Portland State
UNIVERSITY

모델을 이용해서 등고선 그리기

MATLAB R2020a 사용

안 내

- 본 자료는 아래의 사람들이 만들었습니다.
  유현정 (Portland State University)
  이기동 (Portland State University)
  김철희 (한국생산기술연구원, Portland State University)

- 예제 파일은 아래에서 받을 수 있습니다.
  https://deepjoining.github.io/

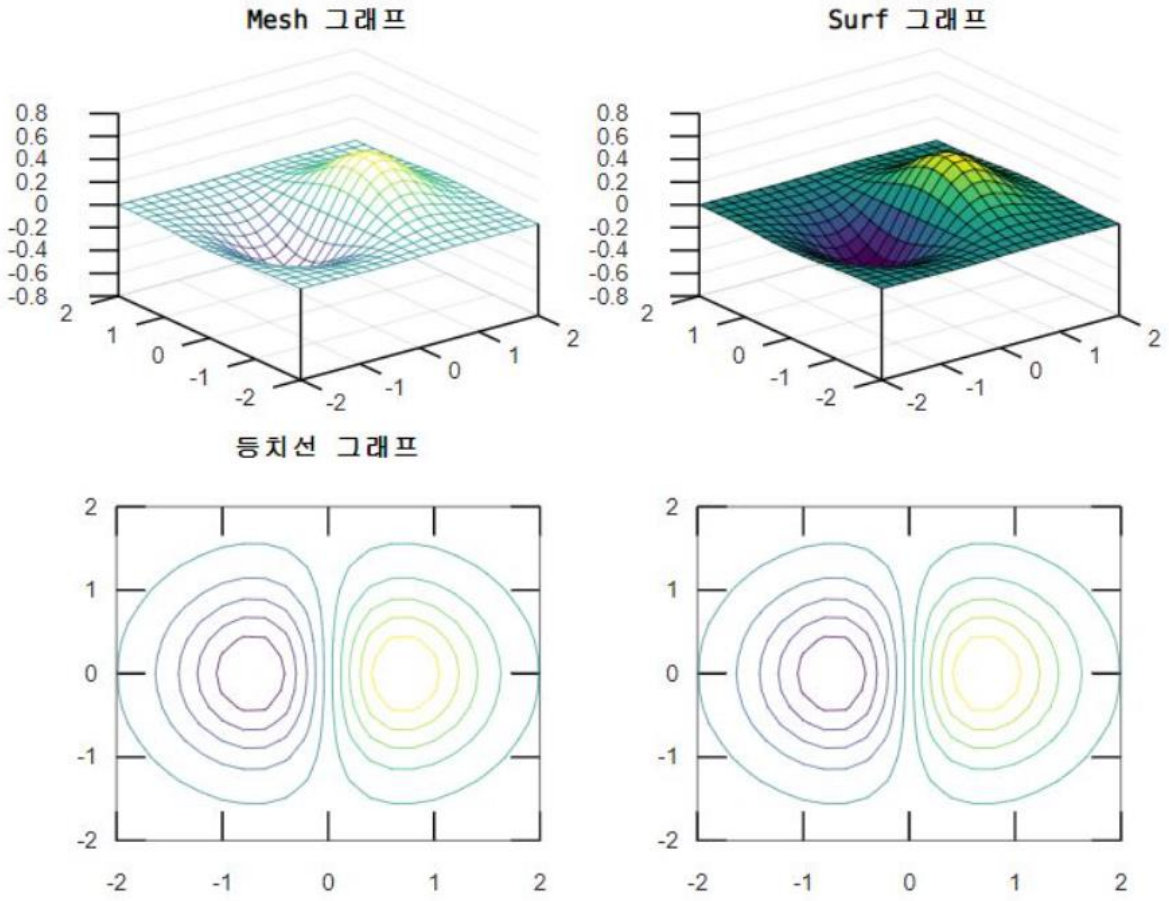
- 문의사항 및 의견: deepjoining@gmail.com

- 자료는 한국생산기술연구원 용접접합그룹 신입대학원생 교육자료입니다.
  일체의 다른 용도 사용을 금지합니다.

1. 모델의 시각화

- 등고선도(Contour plot)

  학습된 모델에 대하여 2가지의 수치형 변수를 활용하여 시각화하는 방법

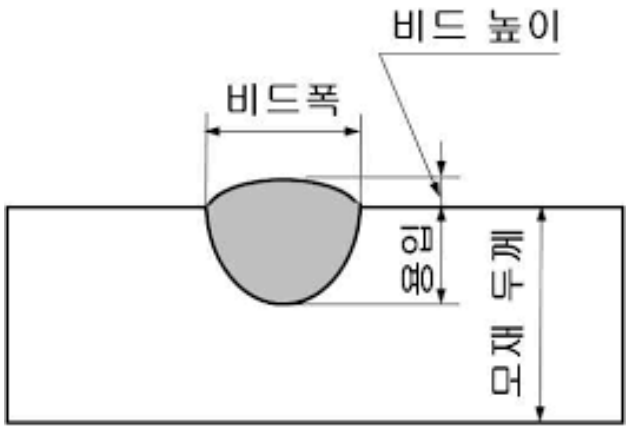


2. 풀어야 할 문제

- 소재: 여러 종류의 연강과 스테인레스 강
- 용접방법: 레이저 용접
- 사용한 레이저 종류: 디스크 레이저, 파이버 레이저
- 용접부 평가방법: 용입깊이 측정

- 모델링할 문제
  * 다양한 레이저 출력, 용접속도, 빔 직경에 따른 용접부의 용입깊이

Input data　　Output data



3. 용입깊이 GPR 모델에서 등고선 그리기(MATLAB)

```
%변수 설정 (4kW의 출력을 가진 모델 학습시키고자 할 때)
%x1: laser power, x2: welding speed, x3: beam diameter
x1=4;
%linspace(1,7)은 1부터 7까지 100등분한 행렬
x2=linspace(1,7);
%linspace(0.1,0.7)은 0.1부터 0.7까지 100등분한 행렬
x3=linspace(0.1,0.7);
[numRows,numCols]=size(x2);

% traindata2는 사용한 데이터 파일
% input은 1~3열, output은 4열에 있으며, input data 사용
% 1: power, 2: welding speed, 3: beam diameter, 4: penetration depth
A=laserdata([1],[1:3]);

%trainedModel은 회귀 학습 후 모델 내보내기를 통해 불러온 파일
Z2=Cal(x1,x2,x3,numCols,A,trainedModel);

%그래프 label 지정
axlbl = @(h) [xlabel(h,'Welding speed (m/min)'); ylabel(h,'Beam diameter
(mm)')];
figure(1)
contour(x2,x3,Z2,'ShowText','on');
title('Contour plot for penetration depth for 4 kW laser power');
axlbl(gca);
```

```
%원하는 변수에 따라 등고선 그리기
function[Z]=Cal(x1,x2,x3,numCols,A,trainedModel)
for i = 1:numCols
    for j = 1:numCols
        A{1,1}=x1;
        A{1,2}=x2(j);
        A{1,3}=x3(i);

        Z(i,j)=trainedModel.predictFcn(A([1],:));
    end
end
end
```

3. 용입깊이 SNN 모델에서 등고선 그리기(MATLAB)

```
%변수 설정 (4kW의 출력을 가진 모델 학습시키고자 할 때)
%x1: laser power, x2: welding speed, x3: beam diameter
x1=4;
%linspace(1,7)은 1부터 7까지 100등분한 행렬
x2=linspace(1,7);
%linspace(0.1,0.7)은 0.1부터 0.7까지 100등분한 행렬
x3=linspace(0.1,0.7);
[numRows,numCols] = size(x2);

%그래프 label 지정
Z2 = NNCal(x1,x2,x3,numCols);
axlbl = @(h) [xlabel(h, 'Welding speed (m/min)'); ylabel(h, 'Beam diameter (mm)')];
figure(1)
contour(x2,x3,Z2,'ShowText','on')
title('Contour plot for penetration depth for 4 kW laser power')
axlbl(gca);

%원하는 변수에 따라 등고선 그리기
function[Z] = NNCal(x1,x2,x3,numCols)
Z = zeros(numCols, numCols);
 for i = 1:numCols
     for j = 1:numCols
    testx=[x1,x2(j),x3(i)];
    Z(i,j) = myNeuralNetworkFunction(testx);
     end
 end
end
```

4. 용입깊이 DNN 모델에서 등고선 그리기(Anaconda3)

```
# Prediction of penetration depth in high power laser welding

import keras
import tensorflow as tf
import numpy as np

import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import copy

from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from keras.optimizers import Adam
from keras import backend as K
from keras import metrics

from sklearn.model_selection import train_test_split

xy = np.loadtxt('C:/temp/steel.csv', delimiter=',', dtype=np.float32)

# 1: laser power, 2: Welding speed, 3: Beam diameter, 4: Penetration depth
x_data = xy[:, 0:-1]
#x_org = copy.deepcopy(x_data)

# Normalizing data
xmean=x_data.mean(axis=0)
xstd=x_data.std(axis=0)
x_data -= xmean
x_data /= xstd

y_data = xy[:, [-1]]

#원래는 데이터 나눠서 train데이터로 정규화해야함
ymean = y_data.mean(axis=0)
ystd = y_data.std(axis=0)

y_data -= ymean
y_data /= ystd

# split data: Training data를 70%
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data,
                                   train_size=0.70, random_state=42)

# split data: Validation과 Test data를 50:50 으로
x_val, x_test, y_val, y_test = train_test_split(x_test, y_test,
                               train_size=0.50, random_state=42)
```

4. 용입깊이 DNN 모델에서 등고선 그리기(Anaconda3)

```
model = Sequential()
# Dense(64) is a fully-connected layer with 64 hidden units.
# in the first layer, you must specify the expected input data shape:
# here, 6-dimensional vectors
model.add(Dense(64, activation='relu', input_shape=(3,)))
model.add(Dense(64, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(1))

#sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
adam = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-8)
model.compile(loss='mse', optimizer='adam', metrics=['mae'])

history=model.fit(x_train, y_train, epochs=200, batch_size=10,
validation_data=(x_val,y_val))

# 교재 Fig. 3-9
plt.plot(history.history['loss'], label='Training loss')
plt.plot(history.history['val_loss'], label='Validation loss')
plt.title('model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.show()

score = model.evaluate(x_test, y_test, batch_size=10)
```

등고선 그리는 학습 코드

```
# 1: laser power, 2: Welding speed, 3: Beam diameter, 4: Penetration depth
# delta = 0.1 은 x와 y 데이터를 0.1 간격으로 등분하여 행렬 생성
delta  = 0.1
x = np.arange(1.0, 7.0, delta)
y = np.arange(0.1, 0.7, delta)
X, Y = np.meshgrid(x, y)

zz=np.zeros((len(y),len(x)),dtype=float)

for i in range(len(y)):
    for j in range(len(x)):
        graphx = [[4, x[j], y[i]]]
        graphx -= xmean
        graphx /= xstd
        result=model.predict(graphx)*ystd+ymean
        zz[i,j]=result

fig, ax = plt.subplots()
CS = ax.contour(X, Y, zz)
ax.clabel(CS, inline=1, fontsize=10)
ax.set_title(' Contour plot for penetration depth for 4 kW laser power ')
```

감 사 합 니 다