Enhancing Vision Transformer: Amplifying Non-Linearity in Feedforward Network Module

Yixing Xu, Chao Li, Dong Li, Xiao Sheng, Fan Jiang, Lu Tian, Ashish Sirasao, Emad Barsoum
Advanced Micro Devices, Inc.

AMD ICML International Conference On Machine Learning

Abstract

- Vision transformer contains two important components which are self-attention module and feedforward network (FFN) module.
- The majority of research tends to concentrate on modifying the former while **leaving the latter in its original form.**
- Through theoretical analysis, we demonstrate that the effect of the FFN module primarily lies in **providing non-linearity**, whose degree corresponds to the hidden dimensions.
- Thus, the computational cost of the FFN module can be reduced by **enhancing the degree of non-linearity** in the nonlinear function.
- We propose an **improved FFN** (IFFN) module for vision transformers which involves the usage of the **arbitrary GeLU** (AGeLU) and integrating multiple instances of it to augment non-linearity so that the hidden dimensions can be reduced.
- A **spatial enhancement part** is involved to further enrich the non-linearity in the proposed IFFN module.
- Experimental results show that we can **reduce FLOPs and parameters without compromising classification accuracy** on the ImageNet dataset irrespective of how the baseline models modify their self-attention part and the overall architecture.

FFN Module is a Non-linearity Generator

- Given an input matrix $\boldsymbol{X} \in \mathbb{R}^{N \times C}$ where N is the number of patches and C is the dimension of each patch, the output of FFN module is:

$$\boldsymbol{Y} = \mathrm{FFN}(\boldsymbol{X}) = \phi(\boldsymbol{X}\boldsymbol{W}^a)\boldsymbol{W}^b,$$

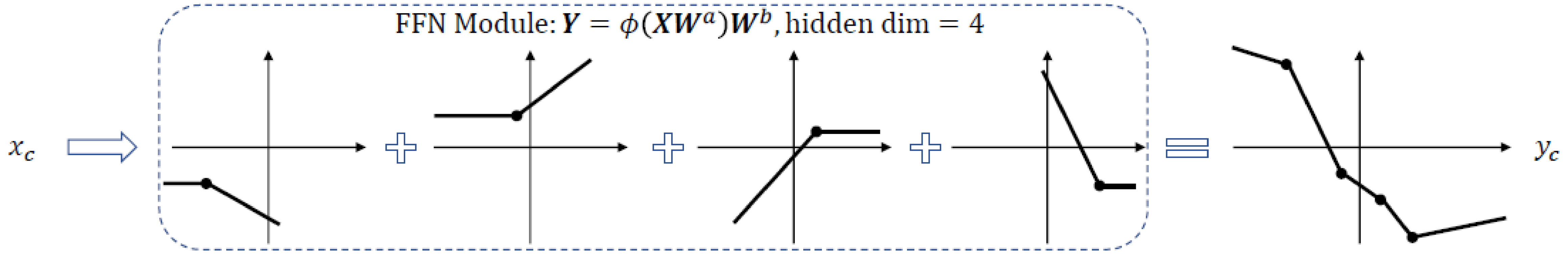
where $\boldsymbol{W}^a = \{w_{ij}^a\} \in \mathbb{R}^{C \times C'}$ and $\phi(\cdot)$ is the non-linear function.

- By representing the above equation in its element-wise form (assume $N = 1$ without loss of generality:

$$\boldsymbol{y} = \phi(\boldsymbol{x}\boldsymbol{W}^a)\boldsymbol{W}^b = \left(\sum_{j=1}^{C'} w_{jc}^b\, \phi(m_{cj}x_c + n_{cj})\right)_{c=1}^{C},$$

where $m_{cj} = w_{cj}^a$ and $n_{cj} = \sum_{i=1, i\neq c}^{C} w_{ij}^a x_i$, we can derive the following corollary:

1) Each element y_c in $\boldsymbol{y}$ is the linear combination of C' different nonlinear functions to the input element x_c.
2) Distinct scales and biases are applied to different input elements x_c before passing through the nonlinear function $\phi(\cdot)$.
3) The scale is a learnable weight independent to the input element x_c, while the bias is dependent to all other input elements in $\boldsymbol{x}$.



IFFN Module

A more Powerful Nonlinear Function

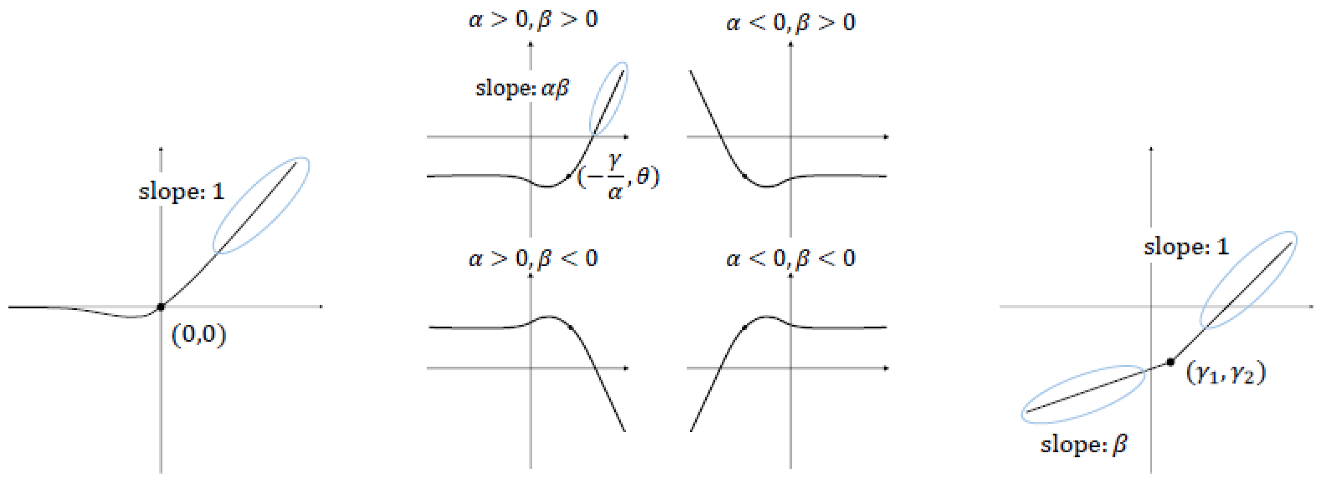
- Arbitrary nonlinear function is defined as:

$$\phi'(x) = \beta\phi(\alpha x + \gamma) + \theta$$

- Specifically, we introduce the arbitrary GeLU (AGeLU) to our model:

$$\mathrm{AGeLU}(x) = \beta \mathrm{GeLU}(\alpha x + \gamma) + \theta$$

- AGeLU is more flexible than other modified nonlinear functions such as RPReLU by having learnable slope and can switch the whole shape.



(a) GeLU (b) AGeLU (c) RPReLU

Channel-wise Enhancement Part

- We integrate two AGeLU functions and form a powerful nonlinear function to replace the original GeLU and halve the hidden dimension of the original FFN module:

$$\begin{aligned}\boldsymbol{Y}' &= \mathrm{AFFN}(\boldsymbol{X}) \\ &= \mathrm{concat}(\mathrm{AGeLU}(\boldsymbol{X}\boldsymbol{W}^d), \mathrm{AGeLU}'(\boldsymbol{X}\boldsymbol{W}^d))\boldsymbol{W}^e\end{aligned}$$

where $\boldsymbol{W}^d = \{w_{ij}^d\} \in \mathbb{R}^{C \times \frac{C'}{2}}$ and $\boldsymbol{W}^e = \{w_{ij}^e\} \in \mathbb{R}^{C' \times C}$ are weight matrices of two FC layers, and $\mathrm{AGeLU}(\cdot)$ and $\mathrm{AGeLU}'(\cdot)$ are two nonlinear functions with different parameters.

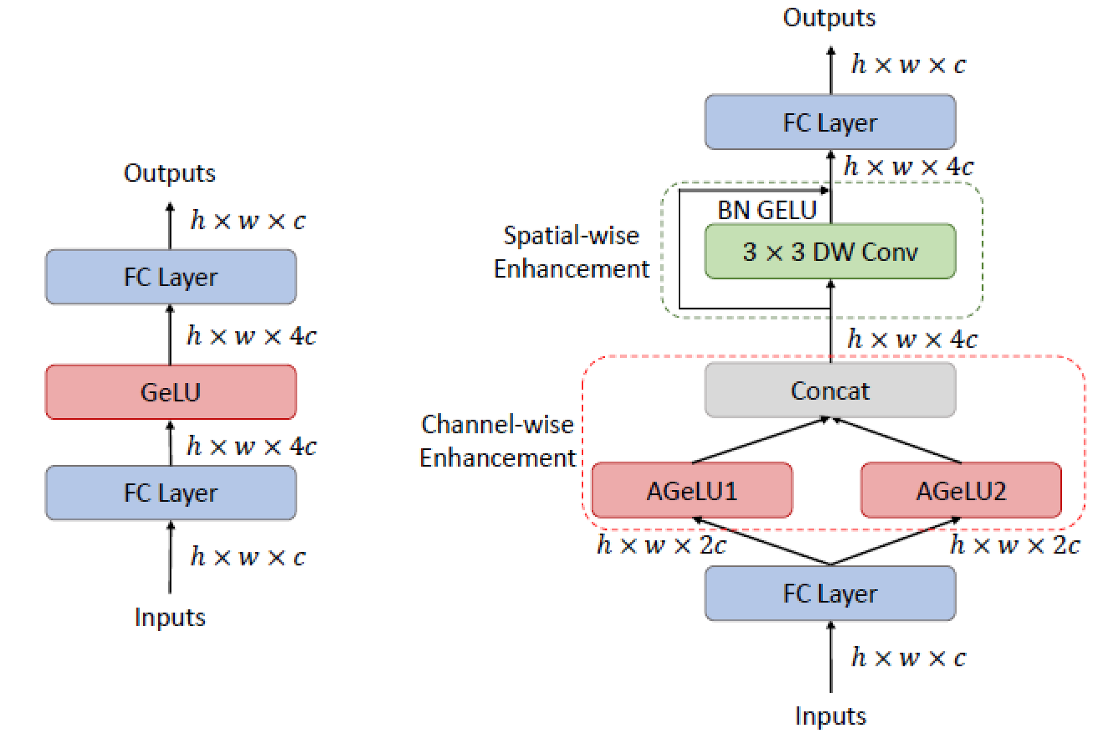
- Represent the above equation into its element-wise form:

$$\boldsymbol{y}' = \left(\sum_{j=1}^{C'} w_{jc}^{e}{}' \mathrm{GeLU}(m_{cj}' x_c + n_{cj}') + \theta_j\right)_{c=1}^{C},$$

- Compared to FFN module, AFFN module can generate the same degree of non-linearity. Each element y_c' in $\boldsymbol{y}'$ can also be treated as a linear combination of C' different nonlinear functions to the input element x_c with distinct scales and biases. Each scale is a learnable weight independent to the input while each bias is dependent on other input elements.

Spatial-wise Enhancement Part and Overall Architecture

- Channel-wise enhancement part extend non-linearity through the channel dimension. Thus ,we further enhance non-linearity with spatial dimension. We modify AFFN module by introducing a DW Block (DW Conv with BN and GeLU) after AGeLU, and form the final improved FFN (IFFN) module.



(a) Original FFN Module (b) IFFN Module (ours)

Experiments

Table 1: Image Classification results on ImageNet-1k dataset.

Methods	Architecture	Parameters (M)	FLOPs (G)	Top-1 Accuracy (%)
DeiT	DeiT-Ti	5.72	1.26	72.2
	+ IFFN	5.00 (-12.6%)	1.10 (-12.7%)	72.6
	DeiT-S	22.05	4.60	79.9
	+ IFFN	18.84 (-14.6%)	3.93 (-14.6%)	80.0
	DeiT-B	86.57	17.57	81.8
	+ IFFN*	73.66 (-14.9%)	14.92 (-15.1%)	81.8
Swin	Swin-Ti	28.29	4.50	81.2
	+ IFFN	24.29 (-14.1%)	3.88 (-13.8%)	81.5
	Swin-S	49.61	8.75	83.2
	+ IFFN	42.40 (-14.5%)	7.49 (-14.4%)	83.2
	Swin-B	87.77	15.44	83.5
	+ IFFN*	75.45 (-14.0%)	13.34 (-13.6%)	83.4
PoolFormer	PoolFormer-S12	11.92	1.82	77.2
	+ IFFN	9.80 (-17.8%)	1.48 (-18.7%)	77.2
	PoolFormer-S24	21.39	3.40	80.3
	+ IFFN	17.15 (-19.8%)	2.72 (-20.0%)	80.7
	PoolFormer-S36	30.86	4.99	81.4
	+ IFFN	24.50 (-20.6%)	3.97 (-20.4%)	81.5
	PoolFormer-M36	56.17	8.78	82.1
	+ IFFN	44.19 (-21.3%)	6.93 (-21.1%)	82.1
	PoolFormer-M48	73.47	11.56	82.5
	+ IFFN*	58.62 (-20.2%)	9.46 (-18.2%)	82.3
Portable ViT	LVT-R1	5.52	0.76	73.9
	+ IFFN*	4.98 (-9.8%)	0.68 (-10.5%)	74.0
	LVT-R2	5.52	0.84	74.8
	+ IFFN*	4.98 (-9.8%)	0.76 (-9.5%)	74.6
	LVT-R3	5.52	0.92	74.6
	+ IFFN*	4.98 (-9.8%)	0.84 (-8.7%)	74.8
	LVT-R4	5.52	1.00	74.9
	+ IFFN*	4.98 (-9.8%)	0.92 (-8.0%)	74.9

Table 2: Ablations on channel- and spatial-wise enhancement part.

Methods	Parameters (M)	FLOPs (G)	Top-1 Accuracy (%)
DeiT-Ti	5.72	1.26	72.2
w/ channel	4.89	1.08	70.5
w/ spatial	5.83	1.28	72.8
w/ channel & spatial	5.00	1.10	72.6

Table 3: Using different kernel size in spatial-wise enhancement part.

n	Parameters (M)	FLOPs (G)	Top-1 Acc (%)
1	4.92	1.08	72.0
3	5.00	1.10	72.6
5	5.15	1.13	72.8
7	5.37	1.17	72.9