
БИЗНЕС-ИНФОРМАТИКА

УДК [004.8.032.26:004.45]:[621.182.12:628.16.048]

НЕЙРОСЕТЕВАЯ СИСТЕМА УПРАВЛЕНИЯ ВАКУУМНЫМ ДЕАЭРАТОРОМ ВД-400

П.П. Алексеев, И.А. Щербатов

Астраханский государственный технический университет

E-mail: pavel_alekseev22@mail.ru, sherbatov2004@mail.ru

В статье рассматриваются алгоритмы, применяемые в программной реализации нейросетевой системы управления концентрацией кислорода деаэрированной воды. Выполнен анализ вакуумного деаэратора ВД-400 как объекта управления. Выявлены регулирующие и возмущающие воздействия, а также регулируемые параметры ВД-400. Особое внимание уделено программному обеспечению создающего модели многослойных нейронных сетей прямого распространения и обучающего их в режиме реального времени. Демонстрируются интерфейсы обучения и моделирования. Приведена общая схема управления вакуумным деаэратором ВД-400.

Ключевые слова: вакуумный деаэратор, многослойные нейронные сети, обучение и моделирование нейронных сетей, линейная функция.

NEURAL NETWORK CONTROL SYSTEM OF VD-400 VACUUM DEAERATOR

P.P. Alekseev, I.A. Sherbatov

Astrakhan State Technical University

E-mail: pavel_alekseev22@mail.ru, sherbatov2004@mail.ru

The article considers the algorithms, applied in program realization of a neural network control system of concentration of oxygen of deaerated water. The analysis of the VD-400 vacuum deaerator as the object of control is carried out. Control and perturbation actions, as well as adjustable parameters of VD-400 are revealed. The special attention is paid to the software, creating models of the feedforward multilayered neural networks with live training. Interfaces of training and modeling are shown. The general scheme of control of the VD-400 vacuum deaerator is presented.

Keywords: vacuum deaerator, multilayered neural networks, training and modeling of neural networks, linear function.

ВВЕДЕНИЕ

Методы искусственного интеллекта эффективно применяются в различных отраслях промышленности [1]. Значительное число работ посвящено теории нечетких множеств и нечеткой логики [2]. Системы, основанные на знаниях, являются одним из ключевых направлений исследований в

области искусственного интеллекта. Возникающие на практике сложные задачи решаются с использованием знаний экспертов [4]. Эта область искусственного интеллекта включает в себя алгоритмы, модели и методы, направленные на автоматическое накопление и формирование знаний, используя процедуры анализа и обобщения данных. Искусственные нейронные сети прочно встраиваются в структуру автоматизированных систем управления промышленными предприятиями, робототехническими комплексами [3] и пр., с помощью которых решаются задачи автоматизации и управления, где стандартные локальные средства автоматизации оказываются малоэффективными. Таким образом, в ряде случаев использование систем искусственного интеллекта позволяет повысить качество управления и эффективность функционирования технологического оборудования.

В качестве объекта исследования выступает вакуумный деаэратор ВД-400 (ООО «ЛУКОЙЛ-Астраханьэнерго» подразделения Астраханская ТЭЦ-2), предназначенный для удаления коррозионно-агрессивных газов из подпиточной воды энергетических котлов. В соответствии с ГОСТ 16860–77 ВД-400 должен обеспечить средний подогрев воды на величину от 15 до 25 °С при изменении производительности в деаэраторе от 30 до 120 % от номинальной (содержание кислорода в деаэрированной воде не должно превышать 30 мкг/кг, свободная углекислота должна отсутствовать).

В настоящее время в системе управления процесса химводоподготовки котельной ТЭЦ-2 (г. Астрахань) применяется локальная автоматика на основе аналоговых технических средств. Технологический процесс рассматривается как набор обособленных параметров, управление которыми осуществляется локальными системами автоматического регулирования. На Дзержинской ТЭЦ разработан на базе программно-технического комплекса Teleperm проект системы управления котла-утилизатора фирмы «Siemens A&G», в котором согласно регламенту предусмотрены две одно-контурные схемы регулирования, не связанные между собой.

На основании результатов анализа существующих аналогов САУ, современного состояния автоматизации технологических процессов регулирования деаэрации питательной воды можно сделать следующие выводы:

1. Показания прибора, определяющего концентрацию кислорода, не используются в качестве регулируемой величины, а применяемые системы управления обеспечивают стабилизацию только косвенных величин: $T_{\text{вх}2}$ (температура питательной воды перед деаэратором) и $T_{\text{вых}}$ (температура питательной воды после деаэратора).

2. Ни одна из анализируемых систем не использует нейросетевое управление, хотя имеется возможность провести модернизацию производственных объектов и применить современные управляющие системы, позволяющие повысить эксплуатационные характеристики действующих технологических установок.

3. Создание высокоэффективной нейросетевой системы управления процессом деаэрации питательной воды, в частности регулирования температуры химически обессоленной воды и давления питательной воды, является задачей актуальной и имеет практическое значение.

ПОСТАНОВКА ЗАДАЧИ

Предметом данного исследования являются алгоритмы работы нейросетевой системы управления концентрацией кислорода деаэрированной воды вакуумного деаэратора ВД-400, а также их программная реализация.

Целью данного исследования является повышение эффективности управления вакуумным деаэратором за счет разработки программного обеспечения, главные функции которого будут заключаться в регулировании температуры химически обессоленной воды и давления питательной воды. Выбор параметров производится с помощью моделей многослойных нейронных сетей прямого распространения и обучении их в режиме реального времени. В качестве функций активации нейронов сети используются линейная функция $f(u) = u$ и гиперболический тангенс $f(u) = (e^u - e^{-u}) / (e^u + e^{-u})$.

ПРИМЕНЕНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Существует два вида реализации искусственных нейронных сетей: аппаратный и программный. Аппаратная нейронная сеть несколько выигрывает у программной по автономности и скорости работы, но ее сложно интегрировать в уже существующую автоматизированную систему управления.

За несколько десятков лет программных разработок нейронных сетей, подобрать универсальную библиотеку программ, которые позволят не только моделировать, но и обучать многослойные сети прямого распространения довольно сложно. На сегодняшний день самой распространенной средой моделирования нейронных сетей является Matlab Neural Network Toolbox. Однако нейронные сети, реализованные в Matlab, достаточно объемны и в ряде случаев не обеспечивают решение задач управления с требуемым быстродействием [6]. Кроме того, применение Matlab в режиме реального времени сопряжено с рядом сложностей. Учет этих факторов обуславливает потребность в интегрировании комплекса нейронных сетей в функционирующую систему управления, в целях повышения ее адаптивных свойств. Это становится возможным путем разработки программного обеспечения с условным названием SNN (System Neural Network), главные функции которого заключаются в создании моделей многослойных нейронных сетей прямого распространения и обучении их в режиме реального времени. Язык программирования – C++, достаточно легко портируемый на различные платформы. Эффективность синтезированных алгоритмов – соотношение времени работы и объема потребляемых ресурсов. Поэтому при одинаковых показателях эффективности предпочтение будет отдаваться алгоритмам наиболее простым в реализации [7].

АНАЛИЗ ОБЪЕКТА И РАЗРАБОТКА СТРУКТУРЫ СИСТЕМЫ УПРАВЛЕНИЯ

Предлагаемая схема системы управления вакуумным деаэратором (рис. 1) состоит из двух уровней: 1-й уровень АСУ – локальная сеть контроллеров, 2-й уровень АСУ – рабочая станция. 1-й уровень АСУ – локаль-

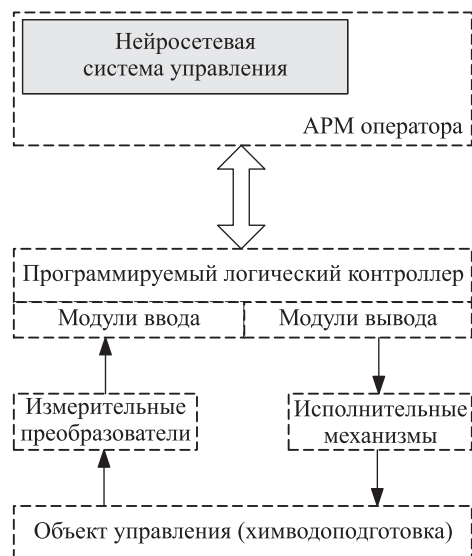


Рис. 1. Структурная схема разрабатываемой системы управления

ная сеть контроллеров представляет собой процесс химводоподготовки, включая ВД-400, технологические параметры которых измерительными приборами через модули ввода передаются на программируемый логический контроллер S7-1200. В свою очередь контроллер через модули вывода исполнительными механизмами управляет регулирующей и запорной арматурами. По линии связи данные передаются на 2-й уровень АСУ – рабочую станцию (ЭВМ). По стандарту физического уровня для асинхронного интерфейса, данные регламентируются электрическими параметрами дифференциальной линии связи типа «общая шина» RS-485 в систему управления деаэратором.

Рассматривая вакуумный деаэратор как объект управления, можно выделить следующие воздействия и параметры (рис. 2). **Регулирующие воздействия:** $F_{\text{п}}$ – расход пара на вакуумный деаэратор; $P_{\text{п}}$ – давление пара.

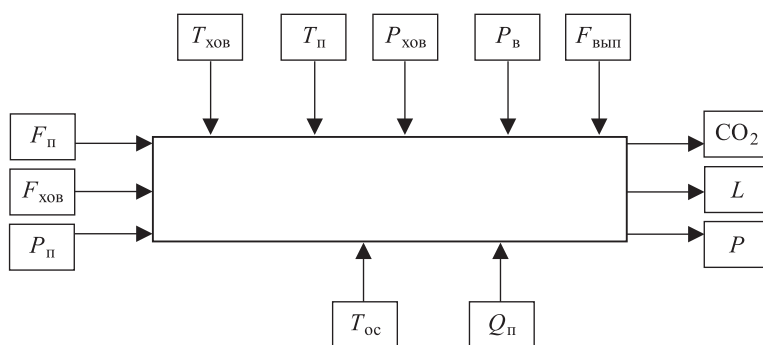


Рис. 2. Структура существующей системы управления на АТЭЦ-2

Регулируемые параметры: CO_2 – концентрация деаэрированной воды из вакуумного деаэратора; L – уровень деаэрированной воды в баке сбора деаэрированной воды; P – давление в вакуумном деаэраторе.

Возмущающие воздействия: **Контролируемые:** $T_{\text{хов}}$ – температура химически очищенной воды; $T_{\text{п}}$ – температура пара; $P_{\text{в}}$ – давление вакуума; $F_{\text{вып}}$ – расход выпара; $P_{\text{хов}}$ – давление химически очищенной воды. **Неконтролируемые:** $T_{\text{ос}}$ – температура окружающей среды; $Q_{\text{п}}$ – тепловые потери.

Далее по входным и выходным величинам можно выбрать структуру нейронной сети. Известно, что при моделировании любого процесса достаточно нейронной сети с одним скрытым слоем (рис. 3), с необходимым количеством нейронов равным $2 \cdot n + 1$ [5].

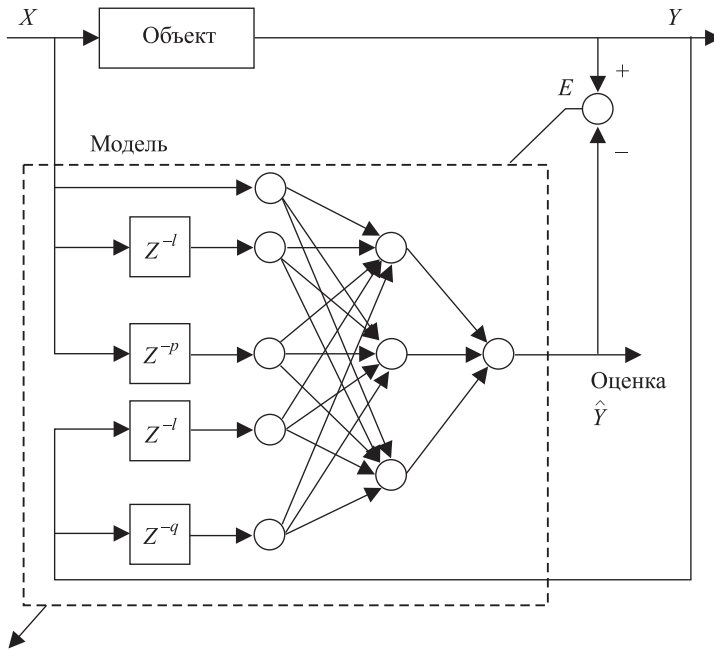


Рис. 3. Схема обучения нейросетевой модели

Перед настройкой нейросетевой модели необходимо выполнить фильтрацию измерений, для этого (в режиме реального времени) предлагается использовать рекуррентный метод наименьших квадратов, а также метод сингулярного стохастического анализа [8].

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НЕЙРОННЫХ СЕТЕЙ ПРЯМОГО РАСПРОСТРАНЕНИЯ

Важное место в системе управления занимает программное обеспечение, позволяющее создавать модели нейронных сетей. Класс нейронных сетей CNeuralNetwork является основным классом библиотеки программного обеспечения SNN. Он реализует нейронные сети разнообразной структуры, их работу и обучение.

Класс нейронных сетей имеет два варианта конструкторов:

1) Создание нейронной сети непосредственно через передачу в нее всех необходимых параметров в виде переменных и массивов.

```
int LS[4]={2,5,5,1},bias[3]={1,1,1};
CTransferFunction *TF[3]={tansig,tansig,purelin};
//1 – минимальное значение для входа сети
//-1 – максимальное значение для входа сети
//LS – массив с количеством нейронов в слоях
//bias – массив, определяющий наличие весов смещения
//TF – массив указателей на передаточные функции
CNeuralNetwork *net=new CNeuralNetwork(-1,1,3,LS,bias,TF);
```

2) Метод SaveToFile, Метод SaveToFile, записывающий структуру в текстовый файл, позволит сохранять и использовать настроенные нейронные сети.

Алгоритм обучения RPROP (resilient backpropagation) при решении поставленных задач выбран как наиболее удобный из эвристических алгоритмов. Его главное преимущество над другими методами, например, над методом сопряженных градиентов – упрощенная реализация при высокой скорости сходимости и небольших требованиях к погрешности вычисления градиента [9]. Поскольку алгоритм обучения RPROP обуславливается изменением знаков градиентов, для него нет необходимости в сложных вычислениях, нет зависимости от величин производных, индивидуально для каждого веса вычисляется приращение. Вычисление производится по следующей формуле:

$$\Delta_l^{(i)} = \begin{cases} \eta^+ \Delta_l^{(i-1)}, & \text{если } \frac{\partial E(w)}{\partial w_l}^{(i-1)} \frac{\partial E(w)}{\partial w_l}^{(i)} > 0, \\ \eta^- \Delta_l^{(i-1)}, & \text{если } \frac{\partial E(w)}{\partial w_l}^{(i-1)} \frac{\partial E(w)}{\partial w_l}^{(i)} < 0, \\ \Delta_l^{(i-1)}, & \text{если } \frac{\partial E(w)}{\partial w_l}^{(i-1)} \frac{\partial E(w)}{\partial w_l}^{(i)} = 0, \end{cases} \quad (1)$$

где $0 < \eta^- (= 0,5) < 1 < \eta^+ (= 1,2)$.

Величина приращения увеличивается при условии схождения алгоритма к минимуму, в результате чего производная не меняет знак. На плоских участках этот способ позволяет ускорить операцию и тормозит поиск, когда локальный минимум пропущен [11].

Значения весов изменяются в том же направлении, в котором убывает градиент:

$$\Delta w_l^{(i)} = \begin{cases} \Delta_l^{(i)} \operatorname{sgn} \left[\frac{\partial E(w)}{\partial w_l}^{(i)} \right], & \text{если } \frac{\partial E(w)}{\partial w_l}^{(i-1)} \frac{\partial E(w)}{\partial w_l}^{(i)} \geq 0, \\ -\Delta_l^{(i)}, & \text{если } \frac{\partial E(w)}{\partial w_l}^{(i-1)} \frac{\partial E(w)}{\partial w_l}^{(i)} < 0, \end{cases} \quad (2)$$

где $\operatorname{sgn}[*]$ – функция знака. В результате возврат к предыдущему значению веса $w_l^{(i-1)}$ выполняется, когда производная ошибки изменяет знак, что указывает на пропуск минимального значения. Однако при обнулении значения производной $\frac{\partial E(w)}{\partial w_l}^{(i)}$, изменение знака на следующем шаге можно избежать.

Достаточно объемным этапом обучения нейронной сети является нахождение градиента целевой функции [11]. Для генерации градиента ошибки обучения выбран метод сопряженных графов, позволяющий построить простые правила формирования компонентов градиента, которые имеют постоянную структуру, не зависящую от сложности сети. Это связано с тем, что формулы расчета градиента, взятые в явном виде от целевой функции, очень сложны и неудобны для практического применения, особенно в том случае, если сеть содержит больше одного скрытого слоя. Количество итераций обучения напрямую зависит от начальной инициализации весов [12]. Для упрощения работы начальные значения весов выбираются случайно

из некоторого промежутка. Предложенный метод Д. Нгуеном и Б. Уидроу позволяет в несколько раз сократить количество итераций обучения: с помощью большей приближенности начальных значений к оптимальным [10]. Суть метода заключается в значении весов, позволяющих кусочно-линейно аппроксимировать входной сигнал с помощью линейных участков функции активации. Для сигмоидальной функции активации линейная область расположена на интервале аргумента $(-1; 1)$. Следовательно, для случая слоя с одним входом и H выходами необходимо, чтобы выполнялось условие $(-1; 1)$:

$$-1 < w_i x + w_b < 1, \quad (3)$$

где x – равномерно распределенная величина из интервала $[x_{\min}, x_{\max}]$.

Приняв ширину интервала для всех нейронов одинаковой и в сумме заполняющей входной диапазон:

$$w_i = \frac{1}{(x_{\max} - x_{\min})} H, \quad (4)$$

получим значения весов смещения как равномерно распределенную величину на интервале

$$w_b \in [-x_{\max} w_i, -x_{\min} w_i]. \quad (5)$$

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ УПРАВЛЕНИЯ

Алгоритм управления состоит из повторяющихся вычислительных операций между моментами поступления измеряемой информации о переменных состояния объекта. Дадим словесное описание алгоритма.

1. Начало. Ожидаем поступления в i -й момент времени новой информации об измеряемых параметрах.

2. Выполняем фильтрацию полученных измерений с использованием предыдущих k значений измеряемых величин.

3. Проверяем на соответствие полученных оценок состояний объекта управления требуемым по графику-заданию. Если величина отклонения превышает заданное пороговое значение ε , то возникает необходимость в корректировке управляющего воздействия, иначе переходим к п. 1.

4. Проверяем величину времени, прошедшую с момента предыдущей корректировки управляющего воздействия. Если эта величина меньше некоторой фиксированной величины $\Delta\tau_z$, то переходим на п. 1.

5. С использованием нейросетевой модели находим новое управляющее воздействие, которое обеспечивает выход состояния объекта на заданный график пуска.

6. Найденное в п. 5 управляющее воздействие выдаем в качестве задания для регуляторов температуры химически обессоленной воды в вакуумном деаэраторе и давления питательной воды.

7. Переходим к п. 1 алгоритма. Конец.

Величина $\Delta\tau_z$ – время запаздывания реакции объекта управления на корректировку управляющего воздействия. Эта величина введена для того, чтобы обеспечить получение достоверной информации о завершении пере-

ходных процессов, вызванных новым управляющим воздействием, а также с целью ограничения частоты изменения задания для регуляторов температуры химически обессоленной воды в вакуумном деаэраторе и давления питательной воды, что требуется из соображений надежности и устойчивости работы комплекса «регулятор–объект управления». Величина $\Delta\tau_z$ определяется на основе анализа динамических характеристик объекта управления, полученных в результате экспериментальных исследований [11].

Обобщенная функциональная схема управления приведена на рис. 4. Она включает в себя первичную фильтрацию данных, так как результаты измерения довольно сильно зашумлены. Затем настройку нейросетевой модели с возможностью последующей настройки ее в процессе работы. После этого нейросетевую модель можно использовать для управления на основе прогнозирования.

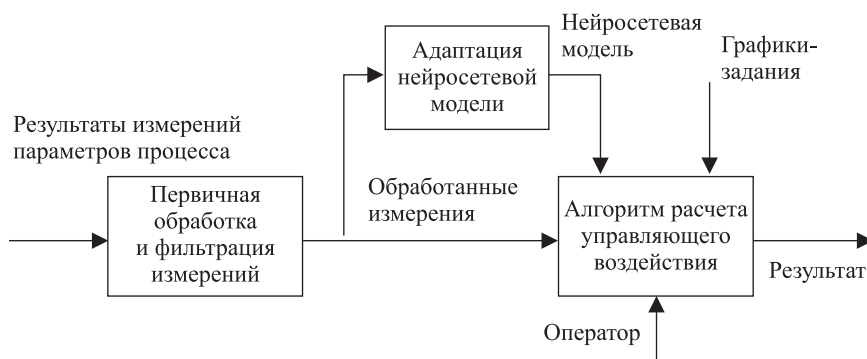


Рис. 4. Функциональная схема управления вакуумным деаэратором

ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

Для удобства использования библиотеки в среде программирования C++ Builder 6.0 был разработан графический интерфейс. Главное окно программы имеет две вкладки, позволяющие обучать и моделировать нейронные сети.

На вкладке обучения (рис. 5) пользователь может построить структуру нейросетевой модели, которую необходимо настраивать: число слоев нейронной сети, минимальное и максимальное значение входного сигнала, количество нейронов в каждом слое, указать, использовать ли веса смещения в модели нейрона. Далее пользователь должен указать файл с обучающим набором данных, файл для сохранения нейросетевой модели и длину обучающего набора данных, желаемую среднюю квадратичную погрешность обучения, максимально допустимое количество итераций. Для удобства использования программы с моделями различной размерности также можно указать число итераций, через которое программа будет показывать достигнутую погрешность обучения, если это число не указано, то вывод промежуточных результатов не производится.

На вкладке «Моделирование» (рис. 6) пользователь может использовать уже настроенную модель. Результаты моделирования – выходы нейронной сети будут записаны в указанный файл.

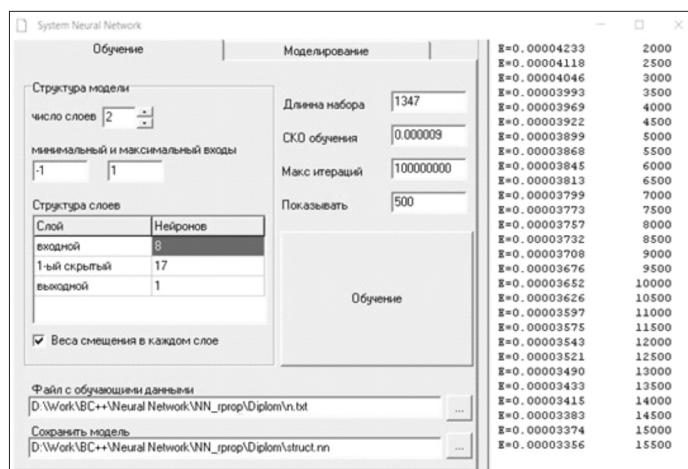


Рис. 5. Интерфейс обучения

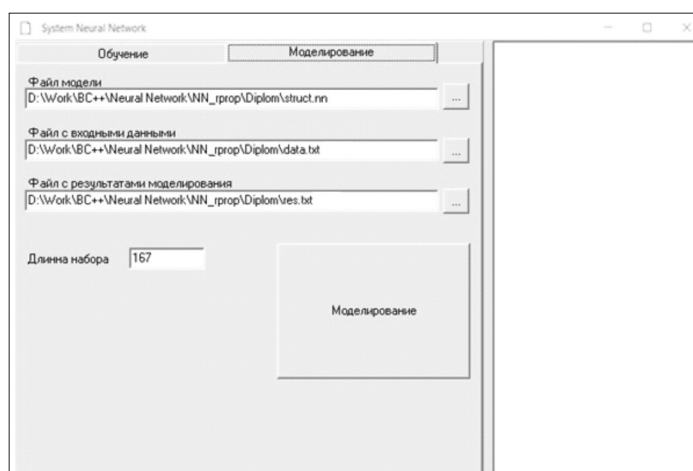


Рис. 6. Интерфейс моделирования

ТЕСТИРОВАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для независимого тестирования библиотеки программ SNN и Matlab Release 13 были предусмотрены следующие условия:

- 1) алгоритм инициализации – `initnw` (реализация метода Нгуена–Уидроу);
- 2) тестируемый алгоритм обучения – `trainrp`;
- 3) реализующий алгоритм – `RPROP`;
- 4) реализующий метод масштабируемых сопряженных градиентов – `trainscg`.

Для тестирования разработанного программного обеспечения применялся компьютер с установленной операционной системой Windows 10; технические характеристики: процессор Intel Core i7 2,3ГГц и 8Гб опера-

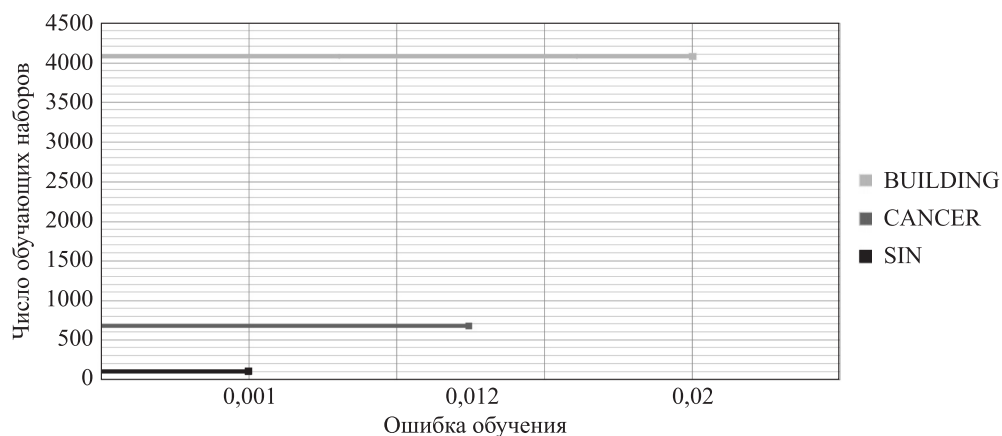


Рис. 7. Краткие характеристики тестовых наборов

тивной памяти. Бэнчмарки Sin (небольшая задача аппроксимации периода функции синуса), Cancer (задача средней размерности) и Building (задача большой размерности в зависимости от внешних условий) – выбраны в качестве обучающих множеств (рис. 7). С помощью них проведены 30 запусков обучения каждого из тестов. Результаты тестирования представлены в таблице.

Результаты тестирования SNN и Matlab

Программа	Среднее время обучения, с	Среднее число итераций	Среднее время одной итерации, с
SIN			
SNN	0,15	325	0,00040
Mathlab trainrp	2,1	586	0,0048
Mathlab trainscg	0,69	99	
CANCER			
SNN	30,02	3059	0,0098
Mathlab trainrp	30,18	2293	0,0131
Mathlab trainscg	29,89	935	
BUILDING			
SNN	174,75	684	0,252
Mathlab trainrp	381,24	1298	0,285
Mathlab trainscg	1080,35	1972	

При реализации информационно-управляющих систем в режиме реального времени главным условием сравнения программ является время обучения нейронных сетей. Помимо всего при сравнении SNN и trainrp по количеству итераций, есть возможность выявить наиболее оптимальную реализацию алгоритма RPROP, в плане выбора первичных значений и коэффициентов. В свою очередь, показать эффективность методов для задач различной размерности становится возможным при сравнении тестируемых алгоритмов обучения trainrp и trainscg.

При тесте SIN – небольшой задачи аппроксимации периода функции синуса, разработанное программное обеспечение показало более эффек-

тивные результаты по основным параметрам, а именно общему времени обучения и времени одной итерации (в сравнении с их количеством). Для задач малой размерности следует применять метод SCG, который использует вторые производные для более быстрой сходимости.

В бэнчмарке Canceg время обучения всех программ примерно одинаково, а лучшие результаты по количеству итераций в сравнении с SNN показала реализация метода RPROP в Matlab. Это объясняется существенно отличающейся от SNN реализацией алгоритма начальной инициализации сети, показывающей более высокие результаты для слоев с малым количеством нейронов [9].

Как показал тест в последнем бэнчмарке (Building), SNN сократило время обучения более чем в два раза за счет большей скорости вычислений и изменений в реализации RPROP. Этот бэнчмарк продемонстрировал свое превосходство над методом SCG для задач большой размерности RPROP.

ЗАКЛЮЧЕНИЕ

В качестве объекта исследования выступил вакуумный деаэратор ВД-400 (ООО «ЛУКОЙЛ-Астраханьэнерго» подразделения Астраханская ТЭЦ-2), предназначенный для удаления коррозионно-агрессивных газов из подпиточной воды энергетических котлов. Выполнен анализ деаэратора как объекта управления. Выявлены регулирующие воздействия (расход и давление пара), регулируемые параметры (концентрация, давление и уровень деаэрированной воды), а также возмущающие воздействия (контролируемые и неконтролируемые). По входным и выходным величинам выбрана структура нейронной сети, приведена методика ее создания. Алгоритм управления состоит из повторяющихся вычислительных операций между моментами поступления измеряемой информации о переменных состояния объекта. Приведено описание алгоритма в виде псевдокода.

Для генерации градиента ошибки обучения выбран метод сопряженных графов, позволяющий построить простые правила формирования компонентов градиента, которые имеют постоянную структуру, не зависящую от сложности сети. Это связано с тем, что формулы расчета градиента, взятые в явном виде от целевой функции, очень сложны и неудобны для практического применения, особенно в том случае, если сеть содержит больше одного скрытого слоя.

Разработано программное обеспечение, позволяющее использовать нейронные сети в системе управления, в котором для удобства использования библиотеки был реализован графический интерфейс. Главное окно программы имеет две вкладки, позволяющие обучать и моделировать нейросетевые модели. Приведена общая схема системы управления на основе нейросетевой модели.

Литература

1. *Бобырь М.В., Титов В.С.* Интеллектуальная система управления температурными деформациями при резании // Автоматизация. Современные технологии. 2011. № 5. С. 3–7.

2. *Бобырь М.В.* Диагностика оборудования с ЧПУ методами нечеткой логики // Промышленные АСУ и контроллеры. 2010. № 1. С. 18–20.
3. *Глазков В.П., Большаков А.А., Кулик А.А.* Использование нейросетевого компенсатора для стабилизации движения полуавтоматических протезируемых систем // Мехатроника, автоматизация, управление. 2014. № 1. С. 13–17.
4. *Немчинов Д.В., Проталинский О.М.* Снижение риска аварийной ситуации на производственном объекте // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2009. № 1. С. 111–116.
5. *Осовский С.* Нейронные сети для обработки информации / пер. с пол. М.: Финансы и статистика, 2002. 344 с.
6. *Проталинский О.М.* Проверка достоверности первичной информации в АСУ ТП с использованием нечетких множеств // Известия высших учебных заведений. Северо-Кавказский регион. Серия: Технические науки. 2003. № 3. С. 60.
7. *Проталинский О.М., Дианов Р.С.* Автоматизированная система управления разработкой газового месторождения с применением нейронной сети // Промышленные АСУ и контроллеры. 2003. № 12. С. 30.
8. *Скобцов Ю.А., Ткаченко А.В.* Программная реализация нейронных сетей для обучения в системах реального времени // Наукові праці Донецького державного технічного університету. Сер. обчислювальна техніка та автоматизація, випуск 66. Донецьк: ДонНТУ. 2005.
9. *Moller M.F.* 'A scaled conjugate gradient algorithm for fast supervised learning', Neural Networks. 1993. Vol. 6. P. 525–533.
10. *Nguyen D., Widrow B.* Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. Proceedings of the International Joint Conference on Neural Networks. 1990. Vol. 3. P. 21–26.
11. *Riedmiller M.* A direct method for faster backpropagation learning. Proceedings of the 1993 IEEE International Conference on Neural Networks (ICNN '93). San Francisco. Vol. 1. P. 586–591.
12. *Rumelhart D.E., Hinton G.E., Williams R.J.* Learning internal representations by error propagation // Parallel Distributed Processing: Exploration in the Microstructure of Cognition, D.E. Rumelhart and J.L. McClelland (Eds.). Cambridge, MA: MIT Press, 1986. Vol. 1. P. 318–362.

Bibliography

1. *Bobyry' M.V., Titov V.S.* Intel'ektual'naja sistema upravlenija temperaturnymi deformacijami pri rezanii // Avtomatizacija. Sovremennye tehnologii. 2011. № 5. P. 3–7.
2. *Bobyry' M.V.* Diagnostika oborudovanija s ChPU metodami nechetkoj logiki // Promyshlennye ASU i kontrollery. 2010. № 1. P. 18–20.
3. *Glazkov V.P., Bol'shakov A.A., Kulik A.A.* Ispol'zovanie nejrosetevogo kompensatora dlja stabilizacii dvizhenija poluavtomaticheskijh proteziruemyh sistem // Mehatronika, avtomatizacija, upravlenie. 2014. № 1. P. 13–17.
4. *Nemchinov D.V., Protalinskij O.M.* Snizhenie riska avarijnoj situacii na proizvodstvennom ob#ekte // Vestnik Astrahanskogo gosudarstvennogo tehničeskogo universiteta. Serija: Upravlenie, vychislitel'naja tehnika i informatika. 2009. № 1. P. 111–116.
5. *Osovskij S.* Nejrornyie seti dlja obrabotki informacii / per. s pol. M.: Finansy i statistika, 2002. 344 p.
6. *Protalinskij O.M.* Proverka dostovernosti pervichnoj informacii v ASU TP s ispol'zovanijem nechetkih mnozhestv // Izvestija vysshijh uchebnyh zavedenij. Severo-Kavkazskij region. Serija: Tehničeskije nauki. 2003. № 3. P. 60.
7. *Protalinskij O.M., Dianov R.S.* Avtomatizirovannaja sistema upravlenija razrabotkoj gazovogo mestorozhdenija s primenenijem nejronnoj seti // Promyshlennye ASU i kontrollery. 2003. № 12. P. 30.

8. *Skobcov Ju.A., Tkachenko A.V.* Programmna realizacija nejronnyh setej dlja obuchenija v sistemah real'nogo vremeni // Naukovi praci Donec'kogo derzhavnogo tehničnogo universitetu. Ser. obchisljuval'na tehnika ta avtomatizacija, vipusk 66. Donec'k: DonNTU. 2005.
9. *Moller M.F.* 'A scaled conjugate gradient algorithm for fast supervised learning', *Neural Networks*. 1993. Vol. 6. P. 525–533.
10. *Nguyen D., Widrow B.* Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *Proceedings of the International Joint Conference on Neural Networks*. 1990. Vol. 3. P. 21–26.
11. *Riedmiller M.* A direct method for faster backpropagation learning. *Proceedings of the 1993 IEEE International Conference on Neural Networks (ICNN '93)*. San Francisco. Vol. 1. P. 586–591.
12. *Rumelhart D.E., Hinton G.E., Williams R.J.* Learning internal representations by error propagation // *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, D.E. Rumelhart and J.L. McClelland (Eds.). Cambridge, MA: MIT Press, 1986. Vol. 1. P. 318–362.