

Pytorch_기초



이초연 강사

2024. 11. 07

Pytorch

목차

1. Pytorch란?
2. 손실함수
3. 미분(Autograd)
4. 경사하강법

1. Pytorch란?

- 페이스북의 인공지능 연구팀이 개발
- 파이썬 기반 머신러닝 라이브러리
- 딥러닝 활용- 이미지 인식, 음성 인식, 객체 생성 등
- 이미지 처리, 자연어 처리, 시각화 등
- 자동 미분

Pytorch란?

파이토치 한국 사용자 모임

<https://tutorials.pytorch.kr/>

- <https://pytorch.org/get-started/locally/>
- 구글 Colab 사용
- 버전 확인 `print(torch.__version__)`



PyTorch 빌드	Stable (2.4.0)		Preview (Nightly)	
OS 종류	Linux		Mac	Windows
패키지 매니저	Conda	Pip		LibTorch
언어	Python		C++ / Java	
플랫폼	CUDA 11.8	CUDA 12.1	CUDA 12.4	ROCm 6.1
이 명령을 실행하세요:	pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121			

Pytorch란?

■ 텐서(tensor)

- * 데이터 표현을 위한 기본 구조
- * 텐서는 데이터를 담기 위한 컨테이너(container)로서 **수치형 데이터를 저장**
- * GPU를 사용한 연산 가속 가능
- 3차원부터 텐서
- 스칼라, 벡터, 행렬
- rank(차원): ndimension()

■ #라이브러리 импорт

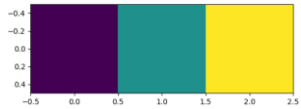
- import torch # 파이토치
- import numpy as np # 수치 계산
- import pandas as pd # 데이터 분석
- import matplotlib.pyplot as plt # 시각화

Pytorch란?

텐서 생성, 1차원 텐서, 벡터

```
tensor = torch.tensor([1, 2, 3])
```

```
print(tensor)
```



1D Tensor(Vector)

Rank: 1

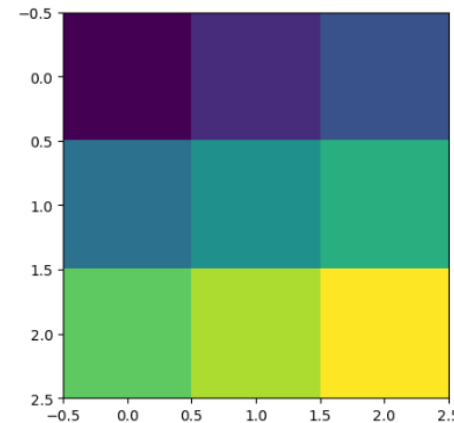
Shape: (3,)

■ ## 텐서 생성, 2차원 텐서, 행렬

```
tensor=torch.tensor([[1, 2, 3],[4, 5, 6],[7, 8, 9]])
```

```
print(tensor)
```

```
plt.imshow(tensor.view(3,3))
```



Pytorch란?

■ # 텐서 생성, 3차원 텐서

```
tensor = torch.tensor([[[1,2,3], [4,5,6],  
[7,8,9]], [[1,2,3], [4,5,6], [7,8,9]], [[1,2,3], [4,5,6],  
[7,8,9]]])
```

```
print(tensor)
```

```
print("size", tensor.size())
```

```
print("shape", tensor.shape)
```

```
print("차원", tensor.ndimension())
```

결과

```
tensor([[[1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]],  
        [[1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]],  
        [[1, 2, 3],  
        [4, 5, 6],  
        [7, 8, 9]]])  
size torch.Size([3, 3, 3])  
shape torch.Size([3, 3, 3])  
차원 3
```

2. 손실함수

- 손실함수(loss function): 모델이 입력 x 와 가중치 w 의 곱으로 출력되는 예측 값과 실제 값 사이의 **오차**를 수치화하는 함수로 모델의 성능을 평가함
- 차이 값이 0에 가까울수록 무손실에 가깝다

$$\text{cost}(W) = \frac{1}{m} \sum_{i=1}^m (Wx^{(i)} - y^{(i)})^2$$

- 손실 함수를 최소화하는 **파라미터를 업데이트 해야함 (가중치)**
 - 예측 성능 향상됨

손실함수

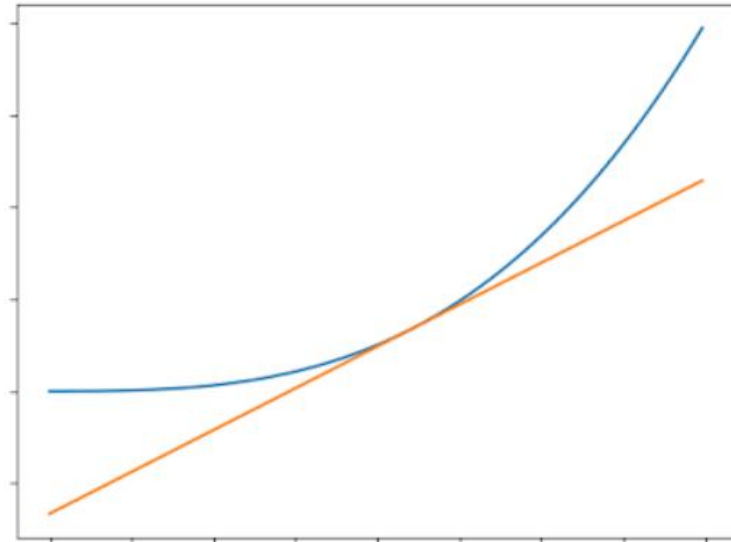
- 손실함수 종류
- 평균제곱오차(MSE): 회귀 문제
`torch.nn.MSELoss(reduction='mean')`
- 교차엔트로피 오차(CEE): 분류 문제
- `loss_fn = nn.CrossEntropyLoss()`

3. 미분(Autograd)

#Autograd

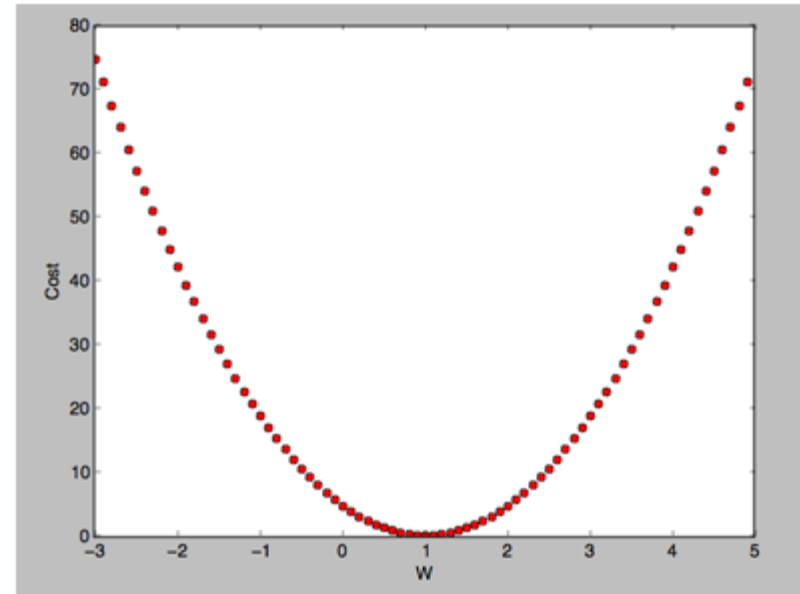
- 파이토치에서 자동 미분을 수행
- 손실함수의 오차가 최소화 되도록 파라미터의 미분 계산이 필요함
- torch.autograd 자동 미분 계산
- 모든 계산 그래프에 대한 변화도의 자동 계산을 지원함

`w = torch.tensor(1.0, requires_grad=True)`



4. 경사하강법

- 손실함수 오차의 최솟값을 찾기 위해 미분하여 기울기를 낮은 쪽으로 이동시켜 최적의 값을 찾는 방법
- 파라미터 : 가중치, 학습률(lr), 에폭, 배치 사이즈 등



경사하강법

옵티마이저 설정

■ `torch.optim.Adam(params, lr=0.001, eps=1e-08)`

학습률, 배치 사이즈, 에폭수

`learning_rate = 1e-3`

`batch_size = 64`

`epochs = 5`

작업 환경

작업 환경 설치

- colab(코랩), 파이썬, 파이토치

무료

노트 설정

런타임 유형

Python 3

하드웨어 가속기 ?



CPU



T4 GPU



A100 GPU



L4 GPU



TPU v2-8

프리미엄 GPU를 이용하시겠어요? [추가 컴퓨팅 단위 구매](#)



모든 실행에서 첫 번째 셀 또는 섹션 자동 실행



이 노트를 저장할 때 코드 셀 출력 생략

취소

저장

유료

A100GPU 월 9000원

colab

#구글 로그인

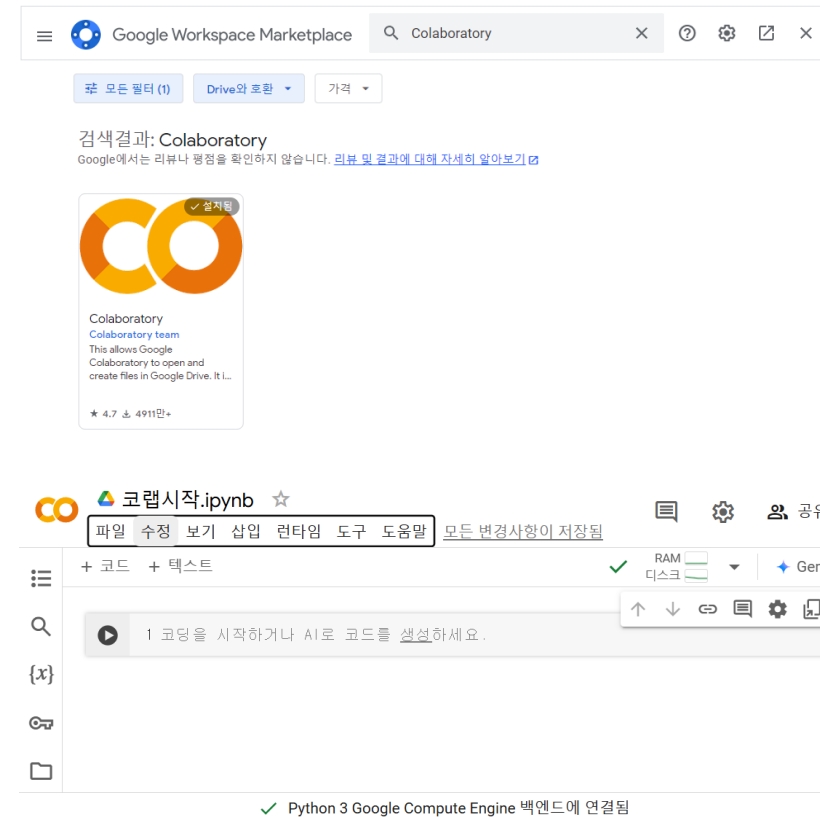
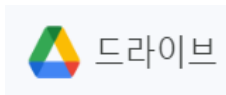
- 구글 앱(점9개)- 구글 드라이브

새로 만들기- 더보기- 연결할 앱 더보기

(기존 설치 후: 신규 - 더보기- Google Colaboratory)

Colaboratory 검색- 앱 설치

연결- RAM과 디스크 표시



colab

GPU 연결

수정- 노트 설정

노트 설정

런타임 유형

Python 3

하드웨어 가속기 ?

☐ CPU ☒ T4 GPU ☐ A100 GPU ☐ L4 GPU ☐ TPU v2-8

프리미엄 GPU를 이용하시겠어요? [추가 컴퓨팅 단위 구매](#)

☐ 모든 실행에서 첫 번째 셀 또는 섹션 자동 실행

☒ 이 노트를 저장할 때 코드 셀 출력 생략

취소

저장



코랩시작.ipynb ☆

파일 수정 보기 삽입 런타임 도구



+ 코드 + 텍스트



✓
0초



```
1 print("Hello, pytorch!")
```



Hello, pytorch!

