Team 8

2023-06-16

실전기계학습 기말 프로젝트

2023 Spring KHU Competition

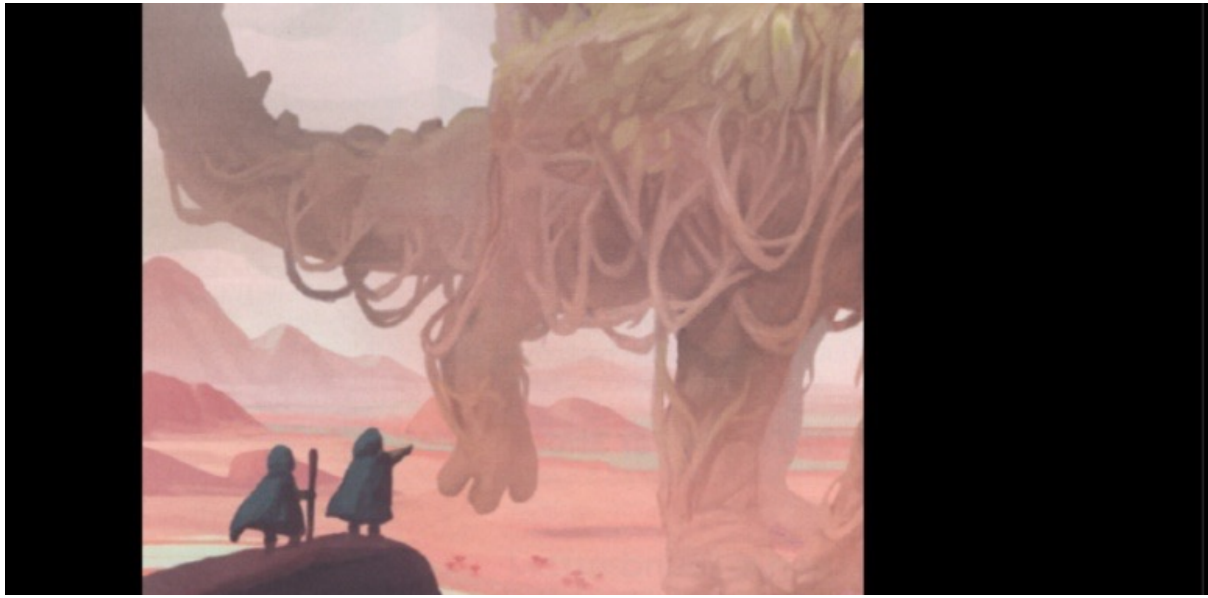
강민수 인공지능학과 2022105764
신정안 인공지능학과 2022105782
이혁승 인공지능학과 2022105793

Contents

01
문제인식

- 데이터 변환
- 모델 아키텍쳐

문제인식 - 데이터 변환



Normalization을 하기 전

Normalization을 한 후

[Image Restoration - CNN]

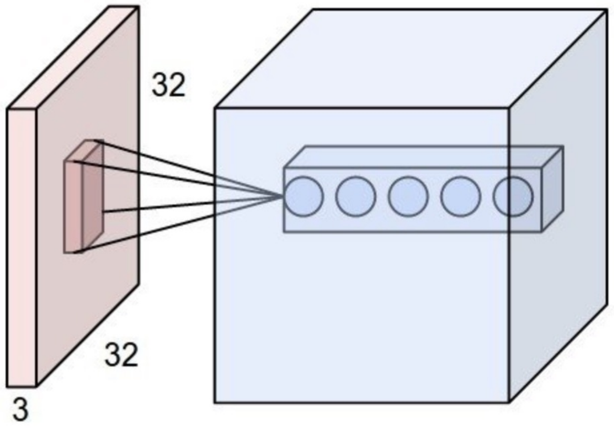
장점

Large-scale dataset로부터 generalizable image prior를 학습하는데 탁월

단점

Limited receptive field

Static weights at inference



02
데이터 변환

- Baseline에서 Data가 어떻게 처리되는가?

Baseline에서 Data가 어떻게 처리되는가?

1.Image Load Function

Image Data를 BGR채널에서 RGB채널로 변경
Pixel값 범위: 0~255

2.Torchvision.transforms 적용

2-1. ToTensor()
넘파이배열을 텐서로 변경
Pixel값 범위: 0~1

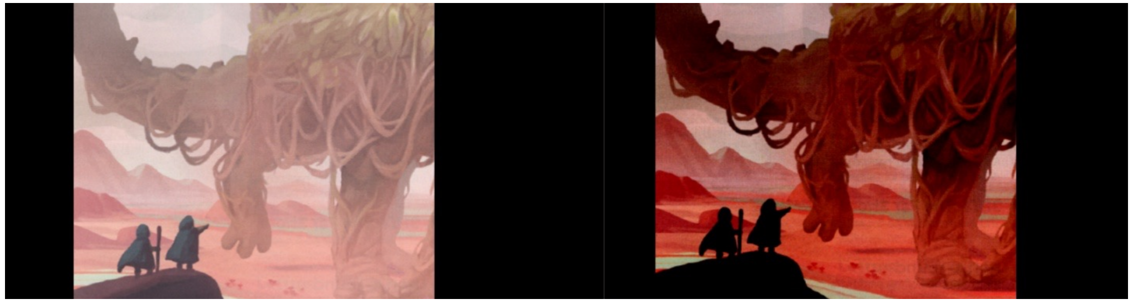
2-2. Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])
각 픽셀에 대해 평균을 빼고 표준편차로 나누어주는 정규화 진행
Pixel값 범위: -1~1

3. Model 학습

Pixel값의 범위가 **-1 ~ 1**인 Data에 대해 Model이 학습

4.Clamping(0,1) 및 PILImage로 변환

0보다 작은 값은 0, 1보다 큰 값은 1로 매핑 후
PILImage로 변환



왜 Pixel의 Detail이 사라지고 검정색으로 바뀌는가?

-1 ~1 범위의 값을 **0~1**로 클램핑 하니, **-1~0 범위의 Pixel값이 0으로 매핑**되어
Detail 무시 및 검정색으로 보이게 됨

03
모델 아키텍쳐

- 기존 CNN & Transformer
- MDTA
- GDFN

Restormer: Efficient Transformer for High-Resolution Image Restoration

Syed Waqas Zamir[1] Aditya Arora[1] Salman Khan[2] Munawar Hayat[3]
Fahad Shahbaz Khan[2] Ming-Hsuan Yang[4,5,6]
[1]Inception Institute of AI [2]Mohamed bin Zayed University of AI [3]Monash University
[4]University of California, Merced [5]Yonsei University [6]Google Research

Abstract

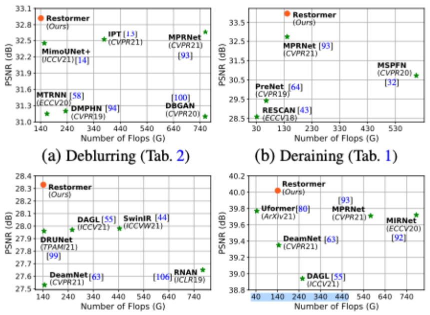
Since convolutional neural networks (CNNs) perform well at learning generalizable image priors from large-scale data, these models have been extensively applied to image restoration and related tasks. Recently, another class of neural architectures, Transformers, have shown significant performance gains on natural language and high-level vision tasks. While the Transformer model mitigates the shortcomings of CNNs (i.e., limited receptive field and inadaptability to input content), its computational complexity grows quadratically with the spatial resolution, therefore making it infeasible to apply to most image restoration tasks involving high-resolution images. In this work, we propose an efficient Transformer model by making several key designs in the building blocks (multi-head attention and feed-forward network) such that it can capture long-range pixel interactions, while still remaining applicable to large images. Our model, named Restoration Transformer (Restormer), achieves state-of-the-art results on several image restoration tasks, including image deraining, single-image motion deblurring, defocus deblurring (single-image and dual-pixel data), and image denoising (Gaussian grayscale/color denoising, and real image denoising). The source code and pre-trained models are available at https://github.com/swz30/Restormer.

1. Introduction

Image restoration is the task of reconstructing a high-quality image by removing degradations (*e.g.*, noise, blur, rain drops) from a degraded input. Due to the ill-posed nature, it is a highly challenging problem that usually requires strong image priors for effective restoration. Since convolutional neural networks (CNNs) perform well at learning generalizable priors from large-scale data, they have emerged as a preferable choice compared to conventional restoration approaches.

The basic operation in CNNs is the '*convolution*' that provides local connectivity and translation equivariance. While these properties bring efficiency and generalization to CNNs, they also cause two main issues. **(a)** The convolution operator has a limited receptive field, thus preventing it from modeling long-range pixel dependencies. **(b)** The convolution filters have static weights at inference, and thereby cannot flexibly adapt to the input content. To deal with the above-mentioned shortcomings, a more powerful and dynamic alternative is the *self-attention (SA)* mechanism [17, 77, 79, 95] that calculates response at a given pixel by a weighted sum of all other positions.

Self-attention is a core component in Transformer models [34, 77] but with a unique implementation, *i.e.*, *multi-head SA* that is optimized for parallelization and effective representation learning. Transformers have shown state-of-the-art performance on natural language tasks [10, 19, 49, 62] and on high-level vision problems [11, 17, 76, 78]. Although SA is highly effective in capturing long-range pixel interactions, its complexity grows quadratically with the spatial resolution, therefore making it infeasible to apply to high-resolution images (a frequent case in image restoration). Recently, few efforts have been made to tailor Transformers for image restoration tasks [13, 44, 80]. To reduce the computational loads, these methods either apply SA on small spatial windows of size 8×8 around each pixel [44, 80], or



(a) Deblurring (Tab. 2) (b) Deraining (Tab. 1)
(c) Gaussian Denoising (Tab. 4) (d) Real Denoising (Tab. 6)

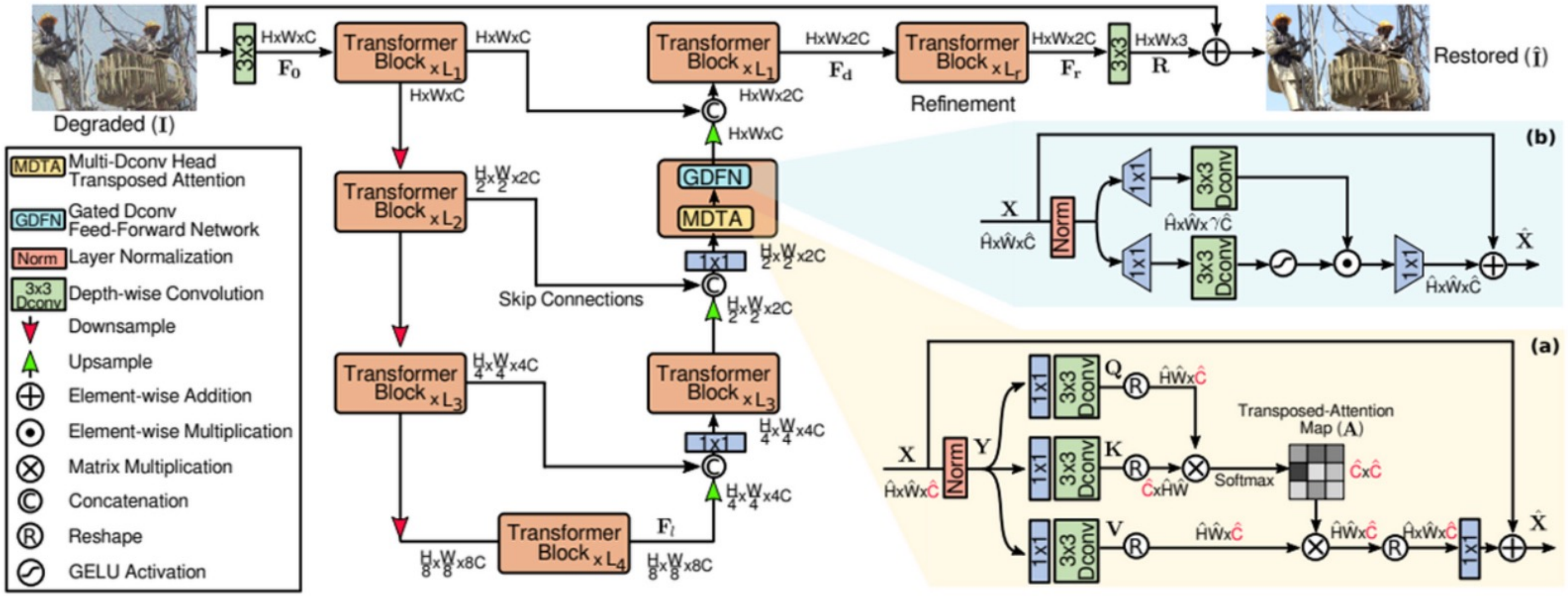
Figure 1. Our Restormer achieves the state-of-the-art performance on image restoration tasks while being computationally efficient.

Transformer가 비전분야에 정착하며 CNN기반
모델들의 문제 'Receptive Field'문제를 어느정도 해결

그러나 이미지 크기의 증가에 따라 연산부하 발생

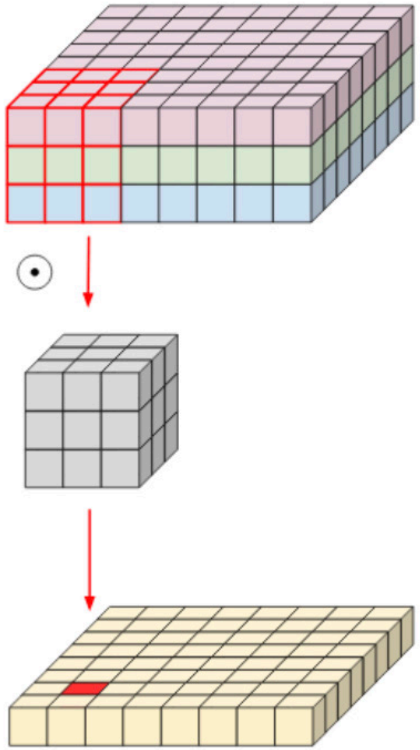
Restormer 연산부담을 최대한으로 낮추면서 성능 유지

모델 아키텍처

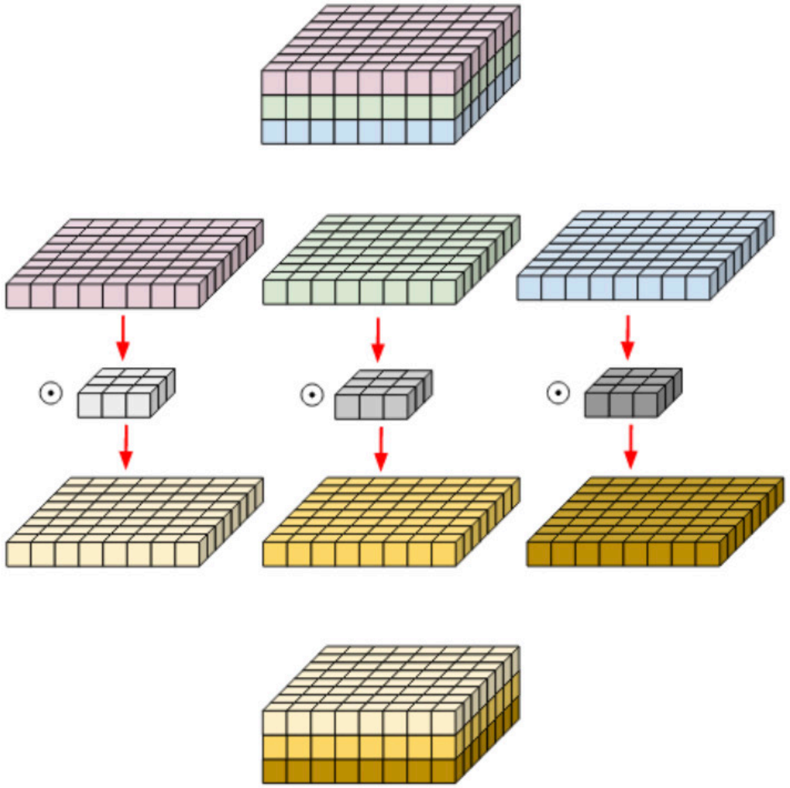


Decoder 중 Transformer Block 하나를 MDTA 와 GDFN로 구현한 Self-Attention Block으로 대체 한것이 Restormer의 핵심

모델 아키텍쳐 MDTA

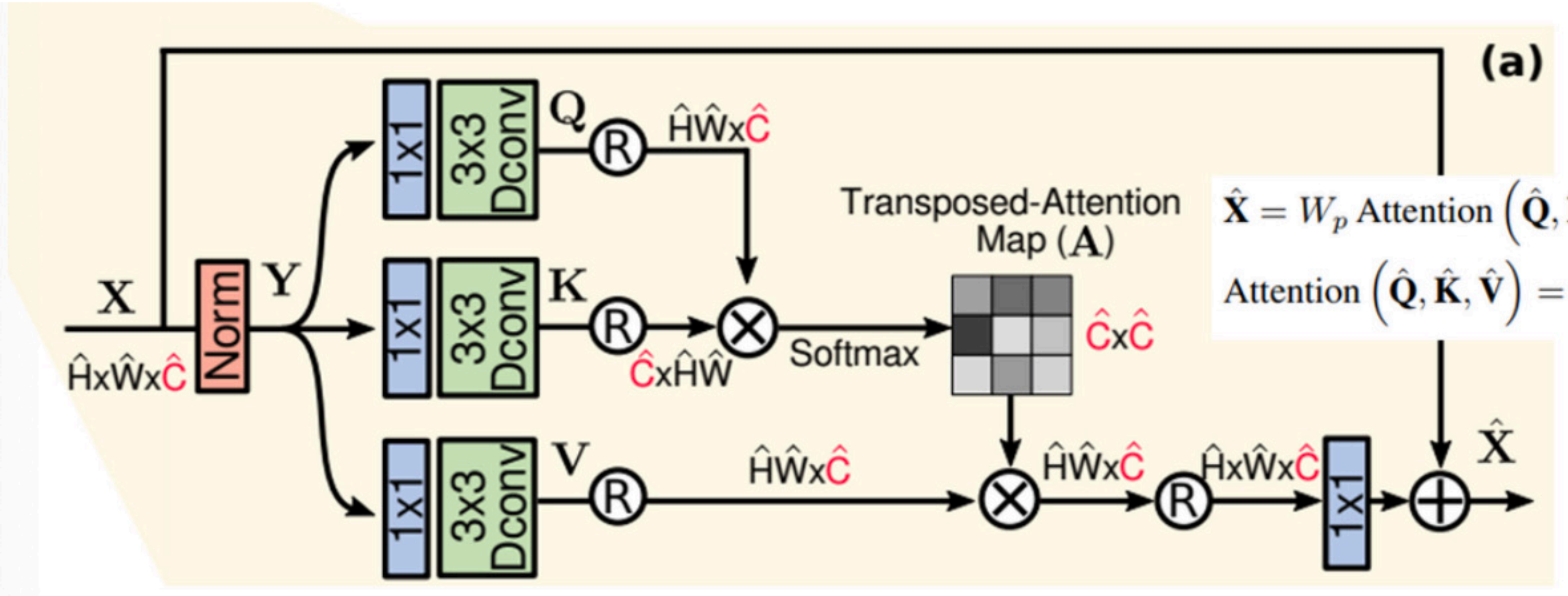


기존 Convolution



Depth-wise Convolution

모델 아키텍쳐 MDTA

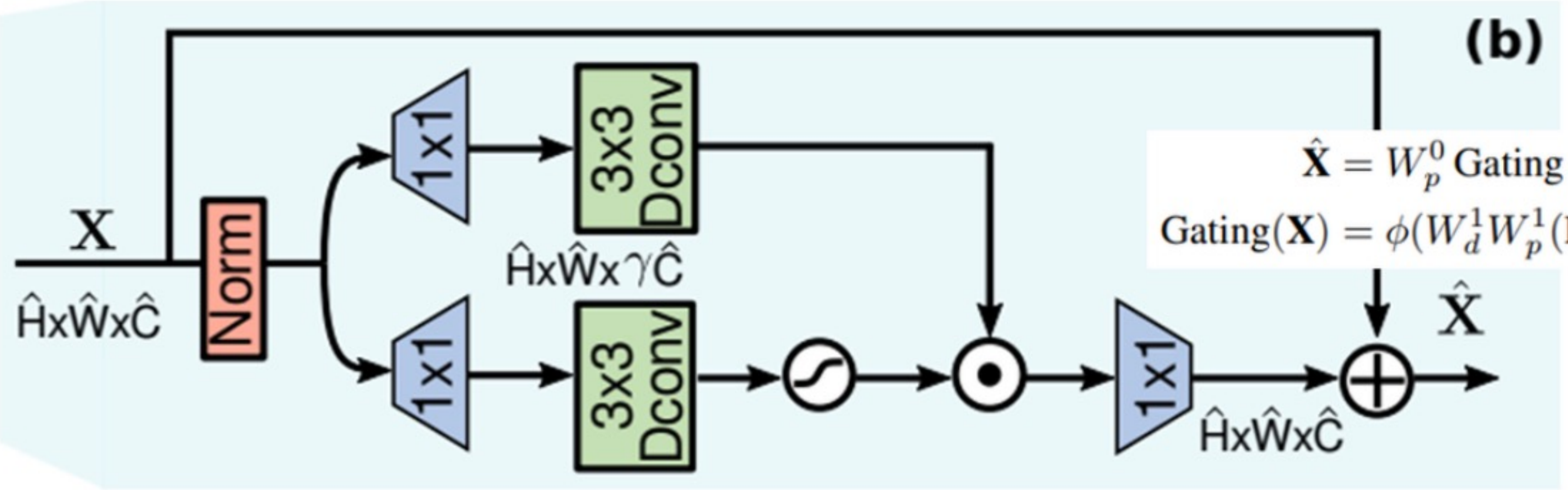


$$\hat{\mathbf{X}} = W_p \,\text{Attention}\left(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\right) + \mathbf{X},$$

$$\text{Attention}\left(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}\right) = \hat{\mathbf{V}} \cdot \text{Softmax}\left(\hat{\mathbf{K}} \cdot \hat{\mathbf{Q}}/\alpha\right)$$

적은 연산량으로도 long-range pixel dependencies를 활용한 self-attention 구현
기존 CNN의 장점인 Local한 공간 정보 또한 얻음

모델 아키텍쳐 GDFN



$$\hat{\mathbf{X}} = W_p^0 \, \text{Gating}(\mathbf{X}) + \mathbf{X},$$

$$\text{Gating}(\mathbf{X}) = \phi(W_d^1 W_p^1(\text{LN}(\mathbf{X}))) \odot W_d^2 W_p^2(\text{LN}(\mathbf{X}))$$

Encoder - Decoder pipeline의 각 level (H, W는 감소하고 C는 증가하면서
네트워크가 깊어지는 것)이 다른 level과 보완하여 task에 적합한 정보에 집중

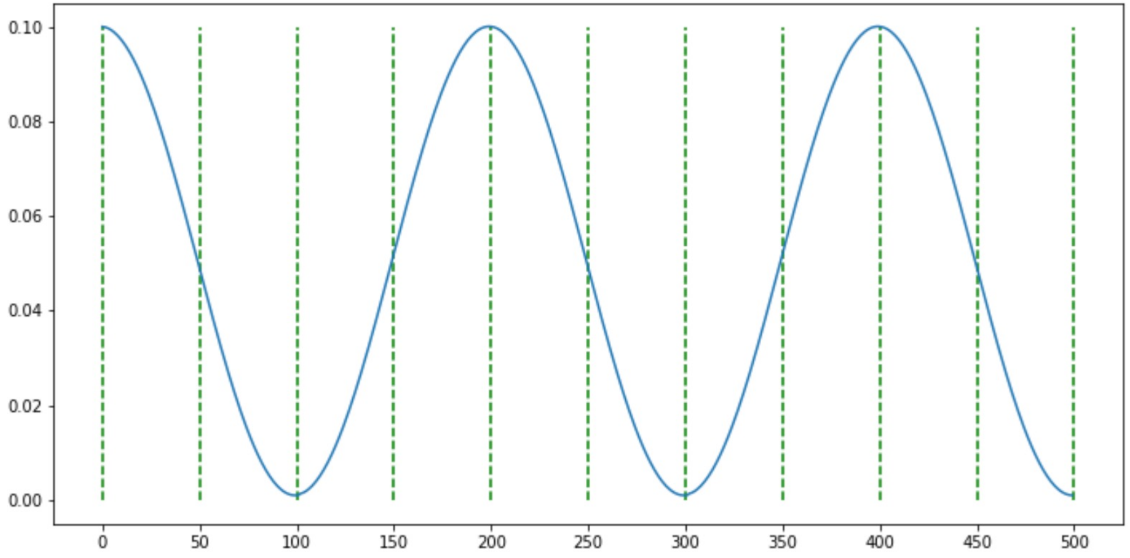
04

학습 기법

- Learning Rate Schedular
- Loss Function
- Optimizer

Learning Rate Scheduler : CosineAnnealingLR

EX) scheduler = CosineAnnealingLR(optimizer, T_max=100, eta_min=0.001)



Reference : https://gaussian37.github.io/

cosine 그래프를 그리면서 learning rate가 진동하는 방식

(범위: 초기설정값(최다)~eta_min(최소),주기:2*T_max)

안정적인 학습이 가능! Why?
CosineAnnealingLR은 학습률이 고정된 상태로 유지되지 않고, 학습률이 주기적으로 감소 및 증가시킴으로써 모델이 최적의 지점 주변에서 수렴이 더욱 안정적으로 이루어짐.

Loss Function: L1 norm? L2 norm?

L1 norm 400 epochs

L1 norm loss:0.0281
L2 norm loss: 0.0039

<<

L2 norm 400 epochs

L1 norm loss:0.0321
L2 norm loss:0.0043

위 결과를 토대로 **L1 Norm Loss** 선택

※ 비교가 가능하도록 L1 norm loss, L2 norm loss 모두 출력하도록함

Optimizer: AdamW

논문 환경 구현 후, 많은 epochs에는 SGD가 뛰어나다는 내용이 생각나 SGD와 비교해봄.

SGD(Stochastic Gradient Descent)

L1 norm Loss:0.0277

>>

AdamW(Adam Weight Decay)

L1 norm Loss: 0.0272

위 결과를 토대로 **AdamW** 선택!

※ 이전 장에서 L1 norm Loss가 성능이 더 좋아 L1 norm Loss를 기준으로 비교

05
결론

구현 정보

```
optimizer = optim.AdamW(model.parameters(), lr=learning_rate, weight_decay=1e-4)
criterion = nn.L1Loss()
scheduler = CosineAnnealingLR(optimizer, T_max=num_epochs)
```

600 epoch 기준
Kaggle 제출결과 Loss 2.46602

Restormer_test_600.csv Complete · Minnsu_03 · 17h ago	**2.46602**	**2.46602**

Ground Truth

Restormer Output

THANK YOU

Any Question?

강민수 인공지능학과 2022105764
신정안 인공지능학과 2022105782
이혁승 인공지능학과 2022105793