Transfer Learning in PyTorch

Case: freeze all the weights and only train a new last layer (FC)

```
# load the pretrained model
vgg16 = models.vgg16(pretrained=True)

# we can access components of model
print(vgg16.classifier[6].in_features

# freeze all but last layers in model
for param in vgg16.features.parameters():
    param.requires_grad = False

# change the final classifier layer
vgg16.classifier[6] = nn.Linear(4096, 5)

# loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(vgg16.classifier.parameters(), lr=0.01)
```

we only want to change the last classifier group

```
# send to gpu if possible
if train_on_gpu:    vgg16.cuda()

# train loop

for epoch in range(n_epochs):          → small; because we are using a pretrained model.
    train_loss = 0.0
    for batch_i, (data, target) in enumerate(train_loader):
        if train_on_gpu:
            data, target = data.cuda(), target.cuda()
        optimizer.zero_grad()
        output = vgg16(data)
        loss = criterion(output, target)
        loss.backward()
        optimizer.step()
        train_loss += loss.item()
train_loss /= len(train_loader)
```