ex14

September 25, 2024

[1]:
```
import numpy as np
from scikeras.wrappers import KerasClassifier
from sklearn.model_selection import GridSearchCV
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import RMSprop, Adam
```

[2]:
```
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.reshape((60000, 28 * 28)).astype('float32') / 255
test_images = test_images.reshape((10000, 28 * 28)).astype('float32') / 255
train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)
```

[3]:
```
def create_model(units=512, optimizer='rmsprop'):
    model = models.Sequential()
    model.add(layers.Dense(units, activation='relu', input_shape=(28 * 28,)))
    model.add(layers.Dense(10, activation='softmax'))

    model.compile(optimizer=optimizer, loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
    return model
```

[4]:
```
model = KerasClassifier(model=create_model, units=512, optimizer='rmsprop')

param_grid = {
    'units': [128, 256, 512],
    'batch_size': [128, 256, 500],
    'epochs': [10, 20],
    'optimizer': [RMSprop(), Adam()]
}
```

[6]:
```
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(train_images, train_labels)
```

```
c:\Python311\Lib\site-packages\keras\src\layers\core\dense.py:87: UserWarning:
Do not pass an `input_shape`/`input_dim` argument to a layer. When using
Sequential models, prefer using an `Input(shape)` object as the first layer in
```

```
the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
Epoch 1/20
469/469             2s 3ms/step -
accuracy: 0.8674 - loss: 0.4520
Epoch 2/20
469/469             2s 3ms/step -
accuracy: 0.9662 - loss: 0.1136
Epoch 3/20
469/469             1s 3ms/step -
accuracy: 0.9777 - loss: 0.0737
Epoch 4/20
469/469             1s 3ms/step -
accuracy: 0.9854 - loss: 0.0484
Epoch 5/20
469/469             2s 3ms/step -
accuracy: 0.9896 - loss: 0.0373
Epoch 6/20
469/469             2s 3ms/step -
accuracy: 0.9921 - loss: 0.0272
Epoch 7/20
469/469             1s 3ms/step -
accuracy: 0.9941 - loss: 0.0207
Epoch 8/20
469/469             1s 3ms/step -
accuracy: 0.9958 - loss: 0.0166
Epoch 9/20
469/469             1s 3ms/step -
accuracy: 0.9972 - loss: 0.0116
Epoch 10/20
469/469             1s 3ms/step -
accuracy: 0.9983 - loss: 0.0081
Epoch 11/20
469/469             1s 3ms/step -
accuracy: 0.9989 - loss: 0.0061
Epoch 12/20
469/469             2s 3ms/step -
accuracy: 0.9993 - loss: 0.0042
Epoch 13/20
469/469             2s 3ms/step -
accuracy: 0.9995 - loss: 0.0033
Epoch 14/20
469/469             2s 3ms/step -
accuracy: 0.9998 - loss: 0.0019
Epoch 15/20
469/469             2s 3ms/step -
accuracy: 0.9998 - loss: 0.0017
```

```
Epoch 16/20
469/469             2s 3ms/step -
accuracy: 0.9999 - loss: 9.0990e-04
Epoch 17/20
469/469             1s 3ms/step -
accuracy: 1.0000 - loss: 5.8648e-04
Epoch 18/20
469/469             1s 3ms/step -
accuracy: 1.0000 - loss: 5.2211e-04
Epoch 19/20
469/469             1s 3ms/step -
accuracy: 1.0000 - loss: 3.7025e-04
Epoch 20/20
469/469             2s 3ms/step -
accuracy: 1.0000 - loss: 3.2601e-04
```

[7]:
```
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
```

```
Best: 0.979900 using {'batch_size': 128, 'epochs': 20, 'optimizer':
<keras.src.optimizers.adam.Adam object at 0x000001CA1DC01D90>, 'units': 512}
```

[8]:
```
best_model = grid_result.best_estimator_
test_acc = best_model.score(test_images, test_labels)

print("Test accuracy with best model:", test_acc)
```

```
79/79             0s 2ms/step
Test accuracy with best model: 0.9842
```