Аннотация

В данной работе исследуется способ уменьшения размерности пространства обучаемых параметров в задаче детектирования ai текстов, задача многоклассовой классификации. Для fine tuning использовалась модель RoBerta с LoRA адаптером. Было проведено несколько экспериментов, чтобы выяснить, является ли использование LoRA для аппроксимации матрицы весов эффективным с точки зрения времени, ресурсов или точности. Было показано, что при меньших ресурсах модель distilled RoBerta base с LoRA адаптером может получить те же показатели метрик для классификации текстов, написанных человеком, что и vanilla distilled RoBerta base на наборе данных с 4 классами.

Ключевые слова: машинное обучение; линейная алгебра; аппроксимация матриц; уменьшение размерности пространств; классификация AI текстов; многоклассовая классификация текстов; большие языковые модели.

Содержание

1 Введение

Актуальность темы. Уменьшение размерности пространства обучаемых параметров в задаче адаптации к домену упрощает процесс обучения и улучшает вычислительную эффективность. Путем сокращения количества параметров, которые необходимо обновить во время обучения, модель может потенциально быстрее сходиться и затрачивать меньше вычислительных ресурсов; это связано с структурой нейронной сети — ее сложность и потребление ресурсов зависят от количества обучаемых параметров. Уменьшение размерности может быть особенно важным в сценариях адаптации к домену, где обрабатываются большие объемы данных и происходит обучение с большим числом параметров.

Методы, направленные на решение проблемы снижения размерности: метод главных компонент [1] и его адаптации: тензорное разложение [2, 3], каноническое полиадическое разложение [4] выбирают наиболее важные векторы признаков из набора данных, используя сингулярное разложение матрицы для нахождения первых K собственных векторов с наибольшим собственным значением. Методы, осуществляющие отбор признаков: регуляризация LASSO (L1) [5], оценка Фишера [6] или тест Хи-квадрат [7]. Метод снижения размерности, основанный на дообучении больших текстовых моделей — дистилляция [8]; в этом методе большая генеративная модель является *учителем*, а меньшая — *учеником*. Модель ученика обучают с использованием прогнозов *учителя*. Эти идеи были впервые представлены в работах Дж. Хинтона [9] и В.Н. Вапника [10].

Метод, рассмотренный в данной работе — низкоранговое разложение (англ. Low Rank Adaptation) [11], который разработан на основе идеи о том, что предварительно обученные языковые модели имеют низкую внутреннюю размерность и могут эффективно обучаться, несмотря на проецирование на меньшее подпространство [12]. Данный метод, как и метод главных компонент, использует сингулярное разложение матрицы для нахождения низкоранговых приближений матрицы весов. LoRA используется для решения различных проблем seq2seq, таких как: [13, 14, 15]. Этот подход особенно популярен в

задачах преобразования видео в текст, так как этим задачам свойственны вариативность распределения входных данных и разнообразие задач, обусловленные дополнительным визуальным входным данным [15].

В данной работе метод LoRA применяется к задаче обнаружения текстов, написанных искуственным интеллектом или человеком. Задача обнаружения текстов, написанных искусственным интеллектом стала особенно популярна с выходом новых больших моделей от OpenAI и Google [16, 17], так как определить кем написан тот или иной текст все сложнее [18, 19]. В данной статье мы работали над дообучением популярной модели RoBerta с использованием LoRA адаптеров с целью понижения размерности пространства обучаемых параметров. Предполагается, что LoRA может быть так же эффективен в решении задач классификации, как и в задачах генерации: LoRA доказала свою эффективность во многих задачах генерации [14, 11, 15].

Как отмечено в работах [20, 21], все подходы к решению задачи классификации ai текстов разделяются на два типа: ориентированные на анализ признакового пространства и ориентированные на дообучение моделей. Анализ признакового пространства основывается на извлечении и анализе характеристик текста — лексических, синтаксических, семантических или стилистических характеристик: [22, 23, 24]. Дообучение больших языковых моделей основывается на изучении параметров и возможностей модели и последующем дообучении модели к задаче классификации текстов: [25, 25, 26].

Цели работы. Исследуются методы снижения пространства обучаемых параметров при помощи сингулярного разложения матриц, а также коорректность применения изучаемых методов к задаче классификации текстов.

Методы исследования. Применяется низкоранговое разложение (англ. Low Rank Adaptation) к матрицам параметров в больших языковых моделях. Используются статистические методы оценки функции минимизации функции потерь, а также свойтва матриц для доказательства применимости предложенного метода к задаче классификации.

Теоретическая значимость. В работе проведен теоретический анализ проблемы снижения размерности пространства обучаемых параметров. Доказана теорема об применимости модели BERT [27] с адаптером LoRA к задаче многоклассовой классификации.

Практическая значимость. Проведен вычислительный эксперимент, показывающий улучшение качества и экономию ресурсов при решении задачи классификации текстов.

2 Постановка задачи классификации текстов

Модель трансформера решает задачу генерации последовательность в последовательность (англ. seq2seq), которая формулируется следующим образом согласно [28]:

$$f_\theta : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}, \tag{2.1}$$

где модель f_θ отображает неупорядоченный набор из n элементов в $\mathbb{R}^d$ в другой неупорядоченный набор из n элементов в $\mathbb{R}^d$. Для решения задачи классификации текстов(?) сформулируем постановку задачи.

$$f_\theta : \hat{V} \to [N_c], \tag{2.2}$$

где $\hat{V} \subset V^*$; V — словарь токенов и V^* — его замыкание или множество всех последовательностей над V, $[N_c]$ — множество классов. Таким образом, модель отображает текст из $\hat{V}$ в класс из $[N_c]$.

Тогда $(X_i, c_i) \in \hat{V} \times [N_c]$ для $i \in [N_{data}]$ является парой текст — класс, выбранной из $P(X, c)$, где X_i — входной текст, а c_i — его класс. Таким образом, наша цель — оценить $P(c|X)$.

Согласно [11] при дообучении модель инициализируется предварительно обученными весами Φ_0 и обновляется до $\Phi_0 + \Delta\Phi$, где $\Delta\Phi$ — набор дообучаемых параметров такой, что $\mid \Delta\Phi \mid = \mid \Phi_0 \mid$. Тогда задача минимизации функции потерь имеет вид:

$$\begin{aligned} \min_{\Phi} \left(- \sum_{X_i \in \hat{V} \subset V^*} \sum_{c_i \in [N_c]} \log \left(P_\Phi \left(c_i \mid X_i \right) \right) \right) = \\ = \max_{\Phi} \sum_{X_i \in \hat{V} \subset V^*} \sum_{c_i \in [N_c]} \log \left(P_\Phi \left(c_i \mid X_i \right) \right), \end{aligned} \tag{2.3}$$

В то время как при использовании LoRA $\Delta\Phi$ задается набором параметров Θ намного меньшего размера: $\Delta\Phi = \Theta$, где $\mid \Theta \mid \ll \mid \Phi_0 \mid$ и

задача минимизации функции потерь имеет вид:

$$\begin{aligned}\min_{\Theta}\left(-\sum_{X_i\in\hat{V}\subset V^*}\sum_{c_i\in[N_c]}\log\left(P_{\Phi_0+\Theta}\left(c_i\mid X_i\right)\right)\right)=\\ =\max_{\Theta}\sum_{X_i\in\hat{V}\subset V^*}\sum_{c_i\in[N_c]}\log\left(P_{\Phi_0+\Theta}\left(c_i\mid X_i\right)\right).\end{aligned}\tag{2.4}$$

3 Предложенный метод и его корректность

В данной работе LoRA применяется к задаче классификации. Структура обновления весов при использовании LoRA адаптера описана в таблице 1,

Fine tuning	LoRA fine tuning
$W_{upd} = W + \Delta W$ $\hat{y} = xW_{upd} = x(W + \Delta W)$ $\hat{y} = xW + x\Delta W$	$W_{upd} = W + AB$ $\hat{y} = xW_{upd} = x(W + AB)$ $\hat{y} = xW + xAB$

Таблица 1: Структура обновления весов при использовании LoRA адаптера

где $W \in \mathbb{R}^{d\times k}$ — предобученные веса, $\Delta W \in \mathbb{R}^{d\times k}$ — матрица обновленных весов. ΔW приближается с помощью метода LoRA произведением AB, где $A \in \mathbb{R}^{d\times r}$, $B \in \mathbb{R}^{r\times k}$ и r — ранг матрицы, являющийся гиперпараметром модели. Здесь $A \sim \mathcal{N}(0,\, \sigma^2)$ и $B = [0]_{r\times k}$.

3.1 Состоятельность предложенной модели

Состоятельность модели трансформер была доказана в работе [29]. Доказательство приведено для задачи классификации:

Теорема 1. *Будем считать, что:*

1. *Задана модель с набором параметров Θ^*, генерирующая эмпирическое распределение данных $P_{model}(\cdot, \Theta^*)$, которое аппроксимирует истинное распределение данных P_{true} с минимальным расхождением по KL-дивергенции:*

$$\exists\Theta^* : \Theta^* = \arg\min_{\Theta} D_{KL}(P_{true} \,||\, P_{model}(\cdot, \Theta)), \tag{3.1}$$

2. *При увеличении размера выборки $\hat{V}$ эмпирическое распределение данных $P_{model}(\cdot, \Theta^*)$ приближается к истинному распределению, генерирующему данные.*

3. *Функция ошибки* $\mathscr{L}(\Theta)$ — *непрерывная, дифференцируемая. Где*

$$\mathscr{L}(\Theta) = -\frac{1}{\mid \hat{V} \mid} \sum_{X_i \in \hat{V}} \sum_{c_i \in [N_c]} \log \left(P_{\Phi_0 + \Theta} \left(c_i \mid X_i \right) \right). \tag{3.2}$$

Тогда минимизация функции потерь $\mathscr{L}(\Theta)$ *приводит к состоятельной оценке истинного распределения, порождающего данные. Т.е.:*

$$\lim_{|\hat{V}| \to \infty} \arg\min_{\Theta} \mathscr{L}(\Theta) = \Theta^*. \tag{3.3}$$

Доказательство. В силу равномерной сходимости функции потерь и силу утверждения 3 из статьи [30], которое приведено в приложении: минимум $\mathscr{L}(\Theta)$ стремится к минимуму ожидаемого риска

$$R_{exp} = \mathbb{E}_{X_i \sim P_{true}}[\mathscr{L}(X_i; \Theta)], \tag{3.4}$$

при размере $\hat{V}$ стремящемся к бесконечности:

$$\lim_{|\hat{V}| \to \infty} \arg\min_{\Theta} \mathscr{L}(\Theta) = \arg\min_{\Theta} R_{exp} = \arg\min_{\Theta} \mathbb{E}_{X_i \sim P_{true}}[\mathscr{L}(X_i; \Theta)]. \tag{3.5}$$

Достаточно доказать

$$\arg\min_{\Theta} \mathbb{E}_{X_i \sim P_{true}}[\mathscr{L}(X_i; \Theta)] = \Theta^*. \tag{3.6}$$

Подставим значение функции потерь:

$$\begin{gathered} \arg\min_{\Theta} \mathbb{E}_{X_i \sim P_{true}}[\mathscr{L}(X_i; \Theta)] = \\ \arg\min_{\Theta} \mathbb{E}_{X_i \sim P_{true}}[\sum_{c_i \in [N_c]} \log \left(P_{\Phi_0 + \Theta} \left(c_i \mid X_i; \Theta \right) \right)]. \end{gathered} \tag{3.7}$$

В силу определения KL-дивергенции,

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx = \mathbb{E}_{x \sim p(x)}[\log \left(\frac{p(x)}{q(x)} \right)], \tag{3.8}$$

верно

$$\arg\min_{\Theta} \mathbb{E}_{X_i \in P_{true}}[\sum_{c_i \in [N_c]} \log\left(P_{\Phi_0+\Delta\Phi(\Theta)}\left(c_i \mid X_i\right)\right] = \arg\min_{\Theta} D_{KL}(P_{true} \mid\mid P_{model}(\Theta)) = \Theta^*, \tag{3.9}$$

Последнее равенство справедливо по условию. И оценка

$\arg\min_{\Theta} \mathbb{E}\,\mathscr{L}(\Theta)$ является состоятельной оценкой распределения P_{true}. (3.10)

■

3.2 О применимости LoRA к задаче классификации

Докажем, что LoRA применима к задаче классификации. Для решения задачи классификации с помощью BERT [27] требуется не более чем дополнительный softmax слой после BERT [31]:

$$\begin{aligned} p(c \mid \mathbf{x}) &= \text{softmax}(W^T\mathbf{x}) \\ \hat{\mathbf{y}} &= \text{softmax}\left(W^T\mathbf{x}\right) = \frac{\exp\left(W^T\mathbf{x}\right)}{\sum_{i=1}^{k}\exp\left(W^T\mathbf{x}\right)_i}, \end{aligned} \tag{3.11}$$

где $\mathbf{x}$ — это выходной результат последнего слоя BERT, а W — матрица весов. Структура BERT представлена на рис. 1, где, согласно [28], слой внимания (англ. attention):

$$\begin{aligned} Q^{(h)}\left(\mathbf{x}_i\right) &= W_{h,q}^T\mathbf{x}_i, \\ K^{(h)}\left(\mathbf{x}_i\right) &= W_{h,k}^T\mathbf{x}_i, \quad W_{h,q}, W_{h,k}, W_{h,v} \in \mathbb{R}^{d\times k}, \\ V^{(h)}\left(\mathbf{x}_i\right) &= W_{h,v}^T\mathbf{x}_i, \end{aligned} \tag{3.12}$$

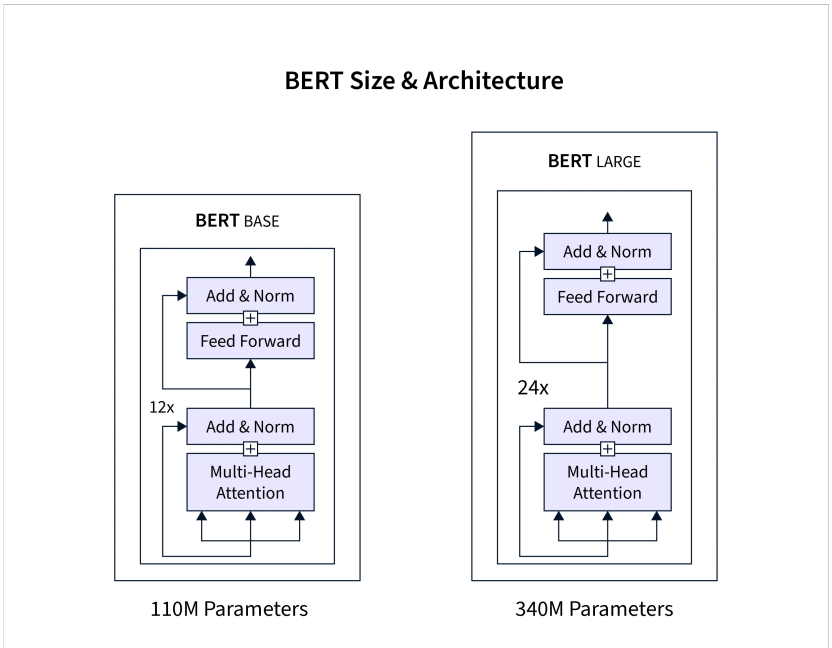

Рис. 1: Архитектура модели BERT

$$\begin{aligned} \alpha_{i,j}^{(h)} &= \text{softmax}_j \left(\frac{\left\langle Q^{(h)}\left(\mathbf{x}_i\right), K^{(h)}\left(\mathbf{x}_j\right) \right\rangle}{\sqrt{k}} \right) V^{(h)}\left(\mathbf{x}_j\right), \\ \mathbf{u}_i' &= \sum_{h=1}^{H} W_{c,h}^T \sum_{j=1}^{n} \alpha_{i,j}^{(h)}, \end{aligned} \tag{3.13}$$

где $Q^{(h)}\left(\mathbf{x}_i\right), K^{(h)}\left(\mathbf{x}_i\right), V^{(h)}\left(\mathbf{x}_i\right)$ — линейные проекции входного вектора $\mathbf{x}$ запрос, ключ, значение (англ. Query, Key, Value) и $\alpha_{i,j}^{(h)}$ — вектор внимания, $\mathbf{u}_i'$ — выходная матрица слоя attention такого же размера, как и входная $\mathbf{x}_i$ [27].

Нормализация по слою (англ. Layer normalization) — нормализует активации предыдущего слоя для каждого входа в партиции (англ. batch) независимо друг от друга, а не по всей партиции, как при batch нормализации. То есть, применяет преобразование, которое поддер-

живает среднее в пределах каждого входа близким к 0 и стандартное отклонение активации близким к 1 [32].

$$\mathbf{u}_i = \text{LayerNorm}\left(\mathbf{x}_i + \mathbf{u}'_i; \gamma_1, \beta_1\right), \tag{3.14}$$

Нормализация по слою может быть переписана в соответствии с [32] следующим образом:

$$\begin{aligned} &\text{LayerNorm}(\mathbf{z}; \gamma, \beta) = \gamma \frac{(\mathbf{z} - \mu_{\mathbf{z}})}{\sigma_{\mathbf{z}}} + \beta, \\ &\gamma, \beta \in \mathbb{R}^k. \\ &\mu_{\mathbf{z}} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{z}_i, \quad \sigma_{\mathbf{z}} = \sqrt{\frac{1}{k}\sum_{i=1}^{k} (\mathbf{z}_i - \mu_{\mathbf{z}})^2}. \end{aligned} \tag{3.15}$$

Сеть прямого распространения (англ. Feed Forward Network) представляет собой двухслойную нейронную сеть с активацией ReLU:

$$\mathbf{z}'_i = W_2^T \,\text{ReLU}\left(W_1^T \mathbf{u}_i + b_1\right) + b_2. \tag{3.16}$$

Нормализация по слою:

$$\mathbf{z}_i = \text{ LayerNorm }\left(\mathbf{u}_i + \mathbf{z}'_i; \gamma_2, \beta_2\right), \tag{3.17}$$

Теорема 2. *В рамках задачи классификации, при заданных условиях:*

1. *Модель семейства BERT с указанной выше математической структурой и дополнительным слоем*

$$\hat{\mathbf{y}} = \text{softmax}\left(W_{upd}^T \mathbf{x}\right) = \frac{\exp\left(W_{upd}^T \mathbf{x}\right)}{\sum_{i=1}^{k} \exp\left(W_{upd}^T \mathbf{x}\right)_i}, \tag{3.18}$$

где

$$W_{upd} = \underset{(d\times k)}{W} + \underset{(d\times k)}{\Delta W}, \tag{3.19}$$

и x —это выходной результат BERT, W — матрица весов, ΔW — матрица обновленных весов.

2. *Данная модель BERT без дополнительного слоя также корректно работает с аппроксимацией*

$$\underset{(d\times k)}{\Delta W} = \underset{(d\times r)}{A} \times \underset{(r\times k)}{B}, \tag{3.20}$$

3. *Условия теоремы выполняются 1. (можно считать данную модель состоятельной).*

Тогда можно утверждать, что при (3.20) *заданная модель BERT с дополнительным слоем гарантирует корректную выходную матрицу.*

Доказательство. Докажем, что выход из дополнительного слоя корректен. По дистрибутивному свойству сложения матриц и ассоциативному свойтсву произведения матриц:

$$\hat{\mathbf{y}} = \text{softmax}\left(W_{upd}^T \mathbf{x}\right) = \text{softmax}\left((W + \Delta W)^T \mathbf{x}\right) =$$

$$= \frac{\exp\left(W^T\mathbf{x} + \Delta W^T \mathbf{x}\right)}{\sum_{i=1}^{k} \exp\left(W^T\mathbf{x} + \Delta W^T \mathbf{x}\right)_i} = \tag{3.21}$$

$$= \frac{\exp\left(W^T\mathbf{x}\right)\exp\left(\Delta W^T \mathbf{x}\right)}{\sum_{i=1}^{k} \exp\left(W^T\mathbf{x}\right)_i \exp\left(\Delta W^T \mathbf{x}\right)_i},$$

где $\mathbf{x}$ — выходная матрица последнего слоя BERT. $\mathbf{x}$ корректна по условию. В предложенной модели с использованием LoRA:

$$\hat{\mathbf{y}} = \text{softmax}\left(W_{upd}\mathbf{x}\right) = \text{softmax}\left((W + AB)^T \mathbf{x}\right) =$$

$$= \frac{\exp\left(W^T\mathbf{x} + (AB)^T \mathbf{x}\right)}{\sum_{i=1}^{k} \exp\left(W^T\mathbf{x} + (AB)^T \mathbf{x}\right)_i} = \tag{3.22}$$

$$= \frac{\exp\left(W^T\mathbf{x}\right)\exp\left((AB)^T \mathbf{x}\right)}{\sum_{i=1}^{k} \exp\left(W^T\mathbf{x}\right)_i \exp\left((AB)^T \mathbf{x}\right)_i},$$

где $\mathbf{x}$ также выходная матрица BERT с LoRA и $\mathbf{x}$ корректна по условию.

Так как финальные размерности остались неизменными, как и $W^T\mathbf{x}$, легко заметить:

$$\begin{aligned} \underset{(k\times d)}{\Delta W^T} \times \mathbf{x} &= u \\ (\underset{(d\times r)}{A} \times \underset{(r\times k)}{B})^T \times \mathbf{x} &= (\underset{(k\times r)}{B^T} \times \underset{(r\times d)}{A^T}) \times \mathbf{x} = u^*. \end{aligned} \tag{3.23}$$

Так как (3.22), можно заключить что $u^* = u$ и является корретной матрицей, а следовательно предложенная модель работает корректно. ■

4 Вычислительный эксперимент

Открытый исходный датасет для мультиклассовой классификации текстов, написанных человеком и различными языковыми моделями [33]. Изначально было представлено 6 классов включая человека, но так как обучение на большем датасете требовало больше ресурсов, а основная задача сравнителного анализа - оценить качество предложенного метода при классификации человек vs языковая модель, то количество классов было сокращено до 4: ChatGPT, Davinci, Cohere, Humans. Это не повлияло на выводы, сделанные в данной работе.

Всего в датасете 47327 текстов с разметкой по классам. Средняя длина текста по всему датасету — 400 слов, средняя длина текстов в зависимости от класса представлена в таблице 3. Средняя длина слова — 5 символов. Вес каждого класса — безразмерная величина, показывающая насколько несбалансированна выборка и к каким классам применять большие веса. Статистика по весам классов приведена в таблице 2.

имя класса	**вес, б/р**
chatGPT	0.986
cohere	1.043
davinci	0.986
human	0.986

Таблица 2: Вес каждого класса

имя класса	**длина текста, слова**
chatGPT	362
cohere	279
davinci	343
human	607

Таблица 3: Средняя длина текста

Заметим, что текст, написанный человеком, в среднем в 2-3 раза длиннее написанного машиной. При этом средняя длина слова для

каждого класса одна и та же — 5 символов. Тексты были токенизированы при помощи RobertaTokenizer [25], токенизатора использующего пары байтов для кодировки (англ. Byte-Pair Encoding) [34]. Дополнительная обработка не требуется из-за структуры модели [27]. Для всех экспериментов использовалась предобученная модель DistilRoBERTa base [35]. Модель использовалась со следующими гиперпараметрами: доля тренировочного/тестового набора данных — 0.9/0.1; 3 эпохи обучения. Для экспериментов с использованием алгоритма LoRA были использованы все вышеуказанные параметры, а также ранг матриц аппроксимации $r = 5$.

После обучения для оценки использовались матрица ошибок и метрики точности, полноты, F1-меры, а также для визуализации ошибки использовалась матрица ошибок (англ. Confusion matrix) как наиболее точно отображающие качетво моделей мультиклассовой классификации [36]. В матрице ошибок по вертикали указаны истинные метки классов, а по горизонтали — предсказанные.

4.1 Предобученная модель DRoBERTa-base, мультиклассовая классификация.

Результаты обучения данной модели представлены в таблице 4. Для наглядного описания качества модели используется матрица ошибок: таблица 5. Модель показала высокое качество на уровне 99%, но по-

имя класса	**precision**	**recall**	**f1-score**
chatGPT	1.000	0.993	0.997
cohere	0.963	0.999	0.981
davinci	0.986	0.996	0.991
human	0.991	0.952	0.971

Таблица 4: Метрики качетва DRoBERTa-base

требовала много вычислительных ресурсов: время обучения заняло: 4041.3188 секунд. Построим модель с использованием LoRA, чтобы ускорить обучение и уменьшить ресурсозатратность.

	chatGPT	Cohere	Davinci	Human
chatGPT	**0.993**	0.002	0.0	0.005
Cohere	0.0	**0.999**	0.0	0.001
Davinci	0.0	0.001	**0.996**	0.003
Human	0.0	0.035	0.013	**0.952**

Таблица 5: Матрица ошибок, DRoBERTa-base

4.2 Предобученная модель DRoBERTa-base & LoRA, мультиклассовая классификация.

Только 0.828% параметров обучаются при использовании LoRA: *trainable params: 685828, all: 82807304 // trainable%: 0.8282*, а затраченное время обучения — 3210.977 секунд, по сравнению с 4041.3 из первого эксперимента. Но качество модели заметно упало. Это наиболее заметно в классе human: recall составил всего 31,7%, что отображено в таблице 6. Матрица ошибок также демонстрирует разкое ухудшение метрик;матрица ошибок для данноого эксперимента представлена в таблице 7.

model	**precision**	**recall**	**f1-score**
chatGPT	0.997	0.786	0.879
cohere	0.667	0.940	0.780
davinci	0.703	0.971	0.816
human	0.717	0.317	0.440

Таблица 6: Метрики качетва DRoBERTa-base & LoRA

Метрики качества заметно упали. В большей степени для класса human, который представляет наибольший интерес: в первую очередь важнее всего классифицировать human vs ai, и если текст написала языковая модель, то определять какая. В следующем эксперименте проверим гипотезу: предполагаемая скорость обучения сохранится, а метрики качества вырастут.

истинные метки \ предсказаные метки	chatGPT	Cohere	Davinci	Human
chatGPT	**0.79**	0.01	0.08	0.12
Cohere	0.0	**0.94**	0.06	0.003
Davinci	0.001	0.03	**0.98**	0.0
Human	0.002	0.43	0.25	**0.32**

Таблица 7: Confusion matrix,
DRoBERTa-base & LoRA

4.3 Три независимые модели DRoBERTa-base & LoRA, бинарная классификация.

ChatGPT vs Human

Эксперимент, представленный здесь, аналогичен предыдущему, но три независимые модели решают задачи бинарной классификации, а потом их результат успедняется. В первой части этого эксперимента рассмотрим классы chatGPT и human. Время обучения сократилось в несколько раз, как и предполагалось, и составило 1633.8114 секунд. Метрики качества также выросли и в среднем составили 95%. Особенно хоршие результаты показаны на классе human — до 100%. Результаты представлены в таблицах 8, 9.

время обучения: 1633.8114 секунд

model	**precision**	**recall**	**f1-score**
chatGPT	1.000	0.891	0.942
human	0.902	1.000	0.950

Таблица 8: Метрики качетва DRoBERTa-base & LoRA,
chatGPT vs Human

Cohere vs Human

Аналогично, заметен резкий рост в качетсве, в особенности для класса human. Также время обучения значительно сократилось и составило 1583.556 секунд. Выводы аналогичны первой части данного эксперимента. Результат представлен в таблице 10 и матрица ошибок

	chatGPT	Human
chatGPT	**0.892**	0.108
Human	0.0	**1.00**

Таблица 9: Матрица ошибок,
DRoBERTa-base & LoRA, chatGPT vs Human

представлена в таблице 11.

время обучения: 1583.556 секунд

model	**precision**	**recall**	**f1-score**
cohere	0.999	0.837	0.911
human	0.853	0.999	0.920

Таблица 10: Метрики качетва DRoBERTa-base & LoRA,
Cohere vs Human

	Cohere	Human
Cohere	**0.837**	0.163
Human	0.001	**0.999**

Таблица 11: Матрица ошибок,
DRoBERTa-base & LoRA, Cohere vs Human

Davinci vs Human

Наблюдения аналогичны предыдущим в рамках эксперимента. Время обучения составило 1632.395 секунд. Результат эксперимента представлен в таблице 12 и матрица несоответсвий представлена в таблице 13. Если усреднить показатели трех моделей эксперимента, то можно заметить улучшение качества по сравнению с метриками качетва DistilRoBERTa base & LoRA для мультиклассовой классификации; для класса human recall составил 99%, для других классов особенно выросла метрика precision — до 99% в среднем, в сравнении с 97% из первого эксперимента, где к модели DistilRoBERTa base не применялся LoRA адаптер. Результат представлен в таблицах 14 и 15.

время обучения: 1632.395 секунд

model	precision	recall	f1-score
davinci	0.996	0.851	0.918
human	0.870	0.997	0.929

Таблица 12: Метрики качетва DRoBERTa-base & LoRA, Davinci vs Human

	Davinci	Human
Davinci	**0.852**	0.148
Human	0.003	**0.997**

Таблица 13: Матрица ошибок, DRoBERTa-base & LoRA, Davinci vs Human

model	precision	recall	f1-score
chatGPT	1.000	0.891	0.942
cohere	0.999	0.837	0.911
davinci	0.996	0.851	0.918
human	0.875	0.999	0.933

Таблица 14: Метрики качетва DRoBERTa-base & LoRA, бинарные классификаторы

model	precision	recall	f1-score
chatGPT	1.000	0.993	0.997
cohere	0.963	0.999	0.981
davinci	0.986	0.996	0.991
human	0.991	0.952	0.971

Таблица 15: Метрики качетва DRoBERTa-base, мультиклассовая классификация

Итого, в эксперименте, использующем DistilRoBERTa base & LoRA для бинарной классификации и последующего усреднения, качество классификации выросло, не потеряв во времени обучения, по срав-

нению с предобученной моделью DistilRoBERTa base. А также по результатам эксперимента с использованием DistilRoBERTa base & LoRA для бинарной классификации, модель сильно выиграла в качестве у модели DRoBERTa-base & LoRA для мультиклассовой классификации, но проиграв ей во времени обучения.

5 Заключение

В работе рассматривался метод LoRA снижения размерности пространства обучаемых параметров в задаче классификации текстов, написанных большими языковыми моделями. Была сформулирована и доказана теорема о конструктивности предложенного метода.

В ходе эксперимента на датасете из текстов, написанных как языковыми моделями, так и человеком, была доказана эффективность предложенного метода. При решении задачи мультиклассовой классификации предложенная модель BERT & LoRA тратит меньше ресурсов, чем модель без использования LoRA, но метрики качества падают. Однако, при решении тремя одинаковыми независимыми моделями задачи бинарной классификации с последующим усреднением метрики качетва растут, а использование ресурсов — нет. Таким образом, в данной работе теоритически и экспериментально доказана состоятельность и эффективность предложенного метода.

Список литературы

[1] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[2] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[3] Zubair Qazi, William Shiao, and Evangelos E Papalexakis. Gpt-generated text detection: Benchmark dataset and tensor-based detection method. *arXiv preprint arXiv:2403.07321*, 2024.

[4] Ali Zare, Alp Ozdemir, Mark A Iwen, and Selin Aviyente. Extension of pca to higher order data structures: An introduction to tensors, tensor decompositions, and tensor pca. *Proceedings of the IEEE*, 106(8):1341–1358, 2018.

[5] Valeria Fonti and Eduard Belitser. Feature selection using lasso. *VU Amsterdam research paper in business analytics*, 30:1–25, 2017.

[6] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

[7] Yujia Zhai, Wei Song, Xianjun Liu, Lizhen Liu, and Xinlei Zhao. A chi-square statistics based feature selection method in text classification. In *2018 IEEE 9th International conference on software engineering and service science (ICSESS)*, pages 160–163. IEEE, 2018.

[8] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

[9] G Hinton, O Vinyals, and J Dean. Nips deep learning and representation learning workshop, 2015.

[10] Vladimir Vapnik, Rauf Izmailov, et al. Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, 16(1):2023–2049, 2015.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[12] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[13] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[14] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Openai, gpt-4. `https://openai.com/index/gpt-4/`.

[17] Google, gemini. `https://blog.google/technology/ai/google-gemini-ai/`.

[18] Nash Anderson, Daniel L Belavy, Stephen M Perle, Sharief Hendricks, Luiz Hespanhol, Evert Verhagen, and Aamir R Memon. Ai did not write this manuscript, or did it? can we trick the ai text detector into generated texts? the potential future of chatgpt and ai in sports & exercise medicine manuscript generation, 2023.

[19] Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltỳnek, Jean Guerrero-Dib, Olumide Popoola, Petr Šigut, and Lorna Waddington. Testing of detection tools for ai-generated text. *International Journal for Educational Integrity*, 19(1):26, 2023.

[20] Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*, 2023.

[21] Sara Abdali, Richard Anarfi, CJ Barberan, and Jia He. Decoding the ai pen: Techniques and challenges in detecting ai-generated text. *arXiv preprint arXiv:2403.05750*, 2024.

[22] Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7), 2023.

[23] Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Weiming Zhang, and Nenghai Yu. Gpt paternity test: Gpt generated text detection with gpt genetic inheritance. *arXiv preprint arXiv:2305.12519*, 2023.

[24] Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*, 2023.

[25] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[26] Rukhma Qasim, Waqas Haider Bangyal, Mohammed A Alqarni, and Abdulwahab Ali Almazroi. A fine-tuned bert-based transfer learning

approach for text classification. *Journal of healthcare engineering*, 2022, 2022.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] John Thickstun. The transformer model in equations. *University of Washington: Seattle, WA, USA*, 2021.

[29] Minhyeok Lee. A mathematical investigation of hallucination and creativity in gpt models. *Mathematics*, 11(10):2320, 2023.

[30] Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *Advances in neural information processing systems*, 31, 2018.

[31] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer, 2019.

[32] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[33] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2041–2063, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[34] Changhan Wang, Kyunghyun Cho, and Jiatao Gu. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9154–9160, 2020.

[35] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[36] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

Лемма 3. *Утверждение из статьи [30]: В терминах постаноки задачи классификации текстов: для функции риска* $L(\Theta)$*:*

$$L(\Theta) = \mathbb{E}_{X_i} \mathscr{L}(X_i; \Theta) \tag{5.1}$$

и функции эмпирического риска $\hat{L}(\Theta)$*:*

$$\hat{L}(\Theta) = \frac{1}{\mid \hat{V} \mid} \sum_{X_i \in \hat{V}} \mathscr{L}(X_i; \Theta) \tag{5.2}$$

верно следующее:

$$\sup_{\Theta} \mid L(\Theta) - \hat{L}(\Theta) \mid \leq \delta(\mid \hat{V} \mid) : \tag{5.3}$$

$$\delta(\mid \hat{V} \mid) \xrightarrow[|\hat{V}| \to \inf]{} 0 \tag{5.4}$$