BREAKING THE SOFTMAX BOTTLENECK: A HIGH-RANK RNN LANGUAGE MODEL

Бурштеин Денис

МОП 162

Постановка задачи языкового моделирования

- По предыдущим словам (контексту) спрогнозировать следующее слово
- $X = (X_1, \ldots, X_T)$ - текст (корпус слов)
- $P(X) = \prod_t P(X_t \mid X_{<t}) = \prod_t P(X_t \mid C_t),\ C_t$ - контекст слова X_t
- Цель – смоделировать $P(X)$

Некоторые обозначения

- $\mathcal{L} = \{(c_1, P^*(X|c_1)), ..., (c_N, P^*(X|c_N))\}$ - естественный язык, пары (контекст, условное распределение на следующее слово)
- $\{x_1, ..., x_M\}$ - слова языка $\mathcal{L}$
- $P_\theta(X|C)$ - распределение, моделируемое нейросетью с параметрами θ
- $P^*(X|C)$ - истинное распределение

RNN с Softmax

- $P_\theta(x|c) = \frac{exp(h_c^T w_x)}{\sum_{x'} exp(h_c^T w_{x'})}$ - выход нейросети

- h_c - вектор контекста (hidden state), размерность d
- w_x - эмбеддинг слова, размерность d
- $h_c^T w_x$ - logit

Матричное представление

- $H_\theta \in \mathbb{R}^{N \times d}$, строки – векторы контекста
- $W_\theta \in \mathbb{R}^{M \times d}$, строки – эмбеддинги слов
- $A = \{A_{ij} = \log P^*(x_j|c_i)\}_{i,j=1}^{N,M}$ - логарифмы истинных вероятностей

- Введём множество $F(A) = \{A + \Lambda J_{N,M} | \Lambda \text{ is diagonal and } \Lambda \in \mathbb{R}^{N \times N}\}$

  $J_{N,M}$ - матрица размера $N \times M$, заполненная единицами

У полученного множества есть полезные свойства:

1) $A' \in F(A) \Leftrightarrow Softmax(A') = P^*$

Матрицы из этого множества – все возможные логиты, которые при подстановке в Softmax дают истинное распределение

2) $\forall\ A_1 \neq A_2 \in F(A), |rank(A_1) - rank(A_2)| \leq 1$

Ранги любых двух матриц из этого множества различаются не больше, чем на 1

Представление в виде матричного разложения

- Из первого свойства следует следующая Лемма:

$$H_\theta W_\theta^T \in F(A) \Leftrightarrow P_\theta(X|c) = P^*(X|c) \; \forall \; c \; in \; \mathcal{L}$$

- Таким образом, задача поиска оптимальных параметров θ представима как задача матричного разложения некоторой матрицы $A' \in F(A)$: $H_\theta W_\theta^T = A'$
- Исходя из размерностей матриц: $rank(H_\theta W_\theta^T) \leq d$
- Таким образом, если $d < rank(A')$, то невозможно обучить рассматриваемую нейросеть до реального распределения

Softmax bottleneck

- Из всех этих рассуждений следует проблема:

  Softmax bottleneck: Если $d < rank(A) - 1$, то для любой модели с Softmax на выходе и для любых её параметров θ существует такой контекст $c \in \mathcal{L}$, что: $P_\theta(X|c) \neq P^*(X|c)$

- Другими словами: при недостаточно большой размерности эмбеддингов Softmax на выходе ограничивает языковую модель, даже теоретически не давая обучиться до истинного распределения

Ранг естественного языка

- Предполагается, что естественный язык является высокоранговым
- Слишком большое количество значений для одного контекста
- Интуитивно кажется, что не существует некоторого базиса из нескольких сотен объектов, которыми можно было бы объяснить всё
- Так как размерность эмбеддингов, как правило, порядка сотен, возникает тот самый Softmax bottleneck

Размерность эмбеддингов

- Почему бы не увеличить размерность эмбеддингов до размера словаря M?

Размерность эмбеддингов

- Почему бы не увеличить размерность эмбеддингов до размера словаря M?
- Настолько большое количество параметров ($M \times M$ только для эмбеддингов) приводит к переобучению

Mixture of Softmaxes

- Для решения проблемы, Softmax заменяется на Mixture of Softmaxes:

$$P_\theta(x|c) = \sum_{k=1}^{K} \pi_{c,k} \frac{exp(h_{c,k}^T w_x)}{\sum_{x'} exp(h_{c,k}^T w_{x'})}, \quad \sum_{k=1}^{K} \pi_{c,k} = 1$$

$$\pi_{c_t,k} = \frac{exp(w_{\pi,k}^T g_t)}{\sum_{k'=1}^{K} exp(w_{\pi,k'}^T g_t)} \qquad h_{c_t,k} = \tanh(W_{h,k} g_t)$$

- $(g_1, ..., g_T)$ - hidden states; $w_{\pi,k}, W_{h,k}$ - параметры модели

Матричная форма MoS

$$\hat{A}_{MoS} = \log \sum_{k=1}^{K} \Pi_k exp(H_{\theta,k} W_\theta^T),\ \Pi_k = diag(\pi_{c_1,k}, ..., \pi_{c_N,k})$$

- Так как это нелинейная функция (log_sum_exp), она может быть сколько угодно высокоранговой, а значит Softmax bottleneck больше не проблема

Преимущества и недостатки MoS

- Преимущества:

1. Высокоранговость
2. Из-за высокоранговости можно существенно уменьшить размерность эмбеддингов, тем самым добившись того же количества параметров, что и с обычным Softmax, несмотря на новые параметры в модели

- Недостатки:

1. Существенное увеличение времени обучения из-за нескольких Softmax

Model	#Param	Validation	Test
Mikolov & Zweig (2012) – RNN-LDA + KN-5 + cache	9M‡	-	92.0
Zaremba et al. (2014) – LSTM	20M	86.2	82.7
Gal & Ghahramani (2016) – Variational LSTM (MC)	20M	-	78.6
Kim et al. (2016) – CharCNN	19M	-	78.9
Merity et al. (2016) – Pointer Sentinel-LSTM	21M	72.4	70.9
Grave et al. (2016) – LSTM + continuous cache pointer†	-	-	72.1
Inan et al. (2016) – Tied Variational LSTM + augmented loss	24M	75.7	73.2
Zilly et al. (2016) – Variational RHN	23M	67.9	65.4
Zoph & Le (2016) – NAS Cell	25M	-	64.0
Melis et al. (2017) – 2-layer skip connection LSTM	24M	60.9	58.3
Merity et al. (2017) – AWD-LSTM w/o finetune	24M	60.7	58.8
Merity et al. (2017) – AWD-LSTM	24M	60.0	57.3
Ours – AWD-LSTM-MoS w/o finetune	22M	58.08	55.97
Ours – AWD-LSTM-MoS	22M	**56.54**	**54.44**
Merity et al. (2017) – AWD-LSTM + continuous cache pointer†	24M	53.9	52.8
Krause et al. (2017) – AWD-LSTM + dynamic evaluation†	24M	51.6	51.1
Ours – AWD-LSTM-MoS + dynamic evaluation†	22M	**48.33**	**47.69**

Table 1: Single model perplexity on validation and test sets on Penn Treebank. Baseline results are obtained from Merity et al. (2017) and Krause et al. (2017). † indicates using dynamic evaluation.

Model	#Param	Validation	Test
Inan et al. (2016) – Variational LSTM + augmented loss	28M	91.5	87.0
Grave et al. (2016) – LSTM + continuous cache pointer†	-	-	68.9
Melis et al. (2017) – 2-layer skip connection LSTM	24M	69.1	65.9
Merity et al. (2017) – AWD-LSTM w/o finetune	33M	69.1	66.0
Merity et al. (2017) – AWD-LSTM	33M	68.6	65.8
Ours – AWD-LSTM-MoS w/o finetune	35M	66.01	63.33
Ours – AWD-LSTM-MoS	35M	**63.88**	**61.45**
Merity et al. (2017) – AWD-LSTM + continuous cache pointer †	33M	53.8	52.0
Krause et al. (2017) – AWD-LSTM + dynamic evaluation†	33M	46.4	44.3
Ours – AWD-LSTM-MoS + dynamical evaluation†	35M	**42.41**	**40.68**

Table 2: Single model perplexity over WikiText-2. Baseline results are obtained from Merity et al. (2017) and Krause et al. (2017). † indicates using dynamic evaluation.

Model	#Param	Train	Validation	Test
Softmax	119M	41.47	43.86	42.77
MoS	113M	**36.39**	**38.01**	**37.10**

Table 3: Perplexity comparison on 1B word dataset. Train perplexity is the average of the last 4,000 updates.

Сравнение рангов

Model	Validation	Test
Softmax	400	400
MoC	280	280
MoS	**9981**	**9981**

Table 6: Rank comparison on PTB. To ensure comparable model sizes, the embedding sizes of Softmax, MoC and MoS are 400, 280, 280 respectively. The vocabulary size, i.e., M, is 10,000 for all models.

#Softmax	Rank	Perplexity
3	6467	58.62
5	8930	57.36
10	9973	56.33
15	9981	55.97
20	9981	56.17

Table 7: Empirical rank and test perplexity on PTB with different number of Softmaxes.

Проверочные вопросы

- 1) Записать задачу поиска лучшей модели с обычным Softmax на выходе через матричное разложение.
- 2) Что из себя представляет проблема Softmax bottleneck (простыми словами).
- 3) Записать формулу Mixture of Softmaxes (без формул для весов и контекстных векторов).

Источники

- https://arxiv.org/pdf/1711.03953.pdf