Melika Mohammadi Fakhar - 99522086 - HW6

1.

Scenario 1: Pretrained BERT Weights

- Initial Weights: The model starts with weights that have been pretrained on a large corpus of text.
- Training Process: This process is known as "fine-tuning." The model has already learned general language representations and only needs to adjust these representations to fit the specific task at hand.
- Performance During Training: The model is likely to perform well right from the start because it already has a good understanding of language. It will quickly adapt to the specific nuances of the new task.
- Performance After Training: The model is expected to perform very well after training, achieving high accuracy and efficiency on the specific task due to its strong foundational knowledge and fine-tuning.

Scenario 2: Random Initial Weights

- Initial Weights: The model starts with weights initialized to random values.
- Training Process: This process is known as "training from scratch." The model needs to learn everything from the beginning, including basic language representations and the specific task.
- Performance During Training: The model is likely to perform poorly at the start because it has no prior knowledge. It will take much longer to converge and may require significantly more data and computational resources.
- Performance After Training: The model may eventually perform well, but it is unlikely to reach the same level of performance as the fine-tuned model, especially if the dataset is not extremely large and diverse.

Expected Performance Comparison

- Training Process: The pretrained model will perform better during the training process, showing quicker improvements and higher initial performance.

- After Training: The pretrained model is expected to outperform the model with random initial weights, especially in scenarios with limited data or specific tasks that benefit from prior language understanding.

2.

Catastrophic Forgetting in Neural Networks

- Challenge Explanation: Catastrophic forgetting, also known as catastrophic interference, is a significant challenge in training neural networks, particularly during fine-tuning. It occurs when a neural network, after learning a new task, substantially forgets previously learned tasks. This issue arises because the training process updates the model weights, and new updates can overwrite the weights important for the old tasks, leading to a drastic decline in performance on those previously learned tasks.

Reducing or Eliminating Catastrophic Forgetting

- One effective method to mitigate catastrophic forgetting is Elastic Weight Consolidation (EWC). EWC works by selectively slowing down the learning on weights that are crucial for previously learned tasks. It uses the Fisher Information Matrix to identify these important weights and applies a regularization term to preserve them during the learning of new tasks. This approach allows the model to retain knowledge of old tasks while still learning new ones effectively.

Sources:

1. Wikipedia on Catastrophic Interference
2. Overcoming Catastrophic Forgetting in Neural Networks
3. Papers with Code on Overcoming Catastrophic Forgetting

3.

Transfer Learning vs. Fine-Tuning

Transfer Learning:

- Concept: Uses a pre-trained model on a new but related task. The early layers are frozen, and only the final layers are retrained on the new dataset.
- Usage: Ideal for small datasets and tasks similar to the original training task.

- Conditions: Effective when data is limited and the new task is somewhat related to the original task. Requires fewer resources and less time.

Fine-Tuning:
- Concept: Involves updating some or all of the pre-trained layers along with the new layers. This allows the model to adapt more specifically to the new task.
- Usage: Beneficial for larger datasets and tasks closely related to the original dataset.
- Conditions: Suitable for tasks with a larger dataset and high similarity to the original task. Requires more resources and careful tuning to avoid overfitting.

Key Differences:
Training Approach:
- Transfer Learning: Freeze most pre-trained layers.
- Fine-Tuning: Retrain some or all pre-trained layers.

Domain Similarity:
- Transfer Learning: Effective for somewhat similar tasks.
- Fine-Tuning: More effective for highly similar tasks.

Resources:
- Transfer Learning: Fewer resources needed.
- Fine-Tuning: More resources required.

Dataset Size:
- Transfer Learning: Suitable for small datasets.
- Fine-Tuning: Suitable for larger datasets.

4.
Effects of Masking Methods on MLM Training and Performance

1. Random Masking Method:
Concept: Randomly selects 15% of tokens to mask in each training instance.
Impact on Training:
Pros: Provides diverse masked tokens, helping the model learn robust representations.

Cons: Might not capture complex dependencies well.
Performance: Generalizes well across tasks and is computationally efficient.

2. Part of Speech (POS)-based Masking Method:
Concept: Selects tokens based on their part of speech (e.g., nouns, verbs).
Impact on Training:
Pros: Improves understanding of syntactic and semantic roles, beneficial for tasks requiring linguistic details.
Cons: More complex and computationally intensive due to the need for POS tagging.
Performance: Enhances performance on tasks needing syntactic knowledge (e.g., named entity recognition).

Determining the Amount of Maskable Tokens
Masking Rate:
Standard Rate (15%): Balances context and learning challenges.
Lower Rate (<15%): Provides more context but may lead to underfitting.
Higher Rate (>15%): Increases learning difficulty and robustness but might lead to overfitting.

5.
Performance Comparison of CLM, MLM, and Seq2Seq Architectures

1. Causal Language Models (CLM):
Example: GPT-3
Method: Predicts the next word based on previous words.
Advantages: Excellent at generating coherent text; simple implementation.
Disadvantages: Only uses previous context, limiting understanding of surrounding context; less effective for tasks needing bidirectional context.
Source: GPT-3 by OpenAI

2. Masked Language Models (MLM):
Example: BERT
Method: Predicts missing words in a sentence by masking certain tokens.
Advantages: Considers both previous and future context, versatile for many tasks like text classification and question answering.
Disadvantages: Complex training process; not ideal for text generation.
Sources: BERT by Google, RoBERTa

3. Sequence-to-Sequence Models (Seq2Seq):
Example: T5
Method: Transforms an input sequence into an output sequence using encoder-decoder architecture.
Advantages: Excels at tasks like translation and summarization; handles complex input-output transformations.
Disadvantages: Resource-intensive and complex to train; dependent on data quality.
Sources: T5 by Google, BART by Facebook

Comparison
CLM (e.g., GPT-3):
Pros: Great for text generation, simple to implement.
Cons: Limited by unidirectional context.
MLM (e.g., BERT):
Pros: Bidirectional context, versatile.
Cons: Complex training, less suited for generation tasks.
Seq2Seq (e.g., T5):
Pros: Ideal for input-output tasks, versatile.
Cons: Complex and resource-heavy, quality-dependent.

6.
Using MLM Models for Text Generation
1. Masked-to-Unmasked Iterative Process:
Method: Iteratively mask and predict tokens, gradually building up the text.
Advantages: Ensures contextually coherent text.
Disadvantages: Computationally intensive and may not generate fluid text.
Example: Start with "The capital of France is [MASK]." and iteratively predict and replace the masked token.

2. Conditional Generation:
Method: Use MLMs to generate text based on a given prompt.
Advantages: Strong contextual understanding.
Disadvantages: Maintaining coherence can be challenging.
Example: Provide a context and mask a portion for MLM to predict and expand.

3. Combining MLM with Generative Models:
Method: Use MLMs for context understanding and autoregressive models for text generation.

Advantages: Leverages the strengths of both models for better text generation.
Disadvantages: Complex implementation and high computational requirements.
Example: Use BERT to refine context and GPT-3 to generate text.

Overall
Iterative Process: Good for coherent context but resource-heavy.
Conditional Generation: Effective for context-dependent text.
Hybrid Models: Combines strengths for optimal text generation.

7.
Q1: Understanding the Masking Strategy in Masked Language Models

Masked Language Models (MLMs) like BERT use a specific strategy during pre-training to help the model learn to predict missing words within a sentence. This strategy involves replacing certain tokens in the training data with special markers or random tokens. Here's a detailed explanation of the rationale behind this strategy:

80% Masked with [MASK] Token
Rationale:
Direct Learning Target: Replacing 80% of the masked tokens with the [MASK] token provides the model with a clear learning target. The model learns that whenever it sees the [MASK] token, it needs to predict the original token based on the surrounding context.
Impact on Training:
Contextual Learning: This percentage ensures that the model focuses on understanding and leveraging the context provided by the surrounding words. By frequently seeing the [MASK] token and having to predict the missing word, the model learns to capture the dependencies and relationships between words.
Effective Utilization: The frequent occurrence of the [MASK] token helps the model effectively utilize the pre-training phase to develop a robust understanding of language patterns.

10% Replaced with Random Words
Rationale:

Noise Introduction: Replacing 10% of the masked tokens with random words introduces noise into the training data. This noise simulates the natural variability and unpredictability of real-world language use.
Impact on Robustness:
Handling Novel Inputs: By encountering random words in place of masked tokens, the model becomes more robust to unexpected or novel inputs. It learns to handle scenarios where the input may not be clean or predictable.
Avoiding Overfitting: This strategy also helps in preventing the model from overfitting to the [MASK] token, as it learns to make predictions even when the input is not straightforward.

10% Left Unchanged
Rationale:
Realistic Scenario Simulation: Leaving 10% of the masked tokens unchanged helps in simulating a more realistic scenario where some tokens are not masked. This reflects the natural use of language where not every word needs to be predicted or masked.
Impact on Generalization:
Avoiding Overfitting to [MASK]: By having a portion of the tokens left unchanged, the model learns that it should not always rely on the presence of the [MASK] token for making predictions. This helps in preventing overfitting specifically to the [MASK] token.
Generalization Ability: This strategy encourages the model to generalize better across different contexts, as it learns to predict and understand words in their natural form as well.

Q2.
1. Pretraining on a Large Corpus
Action: Pretrain on a large, diverse dataset like Wikipedia or Common Crawl.
Benefit: Helps the model learn general language patterns and knowledge.
2. Fine-tuning on Task-Specific Data
Action: Fine-tune the pretrained model on your specific dataset with a lower learning rate.
Benefit: Adapts general language knowledge to the nuances of your specific task.
3. Data Augmentation

Action: Use techniques like back-translation and synonym replacement.
Benefit: Increases data diversity and helps the model generalize better.

4. Hyperparameter Optimization

Action: Conduct hyperparameter searches (e.g., grid search, Bayesian optimization).
Benefit: Identifies optimal settings for improved model performance.

5. Regularization Techniques

Action: Apply dropout and weight decay.
Benefit: Prevents overfitting, improving the model's generalization.

6. Incorporating Additional Features

Action: Integrate linguistic features like part-of-speech tags or named entities.
Benefit: Provides additional context, enhancing model performance.

7. Leveraging Transfer Learning

Action: Use models pretrained on related tasks and fine-tune them.
Benefit: Offers a good initialization and improves performance on your task.