model_resisc45_2

April 16, 2023

1 Model RESISC45

1.1 Train details

- ResNet18 pretrained and fine-tune only last layer:
  - weights are trained on IMAGENET1K_V1
  - acc@1 (on ImageNet-1K): 69.758
  - acc@5 (on ImageNet-1K): 89.078
  - categories: tench, goldfish, great white shark, …
  - 11.689.512 num params
  - 1.81 GFLOPS
  - File size: 44.7 MB
- dataset RESISC45
- 45 classes (all from dataset)
- cross-entropy optimizer
- SGD (stochastic gradient descent) with learning rate 0.001 and momentum 0.9
- dataset split in 5 folds
- model input resolution: 3 x 224 x 244
- image transforms at train (augmentations):
  - random resized crop at 224 x 224 resolution (from original 256 x 256)
  - random horizontal flip
  - normalize with mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225]
- image transforms at validation:
  - central crop at 224 x 224 resolution (from original 256 x 256)
  - normalize with mean [0.485, 0.456, 0.406] and std [0.229, 0.224, 0.225]
- number of trained epochs: 15
- batch size of 32

[ ]:

```python
from utils import train_pretrained_resnet, TrainConfig
```

1.2 Validation on fold 1

[ ]:

```python
print("Start.")
train_pretrained_resnet(
    TrainConfig(
        num_epochs=15,
        batch_size=32,
        val_fold=1,
    )
)
print("Done.")
```

```
Start.
macOS-13.3.1-arm64-arm-64bit
PyTorch Version:  2.0.0
Torchvision Version:  0.15.1
Using mps device
```

```
/Users/cristianion/Desktop/satimg_model/.venv/lib/python3.11/site-
packages/torchvision/models/_utils.py:208: UserWarning: The parameter
'pretrained' is deprecated since 0.13 and may be removed in the future, please
use 'weights' instead.
  warnings.warn(
/Users/cristianion/Desktop/satimg_model/.venv/lib/python3.11/site-
packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a
weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed
in the future. The current behavior is equivalent to passing
`weights=ResNet18_Weights.IMAGENET1K_V1`. You can also use
`weights=ResNet18_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)
```

```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer2): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer3): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=512, out_features=45, bias=True)
```

```
)
Params to learn:
         fc.weight
         fc.bias
SGD (
Parameter Group 0
    dampening: 0
    differentiable: False
    foreach: None
    lr: 0.001
    maximize: False
    momentum: 0.9
    nesterov: False
    weight_decay: 0
)
Fold: 1
------------------------------
Params to learn:
         fc.weight
         fc.bias
Epoch 1
-------------------------------
Started train.
loss: 3.890680  [   32/25200]
loss: 3.182249  [ 3232/25200]
loss: 2.741067  [ 6432/25200]
loss: 2.386566  [ 9632/25200]
loss: 2.038327  [12832/25200]
loss: 1.955219  [16032/25200]
loss: 2.041826  [19232/25200]
loss: 1.697521  [22432/25200]
Started validation.
Test Error:
 Accuracy: 69.5%, Avg loss: 1.321998

Epoch 2
-------------------------------
Started train.
loss: 2.177585  [   32/25200]
loss: 1.545591  [ 3232/25200]
loss: 1.262399  [ 6432/25200]
loss: 1.892634  [ 9632/25200]
loss: 1.128474  [12832/25200]
loss: 1.068921  [16032/25200]
loss: 1.356302  [19232/25200]
loss: 1.596925  [22432/25200]
Started validation.
Test Error:
 Accuracy: 74.2%, Avg loss: 1.001065

Epoch 3
-------------------------------
Started train.
loss: 1.502834  [   32/25200]
loss: 1.505419  [ 3232/25200]
loss: 1.115209  [ 6432/25200]
loss: 1.152848  [ 9632/25200]
loss: 0.811948  [12832/25200]
loss: 1.642592  [16032/25200]
loss: 1.445277  [19232/25200]
loss: 1.380301  [22432/25200]
Started validation.
Test Error:
 Accuracy: 75.2%, Avg loss: 0.908210

Epoch 4
-------------------------------
Started train.
loss: 1.187894  [   32/25200]
loss: 0.961881  [ 3232/25200]
loss: 1.120513  [ 6432/25200]
loss: 1.319457  [ 9632/25200]
loss: 1.367835  [12832/25200]
loss: 0.970807  [16032/25200]
loss: 0.921183  [19232/25200]
loss: 0.926089  [22432/25200]
Started validation.
Test Error:
 Accuracy: 76.6%, Avg loss: 0.828743

Epoch 5
-------------------------------
Started train.
loss: 0.755459  [   32/25200]
loss: 1.635669  [ 3232/25200]
loss: 0.849786  [ 6432/25200]
loss: 1.286540  [ 9632/25200]
loss: 1.082298  [12832/25200]
loss: 1.341350  [16032/25200]
loss: 0.986091  [19232/25200]
loss: 1.051346  [22432/25200]
Started validation.
Test Error:
 Accuracy: 76.6%, Avg loss: 0.811545

Epoch 6
```

```
-------------------------------
Started train.
loss: 1.487782  [   32/25200]
loss: 1.469378  [ 3232/25200]
loss: 0.852992  [ 6432/25200]
loss: 0.956591  [ 9632/25200]
loss: 0.854385  [12832/25200]
loss: 1.251342  [16032/25200]
loss: 0.814736  [19232/25200]
loss: 1.263947  [22432/25200]
Started validation.
Test Error:
 Accuracy: 77.9%, Avg loss: 0.764266

Epoch 7
-------------------------------
Started train.
loss: 1.065741  [   32/25200]
loss: 1.159396  [ 3232/25200]
loss: 0.990760  [ 6432/25200]
loss: 0.685865  [ 9632/25200]
loss: 1.057520  [12832/25200]
loss: 1.105249  [16032/25200]
loss: 1.113102  [19232/25200]
loss: 1.012909  [22432/25200]
Started validation.
Test Error:
 Accuracy: 78.3%, Avg loss: 0.748739

Epoch 8
-------------------------------
Started train.
loss: 1.341325  [   32/25200]
loss: 1.048397  [ 3232/25200]
loss: 0.700002  [ 6432/25200]
loss: 0.644860  [ 9632/25200]
loss: 0.835713  [12832/25200]
loss: 0.945018  [16032/25200]
loss: 1.122493  [19232/25200]
loss: 1.356460  [22432/25200]
Started validation.
Test Error:
 Accuracy: 78.7%, Avg loss: 0.734540

Epoch 9
-------------------------------
Started train.
loss: 1.033965  [   32/25200]
```

```
loss: 1.210716  [ 3232/25200]
loss: 0.699580  [ 6432/25200]
loss: 0.999322  [ 9632/25200]
loss: 1.356173  [12832/25200]
loss: 0.816906  [16032/25200]
loss: 1.269798  [19232/25200]
loss: 0.843787  [22432/25200]
Started validation.
Test Error:
 Accuracy: 78.6%, Avg loss: 0.721440

Epoch 10
-------------------------------
Started train.
loss: 1.223247  [   32/25200]
loss: 0.693740  [ 3232/25200]
loss: 0.797439  [ 6432/25200]
loss: 0.760607  [ 9632/25200]
loss: 0.707379  [12832/25200]
loss: 0.438519  [16032/25200]
loss: 0.941199  [19232/25200]
loss: 0.719749  [22432/25200]
Started validation.
Test Error:
 Accuracy: 78.2%, Avg loss: 0.715150

Epoch 11
-------------------------------
Started train.
loss: 1.021843  [   32/25200]
loss: 0.915179  [ 3232/25200]
loss: 1.005942  [ 6432/25200]
loss: 0.847555  [ 9632/25200]
loss: 0.668349  [12832/25200]
loss: 0.786110  [16032/25200]
loss: 0.720083  [19232/25200]
loss: 0.501673  [22432/25200]
Started validation.
Test Error:
 Accuracy: 79.2%, Avg loss: 0.699072

Epoch 12
-------------------------------
Started train.
loss: 0.794973  [   32/25200]
loss: 1.343701  [ 3232/25200]
loss: 1.006589  [ 6432/25200]
loss: 1.054204  [ 9632/25200]
```

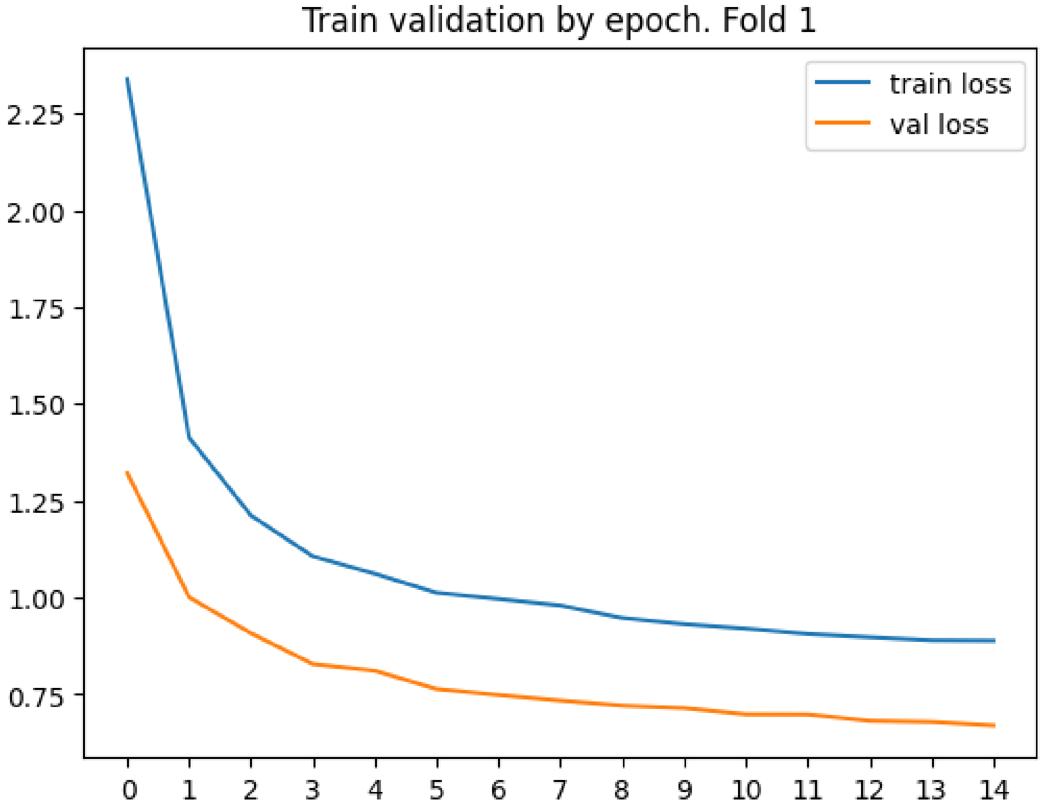
```
loss: 0.875473  [12832/25200]
loss: 1.024033  [16032/25200]
loss: 0.730604  [19232/25200]
loss: 1.066857  [22432/25200]
Started validation.
Test Error:
 Accuracy: 79.2%, Avg loss: 0.698682

Epoch 13
-------------------------------
Started train.
loss: 0.800665  [   32/25200]
loss: 0.809411  [ 3232/25200]
loss: 0.525107  [ 6432/25200]
loss: 0.731310  [ 9632/25200]
loss: 0.843429  [12832/25200]
loss: 1.317476  [16032/25200]
loss: 0.628172  [19232/25200]
loss: 0.984468  [22432/25200]
Started validation.
Test Error:
 Accuracy: 79.5%, Avg loss: 0.682057

Epoch 14
-------------------------------
Started train.
loss: 1.587573  [   32/25200]
loss: 1.116967  [ 3232/25200]
loss: 0.822276  [ 6432/25200]
loss: 1.099738  [ 9632/25200]
loss: 0.789875  [12832/25200]
loss: 0.569978  [16032/25200]
loss: 0.853551  [19232/25200]
loss: 1.324395  [22432/25200]
Started validation.
Test Error:
 Accuracy: 79.3%, Avg loss: 0.679324

Epoch 15
-------------------------------
Started train.
loss: 0.813406  [   32/25200]
loss: 1.304135  [ 3232/25200]
loss: 1.094846  [ 6432/25200]
loss: 0.633451  [ 9632/25200]
loss: 1.092720  [12832/25200]
loss: 0.765974  [16032/25200]
loss: 1.154358  [19232/25200]
```

```
loss: 0.631918  [22432/25200]
Started validation.
Test Error:
 Accuracy: 79.9%, Avg loss: 0.670507
```



```
Done.
```

[ ]: