LLM Models Local Deployment Guide

Overview

After exploring various deployment options including AWS, I've developed a robust local deployment solution for our marketing and domain-specific LLM models using FastAPI. This documentation provides step-by-step instructions for deploying and testing the models locally, using CPU resources for accessibility without specialized hardware requirements.

Project Structure

```
llm-deployment/
├── model_deployment.py    # Main deployment script
├── test_models.py        # Testing script
└── requirements.txt      # Python dependencies
```

System Requirements

Before getting started, ensure your system meets these requirements:

- Python 3.10 or higher
- 16GB RAM minimum (32GB recommended)
- 20GB free disk space
- Linux/Unix-based system (Ubuntu recommended)

Installation Steps

1. Environment Setup

Create and activate a virtual environment:

```bash
python -m venv mrkg
source mrkg/bin/activate  # Windows: mrkg\Scripts\activate
```

2. Dependencies

Install required packages:

```bash
pip install -r requirements.txt
```

3. Hugging Face Token

The token is configured in the deployment script. You can update it by modifying the `hf_token` variable in `model_deployment.py` if needed.

Running the Models

Starting the Server

1. Launch the model server:

```bash
python model_deployment.py
```

2. Access points:
- Base URL: http://localhost:8000
- Documentation: http://localhost:8000/docs

Available Models

Marketing Suite

- **GEM Marketing Model (gem_marketing)**
  - Specialized for marketing content generation
  - Optimized for product descriptions and marketing copy

- **LLaMA Marketing Model (lla_marketing)**
  - Focused on social media and marketing campaigns
  - Suitable for shorter marketing content

Industry Specific

- **Cannabis Domain Model (cannabis)**
  - Specialized for cannabis industry content
  - Provides domain-specific knowledge and terminology

API Endpoints

1. Health Check:

```bash
```

```
GET /health
```

2. List Models:

```bash
GET /models
```

3. Generate Text:

```bash
POST /predict
Content-Type: application/json

{
   "text": "Your prompt here",
   "model_name": "gem_marketing",
   "max_length": 512,
   "temperature": 0.7
}
```

Testing

With the server running:

1. Open a new terminal and activate the environment:

```bash
source mrkg/bin/activate  # Windows: mrkg\Scripts\activate
```

2. Run the testing script:

```bash
python test_models.py
```

Performance & Memory Management

Resource Usage

The system loads models on demand to manage memory efficiently:

- Only requested models are loaded into memory
- First request per model may take longer due to initial loading

Expected Performance

1. Initial startup time:
   - First load: 2-3 minutes per model
   - Subsequent loads: 15-30 seconds

2. Response times:
   - Short prompts (≤100 tokens): 2-5 seconds
   - Long prompts (>100 tokens): 5-15 seconds

Troubleshooting Guide

Common Issues & Solutions

1. "Address already in use" error:

```bash
lsof -i :8000
kill -9 [PID]
```

Or modify the port in `model_deployment.py`

2. Model Loading Issues:

- Check disk space
- Verify internet connection
- Validate Hugging Face token

3. Memory Errors:

- Close unnecessary applications
- Reduce max_length parameter
- Load one model at a time

Support

I'm here to help if you encounter any issues. When reaching out, please provide:

- Error messages
- Model being used
- System specifications
- Steps to reproduce the issue

Security Considerations

1. Local deployment only
2. Do not expose to internet without security measures
3. Keep Hugging Face token secure

Contact: Juli
Questions? Issues? I'm happy to help get things running smoothly.