BIS - MT · 12/05/2026

🤖 Bản Tin AI Hằng Ngày

Cập nhật công nghệ AI mới nhất

✨ *"Programs must be written for people to read, and only incidentally for machines to execute."*

↪ Chương trình phải được viết cho con người đọc, và chỉ ngẫu nhiên cho máy móc thực thi.

— Harold Abelson

💡 *Code tốt là code rõ ràng và dễ hiểu — viết code sạch, có tài liệu tốt là biểu hiện của sự chuyên nghiệp và tôn trọng đồng đội.*

📰 TIN TỨC NỔI BẬT

1 Các kỹ sư phần mềm của Google đang chuyển từ viết mã sang đưa ra quyết định

🇬🇧 *Google's software engineers are shifting from coding to calling the shots*

📰 Business Insider 🔗 Đọc bài viết →

> Các kỹ sư phần mềm của Google đang ngày càng đảm nhận các vai trò lãnh đạo trong công ty, rời xa trọng tâm truyền thống là lập trình. Sự thay đổi này là một phần của xu hướng rộng lớn hơn nơi chuyên môn kỹ thuật được đánh giá cao cùng với kiến thức kinh doanh. Nhiều kỹ sư của Google hiện đang lãnh đạo các đội, đưa ra quyết định chiến lược và thúc đẩy phát triển sản phẩm. Sự thay đổi này được cho là do công ty nhận ra rằng chuyên môn kỹ thuật là điều cần thiết cho thành công kinh doanh. Các kỹ sư của Google đang được đào tạo để đảm nhận các vai trò cao cấp hơn, nơi họ có thể tận dụng kiến thức kỹ thuật của mình để thông báo quyết định kinh doanh. Công ty cũng đang đầu tư vào các chương trình phát triển kỹ năng lãnh đạo và kinh doanh của các kỹ sư. Kết quả là, các kỹ sư phần mềm của Google đang trở nên có ảnh hưởng hơn trong việc định hình lộ trình sản phẩm và chiến lược kinh doanh của công ty. Sự thay đổi này có khả năng sẽ có tác động đáng kể đến hướng đi và thành công tương lai của công ty.

2 Marc Benioff nghĩ rằng AI chưa sẵn sàng để thay thế các kỹ sư phần mềm

🇬🇧 *Marc Benioff thinks AI isn't quite ready to replace software engineers*

📰 IT Pro 🔗 Đọc bài viết →

Giám đốc điều hành Salesforce Marc Benioff đã bày tỏ quan điểm của mình về vai trò của trí tuệ nhân tạo (AI) trong phát triển phần mềm. Theo Benioff, AI chưa sẵn sàng để thay thế các kỹ sư phần mềm con người. Mặc dù thừa nhận khả năng của AI trong việc tự động hóa một số nhiệm vụ, ông tin rằng công nghệ này vẫn thiếu sự phức tạp và tinh tế cần thiết để hoàn toàn sao chép công việc của các kỹ sư phần mềm có kỹ năng. Các bình luận của Benioff gợi ý rằng AI phù hợp nhất để tăng cường công việc của các nhà phát triển con người, chứ không phải thay thế họ hoàn toàn. Quan điểm này phù hợp với quan điểm của nhiều chuyên gia trong ngành, những người coi AI là một công cụ để tăng cường năng suất và hiệu quả, chứ không phải là sự thay thế cho chuyên môn con người. Tuyên bố của Benioff làm nổi bật cuộc tranh luận đang diễn ra về tác động tiềm năng của AI đối với ngành công nghiệp phát triển phần mềm. Khi AI tiếp tục phát triển, sẽ rất thú vị khi xem nó được tích hợp vào quá trình phát triển như thế nào và nó thay đổi vai trò của các kỹ sư phần mềm con người.

3 'Một kỷ nguyên mới của phát triển phần mềm': Claude Code khiến các kỹ sư tại Seattle phấn khích khi mã hóa AI đạt giai đoạn mới

🇬🇧 *'A new era of software development': Claude Code has Seattle engineers buzzing as AI coding hits new phase*

📰 GeekWire 🔗 Đọc bài viết →

Các kỹ sư tại Seattle đang sôi sục với sự phấn khích khi một kỷ nguyên mới của phát triển phần mềm xuất hiện với sự ra mắt của Claude Code. Công cụ mã hóa được hỗ trợ bởi AI này đang cách mạng hóa cách phần mềm được tạo ra, đánh dấu một cột mốc quan trọng trong sự tiến hóa của trí tuệ nhân tạo trong mã hóa. Claude Code là một nền tảng mã hóa AI cho phép các nhà phát triển viết mã nhanh hơn và hiệu quả hơn. Nó sử dụng các thuật toán học máy để phân tích và tạo mã, cho phép các kỹ sư tập trung vào các nhiệm vụ cấp cao hơn và giải quyết vấn đề sáng tạo. Khả năng của nền tảng đã tạo ra một làn sóng quan tâm trong cộng đồng công nghệ của Seattle, với nhiều kỹ sư háo hức khám phá tiềm năng của nó. Sự ra mắt của Claude Code đại diện cho một sự thay đổi đáng kể trong phong cảnh phát triển phần mềm, khi AI đảm nhận vai trò nổi bật hơn trong quá trình mã hóa. Khi ngành công nghệ tiếp tục phát triển, sẽ rất thú vị khi xem

Claude Code và các công cụ được hỗ trợ bởi AI khác định hình tương lai của phát triển phần mềm.

4 Tại sao việc□□EVM là thiết yếu cho các hệ thống không dây thế hệ tiếp theo

🇬🇧 *Why Mastering EVM Is Essential for Next-Generation Wireless Systems*

📰 IEEE Spectrum 🔗 Đọc bài viết →

Khi công nghệ truyền thông không dây tiếp tục phát triển, nhu cầu về tốc độ truyền dữ liệu cao hơn đã dẫn đến các sơ đồ điều chế ngày càng phức tạp. Các hệ thống hiện đại sử dụng điều chế biên độ quadrature (QAM) với các cấp độ cao, chẳng hạn như 4096QAM, nơi các sai lệch nhỏ về biên độ hoặc pha có thể gây ra lỗi bit. Để đo độ chính xác của các sơ đồ điều chế này, độ lớn của vector lỗi (EVM) đã trở thành chỉ số chính. Bài báo này nhằm mục đích bao gồm các nguyên tắc cơ bản của điều chế số, định nghĩa EVM và giải thích các phương pháp tính toán của nó. Nó cũng khám phá các nguồn gây suy giảm EVM phổ biến và thảo luận về cách các biểu đồ constellation có thể được sử dụng để chẩn đoán các suy giảm điều chế trong các hệ thống không dây thực tế. Việc hiểu EVM là điều cần thiết cho việc phát triển các hệ thống không dây thế hệ tiếp theo có thể cung cấp truyền dữ liệu đáng tin cậy và hiệu quả.

5 Google giới thiệu dự đoán đa token Gemma 4 LLM

🇬🇧 *Google brings multi-token prediction Gemma 4 LLMs*

📰 BD Tech Talks 🔗 Đọc bài viết →

Google đã thực hiện một bản nâng cấp đáng kể cho dòng Gemma 4 của các Large Language Models (LLMs) mã nguồn mở bằng cách tích hợp công nghệ dự đoán đa token (MTP). Sự đổi mới này cho phép Gemma 4 thoát khỏi cách tiếp cận truyền thống một token tại một thời điểm, dẫn đến tăng lưu lượng trên phần cứng cấp người tiêu dùng. Bằng cách dự đoán nhiều token song song, Gemma 4 tận dụng khả năng xử lý song song của các GPU hiện đại một cách hiệu quả hơn, giảm số lần phải tải trọng của model từ bộ nhớ. Điều này dẫn đến sự tăng cường hiệu suất trực tiếp cho người dùng cuối, khiến các tương tác cục bộ cảm giác tức thời và mượt mà hơn. Kiến trúc Gemma 4 bao gồm các model nhỏ, chuyên dụng hoạt động như những người soạn thảo, làm việc trước model chính để đoán những token tiếp theo.

Model chính sau đó xem xét các đề xuất này trong một lần đi qua, xác minh chúng bằng cơ chế chú ý của nó. Quá trình xác minh này cho phép model chính xác nhận một chuỗi toàn bộ trong khoảng thời gian tương đương với thời gian cần thiết để tạo ra chỉ một token. Các thí nghiệm cho thấy Gemma 4 có thể đạt được tốc độ tăng lên đến 3 lần trên cả phiên bản trên thiết bị và phiên bản lớn hơn của model.

6

Các khối xây dựng cho đào tạo và suy luận mô hình nền tảng trên AWS

🇬🇧 *Building Blocks for Foundation Model Training and Inference on AWS*

📰 Hugging Face Blog 🔗 Đọc bài viết →

AWS đã giới thiệu một cơ sở hạ tầng mới, được gọi là Building Blocks, để hỗ trợ việc đào tạo và suy luận của các mô hình nền tảng (foundation models). Truyền thống, việc mở rộng quy mô các mô hình nền tảng có nghĩa là đầu tư vào khả năng tính toán quy mô lớn, nhưng bối cảnh đã thay đổi. Giờ đây, hiệu suất được mở rộng thông qua các phương pháp sau đào tạo, tính toán thời gian thử nghiệm và các yếu tố khác. Sự thay đổi này yêu cầu một cơ sở hạ tầng hội tụ với khả năng tính toán gia tốc□ hợp chặt chẽ, mạng lưới băng thông cao độ trễ thấp và lưu trữ phân tán. Để quản lý các tài nguyên này, các công cụ điều phối như Slurm và Kubernetes là thiết yếu. Chu kỳ sống của mô hình nền tảng cũng phụ thuộc vào một hệ sinh thái phần mềm mã nguồn mở, bao gồm các framework như PyTorch và JAX cho việc phát triển mô hình và đào tạo phân tán. Các công cụ quan sát như Prometheus và Grafana được sử dụng để giám sát và trực quan hóa hiệu suất của hệ thống. Cơ sở hạ tầng Building Blocks của AWS tương tác với các chồng mã nguồn mở này để cung cấp một nền tảng kỹ thuật để hiểu các nút thắt và đặc điểm mở rộng của hệ thống. Loạt bài đăng này sẽ khám phá cách kiến trúc phân lớp này được thực hiện trên AWS, bao gồm cơ sở hạ tầng, điều phối tài nguyên, chồng phần mềm ML và khả năng quan sát.

7

GitLab Act 2

🇬🇧 *GitLab Act 2*

📰 Simon Willison 🔗 Đọc bài viết →

GitLab đã công bố chiến lược mới của mình, được gọi là "GitLab Act 2", bao gồm việc giảm lực lượng lao động và thay đổi cấu trúc để đáp

ứng với "thời đại agentic". Công ty tin rằng thời đại agentic, được đặc trưng bởi nhu cầu ngày càng tăng về phần mềm, sẽ dẫn đến sự mở rộng đáng kể của thị trường nền tảng developer. Khi chi phí sản xuất phần mềm giảm, nhu cầu dự kiến sẽ tăng, với thị trường chuyển từ vài chục đô la mỗi người dùng mỗi tháng sang hàng trăm hoặc thậm chí hàng nghìn. GitLab đang định vị mình để phục vụ một lượng lớn phần mềm và developer ngày càng tăng, dự đoán sự tăng trưởng nhu cầu. Tuy nhiên, giá cổ phiếu của công ty đã giảm từ khoảng 52 đô la xuống 26 đô la trong năm qua, làm dấy lên lo ngại về triển vọng tăng trưởng liên tục của công ty trong bối cảnh tác động của kỹ thuật agentic đối với thị trường cốt lõi của nó.

8

OpenAI vừa phát hành câu trả lời của mình cho Claude Mythos

🇬🇧 *OpenAI just released its answer to Claude Mythos*

📰 The Verge AI 🔗 Đọc bài viết →

OpenAI đã ra mắt Daybreak, một sáng kiến AI nhằm phát hiện và vá các lỗ hổng trong mã trước khi các kẻ tấn công có thể khai thác chúng. Động thái này được thực hiện để đáp lại thông báo của đối thủ Anthropic về Claude Mythos, một mô hình AI tập trung vào bảo mật ban đầu chỉ được chia sẻ riêng do lo ngại về các rủi ro tiềm năng của nó. Daybreak sử dụng đại lý AI bảo mật Codex của OpenAI để tạo một mô hình mối đe dọa dựa trên mã của một tổ chức và tự động hóa việc phát hiện các lỗ hổng có rủi ro cao. Sáng kiến này kết hợp nhiều mô hình AI, bao gồm Codex và GPT-5.5 với Truy cập Tin cậy cho An ninh mạng, và liên quan đến sự hợp tác với các đối tác trong ngành và chính phủ. Mục tiêu của Daybreak là cung cấp một giải pháp bảo mật toàn diện, chuẩn bị cho việc triển khai các mô hình có khả năng an ninh mạng ngày càng tăng trong tương lai.

⚡ TIPS & TRICKS CHO DEV

⚡ Tối ưu hóa GitHub Copilot

Vấn đề: Khi sử dụng GitHub Copilot, đôi khi gợi ý không chính xác do không hiểu rõ context.

Cách làm: Sử dụng lệnh `//` để cung cấp thêm thông tin cho Copilot về chức năng và yêu cầu cụ thể. Ví dụ: `// Hàm tính tổng của mảng số`.

Đánh giá: Hiệu quả khi cần viết code nhanh chóng và chính xác, đặc biệt với các dự án có yêu cầu cụ thể.

⚡ ChatGPT cho Debug

Vấn đề: Debug code tốn nhiều thời gian và công sức.

Cách làm: Sử dụng ChatGPT để mô tả lỗi và yêu cầu giúp đỡ, ví dụ: "Lỗi tại dòng 10, không hiểu lý do".

Đánh giá: Tiết kiệm thời gian khi cần giải quyết vấn đề nhanh chóng, nhưng nên kiểm tra lại kết quả.

⚡ Claude cho Document

Vấn đề: Viết tài liệu cho code là công việc nhàm chán và tốn thời gian.

Cách làm: Sử dụng Claude để tự động tạo tài liệu cho code, ví dụ: `Generate doc for this function`.

Đánh giá: Hiệu quả khi cần tạo tài liệu nhanh chóng, nhưng cần kiểm tra và chỉnh sửa lại cho chính xác.

📚 BÀI HỌC AI HÔM NAY CHO DEV

1. **Tối ưu chi phí & hiệu năng LLM**
2. Dev cần biết cách tối ưu hóa chi phí và hiệu năng của Large Language Model (LLM) để áp dụng vào dự án thực tế. Điều này giúp giảm thiểu chi phí tính toán và tăng tốc độ xử lý.
3. Ví dụ, sử dụng kỹ thuật pruning để giảm số lượng tham số trong mô hình.
4. 💡 Tip: Sử dụng thư viện như Hugging Face Transformers để tối ưu hóa hiệu năng LLM.