# Лабораторная работа №1 **Реализация метода обратного распространения ошибки для двухслойной полностью связанной нейронной сети**

Дисциплина: Современные нейросетевые технологии

## Цель:

Настоящей работы состоит в том, чтобы изучить метод обратного распространения ошибки для обучения глубоких нейронных сетей на примере двухслойной полностью связанной сети (один скрытый слой).

## Задачи:

Выполнение практической работы предполагает решение ***следующих задач***:

1. Изучение общей схемы метода обратного распространения ошибки.
2. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
3. Проектирование и разработка программной реализации.
4. Тестирование разработанной программной реализации.
5. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

В процессе выполнения лабораторной работы предполагается, что сеть ориентирована на решение задачи классификации одноканальных изображений. Типичным примером такой задачи является задача классификации рукописных цифр. Именно ее предлагается использовать в качестве тестовой задачи на примере набора данных MNIST [1].

Метод обратного распространения ошибки разрабатывается, исходя из следующих предположений:

1. На входе сети имеется 𝑤×ℎ нейронов, что соответствует разрешению изображения.

2. На выходе сети имеется 𝑘 нейронов, что соответствует количеству классов изображений.

3. Скрытый слой содержит 𝑠 нейронов.

4. В качестве функции активации на втором слое используется функция softmax.

## Последовательность выполнения работы :

Работа предусматривает выполнение следующей последовательности действий:

1. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
2. Подготовка пошагового описания метода обратного распространения ошибки с выводом всех математических формул для сети, описанной в разделе «Задачи». Примечание: для удобства может быть подготовлен псевдокод.
3. Проектирование и разработка программной реализации метода обратного распространения ошибки.
4. Разработка тестового приложения для классификации рукописных цифр на примере базы MNIST. Для работы с матрицами и векторами можно воспользоваться типом данных **Mat** библиотеки OpenCV [2], а для чтения изображений – функцией **imread** указанной библиотеки. Примечание: приложение должно обеспечивать обучение и тестирование сети, получая на вход пути до директорий с соответствующими выборками. Также в качестве параметров приложение должно принимать количество итераций, выполняемых в ходе обучения, и значение параметра скорости обучения.
5. Обучение сети на тренировочной выборке и тестирование на тестовой выборке набора данных MNIST. Измерение ошибки классификации и сбор результатов экспериментов при разном наборе параметров метода в отчет по работе. Примечание: обратите внимание, что на официальной странице набора данных MNIST опубликованы результаты экспериментов на разных конфигурациях полносвязных нейронных сетей, полученные результаты должны соотносится с опубликованными.
6. Подготовка краткого описания программной реализации, инструкции и по сборке и запуску приложения и результатам экспериментов. Примечания: приветствуется, если вместо инструкции будут разработаны скрипты для автоматической сборки и запуску.

## Формулы для вычисления градиентов функции ошибок по параметрам нейроной сетии формул коррекции весров

Для вычисления градиентов функции ошибки по параметрам нейронной сети мы будем использовать метод обратного распространения ошибки (backpropagation). Этот метод основан на правиле дифференцирования сложной функции и позволяет вычислить градиенты по всем параметрам сети.

Пусть у нас есть нейронная сеть с функцией активации f(x), которая принимает на вход вектор x и выдает на выходе вектор y. Пусть также у нас есть функция ошибки E(y, t), где y - выход сети, а t - правильный ответ. Хотим вычислить градиенты функции ошибки по всем параметрам сети.

Шаг 1: Вычисляем градиент функции ошибки по выходу сети:

dE/dy = (y - t)

Шаг 2: Вычисляем градиент функции активации по входу сети:

df/dx = f(x) \* (1 - f(x))

Шаг 3: Вычисляем градиент функции ошибки по входу сети:

dE/dx = (y - t) \* f(x) \* (1 - f(x))

Шаг 4: Вычисляем градиенты функции ошибки по весам и смещениям каждого нейрона в сети, используя цепное правило дифференцирования:

dE/dw = dE/dx \* x

dE/db = dE/dx

где w - веса нейрона, b - смещение нейрона.

Шаг 5: Обновляем веса и смещения каждого нейрона в сети, используя формулу коррекции весов:

w\_new = w - learning\_rate \* dE/dw

b\_new = b - learning\_rate \* dE/db

где learning\_rate - коэффициент скорости обучения, который определяет, насколько быстро сеть будет менять свои веса в процессе обучения.

Эти шаги повторяются на каждой итерации обучения до тех пор, пока функция ошибки не достигнет минимума.

Следующие критерии

## Описание метода обратного распространения ошибки с выводом всех математических формул для сети, описанной в разделе задачи

Есть пошаговое описание метода с выводом всех математических формул для сети – 2 балла

Есть пошаговое описание метода ошибки без вывода формул для сети – 1 балл

## Разработана программная реализация метода для рассматриваемого частного случая

Программа работает правильно и корректно – 1 балл

## Разработано приложение для решения задачи классификации рукописных цифр на примере базы MNIST

Приложение разработано – 1 балл

## Подготовлено краткое описание разработанного программного кода

Представлена краткая инструкция по сборке программного кода и запуску приложения – 2 балла

Представлена не полная инструкция ( отсутствует или сборка или запуск) – 1 балл

## Подготовлена краткая инструкция по сборке программного кода и запуску приложения на данных базы MNIST

Представлена краткая инструкция по сборке программного кода и запуску приложения – 2 балла

Представлена не полная инструкция ( отсутствует или сборка или запуск) – 1 балл

## Подготовлены результаты классификации для тестового набора данных MNIST

Представлены результаты классификации – 1 балл

## Программная реализация, инструкция по сборке и запуску, описание метода выложены в личный репозиторий на GitHub

Выложены в личный репозиторий на GitHub программная реализация и инструкция по сборке и запуску – 2 балла

Частично выложены в личный репозиторий на GitHub программная реализация и инструкция по сборке и запуску – 1 балл

***Литература***

1. MNIST dataset [http://yann.lecun.com/exdb/mnist]

2. OpenCV [http://opencv.org]