

УДК 004.85

DOI 10.17513/snt.40603

АЛГОРИТМ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ ДЛЯ ПРОГНОЗИРОВАНИЯ ИТОГОВЫХ ОЦЕНОК ОБУЧАЮЩИХСЯ

¹Димитриев А.П. ORCID ID 0000-0002-7345-9790,

¹Лавина Т.А. ORCID ID 0000-0002-7622-2246, ¹Яруськина Е.Т.,

²Баженов Р.И. ORCID ID 0000-0003-2668-1142.

¹Федеральное государственное бюджетное образовательное учреждение высшего образования
«Чувашский государственный университет имени И.Н. Ульянова», Чебоксары,
Российская Федерация, e-mail: dimitriev1@yandex.ru;

²Федеральное государственное бюджетное образовательное учреждение высшего образования
«Приамурский государственный университет имени Шолом-Алейхема», Биробиджан,
Российская Федерация

Рассматривается задача прогнозирования итоговых оценок студентов вуза. Прогноз высокой оценки может служить мотивацией студентам для успешного освоения соответствующей дисциплины. Для решения задачи прогнозирования итоговых оценок задействован математический аппарат искусственных нейронных сетей. Цель исследования – разработка практически реализуемого алгоритма обучения нейронной сети для прогнозирования оценок обучающихся. Исследование производилось в течение 11 месяцев с использованием тестового примера, основанного на наборе данных из 84 признаков, собранных о 126 студентах. Для обучения нейронной сети предложен базовый алгоритм оптимизации весов дуг, который можно классифицировать как стохастический градиентный спуск, который не впервые применяется при машинном обучении. Исследованы различные модификации базового алгоритма с целью выявления оптимальной модификации по отношению к скорости работы. В процессе исследования выполнены сотни однократных обучений нейронной сети. Определены значения параметров оптимальной модификации алгоритма, применение которых целесообразно с точки зрения скорости сходимости и количества ошибочных прогнозов. В частности, выявлено, на сколько подмножеств является наиболее целесообразным разделять набор данных при осуществлении кросс-валидации. В результате обучения точность прогнозирования в контрольной выборке является достаточно приемлемой.

Ключевые слова: стохастический градиентный спуск, нейронная сеть, учебная дисциплина, прогнозирование, итоговая оценка

NEURAL NETWORK TRAINING ALGORITHM FOR PREDICTING STUDENTS' FINAL GRADES

¹Dimitriev A.P. ORCID ID 0000-0002-7345-9790,

¹Lavina T.A. ORCID ID 0000-0002-7622-2246, ¹Yaruskina E.T.,

²Bazhenov R.I. ORCID ID 0000-0003-2668-1142

¹Federal State Budgetary Educational Institution of Higher Education "Chuvash State University named after I.N. Ulyanov", Cheboksary, Russian Federation, e-mail: dimitriev1@yandex.ru;

²Federal State Budgetary Educational Institution of Higher Education "Amur State University named after Sholom Aleichem", Birobidzhan, Russian Federation

The problem of predicting final grades of university students is considered. Predicting a high grade can serve as motivation for students to successfully master the corresponding discipline. To solve the problem of predicting final grades, the mathematical apparatus of artificial neural networks is employed. The objective of the study is to develop a practically implementable algorithm for training a neural network for predicting student grades. The study was conducted over a period of 11 months using a test example based on a dataset of 84 features collected from 126 students. To train the neural network, a basic algorithm for optimizing arc weights is proposed, which can be classified as stochastic gradient descent, which is not the first time used in machine learning. Various modifications of the basic algorithm were studied in order to identify the optimal modification in relation to operation speed. During the study, hundreds of one-time neural network training sessions were performed. The values of the parameters of the optimal modification of the algorithm, the use of which is advisable in terms of convergence rate and the number of erroneous predictions, were determined. In particular, it was revealed how many subsets it is most appropriate to divide the data set into when performing cross-validation. As a result of training, the prediction accuracy in the validation set was quite acceptable.

Keywords: stochastic gradient descent, neural network, academic discipline, forecasting, final assessment

Введение

Удовлетворение потребностей государства в квалифицированных (подготовленных к работе) кадрах – это одна из главных

целей любого образовательного учреждения профессионального образования [1]. В частности, это декларируется в качестве первой цели в уставе Чувашского государ-

ственного университета (ЧувГУ) [2]. Качество подготовки таких кадров является важным фактором для определения конкурентоспособности образовательного учреждения [3].

Качество подготовки выпускника выражает совокупность степеней его подготовленности со стороны различных преподавателей. Отдельные преподаватели сообщают о степени подготовленности обучающихся при помощи итоговых оценок по своим учебным дисциплинам. Высокая оценка может служить мотивацией студенту для успешного освоения соответствующей дисциплины. Подробнее мотивирующие факторы для получения той или иной оценки студентом рассмотрены в работе [4]. Для информирования студентов об ожидаемых оценках возможно применение прогнозирования этих оценок. Кроме мотивации, с учетом получаемого прогноза возможно также проведение некоторых мероприятий с целью повышения успеваемости [5]. Об актуальности прогнозирования успешности обучения студентов сообщается, к примеру, М.В. Носковым и соавт. [6; 7].

Одним из подходов к прогнозированию оценок является использование искусственной нейронной сети (НС). Для применения НС требуется выполнить ее обучение. В ЧувГУ разработано приложение, позволяющее обучать НС, предназначенную для прогнозирования оценок по дисциплине «Базы данных». Ожидается, что, чем больше будет данных, тем точнее будут прогнозы [8, с. 20]. Например, в [9] сообщается об учете при прогнозировании 86 признаков. В связи с этим приложение позволяет при прогнозировании оценок учитывать приблизительно столько же – 84 фактора (признака).

Прогнозирование осуществляется при помощи НС, представляющей собой простой персепtron. Всего на входе НС $m = 233$ значения по каждому студенту, полученные из 84 признаков. Имеется входной слой, состоящий из 233 нейронов, и один выходной нейрон. Каждое из 233 входных значений подается на входы каждого из 233 нейронов входного слоя с помощью m^2 дуг, обладающих весами. Как обычно в персептранах, каждый нейрон входного слоя вырабатывает выходное значение на основе функции активации (см. ниже) от суммы произведений входных значений на веса соответствующих дуг. Выходные значения каждого нейрона входного слоя подаются на вход выходного нейрона с помощью 233 дуг, имеющих веса. Выходной нейрон по такому же принципу, как и остальные нейроны,

выставляет выходное значение, используя функцию активации.

Кодировка признаков осуществляется следующим образом. На входе НС часть факторов учитывается как прямое числовое значение, например средняя оценка за второй семестр, а другая часть – как значение 0 или 1, выражающее принадлежность к тому или иному кластеру, например, к одной из 12 групп административно-территориальных районов. Существуют такие признаки, которые учитывают широкий диапазон чисел, например число жителей населенного пункта, откуда прибыл студент. В данном случае в качестве входного значения используется десятичный логарифм указанного числа, таким образом, диапазон значительно сужается, наподобие нормирования признаков. Некоторые признаки нормируются, например балл ЕГЭ по физике делится на 100 как на максимальное значение. В итоге все признаки представлены небольшими числами от 0 до 7, среди них нет стремящихся к нулю.

Задача обучения НС может быть выражена как задача оптимизации:

$$f(x) \rightarrow \min,$$

где x – набор входных данных:

1) матрица $n \times m$, в которой каждая строка соответствует одному из n студентов, а столбец – одному из m значений параметров, присущих студентам; $\min \geq 0$; $\min \in R$, где R – множество действительных чисел; f – целевая функция;

2) вектор-столбец итоговых оценок, по одной оценке O_i на студента, $i = 1, \dots, n$, где n – число студентов,

при ограничениях на неравенства и принадлежность к доменам, присущим каждому из указанных 233 значений. Например, значение «средний балл за 2 семестр» является вещественным числом от 2 до 5. Имеются некоторые специфические ограничения: например, студент может быть из одной и только одной группы административно-территориальных районов.

Поскольку числовое значение на выходе крайнего правого нейрона, обозначаемое S , однозначно определяется из совокупности весов дуг, обозначаемой W , и m входных значений НС, целевая функция может быть выражена как усредненное по всем студентам значение табличной функции от значения S_i на выходе крайнего правого нейрона и оценки O_i , полученной студентом по дисциплине «Базы данных»:

$$f = \sum_{i=1}^n f(S_i, O_i) / n,$$

$$f(S, O) = \begin{cases} 1700 - S, & O = 5 \& S < 1700, \\ 1300 - S, & O = 4 \& S < 1300, \\ S - 1700, & O = 4 \& S \geq 1700, \\ 900 - S, & O = 3 \& S < 900, \\ S - 1300, & O = 3 \& S \geq 1300, \\ S - 900, & O = 2 \& S \geq 900. \end{cases}$$

где S_i определяется как значение функции возбуждения нейрона (см. ниже).

При обучении НС некоторыми разработчиками применяется метод обратного распространения ошибки [10, с. 174]. Разработан программный модуль, реализующий данный метод для рассматриваемой сети, однако при анализе его кода выявлена положительная обратная связь, приводящая к чрезвычайному увеличению весов дуг (до значений 10 в 308 степени и т.п.). Это не только противоестественно, но и, в свою очередь, приводит к аварийному завершению программы из-за ошибок операций с плавающей запятой еще при неприемлемо высоких значениях целевой функции (около 180, см. контрольные значения ниже).

Другой вариант оптимизации весов дуг – использование метода однокоординатного спуска (ОКС), однако он работает неприемлемо долго. Реализация метода ОКС показывает, что для совершения всего одной итерации (то есть исследования изменений в каждой из координат) требуется около 4 ч машинного времени на указанной ниже вычислительной системе.

Время одной итерации для метода ОКС пропорционально произведению числа студентов n на число координат ($m^2 + m$, поскольку одна координата – это вес одной дуги НС), на время однократного вычисления целевой функции (пропорционально $m^2 + m$) и на количество экспериментальных значений для одной координаты (около 20). То есть временная сложность составляет $n m^4$. Можно попытаться несколько оптимизировать время вычислений за счет хранения части промежуточных результатов в дополнительных массивах, однако при этом потребовались бы массивы вещественных чисел размерностью $m \times m \times n$, что затратно по размеру требуемой оперативной памяти. В связи с этим актуальна разработка более эффективного алгоритма обучения рассматриваемой НС.

Цель данного исследования – разработка практически реализуемого алгоритма обучения НС для прогнозирования оценок обучающихся. Решаемые задачи: разработать базовый алгоритм, произвести его

программную реализацию, исследовать параметры алгоритма на оптимальность с помощью вычислительных экспериментов.

Материал и методы исследования

В качестве тестового примера рассматривается прогнозирование оценок 126 обучающихся в бакалавриате 3-го курса на кафедре компьютерных технологий ЧувГУ по дисциплине «Базы данных» за 2 года, по очной и очно-заочной формам обучения. В наборе данных имеются следующие оценки: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Данные получены из информационных систем ЧувГУ «Личный кабинет и портфолио обучающихся ЧувГУ», служебного портала ЧувГУ, годовых отчетов студенческого научного общества факультета, бумажного рабочего журнала преподавателя и других бумажных документов. Учитывается 84 признака студента. Студенты, по которым не удалось собрать все данные, не учитываются в исследовании. Полученный набор данных случайным образом делится на обучающую и контрольную выборки в соотношении, близком к используемому в [11] соотношению, то есть $101 : 26 \approx 80 : 22$, при этом для балансировки классов целевой переменной по каждому из классов, то есть для каждого из четырех значений оценки, отбирается приблизительно 80% соответствующих студентов. Так обеспечивается наличие представителей всех классов в контрольной выборке.

Общим методом исследования является проведение вычислительного эксперимента с вычислительной системой: ноутбук с процессором Intel(R) Core(TM) i5-7200U, 2712 МГц. Результаты эксперимента обрабатываются методами математической статистики. Выявление оптимальных параметров алгоритма оптимизации весов дуг НС представляет собой метаоптимизацию. При обучении НС используется кросс-валидация типа K-fold [12]. Сокращение набора признаков производится с помощью алгоритма AdDel, принцип работы которого сообщается в [11, с. 470]. Исследования производились с 05.12.2024 по 18.11.2025.

Разработка базового алгоритма

Предлагаемый алгоритм может рассматриваться как одна из вариаций стохастического градиентного спуска, нередко применяемого в машинном обучении [13]. Особенности данного алгоритма:

– целевой функцией f при оптимизации весов дуг является функция, напрямую зависящая от среднего отклонения по модулю

вычисленных НС оценок от фактических для всех студентов из обучающей выборки;

– исследуется окрестность текущей точки многомерного пространства, координаты которой представляют собой веса дуг НС;

– производится ряд случайных изменений в относительно небольшом числе случайно выбранных координат (практически используется 120 изменений, то есть изменения производятся в 0,22% координат);

– при нахождении удачных изменений формируется вектор изменений Δ , вдоль которого производятся новые изменения в координатах. При разработке алгоритма первая версия не предусматривала формирования такого вектора, производился каждый раз новый независимый ряд изменений, при этом оптимизационный процесс, как правило, останавливался в локальных экстремумах с неприемлемо высокими (около 60) значениями f ;

– при удачном, то есть приводящем к снижению f , перемещении текущей точки вдоль Δ , все компоненты вектора Δ умножаются на число 2 и происходит повторное перемещение по полученному вектору, и так несколько раз. Число подобных перемещений определяется виртуальным числом, которое не должно превышать максимума, например 100, показывающего, сколько могло быть сделано перемещений без умножения Δ на 2. Ранние варианты алгоритма не предусматривали умножения на 2, при этом скорость работы была существенно ниже. Максимум используется для приближения к принципам естественной НС, в которой веса дуг не могут стремиться к бесконечности.

Базовый алгоритм обучения НС следующий.

1. Загрузить:

1.1. Обучающую выборку, путем загрузки всего набора данных и загрузки сведений о разделении этого набора на обучающую и контрольную выборки.

1.2. Совокупность весов дуг НС, обозначаемую W .

1.3. Сведения об активности признаков.

2. Установить параметры:

2.1. Промежуточное число итераций H .

2.2. Максимальное число изменений НС на итерации N_{\max} .

2.3. Число изменений K в Δ .

2.4. Общее число совершенных итераций $It \leftarrow 0$.

3. Повторять H раз, то есть совершать итерации:

3.1. Сделать копию W , обозначаемую W_1 .

3.2. Вычислить целевую функцию $f_0 = f(W_1)$.

3.3. Установить число N изменений НС на данном шаге как случайное число от 1 до N_{\max} .

3.4. Произвести N случайных изменений в W_1 , равномерно распределенных между входными и выходными связями (дугами входного и выходного слоев) по их количеству. Изменения представляют собой действительные числа от -5 до +5 с точностью до 3 знаков после запятой. Очистить Δ и сохранить сведения об изменениях в Δ .

3.5. Вычислить целевую функцию $f = f(W_1)$.

3.6. Если $f \geq f_0$, то перейти к следующей итерации цикла по H .

3.7. Иначе:

3.7.1. Скопировать $f_0 \leftarrow f$, $W \leftarrow W_1$. Установить фактор масштабирования $F_m \leftarrow 0,5$.

3.7.2. Отобразить на экран номер итерации, N, f .

3.7.3. Установить счетчик шагов $c \leftarrow 0$.

3.7.4. Повторять, пока $c < 100$:

3.7.4.1. Удвоить F_m ; $c \leftarrow c + F_m$. То есть произвести виртуальное увеличение c на удвоенное значение.

3.7.4.2. Выполнить повторные изменения в весах дуг W_1 на соответствующие значения Δ , умноженные на F_m .

3.7.4.3. Вычислить целевую функцию $f_2 = f(W_1)$.

3.7.4.4. Если $f_2 < f_0$, то:

3.7.4.4.1. Скопировать $f_0 \leftarrow f_2$, $W \leftarrow W_1$.

3.7.4.4.2. Отобразить на экран номер итерации, N, f .

3.7.4.4.3. Перейти к следующему шагу (п. 3.7.4).

3.7.4.5. Иначе:

3.7.4.5.1. Произвести обратные изменения в W_1 по информации из Δ , то есть изменения по направлению, противоположному Δ .

3.7.4.5.2. $c \leftarrow c + 7$, то есть произвести виртуальное увеличение c на 7.

3.7.4.5.3. Перейти к следующему шагу (п. 3.7.4).

4. Отобразить на экран It и улучшение в целевой функции, полученное за шаг 3.

5. Если $f > 0 \& It < 200000$, при указании пользователя откорректировать H, N_{\max}, K и перейти к шагу 3.

6. Сохранить W как результат обучения. Конец.

Алгоритм является стохастическим, вследствие чего его временная сложность как функция точно не определяется, однако можно утверждать, что он многократно для каждого из n студентов использует подпрограмму вычисления целевой функции с временной сложностью t^2 .

Результаты исследования и их обсуждение

Произведено исследование работы предложенного алгоритма с помощью программной реализации с целью разработки наиболее оптимального алгоритма из неко-

торых вариантов. Модификации алгоритма, образующие различные варианты, такие:

1. В п. 3.7.4.5, если $F_m > 1$, производить также уменьшение F_m в четыре раза, предполагая, что можно успешно перемещать текущую точку в направлении Δ , но не в два раза дальше, чем ранее, а только в 0,5 раза.

2. Между п.п. 3.7.4.5.1 и 3.7.4.5.2 вставить еще один пункт – выполнить K случайных преобразований Δ , если $F_m < 1$ (либо $F_m \leq 1$), так как существует вероятность, что если текущая точка ранее успешно перемещалась в этом направлении, она будет успешно перемещаться в немного измененном направлении.

Использовались как различные наборы признаков, генерируемые случайным образом, так и один и тот же набор признаков. В каждом эксперименте оптимизация производилась от исходного значения f (около 1207) до контрольных значений – 100, затем 10. Значения 1 и тем более 0 в некоторых случаях оптимизационный процесс не достигал, несмотря на работу свыше 2 часов, что выяснилось в результате седьмого по счету запуска этого процесса, поэтому до таких значений оптимизация далее не некоторого момента не рассматривалась.

Всего выполнено 48 подобных экспериментов. Средние значения времени достижения значений $f = 100$ и $f = 10$ для различных вариантов алгоритма следующие (в секундах):

1. С уменьшением F_m и случайными преобразованиями Δ – 26,5 и 236.

2. Без уменьшения F_m , но со случайными преобразованиями Δ – 19 и 130,5.

3. С уменьшением F_m , но без случайных преобразований Δ – 69 и 722.

4. Без уменьшения F_m и без случайных преобразований Δ – 27 и 206.

Таким образом, наивысшую скорость работы показывает вариант 2, который представлен вышеизложенным алгоритмом.

Исследована функция возбуждения (называемая также передаточной функцией, функцией активации) нейрона F для варианта 2 как наиболее перспективного. Использовались следующие значения: максимальное значение порога возбуждения $p_2 = 500$, а минимальное: $p_1 = 0,95p_2$. В качестве функции возбуждения вначале применялась кусочно-линейная функция, состоящая из трех линий, которая описывается следующим образом.

До достижения суммарного значения на входе $\sigma = p_1$ значение $F = (\sigma - p_1) / 1000$. При $p_1 \leq \sigma \leq p_2$ функция линейно возрастает по формуле $F = (\sigma - p_1) / (p_2 - p_1)$ до значения $F = 1$, и далее вычисляется по формуле $F = 1 + (\sigma - p_2) / 1000$. Это значение выставляется на выход нейрона.

Однако обычно чаще в литературе говорится об использовании гладкой функции возбуждения. Проанализировано использование в базовом алгоритме такой функции на основе плавной функции знака – softsign [14] со смещением: $F = (\sigma - p) / (|\sigma - p| + 1)$, где p – смещение относительно начала координат (центр порога), $p = 0,975p_2$. Выполнено 20 экспериментов, в результате выявлено, что при $p = 0$ алгоритм сходится к значению 10 примерно в 1,67 раз быстрее, чем при $p = 487,5$. Для применения метода обратного распространения применялся также другой вид гладкой функции возбуждения – логистическая функция (сигмоида) [10, с. 170], однако в этом случае требуется проверка ограничений для возведения в степень во избежание ошибок арифметического переполнения. Наложение подобных ограничений приводило к остановке оптимизационного процесса в локальных экстремумах, поэтому далее сигмоида не рассматривалась.

Применение гладкой или кусочно-линейной функции практически не оказывает влияния на скорость сходимости при $p = 487,5$, то есть при смещении порога относительно 0, равном 487,5.

Исследовано влияние величины порога на скорость сходимости к $f \leq 10$ при кусочно-линейной функции возбуждения. Для этого проведено по 10 экспериментов для каждого из трех значений $p_2 \in \{1, 10, 500\}$. При $p_2 = 500$ среднее время сходимости составило 118,6 с, при $p_2 = 10$ составило 95,9 с, при $p_2 = 1$ соответственно 92,2 с. Значение $p_2 = 0$ не исследовалось, поскольку минимум порога в таком случае будет совпадать с его максимумом.

Таким образом, целесообразно использовать несмещенный относительно 1 порог. Далее в экспериментах использована гладкая функция возбуждения на основе функции softsign и параметры, оптимальность которых выявлена путем проведения ряда экспериментов с варьированием этих параметров:

1. Число случайных преобразований (при добавлении таких преобразований после п. 3.7.4.5.1 базового алгоритма) $K = 10$.

2. Константа $cstop = 100$ во фразе «повторять, пока $c < 100$ » (см. п. 3.7.4 базового алгоритма; число 100 программное приложение позволяет изменять на другое целое число).

3. Центр порога возбуждения $p = 0,975$, что соответствует значению $p_2 = 1$.

4. $N_{\max} = 120$.

Для повышения эффективности базового алгоритма выполнено:

1. Сокращение набора признаков с помощью алгоритма AdDel. В итоге исполь-

зуется 33 признака из 84. Это направлено не только на сокращение времени по сбору исходных данных, но и на избежание проблемы «переобучения» НС [15].

2. Кросс-валидация типа K-fold. При этом исследована зависимость точности прогнозирования от количества подмножеств (так называемых «фолдов»), на которые делится обучающая выборка набора данных: для 14 значений количества от 4 до 50 проведено в среднем по 25,7 экспериментов. Выявлена наибольшая целесообразность деления набора данных x на $n_{fold} = 11$ подмножеств x_1, \dots, x_{11} , что уточнено проведением повышенного количества (40) экспериментов с этим значением. Точность выше на 4% от среднего значения точности для 14 вышеуказанных значений. При этом усредненное время обучения для разных значений количества изменяется от -4 до +61% по сравнению с временем при $n_{fold} = 11$.

При использовании кросс-валидации исследована зависимость качества прогнозов от метода приближения к минимуму f в обучающей выборке. Для выявления среднестатистических показателей произ-

ведено по 10 раз обучение тремя методами (30 экспериментов):

1. Обучение по каждому x_i ($i = 1, \dots, 11$) непосредственно до $f(x_i) = 0$, то есть $f(x_i) \rightarrow 0$.

2. Сначала обучение по каждому x_i ($i = 1, \dots, 11$) до значения $f(x_i) = 100$, затем то же, но до $f(x_i) = 10$, затем, аналогично, до $f(x_i) = 0$, что обозначим как $f(x_i) \rightarrow \{100, 10, 0\}$.

3. По аналогии с предыдущим методом, но вместо трех значений с пятью значениями $f(x_i)$: 100, 20, 7, 1, 0, что обозначим как $f(x_i) \rightarrow \{100, 20, 7, 1, 0\}$.

Обученные таким образом НС затем применялись для прогнозирования оценок в контрольной выборке. Наименьшее число ошибочных прогнозов получено третьим методом; первый метод производит в среднем на 16% больше таких ошибок, а второй на 5% больше. Время работы для этих трех методов отличается ненамного: второй метод наискорейший, третий требует в среднем на 10% больше времени, а первый – на 24%.

Поскольку число ошибок важнее при незначительном отличии скорости работы, наилучшим следует считать третий метод.

Усредненные значения достигаемой точности и времени исполнения при вариации параметров алгоритма

Измененный параметр	Время, с	Точность, %	Количество экспериментов
Ничего не изменилось (базовая совокупность)	81	72,9	40
$n_{fold} = 4 \dots 9$	92–131	67,7…71,4	130
$n_{fold} = 10$	112	70,9	30
$n_{fold} = 12$	92	70,7	30
$n_{fold} \in \{15; 20; 25; 30; 50\}$	78–103	69,1…70,5	130
$f(x_i) \rightarrow \{100, 10, 0\}$	99	70,6	20
$f(x_i) \rightarrow 0$	85	67,4	20
Признаки все (AdDel не применяется)	202	69,2	20
Функция активации кусочно-линейная	137	71,8	20
$p = 487,5$	238	71,4	20
$p = 9,75$	121	70,6	20
Осуществляется уменьшение F_m	98	68,6	20
$K = 0$	98	70,4	20
$K = 5$	103	69,6	20
$K = 20$	89	71,8	20
$cstop = 10$	107	66,8	20
$cstop = 1000$	169	73,4	20
$N_{max} = 60$	101	69,4	20
$N_{max} = 400$	58	70,8	20

Примечание: составлена авторами на основе полученных данных в ходе исследования.

Числовые показатели обучения по данным экспериментов с оптимальными параметрами алгоритма следующие:

- среднее время работы составляет 81 с (от 50 до 152 с) при среднеквадратичном отклонении 31 с;
- среднее количество ошибочных прогнозов в контрольной выборке 27,1%, то есть точность прогноза 72,9% (6,8 ошибок из 25, от 4 до 10 ошибок). Это можно считать приемлемым результатом: например, в [1] сообщается о точности прогноза 67% и выше.

Затем проведена серия дополнительных экспериментов. При следующей совокупности значений параметров алгоритма (назовем ее базовой) наблюдается повышенная точность прогнозирования при относительно небольшом времени исполнения:

- количество признаков сокращается в соответствии с алгоритмом AdDel;
- $f(x_i) \rightarrow \{100, 20, 7, 1, 0\}; cstop = 100;$
- $N_{\max} = 120; K = 10; n_{fold} = 11;$
- уменьшение F_m не осуществляется;
- функция активации softsign со смещением $p = 0,975$.

В таблице приводятся усредненные сведения о достигаемой точности и затрачиваемом для этого времени при изменении одного из параметров по сравнению с базовой совокупностью с указанием количества проведенных при этом экспериментов.

В приведенных данных только при $cstop = 1000$ точность выше, чем при базовой совокупности значений параметров, однако при этом время выполнения увеличивается более чем вдвое. Остальные изменения характеризуются более низкой достигаемой точностью. Таким образом, базовая совокупность – это, с высокой вероятностью, оптимум, во всяком случае, локальный, для задачи выбора оптимальных значений параметров.

Заключение

Применение кросс-валидации позволяет значительно чаще достигать нулевых ошибок при обучении НС. Без использования данного метода оптимизационный процесс практически останавливался в локальном минимуме в одном из нескольких экспериментов. С использованием кросс-валидации такого не произошло ни в одном из 600 выполненных подряд экспериментов.

Выявлены следующие значения параметров и варианты использования алгоритма, применение которых представляется целесообразным с точки зрения скорости сходимости и количества ошибочных прогнозов:

- в векторе изменений после безрезультатного перехода по нему выполня-

ются случайные преобразования в количестве $K = 10$;

- максимум виртуального числа повторных переходов по вектору изменений – 100;
- функция возбуждения гладкая на основе softsign с центром порога срабатывания нейрона $p = 0,975$;
- максимальное число изменений весов дуг для формирования вектора изменений $N_{\max} = 120$.
- уменьшения фактора масштабирования не производится;

– кросс-валидация типа K-fold с делением набора данных на 11 подмножеств и последовательным обучением по каждому подмножеству до пяти убывающих значений целевой функции, заканчивая нулевым.

Программное приложение, реализующее предложенный алгоритм, позволяет прогнозировать не только итоговую оценку по дисциплине «Базы данных», но и средний балл по всем дисциплинам за 5 семестр. Таким образом, алгоритм обладает определенной универсальностью.

Вышеизложенный алгоритм с модификацией, представляющей собой перебор сочетаний по три признака, реализуется магистрантами кафедры компьютерных технологий в качестве лабораторной работы по выбору наиболее информативной системы признаков для прогнозирования на наборах данных по вариантам.

Список литературы

1. Помян С.В., Белоконь О.С. Прогноз результатов успеваемости студентов вуза на основе марковских процессов // Вестник Вятского государственного университета. 2020. № 4 (138). С. 63–73. URL: [https://vestnik43.ru/assets/mgr/docs/vyatsu/138\(4\)2020/pomyanbelokoni.pdf](https://vestnik43.ru/assets/mgr/docs/vyatsu/138(4)2020/pomyanbelokoni.pdf) (дата обращения: 29.10.2025). DOI: 10.25730/VSU.7606.20.057. EDN: SSKESZ.
2. Устав Федерального государственного бюджетного образовательного учреждения высшего образования «Чувашский государственный университет имени И.Н. Ульянова»: утвержден Приказом Минобрнауки РФ от 27 декабря 2018 г. № 1317. 41 с. [Электронный ресурс]. URL: https://www.chuvsu.ru/sveden/files/ustav_obraz_org_chgu.pdf (дата обращения: 22.09.2025).
3. Мутуев Г.И. Методика «Контрольных недель» в анализе и оценке качества образовательного процесса в вузе // Концепт. 2017. Т. 41. С. 35–37. URL: <http://e-koncept.ru/2017/771346.htm> (дата обращения: 22.09.2025). EDN: YTTNYF.
4. Апатова Н.В., Гапонов А.И., Майорова А.Н. Прогнозирование успеваемости студентов на основе нечеткой логики // Современные научно-исследовательские технологии. 2017. № 4. С. 7–11. URL: <https://top-technologies.ru/ru/article/view?id=36630> (дата обращения: 22.09.2025). EDN: YNEOXV.
5. Мойсеенко С.С. Адаптивный метод прогнозирования успеваемости студентов // Известия Балтийской государственной академии рыбопромыслового флота: психологопедагогические науки. 2025. № 1 (71). С. 7–9. URL: <https://bgafpf.elpub.ru/jour/article/view/213/202> (дата обращения: 22.09.2025). EDN: HOGGBE.
6. Носков М.В., Вайнштейн Ю.В., Сомова М.В. Ключевые элементы дизайна электронного обучающего курса,

- обеспечивающего качественное прогнозирование успешности обучения студентов // Вестник Российской университета дружбы народов. Серия: Информатизация образования. 2025. Т. 22. № 3. С. 288–303. URL: <https://journals.rudn.ru/informatization-education/article/view/46156> (дата обращения: 31.10.2025). DOI: 10.22363/2312-8631-2025-22-3-288-303. EDN: QQCBZI.
7. Kustitskaya T.A., Esin R.V., Vainshtein Yu.V., Noskov M.V. Hybrid Approach to Predicting Learning Success Based on Digital Educational History for Timely Identification of At-Risk Students // Education Sciences. 2024. Vol. 14. Is. 6. P. 657. URL: <https://www.mdpi.com/2227-7102/14/6/657> (дата обращения: 31.10.2025). DOI: 10.3390/educsci14060657. EDN: ARKNTV.
8. Майер-Шенбергер В., Кукея К. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим. М.: Мани, Иванов и Фербер, 2014. 240 с. [Электронный ресурс]. URL: https://www.mann-ivanov-ferber.ru/assets/files/bookparts/bolshie-dannye/big_data_read.pdf (дата обращения: 22.09.2025). ISBN: 978-5-91657-936-9.
9. Brahim G.B. Predicting Student Performance from Online Engagement Activities Using Novel Statistical Features // Arabian Journal for Science and Engineering. 2022. Vol. 47. Is. 8. P. 10225–10243. URL: <https://link.springer.com/article/10.1007/s13369-021-06548-w> (дата обращения: 31.10.2025). DOI: 10.1007/s13369-021-06548-w.
10. Рассел С., Норвиг П. Искусственный интеллект: современный подход / Обучение, восприятие и действие. 4-е изд. Т. 3. Пер. с англ. СПб.: ООО «Диалектика», 2022. 640 с. [Электронный ресурс]. URL: <https://www.williamspublishing.com/cgi-bin/more.cgi?isbn=978-5-907365-25-4> (дата обращения: 29.10.2025). ISBN 978-5-907365-27-8.
11. Борисова И.А., Кутненко О.А. Очистка данных от диагностических ошибок в признаковых пространствах большой размерности // Математическая биология и биоинформатика. 2019. Т. 14. № 2. С. 464–476. URL: https://www.matbio.org/2019/Borisova_14_464.pdf (дата обращения: 22.09.2025). DOI: 10.17537/2019.14.464. EDN: ULTEVJ.
12. Аджан И.Х., Авксентьев Е.Ю. Методы прогнозирования когнитивной нагрузки обучающихся в средах электронного обучения с использованием данных отслеживания взгляда // Вестник Российской нового университета. Серия: Сложные системы: модели, анализ и управление. 2025. № 1. С. 81–91. URL: <https://vestnik-rosnou.ru/сложные-системы-модели-анализ-и-управление-complex-systems-models-analysis-management/2025/1/81> (дата обращения: 31.10.2025). DOI: 10.18137/RNU.V9187.25.01.P81. EDN: QSPRLU.
13. Бритов В.С., Мартышкин А.И., Данилов Е.А. Обзор и сравнение методов оптимизации применяемых в машинном обучении // Тенденции развития науки и образования. 2023. № 97–12. С. 45–49. URL: https://doicode.ru/doifile/lj/97/trnio05.2023_p12.pdf (дата обращения: 30.09.2025). DOI: 10.18411/trnio-05-2023-655. EDN: WPBHB.
14. Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks // Journal of Machine Learning Research. 2010. Vol. 9. P. 249–256. URL: https://www.researchgate.net/publication/215616968_Understanding_the_difficulty_of_training_deep_feedforward_neural_networks (дата обращения: 30.09.2025).
15. Guyon I., Weston J., Barnhill S., Vapnik V. Gene Selection for Cancer Classification using Support Vector Machines // Machine Learning. 2002. Vol. 46. Is. 1. P. 389–422. URL: <https://link.springer.com/article/10.1023/A:1012487302797> (дата обращения: 22.09.2025). DOI: 10.1023/A:1012487302797.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.