1. **Постановка задачи.**

**Краткое описание набора данных и решаемой 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.

1. **Подход к обработке данных.**

Price – был выброс и неравномерное распределение.

Totsp – был выброс и неравномерное распределение.

Livesp – был выброс и неравномерное распределение.

Kitsp – потребовалось приведение к интервалу (0, 1).

Dist - потребовалось приведение к интервалу (0, 1).

Metrdist – был выброс и неравномерное распределение.

Walk, brick и floor – изменения не потребовались.

Code – потребовалось унитарное кодирование.

1. **Базовые параметры нейронной сети.**

функция активации - ReLU

Количество нейронов входного слоя – 7

Первый скрытый слой – 70

Второй скрытый слой – 25

Третий скрытый слой – 10

Выходной слой – 1

Количество эпох – 100

Батчей – 16

1. **Описание проведённых экспериментов.**

Первая модель работал корректно, требует большого количество нейронов, графики среднеквадратического отклонения и среднего абсолютного отклонения немного не “плавные”. Для второй модели я выбрал активационную функцию сигмоидную, данная функция должна сгладить графики, и она требует меньшего количества нейронов. Для нее в первом и третьем скрытом слое поставил 30 нейронов, во втором поставил 15 нейронов. График стал более плавным, среднеквадратичная ошибка на тестовой выборке незначительно меньше чем на первой модели. Relu требует большого количества нейронов, relu требует 2536 параметра а функция сигмоидная требует всего 881 параметр. В третьей модели я использовал активационную функцию Softmax. Часто Softmax используется для последнего слоя глубоких нейронных сетей для задач классификации. Первый и третий скрытый слой – 10 нейронов, второй скрытый слой 5 нейронов. Точно не изменилась, но количество необходимых параметров для обучения сильно уменьшилось необходимо всего 196. Графики наиболее плавные у 3 модели.

1. **Итоговый вывод.**

Мой набор данных не очень большой и использовать на нем активационную функцию Relu не нужно, relu требует большого количество нейронов, функция очень медленная. Функция не линейна и функция представляет из себя горизонтальную линию, некоторые нейроны могут из-за этого не работать. Вторая модель лучше реагирует на “шероховатости” данных. У краев функции реакции функции на изменение “X” слабее, это значит, что некоторые предсказания могут быть занижены. Softmax решил просто попробовать, она отлично подходит для многомерного случая. Данная функция за наименьшее количество параметров показала наиболее хорошие графики и среднеквадратичная ошибка как у 2 модели. Softmax не требует большого количества нейронов, функция устойчивая из-за чего ее часто используют в связке с Relu.