Not All Layers Are Equal: A Layer-Wise Adaptive Approach Toward Large–Scale DNN Traiing

THE WEB CONFERENCE ACM

2022-04-27

Yunyong Ko[1], Dongwon Lee[2], and Sang–Wook Kim[1]
Hanyang University, Republic of Korea[1]
The Pennsylvania State University, PA, USA[2]

Table of Contents

- Background
- Motivation
- Proposed algorithm: LENA
  - Layer-wise LR scaling
  - Layer-wise state-ware warm-up
- Experiments
- Conclusions

Background

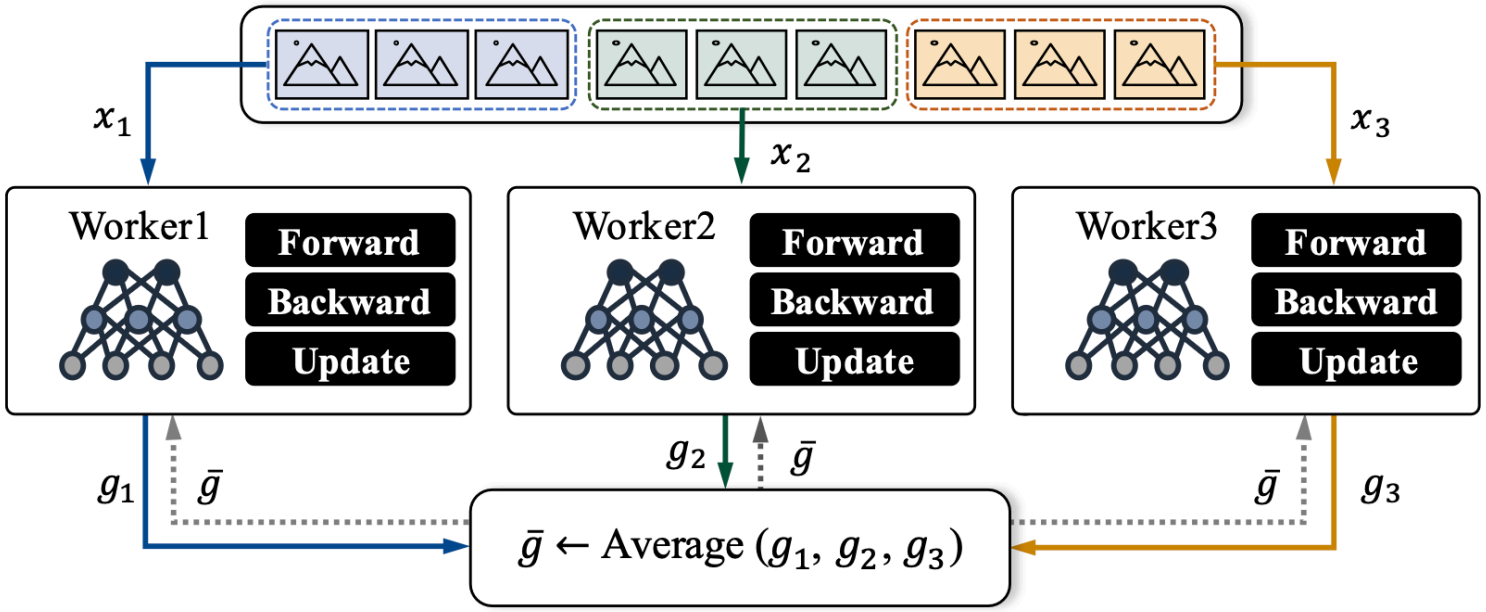
- **Web-based applications adopt deep learning (DL) techniques**
  - Training a large amount of ***web-scale data*** generated from users

- **Training of a deep neural network (DNN) model with web-scale data requires massive time**

Training with data parallelism

☐ Splitting and distributing *training data* into multiple workers

- Each worker computes its gradient based on its local data ***in parallel***
- Their gradients are averaged and applied to the model of each worker

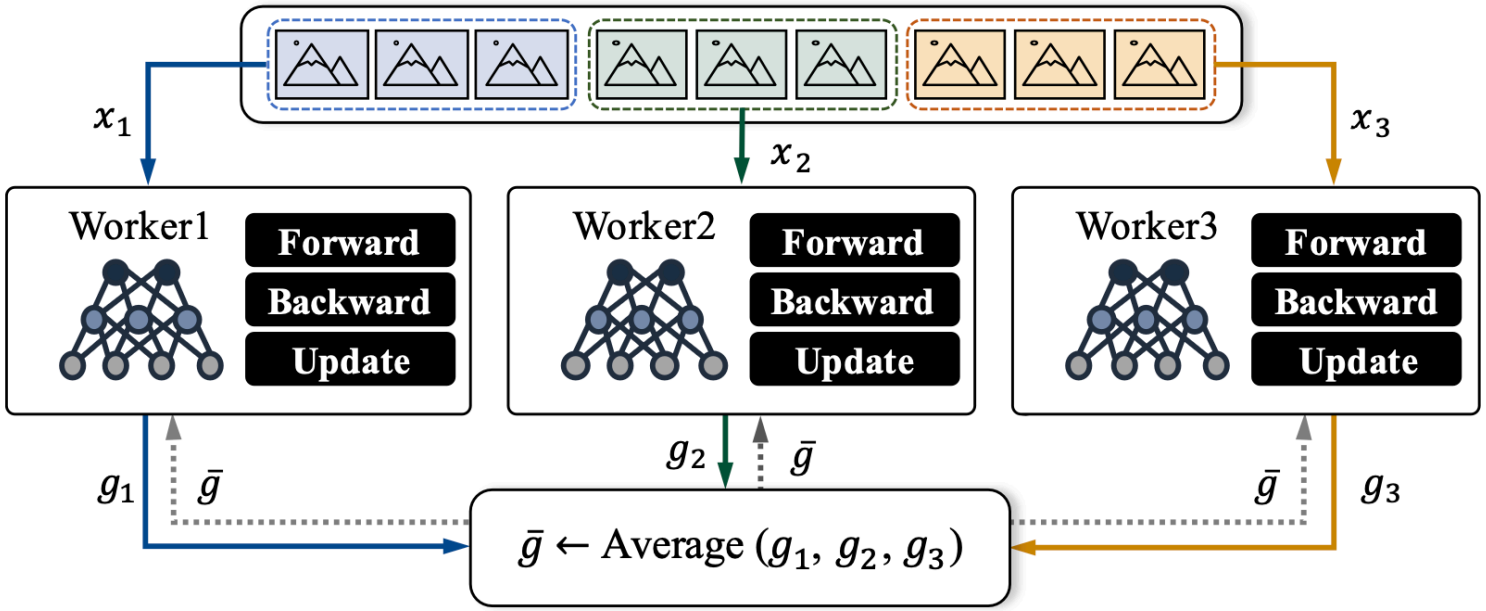


<Architecture of a large-batch training with data parallelism>

Training with data parallelism

Reducing the total number of training iterations (and time)

- As the number of workers increases, the processed data increases proportionally



<Architecture of a large-batch training with data parallelism>

Learning rate (LR) scaling

- **Too small number of iterations may *degrade the model quality***
  - *Fewer chances toward the model convergence*

- **Learning rate (LR) scaling**
  - ***Increasing the learning rate* in order to conpensate for fewer iterations**
    - i.e., boosting each gradient update to be larger
  - Given the model *w*, dataset *x*, and *k* workers, the goal of lr scaling is defined as

$$F(w,x)^{(1)} \approx F(w,x)^{(k)}$$

- $F(w,x)^{(1)}$: the result of single-worker training
- $F(w,x)^{(k)}$: the result of multi-worker (large-batch) training

Existing LR scaling methods

Linear scaling with warm-up (LSW, arXiv'17)

- Multiplying LR by k when the batch size increases by k times (linear scaling)
- Gradually increasing LR from a small value to the target value (warm-up)

Gradient variance based scaling (AdaScale, ICML'20)

- *Estimating the state of the training via the gradient variance*
- Then, *adaptively adjusting LR depending on the gradient variance* at each iteration
  - Case 1 (high gradient variance): it increases LR almost linearly
    - Similar to the linear scaling
  - Case 2 (low gradient variance): it rarely increases LR
    - Similar to the identical scaling

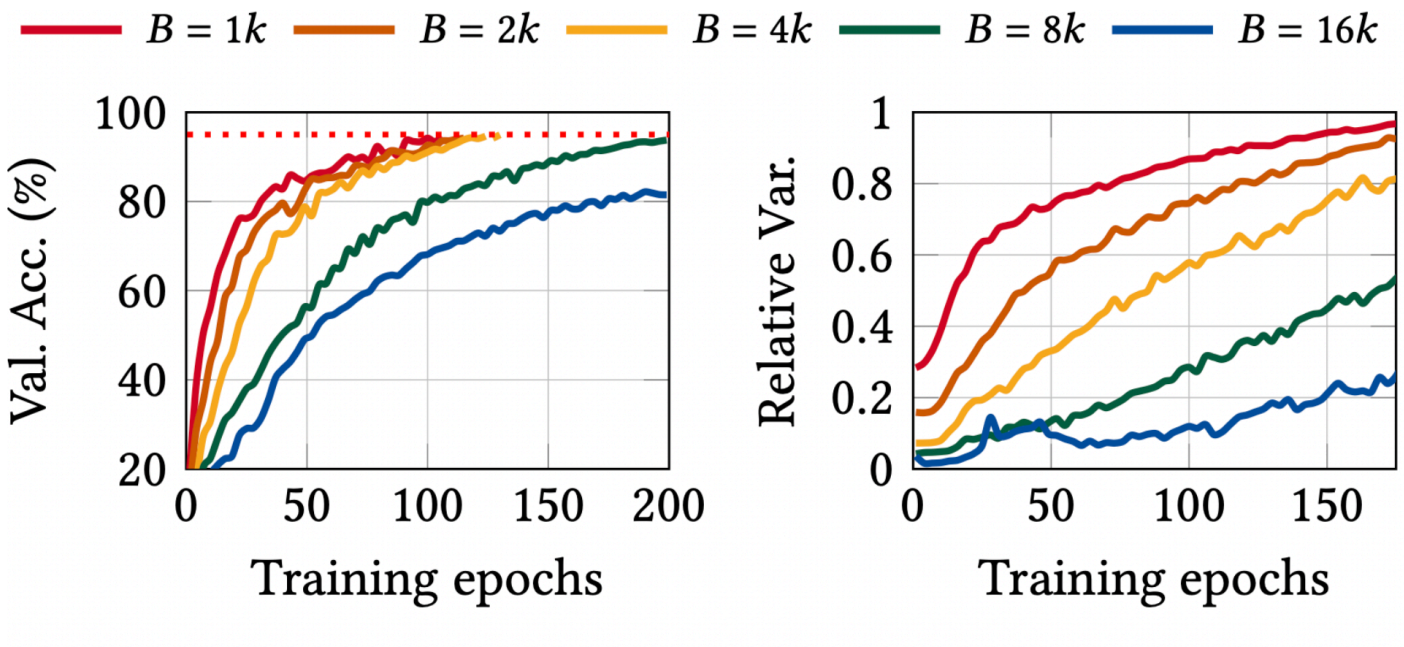
Table of Contents

- Background
- Motivation
- Proposed algorithm: LENA
  - Layer-wise LR scaling
  - Layer-wise state-ware warm-up
- Experiments
- Conclusions

Motivation

☐ Gradient variance tends to increase as the model gets trained

- Estimating ***how well the model has been trained*** (i.e., state of the training)



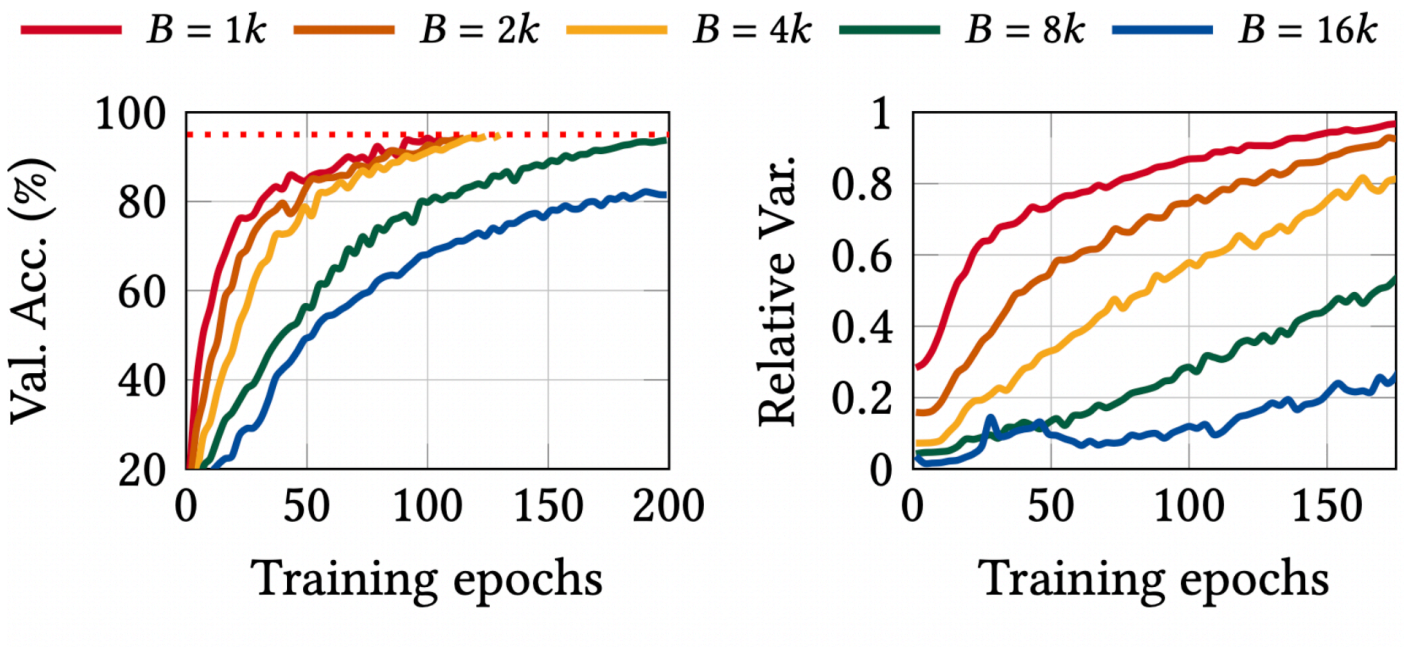
(a) Model accuracy. **(b) Relative variance.**

<The relationship between the model quality and gradient variance in large-batch training with AdaScale>
(B indicates the total batch size)

Motivation

☐ The model quality of AdaScale degraded with the very large batch size

- E.g., 16k of the batch size in the training of ResNet-18 on CIFAR-10



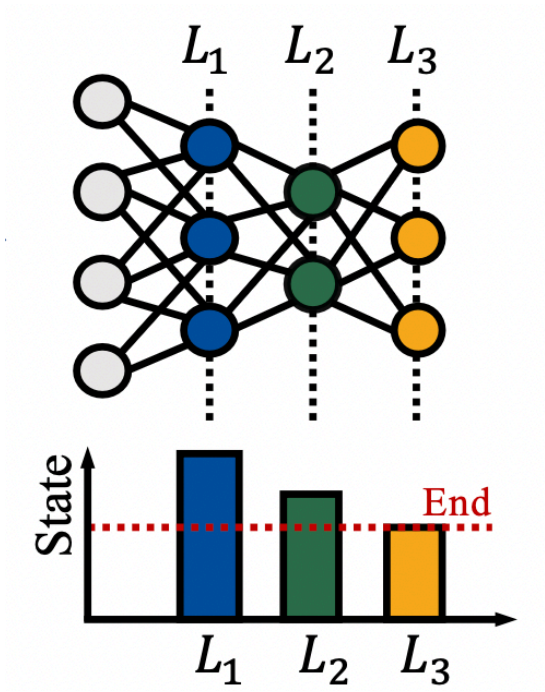
(a) Model accuracy. (b) Relative variance.

<The relationship between the model quality and gradient variance in large-batch training with AdaScale>
(B indicates the total batch size)

Motivation

Our hypothesis

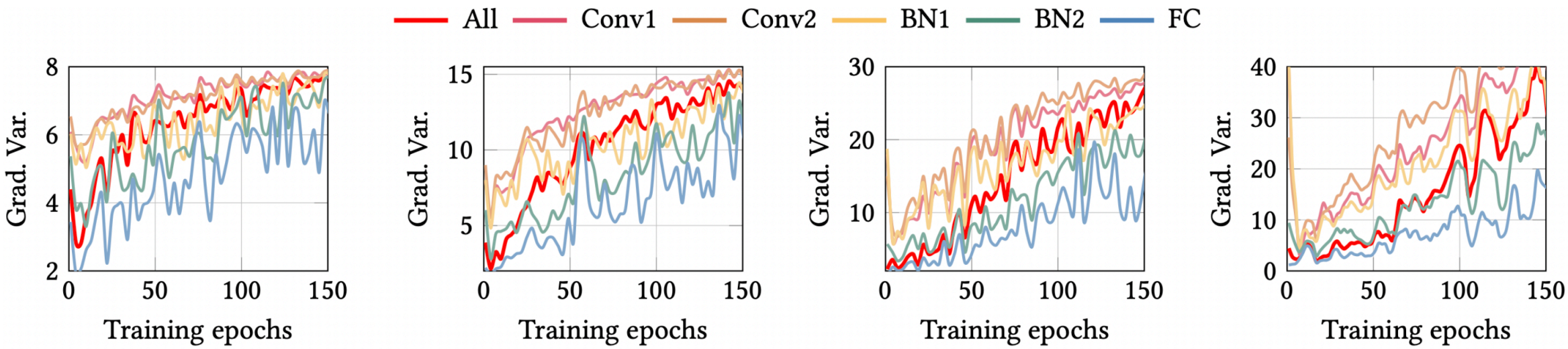
- The parameters of a DNN model have their own roles
  - i.e., the information to capture from data is not equal across layers
- ***The state of the training can vary across layers*** during the training of a DNN model



Observation

The gradient variance tends to vary across layers

- ***The status of the training of parameters are not uniform across layers***
  - The differences get larger in larger-batch training



(a) CIFAR-10 ($B = 1024$) **(b) CIFAR-10** ($B = 2048$) **(c) CIFAR-10** ($B = 4096$) **(d) CIFAR-10** ($B = 8192$)

Limitation of existing LR scaling method

- ***Not taking into account the differences among layers***
  - Adjusting LR based on the gradient variance of parameters for the entire model
  - Then, applying the adjusted LR to ***all model parameters equally!***

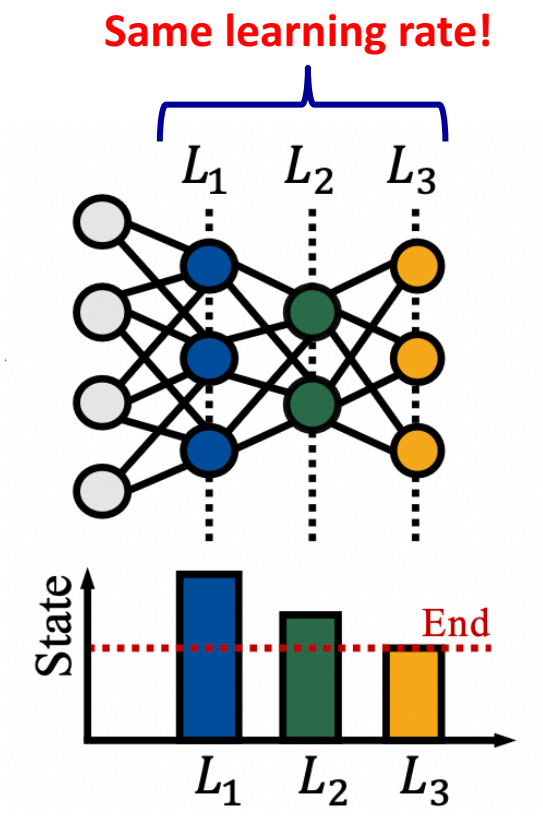


Table of Contents

- Background
- Motivation
- Proposed algorithm: LENA
  - Layer-wise LR scaling
  - Layer-wise state-ware warm-up
- Experiments
- Conclusions

Layer-wise LR scaling

☐ *A layer-wise adaptive LR scaling considering the difference across layers*

- Computing the gradient variance of each layer ***individually***
  - ☐ $grad_var_{(t,k)}$: the gradient variance of layer *k* at iteration *t*
  - ☐ $g^{(i)}$*:* the gradient of worker *i,* $\bar{g}$*: the averaged gradient*

$$grad_var_{(t,k)} = \frac{\frac{1}{n}\sum_{i=0}^{n-1} \|g_{(t,k)}^{(i)}\|^2}{\|\bar{g}_{(t,k)}\|^2}$$

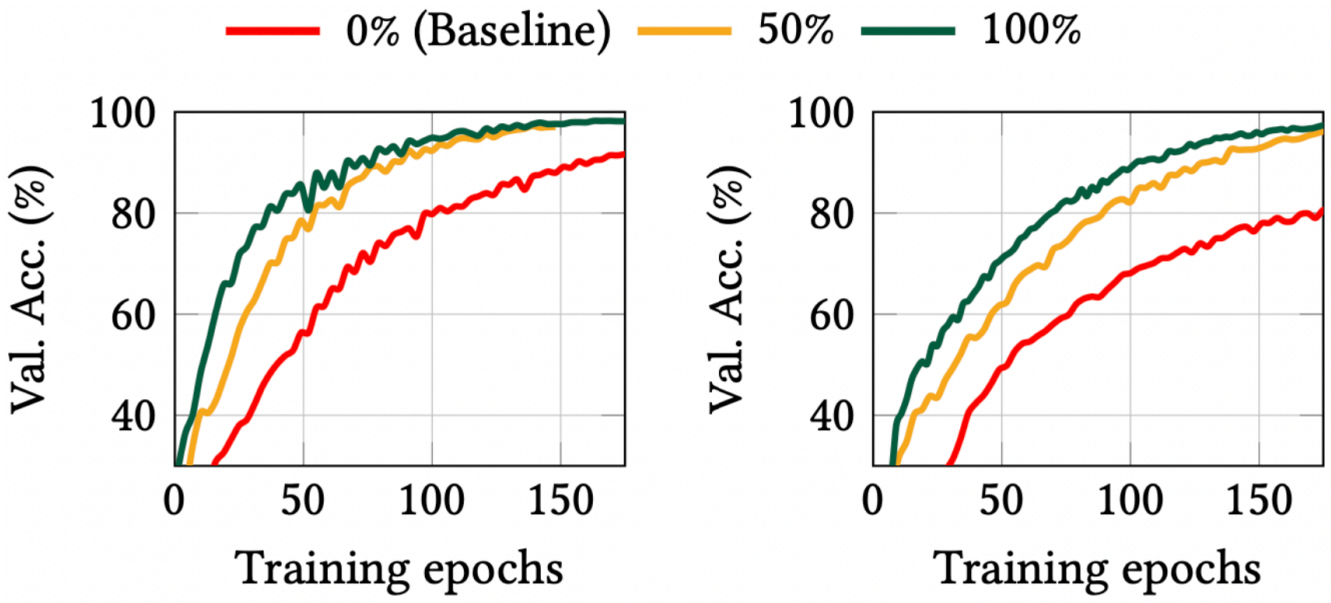
- Adjusting LR for each layer based on its gradient variance
  - ☐ $\eta_{(t,k)}$: the learning rate for layer *k* at iteration *t*

$$\eta_{(t,k)} = grad_var_{(t,k)} \cdot \eta_0$$

Potential of our layer-wise LR scaling

☐ *Considering the training differences across layers is critical*

- The existing LR scaling methods ignore the subtle differences across layers
  - ☐ Thus, suffering from the degradation of the model quality

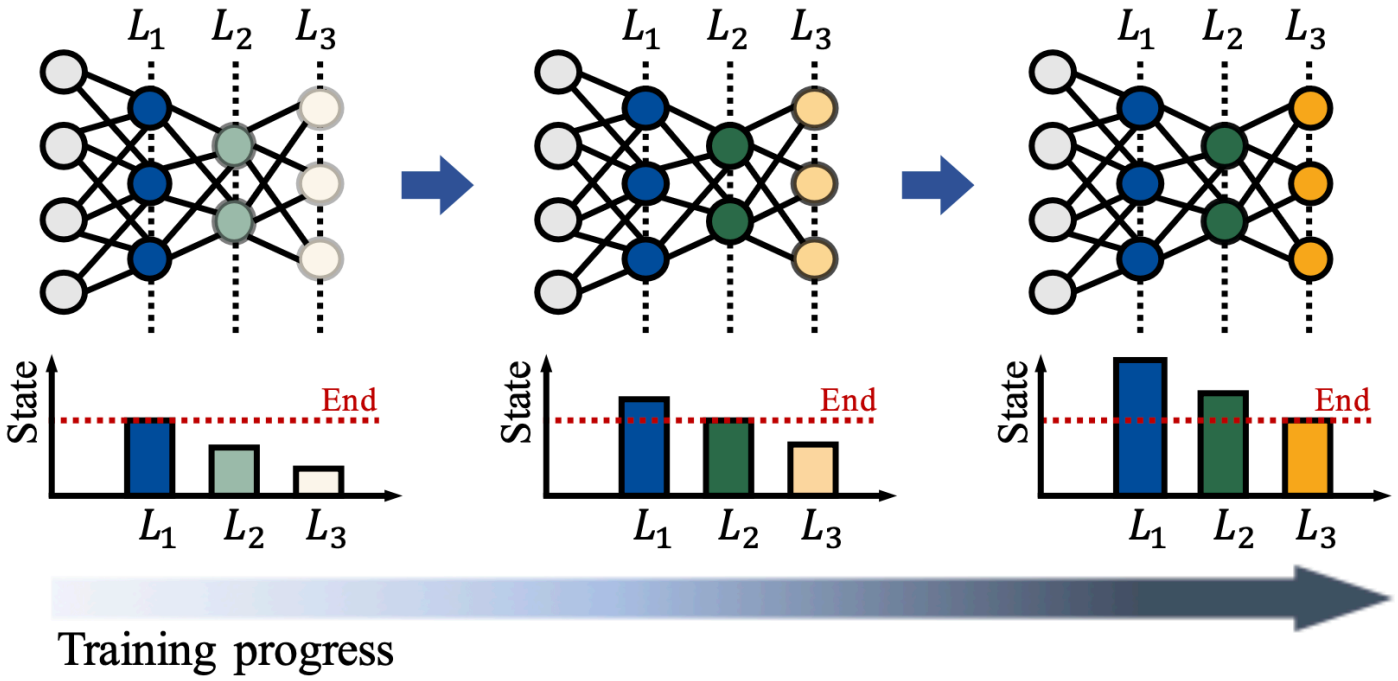


(a) CIFAR-10 ($B = 8k$) **(b) CIFAR-10** ($B = 16k$)

Layer-wise state-aware warm-up

☐ The training speed can vary across layers

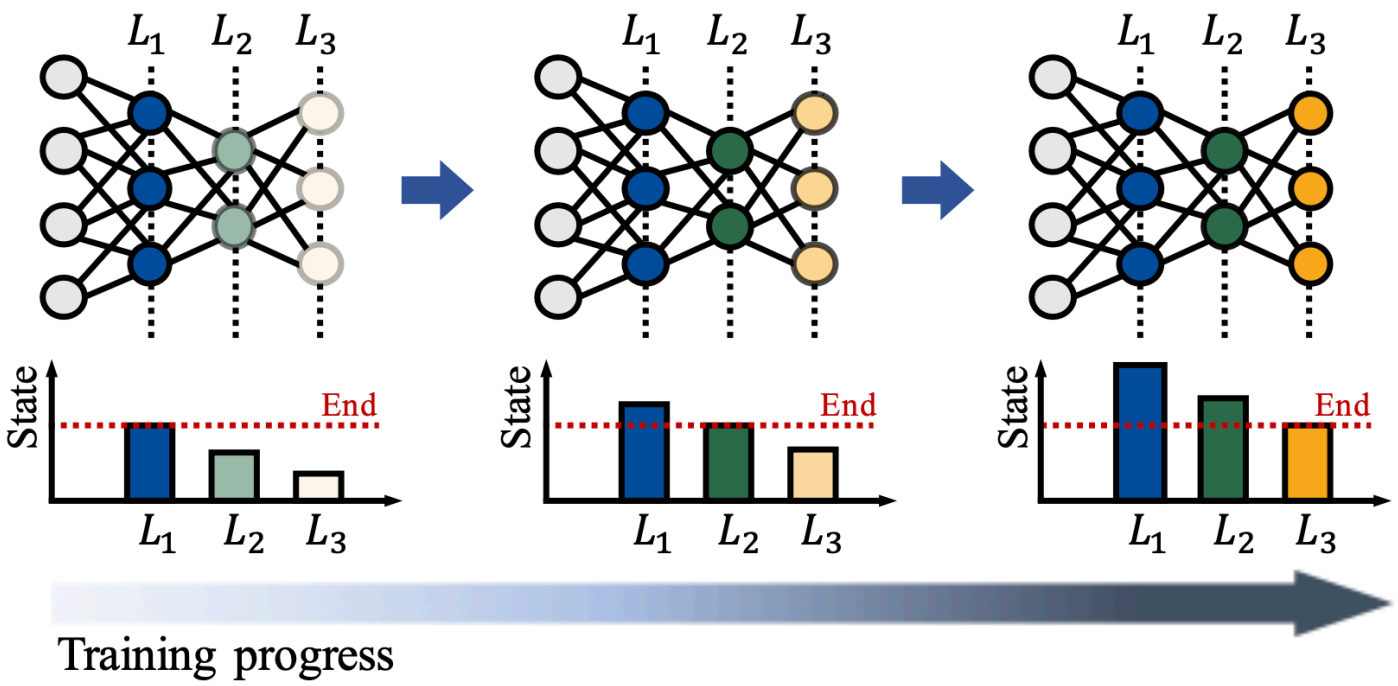
- The point at which each layer becomes stable can be different across layers
- Thus, ***the optimal endpoint of the warm-up may differ across layers***



Layer-wise state-aware warm-up

Limitation of the existing warm-up method

- It applies the warm-up period to all parameters in a DNN model
  - For layers with high gradient variance, the warm-up is likely to be applied unnecessarily long
  - For layers with low gradient variance, the warm-up likely to be end early



Layer-wise state-aware warm-up

Estimating the state of the training for each layer

- The state of the training for layer *k* at iteration *t* is defined as

$$S_{(t,k)} = \sum_{i=0}^{t-1} grad_var_{(t,k)}$$

  - $grad_var_{(t,k)}$: the gradient variance of layer *k* at iteration *t*

- Then, the learning rate for layer *k* at iteration *t* is as follow

$$\begin{cases} \eta_{(t,k)} \cdot \dfrac{S_{(t,k)}}{\theta}, & \text{if } S_{(t,k)} \leq \theta, \\ \eta_{(t,k)}, & \text{otherwise.} \end{cases}$$

  - θ: the threshold for determining the endpoint of the warm-up

Algorithm of LENA

Algorithm 1 Large-batch training with LENA

Function LargeBatchSGD_LENA($w_0, X, f, n, k, \eta_0, \theta$):
  for $t = 0, 1, \ldots$ **do**
    for $i = 0, \ldots, n-1$ **do** *# in n parallel workers*
      $b \leftarrow$ sample_batch(X)
      $g_t^{(i)} \leftarrow \frac{1}{|b|} \sum_{x \in b} \nabla f(w_t, x)$
    end for
    $\bar{g}_t \leftarrow \frac{1}{n} \sum_{i=1}^{n} g_t^{(i)}$
    for $j = 0, \ldots, k-1$ **do** *# Computing gradient variance*
      $grad_var_{(t,j)} \leftarrow \frac{\frac{1}{n} \sum_{i=0}^{n-1} \|g_{(t,j)}^{(i)}\|^2}{\|\bar{g}_{(t,j)}\|^2}$
    end for
    $\eta_t \leftarrow$ LENA_lr($grad_var, t, k, \eta_0, \theta$)
    $w_{t+1} \leftarrow w_t - \eta_t \odot \bar{g}_t$
  end for
  return w_t

Function LENA_lr($grad_var, t, k, \eta_0, \theta$):
  for $j = 0, \ldots, k-1$ **do** *# Layer-wise lr scaling*
    $\eta_{(t,j)} \leftarrow grad_var_{(t,j)} \cdot \eta_0$
    $S_{(t,j)} \leftarrow S_{(t,j)} + grad_var_{(t,j)}$
    if $S_{(t,j)} \leq \theta$ **then** *# State-aware warm-up*
      $\eta_{(t,j)} \leftarrow \eta_{(t,j)} \cdot \frac{S_{(t,j)}}{\theta}$
    end if
  end for
  return η_t

Table of Contents

- Background
- Motivation
- Proposed algorithm: LENA
  - Layer-wise LR scaling
  - Layer-wise state-ware warm-up
- Experiments
- Conclusions

Questions to Answer

Q1: Model accuracy

- Does LENA improve the model quality in large-batch training better than existing LR scaling methods?

Q2: Ablation study

- How effective are the layer-wise strategies of LENA in improving the model quality in large-batch training?

Q3: Hyperparameter sensitivity

- How sensitive is the model quality of LENA to the hyperparameter?

Experimental Setup

Models

- ResNet-18 (11M parameters), ResNet-50 (23M parameters)

Dataset

- CIFAR-10 (50K images with 10 labels)
- CIFAR-100 (50K images with 100 labels)

Competing algorithms

- Linear scaling with warm-up (LSW): a baseline
- AdaScale (AS): a state-of-the-art gradient variance based LR scaling method

Hyperparameters

- Batch sizes: 1024, 2048, … at most 16384
- Momentum SGD (momentum: 0.9)
- Weight decay: 0.0005, base learning rate η_0: 0.1

Q1: Model Accuracy

- **LENA always achieves the highest accuracy with the same amount of training data**

Dataset	n	Batch size	Valid. Acc.			Epochs			Iterations			Time (sec.)		
			LSW	AS	LENA	LSW	AS	LENA	LSW	AS	LENA	LSW	AS	LENA
CIFAR-10	8	1.02k	0.9494	0.9424	**0.9488**	104	109	**104**	4.58k	4.80k	**4.58k**	**579**	625	636
	16	2.05k	0.9444	0.9296	**0.9498**	109	117	**102**	2.40k	2.57k	**2.24k**	582	628	**566**
	32	4.10k	0.9186	0.9104	***0.9514***	124	126	***98***	1.36k	1.39k	**1.07k**	636	650	**515**
	64	8.19k	0.8224	0.7980	**0.9446**	183	195	**107**	1.01k	1.07k	**588**	923	987	**546**
	128	16.4k	0.3084	0.6856	**0.9058**	N/A	N/A	**128**	N/A	N/A	**352**	N/A	N/A	**645**
CIFAR-100	8	1.02k	0.9410	0.9406	***0.9454***	152	151	***141***	6.69k	6.64k	**6.20k**	1737	1794	**1692**
	16	2.05k	0.9352	0.9094	***0.9478***	145	171	***133***	3.19k	3.76k	**2.92k**	1534	1881	**1477**
	32	4.10k	0.9334	0.8856	***0.9434***	167	183	***141***	1.83k	2.01k	**1.55k**	1704	1941	**1510**
	64	8.19k	0.8428	0.1664	***0.9450***	183	N/A	***141***	1.00k	N/A	**775**	1823	N/A	**1475**
	128	16.4k	0.1650	0.0914	**0.9352**	N/A	N/A	**174**	N/A	N/A	**478**	N/A	N/A	**1779**

Q1: Model Accuracy

- **LENA achieves the target accuracy within *the fewest iterations***
  - *Within up to 45.2% fewer iterations and 44.7% shorter time*

Dataset	n	Batch size	Valid. Acc.			Epochs			Iterations			Time (sec.)		
			LSW	AS	LENA	LSW	AS	LENA	LSW	AS	LENA	LSW	AS	LENA
CIFAR-10	8	1.02k	0.9494	0.9424	**0.9488**	104	109	**104**	4.58k	4.80k	**4.58k**	**579**	625	636
	16	2.05k	0.9444	0.9296	**0.9498**	109	117	**102**	2.40k	2.57k	**2.24k**	582	628	**566**
	32	4.10k	0.9186	0.9104	***0.9514***	124	126	***98***	1.36k	1.39k	**1.07k**	636	650	**515**
	64	8.19k	0.8224	0.7980	**0.9446**	183	195	**107**	1.01k	1.07k	**588**	923	987	**546**
	128	16.4k	0.3084	0.6856	**0.9058**	N/A	N/A	**128**	N/A	N/A	**352**	N/A	N/A	**645**
CIFAR-100	8	1.02k	0.9410	0.9406	***0.9454***	152	151	***141***	6.69k	6.64k	**6.20k**	1737	1794	**1692**
	16	2.05k	0.9352	0.9094	***0.9478***	145	171	***133***	3.19k	3.76k	**2.92k**	1534	1881	**1477**
	32	4.10k	0.9334	0.8856	***0.9434***	167	183	***141***	1.83k	2.01k	**1.55k**	1704	1941	**1510**
	64	8.19k	0.8428	0.1664	***0.9450***	183	N/A	***141***	1.00k	N/A	**775**	1823	N/A	**1475**
	128	16.4k	0.1650	0.0914	**0.9352**	N/A	N/A	**174**	N/A	N/A	**478**	N/A	N/A	**1779**

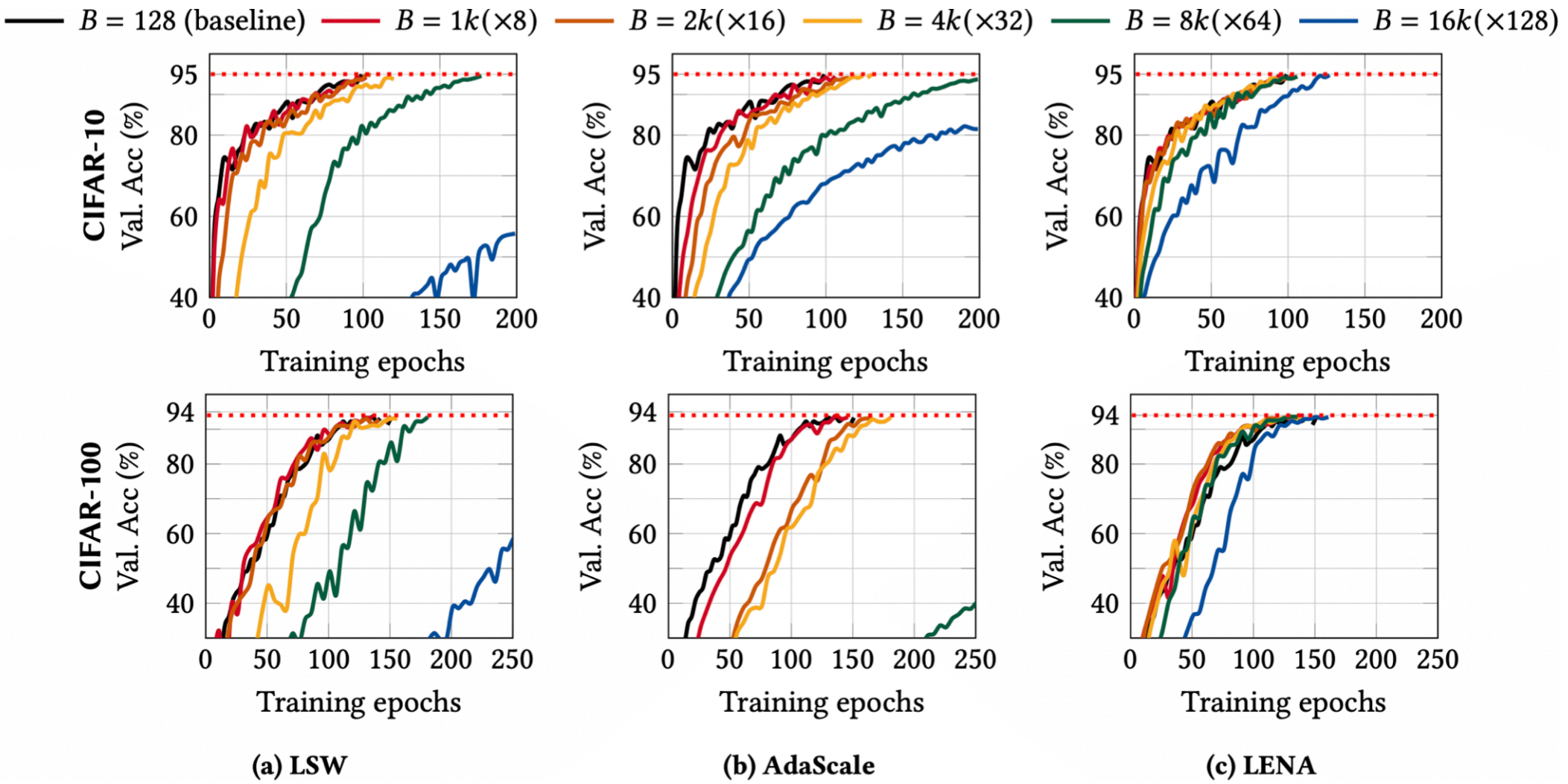
Q1: Model Accuracy

- **LENA achieves the target accuracy with *very large-batch training***
  - *Surpassing the limits of existing lr scaling methods* (e.g., **8.19k** and **16.4k**)

Dataset	n	Batch size	Valid. Acc.			Epochs			Iterations			Time (sec.)		
			LSW	AS	LENA	LSW	AS	LENA	LSW	AS	LENA	LSW	AS	LENA
CIFAR-10	8	1.02k	0.9494	0.9424	**0.9488**	104	109	**104**	4.58k	4.80k	**4.58k**	**579**	625	636
	16	2.05k	0.9444	0.9296	**0.9498**	109	117	**102**	2.40k	2.57k	**2.24k**	582	628	**566**
	32	4.10k	0.9186	0.9104	***0.9514***	124	126	***98***	1.36k	1.39k	**1.07k**	636	650	**515**
	64	8.19k	0.8224	0.7980	**0.9446**	183	195	**107**	1.01k	1.07k	**588**	923	987	**546**
	128	16.4k	0.3084	0.6856	**0.9058**	N/A	N/A	**128**	N/A	N/A	**352**	N/A	N/A	**645**
CIFAR-100	8	1.02k	0.9410	0.9406	***0.9454***	152	151	***141***	6.69k	6.64k	**6.20k**	1737	1794	**1692**
	16	2.05k	0.9352	0.9094	***0.9478***	145	171	***133***	3.19k	3.76k	**2.92k**	1534	1881	**1477**
	32	4.10k	0.9334	0.8856	***0.9434***	167	183	***141***	1.83k	2.01k	**1.55k**	1704	1941	**1510**
	64	8.19k	0.8428	0.1664	***0.9450***	183	N/A	***141***	1.00k	N/A	**775**	1823	N/A	**1475**
	128	16.4k	0.1650	0.0914	**0.9352**	N/A	N/A	**174**	N/A	N/A	**478**	N/A	N/A	**1779**

Q1: Model Accuracy

☐ LENA outperforms the existing LR scaling methods in all cases



Q2: Ablation Study: Our Layer-Wise LR Scaling

- **LENA can achieve the highest accuracy when the same warm-up applied**
  - Verifying ***the effectiveness of our proposed layer-wise LR scaling***

Dataset	n	Batch size	Linear Scaling (LS)			AdaScale (AS)			LENA		
			No	***Fixed***	Ours	***No***	Fixed	Ours	No	Fixed	***Ours***
CIFAR-10	8	1.02k	0.9340	**0.9494**	0.9462	0.9424	**0.9478**	0.9438	0.9482	0.9466	**0.9488**
	16	2.05k	0.9326	0.9444	**0.9448**	0.9296	0.9436	**0.9486**	0.9456	0.9484	**0.9498**
	32	4.10k	0.8952	0.9186	**0.9364**	0.9104	0.9434	**0.9480**	0.9472	0.9488	**0.9514**
	64	8.19k	0.8414	0.8224	**0.8836**	0.7980	0.9024	**0.9376**	0.9340	0.9444	**0.9446**
	128	16.4k	0.1594	0.3084	**0.6364**	0.6856	0.7274	**0.8946**	0.8378	0.8854	**0.9058**
CIFAR-100	8	1.02k	0.9300	0.9410	**0.9450**	0.9406	**0.9478**	0.9446	0.9398	0.9454	**0.9454**
	16	2.05k	0.9218	0.9352	**0.9446**	0.9094	0.9434	**0.9438**	0.9442	0.9456	**0.9478**
	32	4.10k	0.9198	0.9334	**0.9352**	0.8856	0.9358	**0.9384**	0.9378	0.9410	**0.9434**
	64	8.19k	0.8578	0.8428	**0.8932**	0.1664	0.8560	**0.9016**	0.6210	0.9390	**0.9450**
	128	16.4k	0.1482	0.1650	**0.2842**	0.0914	0.0681	**0.4548**	0.4442	0.4946	**0.9352**

Q2: Ablation Study: Our Layer-Wise Warm-Up

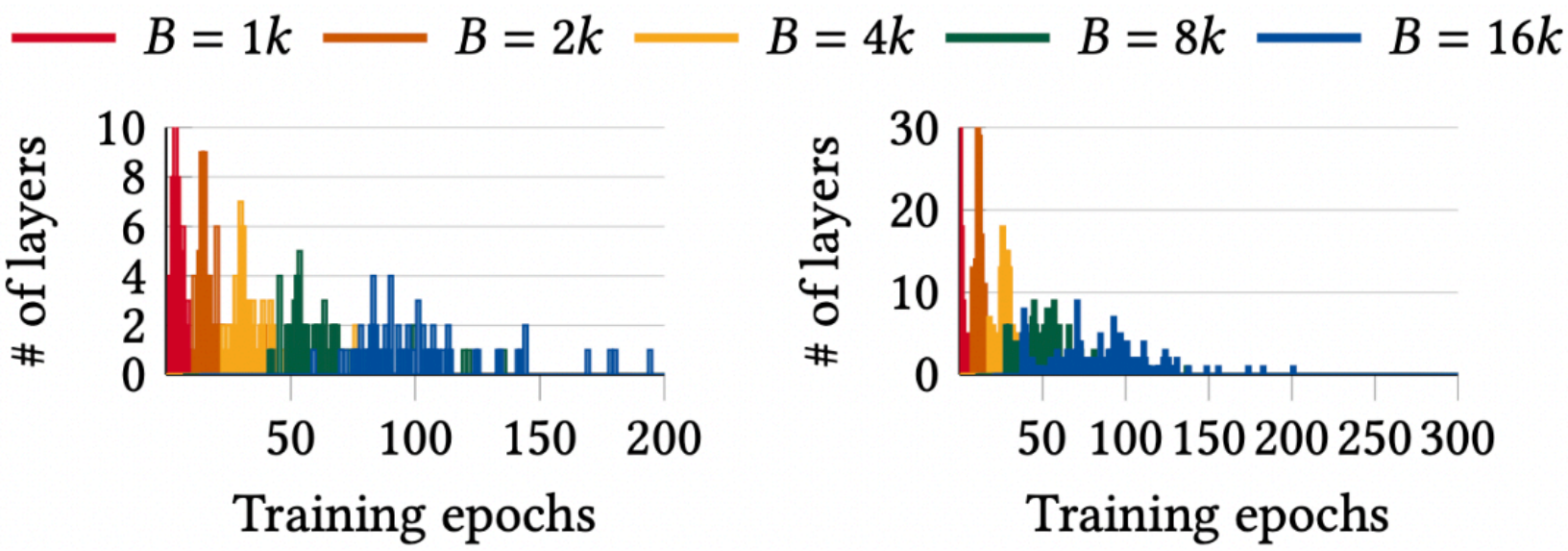
- ***Our state-aware warm-up*** **improves the model quality more than others**
  - Successfully addressing the issue that the model fluctuates steeply in the early stage of large-batch training

Dataset	n	Batch size	Linear Scaling (LS)			AdaScale (AS)			LENA		
			No	***Fixed***	Ours	***No***	Fixed	Ours	No	Fixed	***Ours***
CIFAR-10	8	1.02k	0.9340	**0.9494**	0.9462	0.9424	**0.9478**	0.9438	0.9482	0.9466	**0.9488**
	16	2.05k	0.9326	0.9444	**0.9448**	0.9296	0.9436	**0.9486**	0.9456	0.9484	**0.9498**
	32	4.10k	0.8952	0.9186	**0.9364**	0.9104	0.9434	**0.9480**	0.9472	0.9488	**0.9514**
	64	8.19k	0.8414	0.8224	**0.8836**	0.7980	0.9024	**0.9376**	0.9340	0.9444	**0.9446**
	128	16.4k	0.1594	0.3084	**0.6364**	0.6856	0.7274	**0.8946**	0.8378	0.8854	**0.9058**
CIFAR-100	8	1.02k	0.9300	0.9410	**0.9450**	0.9406	**0.9478**	0.9446	0.9398	0.9454	**0.9454**
	16	2.05k	0.9218	0.9352	**0.9446**	0.9094	0.9434	**0.9438**	0.9442	0.9456	**0.9478**
	32	4.10k	0.9198	0.9334	**0.9352**	0.8856	0.9358	**0.9384**	0.9378	0.9410	**0.9434**
	64	8.19k	0.8578	0.8428	**0.8932**	0.1664	0.8560	**0.9016**	0.6210	0.9390	**0.9450**
	128	16.4k	0.1482	0.1650	**0.2842**	0.0914	0.0681	**0.4548**	0.4442	0.4946	**0.9352**

Q2: Ablation Study: Optimal Warm-up End Point

The optimal warm-up end points differs across layers

- The warm-up tends to be applied for a longer period as the batch size increases



(a) CIFAR-10 (b) CIFAR-100

Q3: Hyperparameter sensitivity

☐ The model quality of LENA is *insensitive* to α

- LENA successfully achieves high quality of the model with a wide range of α

	n	B	Warm-up threshold α				
			1%	3%	**5%**	7%	9%
CIFAR-10	8	1k	0.9488	0.9418	0.9448	**0.9514**	0.9492
	16	2k	0.9420	0.9454	**0.9498**	0.9432	0.9413
	32	4k	0.9428	0.9508	**0.9514**	0.9504	0.9490
	64	8k	0.9432	0.9438	**0.9446**	0.9436	0.9434
	128	16k	0.8412	0.8932	**0.9058**	0.8928	0.8892
CIFAR-100	8	1k	0.9448	0.9458	0.9454	0.9444	**0.9456**
	16	2k	0.9380	0.9358	**0.9478**	0.9468	0.9446
	32	4k	0.9410	0.9426	**0.9434**	0.9412	0.9392
	64	8k	0.9402	0.9430	**0.9450**	0.9416	0.9396
	128	16k	0.6802	0.9336	**0.9352**	0.9290	0.9266

Conclusions

- **We identified the cause of the degradation in the model quality in existing LR scaling methods**
  - Overlooking the difference across layers in training of a DNN model.
- **We proposed a novel LR scaling method for large-batch training, LENA**
  - Effectively addressing the difference across layers with the layer-wise strategies
    - A layer-wise adaptive lr scaling
    - A layer-wise state-aware warm-up
- **Through comprehensive experiments, we showed that LENA successfully achieves the high quality of models**
  - Within fewest iterations in training (up to 45.2% fewer iterations)
  - With very large-batch sizes up to ×128 larger than the case of single-worker training.

Thank You !

Email: koyunyong@hanyang.ac.kr