

StyleGAN의 이미지 품질 분석 및 개선

테로 카라스
엔비디아

사무리 레인
엔비디아

야코 레티넨
NVIDIA와 알토 대학

미아카 아이탈라
엔비디아

안 헬스텐
엔비디아

티모 아일라
엔비디아

추상적인

스타일 기반 GAN 아키텍처(StyleGAN)는 최첨단 결과 데이터 기반 무조건 생성 이미지 모델링. 우리는 여러 가지를 노출하고 분석합니다. 특성 아티팩트 및 두 가지 모두의 변경 사항을 제안합니다. 이를 해결하기 위한 모델 아키텍처 및 교육 방법. 특히 생성기 정규화를 재설계하고 점진적 성장을 다시 방문하며 생성기를 다음과 같이 정규화합니다. 잠재적인 매핑에서 좋은 컨디셔닝을 권장합니다. 이미지에 코드. 화질을 향상시킬 뿐만 아니라, 이 경로 길이 정규화 프로그램은 다음과 같은 추가 이점을 제공합니다. 발전기를 뒤집기가 훨씬 쉬워집니다. 이것 생성된 이미지를 특정 네트워크. 우리는 또한 얼마나 잘 시각화 생성기는 출력 해상도를 활용하고 식별 용량 문제, 더 큰 모델을 학습하도록 동기 부여 추가 품질 개선. 전반적으로 개선된 모델은 무조건적인 이미지로 최신 기술을 재정의합니다. 기존 배포 품질 메트릭과 인지도 이미지 품질의 측면에서 모델링합니다.

1. 소개

생성적 방법, 특히 생성적 적대 네트워크에 의해 생성된 이미지의 해상도 및 품질 (GAN) [16], 빠르게 개선되고 있습니다 [23, 31, 5]. 현재 고해상도 이미지 합성을 위한 최첨단 방법 안정적으로 작동하는 것으로 입증된 StyleGAN [24] 입니다. 다양한 데이터셋에서 우리의 작업은 특성 아티팩트를 수정하고 결과 품질을 더욱 향상시키는 데 중점을 둡니다.

StyleGAN [24] 의 구별되는 특징은 기존의 생성기 아키텍처와 다릅니다. 먹이를 주는 대신 네트워크의 시작 부분에만 잠재 코드 $z \in Z$ 를 입력하면 매핑 네트워크 f 는 먼저 이를 중간 잠재 코드 $w \in W$ 로 변환합니다. Affine은 합성 네트워크의 레이어를 제어하는 스타일을 생성한 다음 변환합니다.

적응형 인스턴스 정규화(AdaIN)를 통해 [21, 9, 13, 8]. 또한 다음을 제공함으로써 확률적 변동이 촉진됩니다.

추가 랜덤 노이즈는 합성 네트워크에 매핑됩니다. 그것 이 설계가 _ 중간 잠재 공간 W 는 얹힘이 훨씬 적습니다. 입력 잠재 공간 Z . 이 논문에서는 W 에만 모든 분석을 집중합니다.

합성 네트워크의 관점.

많은 관찰자들은 다음에서 특징적인 인공물을 발견했습니다. StyleGAN [3]에 의해 생성된 이미지. 우리는 두 가지 원인을 식별합니다 이러한 아티팩트에 대해 설명하고 아키텍처 및 그들을 제거하는 훈련 방법. 첫째, 우리는 조사 일반적인 얼룩 같은 인공물의 기원, 그리고 제너레이터는 아키텍처의 설계 결함을 우회하기 위해 생성합니다. 섹션 2에서는 사용된 정규화를 재설계합니다. 생성기에서 아티팩트를 제거합니다. 둘째, 우리는 점진적 성장과 관련된 인공물을 분석한다 [23]. 고해상도 GAN 안정화에 큰 성공 훈련. 실현하는 대안적인 디자인을 제안합니다. 동일한 목표 — 저해상도에 초점을 맞추면서 교육을 시작합니다. 이미지를 촬영한 다음 점진적으로 초점을 더 높은 곳으로 이동합니다. 더 높은 해상도 — 훈련 중 네트워크 토폴로지 변경 없이. 이 새로운 디자인을 통해 생성된 이미지의 효과적인 해상도에 대해 추론할 수 있습니다.

이는 예상보다 낮은 것으로 밝혀져 용량 증가를 유발합니다(섹션 4).

생성된 이미지 품질의 정량적 분석

생성 방법을 사용하는 것은 계속해서 도전적입니다.

주제. FID(Frechet inception distance) [20] 는 InceptionV3 분류기 [39] 의 고차원 특징 공간에서 두 분포의 밀도 차이를 측정합니다.

Precision and Recall(P&R) [36, 27] 은 생성된 비율을 명시적으로 수량화하여 추가 가시성을 제공합니다.

훈련 데이터와 유사한 이미지 및 백분율

생성할 수 있는 훈련 데이터의 우리는 사용

개선 사항을 수량화하기 위해 이러한 메트릭을 사용합니다.

FID와 P&R은 모두 분류기 네트워크를 기반으로 합니다.

최근에는 질감보다는 질감에 중점을 두는 것으로 나타났습니다.

모양 [12], 결과적으로 측정항목이 정확하지 않습니다.

이미지 품질의 모든 측면을 캡처합니다. 우리는 관찰

PPL(Perceptual Path length) 메트릭 [24], 원래 잠재 공간의 품질을 추정하는 방법으로 도입됨



그림 1. 인스턴스 정규화로 인해 StyleGAN 이미지에서 물방울과 같은 아티팩트가 발생합니다. 생성된 파일에서 항상 명확한 것은 아닙니다. 그러나 생성기 네트워크 내부의 활성화를 보면 문제는 항상 존재합니다. 64x64 해상도. 모든 StyleGAN 이미지를 괴롭히는 시스템 문제입니다.

보간, 일관성 및 안정성과 상관 관계

모양. 이를 기반으로 합성 네트워크를 정규화합니다.

부드러운 매핑(섹션 3)을 선호하고 명확한

품질 향상. 계산 비용에 대응하기 위해 모든 정규화를 더 적게 실행할 것을 제안합니다.

종종 이것이 com의 약속된 효과 없이 수행될 수 있음을 관찰합니다.

마지막으로, 우리는 잠재된 이미지에 대한 투영이 space W는 원래 StyleGAN보다 경로 길이가 정규화된 새로운 StyleGAN2 생성기에서 훨씬 더 잘 작동합니다. 이렇게 하면 생성된 속성을 더 쉽게 이미지를 소스로 전송합니다(섹션 5).

구현 및 훈련된 모델은 다음에서 사용할 수 있습니다.
<https://github.com/NVLabs/stylegan2>

2. 정규화 아티팩트 제거

우리는 다음으로 생성된 대부분의 이미지를 관찰하는 것으로 시작합니다. StyleGAN은 특징적인 얼룩 모양의 인공물을 나타냅니다. 물방울을 닮았다. 그림 1과 같이 물방울은 최종 이미지에서 명확하지 않을 수 있습니다. 생성기의 중간 기능 맵에 있습니다.¹ 64×64 해상도 부근에서 이상 현상이 나타나기 시작하고, 모든 기능 맵에 존재하며 점진적으로 더 높은 해상도에서 더 강력합니다. 이러한 일관된 인공물의 존재는 식별기가 있어야 하기 때문에 당혹스럽습니다. 감지할 수 있습니다.

우리는 AdaIN 작업의 문제를 정확히 지적합니다. 각 기능 맵의 평균과 분산을 별도로 정규화하여 잠재적으로 발견된 모든 정보를 파괴합니다. 서로에 대한 기능의 크기. 우리 액적 인공물은 생성기가 의도적으로 과거의 신호 강도 정보를 은닉한 결과라고 가정합니다. 인스턴스 정규화: 강력하고 국지적인 스파이크 생성 통계를 지배하는 생성기는 효과적으로 신호를 다른 곳에서 원하는 대로 조정합니다. 우리의 가설은 정규화 단계가

아래에 자세히 설명된 것처럼 생성기에서 제거된 액적 유물이 완전히 사라집니다.

¹ 드문 경우지만(이미지의 0.1%) 물방울이 누락되어 심각하게 손상된 이미지. 자세한 내용은 부록 A를 참조하십시오.

2.1. 제너레이터 아키텍처 재검토

먼저 StyleGAN의 몇 가지 세부 사항을 수정할 것입니다. 생성기를 사용하여 재설계된 정규화를 보다 쉽게 수행할 수 있습니다. 이러한 변화는 중립적이거나 작은 긍정적인 영향을 미칩니다. 품질 메트릭 측면에서 자체적으로.

그림 2a는 원래 StyleGAN 합성 네트워크 g [24]를 보여주고 그림 2b에서는 다이어그램을 전체로 확장합니다. 가중치와 편향을 표시하고 두 가지 구성 요소에 대한 AdaIN 작업: 정규화 및 변조. 이를 통해 개념을 다시 그릴 수 있습니다. 각 상자가 네트워크의 일부를 나타내도록 회색 상자 여기서 하나의 스타일이 활성화됩니다(즉, "스타일 블록"). 재미있게, 원래 StyleGAN은 내부에 바이어스와 노이즈를 적용합니다. 스타일 블록으로 인해 상대적 영향이 현재 스타일의 크기에 반비례합니다. 우리는 그것을 관찰 이러한 작업을 스타일 블록 외부로 이동하여 더 예측 가능한 결과를 얻을 수 있습니다. 또한 이 변경 후

정규화 및 변조가 표준 편차에서만 작동하는 것으로 충분합니다(즉, 평균이 필요). 바이어스, 노이즈 및 정규화의 적응 상수 입력에 대한 또한 관찰 가능한 단점 없이 안전하게 제거될 수 있습니다. 이 변형은 그림 2c에 나와 있으며, 재설계된 정규화의 출발점 역할을 합니다.

2.2. 인스턴스 정규화 재검토

StyleGAN의 주요 장점 중 하나는 스타일 믹싱을 통해 생성된 이미지를 제어합니다. 즉, 추론 시간에 다른 레이어에 다른 잠재 w를 공급함으로써 실제로 스타일 변조는 특정 기능을 증폭할 수 있습니다. 크기 또는 그 이상의 순서로 지도를 작성합니다. 스타일 믹싱을 위해 작업, 우리는 명시 적으로 증폭에 대응해야 합니다 샘플당 기준 — 그렇지 않으면 후속 레이어는 의미 있는 방식으로 데이터를 조작할 수 없습니다. 규모별 통제를 기꺼이 희생한다면(참조 비디오), 우리는 단순히 정규화를 제거하여 아티팩트를 제거하고 FID를 약간 향상시킬 수 있습니다 [27]. 이제 우리는 제거하는 더 나은 대안을 제안할 것입니다. 완전한 제어 가능성을 유지하면서 아티팩트. 주요 생각 앞으로 나올 기능 맵의 예상 통계를 기반으로 정규화하되 명시적인 강제는 하지 않습니다.

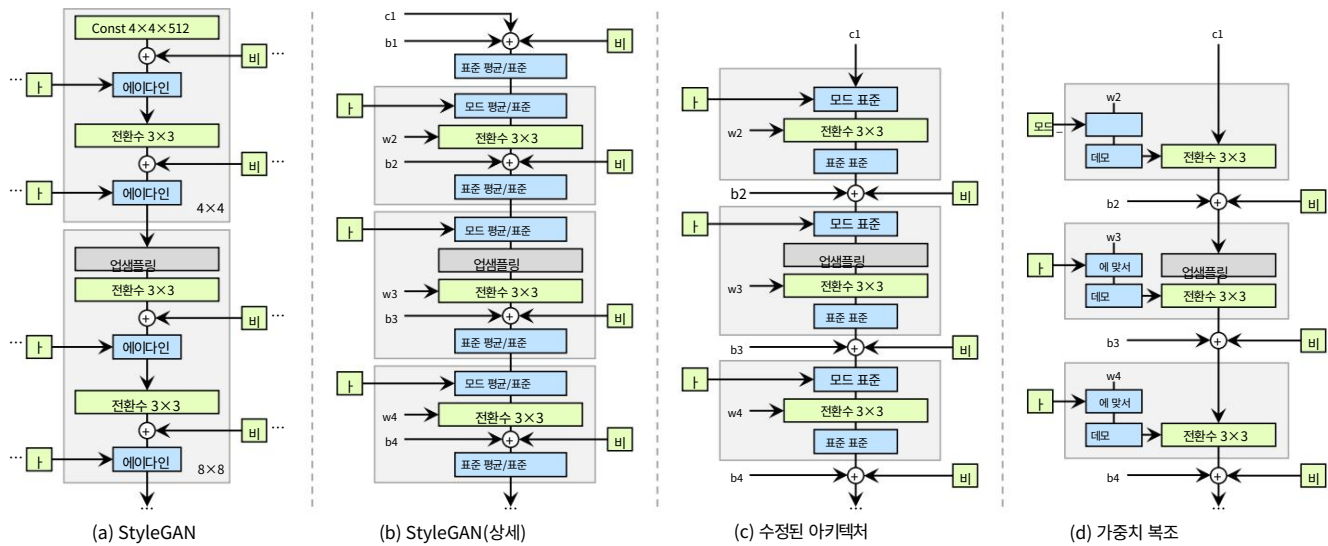


그림 2. StyleGAN 합성 네트워크의 아키텍처를 재설계합니다. (a) 원래 StyleGAN, 여기서 A는 스타일을 생성하는 W로부터 학습된 아핀 변환을 나타내고 B는 노이즈 브로드캐스트 작업입니다. (b) 세부 사항이 포함된 동일한 다이어그램. 여기에서 우리는 AdaIN을 명시적 정규화와 변조가 뒤따르는 것으로 나누었습니다. 둘 다 특징 맵당 평균과 표준 편차에서 작동합니다.

또한 학습된 가중치(w), 편향(b) 및 상수 입력(c)에 주석을 달았고 상당한 하나의 스타일이 활성화되도록 회색 상자를 다시 그렸습니다. 활성화 함수(leaky ReLU)는 항상 편향을 추가한 직후에 적용됩니다. (c) 본문에서 정당화되는 원본 아키텍처를 몇 가지 변경합니다. 처음에 일부 중복 작업을 제거하고 b와 B의 추가를 스타일의 활성 영역 외부로 이동하고 기능 맵당 표준 편차만 조정합니다. (d) 수정된 아키텍처를 통해 인스턴스 정규화를 "복조" 작업으로 대체할 수 있으며, 이를 각 컨볼루션 레이어와 관련된 가중치에 적용합니다.

그림 2c의 스타일 블록은 변조, 컨볼루션 및 정규화로 구성되어 있습니다. 변조와 컨볼루션의 효과를 고려하여 시작하겠습니다.

변조는 들어오는 스타일을 기반으로 컨볼루션의 각 입력 기능 맵을 확장하며, 이는 대안적으로 컨볼루션 가중치를 확장하여 구현할 수 있습니다.

$$w_{ijk} = s_i w_{ijk} \quad (1)$$

여기서 w 와 w 는 각각 원래 가중치와 변조된 가중치이고, s_i 는 i 번째 입력 특징 맵에 해당하는 척도이고, j 와 k 는 각각 출력 특징 맵과 컨볼루션의 공간 풋프린트를 열거합니다.

이제 인스턴스 정규화의 목적은 본질적으로 컨볼루션의 출력 기능 맵 통계에서 s 의 효과를 제거하는 것입니다. 우리는 이 목표를 보다 직접적으로 달성할 수 있음을 관찰했습니다. 입력 활성화가 단위 표준 편차를 갖는 iid 랜덤 변수라고 가정합니다. 변조 및 회선 후 출력 활성화의 표준 편차는 다음과 같습니다.

$$\sigma_j = \sqrt{w_0^2 + s_i^2} \quad (2)$$

즉, 출력은 해당 가중치의 L2 표준에 의해 조정됩니다. 후속 정규화는 출력을 다시 단위 표준 편차로 복원하는 것을 목표로 합니다. 방정식 2를 기반으로 확장("복조")하면 달성됩니다.

각 출력 특징 맵 j 는 $1/\sigma_j$ 입니다. 또는 이것을 컨볼루션 가중치로 다시 구할 수 있습니다.

$$w_{ijk} = \frac{w_{ijk}}{\sigma_j} \quad (3)$$

여기서는 숫자 문제를 피하기 위한 작은 상수입니다.

이제 방정식 1과 3을 사용하여 가중치가 s 에 따라 조정된 단일 컨볼루션 레이어에 전체 스타일 블록을 구성합니다 (그림 2d). instance 또는 malization과 비교하여 우리의 복조 기술은 특성 맵의 실제 내용 대신 신호에 대한 통계적 가정을 기반으로 하기 때문에 약합니다. 유사한 통계 분석이 최신 네트워크 이니셜라이저 [14, 19]에서 광범위하게 사용되었지만 이전에 데이터 종속 정규화를 대체하기 위해 사용되었다는 사실을 알지 못합니다. 우리의 복조는 또한 가중치 텐서를 다시 매개변수화하는 일부로 동일한 계산을 수행하는 가중치 정규화 [37]와 관련이 있습니다. 이전 작업에서는 가중치 정규화가 GAN 교육의 맥락에서 유익한 것으로 확인되었습니다 [43].

우리의 새로운 디자인은 첨부된 비디오에서 보여지는 것처럼 완전한 제어성을 유지하면서 특징적인 인공물을 제거합니다(그림 3). FID는 크게 영향을 받지 않았지만(표 1, 행 A, B), 정밀도에서 재현율로 눈에 띄는 변화가 있습니다. 리콜이 절단을 통해 정밀도로 교환될 수 있기 때문에 이것이 일반적인 것으로 바람직하다고 주장하지만,

구성	FFHQ, 1024×1024 FID					LSUN 자동차, 512×384			
	↓ 경로 길이 ↓	정밀도 ↑	리콜 ↑	4.40	4.39	FID ↓	경로 길이 ↓	정밀도 ↑	재현율 ↑
기본 스타일GAN [24]	3.98	212.1	0.721	0.399		3.27	1484.5	0.701	0.435
B + 가중치 복조 C + 게으른 정규화		175.4	0.702	0.425		3.04	862.4	0.685	0.488
화 D + 경로 길이 정규화 E + 성장 없음, 새로운 G & D 아치.		158.0	0.719	0.427		2.83	981.6	0.688	0.493
		122.5	0.715	0.418		3.43	651.2	0.697	0.452
		124.5	0.705	0.449		3.19	471.2	0.690	0.454
F + 대규모 네트워크(StyleGAN2)		145.0	0.689	0.492		2.32	415.5	0.678	0.514
대규모 네트워크가 있는 구성 A		199.2	0.716	0.422		-	-	-	-

표 1. 주요 결과. 각 훈련 실행에 대해 FID가 가장 낮은 훈련 스냅샷을 선택했습니다. 각 측정항목을 10번 계산했습니다. 다른 임의의 시드를 사용하여 평균을 보고합니다. 경로 길이는 W의 경로 끝점을 기반으로 계산된 PPL 메트릭에 해당합니다. [24], Karras et al. [24]. FFHQ 데이터 세트에는 70,000개의 이미지가 포함되어 있고 판별자는 2,500만 개의 이미지를 보았습니다. 훈련 중, LSUN CAR의 경우 숫자는 893k와 57M이었습니다. ↑는 높을수록 좋고, ↓는 낮을수록 좋다.



그림 3. 정규화를 복조로 대체하면 제거 이미지 및 활성화의 특징적인 아티팩트.

그 반대는 사실이 아니다 [27]. 실제로 우리의 디자인은 부록 B에 자세히 설명된 대로 그룹화된 컨볼루션을 사용하여 효율적으로 구현됩니다. 방정식 3의 활성화 함수에서 활성화를 확장합니다. 예상되는 신호 분산을 유지하도록 기능을 제공합니다.

3. 이미지 품질 및 생성기 부드러움

FID 또는 Precision 및 Recall과 같은 GAN 메트릭 (P&R) 발전기의 여러 측면을 성공적으로 포착하고, 그들은 이미지에 대한 약간의 시각 지대를 계속 가지고 있습니다. 품질. 예를 들어 그림 13 및 14를 참조하십시오. FID 및 P&R 점수는 동일하지만 조영제 생성기 현저하게 다른 전반적인 품질.2

2 우리는 명백한 불일치의 핵심이 FID 또는 P&R의 기초가 아니라 특징 공간의 특정 선택에 있다고 믿습니다. ImageNet [35]을 사용하여 훈련된 분류기는 다음과 같은 경향이 있다는 것이 최근에 발견되었습니다. 인간은 모양보다 질감에 훨씬 더 많은 결정을 내리는 반면 [12], 인간은 모양에 강력하게 초점을 맞춥니다 [28]. 이것은 우리의 맥락과 관련이 있습니다. 왜냐하면



그림 4. 각각 경로 길이와 이미지의 연결 LSUN CAT와 함께 기존 StyleGAN(구성 A)을 사용하는 품질. (↑) 낮은 PPL(≤ 10번째 백분위수)이 있는 임의의 예. (b) PPL이 높은 예(≥ 90번째 백분위수). PPL 점수와 이미지의 의미론적 일관성 사이에는 명확한 상관 관계가 있습니다.

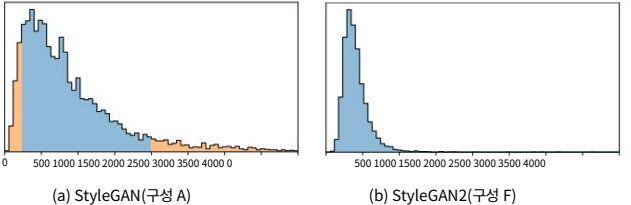


그림 5. (a) 개별 이미지의 PPL 점수 분포 LSUN CAT에서 기존 StyleGAN(구성 A)을 사용하여 생성된 (FID = 8.53, PPL = 924).에 해당하는 백분위수 범위 그림 4는 주황색으로 강조 표시됩니다. (b) StyleGAN2(config F) im은 PPL 분포를 상당히 증명합니다(스냅샷 표시 동일한 FID = 8.53, PPL = 387).

인자된 이미지 품질과 인지 경로 길이(PPL) [24] 사이의 상관 관계를 관찰합니다. 원래 부드러움을 정량화하기 위해 도입된 생성된 이미지 사이의 평균 LPIPS 거리 [50]를 측정하여 잠재 공간에서 출력 이미지로 매핑 잠재 공간의 작은 섭동 아래. 다시 컨설팅 그림 13 및 14에서 PPL(매끄러운 생성기 매핑 핑)이 작을수록 전체 이미지 품질이 더 높은 것으로 나타납니다.

FID 및 P&R은 InceptionV3 [39] 및 VGG-16의 고급 기능을 사용합니다. [39], 각각 이러한 방식으로 훈련되었으며 따라서 예상되는 텍스처 감지에 편향됩니다. 따라서 예를 들어 강한 이미지가 고양이 질감은 인간 관찰자보다 서로 더 유사하게 나타날 수 있습니다. 따라서 밀도 기반 메트릭(FID) 및 매니폴드 커버리지 메트릭(P&R).

다른 메트릭은 변경 사항에 대해 맹목적인 반면, 그림 4는 $w \sim f(z)$ 주변의 잠재 공간을 샘플링하여 계산된 LSUN CAT의 이미지별 PPL 점수를 통해 이상관 관계를 더 자세히 조사합니다. 낮은 점수는 실제로 고품질 이미지를 나타내며 그 반대의 경우도 마찬가지입니다. 그림 5a는 해당 히스토그램을 보여주고 분포의 긴 꼬리를 보여줍니다. 모델의 전체 PPL은 단순히 이러한 이미지별 PPL 점수의 예상 값입니다. Karras et al.과 달리 우리는 항상 전체 이미지에 대해 PPL을 계산합니다. 더 작은 중앙 작물을 사용하는 사람들.

낮은 PPL이 이미지 품질과 상관관계가 있어야 하는 이유는 즉시 명확하지 않습니다. 훈련 중에 판별자가 깨진 이미지에 페널티를 주기 때문에 생성기가 개선하는 가장 직접적인 방법은 좋은 이미지를 생성하는 잠재 공간 영역을 효과적으로 늘리는 것이라고 가정합니다.

이것은 낮은 품질의 이미지가 급격한 변화의 작은 잠재 공간 영역으로 압축되도록 할 것입니다. 이렇게 하면 단계적으로 평균 출력 품질이 향상되지만 왜곡이 누적되면 훈련 역학이 손상되어 결과적으로 최종 이미지 품질이 저하됩니다.

분명히 최소한의 PPL을 권장할 수는 없습니다. 이는 생성기를 리콜이 없는 퇴화 솔루션으로 안내할 것이기 때문입니다. 대신, 우리는 이러한 단점 없이 더 부드러운 생성기 매핑을 목표로 하는 새로운 정규화기를 설명할 것입니다. 결과 정규화 항은 계산하는 데 다소 비용이 많이 들기 때문에 먼저 모든 정규화 기술에 적용되는 일반적인 최적화를 설명합니다.

3.1. 자연 정규화

일반적으로 주 손실 함수(예: 로지스틱 손실 [16])와 정규화 항(예: R1 [30])은 단일 표현식으로 작성되므로 동시에 최적화됩니다. 우리는 정규화 항이 주 손실 함수보다 덜 자주 계산될 수 있으므로 계산 비용과 전체 메모리 사용 시간을 크게 줄일 수 있음을 관찰했습니다. 표 1의 C 행은 R1 정규화가 16개의 미니 배치마다 한 번만 수행될 때 아무런 해가 없음을 보여주고 우리는 새로운 정규화에 대해서도 동일한 전략을 채택합니다. 부록 B는 구현 세부 사항을 제공합니다.

3.2. 경로 길이 정규화

W의 고정 크기 단계로 인해 이미지에서 0이 아닌 고정 크기 변화가 발생하도록 권장하고 싶습니다.

이미지 공간에서 임의의 방향으로 이동하고 해당 w 기울기를 관찰하여 이 이상으로부터의 편차를 경험적으로 측정할 수 있습니다. 이러한 기울기는 w 또는 이미지 공간 방향에 관계없이 거의 동일한 길이를 가져야 하며 이는 잠재 공간에서 이미지 공간으로의 매핑이 잘 조절되었음을 나타냅니다 [33].

단일 $w \in W$ 에서 생성기 매핑 $g(w) : W \rightarrow Y$ 의 로컬 메트릭 스케일링 속성은 야코비 행렬 $J_w = \partial g(w) / \partial w$ 에 의해 캡처됩니다. 관계없이 벡터의 예상 길이를 유지하려는 욕구에 동기 부여

방향으로, 우리는 정규화기를 다음과 같이 공식화합니다.

$$E_{w,y} = \mathbb{E}_{N(0,I)} J_w^T w^2, \quad (4)$$

여기서 y 는 정규 분포된 픽셀 강도를 갖는 임의의 이미지이고 $w \sim f(z)$, 여기서 z 는 정규 분포를 따릅니다. 부록 C에서 고차원에서 J_w 가 임의의 w 에서 직교(글로벌 규모까지)일 때 이 사전이 최소화된다는 것을 보여줍니다. 직교 행렬은 길이를 유지하고 차원을 따라 짜지 않습니다.

Jacobian 행렬의 명시적 계산을 피하기 위해 $wy = \nabla w(g(w) \cdot y)$ 는 표 우리는 ID J 를 사용합니다. ∇ 역전파를 사용하여 효율적으로 계산할 수 있습니다 [6].

상수 a 는 최적화 중에 길이 kJ_w^2 의 장기 실행 지수 이동 평균으로 동적으로 설정되어 최적화가 자체적으로 적절한 전역 규모를 찾을 수 있도록 합니다.

우리의 regularizer는 Odena et al.이 제시한 Jacobian clamping regularizer와 밀접한 관련이 있습니다. [33]. 실제적인 차이점에는 $J_w^T J_w$ 를 $\mathbb{E}_{N(0,I)} J_w^T J_w$ 로 추정하기 위해 유한 차분을 사용하는 반면에 제곱 행렬로 계산한다는 점입니다. 생성기 [46]의 스펙트럼 정규화 [31]는 가장 큰 특이값만 제한하고 다른 값에는 제약을 가하지 않으므로 반드시 더 나은 컨디셔닝으로 이어지는 않는다는 점에 유의해야 합니다.

부록 E에 자세히 설명된 바와 같이, 우리의 기여에 추가로 또는 그 대신에 스펙트럼 정규화를 활성화하면 항상 FID를 약속한다는 사실을 알게 되었습니다.

실제로 우리는 경로 길이 정규화가 더 안정적이고 일관되게 작동하는 모델로 이어져 아티팩트 탐색을 더 쉽게 만든다는 것을 알았습니다. 우리는 또한 부드러운 생성기가 반전하기가 훨씬 더 쉽다는 것을 관찰했습니다(색션 5). 그림 5b는 경로 길이 정규화가 모드를 0으로 푸시하지 않고 이미지별 PPL 점수 분포를 명확하게 강화한다는 것을 보여줍니다. 그러나 표 1의 D 행은 FFHQ보다 덜 구조화된 데이터 세트에서 FID와 PPL 사이의 절충점을 가리킵니다.

4. 점진적 성장 재검토

점진적 성장 [23]은 고해상도 이미지 합성을 안정화하는 데 매우 성공적이었지만 자체 특성 아티팩트가 발생합니다. 핵심 문제는 점진적으로 성장하는 발전기가 세부 사항에 대해 강한 위치 선호도를 갖는 것으로 보인다는 것입니다. 첨부된 비디오는 치아나 눈과 같은 특징이 이미지 위에서 부드럽게 움직여야 할 때 대신 다음 선호 위치로 점프하기 전에 제자리에 고정되어 있을 수 있음을 보여줍니다. 그림 6은 관련 아티팩트를 보여줍니다. 문제는 점진적 성장에서 각 해상도가 일시적으로 출력 해상도로 작용하여 최대 주파수 세부 사항을 생성하도록 하여 훈련된 네트워크가 중간 계층에서 지나치게 높은 주파수를 갖게 되어 시프트 불변성을 손상시킨다는 것입니다 [49]. 부록 A는 예를 보여줍니다. 이것들



그림 6. 점진적 성장으로 인해 "단계" 아티팩트가 발생합니다. 이에 예를 들어 치아는 포즈를 따르지 않고 정렬된 상태를 유지합니다. 파란색 선으로 표시된 대로 카메라.

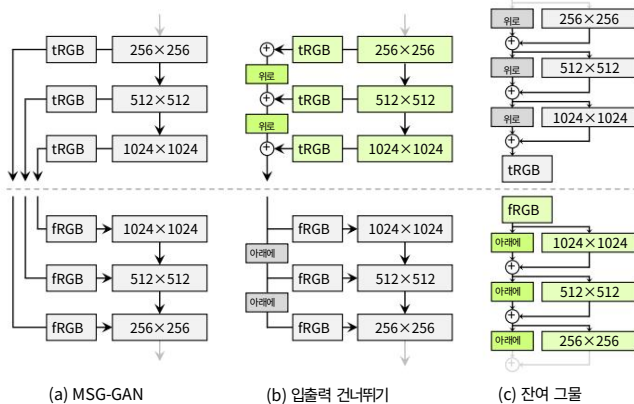


그림 7. 3개의 생성기(점선 위) 및 판별기 아키텍처. Up 및 Down은 각각 상향형 업 및 다운 샘플링을 나타냅니다. 잔차 네트워크에는 다음이 포함됩니다.

특정 맵의 수를 조정하기 위한 1×1 컨볼루션. tRGB 및 fRGB는 RGB와 픽셀당 고차원 간 변환 데이터. 구성 E 및 F에 사용된 아키텍처는 녹색으로 표시됩니다.

문제는 우리가 대체 공식을 찾도록 유도합니다. 단점 없이 점진적 성장의 이점을 유지하는 것입니다.

4.1. 대체 네트워크 아키텍처

StyleGAN은 간단한 피드포워드 디자인을 사용하지만 생성기(합성 네트워크) 및 판별기, 더 나은 네트워크 연구에 전념하는 방대한 작업 아키텍처. 연결 건너뛰기 [34, 22], 잔여 네트워크 [18, 17, 31] 및 계층적 방법 [7, 47, 48]이 있습니다. 생성의 맥락에서도 매우 성공적인 것으로 입증되었습니다. 행동 양식. 따라서 우리는 네트워크를 재평가하기로 결정했습니다. StyleGAN의 디자인과 점진적 성장 없이 고품질의 이미지를 생성하는 아키텍처를 추구합니다.

그림 7a는 MSG-GAN [22]을 보여줍니다. 다중 건너뛰기 연결을 사용하여 생성기와 판별기의 일치하는 해상도. MSG-GAN 생성기 이미지 대신 맵 [42]을 출력하도록 수정됩니다. 각 실수 im에 대해 유사한 표현이 계산됩니다.

FFHQ	D 원본 FID PPL 4.32 265	D 입력 건너뛰기 FID PPL	D 잔차 FID PPL
G 오리지널	4.33 169 4255	4.18 235 3277	3.58 269
G 출력 건너뛰기 G 잔차	3.96 229	3.96 229	3.31 125 3.79 243

엘센카	D 원본 FID PPL 3.75 905	D 입력 건너뛰기 FID PPL	D 잔차 FID PPL
G 오리지널	3.77 544 3983	3.23 758 3186	3.25 802
G 출력 건너뛰기 G 잔차	3.40 667	3.40 667	3.19 471 2.66 645

표 2. 생성기 및 판별기 아키텍처 비교

점진적 성장 없이. 발전기의 조합

출력 건너뛰기와 잔차 판별자는 주 결과 테이블의 구성 E에 해당합니다.

나이도. 그림 7b에서는 RGB 출력의 기여도를 업샘플링하고 합산하여 이 설계를 단순화합니다.

다른 해상도에 해당합니다. 판별자에서, 우리는 유사하게 판별자의 각 해상도 블록에 다운샘플링된 이미지를 제공합니다. 우리는 이중 선형 필터링을 사용합니다.

모든 업 및 다운샘플링 작업. 그림 7c에서는 잔류 연결을 사용하도록 설계를 추가로 수정합니다.3

디자인은 해상도가 없는 LAGAN [7]과 유사합니다.

Denton et al.

표 2는 3개의 생성기 및 3개의 판별기 아키텍처를 비교합니다. 사용된 원래 피드포워드 네트워크

StyleGAN에서 건너뛰기 연결 및 잔여 네트워크, 모두 점진적 성장 없이 훈련. 9가지 조합 각각에 대해 FID와 PPL이 제공됩니다. 우리는 두 가지 넓은 것을 볼 수 있습니다

경향: 생성기의 건너뛰기 연결은 모든 구성에서 PPL을 크게 향상시키고 잔여 판별자

네트워크는 FID에 분명히 유익합니다. 후자는 아마도

판별자의 구조가 잔여 아키텍처가 다음과 같이 알려진 분류자와 유사하기 때문에 놀라운 일이 아닙니다.

도움이 되는. 그러나 잔여 아키텍처는

발전기 - 유일한 예외는 LSUN CAR의 FID였습니다.

두 네트워크가 모두 잔차일 때.

나머지 문서에서는 건너뛰기 생성기와

점진적 성장이 없는 잔차 판별자. 이것

표 1의 구성 E에 해당하며 FID 및 PPL을 크게 향상시킵니다.

4.2. 해상도 사용

점진적 성장의 핵심 측면, 우리는

보존하고 싶은 것은 발전기가 처음에

저해상도 기능에 대한 관심을 천천히 전환합니다.

세부 사항. 그림 7의 아키텍처는 다음을 가능하게 합니다.

생성기가 먼저 저해상도 이미지를 출력하려면

고해상도 레이어의 영향을 크게 받지 않고 나중에 고해상도 레이어로 초점을 이동합니다.

³ 잔여 네트워크 아키텍처에서 두 개의 경로를 추가하면

신호 분산의 두 배로 $1/\sqrt{2}$ 를 곱하여 취소합니다.

이것은 네트워크에 중요하지만 분류에서는 resnet [18]

문제는 일반적으로 배치 정규화에 의해 숨겨집니다.

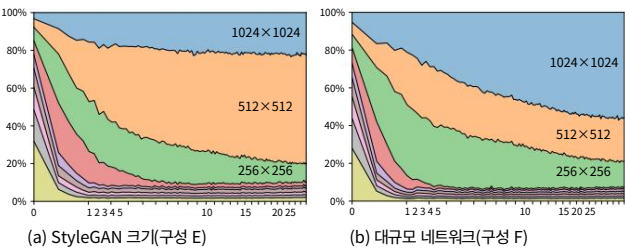


그림 8. 출력에 대한 각 해상도의 기여도
훈련 시간의 함수로 생성기. 세로축은 다른 해상도의 상대 표준 편차 분석
lutions, 수평축은 훈련 진행 상황에 해당하며, 판별자에게 표시되는 수백만 개의 훈련 이미지로 측정됩니다. 처음에는 네트워크가 저해상도 이미지에 초점을 맞추고 훈련이 진행됨에 따라 점차적으로 더 큰 해상도로 초점을 이동한다는 것을 알 수 있습니다. (a)에서 생성기는 기본적으로 1024에 대해 약간의 선명도가 있는 5122 이미지를 출력합니다.
, 동안
(b) 더 큰 네트워크는 고해상도 세부 사항에 더 중점을 둡니다.

훈련이 진행됨에 따라 레이어. 에서 시행하지 않기 때문에
어쨌든, 발전기는 그것이 유익한 경우에만 그것을 할 것입니다. 예제
실제로 행동을 분석하려면 어떻게
생성기는 특정 해상도에 크게 의존합니다.
훈련 과정.
스킵 생성기(그림 7b)는 다음과 같이 이미지를 형성하기 때문에
여러 해상도에서 RGB 값을 명시적으로 합산하고,
우리는 그들이 기여도를 측정함으로써 해당 계층의 상대적 중요성을 추정
할 수 있습니다.
최종 이미지. 그림 8a에서 표준 편차를 플로팅합니다.
함수로 각 tRGB 레이어에 의해 생성된 픽셀 값
훈련 시간의. 우리는 표준 편차를 계산합니다
1024개의 무작위 샘플 w 및 다음과 같이 값을 정규화합니다.
그것들의 합은 100%입니다.
훈련을 시작할 때 새로운 건너뛰기
제너레이터는 점진적 성장과 유사하게 동작합니다. — 지금
네트워크 토폴로지를 변경하지 않고 달성됩니다. 그것은 것
따라서 교육이 끝날 때 가장 높은 해상도가 지배적일 것으로 예상하는 것이
합리적입니다. 그러나 줄거리는,
이것이 실제로 발생하지 않는다는 것을 보여줍니다.
생성기가 tar get 해상도를 "완전히 활용"하지 못할 수 있습니다. 이를 확
인하기 위해 생성된
이미지를 수동으로 확인하고 일반적으로
훈련 데이터에 있는 픽셀 수준 세부 정보 —
이미지는 날카롭게 된 버전으로 설명될 수 있습니다.
실제 10242 이미지 대신 5122 이미지.
이것은 우리 네트워크에 용량 문제가 있다는 가정을 하게 하며, 우리는
숫자를 두 배로 늘려 테스트합니다.
두 네트워크의 최고 해상도 레이어에 있는 기능 맵의 수

4해상도 642-10242 에서 기능 맵 수를 두 배로 늘립니다.
네트워크의 다른 부분을 변경하지 않은 상태로 유지합니다. 이것은 증가
생성기에서 학습 가능한 총 매개변수 수 22%(2,500만 →
30M) 및 판별기에서 21%(24M → 29M).

데이터세트	해결	StyleGAN (A) StyleGAN2 (F) FID PPL 3.27	FID PPL
엘슨 자동차 512×384		1485 8.53 924	2.32 416
LSUN CAT 256×256		742 3.83 1405	6.93 439
울슨 교회 256 × 256		4.21	3.86 342
LSUN 말 256×256			3.43 338

표 3. FID를 사용하여 측정된 LSUN 데이터 세트의 개선 및
PPL. 우리는 57M 이미지에 대해 CAR, 88M 이미지에 대해 CAT, CHURCH 에 대해 교육했습니다.
48M의 경우 HORSE, 100M 이미지의 경우 HORSE.

그림 8b는 최고 해상도 레이어의 기여도가 크게 증가한 것을 보여줍니다.
표 1, F 행
FID와 Recall이 크게 향상됨을 보여줍니다. 마지막 행
기본 StyleGAN이 추가로 이점을 얻을 수 있음을 보여줍니다.
용량은 적지만 품질은 StyleGAN2보다 훨씬 낮습니다.
표 3 은 StyleGAN과 StyleGAN2를 4가지로 비교한 것입니다.
LSUN 카테고리, 다시 한 번 분명한 개선 사항을 보여줍니다.
FID 및 PPL의 중요한 발전. 모피의 크기가 더 커지면 추가 이점을 제공할
수 있습니다.

5. 이미지를 잠재 공간에 투영

합성 네트워크 g를 반전시키는 것은 많은 응용이 있는 흥미로운 문제입
니다. 잠재 가능 공간에서 주어진 이미지를 조작하려면 일치하는 항목을 찾
아야 합니다.
먼저 잠재 코드 w. 이전 연구 [1, 10] 는 다음과 같이 제안합니다 .
공통 잠재 코드 w를 찾는 대신 결과
발전기의 각 레이어에 대해 별도의 w를 선택하면 향상됩니다. 동일한 접근
방식이 초기 인코더 구현에서 사용되었습니다 [32]. 이 안에 잠재 공간을
확장하면서
패션은 주어진 이미지에 더 가까운 것을 찾고
반성이 없어야 하는 임의의 이미지를 투사합니다. 대신 잠재 코드를 찾는 데
집중합니다.
원래의 확장되지 않은 잠재 공간에서
생성자가 생성할 수 있는 이미지에.
우리의 투영 방법은 이전 방법과 다릅니다.
두 가지 방법으로. 첫째, 잠재성을 탐색하기 위해 최적화 동안 잠재성 코드
에 램프다운 노이즈를 추가합니다.
공간을 보다 포괄적으로. 둘째, 우리는 또한 최적화
StyleGAN 생성기의 확률적 노이즈 입력, 일관성 있는 결과를 전달하지 않
도록 정규화
신호. 정규화는 노이즈 맵의 자동 상관 계수가 다음과 일치하도록 강제하는
것을 기반으로 합니다.
여러 스케일에 대한 단위 가우스 노이즈. 우리의 투영 방법에 대한
자세한 내용은 부록 D 에서 찾을 수 있습니다.

5.1. 생성된 이미지의 속성

조작되거나 생성된 이미지의 감지는 매우
중요한 임무. 현재 분류기 기반 방법은 다음을 수행할 수 있습니다.
생성된 이미지를 매우 안정적으로 감지합니다.
정확한 기원 [29, 45, 40, 51, 41]. 그러나 빠른 속도를 감안할 때
생성 방법의 진행 속도, 이것은
지속되는 상황. 가짜 이미지의 일반적인 탐지 외에도,
더 제한된 형태의 문제를 고려할 수도 있습니다.



StyleGAN - 생성된 이미지

StyleGAN2 - 생성된 이미지

StyleGAN2 - 실제 이미지

그림 9. 예시 이미지와 해당 투영 및 재합성 대응물. 각 구성에 대해 맨 위 행에는 대상 이미지가 표시됩니다.

하단 행은 해당하는 투영된 잠재 벡터 및 노이즈 입력의 합성을 보여줍니다. 기존 StyleGAN을 사용하여 투영

중 생성된 이미지와 상당히 일치하는 항목을 찾지만 특히 배경은 원본과 다릅니다. 생성된 이미지

StyleGAN2를 사용하면 생성기 입력으로 거의 완벽하게 다시 투영될 수 있으며, 투영된 실제 이미지(훈련 세트에서)는 다음과 같이 표시됩니다.

예상대로 원본과의 명확한 차이점. 모든 테스트는 동일한 프로세스 방법과 하이퍼파라미터를 사용하여 수행되었습니다.

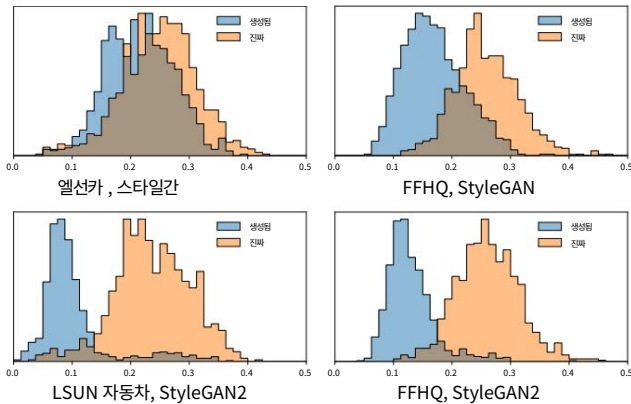


그림 10. 생성된 이미지(파란색)와 실제 이미지(주황색)에 대한 원본 이미지와 투영된 이미지 간의 LPIPS 거리 히스토그램. 개선된 생성기의 더 높은 이미지 품질에도 불구하고

생성된 이미지를 잠재 공간에 투영하는 것이 훨씬 쉽습니다.

W. 모든 경우에 동일한 투영 방법이 사용되었습니다.

가짜 이미지를 특정 소스에 귀속시킬 수 있음 [2].

StyleGAN을 사용하면 존재하는지 확인하는 것과 같습니다.

$w \in W$ 는 해당 이미지를 다시 합성합니다.

우리는 DLPIPS $[x, g(-g$

$-1(x))]$, 여기서 x 는

분석 중인 이미지 및 g^{-1} 대략적인 투영 작업을 나타냅니다.

그림 10은 원본 StyleGAN 및 StyleGAN2를 사용하는 LSUN CAR 및 FFHQ 데이터 세트에 대한 이러한 거리의 히스토그램을 보여 주고 그림 9는 예시 프로젝션을 보여줍니다. StyleGAN2를 사용하여 생성된 이미지

거의 W에 투영될 수 있습니다.

분명히 생성 네트워크에 기인합니다. 그러나 원래 StyleGAN을 사용하면

기술적으로 일치하는 잠재 코드를 찾는 것이 가능하지만 W에서 이미지로의 매핑이 너무 복잡한 것으로 보입니다.

이것이 실제로 안정적으로 성공하려면, StyleGAN2가 소스 속성을 더욱 쉽게

이미지 품질이 크게 향상되었지만.

6. 결론 및 향후 과제

StyleGAN에서 몇 가지 이미지 품질 문제를 식별하고 수정하여 품질을 더욱 개선하고 여러 데이터 세트에서 최신 기술을 상당히 발전시켰습니다.

어떤 경우에는 개선 사항이 다음에서 더 명확하게 나타납니다.

함께 제공되는 비디오에서 볼 수 있듯이 모션입니다. 부록 A에는 우리의 방법을 사용하여 얻을 수 있는 결과의 추가 예가 포함되어 있습니다. 개선된 품질에도 불구하고 StyleGAN2 생성된 이미지를 소스에 더 쉽게 지정할 수 있습니다.

훈련 성과도 향상되었습니다. 1024에서

해상도, 원래 StyleGAN(표 1의 구성 A)

NVIDIA DGX-1에서 초당 37개의 이미지로 훈련

8 Tesla V100 GPU, config E는 40% 더 빠르게 학습

61 img/s에서. 대부분의 속도 향상은

가중치 복조, 자연 정규화로 인한 데이터 흐름,

및 코드 최적화. StyleGAN2(config F, 더 큰 네트워크 작업)는 31 img/s로 학습하므로 약간 더 높습니다.

원래 StyleGAN보다 훈련 비용이 싸입니다. 그것의 총 훈련

시간은 FFHQ의 경우 9일, LSUN CAR의 경우 13일이었습니다.

모든 탐사를 포함한 전체 프로젝트가 소비되었습니다.

132MWh의 전기 중 0.68MWh가

최종 FFHQ 모델을 훈련합니다. 전체적으로 우리는 약

51년 단일 GPU 연산(Volta 클래스 GPU). †

더 자세한 논의는 부록 F에서 볼 수 있습니다.

미래에는 경로 길이 정규화에 대한 추가 개선 사항을 연구하는 것이 유익할 수 있습니다.

데이터 기반 특징 공간이 있는 픽셀 공간 L2 거리

미터법. GAN의 실용적인 배치를 고려하여,

줄일 수 있는 새로운 방법을 찾는 것이 중요할 것이라고 생각합니다.

훈련 데이터 요구 사항. 이는 수만 개를 획득하는 것이 불가능한 애플리케이션에서 특히 중요합니다.

많은 훈련 샘플과

내재적 변이.

감사의 말 일찍 Ming-Yu Liu에게 감사드립니다.

검토, 공개 릴리스에 대한 도움을 받은 Timo Viitanen,

David Luebke는 심층 토론과 유용한 의견을, Tero Kuosmanen은 기술 지원을 위해

컴퓨팅 인프라.

참고문헌

- [1] 라민 압달, 이평 친, 피터 왕카. Image2StyleGAN: StyleGAN에 이미지를 삽입하는 방법
잠재 공간? ICCV에서 2019. 7
- [2] 마이클 올브라이트와 스콧 맥클로스키. 소스 생성기
역전을 통한 귀인. CVPR 워크샵에서, 2019. 8
- [3] 칼 버그스트롬과 제빈 웨스트. 진짜? <http://www.whichfaceisreal.com/learn.html>, 액세스
2019년 11월 15일. 1 어떤 얼굴이
- [4] 크리스토퍼 M. 비숍. 패턴 인식 및 기계
학습. 스프링거, 2006. 17
- [5] Andrew Brock, Jeff Donahue, Karen Simonyan. 크기가 큰
충실도가 높은 자연 이미지 합성을 위한 scale GAN 훈련.
CorR, abs/1809.11096, 2018. 1
- [6] Yann N. Dauphin, Harm de Vries, Yoshua Bengio. 비블록 최적화를 위한 평
형 적응 학습률.
CorR, abs/1502.04390, 2015. 5
- [7] Emily L. Denton, Soumith Chintala, Arthur Szlam,
로버트 퍼거스. 다음을 사용하는 심층 생성 이미지 모델
적대적 네트워크의 라플라시안 피라미드. 코르,
abs/1506.05751, 2015. 6
- [8] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub,
Harm de Vries, Aaron Courville, Yoshua
벤지오. 기능별 변환. 증류, 2018.
<https://distill.pub/2018/feature-wise-transformations>. 1
- [9] Vincent Dumoulin, Jonathon Shlens, Manjunath Kudrur. 예술적 스타일
에 대한 학습된 표현. 코르,
abs/1610.07629, 2016. 1
- [10] 아비브 가바이와 예디드 호센. 이미지 향상 및 애니메이션 버전의 스타일 생성기. 코
르,
abs/1906.11880, 2019. 7
- [11] R. Ge, X. Feng, H. Pyla, K. Cameron, W.
평. Green500용 전력 측정 튜토리얼
목록. <https://www.top500.org/green500/resources/tutorials/>,
2020년 3월 1일에 액세스함. 21
- [12] 로버트 게이로스, 매트리스 루비쉬, 클라우디오 마카엘리스,
Matthias Bethge, Felix A. Wichmann, Wieland Brendel.
ImageNet 훈련 CNN은 텍스처에 편향되어 있습니다. 모양 바이어스를 높이면 정
확도와 견고성이 향상됩니다. 코르,
abs/1811.12231, 2018. 1, 4
- [13] 골나즈 기아시, 이홍락, 만주나트 쿠틀루르, 빈센트
Dumoulin, Jonathon Shlens. 의 구조 탐색
실시간 임의 신경 예술적 양식화 네트워크. 코르,
abs/1705.06830, 2017. 1
- [14] 자비에 글로로와 요슈아 벤지오. 심층 피드포워드 신경망 훈련의 어려움 이해. 인공 지
능 및 통계에 관한 제13차 국제 회의의 회보, 249–256페이지, 2010. 3
- [15] GH Golub 및 CF Van Loan. 행렬 계산. 존스
수리 과학의 홉킨스 연구. Johns Hopkins University Press, 2013. 17
- [16] 이안 굿펠로우, 장 푸제-아바디, 메디 미르자, 빙
Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville,
요슈아 벤지오. 생성적 적대 네트워크. NIPS에서
2014. 1, 5, 11
- [17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent
Dumoulin, 및 Aaron C. Courville. 향상된 교육
Wasserstein GAN. CorR, abs/1704.00028, 2017. 6
- [18] 허카이밍, 장상위, 샤오칭런, 지안
해. 이미지 인식을 위한 딥 레지듀얼 러닝. 코르,
abs/1512.03385, 2015. 6
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.
정류기 자세히 알아보기: ImageNet 분류에서 인간 수준의 성능을 능가합니다.
CorR, abs/1502.01852,
2015. 3
- [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner,
Bernhard Nessler와 Sepp Hochreiter. 에 의해 훈련된 GAN
두 개의 시간 척도 업데이트 규칙은 로컬 내쉬 평형으로 수렴됩니다. 프로시저에서
NIPS, 6626–6637페이지, 2017. 1
- [21] Xun Huang 및 Serge J. Belongie. 적응형 인스턴스 정규화로 실시간 임의의 스타
일 전송. 코르,
abs/1703.06868, 2017. 1
- [22] 아니메쉬 카네와라와 올리버 왕. MSG-GAN: 생성적 적대 네트워크에 대한 다중 스케
일 기울기. 프로시저에서
CVPR, 2020. 6
- [23] 테로 카라스, 티모 아일라, 사무리 레인, 야코 레타넨.
향상된 품질, 안정성,
및 변형. CorR, abs/1710.10196, 2017. 1, 5, 11
- [24] 테로 카라스, 사무리 레인, 티모 아일라. 스타일 기반
생성적 적대 네트워크를 위한 생성기 아키텍처. 에
절차 CVPR, 2018. 1, 2, 4, 5, 11, 13, 16, 20
- [25] 디에데릭 P. 킹마와 지미 바. 아담: 방법
확률 최적화. ICLR에서, 2015. 11, 19
- [26] Alex Krizhevsky, Ilya Sutskever 및 Geoffrey E. Hinton.
심층 컨볼루션 신경망을 사용한 ImageNet 분류는 작동합니다. NIPS, 1097–1105
페이지. 2012. 11 ..
- [27] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen
및 Timo Aila. 생성 모델 추정을 위한 향상된 정밀도 및 재현율 메트릭. Proc.
NeurIPS에서 2019.
1, 2, 4
- [28] Barbara Landau, Linda B. Smith, Susan S. Jones. 그만큼
초기 어휘 학습에서 모양의 중요성. 인지 발달, 3(3), 1988. 4
- [29] Haodong Li, Han Chen, Bin Li 및 Shunquan Tan.Can
법의학 탐지기가 GAN 생성 이미지를 식별합니까? 프로시저에서
아시아 태평양양 신호 및 정보 처리 협회
연례 정상 회담 및 회의(APSIPA ASC), 2018. 7
- [30] Lars Mescheder, Andreas Geiger, Sebastian Nowozin.
GAN에 대한 어떤 훈련 방법이 실제로 수렴합니까?
CorR, abs/1801.04406, 2018. 5, 11
- [31] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and
요시다 유이치. 생성적 adversarial 네트워크에 대한 스펙트럼 정규화. CorR,
abs/1802.05957, 2018. 1, 5, 6,
20
- [32] 드미트리 니키토. StyleGAN – 공식 TensorFlow 구현을 위한 인코더. <https://github.com/Puzer/stylegan> 인코더/, 2019. 7
- [33] 아우구스투스 오데나, 제이콥 벅만, 캐서린 올슨, 톰 B.
브라운, 크리스토퍼 올라, 콜린 라펠, 이안 굿펠로우.
발전기 조절이 GAN 성능과 인과 관계가 있습니까? CorR, abs/1802.08768,
2018. 5, 18

올라프 로네버거, 필립 피셔, 토마스 브룩스. U Net: 생물 의학 이미지 분할을 위한 컨볼루션 네트워크. 프로시저에서 의료 영상 컴퓨팅 및 컴퓨터 지원 종재 (MICCAI), 234-241페이지, 2015. 6

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, San Jeev Sathesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, 마이클 S. 번스타인, 알렉산더 C. 버그, 그리고 Fei-Fei Li. ImageNet 대규모 시각 인식 문제. 프로시저에서 CVPR, 2015. 4

[36] 메디 에스엠 사자디, 올리비에 바캤, 마리오 루치치, 올리비에 부스케, 실뱅 겔리. 생성 모델 평가 정확성과 리콜을 통해. CorR, abs/1806.00035, 2018. 1

[37] 팀 살리만스와 디에데릭 P. 킹마. 가중치 정규화: 훈련을 가속화하기 위한 간단한 한 재매개변수화 심층 신경망. CorR, abs/1602.07868, 2016. 3

[38] Yujun Shen, Jinjin Gu, Xiaoou Tang 및 Bolei Zhou 의미론적 얼굴 편집을 위한 GAN의 잠재 공간 해석. CorR, abs/1907.10786, 2019. 1

[39] Karen Simonyan 및 Andrew Zisserman. 대규모 이미지 인식을 위한 매우 깊은 회선 네트워크. 코르, abs/1409.1556, 2014. 1, 4

[40] Run Wang, Lei Ma, Felix Juefei-Xu, Xiaofei Xie, Jian Wang, 그리고 류양. FakeSpotter: 탐지를 위한 간단한 기준선 AI 합성 가짜 얼굴. CorR, abs/1909.06122, 2019. 7

[41] 왕성유, 왕 올리버, 장 리처드, 앤드류 Owens, Alexei A. Efros. CNN 생성 이미지는 지금은 놀라울 정도로 쉽게 발견할 수 있습니다. CorR, abs/1912.11035, 2019. 7

[42] 랜스 윌리엄스. 피라미드 매개변수. 시그나프 컴 놓다. 그래프., 17(3):1-11, 1983. 6

[43] 시타오 시양과 하오리. 배치 및 중량의 영향 생성적 적대 네트워크의 정규화. 코르, abs/1704.03971, 2017. 3

[44] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, Jianxiong Xiao LSUN: 대규모 이미지 구축 루프에서 인간과 함께 딥 러닝을 사용하는 데이터 세트. 코르, abs/1506.03365, 2015. 11

[45] 닝 유, 래리 데이비스, 마리오 프리츠. GAN에 가짜 이미지 귀속: 생성된 이미지에 서 지문 분석. CorR, abs/1811.08180, 2018. 7

[46] Han Zhang, Ian Goodfellow, Dimitris Metaxas 및 Augustus Odena. 자기 주의 생성 적대 네트워크. CorR, abs/1805.08318, 2018. 5

[47] Han Zhang, Tao Xu, Hongsheng Li, Shaoqing Zhang, Xiao lei Huang, Xiaogang Wang 및 Dimitris N. Metaxas. 스택 GAN: 스택을 사용하여 텍스트에서 사실적인 이미지 합성 생성적 적대 네트워크. ICCV에서, 2017. 6

[48] Han Zhang, Tao Xu, Hongsheng Li, Shaoqing Zhang, Xiaogang Wang, Xiao lei Huang, Dimitris N. Metaxas Stack GAN+: 스택 생성을 통한 사실적인 이미지 합성 적대 네트워크. CorR, abs/1710.10916, 2017. 6

[49] 리처드 장. 컨볼루션 네트워크를 다시 불변으로 이동시킵니다. 프로시저에서 ICML, 2019. 5, 11

[50] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, Oliver Wang. 불합리한 효과 시각적 지표로서의 심층 특징. 프로시저에서 CVPR, 2018. 4, 8, 19

[51] Xu Zhang, Svebor Karaman, Shih-Fu Chang. GAN 가짜 이미지에서 인공물을 감지하고 시뮬레이션합니다. 코르, abs/1907.06515, 2019. 7

A. 화질

우리는 이미지 품질과 관련된 다양한 측면을 보여주는 여러 개의 큰 이미지를 포함합니다. 그림 11은 손으로 고른 것을 보여줍니다. 달성 가능한 품질과 다양성을 보여주는 예 FFHQ에서 우리의 방법을 사용하는 반면, 그림 12는 논문에서 언급된 모든 데이터 세트에 대한 평가되지 않은 결과를 보여줍니다.

그림 13과 14는 FID와 P&R은 직관적이지 않은 결과를 제공하지만 PPL은 더 많은 것 같습니다. 인간의 판단에 따라.

StyleGAN 아티팩트와 관련된 이미지도 포함됩니다. 그림 15는 얼룩 아티팩트가 실패하는 드문 경우를 보여줍니다.

StyleGAN 활성화에 나타나므로 심각한

깨진 이미지. 그림 16은 내부 활성화를 시각화합니다.

표 1 구성 A 및 F. 점진적 성장은 중간 계층에서 더 높은 주파수 콘텐츠로 이어져 네트워크의 이동 불변성을 손상시키는 것이 분명합니다. 우리는 이것이 관찰된 불균일한 원인이 된다고 가정합니다.

점진적 성장이 있을 때 세부 사항에 대한 위치 선호 사용됨.

나. 시행내용

우리는 공식 위에 우리의 기술을 구현했습니다. 해당하는 StyleGAN5의 TensorFlow 구현 Z와 W의 차원을 포함하여 대부분의 세부 사항을 변경하지 않고 유지했습니다.

(512), 네트워크 아키텍처 매핑(완전히 연결된 8개 계층, 100배 더 낮은 학습률),에 대한 균등화된 학습률 모든 훈련 가능한 매개변수 [23], ReLU 활성화 누출 $\alpha = 0.2$, 모든 업/다운샘플링 레이어에서 쌍선형 필터링 [49] [24], 끝에서 미니 배치 표준 편차 레이어 판별자 [23], 발전기 가중치의 지수 이동 평균 [23], 스타일 혼합 정규화 [24], 비포화 로지스틱 손실 [16] 및 R1 정규화 [30],

동일한 하이퍼파라미터 ($\beta_1 = 0.9, \beta_2 = 0.99, \gamma = 10^{-8}$, 미니배치 = 32), 그리고 훈련 데이터 세트 [24, 44]. 우리는 모든 교육 실행을 수행했습니다. Tensor Flow 1.14.0 및 cuDNN 7.4.2를 사용하는 8개의 Tesla V100 GPU가 있는 NVIDIA DGX-1.

발전기 재설계 구성 B-F에서 우리는 다음을 대체합니다. 수정된 아키텍처가 포함된 원래 StyleGAN 생성기. 섹션 2에서 강조 표시된 변경 사항 외에도 N(0, 1)을 사용하여 상수 입력 c1의 구성 요소를 초기화합니다. 단일 사용을 위해 노이즈 브로드캐스트 작업을 단순화합니다. 모든 기능 맵에 대한 공유 배율 인수. Kar et al.과 유사합니다. [24], N(0, 1)을 사용하여 모든 가중치를 초기화하고 모두 우리가 초기화하는 아핀 변환 레이어의 바이어스를 제외하고 바이어스 및 노이즈 스케일링 계수를 0으로 하나에. 우리는 가중치 번조 및 복조를 사용합니다. 출력 레이어(tRGB의 tRGB)를 제외한 모든 컨볼루션 레이어

5https://github.com/NVLabs/stylegan

그림 7) 여기서 우리는 복조를 생략합니다. 10242로 출력 해상도, 생성기는 총 18개의 affine을 포함합니다. 첫 번째 레이어가 4에 해당하는 변환 레이어 해상도, 다음 두 개는 8에 해당합니다. ², 기타 등등.

가중치 복조 수학적 1 및 3의 실제 구현을 고려할 때 다음 사항에 유의하는 것이 중요합니다. 결과 가중치 세트는 미니 배치의 각 샘플에 대해 다르며 직접 구현을 배제합니다.

표준 컨볼루션 프리미티브를 사용합니다. 대신 우리는 원래 그룹화된 컨볼루션을 사용하기 위해 [26] 입력 특징 맵을 여러 개의 독립적인 그룹으로 나누어 계산 비용을 줄이는 방법으로 제안. 각각의 전용 가중치 세트가 있습니다. 우리는 구현 가중치를 일시적으로 재구성하고 수학적 1 및 3 각 컨볼루션이 하나의 샘플을 볼 수 있도록 활성화 N 그룹 — 하나의 그룹이 있는 N 샘플 대신. 이 접근 방식은 작업을 재구성하기 때문에 매우 효율적입니다. 가중치 및 활성화 텐서의 내용을 실제로 수정하지 마십시오.

Lazy regularization 구성 C-F에서 사용 k 학습 반복마다 한 번씩 실행하는 별도의 정규화 단계에서 정규화 용어(R1 및 경로 길이)를 평가하여 지연 정규화(섹션 3.1)를 수행합니다.

우리는 Adam 옵티마이저의 내부 상태를 공유합니다. 옵티마이저가 먼저 k 반복에 대한 기본 손실의 기울기를 볼 수 있도록 주 손실 및 정규화 항, 하나에 대한 정규화 항의 기울기가 뒤따릅니다. 반복. 우리가 지금 수행한다는 사실을 보장하기 위해 k 대신 k+1 훈련 반복, 우리는 옵티마이저를 조정합니다. $\lambda^0 = c \cdot \lambda, \beta_1^0 = (\beta_1) c, \text{ 그리고 } \beta_2^0 = (\beta_2) c$, 여기서 $c = k/(k+1)$. 우리는 또한 정규화를 곱합니다 그라디언트의 전체 크기의 균형을 유지하기 위해 k만큼 항. 판별자에는 k = 16을 사용하고 판별자에는 k = 8을 사용합니다. 발전기.

경로 길이 정규화 구성 D-F에는 다음이 포함됩니다. 우리의 새로운 경로 길이 정규화(섹션 3.2). 우리는 초기화 목표 스케일을 0으로 조정하고 GPU별로 추적합니다. J의 지수 이동 평균으로 β_1 와 β_2 부패를 사용하여 계수 $\beta_1 = 0.99$. 우리는 정규화 기간에 가중치를 씩니다.

$$\gamma_{pl} = \frac{\ln 2}{r 2(\ln r - \ln 2)}, \text{ 여} \quad (5)$$

기서 r은 출력 해상도를 지정합니다(예: r = 1024). 우리 이러한 매개변수 선택이 모든 구성 및 데이터 세트. 레귤러 라이저가 스타일 혼합 정규화와 올바르게 상호 작용하도록 하기 위해 합성 네트워크의 모든 개별 레이어의 평균으로 계산합니다. 부록 C는

W 간의 매핑에 대한 정규화가 미치는 영향 그리고 이미지 공간.



그림 11. StyleGAN2(구성 F)를 사용하여 얻을 수 있는 이미지 품질과 다양성을 보여주는 4가지 예.

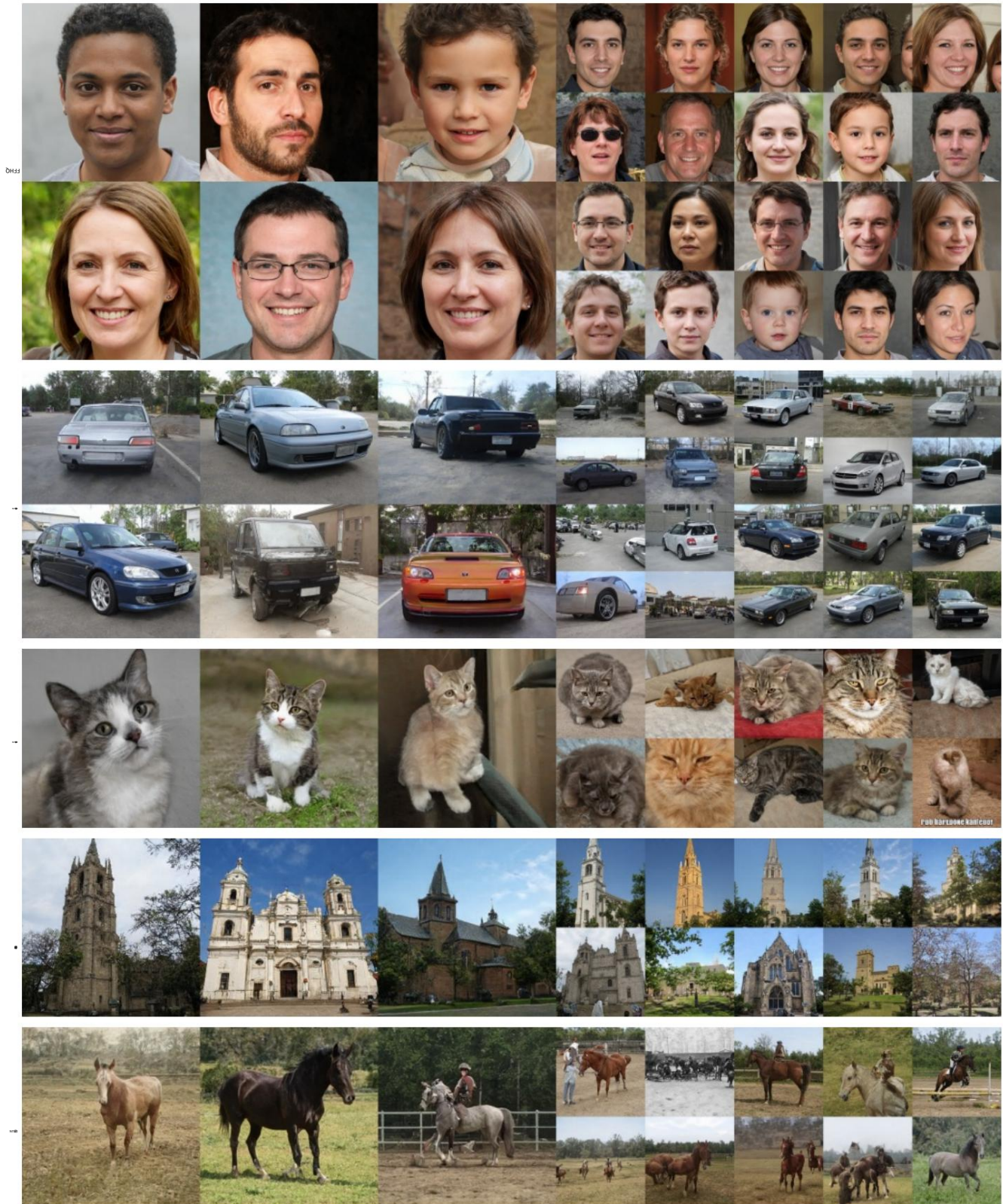
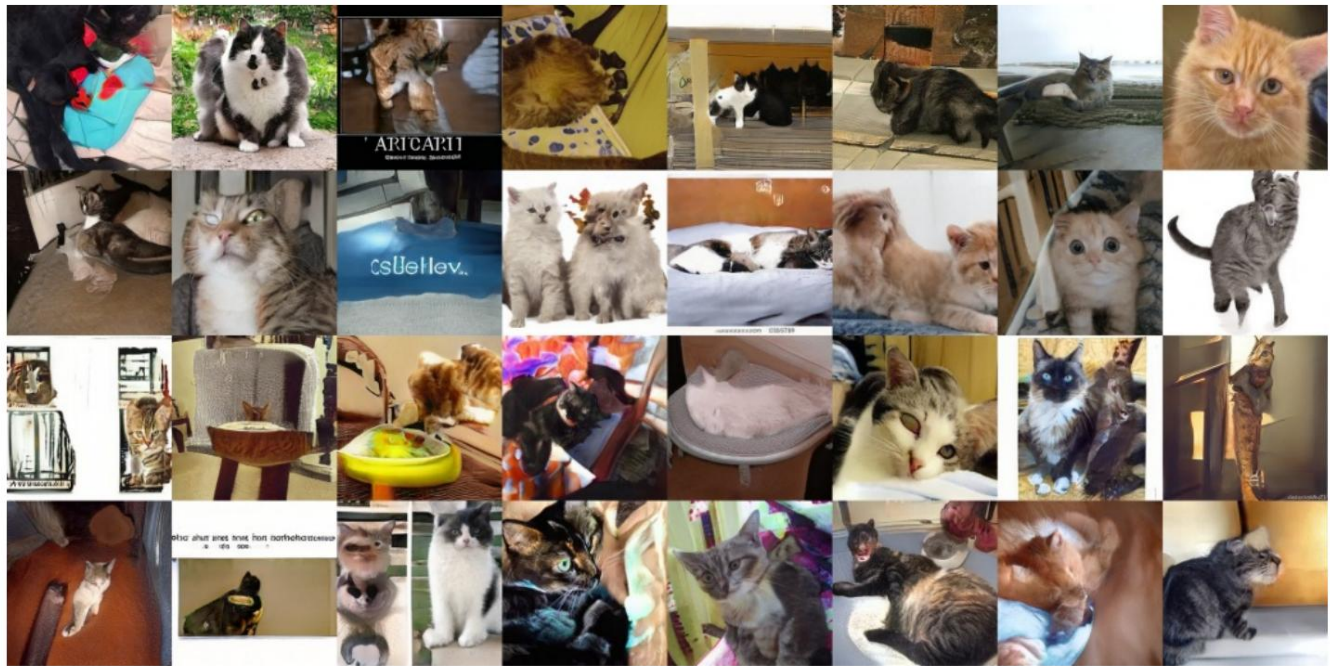
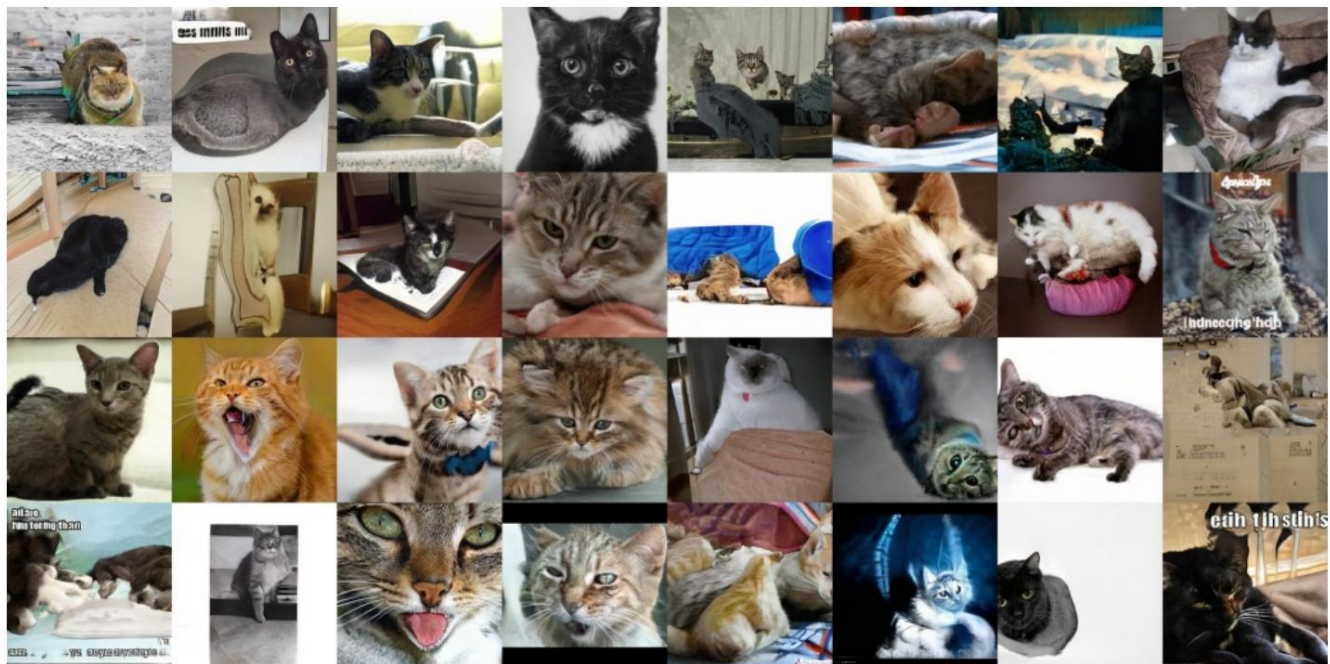


그림 12. 표 1 과 3에 사용된 각 데이터 세트에 대한 선별되지 않은 결과. 이미지는 생성기에 의해 생성된 무작위 출력에 해당합니다. (config F), $\psi = 0.5$ [24] 를 사용하여 모든 해상도에서 잘림이 적용됩니다.

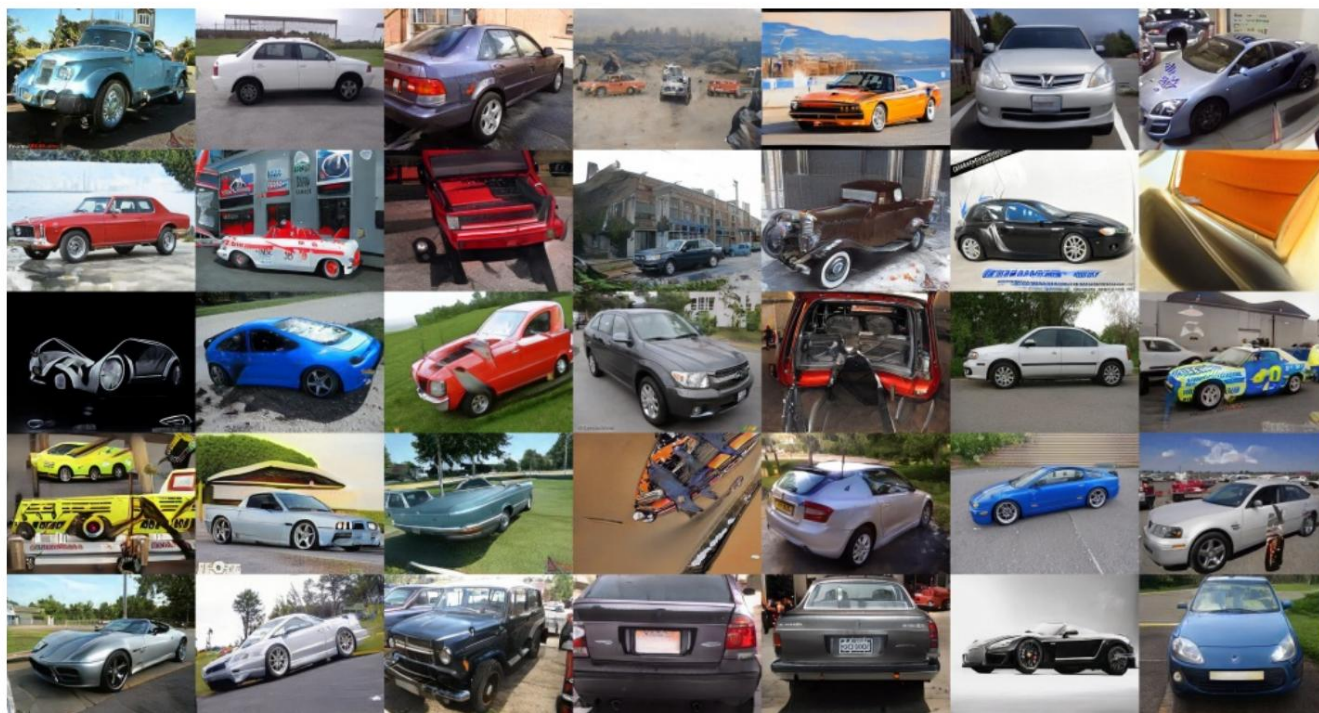


모델 1: FID = 8.53, P = 0.64, R = 0.28, PPL = 924

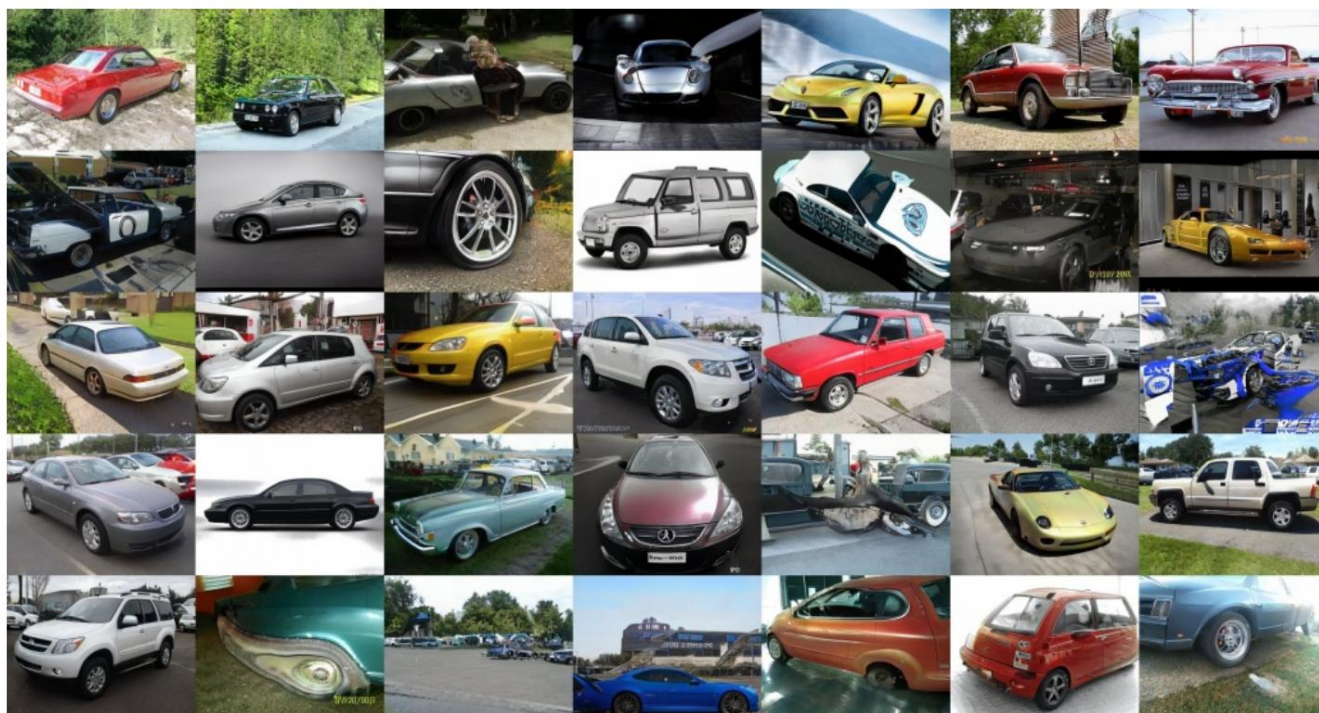


모델 2: FID = 8.53, P = 0.62, R = 0.29, PPL = 387

그림 13. 잘림 없이 LSUN CAT 에서 훈련된 두 생성 모델의 선별되지 않은 예 . FID, 정밀도 및 재현율은 모델 1과 2에서 유사하지만 후자는 고양이 모양의 물체를 더 자주 생성합니다. PPL(자각 경로 길이)은 모델 2에 대한 명확한 선호도를 나타냅니다. 모델 1 은 표 3의 구성 A 에 해당 하고 모델 2는 구성 F 의 초기 교육 스냅샷입니다.



모델 1: FID = 3.27, P = 0.70, R = 0.44, PPL = 1485



모델 2: FID = 3.27, P = 0.67, R = 0.48, PPL = 437

그림 14. LSUN CAR 에서 잘림 없이 훈련된 두 생성 모델의 선택되지 않은 예. FID, 정밀도 및 재현율은 모델 1과 2가 비슷하지만 후자는 자동차 모양의 물체를 더 자주 생성합니다. PPL(자카 경로 길이)은 모델 2에 대한 명확한 선호도를 나타냅니다. 모델 1은 표 3의 구성 A에 해당 하고 모델 2는 구성 F의 초기 교육 스탭입니다.

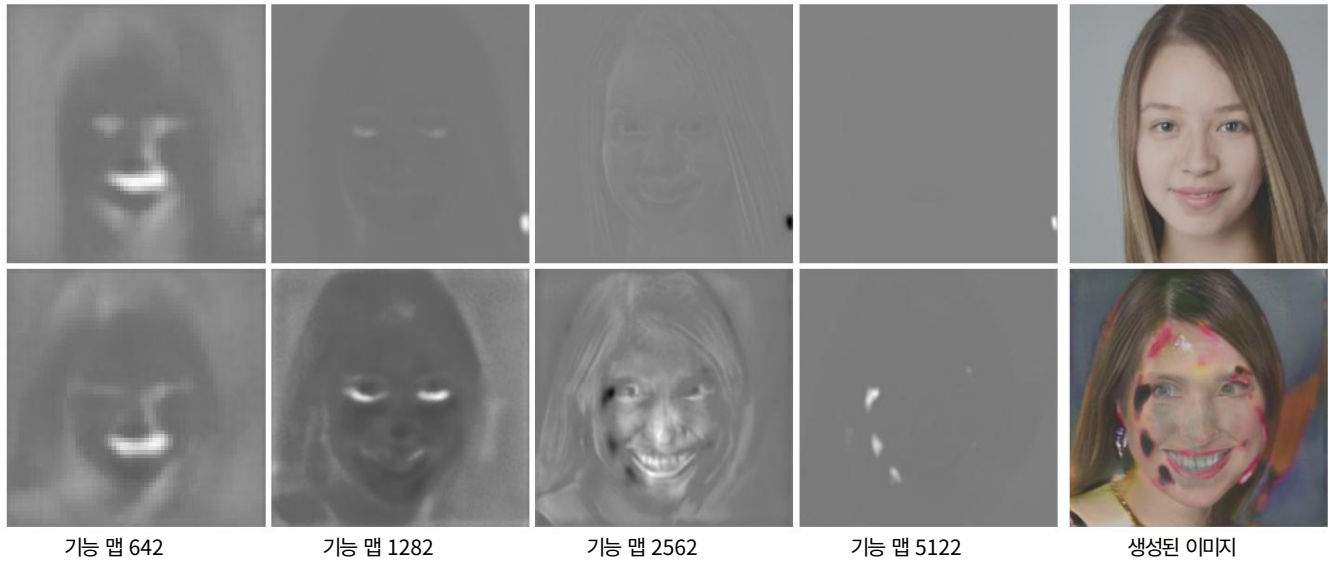


그림 15. StyleGAN 생성기에서 액티브 팩트의 중요성에 대한 예. 생성된 두 이미지를 비교합니다. 하나는 성공한 이미지입니다. 그리고 하나는 심하게 손상되었습니다. 해당 기능 맵은 인스턴스 정규화를 사용하여 볼 수 있는 동적 범위로 정규화되었습니다. 상단 이미지의 경우 액티브 인공물이 642 해상도에서 형성되기 시작하고 1282에서 명확하게 보입니다. , 점점 더 많은 기능을 지배합니다. 더 높은 해상도의 지도. 하단 이미지의 경우, 642는 상단 행과 질적으로 유사하지만 액티브는 1282에서 구체화되지 않습니다. 결과적으로 얼굴 특징은 정규화된 특징 맵에서 더 강합니다. 이는 2562에서 오버슈트로 이어집니다. , 뒤에 여러 후속 해상도에서 형성되는 스푸리어스 액티브. 우리의 경험에 따르면 StyleGAN 이미지에서 물방울이 누락되는 경우는 거의 없습니다. 그리고 실제로 발전기는 그 존재에 전적으로 의존합니다.

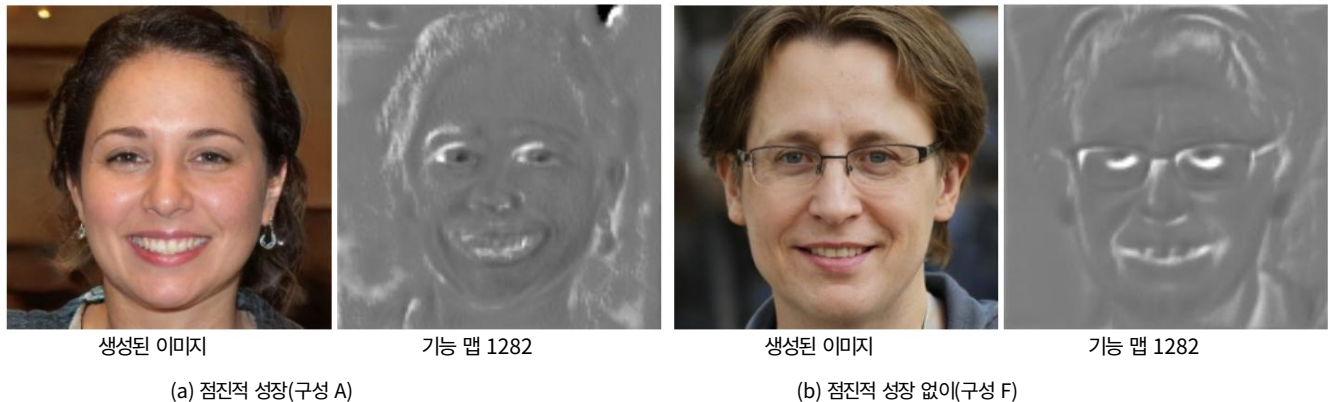


그림 16. 점진적인 성장은 중간 레이어에서 훨씬 더 높은 주파수 콘텐츠로 이어집니다. 이는 네트워크의 이동 불변성을 손상시키고 고해상도 레이어에서 기능을 정확하게 지역화하기 어렵게 만듭니다.

우리가 사용 하는 구성 A-D 에서 점진적 성장
Kar resolution 및 learning rate $\lambda = \frac{1}{2}$ 같은 매개변수로 점진적 성장
raset al. [24] (8시에 시작
10³, 해상도당 600k 이미지 훈련, 600k 이미지에 대한 다음 해상도 페이드
인, 학습률을 점진적으로 증가
3×). 구성 E-F 에서 점진적 성장을 비활성화하고 학습률을 고정 값 $\lambda = 2 \cdot 10^{-3}$ 으로 설정합니다.
최고의 결과를 제공하는 것으로 나타났습니다. 또한 우리는
생성기 및 잔류 연결에서 출력 건너뛰기를 사용합니다.
섹션 4.1 에 설명된 대로 판별자에서 .

데이터 세트별 조정 Karras et al.과 유사합니다. [24] 우리는
FFHQ 데이터 세트를 수평 플립으로 보강하여 훈련 이미지의 수를 70k에서 70k
로 효과적으로 늘립니다.
140k, 그리고 우리는
LSUN 데이터 세트. 우리는 최적의 선택이
훈련 길이 및 R1 정규화 가중치 γ 경향
데이터 세트와 구성 간에 상당히 다릅니다.
표 1의 구성 E와 LSUN CHURCH 및 LSUN 을 제외한 모든 훈련 실행에 대해
 $\gamma = 10$ 을 사용합니다.
 $\gamma = 100$ 을 사용하는 표 3의 HORSE . 가능합니다.
 γ 의 추가 조정은 추가 이점을 제공할 수 있습니다.

성능 최적화 우리는 훈련 실행을 광범위하게 프로파일링했으며 우리의 경우 이미지 필터링, 업/다운샘플링, 편향 추가 및 누출 ReLU에 대한 기본 요소가 훈련 시간 및 GPU 메모리 풋프린트 측면에서 놀랍도록 높은 오버헤드를 가지고 있음을 발견했습니다. 이것은 우리가 손으로 작성한 CUDA 커널을 사용하여 이러한 작업을 최적화하도록 동기를 부여했습니다. 필터링된 업/다운샘플링을 단일 융합 작업으로 구현하고 바이어스 및 활성화를 다른 작업으로 구현했습니다. 10242 해상도의 구성 E에서 최적화를 통해 전체 교육 시간이 약 30%, 메모리 공간이 약 20% 향상되었습니다.

C. 경로 길이 정규화의 효과

색션 3.2에 설명된 경로 길이 정규화 의 형식은 다음과 같습니다.

$$L_{pl} = E w E y J \quad \text{와이}^T \quad \text{2} \quad \text{- 에이}^2, \quad (6)$$

여기서 $y \in \mathbb{R}^M$ 은 생성된 이미지 공간(차원 $M = 3wh$, 즉 RGB 이미지 차원)의 단위 정규 분포 랜덤 변수이고, $J_w \in \mathbb{R}^{M \times L}$ 은 생성기 함수 g 의 야코비 행렬에서 $7 \rightarrow \mathbb{R}^M$ 입니다. R latent space point $w \in \mathbb{R}^L$ 이고, $a \in \mathbb{R}$ 은 디엔트 스케일을 나타내는 전역 값입니다. 엘 원하는 그라

C.1. 점별 야코비 행렬에 대한 영향

이 사진의 값은 y 에 대한 내부 기대가 모든 잠재 공간 점 w 에서 개별적으로 최소화될 때 최소화됩니다. 이 하위 섹션에서 우리는 야코비 행렬 J_w 가 전역 스케일링 인자까지 직교 할 때 내부 기대치가 (대략) 최소화된다는 것을 보여줍니다. 일반적인 전략은 고차원 L 에서 단위 정규 분포의 밀도가 반지름 \sqrt{L} 의 구형 껍질에 집중된다는 잘 알려진 사실을 사용하는 것입니다. 그런 다음 행렬 J 가 기대하는 함수일 때 내부 기대가 최소화됩니다. 이 r adius에서 최소 값을 갖습니다. 이것은 모든 직교 행렬에 의해 달성됩니다(모든 w 에서 동일한 적합한 전역 스케일 사용).

티 저울

우리는 내면의 기대를 고려하는 것으로 시작합니다

$$L_w := E y J \quad \text{티} \quad \text{와이}^T \quad \text{2} \quad \text{- 에이}^2.$$

먼저 y 분포와 $L2$ 노름의 방사 대칭으로 인해 대각선 행렬에만 집중할 수 있습니다. 이것은 특이값 $= U \Sigma V^T$ 를 사용하여 볼 수 있습니다. 분해 $JV \in \mathbb{R}^{M \times M}$ 은 직교 행렬이고 $\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_L \end{bmatrix}$ 은, 여기서 $U \in \mathbb{R}^{L \times L}$ 그리고

대각 행렬 $\Sigma \in \mathbb{R}^{L \times L}$ 과 영행렬 $0 \in \mathbb{R}^{L \times (M-L)}$ [15]의 수평 연결. 직교 $L \times L$ 행렬로 단위 법선 랜덤 변수를 회전하면 분포가 변경되지 않고 벡터를 회전하면

표준이 변경되지 않은 경우 표현식은 다음과 같이 단순화됩니다.

$$L_w = E y U \Sigma V^T \quad \text{와이}^T \quad \text{2} \quad \text{- 에이}^2 \\ = E y \Sigma y \quad \text{2} \quad \text{- 에이}^2.$$

게다가, Σ 의 0행렬은 y 의 차원을 L 너머로 떨어뜨리고, 그 차원에 대한 분포를 효과적으로 주변화합니다. 주변화된 분포는 다시 나머지 L 차원에 대한 단위 정규 분포입니다.

그런 다음 표현의 최소화를 고려해야 합니다.

$$L_w = E y \sim (k \Sigma y - k^2 - a)^2,$$

대각 정사각형 행렬 $\Sigma \in \mathbb{R}^{L \times L}$, 여기서 y 는 차원 L 에 정규분포된 단위입니다. 요약하자면, 모든 행렬 J

Σ 와 동일한 특이값을 공유하는 것은 원래 손실에 대해 동일한 값을 생성합니다.

다음으로, 대각 행렬 Σ 가 모든 대각 항목에서 동일한 특정 값을 가질 때, 즉 단위 행렬의 상수 배수인 경우 이 식이 최소화됨을 보여줍니다. 먼저 y 의 확률 밀도에 대한 적분으로 기대치를 작성합니다.

$$L_w = Z (k \Sigma y - k^2 - a)^2 \quad \text{파아} \sim (y \sim) \text{다아} \\ = (2\pi)^{-\frac{L}{2}} \int_{\mathbb{R}^L} \exp \left(-\frac{1}{2} (k \Sigma y - k^2 - a)^2 \right) \text{특급} - \frac{\text{티} \quad \text{와이} \quad \text{와이}}{2} \text{다아} dy$$

밀도의 방사 대칭 형태를 관찰하여 극좌표 $y = r \phi$ 로 변경합니다. 여기서 $r \in \mathbb{R}^+$ 는 원점으로부터의 거리이고 $\phi \in S^{L-1}$ 는 단위 벡터, 즉 $L-1$ 차원 상의 한 점입니다. 단위 구. 이 변화

변수의 야코비안 인자 r 도입 $L-1$:

$$L \sim w = (2\pi)^{-\frac{L}{2}} \int_0^\infty \int_{S^{L-1}} \exp \left(-\frac{1}{2} (r k \Sigma \phi - k^2 - a)^2 \right) r^{L-1} dr d\phi$$

확률 밀도 $(2\pi)^{-L/2}$ 는 극좌표로 표현된 L 차원 단위 법선 밀도이며, 각도 $\exp \left(-\frac{1}{2} (r k \Sigma \phi - k^2 - a)^2 \right)$ 가 아닌 반경에만 의존합니다. Taylor 근사에 의한 표준 인수는 L 이 높을 때 임의의 ϕ 에 대해 밀도가 밀도 $(2\pi e/L)$ $L/2$ 에 의해 잘 근사된다는 것을 보여줍니다.

특급 $-\frac{1}{2} (r - \mu)^2 / \sigma^2$, 이는 표준 편차 $\sigma = 1/\sqrt{2}$ 의 $\mu = \sqrt{L}$ 을 중심으로 하는 r 의 (정규화되지 않은) 1차원 정규 밀도입니다 [4]. 즉, L 차원 단위 정규분포의 밀도는 반지름 \sqrt{L} 의 껍질에 집중된다.

이 밀도를 적분에 대입하면 손실은 다음과 같습니다.

약

$$Lw \approx (2\pi\epsilon/L) \int_0^{\infty} \frac{(r k \Sigma \phi k^2 - a)^2}{r \sqrt{\rho}^2} dr d\phi, (7)$$

여기서 근사는 무한 차원 L 의 한계에서 정확해집니다.

이 손실을 최소화하기 위해 함수가 구형에서 최소값을 얻도록 Σ 를 설정 $(r k \Sigma \phi k^2 - a)^2$ 합니다.

반경 \sqrt{L} 의 셀. 이것은 $\Sigma = \sqrt{aL}$ 에 의해 달성됩니다. —나, 이로써 함수는 ϕ 에서 일정해지고 식은 다음과 같이 감소합니다.

$$Lw \approx (2\pi\epsilon/L) \int_0^{\infty} \frac{r \sqrt{\rho}^2}{2\sigma^2} dr d\phi, (8)$$

여기서 $A(S)$ 는 단위 구의 표면적입니다(그리고 다른 상수 요인과 마찬가지로 최소화하고 관련이 없음).

포물선의 영점($r \sqrt{L}$)은 최대 확률 밀도와 일치하므로 Σ 를 선택하면 Eq에서 내부 적분이 최소화됩니다. 모든 ϕ 에 대해 별도로 7.

요약하면, 잠재 공간의 높은 차원 L 을 가정하면 생성기의 야코비 행렬의 모든 특이값이 전역 상수와 같을 때 경로 길이 이전 값(Eq. 6)이 최소화 된다는 것을 보여주었습니다. , 모든 잠재 공간 점 w 에서, 즉, 전역적으로 일정한 규모까지 직교합니다.

이론적으로 a 는 속성을 변경하지 않고 매핑 핑의 값을 조정하고 고정 값 (예: 1)으로 설정할 수 있지만 실제로는 훈련의 역학에 영향을 미칩니다. 부과된 척도가 네트워크의 무작위 초기화에 의해 유도된 척도와 일치하지 않으면 훈련은 실제 관심 목표를 시행하기 보다는 가중치를 필요한 전체 크기로 푸시하는 데 중요한 초기 단계를 소비합니다. 이것은 네트워크 가중치의 내부 상태를 저하시키고 이후 훈련에서 최적이지 아닌 성능으로 이어질 수 있습니다. 경험적으로 우리는 고정 척도를 설정하면 훈련 실행 및 데이터 세트 전반에 걸쳐 훈련 결과의 일관성이 감소한다는 것을 발견했습니다. 대신, 우리는 야코비 행렬의 기존 척도의 실행 평균, 즉 $a \approx E_{w,y} J$ 를 기반으로 동적으로 설정합니다. 특정 글로벌 평균. 이것은 또한 야코비 행렬의 적절한 척도를 측정할 필요를 없애줍니다.

$$\frac{1}{\sigma^2} \int_0^{\infty} \frac{r \sqrt{\rho}^2}{2\sigma^2} dr d\phi, (9)$$

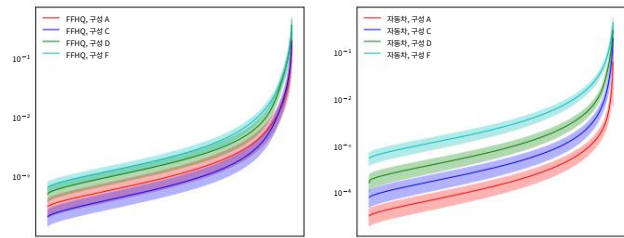


그림 17. 최대 고유값이 1로 정규화된 무작위 잠재 공간 점 w 에서 평가된 야코비 행렬의 정렬된 특이값 크기의 평균 및 표준 편차.

두 데이터 세트 모두에서 경로 길이 정규화(Config D)와 새로운 아키텍처(Config F)가 더 나은 조건을 나타냅니다. 특히, 훨씬 더 많은 가변성을 포함하고 경로 길이 정규화가 PPL 메트릭에 상대적으로 더 강한 영향을 미치는 Cars 데이터 세트에서 효과가 더 두드러집니다(표 1).

명시적으로 Oden et al. [33] 관련 조건화를 사전에 고려하는 사람.

그림 17은 경로 길이 정규화를 사용하거나 사용하지 않고 훈련된 네트워크에 대해 경험적으로 측정된 야코비 행렬의 특이값 크기를 보여줍니다. 직교성에는 도달하지 않았지만, 정규화된 네트워크의 고유값은 서로 더 가깝고, 이는 더 나은 조건을 의미하며 효과의 강도는 PPL 메트릭과 상관관계가 있습니다(표 1).

C.2. 제너레이터 맵의 전역 속성에 미치는 영향

이전 하위 섹션에서 우리는 사전이 생성기 매핑의 야코비 행렬이 모든 곳에서 직교하도록 권장한다는 것을 발견했습니다. 그림 17은 매핑 ping이 실제로 이 제약 조건을 정확히 충족하지 않는다는 것을 보여주지만 제약 조건이 충족하는 매핑에 대해 의미하는 전역 속성을 고려하는 것이 좋습니다. 일반성을 잃지 않고 표현을 단순화하기 위해 행렬에 대한 단위 전역 규모를 가정합니다. $7 \rightarrow RM$ 핵심 속성은 모든 곳에서 직교 야코비 행렬이 있는 매핑 $g:R$ 이 길이를 보존한다는 것입니다.

엘

곡선의. 이를 확인하기 위해 $u: [t_0, t_1] \rightarrow$ 잠재 공간 \mathcal{W} 에서 R 곡선. 생성기 g 를 통해 곡선을 매핑하면 이미지 공간에서 곡선 $u \sim g \circ u$ 를 얻습니다. 그것의 호 길이는

$$\text{패} = \int_{t_0}^{t_1} \sqrt{\langle \dot{u}, \dot{u} \rangle} dt, (8)$$

여기서 소수는 t 에 대한 도함수를 나타냅니다. 연쇄법칙에 의해 이것은 같음

$$\text{패} = \int_{t_0}^{t_1} \sqrt{\langle Jg(u(t)) \dot{u}, Jg(u(t)) \dot{u} \rangle} dt, (9)$$

여기서 $Jg \in R^{L \times M}$ 에서 평가된 g 의 야코비 행렬입니다. 우리의 가정에 따르면, 야코비 행렬은 직교하고,

결과적으로 벡터 u 의 2-노름을 남깁니다. $^{\circ}(t)$ 나프
감염:
$$\|u\|_2 = \sqrt{\int_{t_0}^{t_1} |u^{\circ}(t)|^2 dt.} \quad (10)$$

이것은 잠재 공간에서 곡선 u 의 길이입니다.
 g 로 매핑 따라서 u 와 u° 의 길이는 같습니다.
따라서 g 는 모든 곡선의 길이를 유지합니다.

미분 기하학의 언어에서 g 등적성
유클리드 잠재 공간 R 을 포함합니다. RM 에서 하위 매니폴드 M 으로 — 예를 들어, 가능한 모든 RGB 공간 내에 포함된 얼굴을 나타내는 이미지의 매니폴드
이미지. 아이소메트리의 결과는 잠재 공간의 직선 세그먼트가 이미지 매니폴드의 측지선 또는 최단 경로에 매핑된다는 것입니다. 두 개의 잠재 공간 점을 연결하는 직선 v 는 더 짧게 만들 수 없으므로

더 짧은 은매니폴드 이미지 공간 경로도 있을 수 없습니다.
예를 들어, 얼굴 이미지의 다양체에 대한 측지선은 최소 총계를 발생시키는 두 점 사이의 연속적인 모프입니다.

변화량(RGB의 l2 차이로 측정)
space) 각 단계의 이미지 차이를 요약할 때
모프의.

Isometry는 실제로 달성되지 않습니다.
이전 하위 색선의 경험적 실험. 전체
훈련의 손실 함수는 잠재적으로
상충되는 기준이 있으며, 진정으로 등각 투영법 매핑이 관심 있는 사람의 이미지를 표현할 수 있는지 여부는 분명하지 않습니다. 그럼에도 불구하고 맵 핑을 가능한 등적성으로 만들려는 압력은 바람직한 결과를 가져옵니다. 예

특히, 불필요한 "우회"를 방지합니다. 비제약 생성기 매핑에서 잠재 공간 보간

두 개의 유사한 이미지 사이는 임의의 숫자를 통과할 수 있습니다.
RGB 공간에서 먼 이미지의. 정규화를 통해
매핑은 멀리 떨어진 이미지를 다른 위치에 배치하는 것이 좋습니다.
짧은 이미지 경로를 얻기 위해 잠재 공간의 영역
두 끝점 사이.

D. 투사 방법 세부 사항

대상 이미지 x 가 주어지면 해당 $w \in W$ 및 $n_i \in R^{r_i \times r_i}$ 로 표시된 레이어별 노이즈 맵을 찾습니다.
여기서 i 는 레이어 인덱스이고 r_i 는 해상도를 나타냅니다.
1번째 노이즈 맵. 기본 StyleGAN 생성기
1024×1024 해상도에는 18개의 노이즈 입력이 있습니다.
4×4에서 1024×1024 픽셀까지의 해상도. 우리의 개선된
아키텍처에는 노이즈 입력이 하나 더 적습니다.
학습된 4×4 상수에 노이즈를 추가합니다(그림 2).
최적화 전에 매핑 네트워크 f 를 통해 10,000개의 임의 잠재 코드 z 를 실행하여 $\mu_w = E_z f(z)$ 를 계산합니다. 우리는 또한 계산하여 W 의 규모를 근사화합니다.
 $\sigma_w^2 = E_z \|f(z) - \mu_w\|_2^2$. $\frac{1}{2}$, 즉, 평균 제곱 유클리드 중심까지의 거리.

최적화 시작 시 $w = \mu_w$ 를 초기화합니다.
모든 i 에 대해 $n_i = N(0, I)$ 입니다. 훈련 가능한 매개변수는

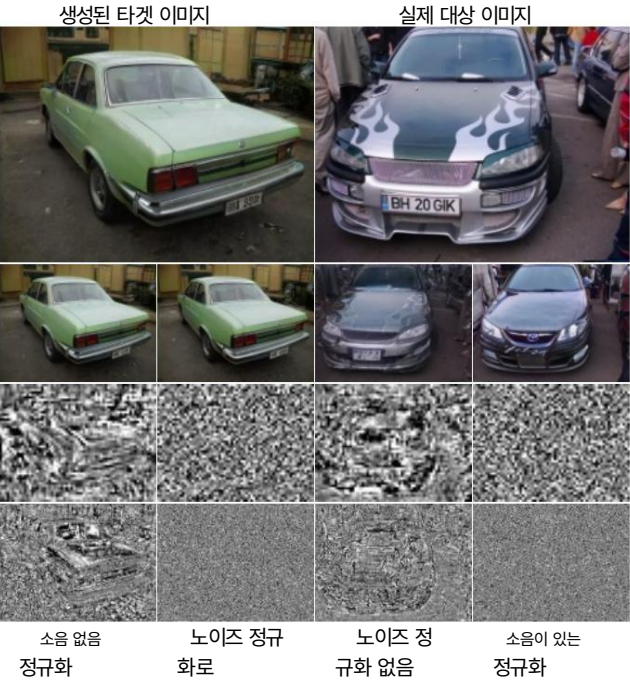


그림 18. 잠재 공간 투영에서 노이즈 정규화의 효과
여기서 우리는 또한 노이즈 입력의 내용을 최적화합니다.
합성 네트워크. 위에서 아래로: 대상 이미지, 재합성된 이미지, 해상도가 다른 두 노이즈 맵의 내용. 현재
이 테스트에서는 정규화가 꺼져 있으며 노이즈만 정규화합니다.
최적화를 이끄는 제로 평균 및 단위 분산에 매핑
신호를 노이즈 맵에 몰래 넣습니다. 노이즈 정규화를 활성화하면 이를 방지할 수 있습니다. 여기에 사용된 모델은 표 1의 구성 F에 해당합니다.

w 의 구성 요소와 모든 노이즈의 모든 구성 요소
용하여 Adam 옵티마이저(2.6)를 사용하여 100,000번의 학습 단계를 반복하며, σ_w^2 를 사용하여

처음 50회 반복 동안 선형적으로 0이 되고 감소합니다.
지난 250회 반복 동안 코사인 일정을 사용하여 0으로 만듭니다. 최적화의 처음 3/4 분기에는 다음을 추가합니다.
손실 함수를 다음과 같이 평가할 때 w 에 대한 가우시안 잡음 $\tilde{w} = w + N(0, 0.05 \sigma_w^2)$, 여기서 t 는 1에서 0으로 이동합니다.
처음 750번의 반복 동안. 이것은 stochasticity를 추가합니다.
최적화하고 전역 최적의 찾기를 안정화합니다.

노이즈 맵을 명시적으로 최적화하고 있다는 점을 감안할 때,
최적화가 실제 신호를 몰래 들어가지 않도록 주의해야 합니다. 따라서 우리는 몇 가지 노이즈를 포함합니다.
손실 함수의 정규화 항을 매핑하고
이미지 품질 용어로. 이미지 품질 용어는
LPIPS [50] 대상 이미지 x 와 합성 크기 이미지 사이의 거리: $L_{\text{image}} = \text{DLPIPS}(x, g(\tilde{w}, n_0, n_1, \dots))$. 을 위한
성능과 안정성이 향상되면 두 가지 모두를 다운샘플링합니다.
LPIPS를 계산하기 전에 이미지를 256×256 해상도로
거리. 노이즈 맵의 정규화는 다음에서 수행됩니다.

	SN-G	SN-D	Demod	P.reg	FID ↓	PPL ↓	Pre. ↑	기록. ↑
1	-	-	-	XX	2.83	145.0	0.689	0.492
2	13	1.4	0.700	0.468	2.98			
3	XX	3.40	130.9	0.720	0.435			
4	XX	-	-	X	3.38	162.6	0.705	0.468
5	XX	-	-	-	3.33	394.9	0.705	0.463
6	X	-	-	X	3.36	217.1	0.695	0.464
7	X	-	-	-	3.22	394.4	0.692	0.489

표 4. 1024에서 FFHQ를 사용한 스펙트럼 정규화의 효과 .
첫 번째 행은 StyleGAN2, 즉 표 1의 구성 F에 해당합니다.
후속 행에서 스펙트럼 정규화를 활성화합니다.
생성기(SN-G) 및 판별기(SN-D)에 있습니다. 우리는 또한 테스트
가중치 복조(Demod) 및 경로 길이 없는 훈련
정규화(P.reg). 이러한 모든 구성은 FID와 Recall에 매우 해롭습니다. ↑는 높을 수
록 좋은 것이고,
↓ 낮을수록 좋습니다.

다중 해상도 스케일. 이를 위해 우리는
크기가 8×8보다 큰 각 노이즈 맵은 피라미드 아래로
2×2 픽셀 이웃을 평균화하여 8×8 해상도로
예상 단위를 유지하기 위해 각 단계에서 2를 곱합니다.
변화. 이러한 다운샘플링된 노이즈 맵은 정규화에만 사용되며 합성에는 포
함되지 않습니다.
원래 노이즈 맵을 $n_{i,0} = n_i$ 로 표시 하고
 $n_{i,j} > 0$ 으로 다운샘플링된 버전 . 유사하게, $r_{i,j}$ 를
원본($j = 0$) 또는 다운샘플링($j >$
0) $r_{i,j+1} = r_{i,j}/2$ 가 되도록 노이즈 맵 . 정규화
노이즈 맵 $n_{i,j}$ 에 대한 항은 다음과 같습니다.

$$r_{i,j} = \frac{1}{2^j} \cdot \sum_{(x,y) \in \mathcal{N}_{i,j}} n_{i,j}(x,y) \cdot n_{i,j}(x-1,y)^2 + \frac{1}{2^j} \cdot \sum_{(x,y) \in \mathcal{N}_{i,j}} n_{i,j}(x,y) \cdot n_{i,j}(x,y-1)^2,$$

노이즈 맵은 가장자리에서 래핑되는 것으로 간주됩니다. 그만큼
따라서 정규화 항은 한 픽셀 이동에서 해상도 정규화된 자기상관 계수의
제공의 합입니다.
수평 및 수직으로, 일반적으로 분포되지 않은 신호의 경우 0이어야 합니다.
전체 손실 기간은 다음과 같습니다.
 $L_{total} = L_{image} + \alpha P$ 리, J. 모든 테스트에서 우리는
사용된 잡음 정규화 가중치 $\alpha = 105$. 또한 우리는
모든 노이즈 맵을 제로 평균 및 단위 분산으로 재정규화
각 최적화 단계 후. 그림 18은 효과를 보여줍니다.
결과 노이즈 맵에 대한 노이즈 정규화

E. 스펙트럼 정규화 결과

스펙트럼 정규화(SN)가 널리 사용되기 때문에
GAN [31]에서 StyleGAN2에 미치는 영향을 조사했습니다. 표 4는 다
양한 구성에 대한 결과를 제공합니다.
스펙트럼 정규화는 우리의 기술(가중치 복조, 경로 길이 정규화)에 추가로
활성화됩니다.
그들 대신.

항목	GPU 년(볼타) 20.25	전기(MWh)
초기 탐색 페이퍼 탐색	13.71 0.23	58.94
FFHQ 구성 F 기타 페이	7.20 4.73	31.49
퍼 실행 백업 실행 누락	0.31 4.62	0.68
동영상, 파규어 등		16.77
		12.08
		0.82
공개 릴리스		10.82
총	51.05	131.61

표 5. 이 프로젝트의 계산 노력 지출 및 전력 소비 데이터. 계산 단위는 단일 NVIDIA
V100 GPU의 GPU 연도입니다. 단일 GPU를 사용하여 이 프로젝트를 실행하려면
약 51년이 걸렸을 것입니다.

계산 및 에너지 소비 추정치에 대한 자세한 내용은 텍스트를 참조하십시오. 초기 탐
색에는 이 프로젝트를 시작하기로 한 결정에 영향을 준 StyleGAN [24] 릴리스 이
후의 모든 훈련 실행이 포함됩니다 . 종이 탐색에는 모든 교육이 포함됩니다.

이 프로젝트를 위해 특별히 수행되었지만 문서에서 있는 그대로 사용되지 않는 실행.
FFHQ 구성 F는
최종 네트워크의 훈련. 대략적인 비용입니다
하이퍼파라미터가 없는 다른 데이터 세트에 대한 네트워크 훈련
동조. 문서의 다른 실행은 문서에 표시된 다른 모든 네트워크 작업의 교육을 다룹니
다. 누락된 백업 실행에는 잠재적으로 표시될 수 있는 다양한 네트워크의 교육이 포
함됩니다.
그러나 궁극적으로 박람회 를 더 많이 유지하기 위해 제외되었습니다.
집중했다. 비디오, 수치 등은 소비된 계산을 포함합니다.
종이에 이미지와 그래프를 만드는 것뿐만 아니라
결과 영상. 공개 릴리스는 테스트, 벤치마킹 및
공개 릴리스와 관련된 대규모 이미지 덩어리.

흥미롭게도 발전기에 스펙트럼 정규화를 추가하는 것은 거의 불가능하
입니다. 구현 수준에서 SN
스칼라 값으로 각 레이어의 가중치 텐서를 조정합니다.
 $1/\sigma(w)$. 그러나 그러한 스케일링의 효과는 무시됩니다.
메인 컨볼루션 레이어에 대한 식 3과
아핀 변환 레이어 따라서 유일한 것은
SN은 그 효과를 통해 가중치 복조 위에 추가됩니다.
tRGB 레이어에서
판별자에서 스펙트럼 정규화를 활성화하면 FID가 약간 손상됩니다. 예
서 활성화
제너레이터는 결과를 훨씬 더 나쁘게 만듭니다.
그 효과는 tRGB 레이어에 격리됩니다. SN을 떠나다
활성화되지만 기여의 하위 집합을 비활성화하는 것은
상황을 개선합니다. 따라서 우리는 StyleGAN2가
스펙트럼 정규화 없이 더 나은 결과를 제공합니다.

바. 에너지 소비

컴퓨팅은 모든 기계 학습의 핵심 리소스입니다.
프로젝트: 가용성 및 비용, 관련
에너지 소비는 연구 방향을 선택하고 실제 채택하는 데 있어 핵심 요소입니
다. 우리는 표 5에 전체 프로젝트에 대한 상세한 분석을 제공합니다.

GPU 시간과 전력 소비 모두.
소모된 계산 노력을 단일 GPU로 보고합니다.
년(Volta 클래스 GPU). 우리는 다양한 수를 사용했습니다.

프로젝트의 여러 단계를 위한 NVIDIA DGX-1 및 단순히 각 실행을 단일 GPU에 해당하는 것으로 변환했습니다. 사용된 GPU 수에 따른 확장.

전체 프로젝트는 약 131.61을 소비했습니다. 메가와트시(MWh) 전기. 우리는 따라갔다 Green500 전력 측정 지침 [11]은 다음과 같습니다. 각 작업에 대해 정확한 기간, 작업 수를 기록했습니다. 사용된 GPU 및 두 개의 개별 컴퓨팅 클러스터 중 작업이 실행된 클러스터. 그런 다음 훈련 중일 때 8-GPU DGX-1의 실제 전력 소모량을 측정했습니다. FFHQ 구성 F. 두 클러스터는 서로 다른 DGX-1 SKU를 사용하기 때문입니다. 그만큼 대부분의 훈련 실행은 8개의 GPU를 사용했으며 나머지는 선형으로 스케일링하여 전력 소비를 근사화했습니다. 여기서 n 은 GPU 수입니다.

전체 에너지의 약 절반이 초기에 소비되었습니다. 탐색 및 형성 아이디어. 그런 다음 4분의 1은 더 표적화된 실험에서 이러한 아이디어를 개선하는 데 사용되었으며 마지막으로 1/4은 이 문서를 작성하는 데 사용되었습니다. 코드, 훈련된 모델의 공개 릴리스 준비, 큰 이미지 세트. 단일 FFHQ 네트워크 훈련 (구성 F)는 약 0.68MWh(총 프로젝트 지출의 0.5%)를 사용했습니다. 이것은 하나의 비용입니다. 네트워크를 처음부터 훈련할 때 비용을 지불하고 다른 데이터 세트를 사용할 수 있습니다. 간단히 말해서, 사용된 전기의 대부분은 아이디어를 구체화하고 가설을 테스트하는 데 사용되었습니다. 및 초매개변수 조정. 자동화된 도구를 사용하지 않았습니다. 하이퍼파라미터를 찾거나 네트워크 아키텍처를 최적화합니다.