

train.py

```
1 import json
2 import numpy as np
3 import pickle
4 from sklearn.preprocessing import LabelEncoder
5 import tensorflow as tf
6 from tensorflow.keras.models import Sequential
7 from tensorflow.keras.layers import Dense, Dropout
8 from tensorflow.keras.optimizers import Adam
9 from nltk.tokenize import word_tokenize
10 from nltk.stem import PorterStemmer
11
12 # Load dataset
13 with open("dataset/intents.json") as file:
14     data = json.load(file)
15
16 stemmer = PorterStemmer()
17 sentences, labels = [], []
18
19 for intent in data['intents']:
20     for pattern in intent['patterns']:
21         sentences.append(pattern)
22         labels.append(intent['tag'])
23
24 # Preprocessing
25 tokenizer = tf.keras.preprocessing.text.Tokenizer()
26 tokenizer.fit_on_texts(sentences)
27 X = tokenizer.texts_to_matrix(sentences)
28
29 label_encoder = LabelEncoder()
30 y = label_encoder.fit_transform(labels)
31
32 # Model building
33 model = Sequential([
34     Dense(128, activation='relu', input_shape=(len(X[0]),)),
35     Dropout(0.5),
36     Dense(64, activation='relu'),
37     Dropout(0.5),
38     Dense(len(set(labels))), activation='softmax')
39 ])
40
41 model.compile(loss="sparse_categorical_crossentropy", optimizer=Adam(0.01), metrics=['accuracy'])
42 history = model.fit(X, y, epochs=150, batch_size=8)
43
44 model.save("model/chatbot_model.h5")
45 pickle.dump(tokenizer, open("model/tokenizer.pkl", "wb"))
46 pickle.dump(label_encoder, open("model/label_encoder.pkl", "wb"))
47
```

```
48 | print("Model trained & saved successfully!")  
49 |
```