Intelligent Chatbot using deep learning

Deep Learning Project

By Sejal Wasule

1. Abstract

The project title is “Intelligent Chatbot” which can be developed for numerous applications. Intelligent Chatbots are the programs based on AI i.e Artificial Intelligence which can simulate human conversation. They are interactive and trained as per the requirement of a particular application, which uses natural language processing. There are different types of chatbots which respond actively improving the customer services in an organization and help the people by providing them with the most suitable information and the answers for the queries. Many times when we visit a website, we tend to search for the relevant information which may consume our time and might end up with annoyance. In this case, if there is quick help to get the most relevant information, we can save our time. Intelligent Chatbot just needs our question so to revert with the related solution and it reduces the work of a human.

2. Introduction

In this project, we provide input to our program which is in the form of a text and can be a greeting, a question or any query according to the application. Natural language processing (NLP) is the technology that is mainly applied and driving the growth of chat bots. This Intelligent chatbot works by the intent classifier engine which is a component of NLP where An intent classifier maps between what a user asks and the type of action performed by the software. The AI chatbot identifies the language, context, and intent, which then reacts accordingly. The output we get after this are the most relevant ones as per the bot and depends on the type of our input we have provided.

For example:- If we type a word “hello” to the chatbot it will respond to us by greeting which is “hello/hi/hola” etc. It will process the input word, identify it and will perform the action i.e give us the output. At times, the chatbot may not give the expected output because it either couldn’t not process the input or the training was not done at advance level.

3. Related work

The application of artificial intelligence in chatbots has undergone a significant evolution since the 1980s. The Artificial Intelligence Mark-up Language (AIML), an extension of XML, serves as the foundation for A.L.I.C.E. (Artificial Intelligent Internet Computer Entity). It was created specifically to enable the expansion of A.L.I.C.E.'s software's knowledge base through the addition of dialogue pattern knowledge. Topics and classifications make up data objects in AIML. The fundamental unit of knowledge is the category, which serves as a rule to match user inputs with chatbot outputs. Rule patterns reflect user input, and rule templates and the A.L.I.C.E. knowledge base define the chatbot's output. Since the knowledge base was updated, the addition of new data objects in AIML marked a major improvement over earlier pattern matching systems.

4. Dataset and Features

In this project our chatbot is an intent-based chatbot. A chatbot that detects this user intent is known as an intent-based chatbot. As a result, the chatbot can determine the meaning of the message being sent rather than relying on specific input. (And then provide a pertinent, appropriate response).Our data set is called intents which contain tag(keyword), pattern(different ways in which a sentence can be given as input), response(appropriate response to the input) and context. All the above -mentioned things are in the form of a nested dictionary. We created our own dataset of intents for this project.

Talking about preprocessing first tokenization is done so that each word is separated from the sentence and stored in a list as an element. Then we lemmatized so that the words are first converted in lower case then prefix and suffix is removed from each word so only the root word is left (Suppose Input: walking, walker, walked; Output: walk).This is also known as text normalization. Then we create a bag of words i.e we keep the useful words and discard the stop words (like ‘is’,’am’,’or’,’but’ and punctuation). Then we convert this bag of words into one hot encoded vector which represents categorical variables as binary vectors. To implement this the categorical values first ought to be mapped to an integer, then each integer is represented as a binary vector — meaning that the vector would be all 0’s except the index of the integer.

5. Method

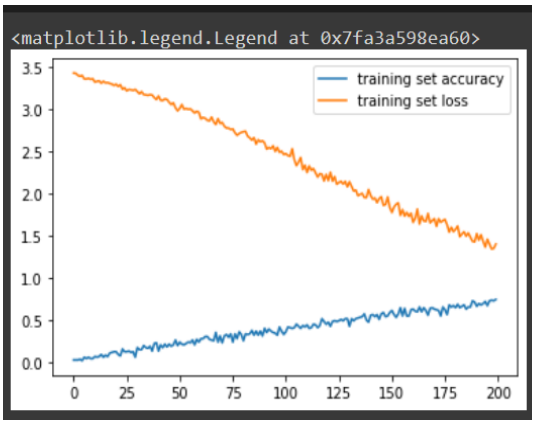
Data cleaning: In order to develop our bot, we'll need several built-in Python functions as well as well-known deep learning and NLP libraries as well as the excellent array manipulation package NumPy. Then, all the data is appropriately serialized thanks to the creation of an intents file in json format. Following that, this is lemmatized, tokenized, and transformed to numerical format (One Hot Encoded**).**

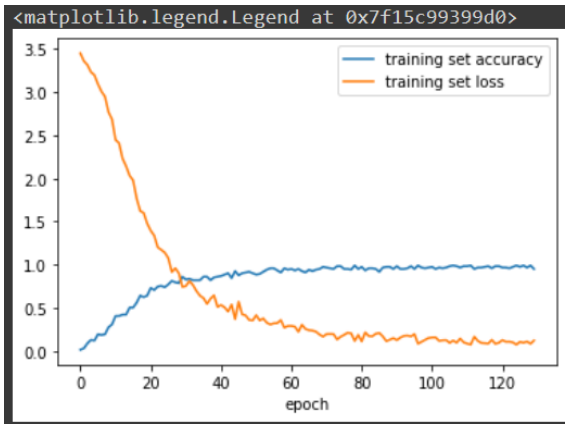
Defining Model: With our data converted into a numerical format, we can now build a Neural Network model that we are going to feed our training data into. The idea is that the model will look at the features and predict the tag associated with the features then will select an appropriate response from that tag. In our sequential model, we used a dense layer of shape 128 with 'relu ' activation then a dropout layer of 0.3 then again a dense layer of shape 64 with 'relu ' activation and a dropout layer of 0.3 then a dense layer with softmax activation function. The optimizer we used here is adam with learning_rate=0.001, decay=1e-5. Dropout layers are very effective at preventing deep learning models from overfitting to data.

Compiling and fitting model: We are using categorical cross entropy as a loss function epochs=130,batch_size=5, verbose=1 for fitting the model with accuracy for metrics.

Input Text Cleaning, processing: The user can enter a query into a while loop that is then cleaned, which means that we take the tokens and lemmatize each word. Then, using our bag of words model, we translate our language into numerical values and forecast which tag in our intents the features most accurately represent. In order to react to the question, we would then choose at random one of our responses from those contained under that intents tag.

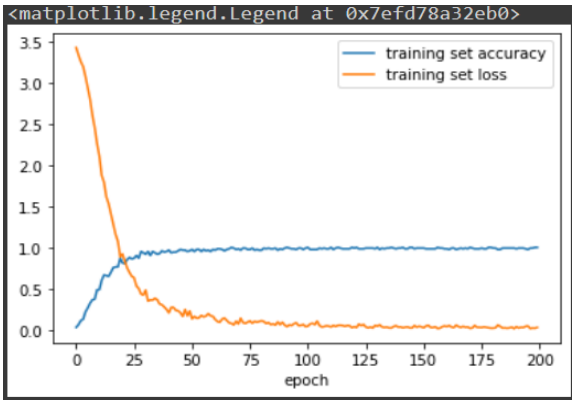
6. Experiments/Results/Discussion:





For 0.0001 learning rate and 200 epochs **For learning rate 0.001 and 130 epochs**

We experimented a lot with the hyperparameters , with various learning rates changing them from 1 to 0.0001. We tried to vary the decay , change the dropout layer parameters. We also noted the time for each variation. When the learning rate was increased (1,0.1)) the Graphs were diverging the but time for model fitting was about 5 to 10 seconds. But as we decreased the learning rate to 0.01, 0.001 we saw change in the graph and it was converging and loss is minimizing. But time taken was increased by 5 seconds more .which is alright. Although the graph keeps fluctuating after breathing 100% accuracy if epochs are increased.



Best stable outcome so far (learning rate is 0.001 with 200 epochs and loss 0.0311)

Time (Total for fitting model)	Learning Rate(200 Epochs)	Accuracy when learning rate varied	loss
11s	1	0.0650	3.9634
12s	0.1	0.0731	3.3439
11s	0.5	0.0244	3.5545
15s	0.01	0.9675	0.0435
13s	0.05	0.2683	2.5797
18s	0.001	1.000	0.0311
16s	0.005	0.9837	0.0314
21s	0.0001	0.8374	0.9472
19s	0.0005	0.9837	0.0674

Figure: Epochs vs Accuracy and learning rate vs loss vs time

7. Conclusion/Future Work:

We saw various ways through which a chatbot can be programmed. We tried Various architectures, models with various layers, but this model was giving us precision every time we run it with a dataset. The outcome was in a similar range. We have also shown a table which gives the accuracy along the epoch. As we know precision matters along with accuracy this model was stable and running smoothly. This chatbot was a general purpose chatbot like a google assistant which classified text. Variations of datasets can be made for various applications.

8 References/Bibliography :

1. **How To Build Your Own Friend by Published in Towards Data Science by Kurtis Pykes published on Mar 31, 2021.**

2. **What is NLP (Natural Language Processing) Tokenization?, Published in Toknizer(blog) By ANNI BURCHFIEL on MAY 16, 2022**

3. **Perplexity in Language Models Published in Towards Data Science by Chiara Campagnola published on May 18, 2020**

4. **Machine Learning with Applications published on 15 December 2020**

5. **Chatbots: History, technology, and applications - ScienceDirect Volume 2, 15 December 2020, 100006**