

# Нейросетевое распознавание классов в режиме реального времени

С.П. Алёшин, Е.А. Бородина

## Введение

Адекватная математическая модель, аналитически связывающая входные факторы ССТС с выходными индикаторами её состояния, является основой построения оптимальной системы принятия решений. Однако в условиях большой размерности вектора входных данных и сложности взаимосвязей с индикаторами состояния объекта, построить адекватную многомерную зависимость сразу для всего массива данных, как правило, не удастся [1,2]. Поэтому нахождению многофакторной зависимости, как решение задачи многомерной регрессии, должна предшествовать процедура структурирования исходных данных с целью формирования однородных подмножеств. В этом случае зачастую удается добиться заданных границ производительности при построении модели регрессионного анализа для каждого из полученных подмножеств. Вместе с тем, формирование однородных подмножеств, в ряде случаев [3], совпадает с процедурой описания классов, в пространстве имеющихся признаков исследуемого объекта. Это особенно очевидно в задачах дихотомии (больной - здоровый, кондиция - фальсификат, успешный - лузер и т.п.). Тогда в динамичном, быстро меняющемся признаковом множестве, актуальна задача автоматического построения алфавита классов в режиме реального времени. В этом заинтересована, например, такая структура, как ситуационный центр быстрого реагирования [4]. Решение задачи автоматизации процедур, обеспечивающих распознавание классов в реальном времени, целесообразно осуществлять в нейросетевом формате в среде нейроэмуляторов, например, пакета технического анализа Statsoft.

## Постановка задач

Формально построение математической модели внутренней структуры индикаторов состояния объекта на основе исходных данных входного множества факторов сводится к отображению пространства факторов на пространство состояний с заданной надежностью и точностью как в прямой задаче распознавания:

$$F : X \rightarrow Y_{opt}, X \subset \mathfrak{R}^m, Y_{opt} \subset \mathfrak{R}, \quad (1)$$

так и в обратной (адаптация множества признаков под целевое состояние объекта)

$$F_0 : Y_Z \rightarrow X_{Z_0} \subset \mathfrak{R}^m, Y_Z \subset \mathfrak{R}, \quad (2)$$

путем нахождения соответствующих функционалов преобразования  $F$  и  $F_0$ , где  $X$  – множество выборок признаков описания состояния объекта;  $X_{Z_0}$  – множество выборок признаков описания целевого состояния объекта;  $Y_{opt}$  – верифицированное выходное значение функции регрессии;  $Y_Z$  – требуемое выходное значение функции регрессии.

При конечном массиве обучающих выборок  $X^n = \{x_1, x_2, \dots, x_n\} \subset X$ , выбранной функции расстояния между примерами  $\rho(x, x^*)$  (евклидово расстояние), необходимо на первом этапе разделить массив данных на непересекающиеся подмножества, так, чтобы каждое состояло из объектов, близких по метрике  $\rho(x, x^*)$ , а объекты разных подмножеств существенно отличались между собой, то есть оптимизировать суммы:

$$\sum_i^n d(X_i, \bar{X}_j) \Rightarrow \min; \sum_j^K d(\bar{X}_h, \bar{X}_j) \Rightarrow \max, \quad (3)$$

$$\bar{X} = (x_1, \dots, x_n) \in X; \bar{E} = (k_1, \dots, k_j) \in K,$$

где  $X$  – множество выборок признаков описания инвестиционных состояний;  $K$  – множество номеров кластеров;  $\bar{X} = (x_1, \dots, x_n) \in X; \bar{E} = (k_1, \dots, k_j) \in K$  – множества значений входных факторов и номеров классов;  $h, j = 1, 2, \dots, K$  – средние значения в кластерах;  $d(X_i, \bar{X}_j)$  – расстояние между объектом и центром кластера;  $d(\bar{X}_h, \bar{X}_j)$  – расстояние между центрами кластеров.

Алгоритмическая реализация группирования однородных данных на входе по выбранной метрике осуществляется решением задачи классификации без учителя в нейросетевом формате [1,2]. Что является одной из обеспечивающих распознавание процедур. Формализуем и решим эту задачу в нейросетевой среде.

Рассмотрим пространственный вариант набора входных факторов гипотетического исследуемого объекта из  $m$  точек  $\{x^p\}$  в  $n$ -мерном пространстве. Необходимо разбить множество точек  $\{x^p\}$  на  $k$  кластеров согласно условия (3) для выбранной меры близости (квадрат евклидова расстояния). Для этого необходимо найти  $k$  точек  $\alpha^l$  таких, что,

$$D = \sum_{l=1}^k \sum_{x \in P_l} \|\alpha^l - x\|^2 \text{ минимально.}$$

### Решение задачи

1. Зададимся набором начальных точек  $\alpha^l$  из исходного множества факторов исследуемой ССТС и разобьем множество точек  $\{x^p\}$  на  $k$  кластеров по правилу  $P_l = \{x : \|\alpha^l - x\| < \|\alpha^q - x\|, \forall l \neq q\}$

2. По полученному разбиению вычислим новые точки  $\alpha^l$  из условия минимальности  $D_l = \sum_{x \in P_l} \|\alpha^l - x\|^2$

3. Обозначим число точек в  $i$ -ом кластере через  $|P_i|$ .

4. Запишем решение выбора точек на втором шаге алгоритма в виде

$$\alpha^i = \frac{1}{|P_i|} \sum_{x \in P_i} x$$

5. Выполняем процедуры 1-3 алгоритма до тех пор, пока набор точек  $\alpha^l$  не перестанет изменяться.

6. Фиксируем сеть способную для произвольной точки  $x$  вычислить квадраты евклидовых расстояний от этой точки до всех точек  $\alpha^l$  и отнести ее к одному из  $k$  кластеров.

7. Ответом является номер нейрона, выдавшего минимальный сигнал.

Созданные для каждого из кластеров отдельные нейронные сети, позволяют получить физическую модель связи входных факторов с индикаторами состояния объекта. При этом индикаторы могут соответствовать как текущему состоянию так и целевому.

Таким образом, технология формирования компактных, однородных подмножеств во входных данных алгоритмически формализована и может быть реализована

существующими инструментальными средствами пакетов технического анализа [3].

Таким образом, после группировки входных данных как компактных множеств в многомерном пространстве на основе выбранной метрики близости, новые наблюдения можно классифицировать по принадлежности к тому или иному кластеру с минимальным значением меры близости. Это эквивалентно, для решаемой задачи, процедуре описания классов на языке их информативных признаков и является следующей обязательной, обеспечивающей распознавание, процедурой.

Верифицированное заключение (при наличии времени) может быть получено с привлечением предметного специалиста. Следующей обязательной процедурой является выбор и формализация решающего правила.

### Формализация решающего правила

В условиях многомерности входных факторов, неполноты данных о их взаимном влиянии, неопределенности исходной информации целевые критерии для решения задач поддержки принятия решений, как правило, формализуются в виде лингвистических переменных или в виде правил логического вывода относительно эффективности функционирования объекта. Массив выборок  $X^n = \{x_1, x_2, \dots, x_n\} \subset X$  совместно с соответствующим массивом классов  $\vec{E} = (k_1, \dots, k_j) \in K$  используется для обучения в задаче распознавания актуальных состояний объекта исследования. Эту процедуру можно выразить в требовании, а именно оптимизировать функционал процесса распознавания текущего состояния объекта:

$$\mathfrak{S} = F[d(\Omega_p), D(\Omega_p, \Omega_g), (L(\omega, \{\omega_g\}))] \quad (4)$$

где  $d(\Omega_p)$  – расстояние между объектами (признаками) внутри класса по заданной метрике, вычисляется с помощью следующего выражения:

$$d(\Omega_p) = \sqrt{\frac{1}{k_p} \frac{1}{k_p - 1} \sum_{k=1}^{k_p} \sum_{i=1}^{k_p-1} \sum_{j=1}^N e_j (x_{pk}^{(j)} - x_{pl}^{(j)})^2}, \quad (5)$$

$D(\Omega_p, \Omega_g)$  – расстояние между объектами (признаками) в разных классах по заданной метрике, вычисляется с помощью следующего выражения:

$$D(\Omega_p, \Omega_g) = \sqrt{\frac{1}{k_p} \frac{1}{k_g} \sum_{k=1}^{k_p} \sum_{i=1}^{k_g} \sum_{j=1}^N e_j (x_{pk}^{(j)} - x_{gl}^{(j)})^2}, \quad (6)$$

$L(\omega, \{\omega_g\})$  – правило отнесения объекта  $\omega_g$  к соответствующему классу;  $e_j$  – индикатор наличия (отсутствия) признака ( $e_j = 1, 0$ );  $\{\omega\}$  – множество объектов в классе  $(\omega_{pk}, \omega_{gl})$ .

$d^2(\omega_{pk}, \omega_{gl}) = \sum_{j=1}^N (x_{pk}^{(j)} - x_{gl}^{(j)})^2$  – заданная метрика. Тогда решение (4) достигается минимизацией (5), максимизацией (6):

$$\sum_i^n d(X_i, \bar{X}_j) \Rightarrow \min; \sum_j^K D(\bar{X}_h, \bar{X}_j) \Rightarrow \max \quad (7)$$

и реализацией преобразования  $L(\omega, \{\omega_g\}) \rightarrow \omega_g \in \Omega_k$ , в соответствии решающего правила:

$$\omega_g \in \Omega_k, \quad \text{если} \quad L(\omega, \{\omega_g\}) = \sup_i L(\omega, \{\omega_i\}), \quad (8)$$

где  $\vec{X} = (x_1, \dots, x_n) \in X, \vec{E} = (k_1, \dots, k_j) \in K$ ;  $h, j = 1, 2, \dots, K$  – средние значения в кластерах;  $d(X_i, \vec{X}_j)$  – расстояние между объектом и центром кластера;  $D(\vec{X}_h, \vec{X}_j)$  – расстояние между центрами кластеров;  $X$  – множество выборок признаков состояний;  $K$  – множество номеров кластеров;  $\vec{X} = (x_1, \dots, x_m) \in X, \vec{E} = (k_1, \dots, k_j) \in K$ .

### Алгоритмическая реализация решающего правила

1. Полагаем все веса элементов равными нулю.
2. Проводим цикл предъявления примеров. Для каждого примера выполняется следующая процедура:
  - 2.1. Если сеть выдала правильный ответ, то переходим к шагу 2.4.
  - 2.2. Если на выходе персептрона ожидалась единица, а был получен ноль, то веса связей, по которым прошел единичный сигнал, уменьшаем на единицу.
  - 2.3. Если на выходе персептрона ожидался ноль, а была получена единица, то веса связей, по которым прошел единичный сигнал, увеличиваем на единицу.
  - 2.4. Переходим к следующему примеру. Если достигнут конец обучающего множества, то переходим к шагу 3, иначе возвращаемся на шаг 2.1.
3. Если в ходе выполнения второго шага алгоритма хоть один раз выполнялся шаг 2.2 или 2.3 и не произошло заикливания, то переходим к шагу 2. В противном случае обучение завершено.

При этом правило, реализующее приведение выходной ошибки обучения ко входу каждого фактора, имеет строгое математическое описание [1]. Это крайне важно при адаптации входного вектора факторов к целевому состоянию (вектору индикаторов) объекта, поэтому зафиксируем необходимые аналитические соотношения в обозначениях исходной задачи. Ограничимся архитектурой многослойного персептрона и единичной обучающей выборкой. Если мы располагаем целевым вектором индикаторов  $Yz^k$ , то функция невязки примет вид:

$$E(W) = \frac{1}{2} \sum_{i=1}^M (y_i(\alpha) - y_{iz})^2.$$

Модификация синаптических коэффициентов осуществляется на основе градиентного метода:

$$W^{t+1} = W^t - \eta \cdot \text{grad}E(W^t).$$

Компоненты вектора градиента функции невязки рассчитываются согласно выражений, для расчета элементов многослойного персептрона[1]:

$$E(W) = \frac{1}{2} \sum_{i=1}^M (f(\sum_{l=0}^L w_{il}^{(2)} \cdot f(\sum_{j=0}^N w_{lj}^{(1)} x_j)) - y_z)^2$$

Выходы 1 и 2 слоев нейронов определяются из выражений:

$$g_l = f\left(\sum_{j=0}^N w_{lj}^{(1)} \cdot x_j\right), l = 1, 2, \dots, L;$$

$$y_i = f\left(\sum_{l=0}^L w_{il}^{(2)} \cdot g_l\right) = f\left(\sum_{l=0}^L w_{il}^{(2)} \cdot f\left(\sum_{j=0}^N w_{lj}^{(1)} \cdot x_j\right)\right), i = 1, 2, \dots, M$$

$$E(W) = \frac{1}{2} \sum_{i=1}^M \left(f\left(\sum_{l=0}^L w_{il}^{(2)} g_l\right) - y_z\right)^2 = \frac{1}{2} \sum_{i=1}^M \left(f\left(\sum_{l=0}^L w_{il}^{(2)} \cdot f\left(\sum_{j=0}^N w_{lj}^{(1)} x_j\right)\right) - y_z\right)^2$$

Соотношения для частных производных функции невязки по весам нейронов выходного слоя имеют вид:

$$\frac{\partial E(W)}{\partial w_{il}^{(2)}} = (y_i - y_{iz}) \cdot \frac{\partial f(a_i^{(2)})}{\partial a_i^{(2)}} \cdot g_l, \text{ где } a_i^{(2)} = \sum_{l=0}^L w_{il}^{(2)} \cdot g_l, a$$

$$b_i^{(2)} = (y_i - y_{iz}) \cdot \frac{\partial f(a_i^{(2)})}{\partial a_i^{(2)}}.$$

Тогда соответствующий компонент вектора градиента выглядит следующим образом:

$$\frac{\partial E(W)}{\partial w_{il}^{(2)}} = b_i^{(2)} \cdot g_l.$$

Компоненты градиента для весов предпоследнего слоя определяются по методике дифференцирования сложной функции [1]:

$$\frac{\partial E(W)}{\partial w_{lj}^{(1)}} = \sum_{i=1}^M (y_i - y_{iz}) \cdot \frac{\partial y_i}{\partial g_l} \cdot \frac{\partial g_l}{\partial w_{lj}^{(1)}} = \sum_{i=1}^M (y_i - y_{iz}) \cdot \frac{\partial f(a_i^{(2)})}{\partial a_i^{(2)}} \cdot w_{il}^{(2)} \cdot \frac{\partial f(a_l^{(1)})}{\partial a_l^{(1)}} \cdot x_j,$$

$$\text{где } b_l^{(1)} = \sum_{i=1}^M (y_i - y_{iz}) \cdot \frac{\partial f(a_i^{(2)})}{\partial a_i^{(2)}} \cdot w_{il}^{(2)} \cdot \frac{\partial f(a_l^{(1)})}{\partial a_l^{(1)}}.$$

В компактном виде весовые коэффициенты всего синаптического пространства обученной модели можно представить выражением:

$$\frac{\partial E(W)}{\partial w_{lj}^{(1)}} = b_l^{(1)} \cdot x_j.$$

Анализ приведенных выражений для производных функции ошибки по весам различных слоев сети позволяет получить правило расчета компонентов вектора градиента от функции ошибки по параметрам (весовым коэффициентам)  $E(W)$  для любого скрытого слоя нейронной сети:

$$\frac{\partial E(W)}{\partial w_{ij}^{(k)}} = b_i^{(k)} \cdot x_j, \quad (9)$$

где  $x_j$  — сигнал на входе в сеть,  $b_i^{(k)}$  — величина погрешности обучения ( $y_i - y_{iz}$ ), перенесенная с выхода сети к входу (к ее предыдущим слоям).

### Обеспечение временных ограничений

С учетом того, что по условию задачи эти процедуры ограничены во времени, необходимо реализовать технологию, позволяющую удовлетворить временным ограничениям и допустимым ошибкам на тестовом множестве. Максимальную автоматизацию принятия решений обеспечивает разработка библиотеки программ (On-Line System Neural Network), позволяющей моделировать и обучать многослойные нейронные сети анализа исследуемых объектов в режиме реального времени. Обучение нейронных сетей с точки зрения используемого математического аппарата эквивалентно задаче многомерной оптимизации[1]. В разряд ограничений отводятся: время работы алгоритма и объем требуемых ресурсов. Предлагается следующая структура мер ускорения принятия решений (рис.1.):

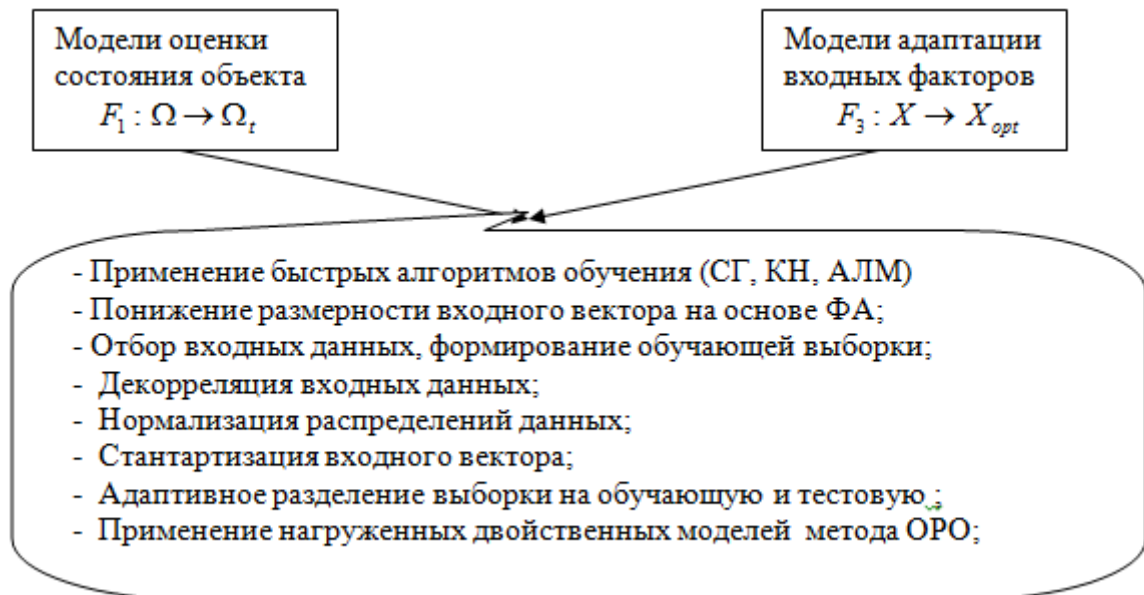


Рис. 1. – Структура ускоряющих процессов

При такой структуре моделирования процессов или объектов, разрешение проблемы оперативного реагирования при принятии решений, сводится к реализации указанных функций путем создания алгоритмов и программ отображения массива входных данных на состояния и динамику их эволюции под влиянием управляющих факторов с учетом директивного времени. Если эта задача имеет решение, то автоматизированная система (АСППР), оперативно (время принятия решения определяется техническими возможностями вычислителей) идентифицирует состояние объекта, прогнозирует его динамику и определяет управляющие факторы, адекватные целевому (требуемому) состоянию.

Для анализа и моделирования базовых отмеченных процессов целесообразно исследовать возможности пакета технического анализа данных StatSoft с нейросетевым модулем STATISTICA Neural Networks [3]. Тогда синтез моделей реализуется в пространстве процедур ускоряющих адаптацию моделей в границах выбранных дисциплинирующих условий.

Логическим продолжением решаемой задачи является комплексное использование всех доступных ускоряющих процедур из пакета Statistica [4] с последующим выделением продуктивных нейросетей из всего ансамбля обученных моделей. Так как все описанные процедуры представляют собой относительно самостоятельные процессы, их совместное использование целесообразно представить в виде циклов в общем алгоритме адаптации:

- 1 Формируем исходные данные. Определяем рабочую и тестовую выборки.
- 2 Определяем директивное время для рассматриваемой предметной области и допустимые ошибки.
- 3 Выбираем ЦФ и адаптируем её к нейросетевому формату.
- 4 Устанавливаем границы возможных изменений элементов входного вектора управляющих факторов.
- 5 Задаем метод обучения в исходном ансамбле моделей с выбором, архитектуры, сложности, функций активации.
- 6 Выбираем класс задач и проводим обучение исходных моделей. Фиксируем время решения и ошибки обучения.
- 7 Сравниваем полученные время и ошибку с директивными для данной задачи (выбранной предметной области). Если условие соответствия выполнено, то переход на шаг 11, если нет – 5, а при условии  $n = N_{\max}$  – переходим на шаг 8. Здесь  $n$  и  $N_{\max}$  соответственно порядковый номер исходных методов обучения и их максимальное число.
- 8 Формируем исходные данные для понижения размерности.
- 9 Проводим факторный анализ, ранжируем входы по информативности, исключаем последний элемент мажоритарного ряда. Переходим на шаг 6 и 7. При выполнении условия соответствия – переход на шаг 11, в противном случае – 9. При условии  $m = M_{\max}$  – переходим на шаг 10. Здесь  $m$  и  $M_{\max}$  соответственно, номер исключенного фактора и их максимальное число.
- 10 При наличии времени переходим к экспертным оценкам и после анализа принимаем их как элементы решения данной задачи.

Оценки вероятности используются непосредственно, так как решается задача отнесения набора признаков объекта исследования к наиболее вероятному классу. Так как задача распознавания рассматривается как в прямой трактовке так и в обратной [5], то необходимо учесть имеющиеся особенности при решении обратной задачи или задачи распознавания такого набора факторов, который порождает требуемое состояние объекта исследования.

### **Особенности подстройки входного вектора к требуемому выходу**

Для этого случая метод обратного распространения ошибки применяется для подстройки входного вектора через градиент функции ошибки по входным сигналам сети [6]. В основе решения этой задачи лежит приведение ошибки обучения нейронной сети к входному слою нейронов и расчету частных производных градиента функции ошибки по параметрам нейросети и входным факторам. Метод позволяет вычислять производные выходной функции невязки текущего и требуемого состояний объекта по каждому элементу вектора входных факторов и запоминать их. При обратном функционировании эти, ранее вычисленные производные, участвуют в вычислениях градиента функции ошибки по входным сигналам сети.

Эти процедуры выполняются в комплексной схеме нейроуправления с оценками и сохранением в памяти на один такт производных выходных функций каждого элемента всего синаптического пространства нейросети. Технологию инструментальной реализации этих задач проследим на примере известных способов нейроуправления [7].

Адаптивная система контроллера решает задачу отслеживания и сравнения текущего значения выхода с требуемым, заданным вектором уставки. В схемах нейроуправления функции учителя выполняет контроллер, оценивающий невязку текущего и целевого

состояния объекта исследования и определяющий сигнал управления  $Z(t)$ . Обучение – согласно алгоритма обратного распространения ошибки, при этом ошибка сети  $E^*$  определяется разностью уставки и текущего значения выхода эмулятора.

На рис.2 представлена одна из схем нейроуправления объектом. Нейроэмулятор выполняет функции адаптивной модели управляемого объекта. На его входы поступают текущие и задержанные во времени значения векторов управления  $Z(t), Z(t - \theta), Z(t - 2\theta), \dots$  и значения разности между векторами входа и уставки  $Y(t) - U(t), Y(t - \theta) - U(t - \theta), \dots$ . Здесь  $\theta$  - величина шага задержки. Выходом нейроэмулятора является ожидаемое значение выхода управляемого объекта  $Y^*$ , а также значение вектора ошибки  $E$  для обучения нейроконтролера.

Величина  $E^*(t) = Y(t) - Y^*(t)$  представляет вектор ошибки, который используется в алгоритме обучения нейроэмулятора. Ошибка, поступающая в схему контролера поступает сигнал, представляющий отображение ошибки на выходе сети приведенной ко входу нейроэмулятора. Применим каскадную схему двух нейросетей сети, одна из которых имитирует исследуемый объект, а вторая отслеживает и минимизирует интегральную невязку индикаторов текущего и целевого состояний объекта исследования путем формирования соответствующих входных величин [7].

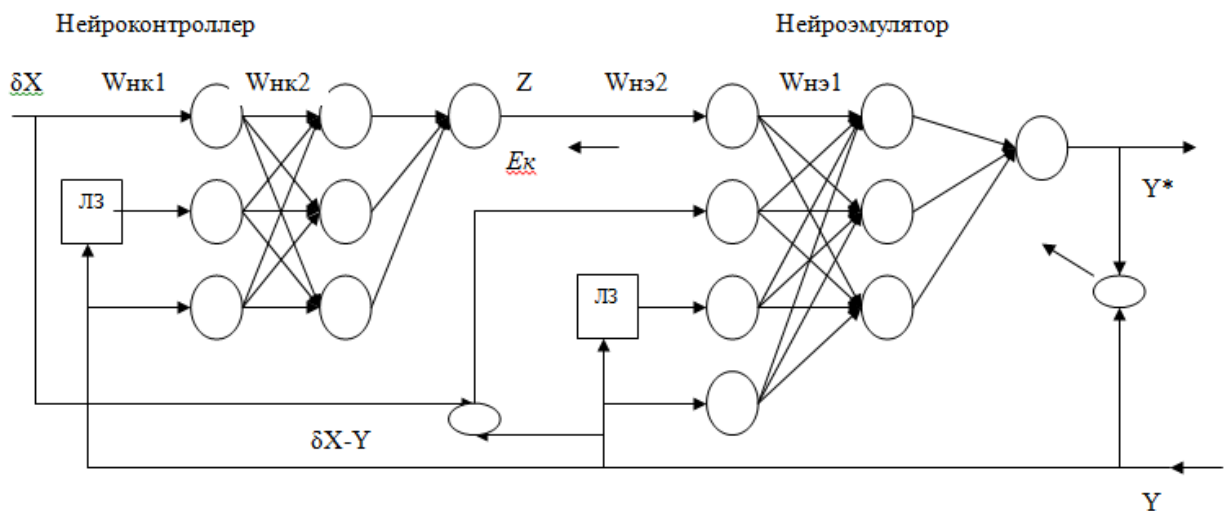


Рис. 2. – Функциональная схема нейроуправления:

$Y(t)$  – текущие значения процесса на выходе;  $Y(t - \theta)$  – текущие значения процесса задержанные на один такт  $\theta$ ;  $U(t) - Y(t)$  – текущее значение невязки на выходе системы;  $Z(t)$  – текущее управляющее воздействие.

Выходом нейроэмулятора являются текущие значения индикаторов состояния модели объекта  $Y^*(t)$ .

При обучении НЭ осуществляется модификация синаптического пространства путем подстройки весовых коэффициентов в направлении антиградиента невязки [6]. Таким образом, как в прямой так и в обратной задаче распознавания организация и выполнение обучающих процедур базируется на одних и тех же базовых функциях (формирование выборки, описание классов, синтез решающего правила, его реализация, верификация результата). Поэтому качество результатов исследования оценим по степени адекватности нейросетевых моделей и достигнутой при этом производительности.

Моделирование градиентного спуска с оценкой частных производных как по параметрам сети так и по входным сигналам, представляющим пространство признаков текущего состояния объекта, показало устойчивую сходимость итерационного процесса обучения сети с приемлемыми для практических задач показателями качества (Рис.3).



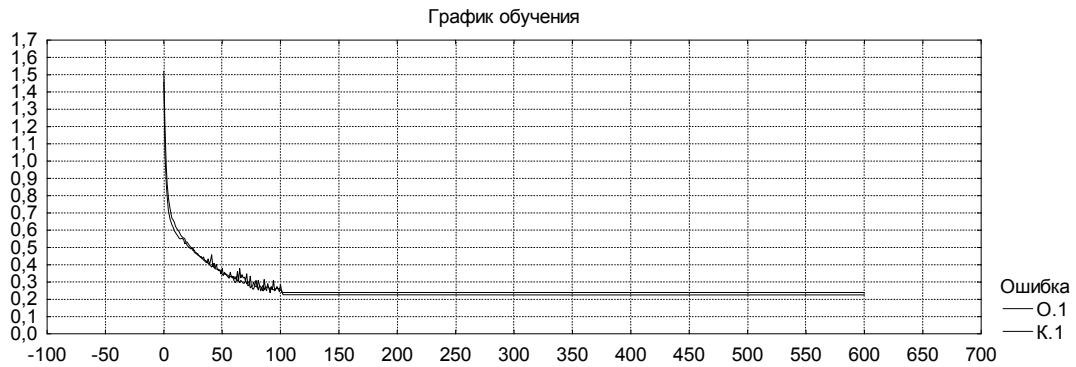


Рис. 3. – График сходимости процесса обучения

Для различных условий моделирования получен ансамбль продуктивных нейросетей, которые могут использоваться в отдельных приложениях на основе их сохранения в формате основного кода.

### Результат моделирования распознающего устройства.

В качестве инструмента моделирования использовался модуль Statistika Neural Network пакета технического анализа данных Statistika. Для кластеризации и распознавания состояний объекта исследования на основе массива данных формировалась последовательность примеров максимально возможной размерности, формируя словарь признаков. Работа нейросети в режиме кластеризации позволяет формировать алфавит классов исследуемых объектов на языке этого словаря.

Целевое состояние объекта характеризуется набором значений тех же пространств информативных признаков, которые формируются экспертом предварительно, до начала эксперимента.

Так как каждому классу соответствует свой набор показателей, то процедура классификации объекта свелась к анализу пространства признаков текущего состояния и сравнения результатов анализа с описаниями выбранных классов. В нейросетевом формате задача распознавания классов текущего состояния исследуемого субъекта решалась с использованием известных правил (например, дельта - правила [1]). При наличии двух классов состояний объектов (исправно-неисправно) формирование обучающего множества не вызвало затруднений. Реализована технология моделирования в среде нейроэмуляторов по методике [3,4], что позволило получить ансамбль моделей (Рис.4):

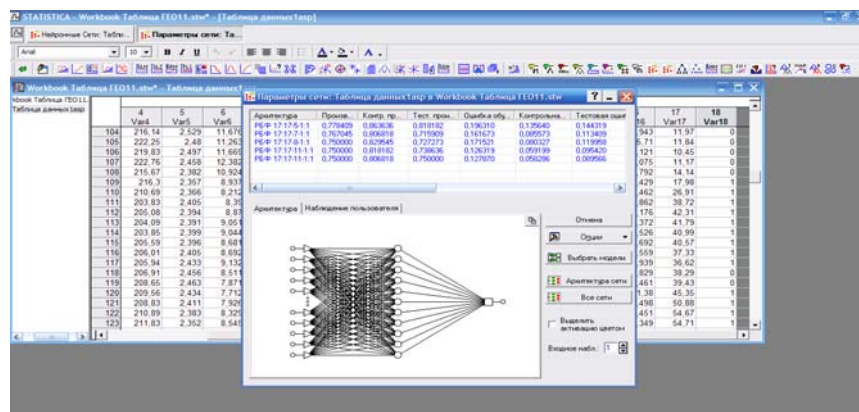


Рис. 4. – Результаты моделирования классификатора

Модели на основе радиально-базовых функций в целом подтверждают вывод о реализуемости базовых функций, но мощность обучающей выборки в приведенных примерах не позволила достичь производительности по условию задачи. Это видно из эксперимента на ансамбле РБФ-сетей (рис. 5.,6)..

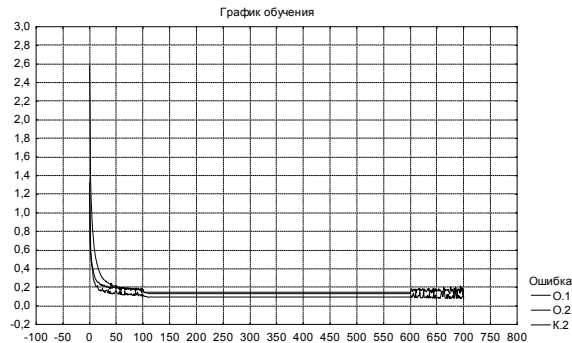


Рис.5. – График обучения модели (Вариант 1)

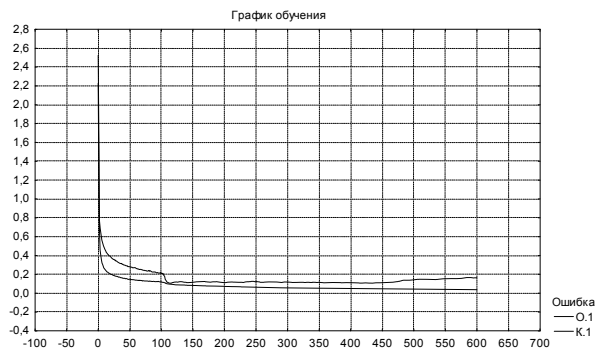


Рис.6. – График обучения модели (Вариант 2)

Производительность моделей (характер динамики обучения) имеет устойчивый вид асимптотических кривых, зависящих от репрезентативности (представительности) обучающего множества (Варианты 1,2), что позволяет улучшать модели при пополнении базы данных.

Как видно из графиков, применение представленной технологии, сокращает время адаптации нейросетевых моделей исследуемых процессов при сохранении их адекватности в допустимых границах и позволило получить устойчивую сходимость итерационного процесса модификации синаптического пространства. Число эпох не превышает нескольких сотен, что в пересчете на общие временные затраты соответствует нескольким десяткам секунд. Массивы привлеченных данных не только из технической области, но и из экономики, медицины позволяют эти временные интервалы привести в соответствие с директивным временем принятия решений для этих предметных областей и подчеркнуть инвариантность технологии к типам входных данных.

Следует учесть также, что показатели производительности моделей зависят одновременно от мощности обучающей выборки, сложности сети, её типа и архитектуры.

Таким образом, получено теоретическое и практическое подтверждение вывода о возможности решения поставленной задачи в реальном времени при сохранении адекватности моделей в заданных границах и корректных допущениях.

### **Заключение**

1. Нейросетевой базис для построения однородных подмножеств, как классов состояний объектов на языке их информативных признаков, позволяет автоматизировать реализацию базовых функций распознавания: формирование словаря признаков и алфавита классов, реализацию решающего правила, и адаптацию входов в режиме реального (директивного) времени.

2. Инструментальная продуктивная автоматизация базовых функций распознавания возможна и целесообразна с применением стандартных нейроэмуляторов существующих и перспективных пакетов технического анализа данных.

3. Адекватность и надежность моделей реализуемых процессов установлена на основе анализа ошибок на тестовом множестве данных, не принимающего участия в обучении, что обеспечивает состоятельность сделанных выводов.

### **Литература**

1. Хайкин, С. Нейронные сети: полный курс; пер. с англ. / С. Хайкин ; 2-е изд. – М. : Издательский дом «Вильямс», 2006. – С. 1104.

2. Горбань, А.Н. Обучение нейронных сетей / А.Н. Горбань. – М. : Изд-во СССР-США СП «ParaGraph», 1990. – С. 160.

3. Боровиков, В.П. STATISTICA NN : техническое описание / В.П. Боровиков. – М.: Мир, 1999. – С. 239.

4. Алёшин С.П. Ситуационные центры быстрого реагирования: принятие решений в среде нейроэмуляторов / С.П. Алёшин // Системы управления, навигации и связи – 2011. – № 1 (17). – С. 240 – 247.

5. Ляхов А. Л., Алёшин С.П., Бородин Е.А. Нейросетевая модификация текущего пространства признаков к целевому множеству классов. 11- Міжнародна науково-практична конференція «Нейросітьові технології і їх застосування»: Зб. наук. праць / под ред. проф. С. В. Ковалевского. – Краматорск: ДДМА – 2012. – С. 93 – 99.

6. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере // Новосибирск: Наука, 1996.- 276 с.

7. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления: Учеб. пособие для вузов. – М.: Высш. школа 2002. – 183 с.