

# main

December 17, 2021

```
[10]: import pickle

from train_model import train_step, test_step
from utils.load_data import get_data
from utils.make_dict import train_bow, get_bow
```

```
[11]: args ={'dataset': 'cifar10',
            'dataroot': './data',
            'model': 'custom_SVM',
            'kernel': 'gaussian',
            'validation': 0.1,
            'C': 5.0,
            'sigma': 1.0,
            'batch': 1000,
            'dict_size': 100,
            'train': True,
            'load_cluster': False,
            'cuda': True
        }
```

```
[12]: hyper_C = [0.8, 0.9, 0.95, 1.0, 1.25]
hyper_sigma = [1.0, 1.25, 1.5]
```

```
[14]: trainX, trainy = get_data(dataset=args['dataset'], train=True,
    ↪ dataroot=args['dataroot'])

if args['dataset'] == 'cifar10':
    trainX = trainX.reshape((-1, 32, 32, 3), order='F')

if args['load_cluster']:
    with open("./cluster.dump", "rb") as f:
        cluster = pickle.load(f)
else:
    cluster = train_bow(trainX, num_dict=args['dict_size'], num_select=10000)
    with open("./cluster.dump", "wb") as f:
        pickle.dump(cluster, f)

trainFeature = get_bow(trainX, cluster, num_dict=args['dict_size'])
```

```

[15]: best_C = None
best_sigma = None
best_valid = 0.0

for C in hyper_C:
    for sigma in hyper_sigma:
        # Test hyperparameter
        args['C'] = C
        args['sigma'] = sigma

        # Get result
        _, train_acc_list, valid_acc_list = \
            train_step(args, trainFeature, trainy)

        # Evaluation parameter
        tra = sum(train_acc_list) / len(train_acc_list)
        val = sum(valid_acc_list) / len(valid_acc_list)

        if val > best_valid:
            best_valid = val
            best_C = C
            best_sigma = sigma

        # Print result
        print("C: %f Sigma: %f Train accuracy: %f Valid accuracy: %f"%(C,
↪sigma, tra, val))

print("Best C: %f Best sigma: %f"%(best_C, best_sigma))

```

```

100%|      | 10/10 [00:01<00:00,  5.05it/s]
C: 0.800000 Sigma: 1.000000 Train accuracy: 100.000000 Valid accuracy: 26.784000
100%|      | 10/10 [00:01<00:00,  5.90it/s]
C: 0.800000 Sigma: 1.250000 Train accuracy: 100.000000 Valid accuracy: 19.142000
100%|      | 10/10 [00:01<00:00,  5.68it/s]
C: 0.800000 Sigma: 1.500000 Train accuracy: 100.000000 Valid accuracy: 51.034000
100%|      | 10/10 [00:01<00:00,  5.81it/s]
C: 0.900000 Sigma: 1.000000 Train accuracy: 100.000000 Valid accuracy: 67.120000
100%|      | 10/10 [00:01<00:00,  5.72it/s]
C: 0.900000 Sigma: 1.250000 Train accuracy: 100.000000 Valid accuracy: 67.054000
100%|      | 10/10 [00:01<00:00,  5.63it/s]
C: 0.900000 Sigma: 1.500000 Train accuracy: 100.000000 Valid accuracy: 50.852000
100%|      | 10/10 [00:02<00:00,  4.59it/s]

```

```

C: 0.950000 Sigma: 1.000000 Train accuracy: 100.000000 Valid accuracy: 50.792000
100%|      | 10/10 [00:01<00:00,  5.73it/s]
C: 0.950000 Sigma: 1.250000 Train accuracy: 100.000000 Valid accuracy: 34.934000
100%|      | 10/10 [00:01<00:00,  5.75it/s]
C: 0.950000 Sigma: 1.500000 Train accuracy: 100.000000 Valid accuracy: 58.814000
100%|      | 10/10 [00:01<00:00,  5.86it/s]
C: 1.000000 Sigma: 1.000000 Train accuracy: 100.000000 Valid accuracy: 50.060000
100%|      | 10/10 [00:01<00:00,  5.79it/s]
C: 1.000000 Sigma: 1.250000 Train accuracy: 100.000000 Valid accuracy: 34.998000
100%|      | 10/10 [00:01<00:00,  5.78it/s]
C: 1.000000 Sigma: 1.500000 Train accuracy: 100.000000 Valid accuracy: 59.056000
100%|      | 10/10 [00:01<00:00,  5.89it/s]
C: 1.250000 Sigma: 1.000000 Train accuracy: 100.000000 Valid accuracy: 26.958000
100%|      | 10/10 [00:01<00:00,  5.91it/s]
C: 1.250000 Sigma: 1.250000 Train accuracy: 100.000000 Valid accuracy: 50.904000
100%|      | 10/10 [00:01<00:00,  5.90it/s]
C: 1.250000 Sigma: 1.500000 Train accuracy: 100.000000 Valid accuracy: 35.100000
Best C: 0.900000 Best sigma: 1.000000

```

```

[16]: args['C'] = best_C
      args['sigma'] = best_sigma
      args['part'] = False
      models, train_acc_list, valid_acc_list = \
          train_step(args, trainFeature, trainy)

```

```

100%|      | 10/10 [00:01<00:00,  5.87it/s]

```

```

[17]: testX, testy = get_data(dataset=args['dataset'], train=False,
    ↪ dataroot=args['dataroot'])
      if args['dataset'] == 'cifar10':
          testX = testX.reshape((-1, 32, 32, 3), order='F')
      testFeature = get_bow(testX, cluster, num_dict=args['dict_size'])

```

```

[18]: test_acc_list = test_step(args, testFeature, testy, models)

```

```

90%|      | 9/10 [00:01<00:00,  5.61it/s]

```

```

[19]: print("Test average accuracy:", sum(test_acc_list) / len(test_acc_list))

```

```

Test average accuracy: 74.0

```