УДК 004.032.26

ПОИСК ДОПУСТИМЫХ ЗНАЧЕНИЙ ПАРАМЕТРОВ ЗАЯВКИ НА КРЕДИТОВАНИЕ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

Н. И. Цуканова, к.т.н., доцент кафедры ВПМ РГРТУ, Рязань, Россия;
orcid.org/0000-0001-7337-8037, e-mail: ninakorobova77@gmail.com
К. Г. Шитова, ведущий инженер по разработке, ПАО Сбербанк, Рязань, Россия;
orcid.org/0000-0002-7809-564X, e-mail: kshitova@inbox.ru

*Рассматриваются вопросы подбора параметров заявки клиента на кредитование с целью перевода заемщика из класса «отказано в кредитовании» в класс «разрешена выдача кредита». **Целью работы** является повышение качества процедуры оценки кредитоспособности клиента, направленное на увеличение числа заемщиков путем согласования с ними и/или выдачи им рекомендаций по изменению параметров заявки на кредит. Для достижения поставленной цели в работе были решены следующие задачи. Разработана на языке Python глубокая нейронная сеть, решающая задачу бинарной классификации клиентов по их характеристикам на два класса –* (*допущен, не допущен*) *к кредитованию. Показано, что недостатком такой процедуры является большой отсев клиентов, которые могли бы получить кредит и принести банку доход, но на других более жестких условиях. Предложены методика и алгоритм поиска значений параметров заявки на кредит, позволяющих допустить клиента к кредитованию.*

Ключевые слова: *оценка кредитоспособности клиента, заявка на кредит, глубокие нейронные сети, язык Python, библиотека Keras, алгоритм обратного распространения ошибки, сопряженный вектор, функция цели* (*потерь*)*.*

DOI: 10.21667/1995-4565-2021-78-89-101

Введение

В последнее время при широком распространении кредитования как физических, так и юридических лиц приобрела особую важность процедура оценки кредитоспособности заемщика. В статье [1] рассмотрены вопросы создания нейронной сети (НС), позволяющей делать оценку кредитоспособности предприятий малого бизнеса. Настоящая статья является продолжением идей, заложенных в [1], но реализованных на базе новых программных продуктов: языка Python и библиотек Keras и TensorFlow. Рассматривается задача оценки кредитоспособности как физических, так и юридических лиц.

В настоящей статье, кроме реализации классификатора в виде глубокой нейронной сети на языке Python, дополнительно рассматривается новая задача – как не потерять клиента, попавшего в разряд «некредитоспособных».

Клиент при обращении в банк за получением кредита вводит (сообщает) о себе данные, которые включают: а) характеристики покупаемого им кредитного продукта (сумму кредита, срок кредитования и т.п.); б) показатели, характеризующие его финансово-экономическую устойчивость (доход, сумму залога, кредитную историю и т.п.). Все эти данные образуют вектор характеристик клиента, в дальнейшем называемый **заявкой на кредит**. Оценивая эти данные, классификатор (эксперт или программа) принимает решение о разрешении или отказе в выдаче кредита.

Банки заинтересованы удержать клиента, так как его обслуживание приносит прибыль. Удержать, следовательно, допустить к кредитованию можно только при изменении параметров заявки на кредитование. Изменения могут коснуться как характеристик кредитного продукта, так и характеристик финансово-экономической устойчивости клиента.

Целью изменений параметров заявки является перевод заемщика из класса 0 (не допущен) в класс 1 (допущен к кредитованию).

Для оценки кредитоспособности клиента разработано много различных методов [1 – 4]. В последние годы исследователи отдают предпочтение методам машинного обучения [1, 5 – 8], суть которых состоит в настройке параметров некоторой модели в процессе обучения на экспериментальных данных, накапливаемых в базах данных. В качестве моделей обучения наиболее успешно используются деревья решений, метод ближайших *k*-соседей, нейронные сети (НС) и ансамбли моделей.

В работе [1] предлагается для оценки кредитоспособности предприятий использовать многослойную (глубокую) нейронную сеть. В последние годы этим моделям уделяется особое внимание [8 – 10, 15 – 19], в результате накоплен опыт по методикам повышения эффективности обучения глубоких нейронных сетей, который нашел свое отражение в библиотеках TensorFlow, Keras, Theano. Именно эти библиотеки и язык Python были выбраны для решения поставленной задачи.

Цель работы – создать на языке Python с использованием библиотек *TensorFlow*, *Keras* программу для оценки кредитоспособности клиента, которая, в случае отрицательного решения о допуске к кредитованию, реализует поиск приемлемых по допустимости к кредитованию значений параметров заявки на кредит. Программа должна быть основана на машинном обучении многослойных нейронных сетей.

Для достижения этой цели необходимо:

– рассмотреть постановку задачи кредитования заемщиков и показать возможность ее решения с помощью нейронной сети;

– провести анализ математической модели процесса обучения нейронной сети как задачи управления дискретным динамическим процессом и на ее основе разработать методику подбора параметров заявки на кредит;

– экспериментально исследовать ее применимость к решению поставленной задачи и дать интерпретацию полученным результатам.

Теоретическая часть

Постановка задачи оценки кредитоспособности заемщиков

Кредитоспособность заемщика – это его способность своевременно и в полном объеме погашать свои краткосрочные обязательства. Уровень кредитоспособности клиента характеризует его финансовое состояние. Чем выше финансовая устойчивость, тем выше кредитоспособность. Задача оценки платежеспособности заемщика заключается в классификации его по степени финансового риска. Если степень финансового риска высока, то принимается решение отказать в выдаче кредита. Оценка степени финансового риска предполагает анализ информации о заемщике. Для предприятий можно рассмотреть определенные финансовые коэффициенты (например, выручка, деленная на общие активы, промышленность и т.п.). Для физических лиц такими финансовыми коэффициентами являются возраст, семейное положение, доход, кредитная история и т.п. Все эти коэффициенты называются признаками объекта классификации.

В настоящей работе ставится задача разработать классификатор на базе нейронной сети для автоматизированной оценки кредитоспособности клиента по его заявке. Для решения поставленной задачи необходимо выбрать архитектуру сети, обучить нейронную сеть на исторических данных, затем оценить ее качество на тестовых выборках и только потом применять на практике.

Предполагается, что историческая информация доступна в виде данных, где каждая запись содержит заявку на кредит заемщика и его кредитный статус [допущен (класс 1)/не допущен (класс 0) к кредитованию], который был присвоен ему либо экспертами, либо рейтинговыми агентствами. Классификатор (НС) затем используется для присвоения класса (0 или 1) новым клиентам. На практике полученные значения, скорее всего, будут считаться предварительными.

Процесс обучения нейронной сети как задача оптимального управления дискретным динамическим процессом

В работе [1] задача подстройки весовых коэффициентов многослойной нейронной сети формулируется как задача оптимального управления [14] дискретным динамическим процессом.

Рассматривается многослойная нейронная сеть, число слоев нейронов в ней равно N.

Пусть n – номер очередного слоя сети;

$\boldsymbol{x(n)}=\{x_0(n),x_1(n),\ x_2(n),\dots x_{m_n}(n)\}$ – сигналы на выходе n-го и на входе $n+1$-го слоя сети, m_n – число нейронов в слое n; $\boldsymbol{x(n)}$ – вектор-строка; $\boldsymbol{x^T(n)}$ – вектор-столбец; T – знак транспонирования;

$\boldsymbol{u1(n)} = \boldsymbol{W(n)}$ – матрица весовых коэффициентов n-го слоя сети размером $[m_{n-1}: m_n]$;

$\boldsymbol{u2(n)} = \{b_i(n)\}, i = \overline{1, m_n}$ – вектор смещений для n-го слоя;

$\boldsymbol{x(0)}=\{x_1(0)\ x_2(0),\dots,\ x_r(0)\}$ – сигналы, подаваемые на вход нейронной сети – параметры заявки на кредитование; r – число параметров;

$n = 1,2,\dots,N$;

$\boldsymbol{x(N)}$ – сигнал на выходе нейронной сети – соответствует классу заемщика (0 – допущен, 1 – не допущен к кредитованию);

$(\boldsymbol{x^k(0)} = \boldsymbol{a^k}) \rightarrow (\boldsymbol{x^k(N)} = \boldsymbol{d^k}),\ k = 1,2,\dots P$ – обучающие примеры: при k-м входном сигнале $(\boldsymbol{x^k(0)} = \boldsymbol{a^k})$ на выходе сети необходимо обеспечить сигнал $(\boldsymbol{x^k(N)} = \boldsymbol{d^k})$ путем выбора значений весовых коэффициентов $\boldsymbol{W(n)}$ и величины смещений $\boldsymbol{b(n)}$, $n = 1,2,\dots,N$.

Тогда задача определения параметров нейронной сети может быть сформулирована следующим образом.

Найти такие значения оптимального управления

$\boldsymbol{u(n)}=\{\boldsymbol{u1(n)},\boldsymbol{u2(n)}\} = \{\boldsymbol{W(n)},\boldsymbol{b(n)}\}$, $n = 1,2,\dots,N$,

при которых будут выполнены следующие ограничения:

$$\begin{gathered}\boldsymbol{x(0)} = \boldsymbol{a^k};\ n = \overline{1,N}\\ \boldsymbol{z(n)} = \boldsymbol{W(n)}\bullet\boldsymbol{x(n\text{-}1)}+\boldsymbol{b(n)};\ \boldsymbol{x(n)} = \boldsymbol{f^n(z(n))};\\ \boldsymbol{x(N)} = \boldsymbol{d^k};\ k = \overline{1,P}\end{gathered} \tag{1}$$

и критерий качества $C(x(N)$ классификатора (НС) достигнет своего минимального значения

$$\begin{gathered}min\ C(\boldsymbol{x(N)}) = min \sum_{k=1}^{P} (x^k(N) - d^k)^2 .\\ \boldsymbol{u(n)},\ n = 1,2,\dots,N.\end{gathered} \tag{2}$$

Здесь $f^n(\boldsymbol{z(n)})$ – активационная нелинейная функция; $\bullet$ – знак умножения матрицы на вектор; $\boldsymbol{z(n) = W(n)\bullet x(n-1)+b(n)}$ – вектор сумм взвешенных сигналов для нейронов n-го слоя сети; $n = 1,2,\dots N$; P – число обучающих примеров.

Сопряженная система

Система (1) называется прямой системой уравнений. Она используется для получения сигнала на выходе нейронной сети $\boldsymbol{x(N)}$ при заданных значениях весовых коэффициентов $\boldsymbol{W}$ и смещений $\boldsymbol{b}$ и известном входном сигнале $\boldsymbol{x(0)}$.

Наряду с прямой системой важную роль играет сопряженная (двойственная) система, которая определяется соотношениями:

$$\begin{gathered}p(N) = \frac{\partial C(x(N))}{\partial x(N)};\\ p(n-1) = \left[\frac{\partial f(x(n-1),u(n))}{\partial x(n-1)}\right]^T \cdot p(n),\ (n = N,\dots,1),\end{gathered} \tag{3}$$

где $p(n) = \{p_1(n),\dots, p_{mn}(n)\}$,

$$\frac{\partial f(x,u)}{\partial x}=\left[\frac{\partial f_i(x,u)}{\partial x_j}\right] \quad i=1,...,m_n; \quad j=1,...,m_{n-1};$$

m_{n-1}, m_n – число нейронов в слоях $n-1$, n.

Она описывает процесс, который в литературе принято называть «обратным распространением ошибки» $\boldsymbol{p(n)}$.

Векторы $\boldsymbol{p(n)}$, $n = 0,1,2,\ldots,N$ называются сопряженными, они получаются в результате решения системы (3), им можно приписать следующий смысл.

Во-первых, они являются коэффициентами чувствительности $\left\{\frac{\partial C(x(N))}{\partial x(n)}\right\}$ критерия качества (2) к изменениям входного сигнала и выходных сигналов слоев нейронной сети $\boldsymbol{x(n)}$, $n = 0,1,2,\ldots,N$. Во-вторых, каждый из них $\boldsymbol{p(n)}$ является вектором градиента функции цели $C(\boldsymbol{x(N)})$ в зависимости от переменной $\boldsymbol{x(n)}$, следовательно, задает направление наискорейшего возрастания $C(\boldsymbol{x(N)})$ при изменениях сигнала $\boldsymbol{x(n)}$ на выходе слоя n, чем можно воспользоваться в градиентных методах оптимизации.

Обратим особое внимание на вектор $\boldsymbol{p(0)}$. Он характеризует влияние входного сигнала нейронной сети $\boldsymbol{x(0)}$ на значения функции цели $C(\boldsymbol{x(N)})$, а именно, задает вектор градиента для $C(\boldsymbol{x(N)})$ в зависимости от входного сигнала. В нашей задаче входной сигнал – это параметры заявки клиента на кредит.

Если задать функцию цели как переход в новый класс (из класса 0 в класс 1), следовательно, решить задачу минимизации ошибки между текущим классом и заданным, то задачу подбора параметров заявки на кредит можно решить методом градиентного спуска, каждая итерация которого описывается формулой: $\boldsymbol{x^{k+1}(0) = x^k(0) - \lambda \cdot p^k(0)}$, $k = 1,2,\ldots$. Здесь k – номер итерации в алгоритме градиентного спуска; λ – длина шага на k – той итерации.

Есть еще одна важная особенность вектора $\boldsymbol{p(0)}$. Он позволяет построить рейтинг параметров заявки на кредит по своему воздействию на функцию цели и выявить среди них параметры с наибольшим влиянием. Для этого нужно с помощью обученной нейронной сети найти среднее значение модуля вектора $\boldsymbol{p(0)}$ по всем примерам выборки, затем составляющие полученного вектора упорядочить по убыванию и выделить 3 – 5 первых элементов.

Вся программа может быть разделена на две части. В первой части создается и обучается нейронная сеть, определяются ее архитектура и весовые коэффициенты. Во второй части рассматривается алгоритм подстройки входного сигнала (параметров заявки на кредит) для перевода заявки в заданный класс.

Для определения весов многослойной нейронной сети используются методы обратного распространения ошибки [1, 9, 10, 15, 18, 19], которые реализованы в различных процедурах библиотек *Keras, TensorFlow*.

Создание и обучение классификатора (НС). Настройка параметров заявки на кредит

Предполагается, что примеры выборки данных включают параметры заявки на кредит и класс клиента. В состав заявки должны входить характеристики кредитного продукта, заказанного клиентом, и финансово-экономические показатели кредитоспособности заемщика.

Программа решает задачу бинарной классификации заемщиков, а именно, по параметрам заявки на выдачу кредита относит клиента к одному из двух классов: 0 – «не допущен», 1 – «допущен к кредитованию». Таким образом, осуществляется принятие решения о выдаче кредита заемщику. Данные обучающей выборки содержатся в файле *data_train.csv*. Тестовые данные – в файле *data_unpredicted.csv*, а ответы к тестовым данным содержатся в файле *answers.csv*.

В число параметров заявки на кредит были включены такие данные: № *сделки, Сумма кредита* (*руб*), *Рейтинг заёмщика* (1-10 *баллов*), *Кредитная история, Результат верификации, Голос кредитного инспектора, Голос клиентского менеджера, Сумма залога* (*руб*), а также *Решение о выдаче кредита*. Среди них № сделки используется как ключевой для поиска нужной записи.

Вначале выполняются типичные шаги разработки нейронной сети: загрузка набора данных; предобработка набора данных (преобразование категориальных данных в числа, нормализация числовых данных и т.д.); выбор архитектуры и параметров НС; обучение НС; проверка качества нейронной сети на обучающей, валидационной и тестовой выборке.

В качестве классификатора была использована нейронная сеть, архитектура которой приведена в следующем фрагменте программы на языке Python.

```
model = Sequential()
#В первом слое задаем 100 нейронов и активационную функцию ReLU
model.add(Dense(100, input_dim = 10, activation="relu"))
#Во втором слое задаем 200 нейронов и активационную функцию ReLU
model.add(Dense(200, activation="relu"))
#В третьем слое задаем 400 нейронов и активационную функцию ReLU
model.add(Dense(400, activation="relu"))
#В последнем слое 1 нейрон с сигмоидальной активационной функцией
model.add(Dense(1, activation="sigmoid"))
# Компилируем модель
model.compile(loss="binary_crossentropy",
                          optimizer="adam", metrics=["accuracy"])
# Модель обучаем в течение 25 эпох
model_train = model.fit(x_train, y_train, batch_size=200,
                    epochs=25,  verbose=1, validation_split=0.2)
 # тестирование модели на тестовых данных с помощью метода evaluate
 test_eval = model.evaluate(x_test, y_test, verbose=0)
 # выводим значение функции потерь
 print('Test loss:', test_eval[0])
 # выводим значение точности
 print('Test accuracy:', test_eval[1])
```

На рисунке 1 приведены результаты тестирования обученной нейронной сети. Как видим, при тестировании функция потерь достаточно мала, а точность тестирования равна 0,985, что говорит о высоком качестве модели. Графики на обучающей и валидационной выборке практически сходятся, что свидетельствует об отсутствии переобучения.

Следующий этап – это эксплуатация нейронной сети на новых данных. На вход нейронной сети подаются параметры заявки на кредитование для нового клиента. Результат работы нейронной сети может иметь одно из двух значений (допущен (1) или не допущен (0) к кредитованию). Если на выходе получена 1, то кредит выдается, и классификатор заканчивает свою работу. Если на выходе 0, то программа переходит в режим поиска таких значений параметров заявки на кредитование, при которых клиент будет допущен к кредитованию (переведен в класс 1). При поиске желательно, чтобы новый вектор параметров заявки не только обеспечил переход клиента из класса 0 в класс 1, но и был как можно ближе к исходному вектору.

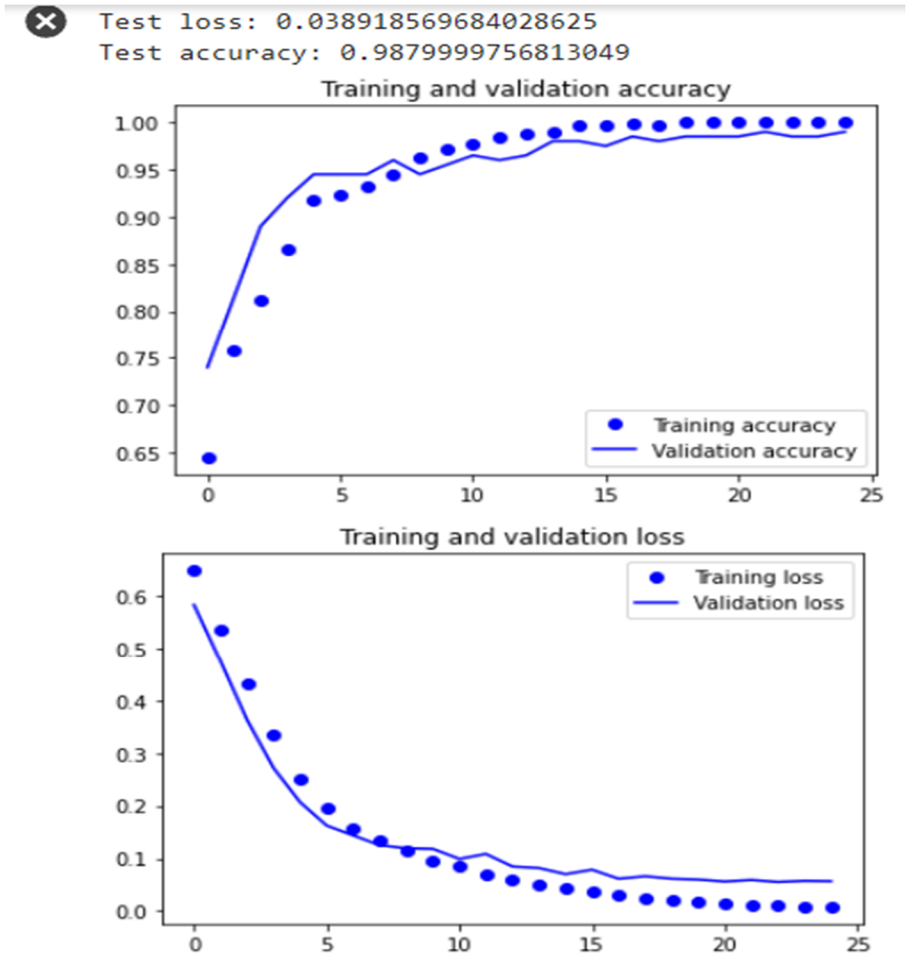

Рисунок 1 – Результаты тестирования обученной нейронной сети – классификатора
Figure 1 – Results of testing a trained neural network classifier

Сам поиск состоит из следующих шагов.

Шаг 1. Цель поиска – перевести клиента в заданный (в данном случае – противоположный) класс, сохраняя по возможности близость к исходной заявке, описываем с помощью следующей функции.

```
# Описываем функцию цели. Она имеет 2 составляющие:
# одна (l0) штрафует за ошибку (y_true - y_pred) в достижении цели,
# здесь y_true – целевой класс (1),
# y_pred – класс, полученный на выходе НС
# вторая(l1)  - за удаленность image от  input
# kda – коэффициент важности второй составляющей
# input – исходный вектор параметров заявки на кредит
# image – текущий искомый вектор параметров заявки на кредит
def loss_fun(input,image,y_true, y_pred):
    kda = 0.00001
    l0 = tf.reduce_mean(tf.square(y_true - y_pred), axis=-1)
    l1=kda*tf.reduce_mean(tf.square(image - input), axis=-1)
    loss = l0+l1
    return loss
```

Шаг 2. Затем начинается процесс подстройки параметров заявки (входного сигнала), описанный ниже с помощью функции `class1_v_class2 (input)`. В течение этого процесса сами весовые коэффициенты нейронной сети не меняются.

Шаг 3. Входной сигнал – исходная заявка клиента $\boldsymbol{x(0)}$ подается на вход нейронной сети. Нейронная сеть выполняет прямой ход. Сигнал на выходе $\boldsymbol{x(N)}$ сравнивается с заданным эталоном – классом 1, вычисляется ошибка, которая в соответствии с сопряженной системой (3) распространяется в обратном направлении и в результате позволяет найти $\boldsymbol{p(0)}$**.** Вектор $\boldsymbol{p(0)}$ показывает влияние входных признаков заявки на значение функции цели, т.е. представляет собой градиент функции цели по параметрам заявки на кредитование.

Шаг 4. В соответствии с методом градиентного спуска (функцию цели надо уменьшать) делается шаг небольшого размера по изменению параметров заявки в направлении, противоположном вектору $\boldsymbol{p(0)}$ (`optimizer = tf.keras.optimizers.Adam(learning_rate=0.1)`).

Шаг 5. Затем весь процесс повторяется до тех пор, пока параметры заявки на кредитование не обеспечат переход клиента в класс благонадежных (класс 1).

```
# Функция поиска значений параметров заявки на кредит (input),
# при которых клиент переходит из класса 0 в класс 1
# input – заявка на входе; image – на выходе преобразованная заявка
def class1_v_class2 (input):
    min_loss = 0.001
    image = tf.Variable(input) # Преобразуем вход в переменную tensor flow
    for iteration in range(400):
# Цикл по количеству шагов подстройки признаков объекта
# Откроем GradientTape, чтобы записать операции,
# выполняемые во время прямого прохода
        with tf.GradientTape() as tape:
      # Запустим прямой проход  по слоям.
      # Операции, применяемые слоями к своим
      # входным данным, будут записаны
      # на GradientTape.
            tape.watch(image)
            prediction = model(image, training=False)
# prediction - Выход НС при текущем значении входного сигнала
                    # Вычисляется функция цели
            loss_value = loss_fun(input, image, y_true, prediction)
# вычисление и нормализация градиентов функции цели по входному сигналу
        grads = tape.gradient(loss_value, image)
        grads /= tf.maximum(tf.reduce_mean(tf.abs(grads)), 1e-6)
        grads = np.array(grads)   # Преобразуем градиенты в массив
        optimizer = tf.keras.optimizers.Adam(learning_rate=0.1)
# На основе градиентов изменяем значения входного сигнала
        optimizer.apply_gradients(zip([grads], [image]))
        if  loss_value < min_loss:
#  Если функция цели меньше минимального значения, то выходим из цикла
    return loss_value, image
```

Полученный результат может быть интерпретирован следующим образом.

Новые значения (*image*) признаков заявки, относящиеся к параметрам кредитного продукта, должны быть согласованы с клиентом как новый кредитный продукт.

А новые значения (*image*) параметров, относящихся к кредитоспособности клиента, должны быть рекомендованы клиенту как способ работы над своим финансово-экономическим положением. Например, улучшить кредитную историю, увеличить доход, увеличить сумму залога и т.п.

Экспериментальные исследования программы

Программа написана на языке Python с использованием библиотек *Keras* и *TensorFlow*. Экспериментальные исследования программы проводились в среде *Google Colab*.

Разработанная программа выполняет следующие функции:

– создает классификатор в виде нейронной сети, обученной на наборе данных по кредитованию заемщиков;

– оценивает качество классификатора на тестовой выборке;

– может использоваться как классификатор при рассмотрении новых заявок на кредитование;

– позволяет получить оценку степени влияния каждого параметра заявки на достижение цели «получить кредит»;

– предоставляет возможность для не допущенной к кредитованию заявки подобрать такие значения параметров, при которых она переходит в разряд допущенных к кредитованию;

– полученные в результате поиска значения заявки на кредит позволяют дать рекомендации клиенту как по параметрам кредитного продукта, так и по характеристикам самого заемщика.

Сначала познакомимся и исследуем набор данных по кредитованию заемщиков, любезно предоставленный одним из банков России. Это знакомство покажет, в чем различие положительных и отрицательных заявок в наборе, к каким значениям параметров надо стремиться, чтобы получить кредит. Набор данных содержит 1000 примеров для обучения и 1000 примеров для тестирования. Данные на рисунках 2 – 7 являются скриншотами результатов, полученных в процессе выполнения программы в среде *Google Colab*.

	№ сделки	Сумма кредита (руб)	Рейтинг заёмщика (1-10 баллов)	Кредитная история	Результат верификации	Голос кредитного инспектора	Голос клиентского менеджера	Сумма залога (руб)	Решение
0	1	180000	7	NaN	отрицательный	NaN	NaN	293000.0	Отклонен
1	2	290000	5	зеленая	положительный	NaN	NaN	nan	Одобрен
2	3	420000	8	NaN	положительный	NaN	NaN	507000.0	Одобрен
3	4	900000	9	зеленая	отрицательный	NaN	NaN	nan	Отклонен
4	5	60000	5	жёлтая	отрицательный	NaN	NaN	nan	Отклонён

Рисунок 2 – Пять первых записей обучающего набора данных
Figure 2 – First five records of training dataset

Рисунок 2 знакомит с 5-ю первыми записями обучающего набора. Видно, что среди признаков есть как категориальные данные, так и числовые. Для обучения нейронной сети все параметры были преобразованы в числовую форму и нормализованы. На рисунке 3 представлен набор данных после предобработки, именно на нем обучалась нейронная сеть.

	DealID	CreditSumm	Rating	CredHistory	Verification	VoteCI	VoteCM	PledgeSumm	Decision	I
0	1	180000	7	0	0	0.0	0.0	293000.0	0	
1	2	290000	5	1	1	0.0	0.0	0.0	1	
2	3	420000	8	0	1	0.0	0.0	507000.0	1	
3	4	900000	9	1	0	0.0	0.0	0.0	0	
4	5	60000	5	2	0	0.0	0.0	0.0	0	

Рисунок 3 – Примеры из обучающего набора данных после предобработки
Figure 3 – Examples from training dataset after preprocessing

На рисунках 4 и 5 приведены результаты исследования [с помощью процедуры describe()] отрицательных (класс 0) и положительных примеров (класс 1) обучающей выборки.

В результате исследования получаем следующие характеристики поднабора: *count* – число примеров; *mean, std, min, max* – соответственно среднее значение, среднеквадратичное отклонение, минимальное и максимальное значение в подвыборке. Анализ средних значений параметров заявки показывает, что значения признаков: *Рейтинг заёмщика, Кредитная ис-*

тория, Результат верификации, Голос кредитного инспектора, Голос клиентского менеджера, Сумма залога больше у положительных примеров, чем у отрицательных.

```
row0.describe()
```

	DealID	CreditSumm	Rating	CredHistory	Verification	VoteCI	VoteCM	PledgeSumm	Decision	IsPledgeExists	IsVoteExists
count	544.0	544.0	544.0	544.0	544.0	544.0	544.0	544.0	544.0	544.0	544.0
mean	484.7	529614.0	6.0	1.2	0.5	0.2	0.2	283189.3	0.0	0.4	0.5
std	286.3	288652.2	2.7	1.0	0.5	0.4	0.4	432694.8	0.0	0.5	0.5
min	1.0	50000.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	233.8	280000.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	477.0	545000.0	6.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	717.8	790000.0	8.0	2.0	1.0	0.0	0.0	452250.0	0.0	1.0	1.0
max	1000.0	990000.0	10.0	3.0	1.0	1.0	1.0	1481000.0	0.0	1.0	1.0

Рисунок 4 – Результаты исследования множества отрицательных примеров выборки
Figure 4 – Results of negative sample examples set study

```
row1.describe()
```

	DealID	CreditSumm	Rating	CredHistory	Verification	VoteCI	VoteCM	PledgeSumm	Decision	IsPledgeExists	IsVoteExists
count	456.0	456.0	456.0	456.0	456.0	456.0	456.0	456.0	456.0	456.0	456.0
mean	519.3	490241.2	7.4	1.0	1.0	0.3	0.3	474557.0	1.0	0.6	0.5
std	291.0	266719.1	1.8	0.8	0.0	0.5	0.4	511046.4	0.0	0.5	0.5
min	2.0	50000.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
25%	270.8	270000.0	6.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
50%	532.5	480000.0	8.0	1.0	1.0	0.0	0.0	297000.0	1.0	1.0	0.0
75%	777.2	710000.0	9.0	2.0	1.0	1.0	1.0	927750.0	1.0	1.0	1.0
max	999.0	990000.0	10.0	2.0	1.0	1.0	1.0	1491000.0	1.0	1.0	1.0

Рисунок 5 – Результаты исследования множества положительных примеров выборки
Figure 5 – Results of many positive sample examples set study

Затем для обученного классификатора оценим степень влияния каждого из параметров заявки на функцию цели. В результате применения предложенной в статье процедуры был получен следующий результат:

```
Среднее значение градиента по входам всех примеров выборки  =
   CrSumm  Rat  CredHi  Ver  VCI  VCM  PlSumm  PledEx  VoteEx  PledMoreCr
0     0.1  0.6     0.3  1.3  0.5  0.1     0.0     0.2     0.6         0.4
 максимальная чувств. =  1.2797481897882694
 номер признака с наибольшим влиянием на функцию цели= 3   Имя признака    Ver
```

Из результатов видно, что все параметры по степени влияния на достижение цели можно упорядочить по убыванию следующим образом:

Верификация (*Ver*=1.3); Рейтинг заемщика (*Rat*=0.6); Есть голос кредитного инспектора или менеджера (*VoteEx* = 0.6); Голос кредитного инспектора (*VCI*=0.5); Залог больше суммы кредита (*PledMoreCr* = 0.4); Кредитная история (*CredHi* = 0.3); Есть залог (*PledEx* = 0.2); Голос кредитного менеджера (*VCM* = 0.1)

И наконец, рассмотрим результаты применения процедуры подстройки параметров заявки на кредит с целью перевода заявки из класса 0 (не допущен) в класс 1 (допущен).

Эксперименты проводились при следующих условиях. Рассматривались только те заявки, которые классификатор отнес к классу 0. Поиск новых значений параметров осуществлялся на основе градиентного спуска (использовался оптимизатор *Adam*). Число итераций было ограничено и равнялось 400. Выход из цикла происходил, как только квадрат ошибки на выходе нейронной сети становился меньше заданного значения 0,001.

Рассматривались 2 случая. В *первом случае* на каждой итерации изменялись все параметры заявки в соответствии с их градиентом. Были получены следующие результаты на всей выборке.

```
Sr= [0.00013483]
ktr= 372    knom= 567    acc= 0.656084656084656
```

Здесь *Sr* – квадрат ошибки на выходе НС; *knom* = 567 – число примеров класса 0; из них *ktr* = 372 удалось перевести из класса 0 в класс 1, что соответствует 65 % от *knom acc* = 0,65608. Ниже на рисунке 6 приведены примеры результатов преобразования параметров заявки. В первой строке каждого примера находятся параметры заявки класса 0, а во второй параметры той же заявки, преобразованные к классу 1.

CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
350000.0	1.0	0.0	1.0	0.0	0.0	569000.0	1.0	1.0	1.0
516878.1	3.0	-0.0	1.3	-0.3	-0.1	569000.4	1.4	0.7	1.4
CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
310000.0	1.0	2.0	1.0	0.0	1.0	603000.0	1.0	1.0	1.0
560331.3	3.2	1.2	1.4	0.2	1.4	267721.9	0.6	0.6	1.4
CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
220000.0	9.0	2.0	1.0	0.0	1.0	45000.0	1.0	1.0	0.0
108741.6	10.0	1.6	1.2	0.2	1.0	45003.3	1.1	0.8	0.2
CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
400000.0	1.0	3.0	1.0	1.0	0.0	724000.0	1.0	1.0	1.0
566888.6	3.5	2.1	1.4	0.8	-0.2	1202981.5	1.5	1.0	0.5
CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
310000.0	8.0	0.0	0.0	0.0	0.0	943000.0	1.0	0.0	1.0
226558.4	9.7	0.1	0.3	0.1	-0.3	607713.2	1.3	-0.2	1.3

Рисунок 6 – Примеры преобразования параметров заявки для перехода ее из класса 0 в класс 1 в случае изменения всех параметров
Figure 6 – Examples of application parameters conversion for its transition from Class 0 to Class 1 in case of all parameters change

Во *втором случае* ряд параметров исключался из процесса поиска, соответствующие этим параметрам градиенты обнулялись:

```
grads[0,3:6] = 0
grads[0,7:9]=0
```

В этом случае были получены следующие результаты. На всей выборке:

```
Sr= [0.00042428]
ktr= 338    knom= 567    acc= 0.5961199294532628
```

и примеры результатов преобразования параметров заявки (рисунок 7).

Из анализа полученных результатов можно сделать следующие выводы.

– Предложенная процедура подбора параметров заявки на кредит работает: 60 % примеров из класса 0 удается перевести в класс 1. Все примеры перевести не удается, потому что, во-первых, многие из них находятся далеко по своим характеристикам от класса 1, во-вторых, надо получать заявку, более или менее близкую к исходной; в-третьих, заданы жесткие условия поиска. Можно поменять число итераций, изменить критерий выхода из цикла, но это приведет к получению результатов, реализация которых в предметной области будет нецелесообразна.

– Предложенная в работе процедура оценки влияния параметров заявки (входного сигнала нейронной сети) на функцию цели также показала свою полезность. Благодаря ей можно выделить параметры, на которые нужно обратить внимание в первую очередь. Кроме того, ее можно использовать для сокращения числа признаков при решении задачи классификации, исключив из рассмотрения параметры с незначительным влиянием.

– Еще одним достоинством предложенного алгоритма является возможность менять в процессе поиска только заданные параметры.

CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
850000.0	8.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
1155951.5	8.7	0.8	1.0	0.0	0.0	1101639.0	-0.0	1.0	1.1

CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
800000.0	5.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0
1022515.6	7.0	0.8	1.0	1.0	0.0	-4.1	-0.0	1.0	0.3

CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
950000.0	10.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
1311624.5	8.8	0.8	1.0	0.0	0.0	1101656.6	-0.0	1.0	1.1

CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
300000.0	4.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
160948.1	5.2	1.5	1.0	0.0	0.0	-239491.9	-0.0	-0.0	0.2

CrSumm	Rat	CredHi	Ver	VCI	VCM	PlSumm	PledEx	VoteEx	PledMoreCr
570000.0	4.0	2.0	1.0	0.0	1.0	1156000.0	1.0	1.0	1.0
681258.4	5.0	1.6	1.0	0.0	1.0	1347592.1	1.0	1.0	1.2

Рисунок 7 – Примеры преобразования параметров заявки для перехода ее из класса 0 в класс 1 в случае изменения только отдельных параметров
Figure 7 – Examples of application parameters conversion for its transition from Class 0 to Class 1 in case of only individual parameters change

Заключение

Основные результаты работы сводятся к следующему.

1. Для принятия решения о выдаче кредита заемщику создан и обучен классификатор в виде нейронной сети (НС). Выполнена оценка его качества. Ошибка и точность (accuracy) на тестовой выборке составили соответственно 3 % и 98 %.

2. На вход классификатора подаются параметры заявки на кредитование, среди которых есть как характеристики заемщика, так и параметры приобретаемого кредитного продукта. Сигнал на выходе НС сообщает о принадлежности заемщика (его заявки) к одному из классов: 1 – «допущен», 0 – «не допущен» к кредитованию.

3. В работе задача обучения НС представлена как задача оптимального управления дискретным процессом. Такая постановка задачи позволила показать, что сопряженный вектор $\boldsymbol{p(0)}$ – вектор чувствительности функции цели к изменениям входного сигнала НС (параметрам заявки на кредит). Этот вектор $\boldsymbol{p(0)}$ является градиентом функции цели по входному сигналу, что позволяет использовать его при поиске параметров заявки на кредит методом градиентного спуска, а также во всех случаях, когда надо направленно воздействовать на входной сигнал НС ради достижения поставленной цели.

4. Разработаны алгоритм и программа на языке Python для поиска параметров заявки на кредитование, обеспечивающих переход заявки из класса 0 в класс 1. Результаты исследования алгоритма показали, что 60 % всех заявок класса 0 были преобразованы в класс 1. Это неплохой результат для практического применения алгоритма.

5. Разработана и описана на языке Python процедура оценки степени влияния каждого из параметров заявки на кредит (входного сигнала НС) на функцию цели. С помощью этой процедуры можно построить рейтинг параметров по степени влияния на достижение поставленной цели.

6. Применение предложенной в работе методики можно расширить на клиенториентированные области бизнеса, т.е. на те области, где на основе заявки клиента принимается решение о возможности (Да/Нет) реализовать запросы клиента при его социальных и финансово-экономических показателях.

Библиографический список

1. **Шитова К. Г., Цуканова Н. И.** О применении глубоких нейронных сетей к оценке кредитоспособности предприятия // Вестник Рязанского государственного радиотехнического университета. 2019. № 68. С. 44-55.

2. О типичных банковских рисках: письмо Банка России от 23.06.2004 № 70-Т. // Вестник Банка России. 2004. № 38.

3. **Кабушкин С. Н.** Управление банковским кредитным риском: учеб. пособие. М.: Новое знание, 2008.

4. **Сахарова М. О.** К вопросу о кредитоспособности предприятия // Деньги и кредит. № 3.

5. **Паклин Н. Б., Орешков В. И.** Бизнес-аналитика: от данных к знаниям (+CD): учеб. пособие. 2-е изд., испр. СПб.: Питер, 2013. 704 с.: ил.

6. **Брюхнова В. О., ЦукановаН. И.** Ансамбли нейронных сетей при прогнозировании объемов продаж в торговой сети. // Вестник Рязанского государственного радиотехнического университета. 2018. № 66-1. С. 90-99.

7. **Гиголаев А. В., Цуканова Н. И.** Анализ семантической близости слов с помощью карт Кохонена. // Вестник Рязанского государственного радиотехнического университета. 2018. № 64. С. 85-92.

8. **Гудфеллоу Я., Бенджио И., Курвилль А.** Глубокое обучение / пер. с анг. А.А. Слинкина. 2-е изд. испр. М.: ДМК Пресс, 2018. 652 с.

9. **Николенко С., Кадурин А., Архангельская Е.** Глубокое обучение. СПб.: Питер, 2018. 480 с.

10. **Michael Nielsen**. Neuralnetworksanddeeplearning.com.

11. Basel Committee on Banking Supervision. «International Convergence of Capital Measurement and Capital Standards: A Revised Framework». Bank for International Settlements (BIS), comprehensive version, June 2006. Available at: http://www.bis.org/publ/bcbsca.htm.

12. **Altman E.** «Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy». Journal of Finance, vol. 23, no. 4, (Sep. 1968), pp. 589-609.

13. **Loeffler G. and P. N. Posch.** Credit Risk Modeling Using Excel and VBA, West Sussex, England: Wiley Finance, 2007.

14. **Пропой А. И.** Элементы теории оптимальных дискретных процессов. М.: Наука. Главная редакция физико-математической литературы, 1973. 254 с.

15. **Tieleman Tijmen and Geoffrey Hinton.** Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning 4 (2012):2.

16. **Duchi, John, EladHazan, and Yoram Singer**. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research 12 (2011): 2121-2159.

17. **Sutskever Ilya et al.** On the importance of initialization and momentum in deep learning. Proceeding sof the 30thinternational conference on machine learning (ICML-13). 2013.

18. **Diederik P., Kingma, Jimmy LeiBa.** Adam: A Method for Stochastic Optimization. Published as a conferens Paper at ICLR 2015. https://arxiv.org/pdf/1412.6980v8.pdf.

19. **Ruder S.** An Overview of Gradient Descent Optimization Algorithms//arXiv.org/abs/1609.04747.

UDC 004.032.26

SEARCH FOR ACCEPTABLE VALUES OF APPLICATION PARAMETERS FOR LENDING USING A NEURAL NETWORK

N. I. Tsukanova, Ph.D. (Tech.), associate professor of the Department of Computational and Applied Mathematic, RSREU, Ryazan, Russia;
orcid.org/0000-0001-7337-8037, e-mail: ninakorobova77@gmail.com
K. G. Shitova, leading development engineer, Sberbank PJSC; Ryazan, Russia;
orcid.org/0000-0002-7809-564X, e-mail: kshitova@inbox.ru

The issues of selecting the parameters of client application for lending in order to transfer the borrower from the class "denied lending" to the class "loan issuance is allowed" are considered. ***The aim of the work*** *is to improve the quality of client's creditworthiness assessment procedure, aimed at increasing the number of borrowers by agreeing with them and/or giving them recommendations on changing loan application parameters. To achieve this aim, the following tasks were solved in the work. A deep neural network has been developed in Python, solving the problem of binary classification of clients by their characteristics into two classes – (admitted, not admitted) to lending. It is shown that the disadvantage of such a procedure is a large dropout of customers who could get a loan and bring the income to bank, but on other stricter conditions. Methodology and algorithm of searching loan application parameters values, allowing the client to be admitted to lending, are proposed.*

Keywords*: client creditworthiness assessment, loan application, deep fully connected neural networks, Python language, Keras library, error back propagation algorithm, conjugate vector, target (loss) function.*

DOI: 10.21667/1995-4565-2021-78-89-101

References

1. **Shitova K. G., Tsukanova N. I.** O primenenii glubokih nejronnyh setej k ocenke kreditosposobnosti predprijatija. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2019, no. 68, pp. 44-55. (in Russian).

2. O tipichnyh bankovskih riskah: pis'mo Banka Rossii ot 23.06.2004 no. 70-T. *Vestnik Banka Rossii*. 2004, no. 38. (in Russian).

3. **Kabushkin S. N.** *Upravlenie bankovskim kreditnym riskom: ucheb. posobie.* Moscow: Novoe znanie, 2008. (in Russian).

4. **Saharova M. O.** K voprosu o kreditosposobnosti predprijatija. *Den'gi i kredit*, no. 3. (in Russian).

5. **Paklin N. B., Oreshkov V. I.** *Biznes-analitika: ot dannyh k znanijam (+SD): ucheb. posobie*. 2-e izd., ispr. SPb.: Piter, 2013. 704 p.: il. (in Russian).

6. **BrjuhnovaV. O., Tsukanova N. I.** Ansambli nejronnyh setej pri prognozirovanii ob"emov prodazh v torgovoj seti. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2018, no. 66.1, pp. 90-99. (in Russian).

7. **Gigolaev A. V., Tsukanova N. I.** Analiz semanticheskoj blizosti slov s pomoshh'ju kart Kohonena. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2018, no. 64, pp. 85-92. (in Russian).

8. **Gudfellou Ja., Bendzhio I., Kurvill' A.** *Glubokoe obuchenie / per. s ang. A. A. Slinkina*. 2-e izd. ispr. M.: DMK Press, 2018. 652 p. (in Russian).

9. **Nikolenko S., Kadurin A., Arhangel'skaja E.** *Glubokoe obuchenie*. SPb.: Piter, 2018. 480 p. (in Russian).

10. **Michael Nielsen**. Neuralnetworksanddeeplearning.com.

11. Basel Committee on Banking Supervision. «International Convergence of Capital Measurement and Capital Standards: A Revised Framework». Bank for International Settlements (BIS), comprehensive version, June 2006. Available at: http://www.bis.org/publ/bcbsca.htm.

12. **Altman E.** «Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy». *Journal of Finance*, vol. 23, no. 4, (Sep. 1968), pp. 589-609.

13. **Loeffler G. and P. N. Posch.** Credit Risk Modeling Using Excel and VBA, West Sussex, *England: Wiley Finance*, 2007.

14. **Propoj A. I.** *Jelementy teorii optimal'nyh diskretnyh processov*. Moscow: Nauka, Glavnaja redakcija fiziko-matematicheskoj literatury, 1973. 254 p. (in Russian).

15. **Tieleman Tijmen and Geoffrey Hinton.** Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning* 4 (2012):2.

16. **Duchi, John, EladHazan, and Yoram Singer**. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research* 12 (2011): 2121-2159.

17. **Sutskever Ilya et al.** On the importance of initialization and momentum in deep learning. *Proceeding sof the 30th international conference on machine learning (ICML-13)*. 2013.

18. **Diederik P., Kingma, Jimmy LeiBa.** Adam: A Method for Stochastic Optimization. Published as a conferens Paper at ICLR 2015. https://arxiv.org/pdf/1412.6980v8.pdf.

19. **Ruder S.** An Overview of Gradient Descent Optimization Algorithms//arXiv.org/abs/1609.04747.