**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 итоговый набор данных колонки country, coastline, phones, так как у них нет связи с другими столбцами.

Применил функцию MinMaxScaler().fit\_transform(data).flatten() потому что, среди моих данных не было нормального распределения.

Заменил значение 0 во всех столбцах на среднее, так как в статистиках стран не может быть нулевых значений.

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.

Population – из-за больших выбросов решил обрезать до 93973713.

Area – был выброс, срезал данные до 1275200.

Gdp – Был выброс, срезал до 38000.

Literacy , infant\_mortality, arable, birthrate были обработанны 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 и 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 – 8

Первый скрытый слой – 30

Второй скрытый слой – 5

Выходной слой – 1

Количество эпох – 100

Батчей – 16

**4. Описание проведенных экспериментов.**

Изменил количество нейронов и батчей первого и второго слоев. Все модели в итоговом счете сходились. Во второй модели увеличил размер второго слоя с 5 до 9, первый оставил неизменным. В третьем уменьшил количество первого слоя и увеличил количество второго слоя, в итоге график сошелся.

**5. Итоговый вывод.**

Из-за того, что датасет может быть большим и даже огромным, его нельзя разом пустить через нейронную сеть. Поэтому следует поделить данные на небольшие группы, партии или же пакеты. Из этого можно сделать вывод, что дадасет из 100 объектов можно разделить на батчи размером 20 объектов. Тем самым, для завершения одной эпохи понадобится 5 итераций. Если будет большой батч и небольшой объем данных, то потребуется больше эпох или нейронов для обучения нейронной сети.