분류

목차

- 분류 문제란?
- Titanic Dataset
- MNIST Dataset

분류 문제란?

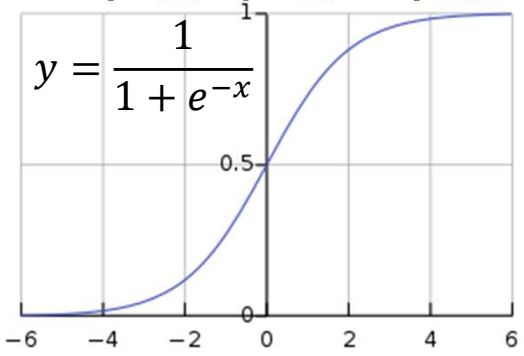
- 예측하고자 하는 값이 부류값(또는 소속)인 경우

- 예) 개와 고양이 분류 문제(개 or 고양이)
  X-ray 데이터로부터 암 발병 여부(True or False)
  MNIST 데이터셋(0~9 까지의 손 글씨 데이터)
  ImageNet 데이터셋(1000가지 클래스의 이미지 데이터)

분류 문제란? – Binary Classification

- 분류 해야 할 클래스가 두 가지(0 or 1)인 경우.
  로지스틱 회귀(Logistic Regression)이라고 칭하기도 함.

- 로지스틱 회귀는 로지스틱 함수를 사용하는 회귀.



로지스틱 함수의 일종인
시그모이드 함수

AIoT

분류 문제란? – Binary Classification

- 로지스틱 함수는 (0,1)의 출력을 가짐
  ->정답(0 or 1)에 최대한 가깝게 출력하는 것이 목표

- 손실 함수는 BCEWithLogitsLoss() 사용

$$J(\Theta) = -\frac{1}{n}\sum_{i=1}^{n}\{y_i \log(\sigma(o_i)) + (1-y_i)\log(1-\sigma(o_i))\}$$

- PyTorch의 BCEWithLogitsLoss()는 BCELoss()이전에 Sigmoid()를 적용 시킨 것

분류 문제란? – Binary Classification

- Binary Classification의 경우 모델의 마지막 layer는 nn.Linear(in_feature,1)인 경우가 많음.
- 해당 경우에는 손실 함수로 BCEWithLogitsLoss() 사용.
- 만약 nn.Sigmoid()로 끝나는 경우 BCELoss() 사용.

분류 문제란? – Multinomial Classification

- 분류 해야 할 클래스가 3가지 이상인 경우.
  소프트맥스 회귀(Softmax Regression)이라고 칭하기도 함.

$$softmax(x[i]) = \frac{e^{x[i]}}{\sum_j^k e^{x[j]}}$$

- 소프트맥스 회귀는 출력이 [0,1]이다.
  각 출력의 의미는 클래스 별 확률(우도,likelihood)로 해석 가능.

분류 문제란? – Multinomial Classification

- 예를 들어 세 클래스 (A,B,C)를 분류하는 문제라면,
  한 입력의 출력은 다음과 같을 수 있음
  o=[0.8, 0.1, 0.1]
  이 경우, 해당 데이터는 클래스 A에 속할 가능성이 높음.

- 손실 함수는 CrossEntropyLoss()를 사용.

$$J(\Theta) = -\frac{1}{n}\sum_{i=1}^{n} \log(softmax(o_i[y_i]))$$

분류 문제란? – Multinomial Classification

- N개의 클래스를 분류해야 하는 문제의 경우 모델의 마지막 layer는 nn.Linear(in_features, N)인 경우가 많음.
  이 경우, CrossEntropyLoss() 사용.

- 만약 nn.Softmax()로 끝나는 경우, NLLLoss() 사용.

Titanic Dataset

- 타이타닉호의 탑승자 데이터를 바탕으로 생존여부 예측
- Binary Classification의 대표적인 데이터
- Train data: 789 rows, Test data: 89 rows
- 출처: Kaggle

Titanic Dataset

- 실제 데이터에서 구현상 편의를 위해 일부 column만 다룰 예정.
- Columns 설명
  Pclass: 좌석의 등급(1=1st, 2=2nd, 3=3rd)
  Sex: 성별(0=남성, 1=여성)
  Age: 나이
  Siblings/Spouses Aboard: 배에 탑승한 형제자매/배우자의 수
  Parents/Children Aboard: 배에 탑승한 부모/자녀의 수
- Target
  Survived: 생존 여부(0=사망, 1=생존)

Titanic Dataset

- Baseline
  Model: Linear(6,10)->Sigmoid()->Linear(10,1)
  Loss: BCEWithLogitsLoss()
  Optimizer: SGD(lr=0.001)
  Num epochs: 100
  Early stop: None

- Result: 0.6830 (BCEWithLogitsLoss), 62.92% (Accuracy)

MNIST Dataset

- 0부터 9까지의 흑백 손 글씨 이미지 데이터셋
- Multinomial Classification과 Computer Vision 분야에서 가장 유명한 데이터셋
- Train data: 60,000 rows, Test data: 10,000 rows
- 출처: Yann Lecun

MNIST Dataset

- MNIST 데이터셋은 28x28 크기의 데이터셋.
- 학습을 진행할 때는 데이터를 784 크기의 벡터로 변환하여 모델에 입력
- 흑백 이미지이기 때문에 모든 경우의 수는 2^{784}로 학습 데이터의 크기(60,000)보다 매우 큼.

MNIST Dataset

- Baseline
  Model: Flatten()->Linear(784,1000)->Sigmoid()->Linear(1000,10)
  Loss: CrossEntropyLoss()
  Optimizer: SGD(lr=0.001)
  Num epochs: 10
  Early stop: None

- Result: 2.1438 (CrossEntropyLoss), 56.48% (Accuracy)

Titanic Dataset & MNIST Dataset

- 성능을 향상시키기 위해

1. 모델의 구조 변경(layer 추가 또는 제거, 활성 함수 변경)
2. Learning rate 조절
3. 최적화 알고리즘 변경(SGD or Adam, …)
4. num_epochs 조절
5. 학습 데이터 변경

등을 적용하여 계속하여 학습, 성능이 향상된다면 원인 분석

6장 Preview

- 이미지나 영상 데이터를 학습하는 데 효과적인 Convolutional Neural Network(CNN)

- 기존에 잘 학습된 모델을 사용하는 Transfer Learning

- 다음 데이터 셋에 대해 실습 진행
  - Dogs and Cats Dataset: 강아지와 고양이 분류

References

[1] https://ratsgo.github.io/machine%20learning/2017/04/02/logistic/

[2] https://github.com/hunkim/DeepLearningZeroToAll

[3] https://www.kaggle.com/c/titanic

[4] https://sdc-james.gitbook.io/onebook/4.-and/5.1./5.1.3.-mnist-dataset

[5] http://yann.lecun.com/exdb/mnist/