

ARTIFICIAL INTELLIGENCE

Programming Homework

MLP 구현을 통한 MNIST 숫자 필기체 인식

1. MNIST Dataset

0부터 9까지의 숫자 이미지로 구성

하나의 이미지는 28 x 28(pixel) 크기로, 각 pixel은 0에서 255 사이의 값을 취함

Txt 파일에 각각의 이미지 정보(784개의 pixel 정보와 정답)가 저장
다음 페이지 참조

※ MNIST Dataset은 txt 파일 형식으로 **이루리에 첨부**

※ 70000개의 데이터로 구성

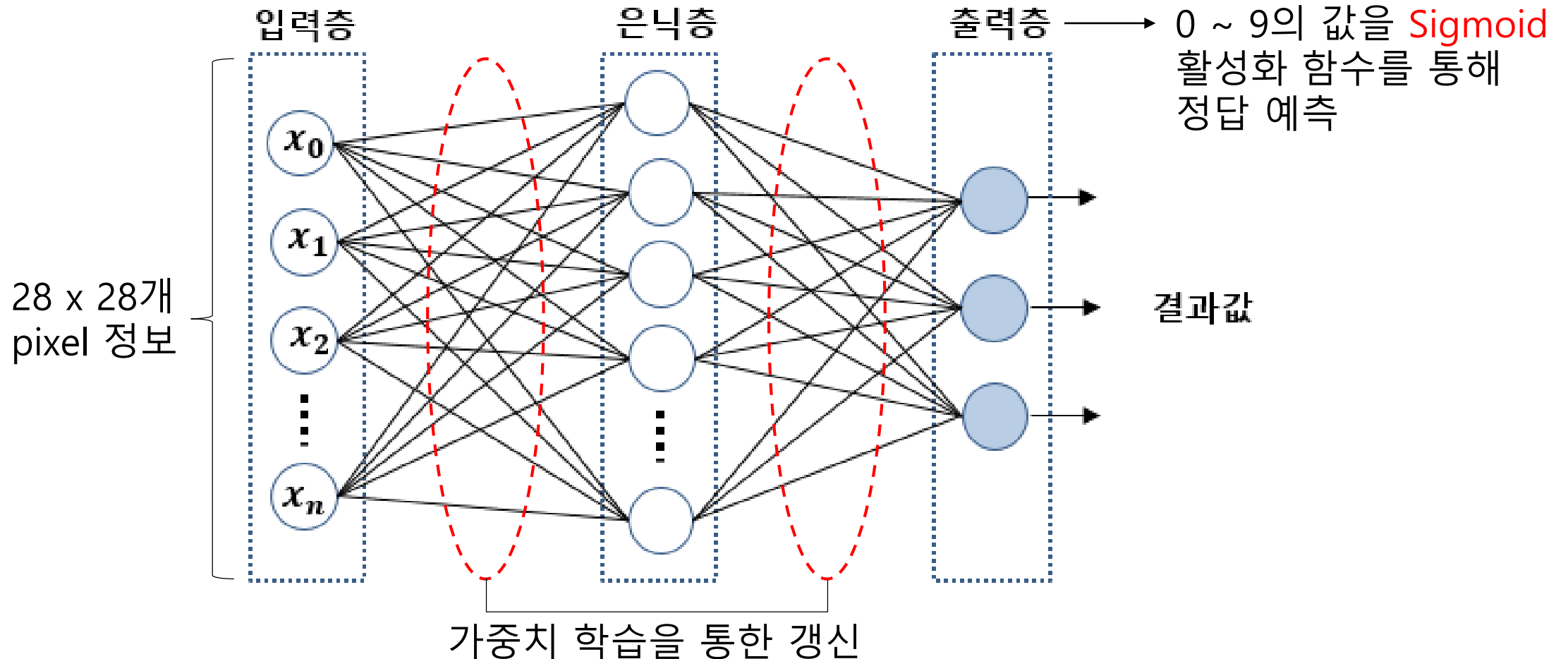
필기 데이터 예제

정답
label

28*28개의
0~255 사
이의 정수

MLP 구현을 통한 MNIST 숫자 필기체 인식

2. MLP(Multi Layer Perceptron)



MLP 구현을 통한 MNIST 숫자 필기체 인식

3. Goal

Training data에서 epoch 진행에 따른 **loss rate** 출력

Test data의 loss rate 출력

※ **loss rate** : 예측 실패 데이터 개수 / 전체 데이터 개수 * 100

※ 70000개의 데이터 중 49000개는 training data, 21000개는 test data로 사용

MLP 구현을 통한 MNIST 숫자 필기체 인식

4. Reference

제한사항 없음 : Hidden layer node, Learning rate

파일 호출 -> 초기화 -> 활성화 -> 가중치 학습

초기화 과정 : hidden layer node와 output layer node의 threshold와 weight 수준을
균등 분포를 따르는 임의의 실수로 지정

활성화 과정 : input, threshold, weight 값으로 sigmoid 활성화 함수 계산
-threshold + $\sum \text{input} * \text{weight}$ ※ Weight 개수만큼 반복

가중치 학습 : Output layer와 hidden layer의 weight 갱신

Output 오차 기울기 : 예측 value * (1 - 예측 value) * (실제 value - 예측 value)

Hidden 오차 기울기 : Output layer의 오차 기울기와 Hidden layer에서 Output layer로 가는
weight의 value의 곱셈 합에 value * (1 - value)를 곱함

MLP 구현을 통한 MNIST 숫자 필기체 인식

5. Warning

Model Structure 필히 작성

(input layer node, hidden layer node, output layer node 개수, Learning rate, epoch)

코드 첨부(**Java or C**)

표절 금지(알고리즘 확인합니다)

MLP 구현을 통한 MNIST 숫자 필기체 인식

5. Sample Code(JAVA)

※ 감이 오지 않는 수강생들을 위한 **샘플 코드**입니다. 대략적인 흐름만 기술한 것으로, 제시한 코드만 완성하는 것이 아니라 추가적으로 더 많은 내용을 기술해야 합니다. 다른 방법으로 코드 작성하셔도 무방합니다.

```
public class MLP {  
    /*  
        global variable list  
        input pattern size, hidden node number, learning rate, training data number,  
        test data number, training repeat number...  
    */  
  
    public static void main(String[] args) throws FileNotFoundException {  
        // file call  
        // initialize variable(hidden layer weight and threshold, output layer weight  
        // and threshold, data array...)
```


MLP 구현을 통한 MNIST 숫자 필기체 인식

5. Sample Code(JAVA)

```
// 1 step : initialize method call(hidden & output layer)

while(hasNext?) {
    /*
        initialize variable(input array, hidden layer value array, output array,
        output layer value array, error gradient array...)
    */

    // One-hot encoding for output
    // 2 step : activation method call(hidden & output layer)
    // 3 step : training weight value
    // observation loss rate

    .
    .
    .
```

MLP 구현을 통한 MNIST 숫자 필기체 인식

5. Sample Code(JAVA)

```
        if(all training data is calculated) {
            // calculate loss rate
            // epoch increase

            if(filled in the number of training iterations) {
                // start the test
                // initialize fail count and use test data fail count
            }
        }
    }

    .
    .
    .
```

MLP 구현을 통한 MNIST 숫자 필기체 인식

5. Sample Code(JAVA)

```
public static void initialize(int[] threshold, int[] weight) {  
    /*  
        Specify the node threshold and weight level as an arbitrary number that  
        follows the even distribution  
    */  
    // repeat  
    double thresholdVal = (Math.random() * (4.8 / weight.length)) - (2.4 /  
                                                                    weight.length);  
    threshold[] = critVal;  
  
    // repeat  
    double weightVal = (Math.random() * (4.8 / weight.length)) - (2.4 /  
                                                                    weight.length);  
    weight[][] = weightVal;  
}
```

MLP 구현을 통한 MNIST 숫자 필기체 인식

5. Sample Code(JAVA)

```
public static void activation([] input, [][] weight, [] threshold, [] result) {  
    // repeat  
    double resultVal = -threshold[];  
  
    // repeat  
    resultVal += input[] * weight[][];  
  
    // Sigmoid activation function  
    result[] = 1 / (1 + Math.exp(-resultVal));  
}  
}
```