

팀프로젝트 결과 보고서 #1

제 목: ReNet5 인공지능망 구성 및 분류기
구현

과 목 명: 딥러닝프로그래밍및실습

학 부: AI융합학부

학 번: 20193125

이 름: 김 건 수

제 출 일: 2024년 4월 26일(금)

담당교수: 한 영 준

팀 프로젝트 결과 보고서 #1

ReNet5 인공신경망 구성 및 분류기 구현

정확도를 높이기 위해 하이퍼 파라미터 튜닝

- batch size, learning rate, epoch를 바꿔가면서 여러번 실험함
- 목적함수와 옵티마이저를 바꿔가면서 정확도에 영향을 미치는지 실험함

CrossEntropy & SGD

| ● accuracy (%) | ≡ batch | ≡ learning rate | ≡ epoch | 👤 사람 | 🖨 device |
|----------------|---------|-----------------|---------|-------|----------|
| 99.28 | 64 | 1e-1 | 100 | 👤 김태균 | cpu |
| 99.26 | 32 | 1e-1 | 200 | 🕒 이강룡 | cpu |
| 99.23 | 32 | 1e-1 | 100 | 👤 김태균 | cpu |
| 99.16 | 64 | 1e-1 | 200 | 🕒 이강룡 | cpu |
| 99.12 | 32 | 1e-2 | 100 | 👤 김태균 | cpu |
| 99.06 | 128 | 1e-1 | 100 | 👤 김태균 | cpu |
| 98.98 | 256 | 1e-1 | 200 | 🕒 이강룡 | cpu |
| 98.94 | 64 | 1e-2 | 200 | 👤 김태균 | cpu |
| 98.89 | 32 | 1e-2 | 200 | 🕒 이강룡 | cpu |
| 98.87 | 128 | 1e-2 | 200 | 🕒 이강룡 | cpu |
| 98.84 | 256 | 1e-2 | 200 | 🕒 이강룡 | cpu |
| 98.84 | 64 | 1e-2 | 100 | 👤 김태균 | cpu |
| 98.62 | 256 | 1e-2 | 100 | 🐶 김건수 | mps |
| 98.56 | 128 | 1e-2 | 100 | 👤 김태균 | cpu |
| 98.49 | 16 | 1e-1 | 100 | 👤 김태균 | cpu |
| 98.41 | 128 | 1e-1 | 200 | 🕒 이강룡 | cpu |
| 98.37 | 32 | 1e-3 | 100 | 👤 김태균 | cpu |
| 98.33 | 256 | 1e-2 | 100 | 👤 김태균 | cpu |
| 97.96 | 64 | 1e-3 | 100 | 👤 김태균 | cpu |
| 95.27 | 32 | 0.001 | 15 | 👤 김민규 | cpu |

| ● accuracy (%) | ≡ batch | ≡ learning rate | ≡ epoch | 👤 사람 | 🖨 device |
|----------------|---------|-----------------|---------|-------|------------|
| 90.33 | 32 | 1e-4 | 100 | 👤 김태균 | <u>cpu</u> |
| 81.16 | 64 | 1e-4 | 100 | 👤 김태균 | <u>cpu</u> |

- logsigmoid 함수를 썼을 때에는 NLLoss 목적함수가 효과적인 결과를 보였다고 해서 적용해봄.

NLLoss & Adam

| ● accuracy (%) | ≡ batch | ≡ learning rate | ≡ epoch | 👤 사람 | 🖨 device |
|----------------|---------|-----------------|---------|-------|------------|
| 99.16 | 64 | 0.001 | 100 | 🐶 김건수 | <u>mps</u> |
| 99.10 | 256 | 0.001 | 100 | 🐶 김건수 | <u>mps</u> |
| 99.05 | 128 | 0.001 | 100 | 🐶 김건수 | <u>mps</u> |
| 99 | 32 | 0.001 | 15 | 👤 김민규 | <u>cpu</u> |
| 98.78 | 256 | 0.001 | 200 | 🐶 김건수 | <u>mps</u> |
| 98.67 | 32 | 0.001 | 20 | 👤 김민규 | <u>cpu</u> |
| 97.16 | 32 | 0.01 | 15 | 👤 김민규 | <u>cpu</u> |
| | | | | | 제목 없음 |

- crossentropy, SGD
 - 학습률이 0.1에 가까울수록 정확도 올라감
 - 학습률 0.1과 0.01로만 설정하는게 좋아보임
 - CPU, 8GB, batch=32, 64
 - 1 epoch에 약 11초, 100 epoch에 약 18분 소요
 - batch 사이즈는 작을수록 좋아보임
- 24.04.18 활성화 함수 : ReLU → Sigmoid
 - batch:64, epoch:100, lr:1e-1
 - accuracy : 98.68%

결론적으로 아래 하이퍼 파라미터들이 가장 높은 정확도를 보인다.

목적함수 : CrossEntropy

옵티마이저: SGD

batch size: 64

learning rate: 1e-1

epoch: 100 (100 이상으로 가도 변화가 없다.)

정확도를 높일 수 있는 방안

```
# hyper parameter optimize
# logsoftmax에 맞는 목적함수와 옵티마이저 수정
loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=1e-1)
```

목적함수와 옵티마이저 수정 및 learning rate 조정

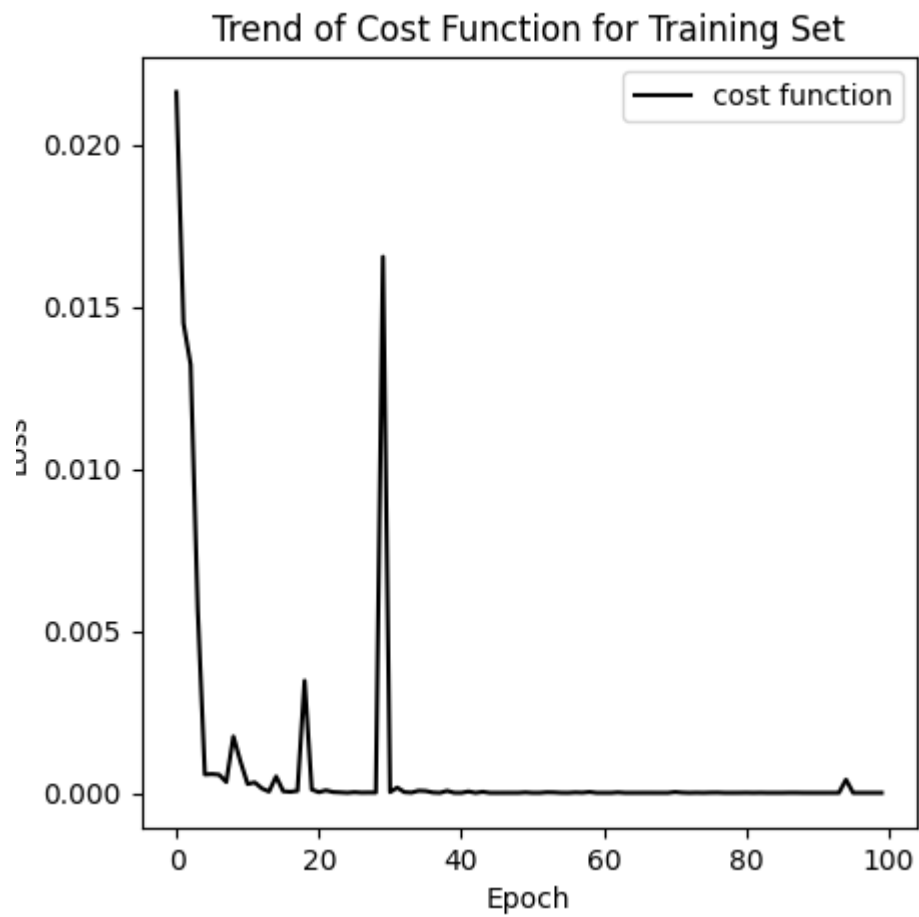
```
# batch size
batch_size = 64
```

batch size 조정

```
epochs = 100
```

epoch 조정

Cost function 추세 그래프



정확하게 분류한 10개 샘플 그림들



잘못 분류한 샘플들

