

УДК 004.032:512.22

ИССЛЕДОВАНИЕ ОБУЧЕНИЯ ОДНОНАПРАВЛЕННОЙ НЕЙРОСЕТИ НА ПРИМЕРЕ ФУНКЦИИ ГАУССА

Л. А. ЛЯХОВИЧ

Научный руководитель А. Г. КОЗЛОВ
Белорусско-Российский университет

При решении задач в случае, когда обычные алгоритмические методы и решения оказываются недостаточно эффективными или вообще невозможными, все чаще используются искусственные нейронные сети. К таким задачам относятся: семантический поиск, распознавание текста, изображения и речи, системы безопасности, игра на бирже, прогнозирование, анализ платежеспособности и многое другое.

Цель исследования – сравнить значения функции Гаусса, рассчитанные по формуле (1) при заданных коэффициентах, и значения коэффициентов, полученных с использованием обученной нейросети.

Функция Гаусса

$$g(x) = ae^{-\frac{(x-b)^2}{2c^2}}, \quad (1)$$

где параметры a , b , c – произвольные вещественные числа.

Выбор структуры нейросети осуществляется в зависимости от особенностей и сложности задачи. В данном случае будет достаточно простой однонаправленной нейросети с тремя слоями (рис. 1), имеющей плотную (Dense) структуру, в которой каждый нейрон связан со всеми нейронами следующего слоя [1]. Выходной слой будет состоять из нейронов, определяющих коэффициенты a , b , c функции Гаусса.

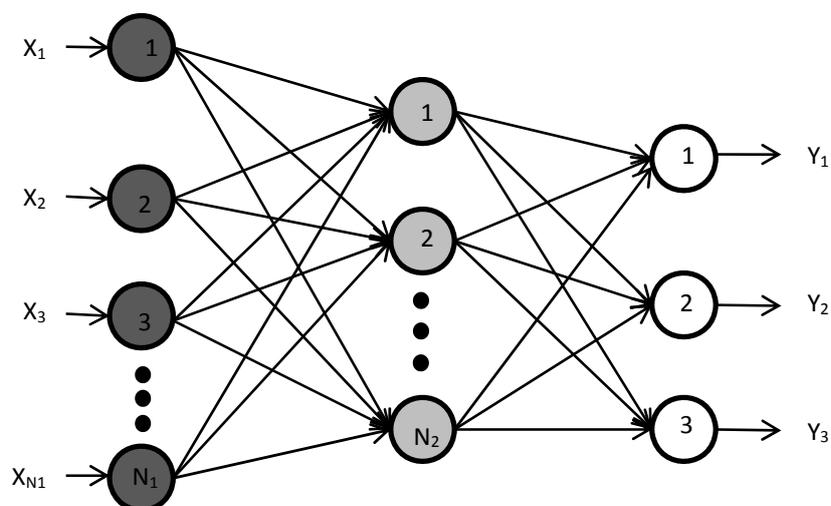


Рис. 1. Структура нейронной сети

Первый слой содержит N_1 узлов, второй – N_2 узлов, выходной – три узла.

Сформируем входной массив и выходной (эталонный) массив, используя библиотеку *numpy* языка программирования Python. Формирование массивов произведем при помощи функции *random.uniform()* из библиотеки *numpy.random* [2].

Задаем компоненты a, b, c и формируем компоненты соответствующего входного вектора. Данную процедуру повторяем M раз, в результате получим входной массив – $M \times N_1$ и выходной массив – $3 \times M$ –элементов.

Используя библиотеку *neurolab* и метод *nl*, создаем нейронную сеть с помощью команды *nl.net.newff*.

В качестве функции активации всех слоев, кроме выходного, будем использовать функцию *LogSig()*. Для выходного слоя воспользуемся функцией *PureLin()* для определения конечного результата.

Для обучения нейронной сети применяем метод

train(input, output, epochs=takts, show=inf, goal=eps),

где *input* – входной массив данных; *output* – выходной массив данных; *epochs* – число циклов обучения; *show* – период вывода информации о состоянии процесса; *goal* – значение ошибки, при котором обучение будет завершено.

Для оценки результата обучения нейронной сети строим график зависимости количества ошибок от числового значения прошедших циклов обучения (рис. 2).

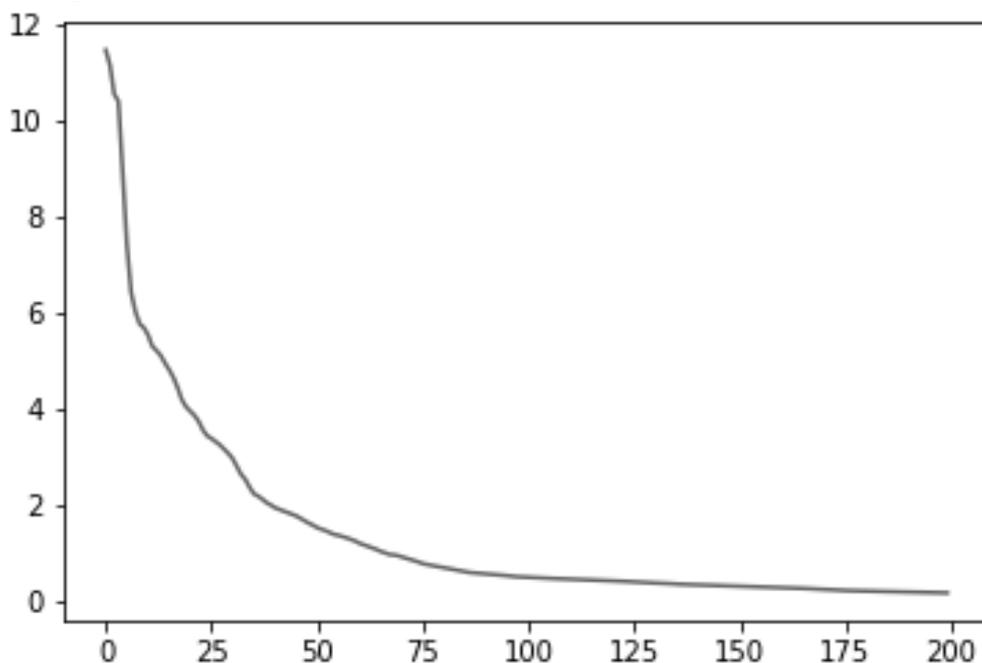


Рис. 2. График зависимости количества ошибок от числового значения прошедших циклов обучения

Произведем построение функции Гаусса с заданными коэффициентами a , b , c и коэффициентами a' , b' , c' , полученными нейросетью.

Построим графики полученных функций (рис. 3).

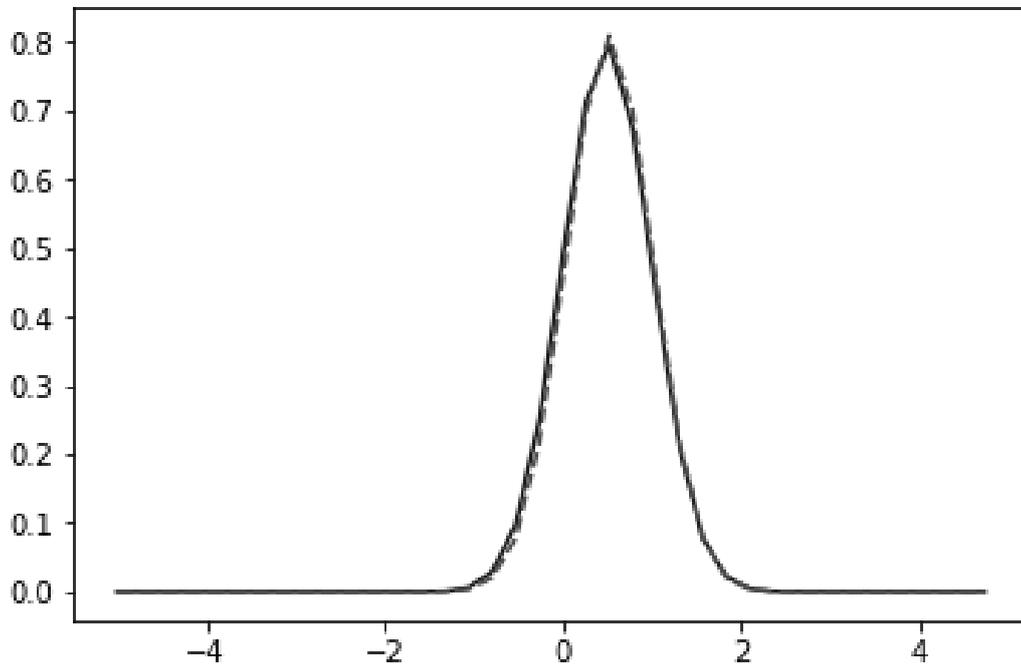


Рис. 3. График функций

Графики функций практически совпали. Это говорит о том, что коэффициенты, полученные нейросетью, практически не отличаются от заданных коэффициентов. Следует заметить, что увеличение слоев нейросети позволяет более точно определить параметры функции.

Актуальность исследований в этом направлении подтверждается ростом различных применений искусственных нейросетей. При решении задач, от результатов которых зависят, например, значительные материальные ресурсы, одновременно используются различные методы и подходы. Обученные нейросети в данном случае демонстрируют уникальные потенциальные возможности, особенно, если задача не решается стандартным способом.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. **Хайкин, С.** Нейронные сети: полный курс / С. Хайкин. – Москва: ООО «И. Д. Вильямс», 2006. – 1104 с.
2. **Шолле, Ф.** Глубокое обучение на Python / Ф. Шолле. – Питер, 2018. – 400 с.
3. **Бринк, Х.** Машинное обучение / Х. Бринк, Д. Ричардс, М. Феверолф. – Питер, 2018. – 336 с.