**1. Постановка задачи**

<https://www.kaggle.com/fernandol/countries-of-the-world>  
Статистическая информация о странах мира.

Структура:  
country - Страна  
region - Регион  
population - Население  
area - Площадь  
coastline - Отношение длины побережья к площади  
infant\_mortality - Детская смертность (на 1000 рождений)  
gdp - ВВП на душу населения  
literacy - Грамотность  
phones - Количество сотовых телефонов на 1000 человек  
arable - Процент территории, используемой под сельское хозяйство  
birthrate - Рождаемость  
deathrate - Смертность

Задача: вычисление значения параметра population на основании остальных параметров.  
Опциональный вариант: вычисление значения параметра region на основании остальных параметров.

**2. Подход к обработке данных.**

Не взял в финальный набор данных 2 колонки, а именно country и coastline, phones. Страны не повторяются и нет связи с другими столбцами, Побережье у большинства стран 0, телефон не влияет на популяцию.

Ко всем столбцам была применена функция MinMaxScaler().fit\_transform(data).flatten() так как в наших данных не было нормального распределения.

Во всех столбцах заменил нули на среднее значение. Очевидно, что в статистике сран не может быть нулевое значение.

Region –я применил get\_dummies так как нейронная сеть может понимать только числовые значения. Затем я соединил данные в 1 столбец и применил квадратный корень. Квадратный корень был ближе к нормальному распределению.

Population – обрезал до 93973713 так как были гигантские выбросы. Применил логарифм так как он дает график, который ближе к нормальному распределению.

Area – был выброс срезал данные до 1275200. Применил логарифм так как он дает график, который ближе к нормальному распределению.

Gdp – Был выброс, срезал до 38000. Применил квадратный корень так как он дает график, который ближе к нормальному распределению.

Literacy , infant\_mortality, arable, birthrate были обработанные одинаково.

К ним было применено data.fillna(0), для замены nan на 0. Потом применили data.apply(pd.to\_numeric, errors='coerce') данная функция меняет тип данных столбца на числовой тип, errors='coerce' игнорирует ошибку если данные нельзя привести к числовому значению.

Во всех этих столбцах квадратный корень был лучше логарифма.

После того я обработал все данные применил dataset[‘col\_name’].fillna(dataset[col\_name’].mean()) что бы точно убрать все nan.

**3.Базовые параметры нейронной сети.**

функция активации - ReLU

Количество нейронов входного слоя – 7

Первый скрытый слой – 25

Второй скрытый слой – 5

Выходной слой – 1

Количество эпох – 100

Батчей – 16

**4. Описание проведенных экспериментов.**

Решил во всех моделях изменять только количество нейронов в 2 слоях и количество батчей. Первая модель работало корректно. Попробовал во 2 модели увеличь число батчей. График не сошлись тогда я увеличил число нейронов в первом слое 40 нейронов, во 2 слое 10 нейронов, количество батчей – 32. Графики сошлись точно модели почти не изменилась. В 3 модели уменьшил количество нейронов до 8, в 1 слое 6 нейронов, во 2 слое 10 нейронов. Точно модели незначительно увеличилась.

**5. Итоговый вывод.**

Нельзя пропустить через нейронную сеть разом весь датасет. Поэтому делим данные на пакеты, сеты или партии, так же, как большая статья делится на много разделов — введение, градиентный спуск, эпохи, Batch size и итерации. Из этого следует что можно разделить полный датасет из 100 объектов на батчи размером 20 объектов. Таким образом, для завершения одной эпохи потребуется 5 итерации. Если у нас будет большой батч и малый объём данных нам потребуется больше эпох для обучения или больше нейронов. Если батч слишком маленький изменения могут быть незначительны для большого датасета.