COMP 3055 Machine Learning Course Work Report

Qingyang LU 20124987

1. Introduction

Computer vision is an important area in machine learning and deep learning. The purpose of this course work is to use the CIFAR-10 dataset to train the MLP and CNN model and do multiclass classification. Methods like PCA was introduced and results were compared. A comparison between MLP and CNN was carried.

2. Dataset

CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. In Task 1 and 2, Due to the limitation of computing power, a subset of the CIFAR-10 dataset was used. The new dataset consists of 6000 32x32 colour images in 10 classes, 5000 training images and 1000 test images.

10 Classes are

Classe	0	1	2	3	4	5	6	7	8	9
name	plane	car	bird	cat	deer	dog	frog	horse	ship	truck

3. Data preprocessing(TASK 1)

To improve the performance and efficiency, standardization was used on both train set and test set.

The CIFAR-10 dataset consists of 3072 features. PCA was used to reduce the number of features. Here is the information kept and noise variance of datasets after PCA with different amounts of features.

Number of features	Information kept	Noise variance
1	26.767033529080514%	0.6179963886133141
4	49.28327729088496%	0.47269039853937345
15	70.05784552371027%	0.2923527359413927
98	90.00647638206063%	0.10248083841374055

4. Multilayer Perception(MLP) (TASK 2)

MLP was applied to do multi-class classification. First, to measure the effect of PCA, the performance of datasets with a different number of features were compared. 5 datasets (1 full dataset, 4 reduced datasets) was used to train the model with 10-fold cross-validation. The MLP has 3 layers with 1024neros each and the parameters of MLP model training is (activation='relu', max_iter=1000,solver='adam',learning_rate_init=0.0001)

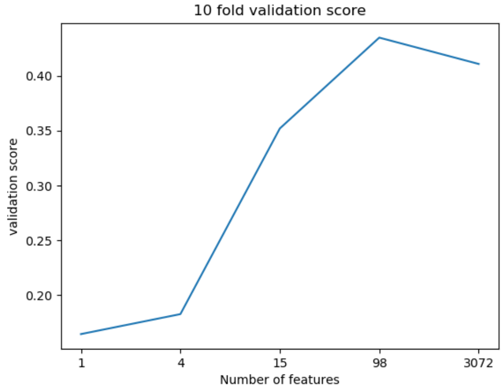
Results of f1 score are:

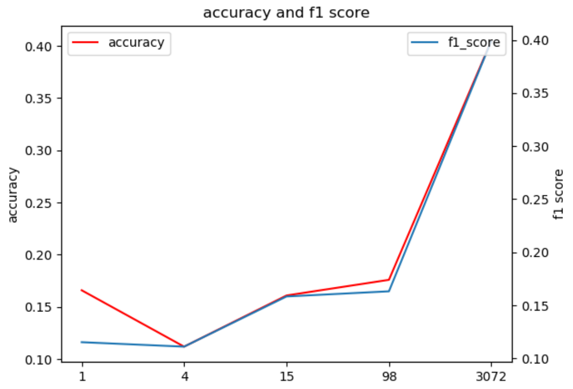
Classe	0	1	2	3	4	5	6	7	8	9
30%PCA data(1*1)	0.2994	0.0000	0.0000	0.0945	0.1053	0.1017	0.2979	0.0000	0.2033	0.0513
50%PCA data(4*4)	0.1290	0.1000	0.1347	0.1100	0.1220	0.0933	0.1333	0.1776	0.0704	0.0410
70%PCA data(15*15)	0.1379	0.1387	0.0702	0.1078	0.1053	0.0952	0.3905	0.2200	0.2192	0.0991
90%PCA data(98*98)	0.1566	0.0870	0.1060	0.0385	0.2041	0.1263	0.3306	0.2286	0.2745	0.0804
Full data (3072*3072)	0.4390	0.4727	0.4234	0.3163	0.1963	0.2591	0.4603	0.3793	0.5794	0.4667

The overall validation score, accuracy and f1_score of 5 models trained by 5 datasets are:

	30%PCA data(1*1)	50%PCA data(4*4)	70%PCA data(15*15)	90%PCA data(98*98)	Full data(3072*3072)
validation score	0.1646	0.1828	0.352	0.4348	0.4108
accuracy	0.166	0.112	0.161	0.176	0.405
F1_score_macro	0.115324	0.111124	0.158393	0.163242	0.399251

In this study, every figure in dataset only has one label. In this case, Micro f1 score is equal to accuracy. So Macro f1 score was used here to compare the performances of datasets with different features and so did precision and recall. All 'f1 score' in the following report refers to Macro f1 score.





Based on the results, generally speaking, dataset with 3072 features had the best performance. Consider validation score, it increased by the increase of feature number and reached highest in data set with 98 features. However, PCA reduced data sets performed badly on test set. Their accuracy and f1 score are below 0.2, which is half of the full dataset.

To get better performance on classification, different hidden layer size and learning rate were used to train the MLP. Four different hidden layer size(128,256,1024,2048) and 3 different initiate learning rate (0.01,0.001,0.0001) were chosen. MLP had three lays and other parameters are (activation='relu', max_iter=1000,solver='adam',learning_rate='adaptive').

Results of 10 fold validation score:

	learning rate init 0.01	0.001	0.0001
hidden layer size 128	0.3584	0.4018	0.3892
256	0.313	0.4028	0.398
1024	0.1038	0.397	0.4288
2048	0.104	0.4076	0.4352

Results of accuracy:

	learning rate init 0.01	0.001	0.0001
hidden layer size 128	0.3584	0.4018	0.3892
256	0.313	0.4028	0.398
1024	0.1038	0.397	0.4288
2048	0.104	0.4076	0.4352

Results of Macro f1 score:

	learning rate init 0.01	0.001	0.0001
hidden layer size 128	0.347849	0.411578	0.402921
256	0.336306	0.425536	0.403436
1024	0.016529	0.39656	0.412611
2048	0.016514	0.398426	0.443537

The precision, recall and f1_score on each class are:

```
Size=128.000000,learning_rate_init=0.010000
Results on test set:
            classes     precision     recall   f1-score    support
               0          0.47         0.33       0.39        103
               1          0.49         0.43       0.46         89
               2          0.32         0.20       0.25        100
               3          0.25         0.19       0.22        103
               4          0.18         0.18       0.18         90
               5          0.20         0.19       0.19         86
               6          0.32         0.65       0.43        112
               7          0.34         0.37       0.35        102
               8          0.54         0.56       0.55        106
               9         0.50         0.41       0.45        109
```

```
Size=128.000000,learning_rate_init=0.001000
Results on test set:
                  classes     precision       recall    f1-score      support
                      0          0.50          0.46          0.48           103
                      1          0.50          0.47          0.49            89
                      2          0.35          0.39          0.37           100
                      3          0.30          0.30          0.30           103
                      4          0.25          0.30          0.27            90
                      5          0.23          0.20          0.21            86
                      6          0.45          0.48          0.47           112
                      7          0.42          0.39          0.41           102
                      8          0.51          0.59          0.55           106
                      9            0.51          0.42          0.46           109

Size=128.000000,learning_rate_init=0.000100
Results on test set:
              classes     precision       recall    f1-score      support
                 0          0.45          0.40          0.42           103
                 1          0.51          0.45          0.48            89
                 2          0.33          0.35          0.34           100
                 3          0.27          0.30          0.29           103
                 4          0.28          0.26          0.27            90
                 5          0.27          0.27          0.27            86
                 6          0.51          0.51          0.51           112
                 7          0.47          0.46          0.47           102
                 8          0.50          0.62          0.56           106
                 9            0.47          0.46          0.46           109

Size=256.000000,learning_rate_init=0.010000
Results on test set:
           classes          precision       recall    f1-score      support
                 0          0.25          0.35          0.29           103
                 1          0.00          0.00          0.00            89
                 2          0.00          0.00          0.00           100
                 3          0.00          0.00          0.00           103
                 4          0.00          0.00          0.00            90
                 5          0.00          0.00          0.00            86
                 6          0.17          0.90          0.28           112
                 7          0.00          0.00          0.00           102
                 8          0.00          0.00          0.00           106
                 9            0.27          0.61          0.37           109

Size=256.000000,learning_rate_init=0.001000,
Results on test set:
```

```
          classes        precision      recall    f1-score     support
             0            0.39          0.33        0.36          103
             1            0.50          0.57        0.53           89
             2            0.32          0.43        0.37          100
             3            0.27          0.24        0.26          103
             4            0.25          0.20        0.22           90
             5            0.30          0.28        0.29           86
             6            0.55          0.49        0.52          112
             7            0.47          0.43        0.45          102
             8            0.44          0.68        0.54          106
             9            0.53          0.36        0.43          109
```

```
size=256.000000,learning_rate_init=0.000100
Results on test set:
          classes      precision      recall    f1-score     support
             0            0.45          0.43        0.44          103
             1            0.52          0.47        0.49           89
             2            0.32          0.36        0.34          100
             3            0.30          0.31        0.31          103
             4            0.25          0.27        0.26           90
             5            0.28          0.28        0.28           86
             6            0.47          0.47        0.47          112
             7            0.47          0.42        0.44          102
             8            0.52          0.61        0.56          106
             9            0.52          0.46        0.49          109
```

```
Size=1024.000000,learning_rate_init=0.010000
Results on test set:
          classes      precision      recall    f1-score     support
             0            0.00          0.00        0.00          103
             1            0.00          0.00        0.00           89
             2            0.10          1.00        0.18          100
             3            0.00          0.00        0.00          103
             4            0.00          0.00        0.00           90
             5            0.00          0.00        0.00           86
             6            0.00          0.00        0.00          112
             7            0.00          0.00        0.00          102
             8            0.00          0.00        0.00          106
             9            0.00          0.00        0.00          109
```

```
Size=1024.000000,learning_rate_init=0.001000
Results on test set:
            classes    precision      recall    f1-score     support
             0            0.47          0.45        0.46          103
```

```
              1         0.57       0.46       0.51         89
              2         0.47       0.28       0.35        100
              3         0.30       0.36       0.32        103
              4         0.30       0.22       0.25         90
              5         0.29       0.36       0.32         86
              6         0.52       0.54       0.53        112
              7         0.39       0.43       0.41        102
              8         0.51       0.68       0.58        106
              9         0.49       0.47       0.48        109

Size=1024.000000,learning_rate_init=0.000100
Results on test set:
           classes   precision     recall   f1-score    support
              0         0.51       0.47       0.49        103
              1         0.55       0.51       0.53         89
              2         0.37       0.38       0.37        100
              3         0.28       0.26       0.27        103
              4         0.30       0.36       0.33         90
              5         0.30       0.29       0.30         86
              6         0.51       0.54       0.53        112
              7         0.45       0.44       0.44        102
              8         0.55       0.63       0.59        106
              9         0.51       0.44       0.47        109

Size=2048.000000,learning_rate_init=0.010000
Results on test set:
           classes   precision     recall   f1-score    support
              0         0.00       0.00       0.00        103
              1         0.00       0.00       0.00         89
              2         0.00       0.00       0.00        100
              3         0.00       0.00       0.00        103
              4         0.00       0.00       0.00         90
              5         0.00       0.00       0.00         86
              6         0.00       0.00       0.00        112
              7         0.00       0.00       0.00        102
              8         0.11       1.00       0.19        106
              9         0.00       0.00       0.00        109

Size=2048.000000,learning_rate_init=0.001000
Results on test set:
                     precision     recall   f1-score    support
              0         0.40       0.45       0.42        103
              1         0.51       0.45       0.48         89
              2         0.25       0.36       0.29        100
```

3	0.26	0.28	0.27	103
4	0.27	0.32	0.30	90
5	0.35	0.26	0.30	86
6	0.49	0.46	0.47	112
7	0.53	0.42	0.47	102
8	0.53	0.63	0.58	106
9	0.61	0.39	0.47	109

Size=2048.000000,learning_rate_init=0.000100

Results on test set:

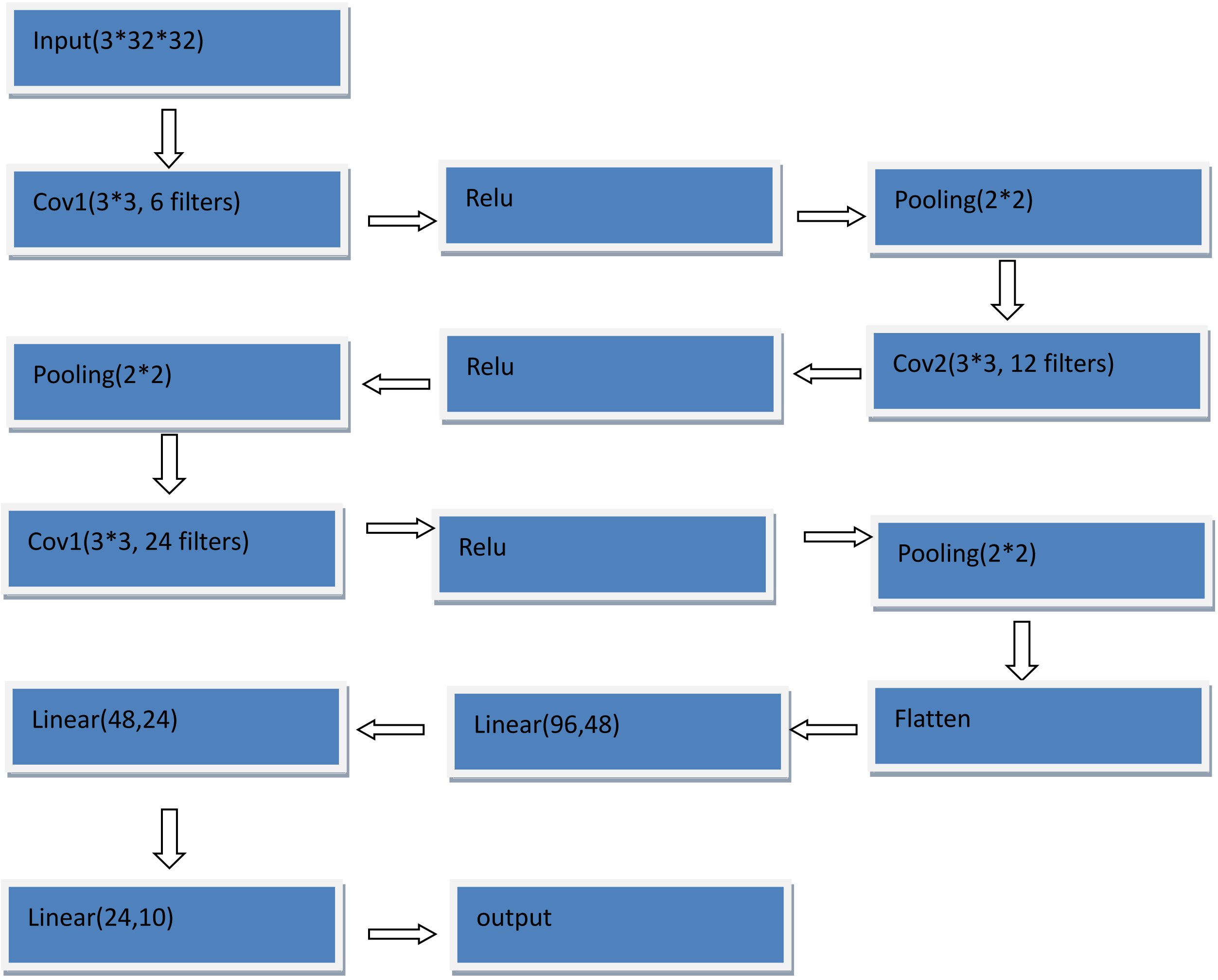
classes	precision	recall	f1-score	support
0	0.54	0.49	0.51	103
1	0.56	0.51	0.53	89
2	0.37	0.43	0.40	100
3	0.33	0.29	0.31	103
4	0.31	0.34	0.32	90
5	0.29	0.26	0.27	86
6	0.52	0.54	0.53	112
7	0.52	0.50	0.51	102
8	0.56	0.66	0.61	106
9	0.50	0.49	0.50	109

Based on the results, MLP model with hidden layer size=2048, Initial learning rate=0.0001 outperformed other models. It has highest validation score(0.4352) on trainset, highest accuracy (0.4352) and f1 score(0.4435) on test set. And in Task5, It will be compared with the best performing CNN.

5. CNN(Task 3)

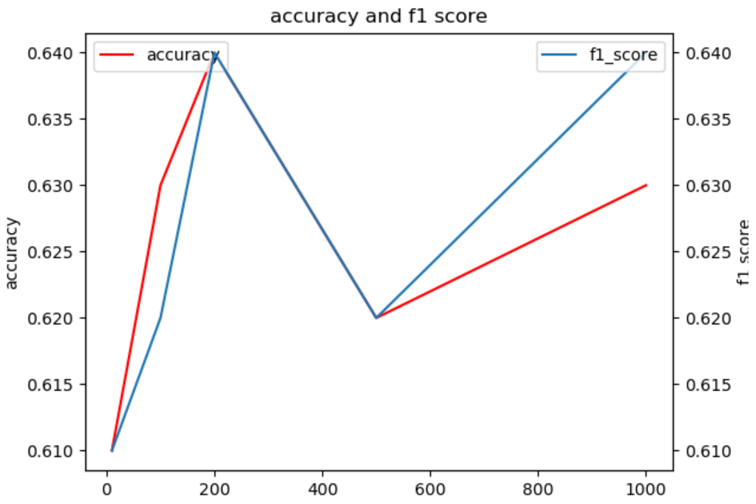
In this task, a CNN was designed to do multi-class classification. After the network design and parameter tuning, The best performing model contains 3 convolution layers, 3 pooling layers, 3 activation layers and 3 linear layers. Loss function is CrossEntropyLoss, optimizer is SGD, batch size is 8. The hole CIFAR-10 dataset was used(50000 train set, 10000 test set) to train the model.

Structure of CNN:



The design philosophy of this network was to use three convolution layers to extract features, a non-linearity followed every convolution layer to do discretization. And pooling layers were used to reduce features and make model invariance of local translation. After three convolution layers, the output had 24 channels. Then a flatten layer and three linear layers were used to reduce the size of output to 10. Since CrossEntropyLoss was chosen as loss function, softmax layer was needless.

With different training epochs, the performance of network changes, and the accuracy decreased after 200 epochs, which may be due to the overfitting of model.



When training epoch=200, model reaches the best performance:

classes	precision	recall	f1-score	support
0	0.66	0.69	0.68	1000
1	0.77	0.77	0.77	1000
2	0.58	0.47	0.52	1000
3	0.46	0.42	0.44	1000
4	0.56	0.60	0.58	1000
5	0.57	0.51	0.54	1000
6	0.65	0.78	0.71	1000
7	0.73	0.64	0.68	1000
8	0.69	0.78	0.73	1000
9	0.70	0.75	0.72	1000
accuracy			0.64	10000
macro avg	0.64	0.64	0.64	10000
weighted avg	0.64	0.64	0.64	10000

accuracy on train set: 0.71152

training time is 7126.88s

From the result of CNN on each class, CNN had the best classification performance in class 1(car) and worst performance in class3(cat).

6. Comparison between MLP and CNN (TASK 4)

To compare two kind of models, the entire cifar10 trainset(50000*3072) was used to train the MLP model with best performance parameters in task2, results on the entire test set were:

class	precision	recall	f1-score	support
0	0.55	0.72	0.62	1000
1	0.64	0.69	0.66	1000
2	0.45	0.38	0.41	1000

```
           3       0.39      0.34      0.36      1000
           4       0.54      0.42      0.48      1000
           5       0.46      0.50      0.48      1000
           6       0.60      0.63      0.61      1000
           7       0.60      0.64      0.62      1000
           8       0.68      0.69      0.68      1000
           9       0.63      0.57      0.60      1000

    accuracy                           0.56     10000
   macro avg       0.55      0.56      0.55     10000
weighted avg       0.55      0.56      0.55     10000
accuracy on train set: 0.99386
training time: 12954.574558973312
```

From the result of MLP on each class, MLP had the best classification performance on class 8(ship) and worst performance on class3(cat).

Compare the results of MLP and CNN, generally speaking, CNN had better performance on the test set. It had higher accuracy precision, recall and f1-score on all classes. The overfitting level of CNN was less than MLP. For CNN, the accuracy difference between train and test set was 0.07 and for MLP was 0.43. Additionally, the training time of CNN is only about 55% of MLP.

In conclusion, In this classification problem, CNN had better performance than MLP. Convolution layers make CNN possible to extract key features instead of analyzing every pixel of picture, which also reduce the time complexity and overfitting. Pooling layers make model invariance of local translation. These might be reasons why CNN performed better.