CNN

May 19, 2019

In [1]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

import torch
import torchvision
from torchvision.datasets import ImageFolder
import torchvision.transforms as transforms

import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
```

In [9]:
```
#Setup Torch CUDA torch device
device = torch.device('cuda:0')
```

0.1 Dataset

In [3]:
```
transform = transforms.Compose([
    transforms.ToTensor(),  # Transform to tensor
    transforms.Normalize([0.5,0.5,0.5],[0.5,0.5,0.5])
    #transforms.Normalize((0.5,), (0.5,))  # Min-max scaling to [-1, 1]
])

data_dir = os.path.join('fruits')
print('Data stored in %s' % data_dir)

trainset = ImageFolder("./fruits/Training",transform=transform)
testset = ImageFolder("./fruits/Test",transform=transform)
```

```
Data stored in fruits
```

In [4]:
```
def generate_labels():
    trainset_labels = []
    testset_labels = []
    for i in trainset.imgs:
```

```
            trainset_labels.append(i[1])

        for j in testset.imgs:
            testset_labels.append(j[1])

        return (trainset_labels, testset_labels)
```

```
In [5]: # Total classes
        classes_idx_dict = trainset.class_to_idx # {'Class Name': idx }
        classes = len(trainset.classes)
        len_trainset = len(trainset)
        len_testset = len(testset)

        train_labels, test_labels = generate_labels()
        print(f'Trainset has total of {classes} classes')
```

```
Trainset has total of 103 classes
```

```
In [6]: trainloader = torch.utils.data.DataLoader(trainset, batch_size=60, shuffle=True)
        testloader = torch.utils.data.DataLoader(testset, batch_size=60, shuffle=False)

        image_shape = iter(trainloader).next()[0].shape
        _, CHANNELS, HEIGHT, WIDTH = iter(trainloader).next()[0].shape
        print(f'Image: batch size={image_shape[0]}, channels={image_shape[1]}, image height={ima
```

```
Image: batch size=60, channels=3, image height=100, image width=100
```

```
In [7]: class CCNet(nn.Module):
            def __init__(self):
                """
                Args:
                  n_channels (int): Number of channels in the first convolutional layer. The num
                                    following layers are the multipliers of n_channels.
                """
                super(CCNet, self).__init__()
                self.conv1 = nn.Sequential(
                    nn.Conv2d(3, 16, kernel_size=5),
                    nn.ReLU(),
                    nn.MaxPool2d(kernel_size=2, stride=2),
                    nn.Conv2d(16, 32, kernel_size=5),
                    nn.ReLU(),
                    nn.MaxPool2d(kernel_size=2, stride=2),
                    nn.Conv2d(32, 64, kernel_size=5),
                    nn.ReLU(),
                    nn.MaxPool2d(kernel_size=2, stride=2),
                    nn.Conv2d(64, 128, kernel_size=5),
                    nn.ReLU(),
```

```
                nn.MaxPool2d(kernel_size=3, stride=3)
            )
            self.fc1 = nn.Sequential(
                nn.Linear(128, 1024),
                nn.ReLU(),
                nn.Dropout2d(p=0.8),
                nn.Linear(1024, 256),
                nn.ReLU(),
                nn.Dropout2d(p=0.8),
                nn.Linear(256, classes),
            )


        def forward(self, x, verbose=False):
            """You can (optionally) print the shapes of the intermediate variables with verb
            x = self.conv1(x)
            x = x.view(x.size(0), -1)
            x = self.fc1(x)
            return x
```

```
In [10]: # Let's test the shapes of the tensors
         net = CCNet()
         net.to(device)

         with torch.no_grad():
             dataiter = iter(trainloader)
             images, labels = dataiter.next()
             images = images.to(device)
             print('Shape of the input tensor:', images.shape)

             y = net(images, verbose=True)
             print(y.shape)
             assert y.shape == torch.Size([60, classes]), f'Bad shape of y: y.shape={y.shape}'

         print('The shapes seem to be ok.')
```

```
Shape of the input tensor: torch.Size([60, 3, 100, 100])
torch.Size([60, 103])
The shapes seem to be ok.
```

```
In [134]: def compute_accuracy(net, testloader):
              net.eval()
              correct = 0
              total = 0
              with torch.no_grad():
                  for images, labels in testloader:
                      images, labels = images.to(device), labels.to(device)
```

```
                    outputs = net(images)
                    _, predicted = torch.max(outputs.data, 1)
                    total += labels.size(0)
                    correct += (predicted == labels).sum().item()
            return correct / total
```

In [135]:
```
initial_learning_rate = 0.001
final_learning_rate = 0.00001
learning_rate = initial_learning_rate

criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(net.parameters(), lr=learning_rate)
```

In [136]:
```
n_epochs=75
net.train()
for epoch in range(n_epochs):
    running_loss = 0.0
    print_every = 200  # mini-batches
    for i, (inputs, labels) in enumerate(trainloader, 0):
        # Transfer to GPU
        inputs, labels = inputs.to(device), labels.to(device)

        # zero the parameter gradients
        optimizer.zero_grad()

        # forward + backward + optimize
        outputs = net(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        # print statistics
        running_loss += loss.item()
        if (i % print_every) == (print_every-1):
            print('[%d, %5d] loss: %.3f' % (epoch+1, i+1, running_loss/print_every))
            running_loss = 0.0

    # Print accuracy after every epoch
    accuracy = compute_accuracy(net, testloader)
    print(f'Accuracy of the network on the {len_testset} test images: {100 * accuracy}

print('Finished Training')
```

```
[1,   200] loss: 4.283
[1,   400] loss: 3.528
[1,   600] loss: 2.849
[1,   800] loss: 2.326
Accuracy of the network on the 17845 test images: 53.28103110114878%
```

```
[2,   200] loss: 0.752
[2,   400] loss: 0.273
[2,   600] loss: 0.151
[2,   800] loss: 0.068
Accuracy of the network on the 17845 test images: 91.3981507425049%
[3,   200] loss: 0.075
[3,   400] loss: 0.082
[3,   600] loss: 0.047
[3,   800] loss: 0.027
Accuracy of the network on the 17845 test images: 90.92743065284394%
[4,   200] loss: 0.077
[4,   400] loss: 0.065
[4,   600] loss: 0.035
[4,   800] loss: 0.019
Accuracy of the network on the 17845 test images: 95.50574390585598%
[5,   200] loss: 0.057
[5,   400] loss: 0.062
[5,   600] loss: 0.050
[5,   800] loss: 0.052
Accuracy of the network on the 17845 test images: 93.65648641075933%
[6,   200] loss: 0.006
[6,   400] loss: 0.003
[6,   600] loss: 0.002
[6,   800] loss: 0.002
Accuracy of the network on the 17845 test images: 95.78033062482488%
[7,   200] loss: 0.001
[7,   400] loss: 0.000
[7,   600] loss: 0.035
[7,   800] loss: 0.161
Accuracy of the network on the 17845 test images: 93.19137013168954%
[8,   200] loss: 0.030
[8,   400] loss: 0.063
[8,   600] loss: 0.060
[8,   800] loss: 0.017
Accuracy of the network on the 17845 test images: 93.63407116839451%
[9,   200] loss: 0.026
[9,   400] loss: 0.040
[9,   600] loss: 0.005
[9,   800] loss: 0.005
Accuracy of the network on the 17845 test images: 93.35388063883441%
[10,   200] loss: 0.007
[10,   400] loss: 0.005
[10,   600] loss: 0.012
[10,   800] loss: 0.111
Accuracy of the network on the 17845 test images: 94.1552255533763%
[11,   200] loss: 0.068
[11,   400] loss: 0.019
[11,   600] loss: 0.003
```

```
[11,   800] loss: 0.015
Accuracy of the network on the 17845 test images: 91.60549173437937%
[12,   200] loss: 0.082
[12,   400] loss: 0.037
[12,   600] loss: 0.024
[12,   800] loss: 0.011
Accuracy of the network on the 17845 test images: 91.7119641356122%
[13,   200] loss: 0.049
[13,   400] loss: 0.009
[13,   600] loss: 0.011
[13,   800] loss: 0.025
Accuracy of the network on the 17845 test images: 95.79153824600728%
[14,   200] loss: 0.017
[14,   400] loss: 0.009
[14,   600] loss: 0.000
[14,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.86999159428412%
[15,   200] loss: 0.000
[15,   400] loss: 0.000
[15,   600] loss: 0.000
[15,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.93723732137853%
[16,   200] loss: 0.000
[16,   400] loss: 0.000
[16,   600] loss: 0.000
[16,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.95965256374335%
[17,   200] loss: 0.000
[17,   400] loss: 0.000
[17,   600] loss: 0.000
[17,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.98767161669936%
[18,   200] loss: 0.000
[18,   400] loss: 0.000
[18,   600] loss: 0.000
[18,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.99327542729056%
[19,   200] loss: 0.000
[19,   400] loss: 0.000
[19,   600] loss: 0.000
[19,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.01008685906416%
[20,   200] loss: 0.000
[20,   400] loss: 0.000
[20,   600] loss: 0.000
[20,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.00448304847296%
[21,   200] loss: 0.000
```

```
[21,   400] loss: 0.000
[21,   600] loss: 0.000
[21,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.95965256374335%
[22,   200] loss: 0.000
[22,   400] loss: 0.000
[22,   600] loss: 0.000
[22,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.1053516391146%
[23,   200] loss: 0.000
[23,   400] loss: 0.000
[23,   600] loss: 0.000
[23,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.04370972261138%
[24,   200] loss: 0.000
[24,   400] loss: 0.000
[24,   600] loss: 0.000
[24,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.16138974502661%
[25,   200] loss: 0.000
[25,   400] loss: 0.000
[25,   600] loss: 0.000
[25,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.05491734379378%
[26,   200] loss: 0.000
[26,   400] loss: 0.000
[26,   600] loss: 0.000
[26,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.8307649201457%
[27,   200] loss: 0.000
[27,   400] loss: 0.000
[27,   600] loss: 0.235
[27,   800] loss: 0.086
Accuracy of the network on the 17845 test images: 93.23059680582796%
[28,   200] loss: 0.022
[28,   400] loss: 0.005
[28,   600] loss: 0.008
[28,   800] loss: 0.033
Accuracy of the network on the 17845 test images: 95.57298963295041%
[29,   200] loss: 0.023
[29,   400] loss: 0.043
[29,   600] loss: 0.074
[29,   800] loss: 0.040
Accuracy of the network on the 17845 test images: 94.4578313253012%
[30,   200] loss: 0.029
[30,   400] loss: 0.006
[30,   600] loss: 0.018
[30,   800] loss: 0.034
```

```
Accuracy of the network on the 17845 test images: 92.86634911739982%
[31,   200] loss: 0.016
[31,   400] loss: 0.012
[31,   600] loss: 0.006
[31,   800] loss: 0.082
Accuracy of the network on the 17845 test images: 94.49145418884841%
[32,   200] loss: 0.012
[32,   400] loss: 0.003
[32,   600] loss: 0.005
[32,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.61782011768003%
[33,   200] loss: 0.000
[33,   400] loss: 0.000
[33,   600] loss: 0.000
[33,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.85878397310171%
[34,   200] loss: 0.000
[34,   400] loss: 0.000
[34,   600] loss: 0.000
[34,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.90921826842252%
[35,   200] loss: 0.000
[35,   400] loss: 0.000
[35,   600] loss: 0.000
[35,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.99887923788177%
[36,   200] loss: 0.000
[36,   400] loss: 0.000
[36,   600] loss: 0.000
[36,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.02689829083776%
[37,   200] loss: 0.000
[37,   400] loss: 0.000
[37,   600] loss: 0.000
[37,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.06612496497618%
[38,   200] loss: 0.000
[38,   400] loss: 0.000
[38,   600] loss: 0.000
[38,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.16138974502661%
[39,   200] loss: 0.000
[39,   400] loss: 0.000
[39,   600] loss: 0.000
[39,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.17820117680023%
[40,   200] loss: 0.000
[40,   400] loss: 0.000
```

```
[40,   600] loss: 0.000
[40,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.17259736620902%
[41,   200] loss: 0.000
[41,   400] loss: 0.000
[41,   600] loss: 0.000
[41,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.27346595685066%
[42,   200] loss: 0.000
[42,   400] loss: 0.000
[42,   600] loss: 0.000
[42,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.26225833566825%
[43,   200] loss: 0.000
[43,   400] loss: 0.000
[43,   600] loss: 0.000
[43,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.28467357803306%
[44,   200] loss: 0.000
[44,   400] loss: 0.000
[44,   600] loss: 0.000
[44,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.3687307369011%
[45,   200] loss: 0.238
[45,   400] loss: 0.085
[45,   600] loss: 0.021
[45,   800] loss: 0.030
Accuracy of the network on the 17845 test images: 95.34883720930233%
[46,   200] loss: 0.027
[46,   400] loss: 0.039
[46,   600] loss: 0.006
[46,   800] loss: 0.001
Accuracy of the network on the 17845 test images: 94.28971700756514%
[47,   200] loss: 0.057
[47,   400] loss: 0.011
[47,   600] loss: 0.008
[47,   800] loss: 0.008
Accuracy of the network on the 17845 test images: 96.49201456990754%
[48,   200] loss: 0.005
[48,   400] loss: 0.006
[48,   600] loss: 0.054
[48,   800] loss: 0.084
Accuracy of the network on the 17845 test images: 91.43177360605212%
[49,   200] loss: 0.050
[49,   400] loss: 0.028
[49,   600] loss: 0.001
[49,   800] loss: 0.001
Accuracy of the network on the 17845 test images: 95.42729055757916%
```

```
[50,   200] loss: 0.000
[50,   400] loss: 0.001
[50,   600] loss: 0.000
[50,   800] loss: 0.055
Accuracy of the network on the 17845 test images: 90.93863827402635%
[51,   200] loss: 0.125
[51,   400] loss: 0.055
[51,   600] loss: 0.073
[51,   800] loss: 0.011
Accuracy of the network on the 17845 test images: 96.14457831325302%
[52,   200] loss: 0.000
[52,   400] loss: 0.000
[52,   600] loss: 0.000
[52,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.81143177360605%
[53,   200] loss: 0.000
[53,   400] loss: 0.000
[53,   600] loss: 0.000
[53,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.89548893247408%
[54,   200] loss: 0.000
[54,   400] loss: 0.000
[54,   600] loss: 0.000
[54,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.87307369010928%
[55,   200] loss: 0.000
[55,   400] loss: 0.070
[55,   600] loss: 0.188
[55,   800] loss: 0.045
Accuracy of the network on the 17845 test images: 93.0456710563183%
[56,   200] loss: 0.031
[56,   400] loss: 0.006
[56,   600] loss: 0.008
[56,   800] loss: 0.009
Accuracy of the network on the 17845 test images: 93.44354160829363%
[57,   200] loss: 0.051
[57,   400] loss: 0.023
[57,   600] loss: 0.003
[57,   800] loss: 0.020
Accuracy of the network on the 17845 test images: 94.08237601569067%
[58,   200] loss: 0.026
[58,   400] loss: 0.069
[58,   600] loss: 0.031
[58,   800] loss: 0.052
Accuracy of the network on the 17845 test images: 94.04314934155225%
[59,   200] loss: 0.022
[59,   400] loss: 0.052
[59,   600] loss: 0.006
```

```
[59,   800] loss: 0.028
Accuracy of the network on the 17845 test images: 94.07677220509947%
[60,   200] loss: 0.005
[60,   400] loss: 0.002
[60,   600] loss: 0.008
[60,   800] loss: 0.015
Accuracy of the network on the 17845 test images: 92.86074530680864%
[61,   200] loss: 0.078
[61,   400] loss: 0.047
[61,   600] loss: 0.064
[61,   800] loss: 0.005
Accuracy of the network on the 17845 test images: 95.09666573269824%
[62,   200] loss: 0.008
[62,   400] loss: 0.016
[62,   600] loss: 0.006
[62,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.44410198935276%
[63,   200] loss: 0.000
[63,   400] loss: 0.000
[63,   600] loss: 0.000
[63,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.64583917063604%
[64,   200] loss: 0.000
[64,   400] loss: 0.000
[64,   600] loss: 0.000
[64,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.8475763519193%
[65,   200] loss: 0.000
[65,   400] loss: 0.000
[65,   600] loss: 0.000
[65,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 95.98206780610815%
[66,   200] loss: 0.000
[66,   400] loss: 0.000
[66,   600] loss: 0.000
[66,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.03250210142897%
[67,   200] loss: 0.000
[67,   400] loss: 0.000
[67,   600] loss: 0.000
[67,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.07733258615859%
[68,   200] loss: 0.000
[68,   400] loss: 0.000
[68,   600] loss: 0.000
[68,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.0941440179322%
[69,   200] loss: 0.000
```

```
[69,   400] loss: 0.000
[69,   600] loss: 0.000
[69,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.13337069207061%
[70,   200] loss: 0.000
[70,   400] loss: 0.000
[70,   600] loss: 0.000
[70,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.13337069207061%
[71,   200] loss: 0.000
[71,   400] loss: 0.000
[71,   600] loss: 0.000
[71,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.16138974502661%
[72,   200] loss: 0.000
[72,   400] loss: 0.000
[72,   600] loss: 0.000
[72,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.16699355561782%
[73,   200] loss: 0.000
[73,   400] loss: 0.000
[73,   600] loss: 0.000
[73,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.20622022975624%
[74,   200] loss: 0.000
[74,   400] loss: 0.000
[74,   600] loss: 0.000
[74,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.20061641916503%
[75,   200] loss: 0.000
[75,   400] loss: 0.000
[75,   600] loss: 0.000
[75,   800] loss: 0.000
Accuracy of the network on the 17845 test images: 96.27346595685066%
Finished Training
```

```
In [137]: accuracy = compute_accuracy(net, testloader)
          print('Accuracy of the network on the test images: %.3f' % accuracy)
```

```
Accuracy of the network on the test images: 0.963
```

```
In [139]: filename = 'cnn.pth'

          try:
              do_save = input('Do you want to save the model (type yes to confirm)? ').lower()
              if do_save == 'yes':
```

```
                torch.save(net.state_dict(), filename)
                print('Model saved to %s' % filename)
            else:
                print('Model not saved')
        except:
            raise Exception('The notebook should be run or validated with skip_training=True.'
```

```
Do you want to save the model (type yes to confirm)? yes
Model saved to cnn.pth
```

```
In [ ]: net = CCNet()
        net.load_state_dict(torch.load(filename, map_location=lambda storage, loc: storage))
        net.to(device)
        print(f'Model loaded from {filename}')
```