

# Chap.-7 Introduction to Deep Learning

## Network Sockets & Connections

- IP Address (Internet Protocol Address - unique address for device identification)
- Port Number (Logical Address - locate Service or specific place on device)
- Socket (Combination of IP Address & Port Number)
- TCP (Transmission Control Protocol)
  - connection oriented
  - reliable ordered delivery
  - guaranteed delivery
  - slower
  - Use Cases : web browsing, email, file transfer, etc.
- UDP (User Datagram Protocol)
  - connection less
  - less reliable and nor ordered delivery
  - no delivery guarantee
  - faster
  - Use Cases : Live Video Streaming, online gaming, VoIP (voice over IP)

```
In [1]: 1 # %%writefile TCP_Server.py
        2
        3 import socket as s
        4 host = s.gethostname()
        5 print("HostName :", host)
```

HostName : 408B-99

```
In [2]: 1 import socket as s
        2 host = s.gethostname()
        3 ip = s.gethostbyname(host)
        4 print("IP :", ip)
        5 print("HostName :", host)
```

IP : 192.168.107.99

HostName : 408B-99

```
In [ ]: 1 import socket as s
        2 host = s.gethostname()
        3 ip = s.gethostbyname('www.google.com')
        4 print("IP :", ip)
        5 print("HostName :", host)
```

```
In [4]: 1 import socket as s
2 host = s.gethostname()
3 ip = s.gethostbyname('localhost')
4 print("IP :", ip)
5 print("HostName :", host)
```

IP : 127.0.0.1  
HostName : 408B-99

```
In [9]: 1 %%writefile TCP_Server.py
2 # magic function of jupyter only
3
4 import socket as s
5 host = 'localhost'
6 port = 3600
7
8 sock = s.socket()
9 sock.bind((host,port)) # bind only take 1 parameter
10 sock.listen() # if pass no. in listen(6) define request - bydefault = 5
11 print('Server is Listening. . .')
12 conn, add = sock.accept()
13 print("Connection From :", str(add))
14
15 while True:
16     data = conn.recv(1024).decode()
17     if data=='CLOSE':
18         break
19     print("client data :",data)
20     msg = input("ServerData : ")
21     conn.send(msg.encode())
22 conn.close()
```

Overwriting TCP\_Server.py

```
In [10]: 1 %%writefile TCP_Client.py
2 import socket as s
3
4 csock = s.socket(s.AF_INET, S.SOCK_STREAM)
5 csock.connect(('localhost', 3600))
6 while True:
7     msg = input("Enter data for server : ")
8     csock.send(msg.encode())
9     if msg=='CLOSE':
10         break
11     data = csock.recv(1024).decode()
12     print("Received from Sever", data)
13 csock.close()
```

Writing TCP\_Client.py

```
In [ ]: 1
```

decode - bytes into a string ASKII encode - askii to bytes

```
In [11]: 1 import os
         2 os.getcwd()
```

```
Out[11]: 'C:\\\\Users\\\\LJENG\\\\Romil Monpara'
```

## Connect to other pc

```
In [ ]: 1 # TCP_Server.py
        2 # cell magic function of jupyter only
        3
        4 import socket as s
        5 host = '192.168.107.99'
        6 port = 3600
        7
        8 sock = s.socket()
        9 sock.bind((host,port))
       10 sock.listen() # if pass no. in listen(6) define request - bydefault = 5
       11 print('Server is Listening. . .')
       12 conn, add = sock.accept()
       13 print("Connection From :", str(add))
       14
       15 while True:
       16     data = conn.recv(1024).decode()
       17     if data=='CLOSE':
       18         break
       19     print("client data :",data)
       20     msg = input("ServerData : ")
       21     conn.send(msg.encode())
       22 conn.close()
```

```
In [ ]: 1 # TCP_Client.py
        2 import socket as s
        3
        4 csock = s.socket(s.AF_INET, s.SOCK_STREAM)
        5 csock.connect(('localhost', 3600))
        6 while True:
        7     msg = input("Enter data for server : ")
        8     csock.send(msg.encode())
        9     if msg=='CLOSE':
       10         break
       11     data = csock.recv(1024).decode()
       12     print("Received from Sever", data)
       13 csock.close()
```

- **User sends the number and server gives a cube.**

```
In [4]: 1 %%writefile TCP_Server.py
2 import socket as s
3
4 host = 'localhost' # Use 'localhost' to match the client
5 port = 3600
6
7 sock = s.socket()
8 sock.bind((host, port))
9 sock.listen()
10 print('Server is Listening. . .')
11 conn, addr = sock.accept()
12 print("Connection From :", str(addr))
13
14 while True:
15     data = conn.recv(1024).decode()
16     if data == 'CLOSE':
17         break
18     try:
19         num = int(data)
20         cube = num ** 3
21         print(f"Received number: {num}, Cube: {cube}")
22         conn.send(str(cube).encode())
23     except ValueError:
24         print("Invalid data received (not a number)")
25         conn.send("Invalid input".encode())
26
27 conn.close()
```

Overwriting TCP\_Server.py

```
In [5]: 1 %%writefile TCP_Client.py
2 import socket as s
3
4 csock = s.socket(s.AF_INET, s.SOCK_STREAM)
5 csock.connect(('localhost', 3600)) # Match the server's IP and port
6
7 while True:
8     msg = input("Enter number for server (or 'CLOSE' to exit): ")
9     csock.send(msg.encode())
10    if msg == 'CLOSE':
11        break
12    data = csock.recv(1024).decode()
13    print("Received from Server:", data)
14
15 csock.close()
```

Overwriting TCP\_Client.py

## • Enter Data convert into Capital

```
In [10]: 1 %%writefile TCP_Server.py
2 import socket as s
3
4 host = 'localhost' # Use 'localhost' to match the client
5 port = 3600
6
7 sock = s.socket()
8 sock.bind((host, port))
9 sock.listen()
10 print('Server is Listening. . .')
11 conn, addr = sock.accept()
12 print("Connection From :", str(addr))
13
14 while True:
15     data = conn.recv(1024).decode()
16     if data == 'CLOSE':
17         break
18     d = data.upper()
19     print(f"Received data: {data}, Converted into Capital: {d}")
20     conn.send(str(d).encode())
21 conn.close()
```

Overwriting TCP\_Server.py

```
In [9]: 1 %%writefile TCP_Client.py
2 import socket as s
3
4 csock = s.socket(s.AF_INET, s.SOCK_STREAM)
5 csock.connect(('localhost', 3600)) # Match the server's IP and port
6
7 while True:
8     msg = input("Enter data for server (or 'CLOSE' to exit): ")
9     csock.send(msg.encode())
10     if msg == 'CLOSE':
11         break
12     data = csock.recv(1024).decode()
13     print("Received from Server:", data)
14
15 csock.close()
```

Overwriting TCP\_Client.py

## UDP Server

```
In [17]: 1 %%writefile UDP_Server.py
2 import socket as s
3 host = 'localhost'
4 port = 3700
5
6 sock = s.socket(type=s.SOCK_DGRAM) # this argument is compulsory in UDP
7 sock.bind((host,port))
8 print("Server is active")
9
10 while True:
11     data,addr = sock.recvfrom(1024)
12     data = data.decode()
13     if data == 'CLOSE':
14         break
15     print('Client :', data)
16     msg = input('Server : ')
17     sock.sendto(msg.encode(),addr)
18 sock.close()
```

Overwriting UDP\_Server.py

```
In [21]: 1 %%writefile UDP_Client.py
2 import socket as s
3 host = '192.168.107.98'
4 port = 3700
5
6 sock = s.socket(type=s.SOCK_DGRAM)
7 addr = (host,port)
8
9 while True:
10     data = input("Client : ")
11     sock.sendto(data.encode(), addr)
12     if data=='CLOSE':
13         break
14     data,addr = sock.recvfrom(1024)
15     print("Server :", data.decode())
16 sock.close()
```

Overwriting UDP\_Client.py

```
In [ ]: 1 Server active and listening...
2 Received from ('192.168.107.99', 58887): hey
3 Data to send: hi
4 Received from ('192.168.107.97', 61067): hello
5 Data to send: hello
6 Received from ('192.168.107.99', 58887): yooo
7 Data to send: yee
8 Received from ('192.168.107.97', 61067): soo ja bhai
9 Data to send: hi
10 Received from ('192.168.107.99', 58887): CLOSE
11 Data to send:
12
13 # from my pc
14 C:\Users\LJENG\Romil Monpara>python UDP_Client.py
15 Client : hey
16 Server : hi
17 Client : yoo
18 Server : yee
19 Client : CLOSE
```

```
In [23]: 1 import socket as sock
2 sock = socket.socket()
3 sock.connect(('www.ljku.edu.in',80))
4 cmd = "GET https://www.ljku.edu.in/lju-at-a-glance HTTP/1.0\n\n".encode()
5 sock.send(cmd)
6
7 while True:
8     data = sock.recv(1024)
9     if len(data)<1:
10         break
11     print(data.decode(), end=' ')
12 sock.close()
```

```
HTTP/1.1 301 Moved Permanently
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 29 May 2025 05:10:01 GMT
Content-Type: text/html
Content-Length: 178
Connection: close
Location: https://ljku.edu.in (https://ljku.edu.in)
```

```
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

## API

```
In [4]: 1 import requests,json
2 key = '7c0dc741e11d10cca8eddba68796b8e3'
3 city = input("City Name : ")
4 url = f"http://api.openweathermap.org/geo/1.0/direct?q={city}&appid={key}"
5 response = requests.get(url)
6 coord = response.json()
7 print(coord)
8 print(json.dumps(coord,indent=5))
```

City Name : Junagadh

```
[{'name': 'Junagadh', 'local_names': {'ar': 'جوناغاد', 'ur': 'جونہ گڑھ', 'hi': 'जूनागढ़', 'ml': 'ജൂനാഗഡ്', 'ta': 'ஜூனாகட', 'he': 'יוֹנָאגָד', 'tr': 'Cünâgerh', 'pa': 'ਜੂਨਾਗੜ੍ਹ', 'te': 'జూనాగఢ్', 'ja': 'ジュナーガド', 'en': 'Junagadh', 'gu': 'જૂનાગઢ', 'kn': 'ಜುನಗಡ್'}, 'lat': 21.5174104, 'lon': 70.4642754, 'country': 'IN', 'state': 'Gujarat'}]
```

```
[
  {
    "name": "Junagadh",
    "local_names": {
      "ar": "\u062c\u0648\u0627\u062f\u0627\u0627\u0627\u0627\u0627",
      "ur": "\u062c\u0648\u0627\u062f \u0627\u0627\u0627\u0627",
      "hi": "\u091c\u0942\u0928\u093e\u0917\u0922\u093c",
      "ml": "\u0d1c\u0d41\u0d28\u0d3e\u0d17\u0d21\u0d4d",
      "ta": "\u0b9c\u0b2\u0ba9\u0bbe\u0b95\u0ba4\u0bcd",
      "he": "\u05d2'\u05d5\u05e0\u05d2\u05d0\u05d3",
      "tr": "C\u00f0fc\u00e2gerh",
      "pa": "\u0a1c\u0a42\u0a28\u0a3e\u0a17\u0a5c\u0a4d\u0a39",
      "te": "\u0c1c\u0c41\u0c28\u0c3e\u0c17\u0c22\u0c4d",
      "ja": "\u30b8\u30e5\u30fc\u30c9\u30c9\u30c9\u30c9",
      "en": "Junagadh",
      "gu": "\u0a9c\u0ac1\u0aa8\u0abe\u0a97\u0aa2",
      "kn": "\u0c9c\u0cc1\u0ca8\u0c97\u0ca6\u0ccd"
    },
    "lat": 21.5174104,
    "lon": 70.4642754,
    "country": "IN",
    "state": "Gujarat"
  }
]
```



```
In [43]: 1 import requests, json
2
3 key = '7c0dc741e11d10cca8eddba68796b8e3'
4 city = input("City Name: ")
5
6 url = f"http://api.openweathermap.org/geo/1.0/direct?q={city}&appid={key}"
7 response = requests.get(url)
8 coord = response.json()
9 print(coord)
10 print(json.dumps(coord, indent=5))
11
12 lat = coord[0]['lat']
13 lon = coord[0]['lon']
14
15 url = f"https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}"
16 response = requests.get(url)
17 weather = response.json()
18
19 print(weather)
20 print(json.dumps(weather, indent=5))
```

City Name: Junagadh

```
{'coord': {'lon': 70.4579, 'lat': 21.5222}, 'weather': [{'id': 804, 'main':
'Clouds', 'description': 'overcast clouds', 'icon': '04d'}], 'base': 'station
s', 'main': {'temp': 307.78, 'feels_like': 310.27, 'temp_min': 307.78, 'temp_
max': 307.78, 'pressure': 1004, 'humidity': 42, 'sea_level': 1004, 'grnd leve
l': 992}, 'visibility': 10000, 'wind': {'speed': 9.06, 'deg': 235, 'gust': 9.
85}, 'clouds': {'all': 98}, 'dt': 1748677570, 'sys': {'country': 'IN', 'sunri
se': 1748651736, 'sunset': 1748699785}, 'timezone': 19800, 'id': 1268773, 'na
me': 'Jūnāgadh', 'cod': 200}
{
  "coord": {
    "lon": 70.4579,
    "lat": 21.5222
  },
  "weather": [
    {
      "id": 804,
      "main": "Clouds",
      "description": "overcast clouds",
      "icon": "04d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 307.78,
    "feels_like": 310.27,
    "temp_min": 307.78,
    "temp_max": 307.78,
    "pressure": 1004,
    "humidity": 42,
    "sea_level": 1004,
    "grnd_level": 992
  },
  "visibility": 10000,
  "wind": {
    "speed": 9.06,
    "deg": 235,
    "gust": 9.85
  },
  "clouds": {
    "all": 98
  },
  "dt": 1748677570,
  "sys": {
    "country": "IN",
    "sunrise": 1748651736,
    "sunset": 1748699785
  },
  "timezone": 19800,
  "id": 1268773,
  "name": "J\u016bn\u0101gadh",
  "cod": 200
}
```

```
In [16]: 1 import requests,json
2 key='7c0dc741e11d10cca8eddba68796b8e3'
3 city=input('city name ')
4 url=f"https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&
5 response=requests.get(url)
6 coord=response.json()
7 print(coord)
8 # print(json.dumps(coord,indent=5))
9 print("\n")
10 # lat=coord[0]['lat']
11 # print("\n")
12 # lon=coord[0]['lon']
13 # print(lat)
14 # print(lon)
15 description=coord['weather'][0]['description']
16 print(description)
17 temprature=coord['main']['temp']
18 print(temprature)
19 pressure=coord['main']['pressure']
20 print(pressure)
21 humidity=coord['main']['humidity']
22 print(humidity)
23 wind_speed=coord['wind']['speed']
24 print(wind_speed)
25 visibility=coord['visibility']
26 print(visibility)
```

```
city name Junagadh
{'coord': {'lon': 70.4579, 'lat': 21.5222}, 'weather': [{'id': 804, 'main': 'Clouds', 'description': 'overcast clouds', 'icon': '04d'}], 'base': 'stations', 'main': {'temp': 306.63, 'feels_like': 309.82, 'temp_min': 306.63, 'temp_max': 306.63, 'pressure': 1004, 'humidity': 48, 'sea_level': 1004, 'grnd_level': 993}, 'visibility': 10000, 'wind': {'speed': 8.78, 'deg': 239, 'gust': 9.84}, 'clouds': {'all': 98}, 'dt': 1748673969, 'sys': {'country': 'IN', 'sunrise': 1748651736, 'sunset': 1748699785}, 'timezone': 19800, 'id': 1268773, 'name': 'Jūnāgadh', 'cod': 200}
```

```
overcast clouds
306.63
1004
48
8.78
10000
```

```

In [18]: 1 himport requests,json
          2 import pandas as pd
          3 key="7c0dc741e11d10cca8eddba68796b8e3"
          4 lat= 23.0216
          5 lon= 72.5797
          6 D={"date_time":[],"temp":[],"pressure":[],"humidity":[],"weather":[]}
          7 url=f"http://api.openweathermap.org/data/2.5/forecast?lat={lat}&lon={lon}&"
          8 response=requests.get(url)
          9 coord=response.json()
          10 for i in coord["list"]:
          11     D["date_time"].append(i["dt_txt"])
          12     D["temp"].append(i["main"]["temp"])
          13     D["pressure"].append(i["main"]["pressure"])
          14     D["humidity"].append(i["main"]["humidity"])
          15     D["weather"].append(i["weather"][0]["description"])
          16 # print(coord)
          17 # print(json.dumps(coord,indent=3))
          18 data=pd.DataFrame(D)
          19 print(data[data["weather"]=="broken clouds"])

```

	date_time	temp	pressure	humidity	weather
2	2025-05-31 15:00:00	309.46	1001	31	broken clouds
3	2025-05-31 18:00:00	305.50	1003	45	broken clouds
12	2025-06-01 21:00:00	302.55	1001	62	broken clouds
15	2025-06-02 06:00:00	304.79	1003	49	broken clouds
18	2025-06-02 15:00:00	305.99	1000	49	broken clouds
22	2025-06-03 03:00:00	304.89	1004	49	broken clouds

```

In [35]: 1 import requests,json
          2 import pandas as pd
          3 key="7c0dc741e11d10cca8eddba68796b8e3"
          4 lat= 23.0216
          5 lon= 72.5797
          6 D={"date_time":[],"temp":[],"pressure":[],"humidity":[],"weather":[]}
          7 url=f"http://api.openweathermap.org/data/2.5/forecast?lat={lat}&lon={lon}&"
          8 response=requests.get(url)
          9 coord=response.json()
          10 for i in coord["list"]:
          11     D["date_time"].append(i["dt_txt"])
          12     D["temp"].append(i["main"]["temp"])
          13     D["pressure"].append(i["main"]["pressure"])
          14     D["humidity"].append(i["main"]["humidity"])
          15     D["weather"].append(i["weather"][0]["description"])
          16
          17 data=pd.DataFrame(D)
          18 print(data[((data["pressure"]>1000) & (data["humidity"]>60)| (data["weathe

```

	date_time	temp	pressure	humidity	weather
2	2025-05-31 15:00:00	309.46	1001	31	broken clouds
3	2025-05-31 18:00:00	305.50	1003	45	broken clouds
5	2025-06-01 00:00:00	301.54	1003	68	few clouds
12	2025-06-01 21:00:00	302.55	1001	62	broken clouds
13	2025-06-02 00:00:00	301.68	1001	66	scattered clouds
15	2025-06-02 06:00:00	304.79	1003	49	broken clouds
18	2025-06-02 15:00:00	305.99	1000	49	broken clouds
22	2025-06-03 03:00:00	304.89	1004	49	broken clouds
37	2025-06-05 00:00:00	302.37	1005	65	moderate rain

```
In [37]: 1 import requests,json
2 import pandas as pd
3 key="7c0dc741e11d10cca8eddba68796b8e3"
4 lat= 23.0216
5 lon= 72.5797
6 D={"date_time":[],"temp":[],"pressure":[],"humidity":[],"weather":[]}
7 url=f"http://api.openweathermap.org/data/2.5/air_pollution/history?lat={lat}&lon={lon}"
8 response=requests.get(url)
9 coord=response.json()
10 print(json.dumps(coord,indent=5))
```

```
{
  "coord": {
    "lon": 72.5797,
    "lat": 23.0216
  },
  "list": [
    {
      "main": {
        "aqi": 5
      },
      "components": {
        "co": 747.68,
        "no": 0,
        "no2": 21.59,
        "o3": 97.28,
        "so2": 23.84,
        "pm2_5": 70.46,
        "pm10": 81.04,
        "nh3": 14.82
      }
    }
  ]
}
```

```

In [44]: 1 import requests, json
          2 import pandas as pd
          3 from datetime import datetime
          4
          5 # API Setup
          6 key = "7c0dc741e11d10cca8eddba68796b8e3"
          7 lat = 23.0216
          8 lon = 72.5797
          9
         10 # Request data
         11 url = f"http://api.openweathermap.org/data/2.5/air_pollution/history?lat={
         12 response = requests.get(url)
         13
         14 # Parse JSON response
         15 data = response.json()
         16
         17 # Convert to DataFrame
         18 rows = []
         19 for item in data.get("list", []):
         20     timestamp = datetime.utcfromtimestamp(item["dt"])
         21     aqi = item["main"]["aqi"]
         22     comp = item["components"]
         23     comp["date_time"] = timestamp
         24     comp["aqi"] = aqi
         25     rows.append(comp)
         26
         27 df = pd.DataFrame(rows)
         28
         29 # Show first few rows
         30 print(df.head())
         31
         32 # Optional: Save to CSV
         33 # df.to_csv("air_pollution_data.csv", index=False)
         34

```

	co	no	no2	o3	so2	pm2_5	pm10	nh3	date_time
\									
0	747.68	0.0	21.59	97.28	23.84	70.46	81.04	14.82	2020-11-27 15:00:00
1	734.33	0.0	18.17	100.14	21.22	72.29	82.97	14.57	2020-11-27 16:00:00
2	707.63	0.0	15.42	105.86	19.31	77.10	87.09	13.30	2020-11-27 17:00:00
3	640.87	0.0	12.51	110.15	17.88	77.92	86.42	11.02	2020-11-27 18:00:00
4	560.76	0.0	9.68	113.01	17.17	73.69	80.47	9.12	2020-11-27 19:00:00

  

	aqi
0	5
1	5
2	5
3	5
4	5

```

In [ ]: 1

```

