**Постановка задачи**

Источник: https://opendata.mkrf.ru/opendata/7705851331-stat\_museum

Данные о музеях России: размер коллекции, график работы и т.п.

Структура:

name - Название музея

base\_fond - Число предметов основного фонда на конец года

science\_fond - Число предметов научно-вспомогательного фонда на конец года

ams - Автоматизированная музейная система: число внесенных музейных предметов

n\_buildings - Число строений

n\_days - Число дней в году, открытых для посещения

guests - Число посещений выставок, экспозиций + экскурсионных посещений, тысяч человек

employees - Численность работников, человек

Задача: вычисление значения параметра guests на основании остальных параметров.

**Подход к обработке данных.**

Ко всем столбцам кроме base\_fond , science\_fond , guests было применено MinMaxScaler().fit\_transform(data).flatten()

**Name** – не стал его обрабатывать так как в нем уникальные и независимые значение.

**base\_fond** –Применил логарифм. Распределение было нормальным, применил tandardScaler().fit\_transform(data).flatten()

**science\_fond** – в данных имели выбросы, применил np.clip(data, 0, 200000). После среза заменил нули на среднее число, так как нули — это явно отсутствующие значения. Был применен логарифм.

**Ams** – в данных был выброс, применил np.clip(data, 0, 300000). Применил квадратный корень.

**n\_buildings** – В данных были выбросы применил np.clip(data, 0, 40). Заменил нули на среднее значение, у любого музея есть здание. Применил квадратный корень.

**n\_days** – Были выбросы применил np.clip(data, 200, 400) . Использовал логарифм.

**Guests** – В данных были выбросы, применил np.clip(data, 1, 400). Использовал логарифм и заменил нули на среднее значение.

**Employees** – В данных были выбросы использовал np.clip(data, 0, 500).

Заменил нули на среднее значение. Применил квадратный корень.

**Базовые параметры нейронной сети.**

функция активации - ReLU

Количество нейронов входного слоя – 6

Первый скрытый слой – 6

Второй скрытый слой – 15

Выходной слой – 1

Количество эпох – 100

Батчей – 16

**Описание проведенных экспериментов.**

Первая модель работала отлично, но были не очень плавные графики. Я дал больше свободы нейронной сети в первом слое и сильнее ее начал фильтровать во 2. Во 2 модели я сделал решил сначала лучше отфильтровать данные, а потом уменьшить количество нейронов. В 1 слое 20 нейронов, во 2 слое 10 нейронов, 50 эпох. Результат почти не изменился, но графики стали более резкими. В 3 модели я добавил 3 слой. 15 нейронов в первом слое, 40 нейронов во 2 слое, 20 нейронов в 3 слое. Результат стал лучше, графики были достаточно далеко друг от друга.

**Итоговый вывод.**

Увеличение числа нейронов относительно предыдущего слоя позволяет лучше обрабатывать отдельные примеры. Уменьшение числа нейронов относительно предыдущего слоя позволяет обобщать информацию и выводить новые правила. Увеличение числа слоев помогает сети работать как более сложная функция. Количество нейронов в выходном слое должно быть равно выходных параметров. Параметры обучения нейронной сети – количество эпох и размер батча.