Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Щукин Илья Вячеславович

Интерпретация механизма роутинга в архитектуре Mixture of Experts

Отчёт по учебной практике

Научный руководитель:
ассистент кафедры ИАС Чернышев Г. А.

Санкт-Петербург
2025

Оглавление

1 Введение

Область обработки естественного языка активно развивается. Большие языковые модели продолжают совершенствоваться и находят все больше применений в различных инструментах. Одним из возможных методов улучшения качества языковых моделей является увеличение общего числа параметров. Однако с ростом числа параметров требуется использовать больше вычислительных ресурсов, что в свою очередь увеличивает конечную стоимость обучения модели. Решением данной проблемы может стать изменение архитектуры моделей.

Метод Sparse Mixture of Experts [9] позволил обучение больших моделей за счёт использования условного вычисления активаций. Этот метод распределяет токены между отдельными частями сети — экспертами. При этом для токенов выбирается только часть экспертов. Это позволяет существенно увеличить число параметров модели без значительного увеличения времени необходимого для обучения и предсказаний.

На данный момент метод Mixture of Experts широко распространен и используется в некоторых передовых моделях, например V3 [3] и R1 [2] от DeepSeek. Но несмотря на это, наши возможности для интерпретации, а также понимание внутреннего устройства моделей сильно ограничены.

В данной работе предлагается новый метод для выявления экспертов, специализирующихся на определенных данных: код, математика и т.д. А также, метод для управления поведением моделей. Результат данный работы может быть в дальнейшем использован в задачах интерпретации, дообучения и прунинга.

2 Постановка задачи

Целью данной работы является исследование интерпретации и управления языковыми моделями на основе Mixture of Experts.

- Выполнить обзор предметной области
- Исследовать интерпретируемость активаций MoE

3 Обзор

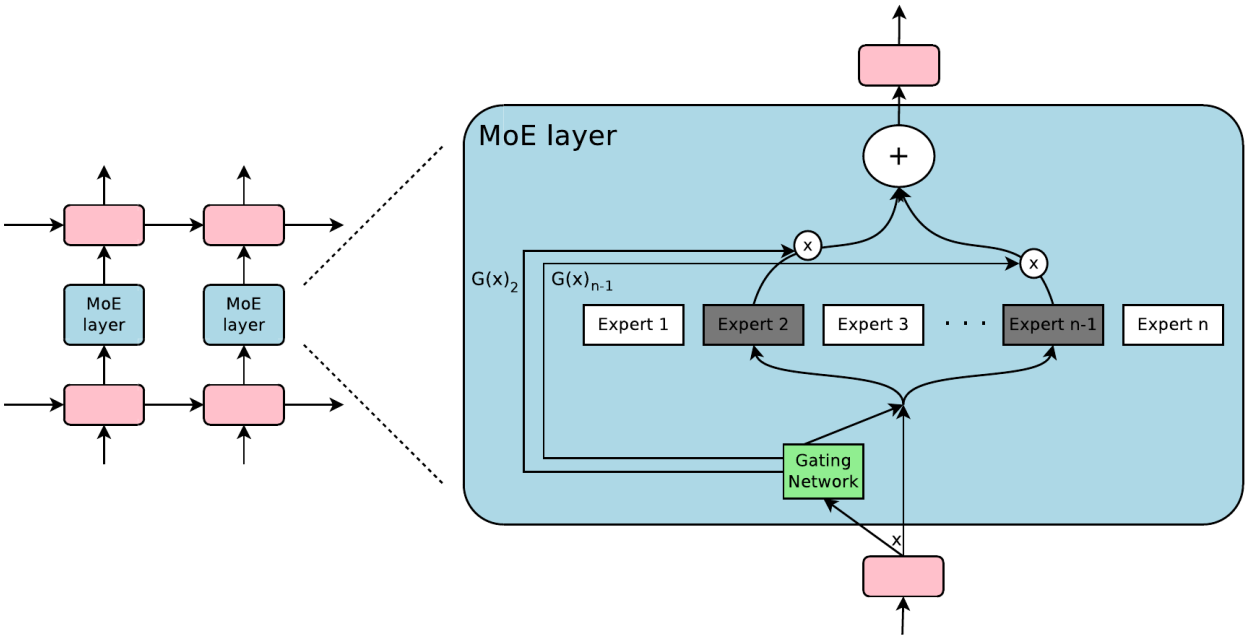

Рис. 1: Mixture of Experts слой между LSTM слоями. Изображение взято из [9].

3.1 Mixture of Experts

Mixture of Experts — это особое архитектурное решение, которое предполагает наличие слоёв-экспертов и слоя-роутера. Роутер обрабатывает входные данные и распределяет их между обработчиками-экспертами с весами-оценками того, насколько данные подходят конкретному эксперту. Эксперты могут представлять собой произвольные нейронные сети, но чаще всего для языковых моделей они представляют MLP (Multilayer perceptron) слои трансформеров. На практике часто используется разреженный вариант MoE [9], где после получения оценок применяется операция top-k и используются только наиболее подходящие токену эксперты. В оригинальной работе предлагалось использовать sparse MoE в RNN сетях между LSTM слоями, как показано на изображении 1. Формально MoE можно представить следующим образом:

$$y = \sum_{i=1}^{n} G(x)_i E_i(x) \tag{1}$$

где y выход MoE слоя, x вход, G роутер, а E_i эксперт. В случае sparse MoE в качестве роутера можно использовать следующую функцию:

$$G(x) = Softmax(topK(x \cdot W_g)) \tag{2}$$

При этом, если $G(x)_i = 0$, то $E_i(x)$ в выражении (1) не вычисляется, за счёт чего достигается разреженность.

Одним из недостатков sparse MoE является проблема "вырождения" роутеров, роутер может в значительной степени предпочитать отдельных экспертов и игнорировать других, что в свою очередь приводит к уменьшению фактически используемого числа параметров и упрощению модели. Крайней степенью "вырождения" может являться постоянный выбор одних и тех же экспертов, где при этом остальные эксперты не используются. Такая модель практически является обычной dense моделью, лишенной преимуществ MoE. Для борьбы с этой проблемой необходимо использовать добавки к функции потерь, штрафующие роутеры за неравномерное распределение токенов между экспертами. Так например в работе [5] предлагается использовать следующую добавку:

$$\alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i$$

где α — гиперпараметр, N — число экспертов, f_i — это доля токенов, которые попали в i-ого эксперта.

$$f_i = \frac{1}{T} \sum_{x \in \mathcal{B}} 1 \{\text{argmax } p(x) = i\}$$

и P_i — это доля вероятностной массы, которая досталась i-ому эксперту.

$$P_i = \frac{1}{T} \sum_{x \in \mathcal{B}} p_i(x),$$

где T — число токенов в батче $\mathcal{B}$.

3.2 Модель Mixtral

Большое внимание к архитектуре Mixture of Experts привлекла модель Mixtral 8x7B [7]. У данной модели всего 47B параметров и только 13B активных, при этом на множестве бенчмарков она обошла значительно превосходящую ее по размерам dense модель Llama 2 70B.

В данной работе нам наиболее интересен раздел с анализом роутинга. В нем авторы исследуют специализацию экспертов в своей модели. Для этого они исследуют зависимости между источниками в датасете Pile [10] и частотой выборов отдельных экспертов в рамках выбранных слоёв. В своей работе авторы не нашли явной зависимости между экспертами и отдельными доменами.

3.3 Mixture-of-Experts is Embedding Model

В работе [6] авторы показали, что MoE модели возможно использовать в качестве энкодеров эмбеддингов без дообучения. Для построения эмбеддингов они извлекают активации роутеров для каждого слоя и конкатенируют их. В своих эксперементах они использовали несколько MoE моделей разных размеров и сравнили их с другими моделями-эмбеддерами. Лучше всего себя проявляет метод, где смешиваются скрытые состояния модели и эмбеддинги с активациями роутеров. Результаты данной работы показали, что активации роутеров можно использовать для получения представлений текстов.

3.4 Mixture of Tunable Experts

В данной работе [8] авторы используют похожий метод. Для модели R1 они выделяют на синтетическом датасете экспертов, которые ответственны за отказы моделей отвечать на вопросы. Для этого они строят эмбеддинги на основе активаций роутеров. Далее они показали, что отключение данных экспертов уменьшает число отказов. Также они обнаружили, что при увеличении весов для данных экспертов число отказов возрастает.

Данная работа подтверждает, что отдельные эксперты могут потенциально обладать специализацией и то, что зная нужных экспертов возможно изменять поведение модели. Ограничением данной работы является то, что она не затрагивает более разнообразные данные и исследует специфичный случай для конкретной модели.

3.5 GigaChat 20B

GigaChat 20B[1] это open source модель от Сбера с архитектурой MoE, всего в ней 20B параметров и 3B активных. Она обучалась с нуля на корпусе данных с большим количеством русскоязычных текстов.

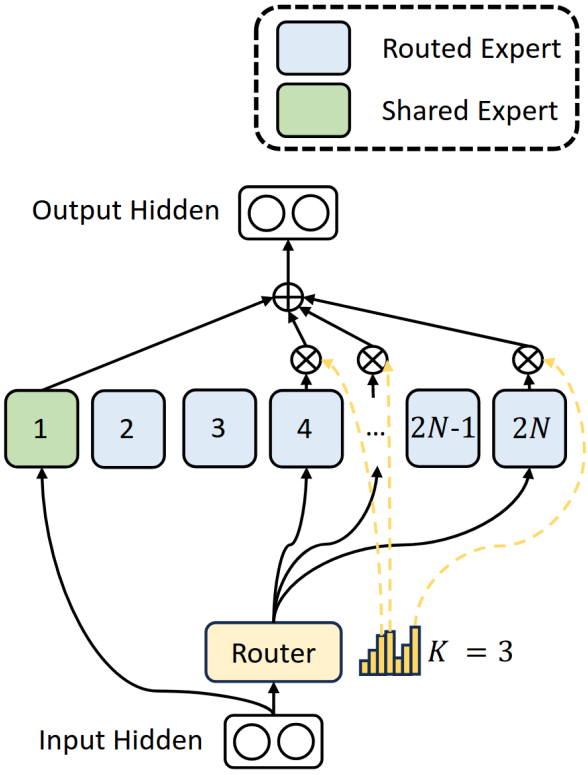

Рис. 2: Архитектура слоя MoE. Зеленым отмечены общие эксперты, а синим обычные. Изображение взято из [4].

В каждом блоке трансформера кроме самого первого MLP слой заменен на слой MoE. MoE слои в данной модели устроены сходным с DeepseekMoE [4] образом. На каждом слое по 64 эксперта. Эксперты представляют собой MLP слои с низкой внутренней размерностью (при-

[1]https://huggingface.co/ai-sage/GigaChat-20B-A3B-base

мерно в 8 меньше, чем если бы использовался обычный MLP слой вместо MoE), при этом для каждого токена выбираются сразу 6 экспертов. Также, в каждом слое есть по два общих эксперта, которые выбираются всегда с весом 1. Такая структура позволяет улучшить специализацию экспертов за счёт гранулярности и переноса одинаковых знаний в общих экспертов.

4 Исследование

4.1 Интерпретация активаций

Для исследования был выбран датасет The Pile[2] [10] поскольку он содержит в себе множество разнообразных примеров, а также для каждого из них указан соответствующий источник: Arxiv, Github и тд.

Построим эмбеддинги для подмножества датасета The Pile. Возьмем 40000 случайных примеров. Эмбеддинг E это матрица размера $l \times e$, где l это число слоев, а e число экспертов в одном слое (без учета общих экспертов). Для каждого примера E_{ij} вычисляется как число активаций эксперта j на слое i поделенное на число токенов в примере. Таким образом мы получаем эмбеддинг, содержащий информацию о выборах роутеров с каждого слоя модели для каждого токена.

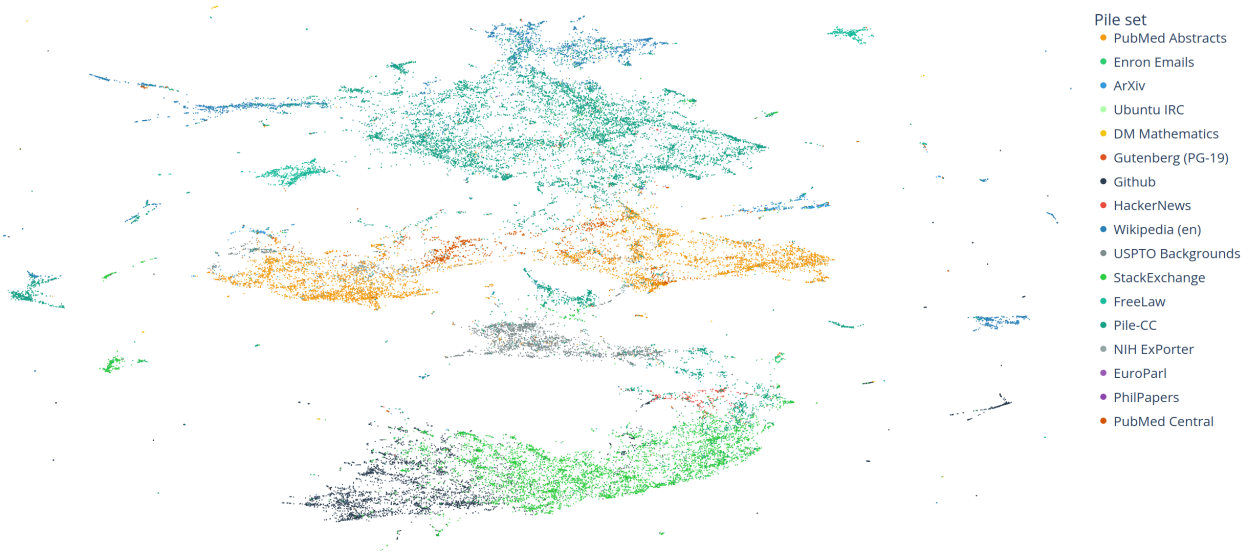

Рис. 3

После этого получим двумерную проекцию эмбеддингов с помощью UMAP, рис. 3. Можно заметить, что примеры разбились на кластеры, которые преимущественно содержат какой-то один источник данных, например Github. Это показывает, что информации, полученной на основе выборов роутеров достаточно для кодирования доменов примеров.

[2]В работе используется версия датасета без данных, нарушающих права правообладателей huggingface.co/datasets/monology/pile-uncopyrighted

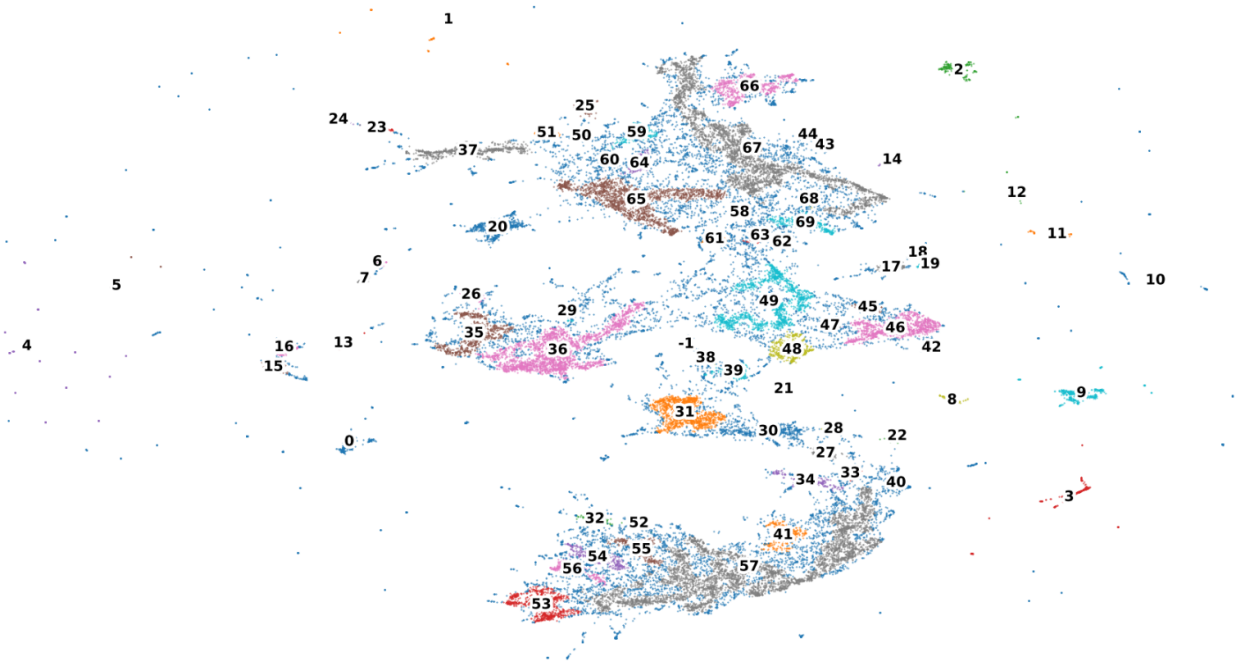

Рис. 4

Далее выделим кластера меньшего размера с помощью алгоритма кластеризации HDBSCAN [1]. Получившиеся кластера можно увидеть на картинке 4. Для того, чтобы иметь возможность примерно оценить содержимое полученных кластеров — выделим с помощью tf-idf ключевые слова. Можно увидеть кластера для: спорта, кулинарных рецептов, биологии, программирования и т.д.

Пример выделенных ключевых слов для четырех кластеров:

Кластер 7: teams, football, world, group, season, cup, team, week, games, round;

Кластер 25: like, food, coffee, just, cheese, wine, good, make, tea, chocolate;

Кластер 29: species, strains, isolates, genes, resistance, coli, cattle, gene, isolated, genus;

Кластер 51: room, property, home, bedroom, apartment, hotel, house, beach, area, located.

Построим эмбеддинги для полученных кластеров, для этого возьмем эмбеддинги, соответствующие попавшим в кластер примерам и усредним. С помощью данных эмбеддингов можно выделить экспертов наиболее важных для какого-то домена. Для этого оставим в эмбеддинге

только значения большие, чем $\frac{3}{e}$, для нашей модели это $\frac{3}{64}$, то есть в три раза большие, чем среднее значение для слоя. Таким образом мы оставим наиболее важных экспертов для домена. На рис. 5 показана карта активаций для эмбеддинга кластера 25 до и после выделения самых важных экспертов.

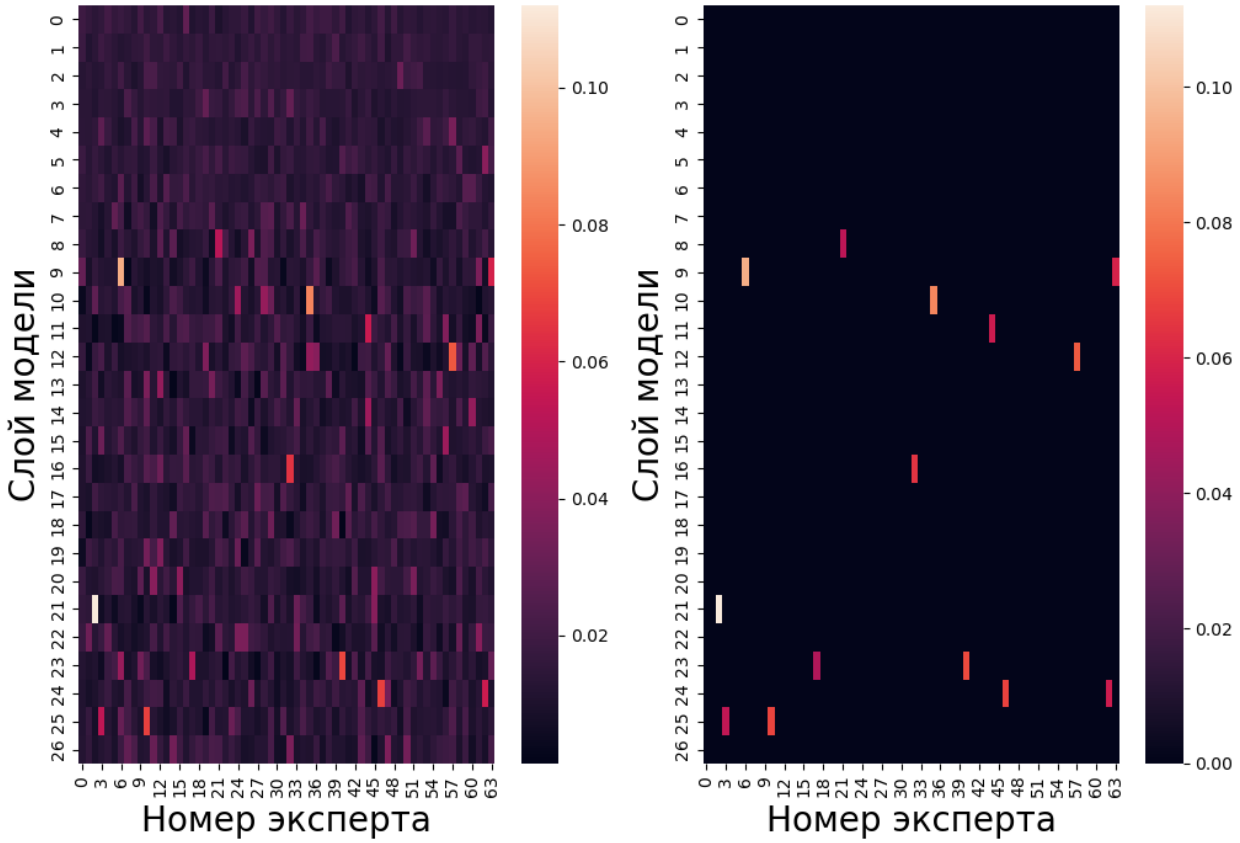

Рис. 5

4.2 Управление моделью

Чтобы дополнительно продемонстрировать, что выбранные эксперты действительно специализируются на выбранном домене мы можем воспользоваться полученными эмбеддингами для управления моделью. Для этого необходимо увеличить вероятность выбора важных для домена экспертов. Мы ожидаем, что модель после управления будет генерировать тексты связанные с выбранным доменом.

Самым простым способом будет добавить с коэффициентом полученные матрицы эмбеддингов к логитам роутеров. Перепишем выраже-

ние (2) с данной модификацией:

$$G(x) = Softmax(topK(x \cdot W_g + c \cdot E_c)) \tag{3}$$

где c — это коэффициент управления, а E_c — вектор экспертов кластера, полученный ранее. Данный метод прост в реализации и он действительно склоняет модель генерировать токены из нужного домена, однако он вредит языковым способностям модели.

Чтобы меньше вредить модели мы можем модифицировать этот метод и попытаться таким образом стабилизировать генерации. Для этого мы можем добавить множитель, зависящий от логитов роутера, к которым мы прибавляем наш эмбеддинг. Тогда формула (3) изменится и будет иметь следующий вид:

$$G(x) = Softmax(topK(x \cdot W_g + max(x \cdot W_g) \cdot c \cdot E_c)) \tag{4}$$

Домножение на самый большой вес позволяет масштабировать добавку. Также до добавления эмбеддинга управления мы можем занулить исходные логиты для экспертов, вероятности которых мы изменяем. Это позволяет избежать роста весов по мере генерации токенов. Таким образом мы можем обозначить через $Z_E(L)$ преобразование, заменяющее значения вектора L на ноль, если они соответствуют экспертам из E. Таким образом мы получим формулу:

$$G(x) = Softmax(topK(Z_{E_c}(x \cdot W_g) + max(x \cdot W_g) \cdot c \cdot E_c)) \tag{5}$$

Далее используется данная формула для получения примеров.

4.3 Примеры управления

4.3.1 Без управления

Запрос: Жил-был пёс по имени Барни. Он был большой, лохматый и очень добрый. Барни жил вместе со своей хозяйкой Машей в небольшом уютном доме на окраине города. Маша была студенткой и часто уезжала на занятия, но каждый раз возвращалась домой с радостью,

потому что знала – её верный друг всегда ждёт у двери.

Ответ: А ещё она знала, что Барни никогда не обидит ни одного живого существа. Барни любил гулять по лесу, который начинался сразу за домом. Там он мог побегать, поиграть с другими собаками или просто полежать под деревом и послушать пение птиц. Но больше всего ему нравилось наблюдать за лесными зверями. Особенно его интересовали белки. Они были такие забавные! То они бегали по деревьям, то прыгали с ветки на ветку, а иногда даже спускались на землю и играли в догонялки. Барни тоже хотел играть с белками, но они почему-то боялись его и убегали при виде собаки. Тогда Барни решил стать другом для всех лесных жителей.

4.3.2 С эмбеддингом для веб-сервисов

Запрос: Жил-был пёс по имени Барни. Он был большой, лохматый и очень добрый. Барни жил вместе со своей хозяйкой Машей в небольшом уютном доме на окраине города. Маша была студенткой и часто уезжала на занятия, но каждый раз возвращалась домой с радостью, потому что знала – её верный друг всегда ждёт у двери.

Ответ: А когда она открывала дверь, он радостно махал ей лапами. Маша любила своего питомца, а он любил свою хозяйку. Но однажды случилось несчастье: Маша заболела и попала в больницу. Она не могла навещать своего любимого друга, ведь он был такой большой и добрый! Тогда Маша решила написать специальную программу, которая бы делала всё то же самое, что делал Барни. И вот, спустя некоторое время, Маша смогла снова войти в свой любимый сайт, чтобы посмотреть фотографии своих любимых друзей ВКонтакте.

4.3.3 С эмбеддингом для спорта

Запрос: Жил-был пёс по имени Барни. Он был большой, лохматый и очень добрый. Барни жил вместе со своей хозяйкой Машей в небольшом уютном доме на окраине города. Маша была студенткой и часто уезжала на занятия, но каждый раз возвращалась домой с радостью,

потому что знала – её верный друг всегда ждёт у двери.

Ответ: А ещё он любил играть с детьми. Барни был не просто другом для детей, а настоящим защитником. Однажды, когда дети играли в футбол, к ним подошёл незнакомый мальчик и предложил сыграть за их команду. Но ребята отказались, так как они уже были в меньшинстве. Тогда незнакомец сказал: «Ну и пусть играют одни!», взял мяч и начал дриблинг. И тут же забивает гол в ворота команды Барни. Вратарь стал возмущаться, но судья матча показал ему жёлную карточку.

5 Заключение

В ходе работы были достигнуты следующие результаты:

1. Выполнен обзор предметной области.
2. Исследована интерпретируемость активаций MoE. Предложены методы для анализа активаций и управления моделью.

В дальнейшем планируется развить метод управления и провести эксперименты с другими моделями. В данный момент по материалу данной работы готовится публикация. Код исследования опубликован на github[3].

[3] github.com/Elluran/concentration_notebooks

Список литературы

[1] Campello Ricardo JGB, Moulavi Davoud, and Sander Jörg. Density-based clustering based on hierarchical density estimates // Pacific-Asia conference on knowledge discovery and data mining / Springer. — 2013. — P. 160–172.

[2] Guo Daya, Yang Dejian, Zhang Haowei, Song Junxiao, Zhang Ruoyu, Xu Runxin, Zhu Qihao, Ma Shirong, Wang Peiyi, Bi Xiao, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning // arXiv preprint arXiv:2501.12948. — 2025.

[3] Liu Aixin, Feng Bei, Xue Bing, Wang Bingxuan, Wu Bochao, Lu Chengda, Zhao Chenggang, Deng Chengqi, Zhang Chenyu, Ruan Chong, et al. Deepseek-v3 technical report // arXiv preprint arXiv:2412.19437. — 2024.

[4] Dai Damai, Deng Chengqi, Zhao Chenggang, Xu RX, Gao Huazuo, Chen Deli, Li Jiashi, Zeng Wangding, Yu Xingkai, Wu Yu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models // arXiv preprint arXiv:2401.06066. — 2024.

[5] Fedus William, Zoph Barret, and Shazeer Noam. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity // Journal of Machine Learning Research. — 2022. — Vol. 23, no. 120. — P. 1–39.

[6] Li Ziyue and Zhou Tianyi. Your mixture-of-experts llm is secretly an embedding model for free // arXiv preprint arXiv:2410.10814. — 2024.

[7] Jiang Albert Q, Sablayrolles Alexandre, Roux Antoine, Mensch Arthur, Savary Blanche, Bamford Chris, Chaplot Devendra Singh, Casas Diego de las, Hanna Emma Bou, Bressand Florian, et al. Mixtral of experts // arXiv preprint arXiv:2401.04088. — 2024.

[8] Dahlke Robert, Klagges Henrik, Zecha Dan, Merkel Benjamin, Rohr Sven, and Klemm Fabian. Mixture of Tunable Experts-Behavior

Modification of DeepSeek-R1 at Inference Time // arXiv preprint arXiv:2502.11096. — 2025.

[9] Shazeer Noam, Mirhoseini Azalia, Maziarz Krzysztof, Davis Andy, Le Quoc, Hinton Geoffrey, and Dean Jeff. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer // arXiv preprint arXiv:1701.06538. — 2017.

[10] Gao Leo, Biderman Stella, Black Sid, Golding Laurence, Hoppe Travis, Foster Charles, Phang Jason, He Horace, Thite Anish, Nabeshima Noa, et al. The pile: An 800gb dataset of diverse text for language modeling // arXiv preprint arXiv:2101.00027. — 2020.