

model-graphconv

March 18, 2024

```
[1]: !pip install torch-geometric
```

Collecting torch-geometric

Downloading torch_geometric-2.5.1-py3-none-any.whl (1.1 MB)

1.1/1.1 MB

11.0 MB/s eta 0:00:00

Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (4.66.2)

Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (1.25.2)

Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (1.11.4)

Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (2023.6.0)

Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (3.1.3)

Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (3.9.3)

Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (2.31.0)

Requirement already satisfied: pyparsing in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (3.1.2)

Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (1.2.2)

Requirement already satisfied: psutil>=5.8.0 in /usr/local/lib/python3.10/dist-packages (from torch-geometric) (5.9.5)

Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (1.3.1)

Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (23.2.0)

Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (1.4.1)

Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (6.0.5)

Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (1.9.4)

Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->torch-geometric) (4.0.3)

Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch-geometric) (2.1.5)
 Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->torch-geometric) (3.3.2)
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->torch-geometric) (3.6)
 Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->torch-geometric) (2.0.7)
 Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->torch-geometric) (2024.2.2)
 Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->torch-geometric) (1.3.2)
 Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->torch-geometric) (3.3.0)
 Installing collected packages: torch-geometric
 Successfully installed torch-geometric-2.5.1

```
[2]: from google.colab import drive
      drive.mount('/content/drive')
```

Mounted at /content/drive

```
[3]: import torch
      chunk1 = torch.load('/content/drive/MyDrive/SCI_data/first.pt')
```

```
[4]: chunk2 = torch.load('/content/drive/MyDrive/SCI_data/second (1).pt')
```

```
[5]: # chunk3 = torch.load('/content/drive/MyDrive/SCI_data/third.pt')
```

```
[6]: chunk1+=chunk2
      # chunk1+=chunk3
```

```
[7]: len(chunk1)
```

```
[7]: 18000
```

```
[8]: from torch.utils.data import Dataset

      class GraphDataset(Dataset):
          def __init__(self, data, preprocess):
              self.data = data
              self.preprocess = preprocess

          def __len__(self,):
              return len(self.data)
```

```

def __getitem__(self, index):
    res = self.data[index]
    for p in self.preprocess:
        res = p(res)
    return res

```

```
# dataset = GraphDataset(chunk1, [drop_self_edges, drop_nodes_with_no_edges])
```

```

[9]: from sklearn.model_selection import train_test_split
rand_seed = 42
X_train, X_test = train_test_split(chunk1, test_size=0.1, random_state =
    ↪rand_seed)
X_train, X_val = train_test_split(X_train, test_size=0.1, random_state =
    ↪rand_seed)
print(len(X_train), len(X_val), len(X_val))

```

```
14580 1620 1620
```

```
[10]: device = 'cuda'
```

```

[11]: from torch_geometric.loader import DataLoader

train_loader = DataLoader(X_train, batch_size=32, shuffle=True)
val_loader = DataLoader(X_val, batch_size=32, shuffle=False)
test_loader = DataLoader(X_test, batch_size=32, shuffle=False)

```

```
[12]: !pip install torchmetrics
```

```
Collecting torchmetrics
```

```
  Downloading torchmetrics-1.3.1-py3-none-any.whl (840 kB)
```

```
      840.4/840.4
```

```
kB 11.3 MB/s eta 0:00:00
```

```
Requirement already satisfied: numpy>1.20.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from torchmetrics) (1.25.2)
```

```
Requirement already satisfied: packaging>17.1 in /usr/local/lib/python3.10/dist-
packages (from torchmetrics) (24.0)
```

```
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-
packages (from torchmetrics) (2.2.1+cu121)
```

```
Collecting lightning-utilities>=0.8.0 (from torchmetrics)
```

```
  Downloading lightning_utilities-0.10.1-py3-none-any.whl (24 kB)
```

```
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from lightning-utilities>=0.8.0->torchmetrics) (67.7.2)
```

```
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.10/dist-packages (from lightning-
utilities>=0.8.0->torchmetrics) (4.10.0)
```

Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->torchmetrics) (3.13.1)

Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->torchmetrics) (1.12)

Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->torchmetrics) (3.2.1)

Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->torchmetrics) (3.1.3)

Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->torchmetrics) (2023.6.0)

Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)

23.7/23.7 MB

24.9 MB/s eta 0:00:00

Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)

823.6/823.6

kB 61.0 MB/s eta 0:00:00

Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)

14.1/14.1 MB

86.8 MB/s eta 0:00:00

Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)

731.7/731.7

MB 747.2 kB/s eta 0:00:00

Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)

410.6/410.6

MB 2.9 MB/s eta 0:00:00

Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)

121.6/121.6

MB 10.0 MB/s eta 0:00:00

Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.10.0->torchmetrics)

 Downloading nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5

MB)

56.5/56.5 MB

16.6 MB/s eta 0:00:00

Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.10.0->torchmetrics)

Downloading nvidia_cusolver_cu12-11.4.5.107-py3-none-manylinux1_x86_64.whl (124.2 MB)

124.2/124.2

MB 10.6 MB/s eta 0:00:00

Collecting nvidia-cuspars-cu12==12.1.0.106 (from torch>=1.10.0->torchmetrics)

Downloading nvidia_cuspars_cu12-12.1.0.106-py3-none-manylinux1_x86_64.whl (196.0 MB)

196.0/196.0

MB 2.3 MB/s eta 0:00:00

Collecting nvidia-nccl-cu12==2.19.3 (from torch>=1.10.0->torchmetrics)

Downloading nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl (166.0 MB)

166.0/166.0

MB 2.3 MB/s eta 0:00:00

Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.10.0->torchmetrics)

Downloading nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)

99.1/99.1 kB

14.1 MB/s eta 0:00:00

Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->torchmetrics) (2.2.0)

Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.10.0->torchmetrics)

Downloading nvidia_nvjitlink_cu12-12.4.99-py3-none-manylinux2014_x86_64.whl (21.1 MB)

21.1/21.1 MB

31.0 MB/s eta 0:00:00

Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->torchmetrics) (2.1.5)

Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->torchmetrics) (1.3.0)

Installing collected packages: nvidia-nvtx-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-cublas-cu12, lightning-utilities, nvidia-cuspars-cu12, nvidia-cudnn-cu12, nvidia-cusolver-cu12, torchmetrics

Successfully installed lightning-utilities-0.10.1 nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105 nvidia-cudnn-cu12-8.9.2.26 nvidia-cufft-cu12-11.0.2.54 nvidia-curand-cu12-10.3.2.106 nvidia-cusolver-cu12-11.4.5.107 nvidia-cuspars-

cu12-12.1.0.106 nvidia-nccl-cu12-2.19.3 nvidia-nvjitlink-cu12-12.4.99 nvidia-nvtx-cu12-12.1.105 torchmetrics-1.3.1

```
[16]: import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torch_geometric.nn import GCNConv, global_mean_pool
from torch_geometric.nn.conv import GraphConv
from torch_geometric.utils import to_undirected
from torch_geometric.data import DataLoader
from torchmetrics.classification import BinaryAUROC
auroc = BinaryAUROC()

class Network(nn.Module):
    def __init__(self, c_in, c_hidden, c_out, p=0.3):
        super().__init__()
        torch.manual_seed(123)
        self.conv1 = GraphConv(c_in, c_hidden)
        self.conv2 = GraphConv(c_hidden, 3*c_hidden)

        self.conv3 = GraphConv(3*c_hidden, c_hidden)

        # self.pool = SAGPooling(c_hidden)

        self.lin1 = nn.Linear(c_hidden, 4*c_out)
        self.lin2 = nn.Linear(4*c_out, c_out)
        self.p = p

    def forward(self, x, edge_index, batch, is_train):
        x = self.conv1(x, edge_index)
        x = x.relu()

        x = self.conv2(x, edge_index)
        x = x.relu()

        x = self.conv3(x, edge_index)
        x = x.relu()

        x = global_mean_pool(x, batch)

        # classifier

        x = F.dropout(x, p=self.p, training=is_train)
        x = self.lin1(x)
```

```

        x = F.dropout(x, p=self.p, training=is_train)
        x = self.lin2(x)

    return x

def evaluate(loader):
    model.eval()
    total_loss = 0.0
    correct = 0
    total_samples = 0
    all_preds = []
    all_labels = []

    with torch.no_grad():
        for batch in loader:
            batch.to(device)
            # print(batch.edge_index)
            pred = model(batch.x.float(), batch.edge_index, batch.batch, False)
            target = F.one_hot(batch.y, 2).float()
            loss = criterion(pred, target)
            total_loss += loss.item()

            # Calculate accuracy
            pred_labels = torch.softmax(pred, -1).argmax(dim=-1)
            correct += (pred_labels == batch.y).sum().item()
            total_samples += len(batch.y)
            all_labels.append(batch.y)
            all_preds.append(pred_labels)

    pred = all_preds[0]
    label = all_labels[0]

    for p, l in zip(all_preds[1:], all_labels[1:]):
        pred = torch.cat([pred, p])
        label = torch.cat([label, l])

    return total_loss / len(loader), correct / total_samples, auroc(pred, label)

# Training loop with validation
num_epochs = 50
best_auroc = 0
model = Network(c_in=5, c_hidden=64, c_out=2).to(device)
optimizer = optim.Adam(model.parameters(), lr=3e-4)
criterion = nn.BCEWithLogitsLoss()

for epoch in range(num_epochs):

```

```

model.train()
epoch_loss = 0

for idx, batch in enumerate(train_loader):

    batch = batch.to(device)

    pred = model(batch.x.float(), batch.edge_index, batch.batch, True)
    target = F.one_hot(batch.y, 2).float()
    loss = criterion(pred, target)
    epoch_loss += loss.item()

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

avg_train_loss = epoch_loss / len(train_loader)
avg_val_loss, val_accuracy, val_auroc = evaluate(val_loader)
if val_auroc > best_auroc:
    best_auroc = val_auroc
    best_epoch = epoch
    torch.save(model.state_dict(), 'best_model_gnn.pth')
print(f'Epoch {epoch + 1}/{num_epochs}, Train Loss: {avg_train_loss:.4f},
↪Val Loss: {avg_val_loss:.4f}, Val Accuracy: {val_accuracy:.4f}, Val AUROC:
↪{val_auroc:.4f}')

```

```

Epoch 1/50, Train Loss: 85.6892, Val Loss: 0.6639, Val Accuracy: 0.6093, Val
AUROC: 0.6147
Epoch 2/50, Train Loss: 0.6774, Val Loss: 0.6755, Val Accuracy: 0.6722, Val
AUROC: 0.6714
Epoch 3/50, Train Loss: 0.6538, Val Loss: 0.6288, Val Accuracy: 0.6870, Val
AUROC: 0.6872
Epoch 4/50, Train Loss: 0.6482, Val Loss: 0.6108, Val Accuracy: 0.6907, Val
AUROC: 0.6909
Epoch 5/50, Train Loss: 0.6276, Val Loss: 0.6154, Val Accuracy: 0.6747, Val
AUROC: 0.6724
Epoch 6/50, Train Loss: 0.6304, Val Loss: 0.6075, Val Accuracy: 0.6833, Val
AUROC: 0.6831
Epoch 7/50, Train Loss: 0.6268, Val Loss: 0.6011, Val Accuracy: 0.7000, Val
AUROC: 0.7004
Epoch 8/50, Train Loss: 0.6227, Val Loss: 0.6034, Val Accuracy: 0.7006, Val
AUROC: 0.7009
Epoch 9/50, Train Loss: 0.6206, Val Loss: 0.5982, Val Accuracy: 0.6975, Val
AUROC: 0.6984
Epoch 10/50, Train Loss: 0.6215, Val Loss: 0.6071, Val Accuracy: 0.6858, Val
AUROC: 0.6879
Epoch 11/50, Train Loss: 0.6201, Val Loss: 0.6013, Val Accuracy: 0.6951, Val

```


AUROC: 0.6939
 Epoch 12/50, Train Loss: 0.6209, Val Loss: 0.5998, Val Accuracy: 0.7068, Val AUROC: 0.7074
 Epoch 13/50, Train Loss: 0.6165, Val Loss: 0.5954, Val Accuracy: 0.6951, Val AUROC: 0.6935
 Epoch 14/50, Train Loss: 0.6153, Val Loss: 0.5972, Val Accuracy: 0.7111, Val AUROC: 0.7106
 Epoch 15/50, Train Loss: 0.6154, Val Loss: 0.5952, Val Accuracy: 0.7074, Val AUROC: 0.7083
 Epoch 16/50, Train Loss: 0.6166, Val Loss: 0.6001, Val Accuracy: 0.6877, Val AUROC: 0.6850
 Epoch 17/50, Train Loss: 0.6135, Val Loss: 0.5986, Val Accuracy: 0.7000, Val AUROC: 0.6983
 Epoch 18/50, Train Loss: 0.6626, Val Loss: 0.6028, Val Accuracy: 0.6846, Val AUROC: 0.6817
 Epoch 19/50, Train Loss: 0.6141, Val Loss: 0.6030, Val Accuracy: 0.6809, Val AUROC: 0.6775
 Epoch 20/50, Train Loss: 0.6128, Val Loss: 0.5911, Val Accuracy: 0.7049, Val AUROC: 0.7044
 Epoch 21/50, Train Loss: 0.6094, Val Loss: 0.5909, Val Accuracy: 0.7031, Val AUROC: 0.7019
 Epoch 22/50, Train Loss: 0.6113, Val Loss: 0.5877, Val Accuracy: 0.7086, Val AUROC: 0.7085
 Epoch 23/50, Train Loss: 0.6072, Val Loss: 0.5909, Val Accuracy: 0.7105, Val AUROC: 0.7111
 Epoch 24/50, Train Loss: 0.6113, Val Loss: 0.5970, Val Accuracy: 0.7062, Val AUROC: 0.7074
 Epoch 25/50, Train Loss: 0.6062, Val Loss: 0.5904, Val Accuracy: 0.7117, Val AUROC: 0.7100
 Epoch 26/50, Train Loss: 0.6079, Val Loss: 0.5891, Val Accuracy: 0.7074, Val AUROC: 0.7066
 Epoch 27/50, Train Loss: 0.6077, Val Loss: 0.5950, Val Accuracy: 0.6926, Val AUROC: 0.6897
 Epoch 28/50, Train Loss: 0.6086, Val Loss: 0.5856, Val Accuracy: 0.7056, Val AUROC: 0.7051
 Epoch 29/50, Train Loss: 0.6091, Val Loss: 0.5968, Val Accuracy: 0.7012, Val AUROC: 0.7030
 Epoch 30/50, Train Loss: 0.6059, Val Loss: 0.5875, Val Accuracy: 0.7123, Val AUROC: 0.7118
 Epoch 31/50, Train Loss: 0.6052, Val Loss: 0.5874, Val Accuracy: 0.7179, Val AUROC: 0.7163
 Epoch 32/50, Train Loss: 0.6062, Val Loss: 0.5925, Val Accuracy: 0.7185, Val AUROC: 0.7179
 Epoch 33/50, Train Loss: 0.6045, Val Loss: 0.5839, Val Accuracy: 0.7241, Val AUROC: 0.7231
 Epoch 34/50, Train Loss: 0.6027, Val Loss: 0.5878, Val Accuracy: 0.7167, Val AUROC: 0.7169
 Epoch 35/50, Train Loss: 0.6035, Val Loss: 0.5849, Val Accuracy: 0.7099, Val

AUROC: 0.7099
 Epoch 36/50, Train Loss: 0.6043, Val Loss: 0.5860, Val Accuracy: 0.7142, Val AUROC: 0.7140
 Epoch 37/50, Train Loss: 0.6024, Val Loss: 0.5877, Val Accuracy: 0.7099, Val AUROC: 0.7080
 Epoch 38/50, Train Loss: 0.6015, Val Loss: 0.5849, Val Accuracy: 0.7136, Val AUROC: 0.7139
 Epoch 39/50, Train Loss: 1.1135, Val Loss: 0.5830, Val Accuracy: 0.7160, Val AUROC: 0.7154
 Epoch 40/50, Train Loss: 0.6008, Val Loss: 0.5836, Val Accuracy: 0.7160, Val AUROC: 0.7154
 Epoch 41/50, Train Loss: 0.6017, Val Loss: 0.5844, Val Accuracy: 0.7142, Val AUROC: 0.7139
 Epoch 42/50, Train Loss: 0.6037, Val Loss: 0.5848, Val Accuracy: 0.7142, Val AUROC: 0.7140
 Epoch 43/50, Train Loss: 0.5993, Val Loss: 0.5932, Val Accuracy: 0.6988, Val AUROC: 0.6960
 Epoch 44/50, Train Loss: 0.6003, Val Loss: 0.5827, Val Accuracy: 0.7173, Val AUROC: 0.7173
 Epoch 45/50, Train Loss: 0.6013, Val Loss: 0.5867, Val Accuracy: 0.7117, Val AUROC: 0.7101
 Epoch 46/50, Train Loss: 0.5965, Val Loss: 0.5839, Val Accuracy: 0.7136, Val AUROC: 0.7123
 Epoch 47/50, Train Loss: 0.5969, Val Loss: 0.5791, Val Accuracy: 0.7130, Val AUROC: 0.7118
 Epoch 48/50, Train Loss: 0.6104, Val Loss: 0.5851, Val Accuracy: 0.7185, Val AUROC: 0.7184
 Epoch 49/50, Train Loss: 0.5995, Val Loss: 0.5885, Val Accuracy: 0.7080, Val AUROC: 0.7059
 Epoch 50/50, Train Loss: 0.5990, Val Loss: 0.5826, Val Accuracy: 0.7173, Val AUROC: 0.7162

```

[17]: # Testing
      test_loss, test_accuracy, test_auroc = evaluate(test_loader)
      print(f'Test Loss: {test_loss:.4f}, Test Accuracy: {100*test_accuracy:.4f}%,  

            ↳Test AUROC: {test_auroc:.4f}')
  
```

Test Loss: 0.5759, Test Accuracy: 72.4444%, Test AUROC: 0.7235

[14]: