****

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ

Кафедра программных систем

Дисциплина

Нейронные сети

ОТЧЕТ

по лабораторной работе №3

Создание простейшей нейронной сети  
«Использование библиотеки Keras»

Студент: Соколова А.Д.

Группа: 6301-020302D

Проверил:

профессор Тюгашев А.А.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара  
2025

СОДЕРЖАНИЕ

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc197898368)

[1 Исходный текст программы 4](#_Toc197898369)

[2 Протокол исполнения 6](#_Toc197898370)

[ЗАКЛЮЧЕНИЕ 7](#_Toc197898371)

# ПОСТАНОВКА ЗАДАЧИ

Целью данной работы является реализация простой полносвязной нейронной сети с использованием библиотеки Keras для задачи классификации отзывов на фильмы.

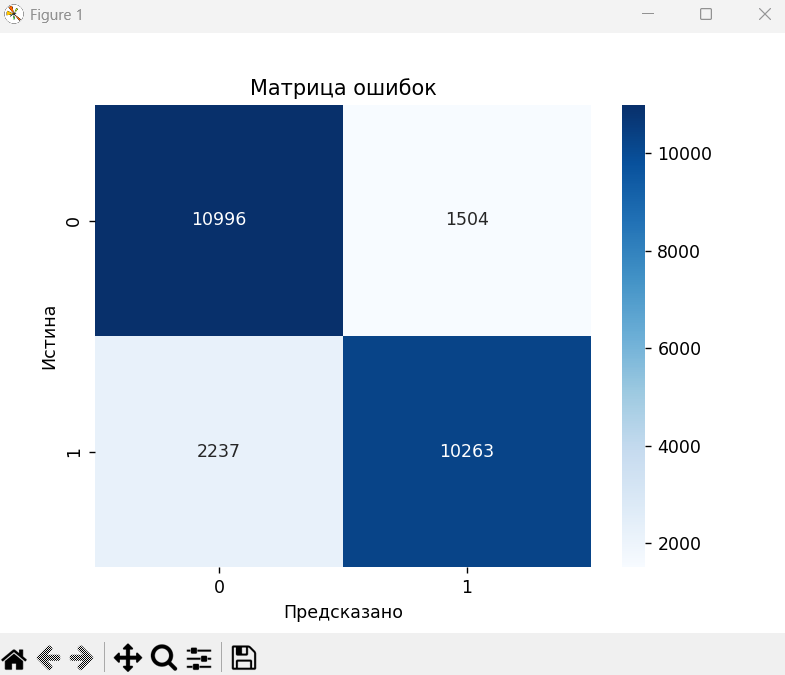
В рамках выполнения работы предполагается выполнение следующих задач:

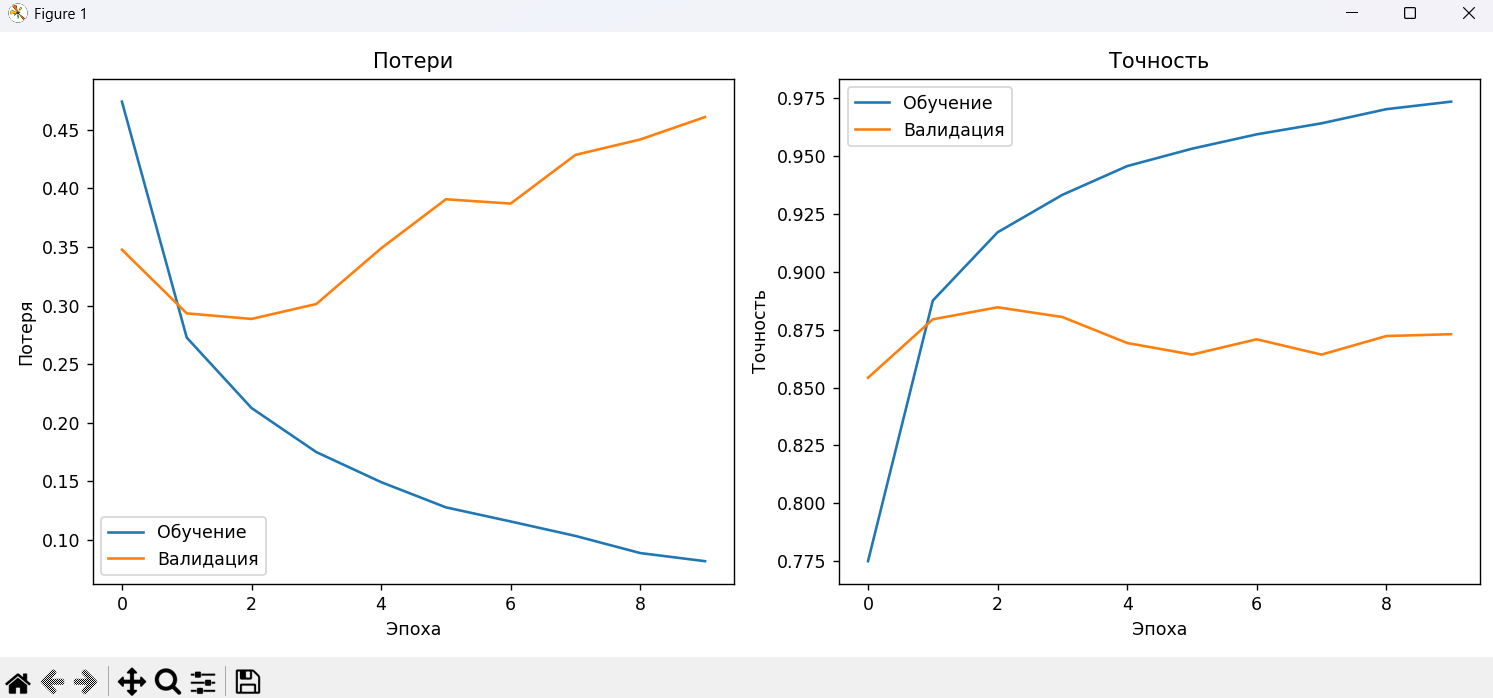
1. загрузка и предобработка текстовых данных (IMDb Reviews);
2. построение и обучение модели нейронной сети;
3. анализ работы модели и визуализация результатов.
4. Исходный текст программы

import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.metrics import confusion\_matrix, classification\_report  
from tensorflow import keras  
from tensorflow.keras import layers  
from tensorflow.keras.datasets import imdb  
from tensorflow.keras.preprocessing.sequence import pad\_sequences  
  
*# 1. загрузка и предобработка датасета IMDb*max\_words = 10000  
maxlen = 200  
  
(x\_train, y\_train), (x\_test, y\_test) = imdb.load\_data(num\_words=max\_words)  
  
x\_train = pad\_sequences(x\_train, maxlen=maxlen)  
x\_test = pad\_sequences(x\_test, maxlen=maxlen)  
  
*# 2. построение модели*model = keras.Sequential([  
 layers.Embedding(input\_dim=max\_words, output\_dim=64, input\_length=maxlen),  
 layers.GlobalAveragePooling1D(),  
 layers.Dense(32, activation='relu'),  
 layers.Dense(1, activation='sigmoid')  
])  
  
model.compile(optimizer='adam',  
 loss='binary\_crossentropy',  
 metrics=['accuracy'])  
  
*# 3. обучение модели*history = model.fit(x\_train, y\_train,  
 epochs=10,  
 batch\_size=32,  
 validation\_split=0.2,  
 verbose=1)  
  
*# 4. оценка на тестовой выборке*loss, accuracy = model.evaluate(x\_test, y\_test)  
print(f"Точность на тесте: {accuracy:.4f}")  
  
*# 5. предсказания*y\_pred\_probs = model.predict(x\_test)  
y\_pred = (y\_pred\_probs > 0.5).astype("int32")  
  
*# 6. матрица ошибок*cm = confusion\_matrix(y\_test, y\_pred)  
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')  
plt.title("Матрица ошибок")  
plt.xlabel("Предсказано")  
plt.ylabel("Истина")  
plt.show()  
  
*# 7. график потерь и точности*plt.figure(figsize=(12, 5))  
plt.subplot(1, 2, 1)  
plt.plot(history.history['loss'], label='Обучение')  
plt.plot(history.history['val\_loss'], label='Валидация')  
plt.title("Потери")  
plt.xlabel("Эпоха")  
plt.ylabel("Потеря")  
plt.legend()  
  
plt.subplot(1, 2, 2)  
plt.plot(history.history['accuracy'], label='Обучение')  
plt.plot(history.history['val\_accuracy'], label='Валидация')  
plt.title("Точность")  
plt.xlabel("Эпоха")  
plt.ylabel("Точность")  
plt.legend()  
plt.tight\_layout()  
plt.show()

i

1. Протокол исполнения

  
Рисунок 1 – Матрица ошибок

  
Рисунок 2 – График потерь (loss) и точности (accuracy)

# ЗАКЛЮЧЕНИЕ

В процессе выполнения лабораторной работы была проведена классификация текстов с использованием нейронной сети, обученной на наборе данных IMDb для предсказания положительных или отрицательных рецензий на фильмы. Модель была обучена на 10 эпохах, и в процессе обучения была достигнута высокая точность на обучающих данных (97.73%) с уменьшением значения потерь. Валидационная точность колебалась в пределах 86-87%, что свидетельствует о хорошем качестве модели на независимом наборе данных.

На тестовых данных модель продемонстрировала точность около 84,98%, что указывает на отличные результаты в обобщении и способности корректно классифицировать рецензии, не встречавшиеся в процессе обучения. Результаты эксперимента подтверждают эффективность модели для решения задачи бинарной классификации текста.