Homework Assignment 2: GNNs, Transformers, RAG, and RoBERTa Fine-Tuning

Due Date: 2025/03/15

Overview

In this assignment, you will:

1. Modify and implement Graph Neural Networks (GNNs) for molecular property prediction.
2. Analyze Transformer self-attention mechanisms.
3. Build a Retrieval-Augmented Generation (RAG) pipeline and experiment with different retrieval models.
4. Fine-tune RoBERTa for a classification task and compare results across different hyperparameters.

Part A: Conceptual Questions

Provide **detailed written answers** to the following questions:

1. **Graph Neural Networks (GNNs)**
    - Describe how **different message-passing strategies** (e.g., mean aggregation, sum aggregation, attention-based) affect model performance.
    - What is the role of **edge features** in GNNs? How would you modify a basic GNN to incorporate them?
    - Given a **highly connected molecular graph**, what challenges arise in GNN training, and how would you address them?
2. **Transformers and Self-Attention**
    - Explain how **multi-head self-attention** improves Transformer performance. Why not just use a single attention head?
    - If **positional encodings were removed** from a Transformer, what impact would this have on performance?
    - How does **changing the number of attention heads** affect the computational cost and model performance?

3. **Retrieval-Augmented Generation (RAG)**
    - How does **retrieval quality** impact the final generation in RAG-based models?
    - Given a **low-resource dataset**, how would you **fine-tune a retriever** in a RAG pipeline?
    - What are some **failure cases** where a RAG model might underperform compared to a standard Transformer?
4. **Fine-Tuning RoBERTa**
    - Explain the trade-off between **batch size, learning rate, and training stability** when fine-tuning large language models.
    - What are **catastrophic forgetting** and **overfitting**, and how can you **detect and mitigate** them in fine-tuning?
    - Given an **imbalanced dataset**, what strategies would you use to ensure **fair evaluation and training**?

Part B: Implementation Tasks

Task 1: Graph Neural Networks (GNN) for Molecular Property Prediction

Objective: Modify and train a **Graph Neural Network (GNN)** to predict **blood-brain barrier permeability (BBBP)** of molecules.

Steps:

1. Load the **BBBP dataset** from **MoleculeNet**.
2. Construct a **Graph Neural Network (GNN)** for classification.
3. Modify the GNN to:
    - **Use Graph Attention Networks (GAT)** instead of standard message passing.
    - **Incorporate edge features** in the aggregation step.
4. Split the dataset:
    - **80% for training**
    - **10% for validation**

    - **10% for testing**
5. Train the model using **cross-entropy loss** and **Adam optimizer**.
6. Evaluate performance using **AUROC (Area Under the Receiver Operating Characteristic Curve)**.

Deliverables:

- **Code** for defining and training the GNN.
- **Plots** of loss and AUROC over epochs.

Task2: Understanding Attention Mechanisms in BERT

Objective: Explore how self-attention works in Transformer-based models and analyze the role of attention heads and positional encodings.

Steps:

1. **Load a pre-trained BERT model** from Hugging Face.
2. **Extract attention weights** from a sample sentence.
3. **Visualize attention weights** using heatmaps to see how different words attend to each other.
4. **Experiment with:**
    - Disabling **a few attention heads** and observing how attention distributions change.
    - Removing **positional encodings** and checking how it affects the model's output.

Deliverables:

- **Heatmaps** showing attention patterns for different heads and layers.
- **Comparison** of model behavior before and after disabling attention heads.
- **Brief explanation** of findings (1-2 paragraphs).

Task 3: Retrieval-Augmented Generation (RAG) Pipeline

Objective: Implement **RAG-based generation** for question-answering.

Steps:

1. **Use research_assistant.ipynb as a reference** to implement basic document retrieval and RAG-based generation.
2. **Build a simple document retriever using ChromaDB**, indexing a small collection of scientific papers.
3. **Use a pre-trained Transformer model** (e.g., bert-base-uncased or distilbert-base-uncased) to generate answers:
    - Without retrieval (standard Transformer).
    - With retrieval (RAG-based Transformer).
4. **Compare two retrieval methods**:
    - **BM25** (text-based search).
    - **Dense embeddings** (sentence-transformers/all-MiniLM-L6-v2).
5. **Modify one retrieval hyperparameter** (number of retrieved documents k = 5 vs. 10) and observe the difference.
6. **Evaluate the answers** based on:
    - **Readability and relevance** (manually rate outputs).
    - **Token length** (compare generated response lengths).
    - **Retrieval effectiveness** (whether the retrieved documents contain the answer).

Deliverables:

- **Code for document retrieval and response generation** using a pre-trained Transformer model.
- **A short comparison** (in markdown or a text file) on how retrieval affects answer quality.
- **Observations** on whether increasing the number of retrieved documents improves or degrades performance.

Task 4: Fine-Tuning RoBERTa for Multi-Label Classification

Objective: Fine-tune **RoBERTa** to classify **toxic online comments**.

Steps:

1. Use the **Unhealthy Comment Corpus** dataset (already included in Fine_tuning_RoBERTa_Unhealthy_Comment_Corpus.ipynb).
2. Modify training settings:
    - **Experiment with different batch sizes** (16, 32, 64).
    - **Test different learning rates** (1e-5, 3e-5, 5e-5).
    - **Implement gradient clipping** to avoid exploding gradients.
3. Evaluate the model on **Precision, Recall, F1-score, and AUROC**.
4. Plot:
    - **Training loss over time**
    - **Validation accuracy over time**
    - **Precision-Recall curves for different classes**.

Deliverables:

- **Fine-tuned RoBERTa model** with evaluation metrics.
- **Plots of loss, accuracy, and PR curves**.
- **Comparison of different hyperparameter settings**.

Submission Guidelines

Submit a zip file named [Lastname]_[Firstname]_HW_2.zip containing:

1. **Code Notebooks**:
    - Task1_GNN.ipynb
    - Task2_Transformers.ipynb
    - Task3_RAG.ipynb

- Task4_FineTuning_RoBERTa.ipynb

2. **Short Report (PDF)**
    - Written answers for **Part A: Conceptual Questions**.
    - Summary of key **results and insights** from **Part B**.

Grading Rubric

Section	Points
Conceptual Questions	20%
GNN Implementation & Performance Evaluation	20%
Transformer Attention Analysis & Visualizations	20%
RAG-Based Generation & Document Retrieval	20%
Fine-Tuning RoBERTa & Hyperparameter Tuning	20%
Total	**100%**