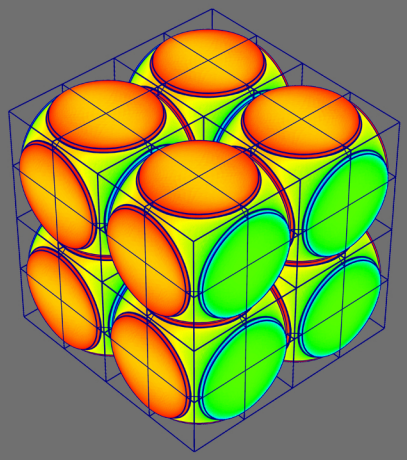
ISSN 2309-3684



Математическое моделирование и численные методы

Кадиев А.Д., Чибисова А.В. Нейросетевые методы решения задачи кредитного скоринга. Математическое моделирование и численные методы, 2022, № 4, с. 81–92.

Источник: https://mmcm.bmstu.ru/articles/295/

Параметры загрузки:
IP: 216.73.217.36
06.11.2026 22:13:16

УДК 004.855.5 DOI: 10.18698/2309-3684-2022-4-8192

Нейросетевые методы решения задачи кредитного скоринга

© А.Д. Кадиев, А.В. Чибисова

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Продемонстрирован математический вывод представленной модели нейронной сети. Сведение задачи классификации к задаче оптимизации. Произведен разведывательный анализ данных, а также их предобработка для дальнейшего использования в обучении алгоритмов классификации. Были спроектированы архитектуры нейронных сетей, зависящих от функции активации, количества скрытых слоев нейронной сети и количества нейронов в скрытых слоях. Обучено более десяти нейронных сетей, решающих поставленную задачу кредитного скоринга. Произведен расчет времени обучения нейронных сетей. Представлено решение задачи при помощи классических алгоритмов машинного обучения. Можно было заметить, что стандартное отклонение accuracy и ROC AUC для нейронных сетей больше, чем у случайного леса. Это происходит из-за того, что мы выбираем начальные веса случайным образом и градиенты считаем не на всей выборке, а на малых частях, что добавляет некоторую погрешность при обучении. Но эти отклонения были не только в худшую сторону. В лучших ситуациях, по обеим метрикам, нейронные сети показывали результат хуже всего на пару процентов. Произведен анализ результатов. Сравнительный анализ показывает, что нейронные сети имеют лучшее качество классификации, чем классические алгоритмы машинного обучения, а также, что нейронные сети имеют меньшее время обучения, чем классические алгоритмы машинного обучения. Представлены графики и таблицы, отображающие имеемые результаты.

Ключевые слова: *нейронные сети, машинное обучение, случайный лес, логистическая регрессия, градиентный бустинг*

Введение. Кредитный скоринг — система оценки кредитоспособности лица, основанная на численных статистических методах. Кредитный скоринг широко используется как крупными банками, финансовыми организациями, так и в потребительском (магазинном) экспресс-кредитовании на небольшие суммы[1].

Задача кредитного скоринга возникла в связи с необходимостью уменьшения рисков, которым может подвергнуть себя организация, а также вследствие имения большого количества данных людей, ежедневно берущих в долг денежные средства для приобретения товаров или услуг.

Одним из основных рисков является невозврат заемщиком суммы кредита в полном объеме или в указанный срок, т.е. нарушение обязательств. Оценка кредитных рисков потенциального заемщика называется кредитным скорингом (от англ. credit scoring). Говоря о кредитном скоринге, как правило, имеют в виду анализ

рисков по кредитованию физических лиц, хотя методы оценки надежности организаций также существуют.

Кредитный скоринг состоит в применении алгоритмов, полученных с использованием математических и статистических методов, с тем, чтобы разделить потенциальные кредитные операции на непересекающиеся группы риска, хорошие и плохие [2]. Плохие риски подразумевают бóльшую вероятность нарушения обязательств заемщиком, поэтому необходимо выявлять факторы кредитного риска, их значимость и взаимозависимость. Предполагается, что созданные модели могут выявлять закономерности, так что кредитные операции в будущем будут иметь такой же исход, как и операции со схожими характеристиками, для которых известна принадлежность к одному из рисков.

Факторы, учитываемые при кредитном скоринге могут отличаться в зависимости от алгоритмов и целей скоринга. К типично используемым факторам можно отнести демографические данные (семейное положение, возраст и др.) и характеристики занятости заемщика (тип занятости, должность и др.), информацию о кредитной истории и предыдущих отношениях с кредитором, характеристики предоставляемой услуги, данные о финансовом благополучии клиента [3].

Целью данной работы является:

- создание оптимальной модели классификации заемщика.

Задачами данной работы являются:

- обработка и фильтрация исходных данных;
- разработка модели кредитного скоринга с использованием методов искусственных нейронных сетей;
- оценка качества прогнозирования моделей;
- сравнение итоговой модели с иными моделями.

Работа была выполнена на языке программирования python с использованием фреймворка Pytorch.

Математическая постановка задачи, принятые допущения. Математическую постановку задачи можно записать следующим образом: пусть X — множество описаний объектов (людей), Y — множество меток классов (ответов на вопрос: был ли выплачен кредит). Существует неизвестная целевая зависимость:

$$y^* : X \to Y, \tag{1}$$

значения которой известны на объектах обучающей выборки.

Требуется построить алгоритм:

$$a : X \to Y, \tag{2}$$

способный классифицировать произвольный объект $x \in X$.

Математический вывод модели нейронной сети. Покажем простейшую архитектуру нейронной сети. Пусть задана однослойная нейронная сеть (нейрон). Используя теорему Колмогорова об аппроксимации всякой непрерывной функции, заданной на единичном d – кубе [4], можно показать, что такая нейронная сеть будет определяться выражением:

$$a(x) = \sum_{i=1}^{2d+1} \sigma_i \left(\sum_{j=1}^{d} f_{ij}(x_j) \right), \tag{3}$$

где $x = [x_1, \ldots, x_d]^T$ — вектор описания объекта; σ_i, f_{ij} — некоторые непрерывные функции, при этом f_{ij} не зависят от a.

Тогда для любой непрерывной функции $f(x)$ найдется нейронная сеть $a(x)$ с линейным выходом [5]:

$$a(x, W) = \sigma^{(M)}(x), \tag{4}$$

$$f^{(k)}(x) = \omega_0^{(k)} + W^{(k)} z^{(k-1)}(x), \tag{5}$$

$$z^{(k)}(x) = \sigma^{(k)}\left(f^{(k-1)}(x) \right), \tag{6}$$

аппроксимирующая $f(x)$ с заданной точностью [6].

Пусть имеется несколько видов архитектур нейронной сети для поставленной задачи: архитектура с убыванием размерности в скрытых слоях, архитектура с одинаковой размерностью и архитектура с увеличением размерности [7].

На первом шаге мы располагаем признаками количеством n, тогда выход линейной части нейронной сети на первом скрытом слое будет записан по формуле (7):

$$v_k^1(\vec{x}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(x_i w_j^i \right) + b^k), \tag{7}$$

где $v_j^1(\vec{x})$ — вектор результирующих значений в скрытом слое; x_i — признак i заемщика; w_j^i — вес j, данный соответствующим нейроном признаку i; b^k — смещение, данное соответствующим нейроном признаку k.

Далее к (7) применим некоторую функцию активации, определяемую в каждом слое соответственно. Тогда результат нелинейной части нейронной сети в первом скрытом слое будет определяться по формуле (8):

$$h_1(\vec{x}) = f_1(v_k^1(\vec{x})). \tag{8}$$

Далее действуем аналогично:

$$v_k^2\left(\vec{x}\right)=\sum_{i=1}^{m}\sum_{j=1}^{m-1}\left(w_i^j f_1\left(v_j^1\left(\vec{x}\right)\right)\right)+b^k, \tag{9}$$

$$h_2\left(\vec{x}\right)=f_2\left(v_k^2\left(\vec{x}\right)\right). \tag{10}$$

Производя аналогичные действия, в конечном итоге можем записать результирующую модель нейронной сети в формуле (9):

$$\begin{cases}\tilde{y}=f_n\left(v^n\left(w,b,\vec{x}\right)\right);\\ L\left(y,\tilde{y}\right)=\dfrac{1}{l}\sum\limits_{k=1}^{l}\left(\tilde{y}\ln\left(y_i\right)+\left(1-y_i\right)\ln\left(1-y_i\right)\right);\\ \left(\overrightarrow{w_*},\overrightarrow{b_*}\right)=\operatorname{argmin}_{w,b}\left(L\left(y,\tilde{y}\right)\right);\\ P\left(\vec{x}\right)=\dfrac{U_*}{U}\to\max;\\ S_{ROC}\to\max.\end{cases} \tag{11}$$

В модели (9) также добавили необходимость увеличение площади под ROC-кривой [8]. Площадь под кривой будет увеличиваться, чем лучше модель предсказывает выплату кредита. Во время обучения данной модели будем пользоваться градиентными методами второго порядка, а именно метод, именуемый BFGS — вследствие необходимости расчета матрицы Гессе данный метод позволит учитывать при спуске кривизну кривой, что скажется, в свою очередь, на времени работы алгоритма и на качестве работы алгоритма.

Обучение классических методов машинного обучения. Обучим модель, используя метод логистической регрессии [9]. Для выбора наилучшей модели логистической регрессии, также будем варьировать количество итераций для обучения модели. Обучение алгоритма заняло 40 с.

Необходимо рассмотреть метрику accuracy [10] на тестовых и тренировочных данных:

$$\text{accuracy}_{test}=0,8595, \tag{12}$$

$$\text{accuracy}_{train}=0,8592. \tag{13}$$

Теперь рассмотрим основную метрику — площадь под ROC-кривой (на график были выведены три наиболее эффективные модели) (рис. 1).

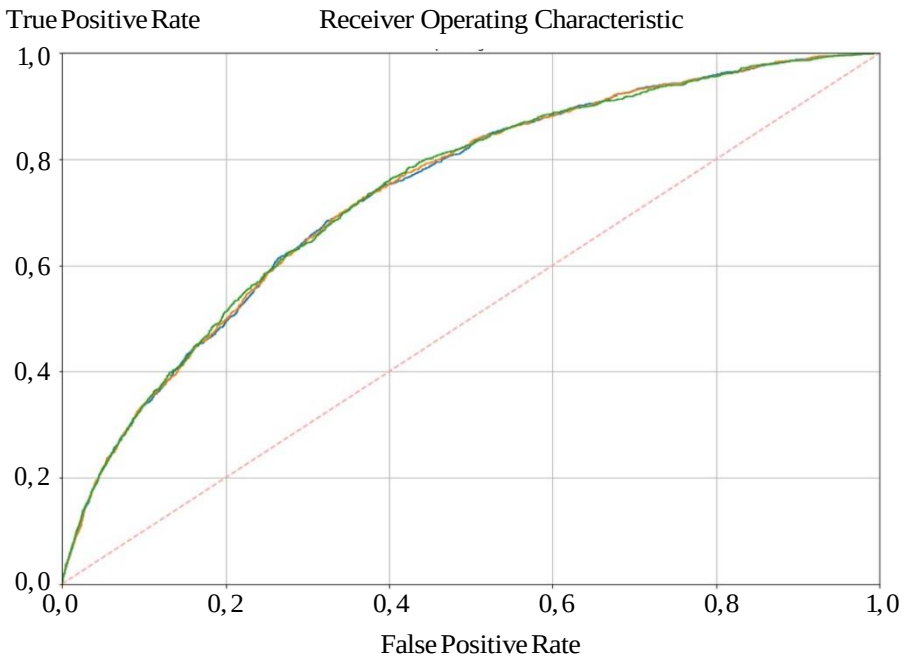


Рис. 1. Результаты предсказаний логистической регрессии.
Основной задачей является увеличение площади под кривой ROC.
Как видно, чем больше шагов (iterations) сделано
в признаковом пространстве,
тем большую площадь под кривой имеет данная модель:
— 10 шагов = 0,73633 ; — 11 шагов = 0,73636 ;
— 12 шагов = 0,73775

Теперь обучим модель, используя метод решающих деревьев [11]. Прежде всего, обучим всего лишь одно решающее дерево. С помощью одного решающего дерева уже можно строить прогнозы относительно клиентов банка.

Необходимо рассмотреть метрику accuracy на тестовых и тренировочных данных:

$$\text{accuracy}_{test} = 0,8598, \tag{14}$$

$$\text{accuracy}_{train} = 0,8585. \tag{15}$$

Продолжим рассмотрение качества алгоритма и построим метрику ROC на рис. 2.

Теперь рассмотрим решающий лес и градиентный бустинг над деревьями [12]. В результате, на рис. 3 получаем следующий график зависимости качества от количества деревьев.

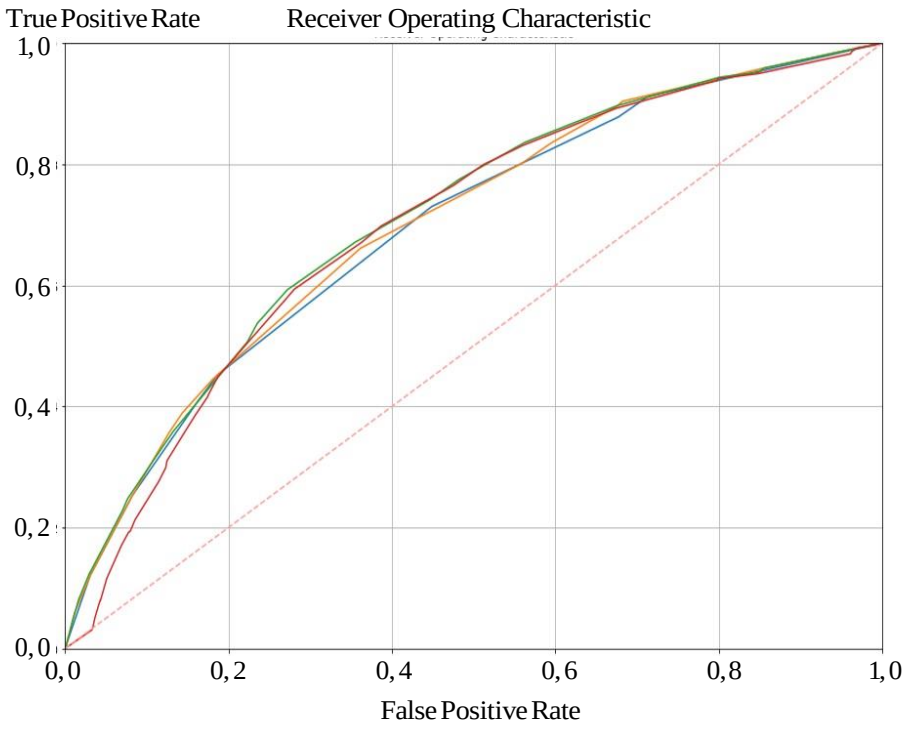


Рис. 2. Результаты предсказаний случайного дерева. Стоит обратить внимание на алгоритм, обученный на деревьях глубины «6». Как мы видим, данный алгоритм допускает случайный выбор ответа для классификации заемщика, поскольку ROC-кривая в окрестности точки (0,0) строится по прямой:
— глубина = 3 : 0,69345 ; — глубина = 4 : 0,69954 ;
— глубина = 5 : 0,70939 ; — глубина = 6 : 0,69783

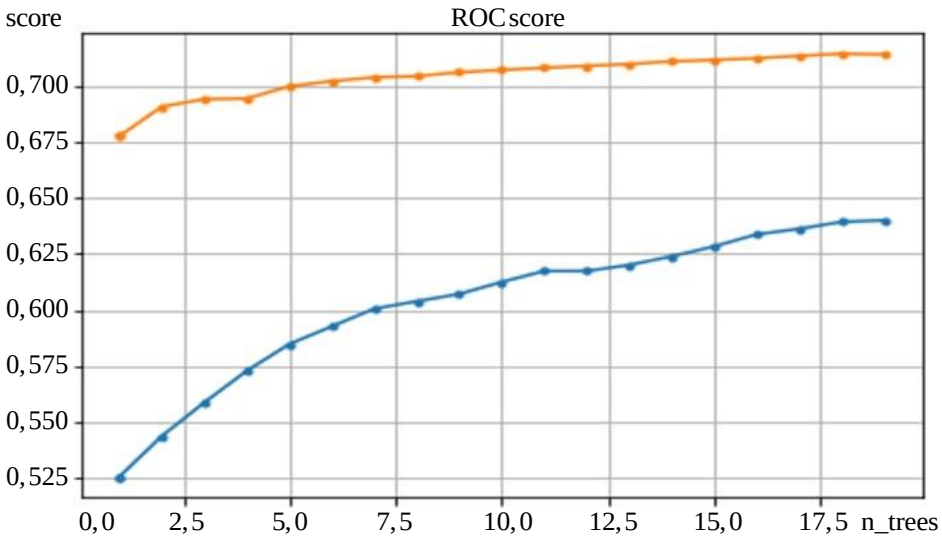


Рис. 3. Результаты предсказаний случайного леса и градиентного бустинга. Как можно было предположить, качество композиционных алгоритмов растет с увеличением количества деревьев. Для обучения нами будет взято 100 деревьев:
— RandomForest; — XGBoost;

Обучение нейронных сетей. При построении нейронной сети необходимо определить ее параметры (скорость обучения сети, количество эпох обучения, размер батчи (batch), значение параметра регуляризации), выбрать количество скрытых слоев и количество нейронов в них, произвести настройку весов нейронов.

Построение нейросетевых алгоритмов не предполагает четких требований при выборе конфигурации и архитектуры сети, зачастую выбор конкретной структуры и настройка параметров осуществляются на основании многочисленных тестов, в результате которых делается оптимальный выбор.

В соответствии с (11) обучим восемнадцать архитектур нейронных сетей. Для решения задачи классификации был выбран многослойный персептрон [13] (multilayer perceptron, MLP).

В качестве активационных функций на выходном слое была использована логистическая функция, которая на выходе позволяет получить вектор с вероятностями принадлежности записи к конкретному классу.

Скорость обучения нейронной сети может задаваться либо константой, либо убывающей функцией от количества эпох обучения [14]. Задание скорости обучения с помощью убывающей функции, например, линейной, позволяет подстраивать скорость обучения. В данной работе скорость обучения была задана константой, равной $2e-3$, так как такая скорость обучения считается эталонной для выбранной архитектуры сети. Она обеспечивает высокую точность настройки весов.

В соответствии с (11), обучим восемнвдцать нейросетей:

- для сигмоидной функции активации — три однослойных нейросети, две двухслойных, четыре трехслойных,
- для функции активации гиперболический тангенс — три однослойных нейросети, две двухслойных, четыре трехслойных.

Примем следующие обозначения $\mathrm{NNXP}(k)$. X — количество слоев в нейронной сети, P — используемая функция активации (S — сигмоидная, TS — тангенс гиперболический). В скобках отображено количество нейронов в каждом скрытом слое. Результаты представим в табл. 1, имеющую следующую структуру: нейросеть (название нейросети по приведенной выше терминологии), доля правильных ответов, ROC-AUC (результат метрики качества), время обучения (сколько требовалось данной нейросети времени, чтобы дать конкурентоспособный ответ).

На рис. 4 покажем лучшие результаты их работы.

Ниже представим таблицу результатов обучения нейронных сетей (табл. 1).

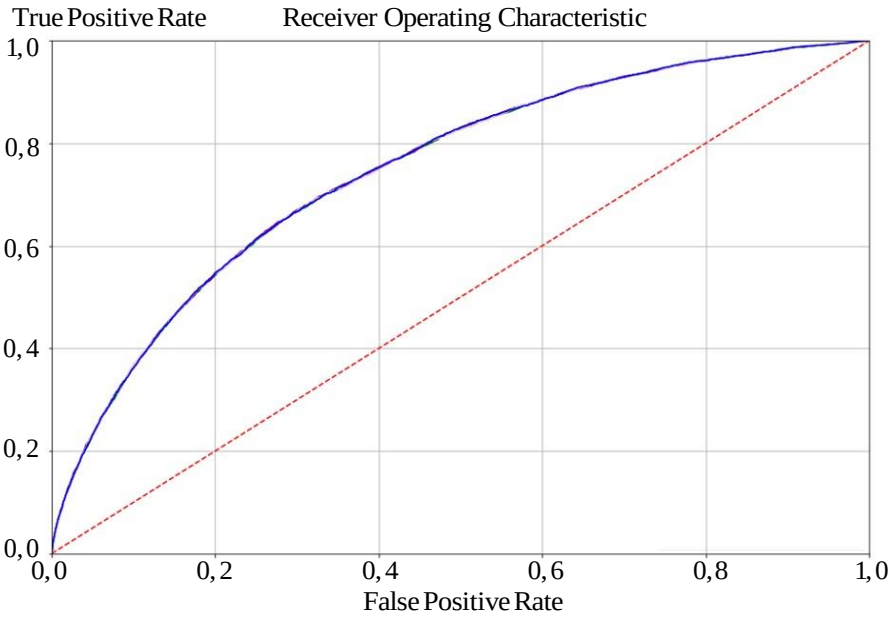


Рис. 4. Результаты предсказаний лучших из построенных архитектур нейронных сетей:

— NN3S_AUC_200_100_50 = 0,74679 ;

— NN3S_AUC_50_100_200 = 0,74762 ;

— NN3S_AUC_150_50_150 = 0,74694 ;

— NN3S_AUC_50_250_50 = 0,74729

Таблица 1

Результаты работы нейронных сетей

№	Нейросеть	Доля правильных ответов	ROC-AUC	Время обучения
1	NN1S (200)	0,8515	0,74607	35 с
2	NN1TS (200)	0,8516	0,70644	34 с
3	NN1S (100)	0,8512	0,74658	30 с
4	NN1TS (100)	0,8519	0,70053	30 с
5	NN1S (50)	0,8507	0,74734	23 с
6	NN1TS (50)	0,8521	0,71643	27 с
7	NN2S (200, 100)	0,8514	0,74510	43 с
8	NN2TS (200, 100)	0,8512	0,71623	44 с
9	NN2S (100, 200)	0,8520	0,74627	40 с
10	NN2TS (100, 200)	0,8499	0,71511	39 с
11	NN3S (200, 100, 50)	0,8526	0,74679	47 с

Продолжение таблицы 1

12	NN3TS (200, 100, 50)	0,8521	0,71502	52 с
13	NN3S (50, 100, 200)	0,8516	0,74762	40 с
14	NN3TS (50, 100, 200)	0,8521	0,72079	45 с
15	NN3S (150, 50, 150)	0,8518	0,74694	43 с
16	NN3TS (150, 50, 150)	0,8524	0,72219	44 с
17	NN3S (50, 200, 50)	0,8521	0,74729	40 с
18	NN3S (50, 200, 50)	0,8513	0,71967	42 с

Результаты. Приведенные результаты, без сомнений, можно улучшить путем дальнейшей работы с признаками: комбинируя их и присоединяя дополнительные таблицы. Также можно провести эксперимент с гиперпараметрами модели[15].

Теперь на рис. 5 построим ROC-кривую, учитывающую также и нейронную модель, выведенную в настоящей работе.

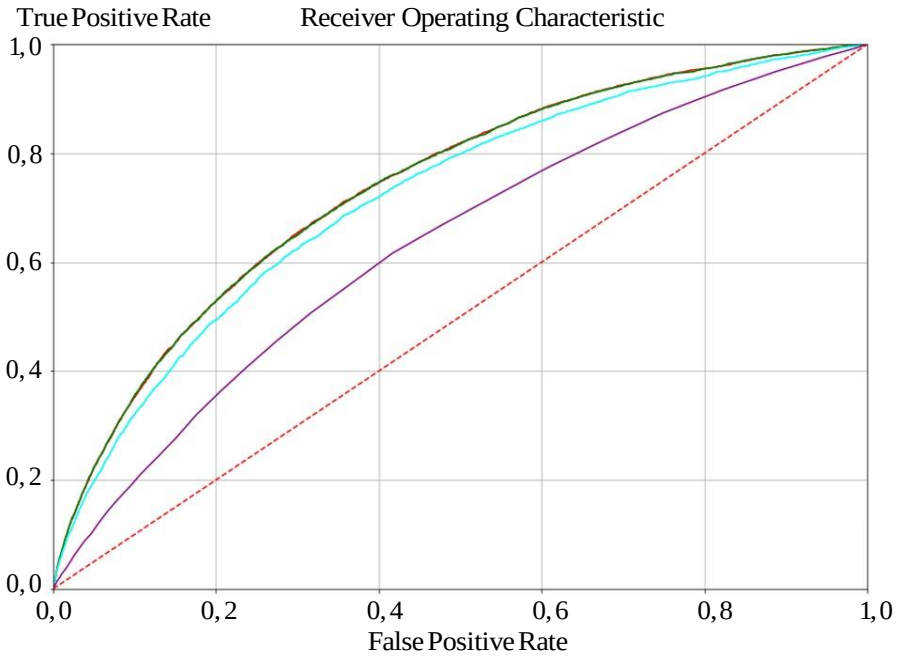


Рис. 5. Результаты предсказаний лучших из построенных моделей машинного обучения и нейронных сетей:
— Custom_model = 0,73940 ; — LogReg = 0,73900 ;
— XGB = 0,71891 ; — RF = 0,63067

Анализируя результаты, представленные в табл. 1 и на рис. 5 делаем следующие выводы:

- наилучшей моделью для решения поставленной задачи является трехслойная нейронная сеть с сигмоидной функцией активации и возрастающим количеством нейронов в скрытых слоях;
- использование сигмоидной функции активации увеличивает устойчивость нейросети относительно количества нейронов в скрытых слоях, а также относительно глубины нейронной сети;
- глубина нейронной сети для решения задачи кредитного скоринга не является решающим фактором для выбора нейронной сети, поскольку трехслойная нейронная сеть (лучшая из полученных) требует большего времени обучения, чем однослойная с малым количеством нейронов, хотя результаты разнятся в тысячных долях;
- использование функции активации гиперболический тангенс не является оптимальным выбором функции активации в нейронной сети (как и смешение функций активаций), хотя время обучения нейронной сети с данной функцией активации не меньше времени, что необходимо обучению нейронной сети с сигмоидной функцией активации;
- обучение нейронных сетей оказалось намного быстрее и эффективнее обучения стандартных моделей машинного обучения. Более того, потребовалось нахождение оптимального количества деревьев для обучения ансамблевых методов деревьев решений;
- логистическая регрессия показывает практически сходный, но худший результат по сравнению с построенной моделью нейронной сети.

ЛИТЕРАТУРА

[1] Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. New York, Sptringer, 2009, 745 p.
[2] Marsland S. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2009, 406 p.
[3] Хайкин С. *Нейронные сети: полный курс, 2-е издание*. Москва, Издательский дом «Вильямс», 2008, 1104 с.
[4] Goodfellow I., Bengio Y., Courville A. *Deep learning*. The MIT Press, 2016, 800 p.
[5] Coelho L.P., Richert W. *Building machine learning systems with Python*. Packt Publishing Ltd, 2013, 290 p.
[6] Воронцов К.В. *Машинное обучение: курс лекций* [Электронный ресурс], 2016. URL: http://www.machinelearning.ru/ (дата обращения: 15.10.2022)
[7] Bellman R. *The Bellman continuum: a collection of the works of Richard E. Bellman*. World Scientific Publishing Company, 1987, 892 p.
[8] Bellman R. *Adaptive control processes: a guided tour*. Princeton University Press, 2015, 276 p.
[9] Powell W.B. *Approximate dynamic programming: solving the curses of dimensionality*. Wiley, 2011, 656 p.
[10] Marimont R.B. Shapiro M.B. Nearest neighbour searches and the curse of dimensionality. *Journal of the Institute of Mathematics and Its Applications*, 1979, vol. 24, iss. 1, pp. 59–70.

[11] Radovanovic M., Nanopoulos A., Ivanovic M. Hubs in space: popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 2010, vol. 11, pp. 2487–2531.
[12] Spencer J. *Ten lectures on the probabilistic method*. Philadelphia, SIAM Publ., 1994, 83 p.
[13] de Grey A.D.N.J. *The chromatic number of the plane is at least 5*. Cornell University, 2018. DOI: arXiv:1804.02385
[14] Лекун Я. *Как учится машина. Революция в области нейронных сетей и глубокого обучения*. Москва, Интеллектуальная Литература, 2021, 335 с.
[15] Werbos P.J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 1990, vol. 78, iss. 10, pp. 1550–1560.

Статья поступила в редакцию 16.10.2022

Ссылку на эту статью просим оформлять следующим образом:
Кадиев А.Д., Чибисова А.В. Нейросетевые методы решения задачи кредитного скоринга. *Математическое моделирование и численные методы*, 2022, № 4, с. 81–92.

Кадиев Алан Джавидович — студент кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана. e-mail: kastan2010@mail.ru

Чибисова Анастасия Владимировна — старший преподаватель кафедры «Вычислительная математика и математическая физика» МГТУ им. Н.Э. Баумана. e-mail: anachibisova@bmstu.ru

Neural network methods for solving the problem of credit scoring

© A.D. Kadiev, A.V. Chibisova

Bauman Moscow State Technical University, Moscow, 105005, Russia

The mathematical derivation of the presented neural network model is demonstrated. Reduction of the classification problem to an optimization problem. Produced recon-naissance data analysis, as well as their preprocessing for further use in training classification algorithms. The architectures of neural networks were designed depending on the activation function, the number of hidden layers of the neural network and the number of neurons in the hidden layers. More than ten neural networks were trained to solve the task of credit scoring. The calculation of the learning time of neural networks was made. The solution of the problem using classical machine learning algorithms is presented. It could be seen that the standard deviation of accuracy and ROC AUC for neural networks is greater than that of a random forest. This is due to the fact that we choose the initial weights randomly and calculate the gradients not on the entire sample, but on small parts, which adds some learning error. But these deviations were not only for the worse. In the best situations, according to both metrics, neural networks showed the worst result by a couple of percent. The analysis of results is made. Comparative analysis shows that neural networks have better classification quality than classical machine learning algorithms, and also that neural networks have less training time than classical machine learning algorithms. Graphs and tables displaying the results obtained are presented.

Keywords: *neural networks, machine learning, random forest, logistic regression, gradient boosting*

REFERENCES

[1] Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. New York, Sptringer, 2009, 745 p.
[2] Marsland S. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2009, 406 p.
[3] Haykin S. *Neural networks: a comprehensive foundation. Second edition*. Ontario, Canada. Prentice Hall, 1999, 823 p.
[4] Goodfellow I., Bengio Y., Courville A. *Deep learning*. The MIT Press, 2016, 800 p.
[5] Coelho L.P., Richert W. *Building machine learning systems with Python*. Packt Publishing Ltd, 2013, 290 p.
[6] Vorontsov K.V. *Mashinnoe obuchenie: kurs lekcij* [Machine learning: a course of lectures] [Electronic resource], 2016. URL: http://www.machinelearning.ru/ (accessed: 15.10.2022)
[7] Bellman R. *The Bellman continuum: a collection of the works of Richard E. Bellman*. World Scientific Publishing Company, 1987, 892 p.
[8] Bellman R. *Adaptive control processes: a guided tour*. Princeton University Press, 2015, 276 p.
[9] Powell W.B. *Approximate dynamic programming: solving the curses of dimensionality*. Wiley, 2011, 656 p.
[10] Marimont R.B. Shapiro M.B. Nearest neighbour searches and the curse of dimensionality. *Journal of the Institute of Mathematics and Its Applications*, 1979, vol. 24, iss. 1, pp. 59–70.
[11] Radovanovic M., Nanopoulos A., Ivanovic M. Hubs in space: popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 2010, vol. 11, pp. 2487–2531.
[12] Spencer J. *Ten lectures on the probabilistic method*. Philadelphia, SIAM Publ., 1994, 83 p.
[13] de Grey A.D.N.J. *The chromatic number of the plane is at least 5*. Cornell University, 2018. DOI: arXiv:1804.02385
[14] Lekun Ya. *Kak uchitsya mashina. Revolyuciya v oblasti nejronnyh setej i glubokogo obucheniya* [How the machine learns. A revolution in neural networks and deep learning]. Moscow, Intellektual'naya Literatura Publ., 2021, 335 p.
[15] Werbos P.J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 1990, vol. 78, iss. 10, pp. 1550–1560.

Kadiev A.D., Student of Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University. e-mail: kastan2010@mail.ru

Chibisova A.V., senior lecturer of Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University. e-mail: anachibisova@bmstu.ru