Udacity Machine Learning Nanodegree Chat Bot

Grigoris Papapostolou
June 4rth, 2021

1. Domain Background

Chatbots are simulations which can understand human language, process it and interact back with humans while performing specific tasks. Today, almost all companies have chatbots to engage their users and serve customers by catering to their queries.

They represent a potential shift in how people interact with data and services online. While there is currently a surge of interest in chatbot design and development, we lack knowledge about why people use chatbots.

There are many types of chatbots available, a few of them can be majorly classified as follows:

- **Text-based chatbot**: In a text-based chatbot, a bot answers the user's questions via text interface.
- **Voice-based chatbot**: In a voice or speech-based chatbot, a bot answers the user's questions via a human voice interface.

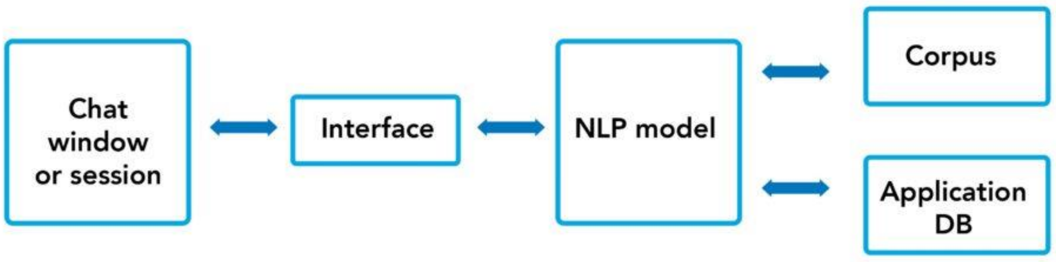
There are mainly two approaches used to design the chatbots, described as follows:

- In a **Rule-based** approach, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex. The bots can handle simple queries but fail to manage complex ones.
- **Self-learning** bots are the ones that use some Machine Learning-based approaches and are definitely more efficient than rule-based bots. These bots can be further classified in two types: Retrieval Based or Generative

The Architecture of chatbots

Typical chatbot architecture consists of the following:

- Chat window/ session/ or front end application interface
- The deep learning model for Natural Language Processing [NLP]
- Corpus or training data for training the NLP model
- Application Database for processing actions to be performed by the chatbot



2. Problem Statement

My parents own a music school for which I have created a website. My goal is to create a initial template of a chat bot which will be tested by users on the website. I will upgrade both the UI and the Machine Learning model at a later time.

The purpose of Chat Bot is to immediately serve our customers by providing instructions on where to find the information they need on our website.

We will try to create a text-based,rule-based Chabot which means we will train our model based on a series of defined rules .

3. Inputs

In the Rule-Based approach generally, a set of ground rules are set and the chatbot can only operate on those rules. We have a json file that contains all the intents and input patterns. It looks like:

```
{
  "tag": "greeting",
  "patterns": ["Hi","Hey","How are you","Is anyone there?","Hello","Good day"],
  "responses": ["Hi there, what can I do for you?","Hi there, how can I help?"]
},
{
  "tag": "goodbye",
  "patterns": ["Bye", "See you later", "Goodbye"],
  "responses": ["See you later, thanks for visiting","Have a nice day","Bye! Come back again soon."]
},
{
  "tag": "thanks",
  "patterns": ["Thanks", "Thank you", "That's helpful", "Thank's a lot!"],
  "responses": ["Happy to help!", "My pleasure"]
},
```

We can see that here we have a ‘tag’ field which actually depicts the intentions. Intentions are usually terms, which are the actual subject or motive behind a sentence given. For example, here there are intentions like ‘thanks’, ‘greetings’, and ‘goodbye’ which are basically motives. Secondly, we have a ‘patterns’ field which actually depicts the patterns or type of sentences that can have the corresponding motive. Then we have the responses field which contains some responses which may be the bot’s response if the corresponding motive is detected. For example, for the ‘tag’ or intention of ‘greeting’ patterns detected can be ‘Hi’, ‘Hello’, and the corresponding responses can be ‘Hello’, ‘Hi There’.

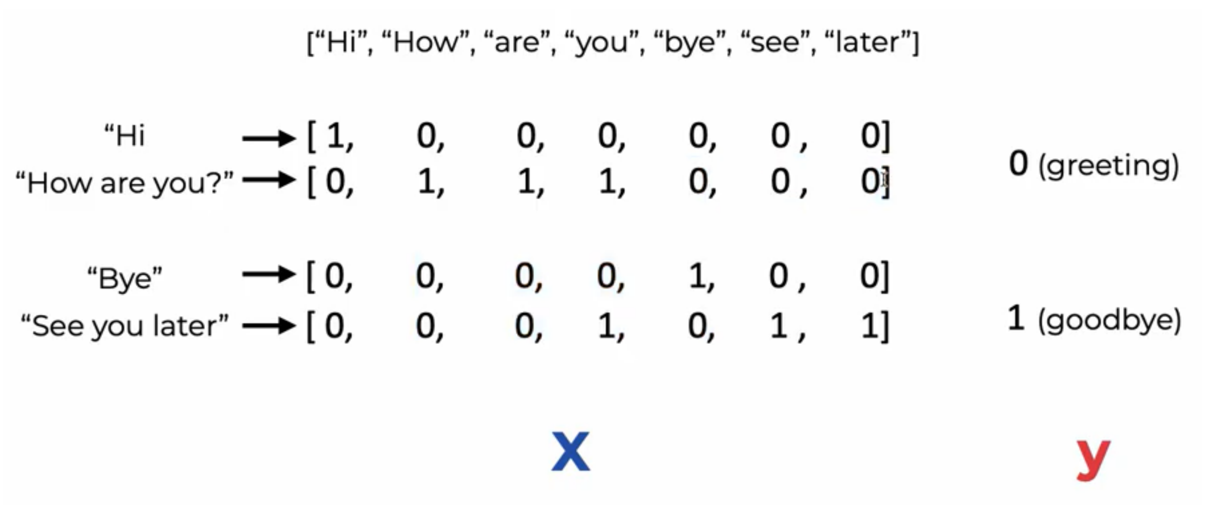
Depending on the type of customers served by the chatbot file we need to put the corresponding intentions.

4. Solution

In order to create our model we need to define the following components :

i. Bag Of Words

Algorithms take vectors of numbers as input, therefore we need to convert documents to fixed-length vectors of numbers. A simple and effective model for thinking about text documents in machine learning is called the Bag-of-Words Model, or BoW. This can be done by assigning each word a unique number. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document. In our case our data look like following:



["Hi", "How", "are", "you", "bye", "see", "later"]

"Hi	→	[1, 0, 0, 0, 0, 0, 0]	0 (greeting)
"How are you?"	→	[0, 1, 1, 1, 0, 0, 0]	
"Bye"	→	[0, 0, 0, 0, 1, 0, 0]	1 (goodbye)
"See you later"	→	[0, 0, 0, 1, 0, 1, 1]	
		X	y

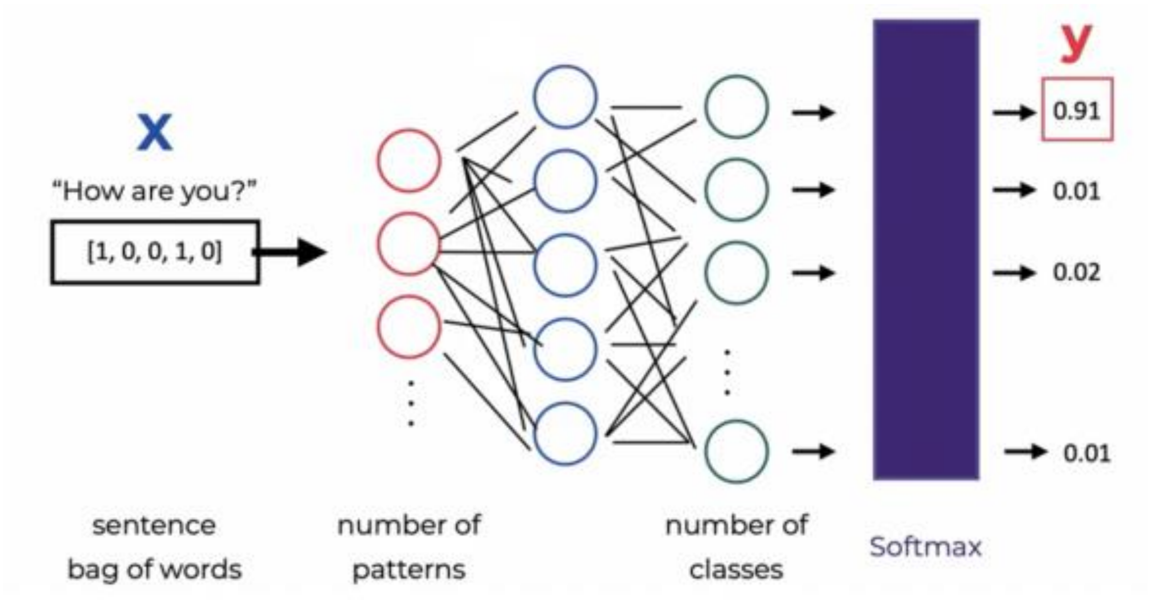
ii. Model

Neural networks are made up of layers of neurons, which are the core processing unit of the network.

The architecture of our deep learning neural network consists of three main components:

a. Input Layer : This is where the training observations are fed.
b. Hidden Layers : These are the intermediate layers between the input and output layers. The deep neural network learns about the relationships involved in data in this component. Our model will consists of 2 layers :
    - 1st Hidden Layer : (Input, Output) = (54 , 8)
    - 2nd Hidden Layer : (Input, Output) = (8 , 8)

c. Output Layer : This is the layer where the final output is extracted from what's happening in the previous two layers. In case of our problem, the output layer will have the same size of tags.



iii. Loss Function

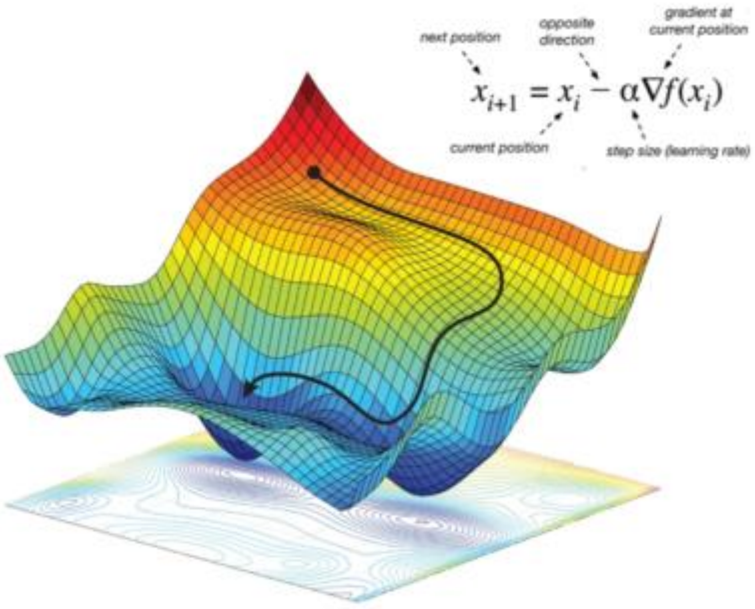
Cross entropy loss is commonly used in classification tasks both in traditional ML and deep learning. The formula is the following :

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i), \text{where } t_i \text{ is the binary truth label for class } i, p_i \text{ is the softmax probability}$$

iv. Optimizer

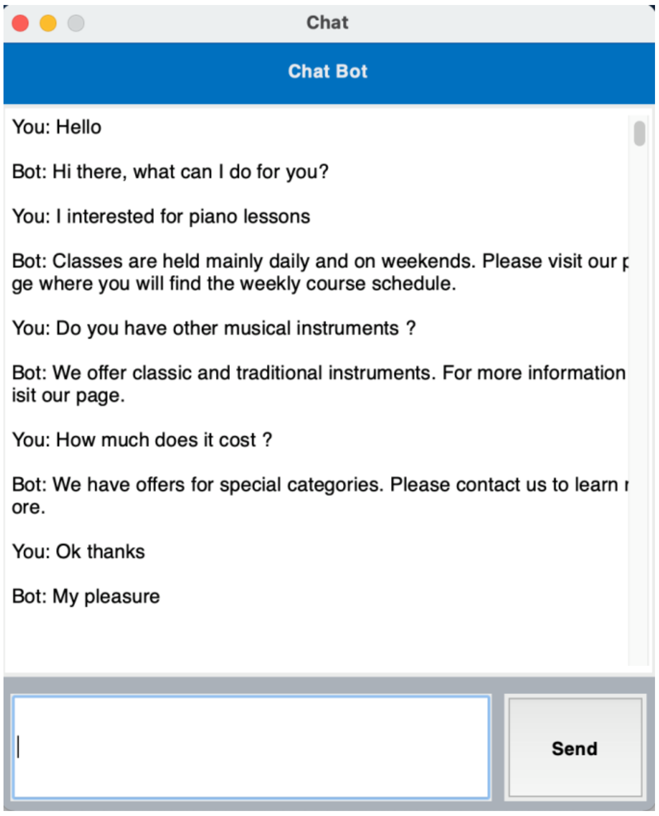
During neural network training, its weights are randomly initialized initially and then they are updated in each epoch in a manner such that they minimize the loss function using an optimizer.

Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.


next position
opposite direction
gradient at current position
$x_{i+1} = x_i - \alpha\nabla f(x_i)$
current position
step size (learning rate)

5. Project Design & Performance Metrics

The final product will look like this:



You: Hello

Bot: Hi there, what can I do for you?

You: I interested for piano lessons

Bot: Classes are held mainly daily and on weekends. Please visit our p
ge where you will find the weekly course schedule.

You: Do you have other musical instruments ?

Bot: We offer classic and traditional instruments. For more information
isit our page.

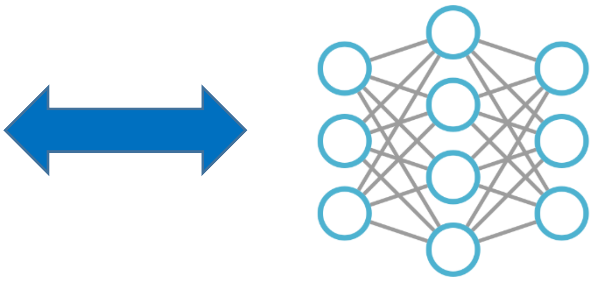
You: How much does it cost ?

Bot: We have offers for special categories. Please contact us to learn r
ore.

You: Ok thanks

Bot: My pleasure

User Interface



ML Model

As I mentioned there is room for improvement in the graphical environment as well in Machine Learning model.

To measure the effectiveness of the chatbot we will use user metrics like the followings:

- **Self-service rate**: percentage of user sessions that did not end with a contact action after using the bot.
- **Performance rate**: number of correct answers divided by the number of active sessions
- **Satisfaction rate**: average grade given when evaluating the chatbot’s answers