УДК 004.85 DOI 10.18522/2311-3103-2026-3-152-172

К.И. Ралко, Н.Е. Сергеев

АЛЬТЕРНАТИВНЫЕ ПОДХОДЫ К МАСШТАБИРОВАНИЮ NLP МОДЕЛЕЙ: АНАЛИЗ ПОДХОДОВ К ОПТИМИЗАЦИИ ОБЪЕМА ДАННЫХ И ВЫЧИСЛЕНИЙ ПРИ ОБУЧЕНИИ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Основное внимание уделяется подходам преодоления системных ограничений парадигмы масштабирования больших языковых моделей (LLM), связанных с исчерпанием данных и экспоненциальным ростом вычислительных затрат. Обход таких ограничений позволяет разработать более эффективные подходы к созданию NLP-моделей без потери в качестве их работы. Целью данного исследования является сравнительный анализ эффективности стандартной трансформерной архитектуры (nanoGPT) и модели, оперирующей семантическими эмбеддингами (nanoSonar), для задач языкового моделирования в условиях ограниченных ресурсов. Работа с концептуальными эмбеддингами позволяет выявить более глубокие языковые закономерности и сократить объем требуемых данных для обучения, что значительно повышает эффективность моделирования. При проведении исследования использовался набор данных TinyStories, который включает короткие нарративы с четкой структурой. Перед реализацией моделей была проведена предобработка данных: для nanoGPT – токенизация методом BPE, для nanoSonar – преобразование текста в семантические эмбеддинги с помощью предобученной модели Sonar. Оценка моделей осуществлялась с использованием метрик функции потерь (loss) и перплексии (perplexity). Результаты исследования показали, что модель nanoSonar обеспечивает значительно более низкую перплексию (6,609 против 39,151 у nanoGPT), а также демонстрирует более устойчивую динамику обучения на поздних этапах. В работе представлен анализ современных подходов к оптимизации масштабирования (MoE, дистилляция, PEFT) и перспективных архитектур (LRM, SSM, RWKV), а также даны практические рекомендации по применению моделей, работающих в пространстве семантических эмбеддингов, для доменно-специфичных задач и систем с ограниченными вычислительными ресурсами. Результаты исследования могут быть полезны при разработке эффективных языковых моделей, сочетающих высокое качество генерации с экономичной архитектурой.

Большие языковые модели; масштабирование; оптимизация вычислений; архитектура Transformer; семантические эмбеддинги; NLP; машинное обучение; эффективность данных.

K.I. Ralko, N.E. Sergeev

ALTERNATIVE APPROACHES TO NLP MODEL SCALE-UP: AN ANALYSIS OF APPROACHES TO OPTIMIZING DATA AND COMPUTATION VOLUME WHEN TRAINING LARGE-SCALE LANGUAGE MODELS

This paper focuses on overcoming the systemic limitations of the large-scale language model (LLM) scaling paradigm, which are related to data exhaustion and exponential growth in computational costs. This enables the development of more efficient approaches to building NLP models without sacrificing their performance. The goal of this study is to compare the performance of a standard transformer architecture (nanoGPT) and a model using semantic embeddings (nanoSonar) for language modeling tasks under resource constraints. Working with conceptual embeddings allows us to identify deeper linguistic patterns and reduce the amount of required training data, significantly improving modeling efficiency. The study utilized the TinyStories dataset, which includes short narratives with a clear structure. Before implementing the models, the data was preprocessed: for nanoGPT, tokenization was performed using the BPE method, and for nanoSonar, text was converted into semantic embeddings using a pretrained Sonar model. The models were evaluated using the loss and perplexity metrics. The results showed that the nanoSonar model provides significantly lower perplexity (6.609 versus 39.151 for nanoGPT) and demonstrates more robust training dynamics at later stages. This paper presents an analysis of modern approaches to scaling optimization (MoE, distillation, PEFT) and promising architectures (LRM, SSM, RWKV), and provides practical recommendations for applying models operating in the space of semantic embeddings to domain-specific problems and systems with limited computational resources. The results of this study can be useful in developing efficient language models that combine high generation quality with a cost-effective architecture.

Large language models; scaling; computational optimization; Transformer architecture; semantic embeddings; NLP; machine learning; data efficiency.

Введение. Современный этап развития обработки естественного языка (Natural Language Processing, NLP) характеризуется доминированием парадигмы масштабирования (Scaling Laws), в рамках которой прогресс в области больших языковых моделей (Large Language Models, LLM) достигается преимущественно за счет увеличения объема обучающих данных, вычислительных ресурсов и количества параметров моделей [1]. Этот подход, доказавший свою эффективность в моделях GPT, PaLM, Llama и других, позволил достичь значительных успехов в решении задач генерации текста, перевода и рассуждений [2].

Однако экстенсивный путь развития, основанный на непрерывном увеличении количества параметров моделей нейронных сетей и обучающих выборок, сталкивается с системными ограничениями. С одной стороны, наблюдается приближение к физическим пределам доступности высококачественных текстовых данных – согласно прогнозам, запасы релевантных общедоступных данных будут исчерпаны в период с 2026 по 2032 годы [3]. С другой стороны, экспоненциальный рост требований к вычислительным ресурсам создает экономические барьеры для внедрения, делая дальнейшее масштабирование экономически нецелесообразным для широкого спектра экономических отраслей.

Наряду с ресурсными ограничениями, проявляются фундаментальные методологические проблемы текущего подхода. Архитектура Transformer, лежащая в основе большинства современных LLM, демонстрирует такие системные недостатки, как поверхностность абстракций, связанная с токенизацией текста, необходимость экстернализации процессов рассуждения и архитектурный разрыв с принципами биологической обработки информации.

В этом контексте актуальной исследовательской задачей становится поиск альтернативных подходов к масштабированию NLP-моделей, позволяющих преодолеть ограничения экстенсивного пути развития. Данная работа посвящена систематическому анализу и экспериментальной оценке таких подходов, сфокусированных на оптимизации использования данных и вычислительных ресурсов без компромисса в качестве генерализации.

Целью исследования является сравнительный анализ современных методов оптимизации объема данных и вычислений при обучении больших языковых моделей, включая архитектурные модификации, алгоритмические улучшения и альтернативные парадигмы представления информации. Особое внимание уделяется экспериментальной верификации эффективности предложенных подходов на реальных задачах языкового моделирования.

Проблемы и ограничения парадигмы масштабирования в развитии NLP-моделей. Проблема масштабирования – это системное противоречие, при котором улучшение способности больших языковых моделей (LLM) к решению новых задач (обобщающей способности) объективно требует экспоненциального роста объема обучающих данных, вычислительной мощности и числа параметров модели. Однако такой рост упирается в практические пределы доступных ресурсов и фундаментальные ограничения существующих архитектур и парадигмы обучения.

Данная проблема является центральной для современного этапа развития искусственного интеллекта. Ее суть раскрывается в двух взаимосвязанных и противоречащих друг другу аспектах, которые наглядно представлены в табл. 1.

Таблица 1

Обзор аспектов формулируемой дилеммы

Аспект	Сущность аспекта	Ключевые вызовы
Позитивный	Строгое следование эмпирически установленным законам масштабирования (Scaling Laws) как основному драйверу прогресса в области LLM и движения к Общему Искусственному Интеллекту (ОИИ).	Достижение универсальной обобщающей способности (generalization) модели, то есть ее способности эффективно работать с новыми, ранее не встречавшимися данными.
Негативный	Столкновение экстенсивного пути роста с практическими пределами и фундаментальными методологическими проблемами, присущими текущему подходу.	Исчерпание данных, экономическая нецелесообразность внедрения решений на базе LLM, архитектурные недостатки (склонность к переобучению), приводящие к непредсказуемому поведению на новых данных.

Ключевые проблемы применения парадигмы масштабирования. Действительно, не смотря на подтвержденный факт улучшения способностей существующих языковых моделей к обработке обобщенных данных, дальнейшее наращивание масштаба сталкивается с рядом ограничений.

Такие ограничения можно рассматривать согласно двум большим группам: ресурсным ограничениям и методологическим ограничениям.

Анализ ресурсных ограничений парадигмы масштабирования. Ресурсные ограничения в контексте больших языковых моделей носят дуальный характер. С одной стороны, вычислительные ограничения, связанные с экспоненциальным ростом требований к аппаратным ресурсам, могут рассматриваться как относительные – потенциально преодолимые за счет развития и удешевления вычислительных мощностей. С другой стороны, ограничения объема качественных данных для обучения носят более фундаментальный, физический характер, поскольку запасы релевантных текстовых данных конечны.

Вычислительные ограничения проявляются в экспоненциальном росте затрат на обучение и инференс (рабочий режим модели) современных LLM.

Обучение современных моделей уровня SotA (англ. State-of-the-Art – уровень модели ИИ, достигающей наилуших результатов в тестировании) требует значительных затрат вычислительных ресурсов и, как следствие, задействования десятков тысяч специализированных GPU (англ. Graphical Processing Unit – Видеоускоритель) таких как NVIDIA A100 или H200 в течение месяцев работы. Это создает экономический барьер, ограничивающий исследования узким кругом крупных корпораций. Более того, чрезмерная зависимость от конкретных аппаратных платформ порождает явление «аппаратной лотереи» (Hardware Lottery), где жизнеспособность инновационных архитектур часто определяется их совместимостью с доминирующим оборудованием, а не их внутренней эффективностью [4].

Помимо прочего, эмпирические наблюдения демонстрируют экспоненциальный разрыв между скоростью роста требований моделей к вычислительным ресурсам и скоростью роста вычислительных возможностей. В то время как, согласно эмпирическому закону Мура, количество транзисторов на интегральной схеме, а следовательно, и вычислительная мощность удваивается каждые два года, наблюдения демонстрируют, что требования к вычислениям при обучении современных моделей нейронных сетей удваиваются каждые 6 мес [1]. Визуализация разрыва роста вычислительных мощностей и требований представлена на рис. 1.

Рис. 1. Визуализация разрыва между ростом вычислительных мощностей и требованиями LLM

Чтобы преодолеть разрыв, в рамках подготовки моделей зачастую применяется подход с горизонтальным масштабированием вычислений, что в свою очередь порождает проблему передачи данных и эффективного применения параллельных вычислений, что выражается в понижении степени полезной работы GPU.

В свою очередь, ограничения данных проявляются в феномене «Великого дефицита данных» (Great Data Famine). Согласно прогнозам, запасы высококачественных общедоступных текстовых данных будут исчерпаны между 2026 и 2032 годами [2]. Это вынуждает разработчиков использовать данные более низкого качества (текстовые корпусы, ко-

торые характеризуются низкой информационной энтропией, высоким уровнем шума, наличием статистических аномалий и смещений, а также недостаточной репрезентативностью), что приводит к нескольким негативным последствиям:

- ♦ модели начинают запоминать шум и артефакты обучающей выборки вместо выявления обобщаемых закономерностей;
- ♦ качество работы на новых данных ухудшается из-за недостаточной репрезентативности обучающего корпуса текста;
- ♦ усиление смещений текстов [2].

Данные ограничения создают петлю зависимостей: для компенсации падения качества данных требуется увеличение вычислительных ресурсов, что в свою очередь усугубляет проблему ресурсных ограничений.

Анализ методологических ограничений парадигмы масштабирования. Современный подход к обучению LLM, основанный на парадигме прогнозирования следующего токена, сталкивается с системными методологическими ограничениями, которые не могут быть решены простым масштабированием параметров или данных.

В рамках таких ограничений можно выделить нижеописанные аспекты.

Поверхностность абстракций. Исходя из того, что основной механизм работы LLM основан на лексической токенизации, а именно сегментации входного текста на части слов или токены, где каждый токен отображается в дискретный вектор эмбеддинга, генерация текста, таким образом, сводится к последовательному предсказанию наиболее вероятного следующего дискретного токена на основе статистических корреляций в высокоразмерном семантическом пространстве. В отличие от такого подхода, человеческое познание оперирует абстрактными, иерархическими концепциями, чья вербализация может принимать форму части слова, предложения или всего текста. Критический анализ показывает, что зависимость LLM от дискретных эмбеддингов приводит к поверхностности абстракций, поскольку модель выявляет статистические паттерны, а не формирует каузальные модели мира или интуитивные теории. Дополнительно, проблема токенизационного «бутылочного горлышка» (например, фрагментация структурно единых сущностей) затрудняет рассуждение, связанное с целостными концептами [5]. Помимо прочего, как следствие проблемы отсутствия иерархии эмбеддингов, LLM, как во время обучения, так и во время инференса, требует разительно большее количество ресурсов для достижения верного ответа, так как «путь» ответа в пространстве эмбеддингов над токенами может содержать большее количество шагов (генерируемых токенов), чем в иерархически более высоком пространстве эмбеддингов концепций. Более того, как следствие проблемы, существующим LLM необходимо большее количество примеров разнородных данных в рамках одной концепции для выделения общности, что в свою очередь усугубляет в том числе и проблему ограничения доступных данных.

Необходимость экстернализации рассуждений. В основе принципа своей работы, существующие LLM выполняют задачу множественной классификации. В контексте построения модели рассуждений и выделения абстракций, важно отметить, что не смотря на множественный проход по входной выборке при инференсе, такие модели не могут воспроизводить процесс мышления без генерации текста. Частичным решением такой проблемы стало появление механизма Chain-of-Thoughts (англ. цепь мышления) в рамках которого такая модель производит предварительную генерацию текста перед выдачей конечного результата, однако такой подход энергозатратен, так как, зачастую, такой текст размышлений модели превышает по объему конечный ответ модели. Более того, такой подход оперирует в рамках того же пространства эмбеддингов что и прямой процесс работы модели и, следовательно, лишь усугубляет проблему поверхностности абстракций [6].

Архитектурный разрыв с биологическим интеллектом. В отличии от биологических нейронных систем, работающих в режиме асинхронной событийно-ориентированной обработки, современные LLM используют синхронные детерминистические вычисления [7]. Действительно, структурное сходство с человеческим мозгом не является самоцелью, однако такое сравнение подсвечивает негативные аспекты существующих моделей LLM, а именно ограниченные способности к эффективной обработке динамических временных последовательностей, адаптивному обучению и энергоэффективной работе.

Обзор подходов к оптимизации данных и вычислений при обучении больших языковых моделей. Решение проблемы масштабирования, сформулированной ранее, требует перехода от экстенсивной парадигмы масштабирования к интенсивным методам развития. Данный раздел систематизирует современные исследовательские направления, нацеленные на оптимизацию использования данных и вычислительных ресурсов без снижения способности моделей к обобщению.

В настоящее время существует большое количество подходов к оптимизации требований к вычислительным ресурсам моделей. Это такие методы как:

- применение типов данных меньшей точности (квантизация) для хранения весов модели;
- обрезка (прунниг) весов;
- аппаратная оптимизация механизмов работы модели: механизм Flash Attention [8] и др.

Действительно, применение таких методов позволяет добиться значительной оптимизации как скорости обучения и работы моделей, так и их инференса, однако являются скорее количественными улучшениями, чем качественными, поэтому не рассматриваются детально в контексте настоящей работы.

Применяемые методы оптимизации процесса обучения и данных. Данная категория методов включает зрелые, широко применяемые в индустрии методы, которые нацелены на повышение эффективности обучения, снижение вычислительных затрат на инференс и сокращение требований к объему обучающих данных.

Смесь экспертов (Mixture of Experts, MoE). Архитектура MoE позволяет достичь большого размера модели (триллионы параметров) при сохранении вычислительной стоимости обработки одного набора входных данных [9]. Это достигается путем маршрутизации входных токенов к активации лишь небольшого подмножества «экспертов». MoE является эффективным решением для масштабирования моделей, поскольку позволяет увеличивать количество параметров, не увеличивая экспоненциально затраты на инференс.

Дистилляция знаний (Knowledge Distillation, KD). Дистилляция знаний представляет собой процесс передачи функционала от большой, вычислительно дорогой модели-«учителя» к меньшей, более эффективной модели-«ученику».[10] Целью является создание компактной и легко развертываемой модели-ученика с сопоставимой точностью. KD может осуществляться через имитацию выходных распределений вероятностей учителя (soft labels) или через использование учителя для генерации синтетических обучающих данных. KD, примененная во время тонкой настройки с учителем, может повысить точность на бенчмарках (таким как MMLU) и потребовать до 30% [11] меньше шагов обучения, чем при стандартном обучении модели на общих датасетах.

Параметрически-эффективное дообучение (Parameter-Efficient Fine-Tuning, PEFT). Методы PEFT, такие как LoRA (Low-Rank Adaptation – Низкорановая адаптация), позволяют адаптировать большие предварительно обученные модели (LLM) к новым задачам без обновления всех параметров. PEFT замораживает основные веса и обучает только малые, низкоранговые матрицы адаптеров. Это открывает путь к доступной персонализации LLM [12].

Метод LoRA основан на гипотезе о том, что эффективное изменение весов (ΔW) имеет низкий внутренний ранг, который может быть аппроксимирован низкоранговым разложением.

Для исходной матрицы весов $W_0 \in R^{d*k}$, обновление ΔW представляется произведением двух матриц меньшего размера, B и A: $\Delta W = BA$, где $B \in Rd * r$ и $A \in Rr * k$. Ранг разложения r выбирается так, что $r < \min(d, k)$.

Во время прямого шага модели, выход h вычисляется с учетом исходных замороженных весов W_0 и обучаемого обновления ΔW:

$$h = W_0 x + \Delta W x = W_0 x + BAx,$$

где x – входной вектор. Поскольку тренируются только параметры матриц A и B, число обучаемых параметров резко сокращается.

Финальные веса $W_{new} = W_0 + BA$ могут быть явно вычислены и сохранены до развертывания модели, что устраняет дополнительную задержку инференса.

Перспективные методы оптимизации архитектуры модели и данных. Данная категория включает инновационные подходы, нацеленные на повышение информационной плотности данных и устранение фундаментальных архитектурных ограничений моделей LLM, решая проблемы, связанные с поверхностью абстракций и вычислительной неэффективностью рассуждений.

Дистилляция данных и внутренних представлений модели. Данное направление фокусируется на повышении интенсивности использования информации, либо путем оптимизации самого обучающего корпуса, либо путем переноса представлений знаний при подготовке ответа моделью.

Дистилляция данных. В ответ на проблему дефицита данных DD направлена на синтез малого, но информационно-плотного набора данных, который сохраняет ключевые обучающие свойства полного исходного датасета [13]. Действительно, применение таких подходов оптимизации вычислений активно применяется в отрасли и, в первую очередь, показывает наивысшую результативность в задачах распознавания и классификации графических изображений. Таким образом, в работе «Алгоритм предварительной обработки видеоизображений для повышения точности обнаружения малоразмерных образов» авторы путем предобработки изображений повышают итоговую метрику MAP (Mean Average Precision – Средняя точность) на 7% [24]. В рамках развития смежного направления – предварительной обработки данных с целью направленного расширения пространства признаков, авторы успешно повышают метрику MAP уже на 23,65% [25], что в свою очередь позволяет производить запуск модели в том числе и на маломощных встраиваемых устройствах [26].

Перспективные подходы дистилляции данных, такие как Matching Training Trajectories (MTT), оптимизируют синтетические данные для имитации динамики изменения параметров экспертной модели, что обеспечивает высокую эффективность обучения производных моделей на таких данных, и потенциал для кросс-архитектурного трансфера знаний [14].

Дистилляция внутренних представлений модели. Эта техника дистилляции моделей требует, чтобы модель-ученик имитировала не только конечные вероятности учителя, но и его внутренние механизмы, такие как распределение внимания (attention maps) или промежуточные векторные представления (эмбеддинги).[15] Это позволяет малой модели научиться выделять наиболее важные для учителя части входных данных, повышая эффективность и точность.[15]

В отличии от классических методов дистилляции, дистилляция внутренних представлений (Internal Representation Matching) направлена на передачу способа получения ответа, а не только конечного результата. Это достигается путем минимизации функции потерь сходства L_{Sim} между промежуточными активациями (эмбеддингами или картами внимания) модели-учителя (T) и модели-ученика (S).

Общая идея заключается в согласовании представлений R_t^l и R_s^l на соответствующем слое l.

$$l_{sim} = \frac{1}{|L|}\sum_{l\in L} E_x\sim x,$$

где L – множество слоев для согласования, а Sim – функция сходства, основанная на L_2 норме.

Вариации архитектуры Transformer. Преимущественное большинство современных LLM являются моделями трансформерной архитектуры. В отличии от рекуррентных архитектур, принимающих входную последовательность данных (токенов) синхронно и последовательно, трансформерные модели оперируют всем входным контекстом за один такт работы, что позволяет производить эффективное распараллеливание вычислений. Однако, при таком подходе, сложность вычислений возрастает до квадратичной, сама работа модели становится требовательной к памяти.

В свою очередь, механизм Attention (англ. Attention – внимание), позволяет трансформерной модели строить сложные взаимосвязи между частями как входных, так и выходных данных, однако такой механизм не позволяет модели оперировать иерархией абстракций.

Ниже приведен перечень применяемых и перспективных вариаций архитектуры Трансформер, снижающих негативные аспекты применяемой архитектуры.

LongFormer и модели с разреженным вниманием. Настоящий подход является попыткой решения проблемы квадратичной сложности механизма внимания. В рамках такого подхода, вместо выделения результатов вычисления функции внимания между всеми токенами, применяется гибридный подход, заключающийся в формировании метода скользящего окна внимания. При таком подходе, с целью выделения локального контекста, каждый токен взаимодействует исключительно с соседними токенами в пределах фиксированного окна, в то время как с целью реализации глобальное внимания, выбираются некоторое количество особо-важных токенов. Такой подход снижает вычислительную сложность до линейной относительно длины последовательности, позволяя модели эффективно обрабатывать документы длиной до десятков тысяч токенов, что критически важно в контексте обработки длинных текстов [16].

Large Reasoning Models (LRM). Архитектура, нацеленная на создание возможности внутренней, ненаблюдаемой калькуляции предобработки ответа (мышления), отделенной от дорогостоящего процесса генерации текста [17]. В отличие от Chain-of-Thought (CoT), который является вынужденным механизмов экстернализации рассуждений, LRM стремятся реализовать энергоэффективную систему, преодолевая вычислительную неэффективность, присущую автоэнкодерным моделям.

Large Concept Models (LCM). Архитектура, представляющая собой попытку преодолеть зависимость от дискретных токенов и перейти к оперированию высокоуровневыми семантическими концептами [5]. LCM работают с концептуальными единицами вместо токенов, что, как ожидается, позволит достичь более глубокого системного обобщения и устранит ограничения, связанные с поверхностью абстракций, свойственных стандартным LLM [18].

С технической точки зрения, основное отличие такой модели от классического трансформера заключается в пространстве оперируемых эмбеддингов. Так, в рамках цикла работы модели происходит предсказание следующего ембеддинга, однако такой эмбеддинг не имеет жесткой привязки к части слова и может отражать более высокоуровневый концепт, что, в общем случае, может позволить прийти к конечному результату за меньшее количество итераций предсказания.

Рассмотрим процесс генерации токенов LLM и LCM на примере генерации текста: «Тим не был очень спортивным. Он думал, что это изменится, если он вступит в спортивную секцию. Он попробовал записаться в несколько команд. Его не приняли ни в одну из них. Он решил тренироваться самостоятельно.»

На рис. 2 представлена визуализация пространства эмбеддингов, применяющихся в LLM и LCM.

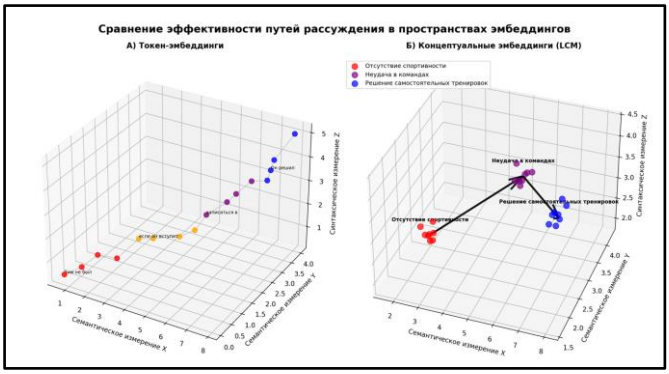

Рис. 2. Визуализация пространства эмбеддингов, применяющихся в LLM и LCM

Визуализация хода предсказания эмбеддингов в моделях LLM и LCM представлена на рис. 3.

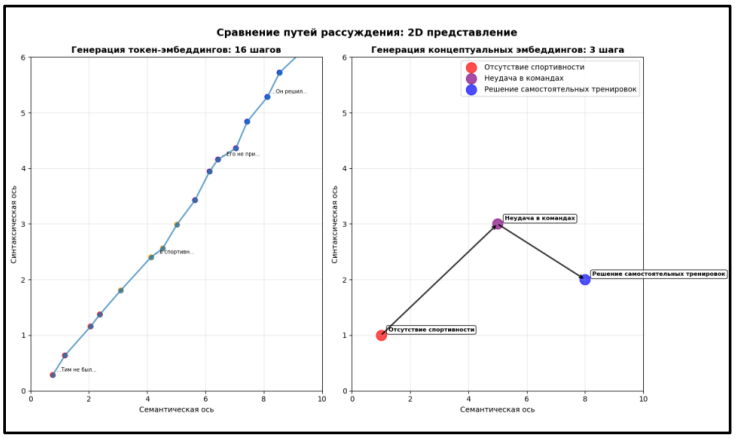

Рис. 3. Визуализация хода предсказания эмбеддингов в моделях LLM и LCM

Исходя из того, что такие концепции не являются человеко-читаемыми текстами, помимо классических блоков архитектуры трансформеров, такие модели включают в себя модуль декодера концепций, вербализующий выходные эмбеддинги в текст.

Дополнительным преимуществом такого подхода является когерентность процесса генерации текста описываемой модели диалектического субъективного познания, описывающей цикл мышления посредством индуктивно-дедуктивного цикла.

Альтернативы архитектуре Трансформера. В то время как вариации архитектуры Трансформера, рассмотренные в предыдущем пункте, направлены на эволюционное улучшение исходной модели, данное направление представляет собой поиск радикально иных архитектурных принципов.

Зачастую, архитектуры, являющиеся альтернативами трансформрам в контексте обработки текста являются модификациями рекуррентных нейронных сетей (англ. RNN – Recurrent Neural Network).

Среди них важно выделить две основные архитектуры, применяемые на данный момент в контексте NLP.

State Space Models (SSM). В основе SSM-подхода лежит идея моделирования скрытого состояния $h(t)$системы, которое эволюционирует во времени под воздействием входной последовательности $x(t)$. Дискретизованная форма может быть представлена следующими уравнениями:

$$h(t) = A * h(t-1) + B * x(t),$$
$$y(t) = C * h(t),$$

где A – матрица состояния, определяющая эволюцию системы, B – матрица входа, C – матрица выхода.

Ключевой инновацией в современных SSM (таких как модель Mamba) является введение селективного механизма, который позволяет параметрам модели (A, B, C) динамически зависеть от входных данных. Это дает модели способность избирательно запоминать или игнорировать информацию в зависимости от контекста, приближаясь по гибкости к механизму внимания, но сохраняя линейную сложность [19].

Важной общей модификацией таких сетей является множественность форм представления таких моделей. С одной стороны, в рамках цикла обучения, такие модели представляются как аналог сверточных сетей, что облегчает параллелизировать процесс вычисления.

Виды представлений архитектуры SSM представлены на рис. 4.

RWKV. Архитектура RWKV (Receptance Weighted Key Value) представляет собой масштабируемую архитектуру рекуррентной нейронной сети (RNN), которая сочетает в себе эффективное параллелизуемое обучение, характерное для Трансформеров, с высокоэффективным инференсом, присущим RNN [20].

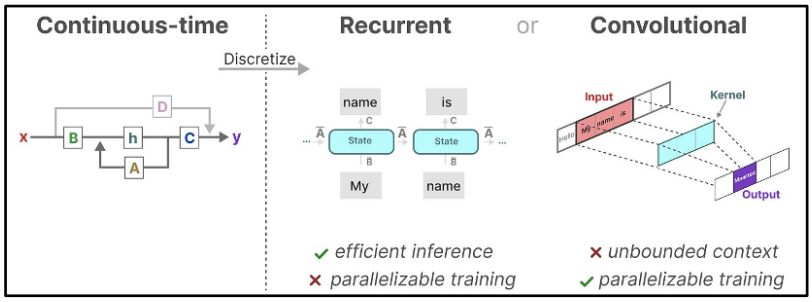

Рис. 4. Визуализация видов представлений архитектуры SSM

Ключевым нововведением RWKV является отказ от классического механизма самовнимания в пользу линейного механизма внимания. Это позволяет формулировать модель в двух режимах:

Временной параллельный режим (time-parallel mode): как и у Трансформеров, используется для эффективного параллелизуемого обучения на больших последовательностях.

RNN-режим: используется на этапе инференса, когда модель обрабатывает последовательность шаг за шагом, рекуррентно обновляя свое скрытое состояние (state).

Благодаря такой архитектуре, RWKV преодолевает фундаментальное ограничение традиционных RNN – неспособность к эффективному параллельному обучению – и одновременно решает проблему квадратичной сложности инференса Трансформеров. Архитектура реализована через чередование двух основных блоков: Time Mixing (временное смешивание) и Channel Mixing (канальное смешивание)

Нейробиологически-инспирированные подходы. Данное исследовательское направление ставит своей целью преодоление фундаментального архитектурного разрыва между искусственными и биологическими нейронными системами. В отличие от синхронных, детерминистических вычислений, присущих Трансформерам и их вариациям, биологический интеллект функционирует на принципах асинхронной, событийно-ориентированной и энергоэффективной обработки информации. Нейробиологически-инспирированные подходы ставят своей целью не столько буквальное воспроизведение структуры мозга, сколько заимствование ключевых вычислительных принципов для создания более эффективных и устойчивых моделей ИИ, способных к непрерывному обучению и адаптации.

Основным направлением в нейробиологических подходах являются **Спайковые нейронные сети (Spiking Neural Networks, SNN).** Спайковые нейронные сети представляют собой третье поколение моделей нейронных сетей, наиболее близко имитирующее принципы работы биологического мозга. Ключевым отличием SNN от традиционных искусственных нейронных сетей (ANC) и даже от архитектур вроде Трансформера является использование временно́го кодирования информации в виде дискретных электрических импульсов – спайков [21].

В SNN нейроны не передают друг другу непрерывные значения активации (как в ANC) или взвешенные суммы (как в механизме внимания). Вместо этого они обмениваются бинарными спайками в различные моменты времени, когда их внутренний мембранный потенциал превышает определенный порог. Этот процесс моделируется с помощью набора уравнений, описываемых моделью. Одним из примеров реализации такого подхода является модель Leaky Integrate-and-Fire (LIF), которая является широко распространенной абстракцией над процессами, происходящими в биологическом нейроне:

1. **Накопление (Integration)**: нейрон накапливает входные сигналы от пресинапсов, увеличивая свой мембранный потенциал.
2. **Утечка (Leak)**: потенциал постепенно уменьшается со временем, что добавляет сети временну́ю динамику и "память" о недавних событиях.
3. **Спайк (Fire)**: если потенциал достигает порогового значения, нейрон генерирует спайк и передает его постсинаптическим нейронам, после чего его потенциал сбрасывается.

Визуализация архитектуры LIF-нейрона, а также временной динамики такого нейрона представлена на рис. 5.

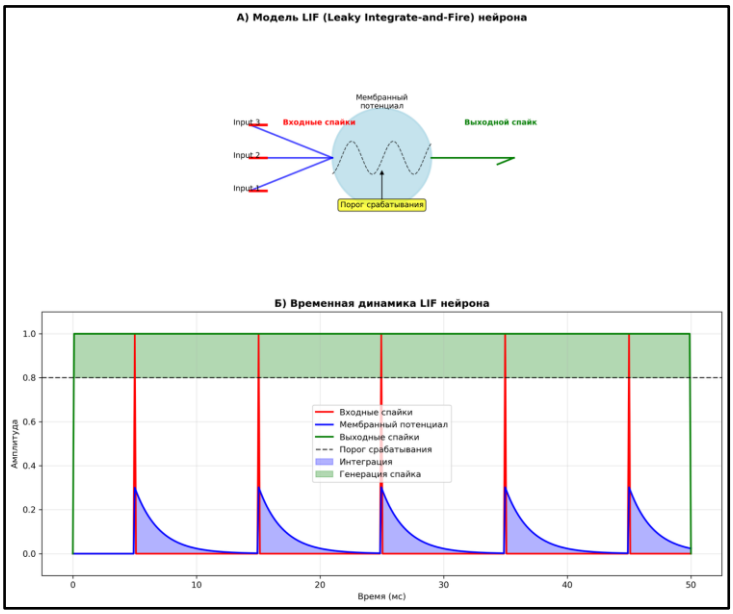

Рис. 5. Визуализация архитектуры LIF-нейрона, а также временной динамики LIF нейрона

Именно событийно-ориентированная природа SNN лежит в основе их потенциального превосходства в энергоэффективности:

♦ **Активность по событию**: вычисления в SNN происходят только при поступлении спайка. В отличие от Трансформеров, где матричные умножения выполняются для всего контекста синхронно на каждом слое, SNN остаются в основном в пассивном состоянии, что радикально снижает энергопотребление. При развертывании на специализированном **нейроморфном оборудовании** (например, чипы Intel Loihi или IBM TrueNorth) демонстрируется повышение энергоэффективности инференса более чем в 32 раза по сравнению с традиционными GPU.

♦ **Временная динамика вместо тактовой частоты**: SNN оперируют непрерывным временем, что позволяет им более естественно и эффективно обрабатывать последовательные данные (речь, видео, сенсорные потоки), извлекая паттерны непосредственно из временных задержек между спайками.

Архитектура SNN предлагает несколько свойств, способствующих применению таких моделей в контексте AGI (Искусственного Общего Интеллекта). Среди них:

♦ **Непрерывное обучение (Continual Learning)**: благодаря синаптической пластичности (изменению силы связей между нейронами на основе их совместной активности, например, по правилу STDP — Spike-Timing-Dependent Plasticity), SNN способны обучаться «на лету», адаптируясь к новым данным без катастрофического забывания, что является серьезной проблемой для современных LLM.

♦ **Нативная внутренняя память**: скрытое состояние сети (мембранные потенциалы нейронов) является формой аналоговой, распределенной памяти, которая существует непрерывно во времени. Это устраняет необходимость в механизмах внешней памяти или явных контекстных окнах, присущих Трансформерам, и позволяет модели поддерживать постоянный, обновляемый контекст.

Помимо "чистых" SNN, существуют гибридные подходы, которые стремятся совместить энергоэффективность спайковых сетей с вычислительной мощностью глубокого обучения. В контексте настоящей работы особенно важно рассмотреть спайкинговые

трансформерные архитектуры (Spiking Transformers) – такое направление моделей, в котором механизм самовнимания (механизм Self-Attention) и/или полносвязные слои адаптируются для работы со спайковыми нейронами.

Такая модификация позволяет применять мощные архитектурные принципы Трансформеров в событийно-ориентированном контексте, что полезно в рамках задач NLP.

Экспериментальное исследование эффективности подходов к оптимизации. Для эмпирической проверки гипотезы о том, что использование семантических представлений высокого уровня способствует более эффективному обучению, было проведено сравнительное исследование двух архитектур: стандартного трансформера и его модификации, адаптированной для работы с концептуальными эмбеддингами

Основная гипотеза: Модель, оперирующая семантическими эмбеддингами на уровне предложений (nanoSonar), достигнет сопоставимого или более высокого качества генерации при меньшем объеме данных и более быстрой сходимости по сравнению с моделью, использующей стандартную токенизацию (nanoGPT), благодаря работе в пространстве более высокоуровневых абстракций.

Эксперимент построен как прямое сравнительное тестирование двух моделей с максимально схожей базовой архитектурой, обученных на идентичном содержании, но представленном в разных формах.

Описание экспериментального стенда. В рамках проведения эксперимента был подготовлен тестовый стенд, спроектированный для обеспечения максимальной сопоставимости результатов.

Ниже приводится описание аспектов тестового стенда.

Архитектура моделей. В качестве базовой модели генеративного трансформера был взят проект nanoGPT.

С целью обеспечения максимального сходства сравниваемых моделей, была подготовлена модификация проекта nanoGPT, садаптированная для обучения на базе концептуальных эмбеддингов в пространстве Sonar. Такая модификация была названа nanoSonar.

Обе модели имеют идентичную базовую архитектуру трансформера с параметрами, обеспечивающими общее количество обучаемых параметров порядка 50 миллионов. Конфигурация моделей представлена в табл. 2.

Таблица 2

Сравнение конфигурации моделей nanoGPT и nanoSonar

Параметр	nanoGPT	nanoSonar
Базовая архитектура	Трансформер (GPT)	Трансформер (GPT)
Количество слоев (n_layer)	8	8
Количество голов внимания (n_head)	8	8
Размерность скрытого слоя (n_embd)	512	512
Входные данные	Токены (BPE)	Эмбеддинги Sonar
Выходные данные	Логиты над словарем	Эмбеддинги Sonar
Количество параметров	50 млн.	50 млн.

Модель nanoSonar была модифицирована для работы с непрерывными семантическими представлениями аналогично методам, применяемым исследовательским коллективом AIRI при подготовке модели SONAR-LLM [22].

Ключевым архитектурным отличием nanoSonar от обычного nanoGPT является использование дополнительных проекционных слоев и декодера Sonar. Входные данные для nanoSonar представлены в виде семантических эмбеддингов размерности 1024, которые проецируются в пространство скрытых состояний трансформера размерности 512 через линейный проектор (forward_proj).

После обработки трансформером скрытые состояния проецируются обратно в пространство Sonar embeddings (1024 измерения) через обратный проектор (reverse_proj), после чего передаются в замороженный декодер Sonar для декодирования выходной последовательности и вычисления функции потерь.

Архитектура модели nanoSonar представлен на рис. 6.

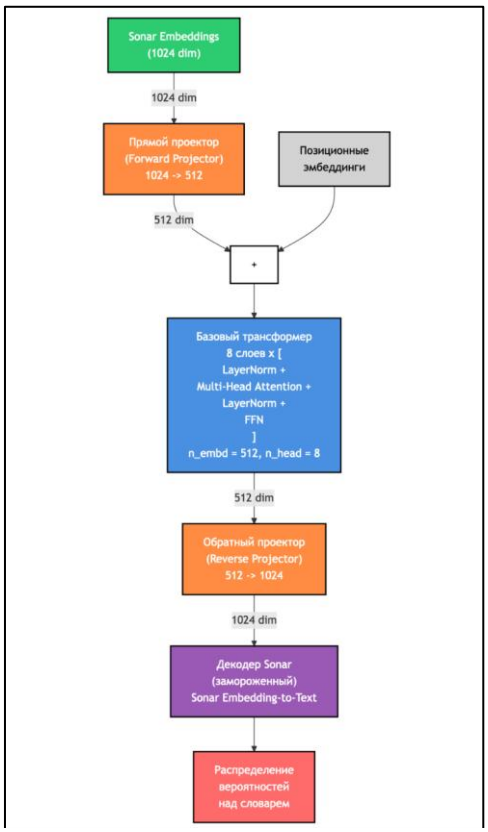

Рис. 6. Архитектура модели nanoSonar

Формирование обучающего набора данных. В качестве базового корпуса для экспериментального исследования использован открытый датасет TinyStories [23], представляющий собой коллекцию коротких детских рассказов, специально разработанную для обучения языковых моделей. Датасет содержит тексты, написанные простым языком с ограниченным словарным запасом, что делает его подходящим для сравнительного анализа различных архитектурных подходов при ограниченных вычислительных ресурсах.

Датасет TinyStories включает предопределенное разделение на обучающую и валидационную выборки. Обучающая выборка содержит тексты, используемые для оптимизации параметров модели, а валидационная выборка применяется для оценки обобщающей способности моделей и предотвращения переобучения.

Для модели nanoGPT данные предобрабатываются посредством токенизации с посредством BPE (BPE, Byte Pair Encoding) посредством применения открытой библиотеки tiktoken. Каждый текст преобразуется в последовательность целочисленных идентификаторов токенов, после чего к последовательности добавляется специальный токен конца текста (EOT token). Полученные последовательности токенов сохраняются в бинарном формате (uint16) для эффективной загрузки во время обучения.

Для модели nanoSonar применяется принципиально иной подход к представлению данных. Тексты предварительно разбиваются на предложения с использованием библиотеки NLTK (Natural Language Toolkit). Каждое предложение кодируется в семантическое

эмбеддинг размерности 1024 с помощью предобученной модели Sonar Text-to-Embedding. Последовательности предложений формируются с ограничением максимальной длины (max_sentences = 8) для оптимизации использования видеопамяти. Эмбеддинги сохраняются в формате PyTorch тензоров (.pt файлы) вместе с метаданными о длинах последовательностей и исходных текстах.

Оба подхода к предобработке данных обеспечивают идентичное содержание обучающих и валидационных выборок, различаясь лишь формой представления: дискретные токены для nanoGPT и непрерывные семантические эмбеддинги для nanoSonar.

Процедура обучения. Для минимизации сторонних факторов и обеспечения сопоставимости, процесс обучения для обеих моделей был унифицирован по ключевым гиперпараметрам, представленным в Таблице 3.

Таблица 3

Гиперпараметры обучения моделей nanoGPT и nanoSonar

Параметр	nanoGPT	nanoSonar
Оптимизатор	AdamW	AdamW
Скорость обучения (Learning Rate)	6×10^{-4}	5×10^{-4}
Расписание LR	Linear Warmup + Cosine Decay	Linear Warmup + Cosine Decay
Период разогрева (Warmup Steps)	200	200
Весовая регуляризация (Weight Decay)	0.1	0.01
Эффективный размер батча	64	64
Всего итераций	12000	12000

Обучение проводилось с использованием половинной точности (тип данных: float16).

Метрики оценки. Для комплексной оценки эффективности и обобщающей способности сравниваемых моделей был использован набор метрик, охватывающий стандартные критерии качества языкового моделирования.

Оценка проводилась каждые 500 итераций на выделенной валидационной выборке. Все эксперименты отслеживались с помощью платформы Weights & Biases (WandB) для обеспечения воспроизводимости.

Ниже приводится описание применяемых метрик.

Функция потерь (Loss). В качестве функции потерь использовалась стандартная кросс-энтропия (Categorical Cross-Entropy), которая непосредственно оптимизируется в процессе обучения. Для последовательности длиной Nтокенов функция потерь вычисляется согласно формуле 4.1.:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} \log\, P(w_i \mid w_{<i}),$$

где w_{i} – целевой (правильный) токен на позиции i; $w_{<i}$ – контекст, предыдущие токены последовательности $w_1, w_2, \ldots, w_{i-1}$; $P(w_i \mid w_{<i})$ – вероятность, назначенная моделью целевому токену w_i в данном контексте.

Для модели nanoGPT метрика loss (функция потерь) вычисляется напрямую по стандартной схеме языкового моделирования. Модель преобразует входную последовательность токенов через:

1. Слой эмбеддингов: $\mathrm{X} = \mathrm{Embedding}(w_{<i})$.
2. Трансформерные блоки: $\mathrm{H} = \mathrm{Transformer(X)}$.
3. Выходной слой: $\mathrm{z}_i = \mathrm{W}_o \mathrm{h}_i$.

где z_i – логитты (необработанные выходные данные с слоя) для позиции i, W_o – матрица выходного слоя.

Вероятность генерации токена с помощью функции Softmax согласно следующей формуле:

$$P(w_i \mid w_{<i}) = \text{Softmax}(\text{z}_i)_{w_i} = \frac{\exp(\text{z}_{i,w_i})}{\sum_{j=1}^{V} \exp(\text{z}_{i,j})},$$

где V – размер словаря.

В отличии от модели nanoGPT, где процесс расчета функции потерь является реализацией стандартного подхода к обучению NLP моделей посредством прямого расчета функции кросс-энтропии между входной обучабщей последовательностью и выходными данными, в модели nanoSonar используется двухшаговая процедура вычисления потерь:

1. Предсказание в пространстве эмбеддингов:

- входные эмбеддинги проходят через прямой проектор: $\text{X} = \text{W}_f \cdot \text{SonarEmbedding}(w_{<i})$;
- трансформерные блоки: $\text{H} = \text{Transformer}(\text{X})$;
- обратный проектор: $\text{e}_i = \text{W}_r \text{h}_i$;

где e_i – предсказанный эмбеддинг для позиции i, W_f и W_r – матрицы прямого и обратного проекторов.

2. Декодирование в пространство токенов:

Предсказанный эмбеддинг e_i передается в замороженный декодер Sonar, который преобразует его в распределение вероятностей выходного токена над словарем модели:

$$P(w_i \mid w_{<i}) = \text{Decoder}_{\text{Sonar}}(\text{e}_i)_{w_i},$$

где $\text{Decoder}_{\text{Sonar}}$ – замороженная предобученная модель, отображающая эмбеддинг обратно в пространство вероятностей токенов.

Таким образом, полная функция потерь для nanoSonar формализуется как:

$$\mathcal{L}_{\text{nanoSonar}} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\text{Decoder}_{\text{Sonar}}(\text{W}_r \cdot \text{Transformer}(\text{W}_f \cdot \text{SonarEmbedding}(w_{<i}))_i)_{w_i}\right).$$

Такой подход обеспечивает эквивалентность вычисляемой метрики, поскольку обе модели, в конечном счете, оптимизируют функцию кросс-энтропии между распределением над словарем и целевыми токенами, что делает их сравнение корректным, несмотря на разницу в архитектуре. Отслеживание потерь на обучающей и валидационной выборках позволяет анализировать сходимость моделей и выявлять переобучение.

Перплексия. Перплексия является стандартной и информативной метрикой для оценки языковых моделей. Она измеряет способность модели предсказывать следующий элемент последовательности. Метрика вычисляется как экспонента от средней кросс-энтропийной потери согласно формуле:

$$\text{Perplexity} = \exp(\mathcal{L}),$$

где $\mathcal{L}$ – средняя кросс-энтропийная потеря на валидационной выборке.

Важно, что несмотря на то, что перплексия не является прямой целевой функцией оптимизации и не используется при вычислении градиентов, она представляет значительную аналитическую ценность, так как, в интерпретации, перплексия показывает среднее количество вариантов, которые модель рассматривает при предсказании следующего токена, что говорит об общей точности языкового моделирования при помощи анализируемой модели.

Анализ результатов эксперимента. Данный раздел представляет сравнительный анализ моделей nanoGPT и nanoSonar по установленным метрикам.

Визуализация результатов обучения моделей. Для наглядного анализа динамики обучения и сравнения эффективности моделей использовались графики из платформы Weights & Biases (WandB).

На рис. 7-14 представлены кривые обучения по ключевым метрикам - функции потерь (loss) и перплексии (perplexity) на тренировочной и валидационной выборках.

По оси X представлены шаги обучения при логировании данных каждые 500 итераций.

С целью соблюдения масштабирования графиков, по оси X отсчет ведется с шага 2, то есть по прошествии первых 1000 итераций. Значения по оси Y приводятся в абсолютных значениях.

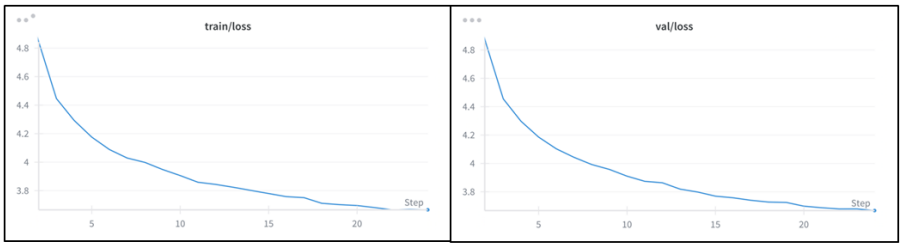

Рис. 7. Динамика функции потерь (loss) модели nanoGPT

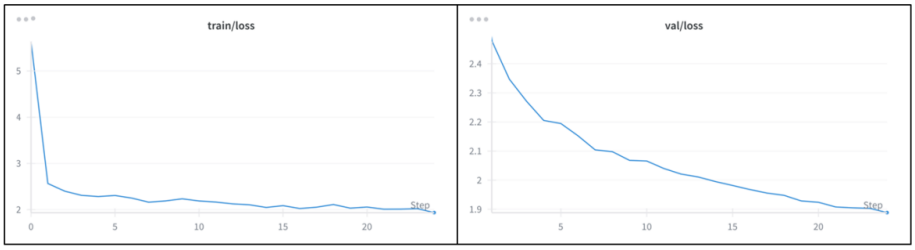

Рис. 8. Динамика перплексии модели nanoGPT

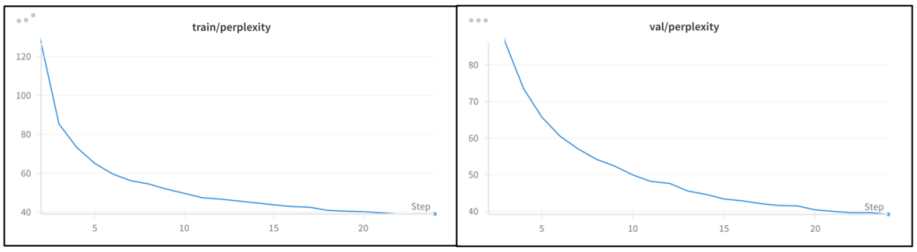

Рис. 9. Динамика функции потерь (loss) модели nanoSonar

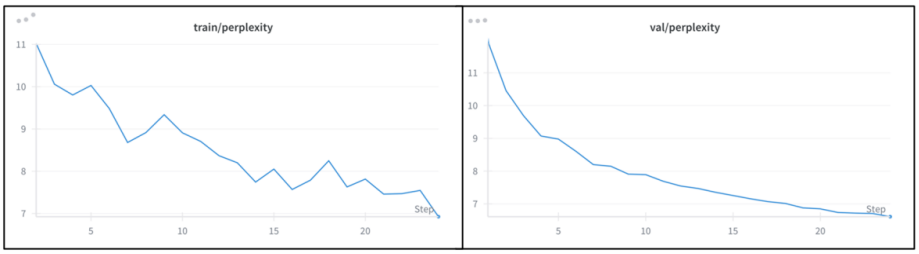

Рис. 10. Динамика перплексии модели nanoSonar

Динамика обучения и сходимости моделей. Анализ кривых обучения выявил существенные различия в динамике сходимости сравниваемых архитектур. Результаты значений функции потерь модели при обучении представлены в табл. 4.

Таблица 4

Значения функции потерь при обучении

№ итерации	nanoGPT	nanoSonar
2500	4,186	2,145
5000	3,910	2,066
7500	3,769	1,982
10000	3,699	1,924
12000	3,667	1,888

На рис. 11 представлена сравнительная динамика функции потерь моделей.

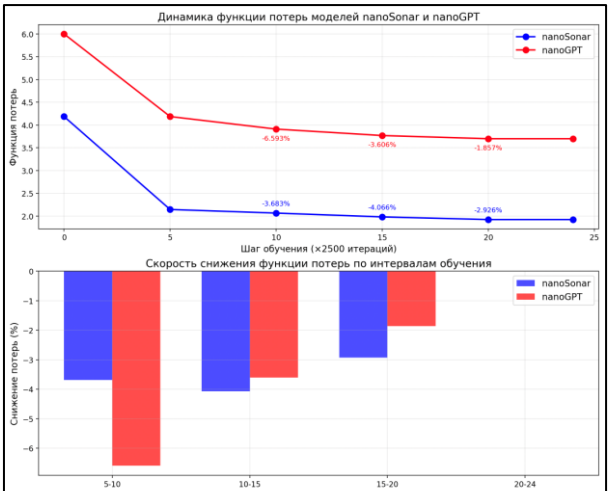

Рис. 11. Сравнительная динамика функции потерь моделей

Исходя из полученных значений, можно сделать вывод что модель nanoSonar изначально имеет более низкий базовый уровень функции потерь (2,145 у nanoSonar против 4,186 у nanoGPT при первых 2500 итераций). Значительно более низкий базовый уровень функции потерь у модели nanoSonar может быть объяснен тем, что в составе nanoSonar находится предобученная модель Sonar Decoder, реализующая функции преобразования эмбеддингов Sonar в текстовый вид. Такая модель содержит в себе базовые структурные принципы естественного языка, что и позволяет на ранних этапах обучения общей модели иметь значительно более низкий уровень функции потерь.

Не смотря на более низкий базовый уровень функции потерь, можно заметить, что в течение последующих 2500 итераций, скорость падения функции потерь у модели nanoGPT заметно превышает скорость падения функции потерь у модели nanoSonar (-6,593% у nanoGPT против -3,683% у nanoSonar). Такое явление может быть объяснено необходимостью наличия значительно большего количества обучающих примеров входных концептуальных эмбеддингов для выделения закономерностей.

Однако, в течение последующих 2500 итераций, скорость схождения модели nanoSonar начинает значительно опережать таковую у модели nanoGPT (-4,066% у nanoSonar против -3,606% у nanoGPT). Это подтверждает гипотезу о более эффективном усвоении языковых закономерностей при работе в семантическом пространстве эмбеддингов.

Такая закономерность усиливается и при дальнейшем обучении. В течение последующих 2500 итераций скорость падения функции потерь составляет -2,926% у nanoSonar против -1,857% у nanoGPT.

Разрыв между тренировочными и валидационными метриками функции потерь на последней итерации обучения составил:

- у nanoGPT: 0.001;
- у nanoSonar: 0.047.

Кривые потерь на обучающей и валидационной выборках для обеих моделей показали устойчивую сходимость без признаков переобучения.

Анализ эффективности использования данных. Важным аспектом исследования является оценка эффективности использования обучающих данных.

Для более глубокого анализа эффективности обучения был проведен детальный анализ динамики перплексии и количества усвоенной информации в битах. Перплексия, будучи экспонентой от функции потерь, предоставляет более интерпретируемую метрику качества языкового моделирования, непосредственно связанную с энтропийной оценкой неопределенности предсказаний модели.

Кросс-энтропия L имеет прямую информационно-теоретическую интерпретацию – она измеряет среднее количество бит, необходимое для кодирования одного токена при использовании распределения, предлагаемого моделью. Таким образом, перплексия может быть выражена через энтропию: $PPL = 2^{H(p)}$, где $H(p)$ – энтропия распределения модели.

Уменьшение перплексии с PPL_1 до PPL_2 соответствует уменьшению энтропии предсказаний модели. Количество усвоенной информации в битах рассчитывается как:

$$\Delta I = \log_2(PPL_1) - \log_2(PPL_2).$$

Эта величина показывает, насколько модель стала эффективнее в предсказании следующих токенов, измеряя сокращение неопределенности в битах на токен. Результаты значений функции перплексии моделей при обучении представлены в табл. 5.

Таблица 5

Значения функции перплексии при обучении

№ итерации	nanoGPT	nanoSonar
2500	65,733	8,977
5000	49,908	7,892
7500	43,348	7,255
10000	40,394	6,849
12000	39,151	6,609

На рис. 12 представлена динамика функции перплексии и усвоения информации моделями.

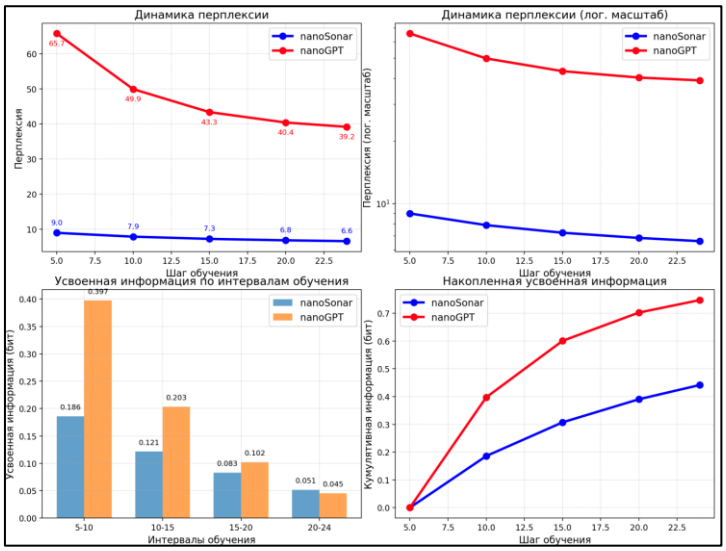

Рис. 12. Динамика функции перплексии и усвоения информации моделями

Модель nanoSonar демонстрирует принципиально более низкий базовый уровень перплексии уже на начальных этапах обучения: 8,977 против 65,733 у nanoGPT на шаге 5. Это преимущество объясняется наличием предобученного Sonar Decoder в архитектуре nanoSonar, который обеспечивает изначально более эффективное представление семантических паттернов.

Анализ динамики снижения энтропии показывает, что хотя абсолютные значения усвоенной информации в битах у nanoSonar ниже на ранних этапах (0,186 бит против 0,397 бит у nanoGPT на интервале шаг 5-10), модель демонстрирует более устойчивую динамику обучения. На завершающем этапе (шаг 20-24) nanoSonar усваивает 0,051 бит информации против 0,045 бит у nanoGPT, что свидетельствует о сохраняющейся эффективности обучения даже при достижении низких значений перплексии.

Критически важным наблюдением является тот факт, что скорость снижения энтропии у nanoSonar замедляется значительно меньше, чем у nanoGPT. Если на интервале шаг 5-10 разница в усвоенной информации составляла 0,211 бит в пользу nanoGPT, то к интервалу шаг 20-24 nanoSonar уже демонстрирует преимущество в 0,006 бит. Это подтверждает гипотезу о том, что работа в пространстве семантических эмбеддингов позволяет более эффективно усваивать сложные языковые закономерности на поздних этапах обучения.

Обсуждение результатов в контексте исследовательской гипотезы. Проведенное исследование демонстрирует несколько значимых закономерностей в контексте преодоления законов сложности масштабирования. Архитектура nanoSonar показала принципиально более эффективное использование данных на поздних этапах обучения, сохраняя способность к усвоению информации даже при достижении низких значений перплексии. Особенно важно отметить, что модель, работающая в пространстве семантических эмбеддингов, достигла качественно лучших конечных показателей (перплексия 6,609 против 39,151) при идентичных вычислительных ресурсах и объеме обучающих данных.

Эксперимент подтвердил работоспособность концепции разделения языкового моделирования на два уровня: семантический (работа с концептуальными эмбеддингами) и вербальный (генерация текста). Такой подход позволяет вынести ресурсоемкую часть – освоение базовых языковых структур – в предобученный модуль, что значительно сокращает требования к вычислительным ресурсам при дообучении модели на конкретных задачах.

Полученные результаты открывают перспективы для создания эффективных языковых моделей для специализированных применений, где доступ к вычислительным ресурсам ограничен. Архитектура типа nanoSonar особенно перспективна для:

- доменно-специфичных моделей, требующих тонкой настройки на ограниченных корпусах текстов;
- систем реального времени, где критически важна скорость инференса;
- приложений с ограниченными вычислительными ресурсами (мобильные устройства, edge-устройства).

Однако, следует отметить несколько существенных ограничений проведенного исследования:

1. Эксперимент проводился на относительно небольших моделях (50 млн параметров), что не позволяет экстраполировать результаты на модели промышленного масштаба
2. Использование упрощенного датасета TinyStories ограничивает обобщаемость выводов для сложных языковых задач
3. Не проводилось сравнение реального времени обучения и инференса, что является критически важным для практических применений
4. Отсутствует комплексная оценка качества генерации за пределами метрик перплексии, включая семантическую согласованность и логичность текстов

На основе выявленных ограничений и полученных результатов можно выделить следующие направления для будущих исследований:

1. Исследование поведения аналогичной архитектуры на моделях среднего и большого масштаба (100 млн – 1 млрд параметров);
2. Детальное изучение вычислительной эффективности подхода, включая время обучения, потребление памяти и скорость инференса;
3. Применение подхода к более сложным и разнообразным датасетам, включая многозадачные бенчмарки;
4. Исследование альтернативных подходов к проектированию проекционных слоев и интеграции с предобученными моделями;
5. Анализ способности моделей, обученных в семантическом пространстве, к переносу знаний между различными доменами и задачами.

Полученные результаты, несмотря на свои ограничения, указывают на перспективность дальнейшего исследования архитектур, работающих в пространстве семантических эмбеддингов, как одного из возможных путей преодоления фундаментальных ограничений современной парадигмы масштабирования языковых моделей.

Заключение. Поиск альтернативных подходов к масштабированию больших языковых моделей (LLM), позволяющих преодолеть системные ограничения парадигмы экстенсивного роста, представляет собой комплексную научно-инженерную задачу, требующую баланса между вычислительной эффективностью, качеством генерации и доступностью данных. Как демонстрируют проведенное исследование и сравнительный анализ, одним из ключевых путей решения является переход от работы с дискретными токенами к операциям в пространстве семантических эмбеддингов. Экспериментальное сравнение архитектур nanoGPT и nanoSonar подтвердило, что модель, оперирующая концептуальными представлениями, достигает качественно более низкой перплексии (6,609 против 39,151) при идентичных ресурсах, затраченных на обучение моделей, что свидетельствует о радикальном повышении информационной эффективности обучения и предлагает практический способ смягчения проблемы «Великого дефицита данных».

Однако реализованный подход не лишен компромиссов. Использование внешней предобученной модели (Sonar) для кодирования и декодирования эмбеддингов, хотя и обеспечивает мощное семантическое ядро, создает архитектурную зависимость и может ограничивать гибкость при работе с узкоспециальными или мультимодальными данными. Кроме того, необходимость проецирования эмбеддингов в пространство трансформера и обратно вводит дополнительные вычислительные слои, что потенциально может влиять на скорость инференса по сравнению с полностью токенными моделями, особенно при генерации очень длинных последовательностей. Тем не менее, полученный шестикратный выигрыш в перплексии доказывает, что эти компромиссы оправданы достигнутым качеством языкового моделирования.

Перспективным направлением для дальнейших исследований остается разработка методов совместной оптимизации или легкой адаптации семантического энкодера под конкретную предметную область, что позволит сохранить преимущества концептуального представления при работе со специализированной терминологией. Аналогично, интеграция архитектур, работающих с эмбеддингами, с альтернативными вычислительными парадигмами – такими как SSM для достижения линейной сложности вычислений или SNN для событийно-ориентированной обработки данных – способны привести к прорыву в энергоэффективности и скорости обработки последовательностей. Отдельного внимания заслуживает исследование возможности создания унифицированного кросс-лингвального и кросс-модального пространства эмбеддингов, что соответствует глобальному тренду развития мультимодальных моделей общего назначения.

Успешное применение подхода, основанного на семантических эмбеддингах, определяется не только выбором архитектуры, но и глубоким пониманием взаимосвязи между уровнем абстракции входных данных, вычислительной сложностью модели и ее способностью к обобщению. Достижение оптимального баланса между этими аспектами открывает путь к созданию нового поколения языковых моделей, эффективно утилизирующих данные при обучении и доступных для развертывания в условиях ограниченных ресурсов, а также архитектурно приближенных к принципам иерархического смыслообразования. Это соответствует стратегическим задачам как фундаментальных исследований в области искусственного интеллекта, стремящихся преодолеть «законы масштабирования», так и прикладных разработок, нацеленных на создание экономичных и специализированных NLP-решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Kaplan J. et al.* Scaling laws for neural language models // arxiv preprint arxiv:2001.08361. – 2020.
2. *Tihanyi N. et al.* Dynamic intelligence assessment: Benchmarking llms on the road to agi with a focus on model confidence // 2024 IEEE International Conference on Big Data (bigdata). – IEEE, 2024. – P. 3313-3321.
3. *Villalobos P. et al.* Will we run out of data? Limits of LLM scaling based on human-generated data // arxiv preprint arxiv:2211.04325. – 2022.
4. *Hooker S.* The hardware lottery // Communications of the ACM. – 2021. – Vol. 64, No. 12. – P. 58-65.
5. *Barrault L. et al.* Large concept models: Language modeling in a sentence representation space // arxiv preprint arxiv:2412.08821. – 2024.

6. *Shojaee P. et al.* The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity // arxiv preprint arxiv:2506.06941. – 2025.
7. *Zhou Y. et al.* Divergences between language models and human brains // Advances in neural information processing systems. – 2024. – Vol. 37. – P. 137999-138031.
8. *Pagliardini M. et al.* Faster causal attention over large sequences through sparse flash attention // arxiv preprint arxiv:2306.01160. – 2023.
9. *Du N. et al.* Glam: Efficient scaling of language models with mixture-of-experts // International conference on machine learning. – PMLR, 2022. – P. 5547-5569.
10. *Gou J. et al.* Knowledge distillation: A survey // International journal of computer vision. – 2021. – Vol. 129, No. 6. – P. 1789-1819.
11. *Mcdonald D., Papadopoulos R., Benningfield L*. Reducing llm hallucination using knowledge distillation: A case study with mistral large and mmlu benchmark // Authorea Preprints. – 2024.
12. *Han Z. et al.* Parameter-efficient fine-tuning for large models: A comprehensive survey // arxiv preprint arxiv:2403.14608. – 2024.
13. *Wang T. et al.* Dataset distillation // arxiv preprint arxiv:1811.10959. – 2018.
14. *Cazenavette G. et al.* Dataset distillation by matching training trajectories // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – P. 4750-4759.
15. *Aguilar G. et al.* Knowledge distillation from internal representations // Proceedings of the AAAI conference on artificial intelligence. – 2020. – Vol. 34, No. 05. – P. 7350-7357.
16. *Beltagy I., Peters M.E., Cohan A.* Longformer: The long-document transformer // arxiv preprint arxiv:2004.05150. – 2020.
17. *Xu F. et al.* Towards large reasoning models: A survey of reinforced reasoning with large language models // arxiv preprint arxiv:2501.09686. – 2025.
18. *Duquenne P.A., Schwenk H., Sagot B.* SONAR: sentence-level multimodal and language-agnostic representations // arxiv preprint arxiv:2308.11466. – 2023.
19. *Hamilton J.D.* State-space models // Handbook of econometrics. – 1994. – Vol. 4. – P. 3039-3080.
20. *Peng B. et al.* Rwkv: Reinventing rnns for the transformer era // arxiv preprint arxiv:2305.13048. – 2023.
21. *Ghosh-Dastidar S., Adeli H.* Spiking neural networks // International journal of neural systems. – 2009. – Vol. 19, No. 04. – P. 295-308.
22. *Dragunov N. et al.* SONAR-LLM: Autoregressive Transformer that Thinks in Sentence Embeddings and Speaks in Tokens // arxiv preprint arxiv:2508.05305. – 2025.
23. *Eldan R., Li Y.* Tinystories: How small can language models be and still speak coherent english? // arxiv preprint arxiv:2305.07759. – 2023.
24. *Ковалев В.В.* Алгоритм предварительной обработки изображений для снижения вероятности переобучения свёрточных нейронных сетей на нейронном ускорителе // Известия ЮФУ. Технические науки. – 2024. – № 5 (241). – С. 29-37.
25. *Ковалев В.В., Сергеев Н.Е.* Расширение признакового пространства в задаче поиска и распознавания малоразмерных объектов на изображениях // Известия ЮФУ. Технические науки. – 2024. – № 1 (237). – С. 267-276.
26. *Ковалев В.В., Сергеев Н.Е.* Реализация сверточных нейронных сетей на встраиваемых устройствах с ограниченным вычислительным ресурсом // Известия ЮФУ. Технические науки. – 2021. – № 6 (223). – С. 64-72.

REFERENCES

1. *Kaplan J. et al.* Scaling laws for neural language models, *arxiv preprint arxiv:2001.08361*, 2020.
2. *Tihanyi N. et al.* Dynamic intelligence assessment: Benchmarking llms on the road to agi with a focus on model confidence, *2024 IEEE International Conference on Big Data (bigdata)*. IEEE, 2024, pp. 3313-3321.
3. *Villalobos P. et al.* Will we run out of data? Limits of LLM scaling based on human-generated data, *arxiv preprint arxiv:2211.04325*, 2022.
4. *Hooker S.* The hardware lottery, *Communications of the ACM*, 2021, Vol. 64, No. 12, pp. 58-65.
5. *Barrault L. et al.* Large concept models: Language modeling in a sentence representation space, *arxiv preprint arxiv:2412.08821*, 2024.
6. *Shojaee P. et al.* The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, *arxiv preprint arxiv:2506.06941*, 2025.
7. *Zhou Y. et al.* Divergences between language models and human brains, *Advances in neural information processing systems*, 2024, Vol. 37, pp. 137999-138031.
8. *Pagliardini M. et al.* Faster causal attention over large sequences through sparse flash attention, *arxiv preprint arxiv:2306.01160*, 2023.

9. *Du N. et al.* Glam: Efficient scaling of language models with mixture-of-experts, *International conference on machine learning*. PMLR, 2022, pp. 5547-5569.
10. *Gou J. et al.* Knowledge distillation: A survey, *International journal of computer vision*, 2021, Vol. 129, No. 6, pp. 1789-1819.
11. *Mcdonald D., Papadopoulos R., Benningfield L.* Reducing llm hallucination using knowledge distillation: A case study with mistral large and mmlu benchmark, *Authorea Preprints*, 2024.
12. *Han Z. et al.* Parameter-efficient fine-tuning for large models: A comprehensive survey, *arxiv preprint arxiv:2403.14608*, 2024.
13. *Wang T. et al.* Dataset distillation, *arxiv preprint arxiv:1811.10959*, 2018.
14. *Cazenavette G. et al.* Dataset distillation by matching training trajectories, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4750-4759.
15. *Aguilar G. et al.* Knowledge distillation from internal representations, *Proceedings of the AAAI conference on artificial intelligence*, 2020, Vol. 34, No. 05, pp. 7350-7357.
16. *Beltagy I., Peters M.E., Cohan A.* Longformer: The long-document transformer, *arxiv preprint arxiv:2004.05150*, 2020.
17. *Xu F. et al.* Towards large reasoning models: A survey of reinforced reasoning with large language models, *arxiv preprint arxiv:2501.09686*, 2025.
18. *Duquenne P.A., Schwenk H., Sagot B.* SONAR: sentence-level multimodal and language-agnostic representations, *arxiv preprint arxiv:2308.11466*, 2023.
19. *Hamilton J.D.* State-space models, *Handbook of econometrics*, 1994, Vol. 4, pp. 3039-3080.
20. *Peng B. et al.* Rwkv: Reinventing rnns for the transformer era, *arxiv preprint arxiv:2305.13048*, 2023.
21. *Ghosh-Dastidar S., Adeli H.* Spiking neural networks, *International journal of neural systems*, 2009, Vol. 19, No. 04, pp. 295-308.
22. *Dragunov N. et al.* SONAR-LLM: Autoregressive Transformer that Thinks in Sentence Embeddings and Speaks in Tokens, *arxiv preprint arxiv:2508.05305*, 2025.
23. *Eldan R., Li Y.* Tinystories: How small can language models be and still speak coherent english?, *arxiv preprint arxiv:2305.07759*, 2023.
24. *Kovalev V.V.* Algoritm predvaritel'noy obrabotki izobrazheniy dlya snizheniya veroyatnosti pereobucheniya svertochnykh neyronnykh setey na neyronnom uskoritele [An algorithm for image preprocessing to reduce the probability of overfitting of convolutional neural networks on a neural accelerator], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2024, No. 5 (241), pp. 29-37.
25. *Kovalev V.V., Sergeev N.E.* Rasshirenie priznakovogo prostranstva v zadache poiska i raspoznavaniya malorazmernykh ob"ektov na izobrazheniyakh [Expansion of the feature space in the problem of searching and recognizing small-sized objects in images] *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2024, No. 1 (237), pp. 267-276.
26. *Kovalev V.V., Sergeev N.E.* Realizatsiya svertochnykh neyronnykh setey na vstraivaemykh ustroystvakh s ogranichennym vychislitel'nym resursom [Implementation of convolutional neural networks on embedded devices with limited computing resources], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2021, No. 6 (223), pp. 64-72.

Ралко Кирилл Игоревич – Южный федеральный университет; e-mail: kralko@sfedu.ru; г. Ростов-на-Дону, Россия; тел.: +79877880940; кафедра вычислительной техники; аспирант.

Сергеев Николай Евгеньевич – Южный федеральный университет; e-mail: nesergeev@sfedu.ru; г. Таганрог, Россия; тел.: +79281742585; кафедра вычислительной техники; д.т.н.; профессор.

Ralko Kirill Igorevich – Southern Federal University; e-mail: kralko@sfedu.ru; Rostov-on-Don, Russia; phone: +79877880940; the Department of Computer Engineering; graduate student.

Sergeev Nikolay Evgenievich – Southern Federal University; e-mail: nesergeev@sfedu.ru; Taganrog, Russia; phone: +79281742585; the Department of Computer Engineering; dr. of eng. sc.; professor.