SVD softmax approximation

Морозов Ярослав, Булгаков Арсений

14 декабря 2021 г.

Введение

В современном машинном обучении в задачах классификации широко используется функция Softmax. Однако, эту функцию довольно затратно вычислять. В нашей работе демонстрируется одно из применений SVD для вычисления функции Softmax, которое позволяет значительно ускорить вычисления

`https://github.com/Yar4ik000/NLA_Ozon`

Softmax

Функция softmax преобразует D-мерный вещественный вектор h в V-мерное распределение вероятностей. Расчет вероятности состоит из двух этапов. Сначала мы получаем выходной вектор размера V, обозначаемый как z, из h путем умножения матрицы как:

$z = Ah + b$, где A - матрица весов, b - вектор сдвига.
Затем мы нормализуем выходной вектор z для подсчета вероятностей каждого слова y_k

$$y_k = softmax(z_k) = \frac{exp(A_k h + b_k)}{\sum_{i=1}^{V} exp(A_i h + b_i)} = \frac{exp(z_k)}{\sum_{i=1}^{V} exp(z_i)}$$

SVD softmax

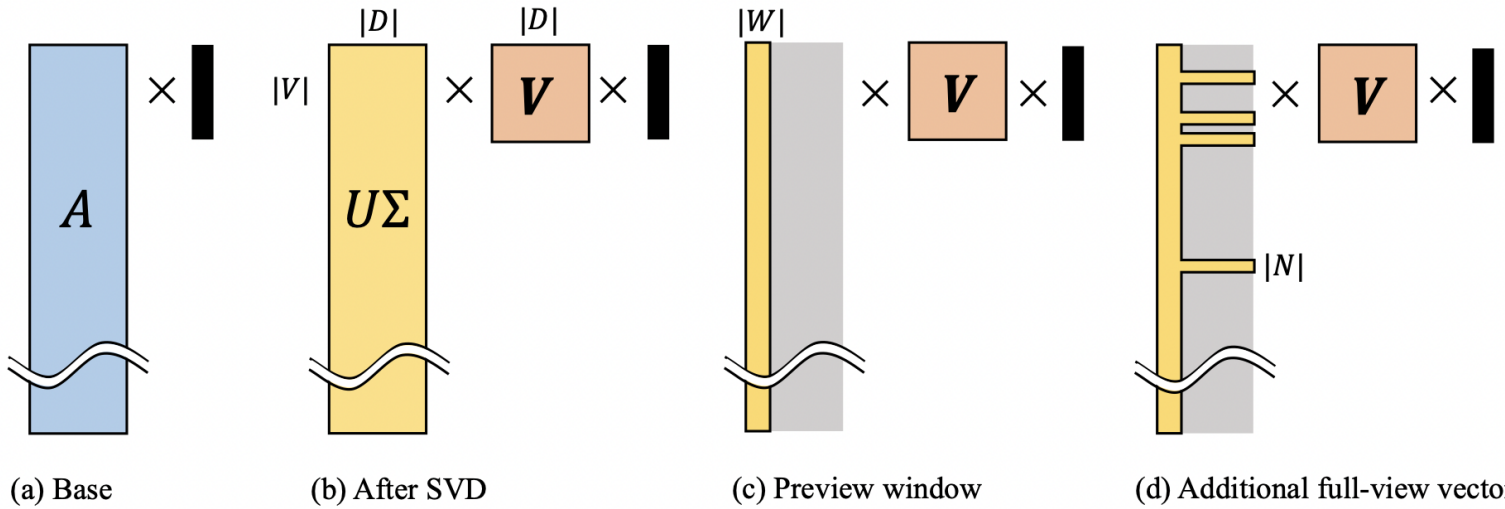
SVD позволяет записать матрицу весов в виде $A = U\Sigma V^T$.
Перемножим U и Σ, тем самым получим факторизацию исходной матрицы A на две части: $U\Sigma$ и V^T
Большие сингулярные числа в Σ умножаются на крайние левые столбцы U. В результате элементы матрицы $B(= U\Sigma)$ статистически упорядочены в порядке убывания, от первого столбца к последнему. Крайние левые столбцы B имеют большее влияние, чем крайние правые столбцы.

SVD softmax

Algorithm 1 Algorithm of the proposed SVD-softmax.

input: trained weight matrix $\mathbf{A}$, input vector $\mathbf{h}$, bias vector $\mathbf{b}$
hyperparameter: width of preview window W, number of full-view vectors N.
initialize: decompose $\mathbf{A} = \mathbf{U\Sigma V}^T$, $\mathbf{B} = \mathbf{U\Sigma}$
$\tilde{\mathbf{h}} = \mathbf{V}^\mathbf{T} \times \mathbf{h}$
$\tilde{\mathbf{z}} = \mathbf{B}[:, :W] \times \tilde{\mathbf{h}}[:W] + \mathbf{b}$ compute preview outputs with only W dimensions
Sort $\tilde{\mathbf{z}}$ in descending order select N words of largest preview outputs
$\mathbf{C}_N$ = Top-N word indices of $\tilde{\mathbf{z}}$
for all id in $\mathbf{C}_N$ **do**
    $\tilde{\mathbf{z}}[id] = \mathbf{B}[id, :] \times \tilde{\mathbf{h}} + \mathbf{b}[id]$ update selected words by full-view vector multiplication
end for
$\tilde{Z} = \sum_V \exp \tilde{z}_i$
$\tilde{\mathbf{y}} = \exp(\tilde{\mathbf{z}})/\tilde{Z}$ compute probability distribution using softmax
return $\tilde{\mathbf{y}}$

SVD softmax



(a) Base (b) After SVD (c) Preview window (d) Additional full-view vectors

Figure 1: Illustration of the proposed SVD-softmax algorithm. The softmax weight matrix is decomposed by singular value decomposition (b). Only a part of the columns is used to compute the preview outputs (c). Selected rows, which are chosen by sorting the preview outputs, are recomputed with full-width (d). For simplicity, the bias vector is omitted.

Complexity

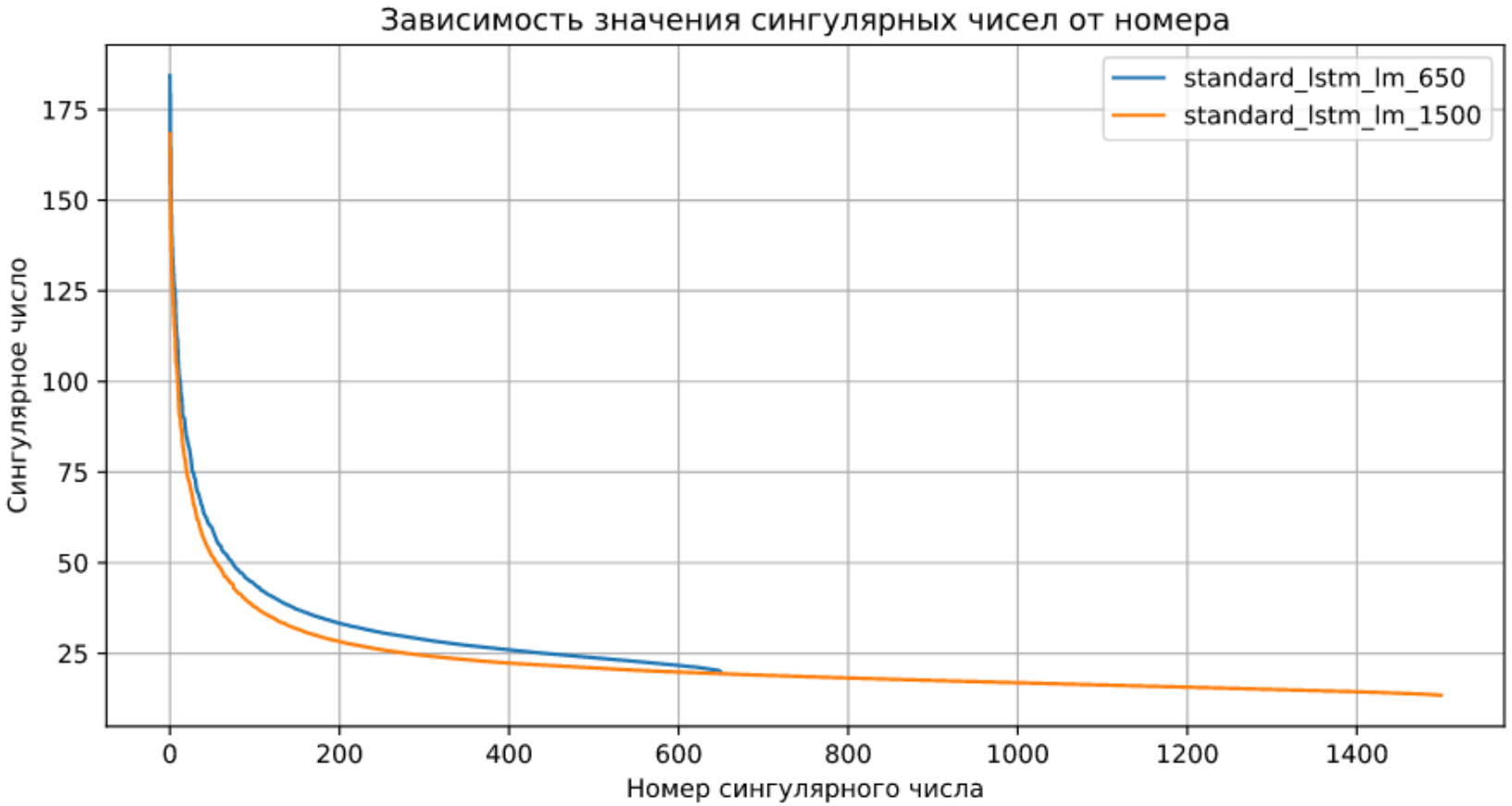
Благодаря апроксимации с помощью SVD сложность уменьшается с $O(VD)$ до $O(VW + ND)$, где

- V - размер словаря
- D - скрытая размерность
- W - размер окна
- N - количество слов

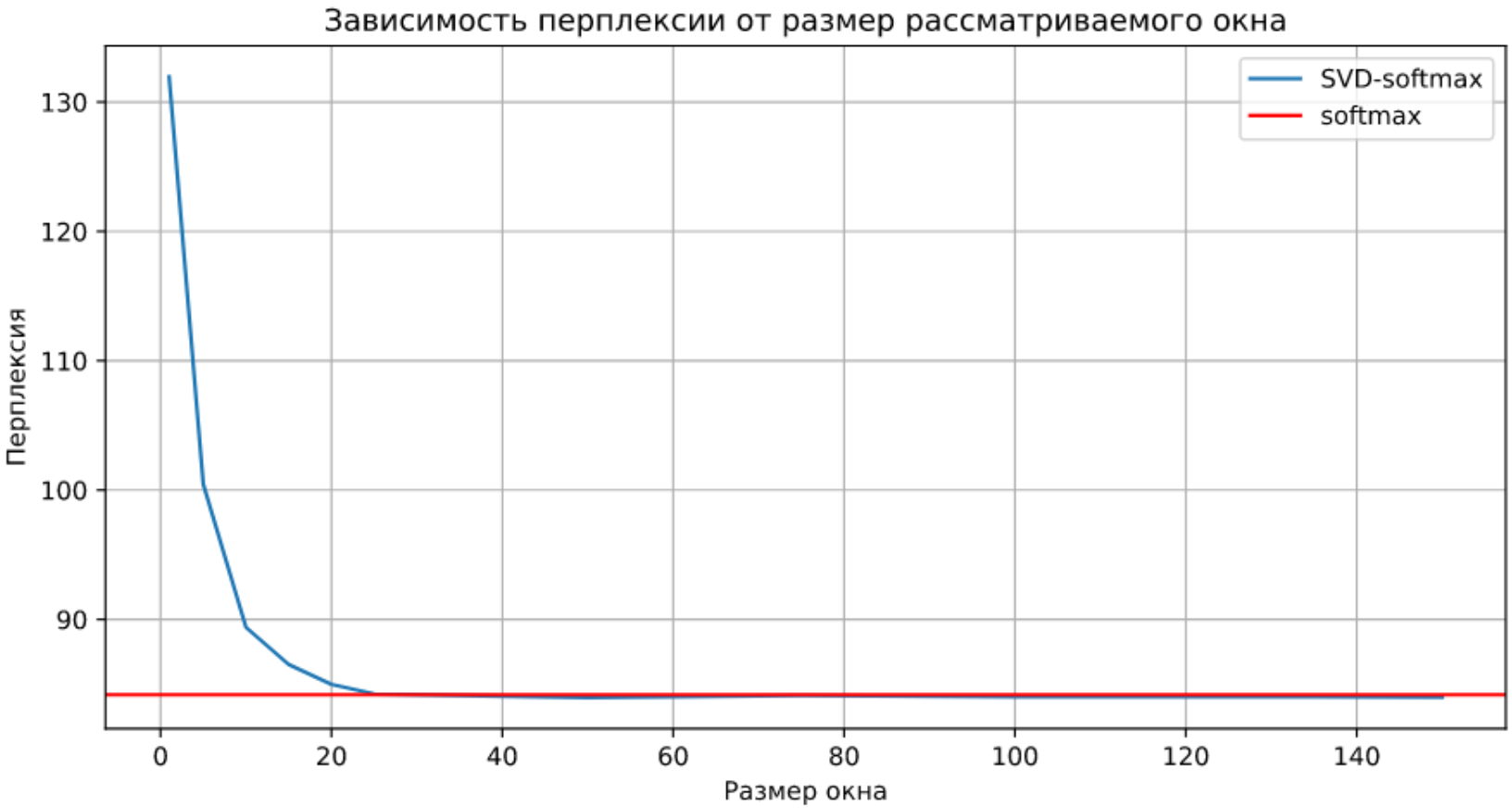
Experiments

- Датасет - wikitext2
- Продемонстрировали резкое уменьшение сингулярных чисел в матрице весов
- Сравнили обычный softmax и svd softmax

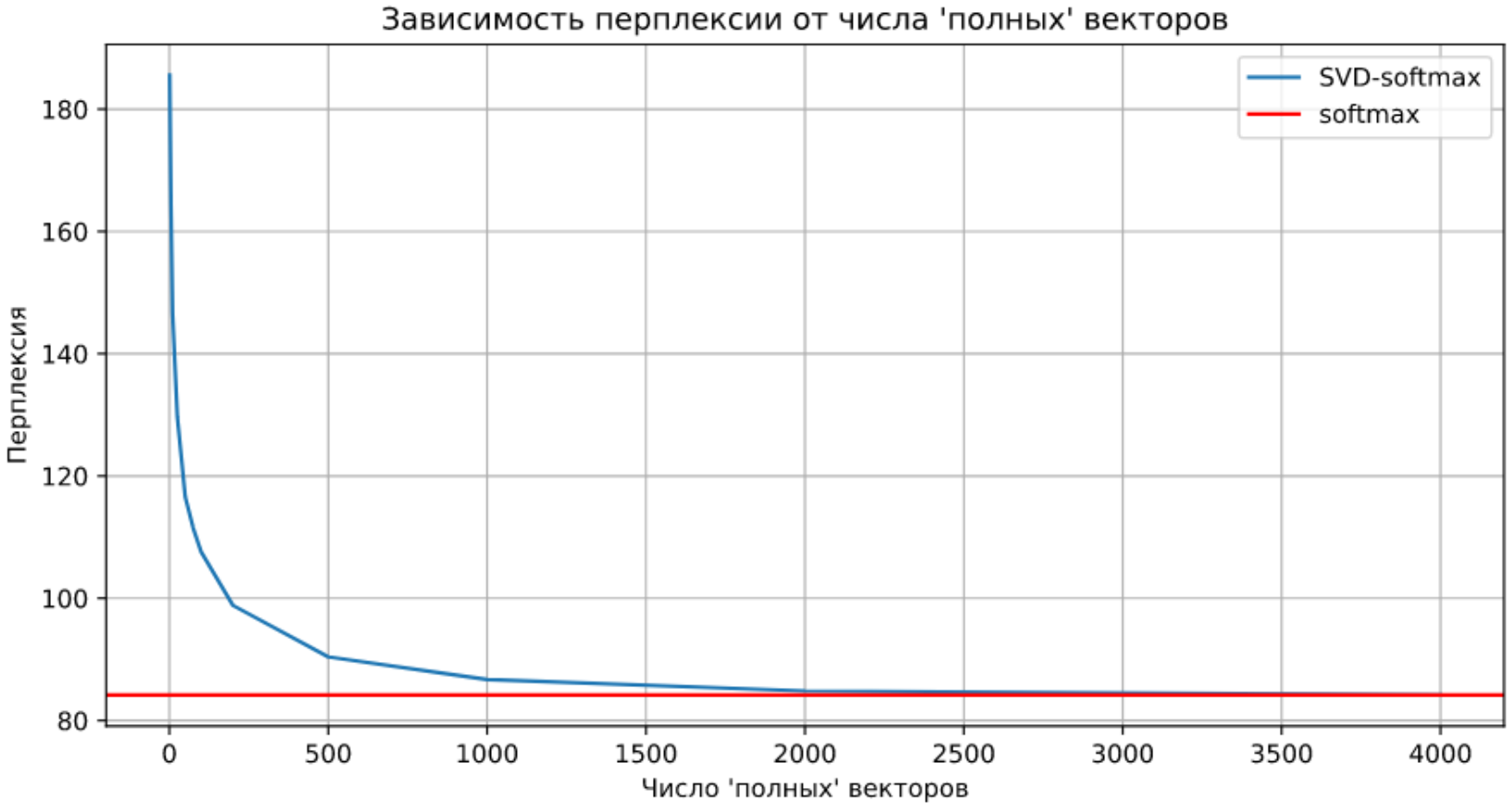
Experiments



Experiments



Experiments



Summary

Апроксимируя softmax с помощью SVD, время вычисления уменьшилось примерно в 3 раза, при этом при достаточной ширине окна и числе "полных векторов"качество примерно одинаковое.

Основная статья

https://proceedings.neurips.cc/paper/2017/file/4e2a6330465c8ffcaa696a5a16639176-Paper.pdf