**Практическая работа №7**

**Задание:**

Реализовать нейронную сеть для определения тональности отзывов фильмов

Пункты 1-8 – 4 балла

Пункты 9-10 – 2 балла

**Решение:**

1. В качестве данных загрузить набор IMDb movie review из keras:

from tensorflow.keras.datasets import imdb

(x\_train,y\_train),(x\_test,y\_test) = imdb.load\_data(num\_words=max\_words)

Здесь max\_words означает, какие первые max\_words частых слов будут отражены в текстах. Сами тексты будут представлены списками чисел, где каждое число – это номер слова в частотном словаре всех слов текстов.

Частоту слов можно получить с помощью следующего кода:

dictionary = imdb.get\_word\_index()

Каждый отзыв имеет либо положительную оценку, либо отрицательную (1/0).

1. Далее необходимо все тексты привести к одной размерности. Для этого необходимо использовать функцию pad\_sequences, где указывается финальная размерность с помощью параметра maxlen. Параметр padding показывает, куда добавлять 0, если длина исходного текста была меньше maxlen.
2. Создать модель типа Sequential, куда будут добавляться слои. Далее добавить Dense слои, в конце обязательно должен быть Dense слой с одним нейроном (так как задача бинарной классификации и на выходе будет значение 0 или 1, в качестве функции активации лучше использовать сигмоидальную функцию, так как она лучше всего подходит к этой задаче). Количество скрытых слоев и их параметры задайте самостоятельно.
3. Настройте модель с помощью метода compile. В качестве функции потерь используйте binary\_crossentropy, а в качестве метрики accuracy.
4. Далее обучите модель на тренировочных данных и постройте графики обучения модели на разных эпохах:

plt.plot(history.history['accuracy'], label = 'Доля верных ответов на тренировочной обучающей выборке')

plt.plot(history.history['val\_accuracy'], label = 'Доля верных ответов на тренировочной проверяющей выборке')

plt.xlabel('Эпоха обучения')

plt.ylabel('Доля правильных ответов')

plt.legend()

plt.show()

Где history – это результат работы метода fit модели (history = model.fit(…))

1. Проверьте полученную модель на тестовых данных. Объясните результаты.
2. Теперь для тех же данных постройте другую модель. В нее необходимо включить слой Embedding (его параметры: кол-во слов (max\_words), длина вектора слова (итоговая), длина входных векторов текстов (maxlen), например, Embedding(max\_words,2,input\_length=maxlen)), слой SimpleRNN (с кол-вом нейронов в нем) и Dense (последний, такой же, как и в первой части). Точно также настроить модель, обучить ее и построить графики обучения.
3. Постройте разные вариации модели (измените параметры при загрузке данных (max\_words, maxlen) и параметры слоев модели. Сравните результаты.
4. Постройте модели с использованием LSTM или GRU. Сравните результаты.
5. Ответьте на вопросы:

Как работают рекуррентные нейронные сети?

В чем преимущества LSTM и GRU перед простыми РНС?