

ОПЫТ РАЗРАБОТКИ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ В РАМКАХ УЧЕБНОГО ПРОЦЕССА БАКАЛАВРОВ

Качалов О.И., Скворцова Л.А.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский Технологический университет» (МТУ), 119454, Россия, г. Москва, проспект Вернадского, д. 78, e-mail: okachalov@inbox.ru, myscvorec@yandex.ru

Описан опыт разработки и исследования систем искусственного интеллекта в рамках учебного процесса бакалавров на примере создания искусственной нейронной сети для распознавания рукописных цифр. Учащимся получены знания о внутренней структуре и процессах, происходящих в искусственных нейронных сетях, выработаны умения по организации обучения нейронной сети (точность распознавания после обучения сети составила 96%), получены навыки по процессу разработки и тестирования нейронной сети.

Ключевые слова: учебный процесс, компетенции, нейронная сеть, обучение, нейрон, синапс, ошибка, распознавание.

EXPERIENCE IN THE DEVELOPMENT OF AN ARTIFICIAL NEURAL NETWORK IN THE FRAMEWORK OF THE EDUCATIONAL PROCESS OF BACHELORS

Kachalov O.I., Skvortsova L.A.

Federal State Educational Institution of Higher Education "Moscow Technological University", 119454, Russia, Moscow, Vernadscogo avenue, 78, e-mail: okachalov@inbox.ru, myscvorec@yandex.ru

The article describes the experience of the development and practical study of artificial intelligence systems in the educational process of bachelors on the example of creating an artificial neural network for recognition of handwritten digits. Students received knowledge about the internal structure and processes occurring in artificial neural networks, developed skills for training neural network (recognition accuracy after learning the network was 96%), obtained skills in the process of development and testing of the neural network.

Key words: educational process, competence, neural network, learning, neuron, synapse, error, recognition.

Введение

Результатом современного комплексного подхода к организации учебного процесса, сочетающего теоретическую и практическую составляющие, является высокая степень готовности выпускников к решению актуальных задач в профессиональной деятельности. Сегодня одним из трендов развития ИТ-индустрии является развитие и широкое применение систем искусственного интеллекта (СИИ). При этом, можно отметить растущая доступность инструментов по созданию интеллектуальных приложений, например, Bonsai, Keras, и TensorFlow упрощают реализацию моделей глубинного обучения (deep learning models), а облачные платформы, как Google API и Microsoft Azure позволяют создавать интеллектуальные приложения, не беспокоясь о настройках и поддержке сопровождающей инфраструктуры. Уже сейчас СИИ становятся доступнее и понятнее даже для неопытных специалистов. Намечается тенденция роста узкоспециализированных приложений искусственного интеллекта, такие как, роботы для личного использования, промышленности и образования, боты: CRM-системы, личные помощники и консультанты клиентов, отраслевые ИИ системы для индустрий: финансы, безопасность, промышленность, продажи и др.

Следствием является и рост интереса со стороны учащихся к разработке узкоспециализированных приложений искусственного интеллекта, в частности нейронных сетей. В данной статье описан опыт решения подобной задачи в рамках учебного процесса бакалавров четвертого семестра на примере создания искусственной нейронной сети для распознавания рукописных цифр.

На сегодняшний день эффективные и распространенные сценарии использования нейронных сетей – это распознавание объектов, их классификация и предсказание событий. Самостоятельная разработка и обучение нейронной сети является занимательным процессом. Методики разработки искусственных нейронных сетей

представлены, в основном, в текстовом виде[1, 2] или в формате видеолекций[3].

Главное преимущество нейронных сетей – способность обучения. Для того, чтобы создать нейронную сеть можно использовать готовые программные библиотеки, например TensorFlow [4], Keras [5], cuDNN [6]. Перечисленные библиотеки являются мощными программными инструментами для создания нейронных сетей. Однако к недостаткам следует отнести затрудненность понимания внутреннего устройства сети, скрытие алгоритма работы методов, хранения данных и внутренней топологии сети и ее процесса обучения. Поэтому для изучения нейронных сетей было принято решение – не использовать готовые библиотеки.

Назначение разрабатываемой нейронной сети, как уже отмечалось, – распознавание рукописных цифр. Сама по себе данная задача не нова и готовые библиотеки способны с легкостью справиться с поставленной задачей. Но поскольку готовые сторонние библиотеки не используются, то данная задача не является тривиальной.

Самостоятельная разработка ИНС для распознавания рукописных цифр

Задачи по распознаванию каких-либо объектов на изображении отлично выполняют сверточные нейронные сети (рис. 1.1). Используя карту признаков, они шаг за шагом анализируют представленное изображение, сравнивая фрагменты изображения с изученными признаками. Сверточные нейронные сети достаточно сложны в реализации по сравнению с обычным перцептроном. В связи со сложностью сверточных нейронных сетей было принято решение отказаться от изучения и разработки данной сети в пользу более простой в понимании и реализации.

Самая простая нейронная сеть – это перцептрон Френка Розенблатта (рис. 1.2) [7]. Именно данный тип сети был выбран для изучения процессов, проходящих в нейронных сетях. Данный тип сети подразумевает i -ое количество входных нейронов (рецепторов), j -ое количество скрытых нейронов и k -ое количество выходных нейронов. При этом все нейроны одного слоя соединены с нейронами другого слоя.

Для обучения перцептрона был выбран итеративный метод обучения, который носит название «метод обратного распространения ошибок». Метод обратного распространения ошибки относится к категории алгоритмов «обучение с учителем». Принцип данного метода заключается в распространении сигнала ошибки в направлении, обратном прямому распространению сигналов в обычном режиме. Важно учитывать, что при использовании данного алгоритма передаточная функция в нейронах должна быть дифференцируемая, иначе использование данного алгоритма невозможно.

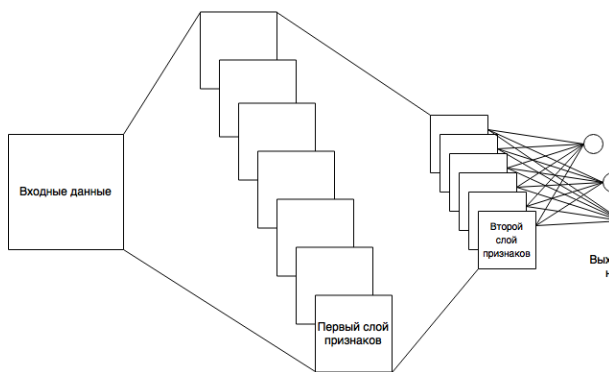


Рис. 1.1 Сверточная нейронная сеть

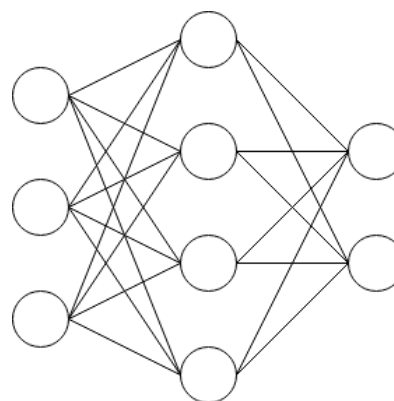


Рис. 1.2 Перцептрон Френка Розенблатта

В качестве передаточной функции используется сигмоидальная функция (рис. 2). Данная функция позволяет перейти от бинарных выходов нейрона к «аналоговым», что очень важно при использовании в задачах распознавания и классификации.

Исходя из поставленной задачи, буквы, которые будет распознавать нейронная сеть, будут представлены в виде изображения. Для более простого считывания изображения и загрузки в нейронную сеть был выбран формат изображения .tga. Внутреннее представление данного формата предполагает кодировку каждого пикселя одним байтом информации, поэтому загрузить изображение в нейронную сеть достаточно просто. Исходный размер изображения равен 12 на 12 пикселей, что дает 144 байт информации на входе в нейронную сеть.

Количество нейронов в скрытом слое должно равняться количеству входных нейронов или больше. В скрытом слое было размещено 200 нейронов.

Выходной слой представляет собой 10 нейронов. Каждый нейрон отображает вероятность изображенной цифры.

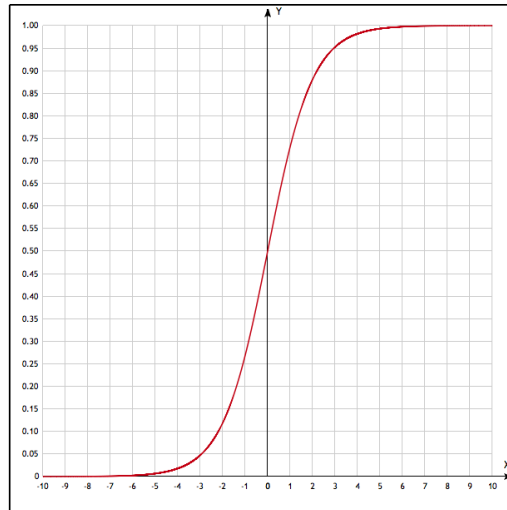


Рис. 2. Сигмоида $y = \frac{1}{1 + e^{-x}}$

Как уже сказано ранее, формат изображения .tga подразумевает кодирование одного пикселя одним байтом. В связи с тем, что нейронная сеть способна работать в пределах от нуля до единицы, входные значения будут приводиться в соответствие с данным диапазоном. Вероятность, получаемая на выходе нейронной сети, изображенной цифры на рисунке будет соответствовать вещественному значению от нуля до единицы.

Программа хранит 2 массива: массив весов синапсов между входным и промежуточным слоем и между промежуточным и выходным слоем. Хранить значения входных, промежуточных и выходных нейронов нет необходимости, потому что значения будут изменяться в зависимости от входных данных.

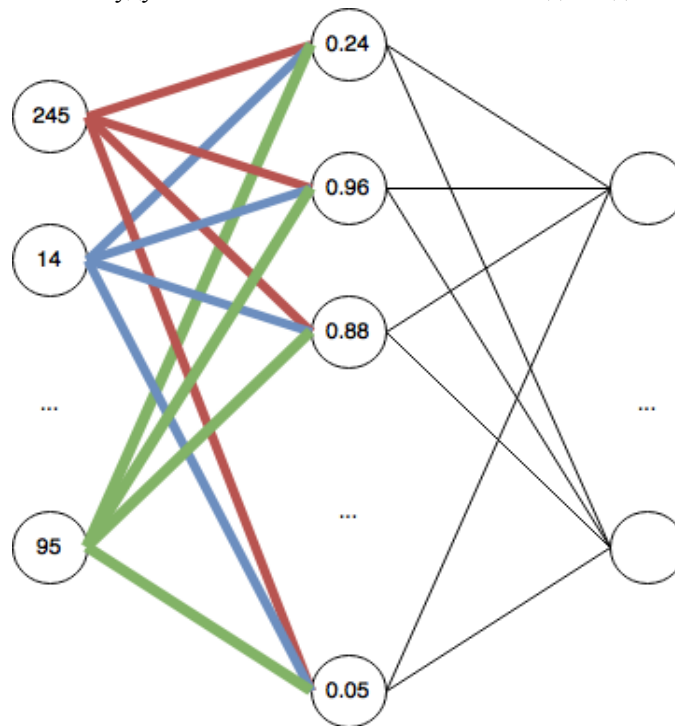


Рис. 3. Нейронная сеть на этапе распространения сигналов от входных нейронов к скрытым

Как работает нейронная сеть? При подаче на данных программа начинает считать значение скрытых нейронов по следующей формуле: $N_j = \sum_0^n w_i * N_i$, где N_j – значение j-ого скрытого нейрона, w_i – вес синапса между входным нейроном и скрытым нейроном, N_i – значение i-ого входного нейрона, n – количество нейронов в входном слое. Далее программа начинает расчет значений выходного слоя по такой же формуле.

На выходе получаются 10 вещественных значений. Каждое из этих значений отвечает вероятности одной из 10 изображений цифр.

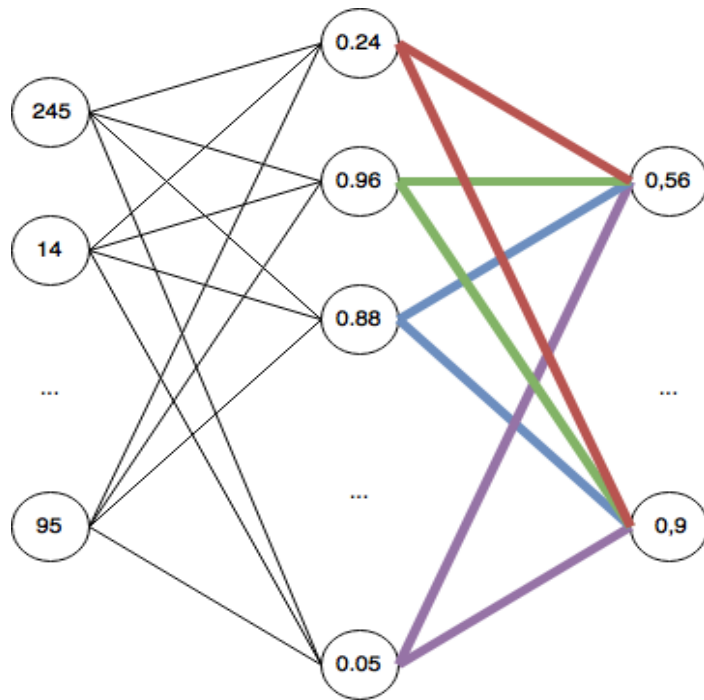


Рис. 4. Нейронная сеть на этапе распространения сигналов от скрытых нейронов выходным

Как же нейронная сеть обучается? Как уже ранее рассказывалось, обучение происходит с помощью распространения сигнала ошибки в обратном направлении. Т.к. метод обратного распространения ошибки относится к методам обучения с учителем, при использовании данного метода необходимо точно знать ответ для каждого конкретного входного значения.

Для успешной коррекции весов синапсов необходимо вычислить значение ошибки для каждого слоя нейронов, кроме входного (Рис. 5). На Рисунке 5, в темных кружках указано значение ошибки нейрона. Для выходного слоя значения ошибок будут равняться разнице правильного ответа и полученного результата нейронной сети. Далее на основе этих данных вычисляется ошибка для скрытого слоя по следующей формуле: $E_j = \sum_0^n E_i * w_i$, где E_j – ошибка j-ого скрытого нейрона, E_i – ошибка i-ого выходного нейрона, w_i – вес синапса между входным нейроном и скрытым нейроном, n – количество скрытых нейронов.

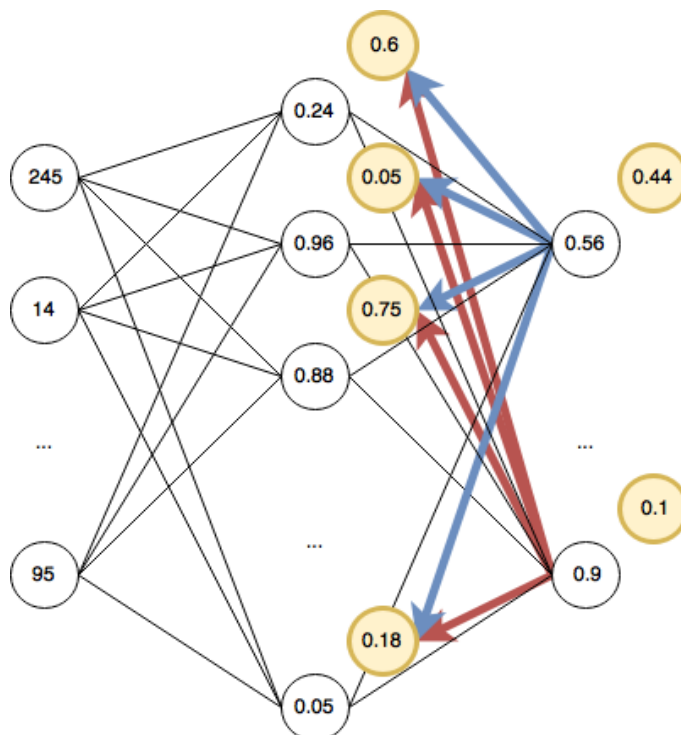


Рис. 5. Расчет ошибок нейронов

После того, как все ошибки нейронов посчитаны необходимо произвести коррекцию весов синапсов. Новый вес синапса вычисляется по формуле: $w_i = \sum_0^n w_i + lr * E_i * \frac{d}{dx}(sigmoid(N_i)) * N_i$, где w_i – вес синапса, lr – коэффициент скорости обучения, E_i – ошибка скрытого слоя нейронов, N_i – значение входного нейрона, n – количество нейронов в скрытом слое. По такому же принципу вычисляются новые веса синапсов между скрытым и выходным слоем нейронов.

При обучении нейронной сети важно не путать количество элементов в обучающей выборке и количеством итераций обучения. Для правильного обучения следует N-количество раз использовать всю обучающую выборку. Также стоит учитывать тот факт, что нейронная сеть не только способна изучать какие-либо признаки объектов и отличать объекты друг от друга, но и также «заучивать» обучающую выборку. Такое свойство нейронной сети называется переобучение. Для проверки качества обучения используется специальная проверочная выборка. Как правило, при обучении ошибка сети уменьшается совместно с увеличением точности распознавания. Но как только ошибка начинает расти, то нейронная сеть начинает адаптироваться к обучающей выборке и обучение нейронной сети необходимо прекратить.

Для качественного обучения была составлена выборка из 500 изображений, по 50 изображений на каждую цифру. Данная выборка была поделена в пропорции 70/30, где 70% элементов были элементы для обучения, оставшиеся 30%, составляли элементы для проверки качества обучения.

При проведении тестирования было выявлено, что сеть способна распознать цифры с вероятностью 96% при небольших изменениях в написании цифры. А что же будет, если изменения в написании цифры будут существенные? Нейронная сеть обучаясь «запоминает» элементы с изображений из обучающей выборки. Если продемонстрировать нейронной сети одно и тоже изображение, но одно изображение будет немного повернуто или уменьшено в размере, либо над ним будут произведены какие-либо другие манипуляции, то, скорее всего, нейронная сеть не сумеет распознать объект верно. Это связано с тем, что при обучении «возбуждаются» конкретные нейроны и нейронная сеть «запоминает» их состояние и верный ответ. При изменении каких-либо выше перечисленных параметров изображения изменяются и нейроны, которые будут реагировать на «возбуждение». Поэтому для более серьезного подхода к распознаванию объектов на изображении применяются другие типы сетей.

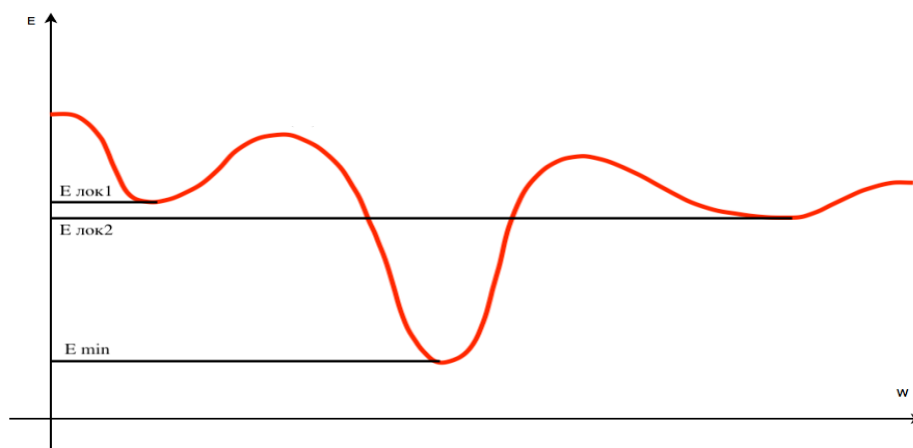


Рис. 6. Интерпретация поверхности ошибок нейронной сети

В связи с тем, что изучение внутренних процессов в нейронной сети достаточно трудоемкая задача для программирования искусственной нейронной сети был выбран язык высокого уровня Python. Язык Python позволил быстро запрограммировать нейронную сеть. В результате разработки был создан класс нейронной сети, в котором были включены методы по вводу данных, распознаванию каких-либо объектов и обучению искусственной нейронной сети. Стоит отметить, что данная нейронная сеть получилась универсальной, поскольку никакие шаги по ограничению применения нейронной сети не были использованы. Данная нейронная сеть будет успешно распознавать чуть-более сложные объекты на изображении, если произвести обучение нейронной сети.

Вывод

В результате опыта по разработке искусственной нейронной сети были изучены внутренние механизмы работы нейронных сетей, а также механизмы их обучения. Как показал опыт разработки – нейронные сети достаточно сложны в своей реализации. Разработанная нейронная сеть – это не готовое программное

обеспечение. Для функционирования нейронной сети ее необходимо обучить. Процесс обучения сети может занять куда больше времени, чем на ее разработку.

В процессе выполнения учащимся поставленной задачи получены знания о внутренней структуре и процессах, происходящих в искусственных нейронных сетях, выработаны умения по организации обучения нейронной сети (точность распознавания после обучения сети составила 96%), получены навыки по процессу разработки и тестирования нейронной сети.

Данная работа позволила учащемуся не только выработать соответствующие компетенции (ОПК-4: способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий; и ПК-1: готовность применять основные методы и инструменты разработки программного обеспечения), но и решить задачу, соответствующую современным тенденциям развития ИТ-индустрии.

Литература

1. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений. — М.: Финансы и статистика, 2004. — 176 с.
2. Жданов А. А. Автономный искусственный интеллект [Электронный ресурс] / А.А.Жданов.—3-е изд. (эл.).—М. : БИНОМ. Лаборатория знаний, 2012. — 359 с.
3. Глубокие нейронные сети на Python [Электронный ресурс]. - Режим доступа: URL: https://www.youtube.com/watch?v=Gx7qxV5nh5o&list=PLtPJ9lKvJ4oiz9aaL_xcZd-x0qd8G0VN_ (21.03.2018)
4. TensorFlow. Programmer's guide [Электронный ресурс]. - Режим доступа: URL: https://www.tensorflow.org/programmers_guide/ (21.03.2018)
5. Keras: The Python Deep Learning library [Электронный ресурс]. - Режим доступа: URL: <https://keras.io> (21.03.2018)
6. nVidia cuDNN GPU Accelerated Deep Learning [Электронный ресурс]. - Режим доступа: URL: <https://developer.nvidia.com/cudnn> (21.03.2018)
7. Саймон Хайкин Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation. — 2-е изд. — М.: «Вильямс», 2006. — С. 1104.

References

1. Barskii A. B. Neironny`e seti: raspoznavanie, upravlenie, prinyatie reshenii. — M.: Finansy` i statistika, 2004. — 176 s.
2. Zhdanov A. A. Avtonomny`i iskusstvenny`i intellekt [E`lektronny`i re- surs] / A.A.Zhdanov.—3-e izd. (e`l.).—M. : BINOM. Laboratoriya znaniy, 2012. — 359 s.
3. Glubokie nejronny`e seti na Python [E`lektronny`j resurs]. - Rezhim dostupa: URL: https://www.youtube.com/watch?v=Gx7qxV5nh5o&list=PLtPJ9lKvJ4oiz9aaL_xcZd-x0qd8G0VN_ (21.03.2018)
4. TensorFlow. Programmer's guide [E`lektronny`j resurs]. - Rezhim dostupa: URL: https://www.tensorflow.org/programmers_guide/ (21.03.2018)
5. Keras: The Python Deep Learning library [E`lektronny`j resurs]. - Rezhim dostupa: URL: <https://keras.io> (21.03.2018)
6. nVidia cuDNN GPU Accelerated Deep Learning [E`lektronny`j resurs]. - Rezhim dostupa: URL: <https://developer.nvidia.com/cudnn> (21.03.2018)
7. Sajmon Hajkin Neironny`e seti: polny`j kurs = Neural Networks: A Comprehensive Foundation. — 2-e izd. — M.: «Vil`yams», 2006. — S. 1104.