



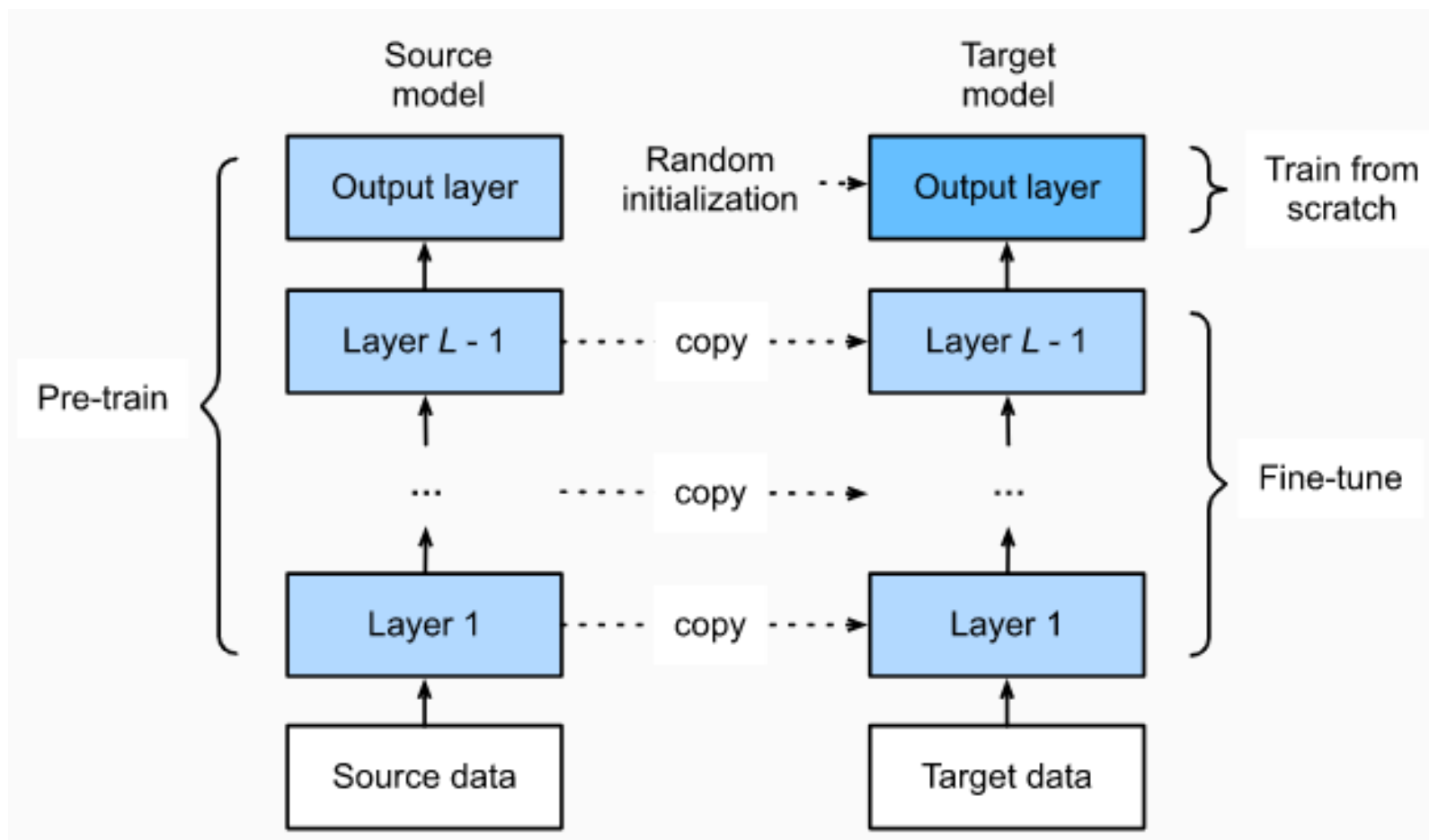
Computer Vision: Fine Tuning

Eng Teong Cheah

Contents

Fine Tuning

Fine Tuning



Hot Dog Recognition



```
%matplotlib inline
import d2l
from mxnet import gluon, init, np, npx
from mxnet.gluon import nn
import os
import zipfile

npx.set_np()
```

Get the Data Set



```
data_dir = '.. /data'
base_url = 'https://apache-mxnet.s3-accelerate.amazonaws.com/'
fname = gluon.utils.download(
    base_url + 'gluon/dataset/hotdog.zip',
    path=data_dir, sha1_hash='fba480ffa8aa7e0febbb511d181409f899b9baa5')
with zipfile.ZipFile(fname, 'r') as z:
    z.extractall(data_dir)
```

Get the Data Set

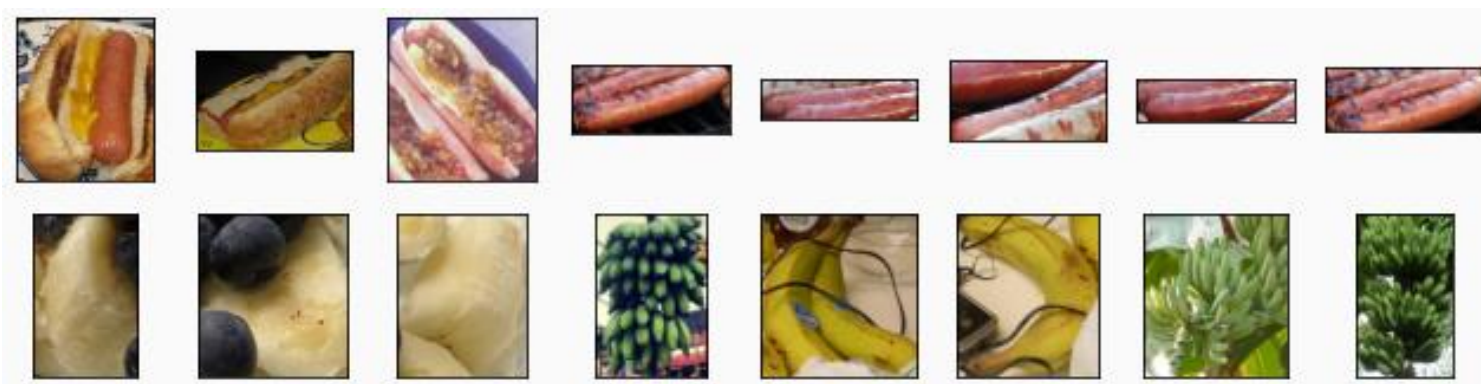


```
train_imgs = gluon.data.vision.ImageFolderDataset(  
    os.path.join(data_dir, 'hotdog/train'))  
test_imgs = gluon.data.vision.ImageFolderDataset(  
    os.path.join(data_dir, 'hotdog/test'))
```


Get the Data Set



```
hotdogs = [train_imgs[i][0] for i in range(8)]  
not_hotdogs = [train_imgs[-i - 1][0] for i in range(8)]  
d2l.show_images(hotdogs + not_hotdogs, 2, 8,  
scale=1.4);
```



Get the Data Set



```
# We specify the mean and variance of the three RGB channels to normalize the image channel.
normalize = gluon.data.vision.transforms.Normalize(
    [0.485, 0.456, 0.406], [0.229, 0.224, 0.225])

train_augs = gluon.data.vision.transforms.Compose([
    gluon.data.vision.transforms.RandomResizedCrop(224),
    gluon.data.vision.transforms.RandomFlipLeftRight(),
    gluon.data.vision.transforms.ToTensor(),
    normalize])

test_augs = gluon.data.vision.transforms.Compose([
    gluon.data.vision.transforms.Resize(256),
    gluon.data.vision.transforms.CenterCrop(224),
    gluon.data.vision.transforms.ToTensor(),
    normalize])
```


Define and Initialize the Model




```
pretrained_net = gluon.model_zoo.vision.resnet18_v2(pretrained=True)
```

Define and Initialize the Model




```
pretrained_net.output
```

Define and Initialize the Model



```
finetune_net = gluon.model_zoo.vision.resnet18_v2(classes=2)
finetune_net.features = pretrained_net.features
finetune_net.output.initialize(init.Xavier())
# The model parameters in output will be updated using a learning rate ten
# times greater
finetune_net.output.collect_params().setattr('lr_mult', 10)
```

Fine Tune the Model

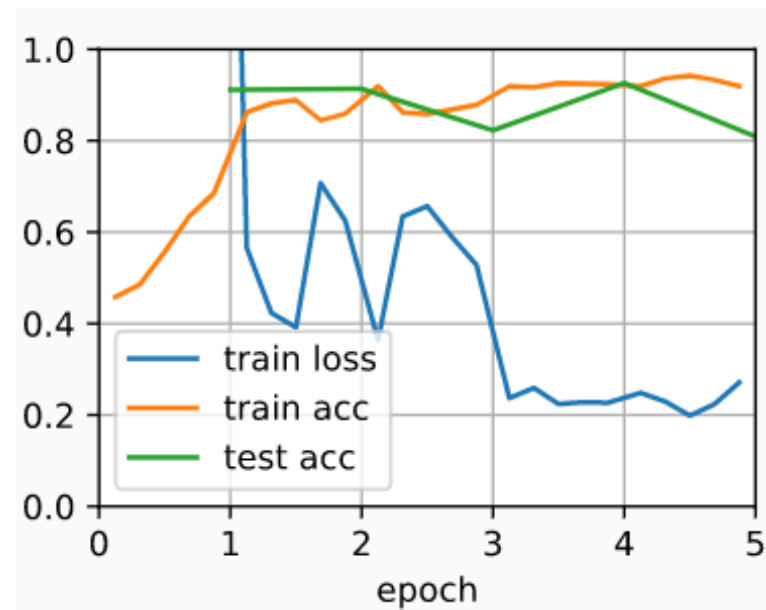


```
def train_fine_tuning(net, learning_rate, batch_size=128, num_epochs=5):
    train_iter = gluon.data.DataLoader(
        train_imgs.transform_first(train_augs), batch_size, shuffle=True)
    test_iter = gluon.data.DataLoader(
        test_imgs.transform_first(test_augs), batch_size)
    ctx = d2l.try_all_gpus()
    net.collect_params().reset_ctx(ctx)
    net.hybridize()
    loss = gluon.loss.SoftmaxCrossEntropyLoss()
    trainer = gluon.Trainer(net.collect_params(), 'sgd', {
        'learning_rate': learning_rate, 'wd': 0.001})
    d2l.train_ch12(net, train_iter, test_iter, loss, trainer, num_epochs,
ctx)
```

Fine Tune the Model



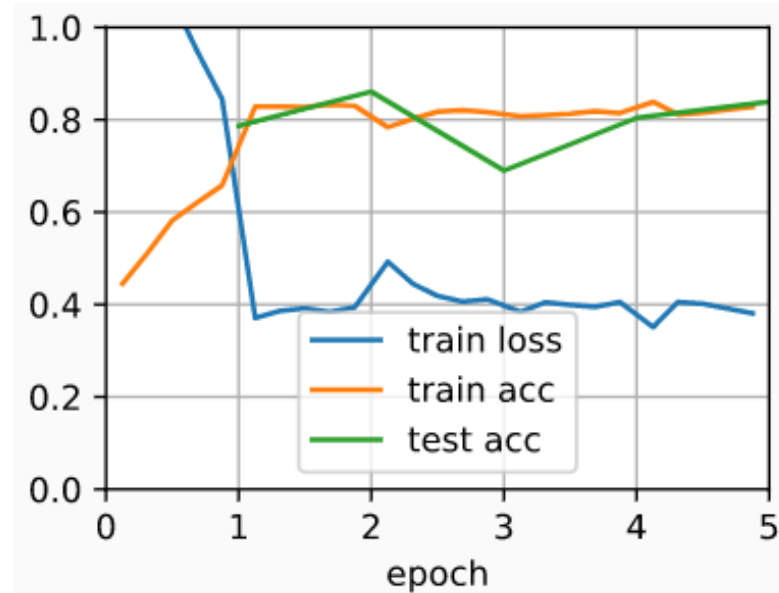
```
train_fine_tuning(finetune_net, 0.01)
```



Fine Tune the Model



```
scratch_net = gluon.model_zoo.vision.resnet18_v2(classes=2)
scratch_net.initialize(init=init.Xavier())
train_fine_tuning(scratch_net, 0.1)
```



Thank You !

Does anyone have any questions?

Twitter: @walkercet

Blog: <https://ceteongvanness.wordpress.com>

Resources

Dive into Deep Learning