**Лабораторная работа № 2. Автоэнкодеры**

|  |
| --- |
| **Цель:** научиться применять автоэнкодеры для осуществления визуализации данных и их анализа |

**Общее задание**

1. Используя выборку по варианту, осуществить проецирование данных на плоскость первых двух и трех главных компонент с использованием нейросетевой модели автоэнкодера (с двумя и тремя нейронами в среднем слое);

2. Выполнить визуализацию полученных главных компонент с использованием средств библиотеки matplotlib, обозначая экземпляры разных классов с использованием разных цветовых маркеров;

3. Реализовать метод t-SNE для визуализации данных (использовать также 2 и 3 компонента), построить соответствующую визуализацию;

4. Сравнить полученные результаты с анализом с использованием PCA, сделанным в ЛР №1, сделать выводы;

5. Оформить отчет по выполненной работе, загрузить исходный код и отчет в соответствующий репозиторий на github.

**Задание по вариантам**

| **№ варианта** | **Выборка** | **Класс** |
| --- | --- | --- |
| 1 | seeds.zip | Последняя колонка |
| 2 | hcv+data.zip | Category |
| 3 | exasens.zip | Diagnosis ID |
| 4 | heart+failure+clinical+records.zip | death\_event |
| 5 | wholesale+customers.zip | Region |
| 6 | seeds.zip | Последняя колонка |
| 7 | hcv+data.zip | Category |
| 8 | exasens.zip | Diagnosis ID |
| 9 | heart+failure+clinical+records.zip | death\_event |
| 10 | wholesale+customers.zip | Region |
| 11 | seeds.zip | Последняя колонка |
| 12 | hcv+data.zip | Category |
| 13 | exasens.zip | Diagnosis ID |
| 14 | heart+failure+clinical+records.zip | death\_event |
| 15 | wholesale+customers.zip | Region |
| 16 | seeds.zip | Последняя колонка |
| 17 | hcv+data.zip | Category |
| 18 | exasens.zip | Diagnosis ID |
| 19 | heart+failure+clinical+records.zip | death\_event |
| 20 | wholesale+customers.zip | Region |

**Важные замечания**

1. При реализации автоэнкодера архитектуру (количество слоев, нейронов, функции активации), а также параметры обучения подбирать экспериментально (рекомендовано начинать с более простых архитектур, использовать функции активации ReLU (особенно в случае глубоких НС), а также оптимизатор Adam);

2. Рекомендуется использовать вариант реализации метода t-SNE, включенный в пакет **sklearn** (**sklearn.manifold.TSNE**);

3. Для t-SNE рекомендуется выполнить визуализацию при различных значениях перплексивности (**perplexity**) и отобрать наиболее презентативную (рекомендуется выбирать значение данного параметра из диапазона от 20 до 60);

4. Для начальной инициализации t-SNE можно использовать метод PCA (опция **init**).

**Ссылки, дополнительные материалы**

1. Описание метода t-SNE из пакета sklearn:

<https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>

2. Разборная статья по t-SNE для тех, кто хочет знать больше о методе:

<https://habr.com/ru/articles/267041/>

3. Сверточный автоэнкодер (пример в лекции по мотивам, для этой работы нужен обычный полносвязный, так что используем линейные слои torch с последующим нелинейным преобразованием):

<https://medium.com/@syed_hasan/autoencoders-theory-pytorch-implementation-a2e72f6f7cb7>