的 ASE-ESN	$N_{\text{pc}}=3$	$1.02(\pm 0.78)e-02$	$1.06(\pm 0.78)e-02$
	$N_{\text{pc}}=5$	$1.09(\pm 0.81)e-03$	$1.58(\pm 1.78)e-03$
	$N_{\text{pc}}=6$	$4.09(\pm 4.04)e-04$	$8.61(\pm 17.7)e-04$
	$N_{\text{pc}}=10$	$4.71(\pm 2.47)e-06$	$6.63(\pm 18.2)e-04$
提出的 ESN	$N_{\text{pc}}=3$	$4.26(\pm 4.34)e-03$	$4.36(\pm 4.36)e-03$
	$N_{\text{pc}}=5$	$6.28(\pm 9.72)e-05$	$7.38(\pm 11.2)e-05$
	$N_{\text{pc}}=6$	$2.56(\pm 3.80)e-06$	$3.06(\pm 4.60)e-06$
	$N_{\text{pc}}=10$	$3.34(\pm 4.02)e-10$	$2.06(\pm 4.86)e-08$

表 5.1 表明:

(1) 当动态库内的神经元数量或主成分数量增加时,AMSE 值减小,ESN 的性能得到提高。

(2) 比较由 ASE-ESN 和提出的 ASE-ESN 方法得到的 AMSE 值知,使用非线性输出层,网络性能得到明显提高。事实上,ASE-ESN 中 $N=60$

和提出的 ASE-ESN 中 $N_{pc}=6$ 的情况下,两个网络中要调整的输出权系数数量大致相等(分别为 60 和 62),提出的网络性能更好。

(3) ESN 和 ASE-ESN 的 AMSE 值存在差异性,这表明,根据 Jaeger 的方案设计动态库权重矩阵会产生更好的性能。当比较提出的所有网络的 AMSE 值时,会发现所得结果是可靠的。当动态库按照 ESN 设计时,所提出的网络具有较好的均衡性能,AMSE 值比较小,这也再次表明了该结构的能力。

(4) 由标准偏差值比较知,即使在最坏的情况下,与 ESN 和 ASE-ESN 相比,所提出网络的训练和测试性能都有显著的改善。

3. 第二信道

1999 年, Montalvao 等证明^[61], 线性或非线性的前馈结构不适用于均衡传递函数在单位圆上呈现零点的信道。因此,这类信道要求在均衡器中使用反馈连接,以允许状态的正确分离。例如,对传递函数为 $H(z)=1+z^{-1}$ 的信道,由两个符号的输入序列($a(n)=-1$, $a(n-1)=1$)和($a(n)=1$, $a(n-1)=-1$)产生的值 $r(n)$ 为 0。表 5.2 给出了每个回声状态网络获得的 AMSE 值。

表 5.2 第二信道时,每个 ESN 获得的 ASME 值

信道 2		AMSE	
网络	参数	训练	测试
ESN	$N=10$	$7.97(\pm 4.87)e-02$	$7.58(\pm 4.84)e-02$
	$N=40$	$1.96(\pm 2.22)e-03$	$2.05(\pm 1.78)e-03$
	$N=60$	$1.54(\pm 1.58)e-04$	$7.34(\pm 9.52)e-04$
ASE-ESN	$N=10$	$6.06(\pm 4.40)e-03$	$4.48(\pm 2.61)e-03$
	$N=40$	$9.62(\pm 4.34)e-05$	$2.15(\pm 2.74)e-04$
	$N=60$	$4.83(\pm 2.31)e-05$	$2.98(\pm 5.74)e-04$
提出的 ASE-ESN	$N_{pc}=3$	$4.58(\pm 2.55)e-02$	$4.42(\pm 2.36)e-02$
	$N_{pc}=5$	$3.71(\pm 3.73)e-03$	$5.21(\pm 6.07)e-03$
	$N_{pc}=6$	$5.90(\pm 7.52)e-04$	$1.56(\pm 1.83)e-03$

表 5.2 表明,动态库中的递归连接使 ESN 能够保留过去接收信号的记忆性,这对于原始信号的充分恢复起决定作用。这意味着固定动态库权系数简化了递归网络的训练过程,为网络提供了足够的处理能力。表 5.2 也表明,所提出的网络在 AMSE 方面没有得到任何改善,这可能

与该均衡问题的实质有关,该均衡问题可能更多地依赖于反馈连接的存在而不是所需映射的内在困难。不过,需要强调的是所提出的网络的表现相当理想。

4. 第三信道

设信道的传输函数为 $H(z) = 0.5 + 0.71z^{-1} + 0.5z^{-2}$ 。根据 Proakis 的说法^[62],这是具有三个系数的线性信道,发射信号在其中传输会产生严重的失真。

首先,在无噪声情况下,每个回声状态网络获得的 ASME 值,如表 5.3 所示。

表 5.3 第三信道时,每个 ESN 获得的 ASME 值

信道 2		AMSE	
网络	参数	训练	测试
ESN	$N=10$	$3.03(\pm 1.07)e-01$	$3.08(\pm 1.13)e-01$
	$N=40$	$4.73(\pm 1.51)e-02$	$5.18(\pm 1.61)e-02$
	$N=60$	$1.72(\pm 0.74)e-02$	$2.01(\pm 0.68)e-02$
ASE-ESN	$N=10$	$2.55(\pm 0.95)e-01$	$2.59(\pm 0.98)e-01$
	$N=40$	$9.64(\pm 4.74)e-02$	$1.05(\pm 0.43)e-01$
	$N=60$	$7.19(\pm 1.96)e-02$	$7.53(\pm 2.01)e-02$
提出的 ESN	$N_{pc}=3$	$2.30(\pm 1.82)e-03$	$2.46(\pm 1.93)e-03$
	$N_{pc}=5$	$5.56(\pm 5.95)e-04$	$6.60(\pm 7.20)e-04$
	$N_{pc}=6$	$9.00(\pm 8.40)e-07$	$1.01(\pm 1.88)e-05$

表 5.3 表明,非线性输出层使 ESN 和 ASE-ESN 性能显著改善。另一方面,由文献[48]中方法设计的网络结构动态库权重矩阵也提供了良好的性能,但性能改善不是很显著。

其次,在高斯加性噪声情况下,回声状态网络的输入为 $u(n) = r(n) + \sigma N(0,1)$,其中 σ 为噪声的标准差, $N(0,1)$ 对应零均值和单位方差的高斯随机变量。

假设 ESN 和 ASE-ESN 的 $N=60$,并且对于所提出的网络, $N_{pc}=6$ 。在五种不同的信噪比的情况下,针对每个网络进行了 20 次独立实验。表 5.4 给出了所获得的 AMSE 值。动态库由文献[47]给出的方法进行设计,因为这种选择在无噪声情况下,会产生更好的性能。

表 5.4 第三信道时,每个回声状态网络获得的 AMSE 值与 SNR 有关

信道 3		AMSE	
SNR	网络	训练	测试
0dB	ESN	$8.0714e-01$	$9.3295e-01$
	ASE-ESN	$8.0659e-01$	$1.0009e-00$
	提出的 ESN	$8.0567e-01$	$9.1407e-01$
5dB	ESN	$6.6104e-01$	$7.5504e-01$
	ASE-ESN	$6.6828e-01$	$7.7295e-01$
	提出的 ESN	$6.5706e-01$	$7.4966e-01$
10dB	ESN	$3.8648e-01$	$4.4366e-01$
	ASE-ESN	$4.5590e-01$	$5.3292e-01$
	提出的 ESN	$3.7231e-01$	$4.2315e-01$
15dB	ESN	$1.6712e-01$	$1.9371e-01$
	ASE-ESN	$2.5561e-01$	$2.9896e-01$
	提出的 ESN	$1.4976e-01$	$1.7238e-01$
20dB	ESN	$7.6860e-02$	$9.0315e-02$
	ASE-ESN	$1.3896e-01$	$1.6526e-01$
	提出的 ESN	$5.8613e-02$	$6.8043e-02$

表 5.4 表明,随着噪声变得更加明显,误差显著增加。尽管如此,SNR 值一定时,每个网络的 AMSE 值之间的差异相对较小,但所提出的网络性能最好。

5. 第四信道

信道的传输函数为 $H(Z)=0.38+0.6z^{-1}+0.6z^{-2}+0.38z^{-3}$,该信道会在其中传输的信号产生严重失真。仿真中,每个网络训练和测试样本数量为 5500 个。为了避免瞬态效应,前 500 个样本被剔除。所得结果如表 5.5 所示。

表 5.5 第四信道时,每个 ESN 获得的 ASME 值

信道 4		AMSE	
网络	参数	训练	测试
ESN	$N=10$	$5.24(\pm 0.89)e-01$	$5.25(\pm 0.90)e-01$
	$N=40$	$1.74(\pm 0.16)e-01$	$1.73(\pm 0.19)e-01$
	$N=60$	$1.37(\pm 0.11)e-01$	$1.39(\pm 0.11)e-01$
ASE-ESN	$N=10$	$4.32(\pm 0.44)e-01$	$4.31(\pm 0.43)e-01$
	$N=40$	$2.95(\pm 0.30)e-01$	$3.00(\pm 0.30)e-01$
	$N=60$	$2.62(\pm 0.22)e-01$	$2.70(\pm 0.24)e-01$

续表

信道 4		AMSE	
网络	参数	训练	测试
提出的 ESN	$N_{pc} = 3$	$3.59(\pm 0.57)e-01$	$3.56(\pm 0.56)e-01$
	$N_{pc} = 5$	$7.84(\pm 2.36)e-02$	$7.82(\pm 2.37)e-02$
	$N_{pc} = 6$	$3.78(\pm 1.33)e-02$	$3.85(\pm 1.35)e-02$

表 5.5 表明：

(1) 所提出的网络比 ESN 和 ASE-ESN 有更好均衡效果。这意味着，在这种情况下不仅有必要采用递归结构，而且非线性输出层也是必不可少的。

(2) 当传输信号中的失真变得更加明显时，所提出的网络确实能够实现更有效的信道均衡。

6. 第五信道

非线性信道为

$$y_{\text{channel}}(n) = y_{(1)}(n) + 0.25y_{(1)}^2(n) \quad (5.5.11)$$

式中， $y_{(1)}(n)$ 为式(5.5.10)定义的第一个线性信道的输出。

信道的非线性给均衡带来了额外的困难，因此非线性映射的执行能力是实现信号恢复的决定性因素。在这个背景下，分析每个 ESN 的处理能力是非常必要的。表 5.6 给出了用 ESN 获得的性能。

表 5.6 非线性信道的 ESN 的性能

信道 5		AMSE	
网络	参数	训练	测试
ESN	$N = 10$	$5.66(\pm 1.84)e-01$	$5.70(\pm 1.85)e-01$
	$N = 40$	$9.97(\pm 3.94)e-02$	$1.04(\pm 0.38)e-01$
	$N = 60$	$4.38(\pm 2.26)e-02$	$5.34(\pm 2.77)e-02$
ASE-ESN	$N = 10$	$5.05(\pm 0.43)e-01$	$5.10(\pm 0.66)e-01$
	$N = 40$	$4.05(\pm 0.23)e-01$	$4.52(\pm 0.39)e-01$
	$N = 60$	$3.79(\pm 0.24)e-01$	$4.64(\pm 0.46)e-01$
提出的 ESN	$N_{pc} = 3$	$4.65(\pm 4.26)e-02$	$4.65(\pm 4.28)e-02$
	$N_{pc} = 5$	$2.94(\pm 3.44)e-04$	$3.10(\pm 3.55)e-04$
	$N_{pc} = 6$	$2.19(\pm 2.20)e-05$	$2.86(\pm 3.25)e-05$
	$N_{pc} = 10$	$1.22(\pm 1.34)e-08$	$3.21(\pm 5.19)e-07$

表 5.6 表明,由所提出的网络获得的 AMSE 值比 ESN 和 ASE-ESN 要小很多个数量级。

基于以上分析知,使用更灵活的结构作为回声状态网络输出层(如使用 Volterra 滤波器),是所提出的 ESN 结构获得良好性能的关键因素,特别是在失真严重情况下,这种设计更加必要。此外,使用 PCA 使均衡性能得到了明显改善,而没有增加网络训练过程的复杂性,这是该方案的另一个有吸引力的特点。

参考文献

- [1] Meloni A, Murroni M. On the genetic optimization of APSK constellations for satellite broadcasting[C]. Processing of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Beijing, 2014: 1-6.
- [2] 赵海全,张家树.混沌通信系统中非线性信道的自适应组合神经网络均衡[J].物理学报. 2008, 57(7): 3996-4006.
- [3] Deleu T, Dervin M, Kasai K, et al. Iterative predistortion of the non-linear satellit channel [J]. IEEE Transactions on Communications, 2014, 62 (8): 2916-2926.
- [4] 郭业才,徐冉.改进的非线性卫星信道均衡器[J].计算机应用, 2013, 33(11): 3039-3041.
- [5] Singerl P, Koepl H. A low-rate identification method for digital predistorters based on volterra kernel interpolation[C]. Proc of The 48th Midwest Symposium on Circuits and Systems, Ohio, 2005: 1533-1536.
- [6] Bolstad A, Miller B A, Goodman J, et al. Identification and compensation of Wiener-Hammerstein systems with feedback [C]. Processing of the IEEE International Conference on Speech and Signal Processing(ICASSP), Acoustics, 2011: 22-27.
- [7] Ogunfunmi T, Drullinger T. Equalization of non-linear channels using a volterra-based non-linear [C]. Processing of the IEEE 54th International Midwest Symposium on Circuits and Systems (MWS CAS),2011:1-4.
- [8] Benammar B, Thomas N, Poulliat C, et al. On linear MMSE based turbo-equalization of non-linear volterra channels [C]. Processing of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), Vancouver,BC,2013: 4703-4707.
- [9] Zerdoumi Z, Chikouche D and Benatia D. Adaptive decision feedback equalizer

- based neural network for non-linear channels [C]. Processing of the 3rd International Conference on Systems and Control(ICSC), Algiers, 2013: 850-855.
- [10] 刘顺兰,蒋树南.非线性信道的 LMS Newton 均衡算法的研究[J].电路与系统学报, 2011, 16(1): 108-115.
- [11] 郭业才,刘振兴.基于平衡正交多小波变换的盲均衡算法[J].兵工学报, 2010, 31(3): 279-284.
- [12] Giacoumidis E, Matin A, Wei J, et al. Blind Nonlinearity Equalization by machine learning based clustering for single-and multi-channel coherent optical OFDM[J]. Journal of Lightwave Technology, 2018,(99): 1-1.
- [13] Abdulkader H, Benammar B, Poulliat C, et al. Neural networks-based turbo equalization of a satellite communication channel [C]. IEEE, International Workshop on Signal Processing Advances in Wireless Communications, 2014: 494-498.
- [14] Vapnik V. Statisticel learning theory[M]. NewYork: John Wiley& Sons,1998.
- [15] Cortes C, Vapnik V. Support vector networks[J]. Machine Learning, 1995, 20 (3): 273-297.
- [16] 李兵,蒋慰孙.混沌优化方法及其应用[J].控制理论与应用, 1997, 14(4): 613-615.
- [17] Grigoras V, rigoras C. Chaos parameter modulation equalization [C]. International Conference on Communications(COMM), Bucharest, 2010: 33-36.
- [18] Zhang Y, Zuo F, Zhai Z, et al. A new image encryption algorithm based on multiple chaos system[C]. International Symposium on Electronic Commerce and Security, IEEE Computer Society, 2008: 347-350.
- [19] Jaeger H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note[R]. Bonn, Germany: German National Research Center for InformationTechnology GMD Technical Report, 2001, 148(34): 13.
- [20] Boccato L, Lopes A, Attux R, et al. An echo state network architecture based on Volterra filtering and PCA with application to the channel equalization problem [C]. Neural Networks (IJCNN), The 2011 International Joint Conference on IEEE, 2011: 580-587.
- [21] Liu D N, Fitz M P. Iterative equalization in non-linear satellite channels[C]. Turbo codes and iterative information processing (ISTC), 2012 7th International Symposium on IEEE, 2012: 220-224.
- [22] 崔鹏辉,杨育红,朱义君,等.非线性卫星信道中的 16APSK 联合处理算法[J].数据采集与处理, 2013, 28(6): 823-827.
- [23] 樊小琴,张焱.功放非线性对 QAM 解调误码率的影响[J].通信技术, 2014, 47 (5): 504-507.
- [24] Guo Y C, Ji J. Variable momentum factor decision feedback blind equalization algorithm based on constant parameter error function [M]. Information and

- Automation. Berlin Heidelberg: Springer, 2011: 282-288.
- [25] Benedetto S, Biglieri E, Daffara R. Modeling and performance evaluation of nonlinear satellite links—a volterra series approach[J]. IEEE Transactions on Aerospace & Electronic Systems, 2007, AES-15(4): 494-507.
- [26] 郭业才,费赛男,王惠.基于多小波双变换的非线性卫星信道盲均衡算法[J].电子学报, 2016, 44(10): 2384-2391.
- [27] 程止兴,张玲玲.多小波分析与应用[J].工程数学学报, 2001, 18(1): 99-107.
- [28] 邵华平,覃征,钱清泉.多小波的构造与应用[J].湖南师范大学自然科学学报, 2005, 28(1): 31-36.
- [29] 郭业才.模糊小波神经网络盲均衡理论、算法与实现[M].北京:科学出版社, 2011.
- [30] 费赛男.基于多小波神经网络的非线性信道盲均衡算法[D].南京:南京信息工程大学, 2015.
- [31] 陈丽,陈静,高新涛,等.基于支持向量机与反 K 近邻的分类算法研究[J].计算机工程与应用, 2010, 46(24): 135-138.
- [32] 郭业才,徐文才,许芳.混沌支持向量机优化小波加权多模盲均衡算法[J].系统仿真学报, 2013, 25(3): 451-455.
- [33] 郭业才,孙静,徐文才.混沌优化的正交小波多模盲均衡方法:中国, CN201110208286.9[P]. 2011-11-30.
- [34] 郭业才,徐文才,许芳.混沌与最速下降法联合优化的正交小波常数模盲均衡方法:中国,ZL201110208373.4[P]. 2014-03-26.
- [35] 郭业才,徐文才.基于混沌支持向量机优化的小波加权多模盲均衡方法:中国, CN201110159986.3[P]. 2012-02-22.
- [36] 王世元,冯久超.基于混沌的通信系统的盲信道均衡[J].西南师范大学学报(自然科学版), 2004, 29(3): 373-378.
- [37] 丁雪洁.基于分集技术的正交小波盲均衡算法[D].南京:南京信息工程大学, 2010.
- [38] 朱婕.基于分集技术的盲均衡算法与仿真研究[D].淮南:安徽理工大学, 2008.
- [39] 郭业才,胡苓苓,丁锐.基于量子粒子群优化的正交小波加权多模盲均衡算法[J].物理学报, 2012, 61(5): 54304-054304.
- [40] 郭业才,刘振兴.基于平衡正交多小波变换的模糊神经网络盲均衡方法:中国, ZL201010267953.6[P]. 2013-10-30.
- [41] 郭业才,王丽华.模糊神经网络控制的混合小波神经网络盲均衡方法:中国, ZL201010267951.7[P]. 2013-07-03.
- [42] 郭业才,王丽华.模糊控制的动态小波神经网络反馈盲均衡方法:中国, ZL201010216296.2[P]. 2013-10-30.
- [43] 潘亚莉.基于混沌神经网络的多用户检测技术的研究[D].重庆:重庆大学, 2006.
- [44] 孙静.基于混沌理论的正交小波变换盲均衡算法[D].南京:南京信息工程大

- 学,2012.
- [45] 郭业才,孙静. 基于混沌系统的正交小波变换盲均衡算法[J]. 控制工程,2012, 19(3): 443-447.
- [46] 高敏,郭业才. 基于混沌萤火虫优化的小波多模盲均衡算法[J]. 计算机工程, 2014, 40(1): 213-218.
- [47] Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560.
- [48] Williams R J, Zipser D. A learning algorithm for continually running fully recurrent neural networks[J]. Neural computation. 1989, 1(2): 270-280.
- [49] Jaeger H. The echo state approach to analyzing and training recurrent neural networks[R]. Bremen: German National Research Center for Information Technology, Tech. Rep. GMD Report 148, 2001.
- [50] Ozturk M C, Xu D, Principe J C. Analysis and design of echo state networks [J]. Neural computation, 2007, 19(1): 111-138.
- [51] Boccato L, Lopes A, Attux R, et al. An Echo State Network Architecture Based on Volterra Filtering and PCA with Application to the Channel Equalization Problem[C]. San Jose: Proceedings of International Joint Conference on Neural Networks, July 31-August 5, 2011, 3(14): 580-588.
- [52] Mathews V J. Adaptive polynomial filters[J]. IEEE signal processing magazine, 1991, 8(3): 10-26.
- [53] Bellman R. Dynamic programming[M]. Courier Corporation, 2013.
- [54] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and intelligent laboratory systems, 1987, 2(1-3): 37-52.
- [55] Hyvärinen A, Hurri J, Hoyer P O. Independent component analysis [M]. Natural Image Statistics. London: Springer, 2009: 151-175.
- [56] Haykin S S. Adaptive filter theory[M]. India: Pearson Education, 2008.
- [57] Adali T. Why a nonlinear solution for a linear problem? [C]. Neural Networks for Signal Processing Ix, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, 1999: 157-165.
- [58] Jaeger H. Short term memory in echo state networks [M]. GMD-Forschungszentrum Informationstechnik, 2001.
- [59] Jaeger H. Adaptive nonlinear system identification with echo state networks[C]. Advances in neural information processing systems, 2003: 609-616.
- [60] Jaeger H, Haas H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication[J]. Science, 2004, 304(5667): 78-80.
- [61] Salmen M, Ploger P G. Echo state networks used for motor control[C]. IEEE International Conference on Robotics & Automation. IEEE, 2005: 1953-1958.
- [62] Erdogmus D, Principe J C. From linear adaptive filtering to nonlinear

- information processing-The design and analysis of information processing systems [J]. IEEE Signal Processing Magazine, 2006, 23(6): 14-33.
- [63] Montalvao J, Dorizzi B, Mota J C M. Some theoretical limits of efficiency of linear and nonlinear equalizers[J]. Journal of the Brazilian Telecommunications Society, 1999, 14(2): 85-92.
- [64] Proakis J G. Digital Communications[M]. McGraw-Hill, 1995.