Project_resnet

November 28, 2022

[3]:

```
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


import torchvision
import torchvision.transforms as transforms

from models import *


global best_prec
use_gpu = torch.cuda.is_available()
device = torch.device("cuda")


batch_size = 128
model_name = "Resnet_20_quant_project"
model = resnet20_quant_project()


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)
```

```
Files already downloaded and verified
Files already downloaded and verified
```

[4]:

```
print_freq = 100
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,adjust_list):
    """For resnet, the lr starts from 0.1, and is divided by 10 at 80 and 120␣
 ↪epochs"""

    if epoch in adjust_list:
        for param_group in optimizer.param_groups:
            param_group['lr'] = param_group['lr'] * 0.1
```

[5]:

```
lr = 4e-2
weight_decay = 1e-4
epochs = 100
best_prec = 0
model.cuda()
criterion = nn.CrossEntropyLoss().cuda()
optimizer = torch.optim.SGD(model.parameters(), lr=lr, momentum=0.
 ↪9,weight_decay=weight_decay)

if not os.path.exists('result'):
    os.makedirs('result')

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

adjust_list = [80,90]

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

    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 0.723 (0.723)      Data 0.588 (0.588)      Loss
2.4852 (2.4852)    Prec 7.031% (7.031%)
Epoch: [0][100/391]     Time 0.055 (0.056)      Data 0.002 (0.008)      Loss
1.8671 (2.0559)    Prec 30.469% (21.047%)
Epoch: [0][200/391]     Time 0.051 (0.055)      Data 0.002 (0.005)      Loss
1.7032 (1.9121)    Prec 39.062% (27.247%)
Epoch: [0][300/391]     Time 0.051 (0.052)      Data 0.002 (0.004)      Loss
1.4083 (1.8174)    Prec 46.875% (31.341%)
Validation starts
Test: [0/79]    Time 0.354 (0.354)      Loss 1.5558 (1.5558)    Prec 42.188%
(42.188%)
 * Prec 41.410%
best acc: 41.410000
Epoch: [1][0/391]       Time 0.428 (0.428)      Data 0.362 (0.362)      Loss
1.4099 (1.4099)    Prec 43.750% (43.750%)
Epoch: [1][100/391]     Time 0.048 (0.052)      Data 0.002 (0.006)      Loss
1.3916 (1.4359)    Prec 50.000% (47.223%)
Epoch: [1][200/391]     Time 0.040 (0.051)      Data 0.001 (0.004)      Loss
1.4276 (1.3768)    Prec 46.094% (49.938%)
Epoch: [1][300/391]     Time 0.050 (0.049)      Data 0.002 (0.003)      Loss
1.1543 (1.3324)    Prec 60.156% (51.744%)
Validation starts
Test: [0/79]    Time 0.426 (0.426)      Loss 1.3888 (1.3888)    Prec 53.125%
(53.125%)
 * Prec 52.660%
best acc: 52.660000
Epoch: [2][0/391]       Time 0.756 (0.756)      Data 0.692 (0.692)      Loss
1.1296 (1.1296)    Prec 58.594% (58.594%)
Epoch: [2][100/391]     Time 0.057 (0.060)      Data 0.003 (0.009)      Loss
1.2002 (1.1215)    Prec 61.719% (60.288%)
Epoch: [2][200/391]     Time 0.050 (0.056)      Data 0.003 (0.006)      Loss
0.9285 (1.0979)    Prec 66.406% (61.050%)
Epoch: [2][300/391]     Time 0.043 (0.054)      Data 0.002 (0.005)      Loss
0.8716 (1.0763)    Prec 66.406% (61.768%)
Validation starts
Test: [0/79]    Time 0.409 (0.409)      Loss 1.1291 (1.1291)    Prec 60.938%
(60.938%)
 * Prec 60.810%
best acc: 60.810000
```

```
Epoch: [3][0/391]       Time 0.659 (0.659)      Data 0.595 (0.595)      Loss
1.0072 (1.0072)    Prec 65.625% (65.625%)
Epoch: [3][100/391]     Time 0.050 (0.059)      Data 0.002 (0.008)      Loss
1.0853 (0.9882)    Prec 60.156% (65.347%)
Epoch: [3][200/391]     Time 0.055 (0.057)      Data 0.003 (0.005)      Loss
0.9933 (0.9642)    Prec 67.188% (65.920%)
Epoch: [3][300/391]     Time 0.053 (0.055)      Data 0.002 (0.004)      Loss
0.8162 (0.9502)    Prec 72.656% (66.562%)
Validation starts
Test: [0/79]    Time 0.873 (0.873)      Loss 0.7776 (0.7776)    Prec 73.438%
(73.438%)
 * Prec 69.230%
best acc: 69.230000
Epoch: [4][0/391]       Time 0.595 (0.595)      Data 0.531 (0.531)      Loss
0.8532 (0.8532)    Prec 63.281% (63.281%)
Epoch: [4][100/391]     Time 0.055 (0.057)      Data 0.003 (0.008)      Loss
0.8541 (0.8590)    Prec 66.406% (69.531%)
Epoch: [4][200/391]     Time 0.046 (0.054)      Data 0.002 (0.005)      Loss
0.7850 (0.8652)    Prec 69.531% (69.551%)
Epoch: [4][300/391]     Time 0.048 (0.054)      Data 0.001 (0.004)      Loss
0.9558 (0.8609)    Prec 66.406% (69.640%)
Validation starts
Test: [0/79]    Time 0.465 (0.465)      Loss 0.7475 (0.7475)    Prec 71.875%
(71.875%)
 * Prec 70.440%
best acc: 70.440000
Epoch: [5][0/391]       Time 0.513 (0.513)      Data 0.448 (0.448)      Loss
0.5259 (0.5259)    Prec 84.375% (84.375%)
Epoch: [5][100/391]     Time 0.056 (0.057)      Data 0.002 (0.007)      Loss
0.6973 (0.8027)    Prec 78.125% (71.937%)
Epoch: [5][200/391]     Time 0.046 (0.053)      Data 0.002 (0.004)      Loss
0.8067 (0.8005)    Prec 71.875% (71.801%)
Epoch: [5][300/391]     Time 0.049 (0.053)      Data 0.002 (0.004)      Loss
0.6789 (0.7961)    Prec 78.125% (72.098%)
Validation starts
Test: [0/79]    Time 0.529 (0.529)      Loss 0.8746 (0.8746)    Prec 70.312%
(70.312%)
 * Prec 66.890%
best acc: 70.440000
Epoch: [6][0/391]       Time 0.467 (0.467)      Data 0.401 (0.401)      Loss
0.7818 (0.7818)    Prec 74.219% (74.219%)
Epoch: [6][100/391]     Time 0.056 (0.061)      Data 0.002 (0.006)      Loss
0.7240 (0.7521)    Prec 74.219% (73.693%)
Epoch: [6][200/391]     Time 0.048 (0.057)      Data 0.002 (0.004)      Loss
0.6805 (0.7511)    Prec 74.219% (73.640%)
Epoch: [6][300/391]     Time 0.052 (0.055)      Data 0.002 (0.004)      Loss
0.7250 (0.7553)    Prec 69.531% (73.544%)
Validation starts
```

```
Test: [0/79]    Time 0.440 (0.440)      Loss 0.8272 (0.8272)    Prec 73.438%
(73.438%)
 * Prec 69.660%
best acc: 70.440000
Epoch: [7][0/391]       Time 0.585 (0.585)      Data 0.520 (0.520)      Loss
0.6912 (0.6912)    Prec 75.000% (75.000%)
Epoch: [7][100/391]     Time 0.058 (0.060)      Data 0.002 (0.007)      Loss
0.7684 (0.7198)    Prec 75.000% (75.217%)
Epoch: [7][200/391]     Time 0.054 (0.057)      Data 0.002 (0.005)      Loss
0.6532 (0.7276)    Prec 75.781% (74.953%)
Epoch: [7][300/391]     Time 0.055 (0.056)      Data 0.003 (0.004)      Loss
0.6186 (0.7290)    Prec 82.031% (74.777%)
Validation starts
Test: [0/79]    Time 0.667 (0.667)      Loss 0.8228 (0.8228)    Prec 71.875%
(71.875%)
 * Prec 71.500%
best acc: 71.500000
Epoch: [8][0/391]       Time 0.534 (0.534)      Data 0.469 (0.469)      Loss
0.6864 (0.6864)    Prec 77.344% (77.344%)
Epoch: [8][100/391]     Time 0.053 (0.057)      Data 0.002 (0.007)      Loss
0.6234 (0.6682)    Prec 76.562% (76.439%)
Epoch: [8][200/391]     Time 0.051 (0.055)      Data 0.002 (0.005)      Loss
0.5489 (0.6639)    Prec 80.469% (76.905%)
Epoch: [8][300/391]     Time 0.053 (0.055)      Data 0.002 (0.004)      Loss
0.6218 (0.6733)    Prec 76.562% (76.575%)
Validation starts
Test: [0/79]    Time 0.389 (0.389)      Loss 1.0358 (1.0358)    Prec 65.625%
(65.625%)
 * Prec 66.550%
best acc: 71.500000
Epoch: [9][0/391]       Time 0.606 (0.606)      Data 0.544 (0.544)      Loss
0.5445 (0.5445)    Prec 81.250% (81.250%)
Epoch: [9][100/391]     Time 0.050 (0.055)      Data 0.002 (0.008)      Loss
0.5818 (0.6570)    Prec 80.469% (76.733%)
Epoch: [9][200/391]     Time 0.047 (0.052)      Data 0.002 (0.005)      Loss
0.7932 (0.6664)    Prec 71.094% (76.481%)
Epoch: [9][300/391]     Time 0.048 (0.051)      Data 0.002 (0.004)      Loss
0.6268 (0.6646)    Prec 78.125% (76.692%)
Validation starts
Test: [0/79]    Time 0.526 (0.526)      Loss 0.7717 (0.7717)    Prec 73.438%
(73.438%)
 * Prec 72.790%
best acc: 72.790000
Epoch: [10][0/391]      Time 0.523 (0.523)      Data 0.458 (0.458)      Loss
0.6118 (0.6118)    Prec 78.906% (78.906%)
Epoch: [10][100/391]    Time 0.052 (0.054)      Data 0.002 (0.007)      Loss
0.6333 (0.6346)    Prec 79.688% (77.916%)
Epoch: [10][200/391]    Time 0.053 (0.054)      Data 0.003 (0.005)      Loss
```

```
0.7144 (0.6350)    Prec 76.562% (78.012%)
Epoch: [10][300/391]    Time 0.059 (0.054)      Data 0.002 (0.004)      Loss
0.6236 (0.6385)    Prec 77.344% (77.876%)
Validation starts
Test: [0/79]    Time 1.304 (1.304)      Loss 0.9510 (0.9510)    Prec 68.750%
(68.750%)
 * Prec 67.620%
best acc: 72.790000
Epoch: [11][0/391]      Time 0.451 (0.451)      Data 0.389 (0.389)      Loss
0.6256 (0.6256)    Prec 76.562% (76.562%)
Epoch: [11][100/391]    Time 0.063 (0.056)      Data 0.003 (0.006)      Loss
0.6800 (0.6013)    Prec 74.219% (78.991%)
Epoch: [11][200/391]    Time 0.051 (0.055)      Data 0.002 (0.004)      Loss
0.6946 (0.6140)    Prec 75.781% (78.751%)
Epoch: [11][300/391]    Time 0.059 (0.054)      Data 0.003 (0.004)      Loss
0.5900 (0.6118)    Prec 82.031% (78.761%)
Validation starts
Test: [0/79]    Time 0.638 (0.638)      Loss 0.7047 (0.7047)    Prec 74.219%
(74.219%)
 * Prec 72.720%
best acc: 72.790000
Epoch: [12][0/391]      Time 0.558 (0.558)      Data 0.463 (0.463)      Loss
0.4608 (0.4608)    Prec 84.375% (84.375%)
Epoch: [12][100/391]    Time 0.050 (0.059)      Data 0.002 (0.007)      Loss
0.4683 (0.5877)    Prec 80.469% (79.394%)
Epoch: [12][200/391]    Time 0.056 (0.056)      Data 0.002 (0.005)      Loss
0.4336 (0.5840)    Prec 85.938% (79.660%)
Epoch: [12][300/391]    Time 0.059 (0.054)      Data 0.002 (0.004)      Loss
0.7537 (0.5889)    Prec 78.906% (79.472%)
Validation starts
Test: [0/79]    Time 0.441 (0.441)      Loss 0.6678 (0.6678)    Prec 78.906%
(78.906%)
 * Prec 77.110%
best acc: 77.110000
Epoch: [13][0/391]      Time 0.573 (0.573)      Data 0.508 (0.508)      Loss
0.6092 (0.6092)    Prec 78.906% (78.906%)
Epoch: [13][100/391]    Time 0.049 (0.057)      Data 0.002 (0.007)      Loss
0.6873 (0.5749)    Prec 78.125% (80.067%)
Epoch: [13][200/391]    Time 0.055 (0.055)      Data 0.002 (0.005)      Loss
0.6250 (0.5819)    Prec 78.125% (79.796%)
Epoch: [13][300/391]    Time 0.057 (0.054)      Data 0.002 (0.004)      Loss
0.4878 (0.5842)    Prec 82.812% (79.698%)
Validation starts
Test: [0/79]    Time 0.359 (0.359)      Loss 0.7049 (0.7049)    Prec 78.906%
(78.906%)
 * Prec 75.850%
best acc: 77.110000
Epoch: [14][0/391]      Time 0.821 (0.821)      Data 0.756 (0.756)      Loss
```

```
0.6366 (0.6366)    Prec 78.906% (78.906%)
Epoch: [14][100/391]    Time 0.056 (0.061)      Data 0.003 (0.010)      Loss
0.5679 (0.5564)    Prec 78.125% (80.678%)
Epoch: [14][200/391]    Time 0.045 (0.055)      Data 0.002 (0.006)      Loss
0.5897 (0.5631)    Prec 74.219% (80.492%)
Epoch: [14][300/391]    Time 0.057 (0.053)      Data 0.003 (0.005)      Loss
0.4574 (0.5612)    Prec 85.156% (80.565%)
Validation starts
Test: [0/79]    Time 0.510 (0.510)      Loss 0.5636 (0.5636)    Prec 80.469%
(80.469%)
 * Prec 77.640%
best acc: 77.640000
Epoch: [15][0/391]      Time 0.494 (0.494)      Data 0.431 (0.431)      Loss
0.6154 (0.6154)    Prec 76.562% (76.562%)
Epoch: [15][100/391]    Time 0.051 (0.058)      Data 0.002 (0.007)      Loss
0.6685 (0.5574)    Prec 77.344% (80.492%)
Epoch: [15][200/391]    Time 0.054 (0.056)      Data 0.002 (0.004)      Loss
0.5135 (0.5601)    Prec 83.594% (80.496%)
Epoch: [15][300/391]    Time 0.062 (0.053)      Data 0.002 (0.004)      Loss
0.4505 (0.5562)    Prec 82.812% (80.721%)
Validation starts
Test: [0/79]    Time 0.483 (0.483)      Loss 0.6223 (0.6223)    Prec 80.469%
(80.469%)
 * Prec 78.700%
best acc: 78.700000
Epoch: [16][0/391]      Time 0.455 (0.455)      Data 0.393 (0.393)      Loss
0.5377 (0.5377)    Prec 84.375% (84.375%)
Epoch: [16][100/391]    Time 0.039 (0.049)      Data 0.001 (0.006)      Loss
0.4802 (0.5335)    Prec 83.594% (81.706%)
Epoch: [16][200/391]    Time 0.051 (0.049)      Data 0.002 (0.004)      Loss
0.4999 (0.5435)    Prec 80.469% (81.285%)
Epoch: [16][300/391]    Time 0.051 (0.049)      Data 0.002 (0.003)      Loss
0.4867 (0.5454)    Prec 78.125% (81.206%)
Validation starts
Test: [0/79]    Time 0.281 (0.281)      Loss 0.5143 (0.5143)    Prec 82.812%
(82.812%)
 * Prec 78.900%
best acc: 78.900000
Epoch: [17][0/391]      Time 0.479 (0.479)      Data 0.390 (0.390)      Loss
0.6850 (0.6850)    Prec 77.344% (77.344%)
Epoch: [17][100/391]    Time 0.041 (0.052)      Data 0.002 (0.006)      Loss
0.5936 (0.5437)    Prec 75.000% (81.273%)
Epoch: [17][200/391]    Time 0.043 (0.050)      Data 0.002 (0.004)      Loss
0.4822 (0.5315)    Prec 83.594% (81.522%)
Epoch: [17][300/391]    Time 0.051 (0.049)      Data 0.002 (0.003)      Loss
0.4882 (0.5284)    Prec 87.500% (81.717%)
Validation starts
Test: [0/79]    Time 0.368 (0.368)      Loss 0.6041 (0.6041)    Prec 79.688%
```

```
(79.688%)
 * Prec 78.790%
best acc: 78.900000
Epoch: [18][0/391]      Time 0.438 (0.438)      Data 0.370 (0.370)      Loss
0.5500 (0.5500)    Prec 78.906% (78.906%)
Epoch: [18][100/391]    Time 0.035 (0.048)      Data 0.002 (0.006)      Loss
0.5410 (0.5185)    Prec 79.688% (81.877%)
Epoch: [18][200/391]    Time 0.039 (0.046)      Data 0.001 (0.004)      Loss
0.5384 (0.5094)    Prec 80.469% (82.210%)
Epoch: [18][300/391]    Time 0.051 (0.047)      Data 0.002 (0.003)      Loss
0.3522 (0.5106)    Prec 85.938% (82.257%)
Validation starts
Test: [0/79]    Time 0.402 (0.402)      Loss 0.4263 (0.4263)    Prec 85.156%
(85.156%)
 * Prec 80.920%
best acc: 80.920000
Epoch: [19][0/391]      Time 0.564 (0.564)      Data 0.507 (0.507)      Loss
0.5488 (0.5488)    Prec 77.344% (77.344%)
Epoch: [19][100/391]    Time 0.064 (0.052)      Data 0.003 (0.007)      Loss
0.4193 (0.5000)    Prec 86.719% (82.851%)
Epoch: [19][200/391]    Time 0.057 (0.048)      Data 0.003 (0.004)      Loss
0.7118 (0.4995)    Prec 75.000% (82.618%)
Epoch: [19][300/391]    Time 0.041 (0.048)      Data 0.001 (0.004)      Loss
0.5429 (0.4973)    Prec 80.469% (82.688%)
Validation starts
Test: [0/79]    Time 0.635 (0.635)      Loss 0.5854 (0.5854)    Prec 81.250%
(81.250%)
 * Prec 78.310%
best acc: 80.920000
Epoch: [20][0/391]      Time 0.625 (0.625)      Data 0.565 (0.565)      Loss
0.5191 (0.5191)    Prec 83.594% (83.594%)
Epoch: [20][100/391]    Time 0.044 (0.053)      Data 0.001 (0.008)      Loss
0.3865 (0.4913)    Prec 87.500% (82.758%)
Epoch: [20][200/391]    Time 0.051 (0.051)      Data 0.002 (0.005)      Loss
0.5962 (0.4881)    Prec 79.688% (82.937%)
Epoch: [20][300/391]    Time 0.048 (0.050)      Data 0.003 (0.004)      Loss
0.6095 (0.4866)    Prec 77.344% (82.981%)
Validation starts
Test: [0/79]    Time 0.271 (0.271)      Loss 0.5186 (0.5186)    Prec 79.688%
(79.688%)
 * Prec 81.200%
best acc: 81.200000
Epoch: [21][0/391]      Time 0.586 (0.586)      Data 0.526 (0.526)      Loss
0.3842 (0.3842)    Prec 87.500% (87.500%)
Epoch: [21][100/391]    Time 0.052 (0.058)      Data 0.002 (0.008)      Loss
0.5291 (0.4863)    Prec 83.594% (83.284%)
Epoch: [21][200/391]    Time 0.051 (0.053)      Data 0.002 (0.005)      Loss
0.4660 (0.4776)    Prec 83.594% (83.664%)
```

```
Epoch: [21][300/391]    Time 0.053 (0.053)      Data 0.002 (0.004)      Loss
0.4639 (0.4778)    Prec 85.156% (83.524%)
Validation starts
Test: [0/79]    Time 0.502 (0.502)      Loss 0.6461 (0.6461)    Prec 78.125%
(78.125%)
 * Prec 77.080%
best acc: 81.200000
Epoch: [22][0/391]      Time 0.501 (0.501)      Data 0.440 (0.440)      Loss
0.4528 (0.4528)    Prec 85.938% (85.938%)
Epoch: [22][100/391]    Time 0.044 (0.054)      Data 0.002 (0.007)      Loss
0.2880 (0.4575)    Prec 93.750% (84.135%)
Epoch: [22][200/391]    Time 0.055 (0.052)      Data 0.003 (0.004)      Loss
0.4604 (0.4609)    Prec 85.156% (84.033%)
Epoch: [22][300/391]    Time 0.049 (0.051)      Data 0.002 (0.004)      Loss
0.5867 (0.4741)    Prec 78.906% (83.599%)
Validation starts
Test: [0/79]    Time 0.448 (0.448)      Loss 0.4444 (0.4444)    Prec 84.375%
(84.375%)
 * Prec 80.510%
best acc: 81.200000
Epoch: [23][0/391]      Time 0.663 (0.663)      Data 0.598 (0.598)      Loss
0.4956 (0.4956)    Prec 84.375% (84.375%)
Epoch: [23][100/391]    Time 0.044 (0.053)      Data 0.002 (0.008)      Loss
0.4670 (0.4576)    Prec 85.938% (84.073%)
Epoch: [23][200/391]    Time 0.039 (0.048)      Data 0.002 (0.005)      Loss
0.5568 (0.4650)    Prec 80.469% (83.975%)
Epoch: [23][300/391]    Time 0.036 (0.046)      Data 0.002 (0.004)      Loss
0.4672 (0.4690)    Prec 82.031% (83.786%)
Validation starts
Test: [0/79]    Time 0.292 (0.292)      Loss 0.6100 (0.6100)    Prec 77.344%
(77.344%)
 * Prec 76.010%
best acc: 81.200000
Epoch: [24][0/391]      Time 0.424 (0.424)      Data 0.363 (0.363)      Loss
0.4234 (0.4234)    Prec 86.719% (86.719%)
Epoch: [24][100/391]    Time 0.042 (0.046)      Data 0.002 (0.005)      Loss
0.5697 (0.4535)    Prec 79.688% (83.934%)
Epoch: [24][200/391]    Time 0.046 (0.044)      Data 0.002 (0.004)      Loss
0.4184 (0.4535)    Prec 85.938% (84.049%)
Epoch: [24][300/391]    Time 0.053 (0.043)      Data 0.002 (0.003)      Loss
0.3722 (0.4617)    Prec 85.938% (83.807%)
Validation starts
Test: [0/79]    Time 0.403 (0.403)      Loss 0.6277 (0.6277)    Prec 78.906%
(78.906%)
 * Prec 78.560%
best acc: 81.200000
Epoch: [25][0/391]      Time 0.452 (0.452)      Data 0.390 (0.390)      Loss
0.4773 (0.4773)    Prec 82.031% (82.031%)
```

```
Epoch: [25][100/391]    Time 0.048 (0.046)      Data 0.002 (0.006)      Loss
0.4087 (0.4403)    Prec 85.938% (84.669%)
Epoch: [25][200/391]    Time 0.046 (0.049)      Data 0.002 (0.004)      Loss
0.6309 (0.4520)    Prec 78.125% (84.301%)
Epoch: [25][300/391]    Time 0.039 (0.047)      Data 0.002 (0.003)      Loss
0.4316 (0.4551)    Prec 87.500% (84.230%)
Validation starts
Test: [0/79]    Time 0.426 (0.426)      Loss 0.5048 (0.5048)    Prec 78.906%
(78.906%)
 * Prec 80.550%
best acc: 81.200000
Epoch: [26][0/391]      Time 0.479 (0.479)      Data 0.392 (0.392)      Loss
0.5335 (0.5335)    Prec 82.031% (82.031%)
Epoch: [26][100/391]    Time 0.058 (0.055)      Data 0.002 (0.006)      Loss
0.3149 (0.4361)    Prec 87.500% (84.916%)
Epoch: [26][200/391]    Time 0.057 (0.054)      Data 0.002 (0.004)      Loss
0.4211 (0.4416)    Prec 83.594% (84.748%)
Epoch: [26][300/391]    Time 0.059 (0.054)      Data 0.002 (0.004)      Loss
0.3963 (0.4464)    Prec 88.281% (84.611%)
Validation starts
Test: [0/79]    Time 0.630 (0.630)      Loss 0.4889 (0.4889)    Prec 82.812%
(82.812%)
 * Prec 80.560%
best acc: 81.200000
Epoch: [27][0/391]      Time 0.608 (0.608)      Data 0.542 (0.542)      Loss
0.4655 (0.4655)    Prec 82.031% (82.031%)
Epoch: [27][100/391]    Time 0.059 (0.059)      Data 0.003 (0.008)      Loss
0.3607 (0.4290)    Prec 86.719% (84.940%)
Epoch: [27][200/391]    Time 0.055 (0.056)      Data 0.003 (0.005)      Loss
0.3756 (0.4351)    Prec 85.938% (84.768%)
Epoch: [27][300/391]    Time 0.055 (0.055)      Data 0.003 (0.004)      Loss
0.3910 (0.4373)    Prec 88.281% (84.731%)
Validation starts
Test: [0/79]    Time 0.392 (0.392)      Loss 0.4796 (0.4796)    Prec 84.375%
(84.375%)
 * Prec 78.330%
best acc: 81.200000
Epoch: [28][0/391]      Time 0.452 (0.452)      Data 0.394 (0.394)      Loss
0.4482 (0.4482)    Prec 86.719% (86.719%)
Epoch: [28][100/391]    Time 0.046 (0.053)      Data 0.003 (0.006)      Loss
0.4714 (0.4219)    Prec 84.375% (85.326%)
Epoch: [28][200/391]    Time 0.047 (0.051)      Data 0.002 (0.004)      Loss
0.4539 (0.4238)    Prec 85.156% (85.215%)
Epoch: [28][300/391]    Time 0.045 (0.049)      Data 0.002 (0.003)      Loss
0.5435 (0.4297)    Prec 78.906% (85.094%)
Validation starts
Test: [0/79]    Time 0.418 (0.418)      Loss 0.6876 (0.6876)    Prec 80.469%
(80.469%)
```

```
 * Prec 75.580%
best acc: 81.200000
Epoch: [29][0/391]      Time 0.649 (0.649)      Data 0.582 (0.582)      Loss
0.2729 (0.2729)    Prec 91.406% (91.406%)
Epoch: [29][100/391]    Time 0.051 (0.057)      Data 0.001 (0.008)      Loss
0.5360 (0.4329)    Prec 81.250% (84.568%)
Epoch: [29][200/391]    Time 0.044 (0.050)      Data 0.002 (0.005)      Loss
0.5160 (0.4319)    Prec 85.156% (84.779%)
Epoch: [29][300/391]    Time 0.042 (0.047)      Data 0.002 (0.004)      Loss
0.3365 (0.4297)    Prec 89.844% (84.943%)
Validation starts
Test: [0/79]    Time 0.272 (0.272)      Loss 0.4617 (0.4617)    Prec 85.156%
(85.156%)
 * Prec 81.000%
best acc: 81.200000
Epoch: [30][0/391]      Time 0.523 (0.523)      Data 0.425 (0.425)      Loss
0.4473 (0.4473)    Prec 84.375% (84.375%)
Epoch: [30][100/391]    Time 0.040 (0.051)      Data 0.002 (0.006)      Loss
0.3495 (0.4219)    Prec 88.281% (85.520%)
Epoch: [30][200/391]    Time 0.047 (0.047)      Data 0.002 (0.004)      Loss
0.3759 (0.4178)    Prec 88.281% (85.623%)
Epoch: [30][300/391]    Time 0.050 (0.047)      Data 0.001 (0.003)      Loss
0.4753 (0.4221)    Prec 82.031% (85.395%)
Validation starts
Test: [0/79]    Time 0.518 (0.518)      Loss 0.4748 (0.4748)    Prec 84.375%
(84.375%)
 * Prec 81.740%
best acc: 81.740000
Epoch: [31][0/391]      Time 0.419 (0.419)      Data 0.356 (0.356)      Loss
0.4778 (0.4778)    Prec 85.938% (85.938%)
Epoch: [31][100/391]    Time 0.051 (0.055)      Data 0.002 (0.006)      Loss
0.4700 (0.4069)    Prec 85.156% (85.999%)
Epoch: [31][200/391]    Time 0.051 (0.053)      Data 0.002 (0.004)      Loss
0.4383 (0.4068)    Prec 81.250% (85.669%)
Epoch: [31][300/391]    Time 0.066 (0.053)      Data 0.003 (0.003)      Loss
0.3968 (0.4116)    Prec 85.156% (85.501%)
Validation starts
Test: [0/79]    Time 0.364 (0.364)      Loss 0.4586 (0.4586)    Prec 85.156%
(85.156%)
 * Prec 81.140%
best acc: 81.740000
Epoch: [32][0/391]      Time 1.215 (1.215)      Data 1.167 (1.167)      Loss
0.4458 (0.4458)    Prec 85.938% (85.938%)
Epoch: [32][100/391]    Time 0.051 (0.060)      Data 0.002 (0.014)      Loss
0.4414 (0.3949)    Prec 82.812% (86.054%)
Epoch: [32][200/391]    Time 0.051 (0.056)      Data 0.002 (0.008)      Loss
0.3734 (0.3972)    Prec 86.719% (85.988%)
Epoch: [32][300/391]    Time 0.053 (0.055)      Data 0.002 (0.006)      Loss
```

```
0.3841 (0.4038)    Prec 88.281% (85.828%)
Validation starts
Test: [0/79]    Time 0.479 (0.479)      Loss 0.4150 (0.4150)    Prec 85.156%
(85.156%)
 * Prec 82.050%
best acc: 82.050000
Epoch: [33][0/391]      Time 0.450 (0.450)      Data 0.354 (0.354)      Loss
0.5132 (0.5132)    Prec 83.594% (83.594%)
Epoch: [33][100/391]    Time 0.051 (0.052)      Data 0.002 (0.006)      Loss
0.4137 (0.4132)    Prec 84.375% (85.528%)
Epoch: [33][200/391]    Time 0.058 (0.049)      Data 0.003 (0.004)      Loss
0.4257 (0.4146)    Prec 85.938% (85.397%)
Epoch: [33][300/391]    Time 0.054 (0.049)      Data 0.003 (0.003)      Loss
0.3985 (0.4122)    Prec 85.156% (85.460%)
Validation starts
Test: [0/79]    Time 0.324 (0.324)      Loss 0.4655 (0.4655)    Prec 83.594%
(83.594%)
 * Prec 79.300%
best acc: 82.050000
Epoch: [34][0/391]      Time 0.544 (0.544)      Data 0.496 (0.496)      Loss
0.3356 (0.3356)    Prec 86.719% (86.719%)
Epoch: [34][100/391]    Time 0.048 (0.052)      Data 0.002 (0.007)      Loss
0.4791 (0.4111)    Prec 82.812% (85.636%)
Epoch: [34][200/391]    Time 0.052 (0.047)      Data 0.002 (0.004)      Loss
0.4486 (0.4037)    Prec 82.812% (85.708%)
Epoch: [34][300/391]    Time 0.038 (0.045)      Data 0.001 (0.004)      Loss
0.3099 (0.3957)    Prec 90.625% (86.036%)
Validation starts
Test: [0/79]    Time 0.481 (0.481)      Loss 0.5298 (0.5298)    Prec 81.250%
(81.250%)
 * Prec 82.130%
best acc: 82.130000
Epoch: [35][0/391]      Time 0.621 (0.621)      Data 0.559 (0.559)      Loss
0.4270 (0.4270)    Prec 85.938% (85.938%)
Epoch: [35][100/391]    Time 0.055 (0.050)      Data 0.002 (0.007)      Loss
0.3185 (0.3930)    Prec 89.062% (86.146%)
Epoch: [35][200/391]    Time 0.054 (0.049)      Data 0.002 (0.005)      Loss
0.4849 (0.4027)    Prec 81.250% (85.759%)
Epoch: [35][300/391]    Time 0.058 (0.049)      Data 0.003 (0.004)      Loss
0.3021 (0.4022)    Prec 89.844% (85.870%)
Validation starts
Test: [0/79]    Time 0.389 (0.389)      Loss 0.4776 (0.4776)    Prec 86.719%
(86.719%)
 * Prec 79.750%
best acc: 82.130000
Epoch: [36][0/391]      Time 0.536 (0.536)      Data 0.488 (0.488)      Loss
0.3817 (0.3817)    Prec 88.281% (88.281%)
Epoch: [36][100/391]    Time 0.052 (0.055)      Data 0.003 (0.007)      Loss
```

```
0.3319 (0.3734)    Prec 87.500% (87.044%)
Epoch: [36][200/391]    Time 0.043 (0.054)      Data 0.002 (0.005)      Loss
0.4001 (0.3830)    Prec 85.156% (86.532%)
Epoch: [36][300/391]    Time 0.049 (0.054)      Data 0.002 (0.004)      Loss
0.2560 (0.3865)    Prec 92.188% (86.449%)
Validation starts
Test: [0/79]    Time 0.360 (0.360)      Loss 0.3966 (0.3966)    Prec 85.156%
(85.156%)
 * Prec 80.600%
best acc: 82.130000
Epoch: [37][0/391]      Time 0.458 (0.458)      Data 0.392 (0.392)      Loss
0.5006 (0.5006)    Prec 80.469% (80.469%)
Epoch: [37][100/391]    Time 0.046 (0.056)      Data 0.002 (0.006)      Loss
0.4404 (0.3768)    Prec 82.031% (86.502%)
Epoch: [37][200/391]    Time 0.055 (0.055)      Data 0.002 (0.004)      Loss
0.3262 (0.3808)    Prec 89.062% (86.416%)
Epoch: [37][300/391]    Time 0.038 (0.054)      Data 0.002 (0.003)      Loss
0.3136 (0.3873)    Prec 87.500% (86.200%)
Validation starts
Test: [0/79]    Time 0.557 (0.557)      Loss 0.3437 (0.3437)    Prec 89.844%
(89.844%)
 * Prec 82.940%
best acc: 82.940000
Epoch: [38][0/391]      Time 0.263 (0.263)      Data 0.200 (0.200)      Loss
0.3633 (0.3633)    Prec 88.281% (88.281%)
Epoch: [38][100/391]    Time 0.045 (0.048)      Data 0.002 (0.004)      Loss
0.3367 (0.3619)    Prec 88.281% (87.492%)
Epoch: [38][200/391]    Time 0.050 (0.050)      Data 0.002 (0.003)      Loss
0.4304 (0.3723)    Prec 81.250% (86.995%)
Epoch: [38][300/391]    Time 0.051 (0.051)      Data 0.002 (0.003)      Loss
0.2578 (0.3825)    Prec 90.625% (86.682%)
Validation starts
Test: [0/79]    Time 0.342 (0.342)      Loss 0.6482 (0.6482)    Prec 78.906%
(78.906%)
 * Prec 78.460%
best acc: 82.940000
Epoch: [39][0/391]      Time 0.414 (0.414)      Data 0.363 (0.363)      Loss
0.3562 (0.3562)    Prec 89.062% (89.062%)
Epoch: [39][100/391]    Time 0.053 (0.049)      Data 0.002 (0.006)      Loss
0.4518 (0.3901)    Prec 87.500% (86.317%)
Epoch: [39][200/391]    Time 0.054 (0.051)      Data 0.002 (0.004)      Loss
0.3538 (0.3882)    Prec 88.281% (86.388%)
Epoch: [39][300/391]    Time 0.053 (0.052)      Data 0.002 (0.003)      Loss
0.3353 (0.3837)    Prec 89.062% (86.483%)
Validation starts
Test: [0/79]    Time 0.396 (0.396)      Loss 0.5270 (0.5270)    Prec 83.594%
(83.594%)
 * Prec 80.960%
```

```
best acc: 82.940000
Epoch: [40][0/391]      Time 0.519 (0.519)      Data 0.456 (0.456)      Loss
0.4222 (0.4222)    Prec 84.375% (84.375%)
Epoch: [40][100/391]    Time 0.050 (0.055)      Data 0.002 (0.007)      Loss
0.4214 (0.3728)    Prec 88.281% (87.059%)
Epoch: [40][200/391]    Time 0.046 (0.054)      Data 0.002 (0.004)      Loss
0.3763 (0.3797)    Prec 85.938% (86.777%)
Epoch: [40][300/391]    Time 0.037 (0.053)      Data 0.002 (0.004)      Loss
0.3789 (0.3748)    Prec 85.156% (86.981%)
Validation starts
Test: [0/79]    Time 0.430 (0.430)      Loss 0.5500 (0.5500)    Prec 81.250%
(81.250%)
 * Prec 78.880%
best acc: 82.940000
Epoch: [41][0/391]      Time 0.397 (0.397)      Data 0.330 (0.330)      Loss
0.3527 (0.3527)    Prec 88.281% (88.281%)
Epoch: [41][100/391]    Time 0.048 (0.056)      Data 0.002 (0.006)      Loss
0.2638 (0.3639)    Prec 89.844% (87.044%)
Epoch: [41][200/391]    Time 0.044 (0.053)      Data 0.002 (0.004)      Loss
0.3601 (0.3685)    Prec 85.156% (86.777%)
Epoch: [41][300/391]    Time 0.042 (0.051)      Data 0.002 (0.003)      Loss
0.4169 (0.3670)    Prec 87.500% (86.919%)
Validation starts
Test: [0/79]    Time 0.550 (0.550)      Loss 0.4779 (0.4779)    Prec 79.688%
(79.688%)
 * Prec 82.210%
best acc: 82.940000
Epoch: [42][0/391]      Time 0.463 (0.463)      Data 0.390 (0.390)      Loss
0.3684 (0.3684)    Prec 85.156% (85.156%)
Epoch: [42][100/391]    Time 0.052 (0.057)      Data 0.002 (0.006)      Loss
0.4166 (0.3593)    Prec 85.938% (87.252%)
Epoch: [42][200/391]    Time 0.058 (0.056)      Data 0.002 (0.004)      Loss
0.4294 (0.3631)    Prec 82.812% (87.240%)
Epoch: [42][300/391]    Time 0.051 (0.055)      Data 0.002 (0.003)      Loss
0.3573 (0.3658)    Prec 86.719% (87.202%)
Validation starts
Test: [0/79]    Time 0.564 (0.564)      Loss 0.4013 (0.4013)    Prec 89.062%
(89.062%)
 * Prec 83.530%
best acc: 83.530000
Epoch: [43][0/391]      Time 0.668 (0.668)      Data 0.602 (0.602)      Loss
0.3096 (0.3096)    Prec 89.844% (89.844%)
Epoch: [43][100/391]    Time 0.060 (0.061)      Data 0.003 (0.008)      Loss
0.2530 (0.3648)    Prec 90.625% (87.283%)
Epoch: [43][200/391]    Time 0.050 (0.055)      Data 0.002 (0.005)      Loss
0.4724 (0.3667)    Prec 82.812% (87.174%)
Epoch: [43][300/391]    Time 0.046 (0.053)      Data 0.002 (0.004)      Loss
0.3570 (0.3706)    Prec 86.719% (86.981%)
```

```
Validation starts
Test: [0/79]    Time 0.335 (0.335)      Loss 0.3239 (0.3239)    Prec 89.062%
(89.062%)
 * Prec 83.220%
best acc: 83.530000
Epoch: [44][0/391]      Time 0.591 (0.591)      Data 0.526 (0.526)      Loss
0.3521 (0.3521)    Prec 87.500% (87.500%)
Epoch: [44][100/391]    Time 0.039 (0.050)      Data 0.002 (0.007)      Loss
0.2638 (0.3456)    Prec 89.062% (87.894%)
Epoch: [44][200/391]    Time 0.054 (0.052)      Data 0.002 (0.005)      Loss
0.4488 (0.3578)    Prec 86.719% (87.469%)
Epoch: [44][300/391]    Time 0.052 (0.053)      Data 0.002 (0.004)      Loss
0.2872 (0.3618)    Prec 90.625% (87.456%)
Validation starts
Test: [0/79]    Time 0.268 (0.268)      Loss 0.4336 (0.4336)    Prec 83.594%
(83.594%)
 * Prec 79.990%
best acc: 83.530000
Epoch: [45][0/391]      Time 0.771 (0.771)      Data 0.709 (0.709)      Loss
0.3331 (0.3331)    Prec 87.500% (87.500%)
Epoch: [45][100/391]    Time 0.047 (0.057)      Data 0.002 (0.009)      Loss
0.2855 (0.3507)    Prec 88.281% (87.709%)
Epoch: [45][200/391]    Time 0.039 (0.052)      Data 0.002 (0.006)      Loss
0.4039 (0.3576)    Prec 85.156% (87.399%)
Epoch: [45][300/391]    Time 0.055 (0.050)      Data 0.002 (0.004)      Loss
0.5707 (0.3687)    Prec 85.938% (86.976%)
Validation starts
Test: [0/79]    Time 0.507 (0.507)      Loss 0.4075 (0.4075)    Prec 85.156%
(85.156%)
 * Prec 82.230%
best acc: 83.530000
Epoch: [46][0/391]      Time 0.462 (0.462)      Data 0.401 (0.401)      Loss
0.4497 (0.4497)    Prec 85.156% (85.156%)
Epoch: [46][100/391]    Time 0.052 (0.054)      Data 0.002 (0.006)      Loss
0.4244 (0.3693)    Prec 84.375% (86.897%)
Epoch: [46][200/391]    Time 0.051 (0.053)      Data 0.002 (0.004)      Loss
0.2887 (0.3634)    Prec 90.625% (87.177%)
Epoch: [46][300/391]    Time 0.050 (0.052)      Data 0.002 (0.004)      Loss
0.4226 (0.3679)    Prec 85.156% (87.087%)
Validation starts
Test: [0/79]    Time 0.500 (0.500)      Loss 0.3780 (0.3780)    Prec 85.156%
(85.156%)
 * Prec 81.600%
best acc: 83.530000
Epoch: [47][0/391]      Time 0.515 (0.515)      Data 0.455 (0.455)      Loss
0.2599 (0.2599)    Prec 89.062% (89.062%)
Epoch: [47][100/391]    Time 0.056 (0.057)      Data 0.002 (0.007)      Loss
0.2829 (0.3643)    Prec 92.969% (87.260%)
```

```
Epoch: [47][200/391]    Time 0.042 (0.055)      Data 0.002 (0.004)      Loss
0.4781 (0.3607)    Prec 82.812% (87.310%)
Epoch: [47][300/391]    Time 0.053 (0.054)      Data 0.002 (0.004)      Loss
0.3456 (0.3665)    Prec 85.938% (87.017%)
Validation starts
Test: [0/79]    Time 0.503 (0.503)      Loss 0.4572 (0.4572)    Prec 83.594%
(83.594%)
 * Prec 81.430%
best acc: 83.530000
Epoch: [48][0/391]      Time 0.415 (0.415)      Data 0.354 (0.354)      Loss
0.3865 (0.3865)    Prec 89.844% (89.844%)
Epoch: [48][100/391]    Time 0.053 (0.050)      Data 0.003 (0.006)      Loss
0.3626 (0.3423)    Prec 88.281% (88.065%)
Epoch: [48][200/391]    Time 0.045 (0.050)      Data 0.002 (0.004)      Loss
0.4742 (0.3499)    Prec 82.031% (87.671%)
Epoch: [48][300/391]    Time 0.056 (0.050)      Data 0.002 (0.003)      Loss
0.3135 (0.3536)    Prec 89.062% (87.557%)
Validation starts
Test: [0/79]    Time 0.537 (0.537)      Loss 0.4952 (0.4952)    Prec 86.719%
(86.719%)
 * Prec 82.410%
best acc: 83.530000
Epoch: [49][0/391]      Time 0.615 (0.615)      Data 0.544 (0.544)      Loss
0.3620 (0.3620)    Prec 88.281% (88.281%)
Epoch: [49][100/391]    Time 0.053 (0.057)      Data 0.003 (0.008)      Loss
0.3872 (0.3390)    Prec 86.719% (88.243%)
Epoch: [49][200/391]    Time 0.060 (0.053)      Data 0.003 (0.005)      Loss
0.3393 (0.3486)    Prec 88.281% (87.893%)
Epoch: [49][300/391]    Time 0.059 (0.053)      Data 0.002 (0.004)      Loss
0.3139 (0.3513)    Prec 89.844% (87.752%)
Validation starts
Test: [0/79]    Time 0.539 (0.539)      Loss 0.4755 (0.4755)    Prec 82.812%
(82.812%)
 * Prec 83.260%
best acc: 83.530000
Epoch: [50][0/391]      Time 0.452 (0.452)      Data 0.389 (0.389)      Loss
0.2974 (0.2974)    Prec 89.844% (89.844%)
Epoch: [50][100/391]    Time 0.052 (0.055)      Data 0.002 (0.006)      Loss
0.2493 (0.3483)    Prec 93.750% (87.871%)
Epoch: [50][200/391]    Time 0.053 (0.054)      Data 0.002 (0.004)      Loss
0.2464 (0.3550)    Prec 94.531% (87.508%)
Epoch: [50][300/391]    Time 0.050 (0.052)      Data 0.002 (0.004)      Loss
0.2101 (0.3530)    Prec 93.750% (87.604%)
Validation starts
Test: [0/79]    Time 0.350 (0.350)      Loss 0.3712 (0.3712)    Prec 89.844%
(89.844%)
 * Prec 83.340%
best acc: 83.530000
```

```
Epoch: [51][0/391]      Time 0.740 (0.740)      Data 0.670 (0.670)      Loss
0.2108 (0.2108)    Prec 94.531% (94.531%)
Epoch: [51][100/391]    Time 0.043 (0.054)      Data 0.002 (0.009)      Loss
0.4852 (0.3316)    Prec 83.594% (88.575%)
Epoch: [51][200/391]    Time 0.040 (0.053)      Data 0.002 (0.006)      Loss
0.3428 (0.3318)    Prec 87.500% (88.390%)
Epoch: [51][300/391]    Time 0.051 (0.052)      Data 0.002 (0.005)      Loss
0.3765 (0.3409)    Prec 85.156% (88.094%)
Validation starts
Test: [0/79]    Time 0.403 (0.403)      Loss 0.3934 (0.3934)    Prec 85.938%
(85.938%)
 * Prec 82.010%
best acc: 83.530000
Epoch: [52][0/391]      Time 0.465 (0.465)      Data 0.419 (0.419)      Loss
0.3077 (0.3077)    Prec 87.500% (87.500%)
Epoch: [52][100/391]    Time 0.059 (0.055)      Data 0.002 (0.006)      Loss
0.2912 (0.3364)    Prec 88.281% (88.312%)
Epoch: [52][200/391]    Time 0.053 (0.049)      Data 0.002 (0.004)      Loss
0.4036 (0.3314)    Prec 83.594% (88.343%)
Epoch: [52][300/391]    Time 0.042 (0.049)      Data 0.002 (0.003)      Loss
0.3998 (0.3391)    Prec 89.062% (88.100%)
Validation starts
Test: [0/79]    Time 0.324 (0.324)      Loss 0.4155 (0.4155)    Prec 85.938%
(85.938%)
 * Prec 84.470%
best acc: 84.470000
Epoch: [53][0/391]      Time 0.554 (0.554)      Data 0.512 (0.512)      Loss
0.3180 (0.3180)    Prec 88.281% (88.281%)
Epoch: [53][100/391]    Time 0.050 (0.049)      Data 0.002 (0.007)      Loss
0.3813 (0.3261)    Prec 83.594% (88.374%)
Epoch: [53][200/391]    Time 0.052 (0.049)      Data 0.002 (0.004)      Loss
0.3150 (0.3289)    Prec 89.062% (88.452%)
Epoch: [53][300/391]    Time 0.039 (0.048)      Data 0.002 (0.003)      Loss
0.3482 (0.3287)    Prec 84.375% (88.419%)
Validation starts
Test: [0/79]    Time 0.301 (0.301)      Loss 0.5554 (0.5554)    Prec 80.469%
(80.469%)
 * Prec 81.480%
best acc: 84.470000
Epoch: [54][0/391]      Time 0.491 (0.491)      Data 0.429 (0.429)      Loss
0.2271 (0.2271)    Prec 92.969% (92.969%)
Epoch: [54][100/391]    Time 0.042 (0.051)      Data 0.002 (0.006)      Loss
0.3058 (0.3154)    Prec 90.625% (88.714%)
Epoch: [54][200/391]    Time 0.042 (0.046)      Data 0.002 (0.004)      Loss
0.3690 (0.3223)    Prec 88.281% (88.616%)
Epoch: [54][300/391]    Time 0.045 (0.045)      Data 0.002 (0.003)      Loss
0.4278 (0.3291)    Prec 82.812% (88.473%)
Validation starts
```

```
Test: [0/79]    Time 0.407 (0.407)      Loss 0.4497 (0.4497)    Prec 84.375%
(84.375%)
 * Prec 83.790%
best acc: 84.470000
Epoch: [55][0/391]      Time 0.638 (0.638)      Data 0.580 (0.580)      Loss
0.2757 (0.2757)    Prec 90.625% (90.625%)
Epoch: [55][100/391]    Time 0.050 (0.056)      Data 0.002 (0.008)      Loss
0.3724 (0.3234)    Prec 86.719% (88.583%)
Epoch: [55][200/391]    Time 0.066 (0.054)      Data 0.004 (0.005)      Loss
0.2573 (0.3248)    Prec 92.969% (88.647%)
Epoch: [55][300/391]    Time 0.058 (0.054)      Data 0.003 (0.004)      Loss
0.3158 (0.3307)    Prec 89.062% (88.502%)
Validation starts
Test: [0/79]    Time 0.470 (0.470)      Loss 0.4672 (0.4672)    Prec 84.375%
(84.375%)
 * Prec 82.690%
best acc: 84.470000
Epoch: [56][0/391]      Time 0.513 (0.513)      Data 0.444 (0.444)      Loss
0.3074 (0.3074)    Prec 87.500% (87.500%)
Epoch: [56][100/391]    Time 0.055 (0.058)      Data 0.002 (0.007)      Loss
0.3448 (0.3015)    Prec 85.938% (89.581%)
Epoch: [56][200/391]    Time 0.059 (0.054)      Data 0.003 (0.005)      Loss
0.2845 (0.3115)    Prec 89.844% (89.202%)
Epoch: [56][300/391]    Time 0.051 (0.053)      Data 0.002 (0.004)      Loss
0.3666 (0.3249)    Prec 86.719% (88.671%)
Validation starts
Test: [0/79]    Time 0.666 (0.666)      Loss 0.5415 (0.5415)    Prec 81.250%
(81.250%)
 * Prec 82.510%
best acc: 84.470000
Epoch: [57][0/391]      Time 0.576 (0.576)      Data 0.514 (0.514)      Loss
0.3623 (0.3623)    Prec 84.375% (84.375%)
Epoch: [57][100/391]    Time 0.054 (0.055)      Data 0.002 (0.007)      Loss
0.2630 (0.3206)    Prec 92.969% (88.939%)
Epoch: [57][200/391]    Time 0.060 (0.054)      Data 0.002 (0.004)      Loss
0.3749 (0.3201)    Prec 84.375% (88.872%)
Epoch: [57][300/391]    Time 0.050 (0.053)      Data 0.001 (0.004)      Loss
0.4654 (0.3260)    Prec 84.375% (88.632%)
Validation starts
Test: [0/79]    Time 0.332 (0.332)      Loss 0.3631 (0.3631)    Prec 86.719%
(86.719%)
 * Prec 83.800%
best acc: 84.470000
Epoch: [58][0/391]      Time 0.800 (0.800)      Data 0.734 (0.734)      Loss
0.3179 (0.3179)    Prec 88.281% (88.281%)
Epoch: [58][100/391]    Time 0.050 (0.052)      Data 0.002 (0.009)      Loss
0.4765 (0.3093)    Prec 82.812% (89.248%)
Epoch: [58][200/391]    Time 0.047 (0.048)      Data 0.002 (0.006)      Loss
```

```
0.3097 (0.3222)    Prec 87.500% (88.748%)
Epoch: [58][300/391]    Time 0.042 (0.047)      Data 0.002 (0.004)      Loss
0.3345 (0.3249)    Prec 85.156% (88.603%)
Validation starts
Test: [0/79]    Time 0.436 (0.436)      Loss 0.4466 (0.4466)    Prec 85.938%
(85.938%)
 * Prec 84.190%
best acc: 84.470000
Epoch: [59][0/391]      Time 0.433 (0.433)      Data 0.373 (0.373)      Loss
0.3775 (0.3775)    Prec 87.500% (87.500%)
Epoch: [59][100/391]    Time 0.055 (0.052)      Data 0.002 (0.006)      Loss
0.3256 (0.3127)    Prec 87.500% (89.101%)
Epoch: [59][200/391]    Time 0.049 (0.050)      Data 0.002 (0.004)      Loss
0.3274 (0.3157)    Prec 88.281% (88.856%)
Epoch: [59][300/391]    Time 0.050 (0.050)      Data 0.002 (0.003)      Loss
0.3032 (0.3234)    Prec 89.062% (88.567%)
Validation starts
Test: [0/79]    Time 0.396 (0.396)      Loss 0.4677 (0.4677)    Prec 85.938%
(85.938%)
 * Prec 83.380%
best acc: 84.470000
Epoch: [60][0/391]      Time 0.456 (0.456)      Data 0.405 (0.405)      Loss
0.2426 (0.2426)    Prec 91.406% (91.406%)
Epoch: [60][100/391]    Time 0.047 (0.049)      Data 0.002 (0.006)      Loss
0.4202 (0.2973)    Prec 84.375% (89.511%)
Epoch: [60][200/391]    Time 0.057 (0.051)      Data 0.002 (0.004)      Loss
0.2923 (0.3104)    Prec 89.062% (89.074%)
Epoch: [60][300/391]    Time 0.044 (0.049)      Data 0.002 (0.003)      Loss
0.2469 (0.3200)    Prec 90.625% (88.795%)
Validation starts
Test: [0/79]    Time 0.268 (0.268)      Loss 0.4548 (0.4548)    Prec 85.156%
(85.156%)
 * Prec 83.240%
best acc: 84.470000
Epoch: [61][0/391]      Time 0.493 (0.493)      Data 0.438 (0.438)      Loss
0.2896 (0.2896)    Prec 90.625% (90.625%)
Epoch: [61][100/391]    Time 0.045 (0.048)      Data 0.002 (0.006)      Loss
0.3485 (0.3213)    Prec 88.281% (88.475%)
Epoch: [61][200/391]    Time 0.041 (0.045)      Data 0.001 (0.004)      Loss
0.4378 (0.3146)    Prec 87.500% (88.783%)
Epoch: [61][300/391]    Time 0.047 (0.044)      Data 0.002 (0.003)      Loss
0.4114 (0.3180)    Prec 82.031% (88.684%)
Validation starts
Test: [0/79]    Time 0.345 (0.345)      Loss 0.4112 (0.4112)    Prec 85.156%
(85.156%)
 * Prec 82.720%
best acc: 84.470000
Epoch: [62][0/391]      Time 0.516 (0.516)      Data 0.451 (0.451)      Loss
```

```
0.3013 (0.3013)    Prec 88.281% (88.281%)
Epoch: [62][100/391]    Time 0.052 (0.057)      Data 0.002 (0.006)      Loss
0.2037 (0.3014)    Prec 93.750% (89.387%)
Epoch: [62][200/391]    Time 0.060 (0.057)      Data 0.003 (0.004)      Loss
0.4131 (0.3083)    Prec 82.812% (89.121%)
Epoch: [62][300/391]    Time 0.060 (0.056)      Data 0.002 (0.003)      Loss
0.2976 (0.3151)    Prec 88.281% (88.943%)
Validation starts
Test: [0/79]    Time 0.395 (0.395)      Loss 0.3594 (0.3594)    Prec 86.719%
(86.719%)
 * Prec 82.960%
best acc: 84.470000
Epoch: [63][0/391]      Time 0.542 (0.542)      Data 0.476 (0.476)      Loss
0.3209 (0.3209)    Prec 89.844% (89.844%)
Epoch: [63][100/391]    Time 0.051 (0.054)      Data 0.002 (0.007)      Loss
0.2915 (0.3171)    Prec 87.500% (88.962%)
Epoch: [63][200/391]    Time 0.041 (0.052)      Data 0.002 (0.005)      Loss
0.2207 (0.3160)    Prec 91.406% (89.051%)
Epoch: [63][300/391]    Time 0.040 (0.049)      Data 0.002 (0.004)      Loss
0.3609 (0.3137)    Prec 87.500% (89.094%)
Validation starts
Test: [0/79]    Time 0.352 (0.352)      Loss 0.3513 (0.3513)    Prec 88.281%
(88.281%)
 * Prec 83.660%
best acc: 84.470000
Epoch: [64][0/391]      Time 0.492 (0.492)      Data 0.425 (0.425)      Loss
0.2535 (0.2535)    Prec 92.188% (92.188%)
Epoch: [64][100/391]    Time 0.050 (0.056)      Data 0.002 (0.006)      Loss
0.3169 (0.3067)    Prec 89.844% (89.233%)
Epoch: [64][200/391]    Time 0.052 (0.054)      Data 0.003 (0.004)      Loss
0.3535 (0.3014)    Prec 86.719% (89.346%)
Epoch: [64][300/391]    Time 0.040 (0.052)      Data 0.002 (0.004)      Loss
0.2387 (0.3049)    Prec 92.188% (89.200%)
Validation starts
Test: [0/79]    Time 0.328 (0.328)      Loss 0.3871 (0.3871)    Prec 88.281%
(88.281%)
 * Prec 82.730%
best acc: 84.470000
Epoch: [65][0/391]      Time 1.107 (1.107)      Data 1.056 (1.056)      Loss
0.4033 (0.4033)    Prec 87.500% (87.500%)
Epoch: [65][100/391]    Time 0.046 (0.063)      Data 0.003 (0.013)      Loss
0.2823 (0.2986)    Prec 90.625% (89.349%)
Epoch: [65][200/391]    Time 0.039 (0.057)      Data 0.002 (0.008)      Loss
0.3514 (0.3088)    Prec 85.938% (88.958%)
Epoch: [65][300/391]    Time 0.051 (0.053)      Data 0.003 (0.006)      Loss
0.2863 (0.3067)    Prec 89.062% (89.135%)
Validation starts
Test: [0/79]    Time 0.308 (0.308)      Loss 0.4870 (0.4870)    Prec 85.938%
```

```
(85.938%)
 * Prec 81.140%
best acc: 84.470000
Epoch: [66][0/391]      Time 0.544 (0.544)      Data 0.481 (0.481)      Loss
0.3355 (0.3355)    Prec 88.281% (88.281%)
Epoch: [66][100/391]    Time 0.057 (0.051)      Data 0.002 (0.007)      Loss
0.3004 (0.3005)    Prec 91.406% (89.240%)
Epoch: [66][200/391]    Time 0.051 (0.050)      Data 0.002 (0.005)      Loss
0.2412 (0.3069)    Prec 90.625% (89.035%)
Epoch: [66][300/391]    Time 0.051 (0.050)      Data 0.002 (0.004)      Loss
0.3339 (0.3088)    Prec 89.844% (89.016%)
Validation starts
Test: [0/79]    Time 0.404 (0.404)      Loss 0.4927 (0.4927)    Prec 82.812%
(82.812%)
 * Prec 81.880%
best acc: 84.470000
Epoch: [67][0/391]      Time 0.483 (0.483)      Data 0.435 (0.435)      Loss
0.3462 (0.3462)    Prec 86.719% (86.719%)
Epoch: [67][100/391]    Time 0.042 (0.057)      Data 0.002 (0.007)      Loss
0.3334 (0.3144)    Prec 84.375% (88.637%)
Epoch: [67][200/391]    Time 0.045 (0.051)      Data 0.002 (0.004)      Loss
0.2308 (0.3144)    Prec 90.625% (88.779%)
Epoch: [67][300/391]    Time 0.049 (0.050)      Data 0.002 (0.003)      Loss
0.3574 (0.3164)    Prec 88.281% (88.808%)
Validation starts
Test: [0/79]    Time 0.356 (0.356)      Loss 0.3967 (0.3967)    Prec 87.500%
(87.500%)
 * Prec 82.420%
best acc: 84.470000
Epoch: [68][0/391]      Time 0.589 (0.589)      Data 0.527 (0.527)      Loss
0.2847 (0.2847)    Prec 88.281% (88.281%)
Epoch: [68][100/391]    Time 0.038 (0.048)      Data 0.002 (0.007)      Loss
0.2286 (0.2930)    Prec 88.281% (89.705%)
Epoch: [68][200/391]    Time 0.036 (0.047)      Data 0.002 (0.005)      Loss
0.2441 (0.2954)    Prec 89.062% (89.440%)
Epoch: [68][300/391]    Time 0.029 (0.044)      Data 0.001 (0.004)      Loss
0.3211 (0.3017)    Prec 89.844% (89.278%)
Validation starts
Test: [0/79]    Time 0.352 (0.352)      Loss 0.3907 (0.3907)    Prec 87.500%
(87.500%)
 * Prec 83.690%
best acc: 84.470000
Epoch: [69][0/391]      Time 0.420 (0.420)      Data 0.359 (0.359)      Loss
0.4164 (0.4164)    Prec 85.938% (85.938%)
Epoch: [69][100/391]    Time 0.038 (0.045)      Data 0.002 (0.005)      Loss
0.2001 (0.2909)    Prec 94.531% (89.712%)
Epoch: [69][200/391]    Time 0.039 (0.044)      Data 0.002 (0.004)      Loss
0.2286 (0.2854)    Prec 94.531% (89.883%)
```

```
Epoch: [69][300/391]    Time 0.040 (0.044)      Data 0.002 (0.003)      Loss
0.3041 (0.2924)    Prec 89.062% (89.727%)
Validation starts
Test: [0/79]    Time 0.436 (0.436)      Loss 0.3961 (0.3961)    Prec 84.375%
(84.375%)
 * Prec 83.180%
best acc: 84.470000
Epoch: [70][0/391]      Time 0.490 (0.490)      Data 0.425 (0.425)      Loss
0.2011 (0.2011)    Prec 92.188% (92.188%)
Epoch: [70][100/391]    Time 0.036 (0.053)      Data 0.002 (0.006)      Loss
0.1857 (0.2972)    Prec 94.531% (89.565%)
Epoch: [70][200/391]    Time 0.041 (0.049)      Data 0.002 (0.004)      Loss
0.3160 (0.2908)    Prec 89.844% (89.673%)
Epoch: [70][300/391]    Time 0.042 (0.048)      Data 0.001 (0.003)      Loss
0.4878 (0.2996)    Prec 82.031% (89.392%)
Validation starts
Test: [0/79]    Time 0.451 (0.451)      Loss 0.4183 (0.4183)    Prec 86.719%
(86.719%)
 * Prec 81.810%
best acc: 84.470000
Epoch: [71][0/391]      Time 0.438 (0.438)      Data 0.376 (0.376)      Loss
0.2099 (0.2099)    Prec 92.188% (92.188%)
Epoch: [71][100/391]    Time 0.045 (0.046)      Data 0.002 (0.006)      Loss
0.2397 (0.2805)    Prec 91.406% (89.998%)
Epoch: [71][200/391]    Time 0.045 (0.045)      Data 0.002 (0.004)      Loss
0.1834 (0.2898)    Prec 94.531% (89.700%)
Epoch: [71][300/391]    Time 0.050 (0.044)      Data 0.002 (0.003)      Loss
0.2231 (0.2917)    Prec 90.625% (89.794%)
Validation starts
Test: [0/79]    Time 0.339 (0.339)      Loss 0.3026 (0.3026)    Prec 88.281%
(88.281%)
 * Prec 84.030%
best acc: 84.470000
Epoch: [72][0/391]      Time 0.629 (0.629)      Data 0.568 (0.568)      Loss
0.3344 (0.3344)    Prec 86.719% (86.719%)
Epoch: [72][100/391]    Time 0.044 (0.050)      Data 0.002 (0.008)      Loss
0.1642 (0.2793)    Prec 94.531% (90.153%)
Epoch: [72][200/391]    Time 0.048 (0.048)      Data 0.002 (0.005)      Loss
0.3628 (0.2942)    Prec 91.406% (89.548%)
Epoch: [72][300/391]    Time 0.050 (0.049)      Data 0.002 (0.004)      Loss
0.2713 (0.3024)    Prec 90.625% (89.273%)
Validation starts
Test: [0/79]    Time 0.345 (0.345)      Loss 0.3914 (0.3914)    Prec 86.719%
(86.719%)
 * Prec 83.870%
best acc: 84.470000
Epoch: [73][0/391]      Time 0.461 (0.461)      Data 0.401 (0.401)      Loss
0.2364 (0.2364)    Prec 93.750% (93.750%)
```

```
Epoch: [73][100/391]    Time 0.046 (0.050)      Data 0.002 (0.006)      Loss
0.2978 (0.2884)    Prec 90.625% (89.882%)
Epoch: [73][200/391]    Time 0.041 (0.048)      Data 0.002 (0.004)      Loss
0.3413 (0.2949)    Prec 89.844% (89.649%)
Epoch: [73][300/391]    Time 0.053 (0.047)      Data 0.002 (0.003)      Loss
0.3919 (0.3068)    Prec 86.719% (89.327%)
Validation starts
Test: [0/79]    Time 0.468 (0.468)      Loss 0.4133 (0.4133)    Prec 86.719%
(86.719%)
 * Prec 84.430%
best acc: 84.470000
Epoch: [74][0/391]      Time 0.469 (0.469)      Data 0.426 (0.426)      Loss
0.2598 (0.2598)    Prec 90.625% (90.625%)
Epoch: [74][100/391]    Time 0.047 (0.047)      Data 0.002 (0.006)      Loss
0.3057 (0.2912)    Prec 89.062% (89.674%)
Epoch: [74][200/391]    Time 0.044 (0.048)      Data 0.001 (0.004)      Loss
0.2056 (0.2946)    Prec 92.188% (89.618%)
Epoch: [74][300/391]    Time 0.060 (0.048)      Data 0.003 (0.004)      Loss
0.1921 (0.2932)    Prec 92.188% (89.589%)
Validation starts
Test: [0/79]    Time 0.399 (0.399)      Loss 0.4394 (0.4394)    Prec 83.594%
(83.594%)
 * Prec 84.360%
best acc: 84.470000
Epoch: [75][0/391]      Time 0.675 (0.675)      Data 0.577 (0.577)      Loss
0.1876 (0.1876)    Prec 92.969% (92.969%)
Epoch: [75][100/391]    Time 0.036 (0.051)      Data 0.002 (0.008)      Loss
0.2268 (0.2831)    Prec 92.188% (89.991%)
Epoch: [75][200/391]    Time 0.052 (0.048)      Data 0.002 (0.005)      Loss
0.2645 (0.2948)    Prec 88.281% (89.572%)
Epoch: [75][300/391]    Time 0.049 (0.047)      Data 0.002 (0.004)      Loss
0.2842 (0.2981)    Prec 86.719% (89.418%)
Validation starts
Test: [0/79]    Time 0.288 (0.288)      Loss 0.3897 (0.3897)    Prec 86.719%
(86.719%)
 * Prec 81.230%
best acc: 84.470000
Epoch: [76][0/391]      Time 0.815 (0.815)      Data 0.756 (0.756)      Loss
0.4097 (0.4097)    Prec 85.938% (85.938%)
Epoch: [76][100/391]    Time 0.040 (0.051)      Data 0.002 (0.009)      Loss
0.3379 (0.2832)    Prec 88.281% (90.076%)
Epoch: [76][200/391]    Time 0.047 (0.048)      Data 0.002 (0.006)      Loss
0.2864 (0.2908)    Prec 91.406% (89.762%)
Epoch: [76][300/391]    Time 0.048 (0.048)      Data 0.001 (0.004)      Loss
0.3412 (0.2943)    Prec 89.062% (89.714%)
Validation starts
Test: [0/79]    Time 0.349 (0.349)      Loss 0.4253 (0.4253)    Prec 88.281%
(88.281%)
```

```
 * Prec 84.250%
best acc: 84.470000
Epoch: [77][0/391]      Time 0.538 (0.538)      Data 0.476 (0.476)      Loss
0.2067 (0.2067)    Prec 90.625% (90.625%)
Epoch: [77][100/391]    Time 0.044 (0.048)      Data 0.002 (0.007)      Loss
0.3172 (0.2900)    Prec 86.719% (90.022%)
Epoch: [77][200/391]    Time 0.043 (0.047)      Data 0.002 (0.004)      Loss
0.3211 (0.2904)    Prec 88.281% (89.914%)
Epoch: [77][300/391]    Time 0.046 (0.045)      Data 0.002 (0.004)      Loss
0.2808 (0.2958)    Prec 91.406% (89.696%)
Validation starts
Test: [0/79]    Time 0.320 (0.320)      Loss 0.3832 (0.3832)    Prec 85.938%
(85.938%)
 * Prec 84.580%
best acc: 84.580000
Epoch: [78][0/391]      Time 0.420 (0.420)      Data 0.356 (0.356)      Loss
0.2135 (0.2135)    Prec 92.188% (92.188%)
Epoch: [78][100/391]    Time 0.047 (0.048)      Data 0.002 (0.005)      Loss
0.3452 (0.2638)    Prec 87.500% (90.764%)
Epoch: [78][200/391]    Time 0.038 (0.045)      Data 0.001 (0.004)      Loss
0.3107 (0.2779)    Prec 87.500% (90.333%)
Epoch: [78][300/391]    Time 0.042 (0.045)      Data 0.002 (0.003)      Loss
0.3062 (0.2863)    Prec 89.844% (90.005%)
Validation starts
Test: [0/79]    Time 0.286 (0.286)      Loss 0.4770 (0.4770)    Prec 84.375%
(84.375%)
 * Prec 83.230%
best acc: 84.580000
Epoch: [79][0/391]      Time 0.433 (0.433)      Data 0.372 (0.372)      Loss
0.3460 (0.3460)    Prec 85.156% (85.156%)
Epoch: [79][100/391]    Time 0.048 (0.045)      Data 0.002 (0.006)      Loss
0.2328 (0.2860)    Prec 92.969% (89.913%)
Epoch: [79][200/391]    Time 0.038 (0.045)      Data 0.002 (0.004)      Loss
0.2968 (0.2876)    Prec 91.406% (89.778%)
Epoch: [79][300/391]    Time 0.048 (0.043)      Data 0.002 (0.003)      Loss
0.2550 (0.2919)    Prec 92.969% (89.610%)
Validation starts
Test: [0/79]    Time 0.301 (0.301)      Loss 0.4312 (0.4312)    Prec 87.500%
(87.500%)
 * Prec 85.250%
best acc: 85.250000
Epoch: [80][0/391]      Time 1.052 (1.052)      Data 0.989 (0.989)      Loss
0.2563 (0.2563)    Prec 89.844% (89.844%)
Epoch: [80][100/391]    Time 0.048 (0.057)      Data 0.002 (0.012)      Loss
0.2235 (0.2384)    Prec 93.750% (91.754%)
Epoch: [80][200/391]    Time 0.038 (0.051)      Data 0.001 (0.007)      Loss
0.2224 (0.2270)    Prec 92.969% (92.086%)
Epoch: [80][300/391]    Time 0.044 (0.048)      Data 0.002 (0.005)      Loss
```

```
0.2408 (0.2178)    Prec 91.406% (92.424%)
Validation starts
Test: [0/79]    Time 0.469 (0.469)      Loss 0.2459 (0.2459)    Prec 92.188%
(92.188%)
 * Prec 87.340%
best acc: 87.340000
Epoch: [81][0/391]      Time 0.472 (0.472)      Data 0.407 (0.407)      Loss
0.1664 (0.1664)    Prec 95.312% (95.312%)
Epoch: [81][100/391]    Time 0.044 (0.049)      Data 0.001 (0.006)      Loss
0.1495 (0.2012)    Prec 96.094% (93.147%)
Epoch: [81][200/391]    Time 0.041 (0.046)      Data 0.001 (0.004)      Loss
0.1373 (0.1888)    Prec 93.750% (93.493%)
Epoch: [81][300/391]    Time 0.044 (0.044)      Data 0.002 (0.003)      Loss
0.1167 (0.1886)    Prec 96.094% (93.579%)
Validation starts
Test: [0/79]    Time 0.549 (0.549)      Loss 0.2333 (0.2333)    Prec 89.062%
(89.062%)
 * Prec 88.110%
best acc: 88.110000
Epoch: [82][0/391]      Time 0.649 (0.649)      Data 0.588 (0.588)      Loss
0.2046 (0.2046)    Prec 92.969% (92.969%)
Epoch: [82][100/391]    Time 0.040 (0.050)      Data 0.002 (0.008)      Loss
0.2171 (0.1878)    Prec 92.188% (93.541%)
Epoch: [82][200/391]    Time 0.043 (0.046)      Data 0.002 (0.005)      Loss
0.1820 (0.1828)    Prec 93.750% (93.746%)
Epoch: [82][300/391]    Time 0.042 (0.045)      Data 0.002 (0.004)      Loss
0.1165 (0.1809)    Prec 96.875% (93.781%)
Validation starts
Test: [0/79]    Time 0.422 (0.422)      Loss 0.2492 (0.2492)    Prec 91.406%
(91.406%)
 * Prec 88.280%
best acc: 88.280000
Epoch: [83][0/391]      Time 0.558 (0.558)      Data 0.492 (0.492)      Loss
0.1080 (0.1080)    Prec 96.094% (96.094%)
Epoch: [83][100/391]    Time 0.054 (0.053)      Data 0.003 (0.007)      Loss
0.2562 (0.1674)    Prec 90.625% (94.291%)
Epoch: [83][200/391]    Time 0.046 (0.050)      Data 0.002 (0.005)      Loss
0.1940 (0.1700)    Prec 94.531% (94.150%)
Epoch: [83][300/391]    Time 0.039 (0.049)      Data 0.002 (0.004)      Loss
0.1636 (0.1743)    Prec 94.531% (93.989%)
Validation starts
Test: [0/79]    Time 0.372 (0.372)      Loss 0.2442 (0.2442)    Prec 91.406%
(91.406%)
 * Prec 88.020%
best acc: 88.280000
Epoch: [84][0/391]      Time 0.523 (0.523)      Data 0.461 (0.461)      Loss
0.2174 (0.2174)    Prec 93.750% (93.750%)
Epoch: [84][100/391]    Time 0.052 (0.053)      Data 0.002 (0.006)      Loss
```

```
0.1422 (0.1685)    Prec 94.531% (94.338%)
Epoch: [84][200/391]    Time 0.050 (0.053)      Data 0.002 (0.004)      Loss
0.1699 (0.1703)    Prec 92.969% (94.236%)
Epoch: [84][300/391]    Time 0.038 (0.052)      Data 0.002 (0.004)      Loss
0.1069 (0.1695)    Prec 95.312% (94.269%)
Validation starts
Test: [0/79]    Time 0.335 (0.335)      Loss 0.2128 (0.2128)    Prec 91.406%
(91.406%)
 * Prec 88.360%
best acc: 88.360000
Epoch: [85][0/391]      Time 0.480 (0.480)      Data 0.416 (0.416)      Loss
0.1610 (0.1610)    Prec 93.750% (93.750%)
Epoch: [85][100/391]    Time 0.056 (0.054)      Data 0.002 (0.006)      Loss
0.1398 (0.1660)    Prec 96.875% (94.160%)
Epoch: [85][200/391]    Time 0.057 (0.052)      Data 0.002 (0.004)      Loss
0.2612 (0.1652)    Prec 91.406% (94.317%)
Epoch: [85][300/391]    Time 0.040 (0.051)      Data 0.002 (0.003)      Loss
0.1643 (0.1637)    Prec 94.531% (94.383%)
Validation starts
Test: [0/79]    Time 0.394 (0.394)      Loss 0.2461 (0.2461)    Prec 92.188%
(92.188%)
 * Prec 88.020%
best acc: 88.360000
Epoch: [86][0/391]      Time 0.643 (0.643)      Data 0.582 (0.582)      Loss
0.1409 (0.1409)    Prec 94.531% (94.531%)
Epoch: [86][100/391]    Time 0.040 (0.051)      Data 0.002 (0.008)      Loss
0.1783 (0.1596)    Prec 95.312% (94.392%)
Epoch: [86][200/391]    Time 0.044 (0.047)      Data 0.002 (0.005)      Loss
0.2012 (0.1636)    Prec 94.531% (94.426%)
Epoch: [86][300/391]    Time 0.050 (0.046)      Data 0.002 (0.004)      Loss
0.2081 (0.1633)    Prec 92.969% (94.407%)
Validation starts
Test: [0/79]    Time 0.252 (0.252)      Loss 0.2445 (0.2445)    Prec 91.406%
(91.406%)
 * Prec 88.340%
best acc: 88.360000
Epoch: [87][0/391]      Time 0.521 (0.521)      Data 0.460 (0.460)      Loss
0.1399 (0.1399)    Prec 95.312% (95.312%)
Epoch: [87][100/391]    Time 0.044 (0.046)      Data 0.002 (0.006)      Loss
0.1843 (0.1537)    Prec 93.750% (94.732%)
Epoch: [87][200/391]    Time 0.048 (0.044)      Data 0.002 (0.004)      Loss
0.1567 (0.1538)    Prec 92.188% (94.745%)
Epoch: [87][300/391]    Time 0.048 (0.043)      Data 0.002 (0.003)      Loss
0.1948 (0.1517)    Prec 93.750% (94.874%)
Validation starts
Test: [0/79]    Time 0.379 (0.379)      Loss 0.2032 (0.2032)    Prec 92.969%
(92.969%)
 * Prec 88.300%
```

```
best acc: 88.360000
Epoch: [88][0/391]      Time 0.553 (0.553)      Data 0.436 (0.436)      Loss
0.1392 (0.1392)    Prec 94.531% (94.531%)
Epoch: [88][100/391]    Time 0.040 (0.047)      Data 0.002 (0.006)      Loss
0.1765 (0.1510)    Prec 92.188% (94.756%)
Epoch: [88][200/391]    Time 0.045 (0.044)      Data 0.002 (0.004)      Loss
0.1156 (0.1497)    Prec 96.094% (94.803%)
Epoch: [88][300/391]    Time 0.045 (0.044)      Data 0.002 (0.003)      Loss
0.1962 (0.1511)    Prec 96.094% (94.726%)
Validation starts
Test: [0/79]    Time 0.440 (0.440)      Loss 0.2273 (0.2273)    Prec 92.188%
(92.188%)
 * Prec 88.340%
best acc: 88.360000
Epoch: [89][0/391]      Time 0.460 (0.460)      Data 0.402 (0.402)      Loss
0.0822 (0.0822)    Prec 99.219% (99.219%)
Epoch: [89][100/391]    Time 0.040 (0.049)      Data 0.002 (0.006)      Loss
0.1377 (0.1356)    Prec 96.094% (95.398%)
Epoch: [89][200/391]    Time 0.042 (0.046)      Data 0.002 (0.004)      Loss
0.1402 (0.1481)    Prec 96.875% (95.033%)
Epoch: [89][300/391]    Time 0.039 (0.046)      Data 0.001 (0.003)      Loss
0.1894 (0.1497)    Prec 93.750% (94.954%)
Validation starts
Test: [0/79]    Time 0.478 (0.478)      Loss 0.2507 (0.2507)    Prec 91.406%
(91.406%)
 * Prec 88.400%
best acc: 88.400000
Epoch: [90][0/391]      Time 0.592 (0.592)      Data 0.522 (0.522)      Loss
0.1079 (0.1079)    Prec 96.094% (96.094%)
Epoch: [90][100/391]    Time 0.039 (0.048)      Data 0.002 (0.007)      Loss
0.1324 (0.1522)    Prec 98.438% (94.825%)
Epoch: [90][200/391]    Time 0.044 (0.045)      Data 0.002 (0.004)      Loss
0.1083 (0.1486)    Prec 95.312% (94.904%)
Epoch: [90][300/391]    Time 0.040 (0.044)      Data 0.002 (0.004)      Loss
0.1052 (0.1450)    Prec 97.656% (95.024%)
Validation starts
Test: [0/79]    Time 0.400 (0.400)      Loss 0.2127 (0.2127)    Prec 92.969%
(92.969%)
 * Prec 88.630%
best acc: 88.630000
Epoch: [91][0/391]      Time 0.572 (0.572)      Data 0.504 (0.504)      Loss
0.2293 (0.2293)    Prec 92.969% (92.969%)
Epoch: [91][100/391]    Time 0.047 (0.055)      Data 0.002 (0.007)      Loss
0.1483 (0.1414)    Prec 95.312% (95.305%)
Epoch: [91][200/391]    Time 0.050 (0.053)      Data 0.002 (0.004)      Loss
0.1434 (0.1441)    Prec 93.750% (95.149%)
Epoch: [91][300/391]    Time 0.059 (0.053)      Data 0.002 (0.003)      Loss
0.1435 (0.1424)    Prec 92.969% (95.141%)
```

```
Validation starts
Test: [0/79]    Time 0.340 (0.340)      Loss 0.2109 (0.2109)    Prec 91.406%
(91.406%)
 * Prec 88.580%
best acc: 88.630000
Epoch: [92][0/391]      Time 0.413 (0.413)      Data 0.355 (0.355)      Loss
0.2125 (0.2125)    Prec 90.625% (90.625%)
Epoch: [92][100/391]    Time 0.044 (0.048)      Data 0.002 (0.005)      Loss
0.1934 (0.1429)    Prec 92.188% (94.995%)
Epoch: [92][200/391]    Time 0.047 (0.046)      Data 0.002 (0.004)      Loss
0.0900 (0.1402)    Prec 97.656% (95.215%)
Epoch: [92][300/391]    Time 0.048 (0.045)      Data 0.002 (0.003)      Loss
0.0871 (0.1384)    Prec 98.438% (95.315%)
Validation starts
Test: [0/79]    Time 0.379 (0.379)      Loss 0.2112 (0.2112)    Prec 91.406%
(91.406%)
 * Prec 88.720%
best acc: 88.720000
Epoch: [93][0/391]      Time 0.466 (0.466)      Data 0.406 (0.406)      Loss
0.0967 (0.0967)    Prec 96.875% (96.875%)
Epoch: [93][100/391]    Time 0.049 (0.047)      Data 0.002 (0.006)      Loss
0.1780 (0.1392)    Prec 95.312% (95.382%)
Epoch: [93][200/391]    Time 0.039 (0.045)      Data 0.002 (0.004)      Loss
0.1657 (0.1400)    Prec 95.312% (95.258%)
Epoch: [93][300/391]    Time 0.048 (0.044)      Data 0.002 (0.003)      Loss
0.1687 (0.1405)    Prec 94.531% (95.214%)
Validation starts
Test: [0/79]    Time 0.350 (0.350)      Loss 0.2156 (0.2156)    Prec 94.531%
(94.531%)
 * Prec 88.590%
best acc: 88.720000
Epoch: [94][0/391]      Time 0.510 (0.510)      Data 0.448 (0.448)      Loss
0.1175 (0.1175)    Prec 96.094% (96.094%)
Epoch: [94][100/391]    Time 0.043 (0.051)      Data 0.002 (0.007)      Loss
0.1700 (0.1418)    Prec 92.969% (95.297%)
Epoch: [94][200/391]    Time 0.040 (0.048)      Data 0.002 (0.004)      Loss
0.2082 (0.1355)    Prec 90.625% (95.495%)
Epoch: [94][300/391]    Time 0.043 (0.047)      Data 0.002 (0.004)      Loss
0.1367 (0.1377)    Prec 95.312% (95.416%)
Validation starts
Test: [0/79]    Time 0.329 (0.329)      Loss 0.2064 (0.2064)    Prec 91.406%
(91.406%)
 * Prec 88.470%
best acc: 88.720000
Epoch: [95][0/391]      Time 0.507 (0.507)      Data 0.406 (0.406)      Loss
0.2039 (0.2039)    Prec 92.969% (92.969%)
Epoch: [95][100/391]    Time 0.054 (0.053)      Data 0.002 (0.006)      Loss
0.1547 (0.1355)    Prec 94.531% (95.514%)
```

```
Epoch: [95][200/391]    Time 0.050 (0.053)      Data 0.002 (0.004)      Loss
0.1613 (0.1378)    Prec 94.531% (95.398%)
Epoch: [95][300/391]    Time 0.049 (0.052)      Data 0.002 (0.003)      Loss
0.1087 (0.1384)    Prec 96.094% (95.307%)
Validation starts
Test: [0/79]    Time 0.290 (0.290)      Loss 0.2053 (0.2053)    Prec 92.188%
(92.188%)
 * Prec 88.450%
best acc: 88.720000
Epoch: [96][0/391]      Time 0.465 (0.465)      Data 0.418 (0.418)      Loss
0.1093 (0.1093)    Prec 95.312% (95.312%)
Epoch: [96][100/391]    Time 0.037 (0.042)      Data 0.002 (0.006)      Loss
0.1506 (0.1408)    Prec 93.750% (95.127%)
Epoch: [96][200/391]    Time 0.045 (0.040)      Data 0.002 (0.004)      Loss
0.1861 (0.1377)    Prec 92.969% (95.250%)
Epoch: [96][300/391]    Time 0.051 (0.042)      Data 0.003 (0.003)      Loss
0.1482 (0.1376)    Prec 93.750% (95.279%)
Validation starts
Test: [0/79]    Time 0.264 (0.264)      Loss 0.2109 (0.2109)    Prec 92.188%
(92.188%)
 * Prec 88.500%
best acc: 88.720000
Epoch: [97][0/391]      Time 0.474 (0.474)      Data 0.406 (0.406)      Loss
0.0436 (0.0436)    Prec 100.000% (100.000%)
Epoch: [97][100/391]    Time 0.042 (0.051)      Data 0.002 (0.006)      Loss
0.0933 (0.1307)    Prec 97.656% (95.684%)
Epoch: [97][200/391]    Time 0.038 (0.046)      Data 0.002 (0.004)      Loss
0.0951 (0.1344)    Prec 96.875% (95.460%)
Epoch: [97][300/391]    Time 0.048 (0.045)      Data 0.003 (0.003)      Loss
0.2157 (0.1355)    Prec 92.188% (95.403%)
Validation starts
Test: [0/79]    Time 0.286 (0.286)      Loss 0.2300 (0.2300)    Prec 89.844%
(89.844%)
 * Prec 88.570%
best acc: 88.720000
Epoch: [98][0/391]      Time 0.582 (0.582)      Data 0.522 (0.522)      Loss
0.1413 (0.1413)    Prec 95.312% (95.312%)
Epoch: [98][100/391]    Time 0.035 (0.047)      Data 0.002 (0.007)      Loss
0.1291 (0.1364)    Prec 96.875% (95.274%)
Epoch: [98][200/391]    Time 0.037 (0.044)      Data 0.001 (0.004)      Loss
0.1309 (0.1329)    Prec 94.531% (95.375%)
Epoch: [98][300/391]    Time 0.040 (0.044)      Data 0.001 (0.003)      Loss
0.1694 (0.1366)    Prec 93.750% (95.294%)
Validation starts
Test: [0/79]    Time 0.347 (0.347)      Loss 0.1940 (0.1940)    Prec 92.188%
(92.188%)
 * Prec 88.520%
best acc: 88.720000
```

```
Epoch: [99][0/391]      Time 0.525 (0.525)      Data 0.460 (0.460)      Loss
0.1084 (0.1084)    Prec 96.875% (96.875%)
Epoch: [99][100/391]    Time 0.035 (0.049)      Data 0.002 (0.006)      Loss
0.1555 (0.1336)    Prec 96.094% (95.560%)
Epoch: [99][200/391]    Time 0.040 (0.046)      Data 0.001 (0.004)      Loss
0.0759 (0.1351)    Prec 96.875% (95.332%)
Epoch: [99][300/391]    Time 0.041 (0.045)      Data 0.002 (0.003)      Loss
0.1100 (0.1359)    Prec 95.312% (95.281%)
Validation starts
Test: [0/79]    Time 0.425 (0.425)      Loss 0.1837 (0.1837)    Prec 93.750%
(93.750%)
 * Prec 88.580%
best acc: 88.720000
```

[6]:

```python
PATH = "result/Resnet_20_quant_project/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: 8872/10000 (89%)
```

[8]:

```python
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)
```

```
7 -th layer prehooked
9 -th layer prehooked
13 -th layer prehooked
15 -th layer prehooked
21 -th layer prehooked
25 -th layer prehooked
27 -th layer prehooked
34 -th layer prehooked
36 -th layer prehooked
42 -th layer prehooked
46 -th layer prehooked
48 -th layer prehooked
54 -th layer prehooked
56 -th layer prehooked
63 -th layer prehooked
65 -th layer prehooked
71 -th layer prehooked
75 -th layer prehooked
77 -th layer prehooked
83 -th layer prehooked
85 -th layer prehooked
```

[ ]:

```
## Layer 9
## Layer 13
## save_outputs.output[1][0]
## save_outputs.output[2][0]
model.layer1[0].conv2.weight_quant.wgt_alpha
```

[63]:

```
### Residual x0_int calculation
x0_bit = 4
w0_bit = 4
x0 = save_output.outputs[0][0]
x0_alpha = model.layer1[0].conv2.act_alpha
w0_alpha = model.layer1[0].conv2.weight_quant.wgt_alpha
x0_delta = x0_alpha/(2**x0_bit-1)
w0_delta = w_alpha/(2**(w0_bit-1)-1)
act_quant = act_quantization(x0_bit)
x0_q = act_quant(x0,(x0_alpha*w0_alpha))
x0_int = x0_q/(x0_delta*w0_delta)
```

[68]:

```
w_bit = 4
weight_q = model.layer1[0].conv2.weight_q
w_alpha = model.layer1[0].conv2.weight_quant.wgt_alpha
w_delta = w_alpha/(2**(w_bit-1)-1)
weight_int = weight_q/w_delta

x_bit = 4
x = save_output.outputs[1][0]
x_alpha = model.layer1[0].conv2.act_alpha
x_delta = x_alpha/(2**x_bit-1)
act_quant = act_quantization(x_bit)
x_q = act_quant(x,x_alpha)
x_int = x_q/x_delta

conv_int = nn.Conv2d(8,8,kernel_size=3, padding=1, bias=False)
conv_int.weight = torch.nn.parameter.Parameter(weight_int)
output_int = (conv_int(x_int))
psum_recovered = output_int*x_delta*w_delta + x0
relu = nn.ReLU(inplace=True)
psum_after_relu = relu(psum_recovered)
difference = (save_output.outputs[2][0] - psum_after_relu).mean()
print("The difference between psum original and psum recovered = {}".
 ↪format(difference))
```

```
The difference between psum original and psum recovered = 2.9711912929997197e-07
```

[41]:

```
act_int = x_int[0,:,:,:]  # pick only one input out of batch
# a_int.size() = [64, 32, 32]

# conv_int.weight.size() = torch.Size([64, 64, 3, 3])  <- output_ch, input_ch,␣
 ↪ki, kj
w_int = torch.reshape(weight_int, (weight_int.size(0), weight_int.size(1), -1))␣
 ↪ # merge ki, kj index to kij

padding = 1
```

```
stride = 1
array_size = 8
x_size = x_int.size()
nig = range(x_size[2])
njg = range(x_size[3])

kijg = range(w_int.size(2))
ki_dim = int(math.sqrt(w_int.size(2)))  ## Kernel's 1 dim size

icg = range(int(w_int.size(1)))  ## input channel
ocg = range(int(w_int.size(0)))  ## output channel

a_pad = torch.zeros((x_size[1],x_size[2]+2*padding,x_size[3]+2*padding))

a_pad[ :, padding:padding+len(nig), padding:padding+len(njg)] = act_int.cuda()
a_pad = torch.reshape(a_pad, (a_pad.size(0), -1))

ic_tile = range(int(int(act_int.size(0))/array_size))
oc_tile = range(int(int(w_int.size(0))/array_size))

a_tile = torch.
 ↪zeros(len(ic_tile),array_size,len(nig)+padding*2,len(njg)+padding*2).cuda()
a_tile = torch.reshape(a_tile,(a_tile.size(0),a_tile.size(1),-1))

for ict in ic_tile:
    a_tile[ict,:,:] = a_pad[(ict*array_size):((ict+1)*array_size),:]

w_tile = torch.zeros(len(ic_tile),len(oc_tile),array_size,array_size,len(kijg)).
 ↪cuda()

for oct in oc_tile:
    for ict in ic_tile:
        w_tile[ict,oct,:,:,:] = w_int[(oct*array_size):
 ↪((oct+1)*array_size),(ict*array_size):((ict+1)*array_size),:]

p_nijg = range(a_pad.size(1)) ## paded activation's nij group

psum = torch.zeros(len(ic_tile),len(oc_tile),array_size,len(p_nijg),len(kijg)).
 ↪cuda()

for kij in kijg:
    for ict in ic_tile:
        for oct in oc_tile:
            for nij in p_nijg:     # time domain, sequentially given input
                m = nn.Linear(array_size, array_size, bias=False)
                m.weight = torch.nn.Parameter(w_tile[ict,oct,:,:,kij])
                psum[ict,oct,:, nij, kij] = m(a_tile[ict,:,nij]).cuda()
```

[42]:

```
import math

a_pad_ni_dim = int(math.sqrt(a_pad.size(1))) # 32 + 2*pad = 34

o_ni_dim = int((a_pad_ni_dim - (ki_dim- 1) - 1)/stride + 1) #34 - 2 - 1 + 1 = 32
o_nijg = range(o_ni_dim**2)

out = torch.zeros(len(ocg), len(o_nijg)).cuda()


### SFP accumulation ###
for o_nij in o_nijg:
    for kij in kijg:
        for ict in ic_tile:
            for oct in oc_tile:
                out[oct*array_size:(oct+1)*array_size,o_nij] =␣
 ↪out[oct*array_size:(oct+1)*array_size,o_nij] + \
                psum[ict,oct,:, int(o_nij/o_ni_dim)*a_pad_ni_dim +␣
 ↪o_nij%o_ni_dim + int(kij/ki_dim)*a_pad_ni_dim + kij%ki_dim, kij]
                ## 2nd index = (int(o_nij/30)*32 + o_nij%30) + (int(kij/3)*32 +␣
 ↪kij%3)
```

[81]:

```
residual = x0_int[0,:,:,:]
residual_reshaped = torch.reshape(residual,(residual.size(0),-1))
out_after_residual = out + residual_reshaped
```

[83]:

```
## dump all the files

## Helper functions to dump files
def dec_to_bin(arr,bit):
    bin_arr = []
    for a in arr:
        a = int(a)
        if a < 0:
            a+=bit
        b = '{0:04b}'.format(int(a))
        bin_arr.append(b)
    return bin_arr

def dec_to_bin_psum(arr,bit):
    bin_arr = []
    for a in arr:
        a = int(a)
        sign = 0
        if a < 0:
            sign = 1
            a+=bit
        b = '{0:16b}'.format(int(a))
        if sign == 1:
            b = b.replace(" ", "1")
        else:
            b = b.replace(" ","0")
        bin_arr.append(b)
    return bin_arr


def convert_to_list(arr):
    return arr.tolist()
```

[84]:

```
## Activation dump

fp_act = open('Renet_activation_project.txt','w')
fp_act.write("#####\n")
fp_act.write("#####\n")
fp_act.write("#####\n")
fp_act_dec = open('Resnet_activation_dec.txt','w')
for i in range(a_pad.size()[1]):
    act_line = a_pad[:,i]
    act_arr = [int(j+0.001) for j in convert_to_list(act_line)]
    for a in act_arr[::-1]:
        #print(int(a))
        fp_act_dec.write(str(int(a)))
        fp_act_dec.write(" ")
    fp_act_dec.write("\n")
    bin_act = dec_to_bin(act_arr,16)
    for b in bin_act[::-1]:
        #print(b)
        fp_act.write(b)
    fp_act.write('\n')


fp_act.close()
fp_act_dec.close()
```

[85]:

```
## Weight dump
fp_wgt = open('Resnet_weight_project.txt','w')
fp_wgt.write("#####\n")
fp_wgt.write("#####\n")
fp_wgt.write("#####\n")
fp_wgt_dec = open('Resnet_weight_dec.txt','w')

for kij in range(9):
    for w in range(8):
        w_line = w_int[w,:,kij]
        w_arr = []
        for i in convert_to_list(w_line):
            if i < 0:
                w_arr.append(int(i-0.001))
            else:
                w_arr.append(int(i+0.001))

        for ww in w_arr[::-1]:
            #print(int(ww))
            fp_wgt_dec.write(str(int(ww)))
            fp_wgt_dec.write(" ")
        fp_wgt_dec.write("\n")
        bin_wgt = dec_to_bin(w_arr,16)
        for b in bin_wgt[::-1]:
            #print("Count = {} , b = {}".format(count,b))
            fp_wgt.write(b)
        fp_wgt.write('\n')

fp_wgt.close()
fp_wgt_dec.close()
```

[86]:

```
## psum dump and output.txt dump
fp_psum = open('Resnet_psum_project.txt','w')
fp_psum.write("#####\n")
fp_psum.write("#####\n")
fp_psum.write("#####\n")

fp_psum_dec = open('Resnet_psum_dec.txt','w')
fp_psum_relu = open('Resnet_output_project.txt','w')
fp_psum_relu.write("#\n")
fp_psum_relu.write("#\n")
fp_psum_relu.write("#\n")
for kij in range(o_ni_dim*o_ni_dim):
    psum_line = out[:,kij]
    psum_arr = []
    out_arr = []
    for i in convert_to_list(psum_line):
        if i < 0:
            psum_arr.append(int(i-0.001))
            out_arr.append(0)
        else:
            psum_arr.append(int(i+0.001))
            out_arr.append(int(i+0.001))
    for p in psum_arr:
        #print(int(a))
        fp_psum_dec.write(str(int(p)))
        fp_psum_dec.write(" ")
    fp_psum_dec.write("\n")
    bin_act = dec_to_bin_psum(psum_arr,65536)
    out_bin_act = dec_to_bin_psum(out_arr,65536)
    for b in bin_act:
        #print(b)
        fp_psum.write(b)
    fp_psum.write('\n')

    for b in out_bin_act:
        fp_psum_relu.write(b)
    fp_psum_relu.write("\n")
fp_psum.close()
fp_psum_dec.close()
fp_psum_relu.close()
```

[91]:

```
### Residual file dump
fp_act = open('Renet_residual_project.txt','w')
fp_act.write("#####\n")
fp_act.write("#####\n")
fp_act.write("#####\n")
fp_act_dec = open('Resnet_residual_dec.txt','w')
for i in range(o_ni_dim*o_ni_dim):
    act_line = residual_reshaped[:,i]
    act_arr = [int(j+0.001) for j in convert_to_list(act_line)]
    for a in act_arr[::-1]:
        #print(int(a))
        fp_act_dec.write(str(int(a)))
        fp_act_dec.write(" ")
    fp_act_dec.write("\n")
    bin_act = dec_to_bin(act_arr,16)
    for b in bin_act[::-1]:
        #print(b)
        fp_act.write(b)
    fp_act.write('\n')


fp_act.close()
fp_act_dec.close()
```

[ ]:

```
## For input activations tiling. Handled in testbench. Written here just for␣
 ↪reference logic
hardware_ni_dim = 6
hor_step = hardware_ni_dim-kernel_dim+1 ## 4
ver_step = (hardware_ni_dim-kernel_dim+1)*a_pad_ni_dim ## 136
stop_point = (a_pad_ni_dim-hardware_ni_dim)*a_pad_ni_dim+1 ## 953
group_count = 0
act_arr = []
```

```
## Vertical movement loop
for v in range(0,stop_point,ver_step):
    ## Now move horizontally
    for h in range(v,v+34,hor_step):
        if h+hardware_ni_dim>v+a_pad_ni_dim:
            break
        group = []
        group_count+=1
        for hh in range(h,h+(hardware_ni_dim)*a_pad_ni_dim,a_pad_ni_dim):

            for hhh in range(hardware_ni_dim):
                group.append(hh+hhh)
        act_group.append(group)
```

[ ]:

```
## For output tiling. Handled in testbench. Written here just for reference logic
our_out_hw = 4
out_ni_dim = int(math.sqrt(out.size(1)))
out_stop_point = (out_ni_dim-our_out_hw)*out_ni_dim+1
out_ver_step = our_out_hw*out_ni_dim
group_count = 0
out_group = []
for v in range(0,out_stop_point,out_ver_step):
    ## Now move horizontally
    for h in range(v,v+32,our_out_hw):
        group = []
        group_count+=1
        for hh in range(h,h+our_out_hw*out_ni_dim,out_ni_dim):
            for hhh in range(our_out_hw):
                group.append(hh+hhh)
        out_group.append(group)
```