In [1]:

```
import numpy as np
import nltk
import random
```

In [2]:

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Sinegalatha\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[2]:

```
True
```

In [3]:

```
import pickle
```

In [4]:

```
import json
```

In [5]:

```
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
```

In [6]:

```
def tokenize(sentence):
    """
    split sentence into array of words/tokens
    a token can be a word or punctuation character, or number
    """
    return nltk.word_tokenize(sentence)
```

In [7]:

```
def stem(word):
    """
    stemming = find the root form of the word
    examples:
    words = ["organize", "organizes", "organizing"]
    words = [stem(w) for w in words]
    -> ["organ", "organ", "organ"]
    """
    return stemmer.stem(word.lower())
```

In [8]:

```
def bag_of_words(tokenized_sentence, words):
    """
    return bag of words array:
    1 for each known word that exists in the sentence, 0 otherwise
    example:
    sentence = ["hello", "how", "are", "you"]
    words = ["hi", "hello", "I", "you", "bye", "thank", "cool"]
    bog   = [  0 ,    1 ,    0 ,   1 ,    0 ,    0 ,      0]
    """
    # stem each word
    sentence_words = [stem(word) for word in tokenized_sentence]
    # initialize bag with 0 for each word
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1


    return bag
```

In [9]:

```
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader
```

In [10]:

```
with open('C:\\Users\\Sinegalatha\\Desktop\\2nd year online class\\internship\\2-1.json', '
    intents = json.load(f)
```

In [11]:

```
print(intents)
```

```
{'intents': [{'Season': 'KHARIF', 'Sector': 'HORTICULTURE', 'Category': 'F
ruits', 'Crop': 'Sapota', 'QueryType': 'Fertilizer Use and Availability',
'QueryText': 'top dressing for sapota', 'KccAns': 'apply FYM 25kg+urea500g
m+SSP500gm+potash750gm/tree once in 6month', 'StateName': 'TAMILNADU', 'Di
strictName': 'TIRUNELVELI', 'BlockName': 'PALAYANKOTTAL', 'CreatedOn': '20
13-02-28T14:08:00', '': ''}, {'Season': 'RABI', 'Sector': 'AGRICULTURE',
'Category': 'Others', 'Crop': 'Others', 'QueryType': 'Weather', 'QueryTex
t': 'Asking about Weather report for Tirupur', 'KccAns': 'Recommended for
today have light rainfall.', 'StateName': 'TAMILNADU', 'DistrictName': 'TI
RUPUR', 'BlockName': 'Avanashi', 'CreatedOn': '2017-12-01T06:29:43.82',
'': ''}, {'Season': 'RABI', 'Sector': 'AGRICULTURE', 'Category': 'Others',
'Crop': 'Others', 'QueryType': 'Weather', 'QueryText': 'Asking about Thiru
ppur district rainfall information', 'KccAns': 'Recommended for having mod
erate rain today', 'StateName': 'TAMILNADU', 'DistrictName': 'TIRUPUR', 'B
lockName': 'Dharapuram', 'CreatedOn': '2017-12-01T07:28:52.01', '': ''},
{'Season': 'RABI', 'Sector': 'AGRICULTURE', 'Category': 'Others', 'Crop':
'Others', 'QueryType': 'Weather', 'QueryText': 'Asking about Weather repor
t for Tirupur', 'KccAns': 'Recommended for today have light rainfall.', 'S
tateName': 'TAMILNADU', 'DistrictName': 'TIRUPUR', 'BlockName': 'Dharapura
```

In [12]:

```
ignore_words=['?','!','.',',','(',')','&','@']
```

In [13]:

```
all_words = []
tags = []
xy = []
patternize=[]
processed_patternize=[]
answer=[]
# loop through each sentence in our intents patterns
for intent in intents['intents']:
    tag = intent['QueryType']              # tag=intent eg-Fertilizer,market price,cultivat
    ans=intent['KccAns']
    bname=intent['BlockName']                  #answers for the query text
    answer.append(ans)
    # add to tag list
    tags.append(tag)
    pattern=intent['QueryText']           #querytext
    patternize.append(pattern)
        # tokenize each word in the sentence
    w = pattern.split(" ")
    w.append(bname)

        # add to our words list
    all_words.extend(w)
    i=w
    i = [stem(k) for k in i if k not in ignore_words]  # i) removing punctuation words from
    i=" ".join(i)
    processed_patternize.append(i)
        # add to xy pair
    xy.append((w, tag))

y_train_1 = tags
```

In [14]:

```
with open('tags.pickle', 'wb') as o:
    pickle.dump(tags, o)
```

In [15]:

```
print(xy)
```

```
n', 'tirupur', 'Avanashi'], 'Weather'), (['Asking', 'about', 'Grey', 'Blig
ht', 'in', 'mango', 'Udumalaipettai'], 'Plant Protection'), (['Asking', 'd
istrict', 'Thirupur', 'district', 'rainfall', 'information', 'Udumalaipett
ai'], 'Weather'), (['Asking', 'about', 'Horticulture', 'department', 'phon
e', 'number', 'Udumalaipettai'], 'Government Schemes'), (['Asking', 'abou
t', 'Weather', 'report', 'for', 'Tirupur', 'Mulanor'], 'Weather'), (['aski
ng', 'about', 'groundnut', 'suitable', 'season', 'Kundadam'], 'Cultural Pr
actices'), (['asking', 'about', 'coconut', 'seedlings', 'availability', 'i
nformation', 'Kangyam'], 'Seeds and Planting Material'), (['asking', 'abou
t', 'coconut', 'fertilizer', 'management', 'Kangyam'], 'Fertilizer Use and
Availability'), (['Asking', 'about', 'Cocoon', 'market', 'information', 'D
harapuram'], 'Market Information'), (['asking', 'about', 'weather', 'repor
t', 'for', 'tirupur', 'district', 'Madathukulam'], 'Weather'), (['asking',
'about', 'Tirupur', 'today', 'weather', 'Report', 'Kangyam'], 'Weather'),
(['asking', 'about', 'paddy', 'top', 'dressing', 'fertilizer', 'managemen
t', 'Dharapuram'], 'Fertilizer Use and Availability'), (['Asking', 'abou
t', 'Thiruppur', 'district', 'rainfall', 'information', 'Dharapuram'], 'We
ather'), (['Asking', 'about', 'Coimbatore', 'veterinary', 'contact', 'numb
er', 'Madathukulam'], 'Dairy Production'), (['Asking', 'about', 'Coimbator
e', 'veterinary', 'contact', 'number', 'Madathukulam'], 'Dairy Productio
```

In [16]:

```
print(processed_patternize)
```

```
['top dress for sapota palayankott', 'ask about weather report for tirupur
avanashi', 'ask about thiruppur district rainfal inform dharapuram', 'ask
about weather report for tirupur dharapuram', 'ask about market rate for g
round nut vellakoil', 'ask about weather detail in tirupur avanashi', 'ask
about grey blight in mango udumalaipettai', 'ask district thirupur distric
t rainfal inform udumalaipettai', 'ask about horticultur depart phone numb
er udumalaipettai', 'ask about weather report for tirupur mulanor', 'ask a
bout groundnut suitabl season kundadam', 'ask about coconut seedl avail in
form kangyam', 'ask about coconut fertil manag kangyam', 'ask about cocoon
market inform dharapuram', 'ask about weather report for tirupur district
madathukulam', 'ask about tirupur today weather report kangyam', 'ask abou
t paddi top dress fertil manag dharapuram', 'ask about thiruppur district
rainfal inform dharapuram', 'ask about coimbator veterinari contact number
madathukulam', 'ask about coimbator veterinari contact number madathukula
m', 'ask about cow milk machin avail inform palladam', 'ask about weather
detail in tirupur udumalaipettai', 'ask about salem weather report today k
angyam', 'ask about fusarium wilt manag in tomato tirrpur', 'ask about top
fertil manag in paddi madathukulam', 'ask about sow season for sesam madat
hukulam', 'ask about weather report for tirupur district madathukulam', 'a
sk about cotton tray foliar applic inform udumalaipettai', 'ask about tiru
```

In [17]:

```
print(all_words)
print(tags)
```

```
['top', 'dressing', 'for', 'sapota', 'PALAYANKOTTAL', 'Asking', 'about',
'Weather', 'report', 'for', 'Tirupur', 'Avanashi', 'Asking', 'about', 'Thi
ruppur', 'district', 'rainfall', 'information', 'Dharapuram', 'Asking', 'a
bout', 'Weather', 'report', 'for', 'Tirupur', 'Dharapuram', 'Asking', 'abo
ut', 'Market', 'rate', 'for', 'Ground', 'nut', 'Vellakoil', 'Asking', 'abo
ut', 'weather', 'detail', 'in', 'tirupur', 'Avanashi', 'Asking', 'about',
'Grey', 'Blight', 'in', 'mango', 'Udumalaipettai', 'Asking', 'district',
'Thirupur', 'district', 'rainfall', 'information', 'Udumalaipettai', 'Aski
ng', 'about', 'Horticulture', 'department', 'phone', 'number', 'Udumalaipe
ttai', 'Asking', 'about', 'Weather', 'report', 'for', 'Tirupur', 'Mulano
r', 'asking', 'about', 'groundnut', 'suitable', 'season', 'Kundadam', 'ask
ing', 'about', 'coconut', 'seedlings', 'availability', 'information', 'Kan
gyam', 'asking', 'about', 'coconut', 'fertilizer', 'management', 'Kangya
m', 'Asking', 'about', 'Cocoon', 'market', 'information', 'Dharapuram', 'a
sking', 'about', 'weather', 'report', 'for', 'tirupur', 'district', 'Madat
hukulam', 'asking', 'about', 'Tirupur', 'today', 'weather', 'Report', 'Kan
gyam', 'asking', 'about', 'paddy', 'top', 'dressing', 'fertilizer', 'manag
ement', 'Dharapuram', 'Asking', 'about', 'Thiruppur', 'district', 'rainfal
l', 'information', 'Dharapuram', 'Asking', 'about', 'Coimbatore', 'veterin
```

In [18]:

```
all_words = [stem(w) for w in all_words if w not in ignore_words]
# remove duplicates and sort
all_words = sorted(set(all_words))
tags = sorted(set(tags))
print(tags)
```

```
['Agriculture Mechanization', 'Animal Breeding', 'Animal Nutrition', 'Animal
Production (Piggery, Goatery, Sheep Farming etc.)', 'Bio-Pesticides and Bio-
Fertilizers', 'Credit', 'Crop Insurance', 'Cultural Practices', 'Dairy Produ
ction', 'Disease Management', 'Fertilizer Use and Availability', 'Field Prep
aration', 'Fish Marketing', 'Fishery Nutrition', 'Government Schemes', 'Inte
grated Farming', 'Livestock Products Processing and Packaging', 'Market Info
rmation', 'Medicinal and Aromatic Plants', 'Nursery Management', 'Nutrient M
anagement', 'Old/Senile Orchard Rejuvenation', 'Organic Farming', 'Plant Pro
tection', 'Seeds', 'Seeds and Planting Material', 'Soil Health Card', 'Soil
Testing', 'Sowing Time and Weather', 'Storage', 'Training and Exposure Visit
s', 'Varieties', 'Vegetative Propagation and Tissue Culture', 'Water Managem
ent', 'Water Management, Micro Irrigation', 'Weather', 'Weed Management']
```

In [19]:

```
with open('tags.new', 'wb') as d:
    pickle.dump(tags, d)
```

In [20]:

```
print(all_words)
```

```
['&brown', '(adt', '(ae),tiruvannamali', '(bio)', '(bpt)', '(chithiraipatta
m)', '(cow', '(days)', '(decemb', '(fusarium', '(jasminum', '(karthigaipatta
m)', '(marghazipattam', '(markazhi', '(mn)', '(n)', '(navarai', '(navarai)',
'(novemb', '(on)5', '(or)', '(pangunipattam)', '(ponni)', '(r)', '(sri)',
'(ssi', '(thaipattam)', '(thiruvannamalai', ')avail', ',thiruvannamalai', '-
37', '-january)', '/', '1', '101.', '13', '156', '16', '2', '242', '3', '3
7', '37)', '4(cumbu', '43', '45', '5', '50', '51', '52', '6', '642', '7', 'a
boptu', 'abot', 'abour', 'about', 'abov', 'acid', 'activ', 'ada', 'address',
'adt', 'adt37', 'adult', 'age', 'agri', 'agricultur', 'agriengin', 'agrl.ent
omolog', 'alga', 'algal', 'aliyar', 'alkalin', 'alo', 'alternaria', 'amarant
hu', 'ambasamudram', 'amirtha', 'anakkavur', 'and', 'andra', 'anim', 'anthiy
ur', 'anthracnos', 'aphid', 'app', 'applic', 'arani', 'ariyalur', 'aromat',
'asd', 'ash', 'ashgourd', 'ask', 'aski', 'askign', 'askina', 'askingh', 'ass
ist', 'at', 'athiyand', 'auriculatum)fertil', 'avail', 'avalurpet', 'avanash
i', 'averag', 'avil', 'avinashi', 'avinasi', 'azola', 'azolla', 'azospirillu
m', 'back', 'bacteri', 'balleri', 'banana', 'basal', 'bean', 'bee', 'beetl',
'bellari', 'below', 'bengal', 'bengalgram', 'bhendi', 'bima', 'bio', 'bio-fe
rtil', 'bitter', 'bittergourd', 'black', 'blackgram', 'blast', 'blight', 'bl
otch', 'boar', 'bollworm', 'boot', 'bordeaux', 'bore', 'borer', 'born', 'bor
on', 'bottl', 'bout', 'bpt', 'breed', 'brinjal', 'brinjal.', 'brown', 'bud',
'budworm', 'bug', 'bulb', 'bulbrot', 'bunch', 'buprofezin', 'bush', 'butto
n', 'c', 'calcium', 'canker', 'card', 'castor', 'cater', 'caterpil', 'caterp
illar', 'cattl', 'cauliflow', 'centr', 'centre,', 'cercospora', 'certif', 'c
hemiac', 'chemic', 'chengam', 'chennai', 'chethupattu', 'chetput', 'cheyya
r', 'cheyyar,', 'chili', 'chilli', 'chinnasalem', 'chithathur,', 'chrysanthe
mum', 'chrysanthimum', 'cigar', 'citru', 'clean', 'climat', 'co', 'co51', 'c
o52', 'coccinia', 'cock', "cock'", 'coconut', 'cocoon', 'coimbator', 'cole
u', 'collar', 'collector', 'colleg', 'collor', 'colocasia', 'comb', 'combi
n', 'committe', 'commod', 'contact', 'control', 'copra', 'corh', 'coriand',
'cotton', 'cow', 'cowpea', 'credit', 'crop', 'crops,', 'crossandra', 'cucum
b', 'cuddalor', 'cultiv', 'cumbu', 'curl', 'cut', 'cutworm', 'dairi', 'dam
p', 'dasagavya', 'datespalm', 'day', 'decompos', 'deffici', 'defici', 'dehus
k', 'delta', 'depart', 'departemnt', 'departmnet', 'dept', 'detail', 'develo
p', 'dharapuram', 'dharmapuri', 'die', 'dioscorea', 'direct', 'director', 'd
iscolour', 'diseas', 'disord', 'dist', 'district', 'district)', 'dosag', 'do
wney', 'downi', 'dress', 'dri', 'drip', 'drop', 'drought', 'drum', 'drumstic
k', 'durat', 'ear', 'earhead', 'earli', 'eat', 'eleph', 'emerg', 'end', 'ene
rgi', 'engin', 'eriophyid', 'erod', 'ethrel', 'excel', 'execut', 'extract',
'fall', 'fals', 'fame', 'farm', 'farmer', 'fasal', 'feed', 'feeder', 'ferti
g', 'fertil', 'fertilzi', 'fetrtil', 'field', 'file', 'fish', 'fisheri', 'fi
ve', 'flase', 'fli', 'floricultur', 'flower', 'fodder', 'folder', 'foliar',
'for', 'forag', 'forcast', 'forecast', 'forest', 'format', 'free', 'frond',
'fruit', 'fund', 'fusarium', 'ga', 'gall', 'garden', 'garlic', 'get', 'gherk
in', 'ginge', 'gingelli', 'ginger', 'goat', 'gourd', 'govt', 'govt.', 'grai
n', 'gram', 'grass', 'grasshopp', 'green', 'greengram', 'grey', 'ground', 'g
roundnut', 'groundund', 'growth', 'grub', 'guava', 'gudimanlam', 'gummosi',
'hairi', 'harvest', 'head', 'health', 'hedda', 'herbicid', 'honeybe', 'hoppe
r', 'hopper,', 'horn', 'horticultur', 'i', 'in', 'incid', 'increas', 'indu
c', 'infoprm', 'inform', 'institut', 'insur', 'inter', 'intercrop', 'interno
d', 'introduc', 'irrig', 'ivi', 'jack', 'jasmin', 'jassid', 'jawathu hil',
'jerkin', 'kalasapakkam', 'kallakurichi', 'kancheepuram', 'kancheepuram,',
'kangayam', 'kangeyam', 'kangyam', 'kankeyam', 'kapa', 'karais', 'karpoora
v', 'kattupakkam', 'keelpenathur', 'keelpennathur', 'keep', 'kendra', 'kendr
a,', 'kernel', 'kisaan', 'kisan', 'kissan', 'kodumudi', 'krishi', 'kundada
m', 'kuthiraivali', 'kvk', 'lab', 'lablab', 'laboratori', 'land', 'latest',
'leaf', 'leafhopp', 'leafi', 'lime', 'limt', 'list', 'littl', 'loan', 'lowla
```

```
nd', 'machin', 'machineri', 'madathukulam', 'maduranthagam', 'magnesium', 'm
aiz', 'maket', 'manag', 'managementin', 'managemn', 'managemnet', 'mango',
'mantri', 'marghazipattam', 'marigold', 'markazhi', 'market', 'mazi', 'mdu',
'meali', 'mealybug', 'measur', 'medicin', 'medium', 'melay', 'melon', 'metho
d', 'methoed)', 'mettupalayam', 'micro', 'micronutri', 'midg', 'mildew', 'mi
lk', 'millet', 'millets,', 'miner', 'mini', 'minor', 'mint', 'mite', 'mobi
l', 'mochai', 'month)', 'monthan', 'moringa', 'mosaic', 'mudra', 'mulanor',
'mulberri', 'mulch', 'mullai', 'muranai', 'muringa', 'mushroom', 'mustard',
'nadu', 'nagapattinam', 'nagar', 'nagarã¢â€â“', 'naiper', 'namakk', 'name',
'nation', 'navarai', 'navarai)', 'need', 'neem', 'nematod', 'new', 'nitroge
n', 'no', 'no.', 'nrhizom', 'nug', 'number', 'nurseri', 'nut', 'nutrient',
'nutrit', 'of', 'off', 'offic', 'oil', 'onion', 'onlin', 'oothukuli', 'orga
n', 'othukuli', 'overdis', 'paddi', 'palayankott', 'palladam', 'palm', 'pana
ma', 'panchagavya', 'panchakavya', 'panjakaviyam', 'papaya', 'parthenium',
'past', 'pathogen', 'pattam)', 'peacock', 'pencil', 'perambalur', 'period',
'pernamallur', 'perunthurai', 'pest', 'pest)', 'pethappampatti', 'phone', 'p
hosphobacteria', 'pig', 'pillar', 'pinch', 'plant', 'plantat', 'pmfbi', 'po
d', 'point', 'polur', 'pomegran', 'pone', 'pongalur', 'ponni', 'post', 'pota
ssium', 'pottasium', 'poultri', 'powderi', 'power', 'pradhan', 'pre-treat',
'prematur', 'prepar', 'prevent', 'price', 'procedur', 'product', 'propag',
'public', 'puddl', 'pudupalayam', 'puls', 'pumpkin', 'purchas', 'qualiti',
'quantiti', 'rabbit', 'ragi', 'rainfal', 'rasipuram', 'rat', 'rate', 'ratoo
n', 'ratton', 'rear', 'recommend', 'red', 'redgram', 'reguat', 'regul', 'reg
ultor', 'relat', 'relief', 'report', 'reporti', 'requir', 'research', 'resis
t', 'rhinocer', 'rhinocero', 'rhizopu', 'rib', 'rice', 'ridg', 'rodent', 'ro
ller', 'roof', 'root', 'rose', 'rot', 'rotov', 'rust', 'salem', 'salin', 'sa
mba', 'sapota', 'sathyamangalam', 'scab', 'schedul', 'scheme', 'season', 'se
ed', 'seeder', 'seedl', 'seeraga', 'senjeri', 'sericultur', 'sesam', 'sesamu
m', 'sett', 'sheath', 'shed', 'sheet', 'shoot', 'short', 'site', 'sivaganga
i', 'size', 'small', 'smut', 'snack', 'snail', 'snake', 'soil', 'sona', 'sor
ghum', 'sow', 'sown', 'soybean', 'space', 'spice', 'spider', 'spiriliuna',
'spot', 'spray', 'sprayer', 'sprinkler', 'squail', 'squar', 'sri', 'ssi', 's
tage', 'station', 'station,', 'station,aliyar', 'station,cuddalor', 'statio
n,veppankulam', 'stem', 'stembor', 'stick', 'storag', 'straw', 'subsidi', 's
uck', 'sucker', 'sudden', 'sugarcan', 'sugarcaneseason', 'suitabl', 'sunflo
w', 'swarm', 'symptom', 'taluk,', 'tamarind', 'tamil', 'tanjor', 'tapioca',
'teak', 'technolog', 'temperatur', 'test', 'thaipattam', 'thaipattam)', 'tha
mmampati', 'thandrampet', 'thanjavur', 'the', 'thellar', 'thi', 'thirunelv',
'thiruppur', 'thirupur', 'thiruvanamalai', 'thiruvannalai', 'thiruvannamala
i', 'thiruvannamali', 'thiruvannmalai', 'thresh', 'thrip', 'thurinjapuram',
'tick', 'tiller', 'time', 'tindivanam', 'tirrpur', 'tirunelv', 'tiruppur',
'tirupur', 'tiruvanamalai', 'tiruvannamalai', 'tiruvannamali', 'tiruvannamla
i', 'tkm', 'tmv', 'tnau', 'tnau,', 'to', 'to[p', 'tobacco', 'today', 'toll',
'tomato', 'top', 'total', 'tractor', 'trade', 'tradit', 'train', 'transplan
t', 'trash', 'treatement', 'treatment', 'tree', 'trichi', 'tube', 'tuber',
'tuberos', 'tungro', 'turmer', 'udumalaipet', 'udumalaipettai', 'udumalpet',
'ulundurpettai', 'ulunthurpettai', 'under', 'univers', 'up', 'use', 'uthukul
i', 'utrc', 'vamban', 'vandavasi', 'varieit', 'varieri', 'varieti', 'varit
i', 'vbn', 'veget', 'velimas', 'vellakoil', 'vellor', 'vellore/tiruvannamala
i', 'vembakkam', 'veppanthattai', 'vera', 'verucosi', 'veterinari', 'vigya
n', 'villupuram', 'viru', 'water', 'watermelon', 'weatehr', 'weather', 'webb
er', 'websit', 'wed', 'weed', 'weeder', 'weevil', 'well', 'west', 'west aran
i', 'wet', 'wheat', 'white', 'whitefli', 'width', 'wild', 'wilt', 'wilt)',
'woolli', 'work', 'worm', 'yam', 'year', 'yellow', 'yield', 'yojana', 'zin
c', 'zone', 'ã¢â€â“']
```

In [21]:

```
remove_words=['(',')','&']          #  removing the symbols in (,),&
all_wordsn=[]
for i in all_words:
  if i[0] in remove_words or i[-1] in remove_words:
    if i[0] in remove_words:
      i=i[1:]
    if i[-1] in remove_words:
      i=i[:-1]
      all_wordsn.append(i)
  else:
    all_wordsn.append(i)
print(all_wordsn)
```

```
['bio', 'bpt', 'chithiraipattam', 'days', 'karthigaipattam', 'mn', 'n', 'n
avarai', 'or', 'pangunipattam', 'ponni', 'r', 'sri', 'thaipattam', ',thiru
vannamalai', '-37', '-january', '/', '1', '101.', '13', '156', '16', '2',
'242', '3', '37', '37', '4(cumbu', '43', '45', '5', '50', '51', '52', '6',
'642', '7', 'aboptu', 'abot', 'abour', 'about', 'abov', 'acid', 'activ',
'ada', 'address', 'adt', 'adt37', 'adult', 'age', 'agri', 'agricultur', 'a
griengin', 'agrl.entomolog', 'alga', 'algal', 'aliyar', 'alkalin', 'alo',
'alternaria', 'amaranthu', 'ambasamudram', 'amirtha', 'anakkavur', 'and',
'andra', 'anim', 'anthiyur', 'anthracnos', 'aphid', 'app', 'applic', 'aran
i', 'ariyalur', 'aromat', 'asd', 'ash', 'ashgourd', 'ask', 'aski', 'askig
n', 'askina', 'askingh', 'assist', 'at', 'athiyand', 'auriculatum)fertil',
'avail', 'avalurpet', 'avanashi', 'averag', 'avil', 'avinashi', 'avinasi',
'azola', 'azolla', 'azospirillum', 'back', 'bacteri', 'balleri', 'banana',
'basal', 'bean', 'bee', 'beetl', 'bellari', 'below', 'bengal', 'bengalgra
m', 'bhendi', 'bima', 'bio', 'bio-fertil', 'bitter', 'bittergourd', 'blac
k', 'blackgram', 'blast', 'blight', 'blotch', 'boar', 'bollworm', 'boot',
'bordeaux', 'bore', 'borer', 'born', 'boron', 'bottl', 'bout', 'bpt', 'bre
ed', 'brinjal', 'brinjal.', 'brown', 'bud', 'budworm', 'bug', 'bulb', 'bul
brot', 'bunch', 'buprofezin', 'bush', 'button', 'c', 'calcium', 'canker',
'card', 'castor', 'cater', 'caterpil', 'caterpillar', 'cattl', 'cauliflo
w', 'centr', 'centre,', 'cercospora', 'certif', 'chemiac', 'chemic', 'chen
gam', 'chennai', 'chethupattu', 'chetput', 'cheyyar', 'cheyyar,', 'chili',
'chilli', 'chinnasalem', 'chithathur,', 'chrysanthemum', 'chrysanthimum',
'cigar', 'citru', 'clean', 'climat', 'co', 'co51', 'co52', 'coccinia', 'co
ck', "cock'", 'coconut', 'cocoon', 'coimbator', 'coleu', 'collar', 'collec
tor', 'colleg', 'collor', 'colocasia', 'comb', 'combin', 'committe', 'comm
od', 'contact', 'control', 'copra', 'corh', 'coriand', 'cotton', 'cow', 'c
owpea', 'credit', 'crop', 'crops,', 'crossandra', 'cucumb', 'cuddalor', 'c
ultiv', 'cumbu', 'curl', 'cut', 'cutworm', 'dairi', 'damp', 'dasagavya',
'datespalm', 'day', 'decompos', 'deffici', 'defici', 'dehusk', 'delta', 'd
epart', 'departemnt', 'departmnet', 'dept', 'detail', 'develop', 'dharapur
am', 'dharmapuri', 'die', 'dioscorea', 'direct', 'director', 'discolour',
'diseas', 'disord', 'dist', 'district', 'district', 'dosag', 'downey', 'do
wni', 'dress', 'dri', 'drip', 'drop', 'drought', 'drum', 'drumstick', 'dur
at', 'ear', 'earhead', 'earli', 'eat', 'eleph', 'emerg', 'end', 'energi',
'engin', 'eriophyid', 'erod', 'ethrel', 'excel', 'execut', 'extract', 'fal
l', 'fals', 'fame', 'farm', 'farmer', 'fasal', 'feed', 'feeder', 'fertig',
'fertil', 'fertilzi', 'fetrtil', 'field', 'file', 'fish', 'fisheri', 'fiv
e', 'flase', 'fli', 'floricultur', 'flower', 'fodder', 'folder', 'foliar',
'for', 'forag', 'forcast', 'forecast', 'forest', 'format', 'free', 'fron
d', 'fruit', 'fund', 'fusarium', 'ga', 'gall', 'garden', 'garlic', 'get',
'gherkin', 'ginge', 'gingelli', 'ginger', 'goat', 'gourd', 'govt', 'gov
t.', 'grain', 'gram', 'grass', 'grasshopp', 'green', 'greengram', 'grey',
'ground', 'groundnut', 'groundund', 'growth', 'grub', 'guava', 'gudimanla
m', 'gummosi', 'hairi', 'harvest', 'head', 'health', 'hedda', 'herbicid',
```

```
'honeybe', 'hopper', 'hopper,', 'horn', 'horticultur', 'i', 'in', 'incid',
'increas', 'induc', 'infoprm', 'inform', 'institut', 'insur', 'inter', 'in
tercrop', 'internod', 'introduc', 'irrig', 'ivi', 'jack', 'jasmin', 'jassi
d', 'jawathu hil', 'jerkin', 'kalasapakkam', 'kallakurichi', 'kancheepura
m', 'kancheepuram,', 'kangayam', 'kangeyam', 'kangyam', 'kankeyam', 'kap
a', 'karais', 'karpoorav', 'kattupakkam', 'keelpenathur', 'keelpennathur',
'keep', 'kendra', 'kendra,', 'kernel', 'kisaan', 'kisan', 'kissan', 'kodum
udi', 'krishi', 'kundadam', 'kuthiraivali', 'kvk', 'lab', 'lablab', 'labor
atori', 'land', 'latest', 'leaf', 'leafhopp', 'leafi', 'lime', 'limt', 'li
st', 'littl', 'loan', 'lowland', 'machin', 'machineri', 'madathukulam', 'm
aduranthagam', 'magnesium', 'maiz', 'maket', 'manag', 'managementin', 'man
agemn', 'managemnet', 'mango', 'mantri', 'marghazipattam', 'marigold', 'ma
rkazhi', 'market', 'mazi', 'mdu', 'meali', 'mealybug', 'measur', 'medici
n', 'medium', 'melay', 'melon', 'method', 'methoed', 'mettupalayam', 'micr
o', 'micronutri', 'midg', 'mildew', 'milk', 'millet', 'millets,', 'miner',
'mini', 'minor', 'mint', 'mite', 'mobil', 'mochai', 'month', 'monthan', 'm
oringa', 'mosaic', 'mudra', 'mulanor', 'mulberri', 'mulch', 'mullai', 'mur
anai', 'muringa', 'mushroom', 'mustard', 'nadu', 'nagapattinam', 'nagar',
'nagarã¢â€â“', 'naiper', 'namakk', 'name', 'nation', 'navarai', 'navarai',
'need', 'neem', 'nematod', 'new', 'nitrogen', 'no', 'no.', 'nrhizom', 'nu
g', 'number', 'nurseri', 'nut', 'nutrient', 'nutrit', 'of', 'off', 'offi
c', 'oil', 'onion', 'onlin', 'oothukuli', 'organ', 'othukuli', 'overdis',
'paddi', 'palayankott', 'palladam', 'palm', 'panama', 'panchagavya', 'panc
hakavya', 'panjakaviyam', 'papaya', 'parthenium', 'past', 'pathogen', 'pat
tam', 'peacock', 'pencil', 'perambalur', 'period', 'pernamallur', 'perunth
urai', 'pest', 'pest', 'pethappampatti', 'phone', 'phosphobacteria', 'pi
g', 'pillar', 'pinch', 'plant', 'plantat', 'pmfbi', 'pod', 'point', 'polu
r', 'pomegran', 'pone', 'pongalur', 'ponni', 'post', 'potassium', 'pottasi
um', 'poultri', 'powderi', 'power', 'pradhan', 'pre-treat', 'prematur', 'p
repar', 'prevent', 'price', 'procedur', 'product', 'propag', 'public', 'pu
ddl', 'pudupalayam', 'puls', 'pumpkin', 'purchas', 'qualiti', 'quantiti',
'rabbit', 'ragi', 'rainfal', 'rasipuram', 'rat', 'rate', 'ratoon', 'ratto
n', 'rear', 'recommend', 'red', 'redgram', 'reguat', 'regul', 'regultor',
'relat', 'relief', 'report', 'reporti', 'requir', 'research', 'resist', 'r
hinocer', 'rhinocero', 'rhizopu', 'rib', 'rice', 'ridg', 'rodent', 'rolle
r', 'roof', 'root', 'rose', 'rot', 'rotov', 'rust', 'salem', 'salin', 'sam
ba', 'sapota', 'sathyamangalam', 'scab', 'schedul', 'scheme', 'season', 's
eed', 'seeder', 'seedl', 'seeraga', 'senjeri', 'sericultur', 'sesam', 'ses
amum', 'sett', 'sheath', 'shed', 'sheet', 'shoot', 'short', 'site', 'sivag
angai', 'size', 'small', 'smut', 'snack', 'snail', 'snake', 'soil', 'son
a', 'sorghum', 'sow', 'sown', 'soybean', 'space', 'spice', 'spider', 'spir
iliuna', 'spot', 'spray', 'sprayer', 'sprinkler', 'squail', 'squar', 'sr
i', 'ssi', 'stage', 'station', 'station,', 'station,aliyar', 'station,cudd
alor', 'station,veppankulam', 'stem', 'stembor', 'stick', 'storag', 'stra
w', 'subsidi', 'suck', 'sucker', 'sudden', 'sugarcan', 'sugarcaneseason',
'suitabl', 'sunflow', 'swarm', 'symptom', 'taluk,', 'tamarind', 'tamil',
'tanjor', 'tapioca', 'teak', 'technolog', 'temperatur', 'test', 'thaipatta
m', 'thaipattam', 'thammampati', 'thandrampet', 'thanjavur', 'the', 'thell
ar', 'thi', 'thirunelv', 'thiruppur', 'thirupur', 'thiruvanamalai', 'thiru
vannalai', 'thiruvannamalai', 'thiruvannamali', 'thiruvannmalai', 'thres
h', 'thrip', 'thurinjapuram', 'tick', 'tiller', 'time', 'tindivanam', 'tir
rpur', 'tirunelv', 'tiruppur', 'tirupur', 'tiruvanamalai', 'tiruvannamala
i', 'tiruvannamali', 'tiruvannamlai', 'tkm', 'tmv', 'tnau', 'tnau,', 'to',
'to[p', 'tobacco', 'today', 'toll', 'tomato', 'top', 'total', 'tractor',
'trade', 'tradit', 'train', 'transplant', 'trash', 'treatement', 'treatmen
t', 'tree', 'trichi', 'tube', 'tuber', 'tuberos', 'tungro', 'turmer', 'udu
malaipet', 'udumalaipettai', 'udumalpet', 'ulundurpettai', 'ulunthurpetta
i', 'under', 'univers', 'up', 'use', 'uthukuli', 'utrc', 'vamban', 'vandav
asi', 'varieit', 'varieri', 'varieti', 'variti', 'vbn', 'veget', 'velima
s', 'vellakoil', 'vellor', 'vellore/tiruvannamalai', 'vembakkam', 'veppant
hattai', 'vera', 'verucosi', 'veterinari', 'vigyan', 'villupuram', 'viru',
```

```
'water', 'watermelon', 'weatehr', 'weather', 'webber', 'websit', 'wed', 'w
eed', 'weeder', 'weevil', 'well', 'west', 'west arani', 'wet', 'wheat', 'w
hite', 'whitefli', 'width', 'wild', 'wilt', 'wilt', 'woolli', 'work', 'wor
m', 'yam', 'year', 'yellow', 'yield', 'yojana', 'zinc', 'zone', 'ã¢â€â“']
```

In [22]:

```
print(len(tags))
```

```
37
```

In [23]:

```
all_words=all_wordsn
print(all_words)
```

```
['bio', 'bpt', 'chithiraipattam', 'days', 'karthigaipattam', 'mn', 'n', 'nav
arai', 'or', 'pangunipattam', 'ponni', 'r', 'sri', 'thaipattam', ',thiruvann
amalai', '-37', '-january', '/', '1', '101.', '13', '156', '16', '2', '242',
'3', '37', '37', '4(cumbu', '43', '45', '5', '50', '51', '52', '6', '642',
'7', 'aboptu', 'abot', 'abour', 'about', 'abov', 'acid', 'activ', 'ada', 'ad
dress', 'adt', 'adt37', 'adult', 'age', 'agri', 'agricultur', 'agriengin',
'agrl.entomolog', 'alga', 'algal', 'aliyar', 'alkalin', 'alo', 'alternaria',
'amaranthu', 'ambasamudram', 'amirtha', 'anakkavur', 'and', 'andra', 'anim',
'anthiyur', 'anthracnos', 'aphid', 'app', 'applic', 'arani', 'ariyalur', 'ar
omat', 'asd', 'ash', 'ashgourd', 'ask', 'aski', 'askign', 'askina', 'asking
h', 'assist', 'at', 'athiyand', 'auriculatum)fertil', 'avail', 'avalurpet',
'avanashi', 'averag', 'avil', 'avinashi', 'avinasi', 'azola', 'azolla', 'azo
spirillum', 'back', 'bacteri', 'balleri', 'banana', 'basal', 'bean', 'bee',
'beetl', 'bellari', 'below', 'bengal', 'bengalgram', 'bhendi', 'bima', 'bi
o', 'bio-fertil', 'bitter', 'bittergourd', 'black', 'blackgram', 'blast', 'b
light', 'blotch', 'boar', 'bollworm', 'boot', 'bordeaux', 'bore', 'borer',
'born', 'boron', 'bottl', 'bout', 'bpt', 'breed', 'brinjal', 'brinjal.', 'br
own', 'bud', 'budworm', 'bug', 'bulb', 'bulbrot', 'bunch', 'buprofezin', 'bu
sh', 'button', 'c', 'calcium', 'canker', 'card', 'castor', 'cater', 'caterpi
l', 'caterpillar', 'cattl', 'cauliflow', 'centr', 'centre,', 'cercospora',
'certif', 'chemiac', 'chemic', 'chengam', 'chennai', 'chethupattu', 'chetpu
t', 'cheyyar', 'cheyyar,', 'chili', 'chilli', 'chinnasalem', 'chithathur,',
'chrysanthemum', 'chrysanthimum', 'cigar', 'citru', 'clean', 'climat', 'co',
'co51', 'co52', 'coccinia', 'cock', "cock'", 'coconut', 'cocoon', 'coimbato
r', 'coleu', 'collar', 'collector', 'colleg', 'collor', 'colocasia', 'comb',
'combin', 'committe', 'commod', 'contact', 'control', 'copra', 'corh', 'cori
and', 'cotton', 'cow', 'cowpea', 'credit', 'crop', 'crops,', 'crossandra',
'cucumb', 'cuddalor', 'cultiv', 'cumbu', 'curl', 'cut', 'cutworm', 'dairi',
'damp', 'dasagavya', 'datespalm', 'day', 'decompos', 'deffici', 'defici', 'd
ehusk', 'delta', 'depart', 'departemnt', 'departmnet', 'dept', 'detail', 'de
velop', 'dharapuram', 'dharmapuri', 'die', 'dioscorea', 'direct', 'directo
r', 'discolour', 'diseas', 'disord', 'dist', 'district', 'district', 'dosa
g', 'downey', 'downi', 'dress', 'dri', 'drip', 'drop', 'drought', 'drum', 'd
rumstick', 'durat', 'ear', 'earhead', 'earli', 'eat', 'eleph', 'emerg', 'en
d', 'energi', 'engin', 'eriophyid', 'erod', 'ethrel', 'excel', 'execut', 'ex
tract', 'fall', 'fals', 'fame', 'farm', 'farmer', 'fasal', 'feed', 'feeder',
'fertig', 'fertil', 'fertilzi', 'fetrtil', 'field', 'file', 'fish', 'fisher
i', 'five', 'flase', 'fli', 'floricultur', 'flower', 'fodder', 'folder', 'fo
liar', 'for', 'forag', 'forcast', 'forecast', 'forest', 'format', 'free', 'f
rond', 'fruit', 'fund', 'fusarium', 'ga', 'gall', 'garden', 'garlic', 'get',
'gherkin', 'ginge', 'gingelli', 'ginger', 'goat', 'gourd', 'govt', 'govt.',
'grain', 'gram', 'grass', 'grasshopp', 'green', 'greengram', 'grey', 'groun
d', 'groundnut', 'groundund', 'growth', 'grub', 'guava', 'gudimanlam', 'gumm
osi', 'hairi', 'harvest', 'head', 'health', 'hedda', 'herbicid', 'honeybe',
'hopper', 'hopper,', 'horn', 'horticultur', 'i', 'in', 'incid', 'increas',
'induc', 'infoprm', 'inform', 'institut', 'insur', 'inter', 'intercrop', 'in
ternod', 'introduc', 'irrig', 'ivi', 'jack', 'jasmin', 'jassid', 'jawathu hi
l', 'jerkin', 'kalasapakkam', 'kallakurichi', 'kancheepuram', 'kancheepura
m,', 'kangayam', 'kangeyam', 'kangyam', 'kankeyam', 'kapa', 'karais', 'karpo
orav', 'kattupakkam', 'keelpenathur', 'keelpennathur', 'keep', 'kendra', 'ke
ndra,', 'kernel', 'kisaan', 'kisan', 'kissan', 'kodumudi', 'krishi', 'kundad
am', 'kuthiraivali', 'kvk', 'lab', 'lablab', 'laboratori', 'land', 'latest',
'leaf', 'leafhopp', 'leafi', 'lime', 'limt', 'list', 'littl', 'loan', 'lowla
nd', 'machin', 'machineri', 'madathukulam', 'maduranthagam', 'magnesium', 'm
aiz', 'maket', 'manag', 'managementin', 'managemn', 'managemnet', 'mango',
```

```
'mantri', 'marghazipattam', 'marigold', 'markazhi', 'market', 'mazi', 'mdu',
'meali', 'mealybug', 'measur', 'medicin', 'medium', 'melay', 'melon', 'metho
d', 'methoed', 'mettupalayam', 'micro', 'micronutri', 'midg', 'mildew', 'mil
k', 'millet', 'millets,', 'miner', 'mini', 'minor', 'mint', 'mite', 'mobil',
'mochai', 'month', 'monthan', 'moringa', 'mosaic', 'mudra', 'mulanor', 'mulb
erri', 'mulch', 'mullai', 'muranai', 'muringa', 'mushroom', 'mustard', 'nad
u', 'nagapattinam', 'nagar', 'nagarã¢â€â“', 'naiper', 'namakk', 'name', 'nat
ion', 'navarai', 'navarai', 'need', 'neem', 'nematod', 'new', 'nitrogen', 'n
o', 'no.', 'nrhizom', 'nug', 'number', 'nurseri', 'nut', 'nutrient', 'nutri
t', 'of', 'off', 'offic', 'oil', 'onion', 'onlin', 'oothukuli', 'organ', 'ot
hukuli', 'overdis', 'paddi', 'palayankott', 'palladam', 'palm', 'panama', 'p
anchagavya', 'panchakavya', 'panjakaviyam', 'papaya', 'parthenium', 'past',
'pathogen', 'pattam', 'peacock', 'pencil', 'perambalur', 'period', 'pernamal
lur', 'perunthurai', 'pest', 'pest', 'pethappampatti', 'phone', 'phosphobact
eria', 'pig', 'pillar', 'pinch', 'plant', 'plantat', 'pmfbi', 'pod', 'poin
t', 'polur', 'pomegran', 'pone', 'pongalur', 'ponni', 'post', 'potassium',
'pottasium', 'poultri', 'powderi', 'power', 'pradhan', 'pre-treat', 'prematu
r', 'prepar', 'prevent', 'price', 'procedur', 'product', 'propag', 'public',
'puddl', 'pudupalayam', 'puls', 'pumpkin', 'purchas', 'qualiti', 'quantiti',
'rabbit', 'ragi', 'rainfal', 'rasipuram', 'rat', 'rate', 'ratoon', 'ratton',
'rear', 'recommend', 'red', 'redgram', 'reguat', 'regul', 'regultor', 'rela
t', 'relief', 'report', 'reporti', 'requir', 'research', 'resist', 'rhinoce
r', 'rhinocero', 'rhizopu', 'rib', 'rice', 'ridg', 'rodent', 'roller', 'roo
f', 'root', 'rose', 'rot', 'rotov', 'rust', 'salem', 'salin', 'samba', 'sapo
ta', 'sathyamangalam', 'scab', 'schedul', 'scheme', 'season', 'seed', 'seede
r', 'seedl', 'seeraga', 'senjeri', 'sericultur', 'sesam', 'sesamum', 'sett',
'sheath', 'shed', 'sheet', 'shoot', 'short', 'site', 'sivagangai', 'size',
'small', 'smut', 'snack', 'snail', 'snake', 'soil', 'sona', 'sorghum', 'so
w', 'sown', 'soybean', 'space', 'spice', 'spider', 'spiriliuna', 'spot', 'sp
ray', 'sprayer', 'sprinkler', 'squail', 'squar', 'sri', 'ssi', 'stage', 'sta
tion', 'station,', 'station,aliyar', 'station,cuddalor', 'station,veppankula
m', 'stem', 'stembor', 'stick', 'storag', 'straw', 'subsidi', 'suck', 'sucke
r', 'sudden', 'sugarcan', 'sugarcaneseason', 'suitabl', 'sunflow', 'swarm',
'symptom', 'taluk,', 'tamarind', 'tamil', 'tanjor', 'tapioca', 'teak', 'tech
nolog', 'temperatur', 'test', 'thaipattam', 'thaipattam', 'thammampati', 'th
andrampet', 'thanjavur', 'the', 'thellar', 'thi', 'thirunelv', 'thiruppur',
'thirupur', 'thiruvanamalai', 'thiruvannalai', 'thiruvannamalai', 'thiruvann
amali', 'thiruvannmalai', 'thresh', 'thrip', 'thurinjapuram', 'tick', 'tille
r', 'time', 'tindivanam', 'tirrpur', 'tirunelv', 'tiruppur', 'tirupur', 'tir
uvanamalai', 'tiruvannamalai', 'tiruvannamali', 'tiruvannamlai', 'tkm', 'tm
v', 'tnau', 'tnau,', 'to', 'to[p', 'tobacco', 'today', 'toll', 'tomato', 'to
p', 'total', 'tractor', 'trade', 'tradit', 'train', 'transplant', 'trash',
'treatement', 'treatment', 'tree', 'trichi', 'tube', 'tuber', 'tuberos', 'tu
ngro', 'turmer', 'udumalaipet', 'udumalaipettai', 'udumalpet', 'ulundurpetta
i', 'ulunthurpettai', 'under', 'univers', 'up', 'use', 'uthukuli', 'utrc',
'vamban', 'vandavasi', 'varieit', 'varieri', 'varieti', 'variti', 'vbn', 've
get', 'velimas', 'vellakoil', 'vellor', 'vellore/tiruvannamalai', 'vembakka
m', 'veppanthattai', 'vera', 'verucosi', 'veterinari', 'vigyan', 'villupura
m', 'viru', 'water', 'watermelon', 'weatehr', 'weather', 'webber', 'websit',
'wed', 'weed', 'weeder', 'weevil', 'well', 'west', 'west arani', 'wet', 'whe
at', 'white', 'whitefli', 'width', 'wild', 'wilt', 'wilt', 'woolli', 'work',
'worm', 'yam', 'year', 'yellow', 'yield', 'yojana', 'zinc', 'zone', 'ã¢â
€â“']
```

In [24]:

```
X_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)     #all_words is a dictionary now.
    X_train.append(bag)
```

In [25]:

```
with open('allwords.pickle', 'wb') as m:
    pickle.dump(all_words, m)
```

In [26]:

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit(y_train_1)
list(le.classes_)
y_train = le.transform(y_train_1)
```

In [27]:

```
y_train
```

Out[27]:

```
array([10, 35, 35, ..., 10,  7,  8], dtype=int64)
```

In [28]:

```
X_train = np.array(X_train)
y_train = np.array(y_train)
print(len(X_train[1]))
```

```
789
```

In [29]:

```
from sklearn.model_selection import train_test_split
```

In [30]:

```
X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.33,random
```

In [31]:

```
class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    # support indexing such that dataset[i] can be used to get i-th sample
    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    # we can call len(dataset) to return the size
    def __len__(self):
        return self.n_samples
```

In [32]:

```
dataset = ChatDataset()
```

In [33]:

```
print(len(dataset))
```

```
2160
```

In [34]:

```
batch_size=8
```

In [35]:

```
train_loader = DataLoader(dataset=dataset,batch_size=batch_size,shuffle=True,num_workers=0)
```

In [36]:

```
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 at the end
        return out
```

In [37]:

```
# Hyper-parameters
num_epochs = 20
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)
```

```
789 37
```

In [38]:

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
C:\Users\Sinegalatha\anaconda3\lib\site-packages\torch\cuda\__init__.py:52:
UserWarning: CUDA initialization: Found no NVIDIA driver on your system. Ple
ase check that you have an NVIDIA GPU and installed a driver from http://ww
w.nvidia.com/Download/index.aspx (http://www.nvidia.com/Download/index.aspx)
(Triggered internally at  ..\c10\cuda\CUDAFunctions.cpp:100.)
  return torch._C._cuda_getDeviceCount() > 0
```

In [39]:

```
model = NeuralNet(input_size, hidden_size, output_size).to(device)
```

In [40]:

```
with open('torchdevice.pickle', 'wb') as n:
    pickle.dump(device, n)
```

In [41]:

```
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

In [42]:

```
# Train the model
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        metrics="accuracy"


        print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')
```

```
Epoch [1/20], Loss: 3.7407
Epoch [1/20], Loss: 3.6939
Epoch [1/20], Loss: 3.6252
Epoch [1/20], Loss: 3.5738
Epoch [1/20], Loss: 3.7225
Epoch [1/20], Loss: 3.7795
Epoch [1/20], Loss: 3.6510
Epoch [1/20], Loss: 3.6252
Epoch [1/20], Loss: 3.6703
Epoch [1/20], Loss: 3.6886
Epoch [1/20], Loss: 3.7098
Epoch [1/20], Loss: 3.5811
Epoch [1/20], Loss: 3.6667
Epoch [1/20], Loss: 3.5205
Epoch [1/20], Loss: 3.5516
Epoch [1/20], Loss: 3.6258
Epoch [1/20], Loss: 3.5786
Epoch [1/20], Loss: 3.6543
Epoch [1/20], Loss: 3.7084
Epoch [1/20], Loss: 3.6934
```

In [43]:

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

```
final loss: 1.6333
```

In [44]:

```
data = {
"model_state": model.state_dict(),
"input_size": input_size,
"hidden_size": hidden_size,
"output_size": output_size,
"all_words": all_words,
"tags": tags
}
```

In [45]:

```
FILE = "data.pth"
torch.save(data, FILE)
```

In [46]:

```
import pandas
```

In [47]:

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

```
training complete. file saved to data.pth
```

In [48]:

```
import torch
```

In [49]:

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

In [50]:

```
data = torch.load(FILE)
```

In [51]:

```
res={}
for cl in range(0,len(patternize)):
  res.update({patternize[cl]:answer[cl]})
print(res)
```

```
{'top dressing for sapota': 'apply FYM 25kg+urea500gm+SSP500gm+potash750g
m/tree once in 6month', 'Asking about Weather report for Tirupur': 'Recomm
ended for today have light rainfall.', 'Asking about Thiruppur district ra
infall information': 'Recommended for having no rain (0 mm)today', 'Asking
about Market rate for Ground nut': 'Recommended for Rs 6436 /quintal.', 'A
sking about weather detail in tirupur': 'recommended for have no rain in t
irupur', 'Asking about Grey Blight in mango': 'Recommended for spray coppe
r oxy chloride 2.5 g/lit of water', 'Asking district Thirupur district rai
nfall information': 'Recommended for having moderate rain today', 'Asking
about Horticulture department phone number': 'Recommended for phone numbe
r:04175-233337', 'asking about groundnut suitable season': 'recommended fo
r Chithiraipattam April-May suitable for sowing', 'asking about coconut se
edlings availability information': 'Recommended for Coconut Research Stati
on, Veppankulam, Thanjavur contact no. 04373 - 260205, 04373 - 202534', 'a
sking about coconut fertilizer management': 'Recommended for apply urea 1.
300kg + super phosphate 2kg + potash 2 kg + neem cake 5kg + farm yard manu
re 50kg + micro mixture 1kg / tree / year', 'Asking about Cocoon market in
formation': 'Recommended for Coimbatore market price Rs.454-558 /Kg(30.11.
17)', 'asking about weather report for tirupur district': 'Recommended for
```

In [52]:

```
input_size = data["input_size"]
hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data['all_words']
tags = data['tags']
model_state = data["model_state"]
```

In [53]:

```
model = NeuralNet(input_size, hidden_size, output_size).to(device)
```

In [54]:

```
model.load_state_dict(model_state)
```

Out[54]:

```
<All keys matched successfully>
```

In [55]:

```
model.eval()
```

Out[55]:

```
NeuralNet(
  (l1): Linear(in_features=789, out_features=8, bias=True)
  (l2): Linear(in_features=8, out_features=8, bias=True)
  (l3): Linear(in_features=8, out_features=37, bias=True)
  (relu): ReLU()
)
```

In [56]:

```
import pickle
with open('neuralnet.pickle', 'wb') as v:
    pickle.dump(model, v)
```

In [57]:

```
from difflib import get_close_matches
```

In [58]:

```
res={}
for cl in range(0,len(patternize)):
  res.update({patternize[cl]:answer[cl]})
print(res)
```

```
{'top dressing for sapota': 'apply FYM 25kg+urea500gm+SSP500gm+potash750g
m/tree once in 6month', 'Asking about Weather report for Tirupur': 'Recomm
ended for today have light rainfall.', 'Asking about Thiruppur district ra
infall information': 'Recommended for having no rain (0 mm)today', 'Asking
about Market rate for Ground nut': 'Recommended for Rs 6436 /quintal.', 'A
sking about weather detail in tirupur': 'recommended for have no rain in t
irupur', 'Asking about Grey Blight in mango': 'Recommended for spray coppe
r oxy chloride 2.5 g/lit of water', 'Asking district Thirupur district rai
nfall information': 'Recommended for having moderate rain today', 'Asking
about Horticulture department phone number': 'Recommended for phone numbe
r:04175-233337', 'asking about groundnut suitable season': 'recommended fo
r Chithiraipattam April-May suitable for sowing', 'asking about coconut se
edlings availability information': 'Recommended for Coconut Research Stati
on, Veppankulam, Thanjavur contact no. 04373 - 260205, 04373 - 202534', 'a
sking about coconut fertilizer management': 'Recommended for apply urea 1.
300kg + super phosphate 2kg + potash 2 kg + neem cake 5kg + farm yard manu
re 50kg + micro mixture 1kg / tree / year', 'Asking about Cocoon market in
formation': 'Recommended for Coimbatore market price Rs.454-558 /Kg(30.11.
17)', 'asking about weather report for tirupur district': 'Recommended for
```

In [59]:

```
with open('dictres.pickle', 'wb') as p:
    pickle.dump(res, p)
```

In [60]:

```
bot_name = "Sinegalatha"
print("Let's chat! (type 'quit' to exit)")
test=[]
while True:
    # sentence = "do you use credit cards?"
    sentencei = input("You: ")
    if sentencei == "quit":
        break
    sentence = sentencei.split(" ")
    X = bag_of_words(sentence, all_words)
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)
    output = model(X)
    _, predicted = torch.max(output, dim=1)
    print(predicted.item())
    tag = tags[predicted.item()]
    print(tag)

    for intent in intents['intents']:
        if tag == intent["QueryType"]:
          test.append(intent["QueryText"])

    p=[]
    p=(get_close_matches(sentencei, test))
    if len(p)==0:
      print("Make a call to Kisan Call Centre ")
    else:
      u=res[p[0]]
      print(u)
```

```
Let's chat! (type 'quit' to exit)
You: what is the market rate of copra
17
Market Information
recommended market rate for copra Tiruchengode-8800-10400Rs/Quintal
You: what is the paddy varieties
31
Varieties
Recommended for Paddy varieties ADT 36, ADT 39, ASD 16, ASD 18, MDU 5, CO 4
7,CORH 3, ADT 43, ADT (R) 45
You: quit
```

In [61]:

```
predict_tag=[]
for X in X_test:
    X = X.reshape(1, X.shape[0])
    X = torch.from_numpy(X).to(device)
    output = model(X)
    _, predicted = torch.max(output, dim=1)

    print(predicted.item())
    predict_tag.append(predicted.item())
```

```
35
20
23
14
35
17
35
23
24
23
35
10

35
35
23
23
10
35
20
```

In [62]:

```
predict_train = np.array(predict_tag)
```

In [63]:

```
test_train = np.array(y_test)
```

In [64]:

```
from sklearn.metrics import accuracy_score
```

In [65]:

```
a=accuracy_score(predict_train, test_train)
```

In [66]:

```
print(a)
```

```
0.7136150234741784
```

In [ ]:

```

```