Vanilla_4BitInput_4BitWeight

December 13, 2025

[41]:
```
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 *
from collections import OrderedDict


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



batch_size = 128
model_name = "VGG16_quant_aware_trained"
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)
```

```
        # error = torch.mean(torch.abs(captured['psum_recovered'] -␣
 ↪captured['next_input']))

        # After output = model(input), inside train() or validate()
        # if 'psum_recovered' in captured and 'next_input' in captured:
        #     # Apply ReLU to recovered psum to match next layer's input
        #     psum_relu = F.relu(captured['psum_recovered'])
        #     next_input = captured['next_input']
        #     error = torch.mean(torch.abs(psum_relu - next_input)).item()

        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)

            # if 'psum_recovered' in captured and 'next_input' in captured:
            #     # Apply ReLU to recovered psum to match next layer's input
            #     psum_relu = F.relu(captured['psum_recovered'])
            #     next_input = captured['next_input']
            #     error = torch.mean(torch.abs(psum_relu - next_input)).item()

            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, 8, kernel_size=(1, 1), stride=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (28): ReLU()
    (29): QuantConv2d(
      8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (30): ReLU()
    (31): QuantConv2d(
      8, 512, kernel_size=(1, 1), stride=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (32): ReLU()
    (33): QuantConv2d(
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (34): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (35): ReLU(inplace=True)
    (36): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (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): QuantConv2d(
      512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
      (weight_quant): weight_quantize_fn()
    )
    (44): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (45): ReLU(inplace=True)
    (46): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
    (47): 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
```

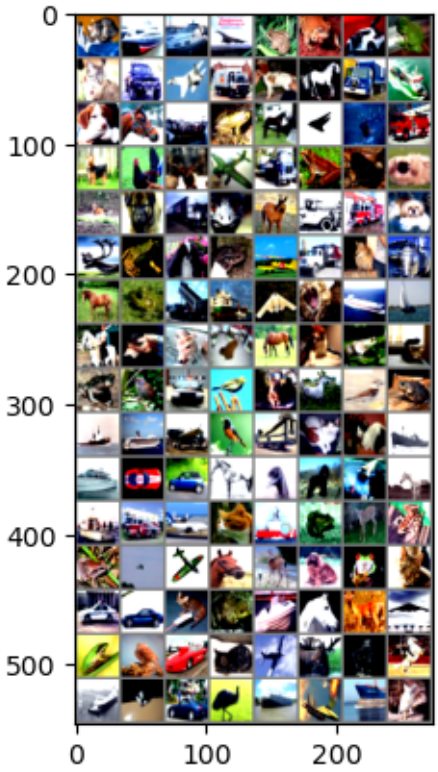
[42]:
```
import matplotlib.pyplot as plt
import numpy as np

def imshow(img):
    img = img / 2 + 0.5     # unnormalize
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1, 2, 0)))
    plt.show()

dataiter = iter(testloader)
images, labels = next(dataiter) ## If you run this line, the next data batch is␣
 ↪called subsequently.

imshow(torchvision.utils.make_grid(images))
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for
floats or [0..255] for integers).
```



[ ]:
```
# This cell won't be given, but students will complete the training

lr = 0.02
weight_decay = 5e-4
epochs = 170
best_prec = 0

#model = nn.DataParallel(model).cuda()
model.cuda()

criterion = nn.CrossEntropyLoss(label_smoothing=0.1).cuda()
optimizer = torch.optim.SGD(model.parameters(), lr=lr, momentum=0.9,␣
 ↪weight_decay=weight_decay)
scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, T_max=epochs)
#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)
```

[27]:
```
fdir = 'result/VGG16_quant/model_best.pth.tar'

checkpoint = torch.load(fdir)
model.load_state_dict(checkpoint['state_dict'])


criterion = nn.CrossEntropyLoss().cuda()

model.eval()
model.cuda()

print('Accuracy for base case')
prec = validate(testloader, model, criterion)
print('First conv layer's weights' absolute sum = ', model.features[0].weight.
  ↪abs().sum())
```

```
Accuracy for base case
Test: [0/79]    Time 0.107 (0.107)      Loss 0.2514 (0.2514)    Prec 94.531%
(94.531%)
 * Prec 92.020%
First conv layer's weights' absolute sum =  tensor(117.6888, device='cuda:0',
grad_fn=<SumBackward0>)
```

[10]:
```
import torch
import torch.nn as nn
import torch.nn.functional as F
```

```
from models.quant_layer import QuantConv2d

DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# ---- find the 8×8 squeezed QuantConv2d ----
features = model.features
sq_idx = None
for i, m in enumerate(features):
    if isinstance(m, QuantConv2d) and m.in_channels == 8 and m.out_channels ==␣
 ↪8:
        sq_idx = i
        break
assert sq_idx is not None, "No 8×8 squeezed conv found"
sq = features[sq_idx]

# next real layer
nx = sq_idx + 1
while isinstance(features[nx], (nn.BatchNorm2d, nn.ReLU, nn.Identity)):
    nx += 1
next_layer = features[nx]

# ---- hook inputs ----
_cache = {}
captured = {}
def hook_sq(m, inp): captured["x"] = inp[0].detach().to(DEVICE)
def hook_nx(m, inp): captured["y"] = inp[0].detach().to(DEVICE)

h1 = sq.register_forward_pre_hook(hook_sq)
h2 = next_layer.register_forward_pre_hook(hook_nx)

# ---- run 1 batch ----
dummy = torch.randn(1, 3, 32, 32).to(DEVICE)
with torch.no_grad():
    model(dummy)

h1.remove()
h2.remove()

assert "x" in captured
x_in = captured["x"]

# ---- quantization params ----
def qparams(alpha, nbit, signed):
    if signed:
        qmax = (2**(nbit-1))-1
        qmin = -(2**(nbit-1))
    else:
```

```
        qmax = (2**nbit)-1
        qmin = 0
    scale = alpha / qmax  # use scale only
    return scale

# ---- quantize weights ----
w_q = getattr(sq, "weight_q", None)
w = sq.weight.detach()
alpha_w = w.abs().max()
d_w = qparams(alpha_w, 4, True)  # compute scale in all cases

if w_q is None:
    w_int = torch.clamp(torch.round(w / d_w), -(2**3), (2**3)-1)
    w_q = w_int * d_w
else:
    w_q = w_q.detach()

# ---- quantize activation ----
aq = getattr(sq, "act_quant", None)
alpha_x = getattr(aq, "alpha", x_in.abs().max())
signed_x = getattr(aq, "signed", False)

d_x = qparams(alpha_x, 4, signed_x)  # only scale
x_int = torch.round(x_in / d_x)
x_q = x_int * d_x

# ---- integer conv ----
stride, pad, groups = sq.stride, sq.padding, sq.groups
bias = sq.bias

psum_int = F.conv2d(x_int.float(), torch.round(w_q/d_w).float(),
                    bias=None, stride=stride, padding=pad, groups=groups)
psum_fp = psum_int * (d_x * d_w)
if bias is not None:
    psum_fp = psum_fp + bias.view(1,-1,1,1)
psum_relu = torch.clamp(psum_fp, min=0.0)

# ---- reference conv(x_q, w_q) ----
conv_ref = nn.Conv2d(
    in_channels=w_q.size(1),
    out_channels=w_q.size(0),
    kernel_size=w_q.shape[2:],
    stride=stride,
    padding=pad,
    bias=(bias is not None)
).to(DEVICE)
conv_ref.weight = nn.Parameter(w_q.clone())
```

```
if bias is not None:
    conv_ref.bias = nn.Parameter(bias.clone())

with torch.no_grad():
    y_ref = torch.clamp(conv_ref(x_q), min=0.0)

# ---- final quantization MSE ----
mse = (psum_relu - y_ref).pow(2).mean().item()
print(f"4A/4W Quantization MSE: {mse:.6e}")
```

```
4A/4W Quantization MSE: 2.620530e-07
```

[5]:
```
def compute_sparsity(model):
    total_zeros = 0
    total_params = 0
    layer_wise = {}
    for name, module in model.named_modules():
        if isinstance(module, nn.Conv2d) or isinstance(module, nn.Linear):
            weight = module.weight.data
            zeros = torch.sum(weight == 0).item()
            params = weight.numel()
            sparsity = zeros / params * 100
            total_zeros += zeros
            total_params += params
            layer_wise[name] = sparsity
    total_sparsity = total_zeros / total_params * 100
    return layer_wise, total_sparsity

def compute_macs_vgg_quant(model, input_size=(3,32,32)):
    C_in, H, W = input_size
    total_macs = 0
    for name, module in model.named_modules():
        if isinstance(module, nn.Conv2d):
            C_out, C_in_layer, kH, kW = module.weight.shape
            strideH, strideW = module.stride
            padH, padW = module.padding
            H_out = (H + 2*padH - kH)//strideH + 1
            W_out = (W + 2*padW - kW)//strideW + 1
            total_macs += H_out * W_out * C_out * C_in_layer * kH * kW
            H, W, C_in = H_out, W_out, C_out
        elif isinstance(module, nn.Linear):
            total_macs += module.weight.shape[0] * module.weight.shape[1]
    return total_macs

def compute_model_size(model, bits_per_weight=32):
    total_params = 0
    for module in model.modules():
```

```
        if isinstance(module, nn.Conv2d) or isinstance(module, nn.Linear):
            total_params += module.weight.numel()
    return total_params * bits_per_weight / 8 / 1024**2
```

[ ]:

[ ]:
```
# HW

#  1. Train with 4 bits for both weight and activation to achieve >90% accuracy
#  2. Find x_int and w_int for the 2nd convolution layer
#  3. Check the recovered psum has similar value to the un-quantized original␣
 ↪psum
#     (such as example 1 in W3S2)
```

[6]:
```
import torch
import torch.nn as nn

# ---------------- Helper functions ----------------
def compute_sparsity(model):
    total_zeros = 0
    total_params = 0
    layer_wise = {}
    for name, module in model.named_modules():
        if isinstance(module, nn.Conv2d) or isinstance(module, nn.Linear):
            weight = module.weight.data
            zeros = torch.sum(weight == 0).item()
            params = weight.numel()
            sparsity = zeros / params * 100
            total_zeros += zeros
            total_params += params
            layer_wise[name] = sparsity
    total_sparsity = total_zeros / total_params * 100
    return layer_wise, total_sparsity

def compute_macs_vgg_quant(model, input_size=(3,32,32)):
    C_in, H, W = input_size
    total_macs = 0
    for name, module in model.named_modules():
        if isinstance(module, nn.Conv2d):
            C_out, C_in_layer, kH, kW = module.weight.shape
            strideH, strideW = module.stride
            padH, padW = module.padding
            H_out = (H + 2*padH - kH)//strideH + 1
            W_out = (W + 2*padW - kW)//strideW + 1
            total_macs += H_out * W_out * C_out * C_in_layer * kH * kW
            H, W, C_in = H_out, W_out, C_out
        elif isinstance(module, nn.Linear):
```

```
            total_macs += module.weight.shape[0] * module.weight.shape[1]
    return total_macs

def compute_model_size(model, bits_per_weight=32):
    total_params = 0
    for module in model.modules():
        if isinstance(module, nn.Conv2d) or isinstance(module, nn.Linear):
            weight = module.weight.data
            total_params += torch.sum(weight != 0).item()  # count only␣
 ↪non-zero weights
    return total_params * bits_per_weight / 8 / 1024**2  # in MB

# ---------------- Load Unpruned Model ----------------
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()

# Compute Accuracy for Unpruned
correct = 0
with torch.no_grad():
    for data, target in testloader:
        data, target = data.to(device), target.to(device)
        output = model(data)
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

unpruned_acc = 100. * correct / len(testloader.dataset)
unpruned_macs = compute_macs_vgg_quant(model, input_size=(3,32,32))
unpruned_model_size = compute_model_size(model, bits_per_weight=4)  # quantized␣
 ↪int4

print("Metrics for Unpruned Model ---")
print(f"Accuracy: {unpruned_acc:.2f}%")
print(f"MACs: {unpruned_macs}")
print(f"Model Size (int4): {unpruned_model_size:.2f} MB")

# ---------------- Load Pruned Model ----------------
save_path = "result/VGG16_quant/model_pruned_finetuned.pth.tar"
checkpoint = torch.load(save_path)
model.load_state_dict(checkpoint['state_dict'])
model.cuda()
model.eval()

# Compute Accuracy for Pruned
```

```
correct = 0
with torch.no_grad():
    for data, target in testloader:
        data, target = data.to(device), target.to(device)
        output = model(data)
        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

pruned_acc = 100. * correct / len(testloader.dataset)
pruned_macs = compute_macs_vgg_quant(model, input_size=(3,32,32))
pruned_model_size = compute_model_size(model, bits_per_weight=4)  # quantized␣
 ↪int4

print("\nMetrics for Pruned Model ---")
print(f"Accuracy: {pruned_acc:.2f}%")
print(f"MACs: {pruned_macs}")
print(f"Model Size (int4): {pruned_model_size:.2f} MB")

# ---------------- Memory Bandwidth Reduction ----------------
reduction_percent = (unpruned_model_size - pruned_model_size) /␣
 ↪unpruned_model_size * 100
print(f"Estimated Memory Bandwidth Reduction: {reduction_percent:.2f}%")
layer_wise_sparsity, total_sparsity = compute_sparsity(model)
print(f"\nTotal model sparsity: {total_sparsity:.2f}%")
```

```
Metrics for Unpruned Model ---
Accuracy: 92.02%
MACs: 12656579584
Model Size (int4): 5.90 MB

Metrics for Pruned Model ---
Accuracy: 88.82%
MACs: 12656579584
Model Size (int4): 2.95 MB
Estimated Memory Bandwidth Reduction: 49.98%

Total model sparsity: 49.98%
```

[7]:
```
#send an input and grap the value by using prehook like HW3
```

[14]:
```
import torch
import torch.nn.functional as F
import numpy as np

conv_layer = model.features[29]  # your squeezed quantized conv layer
w_bit = 4
x_bit = 4
```

```
# ---------- Stochastic rounding function ----------
def stochastic_round(tensor):
    floor = torch.floor(tensor)
    prob = tensor - floor
    return floor + torch.bernoulli(prob)

# ---------- Quantized weights ----------
weight_q = conv_layer.weight_q                        # stored during training
w_alpha = conv_layer.weight_quant.wgt_alpha.data

# Symmetric quantization delta (corrected)
w_delta = w_alpha / (2**(w_bit - 1) - 1)

# Use stochastic rounding instead of torch.round
weight_int = stochastic_round(weight_q / w_delta).to(torch.int32)
# print(weight_int.size())
W_int= torch.reshape(weight_int, (weight_int.size(0),weight_int.size(1),-1))
# print(W_int.size())
Kij = 8;
W = W_int[:,:,Kij]
# print(W.size())

# ---------- Quantized input activations ----------
x = captured['x']                                      # input tensor to layer 29
x_alpha = conv_layer.act_alpha.data

# Symmetric activations quantization (0 to   → ReLU)
x_delta = x_alpha / (2**x_bit - 1)

# Quantize using same function used during training
act_quant_fn = conv_layer.act_alq
x_q = act_quant_fn(x, x_alpha)

x_int = stochastic_round(x_q / x_delta).to(torch.int32)
# print(x_int.size())
x_int_b0 = x_int[0,:,:,:]
# print(x_int_b0.size())
X = torch.reshape(x_int_b0, (x_int_b0.size(0),-1)).float().cuda()
# print(X.size())


# ---------- INT psum simulation (Hardware-like) ----------
output_int = F.conv2d(x_int.float(), weight_int.float(), bias=None,
                      stride=conv_layer.stride,
                      padding=conv_layer.padding)
# print(output_int.size())
```

```
output_recovered = torch.relu(output_int * (x_delta * w_delta))

output_fp32 = torch.relu(conv_layer(x))

difference = torch.abs(output_fp32 - output_recovered)
print('Mean difference:', difference.mean().item())
# np.savetxt("weight_int_layer29.txt", weight_int.cpu().numpy().flatten(),␣
 ↪fmt="%d", delimiter=",")
# np.savetxt("output_recovered.txt", output_recovered.cpu().numpy().flatten(),␣
 ↪fmt="%d", delimiter=",")
```

```
Mean difference: 1.5890691429376602e-08
```

[ ]:
```
import torch


bit_precision = 4
file = open('activation_final.txt', 'w') #write to file
file.write('#time0row7[msb-lsb],time0row6[msb-lst],....,time0row0[msb-lst]#\n')
file.write('#time1row7[msb-lsb],time1row6[msb-lst],....,time1row0[msb-lst]#\n')
file.write('#................#\n')

for i in range(X.size(1)):  # time step
    for j in range(X.size(0)): # row #
        X_bin = '{0:04b}'.format(round(X[7-j,i].item()))
        for k in range(bit_precision):
            file.write(X_bin[k])
        #file.write(' ')  # for visibility with blank between words, you can use
    file.write('\n')
file.close()

p_nij = range(X.size(1))
psum = torch.zeros(8,16,9).cuda()
print(W_int[0,0,0])

print(psum.size())

psum = torch.zeros(8, 16, 9).cuda()

for kij in range(9):
    for nij in range(16):
        # Use raw signed weights, do NOT convert them to 0..15
        weight = W_int[:, :, kij].float().cuda()   # keep negative values intact
        x_input = X[:, nij].unsqueeze(0)           # shape [1,8]
        psum[:, nij, kij] = torch.matmul(weight, x_input.squeeze(0))  # matmul␣
 ↪preserves sign
```

```
kij = 8

psum_int = psum[:,:,kij]
print(W_int[0,0,0])
print(X[0,0])
print(psum_int[0,0])

bit_precision = 16

bit_precision = 4
file = open('weight_kij_8.txt', 'w') #write to file
file.write('#time0row7[msb-lsb],time0row6[msb-lst],....,time0row0[msb-lst]#\n')
file.write('#time1row7[msb-lsb],time1row6[msb-lst],....,time1row0[msb-lst]#\n')
file.write('#................#\n')

for i in range(W.size(1)):
    for j in range(W.size(0)):
       if (W[7-j,i] < 0):
        W[7-j,i] = W[7-j,i] + 16
       W_bin = '{0:04b}'.format(round(W[7-j,i].item()))
       for k in range(bit_precision):
           file.write(W_bin[k])
       #file.write(' ')  # for visibility with blank between words, you can use
    file.write('\n')
file.close()

def int_to_unsigned_bits(x, bits=4):
    """Convert integer x to unsigned binary string of length `bits` (0 to␣
 ↪2^bits-1)."""
    max_val = (1 << bits) - 1  # 15 for 4 bits
    x_clipped = max(min(int(x), max_val), 0)  # clip to 0..15
    return format(x_clipped, '0{}b'.format(bits))

def write_activation_stream_row_major_unsigned(x_int,␣
 ↪filename="x_int_layer29_hw_row_unsigned.txt",
                                               batch_index=0, bits=4):
    """
    Converts quantized tensor x_int to 4-bit unsigned binary stream, **row by␣
 ↪row**.
    x_int: [B, 8, H, W]
    batch_index: which batch to export
    bits: bit precision
    """
    assert x_int.shape[1] == 8, "This helper expects 8 channels."
```

```
    X = x_int[batch_index].cpu().detach().clone()  # [C=8, H, W]
    C, H, W = X.shape

    with open(filename, 'w') as f:
        f.write('# Activation stream (row-major, unsigned): batch_index={}, 8␣
 ↪channels, {}-bit\n'.format(batch_index, bits))
        f.write('# Each line: one cycle, channels 7->0 concatenated\n')
        f.write('# Scanning order: for h in range(H): for w in range(W)\n')

        for h in range(H):         # row-first
            for w in range(W):     # column inside the row
                line = ''
                for ch in reversed(range(C)):  # ch7 -> ch0
                    val = X[ch, h, w].item()
                    line += int_to_unsigned_bits(val, bits)
                f.write(line + '\n')

    print(f"Written hardware stream (row-major, unsigned) to {filename}")
    return filename

# Usage
write_activation_stream_row_major_unsigned(x_int,␣
 ↪filename="x_int_layer29_hw_row_unsigned.txt", batch_index=0, bits=4)

# ---------- Function to convert signed int to 2's complement bits ----------
def int_to_signed_bits(x, bits=4):
    """Convert integer x in range [-2^(bits-1), 2^(bits-1)-1] to 2's complement␣
 ↪binary string."""
    min_val = -(1 << (bits-1))
    max_val = (1 << (bits-1)) - 1
    x_clipped = max(min(int(x), max_val), min_val)
    if x_clipped < 0:
        x_clipped = (1 << bits) + x_clipped
    return format(x_clipped, '0{}b'.format(bits))

# ---------- Write weights row-major for weight-stationary array ----------
def write_weights_row_major_signed(weight_int, filename="weight_layer29_hw_row.
 ↪txt", bits=4):
    """
    Converts conv weights to 4-bit signed binary stream, row by row (input␣
 ↪channel dimension is row).
    weight_int: [out_channels, in_channels, kH, kW]
    """
    O, I, kH, kW = weight_int.shape
    with open(filename, 'w') as f:
        f.write(f'# Weight stream (row-major, signed): {O}x{I}x{kH}x{kW},␣
 ↪{bits}-bit\n')
```

```
        f.write('# Each line: one cycle, input channels 0->7 concatenated\n')
        # For weight-stationary: scan output channels (rows) and kernel spatial␣
 ↪positions
        for oc in range(O):
            for ic in range(I):
                for kh in range(kH):
                    for kw in range(kW):
                        val = weight_int[oc, ic, kh, kw].item()
                        bits_str = int_to_signed_bits(val, bits)
                        f.write(bits_str)
                    f.write('\n')  # each line is one kernel element across␣
 ↪channels
    # print(f"Written weight stream (row-major, signed) to {filename}")
    return filename

# ---------- Usage ----------
write_weights_row_major_signed(weight_int, filename="weight_layer29_hw_row.
 ↪txt", bits=4)

# ---------- Write INT or Recovered outputs ----------
def write_output_stream_row_major(output_tensor, filename="output_hw_row.txt",␣
 ↪bits=4, signed=False, batch_index=0):
    """
    Converts output tensor to 4-bit row-major binary stream for hardware.
    output_tensor: [B, C, H, W]
    signed: whether to use signed 2's complement (-8..7) or unsigned (0..15)
    """
    X = output_tensor[batch_index].cpu().detach().clone()  # [C, H, W]
    C, H, W = X.shape

    if signed:
        int_to_bits = int_to_signed_bits
    else:
        # unsigned helper
        def int_to_bits(x, bits=4):
            x_clipped = max(min(int(round(x)), (1<<bits)-1), 0)
            return format(x_clipped, '0{}b'.format(bits))

    with open(filename, 'w') as f:
        f.write(f'# Output stream (row-major, {"signed" if signed else␣
 ↪"unsigned"}): {C}x{H}x{W}, {bits}-bit\n')
        f.write('# Each line: one cycle, channels 7->0 concatenated\n')
        for h in range(H):        # row-first
            for w in range(W):    # column inside row
                line = ''
                for ch in reversed(range(C)):  # ch7->ch0
                    val = X[ch, h, w].item()
```

```
                    line += int_to_bits(val, bits)
                f.write(line + '\n')

    # print(f"Written output stream (row-major) to {filename}")
    return filename

# ---------- Usage ----------
write_output_stream_row_major(output_int, filename="output_int_hw_row.txt",␣
 ↪bits=4, signed=True)
write_output_stream_row_major(output_recovered,␣
 ↪filename="output_recovered_hw_row.txt", bits=4, signed=False)
```

[43]:
```
import torch
import torch.nn.utils.prune as prune
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()
criterion = nn.CrossEntropyLoss().cuda()

# print("Accuracy of the pre-trained model:")
# validate(testloader, model, criterion)

# print("\n Applying 80% Unstructured Pruning:")
# for name, module in model.named_modules():
#     if isinstance(module, QuantConv2d):
#         prune.random_unstructured(module, name='weight', amount=0.8)

# print("Checking the accuracy after performing 80% unstructured pruning:")
# validate(testloader, model, criterion)

# print("\n Training the unstructured-pruned model for gaining back accuracy :")
best_prec = 0
epochs = 250
optimizer = torch.optim.SGD(model.parameters(), lr=5e-3, momentum=0.9,␣
 ↪weight_decay=1e-4)

# for epoch in range(epochs):
#     train(trainloader, model, criterion, optimizer, epoch)
#     prec = validate(testloader, model, criterion)
#     is_best = prec > best_prec
#     best_prec = max(prec, best_prec)

# print(f"Best accuracy after re-training the unstructured-pruned model:
 ↪{best_prec:.3f}%")
```

```
# for name, module in model.named_modules():
#     if isinstance(module, QuantConv2d):
#         prune.remove(module, 'weight')

print("\n Loading the original model again for structured pruning: ")
checkpoint = torch.load(PATH)
model.load_state_dict(checkpoint['state_dict'])
model.cuda()

print("\nApplying 50% Structured Pruning:")
for name, module in model.named_modules():
    if isinstance(module, QuantConv2d):
        prune.ln_structured(module, name='weight', amount=0.5, n=1, dim=0)

print("Checking the accuracy after performing 50% structured pruning:")
validate(testloader, model, criterion)

print("\n Training the structured-pruned model for gaining back accuracy :")
best_prec = 0
optimizer = torch.optim.SGD(model.parameters(), lr=5e-3, momentum=0.9,␣
 ↪weight_decay=1e-4)

for epoch in range(epochs):
    train(trainloader, model, criterion, optimizer, epoch)
    prec = validate(testloader, model, criterion)
    is_best = prec > best_prec
    best_prec = max(prec, best_prec)

print(f"Best accuracy after finetuning the structured-pruned model: {best_prec:.
 ↪3f}%")
for name, module in model.named_modules():
    if isinstance(module, QuantConv2d):
        prune.remove(module, 'weight')

# --- Save final pruned & fine-tuned model ---
save_path = "result/VGG16_quant/model_pruned_finetuned_pt1.pth.tar"

torch.save({
    'state_dict': model.state_dict(),
    'best_prec': best_prec,
}, save_path)

print("Saved pruned + fine-tuned model to:", save_path)
```

```
Loading the original model again for structured pruning:
```

```
Applying 50% Structured Pruning:
Checking the accuracy after performing 50% structured pruning:
Test: [0/79]    Time 0.235 (0.235)      Loss 2.6400 (2.6400)    Prec 10.156%
(10.156%)
 * Prec 10.000%

 Training the structured-pruned model for gaining back accuracy :
Epoch: [0][0/391]       Time 0.249 (0.249)      Data 0.209 (0.209)      Loss
2.4074 (2.4074)    Prec 15.625% (15.625%)
Epoch: [0][100/391]     Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
2.1087 (2.2385)    Prec 20.312% (15.780%)
Epoch: [0][200/391]     Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
1.8001 (2.1164)    Prec 33.594% (19.831%)
Epoch: [0][300/391]     Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
1.6515 (1.9922)    Prec 39.062% (24.307%)
Test: [0/79]    Time 0.223 (0.223)      Loss 1.4444 (1.4444)    Prec 48.438%
(48.438%)
 * Prec 46.560%
Epoch: [1][0/391]       Time 0.295 (0.295)      Data 0.264 (0.264)      Loss
1.3570 (1.3570)    Prec 49.219% (49.219%)
Epoch: [1][100/391]     Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
1.1242 (1.2156)    Prec 62.500% (56.049%)
Epoch: [1][200/391]     Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
1.0093 (1.1122)    Prec 65.625% (60.479%)
Epoch: [1][300/391]     Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.5857 (0.9944)    Prec 80.469% (65.264%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.5255 (0.5255)    Prec 82.812%
(82.812%)
 * Prec 80.760%
Epoch: [2][0/391]       Time 0.340 (0.340)      Data 0.310 (0.310)      Loss
0.4631 (0.4631)    Prec 86.719% (86.719%)
Epoch: [2][100/391]     Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.5288 (0.4743)    Prec 85.938% (85.125%)
Epoch: [2][200/391]     Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.4960 (0.4502)    Prec 84.375% (85.735%)
Epoch: [2][300/391]     Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.3404 (0.4342)    Prec 89.062% (86.156%)
Test: [0/79]    Time 0.146 (0.146)      Loss 0.3804 (0.3804)    Prec 89.844%
(89.844%)
 * Prec 82.920%
Epoch: [3][0/391]       Time 0.333 (0.333)      Data 0.302 (0.302)      Loss
0.3961 (0.3961)    Prec 87.500% (87.500%)
Epoch: [3][100/391]     Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.3216 (0.3248)    Prec 88.281% (89.248%)
Epoch: [3][200/391]     Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.3182 (0.3262)    Prec 89.062% (89.269%)
Epoch: [3][300/391]     Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.3123 (0.3284)    Prec 89.062% (89.210%)
Test: [0/79]    Time 0.225 (0.225)      Loss 0.3357 (0.3357)    Prec 90.625%
(90.625%)
 * Prec 85.200%
Epoch: [4][0/391]       Time 0.282 (0.282)      Data 0.244 (0.244)      Loss
0.2292 (0.2292)    Prec 90.625% (90.625%)
Epoch: [4][100/391]     Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.2879 (0.2793)    Prec 88.281% (90.586%)
Epoch: [4][200/391]     Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.3678 (0.2834)    Prec 87.500% (90.524%)
Epoch: [4][300/391]     Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.2637 (0.2852)    Prec 92.188% (90.586%)
Test: [0/79]    Time 0.154 (0.154)      Loss 0.3229 (0.3229)    Prec 87.500%
(87.500%)
 * Prec 85.860%
Epoch: [5][0/391]       Time 0.263 (0.263)      Data 0.225 (0.225)      Loss
0.3189 (0.3189)    Prec 86.719% (86.719%)
Epoch: [5][100/391]     Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.2503 (0.2650)    Prec 89.844% (91.043%)
Epoch: [5][200/391]     Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.1710 (0.2682)    Prec 93.750% (90.951%)
Epoch: [5][300/391]     Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.3310 (0.2644)    Prec 88.281% (91.051%)
Test: [0/79]    Time 0.249 (0.249)      Loss 0.3008 (0.3008)    Prec 90.625%
(90.625%)
 * Prec 85.340%
Epoch: [6][0/391]       Time 0.338 (0.338)      Data 0.299 (0.299)      Loss
0.1258 (0.1258)    Prec 96.875% (96.875%)
Epoch: [6][100/391]     Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.2004 (0.2402)    Prec 93.750% (92.002%)
Epoch: [6][200/391]     Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1699 (0.2465)    Prec 93.750% (91.764%)
Epoch: [6][300/391]     Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1813 (0.2474)    Prec 94.531% (91.687%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.2928 (0.2928)    Prec 86.719%
(86.719%)
 * Prec 87.750%
Epoch: [7][0/391]       Time 0.356 (0.356)      Data 0.316 (0.316)      Loss
0.1484 (0.1484)    Prec 96.094% (96.094%)
Epoch: [7][100/391]     Time 0.048 (0.052)      Data 0.001 (0.004)      Loss
0.2153 (0.2323)    Prec 92.969% (92.327%)
Epoch: [7][200/391]     Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.2933 (0.2357)    Prec 89.062% (92.102%)
Epoch: [7][300/391]     Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.3160 (0.2400)    Prec 89.844% (92.011%)
Test: [0/79]    Time 0.246 (0.246)      Loss 0.4413 (0.4413)    Prec 88.281%
(88.281%)
 * Prec 87.190%
```

```
Epoch: [8][0/391]       Time 0.373 (0.373)      Data 0.333 (0.333)      Loss
0.2099 (0.2099)    Prec 93.750% (93.750%)
Epoch: [8][100/391]     Time 0.050 (0.053)      Data 0.001 (0.005)      Loss
0.1387 (0.2159)    Prec 93.750% (92.837%)
Epoch: [8][200/391]     Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1082 (0.2195)    Prec 97.656% (92.642%)
Epoch: [8][300/391]     Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.1701 (0.2219)    Prec 94.531% (92.481%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.3168 (0.3168)    Prec 88.281%
(88.281%)
 * Prec 87.050%
Epoch: [9][0/391]       Time 0.329 (0.329)      Data 0.293 (0.293)      Loss
0.1447 (0.1447)    Prec 95.312% (95.312%)
Epoch: [9][100/391]     Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1655 (0.2034)    Prec 94.531% (93.224%)
Epoch: [9][200/391]     Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.2828 (0.2064)    Prec 93.750% (93.163%)
Epoch: [9][300/391]     Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0752 (0.2086)    Prec 97.656% (93.070%)
Test: [0/79]    Time 0.147 (0.147)      Loss 0.3421 (0.3421)    Prec 89.062%
(89.062%)
 * Prec 87.970%
Epoch: [10][0/391]      Time 0.322 (0.322)      Data 0.291 (0.291)      Loss
0.1955 (0.1955)    Prec 92.969% (92.969%)
Epoch: [10][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.2180 (0.2011)    Prec 91.406% (93.247%)
Epoch: [10][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1796 (0.2006)    Prec 92.969% (93.241%)
Epoch: [10][300/391]    Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.2487 (0.2054)    Prec 89.062% (93.015%)
Test: [0/79]    Time 0.162 (0.162)      Loss 0.3133 (0.3133)    Prec 90.625%
(90.625%)
 * Prec 87.550%
Epoch: [11][0/391]      Time 0.310 (0.310)      Data 0.280 (0.280)      Loss
0.1404 (0.1404)    Prec 96.094% (96.094%)
Epoch: [11][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1596 (0.1873)    Prec 94.531% (93.735%)
Epoch: [11][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1898 (0.1965)    Prec 95.312% (93.354%)
Epoch: [11][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1694 (0.1955)    Prec 93.750% (93.381%)
Test: [0/79]    Time 0.143 (0.143)      Loss 0.3160 (0.3160)    Prec 89.844%
(89.844%)
 * Prec 86.560%
Epoch: [12][0/391]      Time 0.245 (0.245)      Data 0.214 (0.214)      Loss
0.1353 (0.1353)    Prec 96.094% (96.094%)
Epoch: [12][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1742 (0.1835)    Prec 94.531% (93.928%)
```

```
Epoch: [12][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1853 (0.1856)    Prec 91.406% (93.874%)
Epoch: [12][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1617 (0.1877)    Prec 92.969% (93.706%)
Test: [0/79]    Time 0.176 (0.176)      Loss 0.3406 (0.3406)    Prec 92.188%
(92.188%)
 * Prec 85.260%
Epoch: [13][0/391]      Time 0.262 (0.262)      Data 0.223 (0.223)      Loss
0.0731 (0.0731)    Prec 97.656% (97.656%)
Epoch: [13][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.2055 (0.1805)    Prec 92.188% (93.936%)
Epoch: [13][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.2580 (0.1831)    Prec 92.188% (93.855%)
Epoch: [13][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1848 (0.1839)    Prec 92.969% (93.872%)
Test: [0/79]    Time 0.170 (0.170)      Loss 0.3395 (0.3395)    Prec 88.281%
(88.281%)
 * Prec 87.310%
Epoch: [14][0/391]      Time 0.263 (0.263)      Data 0.230 (0.230)      Loss
0.1882 (0.1882)    Prec 94.531% (94.531%)
Epoch: [14][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.2131 (0.1645)    Prec 92.188% (94.624%)
Epoch: [14][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1301 (0.1700)    Prec 96.875% (94.364%)
Epoch: [14][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.2876 (0.1750)    Prec 91.406% (94.186%)
Test: [0/79]    Time 0.139 (0.139)      Loss 0.3134 (0.3134)    Prec 89.844%
(89.844%)
 * Prec 87.540%
Epoch: [15][0/391]      Time 0.255 (0.255)      Data 0.222 (0.222)      Loss
0.1281 (0.1281)    Prec 96.094% (96.094%)
Epoch: [15][100/391]    Time 0.049 (0.051)      Data 0.001 (0.004)      Loss
0.1074 (0.1605)    Prec 95.312% (94.616%)
Epoch: [15][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1889 (0.1690)    Prec 92.188% (94.294%)
Epoch: [15][300/391]    Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.1257 (0.1672)    Prec 92.969% (94.360%)
Test: [0/79]    Time 0.242 (0.242)      Loss 0.3638 (0.3638)    Prec 92.188%
(92.188%)
 * Prec 87.450%
Epoch: [16][0/391]      Time 0.277 (0.277)      Data 0.245 (0.245)      Loss
0.2048 (0.2048)    Prec 92.969% (92.969%)
Epoch: [16][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1547 (0.1548)    Prec 94.531% (94.670%)
Epoch: [16][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1781 (0.1563)    Prec 94.531% (94.632%)
Epoch: [16][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.2233 (0.1678)    Prec 95.312% (94.274%)
```

```
Test: [0/79]    Time 0.190 (0.190)      Loss 0.3743 (0.3743)    Prec 87.500%
(87.500%)
 * Prec 88.310%
Epoch: [17][0/391]      Time 0.238 (0.238)      Data 0.202 (0.202)      Loss
0.1709 (0.1709)    Prec 92.188% (92.188%)
Epoch: [17][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1169 (0.1591)    Prec 96.875% (94.678%)
Epoch: [17][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.2158 (0.1612)    Prec 90.625% (94.481%)
Epoch: [17][300/391]    Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0796 (0.1668)    Prec 97.656% (94.285%)
Test: [0/79]    Time 0.154 (0.154)      Loss 0.4126 (0.4126)    Prec 88.281%
(88.281%)
 * Prec 86.450%
Epoch: [18][0/391]      Time 0.244 (0.244)      Data 0.208 (0.208)      Loss
0.1334 (0.1334)    Prec 94.531% (94.531%)
Epoch: [18][100/391]    Time 0.050 (0.051)      Data 0.001 (0.004)      Loss
0.1468 (0.1557)    Prec 96.875% (94.609%)
Epoch: [18][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1371 (0.1551)    Prec 96.094% (94.671%)
Epoch: [18][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1795 (0.1583)    Prec 92.969% (94.536%)
Test: [0/79]    Time 0.138 (0.138)      Loss 0.4061 (0.4061)    Prec 89.062%
(89.062%)
 * Prec 87.520%
Epoch: [19][0/391]      Time 0.261 (0.261)      Data 0.230 (0.230)      Loss
0.1747 (0.1747)    Prec 96.094% (96.094%)
Epoch: [19][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1032 (0.1513)    Prec 95.312% (94.663%)
Epoch: [19][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.2005 (0.1515)    Prec 93.750% (94.718%)
Epoch: [19][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.2016 (0.1575)    Prec 92.969% (94.555%)
Test: [0/79]    Time 0.204 (0.204)      Loss 0.3429 (0.3429)    Prec 86.719%
(86.719%)
 * Prec 86.750%
Epoch: [20][0/391]      Time 0.285 (0.285)      Data 0.248 (0.248)      Loss
0.2514 (0.2514)    Prec 93.750% (93.750%)
Epoch: [20][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1091 (0.1473)    Prec 95.312% (94.926%)
Epoch: [20][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1552 (0.1509)    Prec 95.312% (94.850%)
Epoch: [20][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1808 (0.1551)    Prec 96.875% (94.614%)
Test: [0/79]    Time 0.140 (0.140)      Loss 0.3380 (0.3380)    Prec 90.625%
(90.625%)
 * Prec 88.450%
Epoch: [21][0/391]      Time 0.355 (0.355)      Data 0.315 (0.315)      Loss
```

```
0.1046 (0.1046)    Prec 94.531% (94.531%)
Epoch: [21][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.2189 (0.1345)    Prec 92.188% (95.312%)
Epoch: [21][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1649 (0.1415)    Prec 94.531% (95.138%)
Epoch: [21][300/391]    Time 0.053 (0.050)      Data 0.001 (0.002)      Loss
0.0842 (0.1432)    Prec 96.875% (95.081%)
Test: [0/79]    Time 0.144 (0.144)      Loss 0.3155 (0.3155)    Prec 89.062%
(89.062%)
 * Prec 87.120%
Epoch: [22][0/391]      Time 0.269 (0.269)      Data 0.233 (0.233)      Loss
0.1690 (0.1690)    Prec 95.312% (95.312%)
Epoch: [22][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1850 (0.1433)    Prec 92.188% (95.096%)
Epoch: [22][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1410 (0.1445)    Prec 95.312% (95.056%)
Epoch: [22][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1625 (0.1428)    Prec 94.531% (95.110%)
Test: [0/79]    Time 0.255 (0.255)      Loss 0.3004 (0.3004)    Prec 90.625%
(90.625%)
 * Prec 88.440%
Epoch: [23][0/391]      Time 0.353 (0.353)      Data 0.319 (0.319)      Loss
0.1794 (0.1794)    Prec 92.969% (92.969%)
Epoch: [23][100/391]    Time 0.052 (0.052)      Data 0.001 (0.004)      Loss
0.0470 (0.1231)    Prec 98.438% (95.769%)
Epoch: [23][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.2840 (0.1354)    Prec 89.844% (95.324%)
Epoch: [23][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1321 (0.1417)    Prec 96.094% (95.105%)
Test: [0/79]    Time 0.200 (0.200)      Loss 0.2258 (0.2258)    Prec 92.188%
(92.188%)
 * Prec 88.170%
Epoch: [24][0/391]      Time 0.248 (0.248)      Data 0.208 (0.208)      Loss
0.0492 (0.0492)    Prec 99.219% (99.219%)
Epoch: [24][100/391]    Time 0.050 (0.051)      Data 0.001 (0.004)      Loss
0.1545 (0.1363)    Prec 94.531% (95.258%)
Epoch: [24][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1562 (0.1371)    Prec 96.094% (95.289%)
Epoch: [24][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1281 (0.1418)    Prec 95.312% (95.157%)
Test: [0/79]    Time 0.149 (0.149)      Loss 0.2906 (0.2906)    Prec 90.625%
(90.625%)
 * Prec 88.720%
Epoch: [25][0/391]      Time 0.253 (0.253)      Data 0.220 (0.220)      Loss
0.1495 (0.1495)    Prec 96.875% (96.875%)
Epoch: [25][100/391]    Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.1364 (0.1327)    Prec 93.750% (95.452%)
Epoch: [25][200/391]    Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
```

```
0.1720 (0.1357)    Prec 92.969% (95.254%)
Epoch: [25][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0811 (0.1381)    Prec 96.875% (95.206%)
Test: [0/79]    Time 0.178 (0.178)      Loss 0.2227 (0.2227)    Prec 92.188%
(92.188%)
 * Prec 88.670%
Epoch: [26][0/391]      Time 0.304 (0.304)      Data 0.269 (0.269)      Loss
0.1449 (0.1449)    Prec 94.531% (94.531%)
Epoch: [26][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0931 (0.1242)    Prec 97.656% (95.854%)
Epoch: [26][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1146 (0.1258)    Prec 96.094% (95.674%)
Epoch: [26][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0887 (0.1285)    Prec 96.875% (95.567%)
Test: [0/79]    Time 0.254 (0.254)      Loss 0.3198 (0.3198)    Prec 92.969%
(92.969%)
 * Prec 87.610%
Epoch: [27][0/391]      Time 0.230 (0.230)      Data 0.197 (0.197)      Loss
0.1452 (0.1452)    Prec 94.531% (94.531%)
Epoch: [27][100/391]    Time 0.051 (0.051)      Data 0.001 (0.003)      Loss
0.1763 (0.1319)    Prec 93.750% (95.614%)
Epoch: [27][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1998 (0.1327)    Prec 92.188% (95.542%)
Epoch: [27][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1837 (0.1349)    Prec 93.750% (95.442%)
Test: [0/79]    Time 0.207 (0.207)      Loss 0.3879 (0.3879)    Prec 87.500%
(87.500%)
 * Prec 87.230%
Epoch: [28][0/391]      Time 0.283 (0.283)      Data 0.252 (0.252)      Loss
0.1470 (0.1470)    Prec 94.531% (94.531%)
Epoch: [28][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0810 (0.1290)    Prec 97.656% (95.715%)
Epoch: [28][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.1453 (0.1258)    Prec 94.531% (95.728%)
Epoch: [28][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1424 (0.1262)    Prec 94.531% (95.650%)
Test: [0/79]    Time 0.141 (0.141)      Loss 0.3589 (0.3589)    Prec 90.625%
(90.625%)
 * Prec 88.620%
Epoch: [29][0/391]      Time 0.255 (0.255)      Data 0.217 (0.217)      Loss
0.1394 (0.1394)    Prec 96.094% (96.094%)
Epoch: [29][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0911 (0.1168)    Prec 96.094% (95.916%)
Epoch: [29][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0940 (0.1224)    Prec 96.875% (95.740%)
Epoch: [29][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.2143 (0.1284)    Prec 94.531% (95.608%)
Test: [0/79]    Time 0.150 (0.150)      Loss 0.3097 (0.3097)    Prec 93.750%
```

```
(93.750%)
 * Prec 88.610%
Epoch: [30][0/391]      Time 0.251 (0.251)      Data 0.211 (0.211)      Loss
0.1547 (0.1547)    Prec 96.094% (96.094%)
Epoch: [30][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0888 (0.1168)    Prec 96.094% (95.800%)
Epoch: [30][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1417 (0.1214)    Prec 94.531% (95.752%)
Epoch: [30][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1182 (0.1239)    Prec 96.875% (95.772%)
Test: [0/79]    Time 0.179 (0.179)      Loss 0.3806 (0.3806)    Prec 89.844%
(89.844%)
 * Prec 88.210%
Epoch: [31][0/391]      Time 0.316 (0.316)      Data 0.277 (0.277)      Loss
0.0867 (0.0867)    Prec 97.656% (97.656%)
Epoch: [31][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1262 (0.1150)    Prec 95.312% (96.202%)
Epoch: [31][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1401 (0.1143)    Prec 95.312% (96.152%)
Epoch: [31][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1415 (0.1212)    Prec 96.094% (95.909%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.3501 (0.3501)    Prec 88.281%
(88.281%)
 * Prec 87.840%
Epoch: [32][0/391]      Time 0.256 (0.256)      Data 0.223 (0.223)      Loss
0.0456 (0.0456)    Prec 98.438% (98.438%)
Epoch: [32][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0855 (0.1160)    Prec 96.875% (96.009%)
Epoch: [32][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1426 (0.1118)    Prec 95.312% (96.117%)
Epoch: [32][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0750 (0.1151)    Prec 96.875% (96.037%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.4622 (0.4622)    Prec 87.500%
(87.500%)
 * Prec 87.810%
Epoch: [33][0/391]      Time 0.250 (0.250)      Data 0.216 (0.216)      Loss
0.1337 (0.1337)    Prec 95.312% (95.312%)
Epoch: [33][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1091 (0.1230)    Prec 94.531% (95.854%)
Epoch: [33][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0887 (0.1235)    Prec 98.438% (95.899%)
Epoch: [33][300/391]    Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0667 (0.1234)    Prec 98.438% (95.928%)
Test: [0/79]    Time 0.261 (0.261)      Loss 0.2165 (0.2165)    Prec 90.625%
(90.625%)
 * Prec 89.370%
Epoch: [34][0/391]      Time 0.276 (0.276)      Data 0.236 (0.236)      Loss
0.0940 (0.0940)    Prec 96.094% (96.094%)
```

```
Epoch: [34][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0932 (0.1092)    Prec 97.656% (96.210%)
Epoch: [34][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0979 (0.1131)    Prec 96.875% (96.094%)
Epoch: [34][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1284 (0.1160)    Prec 95.312% (95.998%)
Test: [0/79]    Time 0.245 (0.245)      Loss 0.2611 (0.2611)    Prec 92.969%
(92.969%)
 * Prec 88.570%
Epoch: [35][0/391]      Time 0.333 (0.333)      Data 0.293 (0.293)      Loss
0.1450 (0.1450)    Prec 94.531% (94.531%)
Epoch: [35][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0408 (0.1035)    Prec 99.219% (96.434%)
Epoch: [35][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.2060 (0.1041)    Prec 93.750% (96.374%)
Epoch: [35][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0798 (0.1091)    Prec 96.875% (96.234%)
Test: [0/79]    Time 0.115 (0.115)      Loss 0.1948 (0.1948)    Prec 92.969%
(92.969%)
 * Prec 88.450%
Epoch: [36][0/391]      Time 0.412 (0.412)      Data 0.381 (0.381)      Loss
0.1083 (0.1083)    Prec 94.531% (94.531%)
Epoch: [36][100/391]    Time 0.049 (0.053)      Data 0.001 (0.005)      Loss
0.0566 (0.1140)    Prec 97.656% (95.939%)
Epoch: [36][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0679 (0.1118)    Prec 98.438% (96.102%)
Epoch: [36][300/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1537 (0.1128)    Prec 96.094% (96.060%)
Test: [0/79]    Time 0.154 (0.154)      Loss 0.3257 (0.3257)    Prec 89.844%
(89.844%)
 * Prec 88.500%
Epoch: [37][0/391]      Time 0.252 (0.252)      Data 0.221 (0.221)      Loss
0.1034 (0.1034)    Prec 96.875% (96.875%)
Epoch: [37][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1558 (0.1007)    Prec 93.750% (96.519%)
Epoch: [37][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1399 (0.1022)    Prec 95.312% (96.533%)
Epoch: [37][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1698 (0.1074)    Prec 92.969% (96.340%)
Test: [0/79]    Time 0.161 (0.161)      Loss 0.2443 (0.2443)    Prec 90.625%
(90.625%)
 * Prec 88.150%
Epoch: [38][0/391]      Time 0.251 (0.251)      Data 0.212 (0.212)      Loss
0.0772 (0.0772)    Prec 96.875% (96.875%)
Epoch: [38][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1290 (0.1064)    Prec 96.094% (96.310%)
Epoch: [38][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1159 (0.1087)    Prec 95.312% (96.300%)
```

```
Epoch: [38][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1460 (0.1071)    Prec 92.188% (96.296%)
Test: [0/79]    Time 0.159 (0.159)      Loss 0.2701 (0.2701)    Prec 91.406%
(91.406%)
 * Prec 88.750%
Epoch: [39][0/391]      Time 0.247 (0.247)      Data 0.213 (0.213)      Loss
0.1194 (0.1194)    Prec 95.312% (95.312%)
Epoch: [39][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1320 (0.1081)    Prec 93.750% (96.341%)
Epoch: [39][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1198 (0.1065)    Prec 97.656% (96.296%)
Epoch: [39][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1666 (0.1057)    Prec 94.531% (96.356%)
Test: [0/79]    Time 0.162 (0.162)      Loss 0.3671 (0.3671)    Prec 89.844%
(89.844%)
 * Prec 88.290%
Epoch: [40][0/391]      Time 0.283 (0.283)      Data 0.213 (0.213)      Loss
0.1275 (0.1275)    Prec 96.094% (96.094%)
Epoch: [40][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0662 (0.1024)    Prec 96.875% (96.481%)
Epoch: [40][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1365 (0.1042)    Prec 94.531% (96.374%)
Epoch: [40][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1299 (0.1061)    Prec 92.969% (96.351%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3850 (0.3850)    Prec 92.188%
(92.188%)
 * Prec 88.390%
Epoch: [41][0/391]      Time 0.263 (0.263)      Data 0.231 (0.231)      Loss
0.0635 (0.0635)    Prec 97.656% (97.656%)
Epoch: [41][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0641 (0.0919)    Prec 98.438% (96.767%)
Epoch: [41][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0827 (0.0975)    Prec 96.094% (96.552%)
Epoch: [41][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0477 (0.1011)    Prec 98.438% (96.410%)
Test: [0/79]    Time 0.157 (0.157)      Loss 0.2867 (0.2867)    Prec 90.625%
(90.625%)
 * Prec 88.550%
Epoch: [42][0/391]      Time 0.259 (0.259)      Data 0.228 (0.228)      Loss
0.0447 (0.0447)    Prec 99.219% (99.219%)
Epoch: [42][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.1154 (0.1024)    Prec 96.875% (96.682%)
Epoch: [42][200/391]    Time 0.050 (0.051)      Data 0.002 (0.002)      Loss
0.0909 (0.1057)    Prec 96.875% (96.494%)
Epoch: [42][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1110 (0.1058)    Prec 96.875% (96.506%)
Test: [0/79]    Time 0.162 (0.162)      Loss 0.2894 (0.2894)    Prec 89.844%
(89.844%)
```

```
 * Prec 88.900%
Epoch: [43][0/391]      Time 0.237 (0.237)      Data 0.200 (0.200)      Loss
0.1155 (0.1155)    Prec 97.656% (97.656%)
Epoch: [43][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0768 (0.0927)    Prec 96.875% (96.844%)
Epoch: [43][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0747 (0.0973)    Prec 98.438% (96.603%)
Epoch: [43][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1149 (0.1016)    Prec 95.312% (96.457%)
Test: [0/79]    Time 0.214 (0.214)      Loss 0.2846 (0.2846)    Prec 92.188%
(92.188%)
 * Prec 88.640%
Epoch: [44][0/391]      Time 0.391 (0.391)      Data 0.357 (0.357)      Loss
0.1244 (0.1244)    Prec 96.094% (96.094%)
Epoch: [44][100/391]    Time 0.049 (0.053)      Data 0.001 (0.005)      Loss
0.1352 (0.0944)    Prec 95.312% (96.666%)
Epoch: [44][200/391]    Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.1864 (0.0960)    Prec 93.750% (96.638%)
Epoch: [44][300/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.1301 (0.1004)    Prec 95.312% (96.483%)
Test: [0/79]    Time 0.243 (0.243)      Loss 0.1900 (0.1900)    Prec 94.531%
(94.531%)
 * Prec 89.110%
Epoch: [45][0/391]      Time 0.229 (0.229)      Data 0.188 (0.188)      Loss
0.0324 (0.0324)    Prec 98.438% (98.438%)
Epoch: [45][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0521 (0.0938)    Prec 98.438% (96.844%)
Epoch: [45][200/391]    Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.1050 (0.1005)    Prec 96.094% (96.587%)
Epoch: [45][300/391]    Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.1043 (0.0970)    Prec 93.750% (96.699%)
Test: [0/79]    Time 0.145 (0.145)      Loss 0.2768 (0.2768)    Prec 92.969%
(92.969%)
 * Prec 88.920%
Epoch: [46][0/391]      Time 0.259 (0.259)      Data 0.219 (0.219)      Loss
0.0159 (0.0159)    Prec 100.000% (100.000%)
Epoch: [46][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0533 (0.0851)    Prec 98.438% (96.991%)
Epoch: [46][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1060 (0.0928)    Prec 96.094% (96.762%)
Epoch: [46][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0760 (0.0961)    Prec 96.094% (96.574%)
Test: [0/79]    Time 0.219 (0.219)      Loss 0.3591 (0.3591)    Prec 87.500%
(87.500%)
 * Prec 88.760%
Epoch: [47][0/391]      Time 0.300 (0.300)      Data 0.259 (0.259)      Loss
0.0655 (0.0655)    Prec 96.875% (96.875%)
Epoch: [47][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
```

```
0.1113 (0.0881)    Prec 95.312% (97.037%)
Epoch: [47][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0617 (0.0884)    Prec 97.656% (96.992%)
Epoch: [47][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1271 (0.0932)    Prec 92.969% (96.802%)
Test: [0/79]    Time 0.161 (0.161)      Loss 0.3476 (0.3476)    Prec 92.188%
(92.188%)
 * Prec 88.030%
Epoch: [48][0/391]      Time 0.246 (0.246)      Data 0.213 (0.213)      Loss
0.0906 (0.0906)    Prec 96.094% (96.094%)
Epoch: [48][100/391]    Time 0.046 (0.051)      Data 0.001 (0.003)      Loss
0.1170 (0.0968)    Prec 94.531% (96.844%)
Epoch: [48][200/391]    Time 0.046 (0.050)      Data 0.001 (0.002)      Loss
0.2518 (0.0990)    Prec 92.188% (96.618%)
Epoch: [48][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1573 (0.0978)    Prec 94.531% (96.670%)
Test: [0/79]    Time 0.148 (0.148)      Loss 0.3482 (0.3482)    Prec 90.625%
(90.625%)
 * Prec 89.250%
Epoch: [49][0/391]      Time 0.204 (0.204)      Data 0.129 (0.129)      Loss
0.0552 (0.0552)    Prec 98.438% (98.438%)
Epoch: [49][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0577 (0.0846)    Prec 98.438% (97.115%)
Epoch: [49][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0850 (0.0894)    Prec 96.094% (96.848%)
Epoch: [49][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0905 (0.0926)    Prec 97.656% (96.714%)
Test: [0/79]    Time 0.221 (0.221)      Loss 0.3233 (0.3233)    Prec 92.188%
(92.188%)
 * Prec 88.790%
Epoch: [50][0/391]      Time 0.296 (0.296)      Data 0.265 (0.265)      Loss
0.0855 (0.0855)    Prec 97.656% (97.656%)
Epoch: [50][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0591 (0.0846)    Prec 97.656% (97.068%)
Epoch: [50][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0372 (0.0842)    Prec 98.438% (97.058%)
Epoch: [50][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0936 (0.0865)    Prec 96.875% (97.015%)
Test: [0/79]    Time 0.154 (0.154)      Loss 0.3101 (0.3101)    Prec 89.062%
(89.062%)
 * Prec 89.140%
Epoch: [51][0/391]      Time 0.344 (0.344)      Data 0.307 (0.307)      Loss
0.0628 (0.0628)    Prec 97.656% (97.656%)
Epoch: [51][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0381 (0.0883)    Prec 98.438% (96.852%)
Epoch: [51][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0671 (0.0900)    Prec 96.094% (96.828%)
Epoch: [51][300/391]    Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
```

```
0.0777 (0.0908)    Prec 98.438% (96.872%)
Test: [0/79]    Time 0.158 (0.158)      Loss 0.3118 (0.3118)    Prec 92.188%
(92.188%)
 * Prec 89.220%
Epoch: [52][0/391]      Time 0.319 (0.319)      Data 0.288 (0.288)      Loss
0.0293 (0.0293)    Prec 100.000% (100.000%)
Epoch: [52][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1501 (0.0883)    Prec 95.312% (97.092%)
Epoch: [52][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1692 (0.0916)    Prec 95.312% (96.809%)
Epoch: [52][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0307 (0.0958)    Prec 99.219% (96.719%)
Test: [0/79]    Time 0.171 (0.171)      Loss 0.3249 (0.3249)    Prec 92.188%
(92.188%)
 * Prec 89.340%
Epoch: [53][0/391]      Time 0.255 (0.255)      Data 0.216 (0.216)      Loss
0.0284 (0.0284)    Prec 100.000% (100.000%)
Epoch: [53][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0593 (0.0847)    Prec 97.656% (97.177%)
Epoch: [53][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0831 (0.0850)    Prec 96.875% (97.248%)
Epoch: [53][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1267 (0.0869)    Prec 96.094% (97.111%)
Test: [0/79]    Time 0.206 (0.206)      Loss 0.3053 (0.3053)    Prec 90.625%
(90.625%)
 * Prec 88.780%
Epoch: [54][0/391]      Time 0.275 (0.275)      Data 0.242 (0.242)      Loss
0.1436 (0.1436)    Prec 95.312% (95.312%)
Epoch: [54][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1306 (0.0846)    Prec 95.312% (97.215%)
Epoch: [54][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1025 (0.0872)    Prec 96.875% (97.023%)
Epoch: [54][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0736 (0.0896)    Prec 96.094% (96.924%)
Test: [0/79]    Time 0.181 (0.181)      Loss 0.3578 (0.3578)    Prec 90.625%
(90.625%)
 * Prec 88.590%
Epoch: [55][0/391]      Time 0.253 (0.253)      Data 0.220 (0.220)      Loss
0.0246 (0.0246)    Prec 100.000% (100.000%)
Epoch: [55][100/391]    Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.0293 (0.0848)    Prec 99.219% (97.115%)
Epoch: [55][200/391]    Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0647 (0.0866)    Prec 97.656% (97.034%)
Epoch: [55][300/391]    Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.1003 (0.0895)    Prec 96.094% (96.901%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3286 (0.3286)    Prec 89.844%
(89.844%)
 * Prec 89.160%
```

```
Epoch: [56][0/391]      Time 0.254 (0.254)      Data 0.218 (0.218)      Loss
0.0528 (0.0528)    Prec 98.438% (98.438%)
Epoch: [56][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0345 (0.0852)    Prec 98.438% (97.161%)
Epoch: [56][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0737 (0.0864)    Prec 96.875% (97.100%)
Epoch: [56][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0798 (0.0868)    Prec 96.875% (97.036%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.4490 (0.4490)    Prec 87.500%
(87.500%)
 * Prec 88.750%
Epoch: [57][0/391]      Time 0.256 (0.256)      Data 0.221 (0.221)      Loss
0.1486 (0.1486)    Prec 94.531% (94.531%)
Epoch: [57][100/391]    Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0546 (0.0822)    Prec 98.438% (97.192%)
Epoch: [57][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0961 (0.0874)    Prec 97.656% (97.023%)
Epoch: [57][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0504 (0.0910)    Prec 97.656% (96.875%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3455 (0.3455)    Prec 92.969%
(92.969%)
 * Prec 88.920%
Epoch: [58][0/391]      Time 0.260 (0.260)      Data 0.220 (0.220)      Loss
0.1354 (0.1354)    Prec 96.875% (96.875%)
Epoch: [58][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0667 (0.0772)    Prec 96.094% (97.362%)
Epoch: [58][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0503 (0.0780)    Prec 99.219% (97.306%)
Epoch: [58][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0865 (0.0783)    Prec 97.656% (97.389%)
Test: [0/79]    Time 0.150 (0.150)      Loss 0.3389 (0.3389)    Prec 92.969%
(92.969%)
 * Prec 88.030%
Epoch: [59][0/391]      Time 0.260 (0.260)      Data 0.224 (0.224)      Loss
0.0695 (0.0695)    Prec 96.875% (96.875%)
Epoch: [59][100/391]    Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.1049 (0.0843)    Prec 97.656% (97.177%)
Epoch: [59][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1645 (0.0872)    Prec 95.312% (97.054%)
Epoch: [59][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1656 (0.0866)    Prec 95.312% (97.070%)
Test: [0/79]    Time 0.169 (0.169)      Loss 0.3535 (0.3535)    Prec 90.625%
(90.625%)
 * Prec 89.110%
Epoch: [60][0/391]      Time 0.314 (0.314)      Data 0.281 (0.281)      Loss
0.0472 (0.0472)    Prec 97.656% (97.656%)
Epoch: [60][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1381 (0.0830)    Prec 94.531% (97.006%)
```

```
Epoch: [60][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0792 (0.0833)    Prec 98.438% (97.065%)
Epoch: [60][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1627 (0.0849)    Prec 93.750% (97.026%)
Test: [0/79]    Time 0.244 (0.244)      Loss 0.3024 (0.3024)    Prec 89.062%
(89.062%)
 * Prec 89.160%
Epoch: [61][0/391]      Time 0.267 (0.267)      Data 0.235 (0.235)      Loss
0.0777 (0.0777)    Prec 96.094% (96.094%)
Epoch: [61][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0400 (0.0836)    Prec 99.219% (97.107%)
Epoch: [61][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0749 (0.0811)    Prec 97.656% (97.178%)
Epoch: [61][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0669 (0.0847)    Prec 97.656% (97.067%)
Test: [0/79]    Time 0.242 (0.242)      Loss 0.3036 (0.3036)    Prec 89.844%
(89.844%)
 * Prec 88.940%
Epoch: [62][0/391]      Time 0.277 (0.277)      Data 0.238 (0.238)      Loss
0.0139 (0.0139)    Prec 100.000% (100.000%)
Epoch: [62][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0762 (0.0810)    Prec 96.875% (97.169%)
Epoch: [62][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0883 (0.0835)    Prec 95.312% (97.050%)
Epoch: [62][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0545 (0.0815)    Prec 98.438% (97.129%)
Test: [0/79]    Time 0.254 (0.254)      Loss 0.2600 (0.2600)    Prec 93.750%
(93.750%)
 * Prec 89.040%
Epoch: [63][0/391]      Time 0.264 (0.264)      Data 0.224 (0.224)      Loss
0.0732 (0.0732)    Prec 97.656% (97.656%)
Epoch: [63][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0431 (0.0802)    Prec 99.219% (97.347%)
Epoch: [63][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0859 (0.0758)    Prec 94.531% (97.435%)
Epoch: [63][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1105 (0.0780)    Prec 97.656% (97.355%)
Test: [0/79]    Time 0.260 (0.260)      Loss 0.4518 (0.4518)    Prec 86.719%
(86.719%)
 * Prec 88.270%
Epoch: [64][0/391]      Time 0.252 (0.252)      Data 0.216 (0.216)      Loss
0.0855 (0.0855)    Prec 96.094% (96.094%)
Epoch: [64][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0428 (0.0755)    Prec 98.438% (97.401%)
Epoch: [64][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0641 (0.0792)    Prec 97.656% (97.229%)
Epoch: [64][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0973 (0.0781)    Prec 96.875% (97.257%)
```

```
Test: [0/79]    Time 0.237 (0.237)      Loss 0.2775 (0.2775)    Prec 93.750%
(93.750%)
 * Prec 89.440%
Epoch: [65][0/391]      Time 0.257 (0.257)      Data 0.221 (0.221)      Loss
0.0730 (0.0730)    Prec 98.438% (98.438%)
Epoch: [65][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0549 (0.0793)    Prec 98.438% (97.293%)
Epoch: [65][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0649 (0.0786)    Prec 97.656% (97.244%)
Epoch: [65][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0097 (0.0807)    Prec 100.000% (97.184%)
Test: [0/79]    Time 0.151 (0.151)      Loss 0.3491 (0.3491)    Prec 89.844%
(89.844%)
 * Prec 88.240%
Epoch: [66][0/391]      Time 0.256 (0.256)      Data 0.224 (0.224)      Loss
0.0573 (0.0573)    Prec 97.656% (97.656%)
Epoch: [66][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0440 (0.0704)    Prec 98.438% (97.741%)
Epoch: [66][200/391]    Time 0.047 (0.050)      Data 0.001 (0.002)      Loss
0.0442 (0.0725)    Prec 98.438% (97.516%)
Epoch: [66][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0561 (0.0762)    Prec 99.219% (97.415%)
Test: [0/79]    Time 0.114 (0.114)      Loss 0.3839 (0.3839)    Prec 88.281%
(88.281%)
 * Prec 89.270%
Epoch: [67][0/391]      Time 0.301 (0.301)      Data 0.260 (0.260)      Loss
0.0447 (0.0447)    Prec 97.656% (97.656%)
Epoch: [67][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.2073 (0.0809)    Prec 95.312% (97.362%)
Epoch: [67][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0770 (0.0798)    Prec 98.438% (97.330%)
Epoch: [67][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1624 (0.0803)    Prec 96.094% (97.311%)
Test: [0/79]    Time 0.111 (0.111)      Loss 0.3109 (0.3109)    Prec 92.188%
(92.188%)
 * Prec 89.250%
Epoch: [68][0/391]      Time 0.304 (0.304)      Data 0.269 (0.269)      Loss
0.0955 (0.0955)    Prec 96.875% (96.875%)
Epoch: [68][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0414 (0.0756)    Prec 97.656% (97.285%)
Epoch: [68][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0763 (0.0753)    Prec 97.656% (97.380%)
Epoch: [68][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0407 (0.0761)    Prec 99.219% (97.332%)
Test: [0/79]    Time 0.207 (0.207)      Loss 0.4196 (0.4196)    Prec 89.062%
(89.062%)
 * Prec 89.380%
Epoch: [69][0/391]      Time 0.288 (0.288)      Data 0.249 (0.249)      Loss
```

```
0.0237 (0.0237)    Prec 100.000% (100.000%)
Epoch: [69][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0794 (0.0689)    Prec 96.875% (97.734%)
Epoch: [69][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1141 (0.0753)    Prec 97.656% (97.446%)
Epoch: [69][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1442 (0.0769)    Prec 95.312% (97.399%)
Test: [0/79]    Time 0.143 (0.143)      Loss 0.2818 (0.2818)    Prec 91.406%
(91.406%)
 * Prec 88.320%
Epoch: [70][0/391]      Time 0.270 (0.270)      Data 0.230 (0.230)      Loss
0.0360 (0.0360)    Prec 99.219% (99.219%)
Epoch: [70][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0663 (0.0750)    Prec 97.656% (97.254%)
Epoch: [70][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0853 (0.0743)    Prec 97.656% (97.349%)
Epoch: [70][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0448 (0.0782)    Prec 98.438% (97.225%)
Test: [0/79]    Time 0.252 (0.252)      Loss 0.4353 (0.4353)    Prec 87.500%
(87.500%)
 * Prec 87.390%
Epoch: [71][0/391]      Time 0.262 (0.262)      Data 0.223 (0.223)      Loss
0.0489 (0.0489)    Prec 98.438% (98.438%)
Epoch: [71][100/391]    Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0707 (0.0795)    Prec 97.656% (97.200%)
Epoch: [71][200/391]    Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0661 (0.0749)    Prec 98.438% (97.349%)
Epoch: [71][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0323 (0.0753)    Prec 100.000% (97.392%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3398 (0.3398)    Prec 92.188%
(92.188%)
 * Prec 89.090%
Epoch: [72][0/391]      Time 0.251 (0.251)      Data 0.214 (0.214)      Loss
0.1190 (0.1190)    Prec 95.312% (95.312%)
Epoch: [72][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0623 (0.0659)    Prec 97.656% (97.749%)
Epoch: [72][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0294 (0.0724)    Prec 99.219% (97.547%)
Epoch: [72][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1006 (0.0753)    Prec 96.875% (97.402%)
Test: [0/79]    Time 0.147 (0.147)      Loss 0.2789 (0.2789)    Prec 90.625%
(90.625%)
 * Prec 88.210%
Epoch: [73][0/391]      Time 0.253 (0.253)      Data 0.221 (0.221)      Loss
0.1198 (0.1198)    Prec 97.656% (97.656%)
Epoch: [73][100/391]    Time 0.048 (0.052)      Data 0.001 (0.003)      Loss
0.0757 (0.0729)    Prec 96.875% (97.494%)
Epoch: [73][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.0412 (0.0742)    Prec 98.438% (97.458%)
Epoch: [73][300/391]    Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0838 (0.0741)    Prec 96.094% (97.384%)
Test: [0/79]    Time 0.158 (0.158)      Loss 0.3369 (0.3369)    Prec 91.406%
(91.406%)
 * Prec 88.770%
Epoch: [74][0/391]      Time 0.252 (0.252)      Data 0.213 (0.213)      Loss
0.0419 (0.0419)    Prec 98.438% (98.438%)
Epoch: [74][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1036 (0.0676)    Prec 95.312% (97.649%)
Epoch: [74][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0744 (0.0710)    Prec 97.656% (97.442%)
Epoch: [74][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0375 (0.0728)    Prec 99.219% (97.373%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.3072 (0.3072)    Prec 92.188%
(92.188%)
 * Prec 88.890%
Epoch: [75][0/391]      Time 0.263 (0.263)      Data 0.226 (0.226)      Loss
0.0560 (0.0560)    Prec 96.875% (96.875%)
Epoch: [75][100/391]    Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.1683 (0.0703)    Prec 95.312% (97.540%)
Epoch: [75][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0212 (0.0706)    Prec 100.000% (97.528%)
Epoch: [75][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0716 (0.0712)    Prec 97.656% (97.475%)
Test: [0/79]    Time 0.145 (0.145)      Loss 0.3909 (0.3909)    Prec 89.062%
(89.062%)
 * Prec 88.740%
Epoch: [76][0/391]      Time 0.299 (0.299)      Data 0.261 (0.261)      Loss
0.1436 (0.1436)    Prec 95.312% (95.312%)
Epoch: [76][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0434 (0.0723)    Prec 99.219% (97.478%)
Epoch: [76][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1686 (0.0712)    Prec 93.750% (97.520%)
Epoch: [76][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0322 (0.0726)    Prec 98.438% (97.475%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.2753 (0.2753)    Prec 92.969%
(92.969%)
 * Prec 89.350%
Epoch: [77][0/391]      Time 0.318 (0.318)      Data 0.279 (0.279)      Loss
0.0792 (0.0792)    Prec 97.656% (97.656%)
Epoch: [77][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0195 (0.0680)    Prec 100.000% (97.618%)
Epoch: [77][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1129 (0.0702)    Prec 96.875% (97.567%)
Epoch: [77][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0784 (0.0694)    Prec 98.438% (97.591%)
Test: [0/79]    Time 0.182 (0.182)      Loss 0.2923 (0.2923)    Prec 92.188%
```

```
(92.188%)
 * Prec 89.350%
Epoch: [78][0/391]      Time 0.248 (0.248)      Data 0.214 (0.214)      Loss
0.0826 (0.0826)    Prec 96.875% (96.875%)
Epoch: [78][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0428 (0.0695)    Prec 97.656% (97.679%)
Epoch: [78][200/391]    Time 0.046 (0.050)      Data 0.001 (0.002)      Loss
0.0828 (0.0690)    Prec 97.656% (97.703%)
Epoch: [78][300/391]    Time 0.046 (0.050)      Data 0.001 (0.002)      Loss
0.1159 (0.0695)    Prec 96.094% (97.669%)
Test: [0/79]    Time 0.240 (0.240)      Loss 0.2996 (0.2996)    Prec 92.188%
(92.188%)
 * Prec 89.290%
Epoch: [79][0/391]      Time 0.263 (0.263)      Data 0.231 (0.231)      Loss
0.0577 (0.0577)    Prec 96.875% (96.875%)
Epoch: [79][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1008 (0.0601)    Prec 95.312% (97.881%)
Epoch: [79][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0334 (0.0650)    Prec 99.219% (97.734%)
Epoch: [79][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0457 (0.0658)    Prec 98.438% (97.656%)
Test: [0/79]    Time 0.151 (0.151)      Loss 0.2573 (0.2573)    Prec 92.969%
(92.969%)
 * Prec 89.310%
Epoch: [80][0/391]      Time 0.257 (0.257)      Data 0.222 (0.222)      Loss
0.0525 (0.0525)    Prec 98.438% (98.438%)
Epoch: [80][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0827 (0.0682)    Prec 97.656% (97.602%)
Epoch: [80][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0481 (0.0673)    Prec 97.656% (97.656%)
Epoch: [80][300/391]    Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0834 (0.0705)    Prec 96.875% (97.571%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.2492 (0.2492)    Prec 92.188%
(92.188%)
 * Prec 89.660%
Epoch: [81][0/391]      Time 0.261 (0.261)      Data 0.225 (0.225)      Loss
0.0694 (0.0694)    Prec 97.656% (97.656%)
Epoch: [81][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0989 (0.0655)    Prec 96.875% (97.780%)
Epoch: [81][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0248 (0.0681)    Prec 100.000% (97.668%)
Epoch: [81][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0662 (0.0691)    Prec 96.875% (97.578%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.4511 (0.4511)    Prec 90.625%
(90.625%)
 * Prec 88.780%
Epoch: [82][0/391]      Time 0.263 (0.263)      Data 0.227 (0.227)      Loss
0.0888 (0.0888)    Prec 97.656% (97.656%)
```

```
Epoch: [82][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0281 (0.0618)    Prec 99.219% (97.826%)
Epoch: [82][200/391]    Time 0.051 (0.051)      Data 0.001 (0.002)      Loss
0.0931 (0.0677)    Prec 97.656% (97.629%)
Epoch: [82][300/391]    Time 0.044 (0.050)      Data 0.001 (0.002)      Loss
0.0724 (0.0695)    Prec 97.656% (97.529%)
Test: [0/79]    Time 0.141 (0.141)      Loss 0.3299 (0.3299)    Prec 92.969%
(92.969%)
 * Prec 89.210%
Epoch: [83][0/391]      Time 0.250 (0.250)      Data 0.217 (0.217)      Loss
0.0602 (0.0602)    Prec 99.219% (99.219%)
Epoch: [83][100/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1208 (0.0659)    Prec 97.656% (97.757%)
Epoch: [83][200/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0996 (0.0661)    Prec 96.094% (97.695%)
Epoch: [83][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0861 (0.0687)    Prec 97.656% (97.545%)
Test: [0/79]    Time 0.146 (0.146)      Loss 0.2463 (0.2463)    Prec 93.750%
(93.750%)
 * Prec 88.960%
Epoch: [84][0/391]      Time 0.258 (0.258)      Data 0.224 (0.224)      Loss
0.0569 (0.0569)    Prec 97.656% (97.656%)
Epoch: [84][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0783 (0.0637)    Prec 96.875% (97.765%)
Epoch: [84][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0282 (0.0634)    Prec 99.219% (97.761%)
Epoch: [84][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0972 (0.0656)    Prec 96.875% (97.721%)
Test: [0/79]    Time 0.242 (0.242)      Loss 0.3596 (0.3596)    Prec 91.406%
(91.406%)
 * Prec 89.200%
Epoch: [85][0/391]      Time 0.258 (0.258)      Data 0.225 (0.225)      Loss
0.0563 (0.0563)    Prec 97.656% (97.656%)
Epoch: [85][100/391]    Time 0.048 (0.052)      Data 0.001 (0.003)      Loss
0.0662 (0.0607)    Prec 99.219% (97.795%)
Epoch: [85][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0522 (0.0619)    Prec 98.438% (97.839%)
Epoch: [85][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0309 (0.0651)    Prec 98.438% (97.757%)
Test: [0/79]    Time 0.213 (0.213)      Loss 0.3419 (0.3419)    Prec 91.406%
(91.406%)
 * Prec 89.310%
Epoch: [86][0/391]      Time 0.398 (0.398)      Data 0.364 (0.364)      Loss
0.0765 (0.0765)    Prec 96.875% (96.875%)
Epoch: [86][100/391]    Time 0.050 (0.053)      Data 0.001 (0.005)      Loss
0.0963 (0.0571)    Prec 96.094% (97.981%)
Epoch: [86][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0705 (0.0597)    Prec 96.094% (97.905%)
```

```
Epoch: [86][300/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1314 (0.0613)    Prec 96.875% (97.841%)
Test: [0/79]    Time 0.224 (0.224)      Loss 0.3843 (0.3843)    Prec 86.719%
(86.719%)
 * Prec 89.150%
Epoch: [87][0/391]      Time 0.194 (0.194)      Data 0.158 (0.158)      Loss
0.0833 (0.0833)    Prec 98.438% (98.438%)
Epoch: [87][100/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0682 (0.0631)    Prec 97.656% (97.788%)
Epoch: [87][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0408 (0.0610)    Prec 98.438% (97.878%)
Epoch: [87][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0680 (0.0611)    Prec 96.875% (97.854%)
Test: [0/79]    Time 0.114 (0.114)      Loss 0.4221 (0.4221)    Prec 89.844%
(89.844%)
 * Prec 88.900%
Epoch: [88][0/391]      Time 0.301 (0.301)      Data 0.261 (0.261)      Loss
0.0597 (0.0597)    Prec 98.438% (98.438%)
Epoch: [88][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0228 (0.0640)    Prec 100.000% (97.896%)
Epoch: [88][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1066 (0.0631)    Prec 96.094% (97.905%)
Epoch: [88][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0204 (0.0657)    Prec 99.219% (97.786%)
Test: [0/79]    Time 0.163 (0.163)      Loss 0.3247 (0.3247)    Prec 92.188%
(92.188%)
 * Prec 89.010%
Epoch: [89][0/391]      Time 0.289 (0.289)      Data 0.256 (0.256)      Loss
0.0343 (0.0343)    Prec 99.219% (99.219%)
Epoch: [89][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0569 (0.0647)    Prec 97.656% (97.741%)
Epoch: [89][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.1072 (0.0657)    Prec 95.312% (97.722%)
Epoch: [89][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0246 (0.0650)    Prec 99.219% (97.739%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.4703 (0.4703)    Prec 88.281%
(88.281%)
 * Prec 89.040%
Epoch: [90][0/391]      Time 0.285 (0.285)      Data 0.253 (0.253)      Loss
0.0220 (0.0220)    Prec 100.000% (100.000%)
Epoch: [90][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1073 (0.0696)    Prec 97.656% (97.610%)
Epoch: [90][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0941 (0.0661)    Prec 96.875% (97.683%)
Epoch: [90][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0721 (0.0649)    Prec 96.875% (97.742%)
Test: [0/79]    Time 0.256 (0.256)      Loss 0.4429 (0.4429)    Prec 90.625%
(90.625%)
```

```
 * Prec 89.520%
Epoch: [91][0/391]      Time 0.268 (0.268)      Data 0.230 (0.230)      Loss
0.0515 (0.0515)    Prec 97.656% (97.656%)
Epoch: [91][100/391]    Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0858 (0.0659)    Prec 96.094% (97.649%)
Epoch: [91][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0394 (0.0624)    Prec 98.438% (97.804%)
Epoch: [91][300/391]    Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0594 (0.0601)    Prec 98.438% (97.885%)
Test: [0/79]    Time 0.166 (0.166)      Loss 0.3721 (0.3721)    Prec 91.406%
(91.406%)
 * Prec 89.350%
Epoch: [92][0/391]      Time 0.310 (0.310)      Data 0.238 (0.238)      Loss
0.1115 (0.1115)    Prec 96.875% (96.875%)
Epoch: [92][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1102 (0.0571)    Prec 95.312% (98.043%)
Epoch: [92][200/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.1071 (0.0595)    Prec 97.656% (97.952%)
Epoch: [92][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0429 (0.0607)    Prec 98.438% (97.861%)
Test: [0/79]    Time 0.207 (0.207)      Loss 0.3007 (0.3007)    Prec 92.969%
(92.969%)
 * Prec 89.020%
Epoch: [93][0/391]      Time 0.360 (0.360)      Data 0.325 (0.325)      Loss
0.0290 (0.0290)    Prec 99.219% (99.219%)
Epoch: [93][100/391]    Time 0.050 (0.053)      Data 0.001 (0.004)      Loss
0.1433 (0.0621)    Prec 95.312% (97.896%)
Epoch: [93][200/391]    Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0316 (0.0624)    Prec 98.438% (97.882%)
Epoch: [93][300/391]    Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0382 (0.0629)    Prec 98.438% (97.856%)
Test: [0/79]    Time 0.185 (0.185)      Loss 0.3650 (0.3650)    Prec 90.625%
(90.625%)
 * Prec 88.720%
Epoch: [94][0/391]      Time 0.252 (0.252)      Data 0.217 (0.217)      Loss
0.0389 (0.0389)    Prec 99.219% (99.219%)
Epoch: [94][100/391]    Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.1102 (0.0667)    Prec 96.094% (97.625%)
Epoch: [94][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0438 (0.0654)    Prec 99.219% (97.683%)
Epoch: [94][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0884 (0.0619)    Prec 97.656% (97.838%)
Test: [0/79]    Time 0.151 (0.151)      Loss 0.3537 (0.3537)    Prec 89.844%
(89.844%)
 * Prec 88.920%
Epoch: [95][0/391]      Time 0.264 (0.264)      Data 0.234 (0.234)      Loss
0.0267 (0.0267)    Prec 99.219% (99.219%)
Epoch: [95][100/391]    Time 0.050 (0.051)      Data 0.001 (0.004)      Loss
```

```
0.0593 (0.0551)    Prec 96.875% (98.144%)
Epoch: [95][200/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0184 (0.0569)    Prec 100.000% (98.138%)
Epoch: [95][300/391]    Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.1011 (0.0575)    Prec 96.875% (98.136%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.3419 (0.3419)    Prec 92.188%
(92.188%)
 * Prec 89.120%
Epoch: [96][0/391]      Time 0.259 (0.259)      Data 0.223 (0.223)      Loss
0.0844 (0.0844)    Prec 96.875% (96.875%)
Epoch: [96][100/391]    Time 0.049 (0.052)      Data 0.002 (0.003)      Loss
0.0349 (0.0568)    Prec 97.656% (98.020%)
Epoch: [96][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1043 (0.0593)    Prec 96.875% (97.921%)
Epoch: [96][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0391 (0.0590)    Prec 98.438% (97.947%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.3625 (0.3625)    Prec 90.625%
(90.625%)
 * Prec 88.900%
Epoch: [97][0/391]      Time 0.323 (0.323)      Data 0.290 (0.290)      Loss
0.0385 (0.0385)    Prec 99.219% (99.219%)
Epoch: [97][100/391]    Time 0.050 (0.052)      Data 0.002 (0.004)      Loss
0.0820 (0.0568)    Prec 96.875% (98.028%)
Epoch: [97][200/391]    Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0793 (0.0622)    Prec 98.438% (97.963%)
Epoch: [97][300/391]    Time 0.048 (0.050)      Data 0.002 (0.002)      Loss
0.0351 (0.0605)    Prec 99.219% (98.020%)
Test: [0/79]    Time 0.207 (0.207)      Loss 0.4100 (0.4100)    Prec 88.281%
(88.281%)
 * Prec 89.070%
Epoch: [98][0/391]      Time 0.317 (0.317)      Data 0.243 (0.243)      Loss
0.0332 (0.0332)    Prec 97.656% (97.656%)
Epoch: [98][100/391]    Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0967 (0.0534)    Prec 97.656% (98.352%)
Epoch: [98][200/391]    Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0286 (0.0530)    Prec 99.219% (98.270%)
Epoch: [98][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1058 (0.0544)    Prec 97.656% (98.191%)
Test: [0/79]    Time 0.237 (0.237)      Loss 0.3976 (0.3976)    Prec 90.625%
(90.625%)
 * Prec 89.040%
Epoch: [99][0/391]      Time 0.257 (0.257)      Data 0.223 (0.223)      Loss
0.1359 (0.1359)    Prec 96.875% (96.875%)
Epoch: [99][100/391]    Time 0.049 (0.051)      Data 0.002 (0.004)      Loss
0.0711 (0.0560)    Prec 96.875% (98.012%)
Epoch: [99][200/391]    Time 0.051 (0.050)      Data 0.001 (0.002)      Loss
0.0167 (0.0583)    Prec 99.219% (97.959%)
Epoch: [99][300/391]    Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.1770 (0.0565)    Prec 95.312% (98.066%)
Test: [0/79]    Time 0.214 (0.214)      Loss 0.3806 (0.3806)    Prec 89.062%
(89.062%)
 * Prec 88.760%
Epoch: [100][0/391]     Time 0.376 (0.376)      Data 0.336 (0.336)      Loss
0.0857 (0.0857)    Prec 96.094% (96.094%)
Epoch: [100][100/391]   Time 0.050 (0.053)      Data 0.001 (0.005)      Loss
0.0284 (0.0578)    Prec 99.219% (98.089%)
Epoch: [100][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0454 (0.0610)    Prec 98.438% (98.002%)
Epoch: [100][300/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0946 (0.0620)    Prec 97.656% (97.924%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.3691 (0.3691)    Prec 90.625%
(90.625%)
 * Prec 89.190%
Epoch: [101][0/391]     Time 0.260 (0.260)      Data 0.228 (0.228)      Loss
0.0216 (0.0216)    Prec 99.219% (99.219%)
Epoch: [101][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0478 (0.0505)    Prec 98.438% (98.314%)
Epoch: [101][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0591 (0.0541)    Prec 98.438% (98.255%)
Epoch: [101][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0635 (0.0558)    Prec 96.875% (98.147%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.3600 (0.3600)    Prec 89.062%
(89.062%)
 * Prec 89.400%
Epoch: [102][0/391]     Time 0.247 (0.247)      Data 0.214 (0.214)      Loss
0.0273 (0.0273)    Prec 98.438% (98.438%)
Epoch: [102][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0164 (0.0530)    Prec 99.219% (98.229%)
Epoch: [102][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0702 (0.0552)    Prec 96.875% (98.142%)
Epoch: [102][300/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0265 (0.0580)    Prec 99.219% (98.027%)
Test: [0/79]    Time 0.217 (0.217)      Loss 0.3552 (0.3552)    Prec 91.406%
(91.406%)
 * Prec 89.380%
Epoch: [103][0/391]     Time 0.353 (0.353)      Data 0.322 (0.322)      Loss
0.0710 (0.0710)    Prec 96.094% (96.094%)
Epoch: [103][100/391]   Time 0.050 (0.053)      Data 0.001 (0.004)      Loss
0.1116 (0.0542)    Prec 96.094% (97.935%)
Epoch: [103][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0357 (0.0560)    Prec 99.219% (97.991%)
Epoch: [103][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0505 (0.0581)    Prec 98.438% (97.975%)
Test: [0/79]    Time 0.151 (0.151)      Loss 0.3127 (0.3127)    Prec 91.406%
(91.406%)
 * Prec 89.370%
```

```
Epoch: [104][0/391]     Time 0.251 (0.251)      Data 0.212 (0.212)      Loss
0.0990 (0.0990)    Prec 96.875% (96.875%)
Epoch: [104][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0854 (0.0590)    Prec 96.875% (97.912%)
Epoch: [104][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0213 (0.0584)    Prec 99.219% (97.987%)
Epoch: [104][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0593 (0.0579)    Prec 99.219% (98.004%)
Test: [0/79]    Time 0.172 (0.172)      Loss 0.5651 (0.5651)    Prec 86.719%
(86.719%)
 * Prec 89.330%
Epoch: [105][0/391]     Time 0.256 (0.256)      Data 0.221 (0.221)      Loss
0.0594 (0.0594)    Prec 97.656% (97.656%)
Epoch: [105][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.2202 (0.0597)    Prec 94.531% (97.942%)
Epoch: [105][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0366 (0.0601)    Prec 100.000% (97.940%)
Epoch: [105][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0964 (0.0586)    Prec 97.656% (97.994%)
Test: [0/79]    Time 0.165 (0.165)      Loss 0.3138 (0.3138)    Prec 92.969%
(92.969%)
 * Prec 89.440%
Epoch: [106][0/391]     Time 0.335 (0.335)      Data 0.295 (0.295)      Loss
0.0399 (0.0399)    Prec 98.438% (98.438%)
Epoch: [106][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0138 (0.0525)    Prec 100.000% (98.151%)
Epoch: [106][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0463 (0.0551)    Prec 98.438% (98.119%)
Epoch: [106][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0572 (0.0565)    Prec 98.438% (98.056%)
Test: [0/79]    Time 0.209 (0.209)      Loss 0.4078 (0.4078)    Prec 88.281%
(88.281%)
 * Prec 89.180%
Epoch: [107][0/391]     Time 0.321 (0.321)      Data 0.244 (0.244)      Loss
0.0435 (0.0435)    Prec 99.219% (99.219%)
Epoch: [107][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0594 (0.0501)    Prec 98.438% (98.213%)
Epoch: [107][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0590 (0.0528)    Prec 98.438% (98.200%)
Epoch: [107][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0695 (0.0552)    Prec 96.875% (98.103%)
Test: [0/79]    Time 0.271 (0.271)      Loss 0.2451 (0.2451)    Prec 92.969%
(92.969%)
 * Prec 89.550%
Epoch: [108][0/391]     Time 0.315 (0.315)      Data 0.281 (0.281)      Loss
0.0149 (0.0149)    Prec 99.219% (99.219%)
Epoch: [108][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0400 (0.0558)    Prec 98.438% (98.074%)
```

```
Epoch: [108][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0543 (0.0531)    Prec 96.875% (98.162%)
Epoch: [108][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1186 (0.0534)    Prec 96.875% (98.147%)
Test: [0/79]    Time 0.235 (0.235)      Loss 0.4149 (0.4149)    Prec 88.281%
(88.281%)
 * Prec 89.200%
Epoch: [109][0/391]     Time 0.254 (0.254)      Data 0.220 (0.220)      Loss
0.0390 (0.0390)    Prec 97.656% (97.656%)
Epoch: [109][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0504 (0.0463)    Prec 97.656% (98.376%)
Epoch: [109][200/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0427 (0.0502)    Prec 99.219% (98.231%)
Epoch: [109][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1025 (0.0513)    Prec 96.875% (98.225%)
Test: [0/79]    Time 0.187 (0.187)      Loss 0.3608 (0.3608)    Prec 90.625%
(90.625%)
 * Prec 89.550%
Epoch: [110][0/391]     Time 0.290 (0.290)      Data 0.258 (0.258)      Loss
0.0697 (0.0697)    Prec 96.094% (96.094%)
Epoch: [110][100/391]   Time 0.042 (0.052)      Data 0.001 (0.004)      Loss
0.0415 (0.0535)    Prec 97.656% (98.144%)
Epoch: [110][200/391]   Time 0.043 (0.051)      Data 0.001 (0.002)      Loss
0.0825 (0.0557)    Prec 98.438% (98.033%)
Epoch: [110][300/391]   Time 0.042 (0.050)      Data 0.001 (0.002)      Loss
0.0224 (0.0548)    Prec 100.000% (98.108%)
Test: [0/79]    Time 0.210 (0.210)      Loss 0.2868 (0.2868)    Prec 93.750%
(93.750%)
 * Prec 89.540%
Epoch: [111][0/391]     Time 0.335 (0.335)      Data 0.261 (0.261)      Loss
0.0907 (0.0907)    Prec 96.875% (96.875%)
Epoch: [111][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0492 (0.0510)    Prec 99.219% (98.260%)
Epoch: [111][200/391]   Time 0.045 (0.051)      Data 0.001 (0.003)      Loss
0.0769 (0.0516)    Prec 96.094% (98.216%)
Epoch: [111][300/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0145 (0.0534)    Prec 100.000% (98.209%)
Test: [0/79]    Time 0.163 (0.163)      Loss 0.3811 (0.3811)    Prec 90.625%
(90.625%)
 * Prec 89.210%
Epoch: [112][0/391]     Time 0.244 (0.244)      Data 0.208 (0.208)      Loss
0.0505 (0.0505)    Prec 97.656% (97.656%)
Epoch: [112][100/391]   Time 0.047 (0.051)      Data 0.001 (0.003)      Loss
0.0941 (0.0562)    Prec 96.094% (98.074%)
Epoch: [112][200/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0192 (0.0545)    Prec 100.000% (98.177%)
Epoch: [112][300/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0394 (0.0542)    Prec 98.438% (98.149%)
```

```
Test: [0/79]    Time 0.156 (0.156)      Loss 0.3400 (0.3400)    Prec 90.625%
(90.625%)
 * Prec 89.830%
Epoch: [113][0/391]     Time 0.261 (0.261)      Data 0.223 (0.223)      Loss
0.0107 (0.0107)    Prec 100.000% (100.000%)
Epoch: [113][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0181 (0.0522)    Prec 100.000% (98.236%)
Epoch: [113][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0418 (0.0504)    Prec 98.438% (98.286%)
Epoch: [113][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1200 (0.0515)    Prec 97.656% (98.243%)
Test: [0/79]    Time 0.226 (0.226)      Loss 0.5252 (0.5252)    Prec 88.281%
(88.281%)
 * Prec 89.290%
Epoch: [114][0/391]     Time 0.382 (0.382)      Data 0.350 (0.350)      Loss
0.0409 (0.0409)    Prec 98.438% (98.438%)
Epoch: [114][100/391]   Time 0.049 (0.053)      Data 0.001 (0.005)      Loss
0.0637 (0.0565)    Prec 96.094% (98.074%)
Epoch: [114][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0648 (0.0543)    Prec 96.094% (98.064%)
Epoch: [114][300/391]   Time 0.050 (0.051)      Data 0.002 (0.002)      Loss
0.0384 (0.0540)    Prec 99.219% (98.069%)
Test: [0/79]    Time 0.237 (0.237)      Loss 0.3223 (0.3223)    Prec 91.406%
(91.406%)
 * Prec 89.320%
Epoch: [115][0/391]     Time 0.290 (0.290)      Data 0.251 (0.251)      Loss
0.0163 (0.0163)    Prec 99.219% (99.219%)
Epoch: [115][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0667 (0.0542)    Prec 98.438% (98.089%)
Epoch: [115][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0345 (0.0531)    Prec 99.219% (98.162%)
Epoch: [115][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1253 (0.0540)    Prec 96.094% (98.116%)
Test: [0/79]    Time 0.101 (0.101)      Loss 0.3839 (0.3839)    Prec 90.625%
(90.625%)
 * Prec 89.500%
Epoch: [116][0/391]     Time 0.320 (0.320)      Data 0.280 (0.280)      Loss
0.0146 (0.0146)    Prec 100.000% (100.000%)
Epoch: [116][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0211 (0.0564)    Prec 100.000% (98.260%)
Epoch: [116][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0441 (0.0529)    Prec 98.438% (98.305%)
Epoch: [116][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0293 (0.0500)    Prec 97.656% (98.375%)
Test: [0/79]    Time 0.152 (0.152)      Loss 0.3298 (0.3298)    Prec 93.750%
(93.750%)
 * Prec 89.420%
Epoch: [117][0/391]     Time 0.249 (0.249)      Data 0.210 (0.210)      Loss
```

```
0.0252 (0.0252)    Prec 100.000% (100.000%)
Epoch: [117][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0183 (0.0524)    Prec 100.000% (98.136%)
Epoch: [117][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0444 (0.0539)    Prec 99.219% (98.022%)
Epoch: [117][300/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0206 (0.0535)    Prec 99.219% (98.090%)
Test: [0/79]    Time 0.226 (0.226)      Loss 0.2736 (0.2736)    Prec 92.969%
(92.969%)
 * Prec 89.540%
Epoch: [118][0/391]     Time 0.290 (0.290)      Data 0.257 (0.257)      Loss
0.0310 (0.0310)    Prec 99.219% (99.219%)
Epoch: [118][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0426 (0.0526)    Prec 98.438% (98.314%)
Epoch: [118][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0303 (0.0522)    Prec 99.219% (98.294%)
Epoch: [118][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1000 (0.0541)    Prec 97.656% (98.238%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.3236 (0.3236)    Prec 92.188%
(92.188%)
 * Prec 89.190%
Epoch: [119][0/391]     Time 0.338 (0.338)      Data 0.298 (0.298)      Loss
0.0644 (0.0644)    Prec 97.656% (97.656%)
Epoch: [119][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0242 (0.0461)    Prec 99.219% (98.507%)
Epoch: [119][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0915 (0.0486)    Prec 96.094% (98.383%)
Epoch: [119][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0642 (0.0510)    Prec 96.094% (98.269%)
Test: [0/79]    Time 0.150 (0.150)      Loss 0.3918 (0.3918)    Prec 86.719%
(86.719%)
 * Prec 89.220%
Epoch: [120][0/391]     Time 0.243 (0.243)      Data 0.203 (0.203)      Loss
0.0726 (0.0726)    Prec 97.656% (97.656%)
Epoch: [120][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0342 (0.0550)    Prec 98.438% (98.004%)
Epoch: [120][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0149 (0.0548)    Prec 100.000% (98.053%)
Epoch: [120][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0725 (0.0539)    Prec 99.219% (98.090%)
Test: [0/79]    Time 0.250 (0.250)      Loss 0.3487 (0.3487)    Prec 89.844%
(89.844%)
 * Prec 89.560%
Epoch: [121][0/391]     Time 0.255 (0.255)      Data 0.222 (0.222)      Loss
0.1020 (0.1020)    Prec 96.875% (96.875%)
Epoch: [121][100/391]   Time 0.057 (0.052)      Data 0.001 (0.003)      Loss
0.0466 (0.0499)    Prec 98.438% (98.321%)
Epoch: [121][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.0784 (0.0493)    Prec 96.875% (98.360%)
Epoch: [121][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0847 (0.0531)    Prec 96.094% (98.243%)
Test: [0/79]    Time 0.170 (0.170)      Loss 0.3735 (0.3735)    Prec 91.406%
(91.406%)
 * Prec 89.170%
Epoch: [122][0/391]     Time 0.340 (0.340)      Data 0.303 (0.303)      Loss
0.0370 (0.0370)    Prec 99.219% (99.219%)
Epoch: [122][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0251 (0.0447)    Prec 99.219% (98.507%)
Epoch: [122][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0157 (0.0493)    Prec 100.000% (98.340%)
Epoch: [122][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0921 (0.0491)    Prec 96.094% (98.339%)
Test: [0/79]    Time 0.163 (0.163)      Loss 0.3189 (0.3189)    Prec 86.719%
(86.719%)
 * Prec 88.750%
Epoch: [123][0/391]     Time 0.225 (0.225)      Data 0.186 (0.186)      Loss
0.0705 (0.0705)    Prec 99.219% (99.219%)
Epoch: [123][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0900 (0.0475)    Prec 99.219% (98.399%)
Epoch: [123][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0483 (0.0486)    Prec 97.656% (98.340%)
Epoch: [123][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0142 (0.0476)    Prec 100.000% (98.365%)
Test: [0/79]    Time 0.183 (0.183)      Loss 0.2998 (0.2998)    Prec 92.969%
(92.969%)
 * Prec 89.530%
Epoch: [124][0/391]     Time 0.255 (0.255)      Data 0.214 (0.214)      Loss
0.0835 (0.0835)    Prec 98.438% (98.438%)
Epoch: [124][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1357 (0.0552)    Prec 94.531% (98.144%)
Epoch: [124][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0352 (0.0548)    Prec 98.438% (98.142%)
Epoch: [124][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0256 (0.0548)    Prec 99.219% (98.149%)
Test: [0/79]    Time 0.169 (0.169)      Loss 0.3536 (0.3536)    Prec 91.406%
(91.406%)
 * Prec 88.990%
Epoch: [125][0/391]     Time 0.259 (0.259)      Data 0.223 (0.223)      Loss
0.0132 (0.0132)    Prec 100.000% (100.000%)
Epoch: [125][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.1013 (0.0463)    Prec 95.312% (98.515%)
Epoch: [125][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0363 (0.0470)    Prec 98.438% (98.441%)
Epoch: [125][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0515 (0.0471)    Prec 98.438% (98.380%)
Test: [0/79]    Time 0.112 (0.112)      Loss 0.2276 (0.2276)    Prec 93.750%
```

```
(93.750%)
 * Prec 89.320%
Epoch: [126][0/391]     Time 0.300 (0.300)      Data 0.269 (0.269)      Loss
0.0794 (0.0794)    Prec 98.438% (98.438%)
Epoch: [126][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0658 (0.0459)    Prec 96.875% (98.499%)
Epoch: [126][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0640 (0.0473)    Prec 98.438% (98.406%)
Epoch: [126][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0803 (0.0482)    Prec 98.438% (98.360%)
Test: [0/79]    Time 0.274 (0.274)      Loss 0.3043 (0.3043)    Prec 91.406%
(91.406%)
 * Prec 88.900%
Epoch: [127][0/391]     Time 0.273 (0.273)      Data 0.232 (0.232)      Loss
0.0734 (0.0734)    Prec 97.656% (97.656%)
Epoch: [127][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1132 (0.0528)    Prec 95.312% (98.190%)
Epoch: [127][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0454 (0.0526)    Prec 97.656% (98.204%)
Epoch: [127][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0787 (0.0490)    Prec 97.656% (98.331%)
Test: [0/79]    Time 0.250 (0.250)      Loss 0.3678 (0.3678)    Prec 91.406%
(91.406%)
 * Prec 89.340%
Epoch: [128][0/391]     Time 0.364 (0.364)      Data 0.331 (0.331)      Loss
0.0397 (0.0397)    Prec 98.438% (98.438%)
Epoch: [128][100/391]   Time 0.049 (0.052)      Data 0.001 (0.005)      Loss
0.0694 (0.0524)    Prec 97.656% (98.182%)
Epoch: [128][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0392 (0.0501)    Prec 98.438% (98.266%)
Epoch: [128][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0251 (0.0494)    Prec 99.219% (98.292%)
Test: [0/79]    Time 0.176 (0.176)      Loss 0.3019 (0.3019)    Prec 92.969%
(92.969%)
 * Prec 89.910%
Epoch: [129][0/391]     Time 0.255 (0.255)      Data 0.218 (0.218)      Loss
0.0442 (0.0442)    Prec 98.438% (98.438%)
Epoch: [129][100/391]   Time 0.051 (0.051)      Data 0.001 (0.003)      Loss
0.0300 (0.0449)    Prec 99.219% (98.468%)
Epoch: [129][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0435 (0.0446)    Prec 98.438% (98.453%)
Epoch: [129][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0318 (0.0483)    Prec 99.219% (98.316%)
Test: [0/79]    Time 0.147 (0.147)      Loss 0.3759 (0.3759)    Prec 90.625%
(90.625%)
 * Prec 89.290%
Epoch: [130][0/391]     Time 0.196 (0.196)      Data 0.128 (0.128)      Loss
0.0895 (0.0895)    Prec 97.656% (97.656%)
```

```
Epoch: [130][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0444 (0.0470)    Prec 98.438% (98.414%)
Epoch: [130][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0389 (0.0486)    Prec 97.656% (98.325%)
Epoch: [130][300/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0299 (0.0506)    Prec 99.219% (98.245%)
Test: [0/79]    Time 0.224 (0.224)      Loss 0.2565 (0.2565)    Prec 92.188%
(92.188%)
 * Prec 89.410%
Epoch: [131][0/391]     Time 0.288 (0.288)      Data 0.257 (0.257)      Loss
0.0326 (0.0326)    Prec 99.219% (99.219%)
Epoch: [131][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.1122 (0.0485)    Prec 95.312% (98.260%)
Epoch: [131][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0224 (0.0486)    Prec 100.000% (98.352%)
Epoch: [131][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0672 (0.0485)    Prec 96.094% (98.352%)
Test: [0/79]    Time 0.163 (0.163)      Loss 0.3374 (0.3374)    Prec 91.406%
(91.406%)
 * Prec 89.140%
Epoch: [132][0/391]     Time 0.257 (0.257)      Data 0.217 (0.217)      Loss
0.0085 (0.0085)    Prec 100.000% (100.000%)
Epoch: [132][100/391]   Time 0.051 (0.052)      Data 0.001 (0.003)      Loss
0.0515 (0.0452)    Prec 96.875% (98.515%)
Epoch: [132][200/391]   Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0458 (0.0468)    Prec 96.875% (98.383%)
Epoch: [132][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0272 (0.0464)    Prec 99.219% (98.391%)
Test: [0/79]    Time 0.255 (0.255)      Loss 0.3988 (0.3988)    Prec 89.844%
(89.844%)
 * Prec 89.520%
Epoch: [133][0/391]     Time 0.259 (0.259)      Data 0.219 (0.219)      Loss
0.0160 (0.0160)    Prec 100.000% (100.000%)
Epoch: [133][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0135 (0.0453)    Prec 100.000% (98.407%)
Epoch: [133][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0349 (0.0460)    Prec 98.438% (98.414%)
Epoch: [133][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0391 (0.0475)    Prec 99.219% (98.344%)
Test: [0/79]    Time 0.224 (0.224)      Loss 0.3709 (0.3709)    Prec 90.625%
(90.625%)
 * Prec 89.620%
Epoch: [134][0/391]     Time 0.284 (0.284)      Data 0.252 (0.252)      Loss
0.1094 (0.1094)    Prec 96.875% (96.875%)
Epoch: [134][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0476 (0.0500)    Prec 97.656% (98.267%)
Epoch: [134][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0446 (0.0503)    Prec 97.656% (98.282%)
```

```
Epoch: [134][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0755 (0.0499)    Prec 96.875% (98.282%)
Test: [0/79]    Time 0.142 (0.142)      Loss 0.3238 (0.3238)    Prec 92.188%
(92.188%)
 * Prec 89.300%
Epoch: [135][0/391]     Time 0.250 (0.250)      Data 0.217 (0.217)      Loss
0.0370 (0.0370)    Prec 98.438% (98.438%)
Epoch: [135][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0807 (0.0454)    Prec 97.656% (98.399%)
Epoch: [135][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1234 (0.0493)    Prec 95.312% (98.290%)
Epoch: [135][300/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0561 (0.0486)    Prec 98.438% (98.318%)
Test: [0/79]    Time 0.337 (0.337)      Loss 0.3000 (0.3000)    Prec 91.406%
(91.406%)
 * Prec 89.280%
Epoch: [136][0/391]     Time 0.288 (0.288)      Data 0.255 (0.255)      Loss
0.0486 (0.0486)    Prec 98.438% (98.438%)
Epoch: [136][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0857 (0.0515)    Prec 96.875% (98.167%)
Epoch: [136][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0678 (0.0505)    Prec 96.875% (98.193%)
Epoch: [136][300/391]   Time 0.051 (0.050)      Data 0.001 (0.002)      Loss
0.0567 (0.0487)    Prec 97.656% (98.282%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.3408 (0.3408)    Prec 92.188%
(92.188%)
 * Prec 89.420%
Epoch: [137][0/391]     Time 0.246 (0.246)      Data 0.214 (0.214)      Loss
0.0197 (0.0197)    Prec 99.219% (99.219%)
Epoch: [137][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0222 (0.0437)    Prec 99.219% (98.422%)
Epoch: [137][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0514 (0.0450)    Prec 98.438% (98.395%)
Epoch: [137][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0194 (0.0454)    Prec 99.219% (98.401%)
Test: [0/79]    Time 0.213 (0.213)      Loss 0.3427 (0.3427)    Prec 90.625%
(90.625%)
 * Prec 89.300%
Epoch: [138][0/391]     Time 0.386 (0.386)      Data 0.355 (0.355)      Loss
0.0522 (0.0522)    Prec 97.656% (97.656%)
Epoch: [138][100/391]   Time 0.049 (0.053)      Data 0.001 (0.005)      Loss
0.0894 (0.0448)    Prec 96.875% (98.414%)
Epoch: [138][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0126 (0.0454)    Prec 100.000% (98.403%)
Epoch: [138][300/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0492 (0.0472)    Prec 97.656% (98.347%)
Test: [0/79]    Time 0.147 (0.147)      Loss 0.2791 (0.2791)    Prec 92.188%
(92.188%)
```

```
 * Prec 89.200%
Epoch: [139][0/391]     Time 0.255 (0.255)      Data 0.224 (0.224)      Loss
0.0460 (0.0460)    Prec 98.438% (98.438%)
Epoch: [139][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0257 (0.0471)    Prec 98.438% (98.430%)
Epoch: [139][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1073 (0.0455)    Prec 96.875% (98.430%)
Epoch: [139][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0761 (0.0442)    Prec 96.875% (98.487%)
Test: [0/79]    Time 0.114 (0.114)      Loss 0.4086 (0.4086)    Prec 89.844%
(89.844%)
 * Prec 89.320%
Epoch: [140][0/391]     Time 0.298 (0.298)      Data 0.261 (0.261)      Loss
0.0564 (0.0564)    Prec 98.438% (98.438%)
Epoch: [140][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0197 (0.0455)    Prec 100.000% (98.438%)
Epoch: [140][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0106 (0.0436)    Prec 99.219% (98.515%)
Epoch: [140][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0105 (0.0452)    Prec 99.219% (98.425%)
Test: [0/79]    Time 0.161 (0.161)      Loss 0.3374 (0.3374)    Prec 91.406%
(91.406%)
 * Prec 88.690%
Epoch: [141][0/391]     Time 0.257 (0.257)      Data 0.223 (0.223)      Loss
0.0086 (0.0086)    Prec 100.000% (100.000%)
Epoch: [141][100/391]   Time 0.049 (0.051)      Data 0.002 (0.004)      Loss
0.0370 (0.0437)    Prec 98.438% (98.538%)
Epoch: [141][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0490 (0.0444)    Prec 98.438% (98.558%)
Epoch: [141][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0118 (0.0448)    Prec 99.219% (98.508%)
Test: [0/79]    Time 0.174 (0.174)      Loss 0.3364 (0.3364)    Prec 90.625%
(90.625%)
 * Prec 89.430%
Epoch: [142][0/391]     Time 0.251 (0.251)      Data 0.219 (0.219)      Loss
0.0561 (0.0561)    Prec 96.875% (96.875%)
Epoch: [142][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0521 (0.0418)    Prec 97.656% (98.422%)
Epoch: [142][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0141 (0.0422)    Prec 99.219% (98.465%)
Epoch: [142][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0185 (0.0448)    Prec 99.219% (98.378%)
Test: [0/79]    Time 0.115 (0.115)      Loss 0.2626 (0.2626)    Prec 89.844%
(89.844%)
 * Prec 89.150%
Epoch: [143][0/391]     Time 0.270 (0.270)      Data 0.229 (0.229)      Loss
0.0326 (0.0326)    Prec 98.438% (98.438%)
Epoch: [143][100/391]   Time 0.041 (0.052)      Data 0.001 (0.003)      Loss
```

```
0.0166 (0.0489)    Prec 100.000% (98.244%)
Epoch: [143][200/391]   Time 0.042 (0.050)      Data 0.001 (0.002)      Loss
0.0029 (0.0474)    Prec 100.000% (98.309%)
Epoch: [143][300/391]   Time 0.042 (0.050)      Data 0.001 (0.002)      Loss
0.0352 (0.0466)    Prec 99.219% (98.399%)
Test: [0/79]    Time 0.159 (0.159)      Loss 0.3184 (0.3184)    Prec 89.062%
(89.062%)
 * Prec 89.720%
Epoch: [144][0/391]     Time 0.242 (0.242)      Data 0.208 (0.208)      Loss
0.0520 (0.0520)    Prec 97.656% (97.656%)
Epoch: [144][100/391]   Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.0328 (0.0404)    Prec 99.219% (98.530%)
Epoch: [144][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0183 (0.0439)    Prec 99.219% (98.465%)
Epoch: [144][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0253 (0.0463)    Prec 99.219% (98.409%)
Test: [0/79]    Time 0.158 (0.158)      Loss 0.3572 (0.3572)    Prec 92.188%
(92.188%)
 * Prec 89.190%
Epoch: [145][0/391]     Time 0.355 (0.355)      Data 0.319 (0.319)      Loss
0.0478 (0.0478)    Prec 97.656% (97.656%)
Epoch: [145][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0471 (0.0445)    Prec 99.219% (98.445%)
Epoch: [145][200/391]   Time 0.048 (0.051)      Data 0.001 (0.003)      Loss
0.0794 (0.0441)    Prec 98.438% (98.484%)
Epoch: [145][300/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.1218 (0.0467)    Prec 96.875% (98.419%)
Test: [0/79]    Time 0.241 (0.241)      Loss 0.3299 (0.3299)    Prec 89.062%
(89.062%)
 * Prec 89.460%
Epoch: [146][0/391]     Time 0.249 (0.249)      Data 0.219 (0.219)      Loss
0.0241 (0.0241)    Prec 98.438% (98.438%)
Epoch: [146][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0163 (0.0449)    Prec 99.219% (98.453%)
Epoch: [146][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0509 (0.0447)    Prec 97.656% (98.391%)
Epoch: [146][300/391]   Time 0.051 (0.050)      Data 0.001 (0.002)      Loss
0.0057 (0.0430)    Prec 100.000% (98.471%)
Test: [0/79]    Time 0.113 (0.113)      Loss 0.3950 (0.3950)    Prec 89.844%
(89.844%)
 * Prec 89.610%
Epoch: [147][0/391]     Time 0.290 (0.290)      Data 0.255 (0.255)      Loss
0.0139 (0.0139)    Prec 99.219% (99.219%)
Epoch: [147][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0461 (0.0460)    Prec 98.438% (98.476%)
Epoch: [147][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0361 (0.0466)    Prec 98.438% (98.403%)
Epoch: [147][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.0230 (0.0458)    Prec 99.219% (98.404%)
Test: [0/79]    Time 0.161 (0.161)      Loss 0.3413 (0.3413)    Prec 92.969%
(92.969%)
 * Prec 89.310%
Epoch: [148][0/391]     Time 0.252 (0.252)      Data 0.222 (0.222)      Loss
0.1147 (0.1147)    Prec 96.875% (96.875%)
Epoch: [148][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0233 (0.0421)    Prec 100.000% (98.577%)
Epoch: [148][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0084 (0.0424)    Prec 100.000% (98.550%)
Epoch: [148][300/391]   Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0203 (0.0427)    Prec 99.219% (98.531%)
Test: [0/79]    Time 0.154 (0.154)      Loss 0.3788 (0.3788)    Prec 90.625%
(90.625%)
 * Prec 89.380%
Epoch: [149][0/391]     Time 0.301 (0.301)      Data 0.269 (0.269)      Loss
0.0578 (0.0578)    Prec 96.875% (96.875%)
Epoch: [149][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0513 (0.0382)    Prec 99.219% (98.693%)
Epoch: [149][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0486 (0.0429)    Prec 98.438% (98.504%)
Epoch: [149][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0937 (0.0456)    Prec 96.875% (98.406%)
Test: [0/79]    Time 0.171 (0.171)      Loss 0.3011 (0.3011)    Prec 93.750%
(93.750%)
 * Prec 89.360%
Epoch: [150][0/391]     Time 0.256 (0.256)      Data 0.223 (0.223)      Loss
0.0440 (0.0440)    Prec 98.438% (98.438%)
Epoch: [150][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0836 (0.0468)    Prec 96.094% (98.321%)
Epoch: [150][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0090 (0.0463)    Prec 100.000% (98.364%)
Epoch: [150][300/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0322 (0.0465)    Prec 98.438% (98.341%)
Test: [0/79]    Time 0.266 (0.266)      Loss 0.3495 (0.3495)    Prec 89.844%
(89.844%)
 * Prec 89.350%
Epoch: [151][0/391]     Time 0.276 (0.276)      Data 0.236 (0.236)      Loss
0.1092 (0.1092)    Prec 96.875% (96.875%)
Epoch: [151][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0698 (0.0422)    Prec 96.094% (98.546%)
Epoch: [151][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0465 (0.0423)    Prec 99.219% (98.519%)
Epoch: [151][300/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0511 (0.0430)    Prec 96.875% (98.482%)
Test: [0/79]    Time 0.242 (0.242)      Loss 0.3881 (0.3881)    Prec 91.406%
(91.406%)
 * Prec 89.730%
```

```
Epoch: [152][0/391]     Time 0.346 (0.346)      Data 0.309 (0.309)      Loss
0.0439 (0.0439)    Prec 98.438% (98.438%)
Epoch: [152][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0881 (0.0449)    Prec 97.656% (98.360%)
Epoch: [152][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.1144 (0.0444)    Prec 96.094% (98.438%)
Epoch: [152][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0314 (0.0439)    Prec 99.219% (98.489%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.4334 (0.4334)    Prec 89.844%
(89.844%)
 * Prec 89.290%
Epoch: [153][0/391]     Time 0.260 (0.260)      Data 0.228 (0.228)      Loss
0.0577 (0.0577)    Prec 97.656% (97.656%)
Epoch: [153][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0487 (0.0457)    Prec 98.438% (98.414%)
Epoch: [153][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0461 (0.0456)    Prec 97.656% (98.426%)
Epoch: [153][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0245 (0.0456)    Prec 98.438% (98.419%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.2874 (0.2874)    Prec 92.969%
(92.969%)
 * Prec 89.450%
Epoch: [154][0/391]     Time 0.256 (0.256)      Data 0.223 (0.223)      Loss
0.0066 (0.0066)    Prec 100.000% (100.000%)
Epoch: [154][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0656 (0.0431)    Prec 97.656% (98.615%)
Epoch: [154][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0523 (0.0428)    Prec 99.219% (98.558%)
Epoch: [154][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0481 (0.0412)    Prec 98.438% (98.643%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3679 (0.3679)    Prec 89.844%
(89.844%)
 * Prec 89.400%
Epoch: [155][0/391]     Time 0.267 (0.267)      Data 0.226 (0.226)      Loss
0.0348 (0.0348)    Prec 99.219% (99.219%)
Epoch: [155][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0273 (0.0432)    Prec 98.438% (98.453%)
Epoch: [155][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0131 (0.0461)    Prec 100.000% (98.348%)
Epoch: [155][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0131 (0.0452)    Prec 100.000% (98.409%)
Test: [0/79]    Time 0.147 (0.147)      Loss 0.3179 (0.3179)    Prec 92.188%
(92.188%)
 * Prec 89.170%
Epoch: [156][0/391]     Time 0.247 (0.247)      Data 0.214 (0.214)      Loss
0.0175 (0.0175)    Prec 99.219% (99.219%)
Epoch: [156][100/391]   Time 0.051 (0.051)      Data 0.001 (0.003)      Loss
0.0266 (0.0405)    Prec 99.219% (98.584%)
```

```
Epoch: [156][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0244 (0.0427)    Prec 99.219% (98.535%)
Epoch: [156][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0479 (0.0432)    Prec 97.656% (98.513%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.4462 (0.4462)    Prec 89.062%
(89.062%)
 * Prec 89.410%
Epoch: [157][0/391]     Time 0.263 (0.263)      Data 0.223 (0.223)      Loss
0.0138 (0.0138)    Prec 99.219% (99.219%)
Epoch: [157][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0349 (0.0438)    Prec 98.438% (98.407%)
Epoch: [157][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0242 (0.0432)    Prec 98.438% (98.449%)
Epoch: [157][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1243 (0.0429)    Prec 95.312% (98.505%)
Test: [0/79]    Time 0.151 (0.151)      Loss 0.3717 (0.3717)    Prec 92.188%
(92.188%)
 * Prec 89.240%
Epoch: [158][0/391]     Time 0.255 (0.255)      Data 0.224 (0.224)      Loss
0.0599 (0.0599)    Prec 97.656% (97.656%)
Epoch: [158][100/391]   Time 0.049 (0.051)      Data 0.001 (0.004)      Loss
0.0366 (0.0434)    Prec 98.438% (98.376%)
Epoch: [158][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0214 (0.0440)    Prec 99.219% (98.426%)
Epoch: [158][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0038 (0.0437)    Prec 100.000% (98.450%)
Test: [0/79]    Time 0.152 (0.152)      Loss 0.4062 (0.4062)    Prec 90.625%
(90.625%)
 * Prec 89.670%
Epoch: [159][0/391]     Time 0.256 (0.256)      Data 0.223 (0.223)      Loss
0.0270 (0.0270)    Prec 99.219% (99.219%)
Epoch: [159][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0659 (0.0408)    Prec 96.094% (98.414%)
Epoch: [159][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0423 (0.0398)    Prec 99.219% (98.535%)
Epoch: [159][300/391]   Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0433 (0.0418)    Prec 98.438% (98.474%)
Test: [0/79]    Time 0.222 (0.222)      Loss 0.4077 (0.4077)    Prec 91.406%
(91.406%)
 * Prec 89.630%
Epoch: [160][0/391]     Time 0.295 (0.295)      Data 0.257 (0.257)      Loss
0.0611 (0.0611)    Prec 98.438% (98.438%)
Epoch: [160][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0105 (0.0443)    Prec 100.000% (98.492%)
Epoch: [160][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0479 (0.0470)    Prec 99.219% (98.391%)
Epoch: [160][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0349 (0.0475)    Prec 98.438% (98.417%)
```

```
Test: [0/79]    Time 0.166 (0.166)      Loss 0.3368 (0.3368)    Prec 90.625%
(90.625%)
 * Prec 89.820%
Epoch: [161][0/391]     Time 0.258 (0.258)      Data 0.224 (0.224)      Loss
0.0550 (0.0550)    Prec 99.219% (99.219%)
Epoch: [161][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0768 (0.0407)    Prec 96.875% (98.530%)
Epoch: [161][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0183 (0.0393)    Prec 100.000% (98.585%)
Epoch: [161][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0535 (0.0394)    Prec 98.438% (98.637%)
Test: [0/79]    Time 0.247 (0.247)      Loss 0.4287 (0.4287)    Prec 90.625%
(90.625%)
 * Prec 89.370%
Epoch: [162][0/391]     Time 0.270 (0.270)      Data 0.237 (0.237)      Loss
0.0623 (0.0623)    Prec 97.656% (97.656%)
Epoch: [162][100/391]   Time 0.058 (0.052)      Data 0.001 (0.004)      Loss
0.0822 (0.0405)    Prec 98.438% (98.584%)
Epoch: [162][200/391]   Time 0.057 (0.051)      Data 0.001 (0.002)      Loss
0.0952 (0.0398)    Prec 96.875% (98.612%)
Epoch: [162][300/391]   Time 0.057 (0.050)      Data 0.001 (0.002)      Loss
0.0222 (0.0418)    Prec 99.219% (98.552%)
Test: [0/79]    Time 0.147 (0.147)      Loss 0.4864 (0.4864)    Prec 86.719%
(86.719%)
 * Prec 89.600%
Epoch: [163][0/391]     Time 0.258 (0.258)      Data 0.224 (0.224)      Loss
0.0186 (0.0186)    Prec 99.219% (99.219%)
Epoch: [163][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.1020 (0.0444)    Prec 96.094% (98.368%)
Epoch: [163][200/391]   Time 0.050 (0.051)      Data 0.002 (0.002)      Loss
0.0853 (0.0452)    Prec 95.312% (98.395%)
Epoch: [163][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0658 (0.0456)    Prec 97.656% (98.388%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.4412 (0.4412)    Prec 89.844%
(89.844%)
 * Prec 89.090%
Epoch: [164][0/391]     Time 0.236 (0.236)      Data 0.203 (0.203)      Loss
0.0341 (0.0341)    Prec 98.438% (98.438%)
Epoch: [164][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0213 (0.0420)    Prec 99.219% (98.615%)
Epoch: [164][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0458 (0.0441)    Prec 98.438% (98.469%)
Epoch: [164][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0515 (0.0453)    Prec 98.438% (98.422%)
Test: [0/79]    Time 0.219 (0.219)      Loss 0.3363 (0.3363)    Prec 92.969%
(92.969%)
 * Prec 89.830%
Epoch: [165][0/391]     Time 0.295 (0.295)      Data 0.265 (0.265)      Loss
```

```
0.0050 (0.0050)    Prec 100.000% (100.000%)
Epoch: [165][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0747 (0.0395)    Prec 98.438% (98.592%)
Epoch: [165][200/391]   Time 0.049 (0.050)      Data 0.001 (0.003)      Loss
0.0186 (0.0414)    Prec 100.000% (98.539%)
Epoch: [165][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0290 (0.0446)    Prec 98.438% (98.432%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.4003 (0.4003)    Prec 90.625%
(90.625%)
 * Prec 89.830%
Epoch: [166][0/391]     Time 0.349 (0.349)      Data 0.308 (0.308)      Loss
0.0124 (0.0124)    Prec 100.000% (100.000%)
Epoch: [166][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0725 (0.0357)    Prec 97.656% (98.786%)
Epoch: [166][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0680 (0.0394)    Prec 96.875% (98.678%)
Epoch: [166][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0057 (0.0401)    Prec 100.000% (98.653%)
Test: [0/79]    Time 0.152 (0.152)      Loss 0.4663 (0.4663)    Prec 87.500%
(87.500%)
 * Prec 89.710%
Epoch: [167][0/391]     Time 0.258 (0.258)      Data 0.219 (0.219)      Loss
0.0491 (0.0491)    Prec 97.656% (97.656%)
Epoch: [167][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0571 (0.0457)    Prec 96.875% (98.453%)
Epoch: [167][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0717 (0.0449)    Prec 96.875% (98.453%)
Epoch: [167][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0093 (0.0442)    Prec 100.000% (98.487%)
Test: [0/79]    Time 0.171 (0.171)      Loss 0.3329 (0.3329)    Prec 90.625%
(90.625%)
 * Prec 89.700%
Epoch: [168][0/391]     Time 0.235 (0.235)      Data 0.194 (0.194)      Loss
0.0308 (0.0308)    Prec 98.438% (98.438%)
Epoch: [168][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0118 (0.0375)    Prec 100.000% (98.739%)
Epoch: [168][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0835 (0.0378)    Prec 98.438% (98.717%)
Epoch: [168][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0977 (0.0384)    Prec 96.094% (98.718%)
Test: [0/79]    Time 0.245 (0.245)      Loss 0.3674 (0.3674)    Prec 90.625%
(90.625%)
 * Prec 88.990%
Epoch: [169][0/391]     Time 0.342 (0.342)      Data 0.309 (0.309)      Loss
0.0591 (0.0591)    Prec 97.656% (97.656%)
Epoch: [169][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0149 (0.0389)    Prec 99.219% (98.685%)
Epoch: [169][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
```

```
0.0361 (0.0412)    Prec 99.219% (98.682%)
Epoch: [169][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0543 (0.0426)    Prec 98.438% (98.578%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.3981 (0.3981)    Prec 90.625%
(90.625%)
 * Prec 89.810%
Epoch: [170][0/391]     Time 0.289 (0.289)      Data 0.257 (0.257)      Loss
0.0135 (0.0135)    Prec 99.219% (99.219%)
Epoch: [170][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0200 (0.0331)    Prec 100.000% (98.840%)
Epoch: [170][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0205 (0.0365)    Prec 99.219% (98.795%)
Epoch: [170][300/391]   Time 0.050 (0.050)      Data 0.002 (0.002)      Loss
0.0109 (0.0366)    Prec 100.000% (98.785%)
Test: [0/79]    Time 0.156 (0.156)      Loss 0.4442 (0.4442)    Prec 89.844%
(89.844%)
 * Prec 89.780%
Epoch: [171][0/391]     Time 0.259 (0.259)      Data 0.228 (0.228)      Loss
0.0644 (0.0644)    Prec 96.875% (96.875%)
Epoch: [171][100/391]   Time 0.049 (0.051)      Data 0.001 (0.004)      Loss
0.0419 (0.0401)    Prec 98.438% (98.724%)
Epoch: [171][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0660 (0.0388)    Prec 96.094% (98.737%)
Epoch: [171][300/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0278 (0.0378)    Prec 99.219% (98.726%)
Test: [0/79]    Time 0.223 (0.223)      Loss 0.3334 (0.3334)    Prec 89.844%
(89.844%)
 * Prec 89.560%
Epoch: [172][0/391]     Time 0.284 (0.284)      Data 0.252 (0.252)      Loss
0.0325 (0.0325)    Prec 99.219% (99.219%)
Epoch: [172][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0474 (0.0420)    Prec 98.438% (98.569%)
Epoch: [172][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0089 (0.0415)    Prec 100.000% (98.605%)
Epoch: [172][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0565 (0.0416)    Prec 97.656% (98.567%)
Test: [0/79]    Time 0.159 (0.159)      Loss 0.5079 (0.5079)    Prec 85.938%
(85.938%)
 * Prec 89.330%
Epoch: [173][0/391]     Time 0.255 (0.255)      Data 0.221 (0.221)      Loss
0.0617 (0.0617)    Prec 96.094% (96.094%)
Epoch: [173][100/391]   Time 0.049 (0.051)      Data 0.002 (0.004)      Loss
0.0448 (0.0355)    Prec 97.656% (98.778%)
Epoch: [173][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0214 (0.0383)    Prec 98.438% (98.640%)
Epoch: [173][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0490 (0.0378)    Prec 98.438% (98.624%)
Test: [0/79]    Time 0.229 (0.229)      Loss 0.3749 (0.3749)    Prec 92.188%
```

```
(92.188%)
 * Prec 89.730%
Epoch: [174][0/391]     Time 0.332 (0.332)      Data 0.257 (0.257)      Loss
0.0159 (0.0159)    Prec 100.000% (100.000%)
Epoch: [174][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0154 (0.0387)    Prec 99.219% (98.693%)
Epoch: [174][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0065 (0.0355)    Prec 100.000% (98.772%)
Epoch: [174][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0388 (0.0378)    Prec 98.438% (98.723%)
Test: [0/79]    Time 0.149 (0.149)      Loss 0.4670 (0.4670)    Prec 89.062%
(89.062%)
 * Prec 89.250%
Epoch: [175][0/391]     Time 0.334 (0.334)      Data 0.303 (0.303)      Loss
0.0714 (0.0714)    Prec 98.438% (98.438%)
Epoch: [175][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0260 (0.0408)    Prec 98.438% (98.608%)
Epoch: [175][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0056 (0.0414)    Prec 100.000% (98.562%)
Epoch: [175][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0308 (0.0432)    Prec 97.656% (98.508%)
Test: [0/79]    Time 0.154 (0.154)      Loss 0.4031 (0.4031)    Prec 92.969%
(92.969%)
 * Prec 89.330%
Epoch: [176][0/391]     Time 0.263 (0.263)      Data 0.224 (0.224)      Loss
0.0458 (0.0458)    Prec 99.219% (99.219%)
Epoch: [176][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0408 (0.0450)    Prec 98.438% (98.461%)
Epoch: [176][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0406 (0.0415)    Prec 98.438% (98.585%)
Epoch: [176][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0042 (0.0409)    Prec 100.000% (98.617%)
Test: [0/79]    Time 0.222 (0.222)      Loss 0.3617 (0.3617)    Prec 89.844%
(89.844%)
 * Prec 89.470%
Epoch: [177][0/391]     Time 0.297 (0.297)      Data 0.257 (0.257)      Loss
0.0304 (0.0304)    Prec 99.219% (99.219%)
Epoch: [177][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0703 (0.0386)    Prec 98.438% (98.716%)
Epoch: [177][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0056 (0.0369)    Prec 100.000% (98.745%)
Epoch: [177][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0354 (0.0392)    Prec 98.438% (98.648%)
Test: [0/79]    Time 0.181 (0.181)      Loss 0.4089 (0.4089)    Prec 90.625%
(90.625%)
 * Prec 89.770%
Epoch: [178][0/391]     Time 0.254 (0.254)      Data 0.222 (0.222)      Loss
0.0499 (0.0499)    Prec 98.438% (98.438%)
```

```
Epoch: [178][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.1181 (0.0388)    Prec 97.656% (98.554%)
Epoch: [178][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0139 (0.0394)    Prec 99.219% (98.577%)
Epoch: [178][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0910 (0.0416)    Prec 96.875% (98.510%)
Test: [0/79]    Time 0.251 (0.251)      Loss 0.4295 (0.4295)    Prec 90.625%
(90.625%)
 * Prec 89.740%
Epoch: [179][0/391]     Time 0.254 (0.254)      Data 0.222 (0.222)      Loss
0.0588 (0.0588)    Prec 98.438% (98.438%)
Epoch: [179][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0097 (0.0409)    Prec 100.000% (98.600%)
Epoch: [179][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0244 (0.0404)    Prec 99.219% (98.558%)
Epoch: [179][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0064 (0.0417)    Prec 100.000% (98.531%)
Test: [0/79]    Time 0.225 (0.225)      Loss 0.3736 (0.3736)    Prec 92.188%
(92.188%)
 * Prec 89.260%
Epoch: [180][0/391]     Time 0.335 (0.335)      Data 0.258 (0.258)      Loss
0.0266 (0.0266)    Prec 98.438% (98.438%)
Epoch: [180][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0373 (0.0397)    Prec 98.438% (98.708%)
Epoch: [180][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0579 (0.0395)    Prec 96.875% (98.663%)
Epoch: [180][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0520 (0.0399)    Prec 98.438% (98.619%)
Test: [0/79]    Time 0.102 (0.102)      Loss 0.3454 (0.3454)    Prec 89.062%
(89.062%)
 * Prec 89.630%
Epoch: [181][0/391]     Time 0.293 (0.293)      Data 0.258 (0.258)      Loss
0.0649 (0.0649)    Prec 98.438% (98.438%)
Epoch: [181][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0407 (0.0334)    Prec 98.438% (98.832%)
Epoch: [181][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0214 (0.0341)    Prec 99.219% (98.842%)
Epoch: [181][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0261 (0.0349)    Prec 99.219% (98.809%)
Test: [0/79]    Time 0.210 (0.210)      Loss 0.4393 (0.4393)    Prec 87.500%
(87.500%)
 * Prec 89.740%
Epoch: [182][0/391]     Time 0.283 (0.283)      Data 0.252 (0.252)      Loss
0.0600 (0.0600)    Prec 98.438% (98.438%)
Epoch: [182][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0651 (0.0373)    Prec 98.438% (98.631%)
Epoch: [182][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0922 (0.0393)    Prec 96.875% (98.566%)
```

```
Epoch: [182][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0852 (0.0419)    Prec 97.656% (98.487%)
Test: [0/79]    Time 0.238 (0.238)      Loss 0.3728 (0.3728)    Prec 92.969%
(92.969%)
 * Prec 89.690%
Epoch: [183][0/391]     Time 0.254 (0.254)      Data 0.221 (0.221)      Loss
0.0456 (0.0456)    Prec 96.875% (96.875%)
Epoch: [183][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0155 (0.0352)    Prec 100.000% (98.801%)
Epoch: [183][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0504 (0.0367)    Prec 96.875% (98.764%)
Epoch: [183][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0492 (0.0378)    Prec 99.219% (98.715%)
Test: [0/79]    Time 0.165 (0.165)      Loss 0.4856 (0.4856)    Prec 89.062%
(89.062%)
 * Prec 89.150%
Epoch: [184][0/391]     Time 0.242 (0.242)      Data 0.207 (0.207)      Loss
0.0217 (0.0217)    Prec 99.219% (99.219%)
Epoch: [184][100/391]   Time 0.052 (0.051)      Data 0.001 (0.003)      Loss
0.0649 (0.0392)    Prec 97.656% (98.708%)
Epoch: [184][200/391]   Time 0.053 (0.050)      Data 0.001 (0.002)      Loss
0.0300 (0.0364)    Prec 99.219% (98.783%)
Epoch: [184][300/391]   Time 0.052 (0.050)      Data 0.001 (0.002)      Loss
0.0093 (0.0370)    Prec 100.000% (98.765%)
Test: [0/79]    Time 0.218 (0.218)      Loss 0.3392 (0.3392)    Prec 92.188%
(92.188%)
 * Prec 89.510%
Epoch: [185][0/391]     Time 0.286 (0.286)      Data 0.245 (0.245)      Loss
0.0816 (0.0816)    Prec 98.438% (98.438%)
Epoch: [185][100/391]   Time 0.050 (0.052)      Data 0.002 (0.004)      Loss
0.0501 (0.0349)    Prec 97.656% (98.902%)
Epoch: [185][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0042 (0.0356)    Prec 100.000% (98.850%)
Epoch: [185][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0232 (0.0366)    Prec 99.219% (98.816%)
Test: [0/79]    Time 0.239 (0.239)      Loss 0.3995 (0.3995)    Prec 91.406%
(91.406%)
 * Prec 89.460%
Epoch: [186][0/391]     Time 0.263 (0.263)      Data 0.223 (0.223)      Loss
0.0261 (0.0261)    Prec 98.438% (98.438%)
Epoch: [186][100/391]   Time 0.042 (0.051)      Data 0.001 (0.003)      Loss
0.0257 (0.0384)    Prec 99.219% (98.608%)
Epoch: [186][200/391]   Time 0.042 (0.050)      Data 0.001 (0.002)      Loss
0.0402 (0.0352)    Prec 99.219% (98.733%)
Epoch: [186][300/391]   Time 0.042 (0.050)      Data 0.001 (0.002)      Loss
0.0375 (0.0366)    Prec 98.438% (98.702%)
Test: [0/79]    Time 0.159 (0.159)      Loss 0.3704 (0.3704)    Prec 90.625%
(90.625%)
```

```
 * Prec 89.470%
Epoch: [187][0/391]     Time 0.257 (0.257)      Data 0.217 (0.217)      Loss
0.0084 (0.0084)    Prec 100.000% (100.000%)
Epoch: [187][100/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0432 (0.0402)    Prec 98.438% (98.615%)
Epoch: [187][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0108 (0.0363)    Prec 99.219% (98.698%)
Epoch: [187][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0445 (0.0380)    Prec 98.438% (98.653%)
Test: [0/79]    Time 0.257 (0.257)      Loss 0.2927 (0.2927)    Prec 94.531%
(94.531%)
 * Prec 90.120%
Epoch: [188][0/391]     Time 0.256 (0.256)      Data 0.224 (0.224)      Loss
0.0185 (0.0185)    Prec 100.000% (100.000%)
Epoch: [188][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0637 (0.0361)    Prec 97.656% (98.770%)
Epoch: [188][200/391]   Time 0.049 (0.051)      Data 0.002 (0.002)      Loss
0.0666 (0.0357)    Prec 98.438% (98.842%)
Epoch: [188][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0518 (0.0353)    Prec 97.656% (98.806%)
Test: [0/79]    Time 0.221 (0.221)      Loss 0.4201 (0.4201)    Prec 90.625%
(90.625%)
 * Prec 90.040%
Epoch: [189][0/391]     Time 0.302 (0.302)      Data 0.263 (0.263)      Loss
0.0773 (0.0773)    Prec 97.656% (97.656%)
Epoch: [189][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0096 (0.0306)    Prec 99.219% (98.886%)
Epoch: [189][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0449 (0.0351)    Prec 96.875% (98.729%)
Epoch: [189][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0160 (0.0365)    Prec 99.219% (98.687%)
Test: [0/79]    Time 0.161 (0.161)      Loss 0.4779 (0.4779)    Prec 90.625%
(90.625%)
 * Prec 90.100%
Epoch: [190][0/391]     Time 0.261 (0.261)      Data 0.224 (0.224)      Loss
0.0985 (0.0985)    Prec 96.875% (96.875%)
Epoch: [190][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0598 (0.0371)    Prec 97.656% (98.677%)
Epoch: [190][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0612 (0.0371)    Prec 97.656% (98.733%)
Epoch: [190][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0311 (0.0375)    Prec 98.438% (98.718%)
Test: [0/79]    Time 0.176 (0.176)      Loss 0.3175 (0.3175)    Prec 92.188%
(92.188%)
 * Prec 89.640%
Epoch: [191][0/391]     Time 0.255 (0.255)      Data 0.222 (0.222)      Loss
0.0777 (0.0777)    Prec 96.875% (96.875%)
Epoch: [191][100/391]   Time 0.046 (0.051)      Data 0.001 (0.003)      Loss
```

```
0.0060 (0.0374)    Prec 100.000% (98.623%)
Epoch: [191][200/391]   Time 0.051 (0.050)      Data 0.001 (0.002)      Loss
0.0603 (0.0404)    Prec 97.656% (98.577%)
Epoch: [191][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0455 (0.0403)    Prec 99.219% (98.611%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.4327 (0.4327)    Prec 90.625%
(90.625%)
 * Prec 89.790%
Epoch: [192][0/391]     Time 0.250 (0.250)      Data 0.215 (0.215)      Loss
0.0339 (0.0339)    Prec 98.438% (98.438%)
Epoch: [192][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0784 (0.0362)    Prec 96.094% (98.731%)
Epoch: [192][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0368 (0.0353)    Prec 99.219% (98.799%)
Epoch: [192][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0219 (0.0359)    Prec 99.219% (98.775%)
Test: [0/79]    Time 0.208 (0.208)      Loss 0.2946 (0.2946)    Prec 92.188%
(92.188%)
 * Prec 89.340%
Epoch: [193][0/391]     Time 0.313 (0.313)      Data 0.252 (0.252)      Loss
0.0261 (0.0261)    Prec 99.219% (99.219%)
Epoch: [193][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0193 (0.0385)    Prec 99.219% (98.708%)
Epoch: [193][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0061 (0.0391)    Prec 100.000% (98.647%)
Epoch: [193][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.1108 (0.0384)    Prec 95.312% (98.653%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.3422 (0.3422)    Prec 91.406%
(91.406%)
 * Prec 89.650%
Epoch: [194][0/391]     Time 0.255 (0.255)      Data 0.220 (0.220)      Loss
0.0236 (0.0236)    Prec 99.219% (99.219%)
Epoch: [194][100/391]   Time 0.049 (0.051)      Data 0.002 (0.003)      Loss
0.0470 (0.0374)    Prec 99.219% (98.724%)
Epoch: [194][200/391]   Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0150 (0.0428)    Prec 100.000% (98.496%)
Epoch: [194][300/391]   Time 0.051 (0.050)      Data 0.002 (0.002)      Loss
0.0276 (0.0418)    Prec 99.219% (98.557%)
Test: [0/79]    Time 0.135 (0.135)      Loss 0.2475 (0.2475)    Prec 92.188%
(92.188%)
 * Prec 89.710%
Epoch: [195][0/391]     Time 0.263 (0.263)      Data 0.221 (0.221)      Loss
0.1075 (0.1075)    Prec 96.875% (96.875%)
Epoch: [195][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0643 (0.0410)    Prec 98.438% (98.600%)
Epoch: [195][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0037 (0.0409)    Prec 100.000% (98.612%)
Epoch: [195][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.0582 (0.0393)    Prec 97.656% (98.666%)
Test: [0/79]    Time 0.160 (0.160)      Loss 0.4354 (0.4354)    Prec 92.188%
(92.188%)
 * Prec 89.330%
Epoch: [196][0/391]     Time 0.299 (0.299)      Data 0.227 (0.227)      Loss
0.0334 (0.0334)    Prec 99.219% (99.219%)
Epoch: [196][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0206 (0.0369)    Prec 98.438% (98.646%)
Epoch: [196][200/391]   Time 0.049 (0.050)      Data 0.001 (0.003)      Loss
0.0294 (0.0406)    Prec 98.438% (98.585%)
Epoch: [196][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0326 (0.0403)    Prec 98.438% (98.591%)
Test: [0/79]    Time 0.218 (0.218)      Loss 0.4638 (0.4638)    Prec 90.625%
(90.625%)
 * Prec 89.130%
Epoch: [197][0/391]     Time 0.353 (0.353)      Data 0.320 (0.320)      Loss
0.0389 (0.0389)    Prec 98.438% (98.438%)
Epoch: [197][100/391]   Time 0.049 (0.053)      Data 0.001 (0.005)      Loss
0.0133 (0.0393)    Prec 100.000% (98.631%)
Epoch: [197][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0119 (0.0388)    Prec 100.000% (98.667%)
Epoch: [197][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0211 (0.0379)    Prec 99.219% (98.661%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.3477 (0.3477)    Prec 91.406%
(91.406%)
 * Prec 89.410%
Epoch: [198][0/391]     Time 0.251 (0.251)      Data 0.218 (0.218)      Loss
0.0072 (0.0072)    Prec 100.000% (100.000%)
Epoch: [198][100/391]   Time 0.048 (0.051)      Data 0.001 (0.003)      Loss
0.0215 (0.0375)    Prec 100.000% (98.631%)
Epoch: [198][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0073 (0.0361)    Prec 100.000% (98.745%)
Epoch: [198][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0788 (0.0384)    Prec 96.875% (98.666%)
Test: [0/79]    Time 0.148 (0.148)      Loss 0.4304 (0.4304)    Prec 89.844%
(89.844%)
 * Prec 90.140%
Epoch: [199][0/391]     Time 0.257 (0.257)      Data 0.220 (0.220)      Loss
0.0274 (0.0274)    Prec 99.219% (99.219%)
Epoch: [199][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0176 (0.0328)    Prec 99.219% (98.840%)
Epoch: [199][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0229 (0.0343)    Prec 99.219% (98.783%)
Epoch: [199][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0260 (0.0356)    Prec 99.219% (98.757%)
Test: [0/79]    Time 0.218 (0.218)      Loss 0.4034 (0.4034)    Prec 92.969%
(92.969%)
 * Prec 89.660%
```

```
Epoch: [200][0/391]     Time 0.290 (0.290)      Data 0.251 (0.251)      Loss
0.0532 (0.0532)    Prec 97.656% (97.656%)
Epoch: [200][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.1000 (0.0374)    Prec 97.656% (98.693%)
Epoch: [200][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0533 (0.0358)    Prec 97.656% (98.776%)
Epoch: [200][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0142 (0.0361)    Prec 100.000% (98.749%)
Test: [0/79]    Time 0.170 (0.170)      Loss 0.3544 (0.3544)    Prec 91.406%
(91.406%)
 * Prec 89.420%
Epoch: [201][0/391]     Time 0.259 (0.259)      Data 0.219 (0.219)      Loss
0.0361 (0.0361)    Prec 98.438% (98.438%)
Epoch: [201][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0393 (0.0432)    Prec 97.656% (98.484%)
Epoch: [201][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.1495 (0.0418)    Prec 95.312% (98.472%)
Epoch: [201][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0476 (0.0402)    Prec 99.219% (98.554%)
Test: [0/79]    Time 0.140 (0.140)      Loss 0.3450 (0.3450)    Prec 92.188%
(92.188%)
 * Prec 89.900%
Epoch: [202][0/391]     Time 0.261 (0.261)      Data 0.221 (0.221)      Loss
0.0291 (0.0291)    Prec 99.219% (99.219%)
Epoch: [202][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0477 (0.0356)    Prec 98.438% (98.755%)
Epoch: [202][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0212 (0.0350)    Prec 99.219% (98.783%)
Epoch: [202][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0292 (0.0356)    Prec 97.656% (98.759%)
Test: [0/79]    Time 0.110 (0.110)      Loss 0.3433 (0.3433)    Prec 92.969%
(92.969%)
 * Prec 89.510%
Epoch: [203][0/391]     Time 0.315 (0.315)      Data 0.276 (0.276)      Loss
0.0205 (0.0205)    Prec 99.219% (99.219%)
Epoch: [203][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0176 (0.0343)    Prec 99.219% (98.809%)
Epoch: [203][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0188 (0.0335)    Prec 100.000% (98.861%)
Epoch: [203][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0713 (0.0354)    Prec 98.438% (98.824%)
Test: [0/79]    Time 0.224 (0.224)      Loss 0.3774 (0.3774)    Prec 90.625%
(90.625%)
 * Prec 89.310%
Epoch: [204][0/391]     Time 0.293 (0.293)      Data 0.253 (0.253)      Loss
0.0621 (0.0621)    Prec 97.656% (97.656%)
Epoch: [204][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0788 (0.0340)    Prec 97.656% (98.878%)
```

```
Epoch: [204][200/391]   Time 0.051 (0.051)      Data 0.001 (0.002)      Loss
0.0620 (0.0365)    Prec 97.656% (98.756%)
Epoch: [204][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0106 (0.0365)    Prec 100.000% (98.744%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.3194 (0.3194)    Prec 91.406%
(91.406%)
 * Prec 89.820%
Epoch: [205][0/391]     Time 0.236 (0.236)      Data 0.204 (0.204)      Loss
0.0519 (0.0519)    Prec 99.219% (99.219%)
Epoch: [205][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0287 (0.0394)    Prec 98.438% (98.623%)
Epoch: [205][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0343 (0.0357)    Prec 97.656% (98.748%)
Epoch: [205][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0060 (0.0369)    Prec 100.000% (98.702%)
Test: [0/79]    Time 0.240 (0.240)      Loss 0.4554 (0.4554)    Prec 88.281%
(88.281%)
 * Prec 88.560%
Epoch: [206][0/391]     Time 0.322 (0.322)      Data 0.251 (0.251)      Loss
0.0465 (0.0465)    Prec 99.219% (99.219%)
Epoch: [206][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0205 (0.0364)    Prec 99.219% (98.724%)
Epoch: [206][200/391]   Time 0.051 (0.051)      Data 0.001 (0.002)      Loss
0.0220 (0.0349)    Prec 99.219% (98.783%)
Epoch: [206][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0433 (0.0374)    Prec 97.656% (98.733%)
Test: [0/79]    Time 0.266 (0.266)      Loss 0.3437 (0.3437)    Prec 91.406%
(91.406%)
 * Prec 90.090%
Epoch: [207][0/391]     Time 0.265 (0.265)      Data 0.226 (0.226)      Loss
0.0178 (0.0178)    Prec 99.219% (99.219%)
Epoch: [207][100/391]   Time 0.046 (0.051)      Data 0.001 (0.003)      Loss
0.0178 (0.0314)    Prec 99.219% (98.878%)
Epoch: [207][200/391]   Time 0.046 (0.050)      Data 0.001 (0.002)      Loss
0.0051 (0.0347)    Prec 100.000% (98.756%)
Epoch: [207][300/391]   Time 0.046 (0.050)      Data 0.001 (0.002)      Loss
0.0126 (0.0359)    Prec 100.000% (98.707%)
Test: [0/79]    Time 0.161 (0.161)      Loss 0.3215 (0.3215)    Prec 91.406%
(91.406%)
 * Prec 90.070%
Epoch: [208][0/391]     Time 0.396 (0.396)      Data 0.321 (0.321)      Loss
0.0586 (0.0586)    Prec 98.438% (98.438%)
Epoch: [208][100/391]   Time 0.049 (0.053)      Data 0.001 (0.004)      Loss
0.0210 (0.0383)    Prec 98.438% (98.685%)
Epoch: [208][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0257 (0.0352)    Prec 98.438% (98.791%)
Epoch: [208][300/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0236 (0.0361)    Prec 100.000% (98.749%)
```

```
Test: [0/79]    Time 0.163 (0.163)      Loss 0.3174 (0.3174)    Prec 92.969%
(92.969%)
 * Prec 89.560%
Epoch: [209][0/391]     Time 0.163 (0.163)      Data 0.132 (0.132)      Loss
0.0424 (0.0424)    Prec 97.656% (97.656%)
Epoch: [209][100/391]   Time 0.052 (0.051)      Data 0.001 (0.003)      Loss
0.0740 (0.0350)    Prec 98.438% (98.739%)
Epoch: [209][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0252 (0.0354)    Prec 98.438% (98.752%)
Epoch: [209][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0390 (0.0365)    Prec 99.219% (98.739%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.3638 (0.3638)    Prec 93.750%
(93.750%)
 * Prec 89.020%
Epoch: [210][0/391]     Time 0.259 (0.259)      Data 0.219 (0.219)      Loss
0.0184 (0.0184)    Prec 99.219% (99.219%)
Epoch: [210][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0234 (0.0324)    Prec 99.219% (98.878%)
Epoch: [210][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0480 (0.0326)    Prec 97.656% (98.912%)
Epoch: [210][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0591 (0.0332)    Prec 98.438% (98.912%)
Test: [0/79]    Time 0.156 (0.156)      Loss 0.3582 (0.3582)    Prec 91.406%
(91.406%)
 * Prec 89.770%
Epoch: [211][0/391]     Time 0.211 (0.211)      Data 0.134 (0.134)      Loss
0.1134 (0.1134)    Prec 97.656% (97.656%)
Epoch: [211][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0290 (0.0339)    Prec 99.219% (98.940%)
Epoch: [211][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0189 (0.0366)    Prec 99.219% (98.787%)
Epoch: [211][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0422 (0.0363)    Prec 98.438% (98.793%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.3625 (0.3625)    Prec 93.750%
(93.750%)
 * Prec 89.790%
Epoch: [212][0/391]     Time 0.355 (0.355)      Data 0.281 (0.281)      Loss
0.0448 (0.0448)    Prec 99.219% (99.219%)
Epoch: [212][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0081 (0.0323)    Prec 100.000% (98.863%)
Epoch: [212][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0128 (0.0351)    Prec 99.219% (98.834%)
Epoch: [212][300/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0171 (0.0361)    Prec 99.219% (98.778%)
Test: [0/79]    Time 0.169 (0.169)      Loss 0.3407 (0.3407)    Prec 94.531%
(94.531%)
 * Prec 89.650%
Epoch: [213][0/391]     Time 0.263 (0.263)      Data 0.223 (0.223)      Loss
```

```
0.0311 (0.0311)    Prec 98.438% (98.438%)
Epoch: [213][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0325 (0.0410)    Prec 98.438% (98.631%)
Epoch: [213][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0656 (0.0392)    Prec 98.438% (98.686%)
Epoch: [213][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0625 (0.0394)    Prec 97.656% (98.671%)
Test: [0/79]    Time 0.241 (0.241)      Loss 0.4279 (0.4279)    Prec 91.406%
(91.406%)
 * Prec 89.390%
Epoch: [214][0/391]     Time 0.282 (0.282)      Data 0.250 (0.250)      Loss
0.0403 (0.0403)    Prec 98.438% (98.438%)
Epoch: [214][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0109 (0.0309)    Prec 100.000% (98.940%)
Epoch: [214][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0325 (0.0308)    Prec 98.438% (98.923%)
Epoch: [214][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0893 (0.0334)    Prec 98.438% (98.853%)
Test: [0/79]    Time 0.165 (0.165)      Loss 0.2739 (0.2739)    Prec 92.969%
(92.969%)
 * Prec 89.750%
Epoch: [215][0/391]     Time 0.257 (0.257)      Data 0.222 (0.222)      Loss
0.0180 (0.0180)    Prec 100.000% (100.000%)
Epoch: [215][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0735 (0.0320)    Prec 98.438% (98.940%)
Epoch: [215][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.1013 (0.0330)    Prec 96.094% (98.892%)
Epoch: [215][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0682 (0.0345)    Prec 97.656% (98.814%)
Test: [0/79]    Time 0.266 (0.266)      Loss 0.4088 (0.4088)    Prec 93.750%
(93.750%)
 * Prec 89.450%
Epoch: [216][0/391]     Time 0.264 (0.264)      Data 0.224 (0.224)      Loss
0.0199 (0.0199)    Prec 99.219% (99.219%)
Epoch: [216][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0339 (0.0325)    Prec 98.438% (98.956%)
Epoch: [216][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0076 (0.0370)    Prec 100.000% (98.748%)
Epoch: [216][300/391]   Time 0.046 (0.050)      Data 0.001 (0.002)      Loss
0.1381 (0.0372)    Prec 96.875% (98.772%)
Test: [0/79]    Time 0.113 (0.113)      Loss 0.2960 (0.2960)    Prec 92.969%
(92.969%)
 * Prec 90.070%
Epoch: [217][0/391]     Time 0.302 (0.302)      Data 0.261 (0.261)      Loss
0.0926 (0.0926)    Prec 97.656% (97.656%)
Epoch: [217][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0405 (0.0350)    Prec 98.438% (98.739%)
Epoch: [217][200/391]   Time 0.050 (0.051)      Data 0.002 (0.003)      Loss
```

```
0.0227 (0.0363)    Prec 99.219% (98.772%)
Epoch: [217][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0264 (0.0360)    Prec 99.219% (98.793%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.3244 (0.3244)    Prec 92.188%
(92.188%)
 * Prec 89.540%
Epoch: [218][0/391]     Time 0.356 (0.356)      Data 0.323 (0.323)      Loss
0.0125 (0.0125)    Prec 99.219% (99.219%)
Epoch: [218][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0322 (0.0336)    Prec 98.438% (98.855%)
Epoch: [218][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0127 (0.0324)    Prec 99.219% (98.900%)
Epoch: [218][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0171 (0.0333)    Prec 99.219% (98.855%)
Test: [0/79]    Time 0.262 (0.262)      Loss 0.2557 (0.2557)    Prec 92.188%
(92.188%)
 * Prec 89.500%
Epoch: [219][0/391]     Time 0.307 (0.307)      Data 0.231 (0.231)      Loss
0.0337 (0.0337)    Prec 98.438% (98.438%)
Epoch: [219][100/391]   Time 0.049 (0.052)      Data 0.002 (0.004)      Loss
0.0908 (0.0356)    Prec 97.656% (98.778%)
Epoch: [219][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0303 (0.0352)    Prec 99.219% (98.756%)
Epoch: [219][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0153 (0.0348)    Prec 99.219% (98.788%)
Test: [0/79]    Time 0.116 (0.116)      Loss 0.4020 (0.4020)    Prec 92.188%
(92.188%)
 * Prec 89.480%
Epoch: [220][0/391]     Time 0.306 (0.306)      Data 0.273 (0.273)      Loss
0.0169 (0.0169)    Prec 99.219% (99.219%)
Epoch: [220][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0239 (0.0368)    Prec 99.219% (98.778%)
Epoch: [220][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0195 (0.0348)    Prec 99.219% (98.780%)
Epoch: [220][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0082 (0.0351)    Prec 100.000% (98.749%)
Test: [0/79]    Time 0.170 (0.170)      Loss 0.3153 (0.3153)    Prec 92.188%
(92.188%)
 * Prec 89.450%
Epoch: [221][0/391]     Time 0.258 (0.258)      Data 0.227 (0.227)      Loss
0.0699 (0.0699)    Prec 98.438% (98.438%)
Epoch: [221][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0484 (0.0319)    Prec 99.219% (98.979%)
Epoch: [221][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0524 (0.0350)    Prec 97.656% (98.850%)
Epoch: [221][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0254 (0.0356)    Prec 98.438% (98.814%)
Test: [0/79]    Time 0.151 (0.151)      Loss 0.3297 (0.3297)    Prec 92.969%
```

```
(92.969%)
 * Prec 89.890%
Epoch: [222][0/391]     Time 0.314 (0.314)      Data 0.274 (0.274)      Loss
0.0358 (0.0358)    Prec 99.219% (99.219%)
Epoch: [222][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0208 (0.0334)    Prec 99.219% (98.871%)
Epoch: [222][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0760 (0.0348)    Prec 96.875% (98.834%)
Epoch: [222][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0081 (0.0353)    Prec 100.000% (98.835%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.3857 (0.3857)    Prec 91.406%
(91.406%)
 * Prec 89.860%
Epoch: [223][0/391]     Time 0.294 (0.294)      Data 0.218 (0.218)      Loss
0.0362 (0.0362)    Prec 97.656% (97.656%)
Epoch: [223][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0096 (0.0376)    Prec 99.219% (98.700%)
Epoch: [223][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0392 (0.0372)    Prec 97.656% (98.717%)
Epoch: [223][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0218 (0.0361)    Prec 100.000% (98.762%)
Test: [0/79]    Time 0.251 (0.251)      Loss 0.2721 (0.2721)    Prec 91.406%
(91.406%)
 * Prec 90.000%
Epoch: [224][0/391]     Time 0.253 (0.253)      Data 0.220 (0.220)      Loss
0.0110 (0.0110)    Prec 100.000% (100.000%)
Epoch: [224][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0448 (0.0294)    Prec 97.656% (99.064%)
Epoch: [224][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0224 (0.0302)    Prec 99.219% (99.001%)
Epoch: [224][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0159 (0.0325)    Prec 100.000% (98.928%)
Test: [0/79]    Time 0.150 (0.150)      Loss 0.4097 (0.4097)    Prec 89.844%
(89.844%)
 * Prec 89.390%
Epoch: [225][0/391]     Time 0.263 (0.263)      Data 0.229 (0.229)      Loss
0.0159 (0.0159)    Prec 99.219% (99.219%)
Epoch: [225][100/391]   Time 0.050 (0.052)      Data 0.002 (0.003)      Loss
0.0910 (0.0336)    Prec 99.219% (98.863%)
Epoch: [225][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0725 (0.0315)    Prec 98.438% (98.958%)
Epoch: [225][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0159 (0.0323)    Prec 99.219% (98.933%)
Test: [0/79]    Time 0.155 (0.155)      Loss 0.3042 (0.3042)    Prec 91.406%
(91.406%)
 * Prec 89.390%
Epoch: [226][0/391]     Time 0.259 (0.259)      Data 0.227 (0.227)      Loss
0.0092 (0.0092)    Prec 100.000% (100.000%)
```

```
Epoch: [226][100/391]   Time 0.053 (0.052)      Data 0.001 (0.004)      Loss
0.0160 (0.0291)    Prec 99.219% (98.979%)
Epoch: [226][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0579 (0.0330)    Prec 98.438% (98.865%)
Epoch: [226][300/391]   Time 0.045 (0.050)      Data 0.001 (0.002)      Loss
0.0900 (0.0336)    Prec 97.656% (98.835%)
Test: [0/79]    Time 0.157 (0.157)      Loss 0.2663 (0.2663)    Prec 92.969%
(92.969%)
 * Prec 89.100%
Epoch: [227][0/391]     Time 0.251 (0.251)      Data 0.211 (0.211)      Loss
0.0175 (0.0175)    Prec 99.219% (99.219%)
Epoch: [227][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0841 (0.0361)    Prec 96.875% (98.693%)
Epoch: [227][200/391]   Time 0.049 (0.050)      Data 0.002 (0.002)      Loss
0.0455 (0.0322)    Prec 98.438% (98.807%)
Epoch: [227][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0195 (0.0332)    Prec 98.438% (98.770%)
Test: [0/79]    Time 0.146 (0.146)      Loss 0.3347 (0.3347)    Prec 91.406%
(91.406%)
 * Prec 89.750%
Epoch: [228][0/391]     Time 0.299 (0.299)      Data 0.231 (0.231)      Loss
0.0389 (0.0389)    Prec 97.656% (97.656%)
Epoch: [228][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0665 (0.0376)    Prec 97.656% (98.762%)
Epoch: [228][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0964 (0.0353)    Prec 97.656% (98.842%)
Epoch: [228][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0545 (0.0371)    Prec 98.438% (98.741%)
Test: [0/79]    Time 0.168 (0.168)      Loss 0.3797 (0.3797)    Prec 91.406%
(91.406%)
 * Prec 89.720%
Epoch: [229][0/391]     Time 0.252 (0.252)      Data 0.220 (0.220)      Loss
0.0250 (0.0250)    Prec 99.219% (99.219%)
Epoch: [229][100/391]   Time 0.052 (0.051)      Data 0.001 (0.003)      Loss
0.0515 (0.0383)    Prec 98.438% (98.646%)
Epoch: [229][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0626 (0.0379)    Prec 97.656% (98.721%)
Epoch: [229][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0154 (0.0369)    Prec 100.000% (98.731%)
Test: [0/79]    Time 0.242 (0.242)      Loss 0.3360 (0.3360)    Prec 93.750%
(93.750%)
 * Prec 90.240%
Epoch: [230][0/391]     Time 0.266 (0.266)      Data 0.234 (0.234)      Loss
0.0170 (0.0170)    Prec 100.000% (100.000%)
Epoch: [230][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0115 (0.0323)    Prec 99.219% (98.917%)
Epoch: [230][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0612 (0.0317)    Prec 97.656% (98.947%)
```

```
Epoch: [230][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0283 (0.0339)    Prec 99.219% (98.848%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.4076 (0.4076)    Prec 91.406%
(91.406%)
 * Prec 89.690%
Epoch: [231][0/391]     Time 0.250 (0.250)      Data 0.218 (0.218)      Loss
0.0504 (0.0504)    Prec 98.438% (98.438%)
Epoch: [231][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0114 (0.0395)    Prec 100.000% (98.623%)
Epoch: [231][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0056 (0.0359)    Prec 100.000% (98.799%)
Epoch: [231][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0178 (0.0366)    Prec 100.000% (98.754%)
Test: [0/79]    Time 0.163 (0.163)      Loss 0.3088 (0.3088)    Prec 92.969%
(92.969%)
 * Prec 89.860%
Epoch: [232][0/391]     Time 0.340 (0.340)      Data 0.308 (0.308)      Loss
0.0081 (0.0081)    Prec 100.000% (100.000%)
Epoch: [232][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0035 (0.0327)    Prec 100.000% (98.933%)
Epoch: [232][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0372 (0.0336)    Prec 99.219% (98.877%)
Epoch: [232][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0350 (0.0355)    Prec 98.438% (98.806%)
Test: [0/79]    Time 0.215 (0.215)      Loss 0.3146 (0.3146)    Prec 93.750%
(93.750%)
 * Prec 89.960%
Epoch: [233][0/391]     Time 0.282 (0.282)      Data 0.249 (0.249)      Loss
0.0104 (0.0104)    Prec 100.000% (100.000%)
Epoch: [233][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0591 (0.0353)    Prec 98.438% (98.778%)
Epoch: [233][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0281 (0.0354)    Prec 99.219% (98.834%)
Epoch: [233][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0227 (0.0366)    Prec 99.219% (98.783%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3522 (0.3522)    Prec 89.844%
(89.844%)
 * Prec 89.350%
Epoch: [234][0/391]     Time 0.265 (0.265)      Data 0.225 (0.225)      Loss
0.0759 (0.0759)    Prec 96.094% (96.094%)
Epoch: [234][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0234 (0.0313)    Prec 98.438% (98.909%)
Epoch: [234][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0134 (0.0314)    Prec 99.219% (98.912%)
Epoch: [234][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0276 (0.0328)    Prec 98.438% (98.863%)
Test: [0/79]    Time 0.223 (0.223)      Loss 0.4002 (0.4002)    Prec 92.188%
(92.188%)
```

```
 * Prec 89.470%
Epoch: [235][0/391]     Time 0.288 (0.288)      Data 0.256 (0.256)      Loss
0.0283 (0.0283)    Prec 99.219% (99.219%)
Epoch: [235][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
0.0165 (0.0376)    Prec 100.000% (98.708%)
Epoch: [235][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0141 (0.0365)    Prec 99.219% (98.737%)
Epoch: [235][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0070 (0.0354)    Prec 100.000% (98.770%)
Test: [0/79]    Time 0.167 (0.167)      Loss 0.4109 (0.4109)    Prec 92.188%
(92.188%)
 * Prec 89.300%
Epoch: [236][0/391]     Time 0.263 (0.263)      Data 0.223 (0.223)      Loss
0.0152 (0.0152)    Prec 99.219% (99.219%)
Epoch: [236][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0245 (0.0347)    Prec 99.219% (98.786%)
Epoch: [236][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0150 (0.0351)    Prec 100.000% (98.791%)
Epoch: [236][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0297 (0.0354)    Prec 99.219% (98.788%)
Test: [0/79]    Time 0.172 (0.172)      Loss 0.3612 (0.3612)    Prec 92.969%
(92.969%)
 * Prec 89.840%
Epoch: [237][0/391]     Time 0.283 (0.283)      Data 0.194 (0.194)      Loss
0.0245 (0.0245)    Prec 98.438% (98.438%)
Epoch: [237][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0124 (0.0337)    Prec 99.219% (98.778%)
Epoch: [237][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0111 (0.0319)    Prec 99.219% (98.884%)
Epoch: [237][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0710 (0.0325)    Prec 97.656% (98.879%)
Test: [0/79]    Time 0.169 (0.169)      Loss 0.4108 (0.4108)    Prec 89.062%
(89.062%)
 * Prec 89.870%
Epoch: [238][0/391]     Time 0.246 (0.246)      Data 0.211 (0.211)      Loss
0.0248 (0.0248)    Prec 99.219% (99.219%)
Epoch: [238][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0101 (0.0283)    Prec 100.000% (99.041%)
Epoch: [238][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0351 (0.0313)    Prec 99.219% (98.904%)
Epoch: [238][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0043 (0.0324)    Prec 100.000% (98.905%)
Test: [0/79]    Time 0.140 (0.140)      Loss 0.3038 (0.3038)    Prec 91.406%
(91.406%)
 * Prec 89.470%
Epoch: [239][0/391]     Time 0.285 (0.285)      Data 0.253 (0.253)      Loss
0.0194 (0.0194)    Prec 98.438% (98.438%)
Epoch: [239][100/391]   Time 0.049 (0.052)      Data 0.001 (0.004)      Loss
```

```
0.0242 (0.0332)    Prec 99.219% (98.863%)
Epoch: [239][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0935 (0.0344)    Prec 96.094% (98.811%)
Epoch: [239][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0656 (0.0322)    Prec 98.438% (98.876%)
Test: [0/79]    Time 0.218 (0.218)      Loss 0.2853 (0.2853)    Prec 92.969%
(92.969%)
 * Prec 89.710%
Epoch: [240][0/391]     Time 0.280 (0.280)      Data 0.239 (0.239)      Loss
0.0493 (0.0493)    Prec 98.438% (98.438%)
Epoch: [240][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0103 (0.0353)    Prec 100.000% (98.832%)
Epoch: [240][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0174 (0.0315)    Prec 99.219% (98.923%)
Epoch: [240][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0163 (0.0337)    Prec 99.219% (98.881%)
Test: [0/79]    Time 0.166 (0.166)      Loss 0.4872 (0.4872)    Prec 89.062%
(89.062%)
 * Prec 89.870%
Epoch: [241][0/391]     Time 0.406 (0.406)      Data 0.366 (0.366)      Loss
0.0375 (0.0375)    Prec 98.438% (98.438%)
Epoch: [241][100/391]   Time 0.049 (0.053)      Data 0.001 (0.005)      Loss
0.0583 (0.0314)    Prec 98.438% (99.002%)
Epoch: [241][200/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0305 (0.0333)    Prec 99.219% (98.861%)
Epoch: [241][300/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0353 (0.0333)    Prec 98.438% (98.845%)
Test: [0/79]    Time 0.179 (0.179)      Loss 0.3753 (0.3753)    Prec 89.844%
(89.844%)
 * Prec 89.940%
Epoch: [242][0/391]     Time 0.261 (0.261)      Data 0.221 (0.221)      Loss
0.0354 (0.0354)    Prec 99.219% (99.219%)
Epoch: [242][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0383 (0.0283)    Prec 98.438% (99.064%)
Epoch: [242][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0247 (0.0313)    Prec 99.219% (98.916%)
Epoch: [242][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0391 (0.0322)    Prec 98.438% (98.925%)
Test: [0/79]    Time 0.164 (0.164)      Loss 0.3760 (0.3760)    Prec 92.188%
(92.188%)
 * Prec 89.270%
Epoch: [243][0/391]     Time 0.217 (0.217)      Data 0.179 (0.179)      Loss
0.0082 (0.0082)    Prec 100.000% (100.000%)
Epoch: [243][100/391]   Time 0.050 (0.051)      Data 0.001 (0.003)      Loss
0.0484 (0.0325)    Prec 99.219% (98.909%)
Epoch: [243][200/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0674 (0.0311)    Prec 98.438% (98.954%)
Epoch: [243][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
```

```
0.0052 (0.0298)    Prec 100.000% (98.983%)
Test: [0/79]    Time 0.205 (0.205)      Loss 0.4114 (0.4114)    Prec 90.625%
(90.625%)
 * Prec 89.750%
Epoch: [244][0/391]     Time 0.384 (0.384)      Data 0.352 (0.352)      Loss
0.0280 (0.0280)    Prec 99.219% (99.219%)
Epoch: [244][100/391]   Time 0.050 (0.053)      Data 0.001 (0.005)      Loss
0.0268 (0.0392)    Prec 98.438% (98.739%)
Epoch: [244][200/391]   Time 0.049 (0.051)      Data 0.001 (0.003)      Loss
0.0241 (0.0351)    Prec 99.219% (98.850%)
Epoch: [244][300/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0036 (0.0351)    Prec 100.000% (98.837%)
Test: [0/79]    Time 0.171 (0.171)      Loss 0.4192 (0.4192)    Prec 91.406%
(91.406%)
 * Prec 89.460%
Epoch: [245][0/391]     Time 0.261 (0.261)      Data 0.221 (0.221)      Loss
0.0351 (0.0351)    Prec 99.219% (99.219%)
Epoch: [245][100/391]   Time 0.050 (0.052)      Data 0.001 (0.003)      Loss
0.0239 (0.0322)    Prec 99.219% (98.855%)
Epoch: [245][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0789 (0.0333)    Prec 97.656% (98.826%)
Epoch: [245][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0065 (0.0333)    Prec 100.000% (98.829%)
Test: [0/79]    Time 0.271 (0.271)      Loss 0.3174 (0.3174)    Prec 90.625%
(90.625%)
 * Prec 90.040%
Epoch: [246][0/391]     Time 0.347 (0.347)      Data 0.315 (0.315)      Loss
0.0403 (0.0403)    Prec 98.438% (98.438%)
Epoch: [246][100/391]   Time 0.042 (0.052)      Data 0.001 (0.004)      Loss
0.0266 (0.0363)    Prec 99.219% (98.762%)
Epoch: [246][200/391]   Time 0.042 (0.051)      Data 0.001 (0.003)      Loss
0.0113 (0.0365)    Prec 100.000% (98.741%)
Epoch: [246][300/391]   Time 0.042 (0.050)      Data 0.001 (0.002)      Loss
0.0421 (0.0368)    Prec 98.438% (98.718%)
Test: [0/79]    Time 0.242 (0.242)      Loss 0.3806 (0.3806)    Prec 88.281%
(88.281%)
 * Prec 89.680%
Epoch: [247][0/391]     Time 0.268 (0.268)      Data 0.231 (0.231)      Loss
0.0090 (0.0090)    Prec 100.000% (100.000%)
Epoch: [247][100/391]   Time 0.050 (0.052)      Data 0.001 (0.004)      Loss
0.0292 (0.0317)    Prec 99.219% (98.886%)
Epoch: [247][200/391]   Time 0.049 (0.051)      Data 0.001 (0.002)      Loss
0.0206 (0.0323)    Prec 98.438% (98.881%)
Epoch: [247][300/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0080 (0.0327)    Prec 100.000% (98.905%)
Test: [0/79]    Time 0.150 (0.150)      Loss 0.2417 (0.2417)    Prec 93.750%
(93.750%)
 * Prec 89.410%
```

```
Epoch: [248][0/391]     Time 0.255 (0.255)      Data 0.224 (0.224)      Loss
0.0555 (0.0555)    Prec 97.656% (97.656%)
Epoch: [248][100/391]   Time 0.049 (0.051)      Data 0.001 (0.004)      Loss
0.0466 (0.0352)    Prec 97.656% (98.793%)
Epoch: [248][200/391]   Time 0.050 (0.050)      Data 0.001 (0.002)      Loss
0.0278 (0.0350)    Prec 99.219% (98.768%)
Epoch: [248][300/391]   Time 0.048 (0.050)      Data 0.001 (0.002)      Loss
0.0065 (0.0355)    Prec 100.000% (98.793%)
Test: [0/79]    Time 0.158 (0.158)      Loss 0.3375 (0.3375)    Prec 92.969%
(92.969%)
 * Prec 89.750%
Epoch: [249][0/391]     Time 0.274 (0.274)      Data 0.198 (0.198)      Loss
0.0043 (0.0043)    Prec 100.000% (100.000%)
Epoch: [249][100/391]   Time 0.049 (0.052)      Data 0.001 (0.003)      Loss
0.0071 (0.0320)    Prec 100.000% (98.956%)
Epoch: [249][200/391]   Time 0.050 (0.051)      Data 0.001 (0.002)      Loss
0.0842 (0.0317)    Prec 97.656% (98.916%)
Epoch: [249][300/391]   Time 0.049 (0.050)      Data 0.001 (0.002)      Loss
0.0224 (0.0305)    Prec 97.656% (98.931%)
Test: [0/79]    Time 0.169 (0.169)      Loss 0.2953 (0.2953)    Prec 92.969%
(92.969%)
 * Prec 89.710%
Best accuracy after finetuning the structured-pruned model: 90.240%
Saved pruned + fine-tuned model to:
result/VGG16_quant/model_pruned_finetuned_pt1.pth.tar
```

[45]:
```python
PATH = "result/VGG16_quant/model_pruned_finetuned_pt1.pth.tar"
checkpoint = torch.load(PATH)
model.load_state_dict(checkpoint['state_dict'])
device = torch.device("cuda")

model.cuda()

def compute_sparsity(model):
    total_zeros = 0
    total_params = 0
    layer_sparsity = {}

    print("\n----- Per-layer sparsity (zeros %) -----")

    for name, module in model.named_modules():
        if isinstance(module, QuantConv2d):
            W = module.weight.detach().cpu()
            zeros = torch.sum(W == 0).item()
            params = W.numel()

            sparsity = 100 * zeros / params
```

```
            layer_sparsity[name] = sparsity
            print(f"{name}: {sparsity:.2f}% sparse")

            total_zeros += zeros
            total_params += params

    total_sparsity = 100 * total_zeros / total_params
    print("\n----- Total model sparsity -----")
    print(f"Overall sparsity: {total_sparsity:.2f}%")

    return layer_sparsity, total_sparsity

compute_sparsity(model)
```

```
----- Per-layer sparsity (zeros %) -----
features.0: 50.00% sparse
features.3: 50.00% sparse
features.7: 50.00% sparse
features.10: 50.00% sparse
features.14: 50.00% sparse
features.17: 50.00% sparse
features.20: 50.00% sparse
features.24: 50.00% sparse
features.27: 50.00% sparse
features.29: 50.00% sparse
features.31: 50.00% sparse
features.33: 50.00% sparse
features.37: 50.00% sparse
features.40: 50.00% sparse
features.43: 50.00% sparse

----- Total model sparsity -----
Overall sparsity: 50.00%
```

[45]:
```
({'features.0': 50.0,
  'features.3': 50.0,
  'features.7': 50.0,
  'features.10': 50.0,
  'features.14': 50.0,
  'features.17': 50.0,
  'features.20': 50.0,
  'features.24': 50.0,
  'features.27': 50.0,
  'features.29': 50.0,
  'features.31': 50.0,
  'features.33': 50.0,
```

```
  'features.37': 50.0,
  'features.40': 50.0,
  'features.43': 50.0},
 50.0)
```

[22]:
```python
def compute_structured_filter_sparsity(model):
    print("\n----- Structured Filter Sparsity -----")
    for name, module in model.named_modules():
        if isinstance(module, QuantConv2d):
            W = module.weight.detach().cpu()    # shape: [C_out, C_in, k, k]
            C_out = W.shape[0]

            # A filter is dead if *all weights* in that channel = 0
            dead_filters = 0
            for c in range(C_out):
                if torch.sum(W[c]) == 0:
                    dead_filters += 1

            sparsity = 100 * dead_filters / C_out
            print(f"{name}: {dead_filters}/{C_out} filters removed ({sparsity:.
 ↪2f}%)")
compute_structured_filter_sparsity(model)
```

```
----- Structured Filter Sparsity -----
features.0: 32/64 filters removed (50.00%)
features.3: 32/64 filters removed (50.00%)
features.7: 64/128 filters removed (50.00%)
features.10: 64/128 filters removed (50.00%)
features.14: 128/256 filters removed (50.00%)
features.17: 128/256 filters removed (50.00%)
features.20: 128/256 filters removed (50.00%)
features.24: 256/512 filters removed (50.00%)
features.27: 4/8 filters removed (50.00%)
features.29: 4/8 filters removed (50.00%)
features.31: 256/512 filters removed (50.00%)
features.33: 256/512 filters removed (50.00%)
features.37: 256/512 filters removed (50.00%)
features.40: 256/512 filters removed (50.00%)
features.43: 256/512 filters removed (50.00%)
```

[46]:
```python
import torch
import torch.nn.utils.prune as prune
import torch.nn as nn

# ------------------------------
# Load your pruned + fine-tuned model
```

```
# ------------------------------
PATH = "result/VGG16_quant/model_pruned_finetuned_pt1.pth.tar"
checkpoint = torch.load(PATH)
model.load_state_dict(checkpoint['state_dict'])
model.eval().cuda()

# ---------------- Compute Metrics ----------------

# Layer-wise and total sparsity
layer_sparsity, total_sparsity = compute_sparsity(model)
print("----- Per-layer sparsity (%) -----")
for layer, sp in layer_sparsity.items():
    print(f"{layer}: {sp:.2f}%")
print(f"\nTotal sparsity: {total_sparsity:.2f}%")

# Compute MACs
macs_after = compute_macs_vgg_quant(model)
print(f"Estimated MACs after pruning: {macs_after/1e6:.2f} M")

# Compute model size reduction (assume 4-bit weights for quantized model)
pruned_model_size = compute_model_size(model, bits_per_weight=4)
print(f"Pruned model size (int4): {pruned_model_size:.2f} MB")

# Memory bandwidth reduction is roughly proportional to model size reduction
print(f"Estimated memory bandwidth reduction: {total_sparsity:.2f}%")

# Final accuracy
def validate(model, testloader, criterion):
    model.eval()
    correct = 0
    total = 0
    with torch.no_grad():
        for images, labels in testloader:
            images = images.cuda()
            labels = labels.cuda()
            outputs = model(images)
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()
    return 100.0 * correct / total

accuracy = validate(model, testloader, criterion)
print(f"\nFinal accuracy after pruning & fine-tuning: {accuracy:.2f}%")
print('Number of MACs before pruning: ', compute_macs_vgg_quant(model,␣
 ↪input_size=(3,32,32)))
```

```
----- Per-layer sparsity (zeros %) -----
features.0: 50.00% sparse
features.3: 50.00% sparse
features.7: 50.00% sparse
features.10: 50.00% sparse
features.14: 50.00% sparse
features.17: 50.00% sparse
features.20: 50.00% sparse
features.24: 50.00% sparse
features.27: 50.00% sparse
features.29: 50.00% sparse
features.31: 50.00% sparse
features.33: 50.00% sparse
features.37: 50.00% sparse
features.40: 50.00% sparse
features.43: 50.00% sparse

----- Total model sparsity -----
Overall sparsity: 50.00%
----- Per-layer sparsity (%) -----
features.0: 50.00%
features.3: 50.00%
features.7: 50.00%
features.10: 50.00%
features.14: 50.00%
features.17: 50.00%
features.20: 50.00%
features.24: 50.00%
features.27: 50.00%
features.29: 50.00%
features.31: 50.00%
features.33: 50.00%
features.37: 50.00%
features.40: 50.00%
features.43: 50.00%

Total sparsity: 50.00%
Estimated MACs after pruning: 12656.58 M
Pruned model size (int4): 2.95 MB
Estimated memory bandwidth reduction: 50.00%

Final accuracy after pruning & fine-tuning: 89.71%
Number of MACs before pruning:  12656579584
```

[ ]:
```python
import torch

# Make sure your model and captured hook are ready
```

```
x_q = captured['x']                # Already quantized activations from hook
weight_q = conv_layer.weight_q     # Already quantized weights

# Flatten and convert to CPU for saving
x_flat = x_q.detach().cpu().numpy().flatten()
w_flat = weight_q.detach().cpu().numpy().flatten()

# Save to text files
import os
save_dir = 'result/VGG16_quant/'
os.makedirs(save_dir, exist_ok=True)

x_path = os.path.join(save_dir, 'feature29_input_activations.txt')
w_path = os.path.join(save_dir, 'feature29_weights.txt')

# Save as float values
with open(x_path, 'w') as f:
    for val in x_flat:
        f.write(f"{val}\n")

with open(w_path, 'w') as f:
    for val in w_flat:
        f.write(f"{val}\n")

print(f"Saved {x_flat.shape[0]} input activations to {x_path}")
print(f"Saved {w_flat.shape[0]} weights to {w_path}")
```

[ ]:

[ ]:
```
w_bit = 4
weight_q = conv_layer.weight_q # quantized value is stored during the training
w_alpha = conv_layer.weight_quant.wgt_alpha.data  # alpha is defined in your␣
 ↪model already. bring it out here
w_delta = 2 * w_alpha / (2**w_bit - 1)    # delta can be calculated by using␣
 ↪alpha and w_bit
weight_int = torch.round(weight_q / w_delta).to(torch.int32) # w_int can be␣
 ↪calculated by weight_q and w_delta
print(weight_int) # you should see clean integer numbers
```

[ ]:
```
x_bit = 4
x = x = captured['x']   # input of the 2nd conv layer
x_alpha  = conv_layer.act_alpha.data
x_delta = 2 * x_alpha / (2**x_bit - 1)

act_quant_fn = act_quantization(x_bit) # define the quantization function
x_q = act_quant_fn(x, x_alpha)         # create the quantized value for x
```

```
x_int = torch.round(x_q / x_delta).to(torch.int32)
print(x_int) # you should see clean integer numbers
```

[ ]:
```
conv_int = torch.nn.Conv2d(in_channels = 64, out_channels=64, kernel_size = 3,␣
 ↪bias = False)
conv_int.weight = torch.nn.parameter.Parameter(weight_int)

output_int =  F.conv2d(x_int.float(), weight_int.float(), bias=None,
                       stride=conv_layer.stride, padding=conv_layer.padding)   ␣
 ↪# output_int can be calculated with conv_int and x_int
output_recovered = output_int * (x_delta * w_delta)  # recover with x_delta and␣
 ↪w_delta
print(output_recovered)
```

[ ]:
```
#### input floating number / weight quantized version

conv_ref = torch.nn.Conv2d(in_channels = 64, out_channels=64, kernel_size = 3,␣
 ↪bias = False)
conv_ref.weight = model.features[3].weight_q

output_ref = conv_ref(x)
print(output_ref)
```

[ ]:
```
difference = abs( output_ref - output_recovered )
print(difference.mean())  ## It should be small, e.g.,2.3 in my trainned model
```

[ ]:
```
#### input floating number / weight floating number version

conv_ref = torch.nn.Conv2d(in_channels = 64, out_channels=64, kernel_size = 3,␣
 ↪bias = False)
weight = model.features[3].weight
mean = weight.data.mean()
std = weight.data.std()
conv_ref.weight = torch.nn.parameter.Parameter(weight.add(-mean).div(std))

output_ref = conv_ref(x)
print(output_ref)
```

[ ]:
```
difference = abs( output_ref - output_recovered )
print(difference.mean())  ## It should be small, e.g.,2.3 in my trainned model
```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]: