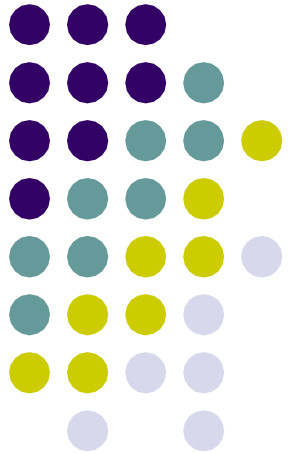
Introduction to Artificial Intelligence

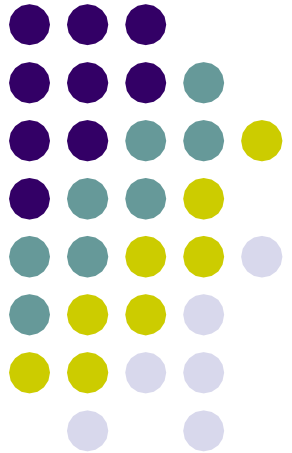
Prof. Inkyu Moon

Dept. of Robotics & Mechatronics Engineering, DGIST

Dept. of Artificial Intelligence, DGIST



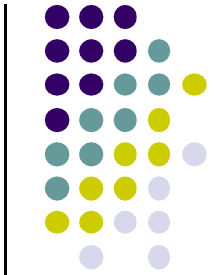
Neural Network Design (Assignments)



1st project
- training: MNIST 60,000개 dataset
- testing: handwriting dataset(10-100개)
- performance: 중요치 않음. (80-90% 사이도 무방.) 성능이 부족해 보인다면 그 이유를 생각해서 report에 기재할 것.
이유 1: 학습 데이터에 handwriting dataset이 없었기 때문. 학습 데이터에 handwriting dataset 일부를 넣어 학습시킨다면 성능이 더욱 향상되었을 것임.
이유 2: ...

2nd project
- training: rotation으로 *3배 증강한 MNIST 60,000개 dataset == total 180,000개
- testing: MNIST 10,000개 dataset (Do Not Use handwriting dataset)
작업환경(labtop 등)의 문제로 시간이 너무 많이 걸리면, training dataset을 60,000개의 일부인, 10,000개 정도로 줄일 것. 이후 10,000개를 증강하여 30,000개로 학습.

#1 Assignment



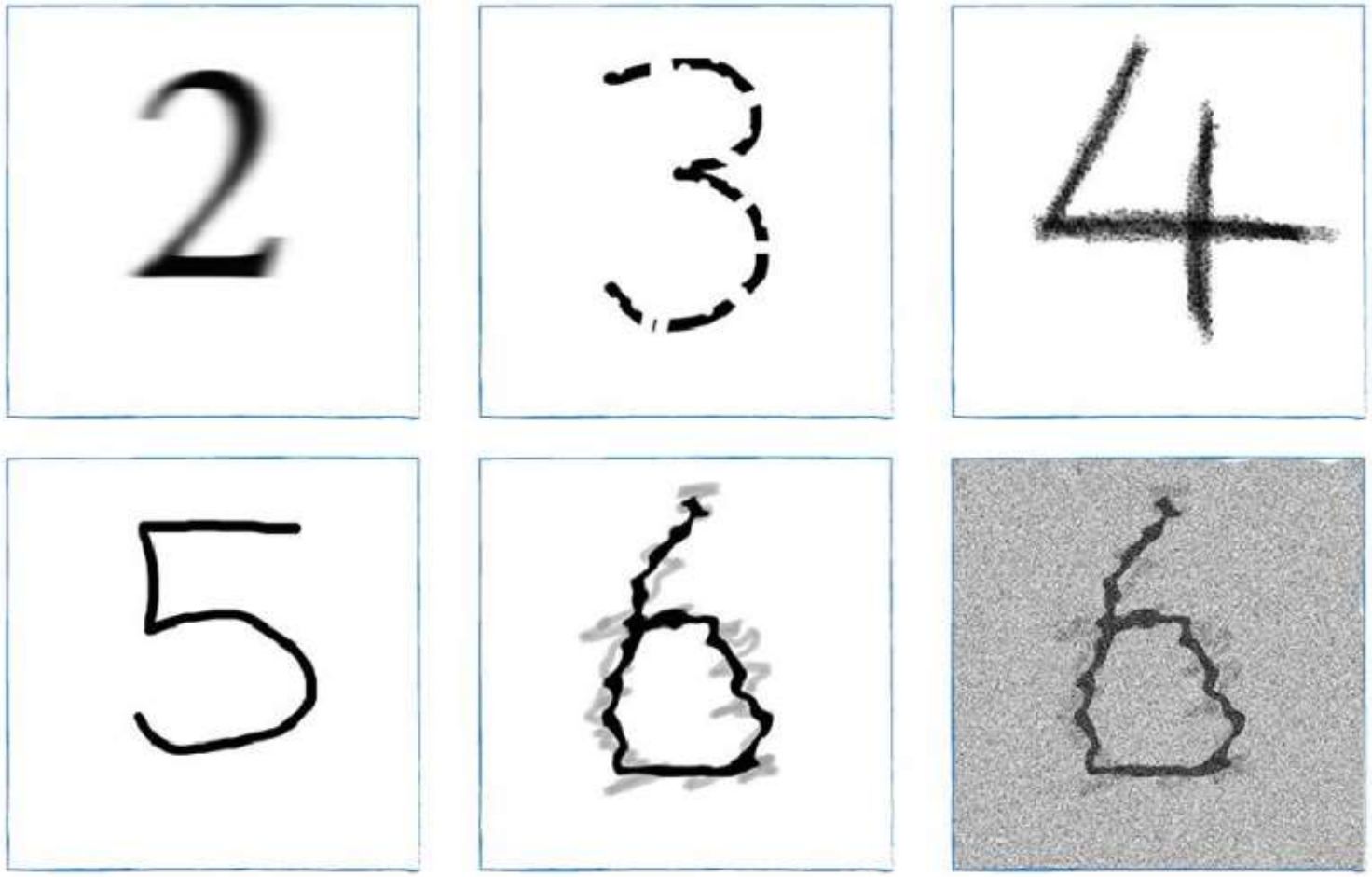
10~100개 사이

Creating New Test Data: Your Own Handwriting data

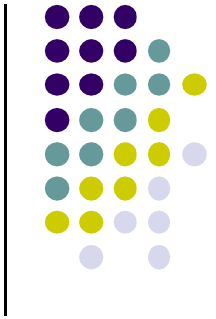
- You try to use different styles of writing, and noisy or shaky images to see how well our neural network copes.
- You can create images using any image editing or painting software.
- You can even use a pen on paper and photograph your writing with a smartphone or camera, or even use a proper scanner.
- The only requirement is that the image is square and you save it as PNG format.

#1 Assignment

- Here are some images I made.



#1 Assignment



MNIST data에 맞게 image size 조정

- We need to create smaller versions of these PNG images rescaled to 28 by 28 pixels, to match what we've used from the MNIST data.
- Python libraries help us with reading and decoding the data from common image file formats, including the PNG format.

image 전처리 과정

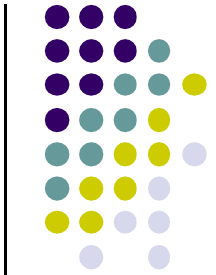
- Look at the following simple code:

```
import scipy.misc
img_array = scipy.misc.imread(image_file_name, flatten=True)

img_data  = 255.0 - img_array.reshape(784)
img_data = (img_data / 255.0 * 0.99) + 0.01
```

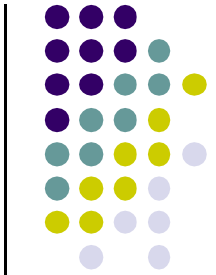
- The scipy.misc.imread() function helps us to get data out of image files such as PNG or JPG files.

#1 Assignment



- The “flatten=True” parameter turns the image into array of floating point numbers, and if the image is colored, the color values will be flattened into grey scale.
- The next line reshapes the array from a 28x28 square into a long list of values.
- The last line rescales the values from 0.01 to 1.0.

#1 Assignment

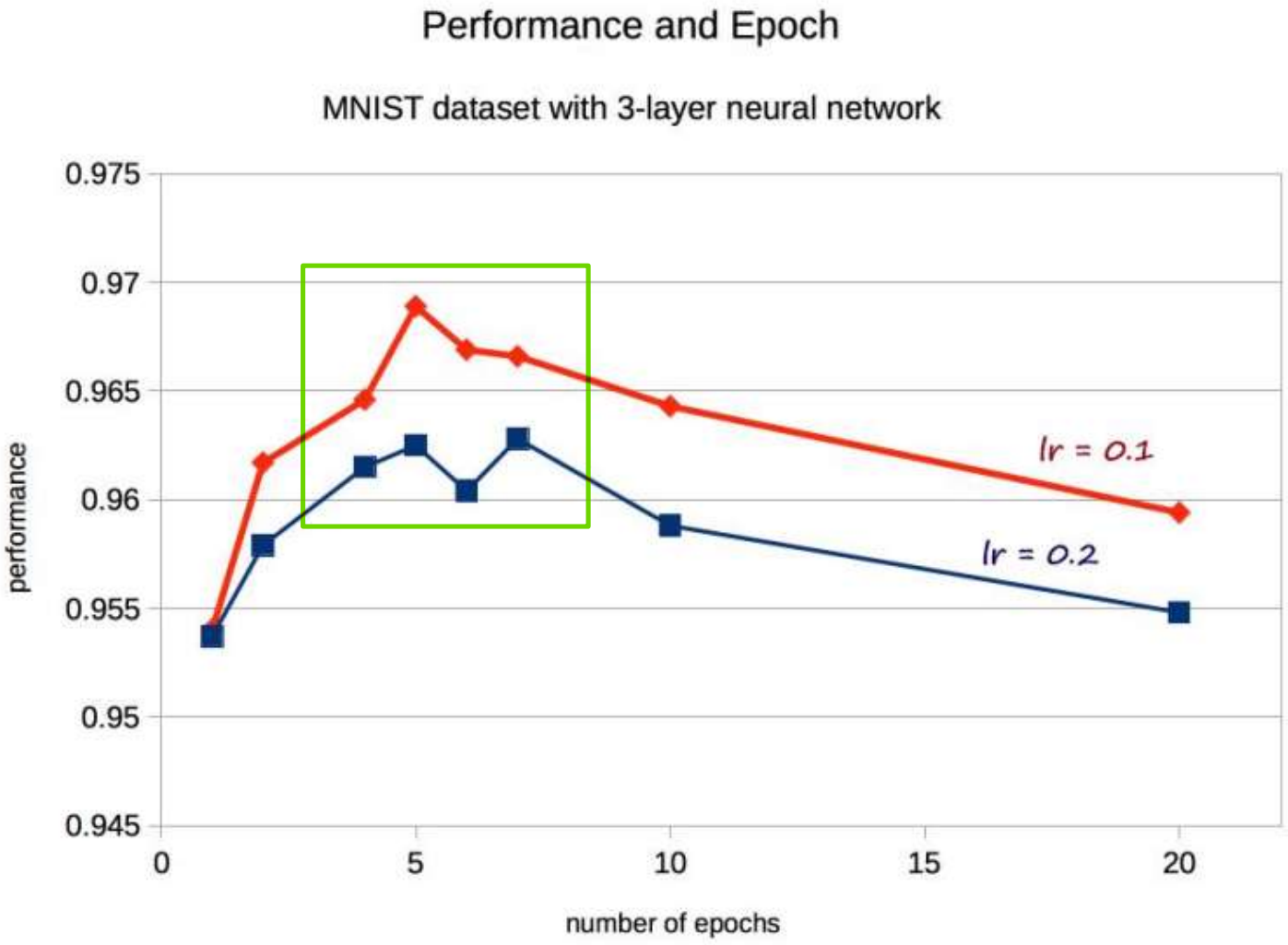


- You can use the developed neural network program with the MNIST training data set.
- However, instead of testing with the MNIST test set, you should test against data created from your own handwritten number images.
- You can set the learning rate to 0.1 and 0.2.
- **Evaluate the neural network performance by varying the total number of epoch.**

1. training: MNIST full-training data set (60,000)
2. testing: "My own handwritten number images"
3. learning은 0.1, 0.2
4. # of epoch: 1, 2, 4, 5, 6, 7, 10, 20
5. NN의 performance(accuracy) 측정: 80-90% 무방. (graph 첨부)

#1 Assignment

- code & 출력 결과
- report: graph 첨부

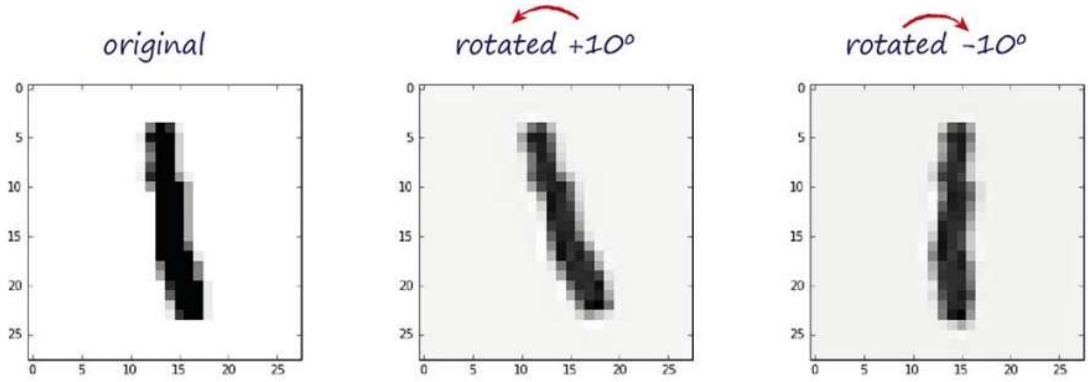


training data augmentation(증강)
Full_training_data: 60,000개 -> rotations를 통해 * 3 = 180,000개 data.
증강된 180,000개의 data로 학습하여 accuracy 계산.

#2 Assignment

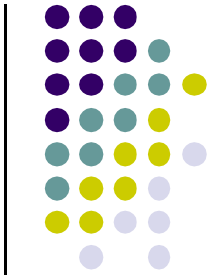
Creating New Training Data: Rotations

- Create new your handwritten number images by rotating them clockwise or anticlockwise, by 10 degrees for example; you can create many more examples with different rotation angles.




```
# create rotated variations
# rotated anticlockwise by 10 degrees
inputs_plus10_img =
scipy.ndimage.interpolation.rotate(scaled_input.reshape(28,28), 10,
cval=0.01, reshape=False)
# rotated clockwise by 10 degrees
inputs_minus10_img =
scipy.ndimage.interpolation.rotate(scaled_input.reshape(28,28), -10,
cval=0.01, reshape=False)
```

#2 Assignment



- You should add the newly created rotated-handwritten number images to the MNIST training data set.
- Then you should test with the MNIST test set.
- The learning rate should be 0.1.
- **You can set the epoch number to 1 and 10.**
- **Evaluate the neural network performance with the various angles of the additional rotated images.**

- learning rate = 0.1
- training: MNIST dataset(rotation해서) == 60,000 * 3 = 180,000
=> augmentation에서 98%정도 accuracy 확인.
- testing: Full-MNIST test dataset(10,000) => Epoch: 5, 10 해볼 것.
- rotation angle: 5, 10, 15, 20, 25도만큼 +-로 회전하여 5번 진행.

#2 Assignment

- code & 출력 결과
- report: graph 첨부