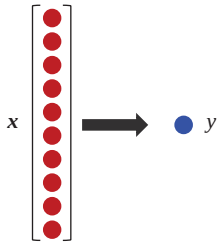
AI and Speech
Homework #2

목표

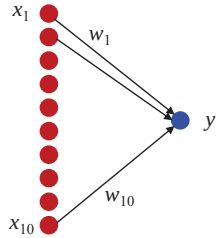
- **Task = regression**
- **System = neural network**
  - 가장 간단한 NN 구조 사용
- **Supervised learning을 C로 구현**
  - Backpropagation에 따라 gradient 계산

Problem Setting

- **Training dataset**
  - Input : 10-D vector of floating-point values
    - $-0.5 \leq x_k \leq 0.5$
  - Output : a floating-point value
    - Example : 10가지 개인 정보 → 신용도 결정
  - 1000 data of $(x_1, x_2, \ldots. x_{10}, y)$ from 1000 people



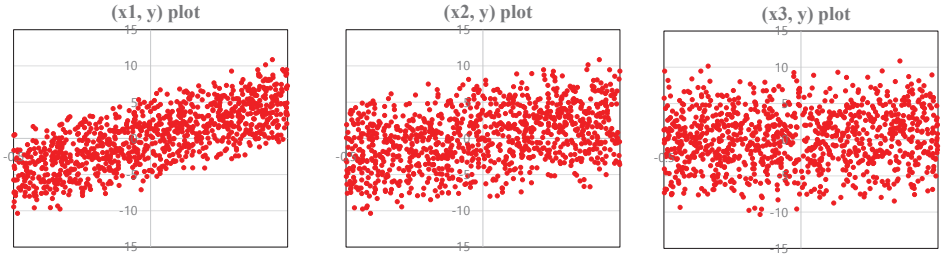
- **Target**
  - $(x_1, x_2, \ldots. x_{10})$ 와 y 사이의 관계 learning



- **Neural network**
  - No hidden layers
  - No bias
  - Linear activation function
  - $y = \sum_{k=1}^{10} w_k x_k$

Training Data 특성

Scatter diagram



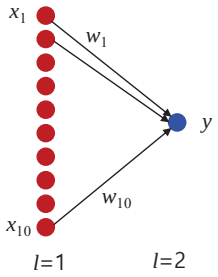
Different input-output dependency

NN Training

- **Training by stochastic gradient descent(SGD) algorithm**
  - Weight initialization
    - Uniform between -0.5 and 0.5
    - C code : w[k] = ((float)rand() / RAND_MAX) - 0.5
  - Batch size = 1
  - 1 epoch
  - $\eta = 0.1$

NN Training

- **Update process**
  - Forward
    - 각 input $\boldsymbol{x}$ 에 대하여 output $y = \boldsymbol{wx}$ 계산
  - Output error
    - $C = \frac{1}{2}(y^* - y)^2$
    - $\delta^{l=2} = \frac{\partial C}{\partial (z=y)} = y - y^*$
  - Gradient
    - $\frac{\partial C}{\partial \boldsymbol{w}^2} = \delta^2(\boldsymbol{a}^1)^T = (y - y^*)\boldsymbol{x}^T$
    - $\frac{\partial C}{\partial w_k} = x_k(y - y^*)$
  - Update
    - $w_k \leftarrow w_k - \eta \frac{\partial C}{\partial w_k} = w_k + \eta x_k(y^* - y)$



$\boldsymbol{z}^l = \boldsymbol{W}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l$

$y = \boldsymbol{wx}$

C code

- **File open, read, write (text file)**

```
float input[DIM] = {0,0,}, target ;
fopen_s(&fin,“trainingDB.dat”, “rt”);
fopen_s(&fo,“output.dat”, “wt”);

for (k = 0; k < DIM; k++)
        fscanf(fin, “%f”, input + k);

 fscanf(fin, “%f”, &target);

for (k = 0; k < DIM; k++)
        fprintf(fo, “%10.7f”, weight[k]);
fprintf(fo, “\n”);
```

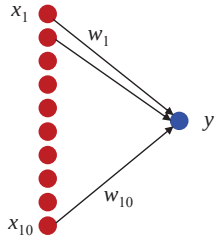

C code

- **전체 구조**

  for (input index)
  - Data read
  - Output computation
  - Gradient computation
  - Weight update
  - Weight and cost write

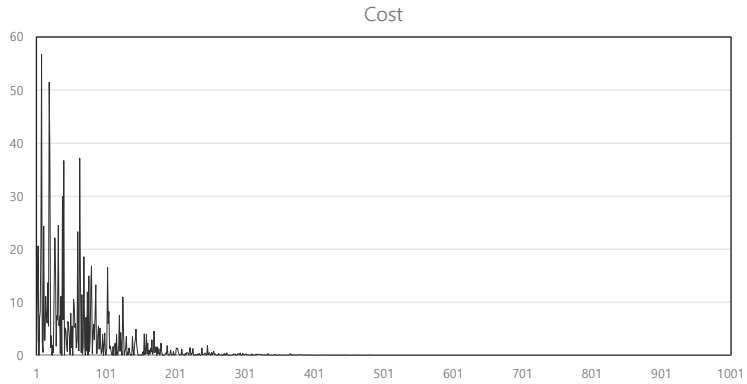
Check-Point

- **500번째 x 에 대한** $C = \frac{1}{2}(y^* - y)^2 \cong 0.00057$
- **1000번째 x 에 대한** $C = \frac{1}{2}(y^* - y)^2 \cong 0.0000012$
- **After 1000 updates, weight의 크기 순서**
  - $w_1 > w_8 > w_2 > \ldots\ldots > w_{10}$



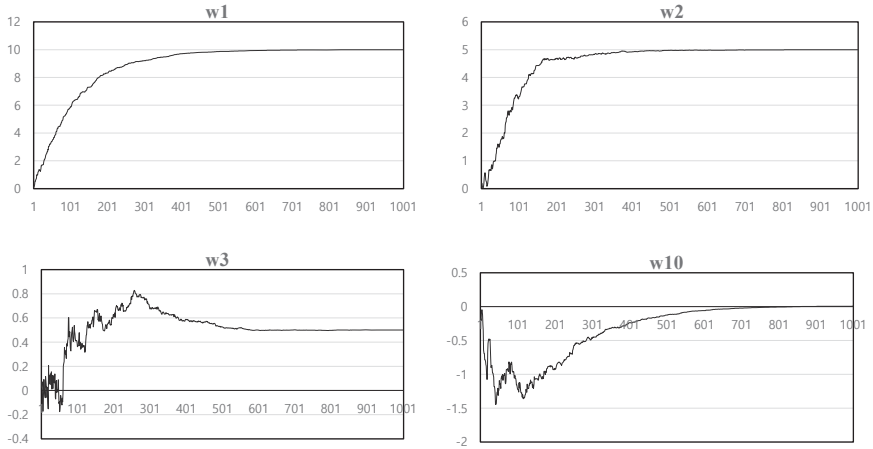
결과 분석

- **각 x 에 대한** $C = \frac{1}{2}(y^* - y)^2$

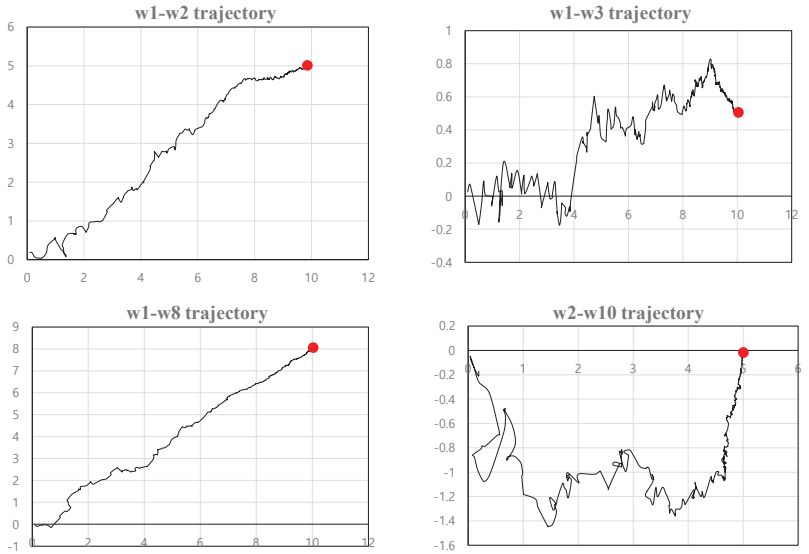


결과 분석

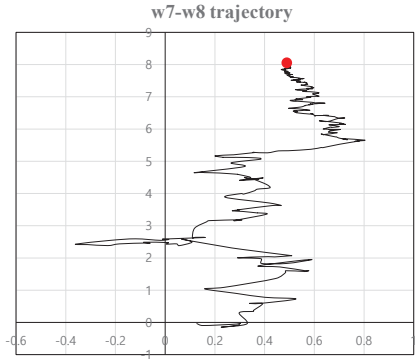
- **Weight trajectory**

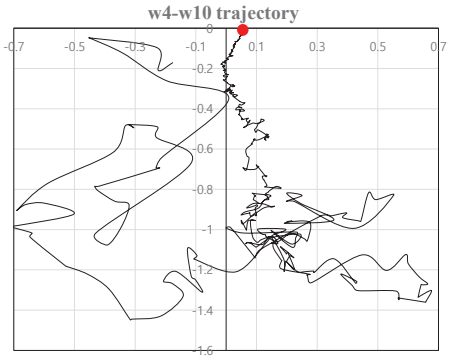


결과 분석



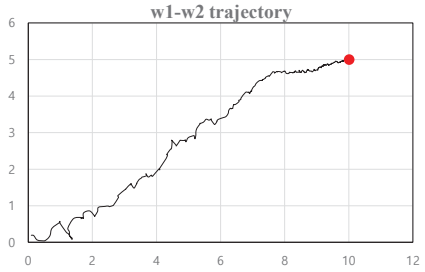
결과 분석

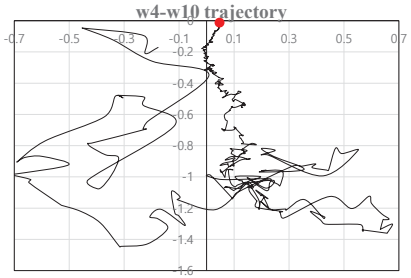




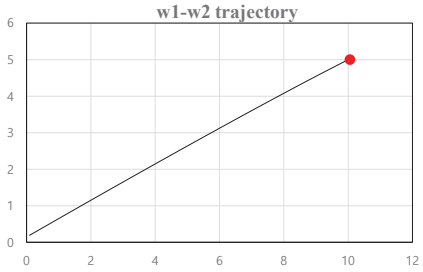
결과 분석 : Batch Size

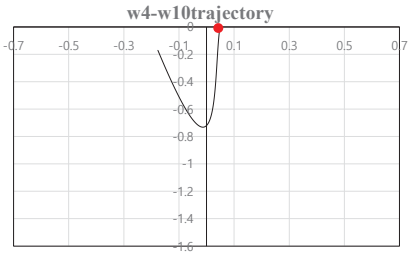
SGD, batch size = 1





GD, batch size = 1000



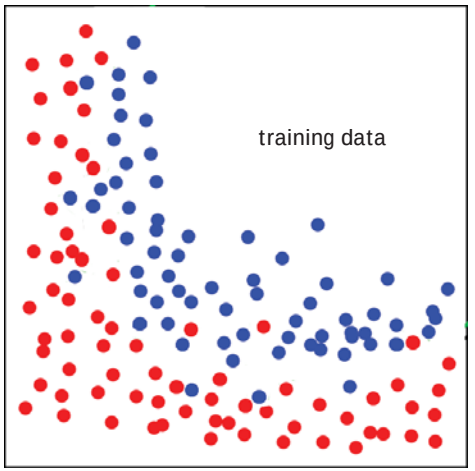


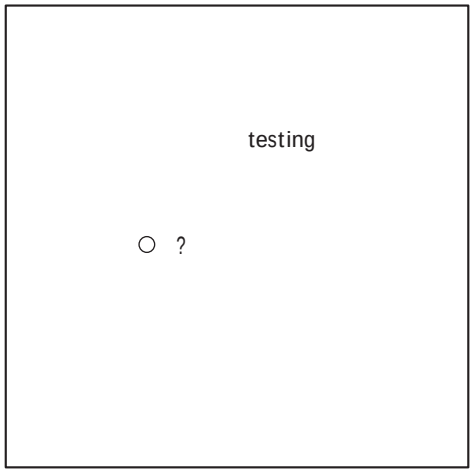
과제 제출

- **Due : 5월 15일(금) 23:59**
- **File name : SpeechHW2_이름_학번.hwp 또는 docx**
- **제출 내용**
  - C code
  - Learning output : weight, cost
  - 다양한 실험 결과와 설명

Overfitting

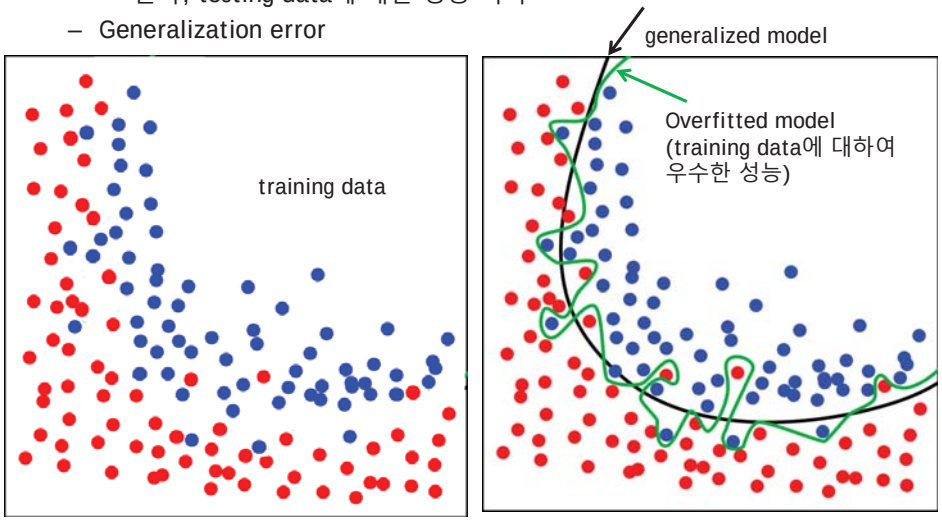
- **Classification**
  - Training data : A와 B에 의한 출력 data (x, y)
    - A/B 경계 찾기
  - Testing : (x, y) 가 주어질 때 A/B 판단





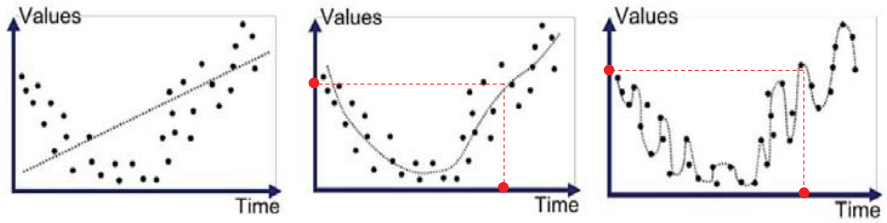
Overfitting

- **Training data에 대하여 과도하게 정확한 modeling**
  - Training data에 특화된 system → 일반화된 system을 찾지 못함
  - 그 결과, testing data에 대한 성능 저하
  - Generalization error



Overfitting

- **Regression**
  - Training data : (x, y), $y = f(x)$ 관계를 가짐
    - 실제 주어진 data는 noise 포함 : $y = f(x) + \text{noise}$
    - x 와 y 관계를 modeling 하는 function 찾기
  - Testing : x 가 주어질 때 y 예측



Underfitted Good fit Overfitted

Overfitting

- **Training data에 포함된 noise를 learning 하면 안 됨**
- **Overfitting 해결 방법**
  - Training data 크기 증가
    - Generalization 증가
  - Training 방법 개선
    - Regularization
    - Drop-out
  - Multi-model

