Lipschitz Normalization for Self-Attention Layers with Application to Graph Neural Networks

George Dasoulas Kevin Scaman Aladin Virmaux

Attention

- A ’ωÎ¯ק’ā̄ӯā̇!¯ℓ̇ωђ method.
- ĕℏ꙰¯ℓ̇ℏӯ focus of structured input.

Attention mechanism

Let ŧ P R$^{ă \; ђ}$ be an input matrix and a **score function** Ô : R$^{ă \; ђ}$ Ñ R$^{g \; ђ}$:

8§§pŧq ḑpŧq n⊤§h9²pÔpŧqq^{J} ;

Standard attention

8§§pŧq ŧ n⊤§h9²pĔ^{J} ŧq^{J} ;

Transformer

8§§pŧq Ŝ n⊤§h9² $\frac{Ĕ^{J} É}{ă}$ ^{J} with

ŧ pĔ||É||Ŝq

Attention

- A ’ωÎ¯˥’ā̇ɏā!¯0̇ωɧ method.
- ĕhʬ¯0̇hɏ focus of structured input.

Attention mechanism

Let t P R$^{ǎ\ \ ɧ}$ be an input matrix and a **score function** $\hat{O}$: R$^{ǎ\ \ ɧ}$ Ñ R$^{g\ \ ɧ}$:

8 §§ptq $\quad$ dptq nT§ h 9 ²p$\hat{O}$p$t$qq^{J} ;

Standard attention

8 §§ptq $\quad$ t nT§ h 9 ²p$\check{E}^{J}$ tq^{J} ;

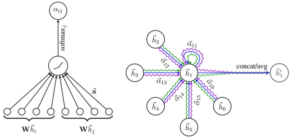
Transformer

8 §§ptq $\quad$ $\hat{S}$ nT§ h 9 ² $\frac{\check{E}^{J}_{?}\acute{E}}{\check{a}}$^{J} with

t $\quad$ p$\check{E}$||$\acute{E}$||$\hat{S}$q

Graph Learning and Attention

- **Neighbor-wise** softmax function.



Graph attention network (Veličković et al, 2018)

- A ’ωÎ ̄ı’ā̇ȳā! ̄0̇ωɧ method.
- ĕhẅ ̄0̇hy̵ focus of structured input.

Attention mechanism

Let t P R$^{ă \quad ɧ}$ be an input matrix and a **score function** $Ô$: R$^{ă \quad ɧ}$ Ñ R$^{g \quad ɧ}$:

$$8 §§p t q \quad ɖ p t q \; nΤ§ h 9^2 p Ô p t q q^J \; ;$$

Standard attention

$$8 §§p t q \quad t \; nΤ§ h 9^2 p Ĕ^J t q^J \; ;$$

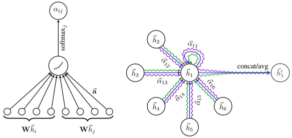
Transformer

$8 §§p t q \quad Ŝ \; nΤ§ h 9^2 \frac{Ĕ^J_? É}{ă}^J$ with

$t \quad p Ĕ || É || Ŝ q$

Graph Learning and Attention

- **Neighbor-wise** softmax function.



Graph attention network (Veličković et al, 2018)

GNN training and model depth

- **Deep** attention models suffer from convergence inability (Alon et al, 2020).
  - Oversmoothing, oversquashing.
  - Gradient explosion/vanishing not explored!
- Gradient flow and Lipschitz continuity.

Overall message and contributions

How to enforce Lipschitz continuity to attention layers?

Main contributions

1. We derive **general bounds** for the Lipschitz constant of attention layers.
2. We propose a **novel normalization** for attention layers that ensures Lipschitz continuity.
3. We apply this normalization to **graph attention networks**.

- Large scores tend to create large gradients!

Lemma

[illegible]

[illegible]

Is attention Lipschitz continuous ?

- Large scores tend to create large gradients!

Lemma

[illegible]

Impact of a scalar normalization:

Normalize the score function by a scalar function [illegible]

- Large scores tend to create large gradients!

Lemma

{ωẅ ƕɧ˓ ŧ P R^{ă} ɧ ¯ɖā ɧωẅg ωǐ ¯ɖā ăāẅ0̇˙ƕ¯0̇˙ā ωǐ ƕ¯¯āɧ¯0̇ωɧ 0̇’ ˷ ʢʢāẅ ɛω˷ ɧăāă Ƹ˓˩

~D 8 §§$_{ŧ}$~$_{\{}$ ¤ } nT§ h 9 ²pÔp ŧ qq}$_{\{}$?$\overline{2}$} ŧ ^{J} }$_{p\ 8\ ;2q}$~DÔ$_{ŧ}$~$_{\{;p2;\ 8\ q}$;

Impact of a scalar normalization:

Normalize the score function by a scalar function ! : R^{ă} ɧ Ñ R : Ôp ŧ q $\frac{\hat{\Theta} p ŧ q}{! p ŧ q}$

Theorem

Íā¯ ¥ 0ǐ ²ĩ ǐωẅ ƕɏɏ ŧ P R^{ă} ɧ ˝ ā ɖƕ˙ā

(1) }Θ̂p ŧ q} $_{8}$ ¤ ! p ŧ q

(2) } ŧ ^{J} }$_{p\ 8\ ;2q}$~DΘ̂$_{ŧ}$~$_{\{;p2;\ 8\ q}$ ¤ ! p ŧ q

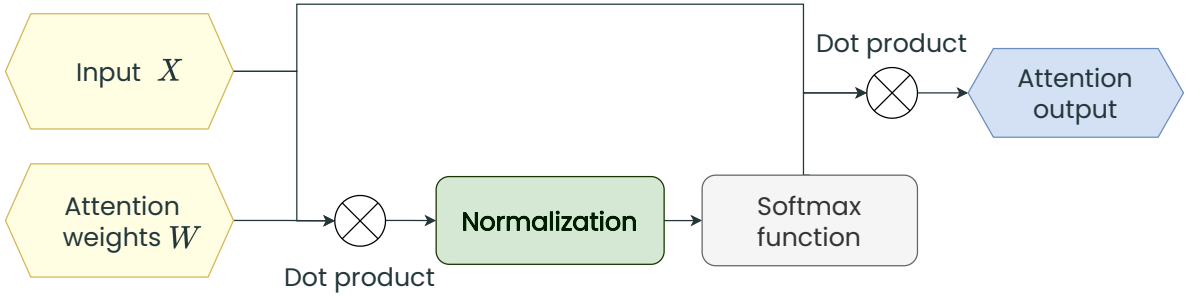
(3) } ŧ ^{J} }$_{p\ 8\ ;2q}$~D!$_{ŧ}$~$_{\{;1}$}Ôp ŧ q}$_{p2;\ 8\ q}$ ¤ ! p ŧ q^{2}

¯ɖāɧ ƕ¯¯āɧ¯0̇ωɧ gωăāɏ’ ˝ 0̇¯ɖ Ôp ŧ q Θ̂p ŧ q{! p ŧ q ƕẅā Í0̇ʢ’!ɖ0̇¯ !ωɧ¯0̇ɧ˷ω˷’ ƕɧă

Í$_{\{}$p 8 §§q ¤ ā c$\overline{\frac{g}{ɧ}}$?$\overline{8}$:

The Íi̇£’!q̇i̇¯_ Ýωẅg normalization

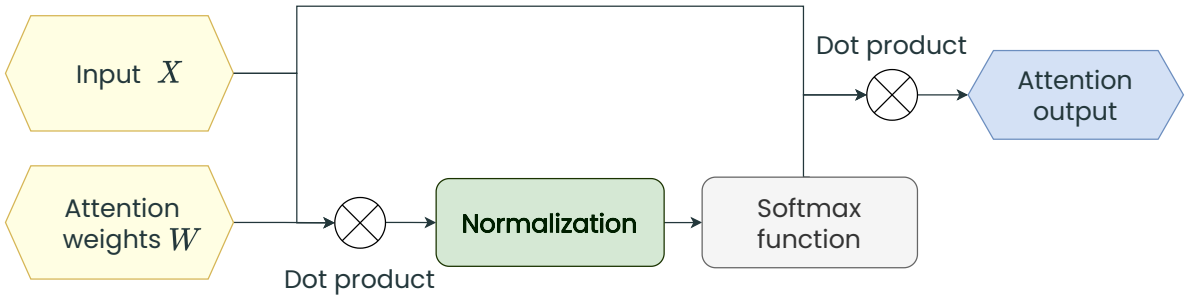
Core Idea: Replace original $\hat{O}pt{q}$ with normalized $\hat{O}pt{q}$ in the attention model!



Pipeline of LipschitzNorm.

The Íi£'!dj¯_ Ýωw g normalization

Core Idea: Replace original Ôptq with normalized Ôptq in the attention model!



Pipeline of LipschitzNorm.

Linear score function Ôptq

$$\hat{O}\mathrm{p}t\mathrm{q} \quad \frac{\check{E}^{\mathrm{J}}\, t}{\}\check{E}\}_{\{}\}t^{\mathrm{J}}\}_{\mathrm{p}\,8\,;2\mathrm{q}}}$$

$$\acute{I}_{\{}\mathrm{p}\,8\,\S\S\mathrm{q} \;\text{¤}\; \bar{a}^{1} \;\; \mathrm{b}\,\frac{\_\_}{\frac{g}{\hbar}} \;\; ?\,\overline{8.}$$

Quadratic score function Ôptq

$$\hat{O}\mathrm{p}t\mathrm{q} \quad \frac{\check{E}^{\mathrm{J}}\acute{E}}{\max \mathrm{t}\breve{}\,\dot{}\,;\breve{}\,''\,;\dot{}\,''\,\mathrm{u}}\,;$$

where $\breve{} \quad \}\check{E}\}_{\{};\dot{} \quad \}\acute{E}^{\mathrm{J}}\}_{\mathrm{p}\,8\,;2\mathrm{q}};''$ $\}\hat{S}^{\mathrm{J}}\}_{\mathrm{p}\,8\,;2\mathrm{q}}$

$$\acute{I}_{\{}\mathrm{p}\,8\,\S\S\mathrm{q} \;\text{¤}\; \bar{a}^{\,?\,\overline{3}} \;\; \mathrm{b}\,\frac{\_\_}{\frac{g}{\hbar}} \;\; 2^{\,?\,\overline{6.}}$$

Gradient Explosion

For **deep attention** models, there is a tight connection between: efficient training and Lipschitz continuity.

Lipschitz constant

Given $\grave{U}$ attention layers $8\S\S_g\,\mathrm{p}\,\mathrm{q}$, $\grave{\imath}\quad 8\S\S_1\quad 8\S\S_2\quad :::\quad 8\S\S_{g\ 1}\quad 8\S\S_g$ is Lipschitz continuous:

$$\acute{I}_{\{}\mathrm{p}\grave{\imath}\mathrm{q}\ \tilde{\mathrm{n}}\ \prod_{g\ 1}^{\grave{\mathbb{W}}}\acute{I}_{\{}\mathrm{p}8\S\S_g\,\mathrm{q};$$

- Multiplicative effect on the gradient flow.
- Enforcing Lipschitz continuity can alleviate gradient explosion.

Gradient Explosion

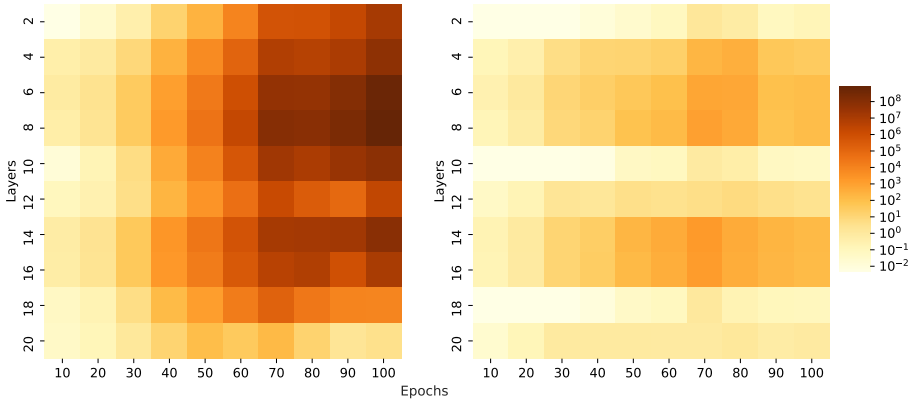
For **deep attention** models, there is a tight connection between: efficient training and Lipschitz continuity.

Lipschitz constant

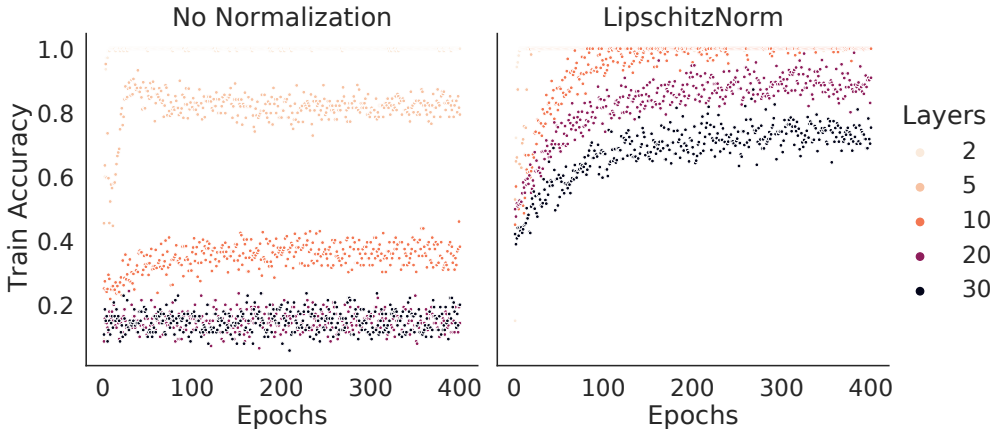
Given $\grave{U}$ attention layers $8\S\S_g\,p\,q$, $\grave{\imath}$ $8\S\S_1$ $8\S\S_2$ $:::$ $8\S\S_{g\ 1}$ $8\S\S_g$ is Lipschitz continuous:

$$\acute{I}_{\{}p\grave{\imath}q \ \text{¤} \prod_{g\ 1}^{\grave{U}} \acute{I}_{\{}p8\S\S_g q;$$

- **Multiplicative effect** on the gradient flow.
- Enforcing Lipschitz continuity can alleviate gradient explosion.



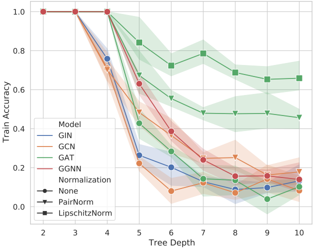
Gradient evolution of 25-layer GAT without/with LipschitzNorm



Model convergence w.r.t. model depth

A. Trees with increasing depth

- Synthetic graph benchmark (Alon et al, 2020).
- Simulation of ω˙ā'tc˜ħ'ḑŋÔ.

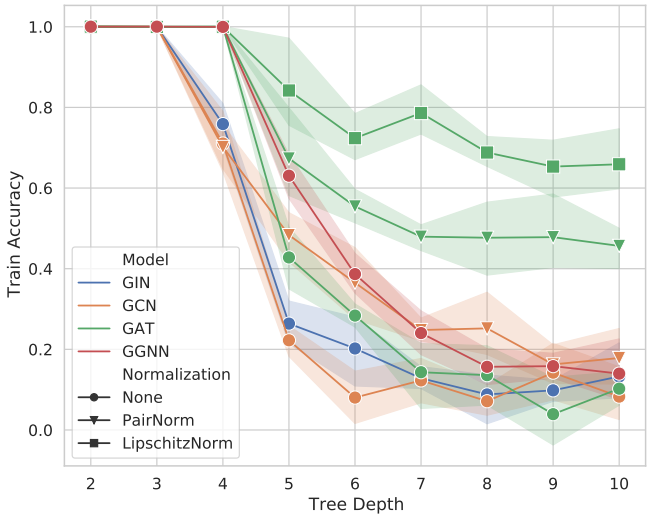


Train accuracies of four GNN models.

Synthetic Study

A. Trees with increasing depth

- Synthetic graph benchmark (Alon et al, 2020).
- Simulation of $\omega \cdot \bar{a}_{\mathbb{W}}^{\mathbb{W}} \cdot t\epsilon \cdot h\upsilon \cdot \dot{q}\ddot{\mathbb{0}}\hbar\hat{O}$.



Train accuracies of four GNN models.

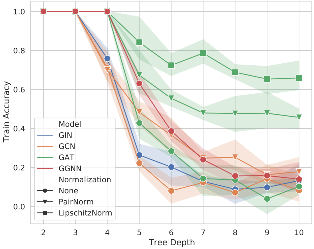
* Normalization always helps 😎

* LipschitzNorm for the win!

Synthetic Study

A. Trees with increasing depth

- Synthetic graph benchmark (Alon et al, 2020).
- Simulation of ω˙ ā$^{W}_{W}$’tɕ˜ ɯ’ɖ0̄ɧÔ.



Train accuracies of four GNN models.

* Normalization always helps 😎

* LipschitzNorm for the win!

B. Node Classification with Missing Features

(Zhao and Akoglou, 2019)

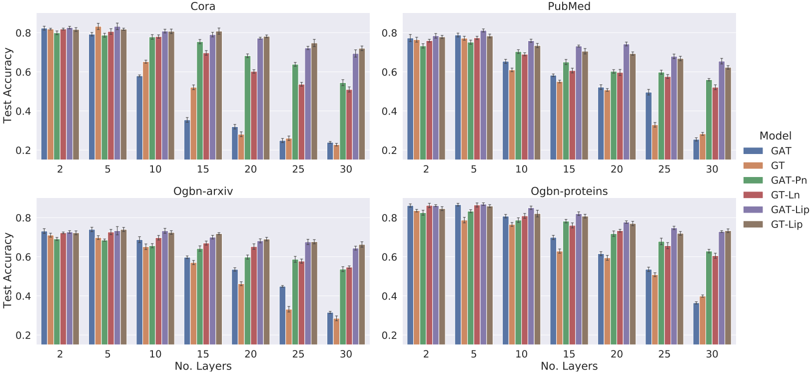
	Cora				CiteSeer				Pubmed			
	0%		100%		0%		100%		0%		100%	
GCN	82.5	1.2 (2)	58.8	3.5 (2)	**69.5**	**2.1 (2)**	31.3	2.7 (2)	77.9	1.4 (2)	44.9	4.4 (2)
GGNN	81.8	2.0 (2)	68.2	2.5 (6)	68.5	1.9 (3)	40.5	1.4 (5)	78.4	2.1 (4)	56.6	1.9 (4)
GAT	82.3	2.3 (2)	65.3	2.1 (4)	69.3	1.6 (2)	42.8	1.6 (4)	77.4	0.5 (6)	63.1	0.7 (4)
GAT-Pn	78.8	0.6 (4)	73.8	1.2 (12)	67.2	0.8 (4)	**51.7**	**1.1 (10)**	77.6	1.6 (8)	70.4	1.1 (12)
GAT-Lip	**83.1**	**0.5 (5)**	**75.3**	**0.9 (11)**	69.1	1.5 (3)	50.9	1.9 (9)	**78.9**	**1.3 (5)**	**73.3**	**1.4 (15)**

Accuracy of GAT for 100%-PubMed

* GAT-Lip: strong across shallow and deep settings!

Model depth in real-world datasets

- We evaluate the performance w.r.t. increasing number of layers.



Test accuracies of a GAT and a Graph Transformer (GT). By '-Lip' we denote the application of Í̊Ȿ'!ḓ̊_ Ýω̈g, by '-Ln' the Íŀ̀ ā̈Ýω̈g and by '-Pn' the ĕŀ̊̈Ýω̈g.

Thank you for your normalized attention!

References

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio,“Graph Attention Networks” ²GÍĜ, 2018.

L.Zhao & L.Akoglou,“PairNorm:Tackling Oversmoothing in GNNs” ²GÍĜ, 2019.

U. Alon & E. Yahav, “On the Bottleneck of Graph Neural Networks and its Practical Implications,” ²GÍĜ, 2021.

W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren,B. Liu, M. Catasta & J. Leskovec, “Open Graph Benchmark: Datasets for Machine Learning on Graphs,” GωĜĜ, 2020.