전반적인 요약

AI

* 머신러닝
  + 딥러닝
    - 뉴럴네트워크 (깊은 네트워크)

1. 모델 선정 (linear regression, logistic regression, MLP, CNN, RNN, transformer)
   1. 우리가 하고자하는 태스크에 맞게 골라요
      1. Regression (회귀)
         1. Linear regression
         2. Elastic net
         3. MLP
      2. Classification (분류)
      3. Image
         1. Object detection
         2. Segmentation
      4. 자연어처리 (NLP)
         1. 번역
         2. Language modeling
         3. Seq2seq generation
   2. 모델을 바꾸는 경우
      1. 언제 바꾸나요?
         1. 기존에 있던 모델이 저희가 원하는 태스크를 하지 못함
         2. 기존 모델들이 성능이 별로 안 좋을 때
      2. 어떻게 바꾸나요?
         1. 하고자 하는 태스크에 중요한 정보들(feature)이 어떻게 잘 적용이 될까?
2. 로스 함수
   1. Regression
      1. L1 loss
         1. MAE loss (mean absolute error
            1. 회귀 ?

1 2 3 -> 4

2 4 8 -> 16

* + - * 1. 1 2 3 -> -2 => 4
        2. -2 (model prediction) 과 4 (label) 간의 차이를 줄여주고 싶어요
        3. |pred-label| = 6 를 minimize
    1. L2 loss
       1. MSE loss (mean squared error)
          1. -2 (model prediction) 과 4 (label) 간의 차이를 줄여주고 싶어요
          2. (prediction – label)^2 = 36 를 minimize
    2. L1 loss 랑 L2 loss의 차이?
       1. Prediction: 0.5 label: 1.0
          1. L1 loss : 0.5
          2. L2 loss : 0.25
       2. Prediction: 2 label: 10
          1. L1 loss : 8
          2. L2 loss : 64
       3. Prediction: 0 label: 10
          1. L1 loss : 10
          2. L2 loss : 100
  1. Classification
     1. Cross entropy loss
  2. Segmentation
     1. Cross entropy loss
     2. Dice loss

1. 데이터셋을 머신러닝 혹은 딥러닝 입력으로 사용할 수 있도록 전처리하는 것
   1. 수치해석
      1. Categorical data -> one-hot vector
      2. Continuous data -> float
   2. 이미지
      1. Size 를 고정시켜서 (resize)
   3. 자연어
      1. Tokenizer
   4. 공통
      1. Normalization (standardization)
         1. Normalization
            1. Min-max normalization
            2. Mean, standard deviation normalization

Z-score normalization = standardization

1. 실제 학습
   1. Epoch
   2. Learning rate
   3. Optimizer
   4. Validation dataset으로 검증
      1. Cross-validation
2. 테스트

틀린 코드 찾는 방법

Python debugger를 사용하자.

import pdb;pdb.set\_trace()

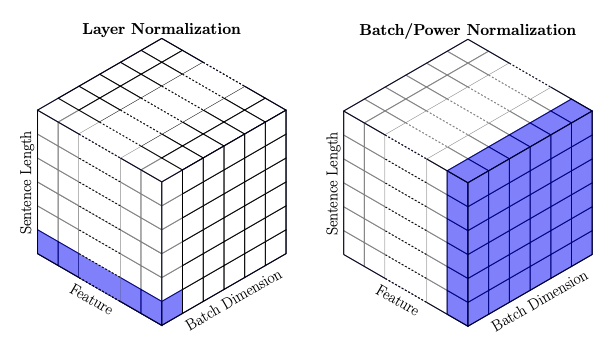
내부적으로 구현되어있는 코드를 한줄씩 실행시켜 볼 수 있다.

가중치 개념

Y = w1x1 + w2x2 + w3x3 라면, w1,w2,w3 가 가중치

각 feature가 얼마나 영향력을 주느냐가 가중치를 의미함

Layer normalization VS Batch normalization



Batch normalization : 주변의 데이터를 보고 비교할 때 유리한 방식

Layer normalization : 그렇지는 않다.

공통점 : 학습 안정성, 성능 향상에 도움이 됨

Loss VS regularization

Regularization : **모델 파라미터가 너무 큰 값을 가지지 않게 해주는 역할**

* L1 regularization (Lasso)
  + 파라미터 값이 크더라도 얼마나 큰지 상관없이 똑같이 영향
  + W1: 10, w2: 100
  + W1: 0.0001, w2:100 -> w1이 0이 되버림
* L2 regularization (Ridge)
  + W1:10, w2: 100 -> w2가 10배 더 많이 줄어들게 됨
  + W1: 0.0001, w2:100 -> w1 이 줄어드는 속도가 엄청나게 작아요
* Loss function에 더해서 추가적으로 사용 (보조 도구)

Loss : 손실 함수 (맞추고 싶은 값을 예측하도록 하는 도구)

* L1 loss
* L2 loss
* 회귀 모델의 손실 함수

분류 모델 loss는 CrossEntropyloss + L1/ L2 regularization (O)

시계열 예측은 최근에 RNN을 많이 쓰나요?

ARIMA, RNN 같은 간단한 모델에서 시작하는게 맞을 것임. 요즘은 Transformer 같은 모델로도 확장하는 추세임.

실제 대회 때 사용하는 스킬

1. State-of-the-art model (가장 좋은 모델)
2. Ensemble
   1. 회귀 모델
      1. Median
         1. outliar에 강함.
      2. Mean
         1. Outliar 도 포함해서 평균을 내기 때문에 성능이 상대적으로 안좋음
   2. 분류 모델
      1. Logit 합
      2. Softmax probability 결과값을 평균
      3. 분류 결과값 중 가장 많이 나온 것
   3. 하는 방법 (예시)
      1. Train 데이터셋을 5분할
      2. 80% 씩 학습한 5개모델을 만듦
3. Feature를 뽑는 방법
   1. 어떤 feature를 사용할까?
      1. Feature를 모두 학습시킨 모델
      2. Feature를 하나씩 빼면서 모델을 학습시켜볼 것

상관관계가 높은 feature

* 인과관계 / 상관관계
* 아이스크림 판매량 -> 워터파크 매출 (예측 됨. 하지만 인과관계 X)
* 온도 -> 워터파크 매출 (인과관계 있음)
* 키-몸무게 -> 비만에 의한 합병증
* BMI 지수 -> 비만에 의한 합병증

Decision tree, gradient boosting

* 캐글 수업할 때 다루도록 하겠습니다

Vector의 개념, 딥러닝에서 내적이 가지는 의미

Scalar : 1,2,3,4, 1.1, 5.2, -3.2

Vector: [1,2,3]

Matrix: [1,2,3]

[3,3,5]

Tensor: [[1,2,3], [1,2,3]

[3,3,5], [3,3,5]]

헷갈린다면 Gilbert strang - Linear algebra 강의를 들어보시기 바랍니다.

내적 (dot product) : vector들간의 **관계성**

V1 o v2 = |v1||v2| cos theta

Transformer key, query

패키지 관리 하는 방법

Conda environment