

vgg

November 7, 2024

```
[1]: from models import *
      from utilities import Utilities as helper
      from utilities import AverageMeter
      from data_loader import trainloader
      from data_loader import testloader
```

=> Building model...

Files already downloaded and verified

Files already downloaded and verified

```
[2]: vgg_model = VGG16().cuda()
      # pre_best_prec = 0
```

```
[ ]: lr = 4e-3
      weight_decay = 1e-4
      epochs = 10
      criterion = nn.CrossEntropyLoss().cuda()

      vgg_optimizer = torch.optim.SGD(vgg_model.parameters(), lr=lr, momentum=0.9,
      ↪weight_decay=weight_decay)
      pre_best_prec = helper.train_model(vgg_model, model_name = "vgg", optimizer =
      ↪vgg_optimizer, trainloader=trainloader, testloader=testloader,
      ↪criterion=criterion, epochs=epochs, pre_best_prec=pre_best_prec)
```

```
[3]: from saveOutput import SaveOutput
      # vgg_first_block = next(vgg_model.modules()).features
      # vgg_save_output = SaveOutput()
      # SaveOutput.hook(vgg_first_block, torch.nn.Conv2d, vgg_save_output)
      helper.test_model(vgg_model, "vgg", testloader)
```

Test: [0/79] Time 0.397 (0.397) Loss 0.3243 (0.3243) Prec 92.188%
(92.188%)
* Prec 90.260%

```
[4]: vgg_model.features[0].weight.abs().sum()
```

```
[4]: tensor(327.6638, device='cuda:0', grad_fn=<SumBackward0>)
```

```
[ ]: # input_tensor = vgg_save_output.outputs[0][0]
# expect_out_tensor = vgg_save_output.outputs[1][0]

# con_lay = vgg_first_block[0]
# b_norm = vgg_first_block[1]
# relu = nn.ReLU(inplace=True).cuda()

# my_output = relu(b_norm(con_lay(input_tensor)))
# error = (my_output - expect_out_tensor).sum()
# error
```

```
[5]: cus_loss_model = VGG16().cuda()
```

```
[ ]: lr = 4e-2
weight_decay = 1e-4
epochs = 10
criterion = nn.CrossEntropyLoss().cuda()
cus_loss_optimizer = torch.optim.SGD(cus_loss_model.parameters(), lr=lr,
↪momentum=0.9, weight_decay=weight_decay)
```

```
[8]: pre_best_prec = 0

pretune_cl_model_name = "vgg_cus_loss_bf_tune"
```

```
[ ]: helper.train_model(cus_loss_model, model_name = pretune_cl_model_name,
↪optimizer = cus_loss_optimizer, trainloader=trainloader,
↪testloader=testloader, criterion=criterion, epochs=epochs,
↪pre_best_prec=pre_best_prec, reg_alpha = 0.5)
```

```
[9]: helper.test_model(cus_loss_model, pretune_cl_model_name, testloader)
```

```
Test: [0/79]    Time 0.208 (0.208)    Loss 0.4128 (0.4128)    Prec 86.719%
(86.719%)
* Prec 80.210%
```

```
[10]: cus_loss_model.features[0].weight.abs().sum()
```

```
[10]: tensor(315.9730, device='cuda:0', grad_fn=<SumBackward0>)
```

```
[11]: cus_loss_model_name = "vgg_cus_loss"
```

```
[ ]: lr = 4e-2
weight_decay = 1e-4

epochs = 100
```

```
helper.train_model(cus_loss_model, model_name = cus_loss_model_name, optimizer=  
    ↪ cus_loss_optimizer, trainloader=trainloader, testloader=testloader,  
    ↪ criterion=criterion, epochs=epochs, pre_best_prec=pre_best_prec, reg_alpha =  
    ↪ 0.6)
```

```
[12]: helper.test_model(cus_loss_model, cus_loss_model_name, testloader)
```

```
Test: [0/79]    Time 0.251 (0.251)    Loss 0.4430 (0.4430)    Prec 85.938%  
(85.938%)  
* Prec 83.970%
```

```
[13]: cus_loss_model.features[0].weight.abs().sum()
```

```
[13]: tensor(321.1955, device='cuda:0', grad_fn=<SumBackward0>)
```

```
[ ]:
```