Министерство образования Республики Беларусь
Учреждение образования
"Белорусский Государственный университет информатики
и радиоэлектроники"

Лабораторная работа №7

“**Рекуррентные нейронные сети для анализа текста”**

по учебной дисциплине “Машинное обучение”

Выполнил: Студент гр. 956241 Дубовик Н.О.

Минск 2020

Данные: Набор данных для предсказания оценок для отзывов, собранных с сайта imdb.com, который состоит из 50,000 отзывов в виде текстовых файлов. Отзывы разделены на положительные (25,000) и отрицательные (25,000). Данные предварительно токенизированы по принципу “мешка слов”, индексы слов можно взять из словаря (imdb.vocab). Обучающая выборка включает в себя 12,500 положительных и 12,500 отрицательных отзывов, контрольная выборка также содержит 12,500 положительных и 12,500 отрицательных отзывов, а также. Данные можно скачать по ссылке **https://ai.stanford.edu/~amaas/data/sentiment/**

Результат выполнения заданий опишите в отчете.

Задание 1.

Загрузите данные. Преобразуйте текстовые файлы во внутренние структуры данных, которые используют индексы вместо слов.

Данные были загружены с помощью библиотеки tensorflow_datasets

```
dataset, info = tfds.load('imdb_reviews/subwords8k', with_info=True,
                          as_supervised=True)
train_dataset, test_dataset = dataset['train'], dataset['test']
encoder = info.features['text'].encoder
train_dataset = train_dataset.shuffle(1000).padded_batch(BATCH_SIZE, padded_s
hapes=((None,), ()))
test_dataset = test_dataset.padded_batch(BATCH_SIZE, padded_shapes=((None,),
 ()))
```

Задание 2.

Реализуйте и обучите двунаправленную рекуррентную сеть (LSTM или GRU). Какого качества классификации удалось достичь?

Созданная нейронная сеть состоит из слоя LSTM. Выходной слой состоит из 1 нейрона с функцией активацией “sigmoid”. Использовался оптимизатор Adam и функция потерь binary_crossentropy. Набор данных обучался в течение 20 эпох.

```
model_1 = tf.keras.Sequential([
    tf.keras.layers.Embedding(encoder.vocab_size, 100),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
    tf.keras.layers.Dense(1,activation='sigmoid')
])
```

```
model_1.compile(optimizer='adam',
   loss='binary_crossentropy',
   metrics=['accuracy'])
```

Были получены следующие результаты: точность на обучающей выборке составляет 98,98%, 84.53% на валидационной, 84.03% на тестовой.

Задание 3.

Используйте индексы слов и их различное внутреннее представление (word2vec, glove). Как влияет данное преобразование на качество классификации?

Было использовано представление glove.

Для этого использовались следующие данные - http://nlp.stanford.edu/data/glove.840B.300d.zip

Реализация представлена ниже:

```
embeddings_index = {}
with open("glove.840B.300d.txt", "r") as in_file:
   for line in in_file:
      values = line.split()

      try:
         word = values[0]
         embeddings_index[word] = np.asarray(values[1:], dtype=np.float32)
      except:
         pass

embedding_matrix = np.zeros((encoder.vocab_size, 300))

for index, word in enumerate(encoder.subwords, 1):
   word = word.lower()

   if word.endswith("_"):
      word = word[:-1]

   embedding_vector = embeddings_index.get(word)
   if embedding_vector is not None:
      embedding_matrix[index] = embedding_vector
```

И обучена следующая модель

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(encoder.vocab_size, 300, weights=[embedding_matrix], trainable=False),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(64,  return_sequences=True), merge_mode='concat'),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(64), merge_mode='concat'),
    tf.keras.layers.Dense(64, activation='elu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(1)
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

В ходе чего были получены следующие результаты: 85.28% на обучающей выборке и 86.46% на валидационной.

Задание 4.

Поэкспериментируйте со структурой сети (добавьте больше рекуррентных, полносвязных или сверточных слоев). Как это повлияло на качество классификации?

Были добавлены 1 рекуррентный слой LSTM и 1 полносвязанный слой с 64 нейронами и функцией активацией relu.

Результаты: точность на обучающей выборке составляет 99,83%, 86,04% на валидационной, 86,76% на тестовой.

Задание 5.

Используйте предобученную рекуррентную нейронную сеть (например, DeepMoji или что-то подобное).

Какой максимальный результат удалось получить на контрольной выборке?

Была использована сеть DeepMoji, для этого использовалась библиотека deepmoji.model_def и ее метод deepmoji_architecture, а также предоставляемый пример, связанный с imdb.

После применения данной сети получилось добиться 82.15% точности на контрольной выборке