**1. 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.

**2. Подход к обработке данных.**

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

base\_fond – Данные были неравномерно распределены, к данным был применен логарифм. Получилось почти нормальное распределение, поэтому к данным применил функцию StandardScaler().

science\_fond – в данных имели выбросы, решил это проблему “отрезав” выброс. После среза заменил нули на среднее число, так как нули это явно отсутствующие значения. Распределение неравномерное. Был применен логарифм так так логарифм показал график к более близкому нормальному распределению. После все манипуляций применил формулу MinMaxScaler().

Ams – в данных был выброс в районе 300000, выброс был срезан. Распределение неравномерное, большое количество нулей. Применил квадратный корень и функцию MinMaxScaler(). Пытался заменить нули на среднее значение, но это дало еще более плохие показатели.

n\_buildings – В данных были выбросы, поэтому данные были “срезаны”. Распределение неравномерное. Заменил нули на среднее значение, у любого музея есть здание. Применил квадратный корень и функцию MinMaxScaler(). Логарифм дал более плохой график.

n\_days – Данные были распределены не равномерно, использовал логарифм и MinMaxScaler().

Guests – В данных были выбросы, выбросы “обрезал”. Применил логарифм и заменил нули на среднее значение. Данные были почти близки к нормальному распределению, но применил MinMaxScaler().

Employees – В данных были выбросы которые были “срезаны”. Заменил нули на среднее значение, у любого музея есть работники. Применил квадратный корень, он дал лучше график чем логарифм. К данным применил функцию MinMaxScaler().

**3.Базовые параметры нейронной сети.**

функция активации - ReLU

Количество нейронов входного слоя – 7

Первый скрытый слой – 70

Второй скрытый слой – 25

Третий скрытый слой – 10

Выходной слой – 1

Количество эпох – 100

Батчей – 16

**4. Описание проведенных экспериментов.**

Решил для всех моделей использовать 3 слоя.

Первая модель работал корректно, требует большого количество нейронов, графики среднеквадратического отклонения и среднего абсолютного отклонения

Немного не “плавные”.

Для второй модели я выбрал активационную функцию сигмоидную, данная функция должна сгладить графики и она требует меньшего количества нейрнов.

Для нее в первом и третьем скрытом слое поставил 30 нейронов, во втором поставил 15 нейронов. График стал более плавным, среднеквадратичная ошибка на тестовой выборки незначительно меньше чем на первой модели. Relu требует большого количества нейронов, relu требует 2536 параметра а функция сигмоидная требует всего 881

параметр.

В третьей модели я использовал активационную функцию Softmax. Часто Softmax используется для последнего слоя глубоких нейронных сетей для задач классификации.

Первый и третий скрытый слой – 10 нейронов, второй скрытый слой 5 нейронов.

Точно не изменилась, но количество необходимых параметров для обучения сильно уменьшилось необходимо всего 196. Графики наиболее плавные у 3 модели.

**5. Итоговый вывод.**

Мой набор данных не очень большой и использовать на нем активационную функцию Relu не нужно, relu требует большого количество нейронов, функция очень медленная.

Функция не линейна и функция представляет из себя горизонтальную линию, некоторые нейроны могут из-за этого не работать.

Вторая модель лучше реагирует на “шероховатости” данных. У краев функции реакции функции на изменение “X” слабее, это значит, что некоторые предсказания могут быть занижены.

Softmax решил просто попробовать, она отлично подходит для многомерного случая. Данная функция за наименьшее количество параметров показала наиболее хорошие графики и среднеквадратичная ошибка как у 2 модели. Softmax не требует большого количества нейронов, функция устойчивая из-за чего ее часто используют в связке с Relu.