On the role of learning rate warmup and Radam

Abstract

Over the recent years, the convergence rate and performance of deep learning models have been significantly improved by strategically tuning the learning rate for adaptative stochastic gradient-based optimization. In particular, one techinque called learning rate warmup has empirically shown great success. Based on a recent paper published in August 2019 [1], the goal of this report is to provide a theoretical explanation describing the role of warmup and introduce a new variant of Adam called Radam which automatically incorporates its effect. Finally, I will empirically study the performance of Radam optimizer.

1 Introduction

The litterature on optimization methods for training neural networks has been extensively vast over the recent decades and some optimizers have witnessed remarkable success. The most commonly used first-order optimization methods are mainly based on gradient descent in which the parameters are updated in the opposite direction of the gradient of the loss function. A range of optimizers using adaptative learning rate and based on gradient descent recently came up and speeded up the convergence rate. AdaGrad [2] greatly improved the robustness of gradient descent especially when gradients are sparse but suffers from a rapid decay of the learning rate in case of dense gradients settings. Then AdaDelta [3] RMSProp was developed and dealt with this issue. Finally, one of the most recent one called Adam [4] inspired from Adagrad and RMSProp, uses estimations of first and second moments of gradient to adjust the learning rate.

However, it has been empirically shown that these algorithms still fail to generalize and converge in some settings. Thus some heuristic methods have been proposed and successfully deal with these issues [5] . One method called warm up heuristic involves using a small learning rate during the first epochs of the training and lead to significant improvements in maintaining early-stage training stability especially in large-scale settings.

As manually tuning a warmup scheduler is a long and tenuous work, the aim of this report is to give a theoretical justification for warm up heuristic and introduce a variant of Adam that produces the same effect without need for tuning .

2 Warmup theory

2.1 Related work

The warmup learning rate heuristic was first introduced in 2017 [6] and is originally proposed to handle gradient variance for SGD when dealing with large batch sizes. Indeed, researchers explained how they obtained no loss of accuracy when training the ImageNet dataset on ResNet-50 with large mini batches of 8192 images by applying an empirical linear scaling rule along with warmup technique.

The empirical linear scaling rule emphasizes the importance of adjusting the learning rate depending on the batch size: when the latter is multiplied by k, the learning rate should be multiplied by k. However this rule breaks down when the training error is changing rapidly and it happens in the early stages of training as weights can change wildly and the training error can spike.

This issue is solved by using the warmup strategy which makes sure that the network trains correctly

in the first training epochs, by adjusting a progressive learning rate.

This technique was successfully adopted on other types of neural networks architecture including transformers [7],[8], or natural language processing [9]. For instance, the image below shows that the algorithm fails to learn when the warmup step is too low at 12k (this is the number of steps at the beginning where warmup technique is applied).

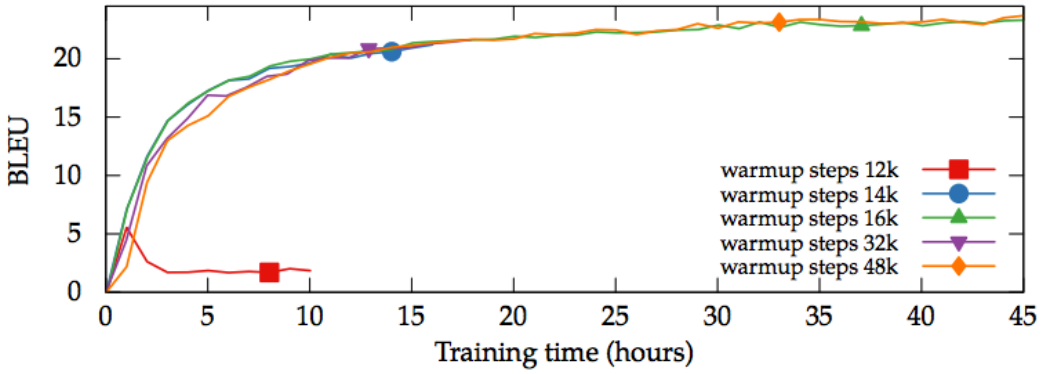


Figure 1: Effect of the warmup steps on the train loss (denoted by BLEU), trained with a transformer on the dataset CzEng 1.0 with batch size 1500 and learning rate 0.2. Image taken from the paper by M.Popel, "Training tips for the Transformer Model"

So far, the litterature on warmup heuristic contains only empirical analysis noticing the benefits of using warmup heuristic. A recent paper released in August 2019 [1] came up with theoretical notions justifying why warmup technique is beneficial on Adam optimizer even though this theory can be extended to other methods using similar adaptive learning rate such as RMSProp or Nadam.

2.2 Theoretical explanation

The paper [1] showed that warmup is a technique used as a way to counteract too large variance of the adaptive learning rate at the beginning of the training due to a low amount of samples and prevent the algorithm from being stuck in a bad local optima. In the following subsections, we will prove that.

2.2.1 Preliminaries

Firstly, let's define a generic algorithm describing adaptive learning rate optimization methods, introduced in [10]:

Algorithm 1 Generic Adaptive Optimization Method Setup

Input: $\{\alpha_t\}_{t=1}^{T}$ step size, $\{\phi(t), \psi(t)\}_{t=1}^{T}$: momentum and adaptative learning rate
θ_0 initial parameter and $f(\theta)$ the objective function
Output: θ_T
for $t = 1$ *to* T **do**
  $g_t \leftarrow \triangle_\theta f_t(\theta_{t-1})$
  $m_t \leftarrow \phi_t(g_1, g_2, ..., g_{t-1})$
  $v_t \leftarrow \psi_t(g_1, g_2, ..., g_{t-1})$
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t m_t v_t$
end

In the case of the Adam optimizer, here is the value of ϕ and ψ :

$$\phi(g_1, g_2, ..., g_t) = \frac{(1-\beta_1)\sum_{i=1}^{t}\beta_1^{t-i} g_t}{1-\beta_1^t} \quad \text{and} \quad \psi(g_1, g_2, ..., g_t) = \sqrt{\frac{1-\beta_2^t}{(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i} g_t^2}}$$

2.2.2 Analysis of the adaptive learning rate variance for the first step

Let's analyse the case when $t = 1$, we find that the adaptative learning rate is, according to the formula above: $\psi(g_1) = \sqrt{\frac{1}{g_1^2}}$. We assume that $\{g_1, g_2, ..., g_t\}$ are approximated by i.i.d random variables drawn from normal distribution $\mathcal{N}(0, \sigma^2)$ given that at the beginning of the training, weights follow normal distribution with mean zero [11]. Thus, it follows that $\frac{1}{g_1^2}$ has the distribution of the scaled inverse chi-squared distribution with parameters $\nu = 1$ and $\tau^2 = \frac{1}{\sigma^2}$.

Theorem 1. *If X is a random variable subject to the scaled inverse chi-squared distribution with parameters $\nu = 1$ and $\tau^2 = \frac{1}{\sigma^2}$, then $Var(\sqrt{X})$ is divergent.*

Proof. Firstly, let's write $Var(\sqrt{X}) = E(X) - E(\sqrt{X})^2$. Moreover,

$$E(X) = \frac{1}{\sqrt{2}\sigma\Gamma(\frac{1}{2})}\int_0^\infty \frac{exp(\frac{-1}{2x\sigma^2})}{\sqrt{x}}dx = \frac{1}{2\sigma^2\sqrt{\pi}}\int_0^\infty \frac{exp(-u)}{u\sqrt{u}}dx$$

$$E(\sqrt{X}) = \int_0^\infty \sqrt{x}\frac{exp(\frac{-1}{2x\sigma^2})}{\sqrt{2}\sigma\Gamma(\frac{1}{2})x^{1.5}}dx = \frac{1}{\sqrt{2}\sigma\Gamma(\frac{1}{2})}\int_0^\infty \frac{exp(-u)}{u}dx = \frac{1}{\sqrt{2}\sigma\sqrt{\pi}}\int_0^\infty \frac{exp(\frac{-1}{2x\sigma^2})}{x}dx$$

where we applied the substitution $u = \frac{1}{2x\sigma^2}$. We obtain :

$$Var(\sqrt{X}) = \frac{1}{2\sigma^2\sqrt{\pi}}\int_0^\infty\int_0^\infty \frac{exp(-x)}{x}(\frac{1}{\sqrt{x}} - \frac{exp(-y)}{\sqrt{\pi}y})dxdy$$

As this integral is divergent, we conclude that $Var(\sqrt{X})$ is divergent. □

It can be deduced from the previous theorem that the adaptive learning rate during the first step may be very large which is problematic. In the next paragraph, we show that the variance of the adaptive learning rate is decreasing monotonically as the number of epochs of training is getting larger.

2.2.3 Evolution of the adaptive learning rate variance

Now let's show that the variance of the adaptive learning rate decreases after each step. Firstly, we simplify the problem by approximating the distribution of the exponential moving average in the Adam algorithm with the simple moving average (justified in this paper [12]):

$$p(\psi(.)) = p(\sqrt{\frac{1-\beta_2^t}{(1-\beta_2)\sum_{i=1}^t \beta_2^{t-i} g_t^2}}) \sim p(\sqrt{\frac{t}{\sum_{i=1}^t g_i^2}}) \tag{1}$$

According to the previous analysis, $\frac{t}{\sum_{i=1}^t g_i^2}$ follows the distribution of the scaled inverse chi-squared distribution with parameters $\nu = t$ and $\tau^2 = \frac{1}{\sigma^2}$, and we can assume that $\frac{1-\beta_2^t}{(1-\beta_2)\sum_{i=1}^t \beta_2^{t-i} g_t^2}$ follows the same distribution with ρ degrees of freedom (discussed in more detail in [1]), where ρ will be estimated in the next section. As a result, the following theorem shows that the variance of $\psi(.)$ decreases when t increases (proof in [1])

Theorem 2. *If $\psi^2(.)$ is drawn from a scaled inverse chi-squared distribution with parameters $\nu = \rho$ and $\tau^2 = \frac{1}{\sigma^2}$, then $Var(\psi(.))$ monotonically decreases as ρ increases.*

3 Radam : Adam optimizer with rectified adaptive learning rate

3.1 Estimation of the number of degrees of freedom

In the last section, $\psi(.)^2$ was approximated by considering a scaled inverse chi-squared distribution. The aim is to estimate the number of degrees of freedom ρ_t based on t which will contribute to better handle the variance of the adaptive learning rate and design a rectified adam optimizer.

As explained previously, we approximate the exponential moving average by a simple moving average [12]:

$$p(\frac{(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}g_t^2}{1-\beta_2^t}) \sim p(\frac{\sum_{i=1}^{f(t,\beta_2)} g_{t+1-i}^2}{t}) \tag{2}$$

$f(t,\beta_2)$ is the length of the simple moving average and since it has the same "center of mass" with the exponential moving average, we deduce the following equation :

$$\frac{(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}i}{1-\beta_2^t} = \frac{\sum_{i=1}^{f(t,\beta_2)}(t+1-i)}{f(t,\beta_2)} \tag{3}$$

As can be noticed, the expression above is slightly different from the one indicated in the paper published on the website arxiv [1], in which the left hand-side numerator of the equation is $(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}(t+1-i)$. Nevertheless, I found another version of this paper on the website openreview [13] with the formula above and I will show in the next paragraph that the solution given by this paper holds only for this equation.

Let's solve this equation. Firstly, we have: $\sum_{i=a}^{b} i = \frac{(a+b)(b-a+1)}{2}$ if $a < b$ with a,b $\in \mathbb{N}$. Thus

$$\sum_{i=1}^{f(t,\beta_2)}(t+1-i) = \frac{f(t,\beta_2)(2t+1-f(t,\beta_2))}{2}$$

The right hand-side is equal to $\frac{(2t+1-f(t,\beta_2))}{2}$. Now, let's write :

$$A = \frac{(1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}i}{1-\beta_2^t} = \frac{1-\beta_2}{1-\beta_2^t}\beta_2^t\frac{1}{\beta_2}\sum_{i=1}^{t}(\frac{1}{\beta_2})^{i-1}i = -\frac{t+1}{\beta_2^t-1} + \frac{\beta_2^{t+1}-1}{(\beta_2^t-1)(\beta_2-1)}$$

As $\sum_{k=1}^{t} kx^{k-1} = \frac{-(t+1)x^t(1-x)+1-x^{t+1}}{(1-x)^2}$. After simplifying, we get : $A = \frac{t}{1-\beta_2^t} - \frac{\beta_2}{1-\beta_2}$. Thus

$$f(t,\beta_2) = 2t+1-2(\frac{t}{1-\beta_2^t}-\frac{\beta_2}{1-\beta_2}) = -1+(2t+2-2(\frac{t}{1-\beta_2^t}-\frac{\beta_2}{1-\beta_2})) = -1+\frac{2}{1-\beta_2}-\frac{2t\beta_2^t}{1-\beta_2^t}$$

Thus $\psi(.)^2$ ~Scale-inv-$\chi^2(f(t,\beta_2), \frac{1}{\sigma^2})$ and we have $\lim_{t\to\infty} f(t,\beta_2) \leq \frac{2}{1-\beta_2} - 1$
By denoting $\rho_t = f(t,\beta_2)$ and based on Theorem 2, we deduce that : min $Var_{\rho_t}(\psi(.)) = Var(\psi)_{\rho_\infty}$
The modified Adam optimizer is built such that at each time step the adaptive learning rate has consistent variance and use a rectification term r_t:

$$Var(r_t\psi(g_1,...,g_t)) = Var(\psi)_{\rho_\infty} \text{with} \quad r_t = \sqrt{\frac{Var(\psi)_{\rho_\infty}}{Var(\psi(g_1,...,g_t))}} \tag{4}$$

Given that $\psi(.)^2$ ~Scale-inv-$\chi^2(\rho_t, \frac{1}{\sigma^2})$, we approximate for $\rho_t > 4 : Var(\psi(.)) \approx \frac{\rho_t}{2(\rho_t-2)(\rho_t-4)\sigma^2}$ and we conclude that $r_t = \sqrt{\frac{\rho_\infty(\rho_t-2)(\rho_t-4)}{\rho_t(\rho_\infty-2)(\rho_\infty-4)}}$.

3.2 Radam Algorithm

The Radam algorithm corrects the adaptive learning rate at each time step. Given that the variance is divergent during the first steps (when $\rho_t \leq 4$), the adaptive learning rate is ignored. Otherwise, the adaptive learning rate is adjusted thanks to the rectification term. A further analysis was conducted in the paper showing that warm up heuristic acts as a variance reduction similarly as Radam while Radam simplifies the task by eliminating the need for manually tuning warm up schedules.

Algorithm 2 Radam algorithm

0: **Inputs:**
$\{\alpha_t\}_{t=1}^{T}$ step size, $\{\phi(t), \psi(t)\}_{t=1}^{T}$: momentum and adaptive learning
θ_0 initial parameter and $f(\theta)$ the objective function
0: **Output:**
θ_T
1: $m_0, v_0 \leftarrow 0, 0$
2: $\rho_\infty \leftarrow \frac{2}{1-\beta_2} - 1$
3: **for** $t = 1$ *to* T **do**
4: $g_t \leftarrow \triangle_\theta f_t(\theta_{t-1})$
5: $m_t \leftarrow \phi_t(g_1, g_2, ..., g_{t-1})$
6: $v_t \leftarrow \psi_t(g_1, g_2, ..., g_{t-1})$
7: $\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$
8: $\rho_t \leftarrow \rho_\infty - \frac{2\beta_2^t}{1-\beta_2^t}$
9: $\theta_t \leftarrow \theta_{t-1} - \alpha_t m_t v_t$
10: **if** $\rho_t > 4$ **then**
11: $\hat{v}_t \leftarrow \sqrt{\frac{v_t}{1-\beta_2^t}}$
12: $r_t \leftarrow \sqrt{\frac{(\rho_t-4)(\rho_t-2)\rho_\infty}{(\rho_\infty-4)(\rho_\infty-2)\rho_t}}$
13: $\theta_t \leftarrow \theta_{t-1} - \alpha_t r_t \hat{m}_t / \hat{v}_t$
14: **else**
15: $\theta_t \leftarrow \theta_{t-1} - \alpha_t \hat{m}_t$
16: **end if**
17: **end for**=0

4 Experimentations

4.1 Implementation details

As CIFAR10 and ImageNet was used to assess the performance of the algorithm on image classification in [1], I decided to empirically evaluate the rectified Adam optimizer on the Fashion Mnist dataset. I chose to focus my investigation on comparing the robustness of Radam, Adam and Adam with warmup to variations of the learning rate. I used the library Pytorch on Python where Radam optimizer was already implemented and visualized the performance of the algorithm with the help of tensorboard (accuracy, loss and information on the distribution of weights). The dataset Fashion Mnist is composed of 60000 training examples with 10000 test examples
Whereas the reseachers used ResNet architecture, I implemented a less complex model as shown below. When it comes to the choice of the hyperparameters (β_1, β_2), I tuned them with the most widely used values indicated in [1]: $\beta_2 = 0.999$ and $\beta_1 = 0.9$.

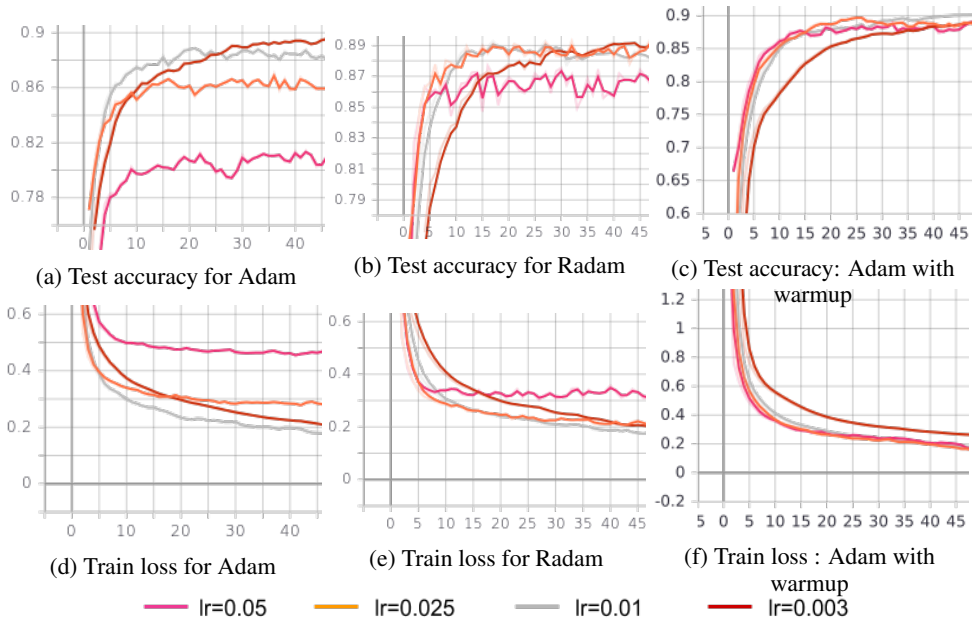
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 6, 24, 24]             156
            Conv2d-2            [-1, 12, 8, 8]           1,812
            Linear-3                 [-1, 120]          23,160
            Linear-4                  [-1, 60]           7,260
            Linear-5                  [-1, 10]             610
================================================================
Total params: 32,998
Trainable params: 32,998
Non-trainable params: 0
```

Figure 2: Architecture of the model for the dataset Fashion Mnist

4.2 Comparison between Adam and Radam

For the first set of experiments, I decided to examine the robustness to the learning rate for Adam and Radam optimizer. Thus I trained the model for different values of learning rate and observed the test accuracy and train loss. The performances on Fashion mnist dataset are summarized in the figure 4.

As can be observed, the final accuracy and loss rate are quite similar when the learning rate is low (red and grey learning curves corresponding to $lr = 0.01, 0.003$). We notice that when $lr = 0.003$, Radam is slower than Adam in the first epochs and this is probably due to the effect of the rectification term. However, Radam considerably outperforms Adam when the learning rate becomes high. Indeed, for $\alpha = 0.025$, we observe a significant decrease in the performance of Adam while Radam learning curves aren't changed. What's more, when $\alpha = 0.05$, there are 6 points of difference between the final accuracy rate of Adam and Radam. This analysis demonstrates that adam is much more senstitive to the change of the learning rate. These results could be explained by the fact that Radam algorithm adjusts at each step the variance of the adaptive learning rate which induces more robustness to the learning rate change.



(a) Test accuracy for Adam (b) Test accuracy for Radam (c) Test accuracy: Adam with warmup

(d) Train loss for Adam (e) Train loss for Radam (f) Train loss : Adam with warmup

Figure 4: Performance of Adam, Radam and Adam with linear warmup with different learning rates on Fasion Mnist dataset with a batch size of 1000. X-axis is the number of epochs.

4.3 Comparison between Radam and Adam with linear warm up

In order to apply warm up to Adam optimizer, I referred to a paper [14] recently published questioning the underpinnings of Radam, claiming that simple untuned warm up heuristic with Adam performs similarly as Radam. They found that using a linear warmup rate during $\frac{2}{1-\beta_2}$ training steps leads to more or less the same results. The formulation of warmup is defined as $\omega_t = min(1, \frac{1-\beta_2}{2}t)$, where ω_t is multiplied by the learning rate. Using the code implemented from the paper, I tested Adam with warmup on Fashion Mnist. The results are summarized in the figure 4. We see that Adam with warmup achieves at least the same performance as Radam and has even a higher accuracy rate when $lr = 0.05$. Furthermore, even if its convergence rate is slower than Adam and Radam, Adam with warmup heuristic shows better training stability with less oscillations. Thus, we conclude that the warmup learning rate scheduler slows down the training process but entails better stability.

5 Conclusion

In this report, I studied the underlying motivation of warm up method's role in Adam based on a recent paper [1]. More precisely, the effectiveness of warmup was explained by its action as a variance reduction for the adaptive learning rate in the early stages of training. This analysis led to the development of a new version of Adam which automatically adjusts the learning rate and doesn't need any tuning in contrast with warmup. The experiments showed convincing results when comparing Adam and Radam as the latter achieves better performance. However, the outcome remains reserved when comparing Adam with warm up and Radam, which supports the analysis conducted in [14]. Indeed, they re-evaluated the large variance of the adaptive learning rate at the beginning of training by criticizing the assumption that gradients are zero mean and pointing out that the first and second moment estimators are correlated. As a result, they claim that the high variance of the adaptive learning rate $\sqrt{\frac{1}{v_t}}$ is balanced with m_t since Adam's updates are proportional to $\frac{m_t}{\sqrt{v_t}}$. Therefore, [14] advises practitionners to keep using warmup schedules for Adam by providing two simple models requiring no tuning.

References

[1] Liu et al., On the Variance of the Adaptive Learning Rate and Beyond, arXiv:1908.03265, 2019: Theoretical and empirical justification for warm up heuristic and introduction of Radam

[2] John Duchi, Elad Hazan, Yoram Singe, Adaptive Subgradient Methods forOnline Learning and Stochastic Optimization, Journal of Machine Learning Research 12 (2011) 2121-2159: Introduction of Adagrad

[3] Matthew D. Zeiler, ADADELTA: An Adaptive Learning Rate Method, arXiv:1212.5701, 2012: Introduction of Adadelta

[4] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, arXiv:1412.6980, 2014: Introduction of Adam

[5] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher: A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. 7th International Conference on Learning Representations, 2019: Heuristic methods improving the performance of optimization algorithms

[6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, Kaiming He, Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour, arXiv:1706.02677, 2017: First paper to mention warm up heuristic

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, arXiv:1706.03762, 2017: Use of warmup for transformers

[8] Martin Popel, Ondřej Bojar, Training Tips for the Transformer Model, arXiv:1804.00247, 2018 : Resort to warmup to improve performance of Transformers

[9] Nikolay Bogoychev, Marcin Junczys-Dowmunt, Kenneth Heafield, Alham Fikri Aji, Accelerating Asynchronous Stochastic Gradient Descent for Neural Machine Translation, 2018: Resort to warmup to improve performance of Neural Machine Translation

[10] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond, arXiv:1904.09237, 2019: Define generic algorithm for adaptive learning rate optimization methods

[11] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams.The shattered gradients problem: If resnets are the answer, then what is the question?, arXiv:1702.08591, 2017 : Justify why weights can be approximated by normal distribution with zero mean

[12] Robert Nau, Forecasting with moving averages, 2014 : Justify why distribution of the exponential moving average is approximated by the distribution of the simple average

[13] Liu et al., On the Variance of the Adaptive Learning Rate and Beyond, https://openreview.net/pdf?id=rkgz2aEKDr : Slight difference with one equation in the paper on the website Arxiv

[14] Jerry Ma, Denis Yarats, On the adequacy of untuned warmup for adaptive optimization, arXiv:1910.04209, 2019 : Question the effectiveness of Radam