

Deployment

딥러닝 서비스 배포 - 서버 처리상태 모니터링 (Progress Bar)

소프트웨어 끈대 강의

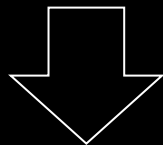
노기섭 교수

(kafa46@cju.ac.kr)

이전 강의를 참고하세요

딥러닝 배포 서버 구축

https://github.com/kafa46/acin_academy/tree/master/01_deployment



이전에 구축한 서버를 업그레이드 합니다.

Original Tutorial Source

“Make a Progress Bar With SocketIO | Flask Tutorial”

<https://youtu.be/bUfUKtJqaxQ>

References

■ Flask-SocketIO

- <https://flask-socketio.readthedocs.io/en/latest/index.html>

■ Flask-Script: 서버 실행 스크립트 manage.py 작성 지원

- <https://flask-script.readthedocs.io/en/latest/>

■ MDN Web Docs, ARIA (Accessible Rich Internet Application)

- <https://developer.mozilla.org/ko/docs/Web/Accessibility/ARIA>

■ Pingcode, Make a Progress Bar With SocketIO | Flask Tutorial

- https://youtu.be/bUfUKtJqaxQ?si=z_XpHC3pqLxduGhv

Course Prerequisites

PREREQUISITE

01. HTML/CSS/JavaScript

02. jQuery

03. Flask (web framework)

04. Concept of Asynchronous Processing

05. Of course, we need Python!!

06. Asynchronous Processing ← 추가

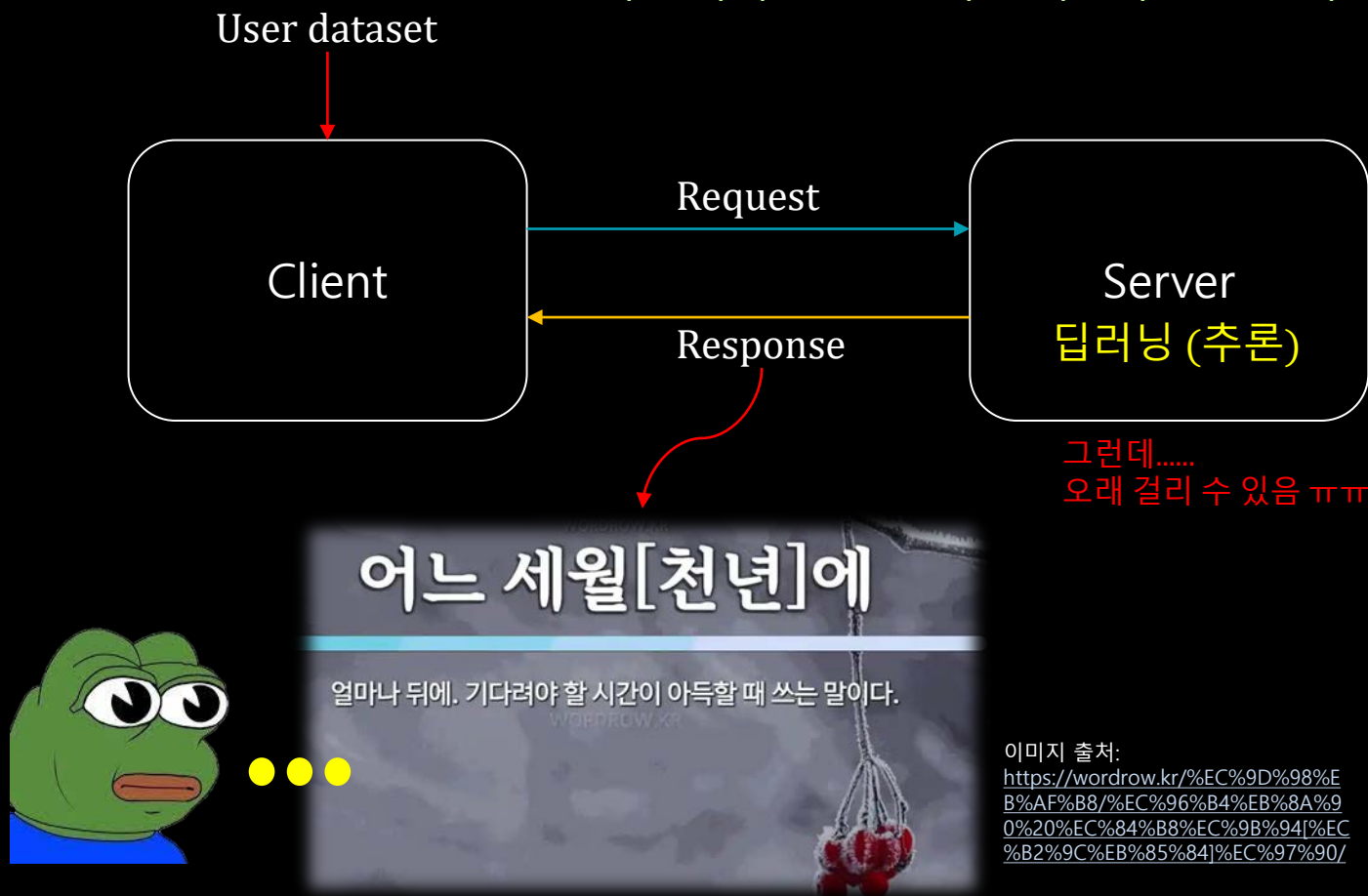
선수 과목

先修科目

배경지식이 필요하거나 실력을
단계적으로 향상시켜야 하는 이유로,
어떤 수업을 듣기 전에 미리 또는
순서대로 공부해야 하는 과목.

Motivation

딥러닝 서비스 - 웬만 하면 이런 구조를 씁니다.



Learning Objectives

사용자에게 진행 상황을 알려주자 ^^,

LOADING....

0%



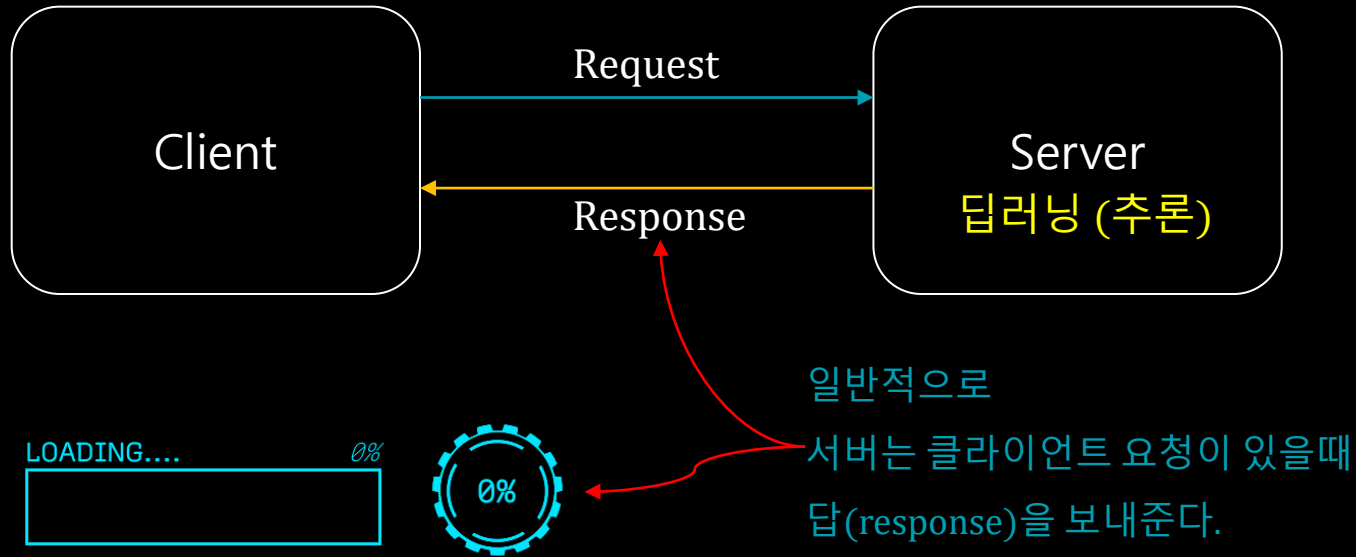
이미지 출처: <https://colab.research.google.com/drive/19fqGHdK33IBAlivQVt25zTajR1Ylhww4>

60.0%



이미지 출처: <https://www.pinterest.fr/pin/25-beautiful-loading-bar-design-examples-gif-animated--405886985147074894/>

The most difficult point

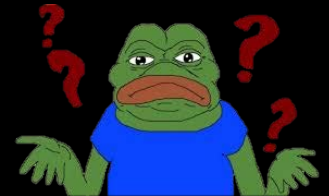
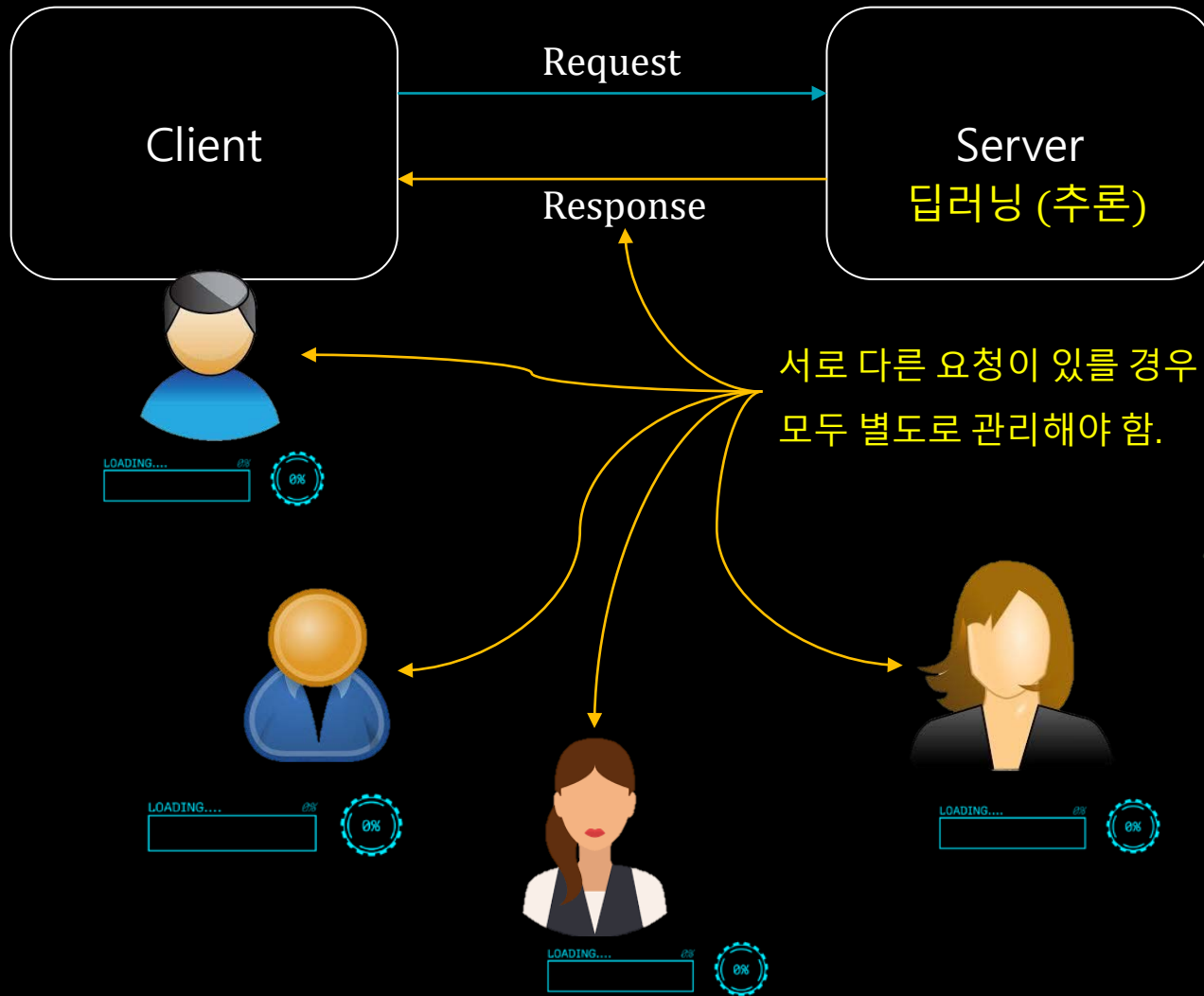


이미지 출처:
<https://colab.research.google.com/drive/19fqGHdK33IBAlivQVt25zTajR1Ylhww4>

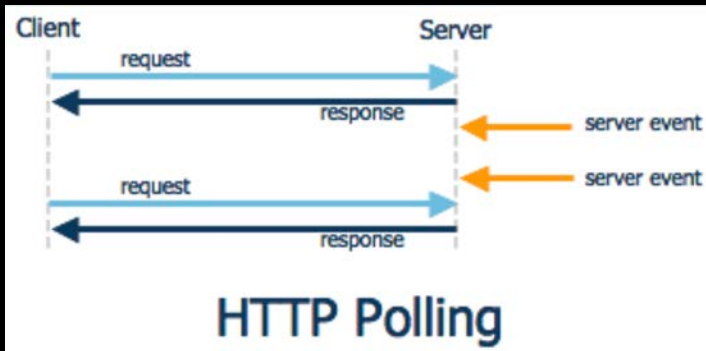


클라이언트는 요청을 보내지 않음
하지만 서버 처리 상태를 지속적으로 확인해야 함.

The most difficult point



One possible solution



Polling:
주기적으로 데이터를
업데이트하는 작업



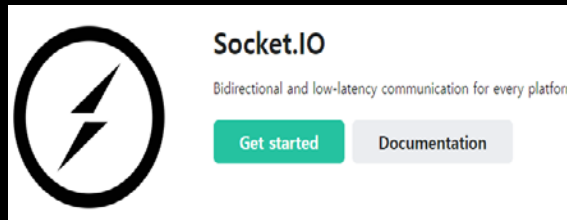
Real time 아님 $\pi\pi$
불필요한 Request (자원낭비) $\pi\pi$
주기가 짧을수록 네트워크 부하 커짐 $\pi\pi$

양방향 통신 (Full-Duplex) + 실시간 네트워킹 (Real Time Networking) +
+ 연결 관리 (Connection Management)



Official link
https://developer.mozilla.org/ko/docs/Web/API/WebSockets_API

HTML 5 표준기술
단순하고 빠름



Official link: <https://socket.io/>

Websocket 기술을 적용한
Node.js 라이브러리

우리는 요걸로 실습 ^^
간단한 사용법만 ~
자세한 것은 각자 공부



YouTube Tutorial: <https://youtu.be/7pGuwV4rwH0>

Transport/IP layer 직접 적용
(TCP/UDP)

How we attack

Image source:
<https://gurneyjourney.blogspot.com/2020/08/19/30s-toys-materials-and-worksmanship.html>



Toy Web Service
(Deeplearning
Application)

Front-end

HTML + CSS + JavaScript

jQuery

BootStrap

Socket.io



Back-end

Of course, Python!

Flask

Asynchronous
Processing



ACIN Academy

Navigation bar (for future use)

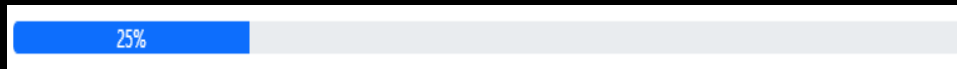
안녕하세요^^, 딥러닝 분석 시스템입니다.

분석할 데이터를 추가해 주세요

Multiple files input example

파일 선택 선택된 파일 없음

분석요청



서버의 분석 작업이 끝나면 progressbar 감추기
+ 서버에서 보내온 자료를 보여주기

Dependency 설치

■ Flask socket.io 모듈 설치

(venv) path/to/project\$ pip install flask-socketio

■ 비동기 처리 모듈 설치 (<https://flask.palletsprojects.com/en/3.0.x/async-await/>)

(venv) path/to/project\$ pip install flask[async]

■ Flask 서버 Manager 작성용 모듈 설치

- (venv) path/to/project\$ pip install Flask-Script

■ socket.io Include

- CDN: <https://socket.io/docs/v4/client-installation/#from-a-cdn>
- base.html에 추가

Front-end upgrade: main.html

■ Progressbar 추가

- Bootstrap: <https://getbootstrap.com/docs/4.0/components/progress/#labels>

```
<div class="progress" style="height: 30px;">
```

```
<div class="progress-bar progress-bar-striped  
progress-bar-animated
```

Options

```
role="progressbar"
```

```
style="width: 25%;"
```

```
aria-valuenow="25"
```

```
aria-valuemin="0"
```

```
aria-valuemax="100"
```

```
id="pb-status">25%<div>
```

```
</div>
```

JavaScript-powered
progress bar widget

width (length) of status

current state

minimum and maximum values
for the progress bar

Front-end upgrade

■ attach_file_handler.js 업그레이드

```
// 테스트 코드
$(function () {
    update_progressbar(95)
})

// Progressbar 업데이트 함수
function update_progressbar(percent) {
    let p_bar = $('#p_bar');
    p_bar.css('width', `${percent}%`);
    p_bar.attr('aria-valuenow', percent);
    p_bar.text(`${percent}%`);
}
```

JavaScript에 socket.io 코드 추가

■ attach_file_handler.js 파일에 socket.io 구현 코드 추가

```
// socket.io 구현
$(function(){
    const socket = io();
    socket.connect('http://192.168.0.16:5678');
    socket.on('connect', function(){
        console.log('success');
    });
});
```

Socket.io 적용을 위한 추가 코딩

■ run_server.py 작성

```
from server import create_app, socketio

server = create_app()

if __name__ == '__main__':
    socketio.run(app=server, debug=True, host='0.0.0.0', port='5678')
```

■ run.sh 수정

```
export FLASK_APP=server
export FLASK_DEBUG=True

# flask run -h 0.0.0.0 -p 5678
python run_server.py
```

run.sh 재실행 하여

→ 브라우저 콘솔에 'success' 메시지 확인

Back-end 코딩

■ __init__.py 수정

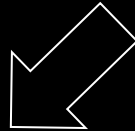
- socket.io 객체 생성
- app에 socketio 등록

■ main_views.py 수정

■ attach_file_handler.js 수정

```
// socket.io 추가 (수신 데이터 확인)
$(function(){
    const socket = io();
    socket.connect('http://192.168.0.16:5678/process');
    // socket.on('connect', function(){
    //     console.log('success');
    // });
    socket.on('process_status', function(percent){
        console.log(`Progress: ${percent}%`);
        update_progressbar(percent);
    });
});
```

```
# 비동기처리: async, await 적용해야 함
@bp.route('/process/', methods=['POST'])
async def process():
    '''딥러닝 요청에 대한 처리'''
    if request.method=='POST':
        :
        # 딥러닝 알고리즘 실행
        # 딥러닝 학습 -> epoch 10회 수행한다고 가정
        NUM_EPOCH = 10
        for x in range(10):
            # Socket.io 통신
            percent = (x+1)/NUM_EPOCH * 100
            socketio.emit('process_status', percent)
            await sleep(2)
        :
```



서버로 파일을 전송해서
progressbar 작동하는 지 확인

progressbar.js 조금 더 업데이트

- 서버 전송하기 클릭하면 나타나도록 설정
- 100% 전송이 되었을 경우
 - progressbar 숨기고,
 - 서버 처리 종료 결과창 보여주기

다중 사용자 처리하기

■ main_views.py 라우트 수정

```
@bp.route('/process/<socket_id>', methods=[ 'POST' ])
async def process(socket_id):
    :
    socketio.emit('process_status', percent, to=socket_id)
    :
```

■ attach_file_handler.js 수정

```
$(function(){
    const socket = io();
    let socket_id = undefined;
    socket.connect(`http://192.168.0.16:5678`);
    socket.on('connect', function(){
        socket_id = socket.id;
    });
    :
    $.ajax({
        method: 'POST',
        url: `/process/${socket_id}`,
```



수고하셨습니다 ..^^..