Разбор статьи "Optimal Brain Compression: A Framework for Accurate Post-Training Quantization and Pruning"

Давыденко Григорий, Шалыгин Игорь

Московский физико-технический институт

Ноябрь 2024 г.

Предыстория проблемы

Плюсы сжатия нейросетей:

1. Снижение затрат по памяти.
2. Понижение требуемых вычислительных мощностей.

Минусы сжатия нейросетей:

1. Снижение точности.
2. Дополнительные затраты на сжатие.

Методы сжатия нейронных сетей

Прунинг

Прунинг - это сжатие путем уменьшения количества параметров нейросети. Алгоритм ищет наименее важные параметры нейросети и удаляет их, превращая нейросеть в разряженную таблицу.

Квантование

Квантование - это сжатие путем понижения точности параметров нейросети. Алгоритм уменьшает количество памяти, отведенное на нейроны, снижая вес сети и упрощая матетематические операции.

Методы сжатия нейронных сетей

Алгоритмы прунинга:

- OBD, OBS, L-OBS, AdaPrune

Алгоритмы квантования:

- BitSplit, AdaRound, AdaQuant, BRECQ

Паттерны сжатия:

- N:M, Block sparsity

Постановка задачи в форме OBS

1. Ставим задачу оптимального обновления весов в строке.

$$f : \mathbb{R}^n \to \mathbb{R}, \; f(\hat{W}) = \|WX - \hat{W}X\|_2^2 \to min$$

2. Раскладываем лосс в ряд до 2 порядка в W.

$$\delta_p := \hat{W} - W$$

$$f(\hat{W}) \approx f(W) + \nabla f(W)^T \delta_p + \frac{1}{2}\delta_p^T H(W)\delta_p = \frac{1}{2}\delta_p^T H(W)\delta_p$$

3. Формируем задачу с ограничением и решаем.

$$\min_{\delta_p} \frac{1}{2}\delta_p^T H\delta_p$$

$$\text{s.t. } e_p^T \delta_p + w_p = 0$$

4. Находим вес, вносящий наименьший вклад.

$$w_p = \text{argmin}_{w_p}\{\delta_p^T H\delta_p \mid e_p^T \delta_p + w_p = 0\} = \text{argmin}_{w_p} \frac{w_p^2}{[H^{-1}]_{pp}}$$

Решение задачи в форме OBS

$$\min_{\delta_p} \frac{1}{2}\delta_p^T H \delta_p$$

$$\text{s.t. } e_p^T \delta_p + w_p = 0$$

Выпуклая функция с аффинным ограничением - KKT:

$$L(\delta_p, \lambda) = \frac{1}{2}\delta_p^T H \delta_p + \lambda(e_p^T \delta_p + w_p)$$

1. $\frac{\partial L}{\partial \delta_p} = H\delta_p + \lambda e_p = 0 \Rightarrow \delta_p = -\lambda H^{-1} e_p$
2. $e_p^T \delta_p + w_p = 0$

Подставляем δ_p во 2 равенство и выражаем λ: $\lambda = \frac{w_p}{e_p^T H^{-1} e_p} = \frac{w_p}{[H^{-1}]_{pp}}$

Подставляя λ в выражение для δ_p, получаем:

$$\delta_p = -\frac{w_p}{[H^{-1}]_{pp}}[H^{-1}]_{:,p}, \quad w_p = \text{argmin}_{w_p} \frac{w_p^2}{[H^{-1}]_{pp}}$$

ExactOBS

Переход к Гессианам для строк

Положим размерность $W = d_{row} \times d_{col}$,

$$\|WX - \hat{W}X\|_2^2 = \sum_{i=1}^{d_{col}} \|W_{i,:}X - \hat{W}_{i,:}X\|_2^2$$

Лемма (об обновлении Гессиана)

После удаления строки обратный Гессиан можно найти по следующей формуле:

$$H_{-p}^{-1} = (H^{-1} - \frac{1}{[H^{-1}]_{pp}}[H^{-1}]_{:,p}[H^{-1}]_{p,:})_{-p}$$

Замечание: $H = 2(XX^T)$

ExactOBS

Algorithm 1 Удаление $k \leq d_{col}$ весов из строки $\mathbf{w}$ с обращенным Гессианом $\mathbf{H}^{-1} = (2\mathbf{X}\mathbf{X}^T)^{-1}$ за время $\mathcal{O}(d_{col} \cdot k^2)$

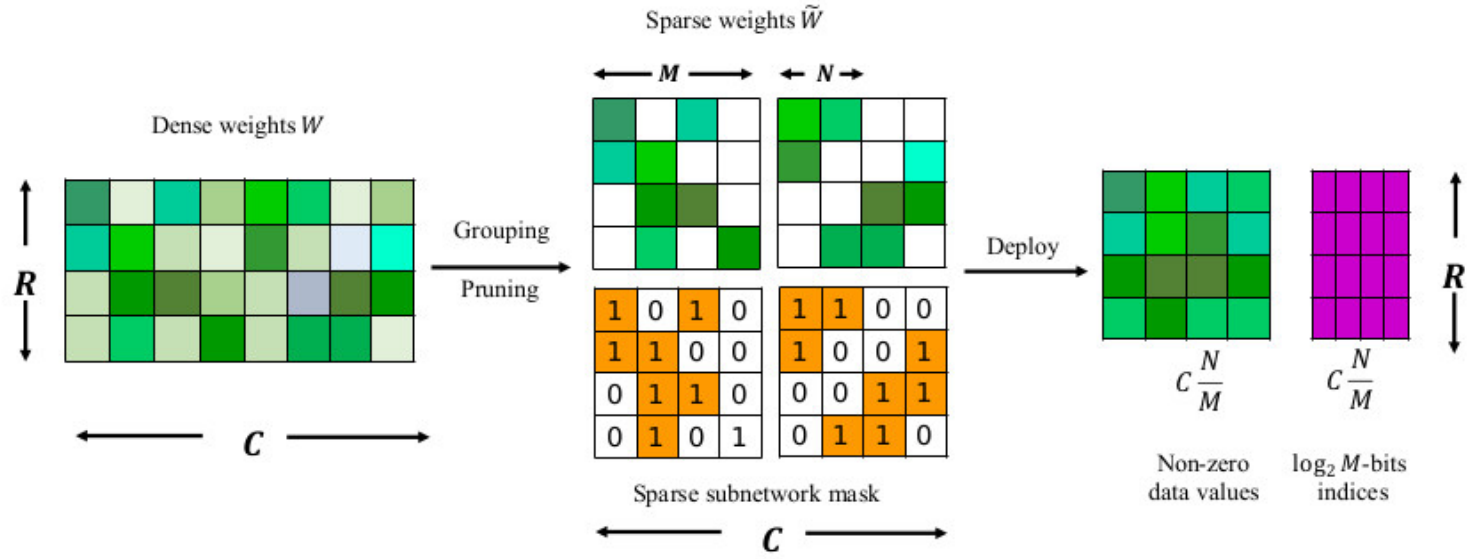
$M = \{1, ..., d_{col}\}$
for $i = 1, ..., k$ **do**
  $p \leftarrow \text{argmin}_{p \in M} \frac{w_p^2}{[H^{-1}]_{pp}}$
  $\mathbf{w} \leftarrow \mathbf{w} - \frac{w_p}{[H^{-1}]_{pp}} [H^{-1}]_{:,p}$
  $\mathbf{H}^{-1} \leftarrow \mathbf{H}^{-1} - \frac{1}{[\mathbf{H}^{-1}]_{pp}} [\mathbf{H}^{-1}]_{:,p} [\mathbf{H}^{-1}]_{p,:}$
  $M \leftarrow M - \{p\}$
end for

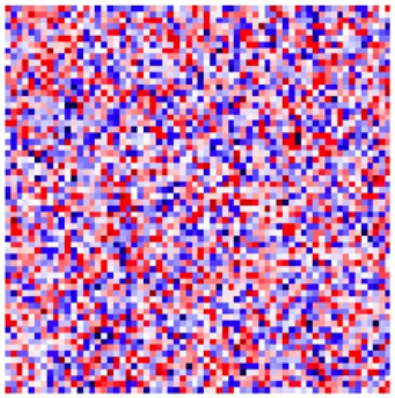
Итоговая формула обновления

Пусть M_i - маска для i-й строки,

$$\delta_{M_i} = -[\mathbf{H}^{-1}]_{:,M_i} ([\mathbf{H}^{-1}]_{M_i})^{-1} \mathbf{w}_{M_i}$$

ExactOBS





Dense weights

Block-sparse weights

0	0	1	1	1	1	1	1
1	1	1	0	0	1	1	0
1	0	0	0	0	0	1	1
1	1	1	1	0	0	1	1
1	0	1	1	1	1	1	1
0	1	0	0	0	0	0	0
1	1	1	0	1	1	0	0
1	0	0	0	0	1	1	1

Corresponding sparsity pattern

Optimal Brain Quantizer

Задача квантования

quant(w_p) - значение веса после квантования,

$$L(\delta_p, \lambda) = \frac{1}{2}\delta_p^T H \delta_p + \lambda(e_p^T \delta_p + w_p - quant(w_p)))$$

$$w_p = \text{argmin}_{w_p} \frac{(w_p - quant(w_p))^2}{[H^{-1}]_{pp}}, \quad \delta_p = -\frac{w_p - quant(w_p)}{[H^{-1}]_{pp}} [H^{-1}]_{:,p}$$

Optimal Brain Quantizer

Algorithm 2 Модификация алгоритма прунинга для задачи квантования

$M = \{1, ..., d_{col}\}$

for $i = 1, ..., k$ **do**

$p \leftarrow \text{argmin}_{p \in M} \frac{1}{[H^{-1}]_{pp}} \cdot (quant(w_p) - w_p)^2$

$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{[H^{-1}]_{pp}} [H^{-1}]_{:,p} \cdot (w_p - quant(w_p))$

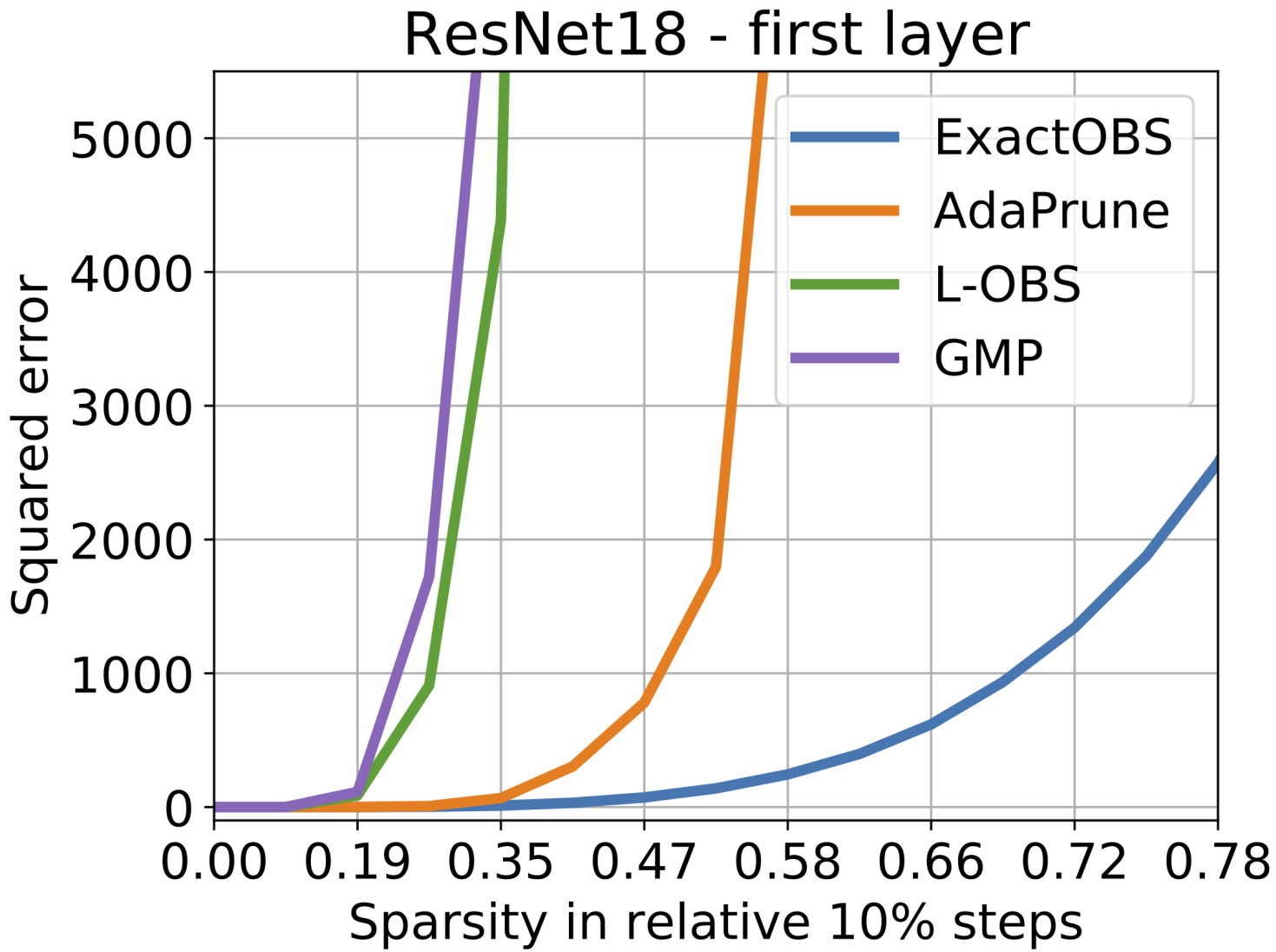
$\mathbf{H}^{-1} \leftarrow \mathbf{H}^{-1} - \frac{1}{[\mathbf{H}^{-1}]_{pp}} [\mathbf{H}^{-1}]_{:,p} [\mathbf{H}^{-1}]_{p,:}$

$M \leftarrow M - \{p\}$

end for

Эксперименты 1

ResNet18 квадратичная ошибка в зависимости от алгоритма



Эксперименты 2

Точность прунинга в зависимости от уменьшения FLOP

	Method	ResNet50			BERT		
		2 ×	3 ×	4 ×	2 ×	3 ×	4 ×
Paper	Dense	76.13			88.53		
	GMP	74.86	71.44	64.84	65.64	12.52	9.23
	L-OBS	75.48	73.73	71.24	77.67	3.62	6.63
	AdaPrune	75.53	74.47	72.39	87.12	70.32	18.75
	ExactOBS	75.64	75.01	74.05	87.81	85.87	82.10
Our results	Dense	75.58					
	ExactOBS	74.20	74.33	73.39			

Результат:

1. Только ExactOBS показывает хорошие результаты сжатия BERT

Эксперименты 3

Точность ResNet в зависимости от N:M сжатия и алгоритма

Model	**Paper results**				**Our results**		
	Dense	AdaPrune	ExactOBS		Dense	ExactOBS	
		4:8	2:4	4:8		2:4	4:8
ResNet18	69.76	68.63	68.81	69.18	70.00	68.70	69.36
ResNet34	73.31	72.36	72.66	72.95	71.93	70.97	71.37
ResNet50	76.13	74.75	74.71	75.20	75.58	74.28	75.04

Результат:

1. 2:4 прунинг ExactOBS $>$ 4:8 прунинг AdaPrune
2. Более строгая схема 2:4 хорошо поддерживается NVIDIA

Эксперименты 4

Точность 2:4 сжатия BERT в зависимости от алгоритма

	Paper results			Our results	
Model	Dense	AdaPrune	ExactOBS	Dense	ExactOBS
BERT 3	84.66	82.75	83.54	84.64	83.44
BERT 6	88.33	85.02	86.97	88.33	86.94
BERT	88.53	85.24	86.77	88.55	86.73

Результат:

1. Наш эксперимент подтвердил результаты из статьи
2. Качество ExactOBS выше на 1-2 процента, чем у AdaPrune

Эксперименты 5

Точность после ассиметричного квантования слоев

	Method	ResNet18			ResNet50		
		4bit	3bit	2bit	4bit	3bit	2bit
Paper	Dense	69.76			76.13		
	AdaRound	69.34	68.37	63.37	75.84	75.14	71.58
	AdaQuant	68.12	59.21	00.10	74.68	64.98	00.10
	BRECQ	69.37	68.47	64.70	75.88	75.32	72.41
	OBQ	69.56	68.69	64.04	75.72	75.24	70.71
Our results	Dense	70.00			75.58		
	OBQ	69.44	68.21	63.65	75.10	74.51	70.25

Приложение: доказательство леммы об обновлении

Лемма (об обновлении Гессиана)

$$H^{-1}_{-p} = (H^{-1} - \frac{1}{[H^{-1}]_{pp}}[H^{-1}]_{:,p}[H^{-1}]_{p,:})_{-p}$$

План доказательства:

1. $A \to A'$

$$\begin{pmatrix} A_1 & a_1 & A_2 \\ a_2^T & A_{ii} & a_3^T \\ A_3 & a_4 & A_4 \end{pmatrix} - \frac{1}{A_{ii}} A_{:,i} \cdot A_{i,:} = \begin{pmatrix} A'_1 & 0 & A'_2 \\ 0^T & 0 & 0^T \\ A'_3 & 0 & A'_4 \end{pmatrix}$$

2. $AB = I \to (A - \frac{1}{A_{ii}} A_{:,i} \cdot A_{i,:})B = I - \frac{1}{A_{ii}} A_{:,i} \cdot A_{i,:} \cdot B \to A'B = C$

$$\begin{pmatrix} A'_1 & 0 & A'_2 \\ 0^T & 0 & 0^T \\ A'_3 & 0 & A'_4 \end{pmatrix} \cdot \begin{pmatrix} B_1 & b_1 & B_2 \\ b_2^T & B_{ii} & b_3^T \\ B_3 & b_4 & B_4 \end{pmatrix} = \begin{pmatrix} I & c_1 & 0 \\ c_2^T & C_{ii} & c_2^T \\ 0 & c_4 & I \end{pmatrix}$$

3. $A_{-i} \cdot B_{-i} = I$

Приложение: ссылки

- Optimal Brain Compression (текущая статья)
- Репозиторий с экспериментами
- Optimal Brain Damage
- Optimal Brain Surgeon
- SparseGPT
- GPTQ