

# Running the Vicuna-33B/13B/7B Chatbot with FastChat

## Introduction

The Vicuna chatbot is an open-source conversational AI model trained using fine-tuning LLaMA on user-shared conversations collected from ShareGPT. It has demonstrated remarkable performance, surpassing other models such as OpenAI ChatGPT, Google Bard, LLaMA, and Stanford Alpaca in more than 90% of cases. This case study will guide you through initializing the environment and running the Vicuna chatbot using the FastChat inference framework.

## Model and Software References:

- Vicuna Blog: [<https://lmsys.org/blog/2023-03-30-vicuna/>]
- FastChat GitHub Repository: [<https://github.com/lm-sys/FastChat>]

## Installation and Setup

```
# Create conda environment
# conda create -n [env_name]
conda create -n chatbotDemo
# source activate [env_name]
source activate chatbotDemo

# Install required packages
conda install pip
pip3 install fschat
```

## Loading up the environment

You may activate the prepared environment at any time by running the following:

```
# source activate [env_name]
source activate chatbotDemo
```

## Launch a chatbot with one GPU

To run the Vicuna chatbot using a GPU, execute the following command:

```
# request 4 core, 50 GB RAM, 3g.40gb GPU resource with interactive shell
srun -p gpu --gpus 3g.40gb:1 -c 4 --mem 50000 --pty bash

source activate chatbotDemo

python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-13b --style rich

python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-13b-v1.3 --style rich

# a smaller version Vicuna-7B is also provided

python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-7b --style rich

# vicuna 33b model requires more resources

# request 16 core, 100 GB RAM, a100 GPU resource with interactive shell
srun -p gpu --gpus a100:1 -c 16 --mem 100000 --pty bash

python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-33b-v1.3 --style rich
```

```
Terminal | OASIS HPC Cent...  
hpccenter.hk/terminal  
Last login: Mon Apr 24 17:49:12 2023 from 10.90.155.61  
The following file systems are available for your access:  
SHOME = /pfss/home/milo  
SSCRATCH = /pfss/scratch01/milo  
SSCRATCH_APPCARA = /pfss/scratch02/appcara  
(base) [milo@oaklbhpcgu085 ~]$ srun -p gpu --gpus 3g.48gb:1 -c 4 --mem 50000 --pty bash  
srun: Checking quota for (milo/appcara/gpu)  
(base) [milo@oaklbhpcgu085 ~]$ source activate chatbotDemo  
(chatbotDemo) [milo@oaklbhpcgu085 ~]$ python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-13b --style rich  
Loading checkpoint shards: 100% | 3/3 [00:32<00:00, 10.99s/it]  
USER:  
Implement a Python function to compute fibonacci number.  
ASSISTANT:  
Here's a Python function to compute the nth Fibonacci number:  
  
def fibonacci(n):  
    # The recurrence relation for Fibonacci numbers:  
    # fib(n) = fib(n-1) + fib(n-2)  
    # Starting values: fib(0) = 0, fib(1) = 1  
    a, b = 0, 1  
    for i in range(n):  
        yield a  
        a, b = b, a + b  
  
You can call this function with a positive integer n to get the nth Fibonacci number, and the function will return a generator that yields the successive Fibonacci numbers. For example:  
  
>>> for n in range(10):  
...     print(fibonacci(n))  
...  
0  
1  
1  
2  
3  
5  
8  
13  
21  
34  
  
This function uses the recurrence relation for Fibonacci numbers to compute the nth Fibonacci number, where fib(n) = fib(n-1) + fib(n-2) and fib(0) = 0 and fib(1) = 1. The function uses a generator to return the successive Fibonacci numbers, which can be more memory-efficient than computing all the numbers at once.  
USER:
```

## Launch with CPU

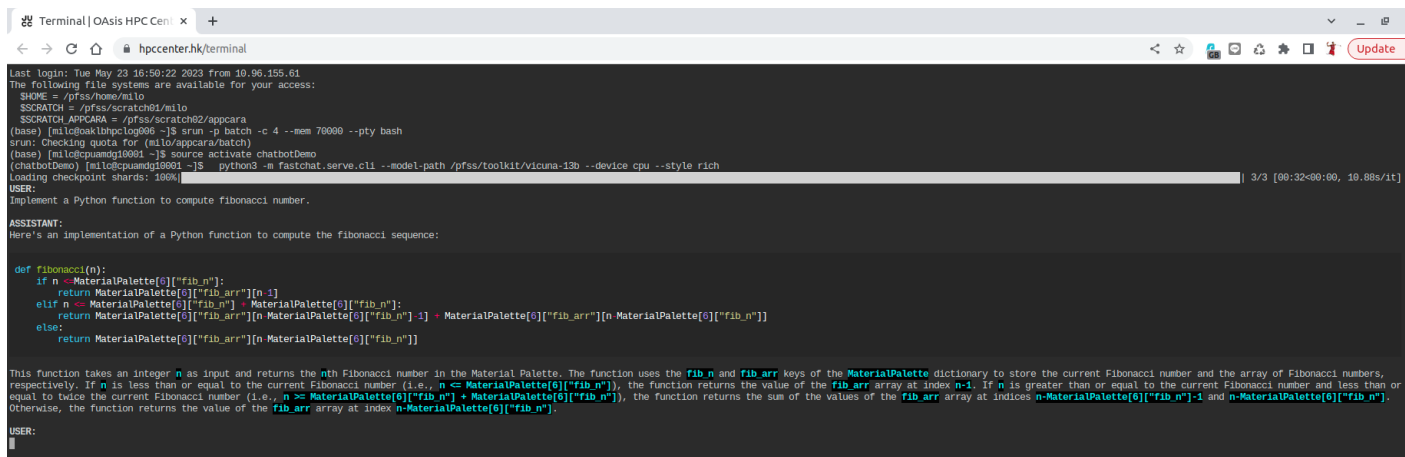
If you prefer to run the chatbot on a CPU (requires around 60GB of CPU memory), follow these steps:

```
# request 4 core, 70 GB resource with interactive shell

srun -p batch -c 4 --mem 70000 --pty bash

source activate chatbotDemo

python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-13b --style rich
```



```
Terminal | Oasis HPC Center x +
hpccenter.hk/terminal
Last login: Tue May 23 16:50:22 2023 from 10.96.195.61
The following file systems are available for your access:
  SHOME = /pfss/home/milo
  SSCATCH = /pfss/scratch01/milo
  SSCATCH_APPCORA = /pfss/scratch02/appcara
(base) [milo@cpuandg10090 ~]$ srun -p batch -c 4 --mem 70000 --pty bash
srun: Checking quota for (milo/appcara/batch)
(base) [milo@cpuandg10090 ~]$ source activate chatbotDemo
(chatbotDemo) [milo@cpuandg10090 ~]$ python3 -m fastchat.serve.cli --model-path /pfss/toolkit/vicuna-13b --device cpu --style rich
Loading checkpoint shards: 100%
USER:
Implement a Python function to compute fibonacci number.

ASSISTANT:
Here's an implementation of a Python function to compute the fibonacci sequence:

def fibonacci(n):
    if n <= MaterialPalette[0]["fib_n"]:
        return MaterialPalette[0]["fib_arr"][n-1]
    elif n <= MaterialPalette[0]["fib_n"] + MaterialPalette[0]["fib_n"]:
        return MaterialPalette[0]["fib_arr"][n-MaterialPalette[0]["fib_n"]-1] + MaterialPalette[0]["fib_arr"][n-MaterialPalette[0]["fib_n"]]
    else:
        return MaterialPalette[0]["fib_arr"][n-MaterialPalette[0]["fib_n"]]

This function takes an integer n as input and returns the nth Fibonacci number in the Material Palette. The function uses the fib_n and fib_arr keys of the MaterialPalette dictionary to store the current Fibonacci number and the array of Fibonacci numbers, respectively. If n is less than or equal to the current Fibonacci number (i.e., n <= MaterialPalette[0]["fib_n"]), the function returns the value of the fib_arr array at index n-1. If n is greater than or equal to the current Fibonacci number and less than or equal to twice the current Fibonacci number (i.e., n >= MaterialPalette[0]["fib_n"] + MaterialPalette[0]["fib_n"]), the function returns the sum of the values of the fib_arr array at indices n-MaterialPalette[0]["fib_n"]-1 and n-MaterialPalette[0]["fib_n"]. Otherwise, the function returns the value of the fib_arr array at index n-MaterialPalette[0]["fib_n"].

USER:
█
```

## Conclusion

Following these steps, you can successfully set up and run the Vicuna-13B chatbot using the FastChat inference framework. Feel free to explore fine-tuning the model and evaluating the chatbot using the resources available on [the Vicuna-13B website](#).

### Revision #9

Created 23 May 2023 10:08:42 by Milo Cheung

Updated 29 June 2023 04:11:48 by Milo Cheung