

Факультет компьютерных технологий и прикладной математики

Кафедра информационных технологий

02.03.03

Приложение нейросетевых алгоритмов

Лабораторная работа № 2

Тема: задачи классификации, многослойные нейронные сети

**Цель работы:** получение знаний и практических навыков построения однослойных нейронных сетей.

**Задание:** требуется разработать алгоритмы и программы решения задач 1-3 на языке Python без использования специализированных библиотек.

**Указания к работе.** Студенты решают задачи № 1, 2, 3. Студент разрабатывает алгоритмы и программы решения задач на языке Python без использования специализированных библиотек, используя систему управления версиями и размещая их на личный Git репозиторий. Студенту будет необходимо подготовить отчёт по лабораторной работе в электронной форме. За лабораторную работу студент может получить оценку «зачтено», «удовлетворительно», «хорошо», «отлично».

*Задача 1.* Ранее для введения нелинейности в нейронные сети в качестве функций активации традиционно использовали сигмоиду или гиперболический тангенс. Однако в последние годы всё большую популярность приобретают различные кусочно-линейные функции активации наподобие тех, которые приведены ниже.

1.  $\Phi(v) = \max\{v, 0\}$  (полулинейный элемент [ReLU]),
2.  $\Phi(v) = \max\{\min[v, 1], -1\}$  (спрямлённый гиперболический тангенс).

В современных нейронных сетях функции активации ReLU (Rectified Linear Unit) и спрямлённый гиперболический тангенс в значительной степени вытеснили сигмоиду и гиперболический тангенс, поскольку их использование упрощает тренировку многослойных нейронных сетей.

Пусть имеется функция XOR, в которой две точки  $\{(0, 0), (1, 1)\}$  принадлежат к одному классу, а две другие точки  $\{(1, 0), (0, 1)\}$  – к другому. Покажите, как разделить два этих класса, используя функцию активации ReLU.

*Задача 2.* Пусть имеется двухмерный набор данных, в котором все точки с  $x_1 > x_2$  принадлежат к положительному классу, а все точки с  $x_1 < x_2$  к отрицательному. Разделителем для этих двух классов является линейная гиперплоскость (прямая линия), определяемая уравнением  $x_1 - x_2 = 0$ .

Создайте набор тренировочных данных с 20 точками, сгенерированными случайным образом в положительном квадранте единичного квадрата. Снабдите каждую точку меткой, указывающей на то, превышает или не превышает её первая координата  $x_1$  вторую координату  $x_2$ .

А. Реализуйте алгоритм перцептрона, обучите его на полученных выше 20 точках и протестируйте его точность на 1000 точках, случайно сгенерированных в единичном квадрате. Используйте для генерирования тестовых точек ту же процедуру, что и для тренировочных.

Б. Замените критерий перцептрона на нейрон типа адалайн (рассмотреть дискретный случай) при реализации тренировки и повторите определение точности вычислений на тех же тестовых точках, которые использовали перед этим.

Удалось ли вам в каком-то из способов получить лучшую точность? Как вы считаете, в каком случае классификация тех же 1000 тестовых точек не изменится значительно, если использовать другой набор из 20 тренировочных точек?

*Задача 3.* Требуется разработать и исследовать нейронную сеть обратного распространения, предназначенную для распознавания образов.

Даны в виде матрицы  $3 \times 3$  (см. таблицу) 4 латинские буквы  $X, Y, L, I$ .

Требуется:

1. Построить и обучить нейронную сеть, которая могла бы решать задачу распознавания символов.
2. Произвести тестирование нейронной сети при добавлении шума.

| X |   |   | Y |   |   | I |   |   | L |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

В соответствии с таблицей входной сигнал для нейронной сети может быть представлен в виде развёрнутого раstra – вектора длиной 9. Например, для буквы X это 101010101.

Теперь определимся с выходами нейронной сети. Очевидно, что для распознавания образов нейронная сеть должна иметь возможность формировать столько выходных сигналов, сколько образов она должна уметь распознавать.

В нашем случае таких образов четыре, поэтому возможны два варианта представления выходных данных нейронной сети:

1. Выходной слой с двумя нейронами (выходами), т.е. каждому символу ставится в соответствие двухпозиционный двоичный код.
2. Выходной слой с четырьмя нейронами (выходами), т.е. каждому символу свой выход.

Предлагается выбрать любой вариант.

Обучение нейронной сети:

Набор обучающих пар, используемых для обучения нейронной сети, составляется с учётом того, какой вариант формирования выходного слоя выбран в предыдущем разделе. Если выбран вариант с двумя выходами – каждой букве ставится в соответствие двухпозиционный

двоичный код, то выходной слой выглядит следующим образом:  
 $X - 00$ ,  $Y - 01$ ,  $I - 10$ ,  $L - 11$ .

Если выбран вариант с четырьмя выходами, то выходной слой такой:  
 $X - 0001$ ,  $Y - 0010$ ,  $I - 0100$ ,  $L - 1000$ .

После того, как набор обучающих пар создан, необходимо обучить нейронную сеть и проверить, насколько корректно она решает поставленную задачу.

Проверка работы нейронной сети:

После качественного обучения нейронной сети, следует внести в исходные данные некоторый шум. Например, вместо раstra буквы  $I - 010010010$  попробуйте подать  $010110010$  и посмотреть: удастся ли нейронной сети распознать символ, несмотря на внесённые в данные шум.

#### **Список источников:**

1. Осовский С. Нейронные сети для обработки информации / Пер. с польск. И. Д. Рудинского. - М.: Финансы и статистика, 2004. - 344 с.

2. Харитонов А. А. Нейрокомпьютерные системы: методические указания к контрольной работе для специальностей / ПГУТИ. - Самара: ПГУТИ, 2009. - 69 с.

3. Глубокое обучение и нейронные сети с Python и Pytorch, введение. Часть I [Электронный ресурс]. URL: <https://pythonist.ru/glubokoe-obuchenie-i-nejronnye-seti-s-python-i-pytorchvvedenie-chast-i/> (дата обращения: 27 января 2024).

4. PyTorch [Электронный ресурс]. URL: <https://pytorch.org/> (дата обращения: 27 января 2024).

5. Нейронные сети на основе радиально-симметричных функций [Электронный ресурс]. URL: <https://neuronus.com/theory/nn/954-nejronnye-seti-na-osnove-radialnosimmetrichnykh-funktsij.html> (дата обращения: 27 января 2024).