Assignment #4 – Sentiment Analysis Using finetuned BERT base

Objective:
The main objective of this assignment is to fine-tune the BERT-base-uncased model from Hugging Face for binary sentiment classification on movie reviews. The main aims is adapting a pretrained masked language model to identify positive and negative sentiments by training a small classification head on top of the [cls] token representation.

About the Dataset:
The dataset which is used in this assignment is the IMDB Movie Reviews Sentiment Analysis dataset which is available on Kaggle. The same training, validation and testing data from Assignment was reused i.e., the same 1000 samples were selected for training and validation and 200 samples were used for testing. The 1000 samples were further split into 800 for training and 200 for validation.

Data Preprocessing:
First I have removed all the HTML tags and special symbols such as <s>, </s>, <br>, <br/>, etc. Replaced removed tags with appropriate punctuation marks and to save sentence boundaries and context for negation handling. Punctuation or stop words were remained same, as punctuation helps in contextual understanding in transformer models.
Each review text was tokenized using the BertTokenizerFast from hugging face with a maximum sequence length of 128 tokens. The tokenizer automatically lowercased the input which was later verified through inspection of a few tokenized samples.

After preprocessing and tokenization, the text is then converted into Pytorch tensors for input ISs, attention masks, and token type IDs. These were wrapped into TensorData and DataLoader objects for efficient batching fine-tuning.

Model Architectures:
Model 1 - BERT Base (No Hidden Layer)
This baseline model uses the default BertForSequenceClassification from Hugging Face. In which the classifier head consists of a single linear layer directly mapping the [CLS] embedding (size 768) to two output classes. This setup represents the simplest fine-tuning configuration.

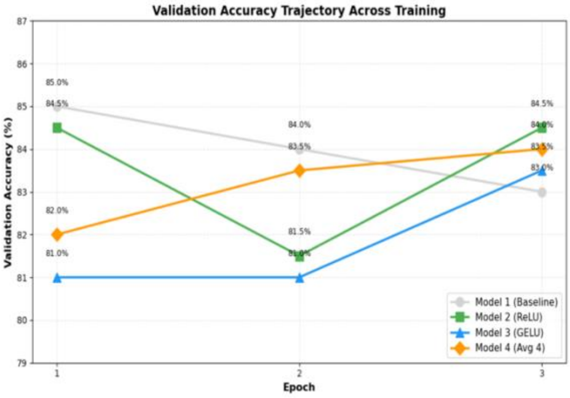
Model 2 - BERT + ReLU Hidden Layer
Before the output layer, a hidden dense layer with 256 units and a ReLU activation function was added to increase the classifier's representational capacity. This makes the model to learn non-linear decision limits within the BERT-generated feature space.

Model 3 - BERT + GELU Hidden Layer
This model has the same model architecture as Model 2, but instead of ReLU here I have used GELU (Gaussian Error Linear Unit). GELU uses smoother activation dynamics, allowing small negative activations to pass through instead of making them zeros. This can improve gradient flow and model generalization.

Model 4 – BERT (Average of Last 4 Layers)

For my fourth model, I used the same setup as Model 1, since it have achieved the best validation accuracy (85%) during training. The one and only change I made was how the [CLS] embedding was extracted from BERT. Instead of taking the [CLS] token from only the final transformer layer, I averaged the hidden states from the last four BERT layers. Which then produced a smoother and rich contextual embedding that captures both low-level and high level semantics.

Training Configuration:



- **Optimizer**: AdamW with learning rate 2e-5
- **Scheduler**: Linear learning rate decay
- **Loss Function**: CrossEntropyLoss
- **Epochs**: 3
- **Batch Size**: 10
- **Max Sequence Length**: 128 tokens

Per-Class Performance Metrics (Test Data):

Model	Class	Precision	Recall	F1-Score	Test Accuracy
Model 1	Negative	0.8317	0.8400	0.8358	83.50%
	Positive	0.8384	0.8300	0.8342	
	Macro Avg	0.8350	0.8350	0.8350	
Model 2(ReLU)	Negative	0.8544	0.8800	0.8670	86.50%
	Positive	0.8763	0.8500	0.8629	
	Macro Avg	0.8653	0.8650	0.8650	
Model3(GELU)	Negative	0.8889	0.8000	0.8421	85.00%
	Positive	0.8182	0.9000	0.8571	
	Macro Avg	0.8535	0.8500	0.8496	
Model 4 (Avg4)	Negative	0.8529	0.8700	0.8614	86.00%
	Positive	0.8673	0.8500	0.8586	
	Macro Avg	0.8601	0.8600	0.8600	

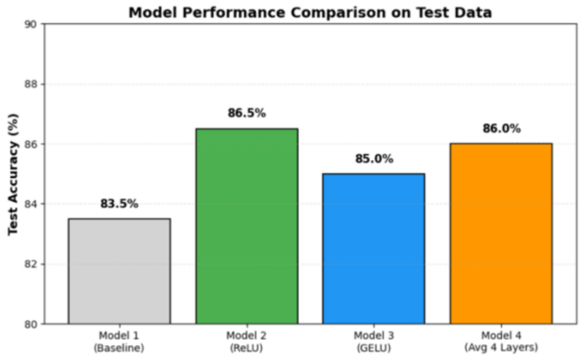
Model 2 have scored the best performance with 86.50% accuracy and the most balanced metrics among all classes. The difference between the precision and recall classes was the smallest for model 2 (0.0041 in F1-Score) among other models, which indicates stable predictions for both sentiments. Model 3 has notable class imbalance with 88.89% precision for negative but only 81.82% for positive which suggesting it was conservative in predicting positive sentiment. In model 4 the averaging approach improved slightly over baseline but couldn’t match Model 2 performance. The 3.12%

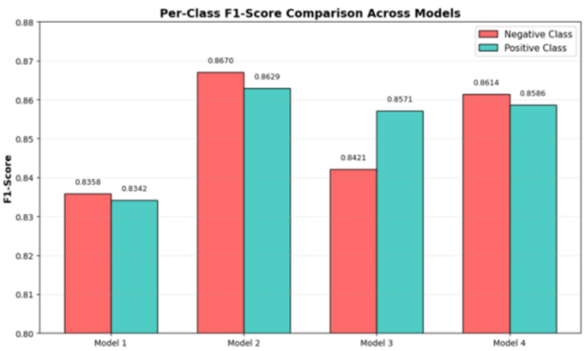
change in negative F1 score (0.8353 → 0.8670) for model 2 can tell us approximately 18% error reduction, which showing us a meaningful practical significance.

Overall Accuracy Comparison:

Model	Val Acc (Ep 1)	Val Acc (Ep 3)	Test Accuracy
Model 1	85.00%	83.00%	83.50%
Model2(ReLU)	84.50%	84.50%	86.50%
Model3(GELU)	81.00%	83.50%	85.00%
Model 4(Avg4)	81.50%	83.50%	86.00%

The validation accuracy trajectories tells us an important training dynamics. Model 1's declining performance (85%→ 83%) indicates overfitting on the small training set, which is being the simplest architecture. Model 2 maintained stable validation accuracy throughout training, which is showing good generalization and optimal capacity for this dataset size. Model 3's improving trajectory (81% → 83.5%) shows us it may benefit from additional training epochs. The difference between validation and test accuracy varies across models: Model 1 has given closet alignment (83% val, 85% test),whereas Mode 2's test performance exceeded its validation score (84.5% val, 86.5% test), which suggesting the validation split may not have been fully representative.





Answers to the Questions:

1) Yes, Model 2 showed statistical improvements. For negative class: precision + 0.0227 (0.8317→0.8544), recall +0.0400 (0.8400→0.8800), F1 +0.0312 (0.8358→0.8670). For positive class: precision +0.0379 (0.8384→0.8763), recall +0.0200 (0.8300→0.8500), F1 +0.0287 (0.8342→0.8629). These shows us approximately 18% error reduction. Model 3 achieved highest negative precision (0.08889) and positive recall (0.9000) but at cost of balance the 0.0707 precision difference between classes tell us us prediction bias. Model 4 has improved over baseline but there is no advantage over simpler Model 2.

2) **When Useful:** Hidden layers add non-linear transformations which are useful when (a) decision boundaries are complex and not linearly separable, (b) Task-specific feature need extraction from general embeddings, (3) dimensionality reduction helps in generalization. **Was It Useful?:** Yes, Both Model 2 (+3.00%) and Model 3 (+1.50%) has

increased over baseline. The 256 unit layer made task specific sentiment feature learning while reducing dimensionality (768→ 256→ 2).

3) **Key Differences**: ReLU: f(x)=max(0,x), hard threshold, zero for negatives, computationally efficient. GELU: f(x)=x·Φ(x), smooth/differentiable, which allows small negative values, generally used in BERT's internal layers, theoretically makes better gradient flow. **Expected**: GELU should perform better due to architectural consistency with BERT and smoother optimization. **Observed**: ReLU won (86.50% vs 85.00%). **Why?** (1) dataset (800 samples) is very small: ReLU's hard threshold provides implicit regularization which prevents overfitting; GELU's smoothness allowed more complex boundaries that didn't generalize. (2) Simple Task: Binary sentiment classification helps from ReLU's straightforward thresholding over GELU's probabilistic weighting.

4) **Comparison**: Best of first 3 (Model 2: 86.50%) vs Model 4 (86.00%) = -0.50 percentage point. Model 2 performed better. **why**? Bert layers capture different linguistics lower: syntax, middle: semantics, upper: task-specific. The final layer already has specialized for sentiment during pre-trained/fine tuning. By using the average of less specialized layers (9-11) diluted these optimized representations. Creating "consensus" from 4 layers made the impact low of task specific adaptations in layer 12, which made to create noise from less tuned layers. **When Averaging Helps:** Multi-task learning, complex task, domain transfer i.e., situations requiring diverse linguistic perspectives. In case of binary sentiment classification, the final specialized layer sufficed.

5) **Performance**: BERT Model 2 achieved 86.50% accuracy vs Naive Bayes' 75.00% (+11.5%) . Best improvement was positive class recall: BERT 0.8500 vs Naive Bayes 0.6100 (+39.3%). BERT showed stable class performance (F1 Score change is 0.0041) while Naive Bayes was biased toward negatives (F1 is 0.0714). The training time for Naive bayes is low compared to BERT. BERT is better for accuracy critical situations despite higher computational cost. The change in accuracy and decreased positive class bias tells about the training time. In resource constrained situation or high throughput situations, Naïve Bayes stays stable at 75% accuracy with instant inference.

Conclusion:

Addition of hidden layers helps in improvement of task specific feature transformation helps pre-trained models. Use of ReLU activation function helps for small data despite theoretical benefits, GELU underperforms ReLU by 1.5%, likely due to stronger regularisation in data limited settings. Model 2 (256 unit ReLU hidden layer, final [CLS] token) achieved 86.50% accuracy best balance of capacity, regularization, and feature specialization. We can conclude that BERT performance on text classification tasks is affected by architecture decisions, specified activation functions and layer averaging.