Task 3

Scenario	Implications	Advantages	Rationale
Freezing the Entire Network	All layers are set to their pre-trained states, with no weights being updated during any operation.	Freezing the entire network offers no advantages in terms of training because no learning occurs.	The approach is strictly for inference purposes where the model's existing knowledge and capabilities match the deployment requirements.
Freezing Only the Transformer Backbone	The transformer layers remain static, providing stable, high-quality embeddings. Only the task-specific heads are trained.	Reduces the risk of catastrophic forgetting; faster training and lower computational costs.	Suitable when tasks are somewhat related but distinct from the original training, especially when adaptation to new datasets is needed.
Freezing Only One of the Task-Specific Heads	Freezing one head while training the other allows for focused adaptation of one part of the model while retaining stability in another.	Allows for targeted improvement on tasks where performance is lacking. Reduces the risk of overfitting in the well-performing part of the model.	Ideal for scenarios where tasks vary in difficulty or data availability. Enables customized improvements without disrupting successful behaviors. Adapts to varying data demands without losing task-specific performance, based on learning parameters.

Table 1: Training Considerations for Different Freezing Scenarios

1. Choosing a Pre-trained Model:

- **Model Selection Based on Task Relevance:** I will choose a model that has been pre-trained on data and tasks similar to my target application. For example, I will consider using BERT or RoBERTa, as they are trained on extensive text chunks for tasks like classification.
- **Consideration of Model Size and Complexity:** I will select a model size that suits my performance needs and computational limitations, for example - between BERT-base and BERT-large.

2. Deciding Which Layers to Freeze/Unfreeze:

There are **different cases for different tasks** at hand-

- **Freezing the Entire Transformer:** If the foundational understanding of the language by the model (such as English) is robust, especially for tasks like sentence classification or sentiment analysis where BERT's [CLS] token usage is directly aligned with task requirements, I will initially freeze the entire transformer backbone and only train the task-specific heads. This approach is computationally efficient as it utilizes the strong pre-existing features while focusing computational resources on adapting the model to specific tasks. A good example here can be Customer feedback Classification of voice calls, where we are good with just fine tuning the custom head and freezing the entire transformer.
- **Freezing Lower Layers, Unfreezing Upper Layers:** For an English language based highly specialized classification task like Medical Dictionary, I would freeze a few lower layers as basic English Language features would still be retained. Whereas, the upper layers (including that of the transformer and the head) must be completely unfrozen and fine-tuned on specialized medical texts that include the terminologies.

3. Rationale Behind Choices:

1. **Freezing the Entire Transformer**: Utilizes the transformer’s pre-trained language capabilities for maximal computational efficiency. And concentrates resources on fine-tuning task-specific heads to adapt to the nuances of customer sentiment.

2. **Freezing Lower Layers, Unfreezing Upper Layers** : Maintains essential language features by freezing lower layers, reducing the risk of overfitting. Adapts the model to specialized unseen texts by fine-tuning the upper layers, enhancing accuracy in domain-specific contexts.

4. Consider Data and Task Similarity:

- **For Closely Related Tasks:** If the new task closely mirrors the original training tasks, I will opt to freeze more layers to capitalize fully on the pre-trained knowledge.
- **For Less Related Tasks:** If the new task substantially differs from the original training, I might need to unfreeze more layers to ensure the model can adapt effectively to new challenges.

Summary

In Task 3, I focused on optimizing the application of transfer learning techniques by selectively freezing and unfreezing layers of a pre-trained model depending on the specific needs of the tasks. For tasks like customer feedback analysis, where foundational language understanding is sufficient, I chose to freeze the entire transformer and only fine-tune the task-specific heads, capitalizing on the efficiency of using pre-existing model features while focusing computational resources on fine-tuning for specific sentiment analysis tasks. In contrast, for the complex and specialized requirements of a medical dictionary task, where domain-specific terminologies diverge significantly from general language, I opted to freeze only the lower layers to maintain basic linguistic structures while extensively adapting the upper layers to accurately process and interpret medical jargon. This strategic layer management enables the model to leverage its robust pre-trained base for general understanding while remaining flexible enough to adapt to specialized tasks. The rationale behind these decisions balances computational efficiency with the need for task-specific accuracy, ensuring the model remains both stable and adaptable across varying contexts. This approach illustrates a pragmatic application of transfer learning, where the choice of which layers to freeze or unfreeze is tailored to maximize performance and efficiency according to the task requirements.