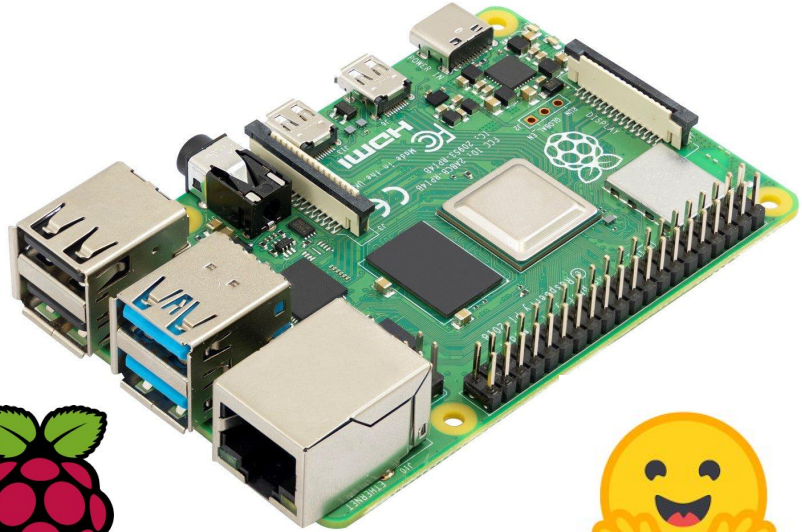


Despliegue de modelos de Lenguaje Natural en dispositivos con pocos recursos.

IoT

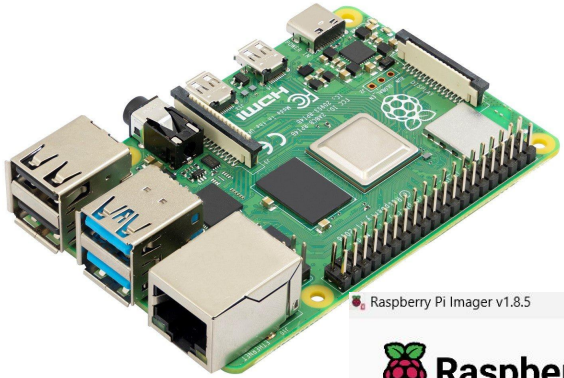
Francisco Durán, Isslam Benali, Oriol Catasus

20/06/2024

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Hardware



- **Raspberry Pi 4**
  - 4GB RAM
  - Storage: 64GB

Creación de API

FastAPI

Desarrollo de API

- API REST con FastAPI
  - Definición de endpoints
- Server con Uvicorn

FastAPI

```
https://localhost:8000/huggingface_models/pythia-410m/torch
```

https://madewithml.com/courses/mlops/api/

Desarrollo de API

```python
@app.post("/huggingface_models/pythia-410m/{engine}", tags=["Hugging Face Models"])
@construct_response
def _predict_pythia_410m(request: Request, payload: PredictPythia_410m, engine: str = None):
    """Codet5p_220m model."""

    input_text = payload.input_text
    print("Input text")
    print(input_text)

    model = Pythia_410mModel()
    print(f"Model: {model.name}")

    if input_text:
        prediction = model.predict(input_text, engine)

        response = {
            "message": HTTPStatus.OK.phrase,
            "status-code": HTTPStatus.OK,
            "data": {
                "model-type": model.name,
                "input_text": input_text,
                "prediction": prediction,
            },
        }
    else:
        response = {
            "message": "Model not found",
            "status-code": HTTPStatus.BAD_REQUEST,
        }
    return response
```

Desarrollo de API

```python
import uvicorn

if __name__ == "__main__":
    # uvicorn app.api_code:app  --host 0.0.0.0 --port 8000  #--reload   --reload-dir app
    uvicorn.run("app.api_code:app", port=8000, log_level="info")
```

Desarrollo de API

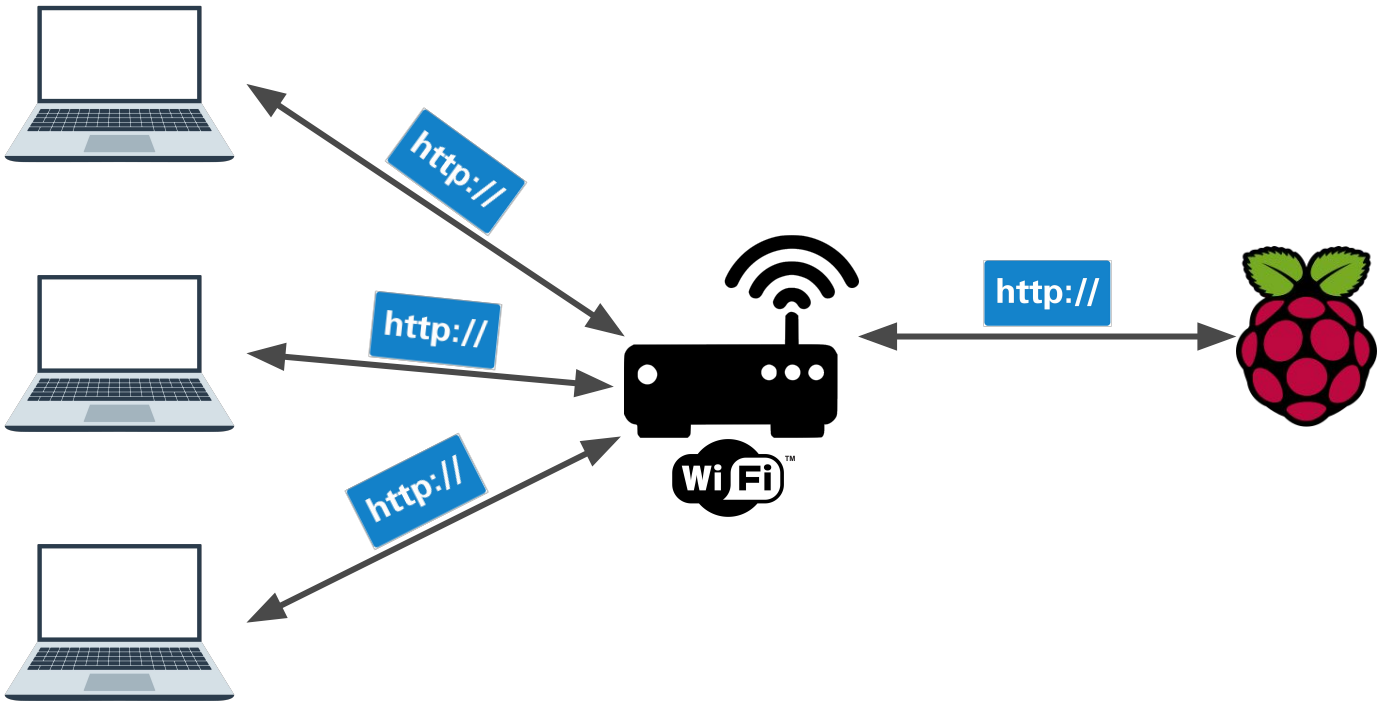
```
payload = {
            "input_text": line,
        }

response = requests.post(f"{url}{endpoints[model]}/{engine}"  , json=payload)
```

Conexión cliente-servidor

Conexión cliente-servidor

1. Conexión WiFi
   - Se configura la Raspberry para conectarse a nuestro AP
   - Conectamos el cliente al mismo AP
2. Deploy de nuestro código
   - Nos conectamos a la Raspberry mediante SSH
   - Descargamos nuestro código mediante GIT
   - Instalamos las dependencias necesarias
3. Ejecución:
   - Se configura uvicorn con la interfaz 0.0.0.0
   - El cliente realiza peticiones HTTP a la Raspberry

Conexión cliente-servidor

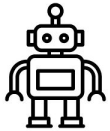


Aplicaciones

Seguridad y vigilancia

Educación

Monitoreo del medioambiente.



Robotica

Healthcare y asistencia.

Smart Home

Futuro

Sensores Inteligentes y Análisis en el Edge: El análisis en el edge permite que los dispositivos IoT procesen datos localmente, reduciendo la latencia y mejorando la eficiencia en tiempo real.

Avances en la Personalización de Modelos: Los modelos LLM y NLP serán más personalizados, aprendiendo y mejorando continuamente según los datos específicos de cada usuario.

Expansión en Nuevos Sectores y Aplicaciones: La tecnología de NLP en dispositivos de bajo recurso se expandirá a sectores como salud, agricultura y educación, abriendo nuevas oportunidades.

Interoperabilidad y Conectividad Mejoradas: La mejora en la conectividad y la interoperabilidad permitirá que los dispositivos colaboren eficazmente en redes de IoT.

Avances en Herramientas de Desarrollo y Plataforma: Nuevas herramientas y plataformas facilitarán el desarrollo y la implementación de soluciones de lenguaje natural en dispositivos con recursos limitados.

Oportunidades de Investigación y Desarrollo Futuro: La investigación se centrará en optimizar modelos, mejorar la seguridad de datos locales y explorar nuevas aplicaciones en sectores emergentes.

¡Gracias!

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB