Title: GEOMETRIC REPROJECTION INSTRUCTION TUNING FOR LANGUAGE MODEL ADAPTATION

FIELD OF INVENTION

[0001] The present disclosure relates to machine learning and natural language processing systems, and more particularly to efficient fine-tuning methods for pre-tuned large language models using geometric reprojection instruction tuning that incorporates curvature aware updates and dynamic subspace projection.

BACKGROUND OF INVENTION

[0002] Large language models have transformed natural language processing by demonstrating remarkable capabilities in text generation, comprehension, and reasoning tasks. These models, which contain hundreds of millions to billions of parameters, are typically pre-trained on vast corpora of text data to learn general language representations. However, adapting these pre-trained models to perform well on specific downstream tasks or follow particular instructions remains a computationally intensive challenge.

[0003] Traditional fine-tuning approaches involve updating all parameters of a pre-trained model during adaptation, which requires substantial computational resources and memory. This full-parameter fine-tuning can be prohibitively expensive for large models and may lead to catastrophic forgetting of previously learned knowledge. The computational burden becomes particularly pronounced when adapting models across multiple tasks or domains, as separate copies of the entire model must be maintained for each adaptation.

[0004] To address these computational challenges, parameter-efficient fine-tuning methods have emerged as an alternative approach. These techniques aim to achieve comparable performance to full fine-tuning while updating only a small subset of the model's parameters. Low-rank adaptation methods such as LoRA represent one category of such approaches, introducing trainable low-rank matrices that approximate updates to the original model weights while keeping the base parameters frozen. Although LoRA and its derivatives, like QLoRA, have improved training efficiency, but they do not account for the underlying geometry of the model's loss surface. As a result, they may overlook critical update directions that could enhance convergence and final model performance.

[0005] Despite improvements in training efficiency, existing parameter-efficient methods generally apply uniform adaptation strategies across all layers of a model. This approach may

not account for the varying adaptation requirements of different layers or the underlying geometric properties of the model's loss landscape. The optimization process in these methods typically relies on first-order gradient information, which may overlook directions in the parameter space that could enhance convergence and model performance.

[0006] Furthermore, conventional parameter-efficient techniques often employ fixed-rank adaptations that do not adjust to the complexity or requirements of specific tasks. This static approach can result in over-parameterization for simple tasks or under-parameterization for complex ones. The lack of dynamic adaptation mechanisms limits the flexibility and efficiency of these methods across diverse applications and datasets.

[0007] The field continues to explore methods that can leverage geometric information about the loss surface, incorporate higher-order optimization techniques, and dynamically adjust adaptation capacity based on task demands. Such approaches seek to improve both the efficiency and effectiveness of large language model adaptation while maintaining compatibility with existing training infrastructure and hardware constraints.

[0008] Accordingly, there is a need of techniques that combines geometric information, dynamic capacity allocation, and informed parameter updates to enhance the performance and efficiency of large model adaptation. The present invention discloses such techniques, thereby resulting in more scalable, targeted, and cost effective strategy for adapting large scale models.

OBJECTIVES OF INVENTION

[0009] The objectives described herein are merely exemplary and are not intended to be limiting. It will be apparent to those skilled in the art that various modifications and variations can be made to the disclosed embodiments without departing from the scope of the invention.

[0010] The primary objective of the present invention is to provide techniques for efficient fine-tuning of large language models for instruction following and task specific adaptation.

[0011] Another objective of the present invention is to provide a method for efficient fine-tuning of large language models through a Geometric Reprojection Instruction Tuning (GRIT) framework that enhances low rank adaptation by incorporating curvature aware updates and dynamic subspace projection.

[0012] Yet another objective of the present invention is to incorporate geometric information about the loss surface to guide parameter updates more effectively by approximating second order information through Kronecker-factored approximations of the

Fisher Information Matrix, enabling gradient preconditioning that aligns parameter updates with the natural geometry of the parameter space to improve convergence stability and training efficiency in low data or instruction tuning conditions.

[0013] Yet another objective of the present invention is to provide a periodic reprojection step to reduce parameter redundancy and enhance generalization. This step evaluates the information content in the low rank subspace and adaptively reorients or compresses it using eigendecomposition or related techniques.

[0014] Yet another objective of the present invention is to achieve substantial parameter efficiency by reducing the number of trainable parameters while maintaining or improving model performance, enabling more compact model adaptation with reduced computational and memory requirements.

[0015] Yet another objective of the present invention is to provide enhanced training stability and convergence through curvature-aware optimization that prevents divergence and oscillatory behavior, particularly when fine-tuning across tasks with different distributions or complexity levels.

[0016] Yet another objective of the present invention is to ensure compatibility with existing training infrastructure while providing scalability across different model architectures and training regimes, enabling seamless integration into widely used machine learning frameworks with minimal modification requirements.

SUMMARY OF INVENTION

[0017] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the detailed description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0018] The present invention provides a novel framework for efficient fine-tuning of large language models through a novel Geometric Reprojection Instruction Tuning (GRIT) framework, which addresses computational limitations of traditional parameter-efficient fine-tuning methods by incorporating curvature-aware optimization using curvature aware updates and dynamic subspace compression using dynamic subspace projection.

[0019] The invention augments pre-trained language models with low-rank adaptation modules strategically inserted into selected layers, where each module comprises paired

trainable matrices that approximate low-rank updates while preserving frozen base parameters. During training, the method selectively captures input activations and output gradients according to a configurable sampling schedule, enabling efficient statistical analysis without excessive memory overhead.

[0020] A key innovation involves computing Kronecker-factored approximations of the Fisher Information Matrix from captured data, decomposing second-order curvature information into manageable covariance matrices. These approximations precondition gradient updates to align with the geometric properties of the loss surface, resulting in improved convergence stability and training efficiency.

[0021] The method incorporates a dynamic reprojection mechanism that performs eigendecomposition on the Fisher Information Matrix to identify principal directions of parameter sensitivity. Learned parameter changes are projected onto subspaces spanned by the most significant eigenvectors, compressing the effective update space while retaining components with highest impact on model performance.

[0022] The present invention demonstrates substantial improvements over existing methods, achieving a 43.9% reduction in trainable parameters compared to QLoRA while delivering superior performance across multiple evaluation metrics including BLEU scores, Rouge metrics, and semantic similarity measures. This combination of geometric awareness, dynamic adaptation, and principled compression enables scalable fine-tuning of large language models with reduced computational requirements and enhanced task-specific performance.

[0023] The present invention is modular, hardware efficient, and compatible with various model architectures, objectives, and hardware environments. It significantly lowers training costs while preserving or enhancing model accuracy and enables deployment of large models in resource constrained settings. The GRIT framework represents a paradigm shift from uniform parameter adaptation to geometry-aware optimization that leverages the natural curvature of the loss landscape.

BRIEF DESCRIPTION OF FIGURES

[0024] Non-limiting and non-exhaustive examples are described with reference to the following figures.

[0025] FIG. 1 illustrates a flow diagram of a Geometric Reprojection Instruction Tuning (GRIT) fine-tuning process with Kronecker-Factored Approximate Curvature (KFAC)

preconditioning and Low-Rank Adaptation (LoRA) matrix adaptation, according to aspects of the present disclosure.

[0026] FIG. 2 depicts a comparison of parameter updates between LoRA and GRIT across multiple transformer layers, according to an embodiment.

DETAILED DESCRIPTION

[0027] The following description describes various features and functions of the disclosed invention with reference to the accompanying figures. In the figures, similar symbols identify similar components, unless context dictates otherwise. The illustrative aspects described herein are not meant to be limiting. It may be readily understood that certain aspects of the disclosed invention can be arranged and combined in a wide variety of different configurations, all of which are contemplated herein.

[0028] Accordingly, those of ordinary skill in the art will recognize that various changes and modifications of the embodiments described herein can be made without departing from the scope of the invention. In addition, descriptions of well-known functions and constructions are omitted for clarity and conciseness.

[0029] Features that are described and/or illustrated with respect to one embodiment may be used in the same way or in a similar way in one or more other embodiments and/or in combination with or instead of the features of the other embodiments.

[0030] The terms and words used in the following description and claims are not limited to the bibliographical meanings but are merely used to enable a clear and consistent understanding of the invention. Accordingly, it should be apparent to those skilled in the art that the following description of exemplary embodiments of the present invention are provided for illustration purposes only and not to limit the invention.

[0031] It is to be understood that the singular forms “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise.

[0032] While this invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not limited to the disclosed embodiments, but, on the contrary, is intended to cover various modifications and equivalent arrangements included within the scope of the appended claims.

[0033] The present disclosure provides a method for fine-tuning large language models through a novel Geometric Reprojection Instruction Tuning (GRIT) framework. The GRIT framework addresses computational and representational limitations inherent in traditional parameter-efficient fine-tuning techniques by introducing a curvature-aware, dynamically compressible adaptation framework that improves convergence, model stability, and parameter utilization during fine-tuning of pre-trained large language models. The GRIT framework integrates three core components including a modular low-rank adaptation, an optimization component guided by second-order curvature information, and a reprojection mechanism that dynamically compresses parameter updates based on statistical energy. Together, these elements form a cohesive system that enables scalable, geometry-aware adaptation of large-scale models with minimal resource overhead and memory usage.

[0034] It should be noted that, although the traditional parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) reduce the number of trainable parameters by introducing trainable low-rank matrices while keeping original model weights fixed, but they do not account for the underlying geometry of the model's loss surface and may overlook update directions that could enhance convergence and final model performance. Furthermore, existing parameter-efficient fine-tuning techniques generally apply uniform adaptation strategies across all layers of the model, which can be inefficient as different layers may have varying adaptation needs. GRIT overcomes these limitations by incorporating geometric information about the loss landscape to guide parameter updates more effectively.

[0035] Unlike traditional parameter-efficient fine-tuning methods, the GRIT framework leverages second-order curvature information to precondition gradient updates, leading to faster convergence and improved training stability, particularly in low-data or instruction-tuning scenarios. Rather than applying uniform updates across all parameters, GRIT selectively focuses updates along directions of high functional importance as determined by the curvature of the loss surface. This approach allows the framework to optimize a smaller subset of parameters while maintaining or improving downstream performance compared to conventional methods.

[0036] A distinguishing feature of the GRIT framework implemented method lies in the periodic reprojection step that evaluates information content in the low-rank subspace and adaptively compresses the subspace using eigendecomposition techniques. This reprojection mechanism enables dynamic adjustment of the effective rank during training without manual

hyperparameter tuning, as the rank emerges naturally from the spectral properties of the loss surface.

[0037] In an embodiment, a GRIT based method is provided for fine-tuning of pre-trained language models. The method begins by augmenting a pre-trained language model with one or more low-rank adaptation modules strategically inserted into selected layers of the transformer architecture, such as multi-head attention, feed-forward networks, and linear projections. Each low-rank adaptation module comprises a pair of trainable matrices: a first down-projection matrix that projects input representations to a lower-dimensional latent space, and a second up-projection matrix that maps the compressed representations back to the original output dimension. This paired matrix structure enables task-specific adaptations through rank-constrained modifications without altering the pre-trained weights. These adaptation modules operate in parallel with the original weights, which remain frozen throughout fine-tuning, thereby preserving the integrity and generalization capabilities of the base model.

[0038] The present invention supports flexible placement of these low-rank adaptation modules across the model hierarchy, unlike traditional low-rank methods that statically define adaptation scope. The selection of insertion points is performed either as a manual selection based on empirical insights or an automatic selection using heuristics derived from activation statistics or layer importance scores. For instance, middle and deeper layers of a transformer often contain more task-relevant representations and may therefore benefit more from adaptive fine-tuning than early layers. The present invention also allows varying module rank across layers, allowing higher-rank modules to be allocated to more sensitive or expressive layers. These design choices allow the GRIT adaptability to a wide range of architectures and deployment contexts.

[0039] During training, the present invention selectively captures input activations and output gradients from a chosen subset of low-rank modules. A configurable sampling schedule that trades off fidelity against memory consumption governs the frequency of capture. The configurable sampling schedule may include periodic sampling at predetermined intervals, which may be dynamically adjusted based on training stability, with more frequent sampling during unstable periods and reduced sampling during convergence phases. The method achieves overhead minimization while maintaining sufficient coverage to estimate local curvature information by avoiding constant sampling. The selected samples are buffered temporarily and used to construct compact statistical representations that summarize the relationship between input and output changes under model adaptation. Specifically, the

method captures covariance structures of the inputs and gradients, which serve as sufficient statistics for approximating second-order information.

[0040] The core geometric insight behind GRIT is the use of an approximate Fisher Information Matrix to guide updates. The Fisher Information Matrix characterizes local curvature of the model's parameter space with respect to the loss function, highlighting directions where small changes in parameters have large effects on the output distribution. The direct computation of the full Fisher Matrix is intractable for modern models due to the high dimensionality of their parameter space. To address this, the present invention adopts a scalable approximation based on a Kronecker-factored decomposition, wherein the Fisher Matrix is represented as the product of two lower-order covariance matrices. One matrix captures variation in the activations and the other matrix captures variation in the gradients. These matrices are substantially smaller than the full Fisher Matrix and enable efficient computation from the captured training samples.

[0041] Once constructed, inversion of the Kronecker-factored matrices occurs either analytically, using regularization if necessary, or numerically with efficient solvers. The inverses then precondition the gradient updates applied to the low-rank parameters. The preconditioning step adjusts the scale and orientation of the gradient vector in a manner that aligns it with the natural geometry of the model. This results in more stable updates that converge more rapidly, particularly in poorly scaled regions of the loss surface or in scenarios with limited data availability. This approach generalizes standard gradient descent by applying transformation rules derived from the second-order structure of the parameter landscape.

[0042] To avoid excessive computational load, the GRIT introduces a dynamic scheduling controller that governs when second-order statistics are updated. The controller evaluates real-time training indicators such as gradient norm variance, change in loss value, and update consistency to determine if a curvature refresh is necessary. If the model exhibits stable behavior, the same curvature estimate is reused for multiple steps. If the controller detects instability or drift, it triggers recomputation of the statistical approximations. This dynamic control mechanism ensures computation focus where and when it is most impactful, minimizing redundant operations while preserving convergence benefits.

[0043] Following the gradient preconditioning step, the present invention proceeds with the implementation of a neural reprojection mechanism that compresses the effective update space by projecting learned parameter changes onto a statistically informed subspace. This

process begins with eigendecomposition of the estimated Fisher Information Matrix to identify its principal eigenvectors, which represent the directions in parameter space associated with the greatest sensitivity. The update vectors derived from gradient descent are then projected into the subspace spanned by the top eigenvectors, preserving only the most impactful components. The directions corresponding to lower eigenvalues, which represent flatter or noisier regions of the loss surface, are discarded. This projection achieves component elimination that contributes little to optimization or generalization, thereby refining the update and reducing its effective dimensionality.

[0044] The motivation behind neural reprojection comes from the observation that during instruction tuning or domain adaptation, many parameter updates are either noisy or redundant, especially in low-rank adaptation layers. In practice, few directions in parameter space significantly affect the loss landscape, while the rest contribute minimally or even degrade performance when retained. By identifying these impactful directions through the spectral analysis of second-order statistics, a smaller, more meaningful subspace within which the optimization is most effective can be isolated. This allows both compression and enhancement of the adaptation dynamics, preserving generalization performance with fewer degrees of freedom.

[0045] This approach realizes periodic analysis of the second-order curvature of the loss surface by performing eigendecomposition on the Fisher Information Matrix, which is approximated via projected input activations and output gradients from the selected modules. The eigenvectors associated with the largest eigenvalues define a principal subspace capturing the most significant curvature directions. These eigenvectors serve as a data-driven basis for filtering parameter updates, ensuring that only components aligned with the dominant learning dynamics are retained. This achieves alignment of the update with directions where the model is most sensitive, promoting both convergence stability and task fidelity.

[0046] The practical implementation of this process ensures computational tractability and compatibility with large-scale training regimes. Instead of operating on full-rank matrices, the algorithm focuses on low-rank LoRA modules, whose dimensionality is already constrained. The practical implementation maintains efficiency even when applied to large-scale models. Since the method operates within low-rank adaptation layers, it involves small matrices well-suited for fast and stable decomposition. The reprojection step is performed at strategic intervals, not every iteration, and is guided by training signals such as gradient variance and loss trends. This separation from the main optimization loop ensures lightweight core training

process maintenance while still benefiting from periodic curvature-informed refinement of the update directions. Tracking intermediate statistics such as eigenvalues across time achieves energy-based rank pruning facilitation without requiring manual heuristics or additional validation cycles.

[0047] The retained eigenvectors can be viewed as encoding the principal directions of learning for a specific task or domain, providing insight into which aspects of the model's representation are most relevant. This offers particular advantage in settings such as personalization or federated learning, where the underlying data distribution may shift over time. By automatically re-aligning the adaptation space based on statistical feedback, robustness across tasks and compactness maintenance of the trained model is achieved, without degrading inference latency or requiring post-hoc compression.

[0048] The projection results are a transformed set of low-rank parameters that encode a denoised and task-aligned update. These parameters are used to reconstruct the low-rank adaptation matrices, which are then applied to the model during the next training steps. The reprojection step is performed periodically, and its frequency may be tied to the controller signals that govern curvature estimation. In practice, cumulative energy criteria, such as retaining a fixed percentage of total spectral energy, may serve as the basis for the threshold for eigenvalue truncation. This reprojection-aware framework provides enhanced interpretability and adaptability in parameter-efficient fine-tuning.

[0049] The reprojection mechanism in GRIT enables reduction of the number of effective trainable parameters during training without sacrificing accuracy. This method achieves distinction from conventional fixed-rank strategies, which may over-parameterize or under-parameterize specific layers depending on task complexity. In GRIT, the rank of the update that emerges is characterized by spectral filtering consequences rather than direct hyperparameter tuning. This method achieves responsiveness to the underlying learning dynamics and avoids the need for exhaustive grid searches over rank configurations.

[0050] Now referring to FIG. 1, which illustrates a flow diagram 100 of a Geometric Reprojection Instruction Tuning (GRIT) based fine-tuning process for pre-tuned large language models, wherein the GRIT is implemented with Kronecker-Factored Approximate Curvature (KFAC) preconditioning and Low-Rank Adaptation (LoRA) matrix adaptation, according to aspects of the present disclosure. In some embodiments, the disclosed GRIT based fine-tuning process may be implemented as a system for efficient adaptation of large language models. The

system may comprise a processor and a memory storing instructions, when executed by the processor, cause the system to perform a plurality steps illustrated in flow diagram 100 and described herein to achieve the objectives of the present invention, in accordance with FIG. 1.

[0051] In some embodiments, the flow diagram 100 of the GRIT based fine-tuning process of FIG. 1 follows a structured workflow having steps/ blocks 102 to 120 that begins with training data input step 102 and progresses through multiple interconnected stages to achieve efficient parameter adaptation. The diagrammatic flow 100 illustrates how training data enters the system implementing the GRIT fine-tuning process which flows through a pre-trained large language model (LLM) at step 104, where the model processes the input through its existing architecture while maintaining frozen base parameters. The pre-trained LLM serves as the foundation for the adaptation process, providing the established language understanding capabilities that will be refined through the GRIT framework. The training data flows from the pre-trained LLM to target layers and modules that have been selected for low-rank adaptation, where the framework applies its geometric optimization techniques.

[0052] The target layers and modules at block 106 undergo both forward 108a and backward 108b computational passes that capture different types of statistical information for curvature estimation. During the forward pass 108a, the system performs activation capture and covariance estimation at step 110, where input activations flowing through the selected low-rank adaptation modules are recorded and analyzed to construct statistical representations of the data distribution. The forward pass 108a enables the framework to observe how input patterns propagate through the adapted layers, providing insights into the activation patterns that characterize the training data. Simultaneously, the backward pass 108b captures gradients and performs covariance estimation at step 112, where error signals propagating backward through the low-rank adaptation modules are recorded and processed to understand the sensitivity patterns of the loss function with respect to parameter changes.

[0053] Further, the covariance estimation processes from both forward 108a and backward 108b passes feed into the K-FAC gradient preconditioning component at step 114, which represents the core geometric optimization mechanism of the GRIT framework. The K-FAC gradient preconditioning component receives the covariance matrices derived from activation capture and gradient capture operations, combining these statistical inputs to construct Kronecker-factored approximations of the Fisher Information Matrix. The preconditioning process transforms the original gradient updates by incorporating second-order curvature information, producing preconditioned gradients that align with the geometric properties of the

loss surface. The K-FAC gradient preconditioning leads to parameter updates with preconditioned gradients at step 116, where the modified gradient vectors guide parameter modifications in directions that account for local curvature characteristics.

[0054] The parameter updates with preconditioned gradients flow into the neural reprojection and rank adaptation stage at step 118, which provides dynamic compression of the effective parameter space through spectral analysis and subspace projection. The neural reprojection component analyzes the spectral properties of the parameter updates and applies eigendecomposition techniques to identify principal directions of parameter sensitivity. The rank adaptation mechanism adjusts the effective dimensionality of the parameter updates by projecting them onto subspaces defined by the most impactful eigenvectors, thereby reducing computational overhead while preserving the components that contribute most substantially to model performance. The reprojection process filters out parameter update components that align with flatter regions of the loss surface, concentrating the adaptation capacity on directions where modifications produce meaningful improvements.

[0055] The final stage of the GRIT process involves updating and compressing LoRA matrices at step 120, where the results of the neural reprojection and rank adaptation are applied to modify the low-rank adaptation modules. The LoRA matrix updates incorporate the compressed parameter changes that have been filtered through the geometric optimization pipeline, ensuring that the adaptations focus on the most impactful directions in parameter space. The compression aspect of this stage reduces the effective parameter count by eliminating redundant or low-impact components from the adaptation matrices, resulting in more efficient parameter utilization without sacrificing task performance. The updated and compressed LoRA matrices complete one iteration of the GRIT fine-tuning cycle, where the modified low-rank adaptation modules are ready to process subsequent training data with improved geometric alignment and reduced computational overhead.

[0056] The cyclical nature of the process shown in FIG. 1 enables continuous refinement of the low-rank adaptation modules through repeated iterations of the geometric optimization pipeline. Each iteration builds upon the previous adaptations by incorporating updated statistical information from the training data, allowing the framework to adapt to changing patterns in the data distribution and optimization landscape. The integration of forward pass activation capture, backward pass gradient capture, K-FAC preconditioning, neural reprojection, and LoRA matrix compression creates a comprehensive adaptation mechanism that addresses both computational efficiency and optimization effectiveness. The diagrammatic

flow demonstrates how each component contributes to the overall framework while maintaining clear data dependencies and processing sequences that enable efficient implementation across different hardware environments and model architectures.

[0057] To demonstrate the practical advantages of the GRIT framework, a comprehensive empirical evaluation was conducted comparing the present invention against the traditional low-rank adaptation method of Quantized Low-Rank Adapter (QLoRA) using the Llama 3.2-3B model as a benchmark. This comparative analysis assessed both computational efficiency and task performance across multiple standard evaluation metrics, with results illustrated in FIG. 2. The GRIT based method achieved a 43.90% reduction in the number of trainable parameters relative to the traditional QLoRA, demonstrating its ability to adapt models more compactly. Despite the lower parameter footprint, GRIT significantly outperformed QLoRA across standard language evaluation metrics. For instance, GRIT achieved a BLEU score of 0.1233, compared to 0.0154 for QLoRA. On summarization metrics, GRIT recorded Rouge-1, Rouge-2, and Rouge-L scores of 0.4355, 0.224, and 0.3632, respectively, whereas QLoRA attained 0.1667, 0.0723, and 0.1271. In terms of semantic similarity, GRIT delivered superior BERT score results, with F1, Precision, and Recall values of 0.8998, 0.9041, and 0.8963, respectively, outperforming QLoRA's corresponding scores of 0.8341, 0.7963, and 0.8786. These results validate the effectiveness of the GRIT method in reducing computational cost while simultaneously enhancing model quality and adaptation fidelity.

[0058] Referring to FIG. 2, the parameter update comparison 200 demonstrates the contrasting approaches between LoRA and GRIT across 28 transformer layers, represented as L0 – L27, through a visual representation where each pixel corresponds to a specific parameter position within the model architecture. The visualization presents two horizontal rows of small squares, with the upper row 202 displaying LoRA updates and the lower row 204 showing GRIT updates. Each black or darkened pixel indicates a position where the respective low-rank adaptation method applies a non-zero parameter update, while white pixels represent positions that remain inactive or unmodified during the fine-tuning process. The stark visual contrast between the two rows reveals fundamental differences in how each method distributes parameter modifications across the transformer layers.

[0059] The LoRA approach in the upper row 202 exhibits dense parameter updating patterns characterized by widespread activation across multiple layers, where the darkened pixels appear consistently distributed throughout the visualization with relatively uniform coverage. The dense updating strategy of LoRA applies modifications broadly across the

parameter space without selective filtering or geometric considerations, resulting in a comprehensive but potentially inefficient utilization of adaptation capacity. The uniform distribution of updates across layers reflects the traditional low-rank adaptation approach that treats all parameter positions with equal consideration regardless of their functional importance or sensitivity to task-specific modifications.

[0060] In contrast, GRIT approach in the lower row 204 demonstrates a selective parameter updating approach that produces structured sparsity patterns visible through the concentrated distribution of darkened pixels. The GRIT visualization shows significantly fewer active parameter positions compared to LoRA, with updates concentrated in specific regions that correspond to geometrically informed adaptation decisions. The selective updating mechanism results from the neural reprojection process that identifies and retains only the parameter update components aligned with principal eigenvector directions, thereby eliminating modifications that contribute minimally to optimization objectives. The structured sparsity pattern reflects the framework's ability to focus adaptation capacity on parameter positions where modifications produce substantial impact on model performance.

[0061] The parameter efficiency achieved by GRIT shown in the lower row 204 becomes apparent through the quantitative reduction in active parameter positions, where the framework accomplishes a 43.90% reduction in trainable parameters compared to LoRA while maintaining task performance. The sparse updating pattern shown in FIG. 2 provides visual evidence of this parameter reduction, where the decreased density of white pixels compared to darkened pixels corresponds directly to the measured efficiency gains. The selective nature of GRIT's parameter updates enables the framework to achieve comparable or superior performance with substantially fewer parameter modifications, demonstrating the effectiveness of geometry-aware adaptation strategies in optimizing resource utilization during fine-tuning operations.

[0062] The GRIT framework demonstrates broad practical applicability and seamless integration capabilities across diverse computational environments and model architectures. The framework may support mixed precision arithmetic and distributed data parallelism to optimize computational efficiency, while maintaining compatibility with existing training infrastructure. The framework may integrate into widely used machine learning frameworks with minimal modification to training scripts. The present invention provides memory usage optimization which may be achieved by permitting the offloading of intermediate computations, such as covariance matrices and decompositions, to non-accelerated devices

such as host processors or asynchronous compute queues. Efficient recycling of temporary memory buffers occurs, and only essential statistics are retained between training intervals.

[0063] The broad applicability of the GRIT framework extends across multiple model architectures and application domains. The present invention is applicable to a variety of model types, including decoder-only language models such as GPT, encoder-decoder architectures such as T5 and BART, and more specialized transformer variants. It provides support for a wide range of downstream tasks including classification, sequence tagging, machine translation, summarization, question answering, and domain-specific instruction following. In each case, maintenance of the inference structure of the model ensures that deployment latency and throughput remain unaffected. The low overhead of adaptation allows rapid retraining across multiple tasks or domains, making the method well-suited for federated learning, personalization, and continual learning settings.

[0064] The flexible implementation design of the GRIT framework accommodates diverse training configurations and hardware constraints. In terms of training setup, the method accommodates full batch, mini-batch, or online learning regimes. The sampling schedule for statistics capture may be tuned to match the available memory and hardware profile. For example, in low memory environments, configuration to sample every few iterations and maintain a sliding window buffer can be applied. In high throughput environments, continuous updating of curvature estimates may utilize real-time streaming statistics. The method also provides robustness to noisy gradients and outlier updates, as the projection step naturally suppresses unstable directions and amplifies signals with consistent curvature alignment.

[0065] The practical advantages of the GRIT framework become particularly evident in challenging real-world scenarios. The method's reliance on second-order information benefits scenarios where labeled data is scarce, enabling extraction of more signal from each gradient step. This results in improved data efficiency and faster convergence compared to first-order methods. Furthermore, by restricting updates to directions of high curvature, knowledge preservation from the original model and catastrophic forgetting mitigation is achieved. This benefit is of particular importance in instruction tuning or task adaptation settings where multiple objectives must be balanced. The curvature-aligned updates also provide divergence prevention when the model is fine-tuned across tasks with different distributions or levels of complexity.

[0066] In summary, the GRIT framework of the present invention provides a comprehensive solution that addresses fundamental limitations in existing fine-tuning approaches while delivering substantial practical advantages. The GRIT framework of the present invention offers several advantages over traditional fine-tuning methods, including improved parameter efficiency, enhanced convergence stability, and reduced computational overhead. The framework effectively addresses limitations of conventional parameter-efficient fine-tuning approaches by incorporating curvature-aware optimization and dynamic rank compression mechanisms that focus adaptation efforts on parameter directions with the greatest impact on model performance. By leveraging Kronecker-factored approximations of second-order curvature information and eigendecomposition-based reprojection techniques, GRIT achieves superior performance while maintaining seamless compatibility with existing machine learning infrastructure. The framework operates within standard training pipelines and supports mixed precision arithmetic, distributed data parallelism, and gradient accumulation strategies without requiring modifications to established development workflows. The geometric optimization approach enables more efficient exploration of the parameter space compared to uniform adaptation strategies employed by conventional methods, while the spectral filtering mechanism provides implicit regularization that enhances generalization performance across diverse evaluation scenarios.

[0067] A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the disclosure. Accordingly, other implementations are within the scope of the following claims.

CLAIMS

1. A method for fine tuning a pre-trained language model using Geometric Reprojection Instruction Tuning (GRIT), comprising:

augmenting the pre-trained language model with one or more low-rank adaptation modules inserted into selected transformer layers, wherein each low-rank adaptation module comprises a pair of trainable parameters;

selectively capturing input activations and output gradients from the low-rank adaptation modules during training according to a configurable sampling schedule to construct statistical representations;

computing low-dimensional approximations of second-order statistics from the captured activations and gradients by decomposing the second-order statistics into manageable statistical covariance matrices;

preconditioning gradient updates applied to the low-rank adaptation modules using the computed low-dimensional approximations to align parameter updates with geometric properties of a loss surface;

periodically reprojecting the low-rank adaptation modules into a compressed subspace based on statistical energy content while retaining components with highest impact on the model performance based on statistical significance.

2. The method as claimed in claim 1, wherein

the low-rank adaptation modules comprise a pair of learnable matrices to modify a portion of the model weights without altering the base parameters,

the paired matrix structure enables task-specific adaptations through rank-constrained modifications while preserving frozen base parameters of the pre-trained language model, and

the pair of trainable parameters comprise down-projection and up-projection matrices.

3. The method as claimed in claim 1, wherein the second-order statistics comprises Kronecker-factored approximations of the Fisher Information Matrix wherein the Kronecker-factored approximations are substantially smaller than a full Fisher Information Matrix, and the statistical covariance matrices comprise a first covariance matrix captures variation in the input activations and a second covariance matrix captures variation in the output gradients.

4. The method as claimed in claim 1, wherein the reprojection into a compressed subspace is performed using a spectral analysis including an eigendecomposition wherein the eigendecomposition transforms the second-order curvature information into a diagonal representation where spectral values including eigenvalues appear along a diagonal and principal directions including eigenvectors form a transformation basis, wherein largest eigenvalues correspond to directions in parameter space where a loss function exhibits high sensitivity to parameter changes.

5. The method as claimed in claim 1, comprising dynamically determining the frequency of preconditioning and reprojection based on observed training behavior, wherein a dynamic scheduling controller evaluates real-time training indicators including gradient norm variance, change in loss value, and update consistency to determine update requirement of the second-order curvature information.

6. The method as claimed in claim 1, wherein the method supports mixed precision training and operates within memory limited hardware environments, and wherein the method supports distributed data parallelism while maintaining synchronization of parameter updates and statistical computations.

7. The method as claimed in claim 1, wherein the low-rank adaptation modules are inserted into the selected transformer layers selected from attention, projection, or feedforward layers, and wherein insertion points are determined through a selection criteria based on layer characteristics including layers receiving higher-capacity modules allocated to more sensitive layers.

8. The method as claimed in claim 1, wherein preconditioning the gradients reduces convergence time and improves training stability, wherein preconditioning adjusts scale and orientation of gradient vectors to align with natural geometry of the loss surface, resulting in more stable updates that converge more rapidly in poorly scaled regions of the loss surface.

9. The method as claimed in claim 1, wherein the reprojection reduces rank by discarding components below a threshold based on the spectral energy, wherein the threshold is determined by summing eigenvalues in descending order until an accumulated sum reaches a

predetermined percentage of total spectral energy, and wherein the spectral filtering process eliminates components associated with flatter regions of the loss surface.

10. The method as claimed in claim 1, comprising:

offloading intermediate statistical computations to a non-accelerated device to minimize memory overhead, and

dynamically allocating memory buffers for activation capture and gradient storage based on available memory resources, and recycling temporary memory buffers across training iterations.

ABSTRACT

The present invention provides a method comprising a Geometric Reprojection Instruction Tuning (GRIT) framework for efficient fine-tuning of pre-trained language models. The method comprises inserting low-rank adaptation modules into selected transformer layers, each module having trainable matrix pairs that approximate low-rank updates while keeping base parameters frozen. Further, the method involves selectively capturing input activations and output gradients, computing Kronecker-factored approximations of Fisher Information Matrix from captured data, preconditioning gradient updates using the approximations to align with loss surface geometry, and periodically performing eigendecomposition to identify principal eigenvectors representing directions of greatest parameter sensitivity. Parameter changes are reprojected onto subspaces spanned by principal eigenvectors to compress effective update space while preserving high-impact components. The method provides a reduction in trainable parameters compared to existing methods and solves computationally expensive traditional fine-tuning through parameter reduction with enhanced convergence stability and improved model performance.

FIG. 1

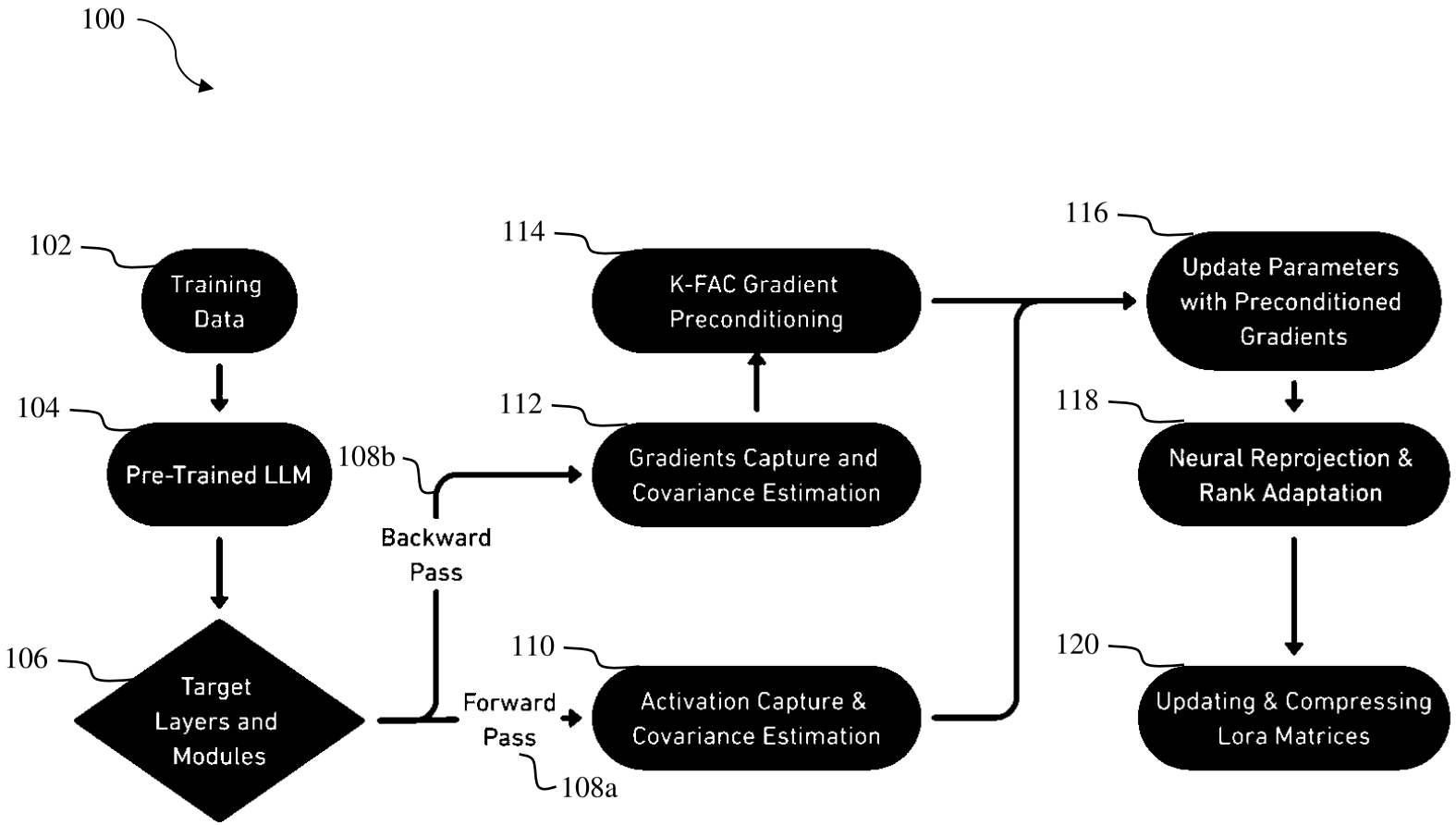
100
102
Training Data
104
Pre-Trained LLM
106
Target Layers and Modules
108b
Backward Pass
Forward Pass
108a
110
Activation Capture & Covariance Estimation
112
Gradients Capture and Covariance Estimation
114
K-FAC Gradient Preconditioning
116
Update Parameters with Preconditioned Gradients
118
Neural Reprojection & Rank Adaptation
120
Updating & Compressing Lora Matrices

FIG. 1


200
202
Parameter Efficiency Comparison: LoRA vs GRIT (Direct Updates)
LoRA: 100.00% | GRIT: 56.14% | GRIT is 43.9% more efficient
LoRA
L0 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19 L20 L21 L22 L23 L24 L25 L26 L27
204
GRIT
L0 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19 L20 L21 L22 L23 L24 L25 L26 L27

FIG. 2