

Национальный исследовательский университет ИТМО
Факультет систем управления и робототехники

Прикладной искусственный интеллект
Лабораторная работа №2

Выполнил:

Никандров Сергей Андреевич

R32352

Преподаватель:

Евстафьев Олег Александрович

Санкт-Петербург

2022 год

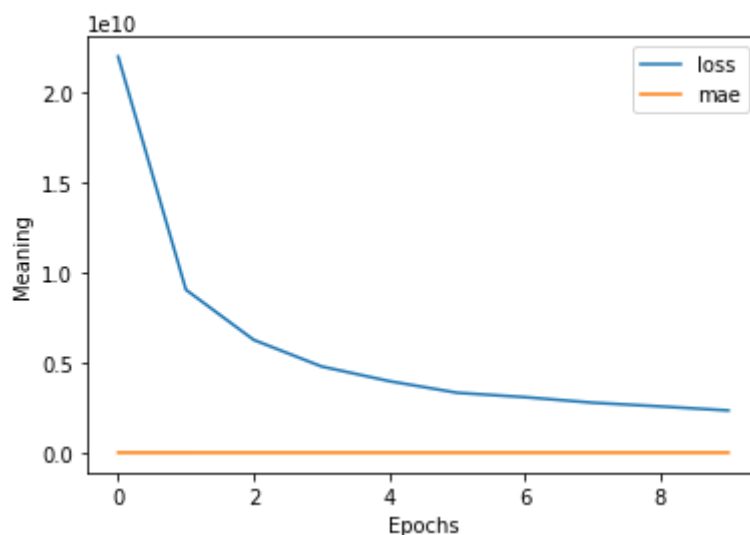
Задание

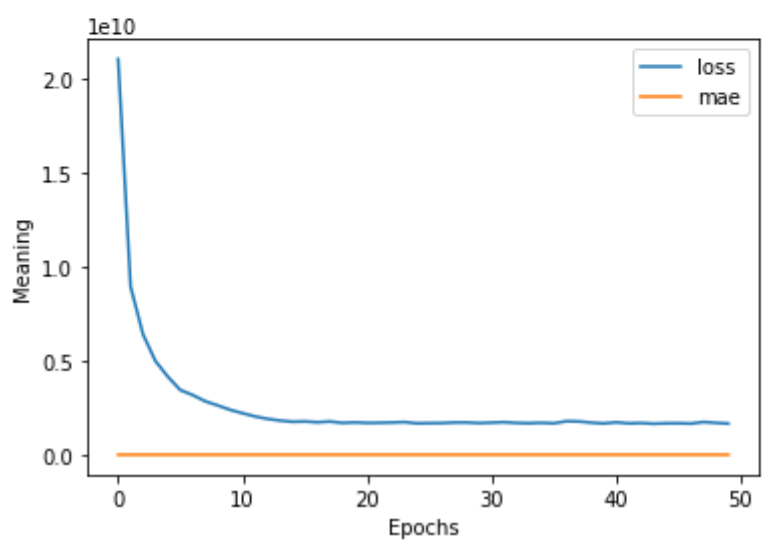
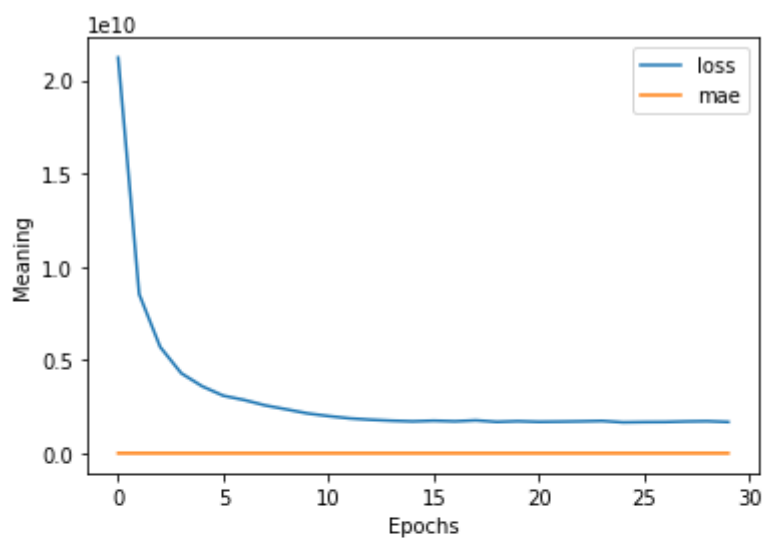
Необходимо по имеющимся данным о ценах на жильё предсказать окончательную цену каждого дома с учётом характеристик домов с использованием нейронной сети. Описание набора данных содержит 80 классов (набор переменных) классификации оценки типа жилья находится в файле data_description.txt

Решение

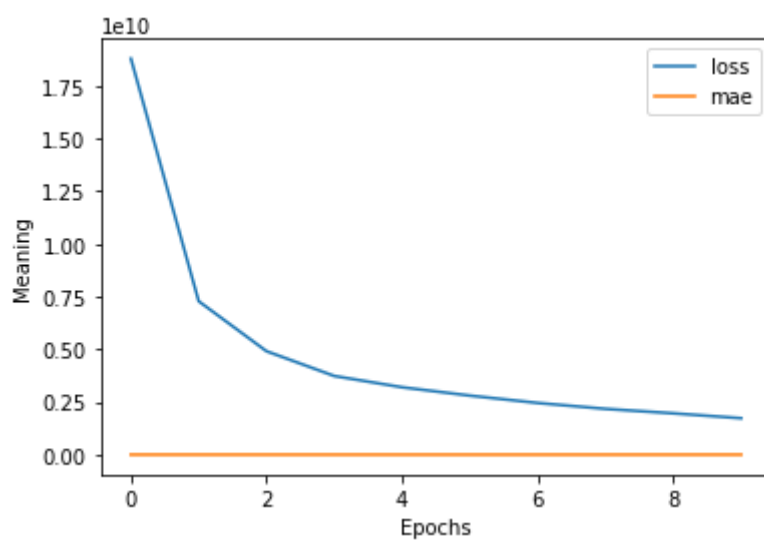
Я решил создать несколько моделей нейронной сети, с разными гиперпараметрами, чтобы выявить их зависимость на результат. Из данного опыта я выяснил, что количество нейронов влияет на нагрузку на компьютер. Количество эпох также влияет на нагрузку, но слишком маленькое количество эпох может привести к недообучению, а слишком больше к переобучению. Я решил перебрать количество нейронов и эпох в моих моделях, чтобы найти более подходящие гиперпараметры. Графики:

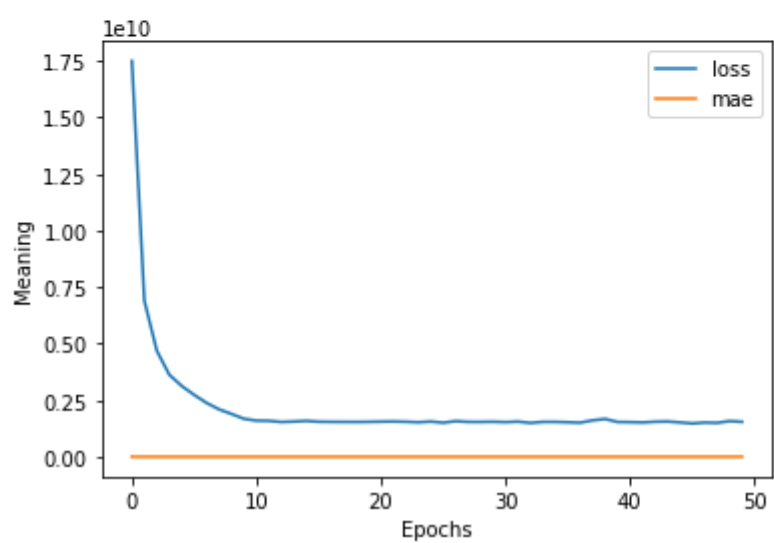
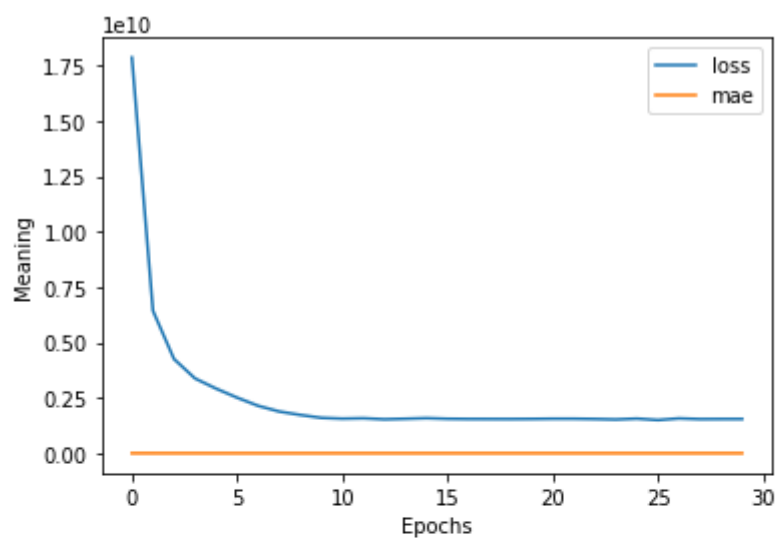
1) Количество нейронов равно 100:



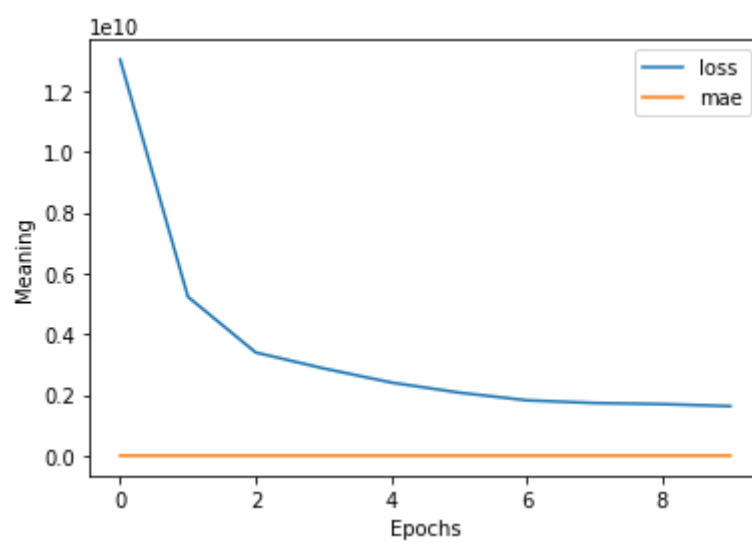


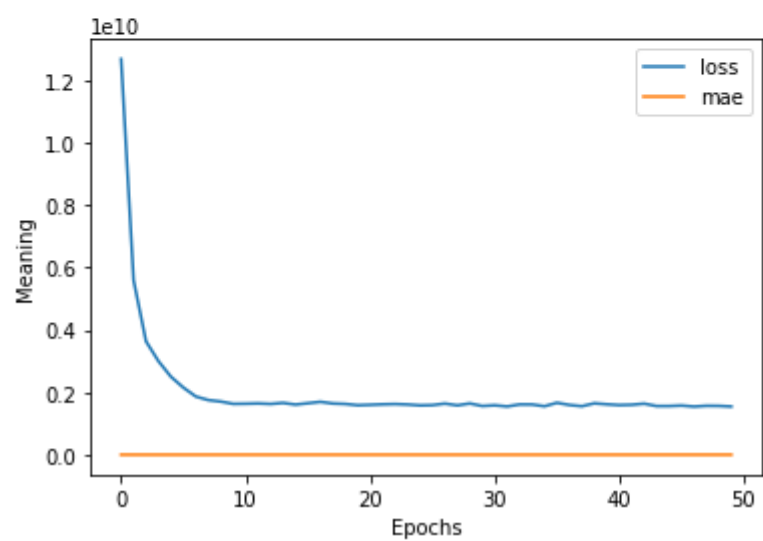
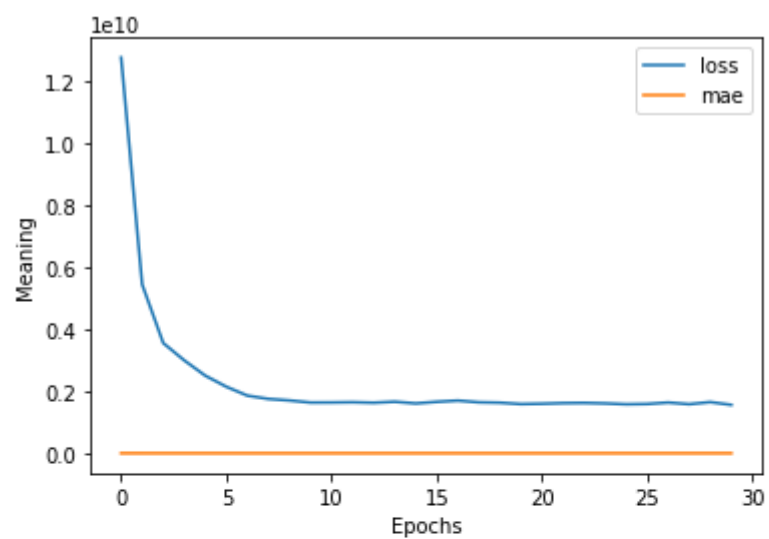
2) Количество нейронов равно 200:



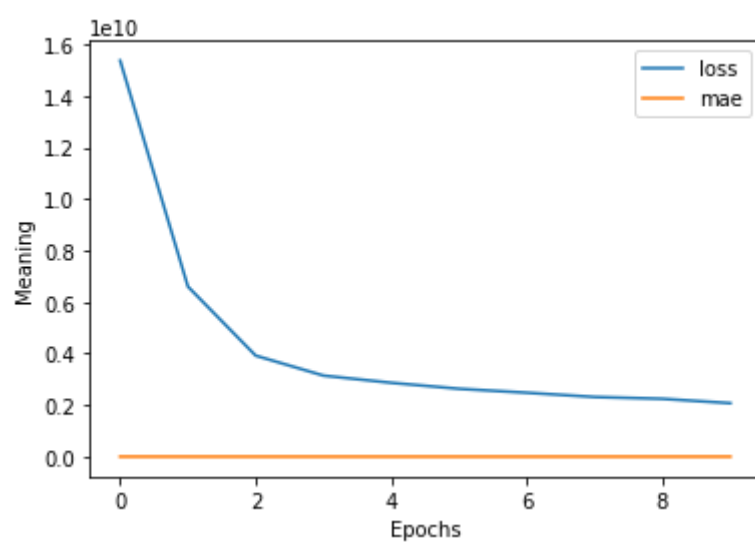


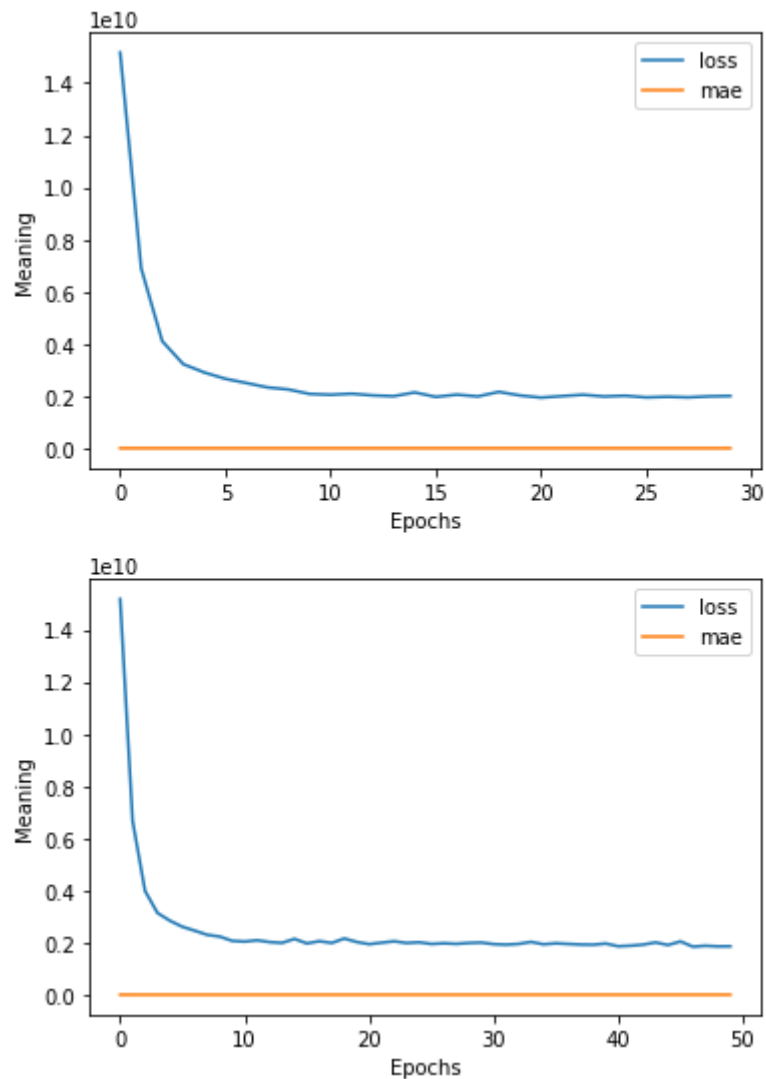
3) Количество нейронов равно 300:





4) Количество нейронов равно 150:





Опираясь на данные графики можно сделать вывод о том, что оптимальное количество нейронов равняется 100 при количестве эпох равном 30 для минимизации ошибки. Поэтому, чтобы оптимальнее всего решить задачу и при этом иметь более-менее быстрое выполнение я решил остановиться на 100 нейронах и 30 эпохах.

После этого я решил выбрать более подходящую функцию потерь. Я рассмотрел `mse` и `categorical_crossentropy`. Если `mse` показал хорошие результаты, то `categorical_crossentropy` показывал, что цены будут отрицательными.

Вывод с mse, вывод с categorical_crossentropy соответственно:

	Id	SalePrice		Id	SalePrice
0	1461	141455.390625	0	1461	2213.813477
1	1462	97702.710938	1	1462	-120.856125
2	1463	190317.765625	2	1463	2711.356201
3	1464	189526.156250	3	1464	1893.675659
4	1465	163050.203125	4	1465	712.352417
...
1454	2915	71625.265625	1454	2915	96.096069
1455	2916	94542.585938	1455	2916	192.715729
1456	2917	199825.906250	1456	2917	3988.361328
1457	2918	89276.531250	1457	2918	1636.463989
1458	2919	231536.359375	1458	2919	1897.434937

[1459 rows x 2 columns] [1459 rows x 2 columns]

В итоге лучше всего подошла mse, а categorical_crossentropy плохо подошла под задачу предсказания. Остальные гиперпараметры не сильно влияют на результаты модели.

Код на datalore

<https://datalore.jetbrains.com/notebook/kJEwvIMafpjQ3KeI9h5Pig/eHboJjgWYy7UDMcs02vCOs/>

Ответы на вопросы

- 1) Optimizers - один из двух аргументов, необходимых для компиляции модели. важный компонент архитектуры нейронных сетей. Они играют важную роль в процессе тренировки нейронных сетей, помогая им делать всё более точные прогнозы. Оптимизаторы определяют оптимальный набор параметров модели, таких как вес и смещение, чтобы при решении конкретной задачи модель выдавала наилучшие результаты.
- 2) Loss - Loss / функция ошибки / функция потерь нейронной сети – математическая дифференцируемая функция, характеризующая разницу между «истинным» значением

целевой переменной и предсказанным нейронной сетью значением.

- 3) Что такое эпоха (Epoch)? В чём отличие от итерации (Iteration)? Произошла одна эпоха (epoch) — весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз. Так как одна epoch слишком велика для компьютера, датасет делят на маленькие партии (batches).

Итерации — число батчей, необходимых для завершения одной эпохи. Отметим: Число батчей равно числу итераций для одной эпохи. Например, собираемся использовать 2000 тренировочных объектов. Можно разделить полный датасет из 2000 объектов на батчи размером 500 объектов. Таким образом, для завершения одной эпохи потребуется 4 итерации.

- 4) Что такое функция активации? Какие вам известны? Как и зачем используются в нейронной сети? Функция активации решает, следует ли активировать нейрон или нет, путем вычисления взвешенной суммы и дальнейшего добавления к ней смещения. Цель функции активации - внести нелинейность в выходные данные нейрона. Объяснение: Мы знаем, что в нейронной сети есть нейроны, которые работают в соответствии с весом, смещением и их соответствующей функцией активации. В нейронной сети мы бы обновляли веса и смещения нейронов на основе ошибки на выходе. Этот процесс известен как обратное распространение. Функции активации делают возможным обратное распространение, поскольку градиенты предоставляются вместе с ошибкой для обновления весов и смещений. Примеры:

РЕЛУ

ReLU означает «Выпрямленный линейный блок».

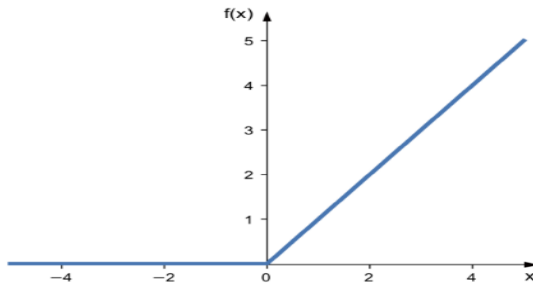
Из всех функций активации это тот, который **наиболее похож на линейный**:

- Для неотрицательных значений он просто применяет тождество.
- Для отрицательных значений возвращает 0.

В математических словах

$$f(x) = \max(x, 0)$$

Это означает, что все отрицательные значения станут 0, в то время как остальные значения останутся без изменений.



Это биологически вдохновленная функция, так как нейроны в мозге будут либо «запускать» (возвращать положительное значение), либо нет (возвращать 0).

СИГМОИД

Сигмовидная функция принимает любое действительное число в качестве входных данных, и **возвращает значение от 0 до 1**. Поскольку оно непрерывно, оно эффективно **smushes** значения:

Если вы примените сигмоид к 3, вы получите 0,95. Примените его к 10, вы получите 0,999 ... И он будет продолжать приближаться к 1, даже не достигнув его

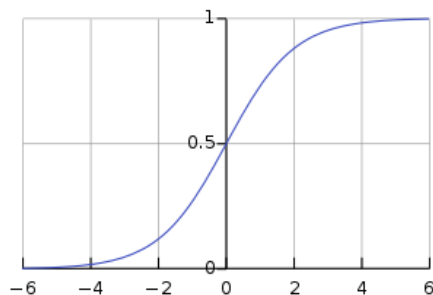
То же самое происходит в отрицательном направлении, за исключением того, что оно сходится к 0.

Вот математическая формула для сигмовидной функции.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Как видите, он приближается к 1, когда x приближается к бесконечности, и приближается к 0, если x приближается к минус бесконечность.

Он также симметричен и имеет значение 1/2, когда его вход равен 0.



Поскольку она принимает значения от 0 до 1, эта функция чрезвычайно полезна в качестве выходных данных, если вы хотите смоделировать вероятность.

Смысл: Нейронная сеть без функции активации - это, по сути, просто модель линейной регрессии. Функция активации выполняет нелинейное преобразование входных данных, что

делает его способным обучаться и выполнять более сложные задачи.

- 5) Что такое MSE(Mean Squared Error) - Средняя квадратичная ошибка? Что такое MAE(Mean Absolute Error)? Для чего используются.

MAE (Mean Absolute Error) - Метрика измеряет среднюю сумму абсолютной разницы между фактическим значением и прогнозируемым значением.

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t|, \text{ where } e_t = original_t - predict_t$$

Средняя квадратичная ошибка (MSE): средняя квадратичная ошибка является наиболее распространенной функцией потерь. Функция потерь MSE широко используется в линейной регрессии в качестве показателя эффективности. Чтобы рассчитать MSE, надо взять разницу между предсказанными значениями и истинными, возвести ее в квадрат и усреднить по всему набору данных.

$$MSE = \frac{1}{m} \sum_{i=1}^m ||\hat{y}^{(i)} - y^{(i)}||^2$$

где $y^{(i)}$ – фактический ожидаемый результат, а $\hat{y}^{(i)}$ – прогноз модели.