Learning Rate Tuning: Warm-up & Linear Decay

Ki Hyun Kim

nlp.with.deep.learning@gmail.com

Previous Methods

SGD

- 가장 기본적인 방법

$$\theta \leftarrow \theta - \gamma \nabla_{\theta} \mathcal{L}(\theta)$$

- Learning rate(LR)에 따른 성능 변화
- 학습 후반부에 LR decay 해주기도

Adam

- Adaptive하게 LR을 조절

Algorithm 1: Generic adaptive optimization method setup. All operations are element-wise.

Input: $\{\alpha_t\}_{t=1}^{T}$: step size, $\{\phi_t, \psi_t\}_{t=1}^{T}$: function to calculate momentum and adaptive rate, θ_0: initial parameter, $f(\theta)$: stochastic objective function.
Output: θ_T: resulting parameters
while $t = 1$ *to* T **do**
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Calculate gradients w.r.t. stochastic objective at timestep t)
  $m_t \leftarrow \phi_t(g_1, \cdots, g_t)$ (Calculate momentum)
  $l_t \leftarrow \psi_t(g_1, \cdots, g_t)$ (Calculate adaptive learning rate)
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t m_t l_t$ (Update parameters)
return θ_T

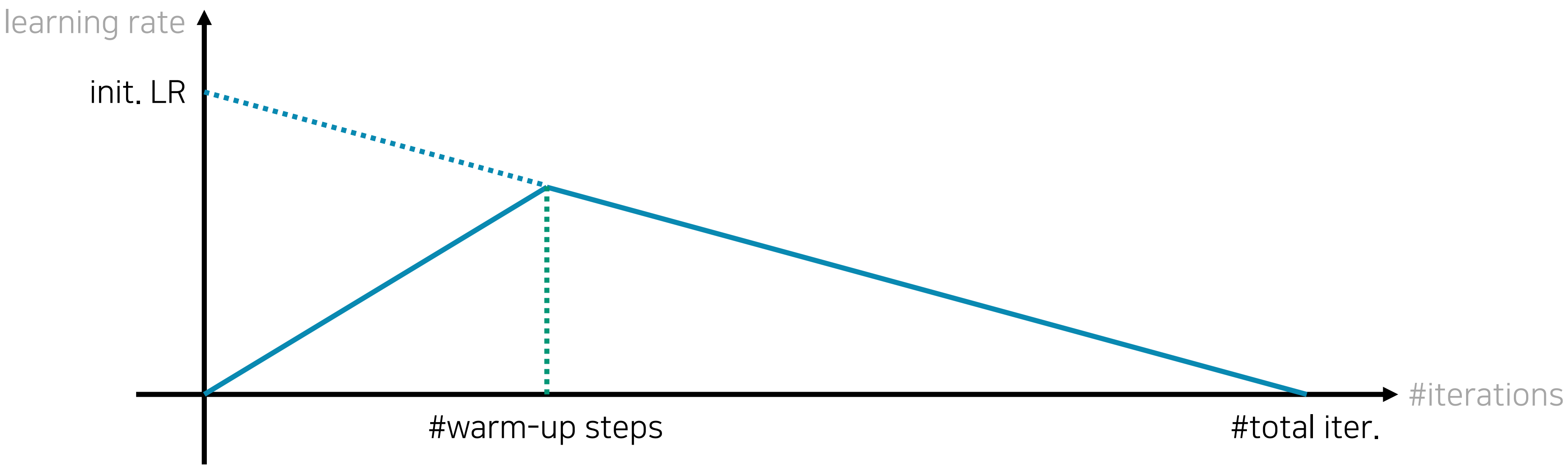
출처: [Liu et al., 2020]

- 일부 깊은 네트워크(e.g. Transformer)에서 성능이 낮음
  - 문제는 지금은 Transformer의 세상

Warm-up and Linear Decay (Noam Decay)

- Heuristic Methods
  - Control learning rate for Adam with hyper-params

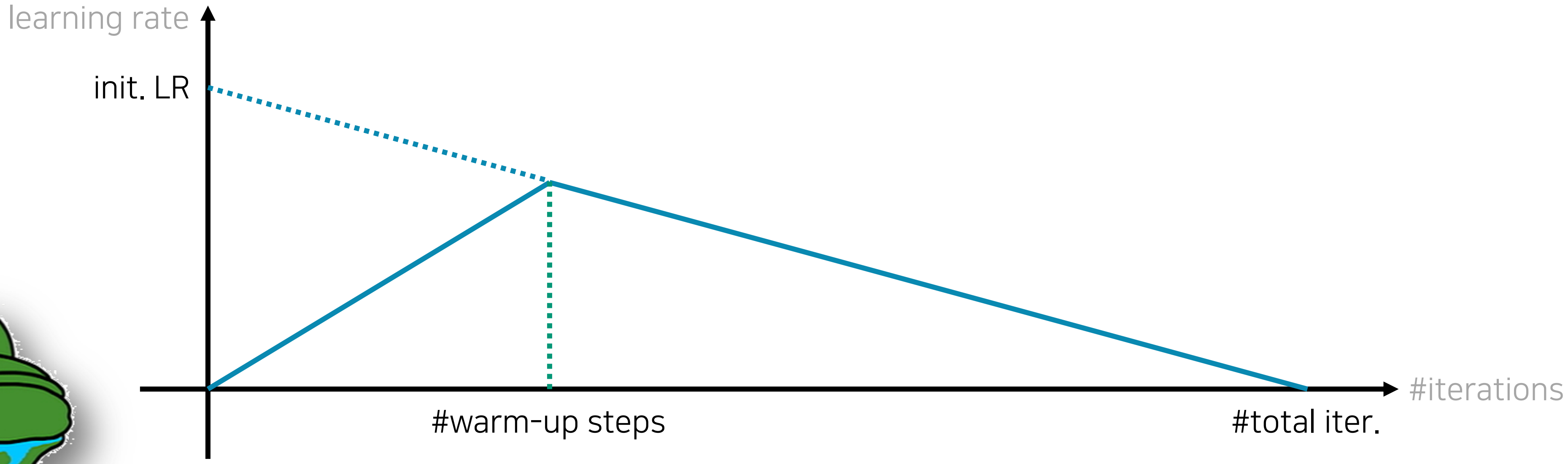
- 학습 초기 불안정한 gradient를 통해 잘못된 momentum을 갖는 것을 방지



Warm-up and Linear Decay (Noam Decay)

- 결국 Trial & Error 방식으로 Hyper-parameter 튜닝을 해야 함
  - 가장 핵심은 #warm-up steps와 #total iterations.
  - 이외에도 다양한 hyper-params: init LR, batch size

- 심지어 튜닝에 따라 SGD + Gradient Clipping이 더 나은 결과를 얻기도 함



Rectified Adam [Liu et al., 2020]

- Adam이 잘 동작하지 않는 이유(가설)
  - Due to the lack of samples in the early stage, the adaptive learning rate has an undesirably large variance, which leads to suspicious/bad local optima. – [Liu et al., 2020]

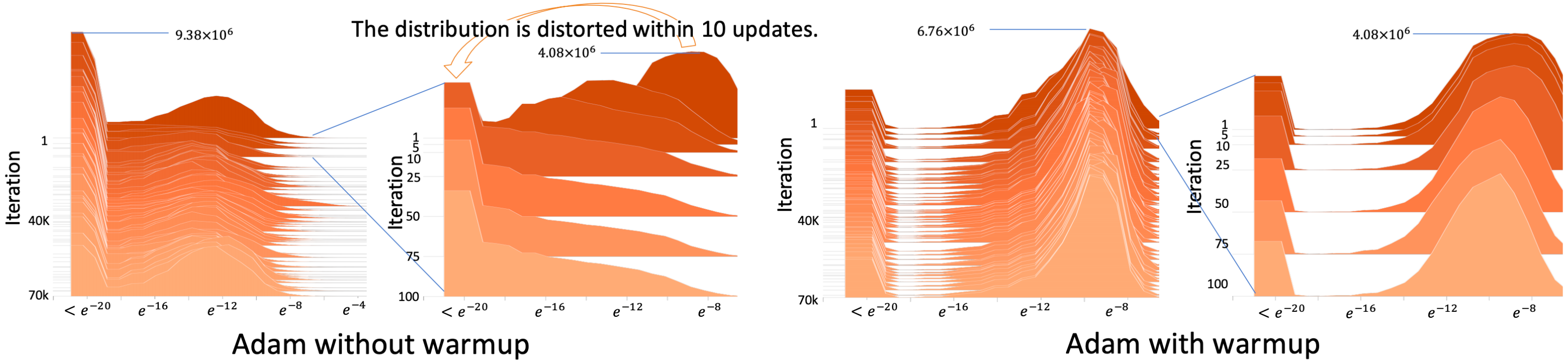


Figure 2: The absolute gradient histogram of the Transformers on the De-En IWSLT' 14 dataset during the training (stacked along the y-axis). X-axis is absolute value in the log scale and the height is the frequency. Without warmup, the gradient distribution is distorted in the first 10 steps.

- PyTorch 구현
  - https://github.com/LiyuanLucasLiu/RAdam
  - $ pip install torch-optimizer