DSE316: Deep Learning
Report
Assignment 1

Name: **Akanksha Singh**
Roll Number: **19022**

Q1:

Model:

1. Data preparation:
    a. Downloaded the data, split it into labels, and data.
    b. The labels are label encoded (not one hot encoding because it is an ordinal set).
    c. The data is scaled.

2. Defining functions used:
    a. The activation function Tanh (ReLU is not used) and SoftMax as the output activation layer were then fed for Categorical Cross-Entropy Loss computation.
    b. For simplicity, Multiply and Add gates have been made.
    c. Adam Optimizer has been written but wasn't implemented for the updation of weights.

3. A two-layer model is written:
    a. The weight and bias matrix are initialized with random values.
    b. Training
        i. Forward Propagation
        ii. Backward Propagation
        iii. Updation
    c. Loss Computation
    d. Prediction

4. The model is run for epochs = 50, 100, 150, 200, and learning rates = 0.1, 0.01, 0.001, 0.0001.
    a. The training loss is computed for all the combinations.
    b. The accuracy of the test set is computed.

5. The graphs for loss to epsilon are plotted for each epoch.

Analysis:

1. The best combination is Epochs = 200 and Learning Rate = 0.01.
2. The losses for lr = 0.1 shows a decrease from epochs 50 (~2.3) to 200 (~1.75), and the best accuracy for the test set for 200 epochs : 0.714
3. The losses for lr = 0.01 shows a very slight decrease from epochs 50 (~1.65) to 200 (~1.63), and the best accuracy for the test set for 200 epochs is 0.7585.

4. The losses for lr = 0.001 shows a trivial decrease from epochs 50 (~1.6511) to 200 (~1.6307), and the best accuracy for the test set for 100 epochs is 0.7585.
5. The losses for lr = 0.001 shows increased values from epochs 50 (~71) to 200 (69.5), and the best accuracy for the test set for 100 epochs is 0.7576.

The model has overfitted for lr = 0.0001. The lr = 0.001 shows no significant decrease with epochs. The model with lr = 0.01 and epoch = 200 shows a similar loss as lr = 0.001. The best combination has the best test accuracy amongst all the combinations, and to avoid overfitting and save computation cost, lr = 0.01 with epoch = 200 is chosen.

Q2:

<u>Model:</u>

1. Data preparation:
    a. The number of processes, batch size, and validation dataset is specified.
    b. The data is transformed into a normalized tensor.
    c. The data then split.
    d. The dataset is loaded as train, validation, and test sets.
    e. The classes of the images are specified.

2. The obtained dataset is visualized.

3. The given CNN architectures are made:
    a. Conv-Pool-Conv-Pool-Conv-Pool-FC
        i. Epoch: 1 Training Loss: 0.582675 Validation Loss: 0.215491
        ii. Epoch: 2 Training Loss: 0.553267 Validation Loss: 0.223704
        iii. Epoch: 3 Training Loss: 0.526494 Validation Loss: 0.212687
        iv. Epoch: 4 Training Loss: 0.499702 Validation Loss: 0.226294
        v. Epoch: 5 Training Loss: 0.469472 Validation Loss: 0.225978
        vi. Epoch: 6 Training Loss: 0.442899 Validation Loss: 0.223693
        vii. Epoch: 7 Training Loss: 0.414405 Validation Loss: 0.240420
        viii. Epoch: 8 Training Loss: 0.385788 Validation Loss: 0.249480
        ix. Epoch: 9 Training Loss: 0.363128 Validation Loss: 0.258889
        x. Epoch: 10 Training Loss: 0.337684 Validation Loss: 0.276507

        xi. Test Loss: 1.370224
        xii. Test Accuracy (Overall): 65% (6591/10000)

    b. Conv-Conv-Pool-Conv-Conv-Pool-FC
        i. Epoch: 1 Training Loss: 1.160348 Validation Loss: 0.244664
        ii. Epoch: 2 Training Loss: 0.908393 Validation Loss: 0.221281
        iii. Epoch: 3 Training Loss: 0.831461 Validation Loss: 0.212766
        iv. Epoch: 4 Training Loss: 0.775786 Validation Loss: 0.211490
        v. Epoch: 5 Training Loss: 0.732582 Validation Loss: 0.203158
        vi. Epoch: 6 Training Loss: 0.693069 Validation Loss: 0.199201
        vii. Epoch: 7 Training Loss: 0.655467 Validation Loss: 0.201940
        viii. Epoch: 8 Training Loss: 0.617636 Validation Loss: 0.205589

ix. Epoch: 9 Training Loss: 0.580544 Validation Loss: 0.204567
x. Epoch: 10 Training Loss: 0.543775 Validation Loss: 0.212910

xi. Test Loss: 1.042511
xii. Test Accuracy (Overall): 66% (6642/10000)

c. Conv-Pool-Conv-Pool-Conv-Pool-FC-FC
    i. Epoch: 1 Training Loss: 1.052777 Validation Loss: 0.248382
    ii. Epoch: 2 Training Loss: 0.930732 Validation Loss: 0.245106
    iii. Epoch: 3 Training Loss: 0.863808 Validation Loss: 0.219897
    iv. Epoch: 4 Training Loss: 0.820059 Validation Loss: 0.217911
    v. Epoch: 5 Training Loss: 0.783895 Validation Loss: 0.208877
    vi. Epoch: 6 Training Loss: 0.751130 Validation Loss: 0.207117
    vii. Epoch: 7 Training Loss: 0.721221 Validation Loss: 0.200606
    viii. Epoch: 8 Training Loss: 0.695245 Validation Loss: 0.206154
    ix. Epoch: 9 Training Loss: 0.671696 Validation Loss: 0.208676
    x. Epoch: 10 Training Loss: 0.649317 Validation Loss: 0.198013

    xi. Test Loss: 0.961351
    xii. Test Accuracy (Overall): 67% (6795/10000)

4. Functions:
    a. Cross Entropy Loss is used.
    b. Stochastic Gradient Descent Optimization is used.
    c. Epochs = 10
    d. Learning Rate = 0.01

5. Training:
    a. The training and validation loss is calculated.
    b. We maintain a track of minimum validation loss and select the best model that is saved.
6. Testing:
    a. Test loss is calculated.
    b. Test accuracy for each class label is calculated, and overall test accuracy is found.

Analysis:

A. How does changing the network size change the accuracy?
With the change in the network size, the computation should be longer and increase accuracy for a well-generalized model; otherwise, it can decrease accuracy if it overfits. It should depend on the dataset and be computationally feasible.

B. Experiment with different pooling sizes and do a detailed analysis of pooling size on the network.
As we increase the pooling size, the convergence and computation should be faster, but it will underfit the model if the pooling size is very large with larger strides.

For pooling layers, when we compare A and C to B, we can see that the convergence time is higher, and the computation time is double for B. Also, the accuracy is less because pooling helps prevent overfitting because of learning from all the given data, including less significant features.

C. How the presence of one or more fully connected layers changes the accuracy?
When we compare A. and C, the fully connected layer helps optimize the probability of logits. There is a decrease in both training and validation loss simultaneously for C, whereas in A, there is an increase in the validation loss. Hence, slightly better accuracy is found for models with 2 FC layers.

The FC layer learns features of the class labels without considering the features responsible for other class labels. After convolutional layers have learned classifying features to distinguish within class labels, the FC layers are added. Hence only increasing the prediction accuracy for class labels by increasing the weights of the most important features corresponding to the specific class label.

A model with no FC layer will have significantly lower accuracy than a model with an FC layer as the penultimate layer.