Deep Dive no Coração do DeepSeek-R1-Zero

Redefinindo os limites do Reinforcement Learning
By: Samuel Fernando

Esse é o primeiro conteúdo técnico em PT-BR que trata do DeepSeek, um modelo chinês que chocou o mundo na última semana. Não custa repetir o que aconteceu. O anúncio da startup chinesa varreu US$ 1 trilhão de ações nos EUA em tão pouco tempo e desencadeou a maior queda da Nvidia na história do mercado de ações dos EUA (US$ 600 bi). Estamos diante de uma disrupção a partir de outra disrupção que é a IA Generativa. O DeepSeek-R1 (600B parâmetros) foi treinado em 2,8M horas de GPU usando 2048 GPUs por 2 meses, custando cerca de US$ 6 milhões. Para efeito de comparação, o Llama 3.1 (405B) da Meta foi treinado em 30,8M horas de GPU e custou cerca de US$ 720 milhões, o que significa que o DeepSeek usou ~11x menos computação e ~120x menos de grana, ao mesmo tempo em que entregou um desempenho muito melhor. Além disso, o DeepSeek foi treinado em GPUs H800, que são um pouco mais lentas do que as H100s da Meta. Vamos dissecar o coração do modelo e entender seu principal diferencial nesse artigo.

1. Introdução aos Modelos DeepSeek R1

Os modelos da família DeepSeek R1 ganharam destaque na comunidade internacional de Inteligência Artificial por apresentarem habilidades avançadas em tarefas complexas de raciocínio, como resolução de problemas matemáticos e geração de código. Seu impacto no cenário de LLMs (Large Language Models) é comparável à revolução observada anos atrás com modelos da série GPT, culminando agora em modelos como GPT-4o e GPT-o1, lançados recentemente em 2024, que também seguem abordagens híbridas de treinamento.

1.1 Histórico

AlphaGo Zero (2017) mostrou que o self-play e a aprendizagem por reforço puro podiam superar abordagens treinadas com conhecimento humano prévio.

Transformers (a partir de 2017) trouxeram a arquitetura que domina hoje o treinamento de LLMs juntamente com treinamento de RLHF (um tipo de Aprendizado por Reforço focado apenas no alinhamento da saída do modelo)

DeepSeek-V3-Base (2024) serviu como fundação para experiências com RL mais intensivas, por já conter um vasto conhecimento linguístico e factual.

O surgimento de DeepSeek-R1-Zero (início de 2025) demonstrou que era possível — e altamente eficiente — desenvolver capacidades de raciocínio complexo quase sem suporte inicial de fine-tuning supervisionado.
Poderíamos a arquitetura de treinamento do DeepSeek-R1 ilustrado na imagem abaixo com a descrição logo a seguir.

1. DeepSeek-V3-Base
- Modelo inicial: é a base pré-treinada sobre a qual são aplicadas as demais técnicas.

2. DeepSeek-R1-Zero
- RL “Cold Start”: treinamento puramente por Reforço (GRPO), partindo do modelo base, usando dados iniciais (“coldstart data”) e special tokens.

- Produz exemplos de cadeias de raciocínio úteis, sem passar por supervisão direta prévia.

3. Coldstart Data
- Conjunto de dados inicial: fornece problemas ou prompts para iniciar o aprendizado por Reforço sem um “chute” supervisionado.

4. pre-DeepSeek-R1
- Versão intermediária: resultante do ajuste (fine-tuning) do modelo base, para já incorporar uma supervisão inicial ou uma etapa extra de refinamento depois do R1-Zero.

5. Large Scale, High Quality Logical Data & Curated High Quality General Data
- Dados de lógica/matemática em larga escala e dados gerais de alta qualidade cuidadosamente selecionados.

- Garantem que o modelo refine habilidades de raciocínio e linguagem.

6. Rejection Sampling
- Filtro de qualidade: descarta respostas ruins do modelo intermediário e retém as melhores, para criar um conjunto de dados aprimorado para o próximo passo.

7. FineTuned pre-DeepSeek-R1
- Modelo pré-ajustado: recebe uma nova rodada de fine-tuning com dados coletados e melhorados (inclusive os filtrados por rejection sampling).

8. Second Pass Fine Tuning Data
- Nova massa de dados: inclui exemplos mais complexos ou variados, resultando num refinamento adicional das capacidades do modelo.

9. GRPO Reinforcement Learning with Human Preference Rewards
- Fase final de RL: o modelo recebe recompensas baseadas em preferências humanas (por exemplo, feedback sobre clareza, coerência, estilo), alinhando-o ainda mais com objetivos de qualidade.

10. DeepSeek-R1
- Modelo final: já combina o conhecimento do treinamento “cold start”, o refinamento de fine-tuning supervisionado e a otimização via RL (GRPO) com recompensas humanas, resultando numa versão mais robusta, coerente e confiável.

2. DeepSeek-R1-Zero: O Paradigma de “Cold Start”

O aspecto mais marcante do DeepSeek-R1-Zero é ter sido treinado inteiramente por Reinforcement Learning a partir do modelo base (DeepSeek-V3-Base) sem usar Supervised Fine-Tuning (SFT) no começo. Em outras palavras, o modelo não recebeu previamente exemplos “corretos” de como responder a perguntas ou resolver problemas; aprendeu somente por meio de um mecanismo de tentativa e erro guiado por recompensas a partir de seu modelo fundacional.

Cold Start: O Que É e Como Funciona? O termo "Cold Start" refere-se a um problema comum em sistemas de aprendizado de máquina e inteligência artificial, onde um modelo ou sistema precisa operar sem informações ou experiência prévia, muito comum em Sistemas de Recomendação. Em outras palavras, ele começa do zero, sem um histórico ou exemplos anteriores que possam guiá-lo no início.

2.1 Principais Surpresas 😲
Emergência de Raciocínio Estruturado

Apesar da ausência de dados rotulados, o modelo aprendeu a estruturar suas respostas, chegando a usar (mesmo que de forma imperfeita) etapas lógicas complexas.

Em tarefas como AIME (American Invitational Mathematics Examination), o modelo gerou soluções explicadas passo a passo, algo inesperado sem SFT.

Para cada pergunta, são geradas 16 respostas amostradas, e a precisão média geral é calculada para garantir uma avaliação estável.
Para cada pergunta, são geradas 16 respostas amostradas, e a precisão média geral é calculada para garantir uma avaliação estável.

Observações:

O DeepSeek-R1-Zero apresentou desempenho superior ao OpenAI-o1-mini em quase todos os benchmarks, especialmente em MATH-500 (86.7%) e GPQA (95.9%).

A pontuação de Bench Rating do DeepSeek-R1-Zero foi inferior (1444) em comparação com os modelos OpenAI (acima de 1800), indicando que ainda há espaço para melhorias na avaliação geral de desempenho.

🔍 Por que isso é importante?
O fato de o modelo desenvolver espontaneamente cadeias de raciocínio estruturadas sem supervisão demonstra que o aprendizado por reforço pode induzir formas sofisticadas de pensamento algorítmico, mesmo sem exemplos humanos. Isso sugere que processos como raciocínio simbólico e heurísticas matemáticas podem emergir naturalmente apenas por meio de tentativa e erro, reforçando a ideia de que grandes modelos podem desenvolver habilidades de descoberta sem serem explicitamente ensinados. Esse fenômeno abre caminho para modelos mais generalistas e adaptáveis, reduzindo a necessidade de

grandes conjuntos de dados supervisionados e permitindo maior autonomia em tarefas cognitivas complexas.

AHA Moments e Backtracking

Observou-se que em certo ponto do treinamento, o modelo passou a reavaliar seu próprio raciocínio, corrigindo erros de maneira autônoma, e mais do que isso. Um dos aspectos mais intrigantes observados durante o treinamento foi o surgimento espontâneo dos chamados "momentos AHA". Para cada questão de raciocínio, o modelo inicialmente segue um processo de Chain-of-Thought (CoT), estruturando um caminho lógico até a solução. No entanto, em um determinado ponto do treinamento, algo surpreendente aconteceu: o modelo percebeu, por conta própria, que sua linha de raciocínio inicial estava errada.

Em vez de simplesmente seguir adiante com uma resposta incorreta, ele voltou atrás, revisou sua abordagem e corrigiu o erro. Esse comportamento nunca foi explicitamente programado e não é exibido no treinamento de nenhum outro modelo, o que significa que não se trata apenas do algoritmo de Reinforcement Learning (RL) ajustando probabilidades de saída. O modelo reavaliou seu próprio processo de raciocínio de maneira autônoma, 100% autônoma! 🤯

Isso foi impressionante. Confesso que ainda estou tentando entender completamente como essa capacidade emergiu sem qualquer dado supervisionado de fine-tuning. Como ele conseguiu desenvolver esse nível de autoavaliação e ajuste apenas a partir de um modelo base? A resposta parece estar na importância de um modelo base extremamente forte, que fornece as fundações necessárias para que esses padrões sofisticados de aprendizado e raciocínio possam emergir naturalmente. Vejamos além disso outro Aha Moment incrível.

Failed to render LaTeX expression — no expression found

O gráfico acima motra a evolução do número médio de tokens (ou comprimento médio) das respostas geradas pelo modelo DeepSeek-R1-Zero ao longo das etapas de treinamento. No eixo X temos os steps — isto é, o avanço progressivo do treinamento por Reinforcement Learning (cada step corresponde a uma iteração ou lote de aprendizado). No eixo Y, vemos a média de tokens por resposta que o modelo produz.

Observe claramente que, no início do treinamento, o modelo gera respostas relativamente curtas, com menos de 1000 tokens. Conforme o treinamento avança (a partir de 1000 a 2000 steps), a curva começa a subir mais acentuadamente, chegando a ultrapassar 4000 tokens de média por volta de 3000 steps. A partir de 4000 a 6000 steps, vemos flutuações mais fortes, mas com uma tendência de crescimento constante — chegando facilmente a 8000 ou até 10.000 tokens de média próximo dos 8000 steps. Esse é o instante em que o DeepSeek-R1-Zero PERCEBE, literalmente percebe, por meio de tentativas, recompensas e revisões internas, sem estar pré-programado para isso, que vale a pena elaborar cadeias de raciocínio mais longas para obter resultados mais precisos.

O que aconteceu foi uma auto-evolução. O DeepSeek-R1 aprendeu a alocar mais tempo de reflexão a um problema, reavaliando sua abordagem inicial. Conforme o modelo evoluiu, ele

passou a “investir” mais tokens em seu raciocínio, desenvolvendo passos de pensamento (Chain-of-Thought) cada vez mais detalhados. Esse fenômeno, o “AHA moment” — não foi codificado explicitamente, mas emergiu graças à natureza iterativa do RL.

Influência do Modelo Base

O DeepSeek-V3-Base foi treinado massivamente em 2023-2024, garantindo ao modelo recursos linguísticos e conhecimentos gerais robustos.

Essa bagagem se mostrou crucial para que o RL “encontrasse” caminhos de solução em espaços de busca enormes.

3. Entendendo os Fundamentos do GRPO
Para viabilizar o treinamento por RL em larga escala, a equipe responsável pelo DeepSeek R1 aplicou uma variação do PPO (Proximal Policy Optimization), chamada GRPO. Revisemos brevemente o PPO antes de detalhar as adaptações:

3.1 Revisão Rápida do PPO
Em aprendizado por reforço, queremos que um agente (como um modelo de IA) aprenda a tomar decisões ótimas para maximizar uma recompensa ao longo do tempo.

Se o modelo mudar muito rápido, ele pode esquecer o que aprendeu antes (esquecimento catastrófico).

Se mudar muito devagar, o aprendizado será ineficiente.

Precisamos de um meio-termo: mudanças suficientes para aprender, mas sem desestabilizar o modelo.

O PPO resolve esse problema ao permitir ajustes controlados na política de aprendizado. A versão clássica utiliza quatro componentes principais:

Modelo Base (π_old): referência da política anterior.

Modelo de Política (π_θ): responsável por gerar as novas ações (tokens/texto).

Modelo de Valor (V_θ): avalia o retorno esperado daquele estado.

Modelo de Recompensa: avalia a qualidade ou adequação da saída.

A otimização busca maximizar o retorno esperado enquanto mantém r_t(θ), a razão entre a nova política e a antiga, em um intervalo controlado (clipping). A função de perda típica do PPO pode ser resumida como:

$$ L^{\text{PPO}}(\theta) = \mathbb{E}_t \Big[\min \big(r_t(\theta) \, A_t, \,\text{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big)\, A_t \big) \Big] $$
Onde:

$$ r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\text{old}}(a_t \mid s_t)} $$
➡️Advantage Function (A^t)

A função de Advantage mede o quanto uma ação ata_t (resposta gerada) foi melhor ou pior que a expectativa média para o estado sts_t, faz isso da seguinte forma:

At=Rt−V(st)A_t = R_t - V(s_t):

RtR_t: Retorno obtido a partir da resposta gerada. A recompensa pode ser determinada por um modelo de recompensa ou via avaliação determinística (ex.: um interpretador Python validando um código gerado).

V(st)V(s_t): Função de Valor, que estima o retorno esperado a partir do estado sts_t. Essa função é ajustada durante o treinamento para minimizar o erro entre a previsão do modelo e o retorno real.

Regularização via Divergência KL (Kullback-Leibler)

Para evitar que a nova política do modelo se desvie excessivamente da política pré-treinada, é aplicada uma penalidade baseada na divergência KL:

$$ R(s)=RM\phi(s)-\beta\log p\theta RL(s)pPT(s)R(s) = RM_{\phi}(s) - \beta \log \frac{p_{\theta}^{RL}(s)}{p^{PT}(s)} $$
Interpretação:

KL penaliza desvios muito grandes da política pré-treinada. Se p_theta^RL(s) (probabilidade da nova política) se distancia muito de p^PT(s) (política pré-treinada), um custo é aplicado para manter a estabilidade:

R(s) = RM_phi(s) - β log (p_theta^RL(s) / p^PT(s)). Isso impede que a nova política do modelo mude drasticamente em relação à política anterior, garantindo um treinamento mais estável.

Função de Perda do Modelo de Valor

A função de valor é otimizada para minimizar o erro de previsão do retorno esperado:

$$ LV(\theta)=Et[(V\theta(st)-Rt)2]L^{V}(\theta) = \mathbb{E}_t \Big[(V_{\theta}(s_t) - R_t)^2 \Big] $$
Esse erro é minimizado via média dos quadrados das diferenças entre a previsão do valor e o retorno real.

Não entendi nada dessas equações😢 Calma. Vamos resumir tudo!

🔍 O truque do PPO
Clipping – Impede que a razão de políticas varie demais, mantendo as atualizações dentro de um limite aceitável e garantindo estabilidade no treinamento.

Minimização – A função de perda usa um operador mínimo, que previne que a perda aumente excessivamente quando a política muda drasticamente. Isso ajuda na convergência estável do aprendizado.

Temos o seguinte:

O agente toma ações baseadas na política atual.

Recebe uma recompensa e avalia se a ação foi melhor ou pior do que o esperado (advantage function).

Ajusta a política, mas com clipping, garantindo que as mudanças não sejam muito bruscas.

Usa divergência KL para evitar que a política treinada se afaste muito da política original.

Esse processo garante que o modelo aprenda de forma eficiente, evitando oscilações bruscas e garantindo que suas novas políticas não se desviem demasiadamente da política original, o que torna o aprendizado mais confiável e robusto. Se a atualização da política for pequena, o modelo segue o ajuste normal. Se for grande demais, o PPO a corta (clip) e evita que o modelo mude muito rápido.

3.2 Modificações do GRPO
Para contornar os altos custos computacionais de manter um modelo de valor em GPUs, o GRPO remove essa componente, estimando a vantagem através de amostras do modelo antigo. Em termos simplificados:

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E} \Big[q \sim P(Q), o \sim \pi_{\theta_{old}} (O|q) \Big] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta} (o_t | q, o_{<t})}{\pi_{\theta_{old}} (o_t | q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta} (o_t | q, o_{<t})}{\pi_{\theta_{old}} (o_t | q, o_{<t})}, 1 - \epsilon, 1 + \epsilon \right) A_t \right],$$
onde:

R_t_bar é a média das recompensas obtidas de várias amostras (por exemplo, o_1, o_2, ..., o_G) do modelo antigo para a mesma pergunta q. Essa média permite uma estimativa mais robusta da qualidade das respostas, reduzindo a influência de variações extremas em respostas individuais.

Ou seja, em vez de depender de um único resultado, o modelo avalia um conjunto de respostas para determinar um valor médio mais representativo do desempenho real da política. Isso ajuda a estabilizar o aprendizado e evita que mudanças aleatórias impactem excessivamente a atualização da política.

O termo KL(π_theta, π_old) representa a divergência Kullback-Leibler (KL) entre a nova política π_theta e a política antiga π_old. Esse termo atua como uma penalização, garantindo que a nova política não se desvie excessivamente da política anterior. A divergência KL mede a diferença entre duas distribuições de probabilidade e, neste caso, impede que a política treinada se afaste demais do comportamento aprendido previamente.

Em vez de ser tratada separadamente como um ajuste na recompensa, essa penalização é incorporada diretamente na função de perda, permitindo que o modelo aprenda a manter um equilíbrio entre exploração de novas respostas e conservação de padrões previamente eficientes.

Essa abordagem permite que o DeepSeek-R1-Zero treine sua política π_theta sem a necessidade de armazenar ou atualizar um modelo de valor extenso na GPU, o que melhora significativamente a escalabilidade. Normalmente, em algoritmos de Reforço, um modelo de valor é utilizado para prever o retorno esperado de um estado ou ação, mas manter esse modelo atualizado consome muita memória e processamento!

4. DeepSeek-R1: Integração entre SFT e RL
Após verificar as capacidades extraordinárias do DeepSeek-R1-Zero, observou-se que a formatação das respostas e a consistência de linguagem ainda necessitavam refinamento. Assim, surgiu o DeepSeek-R1, que adiciona etapas de Supervised Fine-Tuning (SFT) em duas fases, intercaladas com RL:

Fine-Tuning Inicial (SFT 1)

Utilização de um conjunto robusto (milhares) de exemplos de Chain-of-Thought cuidadosamente anotados.

Ajuste do DeepSeek-V3-Base para que o modelo aprenda convenções de resposta, clareza de exposição e formatação de tags.

Treinamento por RL (GRPO)

O modelo passa a gerar soluções com base em sinais de recompensa.

Dois modelos de recompensa são usados:

Recompensa de Acurácia: indica se a solução está correta (muito útil em tarefas de código e matemática, onde há verificação objetiva).

Recompensa de Formatação: garante o uso adequado de <think>...</think> e <answer>...</answer>.

Nova Rodada de Fine-Tuning (SFT 2)

Agora, o próprio modelo “ensina” a si mesmo: gera 800 mil exemplos, incluindo raciocínios (CoT) e respostas.

Esses exemplos são refinados e usados para re-fazer fine-tuning, aumentando a coerência e a legibilidade final.

5. Comparação entre Modelos DeepSeek R1
A tabela abaixo resume as diferenças de estratégia de treinamento e observações-chave de cada variante

6. Descobertas Recentes e Tendências até 2025
Com a consolidação dessas abordagens de RL em LLMs, surgiram diversas tendências e aprimoramentos técnicos no período de 2024 a 2025:

Expansão Maciça de Infraestrutura

Lançamento de clusters exa-escala, como o Frontier Next Gen e o Aurora-2, permitindo treinamento de modelos com trilhões de parâmetros.

Técnicas de gradient checkpointing otimizadas e tensor parallelism avançado reduziram o custo de memória GPU.

Modelos Autônomos com Self-Improvement

Seguindo a lógica de autoaprendizagem, laboratórios de pesquisa investiram em pipelines online onde o modelo gera dados, recebe recompensas e se refina continuamente — quase em “tempo real”.

Interpretação e Debugging de Cadeia de Raciocínio

Ferramentas como CoT Inspector (lançadas em meados de 2025) fornecem visualizações em tempo real das tokens de raciocínio <think>...</think> e permitem comparações com caminhos de raciocínio alternativos.

Isso ajudou a comunidade a compreender melhor a emergência de comportamentos complexos e a evitar alucinações perigosas.

Verificação Formal de Código

Avanço de frameworks de verificação formal que se integram às saídas de modelos como DeepSeek-R1, dando um selo de garantia adicional quando a solução gerada cumpre propriedades matemáticas ou invariantes lógicas.

Ferramentas como HolLight 2025 e Isabelle-Next vêm sendo adaptadas para processar diretamente as cadeias de raciocínio produzidas.

Surgimento de Meta-RL Models

Alguns grupos de pesquisa estão investigando uma camada de RL acima do RL, em que um modelo “master” gerencia a estratégia de treinamento e seleção de recompensas para múltiplos submodelos, incluindo variações de DeepSeek.

Ensaios iniciais mostram potencial para acelerar ainda mais a convergência de tarefas complexas.

7. Conclusões e Perspectivas Futuras

A trajetória do DeepSeek R1 (com destaque para o R1-Zero) ilustra até onde a aprendizagem por reforço pode levar modelos de linguagem de grande porte, especialmente quando apoiados por poderosos modelos base. Eis alguns pontos centrais:

RL e SFT não são excludentes: A união de ambos, em estágios diferentes, parece a forma mais eficaz de alcançar resultados de alta qualidade e legibilidade.

Dados Gerados pelo Próprio Modelo: O uso das próprias saídas do modelo para criar novos dados de treinamento é uma tendência crescente, possibilitando uma espécie de bootstrap contínuo.

Soluções Escaláveis: Remover o modelo de valor na abordagem GRPO ajudou a viabilizar redes gigantescas, abrindo caminho para arquiteturas cada vez mais elaboradas em 2025.

Verificabilidade e Segurança: Para aplicações críticas, a verificação formal e a análise do Chain-of-Thought se tornam instrumentos valiosos, assegurando que o modelo não apenas “adivinhe” corretamente, mas ofereça raciocínios consistentes e confiáveis.

Há muito a explorar em relação à governança, alinhamento ético e segurança desses modelos. Porém, o DeepSeek R1 e suas variantes deixaram evidente que a prática de RL puro (ou quase puro) com LLMs não é apenas um experimento teórico, mas também uma alternativa real e poderosa para a próxima geração de sistemas de IA.

Em suma, a família de modelos DeepSeek R1 redefiniu o que entendemos por RL em larga escala aplicada a geração de texto e resolução de problemas. Os avanços técnicos, especialmente até 2025, sugerem que futuras linhas de pesquisa poderão incorporar métodos de self-improvement contínuo, verificação formal integrada e arquiteturas ainda mais escaláveis, impulsionadas pelo crescimento da infraestrutura computacional exa-escala.

8. Referências

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms.

Silver, D. et al. (2017). Mastering the game of Go without human knowledge. Nature, 550(7676), 354–359.

Vaswani, A. et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS).

OpenAI & DeepSeek Team. DeepSeek-V3-Base Release Notes (2024).

Aurora-2 Project (2025). Exascale HPC Infrastructure for Next-Gen AI. White Paper.

HolLight 2025 & Isabelle-Next. (2025). Integrating Automated Theorem Proving with LLMs. Proceedings of the Large Language Models for Formal Methods Workshop.