

[Deep Learning] 딥러닝의 기초 - 2.1 딥러닝의 기초 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2021-01-31 오전 4:34

URL: <https://continuous-development.tistory.com/227>

Deep Learning

[Deep Learning] 딥러닝의 기초 - 2.1 딥러닝의 기초

2021. 1. 31. 04:33 수정 삭제 공개



2.1 신경망과의 첫 만남

작업순서는 다음과 같다.

먼저 훈련 데이터 `train_images` 와 `train_labels` 를 네트워크에 주입한다.

그러면 네트워크는 이미지와 레이블을 연관시킬 수 있도록 학습된다.

마지막으로 `test_images`에 대한 예측을 네트워크에 요청한다.

그리고 이 예측이 `test_labels`와 맞는지 확인한다.

신경망의 구조

```
from keras import models

from keras import layers

network = models.Sequential()

network.add(layers.Dense(512, activation='relu', input_shape=(28*28)))

network.add(layers.Dense(10, activation='softmax'))
```

신경망의 핵심 구성요소는 일종의 데이터 처리 필터라고 생각할 수 있는 층이다.

층은 주어진 문제에 더 의미 있는 표현을 입력된 데이터로부터 추출한다.

딥러닝 모델은 데이터 정제 필터가 연속되어있는 데이터 프로세싱을 위한 여과기와 같다.

이 예에서는 조밀하게 연결된 신경망 층인 `Dense` 층 2개가 연속 되어 있다. 두번째 층은 10개의 확률 점수가 들어 있는 배열을 반환하는 소프트맥스 층이다. 각 점수는 현재 숫자 이미지가 10개의 숫자 클래스 중 하나에 속할 확률이다.

신경망이 훈련 준비를 마치기위해서 컴파일 단계에 포함될 세 가지가 더 필요하다.

손실함수 - 훈련 데이터에서 신경망의 성능을 측정하는 방법으로 네트워크가 옳은 방향으로 학습될 수 있도록 도와준다.

옵티마이저 - 입력된 데이터와 손실 함수를 기반으로 네트워크를 업데이트하는 메커니즘이다.

훈련과 테스트 과정을 모니터링할 지표 - 여기에서는 정확도만 고려한다.

#컴파일 단계

```
network.compile(optimizer='rmsprop',  
                loss='categorical_crossentropy',  
                metrics = ['accuracy'])
```

훈련을 시작하기 전에 데이터를 네트워크에 맞는 크기로 바꾸고 모든 값을 0과 1사이로 스케일을 조정한다.

#이미지 데이터 준비하기

```
train_images = train_images.reshape((60000, 28*28))  
  
train_images = train_images.astype('float32') / 255  
  
test_images = test_images.reshape((10000, 28*28))  
  
test_images = test_images.astype('float32') / 255
```

#레이블 준비하기

```
from keras.utils import to_categorical

train_labels = to_categorical(train_labels)

test_labels = to_categorical(test_labels)

# fit

network.fit(train_images, train_labels, epochs=5, batch_size=128)
```

2.2 신경망을 위한 데이터 표현

텐서 - 데이터를 위한 컨테이너로서 거의 수치형 데이터를 다루므로 숫자를 위한 컨테이너 이다. 텐서는 임의의 차원 개수를 가지는 행렬의 일반화된 모습이다.

2.2.1 스칼라(0D텐서)

스칼라 - 하나의 숫자만 담고 있는 텐서, 스칼라 텐서의 축 개수는 0이다 (ndim==0) 텐서의 축 개수를 랭크라고도 부른다.

```
import numpy as np
```

```
x = np.array(12)
```

```
x = > array(12)
```

```
x.ndim => 0
```

2.2.2 벡터(1D 텐서)

벡터/1D텐서 - 숫자의 배열

```
x = np.array([12, 3, 6, 14, 7])
```

5개의 원소를 가지고 있으면 5차원 벡터라고 한다.

5D벡터와 5D 텐서는 다르다.

2.2.3 행렬(2D텐서)

행렬 - 벡터의 배열이 행렬 / 2D 텐서

```
x = np.array([[5, 78, 2, 34, 0],  
              [6, 79, 3, 35, 1],  
              [7, 80, 4, 36, 2]])
```

2.2.4 3D 텐서와 고차원 텐서

3차원 이상의 행렬로 구성되어 있는 행렬

```
x = np.array([[[[5, 78, 2, 34, 0],  
                [6, 79, 3, 35, 1],  
                [7, 80, 4, 36, 2]],  
              [[5, 78, 2, 34, 0],  
                [6, 79, 3, 35, 1],  
                [7, 80, 4, 36, 2]],  
              [[5, 78, 2, 34, 0],  
                [6, 79, 3, 35, 1],  
                [7, 80, 4, 36, 2]]]])
```

2.2.5 핵심속성

텐서는 3개의 핵심 속성으로 정의된다.

축의개수(랭크)

크기 : 텐서의 각 축을 따라 얼마나 많은 차원이 있는지를 나타낸 파이썬 튜플이다.

행렬의 크기 (3,5) 이고 3D 텐서의 크기는(3,3,5) 이다. 벡터의 크기는 (5,) 처럼 1개의 원소로 이루어진 튜플이다.

스칼라는 () 처럼 크기가 없다.

데이터 타입 텐서에 포함된 데이터 타입

(60000, 28, 28) = > 28 * 28 크기의 정수 행렬 6만개가 있는 배열

2.2.6 넘파이로 텐서 조작하기

슬라이싱 - 배열에 있는 특정 원소들을 선택하는 것

```
=> train_images[10:100]
```

```
=> my_slice = train_images[:, 14 :, 14 :]
```

2.2.7 배치 데이터

일반적으로 딥러닝에서 사용하는 모든 데이터 텐서의 첫 번째 축은 샘플 축이다.

```
batch = train_images[:128]

batch = train_images[128:256]

batch = train_images[128*n:128*(n+1)]
```

이런 배치 데이터를 다룰 때는 첫 번째 축을 배치 축 또는 배치 차원이라고 한다.

2.2.8 텐서의 실제 사례

- 벡터데이터 : 크기의 2D 텐서
- 시계열데이터 또는 시퀀스 데이터 : 크기의 3D 텐서
- 이미지 : 크기의 4D 텐서
- 동영상 : 크기의 5D 텐서

2.2.9 벡터 데이터

하나의 데이터 포인트가 벡터로 인코딩될 수 있으므로 배치 데이터는 2D 텐서로 인코딩될 것이다. 여기서 첫번째 축은 샘플 축이고 2 번째 축은 특성 축이다.

각 사람은 개의 값을 가진 벡터로 구성되고 10만명이 포함된 전체 데이터셋은 $(10000, 3)$ 크기의 텐서로 저장된다.

2.2.10 시계열 데이터 또는 시퀀스 데이터

데이터에서 시간이 중요할 때는 시간 축을 포함하여 3D 텐서로 저장된다.

ex)주식 가격 데이터셋

2.2.11 이미지 데이터

흑백 이미지 $(128, 256, 256, 1)$

컬러 이미지 $(128, 256, 256, 3)$

이미지 텐서의 크기를 지정하는 방식은 두가지이다. 채널 마지막 방식과 채널 우선 방식이다.

채널 마지막 (samples, height, width, color_depth)

채널 우선(samples, color_depth, height, width)

2.2.12 비디오 데이터

5D 텐서이다.

(samples, frames, height, width, color_depth)

'Deep Learning' 카테고리의 다른 글

[Deep Learning] 딥러닝의 기초 - 2.1 딥러닝의 기초

[Deep Learning] 딥러닝의 기초 - 1. 3 왜 딥러닝일까?

[Deep Learning] 딥러닝의 기초 - 1.2 딥러닝 이전: 머신러닝의 간략한 역사

[Deep Learning] 딥러닝의 기초 - 1.1 인공지능과 머신러닝, 딥러닝

기초

딥러닝

텐서



나아무늘보

혼자 끄적끄적하는 블로그 입니다.