딥 러닝

뉴럴 네트워크

사람의 뇌를 모방하여 기계가 학습하도록 고안된 알고리즘

인공신경망이라고도 하고 딥 러닝이라고도 부른다.

1. 과거의 데이터를 준비한다
2. 모델의구조를 만든다
3. 데이터로 모델을 학습시킨다
4. 모델을 이용한다

표를 다루는 도구 판다스

Import pandas as pd

파일을읽어오는 방법

Pd.read\_csv(‘\경로\파일명.csv’)

모양 확인하기

print(데이터.shape)

칼럼 선택하기

데이터[[‘칼럼명1’,’칼럼명2’,’칼럼명3’]]

칼럼 이름 출력하기

Print(데이터.columns)

맨 위 5개 관측치 출력하기

데이터.head()

Model.fit(독립, 종속, epochs=10) 10번 반복 학습

Loss – 각 학습이 지날 때 마다 해당 모델이 정답에 얼마나 가까운지를 나타냄

정답과 예측치의 차이 제곱의 평균 0에 가까울수록 학습이 잘된 것임

텐서 플로우 라이브러리 사용

Import tensorflow as tf

#모델 만들기

X = tf.keras.layers.Input(shape=[1])#독립데이터의 데이터 수, 인풋레이어를 만든다

Y = tf.keras.layers.Dense(1)(X)#종속변수 column 1개, 덴스 레이어

model = tf.keras.models.Model(X,Y)#종속변수와 독립변수를 넣어서 모델 생성

model.compile(loss='mse')#모델이 학습 할 방법

#모델 학습

model.fit(독립,종속,epochs=10)#model.fit(독립,종속,epochs=10000, verbose=0) 학습결과 표시 안함

#모델을 이용한다

model.predict(독립)

#정답 확인

종속

#모델 적용해서 예측하기

model.predict([[15]])

모델의 수식 확인

Model.get\_weight()

X 는 입력

Y 는 출력

아이리스 예제의 경우 품종 데이터가 문자형인 범주형 데이터로 기존의 회귀알고리즘이 아닌 분류 알고리즘을 사용해야 한다.

범주형 데이터는 수식에 사용할 수 있는 과정을 거친다

모든 범주를 column으로 만들고 해당 범주의 column 값을 증가시켜 0 또는 1로 바꾸는데

이를 원핫인코딩이라고 한다.

코드로는

아이리스 = pd.get\_dummies(아이리스)

이때 각 column들이 출력이 되기 때문에 y설정의 인자값이 바뀌는 것이다.

Softmax 0%~100% 로 비율로 예측시켜줌

값을 softmax로 감싸면 값을 0~1 사이의 값으로 받을 수 있다

코드로는

Y = tf.keras.layer.Dense(3,activation=’softmax’)(X)

회귀에 사용하는 loss = ‘mse’

분류에 사용하는 loss = ‘categorical\_crossentropy’

Model.compile(loss = ‘categorical\_crossentropy’,metrics=’accuracy’)

하면 정확도를 보여준다 1이면 다맞는 것

히든레이어

X 인풋레이어 와 y 아웃풋 레이어 사이에 추가되는 것을 히든레이어 라고 부른다

X = tf.keras.layers.Input(shape=[13])

H = tf.keras.layers.Dense(10, activation='swish')(X)

Y = tf.keras.layers.Dense(1)(H)

뒤 괄호에 오는 것이 바뀜에 주의.

X = tf.keras.layers.Input(shape=[13])

H = tf.keras.layers.Dense(10, activation='swish')(X)

H = tf.keras.layers.Dense(5, activation='swish')(H)

H = tf.keras.layers.Dense(3, activation='swish')(H)

H = tf.keras.layers.Dense(3, activation='swish')(H)

Y = tf.keras.layers.Dense(1)(H)

다음과같이 더 늘릴 수 있다.

#모델에 히든레이어 됬는지 확인용

model.summary()

param은 가중치

print(아이리스.dtypes) 하면 각 colum들이 어떤 자료형인지 보여줌

특정 열의 타입을 바꾸고 싶을 때

아이리스[‘품종’] = 아이리스[‘품종’].astype(‘category’)

하면 된다.

이후 원핫인코딩 해주면 된다.

Na값 체크

아이리스.isna().sum()

하면 각 열에 대해 na값의 수를 보여줌

Na값 체우기

Mean = 아이리스.[‘꽃잎폭’].mean() #꽃잎 폭의 평균을 mean에 저장

아이리스.[‘꽃잎폭’]= 아이리스.[‘꽃잎폭’].fillna(mean) #na 값을 mean 값으로 바꿈

모델 향상

H = tf.keras.layers.Dense(10,activation='swish')(X)

이 코드는

H = tf.keras.layers.Dense(10)(X)

H = tf.keras.layers.Activation(‘swish’)(H)

로 바꾸어 쓸 수 있다.

이렇게 쓰는 이유는 모델 향상을 위해 사용할 BatchNomalization이 저 둘 사이에 있을 때 효과적이기 때문이다.

H = tf.keras.layers.Dense(10)(X)

H = tf.keras.layers.Activation('swish')(H)

H = tf.keras.layers.BatchNormalization()(H)

멀티레이어와 병행하면 더 성능이 올라가는 것을 볼 수 있다.