Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»

На правах рукописи

Грачев Артем Михайлович

Методы сжатия рекуррентных нейронных сетей для задач обработки естественного языка

РЕЗЮМЕ
диссертации на соискание учёной степени
кандидата компьютерных наук НИУ ВШЭ

Москва — 2019

Диссертационная работа выполнена в Национальном исследовательском институте «Высшая Школа Экономики».

Научный руководитель: **Игнатов Дмитрий Игоревич**, кандидат технических наук, доцент, Национальный исследовательский университет «Высшая Школа Экономики».

Общая характеристика работы

Актуальность темы.

В последнее десятилетие быстро развивается направление в машинном обучении, называемое глубоким обучением (deep learning) [1; 2], связанное с успешным обучением нейронных сетей и использованием сложных и **глубоких** архитектур. Подходы, связанные с глубоким обучением, вывели сразу несколько направлений в компьютерных науках на новый уровень. В первую очередь эти направления связаны со сложно формализуемыми задачами, такими как обработка изображений, понимание текста, распознавание речи. Во многих областях и конкретных задачах подходы, основанные на методах глубинного обучения, стали признанным индустриальным решением [3]. При этом появляются всё новые задачи [2; 4], а потенциал этих подходов далеко не исчерпан.

Успех нейронных сетей объясняется несколькими факторами. Во-первых, это теоретическая возможность аппроксимировать произвольные функции с помощью нейронных сетей [5; 6]. Во-вторых, это простота, с которой масштабируется процесс обучения. Алгоритм обучения сети с помощью обратного распространения ошибки позволяет легко считать градиент функции потерь по параметрам модели и делать это параллельно. В-третьих, рост вычислительных мощностей. Масштабируемость процессов обучения ничего бы не стоила, если бы не возможность масштабировать все вычисления. Появление высокомощных графических карт позволяет обучать нейронные сети дёшево и быстро и практически любому специалисту.

При этом, у нейронных сетей есть и недостатки. Например, несмотря на то, что с теоретической точки зрения нейронные сети являются универсальным аппроксиматором, на текущий момент не существует конструктивного способа построения нейронных сетей для каких-то конкретных задач. То есть нет методов, которые заранее позволяют сказать нейронная сеть какой топологии (сколько слоёв, какого размера каждый слой) потребуется для решения той или иной задачи. На практике специалистам приходится подбирать гиперпараметры, отвечающие за топологию сети, в процессе обучения.

Другой важный недостаток становится очевидным, когда появляется потребность использовать нейронные сети на каких-либо устройствах: на мобильных телефонах, телевизорах, пылесосах или даже холодильниках. Это могут быть задачи обработки звуковых сигналов, распознавания объектов,

моделирование подсказок в мобильной клавиатуре или электронной почте. Обычно такие устройства имеют не очень мощные процессоры и не очень большое количество памяти. В то же время нейронные сети требовательны и к тому, и к другому.

Этот второй недостаток делает актуальным задачи сжатия нейронных сетей с целью их последующего размещения на различных устройствах[1][2][3].

В данном диссертационном исследовании мы рассматриваем проблему сжатия нейронных сетей в контексте класса задач, связанного с моделированием естественного языка, и работаем, в основном, с рекуррентными нейронными сетями. Это позволяет нам выделить некоторые характерные особенности этого класса задач и адаптировать наши методы именно для него. При этом большая часть описываемых методов применимы и в более широком контексте, то есть для произвольных нейронных сетей.

Базовая постановка задачи моделирования языка заключается в предсказании следующего слова по предыдущей цепочке слов или, другими словами, по левому контексту. Самые первые методы для решения этой задачи были основаны на прямом сохранении всех возможных вариантов. В момент предсказания выбирался наиболее вероятный вариант или делалось сэмплирование из сохраненного распределения. У такого подхода есть проблемы, связанные с учетом дальних зависимостей и необходимостью хранить все варианты. Так, например, все возможные цепочки длины 4 для некоторых словарей уже невозможно хранить в памяти компьютеров.

Рекуррентные нейронные сети отчасти способны моделировать дальние зависимости и не требуют прямого запоминания всех вариантов. Но они всё ещё слишком большого размера для их повсеместного использования на устройствах, которые, как уже упоминалось выше, обычно очень требовательны к памяти и быстродействию. Особенностью построения языковых моделей является необходимость хранить словарь из нескольких тысяч слов (например, около 20 тысяч слов покрывают лишь 80% русского языка). Кроме того, часто необходимо решать задачу прогнозирования следующего слова или, выражаясь терминами машинного обучения, решать задачу классификации на несколько

[1] https://os.mbed.com/blog/entry/uTensor-and-Tensor-Flow-Announcement/

[2] https://towardsdatascience.com/why-machine-learning-on-the-edge-92fac32105e6

[3] https://petewarden.com/2017/05/08/running-tensorflow-graphs-on-microcontrollers/

тысяч (или даже десятков тысяч) классов. Поэтому, несмотря на то, что нейронные сети занимают гораздо меньше места, чем простые статистические модели, задача сжатия является актуальной для их широкого применения.

Современные методы сжатия включают в себя как простые методы: например, прореживание матриц или квантизация представления чисел, так и более сложные, основанные на разложениях матриц или применении байесовских методов. Среди работ, основанных на прореживании матриц и квантизации, можно отметить [7; 8]. Методы, основанные на разложениях матриц, делятся на несколько категорий. Во-первых, можно раскладывать матрицы в низкоранговые разложения [9], тем самым достигая некоторого сжатия. Во-вторых, можно использовать матрицы специального типа, например URNN [10] или матрицы Теплица [9]. Наконец, можно использовать разложения слоёв в Tensor Train [11]. Вариационный дропаут первоначально использовали для настройки индивидуальной степени дропаута для каждого нейрона [12]. В [13] авторы показали, что вариационный дропаут также может занулять некоторые нейроны, тем самым давая сжатие. Если накладывать априорное распределние на отдельные веса, то мы получаем разреженные матрицы, что, как и в случае с обычным разреживанием, не очень удобно, поэтому в последующей работе предлагается делать структурное сжатие, накладывая априорное распределение сразу на столбцы и строчки [14]. В данном диссертационном исследовании мы детально проанализировали текущие методы сжатия нейронных сетей и предложили ряд новых алгоритмов, решающих эту задачу более эффективно.

Целью данного диссертационного исследования является создание новых алгоритмов сжатия нейронных сетей для применения их в задаче обработки естественного языка. Предлагаемые алгоритмы должны решать задачу сжатия более эффективно с точки зрения использования ресурсов клиентских устройств, например, мобильных телефонов.

Для достижения цели в рамках работы предполагается исследовать различные методы сжатия нейронных сетей и предложить на их основе нейросетевые архитектуры, имеющие сравнительно малый размер и эффективно решающие прикладные задачи. В качестве основных **задач** исследования можно выделить:

1. Эмпирический анализ алгоритмов сжатия нейронных сетей, основанных на техниках разреживания и квантизации, разложении матриц, байесовских методах.
2. Построение новых и модфикация существующих алгоритмов сжатия нейронных сетей с учётом специфики задачи обработки естественного языка и типичных архитектур рекуррентных нейронных сетей.
3. Адаптация разработанных алгоритмов, отвечающих заданным характеристикам, для применения их на различных устройствах, например, мобильных телефонах.

Сформулируем **основные результаты исследования и положения, выносимые на защиту**:

1. Построение алгоритмов сжатия скрытых слоев нейронных сетей для задачи моделирования языка с помощью методов матричных разложений.
2. Эффективное решение задачи классификации для большого числа классов (10–30 тыс.) с использованием методов матричных разложений для входного и выходного слоев нейронной сети.
3. Адаптация байесовских методов разреживания для рекуррентных нейронных сетей в задаче моделирования языка.
4. Портирование эффективной реализации рекуррентных нейронных сетей на мобильные устройства и лабораторное тестирование.

Научная новизна:

1. Впервые было применено TT-разложение для сжатия рекуррентных нейронных сетей в задаче моделирования языка. Также было подробно исследовано применение низкорангового матричного разложения для данной задачи и класса моделей на основе рекуррентных нейросетей.
2. Впервые был применён алгоритм автоматического определения значимых признаков с помощью двойного стохастического вариационного вывода (DSVI-ARD) в задаче моделирования языка.
3. Было выполнено портирование сжатых моделей на мобильные устройства. На тот момент (согласно публикациям на ведущих конференциях) это была первая (задокументированная в [15]) реализация нейронных сетей (тем более сжатых) на мобильном GPU.

4. Была предложена общая схема сжатия для рекуррентных нейронных сетей в задаче моделирования языка.

Практическая значимость работы заключается в разработке новых методов сжатия рекуррентных нейронных сетей с учётом особенностей, возникающих в задаче моделирования языка, и переноса этих моделей на устройства с мобильными GPU.

Достоверность полученных результатов обеспечивается математически корректными моделями сжатия, а также большим количеством экспериментов с различными архитектурами и наборами данных. Результаты сопоставимы с работами других авторов.

Публикации и апробация работы.

Основные результаты по теме диссертации изложены в печатных изданиях [15–18].

Публикации повышенного уровня.

- Artem M. Grachev, Dmitry I. Ignatov and Andrey V. Savchenko. Neural Networks Compression for Language Modeling. — *Pattern Recognition and Machine Intelligence - 7th International Conference, PReMI 2017, Kolkata, India, December5-8, 2017, Proceedings. – 2017. – Pp. 351–35.*
- Elena Andreeva, Dmitry I. Ignatov, Artem M. Grachev and Andrey V. Savchenko. Extraction of Visual Features for Recommendation of Products via Deep Learning. – *Analysis of Images, Social Networks and Texts – 7th International Conference, AIST 2018, Moscow, Russia, July 5-7, 2018, Revised Selected Papers. – 2018. – Pp. 201–210.*
- Artem M. Grachev, Dmitry I. Ignatov and Andrey V. Savchenko. Compression of Recurrent Neural Networks for Efficient Language Modeling. — *Applied Soft Computing – 2019. – Vol. 79. – Pp. 354 – 362.*

Прочие публикации.

- Maxim Kodryan*, Artem Grachev*, Dmitry Ignatov and Dmitry Vetrov. Efficient Language Modeling with Automatic Relevance Determination in Recurrent Neural Networks — *ACL 2019, Proceedings of the 4th Workshop on Representation Learning for NLP, August 2, 2019, Florence, Italy*

Работы [15–17] цитируются в системе научного цитирования **Scopus**. Статья [17] в журнале Applied Soft Computing [17] входит в Q1 **Scopus** и **Web**

of Science. Работа [18] опубликована на воркшопе при конференции CORE A* и входит в **ACL Anthology**.

Апробация работы.

Результаты диссертационного исследования докладывались на:

- 7th International Conference on Pattern Recognition and Machine Intelligence (06.12.2017, Калькутта, Индия). Доклад «Neural network compression for language modeling».
- Аспирантский семинар факультета компьютерных наук НИУ ВШЭ (2017, Москва, Россия). Доклад «Neural network compression for language modeling» («Сжатие нейронных сетей для задачи языкового моделирования»)
- Конференция «Технологии машинного обучения» (октябрь 2018, Москва, Россия). Доклад «Neural Networks for mobile devices» («Нейронные сети для мобильных устройств»)
- Семинар «Автоматическая обработка и анализ текстов» НИУ ВШЭ (20.04.2019, Москва, Россия). Доклад «Методы сжатия рекуррентных нейронных сетей для задач обработки естественного языка».
- Семинар в компании Samsung R&D Russia (2017, Москва, Россия). Доклад «Методы сжатия рекуррентных нейронных сетей для задач обработки естественного языка».
- Семинар в компании Samsung R&D Russia (2019, Москва, Россия). Доклад «DSVI-ARD сжатие в нейронных сетях».

Личный вклад. Все результаты получены автором лично, кроме результатов, полученных совместно с Е. Андреевой в статье [16] и М. Кодряном в статье [18].

Содержание работы

Объём и структура диссертации. Диссертация состоит из введения, четырёх глав и заключения. Полный объём диссертации составляет 105 страниц с 17 рисунками и 9 таблицами. Список литературы содержит 84 наименования.

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы,

сформулированы научная новизна и практическая значимость представляемой работы.

Первая глава является вводной. В ней рассматривается задача моделирования языка и её характерные особенности. В самом общем виде эта задача заключается в том, чтобы оценить вероятность встретить произвольную цепочку слов $(w^1, \ldots, w^T)$ в тексте.

$$\mathsf{P}\left(w^1, \ldots, w^T\right) = \mathsf{P}\left(w^1, \ldots, w^{T-1}\right) \mathsf{P}\left(w^T | w^1, \ldots, w_{T-1}\right) = \\ = \prod_{t=1}^{T} \mathsf{P}\left(w^t | w^1, \ldots, w^{t-1}\right) \quad (1)$$

Эта задача является актуальной как сама по себе, так и в контексте различных приложений. Так, например, такие модели напрямую используются в мобильной клавиатуре (см. рис. 1).

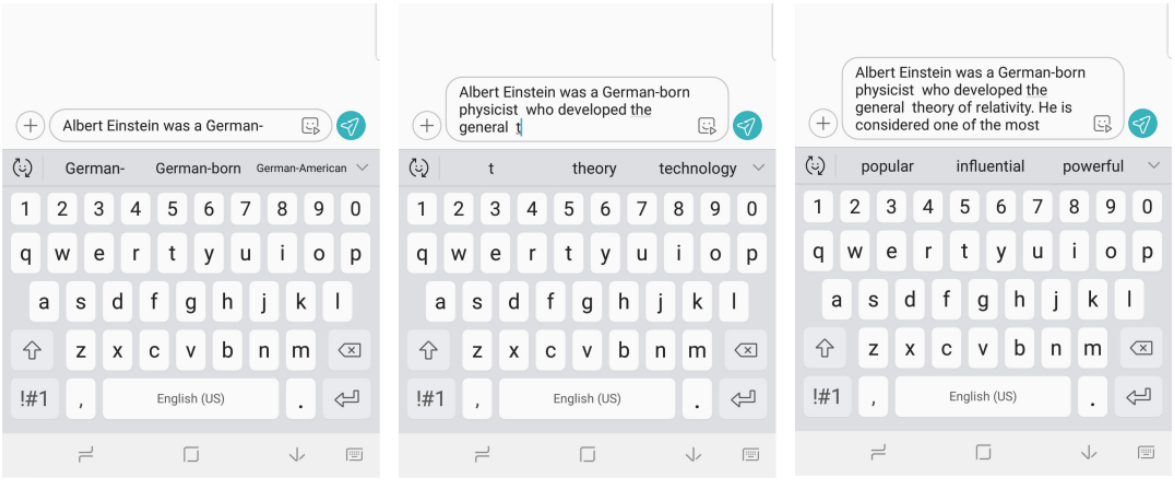


Рис. 1 — Пример индустриального применения задачи моделирования языка. Мобильная клавиатура.

Далее в этой главе обсуждаются различные способы решения этой задачи. Долгое время здесь широко использовались N-граммные модели [19; 20] и их различные вариации. Потом им на смену постепенно стали приходить рекуррентные нейронные сети [21; 22]. Ячейка обычной рекуррентной нейронной

сети может быть описана следующим образом:

$$z_\ell^t = W_\ell x_{\ell-1}^t + U_\ell x_\ell^{t-1} + b_l \tag{2}$$

$$x_\ell^t = \tanh(z_\ell^t), \tag{3}$$

где $t \in \{1, \ldots, N\}$ – это номер текущего шага по времени, $\ell \in \{1, \ldots, L\}$ – это номер текущего слоя, W_ℓ и U_ℓ — это матрицы весов для слоя ℓ и tanh — это гиперболический тангенс, который традиционно используется как функция активации в рекуррентных нейронных сетях. Вектор x_ℓ^t является выходом слоя ℓ в момент времени t. Мы подробно разбираем устройство таких сетей. Рассматриваем их различные вариации, такие как LSTM и GRU, а также обсуждаем алгоритм распространения ошибки через время [23–25], который используется для обучения таких сетей. Также обсуждается, как измеряется качество в этой задаче и различные приёмы, которые используются для улучшения сходимости при обучении.

Во **второй главе** мы рассматриваем рекуррентные нейронные сети с точки зрения того, сколько места они занимают на устройстве. Мы показываем, что все параметры нейронной сети можно разделить на два типа. Параметры, которые отвечают за входной и выходной слой и количество которых связано с размером словаря. И параметры слоёв сети. Более точно они выражаются следующей формулой:

$$n_{total} = 8Lk^2 + 2|\mathbb{V}|k, \tag{4}$$

где L – это число слоёв, k – это размер скрытого слоя, $|\mathbb{V}|$ – размер словаря, а 8 – это константа соответствующая числу матриц в скрытом слое (8 – это количество матриц для LSTM-слоя, в то время как в GRU их 6, а в RNN их 2).

Далее в этой главе приводится обзор основных подходов к сжатию нейронных сетей. Можно выделить два основных направления: методы, которые основаны на матричных разложениях, и методы, которые основаны на прореживании матриц. В заключении главы рассматриваются два базовых подхода к сжатию: прунинг и квантизация.

Прунинг это метод снижения числа параметров в нейронной сети путем удаления весов, которые приблизительно равны нулю. Можно зафиксировать

некоторый порог C и удалить все веса, модуль которых меньше этого порога: $|w_{ij}| < C$.

Квантизация – это метод для уменьшения размера нейронной сети в оперативной памяти. Суть метода заключается в том, что все множество весов сети разбивается на 256 интервалов. И каждому интервалу ставится в соответствие 8-битный integer. Таким образом, если изначально наше число хранилось в памяти как 32-битный float, то теперь оно хранится в 8-битном integer'е и занимает в 4 раза меньше места.

Эти два подхода имеют общие недостатки связанные с тем, что они не поддерживают обучений нейронной сети с нуля. Более того, для применения их на практике приходится затрачивать дополнительные усилия. Так, например, чтобы получить ускорение нейронной сети после применения прунинга, необходимо эффективно поддерживать разреженные матричные вычисления на аппаратном уровне. В случае квантизации мы получаем модель, сохраненную в 8-битном представлении, но для вычислений мы всё ещё должны будем использовать 32-битное представление, что означает, что мы должны использовать тот же объем памяти, что и в оригинальной модели.

Третья глава посвящена методам сжатия, которые связаны с матричными разложениями. Операции умножения матриц хорошо оптимизированы на современных процессорах [4] и скорость их умножения напрямую зависит от их размера. То есть, уменьшая размер матриц весов в нейронной сети, мы одновременно и снижаем количество параметров, и увеличиваем скорость вычислений. В данном исследовании для сокращения размеров матриц мы применяли низкоранговые матричные разложения и Tensor Train разложения [26]. Например, для обычной RNN такое низкоранговое разложение можно применить следующим образом:

$$x_l^t = \text{tanh}\left[W_l^a m_{l-1}^t + U_l^a m_l^{t-1} + b_l\right] \tag{5}$$

$$m_l^t = U_l^b x_l^t \tag{6}$$

$$\tag{7}$$

Здесь $U_\ell^a, W_\ell^a \in R^{k\times r}$ и $U_\ell^b \in R^{r\times k}$, где k – оригинальный размер скрытого слоя, а r — ранг разложения. Сокращение параметров достигается за счёт того, что

[4] Intel matrix multiplication

ранг r выбирается с условием, что $r \ll k$. На рис. 2 представлен иллюстративный пример, который показывает, в каких местах мы применяем разложение матриц. Мы подробно описываем как разложение такого типа можно применять к различным типам рекуррентных нейронных сетей, таких как LSTM, GRU.

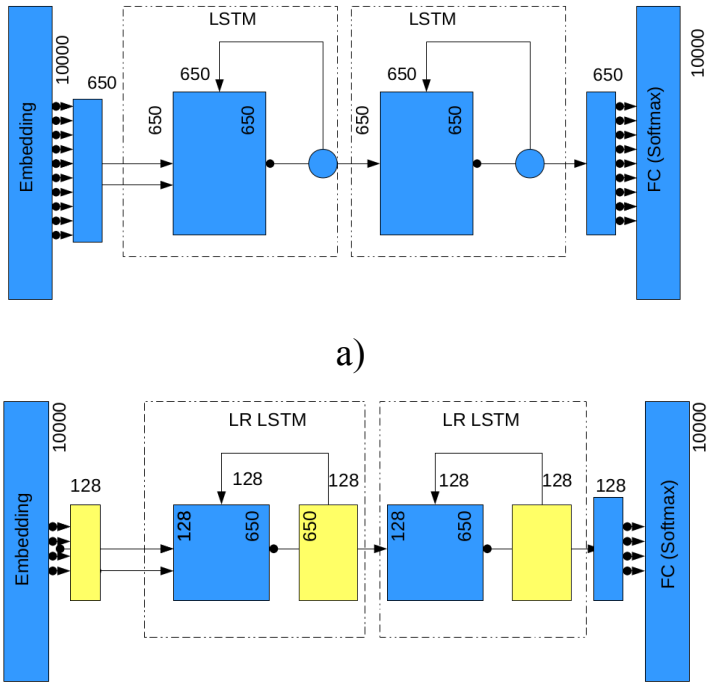
Другое разложение, которое мы здесь использовали, это разложение в Tensor Train. Оно является одним из возможных обобщений SVD на тензоры размерности больше двух. Использование TT-разложения мотивировано возможностью конвертировать матрицы в соответствующие им тензоры, после чего раскладывать и сжимать уже их. За счёт того, что в тензоре мы имеем сразу несколько размерностей, мы получаем потенциально больший размер сжатия и большую гибкость в сжатии.

Опишем, как это разложение можно применить для сжатия нейронных сетей. Для этого рассмотрим, например, матрицу весов $W \in \mathbb{R}^{k \times k}$ RNN слоя (2). Мы можем выбрать такие числа $k_1, ..., k_d$, чтобы $k_1 \times \ldots \times k_d = k \times k$ и выполнить преобразование матрицы W в тензор $\mathcal{W} \in \mathbb{R}^{k_1 \times \ldots \times k_d}$. Здесь d – это размерность нового тензора, $k_1, \ldots, k_d$ – это внутренние размеры каждого измерения. Далее мы можем выполнить TT-разложение тензора $\mathcal{W}$ и получить набор матриц $G_m[i_m] \in \mathbb{R}^{r_{m-1} \times r_m}, i_m = 1, \ldots, k_m$, $m = 1, \ldots, d$ и $r_0 = r_d = 1$ такой, что каждый элемент тензора может быть представлен как $\mathcal{W}(i_1, i_2, \ldots, i_d) = G_1[i_1] G_2[i_2] \ldots G_d[i_d]$. Числа $r_0, \ldots r_m$ называются рангами Tensor Train разложения. Подобное TT разложение может быть эффективно имплементировано с помощью TT-SVD алгоритма, описанного в [26].

Обозначим эти две операции: конвертации матрицы W в тензор $\mathcal{W}$ и её последующее разложение в Tensor Train формат как одну операцию $\mathrm{TT}(W)$. Применяя её к обеим матрицам W and V в рекуррентном слое (2), мы получаем TT-RNN слой в следующей форме:

$$z_\ell^t = \tanh(\mathrm{TT}(W_l) x_{\ell-1}^t + \mathrm{TT}(U_l) x_\ell^{t-1} + b_\ell). \tag{8}$$

Мы провели ряд экспериментов с различными типами ячеек и с их различными размерами. Было показано, что главный выигрыш, который мы получаем от сжатия рекуррентных нейронных сетей с помощью низкоранговых матричных разложений в сравнении с прунингом и квантизацией, заключается в том, что при таком подходе мы почти не теряем в скорости вычислений. Выигрыш в памяти почти напрямую переносится в выигрыш в скорости. С



а)

б)

Рис. 2 — Архитектуры нейронных сетей: (a) оригинальная сеть LSTM 650-650, (b) модель, сжатая с помощью низкорангового разложения.

другой стороны, методы, которые работают с разреженными матрицами, позволяют получить большое сжатие по памяти, но долго работают при исполнении. Наши эксперименты с моделями для мобильного телефона подтверждают, что низкоранговое сжатие модели **LR LSTM 650-650** наиболее эффективное как по памяти, так и по вычислительной сложности.

В **четвертой главе** мы рассматриваем байесовские методы для сжатия нейронных сетей. Байесовские методы могут считаться математическии более обоснованной формой прунинга. В начале главы мы описываем общие принципы байесовских методов, затем рассматриваем применение вариационного дропаута (Variational Dropout, VD) [27] для прореживания нейронных сетей.

Далее мы представляем новый метод для сжатия рекуррентных нейронных сетей, который основан на автоматическом определении значимости с помощью двойного стохастического вариационного вывода. Суть его заключается в том, чтобы применить метод поиска релевантных признаков к выходному слою нейронной сети. Из-за того, что в нашей задаче мы обычно имеем дело с большим словарём, мы получаем высокоразмерную задачу. Для её решения мы используем стохастическую аппроксимацию. В уравнении 9 представлен

финальный функционал, оптимизируя который, мы находим параметры апостериорного распределения $q\left(W \mid \boldsymbol{\mu}, \boldsymbol{\sigma}\right)$, а также оптимальные λ^* для априорного.

$$\mathcal{L}\left(q\left(W \mid \boldsymbol{\mu}, \boldsymbol{\sigma}\right), \Lambda^*\right) = \mathbb{E}_{W \sim q(W|\boldsymbol{\mu},\boldsymbol{\sigma})}\left[\log p(Y \mid W, X)\right] + \\ + \frac{1}{2}\sum_{i=1}^{D}\sum_{j=1}^{K}\log\frac{\sigma_{ij}^2}{\mu_{ij}^2 + \sigma_{ij}^2} \longrightarrow \max_{\boldsymbol{\mu},\boldsymbol{\sigma}} \tag{9}$$

Выбирая уровень отсечения λ_{thresh} на валидационной части выборки, мы можем подобрать такой уровень, при котором мы получаем максимальное сжатие практически без потери качества.

Мы применили данный метод для прореживания выходного слоя в нашей задаче. Кроме того, мы впервые рассмотрели применение байесовских методов вместе с применением комбинирования весов входного-выходного слоя (weights sharing, tied weights). Из результатов экспериментов видно, что удаётся добиться сжатия выходного слоя до 50 раз в сравнении с изначальной моделью (удаляем до 98% параметров). Это больше, чем с помощью техник матричного разложения, но при худшей перплексии.

Байесовские методы сжатия действительно позволяют достичь большего разреживания, чем, например, матричные методы сжатия, но остаётся открытым вопрос эффективной имплементации таких методов для конкретных устройств. В конце главы предлагаются некоторые идеи по дальнейшему развитию этих методов.

В **заключении** приведены основные результаты работы:

1. Были разработаны алгоритмы сжатия скрытых слоев нейронных сетей для задачи моделирования языка с помощью методов матричных разложений.
2. Было предложено эффективное решение задачи классификации для большого числа классов (10–30 тыс.) с использованием методов матричных разложений для входного и выходного слоев нейронной сети.
3. Алгоритм DSVI-ARD был адаптирован для рекуррентных нейронных сетей в задаче моделирования языка.
4. Эффективные реализации рекуррентных нейронных сетей были портированы на мобильные устройства с использованием мобильных GPU и было проведено лабораторное тестирование в максимально приближенном к реальной жизни сценарии.

Список литературы

1. *Schmidhuber Jurgen*. Deep Learning in Neural Networks: An Overview // *Neural Networks*. — 2015. — Vol. 61. — Pp. 85–117.

2. *Bengio Yoshua*. Learning deep architectures for AI // *Foundations and Trends in Machine Learning*. — 2009. — Vol. 2, no. 1. — Pp. 1–127. — Also published as a book. Now Publishers, 2009.

3. *Deng Li, Yu Dong*. Deep Learning: Methods and Application // *Foundations and Trends in Signal Processing*. — 2013. — Vol. 7. — Pp. 197–387.

4. *Goodfellow Ian, Bengio Yoshua, Courville Aaron*. Deep Learning. — MIT Press, 2016. — http://www.deeplearningbook.org.

5. *Cybenko George*. Approximation by superpositions of a sigmoidal function // *MCSS*. — 1989. — Vol. 2, no. 4. — Pp. 303–314. https://doi.org/10.1007/BF02551274.

6. *Hornik Kurt, Stinchcombe Maxwell B., White Halbert*. Multilayer feedforward networks are universal approximators // *Neural Networks*. — 1989. — Vol. 2, no. 5. — Pp. 359–366. https://doi.org/10.1016/0893-6080(89)90020-8.

7. Learning both Weights and Connections for Efficient Neural Network / Song Han, Jeff Pool, John Tran, William J. Dally // Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. — 2015. — Pp. 1135–1143. http://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-ne

8. *Han Song, Mao Huizi, Dally William J.* Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding // *CoRR*. — 2015. — Vol. abs/1510.00149. http://arxiv.org/abs/1510.00149.

9. *Lu Zhiyun, Sindhwani Vikas, Sainath Tara N.* Learning compact recurrent neural networks // 2016 IEEE International Conference on Acoustics, Speech and

Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016. — 2016. — Pp. 5960–5964. https://doi.org/10.1109/ICASSP.2016.7472821.

10. *Arjovsky Martín, Shah Amar, Bengio Yoshua*. Unitary Evolution Recurrent Neural Networks // Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. — 2016. — Pp. 1120–1128. http://jmlr.org/proceedings/papers/v48/arjovsky16.html.

11. Tensorizing Neural Networks / Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, Dmitry P. Vetrov // Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. — 2015. — Pp. 442–450. http://papers.nips.cc/paper/5787-tensorizing-neural-networks.

12. *Kingma Diederik P, Salimans Tim, Welling Max*. Variational Dropout and the Local Reparameterization Trick // Advances in Neural Information Processing Systems 28 / Ed. by C. Cortes, N. D. Lawrence, D. D. Lee et al. — Curran Associates, Inc., 2015. — Pp. 2575–2583. http://papers.nips.cc/paper/5666-variational-dropout-and-the-local-reparameterization-t pdf.

13. *Molchanov Dmitry, Ashukha Arsenii, Vetrov Dmitry P.* Variational Dropout Sparsifies Deep Neural Networks // Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. — 2017. — Pp. 2498–2507. http://proceedings.mlr.press/v70/molchanov17a.html.

14. Structured Bayesian Pruning via Log-Normal Multiplicative Noise / Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, Dmitry P. Vetrov // Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. — 2017. — Pp. 6778–6787. http://papers.nips.cc/paper/7254-structured-bayesian-pruning-via-log-normal-multiplicat

15. *Grachev Artem M., Ignatov Dmitry I., Savchenko Andrey V.* Neural Networks Compression for Language Modeling // Pattern Recognition and Machine Intelligence - 7th International Conference, PReMI 2017, Kolkata, India, December 5-8, 2017, Proceedings. — 2017. — Pp. 351–357. https://doi.org/10.1007/978-3-319-69900-4_44.

16. Extraction of Visual Features for Recommendation of Products via Deep Learning / Elena Andreeva, Dmitry I. Ignatov, Artem M. Grachev, Andrey V. Savchenko // Analysis of Images, Social Networks and Texts - 7th International Conference, AIST 2018, Moscow, Russia, July 5-7, 2018, Revised Selected Papers. — 2018. — Pp. 201–210. https://doi.org/10.1007/978-3-030-11027-7_20.

17. *Grachev Artem M., Ignatov Dmitry I., Savchenko Andrey V.* Compression of recurrent neural networks for efficient language modeling // *Applied Soft Computing*. — 2019. — Vol. 79. — Pp. 354 – 362. http://www.sciencedirect.com/science/article/pii/S1568494619301851.

18. Efficient Language Modeling with Automatic Relevance Determination in Recurrent Neural Networks / Maxim Kodryan, Artem Grachev, Dmitry Ignatov, Dmitry Vetrov // ACL 2019, Proceedings of the 4th Workshop on Representation Learning for NLP, August 2, 2019, Florence, Italy. — 2019.

19. *Kneser Reinhard, Ney Hermann*. Improved backing-off for M-gram language modeling // 1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '95, Detroit, Michigan, USA, May 08-12, 1995. — 1995. — Pp. 181–184. https://doi.org/10.1109/ICASSP.1995.479394.

20. *Jelinek Frederick*. Statistical Methods for Speech Recognition. — MIT Press, 1997. https://mitpress.mit.edu/books/statistical-methods-speech-recognition.

21. A Neural Probabilistic Language Model / Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Janvin // *Journal of Machine Learning Research*. — 2003. — Vol. 3. — Pp. 1137–1155. http://www.jmlr.org/papers/v3/bengio03a.html.

22. *Mikolov Tomáš*. Statistical Language Models Based on Neural Networks: Ph.D. thesis / Brno University of Technology. — 2012. — P. 129. http://www.fit.vutbr.cz/research/view_pub.php?id=10158.

23. *Werbos P.* Backpropagation through time: what it does and how to do it // *Proceedings of the IEEE*. — 1990. — Vol. 78(10). — Pp. 1550–1560.

24. *Rumelhart David E., Hinton Geoffrey E., Williams Ronald J.* Neurocomputing: Foundations of Research / Ed. by James A. Anderson, Edward Rosenfeld. — Cambridge, MA, USA: MIT Press, 1988. — Pp. 696–699. http://dl.acm.org/citation.cfm?id=65669.104451.

25. *Pascanu Razvan, Mikolov Tomas, Bengio Yoshua*. On the difficulty of training recurrent neural networks // Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013. — 2013. — Pp. 1310–1318. http://jmlr.org/proceedings/papers/v28/pascanu13.html.

26. *Oseledets Ivan V.* Tensor-Train Decomposition // *SIAM J. Scientific Computing*. — 2011. — Vol. 33, no. 5. — Pp. 2295–2317. http://dx.doi.org/10.1137/090752286.

27. *Chirkova Nadezhda, Lobacheva Ekaterina, Vetrov Dmitry P.* Bayesian Compression for Natural Language Processing // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. — 2018. — Pp. 2910–2915. https://aclanthology.info/papers/D18-1319/d18-1319.