FullyQT: Quantized Transformer for Improved Translation

Gabriele Prato
Mila, Université de Montréal
pratogab@mila.quebec

Ella Charlaix
Huawei Noah's Ark Lab
ella.charlaix@huawei.com

Mehdi Rezagholizadeh
Huawei Noah's Ark Lab
mehdi.rezagholizadeh@huawei.com

Overview

Problem

State-of-the-art machine translation methods employ massive amounts of parameters. Compressing such models is essential for efficient inference on edge-devices.

Proposition

We propose a quantization strategy for the Transformer. Our goal is to:

- Quantize all operations which can provide a computational speed gain.
- Exploit hardware resources as efficiently as possible.
- Maintain accuracy with respect to full-precision.

FullyQT

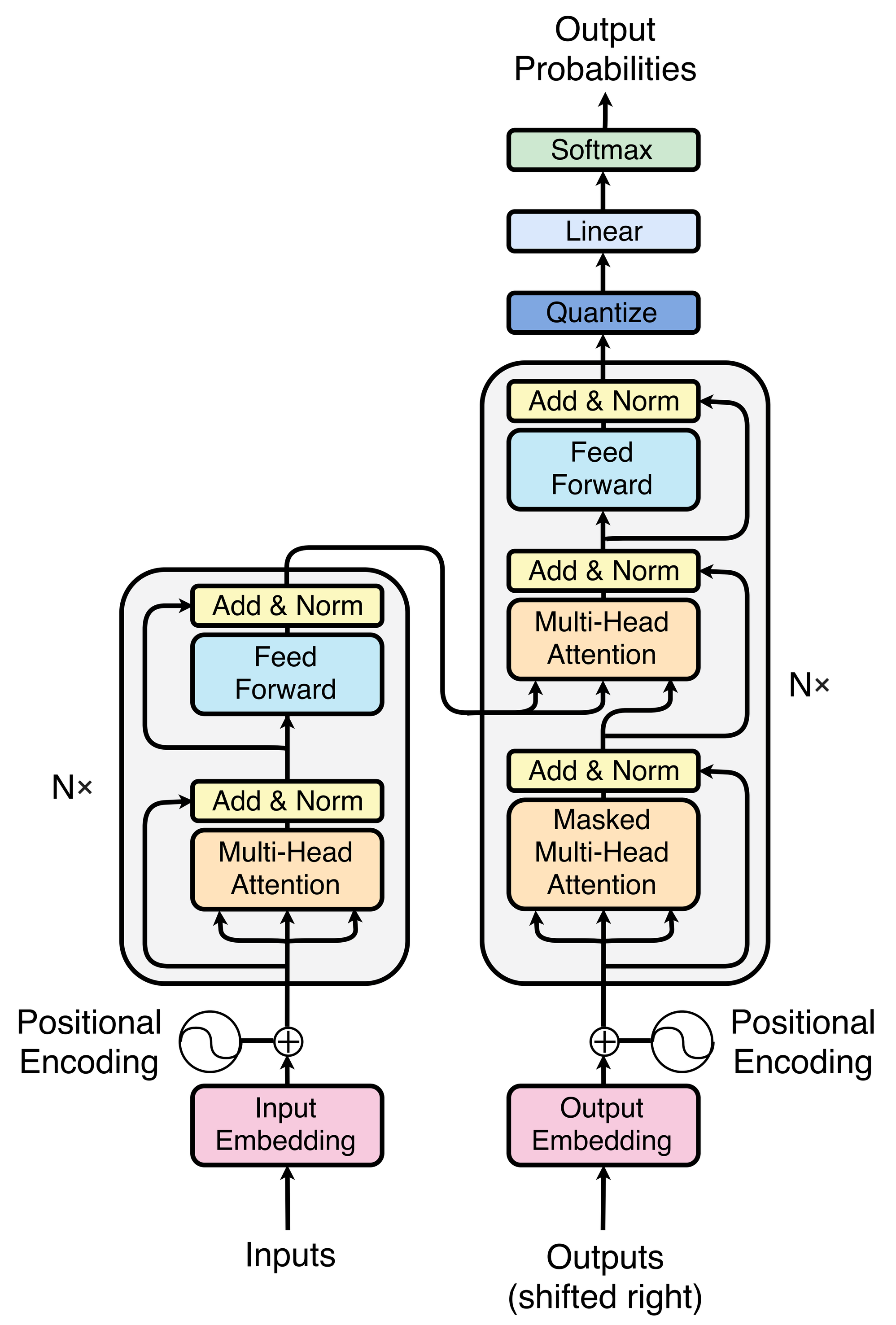

Figure 1: Fully Quantized Transformer

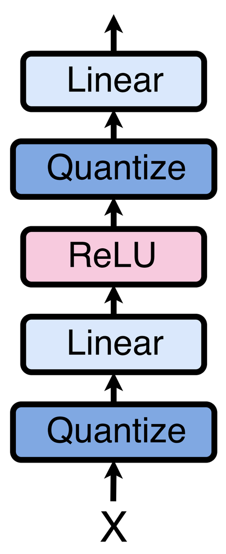

Figure 2: Feed-forward

$$\frac{\mathcal{Q}\left(e^{x}\right)}{\mathcal{Q}\left(\sum e^{x}\right)}$$

Figure 3: Softmax

$$\mathcal{Q}\left(\frac{\mathcal{Q}\left(x-\mu\right)}{\mathcal{Q}\left(\sqrt{\sigma^2+\epsilon}\right)}\right)*\gamma+\beta$$

Figure 4: LayerNorm

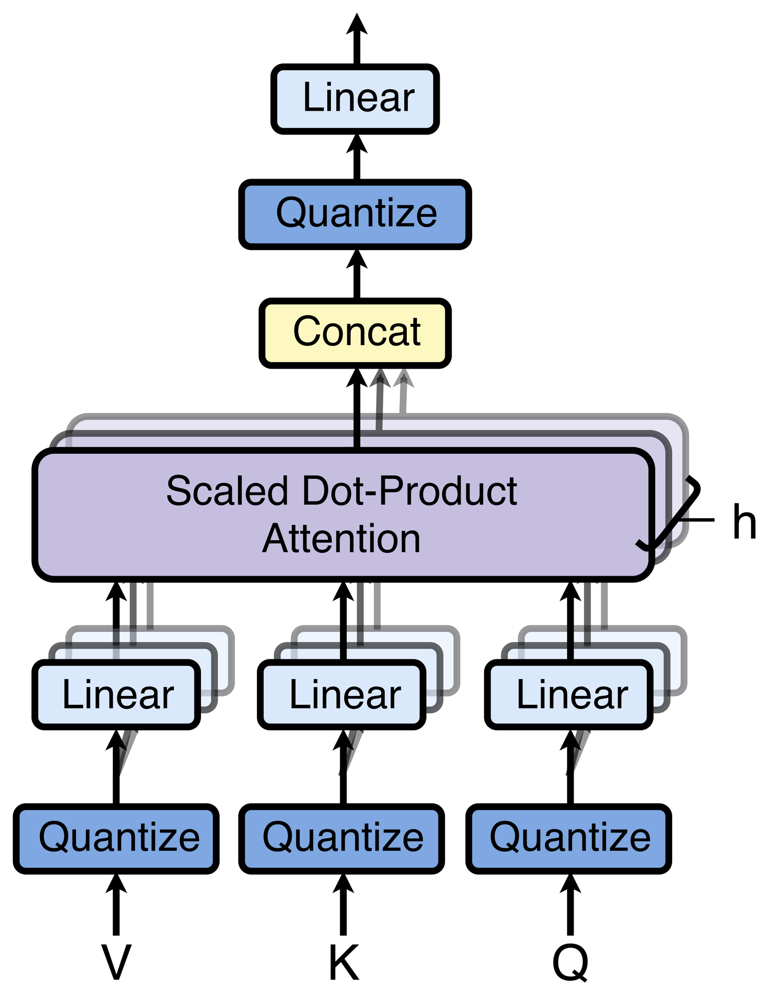

Figure 5: Multi-Head Attention

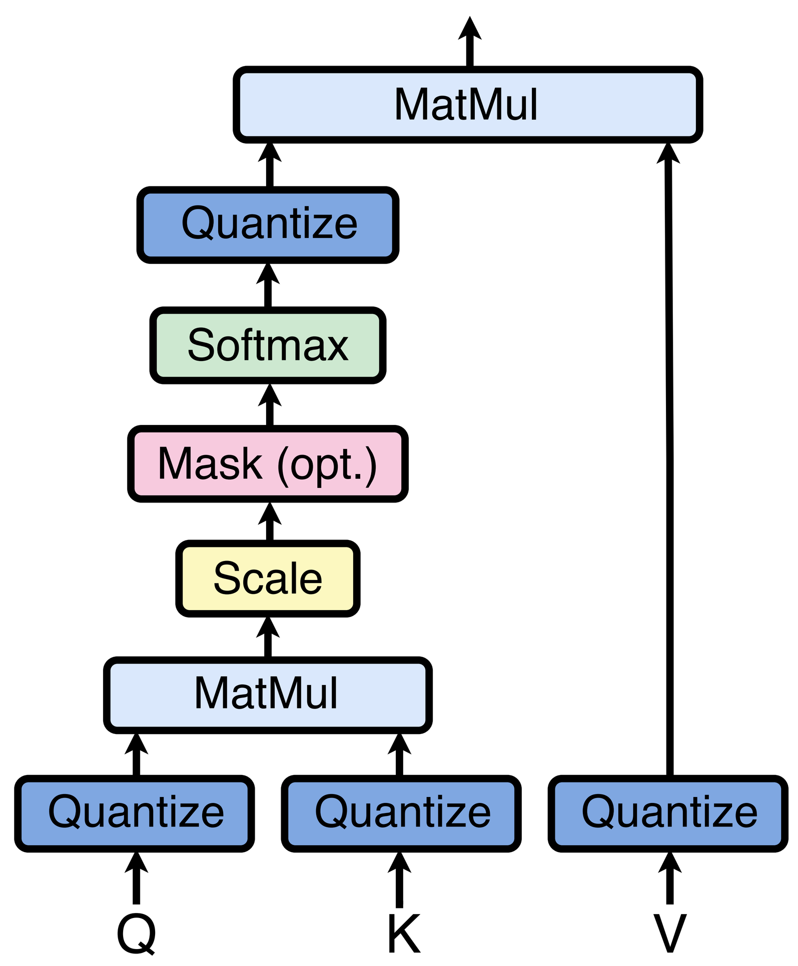

Figure 6: Scale Dot-Product Attention

Quantization [4]

Given an element x of a tensor $\mathbf{X}$, we apply the quantization function $\mathcal{Q}$:

$$\mathcal{Q}(x) = \left\lfloor \frac{\text{clamp}(x; x_{min}, x_{max}) - x_{min}}{s} \right\rceil * s + x_{min} \tag{1}$$

$$s = \frac{x_{max} - x_{min}}{2^k - 1} \tag{2}$$

For weights:

$$x_{min} = \min(\mathbf{X})$$
$$x_{max} = \max(\mathbf{X})$$

For activations:

$$x_{min}, x_{max} = \text{Running Estimates}$$

Results

Method	Fully Quantized	Size (Gb) [EN-DE, EN-FR]	Compr.	EN-DE (2014)	BLEU EN-FR	EN-DE (2017)
Vaswani et al. (2017)		[2.02, 1.94]	1x	27.3	38.1	-
Cheong & Daniel (2019)		0.69	2.92x	-	-	27.38
Bhandare et al. (2019)		$\geq$ 0.96	$\leq$ 2.1x	27.33	-	-
Fan (2019)		$\geq$ 0.51	$\leq$ 3.99x	26.94	-	-
FullyQT	✓	[0.52, 0.50]	3.91x	**27.60**	**39.91**	**27.60**

Table 1: Original Transformer vs quantized variants.

Model	Method	Precision	EN-DE PPL	BLEU	Size (Gb)	Compr.	EN-FR PPL	BLEU	Size (Gb)	Compr.
Base	Baseline	32-bit	4.95	26.46	2.02	1x	3.21	38.34	1.94	1x
	Default Approach	8-bit	74.04	0.21	0.52	3.91x	*nan*	0	0.50	3.91x
	Post-Quantization	8-bit	4.97	26.44	0.52	3.91x	3.26	38.30	0.50	3.91x
	FullyQT	8-bit	4.94	26.38	0.52	3.91x	3.23	**38.41**	0.50	3.91x
	Post-Quantization	6-bit	6.00	24.84	0.39	5.18x	3.98	35.02	0.37	5.17x
	FullyQT	6-bit	5.09	**26.98**	0.39	5.18x	3.38	37.07	0.37	5.17x
	FullyQT	4-bit	11.96	18.32	0.26	7.66x	48.21	1.59	0.25	7.64x
Big	Baseline	32-bit	4.03	26.85	6.85	1x	2.72	40.17	6.69	1x
	Post-Quantization	8-bit	4.27	26.55	1.74	3.95x	2.78	39.78	1.69	3.95x
	FullyQT	8-bit	4.24	**27.95**	1.74	3.95x	2.80	**40.17**	1.69	3.95x
	Post-Quantization	6-bit	5.12	24.86	1.31	5.24x	3.08	37.92	1.28	5.24x
	FullyQT	6-bit	4.78	26.76	1.31	5.24x	2.87	39.59	1.28	5.24x
	FullyQT	4-bit	33.11	10.22	0.88	7.79x	42.42	2.81	0.86	7.79x

Table 2: Evaluation of different bitwidth for quantization.

Method	EN-FR PPL	BLEU
No Bucketing	3.49	37.14
No Gradient Clipping	2549.30	0
No LayerNorm Denominator Quantization	3.22	38.29
8-bit Quantized Weights, Full-precision Activations	3.20	38.36

Table 3: Quantization scheme variations.

Precision	Size (Mb)	Compression	WikiText-2 Loss	PPL	WikiText-103 Loss	PPL
32-bit	243.04	1x	5.65	284.15	5.91	**369.20**
8-bit	61.93	3.92x	5.64	282.67	5.94	377.79
6-bit	46.75	5.20x	5.64	**281.48**	5.93	376.44
4-bit	31.57	7.70x	5.65	284.26	5.94	378.67

Table 4: Language Modeling task.

Pruning

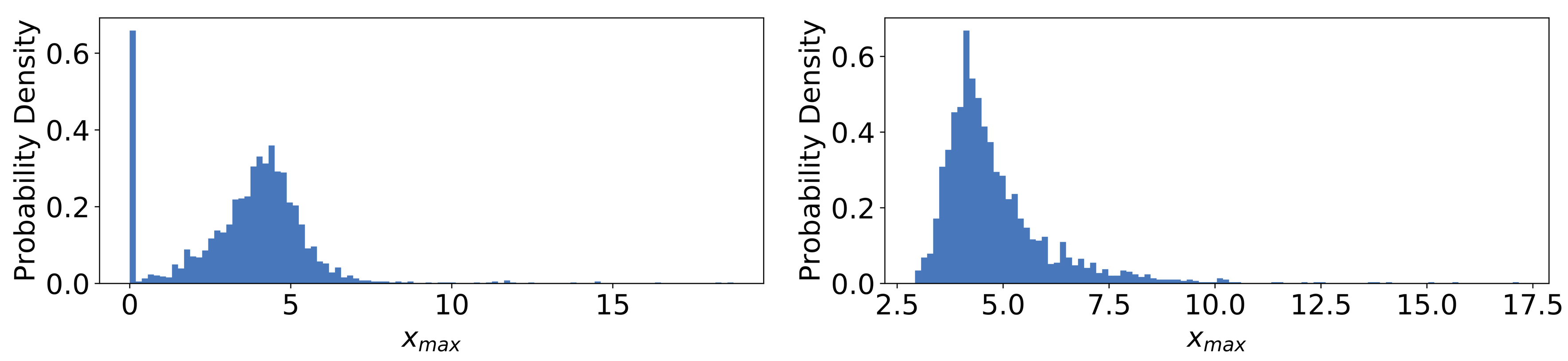

Figure 7: x_{max} of a ReLU layer, one x_{max} per output node. (Encoder left, Decoder right)

Model	Method	EN-DE PPL	BLEU	Nodes Pruned in Encoder FF	Total Compr.	EN-FR PPL	BLEU	Nodes Pruned in Encoder FF	Total Compr.
Base	No pruning	4.39	27.60	0%	3.95x	2.90	39.91	0%	3.95x
	L1-norm fixed	5.57	23.99	13.57%	4.02x	4.38	29.01	9.47%	3.99x
	x_{max} fixed	4.57	27.33	13.57%	4.02x	3.18	39.40	9.47%	3.99x
	x_{max} adaptive	4.40	**27.60**	13.57%	4.02x	2.90	**39.91**	9.47%	3.99x
Big	No pruning	4.24	27.95	0%	3.97x	2.80	40.17	0%	3.97x
	L1-norm fixed	5.80	22.65	26.39%	4.21x	4.16	28.85	28.41%	4.24x
	x_{max} fixed	4.47	27.43	26.39%	4.21x	2.91	39.40	28.41%	4.24x
	x_{max} adaptive	4.25	**27.95**	26.39%	4.21x	2.80	**40.17**	28.41%	4.24x

Table 5: Adaptive vs fixed rate pruning, equal proportions.

References

[1] A. Bhandare, V. Sripathi, D. Karkada, V. Menon, S. Choi, K. Datta, and V. Saletore. Efficient 8-Bit Quantization of Transformer Neural Machine Language Translation Model. *arXiv e-prints*, page arXiv:1906.00532, Jun 2019.

[2] R. Cheong and R. Daniel. transformers.zip: Compressing Transformers with Pruning and Quantization. Technical report, Stanford University, Stanford, California, 2019.

[3] C. Fan. Quantized Transformer. Technical report, Stanford University, Stanford, California, 2019.

[4] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *arXiv e-prints*, page arXiv:1712.05877, Dec 2017.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, Jun 2017.