



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ** ИУК «Информатика и управление»

**КАФЕДРА** ИУК4 «Программная инженерия»

## **Лабораторная работа №2**

### **«Рекуррентные нейронные сети. Улучшенные методы рекуррентных нейронных сетей»**

**ДИСЦИПЛИНА: «Методы глубокого обучения»**

Выполнил: студент гр. ИУК4-21М \_\_\_\_\_ ( Сафронов Н.С. )  
(подпись) (Ф.И.О.)

Проверил: \_\_\_\_\_ ( Белов Ю.С. )  
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

**Цель работы:** получение практических навыков построения глубоких рекуррентных нейронных сетей.

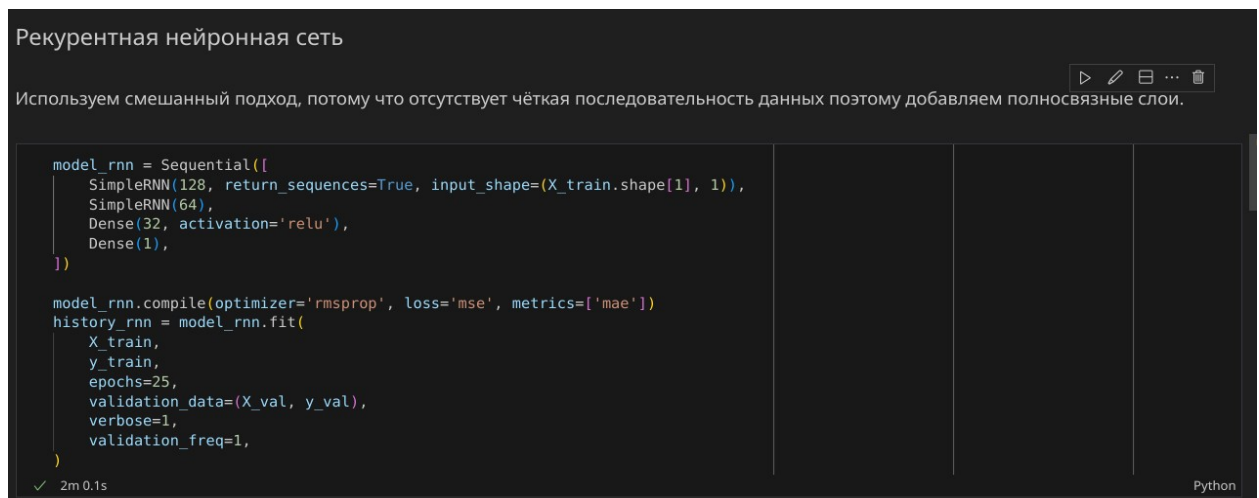
**Задачи:**

1. Реализовать модель глубокой рекуррентной нейронной сети, используя набор данных, согласно варианту.
2. Вывести График точности на этапах обучения.
3. Вывести График изменения по набору данных.
4. Вывести График потерь на этапах обучения и проверки простой полносвязной сети в задаче.

**Вариант 2**

Динамика продаж в супермаркете. Ссылка для скачивания:  
[https://drive.google.com/file/d/1T3C0WsA9p6MPyuKG7naFrSPxtQroD6dj/view?usp=drive\\_link](https://drive.google.com/file/d/1T3C0WsA9p6MPyuKG7naFrSPxtQroD6dj/view?usp=drive_link)

**Результаты выполнения работы**



Рекуррентная нейронная сеть

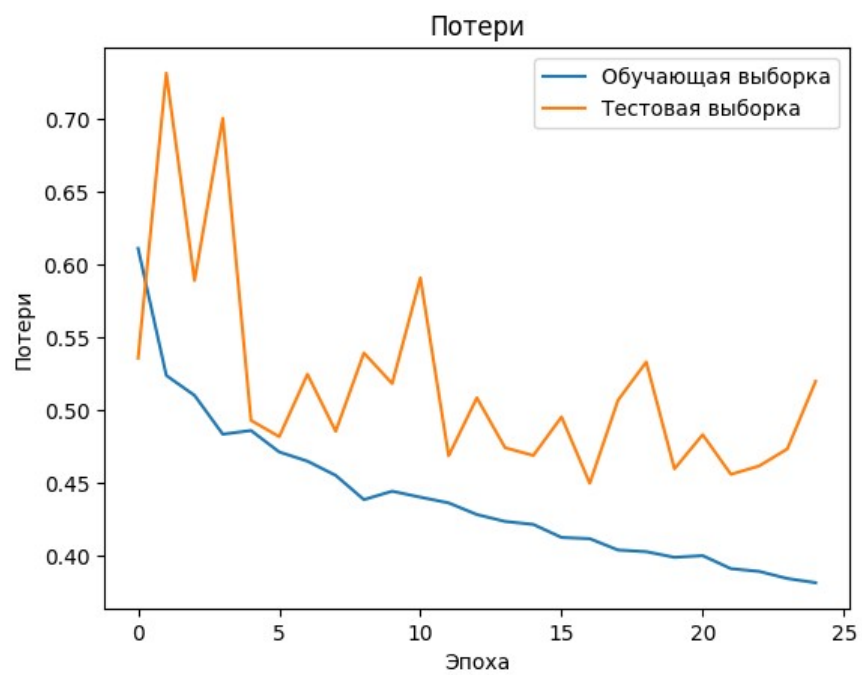
Используем смешанный подход, потому что отсутствует чёткая последовательность данных поэтому добавляем полносвязные слои.

```
model_rnn = Sequential([
    SimpleRNN(128, return_sequences=True, input_shape=(X_train.shape[1], 1)),
    SimpleRNN(64),
    Dense(32, activation='relu'),
    Dense(1),
])

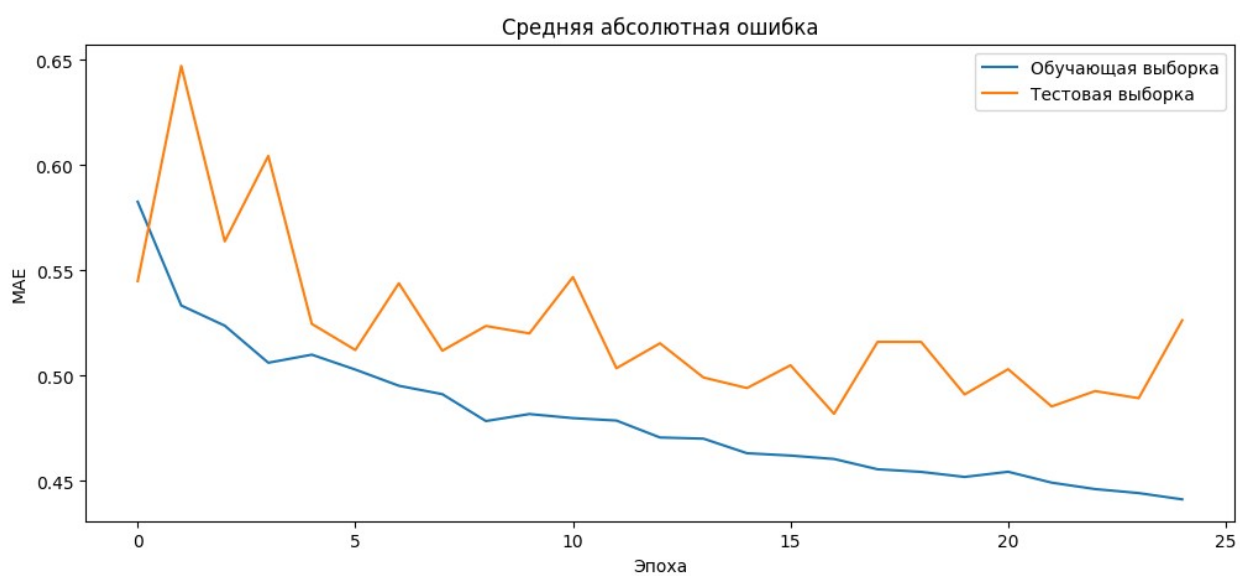
model_rnn.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
history_rnn = model_rnn.fit(
    X_train,
    y_train,
    epochs=25,
    validation_data=(X_val, y_val),
    verbose=1,
    validation_freq=1,
)
```

✓ 2m 0.1s Python

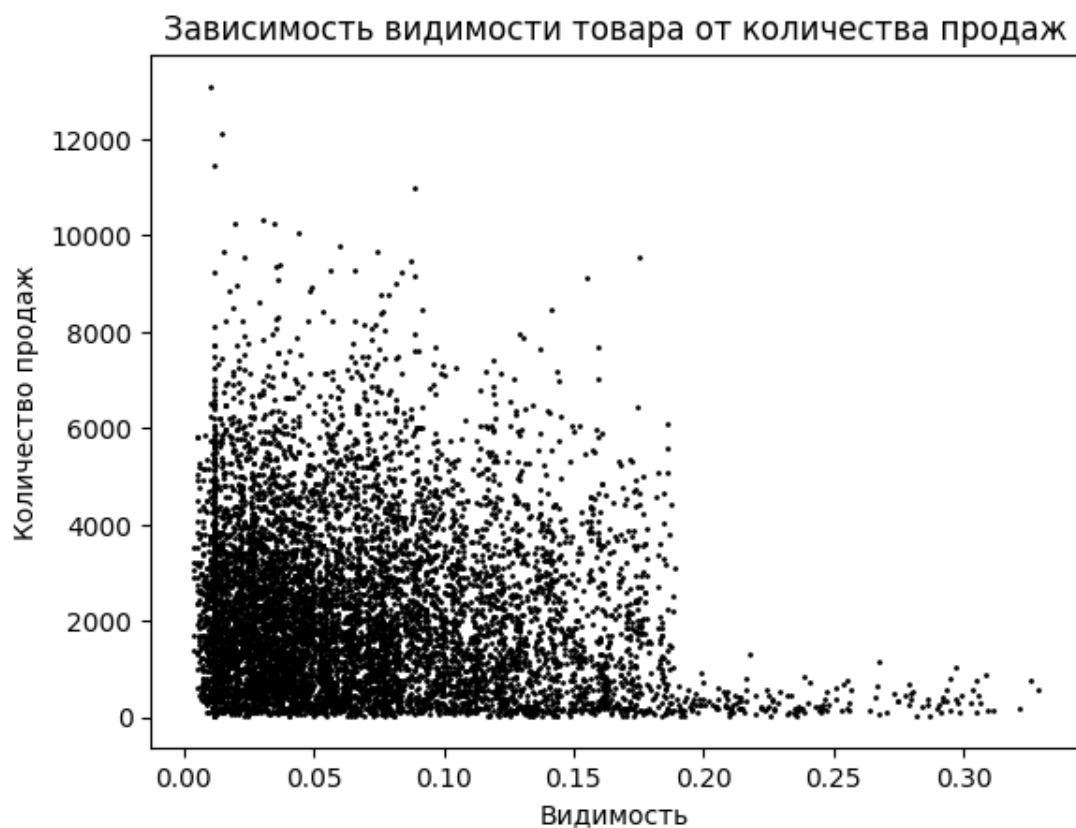
**Рисунок 1** – Архитектура созданной нейронной сети



**Рисунок 2 – График потерь**



**Рисунок 3 – Средняя абсолютная ошибка**



**Рисунок 4** – Зависимость видимости товара от количества его продаж

```

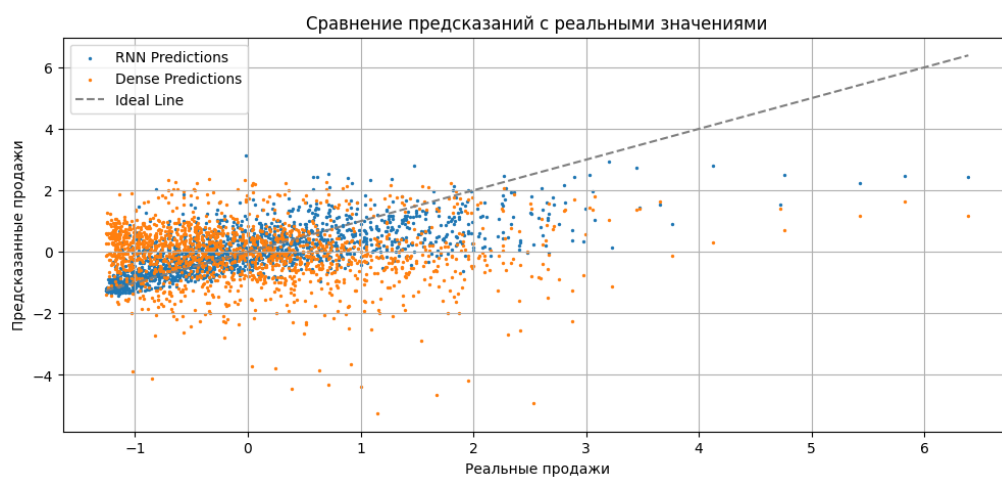
model_dense = Sequential([
    Dense(128, input_shape=(X_train.shape[1],)),
    Dense(32),
    Dense(1)
])

model_dense.compile(optimizer='adam', loss='mse', metrics=['mae'])
history_rnn = model_rnn.fit(X_train, y_train, epochs=25, validation_data=(X_val, y_val), verbose=1)

```

Python

**Рисунок 5** — Архитектура полносвязной сети для сравнения с ней



**Рисунок 6** – Сравнение предсказаний с реальными данными

**Вывод:** в ходе выполнения работы были получены практические навыки построения глубоких рекуррентных нейронных сетей с применением глубокого обучения.