

Таким образом, мы получили минимальный Gross (без агентских и других скидок) бюджет при данных поставках, который составил 134 662,59 \$. В последующем, можно усложнять данную модель и формировать медиапланы на каждом канале по передачам учитывая различные коэффициенты, скидки и прайм-тайм.

Итогом данной научной работы является подтверждение тезиса, что в некоторых случаях рекламный поток можно рассматривать как материальный. Следует сформировать отдельный инструментарий для управления рекламой, основанной на логистических принципах и методах, который позволит в разы ускорить рабочий процесс, минимизировать затраты на рекламу и эффективнее формировать направленность и мощность рекламного потока.

Библиографические ссылки

1. Бузин В. Н., Бузина Т. С. Медиапланирование. Теория и практика. М. : ЮНИТИ-ДАНА, 2012.
2. Информационные логистические системы в деятельности рекламных агентств: на примере рекламного рынка г. Ростова-на-Дону [Электронный ресурс]. URL: <http://www.dissercat.com/content/informatsionnye-logisticheskie-sistemy-v-deyatelnosti-reklamnykh-agentstv-na-primere-reklamn#ixzz5EQ6anljc> (дата обращения 20.04.2018).
3. Кириллов А. В., Целин В. Е. Логистика : метод. указания. Самара : Изд-во СГАУ, 2006.
4. Сурков В. В. Логистические аспекты управления рекламной деятельностью // Российское предпринимательство. 2007. Т. 8. № 7. С. 103–106.

НЕЙРОННЫЕ СЕТИ НА JAVASCRIPT

А. А. Егорочкина, Т. Н. Кухарчук

*ГУО «Институт бизнеса и менеджмента технологий» БГУ,
г. Минск; aneshka.eg@mail.ru;
науч. рук. – А. Ф. Смалюк, канд. физ.-мат. наук, доц.*

Целью работы была разработка приложения, которое распознает рукописные цифры. Для решения данной задачи была использована технология нейронных сетей, был разработан интерфейс с помощью языка разметки HyperText Markup Language и языка программирования JavaScript. По итогам работы было написано две программы: программа для обучения нейронной сети, результат которой подключался к самому приложению, и программа приложения. При запуске приложения в интернет-браузере отображается окно для рисования цифры и кнопки управления «Определить цифру» и «Очистить». При нажатии на кнопку «Определить цифру» после рисования цифры выводится поле, где отображается ответ на то, какая цифра была введена. Программа определяет цифру примерно в 70 % случаях.

Ключевые слова: нейронная сеть; JavaScript; HyperText Markup Language; распознавание текста; рукописный ввод; программа.

ВВЕДЕНИЕ

Что такое нейронная сеть?

Представим себе человеческий мозг: он представляет собой тысячи нейронов, соединенных между собой в единую сеть. Чем-то похожим и является компьютерная нейронная сеть. Рассмотрим это подробнее.

Компьютерный нейрон (рис.1) – это вычислительная единица, содержащая функцию [1]. Эта функция получает входные данные, обрабатывает их и передает дальше по сети. Связи между нейронами называются синапсами. Они могут быть входящие и исходящие. Над входящими синапсами ставятся веса, и входом для нейрона будет произведение выходных данных предыдущего нейрона и веса синапса.

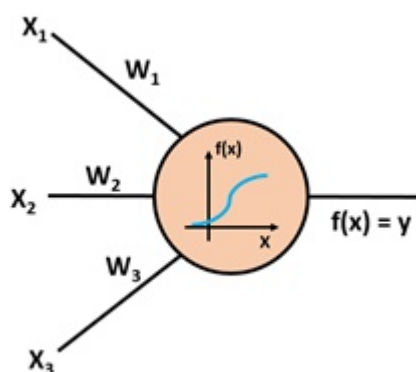


Рис. 1. Компьютерный нейрон

Функция, которую содержит нейрон, называется функция активации. Есть три вида функции активации [2]:

1. Линейная – данная функция используется редко, только в тех случаях, когда нужно протестировать нейронную сеть или передать значение без операций над ним.

2. Сигмоид – это самый популярный вид функции активации, диапазон ее значений от нуля до единицы включая.

3. Гиперболический тангенс – эта функция используется в случае, если нужны и отрицательные значения (например, продажи могут идти как вверх, так и вниз), диапазон ее значений от минус единицы до единицы включая. Использовать данную функцию только с положительными значениями не имеет смысла, так как это ухудшает результаты нейронной сети.

Что использовалось для работы на JavaScript?

1. База данных Modified National Institute of Standards and Technology [3], содержащая базу изображений рукописных цифр. Доступно 60 000 картинок для обучения.

2. Библиотека brain.js [4], которая содержит все необходимые методы для обучения и тестирования сети.

3. Node.js [5] – среда выполнения JavaScript на сервере для того, чтобы не обучать сеть с нуля при каждом тестировании.

4. Notepad++ – программа для написания кода как клиентской, так и серверной части программы.

Концепция нейронных сетей подразумевает обучение. Есть нейроны, они связаны между собой, получают на вход какие-то данные, но нужно расставить веса над синапсами. Это и реализуется путем обучения нейронной сети.

Параметры обучения [6]:

- тренировочный сет – последовательность входных данных для всей нейронной сети;
- итерация – простыми словами это счетчик того, сколько тренировочных сетов было пройдено;
- ошибка – процентная величина, которая показывает расхождение между ожидаемым и полученным результатом. Ошибка должна уменьшаться после каждой итерации, и если этого не происходит, то программа работает неправильно.

Разработка программы распознавания рукописных цифр проходила в четыре этапа.

ЭТАП 1. ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

В отдельном файле создавался объект нейронной сети и вводились настройки обучения, а именно: количество итераций 70, количество картинок для обучения в одной итерации 60000, ошибка 0,0005. Затем этот файл отправлялся на запуск в Node.js. Там сначала подключается база данных MNIST, которая содержит в себе картинки для обучения нейронной сети, а затем выполняется запуск написанной нами программы.

Алгоритм обучения устроен таким образом, что процесс обучения нейронной сети прекращается либо по завершении итераций количеством 70 штук, либо по достижению ошибки равной 0,0005. В нашем случае программу останавливало прохождение через 70 итерация. На обучение такой нейронной сети уходило порядка 5–6 часов.

После обучения нейронная сеть в формате JSON сохраняется в новом файле, для удобства будем его называть «файл нейронной сети». На этом этап обучения завершается.

ЭТАП 2. РАЗРАБОТКА ИНТЕРФЕЙСА

На втором этапе работы разрабатывался интерфейс для работы программы в браузере на базе HyperText Markup Language и JavaScript.

Нужно было поле для рисования цифры, для этих целей использовался Canvas, для которого была включена возможность работы с сенсорными экранами. На устройствах без сенсорного экрана программа тоже работает, но удобнее рисовать от руки. Также были созданы кнопки для очистки этого поля и для запуска программы распознавания.

ЭТАП 3. НАПИСАНИЕ ПРОГРАММЫ РАСПОЗНАВАНИЯ

Как уже отмечалось ранее, введенная цифра сохраняется на Canvas, но в базе данных MNIST есть своя специфика: изображения там хранятся в виде одномерных массивов, где цифра как бы прорисована числовыми значениями отличными от нуля при отображении этих массивов в формате 28x28 элементов. Поэтому изображение, введенное в Canvas, должно быть нормализовано под эту базу. Для этого был создан еще один Canvas размером 28x28 пикселей, куда помещалось изображение с цифрой после уменьшения. Маленький Canvas в пользовательском интерфейсе не отображался.

Далее уменьшенное изображение нужно перевести в числовой вид. Для этого создавался одномерный массив длиной 784 (это количество пикселей в уменьшенном изображении), наполненный нулями, и происходило обращение к каждому пикселю изображения: если пиксель был цвета фона, то в массиве значение, отведенное для этого пикселя, оставалось нулевым, в противном случае оно менялось на единицу. Таким образом, цвет фона и цвет пера можно менять на любые цвета, в данном случае использовались синий фон и красный цвет пера.

После приведения изображения к виду массива можно начинать запуск программы. Сначала в интерфейс подключается файл обученной нейронной сети в виде объекта JSON и библиотека brain.js. Затем осуществляется запуск нейронной сети, и входными данными для нее будет массив «оцифрованного» изображения. В качестве результата нейронная сеть возвращает одномерный массив размером 10, по элементу для каждой цифры от 0 до 9 по порядку. Значения в этом массиве находятся в диапазоне от нуля до единицы, это вероятности того, каким числом является введенная цифра; то есть, например, в нулевом элементе данного массива содержалась информация о том, какова вероятность, что введенная цифра являлась нулем. Из выходного массива отбирается номер элемента, значение которого наибольшее, и выводится результат в интерфейс.

ЭТАП 4. ТЕСТИРОВАНИЕ

Для тестирования программы была создана Google-форма, где была указана ссылка на веб-страницу, где находилась программа (рис.2). Пользователям нужно было указать, распознались ли введенные ими цифры от нуля до девяти. Было опрошено 26 человек, и выяснилось, что программа распознает цифры в 76 % случаях.

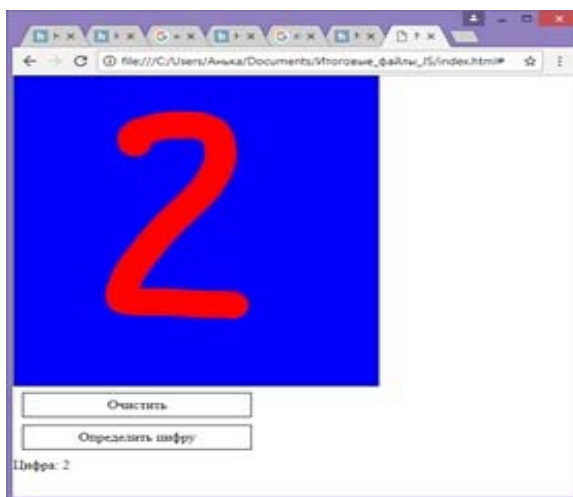


Рис. 2. Результат тестирования программы с цифрой «2»

ЗАКЛЮЧЕНИЕ

Результатом научной работы стала разработка мини-приложения с использованием нейронных сетей. В статье были выделены ключевые моменты, понимание которых поможет в начале работы с нейронными сетями.

Библиографические ссылки

1. Нейронные сети для начинающих. Ч. 1 [Электронный ресурс]. URL: <https://habr.com/post/312450> (дата обращения: 30.03.2018).
2. Функции активации в нейронных сетях [Электронный ресурс]. URL: <http://www.aiportal.ru/articles/neural-networks/activation-function.html> (дата обращения: 02.04.2018).
3. *LeCun Ya., Cortes C., Christopher J. C. Burges* // The MNIST database of hand written digits [Electronic resource]. URL: <http://yann.lecun.com/exdb/mnist> (date of access: 02.04.2018).
4. Brain.js [Electronic resource]. URL: <https://github.com/BrainJS/brain.js> (date of access: 02.04.2018).
5. Node.js [Electronic resource]. URL: <https://nodejs.org/en> (date of access: 02.04.2018).
6. Как работает нейронная сеть [Электронный ресурс]. URL: <https://ideas-garden.com/blog/poiskovye-sistemy/kak-rabotaet-nejronnaya-set> (дата обращения: 01.04.2018).