

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №7
по дисциплине «Искусственные нейронные сети»
Тема: Классификация обзоров фильмов

Студентка гр. 8383

Максимова А.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

Классификация последовательностей - это проблема прогнозирующего моделирования, когда у вас есть некоторая последовательность входных данных в пространстве или времени, и задача состоит в том, чтобы предсказать категорию для последовательности.

Проблема усложняется тем, что последовательности могут различаться по длине, состоять из очень большого словарного запаса входных символов и могут потребовать от модели изучения долгосрочного контекста или зависимостей между символами во входной последовательности.

В данной лабораторной работе также будет использоваться датасет IMDb, однако обучение будет проводиться с помощью рекуррентной нейронной сети.

Задачи

- Ознакомиться с рекуррентными нейронными сетями
- Изучить способы классификации текста
- Ознакомиться с ансамблированием сетей
- Построить ансамбль сетей, который позволит получать точность не менее 97%

Требования

1. Найти набор оптимальных ИНС для классификации текста
2. Провести ансамблирование моделей
3. Написать функцию/функции, которые позволят загружать текст и получать результат ансамбля сетей
4. Провести тестирование сетей на своих текстах (привести в отчете)

Основные теоретические положения

Датасет IMDb:

Датасет, состоящий из 50 000 отзывов на фильмы от пользователей, помеченных как положительные (1) и отрицательные (0).

- Рецензии предварительно обрабатываются, и каждая из них кодируется последовательностью индексов слов в виде целых чисел.
- Слова в обзорах индексируются по их общей частоте появления в датасете. Например, целое число "2" кодирует второе наиболее частое используемое слово.
- Датасет разделен на два набора: 25 000 для обучения и 25 000 на тестирование.

Рекуррентные нейронные сети:

Проблемой, возникающая при анализе текста с помощью полносвязанных нейронных сетей, является невозможность обработки текста как последовательности токенов. Решением является использование рекуррентных нейронных сетей, основное отличие которых заключается в использовании циклов, позволяющих учитывать информацию о том, что было получено на прошлых шагах работы нейронной сети. Таким образом, порядок символов, слов или предложений, имеющих большой смысл в тексте, учитывается.

Рекуррентную нейронную сеть можно представить в виде сети с прямым распространением сигнала для чего используется прием, называемый разворачиванием во времени.

Рекуррентная нейронная сеть может работать в нескольких режимах:

- many to many: на вход подается последовательность, на выходе также возвращается последовательность (используется при автопереводах или генерации текста)
- many to one: вход - последовательность, выход - значение (задачи классификации, например, сентимент-анализ)

- one to many: например, описание изображений
- one to one: нелинейный вычисления, редко используется

Проблемами, возникающими при работе с рекуррентными нейронными сетями, являются: обучение требует длительного времени и больших вычислительных ресурсов, сигнал об уменьшении весов при передаче от слоя к слою уменьшается, ограниченная длительность запоминания предыдущей информации. Решение - использование более сложных архитектур рекуррентных нейронных сетей, например, LSTM и GRU.

Сеть LSTM:

Долгая краткосрочная память; элементом сети является набор из четырех слоев, взаимодействующих друг с другом, по определенным правилам, который называется ячейкой. Плюс: нет ограничения длительности запоминания предыдущей информации. Минус: состоит из множества элементов, поэтому чтобы обучить такую искусственную нейронную сеть необходимы большие вычислительные ресурсы.

Embedded:

Специальный слой, позволяющий экономить память при хранении обучающей выборки. Каждому токenu сопоставляется плотное векторное представление, определяющееся во время обучения нейронной сети.

Ансамблирование нейронных сетей:

Ансамблирование является методом улучшения результатов работы нейронных сетей в решении задач. Суть методов ансамблирования заключается в объединении прогнозов, полученных набором разных моделей, для получения лучшего прогноза.

Выполнение работы

1. Были импортированы все необходимые для работы классы и функции.

```
import numpy as np
import matplotlib.pyplot as plt

from keras.datasets import imdb
from keras.models import Sequential

from keras.layers import Dense
from keras.layers import Conv1D, MaxPooling1D
from keras.layers import LSTM, GRU
from keras.layers import Dropout

from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence

from tensorflow.keras.models import load_model
```

2. Был загружен встроенный в Keras датасет IMDb. Для изменения исходного разбиения (50/50) отношения обучающих и контрольных данных, загруженные данные были объединены с помощью метода concatenate() для последующего разделения в пропорции 80/20.

```
def loadDataIMDb(max_words):
    (training_data, training_targets), (testing_data, testing_targets) = imdb.load_data(num_words=max_words)
    data = np.concatenate((training_data, testing_data), axis=0) # соединяем массивы вдоль указанной оси
    targets = np.concatenate((training_targets, testing_targets), axis=0)
    return data, targets
```

3. Обучающие и тестовые данные были разделены в пропорции 80/20. Исходные последовательности были усечены / дополнены до заданного размера.

```
def editData(data, targets, max_review_length):
    sep = (len(data) // 10) * 8

    X_train, Y_train = data[:sep], targets[:sep] # 80 %
    X_test, Y_test = data[sep:], targets[sep:] # 20 %

    X_train = sequence.pad_sequences(X_train, maxlen=max_review_length)
    X_test = sequence.pad_sequences(X_test, maxlen=max_review_length)

    return X_train, Y_train, X_test, Y_test
```

Модели ИНС:

4. После была определена функция для создания *первой модели ИНС*, состоящей из 6 слоев:

- первый (входной слой) - Embedding с параметрами: input_dim (число слов в словаре) = 10000, output_dim (число выходов в Embedding слое) = 32, input_length (размер входного вектора) = 500;
- второй (скрытый) - слой долгосрочной краткосрочной памяти (рекуррентный), содержащий 100 нейронов;
- третий и пятый (скрытые слои) - Dropout, используемые для предотвращения переобучения ИНС;
- четвертый (скрытый) - полносвязанный слой с 50 нейронами, используется полулинейная функция активации Relu: $\max(0, x)$;
- шестой (выходной) - содержит 1 нейрон, функция активации

Sigmoid: $\frac{1}{1 + e^{-x}}$, выдающая значения из диапазона от 0 до 1.

```
def buildFirstModel(max_words, embedding_vector_length, max_review_length):
    model = Sequential()

    model.add(Embedding(input_dim=max_words,
                        output_dim=embedding_vector_length, input_length=max_review_length))
    # преобразование в плотные векторные представления
    model.add(LSTM(100))
    model.add(Dropout(0.4))
    model.add(Dense(50, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

5. *Вторая модель ИНС*, состоящая из 8 слоев:

- первый (входной слой) - Embedding с параметрами: input_dim (число слов в словаре) = 10000, output_dim (число выходов в Embedding слое) = 32, input_length (размер входного вектора) = 500;
- второй (скрытый) - сверточный слой со следующими параметрами: 32 выходных фильтра, размер ядра = 3;

- третий (скрытый) - слой объединение одномерных данных, размер pool = 2;
- четвертый (скрытый) - слой долгосрочной краткосрочной памяти (рекуррентный), содержащий 100 нейронов;
- пятый, седьмой (скрытые) - Dropout, используемые для предотвращения переобучения ИНС;
- шестой (скрытый) - слой долгосрочной краткосрочной памяти (рекуррентный), содержащий 50 нейронов;
- восьмой (выходной) - - полносвязанный слой, который содержит 1 нейрон, функция активации Sigmoid: $\frac{1}{1 + e^{-x}}$, выдающая значения из диапазона от 0 до 1.

```
def buildSecondModel(max_words, embedding_vector_length, max_review_length):
    # LSTM и сверточная НС
    model = Sequential()

    model.add(Embedding(input_dim=max_words,
                        output_dim=embedding_vector_length, input_length=max_review_length))
    # преобразование в плотные векторные представления
    model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
    model.add(MaxPooling1D(pool_size=2))
    model.add(LSTM(100, return_sequences=True))
    model.add(Dropout(0.4))
    model.add(LSTM(50))
    model.add(Dropout(0.4))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

6. Третья модель ИНС, состоящая из 6 слоев:

- первый (входной слой) - Embedding с параметрами: input_dim (число слов в словаре) = 10000, output_dim (число выходов в Embedding слое) = 32, input_length (размер входного вектора) = 500;
- второй (скрытый) - слой долгосрочной краткосрочной памяти (рекуррентный), содержащий 64 нейрона;
- третий, пятый (скрытые) - Dropout, используемые для предотвращения переобучения ИНС;

- четвертый (скрытый) - слой долгосрочной краткосрочной памяти (рекуррентный), содержащий 32 нейрона;
- шестой (выходной) - полносвязанный слой, который содержит 1 нейрон, функция активации Sigmoid: $\frac{1}{1+e^{-x}}$, выдающая значения из диапазона от 0 до 1.

```
def buildThirdModel(max_words, embedding_vector_length, max_review_length):
    model = Sequential()

    model.add(Embedding(input_dim=max_words,
                        output_dim=embedding_vector_length, input_length=max_review_length))
    # преобразование в плотные векторные представления
    model.add(LSTM(64, return_sequences=True))
    model.add(Dropout(0.35))
    model.add(LSTM(32))
    model.add(Dropout(0.35))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

7. После было запущено обучение сетей с помощью метода fit (адаптирует модель под обучающие данные), были выведены значения точности работы сети для каждой модели, построены графики ошибок и точности в ходе обучения сети и на валидационных данных, модели были сохранены для дальнейшего использования.

```
def fitModels(all_models, X_train, Y_train, X_test, Y_test):
    numb = 1
    for model in all_models:
        history = model.fit(
            X_train, Y_train,
            epochs=2,
            batch_size=64,
            validation_data=(X_test, Y_test)) # 0.2
        printAcc(model)

        loss = history.history['loss']
        val_loss = history.history['val_loss']

        acc = history.history['accuracy']
        val_acc = history.history['val_accuracy']

        epochs = range(1, len(loss) + 1)

        plotLoss(loss, val_loss, epochs, numb)
        plotAcc(acc, val_acc, epochs, numb)
        model.save('Models/INS_' + str(numb) + '.h5')
        numb = numb + 1
```


8. Для построения графиков ошибок и точности в ходе обучения сети были написаны следующие функции:

```
def plotLoss(loss, val_loss, epochs, numb):
    plt.plot(epochs, loss, label="Training loss", linestyle='--', linewidth=2, color='red')
    plt.plot(epochs, val_loss, "b", label="Validation loss", color='blue')

    plt.title("Training and Validation loss")
    plt.xlabel("Epochs")
    plt.ylabel("Loss")

    plt.legend()
    plt.grid()
    plt.savefig('Gr/Loss_Model_' + str(numb) + '.png', format="png", dpi=240)
    plt.show()

def plotAcc(acc, val_acc, epochs, numb):
    plt.clf()
    plt.plot(epochs, acc, label="Training accuracy", linestyle='--', linewidth=2, color='red')
    plt.plot(epochs, val_acc, "b", label="Validation accuracy", color='blue')

    plt.title("Training and Validation accuracy")
    plt.xlabel("Epochs")
    plt.ylabel("Accuracy")

    plt.legend()
    plt.grid()
    plt.savefig('Gr/Accuracy_Model_' + str(numb) + '.png', format="png", dpi=240)
    plt.show()
```

Построенные модели ИНС

Первая модель ИНС (модель с одним рекуррентным слоем): точность на контрольных данных 89,5%

График потерь сети на обучающих и тестовых данных:

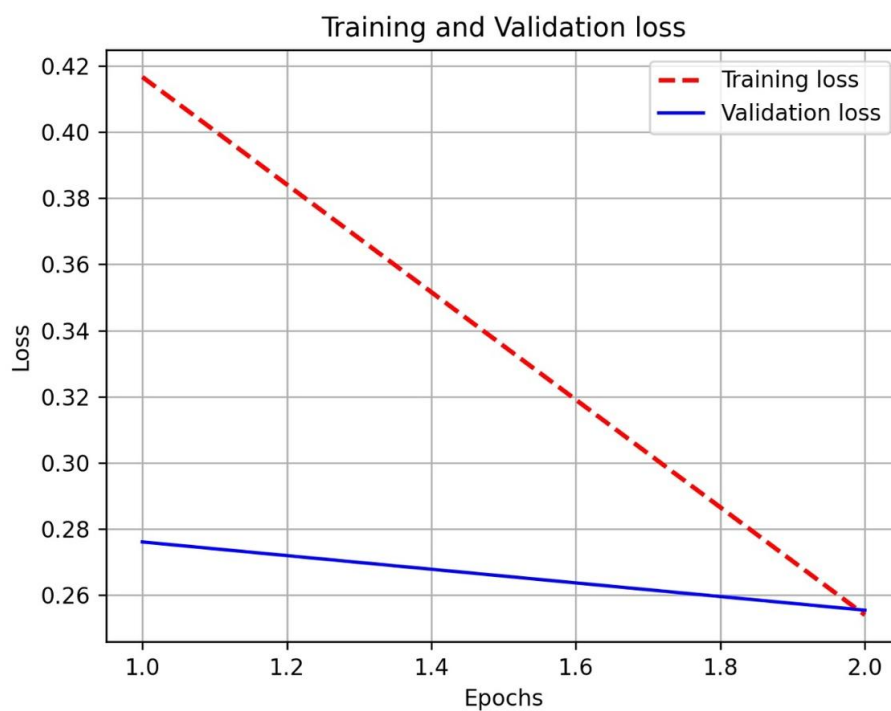
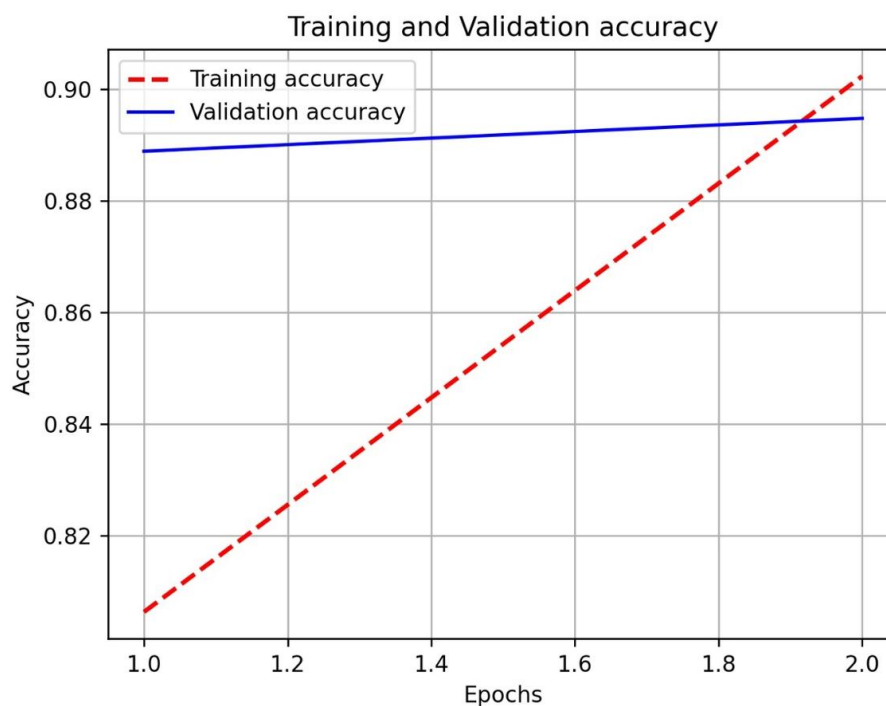


График точности сети на обучающих и тестовых данных:



Вторая модель ИНС (модель с одним сверточным и двумя рекуррентными слоями): точность на контрольных данных 89%

График потерь сети на обучающих и тестовых данных:

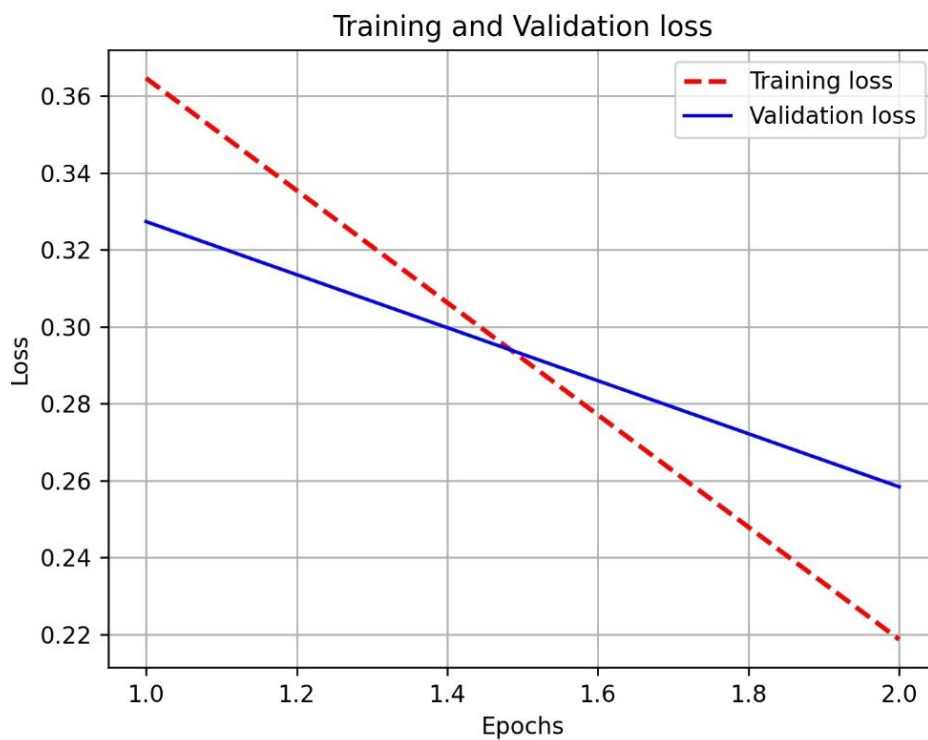
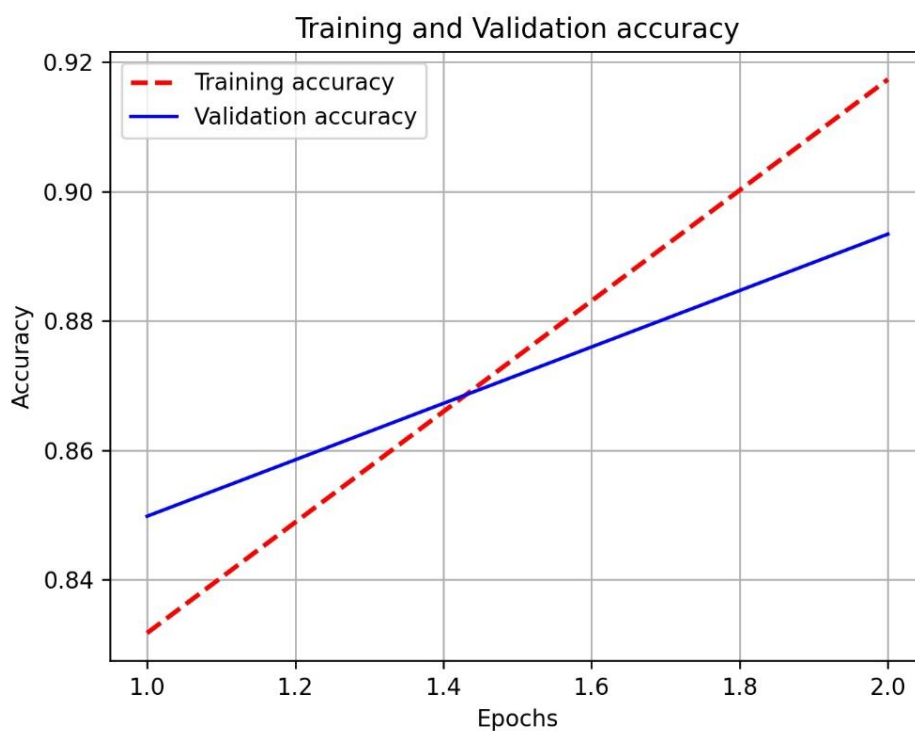


График точности сети на обучающих и тестовых данных:



Третья модель ИНС (модель с двумя рекуррентными слоями): точность на контрольных данных 88%

График потерь сети на обучающих и тестовых данных:

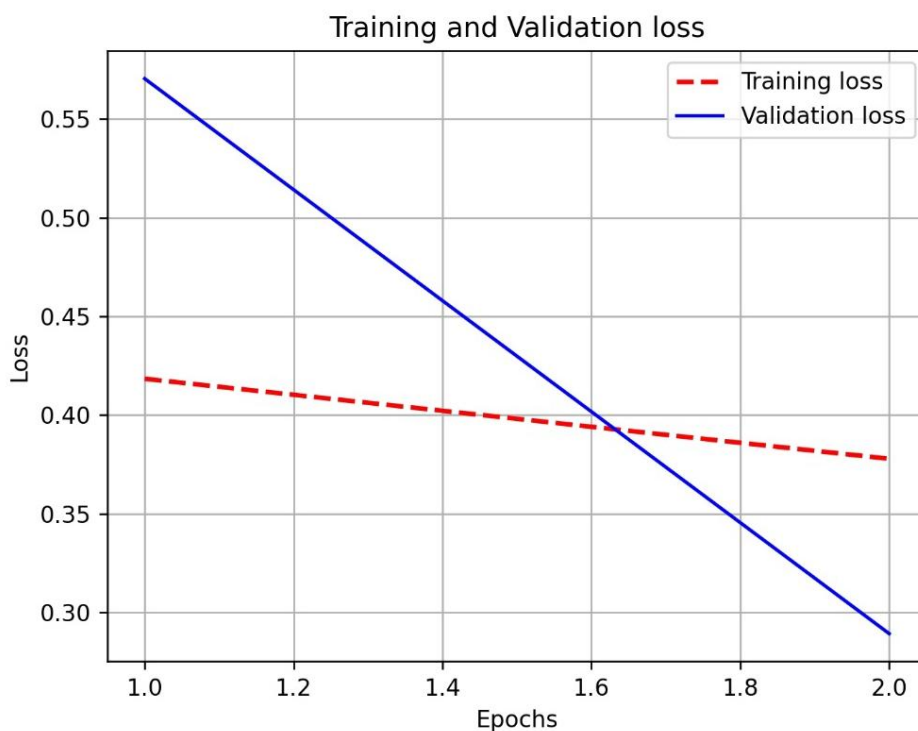
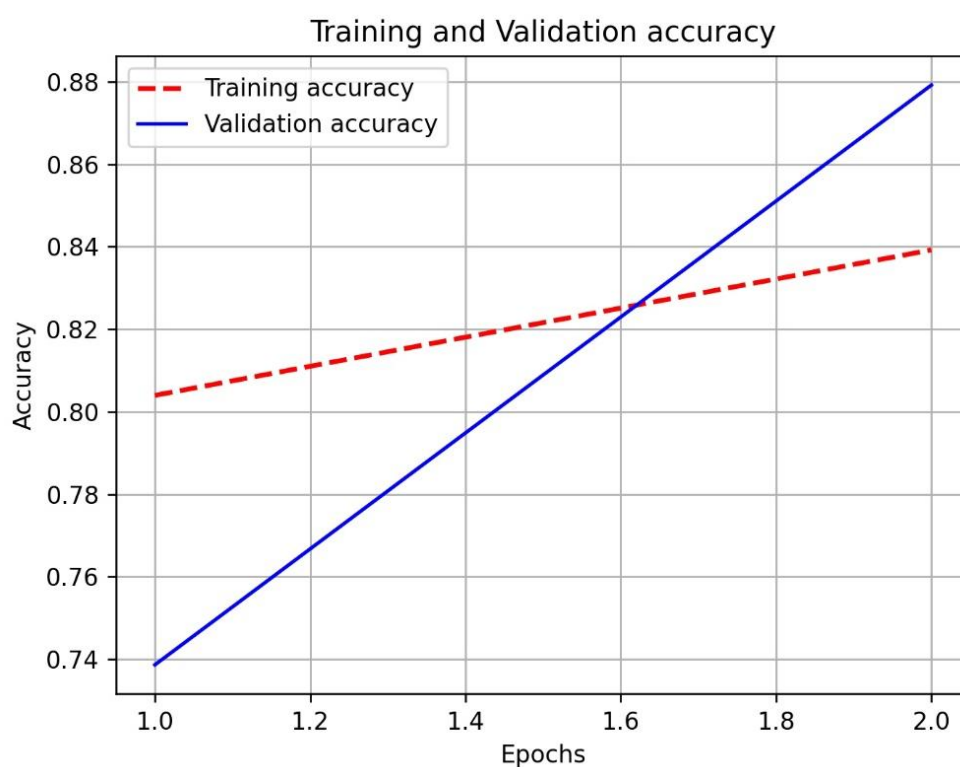


График точности сети на обучающих и тестовых данных:



Ансамблирование моделей ИНС

Была написана следующая функция для ансамблирования всех трех моделей ИНС:

```
def ensemble(validation_data):  
    model1 = load_model("Models/INS_1.h5")  
    model2 = load_model("Models/INS_2.h5")  
    model3 = load_model("Models/INS_3.h5")  
  
    preds_a = model1.predict(validation_data)  
    preds_b = model2.predict(validation_data)  
    preds_c = model3.predict(validation_data)  
  
    final_preds = (preds_a + preds_b + preds_c) / 3  
    return final_preds
```

Были обучены несколько разных алгоритмов на одних и тех же данных. Так как точности работы моделей приблизительно близки, то результат ансамблирования вычислялся как среднее прогнозов.

В результате применения ансамбля получилось достигнуть точности 90.5%, что выше точности каждой из моделей.

Были написаны функции, выполняющие загрузку и обработку пользовательского текста. Для проверки корректности было проведено 5 тестов, 2 полностью негативных отзыва, 2 - положительных и один - нормальный. Полученные прогнозы ансамбля сетей совпали с правильными ответами. Результаты тестирования отзывов представлены ниже.

Тестирование функций, позволяющих загружать пользовательский текст и анализировать его с помощью ансамбля из трех моделей ИНС

Тестирование производилось на англоязычных отзывах разной длины и настроения.

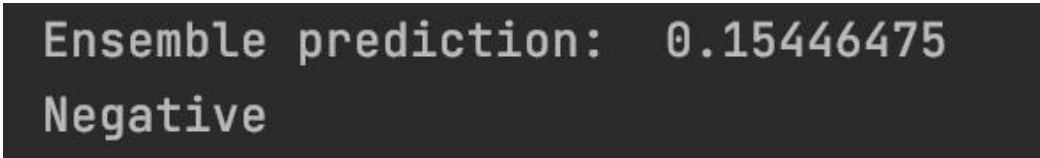
Тест 1

Исходный текст:

This film is not good. The acting is disgusting. But for the beautiful scenery, I will award this film the title of the best of the worst.

Правильный ответ: негативный

Результат:



Ensemble prediction: 0.15446475
Negative

Тест 2

Исходный текст:

As they say, why did I start watching this movie ?! However, the name "Hotel of Psychopaths" ... as they say, he knew what he was doing. It's just awful. By the way, in terms of genre, it is classified as a horror, although this is just a film about a mentally ill man who invited different people to the hotel supposedly for an unforgettable weekend, and he himself "started his game." Shooting is disgusting, low-budget. Acting, well, it's not even a student circle. Funny, among them was Eric Roberts, brother you know who. But the game ... Dialogues from the category: (everyone is half-bloody on the floor) "- Honey, are you pregnant? - Yes, honey. - This is great, I hope after my death, you will be happy again. - Honey, do not die! - No , I'm dying!" Well, at the end, the theatrical closing of the eyes, apparently means that his hero has outlived his life. In general, there is no sanity in the film at all. You look at the behavior of the heroes and it seems that there are absolutely no normal ones there. Perhaps because of a similar acting. And what are the silent grimaces and strained reduction of eyebrows and wrinkles on the forehead of a former psychotherapist. We watched and it all seemed, well, it couldn't be that there was such a movie. And the director could. And the actors did it.

Правильный ответ: негативный

Результат:

Ensemble prediction: 0.13094185
Negative

Тест 3

Исходный текст:

Several days ago I watched a British crime thriller the Legend. Brian Helgeland is the scriptwriter and the director of the film as well. It is adapted from a book The Profession of Violence: The Rise and Fall of the Kray Twins which is based on a real story. The film tells about the life of twin brothers, Reggie and Ronnie Kray, who were violent and vulgar gangsters. They also were iconic figures in the criminal environment of London in 1960s. They headed the most influential gang of bandits of the East-End. They strongarmed, attempted assassination and killed several crime bosses. They also owned a nightclub where came even Hollywood stars. However, it is not easy to be a criminal and it is impossible to give up the crime. It destroyed their lives and they both ended in jail and died alone having lost everything they had and loved. The film is very fascinating and it is a pleasure to watch it even if the plot is rather cruel and heavy. Each part of the movie completes the picture. When you watch it, you dig into the atmosphere by means of sounds, music, costumes, and decorations. Everything is well-orchestrated and the actor that plays the main role is extremely talented. Tom Hardy plays both brothers which have absolutely different characters and personalities. And Hardy acts fantastically! The Legend is definitely one of the greatest films I have ever seen. It can win all hearts!

Правильный ответ: положительный

Результат:

Ensemble prediction: 0.9879996
Positive

Тест 4

Исходный текст:

I really liked this film, despite the genre. I don't like drama, but this movie made an impression on me. I would especially like to note the wonderful play of the actors.

Правильный ответ: положительный

Результат:

```
Ensemble prediction: 0.85347867  
Positive
```

Тест 5

Исходный текст:

After watching this film, I had mixed feelings. On the one hand, there are beautiful landscapes, a lot of funny and witty jokes. On the other hand, the direct, almost terrible acting and the film's small budget. I would say that the film can be called a good one. But it's definitely not the best I've seen. I would not watch this film a second time. Too naive, and in some places and absurd plot.

Правильный ответ: нормальный

Результат:

```
Ensemble prediction: 0.43161824  
Negative
```

Выводы

В результате выполнения лабораторной работы были реализованы три модели ИНС, каждая из которых имела хотя бы один рекуррентный слой. Было выполнено ансамблирование данных моделей, в результате чего получилось немного повысить точность работы сети. Были написаны функции для загрузки пользовательского текста и их последующего анализа с помощью ансамбля

сетей. Было проведено тестирование, результаты предсказания ансамбля оказались правильными во всех случаях.