Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Камалов Руслан Рамилевич

Нейросетевой подход к построению тематических моделей

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Научный руководитель:

академик РАН

К. В. Рудаков

Москва, 2018

Содержание

Аннотация

В данной работе предложена архитектура рекуррентной нейронной сети (RNN), предназначенная для непосредственного захвата глобального смысла, связывающего слова в документе через скрытые темы. Из-за их последовательного характера RNN хороши при захвате локальной структуры последовательности слов как семантической, так и синтаксической, однако, могут столкнуться с трудностями при запоминании долгосрочных зависимостей. Эти долгосрочные зависимости имеют семантический характер. Напротив, тематические модели способны захватывать глобальную семантическую структуру коллекции в целом, но не учитывают природу естественного языка, в частности игнорируют порядок слов. Предложенная модель объединяет достоинства RNN и тематических моделей.

1. Введение

Данная работа посвящена вероятностному тематическому моделированию – статистическому методу анализа коллекций текстовых документов. Для набора документов вводятся скрытые переменные – темы, которые описывают процесс порождения данных.

Тематическое моделирование — активно развивающаяся в последние годы область машинного обучения. Оно позволяет решать задачи тематического поиска, категоризации и кластеризации корпусов текстовых документов. При построении хорошей тематической модели необходимо учитывать внутреннюю структуру языка, семантику слов и связи между ними. В последние годы для решения этой проблемы часто используются [21, 22, 23, 24] рекуррентные нейронные сети.

При чтении текста человек использует некий механизм, который каким-то образом позволяет ему помнить суть того, что он прочитал. Часто человек способен довольно точно сказать, какое слово будет следующим в тексте. Математически это формализуется средствами языковой модели. Существуют различные языковые модели – от простых n-граммных до языковых моделей, основанных на рекуррентных нейронных сетях. Хорошая языковая модель должна захватывать как минимум два важных свойства естественного языка. Первое - правильный синтаксис. Зачастую чтобы сделать адекватный прогноз следующего слова, человеку достаточно лишь нескольких предыдущих слов. Поэтому правильный синтаксис – это локальное свойство. В этом случае порядок слов в предложении имеет значение. Второе свойство – это семантическая согласованность. Чтобы понять глобальный смысл предложения или документа, нам часто нужно рассмотреть большое количество слов. В этом случае их порядок обычно имеет гораздо меньшее значение.

Поскольку большинство моделей рассматривают контекстное окно фиксированного размера, традиционные n-граммные и нейронные вероятностные языковые модели [25] имеют трудности с извлечением глобальной семантической информации из текста. Чтобы преодолеть это, языковые модели на основе RNN [26] используют скрытые состояния для «запоминания» истории последовательности слов. Однако, ни один из этих подходов явно не моделирует два основных свойства упомянутого выше языка, правильный синтаксис и семантическую согласованность.

Несмотря на все достоинства, нейронные сети обладают одним существенным недостатком – отстутствием интерпретируемости. Под интерпретируемостью весов нейрон-

ной сети будем понимать неотрицательность и нормированность весов слоя нейронной сети. В данной работе представлена архитектура нейронной сети, решающая задачу тематической категоризации коллекции документов, а также способ настройки ее весов, частично решающий проблему отсутствия интерпретируемости.

2. Тематическое моделирование

Пусть D обозначает конечное множество (коллекцию) документов (текстов) и пусть W — конечное множество (словарь) всех терминов, из которых состоят эти документы. Под термином подразумевается либо слово, либо целая фраза. В соответствии с гипотезой «мешка слов» каждый документ $d \in D$ представляется в виде подмножества словаря W, где каждому слову w ставится в соответствие число n_{dw} раз, которое он встретился в документе d. Предположим, что появления каждого термина в каждом документе связано с некоторой латентной темой из конечного множества тем T. Текстовая коллекция представляется в виде набора троек $(d_i, w_i, t_i), i = 1, ..., n$, выбранных независимо из дискретного распределения $p(d, w, t)$ над конечным вероятностным пространством $D \times W \times T$. Термины w_i и документы d_i — это наблюдаемые переменные, а темы t_i — скрытые. Вероятностная тематическая модель описывает вероятности $p(w|d)$ появления терминов в документах как смеси распределений слов в темах $\varphi_{wt} = p(w|t)$ и тем в документах $\theta_{td} = p(t|d)$:

$$p(w|d) = \sum_{t \in T} p(w|t)p(t|d) = \varphi_{wt}\theta_{td} \tag{1}$$

Эта смесь напрямую соответствует генеративному процессу, в процессе которого модель порождает документы d: для каждой позиции слова i происходит генерация индекса темы t_i из распределения $p(t|d)$, после чего сэмплируется слово w_i из распределения $p(w|t_i)$. Параметры вероятностной тематической модели часто представляются в виде матриц $\Phi = (\varphi_{wt})_{W \times T}$ и $\Theta = (\theta_{td})_{T \times D}$ с неотрицательными и нормированными столбцами ϕ_t и θ_d, представляющими собой мультиномиальные распределения слов в темах и тем в документах.

2.1. PLSA

В вероятностном латентном семантическом анализе (PLSA) [1], тематическая модель 1 обучается путём максимизации логарифма правдоподобия с линейными ограни-

чениями неотрицательности и нормировки:

$$L(\Phi, \Theta) = \sum_{d \in D} \sum_{w \in D} d_{dw} \ln \sum_{t \in T} \varphi_{wt} \theta_{td} \to \max_{\Phi, \Theta} \tag{2}$$

$$\sum_{w \in W} \phi_{wt} = 1, \varphi_{wt} \geq 0, \sum_{t \in T} \theta_{td} = 1, \theta_{td} \geq 0 \tag{3}$$

где $n_d w$ — абсолютная частота слова w в документе d.

Метод простых итерация для решения этой системы уравнений эквивалентен EM-алгоритму и обычно на практике используется именно он. E-шаг может рассматриваться как применение формулы Байеса для получения вероятностей $p_{tdw} = p(t|d, w)$ для каждого термина w и документа d. M-шаг интерпретируется как частотная оценка условных вероятностей φ_{wt} и θ_{td}. Итеративный процесс обычно начинается со случайных начальных приближений Φ и Θ.

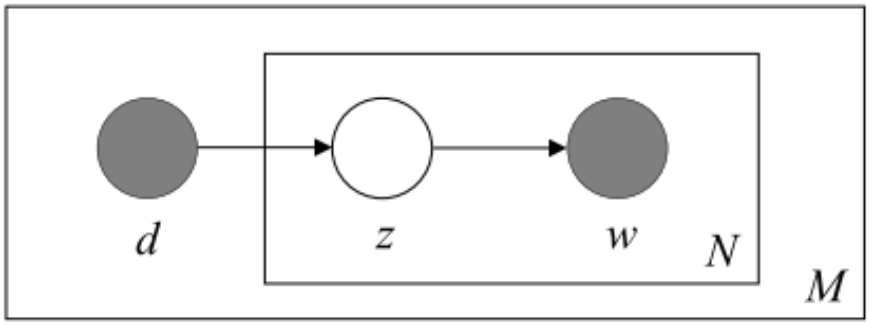

Рис. 1. Графическая модель PLSA.

2.2. LDA

Модель латентного размещения Дирихле (LDA) [2, 3] вводит априорные распределения Дирихле для векторов вероятностей слов в темах $\varphi_t \sim Dir(\beta)$ и для векторов вероятностей тем в документах $\theta_d \sim Dir(\alpha)$ с векторами параметров $\beta = (\beta_w)_{w \in W}$ и $\alpha = (\alpha_t)_{t \in T}$ соответственно. Вывод в LDA обычно производится либо с помощью вариационного приближения, либо с помощью сэмплирования Гиббса. Обычно используется свёрнутая схема Гиббса, где тема t_i для каждой позиции слова (d_i, w_i) итеративно сэмплируется из распределения $p(t_d, w)$, такого же, как в PLSA, но со со сглаженными байесовскими оценками условных переменных:

$$\varphi_{wt} = norm_{w \in W}(n_{wt} + \beta_w), \qquad \theta_{td} = norm_{t \in T}(n_{td} + \alpha_t), \tag{4}$$

где n_{wt} — число раз, которое термин w был сгенерирован из темы t и n_{td} это число раз, которое термины из документы d были сгенерированы из темы t, исключая текущую тройку (d_i, t_i, w_i).

В последние годы было опубликовано много работ с различными расширениями LDA [4, 5, 6, 7]. Для описываемой здесь задачи извлечения пользовательской информации по некоторой специфичной тематике.

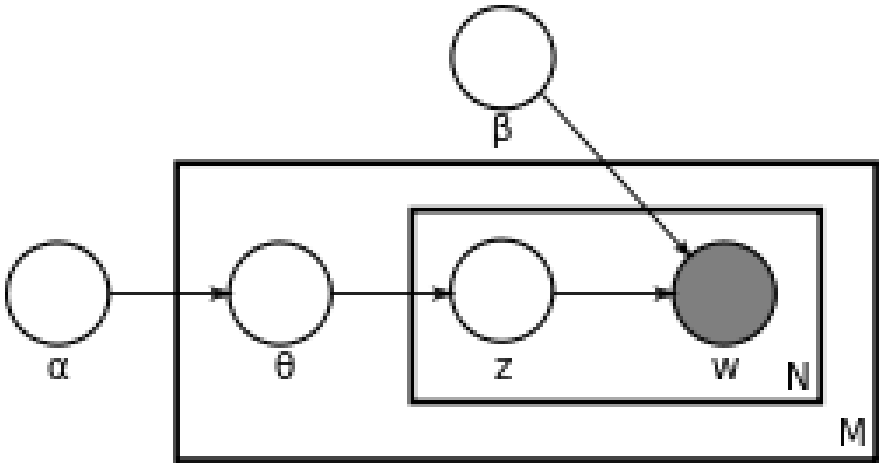

Рис. 2. Графическая модель LDA.

2.3. ARTM

Тематическое моделирование может быть рассмотрено как специальный случай матричного разложения, где задача состоит в том, чтобы найти низкоранговую аппроксимацию $\Phi\Theta$ данной разреженной матрицы счётчиков терминов-документов. Следует отметить, что произведение $\Phi\Theta$ определено с точностью до невырожденного линейного преобразования: $\Phi\Theta = (\Phi S)(S^{-1}\Theta)$. Таким образом, задача является некорректно поставленной и имеет бесконечное множество решений. Эксперименты на модельных и реальных данных показали, что ни PLSA, ни LDA не удаётся достигнуть устойчивого решения. Для увеличения стабильности обучения следует добавить дополнительные оптимизационные ограничения, обычно называемые регуляризаторамих [8]. В аддитивной регуляризации тематических моделей (APTM) модель обучается путём максимизации линейной комбинации логарифма правдоподобия $L(\Phi, \Theta)$ и r регуляризаторов

$R_i(\Phi, \Theta), i = 1, ..., r$ с коэффициентами регуляризации τ_i:

$$r(\Phi, \Theta) = \sum_{i=1}^{r} \tau_i R_i(\Phi, \Theta), \quad L(\Phi, \Theta) + R(\Phi, \Theta) \to \max_{\Phi, \Theta} \tag{5}$$

Условия Каруша-Куна-Такера для этой нелинейной оптимизационной задачи дают (с учётом некоторых технических ограничений) необходимые условия локального максимума как решения следующей системы уравнений [9]:

$$p_{tdw} = norm_{t \in T}(\varphi_{wt}\theta_{td}); \tag{6}$$

$$\varphi_{wt} = norm_{w \in W}(n_{wt} + \varphi_{wt}\frac{\partial R}{\partial \varphi_{wt}}); \quad n_{wt} = \sum_{d \in D} n_{dw} p_{tdw}; \tag{7}$$

$$\theta_{td} = norm_{t \in T}(n_{td} + \theta_{td}\frac{\partial R}{\partial \theta_{td}}); \quad n_{td} = \sum_{w \in d} n_{dw} p_{tdw}; \tag{8}$$

Как и в случае PLSA, для решения этой системы может быть использован EMалгоритм. Преимущество APTM заключается в том, что каждый аддитивный регуляризатор превращается в простую модификацию M-шага. Многие модели, разработанные прежде в рамках байесовского подхода, могут быть несложно интерпретированы, обучены и скомбинированы в рамках теории APTM [10, 9]. Например, PLSA не использует никакой регуляризации, $R = 0$, а LDA с априорными распределениями Дирихле $\varphi_t \sim Dir(\beta)$ и $\theta_d \sim Dir(\alpha)$ and оценками максимума апостериорных вероятностей Φ, Θ соответствует модели со сглаживающим регуляризатором, который интепретируется как минимизатор KL-дивергенций между столбцами Φ, Θ и заданными распределениями β, α соответственно.

3. Рекуррентные нейронные сети

Рекуррентные нейронные сети (Recurrent Neural Network, RNN) — класс моделей машинного обучения, основанный на использовании предыдущих состояний сети для вычисления текущего [11, 12]. Такие сети удобно применять в тех случаях, когда входные данные задачи представляют собой нефиксированную последовательность значений, как, например, текстовые данные, где текстовый фрагмент представлен нефиксированным количеством предложений, фраз и слов. Каждый символ в тексте, отдельные слова, знаки препинания и даже целые фразы — все это может являться атомарным элементом входной последовательности. На каждом шаге обучения t значение скрытого слоя рекуррентной нейронной сети $h^t \in R^m$ вычисляется следующим образом:

$$h^t = f(Wx^t + Uh^{(t-1)} + b^h),$$

где $x^t \in R^n$ — входной вектор в момент времени t (например, векторное представление текущего слова в текстовом фрагменте); $W \in R^{m \times n}, U \in R^{m \times m}, b^h \in R^m$ — обучаемые параметры рекуррентной нейронной сети; f — функция нелинейного преобразования. Чаще всего в качестве нелинейного преобразования применяют одну из следующих функций: сигмоидальная функция 9, гиперболический тангенс 10, ReLu 11:

$$f(x) = \sigma(x) = \frac{1}{1 + \exp(-x)} \tag{9}$$

$$f(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{10}$$

$$f(x) = \max(0, x) \tag{11}$$

В простой рекуррентной нейронной сети Рис. 3 выходное значение $y^t \in R^l$ на текущем шаге t вычисляется по формуле:

$$y^t = Wh^t + b$$

где $W \in R^{l \times m}$ и $b \in R^l$ — обучаемые параметры.

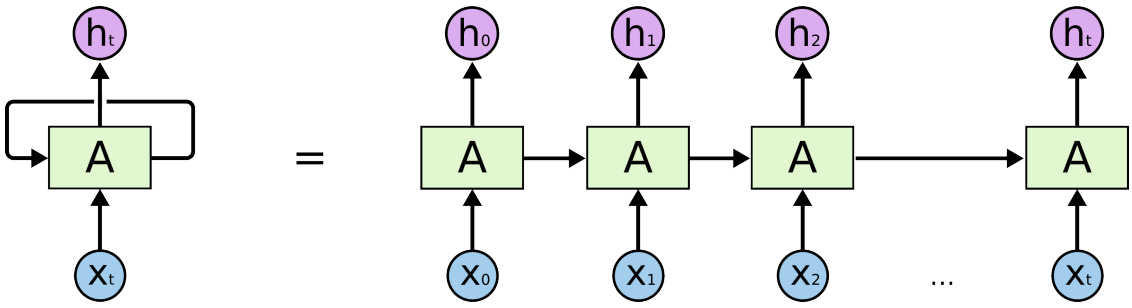

Рис. 3. Простейшая схема рекурентной сети.

3.1. LSTM

В 1997 году Зепп Хохрайтер (Sepp Hochreiter) и Юрген Шмидхубер (Jürgen Schmidhuber) представили новый подход, получивший название LSTM (Long Short-Term Memory — долгая краткосрочная память) [13]. Рекуррентные нейронные сети, основанные на этом подходе, имеет более продвинутый (и более сложный) способ вычисления h^t. Данный способ, помимо входных значений и предыдущего состояния сети, использует также фильтры (gates), определяющие, каким образом информация будет использоваться для вычисления как выходных значений на текущем слое y^t, так и зна-

чений скрытого слоя на следующем шаге h^{t+1}. Весь процесс вычисления h^t для простоты упоминается как LSTM-слой (LSTM layer, LSTM unit).

Рассмотрим подробнее структуру LSTM-слоя. Центральным понятием здесь является запоминающий блок (memory cell), который, наряду с состоянием сети h, вычисляется на каждом шаге, используя текущее входное значение x^t и значение блока на предыдущем шаге c^{t-1}. Входной фильтр (input gate) i^t определяет, насколько значение блока памяти на текущем шаге должно влиять на результат. Значения фильтра варьируются от 0 (полностью игнорировать входные значения) до 1, что обеспечивается областью значений сигмоидальной функции:

$$i^t = \sigma(W^i x^t + U^i h^{t-1})$$

«Фильтр забывания» (forget gate) позволяет исключить при вычислениях значения памяти предыдущего шага:

$$f^t = \sigma(W^f x^t + U^f h^{t-1})$$

На основе всех данных, поступающих в момент времени t, вычисляется состояние блока памяти c^t на текущем шаге, используя фильтры 12 и 13:

$$\tilde{c}^t = \tanh(W x^t + U h^{t-1}); \tag{12}$$

$$c^t = f^t \cdot c^{t-1} + i^t \cdot \tilde{c}^t. \tag{13}$$

Выходной фильтр (output gate) аналогичен двум предыдущим и имеет вид:

$$o^t = \sigma(W^o x^t + U^o h^{t-1}). \tag{14}$$

Итоговое значение LSTM-слоя определяется выходным фильтром 14 и нелинейной трансформацией над состоянием блока памяти 15:

$$h^t = o^t \cdot \tanh(c^t) \tag{15}$$

3.2. GRU

В 2014 году в работе [14] была представлена модель GRU (Gated Recurrent Unit), основанная на тех же принципах, что и LSTM, но использует меньше фильтров и операций для вычисления h^t [15]. Фильтр обновления z^t (update gate) и фильтр сброса

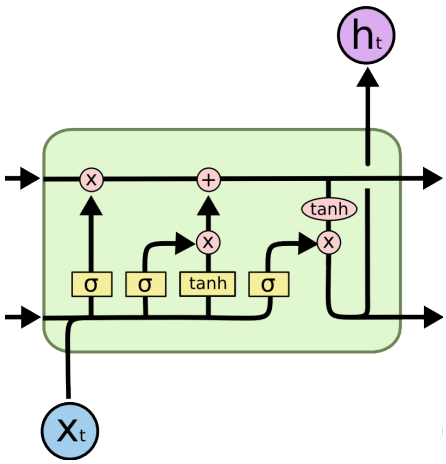

Рис. 4. Схематическое представление ячейки LSTM.

состояния r^t (reset gate) вычисляются по следующим формулам:

$$z^t = \sigma(W^z x^t + U^z h^{t-1}) \tag{16}$$

$$r^t = \sigma(W^r x^t + U^r h^{t-1}). \tag{17}$$

Выходное значение h^t вычисляется на основе промежуточного значения $\tilde{h}^t$, которое, при помощи фильтра сброса состояния 17, определяет, какие значения предыдущего шага h^{t-1} следует исключить (здесь можно видеть прямую аналогию с фильтром забывания из LSTM):

$$\tilde{h}^t = \tanh(W x^t + r^t \cdot U h^{t-1}) \tag{18}$$

Используя фильтр обновления 16 и промежуточное значение 18, имеем:

$$h^t = z^t \cdot h^{t-1} + (1 - z^t) \cdot \tilde{h}^t$$

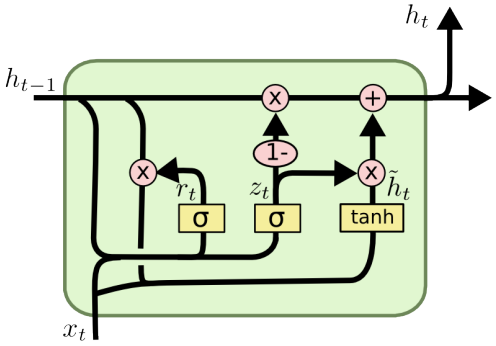

Рис. 5. Схематическое представление ячейки GRU.

4. Оптимизация параметров в вероятностном симплексе

В этой главе описываются алгоритмы, предназначенные для оптимизации целевой функции Q с ограничениями на вероятностном симплексе. Такая постановка часто возникает в задачах статистического обучения, log-linear моделях, таких как CRF [16] и max-margin моделях, таких как maximum-margin Markov Networks [17].

Математическая постановка:

$$\max_{w_i \in \Delta} Q(w_1, w_2, ..., w_n), \tag{19}$$

где w_i – объекты из вероятностного симплекса Δ:

$$\Delta = \{w_i \in R^m : w_{i,j} \geq 0, \quad \sum_{j=j}^{m} w_{i,j} = 1\} \tag{20}$$

4.1. Exponential gradient

Exponential gradient update (EG) первоначально введен Кивиненом и Вартутом [18] в контексте алгоритмов онлайн-обучения. Целью метода (EG) является оптимизация выпуклой (вогнутой) функции $Q(w)$ при ограничениях $w_i \in \Delta$.

Алгоритм EG представляет собой итеративнй процесс. Принимая на вход последовательность распределений $w_i \in \Delta$, обновление каждого из векторов происходит по следующему правилу (в случае оптимизации вогнутой функции):

$$w_{i,j}^{(t+1)} = \frac{1}{Z_i} w_{i,j}^{(t)} \exp(\eta \nabla_{i,j}),$$

где $\nabla_{i,j} = \frac{\partial Q(w)}{\partial w_{i,j}}$, $Z_i = \sum_i \exp(\eta \nabla_{i,j})$, параметр η – темп обучения. Далее в тексте будем использовать обозначение $w_{i,j}^{(t+1)} \propto w_{i,j}^{(t)} \exp(\eta \nabla_{i,j})$.

Также авторами была предложена масштабируемая версия EG алгоритма (Online EG Update), где на каждой итерации выбирается подмножество векторов из Δ и обновление производится только по ним. Псевдокод алгоритмов EG и Online EG для оптимизации вогнутой функции Q представлены ниже:

Algorithm 1 Exponential Gradient Update

Вход: Вогнутая функция $Q : \Delta^n \to R$, темп обучения $\eta > 0$

Инициализация: Инициализировать начальные значения для w_i

for $t = 1, ..., T$ **do**

  for i, j **do**

    $\nabla_{i,j} = \frac{\partial Q(w)}{\partial w_{i,j}}$

  end for

  for i, j **do**

    $w_{i,j}^{(t+1)} \propto w_{i,j}^{(t)} \exp(\eta \nabla_{i,j})$

  end for

end for

Выход: $w_i^{(t+1)}, i = 1, ..., n$

Algorithm 2 Online Exponential Gradient Update

Вход: Вогнутая функция $Q : \Delta^n \to R$, темп обучения $\eta > 0$

Инициализация: Инициализировать начальные значения для w_i

for $t = 1, ..., T$ **do**

  выбрать $k \sim Uniform(1, 2, ..., n)$

  for j **do**

    $\nabla_{k,j} = \frac{\partial Q(w)}{\partial w_{k,j}}$

  end for

  for j **do**

    $w_{k,j}^{(t+1)} \propto w_{k,j}^{(t)} \exp(\eta \nabla_{k,j})$

  end for

end for

Выход: $w_i^{(t+1)}, i = 1, ..., n$

4.2. Multiplicative update

Еще один метод для поиска оптимума в задаче 19 - 20 – метод мультипликативного обновления (Multiplicative update (MU)). Этот алгоритм был предложен в работах [19, 20] и был использован в глубинном обучении для задачи разделения речи (Speech separation) на основе неотрицательной матричной факторизации (NMF).

Также как и EG, алгорит MU представляет собой итеративнй процесс. Принимая на вход последовательсноть распределений $w_i \in \Delta$, обновление каждого из векторов происходит по следующему правилу:

$$w^{(t+1)_i} \propto w_i^{(t)} \frac{1 + [\nabla_{w_i^{(t)}}]_+}{1 + [\nabla_{w_i^{(t)}}]_-},$$

где

$$[\nabla_{w_{i,j}^{(t)}}]_+ = \begin{cases} \nabla_{w_{i,j}^{(t)}} & \nabla_{w_{i,j}^{(t)}} \geq 0 \\ 0 & \nabla_{w_{i,j}^{(t)}} < 0 \end{cases}$$

$$[\nabla_{w_{i,j}^{(t)}}]_- = \begin{cases} |\nabla_{w_{i,j}^{(t)}}| & \nabla_{w_{i,j}^{(t)}} \leq 0 \\ 0 & \nabla_{w_{i,j}^{(t)}} > 0 \end{cases}$$

положительная и отрицательная части градиента соответственно. По аналогии с EG у метода MU есть масштабируемая версия Online MU. Псевдокод алгоритмов MU и Online MU для оптимизации вогнутой функции Q представлены ниже:

Algorithm 3 Multiplicative Update

Вход: Вогнутая функция $Q : \Delta^n \to R$
Инициализация: Инициализировать начальные значения для w_i
for $t = 1, ..., T$ **do**
    for i **do**
        $\nabla_i = \frac{\partial Q(w)}{\partial w_i}$
    end for
    for i **do**
        $w_i^{(t+1)} \propto w_i^{(t)} \frac{1+[\nabla_{w_i^{(t)}}]_+}{1+[\nabla_{w_i^{(t)}}]_-}$
    end for
end for
Выход: $w_i^{(t+1)}, i = 1, ..., n$

Algorithm 4 Online Multiplicative Update

Вход: Вогнутая функция $Q : \Delta^n \to R$

Инициализация: Инициализировать начальные значения для w_i

for $t = 1, ..., T$ **do**

  выбрать $k \sim Uniform(1, 2, ..., n)$

  $\nabla_k = \frac{\partial Q(w)}{\partial w_k}$

  $w_k^{(t+1)} \propto w_k^{(t)} \frac{1+[\nabla_{w_k^{(t)}}]_+}{1+[\nabla_{w_k^{(t)}}]_-}$

end for

Выход: $w_i^{(t+1)}, i = 1, ..., n$

5. Рекуррентная модель тематической категоризации

В этой главе описана архитектура нейронной сети, предназначенной для непосредственного захвата как локальной семантики документа, так и глобальной семантики коллекции в целом. Предложенная модель состоит из двух рекуррентных блоков, отвечающих за генерацию тематического профиля документа и блока линейных слоев, являющихся представлением самих тем как набора распределений над фиксированным словарем коллекции. Предложенная архитектура комбинирует в себе нейросетевой подход, где генерация тематик происходит посредством рекуррентных слоев сети и модели PLSA, рассматривающей тематическую категоризацию как задачу матричного разложения. Схема модели изображена на Рис. 6

5.1. Архитектура

Обработка одного документа происходит следующим образом: первый рекуррентный блок обрабатывает каждое слово в предложении, генерируя при этом текущее скрытое состояние. Скрытое состояние, сгенерированное для последнего слова в предложении объявляется векторным представлением предложения. Второй рекуррентный блок обрабатывает последовательность векторных представлений предложений. Размерность последнего скрытого состояния этого блока совпадает с количеством тем, которые мы пытаемся выделить в коллекции. Последнее скрытое состояние второго рекуррентного блока объявляется скрытым векторным представлением всего документа. Тематический профиль документа вычисляется путем применения функции softmax к его скрытому векторному представлению. В отличие от PLSA, предложенная модель иерархически получает скрытое представление документа через представления пред-

ложений и на его основе вычисляет тематический профиль документа.

Модель: Для документа d, состоящего из последовательности предложений $d = s_1, ..., s_n, \quad |d| = n$, где каждое предложение – последовательность слов $s_i = w_1, ..., w_{k_i}, \quad |s_i| = k_i$,

- для каждого слова w_i в предложении s_j первый рекуррентный блок генерирует скрытое состояние h_1^j
- последнее скрытое состояние h_{k_i} объявляется векторным представлением h^{s_i} i-ого предложения и подается на вход второму рекуррентному блоку
- второй рекуррентный блок генерирует скрытое векторное представление всего документа
- тематика документа $p(t|d)$ вычисляется путем применения функции softmax к скрытому векторному представлению документа

Аналогом распределения $p(w|t)$ модели PLSA в предложенной модели являются $|W|$ линейных слоев размера $1 \times T$. Каждый такой слой представляет собой вектор вероятностей принадлежности данного слова одной из тем $p(w|t_1), ..., p(w|t_T)$.

Параметры модели разбиваются на две группы:

- $|W|$ линейных слоев размера $1 \times T$, являющихся аналогом матрицы Φ в ARTM; обозначим их $\tilde{\Phi}$ – параметры условной оптимизации
- параметры рекуррентных блоков, параметры безусловной оптимизации – обозначим их $\tilde{\Theta}$

Модель обучается путем максимизации логарифма правдоподобия:

$$L(\tilde{\Phi}, \tilde{\Theta}) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln \sum_{t \in T} p(w|t, \tilde{\Phi}) p(t|d, \tilde{\Theta}) \to \max_{\tilde{\Phi}, \tilde{\Theta}},$$

с условием неотрицательности и нормировки совокупности весов из $\tilde{\Phi}$ для каждой компоненты $i \in 1, ..., T$

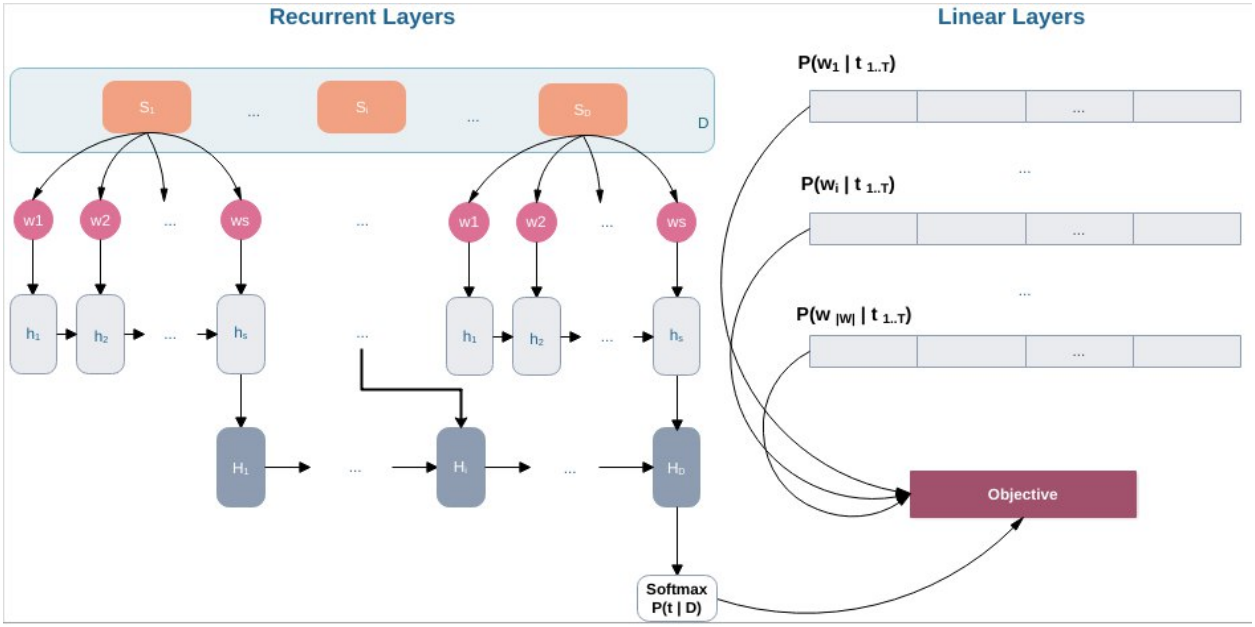

Рис. 6. Схематическое представление предложенной модели.

5.2. Алгоритм обучения

Как было сказано в предыдущем пункте, параметры предложенной модели делятся на две группы: $\tilde{\Phi}$ – параметры условной оптимизации и $\tilde{\Theta}$ – парметры безусловной оптимизации. Для оптимизации параметров $\tilde{\Phi}$ используются методы EG и MU, описанные в главе 4. Группа параметров $\tilde{\Theta}$ настраивается градиентым спуском. Для вычисления градиентов по обеим группам параметров используется алгоритм backpropagation.

Полный алгоритм обучения параметров модели представлен ниже:

Algorithm 5 Model Learning

Вход: Коллекция документов $D = d_1, ..., d_n$; число тематик T; темп обучения η; число эпох *epoch*.

Инициализация: Инициализировать начальные значения для группы параметров $\tilde{\Phi}, \quad \tilde{\Theta}$

for $e = 1, ..., epoch$ **do**

  for $d \in D$ **do**

    for $s \in d$ **do**

      for $w \in s$ **do**

        вычислить скрытый вектор состояние h_1^w

      end for

      получить скрытое представление предложения s: $h^s = h_1^{|s|}$

    end for

    получить скрытое представление документа d: $h_d = h_2^{|d|}$

    вычислить текущую тематику документа $\theta_d = softmax(h_d)$

  end for

  $\nabla_{\tilde{\Phi}} L, \nabla_{\tilde{\Theta}} L =$ backpropagation(L)

  $\tilde{\Phi} =$ update[EG, MU]($\nabla_{\tilde{\Phi}} L$, η)

  $\tilde{\Theta} =$ update[SGD, Rmsprop, Adam]($\nabla_{\tilde{\Theta}} L$)

end for

Выход: $\tilde{\Phi}, \tilde{\Theta}$

6. Эксперименты

Оценка эффективности предложенной модели проводилась на двух коллекциях документов: 20 Newsgroups и Penn TreeBank (PTB) – стандартных бенчмарках для оценки качества тематической категоризации.

6.1. Penn TreeBank Data

Коллекция Penn Treebank (PTB) – это 2500 документов, собранных за три года работы Wall Street Journal. Она состоит из приблизительно одного миллиона английских слов. В коллекции представлены записи телефонных разговоров, новостных лент, транскрибированной речи.

На этой коллекции архитектура LSTM превосходит остальные [27], и улучшить этот результат затруднительно [28]. Поэтому в качестве рекуррентных ячеек была выбрана LSTM. Поскольку PTB представляет собой небольшой набор данных, во избежании переобучения необходимо использовать технику регуляризации. В проведенных экспериментах использовался dropout для рекуррентных сетей [27].

Детали обучения:

В экспериментах была обучена модель, где в качестве первого рекуррентного блока был выбран двухслойный LSTM с размерностью скрытого состяния 40. Второй рекуррентный блок – LSTM с размерностью скрытого состояния 50 (количество тематик корпуса). В качестве оптимизатора использовался ADAM [29] с темпом обучения для рекуррентных блоков $\eta_r = 0.001$, для линейных слоев $\eta_l = 0.0005$. Инициализация весов для рекуррентных блоков – равномерная от -0.1 до 0.1, для линейных слоев – равномерное распределение для каждой из тематик. Модель обучалась в течение 50 итераций по обучающему [1] корпусу.

Предобработка словаря:

Слова заменены их векторными представлениями с помощью предобученных векторных представлений GloVe [30]. Удалены стоп-слова.

[1] Коллекция была разделена на обучающую и тестовую в отношении 9/1, перплексия измерялась на тестовой коллекции

Таблица 1. Пример тем, выделенных моделью с количеством тематик 50

Тема 1	Тема 2	Тема 3	Тема 4	Тема 5
law	democratic	stock	spending	auto
lawer	gop	price	sale	ford
right	senate	investor	advertis	car
attorney	district	standard	employe	british
insurance	party	trade	fiscal	american
court	act	hold	budget	model
mr	report	retirement	state	new
mrs	state	dollar	company	gm
session	american	taxe	buy	headquarter
general	population	money	validity	jaguar

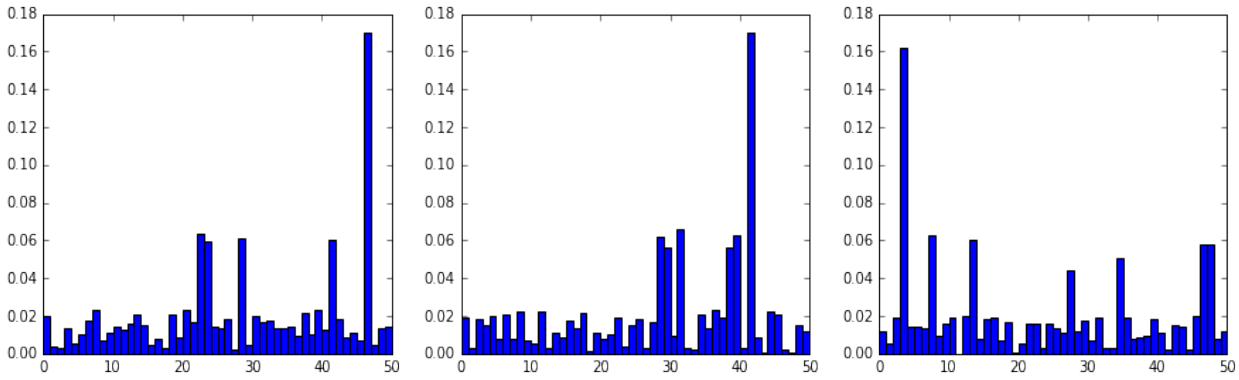

Рис. 7. Выделенные темы для трех случайных документов из коллекции PTB.

Таблица 2. Список архитектур и их тестовая перплексия на коллекции PTB.

Модель	**Перплексия**
KN5	125.7
RNN-LDA	113.7
Deep-RNN	107.5
Sum-Prod Net	100.0
Pointer Sentinel-LSTM (medium)	70.9
Variational LSTM (medium, untied)	78.4
Variational LSTM (medium, untied, MC)	79.7
Variational LSTM (large, untied)	78.6
VD-LSTM	68.5
Предложенная модель	**72.4**

Ссылки на модели: KN5, RNN-LDA [31]; Deep-RNN [32]; Sum-Prod Net [33], Pointer Sentinel-LSTM [34], Variational LSTM [35], VD-LSTM [36].

6.2. 20 Newsgroups

Набор данных 20 Newsgroups (20NG) представляет собой коллекцию из приблизительно 20 000 новостных документов, разделенных почти равномерно на 20 разных групп новостей. Коллекция 20NG является популярным набором данных для экспериментов с обработкой естественного языка, таких как классификация текстов и текстовая кластеризация.

20NG превосходит по размерам PTB. Поэтому модель на этом датасете обучалась без dropout-регуляризации. В качестве рекуррентных блоков использовались GRU и LSTM архитектуры.

Детали обучения: В модели в качестве первого рекуррентного блока был выбран один слой GRU с размерностью скрытого состояния 35. Второй рекуррентный блок – LSTM с размерностью скрытого состояния 20 (количество тематик коллекции). В качестве оптимизатора использовался ADAM с темпом обучения для рекуррентных блоков $\eta_r = 0.001$, для линейных слоев $\eta_l = 0.001$. Инициализация весов для рекуррентных блоков – равномерная от -0.1 до 0.1, для линейных слоев – равномерное распределение для каждой из тематик. Модель обучалась в течение 80 итераций по обучающему

корпусу.

Предобработка словаря: Слова заменены их векторными представлениями с помощью предобученных векторных представлений GloVe. Удалены стоп-слова.

Таблица 3. Пример тем, выделеных моделью с количеством тематик 20

Тема 1	Тема 2	Тема 3	Тема 4	Тема 5
hockey	ball	space	car	atheism
jock	bat	planet	lorry	god
penalty	catch	star	vehicle	apostle
major	coach	comet	engine	bible
board	dugout	asteroid	gear	rite
bullet	exhibition	astronaut	motor	christ
goal	game	spaceship	wheel	church
cage	foul	earth	hood	confession
assist	glove	satellite	shift	cult
minor	run	orbit	light	advent

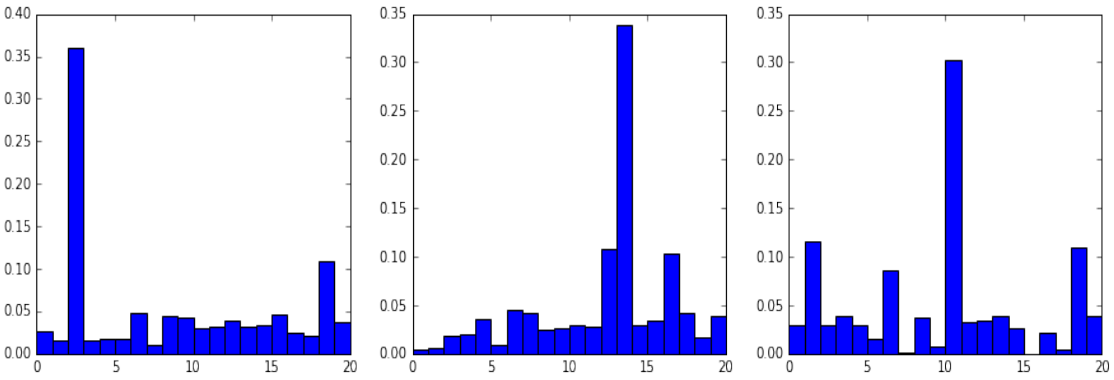

Рис. 8. Выделенные темы для трех случайных документов из коллекции 20 Newsgroup.

Модель	**Перплексия**
LDA	1247
NVLDA	1213
ProdLDA	1695
SenGen	1354
Предложенная модель	**1286**

Таблица 4. Сравнение перплексии предложенной модели. Список архитектур и их перплексия на датасете 20NG.

Ссылки на модели: LDA [2]; NVLDA, ProdLDA [37]; SenGen [38].

7. Заключение

В работе представлена модель тематической категоризации текстов, комбинирующая в себе методы тематического моделирования, хорошо определяющие глобальную семантику коллекции в целом и рекуррентных нейронных сетей, способных выделять локальную семантику. Достоинством предложенной модели является ее интерпретируемость. Интерпретируемость является трудно формализуемым понятием и имеет множество аспектов. В данной работе предлагается формализация, основанная на неотрицательности и нормированности групп весов сети, каждая из которых понимается как распределение (тема) над словарем фиксированной длины.

Проведенные в работе эксперименты показывают, что предложенная модель не уступает по качеству популярным методам тематического моделирования.

7.1. Результаты выносимые на защиту

- Предложена новая модель тематической категоризации коллекции документов
- Реализация предложенной модели с использованием фрэймворка PyTorch, проведение экспериментов

Список литературы

[1] Hoffmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning 42(1), 177–196 (2001)

[2] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research 3(4–5), 993–1022 (2003)

[3] Griffiths, T., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences 101 (Suppl. 1), 5228–5335 (2004)

[4] Andrzejewski, D., Zhu, X., Craven, M.: Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In: Proc. 26th Annual International Conference on Machine Learning. pp. 25–32. ICML '09, ACM, New York, NY, USA (2009)

[5] Andrzejewski, D., Zhu, X.: Latent Dirichlet allocation with topic-in-set knowledge. In: Proc. NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing. pp. 43–48. SemiSupLearn '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)

[6] Bodrunova, S., Koltsov, S., Koltsova, O., Nikolenko, S.I., Shimorina, A.: Interval semi-supervised lda: Classifying needles in a haystack. In: Proc. MICAI 2013, LNCS vol. 8625, pp. 265–274. Springer (2013)

[7] Nikolenko, S.I., Koltsova, O., Koltsov, S.: Topic modelling for qualitative studies. Journal of Information Science (2015)

[8] Tikhonov, A.N., Arsenin, V.Y.: Solution of ill-posed problems. W. H. Winston, Washington, DC (1977)

[9] Vorontsov, K.V., Potapenko, A.A.: Additive regularization of topic models. Machine Learning, Special Issue on Data Analysis and Intelligent Optimization with Applications 101(1), 303–323 (2015)

[10] Vorontsov, K.V., Potapenko, A.A.: Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization. In: AIST'2014, Springer CCIS vol. 436, pp. 29–46 (2014)

[11] Elman J. L. Finding structure in time // Cognitive science. — 1990. — Vol. 14, no. 2. — Pp. 179–211.

[12] Jordan M. I. Serial order: A parallel distributed processing approach // Advances in psychology. — 1997. — Vol. 121. — Pp. 471–495.

[13] Hochreiter S., Schmidhuber J. Long short-term memory. Neural computation. — 1997. — Vol. 9, no. 8. — Pp. 1735–1780.

[14] K. Cho. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259. — 2014.

[15] J. Chung. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. — 2014.

[16] John Lafferty, Andrew McCallum, and Fernando C.N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data . June 2001.

[17] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In Advances in Neural Information Processing Systems 16, 2004a.

[18] 27th Annual ACM Symposium on the Theoryof Computing, pp. 209218, ACM Press, New York, May 1995

[19] J. R. Hershey, J. Le Roux, and F. Weninger, “Deep unfolding:model-based inspiration of novel deep architectures,” arXiv:1409.2574, 2014.

[20] J. Le Roux, J. R. Hershey, and F. Weninger, “Deep NMF for speech separation,” in Proc. of International Conference on Acoustics, Speech, and Signal Processing, Brisbane, Australia, Apr. 2015, pp. 66–70.

[21] Dilated Recurrent Neural Networks, Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark Hasegawa-Johnson, Thomas S. Huang, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA

[22] A Novel Neural Topic Model and Its Supervised Extension, Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, Heng Ji. Twenty-Ninth AAAI Conference on Artificial Intelligence

[23] Topic-RNN: A recurrent neural network with long-range semantic dependency. Adji B. Dieng, Chong Wang, Jianfeng Gao, John Paisley. Conference paper at ICLR 2017.

[24] Autoencoding variational inference for topic models. Akash Srivastava, Charles Sutton. Conference paper at ICLR 2017.

[25] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. journal of machine learning research, 3(Feb):1137–1155, 2003.

[26] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur. Recurrent neural network based language model. In Interspeech, volume 2, page 3, 2010.

[27] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014.

[28] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In ICML, 2015.

[29] Diederik P. Kingma, Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. cs.LG, 2014.

[30] C. D. M. Jeffrey Pennington, Richard Socher. Glove: Global vectors for word representation. 2014

[31] Tomas Mikolov and Geoffrey Zweig. Context dependent recurrent neural network language model. In SLT, pp. 234–239, 2012.

[32] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026, 2013.

[33] Wei-Chen Cheng, Stanley Kok, Hoai Vu Pham, Hai Leong Chieu, and Kian Ming Adam Chai. Language modeling with sum-product networks. In INTERSPEECH, 2014.

[34] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843, 2016.

[35] Yarin Gal. A theoretically grounded application of dropout in recurrent neural networks. arXiv preprint arXiv:1512.05287, 2015.

[36] Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. arXiv preprint arXiv:1611.01462, 2016.

[37] Srivastava, A. and Sutton, C. Autoencoding Variational Inference For Topic Models. ArXiv e-prints, March 2017.

[38] Ramesh Nallapati, Igor Melnyk, Abhishek Kumar, Bowen Zhou. SenGen: Sentence Generating Neural Variational Topic Model