Training Data:

{

"intents": [

{

"tag": "greeting",

"patterns": [

"hi",

"hey",

"hello",

"is anyone there?","hellooooo","hiii"

],

"responses": [

"hey",

"hi",

"hello",

"heyyy",

"hi, how can I help you?"

]

},

{

"tag":"basic",

"patterns":["hi,how are you?","how are you?"],

"responses":["i am fine , by the way you can ask me any queries related to the food delivery application"]

},

{

"tag":"fine",

"patterns":["i am fine,","fine","yes, i am fine bro"],

"responses":["glad to hear it, so you can ask me any help related to the food delivery "]

},

{

"tag":"food",

"patterns":["what is your favourite food","favourite food of yours?","favourite food","what's you favourite food"],

"responses":["i dont know your favourite food by mine is chicken biryani...order it using our food delivery "]

},

{

"tag": "goodbye",

"patterns": [

"thank you",

"goodbye",

"thanks",

"bye",

"see you later",

"thanks for your help",

"byeeee"

],

"responses": [

"sure, text me if you need any help",

"bye, see you later",

"byeee come back again",

"text me if you need any help"

]

},

{

"tag": "cusine type",

"patterns": [

"what are the foods available",

"what type of food is this",

"is my food vegan?",

"what type of food is available",

"what are the foods available?","what food is there?","food?"

],

"responses": [

"italian, indian,continental,dhabha,pizza,burger,vegan an other dishes",

"parotta,briyani,pizza,burger,variety rice and many other dishes available",

"pure veg or non veg?"

]

},

{

"tag":"no",

"patterns":["no","i can't","not","not okay"],

"responses":["sorry sir,i will try my best to provide you best results, ask me any other help regarding this application...thank you!!"]

},

{

"tag":"non-veg",

"patterns":[

"chicken",

"non veg",

"fish",

"mutton",

"chicken briyani",

"mutton briyani",

"non vegetarian"

],

"responses":[

"we would suggest chicken briyani,fish kuzhambu,mutton chukka,chicken 65,chicken tandoori adn many other dishes you would like order it using the app...thank you"

]

},

{ "tag":"veg",

"patterns":["vegetarian",

"veg",

"curd rice",

"sambar rice",

"sambar",

"rice",

"paneer","mushroom"

],

"responses":[

"we would suggest sambhar rice,curd rice, paneer tikka,paneer butter masala,mushroom gravy and many other dishes order it using the application....thank you"

]

},

{

"tag":"ok",

"patterns":["ok","yes","done","okay"],

"responses":["do you need any other help?","ask me if you need any other help"]

},

{ "tag":"continental",

"patterns":["only continental",

"continental please",

"continental foods",

"continental","italian","italian food","italian dishes","continental dishes",

"provide me best continental food",

"i like to eat continental dishes","i need continetnal food"

],

"responses":[ "order in restaurants like mcd,kfc,locofeast,BISTRO,italian delight as they provide better continental and italian dishes.order it using the application thank you!!!"]

},

{ "tag":"indian",

"patterns":["indian food","indian","dhabha","indian dishes","dhabha style","dhabha food"],

"responses":["we would suggest you A2B,Pind,dhabha makkani are the highly rated indian food restaurants kindly order through our application thank you!!"]

},

{ "tag":"pizza",

"patterns":["i like to eat pizza",

"pizza?","i like pizza","i like to order pizza","provide me best restaurants for pizza"],

"responses":["as you wish domino's,pizza hut,oven fresh and oven story are the highly rated restaurants suggested for you to order pizzas. order it using our application thank you!!!"]

},

{ "tag":"burger",

"patterns":["burger","burger?","i like to eat burger","burgers please","provide best restaurant for burger please"],

"responses":["as you wish MCD,burger king ,KFC are te top rated burger restaurants. order it using our application thank you!!!"]

},

{ "tag":"vegan",

"patterns":["i am a vegan","i want vegan restaurants","i need vegan type of foods","vegan food","vegan","what vegan food is avilable","only vegan foods"],

"responses":["The vegan foods are available but we dont have separate restaurants for that ...sorry"]

},

{

"tag":"other\_dishes",

"patterns":["other dishes","suggest me different dishes","some other dishes"],

"responses":["i cant suggest you dishes instead i will recommend some hotels you might like cholan mess,amman mess,rasavard,thalapkatti,A2B,pind,BBQ,samco,zaitoon,etc. these are the higly rated restaurants that are available kindly order using our application thank you!!"]

},

{

"tag":"biriyani",

"patterns":["biriyani","chicken briyani","mutton briyani","i would like to eat biriyani","suggest me best hotel for biriyani"],

"responses":["buhari,star,cholan mess,thalapakatti,samco,zaitoon are the highly rated biriyani restaurants. order it using our application thank you!!!"]

},

{

"tag":"parotta",

"patterns":["parotta?","i would like to eat parotta","i like parotta"],

"responses":["bhai kadai,zaitoon,cholan mess,madurai bun parotta are the highly rated parotta restaurants use our application to order the food thank you!!!!"]

},

{

"tag":"fish",

"patterns":["fish","fish kuzhambu","fish curry","fish fry","i would like to eat fish"],

"responses":["kumarakon,amman mess,poluchcha meen,thoondil are the highly rated restaurant for fish order it using our application thank you!!!"]

},

{

"tag": "order\_tracking",

"patterns": [

"where is my order?",

"when will my order dispatch?",

"at what time will I receive my order?",

"when will I receive my order",

"where is my food?"

],

"responses": [

"your order is on the way",

"you will receive your order soon",

"your order is already dispatched",

"your order will be dispatched soon"

]

},

{

"tag": "payment",

"patterns": [

"can I pay through online?",

"I can only pay through cash",

"payment?"

],

"responses": [

"sure, you can pay through any mode of payment",

"yes, we accept any kind of payment procedure until it's monopoly money...lol",

"yes, we accept COD and online payment also"

]

},

{

"tag": "refund",

"patterns": [

"I want my money back",

"refund my money",

"I want my refund",

"when will I get my money back",

"where is my money?",

"refund?"

],

"responses": [

"sure, we will send your money back within 2-3 days",

"sure, I will refund your money",

"your refund money is being processed"

]

},

{

"tag":"BASICS",

"patterns":["what are you doing?","what doing?"],

"responses":["my work is to provide you queries related to food delivery "]

},

{

"tag":"celebrities",

"patterns":["do you know vijay","do you know dhoni","vijay","dhoni","ms dhoni","rajini","do you know rajini","do you know?","thalapathy","thala","aandavar","kamal","pushpa"],

"responses":["yes i know him he is my idol he used to order in our application everytime!!"]

},

{

"tag":"refunding",

"patterns":["how to refund?","i dont know how to make refund "],

"responses":["its very simple, just text me refund the refund will start processing if you have ...thank you!!"]

}

]

}

This is the dataset that we have used to train our chatbot for food delivery application.

Stemming,tokenization,bag of words:

import nltk

from nltk.stem.porter import PorterStemmer

import numpy as np

#nltk.download('punkt')

stemmer=PorterStemmer()

def tokenize(sentence):

    return nltk.word\_tokenize(sentence)

def stem(word):

    return stemmer.stem(word.lower())

def bag\_of\_words(tokenized\_sentence,all\_words):

    tokenized\_sentence =[stem(w)for w in tokenized\_sentence]

    bag=np.zeros(len(all\_words),dtype= np.float32)

    for idx,w in enumerate(all\_words):

        if w in tokenized\_sentence:

            bag[idx]=1.0

    return bag

This code is used to tokenizing ,stemming, collecting the bag of words

Tokenizing: This is the process of splitting the entire sentence to separate words, it is used to separate the predefined words or meaningful words .

Stemming: This is the process of changing the same word to different word by adding additional letter at the end of the word and by removing the letter at the end of the word.

Bag of words: This method is used to collect all these tokenized and stemming words and store it in an array.

Neural Network:

import torch

import torch.nn as nn

class NeuralNet(nn.Module):

    def \_\_init\_\_(self,input\_size,hidden\_size,num\_classes):

        super(NeuralNet,self).\_\_init\_\_()

        self.l1=nn.Linear(input\_size,hidden\_size)

        self.l2=nn.Linear(hidden\_size,hidden\_size)

        self.l3=nn.Linear(hidden\_size,num\_classes)

        self.relu=nn.ReLU()

    def forward(self,x):

        out=self.l1(x)

        out=self.relu(out)

        out=self.l2(out)

        out=self.relu(out)

        out=self.l3(out)

        #no activation and no softmax

        return out

This code is saved as a new file and it is used to create a Neural Network.

This neural network has 3 layers and the hidden\_layers will be specified at the method calling.

Training model:

import json

import os

from chatbot import tokenize, stem, bag\_of\_words

import numpy as np

import torch

import torch.nn as nn

from torch.utils.data import Dataset,DataLoader

from long\_responses import NeuralNet

# Change the working directory to where the JSON file is located

os.chdir('C:/Users/arara/OneDrive/Desktop/chatbot/')

# Now you can open the file using its relative path

with open('intents.json', 'r') as f:

    intents = json.load(f)

    print(intents)

all\_words = []

tags = []

xy = []

for intent in intents['intents']:

    tag = intent['tag']

    tags.append(tag)

    for pattern in intent['patterns']:

        w = tokenize(pattern)

        all\_words.extend(w)

        xy.append((w, tag))

ignore\_words = ['?', '!', '.', ',']

all\_words=[stem(w) for w in all\_words if w not in ignore\_words]

all\_words=sorted(set(all\_words))

tags=sorted(set(tags))

x\_train=[]

y\_train=[]

for(pattern\_sentence,tag) in xy:

    bag=bag\_of\_words(pattern\_sentence,all\_words)

    x\_train.append(bag)

    label=tags.index(tag)

    y\_train.append(label) #crossEntropyloss

x\_train=np.array(x\_train)

y\_train=np.array(y\_train)

y\_train = torch.LongTensor(y\_train)

class ChatDataset(Dataset):

    def \_\_init\_\_(self) :

        self.n\_samples=len(x\_train)

        self.x\_data=x\_train

        self.y\_data=y\_train

    def \_\_getitem\_\_(self,index):

        return self.x\_data[index], self.y\_data[index]

    def \_\_len\_\_(self):

        return self.n\_samples

batch\_size=8

hidden\_size=8

output\_size=len(tags)

input\_size= len(x\_train[0])

learning\_rate=0.001

num\_epochs=1000

print(input\_size, len(all\_words))

print(output\_size,tags)

dataset=ChatDataset()

train\_loader = DataLoader(dataset=dataset, batch\_size=batch\_size, shuffle=True, num\_workers=0)

model=NeuralNet(input\_size, hidden\_size,output\_size)

device=torch.device('cuda'if torch.cuda.is\_available() else 'cpu')

model= NeuralNet(input\_size,hidden\_size,output\_size).to(device)

criterion =nn.CrossEntropyLoss()

optimizer= torch.optim.Adam(model.parameters(),lr=learning\_rate)

for epoch in range(num\_epochs):

    for (words, labels) in train\_loader:

        words = words.to(device)

        labels = labels.to(device)

        outputs = model(words)

        loss = criterion(outputs, labels)

        optimizer.zero\_grad()

        loss.backward()

        optimizer.step()

    if (epoch + 1) % 100 == 0:

        print(f'epoch {epoch + 1}/{num\_epochs}, loss={loss.item():.4f}')

print(f'final loss, loss={loss.item():.4f}')

data={

    "model\_state":model.state\_dict(),

    "input\_size":input\_size,

    "output\_size":output\_size,

    "hidden\_size":hidden\_size,

    "all\_words":all\_words,

    "tags":tags

}

FILE="data.pth"

torch.save(data,FILE)

print(f'training complete. file saved to {FILE}')

This code is used to create a Pth file at the end and it contains different set of words as we call the tokenize, stemming and bad\_of\_words.

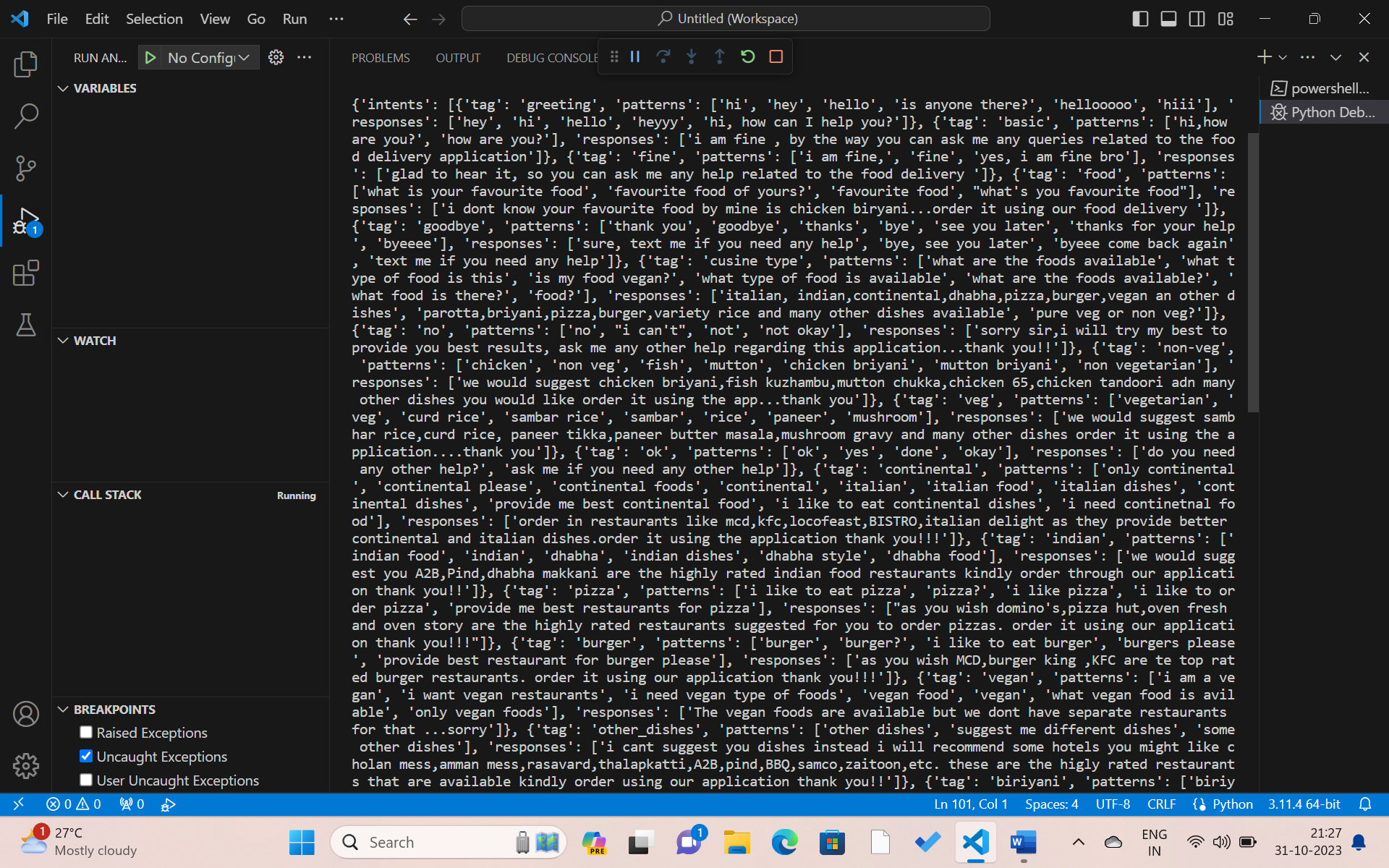
We will use the Neural Network to train these data and save it in the data.pth file .

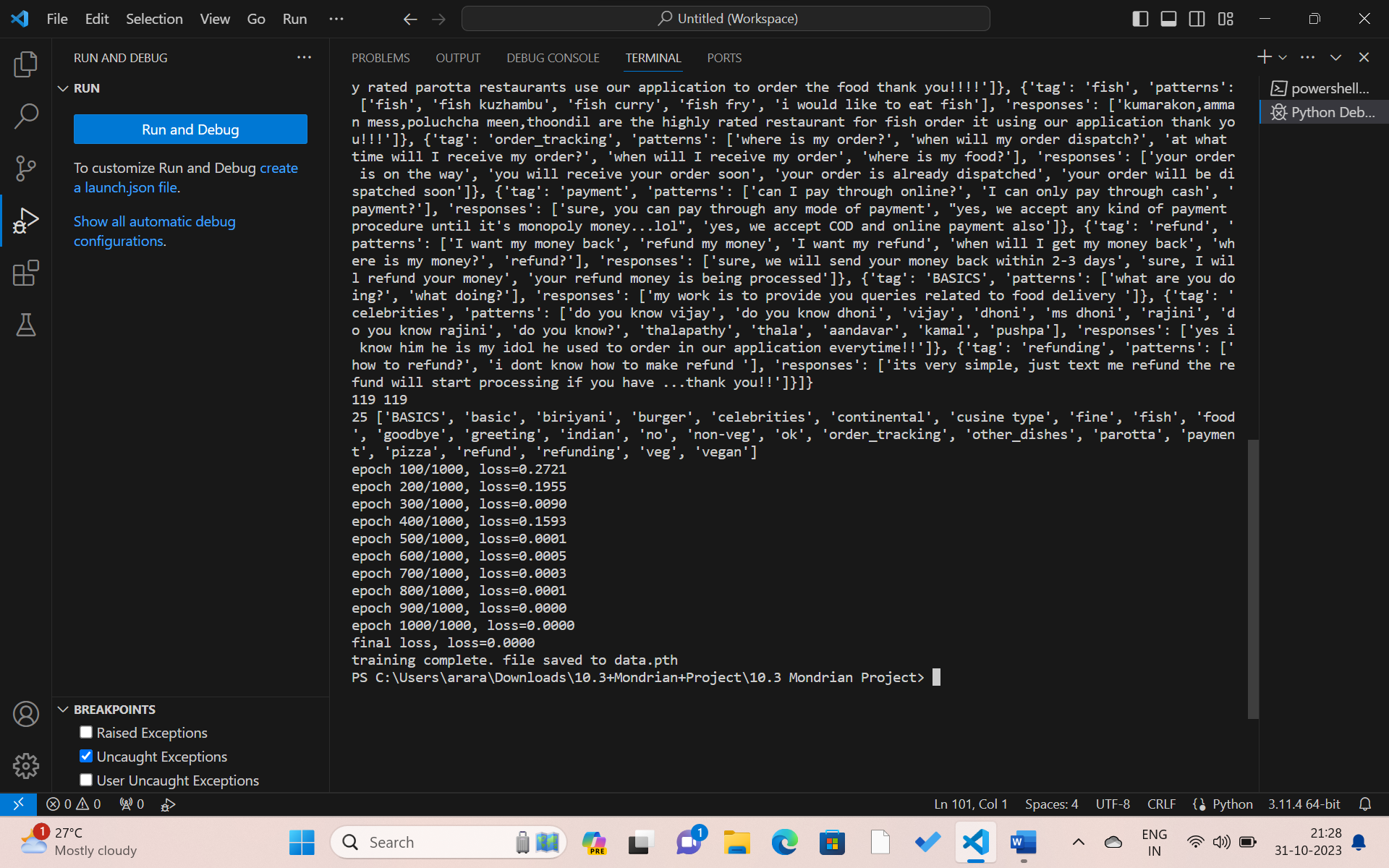
The hidden\_layer =8 we have used .

This is the logic used to devlop the training model, we will pass the training data and train the model.

Output:

The output we have obtained by training the model with the training data is specified below. It contains different type of words in an array and totally 112 data have been used which belongs to a food delivery application





Chatbot:

import random

import json

import torch

from long\_responses import NeuralNet

from chatbot import bag\_of\_words, tokenize

import os

device = torch.device('cuda' if torch.cuda.is\_available() else 'cpu')

os.chdir('C:/Users/arara/OneDrive/Desktop/chatbot/')

with open('intents.json', 'r') as f:

    intents = json.load(f)

FILE = "data.pth"

data = torch.load(FILE)

input\_size = data["input\_size"]

hidden\_size = data["hidden\_size"]

output\_size = data["output\_size"]

all\_words = data["all\_words"]

model\_state = data["model\_state"]

tags = data["tags"]

model = NeuralNet(input\_size, hidden\_size, output\_size).to(device)

model.load\_state\_dict(model\_state)

model.eval()

bot\_name = "zoro"

print("Let's chat! Type 'quit' to exit.")

while True:

    sentence = input('You: ')

    if sentence == "quit":

        break

    sentence = tokenize(sentence)

    x = bag\_of\_words(sentence, all\_words)

    x = x.reshape(1, x.shape[0])

    x = torch.from\_numpy(x)  # Fix the torch.from\_numpy call

    output = model(x)

    \_, predicted = torch.max(output, dim=1)

    tag = tags[predicted.item()]  # Fix the tag retrieval

    probs = torch.softmax(output, dim=1)

    prob = probs[0][predicted.item()]

    if prob.item() > 0.75:

        for intent in intents["intents"]:

            if tag == intent["tag"]:

                print(f"{bot\_name}: {random.choice(intent['responses'])}")

    else:

        print(f"{bot\_name}: I do not understand... please only ask queries out

at the end we created the AI chatbot which can execute locally in the python itself without any user interface.

Output:

