

УДК 004.032.26

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ФОНДОВОГО РЫНКА

© Рязанов А.П.

Самарский национальный исследовательский университет
имени академика С.П. Королева, г. Самара, Российская Федерация

e-mail: apple556@yandex.ru

Машинное обучение не только предоставляет новые инструменты, но и решает другую проблему. Грубо говоря, оно вращается вокруг проблемы предсказания: произвести предсказания Y из X [1]. Успех машинного обучения на интеллектуальных задачах во многом связан с его способностью обнаруживать сложные структуры, которые не были указаны заранее. Оно позволяет приспособливать сложные и очень гибкие функциональные формы к данным без простого переоснащения; оно находит функции, которые работают хорошо на новых данных [2].

В современном мире нейронные сети активно используются в финансах и банковской сфере. Например, для одобрения кредита определенному человеку. Нейронные сети позволяют это делать достаточно эффективно и с высокой точностью [3].

Меня заинтересовала тема прогнозирования курса акции определенной компании относительно котировок предыдущего дня.

В качестве архитектуры сети была выбрана модель перцептрона. Данная модель проста в разработке и эффективна относительно нашей задачи. На выходе – три нейрона для оценки динамики движения акции: вверх, вниз, без изменений. Решение о движении курса акции будет приниматься на основе выходного значения – более приближенного к единице (так как функция активации – сигмоида) [4]. Входные данные: цена открытия, цена закрытия, максимальные и минимальные значения в течение дня, объем торгов. Относительно входных данных будет делаться вывод о цене закрытия акции на следующий день торгов.

В первую очередь для решения данной задачи нужно получить объем исторических данных. Я использовал библиотеку `yfinance` для языка Python, так как нейронная сеть спроектирована с использованием этого же языка программирования.

После получения исторических данных (я взял акции AAPL) я приступил к выбору функции активации и написанию самой нейронной сети.

В качестве функции активации выбрана сигмоида. Библиотека `scipy.special` содержит сигмоиду. Для работы с многомерными массивами использована библиотека `numpy` [4] (рис. 1).

```
25 # использование сигмоиды в качестве функции активации
26 self.activation_function = lambda x: scipy.special.expit(x)
27 pass
```

Рис. 1. Задание сигмоиды в качестве анонимной функции в коде программы

Вот так выглядит функция тренировки сети, которая пересчитывает весовые коэффициенты относительно ошибок, полученных в ходе вычислений (рис. 2).

```

28
29 # тренировка нейронной сети
30 def train(self, inputs_list, target_list):
31     # преобразовать список входных значений в двумерный массив
32     inputs = numpy.array(inputs_list, ndmin=2).T
33     targets = numpy.array(target_list, ndmin=2).T
34
35     # рассчитать входящие сигналы для скрытого слоя
36     hidden_inputs = numpy.dot(self.wih, inputs)
37     # рассчитать исходящие сигналы для скрытого слоя
38     hidden_outputs = self.activation_function(hidden_inputs)
39
40     # рассчитать входящие сигналы для выходного слоя
41     final_inputs = numpy.dot(self.who, hidden_outputs)
42     # рассчитать исходящие сигналы для выходного слоя
43     final_outputs = self.activation_function(final_inputs)
44
45     # расчет ошибки
46     output_errors = targets - final_outputs
47
48     # ошибки скрытого слоя - это ошибки output_errors,
49     # распределенные пропорционально весовым коэффициентам связей
50     # и рекомбинированные на скрытых узлах
51     hidden_errors = numpy.dot(self.who.T, output_errors)
52
53     # обновить весовые коэффициенты связей между скрытым и выходным слоями
54     self.who += self.lr * numpy.dot((output_errors * final_outputs * (1.0 - final_outputs)),
55                                     numpy.transpose(hidden_outputs))
56
57     # обновить весовые коэффициенты связей между входным и скрытым слоями
58     self.wih += self.lr * numpy.dot((hidden_errors * hidden_outputs * (1.0 - hidden_outputs)),
59                                     numpy.transpose(inputs))
60     pass
61

```

Рис. 2. Функция тренировки сети

Согласно описанной методике, достигаются значения в 60–75 % в угадывании тенденции рынка. Можно оценивать данный результат как положительный. Безусловно, стоит использовать нейронные сети в качестве инструмента для анализа и прогноза движений фондового рынка.

Библиографический список

1. What is Machine Learning? 2017. URL: <https://mlplatform.nl/what-is-machine-learning> (дата обращения: 26.04.2021).
2. Machine learning solutions to challenges in finance: An application to the pricing of financial products / Lirong Gan, Huamao Wang, Zhaojun Yang // Technological Forecasting and Social Change. 2020. Vol. 153. P. 3–25.
3. Applications of Machine Learning in Finance. 2020. URL: <https://algorithmxlab.com/blog/applications-machine-learning-finance> (дата обращения: 26.04.2021).
4. Рашид Тарик. Создаем нейронную сеть. СПб.: ООО «Альфа-книга», 2017. 272 с.