ACADENTIAL

Section 11.
Learning Rate Scheduler

ACADENTIAL

Recap from previous chapters

Optimization

Optimization에서는

- (Full-batch, Stochastic, Mini-batch) Gradient Descent 외에 다른 최적화 방법은 어떤 것들이 있었는지
- 학습이 더 안정적이고, 더 빠르게 학습이 수렴하게 해주는 방법을 살펴보았다!

ACADENTIAL

Recap from previous chapters

Optimization

구체적으로는:

- Momentum (관성)의 개념 이해
- Acceleration (가속)의 개념 이해 (Nesterov's Accelerated Gradient)
- Adaptive Learning Rate 기반의 방법 이해
  - AdaGrad
  - AdaDelta
  - RMSProp
- Adaptive Moment Estimation (ADAM) 이해

ACADENTIAL

Learning Rate Scheduler

What is it?

- “Optimization”에서는 Learning Rate을 Gradient의 history에 따라 “adaptive”하게 조절해주는 방법들이 있었다.
  - (e.g. AdaGrad, AdaDelta, RMSProp)
- 하지만 학습이 진행되면서의 학습 경과 (Validation Loss)나 Time step (Gradient descent의 iteration 수)에 따라 Learning Rate을 조절하는 방법은 없을까?

바로 **Learning Rate Scheduler**!

ACADENTIAL

목차

ACADENTIAL

Objective
학습목표

다양한 종류의 Learning Rate Scheduler에 대해서 살펴보자:

1. Step LR Scheduler
2. Exponential LR Scheduler
3. Cosine Annealing with Warm Restarts LR Scheduler
4. Reduce on Plateau LR Scheduler
5. Linear Scheduler with Warmup
6. Cosine Scheduler with Warmup

ACADENTIAL

11-1. Step LR scheduler

ACADENTIAL

Learning Rate Scheduler

Step LR Scheduler

핵심:

계단식으로 **“step-size” iteration**마다 **LR을 줄이는 것**

ACADENTIAL

Learning Rate Scheduler

Step LR Scheduler

핵심:

계단식으로 “step-size” iteration마다 LR을 줄이는 것

Parameters: γ, step-size

Step-size마다 LR에 γ 을 곱해준다. (참고로, $0 < \gamma < 1$)

ACADENTIAL

Learning Rate Scheduler

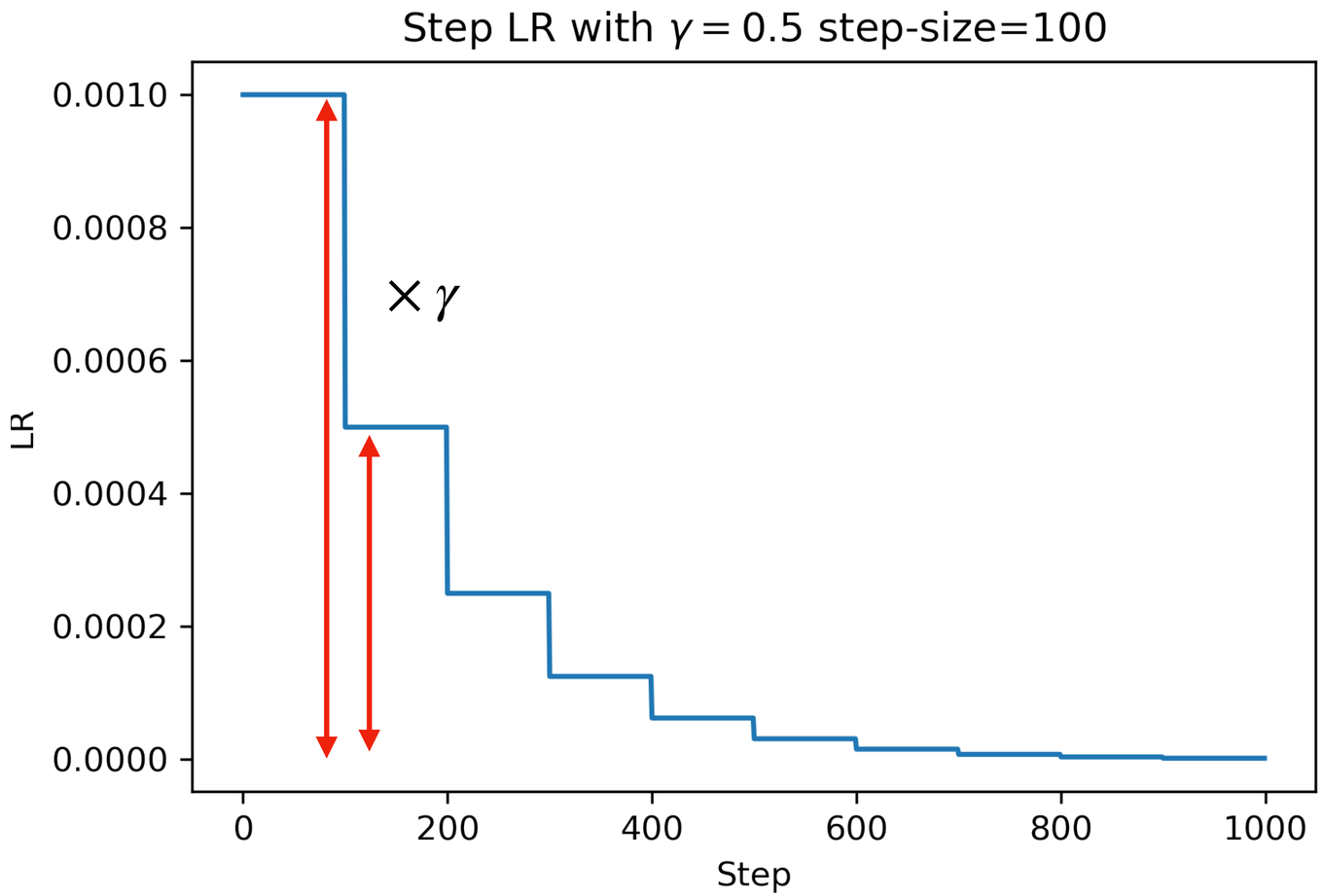
Step LR Scheduler

Parameters: γ, step-size

Step-size마다 LR에 γ을 곱해준다. (참고로, $0 < \gamma < 1$)

오른쪽 예시:

- γ = 0.5, step-size = 100
- **torch.optim.lr_scheduler.StepLR**



ACADENTIAL

Learning Rate Scheduler

Step LR Scheduler

핵심:

계단식으로 **“step-size” iteration**마다 **LR을 줄이는 것**

이유 / Motivation:

- 학습이 진행되면서 모델이 Local / Global minimum에 점점 더 가까워진다.
- learning rate을 줄여주어 minimum 주변에서 oscillate하는 것을 완화해주기 위해서.

ACADENTIAL

11-2. Exponential LR scheduler

ACADENTIAL

Learning Rate Scheduler

Exponential LR Scheduler

핵심:

“Exponential”하게 **LR을 줄이는 것.**

ACADENTIAL

Learning Rate Scheduler

Exponential LR Scheduler

핵심:

“Exponential”하게 **LR을 줄이는 것.**

Parameters: γ

매 **Epoch**마다 **LR에** γ**을 곱해준다.** (참고로, $0 < \gamma < 1$)

Learning Rate Scheduler

Exponential LR Scheduler

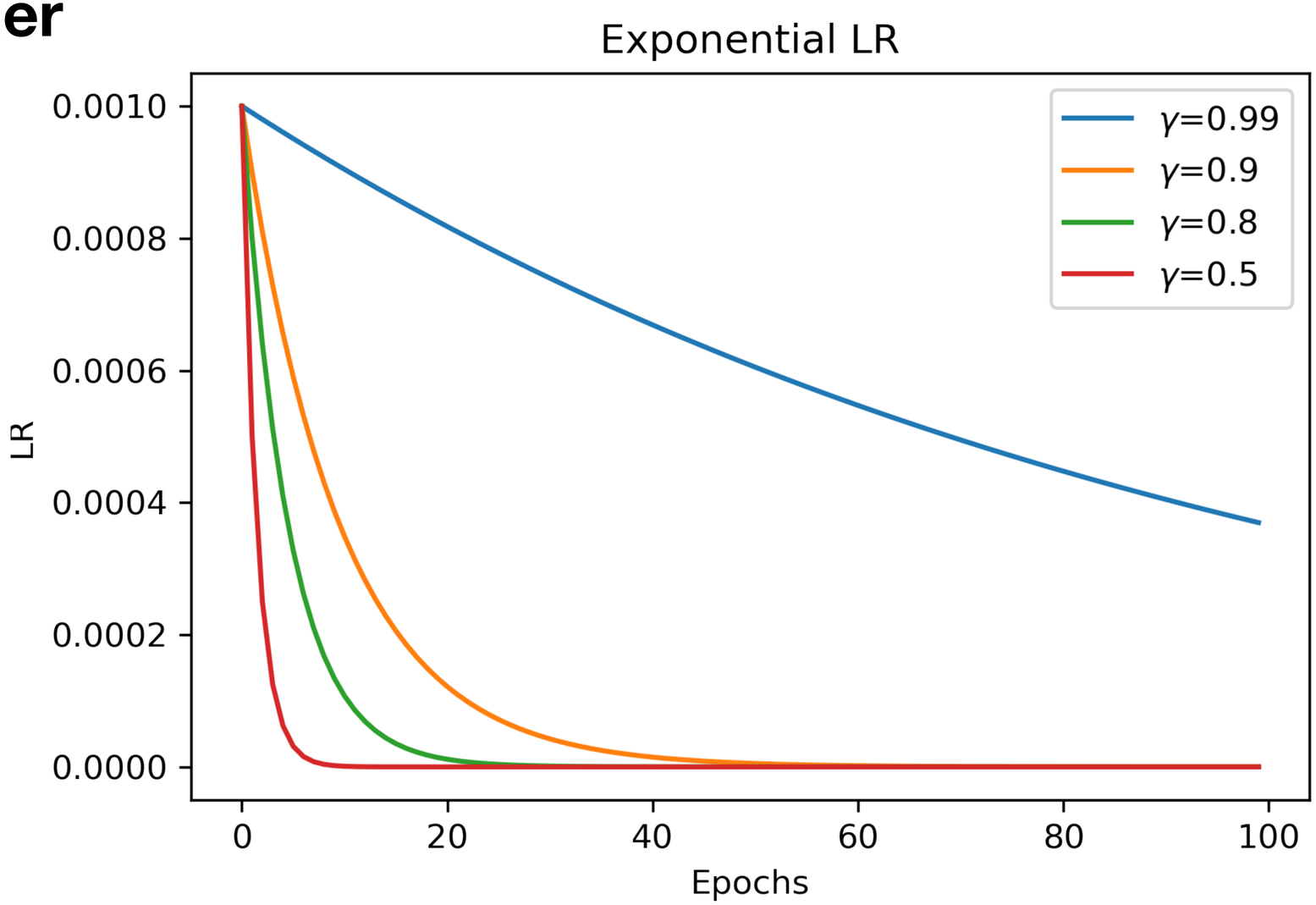
Parameters: γ

매 Epoch마다 LR에 γ을 곱해준다.
(참고로, $0 < \gamma < 1$)

오른쪽 예시:

γ = 0.99, 0.9, 0.8, 0.5

torch.optim.lr_scheduler.ExponentialLR



ACADENTIAL

Learning Rate Scheduler

Exponential LR Scheduler

핵심:

“Exponential”하게 **LR을 줄이는 것.**

이유 / Motivation:

- 학습 초기 단계에서는 데이터의 일반적인 패턴을 학습하는데 상대적으로 높은 LR 필요
- 학습이 진행됨에 따라 모델이 잘 수렴할 수 있게 Learning Rate을 줄여서 모델 weight들을 미세 조정해야함

ACADENTIAL

11-3. Cosine Annealing LR scheduler

ACADENTIAL

Learning Rate Scheduler

Cosine Annealing Scheduler

핵심:

Cosine 함수로 LR을 조절하는 것.

ACADENTIAL

Learning Rate Scheduler

Cosine Annealing Scheduler

$$LR = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{T_{cur}}{T_{max}}\pi\right)\right)$$

Parameters: η_{min}, T_{max}

- T_{max} = cosine함수의 반 주기에 해당되는 epoch개수
- η_{min} = 최소 LR값

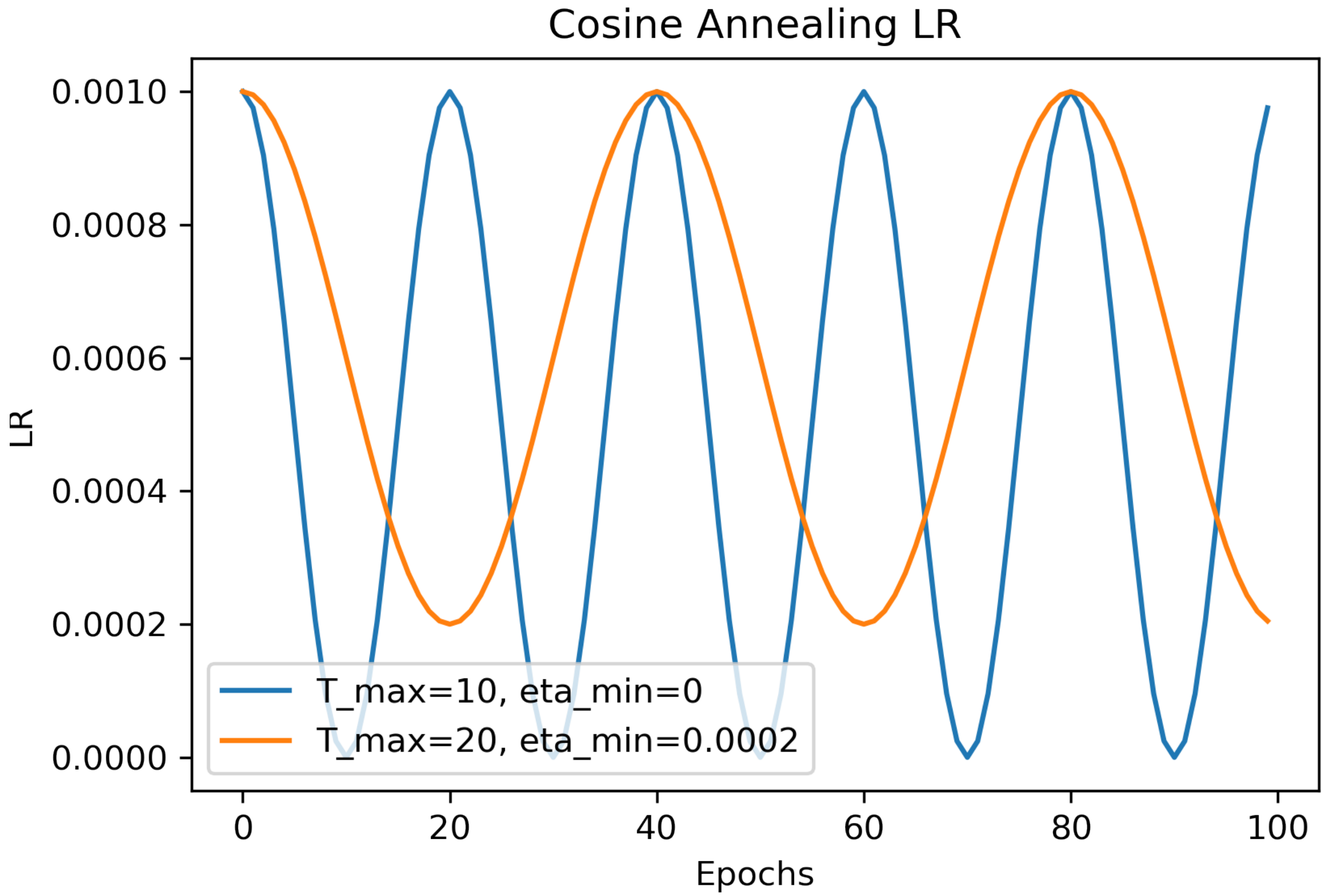
ACADENTIAL

Learning Rate Scheduler

Cosine Annealing Scheduler

Parameters: η_{min}, T_{max}

- T_{max} = cosine함수의 반 주기에 해당되는 epoch개수
- η_{min} = 최소 LR값



ACADENTIAL

Learning Rate Scheduler

Cosine Annealing Scheduler

이유 / Motivation:

- **학습 수렴에 도움**
  - Learning Rate을 주기적으로 감소시키는 방식 → 높은 학습률로 초기에 빠르게 수렴 + 학습률을 감소시켜서 모델이 수렴
- **Local minima 고착화 (Overfitting) 방지**
  - 학습률을 주기적으로 변화 → 모델이 지역 최소값(local minima)에 고착되는 것을 방지 → 더 다양한 파라미터 공간을 탐색하게 되며, 과적합을 피할 수 있음

ACADENTIAL

11-4. Cosine Annealing with Warm Restarts LR scheduler

ACADENTIAL

Learning Rate Scheduler

Cosine Annealing with Warm Restarts Scheduler

핵심:

Cosine 함수로 LR을 줄이다가 다시 Max LR로 주기적으로 Restart하는 것.

Learning Rate Scheduler

Cosine Annealing with Warm Restarts Scheduler

핵심:

Cosine 함수로 LR을 줄이다가 다시 Max LR로 주기적으로 Restart하는 것.

Parameters: T_0, T_{mult},

- T_0 = LR을 처음으로 restart하는데까지의 Iteration 개수 (epoch 개수)
- $T_{mult} = \frac{T_{i+1}}{T_i}$
  - 참고로, T_i은 i-1번째 restart한 후 다음 (즉 i번째) restart 하기까지 걸리는 iteration 개수.

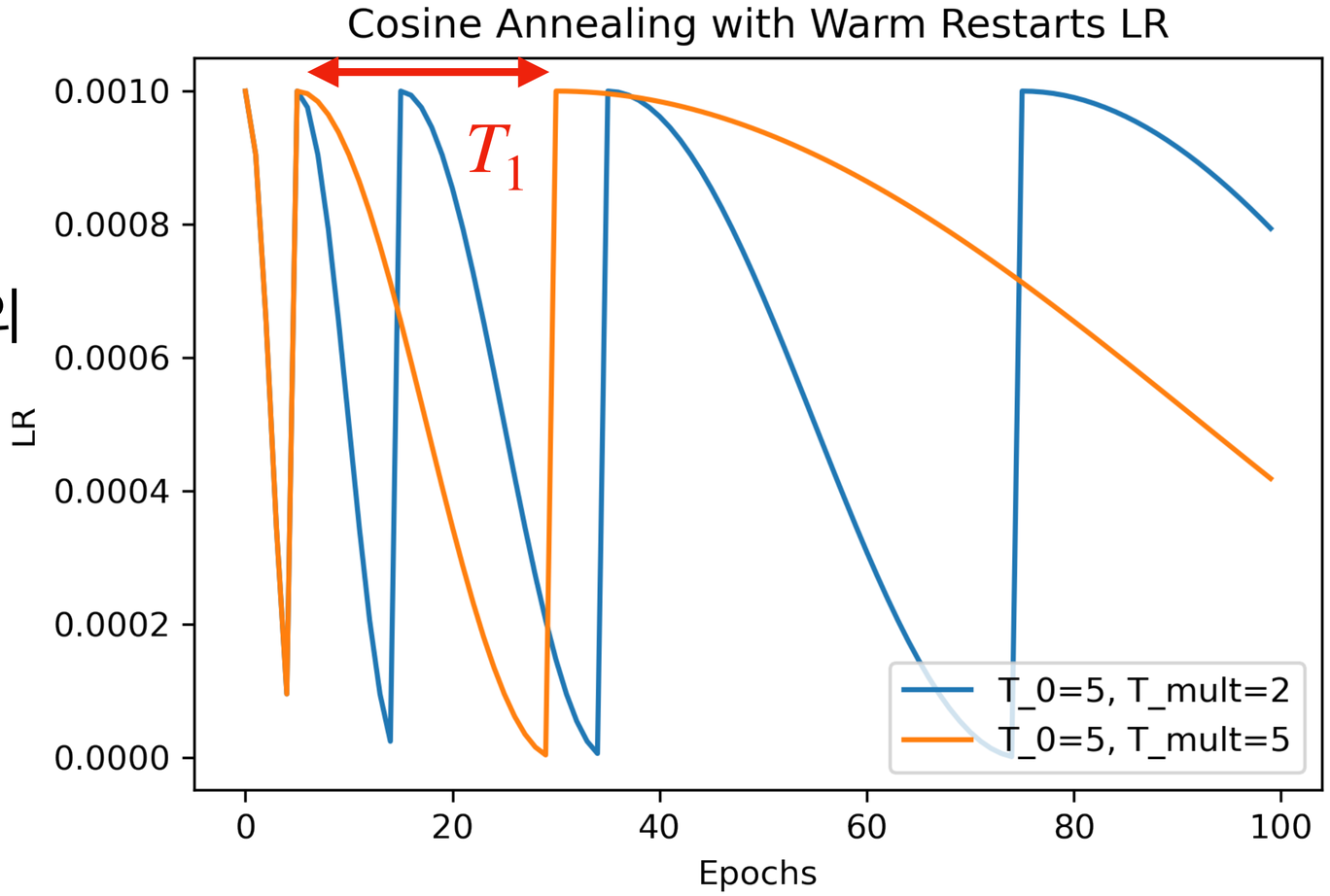
ACADENTIAL

Learning Rate Scheduler

Cosine Annealing with Warm Restarts Scheduler

Parameters: T_0, T_{mult}

- T_0 = LR을 처음으로 restart하는데까지의 Iteration 개수 (epoch 개수)
- $T_{mult} = \frac{T_{i+1}}{T_i}$



ACADENTIAL

Learning Rate Scheduler

Cosine Annealing with Warm Restarts Scheduler

핵심:

Cosine 함수로 LR을 줄이다가 다시 Max LR로 주기적으로 Restart하는 것.

이유 / Motivation:

- **(Cosine Annealing LR Scheduler와 동일)** Learning Rate을 주기적으로 감소시키는 방식 높은 학습률로 초기에 빠르게 수렴 + 학습률을 감소시켜서 모델이 수렴
- **(Cosine Annealing LR Scheduler와 동일)** 학습률을 주기적으로 변화 모델이 지역 최소값(local minima)에 고착되는 것을 방지 더 다양한 파라미터 공간을 탐색하게 되며, 과적합을 피할 수 있음

ACADENTIAL

Learning Rate Scheduler

Cosine Annealing with Warm Restarts Scheduler

이유 / Motivation:

- **Ensemble의 구성**
  - 여러번 Restart → 각 Restart마다 모델 Checkpoint를 얻을 수 있음 → 체크포인트들로 모델 **Ensemble 구성할 수 있음**
- **Cosine Annealing LR Schduler의 한계**
  - 빈번한 restart로 인해서 하나의 minimum에 정착 못함
- **Cosine Annealing with Warm Restarts Scheduler의 해결책**:
  - Restart을 허용하되 Restart 주기를 늘려주어 “차선책”의 Local minimum에 잘 수렴하게끔 도와준다.

ACADENTIAL

11-5. Reduce on Plateau LR scheduler

ACADENTIAL

Learning Rate Scheduler

Reduce on Plateau LR Scheduler

핵심:

“모델의 성능이 개선되지 않을때 LR을 조금씩 줄이는 것”

ACADENTIAL

Learning Rate Scheduler

Reduce on Plateau LR Scheduler

핵심:

“모델의 성능이 개선되지 않을때 LR을 조금씩 줄이는 것”

Parameters: **Factor, Patience**

- **Patience** 이상의 iteration 동안 Monitoring 대상이 개선되지 않을시 Learning Rate을 (곱하기 **Factor**) 하는 것.
- Factor = 0이상, 1이하의 수

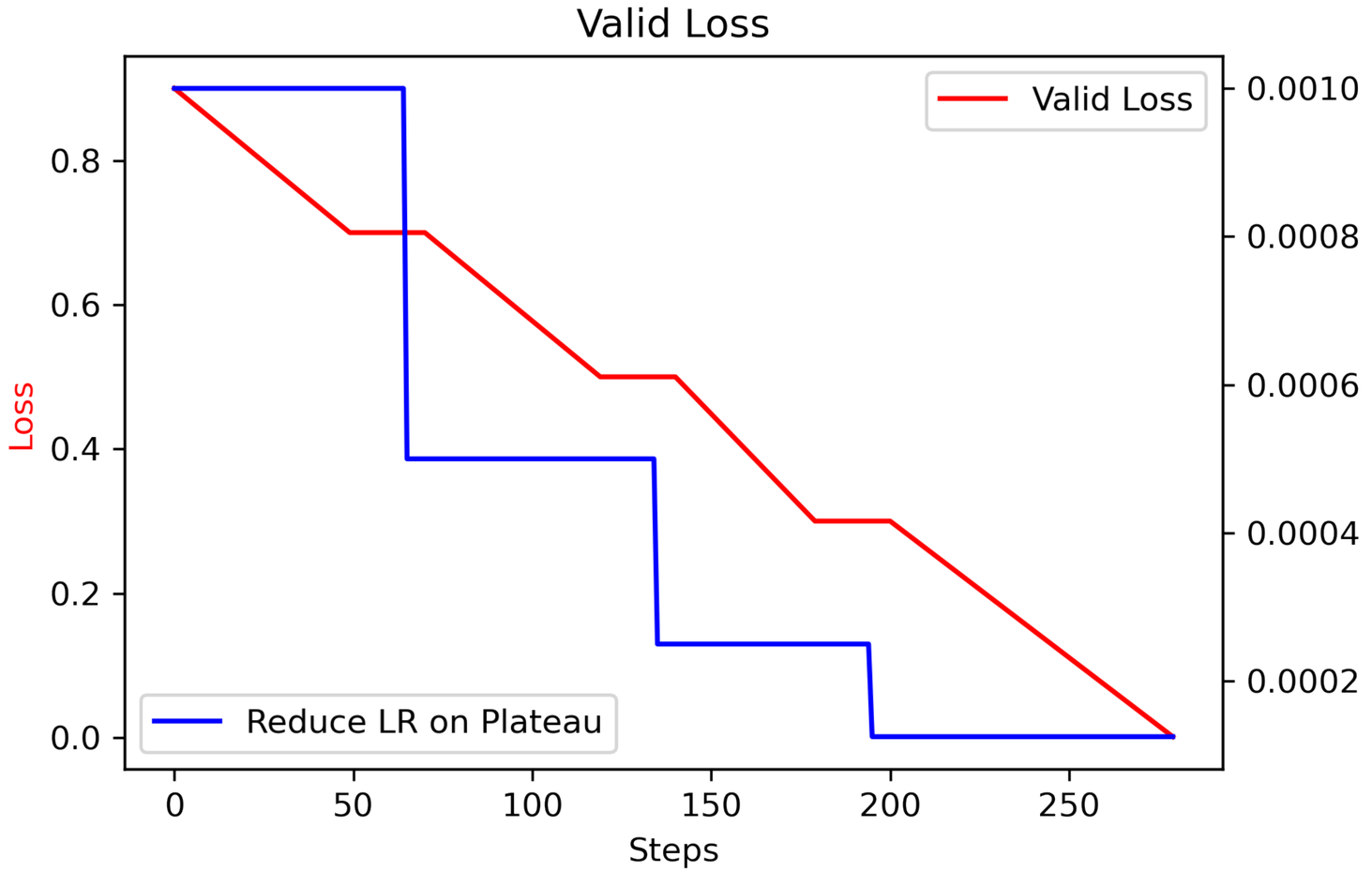
ACADENTIAL

Learning Rate Scheduler

Reduce on Plateau LR Scheduler

Parameters: **Factor, Patience**

- **Patience** 이상의 iteration 동안 Valid Loss가 개선되지 않을시 Learning Rate을 (곱하기 **Factor**) 하는 것.
- Factor = 0이상, 1이하의 수



ACADENTIAL

Learning Rate Scheduler

Reduce on Plateau LR Scheduler

이유 / Motivation:

- **수렴 속도 향상**
  - 모델의 성능이 개선되지 않을때 학습률 감소 → 학습 수렴에 용이함
- **안정적인 학습**
  - 모델 성능 개선을 기준으로 학습률을 동적으로 조절
  - 불안정한 Learning Rate 설정의 위험 피함

ACADENTIAL

11-6.
Linear & Cosine scheduler with Warmup

ACADENTIAL

Learning Rate Scheduler

Linear Scheduler with Warmup

핵심:

“LR을 Linear하게 증가 시켰다가 Linear하게 감소시키는 것”

ACADENTIAL

Learning Rate Scheduler

Linear Scheduler with Warmup

핵심:

“LR을 Linear하게 증가 시켰다가 Linear하게 감소시키는 것”

Parameters: **Warmup steps**

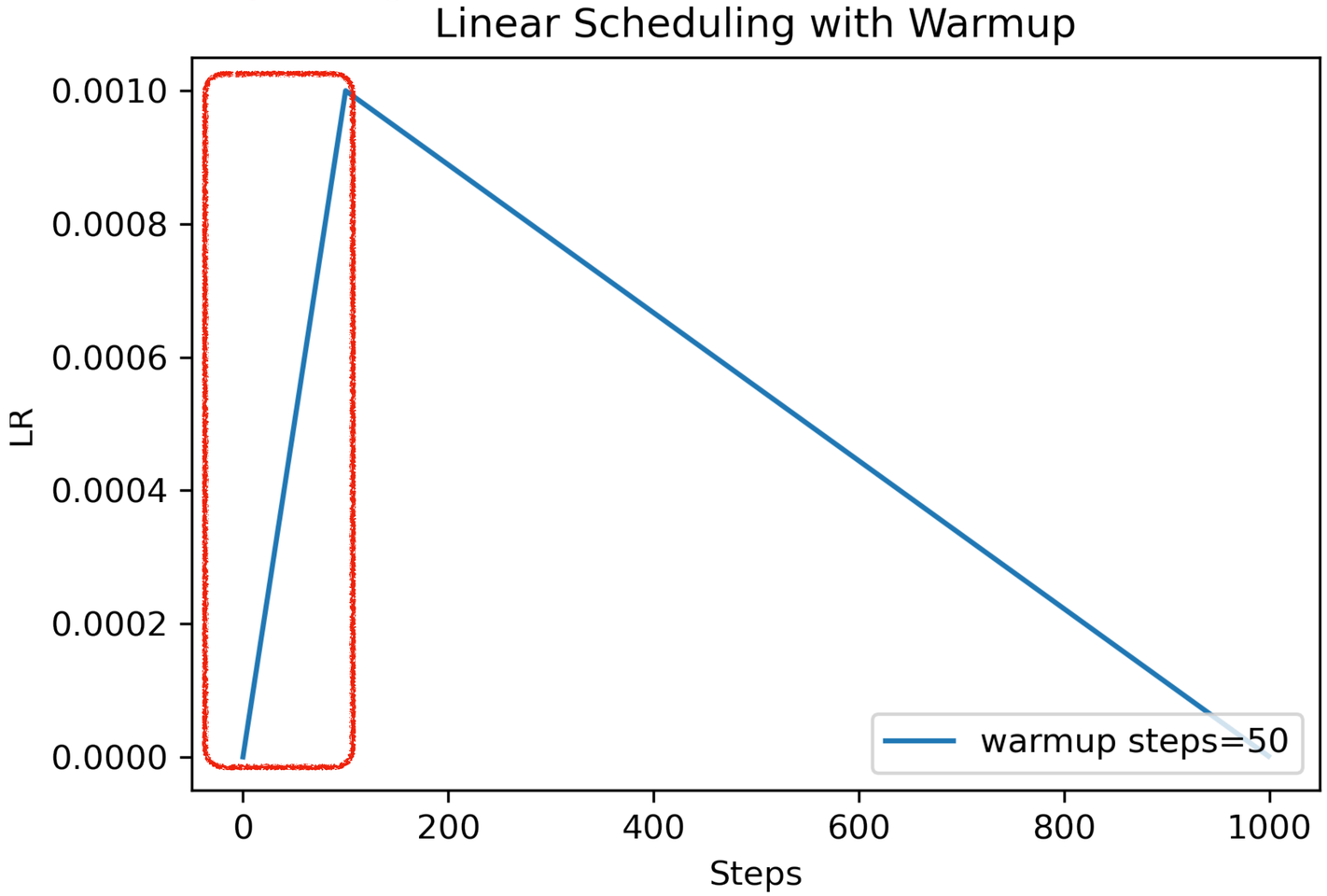
- Warmup 구간에 대해서 Linear하게 LR을 0에서부터 initial LR까지 증가시켰다가 Linear하게 LR을 줄이는 것.
- BERT의 학습에 사용됨.

ACADENTIAL

Learning Rate Scheduler

Linear Scheduler with Warmup

Warmup Stage



Parameters: **Warmup steps**

- Warmup 구간에 대해서 Linear하게 LR을 0에서부터 initial LR까지 증가시켰다가 Linear하게 LR을 줄이는 것.
- BERT의 학습에 사용됨.

transformers.get_linear_schedule_with_warmup

ACADENTIAL

Learning Rate Scheduler

Cosine Scheduler with Warmup

핵심:

“LR을 Linear하게 증가 시켰다가 Cosine 함수로 감소시키는 것”

ACADENTIAL

Learning Rate Scheduler

Cosine Scheduler with Warmup

핵심:

“LR을 Linear하게 증가 시켰다가 Cosine 함수로 감소시키는 것”

Parameters: **Warmup steps**

- Warmup 구간에 대해서 Linear하게 LR을0에서부터 initial LR까지 증가시켰다가 Cosine 함수로 감소.

ACADENTIAL

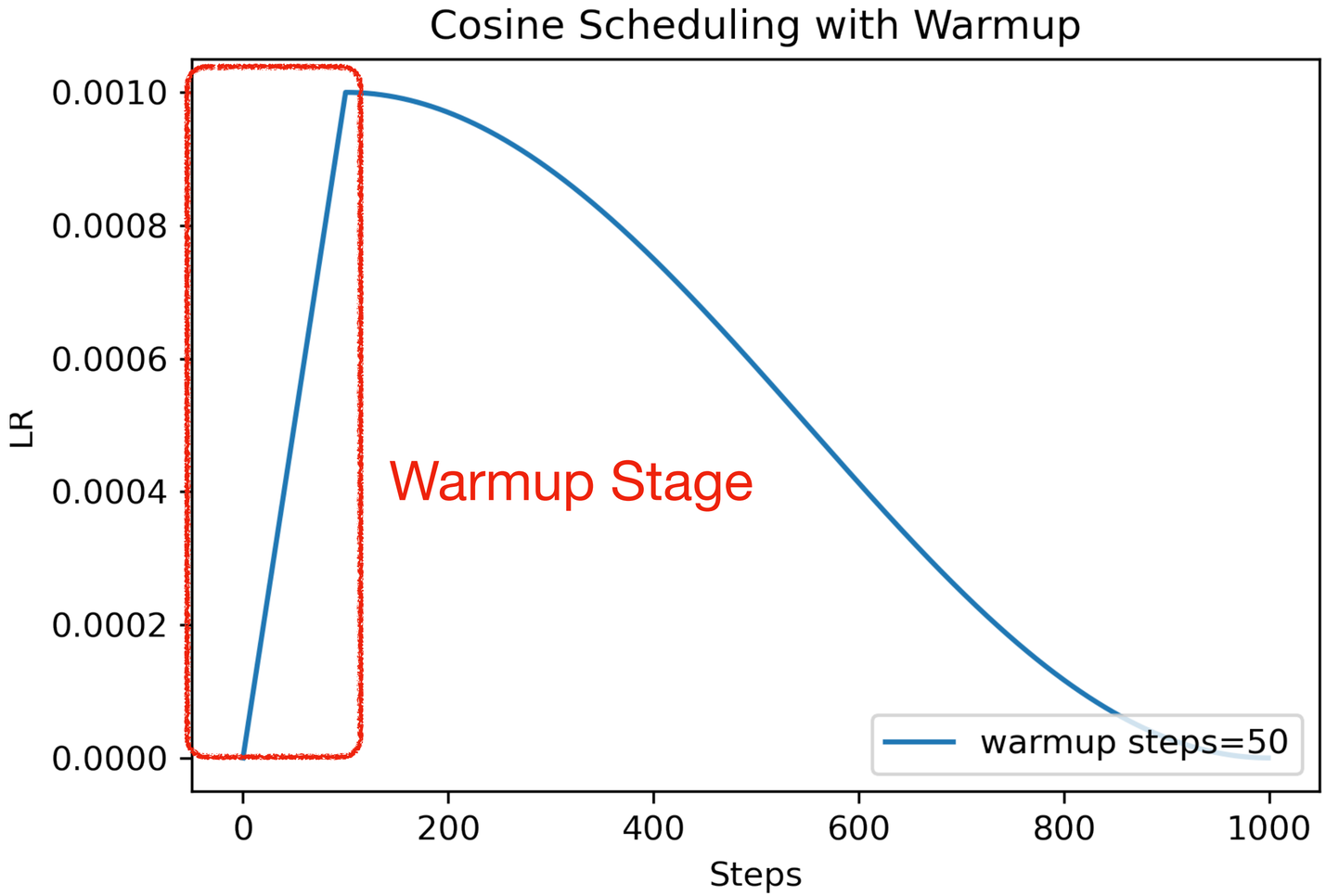
Learning Rate Scheduler

Cosine Scheduler with Warmup

Parameters: **Warmup steps**

- Warmup 구간에 대해서 Linear하게 LR을0에서부터 initial LR까지 증가시켰다가 Cosine 함수로 감소.

transformers.get_cosine_schedule_with_warmup



ACADENTIAL

Learning Rate Scheduler

Linear & Cosine Scheduler with Warmup

이유 / Motivation:

- **Warmup 단계의 효과:**
  - Learning Rate을 초기에 부드럽게 증가 → **안정적인 학습**
- **Warmup 끝난 직후:**
  - 높은 학습률 → Local minimum에서 벗어나 **더 나은 Minimum 탐색**
- **학습률 감소:**
  - 모델이 수렴하는데 도움

ACADENTIAL

11-7.
PyTorch로 구현해보는 LR Scheduler

ACADENTIAL

PyTorch로 구현해보는 LR Scheduler

Overview

- Pytorch에서는 `torch.optim.lr_scheduler` library을 통해서 다양한 LR scheduler을 사용할 수 있다.
- LR scheduler의 기본 뼈대를 살펴보자

ACADENTIAL

PyTorch로 구현해보는 LR Scheduler

Overview

LR scheduler의 Initialization

- Pytorch의 LR scheduler들은 **모두 공통적으로** `optimizer`을 첫번째 input으로 받는다.

(참고 사항)

LR scheduler의 종류에 따라 다른 input argument들(e.g. step_size, gamma 등등)을 추가로 필요로 한다.

```python
import torch
from torch.optim import lr_scheduler
from torch.optim import SGD

optimizer = SGD(model.parameters(), lr=1e-4)
scheduler = lr_scheduler.StepLR(
    optimizer,
    step_size=30,
    gamma=0.1
)
tbar = tqdm(dataset)
for epoch in tbar:
    train(...)
    validate(...)
    scheduler.step()
```

LR scheduler의 initialization

ACADENTIAL

PyTorch로 구현해보는 LR Scheduler

Overview

LR scheduler의 step

- LR scheduler의 `step()` 함수로 learning rate을 조정.

```python
import torch
from torch.optim import lr_scheduler
from torch.optim import SGD

optimizer = SGD(model.parameters(), lr=1e-4)
scheduler = lr_scheduler.StepLR(
    optimizer,
    step_size=30,
    gamma=0.1
)
tbar = tqdm(dataset)
for epoch in tbar:
    train(...)
    validate(...)
    scheduler.step()
```

LR scheduler의 step

ACADENTIAL

Let’s look at Jupyter Notebook!

ACADENTIAL

11-8. Section 11 요약

ACADENTIAL

Section Summary

Learning rate scheduler

Learning Rate Scheduler의 종류

1. Step LR
2. Exponential LR
3. Cosine Annealing with Warm Restarts
4. Reduce on Plateau LR
5. Linear Scheduler with Warmup
6. Cosine Schedule with Warmup

ACADENTIAL

Section Summary

Learning rate scheduler

Learning Rate Scheduler의 종류

1. **Step LR**: “계단식으로 step-size iteration 마다 LR을 줄이는 것”
2. **Exponential LR**: “Exponential하게 LR을 줄이는 것”
3. **Cosine Annealing LR**: “Cosine 함수로 LR을 조절하는 것”
4. **Cosine Annealing with Warm Restarts**: “Cosine 함수로 줄어다가 다시 Max LR로 주기적으로 Restart하는 것”
5. **Reduce on Plateau LR**: “Loss가 줄어들지 않을때 LR을 조금씩 줄이는 것”
6. **Linear Scheduler with Warmup**: “LR을 Linear하게 증가 시켰다가 Linear하게 감소시키는 것”
7. **Cosine Schedule with Warmup**: “LR을 Linear하게 증가 시켰다가 Cosine 함수로 감소시키는 것”