WGAN / WGAN-GP

Generative Deep Learning

GAN의 도전 과제와 개선방향

손실이 일정치 않고 진동하거나 모드 붕괴 현상 개선 필요

- **Mode Collapse**
  - 손실 함수의 Gradient가 0에 가까운 값으로 붕괴된다(Collapse)
  - 생성자가 판별자를 속이는 적은 수의 샘플을 찾을 때 일어남
    >생성자가 다양한 출력을 만들지 않게 됨

- **WGAN**
  - Wasserstein거리에 의한 손실함수의 설계
    > 요소를 만족하기 위해 가중치를 클리핑
    > 학습이 불안정한 문제

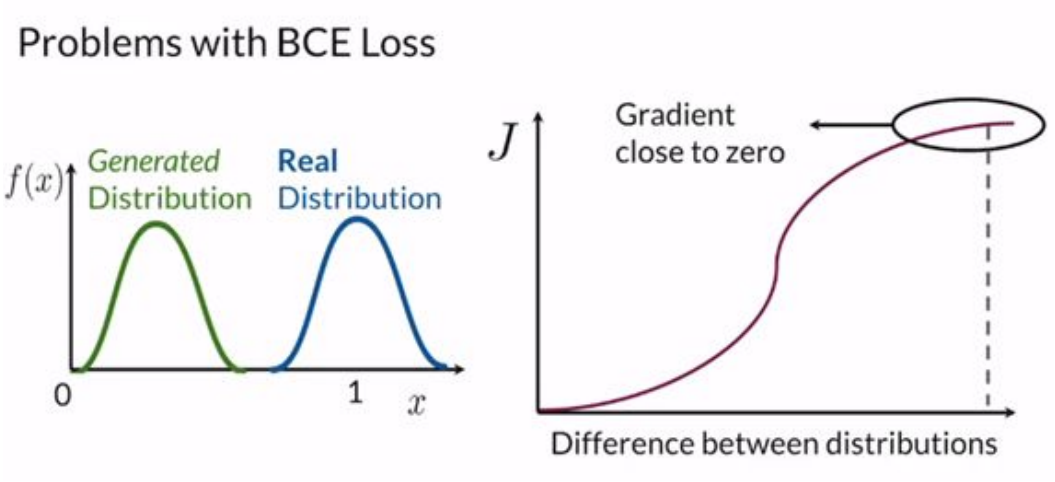
- **WGAN-GP**
  - Gradient penalty를 도입

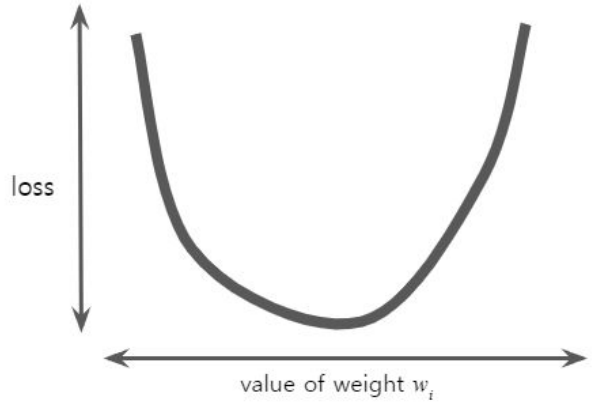
* 생성자는 판별자를 항상 속이는 하나의 샘플(이를 Mode라고 부른다)을 찾으려는 경향이 있다.

Binary Cross Entropy의 문제점 - Gradient Vanishing

학습 중에 판별자가 너무 잘 개선된다면 Gradieng Vanishing 문제가 발생할 수 있음

- 판별자 D의 학습이 잘 될수록 생성되는 분포 $Pmodel(x)$와 $Pdata(x)$를 훨씬 더 구별하게 됩니다.
- 실제 데이터 분포는 $Pdata(x)$에 가까이 위치하고, 생성되는 분포 $Pmodel(x)$는 0에 근접하게됩니다. 결과적으로 판별자 D가 잘 학습될수록 0에 가까운 gradient를 넘겨주게 되고, 이는 생성자 G입장에서 유익하지 않은 피드백입니다. 결국 생성자는 학습을 종료하게됩니다.





loss : 정답값과 예측값의 차이
gradient : loss 함수의 기울기

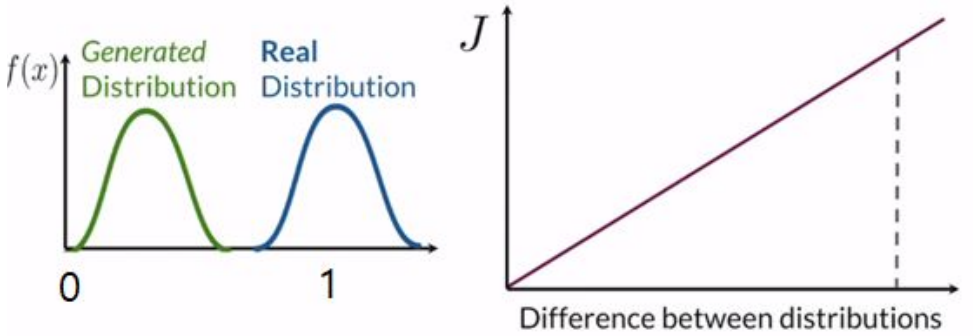
출처: https://velog.io/@tobigs-gm1/basicofgan

BCE loss 문제점 개선 - Wasserstein loss (Earth Mover's Distance)

Earth Mover's Distance: 분포가 얼마나 다른지를 나타내는 수치
Pmodel(x)을 Pdata(x)와 동일하게 만들기 위해 이동해야하는 거리와 양

BCE loss의 문제는 판별자 D가 sigmoid를 사용하여 real/fake의 확률값을 출력으로하여, 판별자가 좋아짐에 따라 두 분포의 차이가 심해져서 cost function의 기울기가 0값인 영역에 위치해 vanishing gradient문제가 생긴다는 것이었습니다.

하지만, Earth Mover's Distance의 결과는 0과 1의 한계가 없습니다. 분포가 얼마나 멀리 떨어져 있든 상관없이 의미있는 gradient(0이 아닌 기울기)를 전달할 수 있습니다. Earth Mover's Distance를 loss에 사용하면 결과적으로 Vanishing Gradient해결할 수 있고, Model collapse가능성을 감소시킴



출처: https://velog.io/@tobigs-gm1/basicofgan

손실함수의 변화 (binary cross entropy -> Wasserstein loss)

•GAN 손실 함수

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

•Wasserstein 손실 함수

- Wasserstein loss는 target을 1과 0 대신 1과 –1 사용
- 마지막 층에서 시그모이드 활성화 함수를 제거하여 예측 범위가 [0, 1]로 국한되지 않고 [–∞, ∞] 범위의 어떤 숫자도 될 수 있음
- 손실함수에 log를 이용하지 않음
- Discriminator의 대신에 Critic(비평가)라 부름

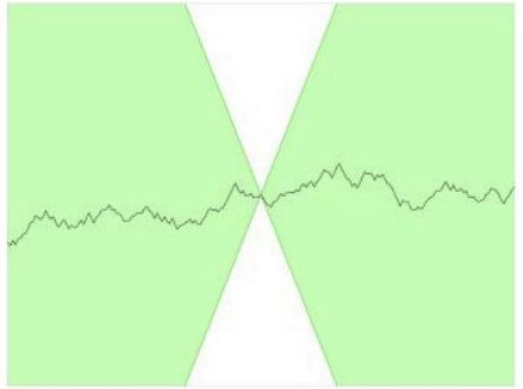
$$L = \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]$$

WGAN – Lipschitz 제약

Wasserstein 손실은 제한이 없어 큰 값일 수 있기에[−∞, ∞], Critic에 제약이 필요

•Lipschitz 제약

- Earth Mover's Distance의 출력이 [0, 1]로 제한되지 않아 loss에 사용하는것이 적적하다는 것을 알아보았는데, 일반적으로 신경망에서 너무 큰 숫자는 피해야 하기 때문에**Lipshitz 제약**이라는 제약조건을 걸어줍니다.



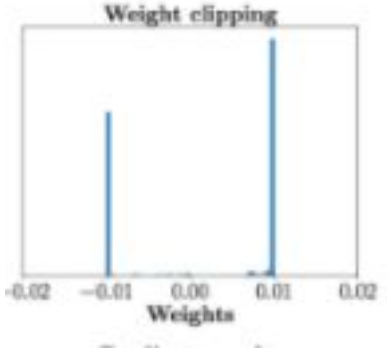
- **비평자 C (판별자 D)가 1-Lipshitz continuous function(1-립시츠 연속함수)이어야 합니다.**

- $|D_{x1}-D_{x2}|$는 비평자 C 예측 간의 차이이고,$x1-x2$는두 이미지의 픽셀의 평균값 차이이다.
  1-립시츠 연속함수의 경우 기울기의 절대값의 최대는 1입니다.
  함수 위 어느 점에 원뿔을 놓더라도 하얀색 원뿔에 들어가는 영역이 없습니다.
  다른 말로 하면 아래 직선은 어느 지점에서나 상승하거나 하강하는 비율이 한정되어있습니다.

$$\frac{|D(x_1) - D(x_2)|}{|x_1 - x_2|} \leq 1$$

•가중치 클리핑

- Critic의 가중치를 [−0.01, 0.01] 안에 놓이도록 가중치 클리핑을 통해 립시츠 제약을 부과함



WGAN의 개선 필요성

•기본 GAN과 차이점

- 기본 GAN은 gradient 소실을 피하기 위해 판별자가 너무 강해지지 않도록 하는 것이 중요함 (*Discriminator.Trainable = False)

  >Wasserstein 손실을 이용하면 이런 어려움을 제거 할 수 있음

  > 일반적으로 생성자를 업데이트 하는동안 Critic을 여러번 업데이트 함

•단점

- Critic에서 가중치를 클리핑했기 때문에 학습 속도가 크게 감소함

  (논문의 저자도 립시츠 제약을 두기 위해 가중치를 클리핑 하는 것은 좋지 않은 방법이다 언급)

- 강한 Critic은 성공의 핵심. 정확한 gradient가 없다면 생성자의 학습이 어려워짐

  –> 가중치 클리핑 보다 다른 방법이 필요

WGAN과 생성자는 동일하고, 비평자(Critic)가 차이가 있음
비평자에 립시츠 제약을 강제하는 다른 방법을 제안함

• WGAN-GP의 Critic

- 비평자 손실 함수에 gradient penalty항을 포함
- 비평자 가중치를 클리핑하지 않음
- 비평자에 배치 정규화 층을 사용하지 않음

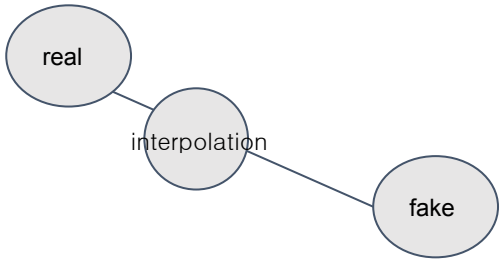
• Gradient Penalty

- 가중치를 클리핑하는 대신 비평자의 gradient norm이 1에서 크게 벗어날 때 페널티를 부과하는 항
  > 입력 이미지에 대한 예측의 gradient L2 norm과 1 사이의 차이를 제곱한 것

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}\left[D(\tilde{\boldsymbol{x}})\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}\left[D(\boldsymbol{x})\right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}.$$

WGAN-GP – Gradient penalty loss

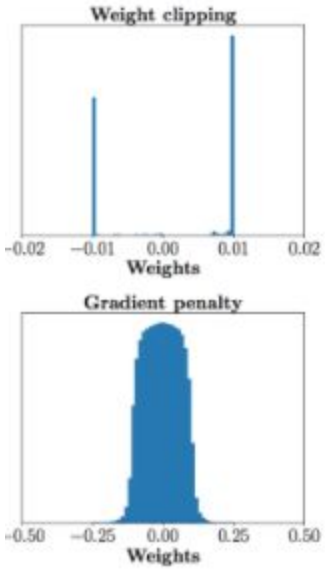
Gradient penalty loss : 비평자에 1-Lipschitz 제약을 강제하는 다른 방식



- 훈련과정에서 모든 곳에서 gradient를 계산하기는 불가능하여, 일부 지점에서만 gradient를 계산한다

- real image와 fake image 사이의 보간된 포인트에서 계산한 gradient와 1사이의 차이를 제곱하여 반환

- 미분가능한 함수는 모든 곳에서 gradients norm이 1이어야만 1-Lipschitz이다.



출처: https://leechamin.tistory.com/232

Norm은 벡터의 길이 혹은 크기를 측정하는 방법(함수)

•L1 Norm

- 벡터의 요소에 대한 절대값의 합

$$L_1 = (\sum_i^n |x_i|)$$
$$= |x_1| + |x_2| + |x_3| + \ldots + |x_n|$$

- L1 Loss : 실제 값과 예측치 사이의 차이(오차) 값의 절대값 구하고 그 오차들의 합

$$L = \sum_{i=1}^{n} |y_i - f(x_i)|$$

•L2 Norm

- L2 Norm은 n 차원 좌표평면(유클리드 공간)에서의 벡터의 크기
- 2차원 좌표 평면상의 최단 거리를 계산

$$L_2 = \sqrt{\sum_i^n x_i^2}$$
$$= \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2}$$

- L2 Loss : 오차의 제곱의 합

$$L = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

* y_i 는 실제 값을, f(x_i)는 예측치를 의미

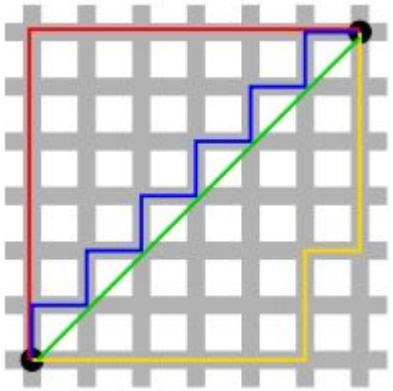
L1 Norm은 Feature selection이 가능하고, L2 Norm은 Unique shortest path를 가진다

•L1 & L2 Feature

- 검정색 두 점사이의 L1 Norm 은 빨간색, 파란색, 노란색 선으로 표현 될 수 있고, L2 Norm 은 오직 초록색 선으로만 표현될 수 있다
- L1 Norm 은 여러가지 path 를 가지지만 L2 Norm 은 Unique shortest path 를 가진다

  > 즉, L2 Norm 은 각각의 벡터에 대해 항상 Unique 한 값을 내지만,

  L1 Norm 은 경우에 따라 특정 Feature(벡터의 요소) 없이도 같은 값을 낼 수 있다



- L1 Norm 은 파란색 선 대신 빨간색 선을 사용하여 특정 Feature 를 0으로 처리하는 것이 가능하다고 이해할 수 있고, L1 Norm 은 Feature selection 이 가능하고 이런 특징이 L1 Regularization 에 동일하게 적용 될 수 있다

출처: https://light-tree.tistory.com/125