

Questions

“Training”

1. What is the purpose of the softmax activation?
2. What is the value of `F.softmax(torch.tensor([[1,1.]]), dim=0)` ?
3. What is one technique to debug a neural network, to make sure that the programming is likely to be correct?
4. What are the advantages of convolutional layers over fully-connected layers?
5. PyTorch convolutions expect data in the “NCHW” format. What does this mean?
6. What is the purpose of zero padding in a convolutional layer?
7. Explain the idea of using AlexNet features to build a neural network.

“Generalization”

1. Identify 3 issues with the neural network model below:

```
class Model(nn.Module):
    def __init__(self):
        super(Model, self).__init__()
        self.layer1 = nn.Linear(28 * 28, 40)
        self.layer2 = nn.Linear(30, 1)
    def forward(self, img):
        flattened = img.view(-1, 28 * 28)
        activation1 = self.layer1(flattened)
        activation2 = self.layer2(activation1)
        return torch.sigmoid(activation2)
```

2. How many multiplications are performed when applying a 3x3 convolution on a greyscale image of size 7x7?
3. Which operations to reduce the number of units in the hidden layer are the *most* and *least* computationally efficient: max pooling, average pooling, and using strides in a convolution?
4. Suppose we learn a convolutional network on a 128x128 image. Can we use the same network to make predictions about images of size 256x256? Why or why not?