УДК 004.912

Я. С. Пикалёв[1], Т. В. Ермоленко[2]
[1]Государственное учреждение «Институт проблем искусственного интеллекта», г. Донецк
83048, г. Донецк, ул. Артема, 118-б
[2]Государственное образовательное учреждение высшего профессионального образования «Донецкий национальный университет», г. Донецк
83000, г. Донецк, пр. Театральный, 13

АДАПТАЦИЯ НЕЙРОСЕТЕВОЙ МОДЕЛИ ALBERT ДЛЯ ЗАДАЧИ ЯЗЫКОВОГО МОДЕЛИРОВАНИЯ

Ya. S. Pikalyov [1], T. V. Yermolenko[2]
[1]Public institution «Institute of Problems of Artificial intelligence», c. Donetsk
83048, Donetsk, Artema str., 118-b
[2]State Educational Institution of Higher Professional Education «Donetsk National University»
83000, c. Donetsk, Teatralnyiy av., 13

ADAPTATION OF ALBERT NEURAL NETWORK MODEL FOR LANGUAGE MODELING PROBLEM

В статье рассмотрены основные особенности архитектуры глубокой сети ALBERT, ее преимущества и недостатки для использования в задаче языкового моделирования. Предложена модификация сети применения процедуры двухэтапного обучения, использования маски внимания для енкодера при предварительном обучении. Исследования показали, что увеличение количества скрытых слоев и размера скрытого слоя приводит к повышению точности модели, а позиционное кодирование является лишней операцией. Использование AdaptiveSoftmax не влияет на величину пословной ошибки и позволит повысить производительность сети.
Ключевые слова: языковая модель, архитектура Transformer, BERT, ALBERT, маскирование текста, BPE-токенизация

The article discusses the main features of the ALBERT deep network architecture, its advantages and disadvantages for use in the task of language modeling. A modification of the network for applying the two-stage training procedure, using the attention mask for the encoder during preliminary training, is proposed. Studies have shown that an increase in the number of hidden layers and the size of a hidden layer leads to an increase in the accuracy of the model, and positional encoding is an unnecessary operation. Using AdaptiveSoftmax does not affect the word error and will improve network performance.
Key words: language model, Transformer architecture, BERT, ALBERT, text masking, BPE-tokenization.

Введение

Экспоненциальный рост доступной информации в текстовом виде, накапливаемой в различных базах знаний, вызвал необходимость построения систем анализа и обработки этой информации, спрос на которые растет в связи со стремительным развитием интернета, с одновременным взрывным ростом социальных сетей и рынка мобильных устройств. Практически во всех программных решениях, связанных с обработкой текста, используется языковая модель. Языковая модель – неотъемлемая часть систем распознавания речи, OCR-систем, систем машинного перевода, проверки орфографии, предикативного ввода и др.

Задача статистического моделирования языка состоит в определении вероятностного распределения над цепочками слов в некотором языке. Главным подходом к ее решению до недавнего времени оставался подход, основанный на сглаженных n-граммных моделях. Однако с начала 2010-х стал стремительно развиваться подход, основанный на рекуррентных нейронных сетях. Новой переломной точкой для развития моделей машинного обучения, направленных на решение задач обработки текста, стал 2018 год, что связано с появлением архитектуры Transformer, на которой основана одна из последних разработок – модель BERT.

Для флективных языков, к которым относится русский, при непосредственном применении нейросетевых моделей возникает проблема разреженности данных и большой вычислительной сложности в связи со свободным порядком слов и большим количеством грамматических форм.

Целью данной работы является создание нейросетевой модели путем модификации модели ALBERT, позволяющей формировать информативные признаки текста с учетом контекста, которые содержат знания о лингвистической структуре.

Описание модели ALBERT

Модель ALBERT [1] основана на модели BERT [2], которая в свою очередь основана на модели Transformer [3]. ALBERT использует следующие механизмы.

1. Языковое моделирование с использованием масок. Языковое моделирование позволяет предсказывает следующее слово, используя контекст (предыдущий набор слов).
2. Архитектура Transformer.
3. Межслойное использование общих весовых параметров.
4. Факторизация векторных представлений слов.

Языковое моделирование с использованием масок позволяет случайным образом маскировать слова в документе и пытается предсказать их, основываясь на окружающем их контексте. Пример маскирования приводится в таблице 1.

Таблица 1. Пример маскирования входного текста

Текст без маскирования	«президент России Владимир Путин поручил правительству РФ к пятнадцатому декабря представить предложения'об изменении законов с учетом поступивших но не внесенных в Конституцию поправок»
Текст с маскированием	«президент России MASK MASK MASK правительству РФ к пятнадцатому декабря представить предложения'об изменении законов с учетом поступивших но не внесенных MASK MASK MASK»

В основе ALBERT лежит ***архитектура Transformer*** – в качестве енкодера/декодера используется группа слоев блока трансформера (рис. 1).

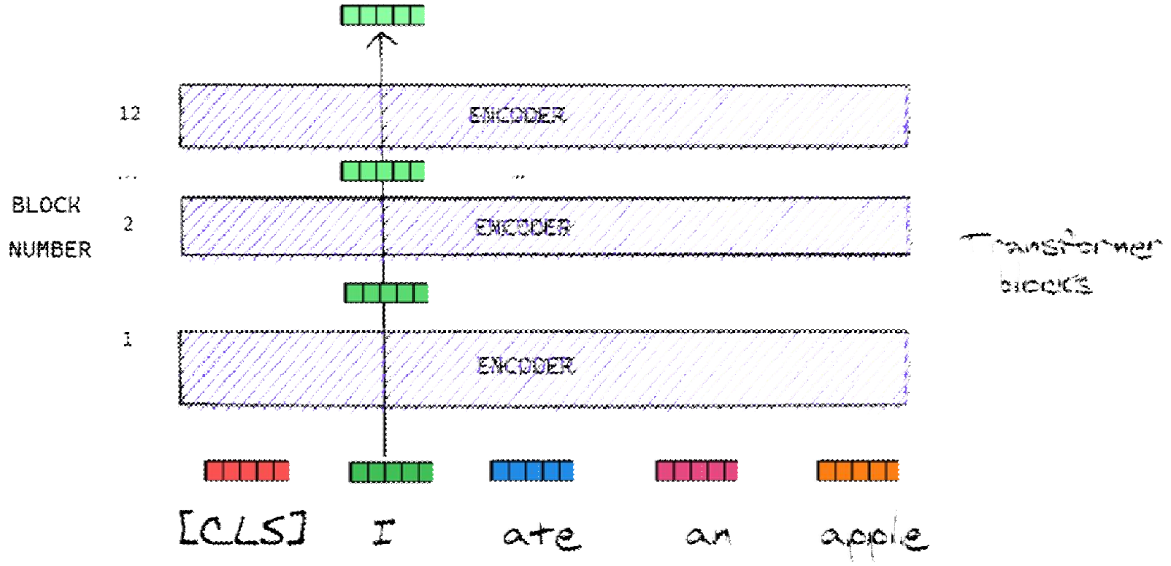

Рисунок 1 – Изображение обобщенной архитектуры трансформер, состоящей из 12 блоков енкодера

Данная модель имеет следующую целевую функцию

$$x_{l+1,i} = LayerNorm(x^3_{l,i} + x^5_{l,i}),$$
$$x^1_{l,i} = LayerNorm(x_{l,i}),$$
$$x^2_{l,i} = MHA(x^1_{l,i}, [x^1_{l,1}, \dots, x^1_{l,N}]),$$
$$x^3_{l,i} = x_{l,i} + x^2_{l,i},$$
$$x^4_{l,i} = LayerNorm(x^3_{l,i}),$$
$$x^5_{l,i} = FFN(x^4_{l,i}),$$
$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V,$$

где Q, K, V – вектора из скрытых слоёв, на вход которых подаётся множество, извлеченное из матрицы векторных вложений токенов

$$MHA(Q,K,V) = concat(h_1, \dots, h_N)W^O$$
$$h_N = Attention(QW^Q_N, KW^K_N, VW^V_N),$$

где $W^Q_N \in R^{d \times d_N}, W^K_N \in R^{d \times d_N}, W^V_N \in R^{d \times d_V},$

$$FFN(h_N) = GeLU(h_N W^1 + b^1)W^2 + b^2,$$

где GeLU – функция активации в скрытых слоях [4]:

$$LayerNorm(v) = \gamma\frac{v-\mu}{\sigma} + \beta,$$

где β – вектор байесовских значений, γ – коэффициент нормализации

$$\mu = \frac{1}{d}\sum_{k=1}^{d} v_k, \quad \sigma^2 = \frac{1}{d}\sum_{k=1}^{d}(v_k - \mu)^2.$$

Межслойное использование общих весовых параметров используется для упрощения модели. Модель BERT-large состоит из 24 слоёв, а BERT-base – из 12. Сложность модели растёт экспоненциально с ростом количества слоёв. Характеристики этих моделей приведены в табл. 2.

Таблица 2 – Характеристики различных видом модели BERT

Тип модели	Количество нейронов в скрытых слоях	Количество скрытых слоёв	Количество параметров, млн
BERT-large	1024	24	340
BERT-base	768	12	110

Согласно концепции межслойного использования весовых параметров, ALBERT вместо изучения уникальных параметров для каждого из слоёв изучает лишь параметры для первого слоя и использует его совместно с оставшимися 11 слоями.

Межслойное использование параметров может применяться лишь для слоёв прямого распространения. Мы можем совместно использовать параметры либо только для уровня прямой связи (FFNN), либо только для параметров внимания, либо совместно использовать параметры всего блока. По сравнению со 110 млн параметров для BERT-base, ALBERT имеет лишь 31 млн с таким же количеством слоёв. Таблица 3 демонстрирует эффект от применения данной концепции.

Таблица 3 – Характеристики модели ALBERT

Модель		Количество параметров, млн.	Средняя точность, %
Albert	All-shared	31	79.8
	Shared-attention	83	81.6
	Shared-ffn	57	79.5
	Not-shared	108	82.3
Albert-base, E=128	All-shared	12	80.1
	Shared-attention	64	81.7
	Shared-ffn	38	80.2
	Not-shared	89	81.6

Поскольку размер матрицы векторных представления слов, используемых в BERT, связан с размером словаря и скрытых слоёв в блоке трансформера, то с увеличением количества скрытого слоя необходимо увеличить и размерность векторного представления слов. ALBERT решает эту проблему при помощи ***факторизации векторного представления слов*** на две матрицы меньшего размера. Эта процедура позволяет использовать отдельно значения размера скрытых слоёв от размера словаря, что позволяет увеличивать размер скрытых слоёв без увеличения размерности ембединга словаря.

Векторные представления в BERT обучаются непосредственно из представления словаря. Они проецируются напрямую в пространство скрытого слоя. Например, есть словарь размером в 30 тыс., а размер векторного представления E=100 и размер скрытого слоя H=768 (рис. 2 а). В ALBERT кодированные вектора проецируются в меньшую размерность векторного пространства E, затем это пространство используется с пространством скрытых слоёв H (рис. 2 б).

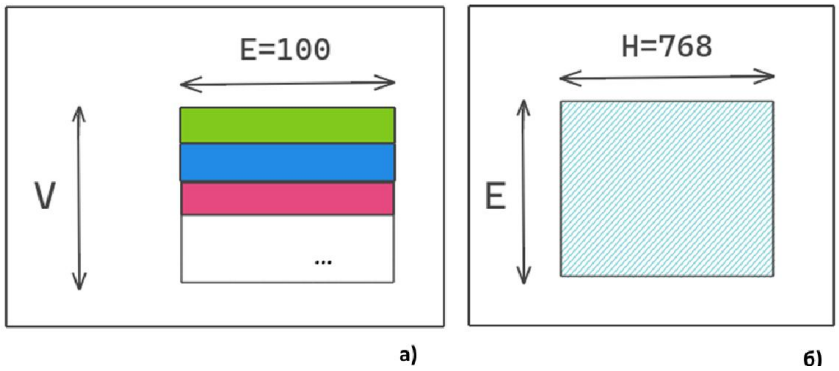

Рисунок 2 – Матрица векторных представлений токенов: а) исходная матрица векторных представлений токенов, размером ExV, где V – размер словаря, а E – размер тензора/слоя векторных представлений; б) факторизованная матрица векторных представлений токенов, размером ExH, где H – размерность скрытого слоя

Преимущества и недостатки ALBERT

ALBERT обладает следующими преимуществами.

1. В 18 раз меньше параметров, чем в BERT-large.
2. Обучается в 1.7 раз быстрее.
3. Показывает SOTA-результаты на известных оценочных наборах данных (GLUE, RACE и SQUAD).

Из недостатков стоит выделить следующие.

1. Токен маскирования [MASK], используемый в предварительной обучении, не используется для процесса fine-tuning. Предобученные модели ALBERT и BERT предназначены для восстановления маскированных токенов. Проблема состоит в том, что данный токен не используется при процессе fine-tuning.

2. Токены, которые не заменены на [MASK], не трансформируются, а просто копируются на вывод. Модели по-прежнему необходимо накапливать стек информации обо токенах в текущей последовательности, чтобы «безопасно» использовать [MASK]. Токен [MASK] является источником ухудшения показателей при тестировании, который может вызвать проблемы во время процедуры fine-tuning. Данную проблему учёные пытались обойти, заменяя некоторые токены случайными реальными токенами во время обучения вместо их замены токеном [MASK]. Однако это составляло только 10% шума. Когда только 15% токенов зашумлены для начала, это составляет только 1,5% всех токенов, так что это полностью не решает проблему.

3. ALBERT генерирует предсказания независимо. Другая проблема связана с тем, что ALBERT прогнозирует замаскированные токены параллельно. Проиллюстрируем это на примере. Предположим, у нас есть следующее предложение: «*Я поехал в* [MASK] [MASK] *и увидел* [MASK] [MASK] [MASK].» Один из возможных способов заполнить это поле: «*Я поехал в Нью-Йорк и увидел Эмпайр Стейт билдинг*». Другой способ «*Я поехал в Сан-Франциско и увидел мост Золотые Ворота*». Однако предложение «*Я поехал в Сан-Франциско и увидел Эмпайр Стейт Билдинг*» будет неверным. Несмотря на это, модель прогнозирует все замаскированные позиции параллельно, что означает, что во время обучения она не учится обрабатывать зависимости между прогнозированием одновременно замаскированных токенов. Другими словами, она не изучает зависимости между своими собственными предсказаниями. Поскольку ALBERT фактически не используется для предсказания токенов, это не является проблемой. Причина, по которой это может стать проблемой,

заключается в том, что это уменьшает количество зависимостей, которые модель изучает одновременно, делая обучающий сигнал слабее, чем он мог бы быть.

Стоит отметить, что ни одна из этих проблем не присутствует в традиционных языковых моделях. В языковых моделях нет токена [MASK], и они генерируют все слова в указанном порядке, поэтому они изучают зависимости между всеми словами в предложении.

Применение модели ALBERT – важный шаг на пути к созданию языковых моделей, которые ставят перед собой не только получение SOTA-результатов, но и подходят для применения в реальном времени без затрат на мощные GPU-сервера.

Подготовка текстовых данных для обучения модели

В качестве текстового материала были использованы: Kaggle Text Normalization Dataset; текстовые расшифровки TedTalks; корпус диалогов из Яндекс.Толоки; субтитры из корпуса OPUS; корпус Tattoeba; корпуса новостных лент и журналов; дамп википедии.

Алгоритм предварительной обработки текста состоял в следующем.

1. Удаление всех символов, не входящих в латинский и русский алфавиты (кроме символа «-«(дефиса)).
2. Деление текста на предложения с приведением первого символа предложения к нижнему регистру. Общая длина предложения должна быть не менее 10 символов и не более 100 символов.
3. Деление предложения на токены (количество токенов не менее 12).
4. Удаление лишних пробелов, а также знаков табуляции и т.п.
5. Предложения записываются в результирующий файл.
6. В файле предложения сортируются по длине, удаляются дубликаты. Окончательный файл именуется bpe_data.
7. В файле bpe_data перемешиваются данные, которые делятся на обучающий и тестовый наборы в отношении 95/5.

В данной работе использовалась BPE-токенизация [5]. BPE (Byte Pair Encoding) – это восходящий алгоритм токенизации подслов, который изучает словарь подслов определенного размера (размер словаря является гиперпараметром). Идея алгоритмов токенизации подслов в том, что более частым словам следует присваивать уникальные идентификаторы, тогда как менее частые слова следует разбивать на единицы подслов, которые лучше всего сохраняют свое значение.

Для BPE-токенизации в данной работе использовались следующие индексы для особых меток:

<MASK> – замаскированный токен, id=size_voc + 1;

<ENG> – токен относится к языку, отличному от русского, id = size_voc + 2;

<NUM> – токен для маскирования чисел и цифровых частей цифро-буквенных комплексов;

<PAD> – токен паддинга, как правило нулевые значения для заполнения вектора до размера N, id=0;

<BOS> – начало фразы id=1;

<EOS> – конец фразы, id=2.

Дополнительно словарь слов дополняется единичными символами нижнего и верхнего регистра русского и латинского алфавита. Полученные словари сохраняются на дисковом пространстве, формируются файлы для конвертации токенов в индексы, а также для конвертации индексов в токены.

В основе модели кодирования данных используется модель ALBERT.

Адаптация модели ALBERT для задачи языкового моделирования

В предлагаемой реализации модели ALBERT для задачи языкового моделирования удалены слои SOP – Sentence order prediction (предсказания порядка следования предложений во фрагменте текста), а также изменены входные ембеддинги за счет двухэтапного обучения. На первом этапе совершается предварительное обучение, после чего – тонкая настройка модели. Ембеддингами при тонкой настройки являются предобученные векторные представления и слой енкодера модели предварительного обучения.

В качестве входных данных для предварительного обучения используется замаскированная последовательность токенов – X_mask. В качестве выходных данных используются позиции и метки замаскированных слов.

Отличиями данной модели являются использование маски внимания для енкодера, а также использование входных данных и меток без позиционных сдвигов.

Предобученные векторные представления и слой енкодера поступают на вход для процесса тонкой настройки на модель с той же архитектурой и гиперпараметрами. Отличием данной модели является использование позиционного сдвига для входных данных (на +1) и для выходных данных на (-1).

Для получения нормализованной вероятности для каждого токена w необходимо провести операцию softmax, которая предполагает суммирование всех юнитов по величине словаря, что повышает вычислительную сложность. Для повышения производительности был использован слой Adaptive Softmax Layer [6]. Архитектура данного слоя позволяет снизить вычислительные затраты для операции softmax. Основной принцип AdaptiveSoftmax состоит в разделении исходного словаря на N кластеров, например, на 2 кластера: V_i и V_j. Пусть k_i, k_j – мощность кластеров, а $p_j = \sum_{w \in V_j} p(w)$ – распределение вероятности для V_i и V_j. Следовательно:

$$p_{i+j} = p_i + p_j,$$
$$p_j\, k_j = (p_{i+j} - p_i),$$
$$p_i\, k_i + p_j\, k_j = p_i\, (k_i - k_j) + p_{i+j} + k_j.$$

Стоит отметить, что для задачи языкового моделирования:

$$w{=}w[bs, ss[1]\ldots ss[n]];$$
$$h = h[bs, hs[0]..hs[HS\text{-}1], ss[0]..ss[SS\text{-}1]],$$

где bs – размер батча, hs – массив последовательности в скрытом слое (HS – размер скрытого слоя), ss – массив текстовой последовательности (SS – размер текстовой последовательности).

В данной работе применялась следующая функция оптимизации:

$$f_{opt}(w) = scale(Lookahead(LAMB(w))),$$

где $LAMB$ – оптимизатор LAMB [7], $Lookahead$ – оптимизатор Lookahead [8], $scale$ – функция скалирования градиента с использованием mixed-precision обучения – одной из новых разработок NVIDIA [9].

Для решения проблемы «взрывных градиентов» использовался известный метод градиентного отсечения, суть которого состоит в обрезании градиентов или установке пороговых значений до максимального значения, что приводит к предотвращению экспоненциального роста градиентов и переполнения (равенство градиентов нулю), или превышения крутых обрывов в функции оценивания.

Результаты численных исследований

Полученная языковая модель использовалась в системе распознавания речи. Оценка качества модели языка, предназначенной для ASR-системы, производится либо измерением перплексии (PPL) на тестовой выборке, либо непосредственным измерением изменения уровня пословной ошибки (WER) в эксперименте по распознаванию. Для оценки эффективности предложенной модели используются следующие метрики.

1. В качестве основной метрики используется точность (accuracy). Вычисляется точность предсказания последовательности закодированных текстовых данных по формуле:

$$accuracy = Nr/N,$$

где Nr – количество верно предсказанных токенов, N – общее количество токенов

2. Уровень пословной ошибки WER, вычисляемый как

$$WER = \frac{S+D+I}{N},$$

где S – количество замен в варианте распознавания, D – количество удалений, I – количество вставок и N – общее число слов.

Для получения WER необходимо иметь полную систему распознавания. WER зависит от настроек конкретной системы распознавания, поэтому оценка эффективности языковой модели опосредованно зависит от качества внешнего по отношению к ней акустического модуля. Однако для сравнения различных языковых моделей на одной системе этот показатель является репрезентативным.

3. Перплексия, которая в экспоненциальной форме выглядит следующим образом:

$$PPL(T) = 2^{loss_{cse}(\theta)}.$$

Выражение, стоящее в показателе степени, называется кросс-энтропией и является эквивалентным показателем качества модели. Кросс-энтропия для входных ембедингов (ембедингов, полученных на основе модели для маскированных данных, h) и меток (оригинального текста, w) равна:

$$loss_{cse}(\theta) = \sum_i \sum_j \log P\left(w_j^i \middle| h_j^i; \theta\right),$$

где θ – языковая модель, $P()$ – вероятность предсказания токена w при использовании ембеддинга (контекста) h.

Фактически перплексия является коэффициентом ветвления в графе поиска, где все гипотезы равновероятны. Чем меньше взаимная энтропия и перплексия, тем лучше модель соответствует тестовым данным. Известна положительная корреляция между PPL и WER. Использование PPL как метрики качества языковой модели объясняется большей «чувствительностью» в сравнении с кросс-энтропией [10].

Основные гиперпараметры модели следующие:

- размер вектора представлений слов (E) – E = H;
- количество слоёв декодера и блоков внимания – H/64;
- максимальная длина последовательности – 128;
- количество групп для скрытых слоёв – 1;
- количество групп в multi head attention – 128;
- коэффициент dropout – 0.0;
- техника инициализации весов – усеченное нормальное распределение;
- коэффициент нормализации – 1e-12.

Исследовалась эффективность модели по трем вышеуказанным критериям при различных гиперпараметрах таких как: размер скрытого слоя (H); количество скрытых слоёв; размер BPE-словаря (V). Результаты отображены в табл. 4 – 8.

Таблица 4 – Влияние количества скрытых слоев на эффективность модели ALBERT (H = 768, V = 10119) по критерию accuracy

Количество скрытых слоёв	accuracy, %
1	52.9
3	71.2
6	77.2
12	81.5
24	**82.1**
48	81.8

Из таблицы видно, что увеличение количества скрытых слоев приводит к повышению точности, для 48 слоев точность незначительно падает, что объясняется значительным усложнением сети, вызванным большим количеством общего числа параметров.

Таблица 5 – Влияние размера скрытого слоя на эффективность модели ALBERT по критерию accuracy

Размер скрытого слоя	Количество параметров, млн.	accuracy, %
1024	18	71.2
2048	60	74.6
4096	225	**76.3**
6144	499	74

Из таблицы видно, что с ростом размера скрытого слоя точность растет, но при максимальном размере скрытого слоя общее количество параметров слишком велико, что приводит к ухудшению точности.

Таблица 6 – Влияние размера BPE-словаря на эффективность моделей ALBERT с различной конфигурацией (для Large H=768, для Small one H = 384, для Small two H = 256) по критерию WER

Тип модели	Размер BPE-словаря, тыс.	Количество параметров, млн.	WER, %
Large	25	123.4	**15.6**
Small one	10	14.7	15.67
Small one	5	11.8	15.73
Small two	10	8.9	15.78
Small two	5	6.8	15.79

После анализа полученных результатов очевидно, что размер BPE-словаря оказывает влияние на величину ошибки, но незначительное: увеличение словаря в 2 раза понижает ошибку не более, чем на 0.1%.

Таблица 7 –Влияние использования AdaptiveSoftmax и размера BPE-словаря на эффективность модели ALBERT с конфигурацией Large по критерию WER

Размер BPE-словаря, тыс.	AdaptiveSoftmax	Количество параметров, млн.	WER, %
25	-	120	15.6
25	+		15.58
10	-	98	15.6
10	+		15.6
5	-	92	15.58
5	+		15.64

Как видно из таблицы, использование AdaptiveSoftmax практически не влияет на точность модели, следовательно, использование этой функции активации на последнем слое для повышения скорости работы сети вполне оправдано.

Таблица 8 – Влияние использования позиционного кодирования на эффективность моделей ALBERT с различной конфигурацией по критерию PPL

Количество скрытых слоев	Позиционное кодирование	Количество параметров, млн.	PPL	
			Обучающая выборка	Тестовая выборка
12	+	243	61.8	66.1
	-		**58**	**63.4**
24	+	281	55.6	60.8
	-		**52.7**	**59.2**
42	+	338	51.2	57.7
	-		54.2	**56.8**

Как видно из таблицы, кодирование позиции слова, выполняемое перед первым кодирующим слоем енкодера с целью сохранения информации о последовательности слов в предложении, только ухудшает показатели модели ALBERT, следовательно, эта операция лишняя, что понижает вычислительную сложность сети. Возможность не осуществлять позиционное кодирование сеть с архитектурой ALBERT приобрела благодаря языковому моделированию с использованием масок.

Выводы

Предложена модификация модели ALBERT для задачи языкового моделирования за счет удаления слоев SOP, применения процедуры двухэтапного обучения, использования маски внимания для енкодера при предварительном обучении. Кроме того, для повышения производительности сети в качестве функции активации на последнем слое использовалась AdaptiveSoftmax.

Обучены и протестированы модели с различным количеством скрытых слоёв, числом нейронов скрытого слоя и размером BPE-словаря. Численные исследования показали, что:

- увеличение количества скрытых слоев и размера скрытого слоя приводит к повышению точности, но до определенного размера, начиная с которого заметно возрастает общее число параметров сети;
- размер BPE-словаря оказывает влияние на величину ошибки, но незначительное: увеличение словаря в 2 раза понижает ошибку не более, чем на 0.1%;
- использование AdaptiveSoftmax в качестве функции активации на последнем слое оправдано, т.к. данная функция практически не влияет на величину пословной ошибки;
- кодирование позиции слова в модели ALBERT является лишней операцией, выполнение которой несколько ухудшает показатель перплексии на тестовой выборке.

Создание языковых моделей с использованием архитектуры ALBERT представляется перспективным в силу их применения в реальном времени без затрат на мощные GPU-сервера.

Список литературы

1. Albert: A lite bert for self-supervised learning of language representations [Электронный ресурс] / Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. – 2019. – arXiv preprint arXiv:1909.11942.
2. Bert: Pre-training of deep bidirectional transformers for language understanding [Электронный ресурс] / Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. – 2018. – arXiv preprint arXiv:1810.04805.
3. Attention is all you need [Текст] / Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. // Advances in neural information processing systems. –2017. – Pp. 5998–6008.
4. Hendrycks, D. Gaussian error linear units (gelus) [Электронный ресурс] / D. Hendrycks, & K. Gimpel. – 2016. – arXiv preprint arXiv:1606.08415.
5. Provilkov I. Bpe-dropout: Simple and effective subword regularization [Электронный ресурс] / Provilkov I. – URL : https://arxiv.org/abs/1910.13267.
6. Joulin, A., Cissé, M., Grangier, D., Jégou, H. Efficient softmax approximation for gpus. In International Conference on Machine Learning (2017, July). - pp. 1302-1310.
7. Интуитивное понимание оптимизатора LAMB [Электронный ресурс]. – URL: https:// https://www.machinelearningmastery.ru/an-intuitive-understanding-of-the-lamb-optimizer-46f8c0ae4866/ (дата обращения: 16.10.2020).
8. Michael R. Zhang. Lookahead Optimizer: k steps forward, 1 [Электронный ресурс] / Michael R. Zhang, James Lucas, Geoffrey Hinton, Jimmy Ba. – URL : https://arxiv.org/abs/1910.13267step back https://arxiv.org/pdf/1907.08610.pdf (дата обращения: 16.10.2020).
9. Mixed Precision Training for NLP and Speech Recognition with OpenSeq2Seq [Электронный ресурс]. – URL: https://developer.nvidia.com/blog/mixed-precision-nlp-speech-openseq2seq/ (дата обращения: 16.10.2020).
10. Mikolov T. Statistical Language Models Based on Neural Networks: Ph.D. thesis. – 2012. – P. 129. http://www.fit.vutbr.cz/research/view_pub.php?id=10158.

References

1. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
4. Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.
5. Provilkov I. Bpe-dropout: Simple and effective subword regularization. URL: https://arxiv.org/abs/1910.13267.
6. Joulin, A., Cissé, M., Grangier, D., Jégou, H. Efficient softmax approximation for gpus. In International Conference on Machine Learning (2017, July). pp. 1302-1310.
7. Интуитивное понимание оптимизатора LAMB. URL: https:// https://www.machinelearningmastery.ru/an-intuitive-understanding-of-the-lamb-optimizer-46f8c0ae4866/
8. Michael R. Zhang, James Lucas, Geoffrey Hinton, Jimmy Ba. Lookahead Optimizer: k steps forward, 1. URL: https://arxiv.org/abs/1910.13267step back https://arxiv.org/pdf/1907.08610.pdf (дата обращения: 16.10.2020).
9. Mixed Precision Training for NLP and Speech Recognition with OpenSeq2Seq. URL: https://developer.nvidia.com/blog/mixed-precision-nlp-speech-openseq2seq/.
10. Mikolov T. Statistical Language Models Based on Neural Networks: Ph.D. thesis. 2012. P. 129. http://www.fit.vutbr.cz/research/view_pub.php?id=10158.

RESUME

Ya. S. Pikalyov, T. V. Yermolenko

Adaptation of ALBERT neural network model for language modeling problem

The language model is an integral part of speech recognition systems, OCR systems, machine translation systems, spell checkers, predictive input, etc. Modern language models are based on neural networks, however, in language modeling for inflectional languages, the problem of data sparseness and great computational complexity arises due to free word order and a large number of grammatical forms. The article proposes a modification of the neural network model ALBERT for the problem of language modeling, which allows the formation of informative features of the text taking into account the context.

The article discusses the main mechanisms that the ALBERT architecture uses, which allow predicting words from the surrounding context, significantly reducing the number of parameters without losing accuracy. An implementation of the ALBERT model for a language modeling problem is proposed, in which network training is divided into two stages: preliminary training and fine-tuning. At the same time, the pre-trained vector representations and the encoder layer of the preliminary training model are fed to the input of the fine-tuning model. A feature of the pre-training model is the use of an attention mask for the encoder. To reduce the computational complexity, the normalized probability for each token is obtained using the AdaptiveSoftmax procedure, and the gradient clipping method was used to solve the problem of "explosive gradients". BPE tokenization was used to obtain tokens.

The results of numerical studies have shown that an increase in the number of hidden layers and the size of a hidden layer leads to an increase in accuracy, the size of the BPE dictionary has practically no effect on the magnitude of the error, and positional coding, which is carried out in models based on the Transformer, is an unnecessary operation, since it degrades the quality models by the perplexity criterion.

РЕЗЮМЕ

Я. С. Пикалёв, Т. В. Ермоленко

Адаптация нейросетевой модели ALBERT для задачи языкового моделирования

Языковая модель – неотъемлемая часть систем распознавания речи, OCR-систем, систем машинного перевода, проверки орфографии, предикативного ввода и др. Современные языковые модели основаны на нейросетях, однако при языковом моделировании для флективных языков возникает проблема разреженности данных и большой вычислительной сложности в связи со свободным порядком слов и большим количеством грамматических форм. В статье предлагается модификация нейросетевой модели ALBERT для задачи языкового моделирования, позволяющей формировать информативные признаки текста с учетом контекста.

В статье рассмотрены основные механизмы, которые использует архитектура ALBERT, позволяющие предсказывать слова по окружающему контексту, в разы сокращать количество параметров без потери точности. Предложена реализация модели ALBERT для задачи языкового моделирования, в которой обучение сети разбивается на два этапа: предварительное обучение и тонкая настройка. При этом на вход модели тонкой настройки подаются предобученные векторные представления и слой енкодера модели предварительного обучения. Особенностью модели предварительного обучения является использование маски внимания для енкодера. Для уменьшения вычислительной сложности нормализованная вероятность для каждого токена получается с помощью процедуры AdaptiveSoftmax, а для решения проблемы «взрывных градиентов» использовался метод градиентного отсечения. Для получения токенов использовалась BPE-токенизация.

Результаты численных исследований показали, что увеличение количества скрытых слоев и размера скрытого слоя приводит к повышению точности, размер BPE-словаря практически не оказывает влияние на величину ошибки, а позиционное кодирование, которое осуществляется в моделях на основе Transformer, является лишней операцией, поскольку ухудшает качество модели по критерию перплексии.

Статья поступила в редакцию 08.04.2020.