

Задание 5

Задача 1

Разработайте программу, обучающую нейронную сеть с одним скрытым слоем методом обратного распространения ошибки.

При создании сети задаётся число нейронов в каждом слое, в том числе число входов (N), и выходов (M).

Обучающая последовательность подготавливается в виде двух массивов NumPy, в первом каждая строка которого содержит N чисел, соответствующих значению входов сети, а во втором – M чисел, соответствующих правильному значению выхода.

Для обучения пользователь задаёт скорость обучения и число итераций.

По завершении обучения:

- 1) печатается текущее значение ошибки (общее среднеквадратическое отклонение по всей обучающей последовательности);
- 2) строится график, показывающий суммарное значение ошибки сети на всех итерациях в процессе обучения.

Попробуйте обучить двухслойную сеть распознавать операцию XOR, используя архитектуру, показаную в лекциях.

Указание

Используйте массивы NumPy и операции с ними для реализации прямого и обратного расчёта.

Для чтения и записи файлов используйте функции numpy.

Обратите внимание, что в лекциях предлагалось для хранения значений производных в узлах использовать диагональные матрицы вот такого вида:

$$D_2 = \begin{pmatrix} o_1^{(2)}(1-o_1^{(2)}) & 0 & \dots & 0 \\ 0 & o_2^{(2)}(1-o_2^{(2)}) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & o_m^{(2)}(1-o_m^{(2)}) \end{pmatrix},$$

а затем умножать их на ошибку для вычисления «обратно распространённой ошибки» с помощью обычного матричного умножения $\delta^{(2)} = D_2 e$.

При реализации этого вычисления в NumPy более разумным будет хранить производные в векторе $D_2 = [o_1^{(2)}(1-o_1^{(2)}), \dots, o_m^{(2)}(1-o_m^{(2)})]$ и умножать его потом на вектор ошибок поэлементным умножением:

```
delta2 = D2 * e # код на Python
```

Результат этих операций будет эквивалентным, однако памяти для хранения потребуется меньше, а синтаксис умножения будет проще. В лекциях был использован матричный формат записи, поскольку в линейной алгебре нет операции поэлементного умножения векторов, реализованной в NumPy.

Задача 2

Доработайте программу, так, чтобы она использовала контрольную выборку:

- 1) добавьте в цикл обучения расчёт ошибки на контрольной выборке, полученная ошибка должна только запоминаться для рисования графика, но никак не использоваться в обучении;
- 2) после обучения печатайте кроме ошибки на обучающей выборке ошибку на контрольной выборке;
- 3) добавьте график изменения ошибки на контрольной выборке к графику ошибки на обучающей выборке.

Для проверки работы программы сгенерируйте искусственные данные путем добавления шума к математической функции на ваш выбор.

Постройте график, на котором будут отображены:

- 1) исходная функция без шума;
- 2) точки обучающей и контрольной выборки;
- 3) вычисляемая обученной сетью функция.

Оценка задания

Задача 1: 65 %.

Задача 2: 35%.