GAN을 이용해 데이터 불균형 해소

MNIST

★GAN의 기본 개념

GAN은 두 개의 인공신경망, 즉 생성자(Generator)와 판별자(Discriminator)로 구성됩니다.

생성자는 실제와 유사한 데이터를 생성하는 역할을 하며, **판별자**는 주어진 데이터가 실제인지 가짜인지 구별하는 역할을 합니다.

GAN에서 'Adversarial'이라는 용어는 생성자와 판별자가 서로 적대적인 관계에 있기 때문에 붙여졌습니다.

생성자는 판별자를 속이기 위해 노력하고, 판별자는 생성자가 만든 가짜 이미지를 식별하기 위해 노력합니다.

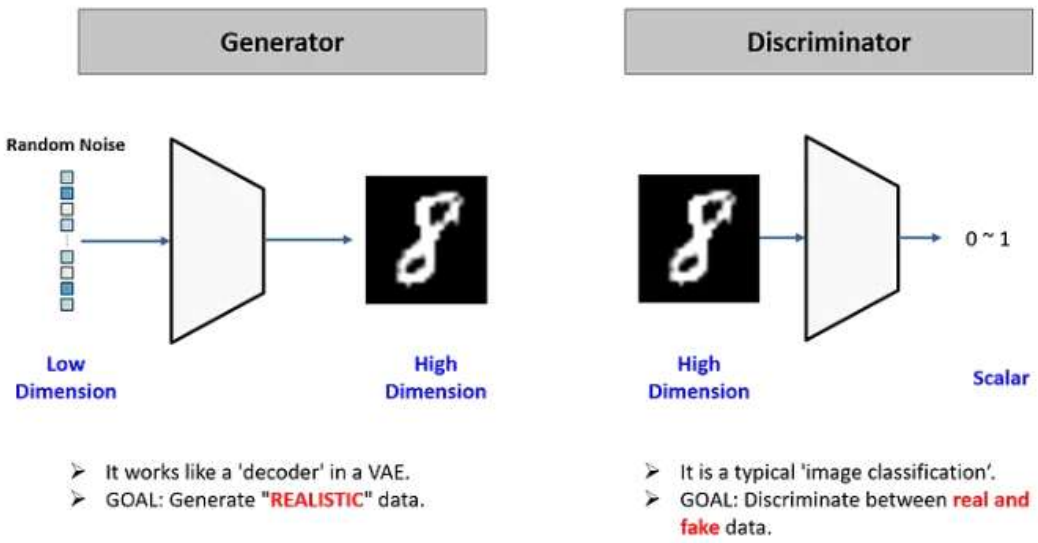
이 적대적인 관계는 두 모델이 동시에 발전하도록 자극하며, 결과적으로 더 정교하고 현실적인 데이터를 생성할 수 있게 만듭니다.

★생성자와 판별자의 역할

생성자(Generator): 랜덤한 노이즈 벡터를 입력받아 실제와 유사한 이미지를 생성합니다.

판별자(Discriminator): 생성자가 만든 이미지와 실제 이미지를 입력받아, 이를 구별합니다.

이 두 신경망은 반복적인 학습 과정을 통해 서로의 성능을 향상시키며, 결국 생성자는 판별자가 구별할 수 없을 정도로 실제와 유사한 데이터를 생성하게 됩니다.



⭐GAN의 학습 과정

GAN의 학습은 두 신경망의 경쟁을 통해 이루어집니다.

생성자는 판별자를 속이기 위해 더욱 정교한 이미지를 생성하고, 판별자는 이를 구별하기 위해 더욱 정확하게 학습합니다.

⭐학습 단계

랜덤 노이즈 생성: 생성자에게 랜덤한 노이즈 벡터를 입력합니다.

가짜 이미지 생성: 생성자는 이 노이즈 벡터를 바탕으로 가짜 이미지를 생성합니다.

실제 및 가짜 이미지 데이터 생성: 실제 이미지와 가짜 이미지를 적절히 섞어 판별자에게 입력합니다.

판별자 학습: 판별자는 실제 이미지를 1로, 가짜 이미지를 0으로 구별하도록 학습합니다.

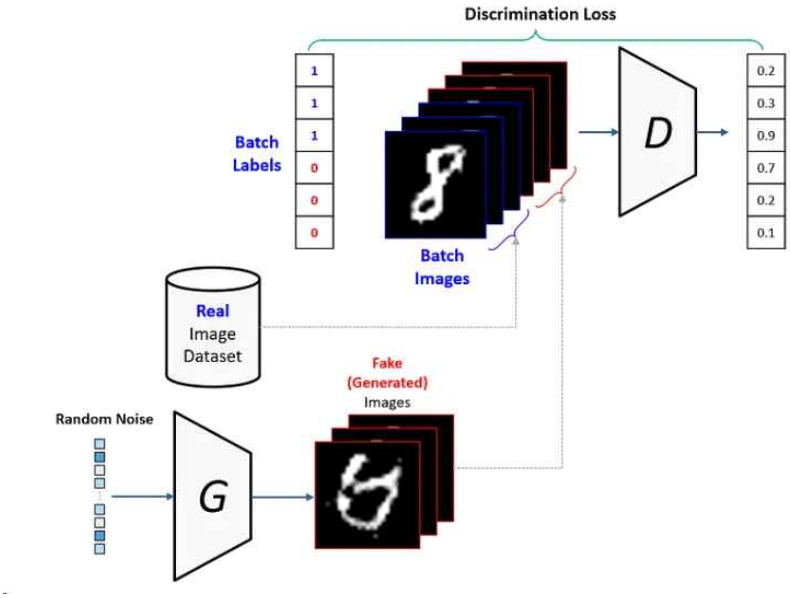
생성자 학습: 생성자는 판별자를 속이기 위해 가짜 이미지를 더욱 정교하게 생성하도록 학습합니다.

★판별자 훈련

위에서 정리한 데이터들을 이용해 판별자를 훈련시킬 수 있습니다.

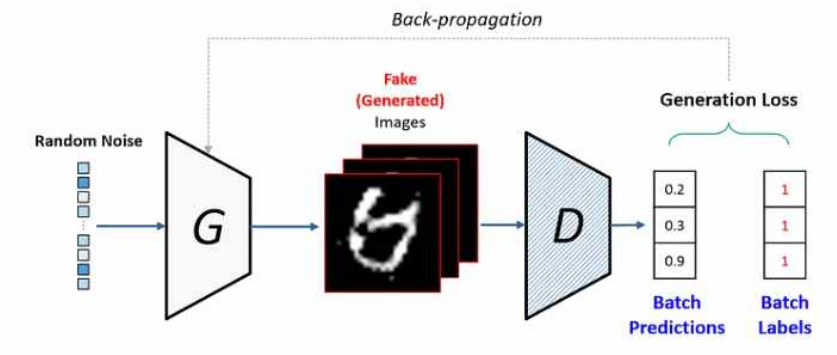
데이터의 비율은 디자인에 따라 달라질 수 있으나 가장 단순하게는 실제 이미지와 생성자에 의해 생성된 이미지를 5:5 로 섞어 훈련시킵니다.

판별자는 실제 이미지에 대해서는 1에 가까운 값을, 생성자가 생성한 이미지에 대해서는 0에 가까운 값을 나타내도록 학습됩니다.



★생성자 훈련
생성자의 경우에는 직접적으로 좋은 이미지인지 아닌지를 가르쳐줄 수 없으므로 자신이 생성한 이미지가 판별자를 속일 수 있도록 하는 학습 방향을 취하게 됩니다.

즉 생성자 입장에서는 생성된 이미지에 해당하는 최종 Label 값이 모두 1로 되어 계산되는 형태로 진행되면 됩니다.



⭐GAN의 장단점

GAN은 매우 강력한 생성 모델이지만, 몇 가지 단점도 가지고 있습니다.

-장점
고품질의 이미지 생성: GAN은 매우 정교하고 현실적인 이미지를 생성할 수 있습니다.

다양한 응용 가능성: 이미지 생성, 데이터 증강, 스타일 변환 등 다양한 분야에 활용할 수 있습니다.

-단점
학습 불안정성: GAN의 학습은 매우 불안정할 수 있으며, 적절한 하이퍼파라미터 설정이 필요합니다.

모드 붕괴: 생성자가 특정 유형의 이미지만 생성하고 다양한 이미지를 생성하지 못하는 문제가 발생할 수 있습니다.

손실값 모니터링의 어려움: GAN의 손실 함수를 모니터링하기 어려워 학습 과정을 추적하는 데 어려움이 있습니다.

★학습 진행

MNIST 1:1장 2:300장 3:300장

```python
class SmallCNN(nn.Module):
    def __init__(self, num_classes=2):
        super().__init__()
        self.net = nn.Sequential(
            nn.Conv2d(1, 16, 3, padding=1), nn.ReLU(), nn.MaxPool2d(2),
            nn.Conv2d(16, 32, 3, padding=1), nn.ReLU(), nn.MaxPool2d(2),
            nn.Flatten(),
            nn.Linear(32 * 7 * 7, 64), nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(64, num_classes)
        )
    def forward(self, x):
        return self.net(x)
```

층 cnn2층에 mlp붙여서 학습

학습 결과 추론

filename,pred,prob
1_001.png,1,0.9927195310592651
1_002.png,2,0.786540150642395
1_003.png,1,0.9233830571174622
1_004.png,0,0.6221650242805481
1_005.png,1,0.9962779879570007
1_006.png,0,0.5067967772483826
1_007.png,0,0.6575164198875427
1_008.png,0,0.6252231001853943
1_009.png,1,0.5148527026176453
1_010.png,2,0.7509703636169434

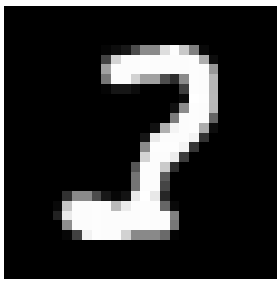
1: 4/10장

2_001.png,1,0.9999997615814209
2_002.png,1,0.9999995231628418
2_003.png,1,0.9999916553497314
2_004.png,1,0.9999446868896484
2_005.png,1,0.9345482587814331
2_006.png,1,0.9999983310699463
2_007.png,1,1.0
2_008.png,1,1.0
2_009.png,1,1.0
2_010.png,1,1.0

2: 10/10장

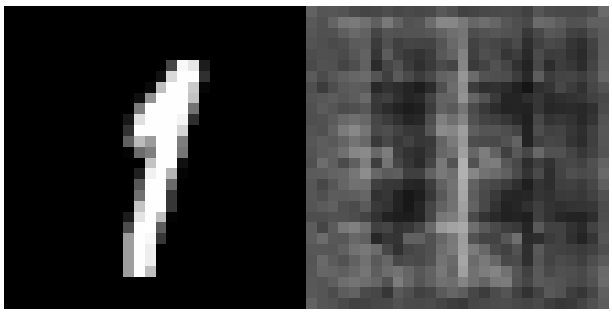
3_001.png,2,1.0
3_002.png,2,0.9999998807907104
3_003.png,2,1.0
3_004.png,2,1.0
3_005.png,2,1.0
3_006.png,2,1.0
3_007.png,1,0.9487519860267639
3_008.png,2,1.0
3_009.png,2,0.9999997615814209
3_010.png,2,0.6353267431259155

3: 9/10장



(틀린 3 그림)

★GAN을 이용한 가짜 데이터 생성



원본 가짜

(데이터 불균형 해소)

```python
class SmallCNN(nn.Module):
    def __init__(self, num_classes=2):
        super().__init__()
        self.net = nn.Sequential(
            nn.Conv2d(1, 16, 3, padding=1), nn.ReLU(), nn.MaxPool2d(2),
            nn.Conv2d(16, 32, 3, padding=1), nn.ReLU(), nn.MaxPool2d(2),
            nn.Flatten(),
            nn.Linear(32 * 7 * 7, 64), nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(64, num_classes)
        )
    def forward(self, x):
        return self.net(x)
```

학습 네트워크는 동일하게

학습 결과 추론

filename,pred,prob
1_001.png,1,0.9995075464248657
1_002.png,0,0.8410381078720093
1_003.png,1,0.9154471755027771
1_004.png,0,0.9803251624107361
1_005.png,1,0.9885687828063965
1_006.png,0,0.9477829337120056
1_007.png,0,0.9965433478355408
1_008.png,0,0.9281999468803406
1_009.png,0,0.6032376289367676
1_010.png,0,0.9303132891654968

1: 7/10장

2_001.png,1,1,0
2_002.png,1,1,0
2_003.png,1,0,9999994039535522
2_004.png,1,0,9998823404312134
2_005.png,1,0,618777871131897
2_006.png,1,0,9999998807907104
2_007.png,1,1,0
2_008.png,1,1,0
2_009.png,1,1,0
2_010.png,1,1,0

2: 10/10장

3_001.png,2,1,0
3_002.png,2,0,9999997615814209
3_003.png,2,0,9999998807907104
3_004.png,2,1,0
3_005.png,2,1,0
3_006.png,2,1,0
3_007.png,1,0,9971338510513306
3_008.png,2,1,0
3_009.png,2,0,9999997615814209
3_010.png,2,0,9999914169311523

3: 9/10장

GAN적용 전 후 비교

filename,pred,prob
1_001.png,1,0.9927195310592651
1_002.png,2,0.786540150642395
1_003.png,1,0.9233830571174622
1_004.png,0,0.6221650242805481
1_005.png,1,0.9962779879570007
1_006.png,0,0.5067967772483826
1_007.png,0,0.6575164198875427
1_008.png,0,0.6252231001853943
1_009.png,1,0.5148527026176453
1_010.png,2,0.7509703636169434

1: 4/10장

2_001.png,1,0,9999997615814209
2_002.png,1,0,9999995231628418
2_003.png,1,0,9999916553497314
2_004.png,1,0,9999446868896484
2_005.png,1,0,9345482587814331
2_006.png,1,0,9999983310699463
2_007.png,1,1,0
2_008.png,1,1,0
2_009.png,1,1,0
2_010.png,1,1,0

2: 10/10장

3_001.png,2,1,0
3_002.png,2,0,9999998807907104
3_003.png,2,1,0
3_004.png,2,1,0
3_005.png,2,1,0
3_006.png,2,1,0
3_007.png,1,0,9487519860267639
3_008.png,2,1,0
3_009.png,2,0,9999997615814209
3_010.png,2,0,6353267431259155

3: 9/10장

filename,pred,prob
1_001.png,1,0.9995075464248657
1_002.png,0,0.8410381078720093
1_003.png,1,0.9154471755027771
1_004.png,0,0.9803251624107361
1_005.png,1,0.9885687828063965
1_006.png,0,0.9477829337120056
1_007.png,0,0.9965433478355408
1_008.png,0,0.9281999468803406
1_009.png,0,0.6032376289367676
1_010.png,0,0.9303132891654968

1: 7/10장

2_001.png,1,1,0
2_002.png,1,1,0
2_003.png,1,0,9999994039535522
2_004.png,1,0,9998823404312134
2_005.png,1,0,618777871131897
2_006.png,1,0,9999998807907104
2_007.png,1,1,0
2_008.png,1,1,0
2_009.png,1,1,0
2_010.png,1,1,0

2: 10/10장

3_001.png,2,1,0
3_002.png,2,0,9999997615814209
3_003.png,2,0,9999998807907104
3_004.png,2,1,0
3_005.png,2,1,0
3_006.png,2,1,0
3_007.png,1,0,9971338510513306
3_008.png,2,1,0
3_009.png,2,0,9999997615814209
3_010.png,2,0,9999914169311523

3: 9/10장

결과

23/30장: 76%

26/30장: 86%

★★10%상승 확인