2025-2 딥러닝응용프로그래밍 기말고사 (20241519 조예성)

이미지 상 물체 인식:
- 역할 2개: ,

위치 찾기 다른 말: ;

객체인식 1단계 종류: ,

객체인식 2단계 종류: , ,

모델 한 번에 객체인식 하는 것:

분류와 검출을 따로 진행하는 것:

이미지를 n*n 그리드로 나누고, 중심점 보유 객체 유무를 파악하는 알고리즘:

한번 예측하여 다양한 크기 및 비율을 예측하는 알고리즘:
- 고해상도 feature Map 강한 부분:
- 저해상도 Feature Map 강한 부분:

초기 영역을 통합하여 객체가 있을 법한 부분만 인식하는 알고리즘:
- 단계: , , ,
- 단점: 바운딩 박스가 많음에 따라 메모리 사용량이 ( )

기존 CNN의 입력 크기가 동일해야 하여 발생하는 문제 해결: ;
- 순서: -> -> 하기 -> -> ,
- 적용한 망:
- 겹치는 BoundingBox: ;

이미지 합성곱 한번 통과 후 나온 특징 맵을 기반으로 처리하는 신경망:
- 하는 처리: ROI 풀링
- 기능 서술: 합성곱 통과 을 n*n으로
- 순서: -> -> -> ,

후보 생성까지 CNN내부에서 하는 것:
- 기존 Fast R-CNN서 추가된 것: ;
- RPN 위치:
- 비율 고정 문제 해결:
- 미리 정한 ( )를 활용한다

신경망 훈련 통해 이미지를 픽셀 단위 분할:
- 핵심: , , , , ,

FCN 설명: 기존 합성곱 신경망의 완전연결층을 (x)합성곱으로 대체하여)출력

역 합성곱 방법: 각 픽셀 주변에

의학 사용 많고 Trade-off 하지 않는 신경망:

U-Net 활용 인식 방식:

이미지 안 객체 위치 정보 출력:

U-Net 핵심 구성: , ,

기존 모델의 작은 패턴만 보고 분석하는 단점 극복하기 위한 신경망:
- 구조: -> , -> ->
-> ->

시간의 흐름에 따라 수집된 데이터:

시간에 따라 변하는 데이터를 사용하여 추이를 분석:

예측 불가능한 돌발적 요인으로 인한 변동:

장기적인 변화 추세:

2~3년 단위 반복으로 인한 파동적 변화:

1년 주기 변동:

명확한 패턴 존재: 패턴 거의 없고 변동성 큰 데이터:

시계열 분석 모델: , , ,
- 자기회귀: , 이동평균: , 두게 합에 차분:

현재 값이 과거 값들의 선형 결합으로 설명되는 모델: 모델
- 이 모델에서 p의 의미: 사용하는

예측이 불가능한 오차:

관성이 있는 현실의 많은 데이터 다루기 적합한 모델: 모델

현재 값이 과거 오차의 누적 효과로 설명되는 모델: 모델

충격이 서서히 ( ) 값:

데이터의 흐름과 충격의 잔향이 동시에 작용한다:

추세 제거 후 과거 값과 충격의 패턴 적용:
- 수식: (, ,)
- d:

차분을 하는 이유:
- 결과: 변

적합한 경우:
부적합한 경우: ,

p: , i: , q:

시퀀스 데이터 처리 위한 신경망:

특징:

h_1, h_2, h_3 뜻: 각 시점의

RNN 종류: : , : , :

RNN 문제: ,

입력 텐서를 시간 축 방향으로 잘라서 변환해주는 함수:

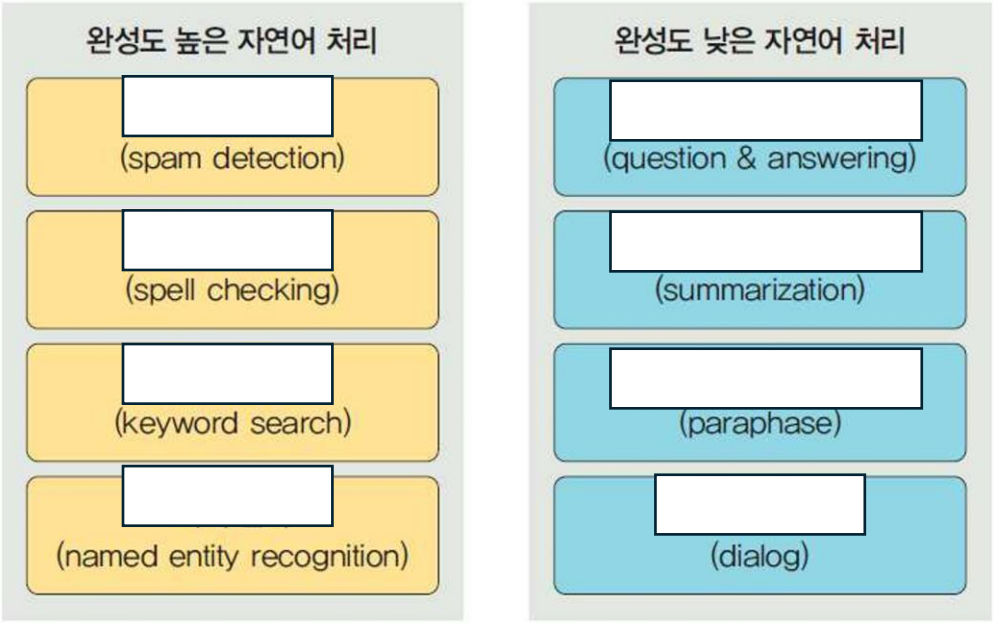
RNN 장기 기울기 소실과 장기 의존성 문제 해결:
- 도입된 개념:

LSTM 4가지 게이트: , , ,

Output Gate 출력 내용:

2개의 게이트만 갖는 구조:
- 게이트: ,
- 사용하는 스테이트:

과거 뿐만이 아닌 미래 시점 데이터 참고까지:



말뭉치 다른 용어로:

문서 나누는 단위:
- 만들기:

단어 분석에 관계는 없지만, 등장 자주:

단어를 기본 형태로 만드는 작업: 어

주어진 문장에서 품사를 식별하기 위해 붙여주는 태그:
- 품사태깅 정보: , , ,

품사 태깅 라이브러리:

자연어 처리 과정: , , ,

전처리 종류: , , ,

의미를 갖는 최소 단위 명칭:

어간 추출과 표제어 추출 차이점:

표현 방법이 다른 단어를 통합시켜 같은 단위로 만들기:
- 값이 크다고 분석에 더 ( ) 요소라고 간주할 수는 없기에

```
std_scale = preprocessing.[ ].fit(train_norm)
```

평균0, 분산 1로 정규화:

사분위수활용:

최대 최소:
- 유사:

자연어를 컴퓨터가 이해가능한 언어 형태인 벡터로 변환한 결과 혹은 과정:

대부분의 값이 0으로 채워진 임베딩:
- 예시:

출연 빈도수 이용 인코딩 하여 벡터화:
- 라이브러리, 클래스 : ,
- 기능 3개: , ,

정보 검색론서 가중치 구할 때 사용하는 알고리즘:
- 문서 내 단어 등장 빈도:
- 흔하지만 중요하지 않은 단어 빈도:
- 결과:

```
[[1.
 [0.224325 1.
 [0.       0.       1.      ]]
```

신경망 구조를 활용한 임베딩:
- 대표적:

문서의 문장단위 분리 메소드:

문장을 단어 단위로 분리:

주변 단어 보고 중심 단어 예측:

CBOW 반대로 특정 단어로 문맥단어 예측:

기존 워드 투 벡터의 단점 보완:

글로벌 동시 발생확률:
- 포함: +

어텐션 구조: ,

2018년 말 구글 공개한 AI 언어 모델:

인코더 블록 구조: ,

새로운 이미지를 만들어 내는 모델:
- 종류: , , , , ,

오토인코더 진행과정: -> ->
- 벡터가 있는 공간: ,

오토인코더 중요한 이유: , ,

오토인코더 + 확률적 분포:
- 장점:
- GAN 보다 이미지가 ( )

VAE 동작: ->

코드 13-11 인코더와 디코더 네트워크 생성

```
encoder = [
    tf.keras.layers (input_shape=(28,28,1)),
    tf.keras.layers (
        filters=32, kernel_size=3, strides=(2,2), activation="relu"
    ),
    tf.keras.layers.Conv2D(
        filters=64, kernel_size=3, strides=(2,2), activation=
    ),
    tf.keras.layers
    tf.keras.layers (units=2*2),
] ------①


decoder = [
    tf.keras.layers (units=7 * 7 * 64, activation="relu"),
    tf.keras.layers (target_shape=(7,7,64)),
    tf.keras.layers (
        filters=64, kernel_size=3, strides=(2,2), padding="SAME", activation="relu"
    ),
    tf.keras.layers.Conv2DTranspose(
        filters=32, kernel_size=3, strides=(2,2), padding="SAME", activation="relu"
    ),
    tf.keras.layers.Conv2DTranspose(
        filters=1, kernel_size=3, strides=(1,1), padding="SAME", activation="sigmoid"
    ),
] ------②
```

생성자 판별자 경쟁:
생성자는 인코더, 디코더 중:
판별자는 인코더, 디코더 중:

이미지 생성 원리:

생성자가 만든 이미지를 디코더가 원본이라고 판별하면 손실:
생성자 손실: , 판별자 손실:
생성자가 만든 이미지를 인코더가 가짜라고 판별하면 손실:
생성자 손실: , 판별자 손실:

GAN 진행 과정: -> -> ->