

МОНОГРАФИЯ

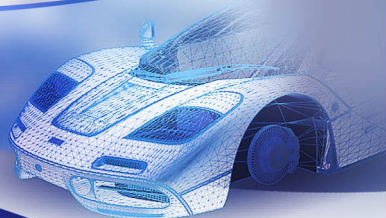
НАУЧНОЕ

КРУЖЕНИЕ

СОВРЕМЕННОГО
ЧЕЛОВЕКА

ТЕХНИКА. ИНФОРМАТИКА. АРХИТЕКТУРА
МЕДИЦИНА. СЕЛЬСКОЕ ХОЗЯЙСТВО

2019





ГЛАВА 2. DOI: 10.30888/2663-5569.2019-02-01-002

**ФОРМАЛИЗАЦИЯ И ОПТИМИЗАЦИЯ ФИСКАЛЬНО-НАЛОГОВОЙ
НАГРУЗКИ РЕГИОНА В НЕЙРОСЕТЕВОМ БАЗИСЕ****Введение**

В современных условиях, когда набирает силу новый технологический уклад, основанный на внедрении в производство и управление новых информационных технологий (нанотехнологии, микробиология, искусственный интеллект, тотальные компьютерные коммуникации и др.) представляется своевременной задача автоматизации контроля и управления фискально-налоговой нагрузкой (ФНН) региона. Это вытекает из необходимости постоянного поиска компромисса в пространстве динамичных параметров и состояний в системе налогообложения и контроля для оптимального роста экономики.

Действительно, экономическая ситуация в регионе существенно зависит всего от одного фактора. Этот фактор – налогово-административная нагрузка на бизнес. Если нагрузку сделать минимальной и таким образом полностью снять с бизнеса всю налоговую и административную нагрузку, то произойдёт расцвет бизнеса в регионе.

Но в этом случае, администрации региона, оставшейся без налогов, нечем будет ремонтировать дороги, платить зарплаты учителям, содержать аппарат управления.

Если налогово-административную нагрузку максимизировать и наложить на бизнес непосильную налоговую и административную нагрузку, то бизнес умрет или уйдет в «тень». Налогов опять не будет.

Выход такой – найти и постоянно удерживать оптимальное значение налогово-административной нагрузки на бизнес (величину платежей, отчетов, проверок, штрафов и т.д.) при которой будет достигнут оптимальный баланс между возможностью бизнеса успешно развиваться и достаточным наполнением регионального бюджета.

Это решаемая проблема, если подойти к ее преодолению как к решению задачи нелинейной оптимизации в многомерном пространстве факторов и состояний высокой размерности [1, 2]. При этом необходимо выделить экспертным путем информативные факторы, поставить их в соответствие со значениями налогово-административной нагрузки на примерах предыстории (ретроспективные данные), а строгие аналитические связи установить и зафиксировать путем принудительного обучения нейросетевых моделей на



ретроспективной выборке.

Таким образом, предметный эксперт описывает задачу на вербальном уровне в терминах фискально-налоговой парадигмы, а формализацию задачи и ее инструментальное решение – специалист по нейросетевым технологиям.

Современные программные платформы (Matlab, Statistica и др.) инструментарий, позволяет оперативно конструировать нейросетевую среду, реализовать широкий класс архитектур нейросетей различной сложности и правил модификации весовых коэффициентов в процессе принудительного обучения [3]. Вместе с тем, применение технологии искусственного интеллекта в управлении сложными объектами требует учитывать особенности каждой предметной области, универсальных решений нет и этот факт раскрывает необходимость адаптации набора стандартных опций соответствующего пакета технического анализа к конкретной области исследования.

Выбор аналитического инструментария обусловлен неоспоримыми и важными для практического использования преимуществами:

- нейросети являются адаптивными обучающимися системами, извлекающими на примерах информацию из реальных процессов, которые динамически промоделировать достаточно трудно, а зачастую, просто невозможно, т.к. они часто содержат значительный массив скрытых, неконтролируемых, неполных и зашумленных параметров и взаимных связей между ними;
- их применение позволяет решать задачи, которые трудно или невозможно решить традиционными методами в силу отсутствия формализованных математических описаний процессов функционирования;
- обладают ассоциативной памятью и в процессе работы накапливают и обобщают информацию, от чего эффективность их со временем возрастает;
- их использование базируется на обучении нейросети для извлечения информации из опытных данных, что обеспечивает объективность результатов и повышает их надежность и достоверность.

2.1. Степень разработанности проблемы

Разработкой, теоретическим обоснованием и внедрением информационных технологий для цифровизации экономики занимались отечественные и зарубежные ученые, в частности: С. Хайкин, Р. Хехт-Нильсен, Т. Кохонен, и др. [4, 5]. В результате их деятельности созданы:

- основополагающие принципы и методы прикладной теории



искусственного интеллекта в социальных и технических системах;

- конструктивные подходы к проектированию и моделированию сложных социальных систем;
- прикладная теория управления социальными и производственными системами;
- концепция применения систем поддержки принятия решений для объектов повышенного риска;
- методы и алгоритмы многоуровневой дифференциальной диагностики в различных предметных областях;
- алгоритмические и программные продукты для оперативного многофакторного анализа данных в задачах классификации, прогноза и адаптации входов к состояниям объекта;
- комплексная квалиметрия базовых процессов сложных объектов.

Вместе с тем, полноценной, комплексной и эффективной программной базы автоматизации мониторинга фискально-налоговой нагрузки на сегодня в открытых публикациях не зафиксировано. Такая ситуация требует создания продуктивных технологий и программного обеспечения построения моделей, позволяющих автоматизировать процессы распознавания состояний, прогнозирования динамики поведения и выбор оптимальных управляющих факторов для обеспечения оптимального экономического климата через влияние на налогово-административную систему на базе интеллектуальных технологий[4, 5].

Возникает необходимость в разработке технологии построения продуктивного ансамбля нейросетевых моделей автоматического мониторинга состояния налоговой нагрузки и продуктивного поиска оптимального соотношения входных факторов и целевых состояний. Конечным продуктом должны стать нейросетевые модели распознавания актуальных состояний и нахождения такого набора входных факторов, который надежно и быстро позволил бы привести всю систему в требуемое состояние.

2.2. Процесс нахождения аналитического отображения входного вектора данных

Необходимо найти аналитические отображения входного вектора данных на результат оценки актуального состояния ФНН региона и обеспечить строгое функциональное влияние этого вектора на целевое состояние ФНН.

Задача сводится к решению прямой и обратной задачам распознавания



классов.

В общем случае признаки поведения и число классов ФНН являются векторными величинами:

$$|Y(t)| = F |X(t)|, \quad (1)$$

где $Y(t)$ – номер класса состояния ФНН; $X(t)$ – вектор текущих значений входных факторов-признаков; F – функционал преобразования массива входных признаков в номер класса.

Задача адаптации входного вектора к требуемому состоянию ФНН состоит в нахождении оптимального набора входного вектора факторов $X_0(t)$, соответствующего вектору целевого состояния $Y_0(t)$ ФНН.

Этот процесс может быть формально представлен процедурой:

$$F_0 : Y_0(t) \rightarrow X_0(t), \quad (2)$$

где $Y_0(t)$ – целевой вектор состояния объекта; $X_0(t)$ – вектор адаптированных значений входных признаков; F_0 – продуктивный функционал модификации массива текущих входных факторов к искомому набору, адекватному целевому состоянию.

Реализация алгоритма (2) позволяет найти оптимальный набор входных факторов как последовательное решение прямой и обратной задачи распознавания состояний ССТС.

Решение обратной задачи в нейросетевом формате обеспечивает нахождение таких значений сигналов на входе обученной сети, при которых достигается выходное состояние, соответствующее нужному классу.

В основу классификации (1) предлагается положить использование обучаемой искусственной нейронной сети с обратным распространением ошибки как инструмента автоматического распознавания состояния [6].

В основу поиска входных факторов ФНН, которые соответствуют ее целевому состоянию, предлагается положить технологию адаптации входов к требуемому состоянию на основе принципов нейроуправления [7]. В качестве инструмента моделирования выбран нейроэмулятор среды StatSoft.

2.3. Классификация актуальных состояний ФНН

Построение математической модели внутренней структуры индикаторов состояния ФНН на основе исходных данных входного множества факторов сводится к отображению пространства факторов на пространство состояний с заданной надежностью и точностью:



$$F : X \rightarrow Y_{opt}, X \subset \mathfrak{R}^m, Y_{opt} \subset \mathfrak{R}, \quad (3)$$

где X – вектор признаков состояния ФНН; Y_{opt} – выходное значение класса состояния ФНН.

Массив признаков ФНН $X^n = \{x_1, x_2, \dots, x_n\} \subset X$ совместно с соответствующим массивом классов состояний ФНН позволяет реализовать известное правило распознавания образов [6]:

$$\omega_g \in \Omega_k, \text{ если } L(\omega, \{\omega_g\}) = \sup_i L(\omega, \{\omega_i\}), \\ L(\omega, \{\omega_g\}) \rightarrow \omega_g \in \Omega_k, \quad (4)$$

где $\vec{X} = \{x_1, \dots, x_n\} \in X$; $L(\omega, \{\omega_g\})$ – правило отнесения состояния ФНН ω_g к соответствующему классу; $\{\omega\}$ – множество состояний ФНН (p, g) в пространстве признаков (k, l) при всех их возможных сочетаниях $(\omega_{pk}, \omega_{gl})$.

Алгоритмическая реализация решающего правила (4) осуществляется решением задачи классификации с учителем.

1. На вход нейронной сети, состоящей из одного слоя нейронов, подается вектор x .

2. Номер нейрона, выдавшего минимальный ответ, является номером класса, к которому принадлежит вектор x .

Покажем алгоритм обучения в этой задаче.

1. Поправки всех синапсов полагаем равными нулю.

2. Для каждой точки множества $\{x^p\}$ выполняем следующую процедуру:

Предъявляем точку сети для классификации.

Пусть при классификации получен ответ – класс l . Тогда для обратного функционирования сети подается вектор Δ , координаты которого определяются

по следующему правилу: $\Delta_i = \begin{cases} 0, i \neq l \\ 1, i = l \end{cases}$.

Вычисленные для данной точки поправки добавляются к ранее вычисленным.

3. Для каждого нейрона производим следующую процедуру.

Если поправка, вычисленная последним синапсом равна 0, то нейрон удаляется из сети.

Полагаем параметр обучения равным величине, обратной к поправке, вычисленной последним синапсом.

Вычисляем сумму квадратов накопленных в первых n синапсах поправок и, разделив на 2, заносим в поправку последнего синапса.



Выбираем шаг обучения с параметрами $h_1 = 0$, $h_2 = 2$.

4. Если вновь вычисленные синаптические веса отличаются от полученных на предыдущем шаге, то переходим к первому шагу алгоритма.

У нейрона, выдавшего минимальный сигнал, первые n поправок будут равны координатам распознававшейся точки x , а поправка последнего синапса равна единице. После завершения второго шага алгоритма поправка последнего синапса i -го нейрона будет равна числу точек, отнесенных к i -му классу, а поправки остальных синапсов этого нейрона равны сумме соответствующих координат всех точек i -го класса. Для получения правильных весов остается только разделить все поправки первых n синапсов на поправку последнего синапса, положить последний синапс равным сумме квадратов полученных величин, а остальные синапсы – полученным для них поправкам, умноженным на 2. Это происходит при выполнении третьего шага алгоритма.

Будем считать, что при предъявлении примера первого класса персептрон должен выдать на выходе нулевой сигнал, а при предъявлении примера второго класса – единичный. Ниже приведено описание алгоритма обучения персептрона.

1. Полагаем все веса 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 4.

4. В противном случае обучение завершено.

При этом правило, реализующее приведение выходной ошибки обучения ко входу каждого фактора, имеет строгое математическое описание. Это крайне важно при адаптации входного вектора факторов к целевому состоянию (вектору индикаторов) ССТС, поэтому зафиксируем необходимые аналитические соотношения. Ограничимся архитектурой многослойного персептрона и единичной обучающей выборкой. Так как ССТС, как правило,



располагает 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 градиента функции невязки рассчитываются согласно выражений:

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

Выходы 1 и 2 слоев нейронов определяются из выражений:

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

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

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

$$\frac{\partial E(W)}{\partial w_{il}^{(2)}} = (y_i - y_{iz}) \cdot \frac{\partial f(a_i^{(2)})}{\partial a_i^{(2)}} \cdot g_l,$$

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

Тогда соответствующий компонент 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 [179, 188]:

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



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

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

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

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

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).

2.4. Адаптация входных факторов к целевым состояниям

Если актуальное состояние ФНН не соответствует целевому состоянию, то необходимо найти соответствующий набор входных факторов, т.е. решить обратную задачу распознавания: по заданному классу ФНН найти определить набор признаков (факторов) адекватных этому состоянию. Формально невязка актуального и целевого состояний в нейросетевом формате формализуется в виде [7]:

$$\frac{1}{mn} \sum_{j=1}^n \sum_{i=1}^m (y_{ij} - d_{ij})^2 \Rightarrow \min(\mathfrak{R} \leq \mathfrak{R}_0), \quad (5)$$

где y_{ij} – вектор индикаторов выходных состояний; d_{ij} – результат обучения сети на j - выходе, при i - м примере обучающей выборки; $j = 1, n$ – номер выхода сети; $i = 1, m$ – номер примера; m, n – размерность массива примеров и числа выходных элементов сети; $\mathfrak{R}_0$ – допустимые ограничения по условию задачи.

Решение задачи минимизации целевой функции (5) в постановке задачи выполняется итерационно в процессе обучения сети по алгоритму обратного распространения ошибки [7] применительно к входным факторам исследуемой ФНН:

Модификация синоптического пространства путем подстройки весовых коэффициентов в направлении антиградиента невязки:



$$w(t+1) = w(t) - \mu \left(\frac{\partial E}{\partial w} \right), \quad (5)$$

где μ – коэффициент обучения; E – функция невязки (ошибка).

Если Y представляет актуальное (текущее) состояние ФНН, а Y^* – реакцию сети, то формально процесс адаптации представляется известным алгоритмом [7]:

представляем ошибку сети как: $E^* = \frac{1}{2}(e^*)^2 = \frac{1}{2}(Y - Y^*)^2$, тогда

$$Y^* = f_{out}^e \left\{ \sum_{i=1}^{N_2} w_i^{e_2} f^{e_2} \left[\sum_{j=1}^{N_1} w_{ij}^{e_1} f^{e_1}(y_j^{e_1}) - b_i^{e_2} \right] - b_{out}^e \right\},$$

где: $f_{out}^e(..)$, $f^{e_2}(..)$, $f^{e_1}(..)$ – активационные функции нейронов 2-го та 1-го слоев; $w_i^{e_2}$, $w_{ij}^{e_1}$ – весовые коэффициенты нейронов на участках между нейронами 2-3-го и 1-2-го слоев; b_{out}^e , $b_i^{e_2}$ – величины порогов нейронов выхода и скрытого слоя.

Для нейронов на входе используется линейная сигнальная функция, а для скрытого слоя – сигмоидная. Порог на выходе обычно равен нулю.

Далее оценивается производная функции невязки и выполняются действия по известной методике [6].

Инструментально задача адаптации входов сети к ее целевому состоянию решена в парадигме существующих градиентных методов адаптации входного вектора к заданной функции на выходе [7]. В этом случае метод обратного распространения ошибки последовательно применяется в два этапа:

- для подстройки параметров нейронной сети при модификации её синаптического пространства;
- для подстройки входного вектора через градиент функции ошибки по входным сигналам сети.

Входные факторы ФНН, соответствующие ее требуемому классу, подавались на вход обученной сети. Имея заданный ответ и ответ выданный сетью, вычислялся градиент функции ошибки по входным сигналам сети по известной технологии метода нагруженных двойственных сетей [7].

Задача обучения НС имеет две фазы. Первая – представляет усвоение начальных функций НЭ и НК, которые приближенно соответствуют динамике управляемого состояния ФНН, а вторая – в поддержке адаптивного управления.

Структура взаимодействия факторов, ошибок и управляющих воздействий поясняет схема:

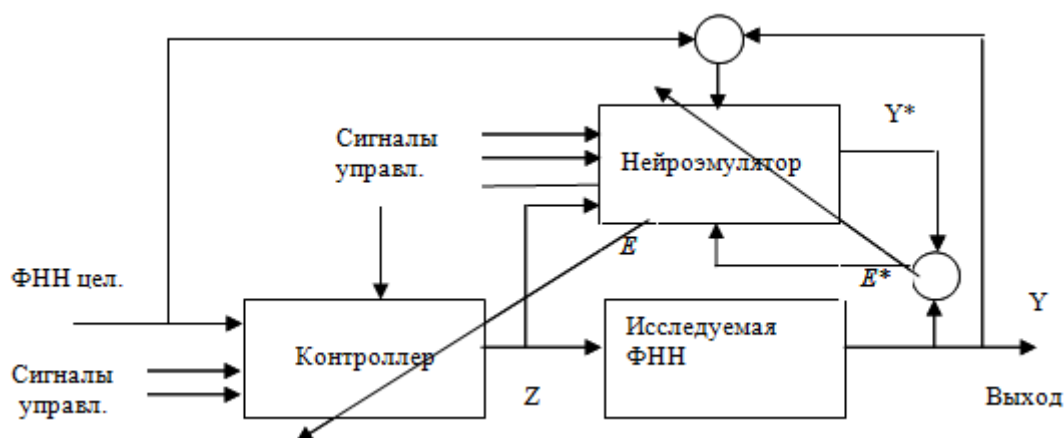


Рис. 1. Структурная схема адаптации входов

После достижения необходимой точности автономной реакции НК проводится дополнительное обучение обоих нейросетей на примерах управления ФНН при различных ее значениях.

Вторая фаза обучения нейросетей системы нейроуправления проходит в рабочем режиме управляемого объекта, что требует соблюдения строгого графика выполнения операций управления и коррекции параметров сетей. Каждая новая реализация пополняет обучающую выборку для оперативного обучения. Чтобы избежать потери полезной информации, что содержится в этой реализации, необходимо в течение одного такта выполнить достаточное количество эпох сетей.

Обобщенный алгоритм для адаптации входов выразим формулой (6):

$$X_{k+1} = X_k + \Delta_k p_k, \quad (6)$$

где вектор p_k задает направление движения, а Δ_k – размер шага на k -ой итерации.

Формулу расчета вектора p_k выразим следующим образом:

$$\vec{p}_k = \vec{g}_k + \sum_{i=1}^k B_k \cdot \vec{g}_{i-1} \quad (6)$$

где вектор p_k задает направление движения; $\vec{g}_k$ – направление антиградиента на k -ой итерации; B_k – коэффициент, определяющий вес i -го градиента; k – порядковый номер текущей итерации.

Таким образом, предлагаемый алгоритм является решением последнего этапа процесса изменения входных факторов при их адаптации к состоянию ФНН заданного класса.

Общий алгоритм этого процесса примет вид:

1. Начало.

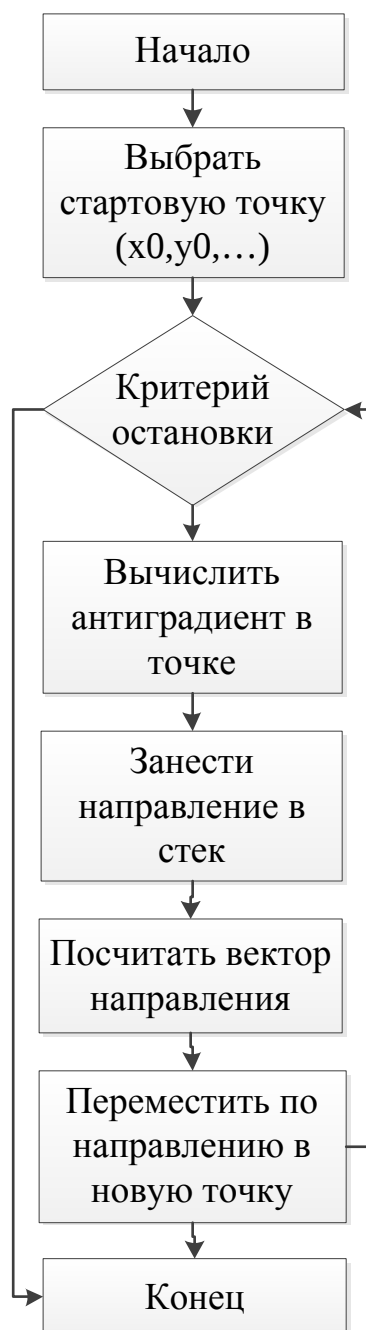


Рис.2 Алгоритм адаптации входов для целевой ФНН

2. Выбор начальной точки с координатами $(x_{10}, x_{20}, \dots, x_{n0})$.
3. Проверка критерия остановки (число итераций, средняя квадратическая ошибка обучения и др.).
4. Вычисление антиградиента в текущей точке $\vec{g}_k$.
5. Запоминание текущего направления в файле направлений.
6. Расчет вектора направления $\vec{p}_k$ по формуле 6.
7. Перемещение в пространстве факторов в новую точку,



соответствующую вычисленному вектору $\vec{p}_k$.

8. Возвращение к шагу 2 и проверка критерия остановки.

9. При положительном результате – окончание алгоритма, а при отрицательном – переход к шагу 3.

10. Остановка. Найденная точка соответствует минимуму функции невязки.

Таким образом, представлены математические, алгоритмические и инструментальные процедуры решения двух относительно самостоятельных процедур:

– построение адекватной модели текущего состояния исследуемого состояния ФНН;

– нахождение вектора входных факторов, соответствующего целевому состоянию в рамках допустимых отклонений.

Результаты моделирования градиентного спуска с оценкой частных производных как по параметрам сети так и по входным сигналам, представляющим пространство признаков текущего состояния ФНН, показало устойчивую сходимость итерационного процесса обучения сети с приемлемыми показателями качества.

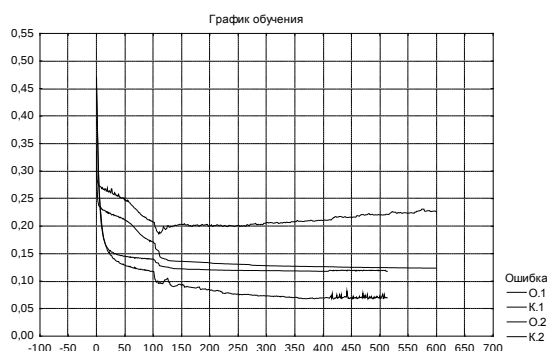


Рис. 3. Результат обучения моделей нейросетей

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

2.5. Интерпретация результатов моделирования

Адекватность нейросетевых моделей установленная по величине продуктивности, ошибкам на обучающих, контрольных и тестовых множествах при обучении на репрезентативной выборке примеров, позволяет утверждать о



состоятельности принимаемых решений по результатам моделирования [1,6,7]. На практике это позволяет подтвердить возможность находить и постоянно удерживать оптимальное значение исследуемой налогово-административной нагрузки на бизнес (величину платежей, отчетов, проверок, штрафов и т.д.) при которой будет достигнут оптимальный баланс между возможностью бизнеса успешно развиваться и достаточным наполнением регионального бюджета. Что является целью исследования.

Выводы

1. Для автоматизации определения величин факторов, приводящих актуальное состояние ФНН к целевому, необходимо найти функциональную зависимость её состояний от значений факторов. Эта задача решена применением технологии нейроруправления и реализована моделями многослойных персептронов как обратная задача распознавания.

2. Расчет частных производных по входным факторам выходной функции невязки текущего и требуемого состояний ФНН и обобщенная оценка градиента выполнены на основе известных свойств нейросетевых схем, функционирующих в связке «эмулятор-контроллер». Это позволило автоматизировать процесс определения элементов вектора градиента при реализации традиционного метода обратного распространения ошибки.

3. Практическая значимость результатов исследования состоит в создании программного инструментария для перехода к автоматическим системам адаптации пространства входных признаков к пространству требуемых классов в системах оптимального управления экономикой.