# HW2 - Written Report

ap9283@nyu.edu

16 March 2025

# 1 Problem 1

## 1.1 1b. Understanding BERT Inputs

**Input_ids**: These are tokenized representations of the text that you input, converting them into numerical vectors (tokens). If you were to use the `.decode()` method on these tensors, you could reconstruct the original sentence.

**Token_type_ids**: These indicate which tokens belong to which sentence in the list of sentences. Sentence A is assigned to 0, and sentence B is assigned to 1.

**Attention_mask**: This tells the model which tokens to pay attention to. Tokens with a value of 1 are attended to, while those with 0 are ignored. This helps the model ignore padding tokens when computing attention scores.

## 1.2 1c. Understanding BERT Hyperparameters

Smaller BERT models are used when computationally expensive resources are not available. They are also more effective when labels are produced by a larger and more accurate teacher.

**Hidden layers (H)**: The number of neurons per layer. A higher number of hidden layers makes the model more complex. Possible values: {128, 256, 512, 768}.

**Transformer Layers (L)**: The number of layers (depth) of the network. More layers make the model more complex but also more computationally expensive. Layers range from 2 to 12, increasing in steps of 2.

**Batch Size (B)**: The number of samples in a batch that are weighted during each step. Possible sizes: {8, 16, 32, 64, 128}.

**Learning Rate (alpha)**: Controls how much the model updates its weights during each step. The following values were used: {3e-4, 1e-4, 5e-5, 3e-5}. Different learning rates trade off between faster convergence and more stable learning.

# 2 Problem 3: Runing the Experiment

## 2.1 3a. Train Model

| | Validation Accuracy | Learning Rate | Batch Size |
|---|---|---|---|
| Without BitFit | 0.8906 | 0.0001 | 8 |
| With BitFit | 0.7312 | 0.0003 | 8 |

Table 1: Training Results: Validation Accuracy, Learning Rate, and Batch Size

## 2.2 3a. Test Model

**The BitFit paper suggests that fine-tuning only the bias parameters in small BERT models can achieve performance comparable to full fine-tuning. However, our results do not support this claim. We observe that models fine-tuned without BitFit perform significantly better, achieving a 21% relative improvement over the BitFit model.**

| | **Validation Accuracy** | **Learning Rate** | **Batch Size** |
|---|---|---|---|
| Without BitFit | 0.8906 | 0.0001 | 8 |
| With BitFit | 0.7312 | 0.0003 | 8 |

Table 2: Training Results: Validation Accuracy, Learning Rate, and Batch Size

**Thus, we do not observe the expected benefits of BitFit for IMDb classification in this NLP task.**