

[인공지능: 한글 인식 과제 2 보고서]

임베디드시스템공학과 202201658 박소윤

※ 이 보고서는 최종 결과 중심으로 작성하였습니다. 자세한 실험 상세 내용은 프로젝트 보고서에서 작성되어 있습니다. ※

1. 전체 동작 요약(소스 구조) – ipynb 참고

- 1) 한글 흑백 이미지 제작
- 2) 각 글자에 따른 원핫인코딩 실행
- 3) 순전파 및 역전파 수행
- 4) 최적의 가중치 파일 전처리
- 5) 평가
- 6) 반복

2. 동작 상세 설명

115글자: ['충', '청', '남', '도', '홍', '성', '군', '읍', '기', '길', '숙', '사', '호', '동', '박', '소', '윤', '김', '은', '옥', '영', '수', '종', '헌', '조', '인', '의', '경', '봉', '화', '천', '광', '역', '시', '연', '구', '아', '카', '데', '미', '로', '춘', '이', '는', '듣', '만', '하', '여', '가', '슴', '설', '레', '말', '다', '너', '두', '손', '을', '에', '대', '고', '물', '방', '갈', '심', '장', '들', '어', '보', '라', '피', '끓', '뛰', '노', '거', '선', '관', '힘', '있', '것', '류', '를', '꾸', '며', '내', '려', '온', '력', '바', '투', '명', '되', '얼', '음', '과', '으', '지', '혜', '날', '우', '나', '갑', '속', '든', '칼', '니', '더', '면', '간', '마', '쓸', '랴', '싸', '죽', '뿐']

1) 한글 흑백 이미지 제작

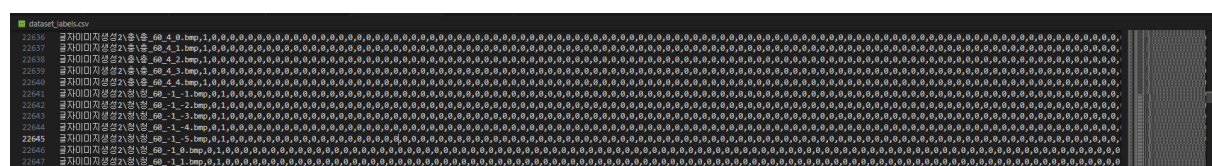
한글 흑백 이미지는 코드로 구현되었습니다. 학습하고자 하는 한글의 중복을 방지하기 위해 중복되지 않는 115개의 글자를 리스트로 생성하였습니다. 각 한글 이미지는 64x64 픽셀 크기로, 폰트 크기는 60으로 설정하여 흑백 BMP 이미지 파일로 저장되었습니다. BMP 파일은 0과 1로 이루어져 있으며, 각 픽셀은 1비트로 표현됩니다. 또한, 한글 폰트 경로를 기준으로 다양한 이미지를 생성할 수 있도록 하였습니다.

중요한 점은, 글자의 픽셀을 고정한 채 학습을 진행한 것이 아니라, 픽셀의 위치를 좌우

또한, 평가를 하기 위해서 새로운 폰트는 고정 픽셀 이미지로 각 글자 폴더에 한 장씩 저장하도록 했습니다.



학습할 한글 리스트를 원핫인코딩 방식으로 라벨링하였습니다. 이 작업은 순전파 및 역전파를 통해 목표값과 출력값을 비교하여 손실 함수를 계산하기 위해 필요합니다. 각 글자는 115개의 폴더에 저장되며, 폴더의 이름에 맞게 원핫인코딩을 적용하였습니다. 위의 115글자 순서대로 라벨링 했습니다. 만약 기존 데이터에 파일이 없다면 새로 생성하고, 기존 데이터에 새로운 데이터를 추가할 수 있도록 하였습니다. 이를 통해 새로운 폰트를 추가하여 추가 학습을 진행할 수 있습니다.



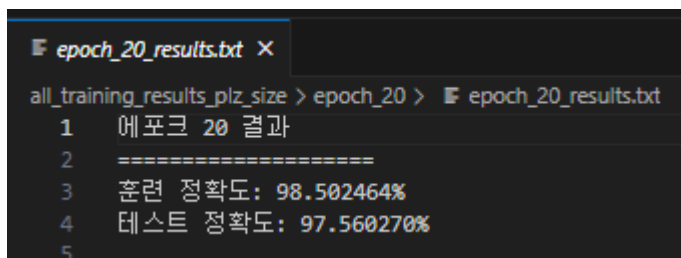
[illegible]

순전파를 통해 계산된 에러로 역전파를 수행합니다. 최적의 실험 방법을 사용해 여러 에폭에 걸쳐 학습을 진행하며, 에포크마다 각 레이어의 출력값, 목표값, 오차, 최적의 가중치 등 다양한 정보를 저장한 파일을 생성합니다. 또한, 스레드를 활용하여 학습 중 터미널에서 경과 시간, 현재 에포크 수, 현재 배치, 남은 배치 등의 정보를 실시간으로 확인

할 수 있습니다.



- 에포크마다 각 레이어의 출력값, 목표값, 오차, 최적의 가중치 등 다양한 정보를 저장한 파일 사진 -



- 약 19만장 데이터를 학습한 경우 최적의 훈련 정확도와 테스트 정확도 사진 -

※ 최적의 실험 방법과 실험에 대한 상세한 내용은 프로젝트 보고서에 작성되어 있습니다.

4) 최적의 가중치 파일 전처리

순전파와 역전파를 통해 계산된 가중치가 어느 지점에서 수렴할 때, 훈련 정확도와 테스트 정확도가 가장 높은 가중치를 '최적의 가중치'로 정의하였습니다. 이 파일에는 가중치 뿐만 아니라 레이어 출력값, 목표값 등 다양한 정보가 포함되어 있기 때문에 필요 없는 데이터를 삭제해야 했습니다. 또한, 가중치는 float64 형식으로 저장되어 있어 이를 float 형식으로 변환하여 최종 파일에 저장했습니다.

```

# processed_weights_with_labels.txt X
all_training_results_pickle > # processed_weights_with_labels.txt
1 Layer 1 Weights:
2 -0.0157 0.0563 0.029 0.0123 -0.043 -0.043 -0.0552 0.0458 0.0126 0.026 -0.0599 0.0587 0.0416 -0.036 -0.0398 -0.0396 -0.0245 0.0031 -0.0085 -0.0261 0.014 -0.0451 -0.026 -0.0167 -0.0055 0.0356 -0.0375 0.0018 0.0116 -0.0567 0.0134 -0.0412
-0.0527 0.0578 0.0599 0.0403 -0.0227 -0.0406 0.0247 -0.0058 -0.0455 0.0011 -0.0565 0.0529 -0.0204 0.022 -0.0218 0.0042 0.0075 -0.0377 0.0604 0.0361 0.0566 0.0511 0.0139 0.0544 -0.0497 -0.0363 -0.0551 -0.0201 -0.0122 -0.0269 0.0428 -0.
0.0162 -0.0253 0.0072 -0.0426 0.0403 -0.051 0.0627 0.0367 -0.0341 -0.0577 0.0447 0.0319 0.0339 0.0388 -0.0409 -0.0137 -0.0438 0.0404 0.0191 -0.0196 -0.0537 -0.0223 -0.0203 0.0303 0.0183 0.0409 -0.0019 -0.0467 0.0278 0.0331 0.0084 0.0348
0.0013 0.0002 -0.0007 -0.0502 -0.0468 -0.0505 0.0192 -0.0227 -0.0003 0.0403 -0.0344 -0.0139 0.0201 -0.0353 -0.0530 -0.0258 -0.0423 0.0548 0.030 0.0167 0.0447 0.0356 -0.0409 0.0474 0.0034 0.0306 0.0506 -0.0209 -0.047 -0.0327 -0.007 0.
0.0408 0.0462 -0.0595 0.0032 -0.0083 -0.0327 -0.0457 -0.0181 0.0578 -0.0191 0.0062 0.031 -0.0116 0.0634 0.0611 -0.0208 0.0026 -0.0226 -0.0253 -0.0562 0.0141 0.0007 -0.0556 -0.0274 0.0512 -0.0326 -0.0443 -0.0007 0.061 -0.0316 0.0222 0.
0.0339 -0.0314 0.0312 -0.0136 0.0194 0.0196 0.0071 -0.0406 0.0451 -0.0213 -0.0407 -0.0599 0.0071 0.0175 -0.0064 -0.0026 -0.0379 0.0161 -0.0422 0.0235 -0.0166 0.0512 -0.0403 -0.0231 -0.051 0.0511 0.0453 -0.0296 0.0214 0.0418 0.0087 0.0051
-0.0302 -0.0491 0.0508 0.052 0.0192 -0.0177 -0.017 0.0305 0.0521 0.0511 0.0382 0.0212 -0.0405 -0.0369 0.0542 0.0168 -0.058 -0.0473 0.0227 -0.0507 -0.0407 0.0069 0.0243 0.0196 -0.0341 0.0264 -0.0332 -0.0274 0.0306 0.018 0.0435 0.0194 0.
0.0083 -0.0508 -0.0152 -0.0272 -0.0303 0.0024 -0.0104 0.052 0.0205 0.0377 -0.0024 0.005 -0.0003 -0.0427 0.0222 -0.0326 -0.0646 0.0136 -0.0443 0.0511 0.0513 0.0401 -0.0221 -0.0601 0.0401 -0.013 0.0352 0.0276 0.0440 -0.0241 -0.0121 0.0455
-0.0209 -0.0396 0.0082 0.0564 0.0266 0.0109 -0.0403 0.0161 0.0637 -0.0422 0.0059 0.052 0.0368 0.03 0.0299 -0.0136 -0.0226 0.0412 0.0421 0.05 0.0503 0.004 0.0022 0.0308 0.0212 0.0275 0.0309 0.0511 -0.0179 -0.0134 -0.0402 0.0123 -0.056
-0.0014 0.0092 -0.0229 0.0156 -0.0531 -0.0516 0.047 -0.0112 -0.0437 0.0031 0.0318 -0.0398 0.0126 -0.0538 -0.0578 0.0016 0.0041 0.0157 0.0271 0.0557 -0.0041 -0.0274 0.0316 -0.0325 -0.0103 -0.0543 -0.0596 0.0593 0.0436 0.0268 -0.0096 -0.
0.009 -0.0414 -0.0304 0.0076 0.0209 0.0215 -0.0207 0.0576 0.0311 0.0096 0.0173 -0.0062 -0.0259 -0.0127 0.037 -0.0566 -0.045 -0.0525 -0.0534 0.0491 0.0316 0.0006 -0.0405 0.002 -0.0011 -0.039 -0.0064 -0.0102 0.0165 0.0192 -0.0536 -0.0121
0.0165 0.0042 0.0404 0.0244 -0.0373 -0.0404 0.0237 -0.0520 0.0168 0.0569 0.0003 -0.0168 0.0133 -0.0001 0.0036 0.0537 -0.0107 0.0508 0.0526 -0.036 -0.0547 -0.054 -0.0049 -0.0551 0.0203 -0.0554 -0.0202 0.043 -0.0577 0.0411 -0.0151 -0.
0.046 0.0265 0.0177 0.0408 0.0309 0.04 -0.0258 -0.0397 0.0319 0.0395 0.0627 -0.0083 -0.0128 0.04 -0.0154 0.0504 0.0406 -0.006 0.0344 0.0358 -0.0449 0.0556 0.004 0.0042 -0.0201 0.0525 -0.0119 -0.0502 0.0542 -0.0479 -0.0197 0.0504 0.0602
0.0122 0.0204 -0.0011 -0.0207 -0.0165 0.0277 0.0306 0.0432 0.0416 -0.0492 -0.0018 -0.0504 0.0015 -0.0108 0.0451 -0.0212 -0.0492 -0.0438 0.0352 0.0171 -0.0516 -0.0562 0.0182 -0.0595 0.0355 0.0236 -0.0529 -0.051 0.0635 -0.013 -0.0131 0.
0.0413 0.0506 0.063 0.0335 -0.0142 -0.0511 0.0356 0.0076 -0.0095 0.0511 -0.0403 0.0001 -0.0592 0.0009 -0.0504 -0.0439 -0.0439 0.0211 0.0343 0.015 0.0631 -0.0098 -0.0236 0.0402 -0.0326 0.0591 -0.0599 0.0003 -0.055 0.0513 0.0001 0.0654 -0.
95 0.0092 0.0019 0.0002 0.021 0.0206 0.0007 0.0232 0.017 0.0561 -0.0535 -0.0272 0.0533 0.0443 -0.0004 0.0110 -0.0290 -0.0304 -0.0025 -0.015 0.0143 -0.0545 0.0542 0.0523 0.0171 0.0004 -0.0257 0.0307 0.035 -0.0370 0.0540 0.0437 0.0504 0.
0.01 0.0163 -0.0009 0.0542 0.0406 -0.056 -0.0598 -0.0163 0.0378 0.0004 -0.045 0.0124 -0.0139 0.0611 0.0431 0.0429 -0.0047 -0.0098 -0.0278 -0.0546 0.0475 0.0305 0.0612 0.0612 0.0052 0.0309 0.0538 0.0427 -0.0303 -0.0056 -0.0448 0.0578 0.
0.0139 -0.0321 0.0246 0.0175 -0.0149 -0.0443 0.0247 0.0006 0.0361 0.0022 0.0417 -0.0005 -0.0022 0.0392 -0.0174 -0.0409 -0.062 0.0306 0.0154 0.025 -0.0401 -0.0545 -0.0735 -0.0206 0.0037 -0.0194 -0.0108 0.0514 -0.0154 0.0041 0.0308 -0.
0.0111 0.0178 0.0474 0.0573 -0.0442 0.0528 -0.0011 -0.031 -0.0004 0.0501 -0.0041 -0.0253 0.015 -0.0341 -0.055 -0.0493 -0.0493 -0.0473 -0.0471 0.0166 -0.0397 -0.0106 0.0408 -0.0041 0.0193 -0.0424 -0.0399 -0.0592 -0.0431 0.0208 -0.0413
-0.0511 -0.0472 -0.0005 -0.0364 -0.016 0.002 0.0255 -0.0546 0.0405 0.0163 -0.0500 0.0440 0.0403 -0.0631 -0.0302 0.0305 0.0253 -0.0432 -0.0309 -0.0153 0.0000 0.016 -0.0104 -0.0135 0.0179 -0.0070 -0.0375 0.0210 0.0401 0.0021 0.0070 -0.
0.0467 -0.0036 0.006 -0.0292 -0.0262 -0.0142 -0.0008 -0.0239 -0.0381 -0.0237 -0.0502 0.0456 0.0066 0.0165 0.0314 -0.0023 -0.0520 0.0016 0.0003 0.0271 -0.011 -0.0470 -0.0273 -0.0177 0.0177 0.0005 -0.0172 0.0007 0.0128 -0.0328 -0.0501 -0.
0.042 -0.0317 -0.0415 -0.0304 -0.0269 -0.0409 0.0503 -0.0518 0.0038 -0.0004 0.0631 -0.0462 -0.0107 0.0572 0.041 0.0319 -0.0411 -0.0503 0.0145 0.0408 0.0039 0.0003 -0.0254 0.0343 -0.0445 -0.0316 -0.0225 -0.0096 -0.04 -0.0545 0.0111 -0.

```

- 최적의 가중치를 만들기 위해 전처리 한 후 파일 사진 -

5) 평가

전처리된 최적화된 가중치를 사용하여 순전파를 통해 평가를 진행합니다. 평가에 필요한 파일은 BMP 이미지 파일, 원한인코딩으로 라벨링된 CSV 파일, 최적화된 가중치 파일입니다. CSV 파일을 통해 이미지 경로를 읽어들이고, 순전파를 통해 가장 높은 출력값을 도출하고, 이를 예측된 라벨로 사용합니다.

결과는 두 가지 파일에 저장됩니다:

1. **comparison 파일:** 이미지 경로, 실제 라벨, 목표 인덱스, 예측 라벨, 예측 인덱스, 타겟 벡터, 예측 벡터, 예측 벡터의 최대값, 상위 4개의 예측값 등을 포함합니다.
2. **prediction 파일:** 실제 라벨과 예측 라벨이 일치하는지 여부를 기록합니다.

또한, 터미널에는 평가 결과와 함께 정확도가 출력됩니다.


```
prediction_results.txt
70 실제: 라, 예측: 라 - 예측이 맞았습니다.
71 실제: 피, 예측: 미 - 예측이 틀렸습니다.
72 실제: 꿏, 예측: 꿏 - 예측이 맞았습니다.
73 실제: 뵈, 예측: 뵈 - 예측이 맞았습니다.
74 실제: 노, 예측: 노 - 예측이 맞았습니다.
75 실제: 거, 예측: 거 - 예측이 맞았습니다.
76 실제: 선, 예측: 선 - 예측이 맞았습니다.
77 실제: 관, 예측: 관 - 예측이 맞았습니다.
78 실제: 힘, 예측: 힘 - 예측이 맞았습니다.
79 실제: 있, 예측: 있 - 예측이 맞았습니다.
80 실제: 것, 예측: 것 - 예측이 맞았습니다.
81 실제: 류, 예측: 류 - 예측이 맞았습니다.
82 실제: 를, 예측: 를 - 예측이 맞았습니다.
83 실제: 꾸, 예측: 꾸 - 예측이 맞았습니다.
84 실제: 며, 예측: 며 - 예측이 맞았습니다.
85 실제: 내, 예측: 내 - 예측이 맞았습니다.
86 실제: 려, 예측: 려 - 예측이 맞았습니다.
87 실제: 온, 예측: 온 - 예측이 맞았습니다.
88 실제: 력, 예측: 력 - 예측이 맞았습니다.
89 실제: 바, 예측: 바 - 예측이 맞았습니다.
90 실제: 투, 예측: 들 - 예측이 틀렸습니다.
91 실제: 명, 예측: 명 - 예측이 맞았습니다.
92 실제: 되, 예측: 되 - 예측이 맞았습니다.
93 실제: 얼, 예측: 얼 - 예측이 맞았습니다.
94 실제: 음, 예측: 음 - 예측이 맞았습니다.
95 실제: 과, 예측: 과 - 예측이 맞았습니다.
96 실제: 으, 예측: 으 - 예측이 맞았습니다.
97 실제: 지, 예측: 지 - 예측이 맞았습니다.
98 실제: 해, 예측: 해 - 예측이 맞았습니다.
99 실제: 날, 예측: 날 - 예측이 맞았습니다.
100 실제: 우, 예측: 우 - 예측이 맞았습니다.
101 실제: 나, 예측: 나 - 예측이 맞았습니다.
102 실제: 갑, 예측: 갑 - 예측이 맞았습니다.
103 실제: 속, 예측: 속 - 예측이 맞았습니다.
104 실제: 든, 예측: 든 - 예측이 맞았습니다.
105 실제: 칼, 예측: 칼 - 예측이 맞았습니다.
106 실제: 니, 예측: 니 - 예측이 맞았습니다.
107 실제: 더, 예측: 더 - 예측이 맞았습니다.
108 실제: 먼, 예측: 먼 - 예측이 맞았습니다.
109 실제: 간, 예측: 간 - 예측이 맞았습니다.
110 실제: 마, 예측: 마 - 예측이 맞았습니다.
111 실제: 쓸, 예측: 쓸 - 예측이 맞았습니다.
112 실제: 라, 예측: 라 - 예측이 맞았습니다.
113 실제: 싸, 예측: 바 - 예측이 틀렸습니다.
114 실제: 족, 예측: 족 - 예측이 맞았습니다.
115 실제: 뿐, 예측: 뿐 - 예측이 맞았습니다.
...

문제 ESP-IDF JUPYTER 포트 출력 디버그 콘솔
> 터미널
AAAAAAAAAAAA
UnicodeDecodeError: 'euc_kr' codec can't decode byte 0xa9 in position 13: illegal multibyte sequence
C:\AI>C:/Users/asx12/AppData/Local/Programs/Python/Python311/python.exe c:/AI/평가_ALL.py

평가 정확도: 93.913043%
```

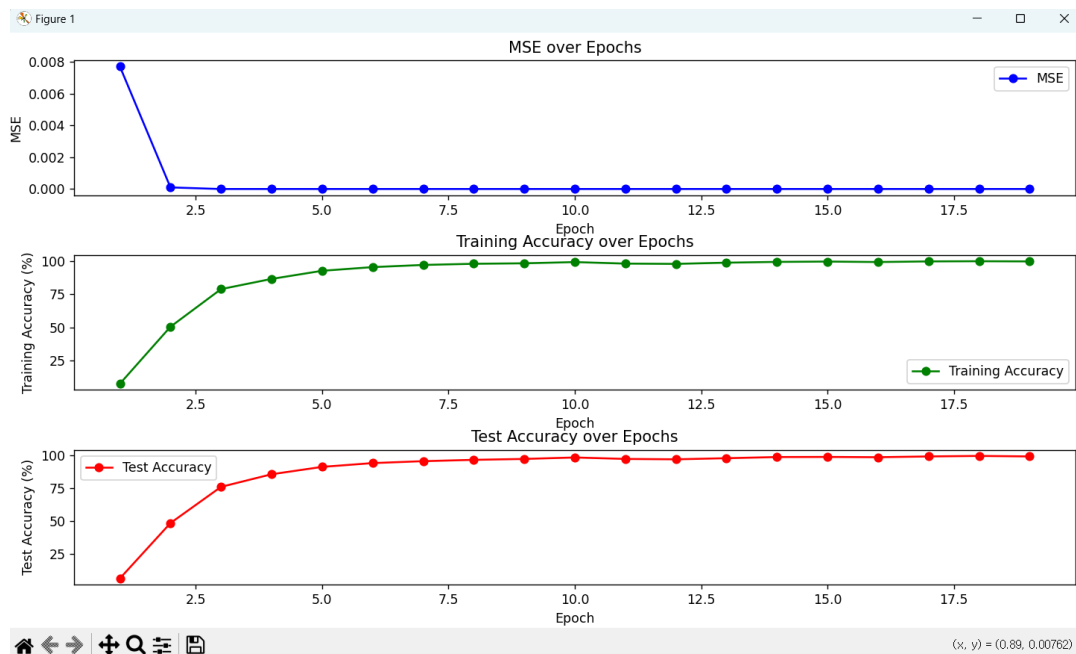
- prediction 파일 내용 및 터미널 평가 정확도 예시 사진 -

3. 동작 결과

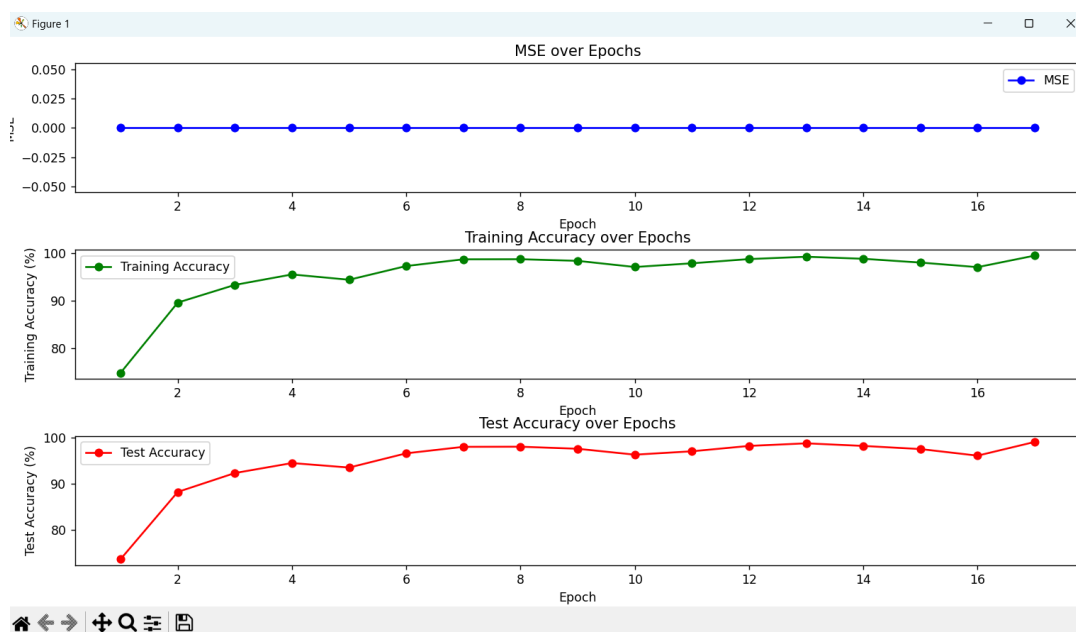
위의 2번 동작 상세 설명 - (1)~(5)의 과정을 폰트 1개부터 시작해서 15개까지 반복합니다. 총 그렇게 데이터셋을 점점 늘리며 약 189980장 이미지로 학습을 시키고 새롭게 가중치를 업데이트 했습니다.

1) 폰트 개수에 따른 학습 결과 비교

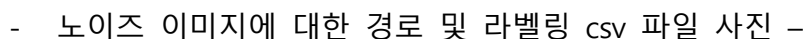
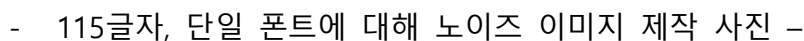
[폰트 1개 학습 결과 - 에폭 당 MSE 및 정확도]



[폰트 15개 학습 결과 - 에폭 당 MSE 및 정확도]



한 폰트에 대하여 세로, 가로 (한 줄, 여러 줄), 대각선 등 선에 대해 노이즈를 부여한 코드를 제작하였습니다. 노이즈에 대해 노이즈 데이터셋을 추가해서 학습하지 않은 상태입니다. 한 글자 당 여러 노이즈를 추가하여 805장에 대한 평가를 진행했습니다.



4. 결론

여러 폰트를 사용해 학습을 진행한 결과, 훈련 정확도와 테스트 정확도가 비슷하게 높은 양상을 보였습니다. 현재 약 19만 장의 데이터를 학습한 상태에서 훈련 정확도와 테스트 정확도는 모두 약 98%에 도달했습니다. 노이즈 이미지를 포함하지 않고 학습한 경우에도 한 폰트에 대해 다양한 노이즈를 추가하여 평가한 결과, 96.52%의 높은 정확도를 얻을 수 있었습니다.

다만, 새로운 폰트로 평가했을 때 두 가지 경우에 대해 테스트를 진행한 결과, 서로 다른 정확도를 보였습니다. 하나의 평가에서는 55.65%, 다른 평가에서는 93.91%로 차이가 있었습니다. 이러한 결과를 바탕으로 추가적인 학습과 비교, 분석이 필요하다고 생각하며, 프로젝트를 완성하기 위해 더 깊은 연구를 진행할 계획입니다.