GradInit: Learning to Initialize Neural Networks for Stable and Efficient Training

Доклад

Plan

1. What is a good initialization?
2. A general idea of GradInit
3. Details
4. Experiments with VGG/Res-Nets
5. Transformers w/o LN and warm-up

What is a good initialization?

1. prevents gradient explosion/vanishing

What is a good initialization?

1. prevents gradient explosion/vanishing

2. does not require tricks to train a network with different lrs (eg warm-up)

What is a good initialization?

1. prevents gradient explosion/vanishing
2. does not require tricks to train a network with different lrs (eg warm-up)
3. does not require hyperparameters tuning - always good

GradInit. General Idea

find the scales of the initialized weights matrices so that the loss after the first gradient step taken by a stochastic optimizer (SGD or Adam) is as low as possible

GradInit. Idea

Definitions:

1. scales: $\boldsymbol{m} = \{\alpha_1, \ldots, \alpha_M\}$
2. $\boldsymbol{\theta_m} = \{\alpha_1 \boldsymbol{W}_1, \ldots, \alpha_M \boldsymbol{W}_M\}$
3. $L(S; \boldsymbol{\theta}) = \frac{1}{|S|} \sum_{x \in S} \ell(x; \boldsymbol{\theta})$
4. $\boldsymbol{g}_{S,\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} L(S; \boldsymbol{\theta})$
5. A - optimization algorithm

GradInit. Idea

Definitions:

1. scales: $\boldsymbol{m} = \{\alpha_1, \ldots, \alpha_M\}$
2. $\boldsymbol{\theta_m} = \{\alpha_1 \boldsymbol{W}_1, \ldots, \alpha_M \boldsymbol{W}_M\}$
3. $L(S; \boldsymbol{\theta}) = \frac{1}{|S|} \sum_{x \in S} \ell(x; \boldsymbol{\theta})$
4. $\boldsymbol{g}_{S,\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} L(S; \boldsymbol{\theta})$
5. A - optimization algorithm

$$\begin{aligned} \underset{\boldsymbol{m}}{\text{minimize}} \quad & L(\tilde{S}; \boldsymbol{\theta_m} - \eta \mathcal{A}\left[\boldsymbol{g}_{S,\boldsymbol{\theta_m}}\right]), \\ \text{subject to} \quad & \|\boldsymbol{g}_{S,\boldsymbol{\theta_m}}\|_{p_{\mathcal{A}}} \leq \gamma, \end{aligned}$$

Questions?

What is A?

- Adam: $\text{sign}(\boldsymbol{g}_{S,\boldsymbol{\theta}_m})$
- SGD: $\gamma \boldsymbol{g}(S;\boldsymbol{\theta}_m)/\|\boldsymbol{g}(S;\boldsymbol{\theta}_m)\|_2$

Algorithm 1 *GradInit* for learning the initialization of neural networks.

Input: Target optimization algorithm $\mathcal{A}$ and learning rate η for model training, initial model parameters $\boldsymbol{\theta}_0$, learning rate τ of the GradInit scales $\boldsymbol{m}$, total iterations T, upper bound of the gradient γ, lower bound for the initialization scalars $\underline{\alpha} = 0.01$.
$\boldsymbol{m}_1 \leftarrow \mathbf{1}$
for $t = 1$ **to** T **do**
    Sample S_t from training set.
    $L_t \leftarrow \frac{1}{|S_t|} \sum_{x_k \in S_t} \ell(x_k; \boldsymbol{\theta}_{\boldsymbol{m}_t}),\ \boldsymbol{g}_t \leftarrow \nabla_{\boldsymbol{\theta}} L_t$
    if $\|\boldsymbol{g}_t\|_{p_{\mathcal{A}}} > \gamma$ **then**
        $\boldsymbol{m}_{t+1} \leftarrow \boldsymbol{m}_t - \tau \nabla_{\boldsymbol{m}_t} \|\boldsymbol{g}_t\|_{p_{\mathcal{A}}}$
    else
        Sample $\tilde{S}_t$ from training set.
        $\tilde{L}_{t+1} \leftarrow \frac{1}{|\tilde{S}_t|} \sum_{x_k \in \tilde{S}_t} \ell(x_k; \boldsymbol{\theta}_{\boldsymbol{m}_t} - \eta \mathcal{A}[\boldsymbol{g}_t])$
        $\boldsymbol{m}_{t+1} \leftarrow \boldsymbol{m}_t - \tau \nabla_{\boldsymbol{m}_t} \tilde{L}_{t+1}$
    Clamp $\boldsymbol{m}_{t+1}$ using $\underline{\alpha}$

Stochasticity of mini-batching

when the network has large initial gradient variance, the gradients on S and $\tilde{S}$ usually differ a lot, and for $\tilde{S}$ the gradient update step becomes more similar to adding random perturbations to the parameters

Table 1: Accuracies on CIFAR-10 using different overlapping ratios of $\tilde{S}$ and S for GradInit.

Model	$\frac{\lvert\tilde{S}\cap S\rvert}{\lvert S\rvert}$	Acc_1	Acc_{best}
VGG-19	0	21.9 ± 4.4	94.5 ± 0.1
w/o BN	0.5	$\mathbf{29.3 \pm 0.6}$	$\mathbf{94.7 \pm 0.02}$
(20.03 M)	1	28.7 ± 1.0	94.5 ± 0.1

Stochasticity of mini-batching

when the network has large initial gradient variance, the gradients on S and $\tilde{S}$ usually differ a lot, and for $\tilde{S}$ the gradient update step becomes more similar to adding random perturbations to the parameters

“Without excessive tuning, we find that we get

more reliable behavior for different architectures

when is a mixture of 50% samples from S

and 50% re-sampled training data”

Table 1: Accuracies on CIFAR-10 using different overlapping ratios of $\tilde{S}$ and S for GradInit.

Model	$\frac{\lvert\tilde{S}\cap S\rvert}{\lvert S\rvert}$	Acc_1	Acc_{best}
VGG-19	0	21.9 ± 4.4	94.5 ± 0.1
w/o BN	0.5	$\mathbf{29.3 \pm 0.6}$	$\mathbf{94.7 \pm 0.02}$
(20.03 M)	1	28.7 ± 1.0	94.5 ± 0.1

Hyperparameters (gradient norm and threshold)

$$L(S;\theta_{\boldsymbol{m}}-\eta\mathcal{A}[\boldsymbol{g}_{S,\theta_{\boldsymbol{m}}}])-L(S;\theta_{\boldsymbol{m}}) \approx -\eta\mathcal{A}[\boldsymbol{g}_{S,\theta_{\boldsymbol{m}}}]^T\boldsymbol{g}_{S,\theta_{\boldsymbol{m}}} = \begin{cases} -\eta\|\boldsymbol{g}_{S,\theta_{\boldsymbol{m}}}\|_2^2, & \text{if } \mathcal{A} \text{ is SGD,} \\ -\eta\|\boldsymbol{g}_{S,\theta_{\boldsymbol{m}}}\|_1, & \text{if } \mathcal{A} \text{ is Adam.} \end{cases} \quad (2)$$

To effectively bound the approximated change in Eq. 2, we choose $\ell_{p_{\mathcal{A}}}$ to be the ℓ_2 and ℓ_1 norm for SGD and Adam respectively, so when the constraint is satisifed, the maximum change in the loss, according to our local approximation, is $\eta\gamma^2$ for SGD and $\eta\gamma$ for Adam. We recommend setting γ such that $\eta\gamma^2 = 0.1$ for SGD and $\eta\gamma = 0.1$ for Adam. According to the linear approximations, this limits the gradient magnitude so that the first step of SGD can decrease the loss by at most 0.1. This simple rule was used across all vision and language experiments.

Why a constraint and not a penalty?

$$\underset{\boldsymbol{m}}{\text{minimize}} \quad L(\tilde{S}; \boldsymbol{\theta_m} - \eta \mathcal{A}[\boldsymbol{g}_{S,\boldsymbol{\theta_m}}]) + \lambda \|\boldsymbol{g}_{S;\boldsymbol{\theta_m}}\|_{p_{\mathcal{A}}}$$

1. Penalty requiaries second-order derivative
2. lambda is hard to choose

Experiments. CIFAR-10. Accuracy

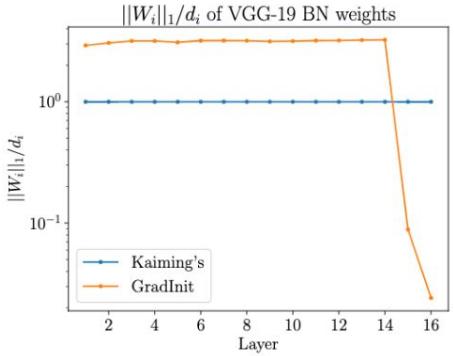
Table 3: First epoch (Acc_1) and best test accuracy over all epochs (Acc_{best}) for models on CIFAR-10. We report the mean and standard error of the test accuracies in 4 experiments with different random seeds. Best results in each group are in bold.

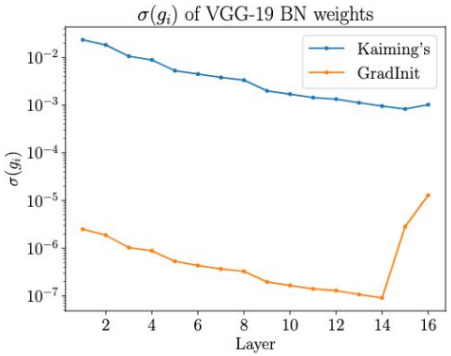
Model (# Params)		VGG-19 w/o BN (20.03M)	VGG-19 w/ BN (20.04M)	ResNet-110 w/o BN (1.72M)	ResNet-110 w/ BN (1.73M)	ResNet-1202 w/ BN (19.42M)
Kaiming	Acc_1	29.1 ± 1.5	12.6 ± 0.6	16.1 ± 2.1	23.2 ± 0.9	12.9 ± 2.8
	Acc_{best}	94.5 ± 0.1	94.4 ± 0.1	94.2 ± 0.1	95.0 ± 0.2	94.4 ± 0.6
+1 epoch (Const. LR)	Acc_1	37.2 ± 1.1	19.6 ± 4.0	21.0 ± 3.8	32.5 ± 3.8	12.6 ± 2.8
	Acc_{best}	94.4 ± 0.1	94.5 ± 0.1	93.9 ± 0.4	94.7 ± 0.3	94.0 ± 0.4
+1 epoch (Warmup)	Acc_1	37.4 ± 1.2	53.5 ± 2.9	19.8 ± 0.5	48.7 ± 1.1	28.1 ± 1.3
	Acc_{best}	94.4 ± 0.1	94.7 ± 0.1	94.1 ± 0.1	95.1 ± 0.1	95.4 ± 0.2
MetaInit	Acc_1	30.5 ± 0.9	35.1 ± 0.6	14.6 ± 2.2	29.0 ± 1.5	11.7 ± 1.6
	Acc_{best}	94.6 ± 0.1	94.6 ± 0.1	94.2 ± 0.1	94.8 ± 0.1	95.0 ± 0.5
GradInit	Acc_1	29.3 ± 0.6	47.8 ± 1.8	36.2 ± 0.8	38.2 ± 0.9	29.0 ± 1.1
	Acc_{best}	**94.7** ± 0.1	**95.1** ± 0.1	**94.6** ± 0.1	**95.4** ± 0.1	**96.2** ± 0.1

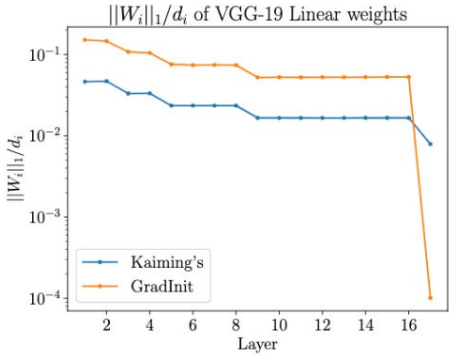
Table 6: Acc_1/Acc_{best} of ResNet-50 models on ImageNet. Result of MetaInit comes from Dauphin and Schoenholz [16] and we reimplemented the rest.

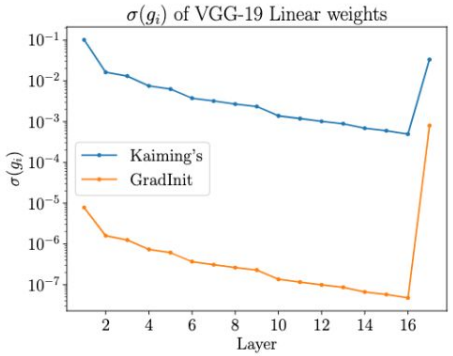
	Kaiming	FixUp	MetaInit	GradInit
w/ BN	14.6/75.9	-	-	19.2/76.2
w/o BN	-	18.0/75.7	-/75.4	19.2/75.8

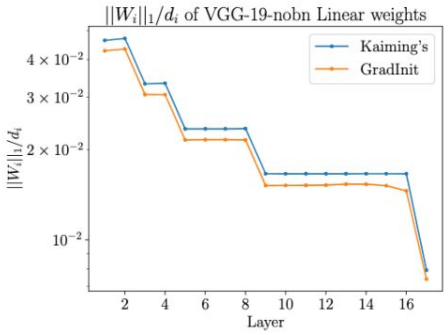
Experiments. CIFAR-10. VGG

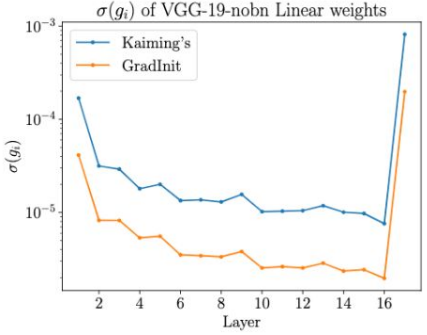


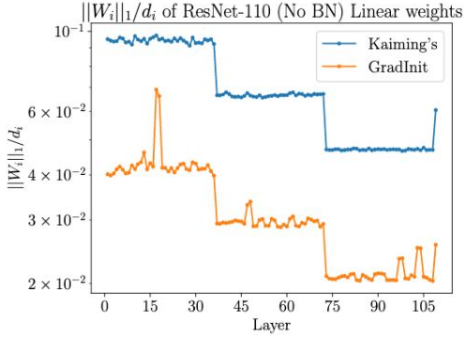


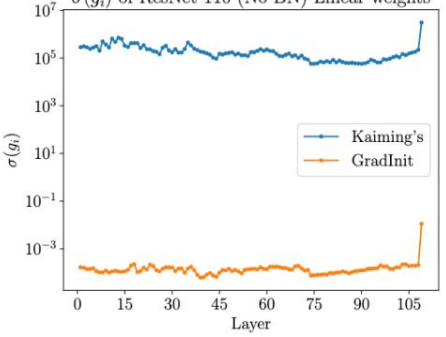












Experiments. The importance of rescaling BN layers

Table 4: Comparing the results of GradInit with fixed BN scale parameters (Fix BN) and only rescale the BN parameters (Only BN).

	Kaiming		GradInit		GradInit (Fix BN)		GradInit (Only BN)	
Model	Acc_0	Acc_{best}	Acc_0	Acc_{best}	Acc_0	Acc_{best}	Acc_0	Acc_{best}
VGG-19 (w/ BN)	12.6 ± 0.6	94.4 ± 0.1	47.8 ± 1.8	95.1 ± 0.1	13.1 ± 0.9	94.6 ± 0.1	14.4 ± 2.1	94.4 ± 0.1
ResNet-110 (w/ BN)	23.2 ± 0.9	95.0 ± 0.2	38.2 ± 0.9	95.4 ± 0.1	24.7 ± 3.1	94.7 ± 0.3	25.4 ± 3.1	94.6 ± 0.3

Experiments. Learnable scales while training

Table 5: Comparing the results with multiplying each weight matrix with a learnable scaler (Learning Scalars) on CIFAR10. The VGG-19 model is not able to converge unless we reduce the initial learning rate to 0.01, which obtained worse final accuracy. The ResNet-110 model's Acc_0 was 10% for 2 of the 4 runs.

Model	Learning Scalars		GradInit	
	Acc_0	Acc_{best}	Acc_0	Acc_{best}
VGG-19 (w/ BN, lr=0.1)	10.0 ± 0.0	10.0 ± 0.0	47.8 ± 1.8	95.1 ± 0.1
VGG-19 (w/ BN, lr=0.01)	50.6 ± 0.8	93.4 ± 0.1	-	-
ResNet-110 (w/ BN)	21.5 ± 6.9	94.7 ± 0.1	38.2 ± 0.9	95.4 ± 0.1

Experiments. Transformers. Comparison

Table 7: A comparison of GradInit with with the results from the papers (top 4 rows), and our reimplementation of Admin for training the Post-LN Transformer model on the IWSLT-14 De-EN dataset. “Standard" refers to training with standard initialization and warmup.

Method	Remove LN	w_{skip}	Warmup	Optimizer	BLEU
Standard [6]			✓	RAdam	35.6
FixUp [9]	✓		✓	Adam	34.5
T-FixUp [5]	✓			Adam	35.5
Admin [6]		✓		RAdam	35.7
Admin		✓		Adam	36.1
Admin		✓		SGD	33.7
GradInit		✓		Adam	36.0
GradInit				Adam	36.1
GradInit				SGD	35.6

Experiments. Transformers. Stability

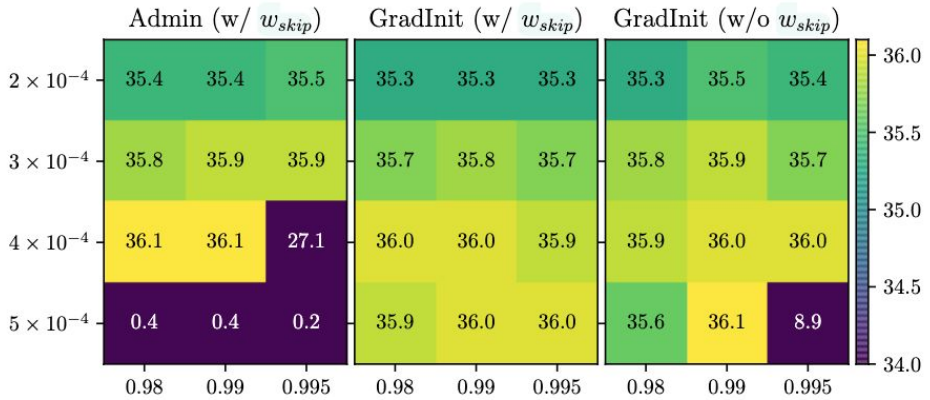


Figure 3: BLEU scores for the Post-LN Transformer without learning rate warmup using Adam on IWSLT-14 DE-EN under different learning rates $\eta_{\max}$ (y axis) and β_2 (x axis). Each result is averaged over 4 experiments.

Рецензия

Достоинства статьи

- Предложенный метод описан подробно и чётко
- Ясно описаны потенциальные преимущества
- Итоговое качество превосходит аналоги
- Достаточно обширные эксперименты для изображений
- Методология описана достаточно подробно

Недостатки статьи

- Результаты экспериментов очень тяжело сравнивать между собой
- Интерпретация результатов экспериментов не очень ясна
- Эксперименты с текстовыми данными не столь обширные
- Отсутствуют конкретные цифры для сравнения с MetaInit с точки зрения вычислительных ресурсов

Рецензии на OpenReview

- NIPS 2021 (Poster)
- Ratings: 6, 6, 7, 7; Overall: 7 (Good submission, accept)
- Также отмечают непоследовательность в визуальном представлении экспериментов
- Финальная версия статьи учитывает предыдущие замечания: добавлены более подробные графики, исправлены найденные опечатки

Практик-исследователь

Публикация

- Первая версия – февраль 2021
- Опубликована на NIPS 2021 (постер)
- 5 цитирований

Авторы

- Chen Zhu, Renkun Ni – PhD @ Maryland
- Tom Goldstein – Prof @ Maryland
- Zheng Xu, W. Ronny Huang – Google Research

Контекст

- Нет явных предпосылок или предшественников, самое близкое – MetaInit
- Интересное цитирование – A Loss Curvature Perspective on Training Instability in Deep Learning

Model	Dataset	Method	*Acc*
WideResnet 28-10 (w/o BN)	CIFAR-10	Warmup 1000	97.2
WideResnet 28-10 (w/o BN)	CIFAR-10	MetaInit	97.1
Resnet-50 (w/o BN)	ImageNet	Fixup	76.0
Resnet-50 (w/o BN)	ImageNet	MetaInit	76.0
Resnet-50 (w/o BN)	ImageNet	GradInit	76.2
Resnet-50 (w/o BN)	ImageNet	Warmup 1000	76.2
Transformer 6L (w/o LN)	WMT	Warmup + Clip + .25x Init	27.10 (BLEU)
Transformer 6L (w/ LN)	WMT	LayerNorm	27.01 (BLEU)