Нейросетевые модели языка для систем распознавания речи

Чучупал Владимир Яковлевич,
кандидат физико-математических наук, ведущий научный сотрудник Вычислительного центра им. А.А. Дородницына Федерального исследовательского центра «Информатика и управление» РАН.
E-mail: v.chuchupal@gmail.com

Аннотация

Общепринятый подход к языковому моделированию в системах распознавания речи до недавнего времени базировался на использовании статистических *n*-граммных моделей.

Разработка и эволюция нейросетевых языковых моделей (ЯМ) привели к повышению качества языкового моделирования фактически «в разы», до значений, которые недавно представлялись маловероятными. В существенной степени эти результаты обусловлены использованием в нейросетях распределенных или векторизованных представлений слов, где величины расстояний коррелируют с синтаксическим или семантическим сходством между словами.

В статье приводится описание основных видов нейросетевых ЯМ на простых полносвязных и рекуррентных сетях, многослойных рекуррентных сетях с клетками LSTM, свёрточных сетях, моделей кодера-декодера и Трансформера. Приведены их преимущества и недостатки, требования к вычислительным ресурсам и достигаемые значения показателей качества моделей.

Выделены недостатки нейросетевых ЯМ, как относящиеся к моделям, так и более технические, вытекающие из реализаций алгоритмов обучения и распознавания.

Для наиболее перспективных подходов приведено сравнение значений показателей качества на эталонных корпусах текстовых данных. Показано, что более выигрышными на момент написания статьи являются модели на основе сети Трансформера.

Ключевые слова: автоматическое распознавание речи, *n*-граммные статистические модели, модели языка, нейро-сетевые модели языка, сквозные системы распознавания речи, модель внимания.

1. Статистические модели языка

Статистические *n*-граммные модели языка (языковые модели, ЯМ) – одна из основных частей систем распознавания речи, машинного перевода и интеллектуального анализа текстов.

Вероятностная формулировка задачи распознавания слитной речи имеет следующий вид [11]: пусть $X = \{x\_t\}$, $t = 1,...T$, наблюдаемая последовательность параметров сигнала, а $W = \{w\_i\}$, $i = 1,...,N$, некоторая последовательность слов.

Наиболее вероятная последовательность слов W^* для X определяется как:

$$\begin{aligned} W^* &= \arg\max_W P(W|X) \\ &= \arg\max_W \frac{P(X|W)P(W)}{P(X)} \\ &= \arg\max_W P(X|W)P(W). \end{aligned} \tag{1}$$

Вероятность $P(X|W)$ в числителе (1) вычисляется с помощью акустических моделей, а $P(W)$ – с помощью модели языка.

Модель языка определяет вероятность появления в речи заданного набора слов:

$$P(W) - P(w_1, \ldots, w_N) - \prod_{n=1}^{N} P(w_n|w_1 \ldots, w_{n-1}) \tag{2}$$

Статистические языковые модели, построенные с помощью аппроксимации выражения (2) называются n-граммными.

Если w_n – текущее слово, а h_n – его левый контекст $w_1,...,w_{n-1}$ («история»), т.е. предыдущие слова, то модель языка определяет вероятности $p(w|h)$ для всех пар (h,w).

Максимальная длина контекста n в (2) на практике ограничивается некоторым значением m, выбираемым обычно исходя из размера обучающих данных. Таким образом, вероятность $P(X|W)$ вычисляется приблизительно как произведение вероятностей m-грамм:

$$\begin{aligned} W^* &= \arg\max_W P(W|X) \\ & P(X|W)P(W) \end{aligned} \tag{3}$$

Аппроксимация в виде произведения вероятностей n-грамм (3), обрезанных контекстов, является неточной, особенно при небольших значениях n.

Очевидный способ оценки качества языковой модели путём оценки точности распознавания очень вычислительно затратен. Принимая во внимание некоторую независимость языковых и акустических моделей, в (1) качество модели языка обычно оценивается автономно в виде значений перплексии или энтропии корпуса данных.

Энтропия определяется как усредненное значение логарифма вероятности появления всех слов корпуса:

$$H(W_1^N) = \frac{1}{N} \log P(w_1, \ldots, w_N) \tag{4}$$

Перплексия – экспонента энтропии, т.е.:

$$PP(W_1^N) = P(w_1, \ldots, w_N)^{-1/N} \tag{5}$$

Максимально правдоподобная оценка параметров для *n*-граммной модели языка давно известна, это частотная оценка, т.е. если через $N(h)$ обозначить, сколько раз в корпусе данных встретилась последовательность слов h, то:

$$p(w|h) = \frac{N(w,h)}{N(h)} \quad (6)$$

Для неограниченного по размеру корпуса данных *n*-граммные модели при больших значениях n можно было бы рассматривать с учётом простоты их реализации, как оптимальные. На практике даже для малых n значительное число возможных *n*-грамм в обучающем корпусе данных не встречается. Появление при распознавании *n*-грамм с нулевой вероятностью в соответствии с (1) обнуляет полную вероятность распознаваемого высказывания, независимо от результата акустической модели, что очень нежелательно. Аппроксимация вероятностей невидимых *n*-грамм может быть выполнена различными методами перераспределения (далее – сглаживания) вероятностной видимых *n*-грамм между невидимыми. В то же время идеального способа назначения вероятностей для событий, которые не появились в обучающих данных, нет, и статистические модели в принципе неточны. Поэтому даже в относительно больших по объёму текстовых корпусах, например [27], порядок моделей ограничен 4-5, чтобы избежать появления большого количества «невидимых» *n*-грамм.

Один из самых удачных подходов к сглаживанию *n*-граммных моделей, метод Kneser-Ney (далее в тексте эти модели обозначаются как KNn, где n – порядок модели), предложен в работе [18], где вероятности *n*-грамм вычисляются как:

$$\tilde{p}(w|h) = \begin{cases} \frac{N^*(w,h)}{N(h)} & N(w,h) > 0 \\ B_h \beta(w|\hat{h}) & N(w,h) \leq 0 \end{cases} \quad (7)$$

В выражении (7) $N(h)=\sum_w N(w,h)$ означает полный счет (т.е. количество появлений) контекста h, $N^*(w,h)= N(w,h) - D$ – модифицированный счёт, $0<D<1$,$\hat{h}$ – обобщённый контекст, обычно укороченный, $\beta(w|\hat{h})$\$ – обобщённое распределение.

Вес отступа в (7) вычисляется как :

$$B_h = \frac{(\sum_{w:N(w,h)>0}(N(w,h) - N^*(w,h)))}{N(h)} \quad (8)$$

Качественное превосходство сглаживания по Kneser-Ney было показано в целом ряде исследований. В частности, в [21] показано, что на небольшом, из примерно 1.2M слов, корпусе (подмножество корпуса Wall Street Journal, WSJ) в сравнении с оценками для сглаживания по методу Good-Turing величина перплексии снизилась примерно на 12% и затем почти на 25% для модели с памятью (cache):

Таблица 1

Величина перплексии для 5-граммных ЯМ со сглаживанием по Good-Turing (GT5) и Kneser-Ney (KN5) из [21]

Модель	Перплексия
GT5	162,3
KN5	141.2
KN5+cache	125.7

Модель с памятью в табл.1 означает интерполяцию основной модели (KN5) униграммной, которая формируется динамически по предыдущим распознанным словам.

Эффективность, простота и формула с отступом в виде конечного вероятностного преобразователя обуславливают широкое использование модели со сглаживанием по методу Kneser-Ney, на которую часто ссылаются как state-of-the-art статистических языковых моделей.

2. Нейросетевые модели языка

2.1. Недостатки n-граммных моделей

Существенный недостаток статистических *n*-граммных моделей заключается в подходе к оценке вероятностей невидимых *n*-грамм. Такие вероятности оцениваются через вероятности более коротких, частичных *n*-грамм, которые чаще встречаются, но при этом обладают другими вероятностными закономерностями. Альтернативный подход заключается в оценке вероятностей *n*-грамм с учётом, например, семантического сходства между словами. Тогда, если в обучающих данных отсутствует конкретная *n*-грамма, её вероятность можно оценить по вероятностям схожих по семантике *n*-грамм того же размера, которые присутствуют в данных.

Эта идея реализуется в статистических *n*-граммных моделях как классовые модели (class models), где слова словаря разбиваются на подмножества – классы и вычисление вероятности *n*-грамм имеет вид [12]:

$$P(w_n|w_1,\ldots,w_{n-1}) = P(w_n|C_n) \times P(C_n|C_1,\ldots,C_{n-1}) \quad (9)$$

где C_i означает класс, в который попало слово w_i.

Классовые статистические *n*-граммные модели часто используются на практике, например, когда классы соответствуют группам именованных сущностей (названия стран, городов, улиц, наименования произведений, песен, имена авторов). Выбор классов чаще осуществляется на экспертной основе. Оценка (9) – из двух – сомножителей, каждый вносит свою погрешность в итоговую аппроксимацию. Кроме этого, сам подход с бинарной мерой сходства слов по принадлежности их к одному классу существенно ограничивают возможности такой модели.

Нейросетевые ЯМ основаны на непрерывных векторных представлениях слов. В отличие от *n*-грамм, где слово полностью определяется скаляром – индексом в словаре, векторные представления позволяют вводить меры схожести между словами, учитывающие семантические, контекстные, морфологические свойства. Такое представление, как написано выше, существенно улучшает качество моделей языка.

2.2. Модели языка на основе нерекуррентных полносвязных нейросетей

Реализация языковой модели для распознавания речи на полносвязной нерекуррентной нейронной сети была предложена в работе [2]. Схематически архитектура сети имеет вид, изображённый на рис.1:

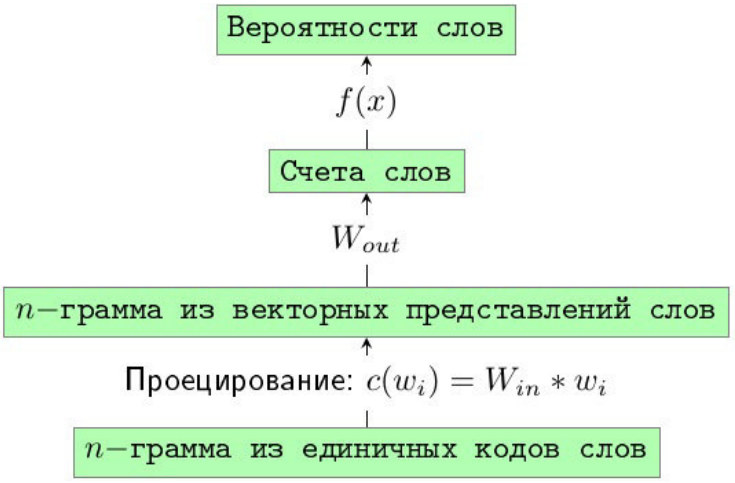

Рис. 1. Архитектура языковой модели на простой нерекуррентной сети

Первый, проекционный, слой сети на рис.1 – линейный, который преобразует |V| мерный единичный код каждого слова *n*-граммы в вектор меньшей размерности с действительными координатами. Следующий слой – полносвязный скрытый, с сигмоидальной нелинейностью *σ(x)*:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

Выходной слой сети сначала вычисляет ненормированные вероятности слов и затем использует нелинейность в виде функции софтмакс:

$$f(w_i) = \frac{e^{w_i}}{\sum_j e^{w_j}} \tag{11}$$

для их преобразования в вероятностное распределение.

Если не учитывать линейный проекционный слой, сеть на рис. 1 имеет всего один скрытый слой. С точки зрения терминологии эту сеть также можно назвать и свёрточной, в зависимости от того, что рассматривать в качестве входа и выхода. Если оценивается вероятность одной *n*-граммы, то это полносвязная сеть, если в качестве входного образа используются *n*-граммы слов всего предложения, то – свёрточная.

Эффективность модели (Рис.1) сравнивалась по показателю перплексии с *n*-граммными [2]: со сглаживанием по Kneser-Ney и со сглаживанием по моделям классов. Тестирование проводилось на корпусе данных с общим размером около 1.2 миллиона слов (0.8M в обучающей выборке, 200K и 181K в тестовой и настроечной) и рабочим словарём 16K словоформ (он получен из общего словаря 47K словоформ включая знаки препинания, заменой редких слов на спецсимвол).

Модель на рис.1 обеспечила понижение перплексии на 13-14% относительно лучшей статистической триграммной модели (это оказалась модель с отступом к классовым моделям). Длина контекста нейросетевой модели была пять слов, размерность скрытого слоя – 100, а размерность векторного представления слов – 30. При этом ошибки, которые допускали нейросетевая и *n*-граммная модели, существенно различались. Поэтому взвешивание с равными весами вероятностей на выходах нейросетевой и *n*-граммной моделей позволило ещё больше (24%) снизить перплексию.

В условиях существенно большего корпуса данных (AP News, 14 млн. слов, по 1 млн словоформ для настроечной и тестовой выборок) авторы из-за ограниченных вычислительных ресурсов получили результаты только одного эксперимента. Там также наблюдалось понижение перплексии (для интерполированных нейросетевой и *n*-граммной), правда, существенно более скромное, до 7%, по сравнению с моделью KN5.

При наличии достаточного объёма данных длина истории в полносвязной нерекуррентной сети обычно выбирается не менее десяти, размерность векторного представления (проекционного слоя) 500-2000, а скрытого слоя 500-1000 элементов. Таком образом, нейросетевая модель использует существенно более длинный контекст, чем *n*-граммные. Например, на корпусе [21] обычно используется 4-граммная языковая модель.

2.3. Рекуррентные модели языка

Рекуррентные сети теоретически используют неограниченный по длине контекст и поэтому натурально подходят для использования в качестве основы вероятностной модели языка.

Языковая модель на основе простой рекуррентной сети Элмана (simple recurrent network) [9] была исследована в [3] и до сих пор успешно применяется [19,15].

Архитектура языковой модели на простой рекуррентной сети представлена на рис.2.

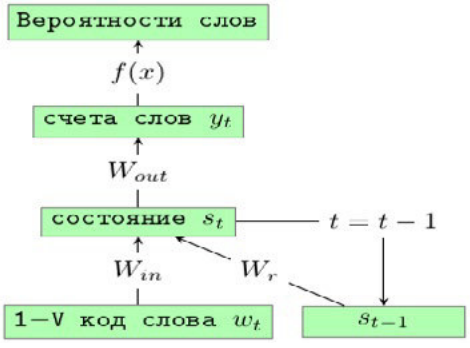

Рис. 2. Схема языковой модели на простой рекуррентной сети

Формально вычисление распределения вероятностей для слов словаря *p(w_{t+1})* в момент времени *$t+1$* осуществляется в соответствии с выражениями:

$$s_t = \sigma(W_{in} * w_t + W_r * s_{t-1})$$
$$p(w_{t+1}) = f(W_{out} * s_t)$$

с сигмоидальной (10) и софт-макс (11) нелинейностями σ и *f*.

Слово на входе сети кодируется унитарным вектором, с длиной кода равной размеру словаря |V|. Размер скрытого слоя выбирается с учётом корпуса данных, например в диапазоне 30-500 элементов. Эффективная длина контекста зависит от типа элементов сети и линейностей, при использовании клеток LSTM она может быть больше 100 слов.

Даже для сети из обычных элементов, когда эффективная длина контекста (из-за эффекта затухания градиентов) невелика, рекуррентная модель при наличии достаточного объёма обучающих данных оказывается эффективнее *n*-граммных. В работе [22] на корпусе 6.4M слов (подмножество из 300 тыс. предложений раздела New York Time корпуса English Gigaword) использование рекуррентной модели, как следует из табл. 2, уменьшило перплексию почти на четверть, а интерполяция выходов трёх рекуррентных сетей – более чем на 30%.

Таблица 2

Сравнение показателя перплексии статистических и рекуррентной моделей языка [22]

Модель	Перплексия	Комбинация с KN5
KN5	221	–
RNN 60	229	186
RNN 90	202	173
RNN 250	173	155
RNN 400	171	152
3xRNN	151	143

В то же время для небольших обучающих данных (уже упоминавшееся подмножество корпуса WSJ) в соответствии с табл. 3 из [21], отмечается отсутствие преимущества нейросетевых моделей: статистическая ЯМ со сглаживанием KN5 оказалась существенно лучше нерекуррентной нейросетевой (перплексия ниже на 10%) и примерно такой же по качеству, как рекуррентная. Если учесть высокие вычислительные требования нейросетевых моделей, то они в данном случае хуже, что *n*-граммные. Однако и в этом случае использование интерполированных нейросетевых и *n*-граммных моделей имеет явное (– 20\% перплексии) преимущество. В [22] отмечено, что эти модели могут рассматриваться как дополняющие друга: *n*-граммы лучше оценивают вероятности для известных и коротких контекстов, а нейросетевые модели более точны при оценке вероятностей на длинных контекстах.

Таблица 3

Величина перплексии для 5-граммных ЯМ со сглаживанием по Good-Turing (GT5) и Kneser-Ney (KN5)[21].(GT5) и Kneser-Ney (KN5)[21]

Модель	Перплексия
KN5+cache	125.7
FF LM	140.2−141.8
RNN LM	124.7
RNN LM+KN5	105.7
RNN LM+KN5+cache	97.5

С увеличением размеров корпуса перплексия статистических и нейросетевых ЯМ, как следует из следующей табл. 4, убывает.

Таблица 4

Зависимость перплексии от количества обучающих данных [22]

Модель	200K слов	1M слов	6.4M слов
KN5	336	287 (δ=49)	221(δ = 66)
KN+RNN	271	225 (δ=46)	156(δ = 69)

Для улучшения качества нейросетевой модели при наличии небольших данных в [22] рассматривалось использование дополнительных признаков, вектора текущей тематики. На корпусе данных (уже упоминавшееся подмножество WSJ) рекуррентные модели с дополнительным тематическим вектором признаков оказались почти на 20% (относительного значения) лучше статистической модели KN5.

Таблица 5

Зависимость перплексии от типа ЯМ на небольшом корпусе данных [22]

Модель	Перплексия
KN5	141.2
NNLM 100	142.1
NNLM 100 + topic	126.4
NNLM 300 + topic	113.7
NNLM 300 + topic + KN5	98.3

Несмотря на разработку существенно менее чувствительных к эффектам затухания градиентов моделей рекуррентных сетей на основе сложных клеток LSTM [15] и GRU, простые рекуррентные сети до сих пор активно используются в языковом моделировании, причём показывают передовые результаты, в том числе на задачах с большими объёмами текстов.

В [16] для корпуса Миллиард слов [5] (со словарём в 1 млн слов) приведены сравнительные значения перплексии для рекуррентной ЯМ на клетках LSTM и существенно более вычислительно эффективной простой рекуррентной сети, которая использовала упрощённое вычисление

функции софтмакс. При относительно большем числе элементов (например, в соотношении 512 к 2048) модель на простой сети имела перплексию ниже (68.3 вместо 68.6) сети на клетках LSTM. Только при дальнейшем уменьшении относительных размеров простой сети использование LSTM клеток становилось явно предпочтительнее.

Возможности простых рекуррентных сетей по моделированию достаточно длинных (более 10-15 интервалов) контекстов очевидно ограничены из-за численных особенностей метода обучения. Поскольку возможность учёта длинных зависимостей является очевидным преимуществом рекуррентных сетей, современные рекуррентные модели языка чаще используют слои из клеток LSTM [15] и GRU (упрощённая версия LSTM, которые дают заметный прирост в эффективности). В работе [14] на корпусе Миллиард слов [5] однослойные и двуслойные рекуррентные сети на LSTM клетках обеспечили снижение перплексии на 35 и 41% соответственно по сравнению с результатами, полученными на модели KN5.

Помимо многослойности на качество рекуррентных языковых моделей влияет возможность двунаправленной реализации. В этом случае активации скрытого сети слоя получаются объединением активаций двух скрытых слоев, которые соответствуют рекуррентной обработке сигнала в прямом и обратном направлениях. В режиме работы в реальном времени такая обработка нереализуема, поэтому такие модели ориентированы либо на задачи обработки текстов, либо на распознавание речи с гарантированной задержкой. Замена двунаправленных слоев на односторонние сопровождается заметной потерей качества модели.

2.4. Модели языка на основе кодеров-декодеров с вниманием и трансформеров

Очевидным направлением дальнейшего роста качества нейросетевых моделей языка является использование многослойных архитектур. Такие языковые модели используются в последнее время при разработке сквозных (end-to-end) систем распознавания речи, в которых вся система распознавания речи реализована как одна многослойная нейросеть.

В сквозных системах нет выделенных модулей для реализации акустического, произносительного и языкового моделирования, так как это делается в GMM-HMM или DNN-HMM системах распознавания. С этой точки зрения структура GMM-HMM или гибридных DNN-HMM выглядит более гибкой, поскольку разные уровни обработки речи: акустический, произносительный, лексический и языковой обособлены и представляются в виде конечных преобразователей (именуемых H,C,L,G – преобразователями). Весь процесс распознавания речи представляется как композиция этих преобразователей, преобразователь HCLG. В сквозной архитектуре процесс обработки речи осуществляется в одной нейросети, более того, представление нейросетевой языковой модели в форме конечного преобразователя в общем случае затруднительно.

Языковая модель в традиционной, n-граммной форме вычисляет условные вероятности появления слов w_n в виде $P(W_n \mid W_{\{n-m+1\}}, ..., W_{\{n-1\}})$. Рекуррентная языковая модель вычисляет вероятности слов в форме $P(w_n \mid w_{\{n-1\}}, C_n)$, где контекст C_n – это набор значений активаций элементов скрытого слоя сети: $C_n = C(w_1, w_2, ..., w_{\{n-1\}})$.

В сквозных методах вероятности появления символов вычисляются в форме $P(c_n|c_{\{n-1\}}, C(x_1,...x_N)$ (модель кодера-декодера) или $P\left(c_n \mid C(x_1,...,x_N;c_{\{n-1\}})\right)$ (модель Трансформера), где контекст $C()$ вычисляется нейросетью непосредственно из акустических параметров и предыдущего выходного символа. В качестве единиц для языкового моделирования обычно выбирают не слова, а их кусочки (морфы) или буквы.

Модель кодера-декодера с вниманием (encoder-decoder with attention) изначально была предложена [8] для решения задачи автоматического перевода текстов. Схематически архитектура на примере системы LAS (listen, attend and spell) [4] представлена на следующем рис. 3 \ref{fig:encoder-decoder} Это глубокая рекуррентная сеть со слоями, которых выделены три компоненты, отдельные модели: кодер, внимание и декодер.

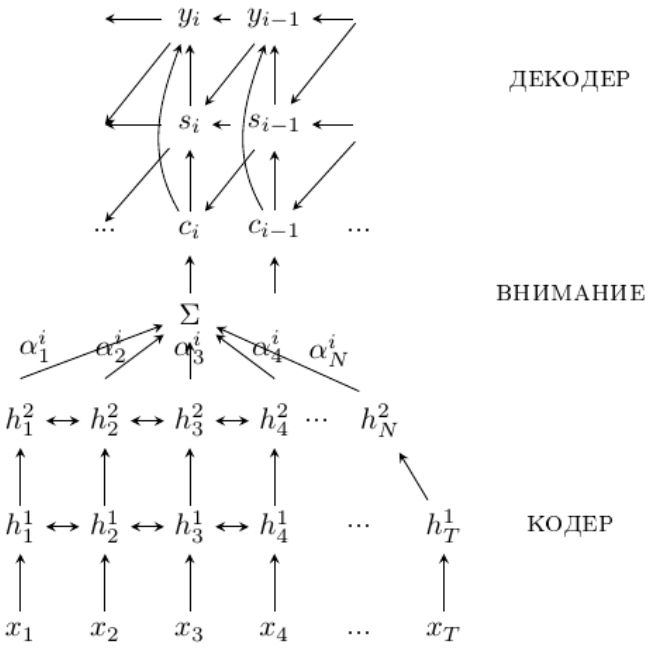

Рис.3. Модель кодера-декодера с вниманием.

Входом сети является последовательность кратковременных параметров речевого сигнала $x_1,...,x_N$, выходом – наиболее вероятная последовательность символов $y_1,...,y_S$

$$\arg\max_y P(y|x) \tag{12}$$

Вероятность (12) (\ref{eq:encdec1}) вычисляется путём аппроксимации \quotes{цепного} правила:

$$P(y|x) = \prod_i P(y_i|y_1,\ldots,y_{i-1},x_1,\ldots,x_T) \tag{13}$$

Кодер, выполненный в виде многослойной рекуррентной трапецивидной сети (размер слоёв уменьшается в 2 раза с каждым слоем), преобразует со сжатием входные параметры x в последовательность производных (высокоуровневых) признаков $h_1, h_2, \ldots, h_N, N \ll T$:

$$h_t = f(x_t, h_{t-1}) \tag{14}$$

LAS использует двунаправленные клетки LSTM так, например, активации элементов первого скрытого слоя вычисляются как:

$$\begin{aligned} h_t^+ &= f(x_t, h_{t-1}) \\ h_t^- &= g(x_t, h_{t+1}) \\ h_t &= [h_t^+, h_t^-] \end{aligned}$$

где h_t^+, h_t^- – активации скрытого слоя в прямом и обратном направлении, h_t результирующий вектор активаций.

Декодер вычисляет решение в виде вектора распределения вероятностей выходных символов, которое ищется в виде:

$$P(y_i|x) = g(y_{i-1}, s_i, c_i) \tag{15}$$

Для определения наиболее вероятной последовательности выходных символов используется алгоритм лучевого (beam search) поиска.

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{16}$$

Модель внимания определяется соотношениями:

$$\begin{aligned} c_i &= \sum_{j=1}^{N} \alpha_{ij} h_j \\ \alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{N} \exp(e_{ik})} \\ e_{ik} &= a(s_{i-1}, h_k) \end{aligned} \tag{17-19}$$

где a – преобразование, выполняемое однослойной полносвязной сетью.

Качество языковой модели LAS сравнить с более традиционными нейросетевыми или статистическими моделями по показателю энтропии или перплексии довольно сложно, поскольку она интегрирована с сетью и, кроме этого, единицами моделирования тут являются не слова, а части слов: буквы либо морфы. Показатели перплексии для таких единиц будут несопоставимы с показателями, которые получены для словных ЯМ. Далее, размер обучающего текстового материала в сквозных моделях весьма ограничен по сравнению с несквозными моделями. Так как вся сеть обучается сразу и вместе, один и тот же речевой материал используется одновременно для оценки параметров акустических, произносительных

и языковых моделей. Размеры даже больших акустических обучающих корпусов с точки зрения текстового покрытия заметно меньше современных текстовых корпусов для оценки параметров языковых моделей. Всё это отражается на качестве встроенной языковой модели и часто заставляет использовать в практических решениях дополнительно к ней внешнюю статистическую *n*-граммную языковую модель.

В эксперименте [4] на корпусе размером в 2 тысячи часов звука использование модели LAS с тремя слоями в кодере (512 элементов на слой) обеспечило уровень пословной ошибки распознавания, что заметно хуже, чем продукционная DNN-HMM система распознавания Google (примерно 2.5% абсолютной разницы или 28.7% относительной). При этом если распознавание осуществлялось с пересчётом вероятностей вершин графа слов с помощью внешней языковой модели (которая обучалась на отдельном корпусе текстовых данных), уровень ошибок снижался на 36% относительно режима с отсутствием такого пересчёта. То есть нейросетевая языковая модель в данном случае была неэффективна. Это объясняется, возможно, очень малым размером обучающих текстовых данных: 2 тысячи часов звука это примерно около 4 миллиона слов.

Увеличение глубины этой сети [7] до пяти слоёв кодера и второго слоя декодера и размера обучающего корпуса (до 12.5 тысячи часов или 15 миллионов фраз) привело к существенному улучшению результатов. Система в этом случае показала заметное превосходство над DNN-HMM продукционной системой распознавания: относительное снижение уровня пословных ошибок составило 18%.

Тем не менее и в этом случае пересчёт вероятностей вершин графа слов с помощью дополнительной внешней статистической модели языка, модели KN5, привёл к снижению пословной ошибки распознавания на 3.4% относительно модели без пересчёта.

Эксперимент с заменой двусторонних сетей на односторонние приводит к заметному снижению точности: уровня ошибки распознавания растёт на 5.6 – 21.6% относительно исходного. При этом, поскольку даже в этом случае для распознавания использовался весь сигнал, это скорее нижняя граница ошибок распознавания при работе в режиме реального времени.

2.5. Модели языка на основе трансформеров

Основной недостаток рекуррентных сетей: невозможность распараллеливания вычислений и вытекающий отсюда большой объем вычислений делают перспективными разработку аналогов модели кодера-декодера с вниманием на основе свёрточных или других вариантов направленных сетей. Примером является модель Трансформера [28], рис. 4, которая в отличие от модели кодера-декодера с вниманием вместо рекуррентных слоёв используют свёрточные слои внимания, самовнимания и поточечные полносвязные слои.

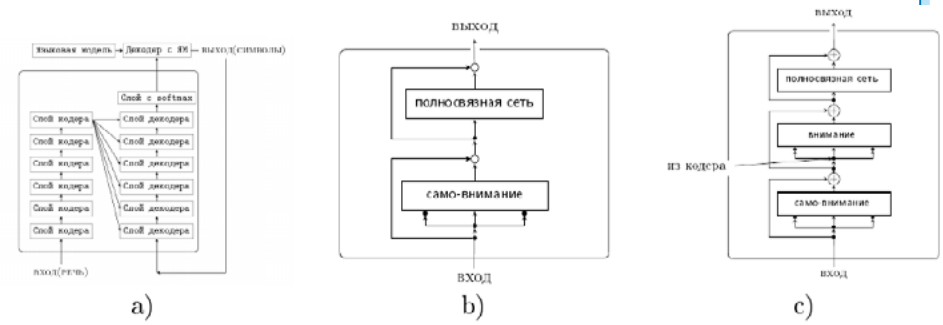

Рис. 4. Модель Трансформера: a) общая структура, b) слой кодера и c) слой декодера

Как видно из рис. 4 Трансформер состоит из двух частей, кодера и декодера, многослойных сетей из идентичных композитных слоёв. Слой кодера (рис. 4b) состоит из подслоя самовнимания (self-attention) и поточечного полносвязного (positional-wise feed-forward) подслоя.

Модель самовнимания, так же как и внимания в (20), вычисляет активацию каждого элемента выходного слоя как средневзвешенное значение активаций элементов предыдущего. При этом значения весов определяются линейно преобразованными значениями активаций элементов входного слоя. Матрицы таких линейных преобразований являются обучаемыми параметрами модели. Если это размерность элементов входного слоя, а $W_k \in R^{\{d\_k \times d\_i\}}$, $W_q \in R^{\{dq \times di\}}$, $W_v \in R^{\{dv \times di\}}$ – проекционные матрицы (ключей, вопросов и значений: «keys, quieries, values», соответственно), то вес t-го элемента определяется как:

$$\alpha_{t,\tau} = SoftMax(\beta \cdot x_t^{'} W_q^{'} W_k x_\tau) \quad (20)$$

где $\beta = \frac{1}{sqrt\{d_i\}}$ – нормирующий множитель, соответственно активация элемента, y_t вычисляется как:

$$y_t = \sum_\tau \alpha_{t,\tau} \cdot W_v x_\tau \quad (21)$$

Размерности d_k, d_q, d_v часто выбираются равными, при этом используется многофокусная (multi--head) модель с h «фокусами» внимания, каждый – с собственным набором проекционных матриц $W_k W_q W_v$. После применения всех фокусов окончательное значение активации элемента y_t рассчитывается как проекция конкатенации для всех фокусов на пространство исходной размерности d_i.

Поточечная свёрточная сеть с одним скрытым слоем имеет ядро единичной длины, т.е. каждый элемент входного слоя x_t преобразуется «в себя», при этом функция преобразования его компонент одна и та же для всех элементов:

$$y_t = ReLU(W_1 x_t + b_1) W_2 + b_2 \quad (22)$$

Число преобразований в (23) – гиперпараметр алгоритма. Например, если их четыре, то при 512-мерных векторах признаков поточечный слой реализует четыре различных преобразования над компонентами каждого вектора признаков.

Выход верхнего слоя кодера соединён со всеми слоями декодера.

Декодер также состоит из композитных слоёв (изображён на рис. 4, каждый включает три подслоя: два из которых (самовнимание и полносвязный) идентичны тем, что в кодере. Первый слой декодера – самовнимание для символов, которые поступают с выхода сети. Следующий слой, аналога которого нет в кодере, – подслой внимания для входных признаков, приходящих из кодера. Это обычный слой внимания, в качестве запросов использованы выходы предыдущего слоя декодера, а ключами и значениями являются выходные активации кодера.

Функционирование трансформера при распознавании аналогично функционированию сети кодера-декодера. На вход поступает речевое высказывание в виде последовательности векторов из параметров. Эта последовательность является входом кодера, то есть активации элементов на его выходе вычисляются по всему сигналу и остаются неизменными в течение декодирования. Декодирование осуществляется посимвольно, очередной выходной символ декодера добавляется в n-грамму его входных символов слева. Выходом декодера служит набор вероятностей для всех слов или символов. Он должен быть на каждом шаге пересчитан для определения наиболее вероятного символа. Для этого обычно пересчитываются вероятности символов в специальной структуре – решётке символов. Для такого пересчёта (как и в LAS) часто используются внешние языковые модели, как статистические n-граммные, так и нейросетевые, качество которых существенно влияет на точность получаемых результатов.

Использование модели самовнимания в трансформере приносит вычислительные преимущества по сравнению с обычными свёрточными или рекуррентными вариантами слоёв. Хотя количество операций, приходящихся на каждый слой внимания, может быть больше, чем в рекуррентных сетях (вычислительная сложность при использовании само-внимания, рекуррентного и свёрточного слоев оценивается как: $O(n^2d)$,$O(nd)$, $O(knd^2)$[28]), где n – длина входной последовательности, d – размерность каждого элемента, k – длина ядра свертки), для самовнимания и свёрточных слоев, в отличие от рекуррентных, можно обеспечить полное распараллеливание вычислений и тогда число операций не зависит от параметра n, в то время как для рекуррентной такой режим вычислений невозможен.

Модель трансформера обеспечивает высокую точность распознавания: лучший из опубликованных на момент написания этого текста результат, зарегистрированный на корпусе LibriSpeech [27], принадлежал сквозной системе распознавания [13], построенной на модели трансформера. В этом случае, так же как и в модели LAS, сложно сделать сравнение качества работы языковой модели.

Автономно, без акустической и произносительной моделей, качество языковой модели Трансформера исследовано в работе [1]. Для двух, наиболее часто используемых больших корпусов данных, Миллиард слов [6] и WikiText [20] (который состоит из отдельных предложений и текстов статей Википедии), языковая модель Трансформера обеспечила лучшие значения показателя перплексии по сравнению

с другими известными результатами (табл.6). Сравнительно низкое число параметров в модели объясняется широким использованием их связывания. Сеть Трансформера была очень глубокой, до 48 слоев (24 композитных двухслойных блока декодера).

Таблица 6

Перплексия нейросетевых языковых моделей [1] на корпусе Миллиард слов

Тип сети	Перплексия	Число параметров
Статистическая 5-граммн.KN [6]	67.6	нет
Нерекуррентная полносвязная [7]	91.2	нет
Простая рекуррентная [15]	68.3	нет
Сверточная [29]	31.9	428M
Рекуррентная, LSTM [16]	30.0	1040M
Стеки, рекуррентные LSTM [20]	28.0	>4371M
Трансформер [1]	23.02	1026M

Таблица 7

Перплексия нейросетевых языковых моделей на корпусе WikiText103 [20]

Тип сети	Перплексия	Число параметров
Рекуррентная, LSTM с памятью[12]	40.8	нет
Сверточная [29]	37.2	229M
Рекуррентная, QRNN [25]	33.0	151M
Рекуррентная, LSTM с памятью [14]	29.2	нет
Трансформер [1]	18.7	247M

Результаты описанного выше автономного тестирования нельзя перенести на вариант полной (с акустической и произносительной моделями) системы распознавания речи, поскольку существующие акустические корпуса данных существенно уступают по размеру текстов корпусам 1 Миллиард слов и WikiText.

3. Недостатки нейросетевых моделей языка

Из приведённых выше результатов очевидно, что нейросетевые языковые модели обладают качественно лучшими характеристиками по сравнению со статистическими *n*-граммными.

В существенной степени это обусловлено возможностью определения и использования векторизованных представлений слов с расстояниями, которые коррелируют с синтаксическим или семантическим сходством между словами. Это позволяет более корректно оценивать вероятности *n*-грамм, особенно невидимых, и в конечном итоге позволяет получить заметно более низкую перплексию.

Тем не менее до сих пор статистические модели продолжают широко использоваться в системах распознавания и вопрос о полной замене одних типов моделей на другие не очевиден, поскольку у нейросетевых моделей есть также и существенные недостатки.

К принципиальным недостаткам нейросетевых ЯМ и методов оценки их параметров относятся:

- недостаточно эффективные алгоритмы распараллеливания вычислений (особенно для рекуррентных нейросетей);

- недостатки методов обучения, приводящие к появлению эффектов типа смещения длительности (length bias) или бесконечного повторения [29]. В первом случае языковая модель, обученная на коротких предложениях, начинает преувеличивать вероятности коротких фраз, и наоборот. Во втором при проверке качества работы языковой модели она, в режиме синтеза, может генерировать бесконечно длинные предложения, рассматривая их как вероятные, хотя они имеют нулевую вероятность;

- доказавшая свою эффективность модель внимания инвариантна к порядку следования входных элементов. Если поменять входные элементы местами, то соответственно местами поменяются их веса, но функция внимания как набор взвешенных значений, не изменится. Очевидно, это неадекватно речевому сигналу.

Существуют также текущие практические проблемы, решение которых является важным условием для расширения сферы использования нейросетевых языковых моделей.

К таким проблемам относится условие наличия достаточно большого обучающего корпуса текстовых данных, что сложно сделать во многих прикладных областях. На небольших корпусах превосходство нейросетевых языковых моделей неочевидно, более качественные модели получаются интерполяцией нейросетевых и статистических моделей.

Существенным недостатком является также огромный объём используемых вычислительных ресурсов. В частности, в том случае, когда на выходном слое сети генерируется распределение вероятностей слов, при словарях в миллионы слов слой будет содержать миллиарды параметров и почти вся сеть будет сосредоточена на выходе. Нормализация выходных активаций потребует огромного объёма вычислений, что делает проблематичным использование такой модели в системах распознавания речи с небольшой задержкой.

Одна из причин сохраняющегося использования *n*-граммных моделей связана с тем, что публикуемые нейросетевые ЯМ в основном не ориентированы на обработку данных в реальном режиме времени. Входом является сразу все, а не поступающие синхронно со временем символы. Это существенно влияет на качество работы моделей: использование даже неполной имитации обработки в реальном времени сопровождается заметной потерей качества работы ЯМ.

Таким образом, используемые алгоритмы нейросетевых языковых моделей нуждаются в оптимизации с целью сокращения вычислительных требований, особенно в режиме распознавания.

Тенденции развития нейросетевых языковых моделей

Создание и развитие нейросетевых моделей привело к качественному улучшению характеристик моделей языка: корпусная перплексия на основных тестовых корпусах данных снизилась по сравнению с *n*-граммными статистическими до значений, которые недавно представлялись маловероятными и далёкими от потенциально достижимых.

Важное преимущество нейросетевых моделей перед статистическими заключается в использовании непрерывных векторных представлений слов с мерами сходства, которые коррелируют с их семантическими и синтаксическими признаками. Другое важное преимущество состоит в использовании более длительных по времени контекстов.

Качество современных нейросетевых языковых моделей существенно зависит от размеров и свойств обучающих текстовых корпусов.

Если для языковых моделей на простых нейросетях с относительно небольшими обучающими корпусами данных минимальной перплексией обладали интерполированные статистические *n*-граммные и нейросетевые языковые модели, то современные результаты на очень больших корпусах текстовых данных для моделей с клетками LSTM/GRU или свёрточных сетей с вниманием (табл. 6 и табл. 7) можно интерпретировать как отсутствие дальнейшей необходимости в использовании статистических моделей.

Тем не менее нельзя утверждать, что нейросетевые модели стали основой современного языкового моделирования в системах распознавания. В практических, продукционных системах распознавания речи до сих пор широко используются статистические *n*-граммные модели. Одной из основных причин этого являются повышенные требования к вычислительным ресурсам и корпусам данных, которые предъявляют нейросетевые модели.

В экспериментах по достижению минимальных показателей перплексии на корпусе речевых данных *Миллиард слов* [17] (эти результаты соответствуют 4 строке табл. 6) большая часть усилий оказалась посвящена поиску способов сокращения количества параметров сети без заметного ухудшения качества работы языковой модели. Несмотря на это, для массива из 40 мощных графических ускорителей обучение только одной модели потребовало несколько сотен часов работы. Соответственно, если продукционную систему распознавания с нейросетевой языковой моделью требуется регулярно дообучать и адаптировать к изменениям данных, что является довольно распространенным условием, возникает вопрос о возможности реализации этого на практике.

В рамках используемого сейчас экстенсивного подхода к развитию нейросетевых моделей алгоритмы обучения должны уметь эффективно использовать огромные выборки данных, и обладать большой ёмкостью. Соответственно, они предъявляют рекордные требования к вычислительным ресурсам. Одна из самых больших в настоящее время моделей на основе трансформера GPT-3 с символьной языковой моделью (Generative Pre-Trained Transformer) [3] имеет 175 миллиардов параметров. Обучение этой модели на корпусе Common Crawl [25] примерно из триллиона слов занимает 3.6E+03 петафлопс-дней (точнее, петафлопс-секунда-дней, то есть свыше

трёх тысяч дней при условии выполнения 10^{15} нейро-операций в секунду в течение одного дня или примерно 10^{20} всего операций).

Значительное количество усилий разработчиков уже сейчас направлено на создание моделей с существенно меньшими требованиями к объёму памяти и вычислительных мощностей. Основные исследуемые направления решения этой проблемы и полученные результаты в виде субоптимальных решений (например, иерархический софтмакс и его модификации, сэмплинг с учётом важности, оценка по контрасту с шумом, использование словарей на основе подслов и т.п.) работают все же хуже, чем решения на основе вычисления софтмакс для всех слов словаря.

Другое перспективное направление возникает из результатов работы с моделями типа GPT-2, GPT-3. Очень большая языковая модель обучается один раз и затем используется для решения различных задач так, как она есть, либо с относительно несложной адаптацией к новым задачам и проблемным областям. Языковая модель GPT-3 проверялась на целом ряде задач из разных прикладных областей, с собственными наборами тестовых данных и оказалась достаточно конкурентоспособной, более того, во многих случаях существенно эффективнее лучших на тот момент нейросетевых моделей, которые были адаптированы к специфике решаемой задачи и текстовым данным. Например, перплексия на текстовом корпусе Penn Tree Bank составила 20.5 по сравнению с лучшей на тот момент 35.8 полученной в [23].

Тем не менее нейросетевые ЯМ на эталонных текстовых корпусах имеют показатели качества (перплексию и энтропию) в разы меньше, чем статистические моделями. Можно ожидать, что это будет сопровождаться симметричным увеличением точности распознавания речи. Поэтому одним из наиболее перспективных направлений совершенствования систем распознавания речи на ближайшее будущее будет разработка и использование нейросетевых ЯМ.

Список литературы

1. *Alexei Baevski and Michael Auli.* Adaptive input representations for neural language modeling, 2019, arXiv: 1809.10853v3 [cs.CL].
2. *Yoshua Bengio, R_ejean Ducharme, Pascal Vincent, and Christian Janvin.* A neural probabilistic language model. J. Mach. Learn. Res., 3:1137–1155, 2003.
3. *Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Je_rey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever and Dario Amodei.* Language models are few-shot learners, 2020, arXiv:2005.14165 [cs.CL].

4. *William Chan, Navdeep Jaitly, and Oriol Vinyals Quoc V.* Le. Listen, attend and spell, 2015, arXiv: 1508.01211v2 [cs.CL].
5. *Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Tony Robinson, Phillipp Koehn*. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google, 2013.
6. *Grangier David Chen Welin and Auli Michael.* Strategies for training large vocabulary neural language models, 2015, arXiv: 1512.04906 [cs.CL].
7. *Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani.* State-of-the-art speech recognition with sequence-to-sequence models, 2018, arXiv: 1712.01769v6 [cs.CL].
8. *Bahdanau D., Cho K., and Bengio Y.* Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations, 2015.
9. *Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier.* Language modeling with gated convolutional networks, 2017, arXiv: 1612.08083v3 [cs.CL].
10. *Je_rey L. Elman.* Finding structure in time. Cognitive Science, 14(2): 179–211, 1990.
11. *Jelinek F.* Statistical Methods for Speech Recognition. Massachusetts, MIT Press, 1997.
12. *Peter F.Brown, Vincend J.Della Pietra, Peter V.deSouza, Jenifer C.Lai, and Robert L.Mercer.* Class based n_gram models of natural language. In Proceedings of the IBM Natural Language ITL, pages 283–298, 1990.
13. *Synnaeve G., Xu Q., and Kahn J. at al.* End-to-end asr: From supervised to semi-supervised learning with modern architectures, 2019, arXiv: 1911.08460v1 [cs.LG].
14. *Edouard Grave, Armand Joulin, Moustapha Ciss_e, David Grangier and Herv_e J_egou.* E_cient softmax approximation for gpus, 2016, arXiv:1609.04309 [cs.CL].
15. *Sepp Hochreiter and J_urgen Schmidhuber.* Long short-term memory. Neural computation, 9:1735_80, 12 1997.
16. *Shihao Ji, S.V. N. Vishwanathan, Nadathur Satish, Michael J, Anderson, and Pradeep Dubey.* Blackout: Speeding up recurrent neural network language models with very large vocabularies, 2016, arXiv:1511.06909v7 [cs.CL].
17. *Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu.* Exploring the limits of language modeling, 2016, arXiv:1602.02410v2 [cs.CL].
18. *R. Kneser and H. Ney.* Improved backing-o_ for m-gram language modeling. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-95, volume 1, page 181–184, 1995.
19. *Stephen Merity, Nitish Shirish Keskar, and Richard Socher.* An analysis of neural language modeling at multiple scales., 2018, arXiv: 1803.08240 [cs.CL].
20. Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016, arXiv: 1609.07843 [cs.CL].
21. *Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukas Burget, and Jan Cernocky.* Neural machine translation by jointly learning to align and translate. In ISCA, editor, INTERSPEECH, pages 605–608, 2011.
22. *Tomas Mikolov, Martin Kara_at, Jan Cernocky, and Sanjeev Khudanpur*. Recurrent neural network based language model. In Interspeech, 2010.
23. *Alec Radford, Je_rey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever.* Language models are unsupervised multitask learners. Technical report, OpenAi, 2019.
24. *Jack W. Rae, Chris Dyer, Peter Dayan,and Timothy P. Lillicrap.* Fast parametric learning with activation memorization., 2018, arXiv: 1803.10049 [cs.LG].

25. *Colin Ra_el, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou,Wei Li, and Peter J. Liu.* Exploring the limits of transfer learning with a uni_ed text-to-text transformer, 2019, arXiv: 1910.10683 [cs.LG].
26. *Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geo_rey E. Hinton, , and Je_ Dean.* Outrageously large neural networks: The sparsely-gated mixture-of-experts layer., 2017, arXiv: 1701.06538 [cs.LG].
27. *Panayotov V., G. Chen, D. Povey, and S. Khudanpur.* «librispeech»: an asr corpus based on public domain audio books. In Proceedings of International Conference on Acoustics, Speech and Signal Processing, ICASSP_2015, pages 5206–5210, 2015.
28. *Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin.* Attention is all you need, 2017, arXiv: 1706.03762v5 [cs.CL].
29. *Sean Welleck, Richard Yuanzhe Ilia Kulikov, Jaedeok Kim, and Pang Kyunghyun Cho*. Consistency of a recurrent language model with respect to incomplete decoding, 2020, arXiv: 2002.02492v1 [cs.CL].

NEURAL LANGUAGE MODELS FOR AUTOMATIC SPEECH RECOGNITION

V.Y. Chuchupal,
Candidate of Physical and Mathematical Sciences, Leading Researcher at the A.A. Dorodnitsyn Computing Center of the Federal Research Center «Informatics and Management» of the Russian Academy of Sciences.
E-mail: v.chuchupal@gmail.com

Abstract

Until recently, the generally accepted approach to language modeling in speech recognition systems was based on the use of statistical ngram models.

The development and evolution of neural network language models have led to substantial increase of the language modeling quality. The value of the perplexity have been lowered in several times, to the values that have recently seemed inlikely.

These results mostly are due to the use of distributed word representations in neural models, where distance correlates with syntactic or semantic similarity between words.

The review describes the main types of modern neural language models, namely the models based on fully connected and simple recurrent networks, multilayer recurrent networks with LSTM cells, convolutional networks, encoder-decoder with attention and transformer models.

Their advantages and disadvantages, requirements for computing resources as well as the achieved language model quality indicators values are presented.

The drawbacks of neural network NMs, both related to model structure and more technical ones, arising from the implementation of training and recognition algorithms are highlighted.

A comparative values of the perplexy for the state-of-the-art approaches on the reference datasets is given. It is shown that Transformer based language models are more advantageous at least at the time of the writing of the paper.

Keywords: automatic speech recognition, language modeling, *n*-gram language models, neural language models, end-to-end speech recognition, attention model.