ECS795P Deep Learning and Computer Vision, 2022

Course Work 1: Image Super-resolution Using Deep Learning

2. Understanding deep learning by convolutional neural network

Objective: To understand the principles of deep convolutional neural network.

Questions:

What is the difference between the training and testing stage of a CNN?

- Training stage: During the training stage our goal is to learn the weights that make the network extract meaningful features from the input.
- Testing Stage: In the testing phase we feed the network new inputs and we test its ability to extract good features.

In conclusion, it can be said that for training stage, we fit the model using training samples and in testing we check the performance of the model against the testing set.

Exercises:

1. To set show the weights of the third convolutional layer
    - To set the channel number of the input
    - To set the filter number
    - To set the filter size
    - To set the padding
    - To show the weight of the 1st filter in command window
    - To show the bias of the 1st filter in command window
2. To perform 2-d convolutional operation on a 2-d matrix with a given filter (Tip: *conv2d* and *relu* are PyTorch build-in functions)

```
#Code
"""Define the model weights and biases
"""
## ------ Add your code here: set the weight of three conv layers
# replace 'None' with your hyper parameter numbers
# conv3 layer with biases and NO relu: 1 filter with size 5 x 5

class SRCNN(nn.Module):
  def __init__(self):
    super(SRCNN, self).__init__()
    self.conv3 = nn.Conv2d(in_channels=32, out_channels=1, kernel_size=5, padding=2)

  def forward(self, x):
    out = F.relu(self.conv1(x))
    out = F.relu(self.conv2(out))
    out = self.conv3(out)
    return out

"""Load the pre-trained model file
"""
model = SRCNN()
model.load_state_dict(torch.load('./model/model.pth'))
model.eval()

"""Read the test image
"""
```

```
LR_image, HR_image = preprocess('./image/butterfly_GT.bmp')
# transform the input to 4-D tensor
input_ = np.expand_dims(np.expand_dims(LR_image, axis=0), axis=0)
input_ = torch.from_numpy(input_)

"""Run the model and get the SR image
"""
with torch.no_grad():
    output_ = model(input_)
    output_ = output_.numpy()
    output_ = np.reshape(output_, (255, 255))
"""Run the model and get the SR image
"""
with torch.no_grad():
    output_ = model(input_)
print("Weights of 3rd Convolutional layer",model.conv3.weight.data.numpy())
print("Weights of 1st filter of 3rd convolutional layer",model.conv3.weight[0].data.numpy())
print("Bias of 1st filter of 3rd convolutional layer", model.conv3.bias[0].data.numpy())
```

Output: