

과제 #2

임 동혁

Knowledge and Data Engineering Lab.
Computer Engineering

과제 설명

- 제출일 : 6월 29일 12:00 구글 클래스 제출
- 제출 방법 : 주피터 소스 파일로 학번_이름.ipynb로 제출
(어길 시 감점 -1: 파일 제목, 형식 등)
- 점수 : 25점 (전체 점수에서 25점)
- 수업 시간에 다룬 MNIST 문제를 확장하여 문자
MNIST(EMNIST)로 구현
- 과제 문의는 조교에게 할 것
(박준희, 20205223@vision.hoseo.edu)

문제

- MNIST 대신 EMNIST 'balanced' 사용

```
from emnist import extract_training_samples
train_images, train_labels = extract_training_samples('balanced')
from emnist import extract_test_samples
test_images, test_labels = extract_test_samples('balanced')
```

EMNIST 데이터 셋

Dataset Summary

There are six different splits provided in this dataset. A short summary of the dataset is provided below:

- EMNIST ByClass: 814,255 characters. 62 unbalanced classes.
- EMNIST ByMerge: 814,255 characters. 47 unbalanced classes.
- EMNIST Balanced: 131,600 characters. 47 balanced classes.
- EMNIST Letters: 145,600 characters. 26 balanced classes.
- EMNIST Digits: 280,000 characters. 10 balanced classes.
- EMNIST MNIST: 70,000 characters. 10 balanced classes.

데이터 설명

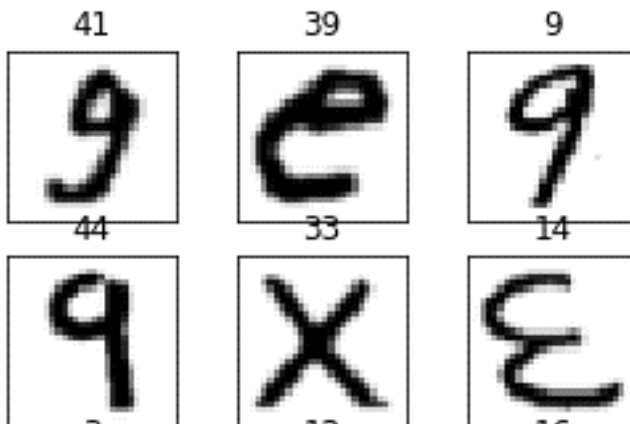
- train_labels, test_labels

label 0-9 : 0-9 (숫자 이미지)

label 10-35 : a-z (알파벳 이미지)

label 36-47 : 대문자, 소문자가 다른 알파벳 이미지

ex)



데이터 처리 관련 사항

- Train_images, test_images는 배열로 변경 후 reshape() 함수등을 이용하여

shape(112800, 784) : 112800은 training 데이터 수

shape(18800, 784) : 18800은 testing 데이터 수

로 변경해야 함

```
train_images.shape
```

```
(112800, 28, 28)
```

Figure 1 image shape 변경 전

```
train_images.shape
```

```
(112800, 784)
```

Figure 2. image shape 변경 후

One-hot encoding 변환

- train_labels, test_labels은 배열로 변경 후 one-hot encoding을 위한 np.eye() 함수 등을 이용하여 shape (112800, 47), (18800, 47)로 변경

```
test_labels = np.eye(47)[test_labels]
```

ex) [0,0,0,0,1,0,0,.....,0,0,0] -> 5 (숫자 이미지)

[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,.....0] -> a (알파벳 이미지)

↵ |

```
test_labels.shape
```

```
(18800,)
```

Figure 3. label shape 변경 전 ↵

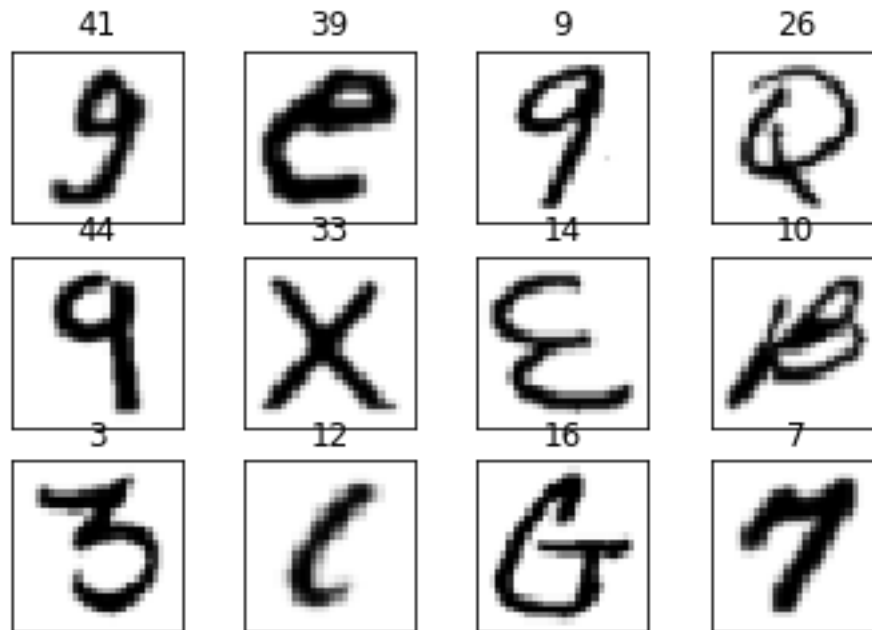
```
test_labels.shape
```

```
(18800, 47)
```

Figure 4. label shape 변경 후 ↵

구현 사항

- 위의 사항대로 변경한 `tain_images`, `train_labels`, `test_images`, `test_labels`를 이용하여 기존 dropout 코드에 적용 후 모델 구성 및 학습
- Matplot 확인 (12개의 테스트 이미지 출력)



평가 방법

<실행 예>

```
is_correct = tf.equal(tf.argmax(model, 1), tf.argmax(Y, 1))  
accuracy = tf.reduce_mean(tf.cast(is_correct, tf.float32))  
print('정확도:', sess.run(accuracy, feed_dict={X: test_images, Y:test_labels, keep_prob: 1}))
```

- 정확도 85%이상 25점, 90% 이상 가산점 3점