NLP Assignment 3 Report

Implementing the BERT from scratch for fake news classification

Name: Diganta Mandal
Roll: 22CS30062
Course: Natural Language Processing

Introduction

This report presents a comprehensive analysis of the parameters and architecture of the BERT model implemented from scratch. The **Fakeddit dataset** was utilized for the task of **fake news classification**. The report provides a detailed justification for adopting the **post–comment–reply structure**, supported by descriptive statistics of the dataset. Furthermore, it outlines the **hyperparameters used during fine-tuning** of the BERT model and summarizes the **evaluation results obtained on the test set**, including key performance metrics.

1. Implement BERT class from scratch

Sentence used : ”My name is Diganta Mandal. I am enjoying the NLP Course.”

1.1 Shapes of Embeddings and Intermediate Outputs

Input Embeddings		
Component	**Description**	**Shape**
Token Embeddings	Each token is mapped to a 768-dimensional vector using a learned embedding table.	[1,17,768]
Positional Embeddings	Encodes the position of each token in the input sequence.	[1,17,768]
Combined Embeddings	Sum of token and positional embeddings (optionally with token type embeddings).	[1,17,768]

Transformer Encoder Layer		
Component	**Description**	**Shape**
Multi Head Attention	Contextual representation after self-attention.	[1,17,768]
Feed Forward Layer	Output after two-layer feed-forward transformation.	[1,17,768]
Transformer Encoder Layer Output	Final output after residual connections and normalization.	[1,17,768]

Classification Head		
Component	**Description**	**Shape**
Logits (Pre-Softmax)	Linear projection of the [CLS] embedding to 2 output classes.	[1, 2]
Output Probabilities	Softmax-normalized probabilities across two classes.	[0.4449, 0.5551]

1.2 Shapes of Parameters

Embedding Layer	
Parameter	**Shape**
token_embeddings.weight	[30522, 768]
position_embeddings.weight	[512, 768]
token_type_embeddings.weight	[2, 768]
embedding_layernorm.weight	[768]
embedding_layernorm.bias	[768]

Transformer Encoder Layer	
Parameter	**Shape**
Attention (Query, Key, Value, Output) weights/biases	Each [768, 768] / [768]
Feed-Forward Network weights/biases	[3072, 768], [3072], [768, 3072], [768]
Layer Normalization weights/biases	Each [768]

Same for Layer 2

Pooler and Classifier	
Parameter	**Shape**
Pooler.weight, Pooler.bias	[768, 768], [768]
Classifier.weight, Classifier.bias	[2, 768], [2]

Total Parameters = 38,605,058

2. Data Preprocessing (Fakeddit)

2.1 Post-Comment-Reply Structure

I tried out 3 approaches:

- **Title + Comment**
  E.g., TITLE: "Some news headline" | COMMENTS: This is fake! [SEP] Source?
- **Title + Comment + Replies (Two-level)**
  E.g., TITLE: "Some news headline" | COMMENTS: [COMMENT] This is fake! [REPLY] Are you sure? [SEP] [COMMENT] Check the link, it's true.
- **Multi-Level(Full Hierarchy)**
  E.g., TITLE: "Some news headline" | COMMENTS:
  [COMMENT] It's fake
  [REPLY 1] How do you know?
  [REPLY 1.1] Because source says so
  [REPLY 1.2] That's wrong
  [REPLY 2] Agreed

Multi-level comment inclusion was found to perform poorly when various comment aggregation strategies were tested for the **bert-base-uncased** model. This is mostly because adding nested reply structures readily exceeds BERT's **512 token** maximum sequence length. The model's performance suffered as a result of significant portions of the input, especially the post title and opening remarks, being frequently truncated.

Subsequent testing showed that similar problems occurred in longer threads even with the Title + Comment + Replies (two-level) setup. On the other hand, the Title + Comment(with no distinction between comment and replies) configuration continuously produced higher accuracy and better generalization on the test set.

This improvement can be attributed to the fact that **top-level comments typically carry the most relevant and direct reactions** to the post content. Including replies often introduces two extra tokens [COMMENT] and [REPLY] which are redundant or off-topic information while pushing the more meaningful top-level comments out of the model's limited context window.

In summary, for **`bert-base-uncased`**, restricting the input to **Title + Top-Level Comments** provides a more concise, information-dense representation that fits within the model's context limit and leads to superior downstream performance.

2.2 Dataset Statistics

a. Number of posts in the original dataset = 1063106
b. Balanced Dataset:
    1. Fake Posts = 96983
    2. Non-Fake Posts = 96983
c. Distribution of posts in train and test set:
    1. Total training posts = 155172
    2. Total testing posts = 38794
d. Statistics from comment enrichment:
    1. Number of posts with atleast one comment = 584098
    2. Entire Dataset statistics
       Fake Posts:
            Mean = 14.88 std = 56.06
       Non-Fake Posts:
            Mean = 3.85 std = 18.07
    3. Balanced Dataset Statistics
       Fake Posts:
            Mean = 14.78 std = 55.77
       Non-Fake Posts:
            Mean = 3.90 std = 18.39

3. Fine-tuning BERT

3.1 Hyperparameters

- Batch Size = 16
- Learning Rate = 2*1e-5
- Epochs = 4
- Optimizer = AdamW
- Maximum Sequence Length = 512
- Longer sentences were truncated, and shorter ones were padded to maintain uniform length

3.2 Evaluation

Test Set Performance	
Metric	**Score**
Accuracy	0.936021
Precision	0.936696
Recall	0.935248
F1-Score	0.935972