

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И
ОПТИКИ**

Факультет систем управления и робототехники

**Отчет по лабораторной работе №1 «Прогнозирование
цен на жилье с помощью нейросетевой
регрессионной модели»**

Выполнил: студент гр. **R3238**
Рамзай Александр-Лилливин Станиславович

Преподаватели: Михаил Каканов,
Олег Эстафьев

1. Цель работы

Необходимо по имеющимся данным о ценах на жильё предсказать окончательную цену каждого дома с учетом характеристик домов с использованием нейронной сети. Описание набора данных содержит 80 классов (набор переменных) классификации оценки типа жилья, и находится в файле data_description.txt.

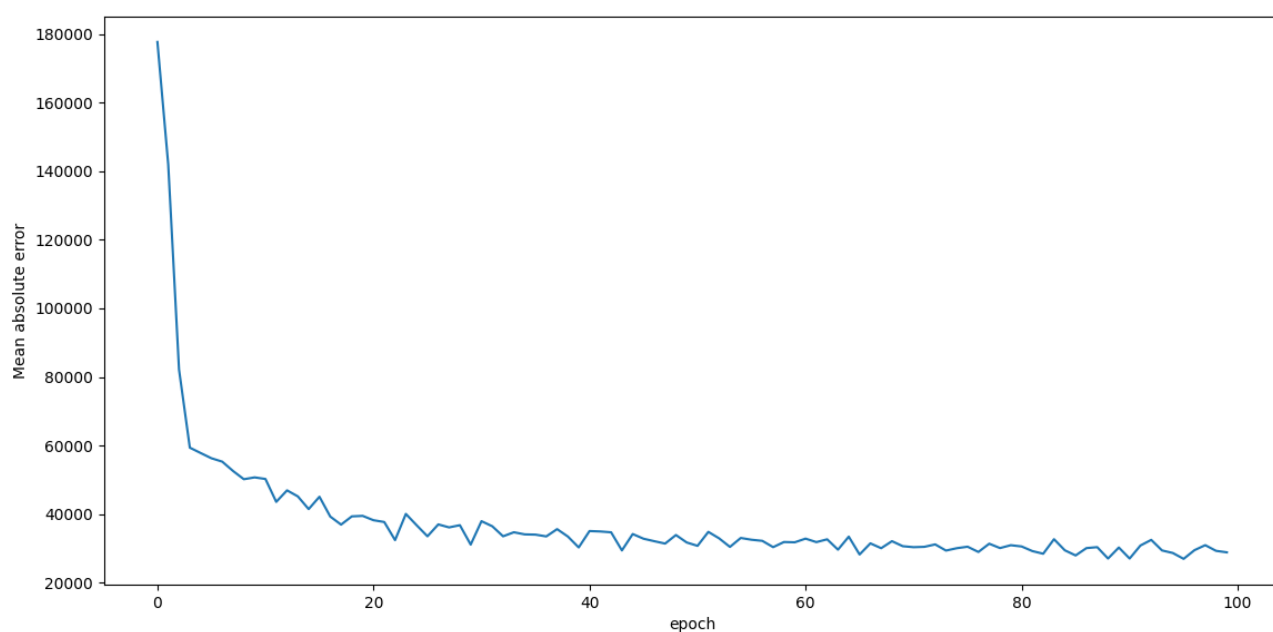
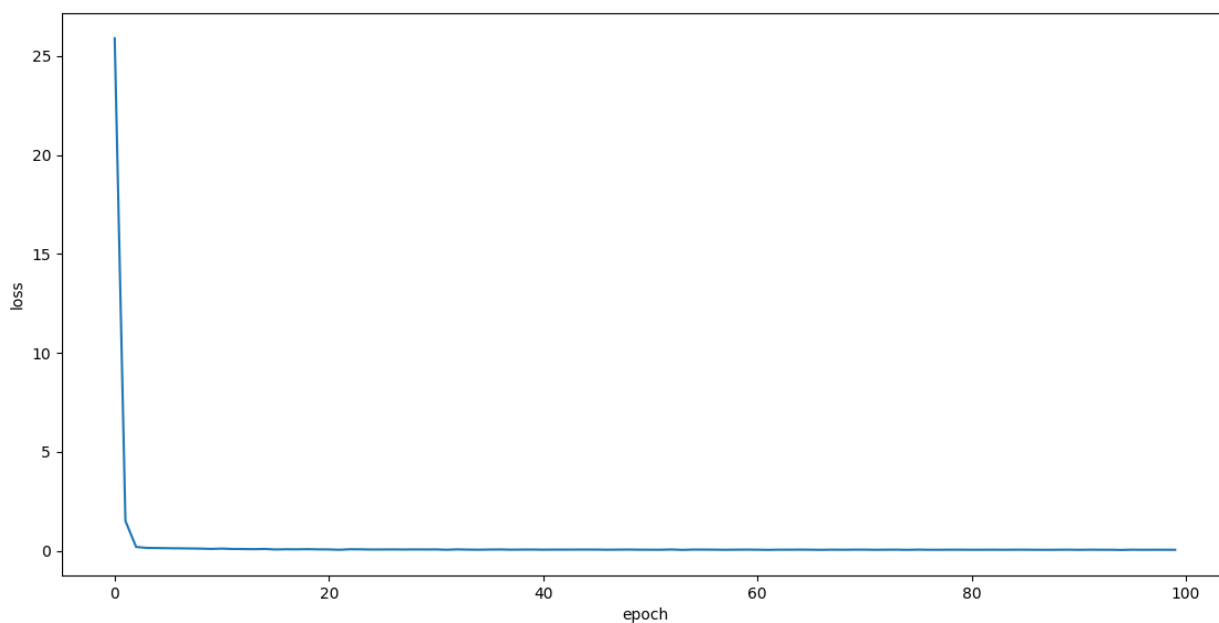
В работе требуется дополнить раздел «Моделирование» в подразделе «Построение и обучение модели» создать и инициализировать последовательную модель нейронной сети с помощью фреймворков тренировки нейронных сетей как: Torch или Tensorflow. Скомпилировать нейронную сеть выбрав функцию потерь и оптимизатор соответственно. Оценить точность полученных результатов. Вывести предсказанные данные о продаже.

2. Материалы работы

При следующих параметрах нейронной сети мне удалось добиться наилучшего результата:

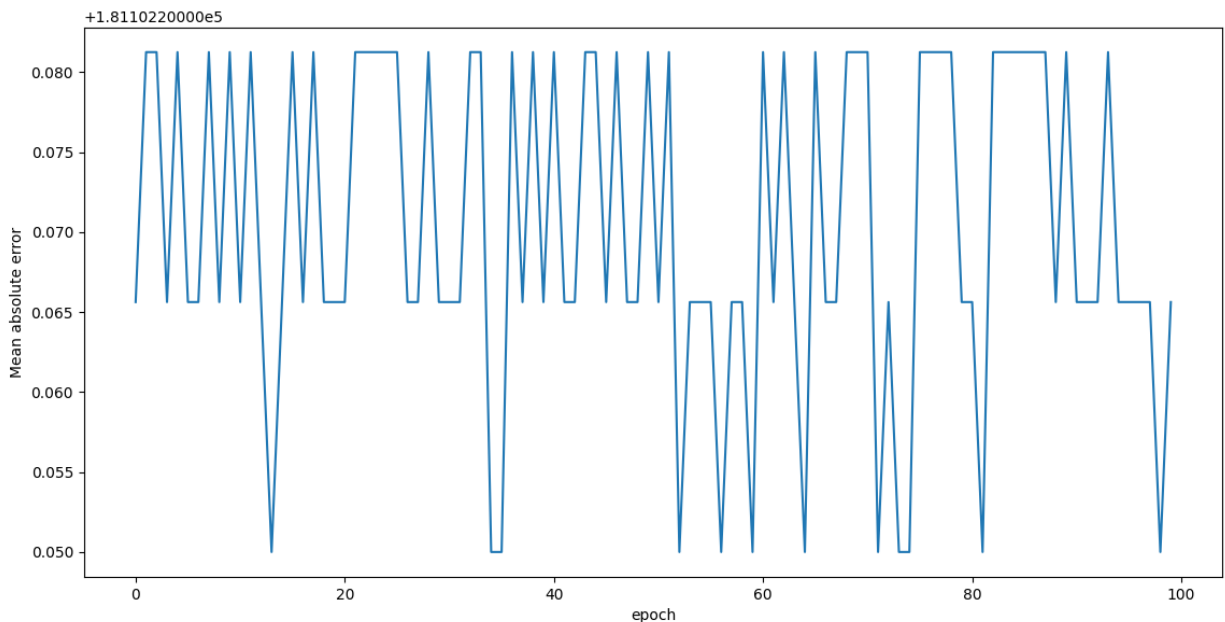
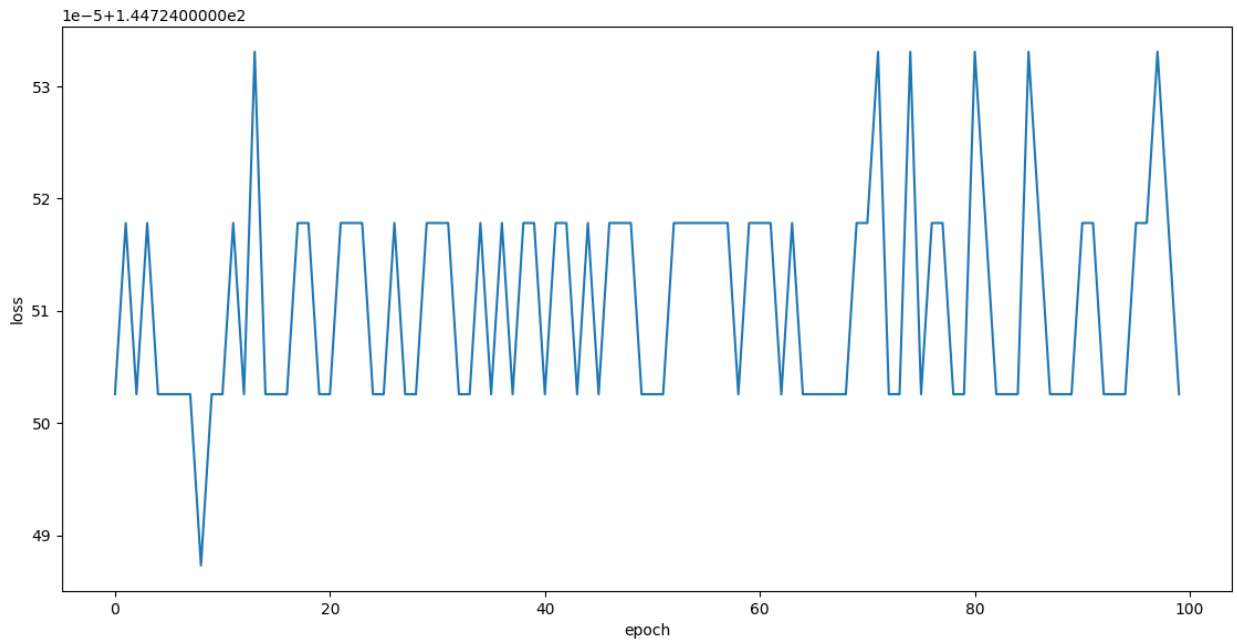
```
74 # =====NEURAL NETWORK=====
75
76 model = Sequential(Dense(100, input_dim=75))
77 model.add(layers.Dense(100, activation='relu'))
78 model.add(layers.Dense(100, activation='softplus'))
79 model.add(layers.Dense(100, activation='softplus'))
80 model.add(layers.Dense(100, activation='softplus'))
81 model.add(layers.Dense(100, activation='softplus'))
82 model.add(layers.Dense(100, activation='relu'))
83 model.add(layers.Dense(100, activation='relu'))
84 model.add(layers.Dense(1))
85
86 model.compile(optimizer="rmsprop", loss="MSLE", metrics=["mae"])
87 model.summary()
88
89 history = model.fit(X_train, y_train, epochs=100, batch_size=100)
```

Результаты:



```
Predictions
[[154311.72 ]
 [146486.84 ]
 [200888.94 ]
 ...
 [202981.39 ]
 [111097.586]
 [233636.7  ]]
```

Однако с такими параметрами на выходе иногда получаются совершенно странные и нереалистичные результаты, как, например, вот этот:



```
Predictions
[[-146.3194 ]
 [-114.26275]
 [-171.59367]
 ...
 [-215.5803 ]
 [-117.763504]
 [ -97.9299  ]]
```

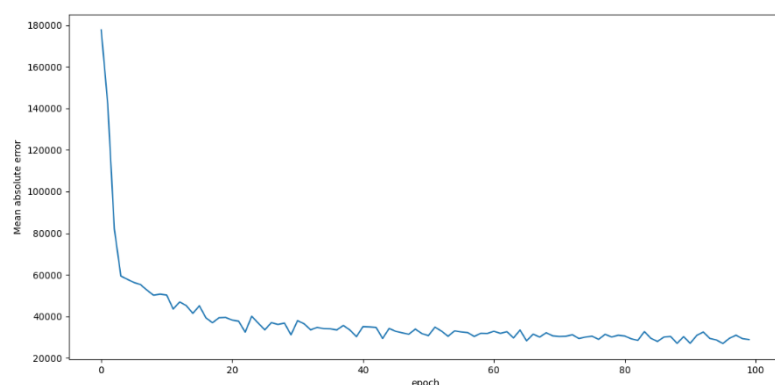
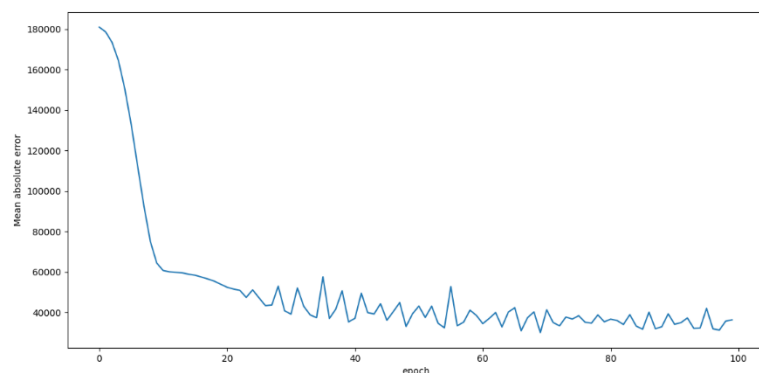
Если использовать такую нейронную сеть без участия человека – то есть ручной проверки результатов на адекватность – или без нужных условий – как минимум, чтобы цены на жильё не были отрицательными – то можно получить не самый приятный результат. По ощущениям, она выдаёт бредовый результат с вероятностью $\frac{1}{4}$.

Вот несколько адекватных результатов работы сети:

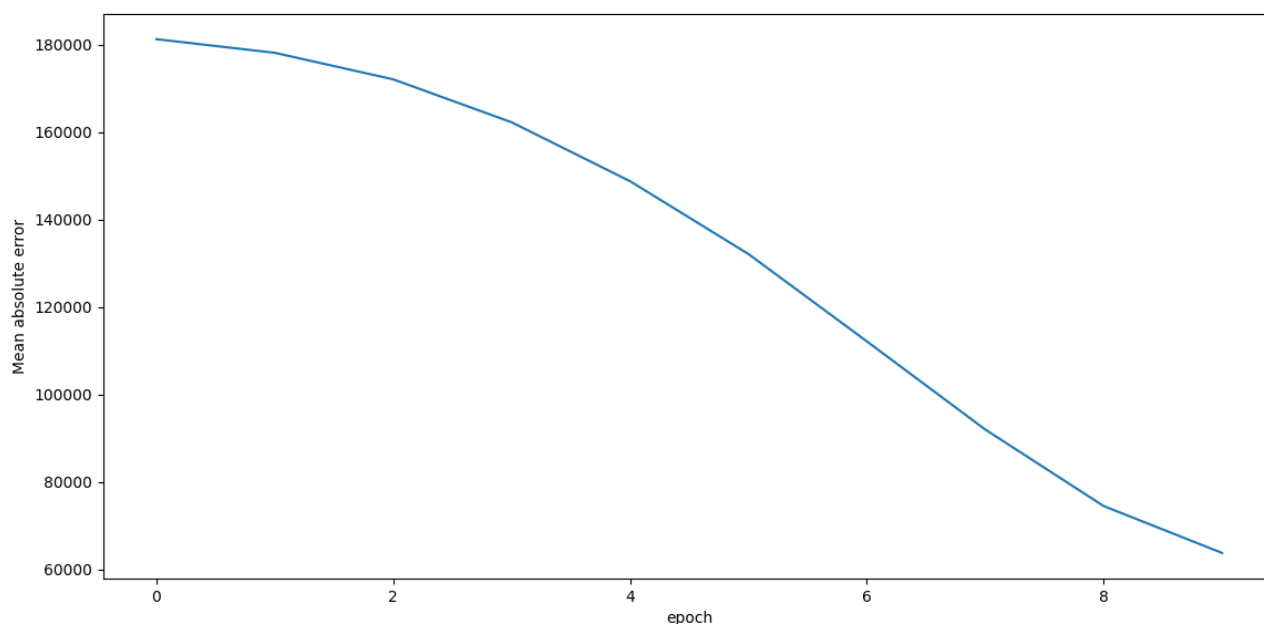
Predictions	Predictions	Predictions
[142097.89]	[167073.39]	[128974.73]
[165391.61]	[167776.19]	[143646.23]
[184511.45]	[217195.42]	[170253.64]
...
[183390.5]	[216301.73]	[158138.95]
[103461.836]	[120445.31]	[95077.836]
[218344.7]]	[258703.2]]	[194421.23]]

3. Вывод

1. Batch size влияет на разброс MAE. Например, вот график MAE при batch_size = 300 (слева) и batch_size = 100 (справа)



2. Количество эпох напрямую влияет на размер MAE, а именно: чем больше эпох — тем установившаяся ошибка меньше. Ниже приведён график при тех же параметрах, что и раньше с batch_size = 300 и 10 эпохами.



Глобальный вывод: чтобы обучить нейронную сеть нужно сначала обучить другую нейронную сеть, которая написала этот текст.