



引用格式:高明,崔钊,李昊,等. 基于 HCPS 多层感知器的污水处理后氨氮浓度测量[J]. 轻工学报,2018,33(6):92-100.

中图分类号:TP273;TS97;X703.1 文献标识码:A

DOI:10.3969/j.issn.2096-1553.2018.06.011

文章编号:2096-1553(2018)06-0092-09

基于 HCPS 多层感知器的污水处理后氨氮浓度测量

Measurement of ammonia nitrogen concentration after sewage treatment based on HCPS multilayer perceptrons

高明¹,崔钊¹,李昊¹,栗三一²

GAO Ming¹, CUI Ke¹, LI Hao¹, LI Sanyi²

关键词:

污水处理;氨氮浓度预测;软测量技术;神经网络结构自调整算法;多层感知器;结构自组织

1. 黄河流域水环境监测中心 监测管理处, 河南 郑州 450004;
2. 郑州轻工业学院 电气信息工程学院, 河南 郑州 450001
1. *Monitoring Management Service, Yellow River Basin Water Environment Monitoring Center, Zhengzhou 450004, China;*
2. *College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China*

Key words:

sewage treatment; ammonia nitrogen concentration prediction; soft measurement; HCPS; multilayer perceptron; structural self-organizing

摘要:针对现污水处理后出水氨氮预测模型中隐含层神经元存在过大冗余而浪费资源的问题,提出了一种基于敏感度和互信息的混合增加删减的神经网络结构调整算法(HCPS)。该算法重新定义了敏感度公式,利用敏感度和互信息自适应地调整网络结构,删除敏感度过低的隐含神经元,分裂过大的隐含层神经元,合并互信息过大的两个隐含层神经元。在污水处理基准仿真平台 BSM1 上的验证结果表明,HCPS 算法可以获得更紧凑的网络结构,用于出水氨氮浓度预测精度较高。

收稿日期:2018-10-18

基金项目:国家自然科学基金项目(61603347)

作者简介:高明(1986—),女,河南省宜阳县人,黄河流域水环境监测中心工程师,硕士,主要研究方向为水环境监测与研究。

Abstract: In the existing prediction model of ammonia nitrogen in sewage treatment effluent, the hidden layer neurons excessively redundant, which wastes resources. A new structure adjustment algorithm (HCPS) based on sensitivity analysis (SA) and mutual information (MI) was proposed. The algorithm redefined the sensitivity formula, adaptively, adjusted the network structure by using sensitivity and mutual information, deleted hidden neurons with low sensitivity, divided hidden neurons with excessive sensitivity, and merged two hidden neurons with excessive mutual information. The results of verification on BSM1, a benchmark simulation platform for sewage treatment, showed that HCPS algorithm could obtain a more compact network structure, and the prediction accuracy of ammonia nitrogen concentration in effluent was higher.

0 引言

随着工业的发展和人们环保意识的增强,国家不断提高城市和工业污水处理排放标准,污水处理厂面临着提高出水水质的挑战.氨氮含量是评价水质的一个重要指标.氨氮的主要成分为硝酸盐氮($\text{NO}_3\text{-N}$),氨氮浓度的增加会导致水体富营养化,污染环境,且对人类健康产生极为不利的影 响,因此国家对污水排放氨氮浓度做了明确规定并对氨氮超标的污水处理厂进行严厉处罚^[1].为了提高污水处理效果,实现出水水质达标,需要对氨氮浓度等关键变量进行在线实时监测.目前污水处理厂大多使用分光光度法进行氨氮浓度的测量,即对污水采样后在实验室中通过化学反应进行测定,该方法易受悬浮物和有色离子干扰且十分耗时,难以达到实时监测的目标^[2].此外,污水处理过程包含复杂的生物化学反应,干扰严重,变量之间具有强耦合性、非线性、时变性和滞后性等特点,因此很难通过机理分析建立测量氨氮浓度的数学模型^[3].为了解决该问题,有学者提出将基于数据驱动的软测量技术应用于污水处理中的氨氮检测.

软测量技术的核心是使用易测变量对难以测得的变量进行预测,主要使用回归分析、状态估计和神经网络等智能方法建模,其中多层感知器神经网络因其工程易实现和强大的非线性拟合能力被广泛应用于多个领域^[4-6].但是,使用多层感知器建立氨氮软测量模型时,其精度

并不理想,主要原因之一是网络结构难以确定.神经网络的结构对网络性能有重要影响:网络规模过小,不能有效解决问题;网络规模过大,则会出现过拟合现象,导致泛化能力变差^[7].为了获得最佳的网络结构,有学者根据生物、数学等机制,动态地增加或删减神经元,提出多种网络结构自调整算法. J. H. Wang 等^[8]基于人类学习的集成原理提出一种增长型神经网络,并且该增长型神经网络构建方法是无监督的,该方法对具有不同特征的数据建立相应的子网络. J. A. Platt^[9]提出一种基于资源分配的增长型网络结构调整算法(RAN),该算法可以根据输入样本对输出进行分类,当训练数据没有类别时会增加隐含层神经元. P. Lauret 等^[10]对全连接单层神经网络提出一种删减算法,该算法基于全局敏感度分析,通过计算每个隐含层神经元对整个网络输出的影响,删除影响较小的神经元,保留影响较大的隐含层神经元. M. M. Islam 等^[11]提出自适应删减增长算法(AM-GA),该算法在网络训练的过程中增加和删减神经元.删除神经元是一种混合操作,在删除两个神经元的同时会增加一个神经元;增加神经元是一种分裂操作,即分裂一个最活跃的隐含层神经元.这种混合增长删减算法保留了隐含层神经元自身的功能. H. G. Han 等^[12]提出的增长删减算法(CPNN)使用敏感度分析评价隐含层神经元的贡献度,删除贡献度过小的隐含层神经元,当网络性能不足时增加神经元,并且在增加和删减神经元的同时对临近神经元的权

值进行补偿,减小网络变化对网络输出带来的影响. 以上算法可以动态地调节网络结构. 然而这些算法均没有考虑隐含层神经元之间的信息,因此会导致隐含层神经元出现冗余. 冗余神经元的存在会浪费计算资源,增加计算成本,同时也会降低网络的泛化能力,影响网络精度.

为了消除隐含层冗余神经元,通过借鉴生物神经元结构自调整机制,本文拟提出一种基于敏感度和互信息的混合增加删减的神经网络结构自调整算法(HCPS),使用互信息衡量隐含层神经元之间的相关性,用敏感度分析方法评价隐含层神经元对网络输出的贡献度,从而充分地利用网络层内和层间信息来调整网络结构,以期获得简洁有效的网络结构,从而用于提高污水处理后的氨氮浓度测量精度.

1 互信息估计与敏感度分析

1.1 多层感知器

多层感知器结构分为3部分,即输入层、隐含层和输出层^[13]. 由于单隐含层的多层感知器可以逼近任意精度的曲线,因此本文主要研究单隐含层多层感知器. 对于多输入多输出多层感知器,设其输入层神经元、隐含层神经元和输出层神经元个数分别为 n_0, n_1 和 n_2 . 输入变量为 $X = [x_1, x_2 \dots, x_{n_0}]$, 则隐含层神经元输入和输出分别为

$$z_j = \sum_{i=1}^{n_0} w_{ij}x_i + b_j \tag{①}$$

$$y_j = f(z_j) = (1 + e^{-z_j})^{-1} \tag{②}$$

其中, z_j 是第 j 个隐含层神经元的输入, b_j 是第 j 个隐含层神经元的阈值, w_{ij} 是第 i 个输入神经元与第 j 个隐含层神经元之间的权值, y_j 是第 j 个隐含层神经元的输出, $f(z_j)$ 是激活函数. 输出层神经元的输入和输出分别为

$$z_k = \sum_{j=1}^{n_1} w_{jk}y_j + b_k \tag{③}$$

$$y_k = z_k \tag{④}$$

其中, z_k 是第 k 个输出层神经元的输入, b_k 是第 k 个输出层神经元的阈值, w_{jk} 是第 j 个隐含层神经元与第 k 个输出层神经元之间的权值, y_k 是第 k 个输出层神经元的输出.

1.2 互信息

熵和互信息是信息论中的两个基本概念. 熵可用于描述信源的不确定度,这里可以表示一个变量中包含的信息量. 互信息可以测量两个随机变量之间的相互关系. 给定一个离散随机变量 $X = \{x_i | i = 1, \dots, K\}$, 则 X 的熵可表示为

$$H(X) = - \sum_{k=1}^K p(x_k) \log p(x_k)$$

其中, $p(x_k)$ 是概率密度函数, K 是变量 X 的维度. 联合熵可表示多个变量中的信息量. 对于二变量 (X, Y) , 联合熵可表示为

$$H(X, Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y)$$

其中, $p(x, y) = p\{X = x, Y = y\}$ 是 X 和 Y 的联合分布律, $x \in \mathcal{X}, y \in \mathcal{Y}$, \mathcal{X} 和 \mathcal{Y} 分别是 X 和 Y 的取值空间.

互信息的计算公式为

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p_X(x)p_Y(y)}$$

其中, $p_X(x)$ 和 $p_Y(y)$ 是 X 和 Y 的边缘概率密度.

互信息与熵的关系可表示为

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \tag{⑤}$$

1.3 多层感知器的互信息估计

估计两个神经元之间的互信息的关键是计算概率分布函数. 目前计算互信息的方法大致有4类,即联合熵法、核函数法、 k 最近邻法和直方图法. 本文采用最简单的、计算效率最高的直方图方法.

假设多层感知器两个神经元的输出变量分别为 X 和 Y , 计算过程分两步.

第一步:估计变量 X 的熵. 直方图方法将变

量 X 的取值范围等分为宽度为 Δx 的 K_x 个片段. 为了增加熵估计的精度, K_x 可由如下公式计算^[14]:

$$K_x = \text{round} \left\{ \frac{\varepsilon}{6} + \frac{2}{3\varepsilon} + \frac{1}{3} \right\}$$

$$\varepsilon = \sqrt[3]{8 + 324N + 12 \sqrt{36N + 729N^2}}$$

其中, N 为样本总数.

由于变量 X 为离散随机变量, 故 X 的熵可以表示为

$$H(X) = - \sum_{k=1}^{K_x} p_k(x) \log p_k(x)$$

其中, $p_k(x)$ 为样本落入第 k 个片段的概率, 可表示为

$$p_k(x) = \frac{n_k}{N}$$

其中, n_k 为样本落入第 k 个片段中的个数. 则 X 的熵为

$$H(X) = - \sum_{k=1}^{K_x} \frac{n_k}{N} \log \frac{n_k}{N} \quad (6)$$

变量 Y 的熵的计算过程与 X 相同.

第二步: 估计变量 X 和 Y 的联合熵. 直方图方法将 $x - y$ 平面平均分为 $(K_x \times K_y)$ 个单元, 每个单元的尺寸为 $(\Delta x \times \Delta y)$, 则 X 和 Y 的联合熵可以估计为

$$H(X, Y) = - \sum_{k_i=1}^{K_x} \sum_{k_j=1}^{K_y} p_{k_i, k_j}(x, y) \log p_{k_i, k_j}(x, y)$$

其中, $p_{k_i, k_j}(x, y)$ 为样本落入单元 (k_i, k_j) 的概率, 可表示为

$$P_{k_i, k_j}(x, y) = \frac{n_{k_i, k_j}}{N}$$

式中, n_{k_i, k_j} 为落入单元 (k_i, k_j) 中的样本的数量. 则 X 和 Y 的联合熵为

$$H(X, Y) = - \sum_{k_i=1}^{K_x} \sum_{k_j=1}^{K_y} \frac{n_{k_i, k_j}}{N} \log \frac{n_{k_i, k_j}}{N} \quad (7)$$

联合式 (5)(6)(7), 变量 X 和 Y 的互信息可以表示为

$$I(X; Y) = - \sum_{k_i=1}^{K_x} \frac{n_{k_i}}{N} \log \frac{n_{k_i}}{N} - \sum_{k_j=1}^{K_y} \frac{n_{k_j}}{N} \log \frac{n_{k_j}}{N} +$$

$$\sum_{k_i=1}^{K_x} \sum_{k_j=1}^{K_y} \frac{n_{k_i, k_j}}{N} \log \frac{n_{k_i, k_j}}{N}$$

1.4 敏感度分析

敏感度分析法用于评价模型输出对输入因子的依赖程度, 在此可以用于评价隐含层神经元的输出变量对网络输出的贡献. 敏感度分析法使用敏感度指标测量输入变量的重要程度, 定义为由该输入变量产生的输出部分占总输出的比重. 本文使用敏感度分析法测量隐含层神经元对网络输出的贡献度. 敏感度分析法的主要步骤如下.

- 1) 确定模型输入变量和模型输出变量;
- 2) 对每个输入变量分配概率密度函数;
- 3) 基于样本抽样建立输入矩阵;
- 4) 计算输出;
- 5) 计算哪个输入变量对输出变量产生的影响最大.

目前已有多种方法分析模型输出的敏感度, 但大多都依赖于敏感度指标. 本文敏感度模型的输入变量为 $[w_{11}y_1, w_{21}y_2, \dots, w_{n_1}y_{n_1}; w_{12}y_1, w_{22}y_2, \dots, w_{n_2}y_{n_1}; w_{1n_2}y_1, w_{2n_2}y_2, \dots, w_{n_2n_2}y_{n_1}]$, 其维数为 $n_2 \times n_1$, 其中 $w_{n_1n_2}$ 为第 n_1 个隐含层神经元与第 n_2 个输出神经元之间的权值, y_{n_1} 为第 n_1 个隐含层神经元的输出, 则敏感度模型可以表示为

$$Y^k = f(w_{1k}y_1, w_{2k}y_2, \dots, w_{n_1k}y_{n_1})$$

其中, Y^k 是第 k 个模型响应; $w_{1k}y_1, w_{2k}y_2, \dots, w_{n_1k}y_{n_1}$ 为输入变量值, 反映了每个隐含层神经元与第 k 个输出神经元之间的关系.

传统的敏感度计算公式为

$$S_j^k = \frac{\text{Var}_{w_{jk}y_j} [Y^k(w_{jk}y_j = \partial_j)]}{\text{Var}(Y^k)} \quad (8)$$

其中, S_j^k 为第 j 个输入变量对第 k 个模型输出的贡献率; $w_{jk}y_j$ 是对于第 k 个模型的第 j 个输入变量, $j = 1, 2, \dots, n_1, k = 1, 2, \dots, n_2$; $[Y^k | w_{jk}y_j = \partial_j]$ 表示第 k 个模型输出值中由第 j 个隐含层神经元产

生的部分; $Var[Y^k]$ 为第 k 个网络输出变量的方差. 为了增加公式的可解释性, 将式 ⑧ 改写为

$$S_j^k = \frac{Var_{w_{aj}}[Y^k | w_{jk}y_j = \partial_j]}{\sum_{i=1}^{n_i} Var_{w_{ai}^{(1)}}[Y^k | w_{ik}y_i = \partial_i]} \quad \text{⑨}$$

由式 ⑨ 可以计算输入因子对输出的影响, 输入变量的活跃性越高, 贡献率就越大.

2 基于敏感度和互信息的混合增加删减神经网络结构自调整算法(HCPS)

2.1 网络结构调整机制

本文提出的 HCPS 算法在网络结构自调整阶段共有 3 种操作, 分别是合并隐含层神经元、删减隐含层神经元和分裂隐含层神经元.

2.1.1 合并隐含层神经元 为了使隐含层神经元之间相关性较小, 即每个隐含层神经元具有各自的功能, HCPS 将互信息超过设定值的两个隐含层神经元合并为一个神经元, 合并的同时调整网络权值使网络输出稳定.

假设 $[x_t, y_t] (t = 1, 2, \dots, N)$ 为训练样本, $x_t = (x_{t1}, x_{t2}, \dots, x_{tm_0})$, $(t = 1, 2, \dots, N)$. 计算第 a, b 两个隐含层神经元之间的互信息 $I(a, b)$, 如果该互信息大于 λ_1 , 则将其合并.

假设 a, b 两个隐含层神经元合并为一个神经元 c , 合并公式为

$$\begin{cases} w_{ic} = mw_{ia} + nw_{ib} \\ w_{ck} = w_{ak} + w_{bk} \\ d_c = md_a + nd_b \end{cases} \quad \text{⑩}$$

其中, w_{ic}, w_{ia} 和 w_{ib} 分别为第 i 个输入神经元与隐含层神经元 c, a 和 b 的连接权值; m 和 n 为系数且 $m + n = 1$; w_{ck}, w_{ak} 和 w_{bk} 分别为第 k 个输出神经元与隐含层神经元 c, a 和 b 的连接权值.

根据公式 ①②③④ 可知, 由神经元 a, b 和 c 产生的网络输出为

$$\begin{cases} y_a(x_t) = \frac{w_{ak}}{1 + e^{-\left(\sum_{i=1}^{n_a} w_{ia}x_{ti} + d_a\right)}} \\ y_b(x_t) = \frac{w_{bk}}{1 + e^{-\left(\sum_{i=1}^{n_b} w_{ib}x_{ti} + d_b\right)}} \\ y_c(x_t) = \frac{w_{ck}}{1 + e^{-\left(\sum_{i=1}^{n_c} w_{ic}x_{ti} + d_c\right)}} \end{cases} \quad \text{⑪}$$

其中, x_t 为第 t 个输入样本; y_a, y_b 和 y_c 分别为由隐含层神经元 a, b 和 c 产生的网络输出. 为了确保网络输出不变, 需要满足

$$y_c(x_t) = y_a(x_t) + y_b(x_t) \quad \text{⑫}$$

根据式 ⑩ 和 ⑪, 公式 ⑫ 可改写为

$$\frac{w_{ak}}{1 + e^{-z_a}} + \frac{w_{bk}}{1 + e^{-z_b}} = \frac{w_{ak} + w_{bk}}{1 + e^{-(mz_a + nz_b)}} \quad \text{⑬}$$

其中, $z_a = \sum_{i=1}^{n_a} w_{ia}x_{ti} + d_a, z_b = \sum_{i=1}^{n_b} w_{ib}x_{ti} + d_b$.

将所有样本训练一遍以后调节网络结构, x_t 为第 N 个训练样本, 由此保证网络输出不变. 公式 ⑬ 可简化为

$$mz_a + nz_b = -\ln A$$

其中

$$A = \frac{w_{ak}e^{-z_a} + w_{ak}e^{-z_a}e^{-z_b} + w_{bk}e^{-z_b} + w_{bk}e^{-z_a}e^{-z_b}}{w_{ak} + w_{bk} + w_{ak}e^{-z_b} + w_{bk}e^{-z_a}}$$

结合条件 $m + n = 1$ 可得

$$m = \frac{\ln A - z_b}{z_a - z_b} \quad n = \frac{z_a - \ln A}{z_a - z_b}$$

由此可以确定式 ⑩.

2.1.2 删减隐含层神经元 当隐含层神经元贡献率很低时, 可认为其为无效神经元, 将其删除. 假设第 j 个隐含层神经元的贡献率 S_j 低于阈值 λ_2 , 则可以将其删除. 为了保证网络输出不变, 距离其最近的神经元(称其为神经元 l , 为与神经元 j 互信息最大的隐层神经元)的权值调整公式为

$$\begin{cases} w'_{il}(t) = w_{il}(t) \\ w'_{lk}(t) = w_{lk}(t) + \frac{y_j}{y_l} + w_{jk}(t) \end{cases} \quad \text{⑭}$$

其中, $w_{il}(t)$ 和 $w'_{il}(t)$ 分别为结构调整前、后第 i 个输入神经元与隐层神经元 l 的连接权值; $w_{lk}(t)$ 和 $w'_{lk}(t)$ 分别为结构调整前、后隐层神经元 l 与输出神经元 k 之间的连接权值; y_j 和 y_l 分别为结构调整前隐层神经元 j 和 l 的输出; $w_{jk}(t)$ 为结构调整前隐层神经元 j 与输出神经元 k 之间的连接权值。

2.1.3 分裂隐含层神经元 当某个隐层神经元贡献度过大时,网络输出会随着该神经元输出剧烈波动,因此需将这种神经元分裂为两个神经元。HCPS 将贡献度大于 λ_3 的隐层神经元分裂为两个神经元。假设隐层神经元 j 的贡献度 S_j 大于 λ_3 , 分裂为两个隐层神经元 a 和 b , 新神经元的权值计算公式为^[15]

$$\begin{cases} w_{ia} = w_{ij} \\ w_{ak} = (1 - \alpha)w_{jk} \\ w_{ib} = w_{ij} \\ w_{bk} = \alpha w_{jk} \end{cases} \quad (15)$$

其中, w_{ia} 和 w_{ib} 分别为输入神经元 i 与隐层神经元 a 和 b 的连接权值; w_{ak} 和 w_{bk} 分别为输出神经元 k 与隐层神经元 a 和 b 的连接权值; α 为突变参数。文献[15]中 α 取值为 0 到 1 之间的随机数。为了避免某个新的隐层神经元贡献度过高, HCPS 设定 α 取值为 0.2 到 0.8 之间的随机数。

2.2 阈值的设定

阈值 λ_1 , λ_2 和 λ_3 的设定是 HCPS 算法的关键,对于不同的训练样本该值是不同的。HCPS 算法中阈值的设定具有自适应性。

1) 假设隐含层神经元个数为 n_1 , 计算每两个隐层神经元之间的互信息。所有互信息的平均值为

$$I_{\text{mean}} = \frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_1-i} I(h_i; h_j)}{n_1(n_1 - 1)/2}$$

其中, $I(h_i; h_j)$ 为隐层神经元 i 和 j 之间的互信息。那么 λ_1 可以设定为

$$\lambda_1 = \partial_1 I_{\text{mean}}$$

其中, ∂_1 为系数,根据经验,对于大部分问题其值设置为 [3, 4] 都比较合适。

2) 计算每个隐层神经元的贡献度。所有贡献度的平均值为

$$S_{\text{mean}} = \sum_{i=1}^{n_1} S_i / n_1$$

其中, S_i 为第 i 个隐层神经元的贡献度。那么 λ_2 和 λ_3 的计算公式分别为

$$\lambda_2 = \partial_2 S_{\text{mean}} \quad \lambda_3 = \partial_3 S_{\text{mean}}$$

其中, ∂_2 和 ∂_3 为系数。根据经验, ∂_2 取值为 [0.3, 0.5] 较合适, ∂_3 取值为 [2, 3] 较合适。

2.3 HCPS 算法流程

HCPS 根据敏感度和互信息自动调整网络结构,目的是使隐含层神经元之间的相关性最小,并且所有隐层神经元都为有效神经元。算法流程如下。

步骤 1 随机初始化隐层神经元,其个数为 n_1 。使用 BP 算法训练网络。

步骤 2 当所有样本训练结束后,计算每个隐含层神经元的贡献度。

步骤 3 计算阈值 λ_2 , 删除贡献度小于 λ_2 的神经元,并根据公式 (14) 调整权值。

步骤 4 计算每两个隐含层神经元的互信息,计算阈值 λ_1 。

步骤 5 将互信息大于 λ_1 的神经元合并,根据公式 (10) 对新神经元权值进行初始化。

步骤 6 计算阈值 λ_3 。将贡献度大于 λ_3 的隐层神经元分裂,根据公式 (15) 对新神经元权值进行初始化。

步骤 7 重新训练神经网络。

步骤 8 当达到期望误差或最大训练步数时停止训练。

3 仿真实验与结果分析

3.1 辅助变量的提取和数据采集

在污水处理过程中,与出水氨氮相关的因

素有出水总磷、出水氧化还原电位、进水总磷、出水 pH 值、温度、厌氧末端氧化还原电位、好氧末端固体悬浮物浓度、好氧前端溶解氧和硝态氮 (NO₃ - N) 等。辅助变量的选择应与待测变量密切相关,如选择过少则预测精度下降。结合文献[15 - 17]综合分析,选取温度、进水总磷、好氧末端固体悬浮物浓度、pH 值、好氧前端溶解氧这 5 个变量作为预测模型的输入变量。

本文以活性污泥污水处理基准仿真模型 BSM1^[18] 为实验平台进行数据采样和验证实验。使 BSM1 模型开环运行 14 d,每 15 min 采样 1 次,记录所有相关变量和出水氨氮浓度,运行 1 次可记录 1344 组数据。改变第 2 分区溶解氧和第 5 分区硝态氮浓度设定值,运行 BSM1 模型 10 次,采用其中 9 次记录的数据作为训练集,剩余的数据作为测试集。最后共采集 13 440 组数据,其中 12 096 组数据作为训练集,剩余的 1344 组数据作为测试集。

3.2 仿真结果与分析

训练前将数据归一化到[-1,1],使用 BP 算法对网络进行训练,初始学习率设为 0.1,最大学习步数为 10 000 步。系数 δ_1, δ_2 和 δ_3 分别为 3,0.45 和 3。为了体现结构调整算法的有效性,初始网络结构设为隐含层神经元很少(2 个)和隐含层神经元很多(100 个)两种情况。

将本文所提出的 HCPS 算法与 CPNN 算法^[19]、mPMIc 算法^[20] 和网络结构固定算法进行对比,结果如表 1 所示。CPNN 算法采用敏感度对网络结构进行调整,mPMIc 算法采用互信息对网络结构进行调整。

表 1 中网络性能评价采用指标为均方根误差,即

$$RMSE(t) = \sqrt{\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^{n_2} (y_k(t) - y'_k(t))^2}$$

其中, $y'_k(t)$ 是在 t 时刻第 k 个输出神经元的期望输出, N 是总样本数。

表 1 不同算法氨氮软测量实验结果

Table 1 Experimental results of soft sensing of ammonia nitrogen using different algorithms

算法	初始隐含层 神经元个数	最终隐含层 神经元个数	RMSE 值	RMSE 标准差
HCPS	100	9	0.390 6	0.081 0
	2	10	0.408 5	0.090 5
网络结构 固定算法	100	100	1.270 9	0.091 3
	2	2	1.632 4	0.097 5
CPNN	100	13	0.052 7	0.095 7
	2	13	0.055 4	0.096 4
mPMIc	100	14	0.061 5	0.094 8
	2	14	0.062 1	0.098 0

从表 1 可以看出,对于固定结构的多层感知器,当隐含层神经元过多或过少时都不能很好地预测氨氮浓度,另外 3 种具有自调整算法结构的神经网络取得的预测精度都比固定结构的高;HCPS 算法获得的最终网络结构最精简,而且网络精度比另外两种结构自调整算法也有所提高。另外,4 种算法的 RMSE 标准差基本相同,说明这 4 种算法的稳定性相近。

图 1 和图 2 分别为初始隐含层神经元为 100 和 2 时的隐含层神经元更新过程。从图 1 和图 2 可以看出,HCPS 算法可以快速收敛到稳定的网络结构。

图 3,图 4 和图 5 分别为 HCPS, CPNN 和

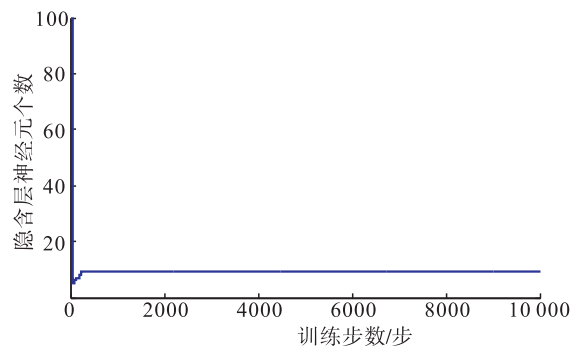


图 1 初始隐含层神经元为 100 时的
隐含层神经元更新过程

Fig. 1 The process of neuron regeneration when the number of initial hidden neurons was 100

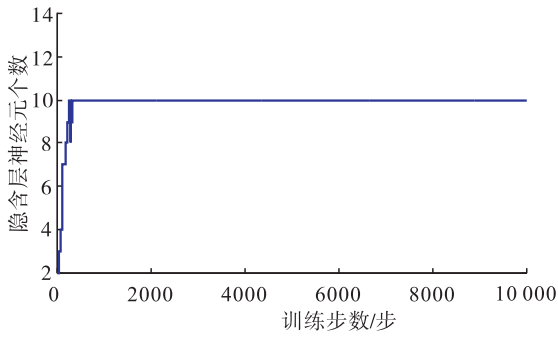


图2 初始隐含层神经元为2时的
隐含层神经元更新过程

Fig.2 The process of neuron regeneration when
the number of initial hidden neurons was 2

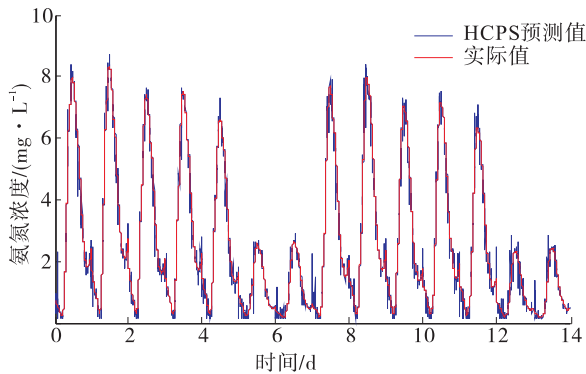


图3 HCPS 算法的出水氨氮浓度预测曲线

Fig.3 The prediction curve of effluent ammonia
nitrogen concentration by HCPS algorithm

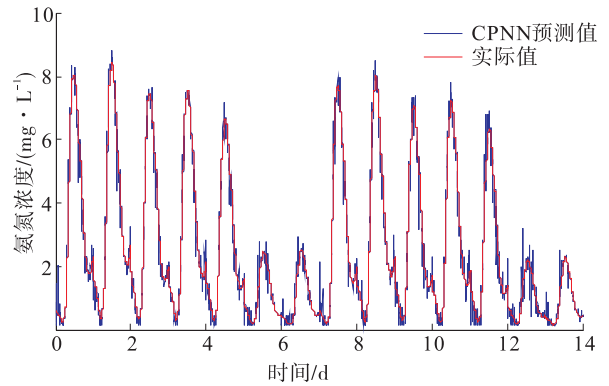


图4 CPNN 算法的出水氨氮浓度预测曲线

Fig.4 The prediction curve of effluent ammonia
nitrogen concentration by CPNN algorithm

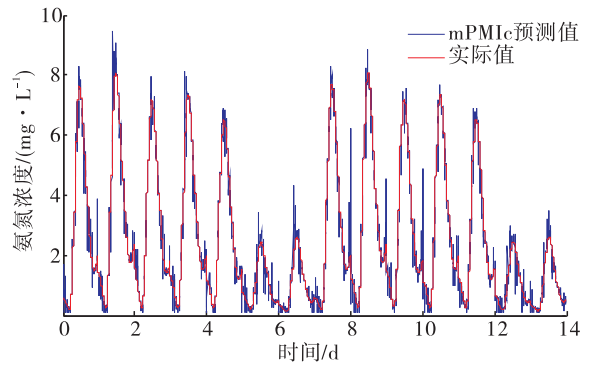


图5 mPMIc 算法的出水氨氮浓度预测曲线

Fig.5 The prediction curve of effluent ammonia
nitrogen concentration by mPMIc algorithm

mPMIc 算法的出水氨氮浓度预测结果. 从图3, 图4 和图5 可以看出, 在峰值处, HCPS 和 CPNN 预测结果的误差小于 mPMIc, 图中 HCPS 和 CPNN 的差别不明显, 但从表1 中测试均方根误差的结果可以看出, HCPS 算法获得的氨氮预测效果比 CPNN 和 mPMIc 的预测效果更好.

4 结论

针对现污水处理后氨氮浓度软测量模型中隐含层神经元存在过大冗余的问题, 本文提出了一种基于敏感度和互信息的混合增加删减神经网络结构自调整算法(HCPS)用于出水氨氮

浓度测量. 该算法核心是通过使用敏感度和互信息同时对网络结构进行调整, 删除贡献过低的隐含层神经元, 分裂贡献度过高的隐含层神经元, 合并互信息过高的隐含层神经元. 由于它同时考虑了层内和层间关系, 因此可以获得更精简的网络结构. 另外, 本文所提出的合并公式可以保证在网络结构变化时网络输出不变; 算法中隐含层神经元在分裂或删除的同时会进行相应的权值补偿, 也保证了网络输出的稳定性, 避免隐含层神经元贡献的不平衡. 在污水处理基准仿真平台 BSM1 上对 HCPS 算法进行验证, 结果表明, HCPS 算法能够获得更紧凑的网络结构, 用于出水氨氮浓度预测精度较高.

参考文献:

- [1] DU R, PENG Y, CAO S, et al. Advanced nitrogen removal from wastewater by combining anammox with partial denitrification [J]. *Biore-source Technology*, 2015, 179:497.
- [2] 杨航涛, 王双保, 王成龙. 基于分光光度法的水质氨氮检测系统设计[J]. *测控技术*, 2012, 31(5):49.
- [3] 乔俊飞, 安茹, 韩红桂. 基于 RBF 神经网络的出水氨氮预测研究[J]. *控制工程*, 2016, 23(9):1301.
- [4] ESFE M H, AFRAND M, WONGWISES S, et al. Applications of feedforward multilayer perceptron artificial neural networks and empirical correlation for prediction of thermal conductivity of $Mg(OH)_2$ -EG using experimental data [J]. *International Communications in Heat and Mass Transfer*, 2015, 67(4):46.
- [5] ZHANG Y D, WANG S H, JI G L, et al. Fruit classification using computer vision and feedforward neural network[J]. *Journal of Food Engineering*, 2014, 143(6):167.
- [6] FARAJZADEH J, FARD A F, LOTFI S. Modeling of monthly rainfall and runoff of Urmia lake basin using "feed-forward neural network" and "time series analysis" model [J]. *Water Resources and Industry*, 2014(7):38.
- [7] OONG T H, ISA N A. Adaptive evolutionary artificial neural networks for pattern classification [J]. *IEEE Trans Neural Netw*, 2011, 22(11):1823.
- [8] WANG J H, WANG H Y, CHEN Y L, et al. A constructive algorithm for unsupervised learning with incremental neural network[J]. *Journal of Applied Research & Technology*, 2015, 13(2):188.
- [9] PLATT J. A resource-allocating network for function interpolation [J]. *Neural Computation*, 1991, 3(2):213.
- [10] LAURET P, FOCK E, MARA T A. A node pruning algorithm based on a Fourier amplitude sensitivity test method [J]. *IEEE Transactions on Neural Networks*, 2006, 17(2):273.
- [11] ISLAM M M, SATTAR M A, AMIN M F, et al. A new adaptive merging and growing algorithm for designing artificial neural networks [J]. *IEEE Transactions on Systems Man and Cybernetics Part B*, 2009, 39(3):705.
- [12] HAN H G, QIAO J F. A structure optimisation algorithm for feedforward neural network construction [J]. *Neurocomputing*, 2013, 99:347.
- [13] GUO W, WEI H, ZHAO J, et al. Theoretical and numerical analysis of learning dynamics near singularity in multilayer perceptrons [J]. *Neurocomputing*, 2015, 151:390.
- [14] HACINE-GHARBI A, RAVIER P, HARBA R, et al. Low bias histogram-based estimation of mutual information for feature selection [J]. *Pattern Recognition Letters*, 2012, 33(10):1302.
- [15] GUO Y M, LIU Y G, ZENG G M, et al. An integrated treatment of domestic wastewater using sequencing batch biofilm reactor combined with vertical flow constructed wetland and its artificial neural network simulation study [J]. *Ecological Engineering*, 2014, 64:18.
- [16] MIRBAGHERI B, EHTESHAMI M, BAGHERI Z. Modeling of a sequencing batch reactor treating municipal wastewater using multi-layer perceptron and radial basis function artificial neural networks [J]. *Process Safety and Environmental Protection*, 2015, 93:111.
- [17] VIVIANO G, SALERNO F, CHIARA E, et al. Surrogate measures for providing high frequency estimates of total phosphorus concentrations in urban watersheds [J]. *Water Research*, 2014, 64:265.

- tions on Power Electronics, 2000, 15(4):778.
- [11] RACITI A, MUSUMECI S, CRISTALDI D. Modeling and simulation of IGBT thermal behavior during a short circuit power pulse [C] // 2015 International Conference on Clean Electrical Power (ICCEP), Taormina; IEEE, 2015: 542 - 547.
- [12] CLEMENTE S. Transient thermal response of power semiconductors to short power pulses [J]. IEEE Transactions on Power Electronics, 1993, 8(4):337.
- [13] 唐云宇, 林燎源, 马皓. 一种改进的并联 IGBT 模块瞬态电热模型 [J]. 电工技术学报, 2017, 32(12):88.
- [14] 唐勇, 汪波, 陈明. IGBT 开关瞬态的温度特性与电热仿真模型 [J]. 电工技术学报, 2012, 27(12):146.
- [15] SANO K, HAYASHI K, KAWAFUJI H, et al. Ultra-small compact transfer molded package for power modules [C] // 2008 58th Electronic Components and Technology Conference, Lake Buena Vista; IEEE, 2008:1832.
-
- (上接第 100 页)
- [18] JEPPSSON U, PONS M N. The COST benchmark simulation model-current state and future perspective [J]. Control Engineering Practice, 2004, 12(3):299.
- [19] HAN H G, QIAO J F. A structure optimization algorithm for feedforward neural network construction [J]. Neurocomputing, 2013, 1(99):347.
- [20] CHEN C, YAN X. Optimization of a multilayer neural network by using minimal redundancy maximal relevance-partial mutual information clustering with least square regression [J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(6):1177.