AI-Based Chatbot using NLP and Deep Neural Network

CSE-454 Term Project

Department of Computer Science and Engineering
Military Institute of Science and Technology

ABSTRACT

AI-based Chatbot is a computer program or an Artificial Intelligence software that can simulate a real human conversation with real-time responses to users based on reinforced learning. AI Chatbots either use text messages, voice commands, or both. AI robots use natural language processing to communicate with Artificial Intelligence features embedded in them. The main aim of creating AI chatbots is to help customers make better-informed decisions. Intelligent planning with the help of heuristics and ANN (Artificial Neural Network) aids an AI chatbot that learns to progress from one user request to another to come up with an intelligent response before completing the task. Planning is a process that relates to a sequence of actions performed by the AI chatbot leading to a structured conversation that involves acknowledgment, questions, and information relay. Internally the AI Bot has to perform intelligence building to form structured responses.

Group no: 1

Group member

Md.Rezwan -A- Rownok(201714035)
Samiha Raisa Zaman(201714052)

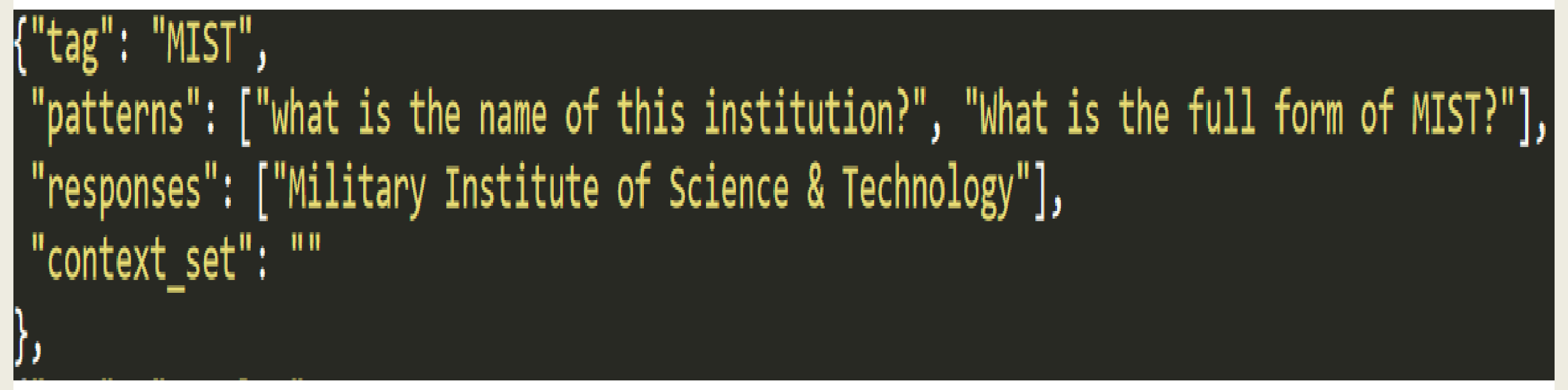
INTRODUCTION

In the world of machine learning and AI, there are many different kinds of chatbots. Some chatbots are virtual assistants; others are just there to talk to; some are customer support agents. Here, we will be creating a chatbot for MIST-website that will be used to answer frequently asked questions. This project has been done in 3 major steps.

- Dataset creation and pre-processing the dataset for feeding it into the neural network.
- Creating the Neural Network Model.
- Predicting responses in real-time.

Dataset

The dataset is created in a JSON file, and a little portion of the dataset is shown below:



```
{"tag": "MIST",
 "patterns": ["what is the name of this institution?", "What is the full form of MIST?"],
 "responses": ["Military Institute of Science & Technology"],
 "context_set": ""
},
```

Each conversational intent contains:

- a tag (a unique name)
- patterns (sentence patterns for our neural network text classifier)
- responses (one will be used as a response)

Data Preprocessing

- For each pattern, word_tokenizer is used to split the words from the sentences, and word-stemmer is used to reduce the vocabulary by replacing the word with its root form.
- Neural networks and machine learning algorithms require numerical input. Therefore, 'Bag of Words' is used to represent the sentences with numbers. Here, each sentence's length from the pattern is the number of words in our model-vocabulary list.
- Each position in the list represents a word from the vocabulary-list. If the position in the list is a 1, then that means the word exists in our sentence; if it is a 0, then the word is not present.
- The output is also formatted to make sense to the neural network. Each position in the output-list represents one distinct label/tag; a 1 in any of those positions shows which label/tag is represented.

Neural Network Model

The architecture and the model summary are shown below:

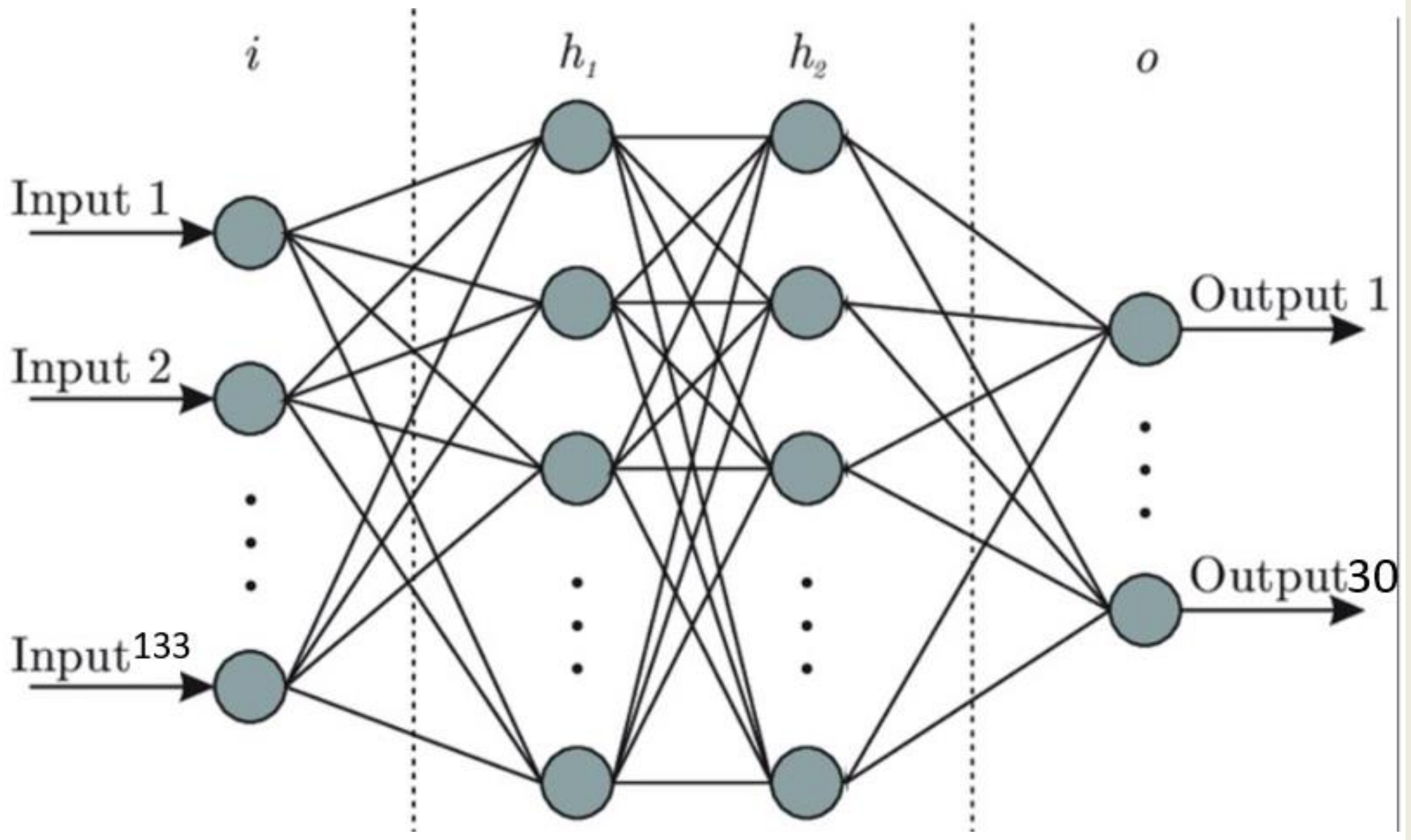
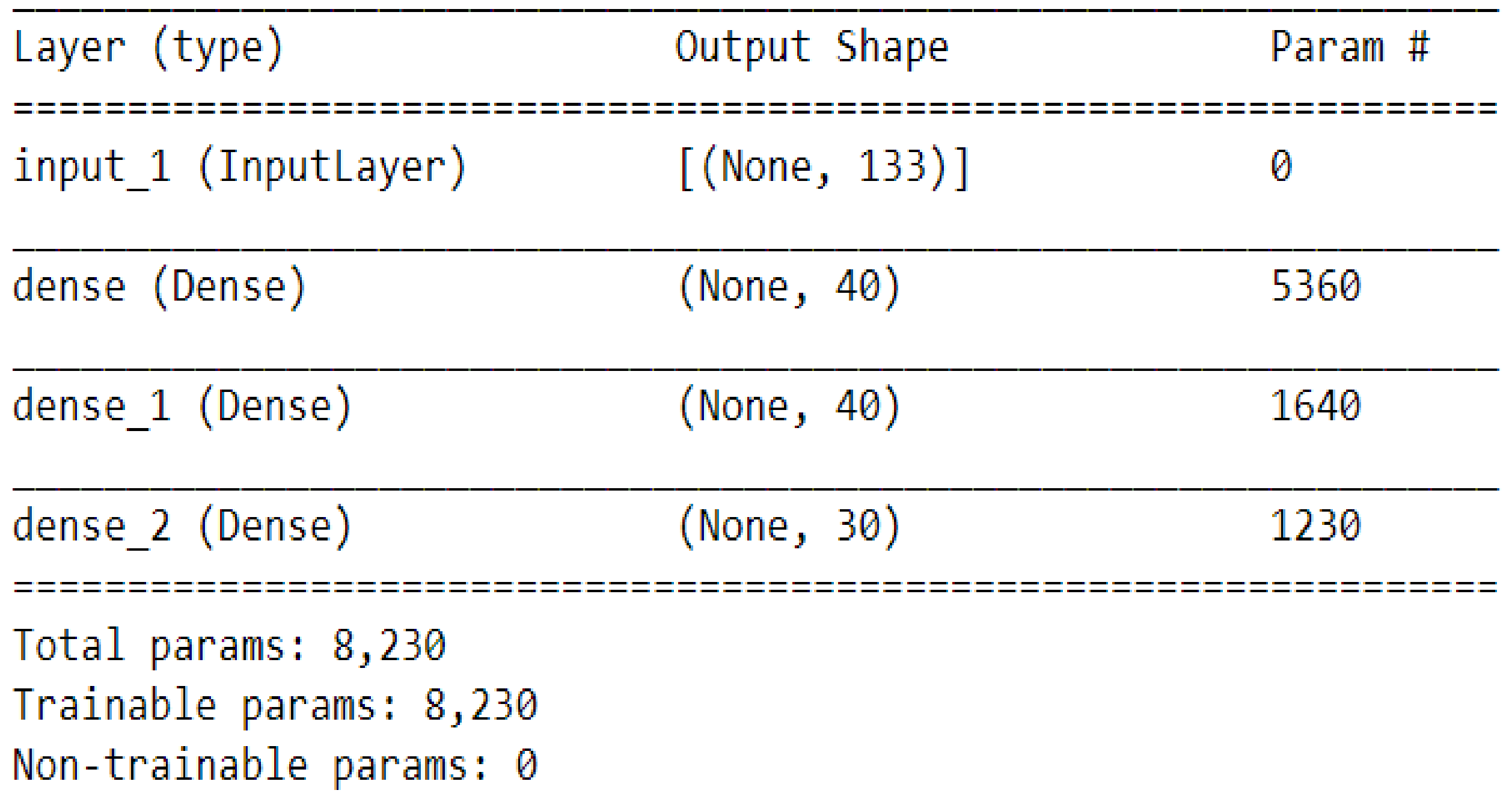


Figure1: Model Architecture



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 133)]             0

_________________________________________________________________
dense (Dense)                (None, 40)                5360

_________________________________________________________________
dense_1 (Dense)              (None, 40)                1640

_________________________________________________________________
dense_2 (Dense)              (None, 30)                1230
=================================================================
Total params: 8,230
Trainable params: 8,230
Non-trainable params: 0
```

Table: Model Summary

The size of the input layer is 133 because there are 133 different patterns in the dataset. Two 'dense' hidden layers are used in the model with 40 neurons each and 'Relu' activation function. The output layer has 30 neurons with 'Softmax' activation function.

Training & Saving the Model

For fitting the data in the model, 1000 epochs are used. In each epoch, the batch-size is 8. The accuracy of the model is 100%. After training, the model is saved. Therefore, there is no need to fit the model in the next run. The datasets and the model simply can be loaded. It will save time.

Making Prediction

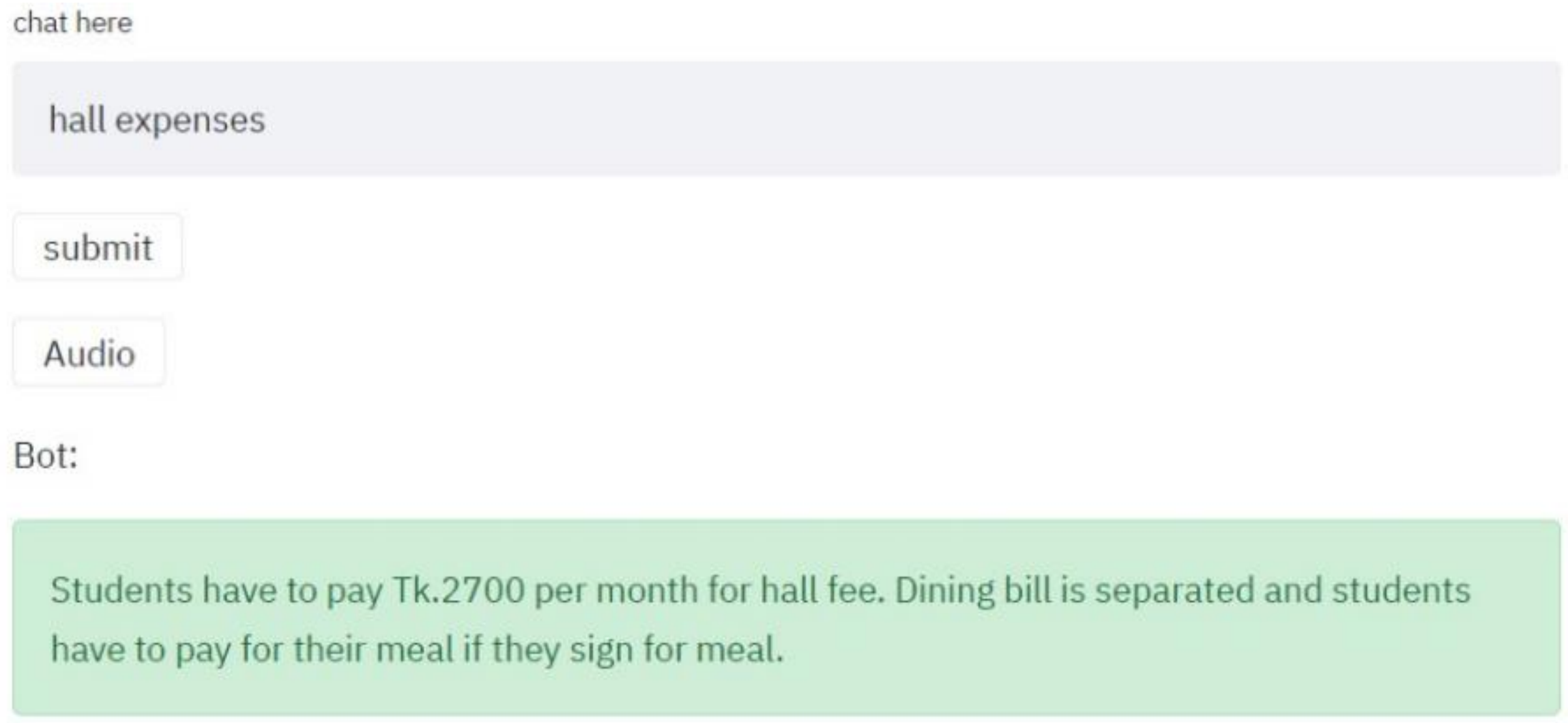
For any input sentence, the model generates a list of probabilities for all of the classes. This makes the process to generate a response look like the following:

- Get some input from the user
- Convert it to a bag of words
- Get a prediction from the model
- Find the most probable class
- Pick a response from that class

Graphical User Interface

In GUI part, Streamlit has been used. Some additional features are implemented in the GUI part:

- Audio Input.
- Database Connection.



Students have to pay Tk.2700 per month for hall fee. Dining bill is separated and students have to pay for their meal if they sign for meal.

Figure2: Snap of GUI Part

Future Work

Future Work:

- Build Similar type of chatbot in "Bangla".
- Make the model more intelligent.