

# IT시스템설계

과제3: 변형된 MNIST Dataset 분류

전자전기공학부

B415156 이승호

## Augmented MNIST Dataset

Rotation, Zoom, Flip이 적용된 Augmented MNIST Dataset을 잘 인식할 수 있도록 네트워크 구조를 변경하거나 학습데이터를 변형함으로써 성능을 향상시키는 방법들을 테스트하였다. 모든 코드는 Google Colab 환경에서 테스트하였고 Keras를 이용해서 인공지능 모델을 만들었다.

### \*Augment Config

Random Rotation Range = 30°

Random Zoom Range = 1/1.3

Horizontal Flip Rate = 0.3

위와 같은 조건으로 데이터를 변형하였다.

### 1) Fully Connected

784(Input)  $\Rightarrow$  784  $\Rightarrow$  392  $\Rightarrow$  10(Output)

Parameter = 927,090

- Original Dataset을 1번 학습

위와 같은 3 Layer의 Fully Connected 모델에서 Original Dataset을 1번만 학습하였다.

### \*Model & Train Config

Hidden Layer Activation = ReLU

Output Layer Activation = Softmax

Optimizer = Adam

Loss = Categorical Cross Entropy

Epoch = 1

Batch Size = 100

### \*Result

	Original Images	Augmented Images
Accuracy	0.9682	0.5798

Original Test Images에 대해서는 Accuracy가 0.95 이상으로 좋은 결과가 나왔으나 Augmented Test Images는 거의 인식하지 못했다.

- Original Dataset, Augmented Dataset을 교대로 학습 동일한 모델에 대해서 먼저 Original Dataset을 1번, 다음에 Augmented Dataset을 1번 교대로 학습하는 방식으로 학습 횟수를 늘려가며 Accuracy의 변화를 테스트하였다.

### \*Model & Train Config

Hidden Layer Activation = ReLU

Output Layer Activation = Softmax

Optimizer = Adam

Loss = Categorical Cross Entropy

Epoch = 1

Batch Size = 100

### \*Result

	Original Images	Augmented Images
1	0.8749	0.9401
2	0.9354	0.9560
3	0.9500	0.9575
4	0.9634	0.9630
5	0.9703	0.9637

1번 교대로 학습하였을 때에는 나중에 학습한 Augmented Image에 편향되어 학습된 모습을 볼 수 있었으나 3번째 학습부터는 Original Image와 Augmented Image 모두 0.95 이상으로 좋은 Accuracy를 확인할 수 있었다.

### 2) Convolutional Neural Network

(28, 28, 1)(Input)  
 $\Rightarrow$  32, (5, 5) Conv  $\Rightarrow$  (2,2) MaxPooling  
 $\Rightarrow$  64, (5, 5) Conv  $\Rightarrow$  (2,2) MaxPooling  
 $\Rightarrow$  Flattening  $\Rightarrow$  1024  $\Rightarrow$  512  
 $\Rightarrow$  10(Output)

Parameter = 3,794,314

- Original Dataset을 1번 학습

위와 같은 2 Conv, 2 Pool, 3 FC 모델에서 Original Dataset을 1번만 학습하였다.

### \*Model & Train Config

Convolution Layer Activation = ReLU

Padding = Same

Hidden Layer Activation = ReLU

Output Layer Activation = Softmax

Optimizer = Adam

Loss = Categorical Cross Entropy

Epoch = 1

Batch Size = 100

### \*Result

	Original Images	Augmented Images
Accuracy	0.9865	0.7067

Full Connected만을 사용해서 학습했을 때보다 학습 효율이 좋았다. Augmented Image를 테스트했을 때도 0.7067로 이전에 비해서 나은 결과를 확인할 수 있었다. 그러나 완전히 학습되었다고 하기에는 부족한 결과이기 때문에 FC와 마찬가지로 Original Dataset과 Augmented Dataset을 번갈아가며 학습하면서 Accuracy의 변화를 측정하기로 했다.

- Original Dataset, Augmented Dataset을 교대로 학습

### \*Model & Train Config

Convolution Layer Activation = ReLU

Padding = Same

Hidden Layer Activation = ReLU

Output Layer Activation = Softmax

Optimizer = Adam  
 Loss = Categorical Cross Entropy  
 Epoch = 1  
 Batch Size = 100

**\*Result**

	Original Images	Augmented Images
1	0.9545	0.9731
2	0.9761	0.9790
3	0.9868	0.9782
4	0.9857	0.9824
5	0.9857	0.9824

FC와 달리 1번의 교대학습에서도 모두 Accuracy가 0.95 이상이 나왔다. 5번의 교대학습 부터는 더이상 학습이 되지 않았다.

**+ 추가 실험**

5번씩 학습된 FC와 CNN Model에 더욱 심하게 변형된 Test Image를 넣어 Accuracy가 얼마나 떨어지는지 확인해보았다.

**\*Augment Config**

Random Rotation Range = 45°  
 Random Zoom Range = 1/1.5  
 Horizontal Flip Rate = 0.5

**\*Result**

	FC	CNN
Accuracy	0.8566	0.9521

**\*Augment Config**

Random Rotation Range = 60°  
 Random Zoom Range = 1/1.7  
 Horizontal Flip Rate = 0.7

**\*Result**

	FC	CNN
Accuracy	0.6114	0.8374

CNN의 경우 더 심하게 변형된 데이터에 관해서도 Accuracy를 어느정도는 유지하나, 결과적으로는 Accuracy가 떨어졌다. 높은 Accuracy는 네트워크 모델의 차이이기도 하나, 변형된 데이터를 학습시켰기 때문인 것 같다.