Graph Neural Networks and Graph Convolutional Networks

Carlos Puigserver Germán Llorente

November 19, 2023

Implementación

En esta sección, se presenta la implementación de Graph Neural Networks (GNN) y redes convolucionales gráficas (GCN) utilizando PyTorch Geometric. Se ha utilizado un entorno específico y un conjunto de datos hipotético de documentos para la tarea de clasificación.

Entorno y Configuración Inicial

Entorno de Desarrollo

La implementación se llevó a cabo en un entorno de desarrollo basado en Python utilizando Jupyter Notebook. Se utilizó Python 3.8 y las siguientes bibliotecas principales:

```
$ pip install torch
$ pip install torch-geometric
```

Configuración Inicial

Se creó un script de Python en un entorno virtual para garantizar la independencia del entorno. A continuación, se presenta un resumen de la configuración inicial:

```
# Activar el entorno virtual
$ source venv/bin/activate

# Instalar bibliotecas necesarias
(venv) $ pip install torch
(venv) $ pip install torch-geometric
```

Experimentación con Conjunto de Datos de Documentos

Se utilizó un conjunto de datos hipotético que contiene documentos representados como grafos. Cada nodo del grafo representa una palabra en el documento, y los enlaces entre nodos indican relaciones semánticas entre palabras. El objetivo es clasificar los documentos en categorías específicas.

Carga y Exploración del Conjunto de Datos

El conjunto de datos se cargó utilizando PyTorch Geometric y se exploró para comprender su estructura:

```
from torch_geometric.datasets import DocumentsDataset

# Cargar el conjunto de datos de documentos
document_dataset = DocumentsDataset(
    root='path/to/document_dataset', download=True
)

# Explorar el conjunto de datos
print(f'N mero de documentos en el conjunto de datos: {len(
    document_dataset)}')

# Visualizar un ejemplo de grafo representando un documento
document_graph = document_dataset[0]
print(document_graph)
```

Definición del Modelo GNN y GCN

Se implementó un modelo GNN con una capa GCN para la clasificación de documentos. La arquitectura del modelo se diseñó de acuerdo con las características del conjunto de datos.

```
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch_geometric.nn import GCNConv

class DocumentGNN(nn.Module):
    def __init__(self, in_channels, out_channels, num_classes):
        super(DocumentGNN, self).__init__()
        self.conv1 = GCNConv(in_channels, out_channels)
        self.conv2 = GCNConv(out_channels, num_classes)

    def forward(self, data):
        x, edge_index = data.x, data.edge_index

        x = F.relu(self.conv1(x, edge_index))
        x = F.dropout(x, training=self.training)
        x = self.conv2(x, edge_index)

        return F.log_softmax(x, dim=1)
```

Entrenamiento del Modelo

El modelo se entrenó utilizando un procedimiento de entrenamiento estándar. Se definieron funciones de pérdida y se utilizó el optimizador Adam.

```
from torch_geometric.data import DataLoader
import torch.optim as optim

# Par metros de entrenamiento
lr = 0.01
epochs = 50

# Configuraci n del DataLoader
train_loader = DataLoader(document_dataset, batch_size=32, shuffle=True)

# Inicializaci n del modelo y del optimizador
model = DocumentGNN(in_channels=64, out_channels=128, num_classes=10)
optimizer = optim.Adam(model.parameters(), lr=lr)

# Funci n de p rdida
criterion = nn.NLLLoss()

# Ciclo de entrenamiento
for epoch in range(epochs):
    model.train()
    for data in train_loader:
        optimizer.zero_grad()
        output = model(data)
        loss = criterion(output, data.y)
        loss.backward()
        optimizer.step()

    # Imprimir informaci n de entrenamiento
    print(f' poca  {epoch + 1}/{epochs}, P rdida: {loss.item()}')
```

Este ejemplo hipotético proporciona una visión general de la implementación de GNN y GCN para la clasificación de documentos utilizando PyTorch Geometric. Los detalles específicos, como la arquitectura del grafo, la estructura del conjunto de datos y los parámetros del modelo, deben ajustarse según los requisitos de la tarea y el conjunto de datos específicos.