МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»

ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра Вычислительной техники

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Нейронные сети»

Тема: «Рекуррентные нейронные сети»

Студенты гр.0305 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Петракова М. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Швец С. А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Петруша П. Г.

Санкт – Петербург

2024

**Цель работы**

Целью данной лабораторной работы является изучение рекуррентных

нейронных сетей и их применение. В процессе работы будет рассмотрено

несколько различных архитектур рекуррентных нейронных сетей, включая

RNN, LSTM и GRU.

**Задание**

1. Подготовка данных
2. Архитектура RNN
3. Обучение RNN
4. Настройка гиперпараметров и выбор модели
5. Сравнение с существующими решениями

**Выполнение работы**

В качестве исходного набора данных выбран датасет, отражающий временную последовательность значений величины электроэнергии, вырабатываемой ветряными турбинами в Германии.

Первые 5 строк набора данных приведены на рис.1. dt – дата и время замера (через каждые 15 минут), MW – количество энергии в МВ. Размер выборки – 50000 строк.

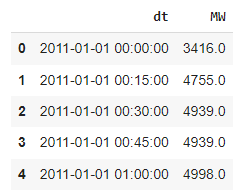


Рисунок 1

В качестве индекса назначена дата, по ней выполнена сортировка данных (рис.2).

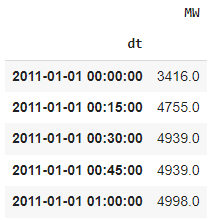


Рисунок 2

На рис.3 приведен график значений параметра.

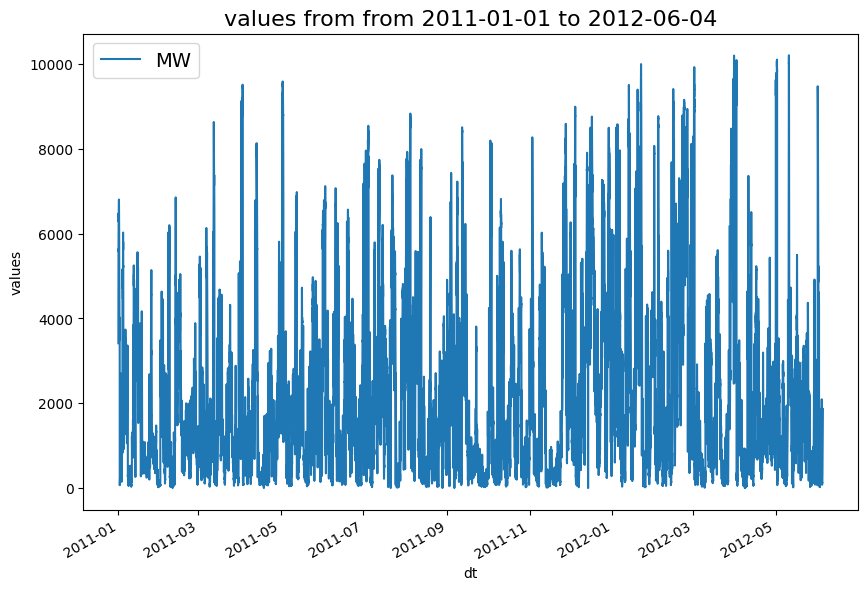


Рисунок 3

Данные разделены на обучающую и валидационную выборку (рис.4). График приведен на рис.5.

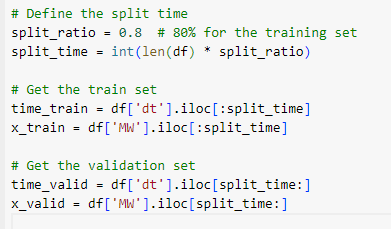


Рисунок 4

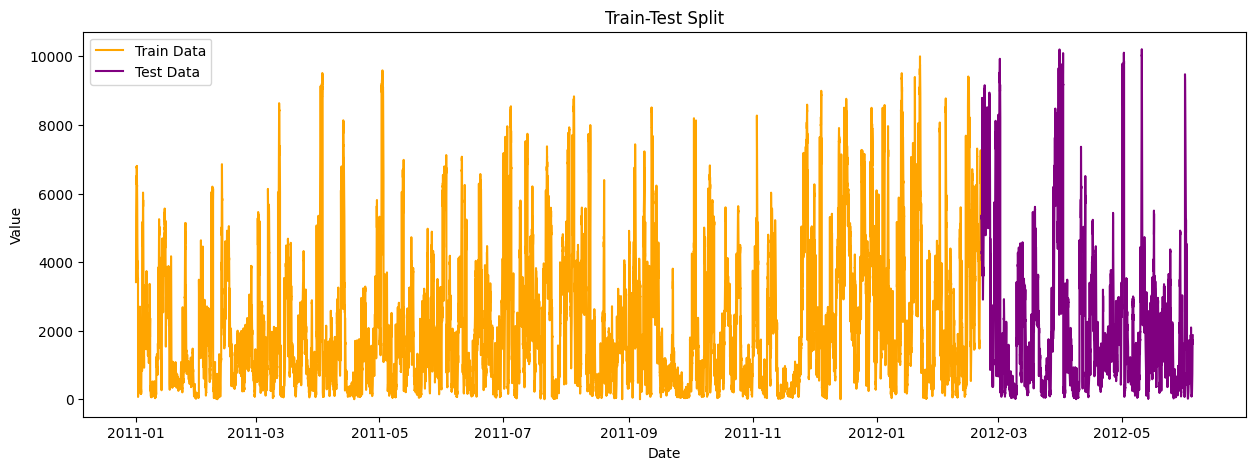


Рисунок 5

**Архитектуры RNN**

1. SimpleRNN

Рекуррентные нейронные сети (Recurrent Neural Networks,RNN) – сети с обратными или перекрестными связями между различными слоями нейронов.

Описание архитектуры модели приведена на рис.6.

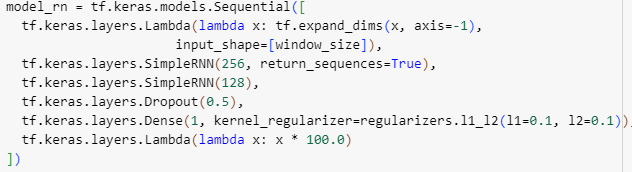


Рисунок 6

Модель обучалась в течение 10 эпох.

Результат запуска с оптимизатором Adam приведен на рис.7. Видно, что модель не переобучается, и достигает лучшего значения функции потерь. На рисунке 8 приведен график функции потерь.

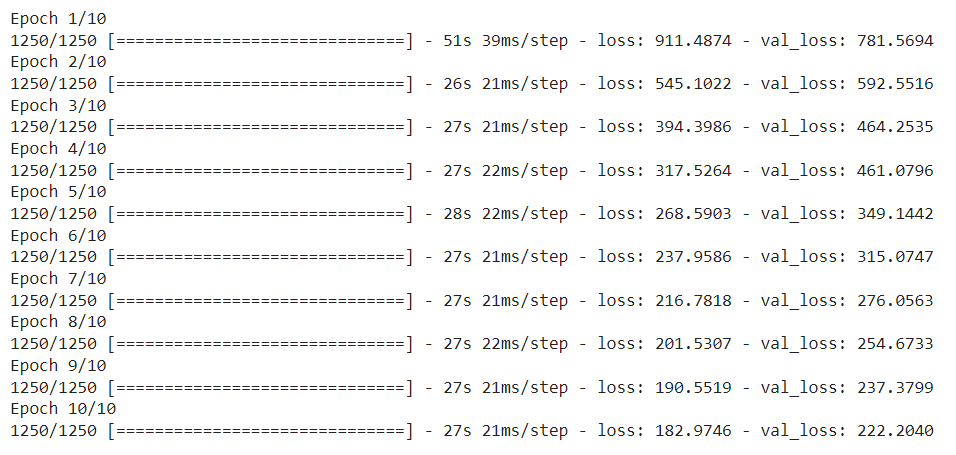


Рисунок 7

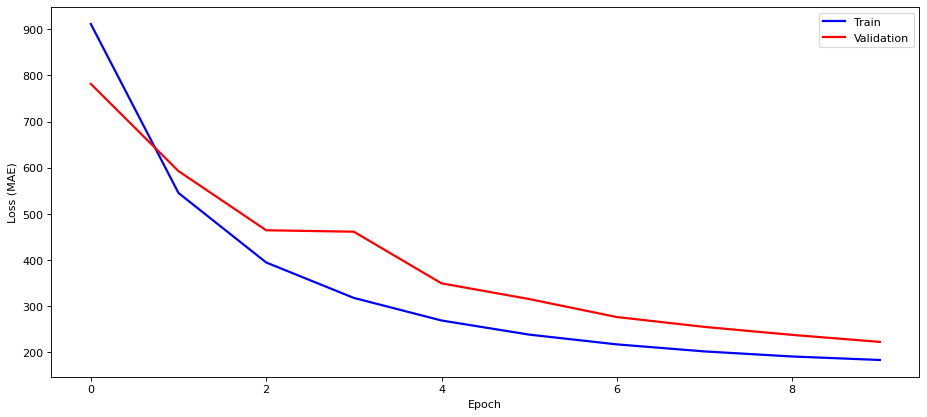


Рисунок 8

Результат запуска с оптимизатором SGD приведен на рис.9. График функции потерь приведен на рис.10.

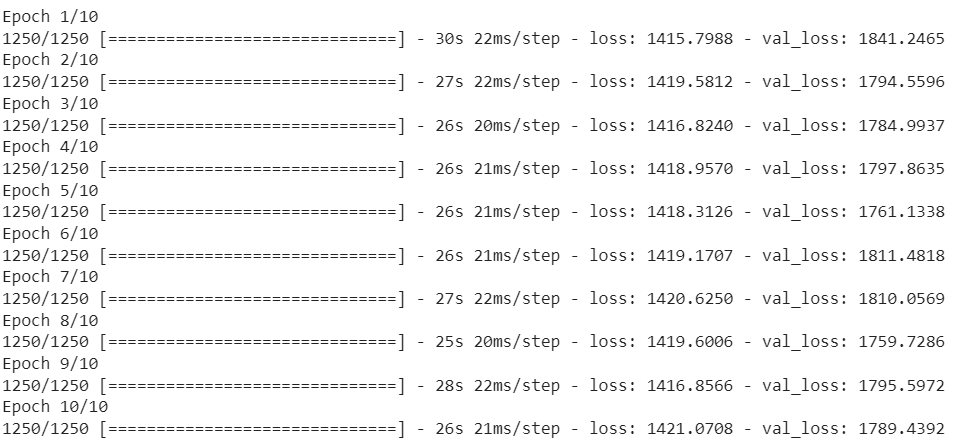


Рисунок 9

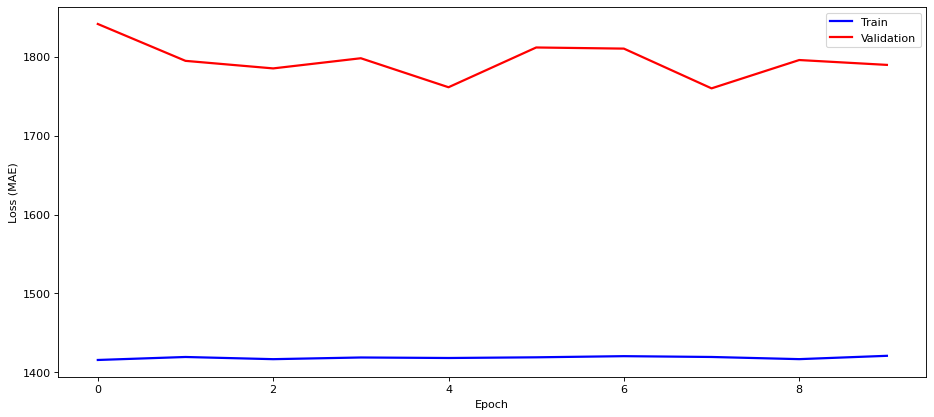


Рисунок 10

Результат запуска с оптимизатором Momentum приведен на рис.11. График функции потерь приведен на рис.12.

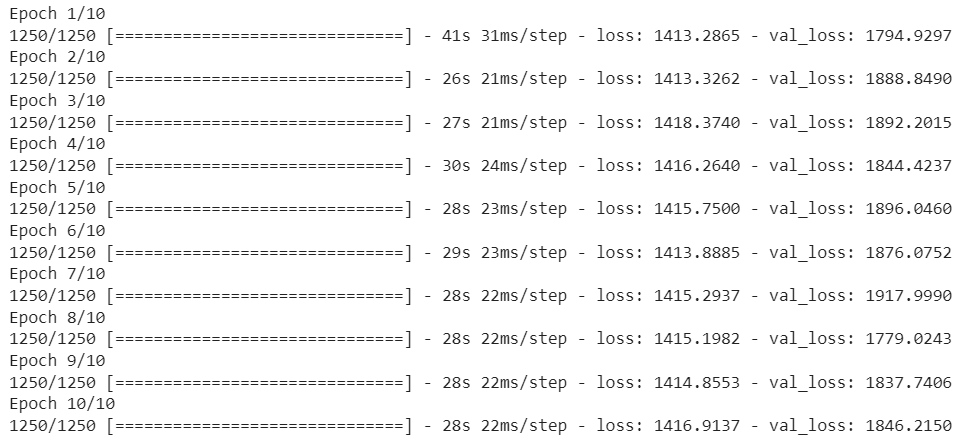


Рисунок 11

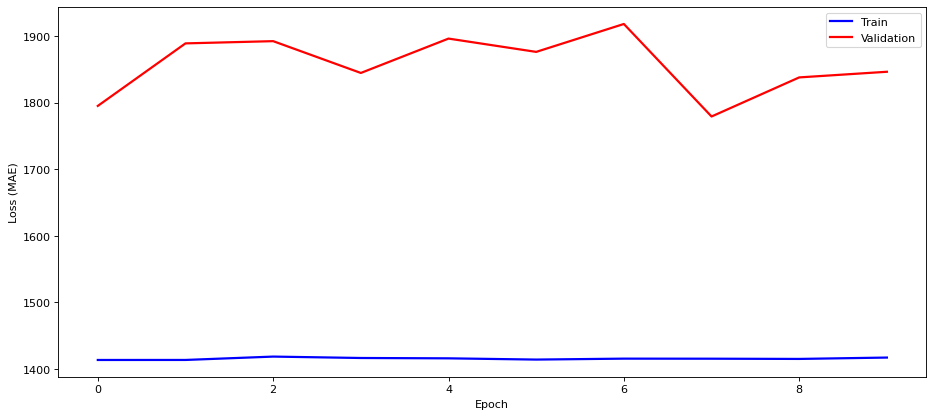


Рисунок 12

Результат запуска с оптимизатором RMSprop приведен на рис.13. График функции потерь приведен на рис.14.

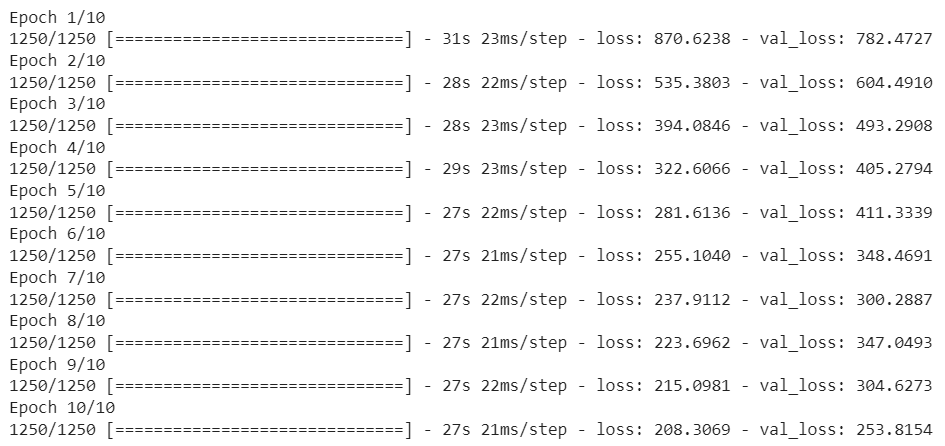


Рисунок 13

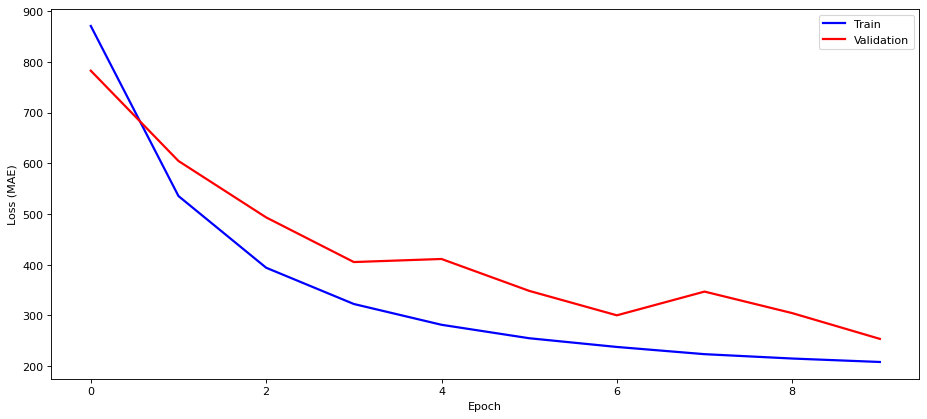


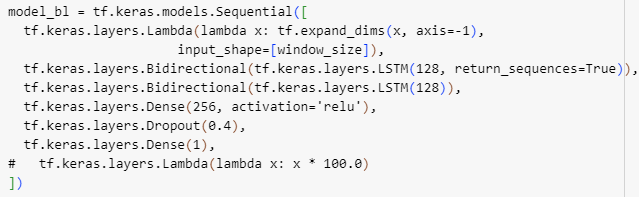
Рисунок 14

Делаем вывод о том, что наилучший оптимизатор для модели SimpleRNN применимо к нашей задаче – Adam.

1. LSTM

Вариация RNN, в которой объем памяти для модели увеличивается для размещения данных за более длительный период.

Описание архитектуры модели приведено на рис.15.

  
Рисунок 15

Модель обучалась в течение 10 эпох.

Результат запуска с оптимизатором Adam приведен на рис.16. Видно, что модель достигает лучшего значения функции потерь. На рисунке 17 приведен график функции потерь.

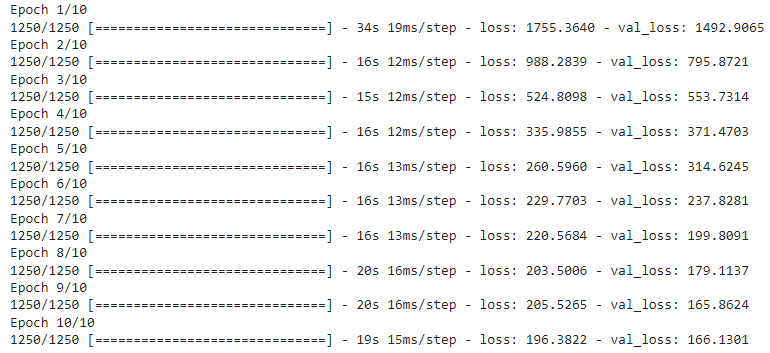


Рисунок 16

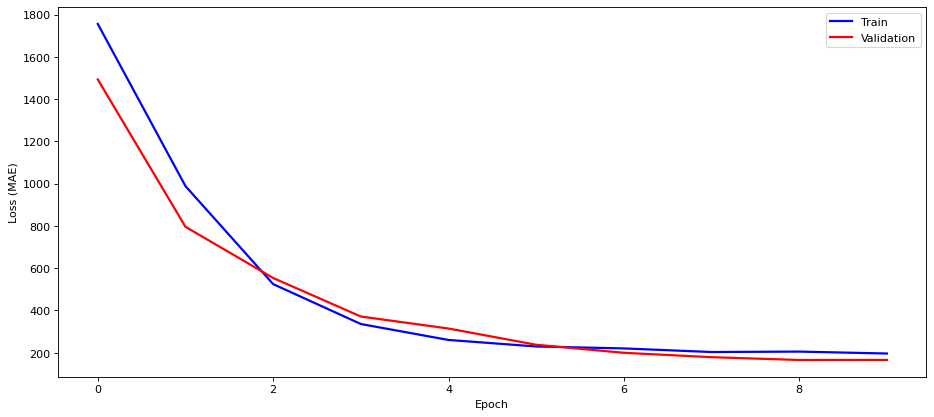


Рисунок 17

Результат запуска с оптимизатором SGD приведен на рис.18. Благодаря наличию слоя Dropout, модель не переобучается. На рисунке 19 приведен график функции потерь.

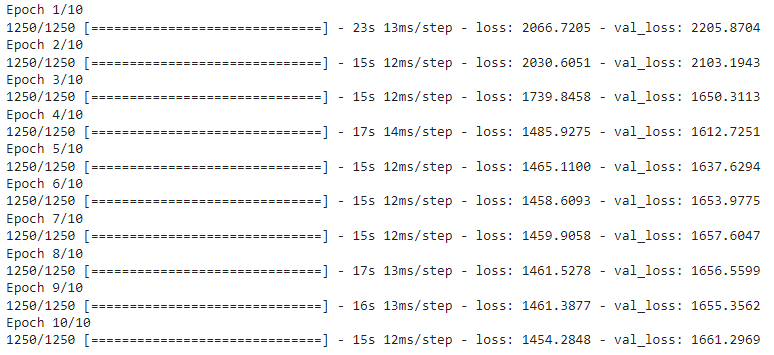


Рисунок 18

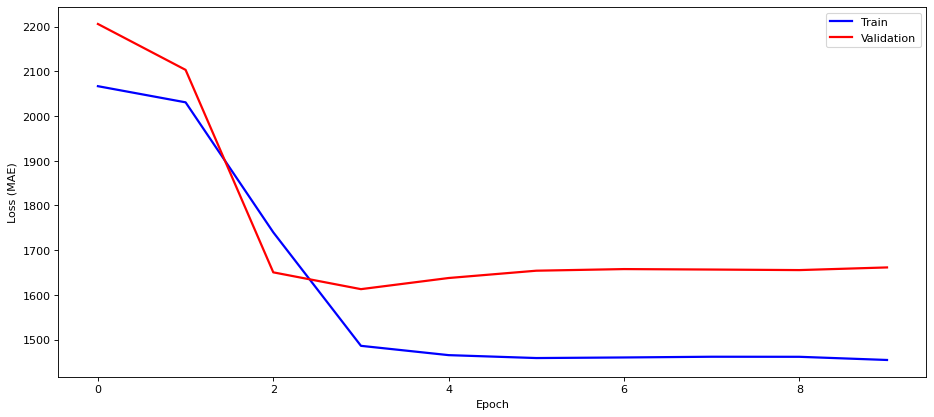


Рисунок 19

Результат запуска с оптимизатором Momentum приведен на рис.20. Потери практически не изменяются в процессе обучения. На рисунке 21 приведен график функции потерь.

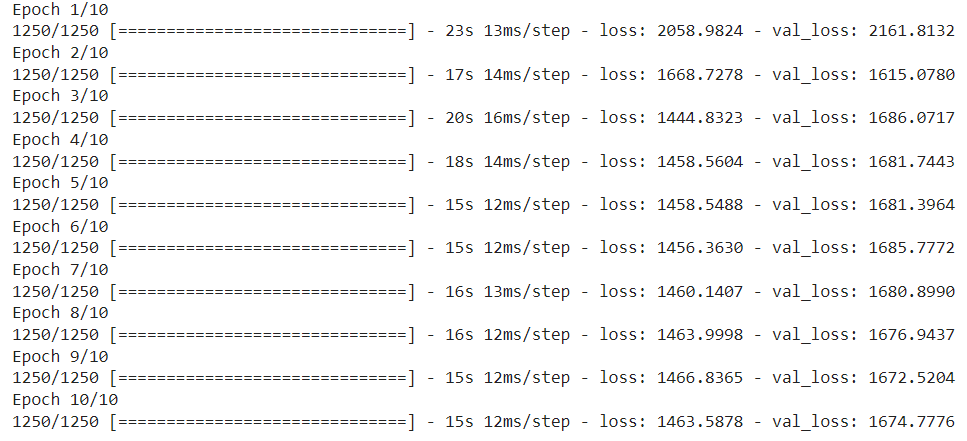


Рисунок 20

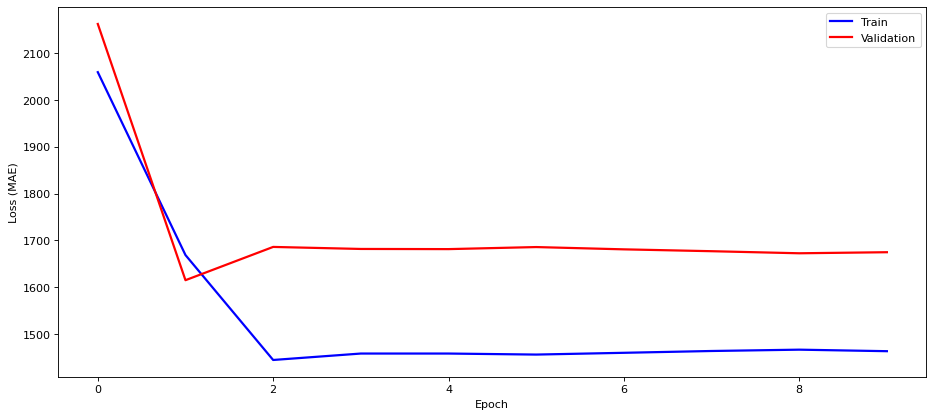


Рисунок 21

Результат запуска с оптимизатором RMSprop приведен на рис.22. Потери значительно снижаются в процессе обучения. На рисунке 23 приведен график функции потерь.

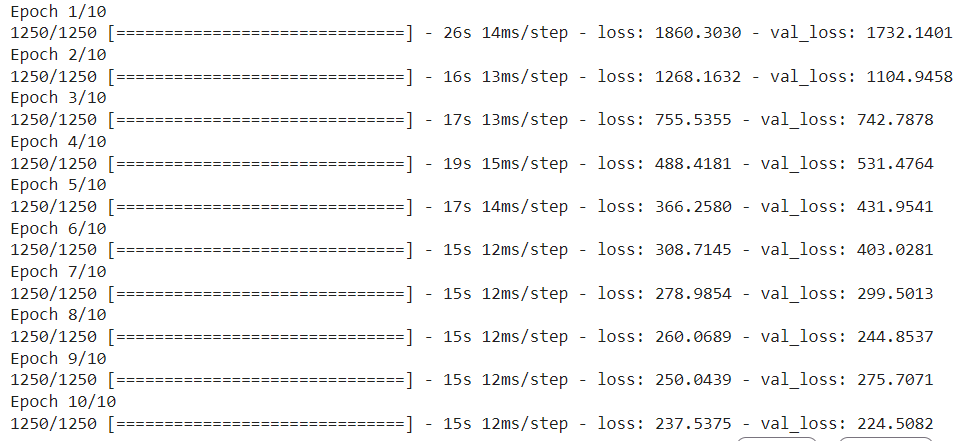


Рисунок 22

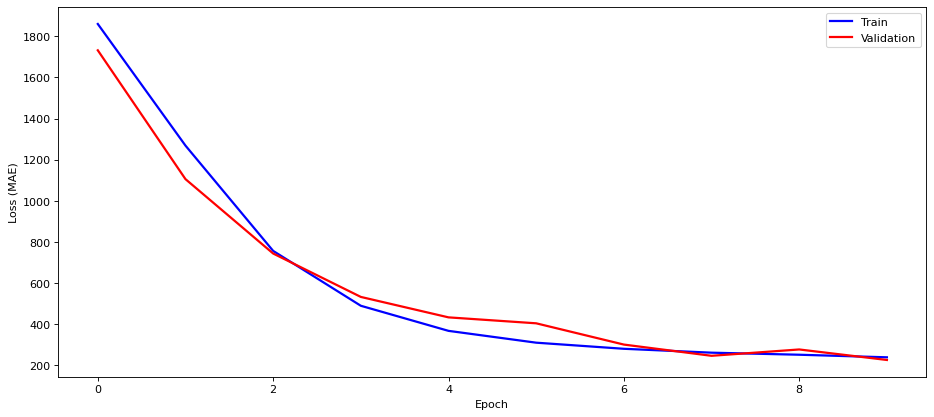


Рисунок 23

Наилучших результатов модель достигает с применением оптимизаторов Adam и RMSprop.

1. GRU

Вариация RNN, поддерживающая выборочное сохранение памяти. В эту модель добавлены шлюзы обновления и забывание в скрытом слое, что позволяет хранить информацию в памяти и удалять ее.

Описание архитектуры модели приведено на рис.24.

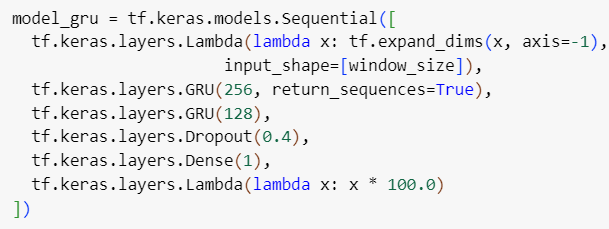


Рисунок 24

Результат запуска с оптимизатором Adam приведен на рис.25. Потери значительно снижаются в процессе обучения. На рисунке 26 приведен график функции потерь.

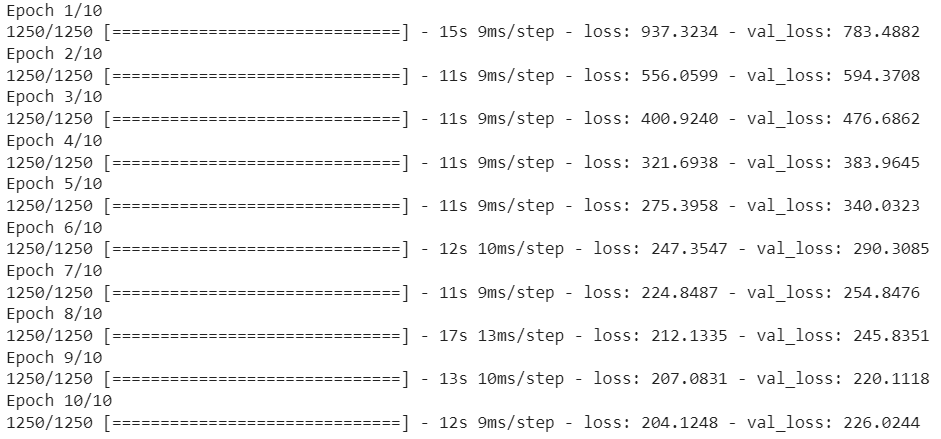


Рисунок 25

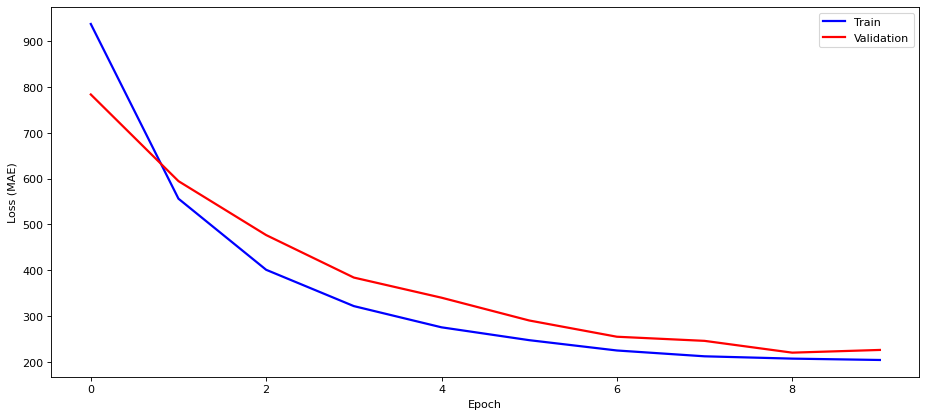


Рисунок 26

Результат запуска с оптимизатором SGD приведен на рис.27. На рисунке 28 приведен график функции потерь.

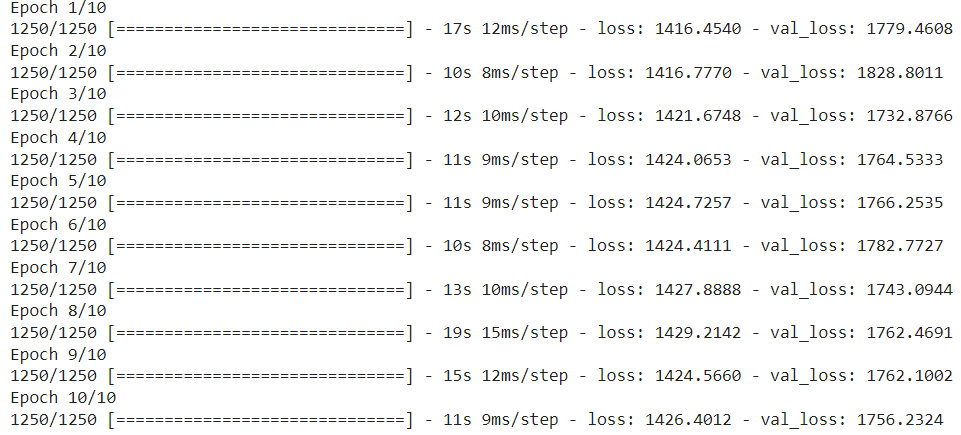


Рисунок 27

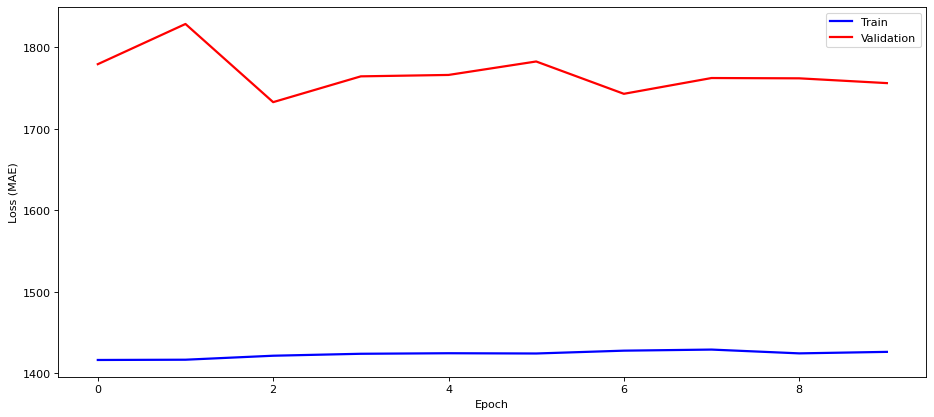
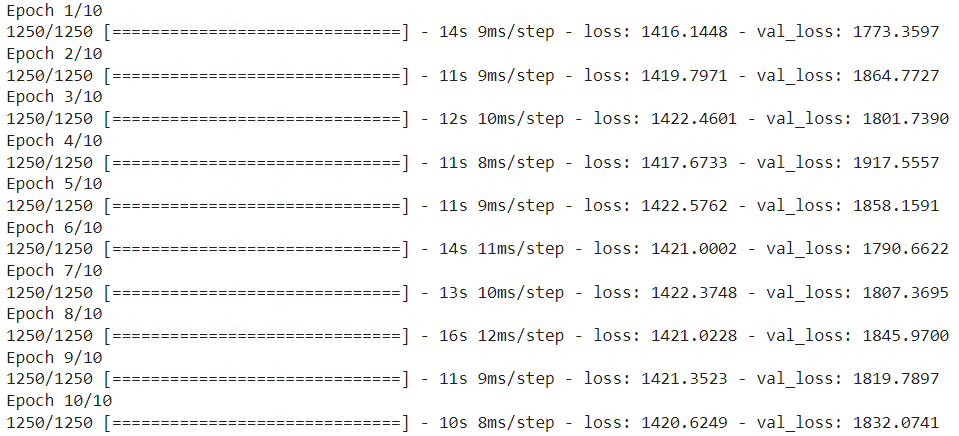


Рисунок 28

Результат запуска с оптимизатором Momentum приведен на рис.29. На рисунке 30 приведен график функции потерь.

  
Рисунок 29

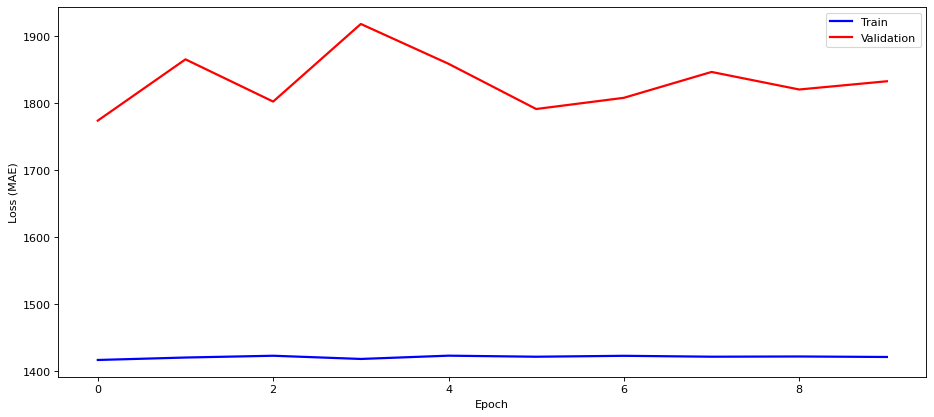


Рисунок 30

Результат запуска с оптимизатором RMSprop приведен на рис.31. На рисунке 32 приведен график функции потерь.

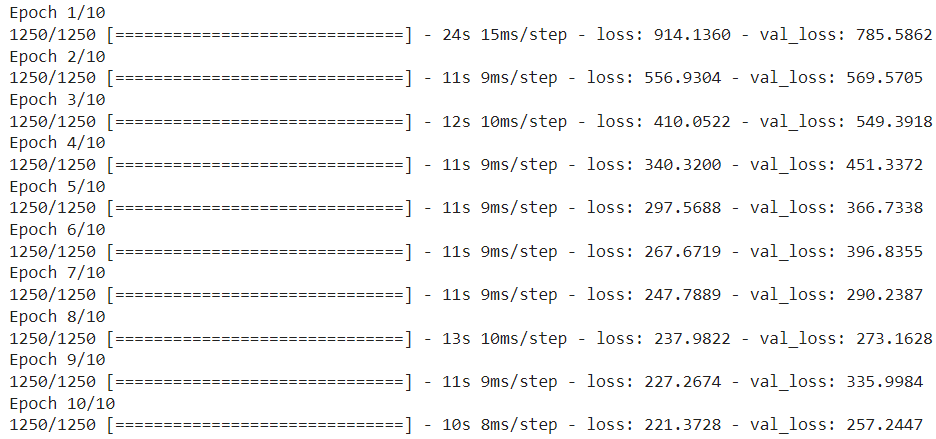


Рисунок 31

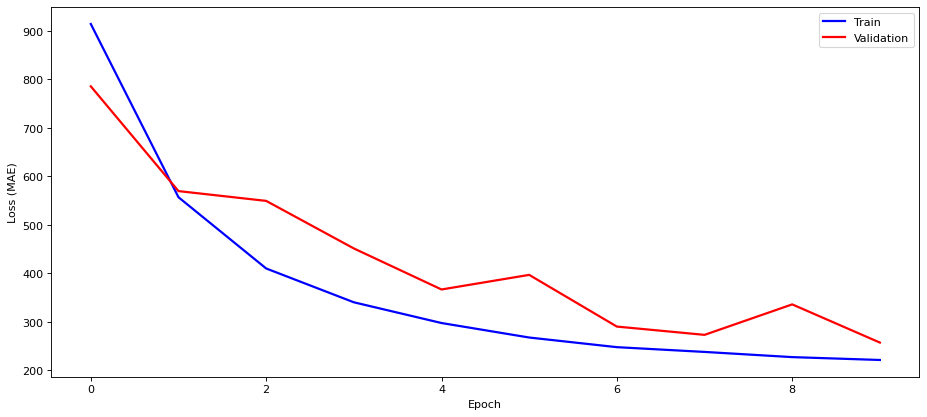


Рисунок 32

Наилучших результатов модель достигает с применением оптимизатора Adam.

Сравним результаты МАЕ (Mean Absolute Error) разных моделей:

simpleRNN + adam: 182.9746

LSTM + adam: 196.3822

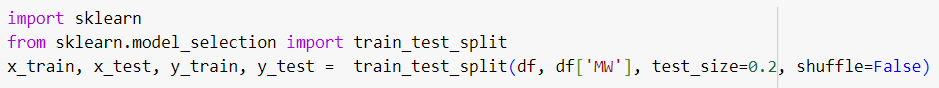
GRU + adam: 204.248

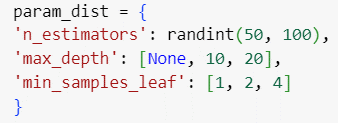
Adam - эффективный метод оптимизации, который адаптирует скорость обучения для каждого параметра. Наилучший результат показала модель simpleRNN в силу особенностей исследуемой последовательности.

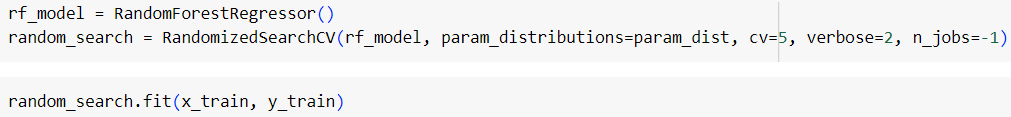
**Настройка гиперпараметров**

Random Search - метод оптимизации гиперпараметров, который заключается в случайном выборе значений гиперпараметров из заданного диапазона и оценке производительности модели с этими значениями. Это простой и эффективный способ настройки гиперпараметров модели машинного обучения.

Определяем параметры, создаем модель и запустим обучение (рис.33).







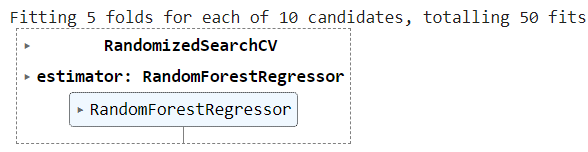
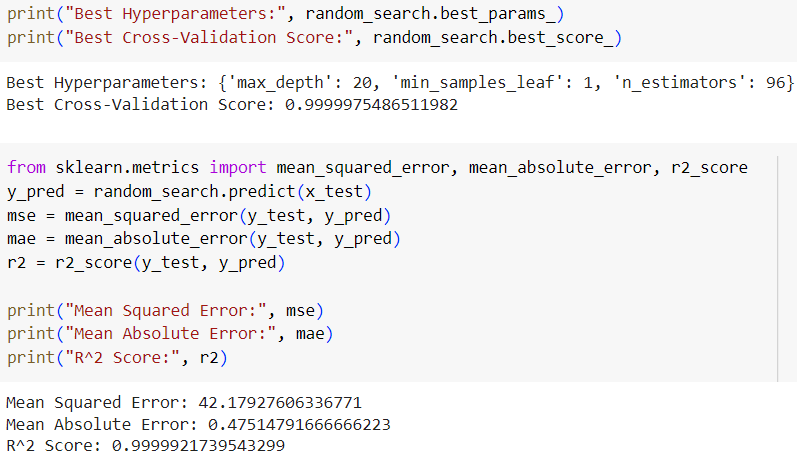


Рисунок 33

Результаты обучения приведены на рис.34.



MSE - метрика, которая сообщает среднеквадратичную разницу между прогнозируемыми значениями и фактическими значениями в наборе данных. R2 оценивает, насколько хорошо регрессионная модель соответствует фактическим данным.

Получены результаты, сигнализирующие об удачном подборе гиперпараметров.

**Ссылки на Google Colab:**

<https://colab.research.google.com/drive/18jK5_7Wj0PxNpRmDw5E9f1J3N7g5yyvs?usp=sharing>

<https://colab.research.google.com/drive/10sr9EDM2_p8rk0PHGVBibaaHmMCtMfgX?usp=sharing>

**Выводы**

В результате выполнения лабораторной работы было проведено исследование рекуррентных нейронных сетей. Построены различные архитектуры – RNN, GRU, LSTM, с применением различных оптимизаторов – Adam, SGD, Momentum, RMSprop. Изучены методы работы моделей и оптимизаторов, определены оптимальные значения гиперпараметров для данного набора данных.