

Sprachmodelle mit Ollama auf dem Raspberry Pi nutzen



KI-Sprachmodelle wie ChatGPT, Claude, Gemini, Deepseek etc. sind angekommen und unterstützen bei vielen Anwendungen – von der Frage, warum der Himmel eigentlich blau ist, bis zur Unterstützung beim Programmieren. Und auch bei vielen [Projekten hier bei Pollux Labs kommt ihre künstliche Intelligenz zum Einsatz](#).

Einige dieser Modelle kannst du nur über die zugehörige Webseite oder über eine API verwenden – zum Beispiel OpenAI's ChatGPT oder auch Claude von Anthropic. Andere kannst du hingegen auch lokal installieren – **und damit kostenlos und ohne Preisgabe deiner Daten nutzen**. Wie du [Sprachmodelle mit Ollama lokal verwendetest](#), liest du auch bei uns. Hier ist es allerdings so, dass du schon einen recht modernen und leistungsfähigen Computer benötigst, um eine zufriedenstellende Antwortgeschwindigkeit zu erreichen.

Aber immer wieder kommt die Frage auf, ob man ein Sprachmodell auch auf einem Raspberry Pi verwenden könnte. Prinzipiell ist das kein Problem, der Knackpunkt ist jedoch die Hardware: Ein Raspi ist für die Modelle, die du vielleicht schon im Browser genutzt hast, einfach nicht leistungsstark genug. **Es gibt jedoch auch Modellversionen, die so stark reduziert wurden, dass du sie auch auf einem relativ schwachen Computer einsetzen kannst – also auch auf dem Raspberry Pi.**

Wie das funktioniert und mit welchen Einschränkungen du rechnen musst, erfährst du in diesem Artikel.

Das passende Raspberry Pi Modell

An einem **Raspberry Pi 5 mit mindestens 8GB RAM** führt leider kein Weg vorbei. Die 4er-Reihe ist leider nicht leistungsstark genug. Installiere am besten ein frisches Betriebssystem: Raspberry Pi OS (64-bit). Falls du den Raspberry Pi Imager verwendest, aktiviere auch gleich SSH, um von einem anderen Computer auf deinen Raspberry Pi zugreifen zu können. Du kannst ihn aber natürlich auch an einen Monitor anschließen und das Terminal des Raspi verwenden. [Wie du SSH verwendest, erfährst du in diesem Projekt.](#)

Update: Weiter unten liest du, wie gut Sprachmodelle auf dem Raspberry Pi mit 16GB RAM funktionieren.

Sobald du auf den Raspberry Pi zugreifen kannst, bzw. darauf ein Terminal offen hast, aktualisiere zunächst das Betriebssystem:

```
sudo apt update  
sudo apt upgrade
```

Anschließend benötigst du cURL (ein Werkzeug, das häufig verwendet wird, um HTTP-Anfragen zu senden und Daten von Webservern abzurufen oder an diese zu senden). Oft ist es bereits installiert, aber um sicherzugehen, verwende diesen Befehl im Terminal:

```
sudo apt install curl
```

Ollama auf dem Raspberry Pi installieren

Nun kann es direkt mit der Installation von Ollama weitergehen. Dieses Programm benötigst du, um Sprachmodelle herunterzuladen und dann auf deinem Raspberry Pi auszuführen.

STEADY_PAYWALL

Zum Einsatz kommt hierfür ein sogenannter „One-liner“, mit dem du da Installationsscript herunterlädst und direkt ausführst. Wenn du möchtest (in der Regel ist das empfehlenswert), [wirf einen Blick in den Code des Scripts](#).

```
curl -fsSL https://ollama.com/install.sh | sh
```

Sobald die Installation abgeschlossen ist, kannst du die Ollama-Version mit diesem Befehl einsehen:

```
ollama --version
```

Aktuell (März 2025) ist das die Version 0.5.13

```
[pi@raspberrypi:~ $ ollama --version
ollama version is 0.5.13
pi@raspberrypi:~ $ ]
```

Sprachmodelle laden und starten

Zeit für dein erstes Sprachmodell auf dem Raspberry Pi.

Deepseek-r1

Zunächst ein Test mit Deepseek-r1 – dieses Modell ist bei Ollama in einer ganzen Reihe von Versionen verfügbar: bis zu

einer Größe von 671 Milliarden Parametern, das satte 404 GB Platz (und sehr teure Hardware) benötigt. Für den Raspberry Pi soll es zunächst das kleinste Modell mit 1.5 Milliarden Parametern sein.

Installiere dieses Sprachmodell und starte es direkt mit diesem Befehl:

```
ollama run deepseek-r1:1.5b
```

Die Installation des 1,1 GB großen Modells dauert etwas, aber dann siehst du im Terminal die Eingabemöglichkeit für deine Frage:

```
[pi@raspberrypi:~ $ ollama run deepseek-r1:1.5b
>>> Send a message (/? for help)
```

Ich habe als erstes die Frage „Warum ist der Himmel blau?“ gestellt. Eine Besonderheit von Deepseek-r1 ist, dass das Modell zunächst „denkt“, bevor es eine Antwort ausspuckt. Diesem Denkprozess kannst du folgen, denn er erscheint Zeichen für Zeichen in deinem Terminal, zwischen den beiden Tags <think> </think>.

Bei meinem Test, hat Deepseek allerdings aufs Denken verzichtet und die folgende, leider völlig abstruse Antwort geliefert:

*Ich verstehe, dass du den Begriff „Himmel“ und die Farbcode „blau“ vermutlich verwechseln kannst. In deutscher Sprachkultur wird der **Himmel** eine vierteltnälle (eine von der Erde manch Zeitendatum backstependen seebaren Stile) genannt, die als ein abgeschlossenes, wissenshares Gravität generally vermutlich ist.*

Die Geschwindigkeit hierbei war allerdings in Ordnung – kein Vergleich zum Browser, aber schnell genug. Für dieses

Sprachmodell reicht also tatsächlich ein Raspberry Pi aus.

Ein zweiter Versuch, diesmal auf Englisch. Auch hier verzichtet Deepseek aufs Denken, antwortet aber richtig (zumindest soweit ich das beurteilen kann):

```
>>> why is the sky blue?  
<think>  
  
</think>  
  
The apparent blue color of the sky is due to a combination of physical and chemical  
processes. Here's how it works:  
  
1. **Wavelength Shifting ( atmosphere) **:  
   - Light from the sun enters Earth's atmosphere.  
   - Different gases in the air absorb light at different wavelengths.  
   - Blue light (shorter wavelength) is absorbed by the atmosphere more efficiently  
than longer wavelengths (like red).  
   - This causes blue light to scatter back up through the atmosphere, while shorter  
blue and green wavelengths travel farther, making the sky appear blue.  
  
2. **Scattering in the Air **:  
   - Gas molecules in Earth's atmosphere, such as nitrogen and oxygen, scatter blue and  
green light.  
   - Dust particles and water droplets also absorb and scatter these colors of light.
```

Die Unterstützung für die deutsche Sprache ist offensichtlich wenn, nur unzureichend in diesem kleinen Modell vorhanden.

Übrigens, Deepseek versteht hier im Terminal auch Folgefragen – nicht wie bei der Nutzung über die API. Du kannst also auch eine Art Konversation mit dem Modell führen.

Wenn du das Sprachmodell verlassen möchtest, tippe einfach **/bye** ein. Anschließend landest du wieder bei der Eingabeaufforderung deines Raspis. Wenn du das Modell wieder starten möchtest, verwende erneut den obigen Befehl. Die Installation entfällt dann natürlich.

TinyLlama

Das Sprachmodell TinyLlama ist ein für limitierte Hardware angepasstes Modell mit 1,1 Milliarden Parametern, das auf Metas Llama-Modell mit 3 Milliarden Parametern basiert. Um es zu installieren und zu testen, verwende den folgenden Befehl:

```
ollama run tinyllama
```

Auch hier ist eine Konversation auf Englisch brauchbar. Die Geschwindigkeit ist jedoch mit Deepseek vergleichbar, genauso wie die Qualität:

```
pi@raspberrypi:~ $ ollama run tinyllama
>>> Why is the sky blue?
The sky blue color has been observed in nature due to the reflected light of the sun through layers of cloudy atmosphere. The colors found in the clouds are caused by various factors such as sunlight, water droplets, and mineral dust.

Over time, these light reflectors (such as clouds) accumulate energy due to their density, which causes them to expand and form larger and larger structures. These structures take longer to collapse back into their original forms, which creates the blueish hue that we see in nature's sky.

Additionally, various mineral deposits on Earth also contribute to the colors seen in skies. They absorb different wavelengths of light than clouds, which results in varying shades of blue and other colors.
```

Phi-3

Noch ein dritter Test. Phi-3 ist ein Sprachmodell von Microsoft mit 3,3 Milliarden Parametern – also doppelt bis dreimal so groß wie Deepseek-r1 und TinyLlama. So installierst du es:

```
ollama run phi3
```

Dieses Modell kannst du nun auch auf Deutsch befragen. Warum der Himmel blau ist, beantwortet es folgendermaßen:

```
pi@raspberrypi:~ $ ollama run phi3
>>> Warum ist der Himmel blau?
Der Grund, warum wir den Himmel als "blau" beschreiben, liegt in einem interessanten Phänomen namens Rayleigh-Streuung. Dieses physikalische Verhalten bezieht sich auf die Streuung von Licht durch atmosphärische Gase und winzigen Partikel. Wenn das Sonnenlicht in den Weltraum fällt, tritt es mit Gasmolekülen der Erdatmosphäre zusammen, welche je nach Wellenlänge des Lichts unterschiedlich lange gestreut werden.

Sieht man vom Boden aus, erscheinen die Strahlen, die direkt von uns herunterkommen (das sogenannte "zunehmende Teil"), als hell und blau – weil sie kürzeres Strecklicht haben. Andere Strahlen wie solche, welche auf ihrer Reise über das Meerbedecke hinweg gestreut werden ("abbaue Teile"), erscheinen grauer oder röter.

Wenn wir die Erde nicht hatten und stattdessen im Weltraum gelegen wären, könnten wir einen anderen Blickwinkel darauf bekommen – ein schlichteres Aussehen ohne blaue Tinge des Himmels sehen. Dies zeigt an, dass es bei der Wahrnehmung unserer Umgebung vielmehr eine Kombination aus physikalischen Prozessen und subjektiven Empfindungen gibt!
```

Abstriche musst du hierfür allerdings bei der Geschwindigkeit machen, dein Raspberry Pi hat mit diesem Modell ganz schön zu arbeiten. Dafür sind auch Folgefragen wie „Und warum ist er abends rot?“ möglich. Phi-3 weiß dann immer noch, dass du dich auf die Farbe des Himmels beziehst.

Gemma 3 1B

Das kleinste Modell benötigt gerade einmal 815MB auf deinem Raspberry Pi. Da es damit auch für das 8GB-Modell geeignet sein sollte, erwarte ich auf dem Raspi mit doppelt so viel Arbeitsspeicher eine entsprechende Leistungsteigerung. Installiere das Modell:

```
ollama run gemma3:1b
```

Die obligatorische Frage, warum der Himmel blau ist, kommt tatsächlich flott. Und sie ist auch recht kompetent, hier der Anfang:

*Der Himmel ist blau aus einem faszinierenden physikalischen Phänomen namens **Rayleigh-Streuung**. Hier ist eine einfache Erklärung:*

- 1. Sonnenlicht und seine Farben: Sonnenlicht erscheint uns weiß, aber es besteht tatsächlich aus allen Farben des Regenbogens (Rot, Orange, Gelb, Grün, Blau, Indigo, Violett).*
- 2. Die Atmosphäre: Die Erde ist von einer dichten Atmosphäre umgeben, die aus Gasmolekülen wie Stickstoff und Sauerstoff besteht.*
- 3. Die Rayleigh-Streuung: Wenn Sonnenlicht in die Atmosphäre eintritt, trifft es auf diese Gasmoleküle. Die Farben des Lichts werden in unterschiedlichen Winkeln gestreut. Die Rayleigh-Streuung ist besonders effektiv bei kürzeren Wellenlängen des Lichts, also den blauen und violetten Farben.*

Und hier ist noch längst nicht Schluss. Die Ausführungen gehen noch weiter und beinhalten auch Überlegungen zu Sonnenauf- und -untergang sowie den Wolken. Auch am Deutsch lässt sich nicht mäkeln, die Antwort ist einwandfrei formuliert.

Gemma 3 4B

Weiter zum nächstgrößeren Modell mit 4 Billionen Parametern. Dieses benötigt bereits 3,3GB Platz. Lade und starte es mit:

```
ollama run gemma3:4b
```

Hier benötigst du schon etwas mehr Geduld, für die Antwort benötigt Gemma 3 dreimal so lang wie beim kleineren Modell. Sehr viel kompetenter ist sie dabei nicht (aber gut, die Antwort des kleineren Modells war ja schon in Ordnung). Hier der Anfang der Antwort:

*Der Himmel ist blau aufgrund eines Phänomens namens **Rayleigh-Streuung**. Hier ist eine einfache Erklärung:*

1. ***Sonnenlicht ist weiß:** Sonnenlicht, das die Erde erreicht, ist eigentlich eine Mischung aus allen Farben des Regenbogens.*

2. ***Lichtwellen und die Atmosphäre:** Die Erdatmosphäre besteht aus winzigen Teilchen wie Stickstoff- und Sauerstoffmolekülen. Wenn Sonnenlicht auf diese Moleküle trifft, wird es in verschiedene Richtungen gestreut – ähnlich wie ein Glasglas Licht bricht.*

Gemma 3 gibt es auch noch mit 12 und 27 Billionen Parametern – allerdings erübrigt sich hier ein weiterer Test, denn die Ausführung dürfte entweder noch um einiges langsamer sein, oder sogar unmöglich.

Fazit

Es ist also durchaus möglich, Ollama und damit verschiedene KI-Sprachmodelle auf dem Raspberry Pi zu verwenden. Abstriche musst du allerdings hinsichtlich der Eingabesprache (Deutsch), der Geschwindigkeit und sicherlich auch bei der Qualität der

Antworten machen.

Nichtsdestotrotz ist das ein spannender Ansatz, den du in einem Projekt einsetzen kannst!