

Лабораторная работа №1

Построение и обучение простейшей нейронной сети

Цель работы – освоить базовые этапы построения и обучения многослойного персептрана (MLP): от подготовки данных и их визуализации до настройки параметров нейросети и анализа качества её работы

Задание: выполнить обучение и предсказание для задачи классификации/регрессии на примере:

1. загрузка исходных данных (загрузить датасет, выполнить визуализацию исходных данных);
2. подготовить данные для загрузки в модель (заполнить пропуски, выполнить замену текстовых категориальных признаков, масштабировать значения при необходимости);
3. построить модель MLP (`from sklearn.neural_network import MLPClassifier`) и выполнить настройку параметров (количество слоев, число нейронов, функции активации);
 - a. * дополнительные баллы при реализации MLP вручную без импорта готовой модели из библиотеки;
4. обучить модель и выполнить оценку ее адекватности при помощи метрик;
5. выполнить визуализацию результатов;
6. составить отчет о проделанной работе в соответствии с требованиями кафедры.

Требования к отчету. Отчет должен содержать постановку задачи, исходные данные, результаты решения задачи, необходимые иллюстративные материалы.

Требования к защите

Защита лабораторной работы происходит индивидуально. Система оценки – рейтинговая.

Критерии оценки:

- корректность выполненного исследования;

- адекватность полученных результатов;
- качество отчета;
- качество ответов на контрольные вопросы;
- срок выполнения работы.

Время выполнения работы – 4 академических часа.

Контрольные вопросы

1. Что такое искусственный нейрон и как он работает?
2. Чем отличается линейная модель от многослойного персептрона?
3. Для чего нужны функции активации?
4. Как выбрать количество эпох и скрытых слоёв?
5. Какие метрики применяются для оценки качества классификации?

Документация: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Исходные параметры для обучения модели:

1. функция активации – ReLU, алгоритм оптимизации – Adam;
2. функция ошибки – softmax или log-loss, L2-регуляризация весов;
3. обратное распространение ошибки – вычисление градиента ошибки по весам.

За студентом остается выбор и настройка:

- количество слоев и размерность (`hidden_layer_sizes`),
- количество эпох (`max_iter`),
- начальная скорость обучения (`learning_rate_init`),
- случайность (`random_state`).