مفتوح المصدر · أبريل 2025

تعرّف على Qwen3.6-35B-A3B الذكاء الاصطناعي المحلي الذي يغيّر القواعد

علي بابا أطلقت موديلاً يشتغل على لابتوبك، يفكر على مستوى Enterprise، ولا يكلفك شيئاً. هنا الدليل الشامل لتشغيله على أي جهاز.

35B إجمالي / 3B فعّال | نافذة سياق 256K | متعدد الوسائط — صور ونصوص | وضع التفكير + المباشر | Apache 2.0 — مجاني كلياً | يشتغل بـ 13GB RAM (نسخة 2-bit)

قراءة 15 دقيقة أبريل 2025 فريق Conneqt AI

ما هو؟

أكثر موديل مفتوح كفاءة من علي بابا حتى الآن

Qwen3.6-35B-A3B هو أحدث موديل من فريق أبحاث Qwen في علي بابا، ويمثل قفزة حقيقية لكل من يشغّل ذكاءً اصطناعياً محلياً. يستخدم هندسة **Mixture of Experts (MoE)** — ما يعني أن الموديل يحتوي على 35 مليار بارامتر إجمالاً، لكن 3 مليار فقط تُفعَّل في كل مرة. تخيّلها كفريق من 35 متخصص: أنت تستدعي فقط الـ3 الأنسب لكل مهمة.

النتيجة موديل يؤدي كأنه أكبر بكثير — مقارب لموديلات بحجم 10 أضعاف حجمه الفعّال — بينما يستهلك ذاكرة وحوسبة موديل 3B فقط. لهذا النسخة المضغوطة 2-bit تشتغل على لابتوب عادي بـ 16GB وتتعامل مع تحليل قواعد الكود الكاملة، والمهام العميقة، والرؤية، والاستدلال المتعدد الخطوات.

نتيجة حقيقية

مستخدمون يشغّلون النسخة 2-bit (13GB RAM) أفادوا بإتمام bug hunt كاملة لقاعدة كود — مع أدلة، وخطوات إعادة إنتاج، وإصلاحات، واختبارات، وكتابة PR كاملة — كل ذلك من جلسة واحدة على لابتوب شخصي.

الموديل مرخّص تحت **Apache 2.0**، ما يعني أنه مجاني تجارياً بدون أي قيود. يمكنك نشره في الإنتاج، وضبطه الدقيق، وبناء منتجات فوقه.

الهندسة المعمارية

لماذا تجعله MoE مختلفاً؟

الموديلات التقليدية "الكثيفة" تُفعّل كل بارامتراتها لكل رمز. موديل 35B كثيف يستخدم 35B بارامتر في كل كلمة. موديل MoE مثل Qwen3.6 يستخدم 3B بارامتر فقط لكل رمز، يوجّه كل مدخل نحو المجموعة الأنسب من الطبقات.

الموديل الكثيف (التقليدي)

كل الـ 35B بارامتر فعّالة في كل وقت. استهلاك عالٍ للذاكرة. سرعة استدلال أبطأ. كل بارامتر يشارك في كل قرار.

تكلفة موارد عالية

موديل MoE (Qwen3.6)

35B إجمالي، 3B فعّال. استهلاك الذاكرة مثل موديل 3B. استدلال سريع. التوجيه يختار أفضل خبراء لكل مهمة تلقائياً.

كفاءة 10 أضعاف

يبني Qwen3.6 أيضاً على هندسة Gated Delta Network، والتي تجمع بين MoE الخفيف وآلية انتباه أكثر كفاءة. مع التدريب المبكر على النصوص، والكود، والصور في نفس الوقت، يمنحك **فهماً متعدد الوسائط أصيلاً** — وليس وحدة رؤية مضافة لاحقاً.

الهاردوير

ما الجهاز الذي تحتاجه؟

الجواب على الأرجح هو ما لديك بالفعل. الموديل يأتي بمستويات ضغط متعددة — نسخ مضغوطة تتنازل عن كمية صغيرة من الجودة مقابل متطلبات ذاكرة أقل بكثير:

الكوانتيزيشن	الرام المطلوب	الجودة	الأنسب لـ
`UD-Q2_K_XL` — 2-bit	**13GB~**		لابتوبات MacBook 16GB، القديم، الأجهزة الاقتصادية
`UD-Q3_K_XL` — 3-bit	**17GB~**		Mac 32GB، رام + GPU 16GB منفصل
`UD-Q4_K_XL` — 4-bit	**22GB~**		M1/M2/M3 Mac 24GB+، NVIDIA 24GB VRAM

قاعدة عامة

مجموع الذاكرة المتاحة (RAM + VRAM) يجب أن يساوي أو يتجاوز حجم ملف الموديل المضغوط للحصول على أفضل أداء. إذا كان أقل، الاستدلال يشتغل لكن بسرعة أبطأ بسبب offloading على الهارد.

كيف تشغّله — كل الطرق

اختر الطريقة المناسبة لمستواك التقني واحتياجاتك:

Ollama | LM Studio | llama.cpp | Unsloth Studio | Cloud / API

الأنسب لـ: معظم المستخدمين. أسرع إعداد. يشتغل على macOS وWindows وLinux. لا يتطلب GPU. نصوص فقط (للصور استخدم llama.cpp).

١ حمّل Ollama من **ollama.com** — مثبّت واحد لنظامك. يعمل كخدمة خلفية تلقائياً.

٢ شغّل الموديل في الـ terminal:

```
TERMINAL
Copy

ollama run qwen3.6:35b-a3b
```

Ollama يحمّل الموديل تلقائياً (13-22GB حسب رام جهازك) ويبدأ جلسة تفاعلية.

٣ تبديل وضع التفكير أثناء الجلسة:

```
TERMINAL
Copy

# تفعيل التفكير العميق (للكود والتحليل)
/set think

# إيقاف للردود السريعة المباشرة
/set nothink
```

٤ استخدم الـ API المتوافق مع OpenAI على `http://localhost:11434/v1` لتوصيل n8n، وOpen WebUI، وأي أداة أخرى.

٥ مثال Python:

```
PYTHON
Copy

from ollama import chat

response = chat(
model='qwen3.6:35b-a3b',
```

```
    messages=[{'role': 'user', 'content': 'مرحبا!'}]
)
print(response.message.content)
```

ميزة أساسية

وضع التفكير مقابل الوضع المباشر

من أقوى ميزات Qwen3.6 هو القدرة على التبديل بين سلوكَين استدلاليَّين مختلفَين كلياً — وأنت تتحكم بذلك في وقت التشغيل، بدون تغيير أي موديل.

وضع التفكير

الموديل يمر بالمسألة داخلياً قبل أن يعطي الجواب — مثل إظهار الخطوات. يستخدم سلسلة تفكير خفية يمكنك فحصها إذا أردت.

الأفضل لـ: التصحيح، الرياضيات، توليد الكود، التحليل المعقد، المهام متعددة الخطوات.

`think/` في الشات أو `enable_thinking: true` في الـ API

الوضع المباشر

ردود فورية مباشرة بدون استهلاك استدلال إضافي. أسرع بشكل ملحوظ. مثالي للمحادثات وتشغيل أتمتة عالية الإنتاجية.

الأفضل لـ: الشات، كتابة المحتوى، الترجمة، الأسئلة والأجوبة، الأدوات اللحظية.

`nothink/` في الشات أو `enable_thinking: false` في الـ API

حالات الاستخدام

ماذا يمكنك أن تبني به؟

🔍

Agentic Coding

تحليل قواعد الكود الكاملة، صيد البقز، PR writeups تلقائية، توليد اختبارات. مُثبّت على repos حقيقية بنسخة 2-bit.

👁

مهام الرؤية

تحليل screenshots وصور واجهات ومستندات ومخططات. متعدد الوسائط أصيل — ليس إضافة.

⚙️

أتمتة n8n

ادمجه في أي OpenAI node عبر Ollama API. نموذج محلي لا يرسل بياناتك لأي مكان.

💬

شات بوت خاص

Open WebUI فوق Ollama يعطيك واجهة ChatGPT كاملة — أوفلاين تماماً.

🌍

متعدد اللغات

201 لغة مع دعم عربي قوي. مثالي لمنتجات السوق الخليجي ومهام NLP العربية.

🎛

ضبط دقيق

ترخيص Apache 2.0 يعني حقوق تدريب كاملة. استخدم Unsloth أو transformers لتكييفه على مجالك.

التكامل

ربطه بمنظومتك الحالية

لأن Ollama وllama.cpp يوفّران API متوافقاً مع OpenAI، يندمج Qwen3.6 مع تقريباً أي أداة تدعم OpenAI — بدون كود مخصص.

مسارات n8n

في أي OpenAI node في n8n، غيّر الـ base URL لـ `http://localhost:11434/v1` واسم الموديل لـ `qwen3.6:35b-a3b`. مساراتك الحالية تشتغل فوراً بموديل محلي — صفر تكاليف API، صفر بيانات تغادر جهازك.

Open WebUI

Open WebUI يكتشف Ollama تلقائياً على نفس الجهاز. Settings ← Connections ← Ollama URL (`http://localhost:11434`). موديلك يظهر في قائمة الاختيار تلقائياً.

Continue.dev / Cursor / Zed

أضف Ollama كـ provider في أي من هذه الإضافات للـ IDE. المزج بين السرعة (3B بارامتر فعّال) والجودة يجعله من أفضل مساعدي الكود المحليين المتاحة.

Claude Code / Clawbot / Codex / OpenCode

كل هذه الأدوات الـ agentic تدعم Ollama. وجّه الـ base URL لـ `localhost:11434/v1`، حدد اسم الموديل، وشغّل نفس مسارات الـ agentic — محلياً، بخصوصية كاملة، مجاناً.

الإعدادات المثلى

الإعدادات للحصول على أفضل أداء

للبرمجة والمهام العميقة

الإعدادات الموصى بها

```
temperature: 0.7
      top_p: 0.8
      top_k: 20
   thinking: ON
presence_penalty: 1.0  (يقلل التكرار)
```

للشات وكتابة المحتوى

الإعدادات الموصى بها

```
temperature: 0.9
top_p: 0.95
top_k: 20
thinking: OFF
presence_penalty: 1.0
```

إذا كان الموديل يكرر نفسه

أضف `presence-penalty 1.5--` لأمر llama.cpp، أو اضبط `presence_penalty: 1.5` في Ollama Modelfile. هذه أكثر مشكلة شائعة في موديلات MoE المحلية ومن أسهل ما يُحَل.

الخلاصة

لماذا هذا الموديل مهم الآن

انتهت حجة "تحتاج GPU قوي لتشغيل ذكاء اصطناعي جاد محلياً". Qwen3.6-35B-A3B بنسخة 2-bit يؤدي مهام agentic coding حقيقية على هاردوير عادي — وليس ديمو للتجربة، بل عمل هندسي حقيقي.

للمبنيّين في منطقة الخليج تحديداً، هذا يفتح باباً مختلفاً: موديل بـ **دعم قوي للغة العربية**، 201 لغة إجمالاً، قدرات رؤية، ونافذة سياق 256K — يشتغل على بنيتك التحتية بخصوصية كاملة بدون تكاليف API ودون أن تغادر بياناتك يدك.

ترخيص Apache 2.0 يعني بناء ونشر وبيع منتجات فوقه بدون قيود. هندسة MoE تعني جودة موديل 35B بتكلفة استدلال 3B. وضغط Unsloth Dynamic يعني تشغيله على هاردوير يملكه معظم الناس بالفعل.

ابدأ من هنا

إذا عندك Ollama مثبّت: `ollama run qwen3.6:35b-a3b` — هذا كل شيء. الموديل يُحمَّل تلقائياً، ويختار الضغط المناسب لهاردوير جهازك، وخلال دقائق أنت تشغّل ذكاءً اصطناعياً على مستوى Enterprise محلياً.