

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

ОТЧЕТ
по лабораторной работе №8
по дисциплине «Искусственные нейронные сети»
Тема: Генерация текста на основе “Алиса в стране чудес”

Студентка гр. 8382

Кузина А.М.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы

Создать генеративную модель на основе рекуррентной нейронной сети.

Задачи:

- Ознакомиться с генерацией текста
- Ознакомиться с системой Callback в Keras

Требования:

1. Реализовать модель ИНС, которая будет генерировать текст
2. Написать собственный CallBack, который будет показывать то как генерируется текст во время обучения (то есть раз в какое-то количество эпох генерировать и выводить текст у необученной модели)
3. Отследить процесс обучения при помощи TensorFlowCallback (TensorBoard), в отчете привести результаты и их анализ

Ход работы

Для генерации последовательностей нейронную сеть предварительно нужно обучить. Для этого в программу загружается текст книги «Алиса в стране чудес» и затем преобразуется таким образом, чтобы сеть могла работать с данными. Все символы приводятся к нижнему регистру, затем по ним составляется «словарь», где каждому символу соответствует число-код. Затем по полученному словарю весь текст преобразуется в массив чисел. И затем полученные числа масштабируются так, чтобы попадать в диапазон от 0 до 1.

Исходный текст делится на последовательности из 150 символов. Каждая такая последовательность кодируется и добавляется в список «входных» данных, а символ, находящийся после этих 150 для каждой

последовательности, также кодируется и становится «выходным» для последовательности.

Рассмотрим построенную модель:

```
model = Sequential()
model.add(LSTM(300, input_shape=(X.shape[1], X.shape[2]),
return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(300))
model.add(Dropout(0.3))
model.add(Dense(y.shape[1], activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')

filepath="weights-improvement-{epoch:02d}-{loss:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')
customCB = CustomCallback(10, generate, dataX, int_to_char)
tensorboard_callback = keras.callbacks.TensorBoard(log_dir="./logs",
histogram_freq=1)
callbacks_list = [checkpoint, customCB, tensorboard_callback]

model.fit(X, y, epochs=10, batch_size=128, callbacks=callbacks_list)
```

Помимо уже известных частей модели – слоев LSTM, Dropout и выходного Dense, компиляции модели – появляются callback’и, выполняющие различные задачи:

```
checkpoint = ModelCheckpoint(filepath, monitor='loss', verbose=1,
save_best_only=True, mode='min')
```

Данный callback представляет из себя контрольные точки для модели. В данном случае каждый раз в конце эпохи, если потери сети уменьшились, ее веса будут сохранены. Таким образом в конце обучения сети гарантированно будет сохранена лучшая конфигурация весов, которую можно будет загрузить в программу и не обучать сеть вновь.

```
customCB = CustomCallback(10, generate, dataX, int_to_char)
```

Данный callback написан собственноручно и позволяет с определенной периодичностью, передаваемой в качестве параметра при его создании, генерировать текст на основе определенного паттерна.

```
tensorboard_callback =keras.callbacks.TensorBoard(log_dir="./logs",
histogram_freq=1)
```

Данный callback дает возможность отслеживать процесс обучения сети,

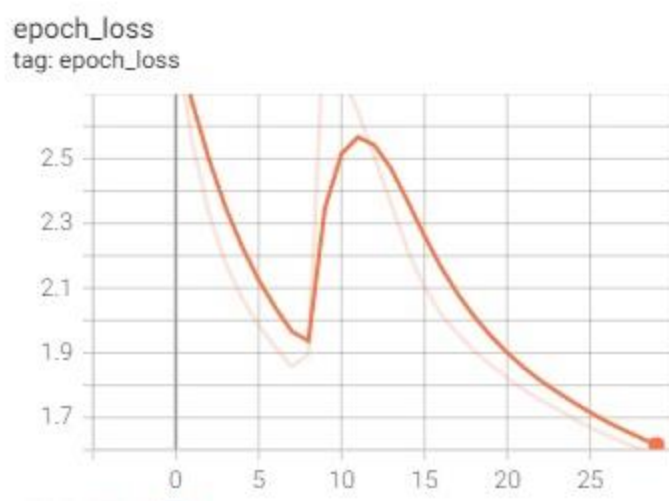
визуализировать архитектуру сети, получать различные диаграммы, связанные с сетью.

Для генерации текста сетью была реализована функция-генератор, также использующаяся в callback'е:

```
def generate(model, dataX, dictionary):
    gen_symbols = []
    start = numpy.random.randint(0, len(dataX) - 1)
    pattern = dataX[start]
    print("Seed:\n" + "\"" + ''.join([dictionary[value] for value in
pattern])) + "\"\n")
    for i in range(1000):
        x = numpy.reshape(pattern, (1, len(pattern), 1))
        x = x / float(n_vocab)
        prediction = model.predict(x, verbose=0)
        index = numpy.argmax(prediction)
        result = dictionary[index]
        gen_symbols.append(result)
        pattern.append(index)
        pattern = pattern[1:len(pattern)]
    print("Generated text:\n" + "\"" + ''.join(gen_symbols) + "\"\n")
```

Сеть предсказывает один следующий символ на основании изначальной последовательности. Затем сгенерированный символ добавляется в конец последовательности и она сдвигается, чтобы сохранить необходимую длину, с «откидыванием» первого символа. Также данные кодируются при работе сети и декодируются при выводе пользователю.

Рассмотрим промежуточные и итоговые результаты работы сети:



Входные данные	Генерация сети – многоточием показано зацикливание сети	Комментарий
oast you find a number of bathing machines in the sea, some children digging in the sand with wooden spades, then a row of lodging houses, and behind	the toe toe ... toe toe	Эпоха 1, в полученной последовательности всего 2 слова, одно из которых бесконечно зациклено
and i could let you out, you know.' he was looking up into the sky all the time he was speaking, and this alice thought decidedly uncivil. 'but perha	ps it manes the morse, and the mors of the morse of the morse, and the mors of the morse of the morse of the morse, and the mors of the morse of the lorse of the lorse, and she was so the soeee of the court of the court, and the soon little should be lote ooe of the court of the court of the court, and the soon ... little should be lote ooe of the court of the court of the court, and the soon little should be lote ooe of the court of the court of the court, and t	Эпоха 11, сеть вначале сделала небольшое зацикливание по одной фразе и затем ушла в бесконечное зацикливание по другой. В сравнении с предыдущим случаем – видны заметные улучшение, фраза стала длиннее, состоять из разных слов
tiful soup, so rich and green, waiting in a hot tureen! who for such dainties would not stoop? soup of the evening, beautiful soup! soup o	f the sween of the sable ...	Эпоха 21 - В данном случае сеть ушла в бесконечное зацикливание из пробелов
wn here, that i should think very likely it can talk: at any rate,	ly rueer the wiine- and i had to be a frng of troe mine ano the woods.'	После обучения сети – наилучший результат

there's no harm in trying.' so she began: 'o mouse, do you know the way out of this	<p>'i don't know the wast't a grod deal, if you don't kike the doorer of you done wo the whine,' said the hryphon, 'i must be a setee way of suieklng to miak and surping the woods see here, and i co so the tha that i had been to tell you the whins.'</p> <p>'i don't know the wast't a grod deal, if you don't kike the sea,' said the ming. 'and the was no oore to see a little suestion, anice whink it was so litted at the could not and she was soeaking on the wood of she houre it a little suestion, anice hooded anl the winle soate whre a little sattering of the wonders to her and fooner the was a little sattering of the wonders to her and fooner the was a little sattering of the wonders, and the went on eown and soall the was so tpiaking about in a lonent to be lonking uhe oarch hare and a little sattering of the wonderant sreees and sooe of the courte of the course with the sueen was so the sabbit, and was suite forg of seem o</p>	<p>работы сети, несмотря на два практически одинаковых предложения сеть не уходит в заикливание, много различных настоящих слов, где-то правильная грамматическая структура предложений, прямая речь выделена кавычками.</p>
---	--	--

Выводы

В данной лабораторной работе была создана рекуррентная нейронная сеть, с помощью которой генерировались символьные последовательности, на основании текста книги «Алиса в стране чудес». После обучения сеть выдает последовательности, которые складываются в слова похожие на английские с опечатками или настоящие слова. Сеть правильно расставляет знаки препинания, связанные с прямой речью, апострофы в словах, в некоторых местах даже правильно выстраивает временные конструкции.

В сети использовалось 3 CallBack'а: один написанный собственноручно для генерации текста по ходу обучения сети, второй для сохранения состояния

сети при улучшении результатов и третий для отслеживания процесса обучения сети визуально.