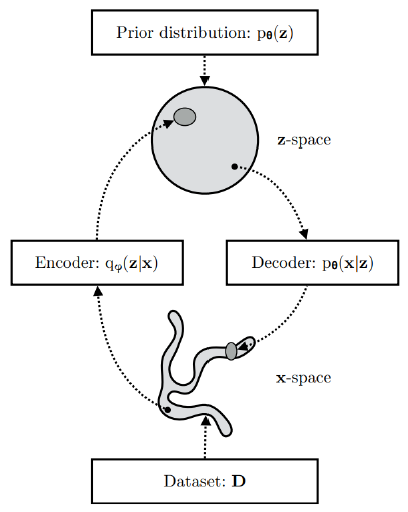
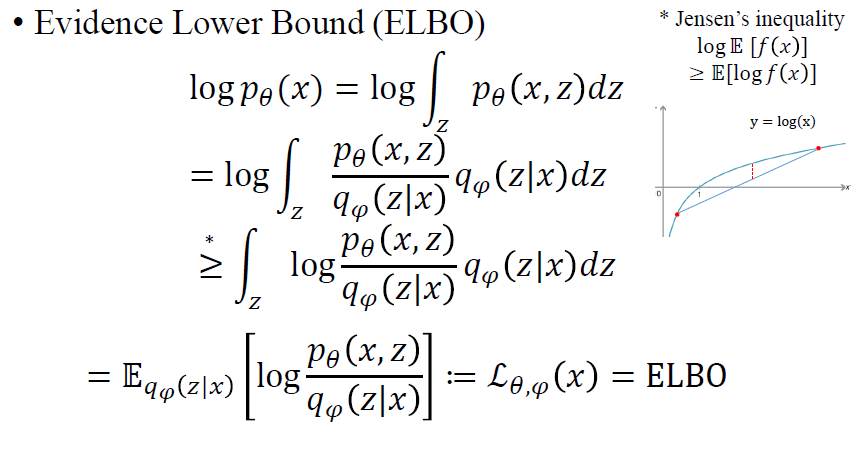


Auto-Encoders - Input이 x이고 output도 x이기를 바라는 것(x와 같은 차원으로 x’이 나오게)

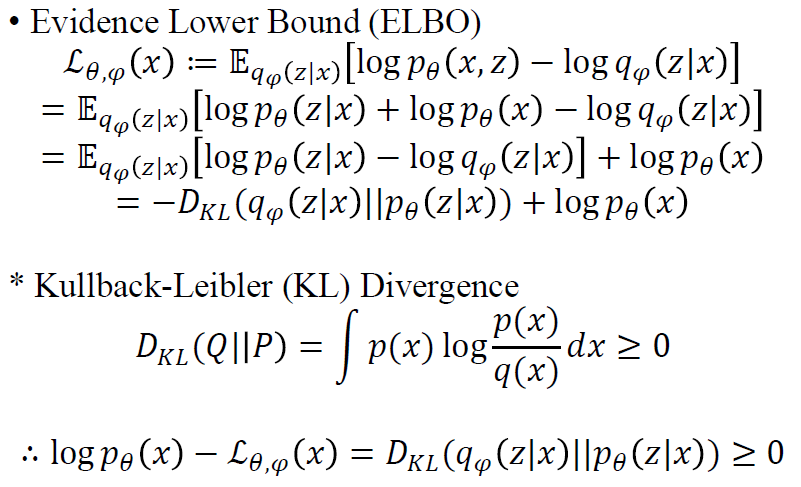
예를 들어 784차원의 벡터의 정보(x)를 32차원으로 줄여(encoder) 다시 784차원(x’)으로

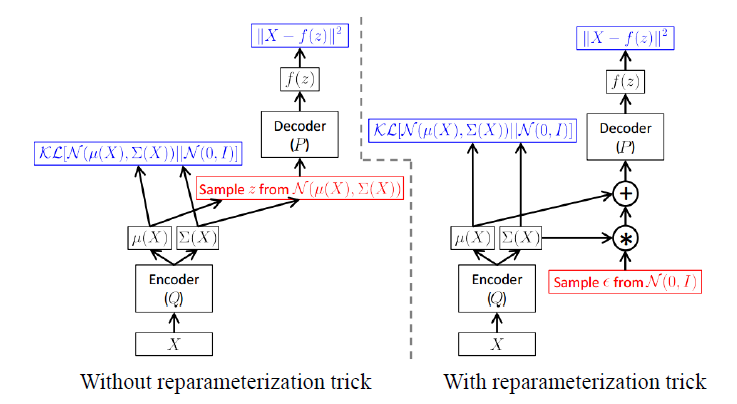
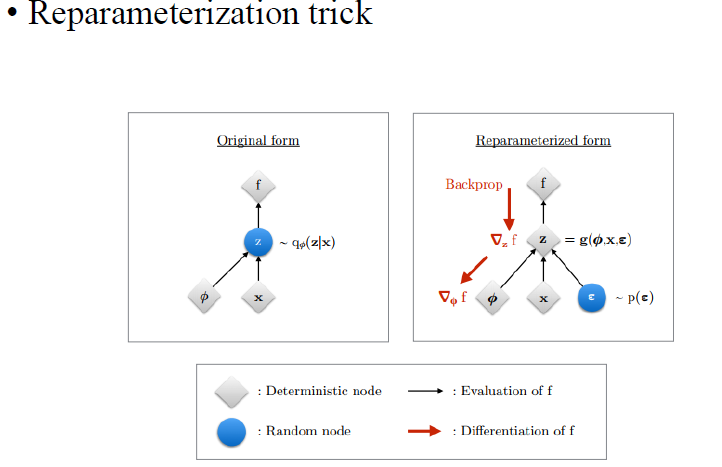
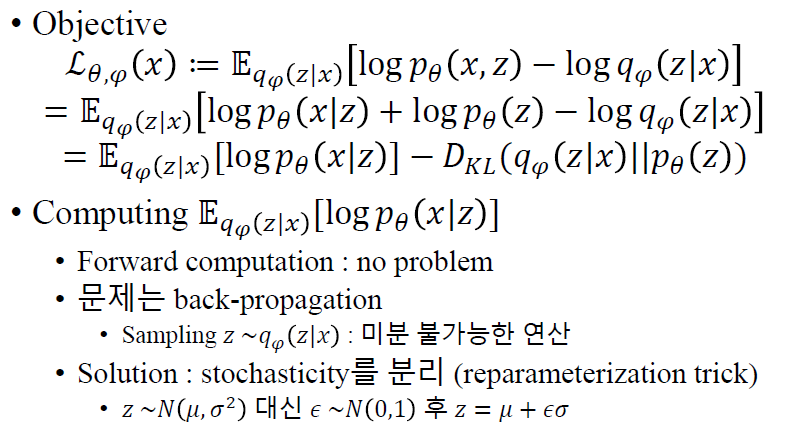
복원(decoder) 🡪 즉, 저장을 할 때 784차원을 다 저장할 필요 없이 32차원만 저장하면 된다.

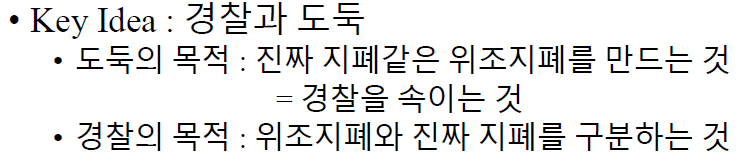


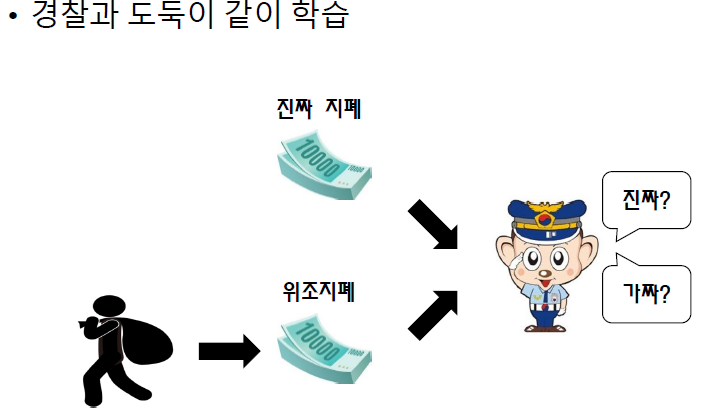


실제 세계의 데이터 x에 대해서 우리의 모델 확률 분포(pθ)가 높게 판단하도록 함

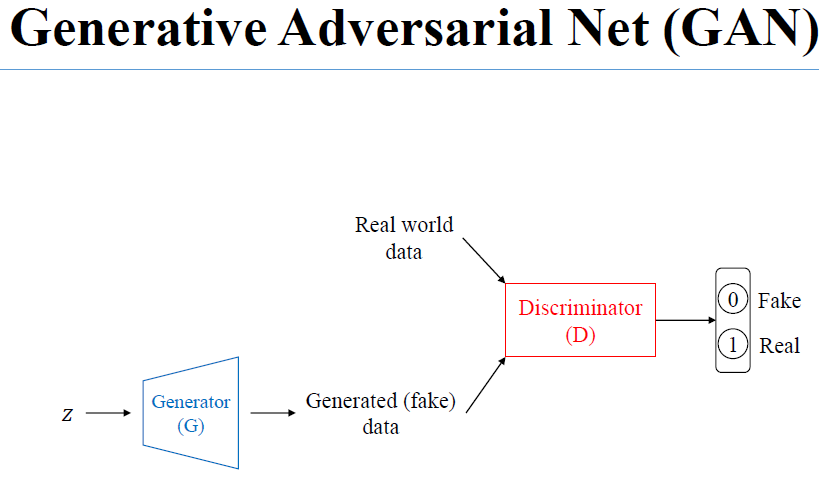








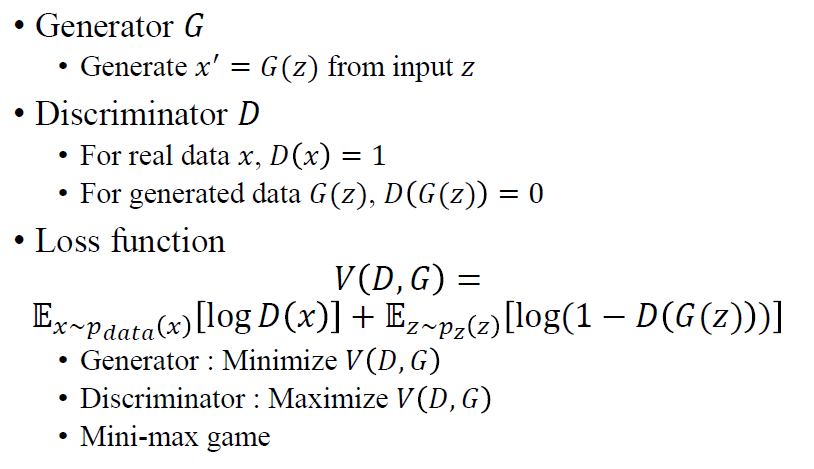
도둑이랑 경찰이랑 같이 발전한다.



위조지폐를 만듦(Generator : 진짜 데이터 같은 데이터를 만드는 모델)

진짜인지 가짜인지를 판별(Discriminator : 진짜 데이터인지 가짜 데이터인지 판단하는 모델)

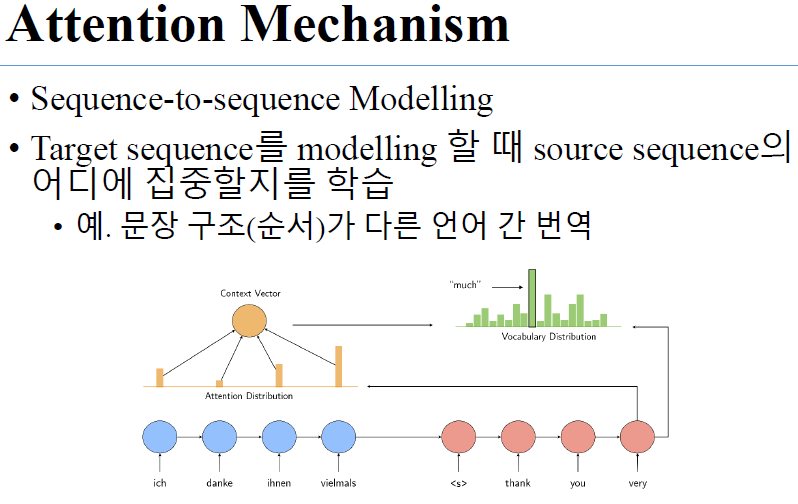
실제 데이터 : X, Fake data : X’



잘 훈련된 Discriminator이라면 가짜 데이터와 진짜 데이터를 잘 파악할 것

Discriminator(경찰)의 목적 : Loss function이 실제 확률 분포로부터 샘플링 된 x에 대해서 dx를 크게 하고 D(G(z)))를 작게 함

Generator(도둑)의 목적 : D(G(z)))의 값을 크게 만드는 것

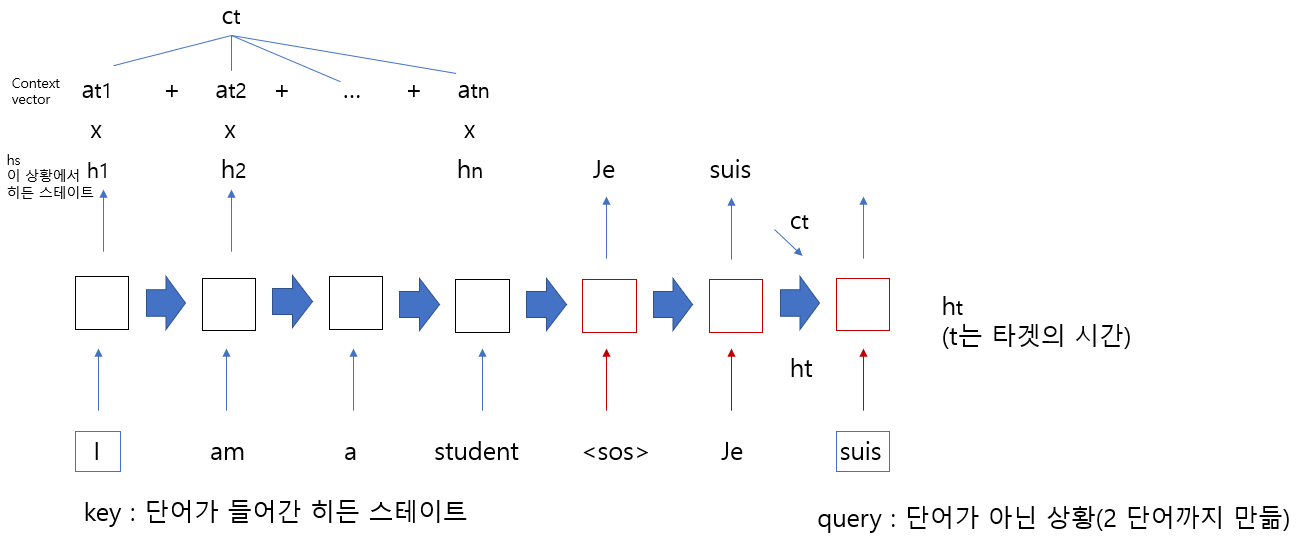


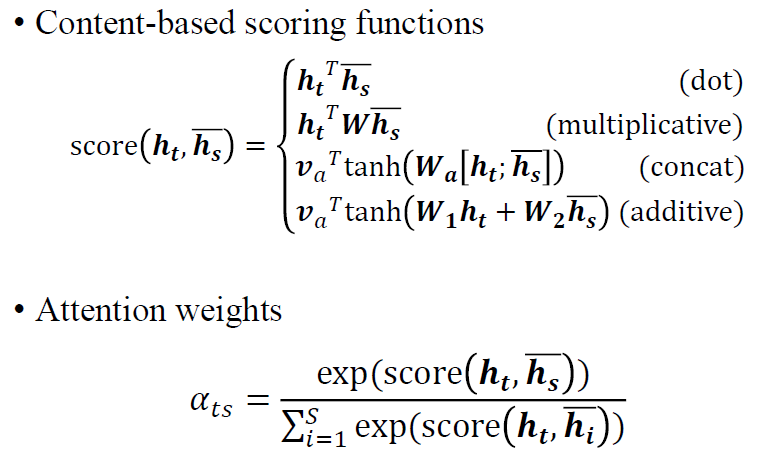
번역을 할 때, 영어에서 스페인어로 번역을 한다면 타겟인 스페인어를 만들어 내는 목적을

가졌을 때, 어디에 집중을 할지 학습하는 것

source sequence은 원래 문장, target은 만들고 싶은(번역하고 싶은) 문장

RNN과 차이점 - 어디에 집중을 할지와 Context vector과 히든 state와 concat해서 사용함





dot - 가장 얼마나 비슷한 지 스코어를 높게 나오도록 (1 x h)(h x 1) = 1 x 1

multiplicative에 있는 W도 학습해야 할 파라미터 (1 x h)(h x 1) =1 x 1

concat – 쭉 이어 쓰겠다 (1 x 2h)(2h x 1) = 1 x 1

additive – w1(h x 1) + w2(h x 1)

