[HW6_prob1]_VGG16_Quantization_aware_train_with_pruning

November 18, 2021

[1]:
```
import argparse
import os
import time
import shutil

import torch
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
import torch.backends.cudnn as cudnn

#from tensorboardX import SummaryWriter

import torchvision
import torchvision.transforms as transforms

from models import *

import os
os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"]="0"

global best_prec
use_gpu = torch.cuda.is_available()
print('=> Building model...')


batch_size = 128
model_name = "VGG16_quant"
model = VGG16_quant()
print(model)

normalize = transforms.Normalize(mean=[0.491, 0.482, 0.447], std=[0.247, 0.243,␣
 ↪0.262])


train_dataset = torchvision.datasets.CIFAR10(
```

```
    root='./data',
    train=True,
    download=True,
    transform=transforms.Compose([
        transforms.RandomCrop(32, padding=4),
        transforms.RandomHorizontalFlip(),
        transforms.ToTensor(),
        normalize,
    ]))
trainloader = torch.utils.data.DataLoader(train_dataset, batch_size=batch_size,␣
 ↪shuffle=True, num_workers=2)


test_dataset = torchvision.datasets.CIFAR10(
    root='./data',
    train=False,
    download=True,
    transform=transforms.Compose([
        transforms.ToTensor(),
        normalize,
    ]))

testloader = torch.utils.data.DataLoader(test_dataset, batch_size=batch_size,␣
 ↪shuffle=False, num_workers=2)


print_freq = 100 # every 100 batches, accuracy printed. Here, each batch␣
 ↪includes "batch_size" data points
# CIFAR10 has 50,000 training data, and 10,000 validation data.

def train(trainloader, model, criterion, optimizer, epoch):
    batch_time = AverageMeter()
    data_time = AverageMeter()
    losses = AverageMeter()
    top1 = AverageMeter()

    model.train()

    end = time.time()
    for i, (input, target) in enumerate(trainloader):
        # measure data loading time
        data_time.update(time.time() - end)

        input, target = input.cuda(), target.cuda()

        # compute output
        output = model(input)
```

```
        loss = criterion(output, target)

        # measure accuracy and record loss
        prec = accuracy(output, target)[0]
        losses.update(loss.item(), input.size(0))
        top1.update(prec.item(), input.size(0))

        # compute gradient and do SGD step
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        # measure elapsed time
        batch_time.update(time.time() - end)
        end = time.time()


        if i % print_freq == 0:
            print('Epoch: [{0}][{1}/{2}]\t'
                  'Time {batch_time.val:.3f} ({batch_time.avg:.3f})\t'
                  'Data {data_time.val:.3f} ({data_time.avg:.3f})\t'
                  'Loss {loss.val:.4f} ({loss.avg:.4f})\t'
                  'Prec {top1.val:.3f}% ({top1.avg:.3f}%)'.format(
                   epoch, i, len(trainloader), batch_time=batch_time,
                   data_time=data_time, loss=losses, top1=top1))



def validate(val_loader, model, criterion ):
    batch_time = AverageMeter()
    losses = AverageMeter()
    top1 = AverageMeter()

    # switch to evaluate mode
    model.eval()

    end = time.time()
    with torch.no_grad():
        for i, (input, target) in enumerate(val_loader):

            input, target = input.cuda(), target.cuda()

            # compute output
            output = model(input)
            loss = criterion(output, target)

            # measure accuracy and record loss
```

```
            prec = accuracy(output, target)[0]
            losses.update(loss.item(), input.size(0))
            top1.update(prec.item(), input.size(0))

            # measure elapsed time
            batch_time.update(time.time() - end)
            end = time.time()

            if i % print_freq == 0:  # This line shows how frequently print out the status. e.g., i%5 => every 5 batch, prints out
                print('Test: [{0}/{1}]\t'
                  'Time {batch_time.val:.3f} ({batch_time.avg:.3f})\t'
                  'Loss {loss.val:.4f} ({loss.avg:.4f})\t'
                  'Prec {top1.val:.3f}% ({top1.avg:.3f}%)'.format(
                   i, len(val_loader), batch_time=batch_time, loss=losses,
                   top1=top1))

    print(' * Prec {top1.avg:.3f}% '.format(top1=top1))
    return top1.avg


def accuracy(output, target, topk=(1,)):
    """Computes the precision@k for the specified values of k"""
    maxk = max(topk)
    batch_size = target.size(0)

    _, pred = output.topk(maxk, 1, True, True)
    pred = pred.t()
    correct = pred.eq(target.view(1, -1).expand_as(pred))

    res = []
    for k in topk:
        correct_k = correct[:k].view(-1).float().sum(0)
        res.append(correct_k.mul_(100.0 / batch_size))
    return res


class AverageMeter(object):
    """Computes and stores the average and current value"""
    def __init__(self):
        self.reset()

    def reset(self):
        self.val = 0
        self.avg = 0
        self.sum = 0
        self.count = 0
```

```
    def update(self, val, n=1):
        self.val = val
        self.sum += val * n
        self.count += n
        self.avg = self.sum / self.count


def save_checkpoint(state, is_best, fdir):
    filepath = os.path.join(fdir, 'checkpoint.pth')
    torch.save(state, filepath)
    if is_best:
        shutil.copyfile(filepath, os.path.join(fdir, 'model_best.pth.tar'))


def adjust_learning_rate(optimizer, epoch):
    """For resnet, the lr starts from 0.1, and is divided by 10 at 80 and 120␣
 ↪epochs"""
    adjust_list = [150, 225]
    if epoch in adjust_list:
        for param_group in optimizer.param_groups:
            param_group['lr'] = param_group['lr'] * 0.1

#model = nn.DataParallel(model).cuda()
#all_params = checkpoint['state_dict']
#model.load_state_dict(all_params, strict=False)
#criterion = nn.CrossEntropyLoss().cuda()
#validate(testloader, model, criterion)
```

```
=> Building model…
VGG_quant(
  (features): Sequential(
    (0): QuantConv2d(
      3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): QuantConv2d(
      64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
```

```
ceil_mode=False)
    (7): QuantConv2d(
      64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (8): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (9): ReLU(inplace=True)
    (10): QuantConv2d(
      128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (11): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (12): ReLU(inplace=True)
    (13): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (14): QuantConv2d(
      128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (15): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (16): ReLU(inplace=True)
    (17): QuantConv2d(
      256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (18): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (19): ReLU(inplace=True)
    (20): QuantConv2d(
      256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (21): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (24): QuantConv2d(
      256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (25): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (26): ReLU(inplace=True)
    (27): QuantConv2d(
```

```
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (28): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (29): ReLU(inplace=True)
    (30): QuantConv2d(
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (31): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (32): ReLU(inplace=True)
    (33): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (34): QuantConv2d(
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (35): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (36): ReLU(inplace=True)
    (37): QuantConv2d(
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (38): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (39): ReLU(inplace=True)
    (40): QuantConv2d(
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (41): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (42): ReLU(inplace=True)
    (43): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (44): AvgPool2d(kernel_size=1, stride=1, padding=0)
  )
  (classifier): Linear(in_features=512, out_features=10, bias=True)
)
Files already downloaded and verified
Files already downloaded and verified
```

[3]:
```
lr = 4e-2
weight_decay = 1e-4
```

```
epochs = 500
best_prec = 0

#model = nn.DataParallel(model).cuda()
model.cuda()
criterion = nn.CrossEntropyLoss().cuda()
optimizer = torch.optim.SGD(model.parameters(), lr=lr, momentum=0.9,␣
 ↪weight_decay=weight_decay)
#cudnn.benchmark = True

if not os.path.exists('result'):
    os.makedirs('result')
fdir = 'result/'+str(model_name)
if not os.path.exists(fdir):
    os.makedirs(fdir)


for epoch in range(0, epochs):
    adjust_learning_rate(optimizer, epoch)

    train(trainloader, model, criterion, optimizer, epoch)

    # evaluate on test set
    print("Validation starts")
    prec = validate(testloader, model, criterion)

    # remember best precision and save checkpoint
    is_best = prec > best_prec
    best_prec = max(prec,best_prec)
    print('best acc: {:1f}'.format(best_prec))
    save_checkpoint({
        'epoch': epoch + 1,
        'state_dict': model.state_dict(),
        'best_prec': best_prec,
        'optimizer': optimizer.state_dict(),
    }, is_best, fdir)
```

```
Epoch: [0][0/391]       Time 0.282 (0.282)      Data 0.211 (0.211)      Loss
0.5377 (0.5377)    Prec 82.812% (82.812%)
Epoch: [0][100/391]     Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.5533 (0.6417)    Prec 77.344% (77.645%)
Epoch: [0][200/391]     Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.4892 (0.6495)    Prec 83.594% (77.694%)
Epoch: [0][300/391]     Time 0.097 (0.102)      Data 0.002 (0.003)      Loss
0.6170 (0.6485)    Prec 74.219% (77.762%)
Validation starts
Test: [0/79]    Time 0.213 (0.213)      Loss 0.7371 (0.7371)    Prec 75.781%
```

```
(75.781%)
 * Prec 73.550%
best acc: 73.550000
Epoch: [1][0/391]       Time 0.284 (0.284)      Data 0.216 (0.216)      Loss
0.6389 (0.6389)    Prec 75.781% (75.781%)
Epoch: [1][100/391]     Time 0.098 (0.103)      Data 0.002 (0.005)      Loss
0.4312 (0.6016)    Prec 83.594% (79.463%)
Epoch: [1][200/391]     Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.6122 (0.6002)    Prec 78.125% (79.660%)
Epoch: [1][300/391]     Time 0.102 (0.102)      Data 0.005 (0.003)      Loss
0.4884 (0.6016)    Prec 83.594% (79.597%)
Validation starts
Test: [0/79]    Time 0.227 (0.227)      Loss 0.5547 (0.5547)    Prec 81.250%
(81.250%)
 * Prec 79.260%
best acc: 79.260000
Epoch: [2][0/391]       Time 0.277 (0.277)      Data 0.210 (0.210)      Loss
0.4609 (0.4609)    Prec 85.156% (85.156%)
Epoch: [2][100/391]     Time 0.102 (0.103)      Data 0.002 (0.004)      Loss
0.4873 (0.5411)    Prec 82.812% (81.351%)
Epoch: [2][200/391]     Time 0.103 (0.102)      Data 0.003 (0.003)      Loss
0.7440 (0.5496)    Prec 76.562% (81.339%)
Epoch: [2][300/391]     Time 0.102 (0.102)      Data 0.003 (0.003)      Loss
0.6303 (0.5526)    Prec 75.781% (81.170%)
Validation starts
Test: [0/79]    Time 0.201 (0.201)      Loss 0.5276 (0.5276)    Prec 83.594%
(83.594%)
 * Prec 79.850%
best acc: 79.850000
Epoch: [3][0/391]       Time 0.311 (0.311)      Data 0.244 (0.244)      Loss
0.5758 (0.5758)    Prec 77.344% (77.344%)
Epoch: [3][100/391]     Time 0.092 (0.103)      Data 0.002 (0.004)      Loss
0.4368 (0.5236)    Prec 86.719% (82.178%)
Epoch: [3][200/391]     Time 0.098 (0.102)      Data 0.002 (0.003)      Loss
0.3808 (0.5242)    Prec 88.281% (82.167%)
Epoch: [3][300/391]     Time 0.099 (0.102)      Data 0.003 (0.003)      Loss
0.3389 (0.5237)    Prec 90.625% (82.125%)
Validation starts
Test: [0/79]    Time 0.255 (0.255)      Loss 0.5481 (0.5481)    Prec 79.688%
(79.688%)
 * Prec 80.280%
best acc: 80.280000
Epoch: [4][0/391]       Time 0.263 (0.263)      Data 0.197 (0.197)      Loss
0.4435 (0.4435)    Prec 83.594% (83.594%)
Epoch: [4][100/391]     Time 0.099 (0.102)      Data 0.002 (0.004)      Loss
0.5022 (0.4867)    Prec 81.250% (83.640%)
Epoch: [4][200/391]     Time 0.103 (0.102)      Data 0.002 (0.003)      Loss
0.3102 (0.4916)    Prec 92.969% (83.403%)
```

```
Epoch: [4][300/391]     Time 0.101 (0.101)      Data 0.002 (0.002)      Loss
0.3325 (0.4963)    Prec 87.500% (83.142%)
Validation starts
Test: [0/79]    Time 0.216 (0.216)      Loss 0.7143 (0.7143)    Prec 78.125%
(78.125%)
 * Prec 77.720%
best acc: 80.280000
Epoch: [5][0/391]       Time 0.243 (0.243)      Data 0.176 (0.176)      Loss
0.3835 (0.3835)    Prec 86.719% (86.719%)
Epoch: [5][100/391]     Time 0.101 (0.102)      Data 0.002 (0.004)      Loss
0.4491 (0.4533)    Prec 85.156% (84.274%)
Epoch: [5][200/391]     Time 0.103 (0.102)      Data 0.001 (0.003)      Loss
0.4759 (0.4525)    Prec 83.594% (84.492%)
Epoch: [5][300/391]     Time 0.100 (0.101)      Data 0.002 (0.002)      Loss
0.4907 (0.4573)    Prec 82.812% (84.354%)
Validation starts
Test: [0/79]    Time 0.243 (0.243)      Loss 0.4038 (0.4038)    Prec 89.844%
(89.844%)
 * Prec 83.610%
best acc: 83.610000
Epoch: [6][0/391]       Time 0.296 (0.296)      Data 0.226 (0.226)      Loss
0.3589 (0.3589)    Prec 85.938% (85.938%)
Epoch: [6][100/391]     Time 0.102 (0.102)      Data 0.002 (0.004)      Loss
0.2951 (0.4206)    Prec 89.844% (85.388%)
Epoch: [6][200/391]     Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.4534 (0.4304)    Prec 81.250% (85.110%)
Epoch: [6][300/391]     Time 0.102 (0.101)      Data 0.002 (0.002)      Loss
0.3900 (0.4319)    Prec 89.844% (85.143%)
Validation starts
Test: [0/79]    Time 0.226 (0.226)      Loss 0.5454 (0.5454)    Prec 78.906%
(78.906%)
 * Prec 83.350%
best acc: 83.610000
Epoch: [7][0/391]       Time 0.279 (0.279)      Data 0.212 (0.212)      Loss
0.3475 (0.3475)    Prec 89.844% (89.844%)
Epoch: [7][100/391]     Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.4305 (0.4150)    Prec 83.594% (85.767%)
Epoch: [7][200/391]     Time 0.104 (0.102)      Data 0.001 (0.003)      Loss
0.4547 (0.4104)    Prec 85.156% (85.922%)
Epoch: [7][300/391]     Time 0.101 (0.101)      Data 0.001 (0.003)      Loss
0.4453 (0.4113)    Prec 84.375% (85.938%)
Validation starts
Test: [0/79]    Time 0.227 (0.227)      Loss 0.5376 (0.5376)    Prec 84.375%
(84.375%)
 * Prec 82.520%
best acc: 83.610000
Epoch: [8][0/391]       Time 0.323 (0.323)      Data 0.255 (0.255)      Loss
0.3857 (0.3857)    Prec 86.719% (86.719%)
```

```
Epoch: [8][100/391]     Time 0.100 (0.103)      Data 0.002 (0.005)      Loss
0.4141 (0.3916)    Prec 87.500% (86.959%)
Epoch: [8][200/391]     Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.4124 (0.3920)    Prec 82.812% (86.820%)
Epoch: [8][300/391]     Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.3618 (0.3934)    Prec 88.281% (86.625%)
Validation starts
Test: [0/79]    Time 0.238 (0.238)      Loss 0.5118 (0.5118)    Prec 83.594%
(83.594%)
 * Prec 83.540%
best acc: 83.610000
Epoch: [9][0/391]       Time 0.287 (0.287)      Data 0.220 (0.220)      Loss
0.3427 (0.3427)    Prec 89.062% (89.062%)
Epoch: [9][100/391]     Time 0.101 (0.102)      Data 0.002 (0.004)      Loss
0.3420 (0.3754)    Prec 88.281% (87.369%)
Epoch: [9][200/391]     Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.4908 (0.3719)    Prec 82.031% (87.570%)
Epoch: [9][300/391]     Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.4253 (0.3739)    Prec 85.938% (87.401%)
Validation starts
Test: [0/79]    Time 0.295 (0.295)      Loss 0.3675 (0.3675)    Prec 88.281%
(88.281%)
 * Prec 84.220%
best acc: 84.220000
Epoch: [10][0/391]      Time 0.316 (0.316)      Data 0.248 (0.248)      Loss
0.4078 (0.4078)    Prec 86.719% (86.719%)
Epoch: [10][100/391]    Time 0.099 (0.103)      Data 0.002 (0.005)      Loss
0.4730 (0.3541)    Prec 85.938% (88.065%)
Epoch: [10][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.5065 (0.3573)    Prec 84.375% (87.741%)
Epoch: [10][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.2518 (0.3551)    Prec 91.406% (87.830%)
Validation starts
Test: [0/79]    Time 0.219 (0.219)      Loss 0.4231 (0.4231)    Prec 84.375%
(84.375%)
 * Prec 84.930%
best acc: 84.930000
Epoch: [11][0/391]      Time 0.262 (0.262)      Data 0.194 (0.194)      Loss
0.2572 (0.2572)    Prec 90.625% (90.625%)
Epoch: [11][100/391]    Time 0.099 (0.103)      Data 0.002 (0.004)      Loss
0.2959 (0.3376)    Prec 89.844% (88.653%)
Epoch: [11][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.4843 (0.3342)    Prec 83.594% (88.588%)
Epoch: [11][300/391]    Time 0.101 (0.101)      Data 0.002 (0.002)      Loss
0.2842 (0.3414)    Prec 85.938% (88.258%)
Validation starts
Test: [0/79]    Time 0.208 (0.208)      Loss 0.3954 (0.3954)    Prec 87.500%
(87.500%)
```

```
 * Prec 86.190%
best acc: 86.190000
Epoch: [12][0/391]      Time 0.276 (0.276)      Data 0.206 (0.206)      Loss
0.3399 (0.3399)    Prec 89.062% (89.062%)
Epoch: [12][100/391]    Time 0.096 (0.103)      Data 0.004 (0.004)      Loss
0.4144 (0.3273)    Prec 85.938% (88.877%)
Epoch: [12][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.2321 (0.3235)    Prec 91.406% (89.218%)
Epoch: [12][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.2754 (0.3231)    Prec 90.625% (89.151%)
Validation starts
Test: [0/79]    Time 0.212 (0.212)      Loss 0.4029 (0.4029)    Prec 85.938%
(85.938%)
 * Prec 85.380%
best acc: 86.190000
Epoch: [13][0/391]      Time 0.332 (0.332)      Data 0.265 (0.265)      Loss
0.4308 (0.4308)    Prec 88.281% (88.281%)
Epoch: [13][100/391]    Time 0.101 (0.103)      Data 0.002 (0.005)      Loss
0.3340 (0.3130)    Prec 85.938% (89.372%)
Epoch: [13][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.2222 (0.3063)    Prec 94.531% (89.583%)
Epoch: [13][300/391]    Time 0.102 (0.102)      Data 0.001 (0.003)      Loss
0.3455 (0.3110)    Prec 89.062% (89.371%)
Validation starts
Test: [0/79]    Time 0.232 (0.232)      Loss 0.3713 (0.3713)    Prec 88.281%
(88.281%)
 * Prec 84.410%
best acc: 86.190000
Epoch: [14][0/391]      Time 0.263 (0.263)      Data 0.200 (0.200)      Loss
0.3019 (0.3019)    Prec 86.719% (86.719%)
Epoch: [14][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.3035 (0.2874)    Prec 92.188% (90.192%)
Epoch: [14][200/391]    Time 0.103 (0.102)      Data 0.002 (0.003)      Loss
0.3060 (0.2904)    Prec 89.062% (90.034%)
Epoch: [14][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.3228 (0.2946)    Prec 89.844% (89.849%)
Validation starts
Test: [0/79]    Time 0.237 (0.237)      Loss 0.3531 (0.3531)    Prec 88.281%
(88.281%)
 * Prec 85.260%
best acc: 86.190000
Epoch: [15][0/391]      Time 0.276 (0.276)      Data 0.208 (0.208)      Loss
0.2787 (0.2787)    Prec 88.281% (88.281%)
Epoch: [15][100/391]    Time 0.101 (0.103)      Data 0.003 (0.004)      Loss
0.3308 (0.2765)    Prec 88.281% (90.548%)
Epoch: [15][200/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.2620 (0.2746)    Prec 91.406% (90.559%)
Epoch: [15][300/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
```

```
0.3121 (0.2775)    Prec 89.062% (90.500%)
Validation starts
Test: [0/79]    Time 0.269 (0.269)      Loss 0.3287 (0.3287)    Prec 89.062%
(89.062%)
 * Prec 83.550%
best acc: 86.190000
Epoch: [16][0/391]      Time 0.317 (0.317)      Data 0.247 (0.247)      Loss
0.3055 (0.3055)    Prec 89.844% (89.844%)
Epoch: [16][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.2406 (0.2630)    Prec 92.188% (91.019%)
Epoch: [16][200/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.3475 (0.2751)    Prec 88.281% (90.722%)
Epoch: [16][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.2542 (0.2747)    Prec 92.188% (90.721%)
Validation starts
Test: [0/79]    Time 0.203 (0.203)      Loss 0.2507 (0.2507)    Prec 91.406%
(91.406%)
 * Prec 86.220%
best acc: 86.220000
Epoch: [17][0/391]      Time 0.291 (0.291)      Data 0.225 (0.225)      Loss
0.2728 (0.2728)    Prec 87.500% (87.500%)
Epoch: [17][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.2652 (0.2530)    Prec 89.062% (91.484%)
Epoch: [17][200/391]    Time 0.099 (0.102)      Data 0.001 (0.003)      Loss
0.2975 (0.2546)    Prec 88.281% (91.208%)
Epoch: [17][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2433 (0.2587)    Prec 89.062% (91.022%)
Validation starts
Test: [0/79]    Time 0.215 (0.215)      Loss 0.4743 (0.4743)    Prec 85.156%
(85.156%)
 * Prec 84.330%
best acc: 86.220000
Epoch: [18][0/391]      Time 0.266 (0.266)      Data 0.198 (0.198)      Loss
0.3471 (0.3471)    Prec 89.062% (89.062%)
Epoch: [18][100/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.2181 (0.2347)    Prec 92.969% (92.087%)
Epoch: [18][200/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.2772 (0.2463)    Prec 88.281% (91.597%)
Epoch: [18][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2080 (0.2487)    Prec 91.406% (91.492%)
Validation starts
Test: [0/79]    Time 0.250 (0.250)      Loss 0.2267 (0.2267)    Prec 90.625%
(90.625%)
 * Prec 87.370%
best acc: 87.370000
Epoch: [19][0/391]      Time 0.319 (0.319)      Data 0.252 (0.252)      Loss
0.2725 (0.2725)    Prec 89.062% (89.062%)
Epoch: [19][100/391]    Time 0.102 (0.103)      Data 0.001 (0.004)      Loss
```

```
0.1504 (0.2352)    Prec 94.531% (91.878%)
Epoch: [19][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.3341 (0.2343)    Prec 87.500% (91.931%)
Epoch: [19][300/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.2970 (0.2385)    Prec 90.625% (91.814%)
Validation starts
Test: [0/79]    Time 0.204 (0.204)      Loss 0.2730 (0.2730)    Prec 93.750%
(93.750%)
 * Prec 86.340%
best acc: 87.370000
Epoch: [20][0/391]      Time 0.290 (0.290)      Data 0.225 (0.225)      Loss
0.1779 (0.1779)    Prec 93.750% (93.750%)
Epoch: [20][100/391]    Time 0.100 (0.103)      Data 0.001 (0.004)      Loss
0.3500 (0.2268)    Prec 89.844% (92.188%)
Epoch: [20][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.3232 (0.2287)    Prec 86.719% (92.156%)
Epoch: [20][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.2676 (0.2319)    Prec 88.281% (91.967%)
Validation starts
Test: [0/79]    Time 0.232 (0.232)      Loss 0.2940 (0.2940)    Prec 90.625%
(90.625%)
 * Prec 86.880%
best acc: 87.370000
Epoch: [21][0/391]      Time 0.285 (0.285)      Data 0.219 (0.219)      Loss
0.1973 (0.1973)    Prec 92.969% (92.969%)
Epoch: [21][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.2149 (0.2198)    Prec 93.750% (92.567%)
Epoch: [21][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.2090 (0.2232)    Prec 90.625% (92.440%)
Epoch: [21][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2589 (0.2248)    Prec 90.625% (92.252%)
Validation starts
Test: [0/79]    Time 0.229 (0.229)      Loss 0.2994 (0.2994)    Prec 88.281%
(88.281%)
 * Prec 87.140%
best acc: 87.370000
Epoch: [22][0/391]      Time 0.307 (0.307)      Data 0.240 (0.240)      Loss
0.1721 (0.1721)    Prec 93.750% (93.750%)
Epoch: [22][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.1747 (0.2019)    Prec 94.531% (93.077%)
Epoch: [22][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.2292 (0.2120)    Prec 93.750% (92.778%)
Epoch: [22][300/391]    Time 0.100 (0.101)      Data 0.002 (0.002)      Loss
0.1709 (0.2171)    Prec 94.531% (92.525%)
Validation starts
Test: [0/79]    Time 0.238 (0.238)      Loss 0.2528 (0.2528)    Prec 92.188%
(92.188%)
 * Prec 87.300%
```

```
best acc: 87.370000
Epoch: [23][0/391]      Time 0.267 (0.267)      Data 0.200 (0.200)      Loss
0.2551 (0.2551)    Prec 91.406% (91.406%)
Epoch: [23][100/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.2824 (0.1972)    Prec 89.062% (93.093%)
Epoch: [23][200/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2086 (0.2067)    Prec 90.625% (92.806%)
Epoch: [23][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2095 (0.2085)    Prec 93.750% (92.746%)
Validation starts
Test: [0/79]    Time 0.196 (0.196)      Loss 0.3409 (0.3409)    Prec 89.844%
(89.844%)
 * Prec 86.240%
best acc: 87.370000
Epoch: [24][0/391]      Time 0.278 (0.278)      Data 0.211 (0.211)      Loss
0.1099 (0.1099)    Prec 96.094% (96.094%)
Epoch: [24][100/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.2014 (0.1972)    Prec 93.750% (93.116%)
Epoch: [24][200/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.2162 (0.2007)    Prec 92.969% (93.046%)
Epoch: [24][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2133 (0.2044)    Prec 92.188% (92.925%)
Validation starts
Test: [0/79]    Time 0.279 (0.279)      Loss 0.2009 (0.2009)    Prec 90.625%
(90.625%)
 * Prec 87.550%
best acc: 87.550000
Epoch: [25][0/391]      Time 0.329 (0.329)      Data 0.263 (0.263)      Loss
0.2002 (0.2002)    Prec 93.750% (93.750%)
Epoch: [25][100/391]    Time 0.100 (0.103)      Data 0.001 (0.004)      Loss
0.2111 (0.1828)    Prec 92.969% (93.673%)
Epoch: [25][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1859 (0.1864)    Prec 94.531% (93.478%)
Epoch: [25][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.1968 (0.1870)    Prec 92.969% (93.483%)
Validation starts
Test: [0/79]    Time 0.215 (0.215)      Loss 0.2358 (0.2358)    Prec 90.625%
(90.625%)
 * Prec 87.710%
best acc: 87.710000
Epoch: [26][0/391]      Time 0.334 (0.334)      Data 0.269 (0.269)      Loss
0.2691 (0.2691)    Prec 88.281% (88.281%)
Epoch: [26][100/391]    Time 0.099 (0.103)      Data 0.002 (0.004)      Loss
0.1242 (0.1815)    Prec 95.312% (93.611%)
Epoch: [26][200/391]    Time 0.102 (0.102)      Data 0.003 (0.003)      Loss
0.1393 (0.1861)    Prec 93.750% (93.458%)
Epoch: [26][300/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.2790 (0.1910)    Prec 92.188% (93.278%)
```

```
Validation starts
Test: [0/79]    Time 0.246 (0.246)      Loss 0.3038 (0.3038)    Prec 88.281%
(88.281%)
 * Prec 88.550%
best acc: 88.550000
Epoch: [27][0/391]      Time 0.317 (0.317)      Data 0.250 (0.250)      Loss
0.1642 (0.1642)    Prec 94.531% (94.531%)
Epoch: [27][100/391]    Time 0.101 (0.103)      Data 0.002 (0.005)      Loss
0.0566 (0.1785)    Prec 98.438% (93.851%)
Epoch: [27][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1501 (0.1815)    Prec 96.094% (93.723%)
Epoch: [27][300/391]    Time 0.102 (0.102)      Data 0.001 (0.003)      Loss
0.2006 (0.1832)    Prec 93.750% (93.561%)
Validation starts
Test: [0/79]    Time 0.234 (0.234)      Loss 0.3231 (0.3231)    Prec 90.625%
(90.625%)
 * Prec 87.620%
best acc: 88.550000
Epoch: [28][0/391]      Time 0.284 (0.284)      Data 0.219 (0.219)      Loss
0.1524 (0.1524)    Prec 92.969% (92.969%)
Epoch: [28][100/391]    Time 0.104 (0.103)      Data 0.002 (0.004)      Loss
0.1438 (0.1625)    Prec 93.750% (94.431%)
Epoch: [28][200/391]    Time 0.102 (0.102)      Data 0.003 (0.003)      Loss
0.1391 (0.1714)    Prec 94.531% (94.108%)
Epoch: [28][300/391]    Time 0.103 (0.102)      Data 0.002 (0.003)      Loss
0.1349 (0.1761)    Prec 94.531% (93.895%)
Validation starts
Test: [0/79]    Time 0.259 (0.259)      Loss 0.2582 (0.2582)    Prec 90.625%
(90.625%)
 * Prec 88.490%
best acc: 88.550000
Epoch: [29][0/391]      Time 0.310 (0.310)      Data 0.242 (0.242)      Loss
0.1276 (0.1276)    Prec 96.094% (96.094%)
Epoch: [29][100/391]    Time 0.103 (0.103)      Data 0.003 (0.005)      Loss
0.2144 (0.1580)    Prec 91.406% (94.407%)
Epoch: [29][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.1134 (0.1647)    Prec 94.531% (94.267%)
Epoch: [29][300/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.1913 (0.1683)    Prec 95.312% (94.142%)
Validation starts
Test: [0/79]    Time 0.200 (0.200)      Loss 0.2557 (0.2557)    Prec 90.625%
(90.625%)
 * Prec 87.290%
best acc: 88.550000
Epoch: [30][0/391]      Time 0.268 (0.268)      Data 0.201 (0.201)      Loss
0.1034 (0.1034)    Prec 95.312% (95.312%)
Epoch: [30][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.1220 (0.1572)    Prec 94.531% (94.570%)
```

```
Epoch: [30][200/391]    Time 0.103 (0.102)      Data 0.003 (0.003)      Loss
0.2320 (0.1662)    Prec 93.750% (94.306%)
Epoch: [30][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.2170 (0.1693)    Prec 92.188% (94.124%)
Validation starts
Test: [0/79]    Time 0.213 (0.213)      Loss 0.4514 (0.4514)    Prec 87.500%
(87.500%)
 * Prec 86.780%
best acc: 88.550000
Epoch: [31][0/391]      Time 0.298 (0.298)      Data 0.232 (0.232)      Loss
0.2141 (0.2141)    Prec 92.969% (92.969%)
Epoch: [31][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.1400 (0.1652)    Prec 93.750% (93.967%)
Epoch: [31][200/391]    Time 0.103 (0.102)      Data 0.002 (0.003)      Loss
0.2012 (0.1582)    Prec 93.750% (94.368%)
Epoch: [31][300/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1612 (0.1622)    Prec 93.750% (94.282%)
Validation starts
Test: [0/79]    Time 0.250 (0.250)      Loss 0.2179 (0.2179)    Prec 92.188%
(92.188%)
 * Prec 88.540%
best acc: 88.550000
Epoch: [32][0/391]      Time 0.300 (0.300)      Data 0.233 (0.233)      Loss
0.1193 (0.1193)    Prec 96.875% (96.875%)
Epoch: [32][100/391]    Time 0.100 (0.102)      Data 0.001 (0.004)      Loss
0.2015 (0.1430)    Prec 90.625% (95.150%)
Epoch: [32][200/391]    Time 0.100 (0.101)      Data 0.001 (0.003)      Loss
0.2964 (0.1509)    Prec 88.281% (94.862%)
Epoch: [32][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.2145 (0.1530)    Prec 92.969% (94.796%)
Validation starts
Test: [0/79]    Time 0.255 (0.255)      Loss 0.3048 (0.3048)    Prec 88.281%
(88.281%)
 * Prec 88.230%
best acc: 88.550000
Epoch: [33][0/391]      Time 0.290 (0.290)      Data 0.225 (0.225)      Loss
0.1375 (0.1375)    Prec 93.750% (93.750%)
Epoch: [33][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.0882 (0.1388)    Prec 96.875% (95.003%)
Epoch: [33][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.1568 (0.1501)    Prec 93.750% (94.796%)
Epoch: [33][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.1481 (0.1532)    Prec 94.531% (94.695%)
Validation starts
Test: [0/79]    Time 0.226 (0.226)      Loss 0.3482 (0.3482)    Prec 89.844%
(89.844%)
 * Prec 88.060%
best acc: 88.550000
```

```
Epoch: [34][0/391]      Time 0.260 (0.260)      Data 0.194 (0.194)      Loss
0.3162 (0.3162)    Prec 88.281% (88.281%)
Epoch: [34][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.1778 (0.1413)    Prec 91.406% (95.220%)
Epoch: [34][200/391]    Time 0.104 (0.102)      Data 0.002 (0.003)      Loss
0.1356 (0.1406)    Prec 96.094% (95.176%)
Epoch: [34][300/391]    Time 0.103 (0.102)      Data 0.001 (0.003)      Loss
0.1116 (0.1425)    Prec 96.094% (95.087%)
Validation starts
Test: [0/79]    Time 0.224 (0.224)      Loss 0.3594 (0.3594)    Prec 92.188%
(92.188%)
 * Prec 88.230%
best acc: 88.550000
Epoch: [35][0/391]      Time 0.301 (0.301)      Data 0.234 (0.234)      Loss
0.2284 (0.2284)    Prec 90.625% (90.625%)
Epoch: [35][100/391]    Time 0.100 (0.103)      Data 0.001 (0.004)      Loss
0.1387 (0.1259)    Prec 95.312% (95.537%)
Epoch: [35][200/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.1064 (0.1323)    Prec 95.312% (95.305%)
Epoch: [35][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.1817 (0.1366)    Prec 96.094% (95.198%)
Validation starts
Test: [0/79]    Time 0.219 (0.219)      Loss 0.2770 (0.2770)    Prec 90.625%
(90.625%)
 * Prec 87.480%
best acc: 88.550000
Epoch: [36][0/391]      Time 0.339 (0.339)      Data 0.275 (0.275)      Loss
0.1876 (0.1876)    Prec 92.188% (92.188%)
Epoch: [36][100/391]    Time 0.100 (0.104)      Data 0.002 (0.005)      Loss
0.1115 (0.1278)    Prec 95.312% (95.490%)
Epoch: [36][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.1588 (0.1318)    Prec 93.750% (95.278%)
Epoch: [36][300/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.1169 (0.1384)    Prec 94.531% (95.087%)
Validation starts
Test: [0/79]    Time 0.243 (0.243)      Loss 0.2451 (0.2451)    Prec 90.625%
(90.625%)
 * Prec 87.420%
best acc: 88.550000
Epoch: [37][0/391]      Time 0.346 (0.346)      Data 0.277 (0.277)      Loss
0.1657 (0.1657)    Prec 95.312% (95.312%)
Epoch: [37][100/391]    Time 0.100 (0.104)      Data 0.002 (0.005)      Loss
0.1013 (0.1299)    Prec 96.875% (95.537%)
Epoch: [37][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.1398 (0.1303)    Prec 95.312% (95.421%)
Epoch: [37][300/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.1720 (0.1371)    Prec 95.312% (95.193%)
Validation starts
```

```
Test: [0/79]    Time 0.237 (0.237)      Loss 0.3997 (0.3997)    Prec 87.500%
(87.500%)
 * Prec 87.950%
best acc: 88.550000
Epoch: [38][0/391]      Time 0.288 (0.288)      Data 0.220 (0.220)      Loss
0.1557 (0.1557)    Prec 94.531% (94.531%)
Epoch: [38][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.0755 (0.1268)    Prec 98.438% (95.606%)
Epoch: [38][200/391]    Time 0.099 (0.102)      Data 0.003 (0.003)      Loss
0.1873 (0.1309)    Prec 93.750% (95.441%)
Epoch: [38][300/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.0568 (0.1325)    Prec 97.656% (95.406%)
Validation starts
Test: [0/79]    Time 0.216 (0.216)      Loss 0.2993 (0.2993)    Prec 91.406%
(91.406%)
 * Prec 89.270%
best acc: 89.270000
Epoch: [39][0/391]      Time 0.321 (0.321)      Data 0.256 (0.256)      Loss
0.1162 (0.1162)    Prec 93.750% (93.750%)
Epoch: [39][100/391]    Time 0.100 (0.103)      Data 0.001 (0.005)      Loss
0.1314 (0.1257)    Prec 96.094% (95.575%)
Epoch: [39][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0995 (0.1274)    Prec 95.312% (95.421%)
Epoch: [39][300/391]    Time 0.101 (0.102)      Data 0.001 (0.002)      Loss
0.1221 (0.1289)    Prec 95.312% (95.484%)
Validation starts
Test: [0/79]    Time 0.245 (0.245)      Loss 0.2878 (0.2878)    Prec 90.625%
(90.625%)
 * Prec 87.800%
best acc: 89.270000
Epoch: [40][0/391]      Time 0.288 (0.288)      Data 0.222 (0.222)      Loss
0.1303 (0.1303)    Prec 95.312% (95.312%)
Epoch: [40][100/391]    Time 0.100 (0.103)      Data 0.001 (0.004)      Loss
0.1607 (0.1206)    Prec 93.750% (95.784%)
Epoch: [40][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0582 (0.1197)    Prec 98.438% (95.911%)
Epoch: [40][300/391]    Time 0.103 (0.101)      Data 0.001 (0.002)      Loss
0.1268 (0.1259)    Prec 95.312% (95.642%)
Validation starts
Test: [0/79]    Time 0.313 (0.313)      Loss 0.1302 (0.1302)    Prec 94.531%
(94.531%)
 * Prec 88.670%
best acc: 89.270000
Epoch: [41][0/391]      Time 0.314 (0.314)      Data 0.249 (0.249)      Loss
0.0677 (0.0677)    Prec 98.438% (98.438%)
Epoch: [41][100/391]    Time 0.100 (0.103)      Data 0.001 (0.004)      Loss
0.1645 (0.1171)    Prec 93.750% (95.916%)
Epoch: [41][200/391]    Time 0.102 (0.102)      Data 0.001 (0.003)      Loss
```

```
0.1677 (0.1177)    Prec 95.312% (95.919%)
Epoch: [41][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.0973 (0.1194)    Prec 96.875% (95.816%)
Validation starts
Test: [0/79]    Time 0.210 (0.210)      Loss 0.2727 (0.2727)    Prec 91.406%
(91.406%)
 * Prec 88.900%
best acc: 89.270000
Epoch: [42][0/391]      Time 0.289 (0.289)      Data 0.223 (0.223)      Loss
0.1052 (0.1052)    Prec 95.312% (95.312%)
Epoch: [42][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.1362 (0.1121)    Prec 95.312% (95.993%)
Epoch: [42][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.1144 (0.1196)    Prec 96.094% (95.818%)
Epoch: [42][300/391]    Time 0.103 (0.102)      Data 0.003 (0.003)      Loss
0.0745 (0.1201)    Prec 96.875% (95.785%)
Validation starts
Test: [0/79]    Time 0.229 (0.229)      Loss 0.3838 (0.3838)    Prec 90.625%
(90.625%)
 * Prec 87.160%
best acc: 89.270000
Epoch: [43][0/391]      Time 0.297 (0.297)      Data 0.227 (0.227)      Loss
0.1132 (0.1132)    Prec 95.312% (95.312%)
Epoch: [43][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.1286 (0.1139)    Prec 95.312% (95.862%)
Epoch: [43][200/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.1276 (0.1215)    Prec 96.875% (95.670%)
Epoch: [43][300/391]    Time 0.104 (0.102)      Data 0.003 (0.003)      Loss
0.1637 (0.1230)    Prec 92.188% (95.671%)
Validation starts
Test: [0/79]    Time 0.242 (0.242)      Loss 0.1469 (0.1469)    Prec 94.531%
(94.531%)
 * Prec 88.950%
best acc: 89.270000
Epoch: [44][0/391]      Time 0.290 (0.290)      Data 0.224 (0.224)      Loss
0.0643 (0.0643)    Prec 97.656% (97.656%)
Epoch: [44][100/391]    Time 0.101 (0.102)      Data 0.001 (0.004)      Loss
0.1418 (0.1106)    Prec 96.094% (96.233%)
Epoch: [44][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.2030 (0.1162)    Prec 94.531% (96.012%)
Epoch: [44][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.0861 (0.1186)    Prec 96.875% (95.878%)
Validation starts
Test: [0/79]    Time 0.192 (0.192)      Loss 0.3452 (0.3452)    Prec 90.625%
(90.625%)
 * Prec 87.690%
best acc: 89.270000
Epoch: [45][0/391]      Time 0.276 (0.276)      Data 0.215 (0.215)      Loss
```

```
0.1256 (0.1256)    Prec 94.531% (94.531%)
Epoch: [45][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.1183 (0.1011)    Prec 96.875% (96.411%)
Epoch: [45][200/391]    Time 0.098 (0.102)      Data 0.001 (0.003)      Loss
0.0866 (0.1056)    Prec 96.094% (96.308%)
Epoch: [45][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.1279 (0.1086)    Prec 96.094% (96.231%)
Validation starts
Test: [0/79]    Time 0.211 (0.211)      Loss 0.2195 (0.2195)    Prec 92.188%
(92.188%)
 * Prec 88.880%
best acc: 89.270000
Epoch: [46][0/391]      Time 0.320 (0.320)      Data 0.254 (0.254)      Loss
0.1159 (0.1159)    Prec 93.750% (93.750%)
Epoch: [46][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.0614 (0.0990)    Prec 98.438% (96.581%)
Epoch: [46][200/391]    Time 0.102 (0.102)      Data 0.001 (0.003)      Loss
0.1078 (0.1083)    Prec 96.094% (96.327%)
Epoch: [46][300/391]    Time 0.101 (0.101)      Data 0.002 (0.002)      Loss
0.0809 (0.1061)    Prec 97.656% (96.364%)
Validation starts
Test: [0/79]    Time 0.269 (0.269)      Loss 0.2622 (0.2622)    Prec 90.625%
(90.625%)
 * Prec 88.970%
best acc: 89.270000
Epoch: [47][0/391]      Time 0.309 (0.309)      Data 0.238 (0.238)      Loss
0.1236 (0.1236)    Prec 96.094% (96.094%)
Epoch: [47][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.0678 (0.1088)    Prec 97.656% (96.140%)
Epoch: [47][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.1313 (0.1017)    Prec 96.094% (96.482%)
Epoch: [47][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.2353 (0.1070)    Prec 92.188% (96.343%)
Validation starts
Test: [0/79]    Time 0.238 (0.238)      Loss 0.3139 (0.3139)    Prec 90.625%
(90.625%)
 * Prec 88.030%
best acc: 89.270000
Epoch: [48][0/391]      Time 0.304 (0.304)      Data 0.239 (0.239)      Loss
0.0798 (0.0798)    Prec 96.094% (96.094%)
Epoch: [48][100/391]    Time 0.102 (0.103)      Data 0.002 (0.004)      Loss
0.1326 (0.0948)    Prec 94.531% (96.759%)
Epoch: [48][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.1292 (0.1005)    Prec 93.750% (96.525%)
Epoch: [48][300/391]    Time 0.097 (0.102)      Data 0.002 (0.003)      Loss
0.0756 (0.1053)    Prec 96.875% (96.322%)
Validation starts
Test: [0/79]    Time 0.219 (0.219)      Loss 0.3032 (0.3032)    Prec 90.625%
```

```
(90.625%)
 * Prec 89.100%
best acc: 89.270000
Epoch: [49][0/391]      Time 0.293 (0.293)      Data 0.228 (0.228)      Loss
0.1027 (0.1027)    Prec 95.312% (95.312%)
Epoch: [49][100/391]    Time 0.104 (0.103)      Data 0.003 (0.004)      Loss
0.0550 (0.1035)    Prec 98.438% (96.279%)
Epoch: [49][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1063 (0.1027)    Prec 95.312% (96.327%)
Epoch: [49][300/391]    Time 0.101 (0.102)      Data 0.003 (0.003)      Loss
0.0772 (0.1040)    Prec 96.094% (96.304%)
Validation starts
Test: [0/79]    Time 0.222 (0.222)      Loss 0.2636 (0.2636)    Prec 92.188%
(92.188%)
 * Prec 88.550%
best acc: 89.270000
Epoch: [50][0/391]      Time 0.281 (0.281)      Data 0.218 (0.218)      Loss
0.0582 (0.0582)    Prec 98.438% (98.438%)
Epoch: [50][100/391]    Time 0.102 (0.103)      Data 0.002 (0.004)      Loss
0.0613 (0.0867)    Prec 96.875% (97.084%)
Epoch: [50][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0682 (0.0930)    Prec 96.875% (96.879%)
Epoch: [50][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.1485 (0.0952)    Prec 95.312% (96.774%)
Validation starts
Test: [0/79]    Time 0.232 (0.232)      Loss 0.2988 (0.2988)    Prec 90.625%
(90.625%)
 * Prec 89.060%
best acc: 89.270000
Epoch: [51][0/391]      Time 0.246 (0.246)      Data 0.181 (0.181)      Loss
0.0302 (0.0302)    Prec 99.219% (99.219%)
Epoch: [51][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.1126 (0.0898)    Prec 96.875% (96.960%)
Epoch: [51][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.1177 (0.0967)    Prec 97.656% (96.650%)
Epoch: [51][300/391]    Time 0.102 (0.102)      Data 0.003 (0.003)      Loss
0.1022 (0.0989)    Prec 96.875% (96.571%)
Validation starts
Test: [0/79]    Time 0.215 (0.215)      Loss 0.2328 (0.2328)    Prec 93.750%
(93.750%)
 * Prec 89.750%
best acc: 89.750000
Epoch: [52][0/391]      Time 0.337 (0.337)      Data 0.269 (0.269)      Loss
0.1539 (0.1539)    Prec 94.531% (94.531%)
Epoch: [52][100/391]    Time 0.106 (0.103)      Data 0.002 (0.004)      Loss
0.0597 (0.0918)    Prec 98.438% (96.829%)
Epoch: [52][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0738 (0.0967)    Prec 96.875% (96.630%)
```

```
Epoch: [52][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.0530 (0.1001)    Prec 99.219% (96.519%)
Validation starts
Test: [0/79]    Time 0.214 (0.214)      Loss 0.2998 (0.2998)    Prec 91.406%
(91.406%)
 * Prec 89.910%
best acc: 89.910000
Epoch: [53][0/391]      Time 0.311 (0.311)      Data 0.249 (0.249)      Loss
0.1103 (0.1103)    Prec 96.094% (96.094%)
Epoch: [53][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.0878 (0.0852)    Prec 97.656% (96.999%)
Epoch: [53][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.0973 (0.0900)    Prec 94.531% (96.836%)
Epoch: [53][300/391]    Time 0.099 (0.102)      Data 0.002 (0.003)      Loss
0.0654 (0.0923)    Prec 97.656% (96.737%)
Validation starts
Test: [0/79]    Time 0.234 (0.234)      Loss 0.3082 (0.3082)    Prec 89.062%
(89.062%)
 * Prec 89.150%
best acc: 89.910000
Epoch: [54][0/391]      Time 0.321 (0.321)      Data 0.256 (0.256)      Loss
0.1304 (0.1304)    Prec 95.312% (95.312%)
Epoch: [54][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.0654 (0.0896)    Prec 97.656% (96.883%)
Epoch: [54][200/391]    Time 0.098 (0.102)      Data 0.001 (0.003)      Loss
0.1251 (0.0887)    Prec 95.312% (96.894%)
Epoch: [54][300/391]    Time 0.100 (0.101)      Data 0.002 (0.003)      Loss
0.0982 (0.0926)    Prec 95.312% (96.782%)
Validation starts
Test: [0/79]    Time 0.217 (0.217)      Loss 0.2908 (0.2908)    Prec 89.062%
(89.062%)
 * Prec 89.090%
best acc: 89.910000
Epoch: [55][0/391]      Time 0.340 (0.340)      Data 0.271 (0.271)      Loss
0.0598 (0.0598)    Prec 97.656% (97.656%)
Epoch: [55][100/391]    Time 0.100 (0.103)      Data 0.001 (0.004)      Loss
0.0203 (0.0849)    Prec 100.000% (97.130%)
Epoch: [55][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0735 (0.0862)    Prec 97.656% (97.050%)
Epoch: [55][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.1050 (0.0925)    Prec 96.094% (96.857%)
Validation starts
Test: [0/79]    Time 0.209 (0.209)      Loss 0.2978 (0.2978)    Prec 91.406%
(91.406%)
 * Prec 88.450%
best acc: 89.910000
Epoch: [56][0/391]      Time 0.342 (0.342)      Data 0.277 (0.277)      Loss
0.0543 (0.0543)    Prec 97.656% (97.656%)
```

```
Epoch: [56][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.1661 (0.0890)    Prec 95.312% (96.867%)
Epoch: [56][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0557 (0.0861)    Prec 97.656% (96.922%)
Epoch: [56][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.0767 (0.0901)    Prec 96.875% (96.810%)
Validation starts
Test: [0/79]    Time 0.226 (0.226)      Loss 0.2917 (0.2917)    Prec 92.969%
(92.969%)
 * Prec 88.310%
best acc: 89.910000
Epoch: [57][0/391]      Time 0.276 (0.276)      Data 0.210 (0.210)      Loss
0.0658 (0.0658)    Prec 97.656% (97.656%)
Epoch: [57][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.0589 (0.0828)    Prec 96.875% (97.045%)
Epoch: [57][200/391]    Time 0.099 (0.102)      Data 0.002 (0.003)      Loss
0.1116 (0.0877)    Prec 95.312% (96.926%)
Epoch: [57][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.1143 (0.0883)    Prec 96.094% (96.961%)
Validation starts
Test: [0/79]    Time 0.210 (0.210)      Loss 0.2156 (0.2156)    Prec 92.969%
(92.969%)
 * Prec 89.310%
best acc: 89.910000
Epoch: [58][0/391]      Time 0.280 (0.280)      Data 0.214 (0.214)      Loss
0.0457 (0.0457)    Prec 98.438% (98.438%)
Epoch: [58][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.0855 (0.0866)    Prec 97.656% (97.045%)
Epoch: [58][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1416 (0.0913)    Prec 95.312% (96.875%)
Epoch: [58][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.1324 (0.0909)    Prec 96.875% (96.880%)
Validation starts
Test: [0/79]    Time 0.265 (0.265)      Loss 0.2226 (0.2226)    Prec 91.406%
(91.406%)
 * Prec 88.830%
best acc: 89.910000
Epoch: [59][0/391]      Time 0.340 (0.340)      Data 0.274 (0.274)      Loss
0.0819 (0.0819)    Prec 96.094% (96.094%)
Epoch: [59][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.0861 (0.0739)    Prec 97.656% (97.409%)
Epoch: [59][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.1467 (0.0797)    Prec 94.531% (97.322%)
Epoch: [59][300/391]    Time 0.100 (0.101)      Data 0.002 (0.002)      Loss
0.0439 (0.0850)    Prec 98.438% (97.106%)
Validation starts
Test: [0/79]    Time 0.226 (0.226)      Loss 0.3274 (0.3274)    Prec 89.062%
(89.062%)
```

```
 * Prec 88.910%
best acc: 89.910000
Epoch: [60][0/391]      Time 0.305 (0.305)      Data 0.239 (0.239)      Loss
0.0462 (0.0462)    Prec 97.656% (97.656%)
Epoch: [60][100/391]    Time 0.099 (0.103)      Data 0.002 (0.005)      Loss
0.0529 (0.0793)    Prec 97.656% (97.177%)
Epoch: [60][200/391]    Time 0.105 (0.102)      Data 0.002 (0.003)      Loss
0.1339 (0.0808)    Prec 96.875% (97.174%)
Epoch: [60][300/391]    Time 0.102 (0.102)      Data 0.003 (0.003)      Loss
0.1017 (0.0817)    Prec 96.094% (97.181%)
Validation starts
Test: [0/79]    Time 0.230 (0.230)      Loss 0.3120 (0.3120)    Prec 92.188%
(92.188%)
 * Prec 88.490%
best acc: 89.910000
Epoch: [61][0/391]      Time 0.316 (0.316)      Data 0.248 (0.248)      Loss
0.1476 (0.1476)    Prec 95.312% (95.312%)
Epoch: [61][100/391]    Time 0.102 (0.103)      Data 0.002 (0.004)      Loss
0.0629 (0.0878)    Prec 97.656% (96.852%)
Epoch: [61][200/391]    Time 0.105 (0.102)      Data 0.002 (0.003)      Loss
0.0662 (0.0852)    Prec 97.656% (96.988%)
Epoch: [61][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.1486 (0.0871)    Prec 95.312% (96.917%)
Validation starts
Test: [0/79]    Time 0.251 (0.251)      Loss 0.3126 (0.3126)    Prec 90.625%
(90.625%)
 * Prec 88.510%
best acc: 89.910000
Epoch: [62][0/391]      Time 0.295 (0.295)      Data 0.227 (0.227)      Loss
0.0413 (0.0413)    Prec 98.438% (98.438%)
Epoch: [62][100/391]    Time 0.102 (0.103)      Data 0.002 (0.004)      Loss
0.0796 (0.0749)    Prec 96.875% (97.223%)
Epoch: [62][200/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.0579 (0.0791)    Prec 98.438% (97.159%)
Epoch: [62][300/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.0540 (0.0848)    Prec 97.656% (96.937%)
Validation starts
Test: [0/79]    Time 0.217 (0.217)      Loss 0.2428 (0.2428)    Prec 92.188%
(92.188%)
 * Prec 89.170%
best acc: 89.910000
Epoch: [63][0/391]      Time 0.284 (0.284)      Data 0.216 (0.216)      Loss
0.0477 (0.0477)    Prec 98.438% (98.438%)
Epoch: [63][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.1004 (0.0750)    Prec 95.312% (97.231%)
Epoch: [63][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1261 (0.0801)    Prec 95.312% (97.217%)
Epoch: [63][300/391]    Time 0.103 (0.101)      Data 0.001 (0.002)      Loss
```

```
0.0731 (0.0841)    Prec 96.875% (97.075%)
Validation starts
Test: [0/79]    Time 0.266 (0.266)      Loss 0.2783 (0.2783)    Prec 91.406%
(91.406%)
 * Prec 89.430%
best acc: 89.910000
Epoch: [64][0/391]      Time 0.322 (0.322)      Data 0.255 (0.255)      Loss
0.0465 (0.0465)    Prec 98.438% (98.438%)
Epoch: [64][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.1134 (0.0759)    Prec 97.656% (97.107%)
Epoch: [64][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0505 (0.0783)    Prec 98.438% (97.244%)
Epoch: [64][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.0679 (0.0801)    Prec 96.094% (97.199%)
Validation starts
Test: [0/79]    Time 0.264 (0.264)      Loss 0.2387 (0.2387)    Prec 93.750%
(93.750%)
 * Prec 89.820%
best acc: 89.910000
Epoch: [65][0/391]      Time 0.346 (0.346)      Data 0.280 (0.280)      Loss
0.0743 (0.0743)    Prec 97.656% (97.656%)
Epoch: [65][100/391]    Time 0.107 (0.103)      Data 0.003 (0.005)      Loss
0.0627 (0.0703)    Prec 97.656% (97.502%)
Epoch: [65][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.1522 (0.0785)    Prec 96.875% (97.275%)
Epoch: [65][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0643 (0.0795)    Prec 97.656% (97.267%)
Validation starts
Test: [0/79]    Time 0.198 (0.198)      Loss 0.2318 (0.2318)    Prec 92.188%
(92.188%)
 * Prec 89.840%
best acc: 89.910000
Epoch: [66][0/391]      Time 0.393 (0.393)      Data 0.318 (0.318)      Loss
0.0681 (0.0681)    Prec 97.656% (97.656%)
Epoch: [66][100/391]    Time 0.099 (0.104)      Data 0.003 (0.005)      Loss
0.0423 (0.0774)    Prec 98.438% (97.416%)
Epoch: [66][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0669 (0.0769)    Prec 96.875% (97.373%)
Epoch: [66][300/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0864 (0.0777)    Prec 96.094% (97.306%)
Validation starts
Test: [0/79]    Time 0.252 (0.252)      Loss 0.1076 (0.1076)    Prec 96.094%
(96.094%)
 * Prec 89.950%
best acc: 89.950000
Epoch: [67][0/391]      Time 0.335 (0.335)      Data 0.268 (0.268)      Loss
0.1218 (0.1218)    Prec 96.875% (96.875%)
Epoch: [67][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
```

```
0.0733 (0.0721)    Prec 98.438% (97.509%)
Epoch: [67][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0874 (0.0747)    Prec 96.094% (97.396%)
Epoch: [67][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.0445 (0.0816)    Prec 98.438% (97.168%)
Validation starts
Test: [0/79]    Time 0.209 (0.209)      Loss 0.2411 (0.2411)    Prec 93.750%
(93.750%)
 * Prec 89.580%
best acc: 89.950000
Epoch: [68][0/391]      Time 0.283 (0.283)      Data 0.217 (0.217)      Loss
0.0582 (0.0582)    Prec 97.656% (97.656%)
Epoch: [68][100/391]    Time 0.101 (0.103)      Data 0.001 (0.004)      Loss
0.0483 (0.0745)    Prec 98.438% (97.509%)
Epoch: [68][200/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0421 (0.0771)    Prec 98.438% (97.357%)
Epoch: [68][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.0791 (0.0793)    Prec 94.531% (97.282%)
Validation starts
Test: [0/79]    Time 0.201 (0.201)      Loss 0.3057 (0.3057)    Prec 90.625%
(90.625%)
 * Prec 89.080%
best acc: 89.950000
Epoch: [69][0/391]      Time 0.377 (0.377)      Data 0.311 (0.311)      Loss
0.0271 (0.0271)    Prec 99.219% (99.219%)
Epoch: [69][100/391]    Time 0.101 (0.103)      Data 0.001 (0.005)      Loss
0.0920 (0.0720)    Prec 95.312% (97.571%)
Epoch: [69][200/391]    Time 0.099 (0.102)      Data 0.001 (0.003)      Loss
0.1073 (0.0778)    Prec 92.969% (97.423%)
Epoch: [69][300/391]    Time 0.100 (0.102)      Data 0.001 (0.003)      Loss
0.0406 (0.0796)    Prec 99.219% (97.345%)
Validation starts
Test: [0/79]    Time 0.210 (0.210)      Loss 0.4089 (0.4089)    Prec 87.500%
(87.500%)
 * Prec 89.580%
best acc: 89.950000
Epoch: [70][0/391]      Time 0.271 (0.271)      Data 0.204 (0.204)      Loss
0.1106 (0.1106)    Prec 95.312% (95.312%)
Epoch: [70][100/391]    Time 0.101 (0.102)      Data 0.001 (0.004)      Loss
0.0504 (0.0702)    Prec 97.656% (97.579%)
Epoch: [70][200/391]    Time 0.099 (0.101)      Data 0.002 (0.003)      Loss
0.1232 (0.0735)    Prec 95.312% (97.454%)
Epoch: [70][300/391]    Time 0.101 (0.101)      Data 0.001 (0.002)      Loss
0.1170 (0.0762)    Prec 96.094% (97.392%)
Validation starts
Test: [0/79]    Time 0.245 (0.245)      Loss 0.2221 (0.2221)    Prec 93.750%
(93.750%)
 * Prec 89.790%
```

```
best acc: 89.950000
Epoch: [71][0/391]      Time 0.349 (0.349)      Data 0.282 (0.282)      Loss
0.0657 (0.0657)    Prec 97.656% (97.656%)
Epoch: [71][100/391]    Time 0.100 (0.103)      Data 0.001 (0.005)      Loss
0.0421 (0.0693)    Prec 98.438% (97.517%)
Epoch: [71][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0876 (0.0720)    Prec 97.656% (97.462%)
Epoch: [71][300/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.1088 (0.0769)    Prec 95.312% (97.301%)
Validation starts
Test: [0/79]    Time 0.214 (0.214)      Loss 0.2847 (0.2847)    Prec 89.844%
(89.844%)
 * Prec 88.450%
best acc: 89.950000
Epoch: [72][0/391]      Time 0.366 (0.366)      Data 0.298 (0.298)      Loss
0.1400 (0.1400)    Prec 95.312% (95.312%)
Epoch: [72][100/391]    Time 0.101 (0.104)      Data 0.002 (0.005)      Loss
0.2479 (0.0798)    Prec 92.969% (97.269%)
Epoch: [72][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0849 (0.0769)    Prec 97.656% (97.338%)
Epoch: [72][300/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.0911 (0.0786)    Prec 96.875% (97.301%)
Validation starts
Test: [0/79]    Time 0.296 (0.296)      Loss 0.3173 (0.3173)    Prec 89.062%
(89.062%)
 * Prec 89.640%
best acc: 89.950000
Epoch: [73][0/391]      Time 0.303 (0.303)      Data 0.234 (0.234)      Loss
0.0196 (0.0196)    Prec 99.219% (99.219%)
Epoch: [73][100/391]    Time 0.102 (0.103)      Data 0.002 (0.004)      Loss
0.1067 (0.0759)    Prec 96.094% (97.386%)
Epoch: [73][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0805 (0.0756)    Prec 96.875% (97.427%)
Epoch: [73][300/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.1124 (0.0757)    Prec 96.875% (97.384%)
Validation starts
Test: [0/79]    Time 0.228 (0.228)      Loss 0.2506 (0.2506)    Prec 89.062%
(89.062%)
 * Prec 89.470%
best acc: 89.950000
Epoch: [74][0/391]      Time 0.244 (0.244)      Data 0.185 (0.185)      Loss
0.0264 (0.0264)    Prec 100.000% (100.000%)
Epoch: [74][100/391]    Time 0.101 (0.102)      Data 0.002 (0.004)      Loss
0.0919 (0.0711)    Prec 96.094% (97.432%)
Epoch: [74][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0590 (0.0706)    Prec 96.875% (97.509%)
Epoch: [74][300/391]    Time 0.100 (0.101)      Data 0.001 (0.002)      Loss
0.1116 (0.0699)    Prec 95.312% (97.545%)
```

```
Validation starts
Test: [0/79]    Time 0.240 (0.240)      Loss 0.4449 (0.4449)    Prec 90.625%
(90.625%)
 * Prec 89.740%
best acc: 89.950000
Epoch: [75][0/391]      Time 0.265 (0.265)      Data 0.198 (0.198)      Loss
0.1102 (0.1102)    Prec 95.312% (95.312%)
Epoch: [75][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.0610 (0.0672)    Prec 97.656% (97.757%)
Epoch: [75][200/391]    Time 0.101 (0.102)      Data 0.001 (0.003)      Loss
0.0434 (0.0726)    Prec 97.656% (97.532%)
Epoch: [75][300/391]    Time 0.101 (0.101)      Data 0.002 (0.002)      Loss
0.0708 (0.0737)    Prec 97.656% (97.482%)
Validation starts
Test: [0/79]    Time 0.201 (0.201)      Loss 0.2467 (0.2467)    Prec 92.188%
(92.188%)
 * Prec 88.880%
best acc: 89.950000
Epoch: [76][0/391]      Time 0.266 (0.266)      Data 0.196 (0.196)      Loss
0.0453 (0.0453)    Prec 97.656% (97.656%)
Epoch: [76][100/391]    Time 0.100 (0.103)      Data 0.002 (0.004)      Loss
0.0277 (0.0728)    Prec 99.219% (97.502%)
Epoch: [76][200/391]    Time 0.104 (0.102)      Data 0.002 (0.003)      Loss
0.0413 (0.0709)    Prec 99.219% (97.594%)
Epoch: [76][300/391]    Time 0.102 (0.102)      Data 0.002 (0.003)      Loss
0.0335 (0.0740)    Prec 99.219% (97.521%)
Validation starts
Test: [0/79]    Time 0.285 (0.285)      Loss 0.1829 (0.1829)    Prec 93.750%
(93.750%)
 * Prec 89.910%
best acc: 89.950000
Epoch: [77][0/391]      Time 0.282 (0.282)      Data 0.215 (0.215)      Loss
0.0215 (0.0215)    Prec 100.000% (100.000%)
Epoch: [77][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.0285 (0.0709)    Prec 99.219% (97.710%)
Epoch: [77][200/391]    Time 0.098 (0.102)      Data 0.002 (0.003)      Loss
0.0408 (0.0770)    Prec 99.219% (97.431%)
Epoch: [77][300/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.0575 (0.0736)    Prec 97.656% (97.519%)
Validation starts
Test: [0/79]    Time 0.214 (0.214)      Loss 0.2028 (0.2028)    Prec 92.188%
(92.188%)
 * Prec 88.970%
best acc: 89.950000
Epoch: [78][0/391]      Time 0.293 (0.293)      Data 0.232 (0.232)      Loss
0.0850 (0.0850)    Prec 96.875% (96.875%)
Epoch: [78][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.0893 (0.0619)    Prec 97.656% (98.004%)
```

```
Epoch: [78][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0642 (0.0637)    Prec 97.656% (97.897%)
Epoch: [78][300/391]    Time 0.098 (0.102)      Data 0.002 (0.003)      Loss
0.1000 (0.0706)    Prec 96.875% (97.656%)
Validation starts
Test: [0/79]    Time 0.217 (0.217)      Loss 0.2482 (0.2482)    Prec 92.188%
(92.188%)
 * Prec 90.500%
best acc: 90.500000
Epoch: [79][0/391]      Time 0.329 (0.329)      Data 0.265 (0.265)      Loss
0.1073 (0.1073)    Prec 97.656% (97.656%)
Epoch: [79][100/391]    Time 0.100 (0.104)      Data 0.004 (0.005)      Loss
0.0413 (0.0664)    Prec 99.219% (97.772%)
Epoch: [79][200/391]    Time 0.102 (0.102)      Data 0.003 (0.004)      Loss
0.0310 (0.0677)    Prec 99.219% (97.656%)
Epoch: [79][300/391]    Time 0.105 (0.102)      Data 0.004 (0.003)      Loss
0.0931 (0.0702)    Prec 96.875% (97.581%)
Validation starts
Test: [0/79]    Time 0.192 (0.192)      Loss 0.3473 (0.3473)    Prec 91.406%
(91.406%)
 * Prec 89.720%
best acc: 90.500000
Epoch: [80][0/391]      Time 0.327 (0.327)      Data 0.261 (0.261)      Loss
0.0413 (0.0413)    Prec 98.438% (98.438%)
Epoch: [80][100/391]    Time 0.101 (0.103)      Data 0.002 (0.005)      Loss
0.0728 (0.0673)    Prec 96.094% (97.780%)
Epoch: [80][200/391]    Time 0.100 (0.102)      Data 0.002 (0.003)      Loss
0.0569 (0.0719)    Prec 97.656% (97.579%)
Epoch: [80][300/391]    Time 0.099 (0.102)      Data 0.002 (0.003)      Loss
0.0377 (0.0719)    Prec 98.438% (97.539%)
Validation starts
Test: [0/79]    Time 0.211 (0.211)      Loss 0.2147 (0.2147)    Prec 92.188%
(92.188%)
 * Prec 89.770%
best acc: 90.500000
Epoch: [81][0/391]      Time 0.285 (0.285)      Data 0.219 (0.219)      Loss
0.1370 (0.1370)    Prec 95.312% (95.312%)
Epoch: [81][100/391]    Time 0.101 (0.103)      Data 0.002 (0.004)      Loss
0.0397 (0.0706)    Prec 98.438% (97.679%)
Epoch: [81][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0322 (0.0705)    Prec 99.219% (97.641%)
Epoch: [81][300/391]    Time 0.102 (0.101)      Data 0.002 (0.002)      Loss
0.0355 (0.0730)    Prec 98.438% (97.506%)
Validation starts
Test: [0/79]    Time 0.215 (0.215)      Loss 0.2849 (0.2849)    Prec 90.625%
(90.625%)
 * Prec 89.480%
best acc: 90.500000
```

```
Epoch: [82][0/391]      Time 0.332 (0.332)      Data 0.261 (0.261)      Loss
0.0514 (0.0514)    Prec 97.656% (97.656%)
Epoch: [82][100/391]    Time 0.101 (0.103)      Data 0.003 (0.005)      Loss
0.0297 (0.0696)    Prec 99.219% (97.602%)
Epoch: [82][200/391]    Time 0.104 (0.102)      Data 0.002 (0.003)      Loss
0.0553 (0.0710)    Prec 96.875% (97.532%)
Epoch: [82][300/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0369 (0.0715)    Prec 97.656% (97.524%)
Validation starts
Test: [0/79]    Time 0.218 (0.218)      Loss 0.4392 (0.4392)    Prec 88.281%
(88.281%)
 * Prec 89.690%
best acc: 90.500000
Epoch: [83][0/391]      Time 0.315 (0.315)      Data 0.249 (0.249)      Loss
0.1128 (0.1128)    Prec 95.312% (95.312%)
Epoch: [83][100/391]    Time 0.103 (0.103)      Data 0.003 (0.005)      Loss
0.0240 (0.0684)    Prec 99.219% (97.618%)
Epoch: [83][200/391]    Time 0.101 (0.102)      Data 0.002 (0.003)      Loss
0.0829 (0.0703)    Prec 96.875% (97.582%)
```

```
      ␣
↪---------------------------------------------------------------------------

       KeyboardInterrupt                         Traceback (most recent call␣
↪last)

       /tmp/ipykernel_4662/382461631.py in <module>
        20     adjust_learning_rate(optimizer, epoch)
        21
   ---> 22     train(trainloader, model, criterion, optimizer, epoch)
        23
        24     # evaluate on test set


       /tmp/ipykernel_4662/1701083248.py in train(trainloader, model,␣
↪criterion, optimizer, epoch)
        88         # compute gradient and do SGD step
        89         optimizer.zero_grad()
   ---> 90         loss.backward()
        91         optimizer.step()
        92


       /opt/conda/lib/python3.9/site-packages/torch/_tensor.py in␣
↪backward(self, gradient, retain_graph, create_graph, inputs)
       253                 create_graph=create_graph,
       254                 inputs=inputs)
```

```
   --> 255         torch.autograd.backward(self, gradient, retain_graph,␣
↪create_graph, inputs=inputs)
       256
       257     def register_hook(self, hook):


       /opt/conda/lib/python3.9/site-packages/torch/autograd/__init__.py in␣
↪backward(tensors, grad_tensors, retain_graph, create_graph, grad_variables,␣
↪inputs)
       145         retain_graph = create_graph
       146
   --> 147     Variable._execution_engine.run_backward(
       148         tensors, grad_tensors_, retain_graph, create_graph, inputs,
       149         allow_unreachable=True, accumulate_grad=True)  #␣
↪allow_unreachable flag


       KeyboardInterrupt:
```

[4]:
```
PATH = "result/VGG16_quant/model_best.pth.tar"
checkpoint = torch.load(PATH)
model.load_state_dict(checkpoint['state_dict'])
device = torch.device("cuda")

model.cuda()
model.eval()

test_loss = 0
correct = 0

with torch.no_grad():
    for data, target in testloader:
        data, target = data.to(device), target.to(device) # loading to GPU
        output = model(data)
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

test_loss /= len(testloader.dataset)

print('\nTest set: Accuracy: {}/{} ({:.0f}%)\n'.format(
        correct, len(testloader.dataset),
        100. * correct / len(testloader.dataset)))
```

```
Test set: Accuracy: 9050/10000 (90%)
```

[5]:
```
#### Prune all the QuantConv2D layers' 90% weights with 1) unstructured, and 2) structured manner.
import torch.nn.utils.prune as prune

#unstructured
'''
for layer in model.modules():
    if isinstance(layer, QuantConv2d):
        prune.l1_unstructured(layer, name='weight', amount=0.9)
'''
#structured

for layer in model.modules():
    if isinstance(layer, QuantConv2d):
        prune.ln_structured(layer, name='weight', amount=0.9, n=1, dim=0)
```

[ ]:
```
print(list(model.features[40].named_parameters())) # check whether there is mask, weight_org, ...
print(model.features[40].weight) # check whether there are many zeros
```

[6]:
```
### Check sparsity ###
for layer in model.modules():
    if isinstance(layer, QuantConv2d):
        mask1 = layer.weight_mask
        sparsity_mask1 = (mask1 == 0).sum() / mask1.nelement()
        print("Sparsity level: ", sparsity_mask1)
```

```
Sparsity level:  tensor(0.9062, device='cuda:0')
Sparsity level:  tensor(0.9062, device='cuda:0')
Sparsity level:  tensor(0.8984, device='cuda:0')
Sparsity level:  tensor(0.8984, device='cuda:0')
Sparsity level:  tensor(0.8984, device='cuda:0')
Sparsity level:  tensor(0.8984, device='cuda:0')
Sparsity level:  tensor(0.8984, device='cuda:0')
Sparsity level:  tensor(0.9004, device='cuda:0')
Sparsity level:  tensor(0.9004, device='cuda:0')
Sparsity level:  tensor(0.9004, device='cuda:0')
Sparsity level:  tensor(0.9004, device='cuda:0')
Sparsity level:  tensor(0.9004, device='cuda:0')
Sparsity level:  tensor(0.9004, device='cuda:0')
```

[7]:
```
## check accuracy after pruning

model.cuda()
model.eval()

test_loss = 0
```

```
correct = 0

with torch.no_grad():
    for data, target in testloader:
        data, target = data.to(device), target.to(device) # loading to GPU
        output = model(data)
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

test_loss /= len(testloader.dataset)

print('\nTest set: Accuracy: {}/{} ({:.0f}%)\n'.format(
        correct, len(testloader.dataset),
        100. * correct / len(testloader.dataset)))
```

```
Test set: Accuracy: 1000/10000 (10%)
```

[8]:
```
## Start finetuning (training here), and see how much you can recover your␣
 ↪accuracy ##
## You can change hyper parameters such as epochs or lr ##

lr = 6e-2
weight_decay = 1e-4
epochs = 500
best_prec = 0

#model = nn.DataParallel(model).cuda()
model.cuda()
criterion = nn.CrossEntropyLoss().cuda()
optimizer = torch.optim.SGD(model.parameters(), lr=lr, momentum=0.9,␣
 ↪weight_decay=weight_decay)
#cudnn.benchmark = True

if not os.path.exists('result'):
    os.makedirs('result')
fdir = 'result/'+str(model_name)
if not os.path.exists(fdir):
    os.makedirs(fdir)


for epoch in range(0, epochs):
    adjust_learning_rate(optimizer, epoch)

    train(trainloader, model, criterion, optimizer, epoch)
```

```
    # evaluate on test set
    print("Validation starts")
    prec = validate(testloader, model, criterion)

    # remember best precision and save checkpoint
    is_best = prec > best_prec
    best_prec = max(prec,best_prec)
    print('best acc: {:1f}'.format(best_prec))
    save_checkpoint({
        'epoch': epoch + 1,
        'state_dict': model.state_dict(),
        'best_prec': best_prec,
        'optimizer': optimizer.state_dict(),
    }, is_best, fdir)
```

```
Epoch: [0][0/391]       Time 0.335 (0.335)      Data 0.262 (0.262)      Loss
2.5078 (2.5078)    Prec 8.594% (8.594%)
Epoch: [0][100/391]     Time 0.107 (0.105)      Data 0.002 (0.004)      Loss
1.8831 (2.0045)    Prec 23.438% (22.068%)
Epoch: [0][200/391]     Time 0.096 (0.103)      Data 0.002 (0.003)      Loss
1.6646 (1.8994)    Prec 39.844% (25.944%)
Epoch: [0][300/391]     Time 0.102 (0.103)      Data 0.001 (0.003)      Loss
1.7335 (1.8295)    Prec 30.469% (28.478%)
Validation starts
Test: [0/79]    Time 0.228 (0.228)      Loss 1.7554 (1.7554)    Prec 30.469%
(30.469%)
 * Prec 32.220%
best acc: 32.220000
Epoch: [1][0/391]       Time 0.270 (0.270)      Data 0.209 (0.209)      Loss
1.6027 (1.6027)    Prec 29.688% (29.688%)
Epoch: [1][100/391]     Time 0.104 (0.104)      Data 0.001 (0.004)      Loss
1.6172 (1.5695)    Prec 37.500% (40.130%)
Epoch: [1][200/391]     Time 0.103 (0.103)      Data 0.001 (0.003)      Loss
1.3075 (1.5391)    Prec 37.500% (41.702%)
Epoch: [1][300/391]     Time 0.105 (0.103)      Data 0.002 (0.002)      Loss
1.6422 (1.5135)    Prec 39.844% (42.808%)
Validation starts
Test: [0/79]    Time 0.217 (0.217)      Loss 1.2983 (1.2983)    Prec 53.906%
(53.906%)
 * Prec 49.380%
best acc: 49.380000
Epoch: [2][0/391]       Time 0.322 (0.322)      Data 0.255 (0.255)      Loss
1.3269 (1.3269)    Prec 53.125% (53.125%)
Epoch: [2][100/391]     Time 0.103 (0.105)      Data 0.002 (0.004)      Loss
1.2577 (1.3537)    Prec 57.031% (50.774%)
Epoch: [2][200/391]     Time 0.102 (0.104)      Data 0.004 (0.003)      Loss
1.3393 (1.3455)    Prec 54.688% (51.263%)
```

```
Epoch: [2][300/391]     Time 0.102 (0.103)      Data 0.001 (0.003)      Loss
1.3606 (1.3300)    Prec 50.000% (51.918%)
Validation starts
Test: [0/79]    Time 0.232 (0.232)      Loss 1.4953 (1.4953)    Prec 50.000%
(50.000%)
 * Prec 47.490%
best acc: 49.380000
Epoch: [3][0/391]       Time 0.292 (0.292)      Data 0.233 (0.233)      Loss
1.4609 (1.4609)    Prec 49.219% (49.219%)
Epoch: [3][100/391]     Time 0.101 (0.105)      Data 0.003 (0.004)      Loss
1.5142 (1.2486)    Prec 44.531% (55.353%)
Epoch: [3][200/391]     Time 0.106 (0.104)      Data 0.002 (0.003)      Loss
1.1569 (1.2369)    Prec 60.156% (55.838%)
Epoch: [3][300/391]     Time 0.105 (0.103)      Data 0.002 (0.003)      Loss
1.1129 (1.2233)    Prec 58.594% (56.118%)
Validation starts
Test: [0/79]    Time 0.209 (0.209)      Loss 1.1827 (1.1827)    Prec 58.594%
(58.594%)
 * Prec 54.200%
best acc: 54.200000
Epoch: [4][0/391]       Time 0.263 (0.263)      Data 0.200 (0.200)      Loss
1.1807 (1.1807)    Prec 56.250% (56.250%)
Epoch: [4][100/391]     Time 0.102 (0.104)      Data 0.001 (0.004)      Loss
1.1509 (1.1676)    Prec 56.250% (58.238%)
Epoch: [4][200/391]     Time 0.104 (0.103)      Data 0.002 (0.003)      Loss
1.1279 (1.1536)    Prec 57.031% (58.761%)
Epoch: [4][300/391]     Time 0.101 (0.103)      Data 0.002 (0.002)      Loss
1.0220 (1.1472)    Prec 59.375% (58.952%)
Validation starts
Test: [0/79]    Time 0.248 (0.248)      Loss 1.2900 (1.2900)    Prec 50.781%
(50.781%)
 * Prec 51.890%
best acc: 54.200000
Epoch: [5][0/391]       Time 0.310 (0.310)      Data 0.243 (0.243)      Loss
1.0424 (1.0424)    Prec 63.281% (63.281%)
Epoch: [5][100/391]     Time 0.100 (0.105)      Data 0.001 (0.004)      Loss
1.2556 (1.1125)    Prec 53.906% (60.729%)
Epoch: [5][200/391]     Time 0.105 (0.104)      Data 0.003 (0.003)      Loss
0.9129 (1.1044)    Prec 67.969% (60.592%)
Epoch: [5][300/391]     Time 0.102 (0.104)      Data 0.003 (0.003)      Loss
1.0772 (1.0988)    Prec 62.500% (60.649%)
Validation starts
Test: [0/79]    Time 0.236 (0.236)      Loss 1.2895 (1.2895)    Prec 57.031%
(57.031%)
 * Prec 55.300%
best acc: 55.300000
Epoch: [6][0/391]       Time 0.266 (0.266)      Data 0.197 (0.197)      Loss
1.0722 (1.0722)    Prec 62.500% (62.500%)
```

```
Epoch: [6][100/391]     Time 0.101 (0.104)      Data 0.001 (0.004)      Loss
1.0807 (1.0631)    Prec 61.719% (61.974%)
Epoch: [6][200/391]     Time 0.102 (0.103)      Data 0.001 (0.003)      Loss
0.9654 (1.0617)    Prec 64.062% (62.049%)
Epoch: [6][300/391]     Time 0.102 (0.103)      Data 0.001 (0.002)      Loss
1.1401 (1.0593)    Prec 57.812% (62.100%)
Validation starts
Test: [0/79]    Time 0.204 (0.204)      Loss 1.0961 (1.0961)    Prec 60.156%
(60.156%)
 * Prec 57.990%
best acc: 57.990000
Epoch: [7][0/391]       Time 0.304 (0.304)      Data 0.228 (0.228)      Loss
1.1250 (1.1250)    Prec 65.625% (65.625%)
Epoch: [7][100/391]     Time 0.101 (0.105)      Data 0.003 (0.004)      Loss
1.1018 (1.0316)    Prec 60.938% (63.506%)
Epoch: [7][200/391]     Time 0.103 (0.104)      Data 0.001 (0.003)      Loss
1.1174 (1.0336)    Prec 64.844% (63.390%)
Epoch: [7][300/391]     Time 0.102 (0.103)      Data 0.002 (0.003)      Loss
0.9435 (1.0222)    Prec 64.844% (63.738%)
Validation starts
Test: [0/79]    Time 0.214 (0.214)      Loss 1.1084 (1.1084)    Prec 59.375%
(59.375%)
 * Prec 60.690%
best acc: 60.690000
Epoch: [8][0/391]       Time 0.259 (0.259)      Data 0.194 (0.194)      Loss
1.1470 (1.1470)    Prec 61.719% (61.719%)
Epoch: [8][100/391]     Time 0.102 (0.105)      Data 0.003 (0.004)      Loss
0.9745 (1.0079)    Prec 63.281% (64.155%)
Epoch: [8][200/391]     Time 0.103 (0.104)      Data 0.002 (0.003)      Loss
1.0573 (0.9968)    Prec 62.500% (64.622%)
Epoch: [8][300/391]     Time 0.103 (0.104)      Data 0.001 (0.003)      Loss
1.2194 (0.9960)    Prec 55.469% (64.784%)
Validation starts
Test: [0/79]    Time 0.248 (0.248)      Loss 1.0247 (1.0247)    Prec 64.062%
(64.062%)
 * Prec 62.470%
best acc: 62.470000
Epoch: [9][0/391]       Time 0.255 (0.255)      Data 0.194 (0.194)      Loss
0.8966 (0.8966)    Prec 67.188% (67.188%)
Epoch: [11][100/391]    Time 0.103 (0.105)      Data 0.002 (0.004)      Loss
0.7816 (0.9397)    Prec 72.656% (66.955%)
Epoch: [11][200/391]    Time 0.100 (0.104)      Data 0.001 (0.003)      Loss
1.0470 (0.9325)    Prec 62.500% (67.409%)
Epoch: [11][300/391]    Time 0.105 (0.103)      Data 0.002 (0.003)      Loss
0.9107 (0.9284)    Prec 71.094% (67.444%)
Validation starts
Test: [0/79]    Time 0.181 (0.181)      Loss 0.8731 (0.8731)    Prec 65.625%
(65.625%)
```

```
 * Prec 67.780%
best acc: 67.780000
Epoch: [12][0/391]      Time 0.286 (0.286)      Data 0.222 (0.222)      Loss
0.8426 (0.8426)    Prec 70.312% (70.312%)
Epoch: [12][100/391]    Time 0.101 (0.105)      Data 0.002 (0.005)      Loss
0.7993 (0.9031)    Prec 72.656% (68.680%)
Epoch: [12][200/391]    Time 0.096 (0.104)      Data 0.001 (0.004)      Loss
0.7088 (0.9093)    Prec 79.688% (68.381%)
Epoch: [12][300/391]    Time 0.103 (0.104)      Data 0.001 (0.003)      Loss
0.6894 (0.9070)    Prec 74.219% (68.527%)
Validation starts
Test: [0/79]    Time 0.222 (0.222)      Loss 0.8878 (0.8878)    Prec 70.312%
(70.312%)
 * Prec 68.040%
best acc: 68.040000
Epoch: [13][0/391]      Time 0.287 (0.287)      Data 0.223 (0.223)      Loss
0.8856 (0.8856)    Prec 67.188% (67.188%)
Epoch: [13][100/391]    Time 0.101 (0.105)      Data 0.001 (0.004)      Loss
0.8264 (0.8842)    Prec 71.094% (69.377%)
Epoch: [13][200/391]    Time 0.104 (0.103)      Data 0.001 (0.003)      Loss
0.7214 (0.8897)    Prec 79.688% (69.345%)
Epoch: [13][300/391]    Time 0.103 (0.103)      Data 0.001 (0.002)      Loss
0.9290 (0.8853)    Prec 68.750% (69.386%)
Validation starts
Test: [0/79]    Time 0.254 (0.254)      Loss 1.0028 (1.0028)    Prec 67.969%
(67.969%)
 * Prec 63.960%
best acc: 68.040000
Epoch: [14][0/391]      Time 0.279 (0.279)      Data 0.214 (0.214)      Loss
1.0965 (1.0965)    Prec 66.406% (66.406%)
Epoch: [14][100/391]    Time 0.110 (0.105)      Data 0.002 (0.004)      Loss
0.8926 (0.8764)    Prec 73.438% (69.933%)
Epoch: [14][200/391]    Time 0.104 (0.104)      Data 0.003 (0.003)      Loss
0.8512 (0.8764)    Prec 72.656% (69.726%)
Epoch: [14][300/391]    Time 0.096 (0.103)      Data 0.002 (0.003)      Loss
0.8026 (0.8675)    Prec 68.750% (69.991%)
Validation starts
Test: [0/79]    Time 0.194 (0.194)      Loss 0.8373 (0.8373)    Prec 70.312%
(70.312%)
 * Prec 68.960%
best acc: 68.960000
Epoch: [15][0/391]      Time 0.250 (0.250)      Data 0.186 (0.186)      Loss
0.7448 (0.7448)    Prec 74.219% (74.219%)
Epoch: [15][100/391]    Time 0.103 (0.104)      Data 0.002 (0.003)      Loss
0.8469 (0.8572)    Prec 69.531% (70.405%)
Epoch: [15][200/391]    Time 0.101 (0.103)      Data 0.002 (0.003)      Loss
0.7532 (0.8576)    Prec 75.000% (70.600%)
Epoch: [15][300/391]    Time 0.103 (0.103)      Data 0.001 (0.002)      Loss
```

```
0.8957 (0.8552)    Prec 64.062% (70.629%)
Validation starts
Test: [0/79]    Time 0.287 (0.287)      Loss 0.7828 (0.7828)    Prec 75.000%
(75.000%)
 * Prec 70.770%
best acc: 70.770000
Epoch: [16][0/391]      Time 0.250 (0.250)      Data 0.191 (0.191)      Loss
0.8060 (0.8060)    Prec 71.875% (71.875%)
Epoch: [16][100/391]    Time 0.103 (0.104)      Data 0.002 (0.004)      Loss
0.9111 (0.8451)    Prec 66.406% (70.251%)
Epoch: [16][200/391]    Time 0.105 (0.104)      Data 0.003 (0.003)      Loss
0.8451 (0.8478)    Prec 71.094% (70.588%)
Epoch: [16][300/391]    Time 0.102 (0.103)      Data 0.001 (0.003)      Loss
0.8801 (0.8491)    Prec 68.750% (70.588%)
Validation starts
Test: [0/79]    Time 0.262 (0.262)      Loss 0.9833 (0.9833)    Prec 67.188%
(67.188%)
 * Prec 65.210%
best acc: 70.770000
Epoch: [17][0/391]      Time 0.324 (0.324)      Data 0.254 (0.254)      Loss
0.7656 (0.7656)    Prec 68.750% (68.750%)
Epoch: [17][100/391]    Time 0.101 (0.105)      Data 0.002 (0.004)      Loss
0.7791 (0.8423)    Prec 75.781% (70.846%)
Epoch: [17][200/391]    Time 0.102 (0.103)      Data 0.001 (0.003)      Loss
0.7724 (0.8436)    Prec 76.562% (70.872%)
Epoch: [17][300/391]    Time 0.099 (0.103)      Data 0.001 (0.002)      Loss
0.8244 (0.8392)    Prec 73.438% (71.161%)
Validation starts
Test: [0/79]    Time 0.216 (0.216)      Loss 0.8952 (0.8952)    Prec 70.312%
(70.312%)
 * Prec 70.330%
best acc: 70.770000
Epoch: [18][0/391]      Time 0.246 (0.246)      Data 0.182 (0.182)      Loss
0.8117 (0.8117)    Prec 76.562% (76.562%)
Epoch: [18][100/391]    Time 0.103 (0.104)      Data 0.003 (0.004)      Loss
0.8321 (0.8266)    Prec 68.750% (71.867%)
Epoch: [18][200/391]    Time 0.104 (0.104)      Data 0.002 (0.003)      Loss
0.7996 (0.8231)    Prec 78.125% (71.677%)
Epoch: [18][300/391]    Time 0.103 (0.103)      Data 0.002 (0.003)      Loss
0.9005 (0.8221)    Prec 69.531% (71.802%)
Validation starts
Test: [0/79]    Time 0.228 (0.228)      Loss 0.9194 (0.9194)    Prec 69.531%
(69.531%)
 * Prec 68.320%
best acc: 70.770000
Epoch: [19][0/391]      Time 0.287 (0.287)      Data 0.213 (0.213)      Loss
0.8314 (0.8314)    Prec 70.312% (70.312%)
Epoch: [19][100/391]    Time 0.100 (0.105)      Data 0.002 (0.004)      Loss
```

```
0.7049 (0.8264)    Prec 78.125% (71.341%)
Epoch: [19][200/391]    Time 0.105 (0.104)      Data 0.001 (0.003)      Loss
0.8770 (0.8095)    Prec 68.750% (71.972%)
Epoch: [19][300/391]    Time 0.099 (0.103)      Data 0.002 (0.003)      Loss
0.9072 (0.8070)    Prec 72.656% (72.067%)
Validation starts
Test: [0/79]    Time 0.273 (0.273)      Loss 0.9524 (0.9524)    Prec 67.188%
(67.188%)
 * Prec 67.550%
best acc: 70.770000
Epoch: [20][0/391]      Time 0.260 (0.260)      Data 0.193 (0.193)      Loss
0.8395 (0.8395)    Prec 67.969% (67.969%)
Epoch: [20][100/391]    Time 0.104 (0.105)      Data 0.002 (0.005)      Loss
0.7477 (0.7910)    Prec 70.312% (72.788%)
Epoch: [20][200/391]    Time 0.100 (0.104)      Data 0.003 (0.003)      Loss
0.7144 (0.7935)    Prec 74.219% (72.621%)
Epoch: [20][300/391]    Time 0.109 (0.104)      Data 0.002 (0.003)      Loss
0.8608 (0.8034)    Prec 71.875% (72.241%)
Validation starts
Test: [0/79]    Time 0.202 (0.202)      Loss 1.0277 (1.0277)    Prec 65.625%
(65.625%)
 * Prec 68.850%
best acc: 70.770000
Epoch: [21][0/391]      Time 0.269 (0.269)      Data 0.203 (0.203)      Loss
0.7917 (0.7917)    Prec 76.562% (76.562%)
Epoch: [21][100/391]    Time 0.103 (0.104)      Data 0.002 (0.004)      Loss
0.7529 (0.7955)    Prec 70.312% (72.563%)
Epoch: [21][200/391]    Time 0.101 (0.104)      Data 0.002 (0.003)      Loss
0.7533 (0.8019)    Prec 75.781% (72.470%)
Epoch: [21][300/391]    Time 0.102 (0.103)      Data 0.002 (0.003)      Loss
0.7123 (0.8011)    Prec 78.906% (72.423%)
Validation starts
Test: [0/79]    Time 0.258 (0.258)      Loss 0.7818 (0.7818)    Prec 71.875%
(71.875%)
 * Prec 70.920%
best acc: 70.920000
Epoch: [22][0/391]      Time 0.300 (0.300)      Data 0.232 (0.232)      Loss
0.6567 (0.6567)    Prec 78.906% (78.906%)
Epoch: [22][100/391]    Time 0.103 (0.105)      Data 0.001 (0.004)      Loss
0.7897 (0.7911)    Prec 75.781% (72.803%)
Epoch: [22][200/391]    Time 0.106 (0.103)      Data 0.002 (0.003)      Loss
0.8702 (0.7942)    Prec 69.531% (72.559%)
Epoch: [22][300/391]    Time 0.103 (0.103)      Data 0.002 (0.002)      Loss
0.8832 (0.7892)    Prec 68.750% (72.693%)
Validation starts
Test: [0/79]    Time 0.220 (0.220)      Loss 0.7841 (0.7841)    Prec 70.312%
(70.312%)
 * Prec 72.570%
```

```
best acc: 72.570000
Epoch: [23][0/391]      Time 0.245 (0.245)      Data 0.182 (0.182)      Loss
0.7174 (0.7174)    Prec 75.781% (75.781%)
Epoch: [23][100/391]    Time 0.106 (0.105)      Data 0.003 (0.004)      Loss
0.6919 (0.7742)    Prec 75.781% (73.584%)
Epoch: [23][200/391]    Time 0.108 (0.104)      Data 0.004 (0.003)      Loss
0.6773 (0.7788)    Prec 81.250% (73.434%)
Epoch: [23][300/391]    Time 0.101 (0.104)      Data 0.001 (0.003)      Loss
0.6819 (0.7767)    Prec 77.344% (73.409%)
Validation starts
Test: [0/79]    Time 0.234 (0.234)      Loss 0.9316 (0.9316)    Prec 72.656%
(72.656%)
 * Prec 70.900%
best acc: 72.570000
Epoch: [24][0/391]      Time 0.322 (0.322)      Data 0.254 (0.254)      Loss
0.7281 (0.7281)    Prec 77.344% (77.344%)
Epoch: [24][100/391]    Time 0.099 (0.105)      Data 0.001 (0.004)      Loss
0.8020 (0.7825)    Prec 71.875% (73.120%)
Epoch: [24][200/391]    Time 0.102 (0.104)      Data 0.002 (0.003)      Loss
0.7702 (0.7727)    Prec 71.094% (73.511%)
Epoch: [24][300/391]    Time 0.101 (0.103)      Data 0.002 (0.003)      Loss
0.6507 (0.7754)    Prec 79.688% (73.360%)
Validation starts
Test: [0/79]    Time 0.257 (0.257)      Loss 0.7447 (0.7447)    Prec 75.000%
(75.000%)
 * Prec 71.270%
best acc: 72.570000
Epoch: [25][0/391]      Time 0.261 (0.261)      Data 0.197 (0.197)      Loss
0.9355 (0.9355)    Prec 65.625% (65.625%)
Epoch: [25][100/391]    Time 0.099 (0.105)      Data 0.003 (0.004)      Loss
0.7132 (0.7588)    Prec 70.312% (73.770%)
Epoch: [25][200/391]    Time 0.109 (0.104)      Data 0.002 (0.003)      Loss
0.8724 (0.7595)    Prec 67.969% (73.958%)
Epoch: [25][300/391]    Time 0.103 (0.104)      Data 0.001 (0.003)      Loss
0.8541 (0.7631)    Prec 66.406% (73.684%)
Validation starts
Test: [0/79]    Time 0.227 (0.227)      Loss 0.8033 (0.8033)    Prec 75.000%
(75.000%)
 * Prec 72.170%
best acc: 72.570000
Epoch: [26][0/391]      Time 0.303 (0.303)      Data 0.234 (0.234)      Loss
0.7088 (0.7088)    Prec 75.781% (75.781%)
Epoch: [26][100/391]    Time 0.102 (0.105)      Data 0.002 (0.004)      Loss
0.6181 (0.7366)    Prec 78.125% (74.575%)
Epoch: [26][200/391]    Time 0.101 (0.104)      Data 0.002 (0.003)      Loss
0.7599 (0.7448)    Prec 72.656% (74.262%)
Epoch: [26][300/391]    Time 0.102 (0.103)      Data 0.002 (0.003)      Loss
0.7961 (0.7547)    Prec 75.000% (74.128%)
```

```
Validation starts
Test: [0/79]    Time 0.240 (0.240)      Loss 0.9146 (0.9146)    Prec 71.094%
(71.094%)
 * Prec 67.640%
best acc: 72.570000
Epoch: [27][0/391]      Time 0.284 (0.284)      Data 0.215 (0.215)      Loss
0.9884 (0.9884)    Prec 65.625% (65.625%)
Epoch: [27][100/391]    Time 0.102 (0.105)      Data 0.002 (0.004)      Loss
0.7534 (0.7503)    Prec 75.781% (73.871%)
Epoch: [27][200/391]    Time 0.102 (0.104)      Data 0.003 (0.003)      Loss
0.7319 (0.7546)    Prec 74.219% (73.756%)
Epoch: [27][300/391]    Time 0.103 (0.103)      Data 0.002 (0.003)      Loss
0.6814 (0.7540)    Prec 75.000% (73.863%)
Validation starts
Test: [0/79]    Time 0.198 (0.198)      Loss 0.7920 (0.7920)    Prec 72.656%
(72.656%)


      ␣
↪---------------------------------------------------------------------------

       KeyboardInterrupt                         Traceback (most recent call␣
↪last)

       /tmp/ipykernel_4662/653216638.py in <module>
        27     # evaluate on test set
        28     print("Validation starts")
   ---> 29     prec = validate(testloader, model, criterion)
        30
        31     # remember best precision and save checkpoint


       /tmp/ipykernel_4662/1701083248.py in validate(val_loader, model,␣
↪criterion)
       122
       123             # compute output
   --> 124             output = model(input)
       125             loss = criterion(output, target)
       126


       /opt/conda/lib/python3.9/site-packages/torch/nn/modules/module.py in␣
↪_call_impl(self, *input, **kwargs)
      1049         if not (self._backward_hooks or self._forward_hooks or self.
↪_forward_pre_hooks or _global_backward_hooks
      1050                 or _global_forward_hooks or␣
↪_global_forward_pre_hooks):
   -> 1051             return forward_call(*input, **kwargs)
```

```
      1052         # Do not call functions when jit is used
      1053         full_backward_hooks, non_full_backward_hooks = [], []


       ~/HW6/models/vgg_quant.py in forward(self, x)
        23
        24     def forward(self, x):
   ---> 25         out = self.features(x)
        26         out = out.view(out.size(0), -1)
        27         out = self.classifier(out)


       /opt/conda/lib/python3.9/site-packages/torch/nn/modules/module.py in␣
↪_call_impl(self, *input, **kwargs)
      1049         if not (self._backward_hooks or self._forward_hooks or self.
↪_forward_pre_hooks or _global_backward_hooks
      1050                 or _global_forward_hooks or␣
↪_global_forward_pre_hooks):
   -> 1051             return forward_call(*input, **kwargs)
      1052         # Do not call functions when jit is used
      1053         full_backward_hooks, non_full_backward_hooks = [], []


       /opt/conda/lib/python3.9/site-packages/torch/nn/modules/container.py in␣
↪forward(self, input)
       137     def forward(self, input):
       138         for module in self:
   --> 139             input = module(input)
       140         return input
       141


       /opt/conda/lib/python3.9/site-packages/torch/nn/modules/module.py in␣
↪_call_impl(self, *input, **kwargs)
      1069             input = bw_hook.setup_input_hook(input)
      1070
   -> 1071         result = forward_call(*input, **kwargs)
      1072         if _global_forward_hooks or self._forward_hooks:
      1073             for hook in itertools.chain(


       ~/HW6/models/quant_layer.py in forward(self, x)
       101
       102     def forward(self, x):
   --> 103         weight_q = self.weight_quant(self.weight)
       104         #self.register_parameter('weight_q', Parameter(weight_q))  #␣
↪Mingu added
```

```
       105         self.weight_q = torch.nn.Parameter(weight_q)  # Store␣
↪weight_q during the training


       /opt/conda/lib/python3.9/site-packages/torch/nn/modules/module.py in␣
↪_call_impl(self, *input, **kwargs)
      1049         if not (self._backward_hooks or self._forward_hooks or self.
↪_forward_pre_hooks or _global_backward_hooks
      1050                 or _global_forward_hooks or␣
↪_global_forward_pre_hooks):
   -> 1051             return forward_call(*input, **kwargs)
      1052         # Do not call functions when jit is used
      1053         full_backward_hooks, non_full_backward_hooks = [], []


       ~/HW6/models/quant_layer.py in forward(self, weight)
        53         mean = weight.data.mean()
        54         std = weight.data.std()
   ---> 55         weight = weight.add(-mean).div(std)      # weights␣
↪normalization
        56         weight_q = self.weight_q(weight, self.wgt_alpha)
        57


       KeyboardInterrupt:
```

[9]:
```
## check your accuracy again after finetuning
model.cuda()
model.eval()

test_loss = 0
correct = 0

with torch.no_grad():
    for data, target in testloader:
        data, target = data.to(device), target.to(device) # loading to GPU
        output = model(data)
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

test_loss /= len(testloader.dataset)

print('\nTest set: Accuracy: {}/{} ({:.0f}%)\n'.format(
        correct, len(testloader.dataset),
        100. * correct / len(testloader.dataset)))
```

```
Test set: Accuracy: 7128/10000 (71%)
```

[10]:
```
## Send an image and use prehook to grab the inputs of all the QuantConv2d␣
 ↪layers

class SaveOutput:
    def __init__(self):
        self.outputs = []
    def __call__(self, module, module_in):
        self.outputs.append(module_in)
    def clear(self):
        self.outputs = []

######### Save inputs from selected layer ##########
save_output = SaveOutput()
i = 0

for layer in model.modules():
    i = i+1
    if isinstance(layer, QuantConv2d):
        print(i,"-th layer prehooked")
        layer.register_forward_pre_hook(save_output)
####################################################

dataiter = iter(testloader)
images, labels = dataiter.next()
images = images.to(device)
out = model(images)
```

```
3 -th layer prehooked
7 -th layer prehooked
12 -th layer prehooked
16 -th layer prehooked
21 -th layer prehooked
25 -th layer prehooked
29 -th layer prehooked
34 -th layer prehooked
38 -th layer prehooked
42 -th layer prehooked
47 -th layer prehooked
51 -th layer prehooked
55 -th layer prehooked
```

[11]:
```
##### Find "weight_int" for features[3] ####
w_bit = 4
```

```
weight_q = model.features[3].weight_q
w_alpha = model.features[3].weight_quant.wgt_alpha
w_delta = w_alpha /(2**(w_bit-1)-1)

weight_int = weight_q / w_delta
print(weight_int)
```

```
tensor([[[[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         …,

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]]],


        [[[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         [[-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000],
          [-0.0000, -0.0000, -0.0000]],

         …,
```

```
 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]]],


[[[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 …,

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]]],


…,


[[[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],
```

```
 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 …,

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]]],


[[[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 …,

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]]],
```

```
[[[-2.0000,  3.0000,  1.0000],
  [-5.0000, -5.0000, -2.0000],
  [-5.0000, -0.0000,  4.0000]],

 [[ 0.0000, -0.0000, -0.0000],
  [ 0.0000, -1.0000, -0.0000],
  [-0.0000, -0.0000,  0.0000]],

 [[-0.0000, -1.0000, -0.0000],
  [-1.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 …,

 [[-2.0000, -2.0000, -1.0000],
  [-1.0000, -1.0000,  0.0000],
  [ 1.0000,  1.0000,  2.0000]],

 [[-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000],
  [-0.0000, -0.0000, -0.0000]],

 [[-1.0000, -0.0000,  0.0000],
  [-1.0000, -1.0000, -0.0000],
  [-1.0000, -0.0000,  1.0000]]]], device='cuda:0',
grad_fn=<DivBackward0>)
```

[12]:
```
#### check your sparsity for weight_int is near 90% #####
#### Your sparsity could be >90% after quantization #####
sparsity_weight_int = (weight_int == 0).sum() / weight_int.nelement()
print("Sparsity level: ", sparsity_weight_int)
```

```
Sparsity level:  tensor(0.9736, device='cuda:0')
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: