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

Докладчик: Еленик Константин

Рецензент: Ким Михаил

Практик-исследователь: Михненко Наталья

Хакер: Сафонов Иван

https://arxiv.org/pdf/2102.08098.pdf

Abstract

GradInit – метод автоматической инициализации нейронных сетей, работающий для любых архитектур.

Метод основан на эвристике: норма каждого слоя сети корректируется так, чтобы один шаг оптимизатора давал минимальное значение лосса. Корректировка достигается за счет умножения каждой группы параметров сети на обучаемый скаляр.

Авторы показали, что метод ускоряет сходимость и улучшает качество на тесте для различных конволюционных сетей, а также позволяет обучить оригинальный трансформер без warmup.

Общая схема

1. Все матрицы весов* изначально инициализируются из нормального распределения с нулевым средним.
2. Полученные матрицы фиксируются. Для каждой из них обучается scale factor.
3. Каждая из матриц масштабируется
4. Модель обучается как обычно (без scale factors)

$$\{W_1, \ldots, W_m\}$$

$$\{\alpha_1 W_1, \ldots, \alpha_m W_m\}$$

$$W_i' \leftarrow \alpha_i W_i$$

$$\{W_1', \ldots, W_m'\}$$

*кроме scales и biases для нормирующих слоев, которые инициализируются 1 и 0 соответственно.

Задача оптимизации

Хотим чтобы один шаг оптимизатора давал минимальное значение лосса

параметры модели после одного шага оптимизатора (A)

$$\begin{cases} L(\hat{S}; \theta_m - \eta \mathcal{A}[g_{S,\theta_m}]) \to \min_m \\ \|g_{S,\theta_m}\|_{p_{\mathcal{A}}} \leq \gamma \end{cases}$$

не даем градиенту стать слишком большим

где:

$m = \{\alpha_1, \ldots, \alpha_M\}$

$\theta_m = \{\alpha_1 W_1, \ldots, \alpha_M W_M\}$

$g_{S,\theta} = \nabla_\theta L(S, \theta)$

$S, \hat{S}$ – два разных мини-батча

η, γ – константные гиперпараметры

$\mathcal{A}$ – алгоритм оптимизации (Adam/SGD)

Алгоритм оптимизации

(По сути просто SGD)

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 $\underline{\alpha}$

оптимизируем либо норму градиента

$$\begin{cases} L(\hat{S}; \theta_m - \eta \mathcal{A}[g_{S,\theta_m}]) \to \min_m \\ \|g_{S,\theta_m}\|_{p_{\mathcal{A}}} \le \gamma \end{cases}$$

считаем константой

либо значение лосса после одного шага

не даем альфам стать слишком маленькими

Выбор батчей

$$\begin{cases} L(\hat{S}; \theta_m - \eta \mathcal{A}[g_{S,\theta_m}]) \to \min_m \\ \|g_{S,\theta_m}\|_{p_{\mathcal{A}}} \leq \gamma \end{cases}$$

- Значение градиента может сильно зависеть от батча в начале обучение, так что сэмплировать S и Ŝ независимо плохо.
- Если S = Ŝ, то выгодно просто увеличить норму градиента.
- Поэтому авторы формируют новый батч (Ŝ) половину сэпмплируя из старого батча (S), а половину из остальных данных.

Выбор констант

$$\begin{cases} L(\hat{S}; \theta_m - \eta \mathcal{A}[g_{S,\theta_m}]) \to \min_m \\ \|g_{S,\theta_m}\|_{p_{\mathcal{A}}} \leq \gamma \end{cases}$$

Ограничение нужно, чтобы уменьшение лосса достигалось за счет выбора правильного направления, а не за счет большого шага в менее оптимальном направление.

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

Рекомендуется выбирать гамму из такого соотношения:

$\eta\gamma = 0.1$ для Adam

$\eta\gamma^2 = 0.1$ для SGD

Почему не …?

$$L(\hat{S}; \theta_m - \eta \mathcal{A}[g_{S,\theta_m}]) + \lambda \|g_{S,\theta_m}\| \to \min_m$$

- градиенты второго порядка
- сложно выбрать универсальную лямбду

Эксперименты

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}	$\mathbf{94.7} \pm 0.1$	$\mathbf{95.1} \pm 0.1$	$\mathbf{94.6} \pm 0.1$	$\mathbf{95.4} \pm 0.1$	$\mathbf{96.2} \pm 0.1$

Стабилизация

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}	$\mathbf{94.7} \pm 0.1$	$\mathbf{95.1} \pm 0.1$	$\mathbf{94.6} \pm 0.1$	$\mathbf{95.4} \pm 0.1$	$\mathbf{96.2} \pm 0.1$

Deep residual networks

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}	$\mathbf{94.7} \pm 0.1$	$\mathbf{95.1} \pm 0.1$	$\mathbf{94.6} \pm 0.1$	$\mathbf{95.4} \pm 0.1$	$\mathbf{96.2} \pm 0.1$

Что-то еще

- рескалирование весов для слоев перед батчнормом влияет на эффективность обучения
- просто добавить scale factors и обучать сеть не работает
- авторы показывают применимость своего метода к трансформерам

Итоги

GradInit – метод автоматической инициализации нейронных сетей, работающий для любых архитектур.

С помощью модифицированного SGD, метод находит scale factors для каждого блока параметров сети так, чтобы один шаг оптимизатора давал минимальное значение лосса.

Авторы показали, что метод позволяет уменьшить дисперсию градиента, ускоряет сходимость и улучшает качество на тесте для различных конволюционных сетей, а также дает возможность обучить оригинальный трансформер без warmup и даже с использование SGD.

Рецензия

В статье предложен автоматический метод инициализации весов, метод учитывает специфику оптимизатора, величину шага и данные, при этом является model-agnostic.

Сильные стороны:

1. Предложен эффективный алгоритм решения оптимизационной задачи

2. Предложен автоматический метод подбора гиперпараметров

3. С помощью большого количества экспериментов было показано, что:

- Метод уменьшает дисперсию градиентов
- Ускоряет сходимость
- Позволяет избавиться от warmup schedule, normalization layers

Рецензия

Слабые стороны:

1. Слабо исследованы случаи, когда метод неэффективен

2. Эксперименты покрывают небольшой набор задач: классификация и машинный перевод

3. Непонятна целесообразность в случае трансформеров

4. Не исследована применимость для GAN, Inverse RL и других моделей, в которых стабильность обучения критична

Рецензия

Актуальность:

1. Модели усложняются, процедуры их обучения тоже

2. Большинство предшествующих методов не учитывают специфику задачи и(или) специфику оптимизатора

3. Вычислительно эффективен

Рецензия

1. Идеи в статье изложены доходчиво, однако некоторые графики довольно сложно разглядеть

2. Код метода доступен

3. В статье перечислены гиперпараметры, необходимые для воспроизведения экспериментов.

Оценка: 8

Уверенность: 5

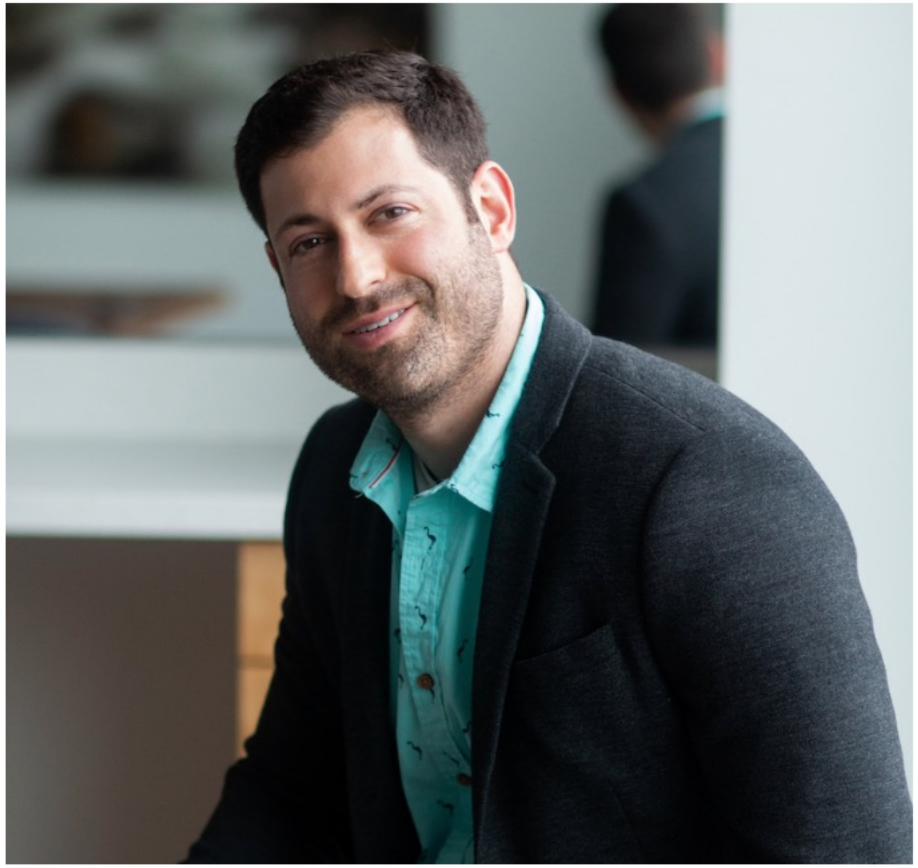
GradInit

Практик-исследователь: Михненко Наталья

Информация о статье

- Написана в феврале 2021
- Представлена на конференции NeurLPS (Neural Information Processing Systems)
- Авторы: Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W. Ronny Huang, Tom Goldstein

Известные авторы



Томас Голдштейн

- Исследования лежат на стыке машинного обучения и оптимизации и нацелены на приложения в области компьютерного зрения и обработки сигналов.
- В числе достижений – в 2012 году получил премию Ричарда ДиПрима, Young Faculty Award (от управление перспективных исследовательских проектов Министерства обороны США) и стипендию Слоана

Конкуренты

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}	$\mathbf{94.7} \pm 0.1$	$\mathbf{95.1} \pm 0.1$	$\mathbf{94.6} \pm 0.1$	$\mathbf{95.4} \pm 0.1$	$\mathbf{96.2} \pm 0.1$

Вклад

- Цитат работы очень мало
- Возможно дальнейшее исследование, может ли алгоритм не зависеть от оптимизатора
- Возможность использования в архитектурах, которые раньше не сходились