Portland State
UNIVERSITY

하이퍼 파라미터 최적화
(교재 7. 3. 2)

Anaconda Jupyter notebook 사용

안 내

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

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

- 문의사항 및 의견: deepjoining@gmail.com

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

1. 하이퍼 파라미터 최적화

- 딥러닝 모델을 만들 때 사용하는 layer의 개수, layer 당 node의 개수 등 여러가지 조건들(하이퍼파라미터, hyperparameter)을 토대로 가장 높은 성능을 갖는 하이퍼파라미터를 선택하는 방법

- Keras Tuner
  - 참고자료

  Ref. https://keras-team.github.io/keras-tuner/

  https://www.tensorflow.org/tutorials/keras/keras_tuner

  https://www.tensorflow.org/tutorials/keras/keras_tuner?hl=ko

  https://keras-team.github.io/keras-tuner/documentation/tuners/

  설치 방법: Anaconda prompt 실행 후 pip install -q -U keras-tuner 명령어 작성

1. 하이퍼 파라미터 최적화

1. 모델의 정의

```
def build_model(hp):
    model = Sequential()
    for i in range(hp.Int('num_layers', 2, 5)):                    #layer를 2에서 5까지
        model.add(layers.Dense(units=hp.Int('units_' + str(i),
                                    min_value=16,
                                    max_value=64,
                                    step=16),                      #node는 16에서 64까지 16간격으로
                               activation='relu'))
    model.add(layers.Dense(1))
    model.compile(
        optimizer=keras.optimizers.Adam(
            hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])),       # Adam에서 학습률은 3개 중에서
        loss='mse',metrics=['mae'])
    return model
```

1. 하이퍼 파라미터 최적화

2-1. 랜덤으로 찾는 방법

```
from kerastuner.tuners import RandomSearch
tuner = RandomSearch(
    build_model,
    objective='val_mae',
    max_trials=100,                                    # 이거는 랜덤으로 100번까지 찾는 방법
    executions_per_trial=3,
    directory='./Data/',
    project_name='Keras Tuner Test')

tuner.search(x_train, y_train,
             epochs=50,
             validation_data=(x_val, y_val))

tuner.search_space_summary()
tuner.results_summary()

# check results
models = tuner.get_best_models(num_models=3)
```

1. 하이퍼 파라미터 최적화

2-2. 하이퍼밴드로 찾는 방법

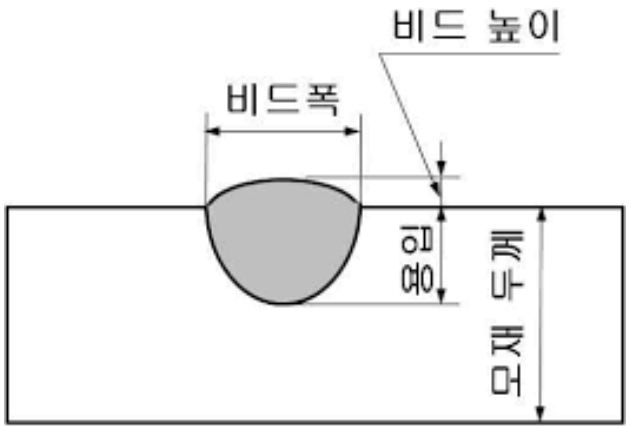
```
from kerastuner.tuners import Hyperband
# 이거는 hyperband tuner 사용하는 방법
tuner = Hyperband(build_model, objective = 'val_mae', max_epochs = 50, factor = 3, directory = './Data',
project_name = 'hyperband')
tuner.search(x_train, y_train, epochs = 50, validation_data = (x_val, y_val))

# Get the optimal hyperparameters
best_hps = tuner.get_best_hyperparameters(num_trials = 1)[0]
print('The hyperparameter search is complete. The optimal number of layer is', best_hps.get('num_layers'))
print('The number of units in each hidden layer is')
for i in range(best_hps.get('num_layers')):
    print(best_hps.get('units_'+str(i)))
print ('The optimal learning rate for the optimizer is', best_hps.get('learning_rate'))

# Build the model with the optimal hyperparameters and train it on the data
model = tuner.hypermodel.build(best_hps)
history=model.fit(x_train, y_train, epochs = 50, validation_data = (x_val, y_val))
```

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



감 사 합 니 다