МИНОБРНАУКИРОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра геометрии

# Повышение качества представления иерархических связей в генеративных языковых моделях методами гиперболической геометрии

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы

направление 02.03.01—Математика и компьютерные науки
код и наименование направления

механико-математического факультета
наименование факультета, института, колледжа

Шильникова Никиты Сергеевича
фамилия, имя, отчество

| | | |
|---|---|---|
| Научный руководитель | | |
| доцент, к.ф.-м.наук | | Ю.В. Шевцова |
| должность,уч.степень,уч. звание | подпись, дата | инициалы, фамилия |
| Зав.кафедрой, д.ф.-м. наук, доцент | | В.Б. Поплавский |
| должность,уч.степень,уч. звание | подпись, дата | инициалы, фамилия |

Саратов 2026

**Введение.** Актуальность темы исследования обусловлена тем, что современные методы обработки естественного языка базируются на гипотезе дистрибутивной семантики, согласно которой смысл элемента языка определяется его контекстом. С математической точки зрения, задача сводится к построению отображения (вложения) дискретного множества слов в непрерывное векторное пространство, где семантическая близость аппроксимируется метрической близостью[1].

Доминирующей архитектурой в этой области является Transformer[2], лежащий в основе больших языковых моделей (LLM), таких как GPT. Стандартные реализации архитектуры Transformer оперируют в плоском евклидовом пространстве $\mathbb{R}^n$.

Под евклидовым пространством $\mathbb{R}^n$ в данной работе понимается $n$-мерное вещественное векторное пространство, снабженное стандартным скалярным произведением

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i,$$

порожденной им нормой

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$$

и евклидовой метрикой

$$d_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|.$$

Слово «плоское» означает, что данное пространство имеет нулевую кривизну, а кратчайшие линии между точками являются прямыми.

Однако эмпирические данные и теоретические исследования показывают, что семантические графы естественного языка обладают иерархической структурой и свойством масштабно-инвариантности, характерным для пространств с отрицательной кривизной. Согласно теореме о вложении деревьев, представление иерархических структур в евклидовом пространстве с малым искажением может требовать размерности, растущей экспоненциально по глубине дерева, тогда как гиперболическое пространство позволяет

[1] Mikolov T., Sutskever I., Chen K. et al. Distributed representations of words and phrases and their compositionality // Advances in neural information processing systems. — 2013. — Vol. 26.

[2] Vaswani A., Shazeer N., Parmar N. et al. Attention is all you need // Advances in neural information processing systems. — 2017. — Vol. 30.

представлять такие структуры с ограниченным искажением в существенно меньшей размерности.

Объектом данного исследования выступают нейросетевые языковые модели на базе архитектуры Transformer, а предметом — математические методы геометрического представления семантических связей на римановых многообразиях.

Целью работы является разработка и теоретическое обоснование модификации архитектуры Transformer, в которой пространство признаков наделяется структурой риманова многообразия постоянной отрицательной кривизны. Для достижения поставленной цели в работе решаются следующие задачи: изучение математических основ токенизации и геометризации семантики; формализация аппарата гиперболической геометрии, в частности модели шара Пуанкаре как римановой модели гиперболического пространства, для адаптации векторных вложений; проектирование гибридной архитектуры языковой модели, а также подготовка математических и программных решений для проведения сравнительного эмпирического тестирования.

Научная новизна работы заключается в математическом обосновании и программной реализации гибридной архитектуры языковой модели, интегрирующей слой гиперболических вложений (на базе модели шара Пуанкаре) с евклидовым механизмом самовнимания. Реализован механизм гетерогенной оптимизации параметров сети (с использованием римановых и классических градиентных методов) и подготовлен инструментарий для эмпирического исследования эффективности сжатия размерности векторного пространства на специально подготовленном русскоязычном корпусе текстов.

Работа состоит из введения, трех глав, заключения и списка использованных источников. В первой главе формализуются математические основы дистрибутивной семантики и описывается классическая архитектура Transformer. Во второй главе рассматривается математический аппарат гиперболической геометрии, включая модель шара Пуанкаре, логарифмическое и экспоненциальное отображения, а также методы римановой оптимизации. Третья глава описывает практическую реализацию гибридной нейросетевой модели, алгоритмы обучения, а также методику и программную базу сравнительных вычислительных экспериментов.

**Основное содержание работы.** В основе современных методов обработки естественного языка и функционирования больших языковых моделей лежит задача построения вычислимой математической модели текста. Для формализации процесса генерации информации необходимо определить дискретные и непрерывные пространства, в которых оперирует модель.

**Определение 1.** Текстом $T$ называется упорядоченная конечная последовательность дискретных символов из фиксированного алфавита $S$:

$$T = (s_1, s_2, \ldots, s_L), \quad s_i \in S,$$

где $S$ — алфавит, $s_i$ — отдельный символ, а $L$ — длина текста в символах.

Множество всех конечных последовательностей символов алфавита $S$ обозначается через $S^*$:

$$S^* = \bigcup_{\ell=0}^{\infty} S^\ell,$$

где $S^\ell$ — множество всех последовательностей длины $\ell$, составленных из символов алфавита $S$.

Построение нетривиальных семантических отображений непосредственно на множестве $S^*$ затруднено из-за проблемы информационной разреженности. Для повышения плотности сигнала осуществляется переход к пространству словаря.

**Определение 2.** Словарем $\mathcal{V} = \{w_1, w_2, \ldots, w_N\}$ называется конечное множество токенов, то есть семантически устойчивых групп символов. Здесь $N = |\mathcal{V}|$ обозначает мощность словаря, то есть количество различных токенов. Пусть $S^L$ обозначает множество всех последовательностей длины $L$, составленных из символов алфавита $S$, а $\mathcal{V}^m$ — множество всех последовательностей длины $m$, составленных из токенов словаря $\mathcal{V}$. Число $L$ задает длину исходного текста в символах, а число $m$ — длину его токенизированного представления. Процедура токенизации задается отображением

$$\tau : S^L \to \mathcal{V}^m,$$

которое переводит исходный текст

$$T = (s_1, s_2, \ldots, s_L), \quad s_i \in S,$$

в последовательность токенов

$$\tau(T) = X = (x_1, x_2, \ldots, x_m), \quad x_j \in \mathcal{V}.$$

Здесь $T$ обозначает исходный текст, $X$ — его токенизированное представление, $s_i$ — отдельный символ, а $x_j$ — отдельный токен. Обычно $m < L$, поскольку один токен может соответствовать группе из нескольких символов.

Поскольку словарь $\mathcal{V}$ конечен и упорядочен, существует биекция между ним и множеством индексов $\{1, 2, \ldots, N\}$. Это позволяет рассматривать токенизированный текст $X$ как целочисленный вектор в дискретном пространстве $\{1, 2, \ldots, N\}^m$. Однако естественная метрика числового ряда не отражает семантической близости понятий, что требует перехода к непрерывному метрическому пространству[3].

**Определение 3.** Отображением вложения (Embedding) называется функция $\mathcal{E} : \mathcal{V} \to \mathbb{R}^d$, сопоставляющая каждому дискретному токену вектор в $d$-мерном евклидовом пространстве. Входной текст $X$ трансформируется в матрицу вложений:

$$E = (\vec{e}_1, \vec{e}_2, \ldots, \vec{e}_m), \quad \vec{e}_j \in \mathbb{R}^d.$$

Ключевым свойством полученного пространства $\mathbb{R}^d$ является кодирование семантических отношений через геометрию векторов, где степень смысловой близости слов аппроксимируется косинусной мерой угла между ними:

$$\mathrm{Sim}(\alpha, \beta) = \frac{\langle \vec{e}_\alpha, \vec{e}_\beta \rangle}{\|\vec{e}_\alpha\| \|\vec{e}_\beta\|}.$$

[3]Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. — 7-е изд. — М.: Физматлит, 2004. — 572 с.

Здесь $\langle \vec{e}_\alpha, \vec{e}_\beta \rangle$ обозначает стандартное скалярное произведение векторов $\vec{e}_\alpha$ и $\vec{e}_\beta$ в евклидовом пространстве:

$$\langle \vec{e}_\alpha, \vec{e}_\beta \rangle = \sum_{i=1}^{d} e_{\alpha i} e_{\beta i},$$

а $\|\vec{e}_\alpha\|$ и $\|\vec{e}_\beta\|$ — их евклидовы нормы. Данная величина является косинусом угла между векторами и принимает значения от $-1$ до 1: чем ближе значение к 1, тем меньше угол между векторами и тем выше их семантическая близость.

Поскольку множество векторов само по себе инвариантно к перестановкам, для фиксации порядка токенов в последовательности к каждому вектору вложения применяется гармоническое позиционное кодирование:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right),$$

где $pos$ — позиция токена, а $i$ — индекс координаты вектора. Данное преобразование кодирует каждую позицию уникальным векторным паттерном, позволяя модели учитывать относительные расстояния между элементами последовательности.

Ядром рассматриваемой архитектуры Transformer является механизм самовнимания, задача которого — контекстно-зависимое преобразование представлений токенов. Пусть после применения слоя вложений и позиционного кодирования входная последовательность представлена матрицей

$$H \in \mathbb{R}^{m \times d},$$

где $m$ — длина последовательности токенов, а $d$ — размерность векторного представления каждого токена. Каждая строка матрицы $H$ соответствует одному токену с учетом его позиции в последовательности.

В механизме самовнимания для каждого токена строятся три векторных представления: запрос (query), ключ (key) и значение (value). Запрос описывает, какую информацию данный токен ищет в других токенах; ключ описывает, какую информацию токен может предоставить для сопоставления;

значение содержит информацию, которая будет передана дальше при наличии высокого веса внимания.

Эти представления вычисляются с помощью линейных преобразований:

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V,$$

где $Q, K, V \in \mathbb{R}^{m \times d}$ — матрицы запросов, ключей и значений соответственно, а $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ — матрицы обучаемых весов. Под обучаемыми весами понимаются параметры нейронной сети, значения которых изменяются в процессе обучения с целью уменьшения ошибки модели. Эта ошибка формализуется с помощью функции потерь.

**Определение 4.** Функцией потерь называется скалярная функция $\mathcal{L}(\theta)$, численно оценивающая ошибку модели при решении задачи предсказания следующего токена. Здесь $\theta$ обозначает совокупность всех обучаемых параметров нейронной сети. Чем меньше значение функции потерь, тем точнее модель аппроксимирует распределение вероятностей естественного языка.

В задаче авторегрессионного языкового моделирования модель по контексту $(x_1, x_2, \ldots, x_t)$ предсказывает вероятность следующего токена $x_{t+1}$. Поэтому в качестве функции потерь обычно используется перекрестная энтропия:

$$\mathcal{L}(\theta) = -\frac{1}{m-1} \sum_{t=1}^{m-1} \log P_\theta(x_{t+1} \mid x_1, x_2, \ldots, x_t),$$

где $m$ — длина токенизированной последовательности, $P_\theta(x_{t+1} \mid x_1, x_2, \ldots, x_t)$ — вероятность, которую модель с параметрами $\theta$ присваивает правильному следующему токену $x_{t+1}$. Минимизация функции потерь приводит к увеличению вероятности генерации правильного продолжения текста.

**Определение 5.** Механизм самовнимания (Self-Attention) определяется как матричная операция

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V,$$

где $Q, K, V \in \mathbb{R}^{m \times d}$ — матрицы запросов, ключей и значений соответственно, $m$ — длина токенизированной последовательности, а $d$ — размерность векторного представления токена.

Здесь $K^\mathsf{T}$ обозначает транспонированную матрицу ключей $K$; если $K \in \mathbb{R}^{m \times d}$, то $K^\mathsf{T} \in \mathbb{R}^{d \times m}$, поэтому произведение $QK^\mathsf{T}$ имеет размерность $m \times m$.

Матрица

$$C = \frac{QK^\mathsf{T}}{\sqrt{d}} \in \mathbb{R}^{m \times m}$$

называется матрицей оценок внимания. Ее элемент $C_{ij}$ пропорционален скалярному произведению запроса $i$-го токена и ключа $j$-го токена:

$$C_{ij} = \frac{\langle q_i, k_j \rangle}{\sqrt{d}},$$

где $q_i$ — строка матрицы $Q$, соответствующая запросу $i$-го токена, а $k_j$ — строка матрицы $K$, соответствующая ключу $j$-го токена. Деление на $\sqrt{d}$ используется для нормировки масштаба скалярных произведений.

Функция softmax применяется построчно к матрице оценок внимания. Для вектора $\mathbf{z} = (z_1, z_2, \ldots, z_r)$ она определяется формулой

$$\operatorname{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{r} e^{z_j}}.$$

Результатом применения softmax является вектор неотрицательных чисел, сумма которых равна единице.

Таким образом, матрица

$$A = \operatorname{softmax}(C) = \operatorname{softmax}\left(\frac{QK^\mathsf{T}}{\sqrt{d}}\right)$$

является матрицей весов внимания. Элемент $A_{ij}$ показывает, насколько сильно представление $i$-го токена должно учитывать информацию, содержащуюся в $j$-м токене. После умножения матрицы весов внимания на матрицу значений получается новая матрица контекстуализированных представлений:

$$H' = AV \in \mathbb{R}^{m \times d}.$$

В генеративных моделях типа GPT используется маска, запрещающая токену учитывать последующие позиции последовательности. Поэтому на практике механизм самовнимания записывается в виде

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}} + M\right)V,$$

где $M$ — маскирующая матрица, элементы которой задаются следующим образом:

$$M_{ij} = \begin{cases} 0, & j \le i, \\ -\infty, & j > i. \end{cases}$$

Благодаря этому при вычислении представления $i$-го токена модель использует только текущий и предыдущие токены, что обеспечивает авторегрессионный характер генерации.

Дальнейшая нелинейная обработка векторов осуществляется через полносвязную нейронную сеть:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2,$$

где $\text{ReLU}(u) = \max(0, u)$ применяется покоординатно, $W_1, W_2$ — матрицы обучаемых весов, а $b_1, b_2$ — векторы смещений.

На выходе последнего блока Transformer получается матрица скрытых состояний

$$H_{\text{out}} \in \mathbb{R}^{m \times d}.$$

В задаче авторегрессионной генерации для предсказания следующего токена используется скрытое состояние последней позиции:

$$h_{\text{last}} \in \mathbb{R}^d.$$

Данный вектор проецируется в пространство логитов над словарем:

$$z = h_{\text{last}} W_{\text{head}} + b_{\text{head}}, \quad z \in \mathbb{R}^N,$$

где $N = |\mathcal{V}|$ — размер словаря, $W_{\text{head}} \in \mathbb{R}^{d \times N}$ — матрица выходной проекции, а $b_{\text{head}} \in \mathbb{R}^N$ — вектор смещений.

Компонента $z_i$ называется логитом и соответствует ненормированной оценке вероятности выбора токена $w_i \in \mathcal{V}$. Для получения вероятностного распределения по словарю к вектору логитов применяется функция softmax:

$$P(w_i \mid \text{context}) = \frac{e^{z_i/\tau_{\text{temp}}}}{\sum_{j=1}^{N} e^{z_j/\tau_{\text{temp}}}},$$

где $\tau_{\text{temp}} > 0$ — гиперпараметр, называемый температурой. При меньших значениях температуры распределение становится более концентрированным около наиболее вероятных токенов, а при больших — более сглаженным.

Процесс синтеза текста представляет собой авторегрессионную процедуру последовательного выбора токенов. На каждом шаге модель вычисляет распределение вероятностей следующего токена с учетом уже сгенерированного контекста, после чего выбранный токен добавляется к последовательности и используется при дальнейшем предсказании. Таким образом, языковая модель может быть представлена как параметрическая функция

$$F_\theta : \{1, 2, \ldots, N\}^m \to \mathbb{R}^N,$$

где $m$ — длина токенизированной последовательности, $N = |\mathcal{V}|$ — размер словаря, $\{1, 2, \ldots, N\}^m$ — множество всех последовательностей длины $m$, составленных из индексов токенов, а $\mathbb{R}^N$ — пространство логитов над словарем.

При переходе от евклидовой геометрии к гиперболической следует отметить, что естественный язык содержит множество скрытых иерархических отношений, таких как таксономии, отношения подчинения (часть-целое) и синтаксические деревья. Стандартные векторные пространства $\mathbb{R}^n$ математически неэффективны для вложения подобных структур с малым искажением.

**Теорема 6** (О вложении деревьев[4])**.** Для вложения дерева глубины $h$ с коэффициентом ветвления $b$ в евклидово пространство с малым искажением может требоваться размерность, растущая экспоненциально по глубине дерева. В то же время гиперболическое пространство позволяет представлять

древовидные структуры с ограниченным искажением в существенно меньшей размерности.

Данное свойство обуславливает математическую целесообразность использования гиперболических представлений: семантическая иерархия может быть закодирована в пространстве существенно меньшей размерности без существенной потери информационной емкости. Гиперболическое пространство может быть задано различными эквивалентными моделями; в данной работе используется модель шара Пуанкаре, широко применяемая в задачах обучения иерархических представлений[5]. Она представляет собой открытый единичный шар в евклидовом пространстве, снабженный римановой метрикой постоянной отрицательной кривизны. Таким образом, точки модели имеют обычные координаты в $\mathbb{R}^n$, однако расстояния между ними вычисляются не по евклидовой, а по гиперболической метрике. В дальнейшем рассматривается случай параметра кривизны $c = 1$, что соответствует постоянной секционной кривизне $-1$.

**Определение 7.** Моделью шара Пуанкаре размерности $n$ называется риманово многообразие $(\mathbb{B}^n, g_{\mathbb{B}})$, где

$$\mathbb{B}^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| < 1\}$$

— открытый единичный шар, а $g_{\mathbb{B}}$ — риманова метрика постоянной отрицательной кривизны. Расстояние между точками $\mathbf{u}, \mathbf{v} \in \mathbb{B}^n$ задается формулой:

$$d_{\mathbb{B}}(\mathbf{u}, \mathbf{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\mathbf{u} - \mathbf{v}\|^2}{(1 - \|\mathbf{u}\|^2)(1 - \|\mathbf{v}\|^2)}\right).$$

Ключевым отличием данного многообразия является экспоненциальный рост объема шара по мере увеличения радиуса, в отличие от полиномиального роста в евклидовой геометрии. Это позволяет естественным образом размещать узлы иерархических графов: корневые понятия располагаются вблизи центра, а дочерние элементы — экспоненциально плотнее к границе шара.

[4]Sarkar R. Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane // Graph Drawing. GD 2011. Lecture Notes in Computer Science. Vol. 7034. — Berlin, Heidelberg: Springer, 2011. — P. 355–366.

[5]Nickel M., Kiela D. Poincaré embeddings for learning hierarchical representations // Advances in neural information processing systems. — 2017. — Vol. 30.

Поскольку стандартные линейные операции могут выводить векторы за пределы шара $\mathbb{B}^n$, в гиперболическом пространстве они заменяются специальными алгебраическими преобразованиями, сохраняющими геометрическую структуру многообразия.

**Определение 8.** Сложение Мёбиуса для векторов $\mathbf{u}, \mathbf{v} \in \mathbb{B}^n$ определяется выражением:

$$\mathbf{u} \oplus \mathbf{v} = \frac{(1 + 2\langle \mathbf{u}, \mathbf{v} \rangle + \|\mathbf{v}\|^2)\mathbf{u} + (1 - \|\mathbf{u}\|^2)\mathbf{v}}{1 + 2\langle \mathbf{u}, \mathbf{v} \rangle + \|\mathbf{u}\|^2\|\mathbf{v}\|^2}$$

**Определение 9.** Умножение Мёбиуса на скаляр $r \in \mathbb{R}$ задается как:

$$r \otimes \mathbf{v} = \tanh\left(r \cdot \operatorname{arctanh}(\|\mathbf{v}\|)\right) \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

Интеграция гиперболического слоя со стандартными механизмами трансформера требует реализации канонических взаимно обратных отображений между искривленным многообразием и плоским пространством. Геометрические переходы осуществляются через касательное пространство в нуле $T_{\mathbf{0}}\mathbb{B}^n$, которое изоморфно евклидову пространству $\mathbb{R}^n$[6].

**Определение 10.** Экспоненциальное отображение $\exp_{\mathbf{0}} : T_{\mathbf{0}}\mathbb{B}^n \to \mathbb{B}^n$ проецирует вектор из касательного пространства в нуле на гиперболическое многообразие:

$$\exp_{\mathbf{0}}(\mathbf{v}) = \tanh(\|\mathbf{v}\|)\frac{\mathbf{v}}{\|\mathbf{v}\|}.$$

**Определение 11.** Логарифмическое отображение $\log_{\mathbf{0}} : \mathbb{B}^n \to T_{\mathbf{0}}\mathbb{B}^n$ является обратной операцией, проецирующей точку многообразия обратно в касательное пространство:

$$\log_{\mathbf{0}}(\mathbf{y}) = \operatorname{arctanh}(\|\mathbf{y}\|)\frac{\mathbf{y}}{\|\mathbf{y}\|}.$$

Во всех приведенных формулах выражения, содержащие деление вектора на его норму, в случае нулевого вектора определяются по непрерывности и принимаются равными нулевому вектору.

[6] Дубровин Б. А., Новиков С. П., Фоменко А. Т. Современная геометрия: Методы и приложения. — 2-е изд., перераб. — М.: Наука, 1986. — 760 с.

Наличие постоянной отрицательной кривизны требует модификации алгоритмов оптимизации языковой модели. Стандартный метод стохастического градиентного спуска (SGD) неприменим напрямую, так как линейное обновление весов нарушает геометрию шара. Оптимизация параметров гиперболического слоя осуществляется с помощью риманова стохастического градиентного спуска (RSGD)[7].

Пусть $\mathcal{M}$ — риманово многообразие, $\mathcal{L} : \mathcal{M} \to \mathbb{R}$ — функция потерь, а $\mathbf{w}_t \in \mathcal{M}$ — текущее значение оптимизируемого параметра на шаге $t$.

В евклидовом пространстве направление наискорейшего убывания функции задается обычным градиентом. На римановом многообразии аналогичную роль выполняет риманов градиент $\operatorname{grad}_R \mathcal{L}(\mathbf{w}_t)$, который является вектором из касательного пространства $T_{\mathbf{w}_t}\mathcal{M}$. Он определяется как такой касательный вектор, который учитывает риманову метрику многообразия. В локальных координатах риманов градиент может быть записан через евклидов градиент следующим образом:

$$\operatorname{grad}_R \mathcal{L}(\mathbf{w}) = G(\mathbf{w})^{-1} \nabla_E \mathcal{L}(\mathbf{w}),$$

где $\nabla_E \mathcal{L}(\mathbf{w})$ — евклидов градиент функции потерь, а $G(\mathbf{w})$ — матрица метрического тензора в точке $\mathbf{w}$.

Обновление параметра выполняется не обычным сложением, а перемещением вдоль геодезической линии многообразия:

$$\mathbf{w}_{t+1} = \exp_{\mathbf{w}_t} \left( -\eta \operatorname{grad}_R \mathcal{L}(\mathbf{w}_t) \right),$$

где $\eta > 0$ — шаг обучения, $\mathbf{w}_t$ — значение параметра на текущей итерации, $\mathbf{w}_{t+1}$ — значение параметра после обновления, а $\exp_{\mathbf{w}_t}$ — экспоненциальное отображение из касательного пространства $T_{\mathbf{w}_t}\mathcal{M}$ на многообразие $\mathcal{M}$.

Геодезической называется кривая на многообразии, которая локально является кратчайшим путем между близкими точками. В евклидовом пространстве геодезическими являются прямые линии, а в шаре Пуанкаре — дуги окружностей, ортогональные границе шара, либо диаметры шара.

[7] Bonnabel S. Stochastic gradient descent on Riemannian manifolds // IEEE Transactions on Automatic Control. — 2013. — Vol. 58, no. 9. — P. 2217–2229.

В качестве практической реализации предложенных теоретических концепций спроектирована модификация архитектуры генеративной модели (на базе открытой реализации `nanoGPT` и языка программирования `Python`). Ключевым архитектурным изменением является замена стандартного евклидова слоя вложений на гиперболический.

Пусть $N = |\mathcal{V}|$ — размер словаря, а $D$ — размерность скрытого пространства модели. Гиперболический слой вложений задается набором обучаемых параметров

$$\Theta = (\theta_1, \theta_2, \ldots, \theta_N), \quad \theta_j \in \mathbb{B}^D,$$

где каждому токену $w_j \in \mathcal{V}$ соответствует вектор $\theta_j$, лежащий внутри $D$-мерного шара Пуанкаре.

Эквивалентно данный слой можно рассматривать как отображение

$$\mathcal{E}_{\mathbb{B}} : \mathcal{V} \to \mathbb{B}^D,$$

которое каждому токену словаря сопоставляет его гиперболическое векторное представление.

Пусть токенизированная входная последовательность имеет вид

$$X = (x_1, x_2, \ldots, x_m), \quad x_t \in \mathcal{V},$$

где $m$ — длина последовательности токенов. Процесс формирования входного векторного представления для $t$-й позиции описывается следующими этапами:

1. Извлечение гиперболического вектора токена:

$$e_t = \mathcal{E}_{\mathbb{B}}(x_t) \in \mathbb{B}^D.$$

2. Проекция гиперболического вектора в касательное евклидово пространство в нуле с помощью логарифмического отображения:

$$v_t = \log_{\mathbf{0}}(e_t), \quad v_t \in T_{\mathbf{0}}\mathbb{B}^D \simeq \mathbb{R}^D.$$

3. Интеграция обучаемого евклидова позиционного кодирования:

$$h_t = v_t + p_t, \quad p_t \in \mathbb{R}^D, \quad h_t \in \mathbb{R}^D.$$

Таким образом, гиперболическая геометрия используется на уровне слоя вложений, а после применения логарифмического отображения дальнейшая обработка представления $h_t$ механизмами самовнимания (Attention) и полносвязными слоями (FFN) выполняется в стандартном евклидовом пространстве. Это позволяет сохранить вычислительную эффективность матричных операций и использовать оптимизированную кодовую базу Transformer.

Программная реализация гиперболических операций базируется на библиотеке `geoopt`. Важной особенностью разработанного пайплайна является использование алгоритма гетерогенной оптимизации: для гиперболического слоя вложений применяется Riemannian Adam, учитывающий геометрию многообразия при обновлении параметров, а для евклидовых параметров сети — классический AdamW.

Для эмпирической оценки эффективности предложенной архитектуры подготовлены два сценария экспериментального исследования. В качестве эталонного (базового) варианта используется набор данных `Tiny Shakespeare`. В свою очередь, основной сценарий предполагает использование специально подготовленного русскоязычного корпуса текстов, включающего данные из открытого набора `google/wiki40b`, а также 30 000 синтетически сгенерированных статей для адаптации стилистики и формы изложения. Генерация синтетических образцов проводилась с помощью модели с открытыми весами `Qwen 3 8B`. Дискретизация обучающей выборки в обоих сценариях осуществляется с применением классического алгоритма токенизации Byte Pair Encoding (BPE)[8]. Разработан инструментарий сравнения евклидовой (Baseline GPT-Euclid) и гиперболической (Hyperbolic GPT) моделей при различных размерностях вложений. В качестве целевой метрики выбран показатель перплексии (Perplexity, PPL), расчет которого реализован для валидационной выборки.

[8]Sennrich R., Haddow B., Birch A. Neural machine translation of rare words with subword units // arXiv preprint arXiv:1508.07909. — 2015.

**Заключение.** В ходе выполнения работы проведено математическое обоснование и проектирование гибридной архитектуры генеративной языковой модели. На основе анализа теоретического материала и разработанных программных решений сформулированы следующие основные результаты:

1. Теоретически обосновано применение гиперболической геометрии. Показано, что использование римановых многообразий постоянной отрицательной кривизны (модели шара Пуанкаре) позволяет эффективно обходить ограничения плоских евклидовых пространств при вложении иерархических семантических структур естественного языка.
2. Формализован математический аппарат переходов. Описаны механизмы взаимодействия между гиперболическим слоем вложений и классическим евклидовым механизмом самовнимания посредством применения логарифмических и экспоненциальных отображений.
3. Спроектированы архитектура и алгоритм оптимизации. Предложена концепция гибридной нейросетевой архитектуры и разработан алгоритм гетерогенной оптимизации параметров сети, базирующийся на методах риманова градиентного спуска.
4. Разработана программная реализация. Подготовлен инструментарий для эмпирической оценки моделей и сравнения базовой евклидовой (Baseline GPT-Euclid) и предложенной гиперболической (Hyperbolic GPT) архитектур.
5. Сформирована база для вычислительных экспериментов. Подготовлены обучающие выборки для двух экспериментальных сценариев, включая сборку, аугментацию и BPE-токенизацию собственного русскоязычного корпуса текстов.

Цель работы достигнута: сформирован математический и программный фундамент для интеграции римановой геометрии в архитектуру Transformer. Подготовленная база открывает перспективы для проведения полномасштабных эмпирических экспериментов с целью подтверждения гипотезы о сохранении качества генерации (метрика Perplexity) при существенном сжатии размерности пространства признаков.