BEYOND FULLY-CONNECTED LAYERS WITH QUATERNIONS

Докладчик: Рахматуллин Рамазан
Рецензент: Сафонов Иван
Практик-исследователь: Медведев Антон
Хакер: Степанов Никита

Описание

- Метод для уменьшения потребляемой памяти в Fully-Connected в произвольное целое число (<=10) раз без значимой потери качества и скорости
- Идея: обобщаем произведение кватернионов (4D чисел)
- Пример произведения для 2D (комплексные числа)

$$i^2 = -1$$

$$(a + bi) \cdot (c + di) = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \cdot \begin{pmatrix} c \\ d \end{pmatrix}$$

Quaternion A Quaternion $Q \in \mathbb{H}$ is a hypercomplex number with one real component and three imaginary components as follows:

$$Q = Q_r + Q_x\mathbf{i} + Q_y\mathbf{j} + Q_z\mathbf{k}, \tag{2.1}$$

whereby $\mathbf{ijk} = \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$. In (2.1), noncommutative multiplication rules hold: $\mathbf{ij} = \mathbf{k}, \mathbf{jk} = \mathbf{i}, \mathbf{ki} = \mathbf{j}, \mathbf{ji} = -\mathbf{k}, \mathbf{kj} = -\mathbf{i}, \mathbf{ik} = -\mathbf{j}$. Here, Q_r is the real component, Q_x, Q_y, Q_z are real numbers that represent the imaginary components of the Quaternion Q.

$$\begin{bmatrix} Q_r & -Q_x & -Q_y & -Q_z \\ Q_x & Q_r & -Q_z & Q_y \\ Q_y & Q_z & Q_r & -Q_x \\ Q_z & -Q_y & Q_x & Q_r \end{bmatrix} \begin{bmatrix} P_r \\ P_x \\ P_y \\ P_z \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_1} \otimes \underbrace{\begin{bmatrix} Q_r \end{bmatrix}}_{\mathbf{S}_1} + \underbrace{\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{A}_2} \otimes \underbrace{\begin{bmatrix} Q_x \end{bmatrix}}_{\mathbf{S}_2} + \underbrace{\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_3} \otimes \underbrace{\begin{bmatrix} Q_y \end{bmatrix}}_{\mathbf{S}_3} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_4} \otimes \underbrace{\begin{bmatrix} Q_z \end{bmatrix}}_{\mathbf{S}_4}$$

(3.5)

Кватернионы

- Если на вход (вектор P) больше чем 4 числа, то разобьем их на 4 примерно равных отрезка, а каждое Q_r, Q_x, Q_y, Q_z заменим на матрицы

$$\begin{bmatrix} Q_r & -Q_x & -Q_y & -Q_z \\ Q_x & Q_r & -Q_z & Q_y \\ Q_y & Q_z & Q_r & -Q_x \\ Q_z & -Q_y & Q_x & Q_r \end{bmatrix} \begin{bmatrix} P_r \\ P_x \\ P_y \\ P_z \end{bmatrix}$$

Предлагаемый метод

Произведение Кронекера

$$X \in R^{n \times m},\ Y \in R^{p \times q} \qquad \mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} x_{11}\mathbf{Y} & \dots & x_{1n}\mathbf{Y} \\ \vdots & \ddots & \vdots \\ x_{m1}\mathbf{Y} & \dots & x_{mn}\mathbf{Y} \end{bmatrix} \in R^{np \times mq}$$

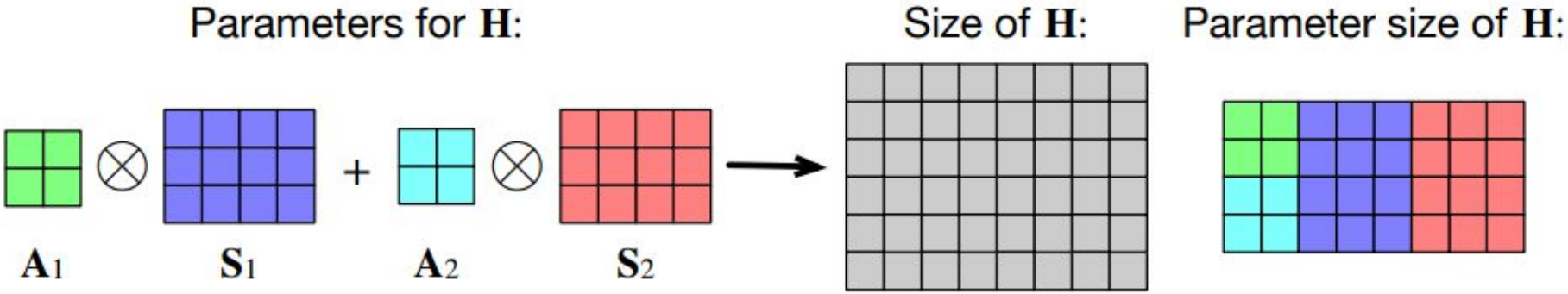
$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \otimes \begin{bmatrix} x & y \\ z & s \end{bmatrix} = \begin{bmatrix} ax & ay & bx & by & cx & cy \\ az & as & bz & bs & cz & cs \\ dx & dy & ex & ey & fx & fy \\ dz & ds & ez & es & fz & fs \end{bmatrix}$$

Предлагаемый метод

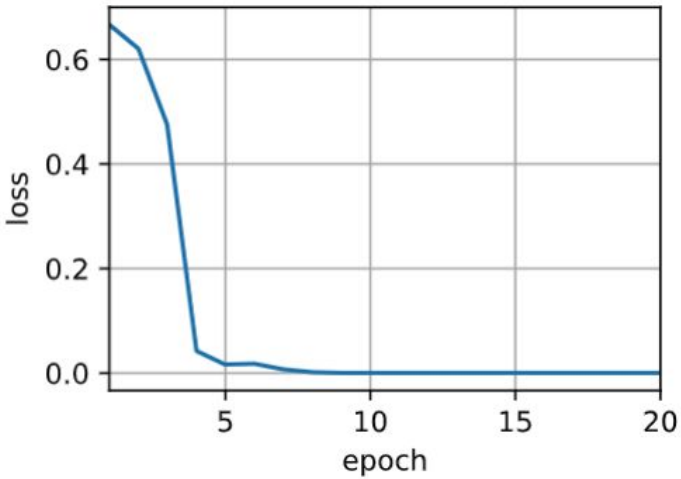
Хотим построить матрицу H размеров (k x d), пусть k и d делят n

$$\mathbf{H} = \sum_{i=1}^{n} \mathbf{A}_i \otimes \mathbf{S}_i \qquad A_i \in \mathbb{R}^{n \times n},\ S_i \in \mathbb{R}^{\frac{k}{n} \times \frac{d}{n}}$$

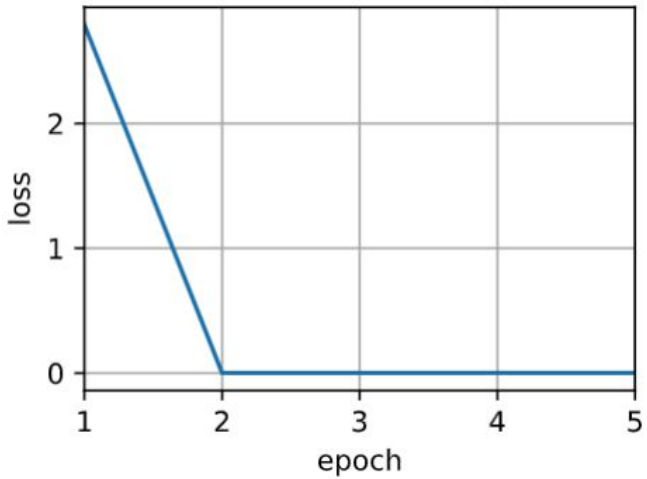
Вместо k*d параметров стало n^3 + kd/n = O(kd/n): d = 512, k = 2048, n <= 16



Эксперименты



(a) Learning rotations in 3D real space



(b) Learning Hamilton products in Quaternion space

Эксперименты

Table 1: Experimental results of natural language inference (accuracy) on five different datasets. The PHM-LSTM reduces the parameters of the standard LSTM model and improves or partially matches performance on four out of five datasets.

Model	#Params	MNLI	QNLI	SNLI	DNLI	SciTail
LSTM	721K	**71.82** / 71.89	84.44	84.18	85.16	74.36
Quaternion LSTM	180K (-75.0%)	71.57 / **72.19**	**84.73**	84.21	86.45	75.58
PHM-LSTM ($n = 2$)	361K (-49.9%)	**71.82** / 72.08	84.39	84.38	85.77	77.47
PHM-LSTM ($n = 5$)	146K (-79.7%)	71.80 / 71.77	83.87	**84.58**	**86.47**	74.64
PHM-LSTM ($n = 10$)	81K (-88.7%)	71.59 / 71.59	84.25	84.40	86.21	**77.84**

Table 2: Experimental results of machine translation (BLEU) on seven different datasets. Symbol † represents re-scaling the parameters with a factor of 2 by doubling the hidden size. The PHM-Transformer does not lose much performance despite enjoying parameter savings. Re-scaling can lead to improvement in performance.

Model	#Params	En-Vi	En-Id	De-En	Ro-En	En-Et	En-Mk	En-Ro
Transformer (Tm)	44M	28.43	47.40	**36.68**	**34.60**	14.17	13.96	**22.79**
Quaternion Tm	11M (-75.0%)	28.00	42.22	32.83	30.53	13.10	13.67	18.50
PHM-Tm $n = 2$	22M (-50.0%)	29.25	46.32	35.52	33.40	**14.98**	13.60	21.73
PHM-Tm $n = 4$	11M (-75.0%)	29.13	44.13	35.53	32.74	14.11	13.01	21.19
PHM-Tm $n = 8$	5.5M (-87.5%)	29.34	40.81	34.16	31.88	13.08	12.95	21.66
PHM-Tm $n = 16$	2.9M (-93.4%)	29.04	33.48	33.89	31.53	12.15	11.97	19.63
PHM-Tm† $n = 2$	44M	**29.54**	**49.05**	34.32	33.88	14.05	**14.41**	22.18
PHM-Tm† $n = 4$	22M (-50.0%)	29.17	46.24	34.86	33.80	14.43	13.78	21.91
PHM-Tm† $n = 8$	11M (-75.0%)	29.47	43.49	34.71	32.59	13.75	13.78	21.43

Table 3: Training time (seconds per 100 steps) and inference time (seconds to decode test sets) with beam size of 4 and length penalty of 0.6 on the IWSLT'14 German-English dataset.

Model	Transformer (Tm)	Quaternion Tm	PHM-Tm ($n = 4$)	PHM-Tm ($n = 8$)
Training time	**7.61**	8.11	7.92	7.70
Inference time	336	293	299	**282**

Table 4: Experimental results of text style transfer. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	BLEU
Transformer (Tm)	44M	11.65
PHM-Tm ($n = 2$)	22M (-50.0%)	12.20
PHM-Tm ($n = 4$)	11M (-75.0%)	**12.42**
PHM-Tm ($n = 8$)	5.5M (-87.5%)	11.66
PHM-Tm ($n = 16$)	2.9M (-93.4%)	10.76

Table 5: Experimental results of subject verb agreement. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	Acc
Transformer (Tm)	400K	94.80
Quaternion Tm	100K	94.70
PHM-Tm ($n = 2$)	200K (-50.0%)	95.14
PHM-Tm ($n = 4$)	101K (-74.8%)	95.05
PHM-Tm ($n = 8$)	56K (-86.0%)	**95.62**

Эксперименты

Table 1: Experimental results of natural language inference (accuracy) on five different datasets. The PHM-LSTM reduces the parameters of the standard LSTM model and improves or partially matches performance on four out of five datasets.

Model	#Params	MNLI	QNLI	SNLI	DNLI	SciTail
LSTM	721K	**71.82** / 71.89	84.44	84.18	85.16	74.36
Quaternion LSTM	180K (-75.0%)	71.57 / **72.19**	**84.73**	84.21	86.45	75.58
PHM-LSTM ($n = 2$)	361K (-49.9%)	**71.82** / 72.08	84.39	84.38	85.77	77.47
PHM-LSTM ($n = 5$)	146K (-79.7%)	71.80 / 71.77	83.87	**84.58**	**86.47**	74.64
PHM-LSTM ($n = 10$)	81K (-88.7%)	71.59 / 71.59	84.25	84.40	86.21	**77.84**

Table 2: Experimental results of machine translation (BLEU) on seven different datasets. Symbol † represents re-scaling the parameters with a factor of 2 by doubling the hidden size. The PHM-Transformer does not lose much performance despite enjoying parameter savings. Re-scaling can lead to improvement in performance.

Model	#Params	En-Vi	En-Id	De-En	Ro-En	En-Et	En-Mk	En-Ro
Transformer (Tm)	44M	28.43	47.40	**36.68**	**34.60**	14.17	13.96	**22.79**
Quaternion Tm	11M (-75.0%)	28.00	42.22	32.83	30.53	13.10	13.67	18.50
PHM-Tm $n = 2$	22M (-50.0%)	29.25	46.32	35.52	33.40	**14.98**	13.60	21.73
PHM-Tm $n = 4$	11M (-75.0%)	29.13	44.13	35.53	32.74	14.11	13.01	21.19
PHM-Tm $n = 8$	5.5M (-87.5%)	29.34	40.81	34.16	31.88	13.08	12.95	21.66
PHM-Tm $n = 16$	2.9M (-93.4%)	29.04	33.48	33.89	31.53	12.15	11.97	19.63
PHM-Tm† $n = 2$	44M	**29.54**	**49.05**	34.32	33.88	14.05	**14.41**	22.18
PHM-Tm† $n = 4$	22M (-50.0%)	29.17	46.24	34.86	33.80	14.43	13.78	21.91
PHM-Tm† $n = 8$	11M (-75.0%)	29.47	43.49	34.71	32.59	13.75	13.78	21.43

Table 3: Training time (seconds per 100 steps) and inference time (seconds to decode test sets) with beam size of 4 and length penalty of 0.6 on the IWSLT'14 German-English dataset.

Model	Transformer (Tm)	Quaternion Tm	PHM-Tm ($n = 4$)	PHM-Tm ($n = 8$)
Training time	**7.61**	8.11	7.92	7.70
Inference time	336	293	299	**282**

Table 4: Experimental results of text style transfer. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	BLEU
Transformer (Tm)	44M	11.65
PHM-Tm ($n = 2$)	22M (-50.0%)	12.20
PHM-Tm ($n = 4$)	11M (-75.0%)	**12.42**
PHM-Tm ($n = 8$)	5.5M (-87.5%)	11.66
PHM-Tm ($n = 16$)	2.9M (-93.4%)	10.76

Table 5: Experimental results of subject verb agreement. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	Acc
Transformer (Tm)	400K	94.80
Quaternion Tm	100K	94.70
PHM-Tm ($n = 2$)	200K (-50.0%)	95.14
PHM-Tm ($n = 4$)	101K (-74.8%)	95.05
PHM-Tm ($n = 8$)	56K (-86.0%)	**95.62**

Эксперименты

Table 1: Experimental results of natural language inference (accuracy) on five different datasets. The PHM-LSTM reduces the parameters of the standard LSTM model and improves or partially matches performance on four out of five datasets.

Model	#Params	MNLI	QNLI	SNLI	DNLI	SciTail
LSTM	721K	**71.82** / 71.89	84.44	84.18	85.16	74.36
Quaternion LSTM	180K (-75.0%)	71.57 / **72.19**	**84.73**	84.21	86.45	75.58
PHM-LSTM ($n = 2$)	361K (-49.9%)	**71.82** / 72.08	84.39	84.38	85.77	77.47
PHM-LSTM ($n = 5$)	146K (-79.7%)	71.80 / 71.77	83.87	**84.58**	**86.47**	74.64
PHM-LSTM ($n = 10$)	81K (-88.7%)	71.59 / 71.59	84.25	84.40	86.21	**77.84**

Table 2: Experimental results of machine translation (BLEU) on seven different datasets. Symbol † represents re-scaling the parameters with a factor of 2 by doubling the hidden size. The PHM-Transformer does not lose much performance despite enjoying parameter savings. Re-scaling can lead to improvement in performance.

Model	#Params	En-Vi	En-Id	De-En	Ro-En	En-Et	En-Mk	En-Ro
Transformer (Tm)	44M	28.43	47.40	**36.68**	**34.60**	14.17	13.96	**22.79**
Quaternion Tm	11M (-75.0%)	28.00	42.22	32.83	30.53	13.10	13.67	18.50
PHM-Tm $n = 2$	22M (-50.0%)	29.25	46.32	35.52	33.40	**14.98**	13.60	21.73
PHM-Tm $n = 4$	11M (-75.0%)	29.13	44.13	35.53	32.74	14.11	13.01	21.19
PHM-Tm $n = 8$	5.5M (-87.5%)	29.34	40.81	34.16	31.88	13.08	12.95	21.66
PHM-Tm $n = 16$	2.9M (-93.4%)	29.04	33.48	33.89	31.53	12.15	11.97	19.63
PHM-Tm† $n = 2$	44M	**29.54**	**49.05**	34.32	33.88	14.05	**14.41**	22.18
PHM-Tm† $n = 4$	22M (-50.0%)	29.17	46.24	34.86	33.80	14.43	13.78	21.91
PHM-Tm† $n = 8$	11M (-75.0%)	29.47	43.49	34.71	32.59	13.75	13.78	21.43

Table 3: Training time (seconds per 100 steps) and inference time (seconds to decode test sets) with beam size of 4 and length penalty of 0.6 on the IWSLT'14 German-English dataset.

Model	Transformer (Tm)	Quaternion Tm	PHM-Tm ($n = 4$)	PHM-Tm ($n = 8$)
Training time	**7.61**	8.11	7.92	7.70
Inference time	336	293	299	**282**

Table 4: Experimental results of text style transfer. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	BLEU
Transformer (Tm)	44M	11.65
PHM-Tm ($n = 2$)	22M (-50.0%)	12.20
PHM-Tm ($n = 4$)	11M (-75.0%)	**12.42**
PHM-Tm ($n = 8$)	5.5M (-87.5%)	11.66
PHM-Tm ($n = 16$)	2.9M (-93.4%)	10.76

Table 5: Experimental results of subject verb agreement. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	Acc
Transformer (Tm)	400K	94.80
Quaternion Tm	100K	94.70
PHM-Tm ($n = 2$)	200K (-50.0%)	95.14
PHM-Tm ($n = 4$)	101K (-74.8%)	95.05
PHM-Tm ($n = 8$)	56K (-86.0%)	**95.62**

Эксперименты

Table 1: Experimental results of natural language inference (accuracy) on five different datasets. The PHM-LSTM reduces the parameters of the standard LSTM model and improves or partially matches performance on four out of five datasets.

Model	#Params	MNLI	QNLI	SNLI	DNLI	SciTail
LSTM	721K	**71.82** / 71.89	84.44	84.18	85.16	74.36
Quaternion LSTM	180K (-75.0%)	71.57 / **72.19**	**84.73**	84.21	86.45	75.58
PHM-LSTM ($n = 2$)	361K (-49.9%)	**71.82** / 72.08	84.39	84.38	85.77	77.47
PHM-LSTM ($n = 5$)	146K (-79.7%)	71.80 / 71.77	83.87	**84.58**	**86.47**	74.64
PHM-LSTM ($n = 10$)	81K (-88.7%)	71.59 / 71.59	84.25	84.40	86.21	**77.84**

Table 2: Experimental results of machine translation (BLEU) on seven different datasets. Symbol † represents re-scaling the parameters with a factor of 2 by doubling the hidden size. The PHM-Transformer does not lose much performance despite enjoying parameter savings. Re-scaling can lead to improvement in performance.

Model	#Params	En-Vi	En-Id	De-En	Ro-En	En-Et	En-Mk	En-Ro
Transformer (Tm)	44M	28.43	47.40	**36.68**	**34.60**	14.17	13.96	**22.79**
Quaternion Tm	11M (-75.0%)	28.00	42.22	32.83	30.53	13.10	13.67	18.50
PHM-Tm $n = 2$	22M (-50.0%)	29.25	46.32	35.52	33.40	**14.98**	13.60	21.73
PHM-Tm $n = 4$	11M (-75.0%)	29.13	44.13	35.53	32.74	14.11	13.01	21.19
PHM-Tm $n = 8$	5.5M (-87.5%)	29.34	40.81	34.16	31.88	13.08	12.95	21.66
PHM-Tm $n = 16$	2.9M (-93.4%)	29.04	33.48	33.89	31.53	12.15	11.97	19.63
PHM-Tm† $n = 2$	44M	**29.54**	**49.05**	34.32	33.88	14.05	**14.41**	22.18
PHM-Tm† $n = 4$	22M (-50.0%)	29.17	46.24	34.86	33.80	14.43	13.78	21.91
PHM-Tm† $n = 8$	11M (-75.0%)	29.47	43.49	34.71	32.59	13.75	13.78	21.43

Table 3: Training time (seconds per 100 steps) and inference time (seconds to decode test sets) with beam size of 4 and length penalty of 0.6 on the IWSLT'14 German-English dataset.

Model	Transformer (Tm)	Quaternion Tm	PHM-Tm ($n = 4$)	PHM-Tm ($n = 8$)
Training time	**7.61**	8.11	7.92	7.70
Inference time	336	293	299	**282**

Table 4: Experimental results of text style transfer. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	BLEU
Transformer (Tm)	44M	11.65
PHM-Tm ($n = 2$)	22M (-50.0%)	12.20
PHM-Tm ($n = 4$)	11M (-75.0%)	**12.42**
PHM-Tm ($n = 8$)	5.5M (-87.5%)	11.66
PHM-Tm ($n = 16$)	2.9M (-93.4%)	10.76

Table 5: Experimental results of subject verb agreement. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	Acc
Transformer (Tm)	400K	94.80
Quaternion Tm	100K	94.70
PHM-Tm ($n = 2$)	200K (-50.0%)	95.14
PHM-Tm ($n = 4$)	101K (-74.8%)	95.05
PHM-Tm ($n = 8$)	56K (-86.0%)	**95.62**

Эксперименты

Table 1: Experimental results of natural language inference (accuracy) on five different datasets. The PHM-LSTM reduces the parameters of the standard LSTM model and improves or partially matches performance on four out of five datasets.

Model	#Params	MNLI	QNLI	SNLI	DNLI	SciTail
LSTM	721K	**71.82** / 71.89	84.44	84.18	85.16	74.36
Quaternion LSTM	180K (-75.0%)	71.57 / **72.19**	**84.73**	84.21	86.45	75.58
PHM-LSTM ($n = 2$)	361K (-49.9%)	**71.82** / 72.08	84.39	84.38	85.77	77.47
PHM-LSTM ($n = 5$)	146K (-79.7%)	71.80 / 71.77	83.87	**84.58**	**86.47**	74.64
PHM-LSTM ($n = 10$)	81K (-88.7%)	71.59 / 71.59	84.25	84.40	86.21	**77.84**

Table 2: Experimental results of machine translation (BLEU) on seven different datasets. Symbol † represents re-scaling the parameters with a factor of 2 by doubling the hidden size. The PHM-Transformer does not lose much performance despite enjoying parameter savings. Re-scaling can lead to improvement in performance.

Model	#Params	En-Vi	En-Id	De-En	Ro-En	En-Et	En-Mk	En-Ro
Transformer (Tm)	44M	28.43	47.40	**36.68**	**34.60**	14.17	13.96	**22.79**
Quaternion Tm	11M (-75.0%)	28.00	42.22	32.83	30.53	13.10	13.67	18.50
PHM-Tm $n = 2$	22M (-50.0%)	29.25	46.32	35.52	33.40	**14.98**	13.60	21.73
PHM-Tm $n = 4$	11M (-75.0%)	29.13	44.13	35.53	32.74	14.11	13.01	21.19
PHM-Tm $n = 8$	5.5M (-87.5%)	29.34	40.81	34.16	31.88	13.08	12.95	21.66
PHM-Tm $n = 16$	2.9M (-93.4%)	29.04	33.48	33.89	31.53	12.15	11.97	19.63
PHM-Tm† $n = 2$	44M	**29.54**	**49.05**	34.32	33.88	14.05	**14.41**	22.18
PHM-Tm† $n = 4$	22M (-50.0%)	29.17	46.24	34.86	33.80	14.43	13.78	21.91
PHM-Tm† $n = 8$	11M (-75.0%)	29.47	43.49	34.71	32.59	13.75	13.78	21.43

Table 3: Training time (seconds per 100 steps) and inference time (seconds to decode test sets) with beam size of 4 and length penalty of 0.6 on the IWSLT'14 German-English dataset.

Model	Transformer (Tm)	Quaternion Tm	PHM-Tm ($n = 4$)	PHM-Tm ($n = 8$)
Training time	**7.61**	8.11	7.92	7.70
Inference time	336	293	299	**282**

Table 4: Experimental results of text style transfer. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	BLEU
Transformer (Tm)	44M	11.65
PHM-Tm ($n = 2$)	22M (-50.0%)	12.20
PHM-Tm ($n = 4$)	11M (-75.0%)	**12.42**
PHM-Tm ($n = 8$)	5.5M (-87.5%)	11.66
PHM-Tm ($n = 16$)	2.9M (-93.4%)	10.76

Table 5: Experimental results of subject verb agreement. The PHM-Transformer may reduce the parameters of the standard Transformer model and improve performance.

Model	#Params	Acc
Transformer (Tm)	400K	94.80
Quaternion Tm	100K	94.70
PHM-Tm ($n = 2$)	200K (-50.0%)	95.14
PHM-Tm ($n = 4$)	101K (-74.8%)	95.05
PHM-Tm ($n = 8$)	56K (-86.0%)	**95.62**

Результаты

- Новый метод параметризации произведения в многомерных пространствах
- Применимость в LSTM и Трансформерах
- Эмпирическая гибкость (можно использовать разные n)

Рецензент

Краткое описание статьи: в статье предлагается способ уменьшения количества параметров линейного слоя в n раз (без значительного уменьшения качества и увеличения времени работы). Он обобщает метод сжатия в 4 раза с помощью кватернионного умножения. Авторы показали применимость метода для сжатия рекуррентных сетей и трансформера.

Сильные стороны

- Придумано интересное обобщение метода, использующего кватернионы. При n=4 представленный метод с теоретической точки зрения не хуже (то есть является обобщением).
- Показано, что метод сжатия дает небольшое уменьшение качества при n кратном уменьшении количества параметров (на NLP задачах).
- С помощью метода можно настраивать во сколько раз уменьшать количество параметров.
- С помощью метода можно увеличить матрицы внутри трансформера, а затем уменьшить количество параметров методом в n раз (тем самым получить модель с тем же количеством параметров), и это может дать улучшение качества относительно изначального транформера.
- Статью легко читать, все математические выкладки очень хорошо поданы. Идеи были теоретически обоснованы (например переход от метода с кватернионами к именно такому методу).
- С научной точки зрения представлен иной (не похожий на предыдущие) метод уменьшения количества параметров нейронной сети.
- Параметры экспериментов достаточно подробно описаны и код выложен в github.

Слабые стороны

- В экспериментах нет сравнения с очевидным и стандартным методом сжатия с помощью SVD (матрица k x d → k x r и r x d).
- Возможно можно ускорить умножение на PHM матрицу в n раз, потому что она имеет специальный вид. В целом возможно можно записать PHM матрицу в тензорном виде через параметры S, A и это может дать какое-то иное представление метода.
- В методе параметры матрицы получаются перемножениями обучаемых параметров, что на практике может быть нестабильно. Не хватило комментариев по этому поводу.
- Эксперименты вызывают вопросы, в целом было много экспериментов с разными NLP задачами, но многие из них кажутся довольно нестандартными; также большой разброс в результатах разных экспериментов вызывает тревогу относительно возможной случайности результатов и их незначимости.

Итог

- **Насколько хорошо написана статья:** статья написана доходчиво, сложностей в понимании не возникло.
- **Воспроизводимость:** статья написана достаточно подробно, также авторы выложили код экспериментов. При самореализации проблем возникнуть не должно, но есть опасения насчет того, что метод успешно обучится.
- **Дополнительные комментарии, предложения по улучшению:** хотелось бы добавить другие методы сжатия в эксперименты, а также доверительные интервалы для всех результатов.

- **Оценка:** 7
- **Уверенность:** 4