

$J^{(G)}$: Generator의 비용함수

$J^{(D)}$: Discriminator의 비용함수

$\Theta^{(G)}$: Generator의 훈련 parameter

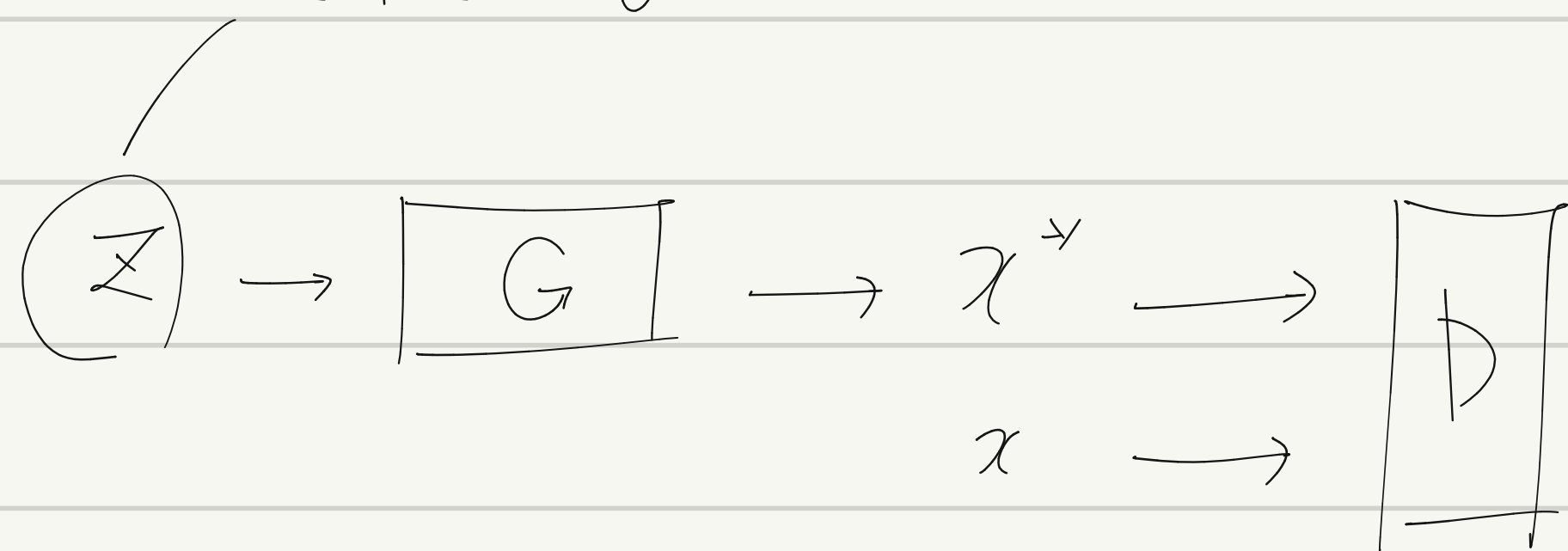
$\Theta^{(D)}$: Discriminator의 훈련 parameter



생성자의 비용함수 = $J^{(G)}(\Theta^{(G)}, \Theta^{(D)})$

판별자의 비용함수 = $J^{(D)}(\Theta^{(G)}, \Theta^{(D)})$

랜덤한 잡음 벡터



판별자의 목적 $\Rightarrow D(x) = 1$, $D(x^*) = 0$

생성자의 목적 $\Rightarrow D(x^*) = 1$

CODE

train_on_batch

```
train_on_batch(x, y, sample_weight=None, class_weight=None)
```

하나의 데이터 배치에 대해서 경사 업데이트를 1회 실시합니다.

인수

- x**: 학습 데이터의 Numpy 배열, 혹은 모델이 다중 인풋을 갖는 경우 Numpy 배열의 리스트. 모델의 모든 인풋에 이름이 명명된 경우 인풋 이름을 Numpy 배열에 매핑하는 딕셔너리를 전달할 수도 있습니다.
- y**: 표적 데이터의 Numpy 배열, 혹은 모델이 다중 아웃풋을 갖는 경우 Numpy 배열의 리스트. 모델의 모든 아웃풋에 이름이 명명된 경우 아웃풋 이름을 Numpy 배열에 매핑하는 딕셔너리를 전달할 수도 있습니다.
- sample_weight**: x와 동일한 길이의 선택적 배열로, 각 샘플에 대한 모델의 손실에 적용할 가중치를 담습니다. 시간적 데이터의 경우, (samples, sequence_length)의 형태를 가진 2D 배열을 전달하여 모든 샘플의 모든 시간 단계에 대해 각기 다른 가중치를 적용할 수 있습니다. 이러한 경우 반드시 `compile()` 내에서 `sample_weight_mode="temporal"` 을 설정해야 합니다.
- class_weight**: 가중치 값(부동소수점)에 클래스 색인(정수)을 매핑하는 선택적 딕셔너리로, 학습 과정 중 모델의 손실을 각 클래스의 샘플에 적용하는데 사용됩니다. 이는 모델이 상대적으로 적은 수의 샘플을 가진 클래스의 샘플에 "더 주의를 기울이도록" 하는데 유용합니다.

반환값

(모델이 단일 아웃풋만 갖고 측정항목을 보유하지 않는 경우) 스칼라 학습 손실, 혹은 (모델이 다중 아웃풋, 그리고/혹은 측정항목을 갖는 경우) 스칼라 리스트. `model.metrics_names` 속성은 스칼라 아웃풋에 대한 디스플레이 라벨을 제공합니다.