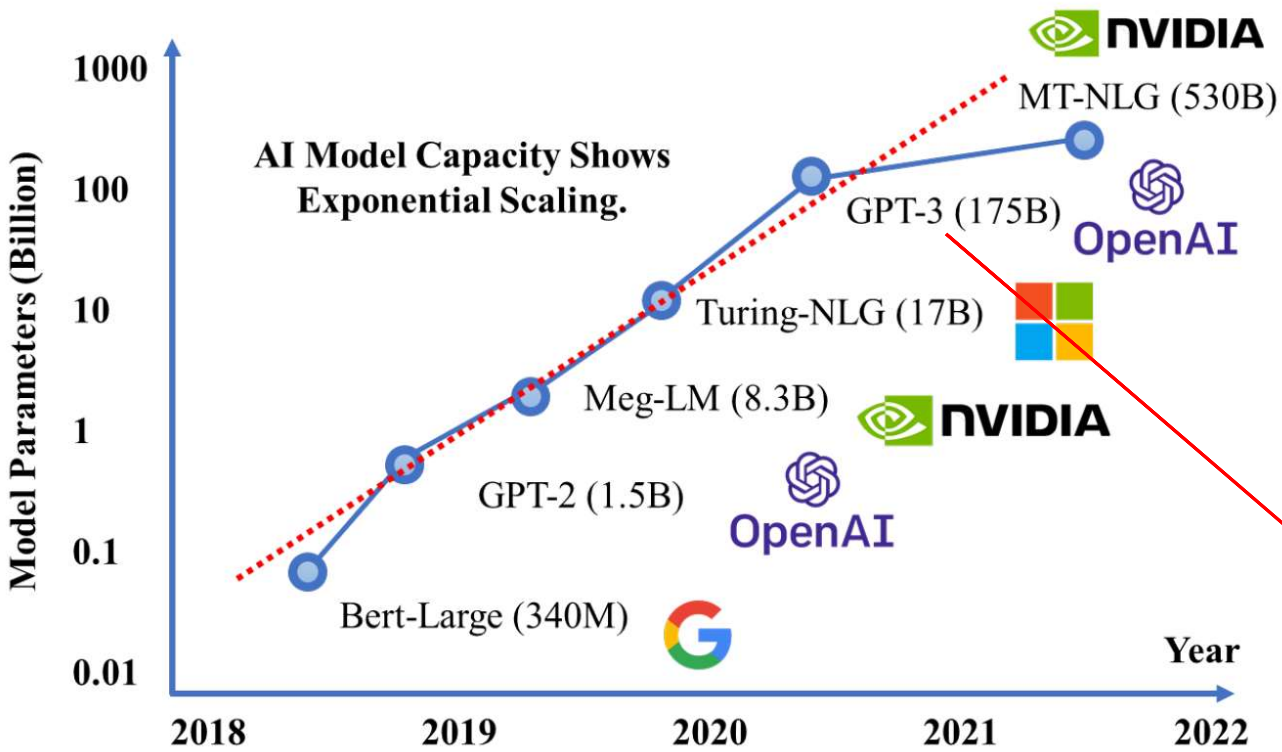
Learning with Auxiliary Activation for Memory-Efficient Training

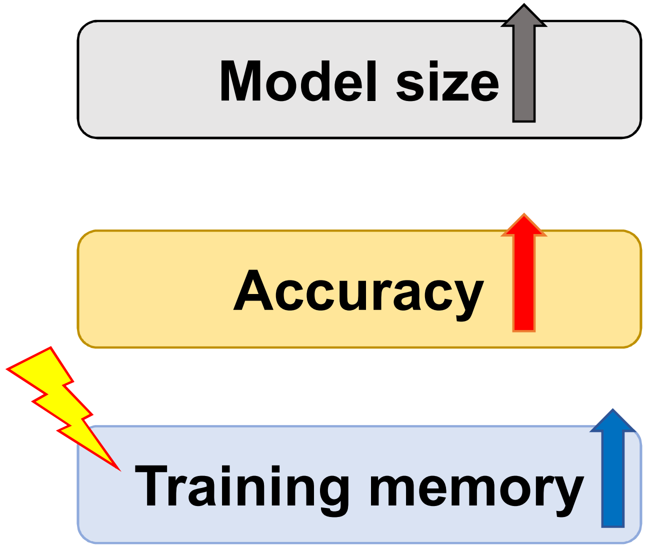
Sunghyeon Woo, Dongsuk Jeon
{wsh0917, djeon1@snu.ac.kr}
Seoul National University

Mobile Multimedia Systems Group

Intordúction



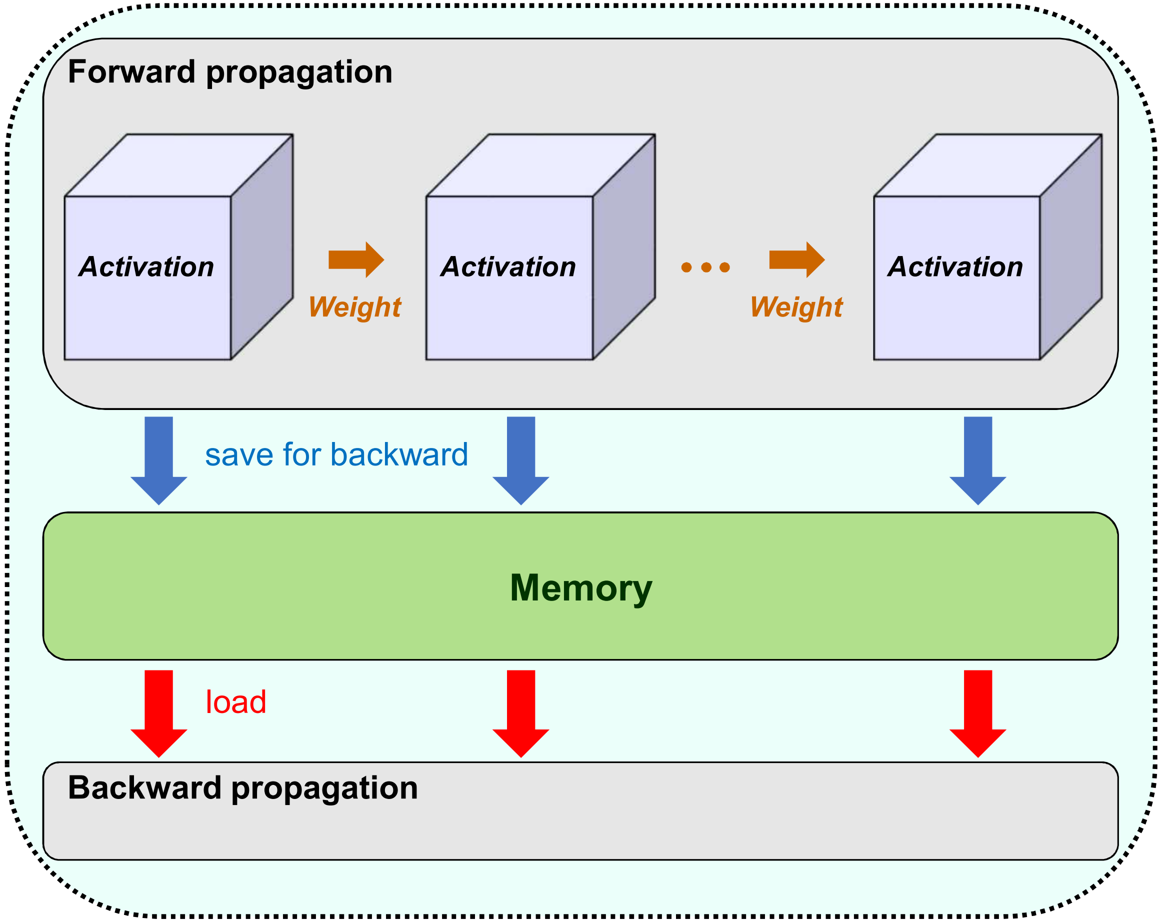
< Rapid growth of model parameters (Yu et al, 2021) >



1024 x A100 Requires …

Intorduction

Backpropagation



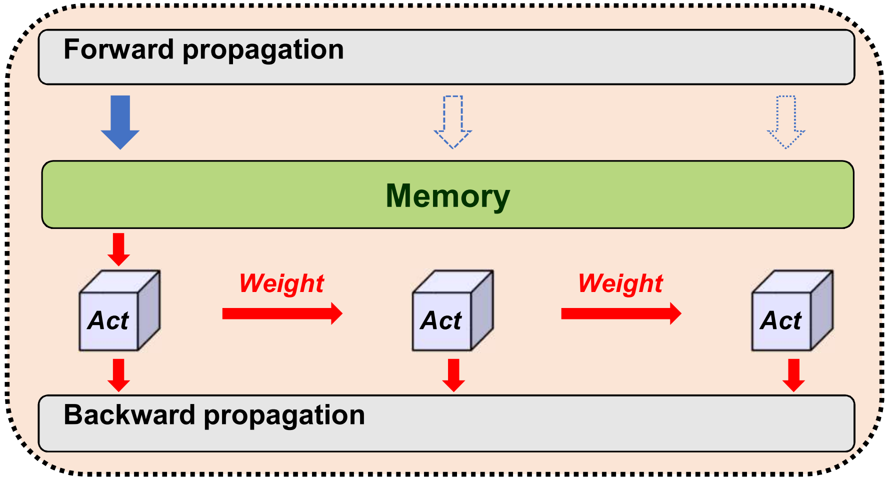
Activations are stored in memory for backward propagation...

Large memory is required!

Intordution

Backpropagation

Activation Recomputation



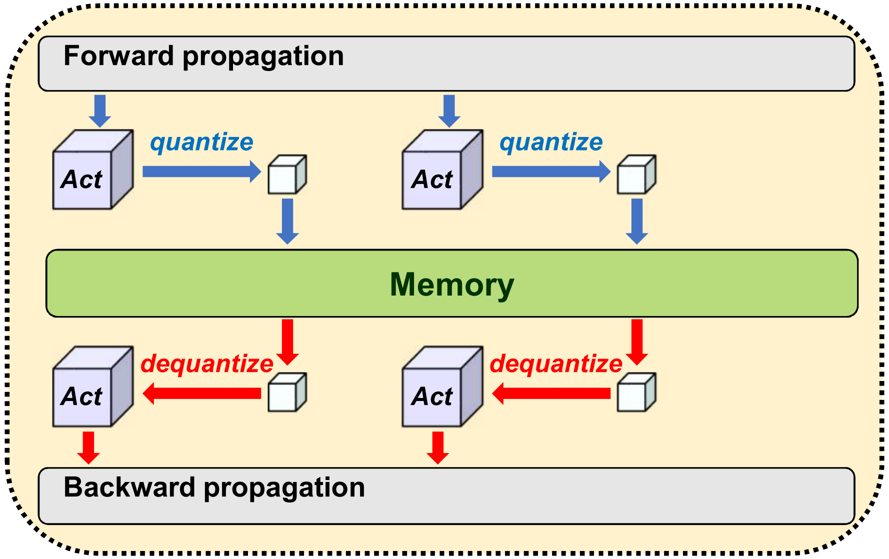
Activation Compression Training

Intorduction

Backpropagation

Activation Recomputation

Activation Compression Training

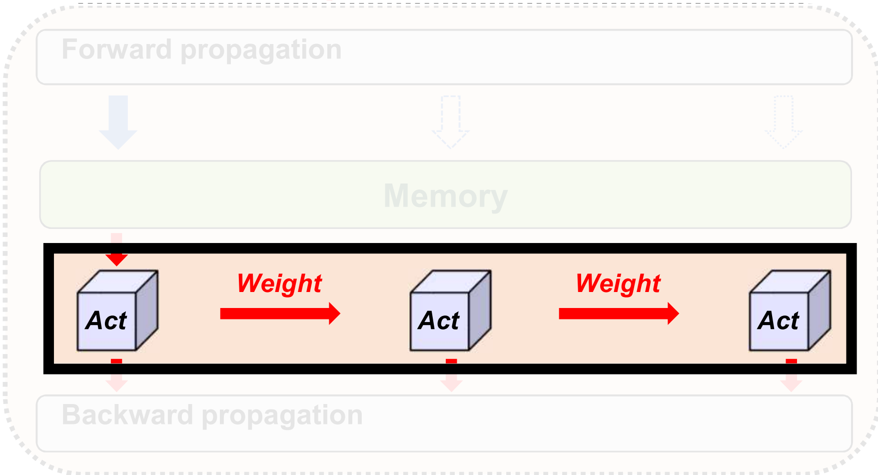


Intorduction

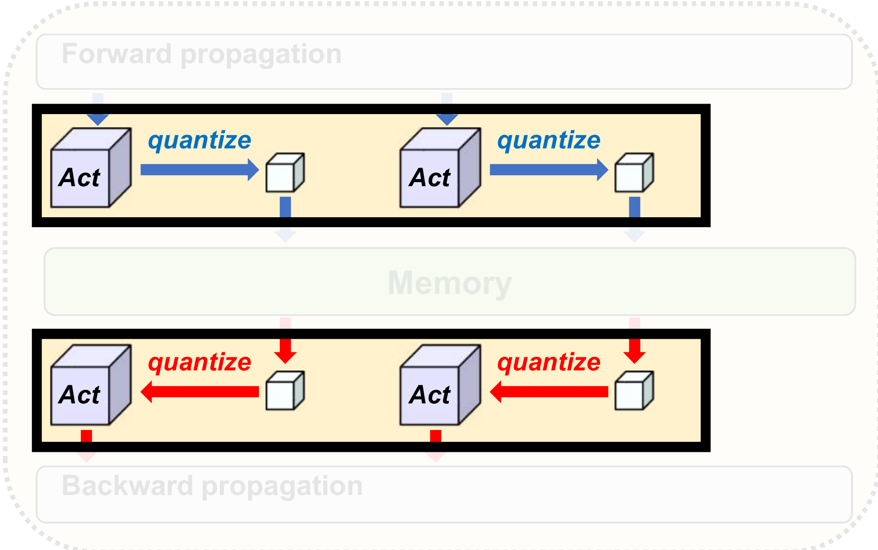
Additional process
in forward or backward propagation...

Tranining speed is reduced! 😢

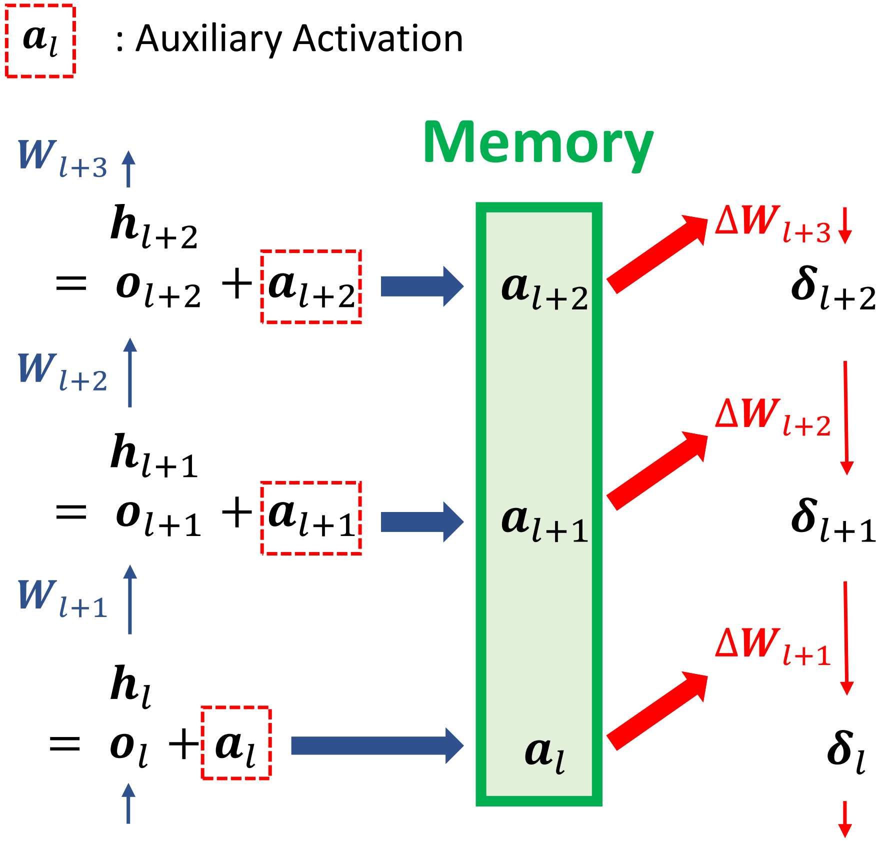
Activation Recomputation



Activation Compression Training



Auxiliary Activation Learning



Auxiliary Activation Learning

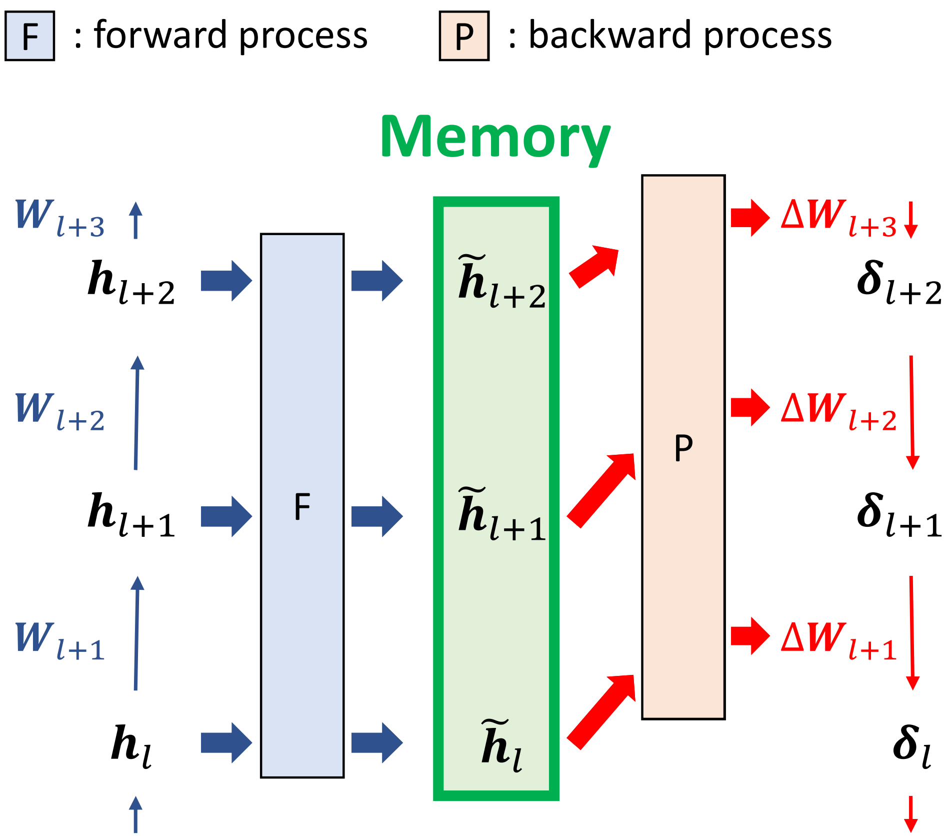
- **Forward Propagation**

  : Add auxiliary activation to output activation of layer and store auxiliary activation instead of actual activation

- **Backward Propagation**

  : Use auxiliary activation for updating weights

Auxiliary Activation Learning



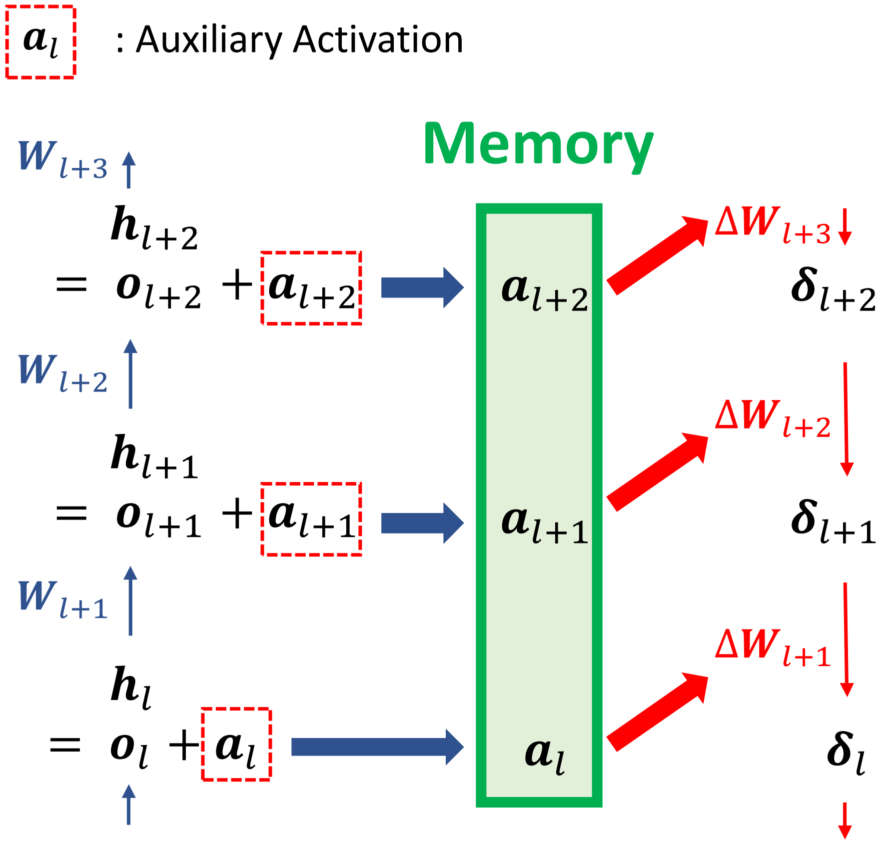
Conventional memory-saving algorithms

Auxiliary Activation Learning

Auxiliary Activation Learning

We can reduce training memory without additional processes

There is no speed reduction! 😀



Auxiliary Activation Learning

Learing criterion for alternative activation

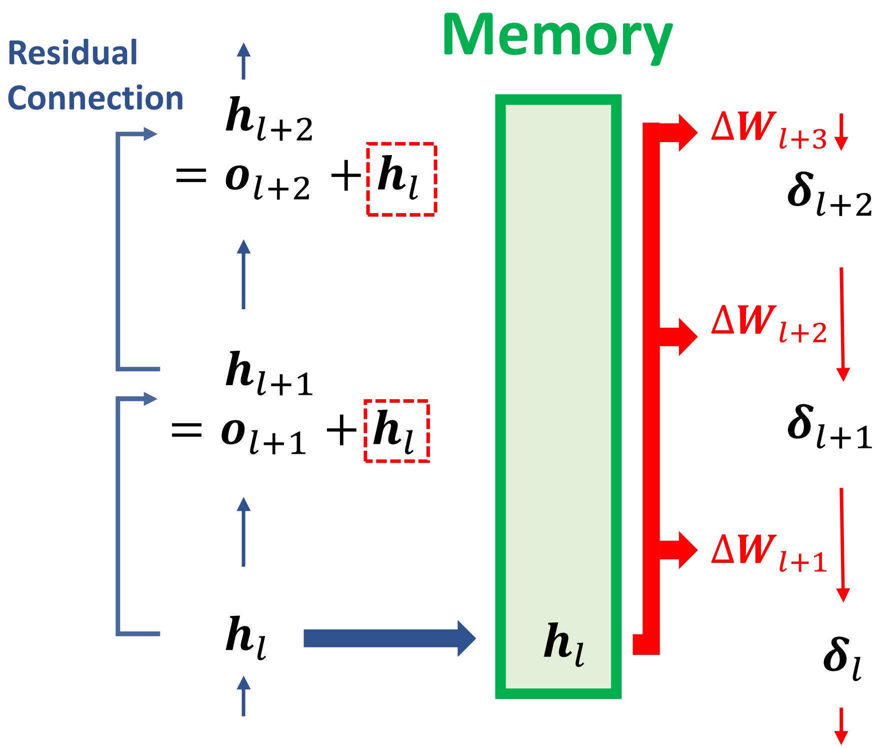
Theorem 1. *Let weight updates* $\boldsymbol{W}_{l+1}$ *is calculated by alternative activation* $\boldsymbol{r}_l$ *instead of actual activation* $\boldsymbol{h}_l$. *In this case, if the gradient of loss function* $f(\boldsymbol{W})$ *is L-Lipschitz continuous, learnin rate* η *satisfies* $0 \leq \eta \leq \frac{1}{L}$, *and* $\boldsymbol{r}_l^T(2\boldsymbol{h}_l - \boldsymbol{r}_l) \geq 0$, *then loss function* $f(\boldsymbol{W})$ *is converged.*

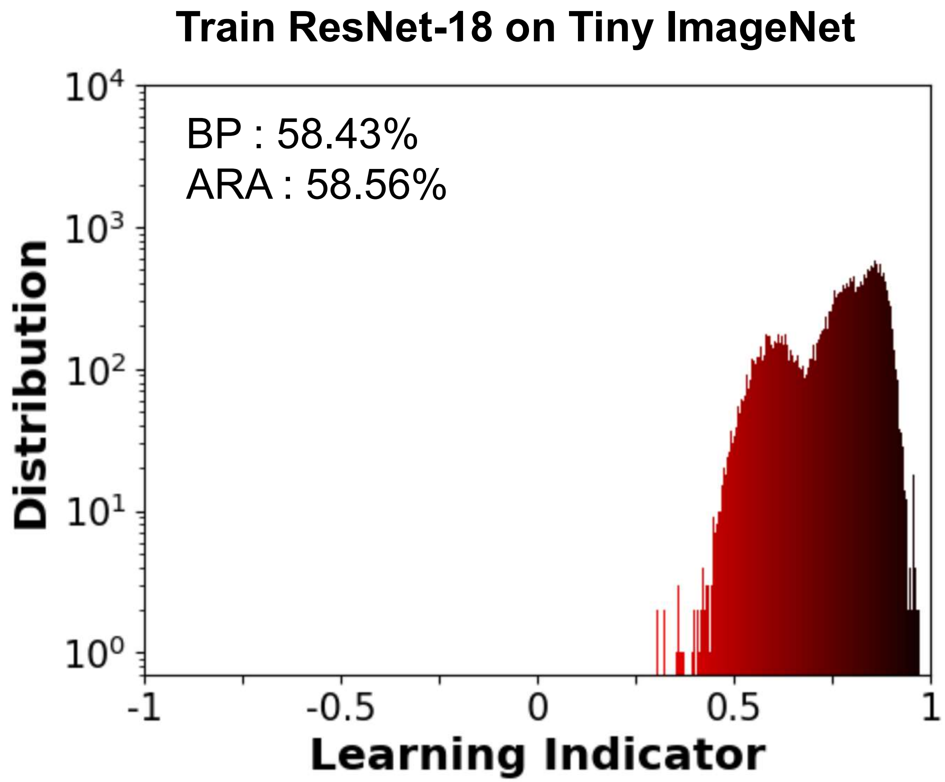
$$\boldsymbol{Learing\ Indicator\ (LI)} \equiv \frac{\boldsymbol{r}_l^T(2\boldsymbol{h}_l - \boldsymbol{r}_l)}{\|\boldsymbol{h}_l\|^2}$$

- $\boldsymbol{LI} > \boldsymbol{0}$: The loss is converged. The larger Learning indicator is, the better loss converges.
- $\boldsymbol{LI} \leq \boldsymbol{0}$: The loss is diverged.

Two candidates of Auxiliary Activation

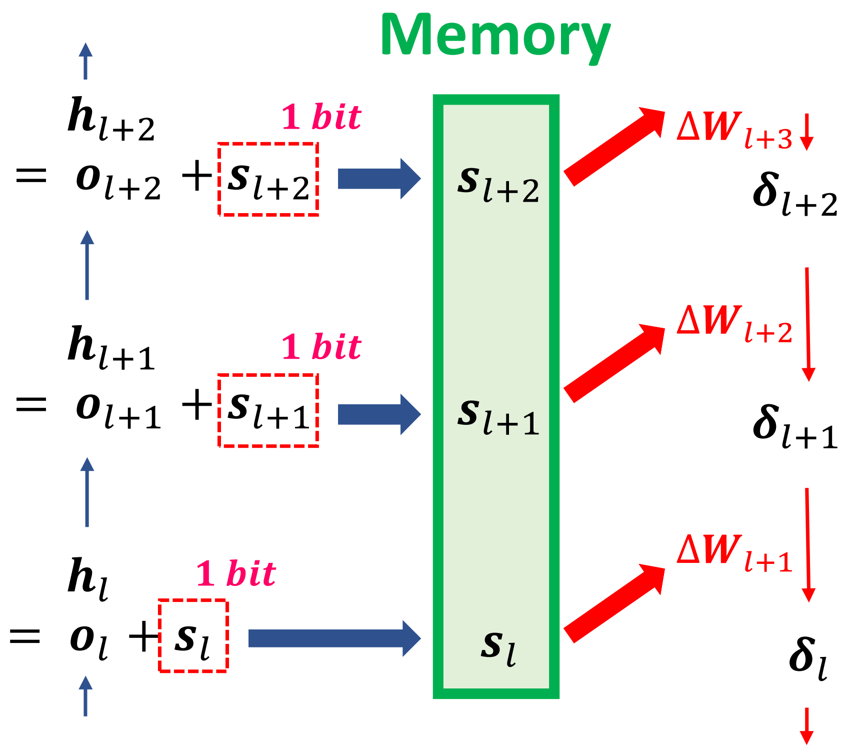
$\boxed{\boldsymbol{h}_l}$: Auxiliary Residual Activation (**ARA**)

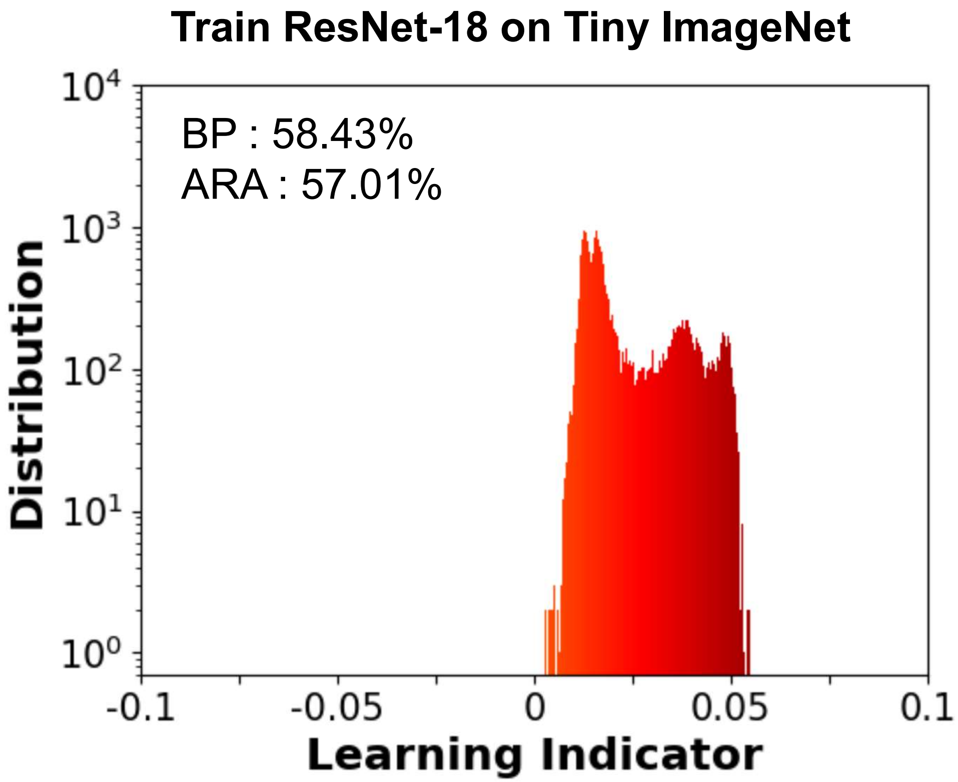




Two candidates of Auxiliary Activation

s_l : Auxiliary Sign Activation (**ASA**)





Results

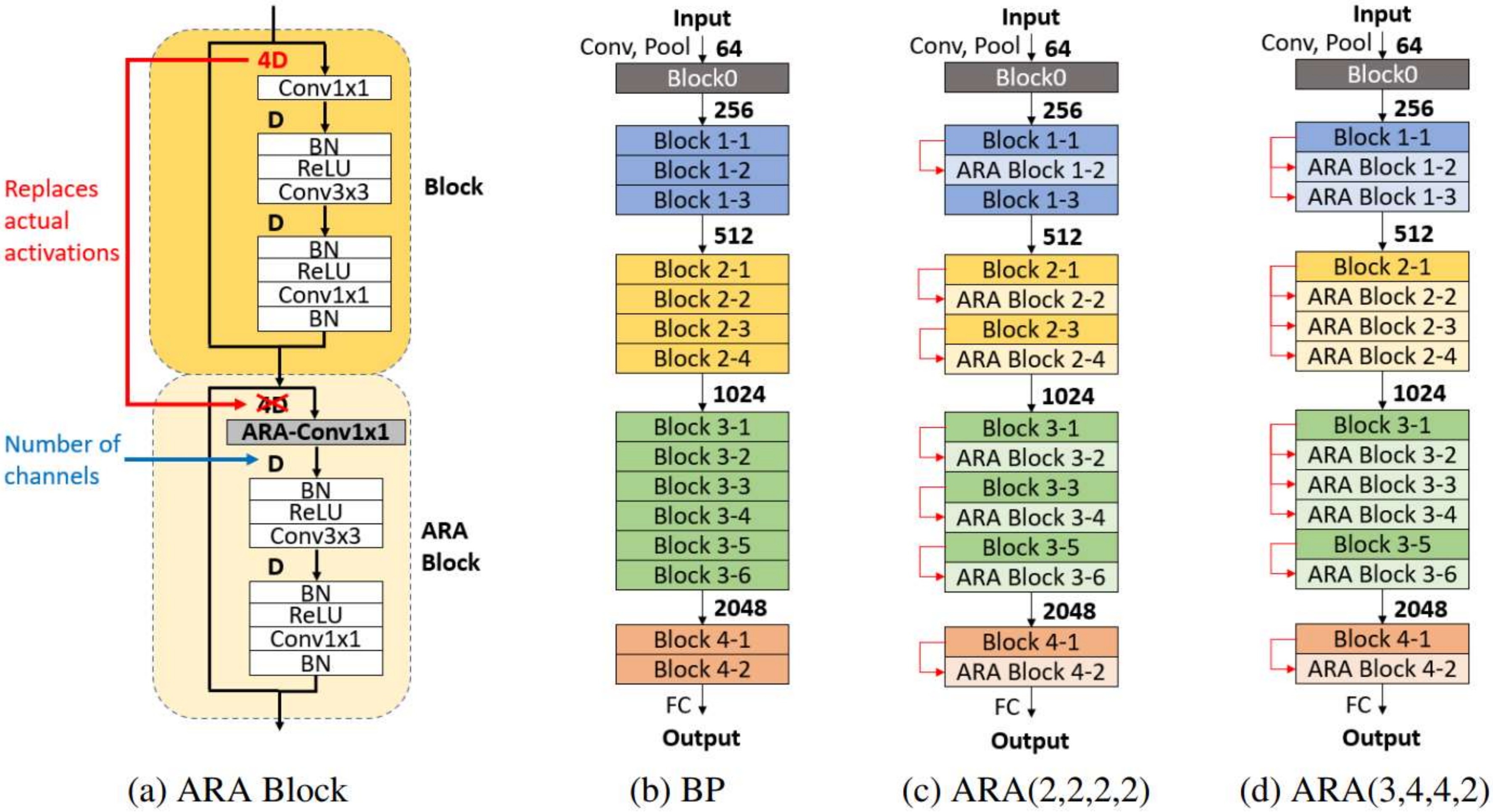


Figure 3: Training ResNet-50 using Auxiliary Residual Activation (ARA).

Results

Table 2: Test accuracy, training memory with compression rate (bracketed), and training time (italic) of one epoch in ResNet training on ImageNet with 512 batch size and six RTX-3090 GPUs.

Models	Baseline	-	ARA (2,2,2,2)	ARA (3,4,2,2)	ARA (3,4,4,2)	ARA (3,4,6,2)
ResNet-50	BP	**76.01** 44.6 GB *17m 35s*	75.97 (1.12x) *17m 59s*	75.89 (1.2x) *17m 54s*	75.62 (1.21x) *17m 55s*	75.23 (1.22x) *18m 3s*
	GCP	**76.01** (2.2x) *36m 42s*	75.97 (2.88x) *37m 2s*	75.89 (3.44x) *37m 9s*	75.62 (3.54x) *36m 58s*	75.23 (3.66x) *37m 14s*
	ActNN	75.96 (11.8x) *48m 48s*	75.93 (14x) *48m 48s*	75.67 (14.6x) *48m 10s*	75.51 (14.8x) *47m 58s*	75.12 (**15.01**x) *47m 08s*
ResNet-152	BP	77.38 90.2 GB *35m 37s*	77.14 (1.16x) *36m 17s*	**77.41** (1.21x) *36m 24s*	76.84 (1.27x) *36m 11s*	76.64 (1.29x) *36m 30s*
	GCP	77.38 (2.1x) *1h 18m*	77.14 (2.92x) *1h 19m*	**77.41** (3.26x) *1h 19m*	76.84 (3.75x) *1h 19m*	76.64 (**3.95**x) *1h 20m*

Save memory without speed reduction

Results

Table 2: Test accuracy, training memory with compression rate (bracketed), and training time (italic) of one epoch in ResNet training on ImageNet with 512 batch size and six RTX-3090 GPUs.

Models	Baseline	-	ARA (2,2,2,2)	ARA (3,4,2,2)	ARA (3,4,4,2)	ARA (3,4,6,2)
ResNet-50	BP	**76.01** 44.6 GB *17m 35s*	75.97 (1.12x) *17m 59s*	75.89 (1.2x) *17m 54s*	75.62 (1.21x) *17m 55s*	75.23 (1.22x) *18m 3s*
	GCP	**76.01** (2.2x) *36m 42s*	75.97 (2.88x) *37m 2s*	75.89 (3.44x) *37m 9s*	75.62 (3.54x) *36m 58s*	75.23 (3.66x) *37m 14s*
	ActNN	75.96 (11.8x) *48m 48s*	75.93 (14x) *48m 48s*	75.67 (14.6x) *48m 10s*	75.51 (14.8x) *47m 58s*	75.12 (**15.01**x) *47m 08s*
ResNet-152	BP	77.38 90.2 GB *35m 37s*	77.14 (1.16x) *36m 17s*	**77.41** (1.21x) *36m 24s*	76.84 (1.27x) *36m 11s*	76.64 (1.29x) *36m 30s*
	GCP	77.38 (2.1x) *1h 18m*	77.14 (2.92x) *1h 19m*	**77.41** (3.26x) *1h 19m*	76.84 (3.75x) *1h 19m*	76.64 (**3.95**x) *1h 20m*

Orthogonality: maximize memory saving

Results

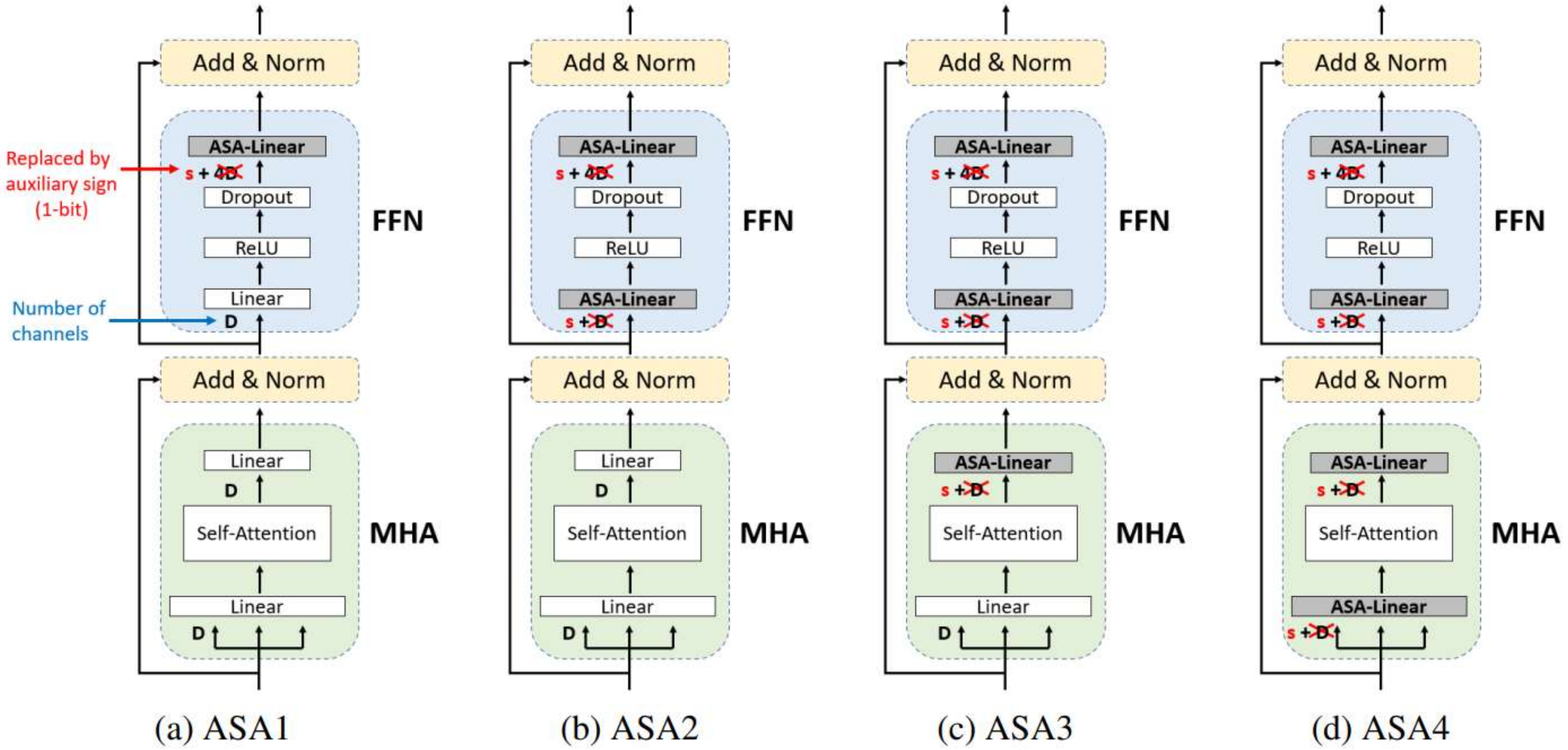


Figure 5: Transformer-like networks trained using Auxiliary Sign Activations (ASA). MHA and FFN represent multihead attention and feedforward network, respectively.

Results

Table 3: Test/validation scores, compression rate (bracketed), and training time (italic) of one epoch of Transformer, BERT, ViT, and MLP-Mixer training with 4096, 32, 512, and 256 batch sizes, respectively.

Models	Dataset	Baseline	-	ASA1	ASA2	ASA3	ASA4
Trans-former	IWSLT	BP	35.23 3.6 GB *3m 15s*	35.44 (1.1x) *3m 21s*	34.87 (1.1x) *3m 23s*	34.9 (1.2x) *3m 26s*	35.02 (1.3x) *3m 38s*
		Mesa	**35.45** (1.6x) *5m 52s*	34.74 (1.7x) *5m 47s*	35.19 (1.7x) *5m 44s*	35.11 (1.7x) *5m 36s*	34.84 (**1.8**x) *5m 8s*
BERT-Large	MRPC	BP	88.56 10.9 GB *1m 28s*	88.69 (1.1x) *1m 29s*	88.23 (1.2x) *1m 30s*	**88.97** (1.3x) *1m 30s*	88.32 (1.3x) *1m 31s*
		Mesa	88.3 (2.1x) *2m 8s*	88.35 (2.3x) *2m 5s*	88.51 (2.4x) *2m 3s*	88.25 (2.4x) *2m 1s*	88.23 (**2.5**x) *1m 55s*
	MNLI	BP	86.52 10.9 GB *2h 37m*	**86.65** (1.1x) *2h 40m*	86.49 (1.2x) *2h 40m*	86.42 (1.3x) *2h 41m*	86.39 (1.3x) *2h 41m*
		Mesa	86.32 (2.1x) *3h 51m*	86.37 (2.3x) *3h 39m*	86.29 (2.4x) *3h 35m*	86.54 (2.4x) *3h 32m*	86.17 (**2.5**x) *3h 24m*
ViT-Large	CIFAR-100	BP	92.93 48 GB *2m 46s*	92.81 (1.3x) *2m 48s*	92.84 (1.5x) *2m 50s*	**92.97** (1.5x) *2m 51s*	92.65 (1.6x) *2m 52s*
		Mesa	92.89 (3.0x) *4m 46s*	92.75 (3.5x) *4m 15s*	92.72 (3.6x) *3m 57s*	92.95 (3.7x) *3m 49s*	92.84 (**4.3**x) *3m 33s*
Mixer-Large	CIFAR-100	BP	91.62 86.8 GB *3m 39s*	91.45 (1.3x) *3m 47s*	91.72 (1.4x) *3m 50s*	91.66 (1.7x) *3m 57s*	**91.91** (**2.0**x) *4m 1s*

Results

Table 4: Largest models that can be trained using a single GPU with 24GB memory. (ResNet: depth = number of layers, width = width of the first bottleneck block. BERT-Large: depth = number of transformer blocks, width = hidden size.)

Models	ResNet				BERT-Large			
Learning rule	BP	ARA	ActNN	ARA+ActNN	BP	ASA4	Mesa	Mesa+ActNN
Depth	146	165	622	718	50	60	64	70
Width	62	76	214	238	1600	1728	1792	1856

Thank you very much!