In [1]:
```
import os, sys, torch, argparse, datetime, importlib, numpy as np
sys.path.append('utils')
sys.path.append('models')
from sklearn.model_selection import GridSearchCV, RandomizedSearchC
V
from ModelNetDataLoader import General_CLSDataLoader_HDF5
from Torch_Utility import copy_parameters
from torch.utils.data import DataLoader
from sklearn.preprocessing import scale
from Dataset_Loc import Dataset_Loc
from sklearn import svm, metrics
from tqdm import tqdm
```

In [2]:
```
os.environ["CUDA_DEVICE_ORDER"] = "PCI_BUS_ID"
os.environ["CUDA_VISIBLE_DEVICES"] = '2'
```

In [3]:
```
datasets = ['modelnet40']

MODEL = importlib.import_module('pointnet_util')
device = torch.device("cuda" if torch.cuda.is_available() else "cpu
")
encoder = MODEL.encoder(num_channel=3).to(device)
encoder = torch.nn.DataParallel(encoder)

for dataset in datasets:

    accs = []
    _, TRAIN_FILES, TEST_FILES = Dataset_Loc(dataset=dataset, fname
='', partial=False)
    TRAIN_DATASET = General_CLSDataLoader_HDF5(file_list=TRAIN_FILE
S)
    TEST_DATASET = General_CLSDataLoader_HDF5(file_list=TEST_FILES)
    trainDataLoader = DataLoader(TRAIN_DATASET, batch_size=32, shuf
fle=True, num_workers=4)
    testDataLoader = DataLoader(TEST_DATASET, batch_size=32, shuffl
e=True, num_workers=4)

    for epoch in range(1, 51, 2):
        checkpoint = torch.load('log/completion/modelnet_pointnet_v
anilla/checkpoints/model_epoch_%d.pth' % epoch)
        encoder = copy_parameters(encoder, checkpoint, verbose=Fals
e)

        X_train, y_train, X_test, y_test = [], [], [], []
        with torch.no_grad():
            encoder.eval()

            for points, target in tqdm(trainDataLoader, total=len(t
rainDataLoader), smoothing=0.9):
                points, target = points.float().transpose(2, 1).cud
```

```
a(), target.long().cuda()
                feats = encoder(points)
                X_train.append(feats.cpu().numpy())
                y_train.append(target.cpu().numpy())

            for points, target in tqdm(testDataLoader, total=len(te
stDataLoader), smoothing=0.9):
                points, target = points.float().transpose(2, 1).cud
a(), target.long().cuda()
                feats = encoder(points)
                X_test.append(feats.cpu().numpy())
                y_test.append(target.cpu().numpy())

        X_train, y_train = np.concatenate(X_train), np.concatenate(
y_train)
        X_test, y_test = np.concatenate(X_test), np.concatenate(y_t
est)

        linear_svm = svm.SVC(kernel='linear')
        linear_svm.fit(X_train[:10000], y_train[:10000])
        y_pred = linear_svm.predict(X_test)
        print("Epoch: ", epoch, "Simple Linear SVC accuracy: ", met
rics.accuracy_score(y_test, y_pred))

        accs.append(metrics.accuracy_score(y_test, y_pred))
    # np.save("svm/%s_oc_pointnet_svm.npy" % dataset, accs)
```

```
Number of Objects:  9840

  0%|          | 0/308 [00:00<?, ?it/s]

Number of Objects:  2468

100%|██████████| 308/308 [00:23<00:00, 13.31it/s]
100%|██████████| 78/78 [00:06<00:00, 12.56it/s]

Epoch:  1 Simple Linear SVC accuracy:  0.7382495948136143

100%|██████████| 308/308 [00:25<00:00, 13.43it/s]
100%|██████████| 78/78 [00:07<00:00, 11.01it/s]
  0%|          | 0/308 [00:00<?, ?it/s]

Epoch:  3 Simple Linear SVC accuracy:  0.8537277147487844

100%|██████████| 308/308 [00:25<00:00, 16.65it/s]
100%|██████████| 78/78 [00:04<00:00, 16.44it/s]

Epoch:  5 Simple Linear SVC accuracy:  0.8602106969205835

100%|██████████| 308/308 [00:22<00:00, 13.72it/s]
100%|██████████| 78/78 [00:07<00:00, 10.79it/s]

Epoch:  7 Simple Linear SVC accuracy:  0.8622366288492707

100%|██████████| 308/308 [00:21<00:00, 14.08it/s]
100%|██████████| 78/78 [00:06<00:00, 11.25it/s]
```

```
Epoch:  9 Simple Linear SVC accuracy:  0.8622366288492707

100%|██████████| 308/308 [00:22<00:00, 15.76it/s]
100%|██████████| 78/78 [00:06<00:00, 11.35it/s]

Epoch:  11 Simple Linear SVC accuracy:  0.8585899513776337

100%|██████████| 308/308 [00:23<00:00, 12.75it/s]
100%|██████████| 78/78 [00:06<00:00, 12.31it/s]
  0%|          | 0/308 [00:00<?, ?it/s]

Epoch:  13 Simple Linear SVC accuracy:  0.8675040518638574

100%|██████████| 308/308 [00:21<00:00, 14.46it/s]
100%|██████████| 78/78 [00:05<00:00, 17.75it/s]

Epoch:  15 Simple Linear SVC accuracy:  0.8634521880064829

100%|██████████| 308/308 [00:23<00:00, 18.43it/s]
100%|██████████| 78/78 [00:04<00:00, 15.77it/s]

Epoch:  17 Simple Linear SVC accuracy:  0.8610210696920584

100%|██████████| 308/308 [00:22<00:00, 13.60it/s]
100%|██████████| 78/78 [00:06<00:00, 11.82it/s]
```

```
---------------------------------------------------------------------
---------
KeyboardInterrupt                         Traceback (most recent c
all last)
<ipython-input-3-a459526ddaac> in <module>
     39
     40         linear_svm = svm.SVC(kernel='linear')
---> 41         linear_svm.fit(X_train[:10000], y_train[:10000])
     42         y_pred = linear_svm.predict(X_test)
     43         print("Epoch: ", epoch, "Simple Linear SVC accurac
y: ", metrics.accuracy_score(y_test, y_pred))

/scratch/hw501/anaconda3/lib/python3.7/site-packages/sklearn/svm/b
ase.py in fit(self, X, y, sample_weight)
    207
    208         seed = rnd.randint(np.iinfo('i').max)
--> 209         fit(X, y, sample_weight, solver_type, kernel,
random_seed=seed)
    210         # see comment on the other call to np.iinfo in thi
s file
    211

/scratch/hw501/anaconda3/lib/python3.7/site-packages/sklearn/svm/b
ase.py in _dense_fit(self, X, y, sample_weight, solver_type, kerne
l, random_seed)
    266                 cache_size=self.cache_size, coef0=self.coe
f0,
    267                 gamma=self._gamma, epsilon=self.epsilon,
--> 268                 max_iter=self.max_iter, random_seed=random
_seed)
    269
    270         self._warn_from_fit_status()

KeyboardInterrupt:
```

In [ ]:

```
datasets = ['modelnet40']

MODEL = importlib.import_module('pointnet_util')
device = torch.device("cuda" if torch.cuda.is_available() else "cpu
")
encoder = MODEL.encoder(num_channel=3).to(device)
encoder = torch.nn.DataParallel(encoder)

for dataset in datasets:

    accs = []
    _, TRAIN_FILES, TEST_FILES = Dataset_Loc(dataset=dataset, fname
='', partial=False)
    TRAIN_DATASET = General_CLSDataLoader_HDF5(file_list=TRAIN_FILE
S)
    TEST_DATASET = General_CLSDataLoader_HDF5(file_list=TEST_FILES)
    trainDataLoader = DataLoader(TRAIN_DATASET, batch_size=32, shuf
fle=True, num_workers=4)
```

```
    testDataLoader = DataLoader(TEST_DATASET, batch_size=32, shuffl
e=True, num_workers=4)

    for epoch in range(1, 51):
        checkpoint = torch.load('log/completion/modelnet_pointnet_v
anilla/checkpoints/model_epoch_%d.pth' % epoch)
        encoder = copy_parameters(encoder, checkpoint, verbose=Fals
e)

        X_train, y_train, X_test, y_test = [], [], [], []
        with torch.no_grad():
            encoder.eval()

            for points, target in tqdm(trainDataLoader, total=len(t
rainDataLoader), smoothing=0.9):
                points, target = points.float().transpose(2, 1).cud
a(), target.long().cuda()
                feats = encoder(points)
                X_train.append(feats.cpu().numpy())
                y_train.append(target.cpu().numpy())

            for points, target in tqdm(testDataLoader, total=len(te
stDataLoader), smoothing=0.9):
                points, target = points.float().transpose(2, 1).cud
a(), target.long().cuda()
                feats = encoder(points)
                X_test.append(feats.cpu().numpy())
                y_test.append(target.cpu().numpy())

        X_train, y_train = np.concatenate(X_train), np.concatenate(
y_train)
        X_test, y_test = np.concatenate(X_test), np.concatenate(y_t
est)

        linear_svm = svm.SVC(kernel='linear')
        linear_svm.fit(X_train[:10000], y_train[:10000])
        y_pred = linear_svm.predict(X_test)
        print("Epoch: ", epoch, "Simple Linear SVC accuracy: ", met
rics.accuracy_score(y_test, y_pred))

        accs.append(metrics.accuracy_score(y_test, y_pred))
```

```
Number of Objects:  9840
Number of Objects:  2468

100%|██████████| 308/308 [00:20<00:00, 15.36it/s]
100%|██████████| 78/78 [00:06<00:00, 12.06it/s]

Epoch:  1 Simple Linear SVC accuracy:  0.7382495948136143

100%|██████████| 308/308 [00:21<00:00, 17.84it/s]
100%|██████████| 78/78 [00:06<00:00, 11.80it/s]

Epoch:  2 Simple Linear SVC accuracy:  0.8537277147487844
```

```
100%|██████████| 308/308 [00:26<00:00, 11.68it/s]
100%|██████████| 78/78 [00:06<00:00, 12.58it/s]

Epoch:  3 Simple Linear SVC accuracy:  0.8537277147487844

100%|██████████| 308/308 [00:16<00:00, 12.93it/s]
100%|██████████| 78/78 [00:06<00:00, 11.65it/s]
  0%|          | 0/308 [00:00<?, ?it/s]

Epoch:  4 Simple Linear SVC accuracy:  0.8581847649918962

100%|██████████| 308/308 [00:16<00:00, 22.69it/s]
100%|██████████| 78/78 [00:06<00:00, 12.92it/s]

Epoch:  5 Simple Linear SVC accuracy:  0.8602106969205835

100%|██████████| 308/308 [00:23<00:00, 13.29it/s]
100%|██████████| 78/78 [00:06<00:00, 12.17it/s]

Epoch:  6 Simple Linear SVC accuracy:  0.8634521880064829

100%|██████████| 308/308 [00:17<00:00, 17.12it/s]
100%|██████████| 78/78 [00:06<00:00, 11.71it/s]

Epoch:  7 Simple Linear SVC accuracy:  0.8622366288492707

100%|██████████| 308/308 [00:22<00:00, 13.97it/s]
100%|██████████| 78/78 [00:06<00:00, 11.89it/s]

Epoch:  8 Simple Linear SVC accuracy:  0.8606158833063209

100%|██████████| 308/308 [00:25<00:00, 11.91it/s]
100%|██████████| 78/78 [00:06<00:00, 12.70it/s]

Epoch:  9 Simple Linear SVC accuracy:  0.8622366288492707

100%|██████████| 308/308 [00:23<00:00, 13.31it/s]
100%|██████████| 78/78 [00:06<00:00, 16.77it/s]

Epoch:  10 Simple Linear SVC accuracy:  0.8589951377633711

100%|██████████| 308/308 [00:21<00:00, 14.30it/s]
100%|██████████| 78/78 [00:05<00:00, 13.33it/s]

Epoch:  11 Simple Linear SVC accuracy:  0.8585899513776337

100%|██████████| 308/308 [00:24<00:00, 14.44it/s]
100%|██████████| 78/78 [00:06<00:00, 11.57it/s]

Epoch:  12 Simple Linear SVC accuracy:  0.8618314424635333

100%|██████████| 308/308 [00:24<00:00, 12.55it/s]
100%|██████████| 78/78 [00:05<00:00, 14.33it/s]

Epoch:  13 Simple Linear SVC accuracy:  0.8675040518638574

100%|██████████| 308/308 [00:23<00:00, 12.94it/s]
100%|██████████| 78/78 [00:06<00:00, 12.04it/s]
```

```
Epoch:  14 Simple Linear SVC accuracy:  0.8545380875202593

100%|██████████| 308/308 [00:23<00:00, 12.90it/s]
100%|██████████| 78/78 [00:04<00:00, 18.70it/s]

Epoch:  15 Simple Linear SVC accuracy:  0.8634521880064829

100%|██████████| 308/308 [00:22<00:00, 13.94it/s]
100%|██████████| 78/78 [00:06<00:00, 11.80it/s]

Epoch:  16 Simple Linear SVC accuracy:  0.8565640194489466

100%|██████████| 308/308 [00:19<00:00, 15.99it/s]
100%|██████████| 78/78 [00:06<00:00, 12.64it/s]

Epoch:  17 Simple Linear SVC accuracy:  0.8610210696920584

100%|██████████| 308/308 [00:24<00:00, 16.45it/s]
100%|██████████| 78/78 [00:04<00:00, 17.81it/s]

Epoch:  18 Simple Linear SVC accuracy:  0.8557536466774717

100%|██████████| 308/308 [00:24<00:00, 12.36it/s]
100%|██████████| 78/78 [00:04<00:00, 16.94it/s]

Epoch:  19 Simple Linear SVC accuracy:  0.8573743922204214

100%|██████████| 308/308 [00:25<00:00, 17.90it/s]
100%|██████████| 78/78 [00:03<00:00, 22.21it/s]

Epoch:  20 Simple Linear SVC accuracy:  0.8610210696920584

100%|██████████| 308/308 [00:22<00:00, 13.68it/s]
100%|██████████| 78/78 [00:07<00:00, 10.61it/s]

Epoch:  21 Simple Linear SVC accuracy:  0.8618314424635333

100%|██████████| 308/308 [00:25<00:00, 12.08it/s]
100%|██████████| 78/78 [00:06<00:00, 11.65it/s]

Epoch:  22 Simple Linear SVC accuracy:  0.8618314424635333

100%|██████████| 308/308 [00:24<00:00, 16.00it/s]
100%|██████████| 78/78 [00:05<00:00, 14.24it/s]

Epoch:  23 Simple Linear SVC accuracy:  0.8630470016207455

100%|██████████| 308/308 [00:22<00:00, 13.54it/s]
100%|██████████| 78/78 [00:07<00:00, 10.98it/s]

Epoch:  24 Simple Linear SVC accuracy:  0.8626418152350082

100%|██████████| 308/308 [00:23<00:00, 25.31it/s]
100%|██████████| 78/78 [00:06<00:00, 15.44it/s]

Epoch:  25 Simple Linear SVC accuracy:  0.8585899513776337
```

```
100%|██████████| 308/308 [00:25<00:00, 12.17it/s]
100%|██████████| 78/78 [00:06<00:00, 12.96it/s]

Epoch:  26 Simple Linear SVC accuracy:  0.8630470016207455

100%|██████████| 308/308 [00:17<00:00, 17.87it/s]
100%|██████████| 78/78 [00:05<00:00, 14.01it/s]

Epoch:  27 Simple Linear SVC accuracy:  0.8658833063209076

100%|██████████| 308/308 [00:23<00:00, 13.09it/s]
100%|██████████| 78/78 [00:05<00:00, 14.92it/s]

Epoch:  28 Simple Linear SVC accuracy:  0.859805510534846

100%|██████████| 308/308 [00:21<00:00, 14.65it/s]
100%|██████████| 78/78 [00:06<00:00, 12.67it/s]

Epoch:  29 Simple Linear SVC accuracy:  0.8577795786061588

100%|██████████| 308/308 [00:22<00:00, 13.87it/s]
100%|██████████| 78/78 [00:05<00:00, 19.40it/s]

Epoch:  30 Simple Linear SVC accuracy:  0.8565640194489466

100%|██████████| 308/308 [00:17<00:00, 17.50it/s]
100%|██████████| 78/78 [00:04<00:00, 15.72it/s]

Epoch:  31 Simple Linear SVC accuracy:  0.8638573743922204

100%|██████████| 308/308 [00:22<00:00, 13.73it/s]
100%|██████████| 78/78 [00:06<00:00, 11.78it/s]

Epoch:  32 Simple Linear SVC accuracy:  0.8626418152350082

100%|██████████| 308/308 [00:25<00:00, 13.00it/s]
100%|██████████| 78/78 [00:07<00:00, 10.80it/s]

Epoch:  33 Simple Linear SVC accuracy:  0.8610210696920584

100%|██████████| 308/308 [00:22<00:00, 13.46it/s]
100%|██████████| 78/78 [00:05<00:00, 13.16it/s]

Epoch:  34 Simple Linear SVC accuracy:  0.8557536466774717

100%|██████████| 308/308 [00:23<00:00, 13.67it/s]
100%|██████████| 78/78 [00:06<00:00, 12.02it/s]
```

In [ ]:

In [ ]:

In [ ]:
```
datasets = ['modelnet40', 'shapenet8']

MODEL = importlib.import_module('pointnet_util')
device = torch.device("cuda" if torch.cuda.is_available() else "cpu
```

```
")
encoder = MODEL.encoder(num_channel=3).to(device)
encoder = torch.nn.DataParallel(encoder)

for dataset in datasets:

    accs = []
    _, TRAIN_FILES, TEST_FILES = Dataset_Loc(dataset=dataset, fname
='', partial=True, bn=True)
    TRAIN_DATASET = General_CLSDataLoader_HDF5(file_list=TRAIN_FILE
S)
    TEST_DATASET = General_CLSDataLoader_HDF5(file_list=TEST_FILES)
    trainDataLoader = DataLoader(TRAIN_DATASET, batch_size=32, shuf
fle=True, num_workers=4)
    testDataLoader = DataLoader(TEST_DATASET, batch_size=32, shuffl
e=True, num_workers=4)

    for epoch in range(1, 51, 10):
        checkpoint = torch.load('log/completion/modelnet_pointnet_v
anilla/checkpoints/model_epoch_%d.pth' % epoch)
        encoder = copy_parameters(encoder, checkpoint, verbose=Fals
e)

        X_train, y_train, X_test, y_test = [], [], [], []
        with torch.no_grad():
            encoder.eval()

            for points, target in tqdm(trainDataLoader, total=len(t
rainDataLoader), smoothing=0.9):
                points, target = points.float().transpose(2, 1).cud
a(), target.long().cuda()
                feats = encoder(points)
                X_train.append(feats.cpu().numpy())
                y_train.append(target.cpu().numpy())

            for points, target in tqdm(testDataLoader, total=len(te
stDataLoader), smoothing=0.9):
                points, target = points.float().transpose(2, 1).cud
a(), target.long().cuda()
                feats = encoder(points)
                X_test.append(feats.cpu().numpy())
                y_test.append(target.cpu().numpy())

        X_train, y_train = np.concatenate(X_train), np.concatenate(
y_train)
        X_test, y_test = np.concatenate(X_test), np.concatenate(y_t
est)

        linear_svm = svm.SVC(kernel='linear')
        linear_svm.fit(X_train[:10000], y_train[:10000])
        y_pred = linear_svm.predict(X_test)
        print("Epoch: ", epoch, "Simple Linear SVC accuracy: ", met
rics.accuracy_score(y_test, y_pred))
```

```
        accs.append(metrics.accuracy_score(y_test, y_pred))
    np.save("svm/%s_oc_pointnet_svm.npy" % dataset, accs)
```

```
Number of Objects:  98430

  0%|          | 0/3076 [00:00<?, ?it/s]

Number of Objects:  24680

100%|██████████| 3076/3076 [01:38<00:00, 31.25it/s]
100%|██████████| 772/772 [00:18<00:00, 41.20it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  1 Simple Linear SVC accuracy:  0.6521069692058347

100%|██████████| 3076/3076 [01:13<00:00, 42.01it/s]
100%|██████████| 772/772 [00:19<00:00, 40.42it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  2 Simple Linear SVC accuracy:  0.7919773095623988

100%|██████████| 3076/3076 [01:14<00:00, 41.12it/s]
100%|██████████| 772/772 [00:17<00:00, 44.24it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  3 Simple Linear SVC accuracy:  0.7967179902755267

100%|██████████| 3076/3076 [01:12<00:00, 42.45it/s]
100%|██████████| 772/772 [00:19<00:00, 38.62it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  4 Simple Linear SVC accuracy:  0.8000810372771475

100%|██████████| 3076/3076 [01:13<00:00, 41.88it/s]
100%|██████████| 772/772 [00:18<00:00, 42.74it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  5 Simple Linear SVC accuracy:  0.8006482982171799

100%|██████████| 3076/3076 [01:14<00:00, 41.22it/s]
100%|██████████| 772/772 [00:18<00:00, 42.62it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  6 Simple Linear SVC accuracy:  0.8005672609400324

100%|██████████| 3076/3076 [01:10<00:00, 43.69it/s]
100%|██████████| 772/772 [00:13<00:00, 55.97it/s]
  0%|          | 0/3076 [00:00<?, ?it/s]

Epoch:  7 Simple Linear SVC accuracy:  0.8022690437601296

100%|██████████| 3076/3076 [01:11<00:00, 43.12it/s]
100%|██████████| 772/772 [00:19<00:00, 40.37it/s]
```

In [ ]: