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.

1. **Подход к обработке данных.**

Не включал столбцы «country», «coastline» и «phones» в обработанный набор данных. Поскольку страны не повторяются и нет связи с другими столбцами, отношение длины побережья к площади особо не интересует и не влияет на популяцию, также, как и наличие телефонов.

* Ко всем столбцам была применена функция «MinMaxScaler().fit\_transform(data).flatten()» так как в данных были ближе к равномерному распределению
* Для столбцов «infant\_mortality», «literacy», «arable», «birthrate», «deathrate» выполнено преобразование в строку, замена символа «,» на «.», а затем преобразование в тип «float».
* В конце обработки данных была применена функция dataset[i] = dataset[i].fillna(dataset[i].mean()), для того чтобы исправить ошибку «not a number» для выше перечисленных столбцов, при обучении нейронной сети
* *Population* – наблюдались выбросы, значения обрезаны до 94000000. Применена функцию логарифма, так как график ближе к нормальному распределению.
* *Area* – был выброс данных, значения урезаны до 1275200
* Применена функцию логарифма, так как график ближе к нормальному распределению.
* *gdp* – был выброс данных, значения урезаны до 38000. Применена функция квадратного корня, так как график ближе к нормальному распределению.
* *literacy* – выбросы в начальных значениях, урезаны в интервал (38, 100) Применена функция квадратного корня, так как график ближе к нормальному распределению.
* *infant\_mortality* - Применена функцию логарифма, так как график ближе к нормальному распределению.
* *arable* - Применена функция квадратного корня, так как график ближе к нормальному распределению.
* *birthrate* - Применена функция квадратного корня, так как график ближе к нормальному распределению.

1. **Базовые параметры нейронной сети.**

Оптимизатор – adam

функция активации - ReLU

Количество нейронов входного слоя – 8

Первый скрытый слой – 50

Второй скрытый слой – 50

Выходной слой – 1

Количество эпох – 100

Батчей – 16

1. **Описание проведенных экспериментов.**

**Оптимизатор – adam**

Первая модель работает правильно.

**Оптимизатор – sgd**

*Вторая модель имеет следующие параметры*:

Количество нейронов входного слоя – 8

Первый скрытый слой – 50

Второй скрытый слой – 30

Выходной слой – 1

Количество эпох – 200

Батчей – 48

Оптимизатор SGD потребовал больше нейронов в каждом слое (при малых значениях наблюдались меньшая сходимость). Обучение происходило тяжелее

Также, при малом размере батча в графиках наблюдались скачки.

**Оптимизатор – Adadelta**

*В третьей модели* использовал аналогичные параметры, что и во второй модели.

В графиках абсолютного и среднеквадратичного отклонения наблюдаются сачки. Остальные графики имеют боле менее нормальный вид.

Обучение с данным оптимизатором происходит легче и быстрее.

*В четвертой модели использовал следующие параметры:*

функция активации - ReLU

Количество нейронов входного слоя – 13

Первый скрытый слой – 6

Второй скрытый слой – 6

Выходной слой – 1

Количество эпох – 50

Батчей – 16

Графики имеют боле менее нормальный вид.

1. **Итоговый вывод.**

Оптимизатор Adam – это стандартный метод, который имеет высокую скорость и точность

Оптимизатор SGD – скорее всего больше подойдёт для использования на сбалансированных и небольших по размеру наборах данных, в которых будет достаточно равномерное их представление. При работе с несбалансированным набором будет наблюдаться низкая сходимость, а также отдельные, данные будут плохо обрабатываться.

Оптимизатор Adadelta также является стандартным и обладает высокой скоростью и точностью. Однако оптимизатор adam, в отличие от данного, имеет боле быструю сходимость.