Санкт-Петербургская школа физико-математических и компьютерных наук

НИУ ВШЭ - Санкт-Петербург

Санкт-Петербург
2024

Разработка методов долгосрочной памяти для больших языковых моделей при помощи методов обучения с подкреплением

Выполнила: Белова Ю. Д.
Научный руководитель: Кривцов А. М.
Консультант: Свидченко О. А.

Механизм внимания в архитектуре трансформера

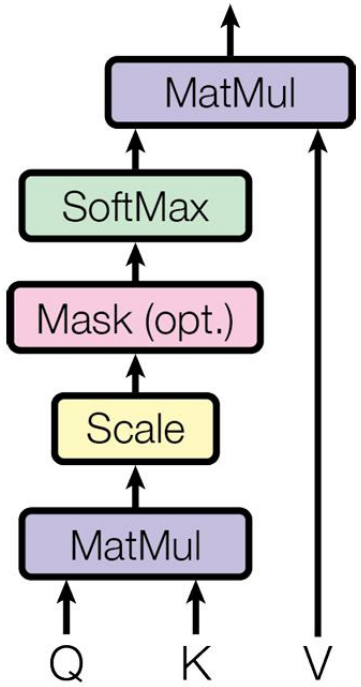

Рисунок 1. Механизм внимания в трансформере[1]

Ограничения классического трансформера

1. Квадратичная сложность операции внимания.

Если увеличили размер входа в 10 раз, то вычислительных ресурсов и памяти на обработку текста потребуется в 100 раз больше.

2. Ограничение на максимальную длину входа.

Языковой модели сложно адаптироваться к новой длине входа, на которой она еще не обучалась.

[1] Ashish, Vaswani., Noam, Shazeer., Niki, Parmar., Jakob, Uszkoreit., Llion, Jones., Aidan, N., Gomez., Lukasz, Kaiser., Illia, Polosukhin. (2017). Attention is All you Need.

Как обрабатывать длинные последовательности?

1. Эффективные трансформеры

Цель: оптимизировать вычислительную сложность механизма внимания вплоть до линейной.
Наиболее популярные работы: Longformer[1], Reformer[2], Big Bird[3], Ring Attention[4].

2. Рекуррентная память

Цель: создать механизм памяти, с помощью которого осуществляется передача информации между сегментами, на которые разделяется входная последовательность.

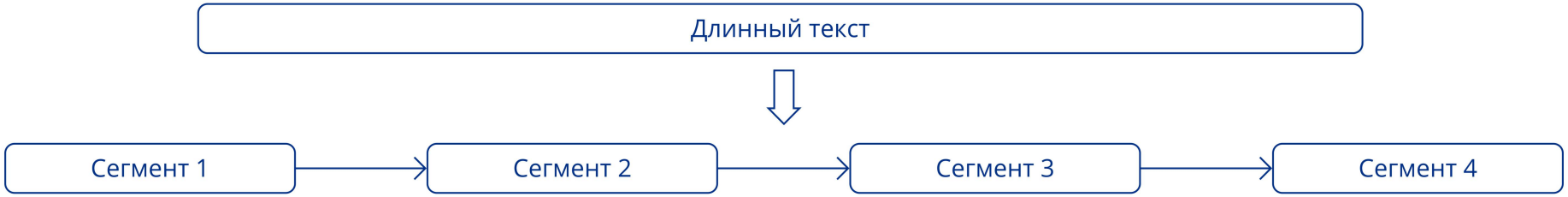

Рисунок 2. Разбиение текста на сегменты при обработке длинных последовательностей

[1] Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." (2020).
[2] Kitaev, Nikita, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer." (2020).
[3] Zaheer, Manzil, et al. "Big bird: Transformers for longer sequences." (2020).
[4] Liu, Hao, Matei Zaharia, and Pieter Abbeel. "Ring attention with blockwise transformers for near-infinite context." (2023).

Подходы к рекуррентной памяти языковых моделей

- Работы: Transformer XL[1], Compressive Transformers[2], Memorizing Transformers[3]
- *В качестве памяти - скрытые состояния или матрицы ключей и значений.*

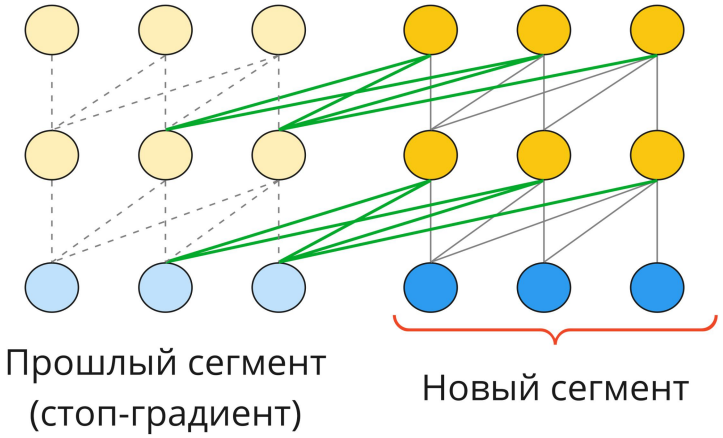

Рисунок 3. Transformer XL

- Работы: Recurrent Memory Transformer[4], AutoCompressors[5]
- *В качестве памяти - специальные токены.*

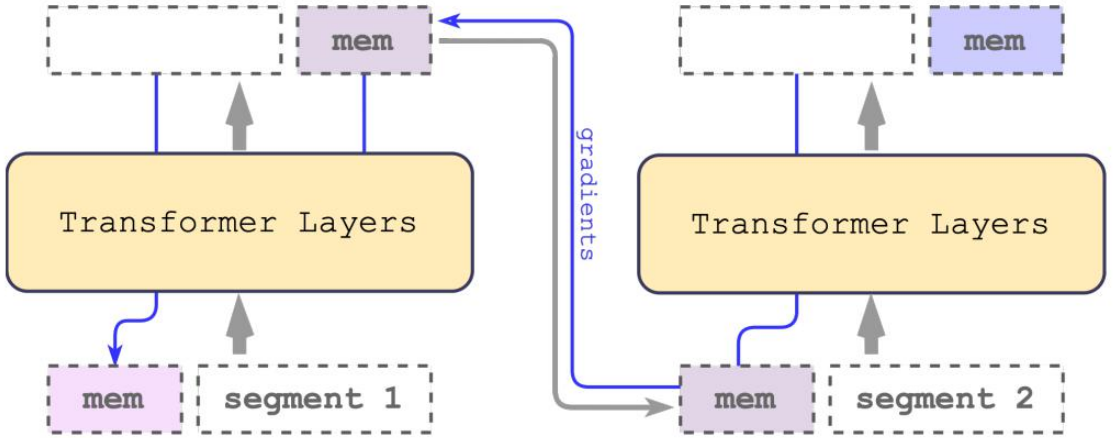

Рисунок 4. Recurrent Memory Transformer

[1] Dai, Zihang, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." (2019).
[2] Rae, Jack W., et al. "Compressive transformers for long-range sequence modelling." (2019).
[3] Wu, Yuhuai, et al. "Memorizing transformers." (2022).
[4] Bulatov, Aydar, Yury Kuratov, and Mikhail Burtsev. "Recurrent memory transformer." (2022)
[5] Chevalier, Alexis, et al. "Adapting language models to compress contexts." (2023).

Цель исследования и поставленные задачи

Цель исследования: разработать и валидировать метод долгосрочной памяти в больших языковых моделях с применением обучения с подкреплением.

Поставленные задачи:

1. Создать датасет, содержащий достаточно длинные тексты.
2. Разработать подход к внедрению долгосрочной памяти в большие языковые модели.
3. Реализовать предложенный метод, провести его обучение и настройку гиперпараметров.
4. Сравнить полученные результаты с выбранным бейзлайном.

Сбор русскоязычного датасета с длинными текстами

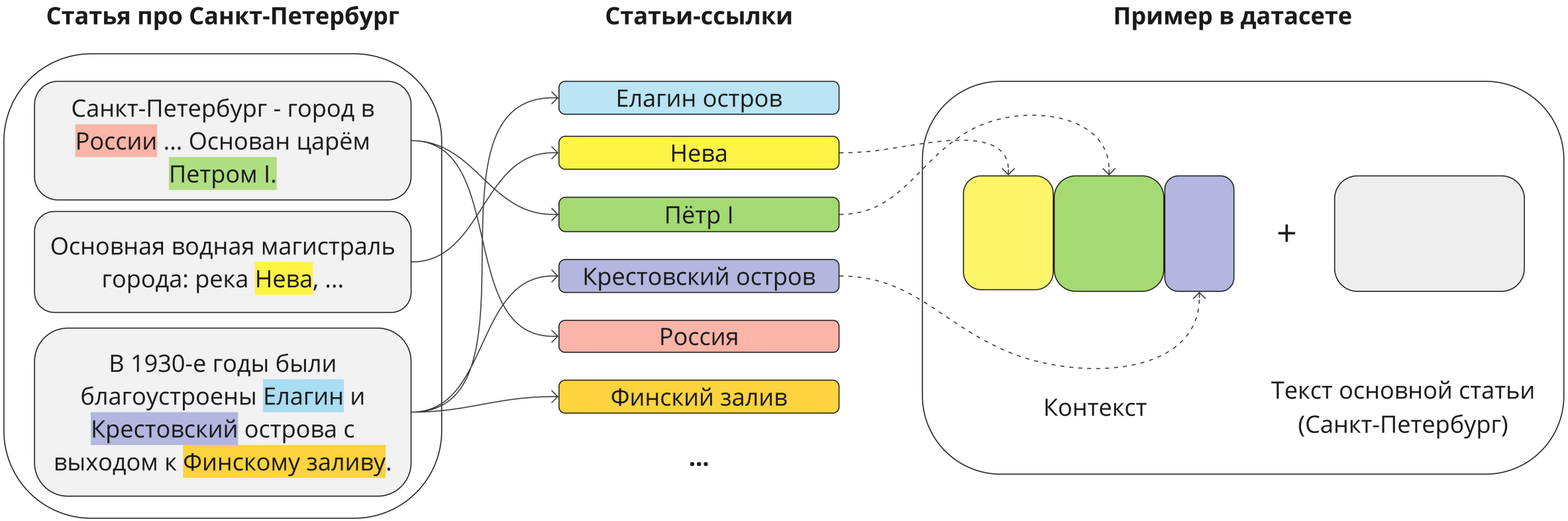

Рисунок 5. Схема формирования элемента датасета

Схема предложенного подхода

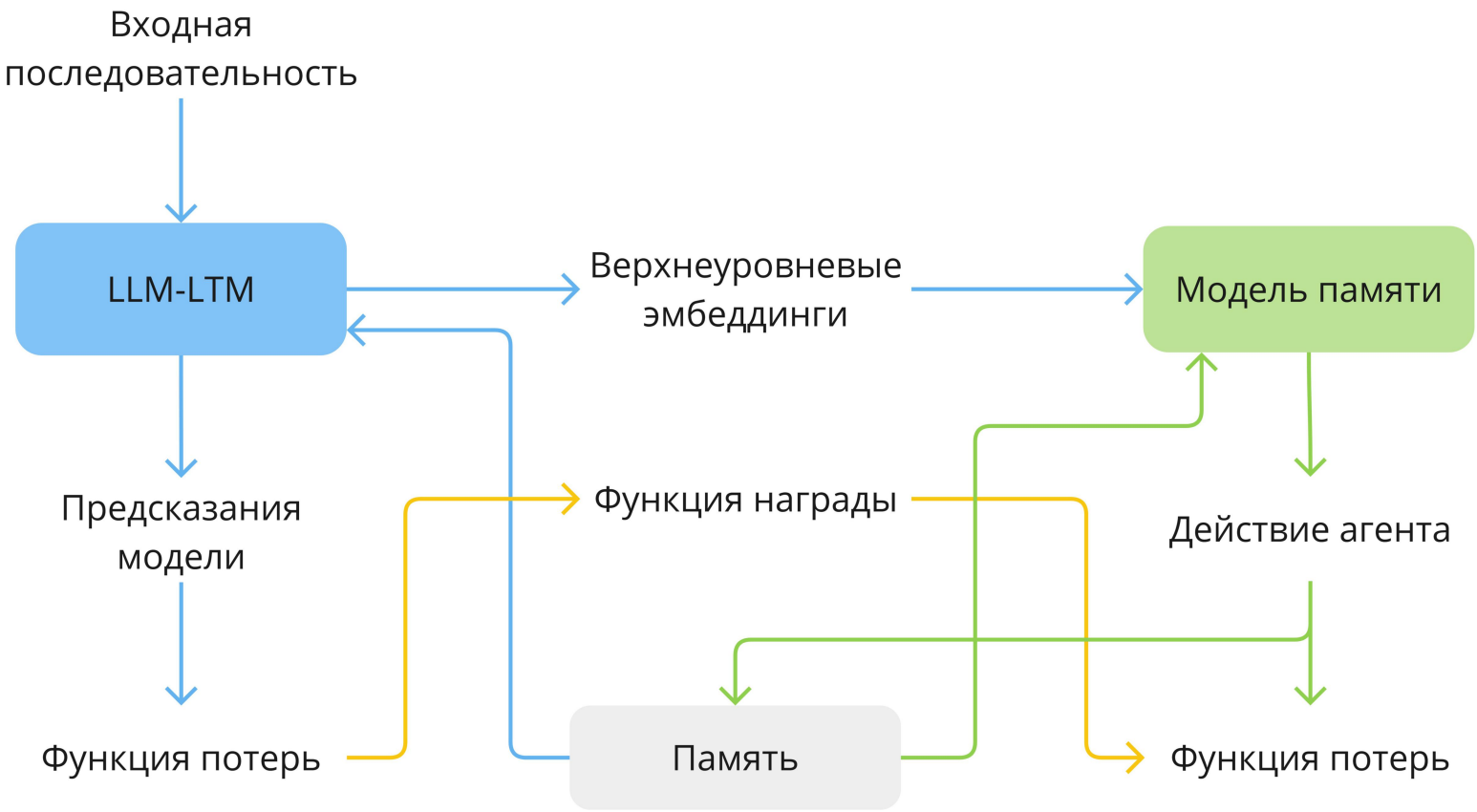

Рисунок 6. Взаимодействие агента (Модель памяти) и дообучаемой языковой модели (LLM-LTM)

Характеристики и параметры обучения

- Предобученная языковая модель, которая лежит в основе LLM-LTM - GPT3small[1].
- Для обучения агента был выбран алгоритм REINFORCE с модификациями.
- Награда агента на сегменте s_t определяется следующим образом:

$$R_t = -\frac{1}{k}\sum_{i=1}^{k}\mathcal{L}_{LLM-LTM}\left(s_{t+1}\left[0 : l_i\right]\right)$$

$$l_i \in [1,\ len(s_{t+1})]$$

где:

- t - номер сегмента, или шаг в эпизоде
- $\mathcal{L}_{\text{LLM-LTM}}$ - функция потерь LLM-LTM
- l_i - длина префикса сегмента, выбираемая случайно

- Каждый пример разбивается на сегменты размером 256 токенов.
- Размер памяти - 10 векторов размерностью 64.
- Оптимизатор - AdamW, скорость обучения - 3e-5.

[1] https://huggingface.co/ai-forever/rugpt3small_based_on_gpt2

Результаты обучения модели

В качестве бейзлайна выступает GPT3, выбранная в качестве основы для LLM-LTM, дообученнная на тренировочной выборке с помощью LoRA.

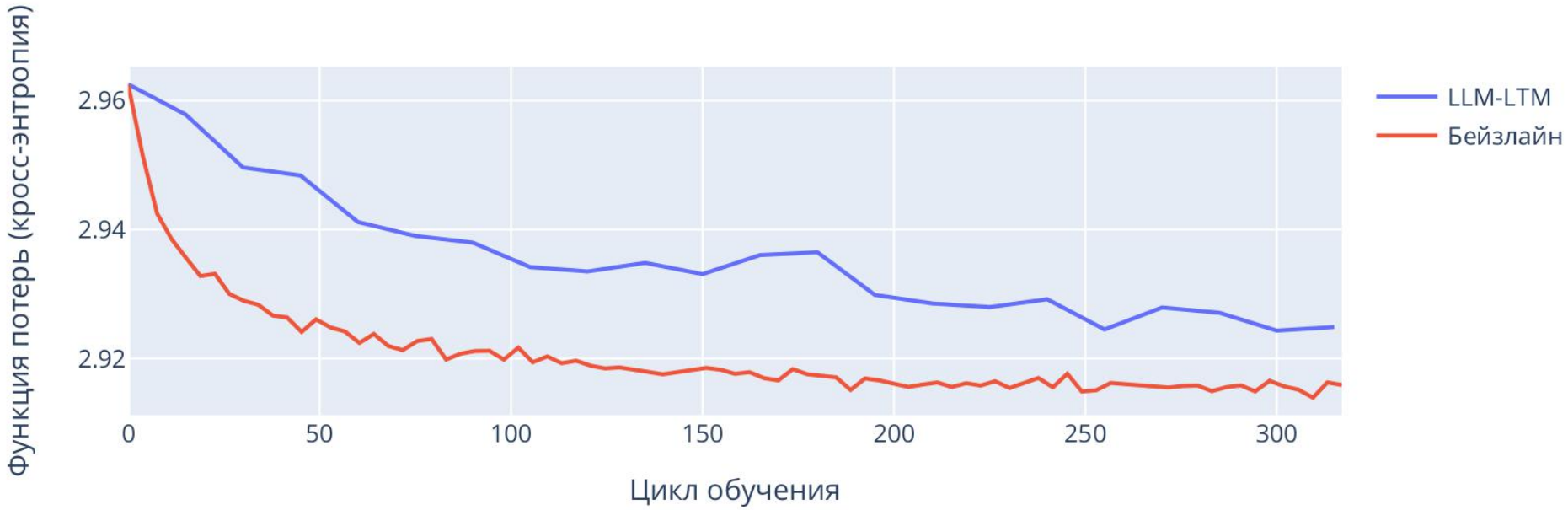

Рисунок 7. Функция потерь бейзлайна и LLM-LTM на валидационной выборке

Гипотеза 1: агенту требуется предобучение

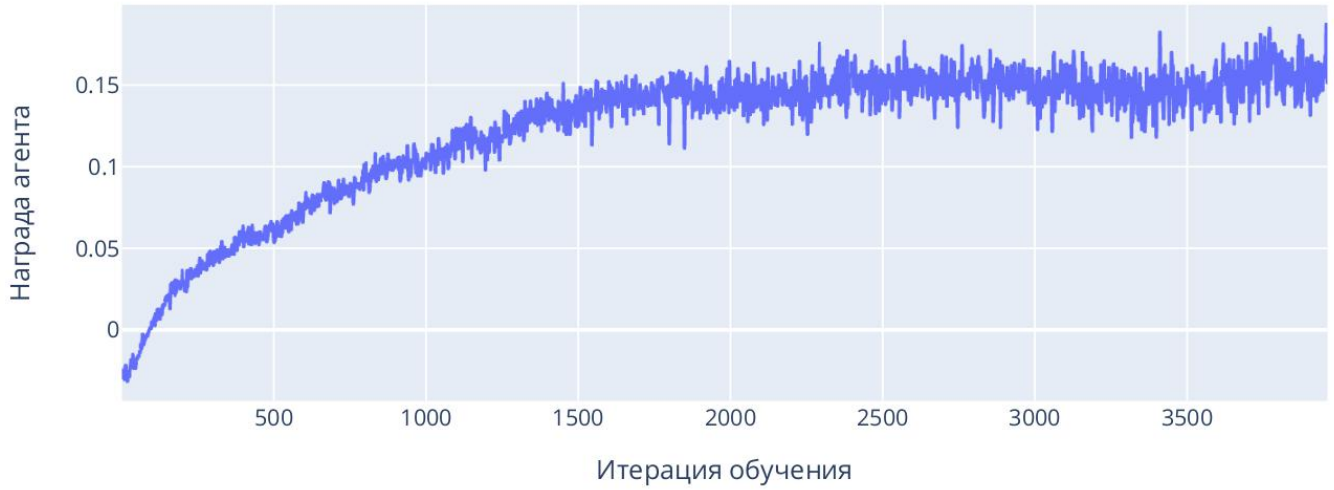

Рисунок 8. Суммарная награда на задаче генерации векторов, похожих на эмбеддинги

Награда агента

$$R_t = \mathrm{d}_{t-1} - \mathrm{d}_t$$

$$\mathrm{d}_t = \sum_i \min_j ||A \cdot \mathrm{memory}_{t,i} - \mathrm{emb}_j||$$

где:

- Rt - награда агента на шаге t
- $\mathrm{memory}_{t,i}$ - вектор в памяти на i-й позиции на шаге t
- emb_j - вектор в матрице эмбеддингов на j-й позиции
- A - случайно инициализированная матрица перехода

Результаты обучения модели с предобучением агента

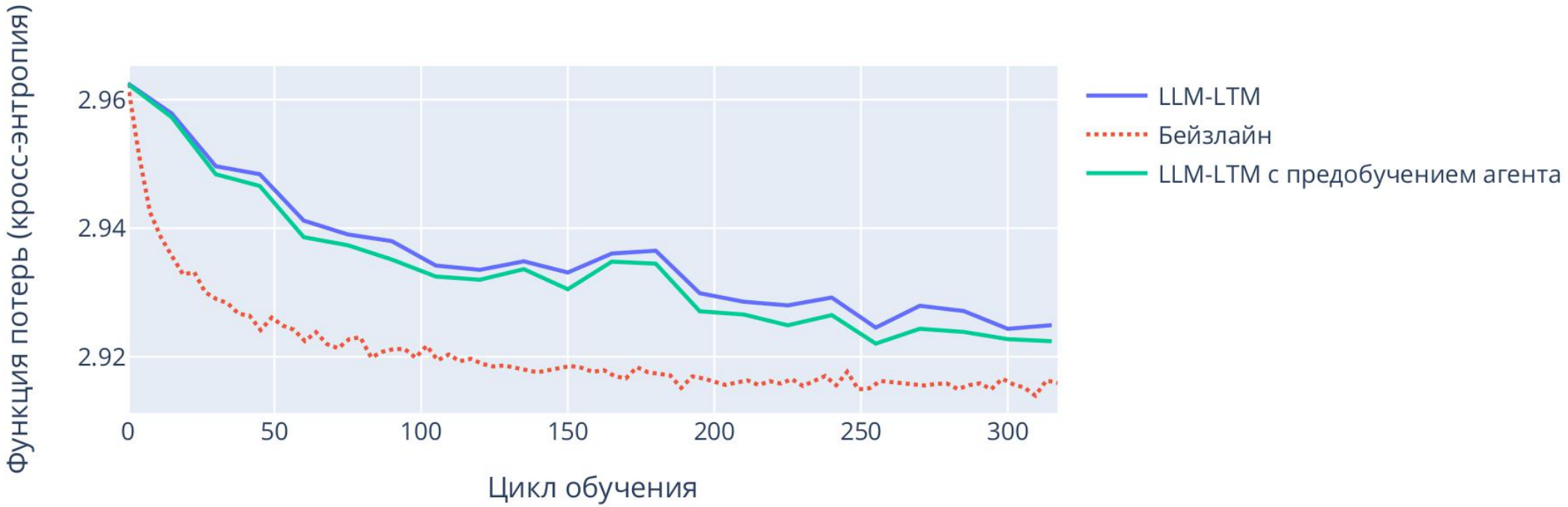

Рисунок 9. Функция потерь бейзлайна, LLM-LTM без предобучения агента и LLM-LTM с предобучением агента на валидационной выборке

Гипотеза 2: требуется предобучить LLM-LTM, помещая в память эмбеддинги с предыдущего сегмента

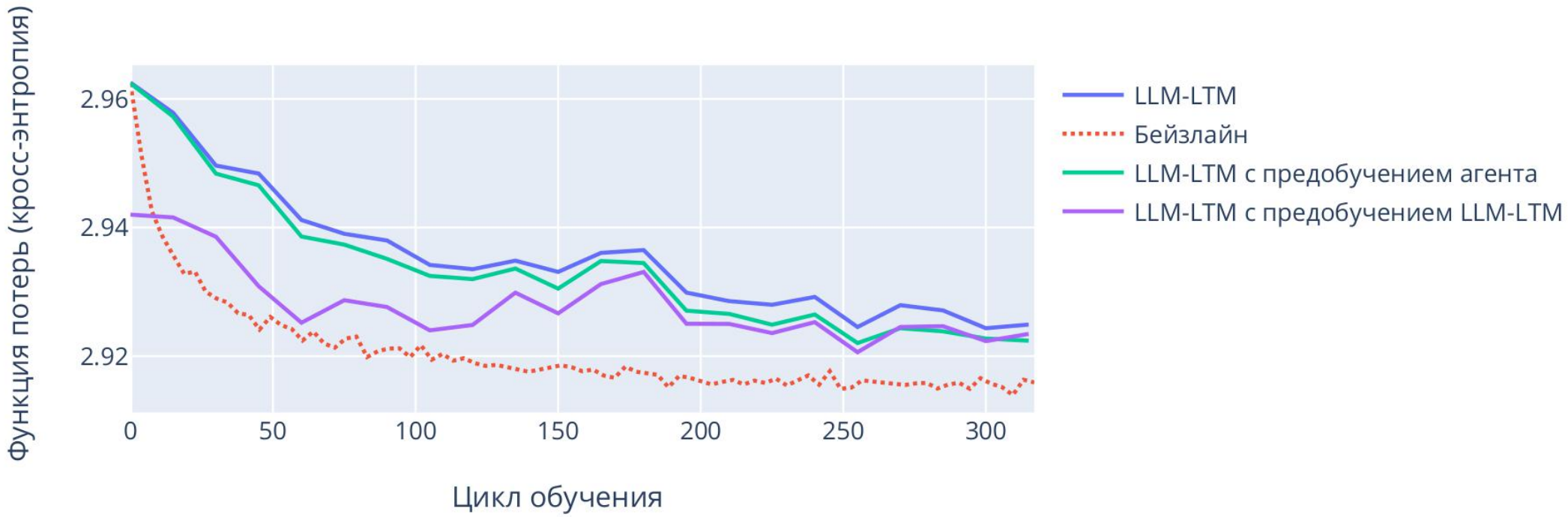

Рисунок 10. Сравнение функции потерь бейзлайна, LLM-LTM без модификаций, LLM-LTM с предобучением агента и LLM-LTM с предобучением LLM-LTM на валидационном множестве

Метрики на тестовой выборке

Модель	Кросс-энтропия	Перплексия
Бейзлайн	**2.925 ± 0.004**	**18.724 ± 0.081**
LLM-LTM с предобучением агента	2.929 ± 0.009	18.808 ± 0.082
LLM-LTM без предобучения агента	2.937 ± 0.004	18.947 ± 0.082

Таблица 1. Сравнение значений кросс-энтропии и перплексии на тестовой выборке

Модель	K=5	K=10	K=20	K=50	K=100
Бейзлайн	**0.6542**	**0.7173**	**0.7730**	**0.8380**	**0.8805**
LLM-LTM с предобучением агента	0.6539	0.7171	0.7728	0.8377	0.8802
LLM-LTM без предобучения агента	0.6531	0.7165	0.7721	0.8371	0.8795

Таблица 2. Сравнение значений метрики top-k accuracy на тестовой выборке

Итоги исследования

1. Собран русскоязычный датасет с поддержкой формирования контекста.
2. Разработан и реализован подход к созданию долгосрочной памяти для больших языковых моделей.
3. Проведено обучение предложенных моделей, выдвинуты и проверены гипотезы по улучшению предложенного подхода.
4. Выбранные метрики оценки качества предложенного подхода демонстрируют схожие, но не превосходящие результаты по сравнению с бейзлайном, не использующим механизмы памяти. Требуется дальнейший анализ метрик, оценивающих эффективность использования памяти.

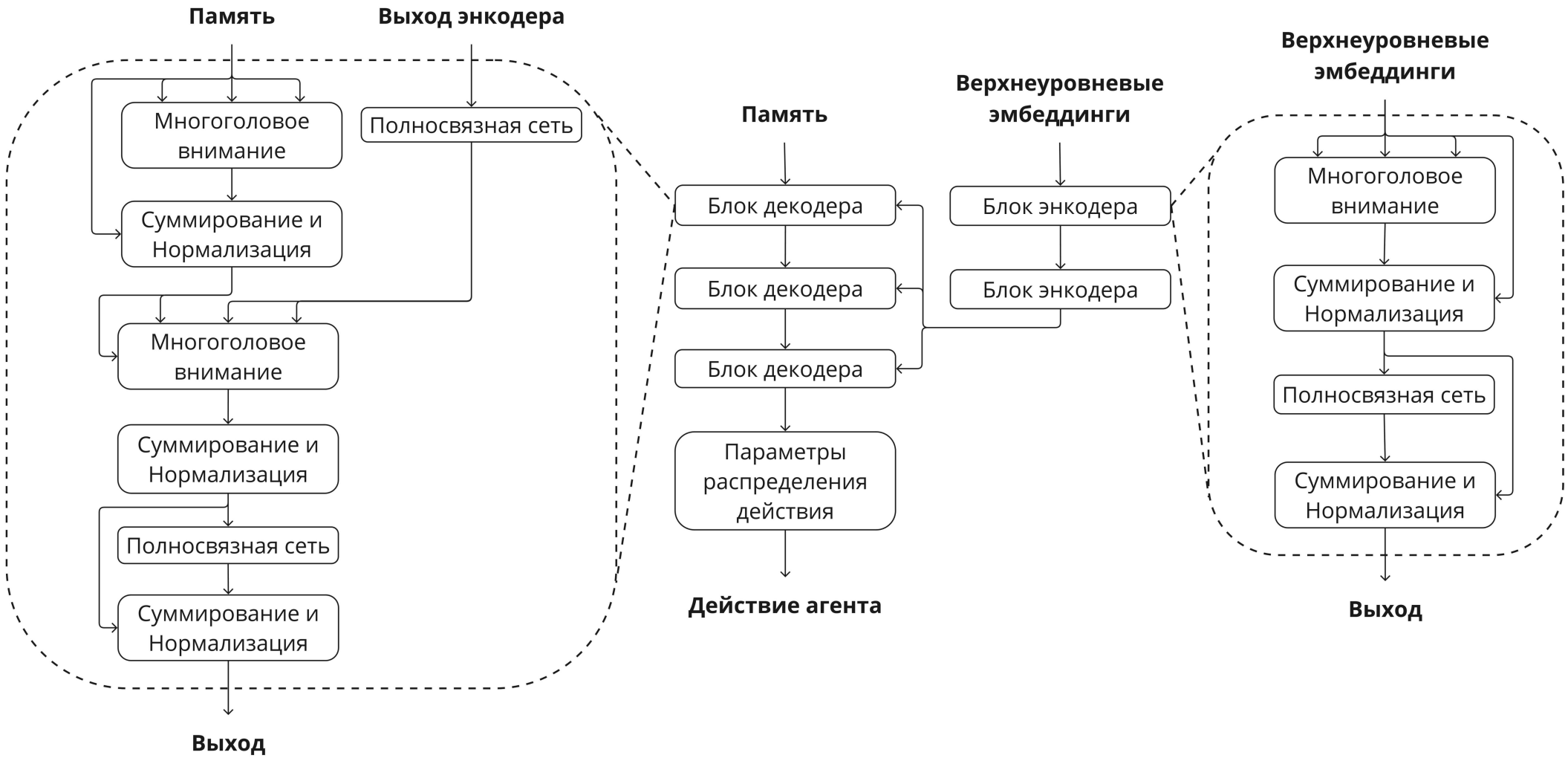

Рисунок 11. Архитектура модели памяти

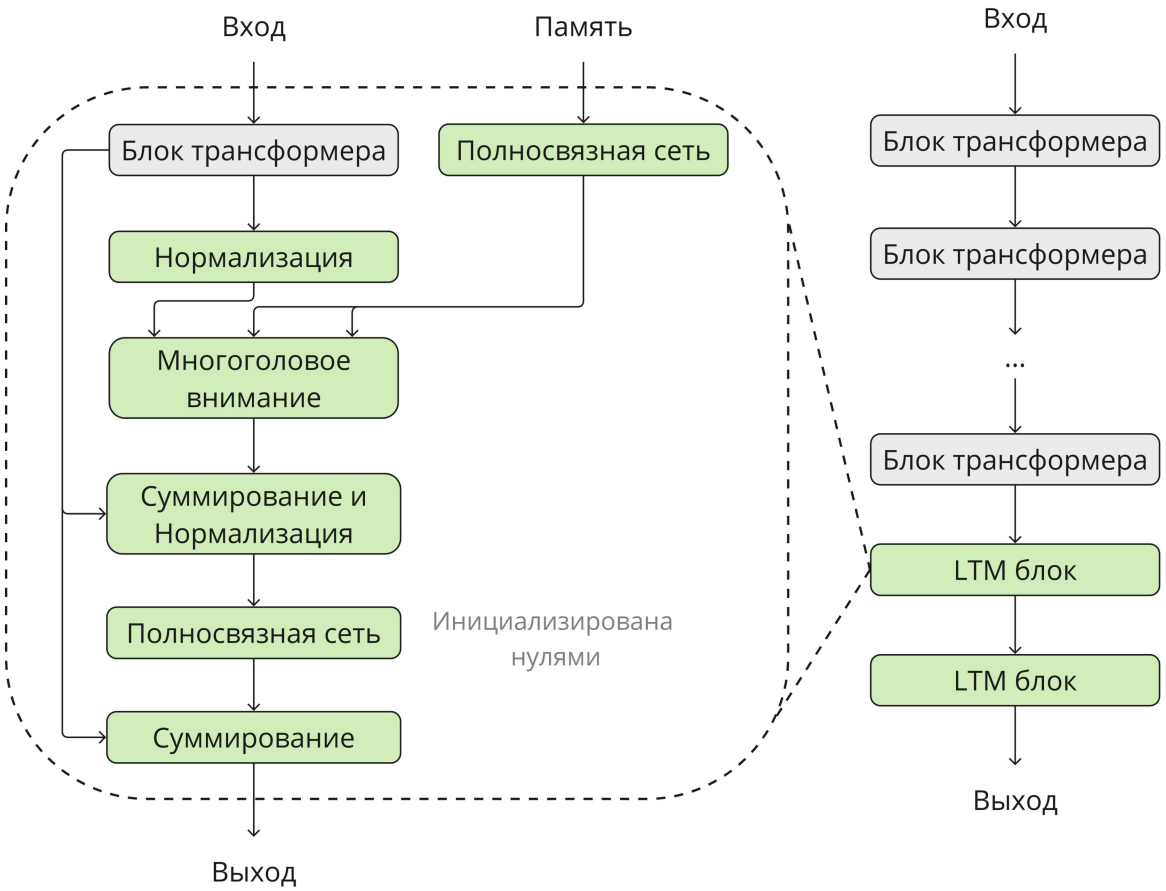

Рисунок 12. Архитектура LLM-LTM

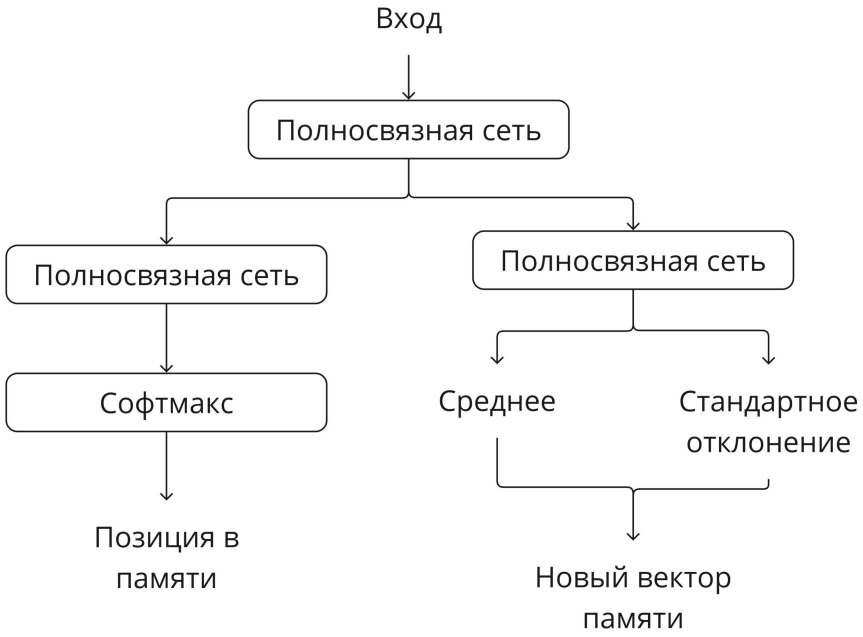

Рисунок 13. Архитектура блока «Параметры распределения действия»