WINA: WEIGHT INFORMED NEURON ACTIVATION FOR ACCELERATING LARGE LANGUAGE MODEL INFERENCE

Sihan Chen[2*], **Dan Zhao**[1], **Jongwoo Ko**[1], **Colby Banbury**[1], **Huiping Zhuang**[3], **Luming Liang**[1], **Pashmina Cameron**[1], **Tianyi Chen**[1†*]
[1]Microsoft, [2]Renmin University of China, [3]South China University of Technology
chensihan@ruc.edu.cn, Tianyi.Chen@microsoft.com

Motivation

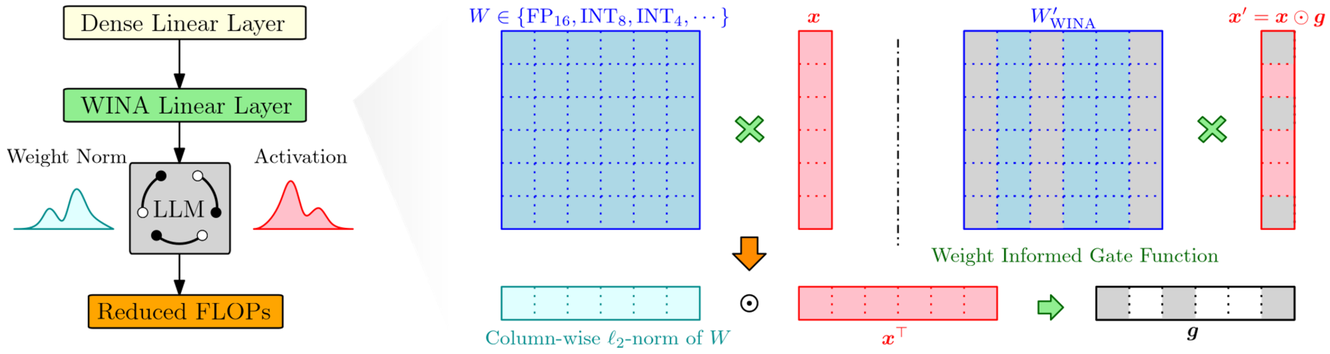
- Training-free sparse activation enables inference speed-up by selectively omitting neurons at runtime.
- Current sparse activation methods rely on a top-K gating mechanism based on **hidden state magnitudes**, overlooking the inherent contribution of the **weight** matrices.

Method

For input x and weight matrix W, select top-k neurons using the importance score

$$|x_i| \cdot \| W_i \|_2,$$

where $\| W_i \|_2$ is the ℓ_2-norm of i-th column of W.



Contributions

- Weighted-Informed Activation.
- Theoretically Tighter Approximation Error.
- Numerical Experiments.

	WINA	TEAL	CATS	R-Sparse
Tight Approx Error	✓	✗	✗	✗
Layer-Agnostic Application†	✓	✓	✗	✓
Layer-Specific Sparsity	✓	✓	✗	✓

†: Some methods (i.e., CATS) are only adopted on specific types of layers.

Theory

- **Lemma**: Optimal approximation error over single linear layer
- **Theorem**: WINA minimizes a provable upper bound on output deviation
- **Remarks**. Our analysis requires column-wise orthogonal weights. We achieve this via a functionally lossless and lightweight transformation.

Results

Table 4: Results over Llama-3-8B on commonsense reasoning.

Sparsity	Method	PiQA	Arc-C	WinoGrande	HellaSwag	SciQ	OBQA	BoolQ	Arc-E	Avg
0%	Baseline (full model)	80.79	53.33	72.61	79.17	93.90	45.00	81.38	77.74	72.99
25%	CATS†	78.62	48.04	70.64	76.32	91.90	41.80	78.13	71.09	69.57
	R-Sparse	79.82	52.05	72.38	78.69	93.50	44.40	80.92	78.75	72.56
	TEAL	80.20	53.16	73.32	78.85	94.10	45.20	80.83	76.89	72.82
	WINA	80.41	52.82	73.80	78.99	94.00	44.60	82.05	78.03	**73.09**
40%	CATS†	59.96	27.82	51.30	40.18	46.10	29.80	42.26	38.09	41.94
	R-Sparse	79.05	50.26	72.14	76.91	94.10	43.00	79.14	77.86	71.56
	TEAL	79.00	48.98	71.82	77.45	93.30	45.00	80.03	77.19	71.60
	WINA	79.87	50.68	72.30	77.91	93.90	45.00	82.23	77.57	**72.43**
50%	R-Sparse	76.22	45.73	66.61	73.22	93.80	42.20	76.70	74.83	68.66
	TEAL	78.29	48.12	70.09	74.83	93.70	42.60	78.23	74.41	70.03
	WINA	79.16	48.81	70.64	76.44	93.50	43.60	81.25	75.00	**71.05**
65%	R-Sparse	68.50	33.36	57.38	51.48	86.00	31.80	65.23	58.80	56.57
	TEAL	73.34	37.37	63.46	61.76	88.90	37.00	69.85	64.48	62.02
	WINA	74.65	41.98	64.48	67.89	90.70	41.60	76.73	67.00	**65.63**

† CATS is unable to reach 50% or 65% sparsity since it only achieves sparse activations over MLP layers.

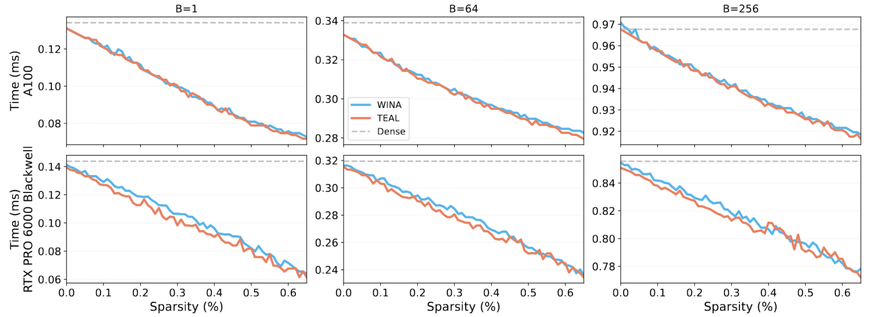


Figure 4: Sparsity vs. latency across different batch sizes $B \in \{1, 64, 256\}$ for GEMV (generalized matrix-vector multiplication) of sizes 5120×1 and 5120×17920. WINA's Triton kernel performance consistently matches that of TEAL across different GPU architectures (A100 top, RTX PRO 6000 Blackwell bottom), achieving similar speedups across our sparsity levels and as sparsity increases.

Code: https://github.com/microsoft/wina