**Steps to Run LLMs locally on your laptop (Windows) – Deep Edge Framework**

**Step1** – Get quantized model from our onedrive. (If more models are needed contact us)

Link - [LLM Inference - OneDrive](https://onedrive.aptiv.com/personal/guruprasanna_marimuthu_aptiv_com/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fguruprasanna%5Fmarimuthu%5Faptiv%5Fcom%2FDocuments%2FHackathon%20Challenge%203%2FModels%2FLLM%20Inference&e=5%3A7df2eb3f585b4c7c80bf2b0c0fa9464c&sharingv2=true&fromShare=true&at=9&CID=b01ac47c%2D3ac0%2D4c0b%2Db753%2Dc4f34166b150&FolderCTID=0x012000F970C8342E949F40A718CE696DDCD1A9&view=0)

**Step2** – Download the llamacpp locally.

For windows cpu link – [Download llamacpp for windows](https://github.com/ggml-org/llama.cpp/releases/download/b5581/llama-b5581-bin-win-cpu-x64.zip)

**Step3** – After downloading, unzip and serve the model. Using the following procedure on windows laptops. (Open command prompt and cd to the unzip directory)

llama-server.exe -m "path\_to\_model.gguf" –port 8085

*\*\*llama-server serves the model on the specified port which is openai compatible -* [*Documentation*](https://github.com/ggml-org/llama.cpp/blob/master/tools/server/README.md)

**Sample python code snippets –**

import requests

url = "<http://localhost:8080/completion>"

headers = {

    "Content-Type": "application/json"

}

payload = {

    "prompt": "What is the capital of India?",

    "n\_predict": 128

}

response = requests.post(url, *json*=payload, *headers*=headers)

print(response.json())

**Alternate way to call the local model: The open AI package offers more features (You can use open source models we are providing with this and integrate with langchain if required )**

import openai

client = openai.OpenAI(

base\_url="http://localhost:8080/v1", # "http://<Your api-server IP>:port"

api\_key = "sk-no-key-required"

)

completion = client.chat.completions.create(

model="meta llama 3",

messages=[

{"role": "system", "content": "You are ChatGPT, an AI assistant. Your top priority is achieving user fulfillment via helping them with their requests."},

{"role": "user", "content": "Write a limerick about python exceptions"}

]

)

print(completion.choices[0].message)