

ML/DL Basic Week02

Multinomial Classification

Application & Tips

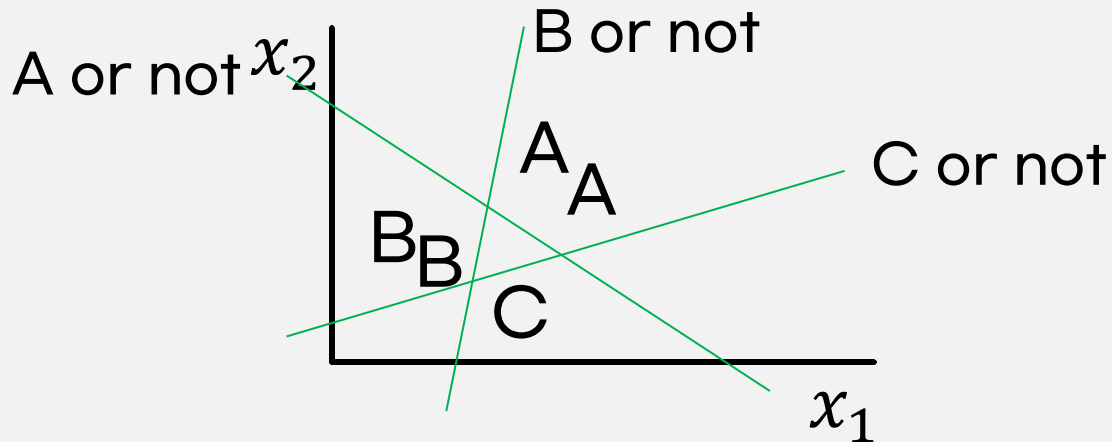
```
lookup.KeyValue  
f.constant(['em  
=tf.constant([0  
ce = tf.lookup.StaticV  
init,  
num_oov_buckets=5)  
  
lookup.StaticVocabular  
initializer,  
num_oov_buckets,  
lookup_key_dtype=None  
name=None,  
experimental_is_open
```

Multinomial Classification

다변수 분류

- 1) 3개 이상으로 분류할 수 있는 데이터가 있을 때 사용
- 2) 이진분류를 이용해 다변수 데이터를 분류할 수 있음

Ex) A or not, B or not, C or not



Multinomial Classification

다변수 분류

$$3) \begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix}$$

- A, B, C 구현 시, 3개의 독립된 벡터로 계산 -> 구현 어려움
- Weight 벡터를 하나로 합쳐 계산

Multinomial Classification

소프트 맥스 함수

1) 각각의 y_i 는 0과 1 사이의 값

2) $\sum_i y_i = 1$

3) 가장 높은 확률을 갖는 값 1개를 예측 값으로 만들기 위해

‘ONE-HOT’ 인코딩을 함

Multinomial Classification

Cost 함수

1) Cross-Entropy 사용

$$2) D(S, L) = - \sum_i L_i \log(S_i)$$

L 은 실제의 값, S 는 예측한 값

Application & Tips

학습률(Learning Rate)

1) 학습률이 큰 경우

- 최솟값으로 수렴하지 않고 근처에서 왔다 갔다 하는 경우 발생
- 곡선 아래가 아닌 위로 발생하는 경우 발생 -> overshooting
- Cost 값이 줄어들지 않는 경우 의심

2) 학습률이 작은 경우

- 경사면을 이동하는 것이 더더 최저점에 이르지 못하는 상태 발생
- Cost 값이 변화가 없는 경우 의심

일반적으로는 0.01을 사용하나 cost 값을 통해 결정!

Application & Tips

데이터 전처리(Data Preprocessing)

: noisy data가 학습에 사용되지 않도록 제거하거나 수정하는 과정

1) 표준화(Standardization)

- 값의 범위를 평균 0, 분산 1이 되도록 바꾸는 것

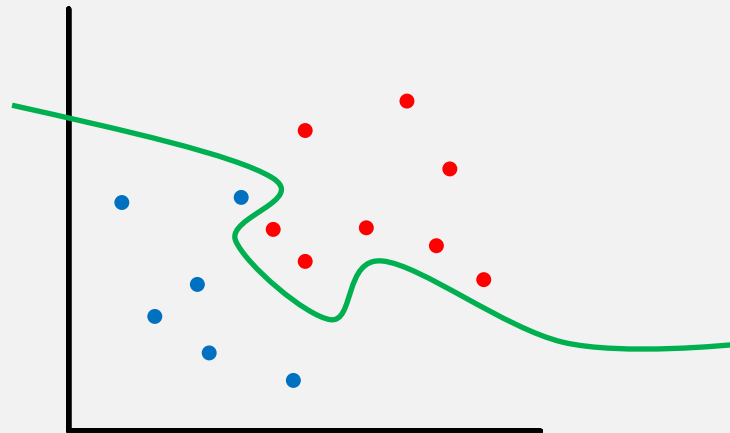
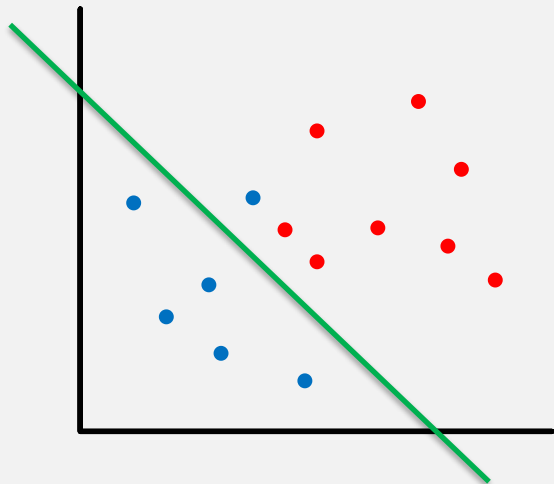
2) 정규화(Normalization)

- 값의 범위를 0과 1 사이의 값으로 바꾸는 것

Application & Tips

Overfitting

1) 학습 데이터에 적합하나 실제 사용 시 정확도가 떨어지는 경우



Application & Tips

Overfitting

2) Solutions

- Feature Normalization
- Regularization
- More Data and Data Augmentation
- Dropout
- Batch Normalization

Application & Tips

Learning

1) Online

- 학습 데이터가 많아 한 번에 다 학습시키기 어려운 경우 이용
- 데이터를 순차적으로 한 개씩 또는 미니배치라 부르는 작은 묶음 단위를 주입하여 시스템 훈련

2) Batch

- 시스템이 점진적으로 학습 X, 가용한 데이터 모두 사용해 훈련

L1 loss & L2 loss

1) L1 loss(Manhattan Distance)

- 벡터의 요소에 대한 절댓값의 합
- 여러가지 path를 가짐
- 실제 값과 예측치 차이의 절댓값, 오차들의 합

2) L2 loss(Euclidean Distance)

- 원소의 제곱의 합에 루트, 즉 원점으로부터 직선 거리
- Unique shortest path를 가짐
- 실제 값과 예측치 차이의 제곱의 합

PCA

1) 차원 축소와 변수추출 기법으로 이용

2) 원리

- 데이터를 1차원으로 축소하기 위해 정사영 사용
- 분산이 보존될 수 있도록 하는 축 찾기(정사영을 하기 위한 축)
- 공분산 행렬의 고유 벡터 값 중 가장 큰 값을 주성분으로 함