

# 뉴럴 스타일 트랜스퍼

## Neural Style Transfer

정우일



Content  
 $C$

+



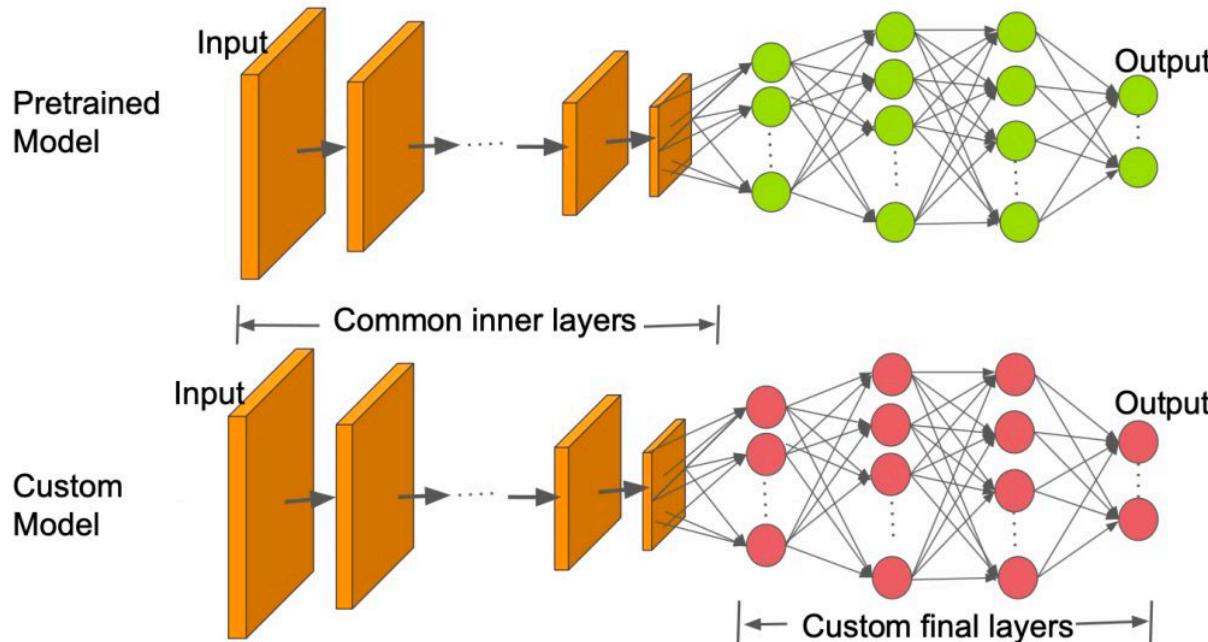
Style  
 $S$

=



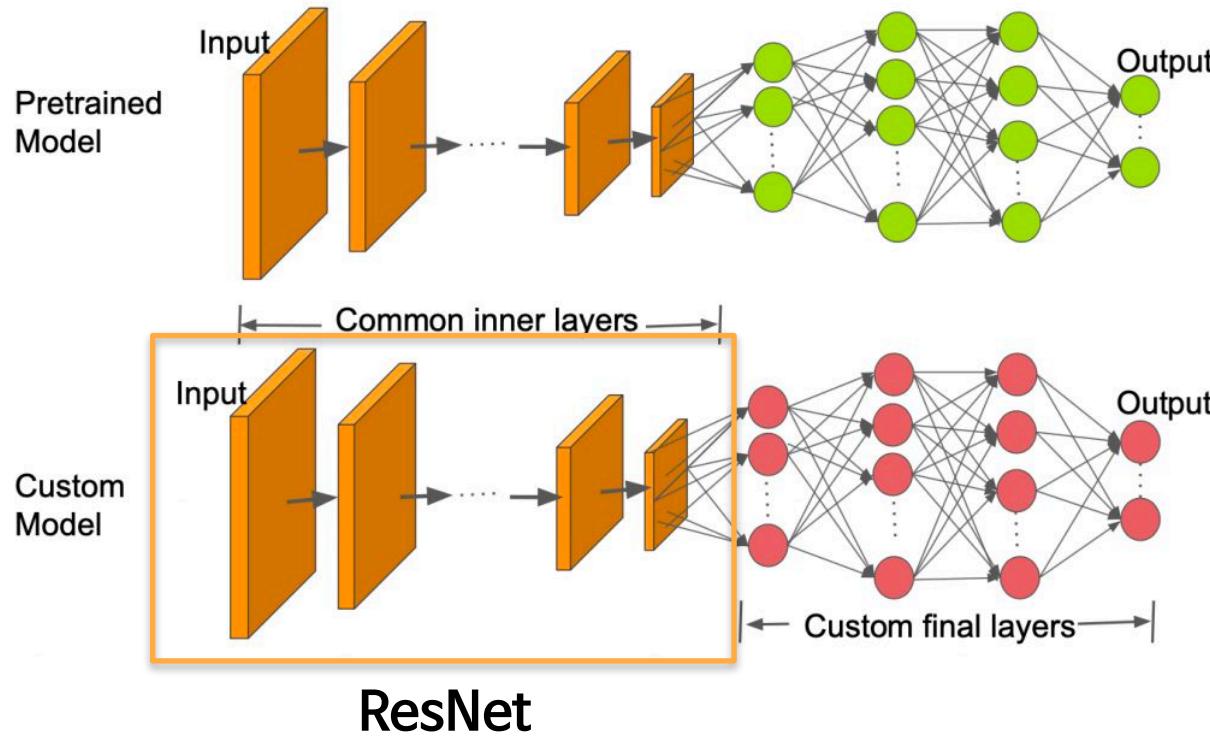
Generated Image  
 $G$

# 8.1 전이학습(Transfer Learning)

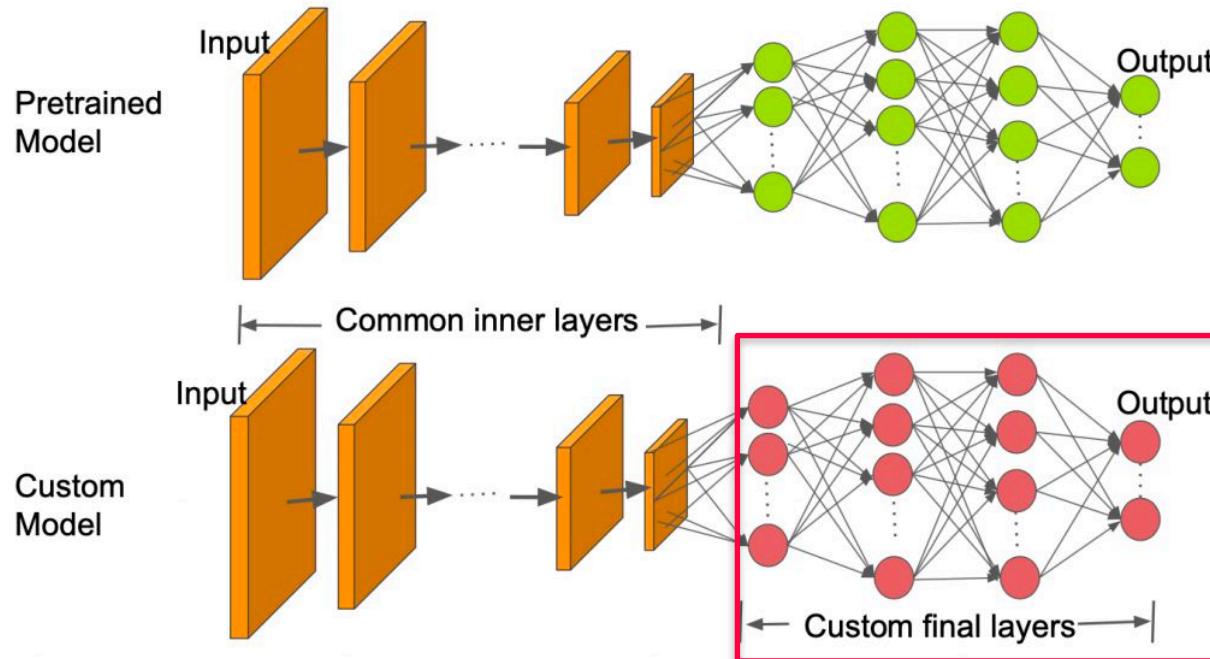


- 데이터 수집 및 연산 비용 절감
- 학습 소요 시간 절감
- 현실 적용 시 안전성 확보(ex. 자율주행자동차 시뮬레이션 환경)

# 8.1 전이학습(Transfer Learning)

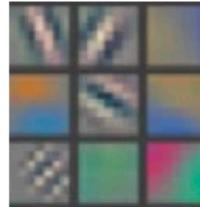


# 8.1 전이학습(Transfer Learning)

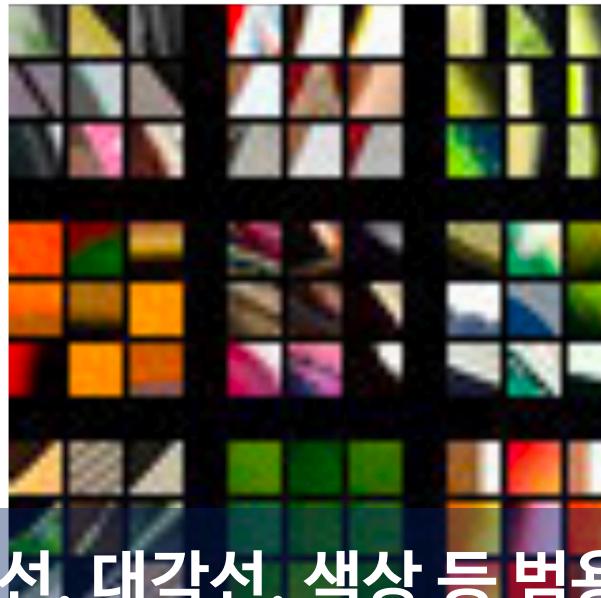


이 부분만 학습

## 8.2 스타일 트랜스퍼(Style Transfer)

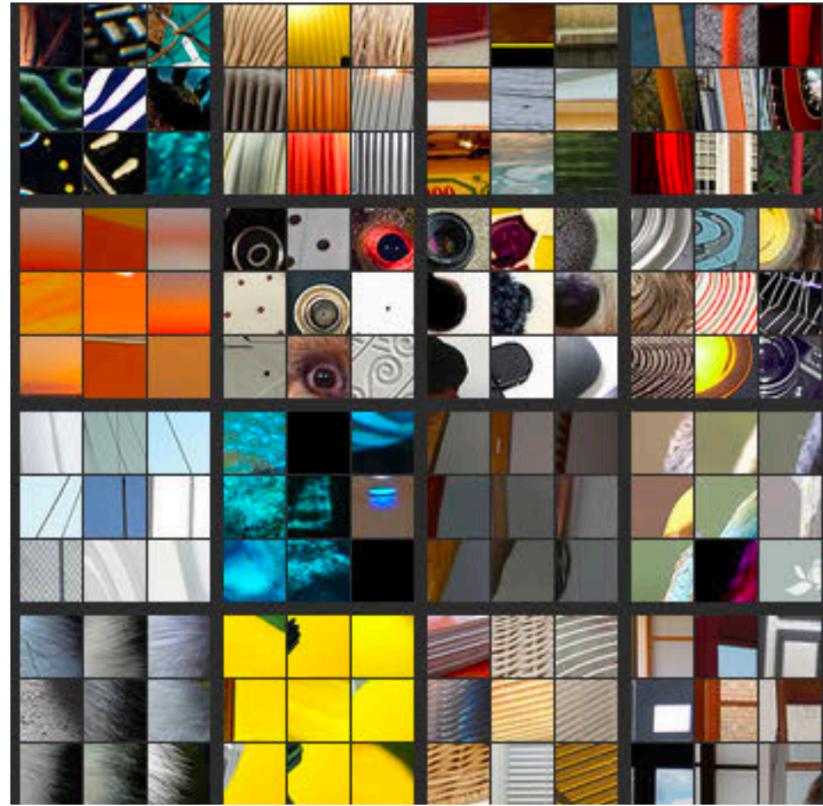
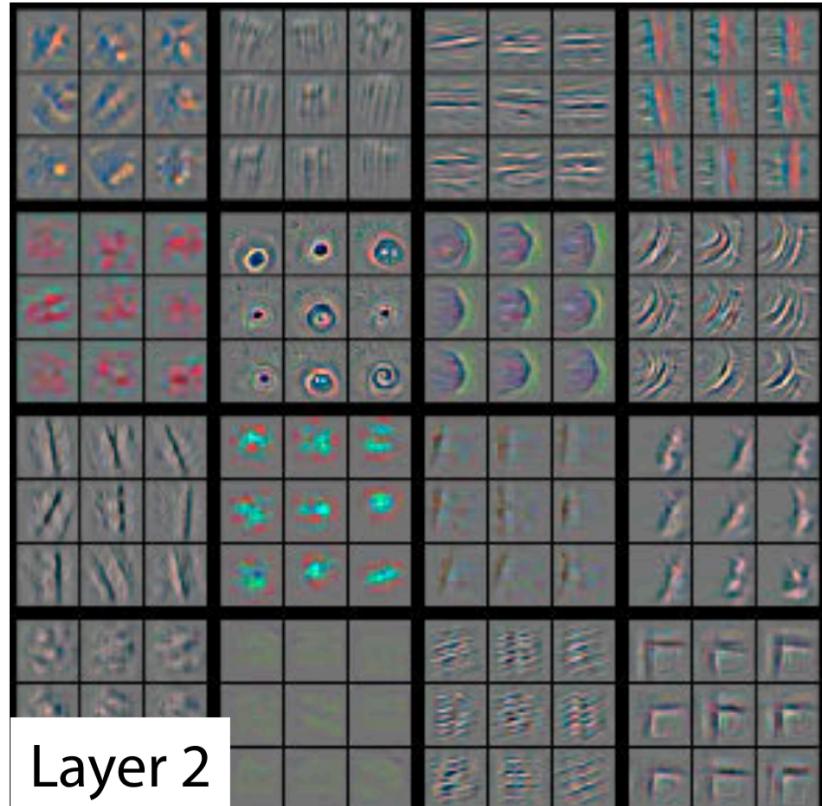


Layer 1

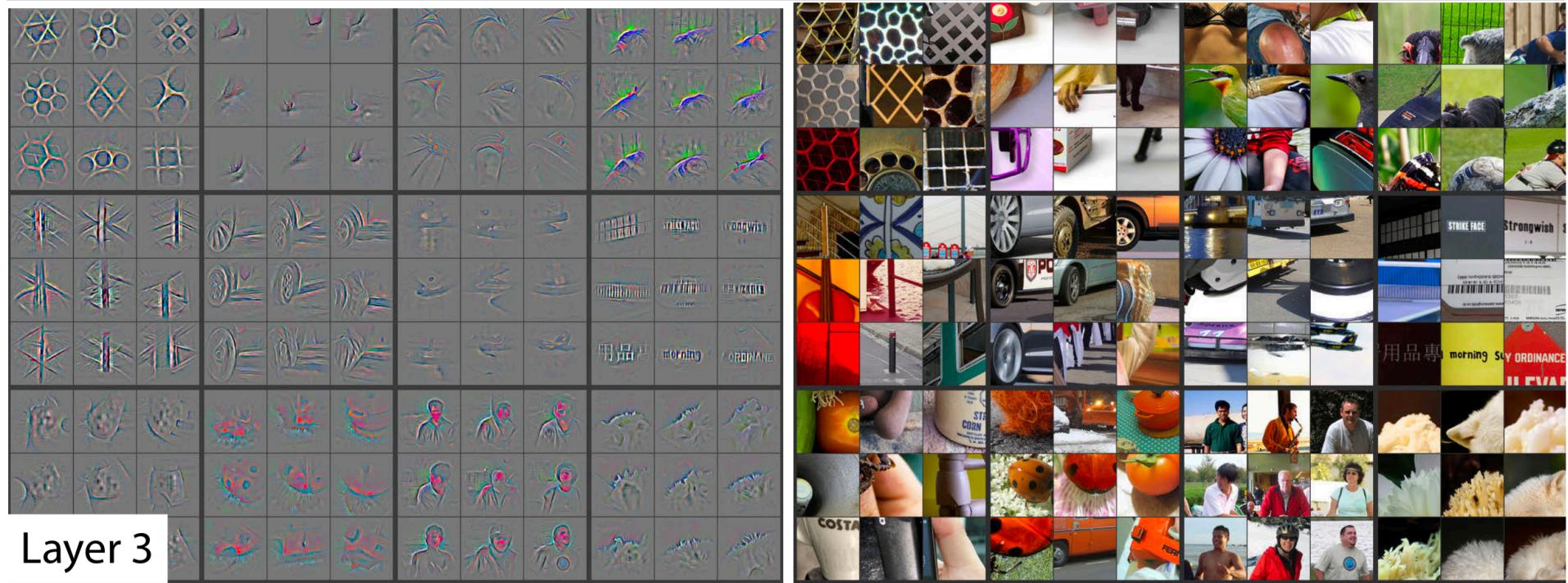


가로선, 세로선, 대각선, 색상 등 범용적인 필터들

## 8.2 스타일 트랜스퍼 (Style Transfer)



# 8.2 스타일 트랜스퍼 (Style Transfer)



## 8.2 스타일 트랜스퍼(Style Transfer)



Layer 4

Layer 5

## 8.3 스타일과 콘텐츠의 정의

### Style

Correlations between the different filter responses

### Content representation

Feature responses in higher layers of the network

# 8.3 스타일과 콘텐츠의 정의

## Style

Correlations between the different filter responses

... ?

## Content representation

Feature responses in higher layers of the network

## 8.4 학습 알고리즘

$$L_{Total} = \alpha L_{Content} + \beta L_{Style}$$

C와 G 사이의 내용 유사도 측정      S와 G 사이의 스타일 유사도 측정

내용 비용의 가중치      스타일 비용의 가중치

## 8.4 학습 알고리즘

$$L_{Total} = \alpha L_{Content} + \beta L_{Style}$$

C와 G 사이의 내용 유사도 측정      S와 G 사이의 스타일 유사도 측정

내용 비용의 가중치      스타일 비용의 가중치

Gradient  
Descent

$$G := G - \frac{\partial}{\partial G} L_{Total}$$

## 8.4 학습 알고리즘

$$L_{Total} = \alpha L_{Content} + \beta L_{Style}$$

- 내용 비율 계산을 위해 온닉층  $k$ 를 사용한다고 가정  
만약,  $k$ 가 아주 작은 값 즉, 1이라면 내용 이미지의 픽셀값과 아주 유사한 이미지를 생성하도록 함.  
반대로,  $k$ 가 아주 큰 값이라면 내용 이미지에 고양이가 있을 때, 생성 이미지의 어딘가에 고양이가 있도록 함.  
(보통  $k$ 는 신경망의 중간층에서 선택)
- 사전학습 된 ConvNet (e.g. ResNet) 사용

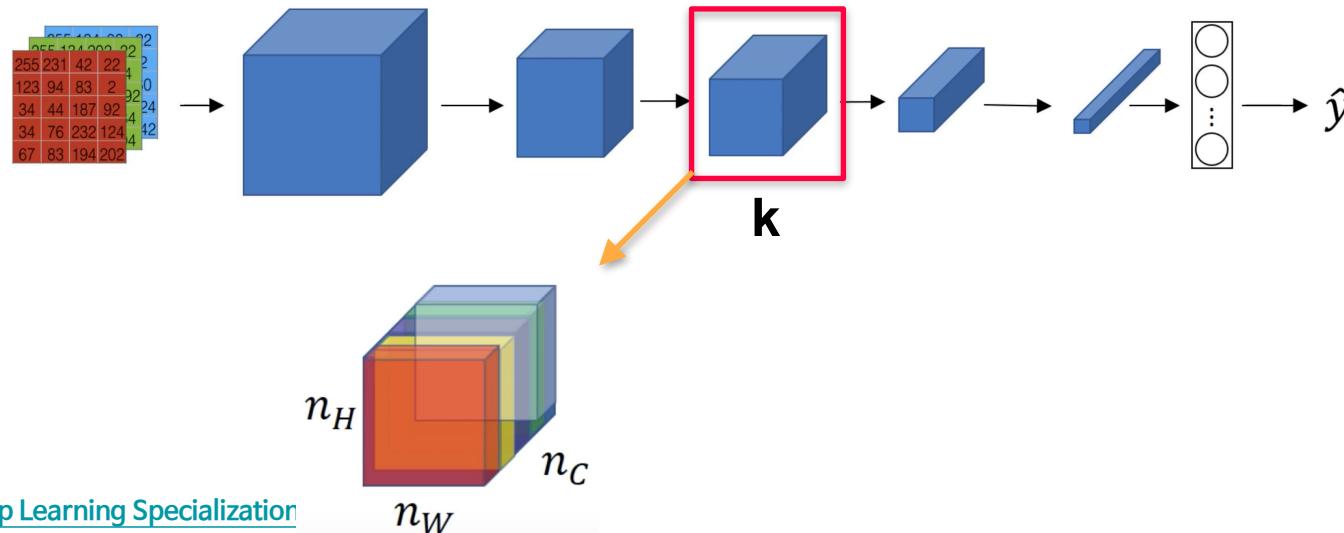
$a^{[k](C)}, a^{[k](G)}$       - 이미지를 C, G의  $k$ 층 활성화 값으로 두 이미지가 내용면에서 얼마나 비슷한지 측정

$$L_{Content} = \frac{1}{2} ||a^{[k](C)} - a^{[k](G)}||^2$$
      -  $k$ 층의 각 이미지 C, G에 대한 활성화값 차이의 제곱합

## 8.4 학습 알고리즘

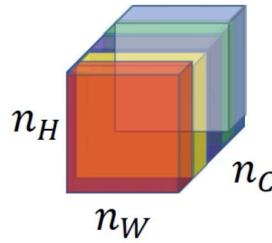
$$L_{Total} = \alpha L_{Content} + \beta L_{Style}$$

- 스타일 비용 계산을 위해 은닉층  $k$ 를 사용한다고 가정
- 스타일 :  $k$ 층에서의 서로 다른 채널들의 활성화 값 사이의 상관관계

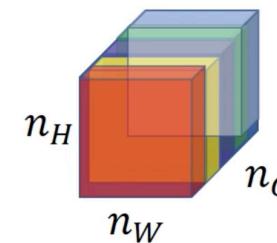


# 8.4 학습 알고리즘

Style image

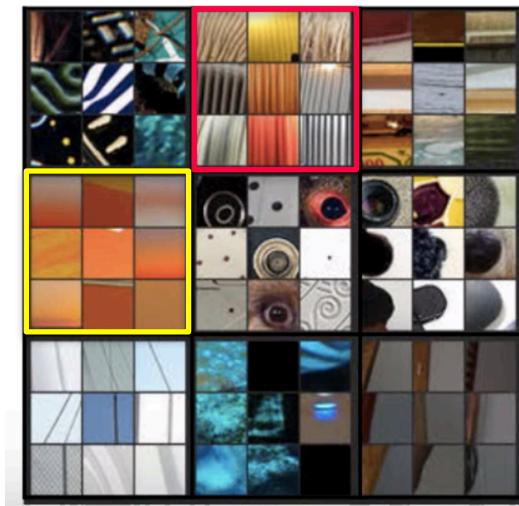


Generated Image



세로선

주황색



이 두 채널 사이의 상관관계가 높다면  
이미지 조각이 세로선을 가질 때,  
보통 주황색을 띤다고 볼 수 있음

## 8.4 학습 알고리즘

### 스타일 행렬(그람 행렬)

$a_{i,j,k}^{[l]}$  은 닉층  $l$ 의 위치  $i$ (높이),  $j$ (너비),  $k$ (채널)에 대한 활성화 값

$G^{[l]}$   $n_c \times n_c$  차원을 갖는 은닉층  $l$ 의 스타일 행렬(높이와 너비가 각각 채널의 수인 행렬)

$$G_{kk'}^{[l](S)} = \sum_i^{n_H} \sum_j^{n_W} a_{ijk}^{[l](S)} a_{ijk'}^{[l](S)}$$

스타일 이미지에 대한  $l$ 층의  $k$ 와  $k'$  채널 사이의 상관관계를 측정

$$G_{kk'}^{[l](G)} = \sum_i^{n_H} \sum_j^{n_W} a_{ijk}^{[l](G)} a_{ijk'}^{[l](G)}$$

생성 이미지에 대한  $l$ 층의  $k$ 와  $k'$  채널 사이의 상관관계를 측정

\*행렬  $G$ 를 계산하기 위해 모든  $k$ 와  $k'$ 에 대해 이를 반복

## 8.4 학습 알고리즘

### 은닉층에서의 스타일 비용 함수 계산

$$L_{Style}^l(S, G) = \lambda ||G^{[l](S)} - G^{[l](G)}||_F^2 = \lambda \sum_k \sum_{k'} (G_{kk'}^{[l](S)} - G_{kk'}^{[l](G)})^2$$

## 8.4 학습 알고리즘

$10^{-4}$



$10^{-3}$



$10^{-2}$



$10^{-1}$



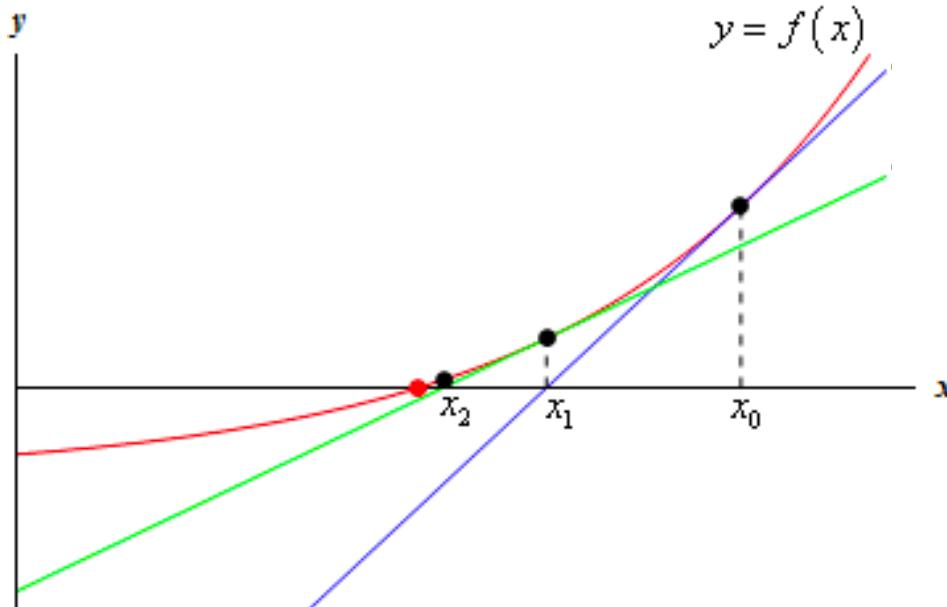
Content Loss 비중( $\alpha$ )을 높일 수록 Content 이미지와 비슷해짐

# 8.5 최적화 알고리즘

## 뉴턴 방법 (Newton Method)

미분가능한 함수  $f(x)$ 에 대하여 방정식  $f(x)=0$ 의 근삿값을 구하는 방법

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



## 8.6 코드 구현

- 전이학습 Colab
- 뉴럴 스타일 트랜스퍼 Colab

# Q&A

감사합니다.