

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

А.А. Бородкин, В.Л. Елисеев

**ОСНОВЫ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ
И ИХ ПРИМЕНЕНИЕ**

СБОРНИК ЛАБОРАТОРНЫХ РАБОТ

Методическое пособие
по курсу
«Нейрокомпьютеры и их применение»
для студентов, обучающихся по направлению
«Управление в технических системах»

УДК 621.398

Утверждено учебным управлением МЭИ

Подготовлено на кафедре управления и информатики

Рецензент: д.т.н., проф. Г.Ф. Филаретов

Бородкин А.А., Елисеев В.Л.

Основы и применение искусственных нейронных сетей.
Сборник лабораторных работ: методическое пособие / А.А.
Бородкин, В.Л. Елисеев. – М.: Издательский дом МЭИ, 2017. – 48 с.

Содержит описания восьми лабораторных работ по курсу
«Нейрокомпьютеры и их применение» и методические указания по их
выполнению. Пособие предназначено для студентов АВТИ, обучающихся по
направлению «Управление в технических системах».

Учебное издание

Бородкин Артем Александрович,
Елисеев Владимир Леонидович.

ОСНОВЫ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ И ИХ ПРИМЕНЕНИЕ

Сборник лабораторных работ

Методическое пособие по курсу «Нейрокомпьютеры и их применение»
для студентов, обучающихся по направлению «Управление в технических системах»

Редактор
Редактор издательства

Темплан издания МЭИ 2017/2018, метод.

Печать офсетная

Тираж 100 экз.

Формат 60х84/16

Изд. №

Подписано к печати . 2017 г.

Физ. печ. л. 3

Заказ №

ЗАО «Издательский дом МЭИ», 111250, Москва, Красноказарменная, д. 14
Отпечатано в типографии

© Национальный исследовательский
университет «МЭИ», 2017

ВВЕДЕНИЕ

В процессе изучения поведенческих актов живых существ в начале XX века исследователи стали сопоставлять зримые эффекты с активностью в нервных структурах. Удалось установить связь некоторых дисфункций организма с характерными особенностями функционирования нервных клеток. Получили объяснение эффекты действия электрического тока на живую материю. Все это привело к всплеску интереса к биологическим нервным сетям. Стало очевидно, что феномен человеческого интеллекта, такие функции мозга как память, способность к обучению и обобщению реализуются громадным числом нейронов, связанных друг с другом.

Тогда же, в 30-е годы, были опубликованы первые труды по кибернетике, науке об управлении. Часть исследователей занялась изучением информационных процессов в биологических нейронных сетях. Проводить опыты на биологических нейронах оказалось чрезвычайно сложно в силу трехмерности их расположения в живой материи и малых размеров клеток. Поэтому распространение получили разнообразные формальные модели. Предмет исследования стал называться искусственные нейронные сети.

Можно выделить несколько компонент искусственной нейронной сети, определяющих ее свойства:

- Нейроны, образующие сеть.
- Архитектура нейронной сети.
- Метод обучения нейронной сети.

Во многом эти составляющие взаимосвязаны. В частности, метод обучения неявно подразумевает некоторую архитектуру связи нейронов и их свойства. Поэтому, несмотря на наличие в литературе некоторой номенклатуры компонент, рассматриваются они, как правило, совместно. Совокупность некоторых компонент и правило построения из них нейросети называется нейросетевой парадигмой.

На конец XX века в мире были известны более четырех десятков различных нейросетевых парадигм и множество их разновидностей. В последние годы обнаруживается значительный рост интереса к искусственным нейронным сетям благодаря возросшим возможностям вычислительной техники. Нейронные сети успешно применяются для решения практических задач в различных прикладных областях.

Теоретические основы

Искусственный нейрон

Основным элементом искусственной нейронной сети является формальный нейрон (рис. 1), выполняющий нелинейное преобразование

входных сигналов. В первом приближении формальный нейрон имитирует свойства биологического нейрона.

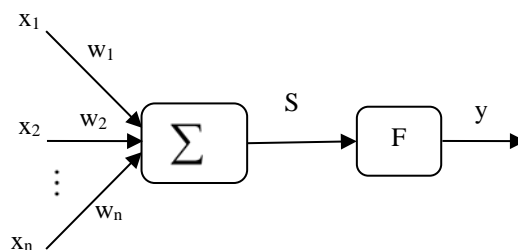


Рисунок 1: Формальный нейрон

На вход формального нейрона поступает некоторое множество сигналов x_1, x_2, \dots, x_n , каждый из которых может являться выходом другого нейрона. Эти входные сигналы, в совокупности обозначаемые вектором X , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес w_1, w_2, \dots, w_n , и поступает на суммирующий блок, обозначенный Σ . Каждый вес соответствует «силе» одной биологической синаптической связи. (Множество весов в совокупности обозначается вектором W .) Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, обозначенный как S . В векторных обозначениях это может быть записано следующим образом: $S = XW$. Сигнал S далее преобразуется активационной функцией F и дает выходной нейронный сигнал y :

$$y = F(S) = F\left(\sum_{j=1}^n w_j x_j\right) \quad (1)$$

Функция, используемая в качестве активационной, может быть различной и определяется решаемой задачей и типом нейронной сети. Наиболее распространенные функции изображены на рис.2 и в таблице 1.

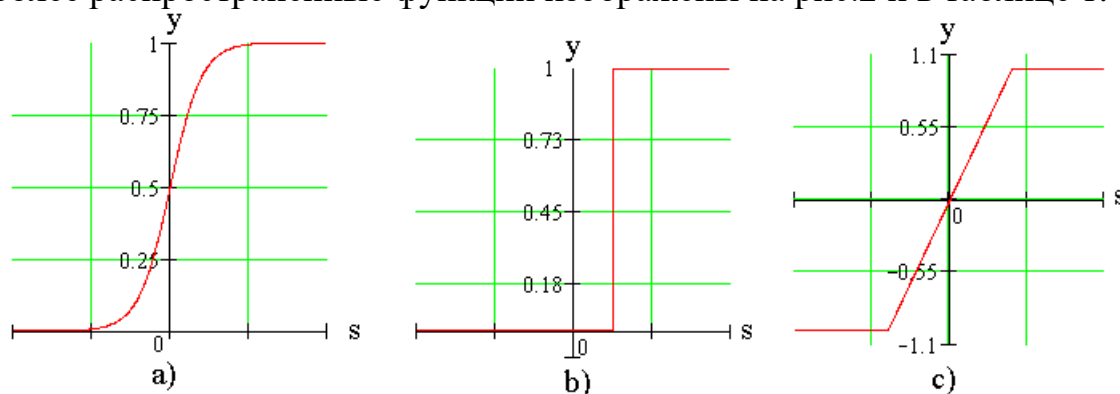


Рис.2. Активационные функции: а) Сигмоидная (логистическая), б) пороговая, с) линейная ограниченная.

Таблица 1. Функции активации

| Название функции | $y=F(S)$ | $F'(S)$ | Параметры |
|----------------------------|--|------------------------|---|
| Линейная | kS | k | $k=const$, коэффициент наклона прямой |
| Пороговая | $\begin{cases} 1, S > 0 \\ 0, S \leq 0 \end{cases}$ | | |
| Линейная ограниченная | $\begin{cases} p, S > 0 \\ -p, S < -\alpha \\ S, -\alpha \leq S \leq \alpha \end{cases}$ | | |
| Модифицированная пороговая | $y(t+1) = \begin{cases} 1, S > 0 \\ y(t), S = 0 \\ -1, S < 0 \end{cases}$ | | t -параметр времени |
| Сигмоидная | $\frac{1}{1 + e^{-cS}}$ | $y(1 - y)$ | $c > 0$, характеризует ширину функции по оси абсцисс |
| Биполярная сигмоидная | $\frac{2}{1 + e^{-S}} - 1$ | $\frac{1}{2}(1 - y^2)$ | |
| Гиперболический тангенс | $th(cS) = \frac{e^{cS} - e^{-cS}}{e^{cS} + e^{-cS}}$ | $(1 - y^2)$ | $c > 0$, характеризует ширину функции по оси абсцисс |
| Радиально-базисная | $\exp(-\frac{0.5S^2}{2\sigma^2})$ | | σ -среднеквадратическое отклонение, характеризующее ширину функции |

Следует отметить, что в некоторых случаях, например, для радиально-базисной функции, величина S может рассчитываться как евклидово расстояние между входным и весовым векторами:

$$S = (\sum_{j=1}^n (x_j - w_j)^2)^{0.5} \quad (2)$$

Архитектура искусственной нейронной сети

В биологических нейронных сетях имеют место разнообразные соединения нервных клеток. В зависимости от типа клеток и их функций связи могут быть как локальным (в пределах нескольких миллиметров), так и весьма протяженными (десятки сантиметров). Некоторые нервные клетки соединяются друг с другом вполне регулярно (например, глазной нерв), другие же связаны на первый взгляд хаотично. В живом организме связи между нейронами могут исчезать, могут появляться, вырастать. Механизм и причины такого свойства связей во многом неизвестен.

В искусственных нейронных сетях используются гораздо более простые схемы связей между нейронами. Во-первых, это группирование нейронных элементов в виде слоев. Нейронный слой – это линейка или матрица, обычно, однотипных нейронов, на которые в каждый такт времени параллельно поступают сигналы от других нейронов сети. Правила соединения между слоями, а также внутри слоя формализованы.

Рисунок 33 демонстрирует два примера связи нейронных слоев. Вариант (а) соответствует полносвязному соединению входов нейронов 2-го слоя с выходами нейронов 1-го слоя. Вариант (б) демонстрирует наличие области связи размером 3: выходы только трех ближайших нейронов слоя №1 подаются на вход нейрона слоя №2. Область связи используется в том случае, когда необходимо гарантировать отсутствие влияния далеко расположенных нейронов предыдущего слоя.

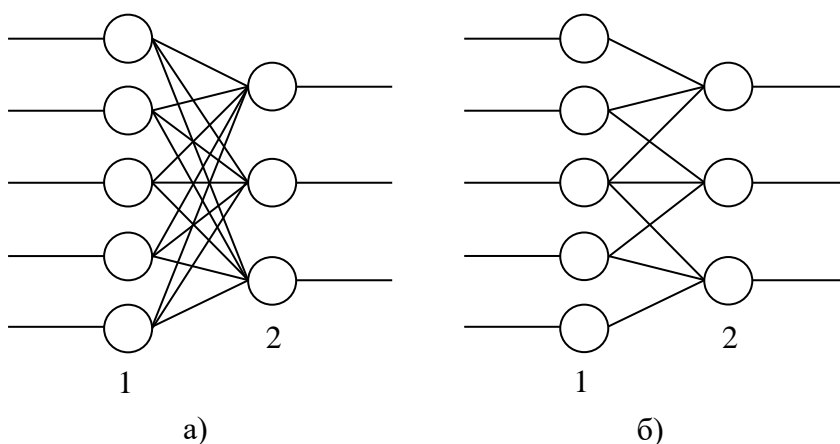


Рисунок 3: Соединение нейронных слоев

Во-вторых, все возможные связи в искусственных нейронных сетях определяются заранее. Если же в процессе обучения оказалось, что некоторые связи не нужны для получения желаемого выхода, им присваивается вес 0, что равносильно исчезновению соединения между двумя нервными клетками в организме.

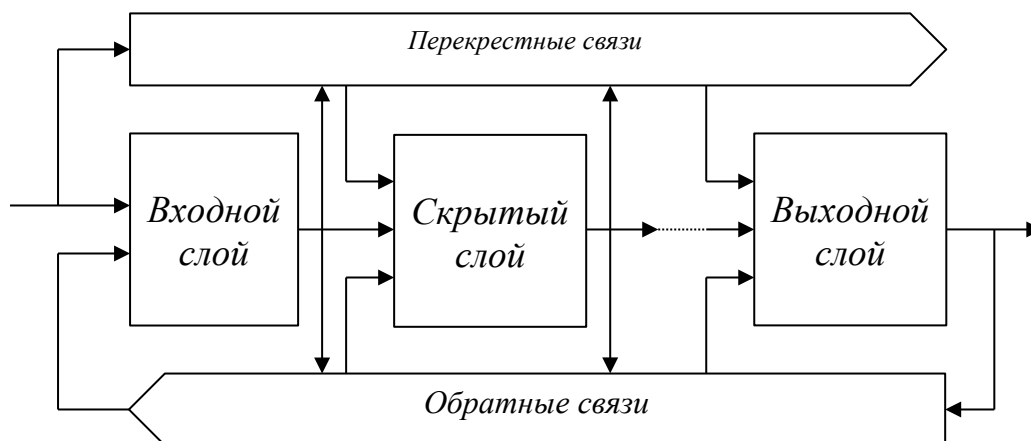


Рисунок 4: Обобщенная архитектура нейронной сети

Достаточно общая модель архитектуры искусственных нейронных сетей представлена в работе [1] (Рисунок 44). Она является многослойной и включает в себя прямые, перекрестные и обратные связи между слоями.

Входной слой обычно состоит из нейронов с линейной функцией активации и выполняет распределительные функции. Скрытые слои осуществляют функции обработки информации. Выходной слой также служит для обработки информации от предыдущих слоев и формирует результат работы нейронной сети.

Примеры архитектур для решения типовых задач

На рис. 6 изображено несколько типовых архитектур нейронных сетей. С помощью многослойного персептрона (рис.6а) можно решать задачи классификации.

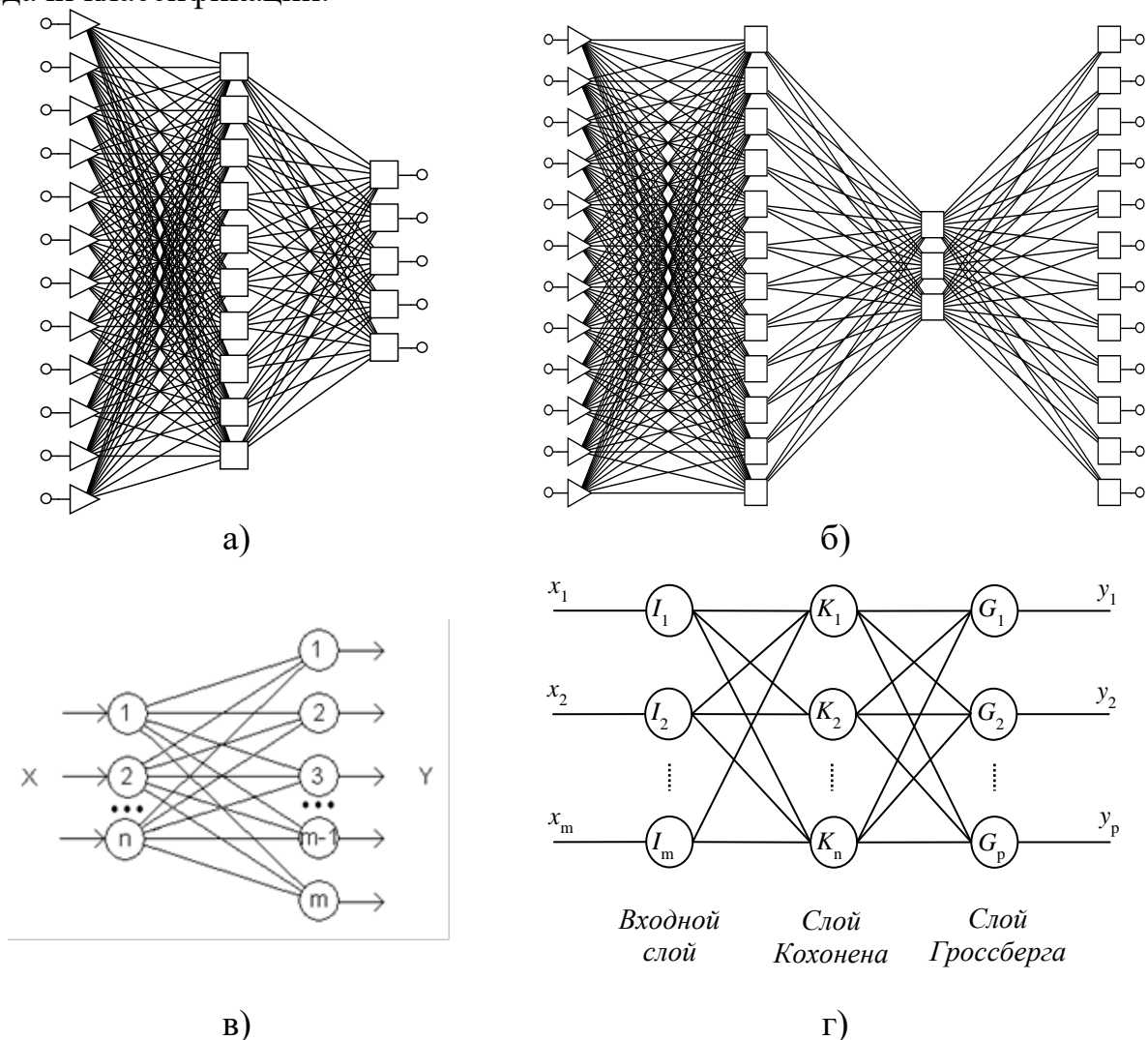


Рисунок 6: Типовые архитектуры нейронных сетей: а) многослойный персептрон; б) автоассоциативная сеть; в) сеть Кохонена; г) сеть встречного распространения сигнала.

Автоассоциативная сеть (рис.6б) позволяет сжимать данные и производить одноклассовую классификацию. Нейронная сеть Кохонена (рис.6в) осуществляет топологическое упорядочивание входного пространства данных и широко применяется в задачах распознавания образов. Сеть Кохонена осуществляет разбиение n -мерного пространства

на m областей решений, каждой из которых соответствует отдельный нейрон. Сеть встречного распространения позволяет решать задачи кластеризации и классификации (рис.6г).

Методы обучения

Под обучением нейронной сети обычно понимается настройка весовых коэффициентов (или других характерных параметров) нейронов, обеспечивающая решение нейронной сетью в режиме функционирования поставленной задачи с заданной точностью. До начала обучения необходимо полностью задать архитектуру нейронной сети, то есть выбрать количество нейронных слоев и нейронов в них, тип нейронов, вид функций активации, установить соединения между слоями и нейронами, определить наличие латеральных связей, задать размеры рецептивных областей.

В зависимости от специфики прикладной области обучение нейронной сети может производиться однажды, эпизодически или постоянно. Первый случай имеет место, когда нейронная сеть предназначена для работы в неизменных условиях. Второй случай соответствует плавно или редко меняющимся внешним условиям, когда необходимость дообучения определяется некоторым критерием. В третьем случае речь идет об адаптивной системе, обучающейся в процессе функционирования.

Можно выделить три подхода к обучению искусственных нейронных сетей:

- аналитическое обучение;
- обучение с учителем;
- обучение без учителя.

Суть аналитического *обучения* сводится к построению нейросетевого алгоритма, оптимизирующего качество функционирования нейронной сети для решения конкретной задачи [2][3]. В некоторых случаях удастся непосредственно из постановки задачи сформулировать алгоритм расчета весовых коэффициентов. Обучение как таковое сводится к заданию рассчитанных весовых коэффициентов [3]. Очевидно, что данный подход применим не всегда. С одной стороны, в процессе проектирования нейронной сети инженер получает оценки точности и предсказуемости функционирования целевой нейронной сети. С другой стороны, детальный анализ предметной области перекладывает интеллектуальную нагрузку выявления закономерностей с нейронной сети на плечи инженера, что, вообще говоря, является отрицанием самой идеи использования нейронных сетей в качестве носителей искусственного интеллекта.

Методы обучения с учителем

Обучение с учителем подразумевает наличие априорных знаний о входных и соответствующих им выходных данных. Как правило, эти данные называются обучающим набором и представлены парами вход – желаемый выход. Аналогичный способ обучения применяется и в биологических системах.

К достоинствам обучения с учителем можно отнести неявную передачу знаний о предметной области в нейросеть (без предварительной детальной формализации человеком), универсальность подхода и возможность учиться как на модельных, так и на реальных данных. Основным недостатком этого подхода к обучению является необходимость знания выходных данных, которые в ряде ситуаций получить сложно или даже невозможно (например, в задаче управления атомным реактором в аварийном режиме). Выходные данные, полученные на модели, могут быть неточными в силу приблизительности самой модели.

В подходе обучения с учителем обычно используют:

- метод обратного распространения ошибки
- квазиньютоновский метод
- метод Левенберга-Маркварда
- метод сопряженных градиентов

Метод обратного распространения ошибки [4][5] применяется для обучения многослойных сетей без перекрестных и обратных связей. Обучение сводится к минимизации среднеквадратичной ошибки нейронной сети с помощью метода градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети:

$$\min_{w_{ij}, T_j} \frac{1}{P} \sum_{p=1}^P E_p \quad E_p = \frac{1}{2} \sum_{j=1}^{N_L} (y_j - t_j)^2,$$

где N_L – количество нейронов в последнем L-ом слое нейронной сети;
 y_j – выход j-го нейрона последнего слоя; t_j – эталонное значение выхода j-го нейрона последнего слоя; E_p – среднеквадратичная ошибка нейронной сети при подаче на вход p-го входного образа из обучающего набора данных.

$$y_j = F(S_j) = F\left(\sum_{i=1}^n w_{ij} y_i - T_j\right), \quad (3)$$

здесь y_i – выход i-го нейрона предпоследнего слоя; w_{ij} – весовой коэффициент связи между i-м нейроном предпоследнего слоя и j-м нейроном последнего слоя. T_j – пороговое значение j-го нейрона выходного слоя, определяющее сдвиг функции активации по оси абсцисс; n – количество нейронов в предпоследнем слое.

Весовые коэффициенты и пороговые значения в процессе обучения изменяются согласно выражениям:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E_p}{\partial w_{ij}(t)} = w_{ij}(t) - \alpha \delta_j^l F'(S_j) y_i \quad (4)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E_p}{\partial T_j(t)} = T_j(t) - \alpha \delta_j^l F'(S_j) \quad (5)$$

где $\alpha \in (0,1)$ — коэффициент скорости обучения; δ_j^l — ошибка, передаваемая через j -й нейрон слоя l в обратном направлении. Эта ошибка вычисляется по следующему правилу:

$$\delta_j^l = \begin{cases} (t_j - y_j), & l = L; \\ (\sum_{r=1}^{N_{l+1}} \delta_r^{l+1} F'(S_r) w_{jr}^{l+1}), & 1 \leq l < L \end{cases} \quad (6)$$

Алгоритм обучения метода обратного распространения ошибки состоит из следующих шагов:

1. Задают шаг обучения α ($0 < \alpha < 1$) и желаемую среднеквадратичную ошибку E_m
2. Случайным образом инициализируются весовые коэффициенты и пороговые значения нейронной сети.
3. Последовательно подаются на вход сети входные образы из обучающей выборки. Для каждого входного образа выполняются следующие действия:
 - 3.1. Производится фаза прямого распространения входного образа по нейронной сети с вычислением выходов всех нейронных элементов сети по формуле

$$y_j = F(\sum_i w_{ij} y_i - T_j),$$
 где индекс j характеризует нейроны следующего слоя по отношению к слою i .
 - 3.2. Осуществляется фаза обратного распространения сигнала, в результате которой определяется ошибка δ_j , $j=1,2,\dots$ нейронных элементов для всех слоев сети.
 - 3.3. Происходит изменение весовых коэффициентов и порогов для каждого слоя нейронной сети по формулам (4) и (5).
4. Вычисляется суммарная среднеквадратичная ошибка нейронной сети

$$E = \frac{1}{2} \sum_{k=1}^P \sum_j (y_j^k - t_j^k)^2 \quad P - \text{размерность обучающей выборки.}$$

5. Если $E > E_m$, то происходит переход к шагу 3. В противном случае обучение заканчивается.

Обучение по методу обратного распространения ошибки имеет два существенных недостатка: большое время обучения (для получения приемлемых по точности результатов) и сбой в обучении, связанные с локальными минимумами.

На скорость процесса обучения влияют выбранное значение шага обучения α , начальная инициализация весовых коэффициентов, архитектура сети (количество слоев, нейронов, вид функций активации). Например, для сигмоидной функции активации при задании больших (положительных и отрицательных) начальных значений весовых коэффициентов выходная активность нейронов будет близка к 1 или 0, а значит изменения весовых коэффициентов будут незначительными. Это может привести либо к остановке в точке локального минимума среднеквадратичной функции ошибки, либо к значительной длительности процесса обучения. Выбор начального шага обучения α близким к 1 может сделать процесс обучения расходящимся или привести к пропуску глобального минимума среднеквадратичной ошибки. Значения $\alpha \approx 0$ увеличивают длительность обучения и могут привести к остановке обучения в точке локального минимума. Все это иногда делает процесс обучения нейронной сети до приемлемой точности решения задачи сложным и многоэтапным. Существует ряд рекомендаций по выбору начальных значений весовых коэффициентов и шага обучения.

Весовые коэффициенты можно:

- 1) задавать согласно выражению $w_{ij} \approx \frac{1}{\sqrt{n(i)}}$, где $n(i)$ – число нейронов в i -м слое;
- 2) инициализировать случайным образом в диапазоне $[-0.05; 0.05]$ или $[-0.1; 0.1]$.

Для выхода из небольших локальных минимумов среднеквадратичной ошибки изменение весовых коэффициентов в алгоритме обучения предлагается модифицировать следующим образом []:

$$\Delta w_{ij}(t+1) = w_{ij}(t+1) - w_{ij}(t) = -\alpha \delta_j F'(S_j) y_i + \gamma \Delta w_{ij}(t),$$

где γ – моментный параметр, выбирающийся из диапазона $[0;1]$. Моментный параметр является подобием памяти последнего изменения веса: при его значении, близком к единице, значительное влияние на шаг градиентной оптимизации оказывает предыдущее значение шага.

Тогда общую рекомендацию к выбору шага обучения и моментного параметра можно сформулировать таким образом:

- 1) выбирать большим начальный шаг обучения с целью быстрого поиска глобального минимума и уменьшать его по мере приближения к заданному числу итераций или выбранной среднеквадратичной ошибки E_m для более точной настройки весов;
- 2) начальное значение моментного параметра γ выбирать из диапазона $[0.7;0.9]$, и в процессе обучения уменьшать до 0.

Также можно использовать адаптивный шаг обучения, найденный по методу наискорейшего спуска [].

Процедура обучения обратным распространением ошибки применяется в двух режимах: оперативном и пакетном. Оперативное обучение подразумевает, как и описывалось выше, коррекцию весовых коэффициентов после предъявления каждой обучающей пары. Пакетный режим предусматривает накопление изменений весов на протяжении некоторой эпохи без изменений самих весовых коэффициентов. Обычно временная протяженность эпохи соответствует предъявлению всего множества обучающих пар один раз. По окончании эпохи веса корректируются полученными кумулятивными значениями. Далее процесс итеративно повторяется.

Ускорение обучения

Как уже отмечалось, обучение по методу обратного распространения ошибки является эквивалентом оптимизации функции стоимости $C(x)$ в пространстве $x \in \mathbb{R}^n$ без дополнительных ограничений.

$$\min_{x \in \mathbb{R}^n} C(x)$$

При этом описанный алгоритм поиска оптимума демонстрирует линейную сходимость (без использования момента):

$$x_{k+1} = x_k - \eta_k \nabla C(x_k)$$

Низкая скорость обучения определяется линейностью сходимости используемого алгоритма оптимизации, опирающегося только на первую производную функции стоимости $C(x)$. Поэтому для улучшения этого параметра применяются более сложные схемы [5].

Представим $C(x)$ вблизи точки $x_0 \in \mathbb{R}^n$ рядом Тейлора до второй производной включительно:

$$C(x) = C(x_0) + g^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x \quad (7)$$

где $\Delta x = x - x_0$ — шаг, $g = \nabla C(x_0)$ — градиент в точке x_0 , $H = \nabla^2 C(x_0)$. Из выражения (1) следует значение оптимального шага:

$$\frac{\partial C}{\partial (\Delta x)} = g + H \Delta x = 0 \quad \Rightarrow \quad \Delta x = -H^{-1} g$$

Записанное правило вычисления шага называется **методом Ньютона** и соответствует квадратичной сходимости, ускоряющей процесс обучения.

В **методе Левенберга-Маркварда** (Levenberg-Marquadt) вводится дополнительное ограничение на шаг $\|\Delta x\|_2 = 1$:

$$L(\Delta x, \lambda) = C(x_0) + g^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x + \frac{\lambda}{2} (\Delta x^T \Delta x - 1)$$

Условие оптимальности:

$$\frac{\partial L}{\partial (\Delta x)} = g + H \Delta x + \lambda \Delta x = 0 \quad \Rightarrow \quad \Delta x = -(H + \lambda I)^{-1} g$$

При $\lambda = 0$ формула для вычисления шага сводится к методу Ньютона. Для больших значений λ формула соответствует градиентному спуску в стандартном методе обучения.

Основной недостаток метода Ньютона применительно к его прикладному применению заключается в необходимости расчета H , то

есть, в вычислении второй производной функции стоимости. На практике это приводит к значительному замедлению шага процедуры обучения. Для устранения указанного недостатка используются различные приближенные оценки H по первой производной $c(x)$. Обобщенное название такого подхода — **метод квази-Ньютона**.

Перепишем (1) в виде $Hd_k = y_k$, где $d_k = x_{k+1} - x_k$ и $y_k = g_{k+1} - g_k$. То есть, имеем линейное отображение между изменением градиента и изменением позиции в пространстве \mathbb{R}^n . Представим теперь, что $c(x)$ не квадратичная и B_k — оценка H . Тогда $B_{k+1} = B_k + \Delta B$.

Следующее уравнение называется условием квази-Ньютона:

$$B_{k+1}d_k = y_k$$

Правило прямого обновления H представлено ниже:

$$B_{k+1} = B_k + \frac{(y_k - B_k d_k)(y_k - B_k d_k)^T}{(y_k - B_k d_k)^T d_k} \quad \text{ранг 1}$$

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T d_k} + \frac{(B_k d_k)(B_k d_k)^T}{(B_k d_k)^T d_k} \quad \text{ранг 2}$$

Если представить условие квази-Ньютона в обратном виде $d_k = R_{k+1}y_k$, получим так называемое обратное обновление H :

$$R_{k+1} = R_k + \frac{d_k d_k^T}{d_k^T y_k} + \frac{R_k y_k y_k^T R_k}{y_k^T R_k y_k}$$

Другой метод оптимизации без обновления матрицы H называется **методом сопряженного градиента** (conjugate gradient). Используется следующая схема:

$$x_{k+1} = x_k + \alpha_k u_k,$$

где u_k — направление поиска, определяемое по правилу:

$$u_0 = -g_0$$

$$u_{k+1} = -g_{k+1} + \beta_k u_k$$

Способы расчета β_k у разных авторов различаются:

$$\text{Флетчер – Ривз:} \quad \beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$$

$$\text{Полак – Рибир:} \quad \beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}$$

Следует отметить, что моментуму в методе обратного распространения соответствуют коэффициенты α и β метода сопряженного градиента.

Характерное свойство метода сопряженного градиента состоит в том, что текущее направление поиска перпендикулярно к градиенту в следующей точке:

$$u_k^T g_{k+1} = 0$$

и направления поиска u_k и u_{k+1} сопрягаются, учитывая H :

$$u_k^T H u_{k+1} = 0$$

Если функция стоимости $c(x)$ чисто квадратичная, метод сопряженного градиента сходится в n -мерном пространстве за n шагов. В пространстве высокой размерности этот метод предпочтительней метода Ньютона.

Методы обучение без учителя

Обучение без учителя требует знания только входных данных для нейронной сети. В процессе обучения нейронная сеть сама выявляет закономерности во входных данных и обучается эти закономерности воспроизводить на выходе. В живой природе аналогом данного подхода к обучению является, например, способность человеческого интеллекта к обобщению и выявлению схожих черт в вещах и явлениях. Недостатками обучения без учителя является непредсказуемость выхода нейросети и неуверенность в актуальности для всего множества входных данных закономерностей, полученных на обучающей выборке. Для идентификации результата работы нейросети как правило требуется дополнительный распознающий блок.

Наиболее известным методом обучения в рамках данного подхода является конкурентное обучение.

Обучение без учителя используется для обучения самоорганизующихся нейронных сетей. Наиболее известными среди них являются сети Кохонена [], разработанные в 80-х годах финским ученым.

В 1976 г. С. Гроссбергом был предложен и затем развит в работах Т. Кохонена метод обучения без учителя названный конкурентным методом обучения. Нейронные сети, использующие такой метод обучения, называются конкурентными.

Суть конкурентного метода обучения рассмотрим на примере конкурентной сети изображенной на рис . Первый слой этой сети выполняет распределительные функции, причем каждый нейрон имеет соединения со всеми нейронными элементами выходного слоя. Второй слой (слой нейронов Кохонена) при подаче на вход сети определенного образа осуществляет конкуренцию между нейронными элементами, в результате которой определяется нейрон-победитель, имеющий максимальную взвешенную активность (правило известно под названием «победитель берет все»):

$$S_j = \sum_i w_{ij} x_i = W_j X^T, \quad (8)$$

где $X = \{x_1, x_2, \dots, x_n\}$ — входной образ, $W_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$ — вектор столбец весовых коэффициентов j -го выходного нейрона.

Пусть $S_k = \max_j S_j$, тогда активность выходных нейронов

$$y_j = F(S_j) = \begin{cases} 1, & \text{если } j = k; \\ 0, & \text{если } j \neq k, \end{cases} \quad (9)$$

где $j = \overline{1, m}$.

В процессе обучения для нейрона-победителя синаптические связи усиливаются, а для остальных нейронов не изменяются или могут уменьшаться.

Выражение (8) эквивалентно скалярному произведению вектора весов соответствующего нейронного элемента на входной вектор нейронной сети:

$$S_j = |W_j| \cdot |X| \cdot \cos \alpha, \quad (10)$$

где α - угол между векторами W_j и X .

Отметим, что для адекватного определения нейрона-победителя вектора W_j и X должны быть нормированы, тогда $S_j = |W_j| \cdot |X| \cdot \cos \alpha = \cos \alpha$ и максимальную активность будет иметь тот нейрон, весовой вектор которого коллинеарен входному вектору. Концы векторов при этом находятся на поверхности n -мерной сферы, радиус которой равен единице. В этом случае правило настройки весовых коэффициентов нейрона-победителя должно соответствовать вращению вектора W_k в сторону вектора X (рис.7).

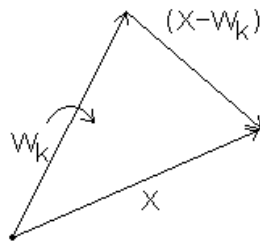


Рисунок 7: Модификация весового вектора нейрона победителя

В результате можно записать следующее правило обучения для вектора весов k -го нейрона-победителя:

$$W_k(t+1) = W_k(t) + \gamma(X - W_k(t)) \quad \text{или} \quad w_{ik}(t+1) = w_{ik}(t) + \gamma(x_i - w_{ik}(t)) \quad (11)$$

, где $0 < \gamma < 1$ характеризует скорость обучения; $i = \overline{1, n}$

С учетом нормировки правило обучения записывается в следующем виде:

$$W_k(t+1) = \frac{W_k(t) + \gamma(X(t) - W_k(t))}{|W_k(t) + \gamma(X(t) - W_k(t))|} \quad \text{или} \quad w_{ik}(t+1) = \frac{w_{ik}(t) + \gamma(x_i(t) - w_{ik}(t))}{|W_k(t) + \gamma(X(t) - W_k(t))|}$$

В случае использования ненормализованных векторов для определения нейрона-победителя нужно оперировать вместо взвешенной активности (6) евклидовым расстоянием

$$D_j = |X - W_j| = \sqrt{(x_1 - w_{1j})^2 + (x_2 - w_{2j})^2 + \dots + (x_n - w_{nj})^2} \quad (12)$$

При помощи (10) определяется нейрон-победитель с номером k , соответствующий минимальному евклидовому расстоянию между входным и весовым векторами:

$$D_k = \min_j |X - W_j| \quad (13)$$

Тогда настройка весового вектора нейрона-победителя происходит следующим образом:

$$W_k(t+1) = W_k(t) + \gamma(X(t) - W_k(t)) \quad (14)$$

Недостатком этого метода обучения является то, что при случайной инициализации весовых векторов может получиться так, что некоторые нейроны никогда не будут победителями. Для нейтрализации этого недостатка можно расширить правило обучения следующим образом []:

$$W_j(t+1) = W_j(t) + \gamma(X - W_j(t)), \forall j = k,$$

$$W_j(t+1) = W_j(t) + \gamma'(X - W_j(t)), \forall j \neq k,$$

где $\gamma' \ll \gamma$

Данное правило позволяет отобразить весовые векторы побежденных нейронов в область, где увеличиваются их шансы в конкуренции.

Объединение различного рода нейронных сетей и концепций их обучения привело к появлению гибридных нейронных сетей. Так, дальнейшим расширением нейронных сетей Кохонена являются сети встречного распространения сигналов (рис.), предназначенные для аппроксимации функций. Эти сети характеризуются сочетанием двух подходов к обучению: с учителем и без учителя. В свою очередь развитием сетей встречного распространения стали нейронные сети с радиально-базисной функцией активации, применяющиеся для решения задач распознавания образов, прогнозирования, сжатия данных и аппроксимации функций.

Рекомендации по литературе

Для изучения архитектуры различных нейронных сетей, методов их обучения и областей применения в рамках учебного курса рекомендуются книги [4–7]. Для глубокого ознакомления с теорией нейронных сетей рекомендуются книги [8–9]. Расширенное представление о способах применения нейронных сетей для решения задач управления можно получить в книге [10].

Нейросетевой пакет NNACS представлен в статьях [11–12]. Доступ к исходным текстам и двоичным сборкам представлен по ссылкам [13–14].

Изучение основных понятий теории ИНС и приемов работы с пакетом Statistica Neural Networks

Цель работы: освоение приемов работы с программным пакетом Statistica Neural Networks и изучение основных понятий теории ИНС на примере простых задач распознавания логических функций (И, ИЛИ, исключающее ИЛИ).

Задание

1. Ознакомиться с интерфейсом и функциональными возможностями программы Statistica Neural Networks. (Пункты меню File, Edit, Statistics, Run).
2. Задать исходные данные для решения задачи распознавания логической функции AND. Для этого выбрать пункт меню File->New Data Set..., указать количество входов (Inputs) равное двум, количество выходов (Outputs) равное единице. В открывшемся окне (Data Set Editor) ввести данные из таблицы

| X | Y | X AND Y |
|---|---|---------|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

Сохранить исходные данные в файле «AND_data.sta» (File->Save as...).

3. Построить нейросеть (НС) для решения задачи из п.2.. Открыть окно Create Network (File->New Network...), указать тип нейронной сети Multilayer Perceptron, количество слоев 2 (No Layers), число входов (Inputs) равное двум, число выходов (Outputs) равное единице, остальные параметры оставить без изменения (Steps = 1, Lookahed = 0). Убедиться в том, что для всех нейронов установлена функция преобразования Minimax и создать НС нажатием кнопки Create. Изучить возможности изменения параметров созданной НС (Edit->Network...).
4. Обучить НС. Вызвать окно отображения ошибки обучения (Statistics->Training Graph...). Открыть окно с параметрами обучения Алгоритма обратного распространения ошибки (Train->Multilayer Perceptrons->Back Propagation...). Добиться приемлемо низкой ошибки обучения, задавая различные параметры обучения (количество эпох, шаг обучения, ...). Ошибки

тестирования нейронной сети можно посмотреть, воспользовавшись пунктом меню Run->Data Set....

При невозможности качественно обучить НС следует изменить структуру НС (количество слоев, количество нейронов в слоях, функции активации нейронов).

5. Решить задачи распознавания логических функций (OR, XOR). Исследовать решение этих задач при задании различных функций активации нейронов обрабатывающих слоев (линейная, пороговая, сигмоидная (логистическая)).

Контрольные вопросы к лабораторной работе №1

1. Дайте определение понятия “формальный искусственный нейрон” и поясните его назначение.
2. Приведите классификацию нейронных сетей и примеры их применения.
3. Какие функции активации вам известны? Расскажите, в каких случаях следует применять ту или иную функцию.
4. С помощью какого алгоритма обучается нейронная сеть при решении задачи распознавания логических функций? Расскажите про этапы создания, обучения и проверки работоспособности нейронной сети.
5. Что такое шаг обучения? Как он влияет на получение конечного результата.
6. Что такое ошибка обучения ИНС?
7. В чем сходство процедуры обучения ИНС и человека и в чем различие?

Лабораторная работа № 2

Многослойный персептрон.

Цель работы: изучение алгоритмов обучения многослойного персептрона, выбор структуры и контроль качества обучения ИНС, решение задачи классификации многомерных данных.

Задание

1. Открыть хранящиеся в файле `min_water.txt` данные о сорока образцах минеральной воды. Проведя предварительный анализ данных, выделить из 23-х признаков наиболее информативные для классификации образцов признаки. Разделить имеющуюся выборку на три части: для обучения, для верификации, для тестирования. Сохранить исходные данные.
2. Создать многослойный персептрон с одним нейроном в выходном слое. Решить задачу классификации имеющихся данных об образцах минеральной воды по пяти классам, используя при обучении алгоритм обратного распространения ошибки. Проанализировать, полученные результаты.
3. Создать многослойный персептрон с пятью нейронами в выходном слое и снова решить задачу классификации имеющихся данных об образцах минеральной воды по пяти классам. Проанализировать, полученные результаты. По результатам п.2 и п.3. выбрать наилучшую структуру ИНС для решения поставленной задачи.
4. Решить задачу классификации образцов минеральной воды с помощью любого другого алгоритма обучения.
5. Решить задачу классификации образцов минеральной воды по четырем классам, предварительно пометив один из классов, как недоступный в процессе обучения. Проанализировать результаты классификации, предъявив обученной ИНС неизвестный сорт минеральной воды. Объяснить полученные результаты.

Контрольные вопросы к лабораторной работе №2

1. Что такое многослойный персептрон? К какому классу ИНС он относится? Какие практические задачи позволяет решать?
2. Как выбирается архитектура ИНС при решении задачи классификации?
3. Какие алгоритмы обучения предназначены для обучения многослойного персептрона? Приведите их достоинства и недостатки.
4. Объясните принцип алгоритма обратного распространения ошибки.

5. Что такое эффект переобучения ИНС? Как проконтролировать его возникновение?
6. Каково назначение моментного параметра в алгоритме обратного распространения ошибки?
7. Как оценивается качество обучения и работы ИНС? Как рассчитывается ошибка обучения?

Применение многослойного персептрона. Автоассоциативная ИНС.

Цель работы: знакомство с применением многослойного персептрона для решения задач сжатия данных, прогнозирования временных рядов и распознавания образов.

Задание

1. Открыть файл с данными по минеральной воде, который использовался при решении задач классификации в предыдущей лабораторной работе. Построить и обучить автоассоциативные нейронные сети с 2-я и 3-я нейронами в скрытом слое:

а). для исходных данных из 5-ти классов;

б). для исходных данных из 4-х классов.

Провести визуализацию данных в скрытом слое каждой сети на плоскость и в 3-х мерное пространство. Проанализировать полученные результаты. Выбрать и сохранить автоассоциативные ИНС, обеспечивающие наилучшее сжатие исходных данных

2. Исследовать возможности ИНС по прогнозированию поведения нелинейных динамических систем (построение странного аттрактора) на примере отображения Хенона. Аттрактор Хенона может быть получен из уравнений:

$$x_{n+1} = 1 - \alpha x_n^2 + y_n$$

$$y_{n+1} = \beta x_n$$

$$\alpha = 1.4 \quad \beta = 0.3$$

Решение данных уравнений представлено в файле NK4_Xenon.sta. Для прогнозирования предлагается использовать многослойный персептрон (МП) и сеть с радиально базисными функциями. Например, для прогнозирования координаты X на один шаг вперед можно создать МП с 7 нейронами во входном слое, 5-тью нейронами в скрытом слое и 1 нейроном в выходном. (File->New->Network, Steps = 7, Lookahead = 1, Layer2 = 5, Create). При этом в исходных данных координату X необходимо пометить как Input/Output, а координату Y как Ignore. В качестве функций активации в выходном слое предлагается использовать логистическую функцию или линейную. Сеть можно обучить с помощью алгоритма обратного распространения ошибки или алгоритма Levenberg-Marquardt-a. После обучения для оценки результатов прогнозирования воспользоваться пунктом меню Run->Time Series....

Проанализировав полученные результаты, постройте прогноз курса доллара на 1-2 дня вперед. В качестве исходных данных можно использовать курс доллара за несколько лет из файла USD.sta или загрузить актуальные данные с сайта центрального банка России (<http://www.cbr.ru>).

3. Решить задачу распознавания 9-ти изображений самолетов. Исходные данные (файлы avia1.bmp, ..., avia9.bmp) необходимо предварительно преобразовать в набор векторов со значениями признаков 0 или 1. Обученная нейронная сеть должна правильно определять модель самолета и его класс (истребитель/бомбардировщик). Принадлежность модели к определенному классу выбирается студентом самостоятельно.

Контрольные вопросы к лабораторной работе №3

1. Какая ИНС называется автоассоциативной? В чем особенности её обучения?
2. Как оценивается степень сжатия данных при использовании автоассоциативной ИНС?
3. Аналогом какого классического метода снижения размерности может считаться снижение размерности с помощью автоассоциативная ИНС и при каких параметрах обучения?
4. С помощью каких ИНС можно решать задачи прогнозирования?
5. Зависит ли результат прогноза от вида используемой ИНС (гомогенная/гетерогенная сеть)?
6. Можно ли использовать для прогнозирования и сжатия данных сеть с радиально-базисными функциями?
7. Какие ИНС могут использоваться в задачах распознавания образов?

Лабораторная работа № 4

Сеть Кохонена

Цель работы: знакомство с алгоритмом обучения ИНС без учителя, изучение сети Кохонена на примере задачи кластеризации данных.

Задание

1. Изучить разделы справки связанные с обучением сети Кохонена. Открыть файл исходных данных Iris.sta, содержащий измерения длины и ширины чашелистика и лепестка 150 экземпляров ириса. Создать сеть Кохонена и выполнить с помощью неё кластеризацию сортов ириса. Проанализировать полученные результаты. Выполнить визуализацию исходных данных.
2. Построить и обучить сеть Кохонена для кластеризации данных скрытого слоя автоассоциативной сети из п.1. лабораторной работы №3. Проанализировать результаты и сравнить их с результатами классификации многослойным персептроном, полученными в лабораторной работе №2.

Контрольные вопросы к лабораторной работе №4

1. Какова структура сети Кохонена? Для решения каких задач предназначена эта сеть? С помощью какого алгоритма обучается?
2. Объясните принцип алгоритма обучения без учителя. Каков критерий останова процедуры обучения сети Кохонена?
3. Как инициализируются весовые коэффициенты сети перед началом обучения? Зачем нужна нормировка входных векторов и весовых коэффициентов?
4. Объясните, с помощью какого алгоритма в Statistica Neural Networks осуществляется присвоение меток кластеров нейронам сети Кохонена?
5. Пояснить, можно ли использовать сеть Кохонена для сжатия данных?
6. Что общего у сетей Кохонена и многослойного персептрона и чем они отличаются?
7. Какой результат можно ожидать при обучении сети Кохонена при количестве выходных нейронов, меньшем (большем) числа обучаемых классов?

Синтез нейросетевого оптимального регулятора

Цель работы Лабораторная работа посвящена задаче синтеза нейросетевого оптимального регулятора (НОР). В ней изучаются принципы применения нейронных сетей в качестве регулятора и модели объекта управления. Исследуется влияние архитектуры нейронной сети на скорость и качество обучения. Полученный НОР используется в последующих лабораторных работах.

Задание Имеется система управления с обратной связью. У объекта управления один управляющий вход и один наблюдаемый выход. Известно, что в канале наблюдения имеется случайная помеха. Дана стохастическая модель сигнала уставки. Необходимо заменить регулятор на нейросетевой с минимизацией квадрата ошибки управления.

Для простоты понимания задачи и в целях последующего сравнения с линейными регуляторами в качестве объекта управления и регулятора берутся линейные звенья, хотя методика синтеза никак не ограничивает в выборе класса управляемого объекта и заменяемого регулятора.

В процессе выполнения промежуточных этапов синтеза исследуется влияние архитектуры нейронной сети (количество слоев, распределение нейронов в них) на скорость обучения и достигаемое качество.

Необходимо последовательно выполнить следующие этапы:

1. Моделирование САУ с линейным регулятором в контуре и со ступенчатой уставкой (меандр) в качестве уставки с целью удостовериться в устойчивости контура и управляемости объекта. При необходимости надо изменить параметры регулятора.
2. Моделирование САУ с линейным регулятором в контуре, со стохастической уставкой и случайной помехой с целью сбора данных для обучения. Следует провести два разных сеанса получением двух независимых выборок: обучающей и контрольной.
3. Обучение нейросетевого регулятора вне контура управления подобно исходному линейному регулятору. При этом следует создать нейронную сеть регулятора и указать файлы данных для обучения: обучающую и контрольную выборки. Для получения лучшего результата провести эксперименты с различными архитектурами нейронных сетей.
4. Моделирование САУ с нейросетевым регулятором, обученным вне контура. На вход подается стохастическая выборка, оценивается качество управления.

5. Обучение нейросетевой модели объекта вне контура управления. При этом следует создать нейронную сеть модели объекта и указать файлы данных для обучения: обучающую и контрольную выборки. Для получения лучшего результата провести эксперименты с различными архитектурами нейронных сетей и наборами входов.
6. Обучение нейросетевого оптимального регулятора в контуре управления. В качестве начального приближения берется нейросетевой регулятор, обученный вне контура управления. Задействуется также предварительно настроенная нейросетевая модель объекта управления. Обучение проводится на стохастической выборке уставки и помехи.
7. Моделирование САУ с нейросетевым оптимальным регулятором, обученным в контуре. На вход подается стохастическая выборка, оценивается качество управления.

Отчет должен включать следующие элементы:

- Описание исходной системы управления.
- Архитектура нейросети регулятора (НС-Р). Перечень опробованных вариантов архитектур.
- Характеристики обучающей и контрольной выборок: формирующий фильтр, длина, минимальная и максимальная амплитуды сигналов в выборках.
- Процесс обучения нейросетевого регулятора: графики ошибки, длительность и результат обучения. Выбор одного из вариантов архитектуры НС-Р
- Проверка нейросетевого регулятора в контуре, сравнение с исходным линейным.
- Численная оценка качества управления нейросетевым регулятором, обученным вне контура (СКО, максимальное перерегулирование).
- Архитектура нейросети модели объекта (НС-О).
- Характеристики обучающей и контрольной выборок: формирующий фильтр, длина, минимальная и максимальная амплитуды сигналов в выборках.
- Процесс обучения нейросетевой модели объекта: графики ошибки, длительность и результат обучения. Выбор одного из вариантов архитектуры НС-О.
- Процесс обучения нейросетевого оптимального регулятора в контуре управления. Параметры обучения. Графики ошибок управления и идентификации.
- Сравнение качества управления (СКО, максимальное перерегулирование) для исходного регулятора, НС-Р, обученного вне

контура, нейросетевого оптимального регуляторов (на одной и той же выборке). Графики уставки и её отработки объектом.

Выводы должны содержать перечисление факторов, оказывающих влияние на процесс обучения нейросети вне и в контуре управления.

Контрольные вопросы к лабораторной работе №5

1. Что такое η и почему значение этого параметра меняется в процессе обучения НС-Р?
2. Почему для настройки нейросетевого регулятора и нейросетевой модели объекта вне контура управления используются стохастические сигналы?
3. Для чего используется нейросетевая модель объекта управления?
4. Какие параметры управляют процессом обучения НОР в контуре и каков их смысл?
5. Чем отличается обучение нейросетевого регулятора в контуре управления и вне его?
6. Можно ли использовать сигмоидальную функцию активации в последнем слое нейросетевой модели объекта?
7. Чем должен отличаться НОР от нейросетевого регулятора, обученного вне контура управления?
8. Можно ли использовать сигмоидальную функцию активации выходного нейрона нейросетевого регулятора?
9. Какие критерии используются для останова процедуры обучения нейросетевого регулятора вне контура управления?

Лабораторная работа № 6

Сравнительное исследование нейросетевого оптимального и ПИД регуляторов

Цель работы Работа посвящена исследованию свойств нейросетевого оптимального регулятора в сравнении с линейным регулятором типа ПИ/ПИД. Сравнение регуляторов проводится в различных условиях на основе интегрального (среднеквадратическая ошибка) и экстремального (максимальная ошибка) критериев.

Задание Имеется система управления с обратной связью и линейным объектом управления. Стохастические параметры уставки и помехи заданы. Для системы синтезированы ПИ/ПИД регулятор и нейросетевой оптимальный регулятор (НОР), например, полученный во время первой лабораторной работы. Сравнение регуляторов следует проводить исходя из достигнутого ими качества управления на достаточно длинном временном интервале. Критериями качества управления являются максимальное по модулю перерегулирование и среднеквадратическая ошибка.

Требуется экспериментально исследовать поведение регуляторов в номинальных условиях (формирующие фильтры уставки и помехи совпадают с использовавшимися при синтезе нейросетевого оптимального регулятора, объект управления тот же самый), на разных видах сигнала уставки (ступенчатом, гармоническом, стохастическом), а также в условиях, отличных от номинальных: иные параметры объекта управления, уставки и помехи (отсутствие, номинальный уровень, повышенный уровень помехи, “цветная” помеха). Экспериментально определить зависимость качества управления от частоты, подавая на вход системы гармоническую уставку.

Варианты Данная работа проводится по тем же вариантам параметров объекта управления и регулятора, как и лабораторная работа №5. Параметры линейного регулятора допускается произвольно изменять (например, подобрав их более оптимально, чем это сделано в таблице вариантов).

План работы Каждый пункт плана работы подразумевает независимый от других пунктов сеанс моделирования для каждого из двух регуляторов. Перед этим необходимо подготовить сигналы уставки и помехи, а также установить параметры объекта управления сообразно заданию.

Перечень экспериментов:

1. Номинальная стохастическая уставка и помеха, номинальный объект управления.
2. Уставка — меандр с амплитудой 1 и длительностью ступеней 50 (достаточной для завершения переходного процесса). Помеха — отсутствует. Объект управления — номинальный.
3. Стохастическая уставка, отличная от номинальной с дисперсией, примерно совпадающей с номинальной. Номинальная помеха. Объект управления — номинальный.
4. Стохастическая уставка, отличная от номинальной с дисперсией, превышающей номинальную (взять больший коэффициент усиления). Номинальная помеха. Объект управления — номинальный.
5. Номинальная стохастическая уставка. Помеха — белый шум с интенсивностью в два раза больше, чем номинальная. Объект управления — номинальный.
6. Номинальная стохастическая уставка. Помеха — “цветной” шум небольшой интенсивности. Объект управления — номинальный.
7. Номинальная стохастическая уставка и помеха. Параметры объекта управления отличаются от номинальных (изменить коэффициент усиления и/или константы времени).
8. Уставка — синусоида с амплитудой 1. Помеха — отсутствует. Объект управления — номинальный. Повторить для нескольких различных периодов уставки (4, 6, 8, 10, 16, 20, 40).

Отчет должен включать описание объектов управления, параметры ПИ/ПИД регулятора, архитектуру НОР и номинальные условия его обучения (формирующие фильтры уставки и помехи, а также их диапазон амплитуд в процессе обучения).

По результатам экспериментов должна быть составлена таблица следующего формата:

| № | Объект | Уставка | | Помеха | | Длина | ПИ/ПИД | | НОР | |
|---|--------|---------|-------|--------|-------|-------|--------|-------|-----|-------|
| | | Вид | A max | Вид | A max | | СК | e max | СК | e max |
| | т | д | х | д | х | а | О | х | О | х |
| | | | | | | | | | | |

Примечания:

- Для объекта в таблице указывается: **ном.** – номинальные параметры и **изм.** – параметры изменены.

- Для уставки и помехи вид: **ном.** – номинальные (как при синтезе НОР), **изм.** – стохастические, отличные от номинальных, **меандр** – знакопеременная ступенька, **гарм,(T)** – синусоида с указанным периодом, **конст.** – постоянное значение.
- $|A|_{\max}$ – максимальная амплитуда.
- Длина – длительность выборки в отсчетах.
- СКО – среднеквадратическая ошибка управления (MSE в терминах программы).
- $|e|_{\max}$ – максимальное перерегулирование.

Также должны быть представлены графики зависимости СКО и $|e|_{\max}$ от частоты (для гармонической уставки) и графически проиллюстрированы результаты управления в экспериментах 2, 4, 7.

Выводы должны на основании приведенных в таблице результатов экспериментов содержать перечисление факторов, оказывающих влияние на качество регулирования линейным и нейросетевым оптимальным регуляторами. Перечислить и аргументировать отличия нейросетевого регулятора от линейного.

Контрольные вопросы к лабораторной работе №6

1. Быстродействие какого регулятора выше: ПИД или НОР? Покажите на графиках.
2. Какой регулятор и почему лучше справляется со ступенчатой уставкой (меандр) в смысле минимума СКО?
3. Почему на гармонической уставке малой частоты ПИ/ПИД лучше НОР?
4. Что нужно сделать, чтобы нейросетевой регулятор лучше обрабатывал отрицательную часть гармонической уставки?
5. Какой из регуляторов более робастный (менее чувствительный к выбранному изменению параметров объекта)?
6. Можно ли построить частотные характеристики для НОР?
7. На каких видах уставок нейросетевой регулятор очевидно хуже ПИД?
8. Почему у нейросетевого регулятора два входа: уставка и ошибка управления?
9. Предположите, смог бы нейросетевой регулятор с единственным входом – ошибкой управления – использоваться в качестве нейросетевой имитации ПИД регулятора?

Лабораторная работа № 7

Нейросетевая модель объекта управления

Цель работы – изучение влияния архитектуры нейросетевой модели объекта управления и пробных сигналов, используемых для её обучения, на настройку нейросетевого оптимального регулятора в контуре управления.

Задание Дана система управления с традиционным регулятором. Считаем, что архитектура нейросетевого регулятора уже выбрана и он обучен вне контура управления. Для дальнейшего обучения этого регулятора в контуре необходимо построить нейросетевую модель объекта управления. Архитектура этой нейронной сети определяется следующими параметрами:

- Количество входов с задержанным сигналом управляющего воздействия u_k, u_{k-1}, \dots
- Количество входов с задержанным сигналом наблюдаемого выхода объекта (с помехой) y_{k-1}, y_{k-2}, \dots
- Количество слоев и распределение нейронов в них.
- Функция активации нейрона выходного слоя.

Данная нейронная сеть не имеет обратных связей и реализует имитацию динамики объекта управления за счет задержанных входных сигналов. Нейронная сеть настраивается на данных, полученных в результате наблюдения за функционированием исходной системы управления (с традиционным регулятором) при подаче в качестве уставки некоторого пробного сигнала. Таким образом, обучающая и контрольная выборка формируются вследствие влияния пробного сигнала уставки на систему управления. Вид обучающих данных наряду с архитектурой оказывает влияние на обучение нейронной сети и на качество её рабочего функционирования.

Особенность рабочего функционирования НС-О заключается в том, что распространение сигнала в прямом направлении (от входа к выходу), является вспомогательным, хотя и позволяет оценить собственно качество имитации. Основным является обратное распространение ошибки управления через НС-О в направлении от выхода к входам, базирующееся на предварительно вычисленной при прямом распространении степени активности нейронов. Приведение ошибки управления к входу u_k нейросетевой модели эквивалентно расчету оценки якобиана объекта (при

условии достаточно точной имитации объекта моделью). Оценить точность расчета якобиана можно косвенно, по эффективности обучения нейросетевого оптимального регулятора в контуре. Если якобиан рассчитывается недостаточно точно, то обучение НОР не будет приводить к уменьшению ошибки управления, наоборот, возможно ухудшение качества управления.

В рамках лабораторной работы необходимо исследовать влияние параметров нейросети и обучающих выборок от разных пробных сигналов на точность оценки якобиана объекта управления.

План работы

1. Задать в качестве уставки в исходной системе управления сигналы разного вида: стохастический, гармонический и меандр (ступенчатый) в диапазоне рабочих значений. Помеха должна быть стохастической небольшой мощности. Провести соответствующие сеансы моделирования и получить выборки u , y для обучения.
2. Контрольную выборку необходимо получить в отдельном сеансе моделирования со стохастической уставкой.
3. Выбрать пять архитектур НС-О, различающихся по количеству слоев с нейронами (обязательно с 0, 1 и 2 скрытыми слоями) и по количеству входов.
4. Провести обучение каждой из архитектур НС-О на обучающих данных с разными пробными сигналами. Зафиксировать, какая архитектура при каком пробном сигнале обучается лучше всего, то есть, имеет в результате обучения вне контура наименьшую ошибку на контрольной выборке. Итого: 15 сеансов обучения и 5 нейросетей – победителей.
5. Создать достаточно длинную выборку стохастической уставки и помехи в рабочих диапазонах значений ($L=10^5$).
6. Провести сеансы дообучения нейросетевого оптимального регулятора в контуре на заданных в п.5 входных выборках с одного и того же исходного НС-Р, но с различающимися НС-О. Итого: 5 сеансов. Оценить качество имитации объекта нейросетевой моделью.

Отчет должен включать следующие элементы:

- Описание исходной системы управления.
- Перечень опробованных вариантов архитектур нейросети модели объекта (НС-О).
- Перечень с описанием и графики пробных сигналов, использовавшихся для сбора обучающих данных.

- Графики обучения НС-О вне контура для различных пробных сигналов и архитектур НС-О.
- Таблица с достигнутыми числовыми оценками качества имитации объекта управления вне контура (на разных пробных сигналах) и в контуре (среднеквадратические ошибки выхода объекта и модели).
- Графики уставки и выхода объекта в диапазонах: начальном, интенсивной подстройки весовых коэффициентов НОР в процессе дообучения и в конце времени моделирования (либо перед потерей устойчивости в контуре).
- Таблица среднеквадратических ошибок управления в начале и в конце времени моделирования для каждой из архитектур НС-О.

Выводы должны содержать формулировку обнаруженных зависимостей точности оценки якобиана и аргументированный выбор наилучшей с точки зрения обучения НОР в контуре комбинации пробного сигнала и архитектуры НС-О.

Контрольные вопросы к лабораторной работе №7

1. Какие входы и выходы у нейросетевой модели объекта?
2. Для чего на вход нейросетевой модели подают задержанные сигналы управляющего воздействия и выхода объекта?
3. Каковы параметры обучения нейросетевой модели объекта?
4. Как алгоритм обратного распространения используется при обучении НОР?
5. Можно ли использовать при моделировании объекта выход нейросетевой модели, а не объекта?
6. Является ли среднеквадратическая ошибка обучения нейросетевой модели вне контура однозначной характеристикой её применимости для дообучения нейросетевого оптимального регулятора?
7. Как влияет структура сети (количество слоев и нейронов в них) на успех обучения нейросетевой модели и её использования для синтеза НОР?
8. Что можно сказать об устойчивости процесса дообучения НОР в контуре? Как устойчивость связана с качеством нейросетевой имитации объекта?
9. Почему необходим этап предварительного обучения НС-Р вне контура управления?

Нейросетевая система управления нестационарным объектом

Цель работы В работе изучается применение нейросетевой модели объекта совместно с алгоритмом кумулятивных сумм для обнаружения изменения параметров объекта управления.

Задание Имеется система управления с обратной связью и линейным объектом управления. Стохастические параметры уставки и помехи заданы. Для системы синтезированы ПИ/ПИД регулятор, нейросетевой оптимальный регулятор (НОР) и нейросетевая модель объекта управления, например, полученные во время первой лабораторной работы. Будем считать, что объект управления в какой-то момент времени меняется свои свойства. Это моделируется тем, что переходная функция объекта с этого момента задается другим уравнением с другими параметрами. Своевременное обнаружение момента изменения свойств объекта позволяет провести перенастройку регулятора и восстановить качество управления.

Изменение параметров объекта управления приводит к изменению статистических параметров сигналов в контуре, в частности, изменяются дисперсии σ^2 ошибки управления e_i и идентификации d_i . Данный процесс называется разладкой. Для её обнаружения разработан алгоритм кумулятивных сумм (АКС), основанный на статистически надежном определении факта изменения параметра наблюдаемого сигнала, например, дисперсии ошибки идентификации, от исходного значения σ_0^2 к некоторому априорно заданному σ_1^2 , свидетельствующему о так называемой номинальной разладке. При функционировании АКС в каждый момент времени i вычисляется значение изображающей точки S_i (в пакете программ обозначается ip – *imaging point*):

$$z_i = -\frac{1}{2} \ln \frac{\sigma_1^2}{\sigma_0^2} - \frac{1}{2} \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_0^2} \right) d_i^2$$
$$S_i = \begin{cases} 0, & i = 0 \\ \max(0, S_{i-1} + z_i), & i > 0 \end{cases}$$

Критерием обнаружения разладки является достижение изображающей точкой некоторого априорно заданного порога H , называемого решающей границей:

$$S_i > H$$

Факт достижения изображающей точкой решающей границы свидетельствует о разладке в вероятностном смысле. Для надежности следует продолжать процедуру АКС и если в течение короткого времени решающая граница снова будет достигнута, вероятность установления факта разладки будет гораздо выше. Если задать слишком низкий порог H , то возможны ложные тревоги. Если порог H очень высокий, факт разладки будет определен с опозданием, что в реальной системе управления приведет к длительной неоптимальности управления.

В лабораторной работе требуется подобрать параметры АКС для безошибочного и быстрого обнаружения разладки по дисперсии ошибки идентификации: σ_0^2 , σ_1^2 , H . Для безошибочного и быстрого обнаружения разладки. Эксперименты следует проводить при случайном цветном сигнале уставки и случайной помехе.

Варианты Данная работа проводится по тем же вариантам параметров объекта управления и регулятора, как и лабораторная работа №1. Объект управления следует изменить незначительно с тем, чтобы контур не потерял устойчивости как под управлением линейного регулятора, так и под управлением нейросетевого.

План работы Для проведения работы следует вызвать пункт главного меню, посвященный обнаружению разладки в контуре управления. При задании параметров объекта, неизменных во времени, можно воспользоваться файлами с расширением .tf, но для задания объекта, меняющего параметры в процессе моделирования, надо задать имя файла с расширением .cof. Редактирование такого объекта позволит задавать разные переходные функции в разные диапазоны времени. При задании диапазона -1 имеет смысл $+\infty$. Таким образом, нестационарный объект, меняющий коэффициент усиления в $t = 250$, описывается таблицей:

| Имя функции | Формула и параметры | От | До |
|-------------|---------------------------|-----|-----|
| Function1 | $0.2 \frac{z-0.5}{z-0.8}$ | 0 | 249 |
| Function2 | $0.3 \frac{z-0.5}{z-0.8}$ | 250 | -1 |

Последовательность действий:

1. Задать параметры исходного объекта, нейросетевую модель объекта управления и исходный линейный регулятор. Длительность моделирования – 2000 отсчетов. Определить исходную дисперсию ошибки идентификации σ_0^2 .
2. Выбрать задать другие параметры исходного объекта, при которых контур сохраняет устойчивость. Длительность моделирования – 2000 отсчетов. Определить дисперсию ошибки идентификации при номинальной разладке σ_1^2 .
3. Задать нестационарный объект управления с изменением параметров в момент времени $t=1000$. Параметры объекта должны отличаться от заданных в п.2. В параметрах АКС задать дисперсии σ_0^2 , σ_1^2 , интервал второго обнаружения должен быть равен -1. В ряде экспериментов подобрать значение решающей границы, обеспечивающее безошибочное и быстрое обнаружение разладки.
4. Задать положительный интервал второго обнаружения разладки. Моделирование будет останавливаться, если разладка будет обнаружиться дважды в пределах временного интервала данной длительности.

Провести несколько экспериментов, убедившись в надежном функционировании АКС.

Контрольные вопросы к лабораторной работе №8

1. Каковы критерии выбора решающей границы?
2. Какие действия надо произвести, чтобы после обнаружения разладки, считая, что объект некоторое продолжительное время не меняется, снова настроить АКС?
3. Для чего нужна нейросетевая модель объекта управления?
4. Можно ли использовать примененную схему с АКС, если в контуре работает ПИ регулятор?
5. Что показывает график изображающей точки?
6. Как отношение σ_1^2 / σ_0^2 влияет на чувствительность АКС?
7. Можно ли при перенастройке нейросетевой модели вне контура изменить её архитектуру?
8. Что такое «ошибка идентификации» и как она используется?

ПРИЛОЖЕНИЯ

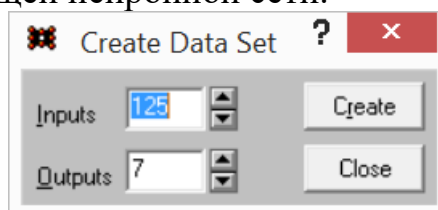
Приложение 1

Statistica Neural Networks. Основные сведения

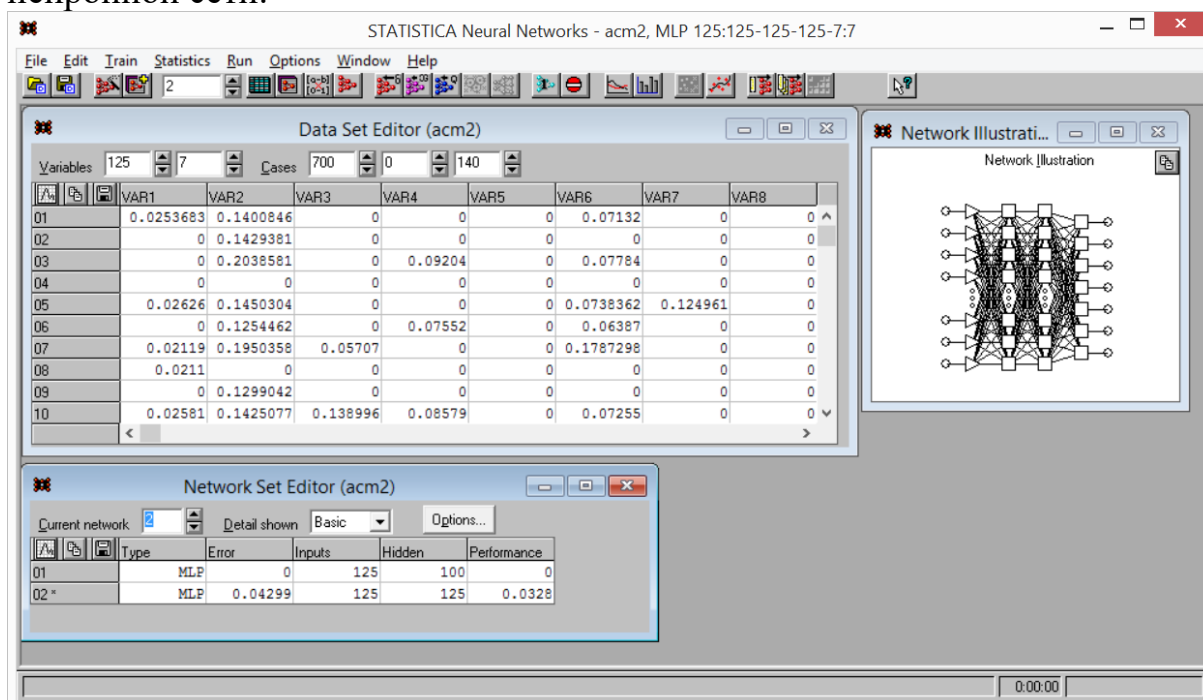
Программный пакет Statistica Neural Networks (идентичен программному пакету Trajan) позволяет обучать и моделировать искусственные нейронные сети нескольких видов: многослойный персептрон, сеть Кохонена, сеть с радиально базисными функциями, вероятностная сеть, обобщенно-регрессионная нейронная сеть.

Работа с программным пакетом Statistica Neural Networks подразумевает выполнение следующих этапов.

1). Задание исходных данных для обучения и тестирования нейронной сети. Создание таблицы исходных данных выполняется с помощью пункта меню File->New->Data Set.... В диалоговом окне задается количество входов и выходов будущей нейронной сети.



В открывшемся после диалогового окна редакторе (Data Set Editor) вводятся числовые или текстовые данные для обучения и тестирования нейронной сети.



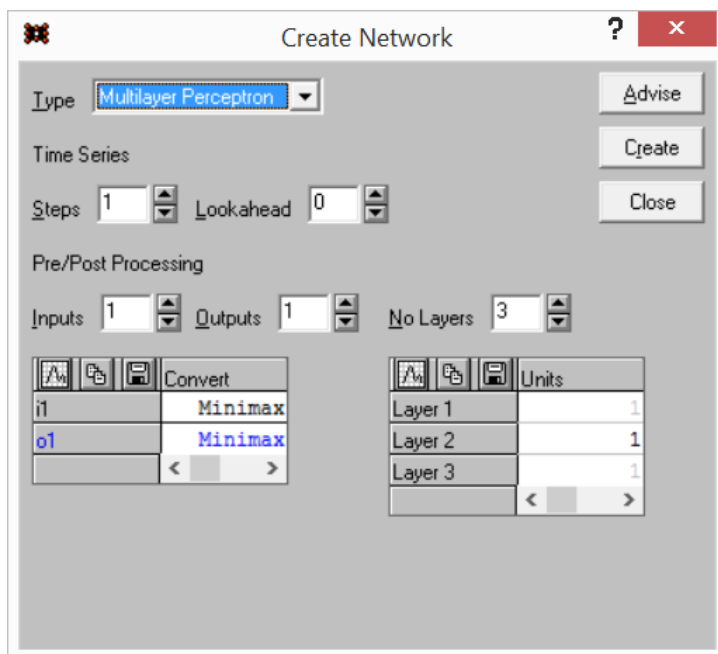
Количество столбцов созданной таблицы данных можно изменить воспользовавшись пунктом меню Edit->Variables и выбрав необходимую команду (Add (Добавить), Move (переместить), Copy (Скопировать), Delete

(удалить)). Чтобы изменить количество строк необходимо воспользоваться пунктом меню Edit->Cases с аналогичными командами Add, Move, Copy, Delete.

Столбцы таблицы данных могут быть помечены как входы нейросети (Input), выходы (Output) или как игнорируемые в процессе работы (Ignore). Для этого нужно выделить столбец, вызвать контекстное меню правой кнопкой мыши и выбрать соответствующую команду.

В ряде случаев исходные данные необходимо делить на три части: для обучения, для верификации, для тестирования. Для того чтобы обозначить часть исходных данных как данные для обучения необходимо выделить нужные строки таблицы, вызвать контекстное меню правой кнопкой мыши и выбрать команду Training. Аналогично строки исходных данных могут быть отмечены как данные для верификации (Verification, эти данные используются для контроля эффекта переобучения по ошибке верификации в процессе обучения), как данные для тестирования уже обученной сети (Test).

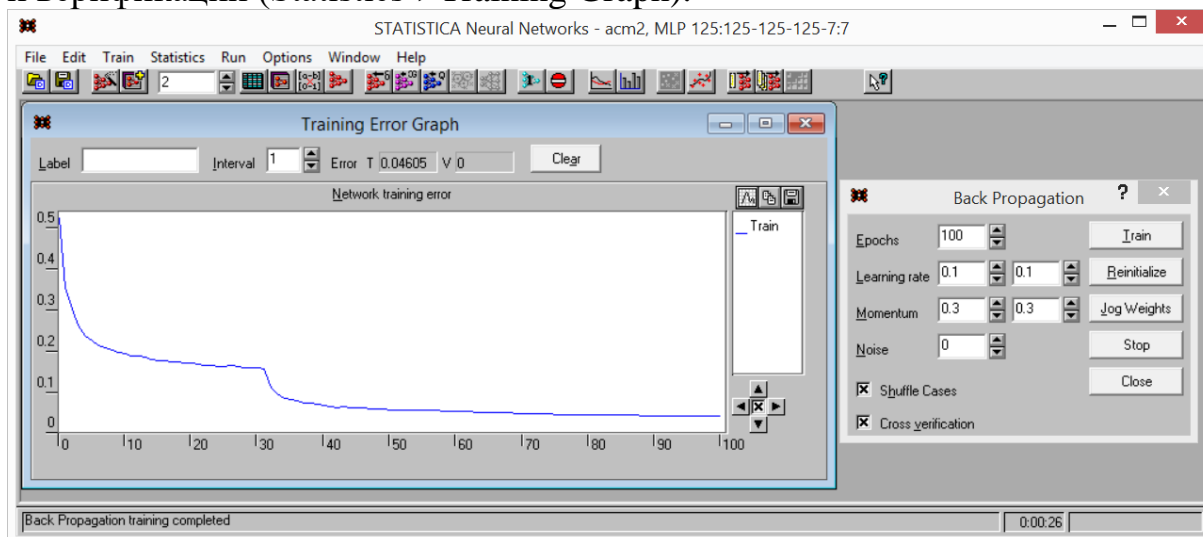
2). Создание нейронной сети. Нейронная сеть создается в Statistica Neural Networks с помощью команды File->New->Network.... В открывшемся окне задается тип нейронной сети, количество входов и выходов, количество слоев, тип предварительной обработки данных (масштабирование, замена пропущенных значений в наборе исходных данных и т.п.).



В ряде случаев оказывается полезной команда Advise, которая предлагает настройки сети по заданным исходным данным. Для создания сети необходимо нажать кнопку Create. Созданную нейронную сеть можно визуализировать с помощью команды Statistics->Network Illustration.... Для редактирования уже созданной нейронной сети предназначена команда

Edit->Network. Окно Network Editor позволяет задавать вид активационных функций в каждом из слоев нейронной сети. При необходимости изменить параметры предварительной и пост- обработки данных следует вызвать команду Edit->Pre/post processing....

3). Обучение нейронной сети. После создания (загрузки) исходных данных и создания (загрузки) нейронной сети становятся доступными алгоритмы обучения в пункте меню Train. При обучении сети в ряде алгоритмов предусмотрен расчет и построение графика ошибки обучения и верификации (Statistics->Training Graph).



Информацию по алгоритмам обучения можно найти в разделах справки программы.

4). Тестирование нейронной сети осуществляется на наборе тестовых данных, например, с помощью команд Run->Data set (кнопка Run) или Run->Single case.

Приложение 2

NNACS. Основные сведения

Программный пакет Neural network applications for control systems (NNACS) разработан на кафедре Управления и информатики НИУ «МЭИ» для выполнения учебных и несложных научно-исследовательских задач по синтезу и исследованию систем управления с применением нейросетевых алгоритмов. Пакет ориентирован на моделирование односвязных и многосвязных стационарных и нестационарных систем управления в дискретном времени. Нейронные сети могут использоваться для управления и моделирования объекта управления.

Установка и настройка программного обеспечения NNACS на рабочее место Windows XP или Windows 7

Файлы для инсталляции должны находиться в одной из локальных папок компьютера. Из этой папки необходимо последовательно:

1. Запустить **vcredist_x86.exe**, со всем согласиться и установить.
2. Запустить **ActiveTcl8.6.0.0.296563-win32-ix86-threaded.exe** (или более новую версию) для 32-битной версии Windows, настроек не менять. Для 64-битной версии Windows необходимо поставить соответствующую 64-битную версию ActiveTcl.
3. Запустить **plotchart.bat**
4. Распаковать **NNACS-1.8.2-win32.zip** (или более новую версию) [14].
5. Каталог **NNACS** перенести в папку **C:\Program Files** (или **C:\Program Files (x86)** для 64-х разрядных версий ОС). Для Windows 7 система может блокировать создание каталога и копирование в него исполняемых файлов. В этом случае необходимо создать копию каталога **NNACS** в каком-то другом месте, например, в **C:\nn**. В этом случае в переменных среды пользователя надо создать переменную с именем **NNACSSYSDIR** и значением **C:\nn\NNACS**.
6. Для файла **C:\Program Files\NNACS\bin\nnacs.tcl** сделать ярлык на рабочий стол.
7. В домашнем каталоге (**C:\Documents and Settings\Пользователь** – для Windows XP, **C:\Users\Пользователь** – для Windows 7) сделать подкаталог **labworks** для хранения параметров систем управления, нейросетей, сеансов моделирования включая данные, результаты и картинки. Можно использовать произвольный каталог, например, **C:\nn\labworks**, создав переменную окружения **NNACSUSERDIR** с соответствующим значением.
8. В этот каталог можно скопировать рабочие проекты или распаковать варианты заданий (обычно из файла **Variants.zip**).

Краткое описание пакета NNACS

При запуске пакета появляется главное меню, состоящее из пунктов, определяющих основные действия пользователя. Первое необходимое действие при работе с программой – **Перед началом Выбор/создание нового пользователя** – показывает диалоговое окно, в котором необходимо выбрать (кнопка **Выбор**) или создать проект/пользователя (кнопка **Создать** и поле ввода имени проекта /пользователя). Это действие приводит к созданию нового или выбору существующего подкаталога в каталоге **labworks**.

Дальнейшая работа с пакетом заключается в выборе одной из рабочих программ, которые будут описаны ниже. Можно одновременно запустить несколько разных программ, но не более одной программы каждого типа.

При запуске каждой из программ выбирается существующий или создается новый сеанс работы – нумерованный каталог внутри каталога проекта/пользователя. Номера состоят из трех цифр и автоматически инкрементируются при создании нового сеанса. Создание нового сеанса возможно «с нуля» (кнопка **Новый**) или путем копирования существующего сеанса (кнопка **Копия**). В последнем случае в новый сеанс копируются все файлы существующего сеанса.

В списке присутствуют только сеансы, подходящие запускаемой программе. Кроме номера сеанса в списке показывается комментарий, который пользователь задавал в окне программы.

Рабочие файлы пакета

Каждый пользователь (вариант лабораторной работы) хранит файлы в своем каталоге, расположенном в каталоге **labworks**. Внутри пользовательского каталога создаются нумерованные (тремя цифрами) подкаталоги сеансов работы с программами пакета. Каждая программа полностью управляется задаваемыми ей на вход параметрами (*.par). Обычно содержимое этого файла формируется пользователем неявно, выбирая те или иные параметры или режимы работы программы.

Параметры могут быть числовыми, а также являться ссылками на файлы описания нейросетей (*.nn), линейных звеньев (*.tf), произвольных функций (*.cof) и рядов данных (*.dat). Файлы нейросетей и рядов данных могут быть как входными, так и выходными. Линейные звенья описываются пользователем с помощью формул дискретного z^* -преобразования с параметрами и в вычислительных программах пакета не изменяются. Использование файлов произвольных функций допустимо там же, где и файлов линейных звеньев. Произвольные функции позволяют задавать последовательность линейных звеньев в разные промежутки времени для моделирования нестационарных объектов и регуляторов.

Все форматы файлов являются текстовыми. В частности, ряды данных можно загрузить и визуализировать в любой подходящей программе (Excel, MATLAB, GnuPlot и т.п.). В пакете приняты следующие обозначения для сигналов (рядов данных), образующихся в процессе моделирования или обучения:

- r (r.dat) – уставка (reference signal)
- e (e.dat) – ошибка управления (control error)
- n (n.dat) – помеха (noise signal)
- u (u.dat) – управляющий сигнал с выхода регулятора на вход объекта
- y (y.dat) – выход объекта без помехи, (ny.dat) – с аддитивной помехой (сумма сигналов u и n)
- d (d.dat) – ошибка идентификации, невязка (discrepancy)
- p (p.dat) – предсказание выхода объекта моделью (plant prediction)
- ip (ip.dat) – изображающая точка АКС (imaging point)

Моделирование контура управления осуществляется в одном из двух режимов: когда и уставка, и помеха заранее подготовлены и находятся в файлах или если и уставка, и помеха задаются формирующими фильтрами из белого шума (заданными соответствующими линейными звеньями), при этом задается длительность моделирования в дискретных отсчетах.

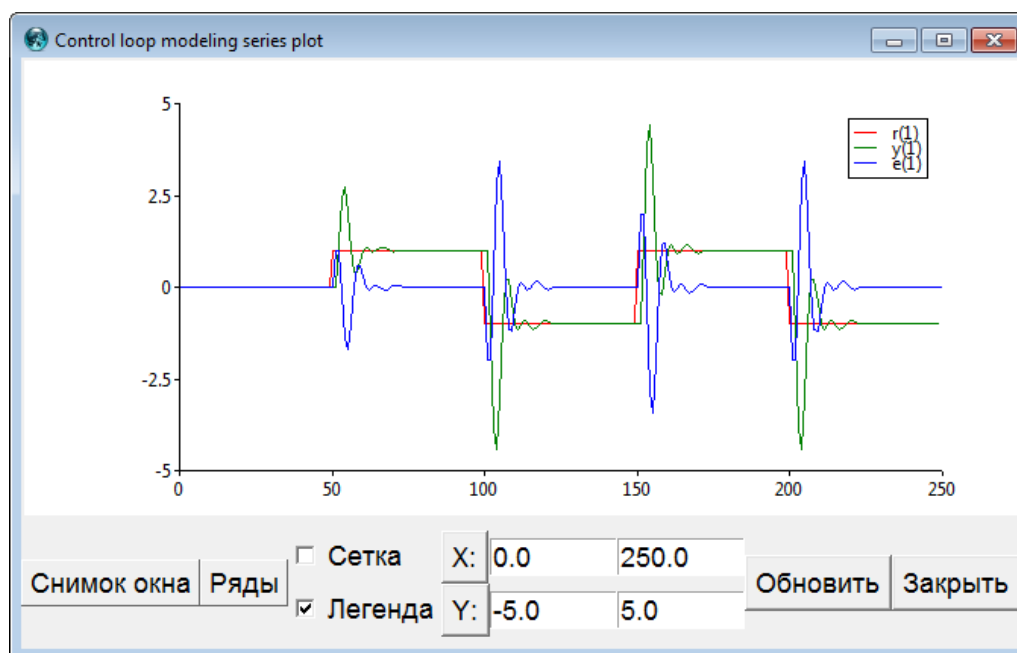
В программах пакета при задании параметров-файлов (линейные звенья, нейронные сети, ряды данных) можно задавать ссылки на файлы, сделанные в других сеансах. Например, можно сделать в сеансе 001 файл с линейным звеном, описывающим объект управления: plant.tf. После этого во всех остальных сеансах можно ссылаться на этот файл: ../001/plant.tf. Аналогично надо поступать с данными: числовые ряды обучающей и контрольной выборок, полученные при моделировании контура управления, должны быть указаны в качестве параметров программ обучения нейросетей.

Моделирование системы автоматического управления

Программа используется для моделирования традиционных и нейросетевых систем управления со стационарными и нестационарными элементами, стохастическими и детерминированными сигналами уставки и помехи. Интерактивные возможности позволяют быстро подбирать параметры элементов системы управления, визуализировать результаты моделирования, сохранять их в графических файлах.

Задание параметров элементов системы управления осуществляется при активации соответствующих элементов на схеме системы управления (**Уставка, Регулятор, Объект, Помеха**). После задания всех параметров сеанс моделирование начинается нажатием кнопки **Запустить**. Пример окна с результатами моделирования приводится ниже. Для получения

этого изображения надо нажать на кнопку **Графики**, а потом нажать контрольные точки r , y и e .

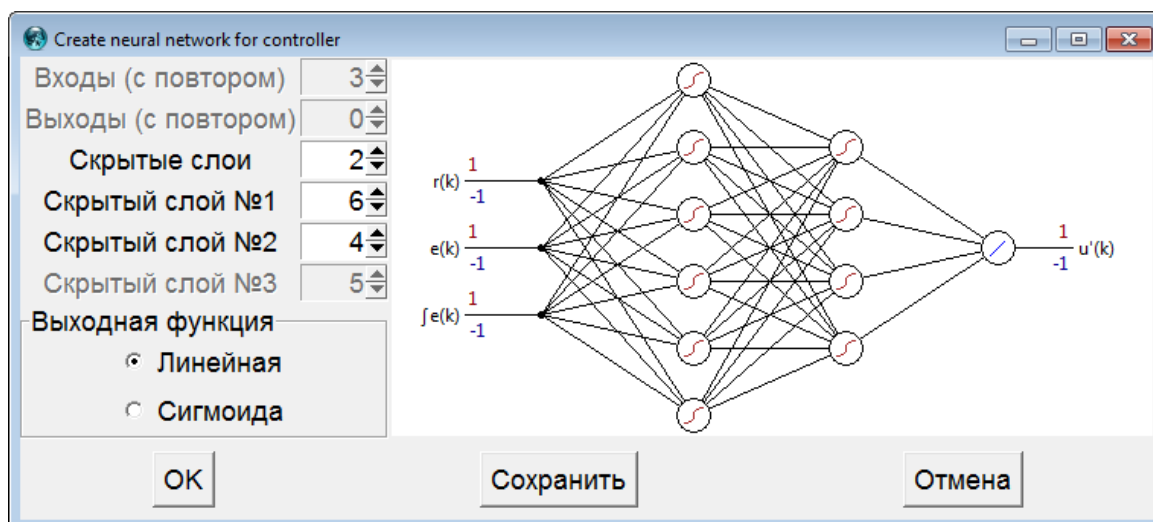


Диапазон визуализации данных можно изменить, задав другой диапазон по оси X (дискретное время) и оси Y (значение), а потом нажав кнопку **Обновить**. Кнопки названия осей X : и Y : восстанавливают полный диапазон значений, доступный для визуализации. Кнопка **Ряды** позволяет добавить в область визуализации данные (*.dat) из других сеансов.

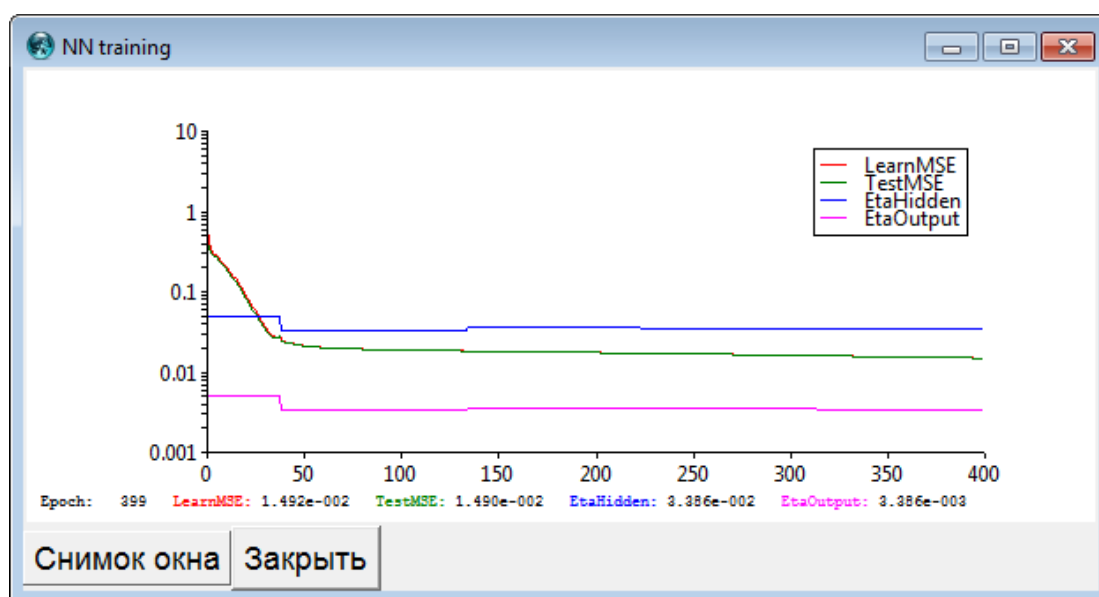
Обучение нейросетевого регулятора вне контура

Программа предназначена для обучения нейросетевого регулятора вне контура управления по примеру другого регулятора (обычно, линейного), представленного в рядах данных, полученных в сеансе моделирования контура управления с исходным регулятором.

Для обучения рекомендуется провести предварительно два сеанса моделирования, один из которых даст набор данных для обучения, а другой – для проверки. Для обучения нейросетевого регулятора обычно требуется задать наборы данных r , e , u . Кроме того, необходимо задать нейронную сеть регулятора, выбрав предварительно структуру входов. Возможны варианты $e+r$ (ошибка и уставка на входе нейросетевого регулятора) и $e+r+se$ (ошибка, уставка и интеграл ошибки на входе нейросетевого регулятора). Диалоговое окно интерактивного задания архитектуры нейронной сети регулятора приведено ниже:

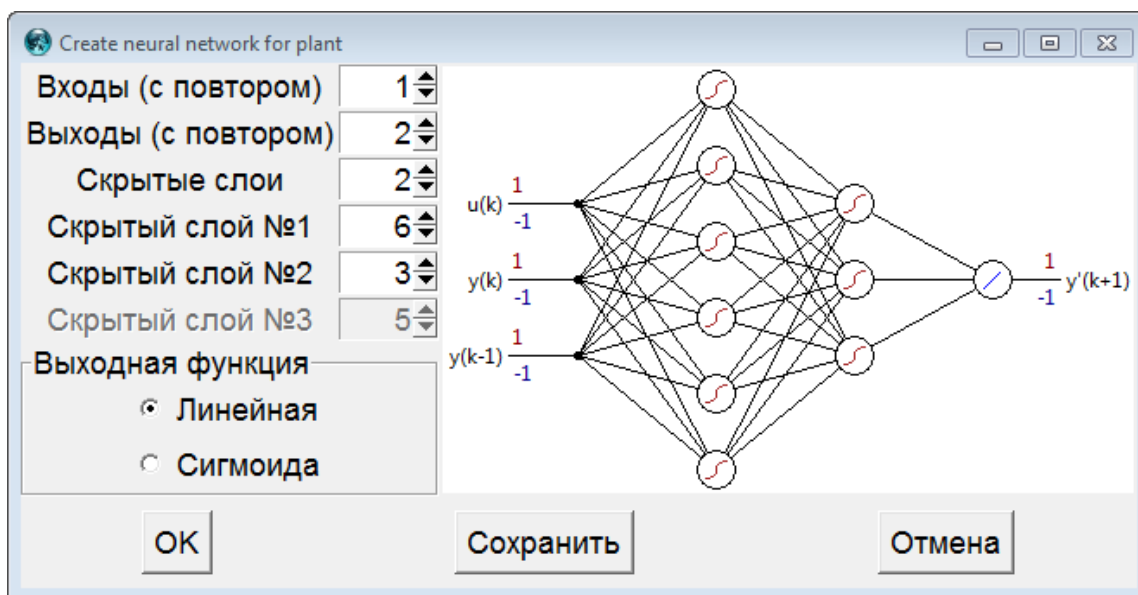


Параметры обучения и визуализации задаются в диалоговом окне, появляющемся при нажатии кнопки **Обучение**. При запуске процесса обучения (кнопка **Запустить**) появляется окно, содержащее графики ошибки и коэффициентов скорости обучения:



Обучение нейросетевой модели объекта управления

Программа используется для обучения вне контура нейросетевой модели объекта управления. В рабочем окне задаются файлы данных для обучения и проверки (**Управление** u , **Наблюдение** y или pu), а также нейронная сеть модели (**НС-О**). Окно создания нейросетевой прогнозирующей модели позволяет определить набор входов и внутреннюю архитектуру сети:

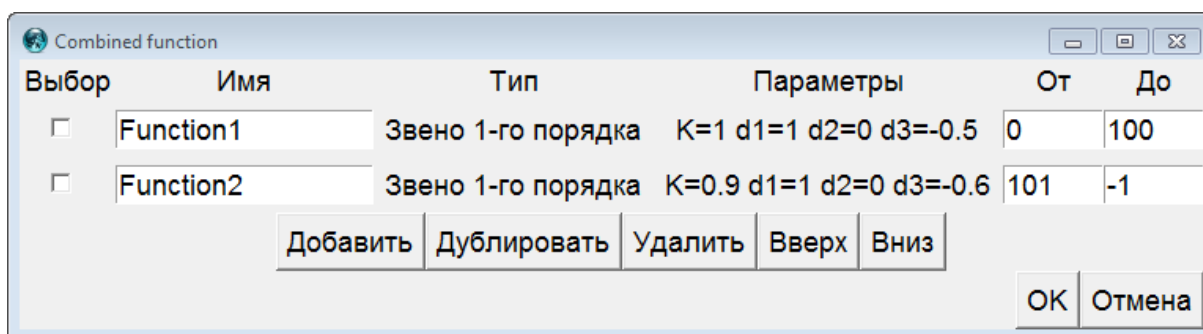


Обучение нейросетевого регулятора в контуре

Данная программа позволяет осуществить обучение нейросетевого регулятора в контуре управления с помощью синтезированной прежде нейросетевой прогнозирующей модели. В качестве начально приближения обычно выбирается обученный вне контура нейросетевой регулятор.

Обнаружение разладки объекта в контуре

В данной программе используется нейросетевая модель, а также присутствуют блоки, реализующие алгоритм кумулятивных сумм (АКС). Регулятор может быть как линейным, так и нейросетевым. Объект управления обычно задается нестационарным. Файл такого объекта должен иметь расширение .cof. При задании объекта управления с таким именем файла появляется диалоговое окно, обеспечивающее ввод нескольких линейных передаточных звеньев с разными параметрами и временем срабатывания, что и позволяет моделировать нестационарное поведение объекта:



В данной программе необходимо подобрать параметры **АКС** так, чтобы обнаружить разладку (изменение параметров объекта управления) с минимальной задержкой при достаточно большом этом средним временем ложного срабатывания.

Дополнительные сведения о пакете

В сообщениях программ пакета используются следующие обозначения:

- Plant – объект управления
- Reference – уставка
- Min/Max/Mean – минимальное/максимальное/среднее значение во временном ряде
- Dispersion – дисперсия
- StdDev – стандартное отклонение (квадратный корень дисперсии)
- RMS – среднеквадратическое отклонение
- MSE – среднеквадратическая ошибка (RMS для ошибки)
- Volume, Length – длина временного ряда
- EtaHidden, EtaOutput – скорость обучения скрытых и выходных нейронов
- Epoch – эпоха обучения
- LearnMSE, TestMSE – СКО ошибка на обучающей и контрольной выборках
- ControlMSE, IdentifMSE – СКО ошибки управления и идентификации

Вещественные числа в программе должны задаваться с десятичной точкой, а не с запятой. То есть, «3.14159» – правильно, а «3,14159» – неправильно.

Не рекомендуется использовать русские буквы и пробелы в именах файлов и каталогов.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Галушкин А.И., Судариков В.А., Шабанов Е.В. «Нейроматематика: методы решения задач на нейροкомпьютерах», Матем. моделирование, 3:8 (1991), с.93–111
2. Галушкин А.И. «Синтез многослойных систем распознавания образов», М.:Энергия, 1974. –367 с.
3. Галушкин А.И., Судариков В.А. «Адаптивные нейросетевые алгоритмы решения задач линейной алгебры» // Нейрокомпьютер – №2, 1992
4. Уоссермен Ф., «Нейрокомпьютерная техника. Теория и практика», М.: Мир, 1992. – 240 с.
5. Осовский С. «Нейронные сети для обработки информации» М.:Финансы и статистика, 2004. – 344 с.
6. Головки В.А. «Нейронные сети: обучение, организация и применение» Кн.4: Учеб. пособие для вузов / Под общей ред. А.И. Галушкина – М.:ИПРЖР, 2001. – 256 с.
7. Сигеру О., Марзуки Х., Рубия Ю. «Нейроуправление и его приложения» Кн.2: Учеб. пособие для вузов / Под ред. А.И. Галушкина, В.А. Птичкина – М.:ИПРЖР, 2000. – 272 с.
8. Галушкин А.И. «Нейронные сети: основы теории», М.: Горячая линия – Телеком, 2012. – 496 с.
9. Хайкин С. «Нейронные сети. Полный курс», М.: Издательский дом «Вильямс», 2008. – 1104 с.
10. Suykens J.A.K., Vandewalle J.P.L., de Moor B.L. «Artificial neural networks for modelling and control of non-linear systems» Springer Science & Business Media, 2012. – 235 с.
11. Елисеев В.Л. Пакет NNACS для решения учебных и исследовательских задач нейросетевого управления // Материалы VIII международной научно-практической конференции «Новые информационные технологии в образовании», Екатеринбург, 10–13 марта 2015, с.561–566.
12. Елисеев В.Л., Свириденков К.И., Филаретов Г.Ф. «Программный пакет для моделирования, анализа и синтеза нейросетевых систем управления» // Нейрокомпьютеры: разработка, применение. 2016. №7. с.74–78
13. NNACS: Neural Networks Applications for Control Systems URL: <https://github.com/evlad/nnacs>
14. Neural Networks Applications for Control Systems (binary builds) URL: https://github.com/evlad/nnacs_binaries

СОДЕРЖАНИЕ

| | |
|---------------------------------------|----|
| ВВЕДЕНИЕ | 3 |
| Теоретические основы | 3 |
| Методы обучения | 8 |
| Ускорение обучения | 12 |
| Рекомендации по литературе | 16 |
| Лабораторная работа № 1 | 17 |
| Лабораторная работа № 2 | 19 |
| Лабораторная работа № 3 | 21 |
| Лабораторная работа № 4 | 23 |
| Лабораторная работа № 5 | 24 |
| Лабораторная работа № 6 | 27 |
| Лабораторная работа № 7 | 30 |
| Лабораторная работа № 8 | 33 |
| ПРИЛОЖЕНИЯ | 36 |
| Приложение 1 | 36 |
| Приложение 2 | 39 |
| СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ | 46 |