Deepseek - 会話

A: 私は DeepSeek-V3 の技術報告書を読んでいて、このモデルのスケールに本当に感動しました。6710 億のパラメータですが、トークンごとに 370 億しか活性化されないのですか？それは巨大な MoE アーキテクチャですね。どうやって動くのでしょうか？

B: そうですね、それはすごい仕事です！DeepSeek-V3 は、Mixture-of-Experts（MoE）フレームワークに基づいて構築されており、各トークンごとにパラメータのサブセットのみを活性化することができます。具体的には、256 のルーティングされた専門家を使用していますが、トークンごとに 8 つのみが活性化されます。これにより、全てのパラメータが各トークンに対して活性化される密なモデルに比べて非常に効率的です。

A: それは納得できます。でも、どの専門家を活性化するかはどうやって決めるのでしょうか？それはランダムなのですか、それとも何かルーティングメカニズムがあるのでしょうか？

B: 素晴らしい質問です！ルーティングは、トークンから専門家へのアフィニティスコアに基づいています。各トークンには、各専門家に対するスコアが割り当てられ、最高の K の専門家が活性化されます。DeepSeek-V3 は、これらのスコアを計算するためにシグモイド関数を使用しており、これにより専門家間の負荷をバランスさせることができます。

A: なるほど、それは学習中に学習されるわけですね。でも、それは専門家の使用が不均衡になるのではないでしょうか？MoE モデルでそのような問題が一般的だという話を聞いたことがあります。

B: その通りです！専門家の使用が不均衡になることは問題ですが、DeepSeek-V3 は、負荷をバランスさせるための補助損失なしの戦略を導入しています。別の損失項目を追加して負荷をバランスさせるのではなく、各専門家のバイアス項目を動的に調整します。専門家が過負荷の場合、そのバイアスは減少し、過負荷でない場合は増加します。これにより、モデルのパフォーマンスを損なうことなく負荷をバランスさせることができます。

A: それは巧妙ですね。つまり、補助損失がないため、メインの学習目標に対する干渉が少ないわけですね。でも、これは伝統的な補助損失を使用する MoE モデルと比較してどうでしょうか？

B: そうです。伝統的な MoE モデルは、負荷をバランスさせるために補助損失を使用することが多いですが、これらの損失は時にはパフォーマンスを損なうことがあります。DeepSeek-V3 の補助損失なしのアプローチは、このトレードオフを避けます。実際、アブレーション研究では、補助損失に依存するモデルよりも、特にコーディングや数学のようなタスクで一貫して優れていることが示されています。

A: 面白いですね。コーディングや数学について話すと、DeepSeek-V3 が HumanEval や MATH のようなベンチマークで非常に優れたパフォーマンスを発揮していることに気づきました。その秘密は何でしょうか？

B: その大きな部分は、マルチトークン予測（MTP）目標にあります。DeepSeek-V3 は、各位置で次のトークンを予測するのではなく、複数の将来のトークンを予測します。これにより、トレーニング信号が濃縮され、モデルが予測を行うことができます。これは、コーディングや数学のような順序的な推論が必要なタスクに特に有用です。

A: ちょっと待ってください、つまり一度に複数のトークンを予測しているのですね？それは推論中はどうなっているのでしょうか？推論中も MTP を使用するのでしょうか、それともトレーニングのみですか？

B: 推論中、MTP モジュールは捨てられ、モデルは標準的な自回帰モデルのように動作します。でも、ここが面白い部分です：MTP モジュールは、推測デコードに再利用することができ、これにより複数のトークンを並行して予測し、その後検証することで生成を加速できます。

A: それは素晴らしいですね。つまり、トレーニング中に MTP の利点を得て、推論を加速するために使用するわけですね。でも、アテンションメカニズムはどうでしょうか？Multi-head Latent Attention（MLA）について何か見ましたが、それはどのように関連していますか？

B: MLA はもう一つの重要な革新です。キーと値（KV）キャッシュのメモリフットプリントを圧縮することで、フルアテンションキーと値を保存するのではなく、低ランクの共通圧縮を使用して表現します。これにより、推論中に KV キャッシュのサイズを大幅に削減しつつ、標準的な Multi-Head Attention と比較してパフォーマンスを維持します。

A: それは効率性にとって大きな勝利ですね。でも、圧縮は情報の喪失を引き起こすのではないでしょうか？どうやってパフォーマンスを維持しているのでしょうか？

B: それは良い指摘です。圧縮は、キーと値の重要な情報を保持するように設計されています。モデルは、Rotary Positional Embedding（RoPE）を使用して位置情報を維持し、これにより圧縮されたキーと値でもトークンの相対位置を理解するのを助けます。

A: わかりました。つまり、MLA は効率性を損なうことなくパフォーマンスを犠牲にすることなく効率を高めることができます。でも、トレーニングはどうでしょうか？このような大きなモデルをトレーニングすることは非常にコストがかかるでしょう。DeepSeek-V3 はどのようにコストを抑えるのでしょうか？

B: トレーニングの効率は重要な焦点です。DeepSeek-V3 は、FP8 混合精度フレームワークを使用しており、これによりメモリ使用量を削減し、計算を加速します。また、パイプライン並列化を最小限に抑えるための DualPipe アルゴリズムを使用しています。これらの最適化により、モデルは 2.788 万 H800 GPU 時間で 14.8 兆トークンでトレーニングできます。

A: それは驚きですね。でも、FP8 トレーニングは難しいと聞きます。精度の問題はどう対処しているのでしょうか？低精度トレーニングは不安定になることがあると聞きました。

B: その通りです。FP8 トレーニングは、限られた動的範囲のために難しいですが、DeepSeek-V3 は細かい量子化戦略を使用してこれを軽減しています。例えば、活性化は 1x128 タイルにグループ化され、重みは 128x128 ブロックにグループ化されます。各グループは独立してスケーリングされ、これにより異常値を処理し、トレーニングを安定させることができます。また、重要な操作のために高精度の累積を使用して正確性を維持しています。

A: それは納得できます。つまり、効率と精度のバランスを取っているわけですね。でも、データはどうでしょうか？14.8 兆トークンは巨大なデータセットですね。どのようなデータでトレーニングされているのでしょうか？

B: データセットは多様で高品質であり、特に英語と中国語のテキストに焦点を当てています。また、数学やプログラミングデータも多く含まれており、これによりモデルがこれらの分野で優れたパフォーマンスを発揮することができます。データパイプラインは、冗長性を最小限に抑えつつ多様性を維持するために最適化されており、ドキュメントのパッキングなどの技術を使用してデータの完整性を確保しています。

A: それはコーディングや数学のタスクで強力なパフォーマンスを発揮する理由がわかります。でも、多言語タスクはどうでしょうか？他の言語もうまく扱えるのでしょうか？

B: はい、DeepSeek-V3 は多言語コーパスでトレーニングされており、MMMLU のようなベンチマークでも優れたパフォーマンスを発揮しています。特に中国語で強力であり、Qwen2.5 などのモデルよりも C-Eval や CMMLU のような中国語ベンチマークで優れています。

A: それは驚きですね。でも、長文脈タスクはどうでしょうか？128K トークンまでサポートしていると聞きました。そんな長い入力をどうやって扱っているのでしょうか？

B: DeepSeek-V3 は、まず 32K トークンに、次に 128K トークンにヤーン技術を使用して文脈長を延長しています。これにより、ドキュメントの要約や検索などの長文脈タスクを効果的に扱うことができます。また、長文脈理解を評価する「Needle In A Haystack」テストでも優れたパフォーマンスを発揮しています。

A: それは前のモデルに比べて大きな改善ですね。でも、デプロイメントはどうでしょうか？そんな大きなモデルの推論をどうやって扱っているのでしょうか？

B: 推論は H800 クラスターで処理され、GPU は NVLink と InfiniBand を使用して相互接続されています。デプロイメント戦略は、高いスループットと低レイテンシーを確保するために、前填充とデコードのステージを分離しています。また、推論中の負荷をバランスさせるために冗長な専門家を使用しており、これにより効率を維持しています。

A: それは多くの最適化ですね。でも、限界は何でしょうか？このような大きなモデルにはトレードオフがあるはずです。

B: 1 つの制限は、効率的な推論のために大きなクラスターが必要であることです。これは小さなチームにとっては課題となるかもしれません。生成速度の改善の余地もありますが、MTP による推測デコードが助けになっています。

A: それは納得できます。でも、全体としては大きな進歩ですね。次に DeepSeek-V3 は何を目指しているのでしょうか？将来の方向性は何でしょうか？

B: 彼らはいくつかの分野を探索しています。例えば、無限の文脈長をサポートするためのアーキテクチャの精緻化、追加のトレーニング信号源の探索、モデルの推論能力の向上などです。また、モデルのパフォーマンスをより包括的に評価するための方法も研究しています。

A: それは止まる気配がないですね。これまでの説明、ありがとうございました—DeepSeek-V3 はオープンソース LLM スペースでゲームチェンジャーですね。

B: もちろんです！オープンソースモデルがどれだけ進化したかを見るのは興奮します。DeepSeek-V3 は境界を押し広げており、次に何をするか楽しみです。

A: FP8 混合精度トレーニングを使用しているとおっしゃいましたが、BF16 や FP16 と比較してどうでしょうか？FP8 は本当にこのような大きなモデルのトレーニングに安定しているのでしょうか？

B: それは素晴らしい質問です。FP8 は動的範囲が限られているために難しいですが、DeepSeek-V3 は細かい量子化戦略を使用してこれを軽減しています。例えば、活性化は 1x128 タイルにグループ化され、重みは 128x128 ブロックにグループ化されます。各グループは独立してスケーリングされ、これにより異常値を処理し、トレーニングを安定させることができます。

A: 面白いですね。つまり、単なる FP8 量子化ではなく、より洗練されたものですね。でも、これらのグループとスケーリング係数を管理するためのオーバーヘッドはないのでしょうか？

B: それはありますが、そのオーバーヘッドは利点に比べて非常に小さいです。FP8 はメモリ使用量を削減し、計算を加速するために重要です。また、重要な操作、例えば行列の乗算のために高精度の累積を使用して数値の安定性を確保しています。

A: わかりました。つまり、精度と効率のトレードオフですが、彼らは良いバランスを取っているわけですね。でも、DualPipe アルゴリズムはどうでしょうか？それはどうやって動作するのでしょうか？

B: DualPipe は、パイプライン並列化におけるパイプラインバブルを最小限に抑えるために設計されています。各ワークチャンクを 4 つのコンポーネントに分割しています：アテンション、全体へのディスパッチ、MLP、全体への結合。後方パス中には、さらに「入力のための後方」と「重みのための後方」に計算を分割して、より効率的なオーバーラップを許可しています。

A: それは複雑ですが、納得できます。つまり、計算と通信のオーバーヘッドを隠すことで、他のパイプライン並列化方法、例えば 1F1B や Zero Bubble と比較してどうでしょうか？

B: DualPipe は 1F1B や Zero Bubble よりもパイプラインバブルが少ないです。また、マイクロバッチをパイプラインの両端からフィードする双方向スケジューリングを許可するため、アイドル時間をさらに減少させ、全体の効率を向上させます。実際、DualPipe は全体への通信オーバーヘッドをほぼゼロにすることができ、MoE モデルをスケールアップするために重要です。

A: それは驚きですね。でも、メモリ使用量はどうでしょうか？DualPipe は他の方法よりもメモリを多く必要とするのでしょうか？

B: はい、モデルパラメータの 2 つのコピーを保持するために少し多くのメモリが必要ですが、その増加は管理可能です。メモリフットプリントは、RMSNorm の再計算や MLA のアッププロジェクションなどの技術を使用して最適化されており、中間活性化を保存する必要がなくなります。

A: なるほど、効率を高めるために少しのメモリを取引するわけですね。それは公平な取引に思えます。メモリについて話すと、128K トークンのような長い文脈長を扱うために KV キャッシュはどう扱っているのでしょうか？それは巨大なキャッシュが必要になるはずです。

B: それは MLA が輝くところです。KV キャッシュを圧縮することで、そのサイズを大幅に削減します。フルアテンションキーと値を保存するのではなく、圧縮された潜在ベクトルを保存します。これにより、DeepSeek-V3 はメモリボトルネックにぶつからずに長い文脈を扱うことができます。

A: それは巧妙な解決策ですね。でも、アテンションの質はどうでしょうか？圧縮はモデルが適切なトークンにアテンションを払う能力に影響を与えるのでしょうか？

B: 圧縮は重要な情報を保持するように設計されており、アテンションの質への影響は最小限です。また、RoPE（Rotary Positional Embedding）を使用して位置情報を維持し、圧縮されたキーと値でもトークンの相対位置を理解するのを助けます。

A: それは納得できます。つまり、MLA はメモリ使用量を削減しつつ、パフォーマンスを犠牲にすることなく効率を高めることができます。でも、トレーニングデータはどうでしょうか？14.8 兆トークンだとおっしゃいましたが、そのような巨大なデータセットの質と多様性をどうやって確保しているのでしょうか？

B: データセットは、高品質で多様なトークンを含むように慎重にカリキュレーションされています。データパイプラインを最適化して冗長性を最小限に抑えつつ多様性を維持し、ドキュメントのパッキングなどの技術を使用してデータの完整性を確保しています。コーパスには、英語と中国語のテキストが混在しており、特に数学やプログラミングのサンプルに焦点を当てています。

A: それはコーディングや数学のタスクで強力なパフォーマンスを発揮する理由がわかります。でも、多言語タスクはどうでしょうか？他の言語もうまく扱えるのでしょうか？

B: はい、DeepSeek-V3 は多言語コーパスでトレーニングされており、MMMLU のようなベンチマークでも優れたパフォーマンスを発揮しています。特に中国語で強力であり、Qwen2.5 などのモデルよりも C-Eval や CMMLU のような中国語ベンチマークで優れています。

A: それは驚きですね。でも、長文脈タスクはどうでしょうか？128K トークンまでサポートしていると聞きました。そんな長い入力をどうやって扱っているのでしょうか？

B: DeepSeek-V3 は、まず 32K トークンに、次に 128K トークンにヤーン技術を使用して文脈長を延長しています。これにより、ドキュメントの要約や検索などの長文脈タスクを効果的に扱うことができます。また、長文脈理解を評価する「Needle In A Haystack」テストでも優れたパフォーマンスを発揮しています。

A: それは前のモデルに比べて大きな改善ですね。でも、デプロイメントはどうでしょうか？そんな大きなモデルの推論をどうやって扱っているのでしょうか？

B: 推論は H800 クラスターで処理され、GPU は NVLink と InfiniBand を使用して相互接続されています。デプロイメント戦略は、高いスループットと低レイテンシーを確保するために、前填充とデコードのステージを分離しています。また、推論中の負荷をバランスさせるために冗長な専門家を使用しており、これにより効率を維持しています。

A: それは多くの最適化ですね。でも、限界は何でしょうか？このような大きなモデルにはトレードオフがあるはずです。

B: 1 つの制限は、効率的な推論のために大きなクラスターが必要であることです。これは小さなチームにとっては課題となるかもしれません。生成速度の改善の余地もありますが、MTP による推測デコードが助けになっています。

A: それは納得できます。でも、全体としては大きな進歩ですね。次に DeepSeek-V3 は何を目指しているのでしょうか？将来の方向性は何でしょうか？

B: 彼らはいくつかの分野を探索しています。例えば、無限の文脈長をサポートするためのアーキテクチャの精緻化、追加のトレーニング信号源の探索、モデルの推論能力の向上などです。また、モデルのパフォーマンスをより包括的に評価するための方法も研究しています。

A: それは止まる気配がないですね。これまでの説明、ありがとうございました—DeepSeek-V3 はオープンソース LLM スペースでゲームチェンジャーですね。

B: もちろんです！オープンソースモデルがどれだけ進化したかを見るのは興奮します。DeepSeek-V3 は境界を押し広げており、次に何をするか楽しみです。