ntran_model_optimize_performance

May 10, 2022

[1]:
```
# handle imports
import os
import tensorflow as tf
from PIL import Image
import random
import pathlib
import numpy as np

from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import Ridge, Lasso, LassoCV, RidgeCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC

from sklearn.metrics import confusion_matrix, classification_report
```

[2]:
```
current_path = os.getcwd()
dataset_path = current_path + '\dataset'
dataset_path
```

[2]:
```
'c:\\Users\\tickn\\ml\\EE257\\EE257 Project\\dataset'
```

[3]:
```
# Load dataset and split

data_dir = pathlib.Path(dataset_path + '\data')
batch_size = 32

def describe_img(filepath):
    rand_img = random.choice(list(filepath.glob('**\*.jpg')))
    width, height = Image.open(str(rand_img)).size

    return width, height

def random_img(filepath):
    return  random.choice(list(filepath.glob('**\*.jpg')))
```

```python
img_width, img_height = describe_img(data_dir)

# load image dataset
train_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split = 0.2,
    subset = "training",
    seed = 123,
    color_mode="grayscale",
    image_size = (img_height , img_width),
    batch_size = batch_size
)

test_ds = tf.keras.utils.image_dataset_from_directory(
    data_dir,
    validation_split = 0.2,
    subset = "validation",
    seed = 123,
    color_mode="grayscale",
    image_size = (img_height , img_width),
    batch_size = batch_size
)
```

```
Found 597 files belonging to 4 classes.
Using 478 files for training.
Found 597 files belonging to 4 classes.
Using 119 files for validation.
```

[4]:
```python
def dataset_to_2D(dataset):
    x = []
    y = []
    for img_batch, label_batch in dataset:
        # flatten images since model fit() needs 2D input
        for img in img_batch:
            x.append(img.flatten())
        for label in label_batch:
            y.append(label)
    return x, y

x_train, y_train = dataset_to_2D(train_ds.as_numpy_iterator())
x_test, y_test = dataset_to_2D(test_ds.as_numpy_iterator())

print(np.shape(x_train))
print(np.shape(y_train))
```

```
(478, 62500)
(478,)
```

[5]:
```
# model = GridSearchCV(
#     Lasso(),
#     param_grid={
#         "alpha" : [0.1 , 1 , 10 , 100 , 1000 , 10000],
#         "fit_intercept" : [True , False],
#         "normalize" : [True , False]
#     },
#     scoring='neg_mean_squared_error'
# )
```

[6]:
```
# model.fit(x_train, y_train)
# print(model.best_params_)
```

[7]:
```
# model2 = GridSearchCV(
#     Ridge(),
#     param_grid={
#         "alpha" : [1 , 10 , 100 , 1000 , 10000],
#         "fit_intercept" : [True , False],
#         "normalize" : [True , False]
#     },
#     scoring='neg_mean_squared_error'
# )

# model2.fit(x_train, y_train)
# print(model2.best_params_)
```

[8]:
```
# print(model2.best_estimator_)
```

[16]:
```
l1_selector = SelectFromModel(estimator=Lasso(alpha=1,max_iter=10000)).
 ↪fit(x_train, y_train)
l2_selector = SelectFromModel(estimator=Ridge(alpha=1)).fit(x_train, y_train)

l1_train = l1_selector.transform(x_train)
l1_test = l1_selector.transform(x_test)

l2_train = l2_selector.transform(x_train)
l2_test = l2_selector.transform(x_test)
```

[17]:
```
print(np.shape(x_train))
print(np.shape(l1_train))
print(np.shape(l1_test))
print(np.shape(l2_train))
print(np.shape(l2_test))
```

```
(478, 62500)
```

```
(478, 176)
(119, 176)
(478, 25357)
(119, 25357)
```

[18]:
```
clf1 =  SVC(C=10.0 , kernel='rbf')
clf1.fit(l1_train,y_train)

print(" Training error l1 reg: %f " %clf1.score(l1_train, y_train))
print(" Test error l1 reg: %f " %clf1.score(l1_test, y_test))

clf1.fit(l2_train,y_train)

print()
print(" Training error with l2 reg: %f " %clf1.score(l2_train, y_train))
print(" Test error l2 reg: %f " %clf1.score(l2_test, y_test))
```

```
 Training error l1 reg: 0.951883
 Test error l1 reg: 0.504202

 Training error with l2 reg: 0.953975
 Test error l2 reg: 0.478992
```

[19]:
```
clf2 = DecisionTreeClassifier(criterion="entropy")

clf2.fit(l1_train,y_train)

print(" Training error l1 reg: %f " %clf2.score(l1_train, y_train))
print(" Test error l1 reg: %f " %clf2.score(l1_test, y_test))

clf2.fit(l2_train,y_train)

print()
print(" Training error with l2 reg: %f " %clf2.score(l2_train, y_train))
print(" Test error l2 reg: %f " %clf2.score(l2_test, y_test))
```

```
 Training error l1 reg: 1.000000
 Test error l1 reg: 0.462185

 Training error with l2 reg: 1.000000
 Test error l2 reg: 0.378151
```

[20]:
```
clf3 = RandomForestClassifier()

clf3.fit(l1_train,y_train)

print(" Training error l1 reg: %f " %clf3.score(l1_train, y_train))
print(" Test error l1 reg: %f " %clf3.score(l1_test, y_test))
```

```
clf3.fit(l2_train,y_train)

print()
print(" Training error with l2 reg: %f " %clf3.score(l2_train, y_train))
print(" Test error l2 reg: %f " %clf3.score(l2_test, y_test))
```

```
 Training error l1 reg: 1.000000
 Test error l1 reg: 0.512605

 Training error with l2 reg: 1.000000
 Test error l2 reg: 0.512605
```

[21]:
```
clf4 = LogisticRegression(solver='newton-cg',max_iter=10000)

clf4.fit(l1_train,y_train)

print(" Training error l1 reg: %f " %clf4.score(l1_train, y_train))
print(" Test error l1 reg: %f " %clf4.score(l1_test, y_test))

clf4.fit(l2_train,y_train)

print()
print(" Training error with l2 reg: %f " %clf4.score(l2_train, y_train))
print(" Test error l2 reg: %f " %clf4.score(l2_test, y_test))
```

```
c:\Users\tickn\anaconda3\lib\site-packages\scipy\optimize\linesearch.py:327:
LineSearchWarning: The line search algorithm did not converge
  warn('The line search algorithm did not converge', LineSearchWarning)
c:\Users\tickn\anaconda3\lib\site-packages\sklearn\utils\optimize.py:195:
UserWarning: Line Search failed
  warnings.warn('Line Search failed')
```

```
 Training error l1 reg: 1.000000
 Test error l1 reg: 0.344538

 Training error with l2 reg: 1.000000
 Test error l2 reg: 0.436975
```

```
c:\Users\tickn\anaconda3\lib\site-packages\scipy\optimize\linesearch.py:327:
LineSearchWarning: The line search algorithm did not converge
  warn('The line search algorithm did not converge', LineSearchWarning)
c:\Users\tickn\anaconda3\lib\site-packages\sklearn\utils\optimize.py:195:
UserWarning: Line Search failed
  warnings.warn('Line Search failed')
```