

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)
Факультет компьютерных технологий и прикладной математики
Кафедра анализа данных и искусственного интеллекта

КУРСОВАЯ РАБОТА

**РЕГРЕССИОННОЕ МОДЕЛИРОВАНИЕ В ПРОГРАММЕ
AUTOMATED NEURAL NETWORK**

Работу выполнил _____ И. А. Бордонос
(подпись)

Направление подготовки 01.04.02 Прикладная математика и информатика

Направленность (профиль) Математическое и информационное обеспечение
экономической деятельности

Научный руководитель
д-р. техн. наук, проф. _____ А.А. Халафян
(подпись)

Нормоконтролер
канд. физ.-мат. наук, доц. _____ Г.В. Калайдина
(подпись)

Краснодар
2021

РЕФЕРАТ

Курсовая работа 39. с., 3 ч., 21 рис., 5 табл., 4 источника.

НЕЙРОННЫЕ СЕТИ, НЕЛИНЕЙНАЯ РЕГРЕССИЯ, МЕТОД BFGS, MLP, AUTOMATED NEURAL NETWORK, STATISTICA

Объект исследования: методы нейросетевой нелинейной регрессии.

Цель курсовой работы – реализация нейросетевой регрессии, представленной в модуле ANN пакета Statistica.

Задачи работы:

- изучить математическую часть методов нейросетевой регрессии и технологию их практической реализации;
- реализовать модели применительно к задаче прогнозирования стоимости автомобиля на основе его технических характеристик;
- выбрать наилучшую модель и создать программное приложение.

Итог работы: построена нейросетевая модель с наилучшими прогностическими свойствами и разработано программное приложение.

СОДЕРЖАНИЕ

Введение.....	4
1 Общие положения регрессионных нейронных сетей.....	5
1.1 Персептрон и нейрон	5
1.2 Структура нейрона и сети	5
1.3 Постановка задачи нейросетевой регрессии	7
1.4 Методы оптимизации.....	9
1.5 Алгоритм BFGS	9
1.6 Функции активации.....	14
2 Работа в программе Automated Neural Network.....	16
2.1 Начало работы в ANN	17
2.2 Автоматизированная Нейронная Сеть.....	21
2.3 Пользовательская нейронная сеть	27
2.4 Метод многократных подвыборок.....	26
2.5 Окно анализа моделей.....	27
2.6 Сохранение и загрузка моделей.....	32
3 Модели и реализация	34
3.1 Отбор моделей.....	34
3.2 Программная реализация	37
Заключение	38
Список использованных источников	39

ВВЕДЕНИЕ

Развитие систем машинного обучения, в частности нейронных сетей делает возможным решение большого количества практических задач. Задачи, решаемые нейронными сетями условно, можно разделить на два класса задачи классификации и задачи нелинейной регрессии. Разница состоит в том, что для задач классификации целевой является категориальная переменная, а для задач регрессии – непрерывная.

Одной из задач регрессии является задача прогнозирования стоимости товаров на основании некоторых его характеристик. Эта задача может быть актуальна с точки зрения покупателя для оценки соответствия практической пользы того или иного товара по сравнению с его ценой. Также он является актуальной с точки зрения продавца при внедрении нового товара на рынок для оценки его цены.

В рамках решения задачи нелинейной регрессии при помощи нейронных сетей возникает вопрос определения архитектуры сетей, их параметров. Очевидно, что далеко не все сочетания параметров будут давать адекватный результат. Даже напротив – плохо созданная модель способна при применении доставить немало проблем.

Таким образом, возникает задача качественного определения параметров нейронных сетей для приемлемого анализа. С этой целью в пакете Statistica был реализован модуль Automated Neural Network (ANN). Он позволяет проводить автоматизированный подбор параметров и обучение моделей нейросетевой классификации и регрессии, а также проводить анализ полученных моделей.

1 Общие положения регрессионных нейронных сетей

1.1 Персептрон и нейрон

Основной структурной единицей нейронной сети является персептрон, или нейрон. Впервые модель искусственного нейрона появилась в работе У. Макколлока и В. Питтса «Логическое исчисление идей, относящихся к нервной активности». Представленный ими нейрон подходил для задач классификации, поскольку могут только выдавать, или не выдавать сигнал. [1]

Модель многослойного персептрона (MLP) была предложена в 1957 году Розенблаттом. Эта модель подразумевает послойное построение сети и деление нейронов на входные S-элементы, выходные R-элементы и A-элементы скрытого слоя. [2]

Программа ANN использует именно MLP. Под S-элементами подразумевается непосредственно входные данные. Они выступают в виде первого слоя сети и непосредственно передаются на элементы скрытого слоя. Выходы этих элементов передаются на следующий слой. Процесс повторяется вплоть до выходного слоя, который образует результат сети. В рамках задачи регрессии в ANN используется один скрытый слой, а также один выходной элемент.

1.2 Структура нейрона и сети

В рамках ANN элементы скрытого и выходного слоя имеют одинаковую структуру. Каждый такой нейрон имеет две составные части – вектор весов w и функцию активации F . Размерность вектора весов соответствует количеству входов нейрона. В процессе обучения функция активации остается неизменной, но варьируется вектор весов. Схематичное изображение такого нейрона представлено на рисунке 1.

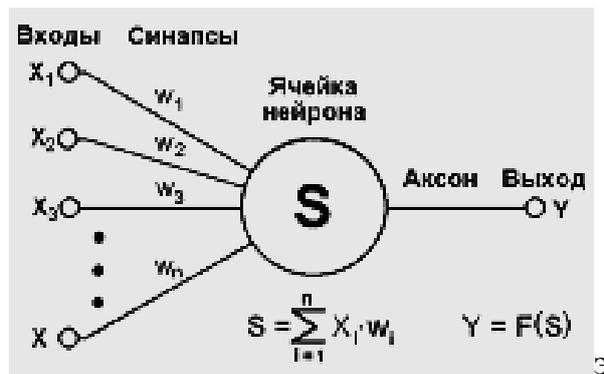


Рисунок 1 – Схематическое изображение нейрона MLP

Функция активации действует на сумму произведений входов и соответствующих им весов. Таким образом, можно утверждать, что функция активации действует на скалярное произведение вектора входов и вектора весов. Обозначим его как $S(x, w)$, где x – вектор входов. Причем, для первого (единственного в ANN) скрытого слоя этим вектором являются непосредственно входные данные сети.

Аналогично выглядит выходной нейрон сети, но с небольшой разницей его выход – является выходом (результатом) сети.

Послойная организация сети подразумевает несколько правил. Во-первых, в рамках одного слоя может быть использована только одна функция активации. Во-вторых, для каждого нейрона из одного слоя набор входов одинаков и представляет собой выходы предыдущего слоя. В-третьих, все выходы одного слоя используются как входы следующего. Схематично это представлено на рисунке 2.

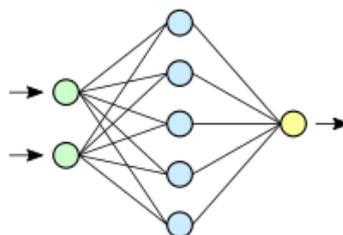


Рисунок 2 – Схематичное представление сети с двумя входами и одним ВЫХОДОМ

1.3 Постановка задачи нейросетевой регрессии

Задача регрессии в рамках рассматриваемой программы подразумевает создание такой модели, которая бы наиболее точно прогнозировала значение целевой переменной. Это подразумевает наличие метрики точности, или её противоположности. В качестве такой метрики в ANN используется сумма квадратов разности, представленная в формуле (1).

$$E = \sum_{i=1}^N (y_i - t_i)^2, \quad (1)$$

где

y_i – предсказанное значение для i наблюдения;
 t_i – значение целевой переменной для i наблюдения;
 N – количество наблюдений в обучающей выборке.

Следует понимать, что t_i является константой, то есть варьируется только y_i .

Рассмотрим способ получения y_i . Положим, что i наблюдение имеет вектор входов x^i . Выходом скрытого слоя будет вектор u^A , зависящий от входов. Координаты этого вектора получаются по формуле (2).

$$u_k^A(x_i) = F_A((w_k, x^i)), \quad (2)$$

где

$u_k^A(x_i)$ – выход k нейрона скрытого слоя;
 w_k – вектор весов k нейрона;
 F_A – функция активации скрытого слоя А.

В силу технических особенностей ANN следует считать, что имеется только один скрытый слой. Таким образом на выходном нейроне по формуле (3) будет получен выход y .

$$y(x_i) = F_R \left((w_R, u^a(x_i)) \right), \quad (3)$$

где

$y(x_i)$ – итоговый выход сети;

w_R – вектор весов выходного нейрона;

F_A – функция активации выходного нейрона.

Обозначим $y(x_i)$ как y_i . Также, обратим внимание, что значение выхода, а также промежуточные значения нейронов зависят от помимо входного вектора еще и от весов на конкретных нейронах. Объединим все веса в один вектор W . Поскольку каждый из выходов нейронов зависит от соответствующего вектора ему вектора весов, то они будут зависеть и от общего вектора весов W . Таким образом получим представление (4), (5).

$$u_k(W, x_i) = F_A \left((W^k, x_i) \right), \quad (4)$$

$$y_i(W) = F_R \left((W^R, u(W, x_i)) \right), \quad (5)$$

где

W^k – подвектор вектора W , состоящий из весов k нейрона скрытого слоя;

W^R – подвектор вектора W , состоящий из весов выходного нейрона.

В итоге получим запись (6).

$$E(W) = \sum_{i=1}^N (y_i(W) - t_i)^2. \quad (6)$$

С учетом вышесказанного мы имеем задачу безусловной многомерной оптимизации (7).

$$E(W) \rightarrow \min \quad (7)$$

1.4 Методы оптимизации

В рамках программы ANN поставленная выше задача решается с помощью одного из трех методов оптимизации:

- Метод градиентного спуска;
- Метод сопряженных градиентов;
- Метод BFGS.

Для большинства задач подходит последний метод, и именно он стоит по умолчанию при работе в программе. Рассмотрим его.

1.5 Алгоритм BFGS

Алгоритм BFGS (Broyden, Fletcher, Goldfarb, Shanno) относится к семейству Квази-Ньютоновских алгоритмов многомерной оптимизации.

Для начала определим разложение целевой функции в полином второй степени (8)

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p. \quad (8)$$

B – это симметричная положительно-определенная матрица размера $n * n$, где n – количество переменных функции f . Матрица обновляется на каждой итерации. Следует отметить, что при $p = 0$ значения функции и ее градиента равны соответственно f_k и ∇f_k . p_k представим в виде (9)

$$p_k = -B_k^{-1} \nabla f_k. \quad (9)$$

Он используется для нахождения направления новой итерации (10)

$$x_{k+1} = x_k + \alpha_k p_k. \quad (10)$$

Длина шага α_k подбирается таким образом, чтобы соответствовать условиям Вольфе (11), (12).

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k, \quad (11)$$

$$\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k, \quad (12)$$

где

c_1, c_2 – константы, соответствующие условию $0 < c_1 < c_2 < 1$.

Во время итераций на B_{k+1} накладываются некоторые ограничения. Градиент m_{k+1} должен совпадать с градиентом целевой функции f на, как минимум, двух итерациях x_k и x_{k+1} . Поскольку $\nabla m_{k+1}(0) = \nabla f_{k+1}$ – второе условие выполняется автоматически. Первое условие может быть записано в виде (13)

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k. \quad (13)$$

Преобразуем (13) в (14)

$$B_{k+1} \alpha_k p_k = \nabla f_{k+1} - \nabla f_k. \quad (14)$$

Для простоты введем векторы (15), (16)

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \quad (15)$$

$$y_k = \nabla f_{k+1} - \nabla f_k. \quad (16)$$

Тогда (14) преобразуется в (17)

$$B_{k+1} s_k = y_k. \quad (17)$$

Полученное равенство (17) назовем уравнением секущей.

При данных s_k и y_k и симметричной положительно-определенной матрице B_{k+1} для выполнения равенства (17) требуется выполнение условия (18)

$$s_k^T y_k > 0 \quad (18)$$

Если f достаточно выпукла, неравенство (18) выполнится для любых двух точек x_k, x_{k+1} . Однако, возможны случаи, когда f недостаточно выпукла. Поэтому требуются дополнительные ограничения на длину шага α . Ограничения Вольфе (11), (12) служат именно этой цели. В действительности подставим в неравенство (12) выражения (15) и (16). Получим (19)

$$y_k^T s_k \geq (c_2 - 1) \alpha_k \nabla f_k^T p_k. \quad (19)$$

Поскольку $c_2 < 1$ и p_k – направление спуска, то выражение в правой части – положительное, следовательно условие (18) выполняется.

При выполнении условия (18) всегда существует B_{k+1} – решение уравнения секущей (17). На самом деле таких решений бесконечно много, поскольку уравнение имеет $n(n+1)/2$ степеней свободы, при n ограничениях, налагаемых уравнением секущей. Условие положительной определенности также накладывает n дополнительных неравенств (все

миноры должны быть положительными). Однако этих ограничений все-же не хватает для поглощения степеней свободы.

Чтобы определить единственную B_{k+1} , введем дополнительное условие – среди всех решений уравнения секущей искомая матрица – та, которая в наименьшей степени отличается от матрицы B_k , полученной на прошлой итерации. Иными словами, решается задача (20), (21)

$$\min_B \|B - B_k\|, \quad (20)$$

$$B = B^T, B s_k = y_k, \quad (21)$$

s_k и y_k удовлетворяют условию (18) и B_k – симметричная, положительно-определенная матрица. Различные нормы могут быть использованы. В рамках данного метода используется норма Фробениуса (22), (23)

$$\|A\|_W \equiv \left\| W^{\frac{1}{2}} A W^{\frac{1}{2}} \right\|_F, \quad (22)$$

$$\|C\|_F = \sum_{i=1}^N \sum_{j=1}^N c_{i,j}^2. \quad (23)$$

Матрица весов W выбирается как любая из матриц, удовлетворяющих условию (17). Не теряя общности, предположим, что $W = \bar{G}_k^{-1}$, где \bar{G}_k – средний Гессиан, выражаемый как (24)

$$\bar{G}_k = \left[\int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau \right]. \quad (24)$$

С учетом этого, единственное решение задачи (20), (21) представимо в виде (25), (26)

$$B_{k+1} = (E - \rho_k y_k s_k^T) B_k (E - \rho_k s_k y_k^T) + \rho_k s_k y_k^T \quad (25)$$

$$\rho_k = \frac{1}{y_k^T s_k}. \quad (26)$$

Формулу (25) также называют формулой DFP (Davidon, Fletcher, Powell).

Обозначим матрицу, обратную B_k как (27)

$$H_k = B_k^{-1}. \quad (27)$$

Она используется в реализации данного метода, поскольку позволяет найти направление поиска (9) с помощью простых операций перемножения матриц. Заменим B_k на нее в равенстве (25)

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \quad (28)$$

В отличие, от DFP, представленного выше BFGS налагает условия не на непосредственные аппроксимации Гессиана B_k , а на обратные им H_k . Заменим B_{k+1} в формуле (17). Получим уравнение текущей (29)

$$H_{k+1} y_k = s_k. \quad (29)$$

Аналогично получается задача (30), (31)

$$\min_H \|H - H_k\|, \quad (30)$$

$$H = H^T, H y_k = s_k. \quad (31)$$

Из этой таким же способом находится единственное решение (32), (33)

$$H_{k+1} = (E - \rho_k y_k s_k^T) H_k (E - \rho_k s_k y_k^T) + \rho_k s_k y_k^T, \quad (32)$$

$$\rho_k = \frac{1}{y_k^T s_k}. \quad (33)$$

При этом H_0 , задается как единичная матрица.[3]

Итерации проходят до тех пор, пока не будет достигнут локальный минимум, определяемый, как отсутствие улучшения результата, либо до ограничения итераций.[4]

1.6 Функции активации

В рамках программы ANN предусмотрено пять функций активации нейрона:

– 1) Тожественная

$$y = x \quad (34)$$

– 2) Гиперболическая

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (35)$$

– 3) Логистическая

$$y = \frac{1}{1 + e^{-x}} \quad (36)$$

– 4) Экспоненциальная

$$y = e^{-x} \quad (37)$$

– 5) Синусоидальная

$$y = \sin (x) \quad (38)$$

Для задач регрессии на скрытых нейронах рекомендуется использовать гиперболическую функцию, на выходных – тождественную.[4]

2 Работа в программе Automated Neural Network

Рассмотрим работу в ANN применительно к задаче регрессии стоимости автомобиля. В качестве множества для анализа используются данные вторичного рынка автомобилей за 2018 год. Представлено 74 образца. На рисунке 3 представлена структура множества данных.

	1 Объем двигателя	2 Мощность двигателя	3 Тип привода	4 Емкость бака	5 Максимальная скорость	6 Время разгона до 100 км/ч	7 Средний расход топлива	8 Количество ступеней	9 Тип коробки передач	10 Цена
Audi A3 Attraction 1.6 5G	1595	102	передний	55	185	11,9	7,65	5	механическая	860384
Audi A3 Ambition 1.6 Aut	1595	102	передний	55	185	11,9	7,65	6	полуавтоматическая	840672
Audi A3 Ambition 2.0 FSI 6G	1984	200	передний	55	211	9,1	7,7	6	механическая	1055312
Audi A3 Ambition 3.2 DSG Q	3189	250	полный	55	211	9,1	7,7	6	автоматическая	1961056
Audi A4 1.8 TFSI	1781	163	полный	63	225	8,6	7,9	6	механическая	1164500
Audi A4 Avant 3.0	2967	233	полный	70	241	7	8,35	6	механическая	849908
Audi A3 Sportback Ambition 2.0 TFSI DSG	1984	200	передний	55	236	7	8,3	6	автоматическая	1219577
Audi S4 4.2 6G Q	4163	344	полный	63	250	5,6	14,65	6	механическая	1085200
Audi S4 Avant 4.2 TC Q	4163	344	полный	63	250	5,9	13,5	6	полуавтоматическая	3582272
Audi RS4 4.2 6G Q	4163	344	полный	63	250	4,8	14,95	6	механическая	2648339
Audi A5 3.2 FSI MC	3197	265	передний	65	250	6,6	9,55	5	автоматическая	1700000
Audi S5 4.2 6G Q	4163	354	полный	63	250	5,1	13,5	6	механическая	2307000
Audi A6 Limousine 2.4 6G	2393	177	передний	70	231	8,9	10,7	6	механическая	535000
Audi A6 allroad 3.2 FSI TC quattro	3123	260	полный	80	240	7,7	12	6	полуавтоматическая	1450000
Audi A8 4.2 FSI quattro	4172	334	полный	90	250	6,3	13,1	6	автоматическая	3056256
Audi TT Coupe 2.0 TFSI 6G	1984	200	передний	55	240	7,8	8,3	6	механическая	2070796
Audi Q5 2.0 TFSI quattro	1984	200	полный	70	222	7,2	8,85	7	механическая	1095444
Audi Q7 3.0 TDI quattro	2967	233	полный	100	210	9,1	11,45	6	полуавтоматическая	2637500
Audi R8 4.2 FSI 6 G Q	4163	309	полный	75	301	4,6	16,15	6	механическая	6688000
Ford Fiesta	1242	96	передний	42	168	13,3	6,05	4	механическая	389000
Ford Mondeo	2000	130	передний	58	210	9,9	8,6	6	механическая	615000
Ford Galaxy	1753	145	передний	75	171	14,3	6,65	4	механическая	832000
Ford Explorer	4600	296	полный	85	160	10,2	15,95	8	автоматическая	1269000
Ford Tourneo Connect	1800	115	передний	60	162	13,5	10,15	6	механическая	515000
Ford Transit bus	2200	102	передний	103	150	13,9	7,95	4	автоматическая	993000
Ford Transit shassi	2300	360	задний	130	172	13,7	9,85	5	автоматическая	920000
Ford Transit van	2200	140	передний	80	145	10,2	10,1	5	механическая	721000
Ford Ranger	2500	143	полный	63	182	12,5	9,35	4	автоматическая	640000
Ford C-MAX	1600	160	передний	55	175	12,9	7,3	4	механическая	510000
Ford Fusion	1388	100	передний	45	163	14	6,9	4	механическая	370000
Ford Focus	1388	145	передний	62	164	14,1	7,05	4	механическая	389000
Ford S-MAX	1753	220	передний	70	187	11,6	6,55	4	механическая	807000
Ford Escape	2300	155	полный	63	160	11,9	15,95	4	автоматическая	845500
Ford Transit Connect	1800	90	передний	60	130	6,8	12,3	5	механическая	559000
Ford Transit Kombi	2200	145	передний	103	142	9,9	13,6	4	механическая	874000

Рисунок 3 – Структура множества данных

Ниже представлены входные выходные данные строимых моделей.

Входные непрерывные предикторы:

- Объем двигателя (мл);
- Мощность двигателя (л/с);
- Объём бака (л);
- Максимальная скорость (км/ч);
- Время разгона до 100 км/ч (с);
- Средний расход топлива (л/100 км).

Входные категориальные предикторы:

- Тип привода;

- Количество ступеней КПП;
- Тип КПП.

Выходная непрерывная переменная:

- Стоимость (руб).

Во всех моделях множество разбивается на обучающую, контрольную и тестовую выборки в соотношениях 70%, 15%, 15% соответственно.

2.1 Начало работы в ANN

Для работы с инструментарием ANN следует выбрать пункт Автоматизированные Нейронные Сети (Automated Neural Networks). Откроется окно, представленное на рисунке 4.

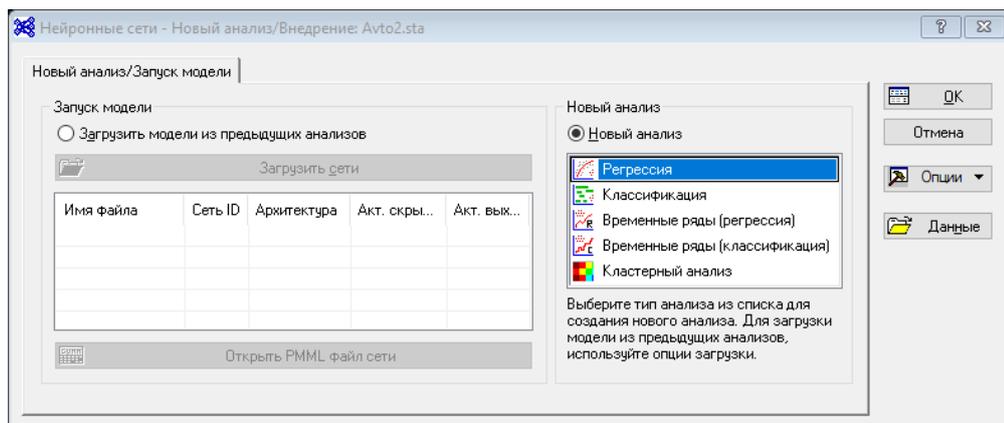


Рисунок 4 – Окно запуска моделей

В окне, называемым окном запуска модели, следует выбрать один из двух вариантов – загрузку модели, или создание модели. Загрузка модели будет описана позже, а сейчас рассмотрим процесс создания новой модели. Сперва следует выбрать тип модели (на самом деле – группы моделей). Возможны пять типов моделей:

- регрессия;
- классификация;
- временные ряды (регрессия);

- временные ряды (классификация);
- кластерный анализ.

В рамках данной работы рассматривается вариант регрессии. Выбрав этот вариант и нажав ОК, перейдём на окно задания модели, представленном на рисунке 5.

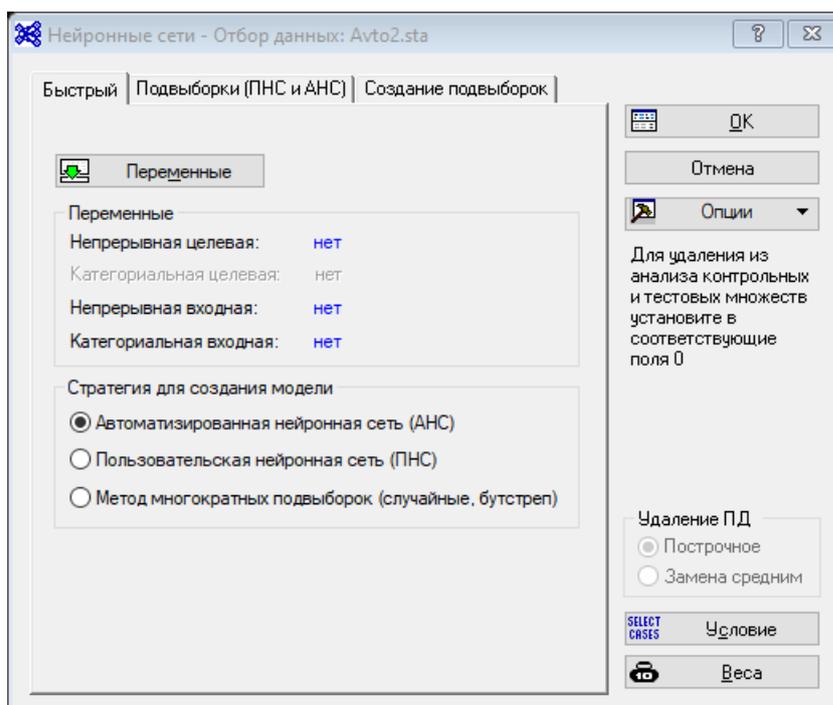


Рисунок 5 – Вкладка «Быстрый» окна задания модели

В первую очередь, для любой нейронной сети следует задать переменные для моделирования. Нажмем кнопку «Переменные», перейдём в окно, представленном на рисунке 6.

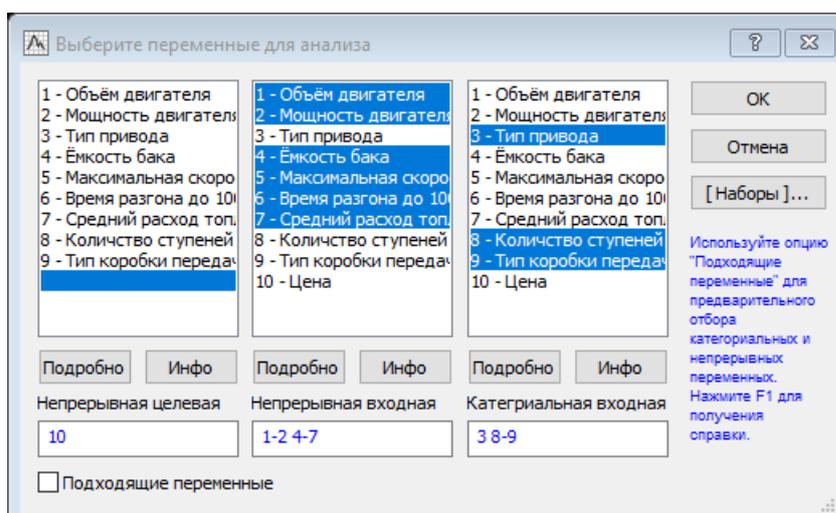


Рисунок 6 – Окно выбора переменных

Выберем соответствующие переменные и нажмем ОК. Вернемся на окно из рисунка 5. Далее следует выбрать режим работы программы. Возможны три режима:

- Автоматизированная Нейронная Сеть (АНС);
- Пользовательская Нейронная Сеть (ПНС);
- Метод многократных подвыборок (случайные, бутстреп).

АНС подразумевает автоматизированный подбор параметров в пределах, задаваемых пользователем. ПНС – позволяет задать сеть с пользовательскими параметрами. Метод многократных подвыборок аналогичен АНС, однако позволяют использовать различные обучающие выборки для разных моделей.

В зависимости от выбранного режима настройки выборки (-ок) определяются на соответствующих вкладках. Для режимов АНС и ПНС – это вкладка «Подвыборки (ПНС и АНС)», представленная на рисунке 7.

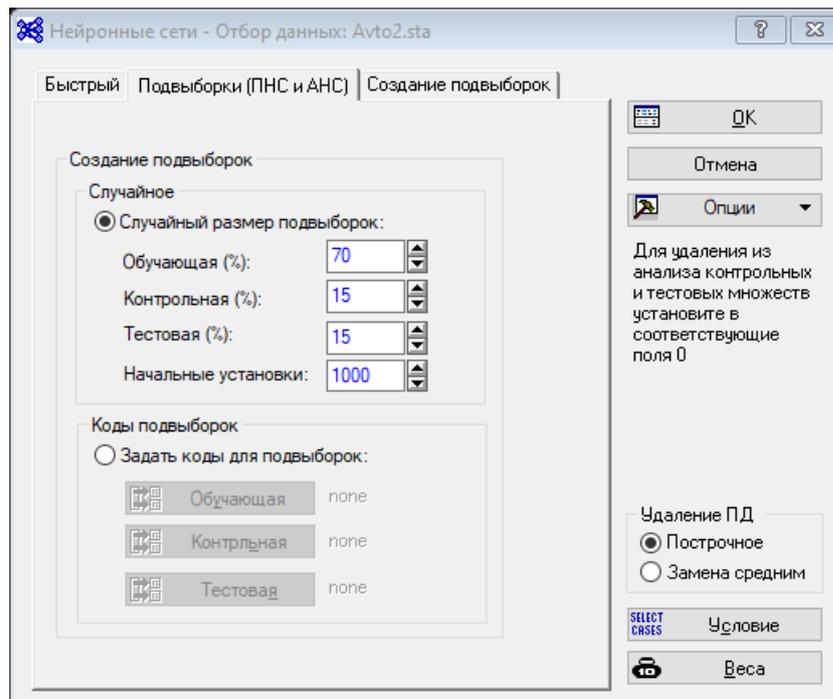


Рисунок 7 – Вкладка «Подвыборки (ПНС и АНС)»

Для данных режимов моделей возможны варианты случайного размера выборок и использования кодов выборок, выбираемых соответствующим флажком. В первом случае необходимо указать – в каком количестве (в процентах от общей выборке) будут заданы обучающая, контрольная и тестовая подвыборки, а также начальное значение генератора случайных чисел. Во втором случае – для каждой из подвыборок требуется указать переменную и ее значение, для определения наблюдения в одну из подвыборок.

Для режима многократных подвыборок эти параметры задаются на вкладке «Создание подвыборок», представленной на рисунке 8.

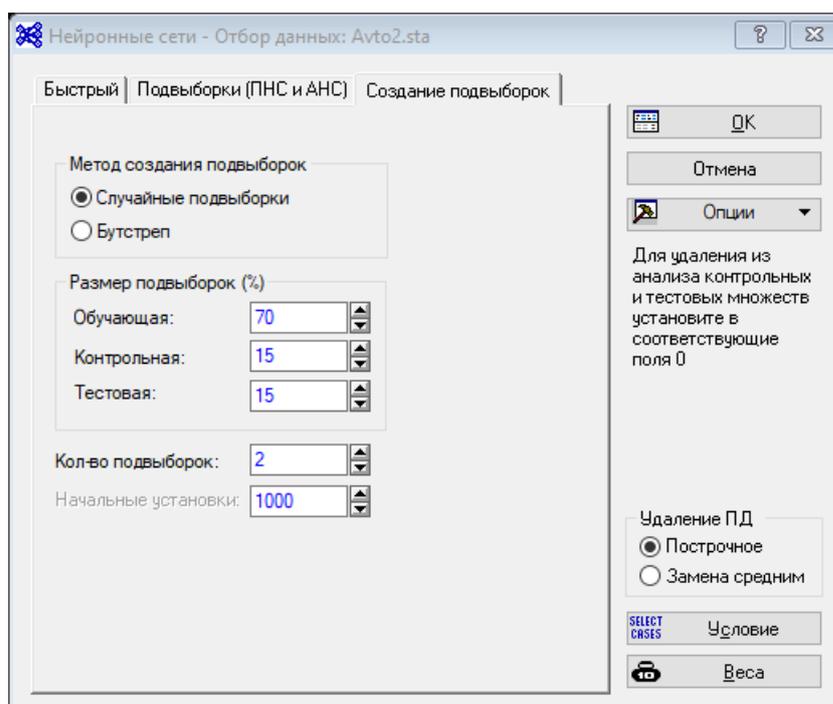


Рисунок 8 – Вкладка «Создание подвыборок»

При использовании многократных подвыборок возможны два способа их формирования – случайный и бутстреп. Разница между ними – в том, что для бутстрепа наблюдения для каждой из подвыборок выбирается вне зависимости от того было ли оно выбрано до того, или нет. На этой вкладке аналогично предыдущему случаю задаются размеры выборок, значение ГСЧ, и, в отличие от него – количество подвыборок.

2.2 Автоматизированная Нейронная Сеть

Для работы с АНС следует выбрать соответствующий режим, после чего задать распределение выборок. Оставим значения по умолчанию и нажмем ОК, перейдем в окно моделей, представленном на рисунке 9.

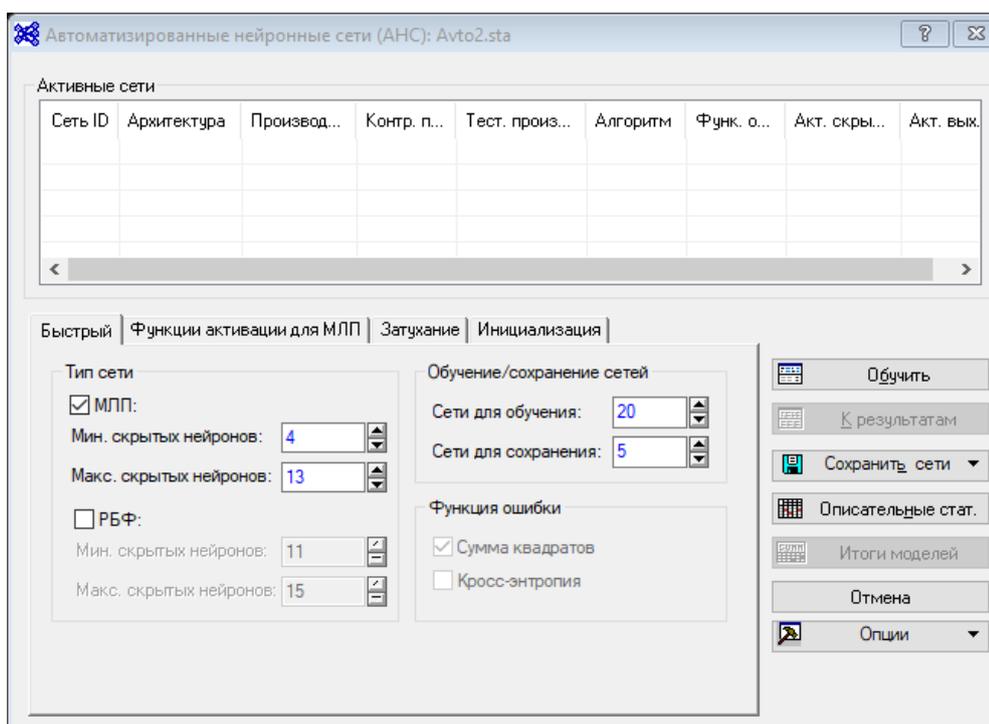


Рисунок 9 – Окно задания параметров моделей

В верхней части окна располагается панель активных сетей. Об ее функционале будет сказано позже. В нижней части – представлены вкладки для задания параметров сетей. Для режима АНС они являются ограничениями для генерируемых сетей.

На вкладке «Быстрый» расположены основные параметры архитектуры сети, а также – количество генерируемых сетей. На панели «Тип сети» задаются возможные типы генерируемых сетей, возможно использовать многослойные перцептроны (МЛП) и/или сети радиально-базисной функции (РБФ). Для каждого из типов задается минимальное и максимальное количество нейронов в скрытом слое. На панели «Обучение/сохранение» задается общее количество перебираемых сетей (окошко «Сети для обучения») и сколько наилучших из них будет сохранено и использовано для анализа. Панель «Функция ошибки» недоступна для задач регрессии, так как для них в качестве такой функции ошибки используется только сумма квадратов.

Для решаемой задачи выставим тип сети – МЛП, минимальной число нейронов в скрытом слое – 5, максимальное – 25. Число генерируемых

моделей – 100, число сохраняемых – без изменений. Перейдем на вкладку «Функции активации для МЛП», представленной на рисунке 10.

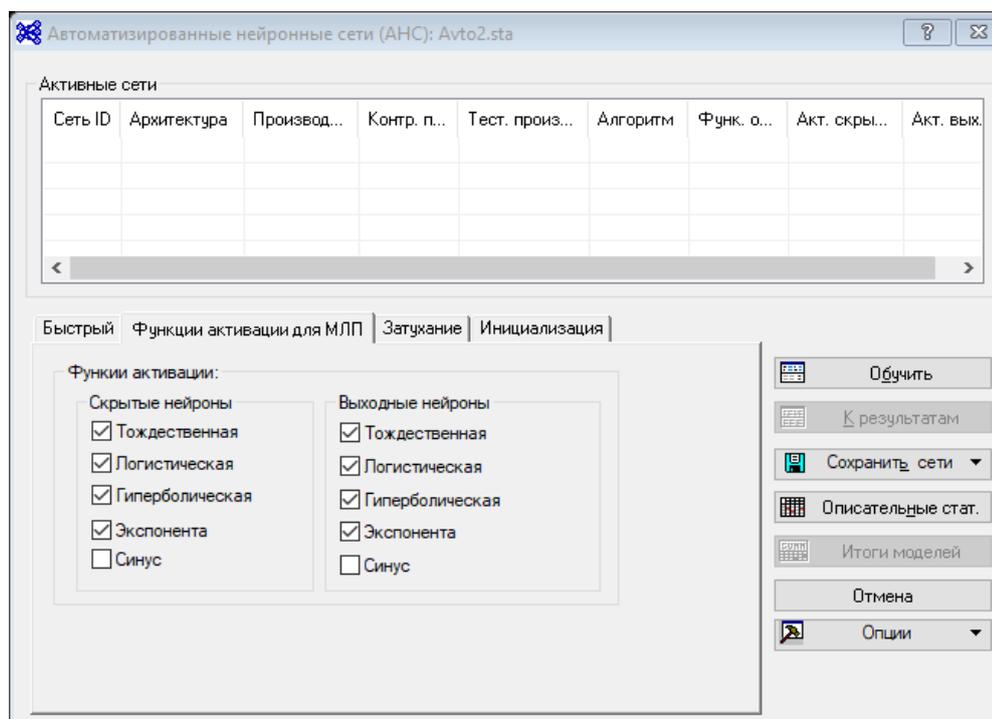


Рисунок 10 – Вкладка «Функции активации для МЛП»

Как следует из названия вкладки – параметры в ней влияют на генерируемые сети типа МЛП. На панели «Скрытые нейроны» задаются функции активации для нейронов скрытого слоя, соответственно на панели «Выходные нейроны» – функции выходных нейронов.

Оставим параметры без изменений и перейдем на вкладку «Затухание», представленную на рисунке 11.

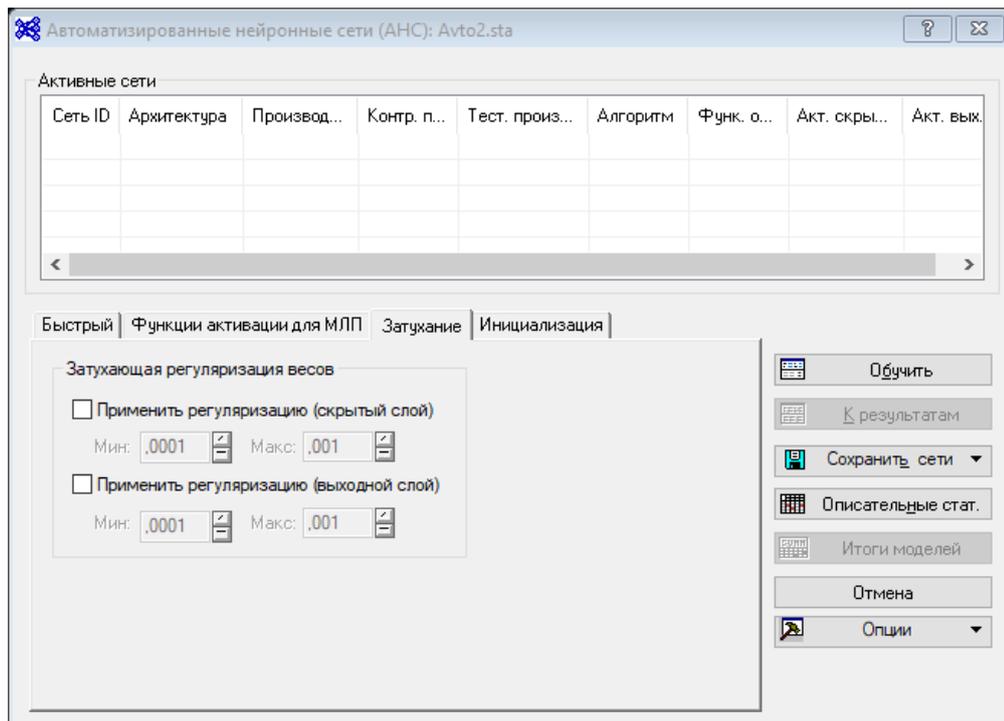


Рисунок 11 – Вкладка «Затухание»

Здесь задается затухающая регуляризация весов модели. Регуляризация скрытого слоя применима только для МЛП, выходного – для обоих типов. Для добавления регуляризации следует выставить соответствующий флажок и указать границы варьирования в соответствующих окошечках.

Выставим регуляризацию для скрытого слоя, оставив параметры без изменений. Перейдем на вкладку «Инициализация», представленную на рисунке 12.

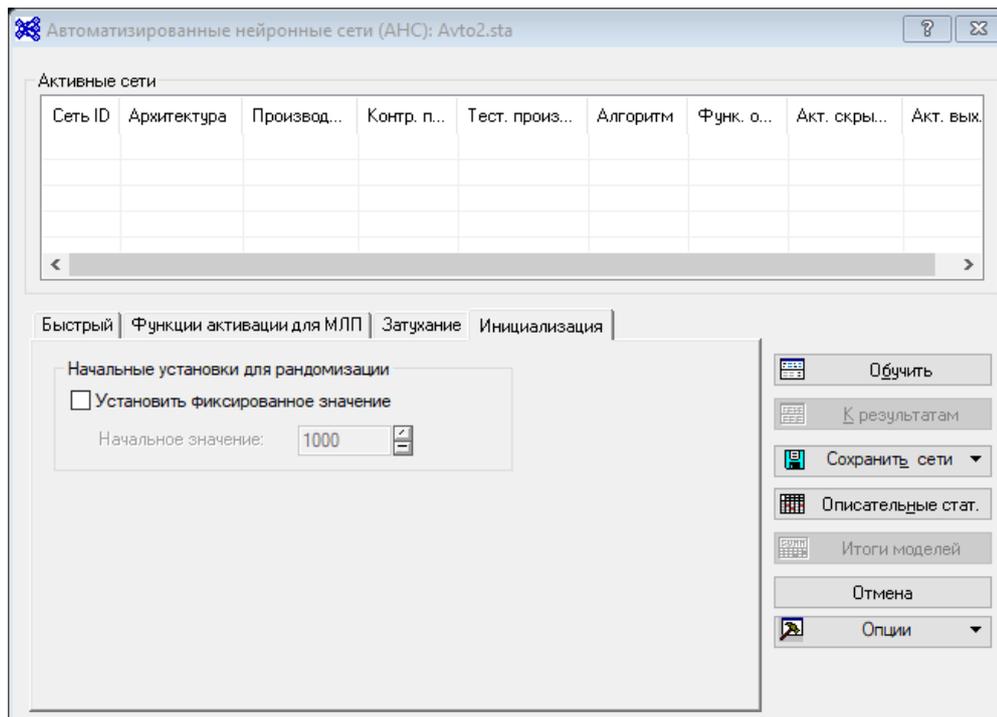


Рисунок 12 – Вкладка «Инициализация»

На этой вкладке задаются начальные значение ГСЧ для строящихся моделей. При выставлении флажка «Установить фиксированное значение» начальное значение ГСЧ будет использоваться для всех генерируемых моделей. В противном случае, для каждой генерируемой модели будет генерироваться свое значение.

Оставим его в таком положении и нажмем кнопку «Обучить». Начнется процесс генерирования и обучения сетей. Информация о процессе отображается в окне, представленном на рисунке 13.

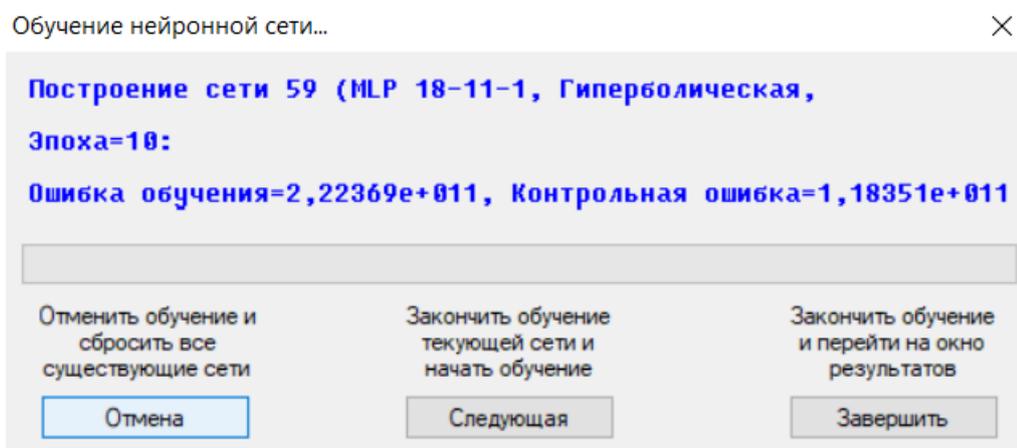


Рисунок 13 – Окно процесса обучения

В этом окне отображаются номер модели, ее архитектура, текущая эпоха обучения, ошибки на обучающих и контрольных данных. Три кнопки снизу позволяют управлять процессом. Кнопка «Отмена» останавливает процесс и сбрасывает все модели. Кнопка «Следующая» позволяет досрочно остановить обучение текущей сети и перейти к следующей. Кнопка «Завершить» – останавливает процесс, как и «Отмена», но в отличие от нее сохраняет модели.

По завершении обучения откроется окно анализа моделей. Подробнее об его функционале будет сказано позже.

2.3 Метод многократных подвыборок

Для запуска этого метода в окне на рисунке 5 следует выбрать соответствующий пункт, после чего выставить параметры создаваемых выборок. Процесс работы в этом режиме почти идентичен таковому в ПНС. Отличием является то, что количество создаваемых систем здесь равно количеству подвыборок, соответственно одна модель – одна подвыборка.

2.4 Пользовательская нейронная сеть

Для задания пользовательской нейронной сети в окне на рисунке 5 следует выбрать соответствующий флажок. Процесс задания обучающей, контрольной и тестовой выборок полностью аналогичен АНС. После нажатия ОК перейдем в окно задания модели, представленном на рисунке 14.

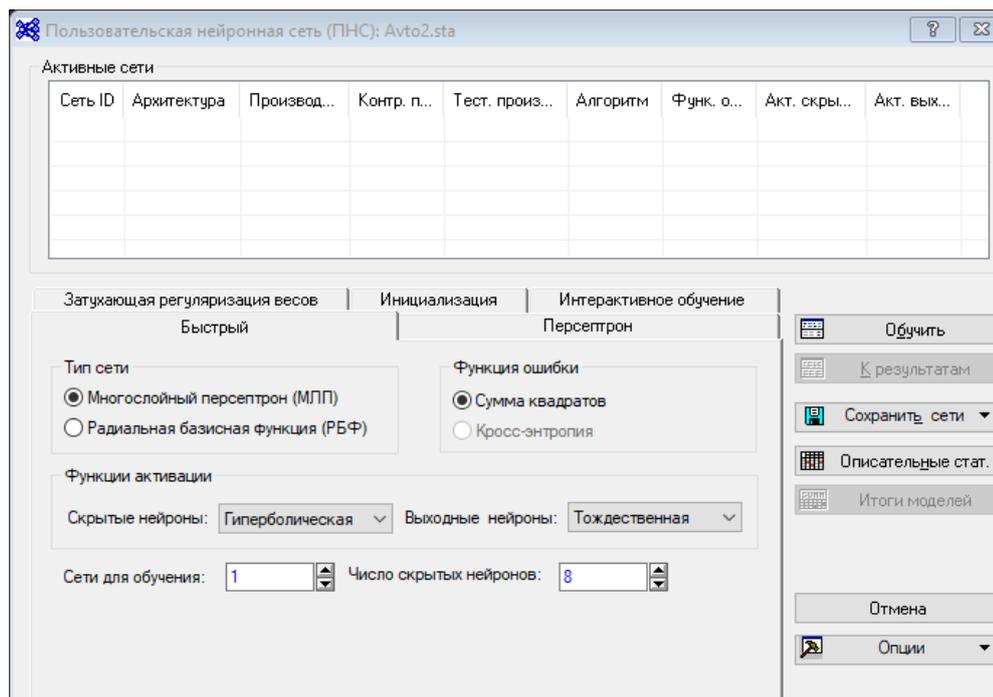


Рисунок 14 – Окно задания модели ПНС, вкладка «Быстрый»

На вкладке «Быстрый» задаются основные параметры сети, а именно – тип сети (МЛП или РБФ), функция ошибки (для классификационных моделей), функции активации нейронов скрытого выходного слоя. Количество сетей для обучения, а также число нейронов в скрытом слое. Важным отличием от АНС является то, что все сети будут иметь одинаковые параметры.

Выставим в качестве функции активации скрытых нейронов логистическую функцию, а количество скрытых нейронов – 7. Перейдем на вкладку «Персептрон», представленную на рисунке 15.

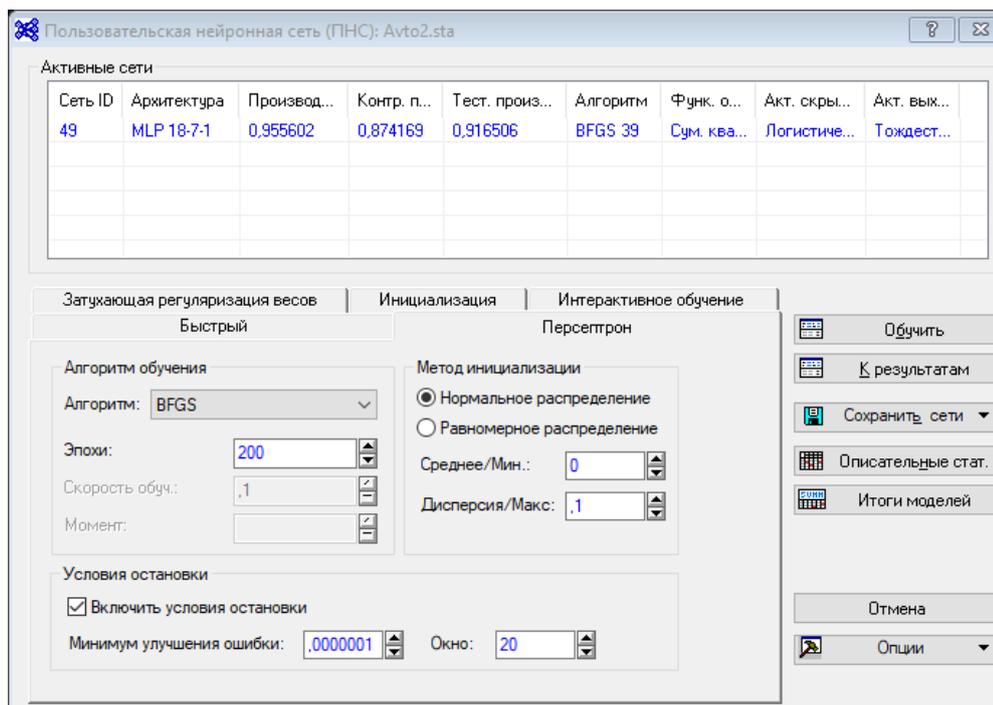


Рисунок 15 – Вкладка «Персептрон»

На этой вкладке выбирается алгоритм оптимизации на панели «Алгоритм обучения», там-же задается максимальное число эпох, методы инициализации весов нейронов и применение досрочной остановки.

Вкладки «Инициализация» и «Затухающая регуляризация весов» соответственно полностью аналогичны вкладкам «Инициализация» и «Затухание» для АНС. На вкладке «Интерактивное обучение» можно включить отображение графиков ошибок, причем можно выбрать как ошибку на обучающей выборке, так и на контрольной. Включать отображение имеет смысл, если процесс обучения занимает относительно большое время. В решаемой задаче такого смысла нет. Нажмем кнопку «Обучить».

По завершении обучения откроется окно анализа моделей, о котором будет рассказано позже.

2.5 Окно анализа моделей

После обучения моделей вне зависимости от режима, открывается окно анализа моделей. Окно представлено на рисунке 16.

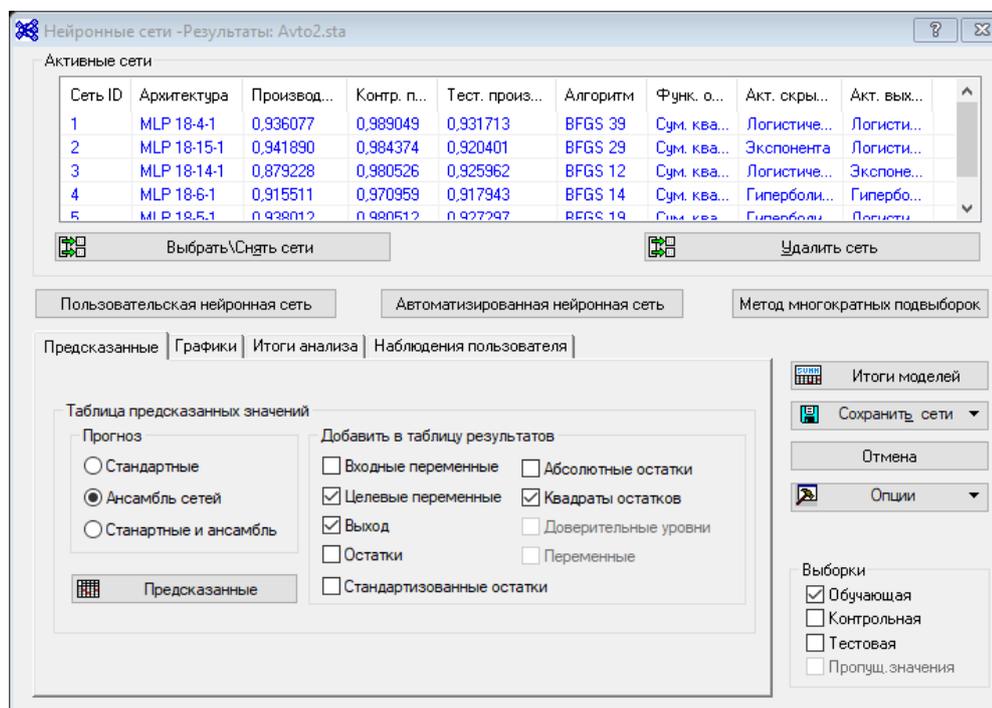


Рисунок 16 – Окно анализа модели

Как и в окне создания модели, в верхней части окна расположены активные модели. Оттуда также можно вернуться в соответствующее окно создания модели, нажав одну из трех кнопок – «Пользовательская нейронная сеть», «Автоматизированная сеть», или «Метод многократных подвыборок». В нижней части окна расположены компоненты одной из четырех вкладок. А также применяемые в анализе выборки.

По умолчанию, открывается вкладка «Предсказанные». Возможны три варианта построения таблицы предсказаний – на основе сетей по отдельности, на основе ансамбля сетей и комбинированный. В таблицу могут быть внесены дополнительные данные. На рисунке 16 представлена таблица предсказанных значений, построенная по параметрам, представленным на рисунке 17.

Наблюд. номер #	Таблица предсказанных значений для Цена (Auto2.sta)		
	Цена Целевая	Цена - Выход Ансамбль сетей	Цена - Квадраты Ост. Ансамбль сетей
Audi A3 Attraction 1.6 5G	860384	645254	4,628076E+10
Audi A3 Ambition 1.6 Aut	940672	962889	4,935858E+08
Audi A3 Ambition 2.0 FSI 6G	1065312	878868	3,476129E+10
Audi A3 Ambition 3.2 DSG Q	1961056	1754770	4,255410E+10
Audi A4 1.8 TFSI	1164500	1163484	1,032869E+06
Audi A4 Avant 3.0	849908	1533159	4,668318E+11
Audi S4 4.2 6G Q	1085200	2368688	1,647342E+12
Audi S4 Avant 4.2 TC Q	3582272	3238368	1,182703E+11
Audi RS4 4.2 6G Q	2648839	2681634	1,075521E+09
Audi A5 3.2 FSI MC	1700000	1377987	1,036926E+11
Audi S5 4.2 6G Q	2307000	2295840	1,245400E+08
Audi A6 Limousine 2.4 6G	535000	1038266	2,532763E+11
Audi A6 allroad 3.2 FSI TC quattro	1450000	2402221	9,067245E+11
Audi A8 4.2 FSI quattro	3056256	3399490	1,178098E+11
Audi TT Coupe 2.0T FSI 6G	2070796	1012244	1,120532E+12
Audi Q5 2.0 TFSI quattro	1095444	1035490	3,594423E+09
Audi R8 4.2 FSI 6 G Q	6688000	5953372	5,396778E+11
Ford Mondeo	615000	892376	7,693769E+10
Ford Transit bus	993000	807240	3,450678E+10
Ford Transit shassi	920000	1013347	8,713584E+09
Ford Transit van	721000	645481	5,703073E+09
Ford C-MAX	510000	620722	1,225945E+10
Ford Fusion	370000	579553	4,391248E+10
Ford S-MAX	807000	667979	1,932687E+10
Ford Transit Connec	559000	621743	3,936636E+09
Hyundai Getz II GL 1.1 5 MT	322000	551629	5,272935E+10
Hyundai Coupe GLS 2.0 5 MT	849473	726251	1,518362E+10
Hyundai Genesis Grand	1744000	1934620	3,633608E+10
Hyundai ix55 Base	1592500	1534152	3,404473E+09
Hyundai Matrix GL 1.6 5 MT	498407	626333	1,636496E+10
Hyundai i30 1.4 Classic MT	530900	664301	1,779595E+10

Рисунок 17 – Таблица предсказанных результатов

На вкладке «Графики» можно построить графики зависимостей входных и выходных данных, а также некоторых показателей данных (таких как среднее значение по выборке).

Из вкладки «Итоги анализа» можно получить различные данные по моделям. Кнопка «Итоги моделей», как в этой вкладке, так и в правой части окна дублирует данные из верхней части окна – панели «Активные модели». Кнопка «Весы» открывает таблицу, представленную на рисунке 18. В ней перечисляются входы нейронов и соответствующие им веса.

Вес ID	Весы (Auto2.sta)				
	Соединения 1.MLP 18-4-1	Значения весов 1.MLP 18-4-1	Соединения 2.MLP 18-15-1		
1	Объем двигателя -> скрытый нейрон 1	-1,64825	Объем двигателя -> скрытый нейрон 1	0,03406	Объем двигателя -> скрытый нейрон 1
2	Мощность двигателя -> скрытый нейрон 1	1,10377	Мощность двигателя -> скрытый нейрон 1	-0,03882	Мощность двигателя -> скрытый нейрон 1
3	Емкость бака -> скрытый нейрон 1	-1,67409	Емкость бака -> скрытый нейрон 1	0,20210	Емкость бака -> скрытый нейрон 1
4	Максимальная скорость -> скрытый нейрон 1	-4,25996	Максимальная скорость -> скрытый нейрон 1	0,24727	Максимальная скорость -> скрытый нейрон 1
5	Время разгона до 100 км/ч -> скрытый нейрон 1	0,20285	Время разгона до 100 км/ч -> скрытый нейрон 1	-0,24822	Время разгона до 100 км/ч -> скрытый нейрон 1
6	Средний расход топлива -> скрытый нейрон 1	-1,73852	Средний расход топлива -> скрытый нейрон 1	0,55032	Средний расход топлива -> скрытый нейрон 1
7	Тип привода(задний) -> скрытый нейрон 1	-0,17734	Тип привода(задний) -> скрытый нейрон 1	0,12022	Тип привода(задний) -> скрытый нейрон 1
8	Тип привода(передний) -> скрытый нейрон 1	-0,79045	Тип привода(передний) -> скрытый нейрон 1	0,14798	Тип привода(передний) -> скрытый нейрон 1
9	Тип привода(полный) -> скрытый нейрон 1	0,59179	Тип привода(полный) -> скрытый нейрон 1	-0,16916	Тип привода(полный) -> скрытый нейрон 1
10	Тип привода(полный) -> скрытый нейрон 1	-0,32405	Тип привода(полный) -> скрытый нейрон 1	-0,09447	Тип привода(полный) -> скрытый нейрон 1
11	Количество ступеней(4) -> скрытый нейрон 1	-0,63924	Количество ступеней(4) -> скрытый нейрон 1	0,12947	Количество ступеней(4) -> скрытый нейрон 1
12	Количество ступеней(5) -> скрытый нейрон 1	-0,48716	Количество ступеней(5) -> скрытый нейрон 1	-0,10497	Количество ступеней(5) -> скрытый нейрон 1
13	Количество ступеней(6) -> скрытый нейрон 1	-0,04100	Количество ступеней(6) -> скрытый нейрон 1	-0,05045	Количество ступеней(6) -> скрытый нейрон 1
14	Количество ступеней(7) -> скрытый нейрон 1	0,50185	Количество ступеней(7) -> скрытый нейрон 1	-0,04934	Количество ступеней(7) -> скрытый нейрон 1
15	Количество ступеней(8) -> скрытый нейрон 1	-0,03932	Количество ступеней(8) -> скрытый нейрон 1	-0,05661	Количество ступеней(8) -> скрытый нейрон 1
16	Тип коробки передач(автоматическая) -> скрытый нейрон 1	-1,54088	Тип коробки передач(автоматическая) -> скрытый нейрон 1	0,04201	Тип коробки передач(автоматическая) -> скрытый нейрон 1
17	Тип коробки передач(механическая) -> скрытый нейрон 1	1,06006	Тип коробки передач(механическая) -> скрытый нейрон 1	0,01497	Тип коробки передач(механическая) -> скрытый нейрон 1
18	Тип коробки передач(полуавтоматическая) -> скрытый нейрон 1	-0,08055	Тип коробки передач(полуавтоматическая) -> скрытый нейрон 1	-0,08696	Тип коробки передач(полуавтоматическая) -> скрытый нейрон 1
19	Объем двигателя -> скрытый нейрон 2	-0,74230	Объем двигателя -> скрытый нейрон 2	0,02163	Объем двигателя -> скрытый нейрон 2
20	Мощность двигателя -> скрытый нейрон 2	2,80075	Мощность двигателя -> скрытый нейрон 2	0,42160	Мощность двигателя -> скрытый нейрон 2
21	Емкость бака -> скрытый нейрон 2	-0,82318	Емкость бака -> скрытый нейрон 2	-0,21268	Емкость бака -> скрытый нейрон 2
22	Максимальная скорость -> скрытый нейрон 2	-4,83476	Максимальная скорость -> скрытый нейрон 2	-0,55758	Максимальная скорость -> скрытый нейрон 2
23	Время разгона до 100 км/ч -> скрытый нейрон 2	3,89788	Время разгона до 100 км/ч -> скрытый нейрон 2	0,06693	Время разгона до 100 км/ч -> скрытый нейрон 2
24	Средний расход топлива -> скрытый нейрон 2	-2,15069	Средний расход топлива -> скрытый нейрон 2	0,23619	Средний расход топлива -> скрытый нейрон 2
25	Тип привода(задний) -> скрытый нейрон 2	-0,63880	Тип привода(задний) -> скрытый нейрон 2	0,49500	Тип привода(задний) -> скрытый нейрон 2
26	Тип привода(передний) -> скрытый нейрон 2	1,61229	Тип привода(передний) -> скрытый нейрон 2	-0,39635	Тип привода(передний) -> скрытый нейрон 2
27	Тип привода(полный) -> скрытый нейрон 2	0,81373	Тип привода(полный) -> скрытый нейрон 2	0,14140	Тип привода(полный) -> скрытый нейрон 2
28	Тип привода(полный) -> скрытый нейрон 2	1,24328	Тип привода(полный) -> скрытый нейрон 2	-0,14606	Тип привода(полный) -> скрытый нейрон 2
29	Количество ступеней(4) -> скрытый нейрон 2	0,65764	Количество ступеней(4) -> скрытый нейрон 2	-0,32172	Количество ступеней(4) -> скрытый нейрон 2
30	Количество ступеней(5) -> скрытый нейрон 2	0,51512	Количество ступеней(5) -> скрытый нейрон 2	0,04704	Количество ступеней(5) -> скрытый нейрон 2
31	Количество ступеней(6) -> скрытый нейрон 2	1,18708	Количество ступеней(6) -> скрытый нейрон 2	0,29026	Количество ступеней(6) -> скрытый нейрон 2
32	Количество ступеней(7) -> скрытый нейрон 2	0,65370	Количество ступеней(7) -> скрытый нейрон 2	0,10057	Количество ступеней(7) -> скрытый нейрон 2
33	Количество ступеней(8) -> скрытый нейрон 2	0,00682	Количество ступеней(8) -> скрытый нейрон 2	0,07109	Количество ступеней(8) -> скрытый нейрон 2
34	Тип коробки передач(автоматическая) -> скрытый нейрон 2	0,96630	Тип коробки передач(автоматическая) -> скрытый нейрон 2	-0,04204	Тип коробки передач(автоматическая) -> скрытый нейрон 2

Рисунок 18 – Весы моделей

По кнопке «Статистики прогноза» можно получить максимумы/минимумы предсказанных значений и остатков для каждой из активных моделей. По кнопке «Коэффициенты корреляции» получаются корреляции между предсказанными значениями и целевыми значениями наблюдений. Кнопка «Описательные статистики» возвращает статистические показатели по выборкам для непрерывных входных переменных и для выходов. Кнопка «Анализ чувствительности (глобальный)» выдает таблицу с указанием влияния переменных на выход по каждой из моделей. Эта таблица представлена на рисунке 19.

		Чувствительность (Auto2_sta)								
		Выборки: Обучающая								
Сети	Максимальная скорость	Мощность двигателя	Время разгона до 100 км/ч	Количество ступеней	Тип привода	Тип коробки передач	Объем двигателя	Средний расход топлива	Ёмкость бака	
1.MLP 18-4-1	5,282509	4,610194	2,833811	1,537586	2,337618	1,848823	1,340631	1,556205	1,023654	
2.MLP 18-15-1	6,097912	3,914755	2,204432	2,046603	1,941073	1,933630	3,423479	1,179148	1,050523	
3.MLP 18-14-1	1,881784	1,024686	1,522640	1,256587	1,550266	1,464034	1,024688	1,298472	1,029921	
4.MLP 18-6-1	3,266016	1,166605	2,152171	1,854847	2,153826	1,657536	1,184429	1,638861	1,046611	
5.MLP 18-5-1	5,141171	3,483821	4,123831	3,234016	1,943149	3,018772	1,057467	1,735187	1,100437	
Среднее	4,333878	2,840012	2,567377	1,985928	1,985186	1,984559	1,606139	1,481575	1,050229	

Рисунок 19 – Таблица чувствительности для моделей

На вкладке «Наблюдения пользователя», представленной на рисунке 20, можно ввести пользовательские данные с целью получения результатов моделей на них.

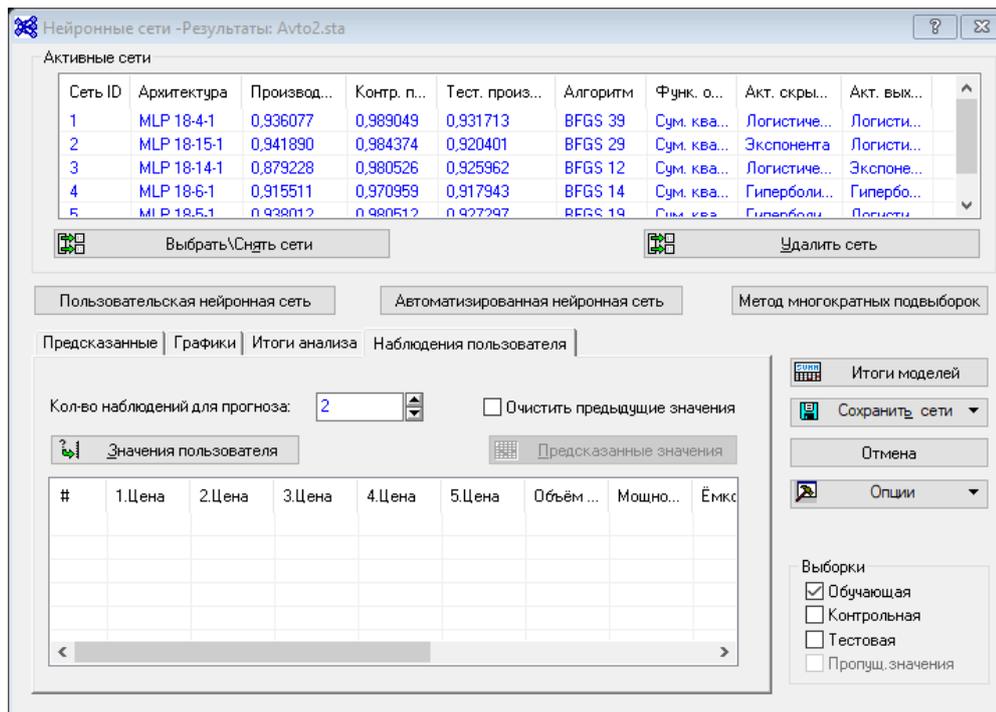


Рисунок 20 – Вкладка «Наблюдения пользователя»

Сперва указывается количество вводимых наблюдений в соответствующем окне, после чего нажать кнопку «Значения пользователя». Откроется окно, в которое следует ввести входные переменные. Введенные наблюдения отобразятся в таблице в нижней части окна наблюдений. Первыми значениями будут выходы соответствующих моделей, последующие – входные. Кнопка «Предсказанные значения» выведет их в отдельную таблицу. Флажок «Очистить предыдущие значения» позволяет очищать итоги моделей при добавлении новых наблюдений для анализа.

2.6 Сохранение и загрузка моделей

После создания моделей появляется возможность сохранить их в файл для дальнейшего использования. Чтобы это сделать, необходимо нажать кнопку «Сохранить сети» в окне анализа, либо в окне создания модели. В списке следует выбрать пункт PMML. В открывшемся окне следует указать место сохранения и префиксы файлов. Данные сохраняются в формате xml файлов по одному на каждую сеть.

Для загрузки этих файлов следует в окне на рисунке 3 выставить флажок «Загрузить модели из предыдущих анализов», после чего – нажать кнопку «Загрузить сети». В открывшемся окне указывается путь к файлам, после чего следует нажать «ОК». Откроется окно анализа моделей.

Важным моментом для загружаемых моделей является то, что в них все данные входят в контрольную выборку. Данные по точности загруженных моделей не сохраняются, недоступен анализ чувствительности. Кроме того, невозможно дополнить эти модели, перейдя в окно создания моделей.

3 Модели и реализация

3.1 Отбор моделей

В рамках решения задачи прогнозирования стоимостей автомобилей было создано пять моделей регрессии. Следует отметить, что в качестве дополнительной метрики предиктивной способности модели используются коэффициенты корреляции между полученными значениями сети, и целевыми значениями соответствующих наблюдений.

Таблица 1 – Сравнительные характеристики модели 1

Архитектура	MLP 17-17-1
Оптимизация	BFGS 21
Корреляция (обуч.)	0,9212494
Корреляция (контр.)	0,79792569
Корреляция (тест.)	0,949118292
Ошибка (обуч.)	102876540300
Ошибка (контр.)	32003927672
Ошибка (тест.)	299895636016
Функция активации (скрыт.)	Гиперболическая
Функция активации (выход.)	Тождественная

Из таблицы 1 видно, что при хороших коэффициентах корреляции на обучающей и тестовой выборках, модель 1 имеет не лучший коэффициент корреляции для контрольной выборки. При этом ошибка на контрольной выборке меньше, чем на тестовой (размеры выборок одинаковы).

Таблица 2 – Сравнительные характеристики модели 2

Архитектура	MLP 17-4-1
Оптимизация	BFGS 15
Корреляция (обуч.)	0,92894775
Корреляция (контр.)	0,895616506
Корреляция (тест.)	0,945925448
Ошибка (обуч.)	92089455720
Ошибка (контр.)	21486365637
Ошибка (тест.)	224714263432
Функция активации (скрыт.)	Гиперболическая
Функция активации (выход.)	Тождественная

Из таблицы 2 видно, что модель 2 превосходит модель 1 по всем показателям, кроме коэффициента корреляции для тестовой выборки. В частности, коэффициент корреляции для контрольной выборки здесь значительно ближе к единице.

Таблица 3 – Сравнительные характеристики модели 3

Архитектура	MLP 17-4-1
Оптимизация	BFGS 41
Корреляция (обуч.)	0,953101784
Корреляция (контр.)	0,920274596
Корреляция (тест.)	0,960677901
Ошибка (обуч.)	61458122655
Ошибка (контр.)	25058422202
Ошибка (тест.)	139118771029
Функция активации (скрыт.)	Гиперболическая
Функция активации (выход.)	Тождественная

Из таблицы 3 видно, что модель 3 превосходит обе представленные ранее модели по всем показателям. При этом, по сути – она представляет из себя дообученую модель 2.

Таблица 4 – Сравнительные характеристики модели 4

Архитектура	MLP 17-4-1
Оптимизация	BFGS 21
Корреляция (обуч.)	0,912518457
Корреляция (контр.)	0,721941678
Корреляция (тест.)	0,946014909
Ошибка (обуч.)	113750983865
Ошибка (контр.)	38684873250
Ошибка (тест.)	253255194918
Функция активации (скрыт.)	Гиперболическая
Функция активации (выход.)	Тождественная

Из таблицы 4 видно, что модель 4 заметно проигрывает моделям 2 и 3 в производительности на контрольной выборки, кроме того, ее ошибка относительно велика и на обучающей выборке.

Таблица 5 – Сравнительные характеристики модели 5

Архитектура	MLP 17-8-1
Оптимизация	BFGS 40
Корреляция (обуч.)	0,954064863
Корреляция (контр.)	0,952417325
Корреляция (тест.)	0,968141305
Ошибка (обуч.)	60253477954
Ошибка (контр.)	12826870213
Ошибка (тест.)	263860476209
Функция активации (скрыт.)	Гиперболическая
Функция активации (выход.)	Тождественная

Из таблицы 5 видно, что сеть 5 превосходит все остальные на обучающей и контрольной выборках, и лишь немного хуже сети 3 на тестовой выборках.

Выберем именно ее для программной реализации.

3.2 Программная реализация

Модель 5 была сохранена в xml. Из полученного файла были использованы веса для нейронов, а также данные для нормализации входных и выходных данных.

Созданная таким образом сеть была реализована средствами языка Python. В процессе использовались инструменты библиотеки NumPy, и фреймворка Tkinter. Окно графического интерфейса этой программы представлено на рисунке 21.

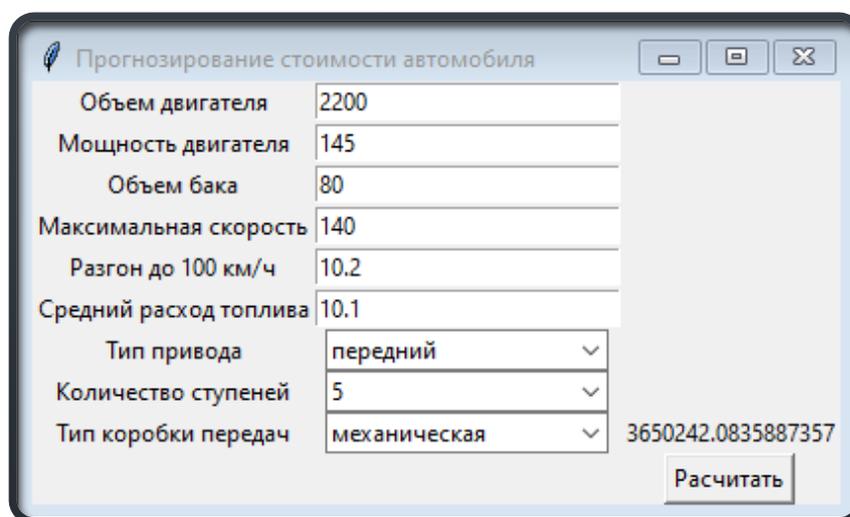


Рисунок 21 – Окно программы прогнозирования стоимости

В интерфейсе пользователь сразу может ввести данные предикторов для получения итоговой оценки стоимости. Особенностью программы является то, что нейронная сеть, используемая для получения результата, не требует обучения, так как ее веса уже сохранены.

ЗАКЛЮЧЕНИЕ

В ходе данной работы изучен процесс работы нейронной сети применительно к задаче нелинейной регрессии. Проведен анализ теоретических основ работы этого класса алгоритмов, его реализация в рамках модуля ANN пакета Statistica. Модуль позволяет автоматизировать часть процессов, связанных с реализацией методов, а именно – процессы создания архитектуры сети и подбора функций активации нейронов. Кроме того, он включает в себя инструменты анализа прогностической способности создаваемых сетей и инструменты сохранения/загрузки сетей.

Как итог работы была построена модель классификации MLP 17-8-1. У этой модели коэффициент корреляции для обучающей выборки – 0,954, для тестовой – 0,968, для контрольной – 0,952. Эта модель была сохранена, а затем использована для создания графического приложения, в котором можно быстро получить прогноз этой сети на данных, вводимых пользователем.

Таким образом, все поставленные задачи полностью достигнуты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Мак-Каллок У.С. Логическое исчисление идей, относящихся к нервной активности / Мак-Каллок У.С., Питтс В. // Автоматы: сб. науч. тр. – М.: Изд-во иностр. лит., 1956. – с.363–384.
- 2 Розенблатт Ф. Принципы нейродинамики: Перцептроны и теория механизмов мозга. / Розенблатт Ф. – М.: Мир, 1965. – 480 с.
- 3 Nocedal J. Numerical Optimization / J. Nocedal, S. J. Wright – М.: Springer Science+Business Media, LLC, 2006 – 664 p.
- 4 Электронное руководство Statistica. // (Engl.). – URL.: https://docs.tibco.com/pub/stat/13.3.0/doc/pdf/TIB_stat_13.3_quick_ref.pdf?id=1 [3 May 2021].