

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: Регрессионная модель изменения цен на дома в Бостоне**

Студент гр. 8382

Мирончик П.Д.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

## **ЗАДАНИЕ**

Реализовать предсказание медианной цены на дома в пригороде Бостона в середине 1970-х по таким данным, как уровень преступности, ставка местного имущественного налога и т. д.

Данный набор содержит относительно немного образцов данных: всего 506, разбитых на 404 обучающих и 102 контрольных образца. И каждый признак во входных данных (например, уровень преступности) имеет свой масштаб. Например, некоторые признаки являются пропорциями и имеют значения между 0 и 1, другие — между 1 и 12 и т. д.

## **ЗАДАЧИ**

- Ознакомиться с задачей регрессии
- Изучить отличие задачи регрессии от задачи классификации
- Создать модель
- Настроить параметры обучения
- Обучить и оценить модели
- Ознакомиться с перекрестной проверкой

## **ТРЕБОВАНИЯ**

1. Объяснить различия задач классификации и регрессии
2. Изучить влияние кол-ва эпох на результат обучения модели
3. Выявить точку переобучения
4. Применить перекрестную проверку по  $K$  блокам при различных  $K$
5. Построить графики ошибки и точности во время обучения для моделей, а также усредненные графики по всем моделям

## ХОД РАБОТЫ

В отличие от классификации задача регрессии заключается в предсказании значения некоторой функции, у которой может быть бесконечно много значений (вес человека, стоимость машины и т.п.).

В программу загружаются данные о недвижимости. Затем производится нормализация данных (все признаки центрируются по нулевому значению и имеют стандартное отклонение, равное 1). Строится модель сети:

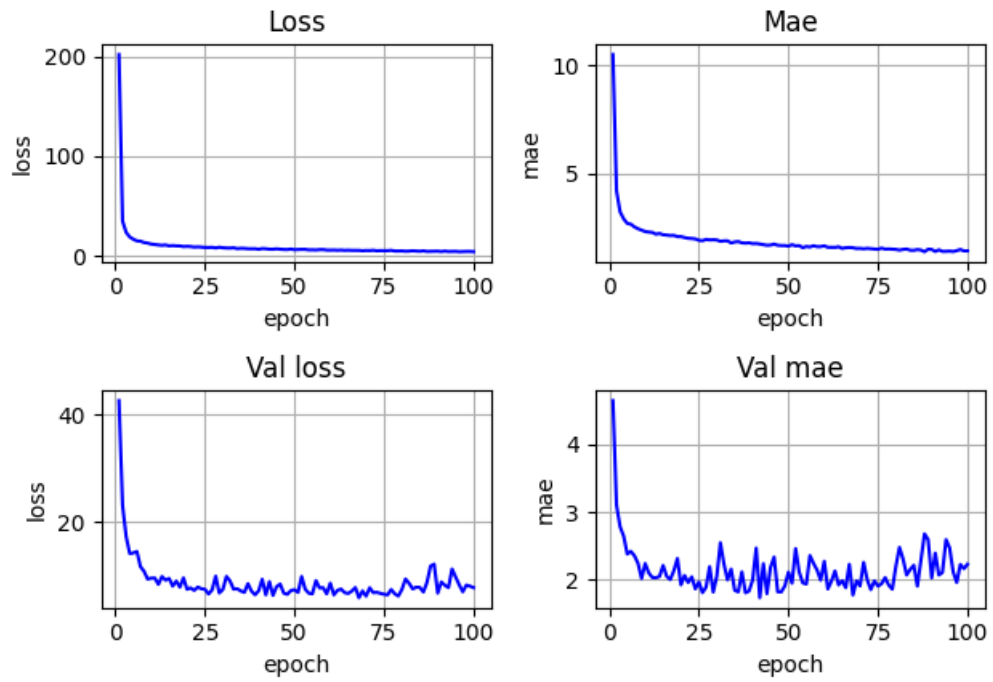
```
model = Sequential()  
model.add(Dense(64, activation='relu',  
input_shape=(train_data.shape[1],)))  
model.add(Dense(64, activation='relu'))  
model.add(Dense(1))  
model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
```

Сеть включает в себя 2 скрытых слоя с функцией активации relu и выходной слой без функции активации для выполнения задачи регрессии. В качестве функции потерь используется функция среднеквадратичной ошибки MSE, для оценки работы нейросети отслеживается величина средней абсолютной ошибки MAE.

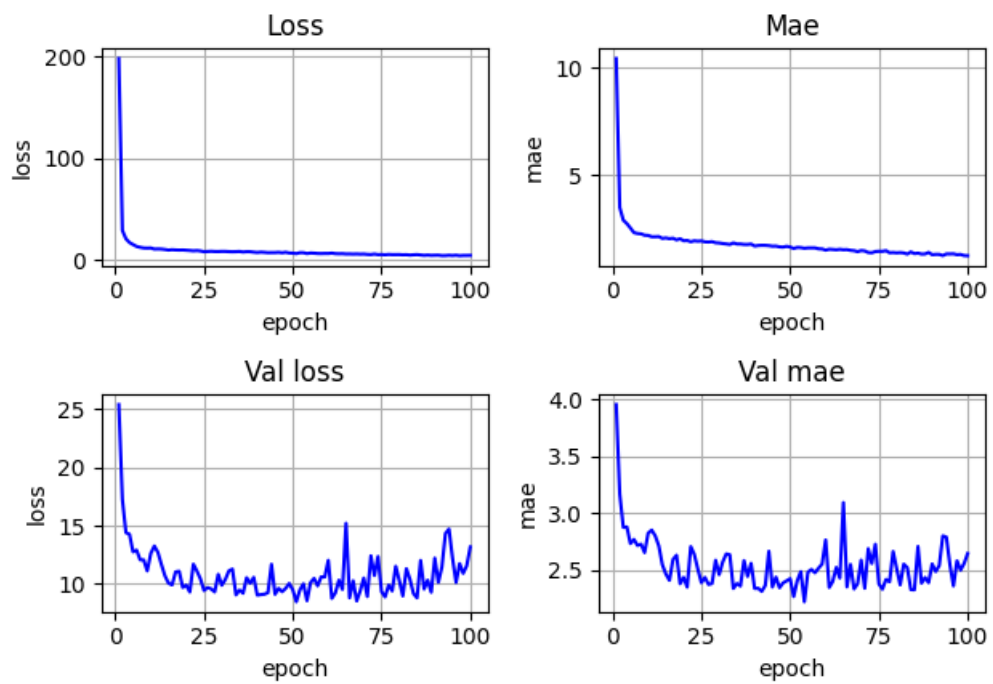
Для более надежной проверки качества модели используется перекрестная проверка по k блокам. Имеющиеся в наличии данные разбиваются на k частей, одна из которых выбирается в качестве валидационной, остальные используются для обучения. Таким образом процедура выполняется k раз с разными валидационными множествами. По полученным результатам составляется среднее значение, которое выбирается за оценку модели.

Проведем перекрестную проверку по 4 блокам на 100 эпохах.

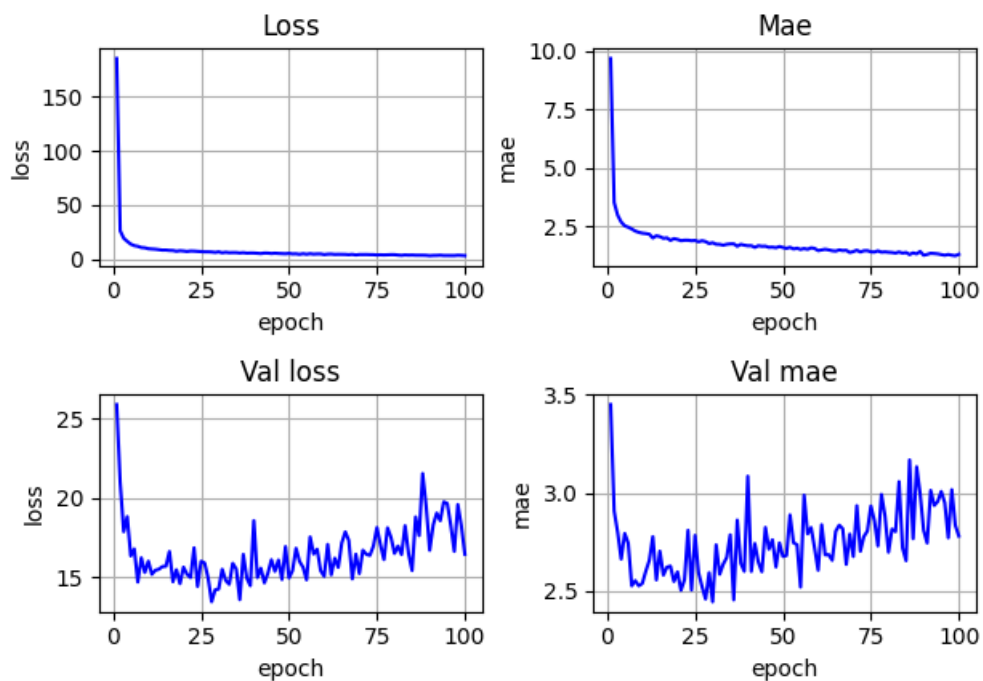
Model 1 of 4. Epochs: 100



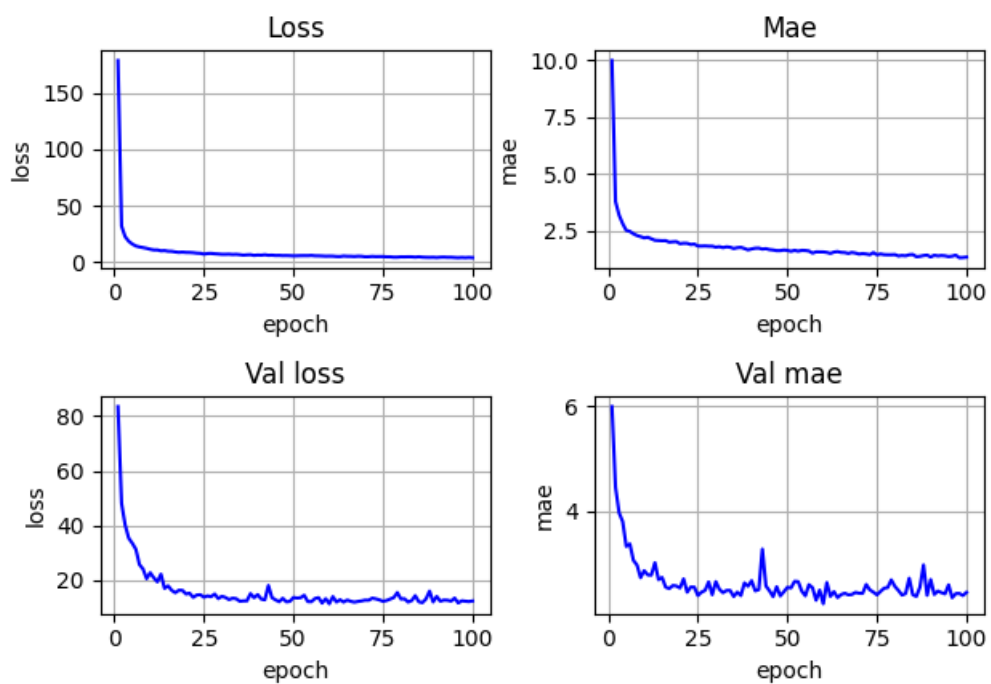
Model 2 of 4. Epochs: 100



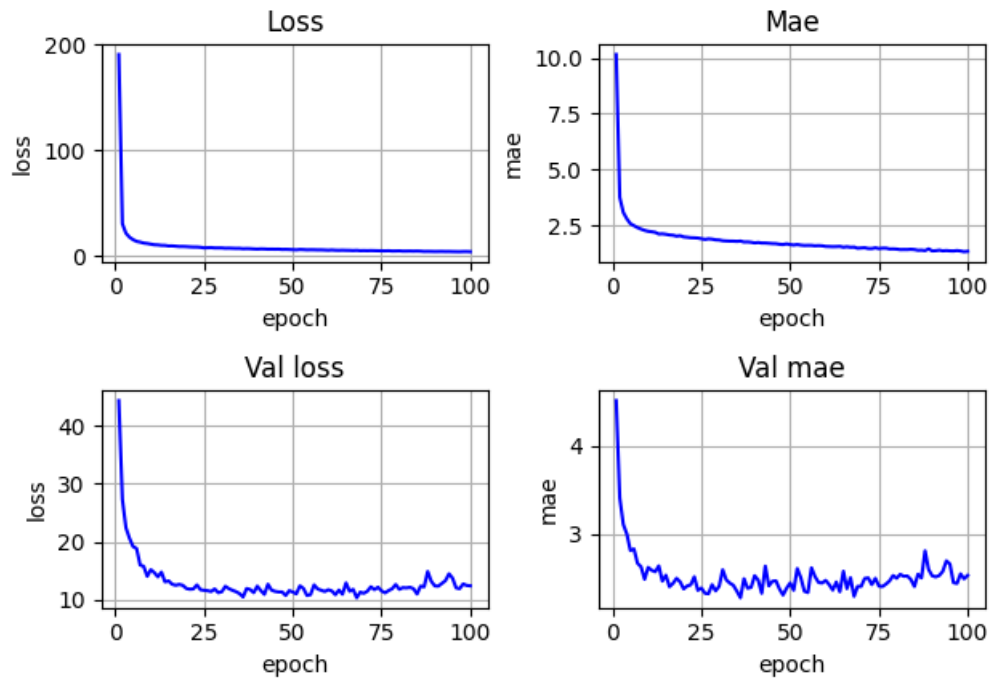
Model 3 of 4. Epochs: 100



Model 4 of 4. Epochs: 100



Mean results. k: 4, epochs: 100

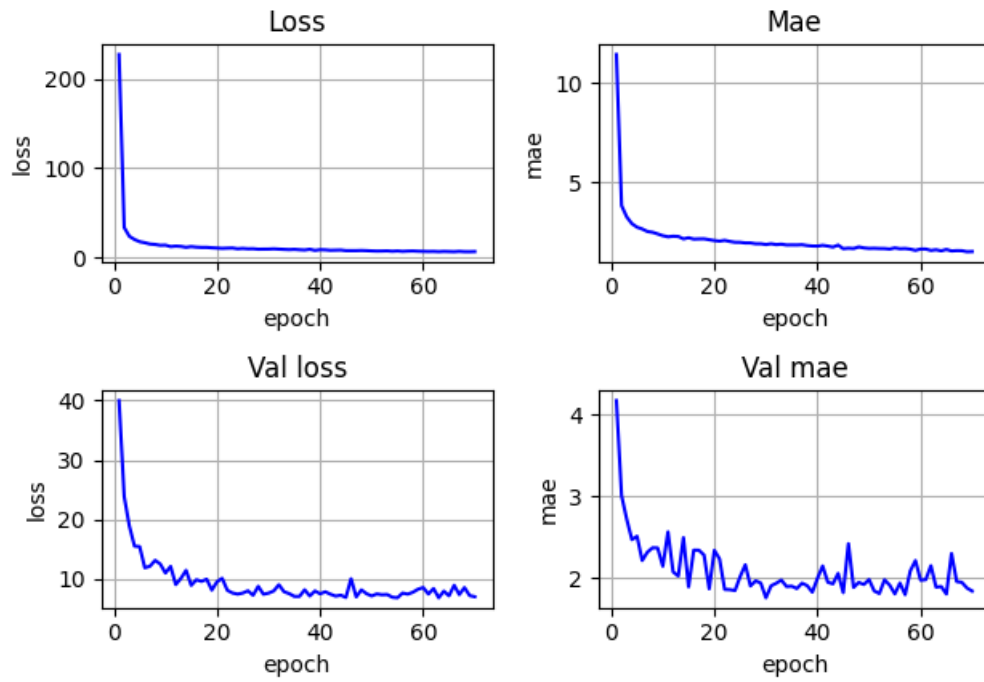


Полученная оценка модели 2.389, т.е. средняя разница между реальной и предсказанной ценой составляет 2389 долларов.

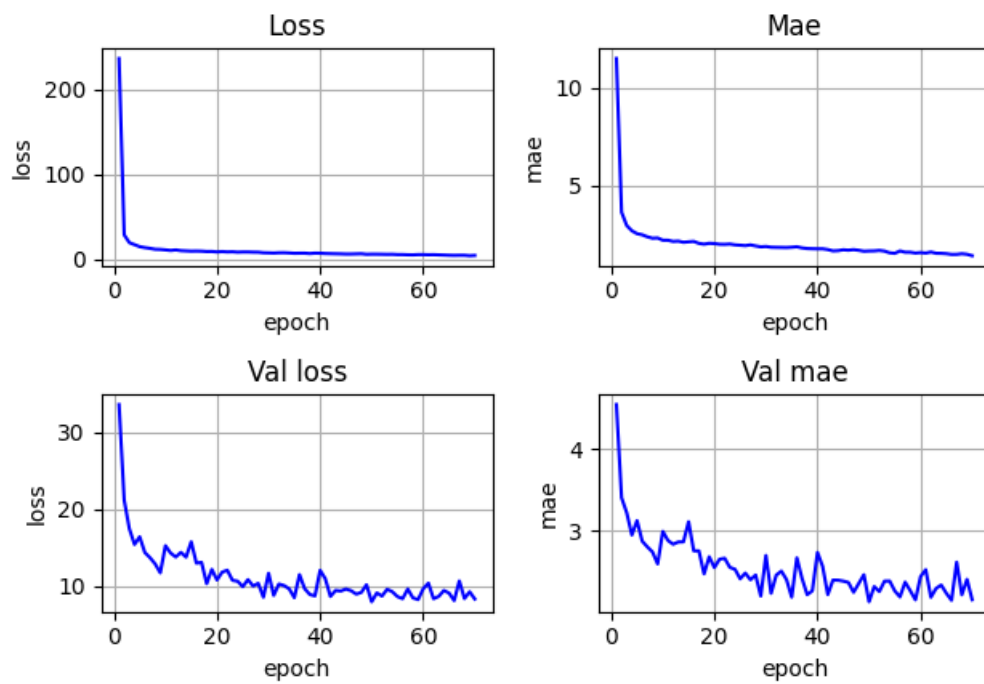
На графиках видно, что примерно после 70-й эпохи средняя абсолютная ошибка на валидационных данных перестает уменьшаться и начинает расти, что свидетельствует о переобучении модели.

Сократим число эпох до 70 и проведем перекрестную проверку по 4 блокам.

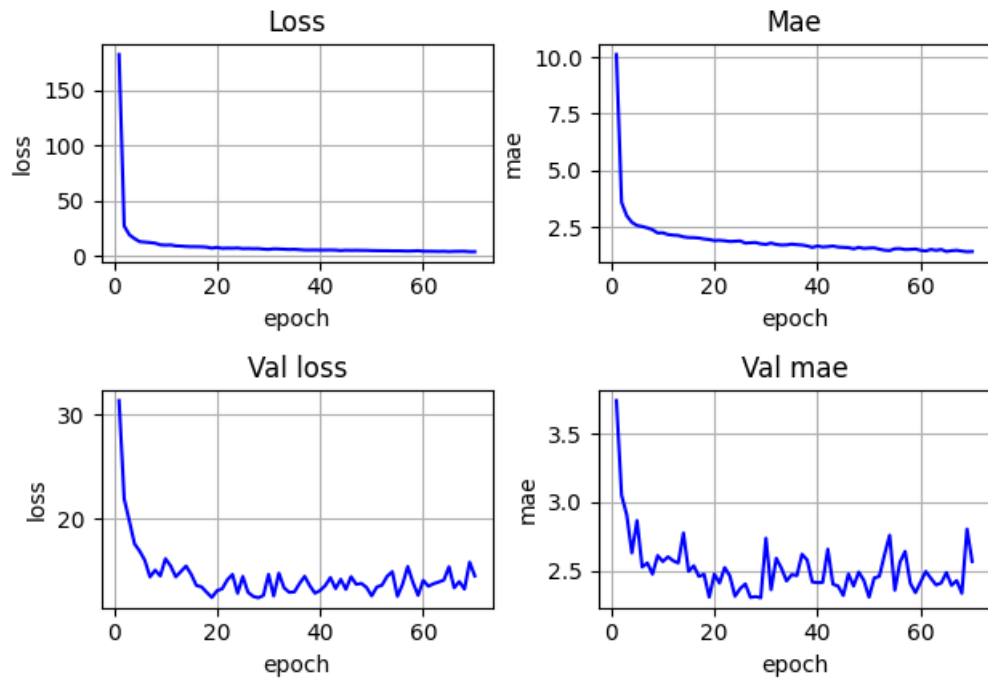
Model 1 of 4. Epochs: 70



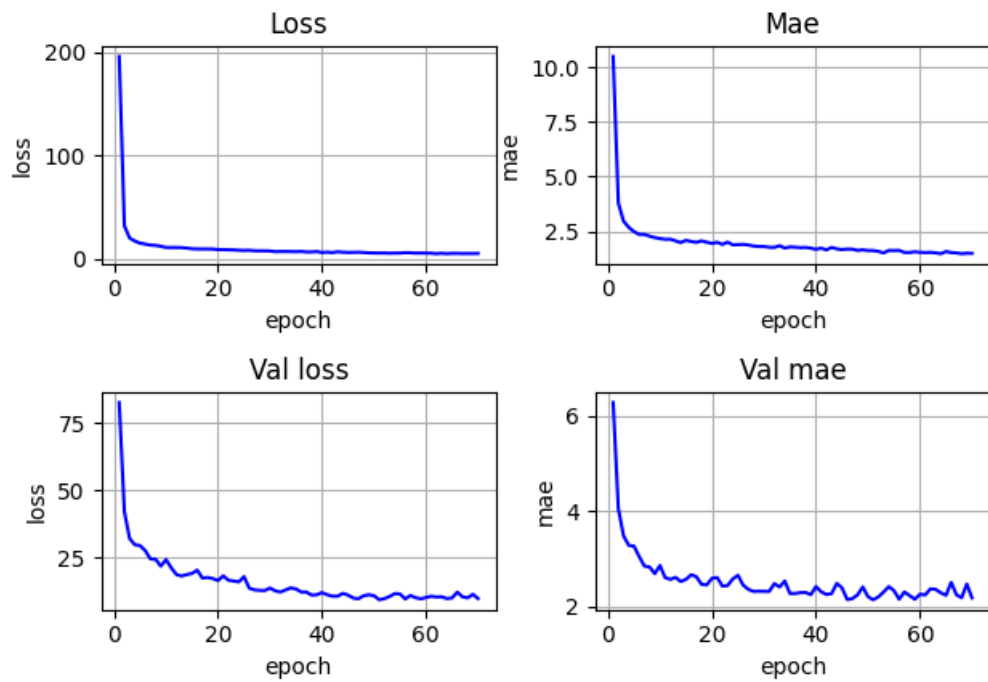
Model 2 of 4. Epochs: 70



Model 3 of 4. Epochs: 70

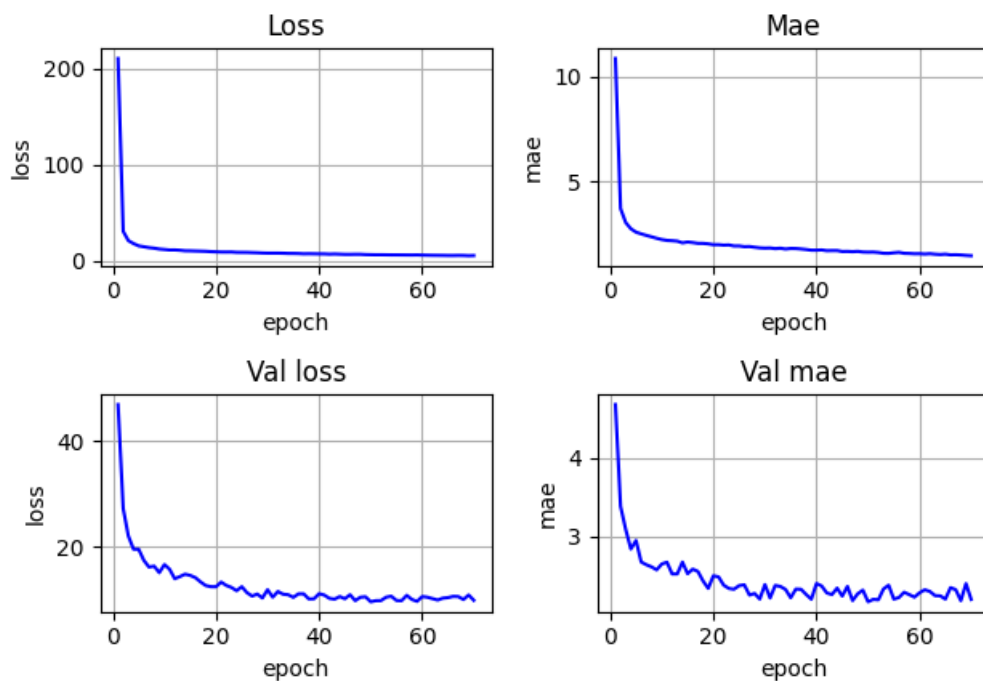


Model 4 of 4. Epochs: 70





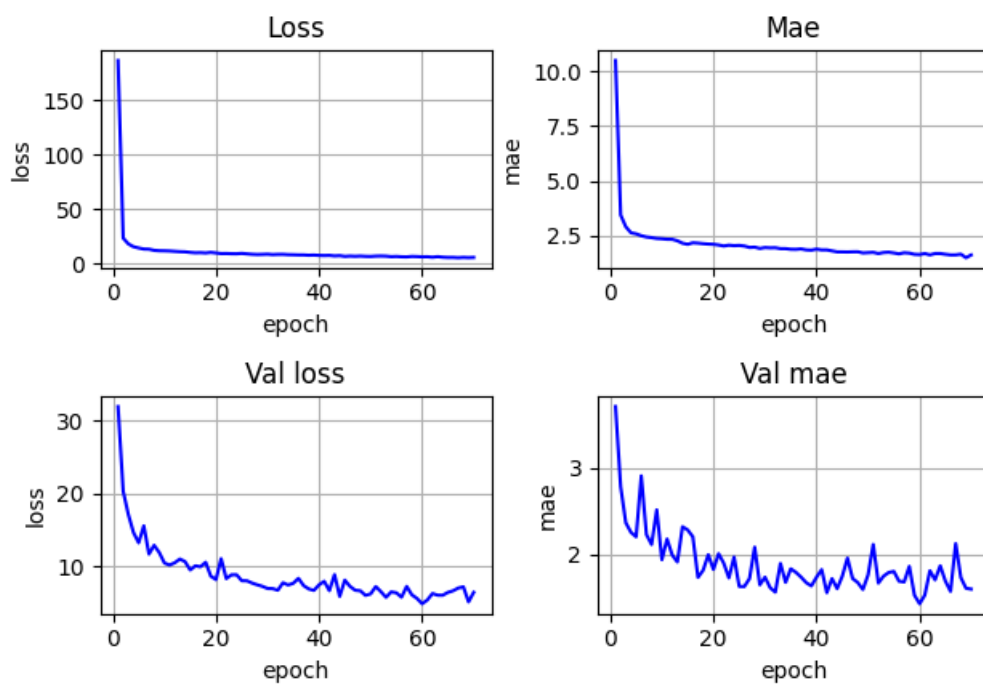
Mean results. k: 4, epochs: 70



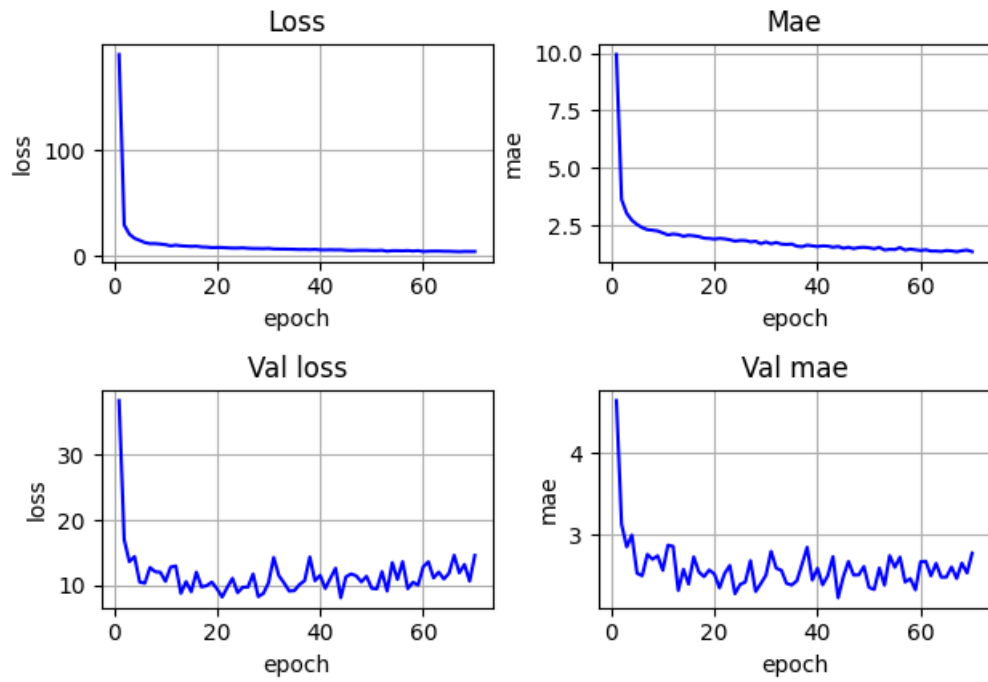
При такой конфигурации среднее значение абсолютной ошибки составило 2.188, что лучше, чем при обучении моделей на 100 эпохах.

Увеличим число блоков k до 6 и проведем обучение на 70 эпохах.

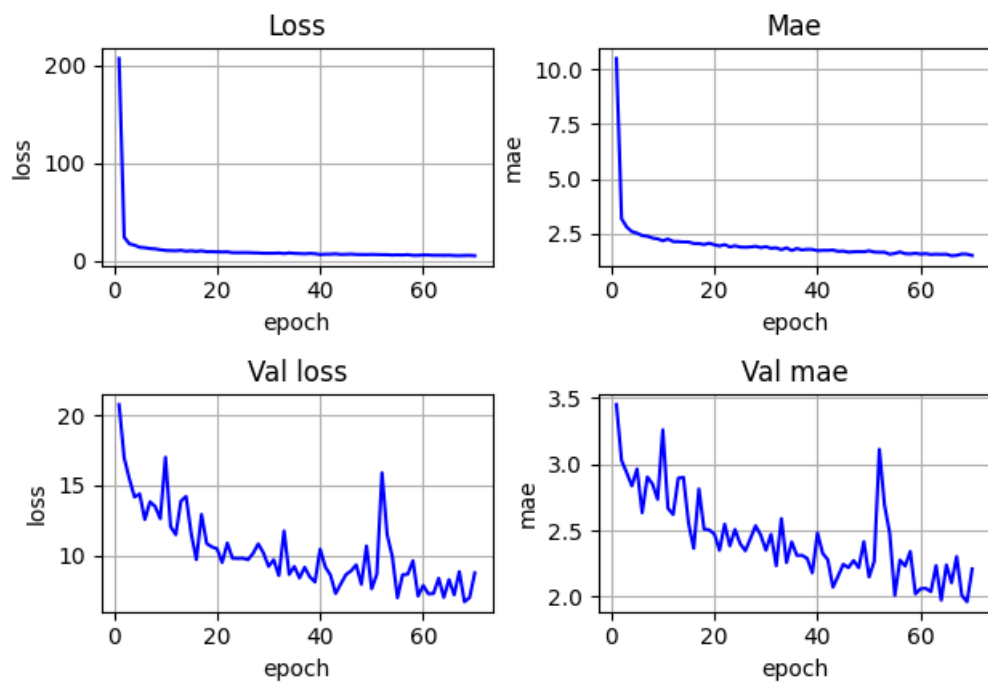
Model 1 of 6. Epochs: 70



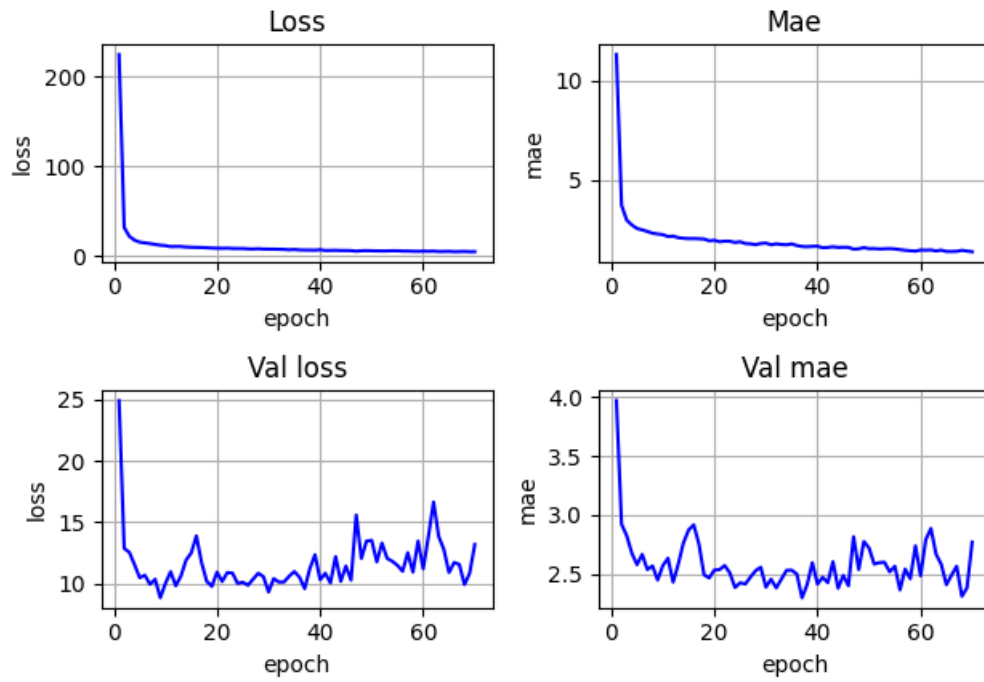
Model 2 of 6. Epochs: 70



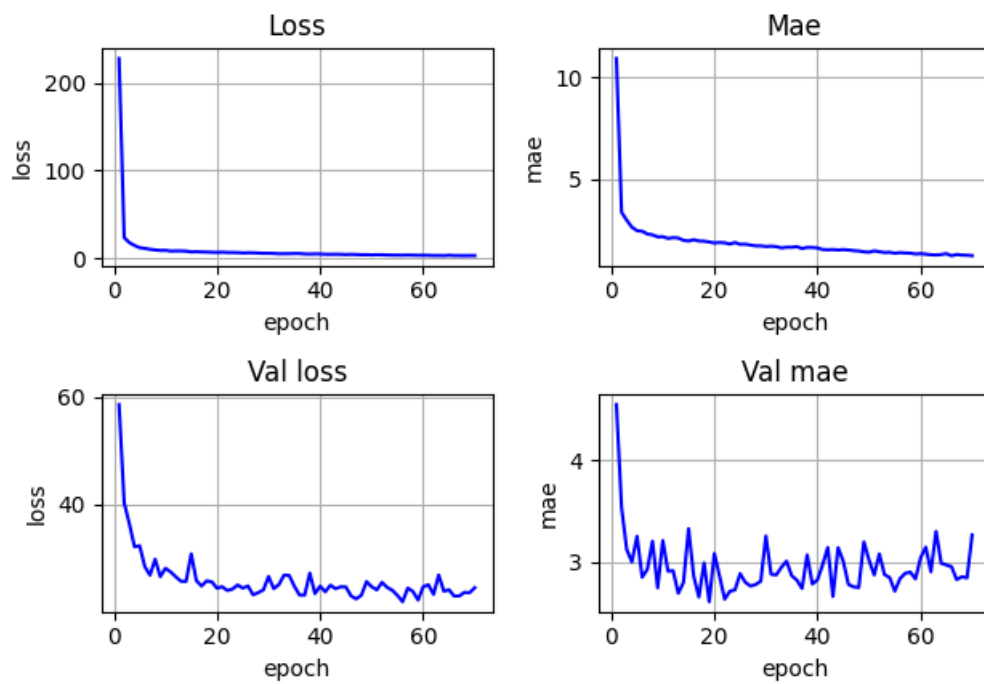
Model 3 of 6. Epochs: 70



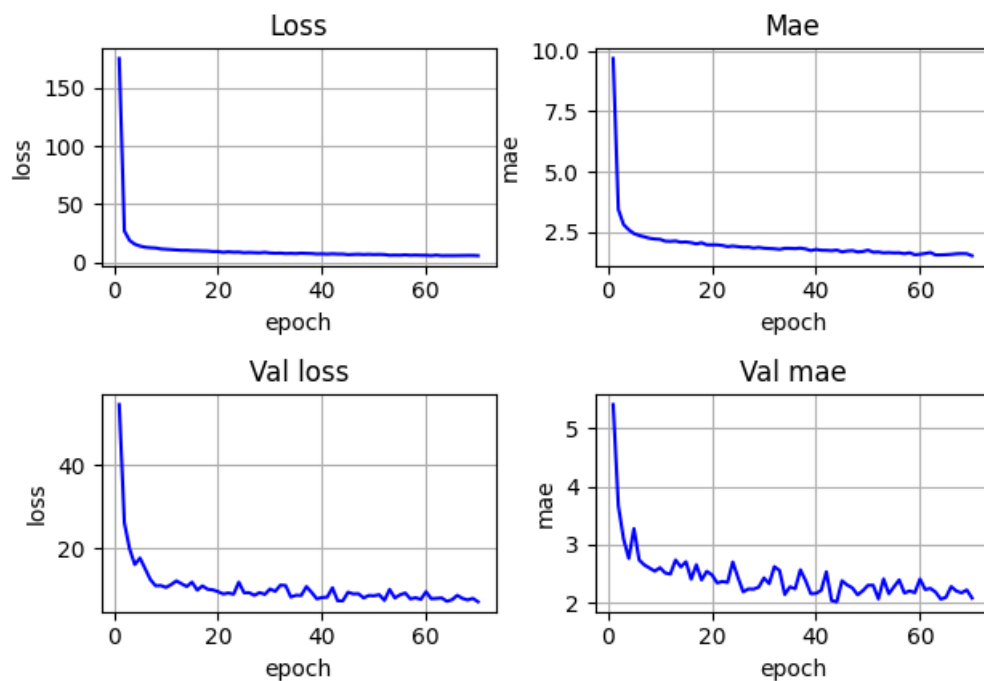
Model 4 of 6. Epochs: 70



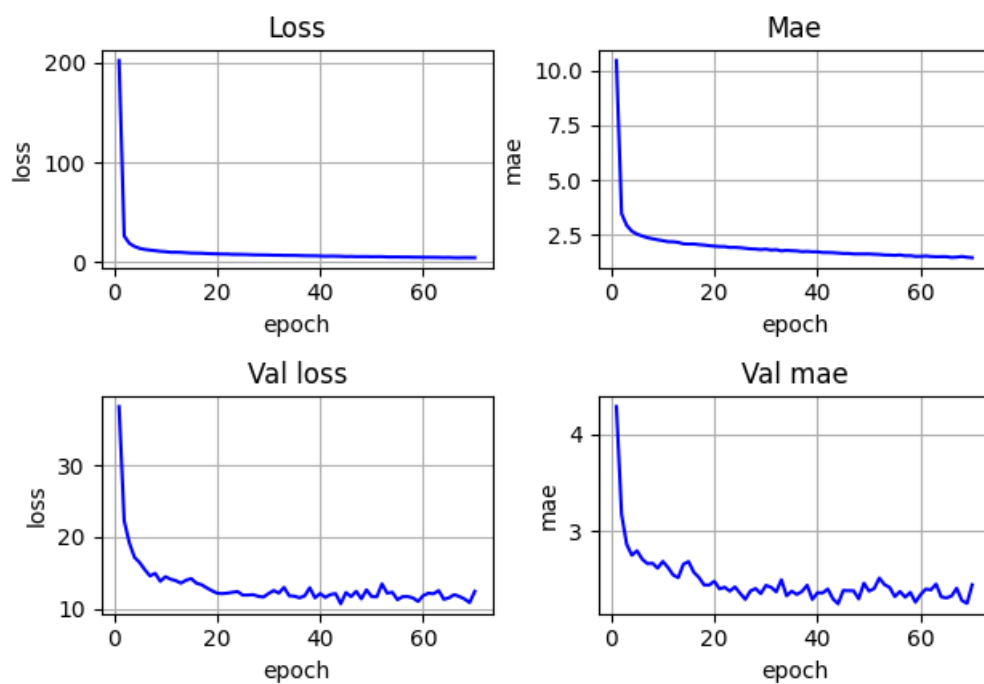
Model 5 of 6. Epochs: 70



Model 6 of 6. Epochs: 70



Mean results. k: 6, epochs: 70



Получили значение средней абсолютной ошибки по всем моделям 2.451, что хуже, чем при перекрестной проверке по 4 блокам.

## **ВЫВОДЫ**

В ходе выполнения работы была изучена задача регрессии и ее отличия от задачи классификации. Создана ИНС, предсказывающая цены на дома в пригороде Бостона. Изучено влияние количества эпох на результат обучения модели: при слишком большом количестве эпох модель переобучается, что ухудшает результат на новых данных. Определена примерная точка переобучения модели и проведена проверка качества модели при помощи перекрестной проверки по 4 и 6 блокам.