

УДК 004.855

DOI: 10.17223/19988605/57/13

**Н.А. Кривошеев, Ю.А. Иванова, В.Г. Спицын****АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ КОРОТКИХ ТЕКСТОВ  
НА ОСНОВЕ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ LSTM И SeqGAN**

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-08-00977 А и в рамках Программы повышения конкурентоспособности Томского политехнического университета.*

Проведено исследование качества генерации коротких текстов на основе применения нейронных сетей LSTM и SeqGAN на русском и английском языках. Для обучения нейронной сети используются следующие подходы: оценка максимального правдоподобия и состязательная сеть, генерирующая последовательность (Sequence Generative Adversarial Nets, SeqGAN). В данной работе реализация SeqGAN не включает алгоритм Монте-Карло. Предложен и реализован подход на основе возведения значений выходного вектора нейронной сети (вектора вероятностей) в степень, большую 1, данная операция позволяет увеличить качество генерируемого текста, но снижает его разнообразие. Обучение и тестирование проводятся на основе следующих выборок данных: сборника русских стихов с сайта Stihi.ru и подписей к изображениям на английском языке из выборки COCO Image Captions. Проведена оценка качества генерации текстов на основе метрики BLEU. Приведены примеры сгенерированных текстов. Проанализированы аналогичные решения.

**Ключевые слова:** SeqGAN; генерация текста; состязательное обучение с подкреплением.

Автоматическая генерация текста является сложной и широко исследуемой задачей, которая используется в таких областях, как машинный перевод текстов, создание субтитров, реферирование текстов, генерация подписей к изображениям.

Одной из проблем генерации текстов с помощью LSTM является снижение качества генерируемого текста при увеличении длины последовательности. В данной работе были рассмотрены решения, позволяющие увеличить качество генерации коротких текстов: SeqGAN [1] и LeakGAN [2].

В статье [1] рассматривается модификация генеративно-состязательной нейронной сети для генерации текстов. Основными модификациями являются использование обучения с подкреплением и алгоритма Монте-Карло. В указанном подходе применяется обучение, состоящее из двух этапов: контролируемое обучение (Maximum Likelihood Estimation, MLE, рассматриваемое далее в данной статье), и состязательное обучение с подкреплением, использующее генеративно-состязательную нейронную сеть для динамической оценки качества текста. По результатам, представленным в [1], можно сделать вывод, что использование SeqGAN позволяет улучшить качество генерации текста.

В работе [2] рассматривается генеративно-состязательная нейронная сеть LeakGAN, основанная на объединении генеративно-состязательных нейронных сетей SeqGAN [1] и RankGAN [3]. Рассматривается подход, основанный на утечке памяти из дискриминатора, который позволяет улучшить качество генерируемого текста. В [2] проведено сравнение таких подходов, как MLE, SeqGAN, RankGAN и предлагаемого LeakGAN. По представленным результатам можно сделать вывод, что подходы SeqGAN, RankGAN и LeakGAN позволяют улучшить качество генерируемого текста по сравнению с подходом на основе MLE. Генеративно-состязательная нейронная сеть RankGAN генерирует более качественный результат, чем SeqGAN, но уступает по качеству генерации нейронной сети LeakGAN по метрике BLEU.

Целью данной работы является оценка качества автоматической генерации коротких текстов на основе сети с долгой краткосрочной памятью (Long Short-Term Memory, LSTM) [4, 5]. Для обучения нейронной сети LSTM используется контролируемое обучение на основе метода максимального правдоподобия (MLE) [6], применяется дальнейшее обучение LSTM в составе состязательной сети,

генерирующей последовательность (Sequence Generative Adversarial Net, SeqGAN) [1]. Следует отметить, что в данной работе не используется метод Монте-Карло, предложенный в работе [1]. Применяется пакетное обучение с увеличенным пакетом данных.

Предложен и реализован подход на основе возведения значений вектора вероятностей (выходного вектора нейронной сети) в степень, большую 1; данная операция позволяет увеличить качество генерируемого текста, но снижает его разнообразие.

Длина генерируемых текстов составляет 10 и 20 слов. Для обучения и тестирования нейронных сетей используются следующие выборки данных: сборник русских стихов с сайта Stihi.ru [7] и подписи к изображениям на английском языке из выборки COCO Image Captions [8]. Максимальная длина текстов в выборке с сайта Stihi.ru составляет 10 слов, а в выборке на основе подписей из выборки COCO Image Captions – 20 слов. Применяется пословная генерация текста. Качество генерации текста оценивается с использованием метрики BLEU [9–11]. Проведены анализ и сравнение аналогичных решений [1–3]. Проанализированы работы [12–14]. Для проведения сравнения данные выборки COCO Image Captions взяты из аналога [2]. Данные размещены на сайте [15]. Таким образом, предварительная обработка выборки данных совпадает с реализованной в аналоге [2].

Было проведено обучение и тестирование подходов на основе MLE и SeqGAN. Сделан вывод, что обучение на основе нейронной сети SeqGAN, по сравнению с подходом на основе MLE, позволяет улучшить качество генерации текста по метрике BLEU. Тексты, сгенерированные на основе нейронной сети SeqGAN, сравнимы по качеству с примерами из обучающей выборки по метрике BLEU [9].

В рамках данной работы создана программная реализация исследуемых моделей на языке Python и проведены тестовые эксперименты. В реализации используется библиотека PyTorch. Программная реализация представлена на сайте [16] и основана на работе [17].

## 1. Описание использовавшихся выборок данных

В данной работе используются следующие выборки данных: сборник русских стихов с сайта Stihi.ru [7] и подписи к изображениям из выборки COCO Image Captions [8].

**Выборка стихов на русском языке.** Выборка, содержащая сборник русских стихов с сайта Stihi.ru [7], состоит из 2 414 243 строк, в среднем по 8–9 слов в строке. Максимальная длина примера в данной выборке составляет 10 слов. Была проведена предварительная обработка текстов, состоящая из следующих последовательных этапов:

1. Разбиение текста на примеры (предложения и строки).
2. Составление словаря наиболее часто встречающихся слов.
3. Удаление примеров, содержащих меньше 8 слов.
4. Удаление примеров, содержащих слова, не вошедшие в словарь.
5. Заполнение или сокращение примеров до длины текста, равной 10 словам. Для заполнения текста до заданной длины используется зарезервированное слово `_AGG_` (от слова *aggregate* – англ. «заполнитель»). Для сокращения длины примера удаляется конец предложения.

Из данной выборки удалены все знаки препинания. После предварительной обработки текста выборка содержит 78 661 предложение, средняя длина предложения – 8,2 слова.

В результате предварительной обработки текста был сформирован словарь из 5 000 слов, включающий зарезервированные слова (удаленное слово `_UNK_`, слово-заполнитель `_AGG_` и стартовое слово `_START_`). Набор данных был разделен на обучающую выборку (70 000 примеров) и тестовую выборку (5 000 примеров).

**Выборка подписей из выборки COCO Image Captions.** Набор данных COCO Image Captions [8] содержит изображения и их описания на английском языке. Описания изображений используются для формирования текстовой выборки данных. Большинство предложений данного набора содержит около 10 слов. Максимальная длина примера в данной выборке составляет 20 слов. Выборка данных взята из источника [15], ее предварительная обработка соответствует представленной в статье [2].

Набор данных [15] представлен следующими файлами: словарь (vocab\_cotra.pkl), обучающая выборка (realtrain\_cotra.txt), тестовая выборка (realtest\_coco.txt). В наборе данных содержится 4 837 оригинальных слов. Обучающий набор данных содержит 80 000 примеров, тестовая выборка содержит 5 000 примеров. Авторы статьи [2] указывают, что из выборки удалены слова, упоминающиеся менее 10 раз, а также содержащие их предложения.

## 2. Топология нейронной сети LSTM

В данной работе применяется рекуррентная нейронная сеть LSTM [4]. Используется пословное преобразование текста в вектор с помощью слоя Embedding [18, 19]. Применяется пословная генерация текста с генерацией одного слова на каждом итерационном шаге.

Архитектура нейронной сети представлена на рис. 1. Нейронная сеть состоит из следующих слоев: Embedding, слой LSTM, полносвязный слой нейронов (Dense) и слой с функцией активации Log\_Softmax. Элемент  $X_i$  является индексом слова в словаре на итерации  $i$ . Элемент  $Y_i$  является выходом нейронной сети на итерации  $i$ , после применения операций экспоненты и multinomial [20].

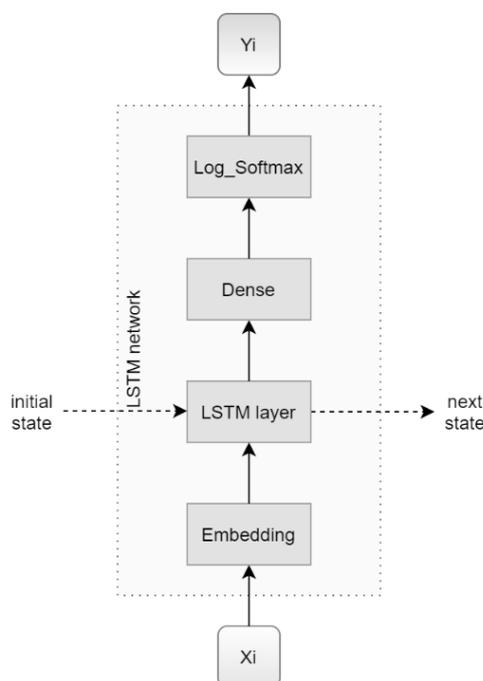


Рис. 1. Архитектура нейронной сети  
Fig. 1. Architecture of the neural network

На каждом итерационном шаге генерации текста происходит инициализация слоя LSTM (initial state). После генерации слова новое состояние (next state) сохраняется и используется для инициализации слоя LSTM на следующем шаге. На первом шаге генерации текста состояние слоя LSTM инициализируется нулями. Полносвязный слой нейронов (с последующим слоем Log\_Softmax) генерирует слово на основе информации из слоя LSTM.

В данной работе длина вектора слова составляет 150 чисел. Слой LSTM содержит 150 нейронов. Количество нейронов в полносвязном слое равняется размеру словаря.

## 3. Описание контролируемого обучения MLE для нейронной сети LSTM и процесса генерации текста

Процесс обучения нейронной сети на основе MLE [6] представлен на рис. 2. Обучение нейронной сети заключается в предсказании некоторой последовательности слов  $[X_1, X_2, \dots, X_t, \dots, X_T]$  (output)

на основе последовательности  $[X_0, X_1, \dots, X_{t-1}, \dots, X_{T-1}]$  (input). В данной работе слово  $X_0$  всегда задается зарезервированным словом `_START_`. На каждом временном шаге проводится инициализация слоя LSTM [4, 5], далее на нейронную сеть LSTM подается слово из входной последовательности (input). Генерируются слово и новое состояние. Состояние нейронной сети сохраняется и передается для инициализации слоя LSTM на следующем шаге.

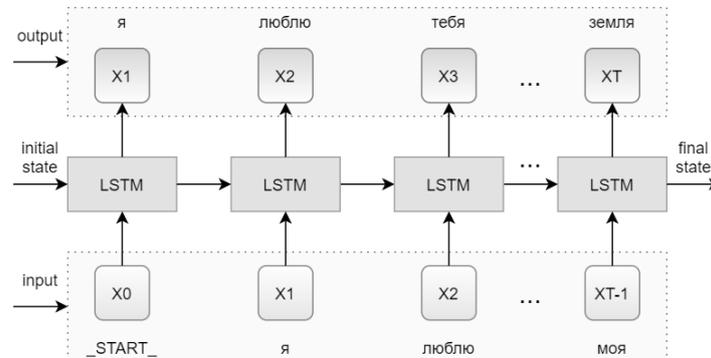


Рис. 2. Процесс обучения нейронной сети  
Fig. 2. Neural network learning process

В процессе генерации текста на нейронную сеть подается некоторая частичная последовательность входных данных. Недостающая часть последовательности восполняется из вывода нейронной сети для генерации полной выходной последовательности (рис. 3). В данной работе на нейронную сеть подается слово  $X_0$ . Новое слово генерируется на основе выхода нейронной сети и операции multinomial [20], которая позволяет выбрать случайное слово на основе распределения вероятностей. Операция multinomial позволяет нейронной сети генерировать случайные тексты.

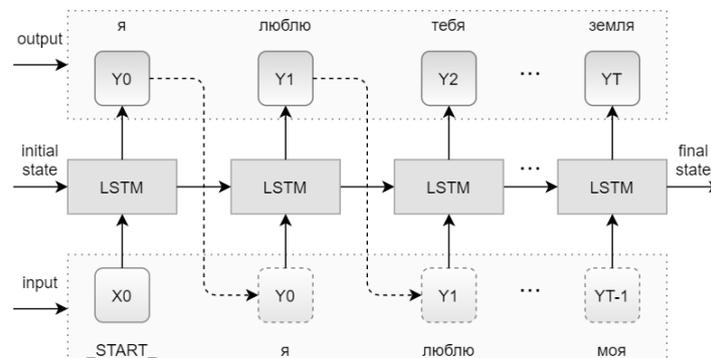


Рис. 3. Процесс генерации текста  
Fig. 3. Text generation process

Таким образом, часть сгенерированной последовательности  $[Y_0, Y_1, \dots, Y_t, \dots, Y_{T-1}]$  является частью входной последовательности  $[X_0, Y_0, Y_1, \dots, Y_t, \dots, Y_{T-1}]$ . Результатом работы нейронной сети является последовательность  $[Y_0, Y_1, \dots, Y_t, \dots, Y_T]$ .

При обучении генератора используется оптимизатор Adam [21].

#### 4. Результаты тестирования нейронной сети LSTM, обученной на основе MLE

Для оценки качества генерации текстов в данной работе используется метрика BLEU, реализованная в библиотеке nltk.

Перед оценкой проводится удаление слова-заполнителя `_AGG_` из текста примера. Данная операция позволяет получить более качественную оценку, так как при оценке не будут учитываться совпадения слов-заполнителей.

При оценке по метрике BLEU используется сглаживание на уровне предложений. Нами применяется метод сглаживания *smoothing 1*, представленный в работе [11]. Данная модификация дает возможность получить более корректные оценки качества текста, но не позволяет сравнить полученные результаты с аналогичными работами, представленными в данной статье, в которых такая модификация не использовалась.

Было проведено тестирование нейронной сети LSTM, обученной на основе MLE на выборках данных: стихи с сайта Stihi.ru [7] и подписи к изображениям из выборки COCO Image Captions [8]. Тестирование проводилось на тестовых выборках, описанных ранее. Для тестирования использовалось 500 сгенерированных примеров. Оценка качества проводилась на основе метрики BLEU. Результаты тестирования представлены в табл. 1:

Таблица 1

**Результаты тестирования LSTM, обученной на основе MLE**

Выборка	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	0,71	0,375	0,187	0,116
COCO Image Captions	0,528	0,393	0,265	0,166

Для оценки качества генерации текста по метрике BLEU было проведено тестирование на основе реальных примеров, написанных людьми, с применением соответствующих обучающих выборок. Полученные результаты используются для последующего сравнения с результатами генерации. Результаты тестирования 500 случайных примеров из обучающих выборок (написанных людьми) на тестовых выборках представлены в табл. 2:

Таблица 2

**Результаты тестирования реальных примеров**

Выборка	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	0,738	0,425	0,224	0,136
COCO Image Captions	0,557	0,443	0,319	0,212

Примеры сгенерированных текстов на основе MLE и текстов из обучающей выборки представлены в табл. 3. Для каждого примера приведена оценка по метрике BLEU [9].

Таблица 3

**Примеры текстов**

Выборка	Тип примеров	№	Пример	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	Из выборки	1	чтоб не вспомнить что есть где то ты	0,943	0,693	0,556	0,437
		2	от того что я с тобой в раю	1,0	0,956	0,707	0,334
		3	пусть в душе моей с каждым днем все сильнее грусть	0,745	0,518	0,211	0,127
	Сгенерированные на основе MLE	1	и тихо мы все вместе с тобой	0,775	0,669	0,599	0,533
		2	и до сих пор в них всё и в ней	0,943	0,693	0,467	0,24
		3	тех кто с такое предал что пишет куда пришла	0,632	0,368	0,163	0,104
COCO Image Captions	Из выборки	1	Several motorcycles are lined up and parked on the side of the street	0,665	0,605	0,529	0,455
		2	A person sitting in the middle of a pile of luggage	0,589	0,513	0,422	0,331
		3	A man wearing a suit , dress shirt and neck tie	0,503	0,348	0,223	0,109
	Сгенерированные на основе MLE	1	A dog laying on top of a desk in front of a TV	0,692	0,664	0,589	0,496
		2	Two people holding remotes together in a living room	0,459	0,388	0,288	0,212
		3	The two beds and a picture of a enjoying from the sky	0,523	0,357	0,228	0,111

Из табл 1, 2 можно увидеть, что качество примеров, сгенерированных нейронной сетью LSTM, обученной на основе подхода MLE, уступает примерам из обучающей выборки, написанным людьми, по метрике BLEU.

## 5. Описание архитектуры SeqGAN

В данной работе обучение на основе SeqGAN [1] является продолжением обучения LSTM на основе MLE. SeqGAN используется для увеличения качества генерации текста.

Нейронная сеть SeqGAN состоит из двух нейронных сетей: генератора и дискриминатора. Целью нейронной сети генератора является генерация текста. В данной работе в качестве нейронной сети генератора выступает предварительно обученная нейронная сеть LSTM, использовавшаяся для обучения с помощью MLE, описанная выше.

Целью нейронной сети дискриминатора является классификация на два класса: реальных образцов текста и сгенерированных. В качестве нейронной сети дискриминатора могут использоваться многие нейронные сети с сигмоидальной функцией активации на выходном слое [22], например сверточная нейронная сеть, рекуррентная нейронная сеть или полносвязная нейронная сеть.

В данной работе в качестве дискриминатора выступает двунаправленная традиционная LSTM [4]. Нейронная сеть дискриминатор состоит из слоя Embedding (длина вектора слова составляет 150 элементов), двух слоев двунаправленной традиционной LSTM (каждый слой состоит из 150 нейронов), скрытого полносвязного слоя (из 150 нейронов с функцией активации relu) и одного полносвязного выходного слоя (с одним нейроном и функцией активации sigmoid).

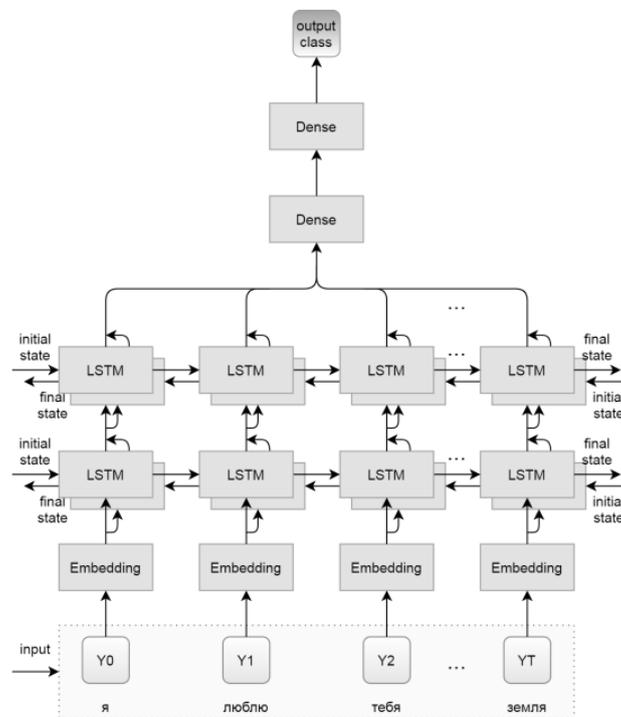


Рис. 4. Топология дискриминатора SeqGAN

Fig. 4. SeqGAN discriminator topology

Топология нейронной сети дискриминатора из SeqGAN [1] представлена на рис. 4. Следует отметить, что обучение генератора и нейронной сети SeqGAN в целом отличается от алгоритма, используемого в традиционной генеративно-состязательной нейронной сети (GAN) [23]. Алгоритм обучения SeqGAN будет описан ниже, информация о данном подходе содержится в статьях [1, 12].

## 6. Обучение SeqGAN

Процесс обучения нейронной сети SeqGAN [1] состоит из следующих этапов:

1. Предварительное обучение генератора на основе MLE.
2. Предварительное обучение дискриминатора.

3. Повторение шагов в цикле:

– обучение генератора на основе обучения с подкреплением (reinforcement learning, RL [24, 25]) в составе SeqGAN;

– обучение дискриминатора.

Первый этап полностью соответствует контролируемому обучению нейронной сети LSTM на основе MLE, которое было описано выше.

На втором этапе необходимо составить выборку для обучения дискриминатора. Данная выборка состоит из реальных примеров и примеров, сгенерированных предварительно обученным генератором (в данной работе размер выборки для обучения дискриминатора составляет 10 000 примеров). На реальные примеры дискриминатор должен учиться выдавать 1, а на сгенерированные генератором 0. Далее происходит обучение дискриминатора. В процессе обучения дискриминатора каждые 10 000 итераций (1 эпоха) проводится обновление обучающей выборки. В результате обучения нейронная сеть должна научиться классифицировать реальные и сгенерированные примеры. От качества обучения дискриминатора зависит качество обучения генератора на третьем этапе (в оригинальной статье [1] авторы советуют проводить наиболее качественное обучение дискриминатора). Обучение нейронной сети дискриминатора проводилось в течение 300 эпох. В данной работе точность нейронной сети дискриминатора составляет 81% на выборке COCO Image Captions [8] и 76% на выборке стихов с сайта Stihi.ru [7]. При обучении дискриминатора используется оптимизатор Adagrad [21], который показал наибольшую стабильность и качество при обучении дискриминатора.

На третьем этапе проводится заданное количество итераций: 25 эпох обучения генератора в составе нейронной сети SeqGAN. Каждая эпоха состоит из следующих шагов:

1. Одна итерация обучения генератора в составе SeqGAN.

2. Обучение дискриминатора в течение 5 эпох по 10 000 итераций на новой сгенерированной обучающей выборке из 10 000 примеров.

Обучение генератора SeqGAN [1] существенно отличается от алгоритма обучения, используемого в традиционной GAN.

Перед вычислением ошибки проводится генерация текста на основе нейронной сети генератора, этап генерации текста представлен в виде псевдокода (рис. 5):

```
# генерация текста
СГЕНЕРИРОВАННЫЙ_ТЕКСТ = ИНИЦИАЛИЗАЦИЯ_МАССИВА(ДЛИНА_ТЕКСТА) # инициализация массива для записи текста
СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ = ИНИЦИАЛИЗАЦИЯ_СОСТОЯНИЯ_LSTM()
ВХОДНОЕ_СЛОВО = ИНДЕКС_СТАРТОВОГО_СЛОВА
i = 0
ПОКА i < ДЛИНА_ТЕКСТА:
    ВЫХОД_НС, СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ = ВЫЧИСЛЕНИЕ_LSTM(ВХОДНОЕ_СЛОВО,
        СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ) # вычислить выход нейронной сети на итерации i
    ВЫХОД_НС_EXP = EXP(ВЫХОД_НС)
    ИНДЕКС_СГЕНЕРИРОВАННОГО_СЛОВА = MULTINOMIAL(ВЫХОД_НС_EXP)
    СГЕНЕРИРОВАННЫЙ_ТЕКСТ[i] = ИНДЕКС_СГЕНЕРИРОВАННОГО_СЛОВА
    ВХОДНОЕ_СЛОВО = ИНДЕКС_СГЕНЕРИРОВАННОГО_СЛОВА
    i = i + 1
```

Рис. 5. Псевдокод этапа генерации текста  
Fig. 5. Pseudocode of the text generation stage

Здесь переменные СКРЫТОЕ\_СОСТОЯНИЕ и СОСТОЯНИЕ\_ЯЧЕЙКИ представляют собой состояние нейронной сети LSTM. СГЕНЕРИРОВАННЫЙ\_ТЕКСТ представляет собой массив сгенерированных индексов слов из словаря. ИНДЕКС\_СТАРТОВОГО\_СЛОВА является индексом слова `_START_`. ВХОДНОЕ\_СЛОВО представляет собой вход на нейронную сеть LSTM на текущей итерации. ВЫХОД\_НС содержит выход нейронной сети генератора на итерации  $i$  до выбора случайного индекса слова из словаря. Переменная ВЫХОД\_НС\_EXP является выходом нейронной сети генератора после вычисления экспоненты. ИНДЕКС\_СГЕНЕРИРОВАННОГО\_СЛОВА содержит результат операции `multinomial` на основе переменной ВЫХОД\_НС\_EXP, является сгенерированным словом. Функция ВЫЧИСЛЕНИЕ\_LSTM проводит одну итерацию вычислений на основе нейронной сети LSTM, в результате чего вычисляются новое скрытое состояние и один выходной вектор вероятностей.

Вычисление функции ошибки нейронной сети генератора представлено в виде псевдокода на рис. 6.

Переменная ОШИБКА является минимизируемой ошибкой нейронной сети генератора. Функция ОТВЕТ\_ДИСКРИМИНАТОРА проводит оценку качества сгенерированного текста. ОЦЕНКА\_ДИСКРИМИНАТОРА является выводом нейронной сети дискриминатора на основе переменной СГЕНЕРИРОВАННЫЙ\_ТЕКСТ.

```
# оценка текста дискриминатором
ОЦЕНКА_ДИСКРИМИНАТОРА = ОТВЕТ_ДИСКРИМИНАТОРА(СГЕНЕРИРОВАННЫЙ_ТЕКСТ)

# вычисление ошибки генератора
СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ = ИНИЦИАЛИЗАЦИЯ_СОСТОЯНИЯ_LSTM()
ВХОДНОЕ_СЛОВО = ИНДЕКС_СТАРТОВОГО_СЛОВА
ОШИБКА = 0 # инициализация ошибки
i = 0
ПОКА i < ДЛИНА_ТЕКСТА: # проход по всем словам сгенерированного текста
    ВЫВОД_НС, СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ = ВЫЧИСЛЕНИЕ_LSTM(ВХОДНОЕ_СЛОВО,
        СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ) # вычислить выход нейронной сети на итерации i
    ОШИБКА = ОШИБКА - ВЫВОД_НС[СГЕНЕРИРОВАННЫЙ_ТЕКСТ[i]] * ОЦЕНКА_ДИСКРИМИНАТОРА # вычислить ошибку для слова на итерации i
    ВХОДНОЕ_СЛОВО = СГЕНЕРИРОВАННЫЙ_ТЕКСТ[i]
    i = i + 1
```

Рис. 6. Псевдокод вычисления ошибок  
Fig. 6. Error computation pseudocode

В [1] предлагается использование метода Монте-Карло [26] для уточнения градиента. В данной работе не реализовывался алгоритм Монте-Карло, SeqGAN может применяться без данного алгоритма. В работе предпринята попытка увеличения точности вычисления градиента на основе использования пакетного обучения с увеличенным до 4 000 примеров пакетом данных.

### 7. Результаты тестирования SeqGAN

Нейронная сеть SeqGAN была реализована и протестирована на задаче пословной генерации коротких текстов [16].

В качестве нейронной сети генератора в SeqGAN выступает LSTM, обученная с помощью MLE. Результаты обучения и тестирования указанной LSTM описаны выше. Далее будут приведены результаты дообучения LSTM с помощью нейронной сети SeqGAN.

Были проведены реализация [16] и тестирование нейронной сети SeqGAN на выборках данных: стихи с сайта Stihi.ru [7] и подписи к изображениям из выборки COCO Image Captions [8]. Тестирование проводилось на тестовых выборках, описанных ранее. Для тестирования использовалось 500 сгенерированных примеров. Оценка качества проводилась на основе метрики BLEU [9]. Результаты тестирования представлены в табл. 4:

Таблица 4

Результаты тестирования SeqGAN

Выборка	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	0,798	0,49	0,259	0,153
COCO Image Captions	0,545	0,430	0,308	0,206

Примеры текстов, сгенерированных с помощью SeqGAN, представлены в табл. 5. Для каждого примера приведена оценка по метрике BLEU [9].

Таблица 5

Примеры сгенерированного текста

Выборка	Тип примеров	№	Пример	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	Сгенерированные с помощью SeqGAN	1	кому то на меня я не знаю что будет	1,0	0,855	0,65	0,495
		2	и без тебя не могу без тебя кричу	0,943	0,822	0,631	0,305
		3	на небесах не так уж не просто как прежде	0,943	0,606	0,237	0,14
COCO Image Captions	Сгенерированные с помощью SeqGAN	1	A group of people standing next to a body of water	0,589	0,577	0,565	0,537
		2	A man sitting on a bench holding a Frisbee as others watch	0,585	0,485	0,428	0,384
		3	A man and woman are playing a video game	0,429	0,313	0,206	0,103

По приведенным в табл. 1, 4 результатам можно сделать вывод, что нейронная сеть SeqGAN [1] позволяет улучшить качество генерации текста по метрике BLEU [9].

### 8. Возведение в степень значений вектора вероятностей

В данной работе предпринята попытка улучшить качество генерации текста на основе возведения значений вектора вероятностей в степень  $k$ , большую 1.

Данная операция заключается в возведении выхода нейронной сети, после вычисления экспоненты, в некоторую степень  $k$ , большую 1, что позволяет увеличить качество генерации текста за счет снижения его разнообразия. Рост качества генерации связан с повышением вероятности выбора более подходящего слова, по мнению нейронной сети. Операцию можно представить в виде псевдокода (рис. 7). Здесь  $k$  – выбранная степень, большая 1.

```
# генерация текста
СГЕНЕРИРОВАННЫЙ_ТЕКСТ = ИНИЦИАЛИЗАЦИЯ_МАССИВА(ДЛИНА_ТЕКСТА) # инициализация массива для записи текста
СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ = ИНИЦИАЛИЗАЦИЯ_СОСТОЯНИЯ_LSTM()
ВХОДНОЕ_СЛОВО = ИНДЕКС_СТАРТОВОГО_СЛОВА
i = 0
ПОКА i < ДЛИНА_ТЕКСТА:
    ВЫХОД_НС, СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ = ВЫЧИСЛЕНИЕ_LSTM(ВХОДНОЕ_СЛОВО,
        СКРЫТОЕ_СОСТОЯНИЕ, СОСТОЯНИЕ_ЯЧЕЙКИ) # вычислить выход нейронной сети на итерации i
    ВЫХОД_НС_EXP = EXP(ВЫХОД_НС)**k
    ИНДЕКС_СГЕНЕРИРОВАННОГО_СЛОВА = MULTINOMIAL(ВЫХОД_НС_EXP)
    СГЕНЕРИРОВАННЫЙ_ТЕКСТ[i] = ИНДЕКС_СГЕНЕРИРОВАННОГО_СЛОВА
    ВХОДНОЕ_СЛОВО = ИНДЕКС_СГЕНЕРИРОВАННОГО_СЛОВА
    i = i + 1
```

Рис. 7. Псевдокод этапа генерации текста на основе возведения вектора вероятностей в степень  $k$

Fig. 7. Pseudocode of the text generation stage based on raising the probability vector to the power  $k$

Плюсом данного подхода является простота его реализации, которая не требует дополнительного обучения нейронной сети. Существенным минусом является снижение разнообразия генерируемых текстов.

Данный подход можно применить к нейронной сети, обученной как на основе MLE, так и на основе SeqGAN. Нами было проведено тестирование данного подхода с использованием нейронной сети, обученной на основе SeqGAN.

Тестирование проводилось на тестовых выборках, описанных ранее. Для тестирования использовалось 500 сгенерированных примеров. Оценка качества проводилась на основе метрики BLEU [9].

Результаты тестирования SeqGAN с возведением значений вектора вероятностей в степень 1,5 представлены в табл. 6.

Таблица 6

Результаты тестирования SeqGAN с возведением значений вектора вероятностей в степень 1,5

Выборка	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	0,936	0,761	0,483	0,278
COCO Image Captions	0,601	0,548	0,466	0,362

Результаты тестирования SeqGAN с возведением значений вектора вероятностей в степень 2 представлены в табл. 7. В табл. 8 приведены примеры сгенерированного теста.

Таблица 7

Результаты тестирования SeqGAN с возведением значений вектора вероятностей в степень 2

Выборка	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	0,968	0,86	0,618	0,383
COCO Image Captions	0,602	0,571	0,511	0,423

По полученным результатам можно сделать вывод, что возведение значений вектора вероятностей в степень, большую 1, перед использованием данного вектора в операции multinomial [20] позволяет увеличить качество генерации текстов.

Примеры сгенерированного текста

Выборка	Тип примеров	№	Пример	BLEU-2	BLEU-3	BLEU-4	BLEU-5
Stihi.ru	Сгенерированные	1	и не важно что ты не со мной	1,0	1,0	0,962	0,844
		2	и в то что я не могу понять	1,0	0,909	0,753	0,351
		3	и не забыть мне все что я не знаю	0,943	0,822	0,531	0,265
COCO Image Captions	с помощью SeqGAN и возведением вектора вероятностей в степень 2	1	A man is sitting next to a fire hydrant on the side of the road	0,795	0,789	0,767	0,721
		2	A man holding a smart phone in his hand and a man	0,692	0,683	0,641	0,555
		3	A young man sitting on a couch next to a little girl	0,613	0,551	0,445	0,345

Следует отметить, что качество генерации текстов значительно превышает качество реальных текстов по метрике BLEU. Существенное увеличение качества генерации текстов по метрике BLEU связано со снижением разнообразия текстов, в результате чего нейронная сеть чаще использует популярные слова и словосочетания. Тексты из обучающей выборки являются более разнообразными, чем тексты, генерируемые нейронной сетью, что могло привести к более низкой оценке примеров из обучающей выборки по метрике BLEU.

### Заключение

В данной работе были рассмотрены и реализованы подходы MLE [6] и SeqGAN [1] для генерации коротких текстов на основе нейронной сети LSTM [4]. В реализации SeqGAN метод Монте-Карло был заменен пакетным обучением с увеличенным пакетом данных. Программная реализация представлена на сайте [16].

Рассматриваемые подходы протестированы на следующих выборках данных: сборник русских стихов с сайта Stihi.ru [7] и подписи к изображениям из выборки COCO Image Captions [8]. На основе полученных результатов сделан вывод, что алгоритмы MLE и SeqGAN способны генерировать короткие тексты. Подход на основе SeqGAN позволяет улучшить качество генерации текста по сравнению с подходом на основе MLE, но в процессе обучения уменьшается разнообразие генерируемых текстов. Нейронная сеть LSTM, обученная на основе SeqGAN, генерирует примеры, близкие по качеству к реальным примерам, по метрике BLEU (см. табл. 1, 2, 4–7).

Предложен, реализован и протестирован подход на основе возведения значений вектора вероятностей в степень, большую 1. Данная операция позволила увеличить качество генерируемого текста по метрике BLEU, но снизила его разнообразие. Нейронная сеть чаще использует некоторый небольшой набор слов и словосочетаний. Для повышения разнообразия текстов данный подход требует модификации.

В результате проделанной работы следует отметить, что требуется качественное предварительное обучение генератора и дискриминатора SeqGAN, что требует дополнительных затрат времени на обучение и реализацию алгоритмов.

Был сделан вывод, что для обучения генератора на основе нейронной сети SeqGAN требуется качественно обученный дискриминатор. Дискриминатор может не справиться с классификацией сгенерированных и реальных примеров при достаточно качественно обученном генераторе. Требуется настройка гиперпараметров нейронной сети дискриминатора, таких как оптимизатор, скорость обучения, топология нейронной сети.

В рамках данной работы сделан вывод, что оценка качества генерации текстов по метрике BLEU имеет свои недостатки. Низкая оценка качества примера по метрике BLEU может быть дана качественному тексту, написанному человеком или сгенерированному алгоритмом. Например, если оцениваемый пример оказался оригинальным и не имеет совпадений с тестовой выборкой. Таким образом, более высокая оценка качества генерации текстов по метрике BLEU, чем у примеров, написанных людьми, может быть вызвана генерацией более популярных текстов, которые чаще встречаются

в тестовой выборке. Для улучшения качества оценки по метрике BLEU требуется увеличение количества примеров в тестовой выборке, что приводит к возрастанию времени тестирования алгоритмов.

#### ЛИТЕРАТУРА

1. Yu L., Zhang W., Wang J., Yu Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient // AAAI'17: Proc. of the Thirty-First AAAI Conference on Artificial Intelligence. 2017. P. 2852–2858.
2. Guo J., Lu S., Cai H., Zhang W., Yu Y., Wang J. Long Text Generation via Adversarial Training with Leaked Information // The Thirty-Two AAAI Conference on Artificial Intelligence, 2018. V. 32, N. 1. P. 5141–5148.
3. Lin K., Li D., He X., Zhang Z., Sun M.-T. Adversarial Ranking for Language Generation // Advances in Neural Information Processing Systems. 2018. P. 3155–3165.
4. Hochreiter S., Schmidhuber J. Long short-term memory // Neural Computation. 1997. V. 9, is. 8. P. 1735–1780.
5. Знакомство с архитектурой LSTM-сетей. URL: <http://datareview.info/article/znakomstvo-s-arhitekturoy-lstm-setey/> (accessed: 20.10.2020).
6. Cramer J.S. Econometric Applications of Maximum Likelihood Methods. Cambridge University Press, 1986. 206 p.
7. StihData. URL: <https://github.com/DenisVorotyntsev/StihData/tree/194107ff98249fd11e8da5c3ee2d> (accessed: 20.10.2020).
8. Chen X., Fang H., Lin T.-Y., Vedantam R., Gupta S., Dollar P., Zitnick, C.L. Microsoft COCO Captions: Data Collection and Evaluation Server. 2015. URL: <https://arxiv.org/pdf/1504.00325.pdf> (accessed: 20.10.2020).
9. Papineni K., Roukos S., Ward T., Zhu W.-J. BLEU: a Method for Automatic Evaluation of Machine Translation // Proc. of the 40th Annual Meeting of the Association for Computational Linguistics. 2002. P. 311–318.
10. Coughlin D. Correlating Automated and Human Assessments of Machine Translation Quality // Proc. of MT Summit IX. 2003. P. 63–70.
11. Chen B., Cherry C. A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU // Proc. of the Ninth Workshop on Statistical Machine Translation. 2014. P. 362–367.
12. Wu Y., Wang, J. Text Generation Service Model Based on Truth-Guided SeqGAN // IEEE Access. 2020. V. 8. P. 11880–11886.
13. Shi Z., Chen X., Qiu X., Huang X. Toward Diverse Text Generation with Inverse Reinforcement Learning. 2018. URL: <https://arxiv.org/pdf/1804.11258.pdf> (accessed: 20.10.2020).
14. Li J., Monroe W., Shi T., Jean S., Ritter A., Jurafsky D. Adversarial Learning for Neural Dialogue Generation // EMNLP. 2017. P. 2147–2159.
15. Выборка данных Image COCO. URL: <https://github.com/CR-Gjx/LeakGAN/tree/master/Image%20COCO/save> (accessed: 20.10.2020).
16. Программная реализация SeqGAN. URL: <https://github.com/NikolayKrivosheev/Generation-of-short-texts-SeqGAN> (accessed: 20.10.2020).
17. SeqGAN. URL: <https://github.com/suragnair/seqGAN> (accessed: 20.10.2020).
18. What the heck is Word Embedding. URL: <https://towardsdatascience.com/what-the-heck-is-word-embedding-b30f67f01c81> (accessed: 20.10.2020).
19. Xu H. Text Classification with Topic-based Word Embedding and Convolutional Neural Networks // Proc. of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. 2016. P. 88–97.
20. Torch.multinomial. URL: <https://pytorch.org/docs/master/generated/torch.multinomial.html> (accessed: 20.10.2020).
21. Ruder S. An overview of gradient descent optimization algorithms. 2017. URL: <https://arxiv.org/pdf/1609.04747.pdf> (accessed: 20.10.2020).
22. Николенко С., Кагурин А., Архангельская Е. Глубокое обучение Погружение в мир нейронных сетей. Спб. : Питер. 2018. 477 с.
23. Goodfellow I. Generative Adversarial Nets // Advances in Neural Information Processing Systems. 2014. V. 27. P. 2672–2680.
24. Sutton R.S. Policy Gradient Methods for Reinforcement Learning with Function Approximation // Advances in Neural Information Processing Systems. 1999. V. 12. P. 1057–1063.
25. Kulkarni T.D. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation // Advances in Neural Information Processing Systems. V. 29. 2016. P. 3675–3683.
26. Metropolis N., Ulam S. The Monte Carlo Method // Journal of the American Statistical Association. 1949. V. 44, is. 247. P. 335–341.

Поступила в редакцию 20 апреля 2021 г.

Krivosheev N.A., Ivanova Y.A., Spitsyn V.G. (2021) AUTOMATIC GENERATION OF SHORT TEXTS BASED ON THE USE OF NEURAL NETWORKS LSTM AND SeqGAN. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 57. pp. 118–130

DOI: 10.17223/19988605/57/13

The aim of this work is to estimate the quality of automatic short texts generation based on a network with long short-term memory (Long Short-Term Memory, LSTM). To train the LSTM neural network, supervised learning based on the maximum likeli-

hood estimation (MLE) method is used. Further LSTM training is used as part of an adversarial network that generates a sequence (Sequence Generative Adversarial Net, SeqGAN). It should be noted that the Monte Carlo method is not used in this work, batch training with a larger data packet is applied instead.

The paper proposes a modification of the output vector by raising its values to a power greater than 1 to increase the probability of choosing the generated word with the greatest weight in the output vector of the neural network. This operation makes it possible to increase the quality of the generated text, but reduces its variety.

The length of the generated texts is 10 and 20 words. The following data samples are used for training and testing neural networks: a collection of Russian poems from the Stihi.ru website and captions to images in English from the COCO Image Captions sample. Word-by-word text generation is applied. The quality of text generation is assessed using the BLEU metric. The analysis and comparison with similar solutions based on the COCO Image Captions data sample are carried out.

Training and testing of MLE and SeqGAN-based approaches was carried out. Based on the presented results, we can conclude that training based on the SeqGAN neural network, in comparison with the MLE-based approach, improves the quality of text generation according to the BLEU metric. The texts generated on the basis of the SeqGAN neural network are comparable in quality to the examples from the training set using the BLEU metric.

The approach based on raising the values of the probability vector to a power makes it possible to increase the quality of text generation according to the BLEU metric, but leads to a reduction in texts variety.

It should be noted that the quality of text generation, based on SeqGAN and modifying the output vector by raising its values to a power greater than 1, significantly exceeds the quality of real texts according to the BLEU metric. A significant increase in the quality of text generation according to the BLEU metric is associated with a reduction the variety of texts, as a result of which the neural network uses popular words and phrases more often. The texts from the training set are more diverse than the texts generated by the neural network, which could lead to a lower score of training set examples according to the BLEU metric.

Keywords: SeqGAN; text generation; adversarial reinforcement learning.

*KRIVOSHEEV Nikolay Anatolyevich* (Post-graduate student of National Research Tomsk Polytechnic University, Tomsk, Russian Federation).

E-mail: nikola0212@mail.ru

*IVANOVA Yulia Aleksandrovna* (Candidate of Technical Sciences, Associate Professor of the Department of Information Technologies of National Research Tomsk Polytechnic University, Tomsk, Russian Federation).

E-mail: jbolotova@tpu.ru

*SPITSYN Vladimir Grigorievich* (Doctor of Technical Sciences, Professor of the Department of Information Technologies of National Research Tomsk Polytechnic University, Tomsk, Russian Federation).

Email: spvg@tpu.ru

## REFERENCES

1. Yu, L., Zhang, W., Wang, J. & Yu, Y. (2017) SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. pp. 2852–2858.
2. Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y. & Wang, J. (2018) Long Text Generation via Adversarial Training with Leaked Information. *The Thirty-Two AAAI Conference on Artificial Intelligence*. 32(1). pp. 5141–5148.
3. Lin, K., Li, D., He, X., Zhang, Z. & Sun, M.-T. (2018) Adversarial Ranking for Language Generation. *Advances in Neural Information Processing Systems*. pp. 3155–3165.
4. Hochreiter, S. & Schmidhuber, J. (1997) Long short-term memory. *Neural Computation*. 9(8). pp. 1735–1780.
5. Datareview.info. (s.n.) *Znakomstvo s arkhitekturoy LSTM-setey* [The architecture of LSTM networks]. [Online] Available from: <http://datareview.info/article/znakomstvo-s-arkhitekturoy-lstm-setey/> (Accessed: 20th October 2020).
6. Cramer, J.S. (1986) *Econometric Applications of Maximum Likelihood Methods*. Cambridge University Press.
7. Github.com. (n.d.) *StihiData*. [Online] Available from: <https://github.com/DenisVorotyntsev/StihiData/tree/194107ff98249fd11e8da5c3ee2d> (Accessed: 20th October 2020).
8. Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollar, P. & Zitnick, C.L. (2015) *Microsoft COCO Captions: Data Collection and Evaluation Server*. [Online] Available from: <https://arxiv.org/pdf/1504.00325.pdf> (Accessed: 20th October 2020).
9. Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. (2002) BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. pp. 311–318.
10. Coughlin, D. (2003) Correlating Automated and Human Assessments of Machine Translation Quality. *Proceedings of MT Summit IX*. pp. 63–70.
11. Chen, B. & Cherry, C. (2014) A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU. *Proceedings of the Ninth Workshop on Statistical Machine Translation*. pp. 362–367.
12. Wu, Y. & Wang, J. (2020) Text Generation Service Model Based on Truth-Guided SeqGAN. *IEEE Access*. 8. pp. 11880–11886.
13. Shi, Z., Chen, X., Qiu X. & Huang, X. (2018) *Toward Diverse Text Generation with Inverse Reinforcement Learning*. [Online] Available from: <https://arxiv.org/pdf/1804.11258.pdf> (Accessed: 20th October 2020).

14. Li, J., Monroe, W., Shi, T., Jean, S., Ritter A. & Jurafsky, D. (2017) Adversarial Learning for Neural Dialogue Generation. *EMNLP*. pp. 2147–2159.
15. Github.com. (n.d.) *Vyborka dannykh Image COCO* [COCO Image Sampling]. [Online] Available from: <https://github.com/CR-Gjx/LeakGAN/tree/master/Image%20COCO/save> (Accessed: 20th October 2020).
16. Github.com. (n.d.) *Programmnyaya realizatsiya SeqGAN* [SeqGAN neural network implementation]. [Online] Available from: <https://github.com/NikolayKrivoshchev/Generation-of-short-texts-SeqGAN> (Accessed: 10th December 2020).
17. Github.com. (n.d.) *SeqGAN*. [Online] Available from: <https://github.com/suragnair/seqGAN> (Accessed: 20th October 2020).
18. Agrawal, S. (2019) *What the heck is Word Embedding*. [Online] Available from: <https://towardsdatascience.com/what-the-heck-is-word-embedding-b30f67f01c81> (Accessed: 20th October 2020).
19. Xu, H. (2016) Text Classification with Topic-based Word Embedding and Convolutional Neural Networks. *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. pp. 88–97.
20. Pytorch.org. (n.d.) *Torch.multinomial*. [Online] Available from: <https://pytorch.org/docs/master/generated/torch.multinomial.html> (Accessed: 20th October 2020).
21. Ruder, S. (2017) *An overview of gradient descent optimization algorithms*. [Online] Available from: <https://arxiv.org/pdf/1609.04747.pdf> (Accessed: 20th October 2020).
22. Nikolenko, S., Kadurin, A. & Arkhangelskaya, E. (2018) *Glubokoe obuchenie Pogruzhenie v mir neyronnykh setey* [Deep learning. Immersion in the world of neural net-works]. St. Petersburg: Piter.
23. Goodfellow, I. (2014) Generative Adversarial Nets. *Advances in Neural Information Processing Systems*. 27. pp. 2672–2680.
24. Sutton, R.S. (1999) Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems*. 12. pp. 1057–1063.
25. Kulkarni, T.D. (2016) Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *Advances in Neural Information Processing Systems*. 29. pp. 3675–3683.
26. Metropolis, N. & Ulam, S. (1949) The Monte Carlo Method. *Journal of the American Statistical Association*. 44(247). pp. 335–341.