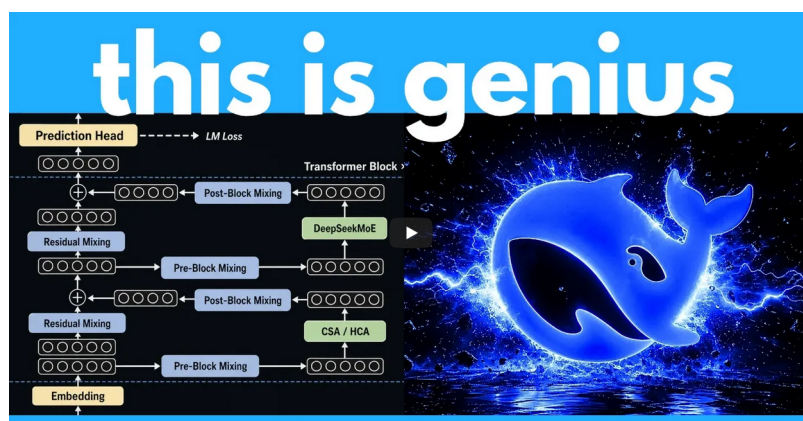


Inside DeepSeek V4: Hoe een team met beperkte middelen een AI-gigant met een miljoen tokens ontwikkelde

Door Frank , 21 mei 2026

Inside DeepSeek V4: Hoe een team met beperkte middelen een AI-gigant met een miljoen tokens ontwikkelde

In een AI-landschap dat wordt gedomineerd door datacenters van miljarden dollars en onbeperkte rekenkracht, heeft een relatief klein team van DeepSeek zojuist een bom laten vallen: DeepSeek V4 Pro, een Mixture-of-Experts (MoE)-model met 1,6 biljoen parameters dat rechtstreeks de concurrentie aangaat met de closed-source giganten uit de industrie. Wat dit verhaal opmerkelijk maakt, is niet alleen de prestatie, maar ook de beperkingen. Naar verluidt werkt DeepSeek met een team dat ongeveer 40 keer kleiner is dan dat van Open AI en heeft het geen toegang tot de meest geavanceerde NVIDIA-chips. Toch hebben ze een model gebouwd met een contextvenster van 1 miljoen tokens dat rivaliseert met Claude Opus 4.6 en GPT-5.4.



Het belangrijkste is dat ze het hebben geopen-sourced onder de MIT-licentie en een technisch paper hebben gepubliceerd met details over hun "krankzinnige engineering".

Hier is de uitleg over hoe ze beperkingen hebben omgezet in een meesterklas in efficiëntie.

De Kernuitdaging: Waarom een Geheugen van een Miljoen Tokens Meestal Breekt. Voordat we de oplossing bespreken, moeten we het probleem begrijpen. De meeste LLM's vertrouwen op aandacht. Wanneer het model een woord (token) leest, vergelijkt het dat met elk woord dat eraan voorafging. Voor een miljoen tokens is dat een miljoen vergelijkingen per token.

Dit creëert twee enorme knelpunten:

1. Rekenexplosie: De wiskunde groeit kwadratisch.
2. Geheugenbloot (KV-Cache): Het model slaat een "opzoektabel" van elk vorig woord op. Bij 1 miljoen tokens vult dit gigabytes aan duur GPU-geheugen.

De conventionele wijsheid zegt dat je brute kracht nodig hebt om dit op te lossen. DeepSeek stelde een slimmere vraag: wat als het model niet naar alles hoeft te kijken?

De Architectuur: Hybride Aandacht (CSA + HCA)

DeepSeek V4 leest tekst niet lineair. Het scant, comprimeert en roept op verzoek op. Het team introduceerde een Hybride Aandachtsarchitectuur die twee verschillende strategieën combineert om het geheugenprobleem op te lossen.

1. De Indexeerder (Gecomprimeerde Sparse Aandacht)

In plaats van elk woord afzonderlijk te onthouden, groepeert Gecomprimeerde Sparse Aandacht (CSA) tokens in kleine clusters (bijv. groepen van 4). Het voegt ze samen in een enkele "samenvattings"-invoer.

- De "Bliksemindexeerder": Wanneer het model informatie nodig heeft, scant het niet de geschiedenis. Het voert een snelle zoekopdracht uit over deze samenvattingen om de top 3-4 meest relevante blokken te vinden.
- Resultaat: Het model negeert 99% van de tekst en richt zich alleen op de relevante 1%.

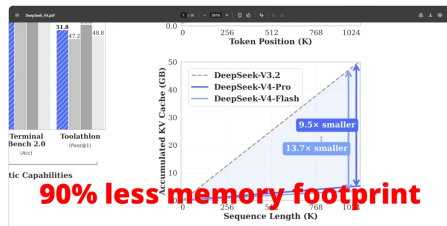
2. Het Grote Geheel (Sterk Gecomprimeerde Aandacht)

Soms heb je geen specifiek detail nodig, maar de essentie van een eerder hoofdstuk. Sterk Gecomprimeerde Aandacht (HCA) gaat nog verder door hele alinea's (128+ tokens) te comprimeren tot een enkele wiskundige vector.

- Resultaat: De sequentiellengte krimpt zoveel dat het model zich kan veroorloven om alles in één keer te bekijken, zodat het de rode draad niet verliest.

Maar wat als je een exact citaat nodig hebt? Als je te veel comprimeert, verlies je precisie. Om dit op te lossen, behield DeepSeek een derde pad: Schuifvenster Aandacht. Dit behoudt de laatste ~4.000 woorden met een perfecte, niet-

gecomprimeerde getrouwheid.



De Efficiëntieopbrengst:

De cijfers zijn verbijsterend. Volgens benchmarks, vergeleken met DeepSeek V3.2:

- Rekenkracht (FLOPs): Verminderd met 73% (draait op 27% van de rekenkracht).
- Geheugen (KV-Cache): Verminderd met 90%.

Het Temmen van het Beest: Signaalstabiliteit bij 1,6 Biljoen Parameters

Bij een biljoen parameters mislukt de training meestal. Signalen versterken zich als een microfoon die piept naast een luidspreker - waarden exploderen en het model crasht.

Traditionele AI gebruikt "residuele verbindingen" (bypassbanen) om lagen over te slaan. DeepSeek had meer nodig. Ze introduceerden Manifold-Constrained Hyper-Connections (mHC).

De Analogie: Stel je water voor dat door leidingen stroomt. Normaal gesproken bouwt de druk zich op en barsten de leidingen. mHC dwingt het water een wiskundige regel te volgen waarbij de totale druk altijd behouden blijft. Het signaal kan niet opblazen omdat de wiskunde het verbiedt. Dit vereiste een aangepast algoritme (Sinkhorn) om deze regels af te dwingen, maar DeepSeek optimaliseerde het zodat het slechts 6,7% van de rekentijd kost.

De "Muon"-verschuiving: Een Nieuwe Optimizer

Jarenlang was AdamW de industriestandaard voor het trainen van modellen. DeepSeek gooide het eruit. Ze gebruikten een aangepaste optimizer genaamd Muon.

Zie het als het stemmen van een gitaar:

- AdamW maakt voortdurend kleine, voorzichtige aanpassingen.
- Muon maakt eerst grote, ruwe aanpassingen (om dicht bij de juiste toonhoogte te komen) en schakelt dan over op kleine, precieze aanpassingen.

Dit stelt DeepSeek in staat sneller en stabielere te trainen op hun enorme dataset van 33 biljoen tokens.

Datacenter Choreografie: Het Verbergen van de Latentie

Een model van 1,6 biljoen past niet op één chip. Het is verspreid over rekken. Als Rek A wacht op gegevens van Rek B, staat de GPU stil - geldverspilling. DeepSeek loste dit op met overlappende communicatie. Ze braken de gegevens op in golven. Terwijl de GPU aan Golf 1 werkt, verzendt het netwerk al Golf 2 op de achtergrond. Met behulp van een laag-niveautaal genaamd Triton en geavanceerde "gefuseerde kernels", bewezen ze wiskundig dat hun code perfect was met behulp van een Z3-solver. Het resultaat? De netwerklatenzie "verdwijnt".

Aanbevelingen voor Ontwikkelaars

Op basis van de geverifieerde mogelijkheden van DeepSeek V4, volgen hier mijn aanbevelingen voor integratie in uw stack:

1. Maak gebruik van "Denk"-modi

DeepSeek V4 biedt configureerbare redeneerdiepte. Gebruik "Non-Think" voor chat en samenvattingen. Schakel over naar "Max/Think High" voor codering, wiskunde of complexe agentische taken. Het model gebruikt meer uitvoertokens in de Max-modus, maar het is nog steeds 7x goedkoper dan Claude Opus 4.7.

2. Gebruik het Contextvenster voor Agents

Het venster van 1 miljoen tokens is ideaal voor langlopende agents. U kunt er een volledige codebase of uren aan gespreksgeschiedenis in voeden. De MRCR 1M-benchmark toont een ophaalnauwkeurigheid van meer dan 83%, wat betekent dat het zelden "vergeet" welke instructies aan het begin van een gesprek zijn gegeven.

3. Let op Hallucinaties (Het "Weet Wat Het Weet"-Probleem)

Dit is cruciaal. Onafhankelijke analyses tonen aan dat DeepSeek V4 een hallucinatiegraad van 94% heeft bij taken met onbekende antwoorden. Als u een vraag stelt buiten zijn kennisdatabase, zal het waarschijnlijk een antwoord verzinnen in plaats van "weet ik niet" te zeggen. Aanbeveling: Veranker het model altijd met Retrieval-Augmented Generation (RAG). Geef de brontekst mee in de prompt. Vertrouw niet op zijn interne parameters voor obscure feiten waar het ~18% achterblijft bij Gemini 3.1.

Oordeel: Een Paradigmaverschuiving in Efficiëntie

DeepSeek V4 bewijst dat de toekomst van AI niet alleen draait om wie het grootste cluster heeft, maar om wie de slimste code schrijft. Door zich te richten op sparse verdeling en compressie, hebben ze de toegang tot geavanceerde intelligentie gedemocratiseerd. Voor de open-source gemeenschap is dit een goldrush. Het paper onthult geheimen over infrastructuur en trainingsstabiliteit die gesloten labs meestal achter slot en grendel houden. U kunt de gewichten vandaag nog downloaden van Hugging Face en lokaal draaien. DeepSeek versloeg de reuzen niet met meer geld. Ze versloegen ze met meer wiskunde.

<https://youtu.be/XJUpuOBpT-4>