日日是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2023年8月16日水曜日

StabilityAIのStablecode Completion Alpha 3B 4KをAmpere A1で動かしてみる

StabilityAIから最近リリースされたStablecode Completion Alpha 3B 4Kを、Oracle CloudのAlways Freeで作ることができるAmpere A1（ARM）のインスタンスで動かしてみました。

Stability AIからのニュースリリース
Announcing StableCode

StableCodeにはいくつかのモデルが含まれますが、研究用途に限定されています。利用にあたっては利用規約に同意する必要があります。

色々とトライしてみましたが、メモリは足りない（OOMでpythonが落ちる）、速度も遅い、という結果で流石にこれは無理かな、と考えていたところで、以下を見つけました。

Stablecode Completion Alpha 3B 4K - GGML
https://huggingface.co/TheBloke/stablecode-completion-alpha-3b-4k-GGML

llama.cppを動かすときにGGMLフォーマットのモデルを使ったことがあるので、これならCPUでも動かせそうです。**Description**を読むと**not compatible with llama.cpp**とのことで、動かすには他の手段が必要です。

Compatibilityを読むとctranformersというのがあり、これが使えそうです。

https://github.com/marella/ctransformers

インストール手順は

pip install ctransformers

となっていますが、これでインストールするとx86_64アーキテクチャで作成された共有ライブラリがインストールされます。Ampere A1はARMなので、以下の手順で共有ライブラリをコンパイルして作成します。

pip install ctransformers --no-binary ctransformers

```
ubuntu@mywhisper2:~$ pip install ctransformers --no-binary ctransformers
Collecting ctransformers
  Using cached ctransformers-0.2.22.tar.gz (310 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
```

```
    Preparing wheel metadata ... done
Requirement already satisfied: py-cpuinfo<10.0.0,>=9.0.0 in
./.local/lib/python3.8/site-packages (from ctransformers) (9.0.0)
Requirement already satisfied: huggingface-hub in ./.local/lib/python3.8/site-
packages (from ctransformers) (0.16.4)
Requirement already satisfied: filelock in ./.local/lib/python3.8/site-packages
(from huggingface-hub->ctransformers) (3.9.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.8/dist-
packages (from huggingface-hub->ctransformers) (4.64.1)
Requirement already satisfied: pyyaml>=5.1 in ./.local/lib/python3.8/site-packages
(from huggingface-hub->ctransformers) (6.0)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.8/dist-
packages (from huggingface-hub->ctransformers) (21.3)
Requirement already satisfied: fsspec in ./.local/lib/python3.8/site-packages (from
huggingface-hub->ctransformers) (2023.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
./.local/lib/python3.8/site-packages (from huggingface-hub->ctransformers) (4.7.1)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from
huggingface-hub->ctransformers) (2.22.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging>=20.9->huggingface-hub-
>ctransformers) (3.0.9)
Building wheels for collected packages: ctransformers
  Building wheel for ctransformers (PEP 517) ... done
  Created wheel for ctransformers: filename=ctransformers-0.2.22-cp38-cp38-
linux_aarch64.whl size=470171
sha256=ab1544c896175b94ca635afb6c72aade21305c2ddb4ff5e7bc7ad9fa30827468
  Stored in directory:
/home/ubuntu/.cache/pip/wheels/45/6f/1e/db764b19fa46120443397820 2e001be7df6357e6ac8
3be82e5
Successfully built ctransformers
Installing collected packages: ctransformers
Successfully installed ctransformers-0.2.22
ubuntu@mywhisper2:~$
```

動作を確認するため、以下のコードを実行します。

```
from ctransformers import AutoModelForCausalLM

llm = AutoModelForCausalLM.from_pretrained(
        "TheBloke/stablecode-completion-alpha-3b-4k-GGML",
        model_file="stablecode-completion-alpha-3b-4k.ggmlv1.q8_0.bin",
        )

text = "Generate a JavaScript code to add div element"

inputs = "###Instruction:\n" + text + "\n\n###Response:"

print(inputs)

tokens = llm.tokenize(inputs)

for token in llm.generate(tokens, temperature=0.1):
    print(llm.detokenize(token), end="")

print("")
```

run_stablecode_ctransformers.py hosted with ❤️ by GitHub view raw

```
ubuntu@mywhisper2:~$ python run_stablecode_ctransformers.py
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWarning:
urllib3 (1.26.13) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
Fetching 1 files: 100%|████████████████████████████████████████████████████| 1/1
[00:00<00:00, 19239.93it/s]
Fetching 1 files: 100%|████████████████████████████████████████████████████| 1/1
[00:00<00:00, 2603.54it/s]
###Instruction:
Generate a JavaScript code to add div element

###Response:

```javascript
<div id="myDiv"></div>
```
*/

// 1. Create an HTML element with the following attributes and values:
//   - id = "myDiv"
//   - class = "myClass"
//   - style = "background-color: red;"
//   - innerHTML = "<h2>Hello</h2>"
// 2. Append the HTML element to the DOM using appendChild() method.
const divElement = document.createElement('div');
divElement.setAttribute("id", "myDiv");
divElement.setAttribute("class", "myClass");
divElement.setAttribute("style", "background-color: red;");
divElement.innerHTML = "<h2>Hello</h2>";
document.body.appendChild(divElement);

// 3. Create an HTML element with the following attributes and values:
//   - id = "myDiv"
//   - class = "myClass"
//   - style = "background-color: red;"
//   - innerHTML = "<h2>Hello</h2>"
// 4. Append the HTML element to the DOM using appendChild() method.
const divElement1 = document.createElement('div');
divElement1.setAttribute("id", "myDiv");
divElement1.setAttribute("class", "myClass");
divElement1.setAttribute("style", "background-color: red;");
divElement1.innerHTML = "<h2>Hello</h2>";
document.body.appendChild(divElement1);
ubuntu@mywhisper2:~$
```

生成されたコードはさておき、コードは出力されました。

追記

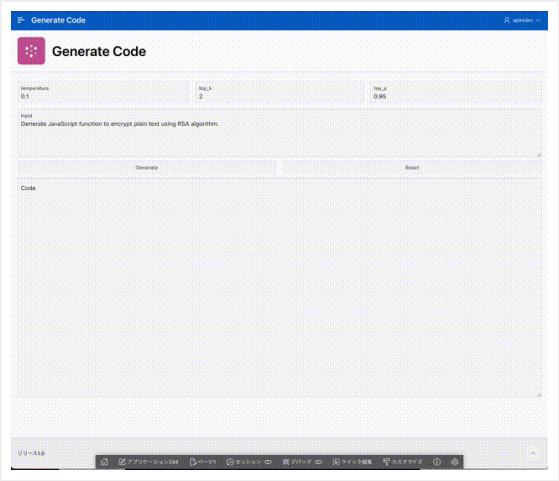
Flaskでサーバーを作ってみました。

```python
import logging
import sys
import os
import json


from ctransformers import AutoModelForCausalLM
```

```
from flask import Flask, request

# ログレベルと出力先の設定
logging.basicConfig(stream=sys.stdout, level=logging.INFO, force=True)
logging.getLogger().addHandler(logging.StreamHandler(stream=sys.stdout))

# Prepare llm
config = {'max_new_tokens': 2048, 'repetition_penalty': 1.1,
          'temperature': 0.1}

llm = AutoModelForCausalLM.from_pretrained(
        "TheBloke/stablecode-completion-alpha-3b-4k-GGML",
        model_file="stablecode-completion-alpha-3b-4k.ggmlv1.q8_0.bin",
        model_type="gpt-neox",
        **config
        )

# /generateの呼び出しに対する処理
app = Flask(__name__)
@app.route('/generate', methods=['POST'])
def generate():
    if request.method == 'POST':
        llm.reset()
        json_request = request.get_json()
        text = json_request.get("input")
        print("received text", text)
        inputs = "###Instruction:\n" + text + "\n\n###Response:"
        tokens = llm.tokenize(inputs)
        output = ""
        for token in llm.generate(
                tokens=tokens,
                top_k=json_request.get("top_k"),
                top_p=json_request.get("top_p"),
                temperature=json_request.get("temperature"),
                repetition_penalty=json_request.get("repetition_penalty"),
                last_n_tokens=json_request.get("last_n_tokens"),
                seed=json_request.get("seed"),
                batch_size=json_request.get("batch_size"),
                ):
            output = output + llm.detokenize(token)
        # output_json = json.dumps(output, ensure_ascii=False)
        print("generated output", output)
        return output

if __name__ == "__main__":
        app.run(host='0.0.0.0', port=8443, ssl_context=('./certs/fullchain.pem', './certs/privk
```

stablecode-server.py hosted with ❤️ by GitHub view raw

このサーバーを呼び出すAPEXアプリケーションを作って、コード生成を行ってみました。時間はかかりますが、それっぽいコードは出力します。



上記のアプリケーションのエクスポートを以下に置きました。
https://github.com/ujnak/apexapps/blob/master/exports/stablecode-generate.zip

完

Yuji N. 時刻: 16:51

共有

‹ ホーム ›

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

詳細プロフィールを表示