

Logistic Regression vs Deep NN 비교 실험

Logistic Regression vs Deep NN 비교 실험

1. 과제 내용

다음 두 가지 모델을 주어진 데이터 세트에 대해 직접 구현하고 이진 분류 성능을 비교

- 1) Logistic Regression
- 2) Deep NN (3 hidden layers)

➡ 두 가지 모델을 직접 구현하고, 주어진 데이터셋에 대해 결과를 비교해서 보고서를 작성

Logistic Regression vs Deep NN 비교 실험

2. 데이터세트 : Chest_Xray_Pneumonia

<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

- 1) 정상 상태 이미지
- 2) 폐렴 상태 이미지

chest_xray > train >

이름

NORMAL

PNEUMONIA

chest_xray > test >

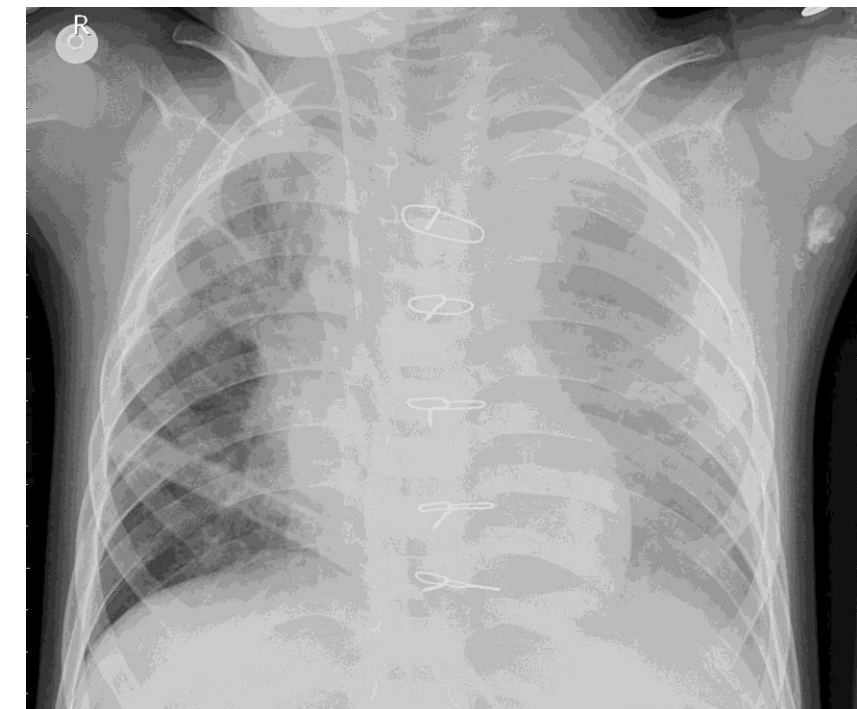
이름

NORMAL

PNEUMONIA



VS



3. 평가 기준

항목	점수	평가 기준 설명
코드 실행 여부	20	코드 정상 실행
두 가지 모델 구현 및 결과	30	Logistic regression, Deep NN 학습 코드 구현 + 각 모델 학습/검증/테스트 정확도 결과
정확도 시각화 그래프 포함	10	학습/검증/테스트 정확도 비교를 위한 시각화 (bar plot) 포함 여부
모델 구조 설명	10	각 모델의 구조 설명 (레이어 수, hidden unit 수, activation function 등) 포함 여부
성능 비교 분석	30	두 모델의 정확도 차이 원인, epoch 변경(2,10,20)에 따른 overfitting/underfitting 분석
지각 제출	-	제출 마감 이후 제출 시 획득 점수의 50%만 인정

4. 제출 파일 구성

1) Code_학번_이름.ipynb (코드 구현 파일)

✓ 코드 구성

- ① 전처리 부분(예시 참고)
- ② Class 로 정의된 각 모델 부분
- ③ 학습 진행하는 함수
- ④ 테스트 진행하는 함수
- ⑤ 학습 함수를 통해 학습을 진행하는 부분
- ⑥ 테스트 함수를 통해 정확도를 도출하는 부분
- ⑦ 학습/검증/테스트 정확도 비교를 위한 시각화 Bar plot 도출 부분

4. 제출 파일 구성

2) Report_학번_이름.pdf (분석 리포트)

① 모델 구조 요약

- a. 각 모델의 layer수, hidden unit구성, activation function 사용 내역 등 명확히 기술

② 정확도 비교 및 해석

- a. Bar plot을 통한 정확도 비교
- b. 정확도 차이의 원인 분석

③ 추가 분석

- a. epoch 변경(2,10,20)에 따른 각 모델의 학습/검증/테스트 정확도
- b. epoch 변경(2,10,20)에 따른 각 모델의 overfitting/underfitting 여부 분석

5. 제출 유의 사항

- 1) Pytorch 사용 필수
- 2) 제출 파일명 통일 : Code_학번_이름.ipynb / Report_학번_이름.pdf
- 3) LMS 과제란에 제출
- 4) 문의사항 : starhyunee@naver.com
- 5) 제출기한 : 4/20(일) 23:59까지

6. 참고 코드

✓ 전처리 부분(예시)

```
root_dir = os.path.expanduser("각자경로/chest_xray")
transform = transforms.Compose([
    transforms.Grayscale(num_output_channels=1),
    transforms.Resize((128, 128)),
    transforms.ToTensor()])
input_dim = 128 * 128

## 학습 / 테스트 데이터 split
trainval_ds = datasets.ImageFolder(os.path.join(root_dir, 'train'), transform=transform)
test_ds = datasets.ImageFolder(os.path.join(root_dir, 'test'), transform=transform)
train_size = int(0.8 * len(trainval_ds))
val_size = len(trainval_ds) - train_size
train_ds, val_ds = random_split(trainval_ds, [train_size, val_size], generator=torch.Generator().manual_seed(42))

# DataLoader 구성
train_loader = DataLoader(train_ds, batch_size=32, shuffle=True)
val_loader = DataLoader(val_ds, batch_size=32)
test_loader = DataLoader(test_ds, batch_size=32)
```


6. 참고 코드

✓ 모델 부분(예시)

```
class LogisticModel(nn.Module):  
    def __init__(self, input_dim):
```

```
class DeepNN(nn.Module):  
    def __init__(self, input_dim):
```

직접 구현



Logistic Regression vs Deep NN 비교 실험

6. 참고 코드

✓ 학습 부분(예시)

```
def train(model, loader, optimizer, criterion):  
    model.train()
```

직접 구현



6. 참고 코드

✓ 테스트 부분(예시)

```
def evaluate(model, loader):  
    model.eval()
```

직접 구현



6. 참고 코드

- ✓ 학습/검증/테스트 정확도 비교를 위한 bar plot (예시)

