Sentence Transformers and Multi-Task Learning

-By Sahana Patte Keshava

Objective: The goal of this exercise is to assess your ability to implement, train, and optimize neural network architectures, particularly focusing on transformers and multi-task learning extensions. Please don't spend more than 2 hours on the exercise.

Task 1: Sentence Transformer Implementation
Implement a sentence transformer model using any deep learning framework of your choice. This model should be able to encode input sentences into fixed-length embeddings. Test your implementation with a few sample sentences and showcase the obtained embeddings. Describe any choices you had to make regarding the model architecture *outside of the transformer backbone.*

The objective is to implement a sentence transformer model capable of encoding input sentences into fixed-length embeddings using a pre-trained BERT model as the backbone. The choice of BERT (bert-base-uncased) is chosen due to its robust pre-training on a large corpus of English text, enabling it to generate high-quality embeddings and its strong performance across a wide range of NLP tasks.To allow flexibility in the embedding size, added a linear layer after the BERT model to transform the output embeddings to a fixed size of 256 dimensions.

Task 2: Multi-Task Learning Expansion
Expand the sentence transformer to handle a multi-task learning setting.

1. **Task A: Sentence Classification – Classify sentences into predefined classes (you can make these up).**
2. **Task B: [Choose another relevant NLP task such as Named Entity Recognition, Sentiment Analysis, etc.] (you can make the labels up)**

Describe the changes made to the architecture to support multi-task learning.

Multitask i chose being,(have used negatives.txt and positive.txt from kaggle dataset)

- **TaskA**: Sentence Classification,classifying sentences as Positive or Negative.
- **TaskB**: Sentiment Analysis (regressing a sentiment score).

Changes to the Architecture

To support multi-task learning, modified the architecture of the sentence transformer to include task-specific heads for each of the tasks. The overall architecture will include:

- Shared Transformer Backbone: A pre-trained BERT model to encode the input sentences.
- Shared Linear Layer: A linear layer to transform the BERT embeddings.
- Task-Specific Heads: Added separate heads for each task:
  - Classification Head: A linear layer for sentence classification.
  - Sentiment Analysis Head: A linear layer for sentiment analysis.

Task 3: Training Considerations
Discuss the implications and advantages of each scenario and explain your rationale as to how the model should be trained given the following:

1. **If the entire network should be frozen.**
2. **If only the transformer backbone should be frozen.**
3. **If only one of the task-specific heads (either for Task A or Task B) should be frozen.**

Consider a scenario where transfer learning can be beneficial. Explain how you would approach the transfer learning process, including:

1. **The choice of a pre-trained model.**
2. **The layers you would freeze/unfreeze.**
3. **The rationale behind these choices.**

Entire network is frozen:

Implications:

The model would not learn any new information specific to the tasks.

Advantages:

Saves computational resources since the majority of the network's parameters remain unchanged.Useful when the pre-trained model is expected to perform well without additional task-specific fine-tuning.

Rationale:

Useful when the tasks are very similar to the pre-trained tasks of the model or when computational efficiency is crucial.

Transformer Backbone Should Be Frozen:

Implications:

The model leverages the powerful representations learned by the transformer during pre-training.Task-specific heads can still be fine-tuned to learn from the task-specific data.

Advantages:

Prevents overfitting on small task-specific datasets by keeping the general language understanding capabilities intact.Reduces computational costs compared to training the entire model.But more cost than previous case.

Rationale:

Suitable when the pre-trained transformer has strong generalization capabilities, and we want to adapt it slightly to new tasks.

Only One of the Task-Specific Heads Frozen:

Implications:

The model can adapt to one task while preserving the learned representations for the other task.Allows for selective fine-tuning where only the unfrozen head learns from new data.

Advantages:

Balances the stability of one task's performance with the adaptability of another task.Efficient when one task has abundant data for fine-tuning while the other does not.

Rationale:

This approach is beneficial when there is a significant difference in the data availability of both tasks or importance of the tasks.

I chose the below.The classification head for sentence classification is frozen. the pre-trained model's classification capabilities were adequate, just wanted to fine-tune for Task B(sentiment analysis),allowing it to learn from the task-specific data and adapt to the nuances of sentiment analysis.

Freezing one of the task-specific heads while fine-tuning the other is an effective strategy when dealing with multi-task learning scenarios where the importance or data availability of tasks varies. This approach ensures stability in one task while allowing adaptability and improvement in another.

Transfer Learning Scenario:

We are using transfer learning to adapt a pre-trained BERT model for multi-task learning involving Sentence Classification and Sentiment Analysis.

Choice of a Pre-Trained Model:

Model: BERT (e.g., bert-base-uncased).

Reason: BERT's robust pre-training on a large corpus provides strong general language understanding, making it suitable for various NLP tasks.

Layers to Freeze/Unfreeze:

Freeze: The lower layers of BERT (e.g., the first 8 layers out of 12).These layers capture general language features that are broadly useful and prevent overfitting.

Unfreeze: The top layers of BERT and the task-specific heads.Allows adaptation to the specific characteristics of the tasks.

Unfreezing Task-Specific Heads: Essential for learning task-specific patterns.

Training Strategy:Fine-tune the entire model on a dataset that covers both tasks to adapt the pre-trained model to the new data distribution.Use a smaller learning rate for the transformer backbone and a larger one for the task-specific heads.Fine-tune the model separately on datasets specific to Sentence Classification and Sentiment Analysis.This allows each head to learn more detailed patterns relevant to its specific task.(but not necessary as both are similar tasks)

Rationale:

- Leveraging the pre-trained model's general language understanding while allowing specific layers to adapt to new tasks ensures efficient learning.
- Freezing certain layers saves computational resources and training time.
- By limiting which layers are trained, we mitigate the risk of overfitting on small datasets while still allowing task-specific adaptation.

The approach leverages the strengths of transfer learning by using a pre-trained BERT model, balancing between retaining general language features and adapting to specific tasks, ensuring effective and efficient multi-task learning for Sentence Classification and Sentiment Analysis.

Task 4: Layer-wise Learning Rate Implementation (BONUS)
Implement layer-wise learning rates for the multi-task sentence transformer.
Explain the rationale for the specific learning rates you've set for each layer.
Describe the potential benefits of using layer-wise learning rates for training deep neural networks. Does the multi-task setting play into that?

The rationale for setting different learning rates was to preserve pre-trained knowledge in the BERT layers (low learning rate) while enabling faster learning in the task-specific heads (higher learning rate).The BERT layers are pre-trained and contain rich linguistic information. A lower learning rate helps retain this valuable pre-trained knowledge while allowing for gradual adaptation to the new tasks.The shared linear layer is crucial for transforming the BERT embeddings into a suitable format for the task-specific heads. A slightly higher learning rate than the BERT layers ensures faster adaptation to the specific requirements of the tasks. The classification head is specific to the sentiment classification task. A higher learning rate is suitable here because it helps the model quickly learn task-specific features from scratch.Similarly, the sentiment head requires a higher learning rate to quickly adapt to the regression task of predicting sentiment scores. This approach balances the retention of general language understanding with the quick adaptation to specific tasks, leading to better performance and faster convergence.

Conclusion:

The report outlines a structured approach to implementing sentence transformers and multi-task learning, considering architectural modifications, training strategies, and the benefits of transfer learning. The proposed methods aim to balance computational efficiency, model adaptability, and task performance.