RUTGERS UNIVERSITY
School of Engineering
Department of Electrical & Computer Engineering

ECE 472 – Robotics & Computer Vision– Fall 2022

Project 1 - Deep Learning & Image Classification

Name (last, first) : Mehmet Ali Soner

netID : mas996

RUID: 196000499

Date: November 4, 2022

1 Problem 1

ResNet is a Convolutional Neural Network that incorporated residual learning. If we take the activation function of one layer to be $\mathscr{F}(x)$ and the input to be x, then we would have what a skip connection is called:

$$H(x) = \mathscr{F}(x) + x$$

We feed the input of one layer into the next one without modifying it all; so it skips one layer and gets added to the next one. This idea was inspired by how the brain functions. The human brain demonstrates a similar phenomenon. The main problem with previous CNN architectures was the inability to train networks that had more depth. It was expected that with more depth, we would achieve higher accuracy but this wasn't the case with traditional CNNs. The figure above shows the higher error rate of the 56-layered network in comparison to the 20-layered network. There are multiple explanations for this but the one that occurs the most is the vanishing/exploding gradients problem. The derivative of the sigmoid function has a maximum value of 0.25. If we have more layers and if we keep on multiplying the partial derivatives of each layer, the gradients will decrease to a value close to zero. This has performance issues since the weights remain unchanged in the network. ResNets overcame that problem through the skip connections and the ReLU function, which doesn't cause a small derivative. With ResNets, we can train deeper networks, which in the end can achieve higher accuracy.

2 Problem 2

The first modification that is need is in the first convolutional layer. The MNIST dataset doesn't contain any RGB values and only has one dimension in the third dimension of the image matrix. To clarify, an image you read from the MNIST dataset has shape:

```
train_data = datasets.MNIST(
    root = 'data',
    train = True,
    transform = transforms.ToTensor(),
    download = True,
)
print(train_data.data[0].size())
```

So we modify the first convolutional layer accordingly to

```
model.conv1 = nn.Conv2d(1, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias
                                            =False)
```

The MNIST dataset is a rather small dataset

3 Problem 3