December 10, 2025

▣ **강의명:** CSCI E-89B: 자연어 처리 입문

▣ **주차:** Lecture 11

▣ **교수명:** Dmitry Kurochkin

▣ **목적:** Lecture 11의 핵심 개념 학습

□ 핵심 요약

이 문서는 자연어 처리(NLP)의 고급 모델들을 다룹니다. 문장의 순서와 구조를 이해하는 **순차 레이블링 모델**(HMM, CRF, BiLSTM-CRF)부터, 새로운 데이터(텍스트나 이미지)를 생성하는 **생성 모델**(VAE, GAN)까지의 핵심 원리를 다룹니다.

각 모델이 **왜 등장했는지**(이전 모델의 한계), **어떻게 작동하는지**(핵심 아이디어), 그리고 **어떤 단점이 있는지**(다음 모델의 등장 배경)를 중심으로 설명합니다. 이 문서 하나만으로 각 모델의 개념을 잡고 서로 비교할 수 있도록 구성되었습니다.

Contents

1 용어 정리

본격적인 학습에 앞서, 이 문서에서 다룰 주요 용어들을 정리합니다.

Table 1: 주요 *NLP* 모델 및 개념 용어

용어	쉬운 설명	원어	비고 (핵심 키워드)
마코프 속성	"미래는 오직 현재에만 의존한다." (과거는 불필요)	Markov Property	HMM의 기본 가정
HMM	'숨겨진 상태'가 '관찰된 결과'를 만든다고 보는 모델	Hidden Markov Model	생성 모델, POS 태깅
CRF	'전체 관찰 결과'를 보고 '가장 확률 높은 상태'를 맞히는 모델	Conditional Random Fields	판별 모델, NER, HMM의 한계 극복
특징 함수	CRF가 특정 패턴(예: 대문자, 이전 단어)을 감지하는 규칙	Feature Function	CRF의 핵심. (수동 정의 필요)
BiLSTM	문장을 양방향으로 읽어 문맥을 파악하는 똑똑한 RNN	Bidirectional LSTM	자동 특징 추출기
오토인코더	데이터를 압축(인코더)했다가 복원(디코더)하는 모델	Autoencoder (AE)	차원 축소, 특징 학습
VAE	잠재 공간을 '확률 분포'로 만들어 부드럽게 연결한 AE	Variational AE	생성 모델, 잠재 공간 구조화
잠재 손실	VAE가 잠재 공간을 구조화하도록 강제하는 패널티	Latent Loss (KL Div.)	μ는 0으로, σ는 1로 유도
GAN	위조범(생성자)과 경찰(판별자)이 경쟁하며 학습하는 모델	Generative Adversarial Network	생성 모델, 고품질 이미지 생성

2 핵심 개념 1: Hidden Markov Models (HMM)

HMM(은닉 마코프 모델)은 순차적인 데이터(예: 문장)를 이해하기 위한 고전적이면서도 중요한 통계 모델입니다.

2.1 시작하기: 마코프 체인 (Markov Chains)

HMM을 이해하려면 먼저 마코프 체인을 알아야 합니다.

□ 핵심 정보

마코프 체인(Markov Chain)은 여러 '상태'(State)가 존재하고, 한 상태에서 다음 상태로 이동할 확률(전이 확률)이 정해져 있는 모델입니다.
핵심 가정: 마코프 속성 (Markov Property)

- "미래의 상태는 오직 **현재 상태**에만 의존한다."
- 즉, 내가 '상태 2'에 도달하기까지 '상태 0 → 상태 1'을 거쳤든, '상태 3 → 상태 1'을 거쳤든 상관없이, '상태 1'에 있다는 **현재 사실**만이 다음 상태(예: '상태 2')로 갈 확률에 영향을 줍니다.
- (비유) 주사위 굴리기: 이전에 1이 10번 연속 나왔다고 해서, 다음번에 1이 나올 확률(1/6)이 변하지 않는 것과 비슷합니다.

□ 예제:

예시: 날씨 마코프 체인 세 가지 날씨 상태가 있다고 가정합니다: (1) 맑음, (2) 흐림, (3) 비

- 오늘 '맑음'일 때, 내일 '맑음'일 확률 ($P(\text{맑음}|\text{맑음}) = 0.7$)
- 오늘 '맑음'일 때, 내일 '흐림'일 확률 ($P(\text{흐림}|\text{맑음}) = 0.2$)
- 오늘 '맑음'일 때, 내일 '비'일 확률 ($P(\text{비}|\text{맑음}) = 0.1$)
- (이 확률들의 합은 1이 되어야 합니다: $0.7 + 0.2 + 0.1 = 1.0$)

이처럼 '흐림'일 때와 '비'일 때의 다음 날 날씨 확률도 모두 정의해 놓은 것이 마코프 체인입니다.

2.2 HMM: 숨겨진 상태와 관찰된 결과

HMM은 마코프 체인에서 한 단계 더 나아갑니다. 우리가 '상태'를 직접 볼 수 없고, '상태'가 만들어낸 '결과물'만 볼 수 있다고 가정합니다.

□ 핵심 정보

HMM(Hidden Markov Model)은 두 가지 층위로 구성됩니다.

1. **숨겨진 상태 (Hidden States, Y)**:
   - 우리는 직접 볼 수 없습니다. (예: 실제 날씨, 단어의 품사)
   - 이 숨겨진 상태들은 마코프 속성을 따르며 서로 이동합니다. (예: 명사 다음에는 동사가 올 확률이 높다)
   - 이 이동 확률을 **전이 확률 (Transition Probability)**이라고 부릅니다.
2. **관찰된 결과 (Observations, X)**:
   - 숨겨진 상태가 만들어낸 결과물이며, 우리가 실제로 보는 데이터입니다.

- (예: 아이스크림 판매량, 실제 단어 'study')
- 특정 숨겨진 상태가 특정 관찰 결과를 만들어낼 확률을 **방출 확률 (Emission Probability)**이라고 부릅니다.

이 외에, 문장이 어떤 상태에서 시작할지 정하는 **초기 상태 확률**이 있습니다.

□ 예제:

예시: 품사 판별 (Part-of-Speech Tagging) 우리의 목표는 "I study"라는 문장(관찰 X)을 보고, "대명사, 동사"라는 품사(숨겨진 상태 Y)를 맞히는 것입니다.

- **관찰 (X)**: "I", "study" (우리가 보는 단어)
- **숨겨진 상태 (Y)**: "대명사(PRP)", "동사(VBP)" (우리가 맞혀야 하는 품사)
- **초기 확률**: 문장은 '대명사'로 시작할 확률이 높다. ($P(Y_1 = PRP)$)
- **전이 확률**: '대명사' 상태(Y_t) 다음에는 '동사' 상태(Y_t+1)가 올 확률이 높다. (P(VBP|PRP))
- **방출 확률**: '대명사' 상태(Y)는 "I"라는 단어(X)를 방출(생성)할 확률이 높다. (P("I"|PRP)) / '동사' 상태(Y)는 "study"라는 단어(X)를 방출할 확률이 높다. (P("study"|VBP))

HMM은 이 확률들을 조합하여 "I study"라는 관찰이 주어졌을 때, 가장 그럴듯한(확률 높은) 숨겨진 상태의 연속(즉, 품사 태그)이 "대명사 → 동사"임을 계산해냅니다.

2.3 HMM의 한계

HMM은 강력하지만 치명적인 한계를 가집니다.

주의사항

한계: 너무 단순한 의존성 가정

- **마코프 속성의 한계**: HMM은 $t+1$ 시점의 상태(Y_t+1)가 **오직 t 시점의 상태(Y_t)에만** 의존한다고 가정합니다.
- (예시) "I study"에서 "study"의 품사는 "I"(대명사)에만 영향을 받습니다.
- **관찰 독립성의 한계**: HMM은 t 시점의 관찰(X_t)이 **오직 t 시점의 상태(Y_t)에만** 의존한다고 가정합니다.
- (예시) "study"라는 단어는 "동사"라는 현재 품사에만 영향을 받습니다.

문제점: 실제 언어는 그렇지 않습니다! "study"가 동사인지 명사인지 판단하려면 "I"라는 단어뿐만 아니라 "I **will** study..."나 "A **recent** study..."처럼 문장 **전체**의 다른 단어들(X)을 모두 참고해야 합니다.

HMM은 현재 상태(Y_t) 외에는 **그 어떤 정보(다른 시점의 Y나 X)도 보지 못하는** 근시안적인 모델입니다.

3 핵심 개념 2: Conditional Random Fields (CRFs)

CRF(조건부 랜덤 필드)는 HMM의 '근시안적인' 한계를 극복하기 위해 등장한 강력한 순차 레이블링 모델입니다.

3.1 HMM의 한계를 극복하다

HMM이 "현재 상태(Y_t)는 오직 이전 상태(Y_t-1)에만 의존한다"고 가정한 반면, CRF는 이 가정을 완전히 버립니다.

□ 핵심 정보

CRF(Conditional Random Field)의 핵심 아이디어:
"레이블(Y)을 맞힐 때, HMM처럼 쪼개서 보지 말고, **관찰(X) 시퀀스 전체를 한꺼번에** 조건으로 사용하자!"

- **HMM (생성 모델)**: $P(X, Y)$를 모델링. (상태가 관찰을 '생성'한다고 봄)
  - "어떤 품사(Y)가 어떤 단어(X)를 생성할까?" (날씨가 아이스크림 판매량을 생성)
  - $P(X, Y) = P(Y) \times P(X|Y)$
- **CRF (판별 모델)**: $P(Y|X)$를 직접 모델링. (관찰을 보고 상태를 '판별'함)
  - "이 단어들(X)이 주어졌을 때, 가장 적절한 품사(Y)는 무엇일까?"
  - (비유) 로지스틱 회귀가 선형 회귀보다 분류에 더 직접적이듯, CRF는 HMM보다 순차 레이블링에 더 직접적입니다. (CRF는 로지스틱 회귀의 시퀀스 버전이라 불림)

이 '판별' 접근 방식 덕분에, CRF는 HMM이 할 수 없었던 **문장 전체의 다양한 특징**을 자유롭게 활용할 수 있습니다.

3.2 CRF의 핵심: 특징 함수 (Feature Functions)

CRF가 문장 전체의 특징을 활용하는 방법은 '특징 함수'를 사용하는 것입니다.

□ 핵심 정보

특징 함수 (f_k)는 우리가 모델에게 알려주는 "규칙" 또는 "패턴"입니다. 이 함수는 (이전 레이블 Y_{t-1}, 현재 레이블 Y_t, 관찰 시퀀스 X, 현재 시점 t)를 입력받아 0 또는 1을 반환합니다.
가중치 (λ_k)는 각 특징 함수(f_k)가 얼마나 중요한지 나타내는 점수입니다. (이 값은 모델이 학습을 통해 스스로 찾습니다.)
CRF는 이 (특징 함수 × 가중치)의 합을 기반으로 가장 점수가 높은 레이블 시퀀스를 선택합니다.

□ 예제:

예시: 이름 인식 (Named Entity Recognition, NER) 문장 "Mr. Smith..."에서 "Smith"가 사람 이름(B-PERSON)임을 맞히고 싶습니다.

1. 우리가 직접 '특징 함수'(f_k)를 설계합니다: (수동 공학)

- $f_1 = 1$ **if** (현재 단어(X_t)가 "Smith" AND 현재 레이블(Y_t)이 "B-PERSON") **else** 0
- $f_2 = 1$ **if** (현재 단어(X_t)가 대문자로 시작 AND 현재 레이블(Y_t)이 "B-PERSON") **else** 0
- $f_3 = 1$ **if** (**이전 단어**(X_{t-1})**가 "Mr."** AND 현재 레이블(Y_t)이 "B-PERSON") **else** 0

- $f_4 = 1$ **if** (이전 레이블(Y_{t-1})이 "O" AND 현재 레이블(Y_t)이 "B-PERSON") **else** 0
- $f_5 = 1$ **if** (**다음 단어**(X_{t+1})**가 쉼표**(,) AND 현재 레이블(Y_t)이 "B-PERSON") **else** 0

2. 모델이 λ_k (가중치)를 학습합니다:

- 학습 데이터에서 f_2, f_3, f_4 같은 패턴이 이름 인식에 매우 유용하다는 것을 발견하면,
- 모델은 $\lambda_2, \lambda_3, \lambda_4$에 높은 양수 값을 할당합니다.

3. 예측: "Mr. Smith"가 입력되면, f_2, f_3, f_4가 모두 활성화(1)되면서, "Smith"의 레이블로 "B-PERSON"을 선택할 때 전체 점수가 가장 높아지게 됩니다. HMM과 달리 **과거("Mr.")와 미래(",")** - **만약 있다면)**의 관찰(X)을 모두 활용할 수 있습니다.

3.3 CRF의 장점과 단점

주의사항

장점:

- **유연한 특징 활용**: HMM과 달리 문장 전체의 어떤 특징(예: 현재 단어, 이전/다음 단어, 접두사, 접미사, 대소문자 여부 등)이든 자유롭게 가져와 사용할 수 있습니다.
- **판별 모델**: $P(Y|X)$를 직접 모델링하여 순차 레이블링 작업에 더 적합합니다.

치명적인 단점:

- **수동 특징 공학 (Intensive Feature Engineering)**:
- 위 예시처럼, 어떤 특징(f_k)이 유용한지는 **사람이 직접** 생각해서 코드로 구현해야 합니다.
- 이는 엄청난 시간과 노력이 필요하며, 도메인 전문가의 지식이 요구됩니다.
- 만약 우리가 f_3 (이전 단어가 "Mr.") 같은 중요한 특징을 빠뜨리면, 모델의 성능은 급격히 저하됩니다.

4 핵심 개념 3: BiLSTM-CRF (자동 특징 공학)

CRF는 강력했지만 '수동 특징 공학'이라는 거대한 벽에 부딪혔습니다. 이 문제를 해결하기 위해 딥러닝, 즉 BiLSTM이 CRF와 결합되었습니다.

4.1 CRF의 단점을 해결하다: BiLSTM의 등장

CRF의 문제는 '좋은 특징'을 사람이 만들어야 한다는 것이었습니다. 만약 '좋은 특징'을 기계가 알아서 문맥을 보고 뽑아주면 어떨까요?

□ 핵심 정보

BiLSTM-CRF 아키텍처는 두 개의 강력한 모델이 각자의 장점을 살려 결합한 형태입니다.

1. **BiLSTM (Bidirectional LSTM) 층: "자동 특징 추출기"**
   - 입력: 단어들의 시퀀스 (보통 임베딩 벡터로 변환됨)
   - 역할: CRF에 필요했던 '특징 함수'(f_k)를 **자동으로 학습**합니다.
   - BiLSTM은 문장을 정방향(왼쪽 → 오른쪽)과 역방향(오른쪽 → 왼쪽)으로 동시에 읽습니다.
   - (예시) "Smith"라는 단어를 처리할 때, 정방향 LSTM은 "Mr."라는 과거 문맥을, 역방향 LSTM은 "lives in..."이라는 미래 문맥을 모두 고려합니다.
   - 출력: 각 단어 위치(t)에서, 과거와 미래 문맥이 모두 풍부하게 반영된 **"특징 벡터"** ($\mathbf{H}_t$) . CRF ′ ′ .
2. **CRF 층: "최종 레이블 결정자"**
   - 입력: BiLSTM이 뽑아준 고품질 특징 벡터들 (H_1, H_2, ...)
   - 역할: 이 특징들을 입력받아, **레이블 시퀀스 간의 의존성**을 학습하고 최종 레이블을 결정합니다.
   - (예시) "Smith"가 B-PERSON(이름 시작)일 확률이 높다는 특징 벡터를 받아도, CRF는 "I-PER(이름 중간) 뒤에는 B-LOC(장소 시작)이 올 수 없다"와 같은 **레이블 간의 규칙**을 학습하여, "B-PERSON → I-PERSON"처럼 문법적으로 올바른 레이블 시퀀스를 출력하도록 보장합니다.

왜 BiLSTM 위에 Softmax만 쓰지 않고 굳이 CRF를 붙이나요? BiLSTM의 각 시점 출력(H_t)에 Softmax를 적용하여 바로 레이블을 예측할 수도 있습니다 (이것을 '독립적인 분류'라고 합니다).

문제점: Softmax는 각 단어의 레이블을 **독립적으로** 예측합니다.

- (예시) "Mr. John Smith"를 예측할 때, "Mr."(B-PER), "John"(I-PER)까지는 잘 맞히다가, "Smith"에서 실수로 B-LOC(장소 시작)를 예측할 수 있습니다.
- 그 결과 "B-PER → I-PER → **B-LOC**"라는, 문법적으로 말이 안 되는(I-PER 다음에는 I-PER나 O가 와야 함) 레이블 시퀀스가 나올 수 있습니다.

CRF의 역할: CRF 층은 $P(Y|X)$를 '전역적(Globally)'으로 최적화합니다.

- 즉, BiLSTM이 "Smith"를 B-LOC로 예측하려는 성향(Emission 점수)을 보여도, CRF 층이 학습한 "I-PER → B-LOC"라는 **전이(Transition) 점수**가 매우 낮다면,
- CRF는 이 경로를 패널티를 주어 선택하지 않고, 대신 "I-PER → I-PER"라는 더 그럴듯한(점수가 높은) 전체 시퀀스를 선택합니다.

결론: BiLSTM이 '문맥을 읽어 특징을 뽑는' 두뇌라면, CRF는 '레이블 간의 문법 규칙을 적용하는' 문법 교정기 역할을 합니다.

4.2 장점과 단점

□ 핵심 정보

장점:

- **자동 특징 공학**: BiLSTM이 문맥을 읽어 CRF가 필요로 하는 특징을 자동으로 학습합니다. (수동 공학 불필요)
- **높은 정확도**: 문맥(BiLSTM)과 레이블 의존성(CRF)을 모두 고려하여, NER, POS 태깅 등에서 SOTA(최고 수준) 성능을 보였습니다.

단점:

- **높은 계산 비용**: BiLSTM과 CRF를 모두 학습해야 하므로 HMM이나 CRF 단독 모델보다 복잡하고 느립니다.
- **복잡한 모델 튜닝**: 하이퍼파라미터가 많아 튜닝이 어렵습니다.

5 실습/코드: BiLSTM-CRF 구현 (Python)

BiLSTM-CRF 모델은 `torch`와 `pytorch-crf` 같은 라이브러리를 사용하여 구현할 수 있습니다. (코드는 개념 이해를 위한 의사 코드(pseudo-code)에 가깝게 단순화되었습니다.)

```python
import torch
import torch.nn as nn
from TorchCRF import CRF

class BiLSTM_CRF(nn.Module):
    def __init__(self, vocab_size, tagset_size, embedding_dim, hidden_dim):
        super(BiLSTM_CRF, self).__init__()

        # 1. 임베딩층단어 ( -> 벡터)
        self.embedding = nn.Embedding(vocab_size, embedding_dim)

        # 2. BiLSTM 층특징 ( 추출기)
        #    hidden_dim // 2 는양방향이므로합치면이   hidden_dim 됨
        self.lstm = nn.LSTM(embedding_dim, hidden_dim // 2,
                            num_layers=1, bidirectional=True,
                            batch_first=True)

        # 3. Linear 층 (LSTM 출력을가 CRF 받을점수로변환 )
        # 이것이의    CRF 'Emission 점수가' 됨
        self.hidden2tag = nn.Linear(hidden_dim, tagset_size)

        # 4. CRF 층레이블 ( 결정자)
        self.crf = CRF(tagset_size, batch_first=True)

    def forward(self, sentences, tags=None, mask=None):
        # 1. 임베딩
        embeddings = self.embedding(sentences)

        # 2. BiLSTM 통과
        lstm_out, _ = self.lstm(embeddings)

        # 3. Emission 점수계산
        emissions = self.hidden2tag(lstm_out)

        if tags is not None:
            # 학습모드 : (emissions, tags) 간의로그가능도 (likelihood)를 계산
            # 이것이( 가Loss 됨. 우리는이값을최소화    = 음수로그가능도최소화   )
            log_likelihood = self.crf(emissions, tags, mask=mask)
            return -log_likelihood
        else:
            # 예측모드 : emissions 점수와의 CRF 전이점수를고려하여
            # 가장점수가높은최적의태그시퀀스를디코딩      (Viterbi)
            tag_seq = self.crf.decode(emissions, mask=mask)
            return tag_seq
```

Listing 1: BiLSTM-CRF 모델 클래스 정의 (PyTorch 예시)

코드 해설

- `nn.Embedding`: 입력된 단어 인덱스(예: 1, 3, 10)를 밀집 벡터(예: 100차원)로 변환합니다.
- `nn.LSTM`: 임베딩된 벡터 시퀀스를 입력받아, 양방향 문맥을 고려한 특징 벡터 시퀀스(`lstm_out`)를 출력합니다.
- `nn.Linear`: `lstm_out`(예: 256차원)을 `tagset_size`(예: 9개 태그) 차원의 '점수(Emission Score)'로 변환합니다. 이 점수는 "이 단어가 이 태그일 확률이 얼마나 높은가"에 대한 BiLSTM의 의견입니다.
- `self.crf`: 이 'Emission 점수'와 자신이 학습한 '전이 점수(Tramsition Score)'를 함께 고려합니다.
- **학습 시 (`forward`의 `if`문)**: 정답 `tags`를 알려주고, 해당 정답 경로의 확률(`log_likelihood`)을 높이도록 모델(BiLSTM의 가중치, CRF의 전이 행렬)을 업데이트합니다.
- **예측 시 (`forward`의 `else`문)**: 정답이 없으므로, `crf.decode`(보통 비터비 알고리즘 사용)를 통해 현재 Emission 점수와 전이 점수를 조합할 때 총점이 가장 높은 '최적의 경로'를 찾아 반환합니다.

6 핵심 개념 4: Variational Autoencoders (VAEs)

지금까지는 주어진 입력(X)에서 레이블(Y)을 맞히는 '판별 모델' 또는 '순차 모델'을 다뤘습니다. 이제부터는 모델이 스스로 무언가를 '생성'해내는 **생성 모델(Generative Models)**을 다룹니다.

6.1 시작하기: 일반 오토인코더 (Autoencoder, AE)

VAE를 이해하려면 먼저 일반 AE를 알아야 합니다.

□ 핵심 정보

오토인코더(AE)는 "입력을 압축했다가 다시 복원하는" 신경망입니다.

- **인코더 (Encoder)**: 입력 데이터(예: 고해상도 이미지)를 저차원의 벡터(예: 30차원 벡터)로 **압축**합니다. 이 압축된 벡터를 **잠재 변수(Latent Variable)** 또는 **코딩(Coding)**이라고 부릅니다.
- **병목 (Bottleneck)**: 이 잠재 변수가 있는 가장 좁은 구간입니다.
- **디코더 (Decoder)**: 압축된 잠재 변수를 입력받아 다시 원본 데이터(이미지)로 **복원**합니다.

학습 목표: (입력)과 (복원된 출력)이 최대한 같아지도록 (즉, 재구성 손실(Reconstruction Loss)을 최소화하도록) 학습합니다. 이 과정에서 인코더는 데이터의 핵심 특징(예: 얼굴 이미지의 눈, 코, 입 특징)만 잠재 변수에 압축하는 법을 배우게 됩니다.

6.2 일반 AE의 한계: 비구조화된 잠재 공간

AE는 데이터 압축에는 유용하지만, '생성 모델'로 쓰기에는 치명적인 한계가 있습니다.

주의사항

문제: 잠재 공간에 구멍(Hole)이 많다.

- AE는 학습 데이터(예: 입력 이미지 1000장)를 잠재 공간(예: 2차원 평면)의 특정 점(1000개의 점)으로 매핑합니다.
- 디코더는 **정확히 그 1000개의 점** 위치에서만 원본을 잘 복원하도록 학습됩니다.
- 만약 우리가 1번 이미지의 잠재 변수(A점)와 2번 이미지의 잠재 변수(B점)의 **중간 지점(C점)**에 있는 값을 디코더에 넣으면 어떻게 될까요?
- **결과:** C점은 학습된 적이 없는 '구멍(Hole)' 영역이므로, 디코더는 완전히 깨지거나 의미 없는 이미지(노이즈)를 생성합니다.

이처럼 잠재 공간이 '점'들로만 이루어져 있고 그 사이가 비어있는 것을 **"비구조화된(Unstructured) 잠재 공간"**이라고 부릅니다.

6.3 VAE의 혁신: 잠재 공간을 확률 분포로

VAE는 "잠재 공간을 점이 아닌, 확률 분포(영역)로 만들어서 구멍을 메우자!"라는 아이디어에서 시작합니다.

□ 핵심 정보

VAE(Variational Autoencoder)의 작동 방식:

1. **인코더**: 입력을 받아 **하나의 점(벡터)**을 출력하는 대신, 이 점 주변의 **확률 분포**를 나타내는 두 개의 벡터를 출력합니다.
   - **평균 (μ, mu)**: 분포의 중심점
   - **로그 분산 (log σ^2, sigma)**: 분포가 퍼진 정도 (분산)
2. **샘플링 (Sampling)**: 이 평균과 분산을 따르는 정규 분포(가우시안 노이즈)에서 **랜덤하게 잠재 변수(z)를 샘플링**합니다. (이것이 '랜덤성 주입'입니다.)
3. **디코더**: 이 랜덤하게 샘플링된 z를 입력받아 원본을 복원합니다.

결과: 인코더는 입력을 받을 때마다 매번 조금씩 다른(랜덤한) z를 디코더에게 줍니다. 디코더는 이 '살짝 흔들린' z값들로도 원본을 잘 복원해야 하므로, 특정 '점'이 아닌 그 '주변 영역' 전체에서 복원하는 법을 배우게 됩니다.

6.4 핵심: 잠재 손실 (Latent Loss / KL Divergence)

여기서 매우 중요한 질문이 생깁니다.

만약 VAE가 재구성에만 집중하면 어떻게 될까요? 모델(인코더)은 '랜덤성'이 재구성을 방해한다는 것을 알고 있습니다.

모델의 꼼수: 인코더가 분산(σ)을 0으로 만들어 버립니다.

- 분산이 0이 되면 확률 분포는 '점'이 되고, 랜덤 샘플링은 항상 평균(μ) 값만 뽑게 됩니다.
- 랜덤성이 사라지고, VAE는 일반 AE와 똑같이 행동하게 됩니다.
- 잠재 공간은 다시 '비구조화된' 상태가 됩니다.

이 꼼수를 막기 위해 VAE는 두 번째 손실 함수, 즉 **잠재 손실**을 도입합니다.

□ 핵심 정보

잠재 손실 (Latent Loss): (정확히는 쿨백-라이블러 발산, D_{KL})
이 손실은 인코더가 만든 모든 확률 분포(μ, σ)가 "특정 기준 분포" (보통 $\mu = 0, \sigma = 1$인 표준 정규 분포)와 비슷해지도록 강제하는 패널티입니다.
잠재 손실의 두 가지 역할:

1. **분산(σ)이 0이 되는 것을 방지**: σ를 0이 아닌 1에 가깝게 유지하도록 강제하여, 인코더가 **적절한 랜덤성(노이즈)**을 유지하게 만듭니다. (잠재 공간에 '영역'을 만들도록 강제)
2. **평균(μ)을 0 근처로 집결**: 모든 분포의 중심(μ)을 원점(0) 근처로 모읍니다.

최종 효과 (가장 중요): 모든 분포가 원점 근처로 모이고(by $\mu \to 0$), 적절한 분산(by $\sigma \to 1$)을 가지게 되면, 서로 다른 이미지(예: A이미지, B이미지)의 잠재 공간 분포가 서로 **오버랩 (Overlap)**하게 됩니다.
이 '오버랩' 덕분에 잠재 공간에는 더 이상 '구멍'이 존재하지 않습니다. (이를 **"구조화된 (Structured) 잠재 공간"**이라 부름)
이제 A이미지의 분포와 B이미지의 분포 사이의 임의의 점 C를 뽑아 디코더에 넣어도, 그럴듯한 (A와 B를 섞은 듯한) 새로운 이미지가 생성됩니다.

7 핵심 개념 5: Generative Adversarial Networks (GANs)

GAN(생성적 적대 신경망)은 VAE와는 완전히 다른 철학을 가진 생성 모델입니다. VAE가 확률과 분포를 이용했다면, GAN은 두 신경망의 '경쟁'을 이용합니다.

7.1 핵심 비유: 위조지폐범 vs 경찰

GAN의 핵심 아이디어는 두 개의 신경망이 서로를 속이고 잡아내기 위해 경쟁(Adversarial)하는 것입니다.

□ 핵심 정보

GAN의 두 가지 구성 요소:

1. **생성자 (Generator, G): "위조지폐범"**
   - **입력:** 무작위 노이즈 (Noise, z) (예: 100차원의 랜덤 벡터)
   - **역할:** 이 노이즈를 입력받아 '가짜' 데이터(예: 가짜 이미지)를 생성합니다.
   - **목표:** 판별자가 "진짜"라고 속을 만큼 정교한 가짜 데이터를 만드는 것.
2. **판별자 (Discriminator, D): "경찰"**
   - **입력:** '진짜' 데이터 (학습 데이터셋) 또는 '가짜' 데이터 (생성자가 만든 것)
   - **역할:** 입력된 데이터가 진짜인지 가짜인지 판별합니다. (이진 분류: 0=Fake, 1=Real)
   - **목표:** 생성자가 만든 가짜를 '가짜(Fake)'라고 정확히 잡아내고, 진짜는 '진짜(Real)'라고 정확히 맞히는 것.

7.2 학습 과정 (Minimax Game)

두 네트워크는 서로의 이익이 반대되는 '제로섬 게임(Minimax Game)'을 벌이며 함께 똑똑해집니다.

GAN 학습 단계

학습은 두 단계를 번갈아 가며 진행됩니다.

1단계: 판별자(D) 학습 (생성자(G)는 고정)

- 생성자(G)가 노이즈로부터 '가짜 이미지'를 생성합니다.
- 판별자(D)에게 (1) '진짜 이미지'와 (2) '가짜 이미지'를 보여줍니다.
- 판별자(D)는 (1)에는 1(Real)이라고 답하고, (2)에는 0(Fake)이라고 답하도록 학습(업데이트)됩니다.
- (비유) 경찰이 진짜 지폐와 위조지폐를 보며 감별법을 익힙니다.

2단계: 생성자(G) 학습 (판별자(D)는 고정)

- 생성자(G)가 노이즈로부터 '가짜 이미지'를 생성합니다.
- 이 '가짜 이미지'를 **고정된** 판별자(D)에게 보여줍니다.
- 생성자(G)는 판별자(D)가 이 이미지를 보고 0(Fake)이 아닌 **1(Real)이라고 답하도록** 자신의 가중치를 학습(업데이트)합니다.
- (비유) 위조지폐범이 경찰(판별자)을 속일 수 있는(즉, 경찰이 '진짜'라고 착각할 만한) 더 정교한 위조지폐를 만드는 법을 배웁니다.

이 과정을 수없이 반복하면, 생성자(G)는 실제 데이터와 구별이 불가능할 정도로 고품질의 가짜 데이터를 생성하게 되고, 판별자(D)는 더 이상 진짜와 가짜를 구별하지 못하게 됩니다 (판별 확률 0.5에 수렴).

7.3 GAN의 진화와 응용

초기 GAN은 학습이 불안정했지만, 이후 DCGAN (합성곱 신경망 적용), StyleGAN 등 고도화된 모델이 등장하며 놀라운 성능을 보였습니다.

□ 예제:

예시: StyleGAN "This Person Does Not Exist" / "This Cat Does Not Exist"와 같은 웹사이트가 바로 StyleGAN을 이용한 예시입니다.

- 이 웹사이트들이 보여주는 사람 얼굴이나 고양이 이미지는 **세상에 존재하지 않는** 이미지입니다.
- GAN(생성자)이 수많은 실제 사진을 학습한 뒤, 데이터의 '특징'(예: 머리 스타일, 눈 색깔, 배경)을 이해하고 이를 조합하여 완전히 새로운(하지만 현실적인) 이미지를 생성해낸 것입니다.

장점: VAE보다 훨씬 더 선명하고 고품질의 이미지를 생성하는 경향이 있습니다. **단점:** 학습이 매우 불안정하고(예: 생성자가 한 가지 이미지만 계속 생성하는 'Mode Collapse'), 많은 데이터와 계산 자원이 필요합니다.

8 체크리스트: 학습 점검표

이 문서를 읽고 다음 질문에 스스로 답할 수 있는지 확인해 보세요.

최종 점검 리스트

- ☐ 마코프 속성이 무엇이며, 왜 HMM의 한계와 연결되는지 설명할 수 있는가?
- ☐ HMM과 CRF의 결정적인 차이는 무엇인가? (생성 모델 vs 판별 모델)
- ☐ CRF가 HMM보다 유연한 이유는 무엇인가? (특징 함수)
- ☐ CRF의 가장 큰 단점(수동 특징 공학)이 무엇을 의미하는지 예시를 들어 설명할 수 있는가?
- ☐ BiLSTM-CRF 아키텍처에서 BiLSTM과 CRF는 각각 어떤 역할을 담당하는가?
- ☐ 왜 BiLSTM의 출력에 Softmax 대신 CRF를 사용하는 것이 더 좋은가? (레이블 의존성)
- ☐ 일반 오토인코더(AE)로 고품질의 '새로운' 이미지를 생성하기 어려운 이유는 무엇인가? (비구조화된 잠재 공간)
- ☐ VAE는 AE와 달리 왜 인코더가 μ와 σ를 출력하는가? (랜덤성 주입)
- ☐ VAE에서 '잠재 손실(KL Divergence)'이 없다면 어떤 문제가 발생하는가? ($\sigma \to 0$)
- ☐ VAE의 잠재 손실이 어떻게 잠재 공간을 '구조화'하는가? (오버랩 강제)
- ☐ GAN의 생성자(G)와 판별자(D)의 목표는 각각 무엇인가?
- ☐ GAN의 학습 과정(2단계)을 '경찰과 위조지폐범' 비유를 사용하여 설명할 수 있는가?

9 FAQ: 초심자 주요 질문

HMM은 이제 쓸모가 없나요? 그렇지 않습니다. HMM은 모델이 단순하고 계산이 빠르며, 데이터가 매우 적을 때도 비교적 안정적으로 작동합니다. 딥러닝 모델(BiLSTM 등)이 성능이 좋지만 학습을 위해 훨씬 많은 데이터와 자원이 필요합니다. 간단한 시퀀스 문제나 초기 베이스라인 모델로 여전히 유용합니다.

CRF와 로지스틱 회귀(Logistic Regression)는 무슨 관계인가요? CRF는 종종 "로지스틱 회귀의 시퀀스(순차) 버전"이라고 불립니다.

- **로지스틱 회귀**: 여러 특징(X)을 입력받아 하나의 레이블(Y)을 예측하는 판별 모델입니다. (예: $P(Y = 1|X)$)
- **CRF**: 여러 특징(X 시퀀스)을 입력받아 **레이블 시퀀스(Y 시퀀스)**를 예측하는 판별 모델입니다. (예: $P(Y\_1, Y\_2, ...|X\_1, X\_2, ...)$)

둘 다 특징을 유연하게 사용하고 $P(Y|X)$를 직접 모델링한다는 공통점이 있습니다.

생성 모델로 VAE와 GAN 중 어느 것이 더 좋은가요? 목적에 따라 다릅니다.

- **VAE**:
  - **장점:** 학습이 안정적이며, 잠재 공간이 잘 구조화되어 데이터의 '분포'를 학습하기 좋습니다.
  - **단점:** 생성된 이미지가 GAN에 비해 다소 흐릿(Blurry)한 경향이 있습니다.
- **GAN**:
  - **장점:** 매우 선명하고 현실적인 고품질 이미지를 생성하는 데 탁월합니다.
  - **단점:** 학습이 매우 불안정하고(Minimax 최적점을 찾기 어려움) 하이퍼파라미터에 민감합니다.

10 빠르게 훑어보기 (1페이지 요약)

HMM (Hidden Markov Model)

- **개념:** 숨겨진 상태(Y)가 마코프 속성을 따르며 전이하고, 각 상태가 관찰(X)을 방출(생성)함.
- **모델:** 생성 모델 ($P(X, Y)$).
- **핵심:** 전이 확률 ($P(Y_t|Y_{t-1})$), 방출 확률 ($P(X_t|Y_t)$).
- **한계:** 미래는 오직 현재 상태(Y_t)에만 의존. 다른 관찰(X)을 참고하지 못함.

CRF (Conditional Random Field)

- **개념:** HMM의 한계 극복. 관찰 시퀀스(X) 전체를 조건으로 레이블 시퀀스(Y)의 확률($P(Y|X)$)을 직접 모델링.
- **모델:** 판별 모델 ($P(Y|X)$).
- **핵심:** 유연한 특징 함수(f_k)와 가중치(λ_k)의 조합.
- **한계:** 유용한 특징(f_k)을 **사람이 직접** 설계해야 함 (수동 특징 공학).

BiLSTM-CRF

- **개념:** CRF의 수동 특징 공학 문제를 BiLSTM으로 자동화.
- **BiLSTM (특징 추출기):** 양방향 문맥을 읽어 고품질 특징(Emission 점수)을 자동 생성.
- **CRF (레이블 결정자):** BiLSTM의 특징을 받아, 레이블 간의 전이(Transition) 규칙을 적용하여 최적의 시퀀스 결정.
- **장점:** 자동 특징 공학 + 높은 정확도.

VAE (Variational Autoencoder)

- **개념:** 일반 AE의 '비구조화된 잠재 공간' 문제를 해결한 생성 모델.
- **핵심:** 인코더가 잠재 변수를 '점'이 아닌 '확률 분포(μ, σ)'로 출력. 이 분포에서 샘플링하여 디코더에 주입 (랜덤성).
- **잠재 손실(D_{KL}):** 모델이 랜덤성을 포기하는 꼼수($\sigma \to 0$)를 막는 패널티. 분포들을 원점 근처로 모아 '오버랩'시켜 잠재 공간을 빽빽하게 채움.

GAN (Generative Adversarial Network)

- **개념:** 생성자(G)와 판별자(D)가 경쟁하며 학습하는 생성 모델.
- **G (생성자/위조범):** 노이즈(z)를 받아 가짜 데이터를 생성. D를 속이는 것이 목표.
- **D (판별자/경찰):** 진짜와 가짜를 구별. G에게 속지 않는 것이 목표.
- **장점:** 지도 데이터 없이 학습 가능. 매우 현실적인 고품질 이미지 생성.