```
1 !pip install deeprobust
```

```
Collecting deeprobust
  Downloading https://files.pythonhosted.org/packages/28/d4/729e089109e33e917e973f87f7169e5f0
     |████████████████████████████████| 184kB 7.9MB/s
Collecting gensim<4.0,>=3.8
  Downloading https://files.pythonhosted.org/packages/5c/4e/afe2315e08a38967f8a3036bbe7e38b42
     |████████████████████████████████| 24.2MB 106kB/s
Requirement already satisfied: scipy>=1.3.1 in /usr/local/lib/python3.7/dist-packages (from d
Collecting texttable>=1.6.2
  Downloading https://files.pythonhosted.org/packages/06/f5/46201c428aebe0eecfa83df66bf3e6caa
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: numpy>=1.17.1 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: scikit-learn>=0.22.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tqdm>=3.0 in /usr/local/lib/python3.7/dist-packages (from deep
Requirement already satisfied: Pillow>=7.0.0 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: torch>=1.2.0 in /usr/local/lib/python3.7/dist-packages (from d
Requirement already satisfied: torchvision>=0.4.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: matplotlib>=3.1.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: scikit-image>=0.0 in /usr/local/lib/python3.7/dist-packages (f
Collecting tensorboardX>=2.0
  Downloading https://files.pythonhosted.org/packages/07/84/46421bd3e0e89a92682b1a38b40efc22d
     |████████████████████████████████| 122kB 48.8MB/s
Requirement already satisfied: numba>=0.48.0 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: six>=1.5.0 in /usr/local/lib/python3.7/dist-packages (from gen
Requirement already satisfied: decorator<5,>=4.3 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from s
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from m
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/pyt
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: PyWavelets>=0.4.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: protobuf>=3.8.0 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from num
Installing collected packages: gensim, texttable, tensorboardX, deeprobust
  Found existing installation: gensim 3.6.0
    Uninstalling gensim-3.6.0:
      Successfully uninstalled gensim-3.6.0
Successfully installed deeprobust-0.2.2 gensim-3.8.3 tensorboardX-2.2 texttable-1.6.3
```

Global Attack for Node classification

```
1 import numpy as np
2 from deeprobust.graph.data import Dataset
3 from deeprobust.graph.defense import GCN
4 from deeprobust.graph.global_attack import Metattack
5 data = Dataset(root='/tmp/', name='cora')
6 adj, features, labels = data.adj, data.features, data.labels
7 idx_train, idx_val, idx_test = data.idx_train, data.idx_val, data.idx_test
8 idx_unlabeled = np.union1d(idx_val, idx_test)
```

```
idx_unlabeled = np.union1d(idx_val, idx_test)
# Setup Surrogate model
surrogate = GCN(nfeat=features.shape[1], nclass=labels.max().item()+1,
                nhid=16, dropout=0, with_relu=False, with_bias=False, device='cpu').to('cpu')
surrogate.fit(features, adj, labels, idx_train, idx_val, patience=30)
# Setup Attack Model
model = Metattack(surrogate, nnodes=adj.shape[0], feature_shape=features.shape,
        attack_structure=True, attack_features=False, device='cpu', lambda_=0).to('cpu')
# Attack
model.attack(features, adj, labels, idx_train, idx_unlabeled, n_perturbations=10, ll_constraint=
modified_adj = model.modified_adj # modified_adj is a torch.tensor
```

```
No module named 'torch_geometric'
Loading cora dataset...
Selecting 1 largest connected components
/usr/local/lib/python3.7/dist-packages/deeprobust/graph/defense/__init__.py:16: UserWarning:
  "geometric. See details in https://pytorch-geom" +
Perturbing graph:   0%|          | 0/10 [00:00<?, ?it/s]GCN loss on unlabled data: 0.5084404
GCN acc on unlabled data: 0.8301296379079124
attack loss: 0.3048456311225891
/usr/local/lib/python3.7/dist-packages/torch/_tensor.py:575: UserWarning: floor_divide is dep
To keep the current behavior, use torch.div(a, b, rounding_mode='trunc'), or for actual floor
  return torch.floor_divide(self, other)
Perturbing graph:  10%|▉         | 1/10 [00:15<02:22, 15.87s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8421993741618239
attack loss: 0.3358333706855774
Perturbing graph:  20%|██        | 2/10 [00:31<02:06, 15.79s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.843540455967814
attack loss: 0.31400004029273987
Perturbing graph:  30%|███       | 3/10 [00:47<01:50, 15.77s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8301296379079124
attack loss: 0.32374322414398193
Perturbing graph:  40%|████      | 4/10 [01:02<01:34, 15.77s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8408582923558338
attack loss: 0.3399897515773773
Perturbing graph:  50%|█████     | 5/10 [01:18<01:18, 15.73s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.843540455967814
attack loss: 0.3399098515510559
Perturbing graph:  60%|██████    | 6/10 [01:34<01:02, 15.67s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8287885561019223
attack loss: 0.33262398838996887
Perturbing graph:  70%|███████   | 7/10 [01:49<00:47, 15.72s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8390701832811801
attack loss: 0.34894293546676636
Perturbing graph:  80%|████████  | 8/10 [02:05<00:31, 15.67s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8314707197139025
attack loss: 0.34770819544792175
Perturbing graph:  90%|█████████ | 9/10 [02:21<00:15, 15.68s/it]GCN loss on unlabled data: 0.
GCN acc on unlabled data: 0.8265534197586053
attack loss: 0.3598272204399109
Perturbing graph: 100%|██████████| 10/10 [02:36<00:00, 15.70s/it]
```

Targeted Attack for Node Classification

```
from deeprobust.graph.data import Dataset
```

```
from deeprobust.graph.defense import GCN
from deeprobust.graph.targeted_attack import Nettack
data = Dataset(root='/tmp/', name='cora')
adj, features, labels = data.adj, data.features, data.labels
idx_train, idx_val, idx_test = data.idx_train, data.idx_val, data.idx_test
# Setup Surrogate model
surrogate = GCN(nfeat=features.shape[1], nclass=labels.max().item()+1,
                nhid=16, dropout=0, with_relu=False, with_bias=False, device='cpu').to('cpu')
surrogate.fit(features, adj, labels, idx_train, idx_val, patience=30)
# Setup Attack Model
target_node = 0
model = Nettack(surrogate, nnodes=adj.shape[0], attack_structure=True, attack_features=True, dev
# Attack
model.attack(features, adj, labels, target_node, n_perturbations=5)
modified_adj = model.modified_adj # scipy sparse matrix
modified_features = model.modified_features # scipy sparse matrix
```

```
Loading cora dataset...
Selecting 1 largest connected components
##### Starting attack #####
##### Attack node with ID 0 using structure and feature perturbations #####
##### Attacking the node directly #####
##### Performing 5 perturbations #####
##### ...1/5 perturbations ... #####
/usr/local/lib/python3.7/dist-packages/numba/core/ir_utils.py:2031: NumbaPendingDeprecationWa
Encountered the use of a type that is scheduled for deprecation: type 'reflected set' found f

For more information visit https://numba.pydata.org/numba-doc/latest/reference/deprecation.ht

File "../usr/local/lib/python3.7/dist-packages/deeprobust/graph/targeted_attack/nettack.py",
@jit(nopython=True)
def compute_new_a_hat_uv(edge_ixs, node_nb_ixs, edges_set, twohop_ixs, values_before, degs, p
^

  warnings.warn(NumbaPendingDeprecationWarning(msg, loc=loc))
Edge perturbation: [   0 1664]
##### ...2/5 perturbations ... #####
Edge perturbation: [   0 1301]
##### ...3/5 perturbations ... #####
Edge perturbation: [   0 1084]
##### ...4/5 perturbations ... #####
Edge perturbation: [   0 1193]
##### ...5/5 perturbations ... #####
Edge perturbation: [   0 1046]
```

1. Load pre-attacked graph data

```
from deeprobust.graph.data import Dataset, PrePtbDataset
# load the prognn splits by using setting='prognn'
# because the attacked graphs are generated under prognn splits
data = Dataset(root='/tmp/', name='cora', setting='prognn')

adj, features, labels = data.adj, data.features, data.labels
idx_train, idx_val, idx_test = data.idx_train, data.idx_val, data.idx_test
```

```
# Load meta attacked data
perturbed_data = PrePtbDataset(root='/tmp/',
                    name='cora',
                    attack_method='meta',
                    ptb_rate=0.05)
perturbed_adj = perturbed_data.adj
```

```
Loading cora dataset...
Selecting 1 largest connected components
Dowloading from https://raw.githubusercontent.com/ChandlerBang/Pro-GNN/master/splits/cora_pro
Dowloading from https://raw.githubusercontent.com/ChandlerBang/Pro-GNN/master/meta/cora_meta_
Loading cora dataset perturbed by 0.05 meta...
/usr/local/lib/python3.7/dist-packages/deeprobust/graph/data/attacked_data.py:135: UserWarnin
  warnings.warn("The pre-attacked graph is perturbed under the data splits provided by ProGNN
```

2. Train victim model on clean/poisoned graph

```
from deeprobust.graph.defense import GCN
gcn = GCN(nfeat=features.shape[1],
    nhid=16,
    nclass=labels.max().item() + 1,
    dropout=0.5, device='cpu')
gcn = gcn.to('cpu')
gcn.fit(features, adj, labels, idx_train, idx_val) # train on clean graph with earlystopping
gcn.test(idx_test)

gcn.fit(features, perturbed_adj, labels, idx_train, idx_val) # train on poisoned graph
gcn.test(idx_test)
```

```
Test set results: loss= 0.5641 accuracy= 0.8290
Test set results: loss= 0.7253 accuracy= 0.7762
0.7761569416498993
```

3. Train defense models

```
from deeprobust.graph.defense import GCNJaccard
model = GCNJaccard(nfeat=features.shape[1],
          nhid=16,
          nclass=labels.max().item() + 1,
          dropout=0.5, device='cpu').to('cpu')
model.fit(features, perturbed_adj, labels, idx_train, idx_val, threshold=0.03)
model.test(idx_test)
```

```
removed 1191 edges in the original graph
=== training gcn model ===
Epoch 0, training loss: 1.907806634902954
Epoch 10, training loss: 1.1862224340438843
Epoch 20, training loss: 0.5860435366630554
Epoch 30, training loss: 0.3268839716911316
Epoch 40, training loss: 0.18484485149383545
```

```
Epoch 50, training loss: 0.14373797178268433
Epoch 60, training loss: 0.13396364450454712
Epoch 70, training loss: 0.10623839497566223
Epoch 80, training loss: 0.08720427751541138
Epoch 90, training loss: 0.06903304904699326
Epoch 100, training loss: 0.09412094205617905
Epoch 110, training loss: 0.0924135223031044
Epoch 120, training loss: 0.08030423521995544
Epoch 130, training loss: 0.07552876323461533
Epoch 140, training loss: 0.061487339437007904
Epoch 150, training loss: 0.06574063003063202
Epoch 160, training loss: 0.06646846979856491
Epoch 170, training loss: 0.06772365421056747
Epoch 180, training loss: 0.07207415997982025
Epoch 190, training loss: 0.06294155865907669
=== picking the best model according to the performance on validation ===
Test set results: loss= 0.6433 accuracy= 0.8003
0.8003018108651911
```

Node embedding attack

```
from deeprobust.graph.data import Dataset
from deeprobust.graph.global_attack import NodeEmbeddingAttack
data = Dataset(root='/tmp/', name='cora_ml', seed=15)
adj, features, labels = data.adj, data.features, data.labels
model = NodeEmbeddingAttack()
model.attack(adj, attack_type="remove")
modified_adj = model.modified_adj
model.attack(adj, attack_type="remove", min_span_tree=True)
modified_adj = model.modified_adj
model.attack(adj, attack_type="add", n_candidates=10000)
modified_adj = model.modified_adj
model.attack(adj, attack_type="add_by_remove", n_candidates=10000)
modified_adj = model.modified_adj
```

```
Loading cora_ml dataset...
Downloading from https://raw.githubusercontent.com/danielzuegner/gnn-meta-attack/master/data/
Done!
Selecting 1 largest connected components
```

Node embedding victim models

```
from deeprobust.graph.data import Dataset
from deeprobust.graph.defense import DeepWalk
from deeprobust.graph.global_attack import NodeEmbeddingAttack
import numpy as np

dataset_str = 'cora_ml'
data = Dataset(root='/tmp/', name=dataset_str, seed=15)
adj, features, labels = data.adj, data.features, data.labels
idx_train, idx_val, idx_test = data.idx_train, data.idx_val, data.idx_test
```

```
10
11 print("Test DeepWalk on clean graph")
12 model = DeepWalk(type="skipgram")
13 model.fit(adj)
14 print(model.embedding)
```

```
Loading cora_ml dataset...
Selecting 1 largest connected components
Test DeepWalk on clean graph
/usr/local/lib/python3.7/dist-packages/numba/core/typed_passes.py:314: NumbaPerformanceWarnin
The keyword argument 'parallel=True' was specified but no transformation for parallel executi

To find out why, try turning on parallel diagnostics, see https://numba.pydata.org/numba-doc/

File "../usr/local/lib/python3.7/dist-packages/deeprobust/graph/defense/node_embedding.py", l
@numba.jit(nopython=True, parallel=True)
def _random_walk(indptr, indices, walk_length, walks_per_node, seed):
^

  state.func_ir.loc))
[[-0.08024385 -0.8360118  -0.47371867 ... -0.00143743  0.46734792
  -0.30357292]
 [ 0.66303325 -0.21016534 -0.6175517  ... -0.06177273  0.26229024
   0.20200323]
 [ 0.5145694  -0.027938   -0.4842296  ...  0.063301    0.16260335
   0.5477671 ]
 ...
 [-0.5318782  -0.07631438 -0.51916534 ...  0.11400343  0.44082317
   0.22816503]
 [ 0.05522015 -0.3257659  -0.20640592 ...  0.63721156 -0.23127624
  -0.5375081 ]
 [-0.35134086  0.264934   -1.0428435  ...  0.5431733  -0.2560392
  -0.10976694]]
```

```
1 print("Test DeepWalk on node classification...")
2 # model.evaluate_node_classification(labels, idx_train, idx_test, lr_params={"max_iter": 1000})
3 model.evaluate_node_classification(labels, idx_train, idx_test)
4 print("Test DeepWalk on link prediciton...")
5 model.evaluate_link_prediction(adj, np.array(adj.nonzero()).T)
```

```
Test DeepWalk on node classification...
Micro F1: 0.8207295373665481
Macro F1: 0.7904730783023871
Test DeepWalk on link prediciton...
ROC error
AUC: 0.0
AP: 1.0
(array([0.9883237 , 0.96201897, 0.98966515, ..., 0.8913061 , 0.8152671 ,
        0.92863464], dtype=float32), 0.0, 1.0)
```

```
1 # set up the attack model
2 attacker = NodeEmbeddingAttack()
3 attacker.attack(adj, attack_type="remove", n_perturbations=1000)
4 modified_adj = attacker.modified_adj
5 print("Test DeepWalk on attacked graph")
6 model.fit(modified_adj)
7 model.evaluate_node_classification(labels, idx_train, idx_test)
```

```
# model.evaluate_node_classification(labels, idx_train, idx_test)
```

```
Test DeepWalk on attacked graph
Micro F1: 0.7384341637010676
Macro F1: 0.6767085291758695
(array([4, 0, 2, ..., 2, 2, 4]), 0.7384341637010676, 0.6767085291758695)
```

0초 오전 1:02에 완료됨