## 친환경적 폭발성 폐기물 처리 공정 운전 최적화

|  |  |
| --- | --- |
| **학습 내용** | |
| [문제] | 폭발성 폐기물 처리 공정의 이해 |
| [방법] | 폭발성 폐기물 처리 공정의 대리 모델(Surrogate model) 구축 |
| [응용] | 대리 모델(Surrogate model)의 최적 운전 조건 도출 |
| [요약] | * 폭발성 폐기물 공정에 대한 이해 * 대상 시스템의 입력과 응답의 관계를 인공신경망으로 모사 * 기울기 정보를 사용하지 않는 최적화를 수행하여 최적 조건 도출 |

### [공정 설명] 폭발성 폐기물 처리 공정의 이해

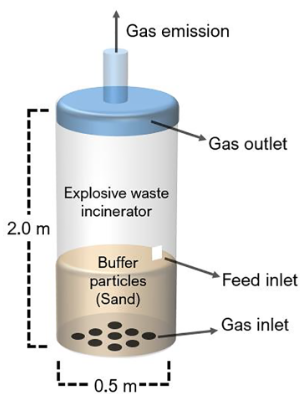


그림 1. 폭발성 폐기물 처리를 위한 유동층 반응기 구조

군용 무기 제작 시 2-methyl-1,3,5-trinitrobenzene(TNT)와 같은 폭발성 물질들이 다량 사용되며 그 결과 많은 양의 폭발성 폐기물이 배출된다. 그러나 이러한 폐기물은 영구적으로 보관할 수 없으며, 보관 가능 기간이 짧기 때문에 일정 기간이 지나면 폐기해야 한다. 이러한 폭발성 폐기물 처리 공정의 처리를 위한 다양한 방법이 연구되어 왔다. 일반적으로 Open detonation 방식과 Open burning 방식을 사용하지만 이들은 많은 양의 폐기물을 처리할 수 없을 뿐만 아니라 다량의 질소산화물(Nitrogen oxides; NOx)를 발생시킨다는 한계를 보인다.

이러한 한계를 극복하고자 고온가스를 이용한 로터리 킬른(Rotary kiln) 반응기 등 소각 공정에 대한 논의가 이루어져 왔다. 소각 공정은 많은 양의 폐기물을 연속적으로 처리할 수 있고 기존 방법에 비하여 NOx 배출량을 90% 이상 줄일 수 있다는 장점이 있다. 하지만, 로터리 킬른 반응기는 반응기 내부 온도 조절이 불가능하다는 한계가 있으며, 이는 반응기 내부에서 온도가 급격히 증가하는 핫스팟(hot spot) 발생을 촉진하며, 핫스팟 발생은 NOx 배출량을 증가시키는 요인으로 작용한다. 유동층 반응기(Fluidized bed reactor)는 로터리 킬른 반응기에 비하여 반응기 내부 온도 조절이 용이하기 때문에 핫스팟 발생을 억제할 수 있고, 결과적으로 NOx 배출량을 대폭 감소시킬 수 있다.

폭발성 폐기물을 다루는 공정의 특성상 실제 실험 전에 반응기 내부 유동의 정확한 모사가 필수적이다. Computational fluid dynamics(CFD)는 식 (1), (2)의 Navier-Stokes 방정식 등을 기반으로 유체의 복잡한 전달 현상(transport phenomena)을 상당히 정확하게 모사할 수 있는 소프트웨어 기반 방법이다.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |

CFD는 높은 정확도를 보이지만 상당한 계산 시간을 필요로 한다. 따라서 일반적으로 CFD를 대상으로 최적화를 수행하는 경우 각 변수를 특정 범위 내에서 임의의 간격으로 분할하여 조사한다. 하지만 이런 경우 전역 최적해(Global optimum)를 구하지 못하는 경우가 많다. 따라서, 이런 경우 데이터를 기반으로 실제 함수(Underlying function)를 모사하는 대리 모델(Surrogate model) 접근 방법이 좋은 해결책이 될 수 있다.

대리 모델은 최적화를 위한 목적함수로서 사용된다. 최적화는 크게 기울기 기반 방법(Derivative-based optimization)과 기울기를 사용하지 않는 방법(Derivative-free optimization)이 있다. 기울기 기반 방법은 수렴 속도가 빠른 반면에 국부 최적해를 찾는 확률이 상대적으로 높다는 단점이 있기 때문에 주로 유전 알고리즘(Genetic algorithm) 등의 기울기를 사용하지 않는 방법이 선호된다.

본 챕터에서는 CFD 시뮬레이션 데이터를 통해 인공신경망 대리 모델을 구축하고 기울기를 사용하지 않는 최적화 방식을 사용하여 효율적인 최적화를 수행하는 것을 목적으로 한다.

### [문제] 온도, 압력, 가스 유입 속도 등 반응기 운전 조건과 유입되는 입자의 조성, 입자의 크기 등 원료 조건이 질소산화물 배출에 미치는 영향을 나타내는 인공신경망 모델을 구축하고 메타휴리스틱(Metaheuristics) 최적화 방법을 적용하여 최적 조건을 규명하자.

### [방법] 폭발성 폐기물 처리 공정의 인공신경망 모델 구축

#### Q1. 관련 라이브러리를 설치하고 데이터를 작업 환경으로 불러오자.

A1. MATLAB을 사용하는 경우에 Statistics and Machine Learning Toolbox를 설치해야 한다. 해당 Toolbox를 사용하여 간편하게 인공신경망을 구축하고 최적화를 수행할 수 있다.

xlsread 를 통해 해당 경로에 있는 엑셀 파일을 불러올 수 있다.

|  |
| --- |
| RawX = xlsread('RawX.xlsx');  Response = xlsread('RawY.xlsx'); |

RawX는 xlsread 를 통해 해당 경로에 있는 엑셀 파일을 불러올 수 있으며, 아래 그림과 같이 Workspace에 두 변수 double 형태로 만들어진다.



#### Q2. 데이터를 표준화해보자.

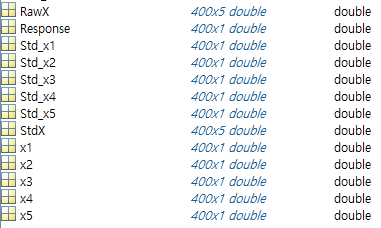
A2. 질소산화물 배출에 영향을 미치는 다섯 파라미터를 표준화하여 모델의 정확도를 향상시킬 수 있다. RawX(:, 1)은 배열 RawX을 인덱싱하는 것이다. x1에 RawX 중 1번째 열(column)에 있는 모든 값을 할당한다.

모든 파라미터를 0부터 1 사이의 값으로 변환시키기 위하여 각 파라미터의 최댓값과 최솟값을 정의하면 Std\_x1과 같이 표준화된 파라미터가 생성된다.

\*데이터 보안 상 실제 수치는 다를 수 있다.

표준화를 마친 다섯 변수 Std\_x1, Std\_x2, … , Std\_x5를 학습에 사용하기 위하여 다시 결합하여 StdX 라는 배열을 생성한다.

|  |
| --- |
| x1 = RawX(:, 1); %배열 인덱싱  x2 = RawX(:, 2);  x3 = RawX(:, 3);  x4 = RawX(:, 4);  x5 = RawX(:, 5);  min\_x1 = k1; % 각 파라미터 최댓값 및 최솟값 설정  max\_x2 = k2;  Std\_x1 = (x1-min\_x1)/(max\_x1-min\_x1); %표준화  Std\_x2 = (x2-min\_x2)/(max\_x2-min\_x2);  Std\_x3 = (x3-min\_x3)/(max\_x3-min\_x3);  Std\_x4 = (x4-min\_x4)/(max\_x4-min\_x4);  Std\_x5 = (x5-min\_x5)/(max\_x5-min\_x5);  StdX = [Std\_x1, Std\_x2, Std\_x3, Std\_x4, Std\_x5]; %데이터 결합 |



#### Q3. 인공신경망 구조를 설정해보자.

A3. 인공신경망에 사용하는 데이터는 열(column)을 기준으로 한다. 따라서 표준화된 데이터를 전치하여 데이터 형태를 맞춘다. Training function은 경사하강법 등 여러 가지를 사용할 수 있지만 일반적으로 다른 방법에 비하여 비교적 빠르게 수렴하는 Levenberg-Marquardt 방법을 사용한다. 은닉층의 노드(node) 개수는 hiddenLayerSize로 정의하는데 은닉층(hidden layer)이 한 층인 경우는 아래와 같이 선언하며 여러 층인 경우는 [10, 5, 4]와 같이 행벡터(Row vector)로 표현한다. net = fitnet(hiddenLayerSize,trainFcn)은 은닉노드의 크기가 hiddenLayerSize이고 trainFcn을 training function으로 하는 신경망 구조를 생성한다. 전체 데이터 중 학습데이터를 70%, 검증데이터를 15%, 평가데이터를 15%로 설정하였다. 학습을 진행할 최대 epochs를 1500으로 설정하였으며 학습은 최대 100초 동안 진행한다.

\*net에 대한 자세한 설명은 MATLAB Command window에 net을 입력하면 볼 수 있다.

|  |
| --- |
| x = StdX' ; %데이터 구조 변환  t = Response' ; %데이터 구조 변환  trainFcn = 'trainlm'; %Training function 설정  hiddenLayerSize = 10; %은닉층 노드(hidden node) 설정  net = fitnet(hiddenLayerSize,trainFcn); %신경망 생성  net.divideParam.trainRatio = 70/100; %학습데이터 비율 설정  net.divideParam.valRatio = 15/100; %검증데이터 비율 설정  net.divideParam.testRatio = 15/100; %평가데이터 비율 설정  net.trainParam.epochs = 1500; %학습할 최대 Epoch 횟수 설정 (default : 1000)  net.trainParam.time = 100; %학습할 최대 시간 설정(단위 : 초) (default : inf) |

위 단계를 수행하면 workspace에 class가 network인 net이 생성된다.

#### Q4. 인공신경망을 학습해보자.

A4. Train 커맨드를 사용해서 신경망을 훈련시키며 train(net, x, t)와 같이 사전에 선언한 신경망 구조와 입력과 응답을 입력한다. gsubtract 커맨드를 통해 실제 응답(t)과 신경망 응답(y) 의 차이를 구하고, perform 커맨드를 통해 신경망 성능을 정량화한다. net과 훈련 기록을 반환하며, workspace에 class가 struct인 훈련 기록 tr이 저장된다(그림 2). view 커맨드는 직접 구성한 신경망 구조를 확인할 수 있다(그림 3). Parity plot을 통해서 인공신경망의 성능과 오버피팅(overfitting) 여부를 확인한다(그림 4).

|  |
| --- |
| [net,tr] = train(net,x,t); %신경망 훈련 및 net, tr(훈련 기록) 반환  y = net(x);  e = gsubtract(t,y); %두 배열 또는 셀의 성분의 차를 구함  performance = perform(net,t,y); %신경망 성능을 계산함  view(net) %신경망 구조 반환 |

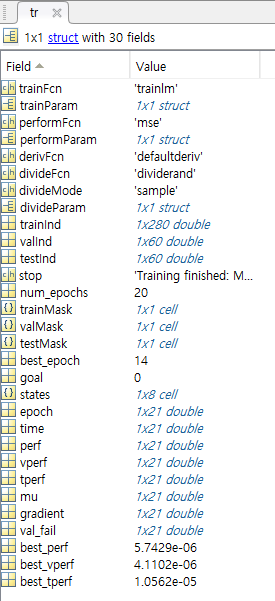
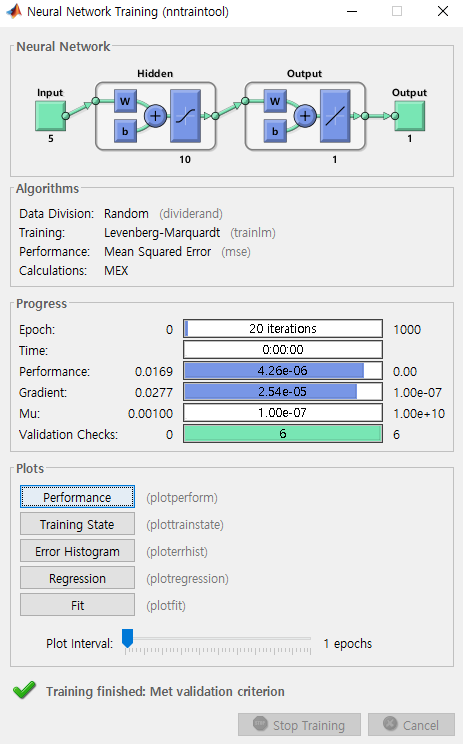


그림 2. 훈련 결과(좌)와 훈련 기록(우)

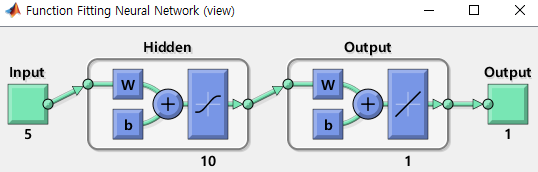


그림 3. 신경망 구조

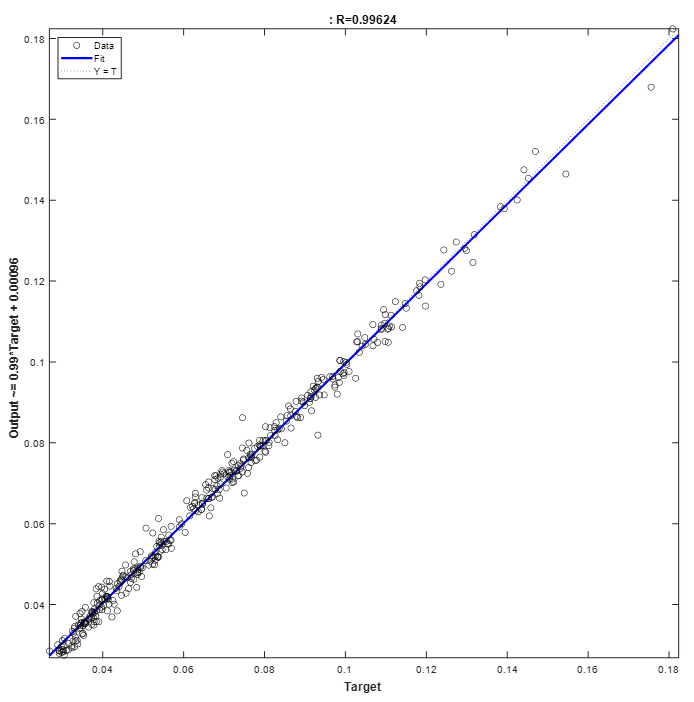


그림 4. Parity plot을 통한 모델 결과 확인

#### Q5. 인공신경망을 최적화에 사용하기 위하여 함수로 불러오자.

A5. genFunction을 통해서 학습한 신경망인 net을 함수 형태로 저장할 수 있다. NNfunction.m 이라는 파일이 MATALB 경로에 생성된다. MATLAB은 이처럼 m파일(‘-.m’형태의 파일)을 함수처럼 사용할 수 있는 기능을 제공한다. fun = @(x) NNfunction(x)는 NNfunction이라는 함수를 x에 대한 함수 fun으로 선언하는 것을 의미한다. 이때, @(x) 는 NNfunction이 x, y, z에 대한 함수인 경우에도 x만을 변수로 한다. Function handle 성능 확인을 위해 x의 첫 번째 열을 인덱싱한 후 fun의 응답을 확인하고 이를 y와 비교한다.

인공신경망 함수를 최적화의 목적 함수로 사용하기 위하여 형태를 변형시켜야 한다. 먼저, genFunction은 함수를 셀(cell)형으로 반환하기 때문에 계산된 값이 스칼라가 아닌 셀형태로 반환된다. 스칼라로 목적 함수를 반환받기 위하여 ‘MatrixOnly’를 ‘Yes’로 선언한다. 다음으로, 데이터 입력 형태에 관한 변환이다. 현재 열을 기준으로 학습되어 있는 인공신경망을 최적화에 사용하기 위하여 행으로 변환해야한다. 따라서 NNfunction의 인수를 전치하여 입력한다.

|  |
| --- |
| genFunction(net,'NNfunction', ‘MatrixOnly’, ‘yes’); %학습된 신경망 코드 생성  fun = @(x) NNfunction(x’); %신경망 구조를 갖는 함수 생성(function handle)  [Command Window]  >> y1 = fun(x(:, 1)) %function handle 결과 확인  ans = 0.0495 |

### [응용] 메타휴리스틱 방법을 통한 최적 조건 규명

대상 함수의 기울기 정보를 사용하는 최적화는 기울기 정보가 정확하고 얻기 쉬운 경우에는 최적해를 빠르게 얻을 수 있다. 하지만, 실제로 기울기에 대한 정보를 얻을 수 있는 경우는 제한적이며 이에 대한 신뢰도를 보장할 수 없다. 이러한 한계는 기울기 정보를 사용하지 않는 derivative-free optimization(DFO)를 통해 극복할 수 있다. 유전알고리즘(Genetic algorithm)은 메타휴리스틱 최적화 방법 중 가장 널리 사용되는 방법으로 변이(mutation), 교배(crossover) 등을 기반으로 최적화를 진행한다.

|  |
| --- |
| Numvar = 5; %입력 파라미터 개수  lowerbound = [0 0 0 0 0]; %각 변수의 최솟값  Upperbound = [1 1 1 1 1]; %각 변수의 최댓값  opts = optimoptions(@ga, 'PlotFcn',{@gaplotbestf}); %최적화 옵션  [x,fval,exitFlag,Output] = ga(fun,Numvar,[],[],[],[],lowerbound,Upperbound,[], opts); |

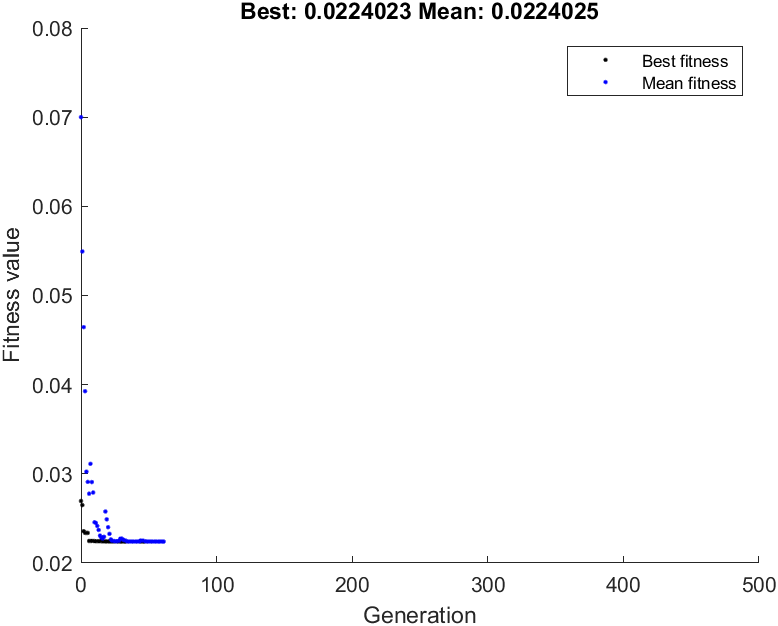


그림 5. 유전알고리즘 수행 과정

최적화 옵션을 통해 다양한 파라미터를 설정할 수 있으며 그림 5와 같이 최적화가 수행되는 과정을 도식화 할 수 있다**.** 최적화 결과 질소산화물의 최솟값은 0.0224로 구해지며 그때의 운전조건은 다음과 같다.

하지만, 이처럼 유전 알고리즘과 같은 DFO 알고리즘을 통해 얻은 해를 전역 최적해(global optimal)이라고 보장할 수 없으며 다양한 알고리즘을 적용하여 최대 혹은 최솟값을 도출하는 알고리즘의 해를 선택한다.

### [결론]

공학 분야에서 어떤 시스템에 내재된 관계(underlying function)는 수학적으로 복잡한 경우가 지배적이기 때문에 시스템 응답의 예측 및 최적화에 어려움이 많다. 본 챕터에서는 폭발성 폐기물을 처리하는 공정에서 발생하는 질소산화물을 저감시킬 수 있는 운전 조건 및 원료 조건을 규명하였다. 인공신경망을 구축하여 대리 모델을 만들고 대리 모델을 목적 함수로 사용하여 기울기 정보를 사용하지 않는 최적화를 수행하였다.

### [학습 결과]

* 학습 내용

공정 운전 조건 최적화를 위한 신경망 대리 모델 구축 및 최적화 적용 방법 익히기

* 학습 결과 확인하기

신경망을 통한 시스템 모사 방법 및 파라미터 변경 방법, 최적화 진행 과정 분석하기

* 학습 결과 응용하기

수학적으로 표현하기 어려운 시스템을 대리 모델 구축과 기울기를 사용하지 않는 최적화를 이용하여 효율적으로 최적화하기

|  |  |  |  |
| --- | --- | --- | --- |
| **핵심 역량 평가** | | | |
| 문제1 | 입력 데이터의 스케일을 0부터 1까지로 표준화 했을 때 기대되는 장점은 무엇인지 쓰시오. | | |
| 정답 |  | O | X |
| 문제2 | 검증데이터(validation data)의 역할을 쓰시오. | | |
| 정답 |  | O | X |
| 문제3 | 은닉층 노드의 개수와 예측 정확도의 관계를 쓰시오. | | |
| 정답 |  | O | X |
| 문제4 | 유전 알고리즘 파라미터 중 population의 수가 결과에 미치는 영향을 쓰시오. | | |
| 정답 |  | O | X |