18786 – HW1 Report

Cheng An Hsieh(chengan2)

Problem 1.

- MLP design and training details:
  - Optimizer: Adam
  - Number of epochs: 10
  - Learning rate: 1e-3
  - Loss: CrossEntropy
  - MLP design:
    - Two linear layers
    - Activation Function: ReLU
    - Hidden size: 64
- Differences in performance for various batch sizes:

Batch Size	Time cost (s)	Train acc(%)	Val acc(%)	Test acc(%)
2	72.10	99.94	99.88	99.86
16	9.04	99.97	100	99.95
128	1.38	99.94	99.92	99.91
1024	0.64	99.80	99.88	99.91

Problem 2.

- MLP design and training details:
  - Optimizer: Adam
  - Number of epochs: 10
  - Learning rate: 1e-3
  - Batch size: 16
  - Loss: CrossEntropy
  - MLP design:
    - Two linear layers
    - Activation Function: ReLU
- Differences in performance for various hidden sizes:

Hidden Size	Time cost (s)	Train acc(%)	Val acc(%)	Test acc(%)
32	41.65	95.23	94.7	94.95
64	48.13	97.85	95.93	95.68
128	67.74	98.53	96.94	96.89
256	103.99	98.81	96.97	96.82

- Analysis:
  - Hidden Layer Size and Time Cost:

    There is a clear trend showing an increase in time cost as the hidden layer size increases. This is due to the larger number of parameters to train in larger networks, leading to longer computational times.

- Accuracy:
  Both the train and test accuracies are high, indicating good generalization of the models to unseen data.
- Challenges and solutions:
  For the largest hidden size 256, there is a slight performance drop. This may be an overfitting and is a challenge. Although the difference is not dramatic, it's a common issue with larger networks. This could be addressed by introducing regularization techniques such as dropout or L2 regularization, or by collecting more training data if possible.

Problem 3.

- Without optimization
  - MLP design and training details:
    - Optimizer: Adam
    - Number of epochs: 10
    - Learning rate: 1e-3
    - Batch size: 16
    - Loss: CrossEntropy
    - MLP design:
      - Two linear layers
      - Activation Function: ReLU
      - Hidden dim: 128

Train acc(%)	Train F1-score	Test acc(%)	Test F1-score
100%	1.0	99.43	0.995

- With optimization
  - Modification:
    - For the Cross-entropy loss, add different weight to label 0 and label1. For label 0, the weight is 1. For label1, the weight is 100.
    - Create sampler for Dataloader. The weight for label 0 is 1 and the weight for label 1 is 100.

Train acc(%)	Train F1-score	Test acc(%)	Test F1-score
100%	1.0	99.95	0.999

Problem 3. Bonus

The MLP design and training details are the same as the previous problem. I added four optimized methods (two are same as previous section):

1. For the Cross-entropy loss, add different weight to label 0 and label1. For label 0, the weight is 1. For label1, the weight is 50.
2. Create sampler for Dataloader. The weight for label 0 is 1 and the weight for label 1 is 100.
3. Data augmentation by random rotation and horizontal flip.
4. Two-stage training. Train with both label 0 and label 1 for 10 epochs. And then only train for label 1 for 2 epochs.

Optimize	N=500	N=1000	N=1500	N=2000	N=2500	N=3000
No	0.992	0.990	0.990	0.988	0.985	0.984
Yes(1,2,3)	**0.995**	**0.993**	**0.992**	0.991	0.991	0.990
Yes(1,2,3,4)	0.994	0.992	0.992	**0.992**	**0.996**	**0.993**

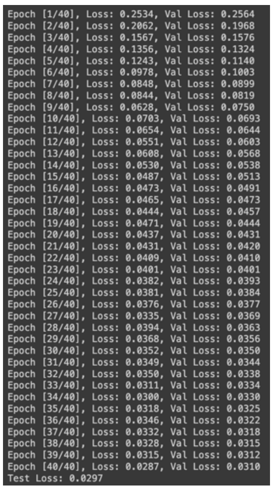
From the results, we can observe that the optimized methods are effective. However, it is also apparent that in the two-stage training approach, if label 1 still has a large number of samples (e.g., N=500, N=1000), the second stage of training may excessively influence the model weights to fit label 1, leading to a drop in the F1 score. Conversely, when label 1 has fewer samples, the F1 score tends to increase.

Problem 4.

- MLP design and training details:
  - Optimizer: Adam
  - Number of epochs: 40
  - Learning rate: 1e-4
  - Batch size: 64
  - Loss: MSE Loss
  - MLP design:
    - Two linear layers
    - Activation Function: Tanh
    - Hidden size: 50
- Result:

  Mean Squared Error:

  Train: 0.0287 Val: 0.0310 Test: 0.0297



```
Epoch [1/40], Loss: 0.2534, Val Loss: 0.2564
Epoch [2/40], Loss: 0.2062, Val Loss: 0.1968
Epoch [3/40], Loss: 0.1567, Val Loss: 0.1576
Epoch [4/40], Loss: 0.1356, Val Loss: 0.1324
Epoch [5/40], Loss: 0.1243, Val Loss: 0.1140
Epoch [6/40], Loss: 0.0978, Val Loss: 0.1003
Epoch [7/40], Loss: 0.0848, Val Loss: 0.0899
Epoch [8/40], Loss: 0.0844, Val Loss: 0.0819
Epoch [9/40], Loss: 0.0628, Val Loss: 0.0750
Epoch [10/40], Loss: 0.0703, Val Loss: 0.0693
Epoch [11/40], Loss: 0.0654, Val Loss: 0.0644
Epoch [12/40], Loss: 0.0551, Val Loss: 0.0603
Epoch [13/40], Loss: 0.0608, Val Loss: 0.0568
Epoch [14/40], Loss: 0.0530, Val Loss: 0.0538
Epoch [15/40], Loss: 0.0487, Val Loss: 0.0513
Epoch [16/40], Loss: 0.0473, Val Loss: 0.0491
Epoch [17/40], Loss: 0.0465, Val Loss: 0.0473
Epoch [18/40], Loss: 0.0444, Val Loss: 0.0457
Epoch [19/40], Loss: 0.0471, Val Loss: 0.0444
Epoch [20/40], Loss: 0.0437, Val Loss: 0.0431
Epoch [21/40], Loss: 0.0431, Val Loss: 0.0420
Epoch [22/40], Loss: 0.0409, Val Loss: 0.0410
Epoch [23/40], Loss: 0.0401, Val Loss: 0.0401
Epoch [24/40], Loss: 0.0382, Val Loss: 0.0393
Epoch [25/40], Loss: 0.0381, Val Loss: 0.0384
Epoch [26/40], Loss: 0.0376, Val Loss: 0.0377
Epoch [27/40], Loss: 0.0335, Val Loss: 0.0369
Epoch [28/40], Loss: 0.0394, Val Loss: 0.0363
Epoch [29/40], Loss: 0.0368, Val Loss: 0.0356
Epoch [30/40], Loss: 0.0352, Val Loss: 0.0350
Epoch [31/40], Loss: 0.0349, Val Loss: 0.0344
Epoch [32/40], Loss: 0.0350, Val Loss: 0.0338
Epoch [33/40], Loss: 0.0311, Val Loss: 0.0334
Epoch [34/40], Loss: 0.0300, Val Loss: 0.0330
Epoch [35/40], Loss: 0.0318, Val Loss: 0.0325
Epoch [36/40], Loss: 0.0346, Val Loss: 0.0322
Epoch [37/40], Loss: 0.0332, Val Loss: 0.0318
Epoch [38/40], Loss: 0.0328, Val Loss: 0.0315
Epoch [39/40], Loss: 0.0315, Val Loss: 0.0312
Epoch [40/40], Loss: 0.0287, Val Loss: 0.0310
Test Loss: 0.0297
```

- Visualize the reconstructed images: