**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 3 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.

К столбцам region, gdp, literacy, infant\_mortality, arable, birthrate был применен квадратный корень. Квадратный корень показывал распределение ближе к нормальному или были нули.

Во всех столбцах заменил нули на среднее значение. Очевидно, что в статистике стран 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.

Выбросы были в столбцах population, area, gdp. Что бы убрать выбросы применил np.clip(data, min, max).

К столбцам Literacy, infant\_mortality, arable, birthrate применил data.fillna(0), 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.

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 – 8

Первый скрытый слой – 10

Второй скрытый слой – 8

Выходной слой – 1

Количество эпох – 50

Батчей – 16

**4. Описание проведенных экспериментов.**

Первая модель работает хорошо с параметрами, указанными в базовых параметрах. Графики сошлись. Во второй модели я добавил скрытый слой. Теперь параметры выглядят так:

Первый скрытый слой – 10

Второй скрытый слой – 6

Третий скрытый слой – 10

Модель работает корректно и график сошлись, Точность почти не изменились.

В 3 модели оставил 1 слой и сделал 26 нейронов. Модель работает корректно и быстро.

**5. Итоговый вывод.**

Увеличение скрытых слоев нейронной сети делает алгоритм нейронной сети более сложным. Время работы сети также увеличивается с количеством слоев. Если у нас малый набор данных и большое количество слоев мы можем переобучить нейронную сеть или сделать ее решение очень специфичным. Чем больше слоев, тем больше нейронной сети требуется параметров.