COMP 551 - Assignment 4 Report

Elizabeth Kourbatski

Galia Oliel-Sabbag

Colin Song

April 19, 2025

Abstract

In this project, we investigate the performance of the BERT base model on the AG News dataset using both probing strategies and end-to-end fine-tuning. We experimented with several sentence representation methods for probing, including the [CLS] token, mean pooling, and last-token embedding, as well as evaluating the effectiveness of extracting [CLS] embeddings from different transformer layers. Classification performance was evaluated using K-nearest neighbors and multi-class logistic regression. Additionally, we fine-tuned the entire BERT model and applied a parameter-efficient adaptation technique, LoRA (Low-Rank Adaptation), to compare performance with reduced trainable parameters. Our best probing-based approach achieved a test accuracy of **86.4%**, while full fine-tuning improved the accuracy to **92.75%**. LoRA achieved **90.25%** test accuracy, demonstrating competitive performance with significantly fewer trainable parameters. Attention visualizations from correctly and incorrectly classified examples highlighted the model's interpretability and revealed insights into how BERT allocates attention during prediction.

1 Introduction

This assignment investigates the effectiveness of pre-trained transformer representations for sentence classification using the AG News dataset. That is, we assess how well sentence-level embeddings extracted from a frozen BERT model perform when used with simple classifiers such as K-Nearest Neighbors (KNN) and logistic regression. We evaluate four embedding strategies: the [CLS] token from the final layer, the first token following [CLS], the last token in the input sequence, and mean pooling over all non-padding tokens.

We conducted a series of probing tasks. Among the strategies, mean pooling produced the best representations, achieving a test accuracy of 85.4% with KNN and 86.4% with logistic regression. The [CLS] token embeddings yielded 78.6% and 82.0% test accuracy respectively, while the first token performed poorly (63.8% with KNN, 76.2% with logistic regression). Last token embeddings achieved moderate performance at 76.8% and 81.6%, respectively.

To further understand the contribution how BERT's depth impacts representational quality, we evaluated [CLS] token embeddings extracted from all 12 layers. We found that layer 5 consistently outperformed other layers, achieving a test accuracy of 82.6% with KNN and 82.8% with logistic regression. This suggests that mid-level transformer layers likely provide better semantic representations.

While probing reveals what a pre-trained model has already learned, fine-tuning takes it a step further, allowing us to tailor the model to a specific task. End-to-end fine-tuning of all BERT parameters on 10,000 labeled examples improved test accuracy to 92.3%, demonstrating the strength of gradient-based adaptation. We later experimented with Low-Rank Adaptation (LoRA), a parameter-efficient alternative that fine-tunes only small adapter modules added to the attention layers. Despite training fewer parameters, LoRA achieved 90.25% test accuracy, proving it works well in situations with limited data or computing power.

Our findings reinforce observations made by Devlin et al. (2019): While [CLS] can serve as a useful sentence representation, richer embeddings, whether through pooling or intermediate layers, can significantly improve classification outcomes. In their study, Devlin et al. reported a test accuracy of 86.3% on the dataset using BERTLARGE—a result closely aligned with our own probing accuracy of 86.4%. Taken together, our results demonstrate that while probing offers valuable insight into the structure of BERT representations, task-specific fine-tuning can unlock even greater performance when adaptation to the data is possible.

2 Methods

2.1 Data Preprocessing

We used the AG News dataset, which was loaded using the Hugging Face `datasets` library. The dataset consists of news articles categorized into four classes: World, Sports, Business, and Sci/Tech. We split the original training set (120,000 examples) into 80% training and 20% validation subsets using stratified sampling to keep the same label distribution. The test set (7,600 examples) was used for final evaluation. For probing tasks, we restricted our training, validation, and test sets to smaller subsets of 2,000, 500, and 500 examples respectively, to save computing power.

2.2 Probing BERT Representations

To assess how well BERT encodes input texts without task-specific fine-tuning, we used a frozen `bert-base-uncased` model to extract sentence-level embeddings. We explored four token-level strategies: (1) the [CLS] token from the final layer, (2) the first token after [CLS], (3) the final non-padding token, and (4) the mean of all non-padding token embeddings. Each representation was used as input to a downstream classifier.

Additionally, we analyzed each layer separately by extracting the [CLS] token embedding from each hidden layer of BERT (layers 1 through 12). This was done to identify the layer that produces the most semantically informative representation for the classification task. We found that embeddings from layer 5 outperformed the final layer across classifiers, highlighting the relevance of intermediate layers for general-purpose semantic encoding.

2.3 K-Nearest Neighbors

We trained K-Nearest Neighbors (KNN) classifiers using the extracted BERT embeddings. We experimented with $k \in \{1, 3, 5, 7\}$ and selected the value yielding the highest validation accuracy for each strategy. The best-performing k was then used to report test accuracy.

2.4 Multiclass Logistic Regression

We also evaluated logistic regression classifiers using the same BERT-based embeddings. A softmax activation function was used to output probability scores over the four categories. Models were trained using cross-entropy loss with L2 regularization. Validation performance was monitored for model selection and early stopping.

2.5 Fine-tuning BERT

For comparison, we fully fine-tuned the `bert-base-uncased` model by adding a classification head and training all layers on 10,000 training samples. We used Hugging Face's `Trainer` API with three training epochs, a batch size of 16, and validation-based model selection. The best model was then evaluated on 2,000 test samples. Fine-tuning enabled all model weights to adapt to the classification task, leading to significantly higher performance compared to frozen probing methods.

2.6 LoRA Fine-Tuning

We further explored parameter-efficient fine-tuning using LoRA (Low-Rank Adaptation). Instead of updating the full weight matrices, LoRA freezes the original model and inserts trainable low-rank matrices into selected attention submodules. Specifically, we injected LoRA into the "query" and "value" projections of the transformer, with a rank $r = 16$ and dropout of 0.1. This reduced the number of trainable parameters while maintaining high performance. LoRA

models were trained under similar hyperparameter settings to the full fine-tuning baseline, using the same data splits and evaluation metrics.

2.7 Attention Visualization

To interpret model predictions, we used the `bertviz` library's `head_view` tool alongside a custom heatmap function. Attention maps were visualized from the [CLS] token to other input tokens, focusing on **Layer 10, Head 0**, which is often specialized for classification. In addition to visualizing the fine-tuned model, we overlaid a heatmap to highlight which tokens received the most attention. By comparing correctly and incorrectly predicted samples, we examined how attention patterns relate to prediction confidence and misclassification. These visualizations help us understand which words BERT focuses on when making its predictions.

3 Datasets

For this lab, we used the AG News dataset for text classification tasks. The dataset can be loaded directly in Python using the `datasets` library as follows:

```
from datasets import load_dataset
train_datasets = load_dataset('ag_news', split='train')
test_dataset = load_dataset('ag_news', split='test')
```

The dataset contains over 120000 training samples and 7600 test samples, each consisting of a news headline, a short description, and a label corresponding to one of four categories: World, Sports, Business, or Sci/Tech.

AG News is compiled from over 1 million news articles collected by the academic news search engine called ComeToMyHead. Articles originate from more than 2000 news sources and span over a year of activity. The dataset was compiled by Xiang Zhang et al. and it serves as a standard dataset for evaluating text classification models, particularly in clustering, classification, and information retrieval tasks.

Dataset	Number of Instances	Features	Target Parameter
AG News (Train)	120000	Headline & Description	News Category
AG News (Test)	7600	Headline & Description	News Category

Table 1: Summary of the AG News dataset used for this assignment. Each instance consists of a news headline and short description labeled into one of four categories: World, Sports, Business, or Sci/Tech.

4 Results

4.1 Probing

To evaluate the representational quality of frozen BERT embeddings, we explored four probing strategies: (1) the embedding of the `[CLS]` token from the final hidden layer, (2) the first actual token following `[CLS]`, (3) the final non-padding token, and (4) the mean of all non-padding token embeddings (mean pooling). For each strategy, we trained both a K-nearest neighbors (KNN) classifier and a multi-class logistic regression model. Performance was evaluated on a held-out validation set for model selection and on the test set for final reporting.

Table 2 summarizes the test accuracies for each probing configuration. Overall, logisitic regression tended to do better than KNN, suggesting that linear boundaries are effective in the high-dimensional space of BERT embeddings.

The [CLS] **token** strategy achieved a test accuracy of **78.6%** with KNN ($k = 7$) and **82.0%** with logistic regression. While this representation is commonly used in fine-tuning scenarios, it underperformed relative to pooling-based approaches in the frozen setting.

The **first token** strategy yielded the weakest results, with test accuracies of **63.8%** (KNN) and **76.2%** (logistic regression). This suggests that early tokens do not effectively capture the overall meaning of the sentence and, on their own, are not well-suited as sentence-level representations.

The **last token** strategy offered a moderate improvement, with **76.8%** (KNN) and **81.6%** (logistic regression) test accuracy. This indicates that sentence-final tokens encode more task-relevant context than the beginning of the sequence.

The best results were obtained through **mean pooling** over non-padding tokens. This strategy achieved **85.4%** accuracy with KNN and **86.4%** with logistic regression. Averaging token embeddings likely captures a more complete representation of sentence meaning, particularly for multi-topic headlines and longer inputs.

We also conducted a **layer-wise probing** analysis, extracting the [CLS] token embedding from each BERT layer to determine which depth produced the most useful representation. Results revealed that the 5^{th} layer consistently yielded the highest validation accuracy across classifiers. In Figure 1, we plot validation accuracy as a function of the layer index. For logistic regression, the layer 5 [CLS] embedding achieved **82.6%** test accuracy, outperforming both the final layer and earlier layers.

These results suggest that the fifth layer of BERT offers a useful balance between capturing syntactic structure and semantic meaning. Unlike the deepest layers, which are more specialized for BERT's pretraining objectives, the middle layers are less task-specific and better suited for general-purpose sentence representations. This makes them particularly effective for transfer tasks like classification in a frozen setting.

Strategy	**KNN (Best k)**	**Logistic Regression**
[CLS] Token (Last Layer)	78.6% (k=7)	82.0%
First Token	63.8% (k=5)	76.2%
Last Token	76.8% (k=5)	81.6%
Mean Pooling	85.4% (k=7)	**86.4%**
[CLS] Token (Layer 5)	82.6% (k=7)	82.8%

Table 2: Test accuracies for each probing strategy using KNN and logistic regression. Layer 5 [CLS] embedding outperformed the final layer.

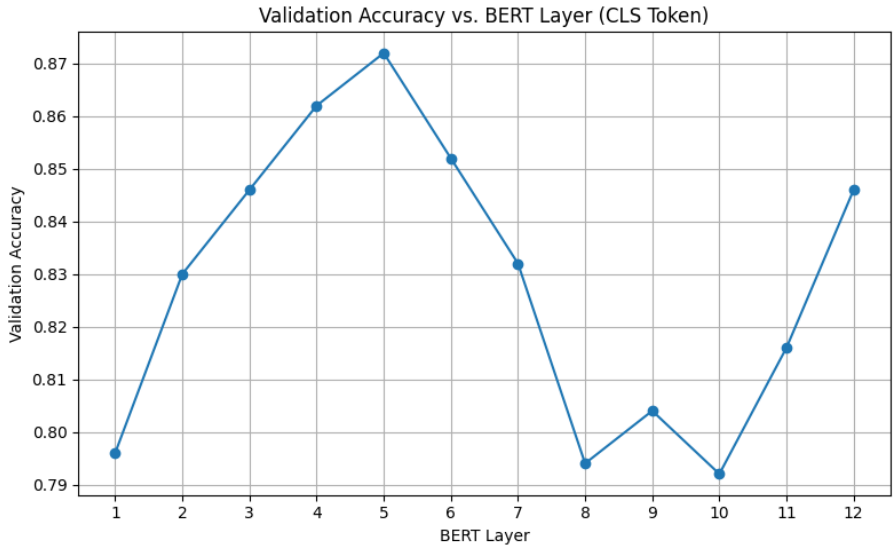

Figure 1: Validation accuracy of logistic regression using `[CLS]` token embeddings from each BERT layer. Layer 5 performs best.

4.2 Fine-tuning Bert and Attention Visualization

After fine-tuning the `bert-base-uncased` model on a subset of 10,000 training samples from the AG News dataset for three epochs, we achieved a test accuracy of **92.7%** on a held-out test set of 2,000 samples. This result significantly outperforms all probing-based approaches explored in this project, demonstrating the effectiveness of allowing gradient updates to propagate through all layers of the BERT architecture. Fine-tuning was conducted using Hugging Face's `Trainer` API with early stopping enabled based on validation accuracy.

To better understand how the model arrived at its predictions, we visualized attention weights using the `bertviz` library and a custom heatmap plotting function. The `head_view` tool in `bertviz` provides a full interactive visualization of self-attention scores between all tokens across each layer and attention head. For our analysis, we specifically examined attention patterns from the `[CLS]` token, which plays a central role in sentence-level classification. Since later layers tend to encode higher-level semantic representations, we focused on **Layer 10, Head 0** to explore how the model distributes attention during classification.

In correctly predicted examples, such as:

> *Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.*

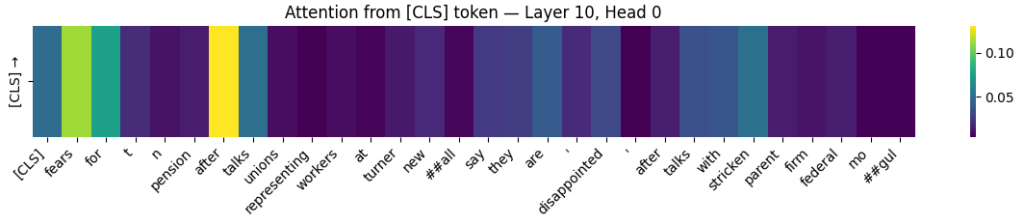

Figure 2: Attention from `[CLS]` token to input tokens (Layer 10, Head 0) for a correctly predicted example. Key thematic words like "fears", "talks", and "unions" receive high attention.

the `[CLS]` token directed strong attention to keywords such as *"fears"*, *"unions"*, and *"talks"*. These terms are highly indicative of the article's topic and likely guided the model's correct clas-

sification. Moderate attention was also observed for words like *"disappointed"* and *"stricken"*, while function words and subword fragments received minimal attention.

In contrast, misclassified samples showed more noisy attention. For example, in the sentence:

> *Some people not eligible to get in on Google IPO. Google has billed its IPO as a way for everyday people to get in on the process.*

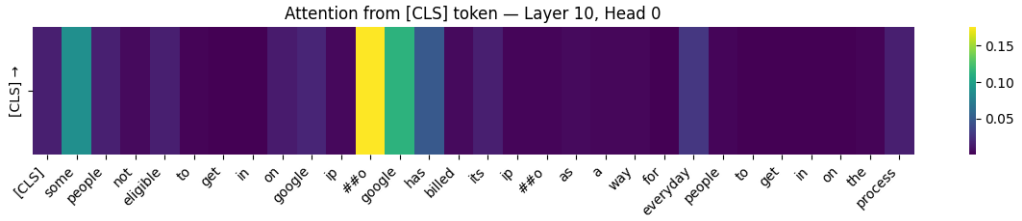

Figure 3: Attention from `[CLS]` token to input tokens (Layer 10, Head 0) for a correctly predicted example. Key thematic words like "some", "##0", and "google" receive high attention.

Here, the model concentrated attention on the word *"google"* and subword tokens like *"##o"*, rather than semantically rich tokens such as *"IPO"*, *"eligible"*, or *"everyday people"*. This misplaced attention may have led the model to focus too heavily on the entity *Google* while missing the broader context, contributing to the incorrect prediction.

These attention heatmaps helped us interpret how BERT allocates focus across tokens and layers. We selected Layer 10 because deeper layers typically encode higher-level semantic representations suited for classification. These observations suggest that different layers specialize in different types of information and highlight the interpretability and diagnostic power of attention visualization tools like `bertviz`.

4.3 LoRA Fine-Tuning

Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning technique introduced to reduce the computational cost and memory overhead associated with full fine-tuning of large language models. Rather than updating all weights in the model, LoRA introduces trainable low-rank matrices into specific layers of a frozen pre-trained model. These low-rank matrices are trained to capture task-specific information while the original model weights remain unchanged. This drastically reduces the number of parameters that need to be updated during training, making LoRA especially useful in resource-constrained environments.[4]

In our experiments, we applied LoRA to the `bert-base-uncased` model by injecting LoRA modules into the query and value projection matrices of the self-attention mechanism. We used a rank $r = 16$ and a LoRA-specific dropout rate of 0.1. Only the added LoRA parameters were trained while the rest of the BERT model remained frozen. Training was performed for three epochs on a 10,000-sample subset of the AG News training data.

The resulting LoRA-fine-tuned model achieved a test accuracy of **90.25%**, which is only slightly below the **92.75%** achieved by full fine-tuning. As shown in the confusion matrix (Figure 5), the LoRA model maintains strong performance, particularly in the `Sports` and `Sci/Tech` categories, with minor degradation in distinguishing `World` and `Business` articles.

Overall, LoRA provides a practical trade-off: significantly fewer trainable parameters with only a modest drop in classification performance. This makes it an attractive strategy for deploying large language models in production settings where computational efficiency and storage constraints are critical.

4.4 Comparison of Confusion Matrices: Full vs. LoRA Fine-Tuning

To evaluate the differences in classification behavior between the fully fine-tuned BERT model and the LoRA fine-tuned model, we compared their normalized confusion matrices on the 2,000-sample AG News test set (Figures 4 and 5). While the overall test accuracy of the fully fine-tuned model reached 92.75%, and the LoRA model closely followed with 90.25%, a closer look at the class-wise distributions reveals differences in their predictions.

In the `World` category, the fully fine-tuned model correctly classified 94.7% of examples, while the LoRA model achieved a lower accuracy of 88.6%. The LoRA model showed increased confusion between `World` and `Business` articles, which is not unexpected given the overlap in geopolitical and economic language used in both domains. For the `Sports` category, the LoRA model marginally outperformed the fully fine-tuned model, achieving 98.5% versus 98.3%. Both models showed near-perfect separation in this category, indicating that sports-related vocabulary was easy to recognize and consistent across examples.

In the `Business` category, both models faced the most classification challenges. The fully fine-tuned model obtained an accuracy of 86.9%, while the LoRA model achieved 86.0%. In both cases, the most frequent misclassifications occurred with the `Sci/Tech` category, likely due to shared terminology in corporate or product-related news. Similarly, in the `Sci/Tech` category, the full model achieved 90.3% accuracy, while LoRA followed closely at 87.2%, with minor confusion again directed toward the `Business` class.

Despite these modest differences, the LoRA model retained similar classification behavior to the fully fine-tuned model. These results support the claim that LoRA, despite training only a small number of additional parameters, preserves much of the predictive performance and decision-making structure of a fully fine-tuned model. Given its smaller computational cost and strong performance, LoRA is a practical alternative to full fine-tuning for adapting large language models to specific tasks.

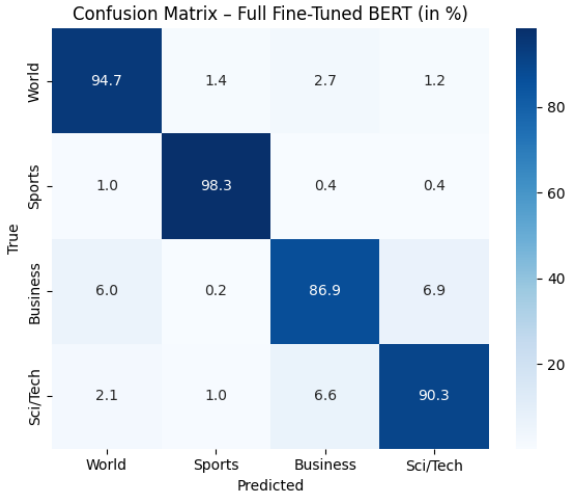

Figure 4: Confusion Matrix – Full Fine-Tuned BERT (in %)

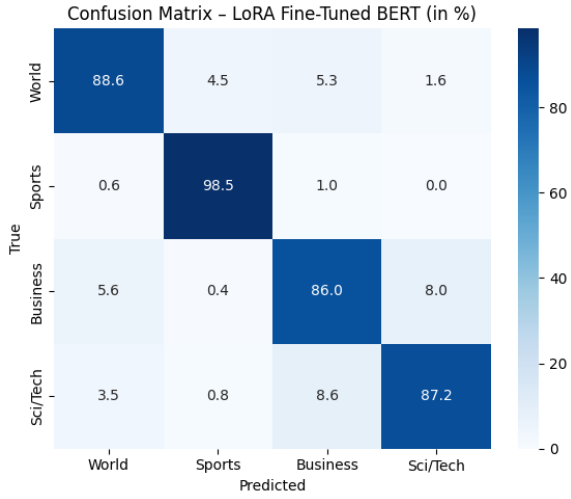

Figure 5: Confusion Matrix – LoRA Fine-Tuned BERT (in %)

5 Discussion and Conclusion

5.1 Probing BERT

This study evaluated the effectiveness of four BERT embedding strategies—[CLS] token, first token, last token, and mean pooling—on downstream news topic classification using K-Nearest Neighbors (KNN) and logistic regression. Our findings show that mean pooling consistently

outperformed all other strategies across both classifiers. Averaging the token embeddings over non-padding tokens achieved the highest KNN test accuracy (85.4%) and further improved under logistic regression (86.4%). This suggests that mean pooling better captures the overall sentence context by integrating information across the entire input, rather than focusing on any single token.

The [CLS] and last token strategies also demonstrated competitive performance, achieving 82.0% and 81.6% test accuracy respectively with logistic regression. These results indicate that these tokens encode useful summary information, likely due to their position at sentence boundaries and their training role in BERT's pretraining objective. In contrast, the first token strategy underperformed significantly, reaching a maximum of only 63.8% with KNN. This underperformance likely reflects the fact that early tokens alone are insufficiently representative of the full sentence's meaning.

Across all embedding strategies, logistic regression consistently outperformed KNN. This can be attributed to its ability to learn class-separating hyperplanes in high-dimensional embedding space, whereas KNN relies solely on local similarity and is less effective in sparse or overlapping distributions.

These results highlight that even without task-specific fine-tuning, frozen BERT representations—particularly when aggregated with mean pooling—provide strong features for classification. Probing offers a computationally lightweight and interpretable approach, making it a valuable baseline for evaluating pretrained language models.

5.2 Fine-tuning BERT

While probing provides a lightweight way to evaluate the semantic quality of frozen BERT embeddings, full fine-tuning unlocks the model's full capacity by updating all internal parameters. This allows the model to learn distinctions between AG News categories more effectively, resulting in the highest observed accuracy of 92.3%. LoRA offers a more computationally efficient alternative by fine-tuning only small, low-rank adapter modules. Remarkably, it achieved 90.3% accuracy, sacrificing only 2% compared to full fine-tuning. These results highlight a trade-off between efficiency and performance, however LORA performed almost as well as full-model tuning proving it is a strong contender for real-world applications where full fine-tuning is infeasible.

5.3 Attention Analysis

Fine-tuned BERT delivers significant performance improvements, achieving a test accuracy of 92.75% compared to 86.4% from the best probing strategy (mean pooling with logistic regression). This difference in test accuracy comes at the cost of substantially higher resource requirements (GPU acceleration, greater memory consumption, and longer training time). Fine-tuning also introduces additional complexity through hyperparameter tuning (e.g., learning rate, batch size, early stopping), which is not present in probing methods. Although probing yields lower test accuracy, it offers reasonable performance with simplicity, better computational efficiency, and potentially improved model interpretability.

To further understand the fine-tuned model, we conducted an attention analysis by visualizing attention matrices from selected transformer layers and heads. Specifically, we looked at how the [CLS] token focuses on other tokens in the input for both correct and incorrect predictions. In well-classified cases, we found that the [CLS] token often places strong attention weights on meaningful words, especially named entities and key topic words. This suggests that the model has learned to identify and leverage key indicators relevant to the news topic classification task. For misclassified examples, the attention distribution is often spread out or concentrated on common words and stopwords. In some cases, attention is misdirected due to formatting or punctuation, causing confusion in our model. This suggests that while BERT can understand complex language patterns, its predictions can still be affected by noise or misleading prompts.

6 Statement of Contribution

All team members contributed evenly to the coding and write-up portions of this project!

References

1. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* https://arxiv.org/abs/1810.04805

2. Zhang, X., Zhao, J., & LeCun, Y. *AG News Corpus.* Hugging Face Datasets. https://huggingface.co/datasets/ag_news

3. Professor Yue Li's GitHub: https://www.cs.mcgill.ca/~yueli/teaching/COMP551_Winter2025/comp551_winter2025.html

4. Understanding LoRA – Low Rank Adaptation For Finetuning Large Models: https://towardsdatascience.com/understanding-lora-low-rank-adaptation-for-finetuning-large-models-936bce1a07c6/