**Постановка задачи.**  
<https://www.kaggle.com/hugoncosta/price-of-flats-in-moscow>  
Данные о стоимости квартир в Москве в зависимости от параметров квартиры и ее расположения.

Структура:  
price - Цена квартиры в $1000  
totsp - Общая площадь квартиры, кв.м.  
livesp - Жилая площадь квартиры, кв.м.  
kitsp - Площадь кухни, кв.м.  
dist - Расстояние от центра в км.  
metrdist - Расстояние до метро в минутах  
walk - 1 – пешком от метро, 0 – на транспорте  
brick - 1 – кирпичный, монолит ж/б, 0 – другой  
floor - 1 – этаж кроме первого и последнего, 0 – иначе  
code - Число от 1 до 8, при помощи которого мы группируем наблюдения по подвыборкам:  
1. Наблюдения сгруппированы на севере, вокруг Калужско-Рижской линии метрополитена  
2. Север, вокруг Серпуховско-Тимирязевской линии метрополитена  
3. Северо-запад, вокруг Замоскворецкой линии метрополитена  
4. Северо-запад, вокруг Таганско-Краснопресненской линии метрополитена  
5. Юго-восток, вокруг Люблинской линии метрополитена  
6. Юго-восток, вокруг Таганско-Краснопресненской линии метрополитена  
7. Восток, вокруг Калиниской линии метрополитена  
8. Восток, вокруг Арбатско-Покровской линии метрополитена

Задача: вычисление значения параметра price на основании остальных параметров.  
Опциональный вариант: вычисление значения параметра code на основании остальных параметров.

**Подход к обработке данных.**  
price – был выброс в районе (500-750) и не очень равномерное распределение . Были применены метод numpy.clip() , логарифм и MinMaxScaler() fit\_transform(data).flatten() для приведения значений к инвервалу (0,1)

totsp – выброс в районе 190 и не очень равномерное распределние . Были применены метод numpy.clip() , квадратный корень и MinMaxScaler() fit\_transform(data).flatten() для приведения значений к инвервалу (0,1)

livesp – выброс (90-100) и неравномерное распределение . Были применены метод numpy.clip() , логарифм и MinMaxScaler() fit\_transform(data).flatten()

kitsp , dist - были применены только MinMaxScaler() fit\_transform(data).flatten() для приведения значений к инвервалу (0,1)

mertdist – выброс в районе 20 и и не очень равномерное распределение . Были применены метод numpy.clip() , квадратный корень и MinMaxScaler() fit\_transform(data).flatten() для приведения значений к инвервалу (0,1)

walk , brick , floor – принимают только 0 и 1 поэтому были оставлены без изменений

code – было применено унитарное кодирование методом pandas.get\_dummies() т.к. стоблец код это не содержит в себе количественную меру , а атрибутивное свойство в данном случае местоположение квартиры .

**Выбранные базовые параметры нейронной сети и алгоритма обучения.**функция активации - ReLU

Количество нейронов входного слоя – 16

Первый скрытый слой – 20

Второй скрытый слой – 10

Выходной слой – 1

Количество эпох – 50

Батчей – 32

Выборка 0.2

**Описание проведенных экспериментов.**Графики первой модели совпали.

Графики второй модели также совпали. Точность чуть хуже.  
Для второй модели первый слой составляет 15, а второй 5 нейронов. Количество эпох – 25.

Графики третьей модели снова совпали. Точность чуть лучше.  
Для третьей модели первый слой составляет 20, а второй 5 нейронов. Количество эпох – 75.

**Итоговый вывод.**При увеличении числа эпох, увеличивается число раз когда нейронная сеть проходит по дата базе. Нейронная сеть с каждым разом лучше подстраивается под данные, что мы наблюдаем в виде совпадения графиков, переходя последовательно из плохо обученного состояния в оптимальное. Если вовремя не остановить нейронная сеть, то может произойти переобучение - когда нейронная сеть подстроился под конкретный набор данных и перестаёт обобщать.