### College Project ###

import random

import json

import pickle

import numpy as np

import nltk

from nltk.stem import WordNetLemmatizer

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Activation, Dropout

from tensorflow.keras.optimizers import SGD

lemmatizer = WordNetLemmatizer()

intents = json.loads(open('intents.json').read())

words = []

classes = []

documents = []

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

for intent in intents['intents']:

for pattern in intent['patterns']:

word\_list = nltk.word\_tokenize(pattern)

words.extend(word\_list)

documents.append((word\_list,intent['tag']))

if intent['tag'] not in classes:

classes.append(intent['tag'])

words = [lemmatizer.lemmatize(word) for word in words if word not in ignore\_letters]

words = sorted(set(words))

classes = sorted(set(classes))

pickle.dump(words, open('words.pkl', 'wb'))

pickle.dump(classes, open('classes.pkl', 'wb'))

training = []

output\_empty = [0] \* len(classes)

for document in documents:

bag =[]

word\_patterns = document[0]

word\_patterns = [lemmatizer.lemmatize(word.lower()) for word in word\_patterns]

for word in words:

bag.append(1) if word in word\_patterns else bag.append(0)

output\_row = list(output\_empty)

output\_row[classes.index(document[1])] = 1

training.append([bag, output\_row])

random.shuffle(training)

training = np.array(training)

train\_x = list(training[:, 0])

train\_y = list(training[:, 1])

model = Sequential()

model.add(Dense(128, input\_shape=(len(train\_x[0]),), activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train\_y[0]), activation='softmax'))

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

model.compile(loss='categorical\_crossentropy', optimizer=sgd, metrics=['accuracy'])

hist = model.fit(np.array(train\_x), np.array(train\_y), epochs=200, batch\_size=5, verbose=1)

model.save('chatbotmodel.h5', hist)

print('Done')