Large Scale GAN Training for High Fidelity Natural Image Synthesis (BigGAN)

시간	11:20 - 11:40
논문링크	https://arxiv.org/abs/1809.11096
발표동영상	http://gofile.me/6UrAI/l4rV9sEvH
발표자	㉾ 김하연
발표자료	

하연님 개인 페이지 정리 : Large Scale GAN Training for High Fidelity Natural Image Synthesis (BigGAN)

▼ Q&A

- embedding : 사람들의 언어 → 기계가 보다 쉽게 읽을 수 있게 변환하는 과정
  (shared embedding: embedding network하나를 공유해서 쓰는 게 memory-efficient.
- **truncated trick*에서 샘플링을 하는 시점이 학습할 때 해줘야 하나? - 학습할 때 안하고 학습된 모델에서 z noise를 건드려봄. (유진)**

  ☐ 코드 올려놓기**

요약 :

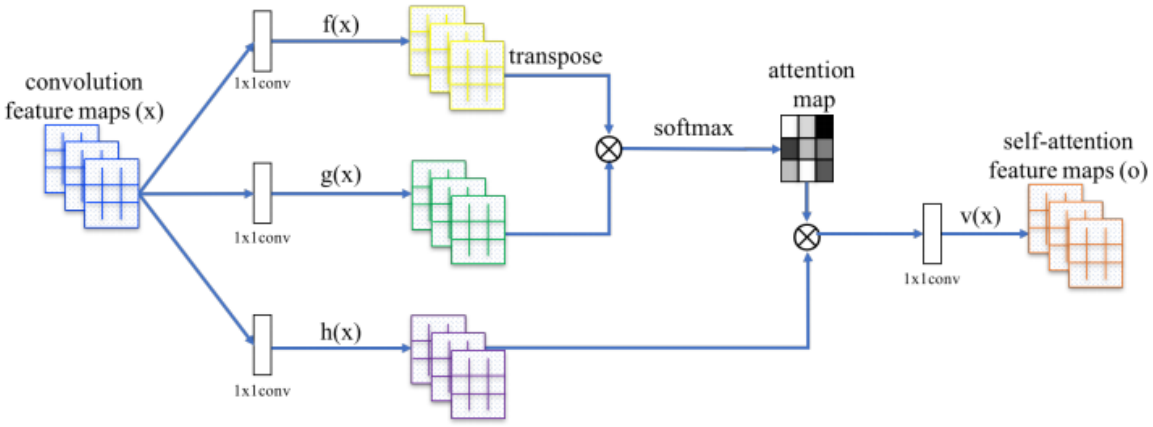
1) SA-GAN을 바탕으로 batch size, channel 수 증가 → 스코어 상승

2) Shared Embedding, Hierarchical Latent Space, Orthogonal Regularization으로 스코어 상승

3) Truncated Trick으로 적당한 Threshold를 골라야 한다.

1. SA-GAN

: 기존 GAN은 오로지 convolution에 강하게 의존해왔으나, Conv net은 구조상 local receptive field를 가지고 있기 때문에 이미지에서 멀리 떨어져 있는 부분에 대한 dependency를 알기 위해서는 많은 Conv 레이어를 거쳐야 한다. 따라서 작은 모델들은 long range dependency를 표현할 수 없게 되므로, self-attention GAN이 등장하였다.(즉, 공간적으로 멀리 떨어져있는 모델링에 효과적)



여기서 image feature를 x라고 했을때, $x \in R^{C \times N}$ (N = width $\times$ height)

여기서 x는 attention을 계산하기 위해서 먼저 두개의 feature space f,g로 변환되어집니다.

$$f(x) = W_f x, g(x) = W_g x$$

$$\beta_{i,j} = \frac{exp(s_{ij})}{\sum_{i=1}^{N} exp(s_{ij})}, \quad where \quad s_{ij} = f(x_i)^T g(x_j),$$

여기서 $\beta_{i,j}$는 모델이 jth region과 합성할 때 ith region에 집중하는 정도(i번째와 j번째 region의 상관정도)를 나타냅니다. → 위 과정으로 attention map이 만들어집니다.

최종 output :

$$o = (o_1, o_2, ..., o_j, ..., o_N) \in \mathbb{R}^{CxN}$$

$$where, o_j = v\left(\sum_{i=1}^{N} \beta_{i,j} \mathbf{h}(\mathbf{x_i})\right), \mathbf{h}(\mathbf{x_i}) = \mathbf{W_h x_i}, \mathbf{v}(\mathbf{x_i}) = \mathbf{W_v x_i}$$

여기서 $h(x_i)$는 i번째 region에 대한 정보를 담고 있고, $\beta_{i,j}$는 i번째 region과 j번째 region을 상관관계를 나타냄으로, 둘을 곱한 값은 어느부분이 부각되어야 하는지를 attention정도를 곱해주어 중요도를 높이거나 낮추는 작업을 하는 것입니다.

따라서, 결과인 o_j는 j번째 region이 전체 region과 얼마만큼의 관계가 있는가를 나타냅니다.

2. Shared Embedding, Hierarchical Latent Space, Orthogonal Regularization

- shared embedding : This reduces computation and memory costs, and improves training speed

  각각의 layer에 따로 embedding을 가지는 것 보다 shared embedding을 사용하여 각각 레이어의 gain과 bias에 linearly projected된다.

- hierarchical latent space :

> we employ hierarchical latent spaces, so that the **latent vector z is split along its channel dimension into chunks** of equal size (20-D in our case), and each chunk is concatenated to the shared class embedding and passed to a corresponding residual block as a conditioning vector. The conditioning of each block is linearly projected to produce per-sample gains and biases for the Batch Norm layers of the block. The bias projections are zero-centered, while the gain projections are centered at 1.

  기존 cGAN에서는 입력 노이즈를 최초의 층만 전달하지만, 여기에서는 그 입력 노이즈를 split하여 각 ResBleck에 전달하고 있다.

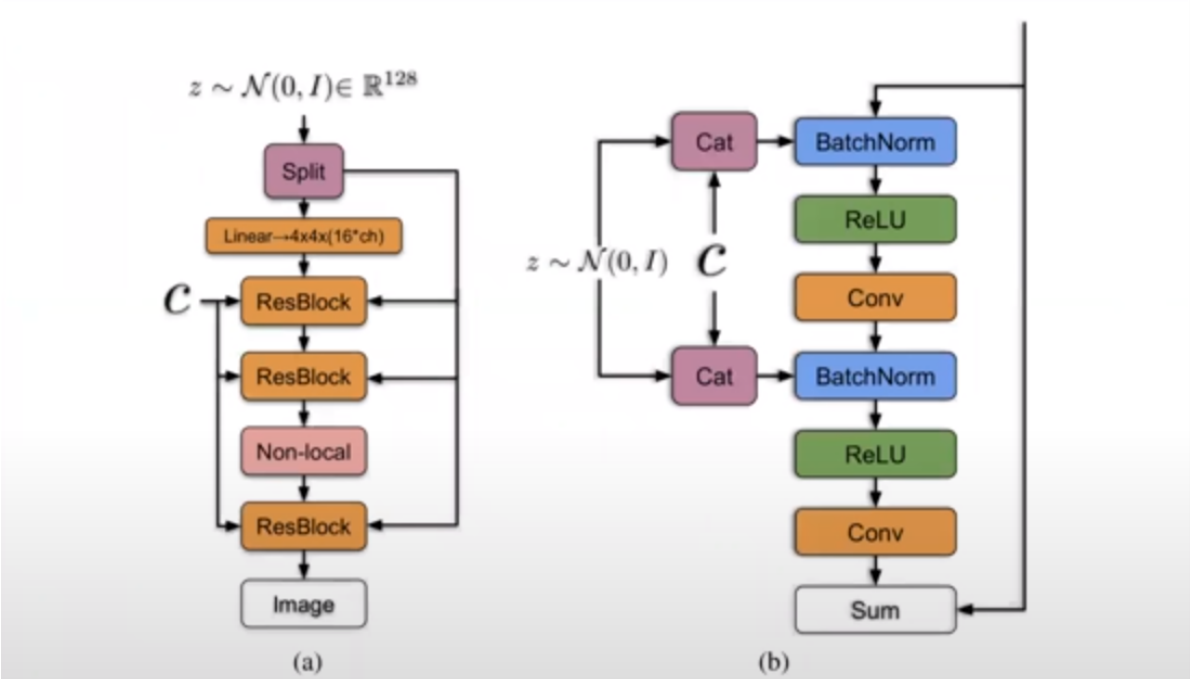
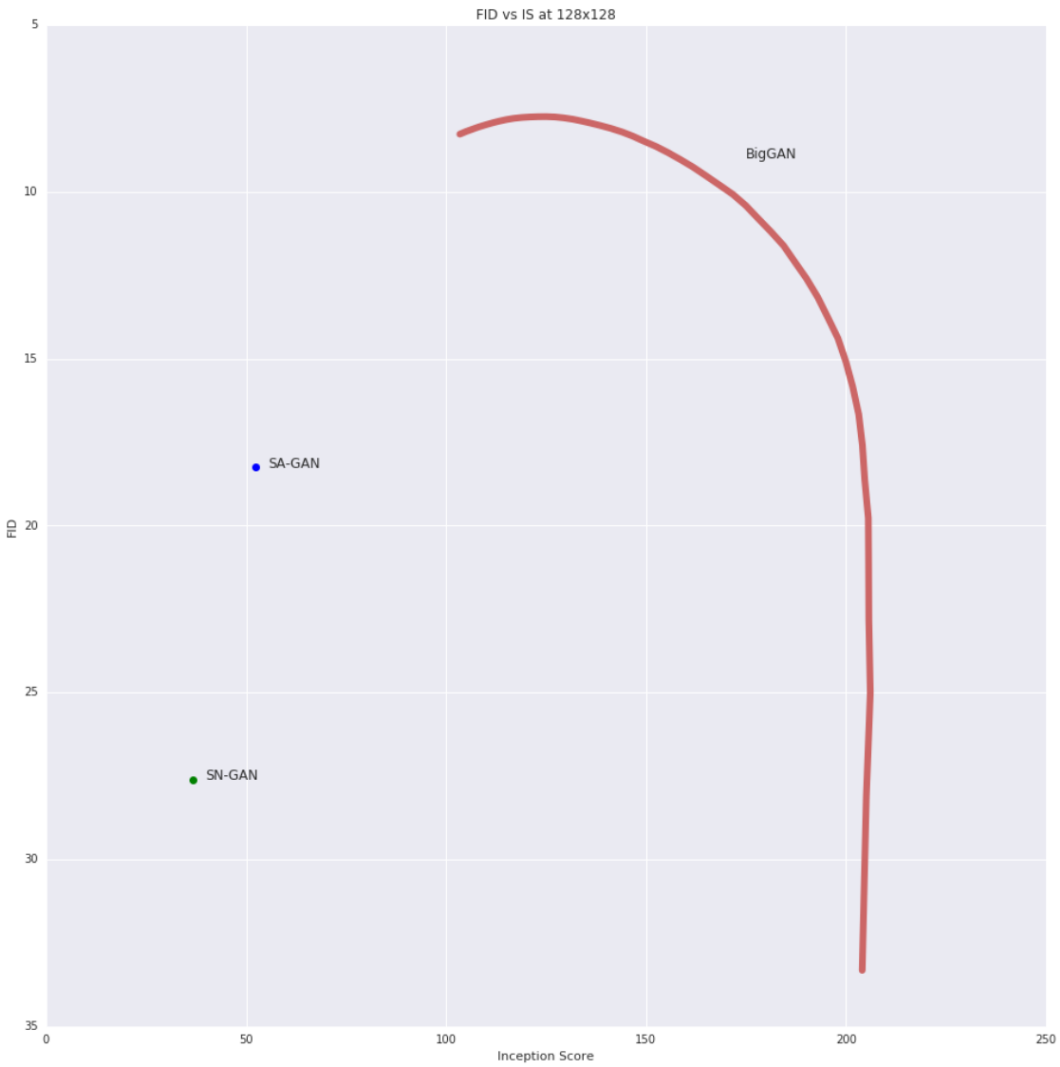


Figure 15: (a) A typical architectural layout for **G**; details are in the following tables. (b) A Residual Block in **G**. c is concatenated with a chunk of z and projected to the BatchNorm gains and biases.

3. Truncated Trick

   : GAN에서는 입력 노이즈는 N(0,1)부터 샘플링하지만, 이 샘플링한 값 중 Threshold를 넘는 것은 재샘플링하여 Threshold 안에 포함되도록한다. → threshold가 작을수록 노이즈의 분산은 더 좁게 된다.(다양성은 감소하나 퀄리티는 높아진다)

   위 그림에서 왼쪽에서 오른쪽으로 갈수록 threshold 값이 작아지고 있는데, threshold값이 작아질수록 다양성은 손실된다. 반대로, 가장 클때는, 질이 떨어진다.



다양한 threshold로 학습하여 FID와 IS를 구하는 것은 위 그림과 같은데, FID는 작을수록, IS는 클수록 좋은 성능을 가지므로, 적당한 threshold가 필요하다. → **orthogonal regularization** 필요

기존 regularization은 제한적이기 때문에, 우리의 model을 smooth하게 학습시키기에는 조금 부족했다. → 새로운 버전필요

The version we find to work best removes the diagonal terms from the regularization, and aims to minimize the pairwise cosine similarity between filters but does not constrain their norm:

$$R_\beta(W) = \beta \|W^\top W \odot (\mathbf{1} - I)\|_{\mathrm{F}}^2$$

> where 1 denotes a matrix with all elements set to 1. We sweep β values and select 10−4, finding this small added penalty sufficient to improve the likelihood that our models will be amenable to truncation.