ReZero is All You Need: Fast Convergence at Large Depth[1]

July 15, 2020

[1] *ReZero is All You Need: Fast Convergence at Large Depth* by Bachlechner et al., 2020

Overview

- Initialization and stability is still an issue for many problems
- Authors propose a simple trick, similar to residual connections:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i F(\boldsymbol{x}_i) \tag{1}$$

  where α_i is learnable and initialized at 0.
- It has the following benefits:
  - Simplicity and wide applicability
  - Faster convergence
  - It allows training of deeper models
- Authors test their approach on
  - Language modelling with Transformer
  - Classification on CIFAR-10
- They show good performance in terms of fast convergence and stability

Residual with zero init (ReZero)

- Dynamical Isometry is a property that all singular values of the input-output Jacobian are close to 1
- It allows to train models much faster and make them much deeper
- Authors propose an easy trick that makes a model satisfy it (at init):

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i F(\boldsymbol{x}_i) \tag{2}$$

where α_i is learnable and initialized at 0.

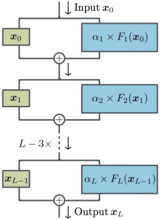


Figure 1: ReZero

- Experiments show that this property remains approximately preserved later on in training as well

Fully-Connected models on CIFAR-10

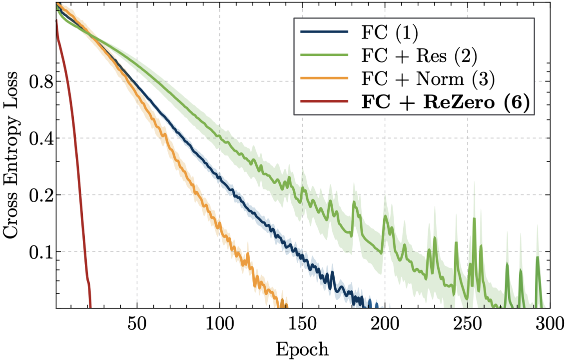


Figure: Convergence speed for different normalization strategies

Convolutional models on CIFAR-10

Model	Val. Error [%]	Change	Epochs to 80% Acc.	Train Loss $\times 1000$
ResNet-56 [2]	6.27 ± 0.06	–	20 ± 1	5.9 ± 0.1
+ Gated ResNet [7, 29]	6.80 ± 0.09	$+ 0.53$	9 ± 2	4.6 ± 0.3
+ zero γ [23, 24]	7.84 ± 0.05	$+ 1.57$	39 ± 4	31.2 ± 0.5
+ FixUp [10]	7.26 ± 0.10	$+ 0.99$	13 ± 1	4.6 ± 0.2
+ **ReZero**	6.58 ± 0.07	$+ 0.31$	15 ± 2	4.5 ± 0.3
ResNet-110 [2]	6.24 ± 0.29	–	23 ± 4	4.0 ± 0.1
+ Gated ResNet [7, 29]	6.71 ± 0.05	$+ 0.47$	10 ± 2	2.8 ± 0.2
+ zero γ [23, 24]	7.49 ± 0.07	$+ 1.25$	36 ± 5	18.5 ± 0.9
+ FixUp [10]	7.10 ± 0.22	$+ 0.86$	15 ± 1	3.3 ± 0.5
+ **ReZero**	5.93 ± 0.12	$- 0.31$	14 ± 1	2.6 ± 0.1
Pre-activation ResNet-18 [22]	6.38 ± 0.01	–	26 ± 2	4.1 ± 0.3
+ **ReZero**	5.43 ± 0.06	$- 0.95$	12 ± 1	1.9 ± 0.3
Pre-activation ResNet-50 [22]	5.37 ± 0.02	–	26 ± 3	2.6 ± 0.1
+ **ReZero**	4.80 ± 0.08	$- 0.57$	17 ± 1	2.2 ± 0.1

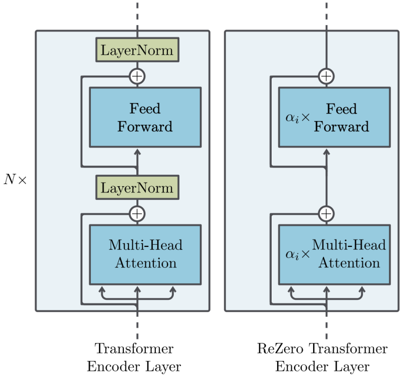
ReZero Transformer

Vanilla Transformer uses Post-Norm normalization:

$$\boldsymbol{x}_{i+1} = \text{LayerNorm}\left(\boldsymbol{x}_i + \text{sublayer}\left(\boldsymbol{x}_i\right)\right) \tag{3}$$

Authors replaced this with:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \alpha_i \, \text{sublayer}\left(\boldsymbol{x}_i\right) \tag{4}$$



Language Modeling results

Table 3: Comparison of various 12 layer Transformers normalization variants against ReZero and the training iterations required to reach 1.2 BPB on `enwiki8` validation set.

Model	Iterations	Speedup
Post-Norm [27]	Diverged	-
+ Warm-up	13,690	1×
Pre-Norm	17,765	0.77×
GPT2-Norm [4]	21,187	0.65×
ReZero $\alpha = 1$	14,506	0.94×
ReZero $\boldsymbol{\alpha = 0}$	**8,800**	**1.56**×

Table 4: Comparison of Transformers (TX) on the `enwiki8` test set. Char-TX refers to the Character Transformer [14] and uses additional auxiliary losses to achieve its performance.

Model	Layers	Parameters	BPB
Char-TX [14]	12	41M	1.11
TX + Warm-up	12	38M	1.17
TX + ReZero $\alpha = 1$	12	34M	1.17
TX + ReZero $\alpha = 0$	12	34M	1.17
Char-TX [14]	64	219M	1.06
TX	64	51M	Diverged
TX + Warm-up	64	51M	Diverged
TX + ReZero $\alpha = 1$	64	51M	Diverged
TX + ReZero $\alpha = 0$	64	51M	1.11
TX + ReZero	128	101M	1.08

Model preserves dynamic isometry by itself

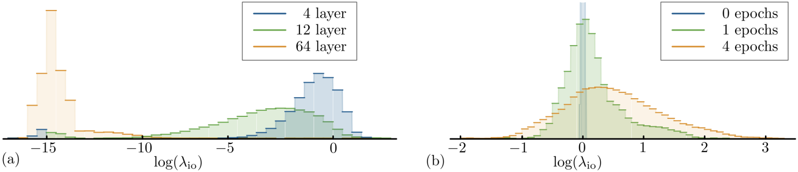


Figure: Histograms of $\log(\sigma)$ of singular values. Left: traditional Transformer. Right: 64-layer ReZero Transformer

Residual weights evolution

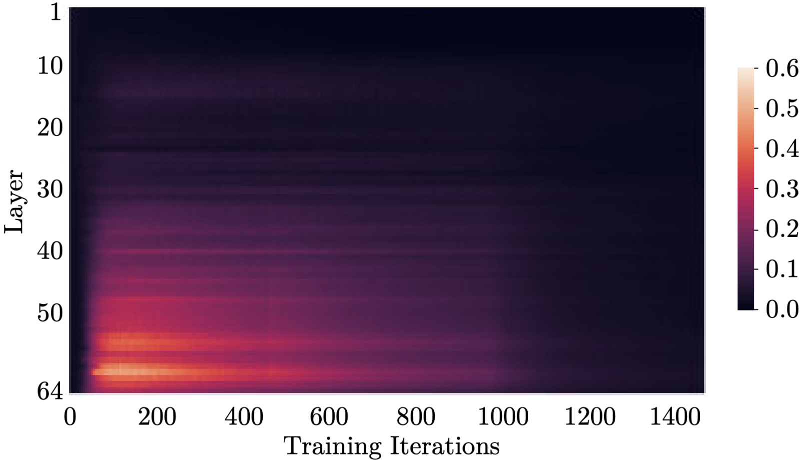


Figure: Evolution of α_i for 64-layer Transformer

- Model first increases α_i for later layers, then decreases them all.
- Authors say that there is a similar pattern for $\alpha = 1$ (for a 12-layer transformer): model first tries to reduce α. But instead of increasing